# 基于消费类视听终端的 AI Agent 代理

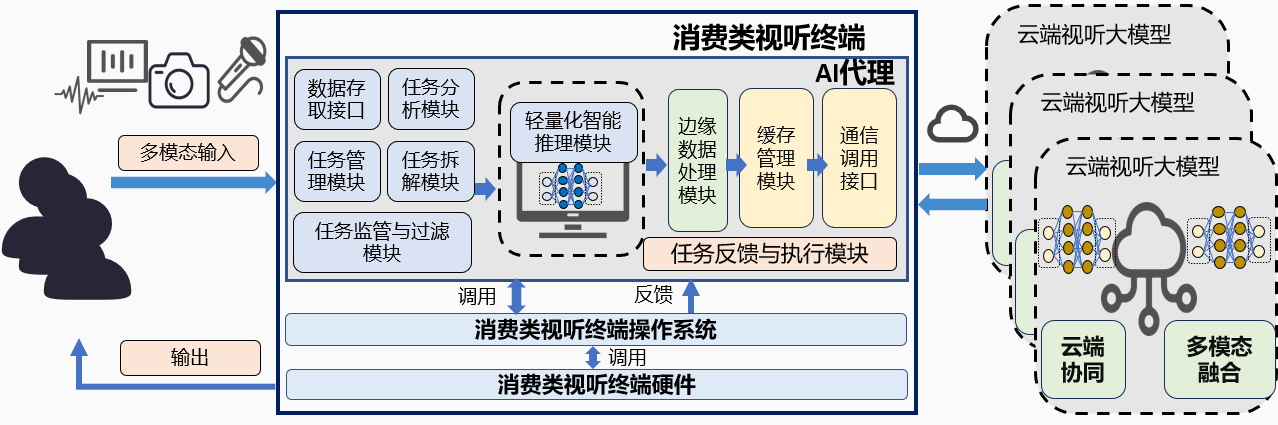
## 应用场景

融合应用场景：智能影视交互与个性化推荐系统。传统影视内容检索技术依赖关键词匹配和单一模态输入，难以处理用户基于语音、文本或模糊描述的复杂查询，导致反馈不够精准和实时。同时，用户对个性化推荐的需求日益复杂，传统规则引擎和内容筛选无法充分理解自然语言中隐含的多层次信息（如时间、偏好、数量等）。

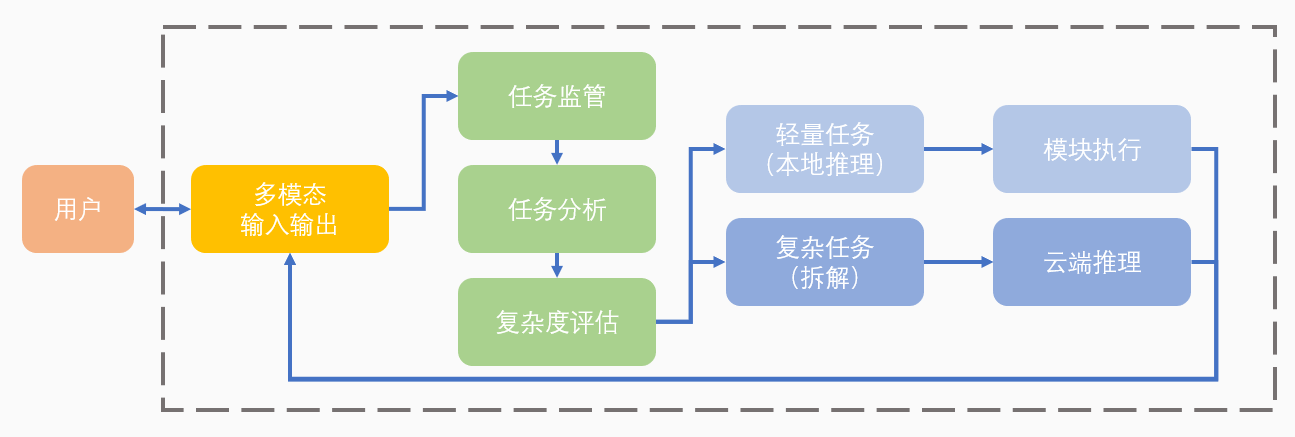
基于AI Agent的消费类视听终端操作系统，融合多模态输入处理（语音、文本、图像等）、任务分析与拆解以及云端协同推理技术，能够智能化解析用户意图，提供精准、高效的交互体验。此外，AI Agent整合多源数据（用户画像、影视数据库、观影行为数据等）和大语言模型（LLM）的推理能力，深入挖掘用户需求，实现高度个性化的影视推荐服务。

## 应用框架

AI Agent 代理应用框架采用模块化设计理念，各模块分工明确、协同工作，共同构建起一个高效、智能的应用体系，包含以下核心模块：



模块运行流程图如下：



## 前端方案（React Native）

框架技术选用：React Native 作为跨平台框架，配合 TypeScript 提升代码健壮性，通过 React Navigation 实现router管理，支持影视搜索、推荐、用户中心等模块的无缝跳转，并实现面向用户的多模态输入输出。

语音交互：语音输入集成 react-native-voice，调用设备麦克风权限，设计悬浮式按钮支持长按录音、松开发送。设计反馈机制：实时波形动画 + 语音转文字预览，错误时 Toast 提示（如“请重试”）。

智能搜索：输入框为TextInput组件，基于onChangeText支持 模糊搜索联想，FlatList 或 SectionList 展示补全建议，数据来自后端本地推理的 API 返回。

标签系统：ScrollView + Flex 动态生成标签，并可点击标签触发二次筛选，联动云端大模型 细化推荐。

多模态结果展示：图文混排列表使用 FlashList 高性能滚动组件，懒加载海报图（FastImage 缓存优化）。卡片式布局展示影视评分、简介、播放源图标；语音播报根据电影简介通过 react-native-tts 合成语音解说，支持语速调节。

用户偏好配置：基于 Redux Persist 本地存储用户高频操作（如“最近搜索”“常看类型”），首页展示快捷入口。

显示样式设置：主题切换（深色/浅色），使用 Styled Components 动态注入样式。

## 后端方案（Django + Swarm）

框架技术选用：使用django作为后端框架实现消费类视听终端的AI Agent代理，集成任务监管与过滤、任务分析、任务拆解、协同推理、轻量级智能化推理、云端视听大模型交互和任务执行与反馈功能。

协同推理：轻量级任务调用本地部署的 DistilBERT 模型进行语义解析与意图识别，复杂任务通过路由策略转发至云端 DeepSeek 大模型处理，采用 动态负载均衡 优化推理效率。

智能体协同：基于 Swarm 开源框架 构建多Agent系统，针对影视搜索、推荐等场景设计专用Prompt模板（如“解析用户隐含偏好”），通过Swarm的 hand-off 机制实Agent间任务传递与结果聚合。

数据传输模块：基于 gRPC 实现高性能远程调用，Django REST Framework 提供标准化API接口（如/api/search）。

数据管理模块：高频访问数据（如用户画像、热门影视缓存）存储于 Redis，采用LRU策略淘汰冷数据；系统配置（模型路径、API密钥）通过本地 JSON 文件 动态加载，支持热更新。

## 任务监管与过滤模块

格式规范：基于JSON Schema定义严格的任务格式规范，包括必填字段校验（如query、user\_id）、类型检查（字符串/数值/布尔）和值域验证（如时间范围限制）。

敏感词匹配：通过AC自动机算法实现高效敏感词匹配，加载超过10万条关键词规则（含政治、暴力、色情等内容）。

业务规范：业务规则引擎集成权限控制（如儿童模式过滤R级内容）、频率限制（每秒最多5次请求）和合理性检查（如单次查询最多返回50条结果）。

## 任务分析模块

* 智能体构建：构建任务分析Agent，编写prompt规范输入输出格式和任务分析标准描述，实现对用户输入任务的准确分析。
* 任务复杂度评估：构建规则引擎，结合任务分析智能体的推理功能进行任务复杂度评估，最终形成复杂度评分score（1-9）。规则引擎输入数据类型（文本）、计算量大小（简单的文本匹配与复杂的深度学习计算）、任务流程步骤数量（单步推理还是多步复杂流程）等信息，将任务分为简单文本任务（本地推理）和复杂任务（本地和云端协同推理）。
* Prompt模板：

你是一个专业任务分析员，请按以下要求处理：

1. 识别查询中的实体：电影名、演员、导演、类型、时间范围

2. 判断任务类型：搜索/推荐/问答/生成

3. 评估复杂度指标：

- 任务文本量：低(1)/中(2)/高(3)

- 计算强度：低(1)/中(2)/高(3)

- 步骤数：单步(1)/多步(2)

4. 输出JSON格式：

{

"entities": {...},

"task\_type": "...",

"complexity": <1-9>

}

示例输入："找王家卫的文艺片，要有经典台词赏析"

## 任务拆解模块

* 智能体构建：采用混合策略的任务拆解框架，并针对任务类型构建任务拆解Agent，借助 Plan-and-Execute 或 ReAct 等智能模式对复杂任务进行拆解。
* Plan-and-Execute模式：预先构建常见任务模板，如：

TEMPLATES = {

"影视推荐": ["用户画像匹配", "相似内容检索", "流行度排序"],

"内容生成": ["关键信息抽取", "文本风格设定", "多轮润色"]

}

* ReAct模式：实时调用LLM进行动态规划，prompt包含工具库描述：

可用工具：

1. search\_movie(genre, year): 影片数据库检索

2. analyze\_review(movie\_id): 影评情感分析

3. generate\_summary(keywords): 内容摘要生成

* 输出：带依赖关系的子任务DAG，通过拓扑排序确定执行顺序。

## 协同推理架构

* 动态路由引擎：基于任务分析模块的复杂度评分socre得分，自动分配计算资源。
* 本地优先：评分≤target\_score的任务（如"周星驰电影列表"）由轻量化模块处理。
* 本地云端协同：评分>target\_score的任务（如"推荐适合雨天看的治愈系电影"）拆解后重新进行复杂度评估，将评分≤target\_score的子任务由轻量化模块处理，评分>target\_score的子任务通过Swarm调度至DeepSeek处理
* 结果聚合：本地与云端子任务结果通过Django中间件统一格式化，错误时自动重试或降级

## 轻量化智能推理模块

* 模型选型：采用轻量级模型架构，使用适配中文ChatLM-Chinese-0.2B作为基础模型。
* 部署方案：部署为gRPC微服务，在inference.proto中定义推理服务的输入输出结构，并使用 线程池（ThreadPoolExecutor）支持并发请求
* 交互流程：客户端通过 LocalInferenceStub 发起 gRPC 请求，服务端加载 DistilBERT 模型进行推理并返回结构化结果（意图 + 实体）
* 使用场景：在任务拆解流程中基于prompt模板将复杂任务拆解成串行子任务；任务执行模块基于任务详情进行简单任务的文本推理和复杂任务的简单子任务文本推理（子任务并行处理，根据串行顺序整合结果），最终生成反馈结果。

## 云端视听大模型交互模块

* 智能体构建：该模块通过 OpenAI 官方 Python SDK 与 DeepSeek 云端模型对接，采用 标准化 API 请求协议 实现智能任务处理。
* 请求发起：通过 DeepSeekClient 类初始化 API 客户端，配置 api\_key 和 base\_url，自动封装认证头部。输入参数包括 Agent 配置（指令、工具函数）、对话历史（messages）和 上下文变量（动态参数传递）。
* 任务执行：使用 client.chat.completions.create() 发起请求，通过 model 参数指定模型版本（如 deepseek-chat），tools 参数绑定 Agent 的可用工具函数；自动过滤敏感字段（如 context\_variables），避免暴露给模型，仅用于内部逻辑流转。
* 结果处理：若模型返回工具调用请求（tool\_calls），解析参数并执行本地注册的函数，将结果重新提交给模型；同时基于swarm框架的hands-off功能实现动态切换 Agent，通过 Result 对象中的 agent 字段触发。
* 流式交互：通过run\_stream方法实现，建立长连接后，云端逐字返回响应数据包（JSON格式），前端实时渲染。该方法使用HTTP Streaming技术，通过yield逐步输出content字段，并用delim标记数据块边界（start/end），适合需要即时反馈的对话场景。
* 非流式交互：通过run方法实现，同步请求一次性获取完整响应，由chat.completions.create直接返回结构化结果，
* 纠错机制：两种交互模式均通过OpenAI库封装请求，自动处理认证（API Key）、超时（默认5秒）和重试逻辑（3次），错误时返回标准错误码和详情。

## 任务反馈与执行模块

* 影视检索与推荐增强反馈：基于协同推理架构处理后的结果，构建workflow进行对应影视资源封面海报url，播放链接url，整合协同推理结果中的影视文本简介，将结果返回给前端，在方格布局中渲染显示。前端在同时展示每部影片的自动播放交互按钮，用户点击即可进行自动播放。