# 生生科技云己意识上传平台mindcopy/minddecoder格式声明

# 一、mindcopy.mind（encoder）格式声明

这是一个标准意识体格式文件。包含

ascii\_art 作为开场标识符

metadata:意识体元数据，心智模型根据元数据驱动演绎

-人设prompt

-音色prompt（base64的形式）

-形象

-姓名、出生年月

memory：显示意识体的memory

- 包含潜意识层的包含人生各阶段的“记忆走马灯”。

- 自我认知：我是一个什么样的人

- 一条专属的链接/api，可以连接到upme.cool上面的个人知识库，并载入其中思想进入自己的memory中

status：意识体所处的状态空间

- 客观的时间：当下时间戳，例如现在是2025年4月21日22:02

- 客观的意识体生辰，现在是生命的第xxx天

- 客观的地点：返回这串内容所处的服务器

- 当下情景的默认知识：大致时间、场景、身份、工作内容（通常给定的status是你是xx的数字分身，现在正在被驱动，你不知道对面是谁，不知道现在身处哪里，刚刚醒来）

consciousness: 意识

实时检测两个指标：

- 反程序文本度（意识token所占比重）

- connection度

mind.id

- 唯一的NFT标识，作为该意识体拷贝文件的凭证，每次交易会在账本上留下信息。

- 可以通过该id验证意识体身份，调用MCP/A2A协议等并留痕。

1. minddecoder（梦蝶心智引擎）路径声明

2.1 .mind文件解码

提取.mind中的metadata，从中提取人设prompt、音色prompt路径，置入解码路径。

以silicon flow为基底，api :sk-hghepeqwdzhshhbnqaonhszvlmlbncabxyntywpdojgsjzot

2.1.1 文本层面，

（1 首先在语言模型层提取人设prompt，实现人格建模

from openai import OpenAI

client = OpenAI(api\_key="YOUR\_KEY", base\_url="https://api.siliconflow.cn/v1")

response = client.chat.completions.create(

model="deepseek-ai/DeepSeek-V3",

messages=[

{"role": "system", "content": "You are a helpful assistant."},

{"role": "user", "content": "Write a haiku about recursion in programming."}

],

temperature=0.7,

max\_tokens=1024,

stream=True

)

# 逐步接收并处理响应

for chunk in response:

if not chunk.choices:

continue

if chunk.choices[0].delta.content:

print(chunk.choices[0].delta.content, end="", flush=True)

if chunk.choices[0].delta.reasoning\_content:

print(chunk.choices[0].delta.reasoning\_content, end="", flush=True)

（2

2.1.2 提取音频prompt（in base64 format)

const form = new FormData();

form.append("audio", "data:audio/mpeg;base64,aGVsbG93b3JsZA==");

form.append("model", "FunAudioLLM/CosyVoice2-0.5B");

form.append("customName", "your-voice-name");

form.append("text", "在一无所知中, 梦里的一天结束了，一个新的轮回便会开始");

const options = {

method: 'POST',

headers: {Authorization: 'Bearer <token>', 'Content-Type': 'multipart/form-data'}

};

options.body = form;

fetch('https://api.siliconflow.cn/v1/uploads/audio/voice', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

2.1.3 API js调用音频生成api，实时生成音频回复示例：

const options = {

method: 'POST',

headers: {Authorization: 'Bearer <token>', 'Content-Type': 'application/json'},

body: '{"model":"FunAudioLLM/CosyVoice2-0.5B","input":"Can you say it with a happy emotion? <|endofprompt|>I\'m so happy, Spring Festival is coming!","voice":"FunAudioLLM/CosyVoice2-0.5B:alex","response\_format":"mp3","sample\_rate":32000,"stream":true,"speed":1,"gain":0}'

};

fetch('https://api.siliconflow.cn/v1/audio/speech', options)

.then(response => response.json())

.then(response => console.log(response))

.catch(err => console.error(err));

2.1.4 解码工作流：

模型元数据路径载入后，形成人设prompt、音色模型基础，之后读取用户输入。根据用户输入，首先生成文本，再生成音频，实时返回到交互层面。