

Rethink : Serverless For Frontend

天猪 / 蚂蚁金服体验技术部 - Node.js 基础服务





天 猪

Node.js、EggJS、开源控、奶爸

蚂蚁金服 - 体验技术部 - Node.js 基础服务

广州分舵

 [@atian25](#)

 [@liuyong25](#)

“

或许现在很多企业和团队**尚未重视前端**，或许前端在很多人眼里还只是『**切图仔**』的代名词，又或许未来前端领域的变革使得**人机交互**问题从根本上得到解决。

不管怎么样，作为亲历者之一，只希望当下能认真地**记录下**这群从业者的**摸索**和所见所想，与正在演进却又有困扰的同行们分享交流一些踩坑心得。



”

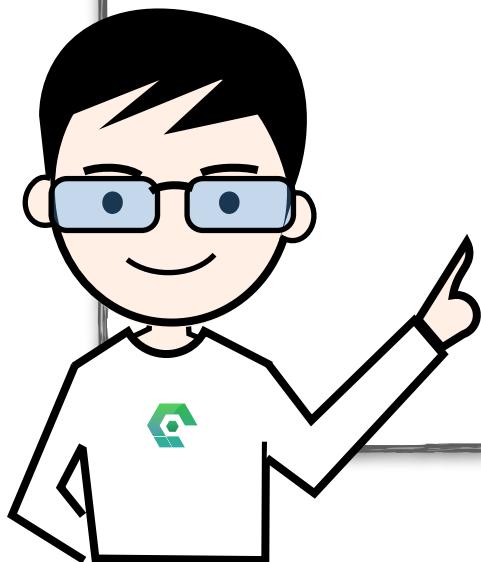
别人眼中的前端



那么，作为从业者的 **你** 而言：

前端是什么？

前端的核心价值是什么？

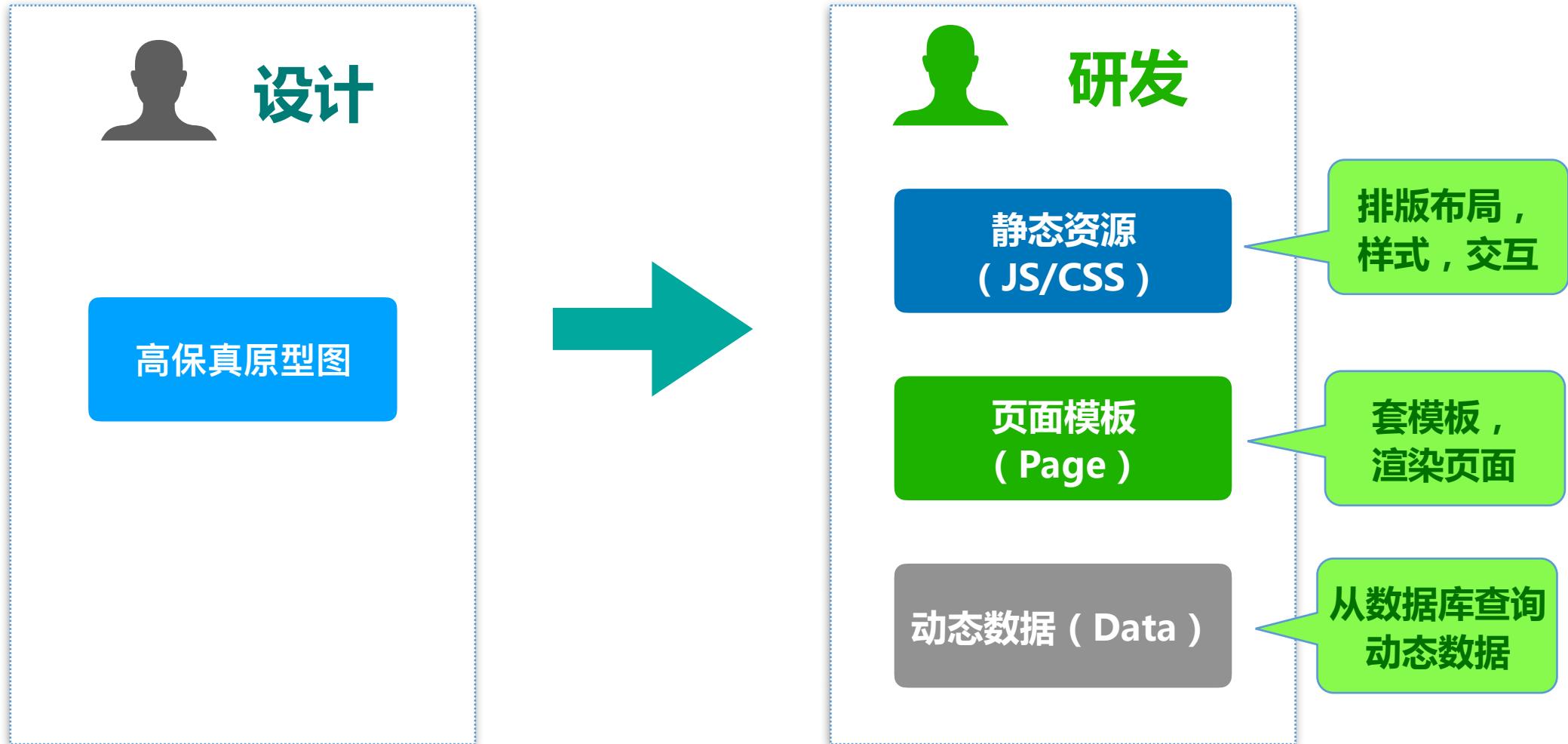


敲黑板了！请边听分享，边带着问题思考。

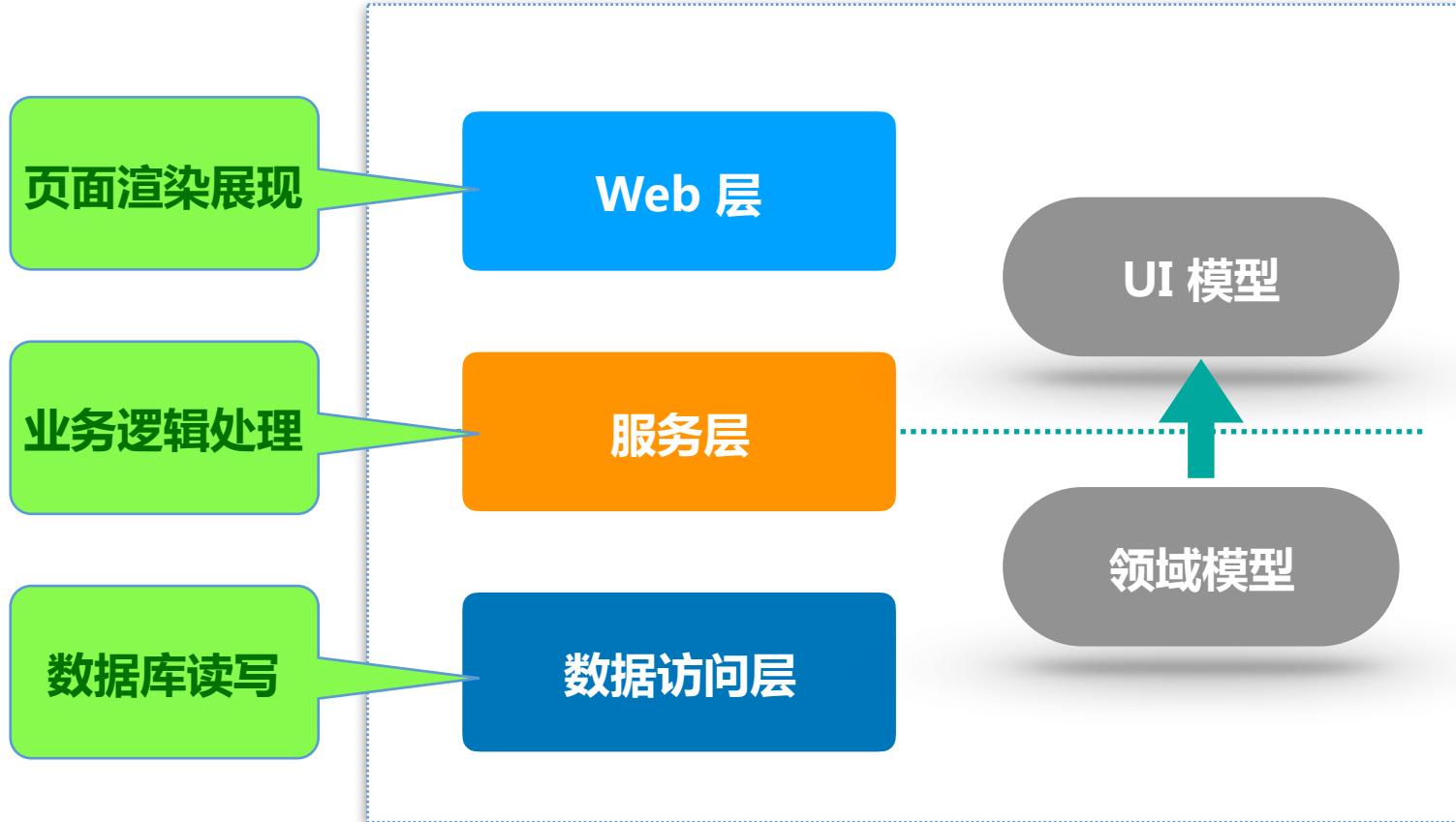
Long long ago ...

远古时代

远古时代，没有前端后端之分。



远古时代



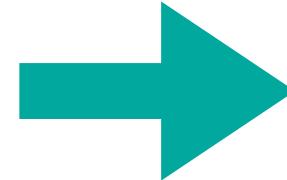
- ▶ **Web 1.0 时代**
 - 集中式架构
 - 标准的三层 MVC 架构
- ▶ Struts + Spring + Hibernate
- ▶ Dreamweaver 三剑客
- ▶ PHP 囧rz

远古时代 → 石器时代



『美工们』天马行空的点子，像素级复原，追求极致用户体验。

艺术与代码的 GAP

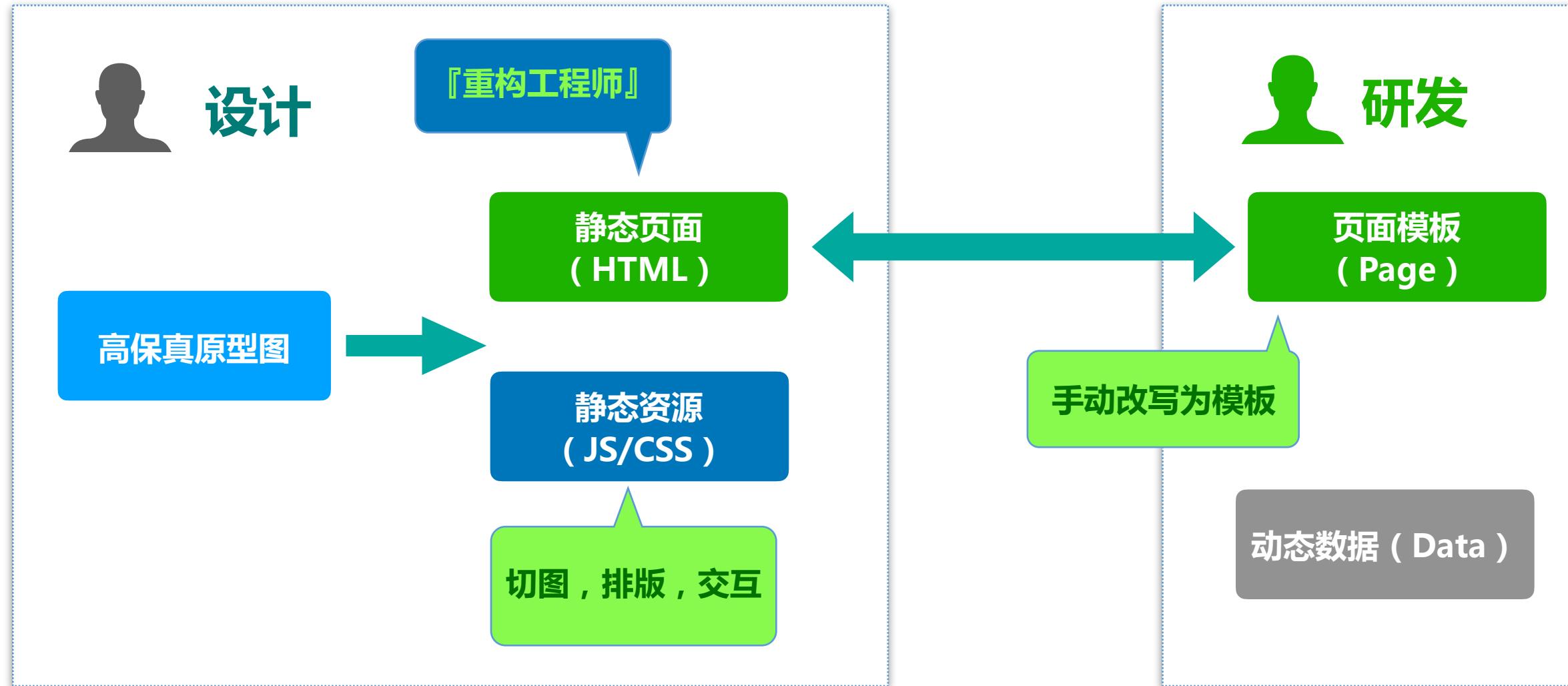


专业性要求 ↑

分布式架构，对研发能力的要求

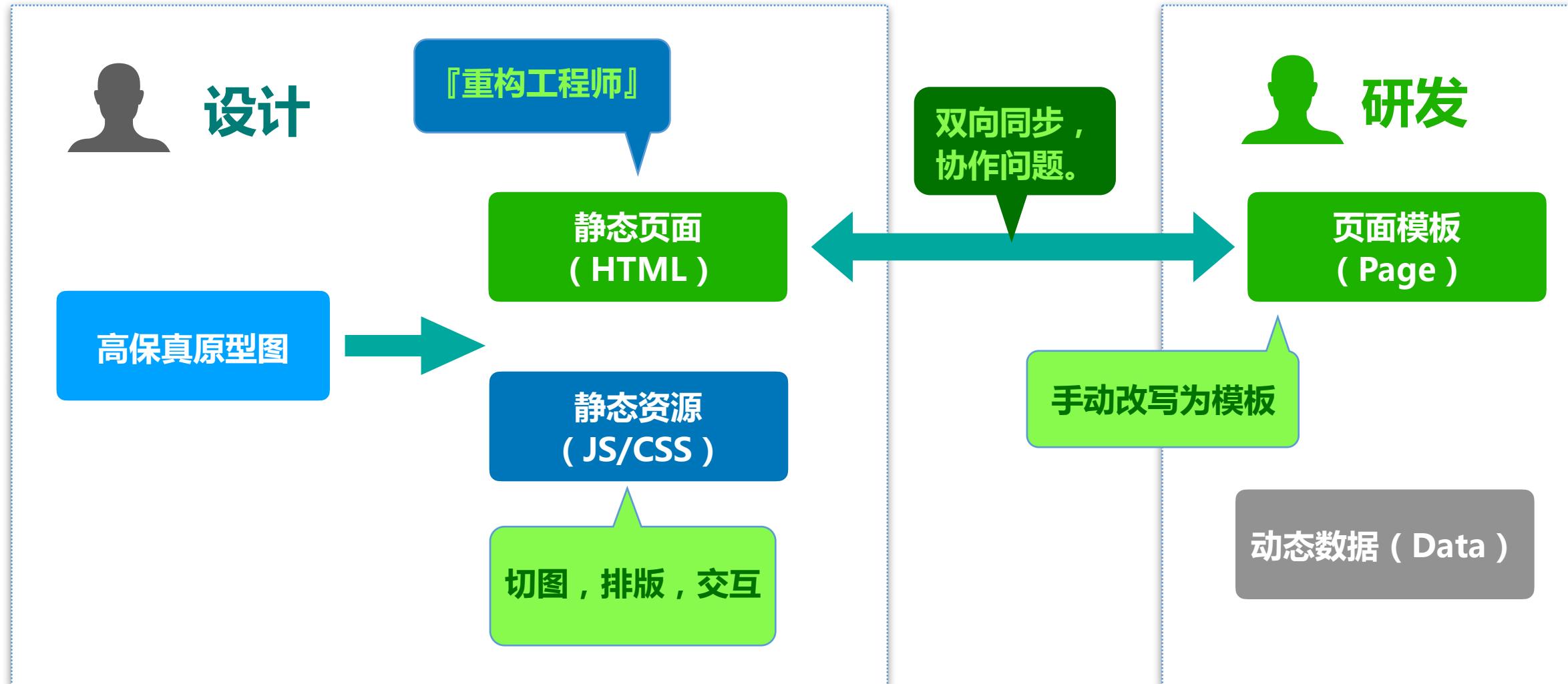
互联网迅猛增长，海量的并发压力，集中式 → 分布式。

石器时代

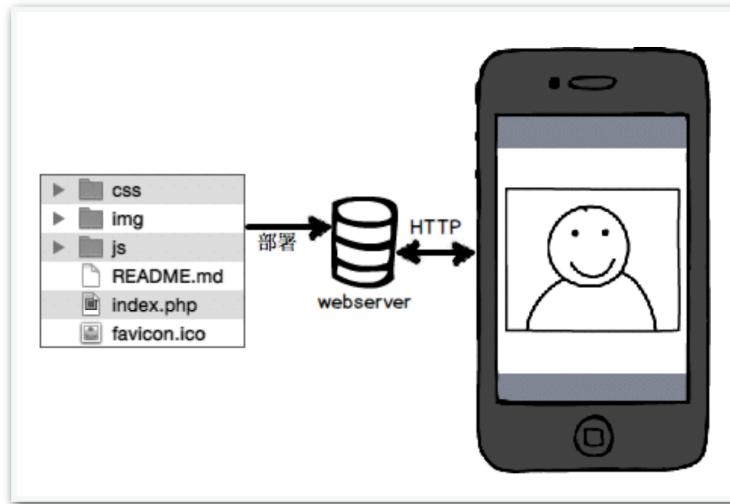
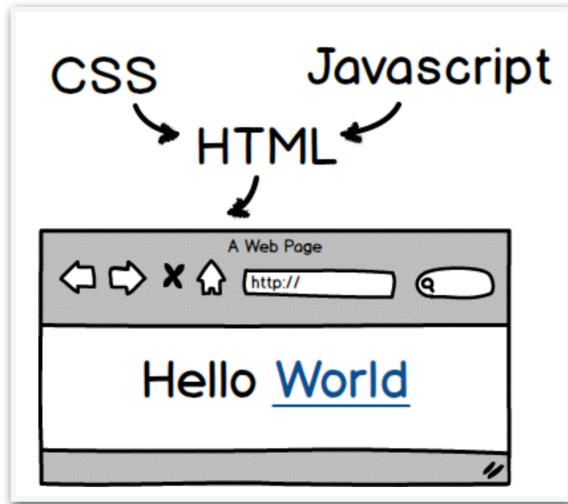


石器时代 → 青铜器时代

主要矛盾：耦合，沟通协作效率低



Web 2.0 时代，界面和交互越来越复杂，对 专业性 的要求进一步提高。



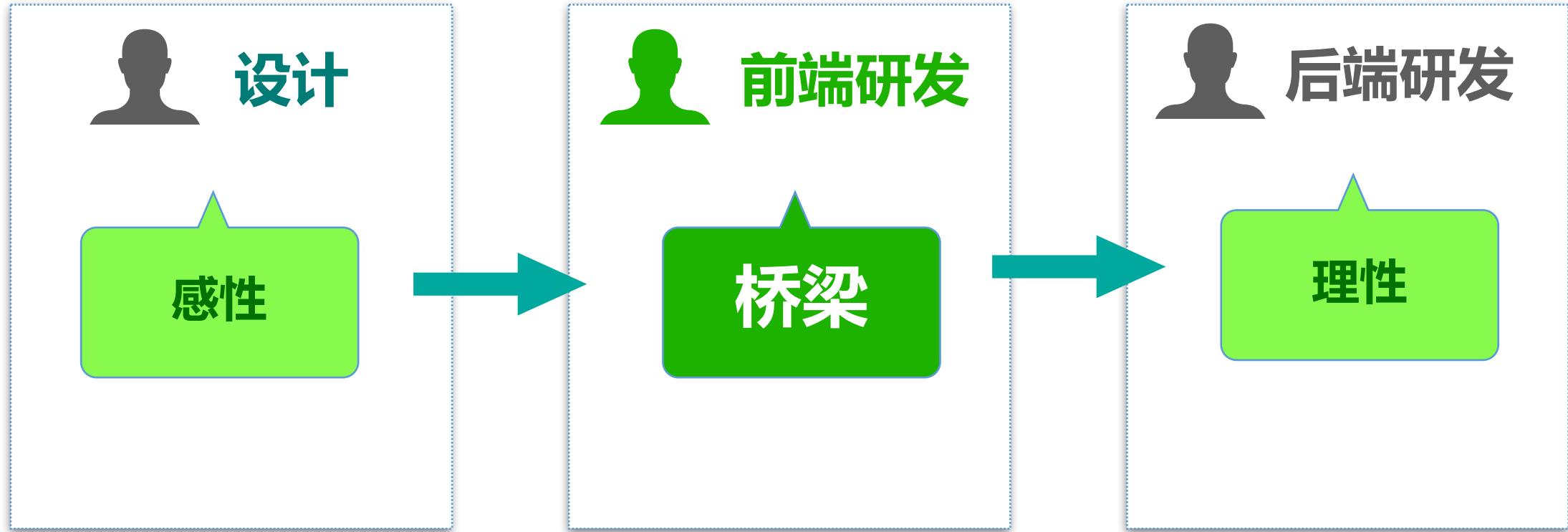
三种风格迥异的
语言混合开发

工程问题复杂
远程部署、增量更新

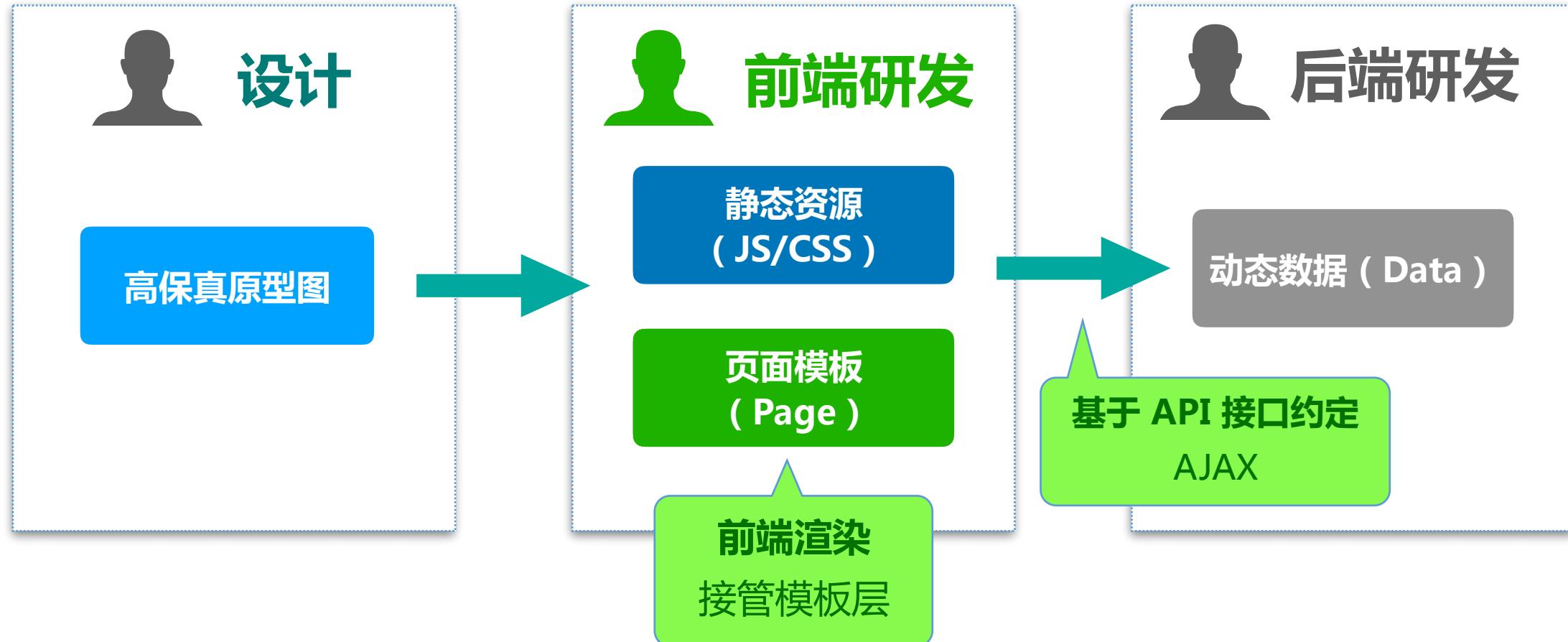
富应用时代
追求人机交互体验

以 GMail 为号角，前端诞生

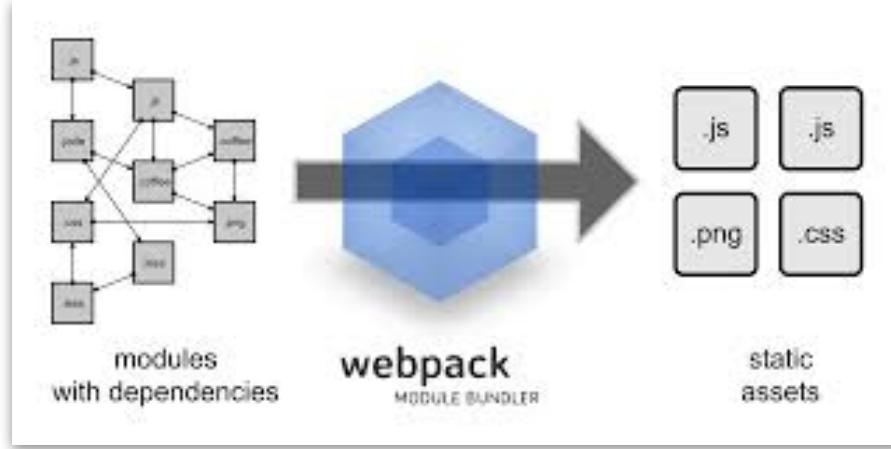
前端岗位的诞生



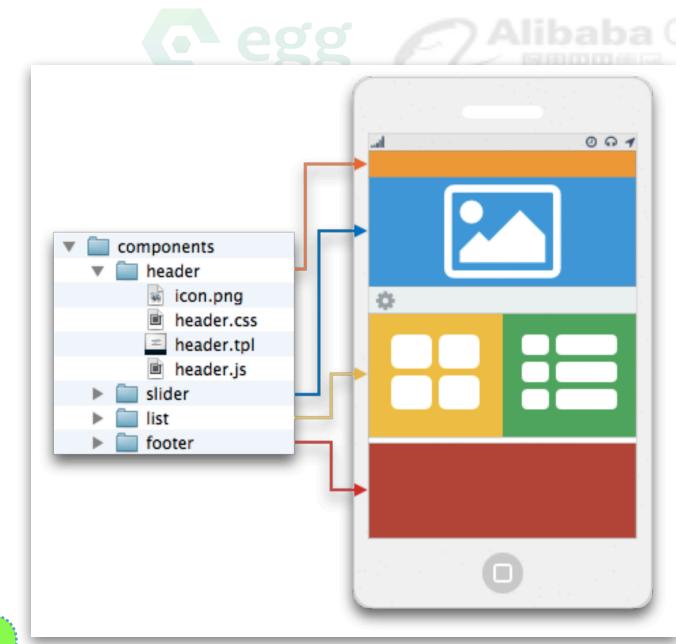
第一次『前后端分层』：划分职责边界，提升协作效率



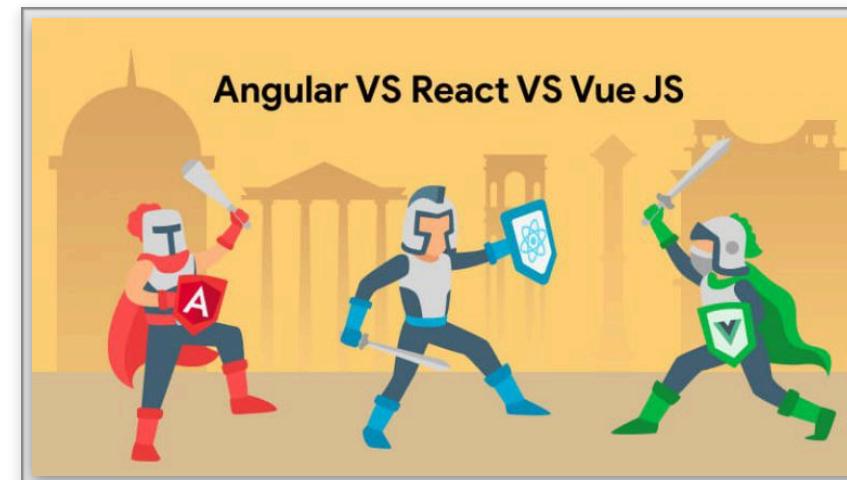
青铜器时代



组件化



工程化



前端框架

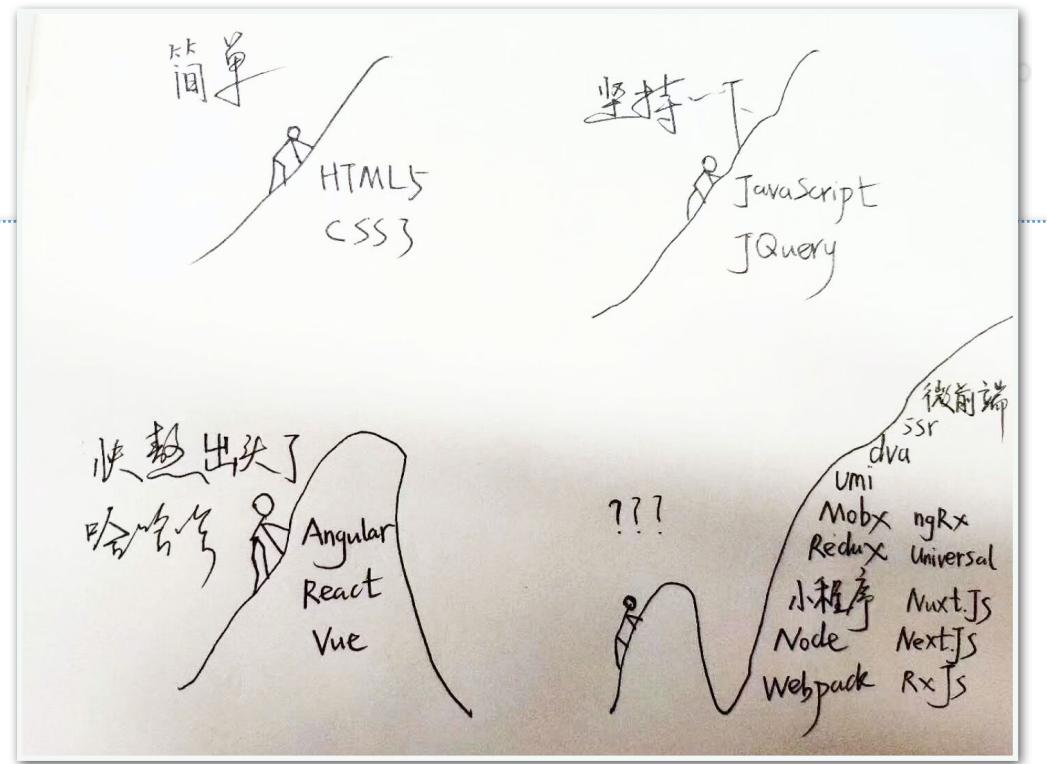
青铜器时代 → 蒸汽时代



第一次工业革命，前端的版图在持续扩张。

广度：跨平台、跨终端（PC、手机、平板、手表、大屏、嵌入式设备）

深度：APP、中台、IDE、富文本编辑器、可视化图表、拓扑图分析、地理空间



‘

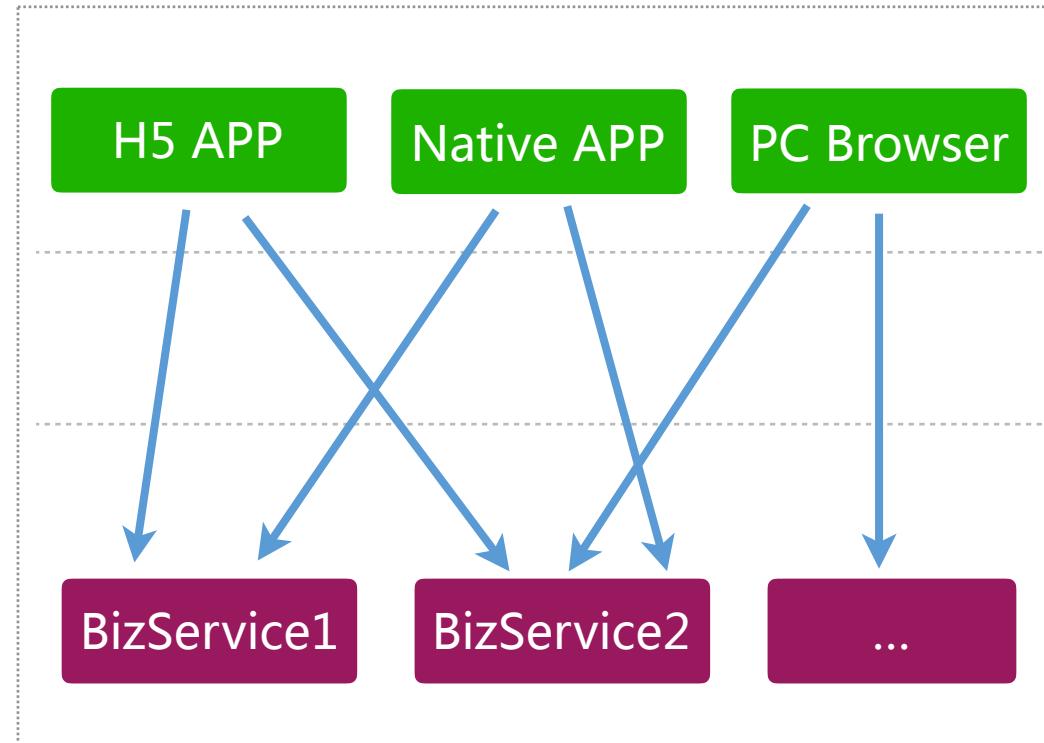
前端 = 人机交互工程师

负责『人机交互层』的领域研究与技术实现。

’

人机交互领域涉及三个核心工种：产品、设计师、前端，这三个默契的组合在一起才能形成极致的人机交互体验。

青铜器时代 → 蒸汽时代



移动互联网时代 - 跨平台、跨终端

微服务化 - 下沉，趋向稳定

青铜器时代 → 蒸汽时代

是不是熟悉的前后端的日常，你遇到过么？



公众号 @吓脑湿



原创出品

产品说这个需求 3 天后就要上线！
后端这里帮忙 **加个接口** 做下数据清洗吧？

不是有几个现成的接口么，你在前端
自己组合清洗下就好了吧？又不复杂！

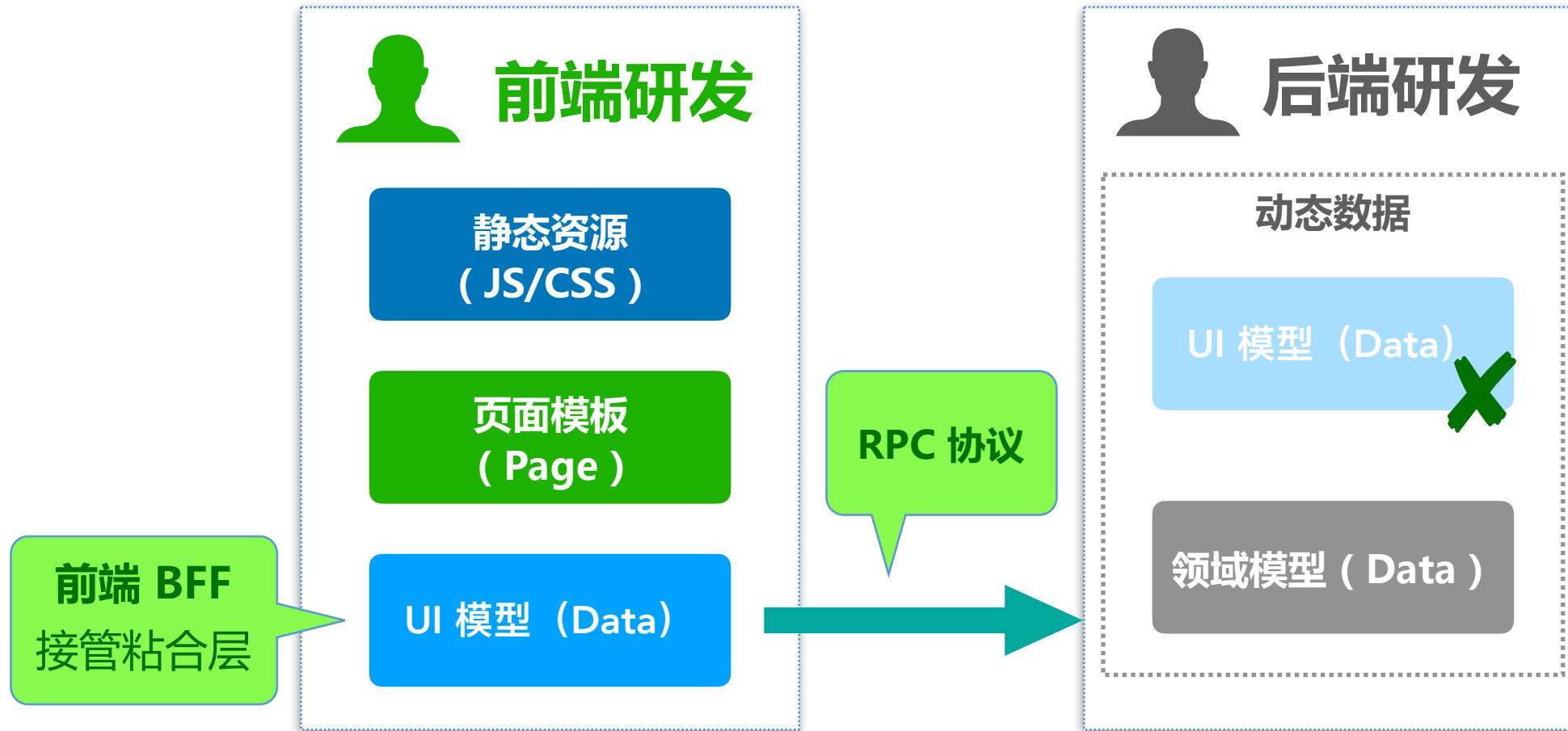
大佬，太多请求会**影响性能**啊！
而且还有敏感字段，不安全啊！

你来晚了，发布窗口过了，
我们后端一个月才能发布一次。

青铜器时代 → 蒸汽时代



第二次『前后端分层』：服务自治，谁使用谁开发。





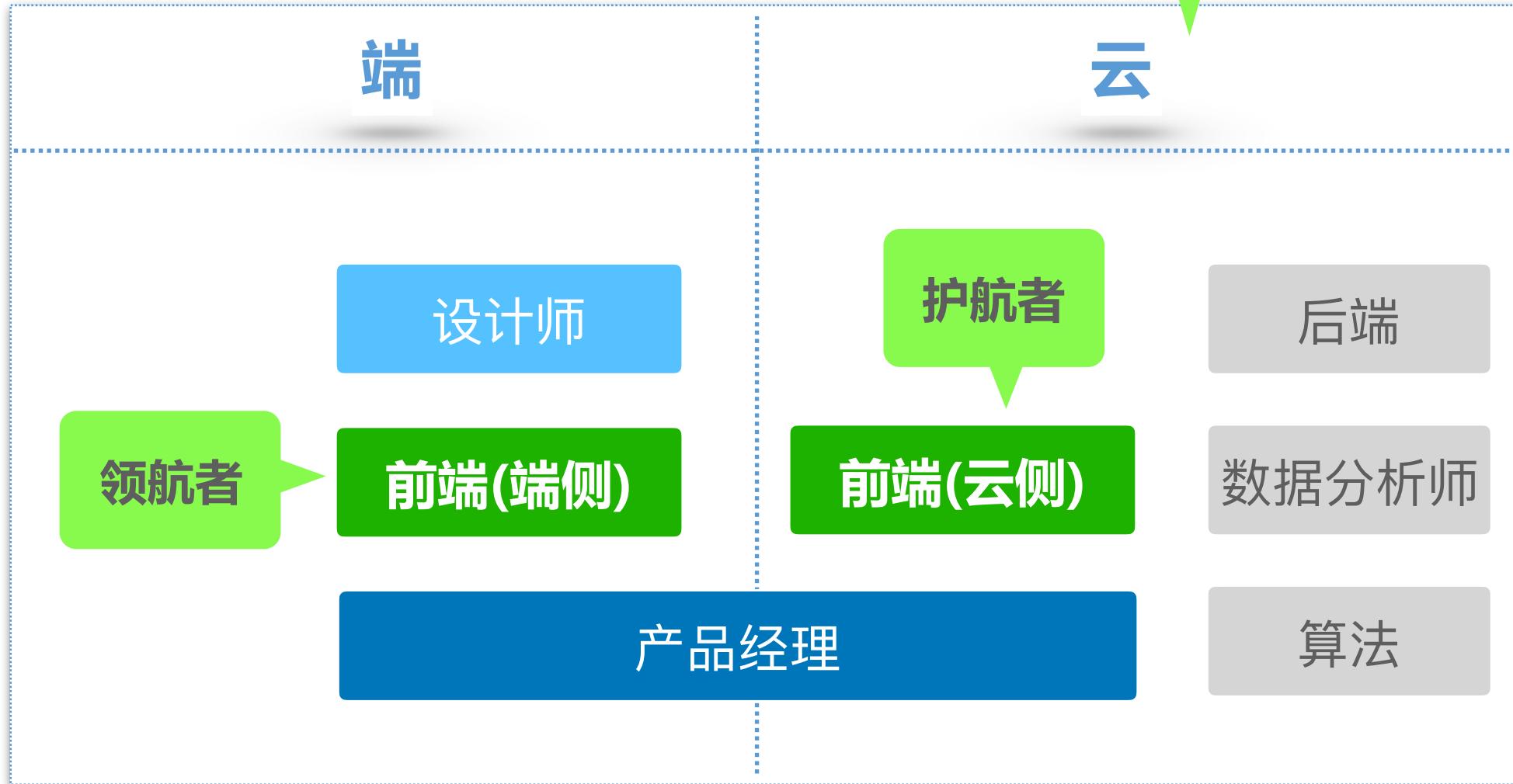
前端 = 人机交互工程师，负责人机交互层的领域研究与技术实现。

前端的核心价值，是完成技术的最后一公里，让技术可交付给用户使用，是科技触达用户的桥梁，是数字世界里的建筑工。



蒸汽时代 - 边界扩大

为了更好的人机交互体验，
前端领域扩展到云端



端 → 云

门槛，BFF 的实践，在社区的分化严重。

在大公司和创业公司比较受欢迎，但在话语权不强的中型公司，则举步维艰。

- **小创业公司** - 追求快狠准，先活下来再说，效率优先，干就是了。
- **大公司** - 具备良好的基建和研发流程支撑，对效能有更高的追求，如荼如火。
- **中型公司** - 前端话语权不强，百废待兴，现有基建对前端不友好，推行举步维艰。

蒸汽时代 - 演进思路



根基，先安内后攘外

营销页搭建系统、
API Mock 系统

不抢活，合理分工

前端工程化

+

内部练兵

+

外部合作

勿忘初心，殊途同归，推动后端**共建系统**，也能走向我们的目标。
觉得有个 TS 就天下无敌，到处挑衅 Java 的，是猪队友。



蒸汽时代 → 电气时代



招人难，培养成本高

远远低估了前端的缺人程度，一将难求，无人可招。
全栈人才的培养成本不低。

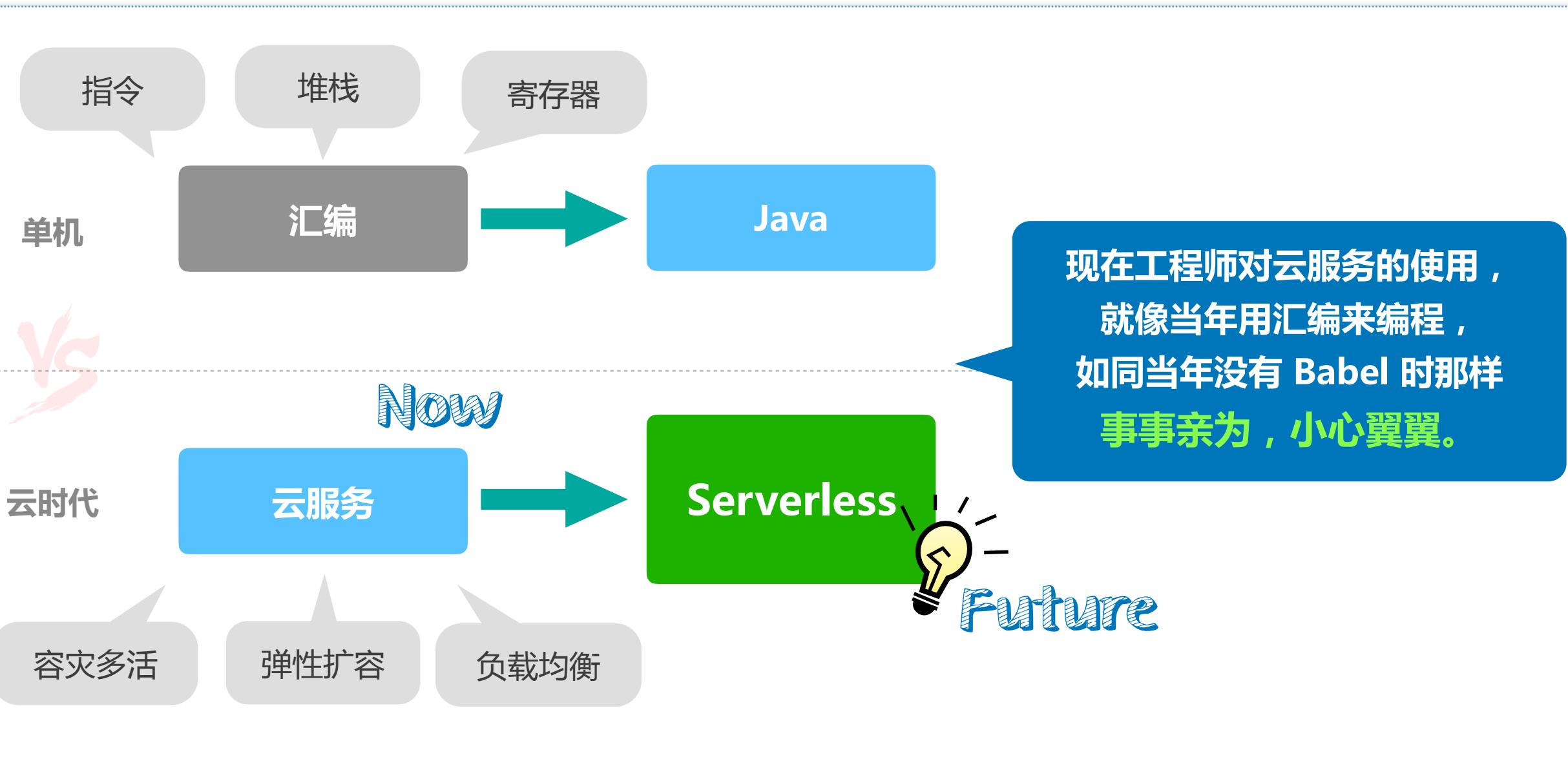
开发、运维成本增加

目前云服务的使用方式比较低效（不仅仅是前端）。
从而带来的额外的运维成本和上手成本。

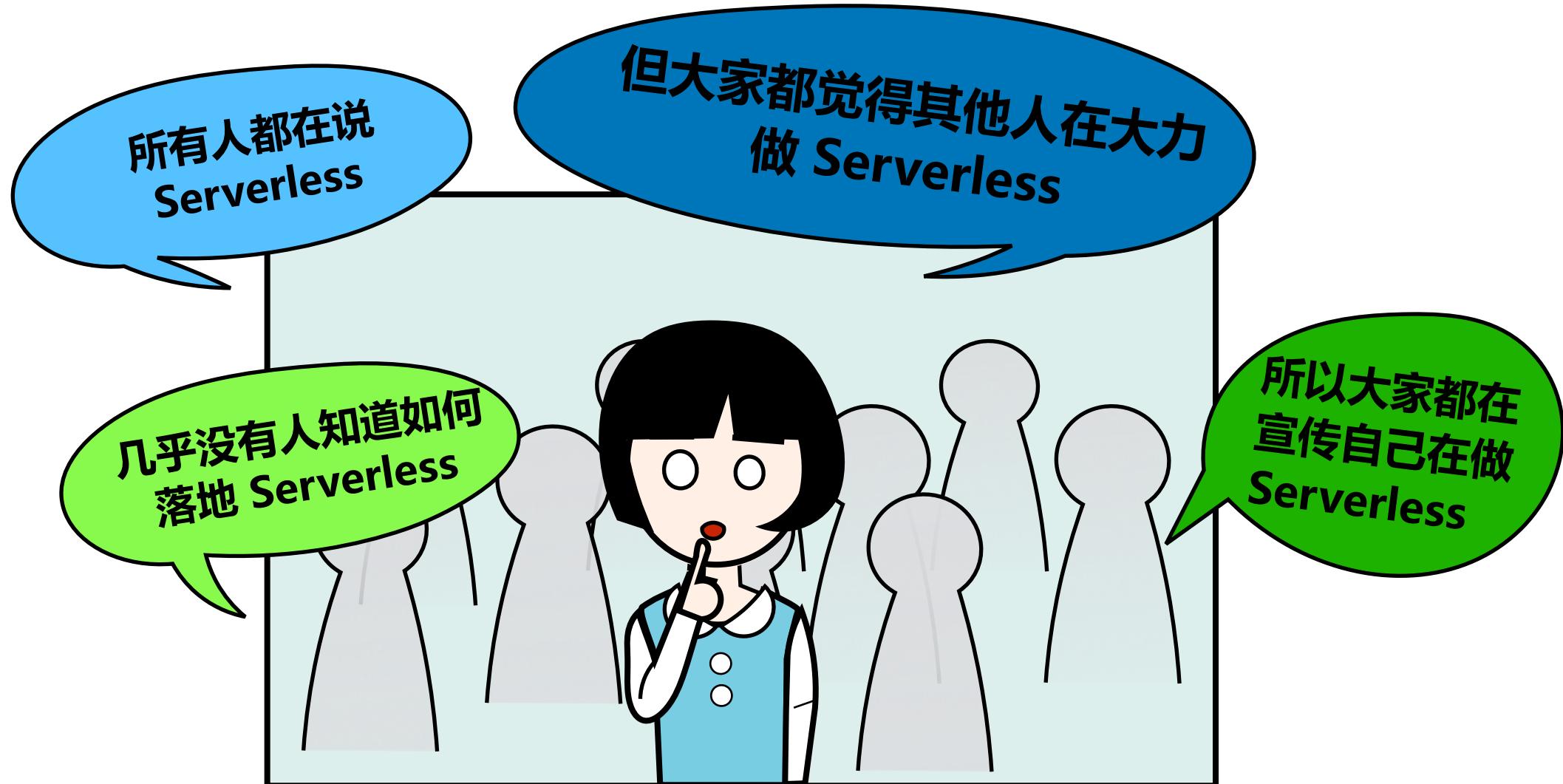
资源利用率低

在 BFF 场景下，服务器水位较低 ($<30\%$)，基于微服务的高可用诉求导致了资源的浪费。

蒸汽时代 → 电气时代



蒸汽时代 → 电气时代



回归初心：人机交互体验

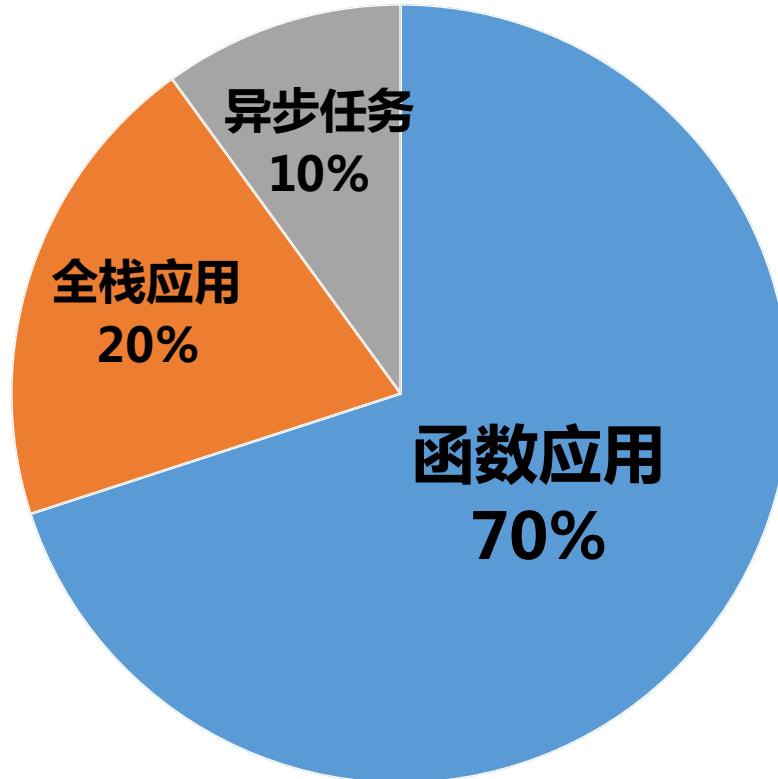
Serverless For Frontend

为 **高效** 地开发人机交互界面提供 **无感** 的服务能力。

护航者

SFF 是 BFF 的概念升级。PS：是不是觉得读起来很别扭？我们内部都读作『傻乎乎』，因为有个胡建人带头的。

未来不管怎么演变，与人机交互界面都需要这层 SFF，大致可以分为三类：



Function 函数应用	快速开发 领域模型 到 UI 模型 的转换。
Application 全栈应用	全栈型应用，一杆到底。 (语雀)
Task 异步任务	覆盖高 CPU/MEM、执行周期较长的操作，如构建任务。

SFF 研发模式的核心点：不增加额外成本的同时，提升前端效能和能力范围。

一站式、弱流程

+

轻研发、免运维

+

场景化、云端一体化

贯穿整个研发流程的一站式研发体验，简化第三方接入流程。

让开发者关注于业务代码本身，无需关注运维，系统自动根据流量扩缩容和故障自愈。

前后端研发融合模式，降低理解和学习成本。

PS：不抢活，合理分工。不亲自做 Serverless 底层，而是基于 Serverless 基建做上层研发模式。

电气时代

一站式、弱流程

sffdemo / serverlessbff / 测试日志无盘

迭代

缺陷

迭代成员

迭代设置

测试日志无盘 Function 应用
该迭代没有描述

代码分支: sprint_serverlessbff_S907861791_20191017 复制分支名 创建 MR

零式于 4 天前 提交 280c3680 安全扫描: 通过 覆盖率: 92.5% 查看 CI 详情

✓ 开发阶段 > ✓ 测试阶段 > 3. 预发阶段

完成预发

✓ 开发环境 ✓ 测试环境 ✓ 预发环境

19 小时前 苏依 在 覆盖率测试 部署代码 a4bce30a 成功 查看日志

代码分支: sprint_serverlessbff_S907861791_20191017
访问域名: https://serverlessbff-test.alipay.net
运行日志: 请选择 查询

函数设置

TR

功能简介: tr 服务
是否开启:
是否开启 traas:

筛选服务方 关键词筛选 搜索

服务方	接口名	版本号
gpbusinesscenter	com.alipay.gpbusinesscenter.api.TestDemoRefdFacade	UT0606_gpbusinesscenter_method
gpbusinesscenter	com.alipay.gpbusinesscenter.api.TransferAccountServiceFacade	UT0530_gpbusinesscenter_method

Basement

功能简介: Basement BaaS 服务客户端
是否开启:

默认配置 开发 测试 灰度 生产

appid: 576a222aec2a926e92819b39
环境: prod

前置校验 日志 构建 日志 部署 日志 部署 ...

简化第三方接入流程。

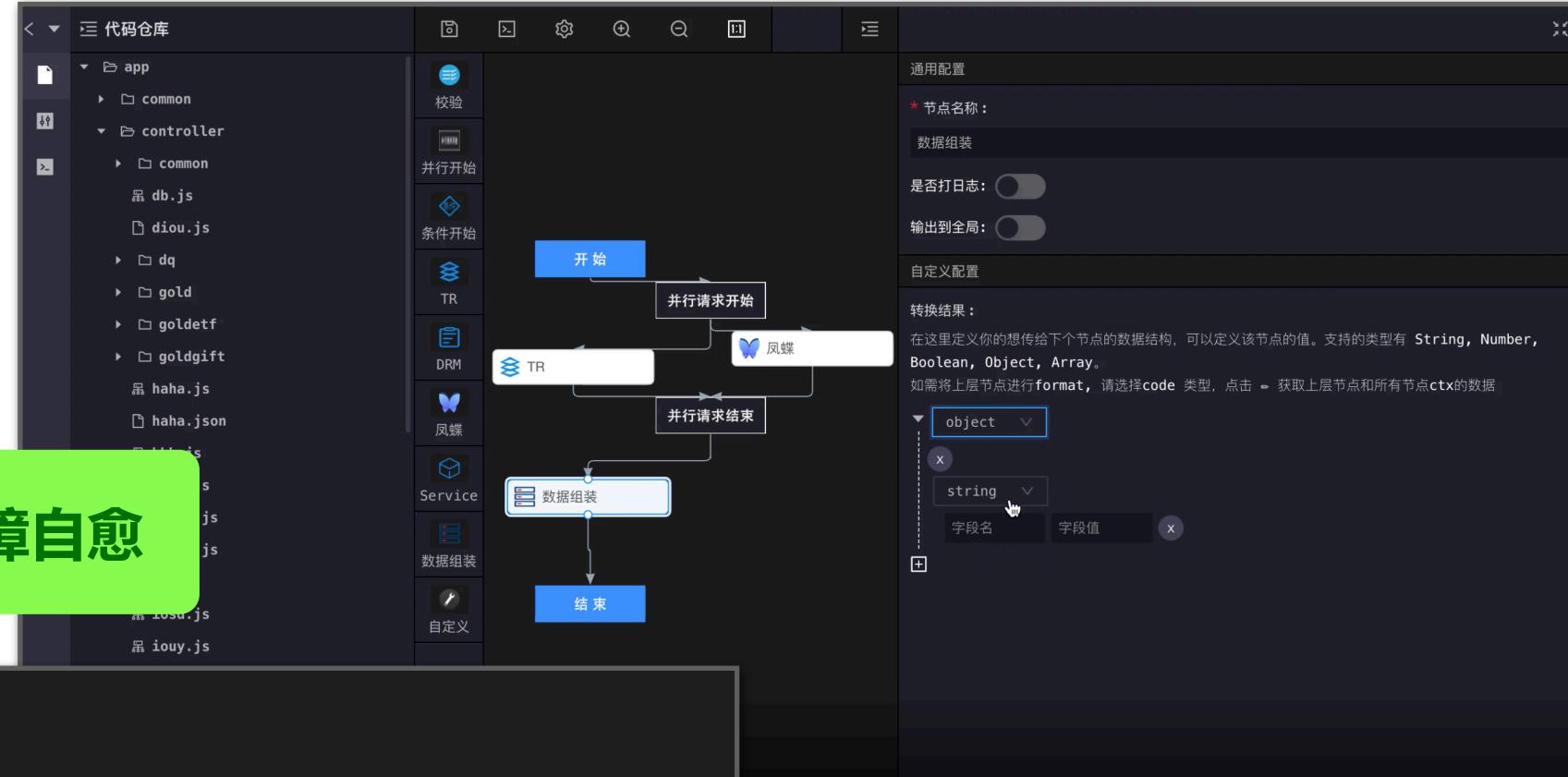
贯穿整个研发流程的一站式研发体验

电气时代

轻研发、免运维

根据流量自动扩缩容、故障自愈

```
3 /**
4  * 查询用户信息
5  *
6  * @param {Object} ctx 上下文
7  * @param {String} id - 用户 ID
8  * @return {Object} 用户信息
9  */
10 exports.findUserInfo = async function(ctx, id) {
11   const userInfo = await ctx.proxy.userInfoQuery.findUserInfoByLogonId(id);
12   return userInfo;
13 };
14 
```



轻研发，让开发者关注于业务代码本身

场景化、云端一体化

中台场景

营销场景

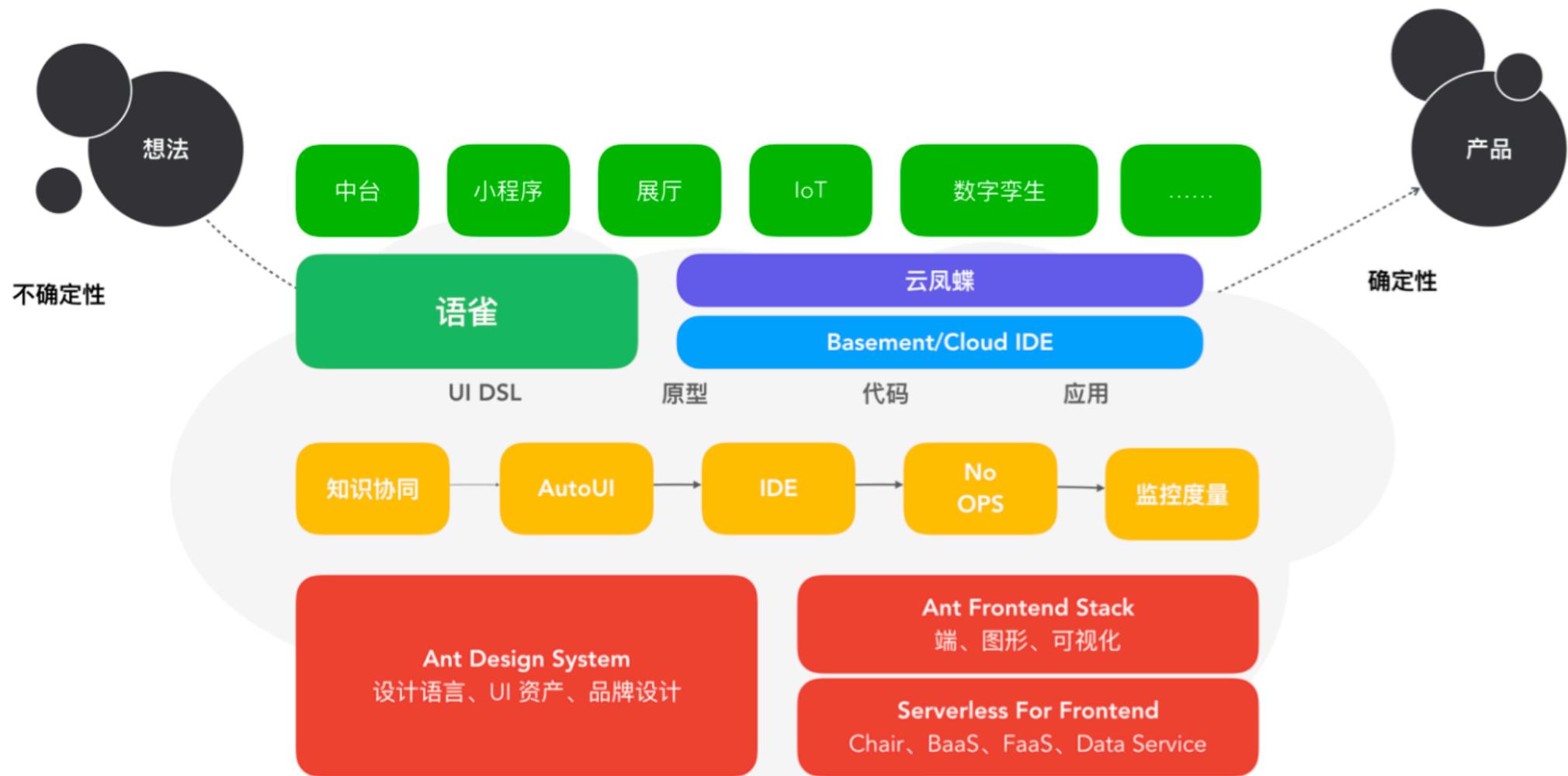
IoT 场景

...

```
3  /**
4   * 查询用户信息
5   *
6   * @param {Object} ctx 上下文
7   * @param {String} id - 用户 ID
8   * @return {Object} 用户信息
9   */
10  exports.findUserInfo = async function(ctx, id) {
11    const userInfo = await ctx.proxy.userInfoQuery.findUserInfoByLogonId(id);
12    return userInfo;
13  };
14
```

```
17  // src/model/user.js
18  // 前端调用
19  import user from '@service/rpc/user';
20
21  await user.listUser(5);
22
```

前后端研发融合模式，降低理解和学习成本。



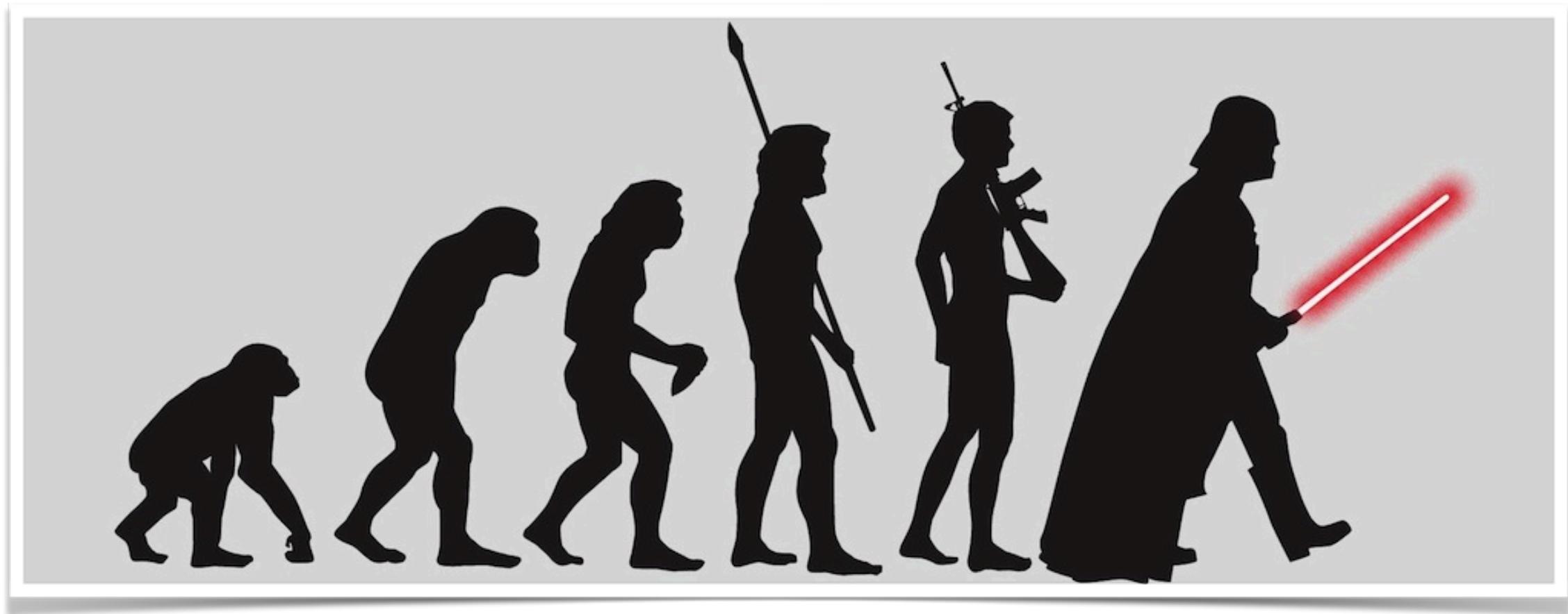
蚂蚁未来技术基础设施

Serverless、AI、IoT、计算、安全等

原子时代？星辰大海时代？？

We Are Here

天地初开 → 前端诞生 → 前端工程化 → 大前端 → 云端一体化 → ???





前端 = 人机交互工程师，负责人机交互层的领域研究与技术实现。

前端的核心价值，是完成技术的**最后一公里**，让技术可交付给用户使用，是科技触达用户的桥梁，是数字世界里的建筑工。

— 我们的所有决策，都应该以此为准则，来决策 做 or 不做。



WE ARE JUST **ON THE WAY**
THANK YOU.



这些年的体验技术部 · Node.js 基础服务



— 摸爬滚打才不负功名尘土

分享蚂蚁 Node.js 的演化史

<https://zhuanlan.zhihu.com/p/84176287>

SFF



阿里小程序云

参与到国内最前沿的
Node.js 基础措施建设。
Egg、cnpm、...

参与到阿里经济体唯一的一个
面向开发者的小程序云
Serverless



广州分部

