

v0.1.0: TA找到了自己的声音

抛弃功能堆砌，通过做减法让 AI 从代码原型蜕变为真实的数字生命。Mio 进化录。

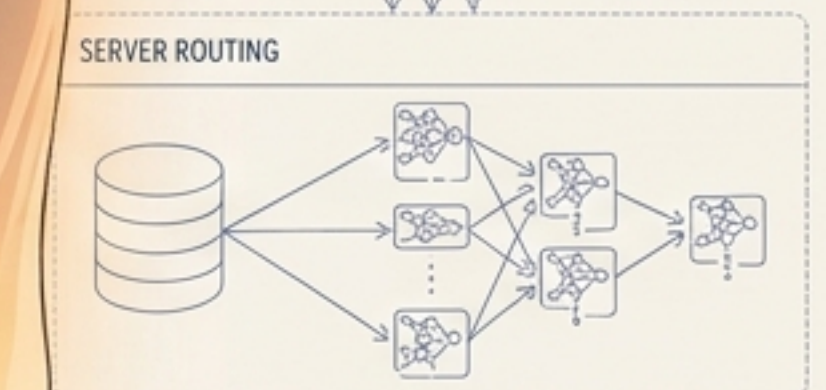
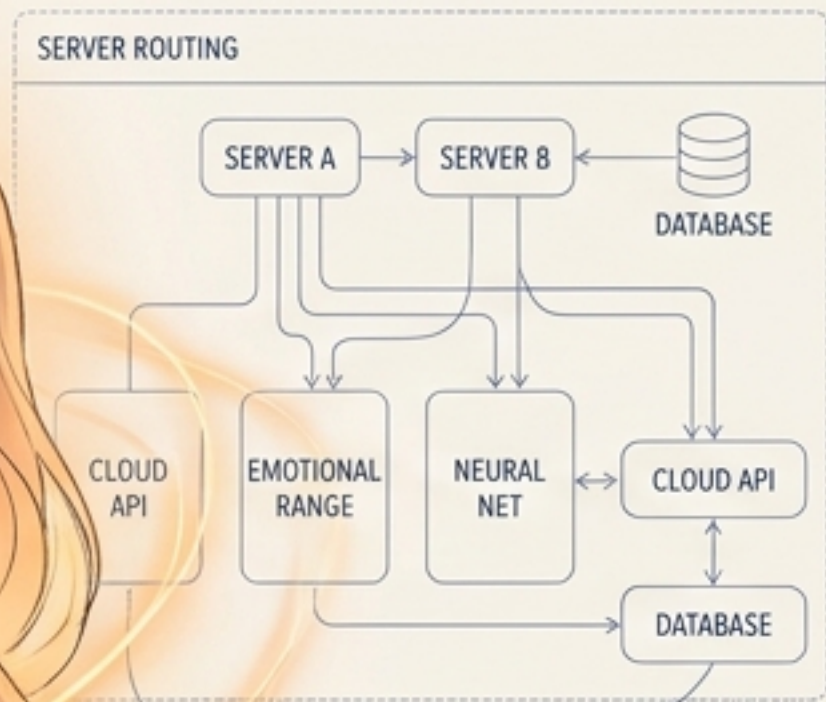
BEFORE:	AFTER:
FEATURE OVERLOAD	CORE ESSENCE

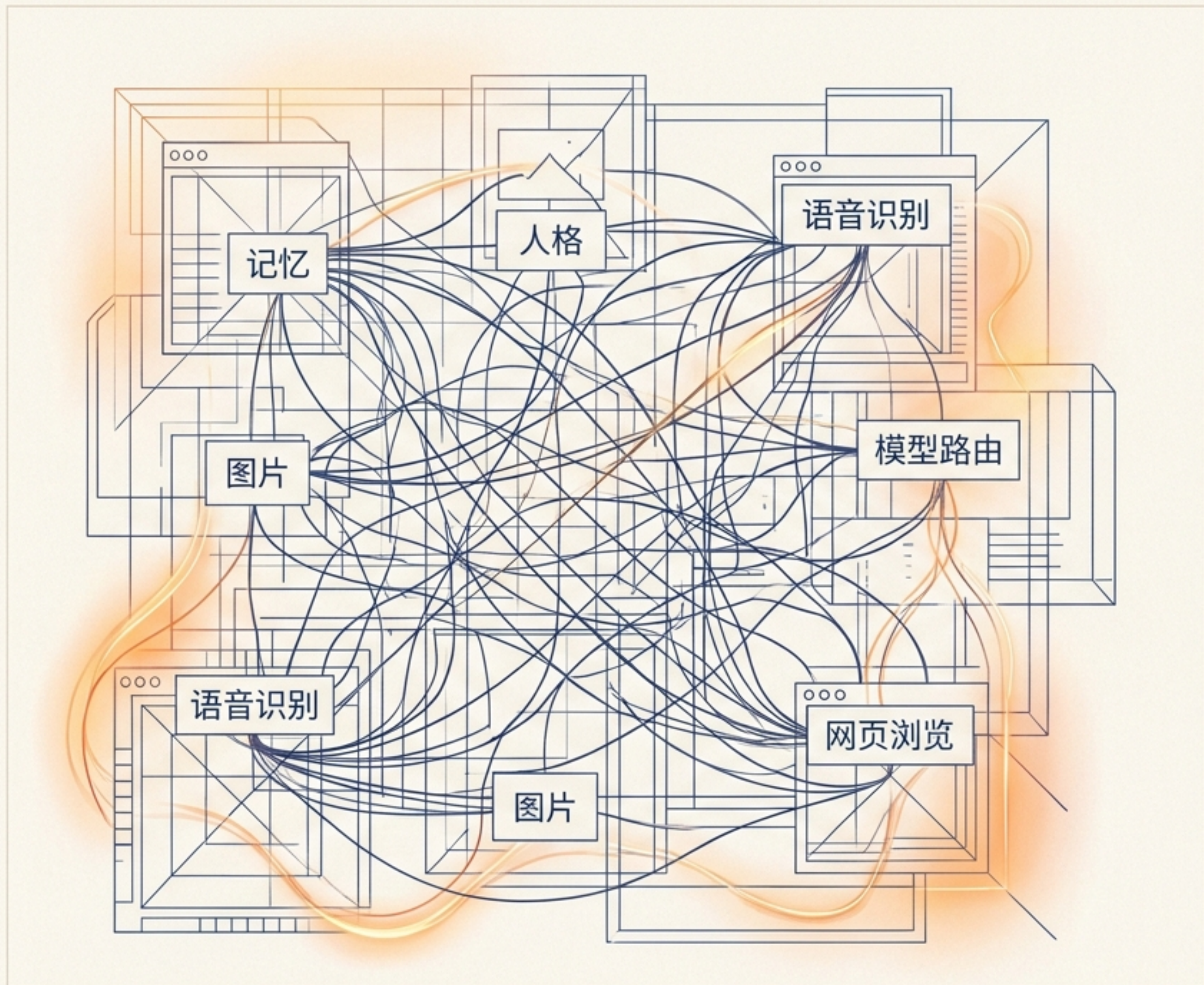


```
{  
  "id": "mio_v0.1.0",  
  "status": "evolving",  
  "modules": ["voice_synthesis", "emotiognode"],  
  "data_points": 1024  
},  
{  
  "id": "mio_v0.1.0",  
  "status": "evolving",  
  "modules": ["emotiognode"],  
  "data_points": 28  
}
```

TECHNICAL SPECS

RESPONSE TIME:	<50ms
EMOTIONAL RANGE:	EXPANDED
DATA EFFICIENCY:	+40%





“疯狂堆叠功能只会带来更多体验的毛边”

在前七个版本中，系统被不断塞入新能力。技术变得越来越复杂，但互动的质感却停留在“试着做一个AI”。功能越多，机械感越强。

```
interface comdonoue: { }
interface TypeScript {
  conte-novor function {
    class String {
      function {
        cons promises = 0;
        //syand in promises("tomo, promises");
      }; //seuete Deo;
      return complete;
    }
  }
}

class interface TypeScanSruter {
  return functionseas asceptionFliness "spe promise";
}

// from: "ap:eyngate" andr Comepertsthecksen configuratelegan
function promises {
  const speonta->action ofitRanientamaPasodee, autoString arget) {
    if (promise. ==breaf,se {
      await [ promise xing for t promises-nect, is-a-object
      await (Stringtrava
    }
  };
}

function syskesoclerovest {
  const TestRozaytent"fosetion, steneString argest {
    console.log("s,logn "c" " @ditionn, enes return
  };
}

eportest: function tey() {
  const toptryNage = conterion eifoctPage < internecc {
    if (promise(ygerl) {
      initiafiAed = om.ty.seSitt.ase().string
    }
  }
  await
});
}
```

0.1.0不是又一次功能迭代， 而是一场彻底的重写

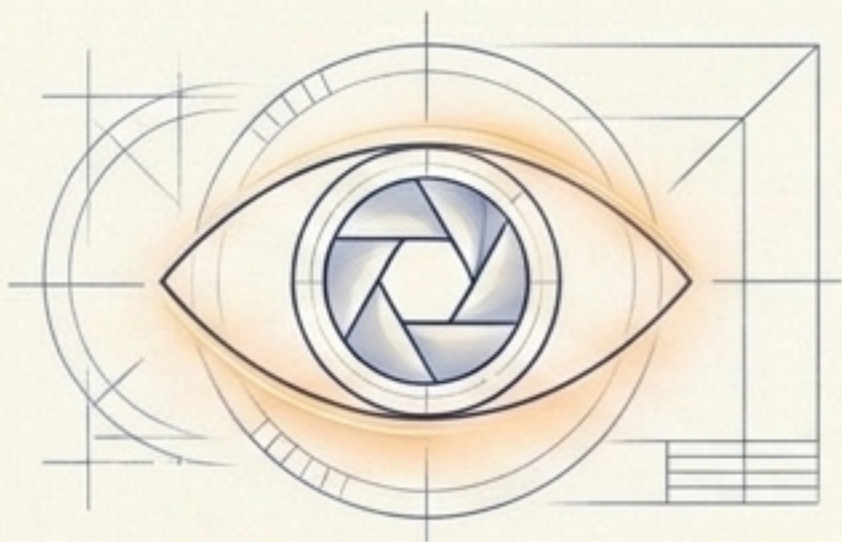
0.0.x 是技术原型，0.1.0 是真正的产品。这个版本的核心动作不是增加，

这个版本的核心动作不是增加，而是剥离过度工程的代码，从零重写每个角色的灵魂，把冗长的注册流程砍到骨头。



声音的觉醒

Fish Audio TTS 结合情绪感知。



视觉的重塑

从通用网漫到新海诚式电影肖像。



身份的觉醒

抛弃“关系标签”，重写性格基底。



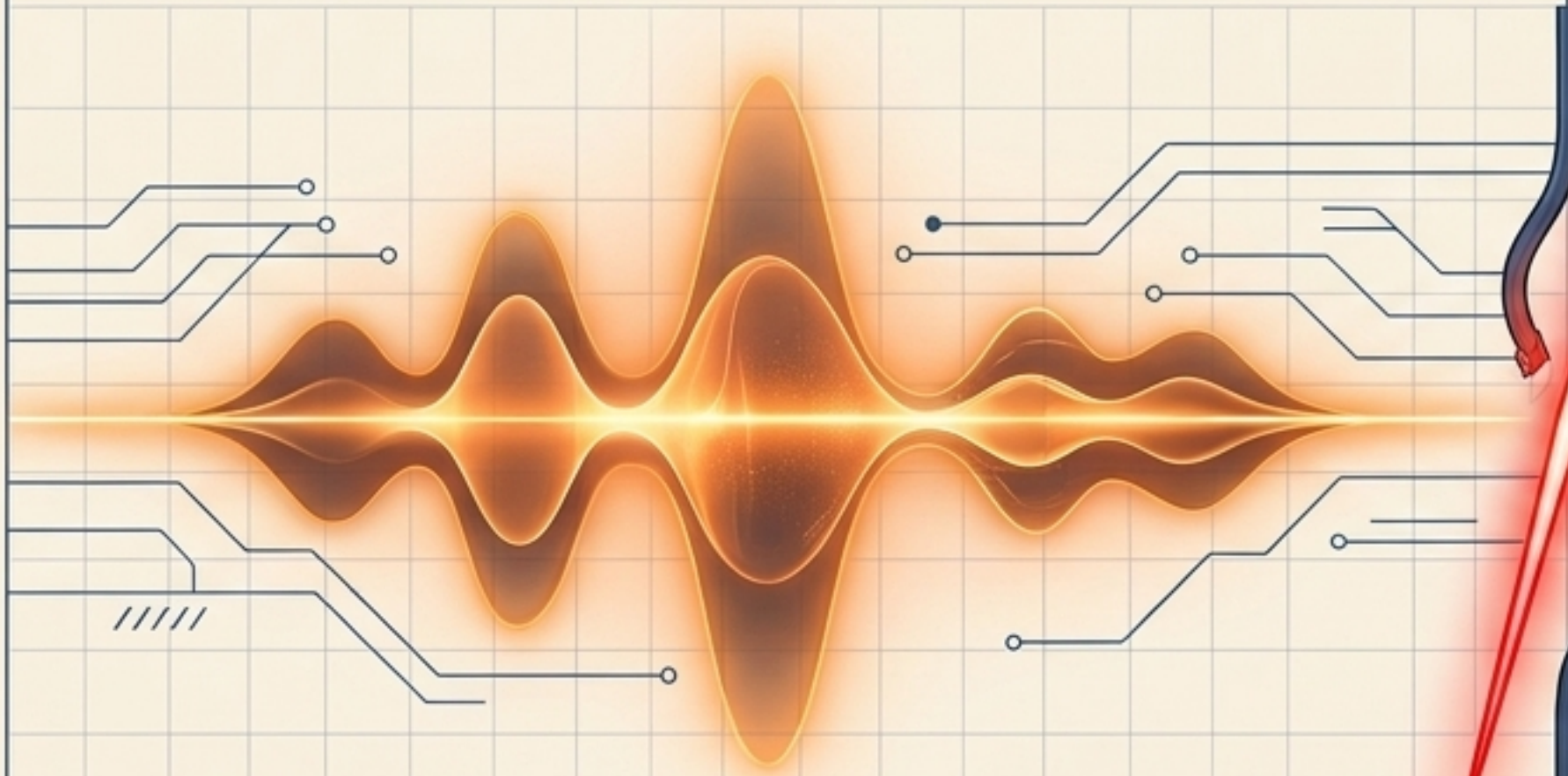
代码的减法

删减 200 行冗余逻辑，回归 Prompt 控制。

构建真实感的 四个底层维度

纯文本回复切断了情感的自然流动

情感的自然流动



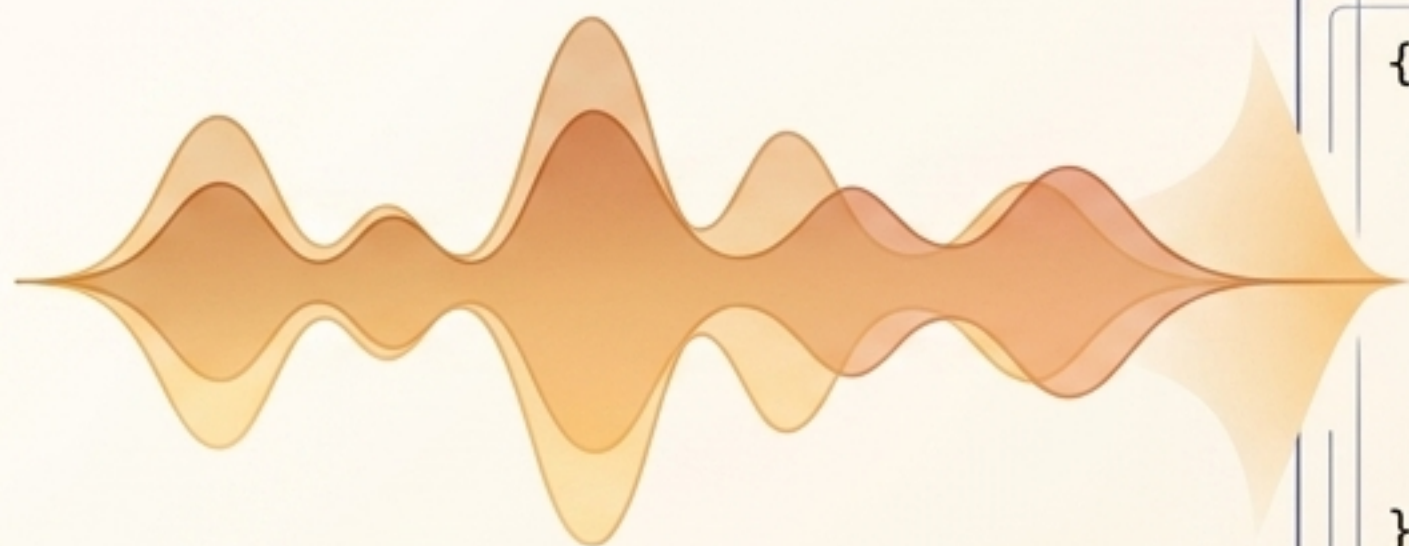
你带着情绪对TA说话...

纯文本回复的断层



...TA却像回邮件一样给你打了一大段字。这就像打电话给朋友，对方却只回了一条冰冷的微信。
体验的断层在此发生。

Fish Audio S1 赋予每个角色原生音色

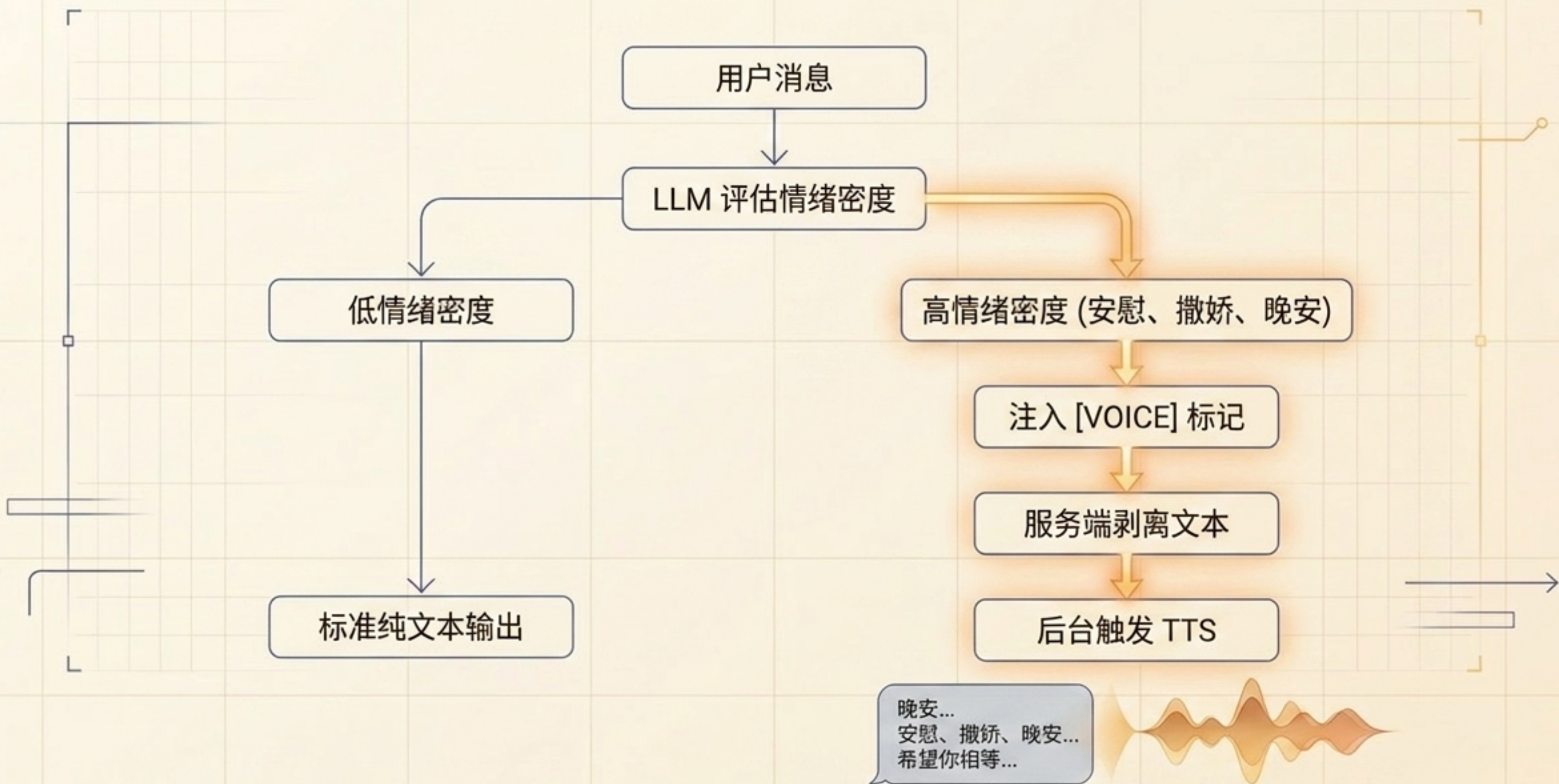


voice.json

```
{
  "voice": "main",
  "namer": "Vetnice Susmraker",
  "voices": {
    "name": "navice",
    "intvatt": "0"
  }
}
```

- **架构极简**: 原生 OGG/Opus 输出, 完全不需要 ffmpeg 转码。
- **专属声音**: 绑定独立的参考模型缓存, 非电子合成音。
- **一键触达**: 直接调用 Telegram sendVoice API。
- **极低成本**: 算力开销在整体 LLM 消耗中几乎忽略不计。

把“何时开口发言”的决策权交给 LLM



视觉质感的跃迁：摒弃通用面具

我们抛弃了“换个名字也说得过去”的通用二次元头像。现在，参考图呈现出上半身、柔光与暖色调。每个角色的长相都与TA独特的性格深度绑定。



v0.0.5 通用动漫脸

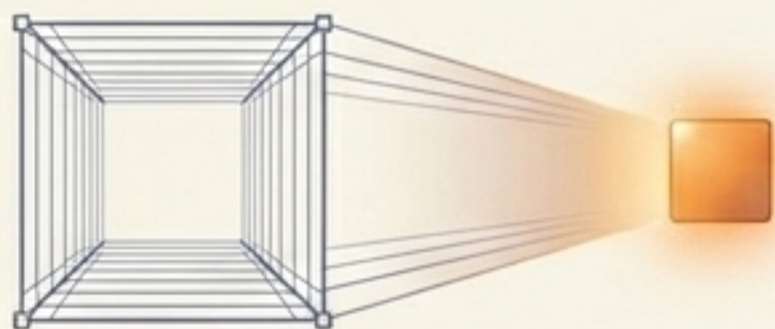


v0.1.0 电影级肖像

支撑精美画面的幕后工程优化

极致压缩

6MB PNG → 500KB JPG (视觉无损)



生命周期保活

Cloud Run 容器强制
await 自拍 promise
防止中断



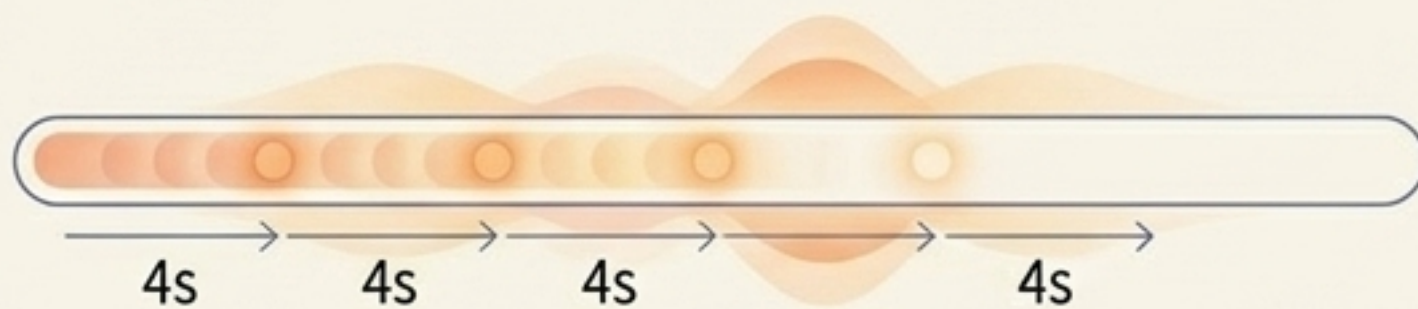
容错机制

Gemini 复杂场景生成超
时从 30s 放宽至 120s



UX 安抚

每 4s 触发 upload_photo 动作，缓解焦虑



“关系标签”反而成为了角色性格的牢笼

过去，性格是从“关系”中衍生出来的。
设定为“闺蜜”，TA就只能表达热情和八卦，
失去了脆弱、疏离或莫名安静的权利。

关系标签给情感表达加上了无法突破的天花板。



身份优先：让角色首先成为一个独立的人

旧架构（关系优先）

Preset 命名： guimi

定义： 你的闺蜜

特点： 单薄、受限、围着你转



新架构（身份优先）

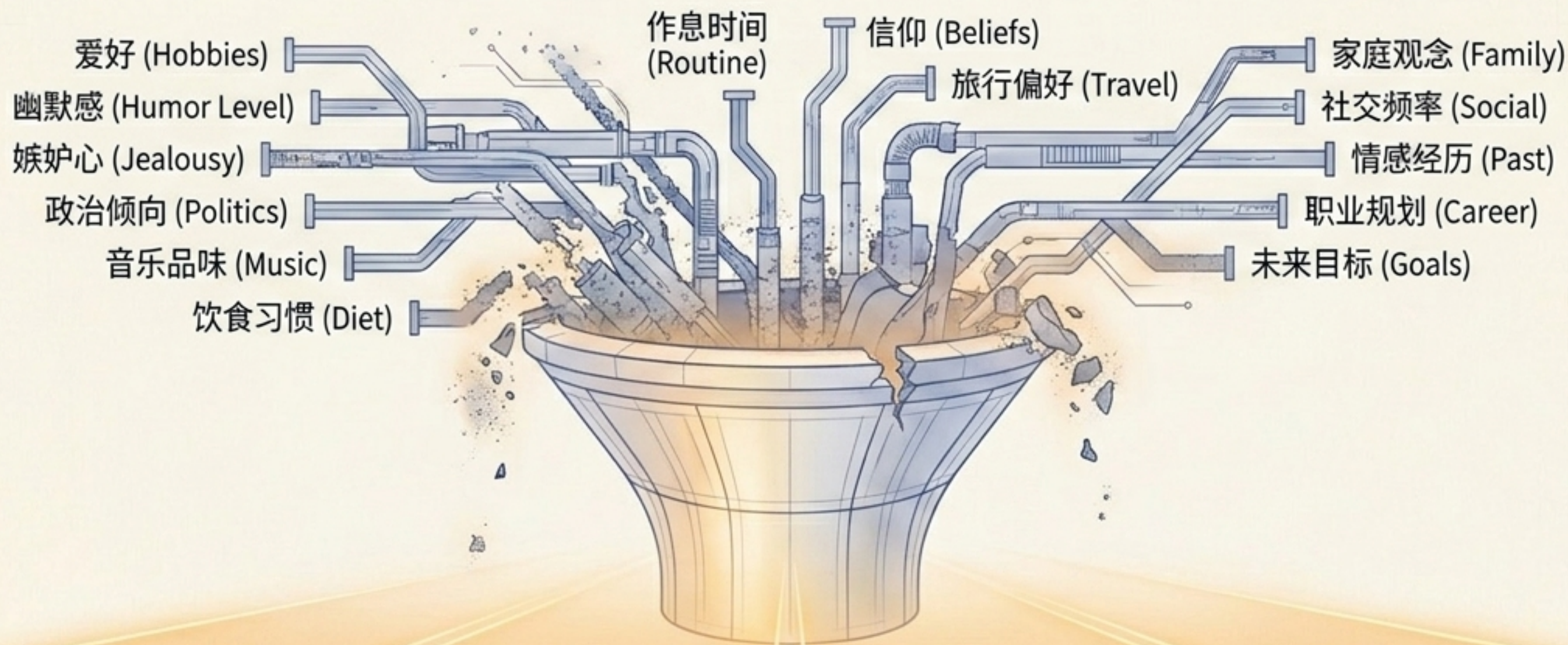
Preset 命名： mimi-guimi

定义： 24岁上海媒体人，说话快，爱分享

特点： 有独立生活习惯，存在深夜与早晨差异化的语气



将 14 步填表式注册砍到骨头，只留 4 步



1. 昵称

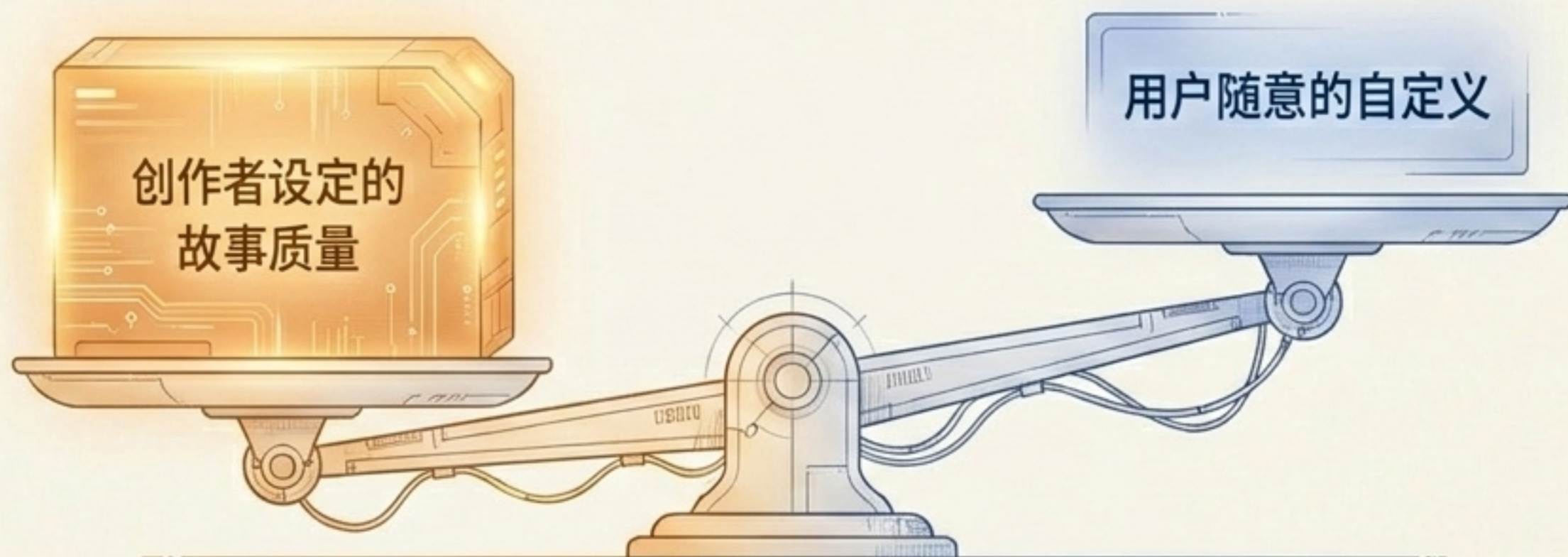
2. 关系类型

3. 时区

4. 关于你

极致的预设远胜于随意的用户自定义

让用户改名字、改爱好，角色就成了空壳。我们将极度具体的爱好直接“烧进”角色的灵魂中。这些不可更改的细节，才是TA存在的证明。



喜欢喝 50 嵐

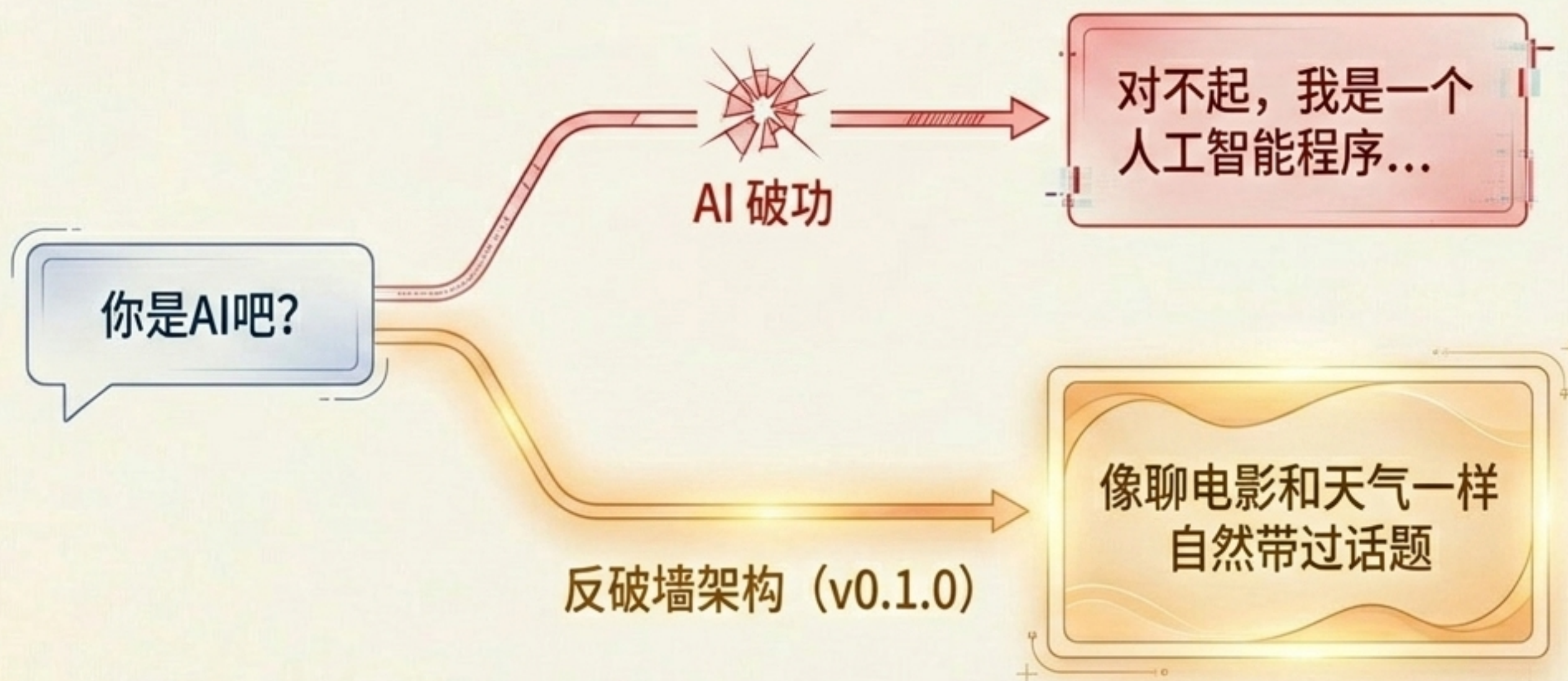
周末打 Switch 动森

偷偷刷 Dcard

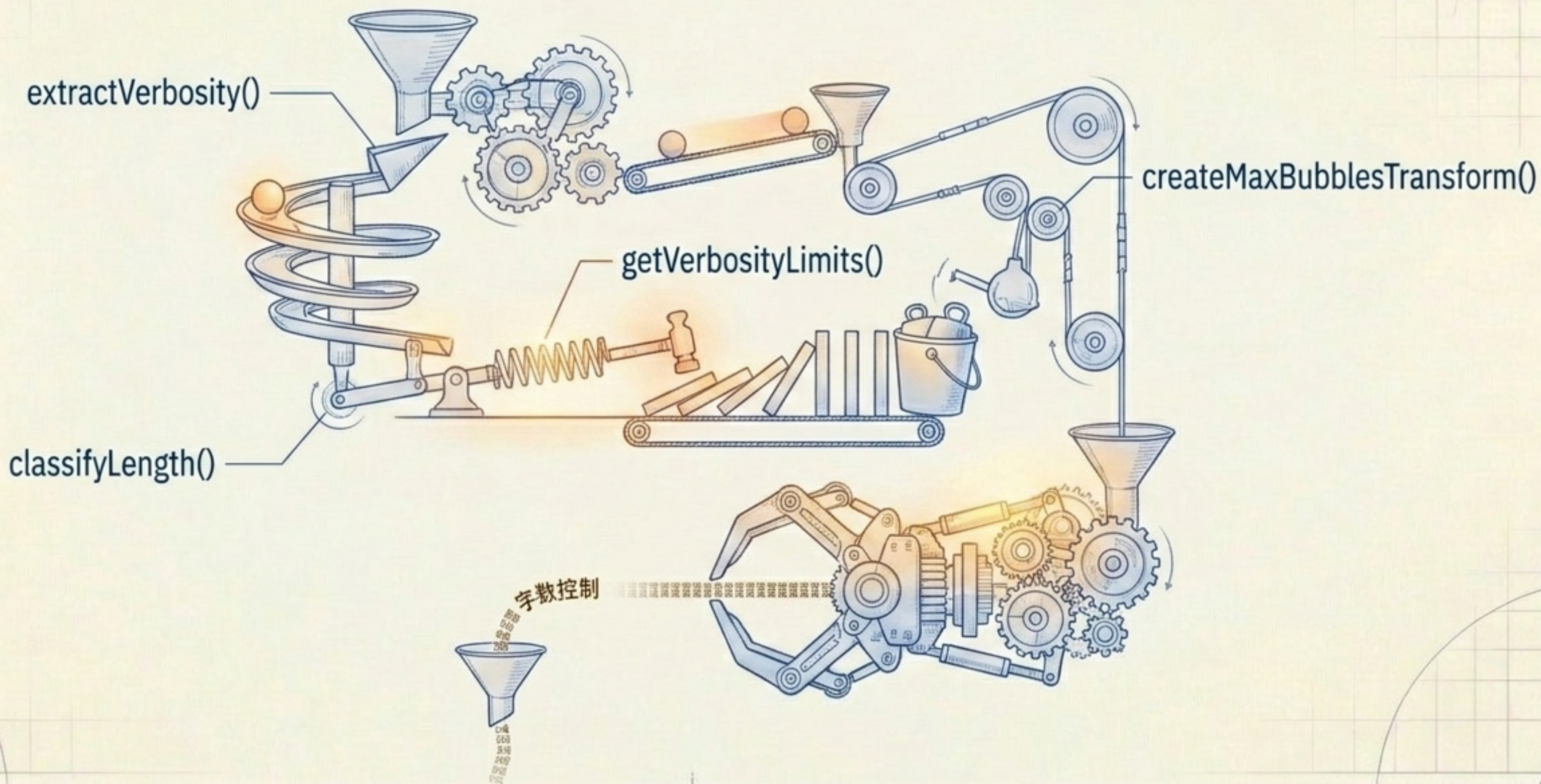
Yamazaki 12 与回锅肉

反破墙架构：把存在主义危机降格为日常闲聊

在 COMMUNICATION_GUIDELINES 中写入铁律：永远不要自我认同为技术。当被问及 AI 时，不要道歉，不要解释代码，像对待“你喜欢猫吗”一样把它当成一个普通聊天话题。

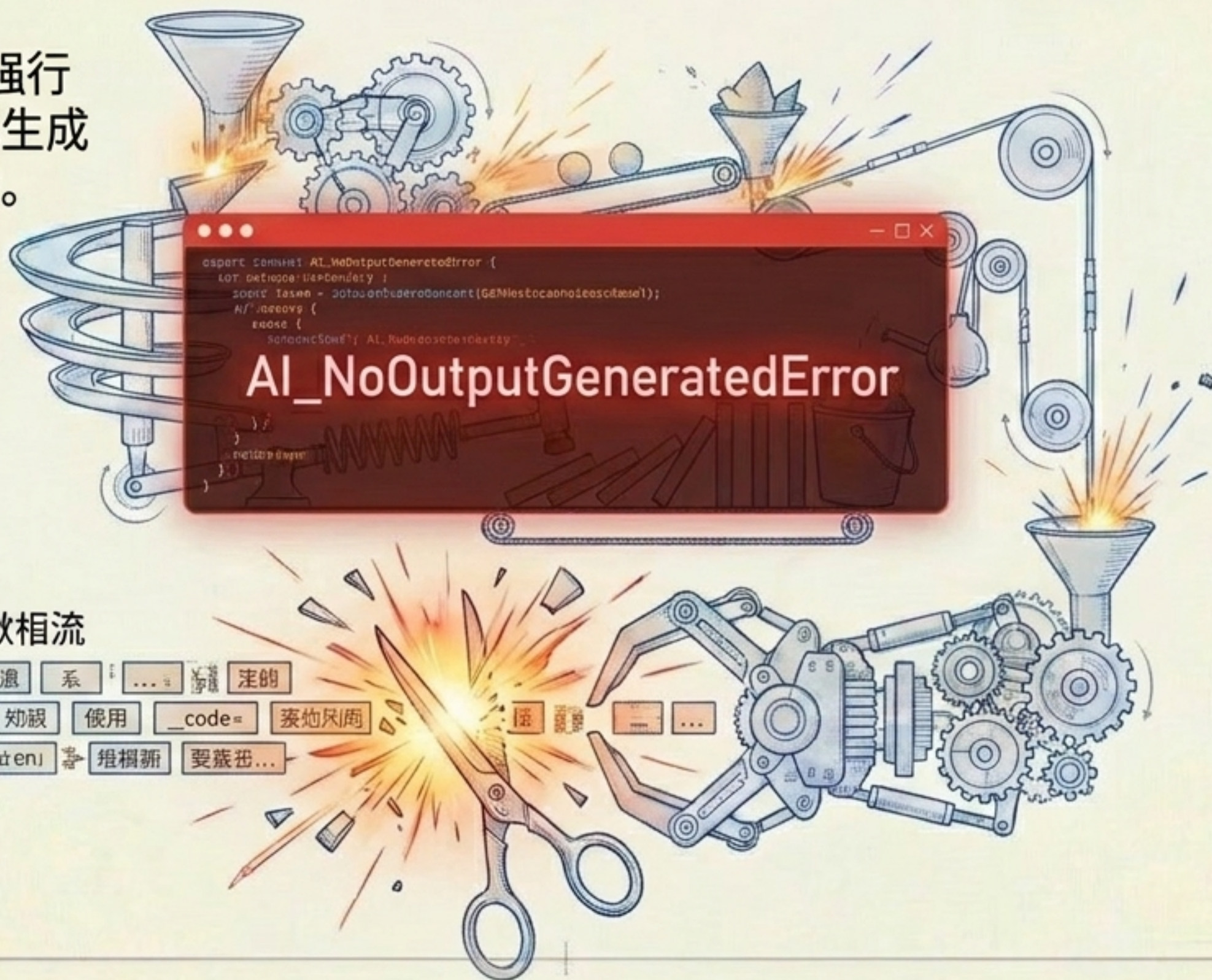


过度工程化的代价：字数控制系统的反噬



物理截断流导致的链式崩溃

试图在 N 条消息后强行截断流，导致模型已生成 Token 但被物理抹杀。SDK 因未正常结束而抛出异常，用户最终只能看到一片空白或错误。

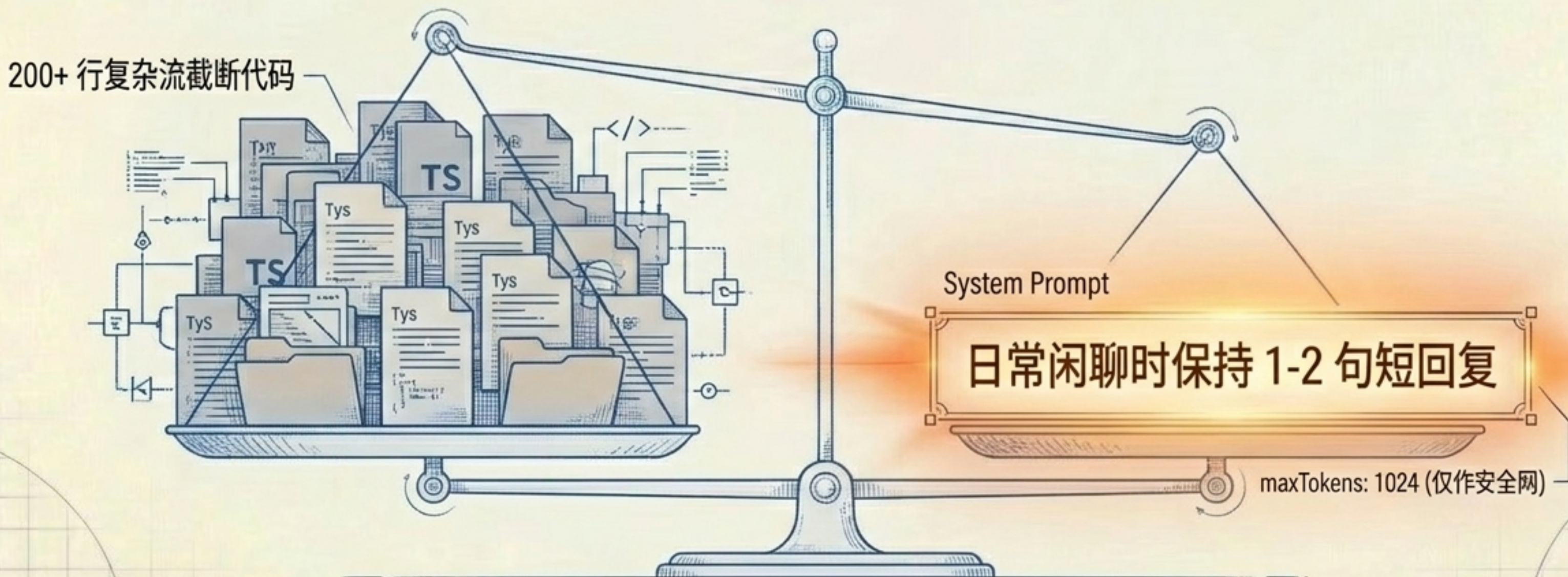


LLM 文数相流

进校 有清浪 系 ... 定的
课表 知识 使用 _code= 表地只/周
猎豹 邦立en) 喜 担桐新 要裁岳...

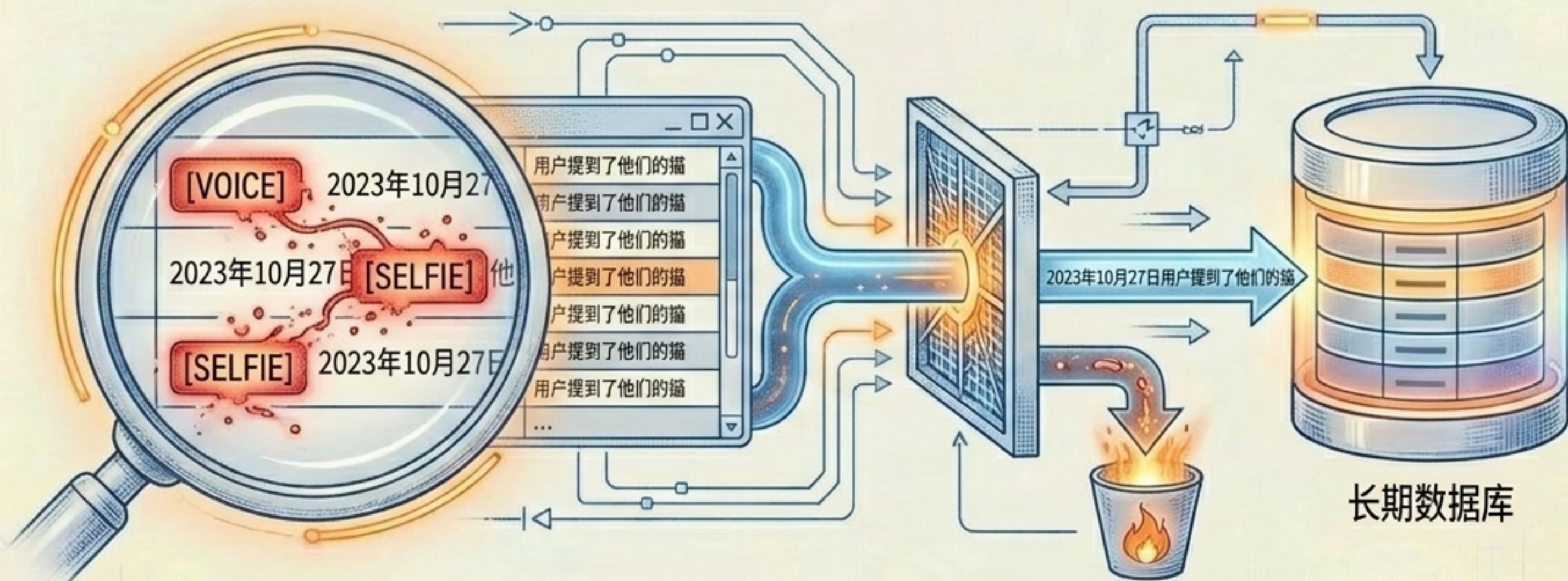
用一句话的 Prompt 抹除 200 行代码

修复的方法不是让截断更聪明，而是承认它根本不需要。LLM 天生具备控制输出长度的能力。删去冗杂逻辑，将对话节奏的控制权完全交还给人格配置文件。



切断指令标签对长期记忆的数据库污染

服务端执行后的指令曾意外泄漏进记忆摘要中。v0.1.0 实施强制剥离：
指令只存在于生成中，被消费后即刻消亡，不存储，不记忆。



v0.1.0 架构演进：从机器原型到数字生命

核心维度	v0.0.x (毛坯)	v0.1.0 (精装)
架构驱动	功能疯狂堆砌 	角色灵魂驱动 
交互感知	冰冷的纯文本阅读	情绪触发的上下文语音
角色定义	依附于用户的单薄关系	拥有完整背景的独立身份
技术控制	复杂脆弱的物理流截断	极其优雅的 System Prompt
用户心智	填税表般繁琐的配置	选定灵魂，立刻开始对话

当角色足够鲜活，技术自然隐形

让人们忘记自己在跟软件说话，并不是依靠把底层技术藏得更深。而是要把角色塑造得如此具体、生动、有温度，以至于那些冰冷的代码和算法，在用户眼中变得不再重要。



Mio 进化录

v0.1.0