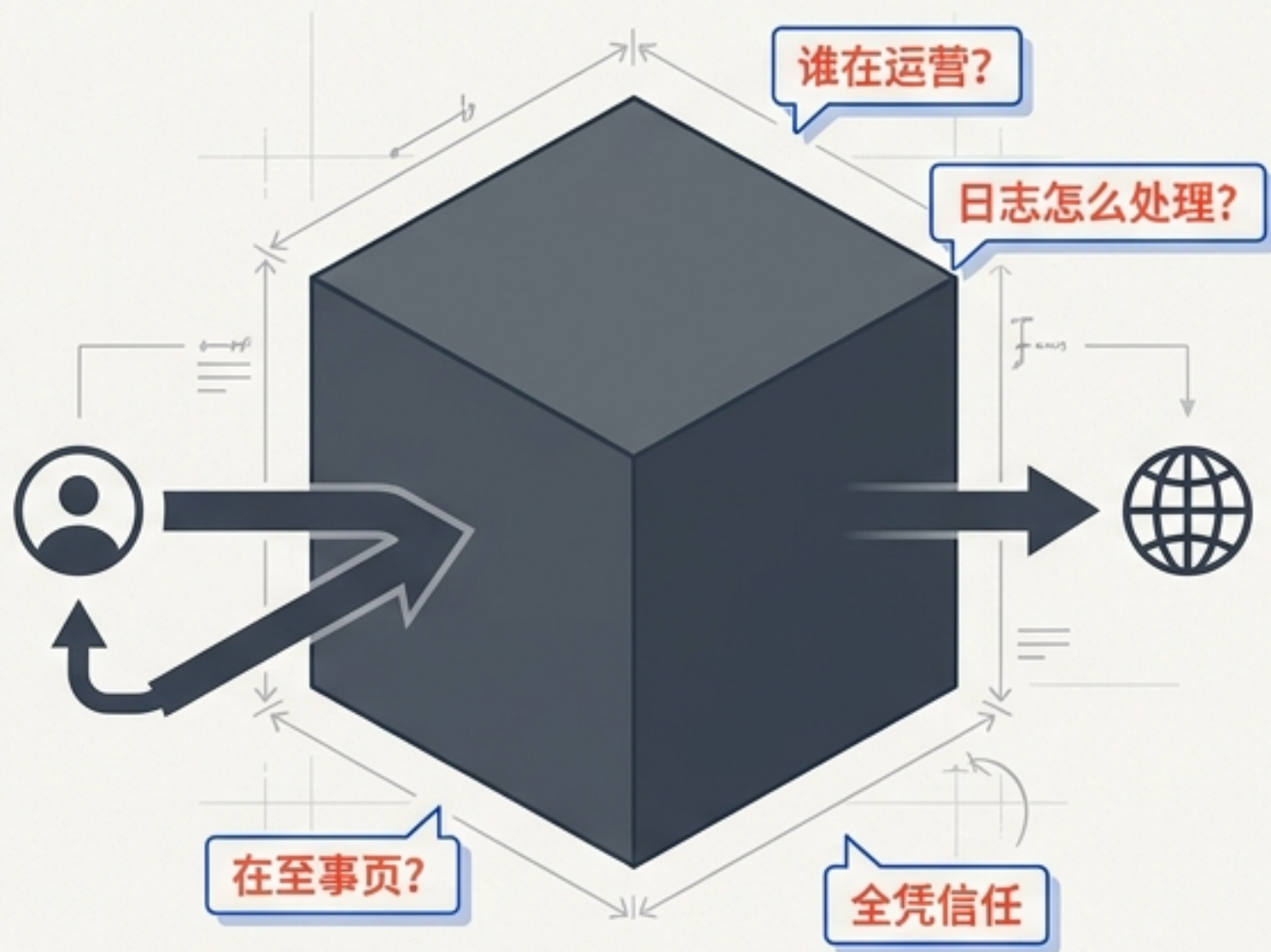


扔台 VPS 给 AI：一次属于周末的 DevOps 实验

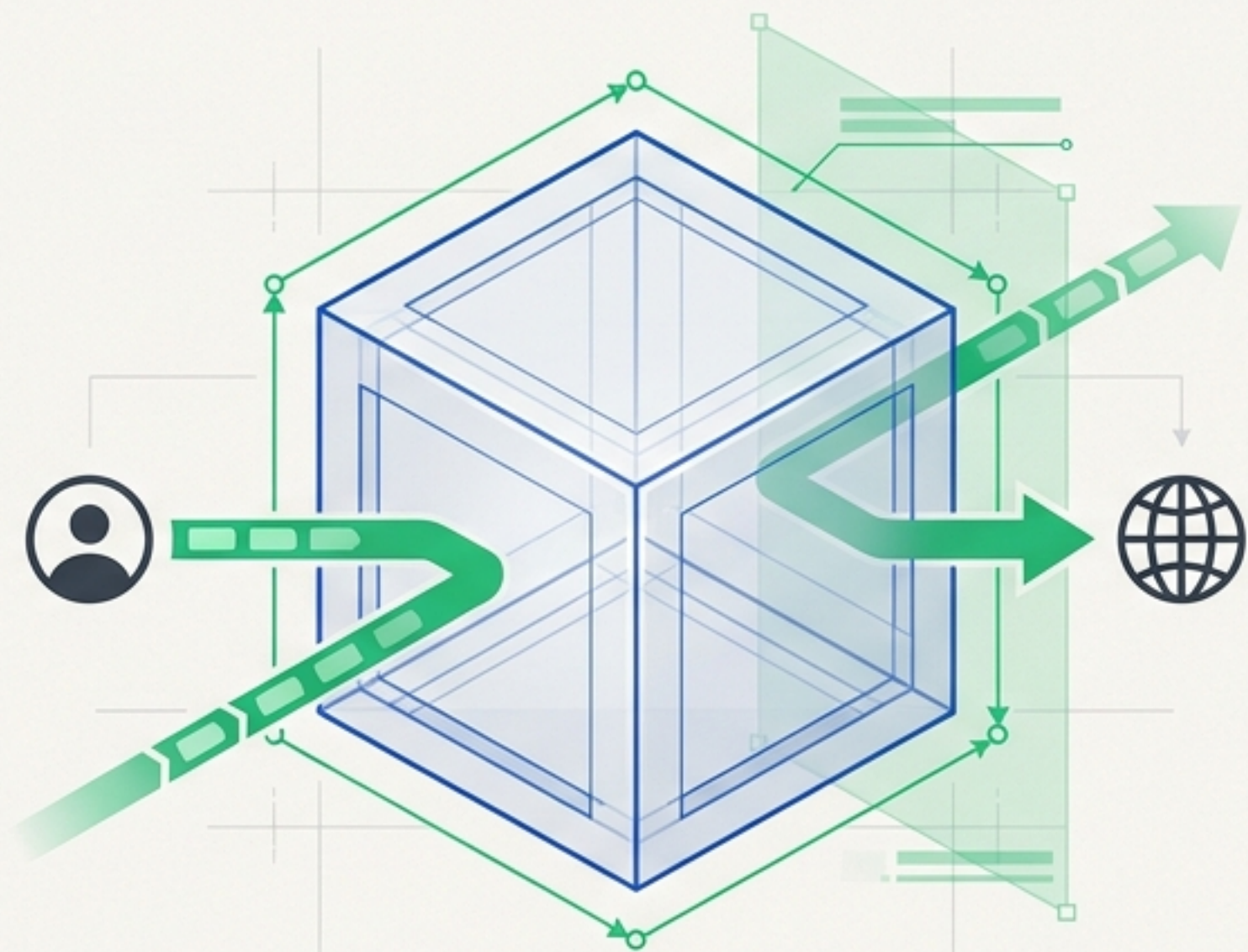
从协议被封到自动订阅：Claude Code 部署个人代理基础设施全纪实。

架构解析与 AI 结推编程报告

商业 VPN 的黑盒困境



自己动手，完全掌控



> 基础设施: Atlas Networks (洛杉矶, 9929 线路, 静态住宅 IP)

> 执行者: Claude Code

> 目标: 一套完整的代理服务 (自用 + 朋友分享)

```
root@atlas-vps:~# ssh -i ~/.ssh/id_rsa root@SERVER_IP
```

- > Claude Code,
- > 这里有一台全新的 Ubuntu VPS。
- > 帮我搭一套完整的代理服务。
- > 要求：我可以自用，也可以极其简单地分享给朋友。
- > 你有完全的 root 权限。开始吧。



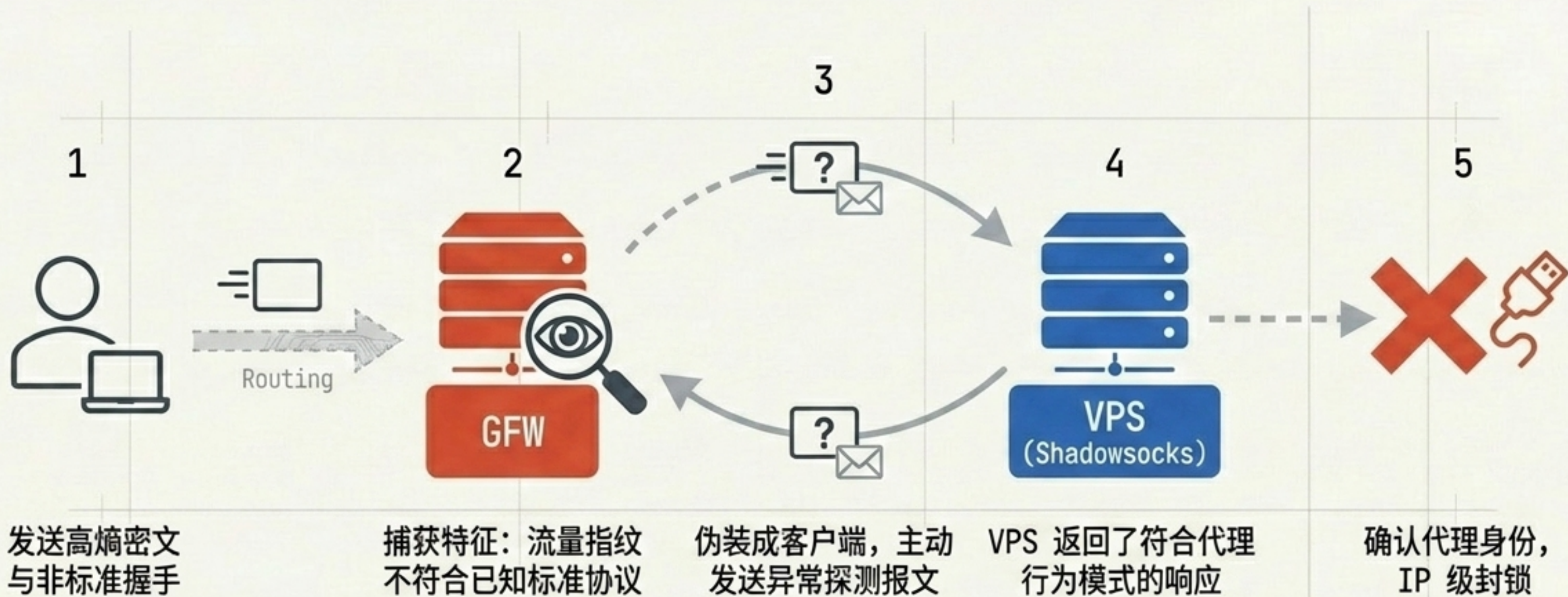
预期：一键脚本，半小时搞定。
现实：出乎意料的失败与重构。

部署与故障时间线



2025 年用新 IP 部署 Shadowsocks, 等于举着牌子喊: '来探我'。

失败溯源：GFW 主动探测 (Active Probing)

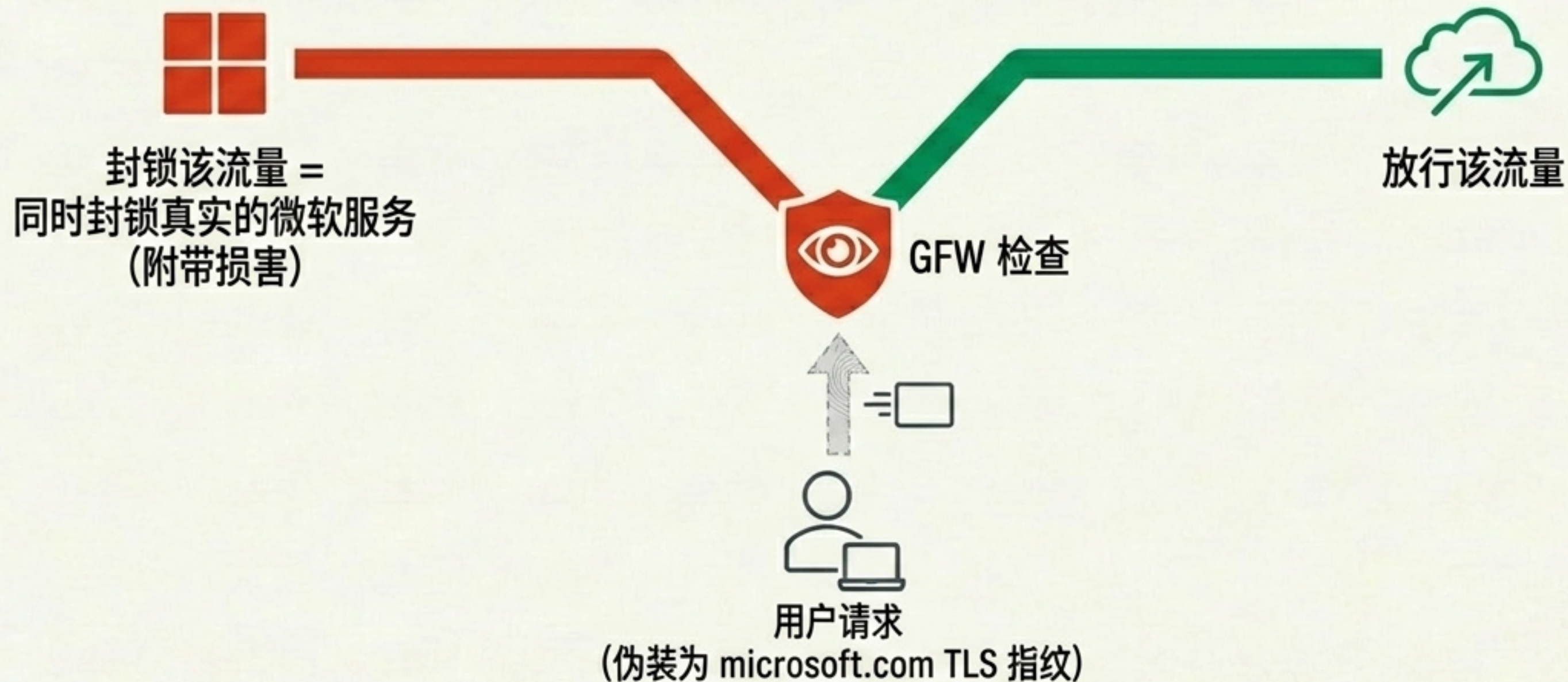


GFW 读不了内容，但能认出'指纹'。

破局思路：从'加密'到'伪装'

	Shadowsocks (上一代)	VLESS-Reality (新方案)
核心策略	纯粹加密 (让流量变乱码)	深度伪装 (借用真实身份)
流量指纹	高熵密文、非标准握手	完美的标准 TLS 握手
GFW 应对	轻易触发主动探测	迫使面临'附带损害'
存活时间	4 小时	长期稳定

VLESS-Reality: 藏在明处的艺术



GFW 要么允许你过，要么把微软也封了。它永远不会选后者。

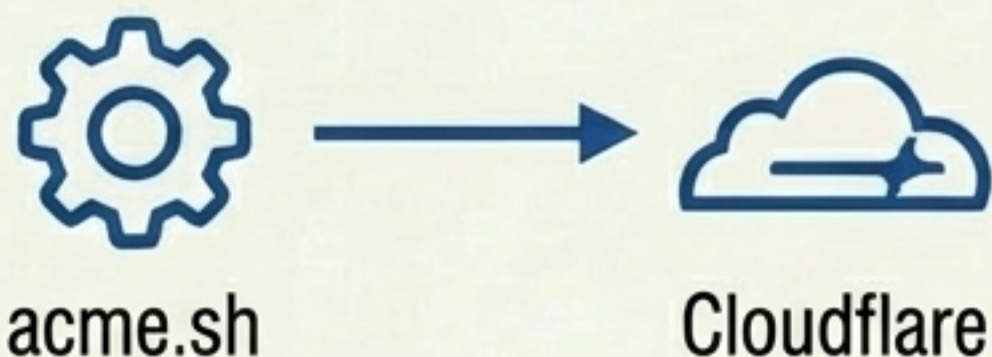
四层协议栈：优雅降级与兜底架构

(sing-box 底层驱动，仅开放四个特定端口，其余全关)



自动路由逻辑：Reality 够用就永远走 Reality。

基础设施建设：SSL 证书与静默验证



工具： acme.sh + Cloudflare DNS API

优势： 服务器无需暴露 80 端口。分配仅含 DNS 编辑权限的 API Token，创建 TXT 记录验证所有权，验完即焚。

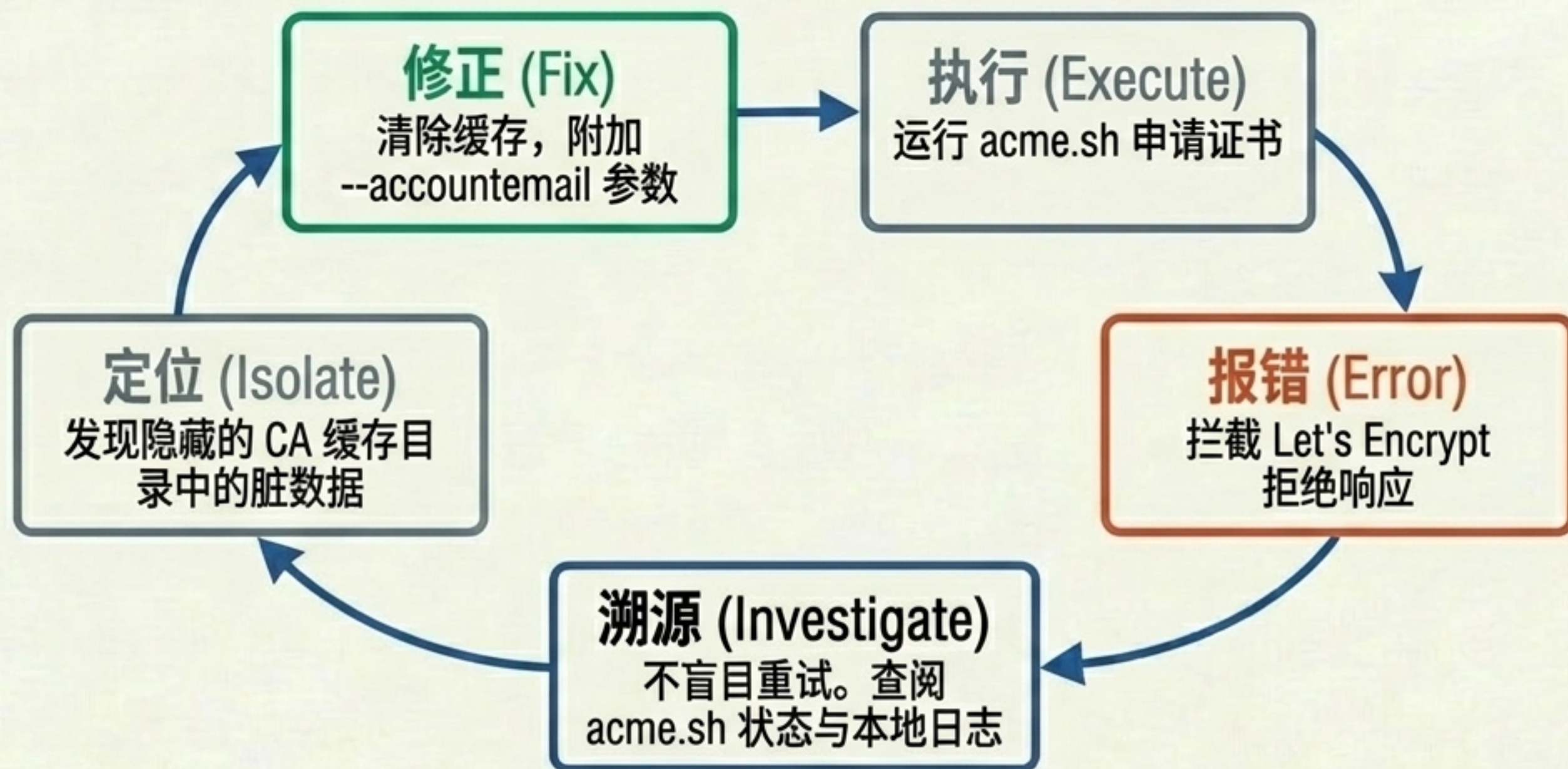
Let's Encrypt Error:
Invalid email format.



“缓存中毒”的踩坑记

sing-box 脚本曾将测试域名错误写入了 acme.sh 的邮箱成缓存字段。单独申请时，旧缓存被复用，导致 Let's Encrypt 直接拒绝请求。

AI 是如何面对失败的？

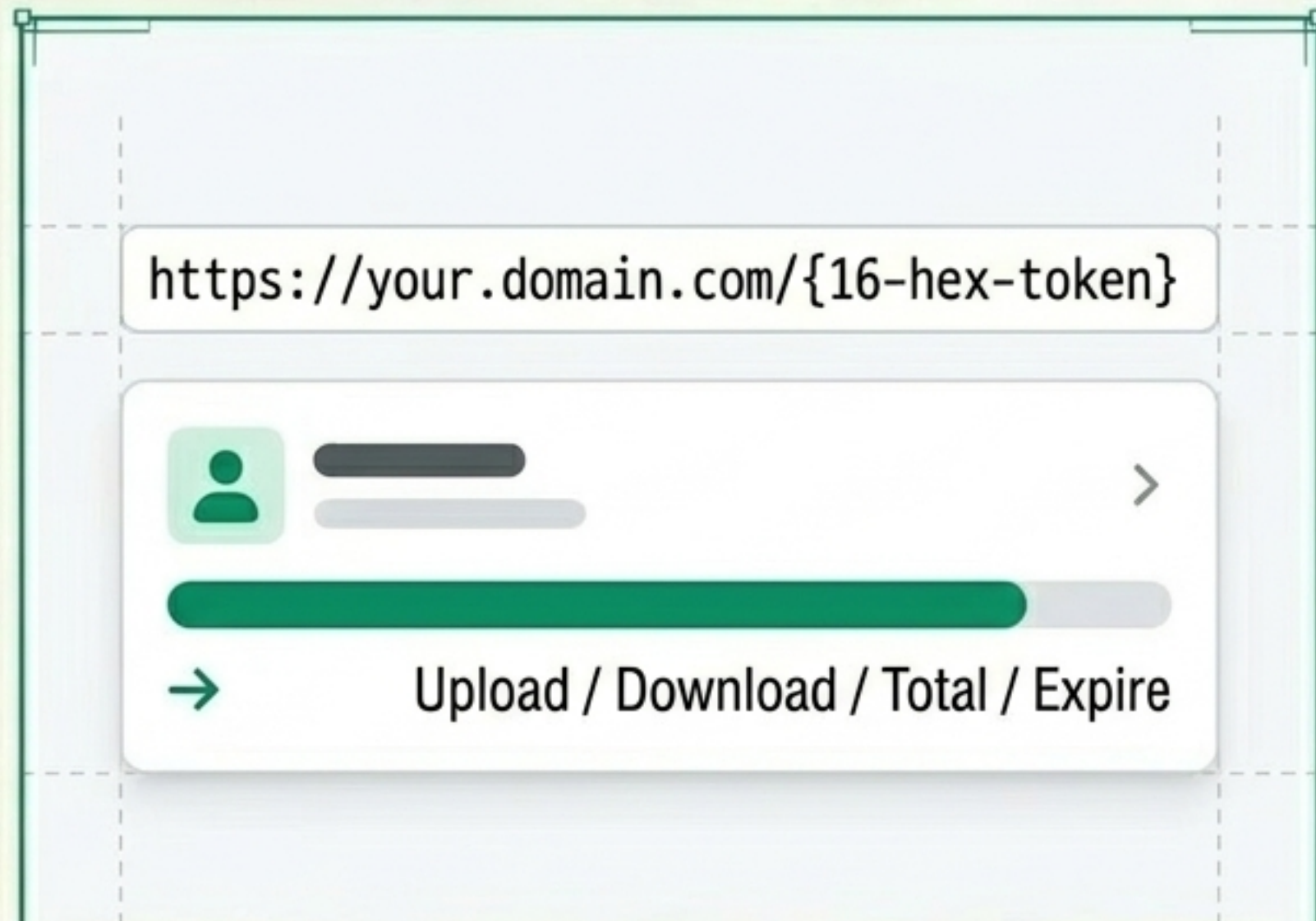


每次失败大约 10-15 分钟。它没有绕开，也没有放弃——一直到找到真正的根因。

最后一公里的 UX 追求：不再是拼凑的字符串

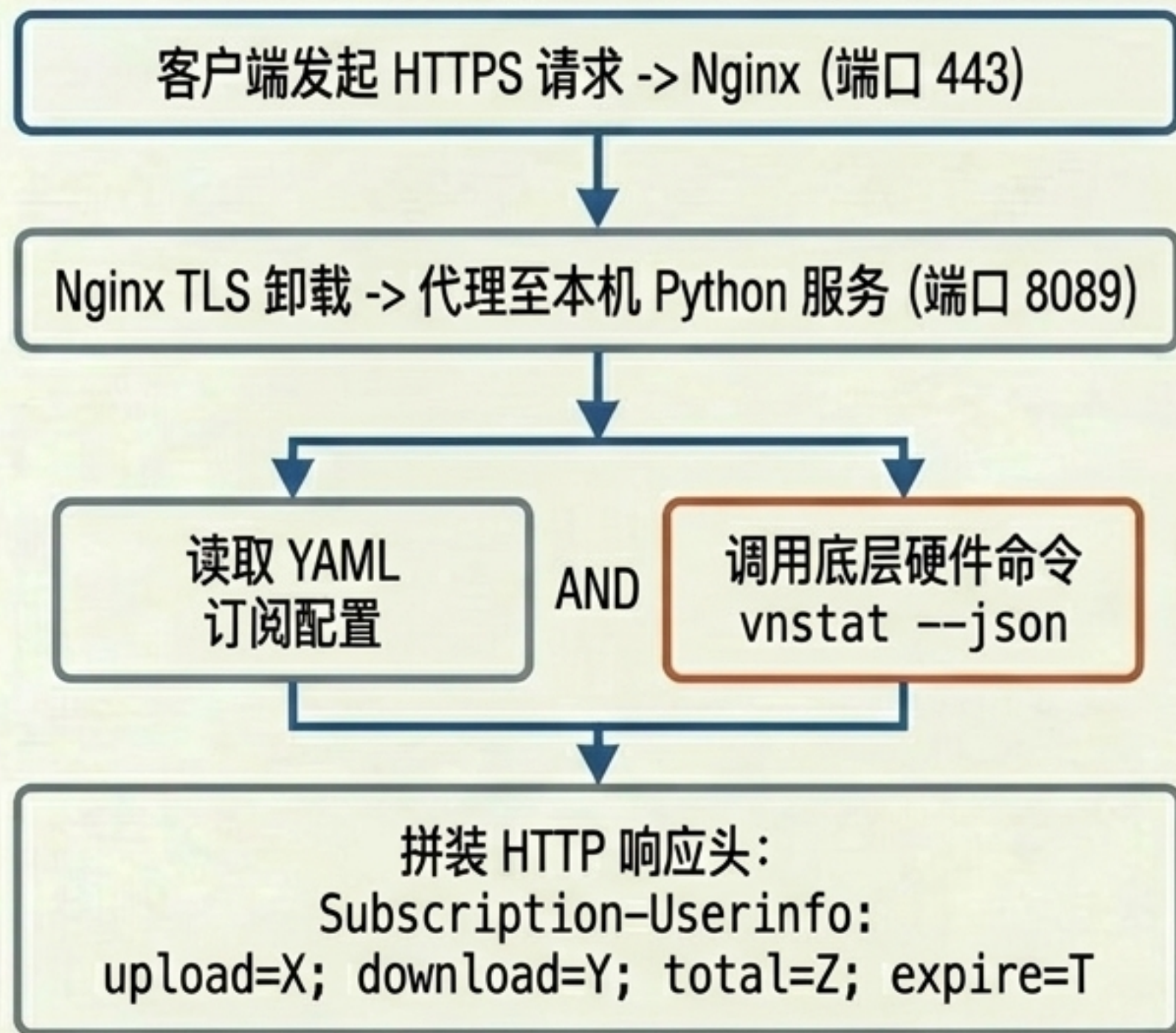


传统方案：手动导入，无状态，
像临时拼凑的工具。



现代方案：动态订阅。粘贴链接，自动拉取
配置与流量统计，体验媲美商业服务。

动态订阅引擎：打通从内核到 UI 的链路



隐藏的暗礁：当客户端遇到 Python Server

问题：Clash Verge 在正式拉取配置前，会先发送 HTTP HEAD 请求来探测 header。最初的服务仅处理了 GET，导致直接报错。

```
class SubscriptionHandler(http.server.BaseHTTPRequestHandler):  
+   def do_HEAD(self):  
+       self.send_response(200)  
+       self.send_header('Subscription-Userinfo', get_traffic_data())  
+       self.end_headers()  
   def do_GET(self): ...
```

结论：这种端到端的协议细节，只有在真实场景跑起来才能发现。

人机协作的边界：客户端的“霸道”逻辑



AI 的盲区

Claude Code 检查了服务器下发的所有 YAML 语法和 Header，一切完美。它无法感知客户端软件内部的黑盒逻辑。



人类的直觉

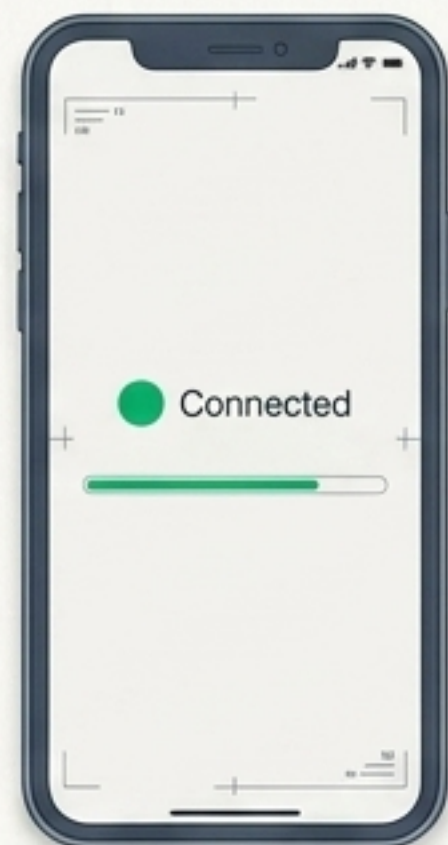
通过本地观察发现：Clash Verge 重启时会强制覆盖/删除手动编辑的配置，只保留通过 URL 导入的体系。

必须完全依赖动态 URL 订阅，放弃任何客户端本地的缝补。

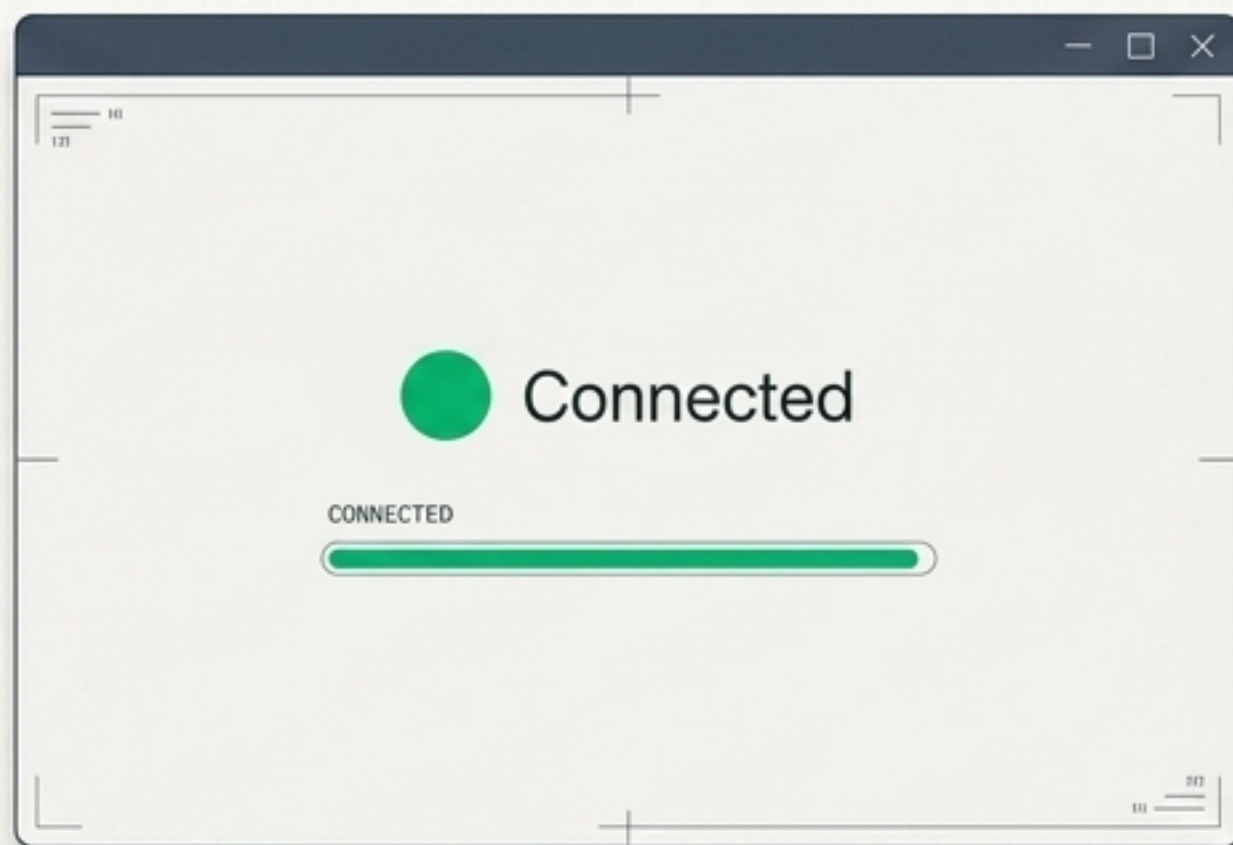
结论：AI 辅助 DevOps 的能力分界线

可验证状态执行 (Verifiable State Execution)	上下文直觉 (Contextual Intuition)
AI 统治的领域	人类不可或缺领域
<ul style="list-style-type: none">- 只要任务有明确的验证标准（如‘URL 必须返回特定格式的 YAML’）。- AI 能够无限循环调试：查阅日志、排查语法、修正缓存、补齐HEAD 请求。- 干净且冷酷。	<ul style="list-style-type: none">- 面对黑盒系统的特有行为（如 GFW 的主动探测逻辑、客户端的配置覆写机制）。- 人类需要建立测试假设、观察客户端表现、并将系统常识传递给 AI。

一个下午的成果



iOS Shadowrocket



Clash Verge

“分工明确，效率就高。一个下午，一套自己托管的代理服务，朋友只需粘贴一个链接就能用——看起来跟商业服务毫无二致。”