

云原生技术发展史与云开发架构设计

讲师：周俊鹏



目录



1. 云原生发展史
2. 传统研发模式的问题
3. 云开发模式的架构演进
4. Q&A



1 云原生发展史

生而为云

云原生的发展历程

- ✓ 2004 年 ~ 2007 年，Google 已在内部大规模地使用像 Cgroups 这样的容器技术；
- ✓ 2008 年，Google 将 Cgroups 合并进入了 Linux 内核主干。
- ✓ 2013 年，Docker 项目正式发布。
- ✓ 2014 年，Kubernetes 项目也正式发布。
- ✓ 2015 年，CNCF 成立。
- ✓ 2017 年，CNCF 达到 170 个成员和 14 个基金项目。
- ✓ 2018 年，CNCF 成立三周年有了 195 个成员，19 个基金会项目和 11 个孵化项目。

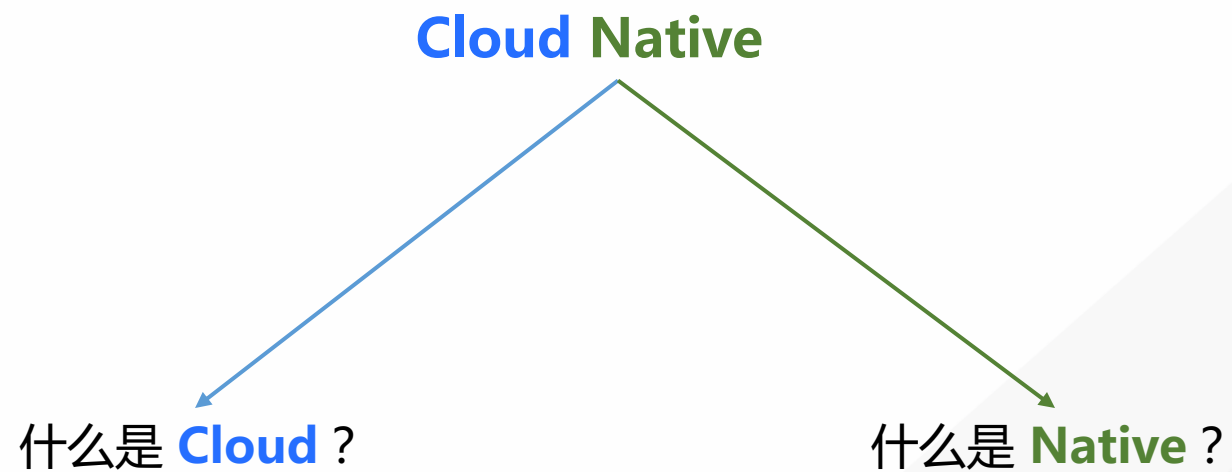
云原生的定义 v1.0

云原生技术有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用。云原生的代表技术包括容器、服务网格、微服务、不可变基础设施和声明式API。

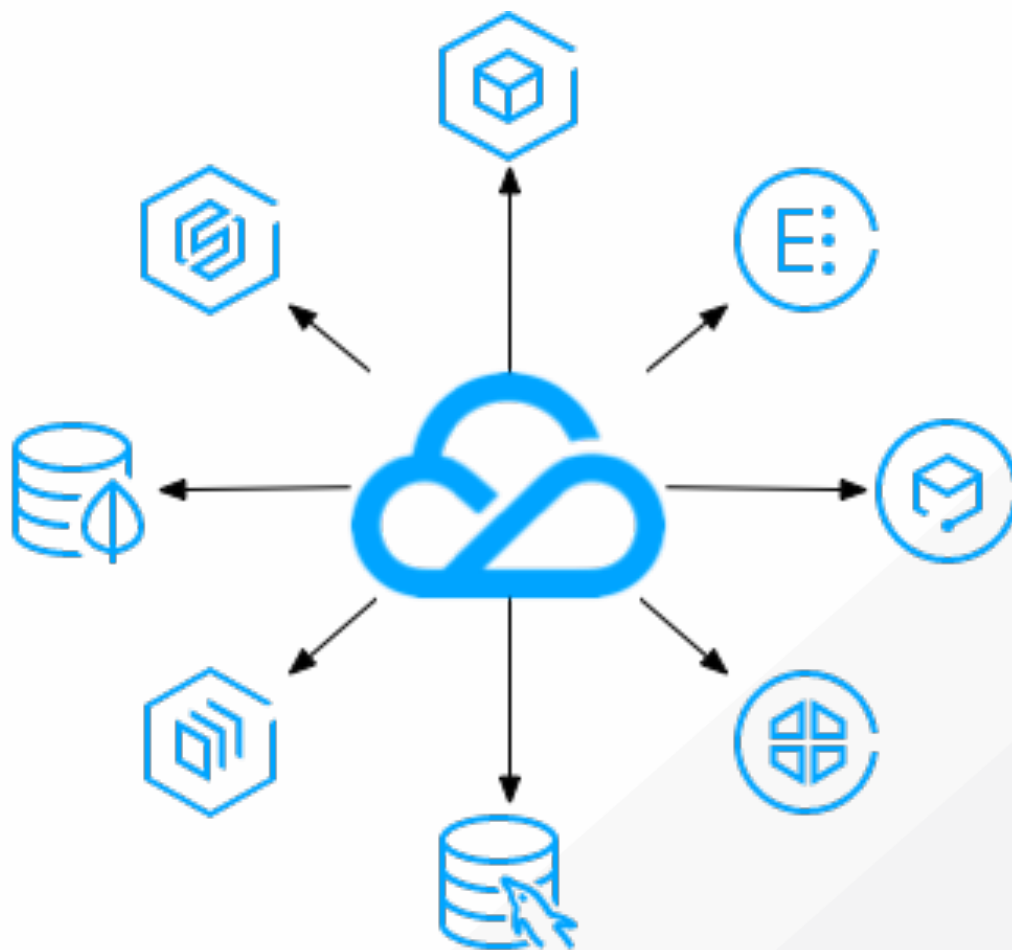
这些技术能够构建容错性好、易于管理和便于观察的松耦合系统。结合可靠的自动化手段，云原生技术使工程师能够轻松地对系统作出频繁和可预测的重大变更。

—— CNCF（云原生计算基金会）2018.06.11

如何理解云原生



如何理解云原生：什么是Cloud



什么是Cloud：云计算发展史



什么是Cloud：云计算发展史

Serverless

传统自建服务体系

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

IaaS

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

PaaS

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

BaaS

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

FaaS

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

SaaS

应用

业务逻辑

数据

运行时

中间件

操作系统

物理机

网络、机房、电力

如何理解云原生：什么是Native

native ☆ ⋮

🔊 英 ['neɪtɪv] 🔊 美 ['neɪtɪv]

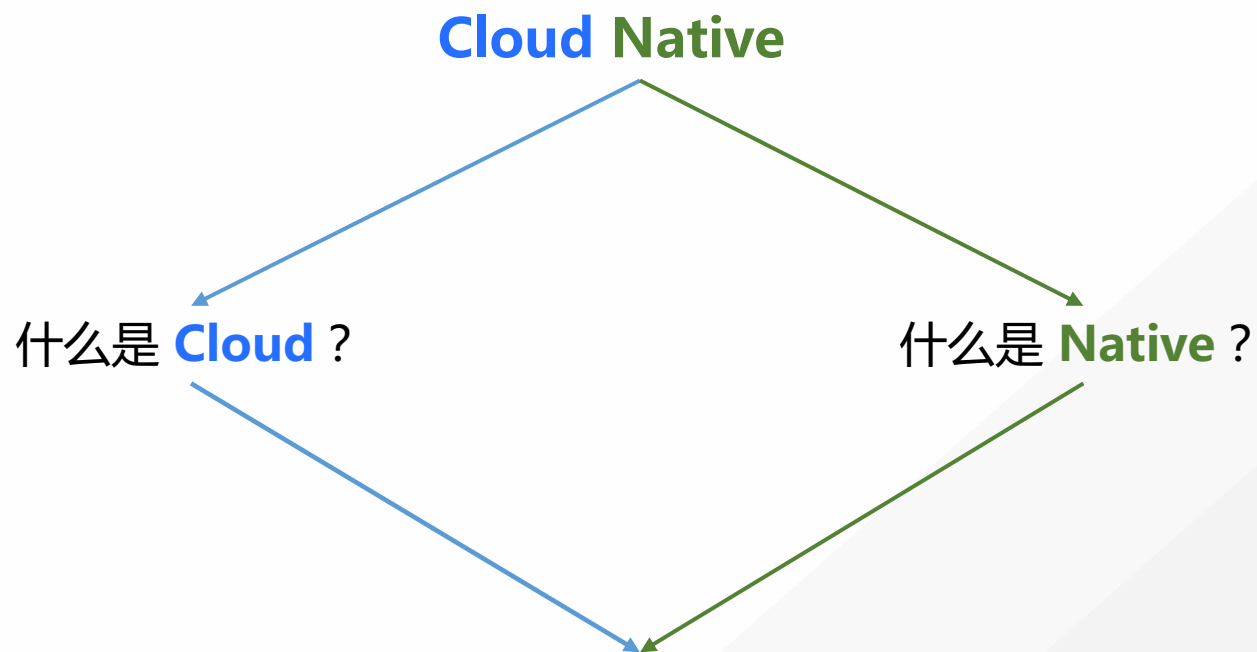
adj. 本国的；土著的；天然的；与生俱来的；天赋的

n. 本地人；土产；当地居民

that you have naturally without having to learn

—— 牛津词典

如何理解云原生



应用程序原生 (**Native**) 被设计为以云 (**Cloud**) 为服务依托



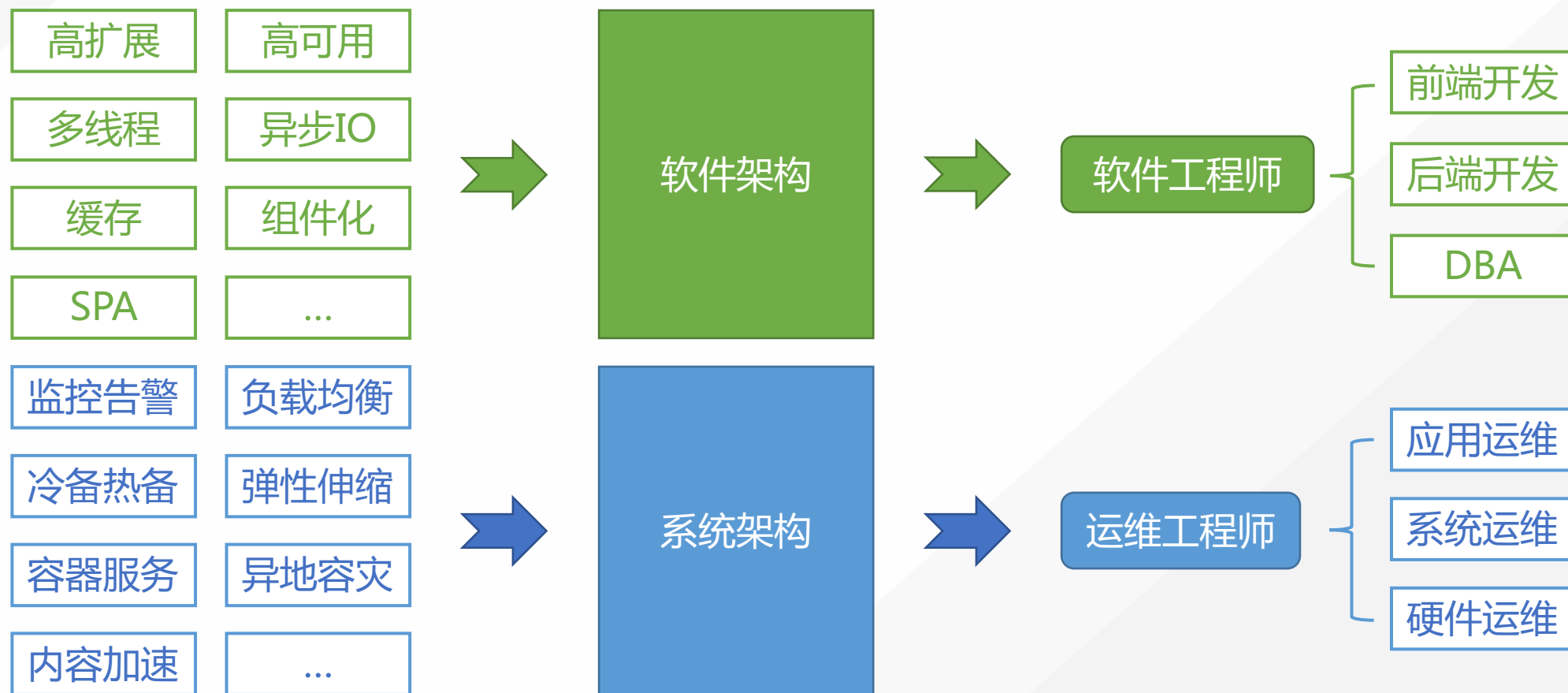
2 传统研发模式的问题

我有一个梦想

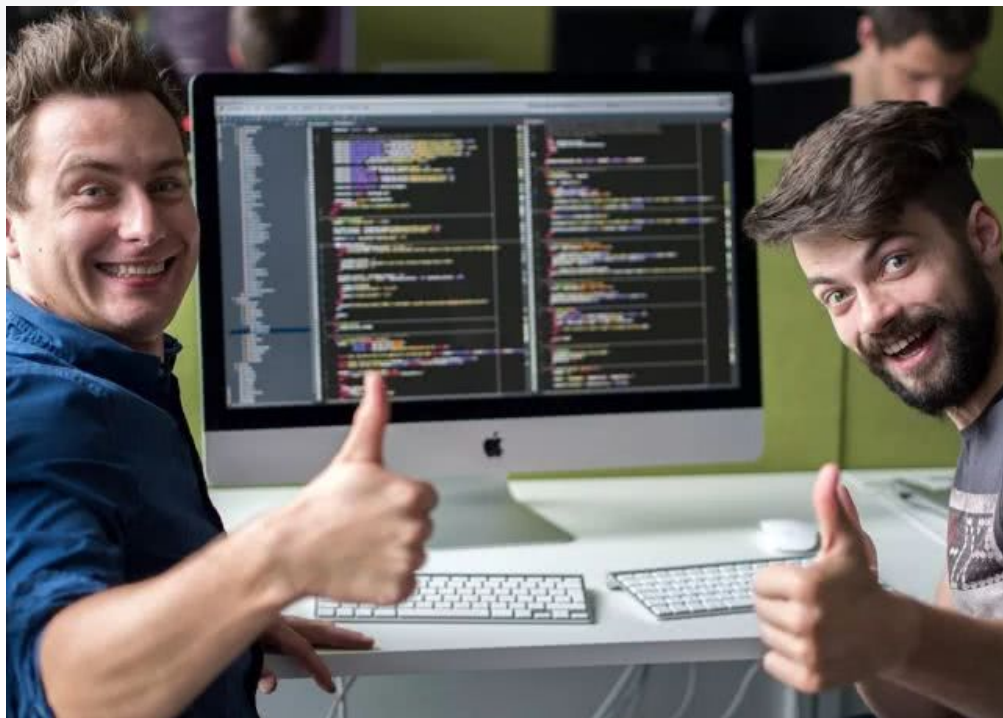
服务端支撑业务逻辑需要什么



运行一款软件需要哪些职能？



“和谐”的前后端协作

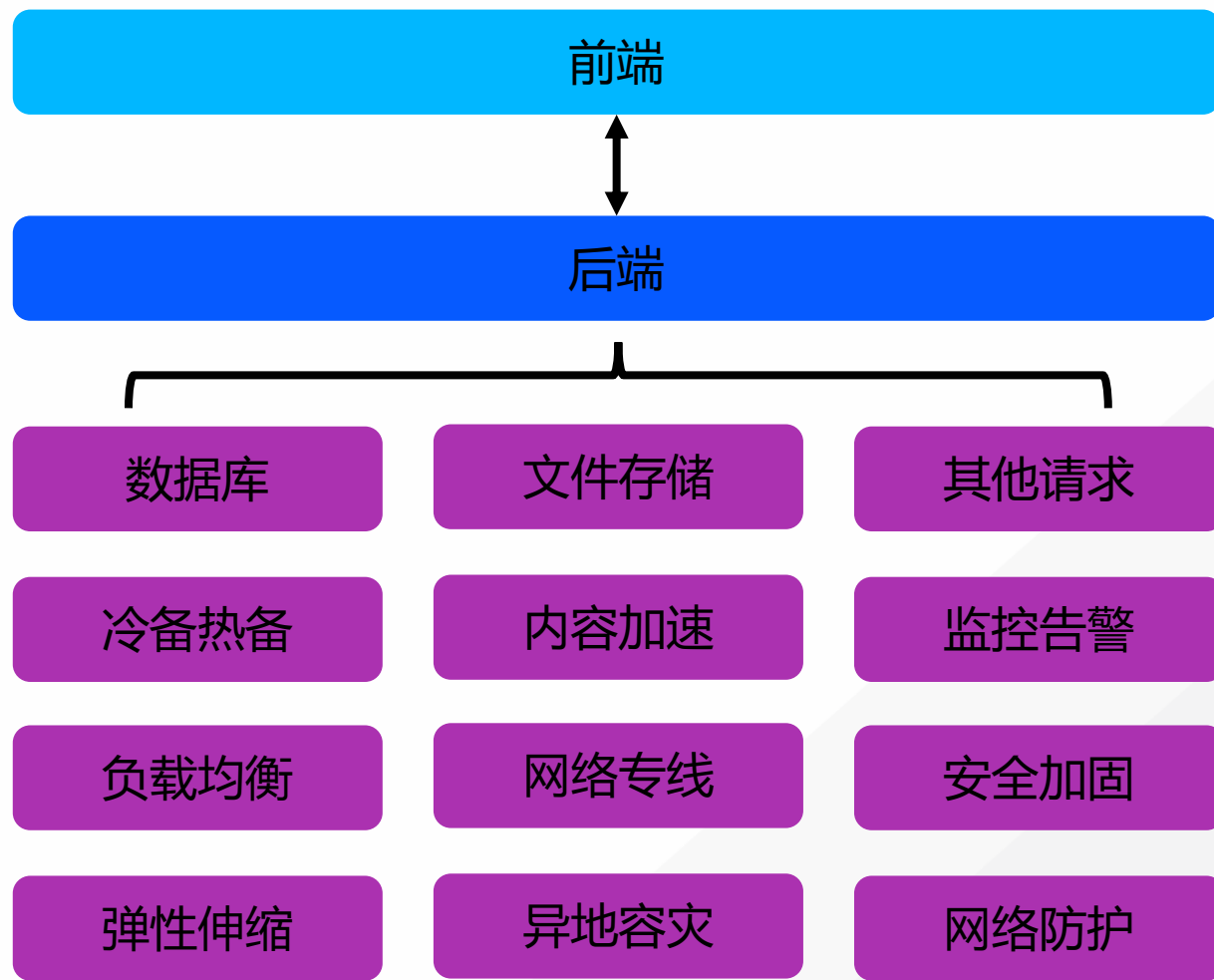


想象中的协作情景

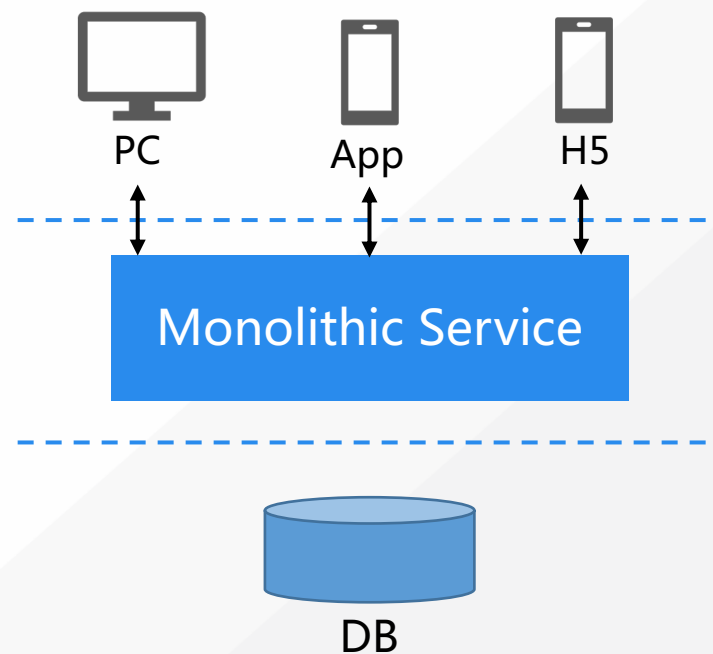
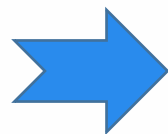
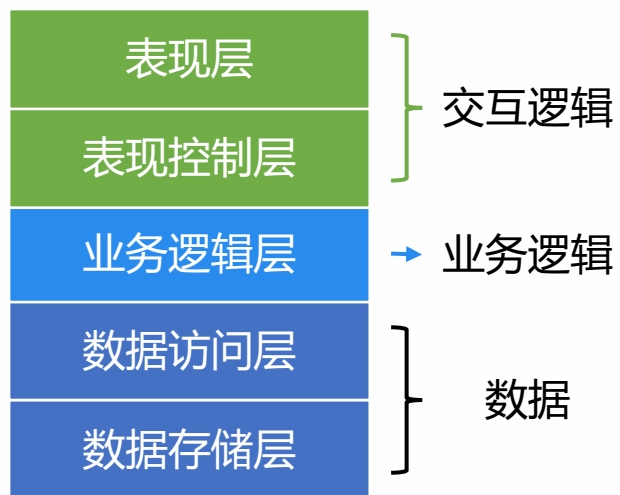


现实中的协作情景

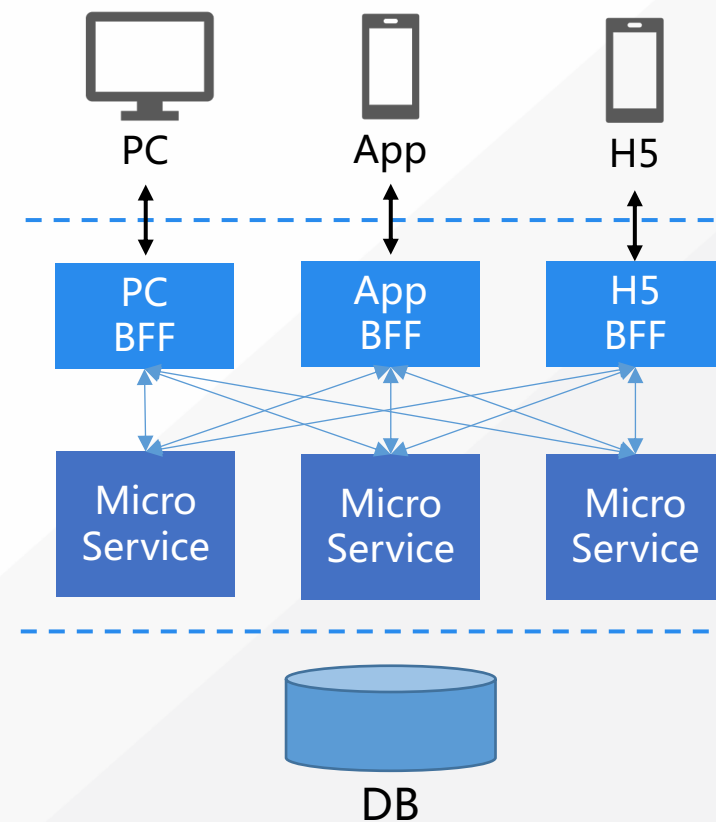
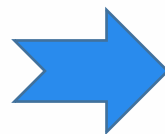
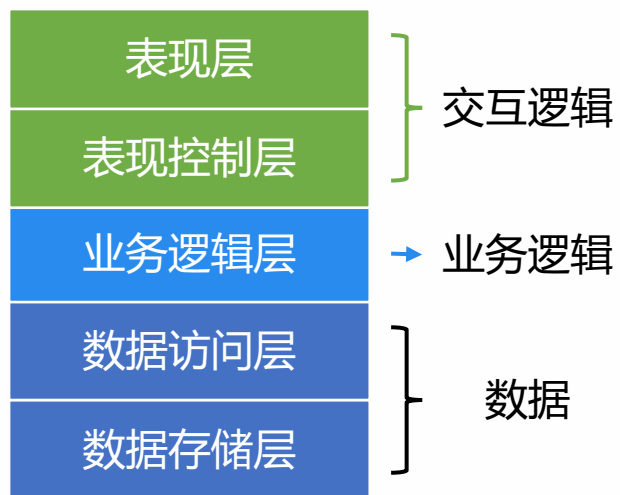
传统前后端分工开发模型



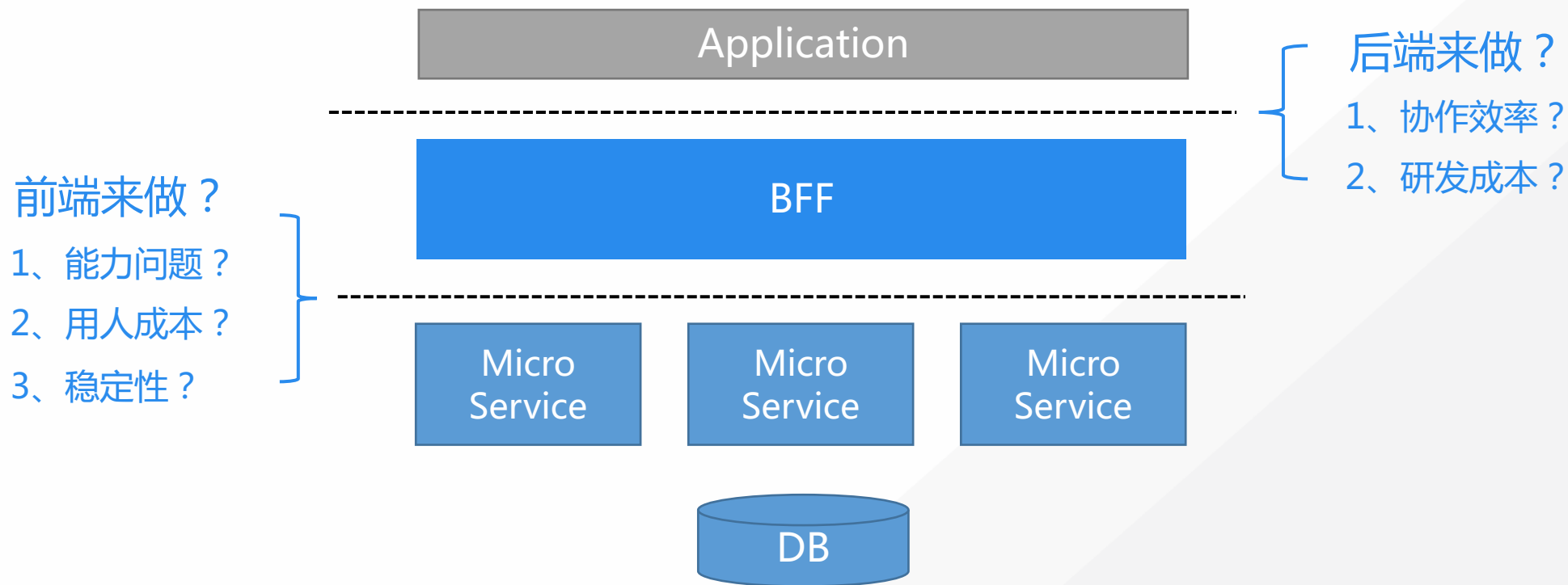
BFF-平台差异化的必然产物



BFF-平台差异化的必然产物



前后端分离的难题：BFF谁来做？



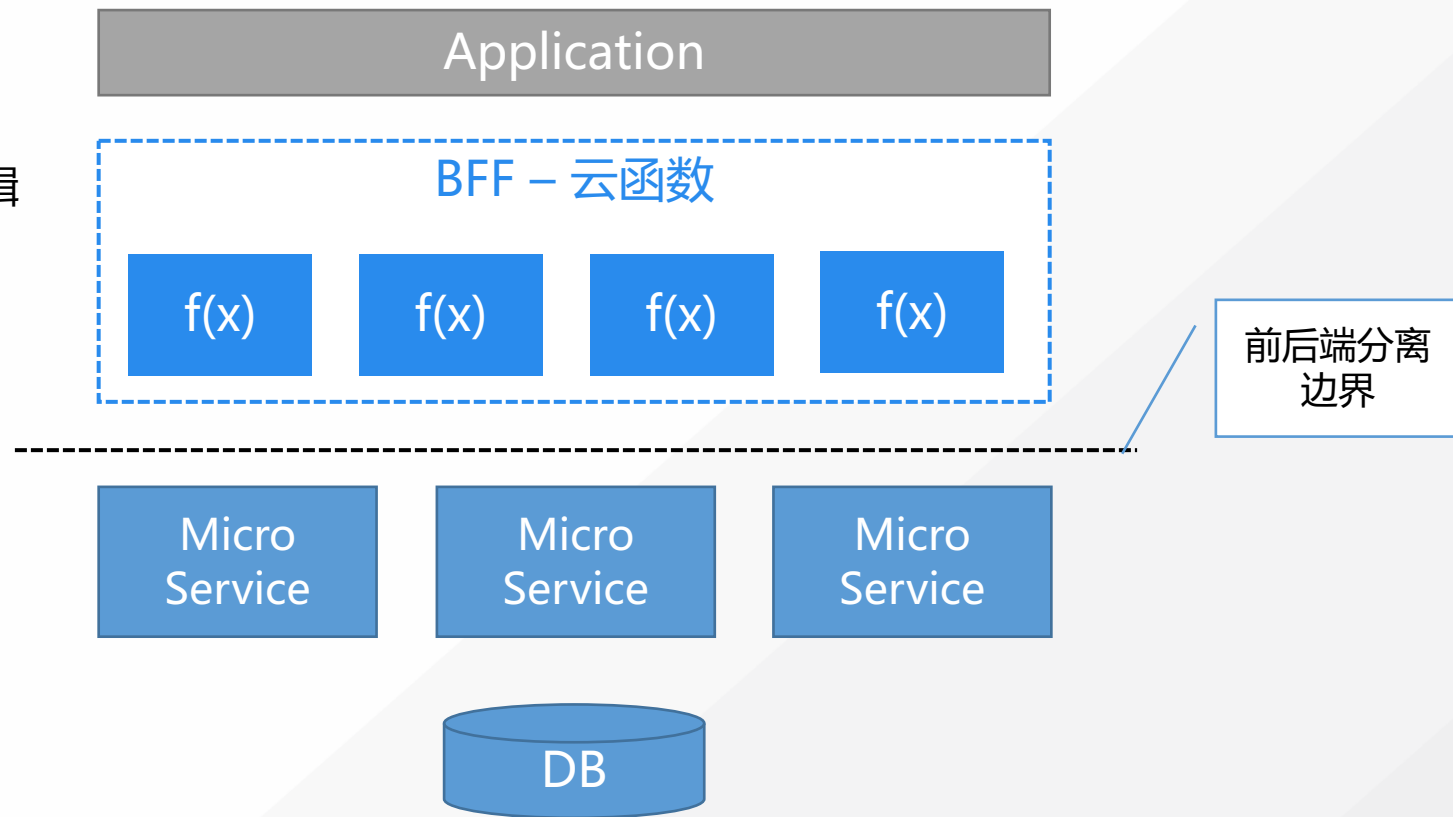


3 云开发模式的架构演进

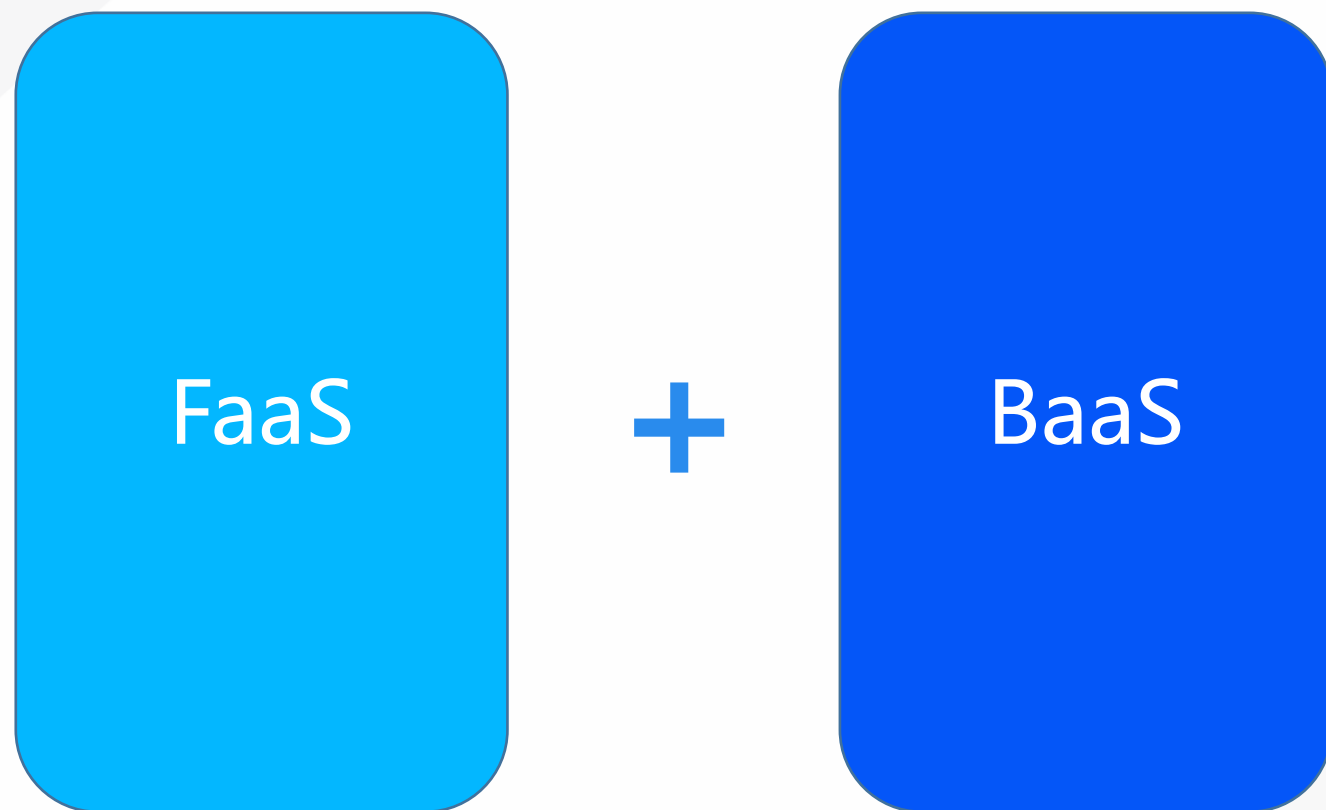
不止于Serverless

云开发模式下的前后端分离模型

1. 云函数承载BFF功能，实现平台差异化业务逻辑；
2. 聚焦业务逻辑本身，负载均衡、弹性伸缩等工作交由云函数底层平台完成；
3. 完善的监控预警功能，及时预警、及时应对；
4. 前后端分离界限下沉，更易解耦。



Serverless在对端领域还缺少什么？



- ❑ 如何触达应用端？
- ❑ 如何支持丰富业务场景？
- ❑ 安全性怎么样？
- ❑ 如何爬虫防刷？
- ❑ 迁移成本高吗？

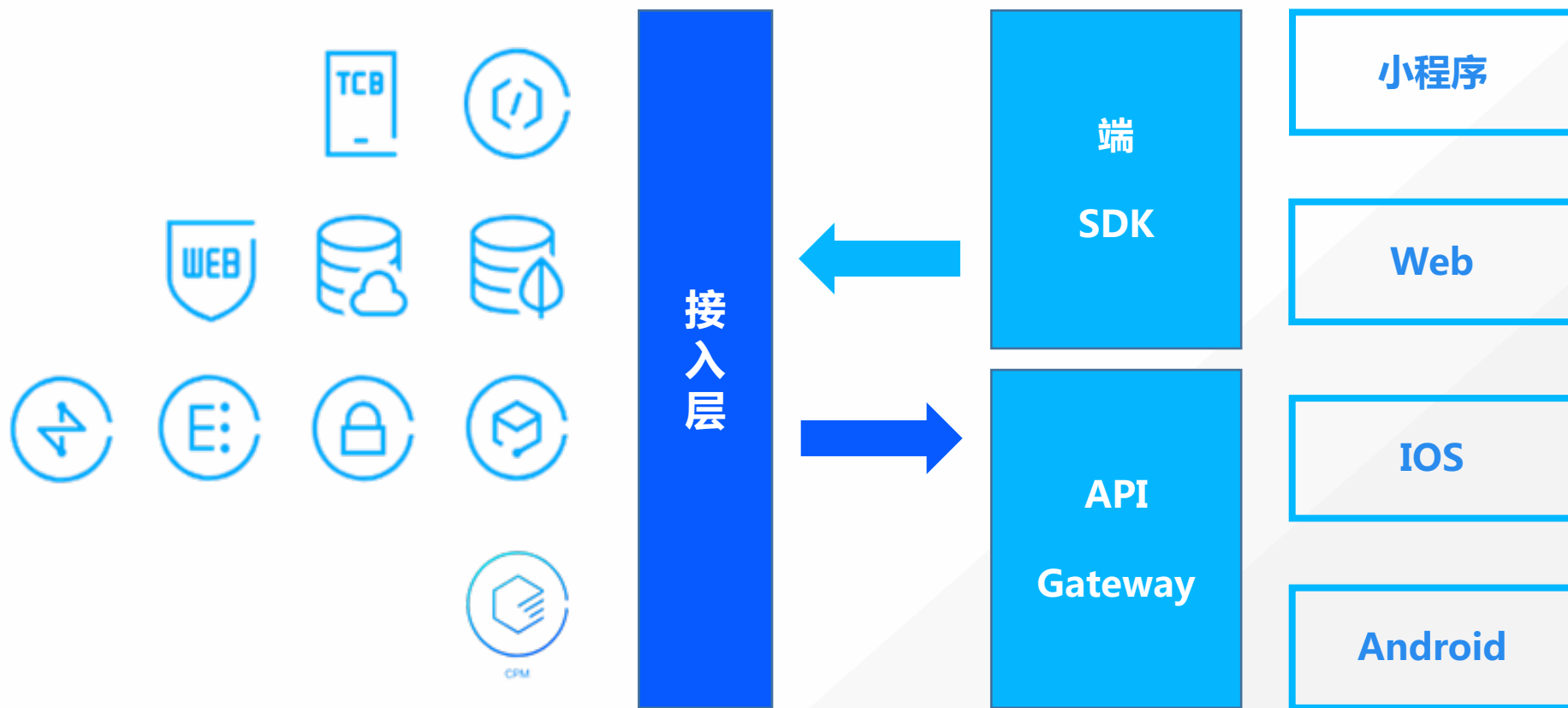
认识云开发：CDN vs 云存储

```
// 上传
wx.cloud.uploadFile({
  cloudPath: 'example.png', // 上传至云端的路径
  filePath: '', // 小程序临时文件路径
  success: res => {
    // 返回文件 ID
    console.log(res.fileID)
  },
  fail: console.error
})

// 下载
wx.cloud.downloadFile({
  fileID: '', // 文件 ID
  success: res => {
    // 返回临时文件路径
    console.log(res.tempFilePath)
  },
  fail: console.error
})
```



云开发打通Serverless对端的“最后一公里”



不止于Serverless：云开发能力矩阵

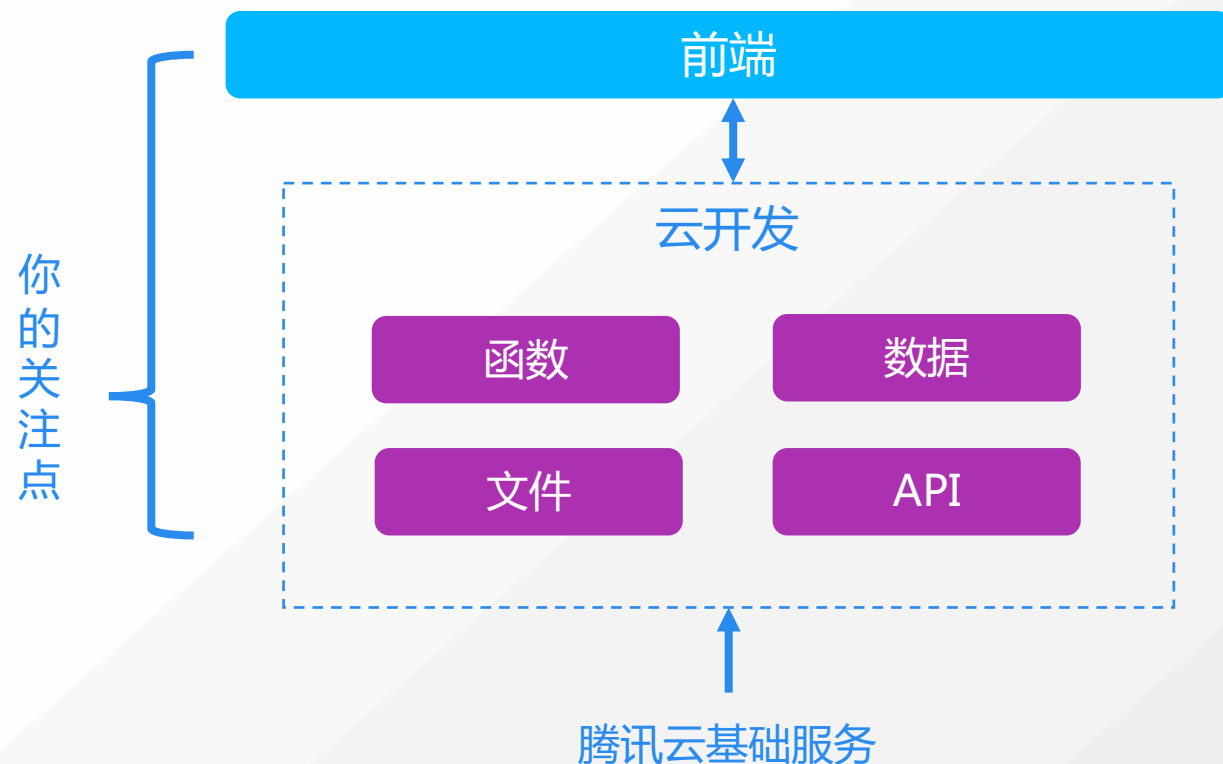
产品类别	技术能力	
基础能力 满足基础开发需求	云函数	在云端运行代码，微信私有协议天然鉴权。开发者仅需关注自身业务逻辑
	云存储	端内直接上传/下载资源，控制后台可视化管理
	云数据库	即可在端内操作，也可以在云函数内操作的JSON数据库
	云托管	基于Knative的云托管服务
	实时推送	广泛适用于即时聊天/通信、多人游戏等场景
增值能力 丰富应用场景需求	扩展能力	与腾讯云能力打通，提供更安全、更便捷的小程序场景人脸融合、滤镜、手势识别等AI能力
	CMS	后台内容管理系统
生态 给予开发者沃土	多语言	Node.js、Java、PHP、C#
	多端	微信小程序、Web、Flutter、Android、IOS
	开发者工具	Cloudbase Framework、VSCode插件

传统开发 vs 云开发

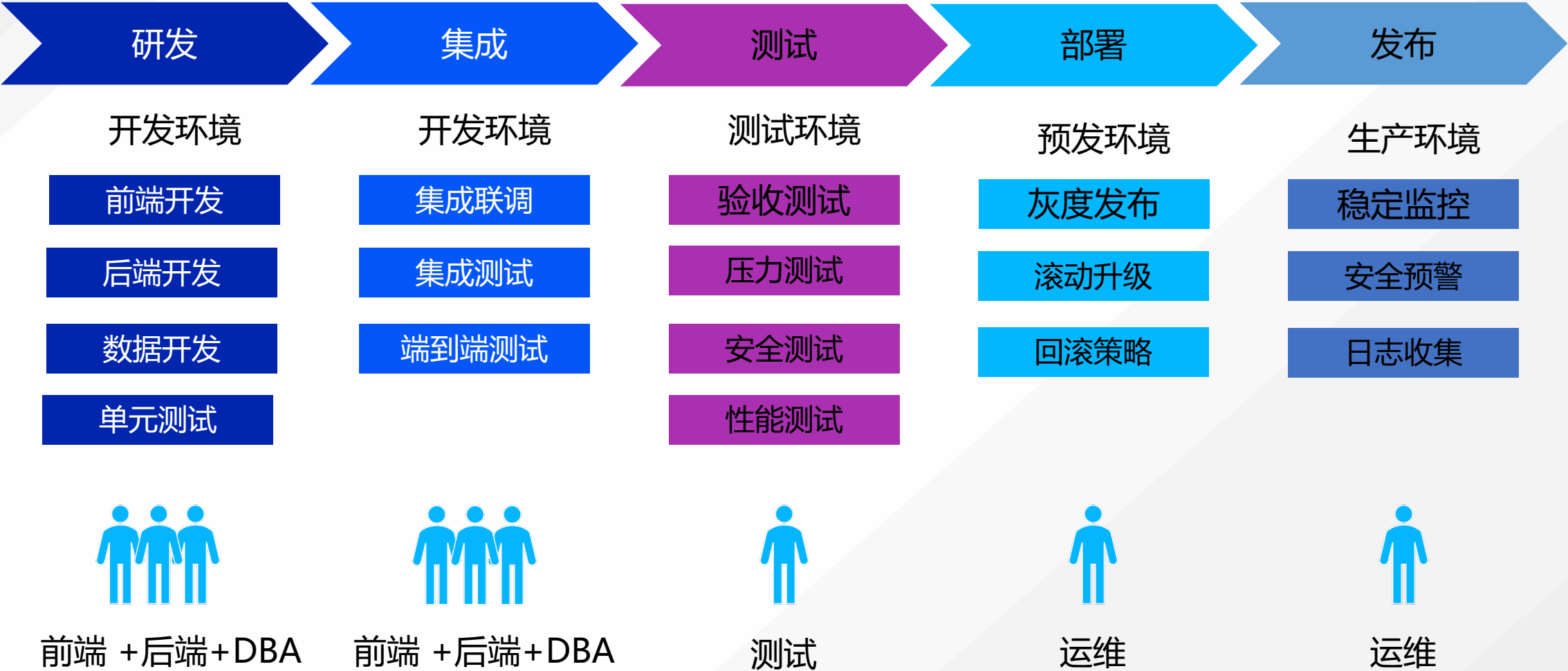
传统开发模式



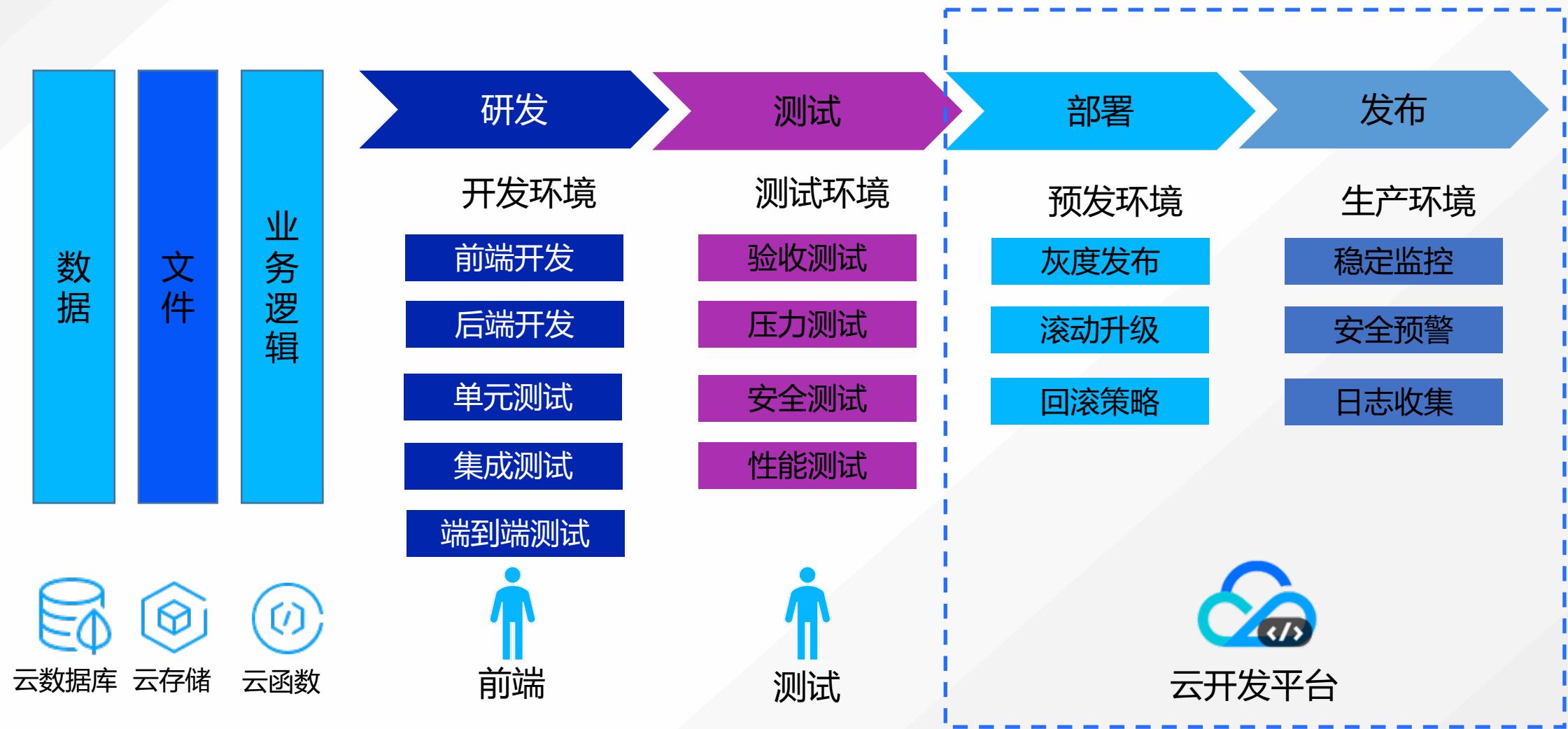
云开发模式



传统开发模式的工程模型



云开发模式的工程模型





Q & A

举例：人机交互设计课程是如何验收的？

XXX：XXXX

1. XXX
2. XXX
3. XXX
4. XXX



谢谢大家