



APACHE
APISIX



中东社交产品 Beeto 的 Apache APISIX网关应用实践

胡丽麟

Beeto 平台研发总监



APACHE
APISIX



CONTENT

01 背景介绍

02 设计思路

03 应用实践



APACHE
APISIX



01

中东社交产品现状 & Beeto 的产品形态

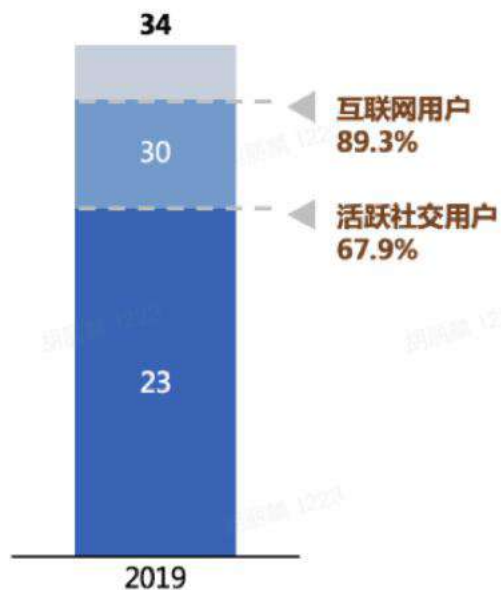


APACHE
APISIX

中东社交产品现状

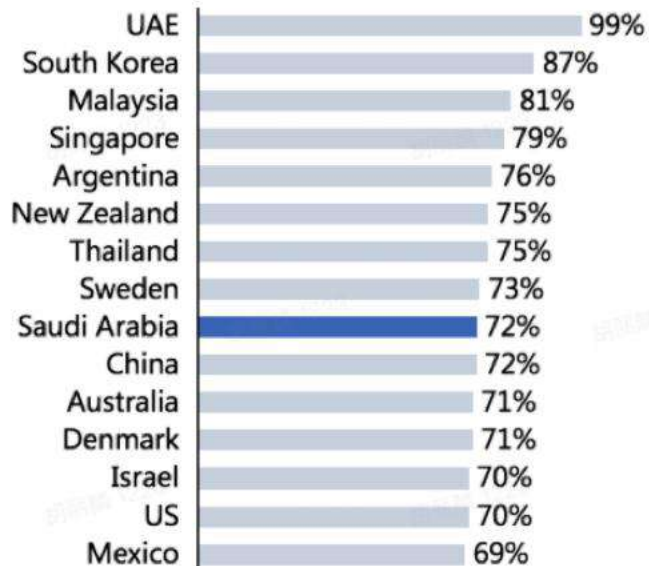
1 2019年沙特互联网渗透率~90%，社交用户占比72%

2019年沙特互联网及社交网络用户数 百万



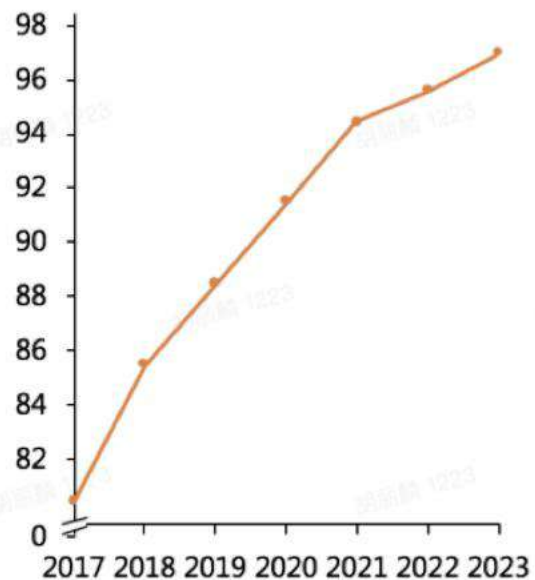
2 2020年，社交活跃渗透率位居前列

2020年1月各国社交网络活跃用户渗透率 %



3 预计未来将持续保持发展

沙特互联网渗透率发展及预测 %





APACHE
APISIX

中东社交产品现状

本地化诉求



去麦加朝圣、
利雅得season、
斋月等本地化事件



埃及沙特内容取向
宗教约束
审核标准差异



纯阿语的app环境
从右往左的阅读习惯



成熟的互联网市场
完整的社交产品体验
体验至上



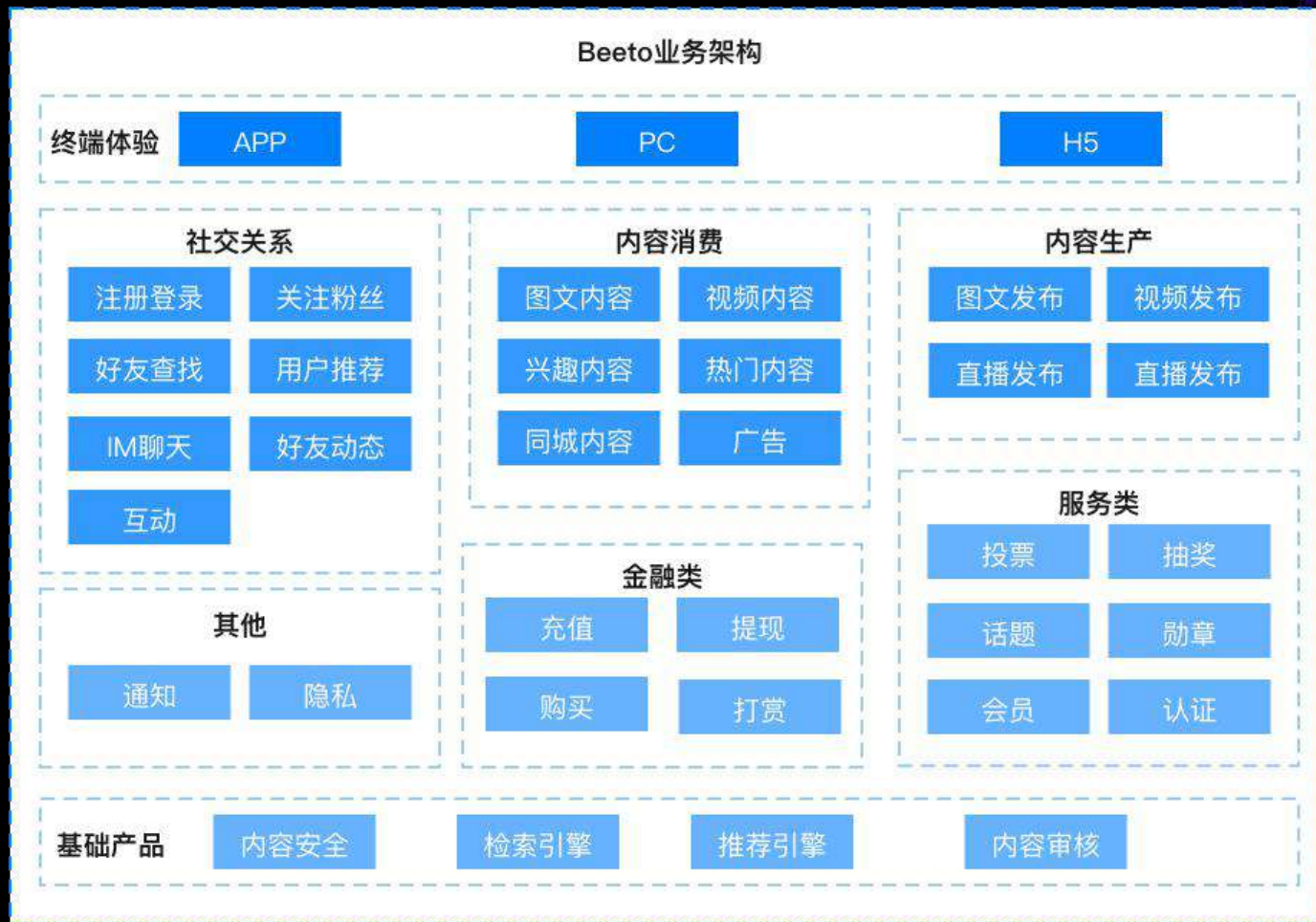
APACHE
APISIX

Beeto的产品形态

Beeto业务架构

- 成熟的社交产品应该具备的业务
- 满足用户需求和平台引导、监管目的

做最大的阿拉伯社交媒体平台、
阿语世界获取每日新鲜事的首选平台、
最好的阿语社区





APACHE
APISIX



02

设计思路

单体服务的架构设计

Beeto的多服务落地复杂度

引入Apisix网关的架构整合



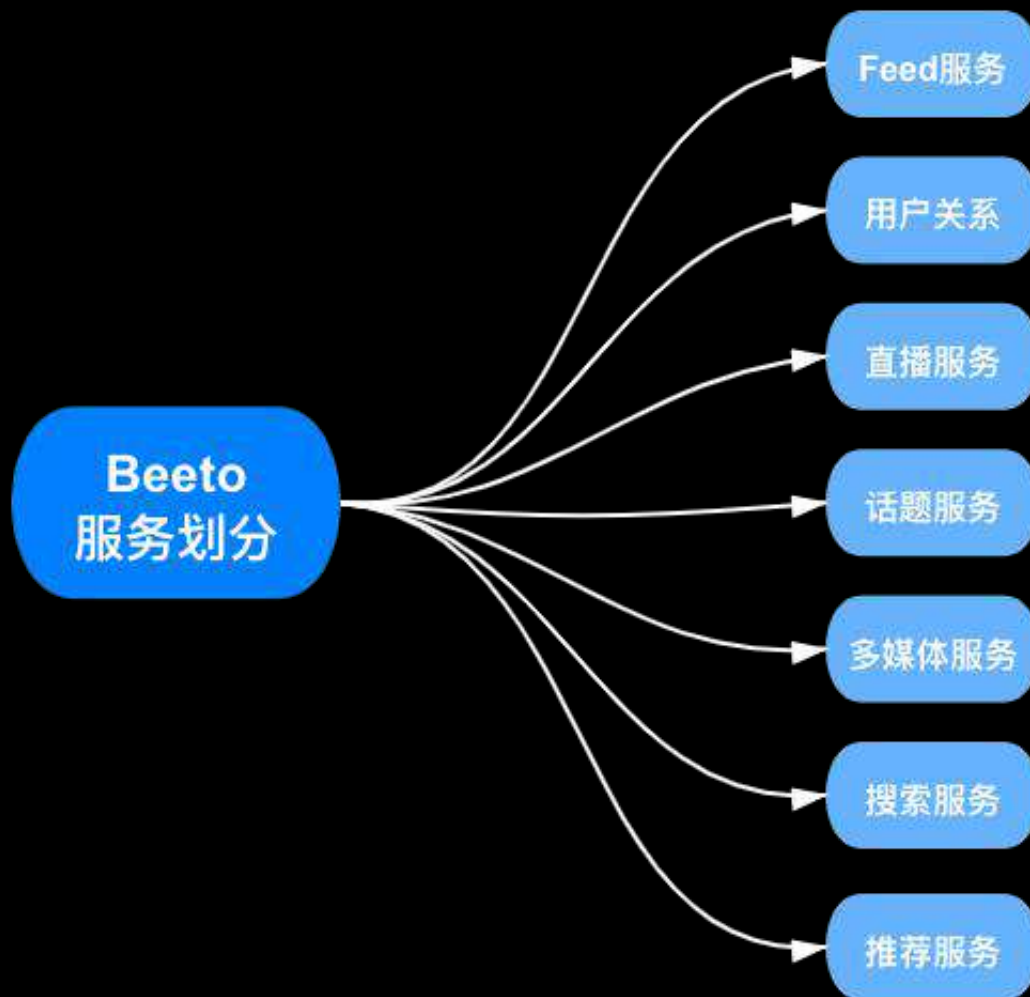
APACHE
APISIX

设计思路

Beeto服务划分

- 按照完整的社交产品进行总体架构设计
- 将不同的业务按照服务进行拆分

业务系统加基础系统十几套服务
创业公司的人力投入和当前系统的
负载规模





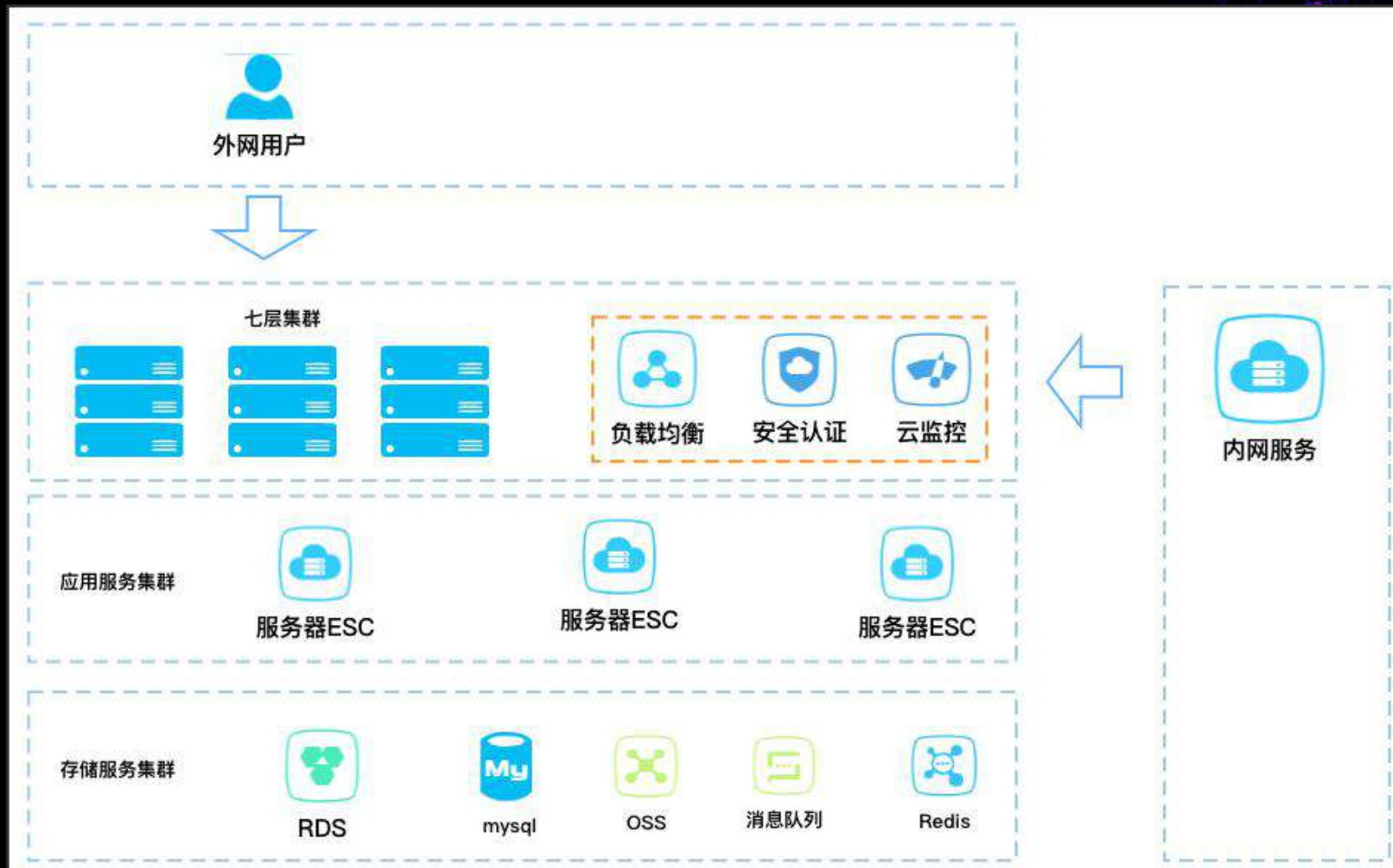
APACHE
APISIX

单体服务的架构设计

架构设计

- 支持公网用户访问（南北流量）
- 支持内网服务互相调用（东西流量）
- 利用七层集群做负载均衡、安全认证和相关监控

每个服务都是独立的系统，有完整的部署架构





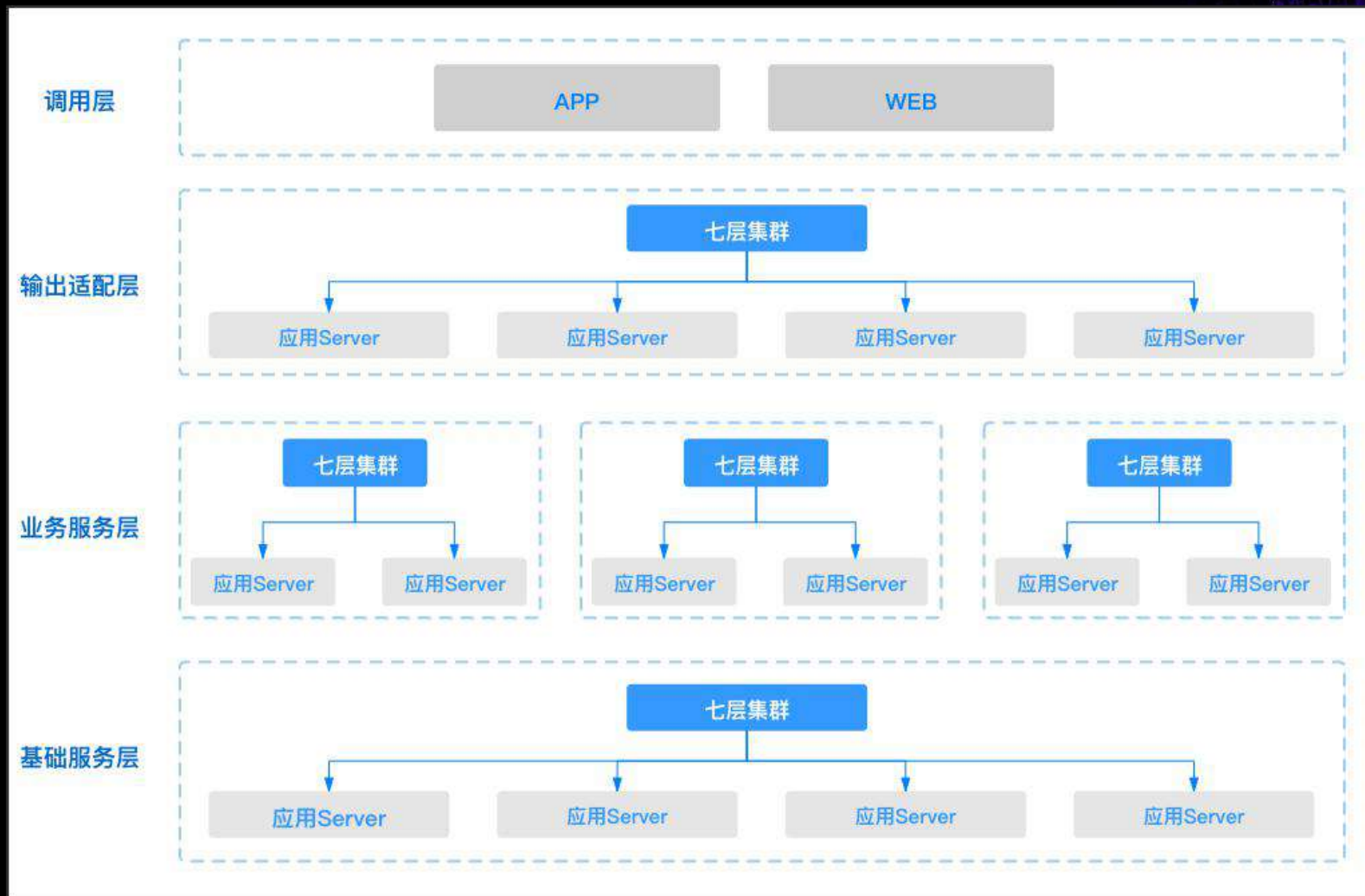
APACHE
APISIX

单体服务的架构设计

所有服务组成整体架构

- 每一个服务都独立部署
- 每个服务都有南北和东西流量
- 都需要做负载均衡和认证控制
- 需要七层集群

服务器成本、部署成本、后期维护成本巨大





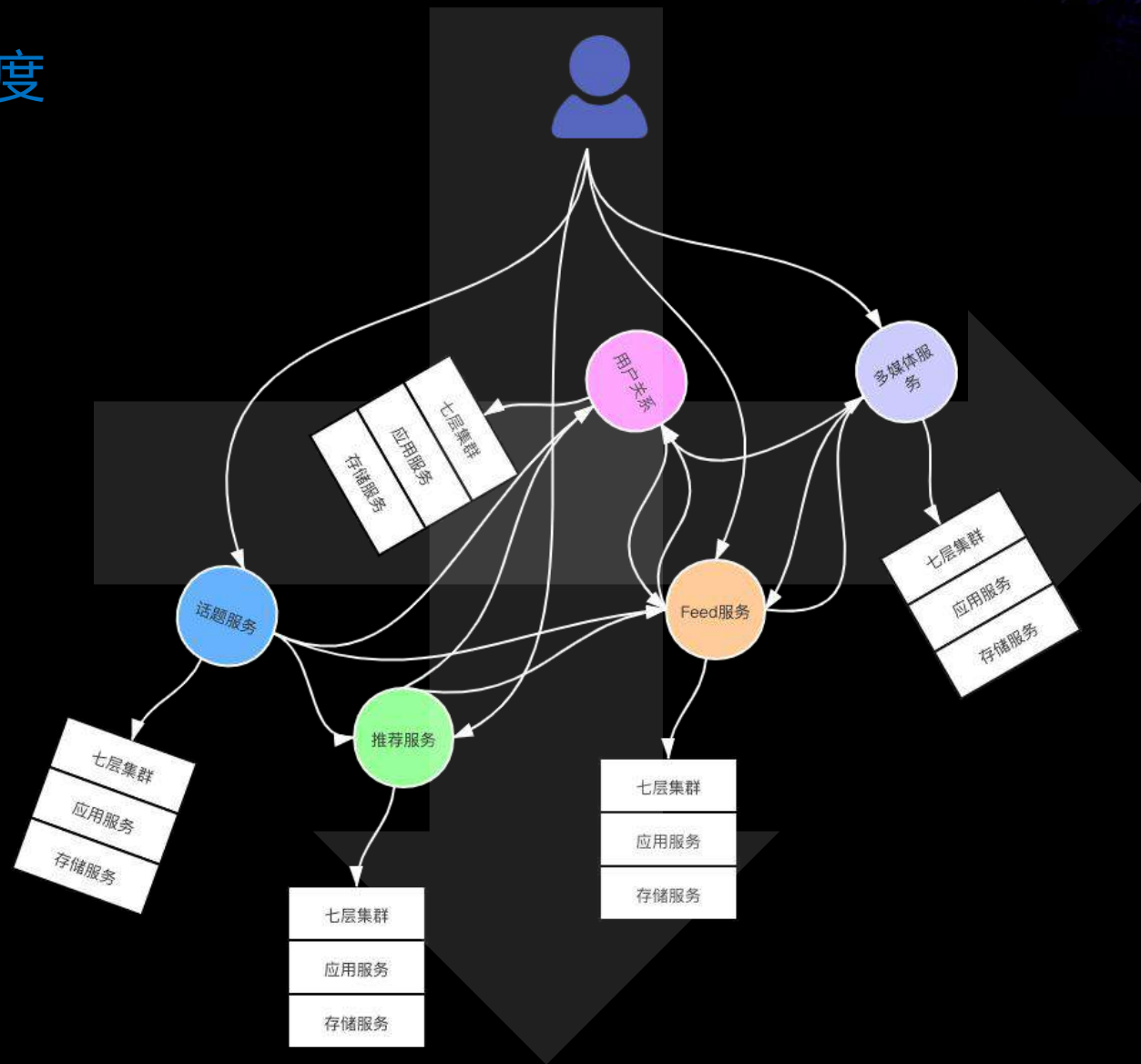
APACHE
APISIX

Beeto多服务落地复杂度

服务之间复杂的调用关系

- 南北向流量分散到各个服务
- 东西向流量交错在各服务之间
- 网状调用关系

每一套服务都需要去维护南北向和东西向流量的调用





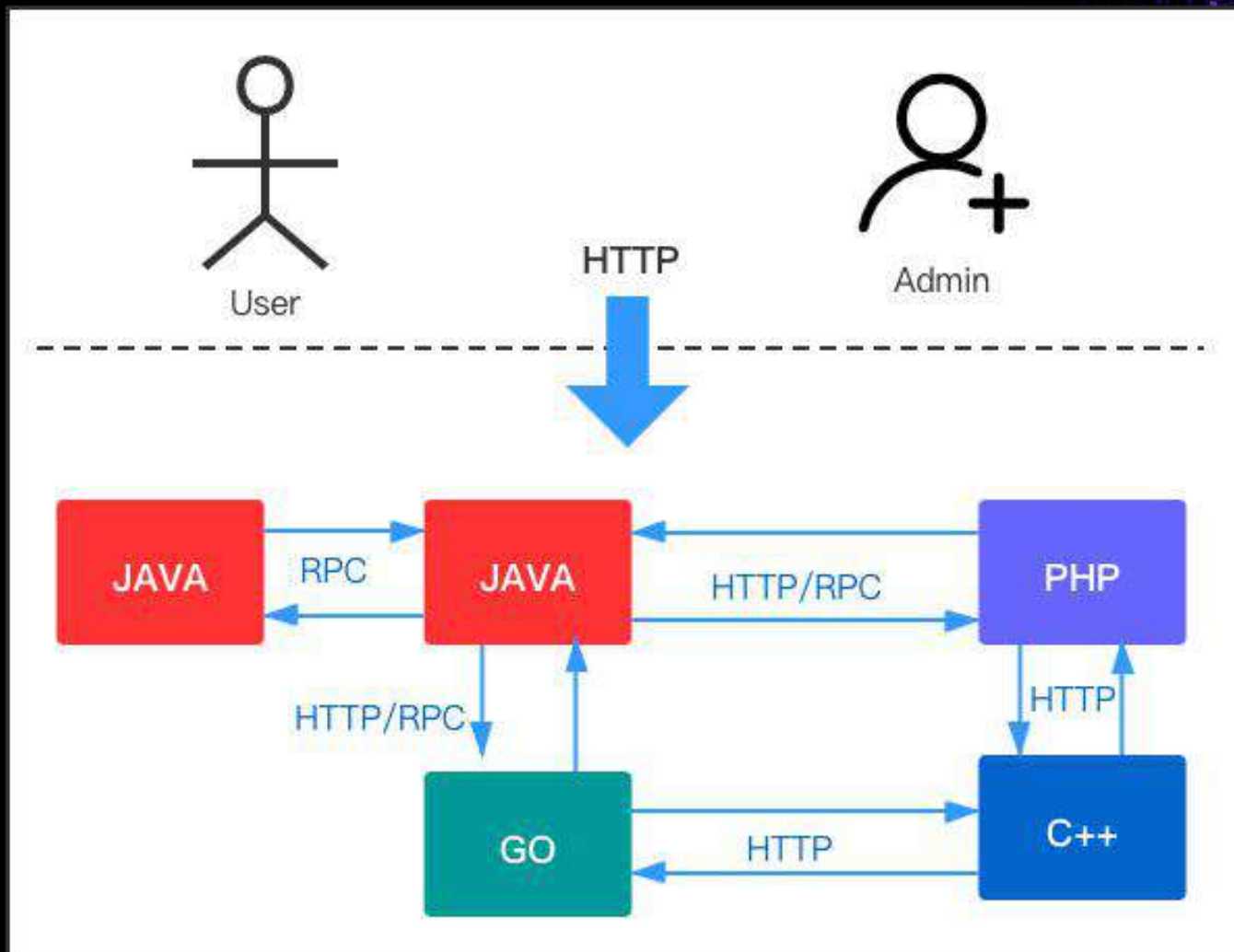
APACHE
APISIX

Beeto多服务落地复杂度

服务的技术栈差异

- 调用协议：HTTP、RPC
- 开发语言：JAVA、PHP、GO、C++

*同一服务可能既有HTTP又有RPC
服务之间背景不同使用的技术栈不同*





APACHE
APISIX

Beeto多服务落地复杂度



部署成本

若干套七层服务部署成本
每个服务配套的监控报警，
路由配置，节点伸缩等管
理成本



资源成本

每套七层服务部署投入的
服务器
服务差异导致流量不均衡，
资源利用率不够，资源浪
费



维护成本

无法统一管理、运维监控、
配置变更等都需要投入大
量人力成本，维护复杂度
高

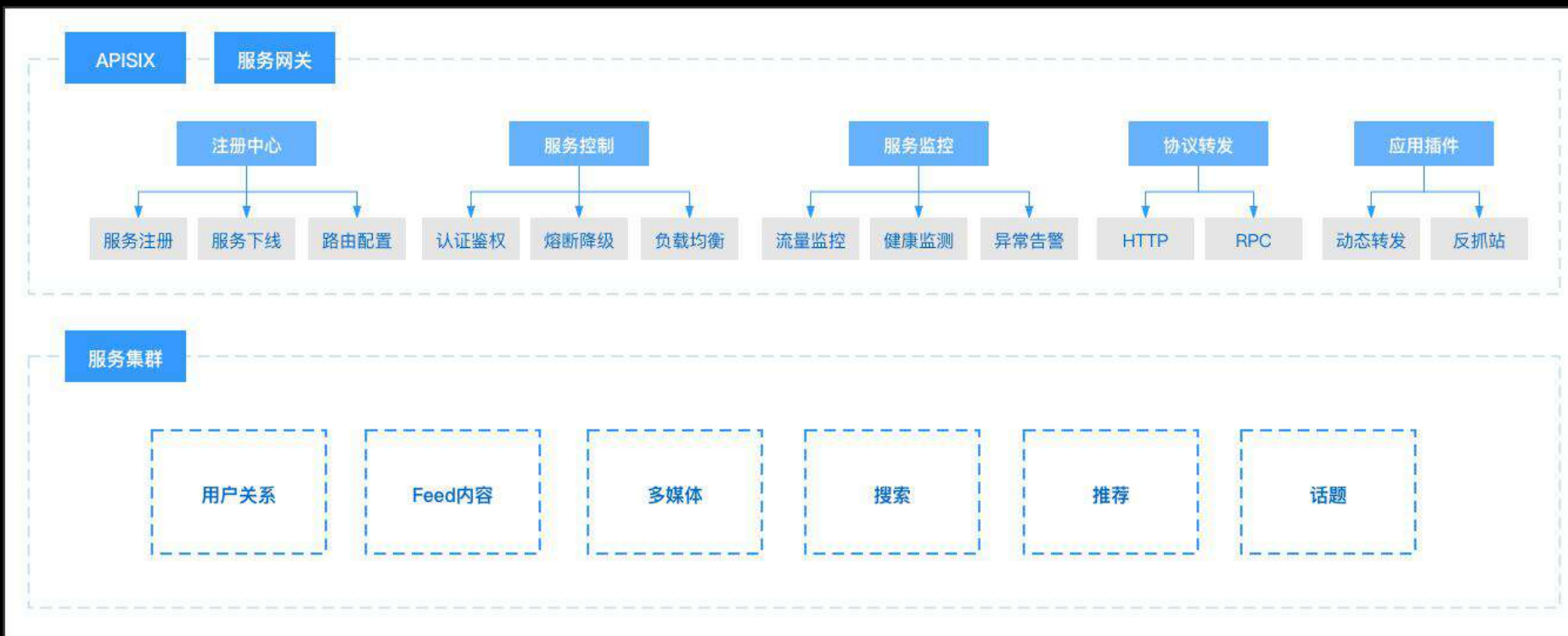


APACHE
APISIX

引入APISIX网关的架构整合

统一网关设计思路

- 利用APISIX构建的服务网关，所有的服务都注册到该网关下



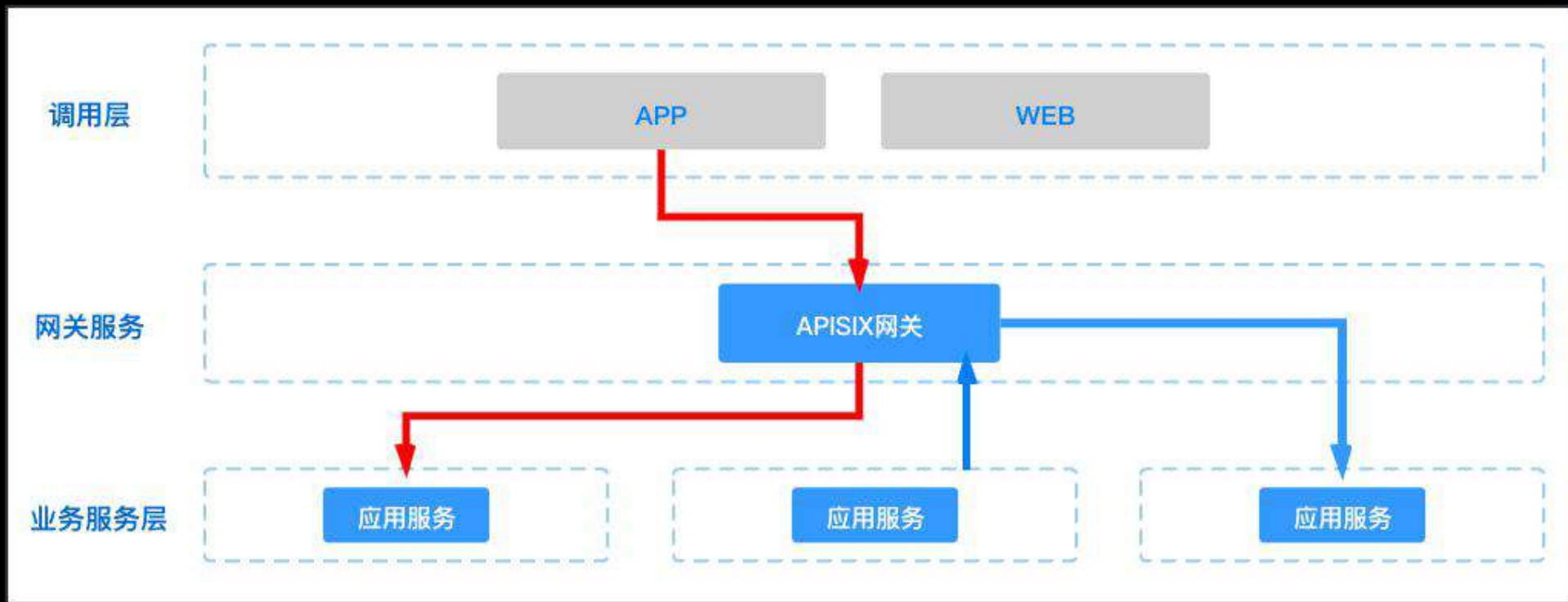


APACHE
APISIX

引入APISIX网关的架构整合

统一网关后调用链路

- 南北流量统一指向网关进行路由转发
- 东西流量同样需要过网关



APACHE APISIX CONNECTS THE WORLD



APACHE
APISIX



引入APISIX网关的架构整合

整合后优势一

- 统一解决掉南北和东西流量问题，节省资源和人力成本，做到统一监控，统一配置
- 业务服务的部署架构重点在服务本身，让网关独立存在，业务无感

整合后优势二

- 通过扩展插件，各服务的权限验证、路由分发、健康检查等均由网关托管
- 新业务上线、服务迁移可以动态完成，运维友好

整合后风险点

- 网关承载所有的流量
- 分久必合合久必分，统一网关对于不断扩张的服务数量风险也会不断增加



APACHE
APISIX



03

应用实践

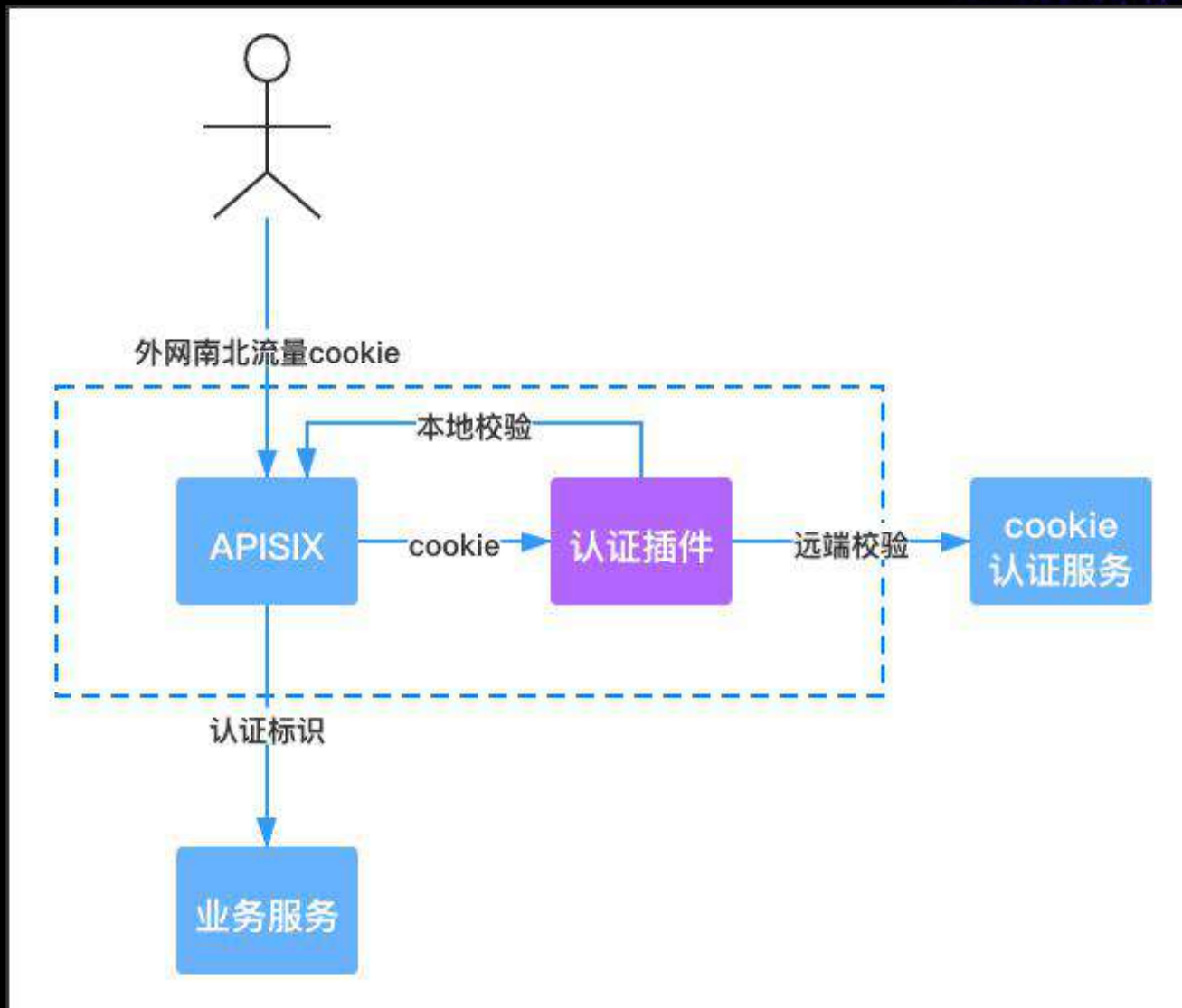
集群安全性-认证插件

集群伸缩性-无状态服务扩容迁移

集群功能扩展性-业务动态转发

南北流量认证-cookie

- 确保用户cookie信息安全，cookie只停留在网关层，避免各服务的不规范使用cookie
- 省去各服务南北流量cookie认证的复杂性
- 因为人的问题、单体服务问题导致cookie泄露，cookie认证的加解密规则泄露，这将加大cookie的管理难度



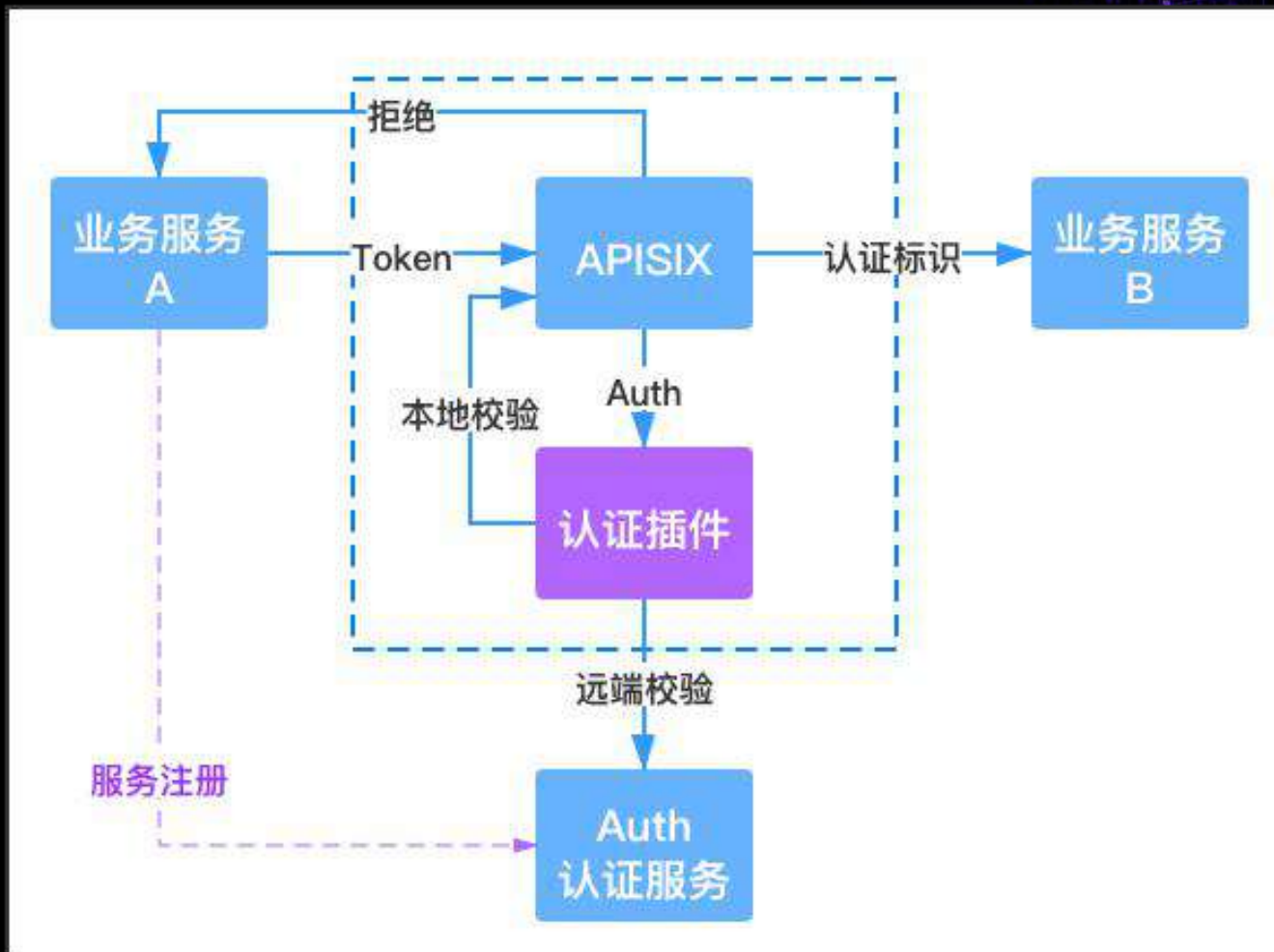


APACHE
APISIX

东西流量认证-Token

- 服务需要进行认证注册
- 服务之间互相调用全站统一token认证规则便于调用溯源，问题追查，权限控制等
- 被调服务不需要关心太多认证细节，认证策略升级也完全透明

避免认证策略五花八门、认证接口遗漏、找不到调用来源、无法阻止调用等问题
统一认证可以较好的保证集群的安全性





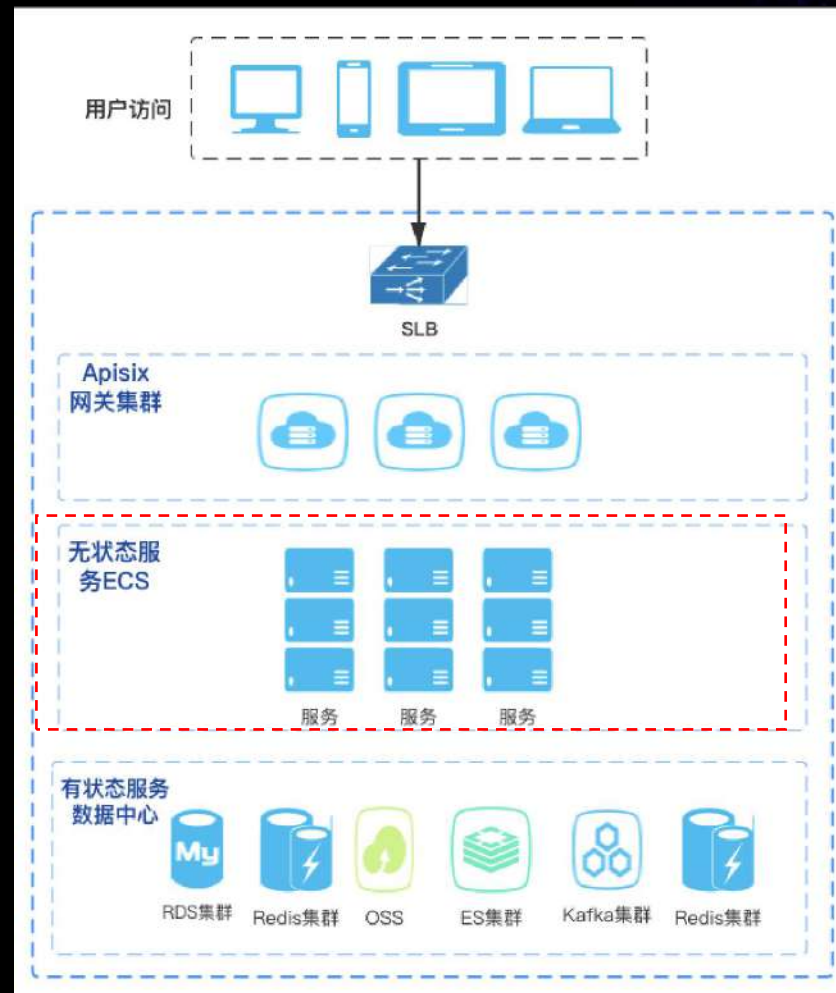
APACHE
APISIX

集群伸缩性-无状态服务扩容迁移

Beeto整体部署架构

- 服务划分：网关、无状态服务、有状态服务
- 流量峰值扩容和迁移重点：无状态服务

无状态服务承载了主要的请求压力
网关确保无状态服务具有良好的伸缩性





APACHE
APISIX

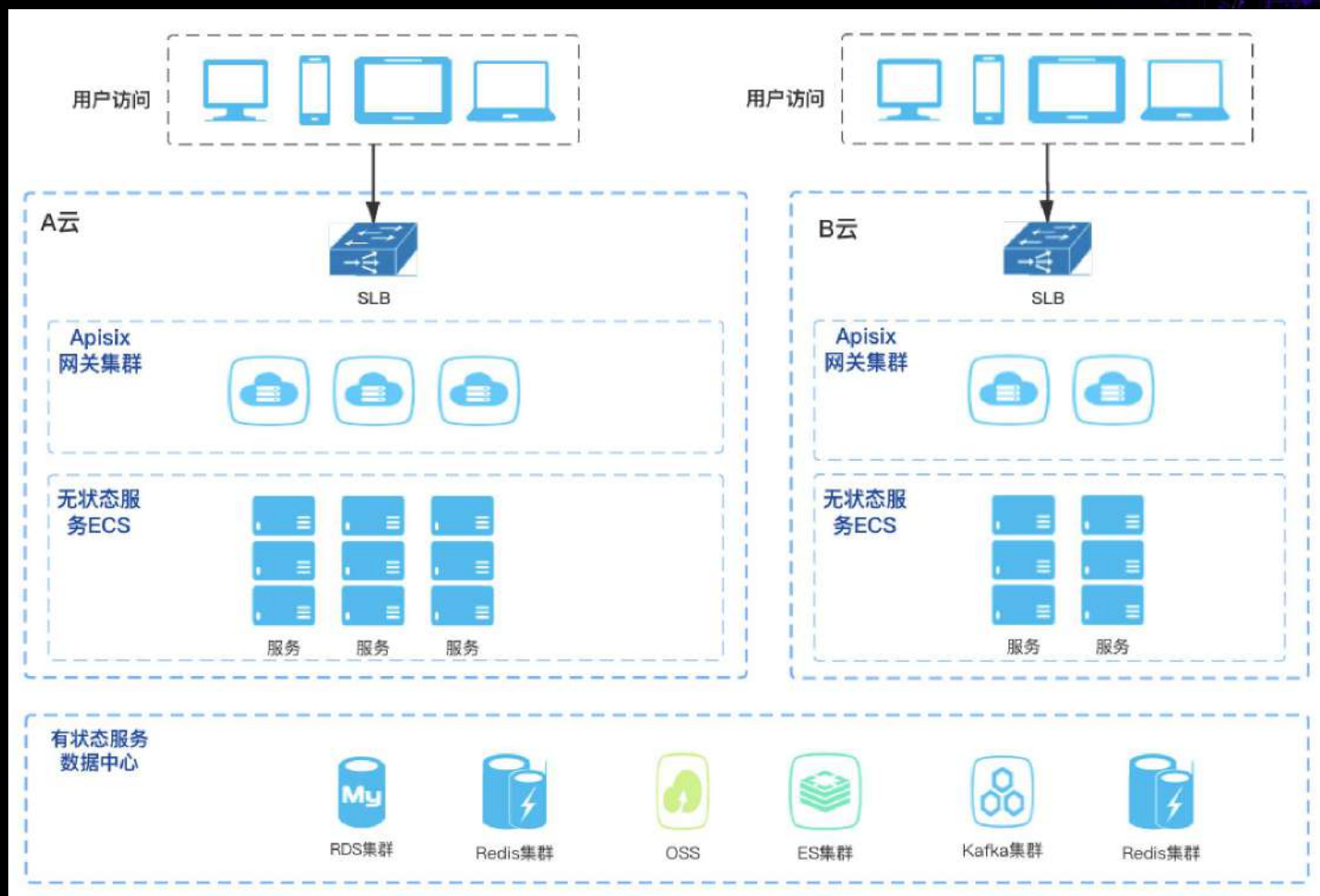
集群伸缩性-无状态服务扩容迁移

网关下的无状态服务扩容迁移

- 单体服务的伸缩性
- 集群的整体伸缩性

在网关层可以实现对各个服务动态扩缩容

整体集群进行云服务转移时，通过网关的迁移可以友好的完成整个云服务的迁移





APACHE
APISIX

客户端Cardlist

- 这是两个“不同”的页面
- 其实它是“同一个”页面
- 页面由不同的模块（Card）组成
- 这个页面叫Cardlist
- Cardlist内容由接口下发的模块数据决定

联合客户端的一种渲染规范，目的是让客户端的渲染view更通用，业务处理更灵活





APACHE APISIX

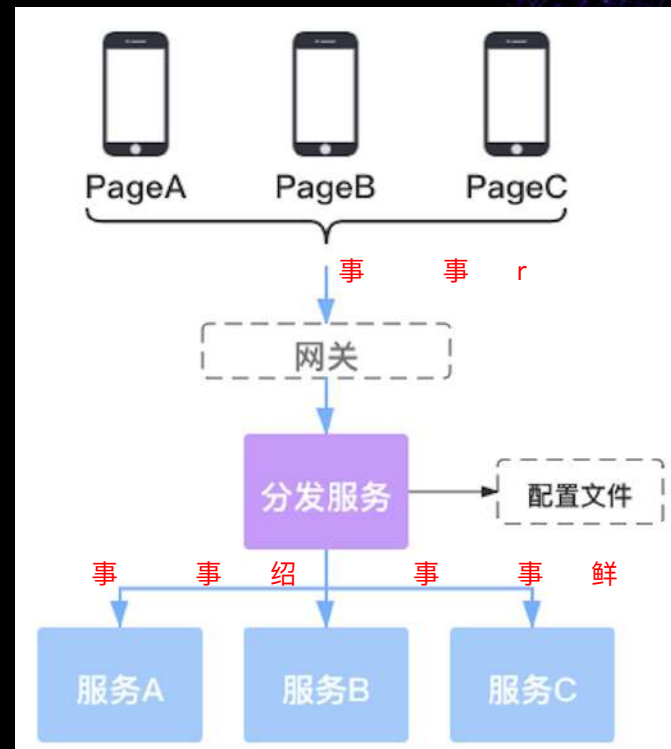
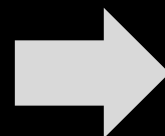
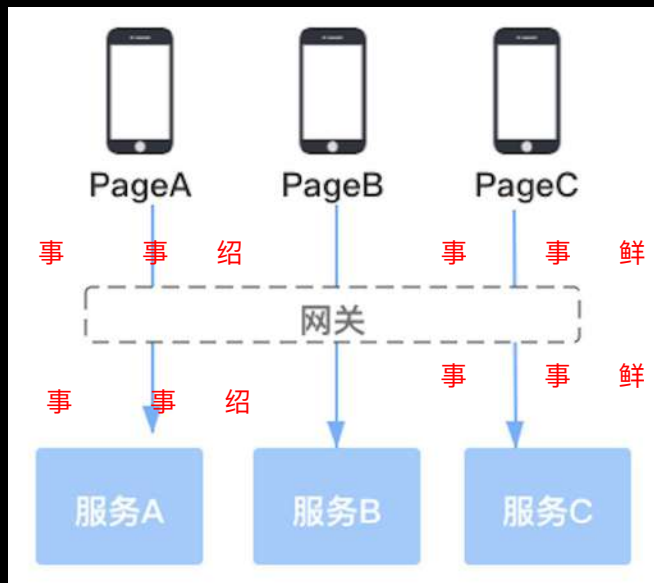
集群功能扩展性-业务动态转发

业务分发服务

- 客户端提供同一类型的展示模板
- 不同页面调用各自不同的服务
- 做一个统一的分发服务

客户端需要维护各个页面的调用
分发服务会增加部署资源和调用链路

目标服务的流量同样会再经过网关



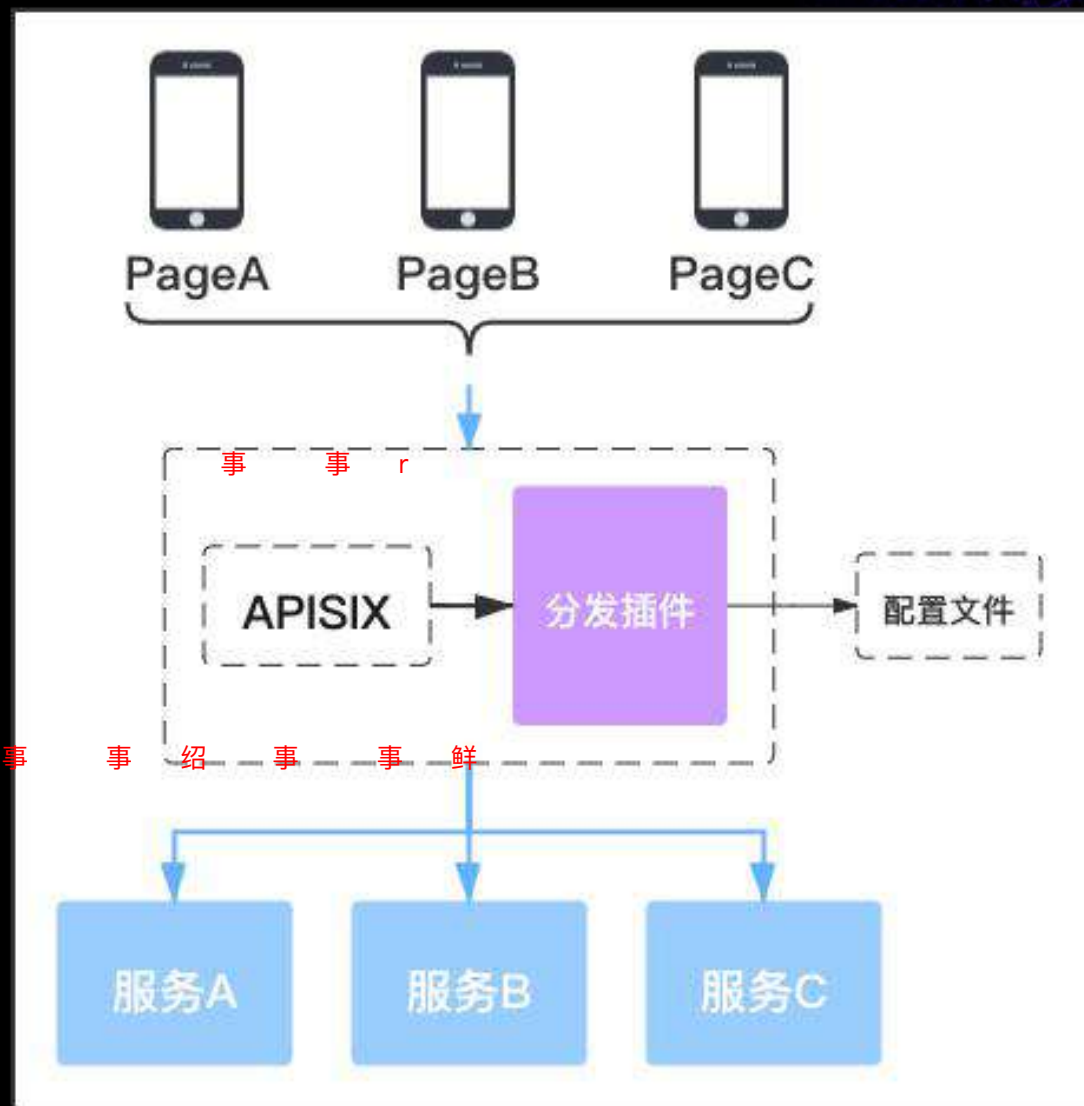


APACHE
APISIX

网关插件做业务分发

- 客户端统一请求一个接口
- 网关识别后引用插件进行请求参数和URL配置规则处理
- 按照配置规则解析后将请求从网关层转发到指定服务

客户端只需要定规范无需关心数据来源
网关承担了分发服务的职责，直接将流量转发到对应服务上
配置文件动态更新，实现根据URL、参数、Header动态转发





APACHE
APISIX

总结

Beeto的网关应用总结

- 引入网关的设计思路
- 网关在整体架构和业务上的具体应用





APACHE
APISIX



感谢聆听
THANKS

APACHE APISIX CONNECTS THE WORLD