

如何打造易用的DevOps工具链

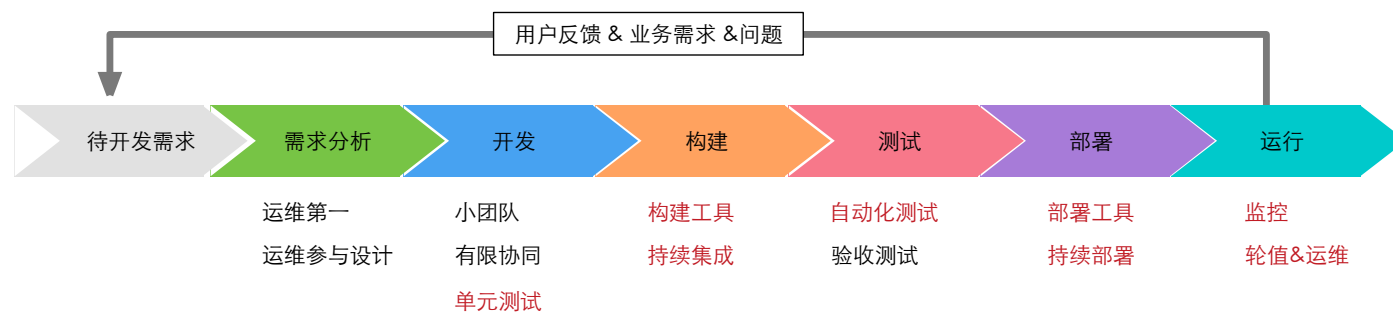
任发科（常新居士）

为什么讲这个主题？

理念

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

-DevOps : A Software Architect's Perspective

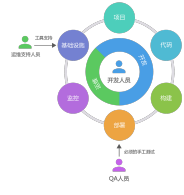


研发运维交付运行的系统

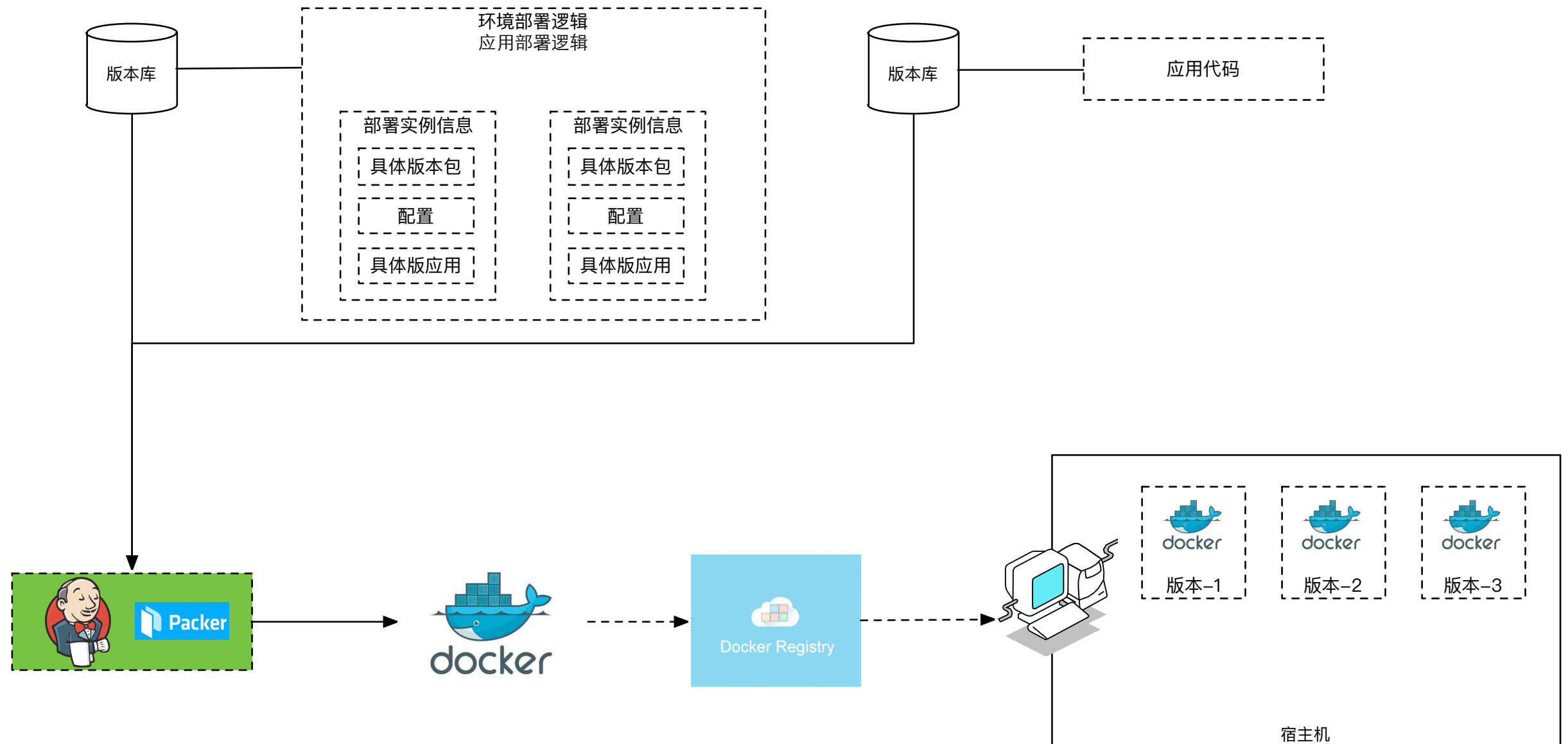


谁构建，谁运维

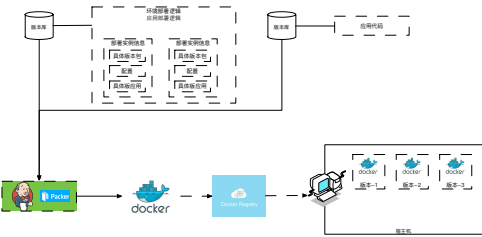
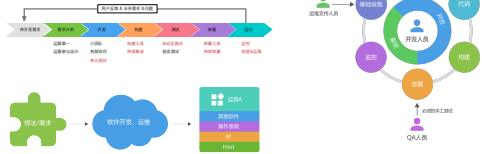
DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.
-DevOps - A Software Architect's Perspective



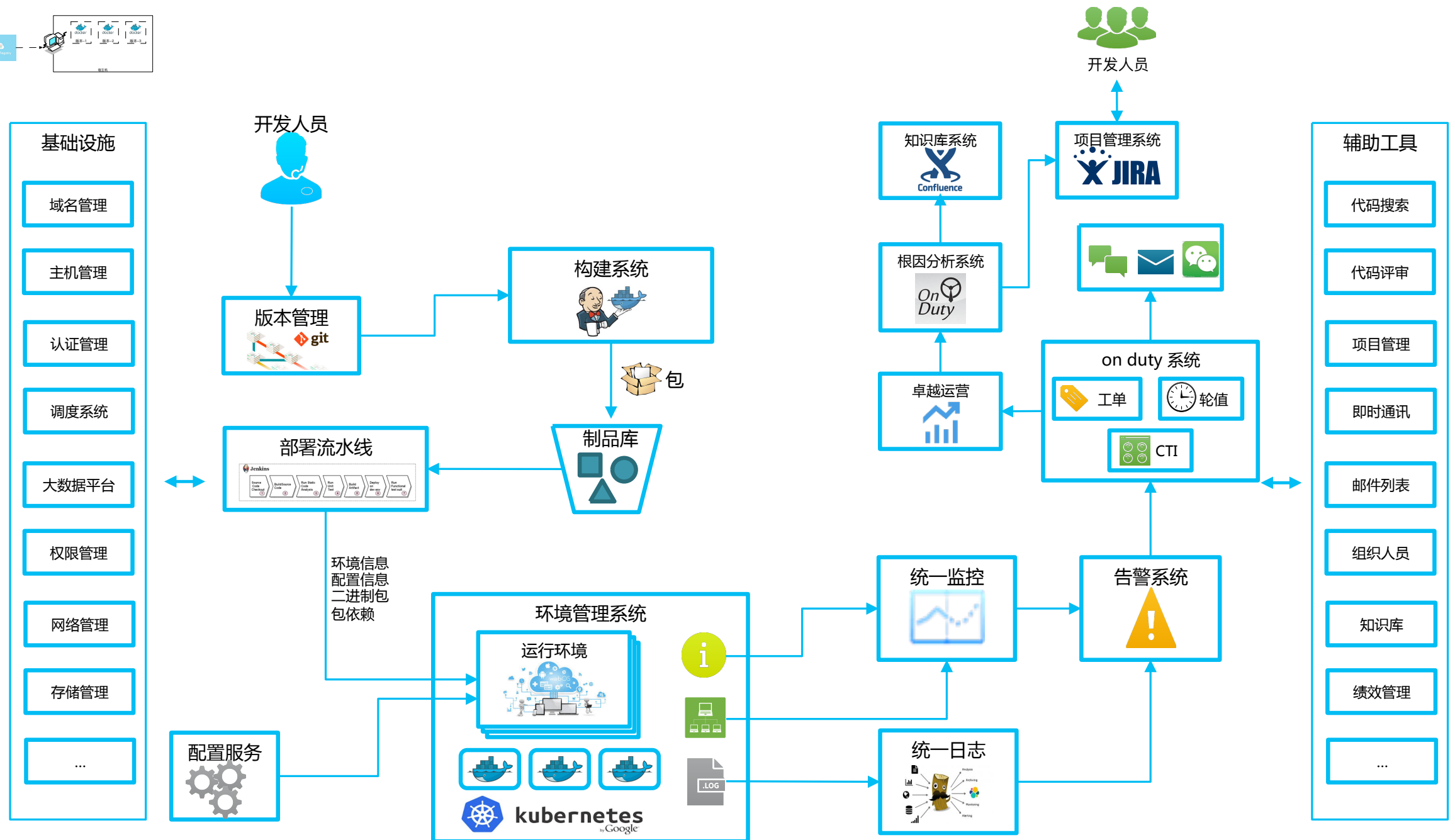
理念->体系

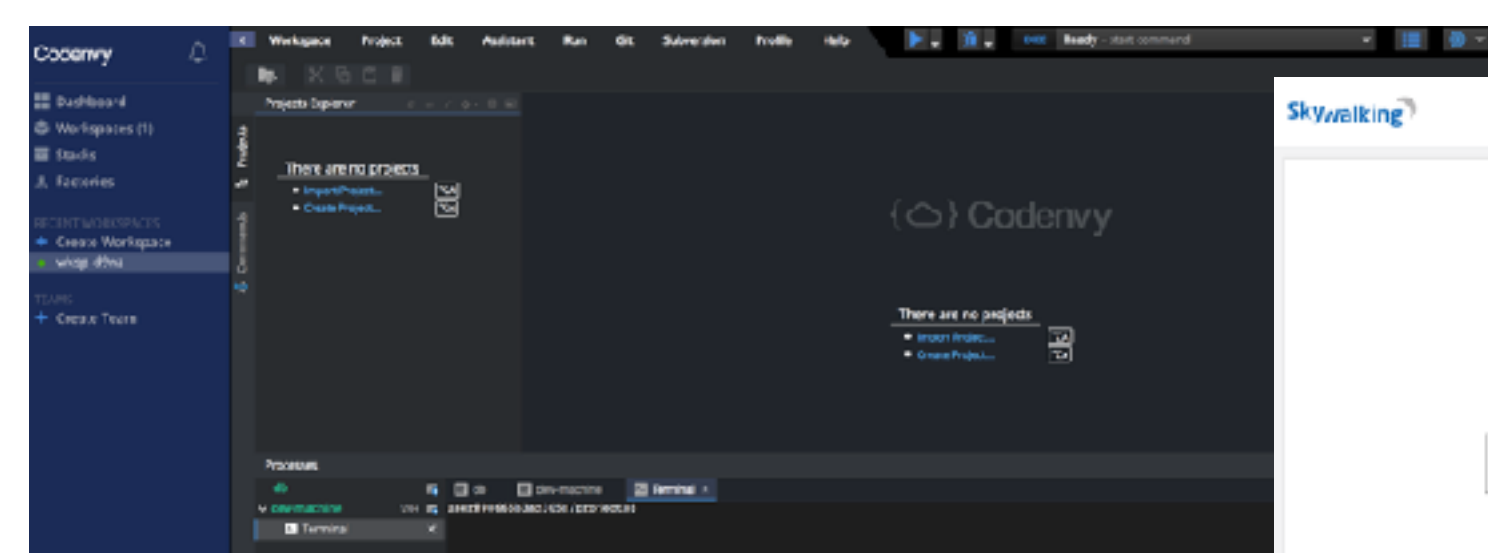


DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.
-DevOps - A Software Architect's Perspective

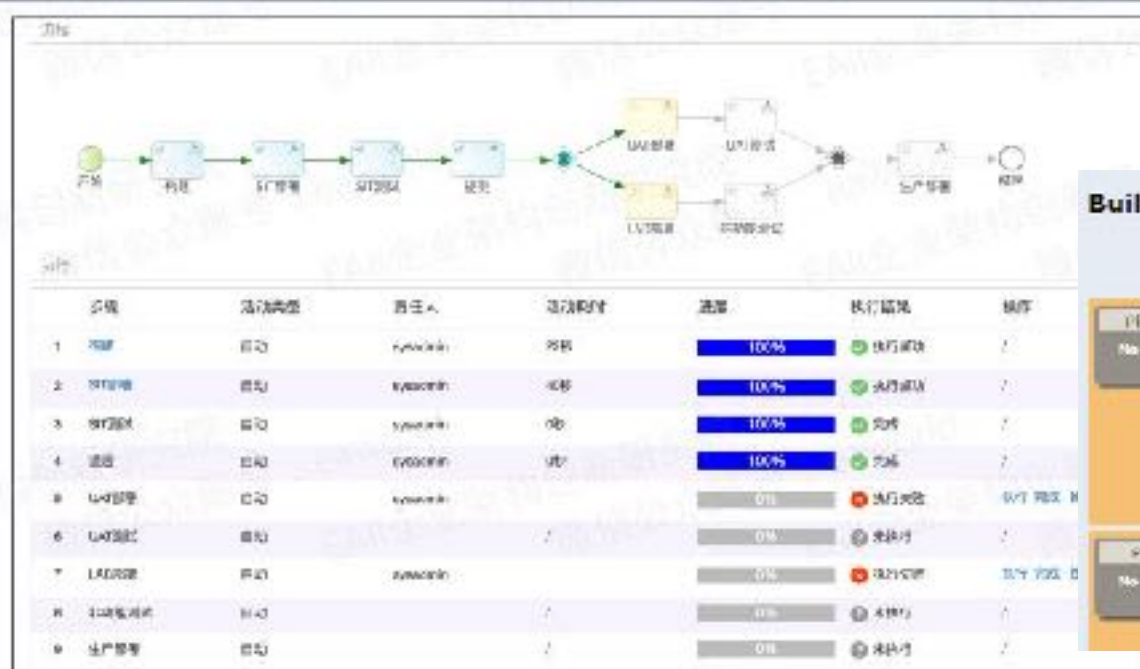


理念->体系





看板 — 面向团队的交付协作

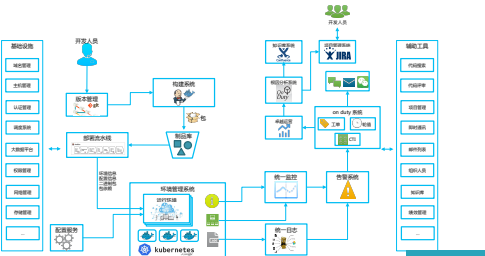
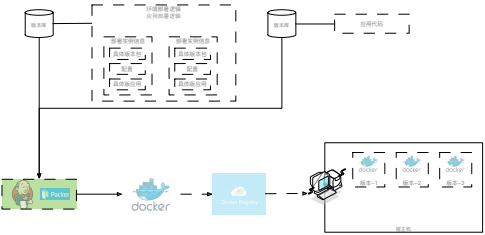
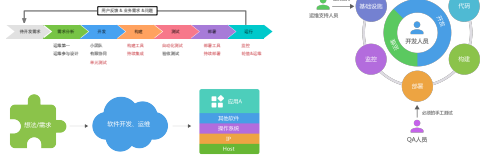


耗时	集群名称	Git标签	说明	执行者	结果
1 分钟	test	1.0.6	test-again	easyops	成功
1 分钟	test	1.0.6	add	easyops	成功
3 分钟	test	1.0.6	test	easyops	失败

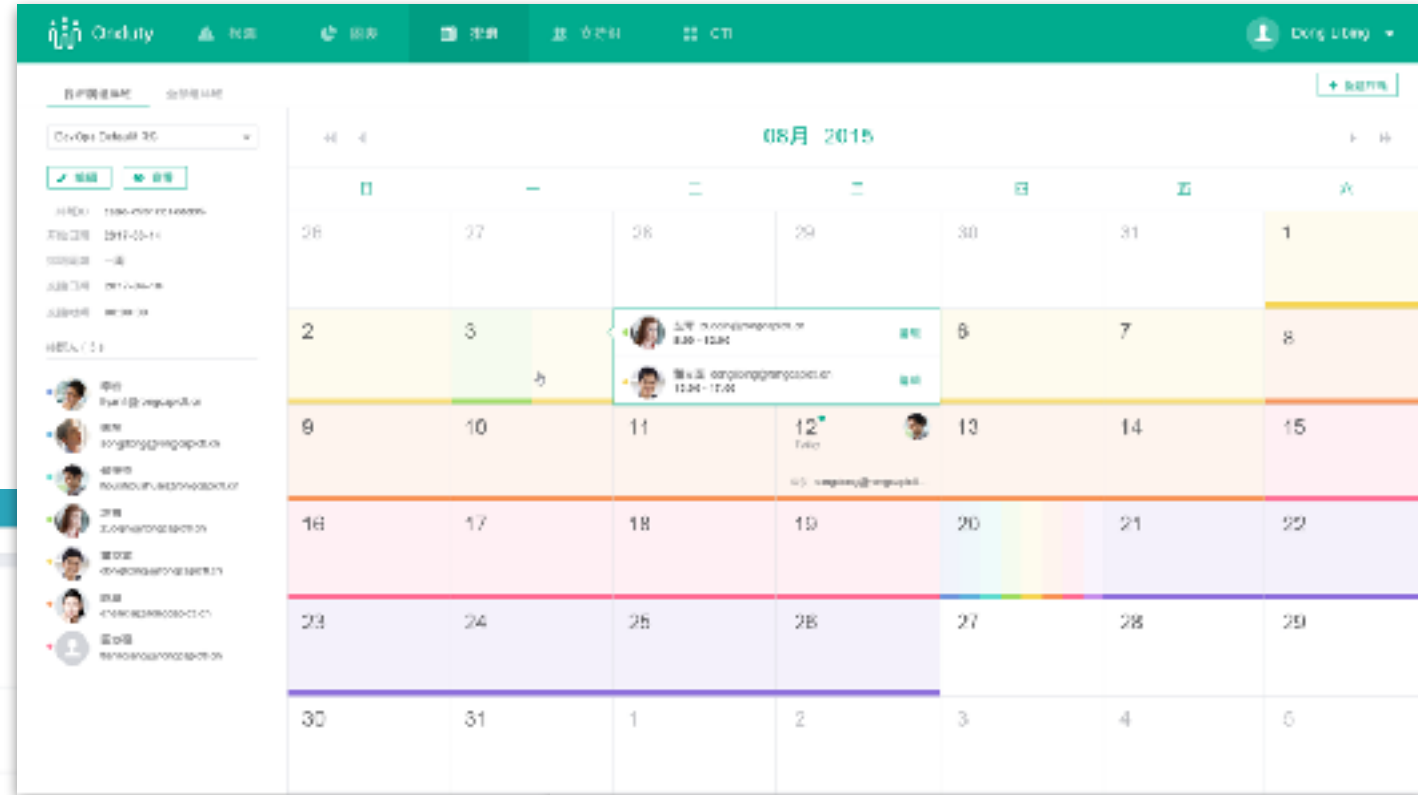
Build Pipeline



DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.
-DevOps - A Software Architect's Perspective



理念->体系->工具



Magna						
Order Service						
Basic Information Name: Order Service Owner: Order Service Manager: Order Service						
Packages This resource contains 11 packages						
Configurations Log: Order Service Log: Order Service Log: Order Service						
Stage View: Test: All: Gradient: All: 1						
The following table shows the configuration of the Order Service stage. The table is organized into columns for the stage name, the configuration name, the configuration value, and the configuration type.						
Stage Name	Configuration Name	Configuration Value	Configuration Type	Configuration Description	Configuration Action	Configuration Status
Order Service1	Order Service1	Order Service1	Order Service1	Order Service1	Order Service1	Order Service1
Order Service2	Order Service2	Order Service2	Order Service2	Order Service2	Order Service2	Order Service2
Order Service3	Order Service3	Order Service3	Order Service3	Order Service3	Order Service3	Order Service3
Order Service4	Order Service4	Order Service4	Order Service4	Order Service4	Order Service4	Order Service4
Order Service5	Order Service5	Order Service5	Order Service5	Order Service5	Order Service5	Order Service5
Log Log: Order Service Log: Order Service Log: Order Service						
The following table shows the log of the Order Service stage. The table is organized into columns for the log name, the log value, and the log type.						
Log Name	Log Value	Log Type	Log Description	Log Action	Log Status	Log Time
Order Service1	Order Service1	Order Service1	Order Service1	Order Service1	Order Service1	Order Service1
Order Service2	Order Service2	Order Service2	Order Service2	Order Service2	Order Service2	Order Service2
Order Service3	Order Service3	Order Service3	Order Service3	Order Service3	Order Service3	Order Service3
Order Service4	Order Service4	Order Service4	Order Service4	Order Service4	Order Service4	Order Service4
Order Service5	Order Service5	Order Service5	Order Service5	Order Service5	Order Service5	Order Service5

Endpoint

可以指定一个或多个关键字
 例如: my-id=example-123

支持以下格式:

☐

刷新counter列表

Counter

可以指定一个或多个关键字
 例如: my-id=example-123

☐

刷新counter列表

一些问题不断地被提及

系统为什么要这样架构？
软件设计为什么要选择不完美的方案？
同样的问题为什么要研发多种工具？
为什么屡试不爽的设计理念不起作用？

.....

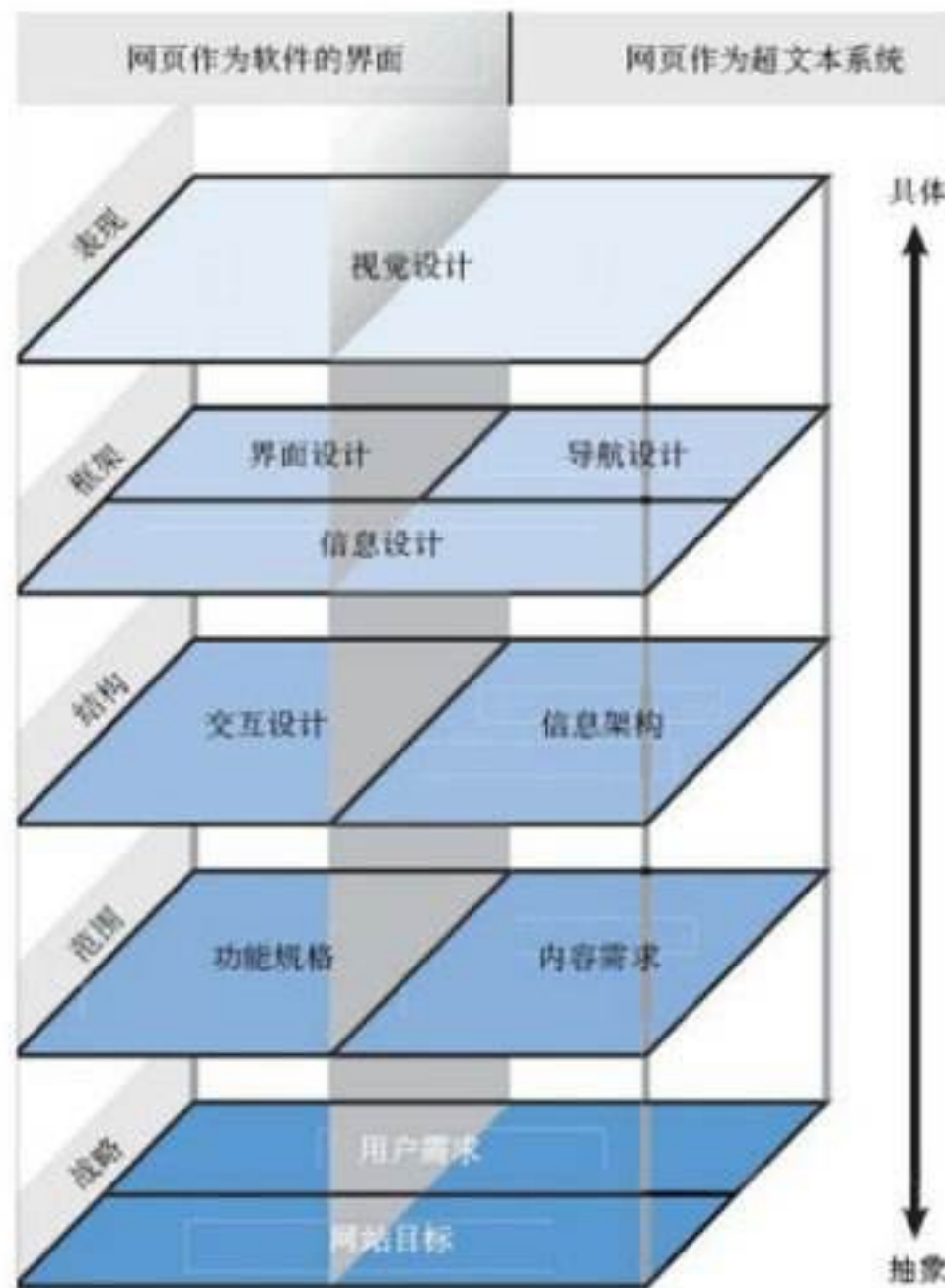
隐藏在这些问题的共性是什么？

易

易用

我们在DevOps落地和工具链研发过程中
遇到的问题以及解决问题的所思所做

简易、易用为什么重要？



表现层 (surface)

视觉设计，或者说最终产品的外观。

框架层 (skeleton)

不管是软件界面还是信息空间，我们必须完成信息设计 (information design)。在软件产品那边，框架层还包括了安排好能让用户与系统的功能产生互动的界面元素。对于信息空间方面来讲，这种界面就是屏幕上的一些元素的组合，允许用户在信息架构中穿行。

结构层 (structure)

在软件方面，结构层将从范围转变成系统如何响应用户的请求。在信息空间方面，结构层则是信息空间中内容元素的分布。

范围层 (scope)

从战略层进入范围层以后，在软件方面它就转变成对产品的“功能组合”的详细描述。而在信息空间方面，范围则是对各种内容元素的要求的详细描述。

战略层 (strategy)

来自企业外部的用户需求 (user need) 是网站的目标—尤其是那些将要使用我们网站的用户。与用户需求相对应的，是我们自己对网站的期望目标。

用户体验要素

易用

≠

易用





≠



2B



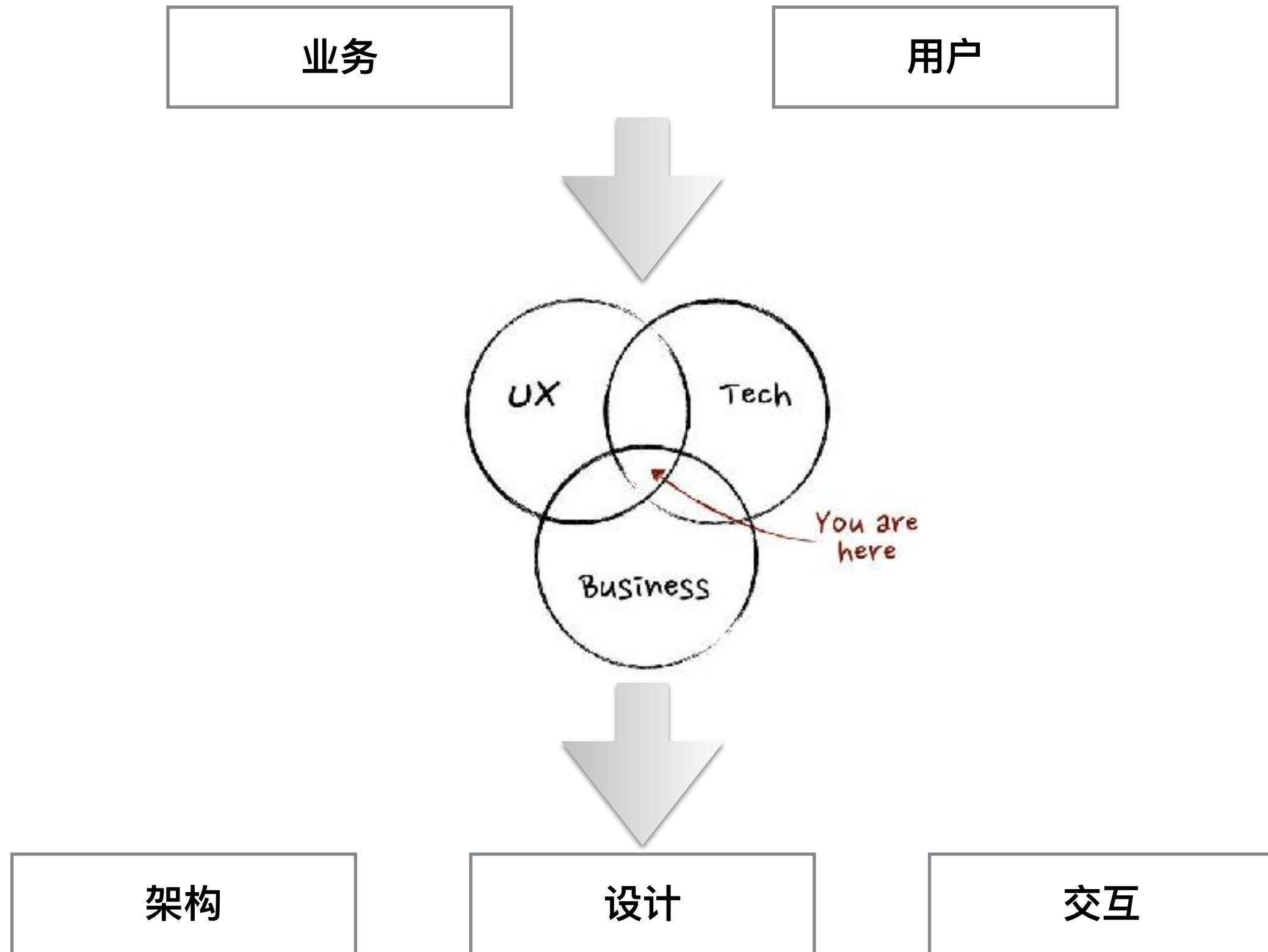
≠

2C



直接影响解决义务问题的效率

谷歌拼音 shu ru fa Google
1. 输入法 2. 输入 3. 书 4. 数 5. 输

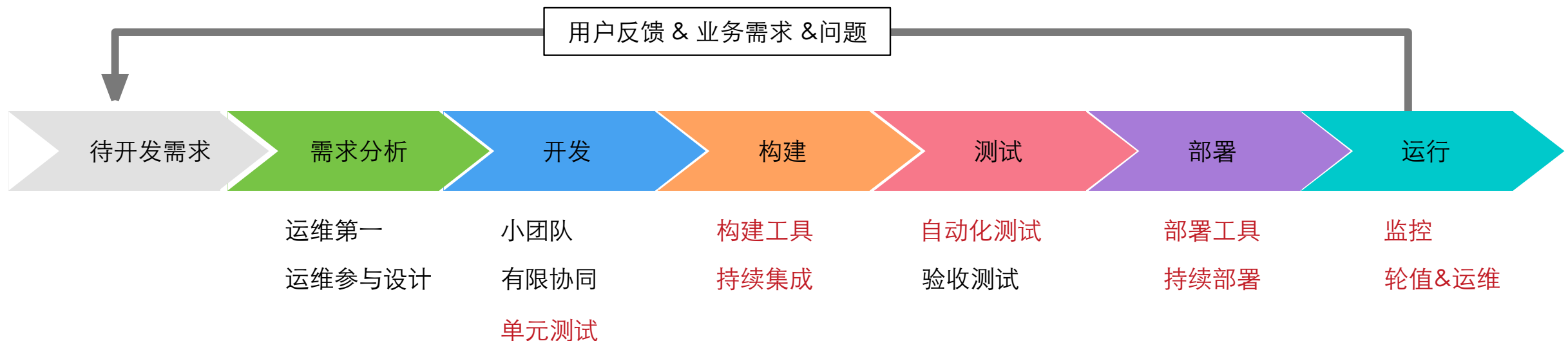


业务

全生命周期管理

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

-DevOps : A Software Architect's Perspective



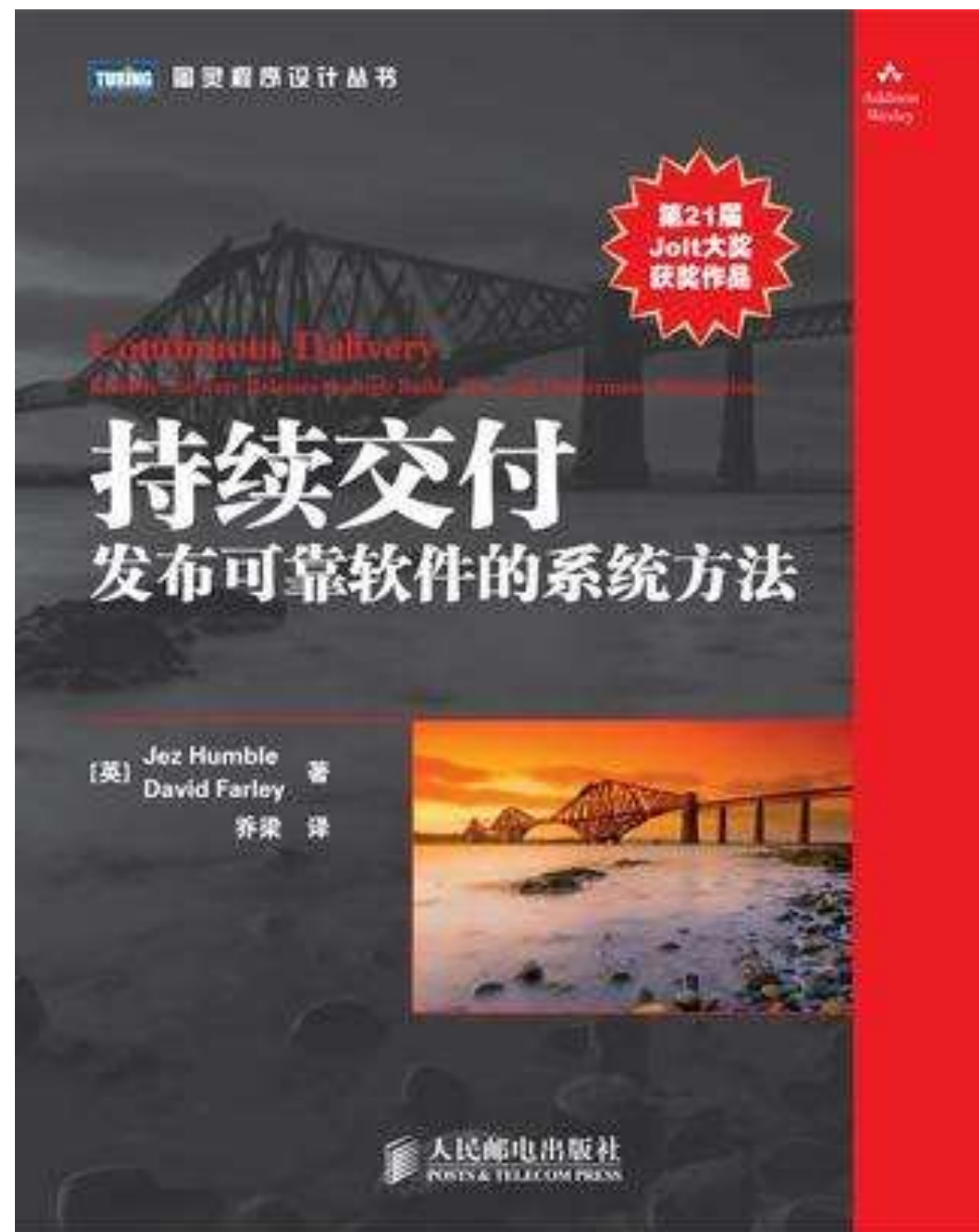
全生命周期管理

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

-DevOps : A Software Architect's Perspective

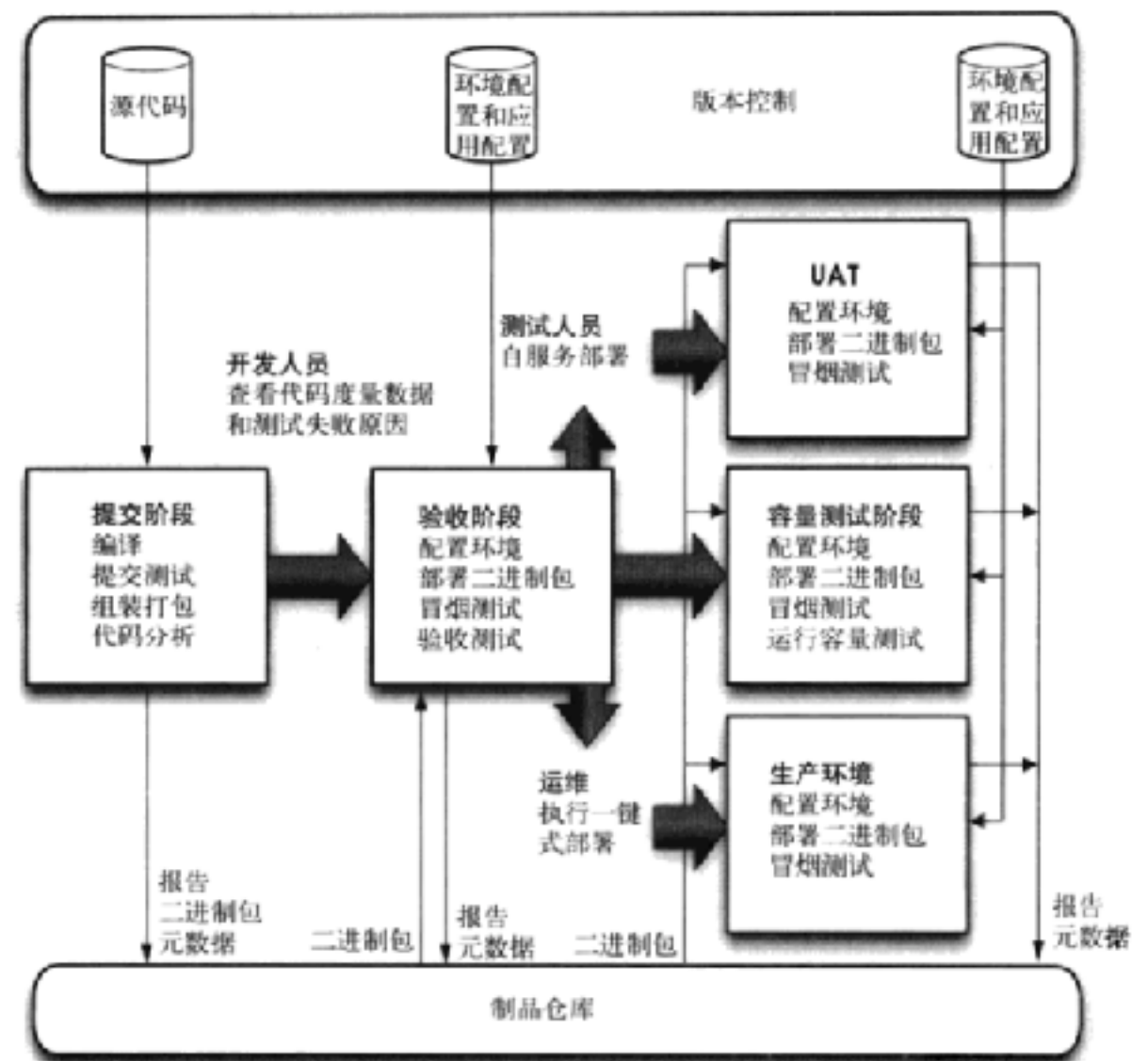


部署流水线



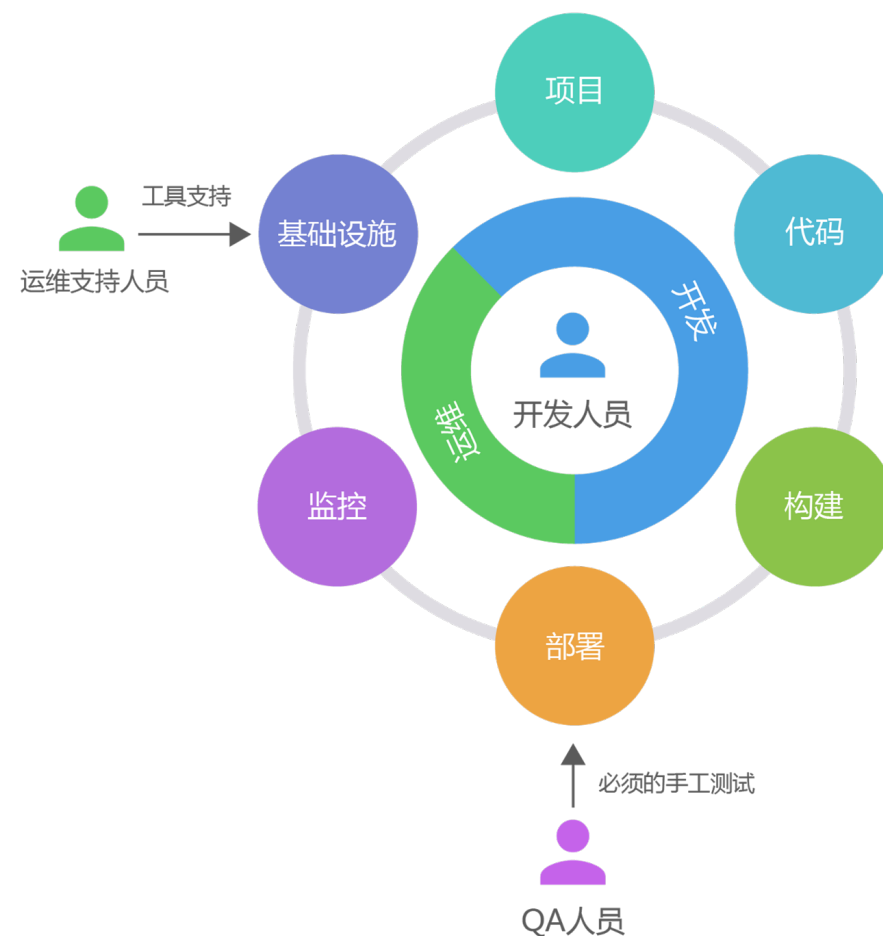
部署流水线

- 只生成一次二进制包
- 测试或运维人员手工选择需要的某个版本，将其部署到相应的环境中
- 操作人员应该可以看到所有构建版本和它们的状态（通过的阶段、包含的修改、提交注释等）
- 一切都是为了尽快得到反馈
- 必须能够看到每个环境中都部署了哪个版本
- 建立全程的追溯性



用户

研发人员完成一切



谁构建，谁运维

技术人员的一些特点

自视甚高

技术人员



最接近神的存在

非技术人员眼中



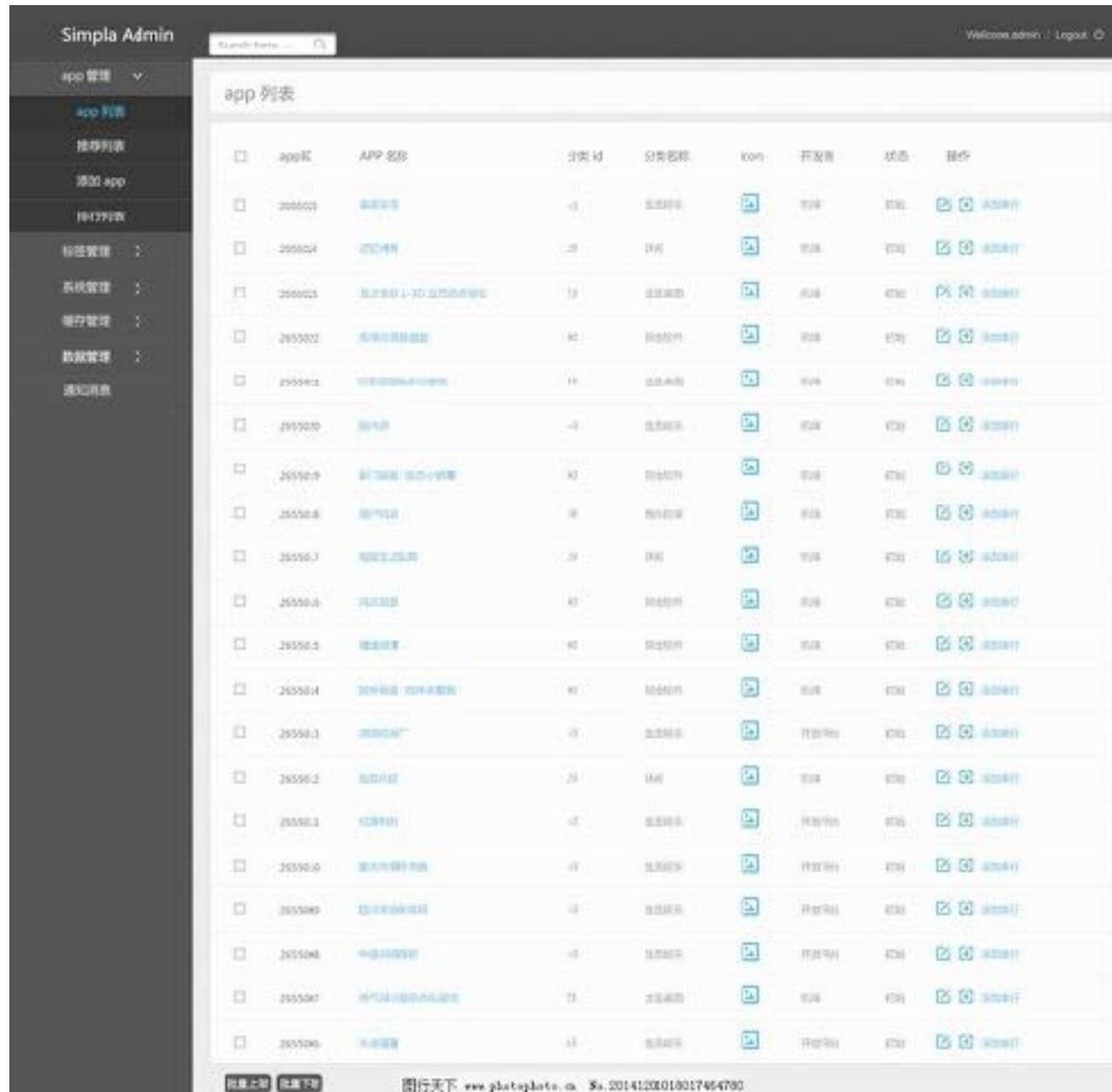
神

自己眼中



大神

有些事标准极低——能用就行



并行处理大量信息



有些事极较真

There are two types of people.

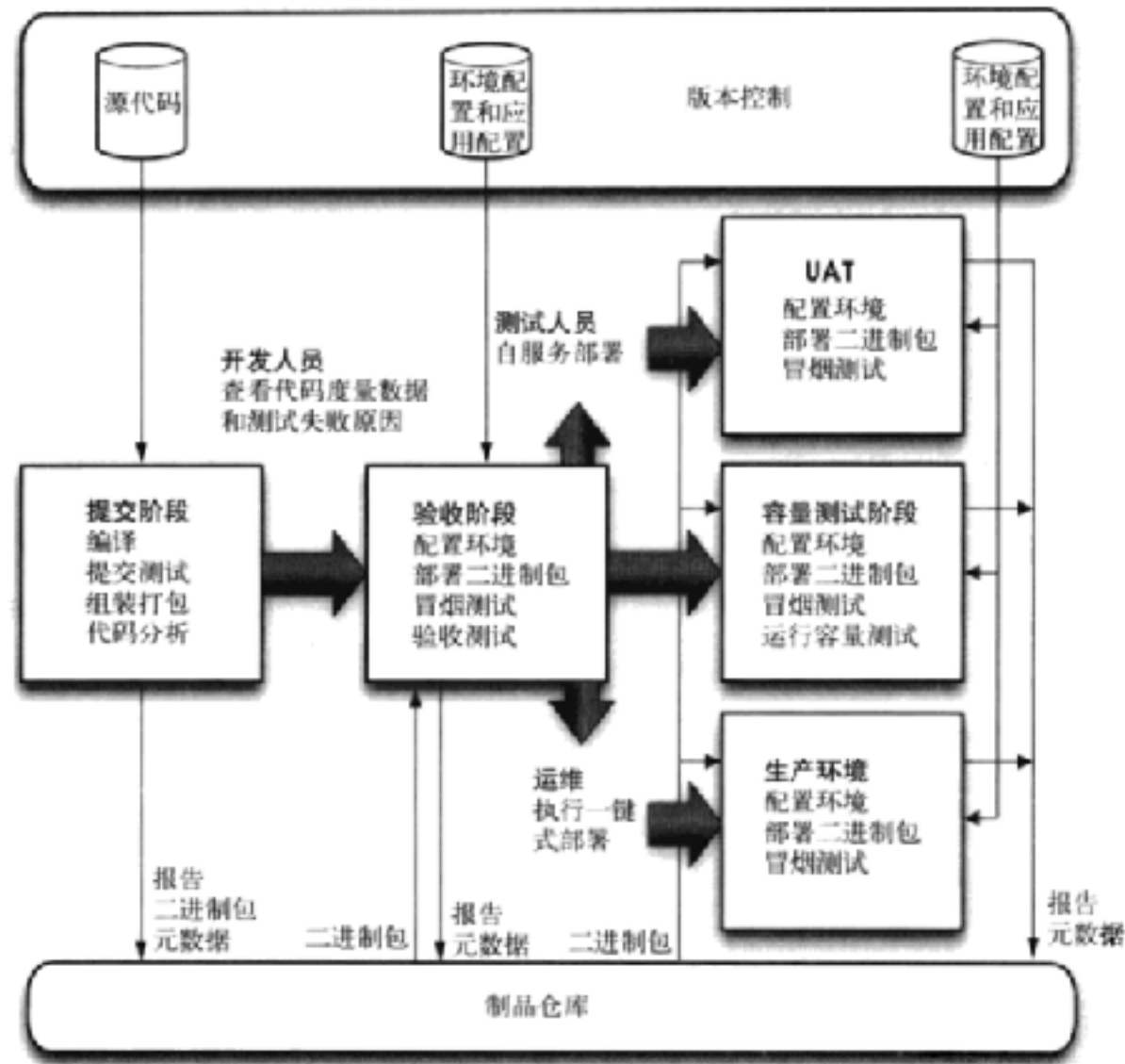
```
if (Condition)
{
    Statements
    /*
     *
     */
}
```

```
if (Condition) {
    Statements
    /*
     *
     */
}
```

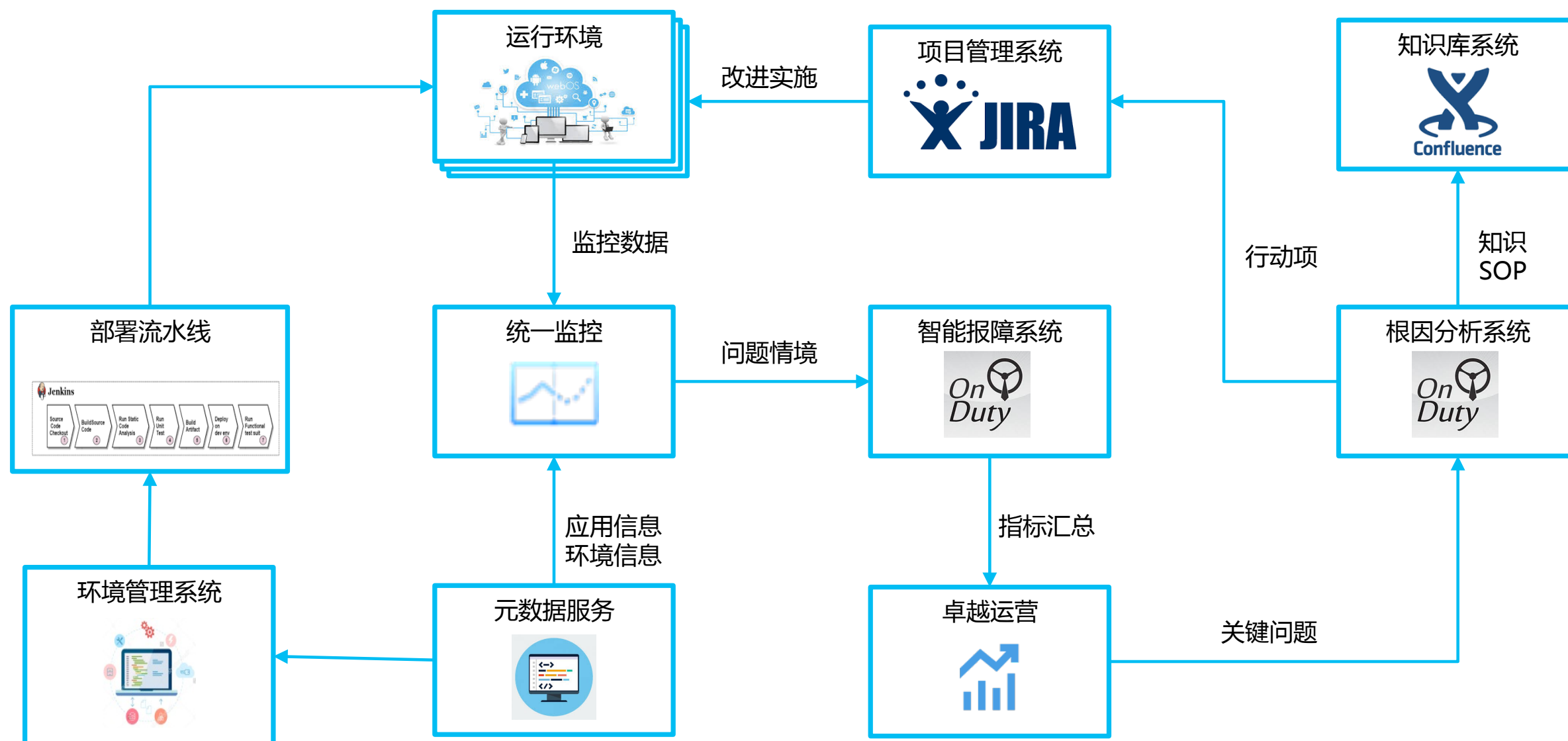
Programmers will know.

经验

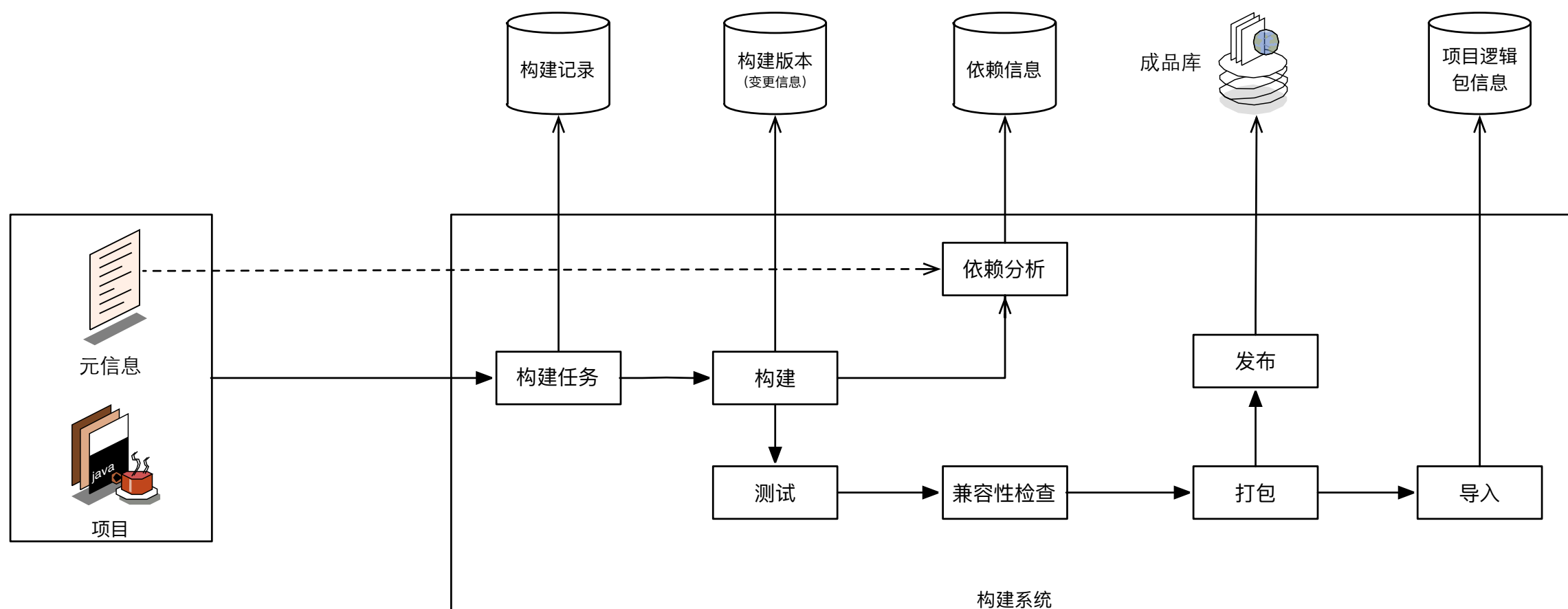
经验1：实现分，服务合



关注点分离，各自独立发展



有些系统有自己的复杂性



必要时可以通过视图集成



来源普元DevOps平台

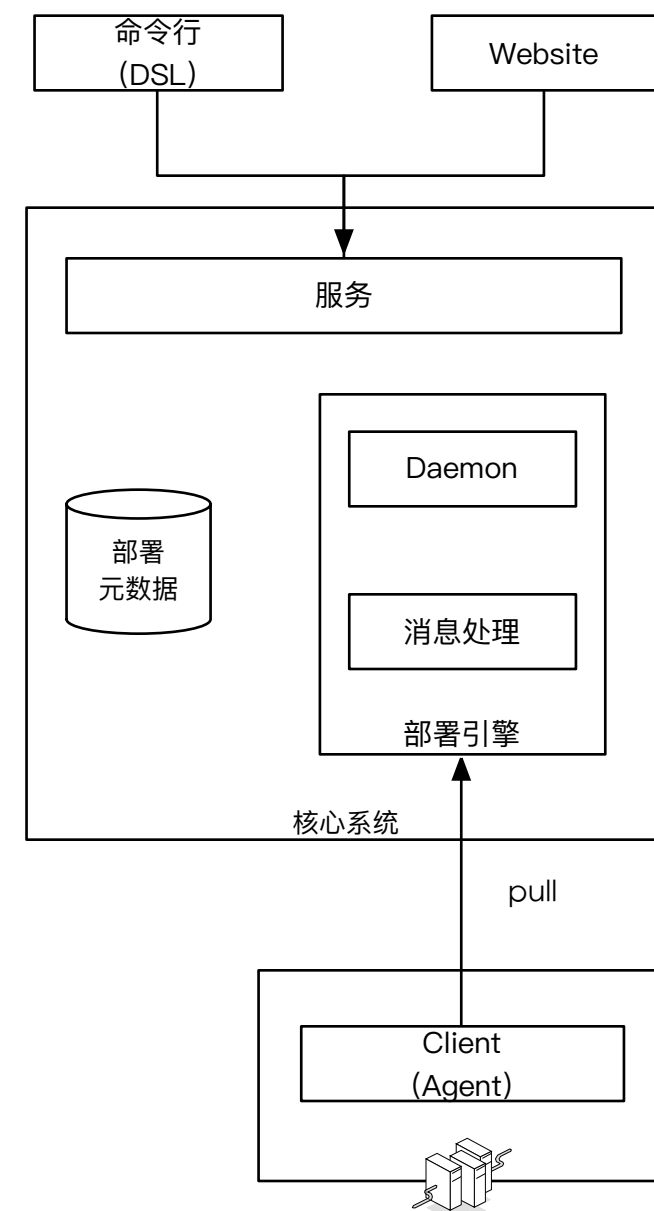
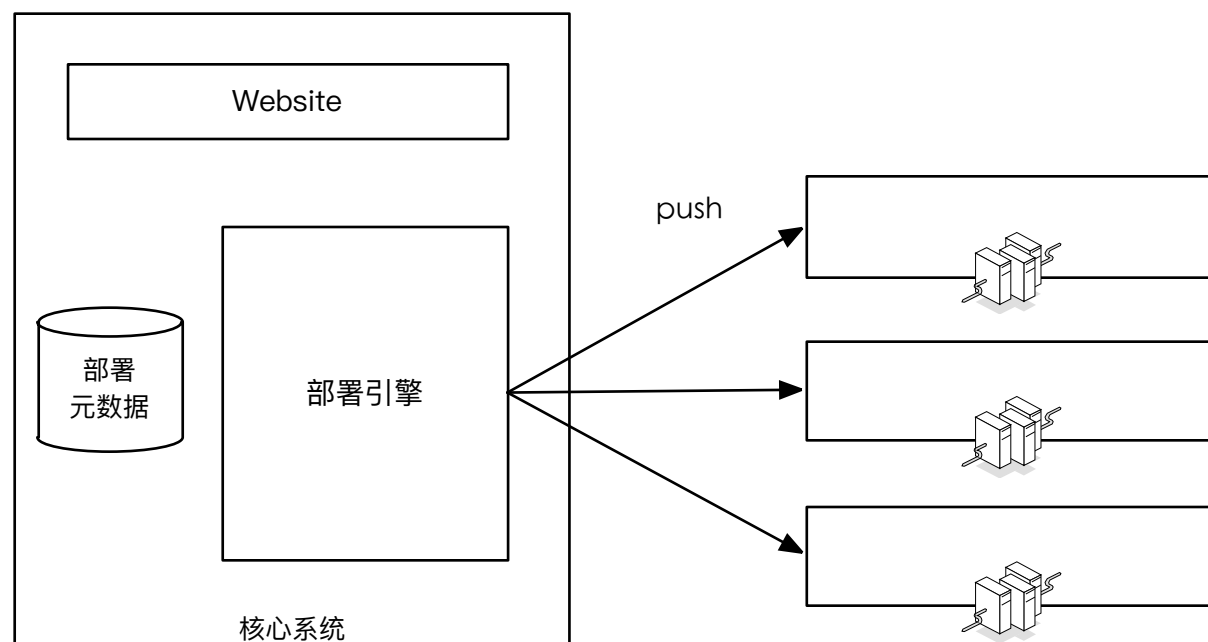
只做一件事，并且把这件事做好

10个系统做10件事比一个系统做10件事好

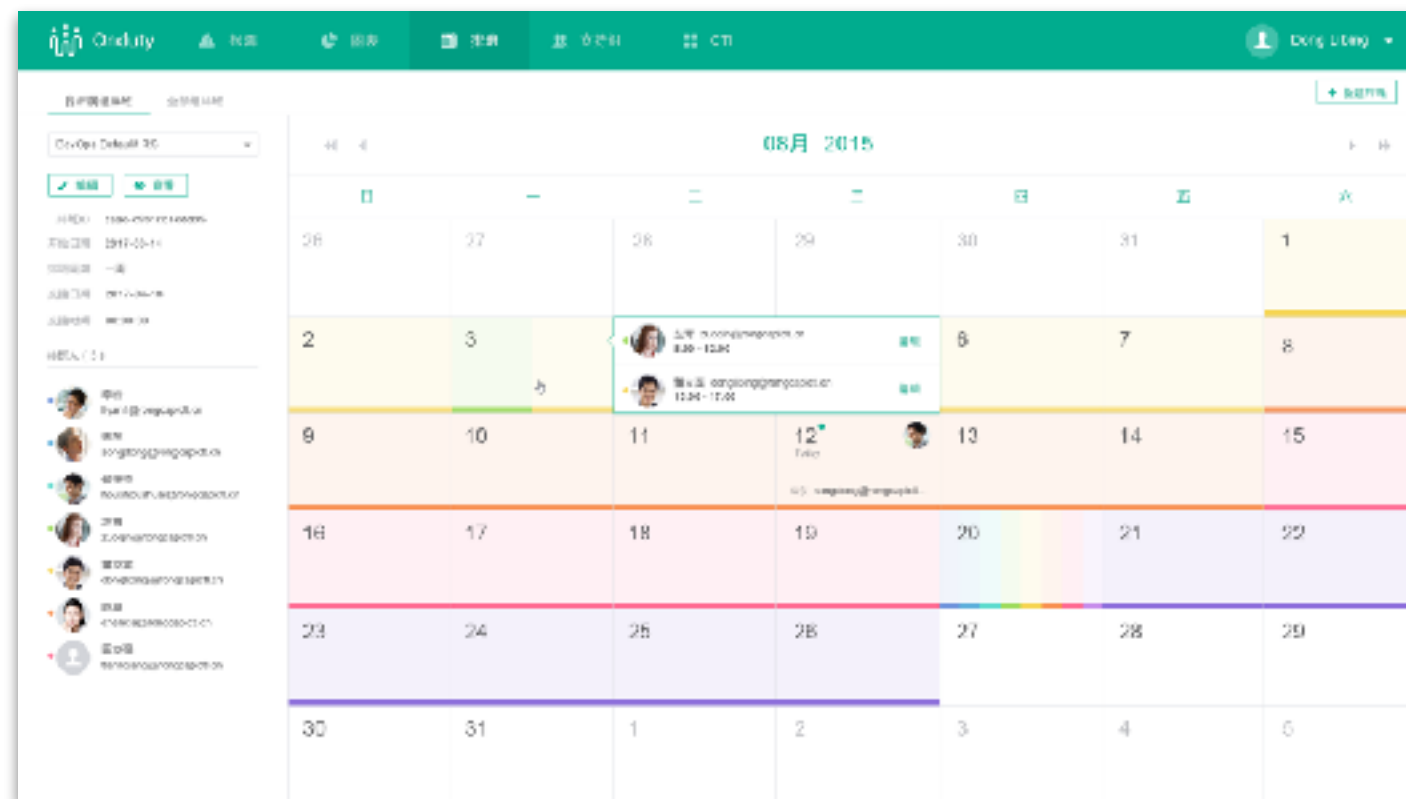
一站式系统大部分是反人类的！

只在真正需要时给出一站式的视图

经验2：设计上的关注点分离

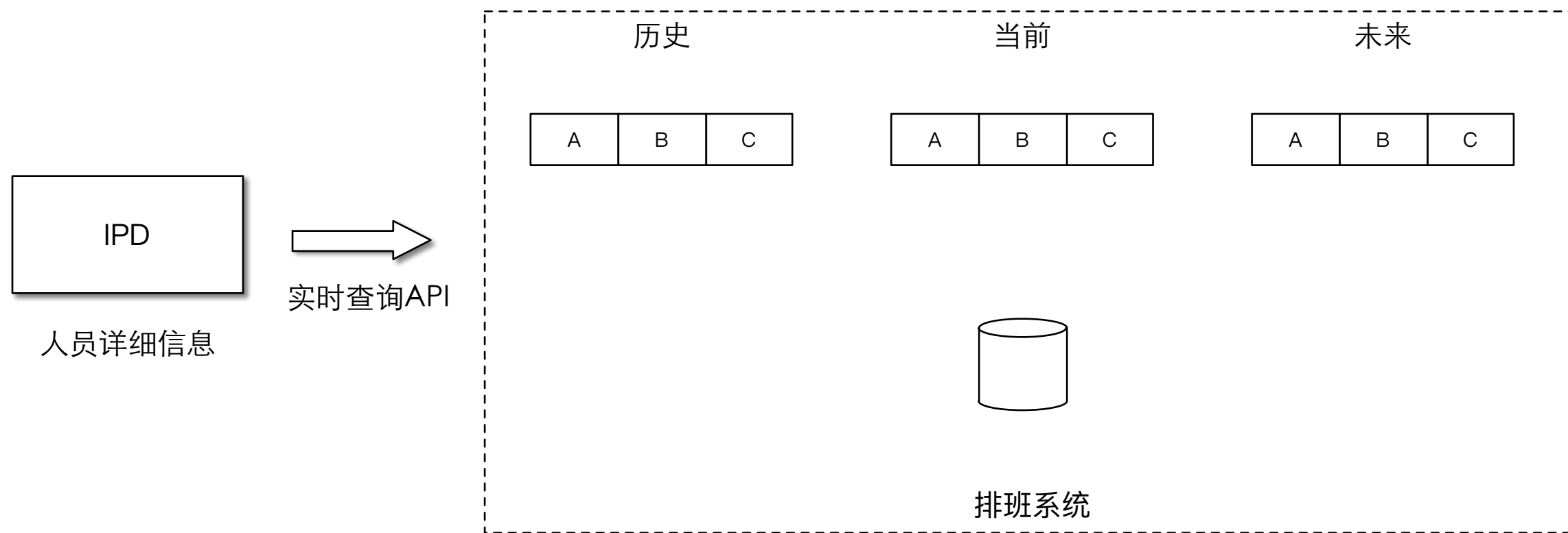


经验3：简单性 > 完美性



轮值排班更新

经验3：简单性 > 完美性

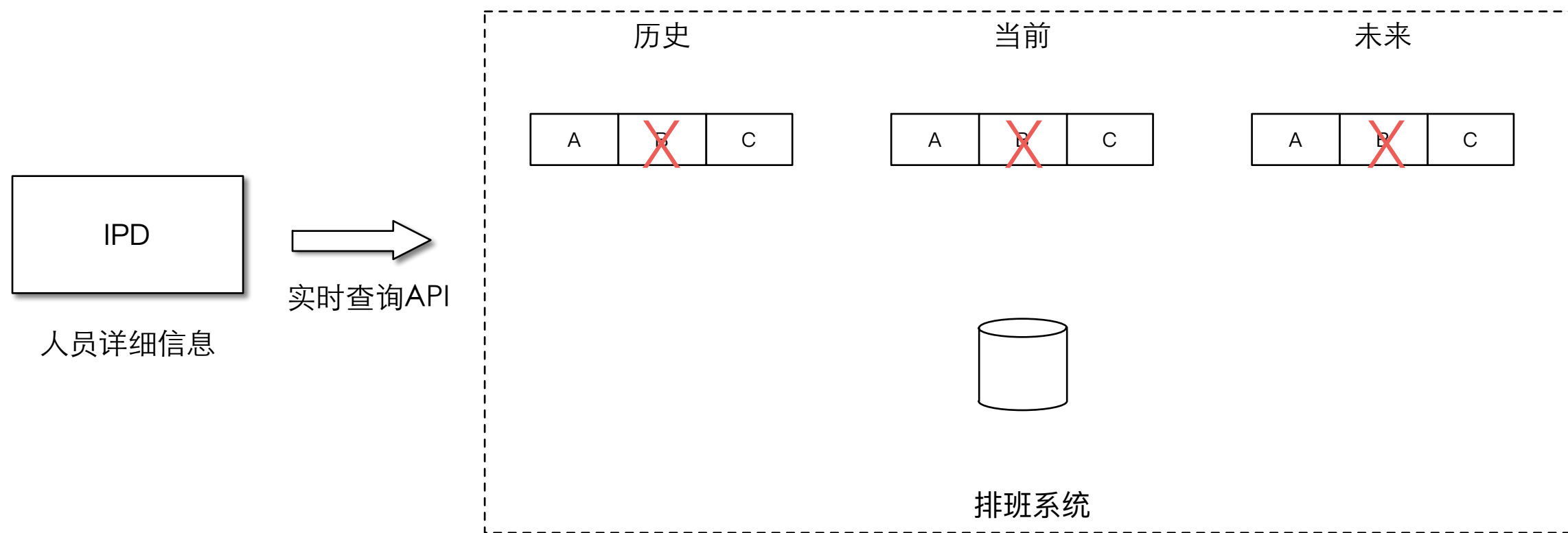


方案1：

追求人员信息的准确性

不考虑历史信息

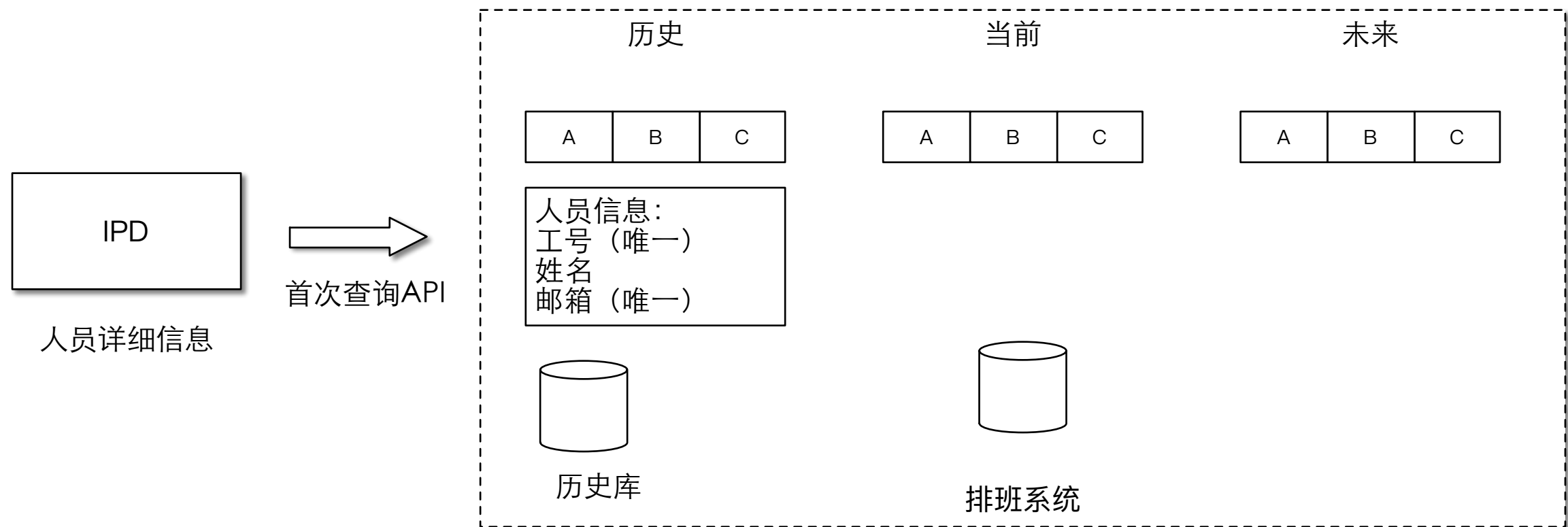
经验3：简单性 > 完美性



方案1：

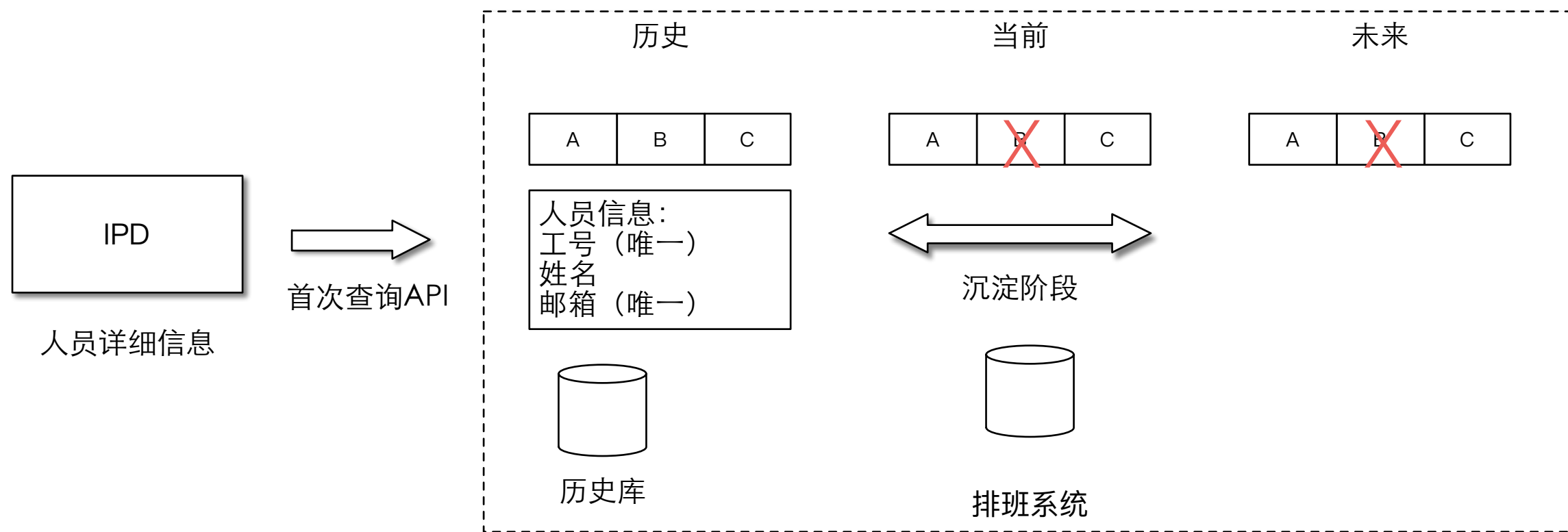
IPD人员删除、人员排班信息调整
无历史信息

经验3：简单性 > 完美性



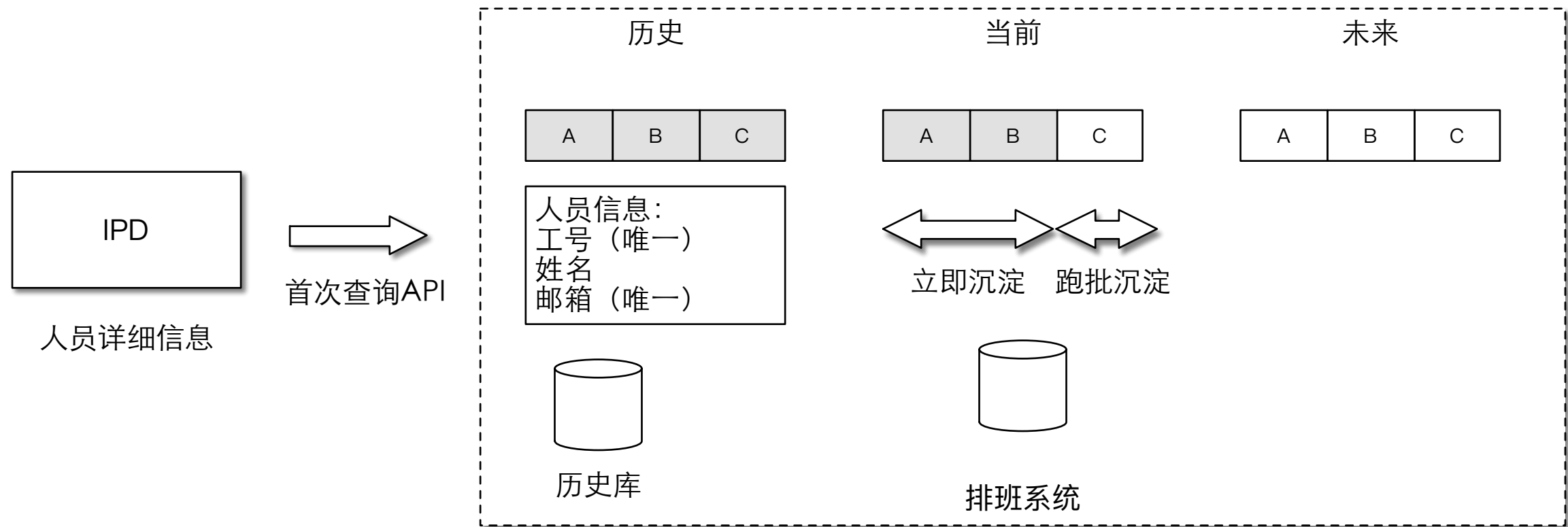
方案2：
无实时查询
历史数据跑批沉淀

经验3：简单性 > 完美性



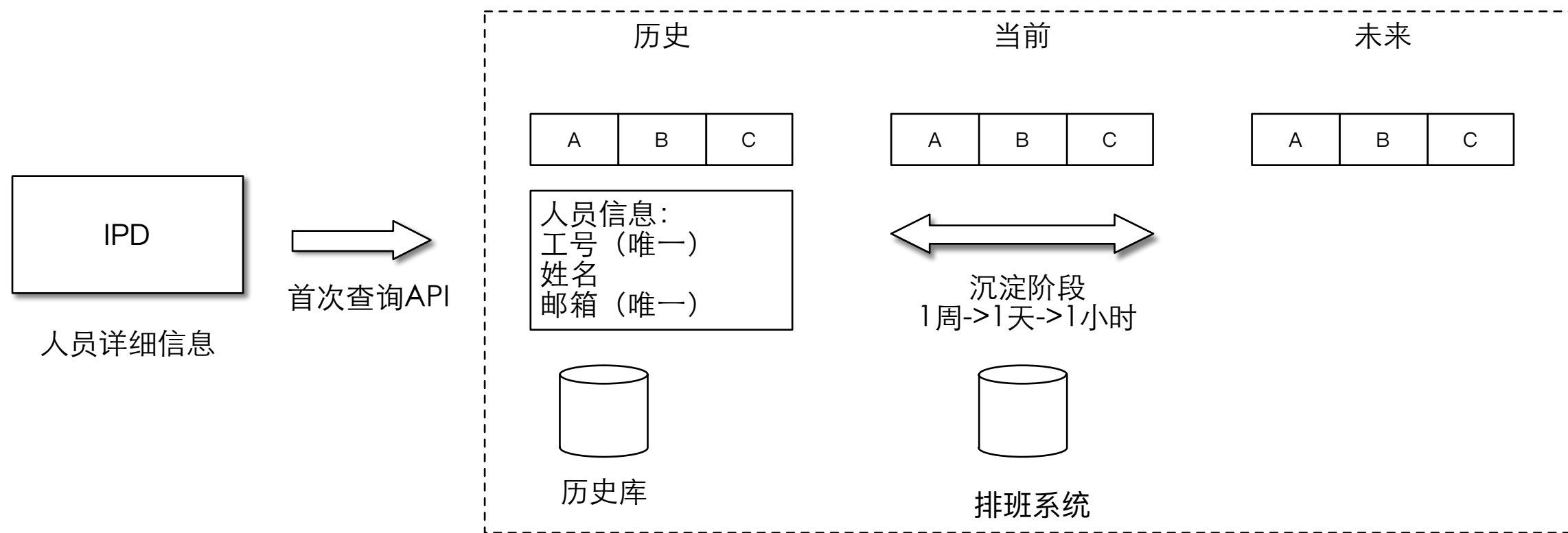
方案2：
沉淀阶段内的数据仍然不一致

经验3：简单性 > 完美性



方案3：
修改即沉淀

经验3：简单性 > 完美性

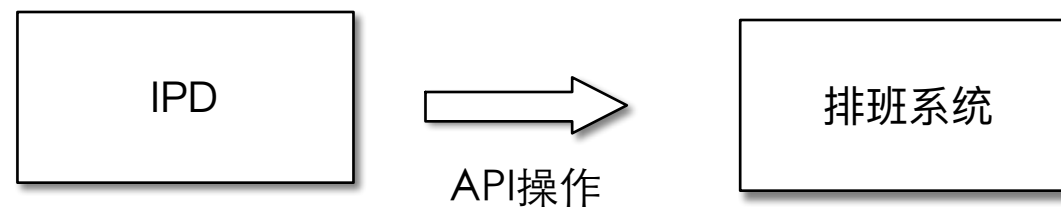


方案4：

缩短沉淀周期，缩小数据错误范围

方案3和方案4孰优？

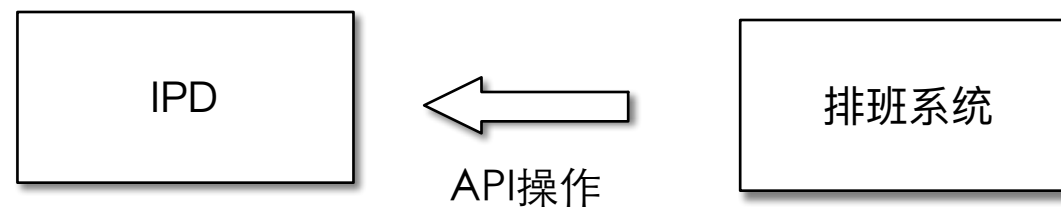
经验3：简单性 > 完美性



方案1：

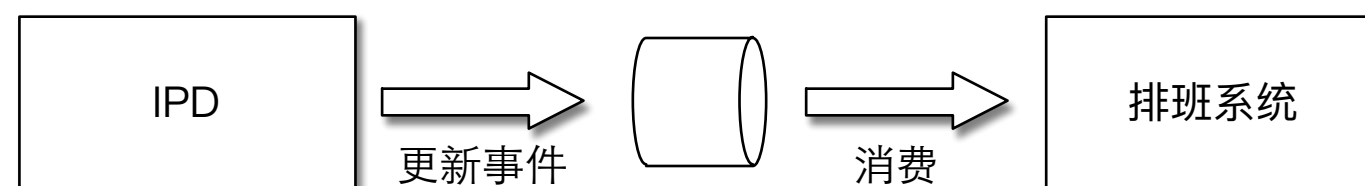
IPD删除/更改一个人员

经验3：简单性 > 完美性



方案2：
轮询信息

经验3：简单性 > 完美性



方案3：
加消息队列

经验3：简单性 > 完美性

IPD

排班系统

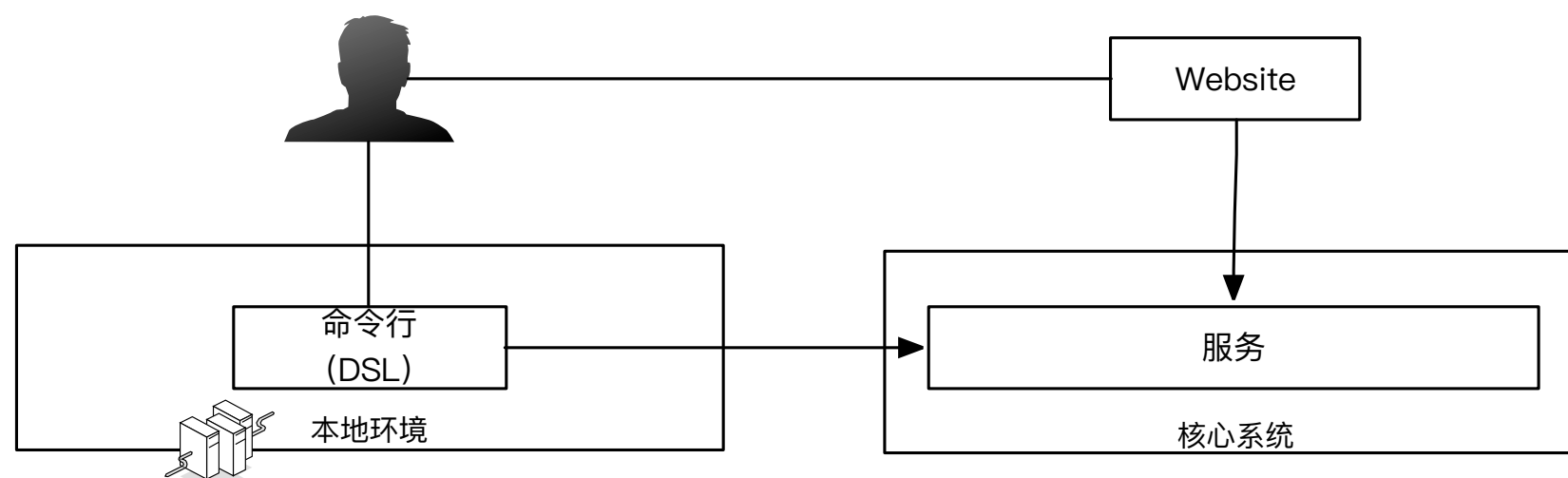


经理会知道的！

方案4：
人肉调整

方案3和方案4孰优？

经验4：关注高级使用



命令行 (DSL)：

扩展性更好

效率更高

经验4：关注高级使用

防止来自脚本的诱惑！！！！

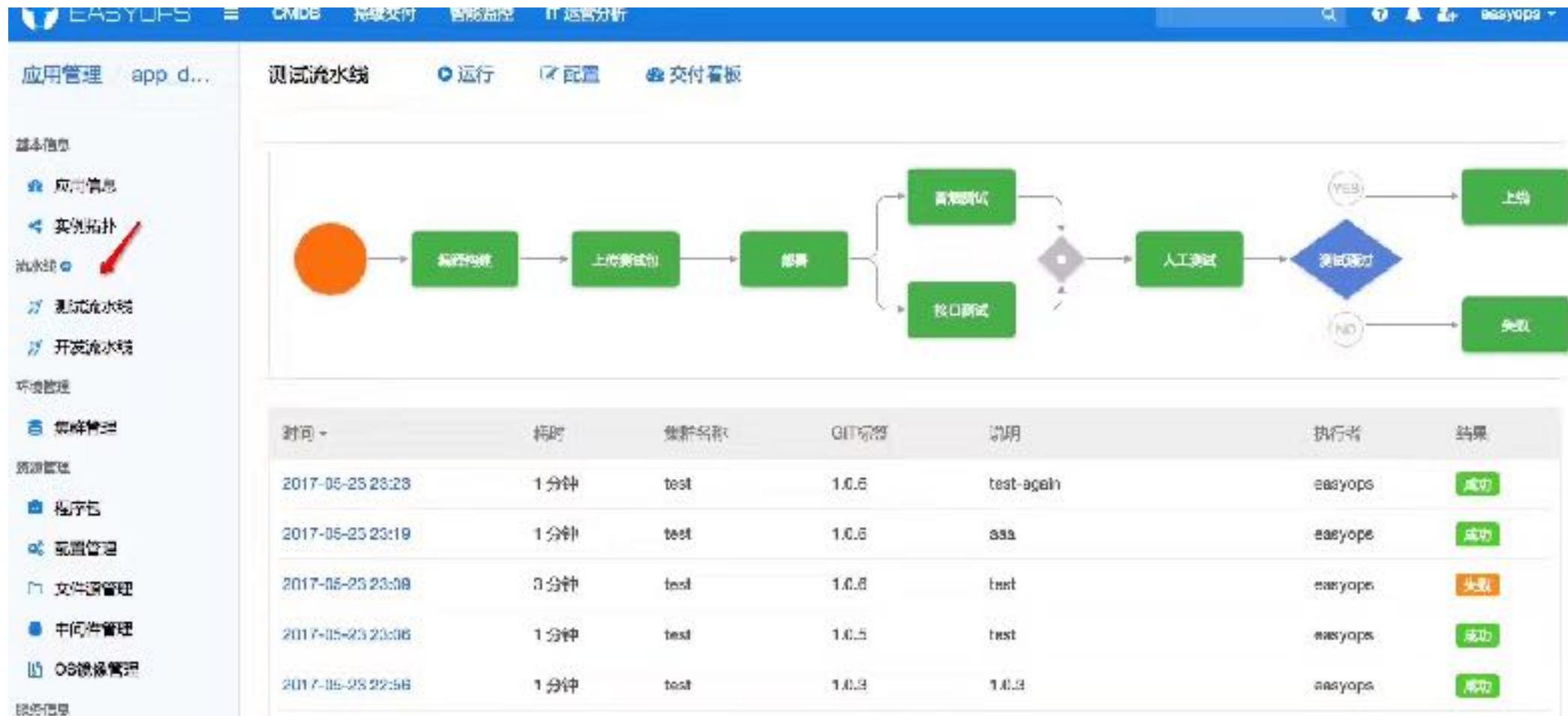
经验4： 关注高级使用

如果脚本的处理不是业务上原子的
则脚本不应该成为系统的组成部分

经验5： Just Work



经验5：Just Work



经验6：重在当下



经验7： 隐藏非关键信息

[← Back to show workflow list](#)
[↻ Refresh](#)

Workflow-Execution Informations			
Workflow Name : VendorFlexLaunchWorkflow		Workflow Version : 1.0	
WorkflowStatus : OPEN		Execution Id : FNL_P_VETO	
warehouseXmlS3Key :VETD.xml	launchCoordinator :Kumar Gandham		realm :EUFulfilment
vendorName :Amazon DE FOB Vendorflex	nodeType :VendorFlex		dataKey :VETO

Executed Tasks						
Task Name	Status	Action	Related Ticket	Time since started	SLA	Related Team
Start Registration Workflow	Done			14s	30m	EFP
Workflow Milestone: Start In-Configuration	Done			0s	30m	EFP
Setup warehouses with FOS	Done			7s	3m	EFP
Create Vendor and Warehouse Code	Done			0s	30m	EFP
Setup New Node for CIT	In_Progress	Mark as done	Ticket:0022530711	1h 29m 35s	720h	Trans-CIT

Unexecuted Tasks			
Task Name	Status	Action	Dependent Tasks
Setup Warehouses in GPI Service	Blocked		Setup New Node for CIT (In_Progress)
Populate User and Vendor Tables	Blocked		Setup Warehouses in GPI Service (Blocked)
Node Launch	Blocked		Setup Warehouses in GPI Service (Blocked)

出了问题才关注细节

经验8：业务为线，不要堆砌功能

您现在的位置：供应商管理平台>>子合同>>子合同列表

子合同（新） 子合同（旧）

子合同编号： 子合同状态： 供应商名称：

供应商编码： 签约模式： 档期编号：

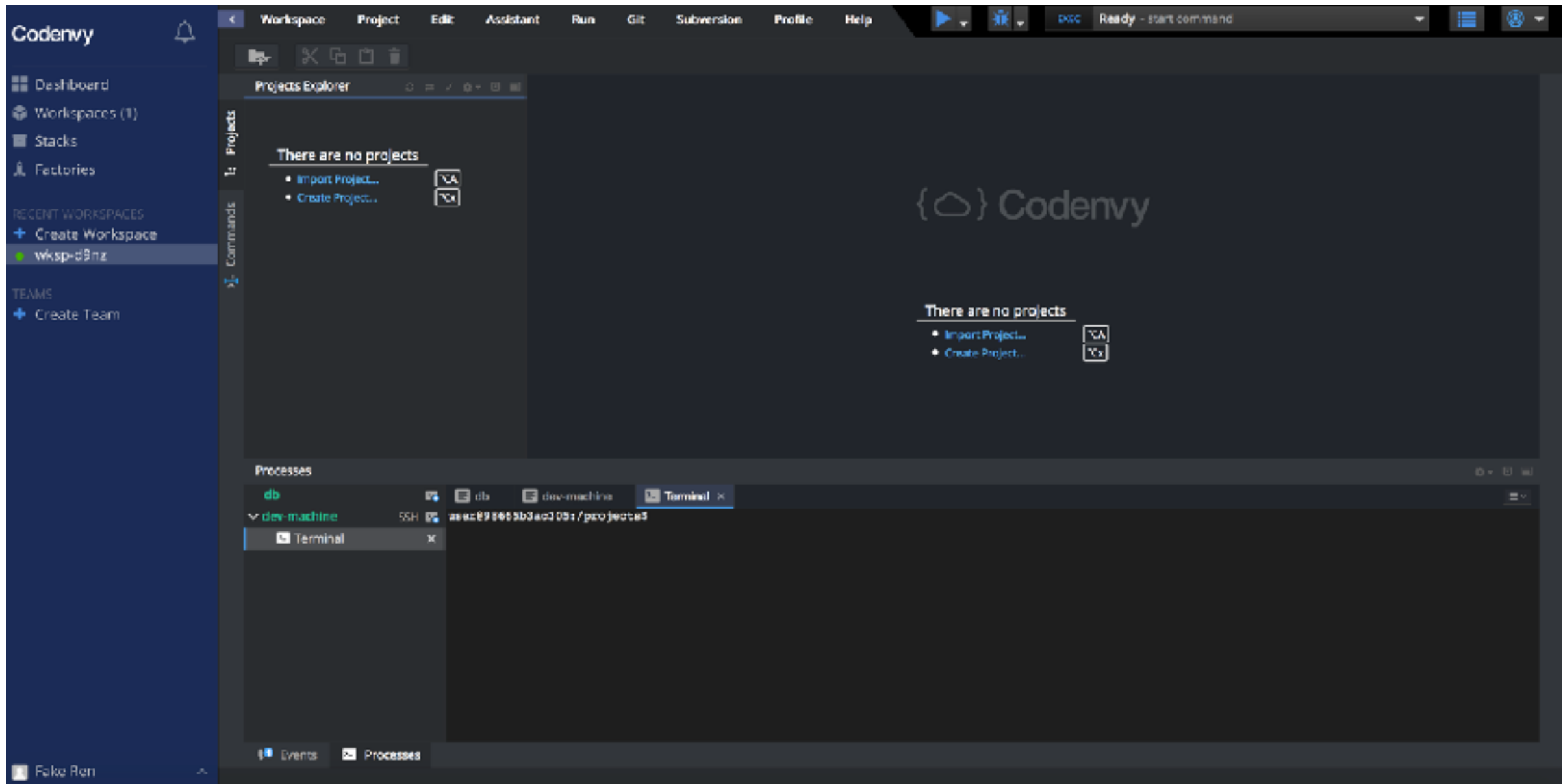
数据范围： 渠道标识： PO号：

操作	子合同编号	子合同状态	签约模式	供应商名称	供应商编码	PO	档期编号	档期名称	售卖区域	总
查看 提交	zht0000000056	待提交	电子印章	雅苑巴巴	889200	6100000291	10002932	味道十足	北京	98
查看	zht0000000055	已完成	普通模式	丹江口旅游公司	779280	2101006227	14585412	丹江口旅游专场	成都	348
查看	zht0000000054	待供应商确认	普通模式	阿狸图百货	880280	2101005902	10003211	大麦场专场	武汉	245
查看 修改	zht0000000052	经理驳回	电子印章	五二二供应商	103530	2101006132	14585916	快看剧玩专场	成都	2
查看	zht0000000051	待经理审批	电子印章	康拉德马莉供应	103530	2101005846	14585912	家电专场	广州	25
查看	zht0000000050	待合同专员确认	普通模式	欧欧安实业集团	103530	6100000276	10003290	欧欧安专场	上海	68
查看	zht0000000049	待总监审批	电子印章	卡尼尼东南公司	103530	6100000283	14585928	代客专场	武汉	34

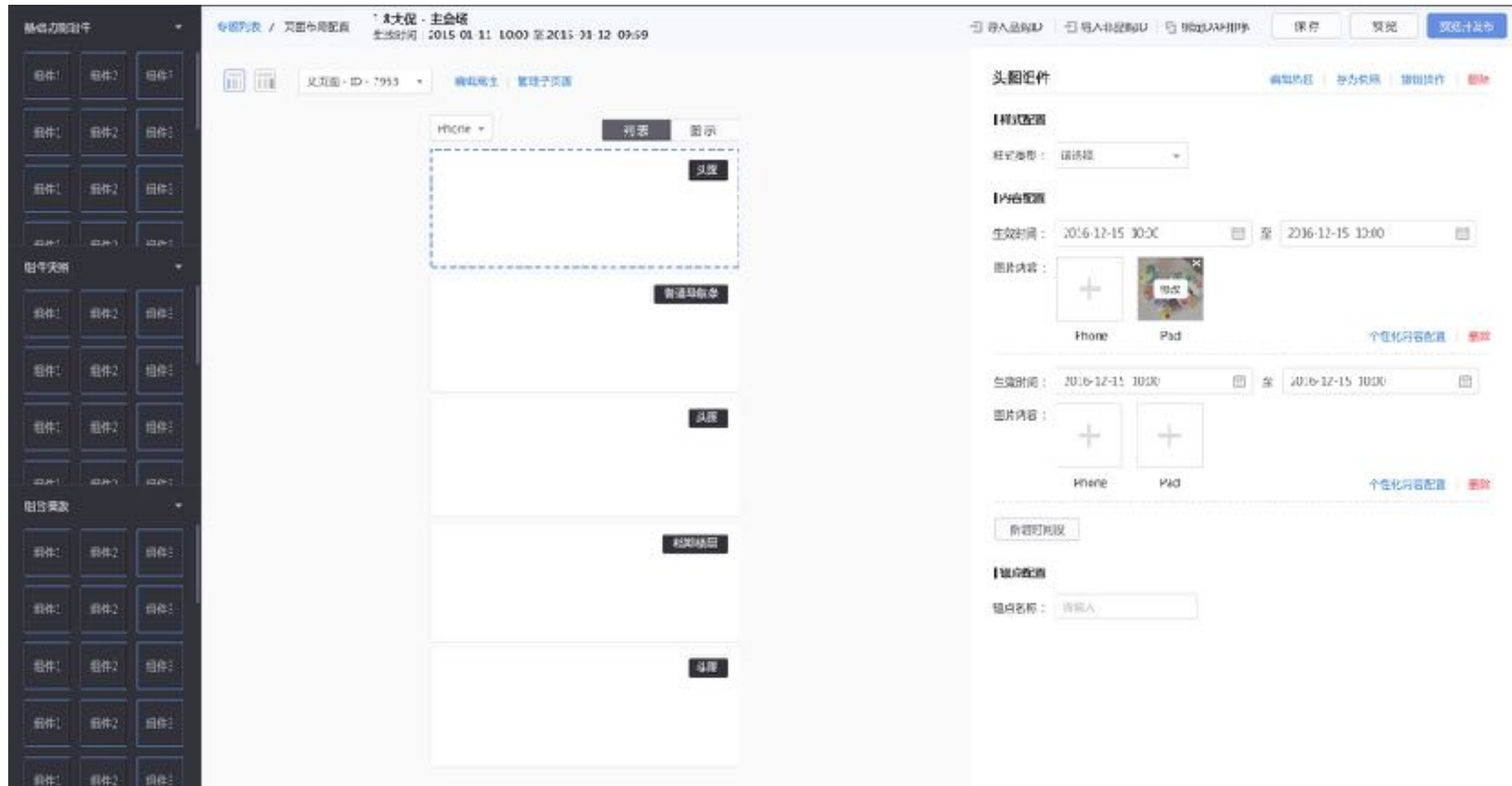
共7条数据 每页显示 10 条 1 [跳转](#)

注意：请从档期清单列表中创建子合同，再到子合同（新）下修改。

经验8：业务为线，不要堆砌功能

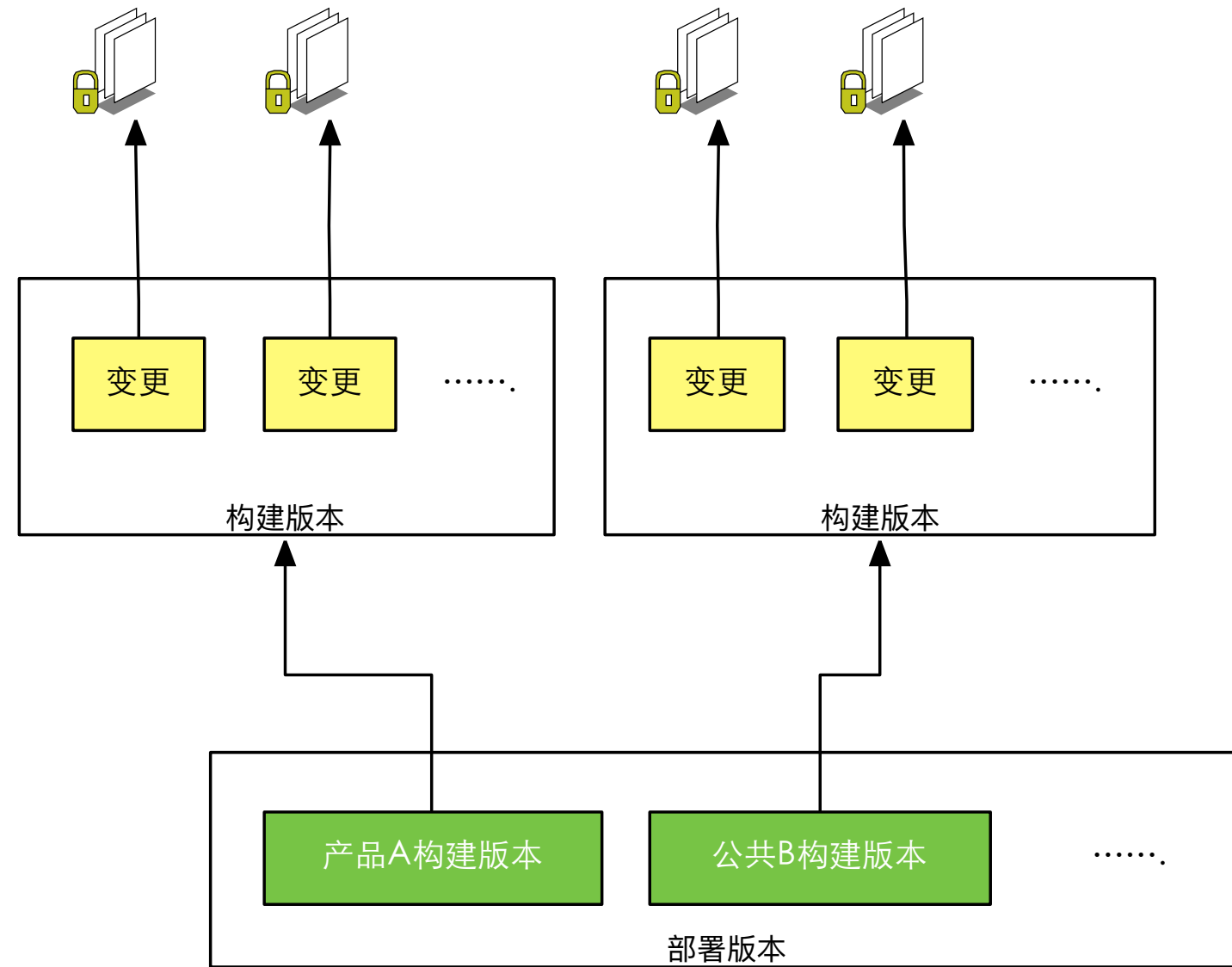


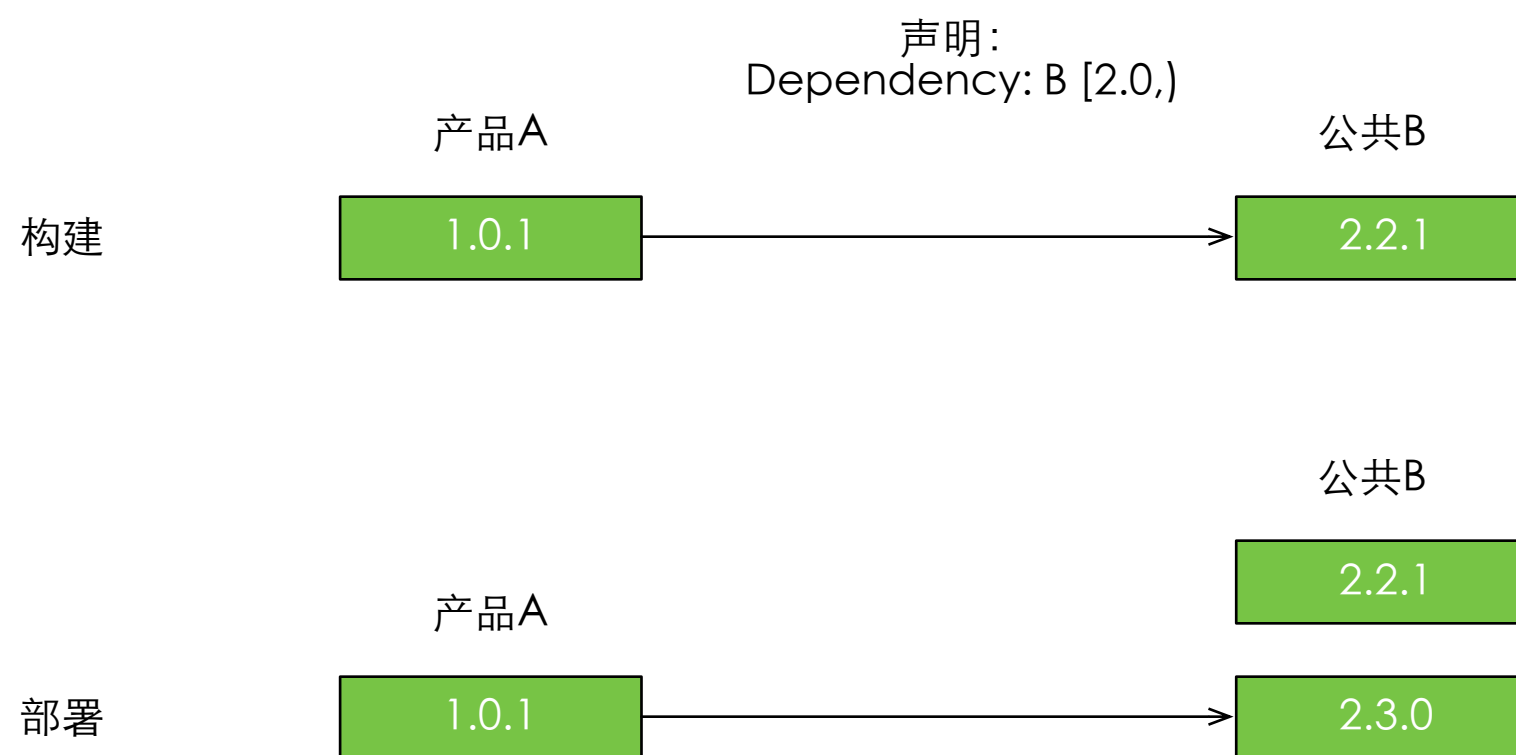
经验8：业务为线，不要堆砌功能



经验9：突出使用，分离配置

经验10：简化操作





深表谢意！