# **Auro-PAI 平台后端软件设计文档 (SDD)**

## **1. 引言**

### **1.1. 目的**

本文档旨在为“Auro-PAI 平台”的后端服务提供全面的软件设计规范。Auro-PAI 旨在通过集成本地大型语言模型（LLM）、本地向量数据库和外部互联网访问能力，为“普通人”（Average Joys, AJs）提供上下文感知、透明且可控的AI协助，其中强大的代码增强功能是其可选但重要的组成部分。本文档将作为构建后端服务的指导蓝图，涵盖架构、模块设计、功能实现、数据管理和部署策略。

### **1.2. 范围**

本 SDD 专注于 Auro-PAI 平台后端服务的设计与实现。它详细说明了核心 FastAPI 应用程序，该应用程序将充当中央协调器，负责处理客户端请求，管理与各种 LLM（特别是通过 llama.cpp 运行的 LLaVA + Mixtral ），集成 RAG 系统 (ChromaDB)，执行外部工具（例如网络搜索和 URL 获取），以及管理会话上下文。

### **1.3. 目标与非目标**

**目标：**

* 构建一个健壮的 FastAPI 后端，作为 Auro-PAI 功能的核心。
* 实现与本地 llama.cpp LLM 服务器（托管 LLaVA + Mixtral 8x7B GGUF 模型）的无缝集成。
* 开发检索增强生成 (RAG) 管道，利用本地代码库和文档进行上下文检索。
* 集成 LLM 驱动的工具使用功能（网络搜索、URL 获取），使 AI 能够智能地获取外部信息。
* 确保 AI 建议（如代码重构）是透明的，并以结构化、可审查的格式呈现。
* 支持多种 LLM 提供商（本地、OpenAI、Gemini）以提供灵活性。

**非目标：**

* 本文档不涵盖前端（例如 VS Code 扩展）的详细设计。
* 不涉及 llama.cpp 或 ChromaDB 本身内部的修改。它们被视为外部依赖或服务。
* 不涉及复杂的认证和用户管理，假定为单用户或信任环境。

### **1.4. 目标受众**

* 软件开发工程师
* 系统架构师
* 质量保证工程师
* 运维工程师

## **2. 高级架构**

Auro-PAI 平台后端采用分层、模块化架构。FastAPI 应用是核心的协调器，它解耦了客户端接口、AI 逻辑、数据检索和外部工具交互。

+--------------------------+  
| Client (e.g., curl, |  
| VS Code Extension) |  
+------------+-------------+  
 | HTTP/JSON API Requests  
 |  
 v  
+--------------------------+  
| FastAPI 后端 |  
| (Python Application Layer)|  
+------------+-------------+  
| - API Endpoints |  
| - Request/Response Models|  
| - Agentic Loop (Tool Orchestration) |  
| - Context Management |  
| - Diff Generation |  
+------------+-------------+  
 | | | |  
 | | | | LLM Inference Requests (HTTP)  
 | | | v  
 | | | +--------------------+  
 | | | | llama.cpp LLM Server |  
 | | | | (Hosts LLaVA + Mixtral 8x7B GGUF) |  
 | | | +--------------------+