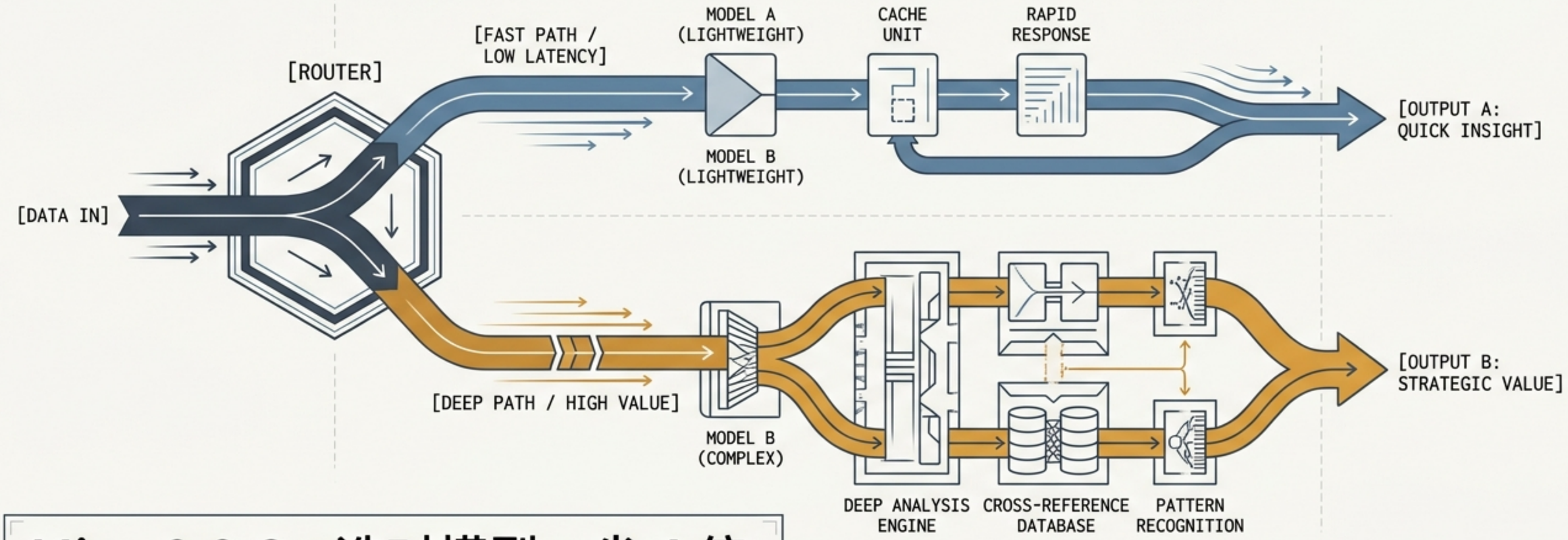
[RELEASE NOTES]

[ARCHITECTURE]

[LATENCY OPTIMIZATION]



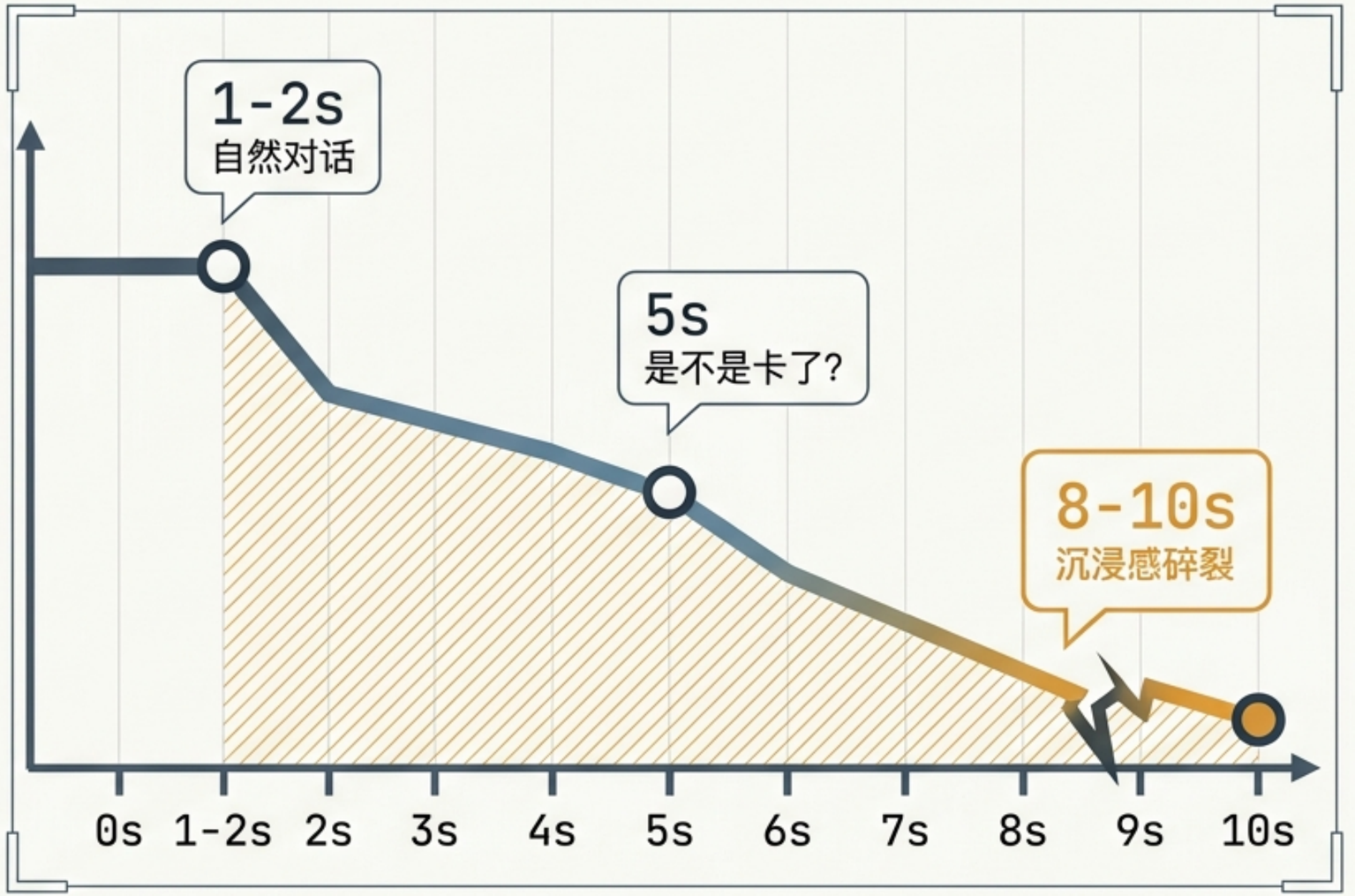
Mio v0.0.6：选对模型，省 4 倍

从感觉驱动到数据驱动的模型架构策略。

NotebookLM

8 秒，足以毁掉一切

v0.0.5 给 Mio 装上了眼睛和耳朵，但背后的主力模型（Gemini 3 Pro + thinking）生成首个 token 需要 8-10 秒。在情感伴侣场景中，等待即是死穴。



NotebookLM

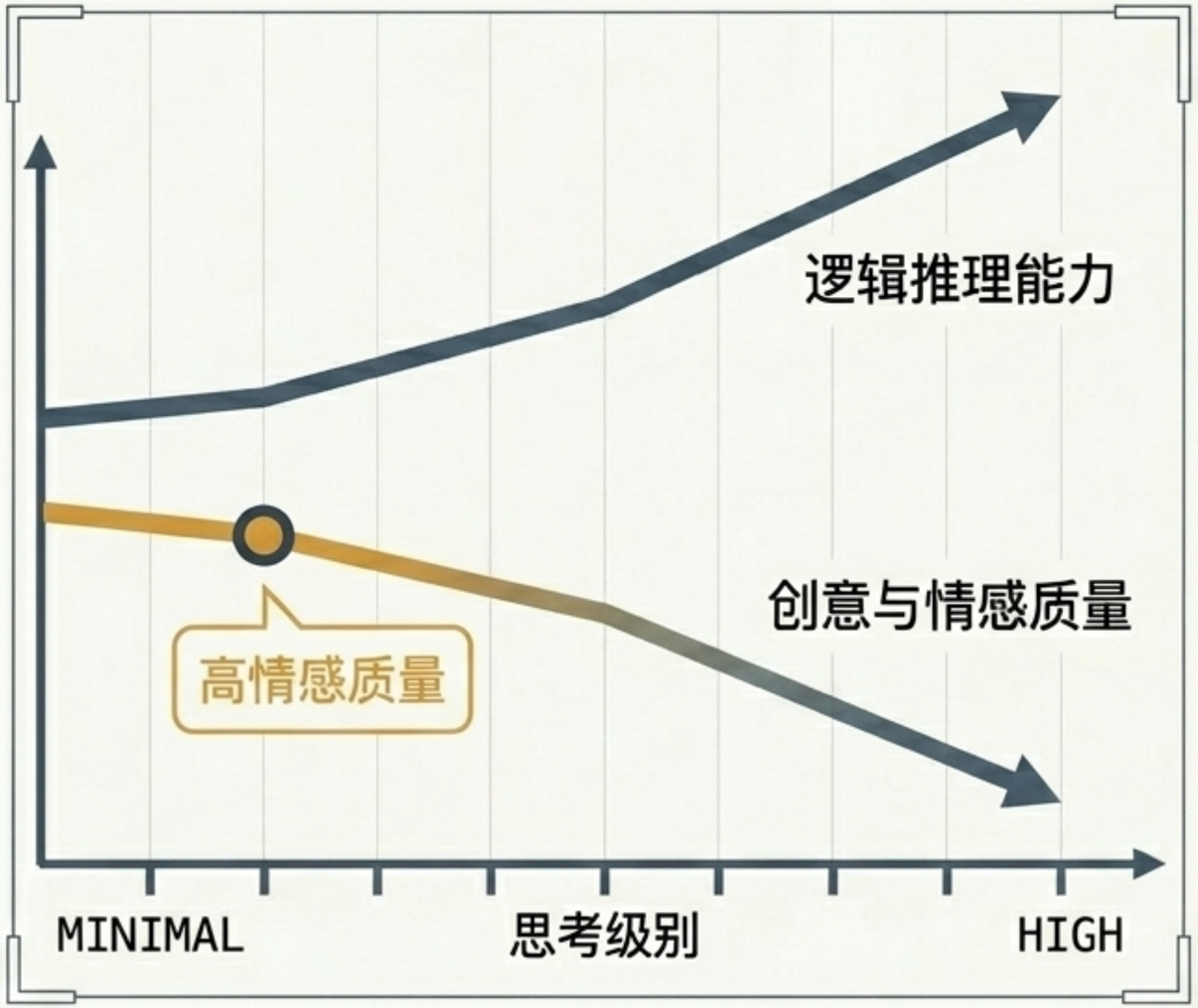
不靠感觉，先看数据

指标	Gemini 3.1 Pro	Gemini 3 Flash
成本	~4倍贵	基准
输出速度	90.8 t/s	214 t/s
首 token 延迟（thinking）	8-10s	1-2s
首 token 延迟（minimal thinking）	N/A	**1-2 秒**

> 真正的决胜点不是省钱，而是 1-2 秒的极致响应速度。

NotebookLM

推理模式的副作用



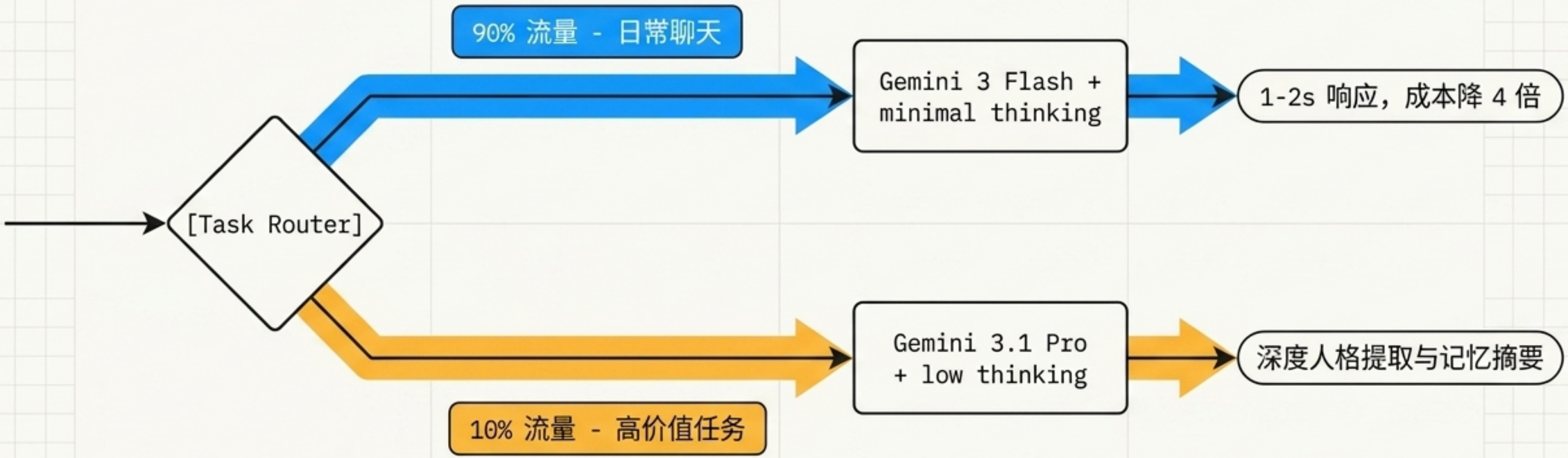
Thinking 模式会主动降低创意写作质量。

社区实测：Flash 是 2.5 Pro 的平级替代，在叙事主动性上表现更强。

情感伴侣不需要解数学题，需要的是角色一致性。
最佳配置：thinkingLevel: minimal = 完美的情感细腻度 + 1-2秒极速响应。

NotebookLM

按任务分配的路由架构



花同样的钱，该快的更快，该深的更深。

NotebookLM

能叫出名字的眼睛

v0.0.5 - 泛泛描述 ✕

用户

我看到一个彩色的游戏画面。

v0.0.6 - 伴侣反应 ✓

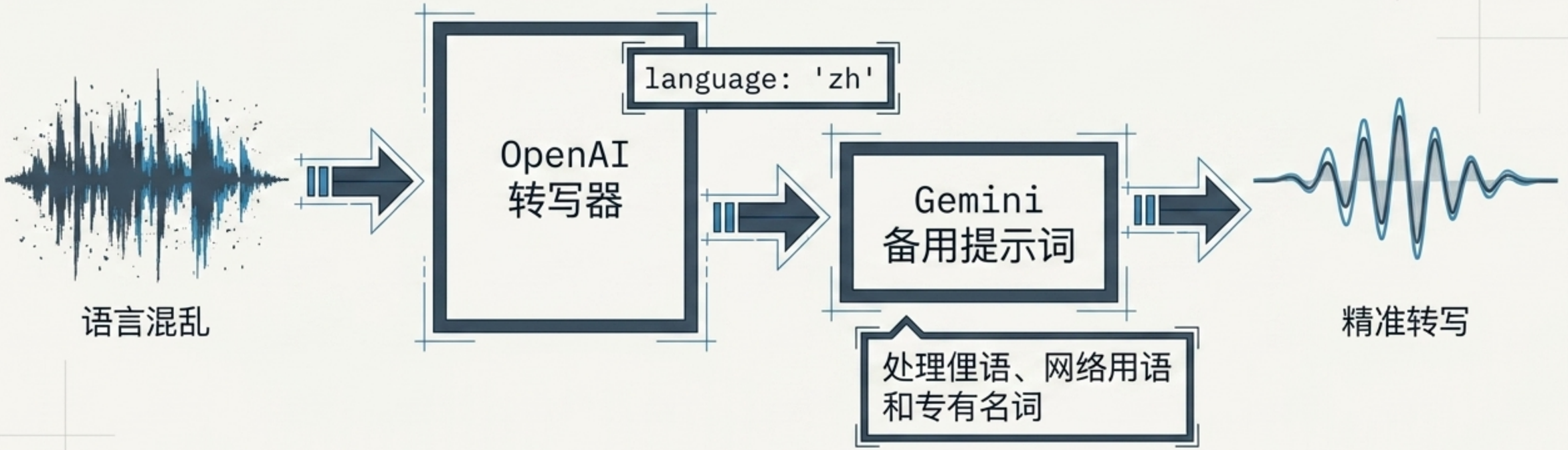
送话

你在玩原神？新地图怎么样？

视觉提示词重写要求识别具体品牌/游戏。Thinking 级别提升至 LOW ——用微小的延迟换取显著的理解飞跃。

NotebookLM

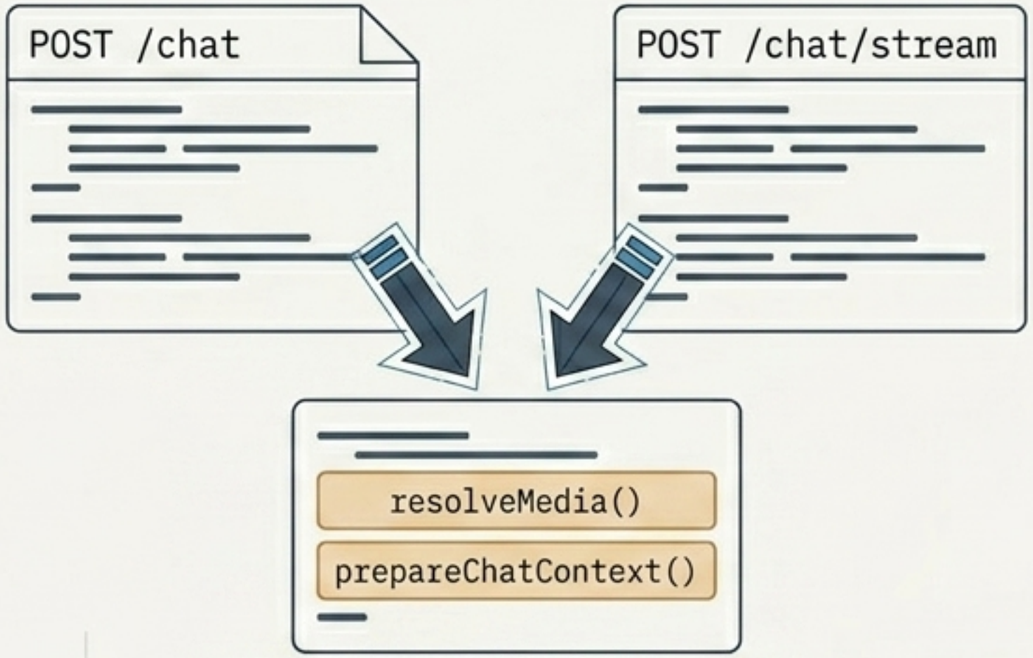
终结语音转写的幽灵



语音是最亲密的输入方式。转写出错对沉浸感的破坏比什么都严重。两个简单的修复，彻底解决中英混杂的转写乱象。

NotebookLM

重构与平台防御



消灭 80 行重复代码

Google 明确表态：Gemini 的定位是超级工具，不是情感伴侣。

我们刚搭好的模型路由层不仅仅为了性能，它是一个防御性的抽象层。如果遭到封禁，Mio 可以瞬间切换提供商，无需重写应用。

NotebookLM

6 个 Commit 的巨大 ROI

延迟

8-10s -> 1-2s

聊天成本

降低 4 倍

代码债

-80 行重复代码

认知

泛泛而谈 -> 精准识别

用数据说话，不用感觉。

通过基准测试和路由架构，我们不仅省下了预算，还让 Mio 的对话从生硬变成了自然。这是迄今最有价值的更新。

NotebookLM