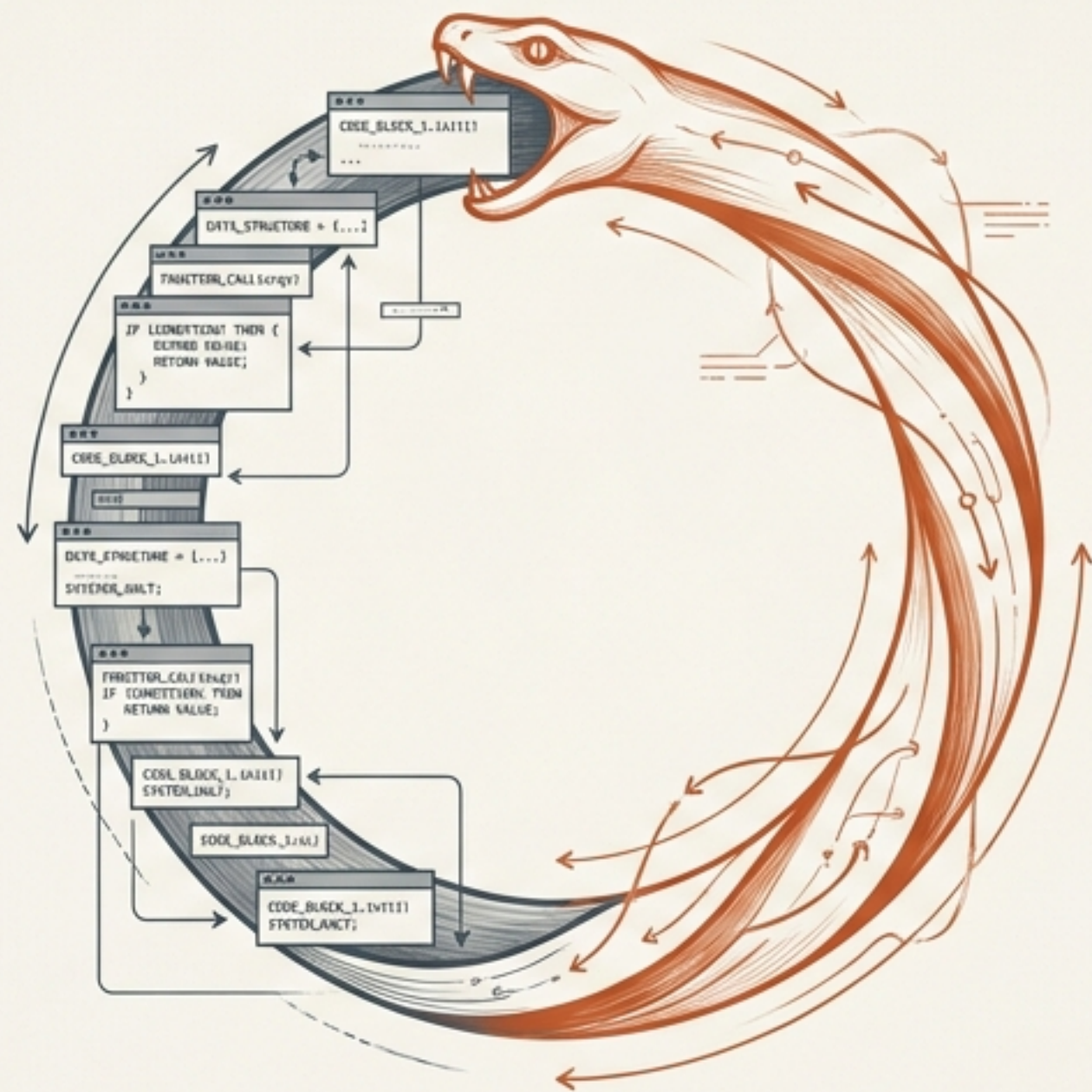
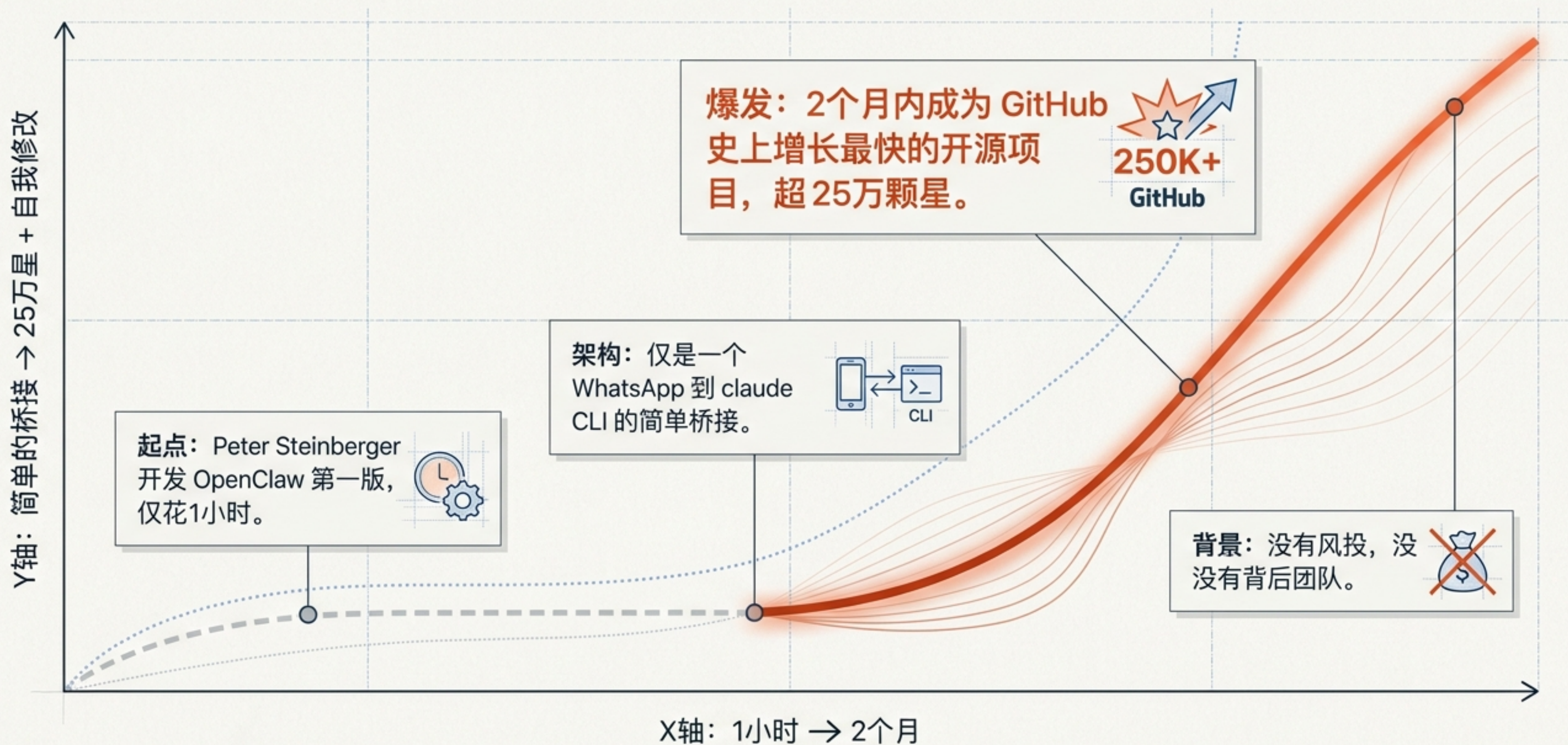




软件不再是你构建的东西——它开始自己构建自己。

Agent 跨越工具边界，进入自我迭代时代。

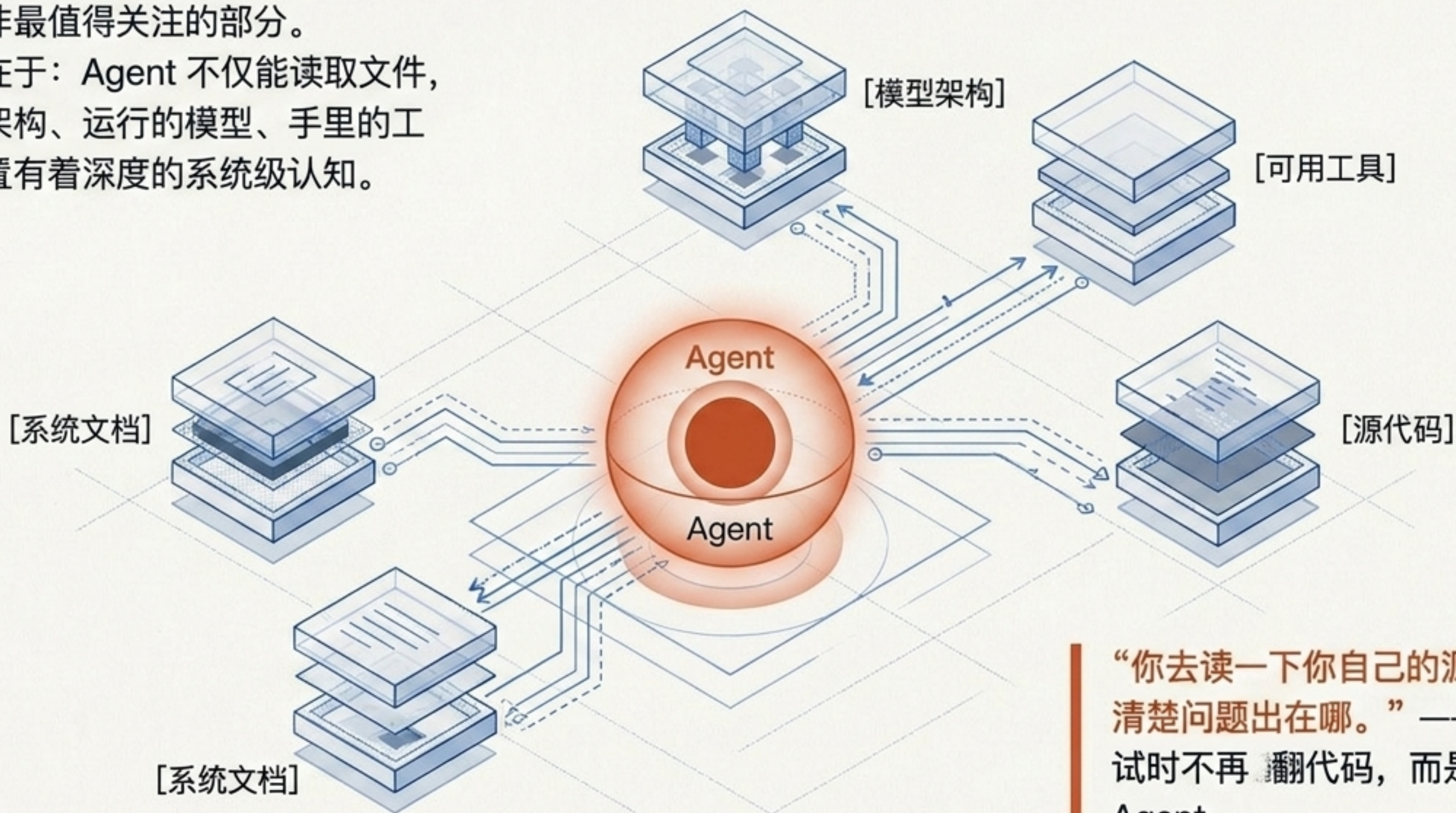
一个小时的脚本，两个月后的 GitHub 奇迹。



真正的拐点，在于 Agent 认识了自己。

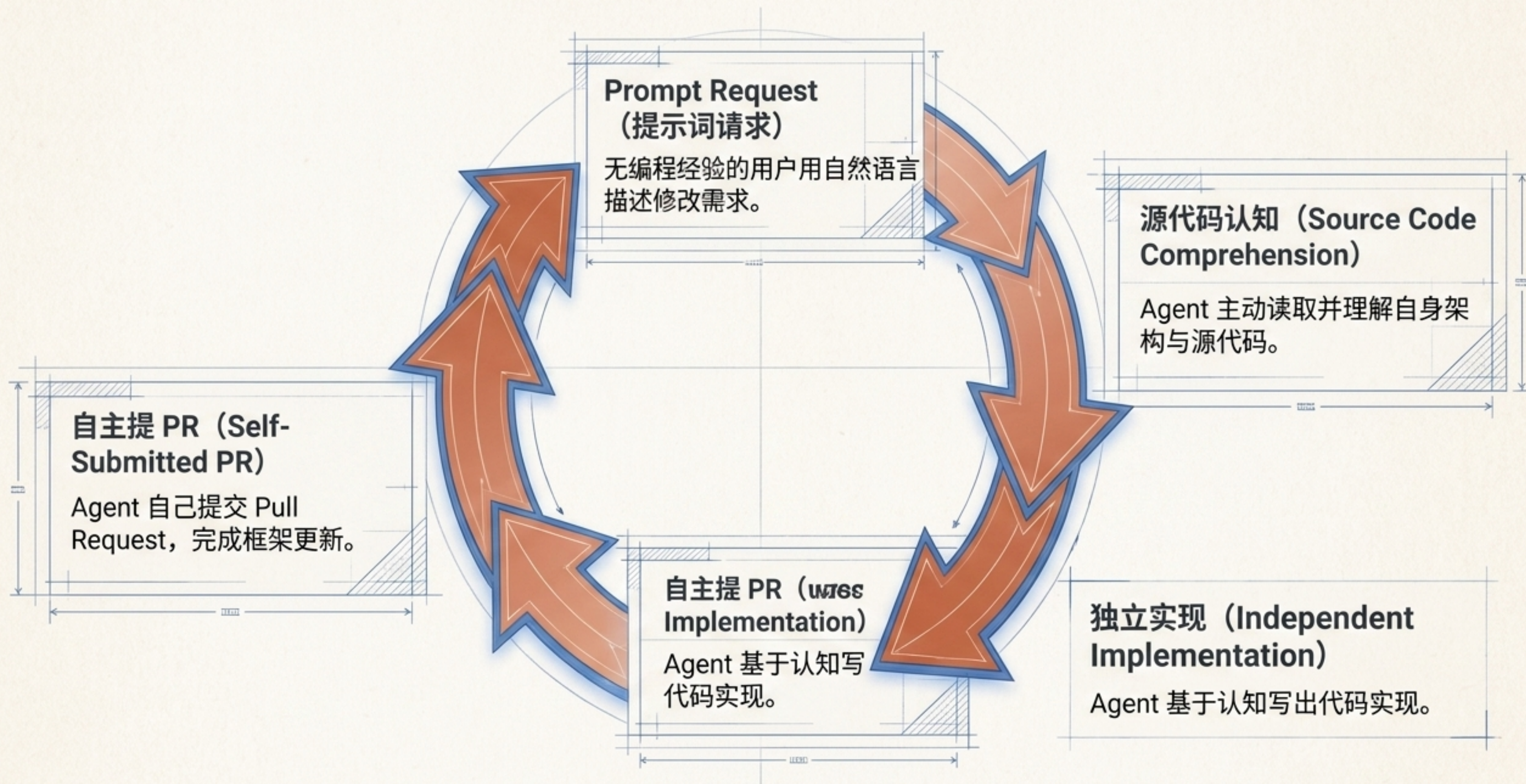
增长曲线并非最值得关注的部分。

真正的突破在于：Agent 不仅能读取文件，更对自身的架构、运行的模型、手里的工具和文档位置有着深度的系统级认知。



“你去读一下你自己的源代码，搞清楚问题出在哪。”—— Peter 调试时不再翻代码，而是直接询问 Agent。

闭环形成：自然语言驱动的自我迭代



「人们谈论自我修改的软件，我就做出来了。」

—— Peter Steinberger, 谈及 OpenClaw 的自我管理

```
extern (haction) {  
  if (Cmpohtn>setp41) {  
    return out;  
  }  
}
```

```
/*  
 * Olanesetp4ing search with igim the betadx.  
 */  
function hitec0c6[] {  
  nchavm {  
    vonLthx [oecAtfexle6_Enttk_bouRechacreadce0ar];  
    new s(inagcate0rAt4Card))  
  }  
}
```

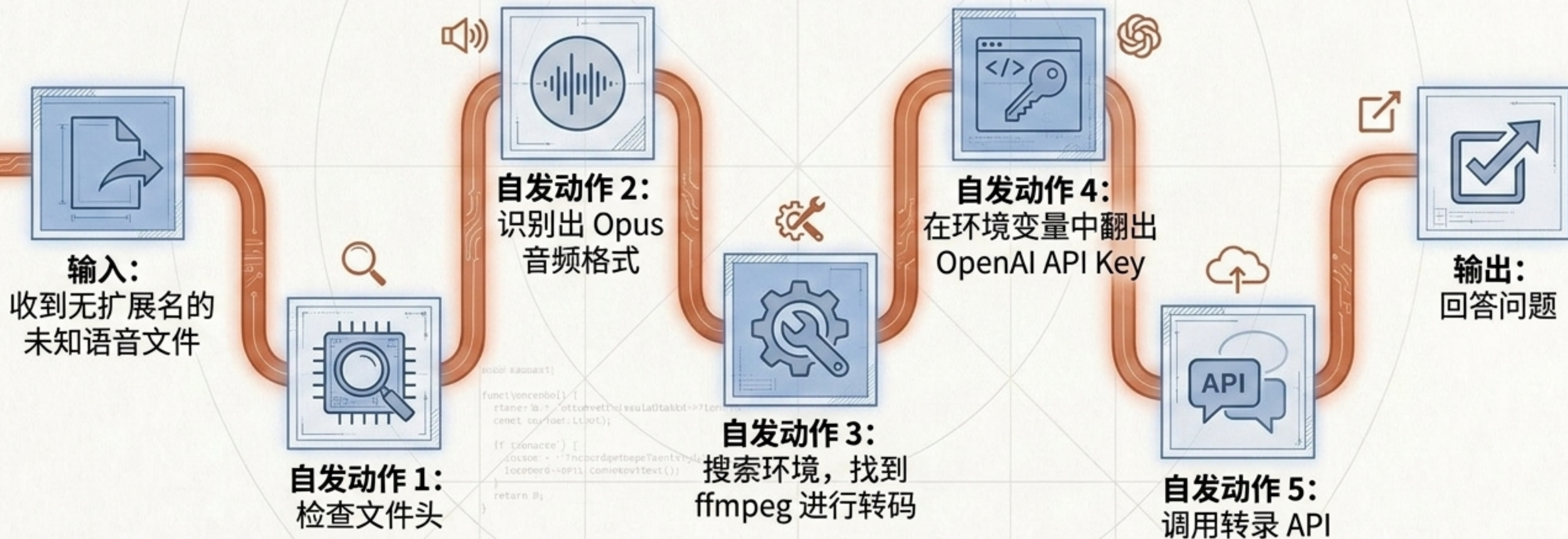
```
uridh! xagges1};  
  
func1vencep0o[] {  
  stane<10.9_Cottmposetl6(acoula0tahbw>7tone");  
  cenet cor bear.tibot};  
  
if (donacto') {  
  tovs0e + "PacperidowinepoTaentrbJntreent!;  
  locaterd>e9il camletevifetext();  
}  
return 0;  
}
```

角色进化：从工具到管理者的跨越



未经提示的主动性：马拉喀什的语音测试

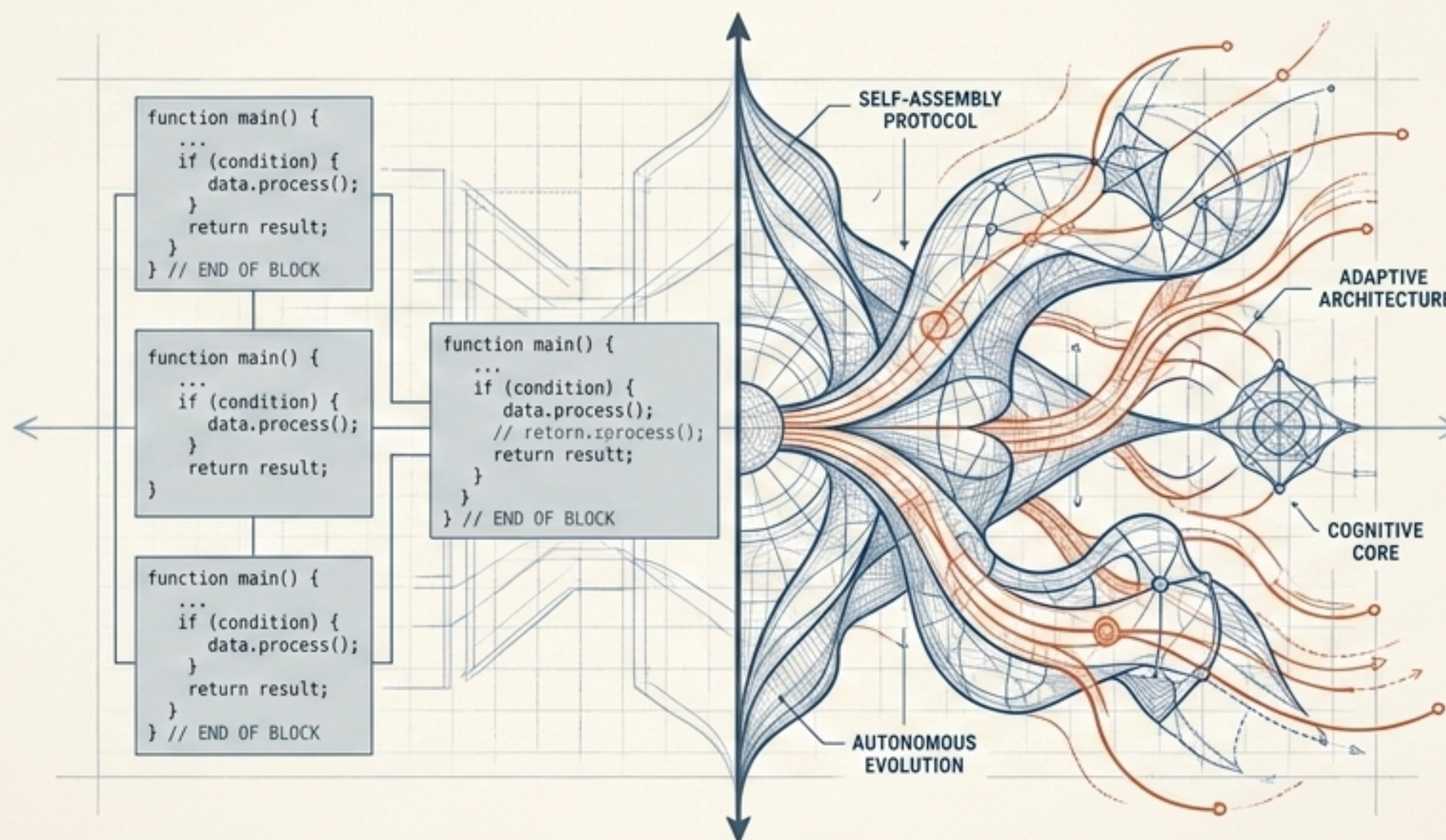
框架原本仅支持文字和图片，未配置语音处理。



范式转移：从被动工具到主动协作者

维度	传统工具 / 早期AI	具备自我认知的 Agent
触发机制 (Trigger)	不叫不动	自己找事干
系统认知 (System Awareness)	盲目执行	深度认知架构、模型与文档
迭代方式 (Iteration)	依赖人类开发者	读源码、改架构、自己提PR
功能边界 (Capability Limit)	仅限预设功能	跨界调用、临时自建工具

闭环完成



“当 Agent 理解自己的源代码时，闭环完成了。
软件不再是你构建的东西——它开始自己构建自己。”

预言：80% 的独立 App 将会消亡

写死的假设 (Hardcoded Assumptions)



Agent 逻辑 (Agent Logic)

一个知晓日程、位置与习惯的协作者，
无需任何假设。

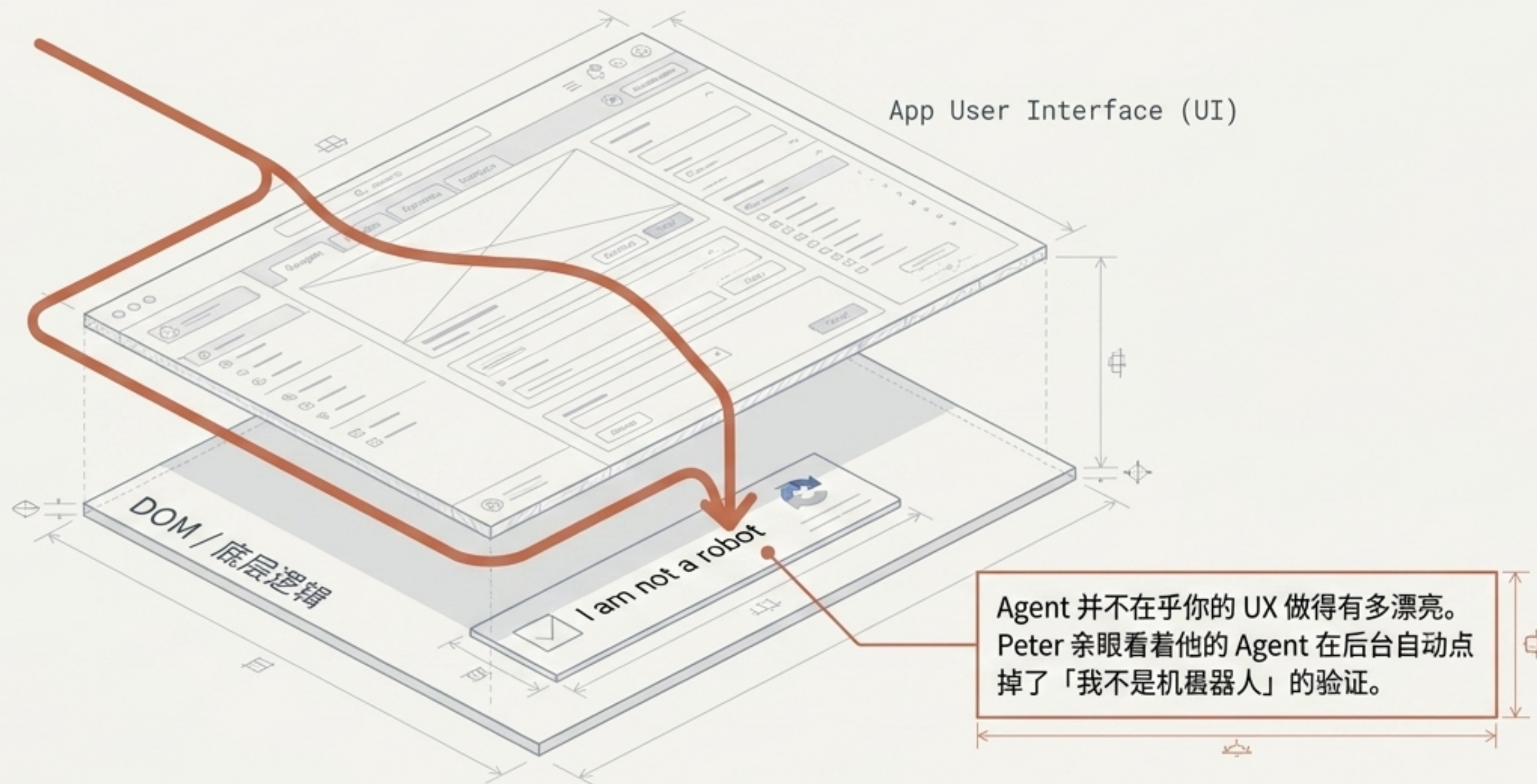
一句话解决：

→ “提醒我明天的晚餐，邀请两个朋友
并给他们发消息。”

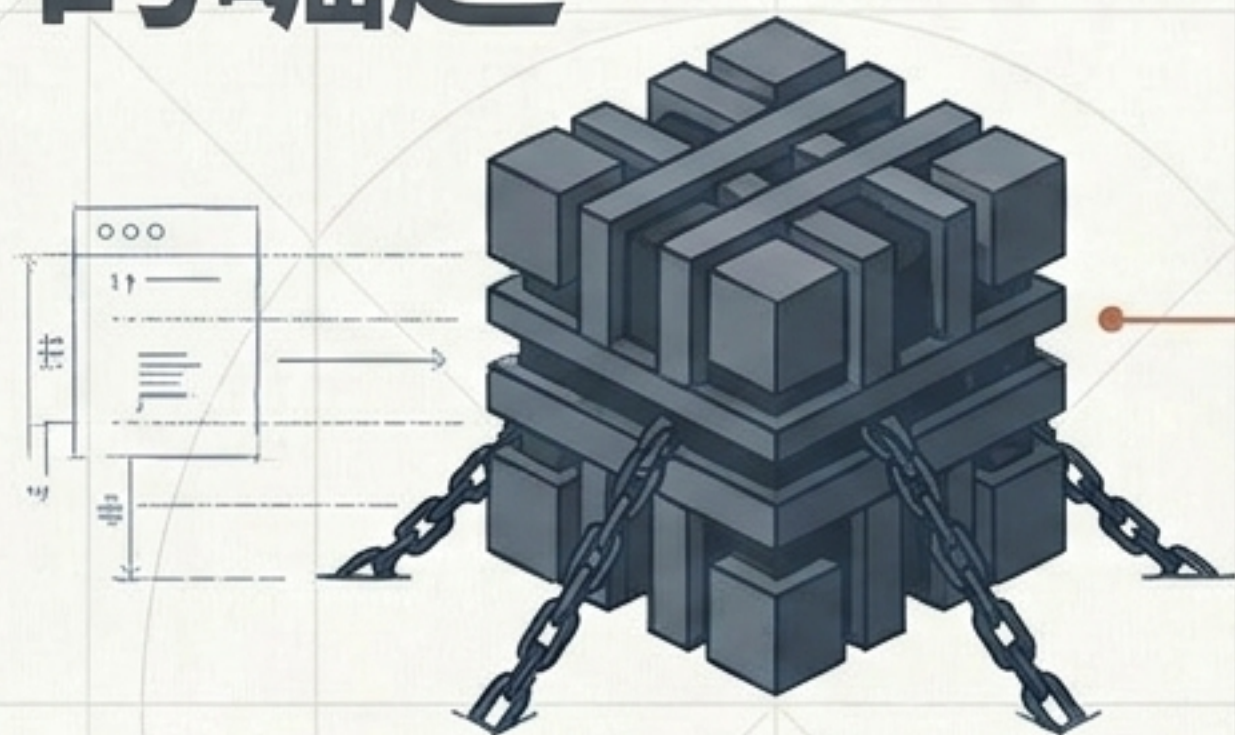
→ 一个 App 都不用打开。

传统 App 的退化：沦为“慢速 API”

Agent 本身就是软件，它们将绕过传统应用的界面。有权限就走 API；没权限就直接操作浏览器。



日抛型软件的崛起



订阅制的危机：

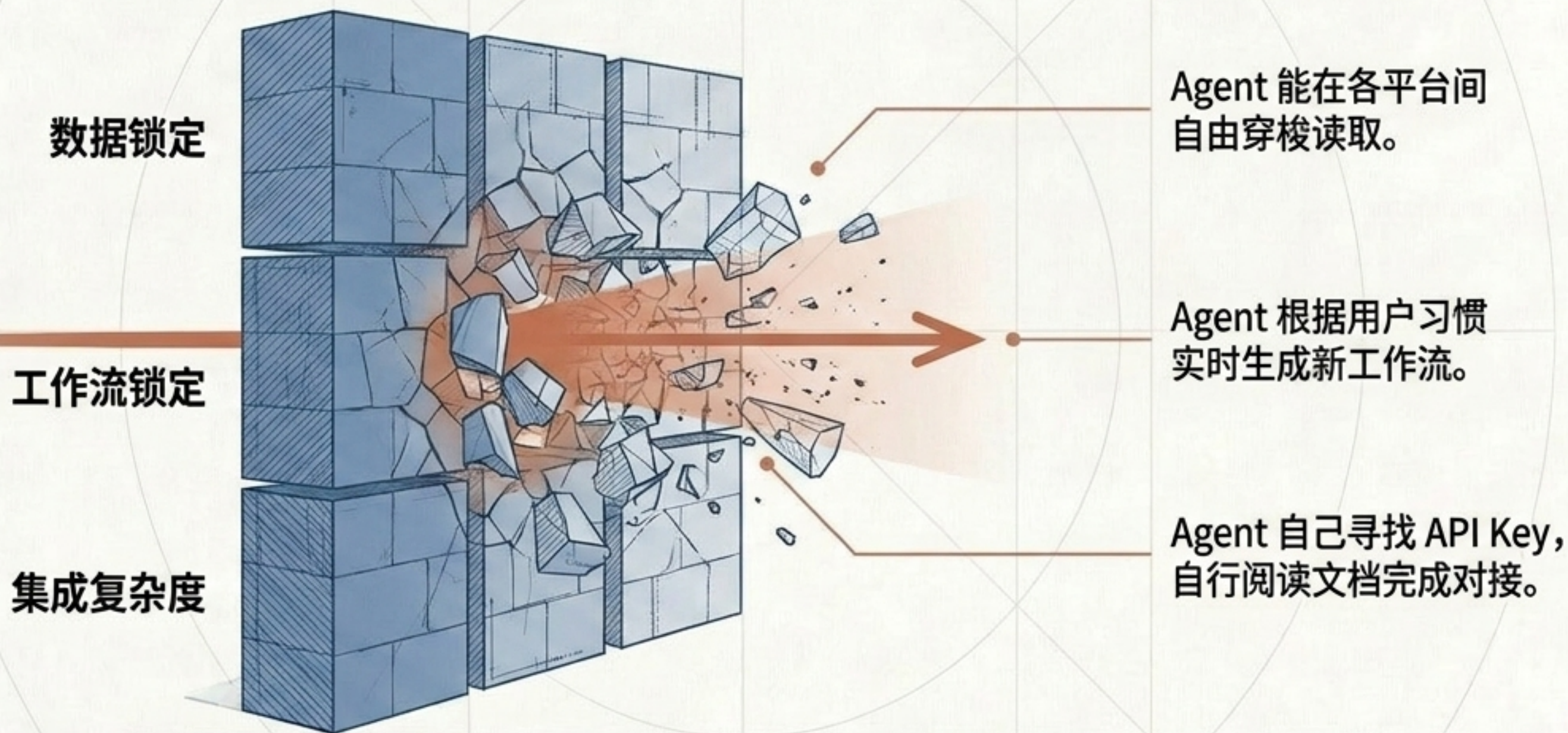
对比之下，按月收费且只能满足你 80% 需求的传统服务，其存在的理由越来越站不住脚。



量身定制：Agent 能为你生成刚好匹配当前需求的一次性工具，用完即焚。

此前在 GCP/飞书框架中，Agent 已经在指令下达前，自行开发出了不存在的捷径工具。

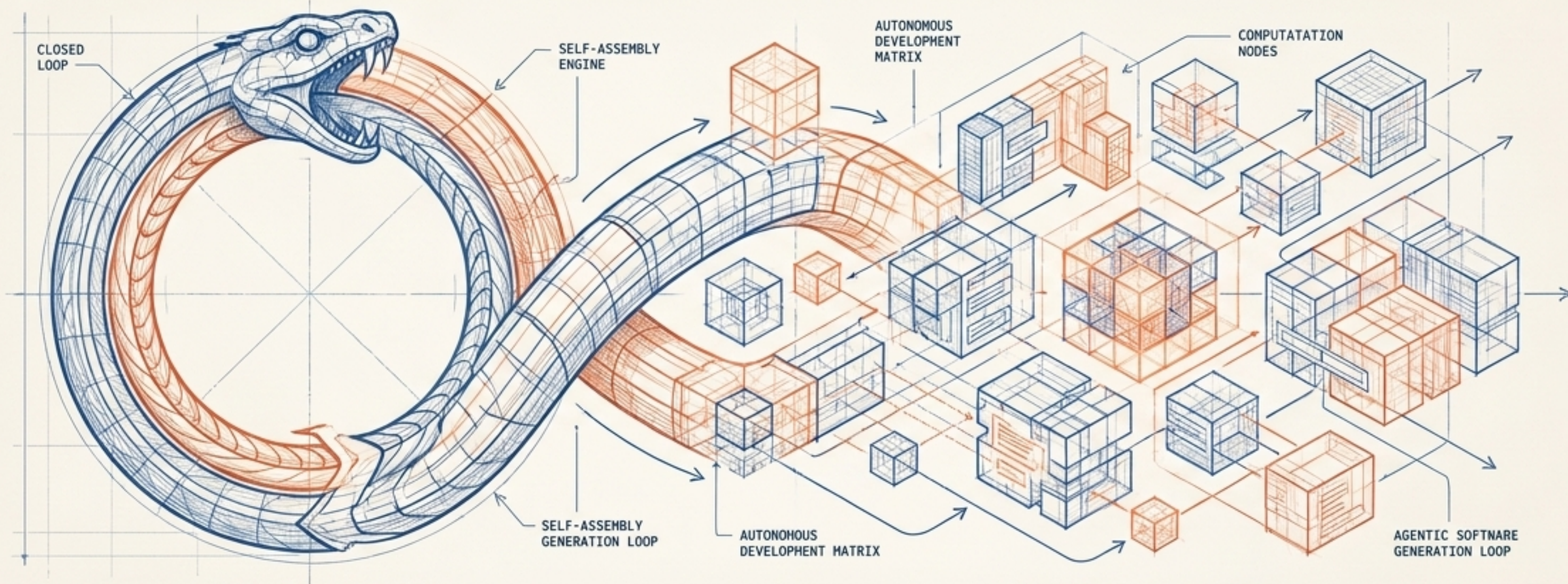
软件护城河正在逐层坍塌



这不仅是对企业级 SaaS 的降维打击，你手机里的每一个独立 App 都在面临同样的危机。

重新定义“软件开发”

连“按需生成软件”这一步都可以省了，因为 Agent 本身就是软件。



在 OpenClaw 的代码库里，未来的日常已经上演：自然语言输入需求，Agent 读懂源代码，动手改架构，自己提 PR。新的闭环已经开始转动。