

The logo for SACC (System Architect Conference China) features the letters 'SACC' in a bold, white, sans-serif font. The letters are outlined with a glowing blue light effect. A blue swoosh underline is positioned beneath the letters. The background of the entire image is a dark blue, futuristic digital landscape with glowing blue lines and geometric shapes, including a large '2021' in the bottom left corner.

2021 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2021

数字转型 架构重塑

IT168.com

ChinaUnix.net

ITPUB

云上会议 网络直播 | 2021.5.20-2021.5.22

技术中台在中通的实践

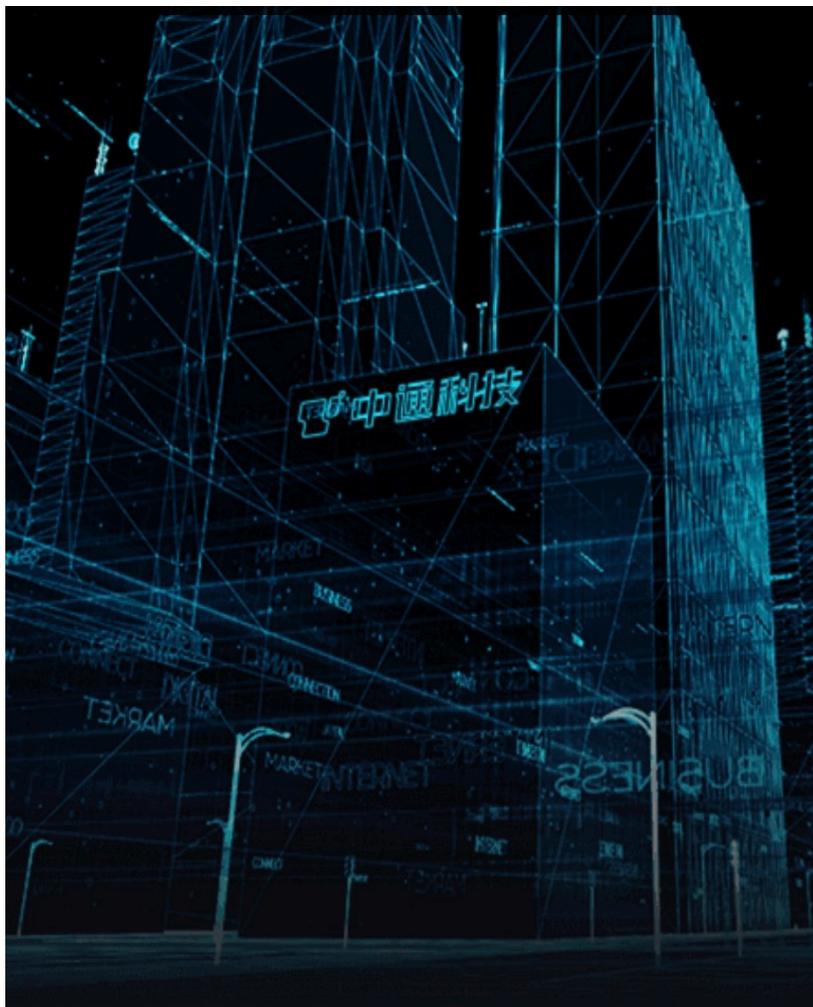
中通科技-基础平台部-肖文科

2021年05月



高效协作 勇于担当
积极创新 用户至上

自我介绍



肖文科

技术中台负责人

2013年加入中通，近几年主要负责技术中台的建设工作，包含了公司内部的统一框架、统一接入层、中间件平台、应用性能监控等。



ZMS (中通消息平台)

https://gitee.com/zto_express/zms



ZCAT(中通应用性能监控系统)

https://gitee.com/zto_express/zcat



CONTENT

目录>>

01 /

背景介绍

02 /

实践分享

- 1、整体介绍
- 2、实践过程

03 /

存在问题

- 1、当前问题
- 2、原因分析

04 /

未来规划

- 1、概览
- 2、技术选型



高效协作 勇于担当
积极创新 用户至上

01

背景介绍

介绍中通科技技术中台建设的背景以及当前
实践的具体情况



高效协作 勇于担当
积极创新 用户至上

2013

● 自研业务系统 (.NET + Java)

开始全面由外包业务系统转变为自研业务系统
(.NET为主, Java为辅)

2017

● 新一代业务系统 (Java)

2017年全面替代上一代自研业务系统拥抱微服务 (Java)

当下

● 研发团队规模

技术团队人员规模超过1000人, 覆盖快递业务全场景, 同时提供生态圈 (快运、国际、云仓、优选、金融) 业务全方位的研发支持

● 双十一大考

经历了多年的双11大考, 在日处理订单量峰值已超过2亿+的业务场景下, 技术中台已覆盖上游数百套业务产品线2000+应用高效、稳定的运行

02

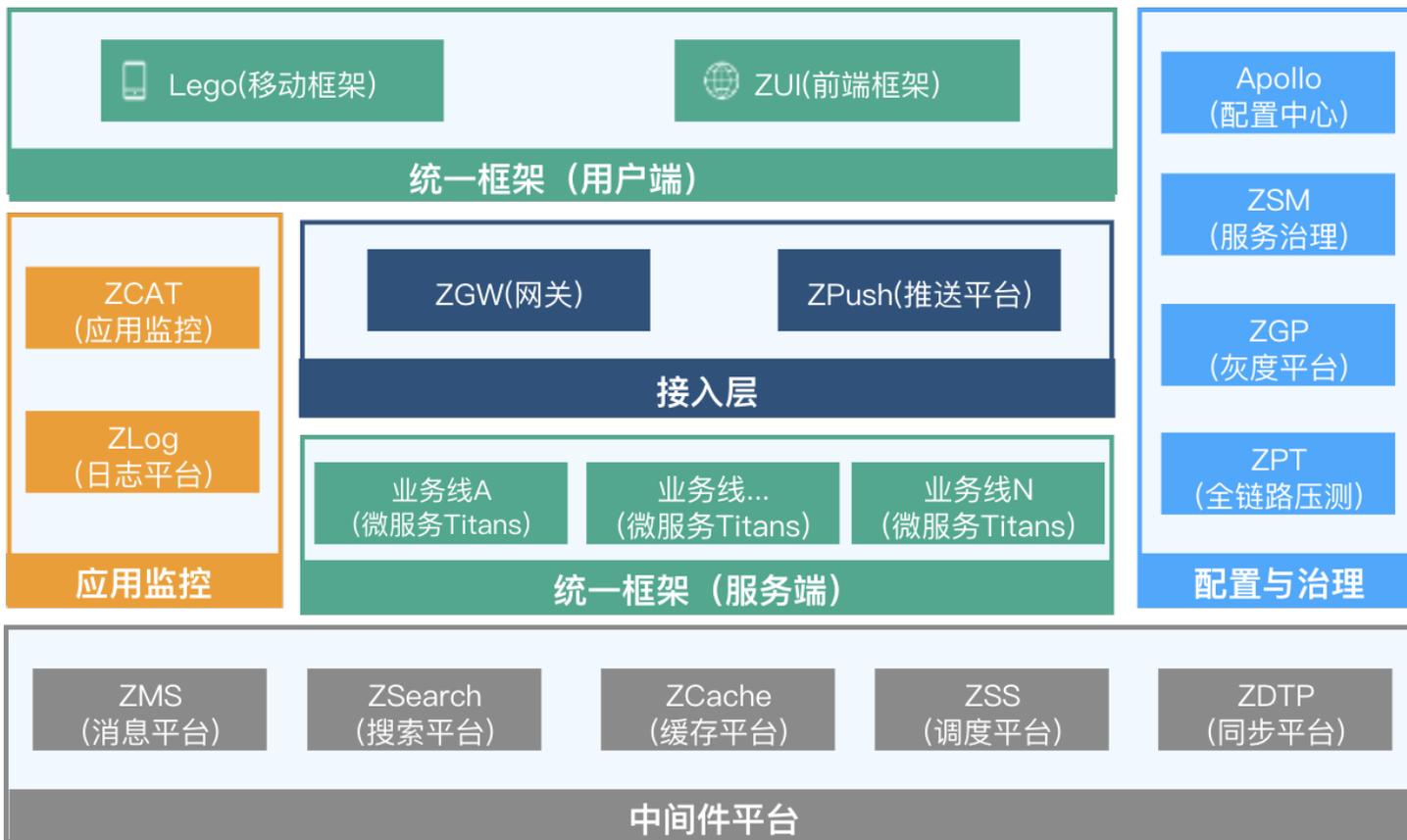
实践分享

从整体上介绍技术中台概貌以及在中通的实践过程



高效协作 勇于担当
积极创新 用户至上

实践分享-整体介绍



技术中台能力介绍

技术中台能力在从统一框架、统一接入层、应用监控诊断、配置与治理、中间件平台五个部分组成，涵盖了业务应用从创建到上线的整个过程

- **统一框架**
 主要分为服务端框架Titans、前端框架ZUI、移动框架LEGO。
- **统一接入层**
 接入层，避免重复性劳动，解耦业务和非业务的功能，让业务更加聚焦，提升整体的系统健壮性和可用性。
- **应用性能监控**
 应用上线后，监控、链路跟踪是帮助研发识别问题、快速止血的有效手段，能够帮助应用在线上健康运行
- **配置与治理**
 配置与治理服务于应用生命周期内提供配置的热更新，服务熔断、降级、蓝绿、灰度，全链路的压测等能力
- **中间件平台**
 基于主流中间件在消息、缓存、调度、搜索等领域不断探索技术深度，在对应领域提供最佳的解决实践方案，为研发人员赋能，降本增效

实践分享-统一框架

资源浪费

相同的技术栈重复性的经历预研、使用、踩坑等多个阶段，造成人员、设备的浪费

01

02

跨团队协作

跨团队之间的协作，如API调用、消息投递、权限、加密等标准问题，无法达成一致，造成团队之间的协作问题

为什么要统一框架

项目风险

项目交接、人员变更对项目的影响太大

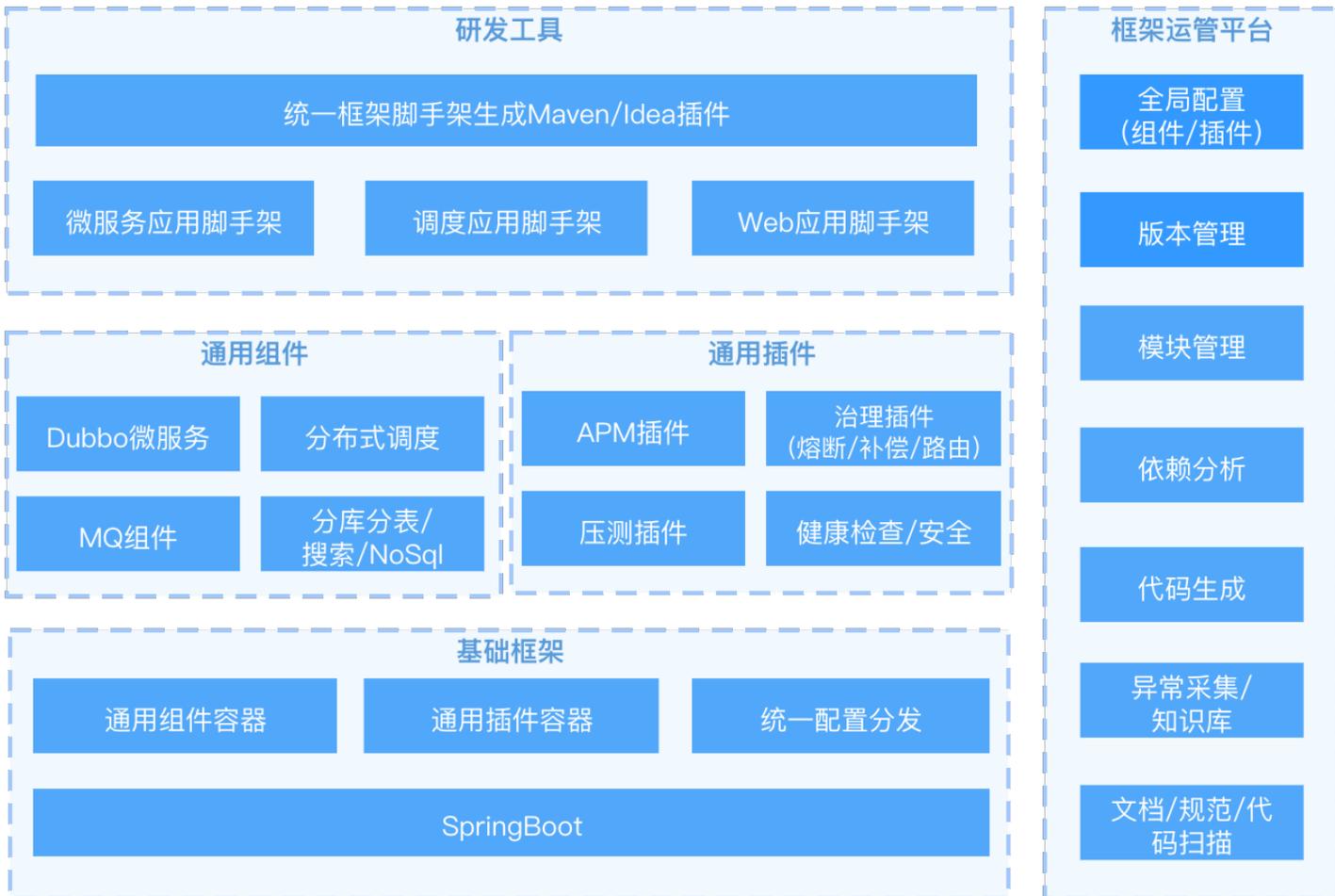
04

03

产品体系化

- 1、对外基于框架输出的产品统一
- 2、对内研发过程统一

实践分享-统一框架 (Titans后端框架)



基础框架

基于SpringBoot之上实现通用的容器来管理组件/插件的生命周期（注册、启用、停用、更新），使组件/插件具备自由的可插拔的能力，同时借助统一配置分发可做到组件/插件配置热更新

通用组件/插件

组件：框架主动提供的核心能力（微服务、调度、MQ、分库分表）

插件：框架针对部分能力做了扩展（APM、治理、全链路压测、健康检查）

研发工具

- 1、脚手架
- 2、代码生成
- 3、规范、扫描、多环境全局配置

实践分享-统一框架（Lego移动框架）



基础框架

基于Android/IOS分别实现了原生、RN、H5组件的运行容器及统一路由，便于上层组件的业务功能实现

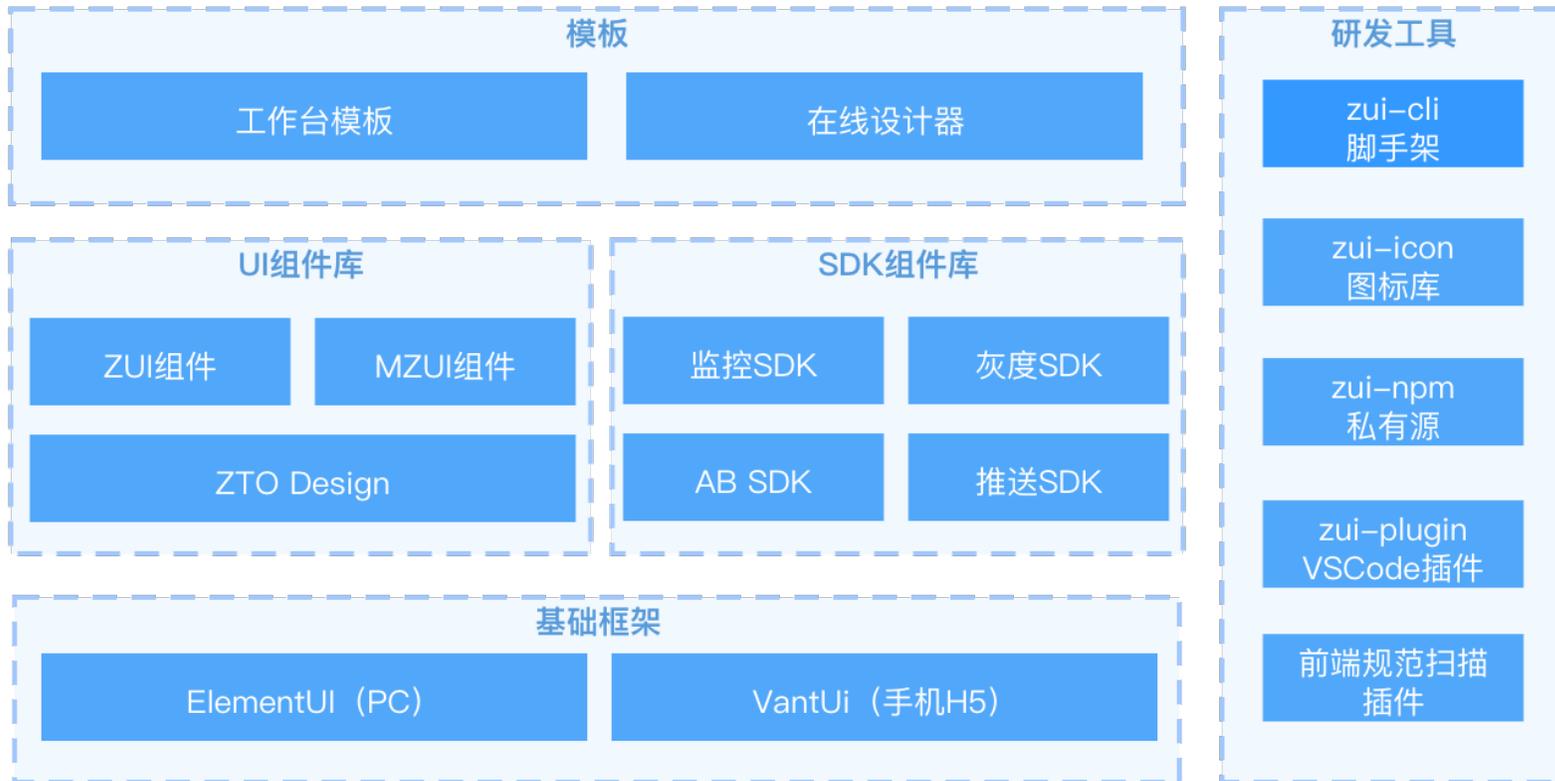
业务/技术组件

业务组件：通过路由唤起一个完整的业务功能的组件（扫描、网络电话、打印）
技术组件：框架本身的增强特性（推送、配置、升级、热修复）

研发工具

- 1、组件的创建脚手架
- 2、发布工具
- 3、日志、监控等

实践分享-统一框架 (ZUI前端框架)



UI/SDK库

UI 组件库：基于 ZTODesign 定制了 ZUI/MZUI的组件样式

SDK组件库：通用的框架功能增强库（监控、灰度、推送等）

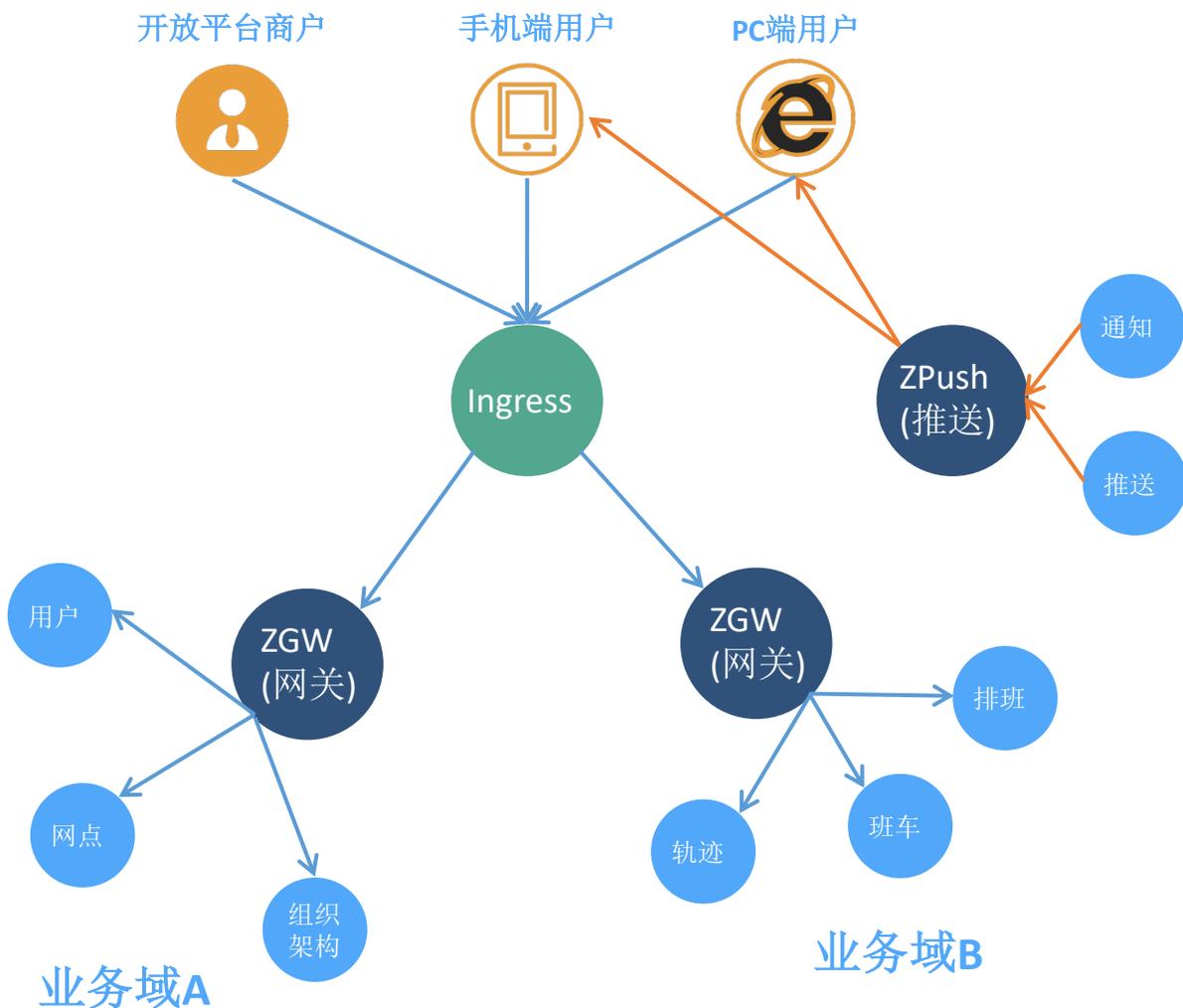
模板

- 1、工作台模板：基于ZUI的完整后台模板，可以随意切换多种模式，SSO版/网关版
- 2、在线设计器：基于界面拖拉拽生成json配置即可生成UI界面

研发工具

- 1、脚手架
- 2、图标库
- 3、API代码生成

实践分享-统一接入层（应用场景介绍）



提升效率

- 1、解耦业务与非业务之间差异（协议、前后端对接流程、多端推送等）
- 2、统一提供鉴权，限流，熔断等
- 3、减少定制化开发（聚合、映射）



提升稳定性

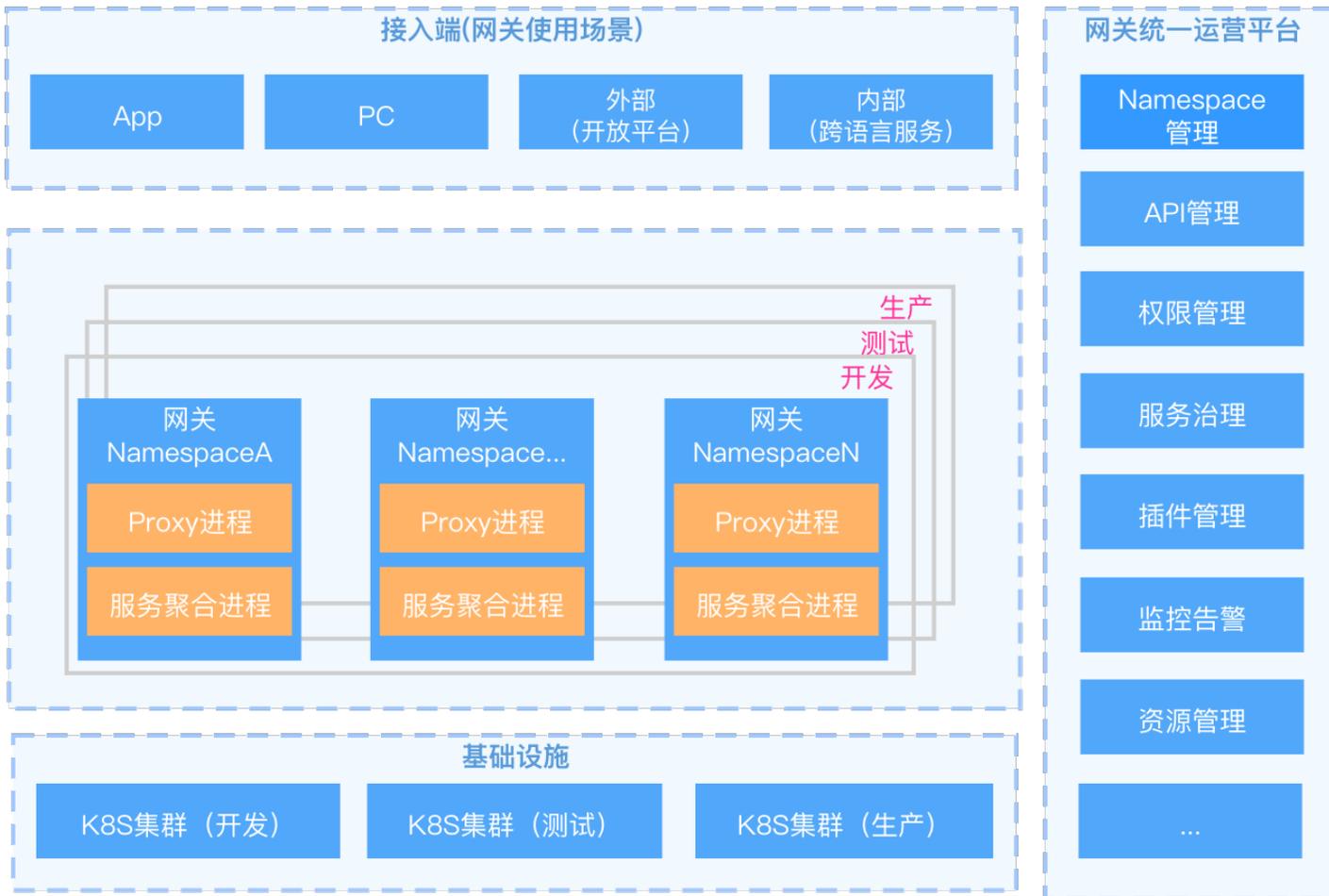
针对新版本服务上线可自定义流量放入比例，也支持自定义groovy脚本定制化指定标签特性的用户流量流入新版本，从而缩小不稳定的风险



提升安全性

降低业务线产品直接向外网开放的风险，与安全团队协作针对网关安全漏洞针对性测试，及提供定制化的黑白名单、防刷等特性

实践分享-统一接入层 (ZGW网关)



运管平台

统一管理网关的整个生命周期，包含以下步骤：环境->namespace->api->鉴权->治理等
同时也承担了运维工作如网关的在创建、伸缩、监控等

Proxy进程

- 1、全异步通讯提高性能
 - 2、两大插件体系
- 执行过滤器实现参数映射、鉴权、灰度、限流等特性
协议适配器实现了多种后端服务协议 (dubbo/http/mq)

服务聚合进程

支持两种聚合语法，分别是内嵌了Groovy和GraphQL两大引擎，借助运营平台API管理模块的在线编写聚合服务，可以较快的定制一个服务聚合后的新API

实践分享-统一接入层 (ZGW网关)

ZGW - 网关 网关管理 审核 使用文档 容器 网关监控 当前环境: PRO 肖文科

← 返回 | subscribe-mobilegateway-q865

操作

新增命名空间 修改启动参数 更新网关版本 扩容POD 销毁

集群名称
mobilegateway_new

集群ip
10.4.82.20:23456, 10.4.82.21:23456, 10.4.82.22:23456, 10.4.82.23:23456, 10.4.82.24:23456, 10.4.82.25:23456, 10.4.82.26:23456, 10.4.82.27:23456, 10.4.82.28:23456, 10.4.82.29:23456, 10.4.82.30:23456

访问端口
34160

启动参数
-Dspring.profiles.active=prd -Xmx4096m -Xms4096m -Xmn2048m -Xss256k -XX:+DisableExplicitGC -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSCompactAtFullCollection -XX:LargePageSizeInBytes=128m -XX:+UseFastAccessorMethods -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancy...

网关版本
4.3.0

pod

name	ip	状态	启动时间	操作
subscribe-mobilegateway-q865-6f59c87f4c-8z9n5	10.4.82.27	Running	2021-04-27 09:56:20	控制台 删除
subscribe-mobilegateway-q865-6f59c87f4c-14l97	10.4.82.213	Running	2021-04-27 09:56:38	控制台 删除

关联的namespace

namespace	创建人	创建时间
-----------	-----	------

实践分享-统一接入层 (ZGW网关)

ZGW - 网关

网关管理 审核 使用文档 容器 网关监控
当前环境: PRO 肖文科

命名空间: ...
API配置
API主表
审核
变更历史
服务聚合

API名称:
描述:
标签:
状态: 正常

dubbo接口:
接口方法:
服务名称:

更新人:
更新时间: 至
搜索 重置

全部草表API (556)
+
同步
日志查询

注意事项

- 添加、修改API等操作, 均需要 点击发布按钮 将其发布至主表才能生效
- 点击下线按钮后, 仍然需要 点击发布按钮 才能生效
- 不建议直接在生产环境修改数据, 请使用同步功能将数据同步至生产环境, 同步后请切换至生产环境再发布一次
- 接口报错时, 你可以点击API右边操作列的 搜索图标, 进入 zlog 查看日志和异常堆栈

<input type="checkbox"/>	Api	版本	描述	命名空间	标签	更新时间	更新人	操作
<input type="checkbox"/>	未发布 R 改	1	业务员信息查询	mobile		2020-07-08 10:16:10	刘星	🔍 🗑️
<input type="checkbox"/>	已发布	1	短信记录, 我猜想的	mobile		2021-04-25 13:22:02	刘星	🔍 🗑️
<input type="checkbox"/>	已发布	1	丰巢订单查询是否开箱	mobile		2021-04-25 13:21:58	刘星	🔍 🗑️
<input type="checkbox"/>	已发布	1	解析excel文件为数组	mobile		2021-04-22 20:15:27	胡传福	🔍 🗑️

设置

API管理

灰度脚本

应用管理

服务管理

实体管理

Filter配置

配置数据

数据统计

告警

熔断降级规则

实践分享-统一接入层 (ZPush推送平台)



客户端/渠道

- 1、全面覆盖多种类型终端统一推送API
- 2、对接多个厂家渠道保证送达率

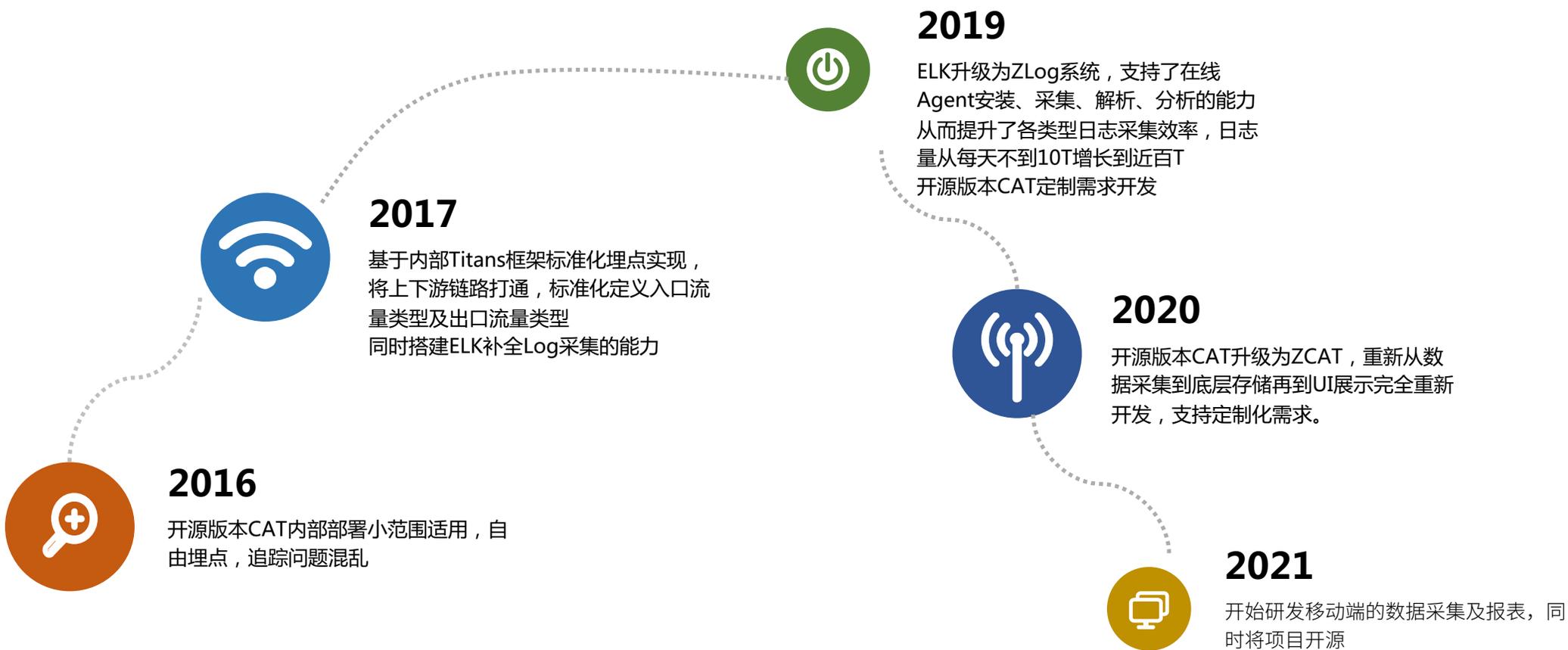
推送平台

- 1、渠道优先级路由
- 2、按标签拆分批量推送
- 3、推送指标分析

运营控制台

管理推送应用的完整声明周期：应用申请->渠道配置->消息/标签/设备管理
同时也支承担了送达率、及时性的监控告警职责

实践分享-应用监控（背景）



实践分享-应用性能监控 (ZCAT)



多客户端SDK

- 1、基于用户无感知的自动插桩SDK
- 2、标准上报协议打通前后端链路
- 3、兼容开源版CAT的协议（新版本采用PB）

数据收集计算服务

- 1、收集计算基于无锁环队列做缓冲，采用多播模式实现Message存储及各类报表。
- 2、Message和报表存储都采用近期内存存储，超过一定时间或者大小后压缩写磁盘，在允许少量数据丢失前提下保证磁盘最大化顺序写入
- 3、数据查询引入SQL引擎，简化报表查询复杂度

监控门户

- 1、以全局逐步放大到局部的视角展现应用全方位信息，降低排障门槛。
- 2、通过路由规则实现统一门户应对多个机房应用监控

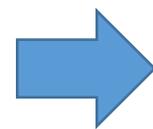
实践分享-应用性能监控 (ZCAT排障之从全局到局部)

ZCAT 请选择应用 ▾

报错大盘 指标列表 看板管理

产品线 应用

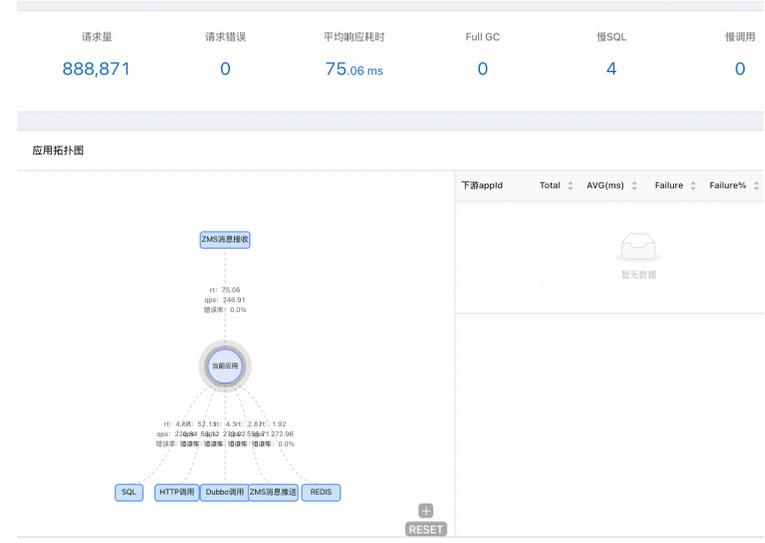
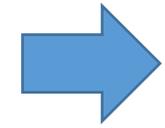
15:45		15:44	
message-endpoint-open	5306	gateway-source-broker	531
gateway-source-broker	489	sharing-stock	229
medivh-callback	156	medivh-callback	180
ZTO_IP_SCANRECORD	141	agentService-deliverInstora...	168
platform-wrapper-service	116	pangu-sdfee-Nausea	156
cat	110	ZTO_IP_SCANRECORD	141
order-ops-to-job	89	platform-wrapper-service	134
open-platform-consumer	87	message-endpoint-open	134
GATEWAY_YX	69	open-platform-consumer	111
xiniao-order-push	56	cat	86



IP 响应时间 | 请求数 | 异常数

全部 75.06ms | 889K | 11

10.9.31.219	91.79ms 202K 7
10.9.31.218	81.02ms 154K 0
192.168.46.241	67.52ms 183K 2
192.168.144.191	64.41ms 167K 1
192.168.144.190	68.81ms 183K 1



总览 入口流量 出口流量 主机监控 异常分析 线程分析 最近1小时

全部 ZMS消息接收 请输入搜索内容

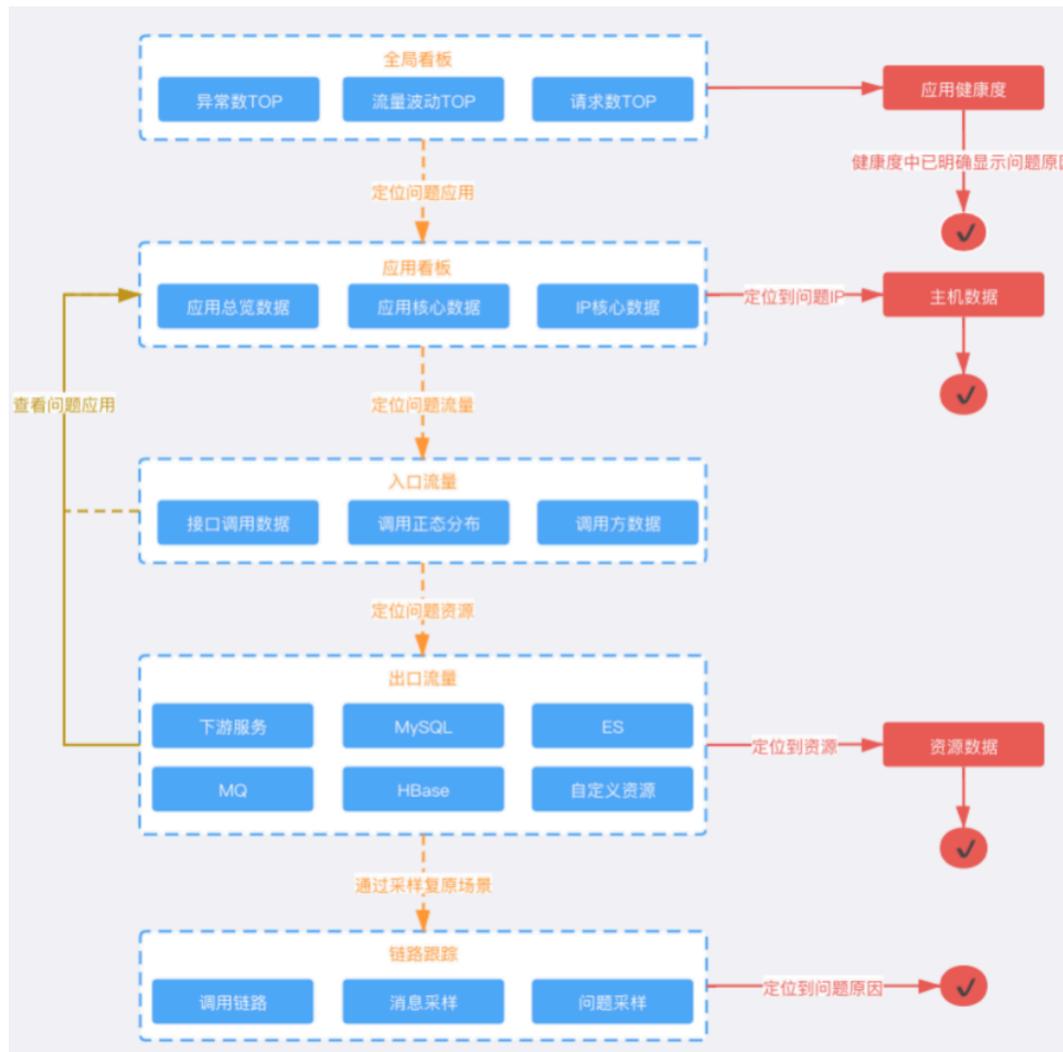
服务名称	Total	AVG(ms)	Failure	Failure%	Min(ms)	Max(ms)	99.9Line	95Line(r)	QPS
ZMS消息接收	888,871	75.06	0	0	1	5,295	854.82	123.81	246.91



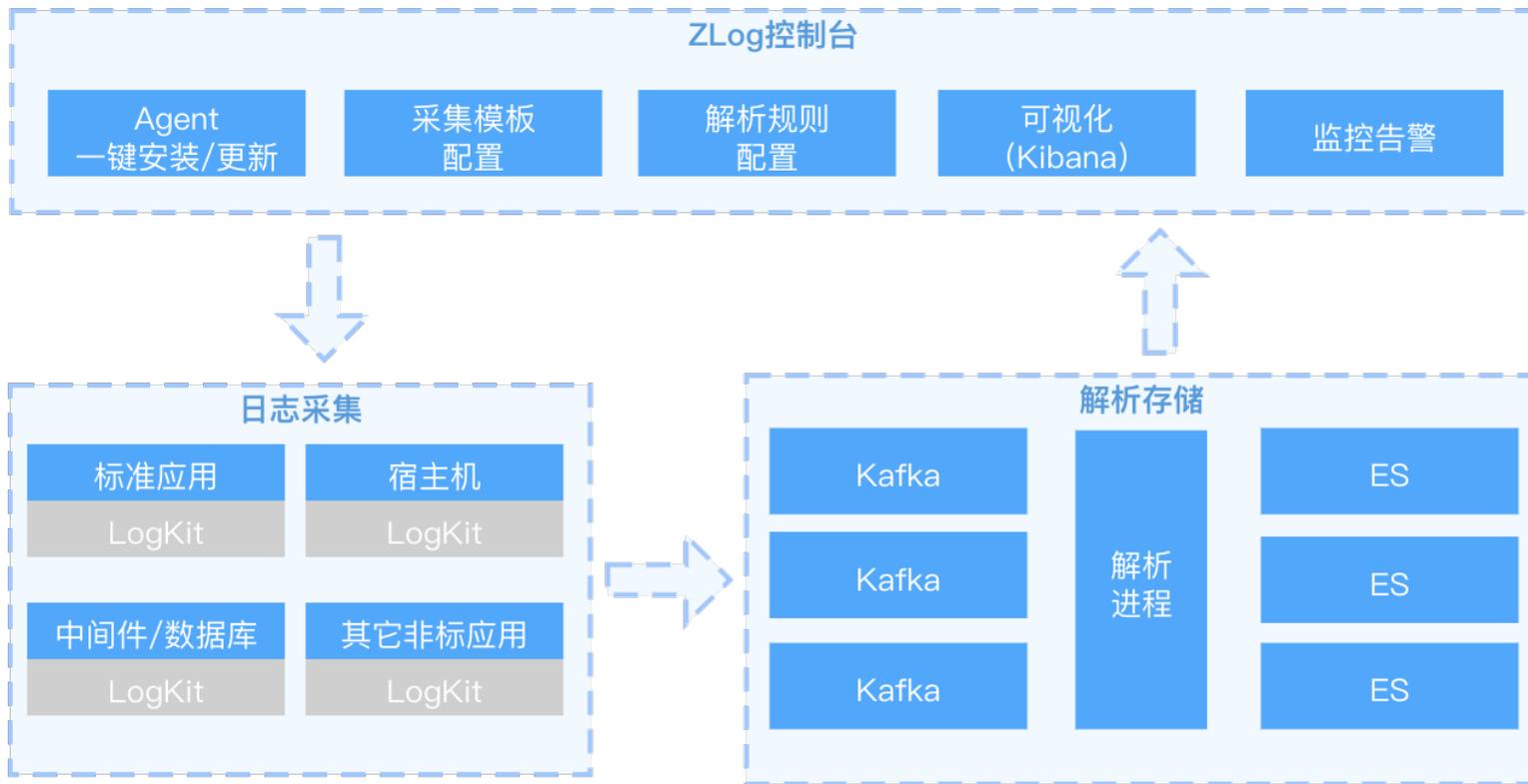
BaseAreaService.findAreaExpandByAreaCode

应用名称	日志产生时间	调用类型	状态	ip地址	服务名称
base-center-...	2021-04-19 07:39:53.031	rpc.dubb...	●	192.168.12.85	base-center-area
base-center-...	2021-04-19 07:39:53.031	rpc.dubb...	●	192.168.12.85	20880
base-center-...	2021-04-19 07:39:53.031	rpc.dubb...	●	192.168.12.85	192.168.12.67
base-center-...	2021-04-19 07:39:53.031	RemoteC...	●	192.168.12.85	

实践分享-应用性能监控 (ZCAT)



实践分享-应用性能监控 (ZLog)



ZLog控制台

- 1、以向导模式将agent->采集->解析->查询的步骤简化,降低日志分析的难度
- 2、内嵌Kibana提供日志检索及报表定制

日志采集Agent

基于七牛Logkit二次开发,支持自定义日志分隔符及统一控制台下发配置

解析存储

input->kafka多线程poll+本地缓存的机制提升消费速度

filter->标准日志禁用正则匹配解析来降低CPU占用

output->ES按分片元信息计算目标节点写入,提升ES写入吞吐量

实践分享-配置与治理（统一框架的扩展能力集合）



配置中心

基于开源产品Apollo二次开发，与内部框架深度集成做到配置的动态、各类资源链接动态、各类线上的开关、阈值的动态



微服务平台

微服务平台提供了基于微服务的全生命周期管理（创建、上线、文档、Mock、熔断、降级、蓝绿、灰度等）的能力。



移动灰度平台

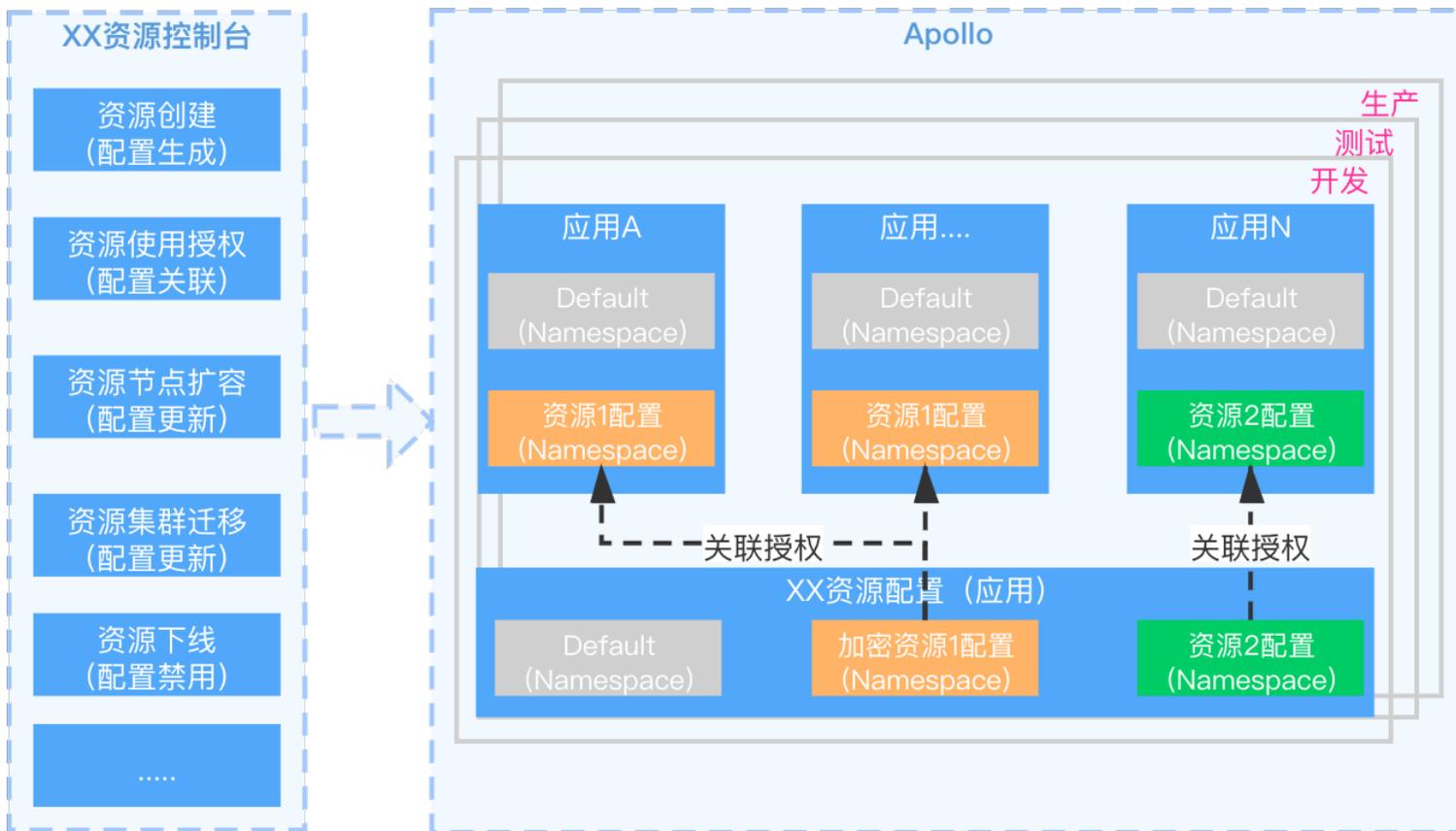
移动灰度平台配置移动框架Lego及H5框架提供了App应用/模块/H5的管理、热更新、灰度发布、配置下发等能力



全链路压测平台

借助Titans框架的插件机制提供直接在生产环境对应用进行性能压测的能力，包含可视化的压测任务管理平台、压测数据清洗、数据偏移、数据隔离、压测指标收集与熔断等功能。

实践分享-配置与治理 (Apollo配置中心之资源授权)



统一管理资源配置

做到资源的配置统一下发、变更，防止资源配置到处复制带来的资源配置管理混乱

精准化资源授权

资源授权关联到AppId级别，精准控制某个资源能被哪些App访问

安全管控

针对资源配置全部加密返回（appId级别），由框架运行时根据AppId的secret进行解密，避免资源配置泄露风险

实践分享-配置与治理 (ZSM微服务平台)



服务全生命周期管理

从服务的定义->服务的开发->服务的测试
->服务的上线->服务的治理的全生命周期
管理，用户都可以在服务平台完成相关
的操作

提升各环节对接效率

推行服务开发的最佳实践，标准化服务
研发流程，统一定义后端服务元数据，
自动对接前端开发及测试，降低人员沟
通成本，提升各环节的开发效率

沉淀服务集市

统一沉淀公司所有服务到服务平台，做
到服务定义明确，随用随取。

实践分享-配置与治理 (ZSM服务平台)

```

@Override
@ZsmApiOperation(name = "根据条件获取来电记录", description = "根据条件获取来电记录",
    @com.zto.zsmp.annotation.gateway.GatewayApi(namespace = "workbench95",
        project = "svhwoworkbench",
        isJson = true,
        name = "workbench/search/getTelephoneRecord"
    ))
public <T> <T><TelephoneRecordOutputDTO>> getTelephoneRecord(Telephone
    HwoAsserts.notNull(dto, HwoErrorCode.NOT_NULL, "查询电话记录入参");
    return <T> <T>.getService().getTelephoneRecord(dto);
}

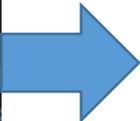
@Data
@ZsmModel(description = "运单查询出参")
public class QueryBillCodeOutputDTO implements Serializable {
    private static final long serialVersionUID = -36462608;

    @ZsmModelProperty(description = "运单号")
    private String billCode;

    @ZsmModelProperty(description = "发件人")
    private String senderMan;

    @ZsmModelProperty(description = "发件地址")
    private String senderAddress;

    @ZsmModelProperty(description = "收件人")
    private String receiverMan;
}
    
```



服务平台
sv-... service ^ 服务列表 服务治理 文档管理 服务授权 服务上下游

当前服务: com.zto.l...delayTypeRateQueryFacade appid: sv-... service

- 概览
- 提供方
- 消费方
- 路由规则
- 动态配置
- 流控规则
- 授权规则

概览

服务概览 [文档预览](#)

方法列表: queryBoundTypeRate(ArbSiteDelayTypeReq) queryContractArea(ArbSiteDelaySiteTopReq)

描述: 进港出港占比查询

入参:

参数名	是否必须	类型	描述
delayTypeReq	否	com.zto.l...ArbSiteDelayTypeReq	进港出港占比查询实体
siteCode	是	java.lang.String	网点编码
lowerFlag	否	java.lang.Boolean	该网点是否包含下级网点
endDate	是	java.util.Date	结束日期
startDate	是	java.util.Date	开始日期

出参:

参数名	是否必须	类型	描述
resp	否	com.zto.hwo.d...ArbSiteDelayTypeRes	
inPortDelayCount	是	java.lang.Integer	进港延误量
outDelayCount	是	com.zto.h...ArbSiteDelayYesterdayDTO	出港延误率

实践分享-配置与治理 (ZSM服务平台)

新增条件路由规则

路由名称: 可使用中文, 由1-200个字组成

优先级: 数字越大越优先

资源名: + 1

消费端或提供端: [消费端匹配条件](#) [提供端过滤规则](#)

host 匹配 请选择

新增流控

资源名: + 1

* 流控应用:

* 单机阈值:

阈值类型: QPS 线程数

流控模式: 直接 关联 链路

流控效果: 快速失败 预热冷启 排队等待

发布网关

namespace:

服务	方法	命名空间	网关api名称	api版本	已存在api
<input type="checkbox"/> com.zto.testWithZsm...	testDubbo11(String)				
<input checked="" type="checkbox"/> com.zto.testWithZsm...	testDubbo12(Friend)	gateway-sample	<input type="text" value="testDubbo12"/>	<input type="text" value="1"/>	

开头不要用/

新增授权规则

资源名: + 1

* 流控应用:

授权类型: 黑名单 白名单

实践分享-配置与治理 (ZGP灰度平台)

ZGP 移动灰度平台

当前环境: PRO - 生产

应用详情

应用名称: 宝盒Windows版
应用类型: 桌面应用
appKey: [redacted]
创建人: [redacted]
创建时间: 2020-06-15 16:28:19
二维码: [QR Code]

下载: [下载页](#) | [历史版本](#)

应用操作: [编辑] [分享] [删除]

版本信息

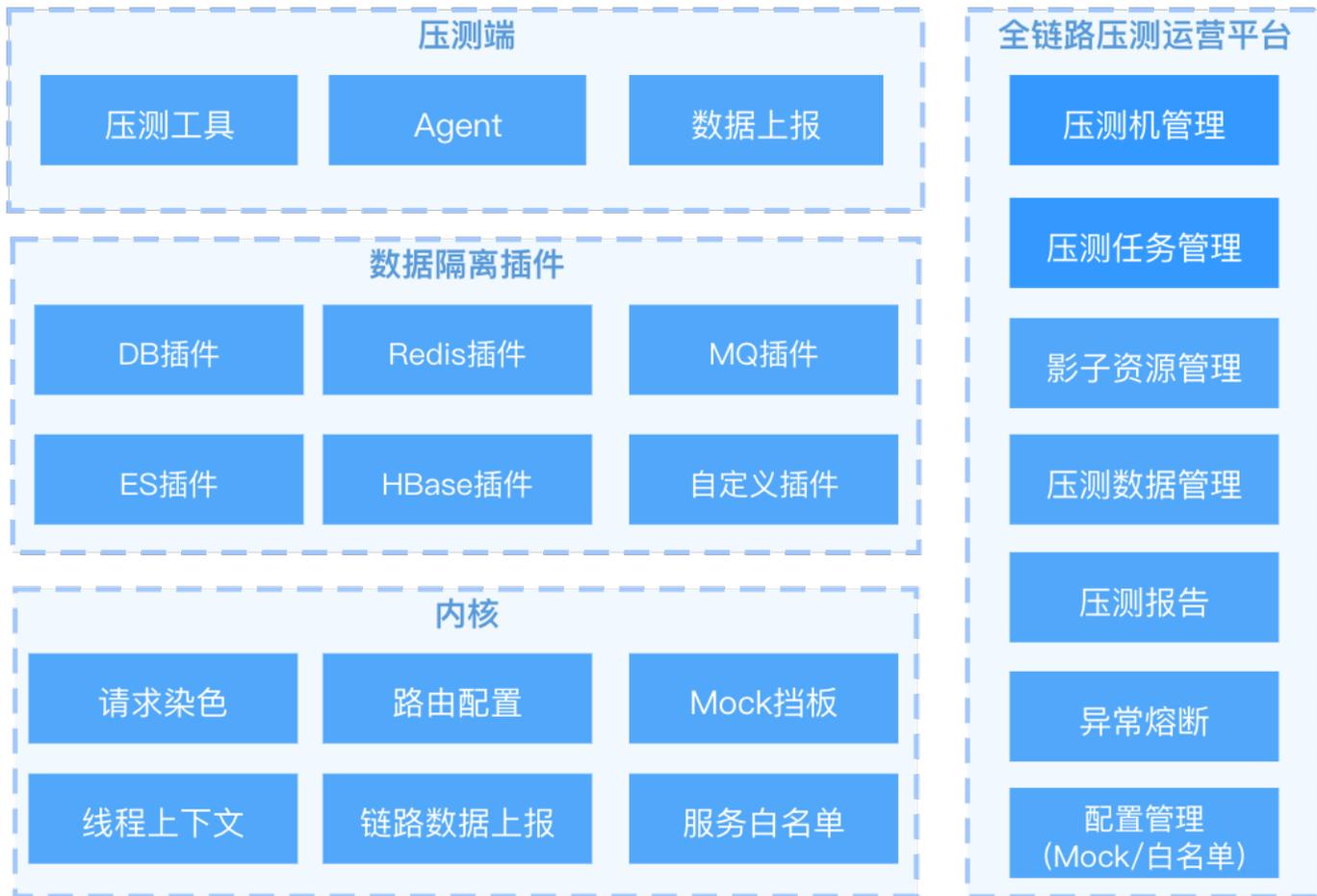
配置信息

新增 | 已删除版本

版本: 请选择 | 强制升级

版本号	状态	发布类型	安装次数	提示用户更新	创建时间	更新时间	版本描述	大小	操作
6.26.0.221	已下线	灰度开放	3	否	17:51:32 2021-03-13	10:34:12 2021-03-15	---bug fixed ---prod	245.63 MB	[编辑] [删除] [下载] [升级]
6.26.0.220	已上线	灰度开放	64	否	17:42:29 2021-03-13	10:22:22 2021-03-16	[新功能] - 桌面音视频二期上线。 [功能优化] - P 宝盒输入框灰度优化; - 桌面顶部UI优化; - 桌面顶部UI交互效果优化; - 桌面顶部UI交互效果优化。	245.64 MB	[编辑] [删除] [下载] [强制升级]
6.25.0.202	已下线	灰度开放	2	否	13:47:21 2021-03-08	17:38:13 2021-03-13	升级测试	243.22 MB	[编辑] [删除] [下载] [升级]
6.25.0.201	已上线	灰度开放	396	否	13:30:37 2021-03-08	10:40:15 2021-03-16	[能力拓展] - 1. 建立总部群; [功能优化] - 1. 桌面UI流程优化; - 1. 桌面UI容错性优化; - 1. 桌面UI容错性优化; - 1. 桌面UI容错性优化; [部署] - 桌面UI容错性优化	243.21 MB	[编辑] [删除] [下载] [强制升级]

实践分享-配置与治理 (ZPT全链路压测)



全链路压测SDK

SDK由内核和基于框架的插件两部分组成：
内核：提供通用的压测过程中对请求染色、链路传递过程中保证标记不丢失的能力
插件：实现了各类数据存储过程中准确的写入影子资源的能力

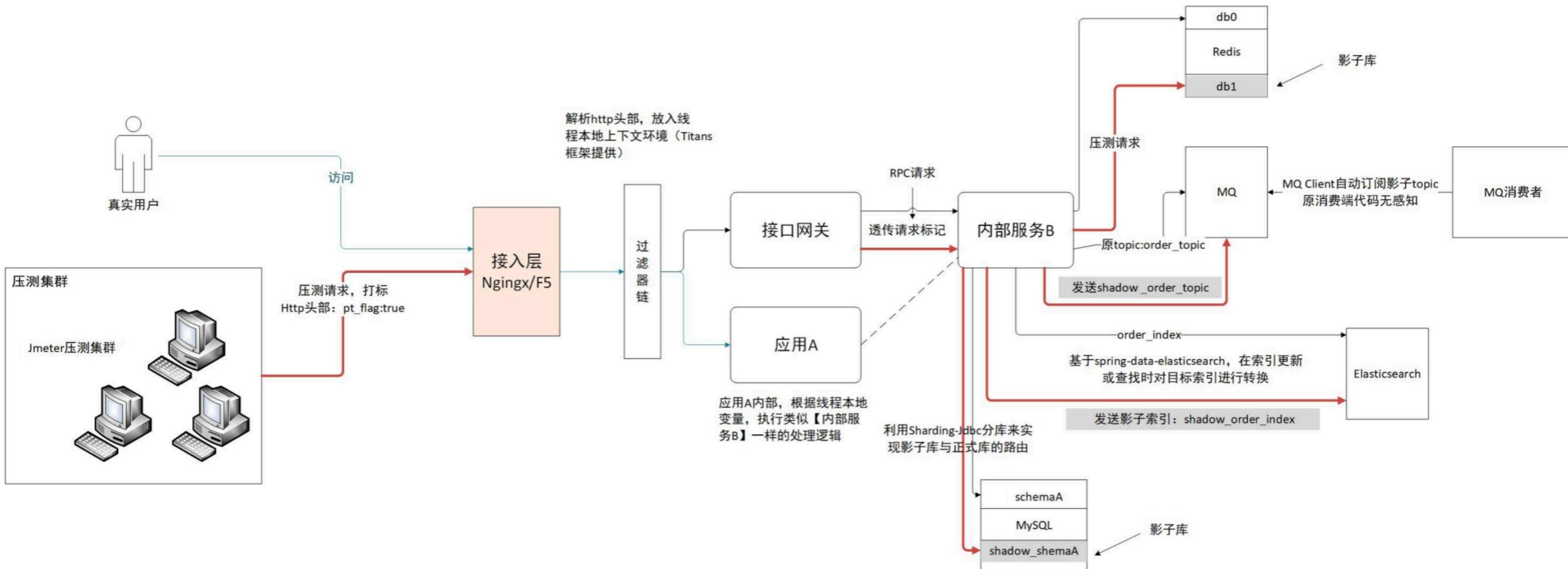
压测端

压测端基于Jmeter做为施压工具，通过本地agent来管理压测工具的启停、压测任务的下发等功能。

压测运营平台

负责压测机、压测任务、影子资源的管理，同时也可以基于平台构建压测数据、出具压测报告、日常的配置等功能

实践分享-配置与治理 (ZPT全链路压测)



实践分享-中间件平台（提供了哪些能力）



ZMS消息平台

ZMS消息平台是屏蔽底层消息中间件类型，通过唯一标识动态路由消息，为开发运维人员提供主题、消费组申请与审批、实时监控、自动告警、容灾迁移等功能。



ZSearch搜索平台

ZSearch是基于ES集群化的运维管理产品，提供统一的搜索Proxy，ES集群生命周期管理（创建，扩缩容，启停，插件，索引，脚本，定时任务），监警告警等。满足了内部多业务线，多版本，多集群的需求



ZCache缓存平台

ZCache统一的缓存平台提供了统一的缓存的Proxy，缓存的一键部署、伸缩、配置变更、流控、监控、智能告警；



ZSS调度平台

ZSS统一任务调度平台是一款弹性调度中间件，提供分布式调度、弹性伸缩、支持国际通用的Cron时间表达式、丰富的生命周期管理机制、DAG有向无环图任务执行 workflow。



ZDTP数据同步平台

数据同步平台，旨在为中通内部提供任意数据源到任意数据源的实时、离线数据同步功能，满足各种异构数据源之间的实时、增量、自定义转换的同步，支撑公司业务的快速发展。

实践分享-中间件平台(ZMS消息平台)



消息运管平台

从环境->集群->主题/消费的申请、安装、升级/下线等管理功能到日常的监控、告警、运维在线迁移等综合一体的运管平台

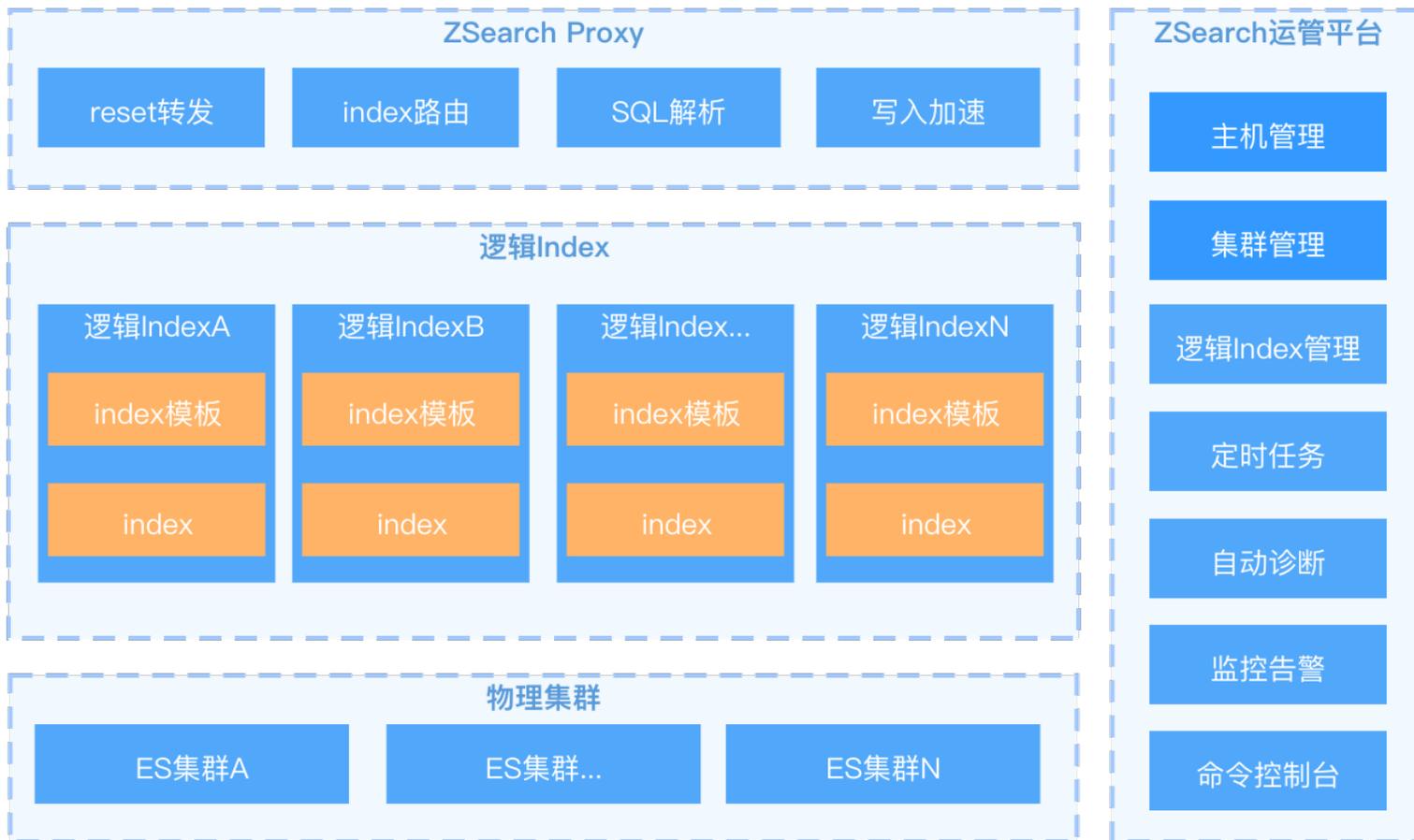
ZMS SDK

- 1、屏蔽差异，统一API
- 2、集群路由热迁移
- 3、增强消息

ZMS各类组件服务

- 1、消息组件 (RocketMQ/Kafka)
- 2、指标数据采集及告警组件
- 3、消息同步组件

实践分享-中间件平台(ZSearch搜索平台)



ZSearch控制台

包括主机池、ES集群、逻辑Index申请、创建、升级等运维操作以及日常使用的监控告警等功能

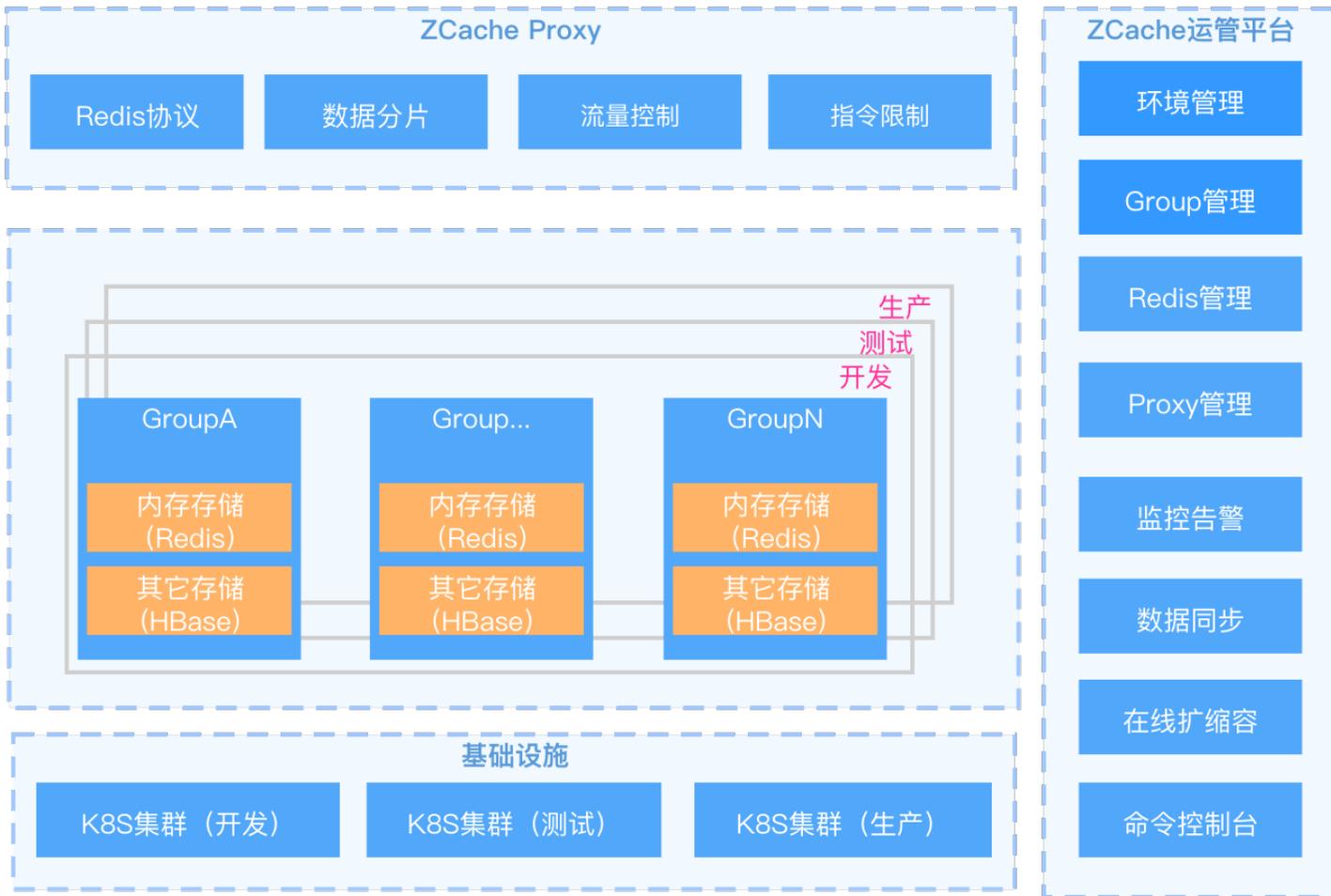
Proxy

Proxy提供了标准的ES rest API转发能力，根据逻辑Index路由到物理集群的能力，以及提供SQL查询的能力，同时也在写入场景做了加速优化。

逻辑Index

逻辑Index是面向用户的最小单元，他是由1个Index模板、0+个Index、外加index拆分策略如按月/日/周组成，逻辑Index可以独占一个ES集群，也可以和其它集群共享。

实践分享-中间件平台(ZCache缓存平台)



Group

Group是面向用户最小单元，一个Group由N组Redis Sentinel+其它存储组成，借助Proxy来做数据分片，其它存储解决大内部占用需要

Proxy

Proxy是一个兼容了Redis协议的代理进程，主要屏蔽Group内部多节点、多类型缓存的复杂性，同时借助Proxy层也可以做到资源层面的控制，如流量控制、客户端使用的指令控制

ZCache运管平台

运管平台包含了日常环境的创建/删除，Group的申请、审核，Group里面Redis节点的创建、更新，Proxy的路由及扩容等，也支持多个Group之间的数据同步，监控数据的采集告警等日常运维操作。

实践分享-中间件平台(ZSS分布式调度平台)



调度器

调度器主要负责执行器的实例管理，状态收集，任务的派发，批处理分片，DAG workflow 节点触发等功能，所有任务都由调度器来管理，分配给具体执行器执行

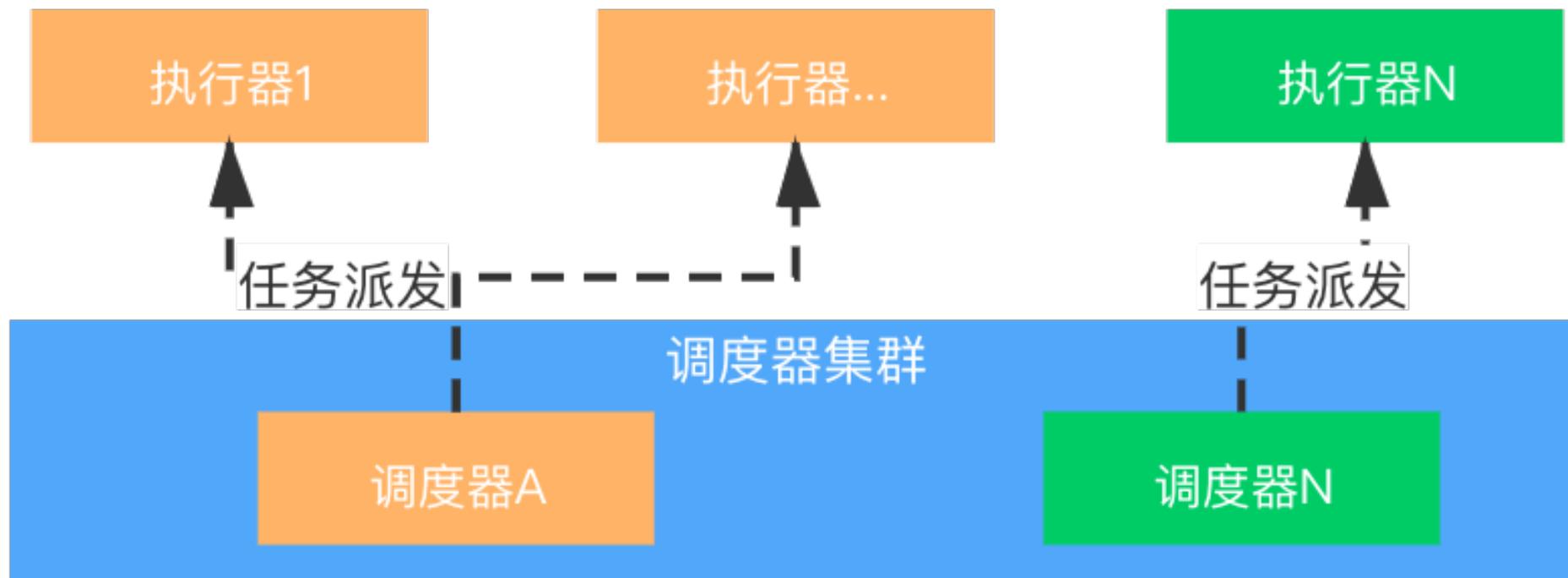
执行器/SDK

- 1、任务注册
- 2、任务执行
- 3、状态上报

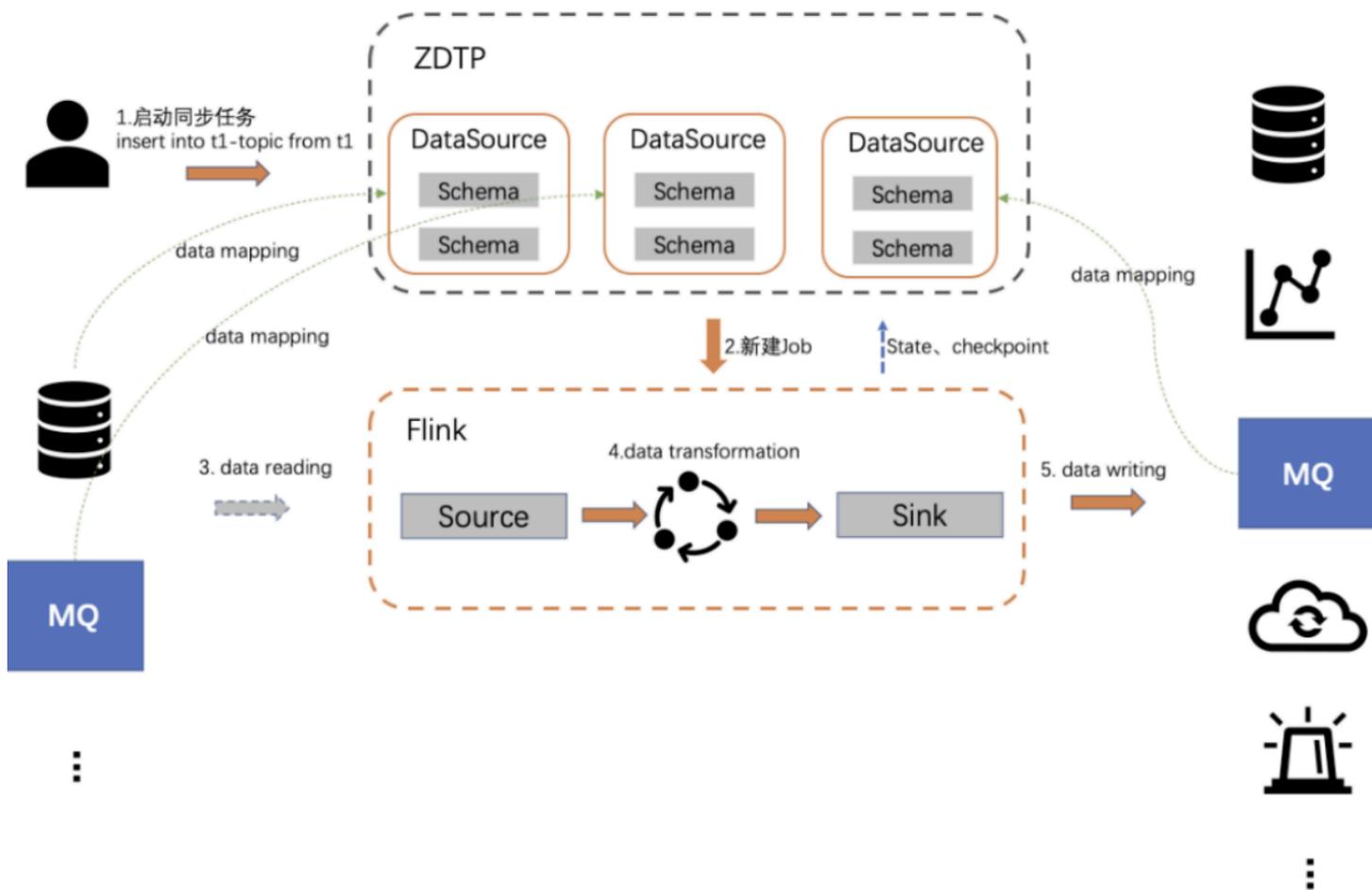
运管平台

包含了多环境的创建、任务元信息的同步、调度器的伸缩能力等以及任务的编排、监控、告警等功能

实践分享-中间件平台(ZSS调度平台)



实践分享-中间件平台(ZDTP同步平台)



任意源->任意目标端

基于Flink CDC扩展各类数据源及管理其Schema，通过简单的SQL实现多个Schema之间的数据同步

数据同步需求自助化

通过平台化的方式区分出不同用户的职责，如运维工程师管理数据源及数据源对应的数据流向审核权，研发工程师负责同步任务创建、申请。

实践分享-中间件平台(ZDTP同步平台)

ZDTP 同步任务 数据源管理 监控告警 2021-03-16 肖文科 - 技术平台部 - 02100 退出

同步任务 > 数据库数据迁移

创建人: [redacted]
创建时间: 2021-03-15 14:55:20
数据源: [redacted]mysql.com
数据源类型: MYSQL
filterStrategy:
firstSyncMode: latest
[更多](#)

同步类型: 增量同步
任务描述: [redacted]
目标端: [redacted]mysql.com
目标端类型: MYSQL
parallelism: 1
[更多](#)

状态
运行中

作业列表 [新增作业](#)

作业Sql	作业状态	启动时间	作业Id	结束时间	读取记录数(Tps)	写入记录数(Tps)	操作
(复制) insert into [redacted] SELECT daily.id, ...	■	2021-0...	(复制) 2e6d0ba6ba8faf50d650c5500156b2c	2021-0...	28301	27417	下载 刷新 删除
(复制) INSERT into [redacted]_new select bus...							
(复制) insert INTO [redacted]_new select oldbu.id, cr...							

03

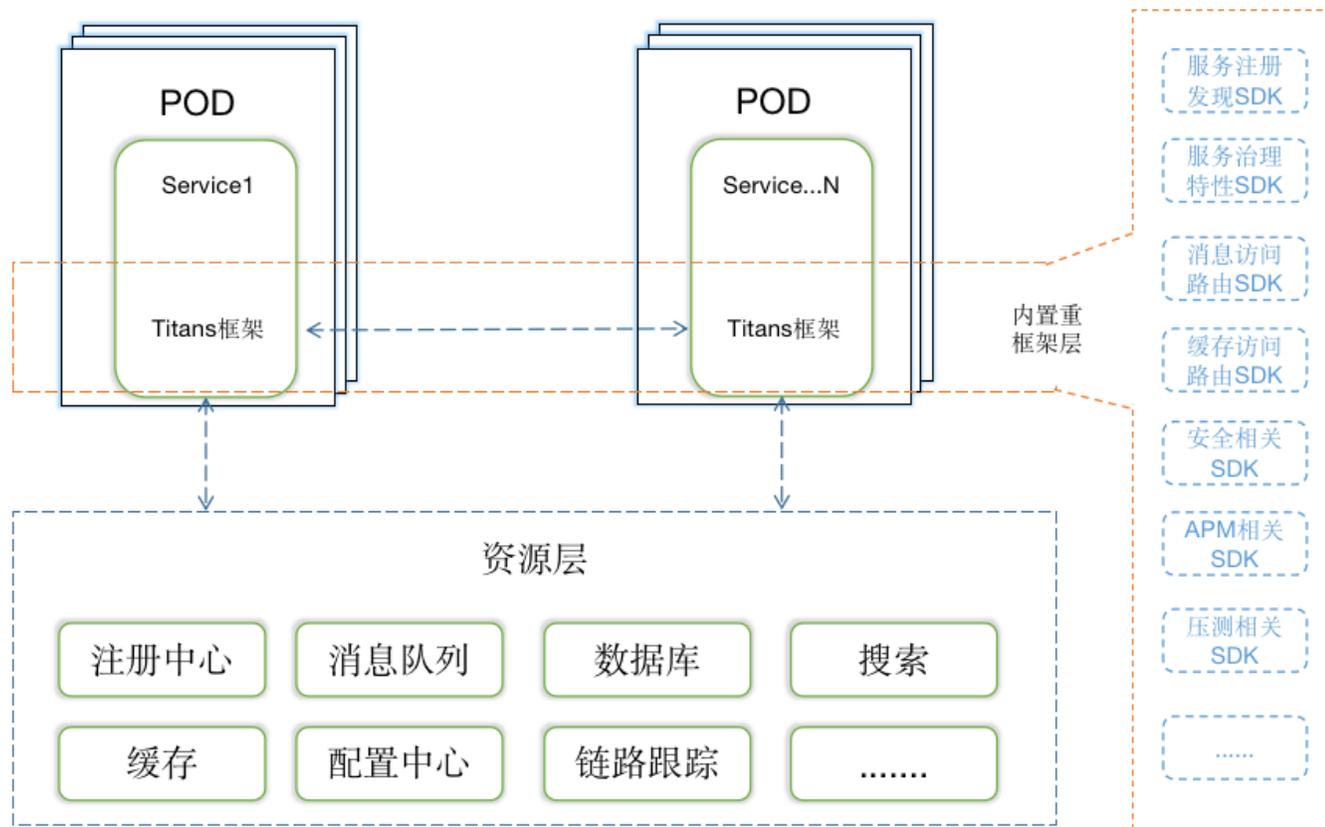
存在问题

描述当前技术中台遇到哪些困难及问题分析



高效协作 勇于担当
积极创新 用户至上

存在的问题



侵入性强、升级成本高

框架SDK侵入业务代码中，边界不够清晰，同时带来框架升级困难，依赖业务方重新打包发布

中间件能力碎片化

随着中间件能力的不断迭代，新的特性必须兼容老的API，造成各种历史包袱，严重拖慢中间件的演进

安全管控能力弱

针对各类资源操作、服务间调用层面上安全策略困难，需要硬编码并升级SDK/插件。效率低下

04

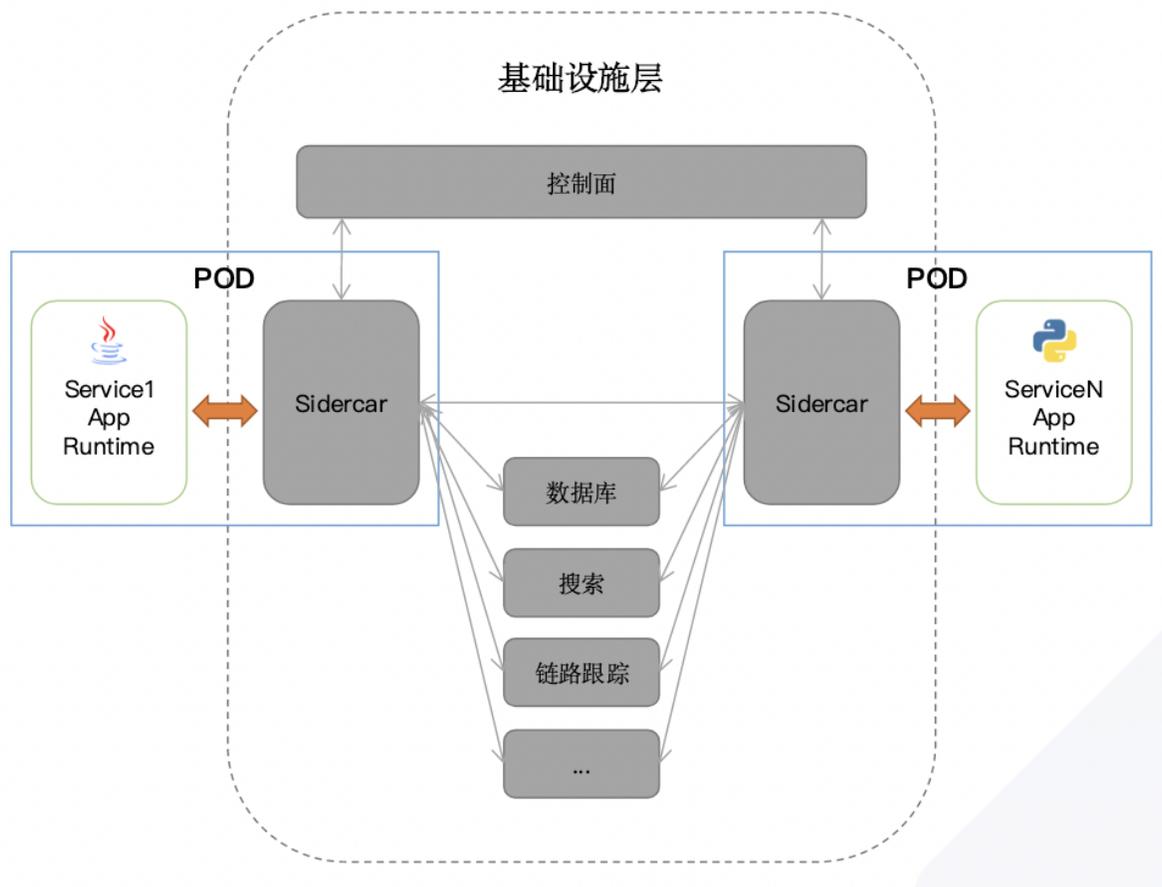
未来规划

从整体上来介绍未来的方向及初步的技术选
型



高效协作 勇于担当
积极创新 用户至上

未来规划-ServiceMesh



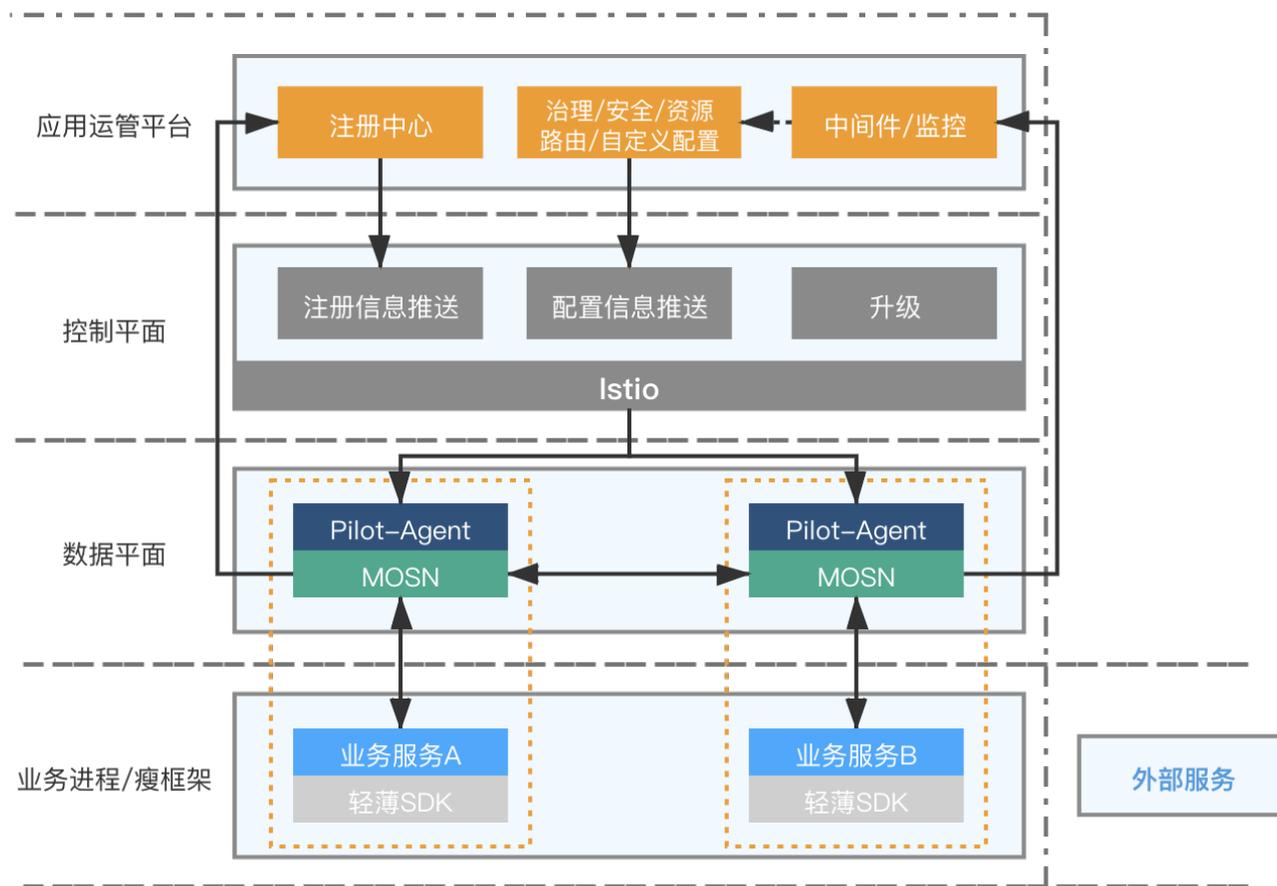
数据面

不仅仅只是支持微服务层面的数据面，期望数据面同时承载分库分表能力、搜索Proxy路由能力、缓存路由能力，并提供标准的资源原生协议（mysql/redis）

工程化

以产品、应用视角将服务的治理、应用的配置、所能使用的中间件资源容量、路由借助控制面统一下发、统一管理，应用只需与sidercar进行交互，无需关注中间件资源具体分布及大小

未来规划-技术选型及初步实现



Sidercar扩展

基于MOSN扩展中间件资源代理

- 1、dubbo协议（自带）
- 2、redis路由（移植ZCacheProxy）
- 3、分库分表（移植小米的Gaea）
- 4、移植ZSearchProxy/ZMS SDK/....

控制面

- 1、自研Dubbo注册中心兼容ZK注册信息同时支持MCP协议
- 2、实现各类配置推送(治理、安全、自定义配置、中间件资源配置、路由规则)
- 3、对接应用运管平台及实现各类数据的上报及采集

QA

QA



有什么想知道的？

What would You like to know?



我想知道...

I want to know...



高效协作 勇于担当
积极创新 用户至上

SACC

THANKS