

从业务中积累 互联网架构的技术体系

韩超

目录

- 第一部分 业务与技术的共同目标
- 第二部分 现代互联网架构的大格局
- 第三部分 架构技术积累的几个方面



第一部分 业务与技术共同目标

业务与技术共同目标

传说：脱离业务场景，空谈架构是要流氓

业务支撑和技术发展“共荣”的几个方面：

- 量级：高并发、大数据中对技术的挑战和促进
- 复杂度：复杂环境中的运作良好的架构
- 业务闭环：用策略架构直接促进业务指标

业务视角

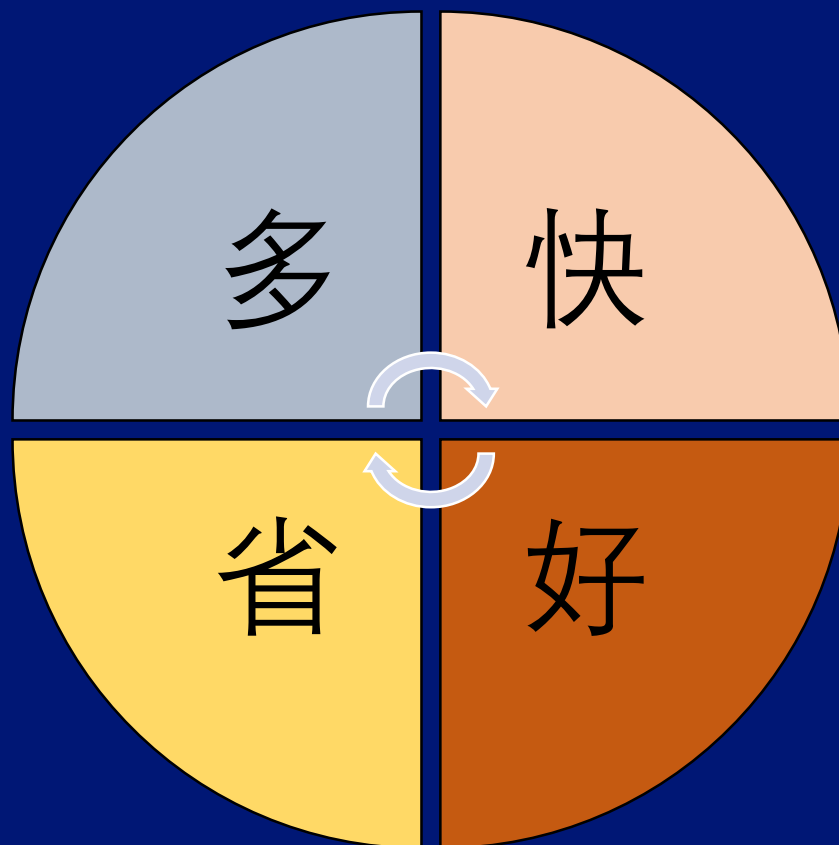
技术视角

功能丰富，满足需求多

架构完备，扩展能力强

节省单项目人力

节省资源、长尾人力



功能上线快

开发、CICD、运维高效

业务指标上升

技术驱动业务闭环

【理念】互联网架构 \leftrightarrow 造车、开车和修车

- 实现业务 => 造新车、开新车
- 量级增加、时间积累 => 需要动态增加更多行驶的车
- 支持业务的线上服务的修复和维护 => 在行驶中修车
- 业务变更，升级线上服务 => 在行驶中换车
- 业务形态复杂 => 不知道该用何种的车.....

+

造车

开车

修车

行驶中换车

行驶中修车

第二部分

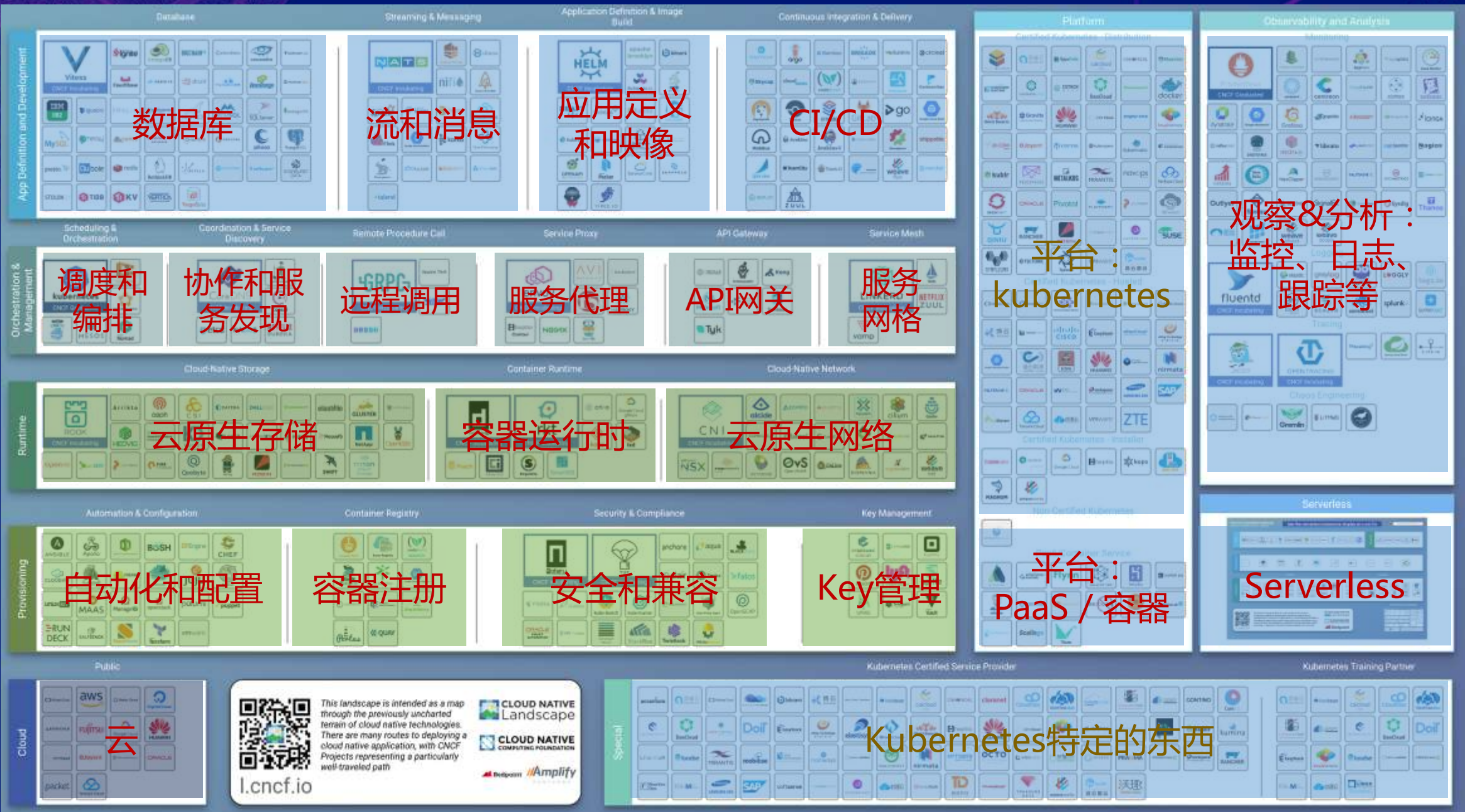
当前互联网架构的格局

当前互联网架构的现状和思考

互联网架构技术的现状和特点：

- 各种基础架构&基础组件已经很丰富（参考[CNCF](#)）
- Docker和kubernetes主导了架构的纵深发展
- 开源的发展非常坚实，主导了主要的技术方向
- 常用的业务系统都有最佳实践：推荐、社交、电商等

看格局 => 找特点 => 寻进步



数据库

流和消息

应用定义和映像

CI/CD

调度和编排

协作和服务发现

远程调用

服务代理

API网关

服务网格

平台: kubernetes

观察&分析: 监控、日志、跟踪等

云原生存储

容器运行时

云原生网络

自动化和配置

容器注册

安全和兼容

Key管理

平台: PaaS / 容器

Serverless

云

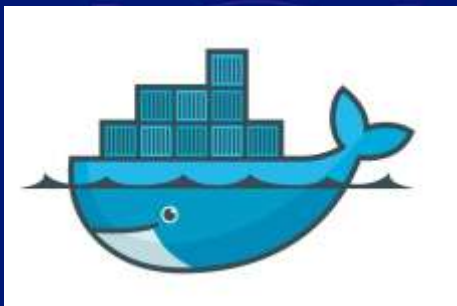
Kubernetes特定的东西



This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path



Docker和Kubernetes主导新架构格局

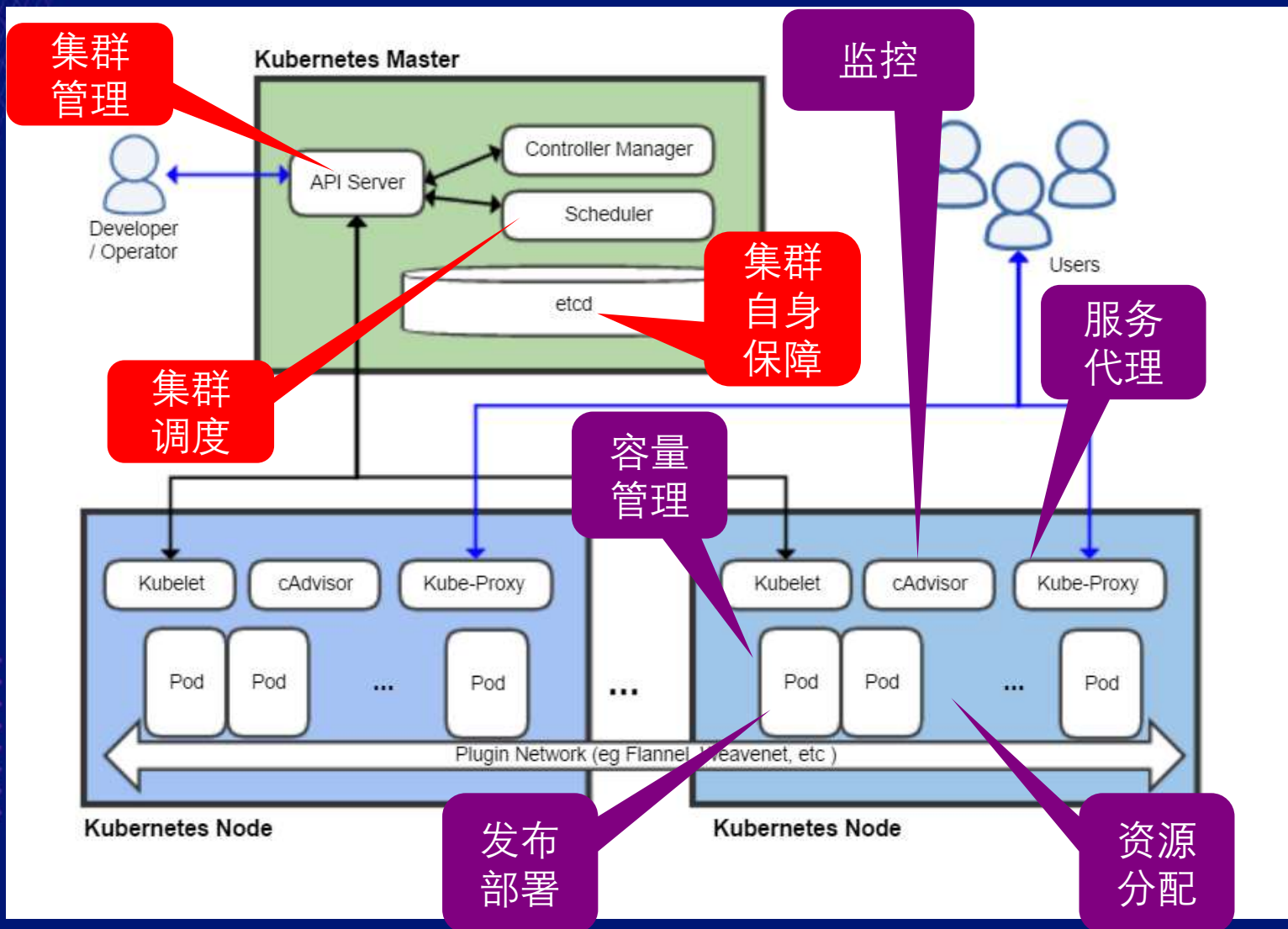


现代互联网架构的龙骨和趋势：

- 容器引擎：Docker
- 容器编排：Kubernetes
- 服务调用：Envoy
- 服务网格：Istio

架构核心的问题

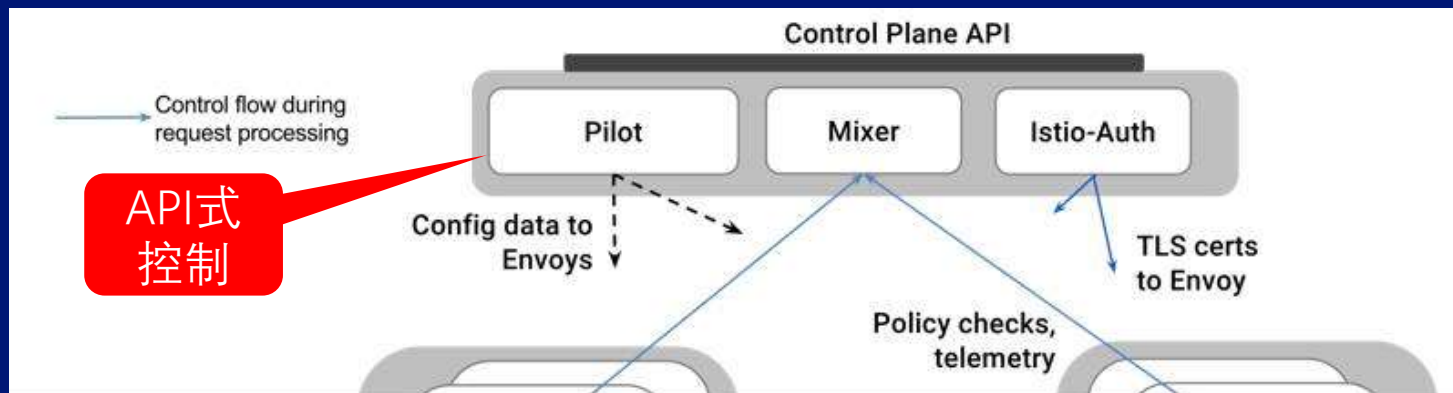
Kubernetes - 船长



仰望Kubernetes的格局：

- 分工比实现更重要：模块化、声明式
- 服务的类型抽象：几种Controllers分类清晰
- 存储的抽象：Volumn、PV等多层的描述
- 监控的实现：侵入式部分和非侵入式部分的分工

Istio - 启航



展望Istio的格局：

- Service Mesh各部分理解
- 服务调用RPC关键作用
- Sidecar解耦合职责作用
- 控制面板与数据面板分工

RPC

Sidecar

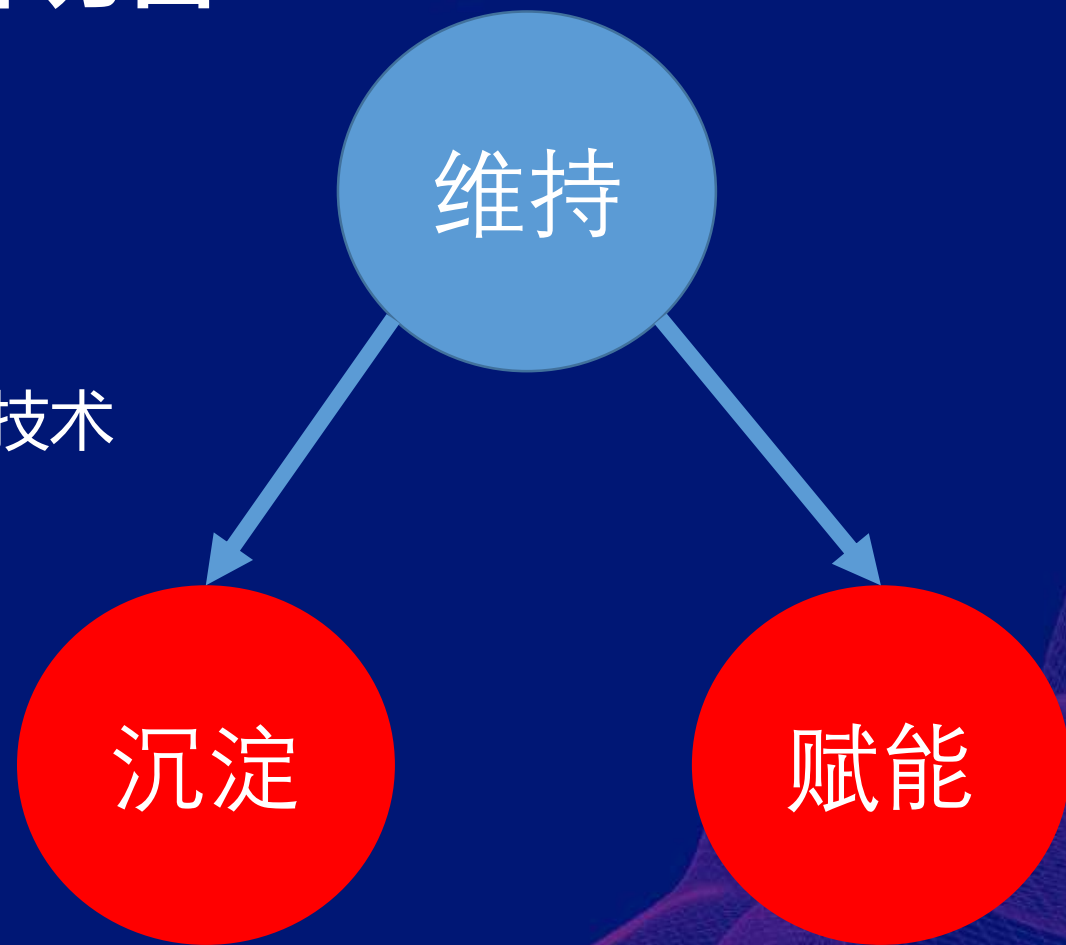
第三部分

架构技术积累的几个方面

架构技术积累和发展架构技术几个方面

三步走，先生存，再发展：

- 第一步：维持，分层架构中、全流程覆盖
- 第二步：沉淀，从业务抽象较基础的架构技术
- 第三步：赋能，用策略架构促进业务



第一步：维持

维持架构对业务的基本支撑：

- 架构的承载能力：量级、复杂度、扩展性
- 基本分层，必不可少
- 开发、发布和维护全流程



业务层

服务层
(中间层)

基础架构和组件

研发
工具
和
流程

第一步：维持，让架构的良性运作

开发

代码

接口设计

服务使用

服务提供

联调

模块分工

发布

构建

部署

分级发布

回滚方案

上线应急预案

调度

流量切换

容量管理

监控

日志监控

拓扑跟踪

资源与环境

服务框架

CI/CD体系

沙盒环境

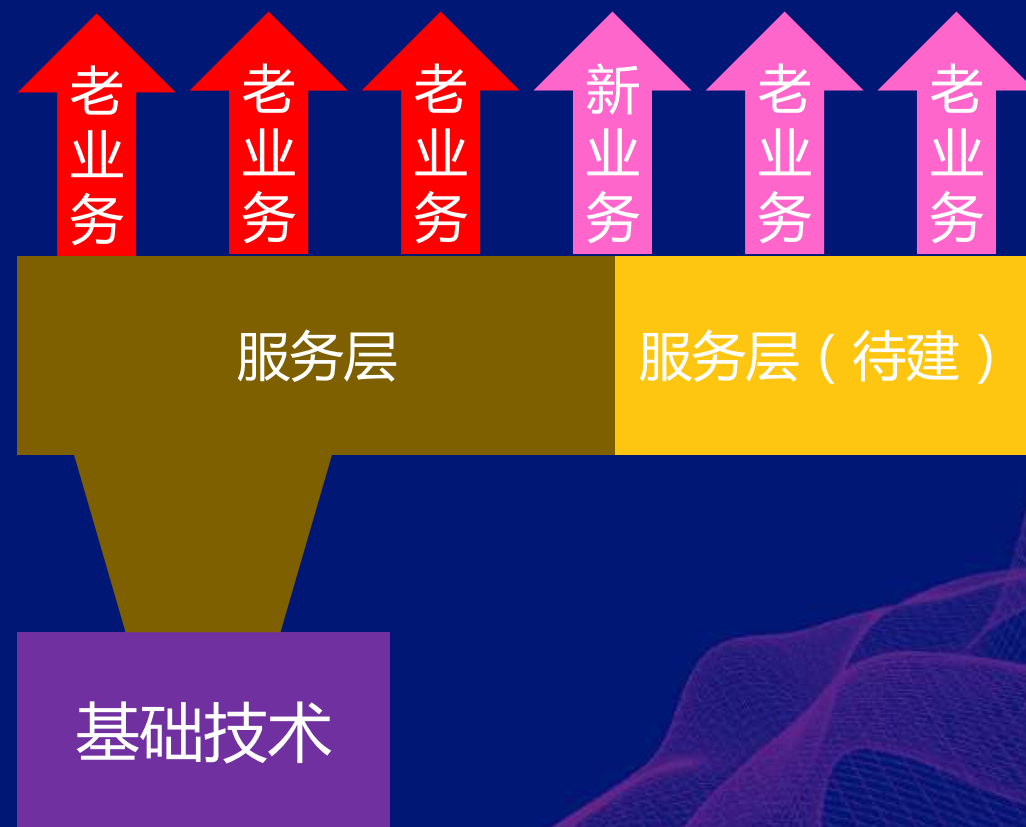
资源评估

机器容器管理

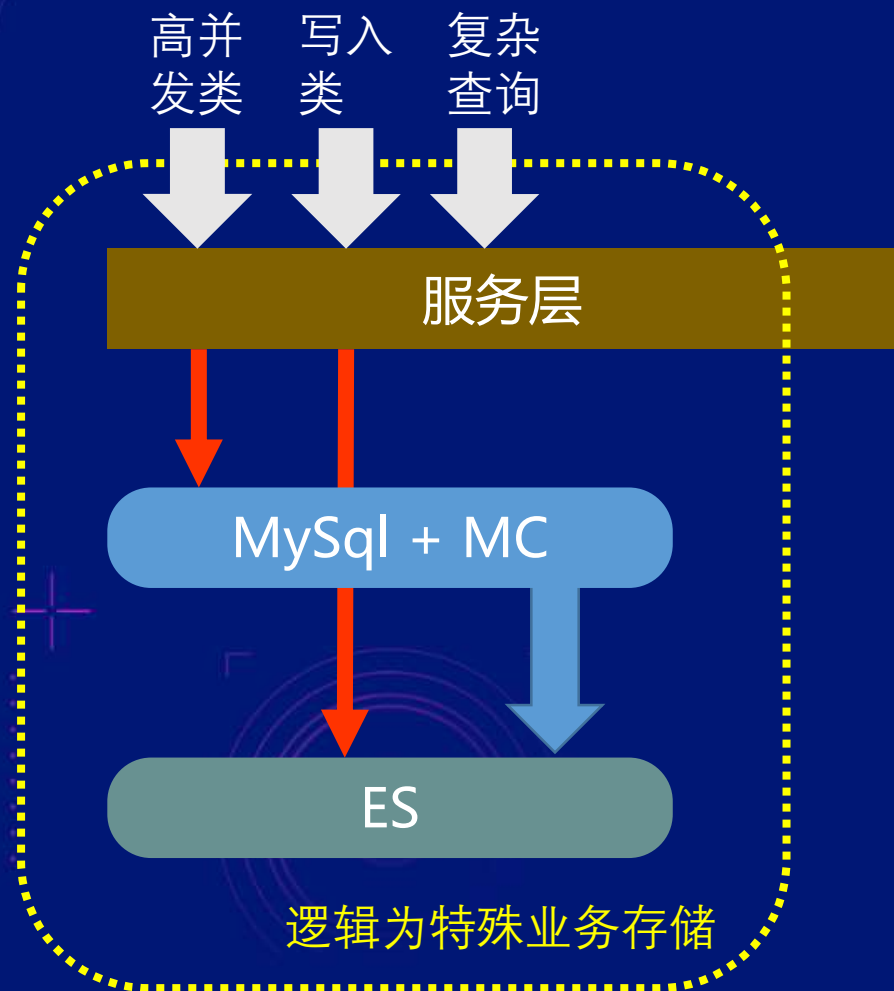
第二步：沉淀

从业务中沉淀基础技术的几个方面：

- 架构是沉淀出来的，不是规划出来的
- 从业务向下沉淀服务层，可更好支持业务
- 从服务层向下沉淀基础技术
- 【理念】收益要综合评估
- 【理念】服务化是大方向，细节看场景
- 【理念】服务化可以比业务实现稍慢
- 【理念】结构化完备才能沉淀基础技术



沉淀技术：做好数据存储的服务化

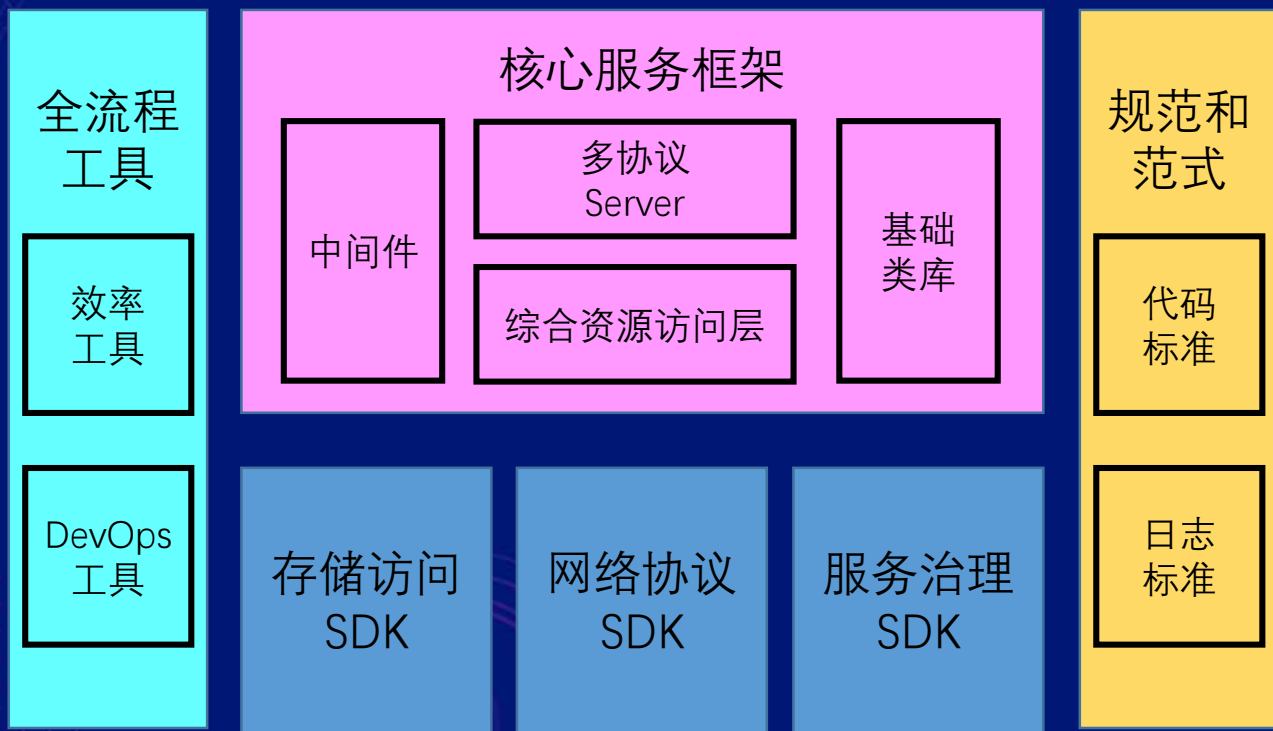


业务对于数据的多方面需求：

- 高并发、高写入的C端类功能
- 复杂查询、搜索类的B端类功能
- 数据平台类的多key查询需求
- CAP中的C和A的平衡

理念：避免分裂、优化索引、平衡重和丢

沉淀技术：GO体系建设成果



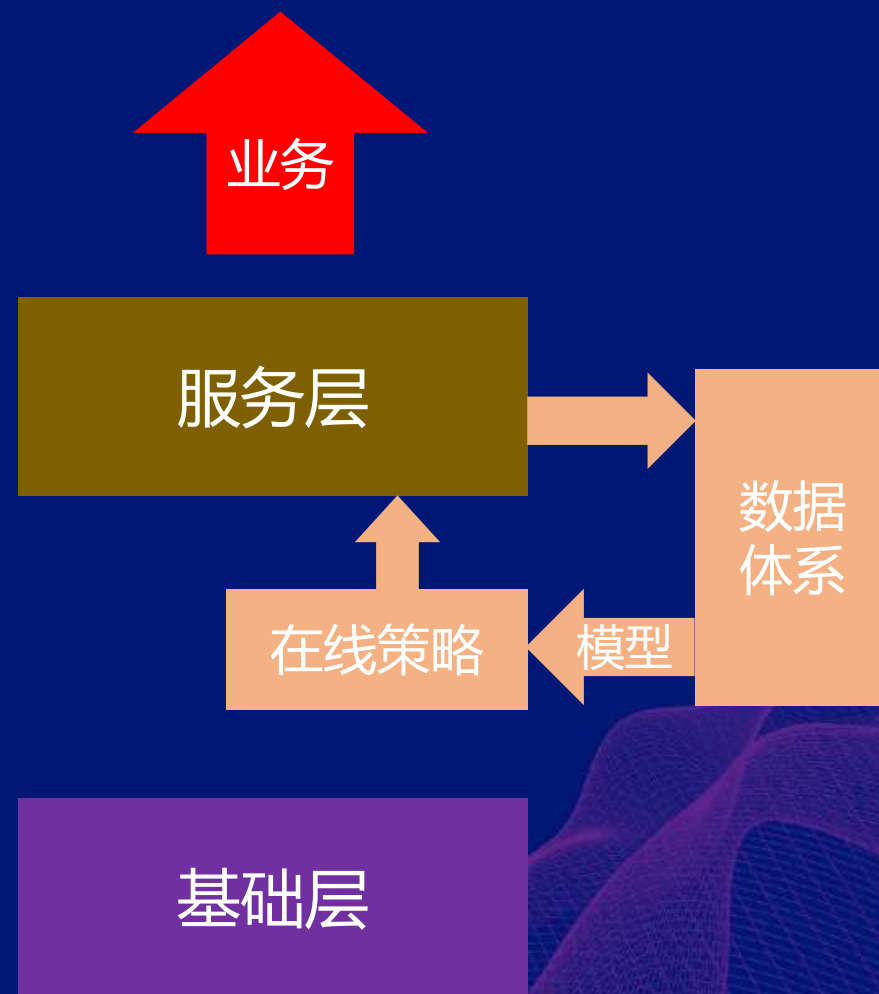
GO体系技术建设的特点：

- 需求：性能、资源开销、迭代效率
- 有利条件：流量大、新业务多
- 实现原则：与PaaS必须要解耦、DevOps配套要健全
- 实现方法：框架、脚手架、包管理、代码工具

第三步：赋能

赋能的意义在于，让技术驱动业务指标的提升：

- 业务建模至关重要
- 规则、非闭环阶段可能是必经之路
- 数据体系一开始就很重要，往往开始做策略闭环才知道其中的缺陷
- 最终策略闭环：数据 -> 模型 -> 在线



本PPT来自2018携程技术峰会
更多技术干货，请关注“携程技术中心”微信公众号

