云瀚

加速开放新趋势

云原生下图书馆的应用模式

2022.03.03

上海阿法迪智能数字科技股份有限公司

演讲人：陈祖乐

目录 CONTENTS

云瀚

1 多租户

三大特征
两种隔离模式

多租户

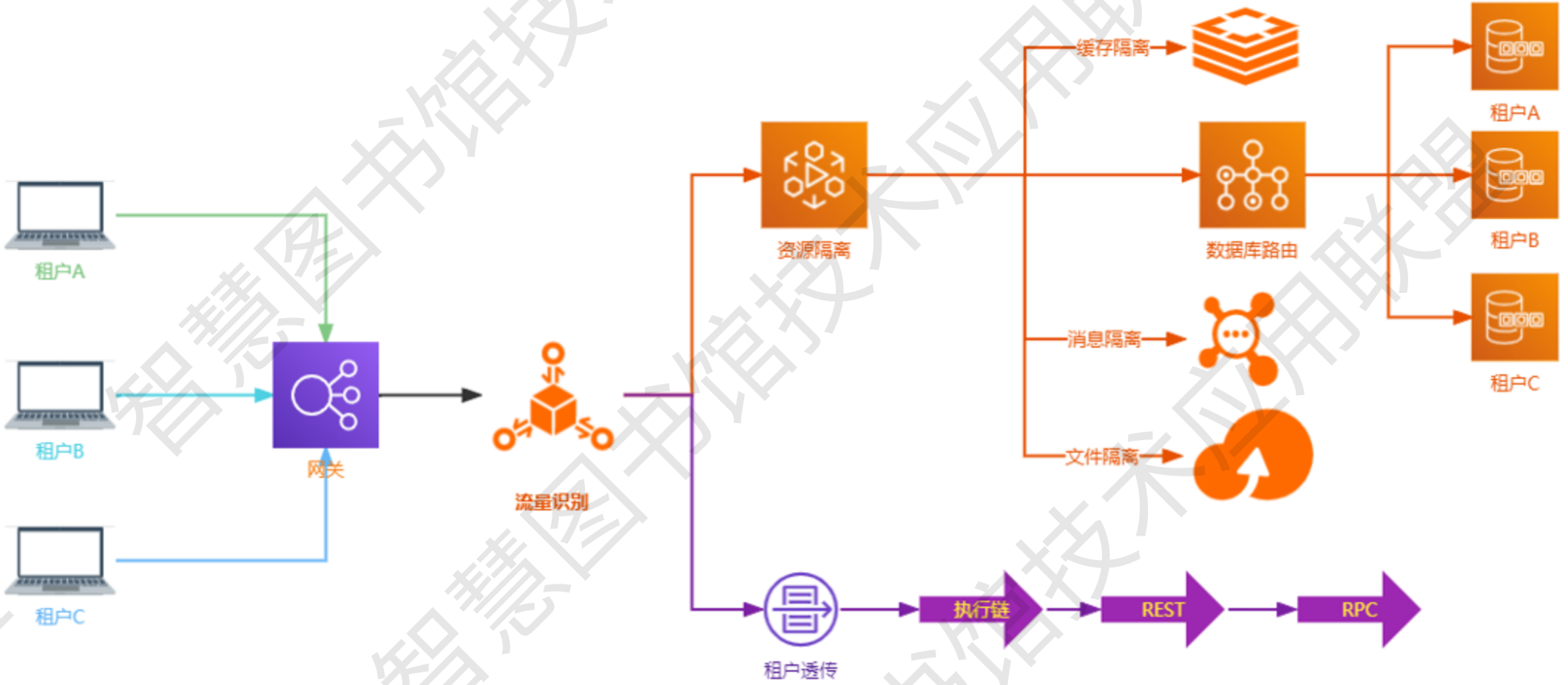
Multiple tenants

流量识别：识别请求来自于哪个租户

租户透传：执行链传递、跨线程传递、Rpc传递、消息传递

资源隔离：缓存、消息、检索、数据库

多租户：流量识别、租户透传、资源隔离



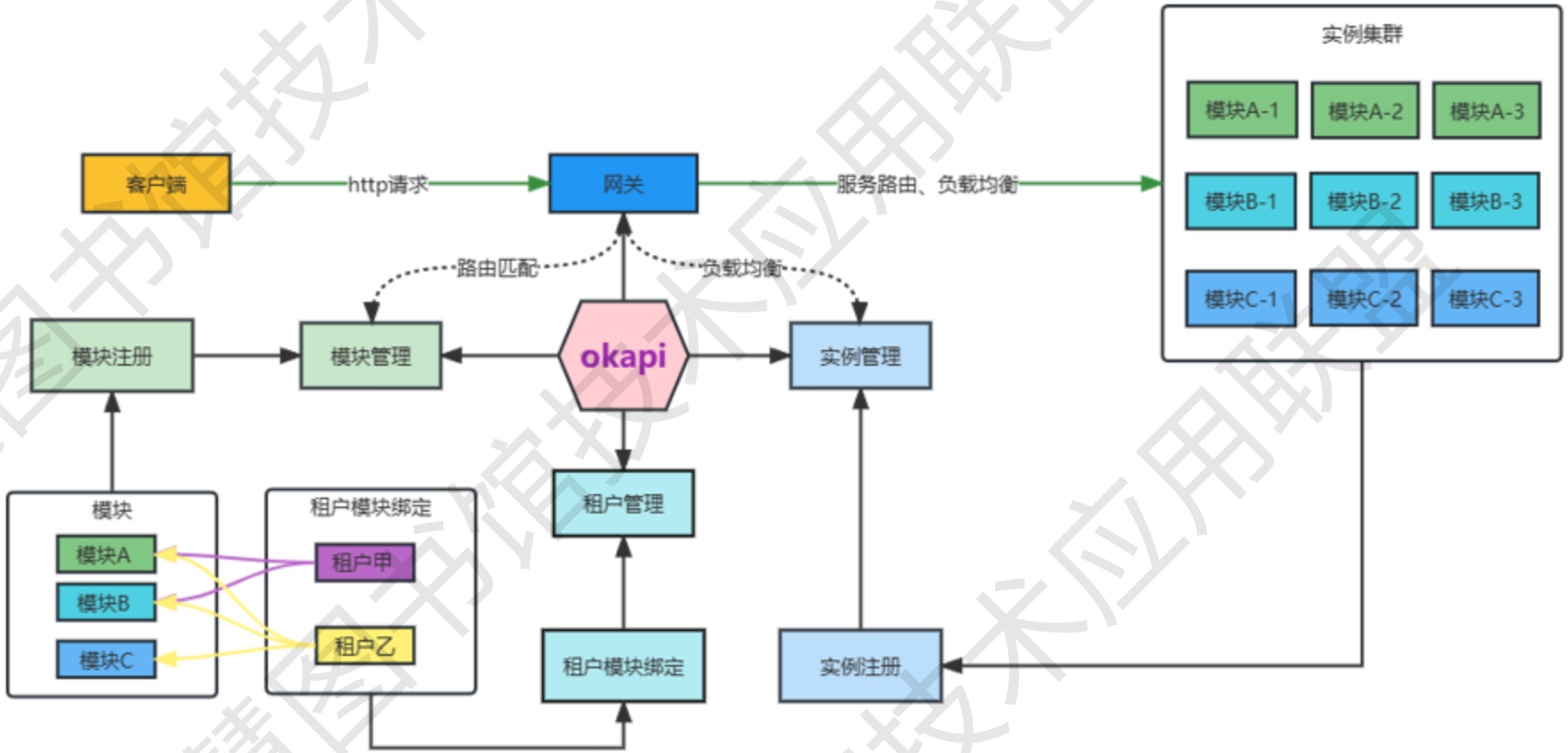
现在

基于应用隔离

资源利用率高

客户成本更低

架构简洁易用



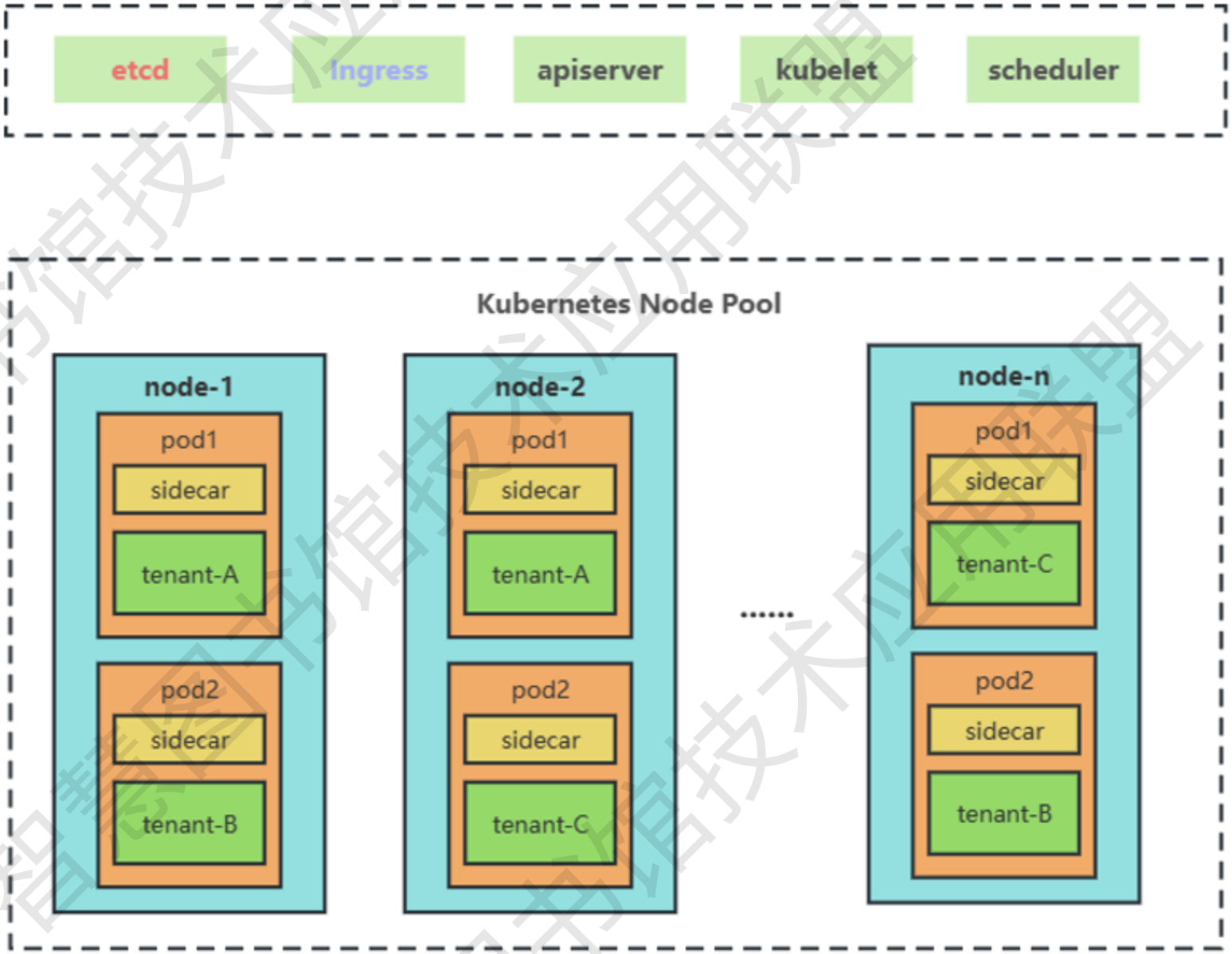
未来

基于容器隔离

隔离性更好

安全性更高

极致扩缩容



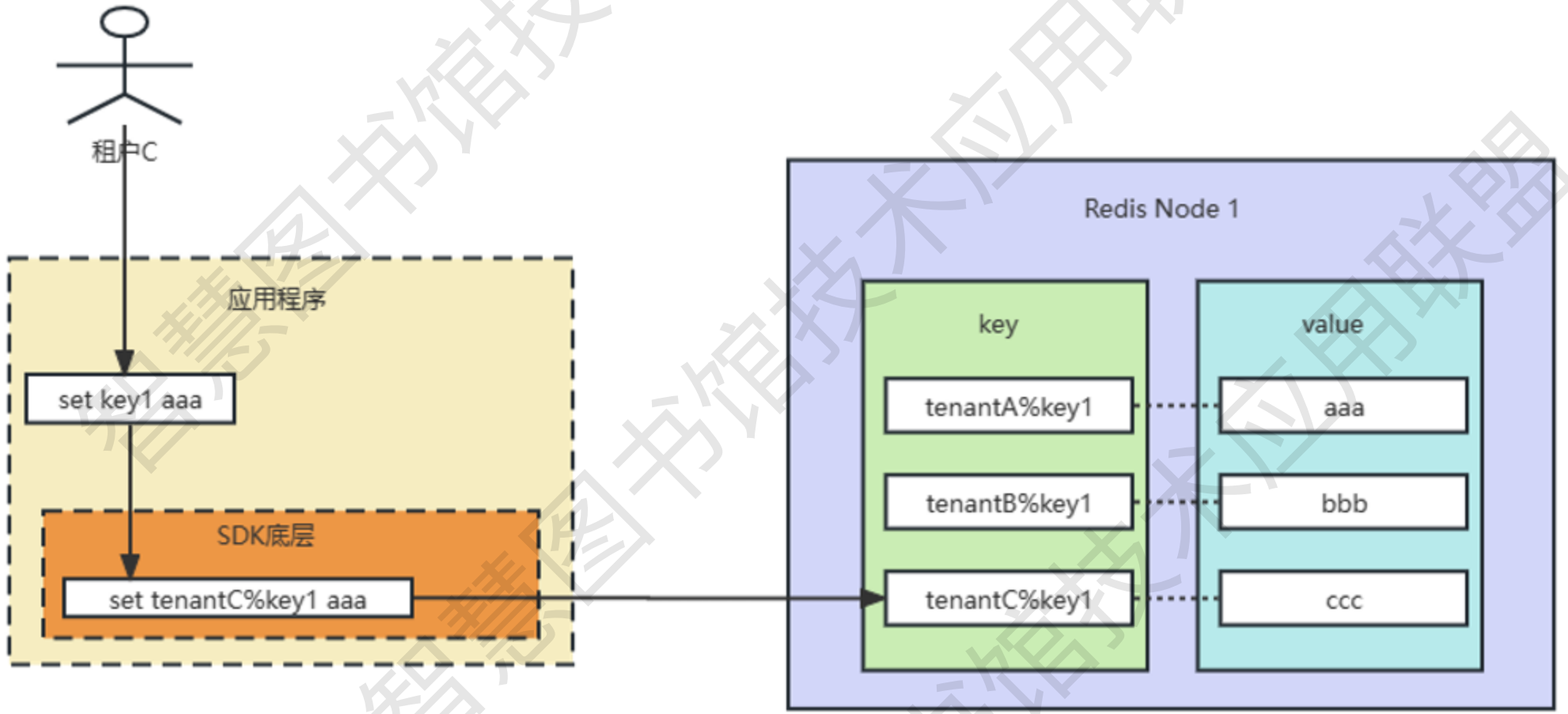
方案对比：应用 VS 容器

云潮

	应用层隔离	容器隔离
隔离模式	线程	pod（进程）
实例:租户	1:N	1:1
租户识别模式	http header、线程上下文	无须识别（流量识别下沉到网关）
隔离性	一般 (cpu、内存共享)	完全隔离
SDK	重量级	轻量级
资源利用率	极高	中等
开发难度	高	低
系统复杂性	中	高

Redis多租户方案-namespace

Redis multi-tenant scheme-namespace



数据库隔离模式

字段隔离

优点：数据库资源利用率高，全部sass化，运维友好

缺点：ORM层改写实现成本高、引入分表场景，架构更复杂、数据混合管理风险高

表隔离

优点：数据库资源利用率高，全部sass化

缺点：ORM层表映射改写实现成本高、表结构升级难、数据混合管理风险略高

库隔离

优点：数据完全隔离、实现方案简单、允许公有云、混合云等多种模式

缺点：资源利用率稍低、运维成本高

云瀚

2 微服务

架构原则
应用架构
组件架构

云瀚架构原则

Principle of Component & middleware architecture

生态化

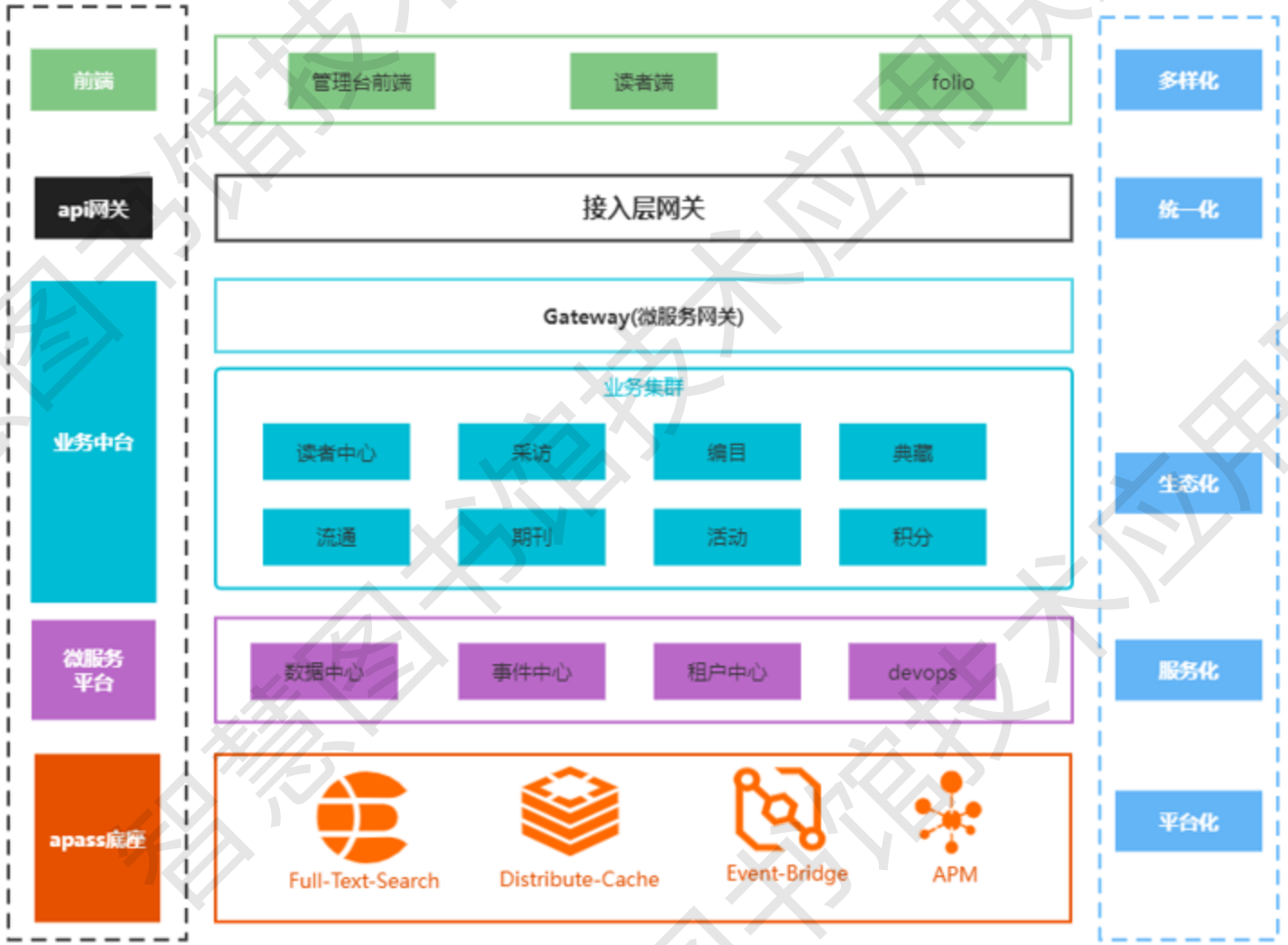
可插拔

可替换

- **生态化**：组件选型时优先考虑生态完备性，以减轻后续的适配工作，发挥组件的最大效能。比如云瀚优先选择spring-cloud-alibaba生态组件
- **可插拔**：根据客户实际情况，可以选择完整部署、部分部署和不部署，比如限流熔断组件sentinel、可观测性链路组件skywalking、配置中心nacos等
- **可替换**：根据客户系统现有组件选型，做到基础组件可更替，比如对象存储可选Minio、OSS商业版或者本地磁盘存储等多种存储方案，比如健康监控可选skywalking或者Prometheus等

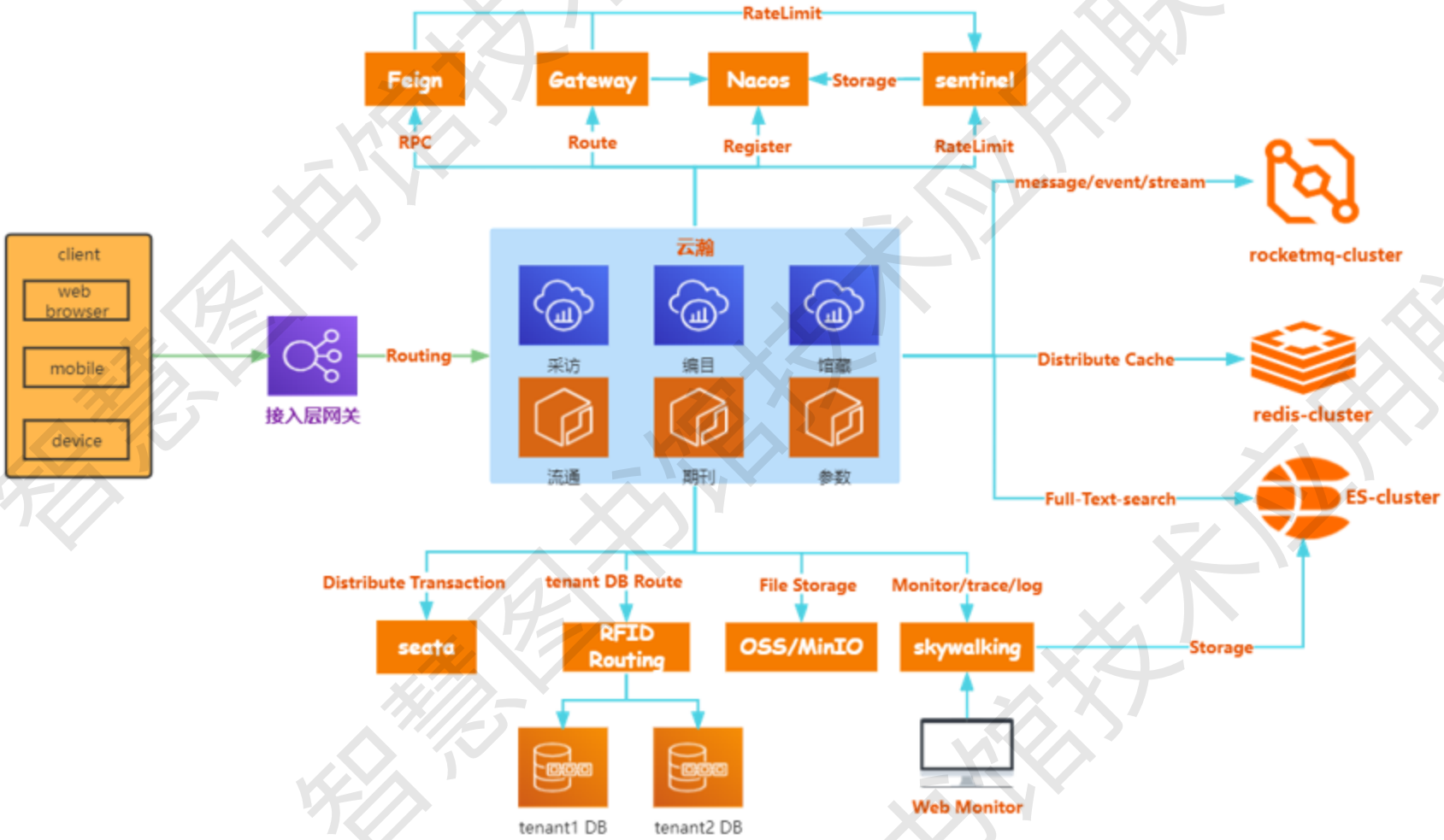
应用架构

Application architecture



组件架构

Component & middleware architecture

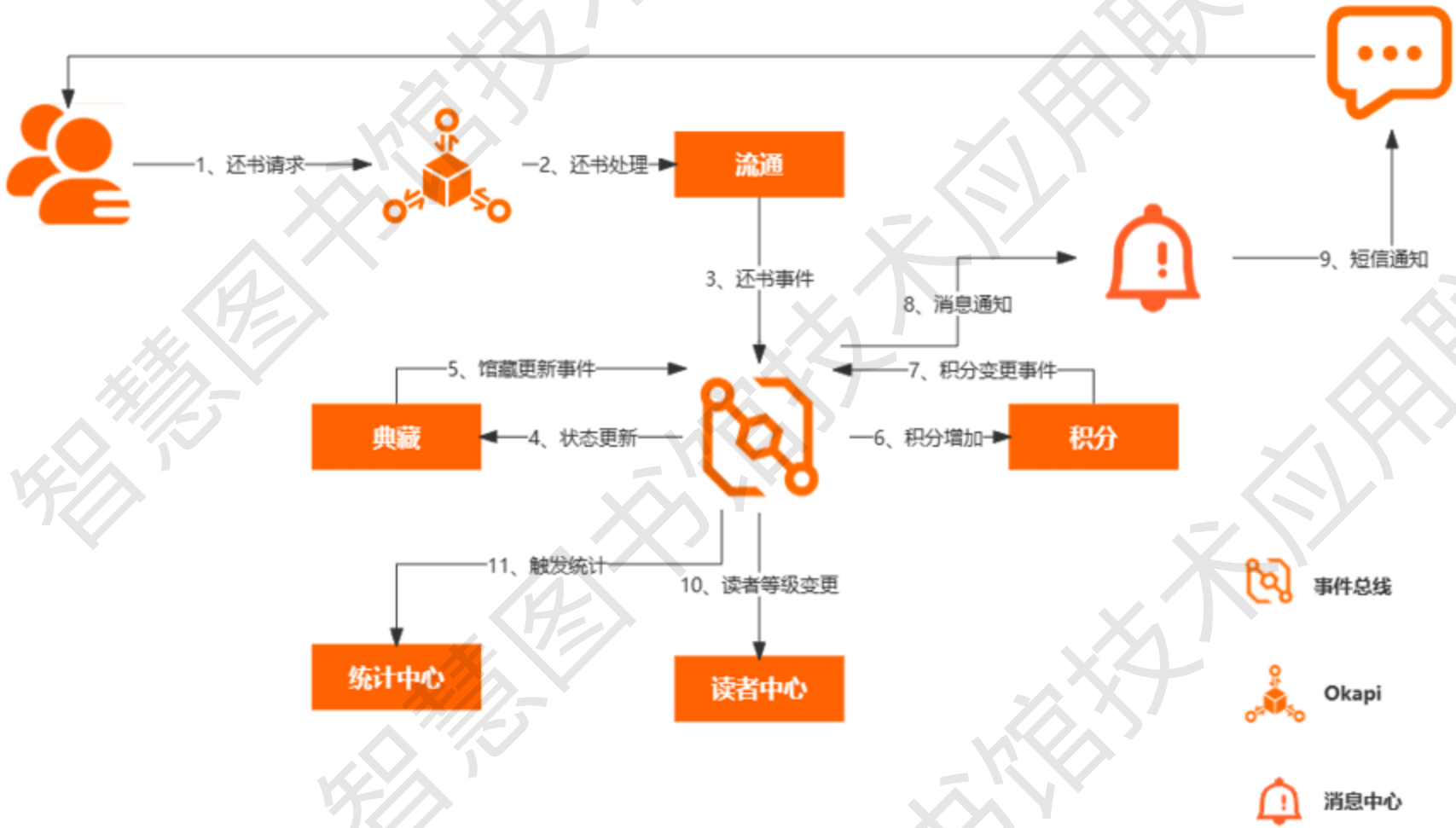


云瀚

3 云平台

事件驱动-EventBridge
存算分离-pulsar
缓存集群-redis

事件驱动-以读者还书为例



存算分离-pulsar

下一代消息平台

persistent://tenant/namespace/topic

broker-计算层

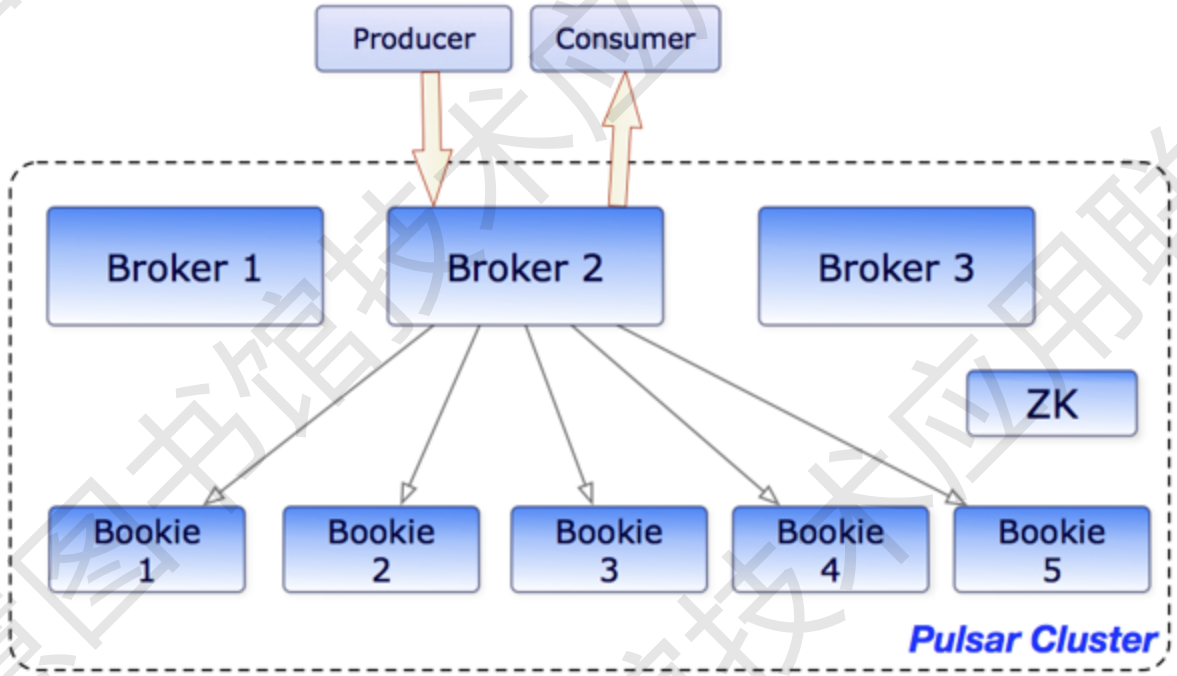
Bookie-存储层

横向扩展

动态扩容

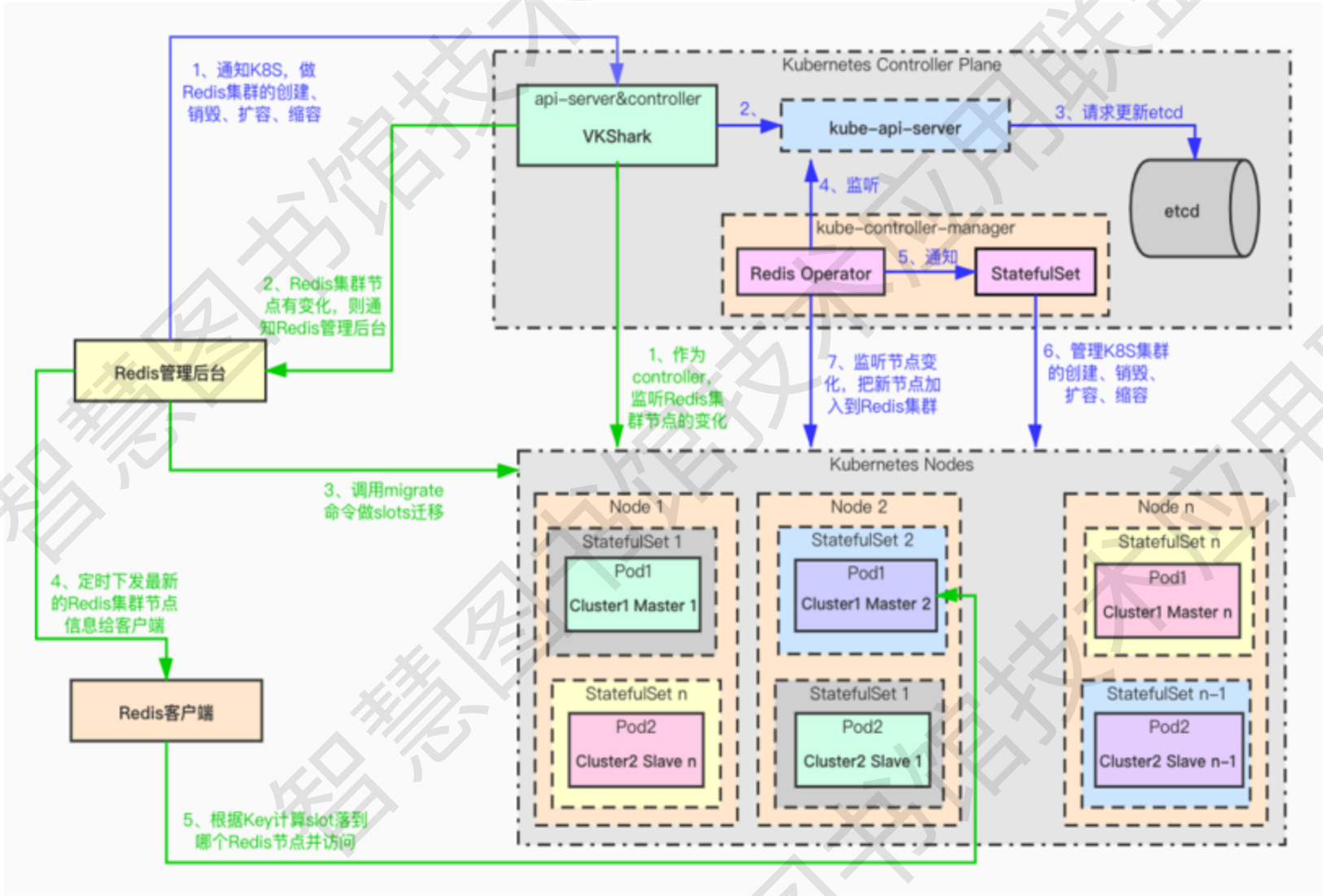
租户管理

消息、事件、流



Redis容器云架构

Redis container cloud architecture

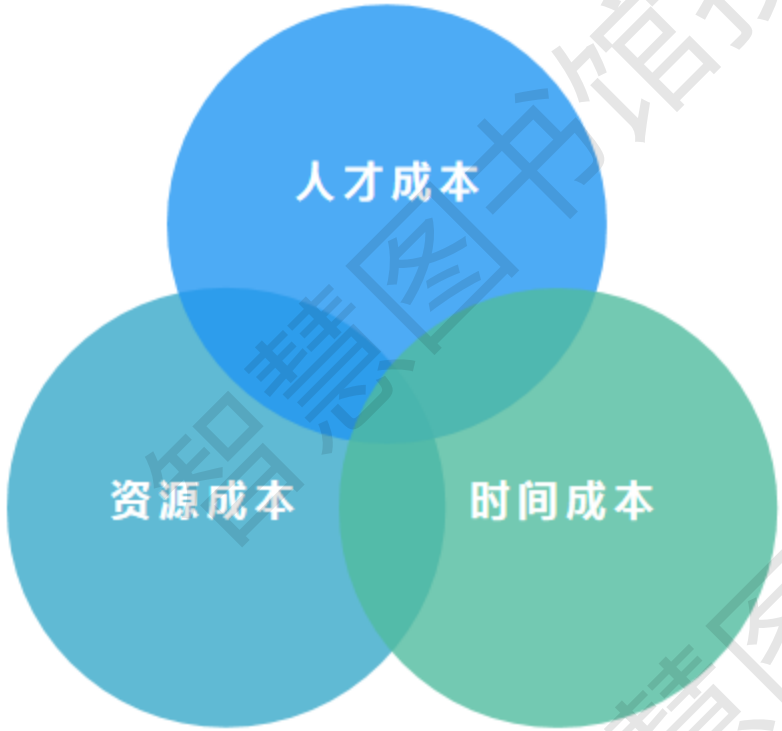


云瀚

4 云厂商

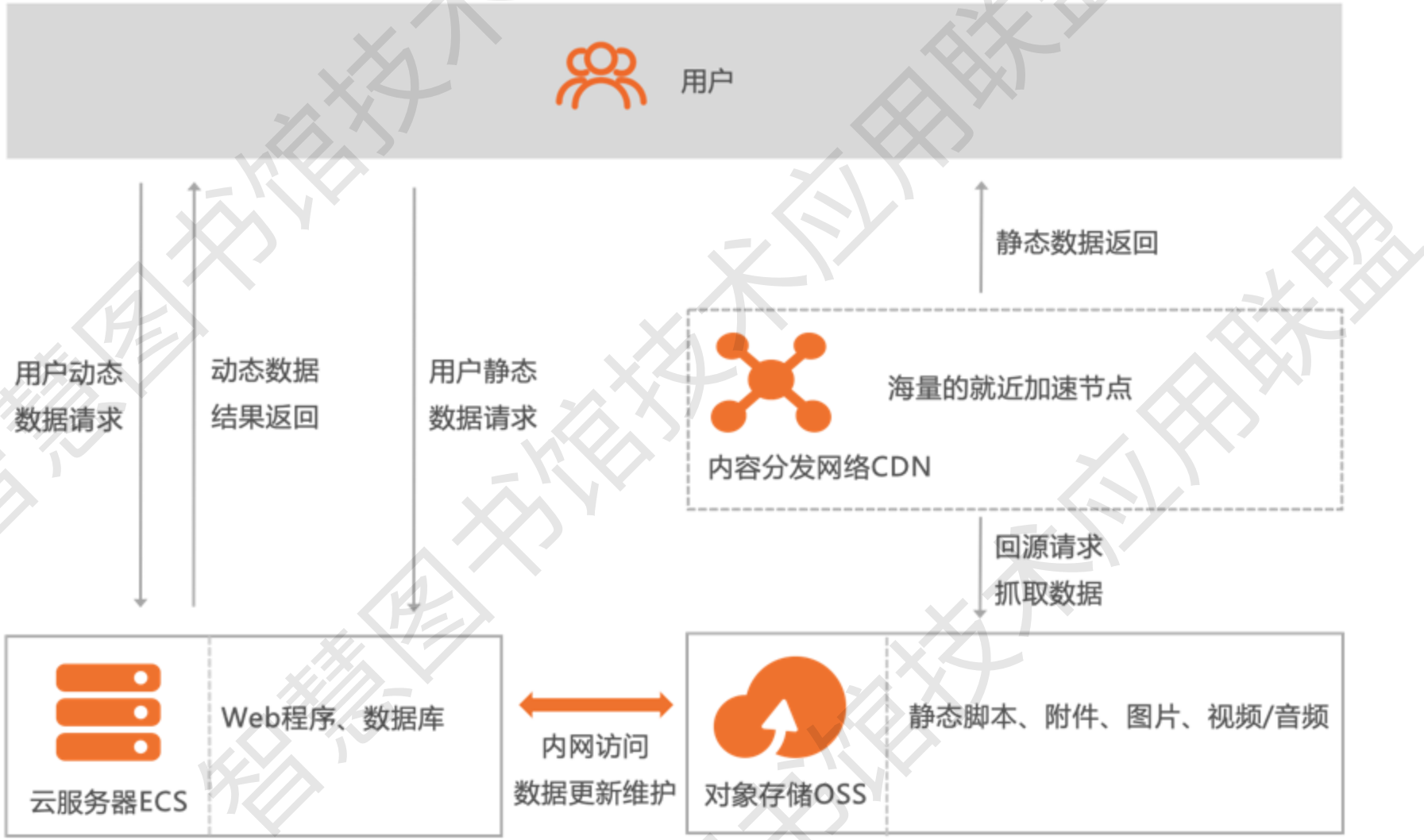
自研云平台困境
自研 VS 商业
对象存储案例

自建云平台困境



- 云平台对研发团队要求高，行业内很难组建相关团队
- 基于最大访问冗余，很难实现资源最大化利用，硬件成本高昂
- 技术沉淀速度无法追赶业务发展的诉求，时间成本导致机会流失

对象存储方案



对象存储方案对比：自建 VS 商业版

云潮

自建

商业

1T 5年

费用高昂

最多可降低
95%

费用低廉

成本
按量计费、弹性资源、0运维

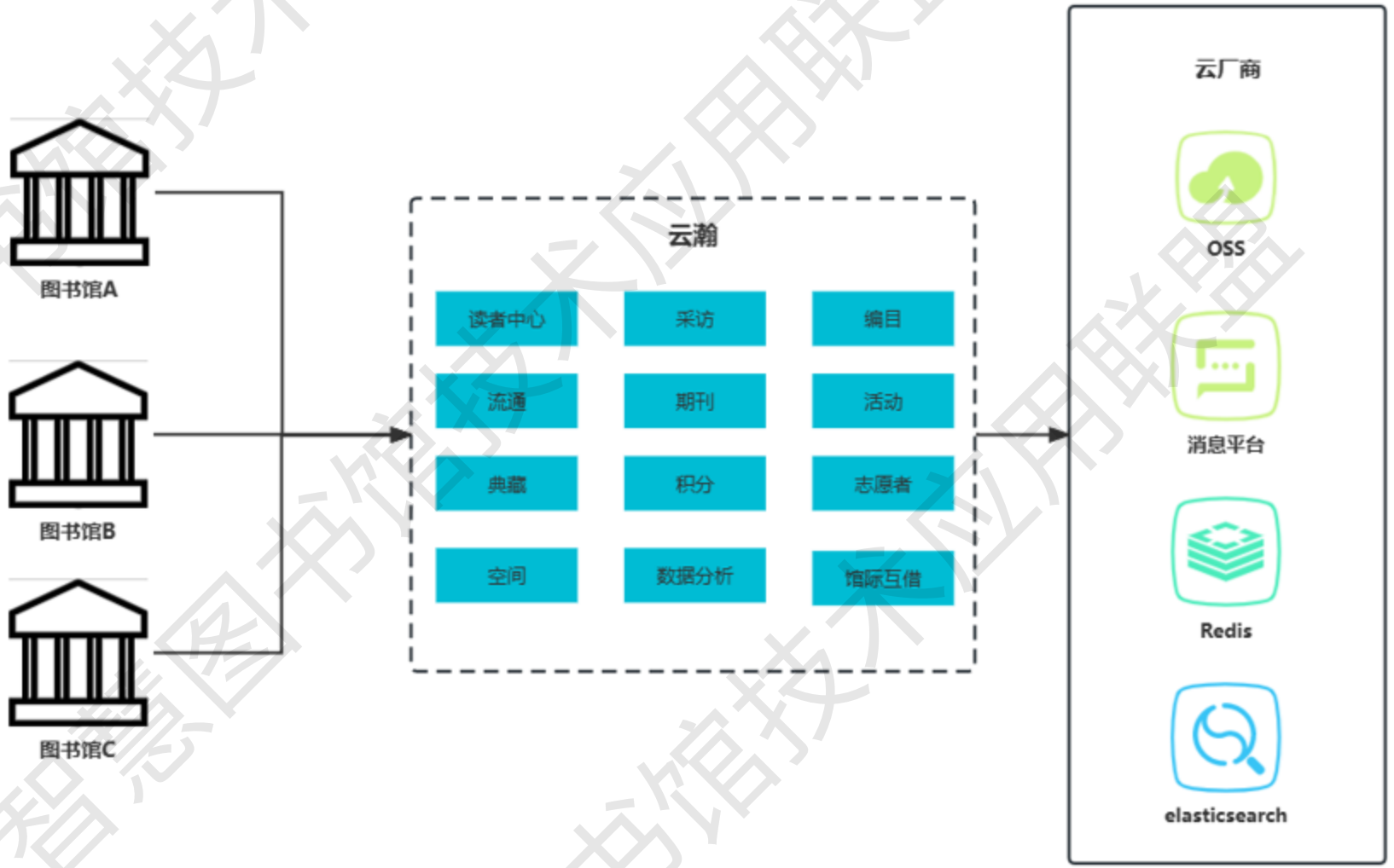
安全
资源隔离、访问限制

完备
容量、权限、容灾、网络

图书馆、云瀚、云厂商

云瀚

云瀚可作为SASS层，
为图书馆提供业务价值；
云厂商提供IASS、
PASS、APASS层支持，
保障系统稳定、安全性，
降低图书馆运维成本



简化部署方案

云潮

图书馆自建服务器

（私有云/政务云/本地服务器）

传统部署

数据库、应用、缓存服务（如redis）、检索服务（如ElasticSearch）、消息队列（如kafka、rocketmq等）、对象存储服务（如MINIO）等公共组件进行逐个安装。

传统方案

图书馆自建服务器

（私有云/政务云/本地服务器）

容器化部署

数据库、应用、缓存服务（如redis）、检索服务（如ElasticSearch）、消息队列（如kafka、rocketmq等）、对象存储服务（如MINIO）等公共组件。

简化方案一

云厂商

缓存服务（如redis）、检索服务（如ElasticSearch）、消息队列（如kafka、rocketmq等）、对象存储服务（如MINIO）等公共组件。

图书馆自建服务器

（私有云/政务云/本地服务器）

数据库、应用等

简化方案二

云瀚

谢谢

THANKS