

阿里的

Kubernetes测试环境[✓]工具箱

开源

云效说码

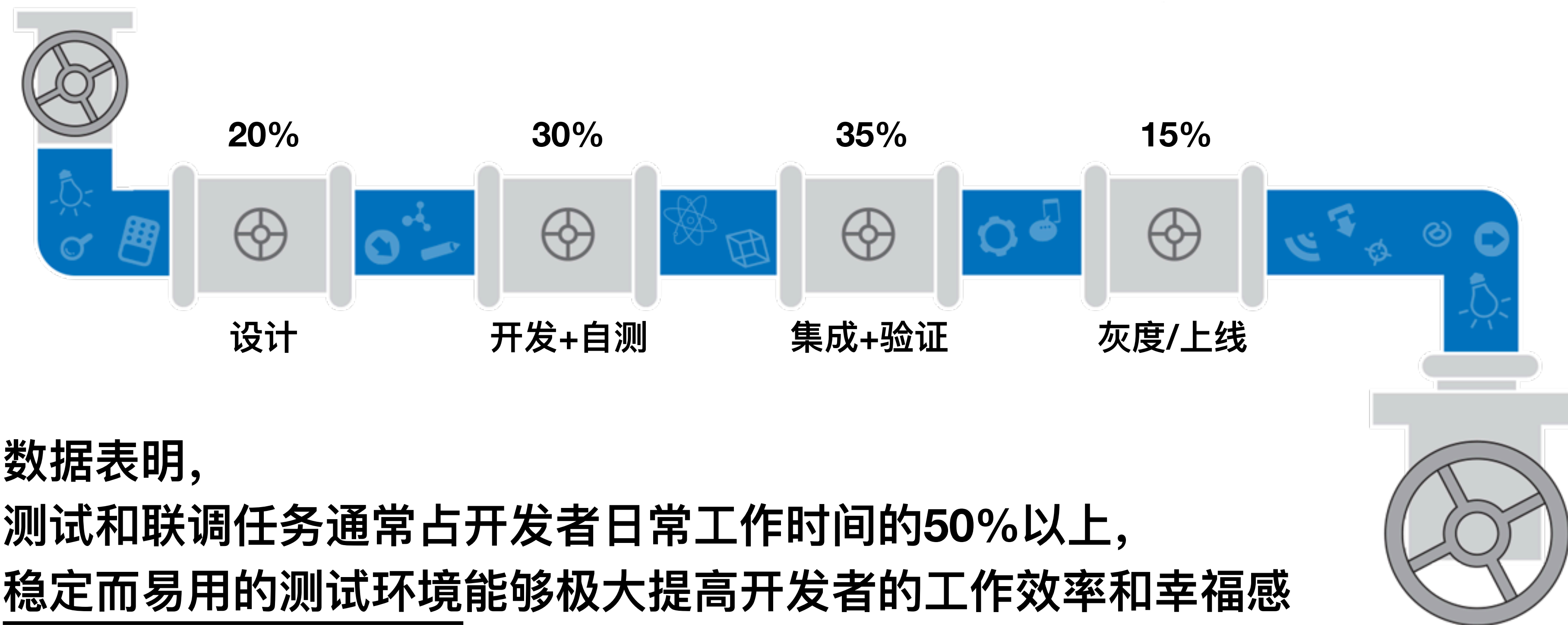
课程提纲

第一期：阿里的测试环境管理实践概述

第二期：单人开发场景下的测试环境实践

第三期：多人协同场景下的测试环境实践

为什么要关注测试环境



理想的测试环境

本地直连，双向访问互通

独占式随意使用，可重启，可调试，可断点

多人随意组合协作，多服务联调不串流量

多个依赖服务版本同时在线，想测哪个即时切换

... ..

自由连接

随时可用

互访可控



现实中的测试环境

本地无公网IP，测试环境Pod使用Cluster IP，本地/集群双向均不通
所有开发者共用测试环境，互相干扰、调用和消息互串
拉起一套独立专用环境费时费力，资源成本难以招架

....

无法直连

稳定性差

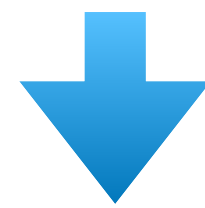
流量混杂



满脸都写着高兴

阿里的测试环境实践

扁平的内网IP + 项目环境

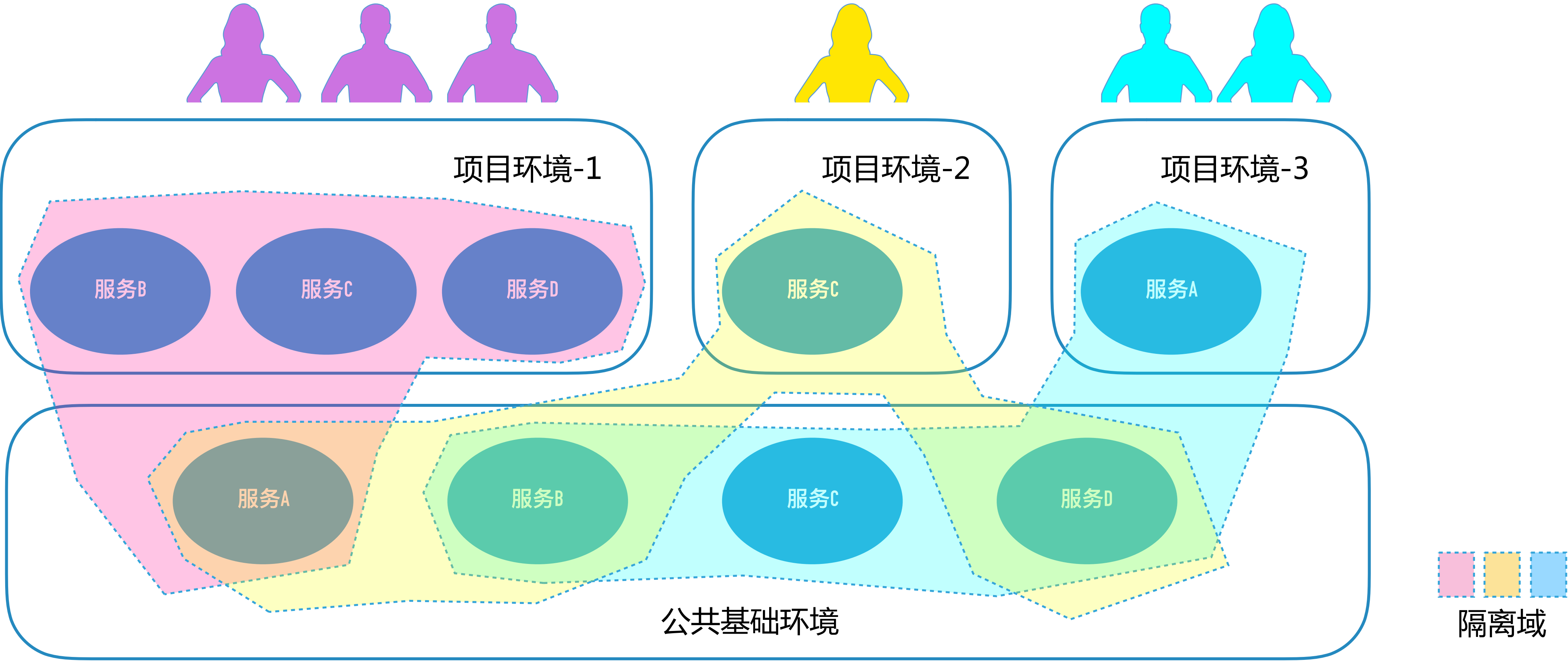


基于CNI机制改造Kubernetes的IP分配逻辑^[1]

基于RPC/消息中间件的虚拟环境

[1] 参考 <https://github.com/containernetworking/cni/blob/master/SPEC.md#ip-allocation>

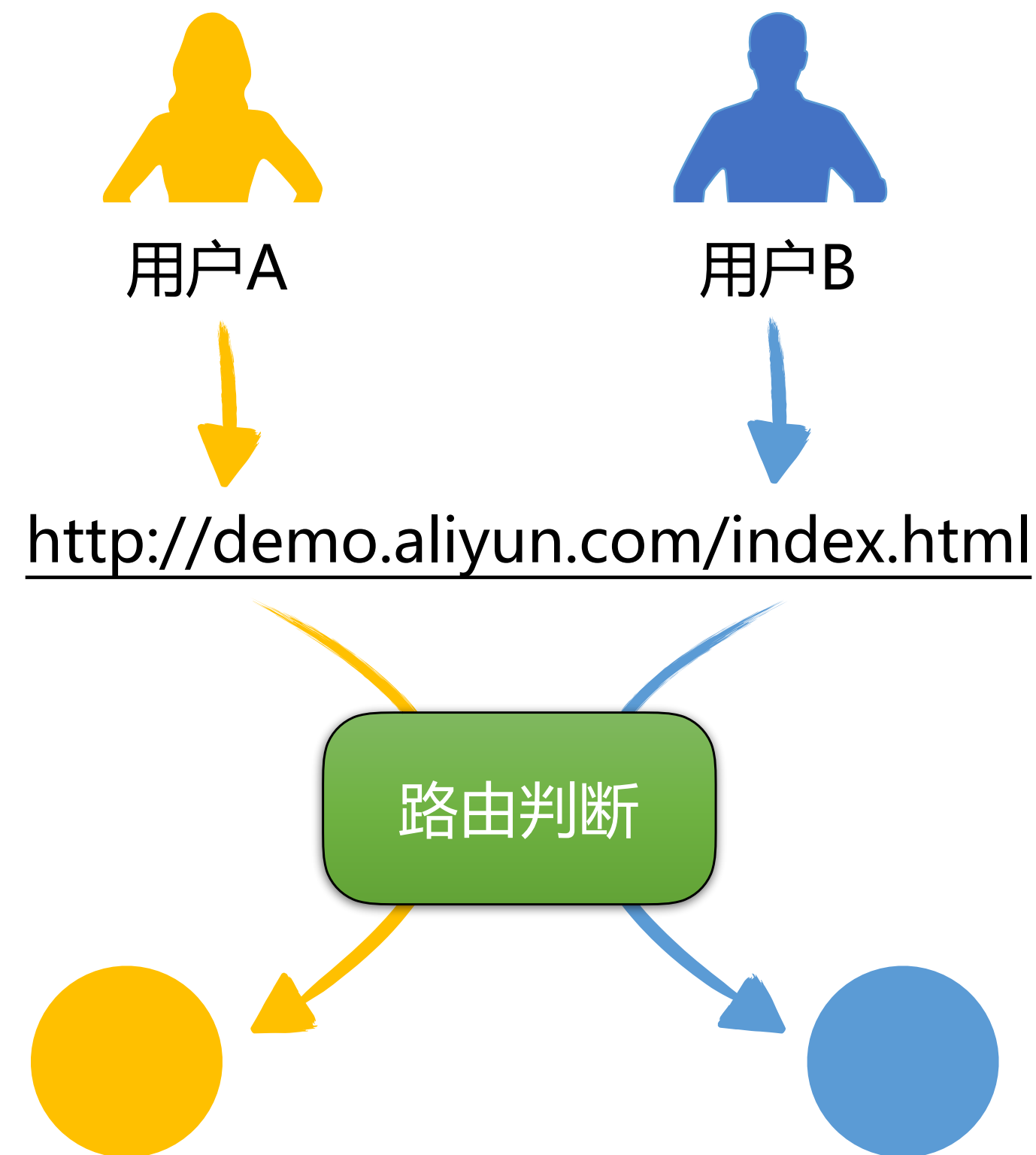
环境复用：解决资源效能问题



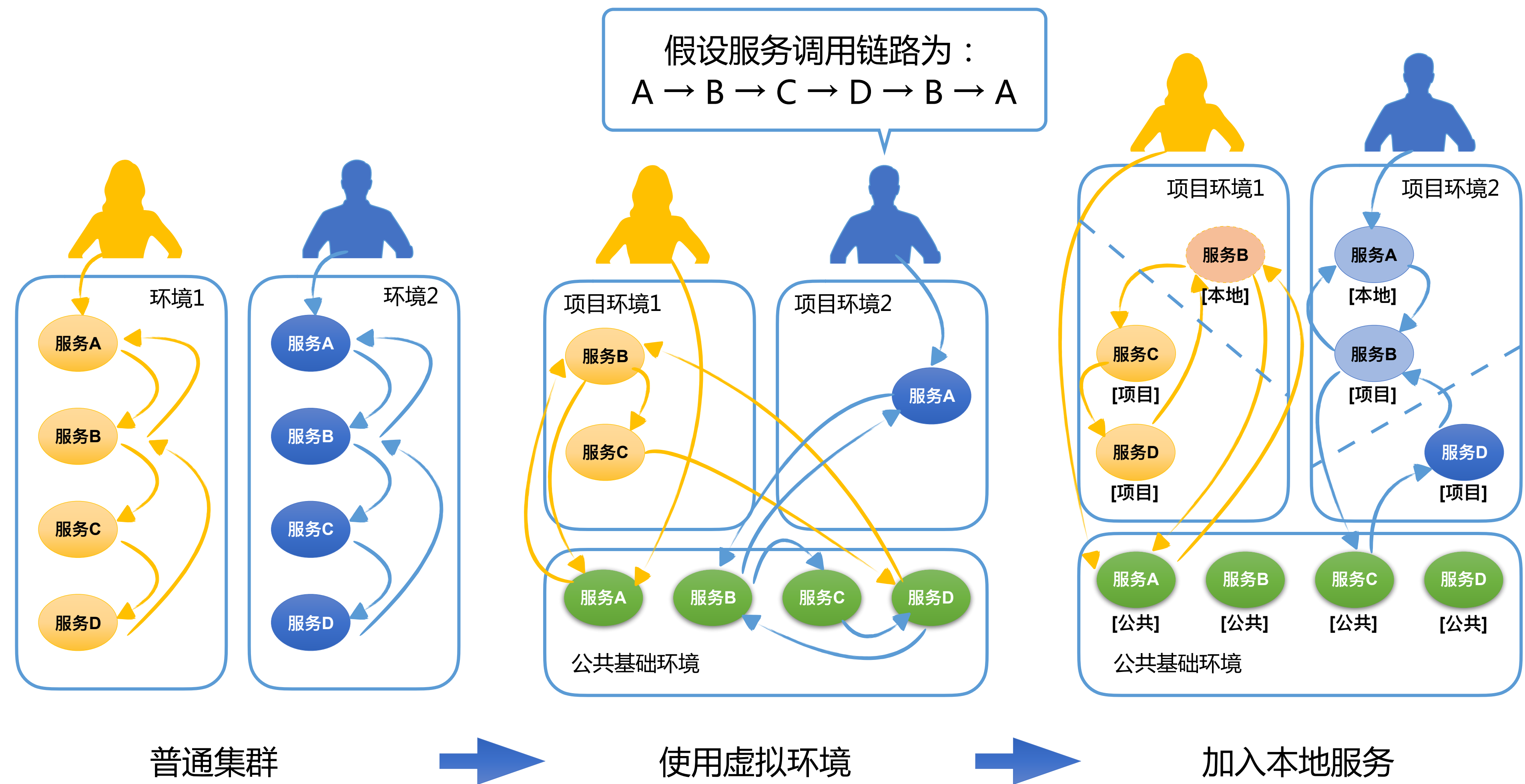
本质：基于调用发起方身份的路由控制

[类似的场景]

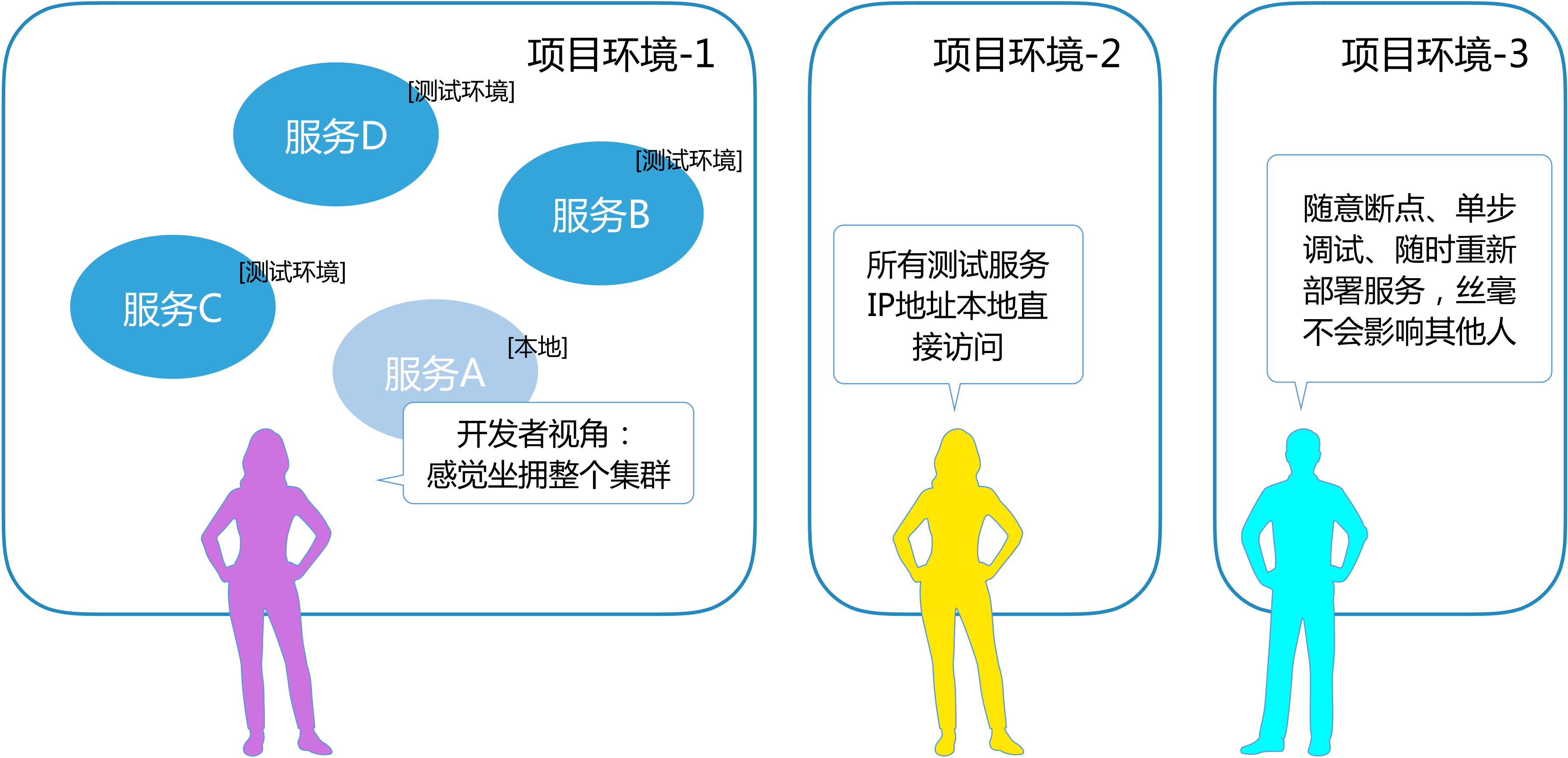
- 金丝雀发布
- 灰度部署
- A/B测试
- ...



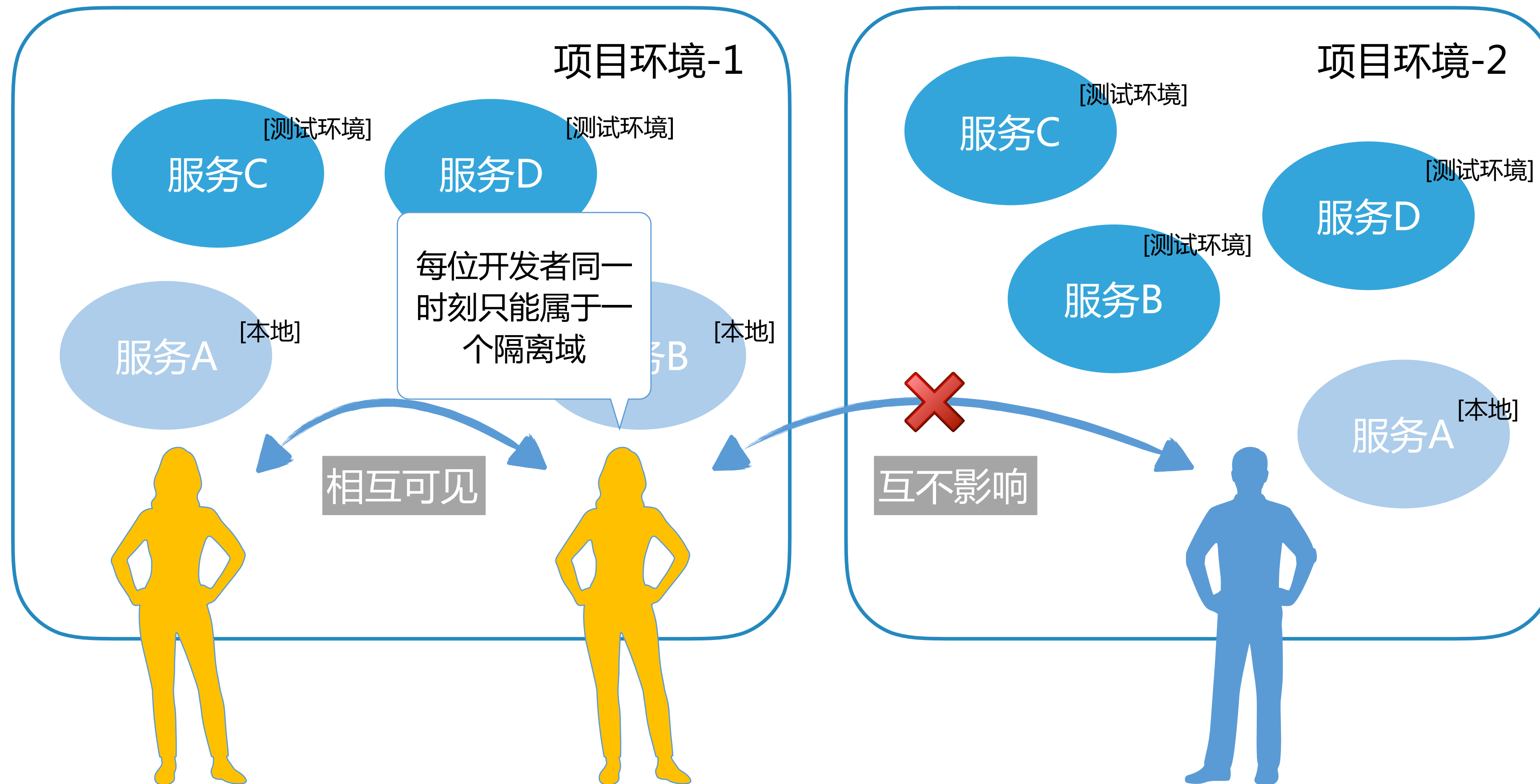
效率：本地服务加入联调



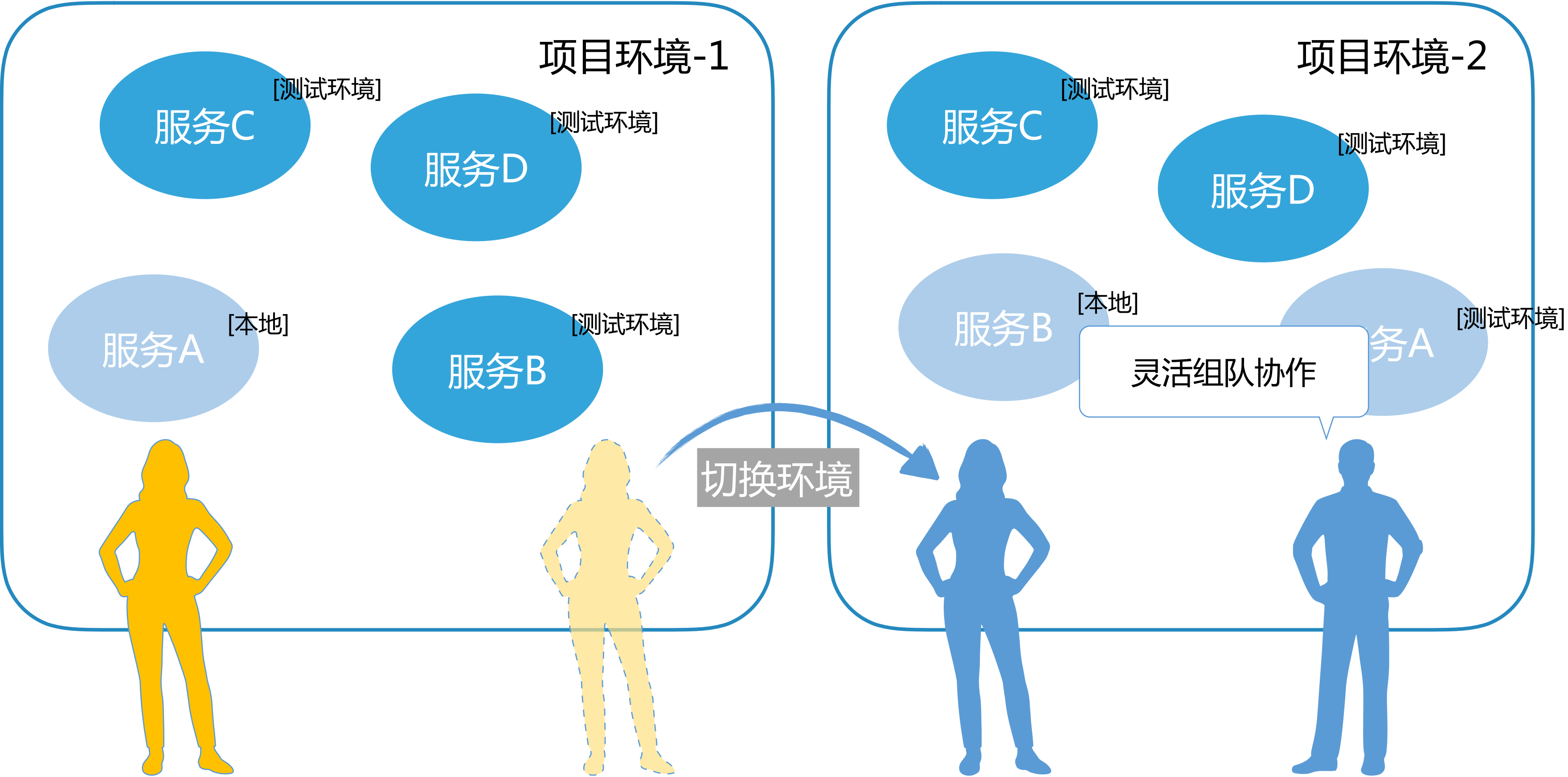
极致的测试环境：开发者视角



极致的测试环境：多人协作



极致的测试环境：自由自助



网络改造：本地与集群互联互通

- 内部方案

Kubernetes生态的开放性（CNI与IPAM）

适配企业网络地址规划

- 开源方案

► 换一个思路，更轻量的方案：KtConnect

Supported by 阿里云·云效

路由控制：项目环境与隔离域

- 内部方案

基于特性分支创建项目环境

基于项目环境和调用者身份建立隔离域

基于中间件实现RPC和消息的虚拟路由控制

- 开源方案

► 换一个思路，更轻量的方案：KtConnect + KtVirtualEnvironment

Supported by 阿里云·云效

THANKS

下期内容预告：

- 单人开发场景下的测试环境实践



钉钉扫码进群



[云效开发者交流群]



[Kt工具用户群]