服务化框架
技术选型实践

章 耿

中生代技术
FRESHMAN TECHNOLOGY

- 前言
- 服务化框架构成
- 京东实践
- 总结

前言

- 不讨论为什么要服务化
- 不讨论微服务和SOA区别
- 不讨论哪个技术最优

- 前言
- 服务化框架构成
- 京东实践
- 总结

基本的服务化框架构成

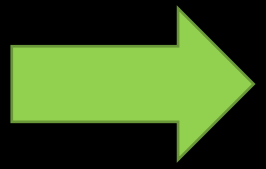
管理端

服务注册中心

统一
RPC框架

RPC框架基本考量

- 代码规范
- 通讯协议
- 序列化协议
- IO模型
- 负载均衡



- 学习成本（包括文档数，社区热度）
- 跨语言需求
- 可扩展性（接口变更时）
- 性能

常见的开源RPC框架比较

	thrift	RESTful	dubbo	gRPC
代码规范	基于Thrift的IDL生成代码	基于JAX-RS规范	无代码入侵	基于.Proto生成代码
通讯协议	TCP	HTTP	TCP	HTTP/2
序列化协议	thrift	JSON	多协议支持默认hessian	protobuf
IO框架	Thrift自带	Servlet容器	Netty3	Netty4
负载均衡	无	无	客户端软负载	无
跨语言	多种语言	多种语言	Java	多种语言
可扩展性	差	好	好	差

其它：SOAP、RMI、Hessian、ICE等

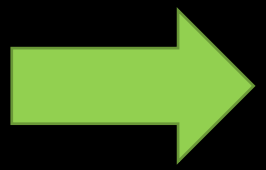
RPC框架选型小结

- 要与前端交互的，适合RESTful、gRPC
- 纯粹后端交互的，适合thrift、dubbo等

- 小公司、新公司选择规范化框架，thrift、RESTful、gRPC
- 已有大量代码的公司选择无代码入侵，dubbo、RESTful

注册中心基本考量

- 服务注册
- 服务订阅
- 状态检测



- 学习成本
- 维护成本
- 数据结构
- 性能
- CAP原则

常见的开源服务发现

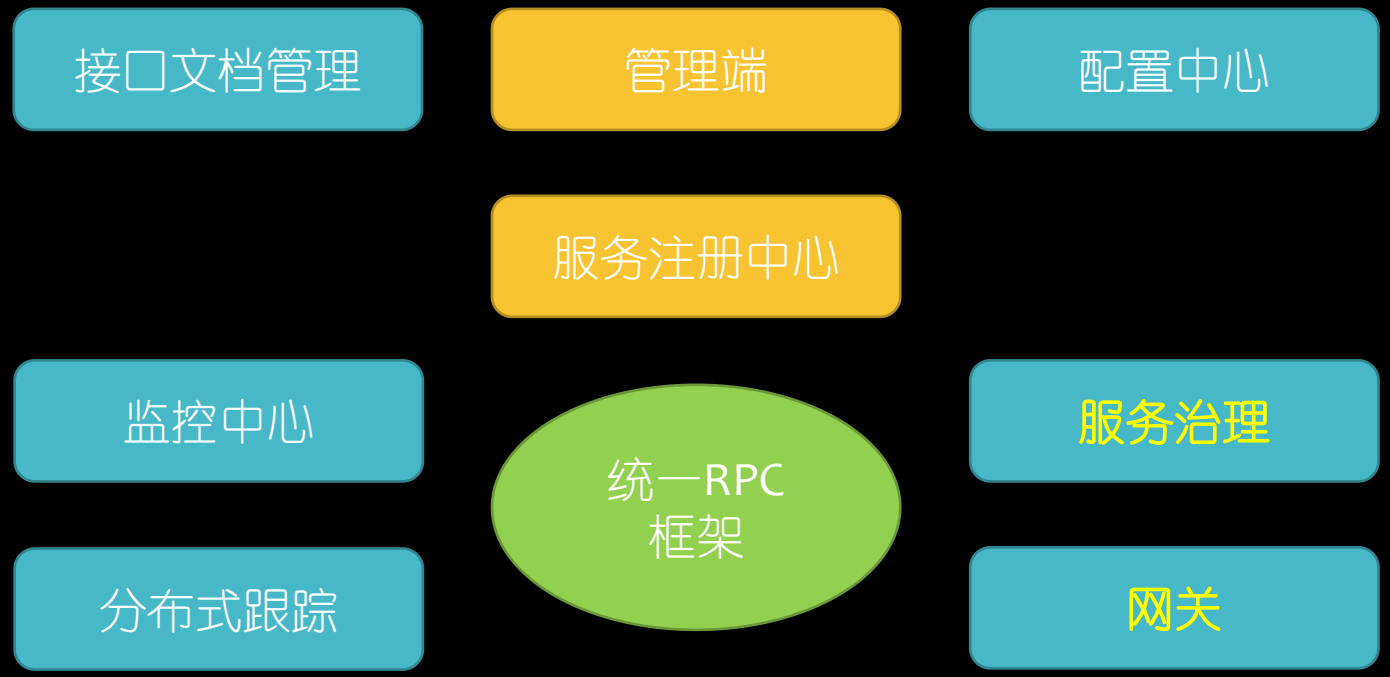
	ZooKeeper	etcd	Consul	Eureka
一致性	强一致性 paxos	强一致性 Raft	强一致性 Raft	弱一致性
数据结构	Tree	K/V	K/V	K/V
通讯协议	TCP	HTTP、gRPC	HTTP、DNS	HTTP
客户端	ZKClient	/	/	Eureka-client
CAP	CP	CP	CP	AP

其它：Redis、MySQL

注册中心选型小结

- 规模小选择CP，RPC框架可以直接接入数据源
- 规模大选择AP， RPC框架不可以直接接入数据源
- 存在跨机房，跨地域的尽量不要选有强一致性协议的系统
- RPC框架必须要有注册中心不可用的容灾策略
- 服务状态检测十分重要

完善的服务化框架构成



网关

- 鉴权
- 限流
- 协议转换
- Mock
- 其它统一处理逻辑（例如请求解析、响应包装）

服务治理

- 服务路由：权重、IP路由、分组路由、参数路由、机房路由等
- 调用授权：应用授权、token、黑白名单
- 动态分组：服务端切分组、客户端切分组
- 调用限流：服务端限流、客户端限流
- 灰度部署：灰度上线、预发标识、接口测试
- 配置下发：接口配置、全局配置
- 服务降级：Mock、熔断、拒绝服务

- 前言
- 服务化框架构成
- 京东实践
- 总结

第一代SAF背景（2012）

- 各个部门框架不统一
- 已有较多代码
- 接口规模在1K左右
- 服务节点在50K左右
- 机器规模在8K左右
- 机房拓扑简单

第一代SAF选择（2012）

- RPC框架：dubbo，做配置扩展，以及功能扩展包括rest（resteasy）、webservice（cxf）、kryo序列化、thrift序列化、调用压缩等
- 注册中心：Zookeeper，RPC框架直接接入数据源
- 监控中心：监控服务+HBase
- 管理平台：读取Zookeeper做管理平台

第二代JSF背景（2014）

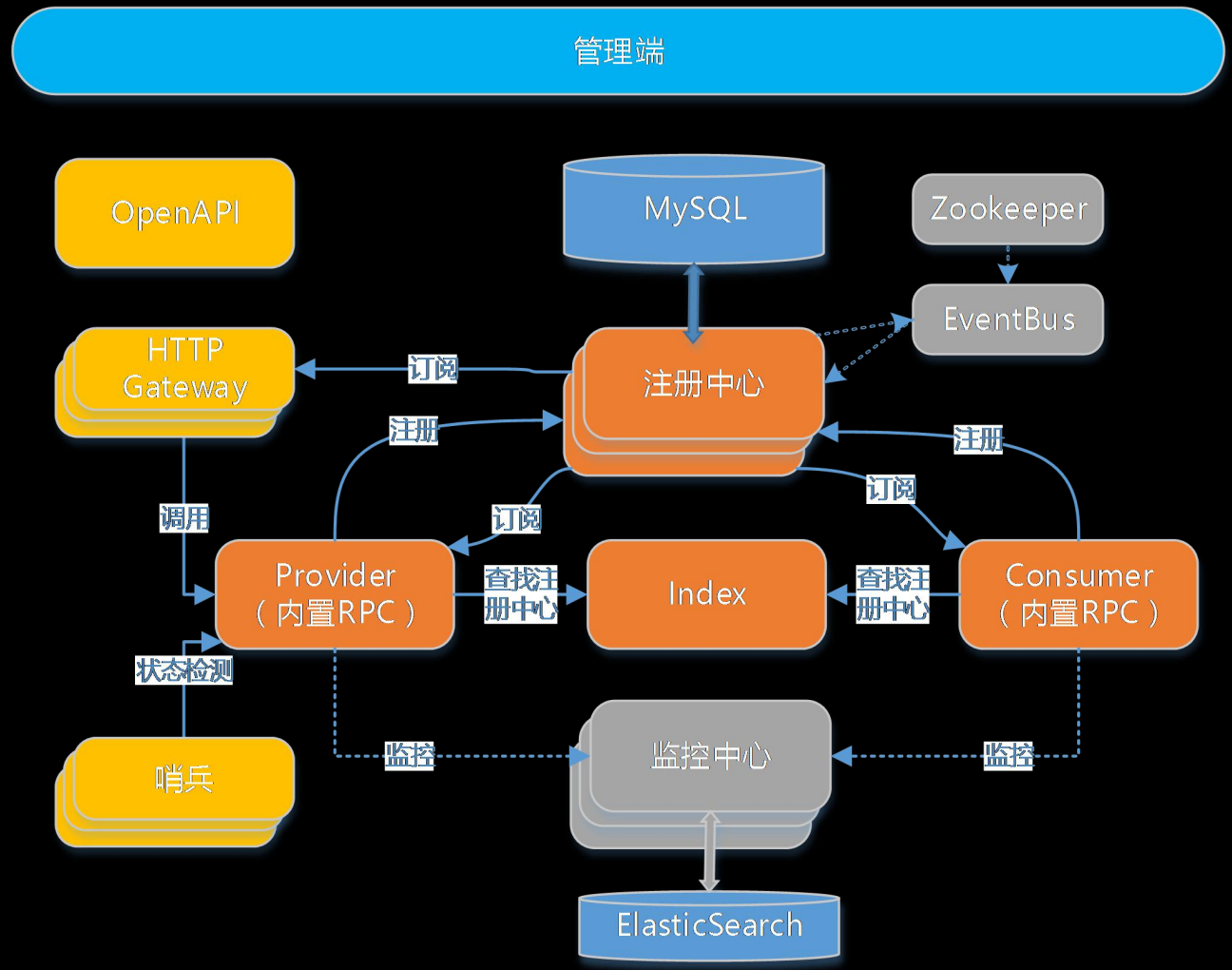
- 业务的不断增长
- 接口、机器数量级增长
- 多机房问题
- 跨语言问题
- 服务治理需求

第二代JSF选择（2014）

- RPC框架：自研框架，兼容dubbo协议
- 注册中心：自研，基于DB作为数据源，前置Index服务
- 监控中心：监控服务+InfluxDB（后来改为ElasticSearch）
- 管理端：基于DB，功能更强大
- HTTP网关：自研

- 开发周期：7人/年　2014.1~2015.1
  包括开发、测试、预发、上线

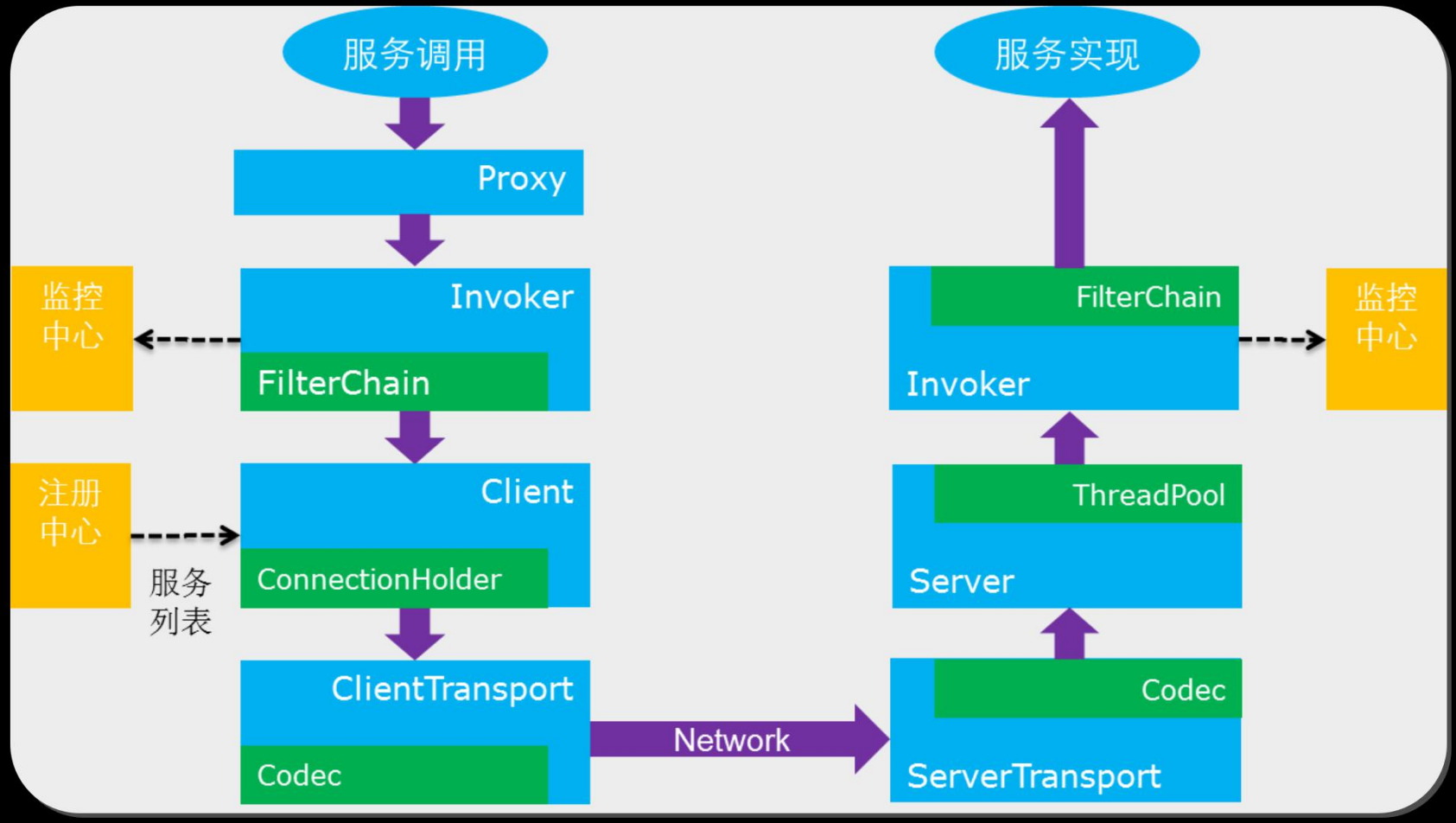
JSF架构简图



JSF 注册中心

- 引入Index服务概念
- 通过数据库保证数据一致性保证，注册中心无状态，且内存有服务列表全量缓存，连不上数据库也保证可读
- 数据库的数据结构更适合各种维度展示、过滤、分析等
- 注册中心就是个JSF服务，监控到压力大即可进行动态水平扩展
- 服务列表推送逻辑改进
- 注册中心与RPC客户端可各种交互

JSF RPC框架



JSF RPC框架

- Config：**Spring**/API/Annotation
- Proxy: **Javassist**/JDK
- Invoker/Filter：内置+自定义
- Client：**Failover**/FailFast/TransportPinpoint/MultiClientProxy
- 调用方式：**同步**/异步并行/异步回调/Callback/泛化
- Loadbalance：**Random**/Roundrobin/ConsistentHash/ LocalPreference/LeastActiveCall
- 路由：参数路由，分组路由，（IP级别路由逻辑在注册中心做）
- 长连接维护：可用/死亡/亚健康

JSF RPC框架

- 协议：JSF/SAF(dubbo)/HTTP/Telnet/HTTP2
  - 第三方：REST/Webservice
- 序列化：**MsgPack**/Hessian/Json/Java/protobuf(c++)
- 压缩：Snappy/LZMA
- 网络：基于Netty4，长连接复用
- 线程模型：BOSS+WORKER+BIZ
- 容灾：本地文件
- 请求上下文：IP，参数，隐式传参
- 事件监听：响应事件，连接事件，状态事件

JSF 管理平台

JSF HTTP网关

- 基于Netty4实现
- 方便跨语言通过HTTP+JSON调用JSF服务
- 解决单点问题，无需自己设置VIP

JSF遇见京东弹性云（Docker）

- 硬件指标
- 网络
- 轻量
- 快速

JSF目前规模

- 接口数：万级
- 实例数：百万级
- 接入IP数：十万级

- 框架调用量：每天千亿级别
- 监控数据：每天70亿条数据， 800G数据量
- HTTP网关：每天百亿级别

- 前言
- 服务化框架构成
- 京东实践
- 总结

没有最好，只有最适合！

It’s just the beginning！

谢 谢！