Claude Code 接国内模型：把账算清楚的那种方案

给已经在用 Claude Code、Cursor，但被模型区域策略卡住的人。全文只有配置、报错和账单，没有别的。

一个具体的下午

上周有个做外包的朋友把账单发我。三家客户的项目，同一个客户端，同一个模型，月底他只知道总共花了两千多，不知道哪家占大头。他说客户问起来他答不上来，只能按工时摊。

那天我们在他机器上折腾了两个小时，一半时间在改配置，另一半在翻日志找一次莫名其妙的超时。改完之后有个副产品：每次调用都留一条本地记录，按项目分开。他说这个东西比"能不能用"更值钱。

这篇写的就是那个下午之后我做的东西，以及它为什么长成这样。

卡点比想象中集中

我抓了掘金和 CSDN 上 198 条相关问题，按浏览量排：

问题	浏览
国内环境下 Claude Code 安装、配置完整指南	16,352
Cursor 限制国内用户使用 Claude 模型？解决方案	15,731
Cursor 国内用户无法使用模型（Model not available）	12,750
Claude Code Router + ModelScope 保姆级教程	11,827

几万人搜同一件事，说明问题不分散。翻评论区，抱怨基本落在三处：Base URL 填错（末尾斜杠、要不要 `/v1`、`http` 写成 `https`），模型名对不上（教程里的名字早就过期），客户端读不到 key（系统环境变量和进程环境变量是两回事）。

这三样我都栽过，后来写成了一张故障对照表，谁问就直接发。

选 MCP 不是因为时髦

一个数字：`@modelcontextprotocol/sdk` 周下载 89,463,331，比 `@anthropic-ai/sdk`（70,671,564）和 `openai`（58,843,658）都高。

这说明接入这件事已经标准化到协议层了。你在客户端里接一次 MCP，以后换模型不用动客户端；接某家的私有 SDK，换一家就得重写一遍。

对做工具的人来说，结论很简单：押协议，别押厂商。

它长什么样

一个 Python 文件，只用标准库，3.8 以上直接跑。

```
mcp-bridge
├── server.py          协议、路由、成本核算
├── config.sample.json 五家 provider 模板
├── install.ps1        Windows 一键装
├── install.sh         macOS / Linux / Git Bash
├── test_stdio.py      7 项协议自检
├── test_project.py    按项目分账测试
└── demo.py            生成一份真实会话记录
```

对外三个工具加一个账本：`bridge_chat` 发一次请求，`bridge_models` 看配了哪些模型，`bridge_cost` 算一次调用的钱，`bridge_usage` 按天、按模型、按项目看账。

几个被坑出来的决定

日志一律走 stderr。 stdin/stdout 留给协议。一行 `print()` 就能让客户端解析失败，这个坑我踩过，排查了两个小时。

stdio 强制 UTF-8。 Windows 默认 cp936，中文输出会变成坏字节，客户端直接抛 `UnicodeDecodeError`。

没配 key 就降级成 mock，不报错。 先让人看见工具列表，再去填 key。配置摩擦是这类工具流失最快的地方。

价格表放配置文件，不写进代码。 厂商调价是常态。

本机端点自动发现模型名。 猜名字是纯浪费，问一次 `/models` 就知道了，问不到再退回配置里的值。

配置文件用 `utf-8-sig` 读。 太多 Windows 编辑器存 JSON 带 BOM，这不该成为客户的第一个报错。

真实会话记录

```
# 客户端 -> 服务端
{"jsonrpc":"2.0","id":5,"method":"tools/call",
 "params":{"name":"bridge_chat",
           "arguments":{"provider":"local","prompt":"只回四个字：链路已通","project":"ACME-官网"}}}


# 服务端 -> 客户端
{"content":[{"type":"text","text":"{\"text\": \"链路已通\", \"model\": \"qwen2.5vl:3b\",
  \"latency_ms\": 63, \"cost_cny\": 0.0}"}]}
```

本机 3B 模型，纯 CPU，热启动 63 毫秒，冷启动 4.2 秒。

账怎么记

每次调用落一条 SQLite 记录：时间、provider、模型、输入输出 token、成本、延迟、是否 mock、错误原因、项目标签。token 优先取上游返回的 `usage`，拿不到才估算。

```
$ python server.py --report 7 --group-by project
最近 7 天调用：19 次，tokens 合计 351，花费 ¥0.0000
key             calls       tok_in      tok_out   cost_cny    avg_ms
(未标记)           17          208          105     0.0000       171
BETA-客服           1           11            8     0.0000         0
ACME-官网           1           11            8     0.0000         0
```

做外包的人最需要这块。没有它，月底只能看着总账单猜是哪个项目烧的钱。

没做到的部分

写技术文章最不缺把没做的说成做了，所以这节必须留：

项	状态
stdio 协议链路	7 项自检全过
真实上游调用	跑通过（本机 ollama）
按项目分账	跑通过
云端 provider	未验证。本机没有云厂商 key，只有 mock 和错误路径被测过
流式输出	没做，下一版

怎么用

```
powershell -NoProfile -ExecutionPolicy Bypass -File install.ps1
```

装完会打印一段可以直接粘进 `~/.claude.json` 或 `~/.cursor/mcp.json` 的配置。

开源版免费，自己折腾就行。要人帮你把环境配通、key 和客户端串起来，我做陪跑：¥499，三小时，跑不通不收费。更大的定制（把你们业务接口封成 MCP 工具、按项目出成本报表）另算。

一句话

这套东西解决的不是"能不能用 AI"，是用了之后账在哪、断了怎么办、数据经过谁。

· · ·

海桥 · 1024720334@qq.com