



万里目少儿

创建前端工程仓储应该考虑什么

关于创建前端工程仓储

- 需要考虑什么
- 我们是怎么做的
- 收益是什么

01

关于创建前端工程仓储
需要考虑什么

大家是怎么创建前端工程仓储的？

第一次创建仓储：直接基于 webpack 创建

- 诉求：后端要用
- 问题：
 - 前端工程新手门槛高
 - 需要考虑组件等必要的第三方库

复制类似业务仓储 + 删除业务代码

- 诉求：
 - 紧急新项目
 - 原来的一些方法需要复用
- 问题：
 - 多余代码一开始没删干净，后面就没人维护了
 - 不是每一次都能找到符合「类似」这个条件的业务仓储，配置该改还是得改

`create-react-app / vue-cli`

- 诉求：
 - 紧急新项目
 - 临时的小项目
- 问题：
 - 要长期维护了，该加该改的还是要搞
 - 需要考虑组件等必要的第三方库
 - 和公司内部一些比较个性化的方案不好结合

复制仓储模板 + 修改

- 诉求：
 - 快速创建
 - 以往的解决方案、个性化配置尽量包含其中
- 问题：
 - 调整配置还是得手动到对应的地方改，还是有上手成本
 - 多个模板轮子，无统一，谁都说自己的好，但都没有绝对的完整

需要解决这些问题

成本问题：

- 新仓储创建到上线的成本：找代码改配置、和公司内部较个性化的方案不好结合、紧急
- 开发上手成本：可复用的第三方库与解决方案

创建仓储一事面向的角色：

- 资深前端
- 新手前端
- 非前端

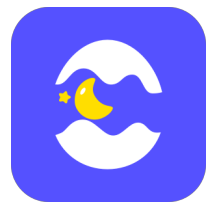
业务情况：

- 业务体量
- 临时/持续维护

其他：

- 新建仓储的干净程度
- 模板难以统一

需要解决这些问题



Pangu 应运而生!

02

关于创建前端工程仓储

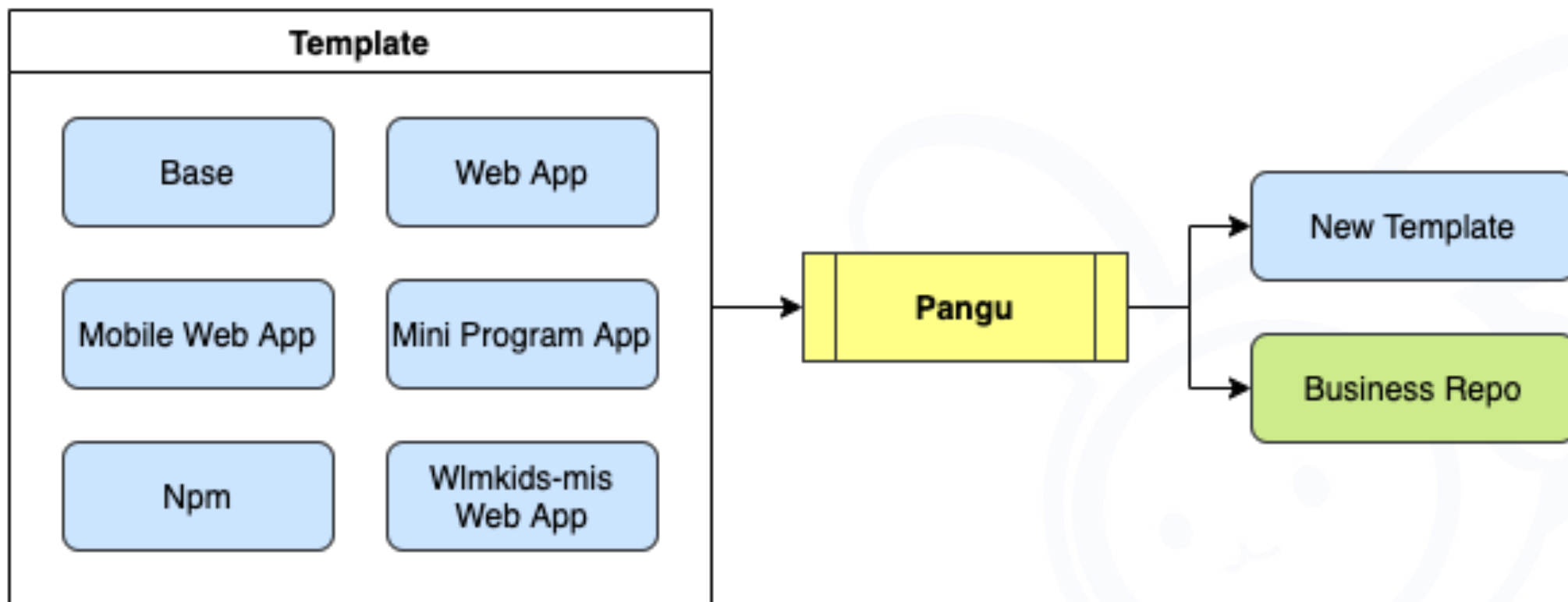
我们是怎么做的

| Pangu 是什么?

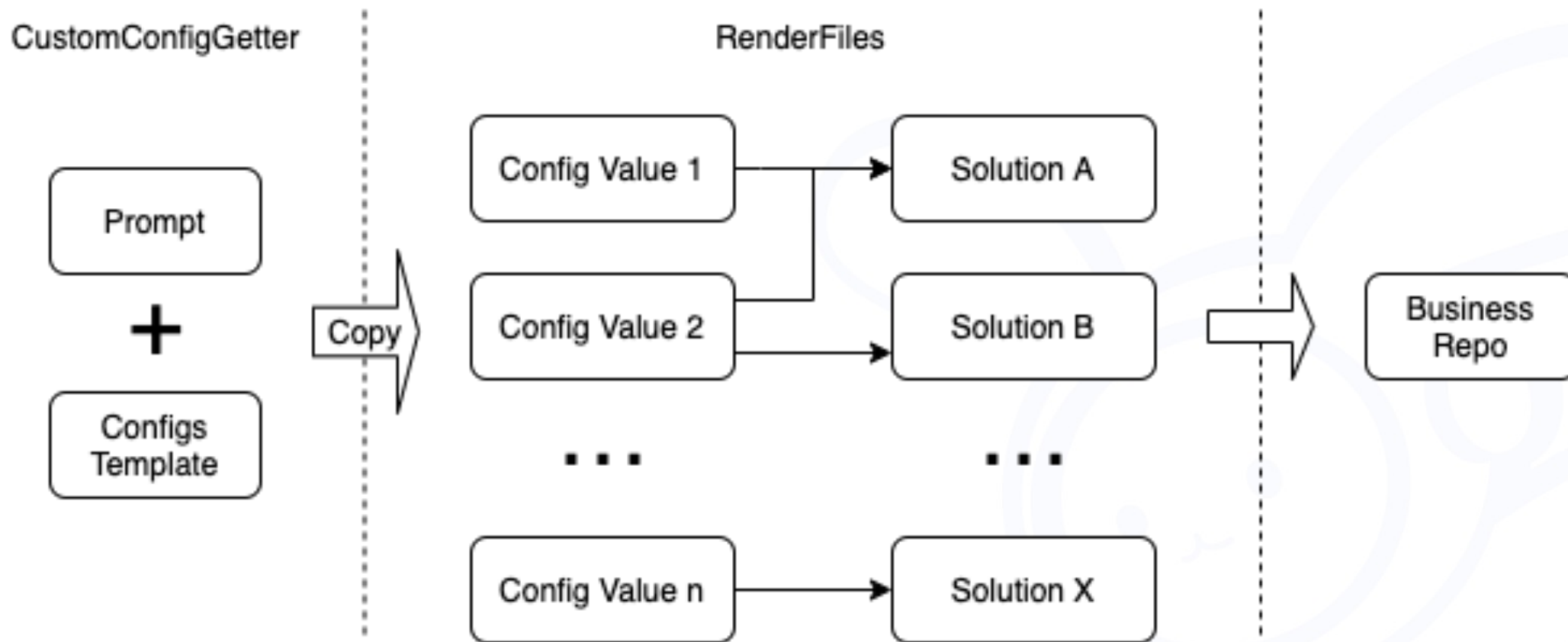


一个配合模板后可**快速**、**简单**地用命令**创建**、**更新**前端工程仓储的工具。

基本模式



创建仓储 – 配置与解决方案



I 创建仓储 – Replace

```
{
  "prompt": {
    "type": "text",
    "name": "someValue",
    "message": "请输入 xxx",
    "initial": "{{{ projectName }}}",
  },
  "globPatterns": [
    "FileName"
  ],
  "replacePattern": "TemplateFileName",
  "replaceInTemplate": true
}
```

| 创建仓储- Mustache Render

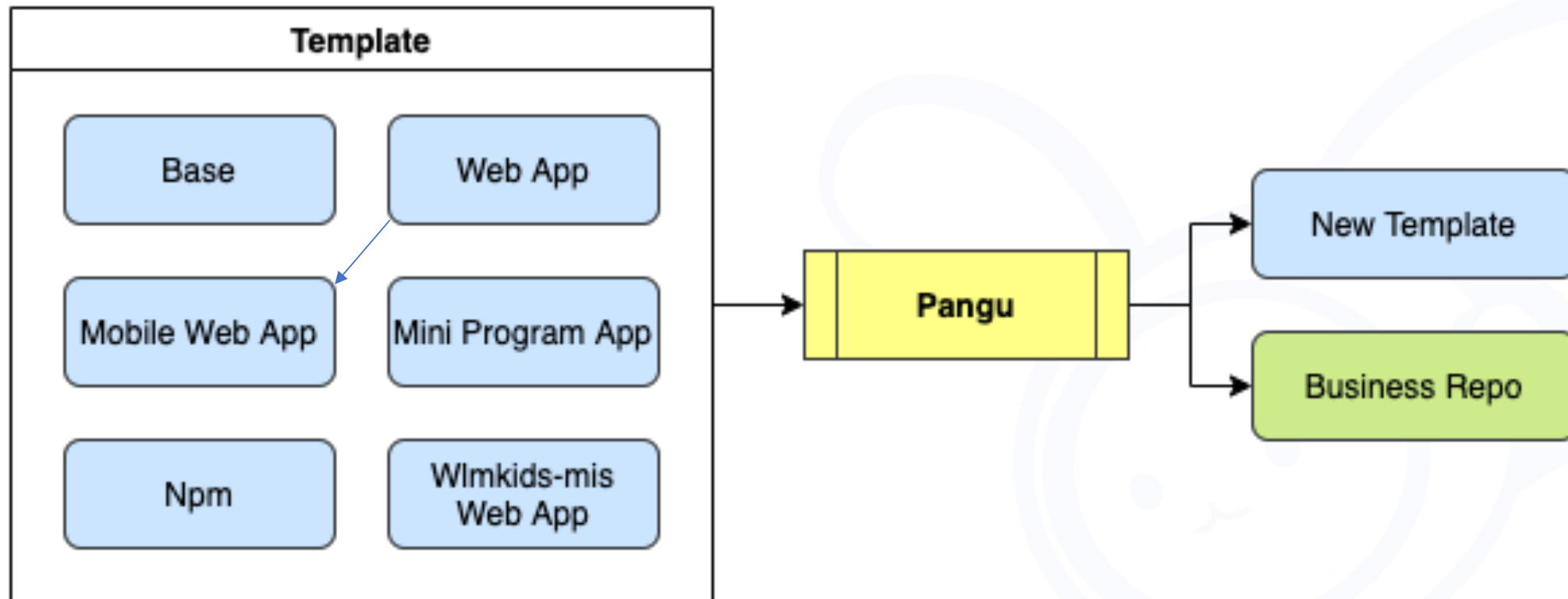


```
{
  "prompt": {
    "type": "text",
    "name": "productionBranch",
    "message": "上线分支 (如果不打开 auto tag, 请忽略, default: \"master\") ",
    "initial": "master"
  },
  "globPatterns": "auto-tag.config.json"
}
```

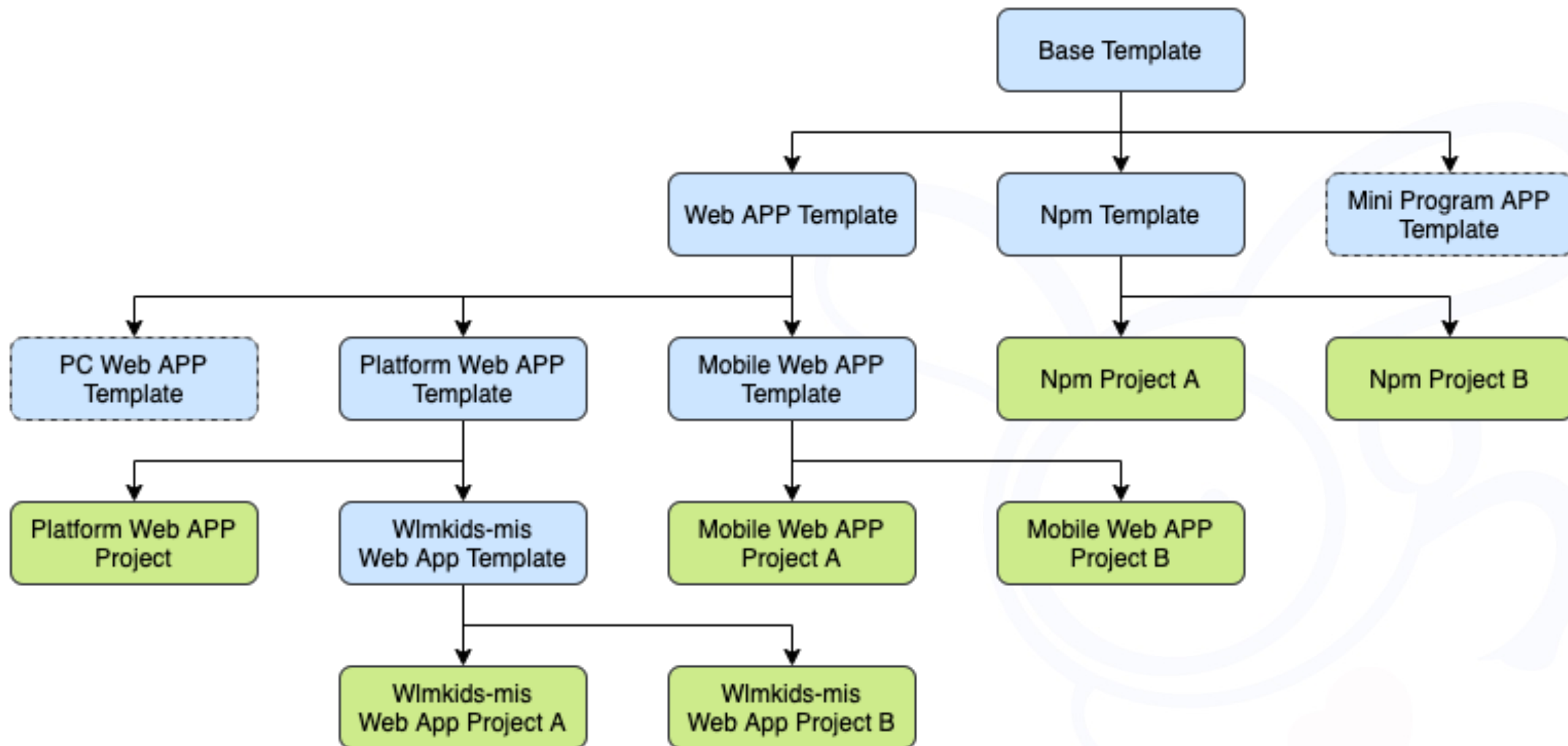

| 创建仓储- Mustache Render

```
{  
  "enable": "enableAutoTag",  
  "dingtalkTokens": "{{{ dingtalkTokens }}}",  
  "branches": "{{{ productionBranch }}}",  
  "tagFormat": "{{{ tagFormat }}}"  
}
```

更新仓储 – 再看基本模式

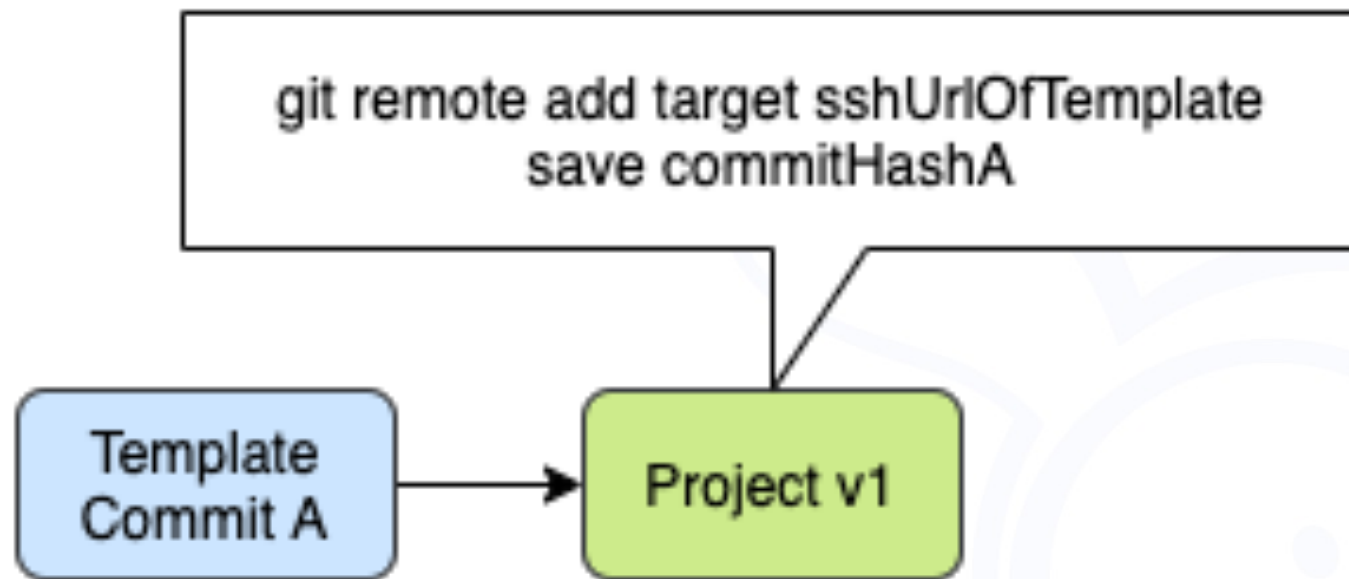


更新仓储 – 继承关系

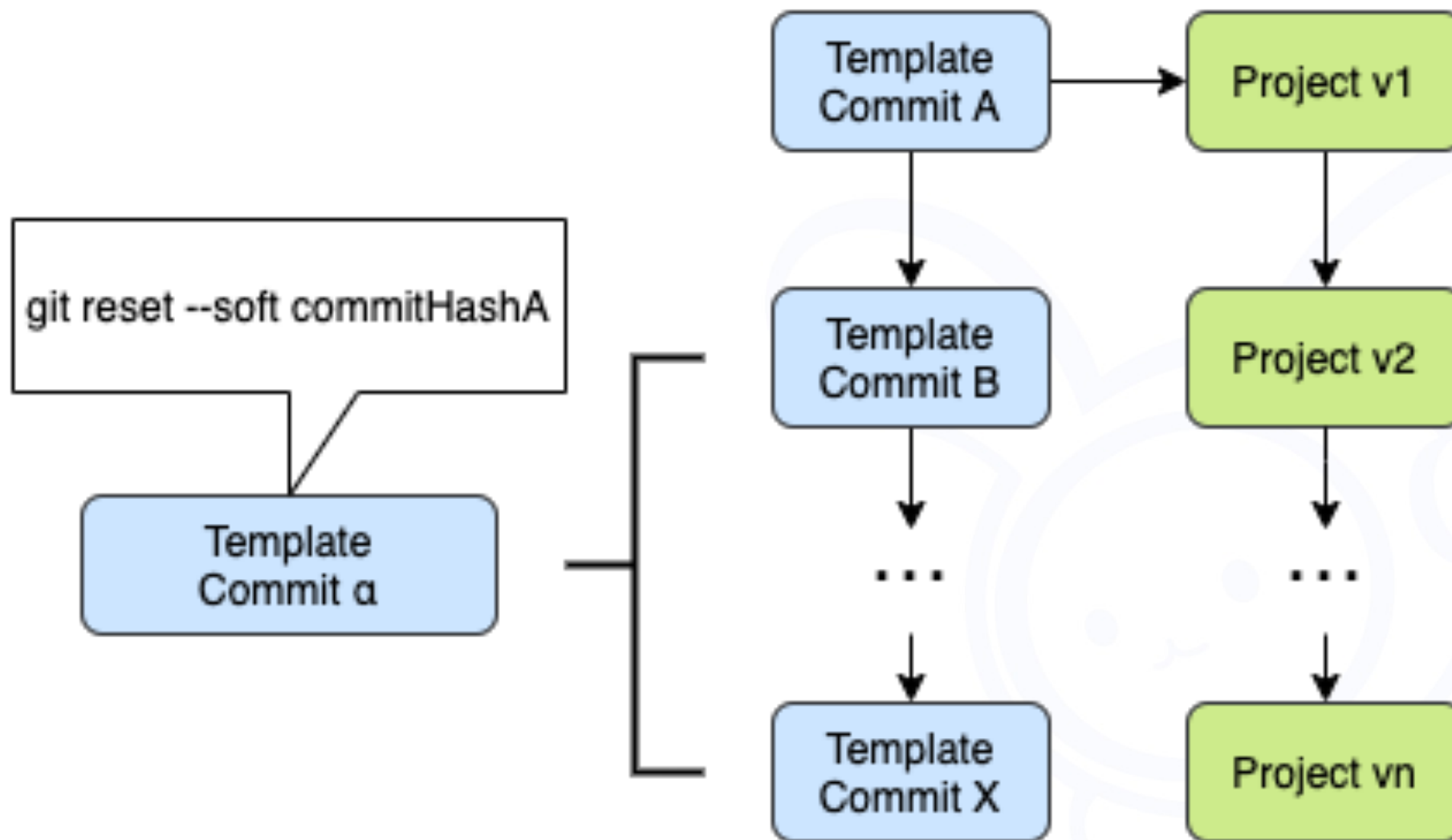


- 减少创建模板的成本
- 模板越精细，创建时需要关注的配置越少
- 基于继承关心，可能可以将模板的迭代更新到它所派生的仓储中

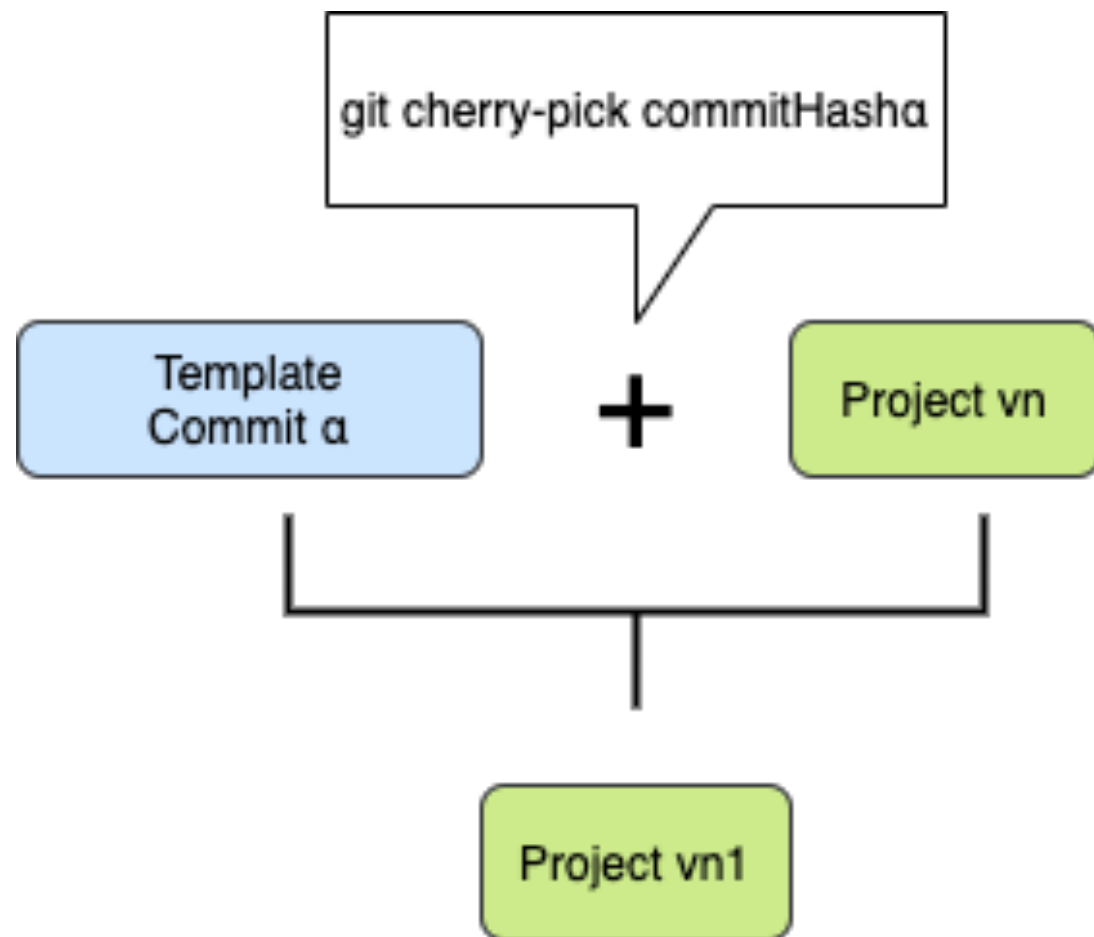
更新仓储 – cherry-pick



更新仓储 – cherry-pick

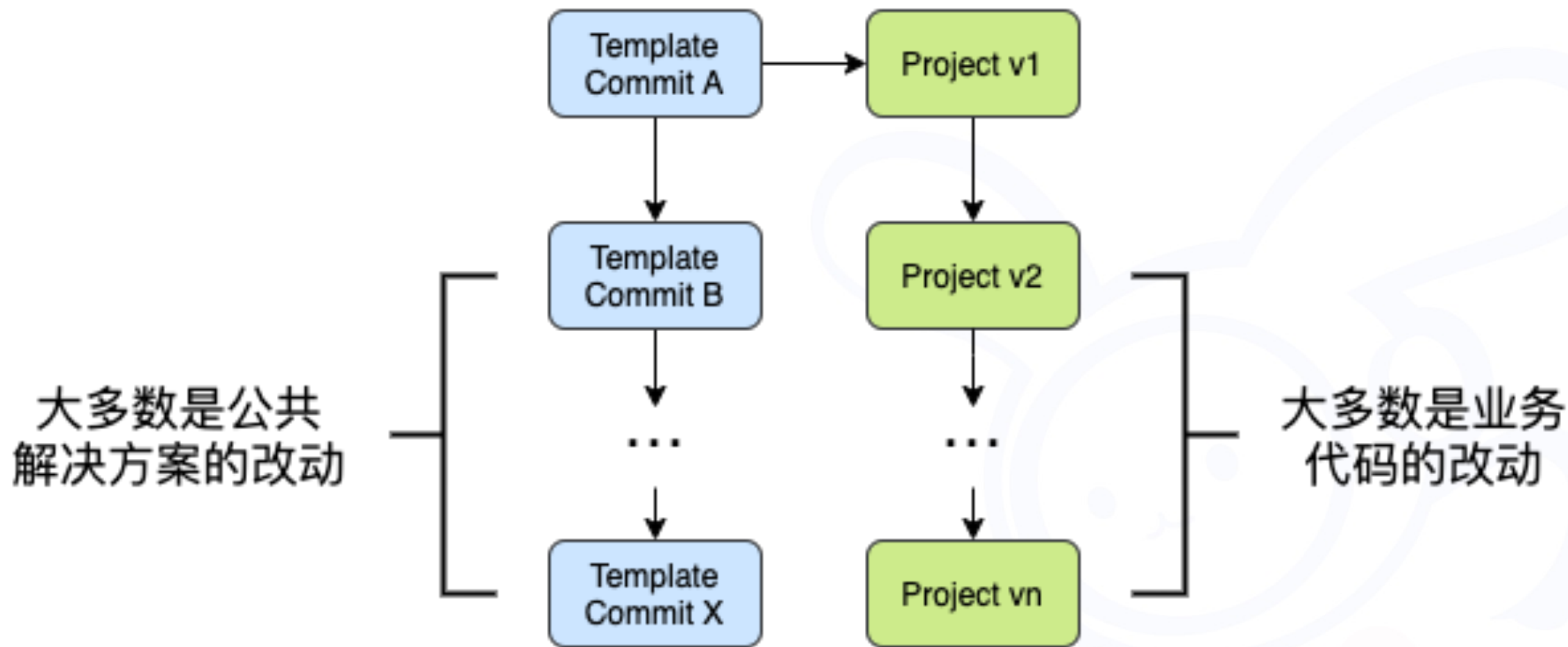


更新仓储 – cherry-pick

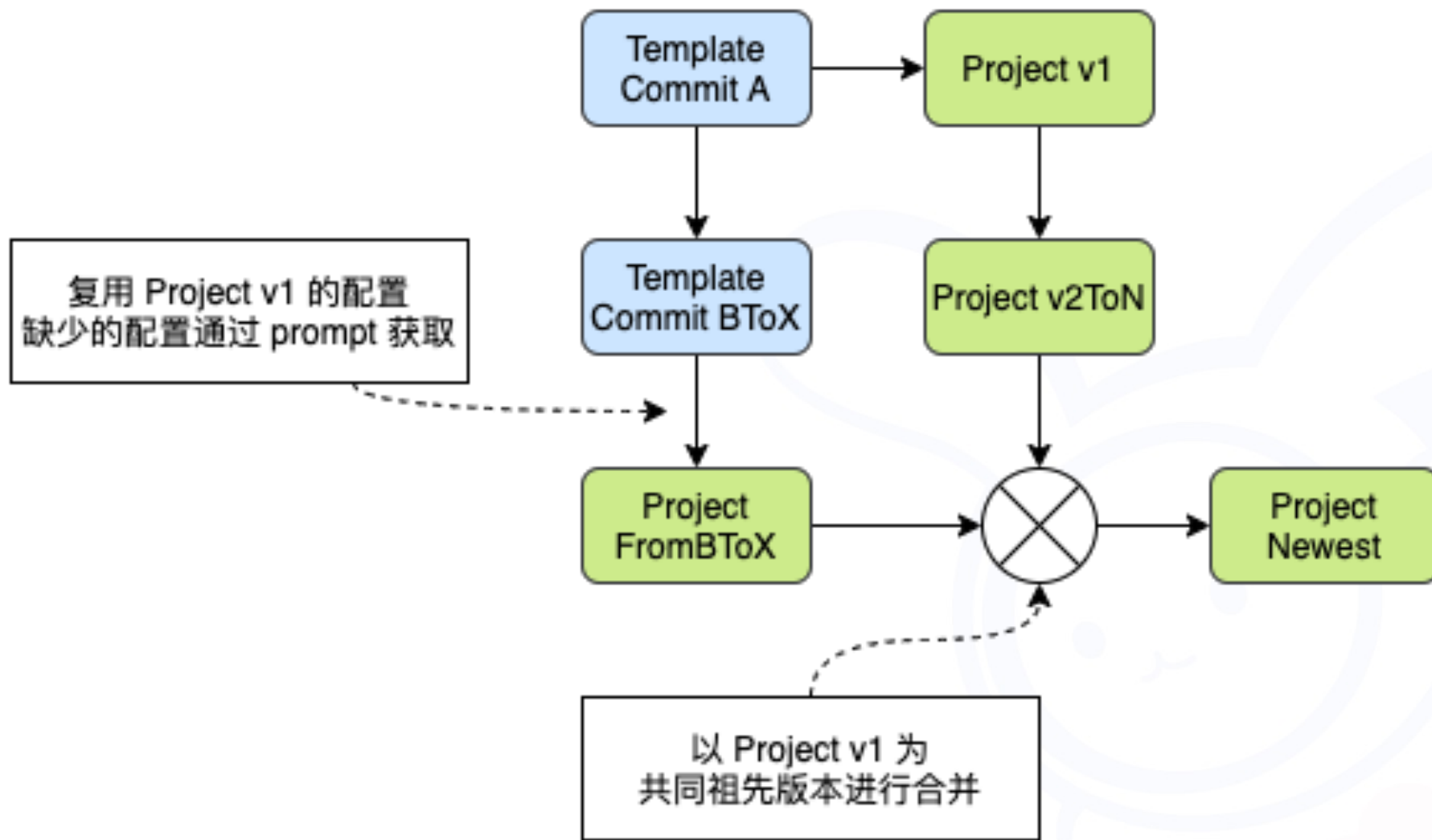


问题：配置发生变更（增删改）时，只 pick 过来未经 render & replace 的模板片段，需要修补的地方很多。

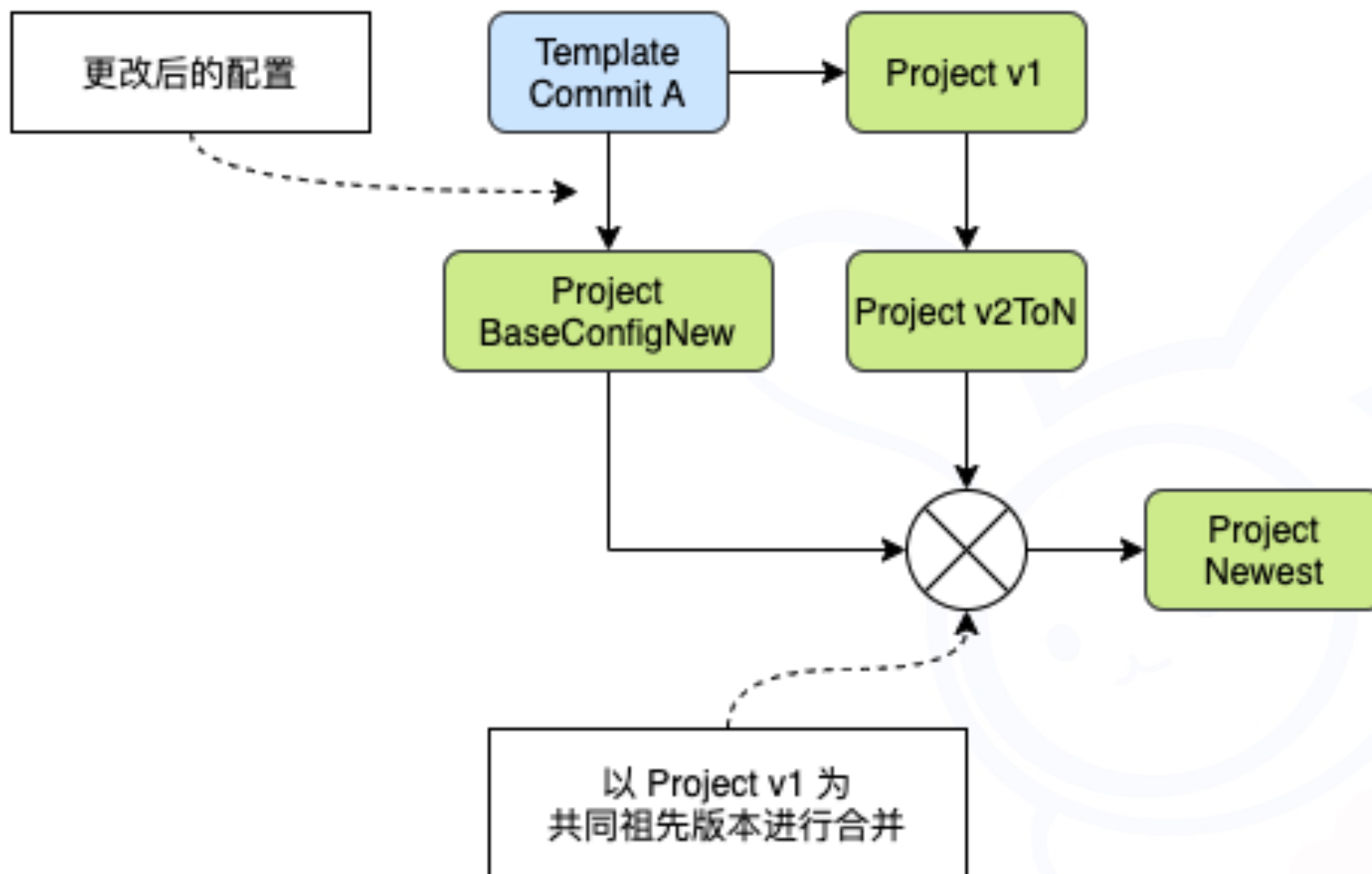
| 更新仓储 – merge



更新仓储 – merge



更新仓储 – merge



03

关于创建前端工程仓储

我们做完后收益是什么

| Pangu 做到了什么？



配合模板后可**快速**、**简单**地用命令**创建**、**更新**前端工程仓储。

再看需要解决的问题

成本问题：

- 新仓储创建到上线的成本：找代码改配置、和公司内部较个性化的方案不好结合、紧急
- 开发上手成本：可复用的第三方库与解决方案

创建仓储一事面向的角色：

- 资深前端
- 新手前端
- 非前端

业务情况：

- 业务体量
- 临时/持续维护

其他：

- 新建仓储的干净程度
- 模板难以统一

技术产出的使用率保证

- 绩效
- 技术成就感
- 技术产出可以经受考验，不断完善

技术栈统一

- 降低跨业务开发成本
- 快速解决其他非前端团队遇到的前端问题
- 缓解前端人力瓶颈
- 便于推广

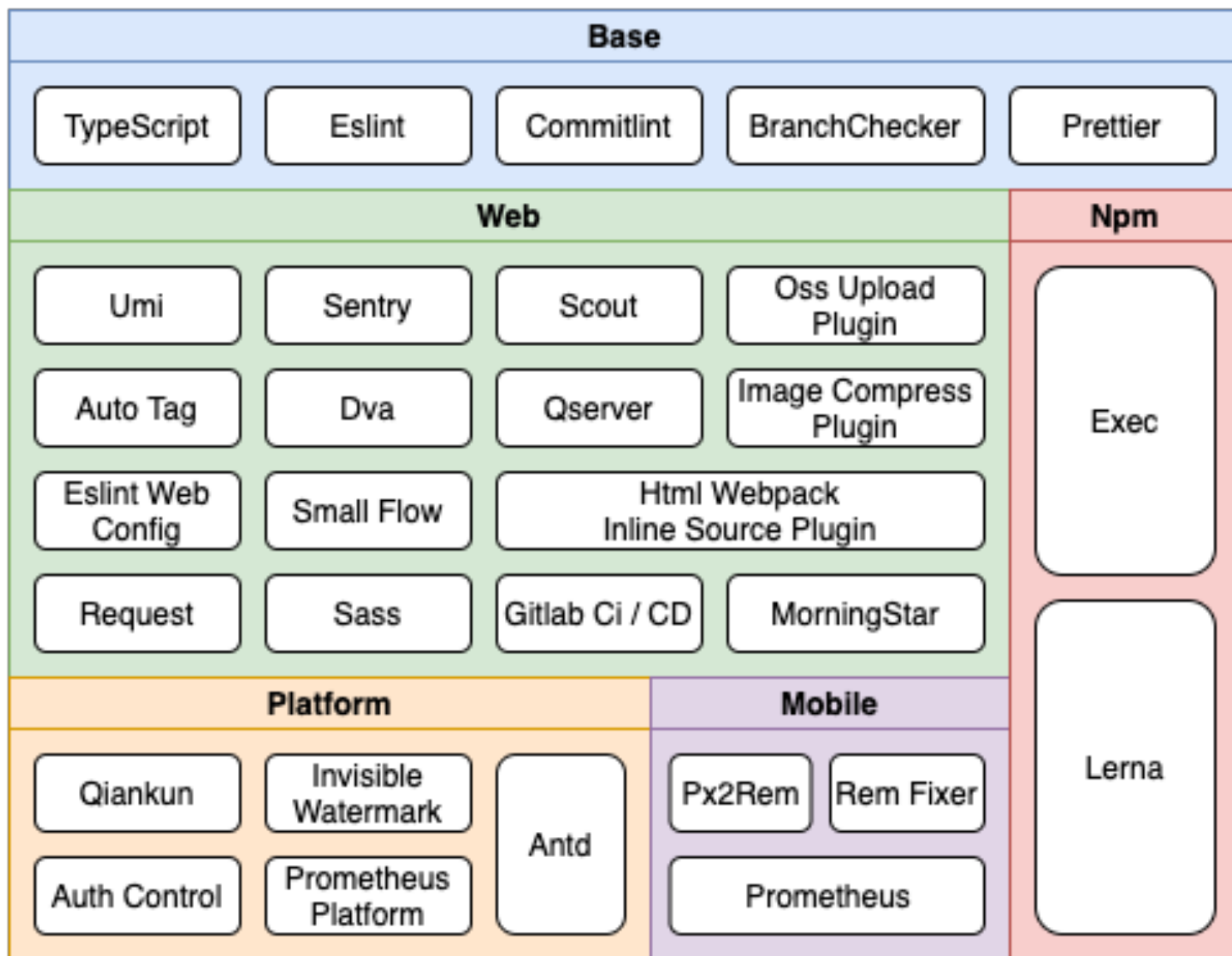
人力效率最大化

- 提倡优化、重构，不提倡造轮子
- 集中到一起，不会因为不知道而造轮子
- 每个人可以精通不同方向

| Pangu 是什么？



可以保证**技术产出使用率**、**保证技术栈统一**、**提高产出和精通人效**的**体系**



- 3 个团队

- 20+ 仓储

- 更多的模板：小程序项目模板、PC web 项目模板。。。
- 更多的解决方案/能力：SSR、性能优化、性能统计。。。
- 已有模板本身优化：插件化规范
- 工具能力：完善根据模板进行更新的能力

The background is a solid blue color. It features several thin white lines that curve across the frame. There are also four white circles of different sizes: one in the upper left, one in the upper center, one in the upper right, and one in the lower left.

THANKS

Q & A



S

中国台湾 云林县



扫一扫上面的二维码图案，加我微信