

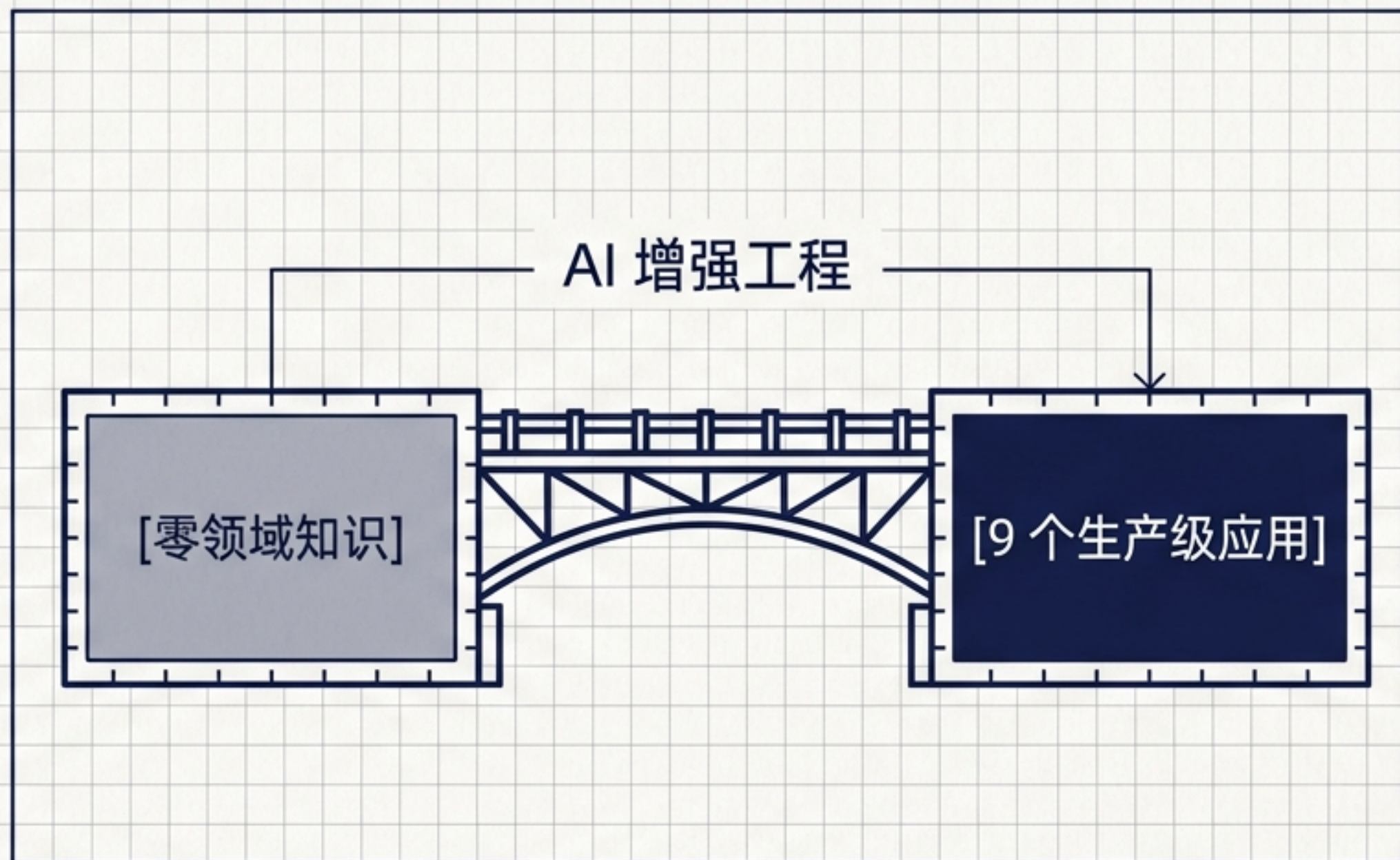
盘盘猫的最后 48 小时：基础设施冲刺记录

- > EXECUTE SPRINT_RETROSPECTIVE
- > TIME_ELAPSED: 29 DAYS
- > COMMITS: 1,134
- > PRS_MERGED: 109
- > PRODUCT_LINES: 9

一个工程师与AI的零知识领域实战日记

核心假设：AI 能否填补领域认知的空白？

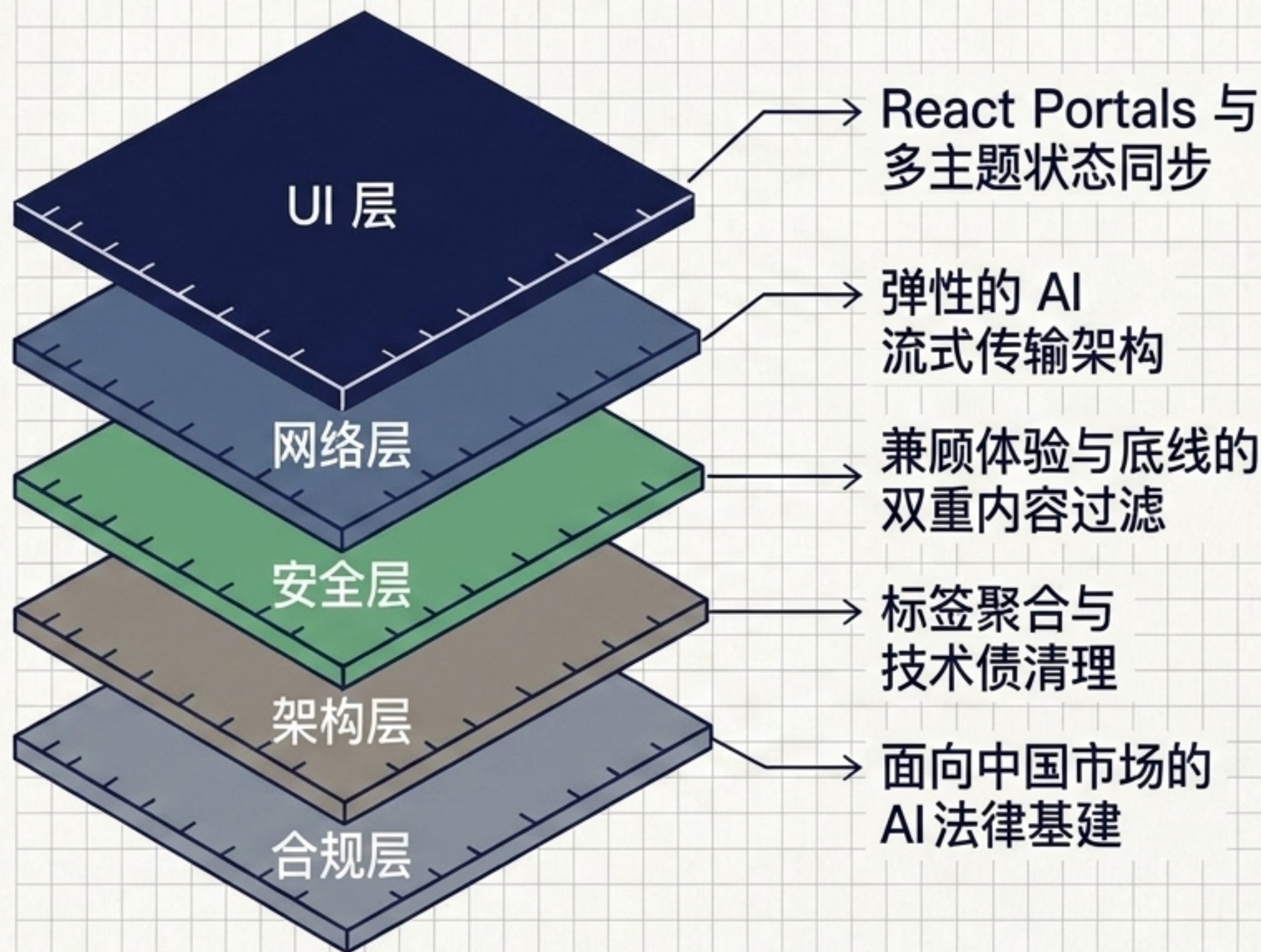
这个项目从立项开始，重点就不在“算命”本身。它是为了验证一个个现代工程假设：一个工程师带着 AI，能否在一个完全陌生的业务领域，从零搭建出一个完整的多产品平台？



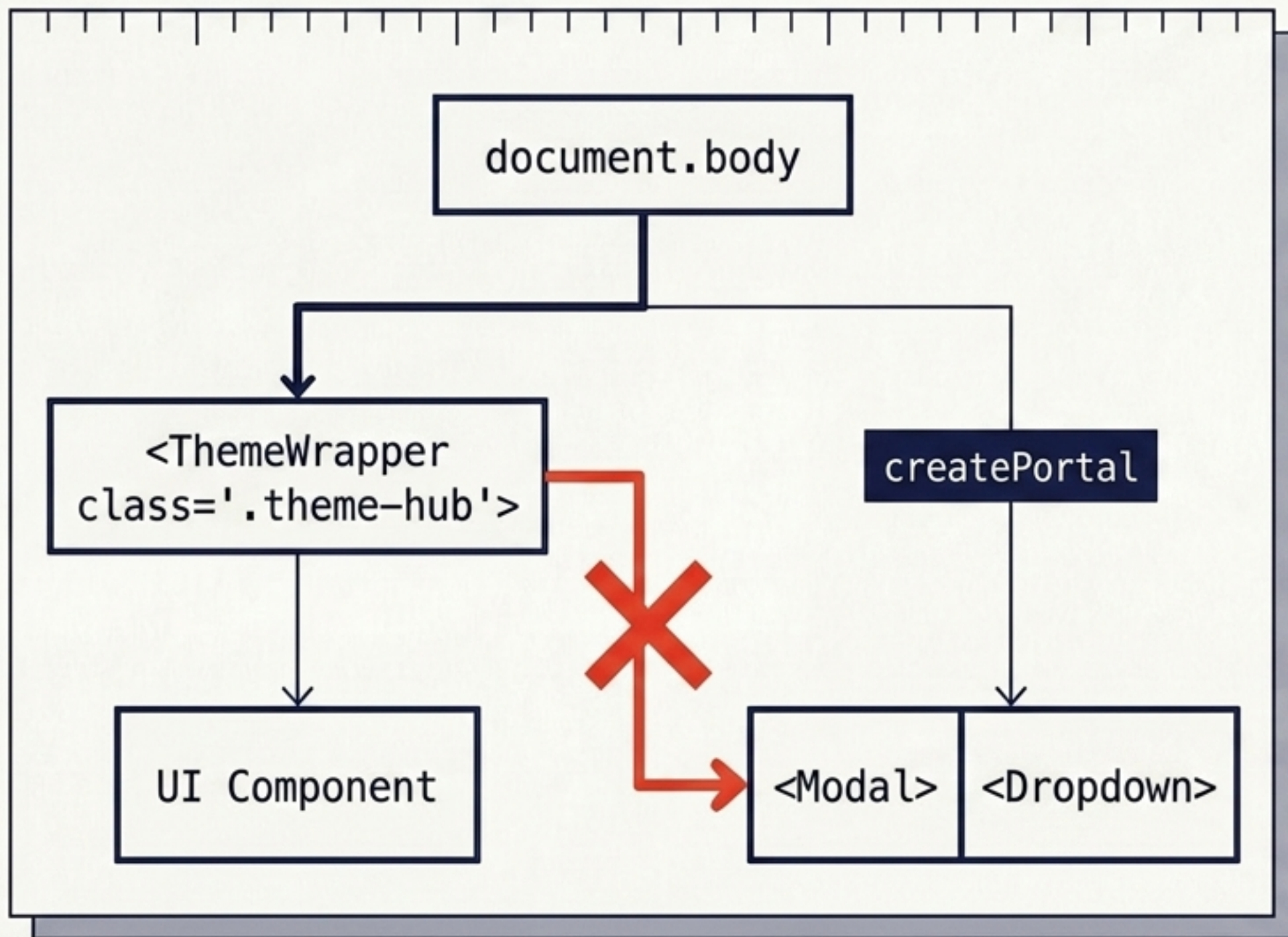
答案在第 29 天揭晓。但这最后两天的“最后一公里”基础设施建设，才是决定产品能否上线的生死线。

冲刺路线：自顶向下的系统重构

最后两天集中解决的 5 个关键技术阻碍。

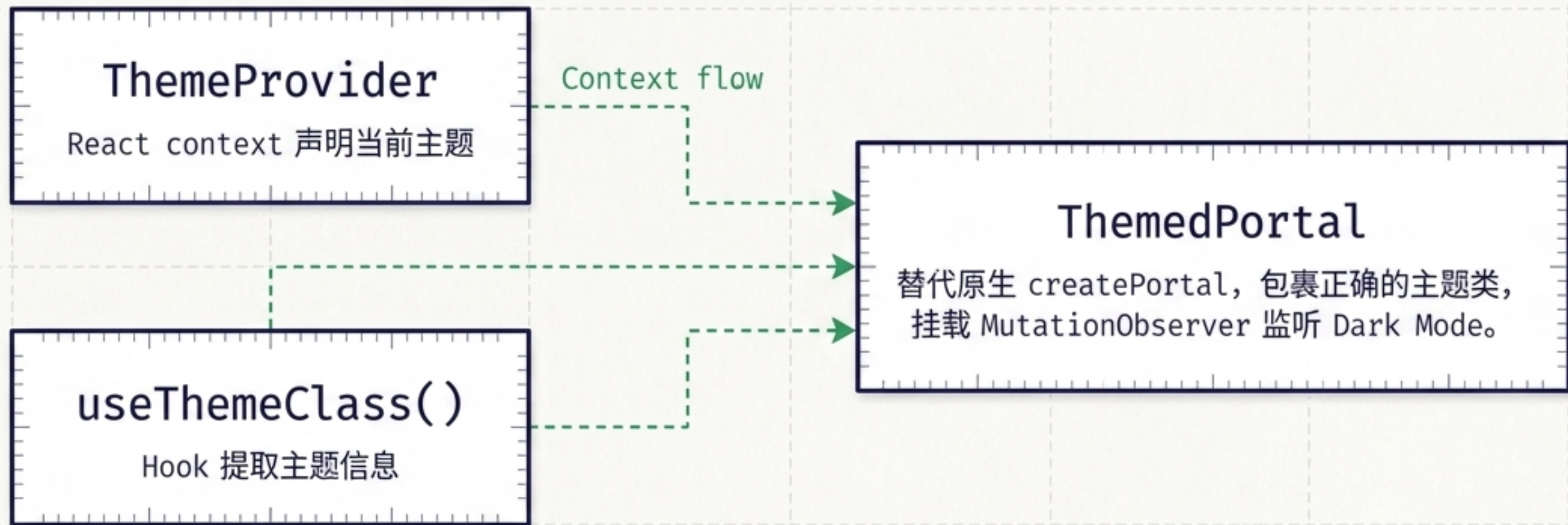


逃逸的 DOM 节点与丢失的 CSS 变量



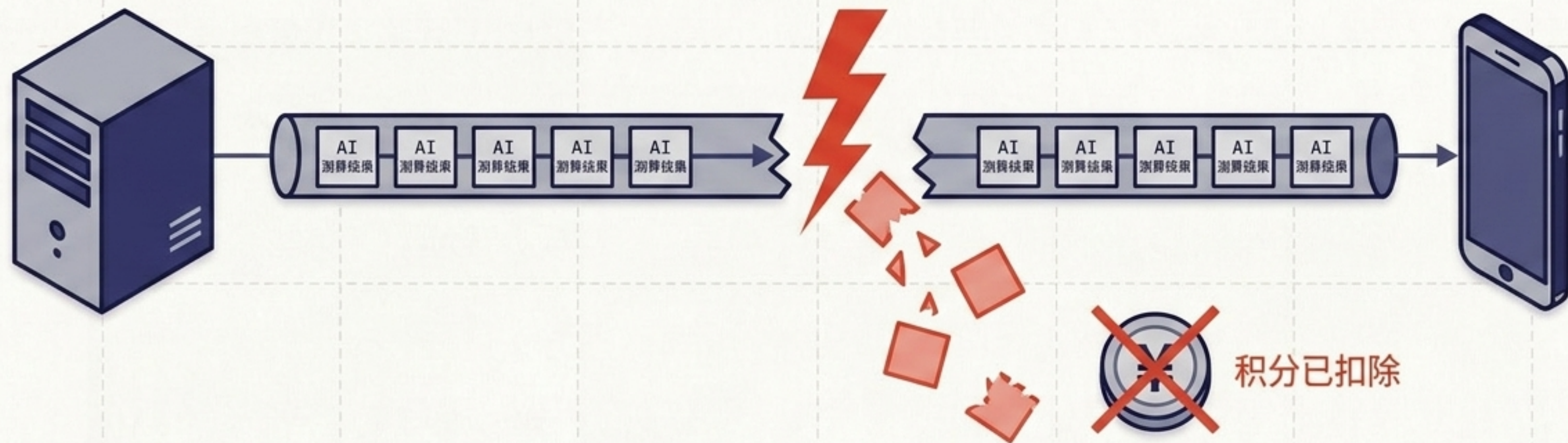
- 症状：9 套视觉主题。一旦使用 portal, Modal 与 Dropdown 完全丢失主题颜色。
- 根本原因：createPortal 将节点直接渲染到 `document.body`, 逃逸了应用主题 wrapper。CSS 变量继承层级脱节。
- 失败的初版 (PR #64) : 13 个组件逐个硬编码 `closest('.theme-hub')` 嗅探, 无法扩展。

优雅降级：构建 ThemedPortal (PR #67)



9 个应用布局全部更新，13 个 portal 组件成功迁移。所有逐组件的硬编码 hack 被彻底删除。

昂贵的断线：流式传输中的状态丢失



Core Conflict

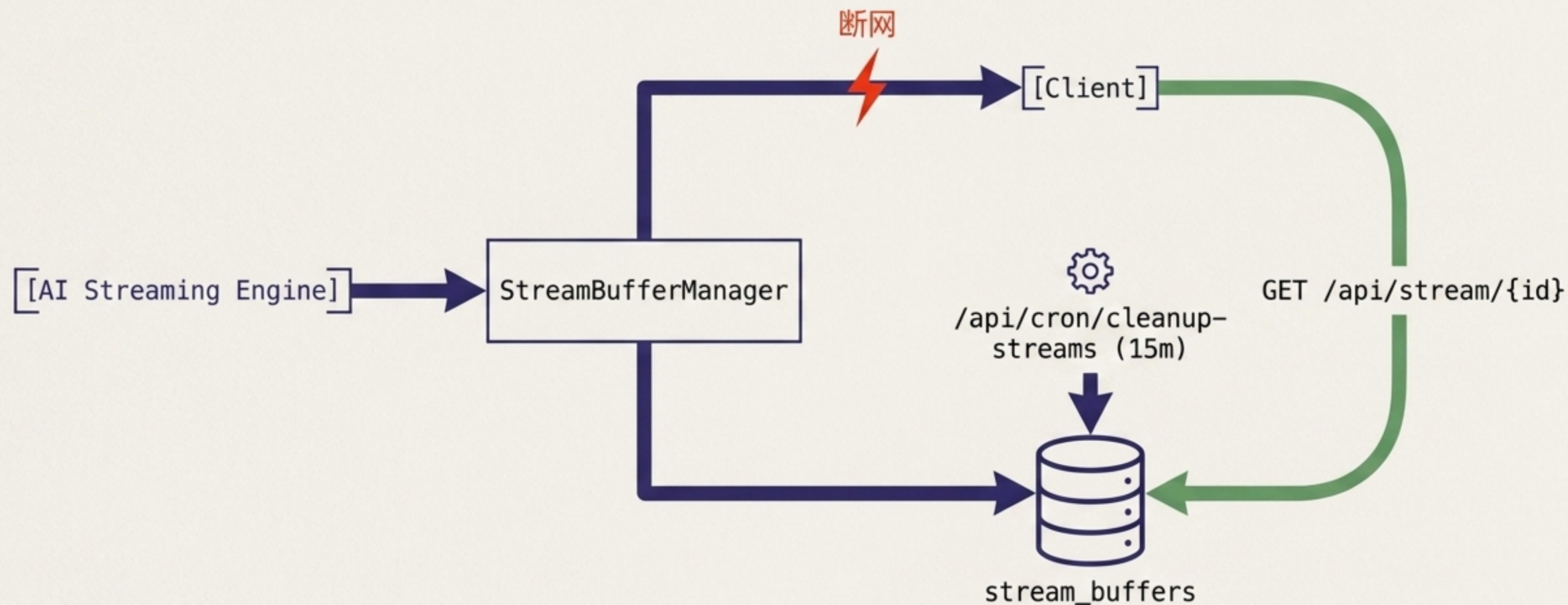
手机端弱网环境下，流式传输极易中断。连接断开导致用户花了积分，却拿不到完整的测算结果。

Engineering Mandate

连接断了没关系，但结果不能跟着一起没了。

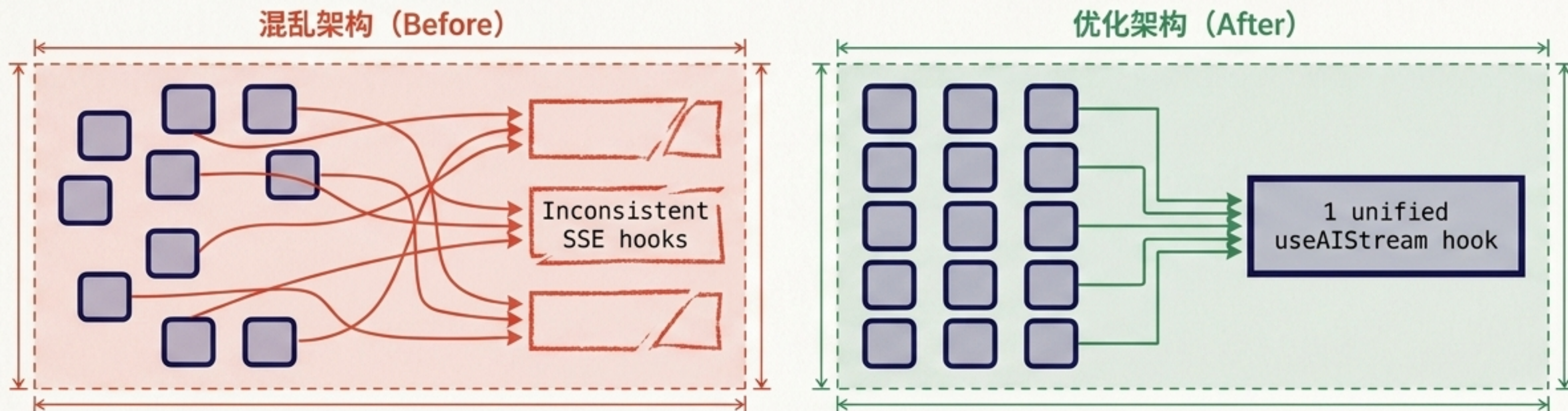
弹性基建：流式缓冲与状态恢复 (PR #105)

改造覆盖 19 条 AI streaming 路由，触及 56 个文件。



架构的悖论：更多功能，更少代码

+ 弹性缓冲基建 = -186 行代码



当添加复杂的系统级功能（缓冲、恢复、Cron清理）时，代码总量反而净减 186 行。

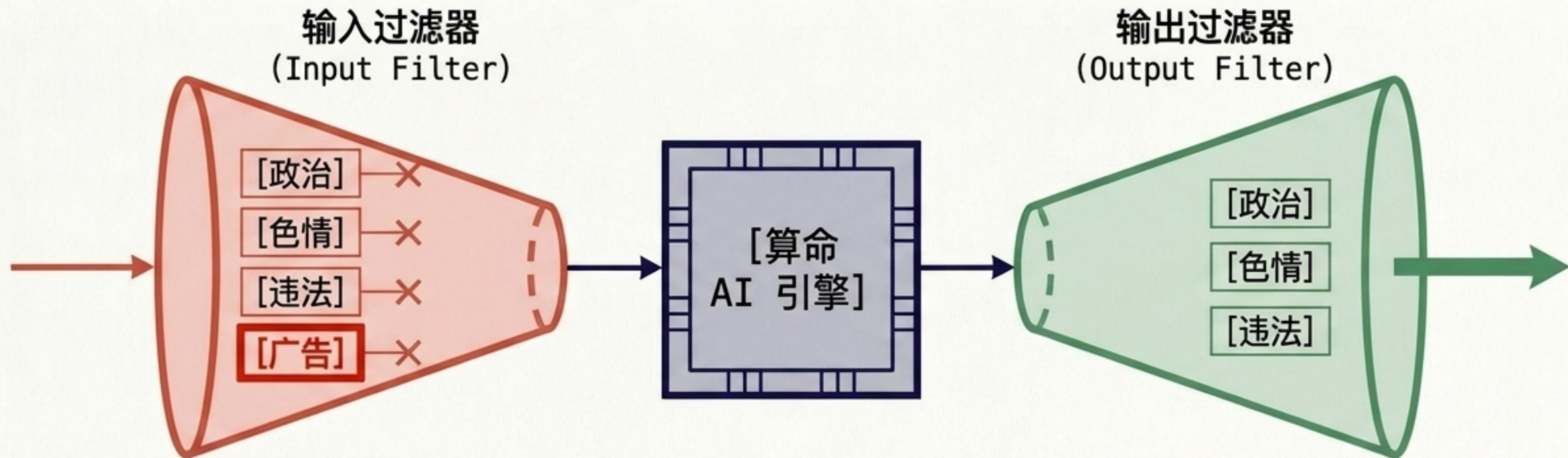
架构方向的正确性，往往体现在复杂度的收敛上。

内容过滤器的残酷进化

版本	机制	处罚结果	用户体验
V1	DFA 关键词过滤	匹配即拦截，整条封锁	极差（简单粗暴）
V2（PR #78 ）	流感知过滤，遇敏感词截断	流中断，积分照扣	差（拿到半截结果）
V3（PR #104 ）	行内遮罩双重过滤	替换为 检测到违禁词	优（流不中断，平滑降级）

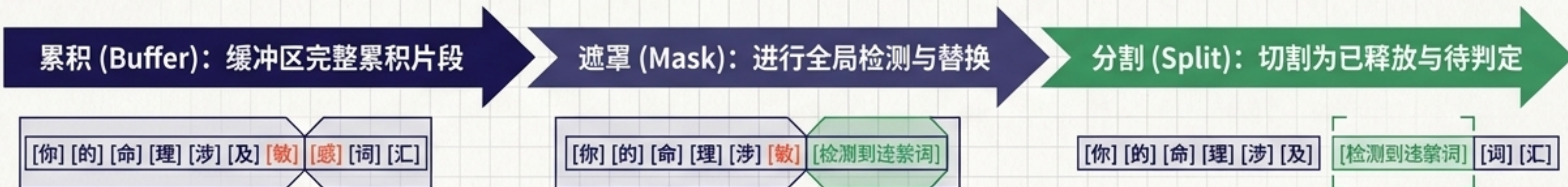
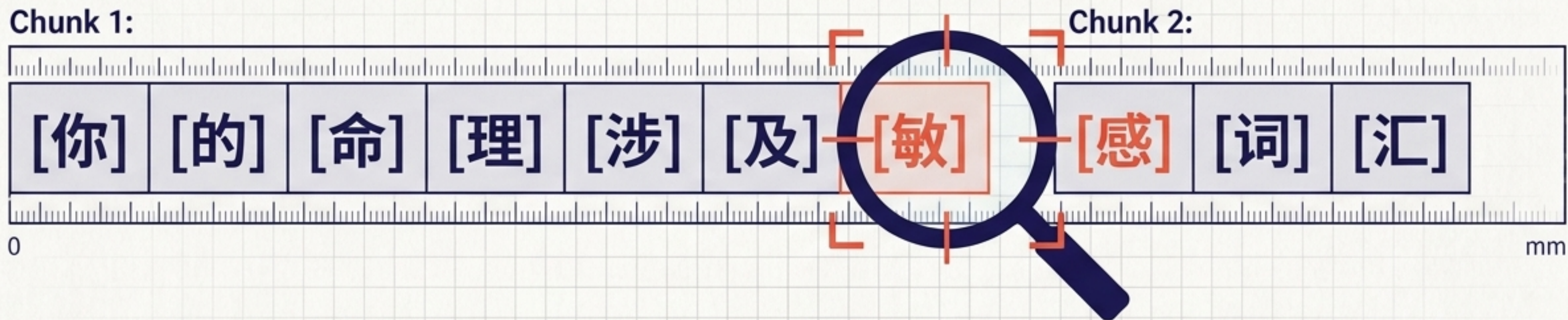
每一版架构重构，都是被用户在生产环境的真实痛点逼出来的。

不对称双重过滤 (V3)



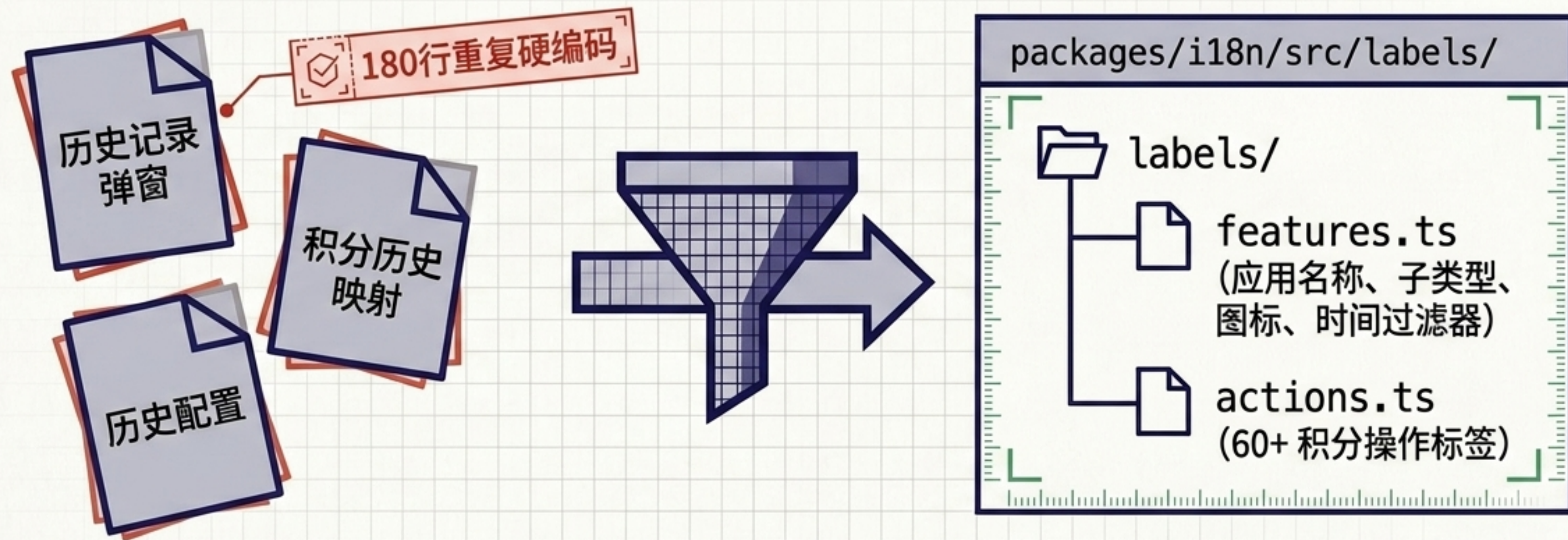
- 输入端严格，输出端宽松。
- 为什么去掉输出端的“广告”拦截？算命 AI 在正常推演中，极易自发生成“兼职”、“招聘”、“桑拿”等词汇。如果坚持双向对称过滤，正常的测算结果将被大面积误杀。

跨 Chunk 边界的幽灵敏感词



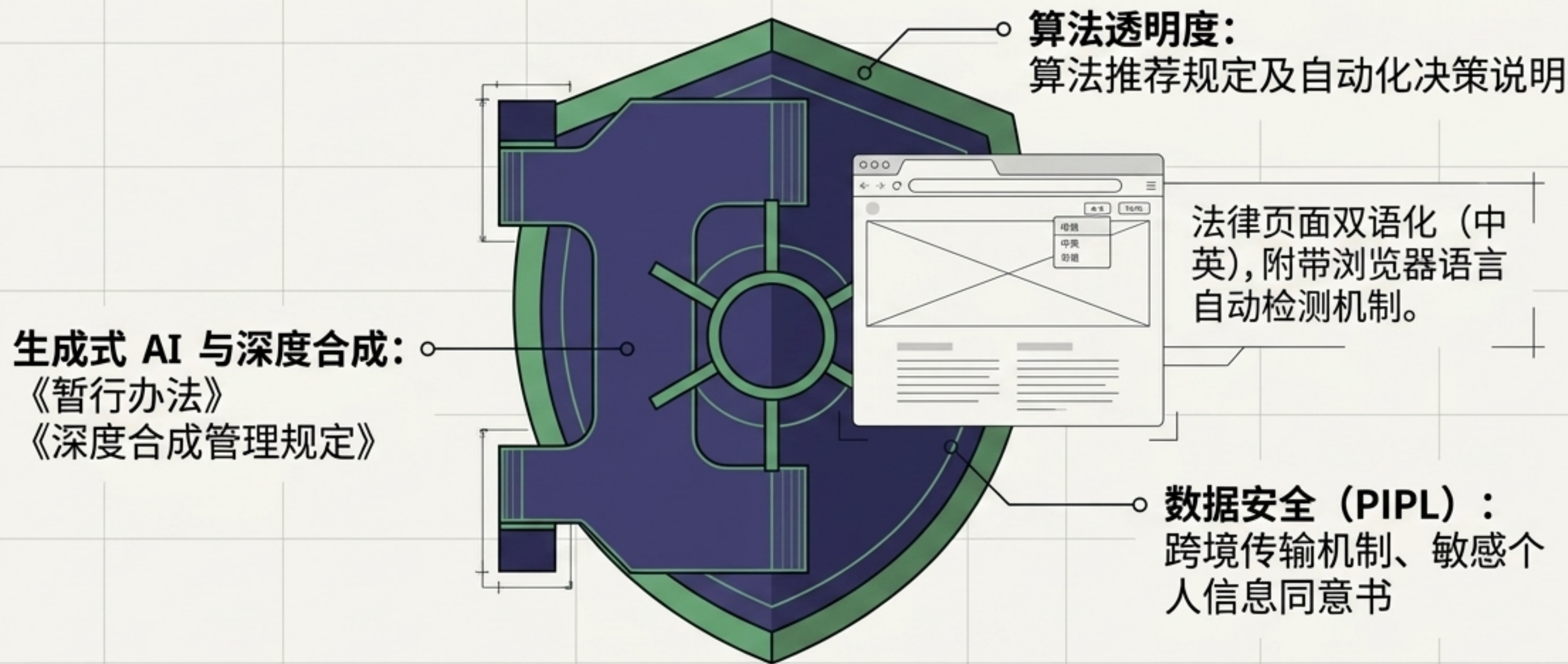
 保证流式切割绝对稳定：6 个专用测试文件，覆盖 158 个自动化测试用例

收敛技术债：i18n 标签的中心化 (PR #102)



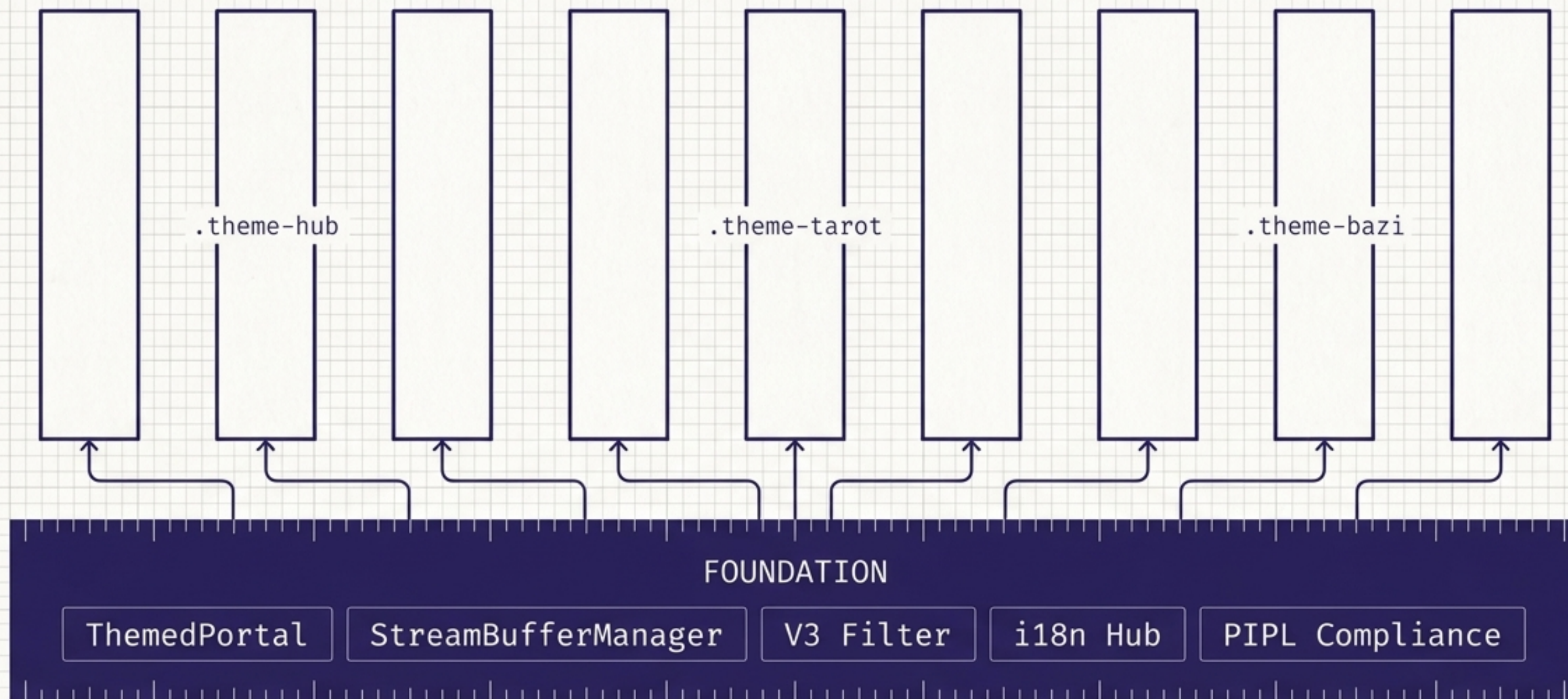
消费方删减 223 行代码，集中源新增 607 行。
从此，新增标签只有一个唯一真实数据源。

绕不开的铁件：中国 AI 法规基建 (PR #83)



附加 72 小时监管通报的 5 步数据泄露应急程序。合规不是写在纸上的口号，而是实打实的工程代码。

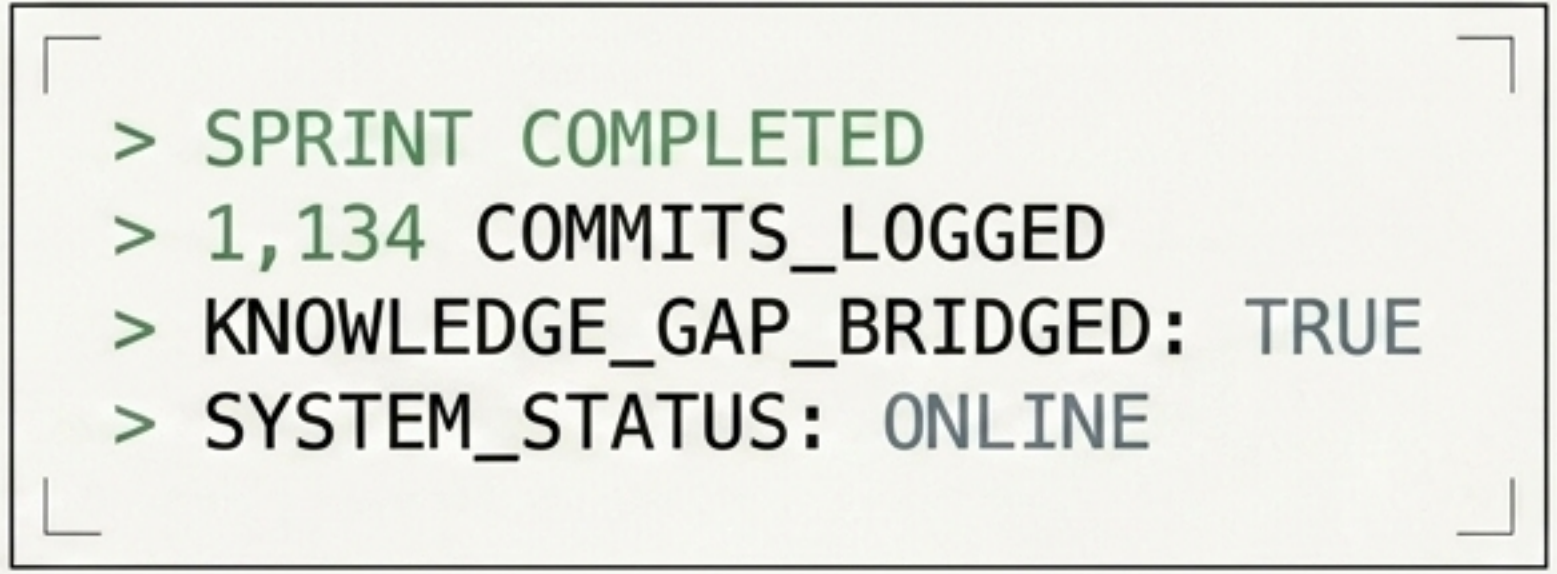
结网：29天的基础设施能承载什么？



UI同步、弹性断线恢复、双重视角过滤、中心化配置、深度合规。这五块存在最后48小时补齐的基建板砖，撑起了横跨9个产品线的商业闭环。

结论：被验证的假设

领域不懂这件事，靠 AI 是能补上的。



```
> SPRINT COMPLETED  
> 1,134 COMMITS_LOGGED  
> KNOWLEDGE_GAP_BRIDGED: TRUE  
> SYSTEM_STATUS: ONLINE
```

把一个多产品平台完整做出来，把工程之外的合规与细节亲手过一遍——
在AI增强的开发范式下，工程边界已被彻底重塑。