

# 抖音业务网关的探索和实践

刘福良 / 抖音服务架构-研发体验和效率团队

# 精彩继续！ 更多一线大厂前沿技术案例

上海站



时间：2023年4月21-22日  
地点：上海·明捷万丽酒店

扫码查看大会详情>>



广州站



时间：2023年5月26-27日  
地点：广州·粤海喜来登酒店

扫码查看大会详情>>



- ✓ 背景和技术挑战
- ✓ 架构设计思路
- ✓ 技术实现和业务案例
- ✓ 下一步规划

# 背景:抖音为什么需要引入网关

- 架构复杂度

业务复杂化 -> 架构复杂化 -> 服务拆分

- 通用逻辑升级困难

通用逻辑（比如风控）SDK编译进Go服务，升级大动干戈（拉所有业务方）

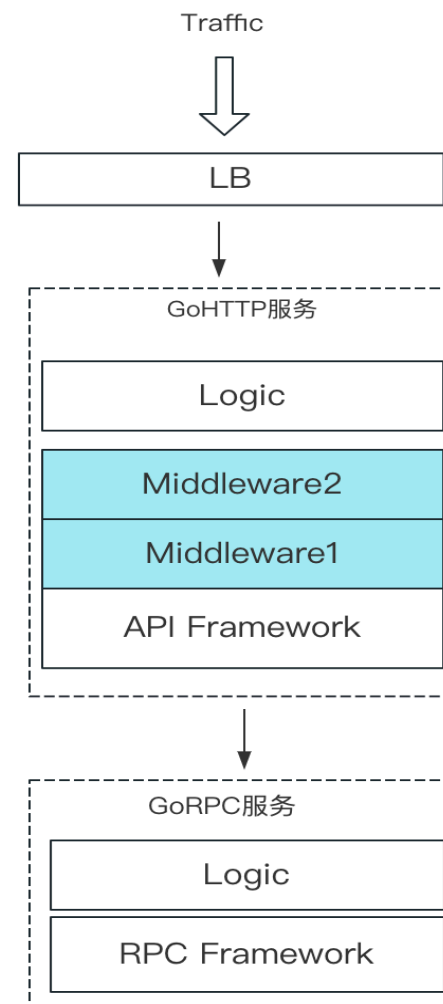
- 冗余的HTTP服务

大部分GoHTTP没有任何逻辑，仅仅是对外暴露HTTP协议

很多GoHTTP服务是为了聚合多个下游接口完成数据编排



维护成本高、链路延迟损耗



2020年之前的架构

# 技术挑战

## 高并发

稳定性要求极高  
5000w+的qps峰值  
运维要求高

## 安全合规

接入成本要极低  
需要统一治理切面  
覆盖所有线上服务

## 前后端解耦

架构灵活性要求高  
多端 iOS/Android/PC/TV/车机  
多版本 抖音/抖音火山版/抖音极速版/抖音PC版

## 端到端协作效率

前后端大型团队协作  
网关作为前后端桥梁  
研发流程串联

- ✓ 背景和技术挑战
- ✓ 架构设计思路
- ✓ 技术实现和业务案例
- ✓ 下一步规划

# 部署模式的选择

- 中心化

独立部署模式，业界主流架构，单点&隔离性差

多接口聚合特点，业务轻量级BFF层，按需接入

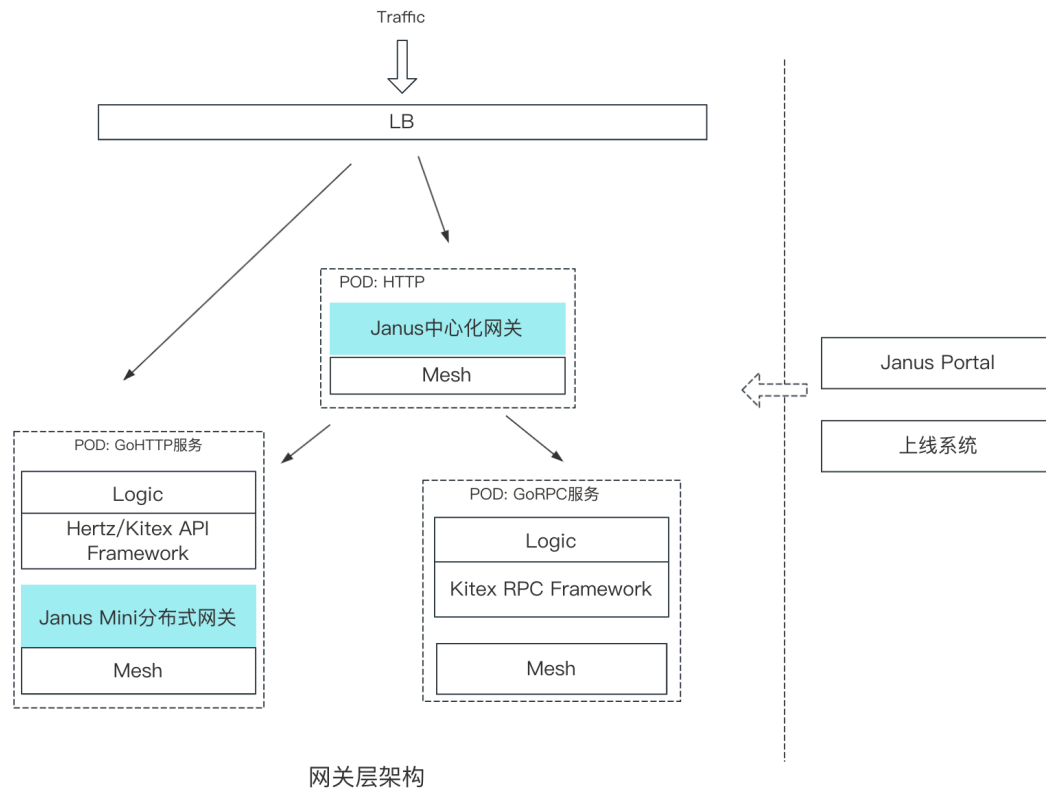
- 分布式

sidecar部署模式，去中心化，极高可用性

可以作为统一网关接入层，默认所有服务接入

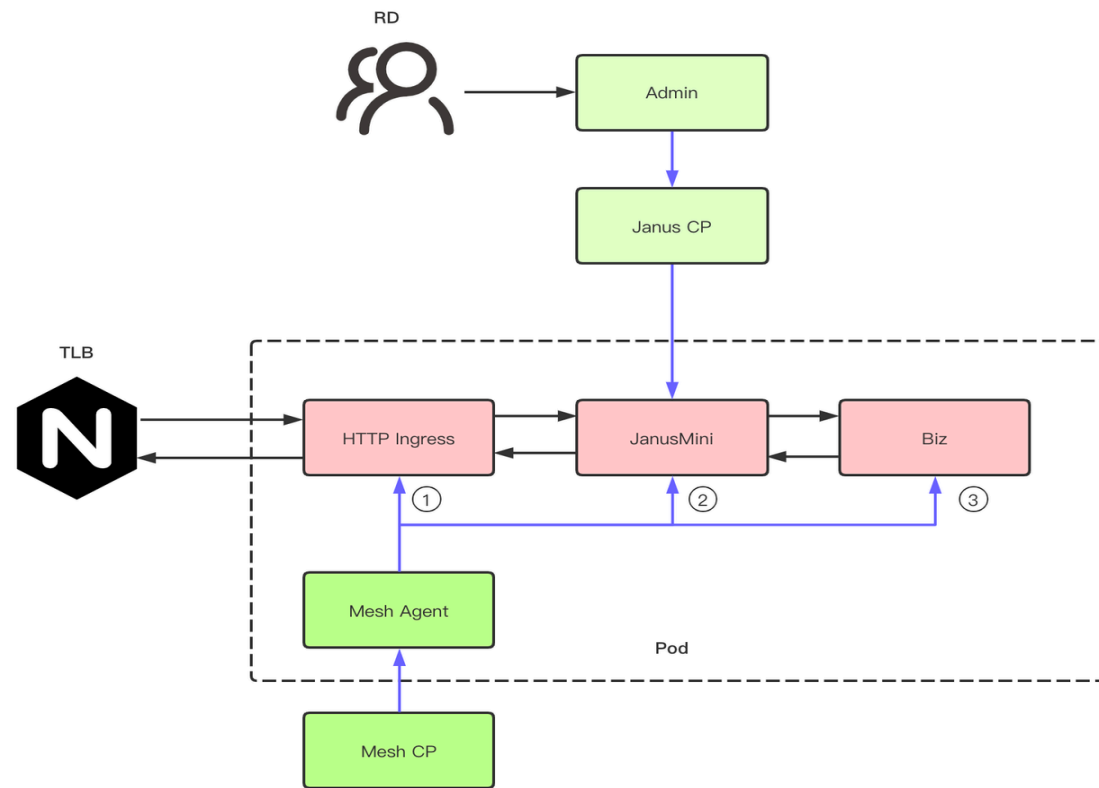


既要还要



# 分布式部署

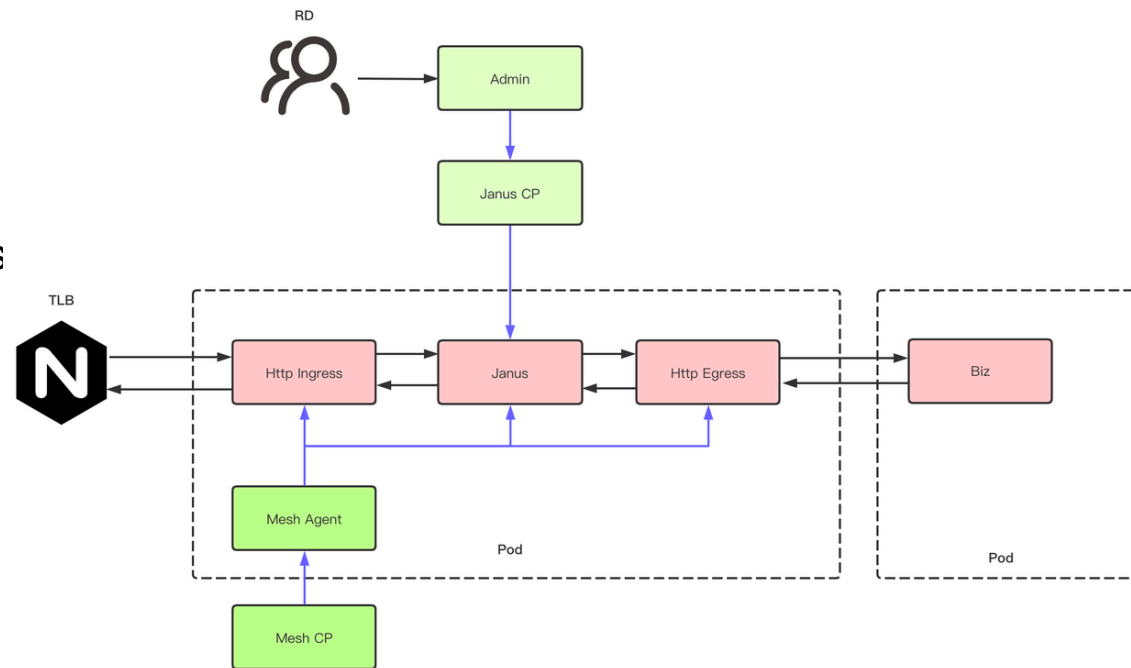
- 网关与Go服务在一个pod内，UDS跨进程通信
- 数据链路: LB -> HTTP Ingress -> 网关 -> Go服务
- mesh\_agent依次启动HTTP Ingress、网关进程、Go服务进程





# 中心化部署

- 网关与Go服务独立部署，跨网络通信
- 数据链路: LB -> HTTP Ingress -> 网关 -> HTTP Ingress -> Go服务
- HTTP/RPC Egress负责服务发现和负载均衡



- ✓ 背景和技术挑战
- ✓ 架构设计思路
- ✓ 技术实现和业务案例
- ✓ 下一步规划

# 网关逻辑执行流程

- **Router**

HTTP协议层

- **Loader**

通用逻辑、扩展能力

- **Protocol Conversion**

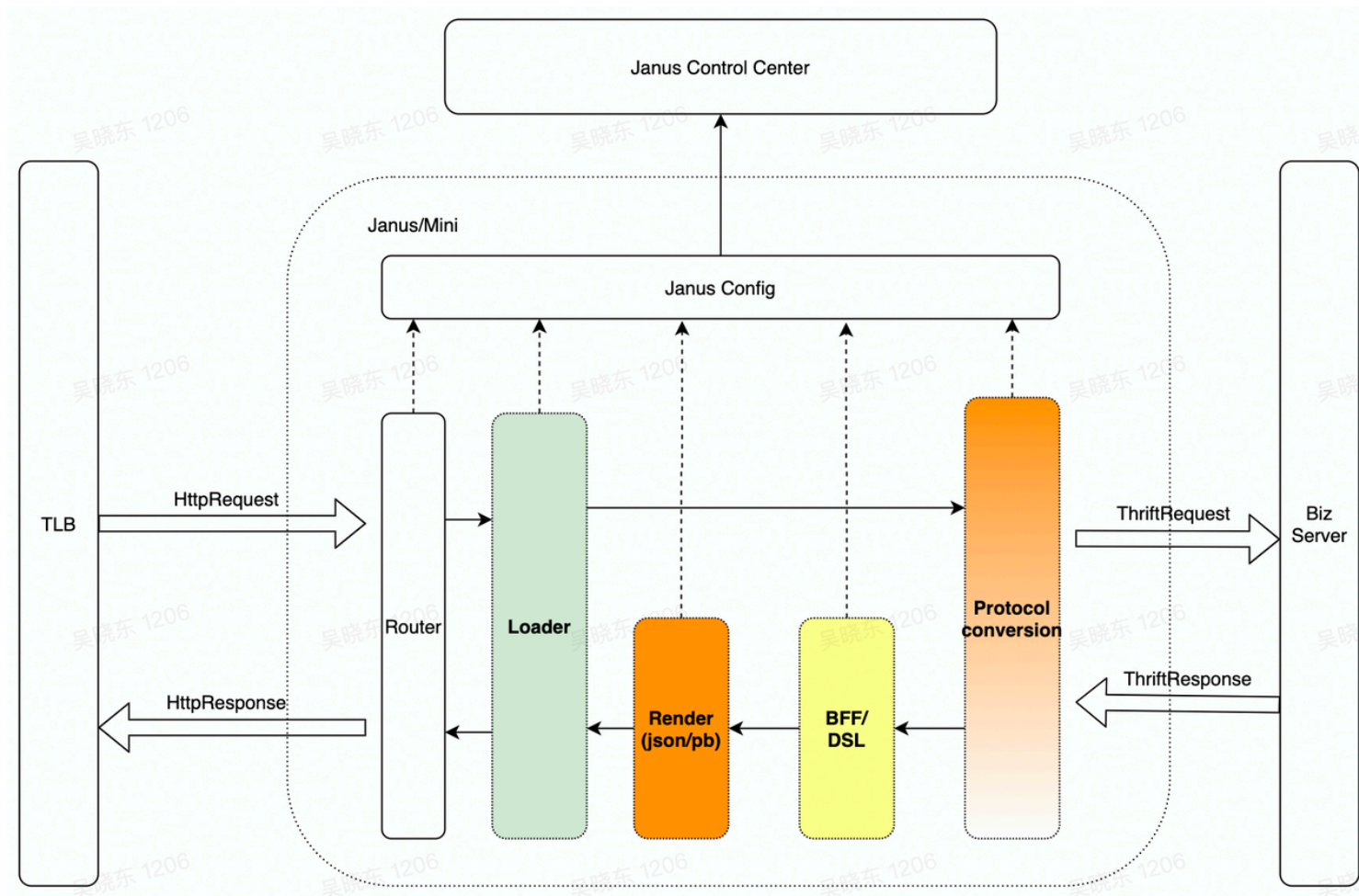
HTTP -> Thrift RPC协议转换

- **BFF/DSL**

BFF接口聚合、数据编排、裁剪

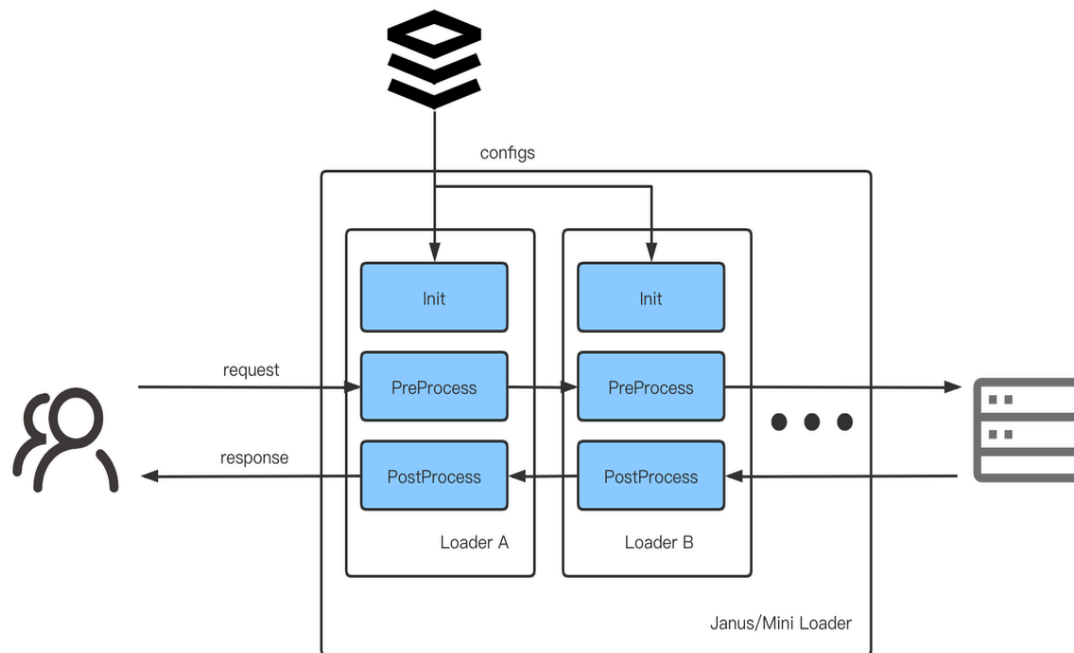
- **Render (json/pb)**

响应数据序列化



# 通用逻辑托管/Loader

- 通用逻辑与业务逻辑解耦，治理团队与业务团队解耦
- 通用逻辑与服务框架（Go/Python/Java）解耦
- 灵活的变更和热升级，高效API治理成为可能
- 案例：全场景身份注入与透传



# BFF/GraphQL

- BFF适配层

接口聚合

数据编排

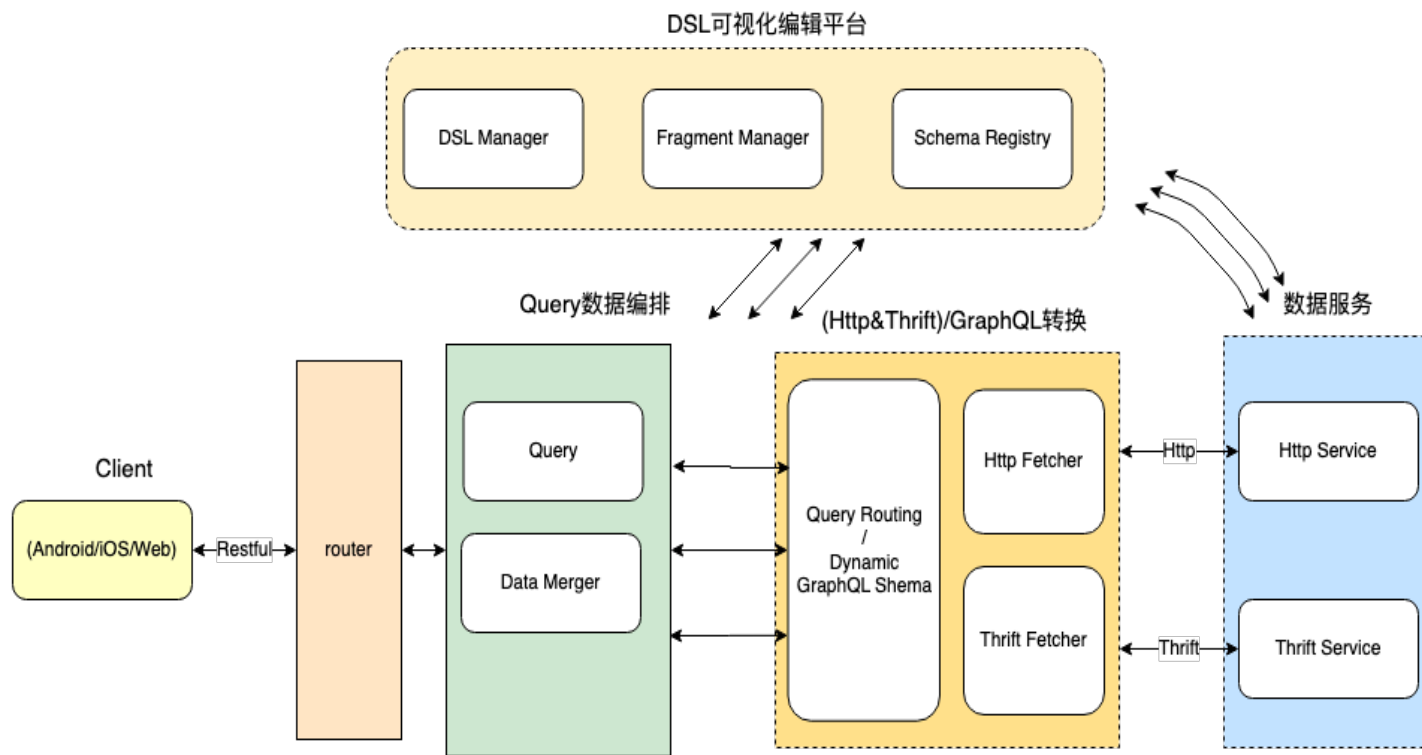
字段裁剪

- 基于GraphQL的网关BFF实现

Thrift IDL -> GraphQL schema

GraphQL引擎与Query表达式

GraphQL的版本管理



# BFF/GraphQL: 一个例子

## API Endpoint

```
// user接口
{
  "id": 1,
  "name": "zhangsan",
  "email": "zhangsan@bytedance.com",
  "extra": {
    "app_id": 1112,
    "update_date": "20200101"
  }
}
```

```
// video接口
{
  "id": "2",
  "title": "test",
  "cover": "cover.png",
  "quality_info": [
    {
      "bit_rate": 20000,
      "uri": "uri_20000.m3u8"
    },
    {
      "bit_rate": 40000,
      "uri": "uri_40000.m3u8"
    }
  ]
}
```



## GraphQL Query

```
{
  $user {
    id
    name
    extra @unbox {
      app_id
    }
  }
  $video @optional{
    id
    cover
    quality_info {
      uri
    }
  }
}
```



```
{
  "data": {
    "user": {
      "app_id": 1112,
      "id": 1,
      "name": "zhangsan"
    },
    "video": {
      "cover": "cover.png",
      "id": "2",
      "quality_info": [
        {
          "uri": "uri_20000.m3u8"
        },
        {
          "uri": "uri_40000.m3u8"
        }
      ]
    }
  },
  "status_code": 0,
}
```

# API Workflow

- API设计优先

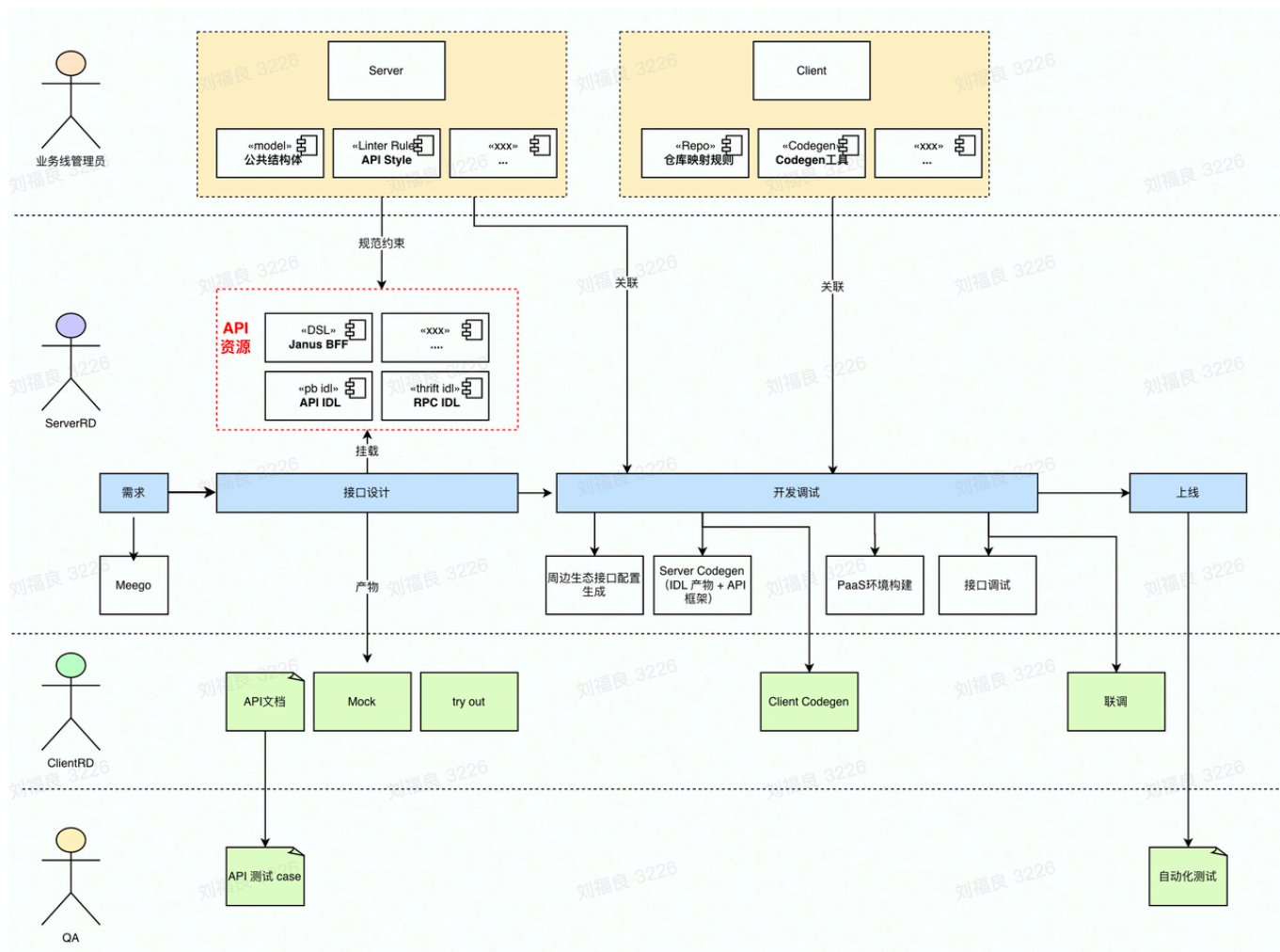
前置IDL设计节点，解耦各端研发流程

IDL规范与流程约束，提高 API 变更效率和稳定性

- 端到端的协作效率

串联研发活动各个节点，提供高效流畅的研发 workflow

研发流程标准化和在线化，研发效率和质量提升



# 业务案例：抖音视频Feed场景

## • 场景

抖音刷视频接口

服务端领域数据 vs 客户端展示数据

P0级接口，稳定性要求极高

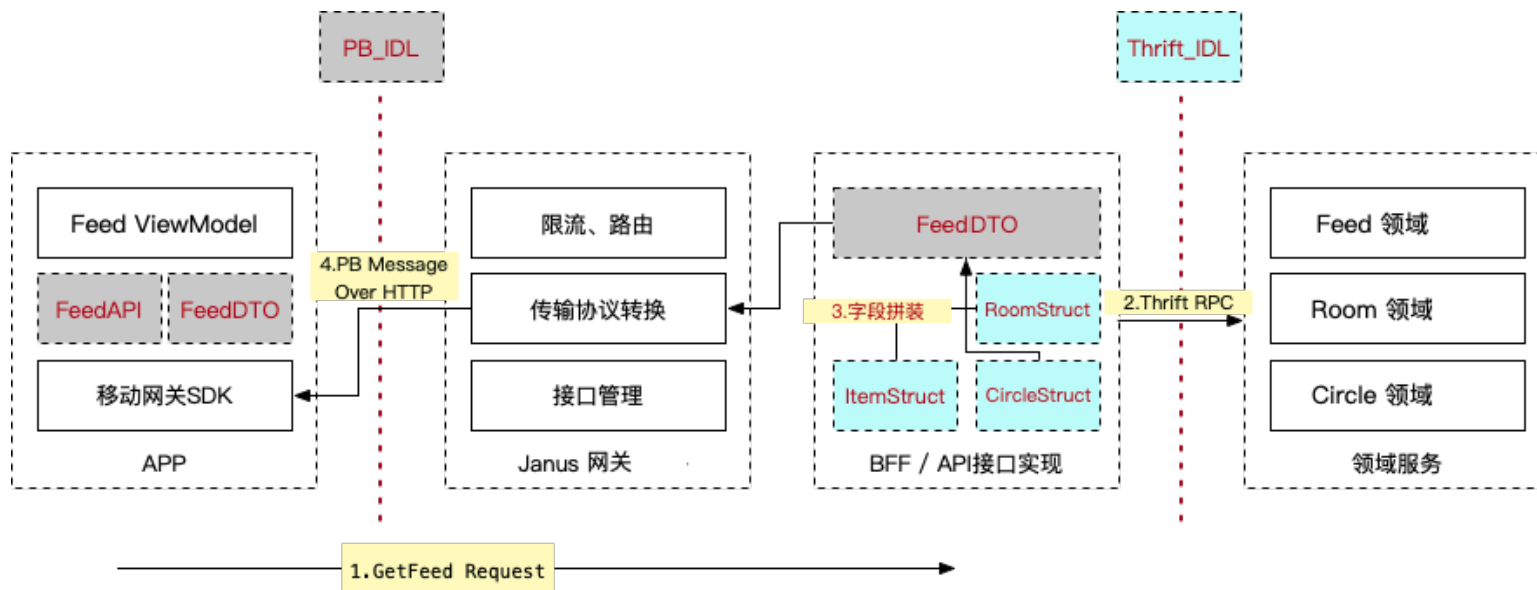
## • 通用逻辑

安全风控、容灾兜底、结果采样校验等

## • BFF

服务端Thrift RPC (DO) -> 客户端PB HTTP (DTO)

字段的重新编排与裁剪（展示逻辑）





- ✓ 背景和技术挑战
- ✓ 架构设计思路
- ✓ 技术实现和业务案例
- ✓ 下一步规划



## 下一步规划

- **API治理平台**

API网关向API治理平台演进，API信息中心、API字段治理与流程卡点、隐私合规等

- **API研发平台**

API Workflow向研发平台演进，打通API&RPC交付全生命周期，开发环境、代码生成、调试等

QA

加我微信，线下继续交流



良小白

北京 海淀



扫一扫上面的二维码图案，加我为朋友。

THANKS

