ArchSummit
全球架构师峰会 2017

主办方 Geekbang 极客邦科技 InfoQ

滴滴出行平台的高可用实践

陈宜明
滴滴出行-技术专家

极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新

扫一扫下载极客时间App

主办方 Geekbang. 极客邦科技 InfoQ

AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14　北京国际会议中心

扫描关注大会官网

SPEAKER
INTRODUCE

陈宜明

滴滴出行 技术专家

来自滴滴稳定性和效率团队，负责异地多活、一键降级、防火放火等工作。

16年加入滴滴，之前在百度工作，有多年架构开发经验。

TABLE OF

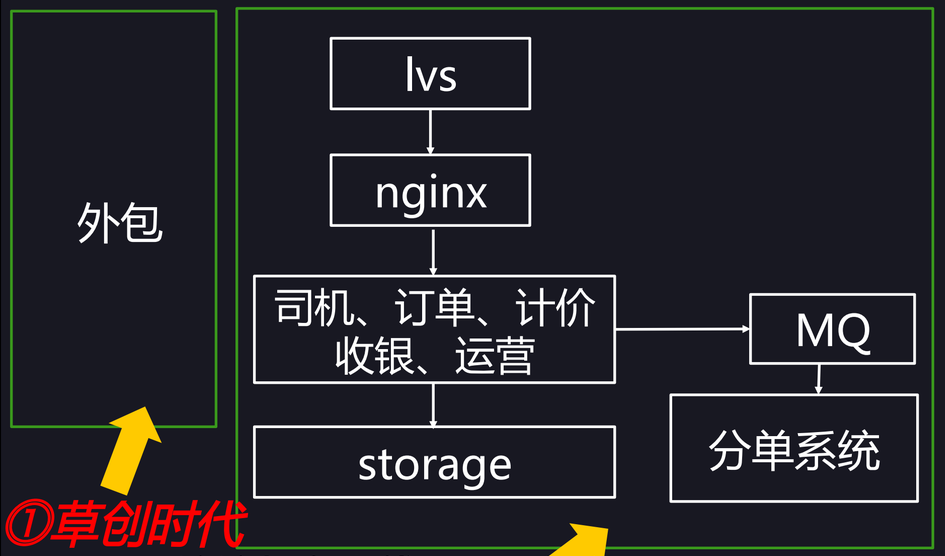
CONTENTS 大纲

- 滴滴的出行业务架构
- 高可用方法论
- 异地多活
- 一键降级
- 防火放火

滴滴业务简介

交易业务： 实时、多状态、长链条

业务架构演进

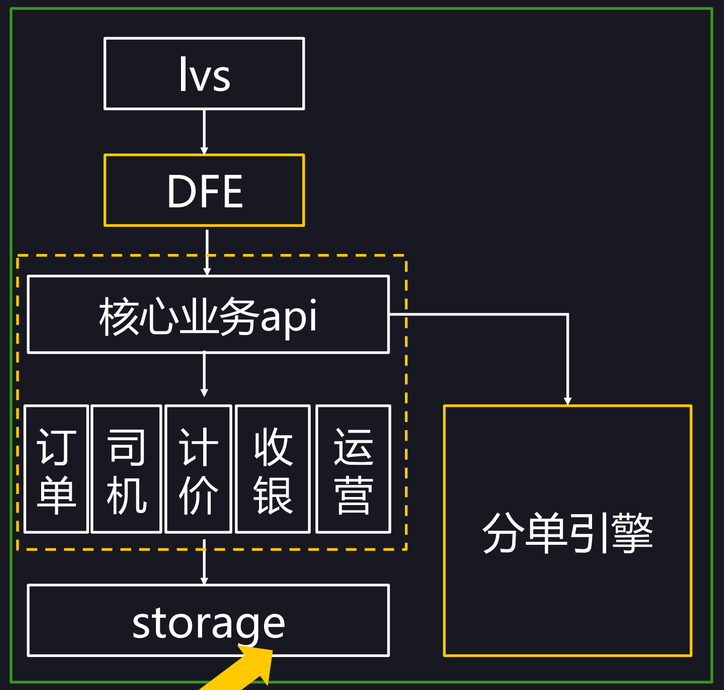


①草创时代

- 2012.9滴滴打车上线
- 2013.8 1kw+用户

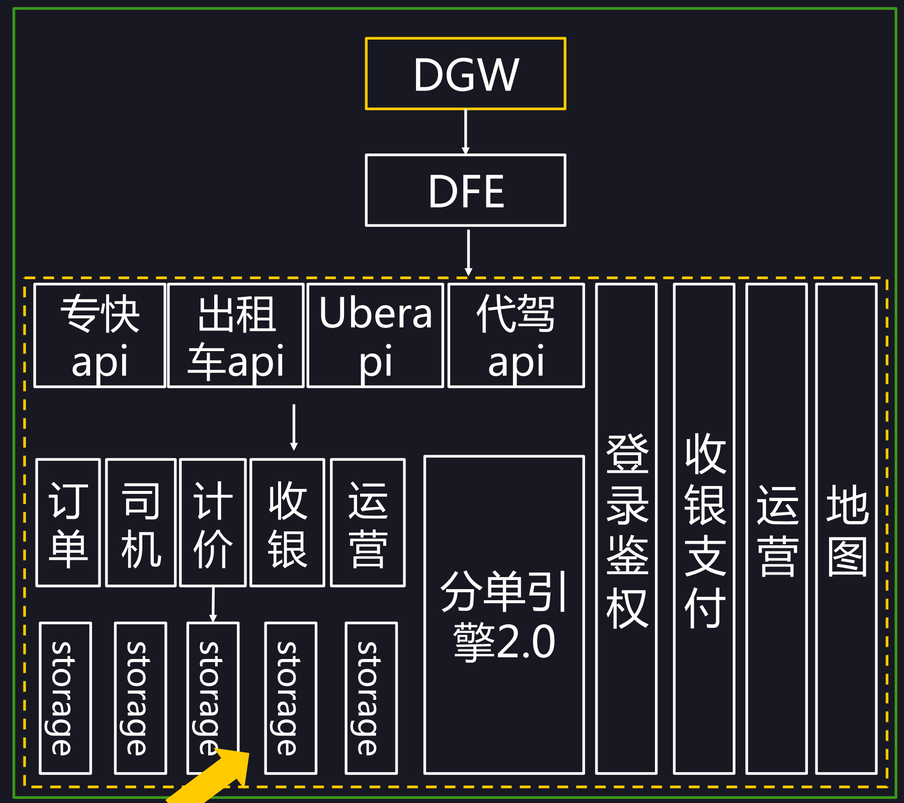
②红包大战

- 2014.3 单量300万单/天
  乘客1亿 司机100万



③专快车上线

- 2014.8 专车上线
- 2015.5 快车上线
- 2016.3 1000万单/天



④Uber合并

- 2016.8 收购Uber中国
- 2016.10 2000万单/天

日订单量： 15年几百万 -> 目前2500w+（仅次于淘宝）

高可用面临的挑战

业务增长迅速
节假日效应明显

实时
多状态
交易型
链路长

新场景多
迭代快

流量增
长迅猛

业务复杂

高速路上
换轮子

稳定性
挑战大

接口调用链条示例

业务调用链路

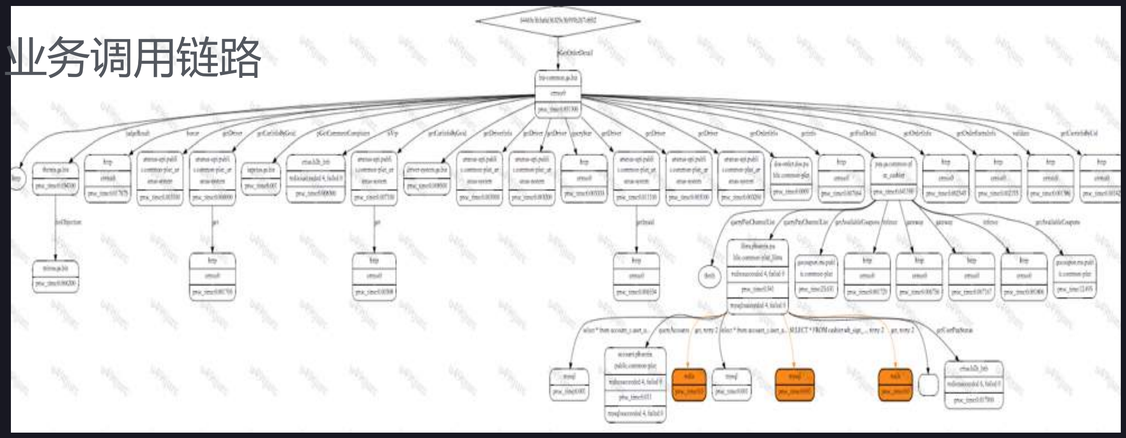


TABLE OF
CONTENTS 大纲

- 滴滴出行的业务架构
- 高可用方法论
- 异地多活
- 一键降级
- 防火放火

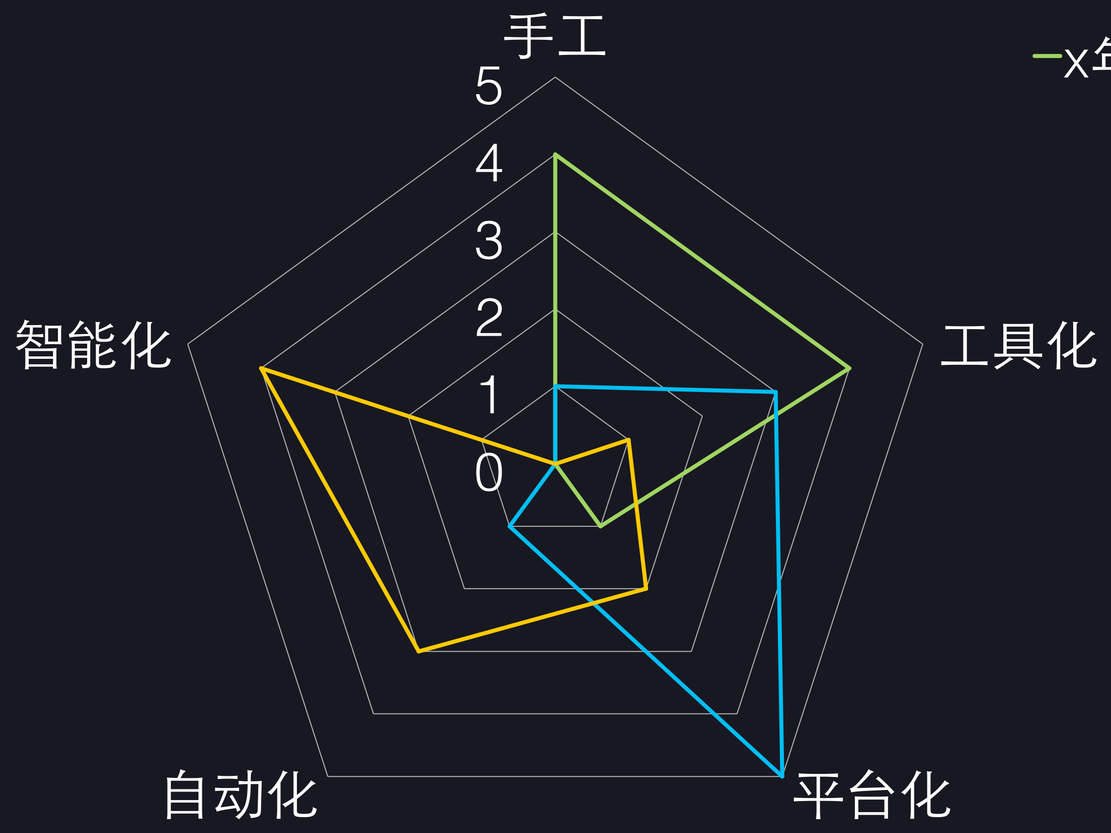
高可用的常见措施

不可用因素	典型case	增大MTBF	缩短MTTR
程序、数据和配置bug	程序出core、配置格式出错	研发质量、测试质量、变更分级、解耦减少变更	监控告警、快速回滚
机器和网段级故障	宕机、边缘交换机板卡故障、光纤抖动	硬件冗余	预警预迁移服务、切流到本机房冗余、数据主从切换
多网段和机房级故障	核心交换机故障、链路割接、机房掉电	硬件冗余（包括多机房）	预警预迁移服务，切流到其他机房
流量	大促、节假日和特殊天气、外部攻击、上游重试雪崩	上游容错调度防雪崩	容量规划、防攻击、其他同容量不足
容量	主流程服务容量不足	容量规划、容量预警	限流、切流其他冗余、降级、熔断弱依赖、快速扩容
依赖服务	账单依赖的到达时间预估故障、分单依赖的特征服务故障	递归使用前述方法提高该依赖的可用性	熔断弱依赖，或递归使用前述方法提高该依赖的可用性

高可用的8大抓手

抓手	典型做法	业务	平台	服务
研发质量	容错设计、cr、单测、稳定性评审	弱依赖化（主流程瘦身）、数据流治理、研发流程	scmpf流程平台	rpc框架、服务组件
测试质量	线下仿真	仿真环境建设、测试流程	仿真环境解决方案、测试框架	支持引流、dump
变更管理	按机器或流量分级发布、多维度质量检测	灰度发布、检查和回滚流程	部署系统、分级发布系统	服务发现、配置中心
监控告警	机器/进程/业务监控及报警	监控大盘、多级报警	监控系统、告警系统	metrics、trace
故障预案	定位和止损的预案	预案建设	异地多活、一键预案/降级	中间件支持切流、限流、熔断、降级
容量规划	全链路压测、子链路压测、哨兵压测	改造支持各压测	压测平台	中间件支持压测
放火盲测	弱依赖验证、预案有效性和完备性验证	请求级放火、资源放火	放火盲测平台	中间件支持放火
值班巡检	例行值班表、节假日值班	例行值班、集中应急处理	值班平台	——

高可用的5级演进目标—43210



43210召回占比演进

TABLE OF

CONTENTS 大纲

- 滴滴出行的业务架构
- 高可用方法论
- 异地多活
- 一键降级
- 防火放火

异地多活

一个脚本引发的“血案”...？

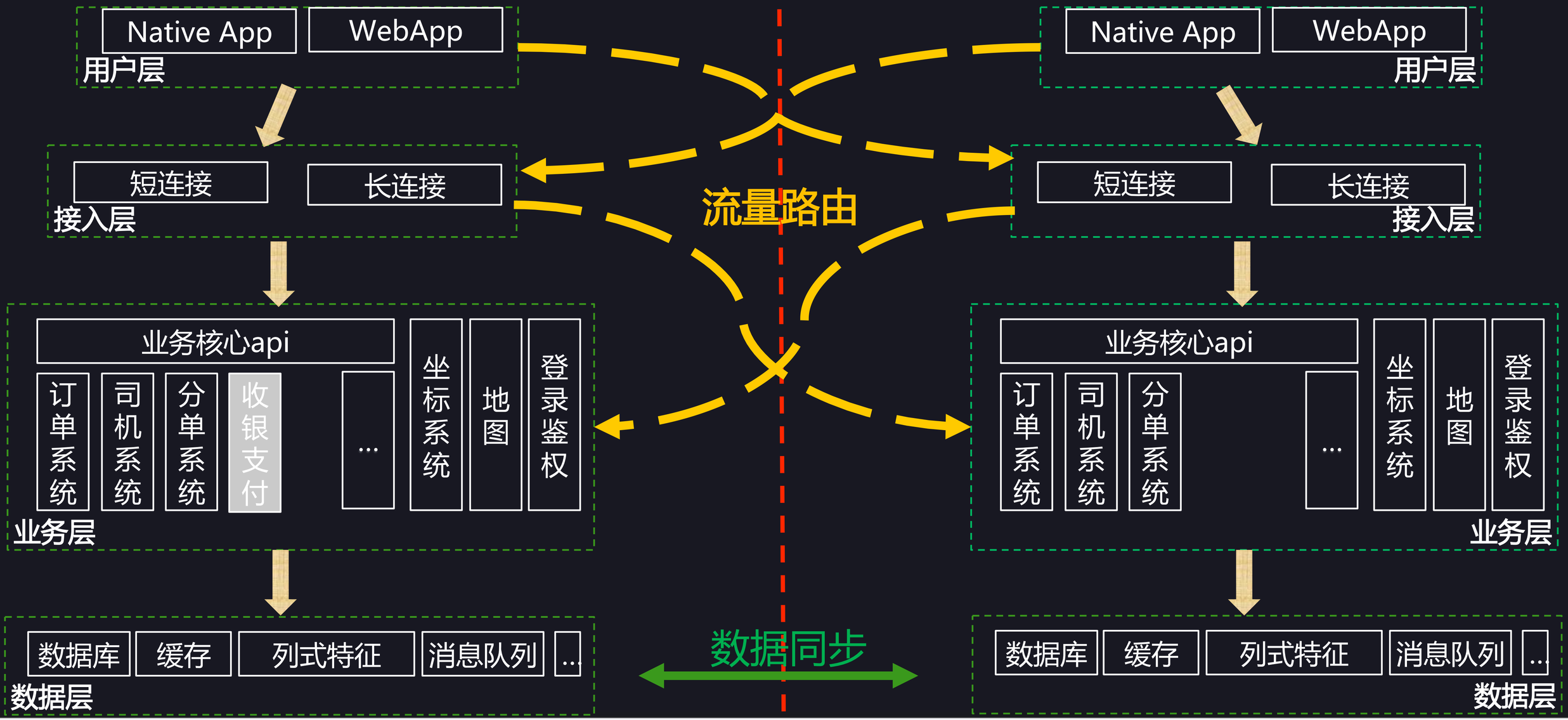
哪些服务多活？

同城还是异地？

https://bbsimg.feidee.com/data/attachment/forum/201508/19/155520wtajnigimiz3jqgk.jpg

如何实现多活？

多活架构


Native App
WebApp
用户层
Native App
WebApp
用户层
短连接
长连接
接入层
流量路由
短连接
长连接
接入层
业务核心api
订单系统
司机系统
分单系统
收银支付
...
坐标系统
地图
登录鉴权
业务层
业务核心api
订单系统
司机系统
分单系统
...
坐标系统
地图
登录鉴权
业务层
数据库
缓存
列式特征
消息队列
...
数据层
数据同步
数据库
缓存
列式特征
消息队列
...
数据层

流量路由

流量如何划分？
流量标识如何传递？
路由如何决策？
单活如何访问多活？
跨城、漫游如何处理？
为什么分层切换？

Native App
WebApp
用户层

短连接
长连接
接入层

业务核心api
订单系统
司机系统
分单系统
收银支付
...
坐标系统
地图
登录鉴权
业务层

流量路由
单元内访问、不要跨机房

Native App
WebApp
用户层

短连接
长连接
接入层

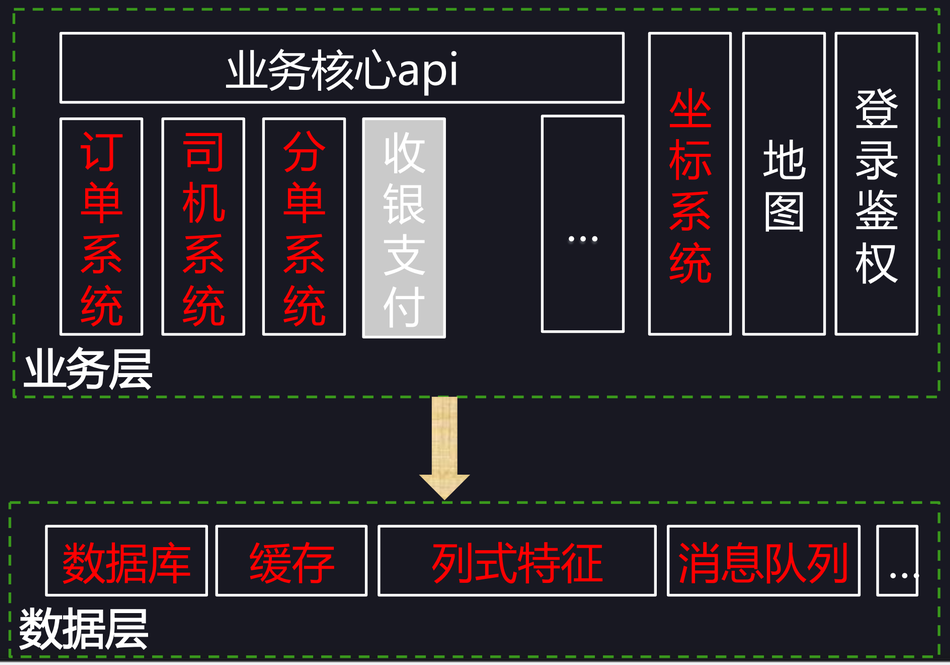
业务核心api
订单系统
司机系统
分单系统
...
坐标系统
地图
登录鉴权
业务层

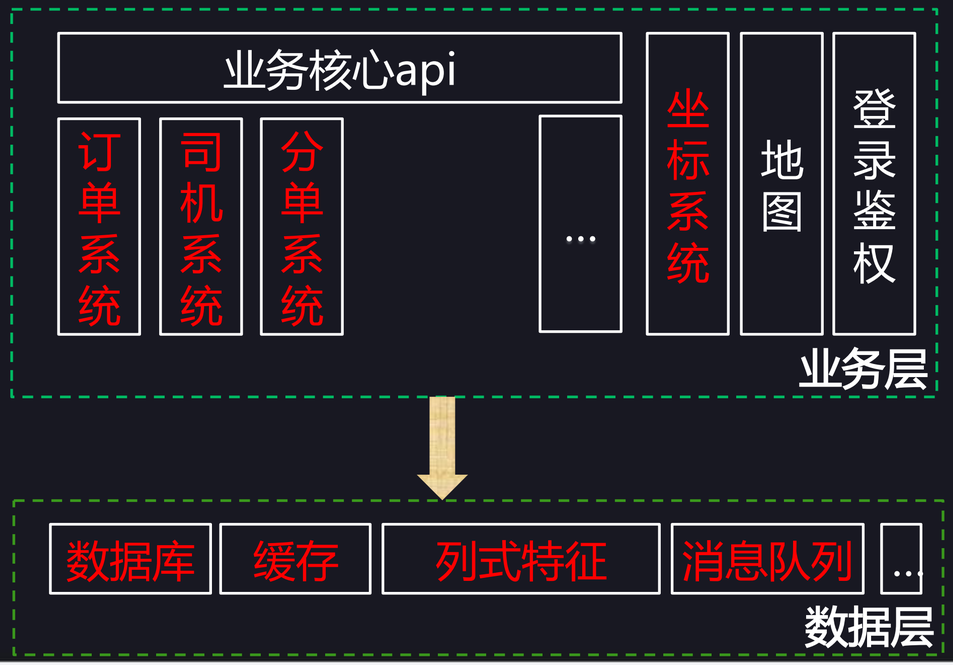
数据同步

一致性挑战：成功率、延迟、有序、不重

业务层的挑战： 不同系统有不同的数据特性

司机系统：短时问题可容忍，但数据修复麻烦
订单系统：强一致性要求，但修复相对简单
分单系统：短时问题可容忍
坐标流：获取最近的数据，部分丢失无影响





数据同步

系统	数据特征	分析	存储	一致性	系统特性	同步方案
司机系统	身份信息	静态变化小	数据库、缓存	无需考虑	1、短时问题可容忍 2、db出问题修复麻烦	1、数据库*主从*同步，写主读从 2、缓存通过*proxy互写*同步
	是否忙碌、是否出车、座位数	关键因子	数据库、缓存、列式特征	中偏高		
	策略数据（服务分、围栏、新政）	非关键因子	列式特征	中偏低		
订单系统	起始位置等信息	静态变化小	数据库、缓存	无需考虑	相对修复简单 1、乘客直接结束订单再次发单 2、客服通过接口强制关单	1、数据库*主从同步*，成交主流程写主读主 2、缓存：有序不重 • 双集群校验 • *binlog反冲*，最终一致
	订单状态6-7个（状态机）	状态错误，无法继续	数据库、缓存	高		
分单系统	司机和乘客特征	短时可接受	列式特征	中偏低	特征出问题，可从数据库回捞： 1、手工，听单检测 → 收车出车 2、服务端旁路检测司机状态	在*业务proxy层*实现主从同步（类数据库）
坐标流	司机乘客坐标信息		内存	低	获取最近产生的数据，可容忍数据丢失	实现容易，在*业务proxy层互写*
mq			消息队列	低	异步数据，一致性要求不高	全量*互同步*

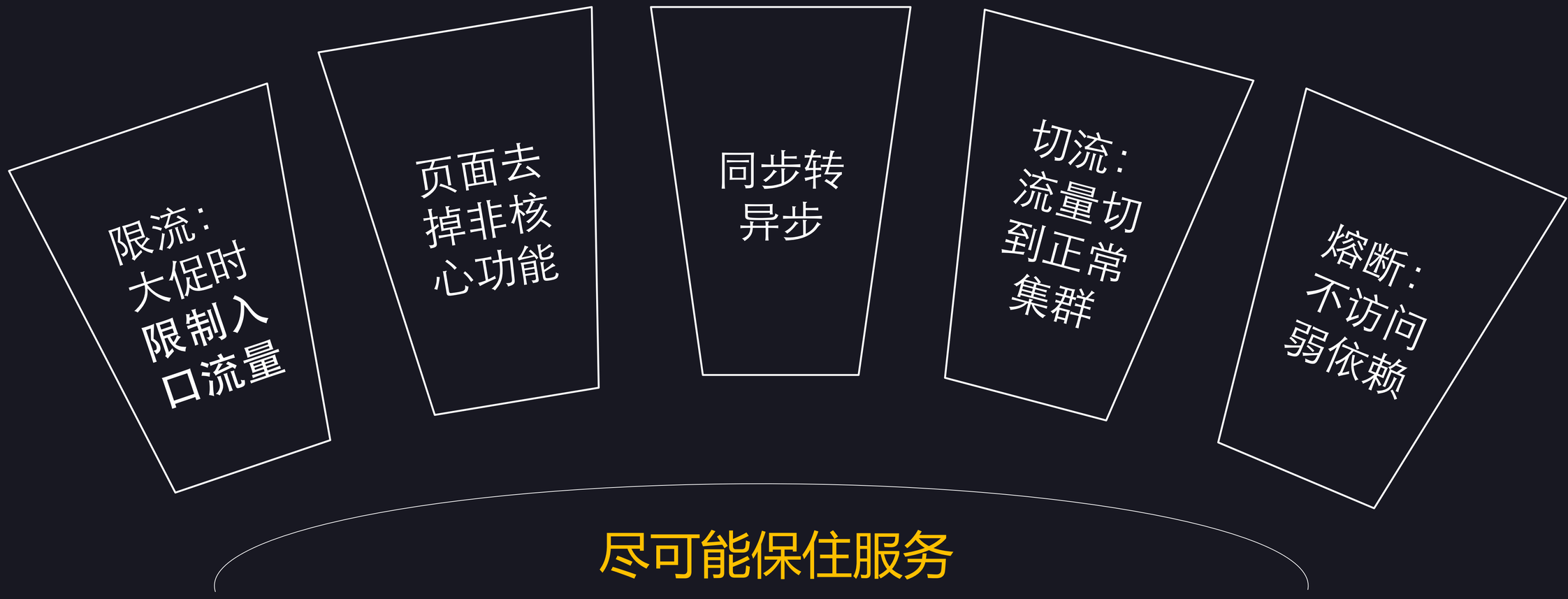
降级预案

无状态业务	数据故障	DB主从延迟	有损降级
多活： 切流 单活： 熔断	特征库数据异常： DB回捞 DB挂了： 主从切换	网络抖动： 短时限流防雪崩、长时切流到主机房 抖动+主力机房挂：超小概率、最小系统	计价、服务分有损：善后补偿

TABLE OF
CONTENTS 大纲

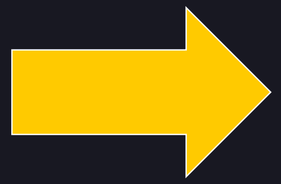
- 滴滴出行的业务架构
- 高可用方法论
- 异地多活
- 一键降级
- 防火放火

What ?


限流：
大促时
限制入
口流量
页面去
掉非核
心功能
同步转
异步
切流：
流量切
到正常
集群
熔断：
不访问
弱依赖
尽可能保住服务

Why ?

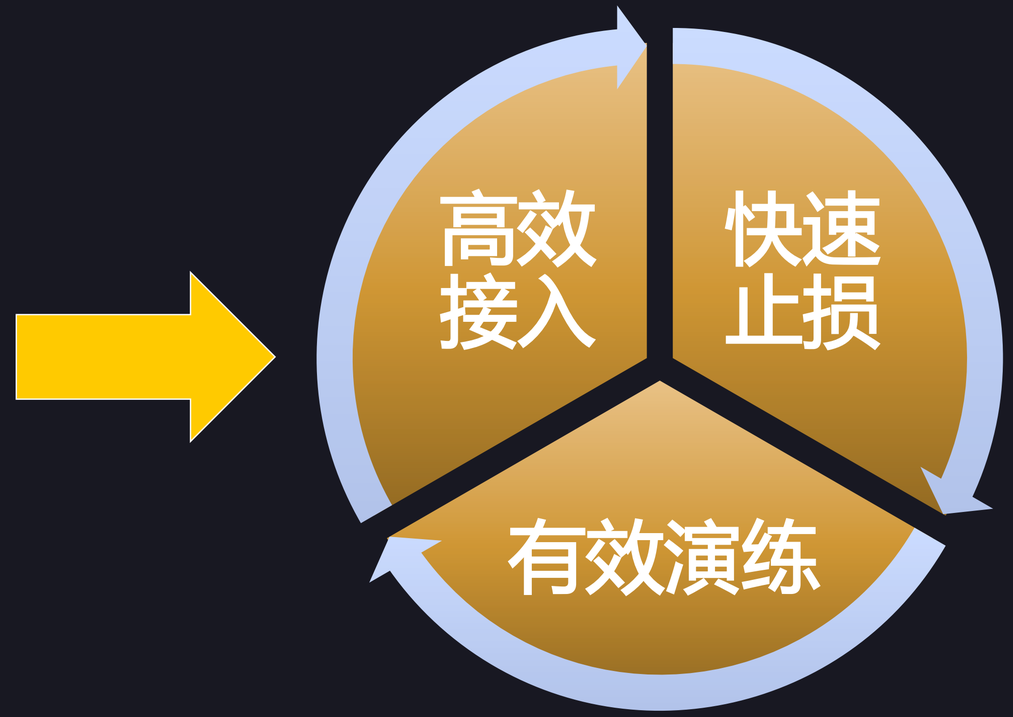
- 业务出问题不可避免
- 需要上线，止损慢
- 预案有冲突、容易失效
- 业务压力大，精力有限



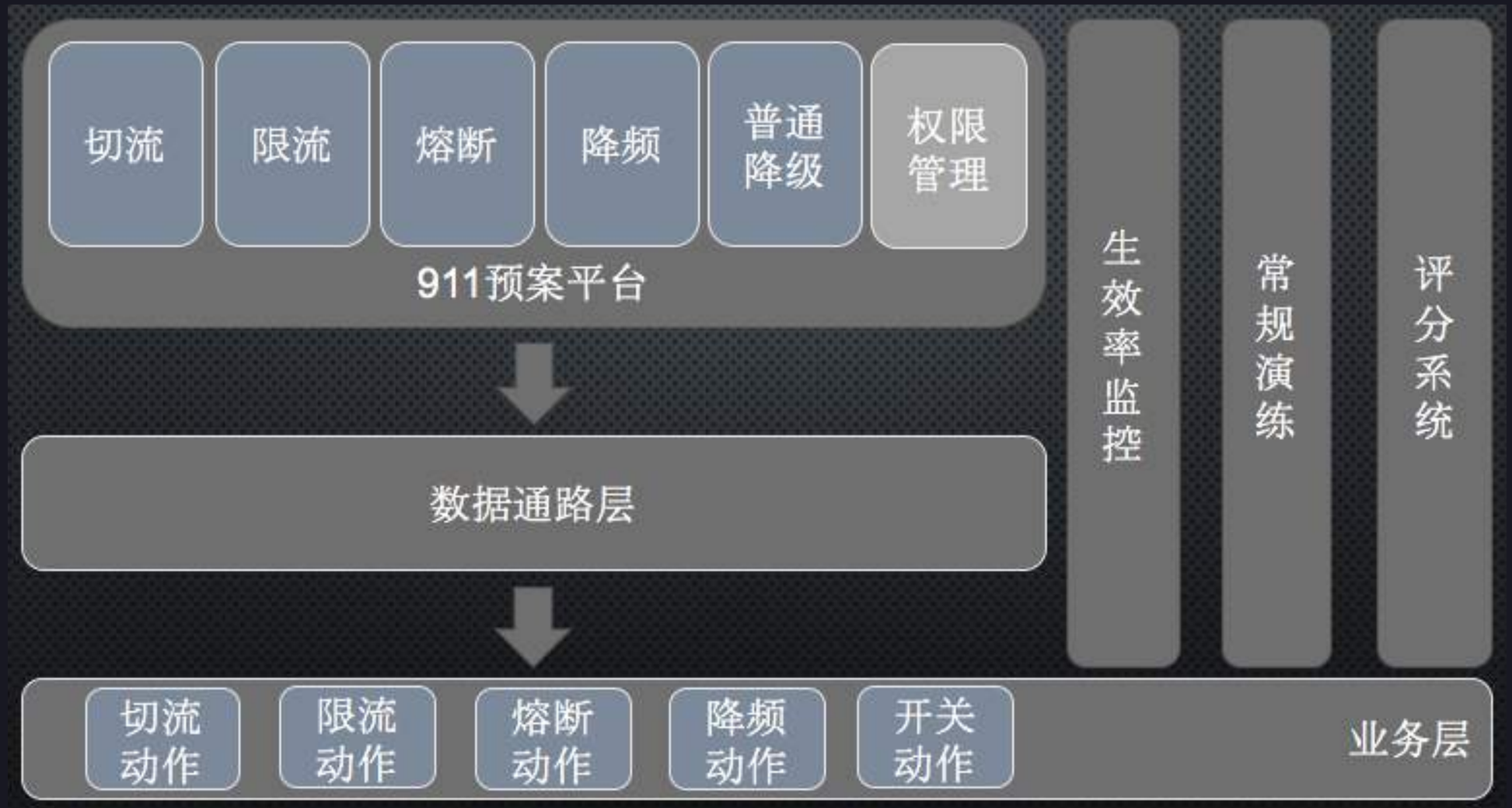
- ✓ 要有降级预案
- ✓ 快速生效止损
- ✓ 预案管理
- ✓ 降低接入成本

How ?

- ✓ 场景预案，一键快速生效
  - L1: 业务无损： 号码保护、不作弊、导流、切流
  - L2: 部分效果受损： 动调，计价（路面距离降级为直线距离）
  - L3: 核心支付效果有损： 收银熔断、乘客未支付可以发单
  - L4: 核心主流程效果受损： 发单限流、内部丢单
- ✓ 移动+pc双端 随时触达
- ✓ 生效率监控、灰度发布、平台双活、互斥管理 安全生效
- ✓ 切流、限流、熔断、普通降级配置语义+中间件action实现
- ✓ 评分系统驱动接入和演练



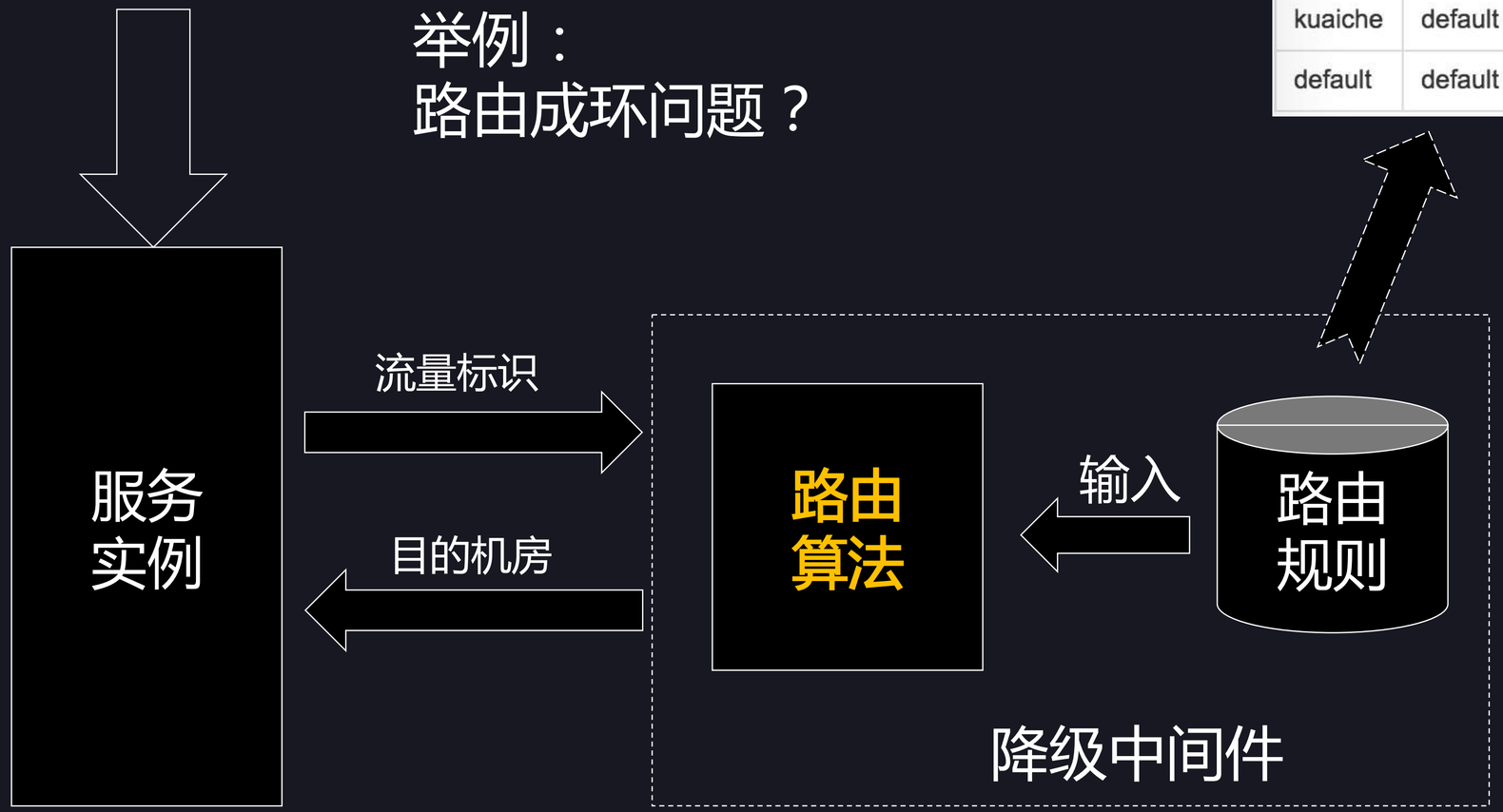
Detail


切流
限流
熔断
降频
普通降级
权限管理
911预案平台
数据通路层
切流动作
限流动作
熔断动作
降频动作
开关动作
业务层
生效率监控
常规演练
评分系统

切流实现

举例：
路由成环问题？

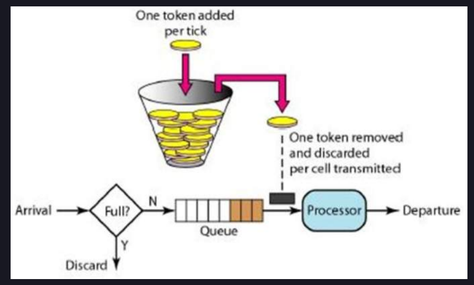
产品线	城市	目的机房	规则编号
taxi	1	idc1	①
kuaiche	13	idc1	②
kuaiche	default	idc2	③
default	default	idc1	④

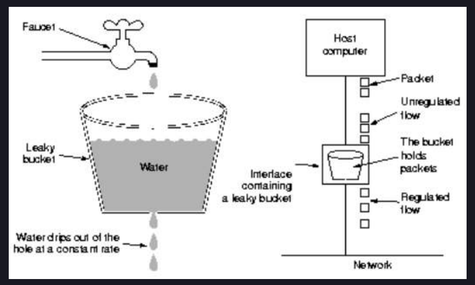


平台动态配置

通路实时配送

限流实现





令牌桶：支持突发
漏桶：强限固定的速度

服务实例

{caller、callee、method}

是否限流

令牌桶算法

输入

限流规则

降级中间件

限流配置

caller	callee	rate	burst	ruleid
taxi-api	/order-v3/getOrderDetail	100	200	①

平台动态配置

通路实时配送

熔断实现

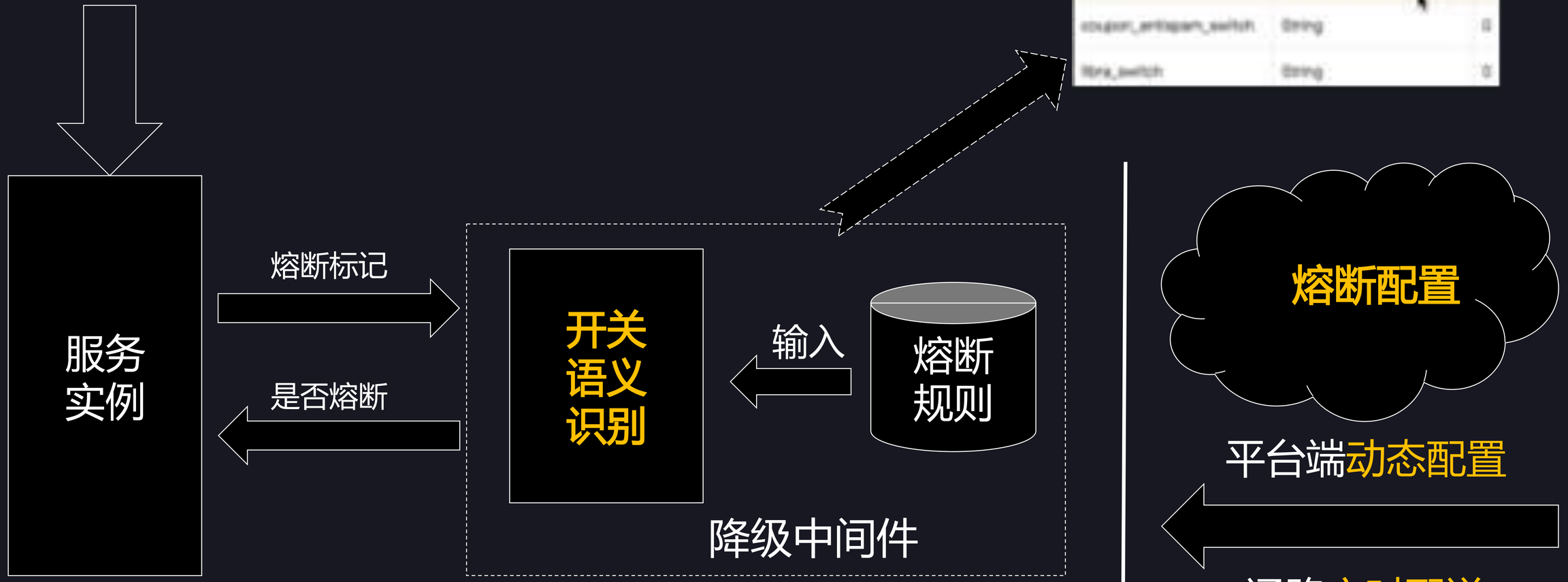


TABLE OF

CONTENTS 大纲

- 滴滴出行的业务架构
- 高可用套路
- 异地多活
- 一键降级
- 防火放火

防火灭火放火的重要项目

降低不可用发生概率：
- 线下仿真
- 灰度发布

防火

缩短止损时间：
- 异地多活
- 一键降级

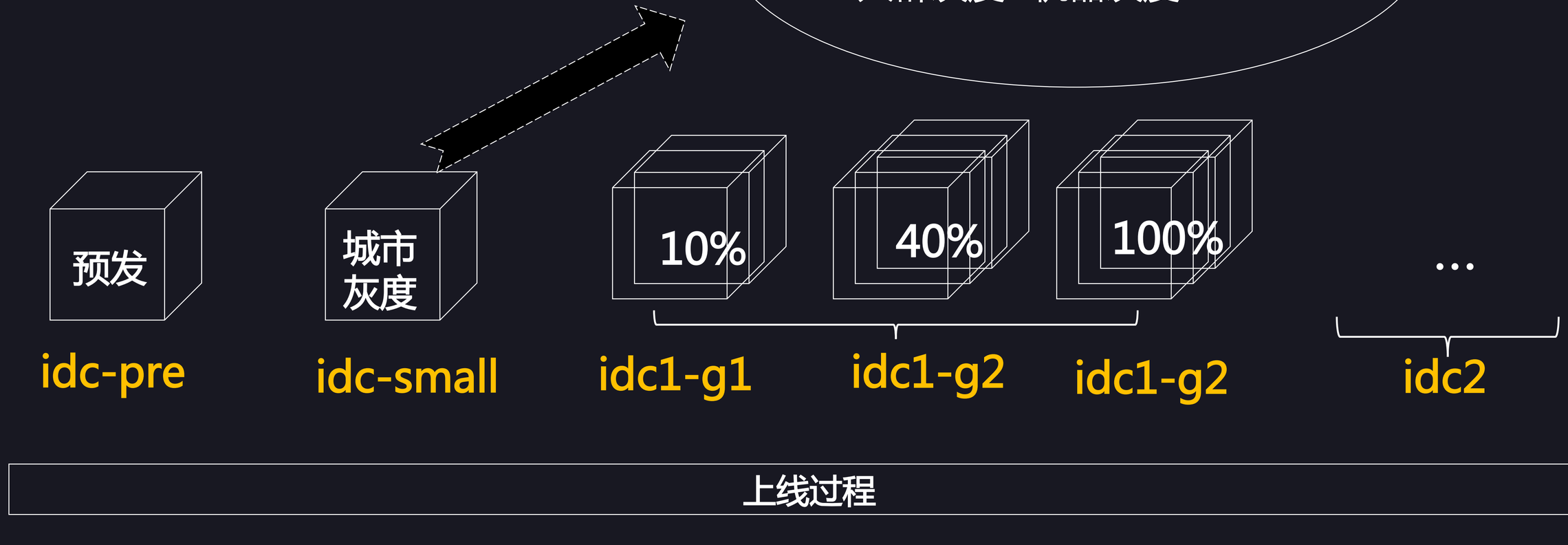
灭火

验证灭火是否有效完备：
- 故障注入
- 压测

放火

防火-灰度发布

人群灰度？X 开关维护成本高
机器灰度？X 指标不聚焦不敏感
So 人群灰度+机器灰度



放火-压测

哨兵系统
- 小规模损失风险换取*及时预警*
- 物理*隔离*
- *流量大于*正常集群
- *动态调控*

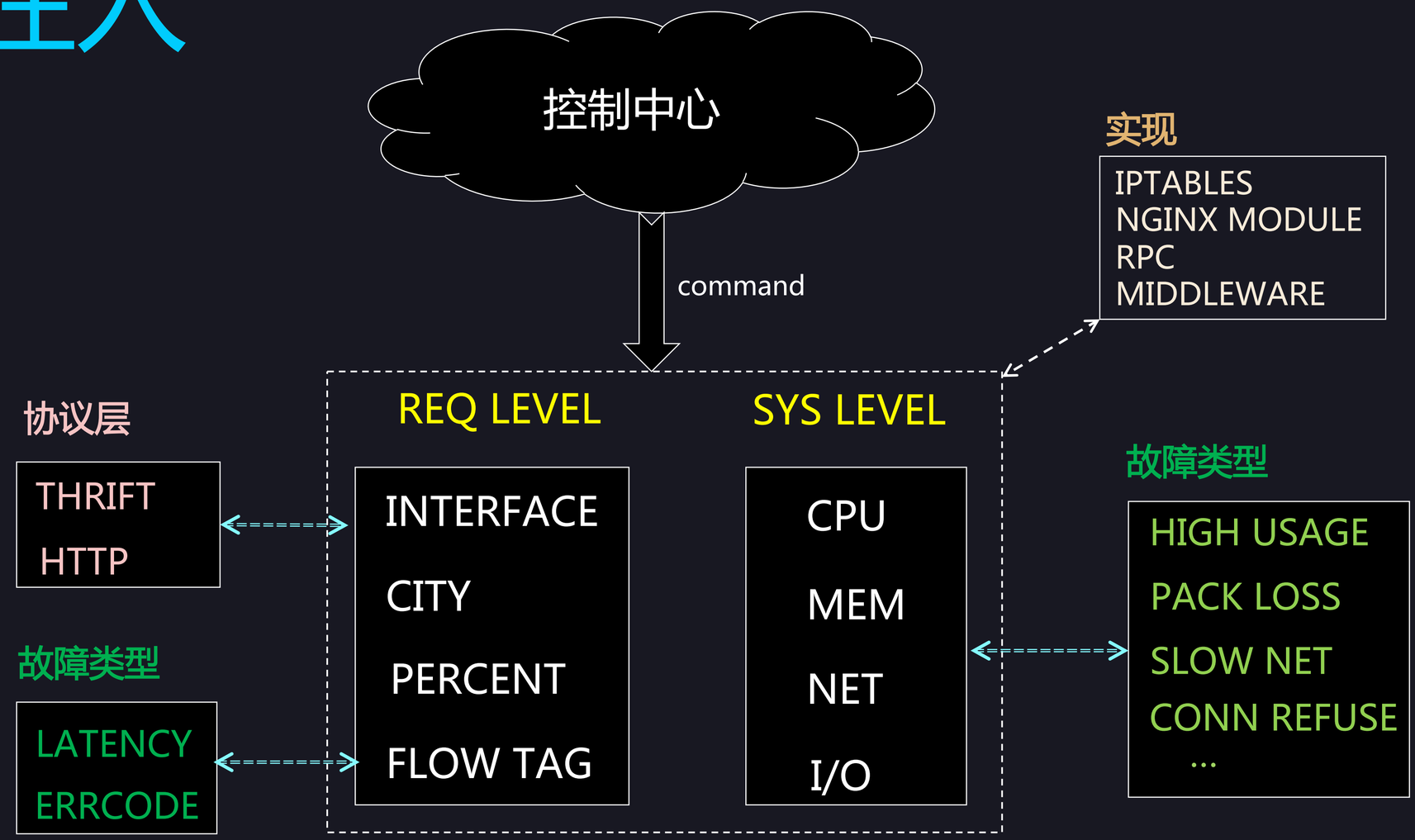
全链路压测
- *仿真*司乘行为
- *透传*压测标识
- *隔离*压测数据

单链路压测
- 注重*子系统*压测
- *隔离*压测数据
- *构造*上游请求
- *Mock*下游结果

放火-故障注入

目标
- 预案完备性检查
- 强弱依赖验证
- 提升异常分支覆盖率

层次
- 线下环境
- 线上测试账号
- 线上单个城市

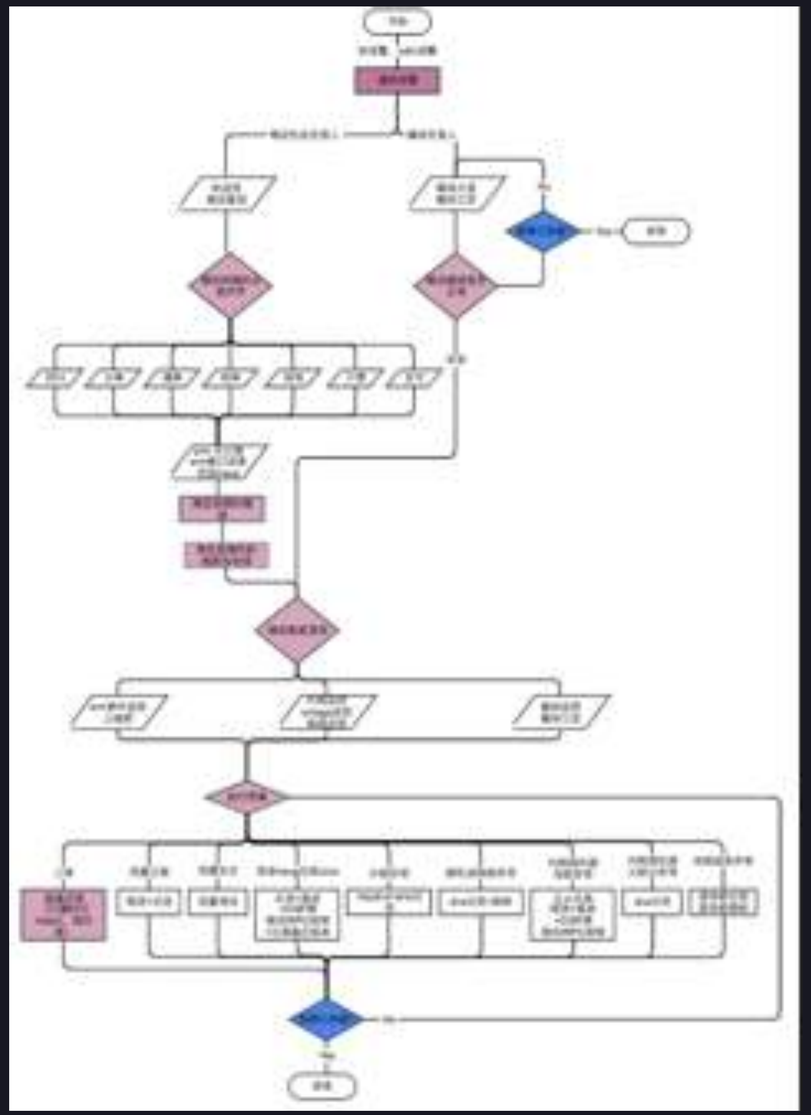


高可用落地

组织结构支撑
- ***公正的第三方组织（星辰花）：***复盘、定级追责、Trace 进展
- ***专项FT：***虚线汇报、项目经理协助推动

狠抓细节
- QA***验证***
- 8大抓手***竞赛***形式推进：如定位止损流程图

收益
- 可用性达到***99.95%***
- 以前可用性不到99.9%



高可用实践小结

- 业务演进： 流量*增长迅猛*、业务*复杂*、5年内已*迭代4次*
- 高可用方法论： *8大*抓手，终极目标*43210*
- 异地多活：*分层单元化*路由、数据*同步*、业务*双写*、降级*预案*
- 一键降级： *场景*、预案*灰度*、生效、*切流*、*限流*、*熔断*
- 防火放火： *城市灰度*发布、多样化*压测*、*三级放火*
- 特色实践：*组织结构*支撑、狠抓*细节*

THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

ArchSummit
全 球 架 构 师 峰 会 2017