

三峡银行云原生底座助力数字化转型

肖晓锋

腾讯云商业银行业务中心

腾讯金融云落地典型案例

国有大行
金融云



建设银行采用腾讯TCE搭建首家金融行业云，目前为国内最大，并持续增加新的能力，超40000节点

国有大行分布
式核心研发



大行核心下移的先驱项目。在全套腾讯云分布式技术架构支撑下，完整承接中行分布式核心下移，承载腾讯产品能力的最佳实践

超大规模集团
云建设



集团数字化转型，构建生态云新模式，集团核心业务云化+金融科技输出

全面数字化转型



全面“中台”战略，打造业务中台、数据中台、AI中台，实现业务敏捷

数字化转型及
互联网核心
分布式

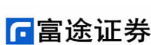


打造“线上化、数据化、智能化、平台化”的智慧新金融，数字广农商

首家全面核心
分布式



包括分布式核心、分布式信贷、分布式渠道、集中作业、风控系统在内的108套业务系统全面上云



目录

- 云原生在金融行业发展趋势
- 云原生转型技术挑战与应对策略
- 应用迁移与改造策略
- 数据库资源建设
- 基础资源层供给建设
- 云原生高可用体系建设

云原生金融行业发展趋势

现有银行关键业务系统所遇到的挑战

目前现有金融核心架构已经不适合新的业务发展，如何更好支撑业务创新是新一代关键业务系统要考虑的发力点：

挑战一：现有系统的业务核心架构设计已经遇到性能瓶颈，尤其过度依赖数据库的性能提高业务吞吐量，严重依靠**物理硬件性能**提升整体性能已经不能满足业务发展

挑战二：新的业务周期变化段，要求平台支撑能力能够快速响应，目前系统由于设计上的**紧耦合**，不能快速响应

挑战三：原有系统目前不能够支撑标准定制化的**配置**，从而快速构建前端业务的变化和微调

挑战四：新的业务模式要求有灵活的定价体系，针对不同群体能够实现**定制化的服务**

挑战五：目前系统的设计不能够解耦，造成不同系统扩展慢，**重复开发**量大，对用户体系管理混乱，降低用户体验

挑战六：系统设计的思路**不是以客户为中心**，多个系统交织在一起，对用户行为分析等带来瓶颈

挑战七：系统设计过度强调**集中性**，非业务相关逻辑和功能不能有效的剥离，造成系统出现问题定位困难

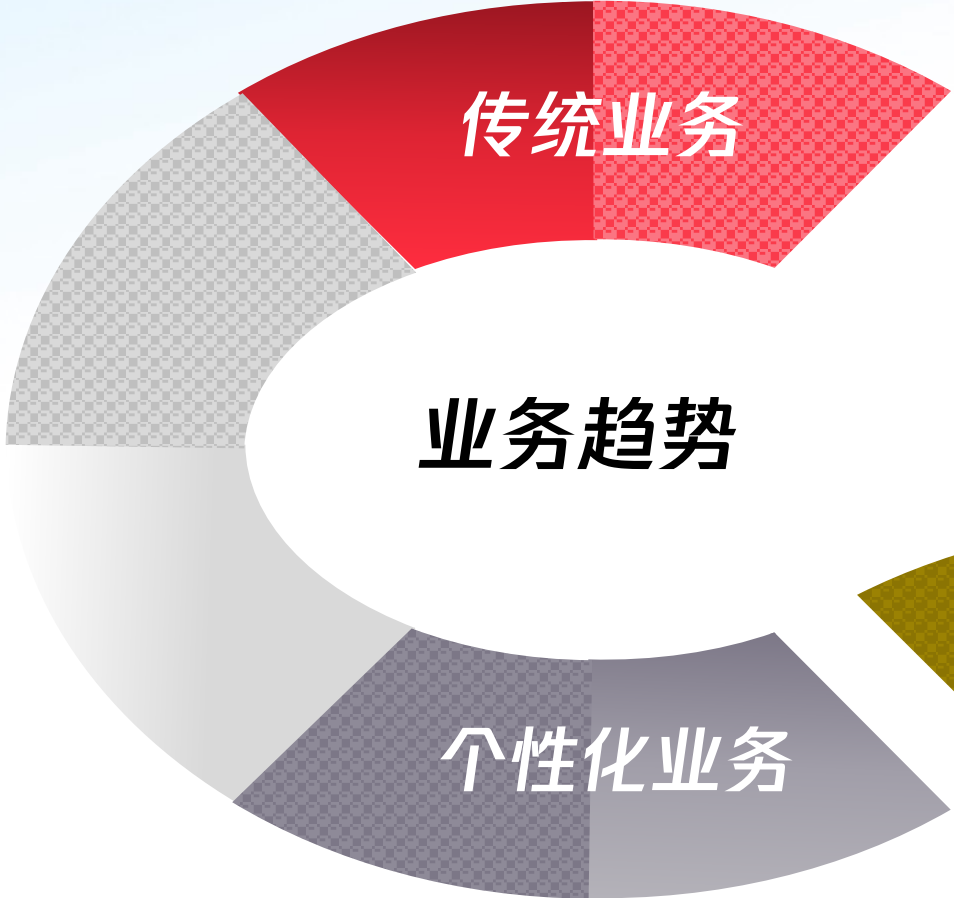
挑战八：**运维成本高**，由于设计的问题，造成运维过程经常要直接操作核心数据库，解决问题周期长

挑战九：无法快速扩展业务的**运营能力**，运营支撑需要依托核心系统的开放能力



业务趋势+技术发展对IT系统提出了新的挑战

- 传统业务的业务发展进入瓶颈
- 传统业务的集中式架构已经不适合业务创新要求
- 传统业务性能出现瓶颈

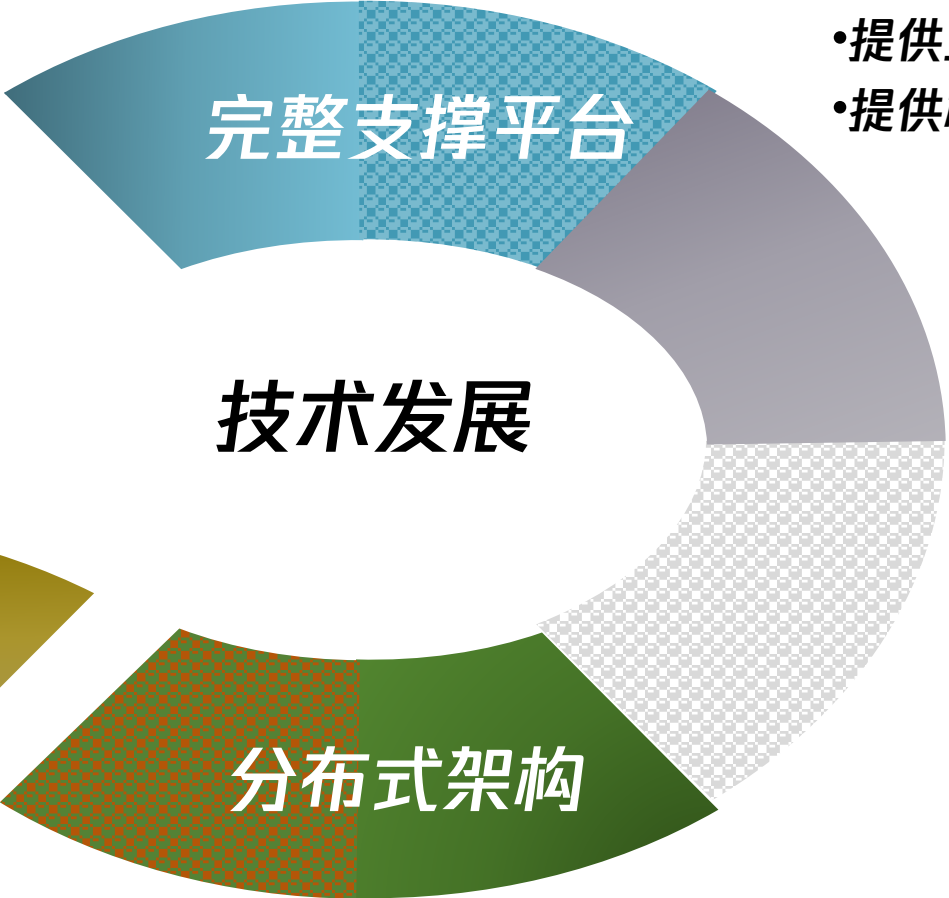


- 个性化服务是未来金融发力点
- 个性化服务提高金融服务水平
- 产生差异性优势的基础
- 快速对接终端用户需求

- 自动化减少运维成本
- 快速定位解决问题
- 平台稳定运行基础



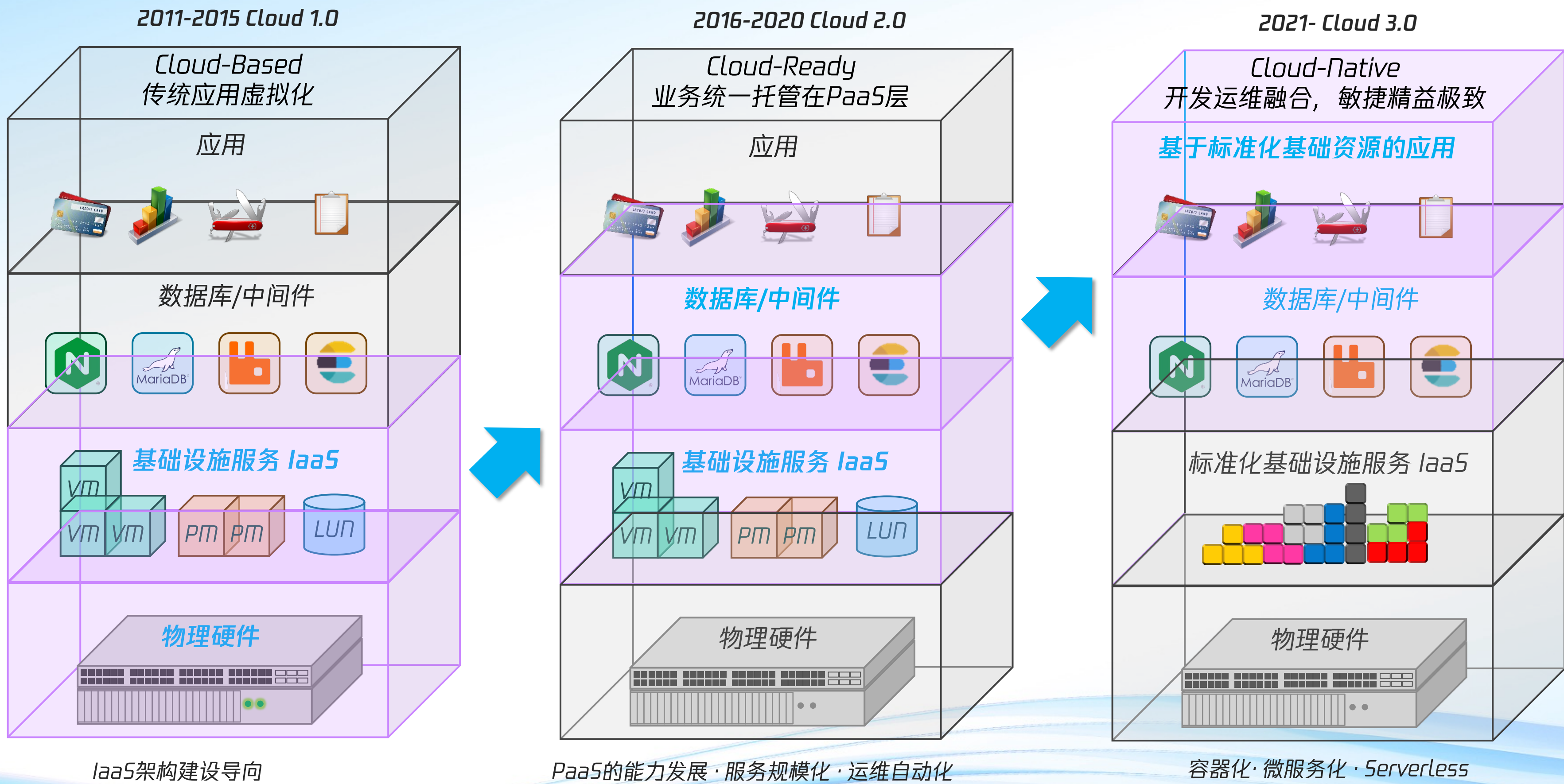
- 互联网用户无缝接轨
- 提高响应速度和时间
- 降低系统投资成本提高管理效率



- 提供基础资源支持服务
- 提供平台资源支持服务
- 提供业务组件支持服务
- 提供标准业务接口服务

- 分布式架构解决当前系统瓶颈
- 更好的高效资源利用
- 更好的结偶系统的复杂性设计

云架构规划从面向资源到面向业务应用转型



更聚焦于上层业务的发展，从稳态到敏态来说，业务需要有更好的抽象

四个基于的分布式核心架构设计理念

基于内聚设计的组件化架构

- 各个组件依据功能进行内聚化设计，彼此通过接口调用、协作

基于关注点分离的分层架构

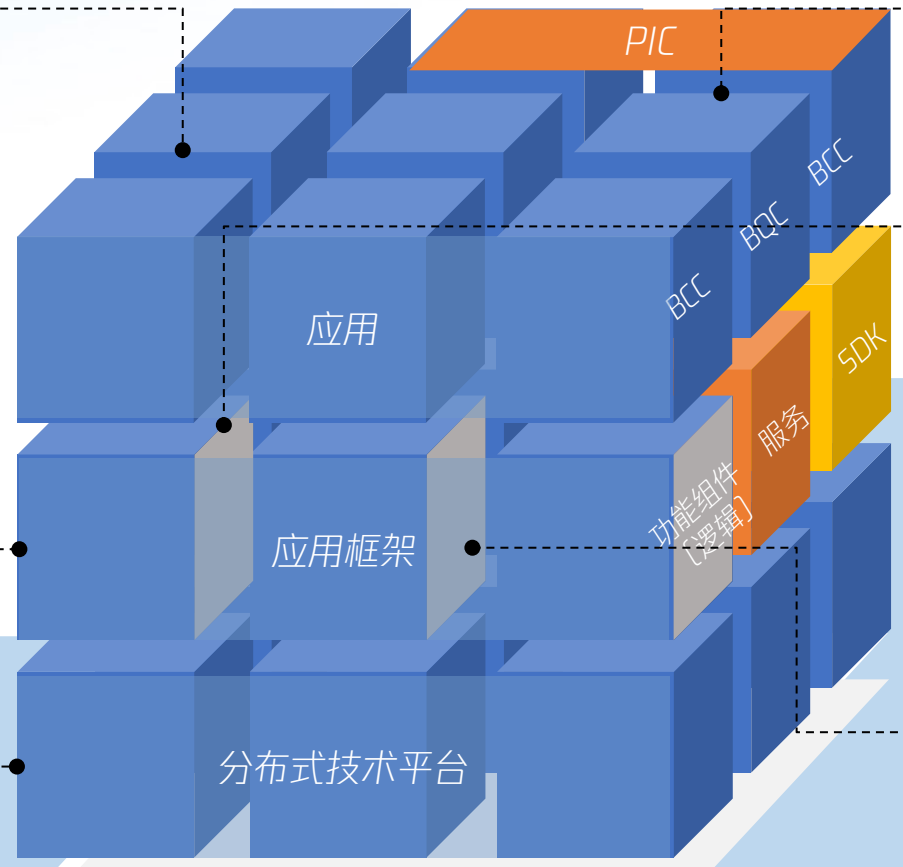
- 技术平台层负责提供与单元化、分布式相适应应用运行环境
- 应用框架层负责提供公共技术服务，方便和规范应用的开发
- 应用层负责具体的业务功能的实现

基于解耦设计的功能

- 应用框架
 - ✓ 服务类组件独立部署和运行，提供服务端点供外部访问其功能
 - ✓ 功能类组件明晰功能边界，实现公共机制和功能，在各个组件有独立实现但统一设计
 - ✓ SDK类组件独立封装，灵活依赖，其余组件可以根据需要进行引入
- 应用构件
 - ✓ BCC：负责对数据进行增删改
 - ✓ BQC：负责对数据进行查询
 - ✓ PIC：负责组装BCC和BQC以实现领域数据聚合，完成特定的业务功能

基于单元化的设计和部署策略

- 根据架构功能的不同定位，归纳设计不同类型的标准化单元，针对单元定位强化功能和差异化其部署



面向云原生的商业银行整体架构



智能运维平台

运维监控

集中告警/展现组件

监控信息处理组件

运维大数据

运维大数据平台

场景模型

运维操作自动化

自动化服务组件

自动化作业组件

自动化巡检组件

安全体系

虚拟化安全

云组件安全

网络安全

主机安全

应用安全

行业先驱者已经在领跑云原生转型

部分头部案例示例

核心数据库

国有大行



500+
金融机构



全球首个主机下移业务，信用卡核心分系统分布式数据库



为关键账务系统提供稳定高性能的金融数据处理服务



首个传统银行核心系统数据库国产化分布式改造



全球首个全行上云标杆，22.5万笔/秒交易峰值

基础架构

新一代金融基础设施



多地多中心IaaS、PaaS、安全等全栈云平台建设，构建稳定、安全、高性能的新一代金融基础设施

超大规模生态云建设



集团数字化转型，构建生态云新模式，集团核心业务云化+金融科技输出

分布式技术应用行业标杆



腾讯云分布式PaaS平台、分布式数据库助力中行、农行核心业务的信创和分布式改造

核心系统国产化



为银行核心系统的国产化提供关键技术支撑，助力银行实现核心技术能力的自主可控

全面数字化转型范例



网络银行全面线上化快速构建风控能力
分布式云底座支撑双模IT管理能力

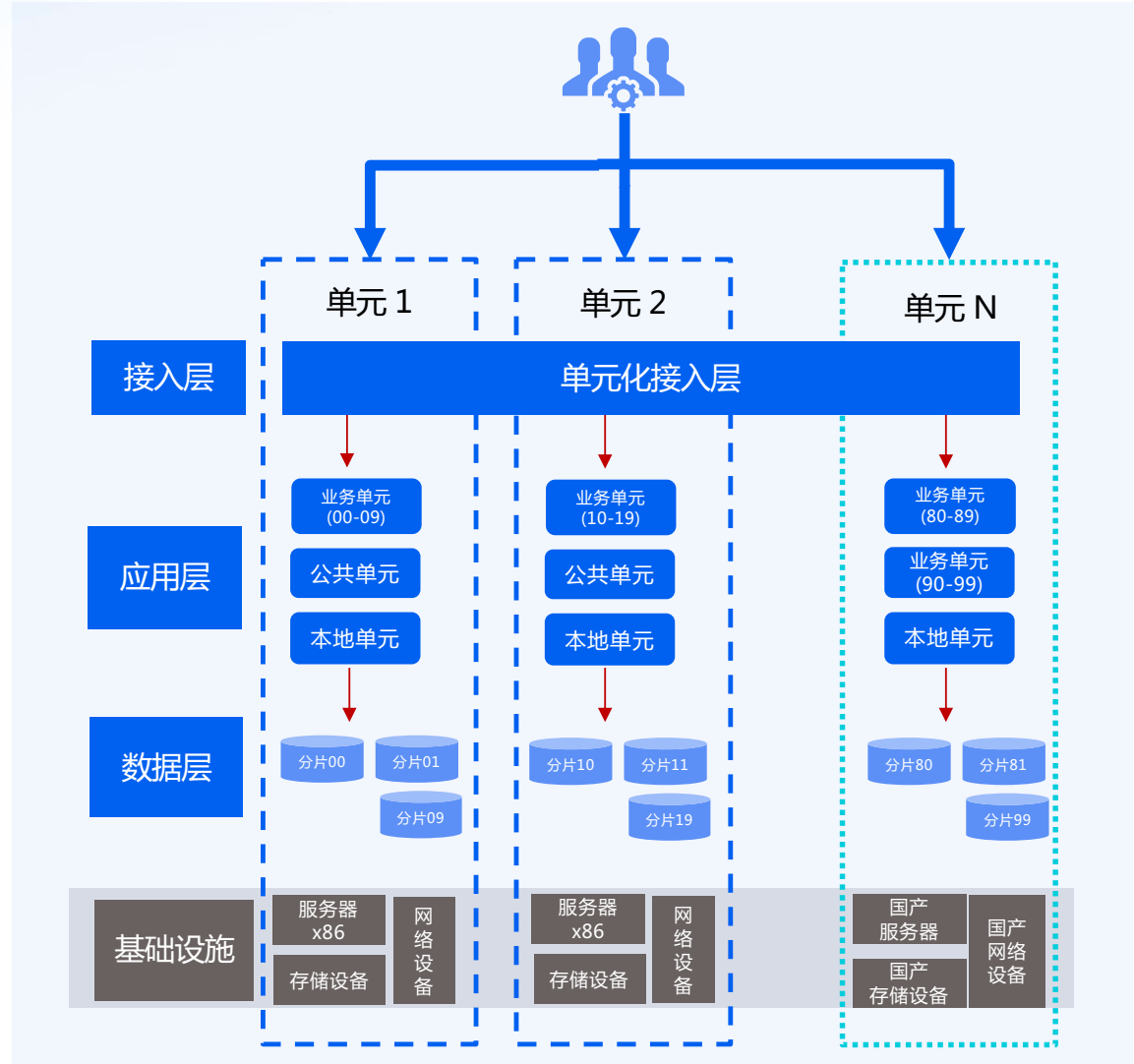
多地多活前沿云原生应用发展进程



业界领先的分布式云原生核心“技术底座”方案



新核心业务系统单元化、跨机房多活架构的落地实践



2家国有大行
核心投产上线



2家头部股份制银行核心
近20家区域银行核心



数据库: TOP20券商
50%+采用



头部保险集团国产数据库
寿险云原生核心系统上线

2023 行稳致远 为金融新核心发展贡献腾讯力量

标准化和单元化架构的技术策略

目标方案	数据切分策略	技术架构标准	微服务实现策略	业务连续性	研发效能	分布式运维
标准化方案	客户号哈希	机房内随机负载 优先同机房调用	以应用软件包内置 领域为服务颗粒度	同城双活+异地灾备 两地三中心	持续集成与发布	标准化指标采集
	分布式分片架构	应用+数据库 处理分布式事务	应用厂商的 交付实施工艺	实例级故障自恢复	快速反馈与迭代	大数据分析告警
	纯数据库扩容	数据库处理聚合查询	应用软件包内置的 技术组件	中心级故障告警+ 人工决策切换	双速开发+自动测试	可视化+自动化
	业务/公共/历史	微服务平台 实现灰度发布		地域级故障告警+ 人工决策切换	过程可视化+精益管理	配置与流程管理
单元化方案	客户号自定义路由	单元内随机负载	以业务建模产出 为服务颗粒度	近期： 两地三中心 远期： 异地多中心	持续集成与发布	标准化指标采集 大数据分析告警
	独立分片架构	自定义就近调用	咨询公司的 IT实施工艺	实例级故障自恢复 单元级故障自恢复	快速反馈与迭代	可视化+自动化
	自定义扩容	应用处理分布式事务	以外购+自研的形式 形成技术组件	中心级故障告警+ 人工决策切换	双速开发+自动测试	配置与流程管理
	业务/公共/历史	全局路由+微服务平台 实现灰度单元		地域级故障告警+ 人工决策切换	过程可视化+精益管理	单元化运维管理

腾讯云平台XC交付案例

监管及旗下机构

央行、证监会

央行云[两地三中心，一云多芯]
央行信创实验室[海光服务器]

证监会监管云[海光服务器]

金融各细分行业公司

国有大行

建行云：金融第一个全栈信创行业云，3朵云多个AZ，鲲鹏、飞腾、海光等10+主流服务器

中行：多地多中心，主要PaaS产品承载核心系统

商业银行

银联云：全量IaaS/PaaS产品[40+]、4Region6AZ一云多芯
上农商：x86/ARM一云多芯
广农商：一云统管，独立信创Region
招行：一期3AZ信创云，海光、鲲鹏服务器

保险

PICC：两地四中心，2020年率先建成信创存储资源池

民生保险：一云统管，独立信创AZ

资管

方正证券：x86/海光/鲲鹏，一云多芯

深交所行业云：多Region，一云统管，x86、海光、鲲鹏多种机型

中金证券：一云多芯

案例多、工艺成熟

已实现规模化自动化交付，落地监管机构、各金融细分行业20+客户及30+AZ

多元架构、按需选择

架构灵活：可支持信创Region、信创AZ、AZ内一云多芯，适应多种需求场景

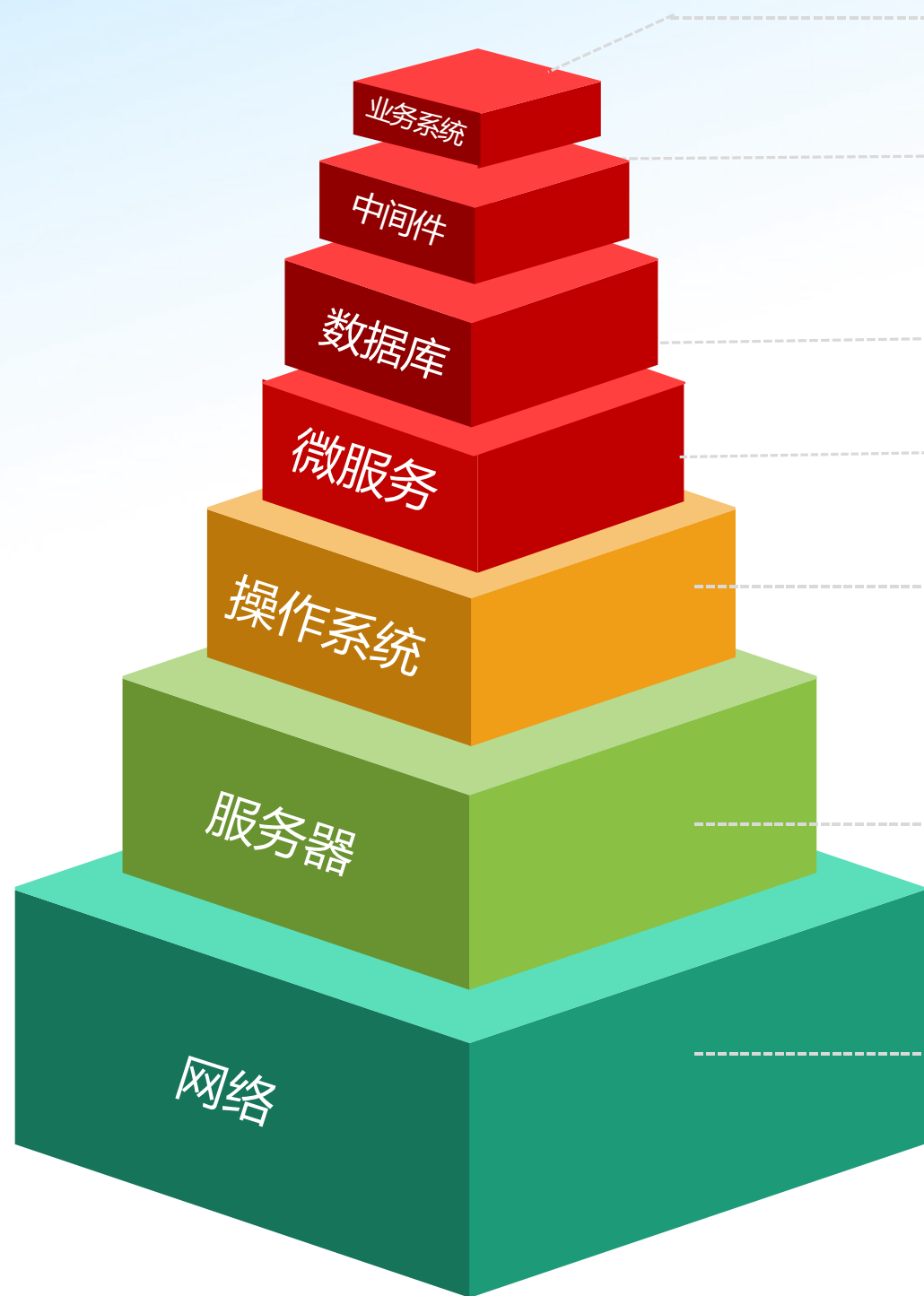
芯片齐全

已适配鲲鹏、海光、飞腾等芯片，10+主流机型交付投产，可支持更多芯片适配兼容

全栈产品兼容

计算、网络、存储、数据库、安全，IPv6、容器编排、容灾管理、国产操作系统[UOS、麒麟]

腾讯云对XC改造的理解



现状

- 完全自研
- 无 IOE/商业软件，自研为主，开源补充
- 采用Oracle/DB2为主
- 采用开源为主
- 采用RedHat, Centos, SUSE等
- 采用多种品牌的 X86 架构服务器
- 采用多种品牌非国产化芯片网络设备



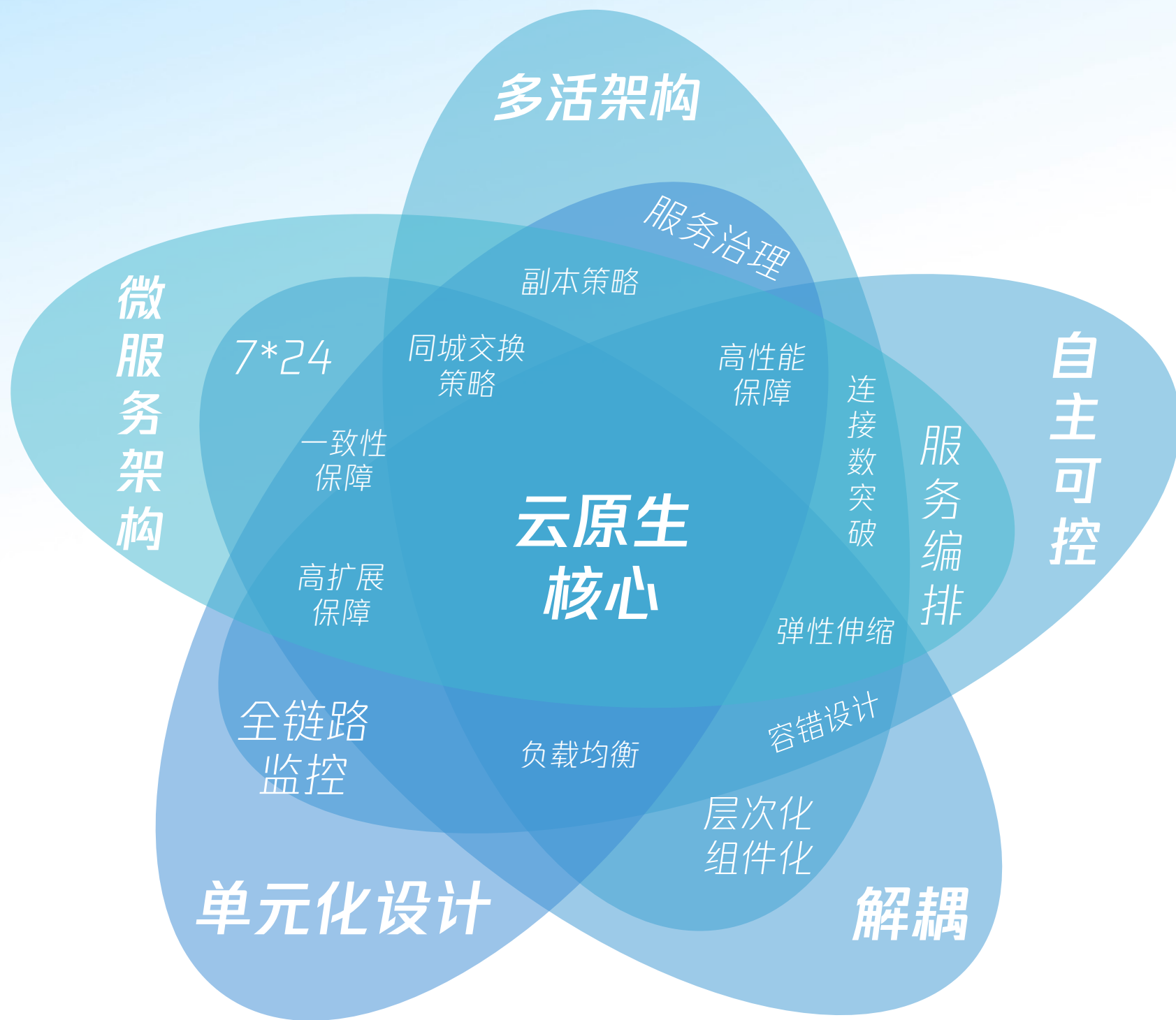
信创试点

- 业务系统适配国产化架构
- 公共组件适配国产化架构
- 国产数据库，如TDSQL, Oceanbase, GaussDB等
- 采用商用国产软件替代
- 适配国产化CPU的 TencentOS, EulerOS
- 采用华为ARM / 海光X86双路线进行适配
- 采用华为 / 华三双路线国产芯片交换机进行适配

云原生转型技术挑战与应对策略

云原生银行核心业务系统建设遇到的困难与挑战

15



API 接入

- 统一全行报文标准、提供分布式路由及链路跟踪支持
- 接口规范化设计，支持新旧应用协同化的服务治理与灰度发布
- 保障接口安全性、统一认证、高性能等保障

微服务设计

- 设计为与业务领域一致、松散耦合、用途单一的服务
- 建立微服务架构体系与设计思想，识别技术关键点，建立分布式技术平台，统一分布式技术标准与规范，降低应用开发人员技术难度，帮助解决分布式环境下的开发运行问题
- 建立全行级的服务设计模式，形成银行分布式应用生态，助力全行数字化转型

研发效能

- 从需求、设计到研发落地、最后生产上线形成敏捷协同体系
- 建立端到端DevOps流程和平台工具
- 从组织架构到微服务组件落地，支持服务独立演进

运维治理

- 治理策略制定〔熔断、降级、限流等〕以确保业务一致性、连续性
- 建立全链路监控、告警结合日志聚合分析实现快速故障定位
- 从快速定位修复到快速隔离恢复的运维转变，实现故障改进与预防

云部署

- 支持横向扩容、存算分离架构、能支持多种PaaS平台，支持对底层隔离
- 支持无状态服务、关键接口幂等能力，按需实现容器化部署
- 可按服务重要程度与技术形态等执行不同的高可用/集群策略
- 采用分层设计架构，组件化设计方法，要求可灵活拆分或组合发布

分布式核心六大设计要点

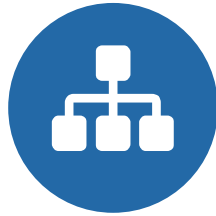
1. 数据切分策略

- 切分维度
- 分布规则
- 扩容策略
- 容量评估



2. 技术架构策略

- 交易处理策略
- 分布式事务策略
 - 扩容策略
- 聚合查询策略
- 灰度发布策略



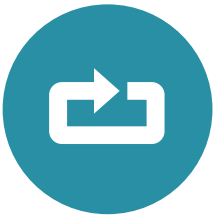
3. 服务模型策略

- 服务粒度
- 落地工艺
- 技术封装



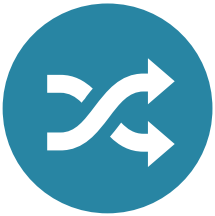
4. 业务连续性

- 弹性故障恢复
- 多中心多活
- 跨地域容灾



5. 研发效能

- 双速开发模式
- 全链路研发效能
- 敏态稳态分离



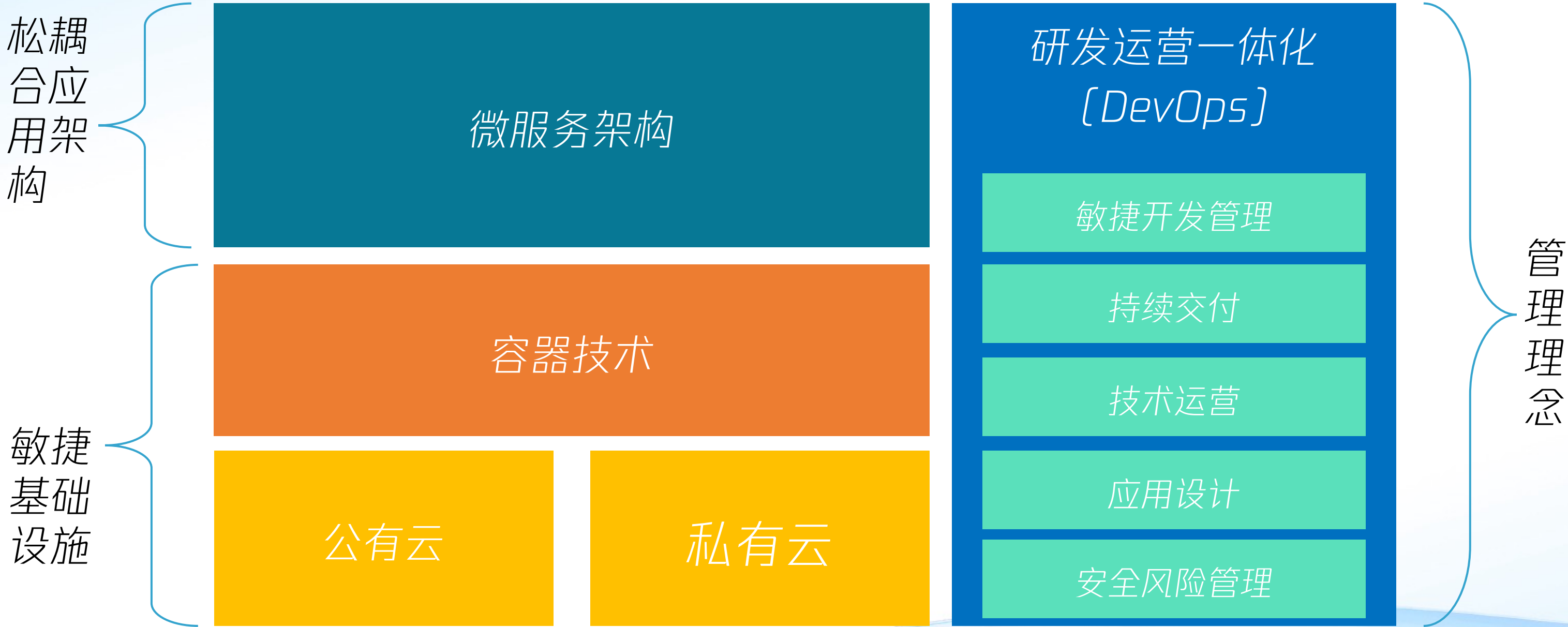
6. 分布式运维

- 标准化指标采集与分析
- 多维度监控与告警
- 智能自动化运维



商业银行拥抱云原生转型

随着云原生架构的普及，市场上对于容器技术的需求越发旺盛，相对IaaS云，容器云能够提供更敏捷的基础设施，也打通了开发和运维之间的界限。



云原生平台能力需求



腾讯云原生平台能力覆盖



分布式服务架构下的挑战

需求爆炸式增长，快速迭代，弹性扩缩的业务场景，对IT架构提出新的挑战

敏捷发布

及时响应业务需求，对于特定用户进行版本发布
小版本灰度迭代，金丝雀发布

实时扩缩

基于访问量和请求耗时等业务参数，
进行实时弹性扩缩容

平滑过渡

基于微服务的spring cloud或服务mesh，
可以进行跨语言，跨系统的和已有业务对话

平台战略

开发的代码或者接口，都可以进行复用，统一标准
达到越来越快的效果



1、微服务架构-代码层演化

演进方式：自上而下
蜂窝化代码结构

2、Devops流水线

自动化开发运维体系

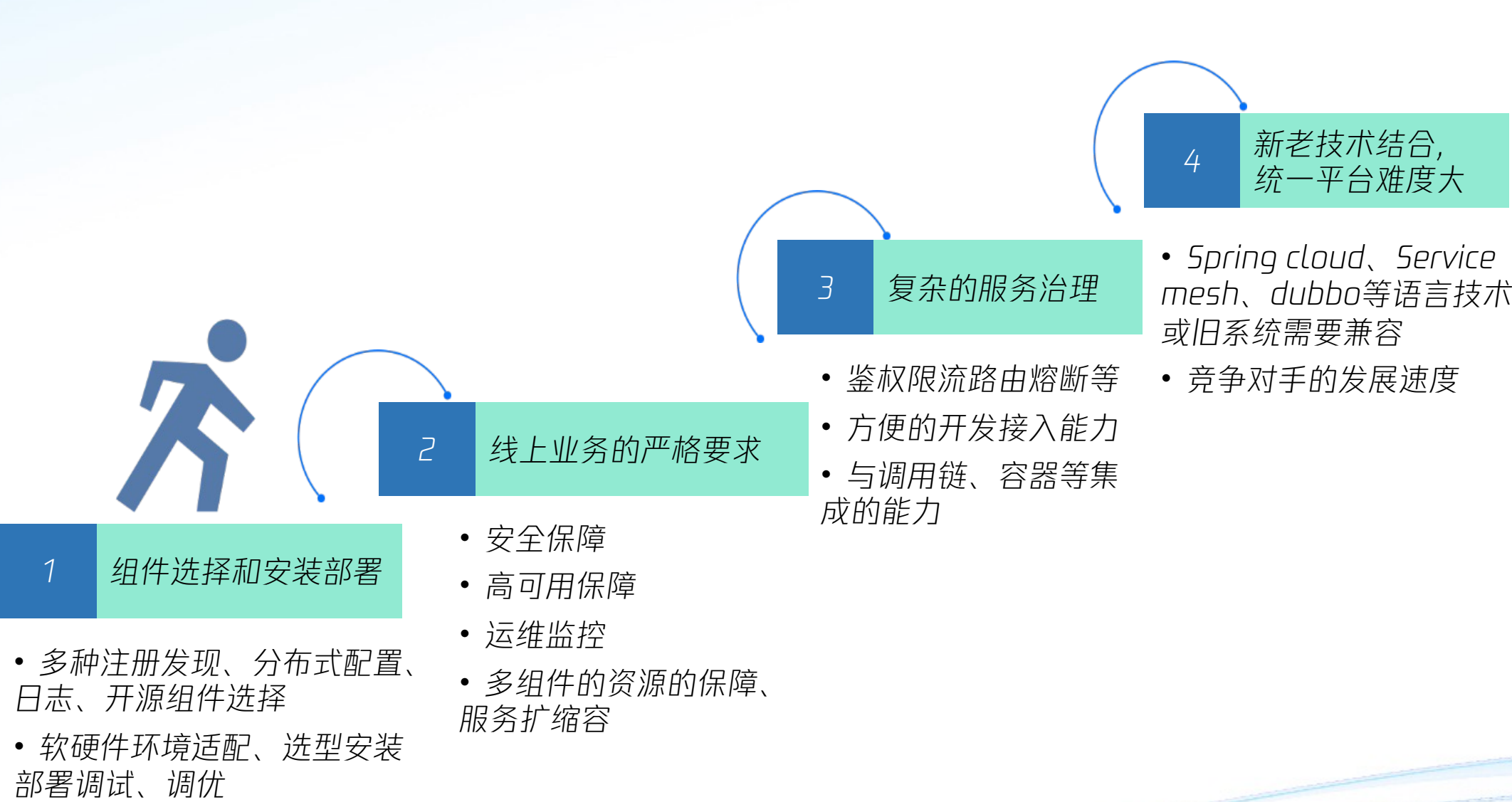
3、容器-资源层演化

演进方式：自下而上
蜂窝化计算资源

自建分布式平台挑战

自建微服务的平台挑战

需要长期的大量研发投入、时间、资源很大，学习成本高，影响业务的发展时机。



与云优势结合

将复杂的组件运维托管给云服务
享受微服务平台的便利



腾讯云一站式分布式平台



构建端到端DevOps流程体系

客户遇到的问题

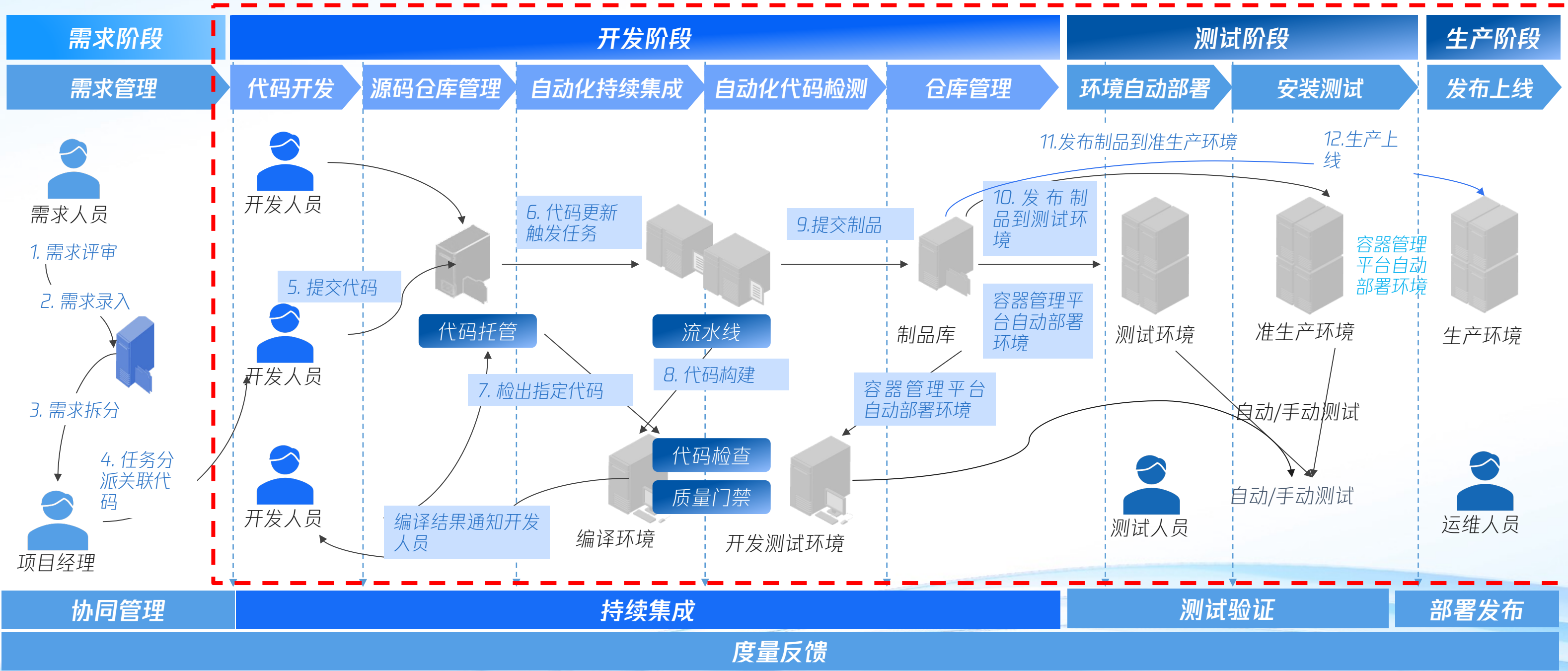
- 研发过程等待时间周期长
- 质量得不到保障
- 各个过程独立，无法跟踪溯源

与其他产品相辅相成

- 作为核心枢纽，打通串联各个产品，形成一体化相辅相成

产品给客户带来价值

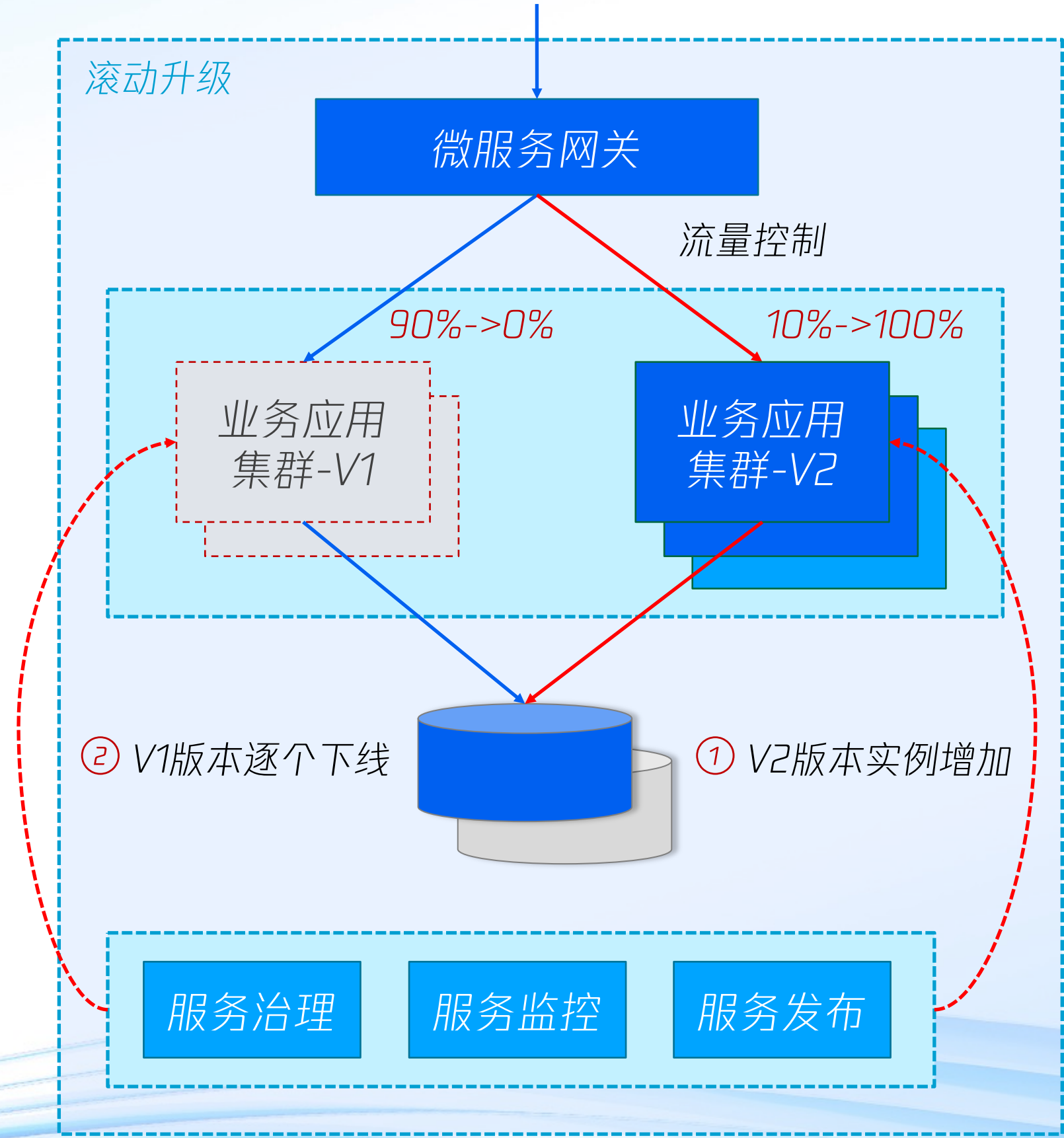
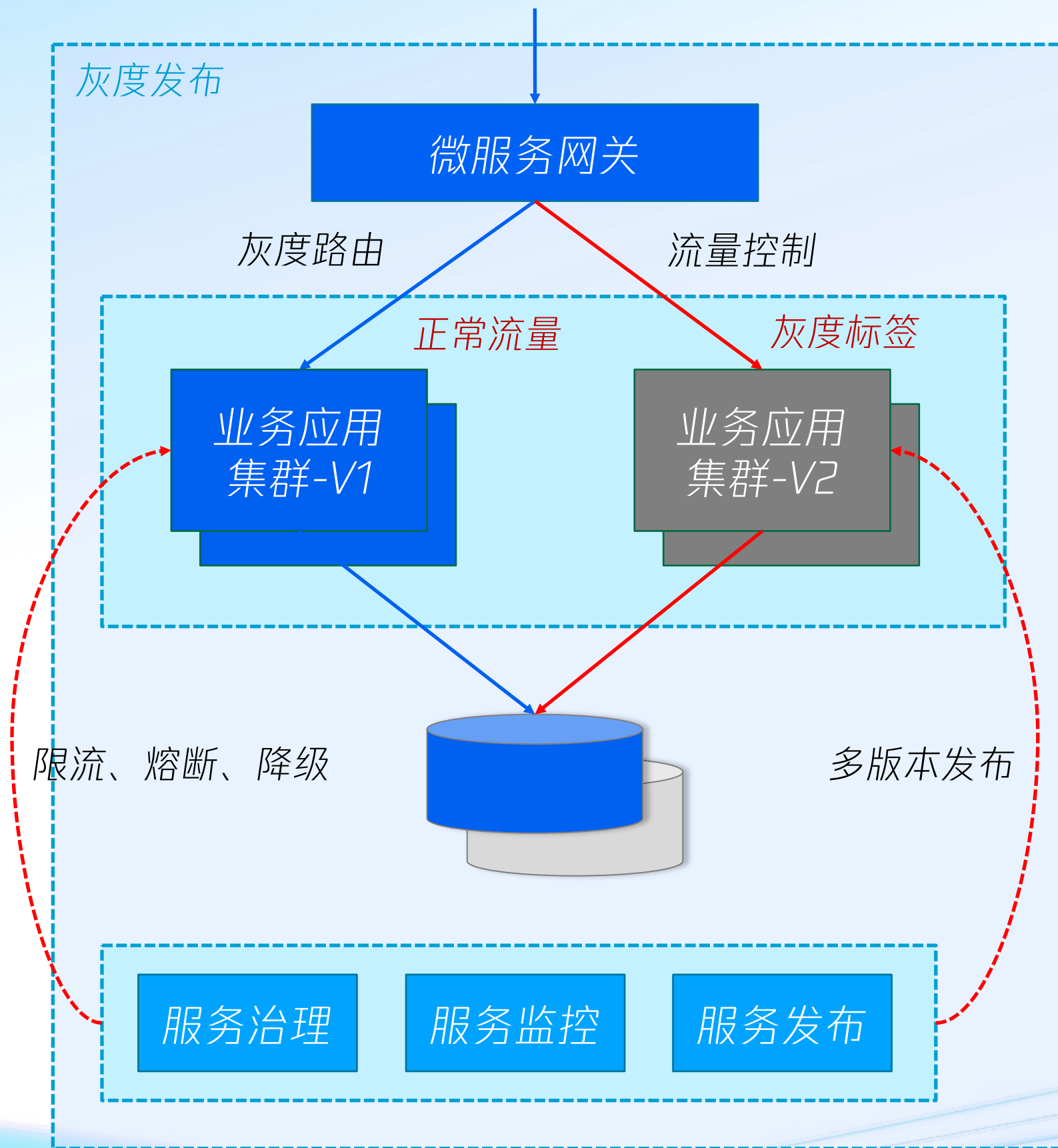
- 让局部效率转化为全局效率
- 让高效交付可以持续
- 让高效交付带来业务成功



以DevOps推动组织效能升级

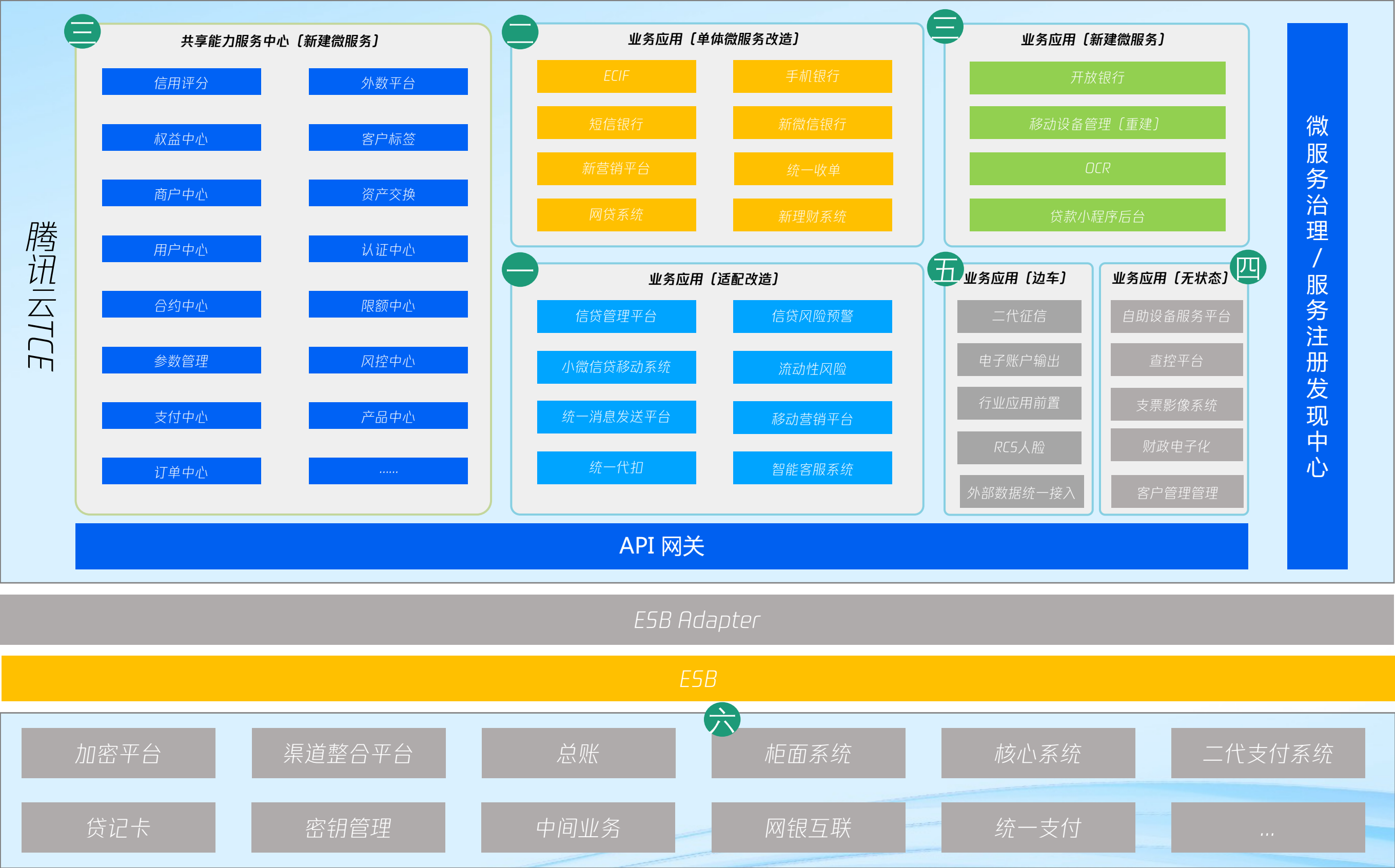


灰度发布与滚动升级方案



应用迁移与改造策略

示例：某银行应用全局整体部署规划



示例：应用迁移上云标准（非业务等级分类）

迁移方案

- 是否存量系统：是
- 开发框架：SpringCloud。
- 运行环境：Linux
- 开发语言：Java
- 容灾方式：集群
- 交易类型：渠道交易/普通交易

- 是否存量系统：是
- 开发框架：Spring/Springboot。
- 运行环境：Linux
- 开发语言：Java
- 容灾方式：集群/主备/单机
- 交易类型：渠道交易/普通交易

- 是否存量系统：否
- 开发框架：SpringCloud。
- 运行环境：Linux
- 开发语言：Java
- 容灾方式：集群
- 交易类型：渠道交易/普通交易

- 是否存量系统：是
- 开发框架：Spring/Springboot
- 运行环境：Linux
- 开发语言：Java
- 容灾方式：集群/主备
- 交易类型：渠道交易/普通交易

- 是否存量系统：是
- 开发框架：Spring/Springboot
- 运行环境：Linux
- 开发语言：Java
- 容灾方式：单机/主备
- 交易类型：渠道交易/普通交易

- 是否存量系统：是
- 开发框架：
- 运行环境：AIX
- 开发语言：C/Java
- 容灾方式：集群/主备/单机
- 交易类型：核心业务类

一类应用

已经做了微服务改造的应用

二类应用

渠道类业务，高并发的单体应用业务

三类应用

计划新建设开发的应用

新建微服务应用

单体应用微服务改造

微服务适配

单体应用无状态化改造

单体应用边车化改造

AIX类小机应用下移

四类应用

不重构，但是有集群化，双活部署需求的

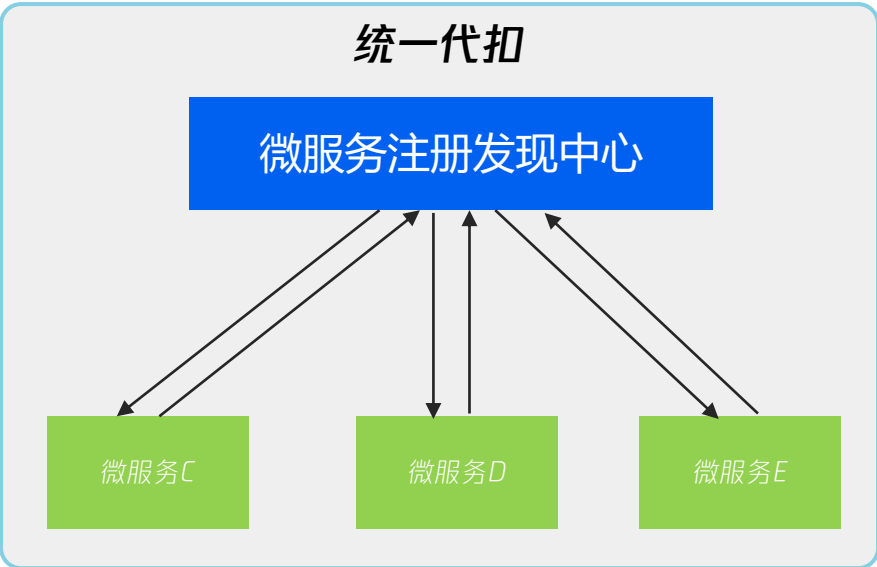
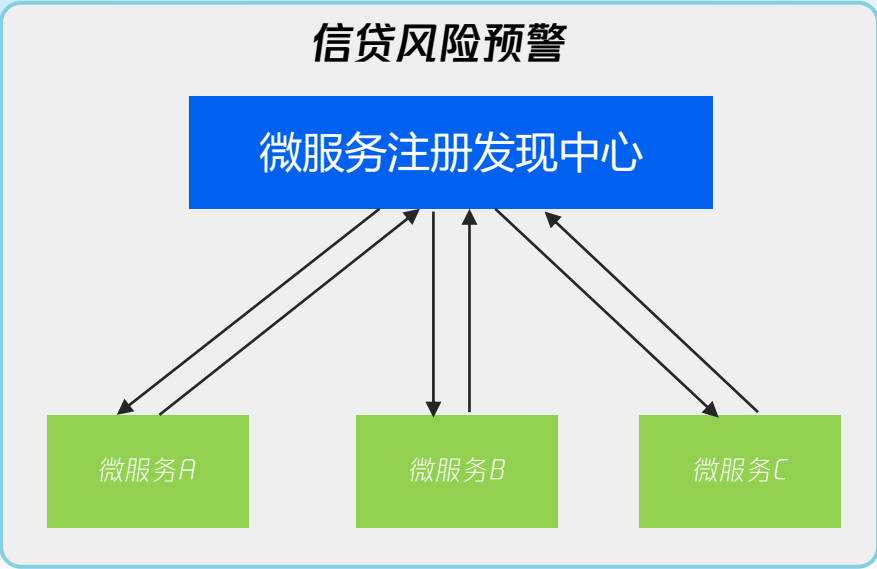
五类应用

不重构，需要接入服务治理体系的

六类应用

小机上的应用进行下移

一类应用：统一微服务开发治理框架



现状总结

1. 孤岛变多个群岛

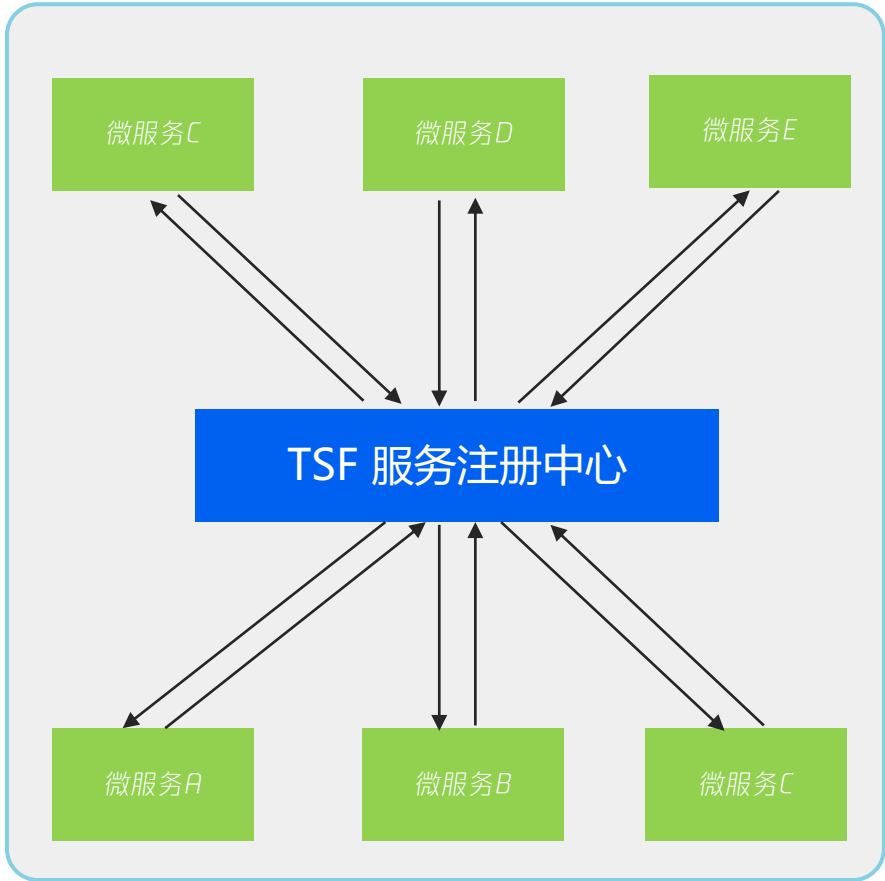
- 局部应用重构进行微服务建设。
- 初步的小范围内服务治理得以建立

2. 服务治理半集中状态

- 交易往往不是停留在单个“群岛内”
- 半集中状态的服务治理无法满足全局治理要求

3. 其它问题

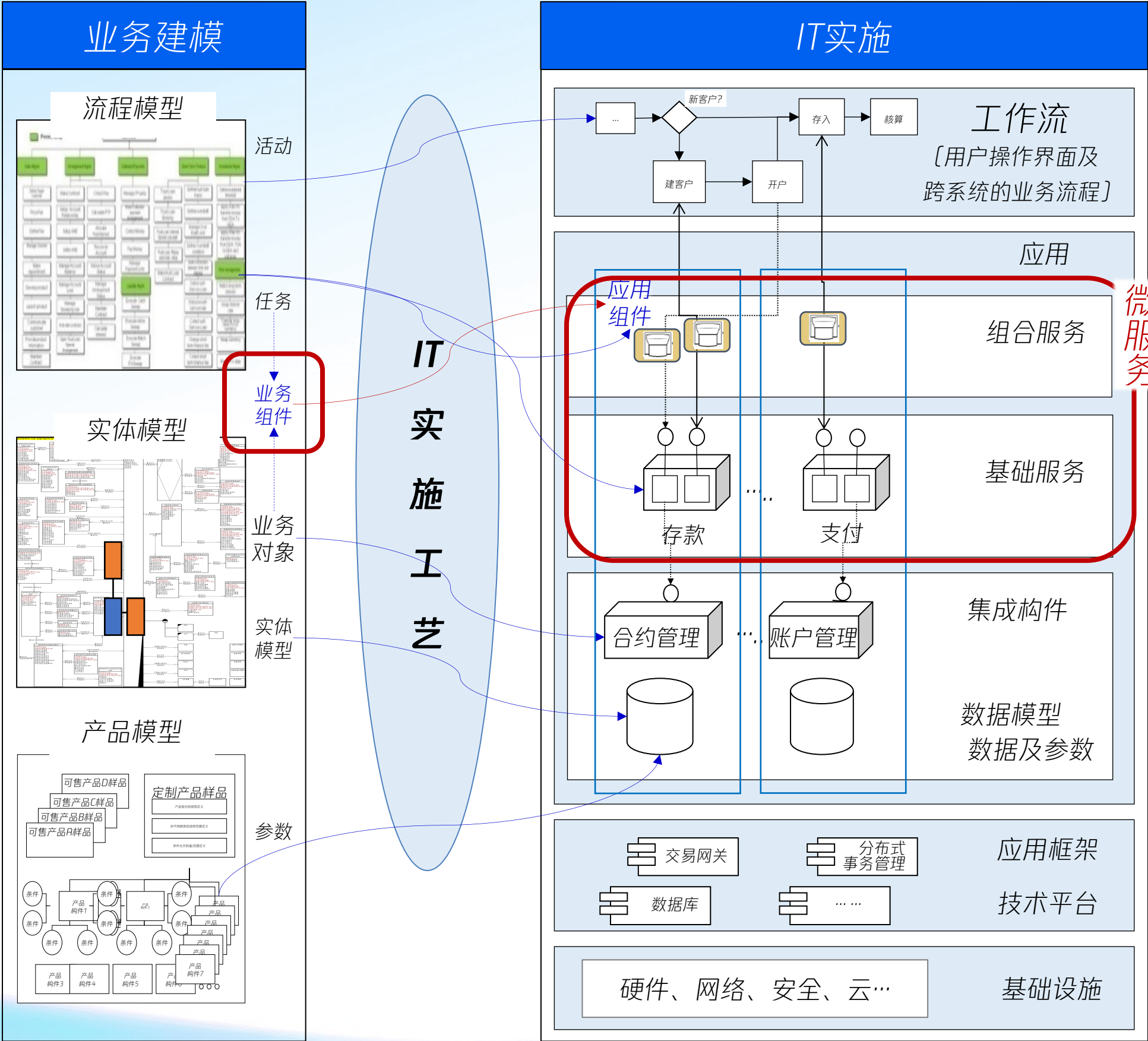
- 多版本微服务体系带来维护治理问题
- 全链路监控体系，依赖自身机制无法建立。



通过TSF统一基于Spring Cloud和Dubbo的开发框架，并通过TSF提供 API 级别的服务治理能力

支持控制台上进行配置服务路由、服务限流、服务鉴权规则在 TSF 控制台上，用户可以通过配置、权重标签的形式进行细粒度的流量控制，实现灰度发布、就近路由、流量限制、访问权限控制等功能。

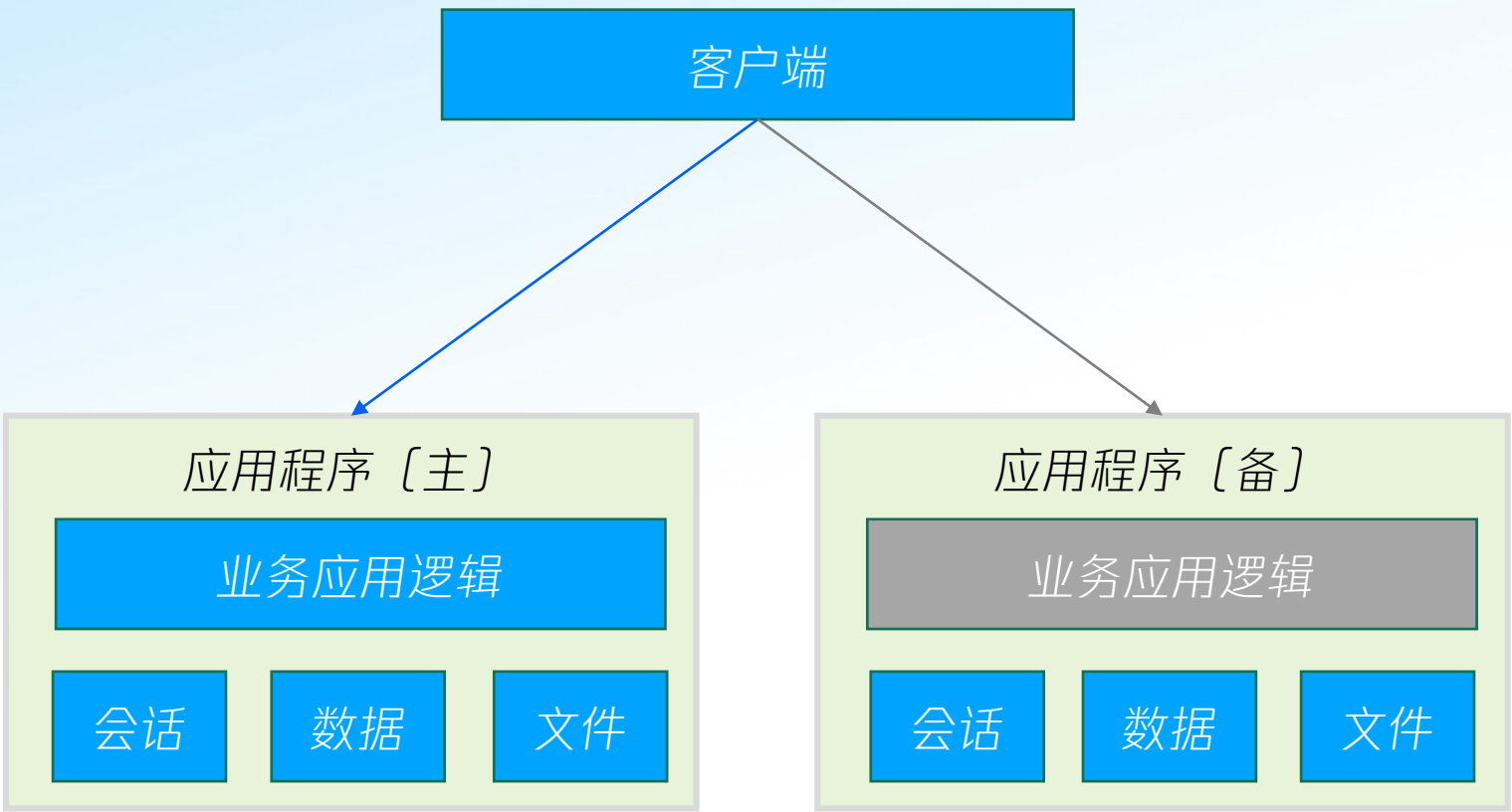
二、三类应用：新建/改造业务应用微服务化



- ## 建设方法 (Construction Method)
- 从业务需求和系统实现、端到端完整的、体系化的**分布式系统建设方法**:
 - ✓ 坚持开放平台、分布式架构的技术路线，通过上云，提高系统稳定性与灵活性，**架构具有前瞻性**
 - ✓ 以X86+信创服务器、开源技术为代表的技术体系，适应国家网络安全战略要求，**形成自主可控的企业级技术平台**
 - 业务模型驱动的微服务设计方法：新建系统采用微服务架构设计，具有以下优势：
 - ✓ 合理的微服务颗粒度划分，有利于组建敏捷开发团队，加上微服务良好的可复用性，**能够快速响应新业务需求**
 - ✓ 系统升级时，可在不停机的情况下发布新版本
 - ✓ 部分服务出现故障时，不影响其他服务的正常运行，**提出了系统整体可用性**

四类应用：暂不做微服务改造的单体应用进行无状态化改造

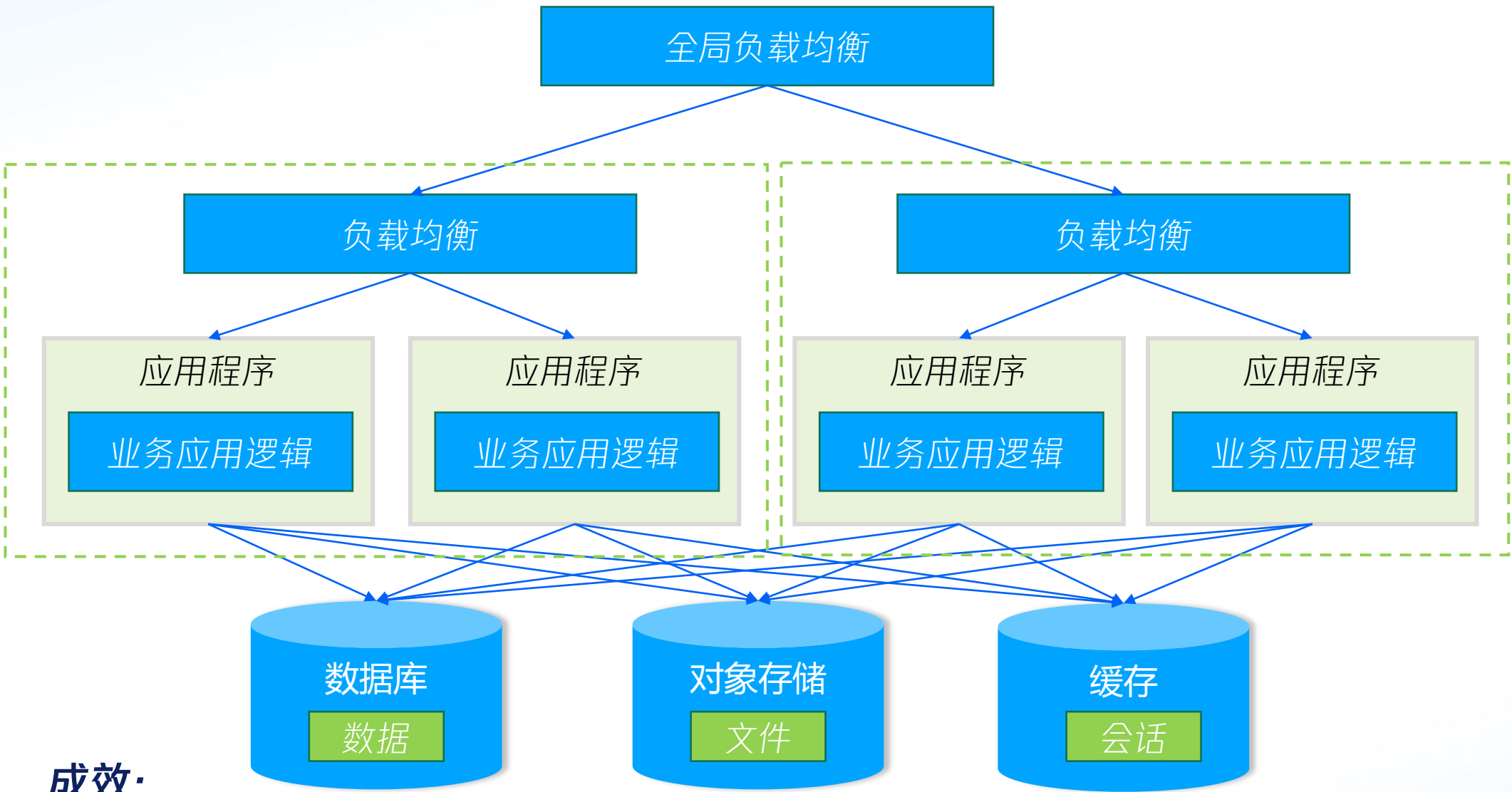
有状态应用



问题：

- ✓ 业务流量需要进行会话保持
- ✓ 业务连续性难以保障
- ✓ 缺乏横向扩展应对业务洪峰压力的能力
- ✓ 双活难以实现
- ✓ ROI只有50%

无状态应用



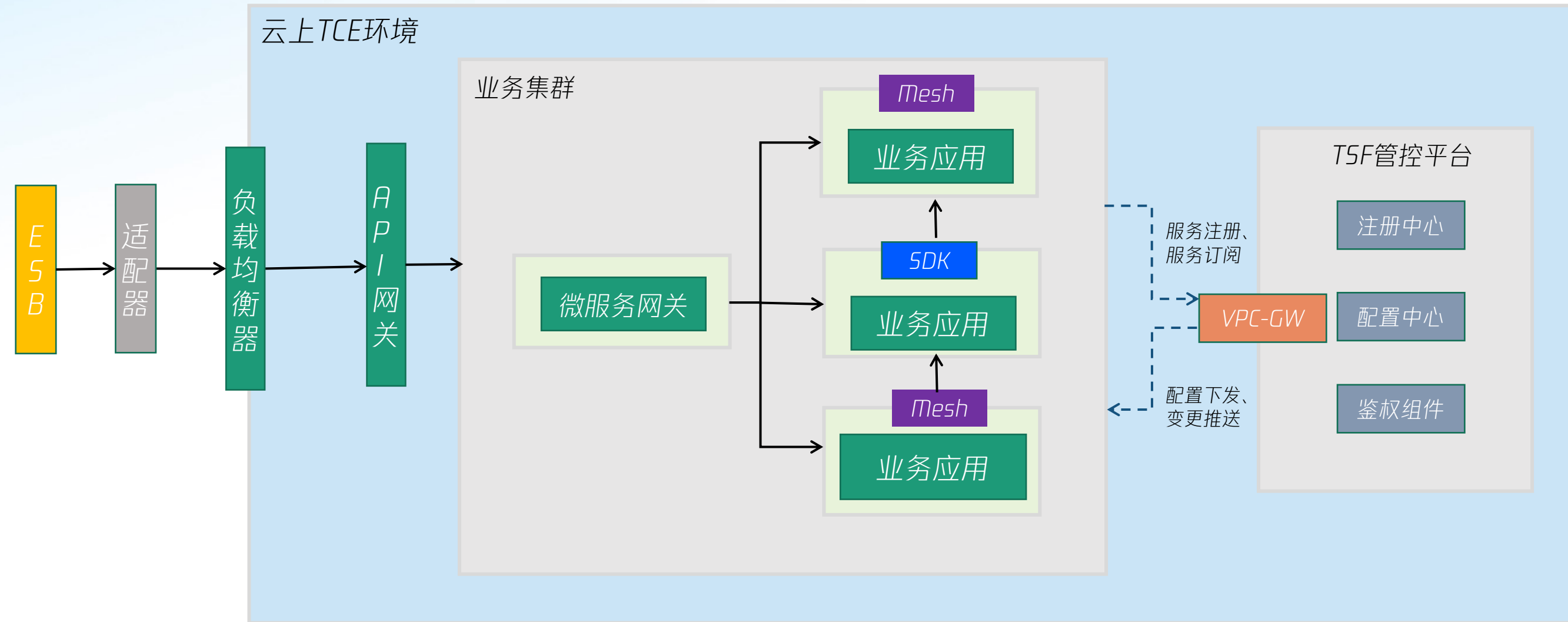
成效：

- ✓ 应用无需会话保持
- ✓ 应用任意扩缩容
- ✓ 跨中心高可用部署
- ✓ 松耦合架构，运维简单
- ✓ ROI可达100%

五类应用：单体应用迁移上云部署技术方案

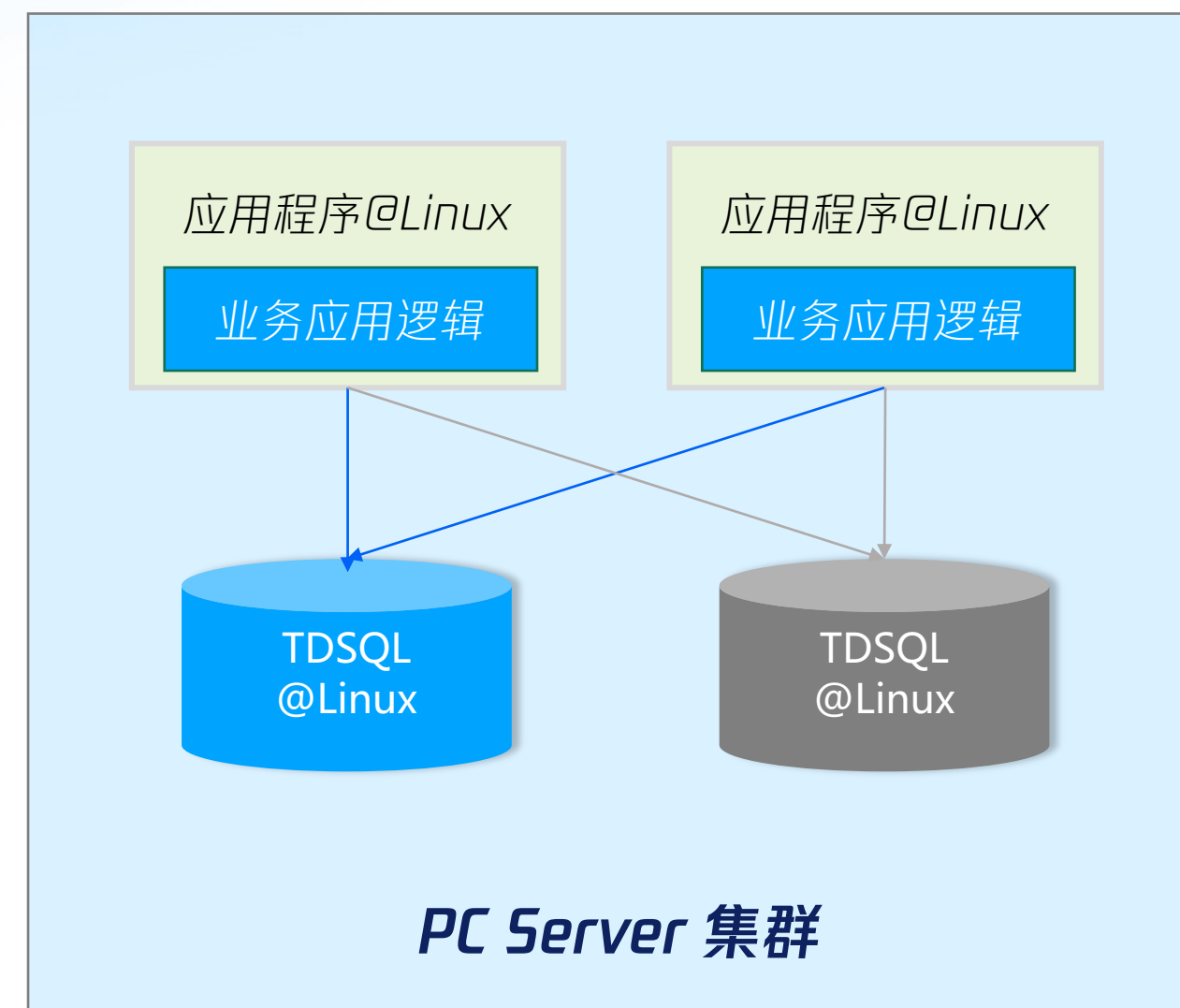
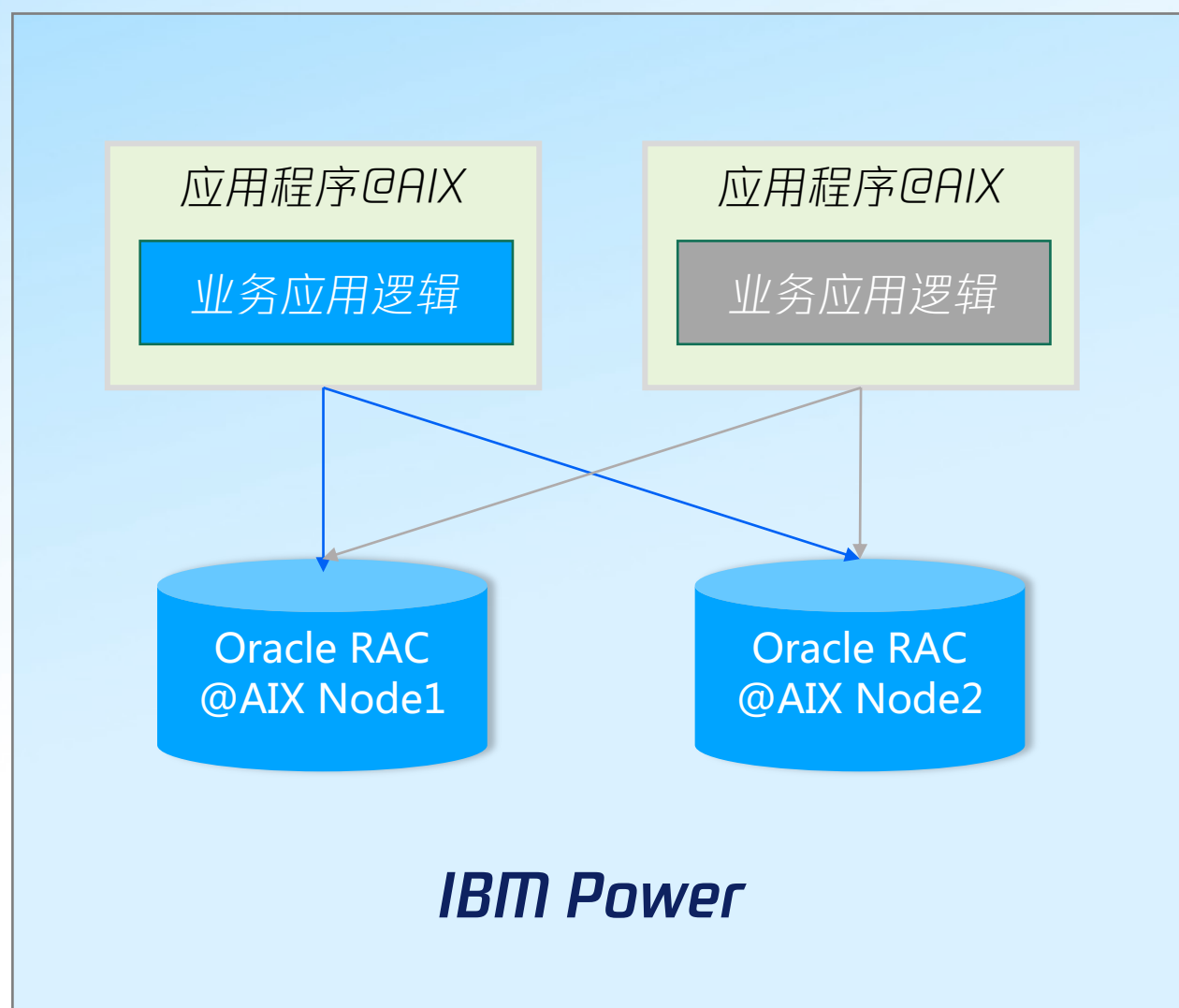
TSF Mesh对通信协议的支持能力：

- 七层协议：支持HTTP&grpc&dubbo协议，包括完整的服务治理能力支持，包括服务注册发现、服务鉴权、服务路由、服务熔断、服务限流
- 四层协议：支持tcp协议，由于无法识别私有通信协议内容，服务治理仅支持服务注册发现，TSF提供服务的可用以及监控运维能力



- ✓ 单体应用在不进行微服务改造的情况下，接入微服务治理体系
- ✓ 单体应用与普通微服务应用进行共同治理。
- ✓ 非所有单体应用都可以接入Servicemesh技术
- ✓ 可以叠加无状态改造

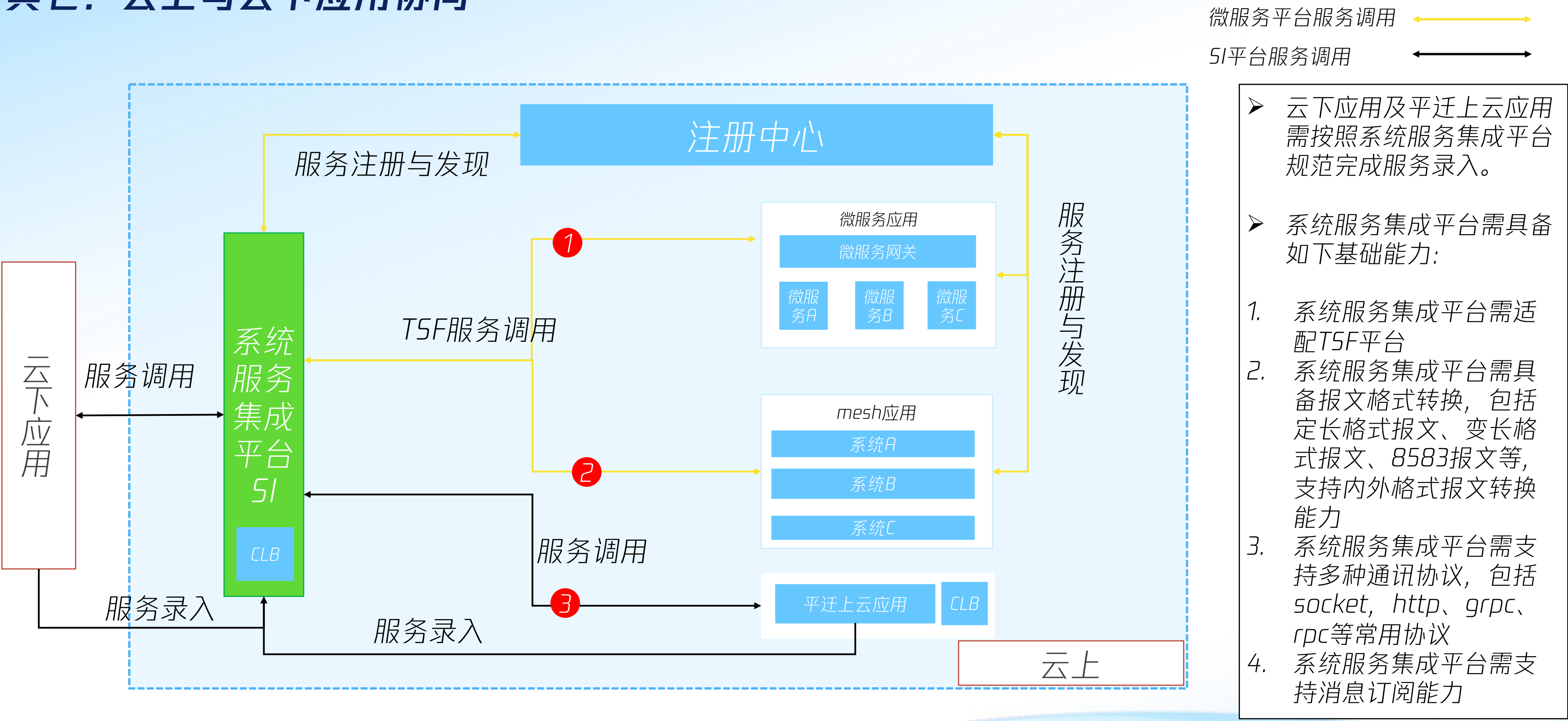
六类应用：@AIX平台类系统迁移计划



AIX类系统下移三大关键工作：

- **C语言程序在Linux平台重新编译：** 如果存在无法找到的引用包源代码，则需要寻求替代包方案。
- **数据库适配：** 没有一个国产数据库能够做到“SQL零改动”，且代码“SQL零改动”不代表运行效果不打折扣。必须进行在不改业务逻辑下的国产数据库适配。
- **应用无状态化改造：** 没有一台独立的PC Server算力可以PK小机，在保留单体应用的前提下，必须进行无状态化改造，确保可以通过横向扩容的方式达到算力的要求。

其它：云上与云下应用协同



注:

1. 对于有特殊运行环境要求的应用系统或者不具备上云条件的应用，保持在云下环境部署
2. 对于不具备适配TSF条件的应用采用平迁上云的方式部署到云上环境

应用上云改造中的难点

困难点

解决之道



应用迁移与改造

统一开发治理框架的执行力
ISV对统一开发框架的接受程度
非XC环境向XC环境迁移的兼容性
联合解决方案成熟度

- ✓ 项目整体咨询规划
- ✓ 兼容行内主流开源框架
- ✓ 联合解决方案充分测试并落地
- ✓ XC架构的充分测试验证



PaaS层建设难点

开发框架与主流开源框架兼容性
PaaS层服务跨中心高可用
北向接口与开源软件兼容性

- ✓ 最小接入代价换来极大收益
- ✓ 跨数据中心集群，确保业务无损
- ✓ 中间件服务无缝切换



IaaS层建设痛点

两地三中心，同城双活跨中心高可用
多种硬件资源兼容性
软件采用松耦合架构

- ✓ 不绑定特定硬件品牌
- ✓ 功能在不同硬件架构间保持一致
- ✓ IaaS, PaaS分层解耦
- ✓ 单层内组件之间松耦合

数据库资源建设

分布式架构对数据库的要求

数据强一致

- 确保多副本架构下数据**强一致**，避免故障后导致集群数据错乱和丢失。

金融级高可用

- 确保**99.999%**以上高可用；跨区容灾；同城双活；故障自动恢复。完善的两地三中心方案保证 $RPO=0$ ， $RTO<40$ 秒；为业务保驾护航；

架构开放性强

- 兼容MySQL接口和标准**开放**，技术人才通用、已有生态复用，TD5QL内核TX5QL完全开源。



企业级安全性

- 三权分立〔安全、审计、管理〕：全链路审计；数据库防火墙；透明加密支持国密；减少用户误操作/黑客入侵带来的安全风险。

数据库池化能力

- 线性水平扩展，数据库集群的物理资源、性能、还是功能，均提供良好的**扩展性**。
- 灵活部署，一个数据库集群内，可同时支持集中式数据库和分布式数据库部署，以租户形式进行资源使用，并提供资源隔离和弹性伸缩能力。

智能化运维

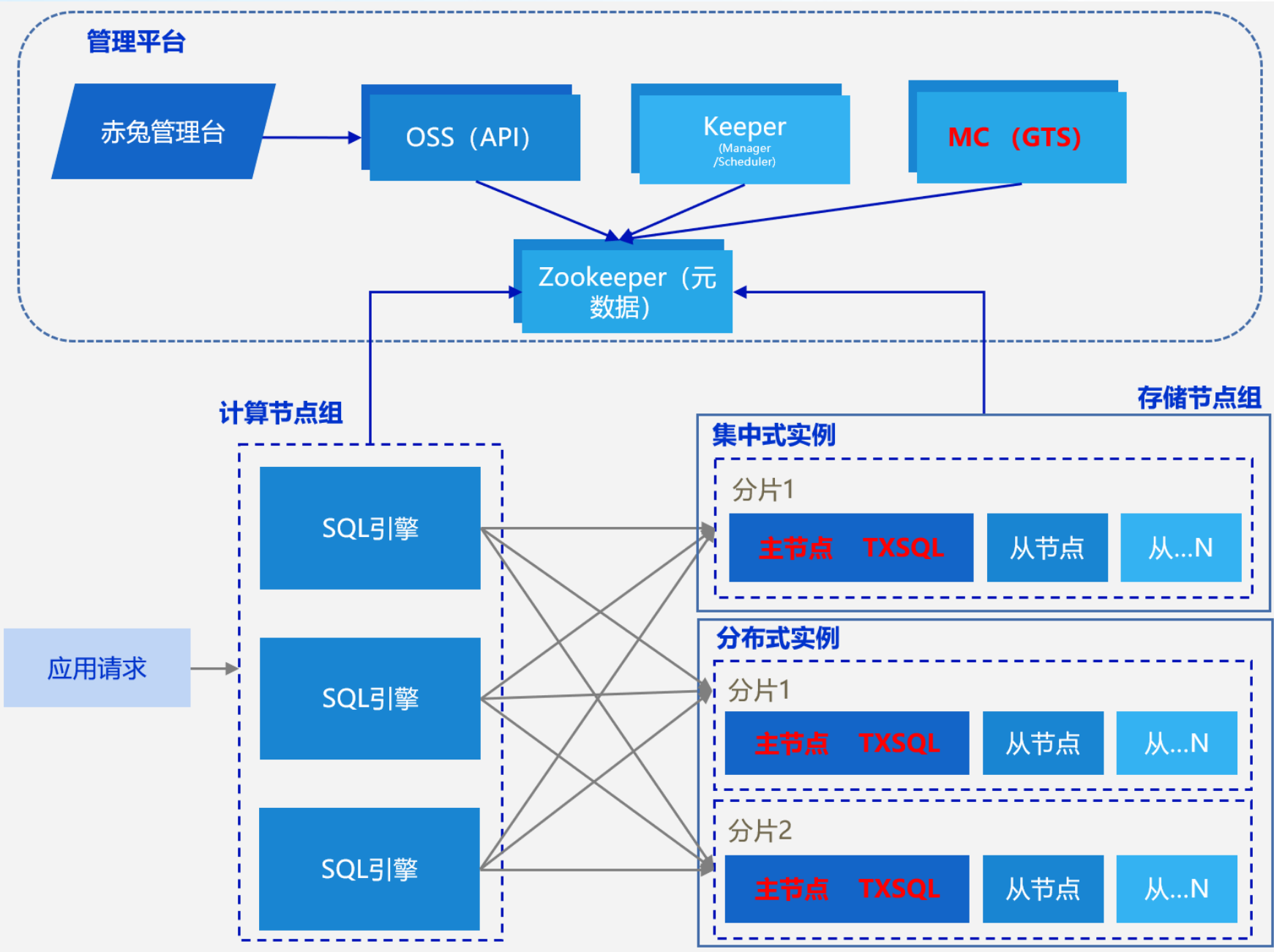
- 全白屏运维操作，包括智能运维〔扁鹊DBBrain〕、自助化运营管理平台〔赤兔〕等，帮助DBA快速上手。

腾讯TDSQL软件架构



一种数据库，两种架构

TDSQL 集群架构：同时承载集中式&分布式实例



TDSQL 高度解耦设计-集中式&分布式一体化的根基

TDSQL 采用分布式集群架构，该架构具有较高的灵活性，不仅简化了各个节点之间的通信机制，也简化了对于硬件的需求。

该架构采用模块化设计理念，存储组件与计算组件、高可用容灾组件、管理组件等解耦，这种设计使得 TDSQL 的集中式实例、分布式实例可以有机的结合在一套物理集群中，实现混合部署。

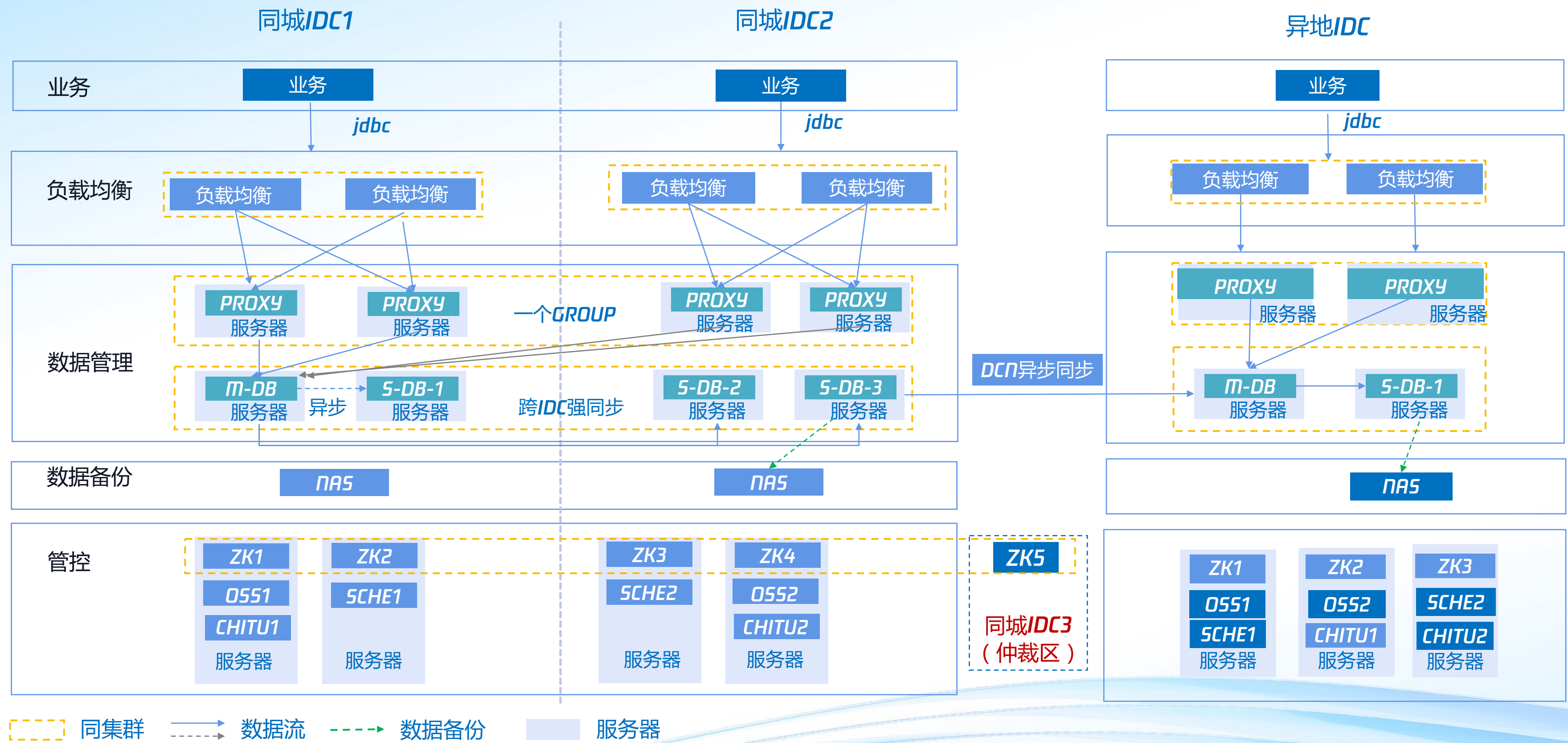
集中式实例：无需分布式改造

- 一个分片，相当于一个集中式数据库实例，即可以理解为 MySQL/Oracle 主备高可用架构的单体数据库。大部银行业务系统的数据量和交易并发都在集中式数据库的可承载范围，选择 TDSQL 集中式实例，业务迁移时无需分布式改造。

分布式实例：无需手工分表

- 多个分片，即组成分布式数据库实例。分布式数据库实例适用于数据量较大、交易请求并发较高的核心类系统。TDSQL 淡化了水平拆分的概念，无需用户手动配置分表逻辑，无需用户额外部署管理中间件，只需要在建表时指定分表关键字即可。

TDSQL两地三中心部署架构



根据银行银行要求，数据库架构满足同城主中心2台DB服务器，备中心2台DB服务器，异地一台DB服务器建设要求

应用系统接入流程

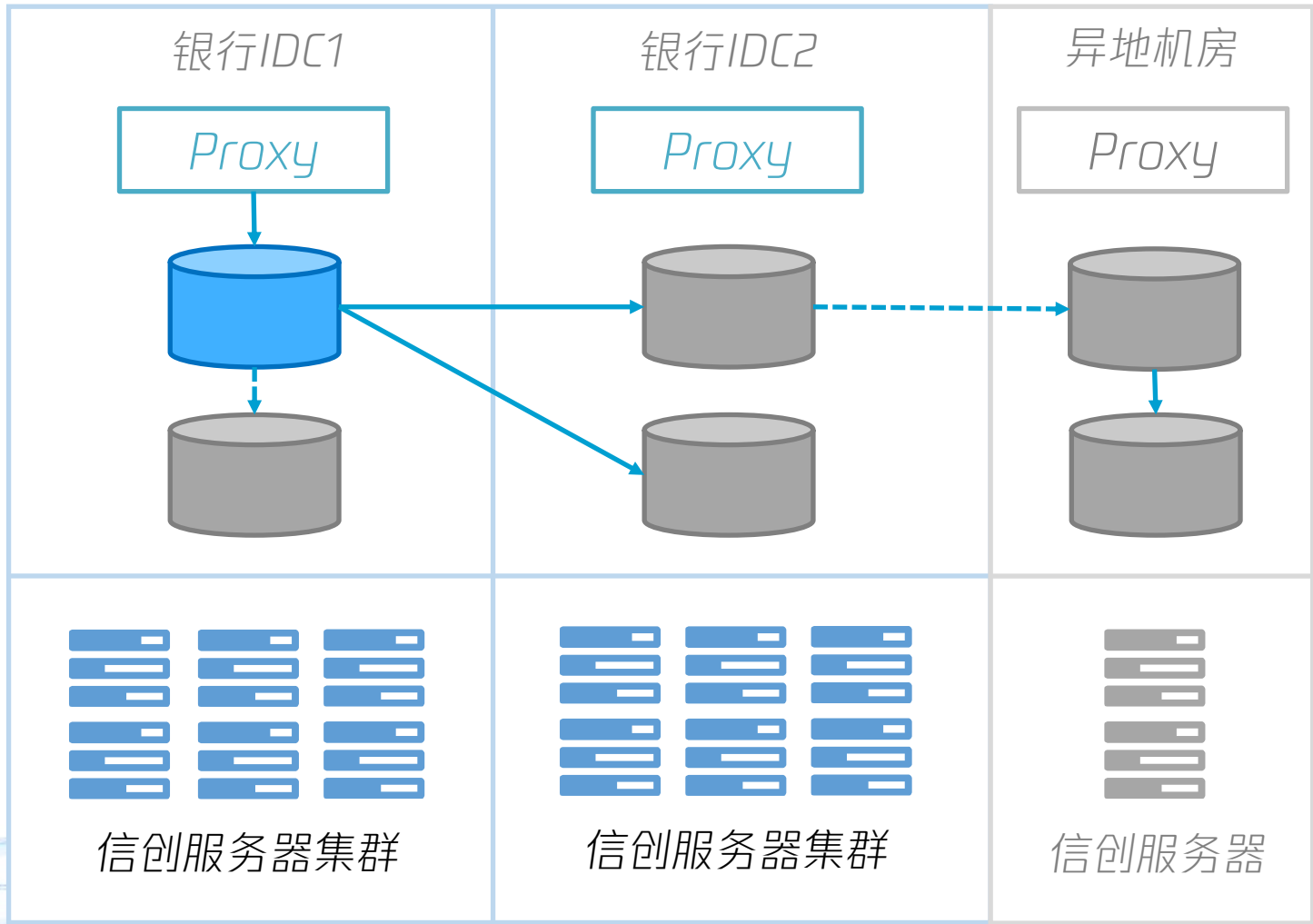


计算及存储资源规划



TDSQL资源池

TDSQL资源池通过在赤兔管理平台中纳管信创物理机后，抽象计算与存储资源形成数据库专用资源池。使用者可以根据自身应用特征创建单中心或双中心和不同副本数量的架构组合。



部署模式选择规则

		数据量与并发量要求	
业务连续性要求	TDSQL架构套餐	集中式架构	分布式架构
	单中心·3副本	✓	✓
	双中心·4副本	✓	✓
	两地三中心·6副本	✓	✓

➤ 单中心3副本 部署原则

对于业务连续性要求不高的，对应《信息安全等级保护管理办法》及银行银行业务系统分级中的3级系统，建议采用单中心部署。

➤ 双中心4副本 部署原则

对于业务连续性有较高要求的，监管有明确要求的，对应《信息安全等级保护管理办法》及城市联盟业务系统分级中的2级系统，建议采用双中心部署。

➤ 两地三中心6副本 部署原则

对于业务连续性有较高要求的及对异地灾备监管有明确要求的，对应《信息安全等级保护管理办法》及银行银行业务系统分级中的1级系统，建议采用两地三中心部署。

✓ 集中式架构 测算原则 [并且]

- 1. 落到数据库的 QPS≤5W，建议采用集中式架构。
- 2. 账户数 ≤500万，建议采用集中式架构。
- 3. 应用系统预估 数据文件大小≤2T的，建议优先采用集中式架构。
- 4. 应用系统分析统计类SQL居多，对函数依赖较多且应用侧改造困难的场景建议采用集中式架构。

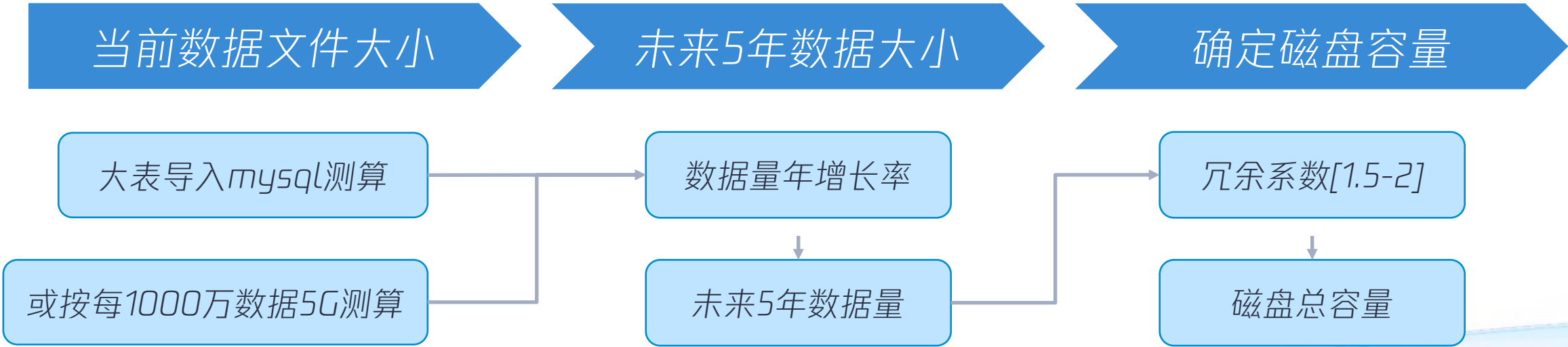
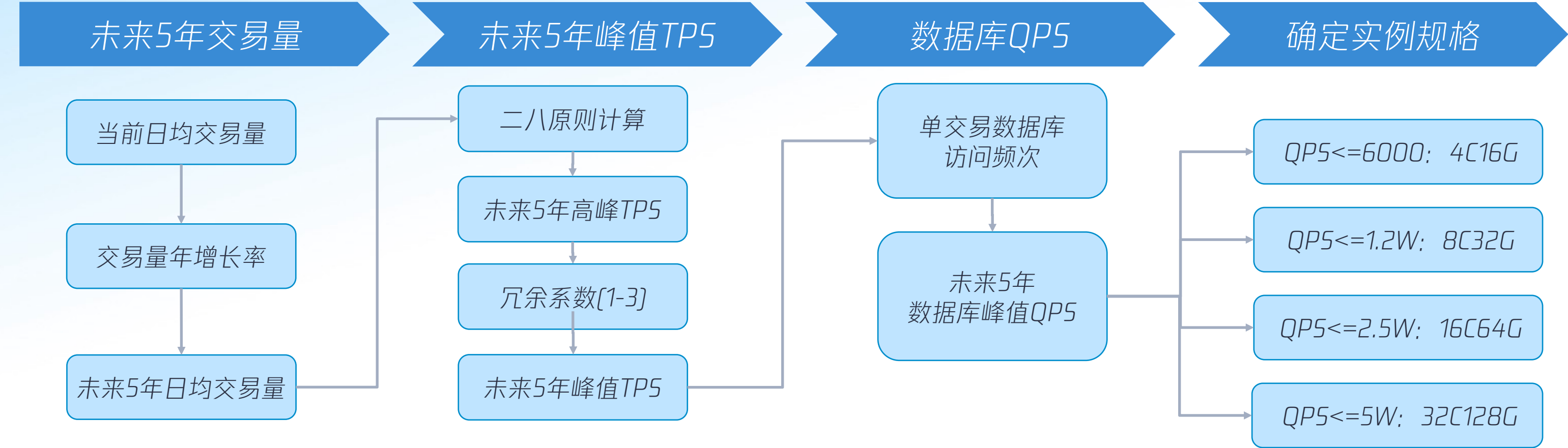
✓ 分布式架构 测算原则 [或者]

- 1. 落到数据库的 QPS>5W，建议采用分布式架构。单库QPS不超过5W。
- 2. 账户数 >500万，建议优先采用分布式架构。单库不超过500万账户。
- 3. 应用系统预估 数据文件大小>2T的，建议优先采用分布式架构。单库不超过2T。

说明：

- 1. 数据文件和QPS按年增长率预估对应5年后的值进行测算。
- 2. 数据库QPS=应用TPS * 单交易数据库访问次数

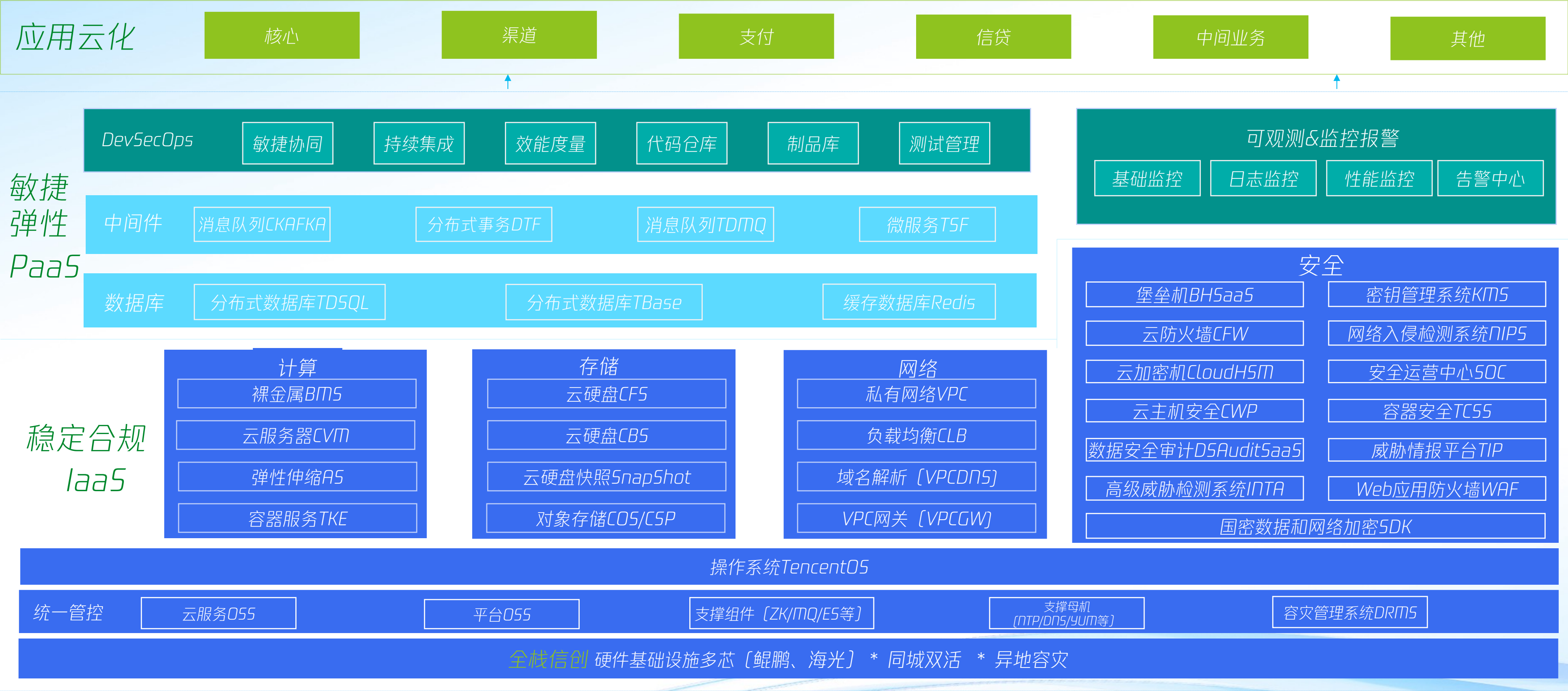
部署实例规格选择



1. 落到数据库的 **QPS > 5W**，建议采用分布式架构。单库QPS不超过5W。
2. 账户数 **> 500万**，建议优先采用分布式架构。单库不超过500万账户。
3. 应用系统预估 **数据文件大小 > 2T**的，建议优先采用分布式架构。单库不超过2T。

基础资源层供给建设

整体云平台软件架构



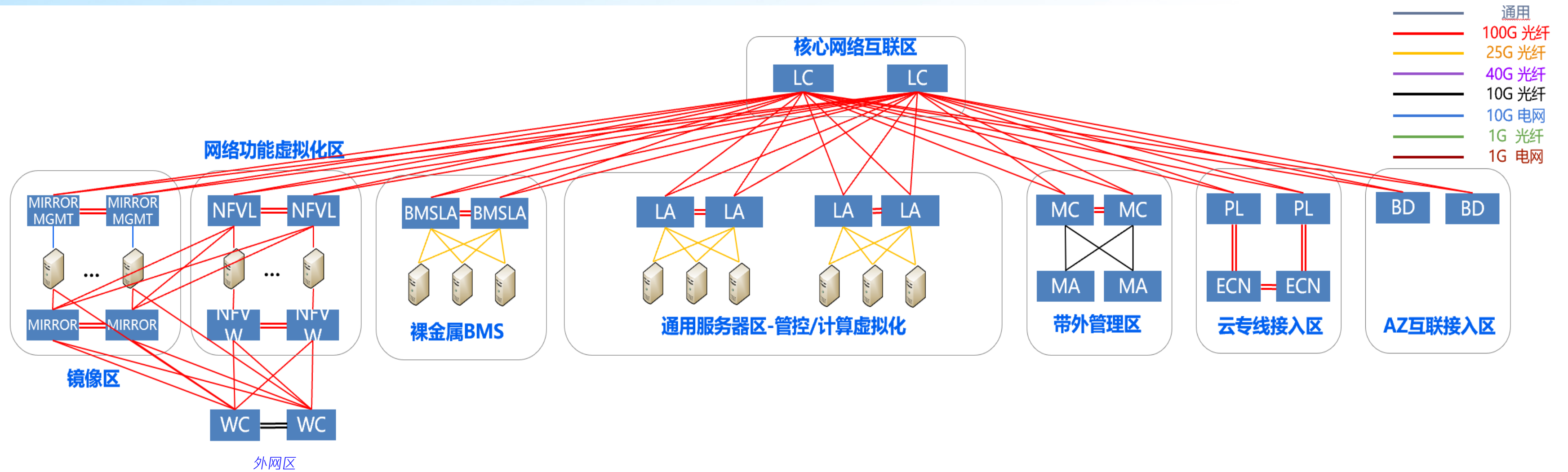
整体部署规划：建设开发测试、生产环境同城双活+异地容灾云平台

建设开发测试、生产环境同城双活+异地容灾云平台



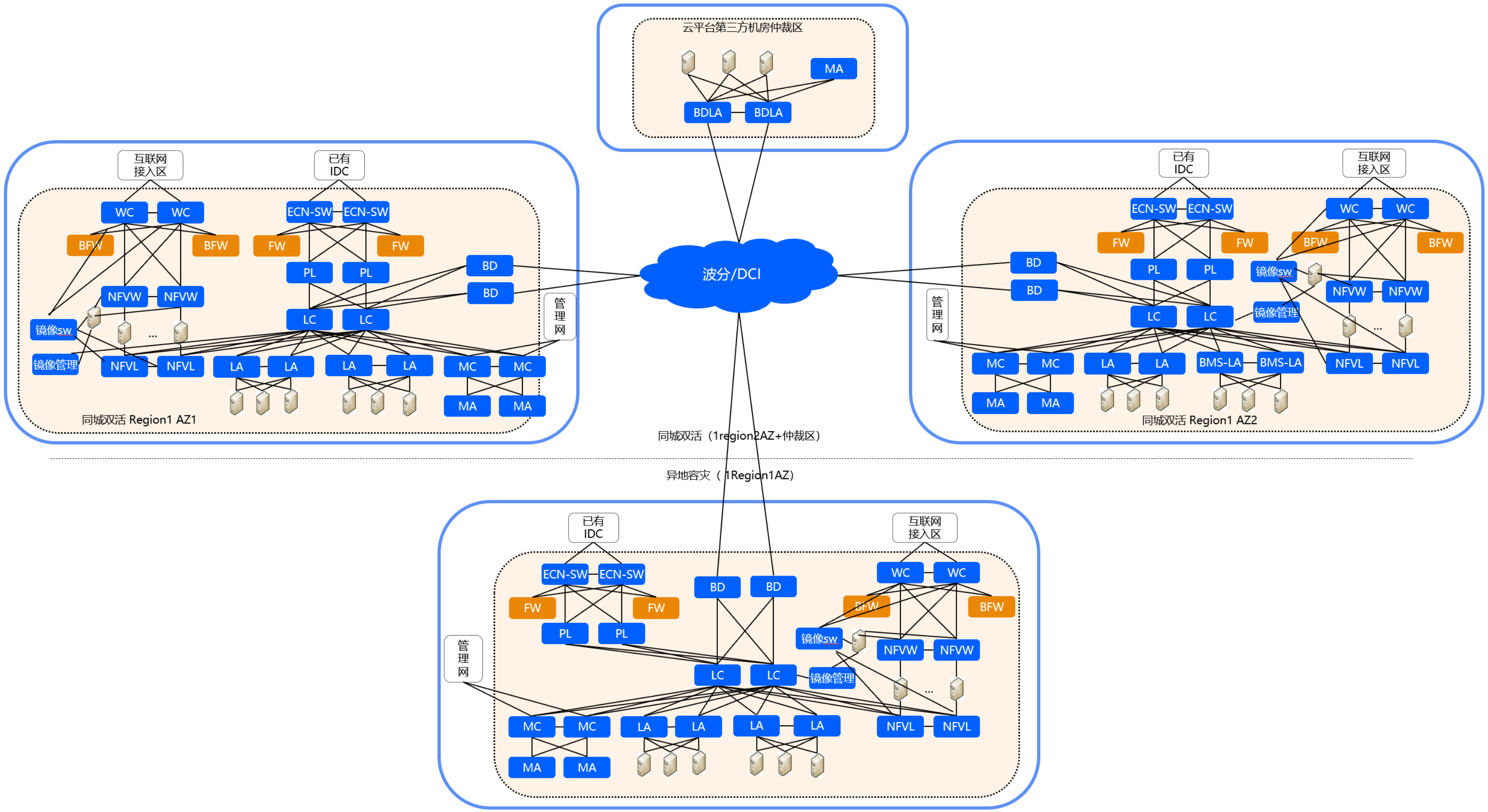
备注：当前数据更新至2023.9.5，后续若有变动，以最终沟通确认数据为准

云平台物理网络区域介绍

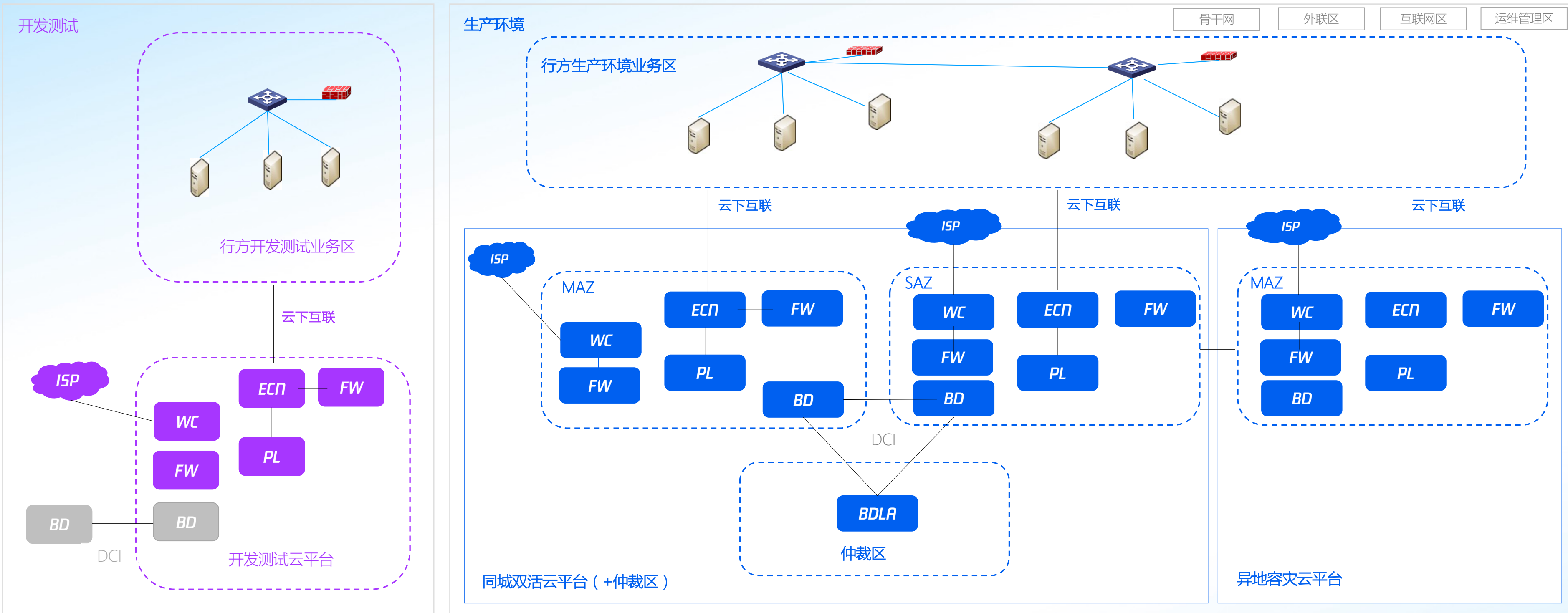


- 核心网络互联区〔内网核心区〕：用于连接模块内各个区域
- 通用服务器区：用于部署的通用云产品及管控服务器，通用云产品按需进行建设
- 裸金属BMS服务器区：用于部署的BMS及自定义服务器〔如TD5QL，TBD5〕
- 网络功能虚拟化区：用于部署的网关服务器
- 带外管理区：连接网络设备带外管理口和服务器iLO口
- 云专线接入区：提供云外用户访问云上业务的接口
- AZ互联接入区：提供AZ之间互联的接口
- 镜像区：用于部署天幕和御界探针服务器
- 外网区：用于连接互联网

云平台生产环境物理网络架构



整体网络规划：开发测试，生产双活+异地容灾云平台与行方各自业务区打通

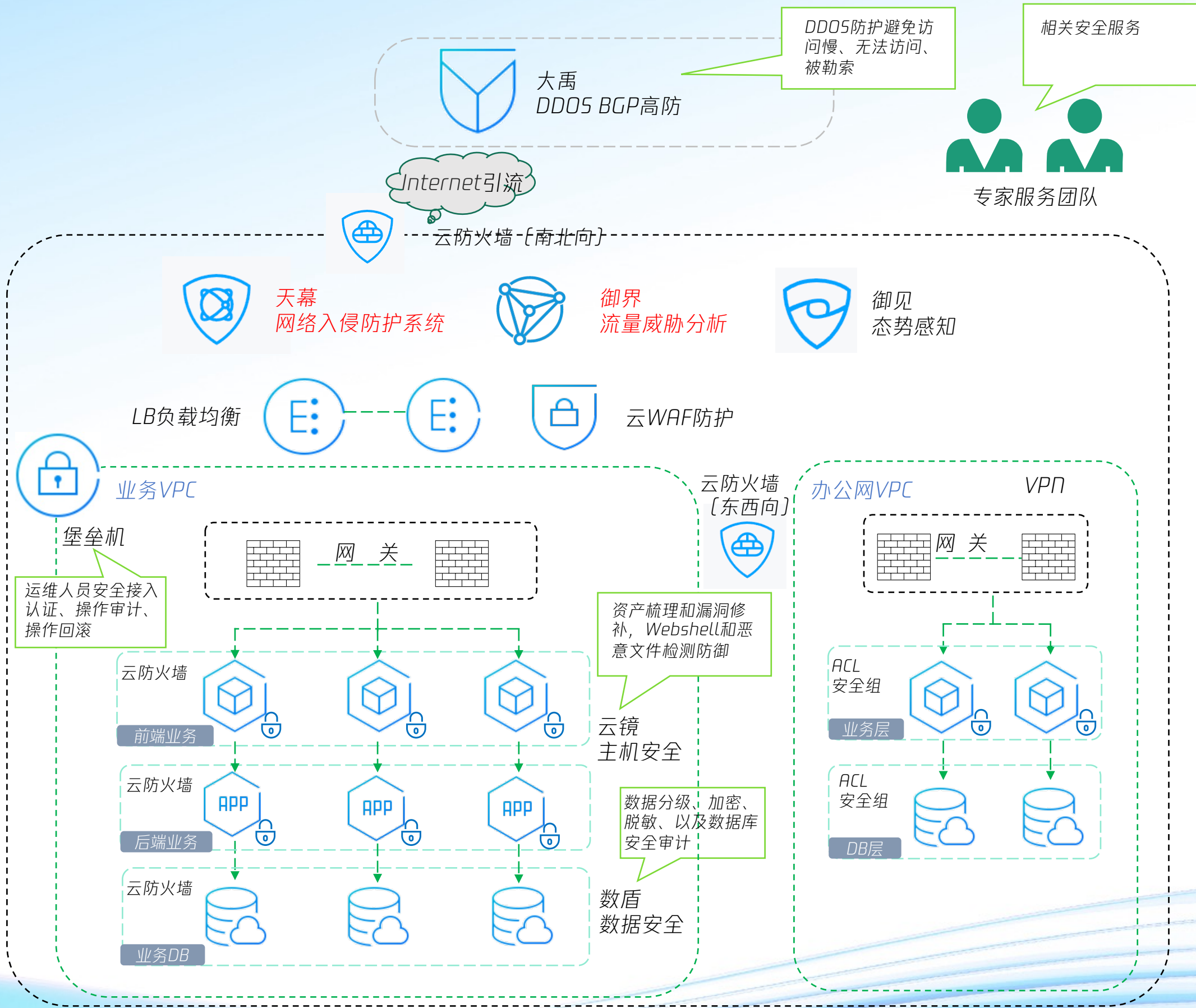


- 开发测试云平台（1region1AZ）通过ECN交换机（加FW）与行方开发测试业务区进行网络联通；若需要连接其他开发测试AZ需求，可以通过扩容BD交换机（DCI）进行其他AZ连接；后期若有互联网需求可以通过WC交换机（加FW）进行连接
- 同城双活生产环境云平台（1region2AZ+仲裁区）通过ECN交换机（加FW）与行方生产环境业务区进行网络联通，互联网需求可以通过WC交换机（加FW）进行连接，双活2AZ通过BD交换机进行互联且分别与仲裁区BDLA互联
- 异地容灾云平台（1region1AZ）通过ECN交换机（加FW）与行方生产环境业务区进行网络联通，互联网需求可以通过WC交换机（加FW）进行连接。

云平台安全建设思路-建设满足等保2.0规范的安全框架



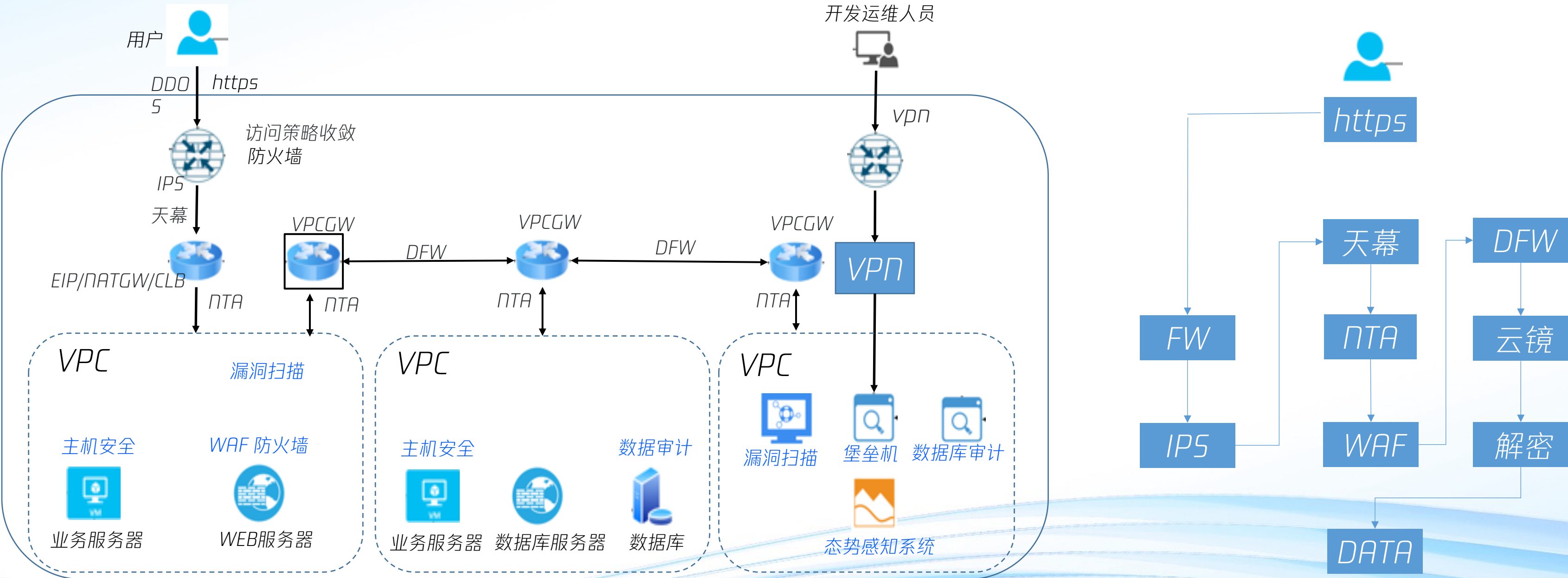
等保2.0规范的安全能力匹配



安全产品	目前部署模式	是否等保必选
云WAF	TCE集成	必选
云镜 主机安全	TCE集成	必选
御界 流量威胁分析	TCE集成	可选
天幕 网络入侵防护系统	TCE集成	必选
数盾 数据库审计	TCE集成	必选
云堡垒机	TCE集成	必选
安全态势感知	TCE集成	必选
密钥管理服务	TCE集成	可选
威胁情报平台	TCE集成	可选
云防火墙	TCE集成	可选
漏洞扫描平台	TCE集成	必选/

网络流量视角建设

一站式构建立体安全防御体系，从互联网边界到业务中心全数据链路的检测和控制，全方位覆盖网络和通讯安全，主机安全，应用安全，数据安全。租户根据业务重要程度、业务特点，通过选择不同的安全防护服务，实现不同强度的安全保护，集中资源优先保护涉及核心业务或关键信息资产。



云原生高可用体系建设

业务连续性设计思考



容灾

- 支持同城、异地容灾， $RPO=0$ ， RTO 很短
 - 数据水平拆分，分片有同城和异地副本，容灾时支持自动切主。
 - 应用流量与数据分片一致，支持无脑切换，快速容灾。
- 单元化多活，缩小故障影响范围
 - 业务分散在多个地域、数据中心、多个单元，故障隔离域粒度非常小
- 自动化容灾平台，支持容灾预案和容灾演练
 - 对接多产品的容灾切换和恢复操作，自动化
 - 提供多场景下的自动化容灾切换和恢复能力
 - 应对故障手段多，控制灵活，故障切换速度快



弹性

- 提升扩容效率，按单元灵活部署
 - 按单元快速扩容，灵活调配流量
- 提升扩展性，异地多活，理论上无限扩展
 - 突破数据库瓶颈，不受单库、容量上限限制
 - 突破城市瓶颈，异地多活
- 提升容量预估能力，支持全链路压测
 - 识别压测流量，全链路压测，容量预估



灰度

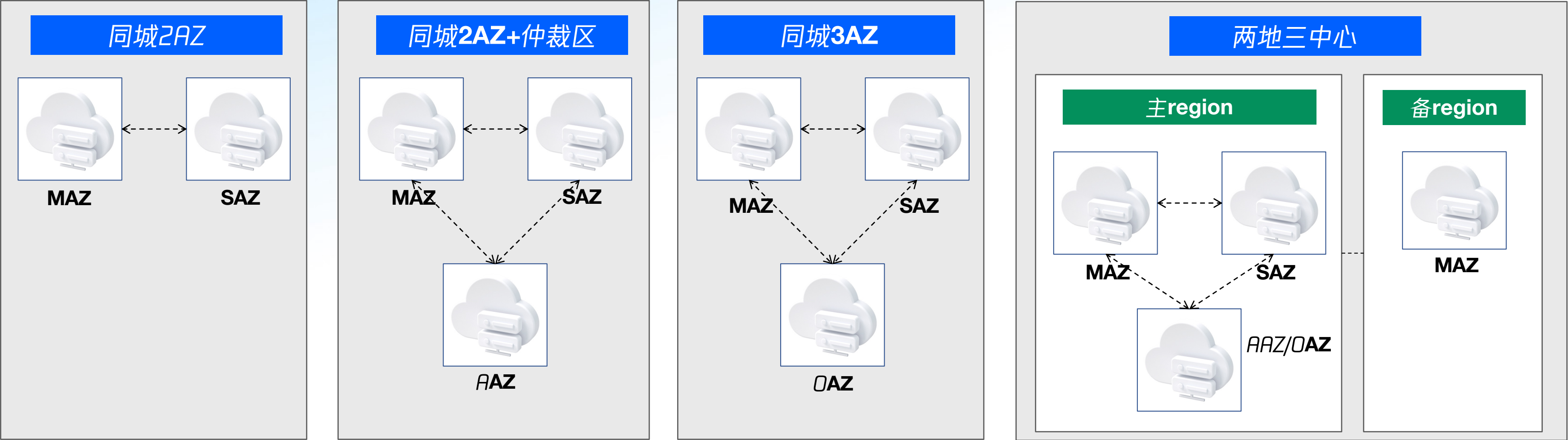
- 支持灰度发布，灵活调拨灰度流量
 - 通过小规模生产流量 [UID, 单元] 充分验证新版本，及时回切应急，保证新产品的平滑上线
- 支持蓝绿发布，新老调用单元隔离
 - 新老单元调用隔离，避免交叉访问兼容性问题，提升发布效率



架构

- 支持架构演进
 - 从单机房->同城->异地>混合云
- 支持异地多活
 - 单元化流量收敛，是实现异地多活的必要条件

业务连续性演进路线



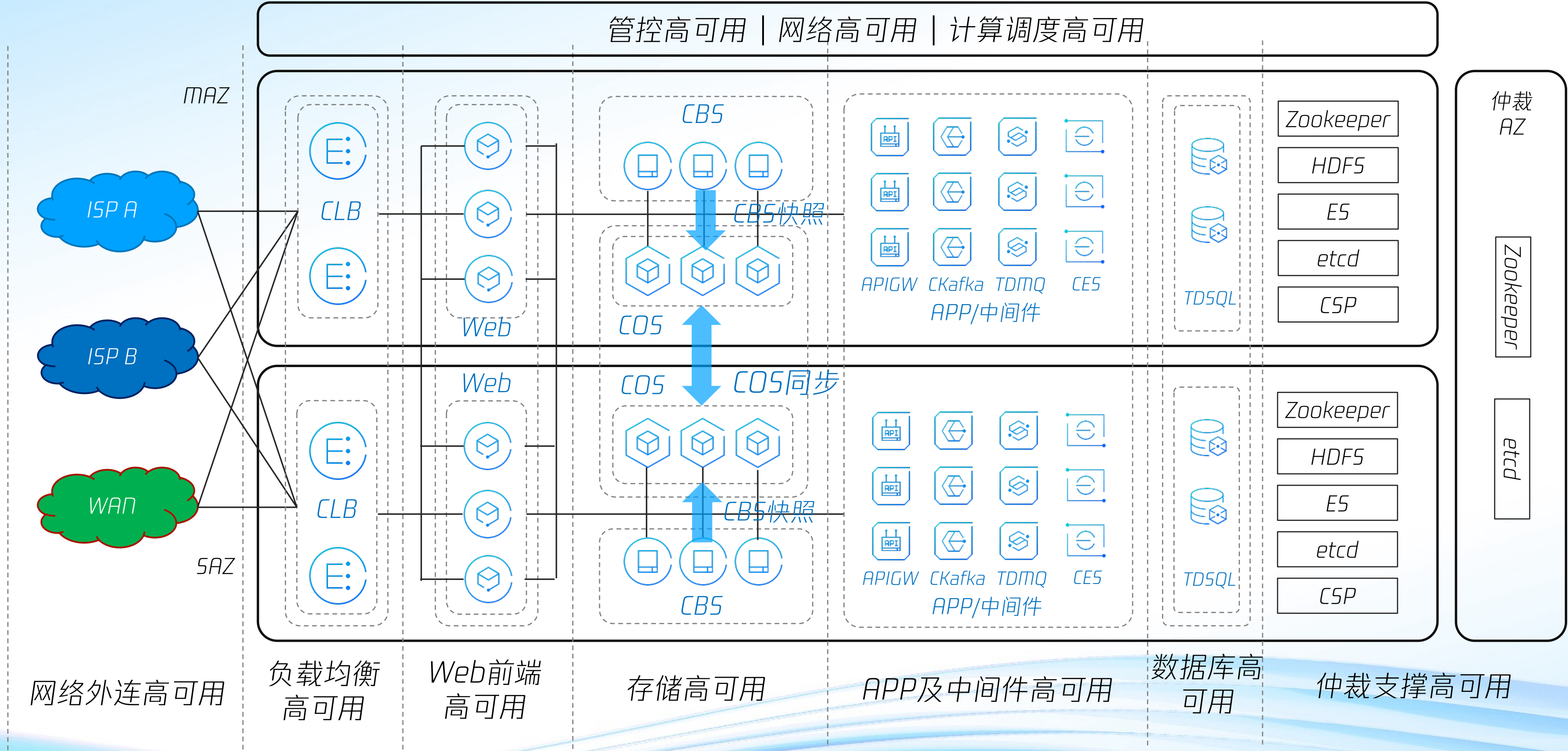
优势

- 支持多种容灾架构，兼顾成本和不同容灾能力等级诉求
- 云平台和产品跨AZ多活，无单点故障
- 数据跨AZ强同步确保关键数据RPO约等于0
- 容灾演练[DRMS]和故障演练[Oscar]能力产品化，容灾演练和故障演练例行化、自动化，日常演练提前发现问题，解除风险，提升业务系统健壮性
- 容灾故障一键式切换能力，分钟级RTO

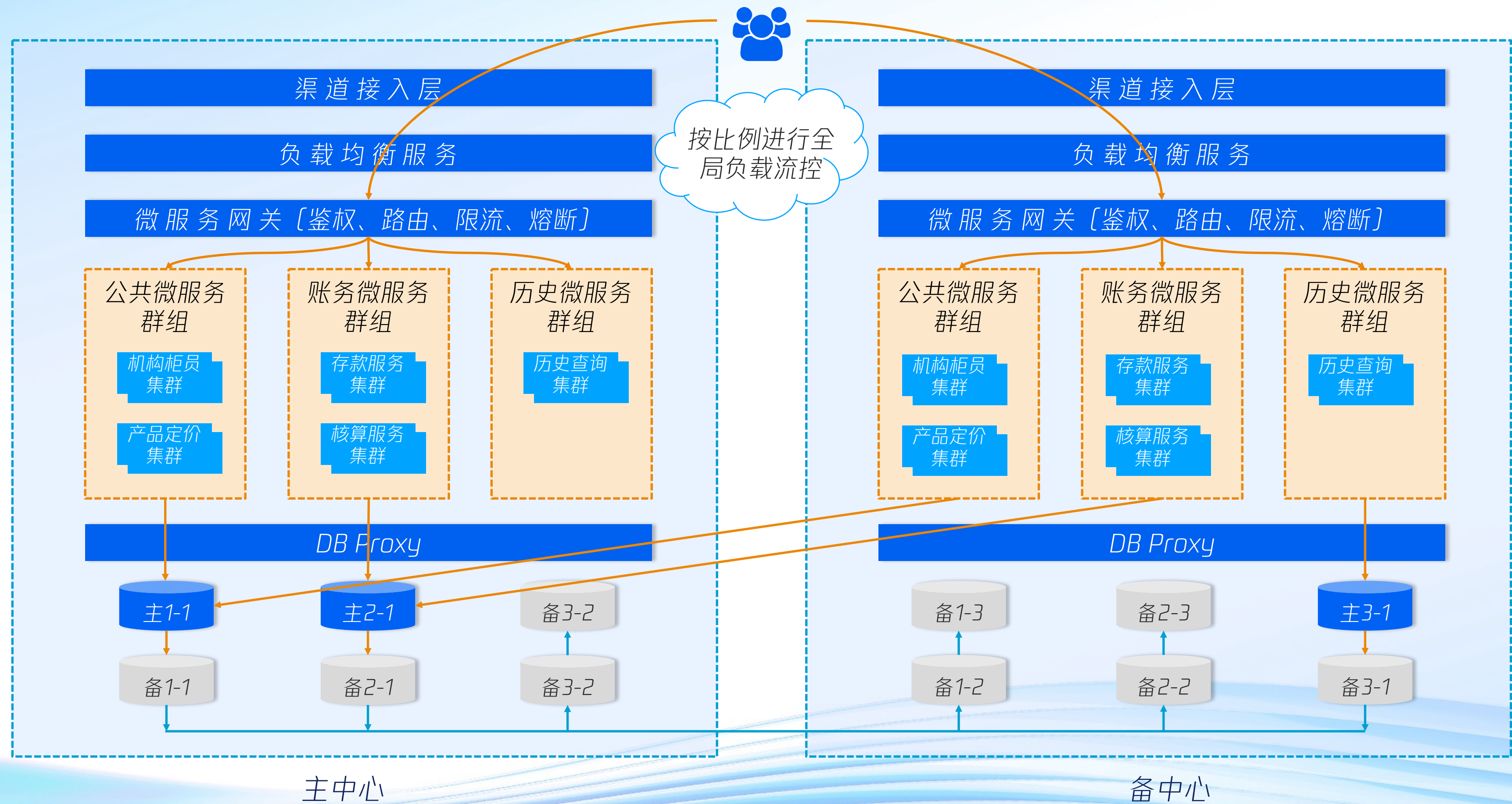
价值

- 满足监管合规要求
- 提升系统健壮性
- 降低灾备建设成本，提升灾难恢复能力
- 演练应急响应机制，减少业务恢复风险，降低误操作风险

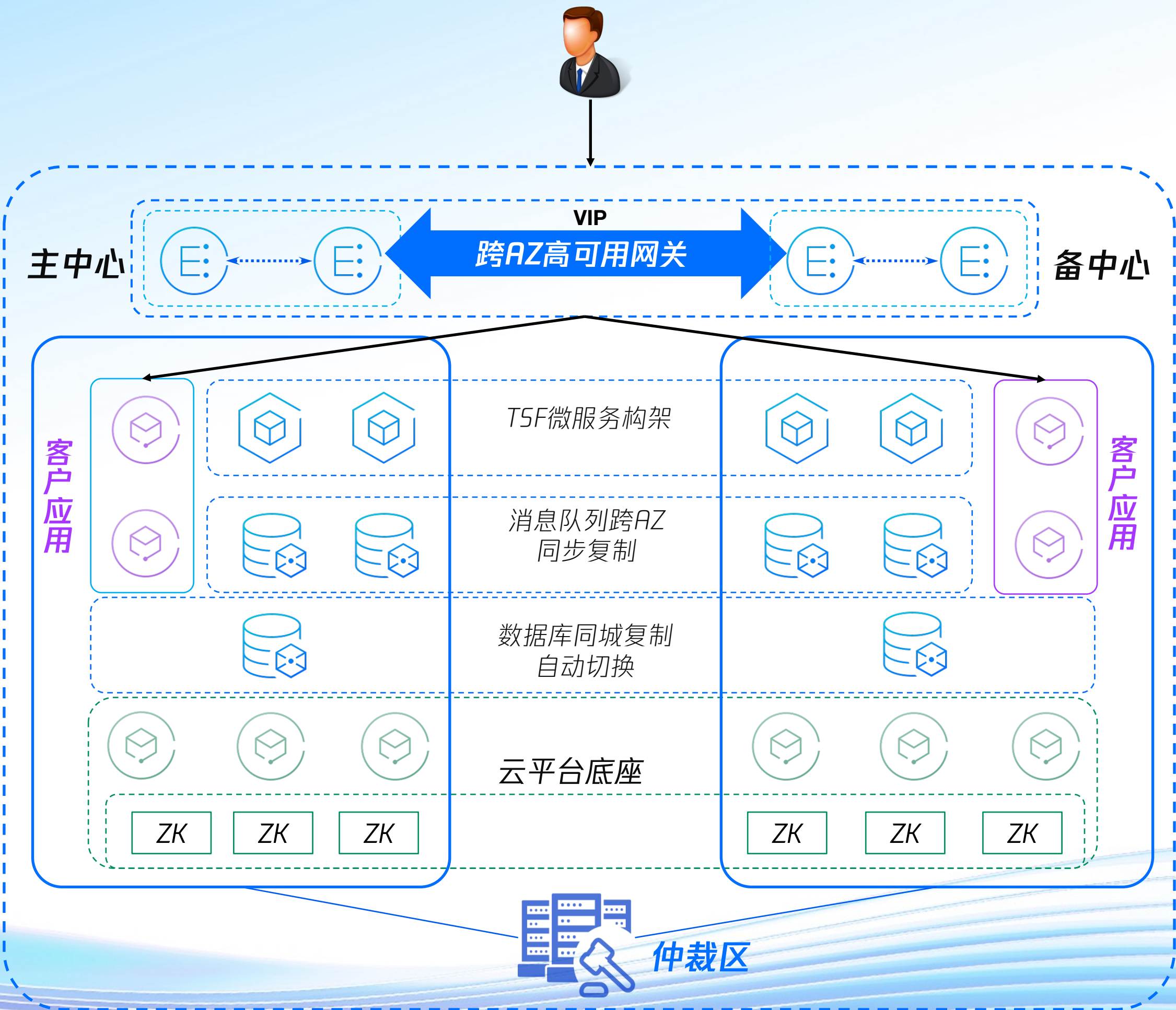
TCE产品实现HA机制



第一阶段：云平台同城双活建设 - 应用视角



第一阶段：云平台同城双活建设 – 基础架构视角



关键点

1.原生的主备/双活支持能力

- 无需借助 GSLB 即可实现流量切换
- 当客户应用支持多实例部署时,具有多活能力
- 当客户应用只支持单实例部署时, 业务获得主备能力

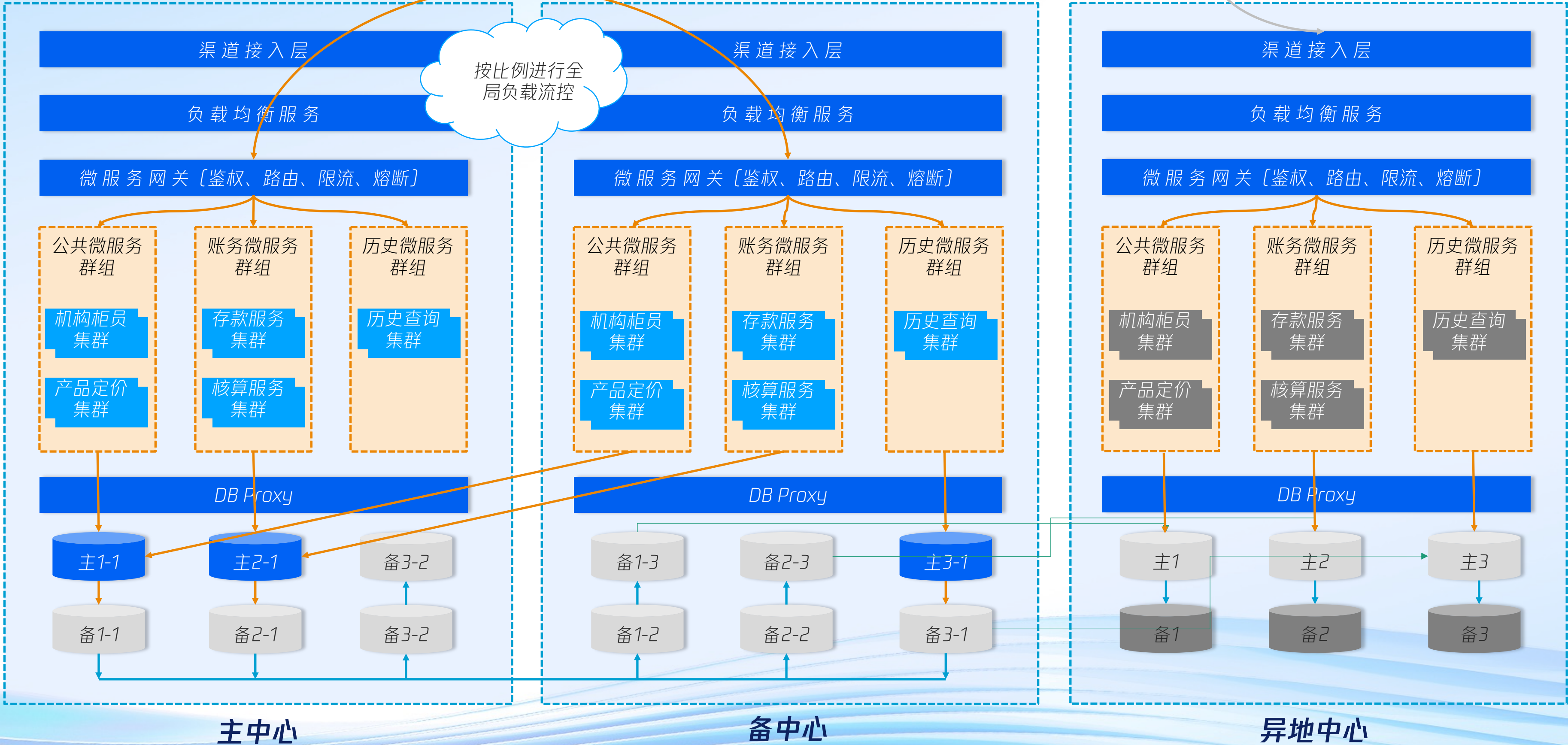
2.快速的灾备恢复能力

- $RPO=0$, RTO 接近于0的恢复能力.
- 减少应用不必要的业务切换
- 应用对跨 AZ 访问延时不敏感

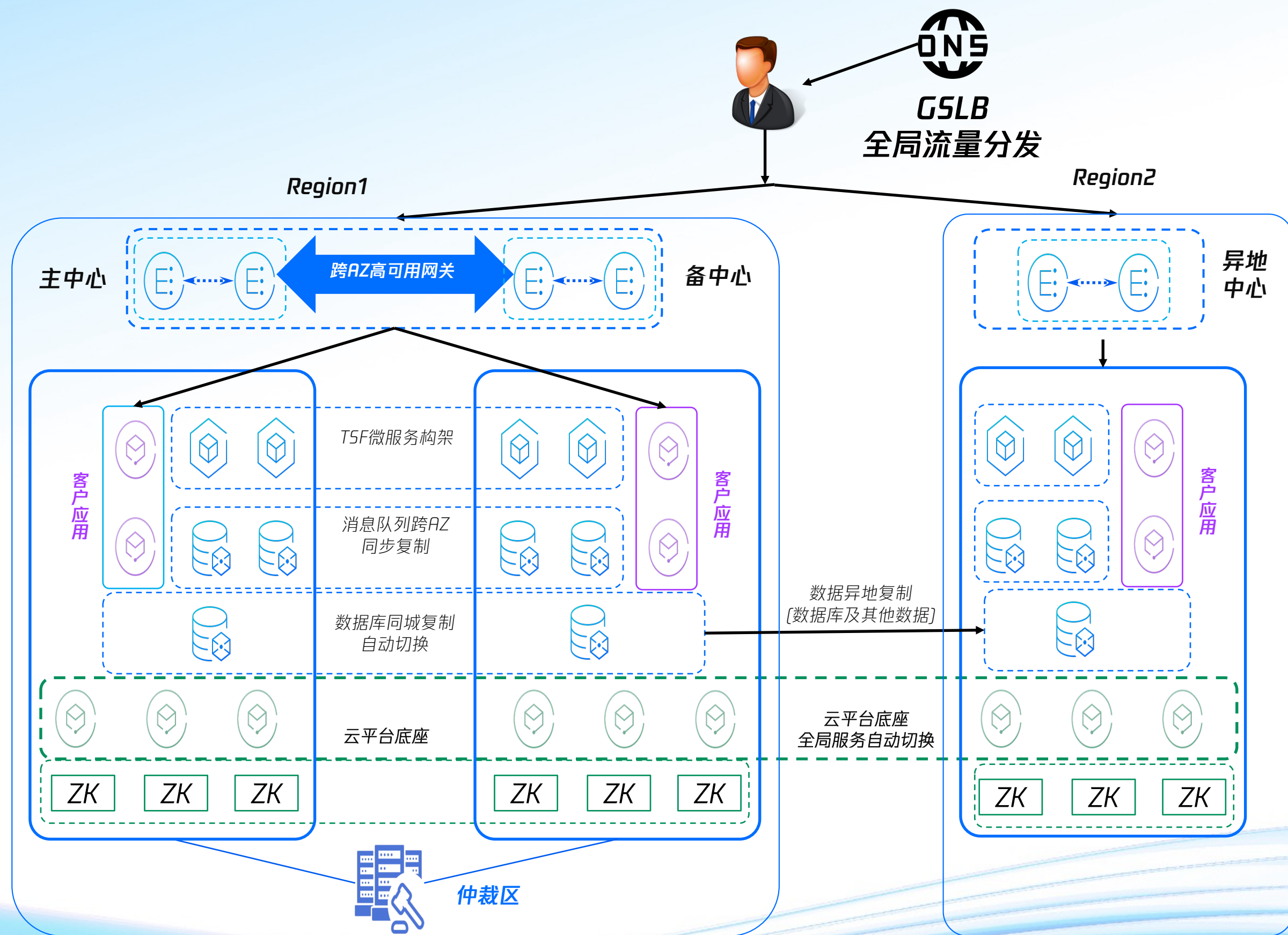
3. 自动化的切换能力

- 提供跨AZ容灾能力,当某AZ整体出现故障时, 由另一AZ完整接管
- 所有故障切换均自动进行,基本无需人工干预

第二阶段：云平台两地三中心建设 - 应用视角



第二阶段：云平台两地三中心建设 - 基础架构视角



关键点

1.Region 间主备/双活支持能力

- 需借助 GSLB 实现流量切换
- 支持应用主备切换能力
- 双活支持能力取决于业务模式及数据库技术

2.Region 间灾备恢复能力

- 中间件没有跨 Region 的高可用能力
- $RPO > 0$, $RTO > 0$ 的恢复能力.

3. Region间自动化的切换能力

- 提只需要运维一套管控平台,平台管控在多 Region 间自动切换.
- 业务的切换依赖Runbook, 需要判断切换条件, 编排切换过程, 切换后检查所有故障切换均自动进行,基本无需人工干预

THANKS

谢谢观看