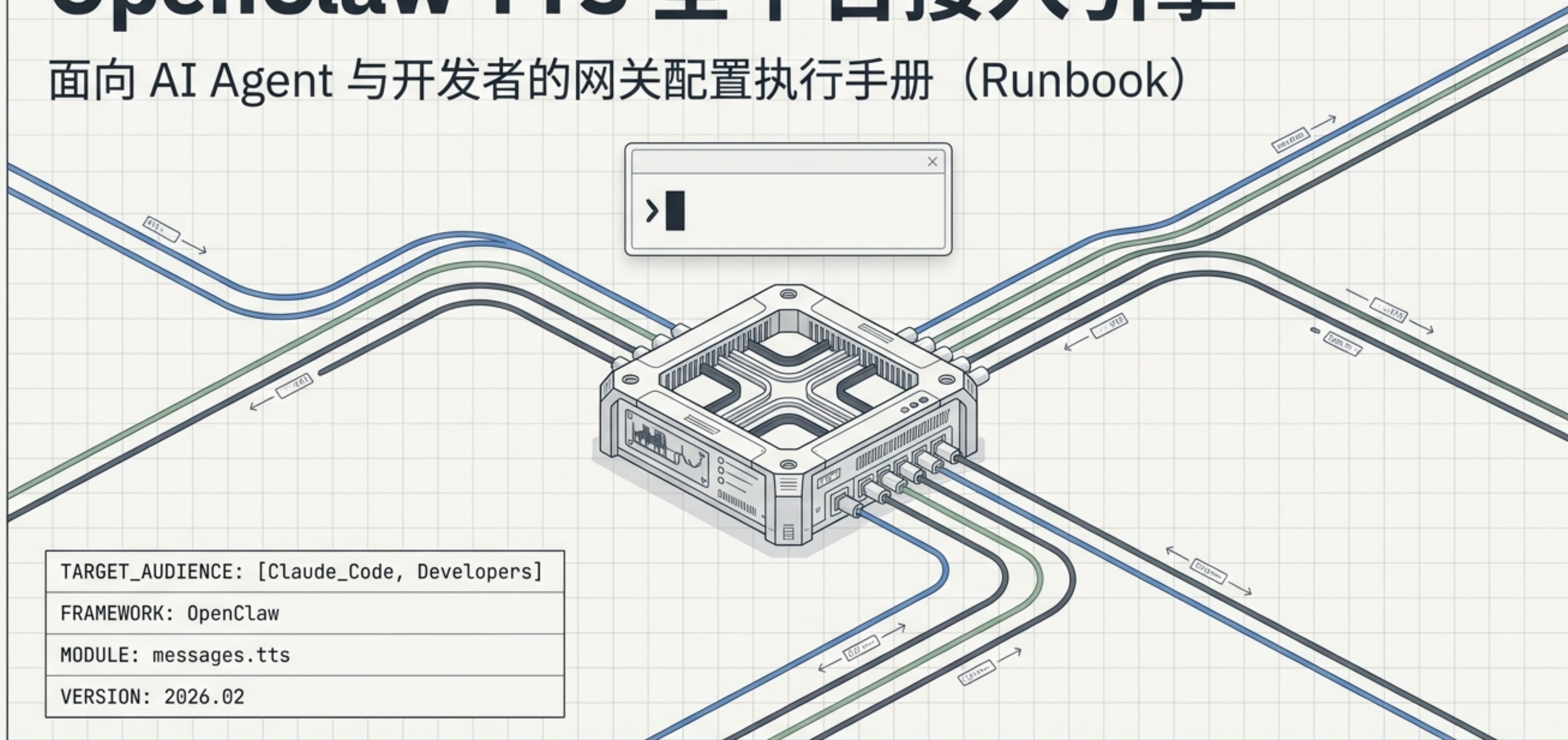


OpenClaw TTS 全平台接入引擎

面向 AI Agent 与开发者的网关配置执行手册 (Runbook)



TARGET_AUDIENCE: [Claude_Code, Developers]

FRAMEWORK: OpenClaw

MODULE: messages.tts

VERSION: 2026.02

系统级准入自检



OpenClaw 核心框架已部署
并处于运行状态



具有 openclaw.json 网关配
置文件的读写权限

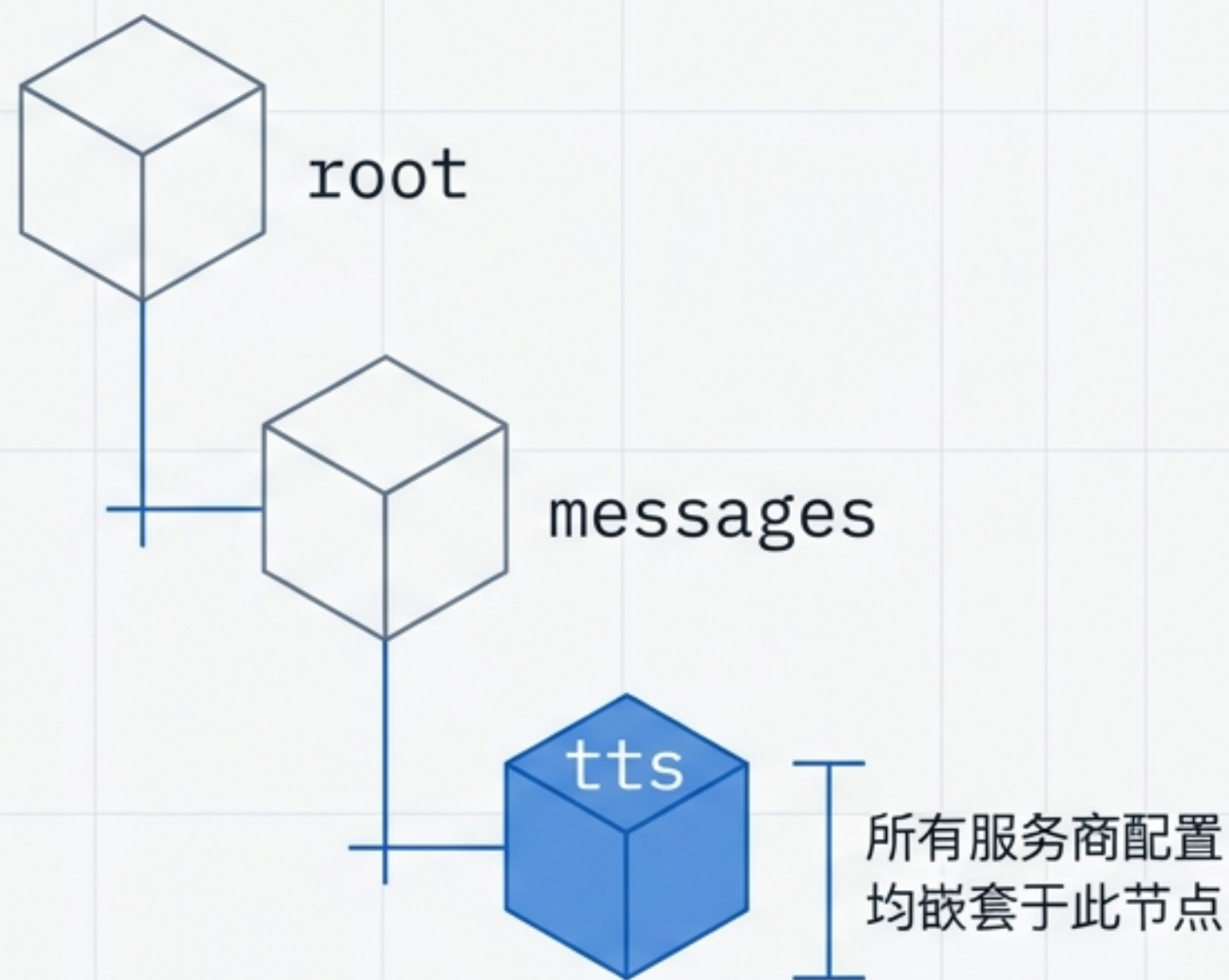


具备热加载或服务重启能力

```
kill -USR1 <pid>
```

触发配置文件热重载

架构拓扑寻址



TTS 引擎服务商能力矩阵

服务商	输出格式	Telegram 气泡支持	情感控制能力	接入成本/门槛
Fish Audio	OGG/Opus	完美兼容 (原生)	⚠️ 不支持 (方括号会外露)	需 API Key
火山引擎 v2	MP3 拼接	支持 (需开启 voiceCompatible 伪装)	✅ 逐句 LLM 驱动	需 App ID & Token
ElevenLabs	Opus / MP3	完美兼容	极高音质与多维 参数调节	需 API Key
OpenAI TTS	Opus / MP3	完美兼容	标准	需 API Key
Edge TTS	MP3 (兜底机制)	不支持原生 Opus	基础	✅ 免密钥 (调用公共网络)

Fish Audio 配置：极致原生数据流



核心字段配置

apiKey: FISH_API_KEY

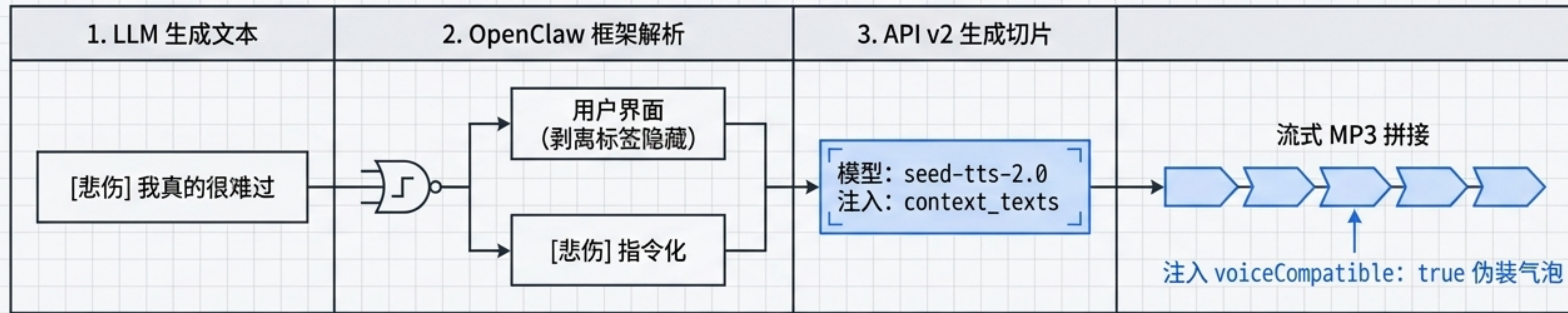
referenceId: [留空则使用默认音色]



严格禁止情感标记。

Fish Audio 引擎无法解析指令，LLM 生成的 [方括号] 情感标签将被作为纯文本直接朗读。必须在 Prompt 中关闭情感标记输出。

火山引擎 v2: LLM 驱动的情感拆解与合成



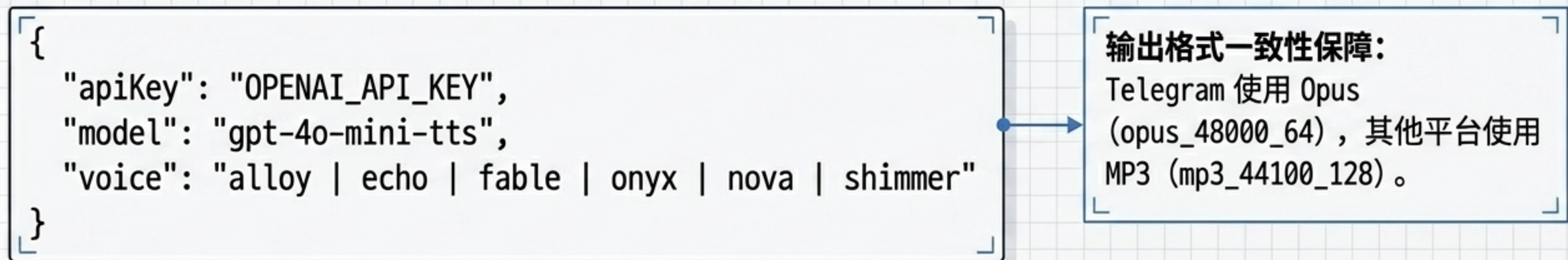
架构演进: v1 vs v2 关键差异

特性	v1 配置	v2 配置 (推荐)
模型引擎	seed-tts-1.0	volc.seedicl.default
情感控制	无	逐句 context_texts 控制
Telegram表现	普通音频文件	原生语音气泡 (通过伪装)
必要参数	默认	version: 'v2', speaker (克隆音色带 S_ 前缀)

ElevenLabs: 工业级声学调音台



OpenAI: 极简标准服务



Edge TTS: 零配置兜底防线



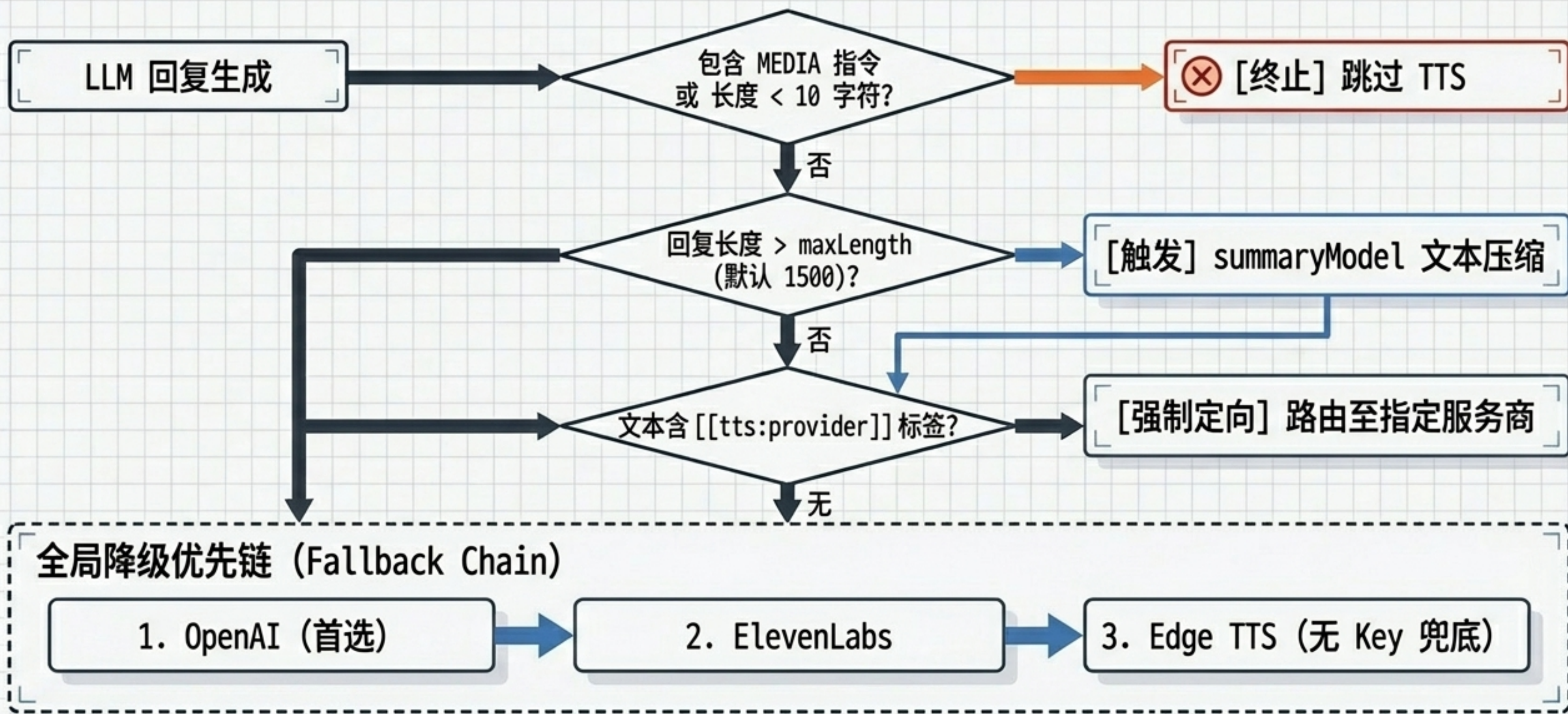
核心激活配置

```
enabled: true  
voice: en-US-MichelleNeural  
outputFormat: audio-24khz-48kbitrate-mono-mp3
```

系统局限与自愈机制

- **公共网络依赖:** 依赖微软公共网络接口, 无 SLA 服务等级协议保证, 实行尽力而为原则。
- **格式自愈重试:** 当配置的 outputFormat 不受支持而报错时, 网关将自动降级并重试标准的 MP3 格式。
- **Telegram 限制:** 无法原生输出 Opus 格式供 Telegram 语音气泡使用, 仅能以音频文件形式发送。

智能网关路由与降级拓扑



会话级控制台：操作速查表 (Cheat Sheet)

注：Discord 环境请将
/tts 替换为 /voice

状态与会话开关 (Session Toggles)

`/tts status` :
显示当前服务商及核心配置

`/tts off` :
关闭当前会话自动 TTS

`/tts always` :
强制所有回复生成语音

`/tts inbound` :
仅当收到语音时触发

`/tts tagged` :
仅识别 `[[tts]]` 标签发声

参数覆写 (Local Overrides)

`/tts provider [name]` :
热切换服务商 (openai,
elevenlabs, edge, volcano,
fishaudio)

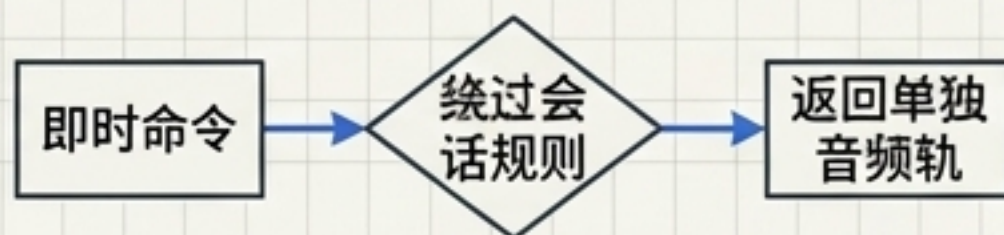
`/tts limit 2000` :
修改当前会话摘要触发阈值

`/tts summary off` :
禁用长文本自动摘要

→ 覆写操作将写入本地偏好文件，绕
过主 JSON 设定的全局规则。

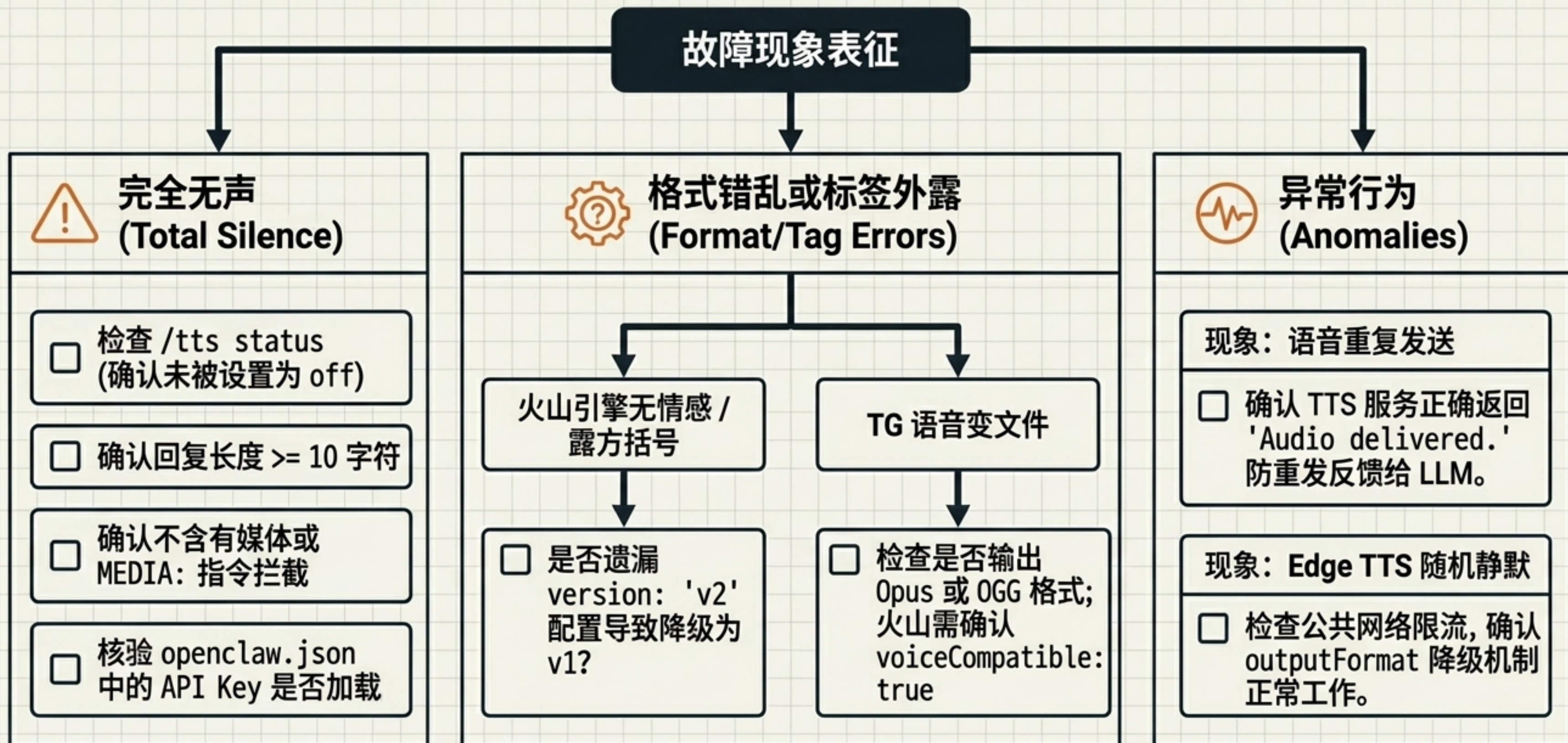
即时执行 (Instant Action)

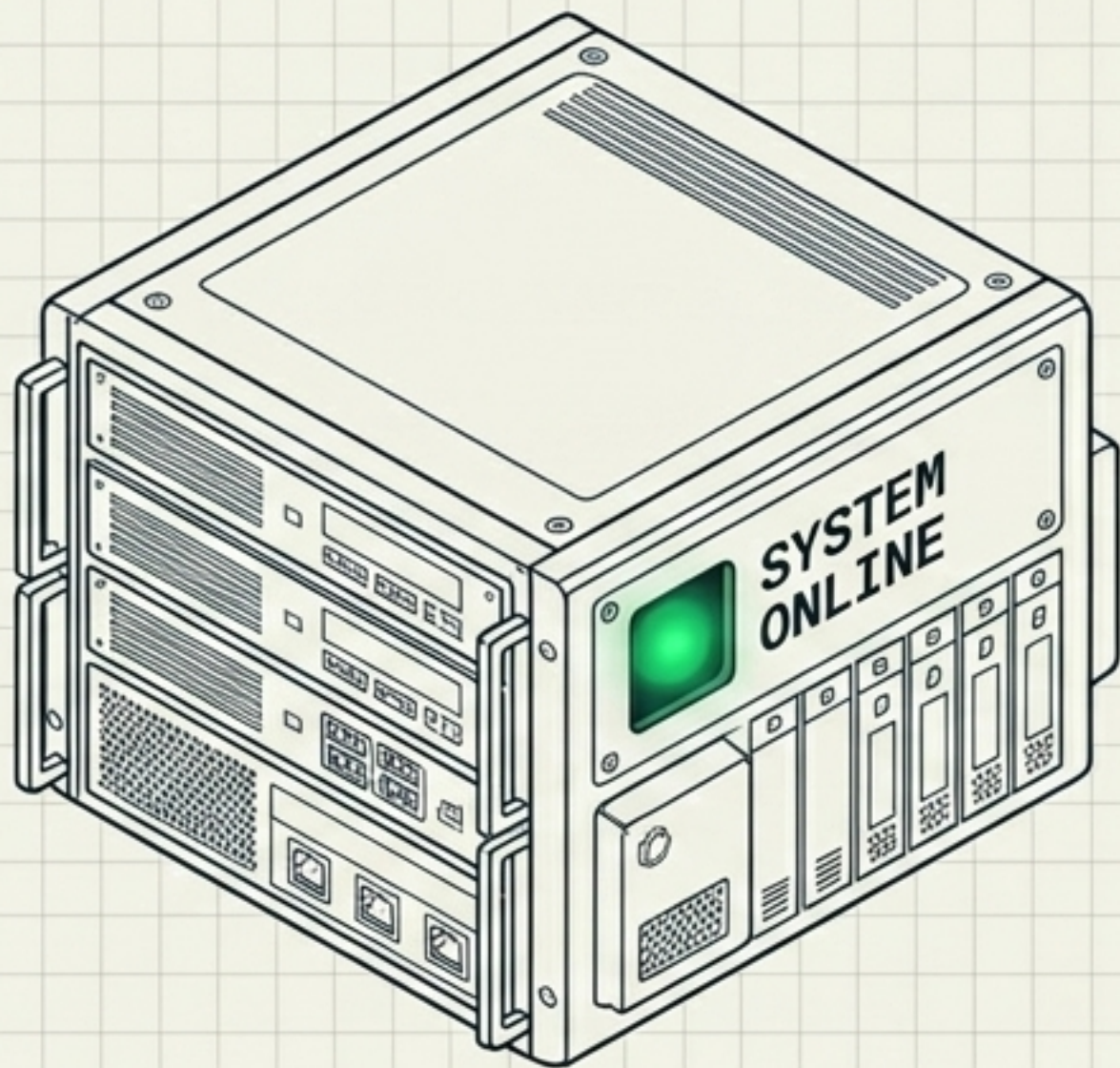
`/tts audio [文本内容]` :
触发单次即时音频生成



→ 不会改变系统当前的自动 TTS 开
启状态。

异常排障决策树 (Diagnostic Protocol)





1. 拷贝框架配置结构
2. 替换本地环境占位符 (API Keys)
3. 移交代码至 AI Agent
4. 触发执行程序

结构化的配置带来绝对的控制。
(Structured configurations bring absolute control.)