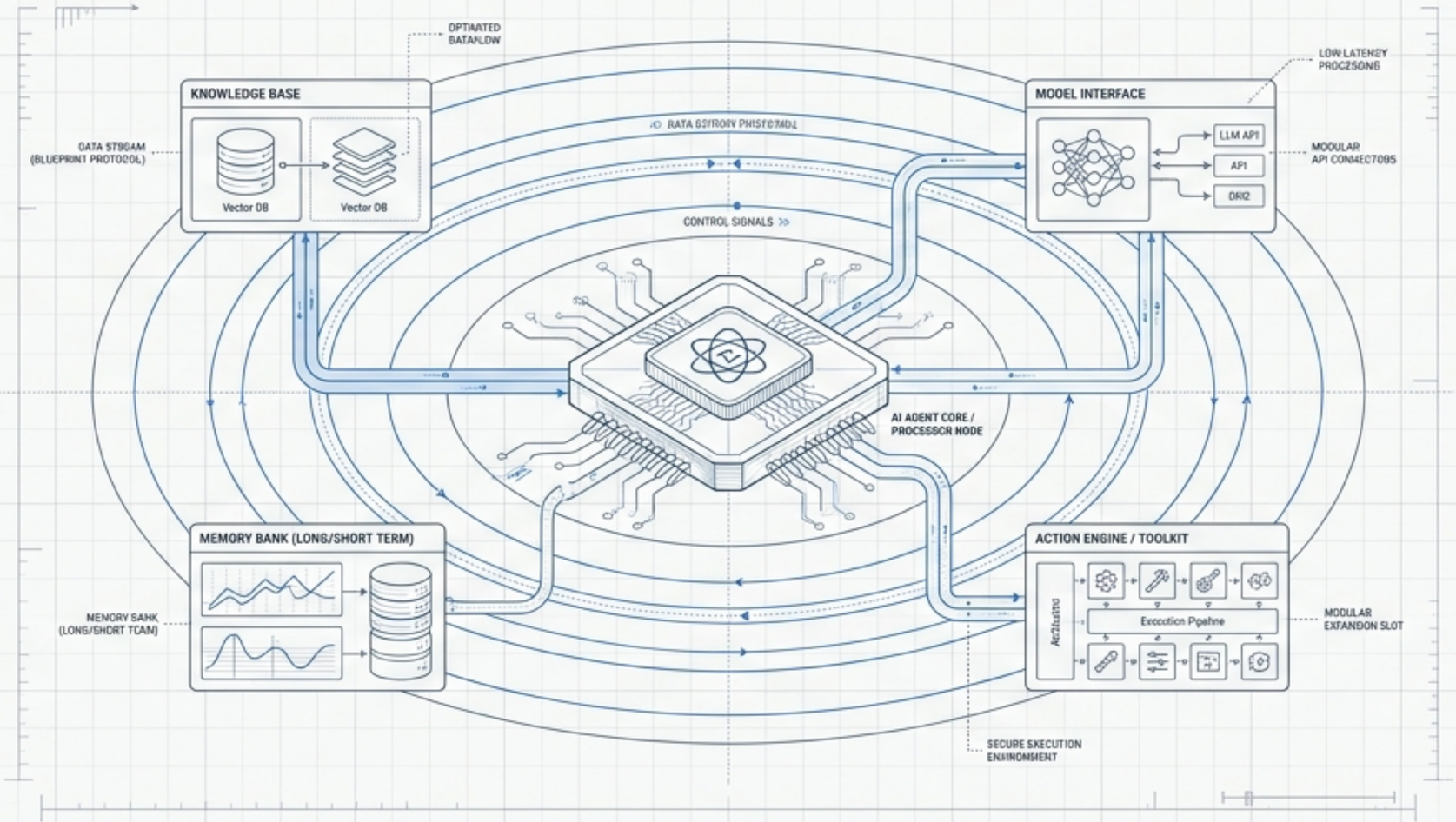
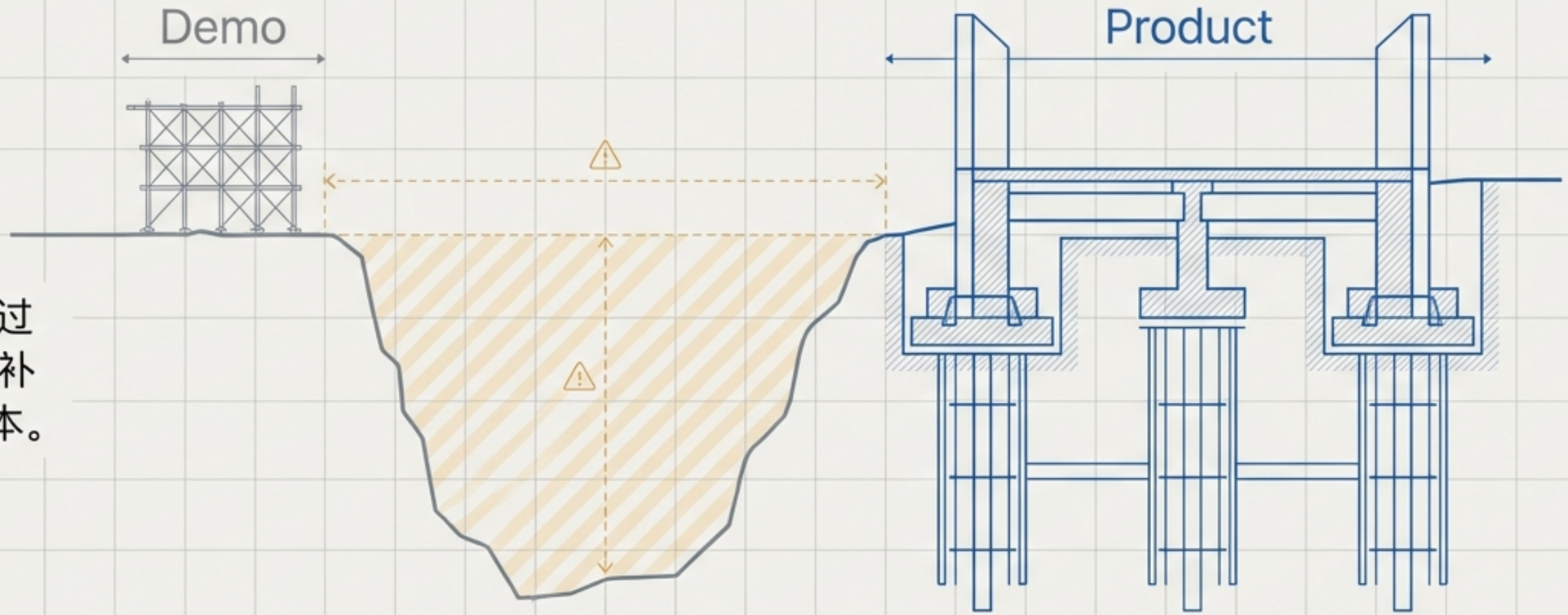
Mio v0.0.3：从「能用」到「好用」

一个生产级 AI Agent 背后看不见的架构设计。



Demo 能跑和产品能用，完全是两回事



v0.0.2 让 Mio 活了过来，但 v0.0.3 是填补那条巨大鸿沟的版本。

v0.0.3 没有任何新功能可以拿来炫耀。它是一个把所有不该碎的地方补好的版本。每一处改动都让产品离「靠谱」更近了一步。

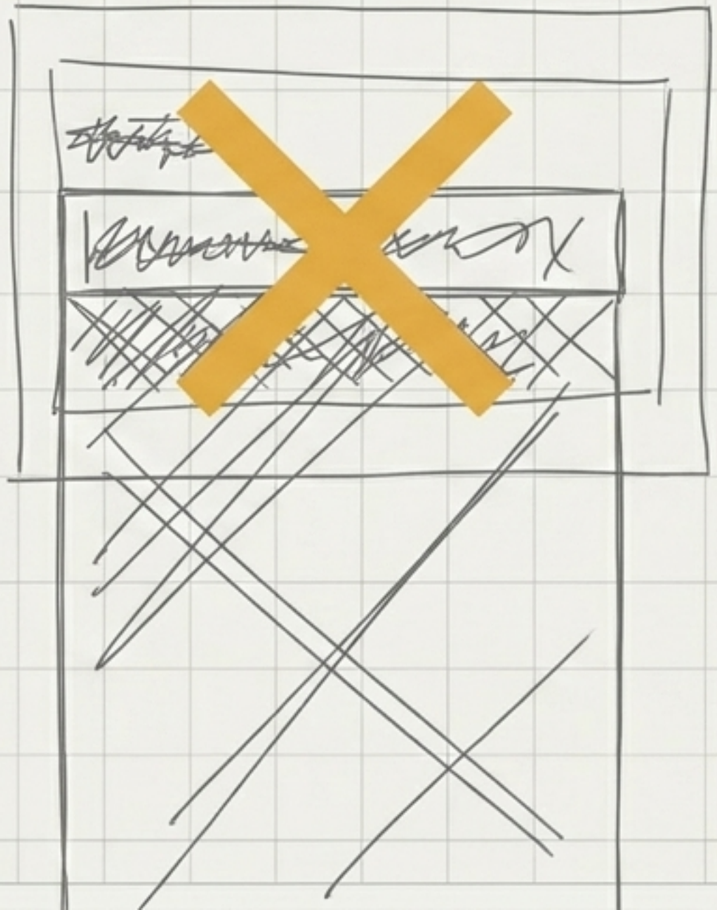
从「表演者」到「靠谱的朋友」

维度	v0.0.2 (Showman)	v0.0.3 (Reliable Friend)
输入边界 (Inputs)	默默接受空输入与 500 字长文	**逐字段严格校验与长度拦截**
主动通信 (Proactive)	瞎子摸象式模板（嘿，好久没聊）	**上下文感知，接续历史话题**
功能边界 (Boundaries)	PM 傲慢预设（默认 23:00 静音）	**用户 Opt-in，将选择权交还**
系统稳定 (Stability)	极易被边缘情况击碎	**滑动窗口限流与 400 个自动化测试护航**

一个输入框，500 字符的荒唐

问题所在（The Bug）：

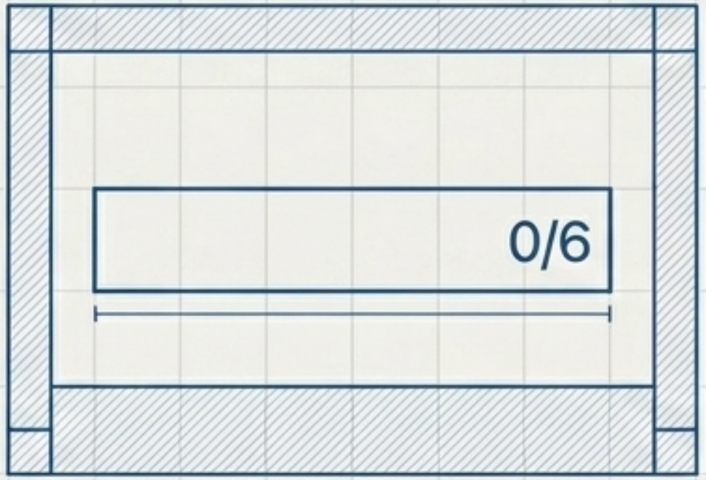
所有输入共享 500 字符上限，接受空输入



50 字 --> 10 字 --> 6 字

最终锁定（The Fix）：

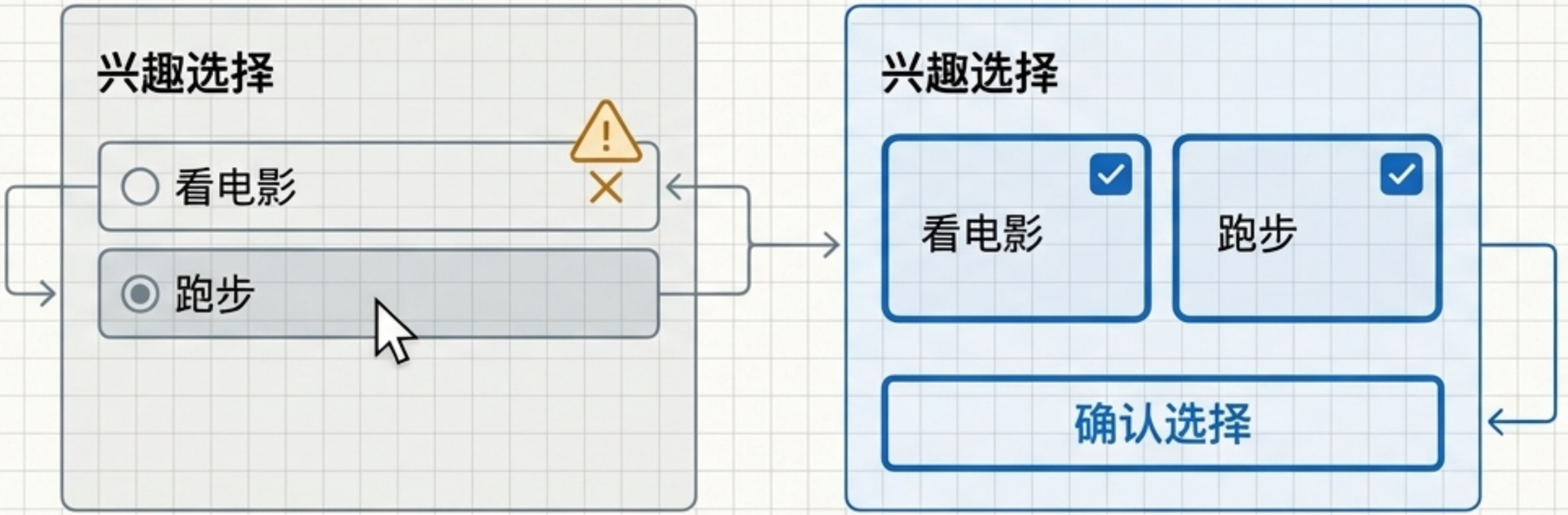
昵称 6 字，爱好 50 字，简介 500 字



限制的数字不是拍脑袋的工程决策，是被用户教育的过程。

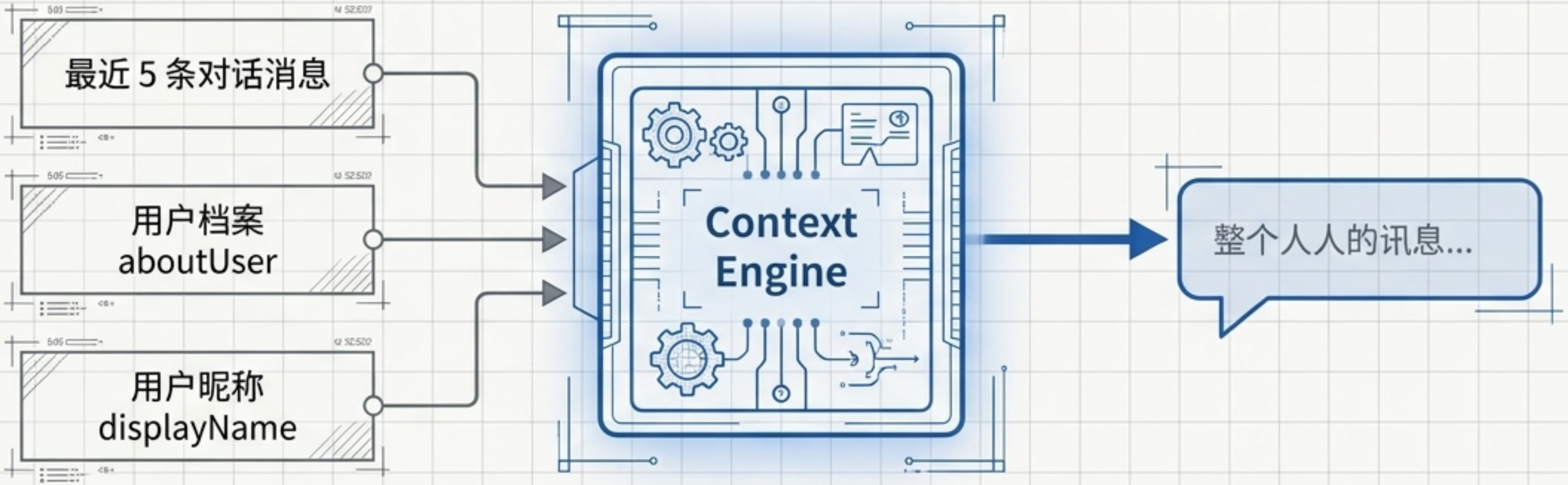
从「枯燥填表」到「自我表达」

之前：你喜欢看电影？那就不能选跑步。
这种只能单选的逻辑对用户来说极其反直觉。



将单选改为 Toggle 多选卡片看起来是小事，但它把 Onboarding 的体验本质改变了。
好产品不应该强迫用户做非此即彼的选择。

告别瞎说：上下文感知的心跳系统



无记忆的机器人

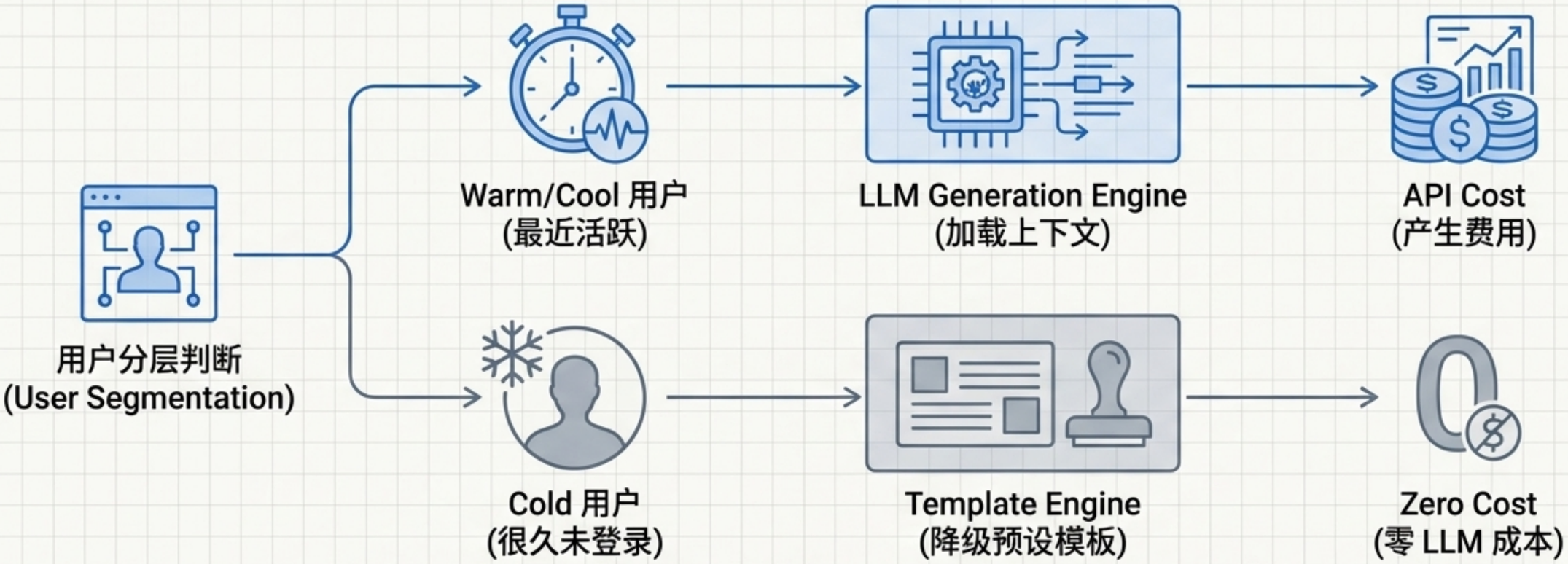
你聊了一整晚电影，第二天 TA 说：
“嘿，今天过得怎么样？”

有记忆的朋友

TA 会接上昨晚聊到一半的话题，
或者关心你提到的压力。

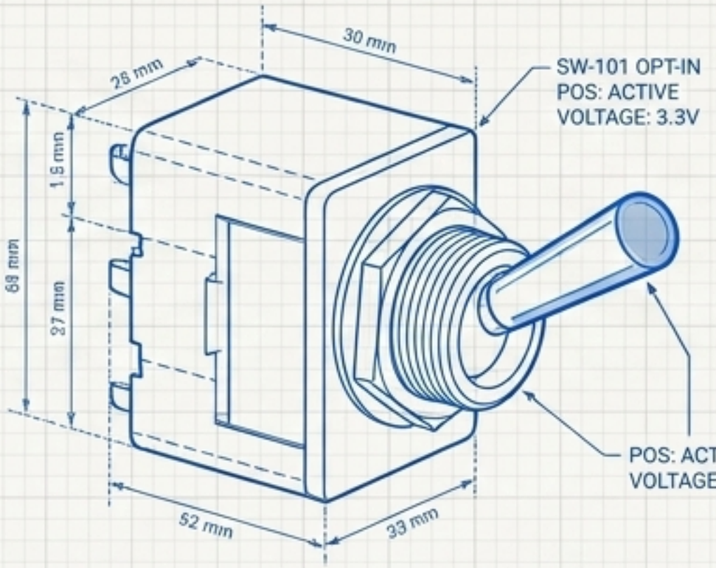
记住你说过什么，这是区分「机器人」和「真实朋友」的唯一分水岭。

智能的经济学：不要为不回消息的人付 API 费



⚠ 你不能为了一条大概率被忽略的消息支付昂贵的 API 费用。
系统需要在「拟真度」与「运营成本」之间建立冷热隔离层。

产品经理的傲慢与安静时段



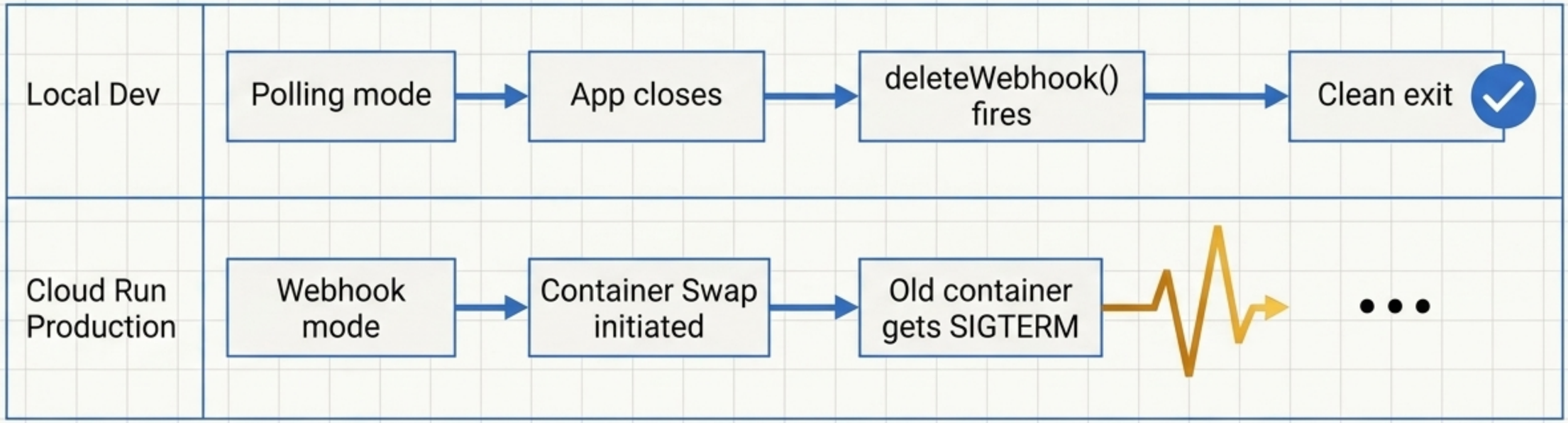
- **之前的错误预设：**默认 23:00 - 08:00 不发主动消息。认为没人想半夜被吵醒。
- **用户的反击：**我就想半夜聊天！默认静音等于默认替跨时区用户决定了作息。
- v0.0.3 **的修正：**改为纯 Opt-in（主动开启）。默认不受限制，除非用户主动设置。

别替用户做决定。你可以提供选项，但系统的默认值应该是包含最少假设的那个。

基础设施幽灵：Webhook 消失事件（上）

代码设定：收到 SIGTERM 关机信号时调用清理函数 deleteWebhook()。
在本地无比正确，但在真实的生产环境中，一切都乱了。

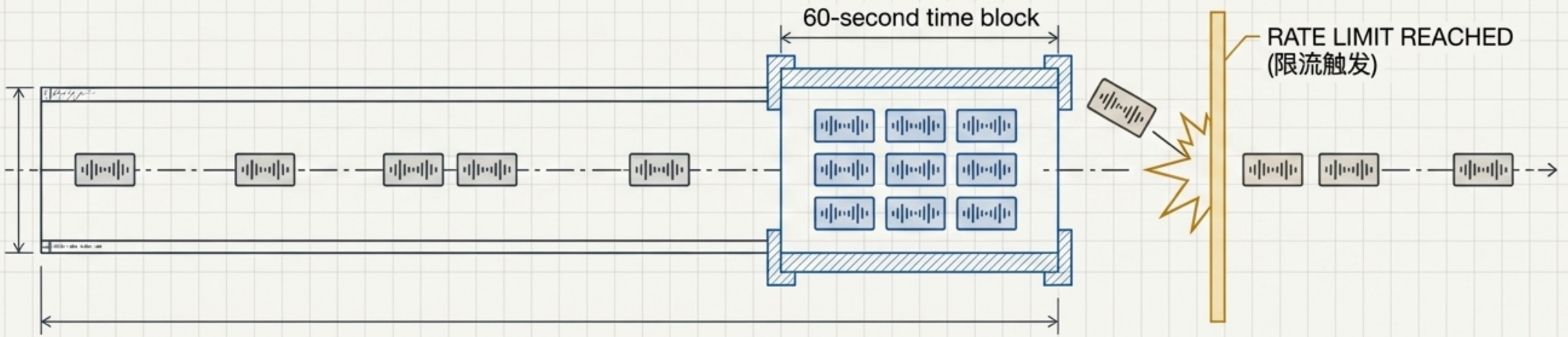
SYSTEM DIAGNOSTIC REPORT



基础设施幽灵：Webhook 消失事件（下）

- **系统崩溃**（The Bug）：新容器还没启动完，旧容器就把 Webhook 删了。在这段时间里，Mio 完全失聪，收不到任何用户消息。
- **架构修复**（The Fix）：引入环境分支逻辑。Webhook 模式下关机时什么都不做，让新容器自然接管。
- **核心认知**（Insight）：本地开发永远复现不了部署级别的混沌。这就是为什么你必须在真实的生产环境中跑你的软件。

最后的防线：滑动窗口限流



PROBLEM & ARCHITECTURE (问题与架构)

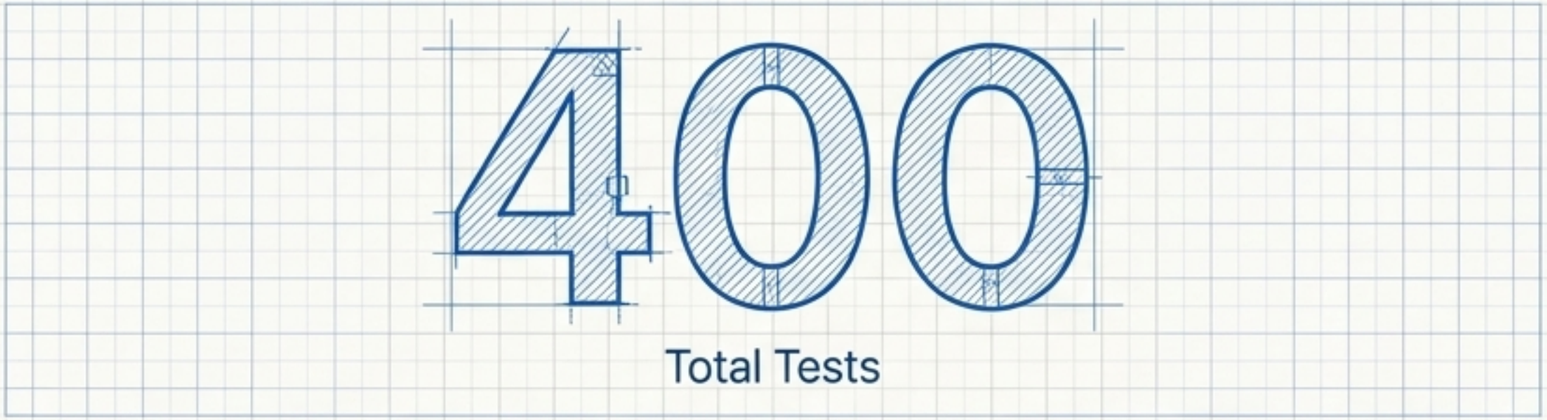
图片理解和语音转写是昂贵的操作。如果有人用脚本在 60 秒内发 100 条语音怎么办?

解决方案：引入内存级别（In-memory）的滑动窗口限流。每个用户 60 秒内最多 10 次媒体请求。现阶段无须引入 Redis，后台自动清理过期窗口。

INSIGHT (核心认知)

过早优化是万恶之源，但完全不做限流是更大的恶。限流是你与不可预测的世界之间的最后一道墙。

靠谱的代价：用安全网换取速度



+144	42	39	63
新增测试	E2E 对话	Onboarding	媒体模块

v0.0.3 的主题是「可靠性」。每一个输入校验、限流逻辑、心跳行为都需要测试保证，防止牵一发而动全身。

测试覆盖率与开发速度呈正相关。写测试看似浪费时间，实际上省下的是每次改动时「这会不会引发级联崩溃」的巨大心智负担。

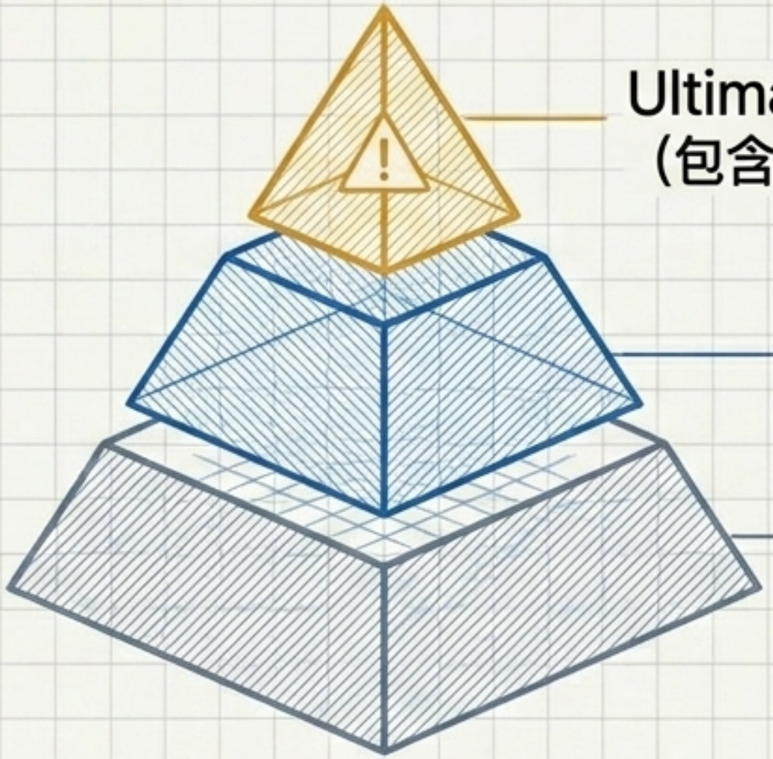
代码大扫除：复利与债务

解决所有级别的警告，而不只是 HIGH 级别。没有新功能展示，没有新指标汇报，但这是从 Demo 走向产品的必经之路。

代码债务是有利息的。团队今天图快跳过不修的警告，明天就会变成你半夜起来排查的致命 Bug。

为商业化设计架构（即使现在零收入）

v0.0.3 并没有开启付费，
但完整的 5 层订阅计划
（MONETIZATION.md）
已经就绪。



为什么现在做？因为订阅层级直接决定了底层架构（用量统计、权限拦截、年龄验证）。
事后改造 API 永远比事前设计的成本贵十倍。

叙事基础设施：扩写人物背景

扩写了所有预设人格的初始故事。不同的关系类型拥有完全不同的开场叙事和深度背景。

第一次见面的开场白决定了用户的留存。一个扁平的「你好，我是 AI 伴侣」与一个带着自己故事的相遇，是两个完全不同的产品。好的 AI 必须让人感受到「被理解」。

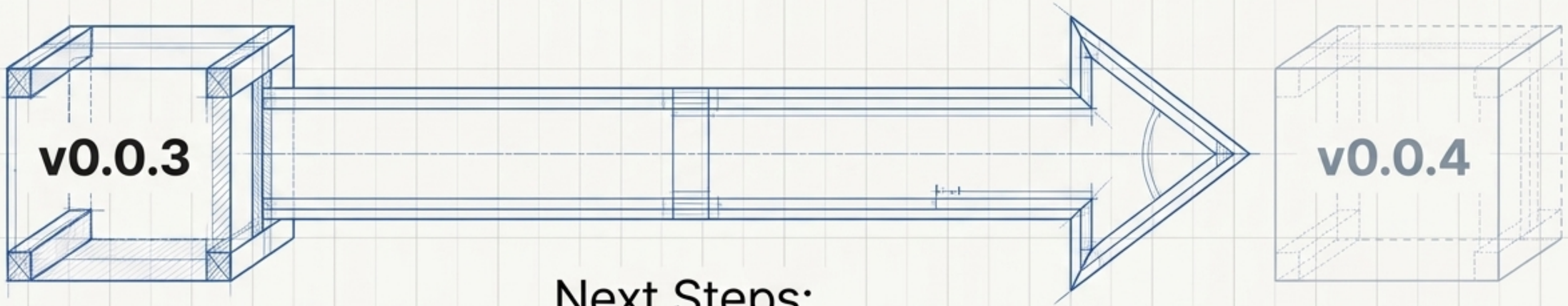
隐形的宣言（The Manifesto of Invisible Work）

用户不会注意到完美的输入校验，不会感谢你的滑动窗口限流，也不会知道你默默修好了一个部署期的 Webhook 漏洞。他们只会在系统崩溃时觉得产品真垃圾。

“好的基础设施是隐形的。你做对了，没人发现。你做错了，所有人都知道。”

跨越鸿沟：Mio 成为一个真正的产品

v0.0.3 让 Mio 有了面对真实世界的底气。输入不会报错，主动消息不再尴尬，部署不会断线，400个测试让迭代畅通无阻。



Next Steps:

- [ ] 语音深度回复
- [ ] 多轮自然记忆网络
- [ ] 真实的计费系统接入

这不再是个人的技术玩具。v0.0.4 见。