

# 我所遇见的微服务演进这十年

好买财富 - 王晔惊



# 自我介绍

- 好买财富(howbuy.com), 任架构部总监
- EGO会员、企业培训师
- 曾任大智慧测试总监, 建立“云测试平台”
- Sun -> Oracle -> WebLogic -> Cloud
- 经历过2000年网络经济泡沫的程序员
- 我的知乎专栏: kiddwyl(爱运动, 能折腾, 撸代码, 勤分享)

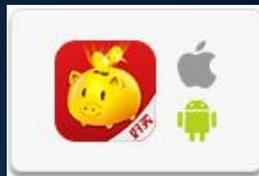


王晔惊

lenovo 联想

Tencent 腾讯

好买 | HOWbuy



储蓄罐



掌上基金



好买财富微信

好买财富是一家专注为个人提供专业理财服务的公司，腾讯和联想旗下的君联资本都是好买的战略股东

- ✓ 2012年，好买获得中国证监会颁发的第一批独立基金销售牌照
- ✓ 2015年成为首家在新三板成功挂牌的独立财富管理公司

# 目录

津津乐道的微服务究竟是什么？

不同时期的服务，不同目标的追求

传统金融服务在微服务转型过程中的那点事



1.



## 用上RPC就是微服务了吗？（如dubbo）

- ① 目的：降低成本，提高效率，屏蔽调用（跨进程）复杂
- ② 职责：远程调用 ≈ 本地调用，分布式服务的表达方式

 RPC是微服务化实现的基础，而dubbo是RPC的一种框架呈现

2.



## 是不是SOA也是微服务的一种？

- ① SOA始于企业领域，目的是划分业务（独立），从紧耦合到松耦合
- ② 微服务始于互联网，更强调分布式，横向伸缩、快速持续或快速升降

 微服务从架构思想的角度来说，是SOA的互联网化

3.

## 用上微服务框架我的系统就松耦合了吗？

- ① 使用微服务框架，帮助你更好地实现系统的每个服务支持独立扩展
- ② 使用微服务框架，帮助你的服务能够得到更适合于服务资源需求的硬件资源

 耦合度松不松，关键看你的系统纵横向怎么拆分，业务层级如何分层

4.

## 大家常说的服务治理就是微服务吗？

- ① 系统服务化不代表需要治理，如单独的业务模块，且调用少，服务治理将显笨重
- ② 如单个模块被多个其他模块依赖，或单个模块需要做主备或者负载均衡，需要对各个服务做治理

 服务规模大了，才需要服务治理，并需要‘带有服务治理功能的框架’，dubbo就是其中之一

在我看来，微服务是这样的...

1

微服务是‘互联网+时代’催生的设计思想/理念，并非多么高大上的技术手段

2

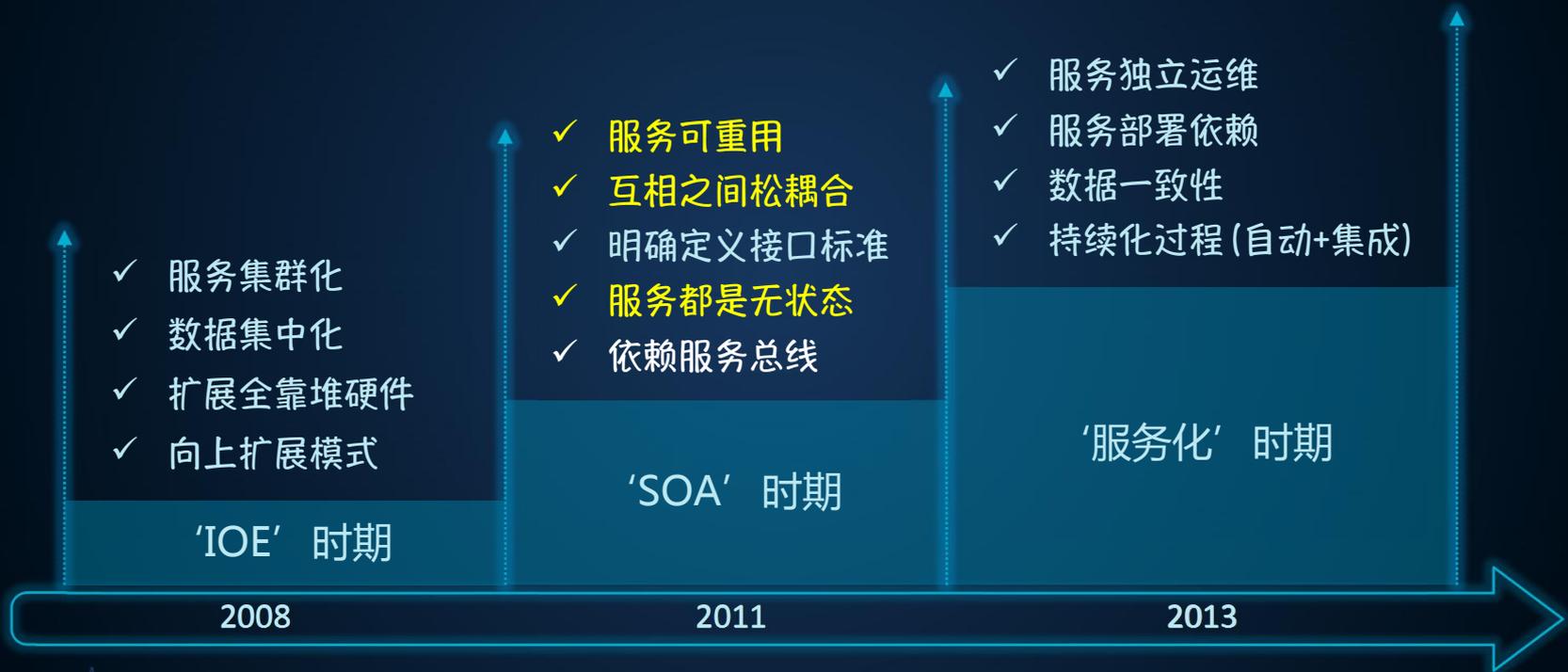
微服务并不是一场技术革命，更像是一场对‘人’的革命

3

微服务给业务带来‘福利’的同时，却给传统测试与传统运维带来了一场‘革命’



# 不同时期的服务，不同目标的演进



1

'IOE' 时期 - 面向单一的项目化客户群

国内知名金融公司的CRM系统项目

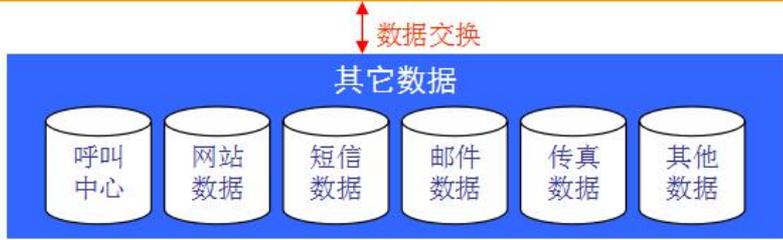
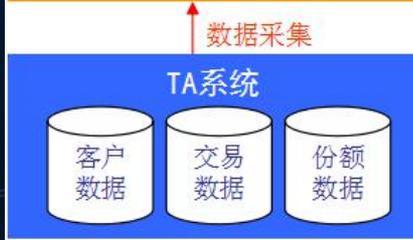
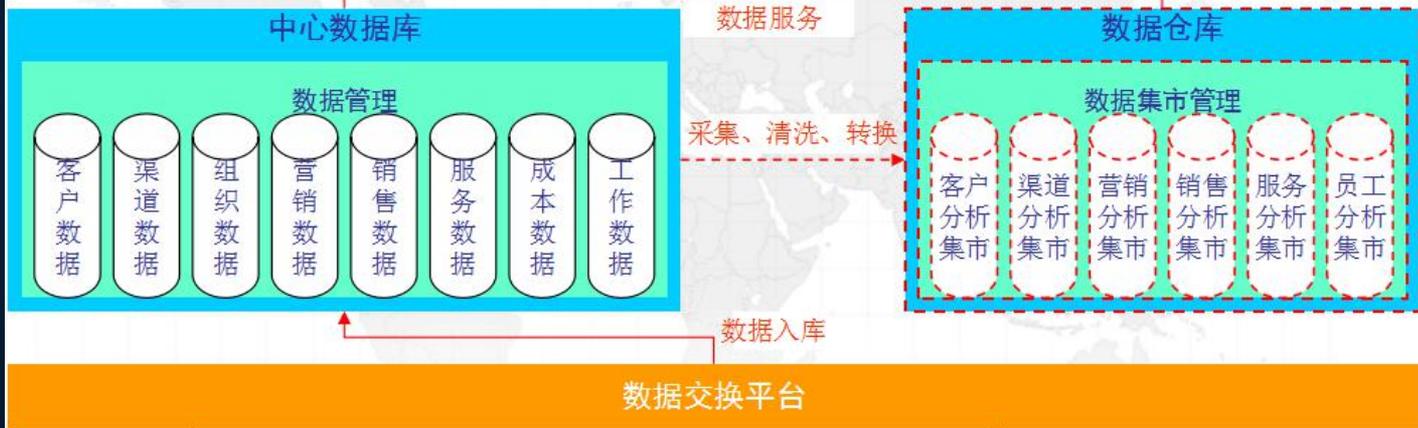
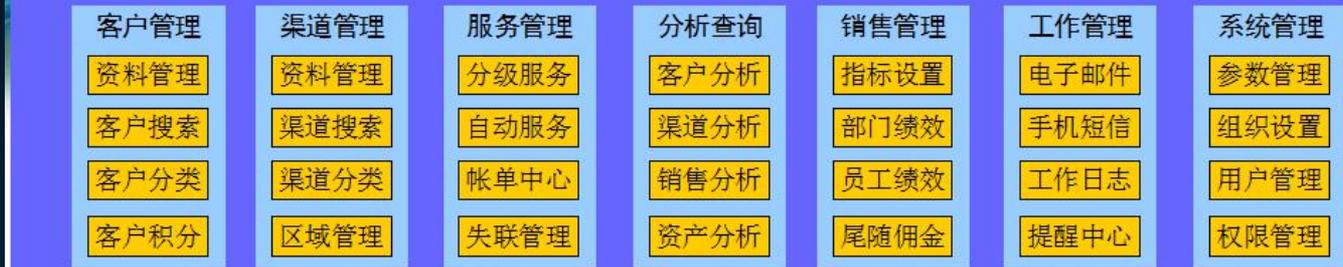


# 不同时期的服务，不同目标的通

沪江技术学院

IT大咖说  
知识分享平台

## 客户关系管理系统

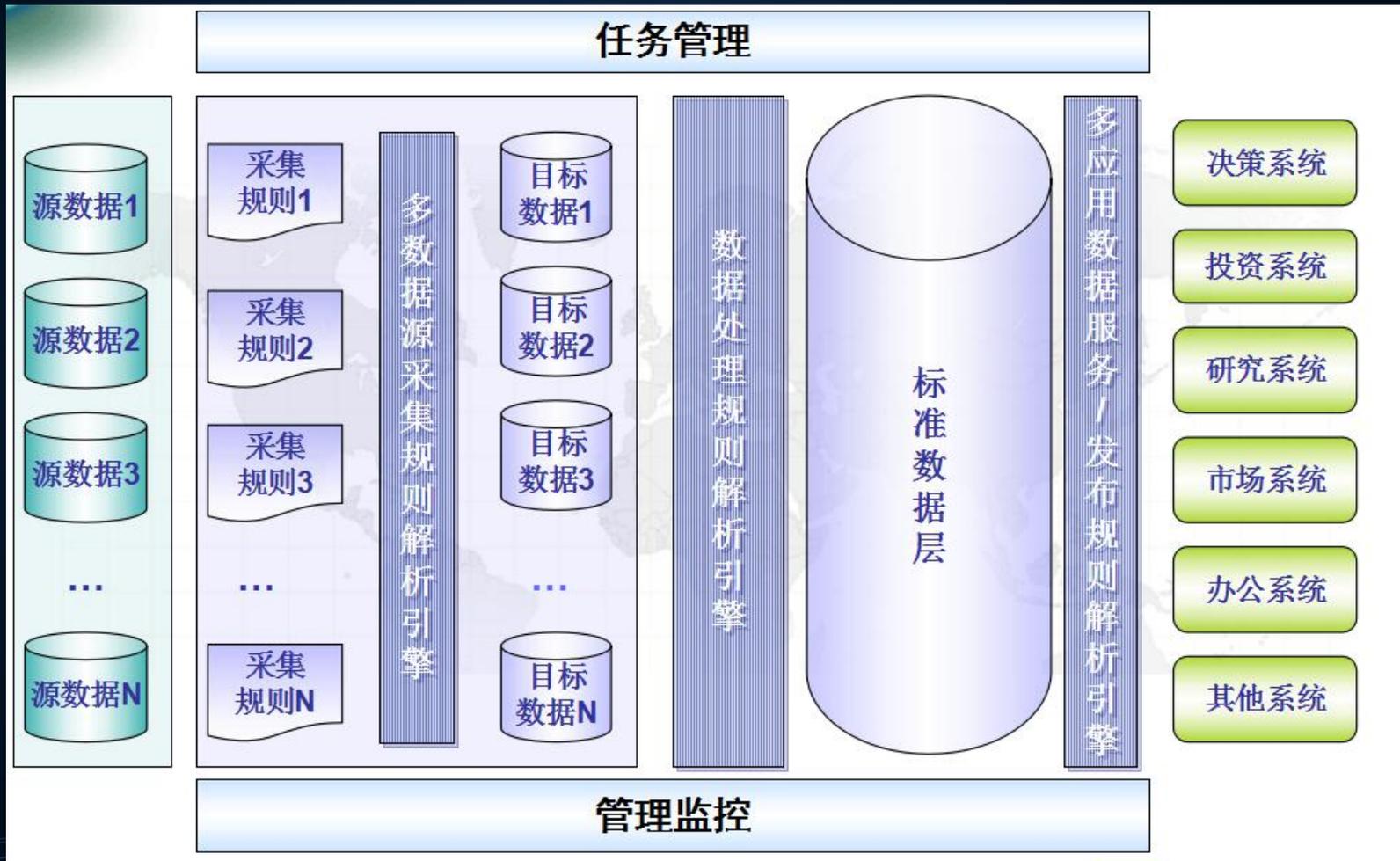


1

整体解决方案

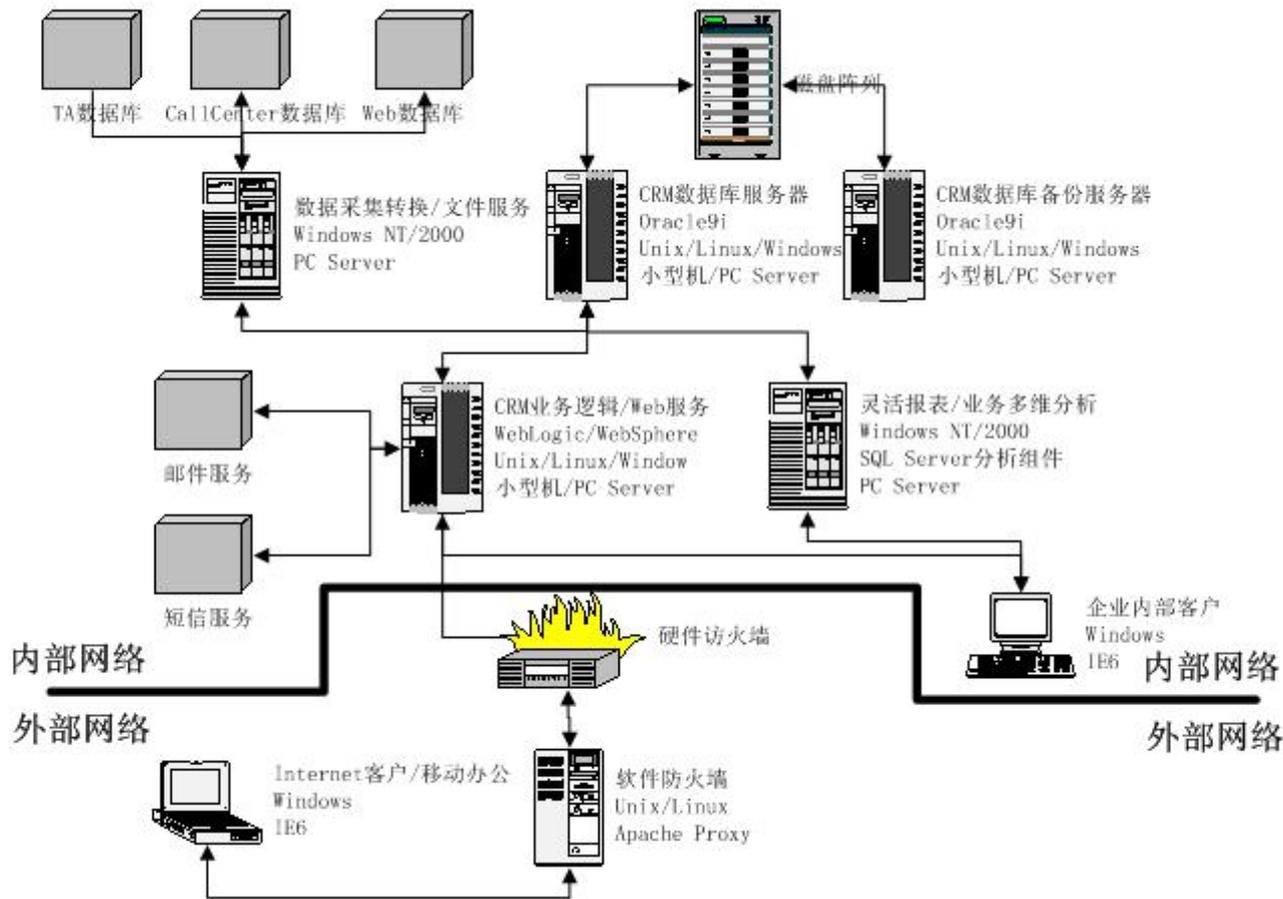
1

## 系统业务逻辑



## 1

# 技术平台部署



## 1 'IOE' 时期 - 面向单一的项目化客户群

### 技术基因：

- ✓ Oracle：1000W用户，每日批处理在3小时内完成
- ✓ WebLogic：稳定性，排障
- ✓ IBM Power：AIX需要玩的溜，出问题迅速解决
- ✓ EMC：良好的存储方案，让他更快一些

### 决定因素：

- ✓ 能力 - '超级' 技术个体
- ✓ 共用 - 丰富多彩，而且好用的开发组件
- ✓ 业务 - 快速理解需求，不需要抽象，实现就成
- ✓ 编码 - 快速敲打键盘，最好写个生成器



2

'SOA' 时期 - 面向多元的产品化需求源

随着客户增多，从项目化转变为产品化





3

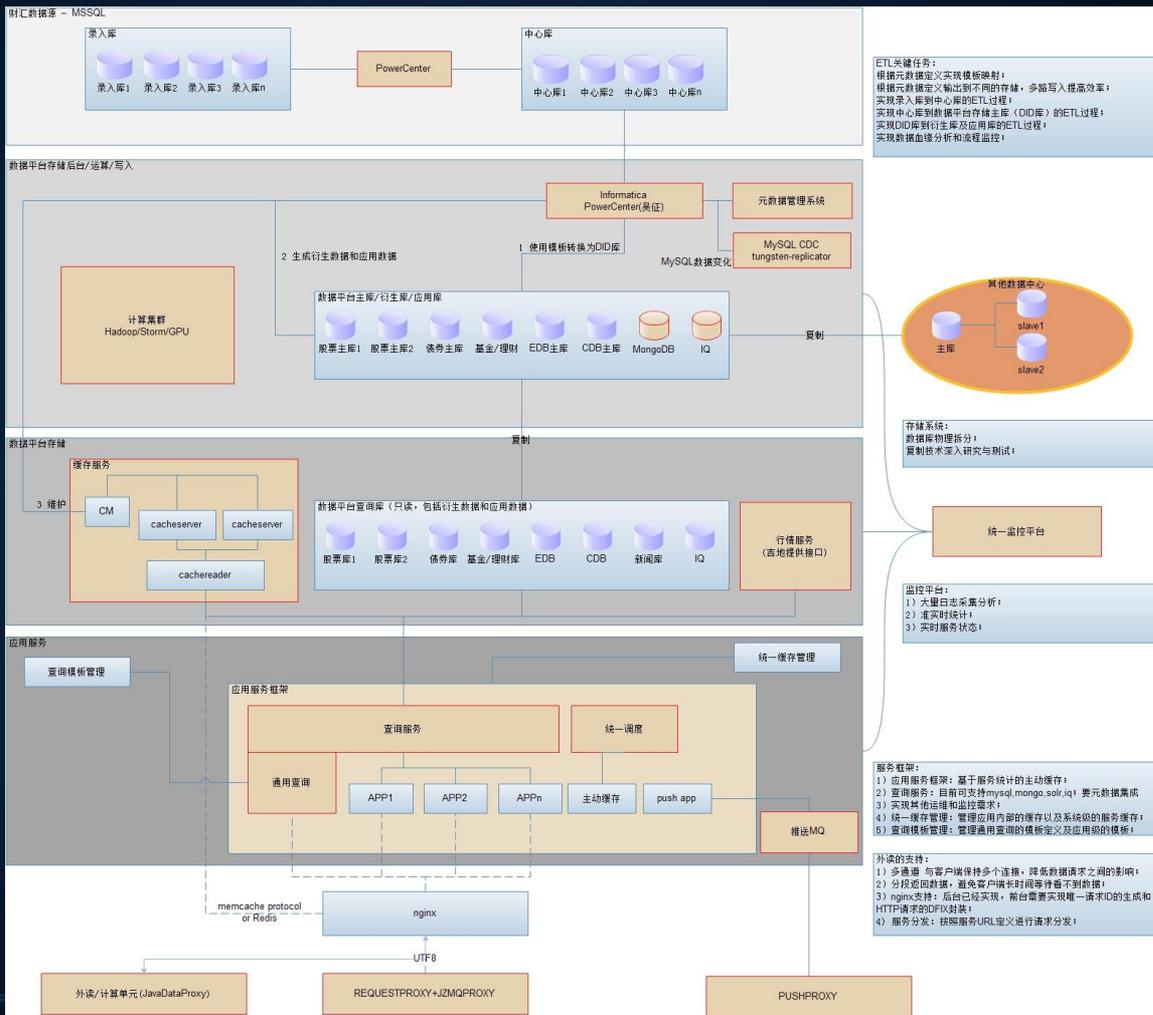
‘服务化’ 时期 - 面向独立的互联网业务线

多条业务线，需求各不同，小步迭代，快速试错



## 3

# 系统架构图



## 3

## ‘服务化’ 时期 - 面向独立的互联网业务线

给我们带来了：

- ✓ 应用开发简单，数据模板化
- ✓ 技术栈多样，利用各自的特点解决问题
- ✓ 独立资源扩展，可按业务线自动化横向扩展
- ✓ 可用性较高，具有多层次及冗余的技术保障

让我们失去了：

- ✓ 运维成本居高不下，矛盾与日俱增
- ✓ 技术栈较多，给(持续)部署与(快速)排障带来了难度
- ✓ 过多的链路，数据一致性
- ✓ 未使用虚拟化，交付速度大打折扣



可能大家会想：故事听完了，现在的微服务你是怎么做的？



## A-为什么需要转向微服务?

- 互联网化的业务需求
- 互联网化的用户行为
- 快速交付与快速排障

## A-传统金融转微服务难在哪?

- 业务驱动强
- 历史孤岛多
- 资源投入少
- 全产品的业务复杂性

A-金融  
H-好买

## H-目前给运维带来了什么?

- 服务之间的依赖
- 服务异常排查效率提升
- 服务水位分析

## H-目前给研发带来了什么?

- 可以向测试甩锅
- 只需关注业务本身
- 技术栈单一



## H

## 设计思想

运营

透明部署、持续交付、统一监控、统一配置

平台

容灾/降级、负载均衡、灰度管理

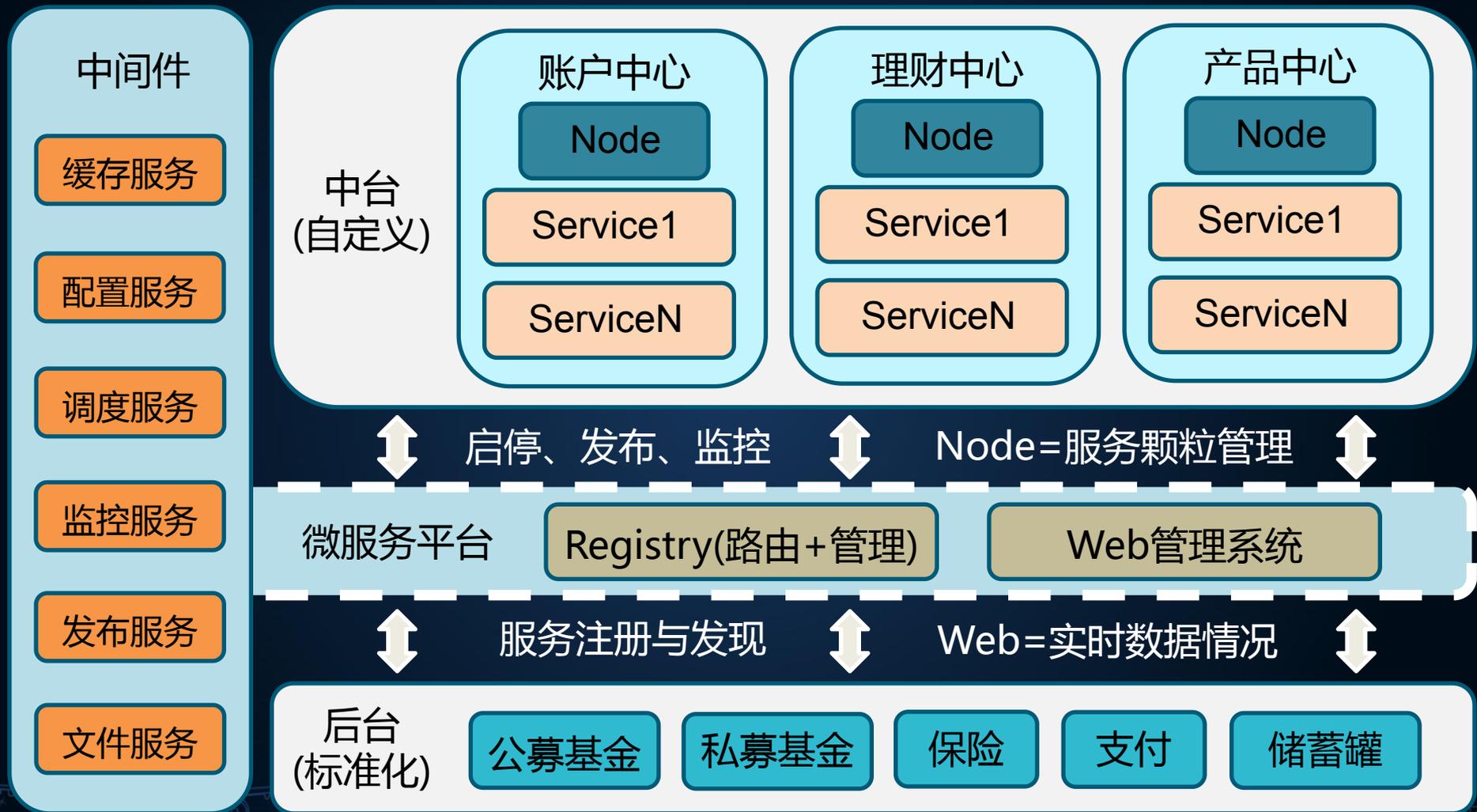
框架

RPC开发框架、中间件无缝连接

资源

快速扩展、容器化

# 传统金融服务在微服务转型过程中的





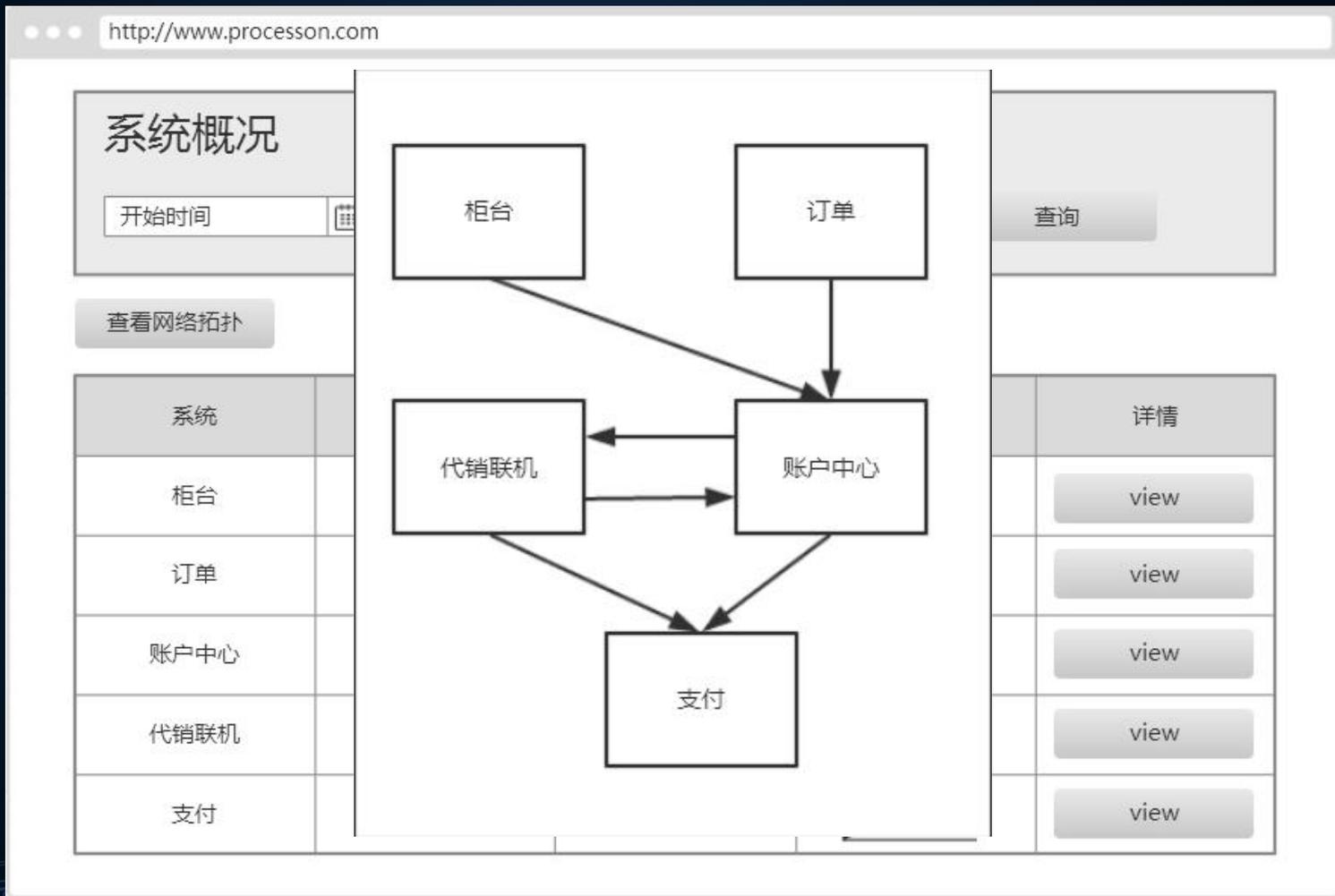
## 部分功能

目前提供些什么？



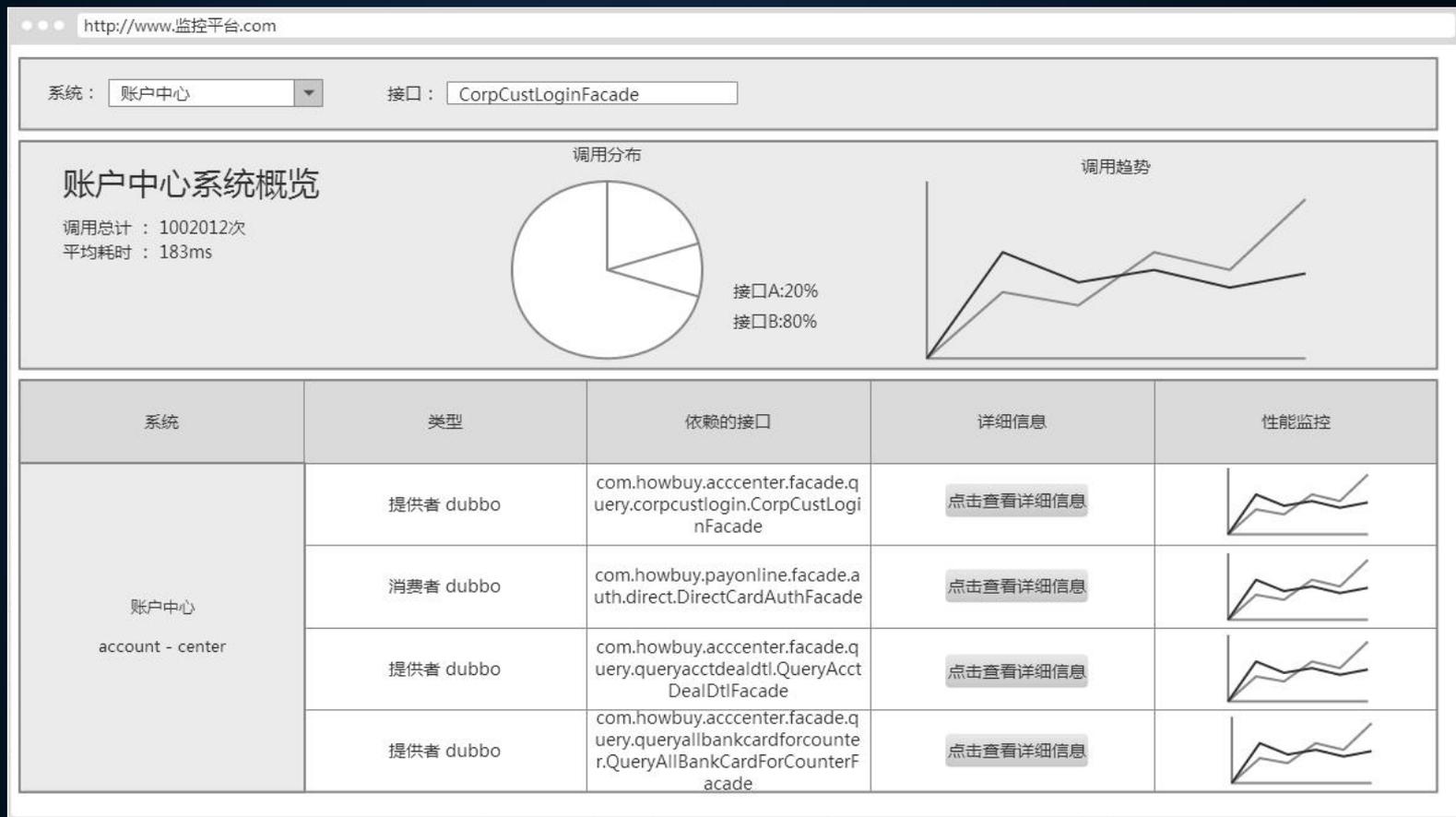
4

## 系统拓扑图



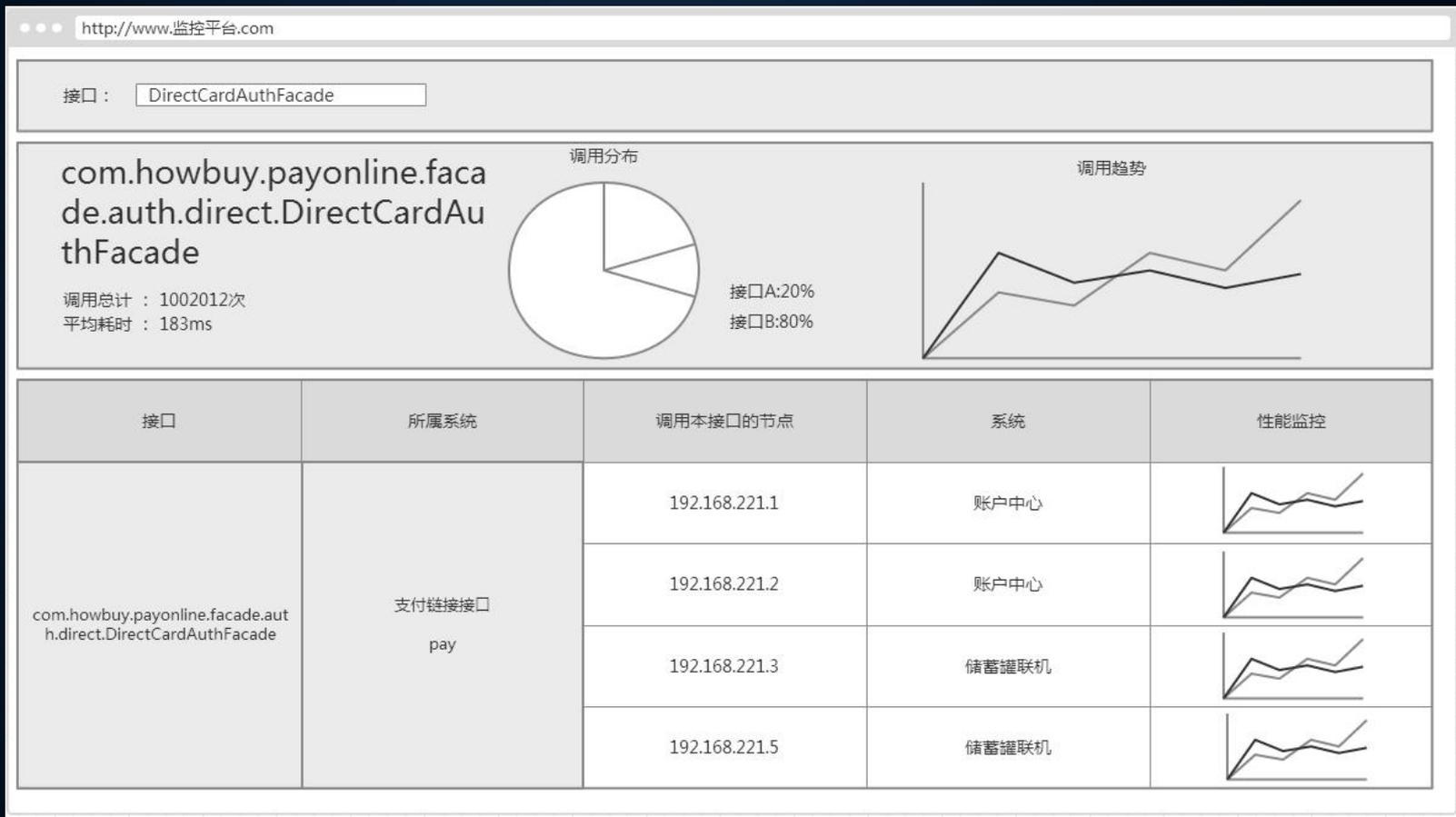
# H

## 接口 依赖 关系



# H

## 压力实时分析



H

链路调用关系





## 部分功能

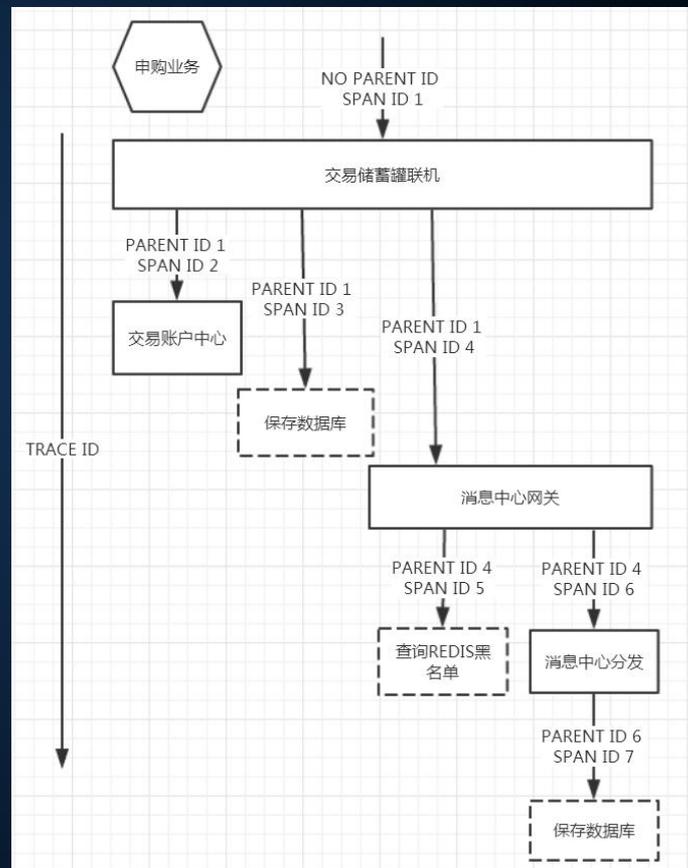
如何实现的呢？



- 基于谷歌的Google Dapper为蓝本
  - 埋点：跟踪树和span
  - 采集：监控服务-Agent
  - 统计分析：监控服务-分析模块
  - 图表展现：基于企业级开发平台开发
  - 告警：短信、微信



- 接口调用：DUBBO、HTTP、WEBSERVICE、RMI
- 技术栈：SPRING
- 链路埋点：
  - TRACE ID 请求ID
  - PARENT ID 父级ID
  - SPAN ID 本级ID



天下大事，分久必合，合久必分



集群式时代

分布式时代

大数据时代

区块链时代



Thank you!  
感谢聆听



王晔惊@好买财富

中国



扫一扫上面的二维码图案，加我微信