TED UNIVERSITY

SENG 491 Senior Project

Local Generative AI Services Super App

High-Level Design Report

27.12.2024

Team Name: DBU

Team Members:

Deniz ÖZCAN, 33577146512, Software Engineering

Betül Ülkü YURT, 11056264926, Software Engineering

Umut ŞAHİN, 11597931646, Software Engineering

Supervisor:

Emin Kuğu

Jury Members:

Tansel Dökeroğlu

Kasım Murat Karakaya

1. Introduction

1.1 Purpose of the System

The purpose of the system is to provide a unified web and mobile application platform that integrates a range of AI-powered mini-apps to address diverse user needs. By leveraging various Large Language Models (LLMs) and implementing user-specific API keys, the system delivers personalized, secure, and innovative solutions for ai-generated services that include but not limited to; language translation, content creation, digital accessibility, and storytelling. Since the range of services we provide and implement will be done in a agile structure that changes based on the market research and user feedback, we will not put a constraint on the feature set of the overall system and implement and ship mini-apps according to current trends. The system emphasizes accessibility, user-centric design, and seamless integration of advanced AI capabilities.

1.2 Design Goals

1. **Scalability**: Enable the addition of new mini-apps without impacting platform performance since mini-apps are the most important attribute to our platform, this will be high priority.
2. **Security**: Using industry leading technology and patterns, ensure the security of user data, this includes but not limited to user credentials, prompts and files uploaded to the mini-apps.
3. **Accessibility**: Provides an intuitive interface for users that is easy to use and master.
4. **Customization**: Allow advanced users to use their private endpoints or API keys and host their models for increased data control and security.
5. **Efficiency**: Deliver fast and accurate responses by leveraging pre-defined prompts and advanced AI models, we will priotize high quality output but also keep eye on acceptable and generally not noticeable wait times for all generations done by the services, these will differ from service to service and the input size.
6. **Sustainability**: Promote efficient resource usage and support local hosting options, since advanced users are paying by token to the API that they utilize, keeping the token sizes small but effective by optimizing the prompts is one of the prioritized quality attributes of our platform.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
API	Application Programming Interface - A set of protocols for building and integrating software applications.
GDPR	General Data Protection Regulation - EU law for data privacy and protection of personal data.
LLM	Large Language Model - AI models trained on vast datasets to understand and generate human-like text.
TTS	Text-to-Speech - A technology that converts text into spoken audio.
MFA	Multi-Factor Authentication - Security system requiring multiple forms of verification for access.
Encryption	Process of converting data into a secure format to protect it from unauthorized access.
Agent Coordination	Mechanism for managing collaboration and communication among agents in multi-agent systems.
Fallback Mechanisms	Predefined strategies to handle failures by retrying or switching to alternative solutions.
Scalability	The ability of a system to handle increased workloads by adding resources without affecting performance.

1.4 Overview

This report details the system's architecture, subsystem decomposition, hardware/software mapping, persistent data management, security policies, and global software control. The report includes UML diagrams to illustrate the proposed architecture and service design.

2. Current Software Architecture

Currently, AI applications are scattered across various platforms, lacking standardization and integration they use a single chatbot to respond to all user needs, and since their focus is too broad, generated content is often not what the user specifically needs. Users face challenges like going back and forth to complete simple tasks, reliance on external cloud services, and limited customization options. Also, existing tools raise privacy and security concerns due to centralized data storage. The absence of real-time grounding and multi-agent systems reduces their capability to provide relevant and accurate responses. There is a concrete need for specialized ai-apps that focus on specific subjects to make them generate better outputs that cater to user needs.

3. Proposed Software Architecture

3.1 Overview

The proposed system integrates various AI-powered mini-apps into a single, scalable platform. It features:

- **Modular design**: The mini-apps and services are easily updatable, expandable and more functionalities can be added to the platform by the developers in line with the current market demands.
- **Secure API key management**: Ensures personalized AI usage by encrypting user data and providing privacy options.
- **Local hosting options**: Allows users to manage their data securely while maintaining operational flexibility.

3.2 Subsystem Decomposition

The system is divided into the following subsystems:

1. **User Interface (UI)**:
    a. Central dashboard displaying all mini-apps.
    b. Intuitive navigation for seamless user experience.
    c. Responsive design supporting web and mobile platforms.
2. **Mini-App Services**:
    a. Independent operation of each mini-app to ensure modularity.
    b. Tailored prompts for specialized outputs (e.g., translation, video captions).
3. **Data Management**:
    a. Secure collection, storage, and processing of user data.
    b. GDPR-compliant policies for data retention and deletion.
4. **AI Integration Services:**
    a. Provides centralized registry, version control, and configuration management for AI models.
    b. Tracks latency, accuracy, and resource usage to optimize AI responses.
    c. Coordinates agents for distributed processing, response aggregation, and performance optimization.
    d. Ensures template consistency, context optimization, and quality assurance for AI interactions.

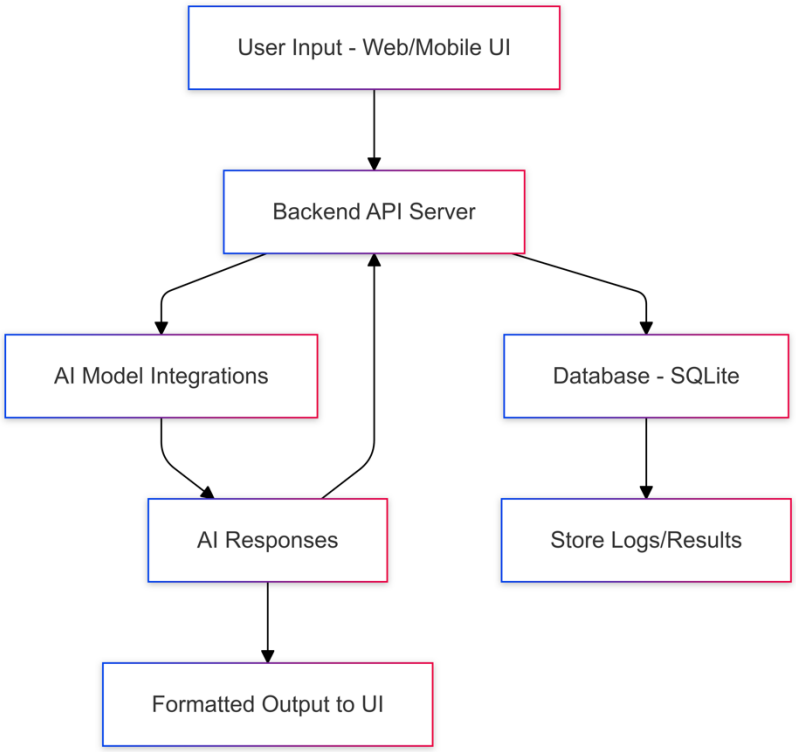
3.3 Hardware/Software Mapping

The hardware and software components are mapped as follows:

- **Frontend**: Built with React.js for web and Swift for mobile compatibility.
- **Backend**: Python-based server using Flask for core business logic, AI processing, and integration with third-party APIs.
- **Database**: SQLite for structured data management, ensuring reliability and scalability.
- **AI Integration**: Gemini, OpenAI, and locally hosted Llama models.

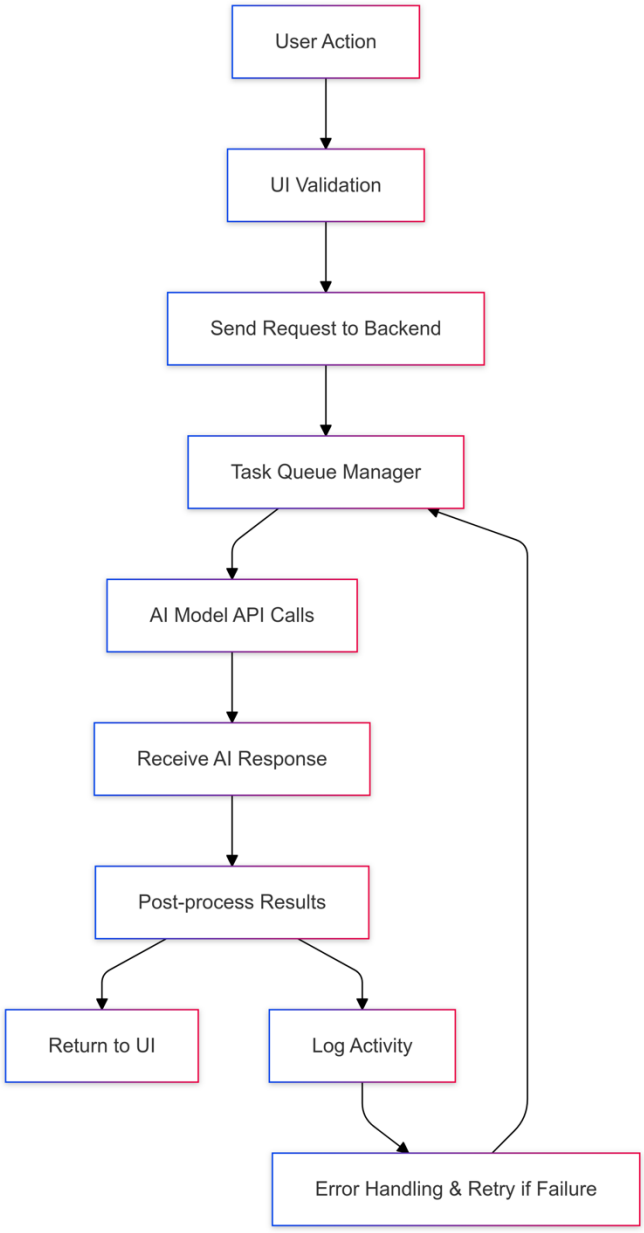
Data Flow:

- User inputs are collected through the UI and sent to the backend for preprocessing.
- The backend validates inputs and directs requests to the appropriate AI models via API integrations.
- Responses from AI models are formatted and sent back to the UI for display.



Control Flow:

- Centralized backend orchestrates task delegation among subsystems.
- Workflow management systems handle asynchronous tasks and retries in case of errors.
- Logging and monitoring components ensure performance tracking and issue diagnosis.



3.4 Persistent Data Management

- **Encryption**: All sensitive data, including user inputs and generated outputs, is encrypted at rest and in transit.
- **User Control**: Provides clear options for data deletion and account removal.
- **Storage Efficiency**: Ensures optimal resource utilization for generated outputs (e.g., videos, images).

3.5 Access Control and Security

- Implements role-based access control to ensure secure module interaction.
- Utilizes secure authentication mechanisms.
- Conducts regular security audits to mitigate potential vulnerabilities.

3.6 Global Software Control

- Event-driven architecture enables real-time response handling.
- Robust error handling ensures uninterrupted user experiences.
- Automated logging supports debugging and continuous improvement.

3.7 Boundary Conditions

- Handles edge cases with clear error messaging for unsupported inputs.
- Ensures graceful degradation under high load conditions.

4. Subsystem Services

This section breaks down each subsystem and its offerings to create a big-picture design plan for the suggested platform. We've designed every subsystem to address specific functional and non-functional requirements, ensuring scalability, usability, and performance.

4.1 User Interface Services

Dashboard:

- **Centralized Access:** Provides a unified interface for accessing all mini-apps, organized into categories for intuitive navigation. Each service is represented by a visually distinct card layout, enhancing discoverability across both mobile and web platforms.
- **Search Bar:** Enables quick filtering of services by name using predictive text and case-insensitive matching, available on both mobile and web interfaces.

Dynamic Design Elements:

- **Interactive Cards:** Service cards display icons, titles, and gradients, visually distinguishing each app while ensuring consistency across the interface on both mobile and web.
- **Navigation Links:** Seamlessly transitions between services using embedded navigation links, allowing smooth user flow whether accessed from a smartphone or desktop browser.

Adaptive Layouts:

- **Responsive Design:** Optimized for various screen sizes, including desktops, tablets, and mobile devices, ensuring uniform experiences across platforms.
- **Grid View:** Dynamically adjusts grid layouts for better content presentation and usability on different resolutions, leveraging flexible layouts for mobile screens and expanded views for desktops.

Theme and Accessibility:

- **Dark Mode Support:** Provides visually comfortable themes, including a dark mode for low-light conditions, available on both web and mobile applications.
- **Contrast and Fonts:** Ensures readability through high-contrast colors and scalable fonts adaptable to all devices.

Feedback Mechanism:

- **Real-time Alerts:** Displays error messages, success confirmations, and progress indicators for actions, synchronized across devices.
- **User Input Validation:** Ensures smooth interactions with immediate feedback when input issues are detected, maintaining consistency between mobile and web platforms.

User Experience Optimization:

- **Status Bar Padding (Mobile):** Maintains spacing for mobile usability while adapting header elements dynamically.
- **Header Adjustments (Web):** Provides fixed headers for ease of access during scrolling.
- **Tutorial System:** Includes onboarding tutorials and tooltips to guide users effectively.

4.2 Mini-App Services

Modular Functionality:

- **Independent Services:** Each mini-app operates independently, enabling modular upgrades and maintenance without affecting other components. This structure is supported across both web and mobile platforms, allowing seamless integration and updates.
- **Scalable Design:** New mini-apps can be added without impacting existing functionalities, leveraging a shared backend architecture and responsive front-end frameworks adaptable for web and mobile environments.

Specialization:

- **Custom Prompts:** Pre-configured AI prompts are optimized for the specific purpose of each mini-app, ensuring consistent performance across mobile and web applications.
- **Algorithm Optimization:** Tailored algorithms handle diverse tasks like image generation, translation, and summarization, providing high-performance outputs regardless of platform.

Extension Support:

- **API Integrations:** Facilitates adding third-party features and AI models without modifying the core system, ensuring compatibility with both web and mobile versions through unified API endpoints.

Specialized Mini-App Services

1. **Video Language Translation Service**
    a. Speech-to-text conversion
    b. Language translation
    c. Text-to-speech synthesis
2. **Multilingual Audio Document Service**
    a. Text extraction
    b. Language detection
    c. Voice synthesis
3. **Bedtime Story Creator Service**
    a. Theme management
    b. Character generation
    c. Story structure templates
    d. Language adaptation
    e. Age-appropriate content filtering
4. **Video Auto-Caption Service**
    a. Speech recognition
    b. Caption generation
5. **Daily Recap Service**
    a. Content aggregation
    b. Summary generation
    c. Personalization
6. **Text-to-Image Generation Service**
    a. Prompt preprocessing
    b. Image generation

4.3 Data Management Services

Secure Storage:

- **GDPR Compliance:** Implements policies to protect sensitive information and privacy rights across both web and mobile platforms.
- **Encryption Standards:** Ensures encryption of data both at rest and in transit, applying standardized protocols suitable for mobile and web security frameworks.

Backup and Recovery:

- **Encrypted Backups:** Provides secure storage of backup files with encryption methods compatible across cloud and local storage for mobile and web.
- **Restoration Options:** Allows users to quickly recover data in case of failure, ensuring cross-platform accessibility and synchronization.

Audit Logs:

- **Activity Tracking:** Maintains detailed logs of all user actions for accountability and debugging, accessible securely from both web and mobile interfaces.
- **Role-Based Access:** Implements authorization mechanisms to ensure logs are only accessible to authorized personnel, maintaining consistency across platforms.

4.4 AI Integration Services

Model Management

- **Model Registry:** Provides centralized storage for managing AI models, enabling tracking and easy deployment of new versions.
- **Version Control:** Ensures consistency and reproducibility by maintaining different versions of models and configurations.
- **Configuration Management:** Allows parameter tuning and storage of configuration files for efficient deployment.
- **Performance Monitoring:** Tracks accuracy, latency, and error rates to ensure optimal performance.
- **Resource Allocation:** Dynamically allocates computing resources to handle workloads and optimize performance.

API Key Management

- **Key Encryption:** Protects API keys using industry-standard encryption algorithms, ensuring security for both web and mobile platforms.
- **Access Control:** Implements role-based access controls to restrict usage based on permissions and roles.
- **Usage Monitoring:** Tracks API key usage to prevent overconsumption or misuse.
- **Key Rotation:** Supports periodic updates and rotation of keys to enhance security.
- **Validation Service:** Verifies key validity and permissions during setup and usage, providing error alerts and reconfiguration options.

- **Third-Party Integration:** Supports APIs like OpenAI, Gemini, and custom AI models, ensuring compatibility across platforms.
- **Scalability:** Facilitates expansion to accommodate future AI services without disrupting web or mobile functionality.

Multi-Agent System

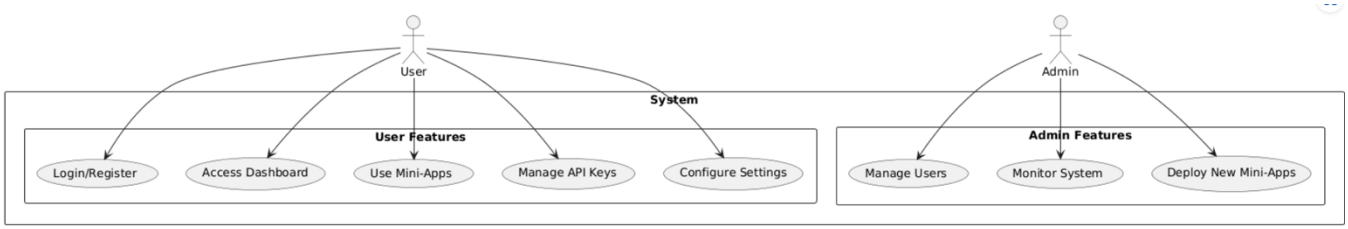
- **Agent Coordination:** Ensures effective task distribution among agents, optimizing collaboration and output quality.
- **Task Distribution:** Delegates tasks to specific agents based on workload and specialization.
- **Response Aggregation:** Combines outputs from multiple agents to deliver unified and high-quality responses.
- **Error Handling:** Provides fault tolerance by retrying failed operations or escalating issues to fallback mechanisms.
- **Performance Optimization:** Continuously evaluates agent performance to improve workflows and reduce latency.

Prompt Management

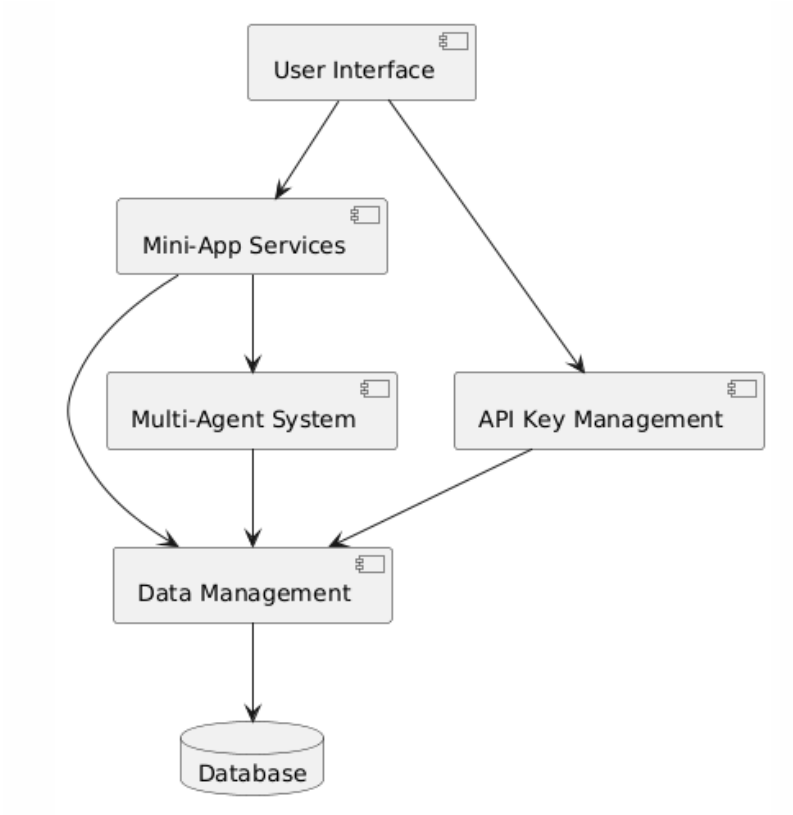
- **Template Management:** Predefined templates for consistent prompts across mini-apps, ensuring structured inputs and outputs.
- **Context Optimization:** Dynamically adapts prompts based on user preferences and historical inputs to improve relevance.
- **Parameter Validation:** Checks prompt parameters to avoid errors and enhance reliability.
- **Response Formatting:** Structures outputs in readable formats, ready for user interpretation or downstream processing.
- **Quality Assurance:** Implements validation checks and automated tests to maintain prompt quality and usability.

5. UML Diagrams

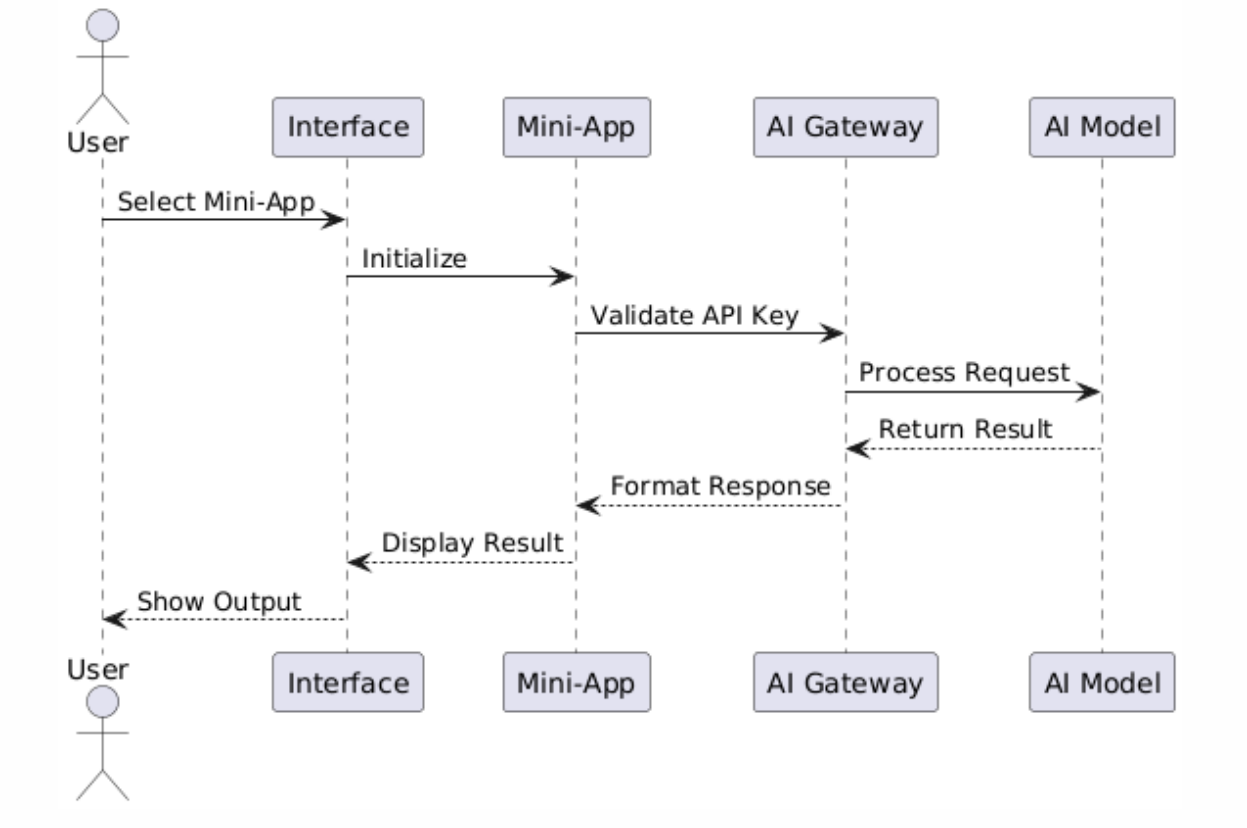
5.1 Use Case Diagram



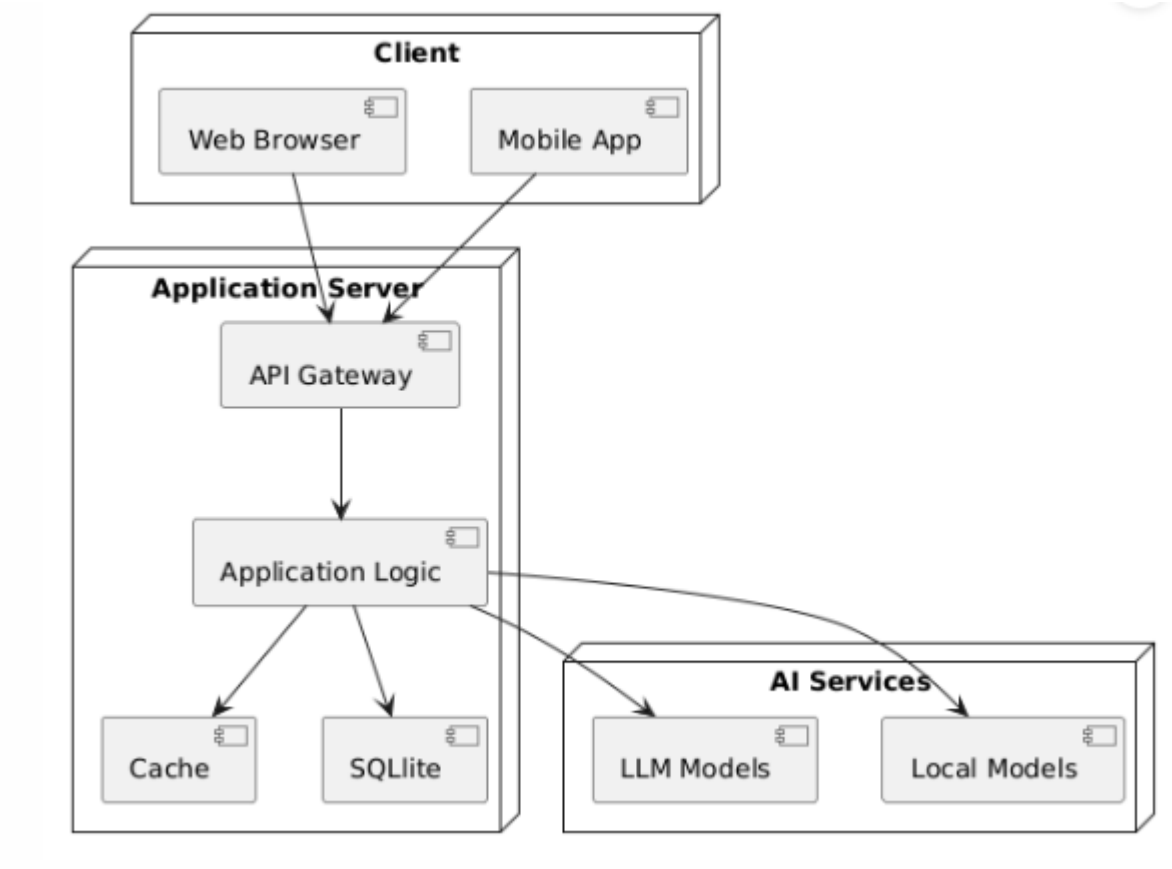
5.2 Component Diagram



5.3 Sequence Diagram



5.4 Deployment Diagram



6 Glossary

Artificial Intelligence (AI): A field of computer science focused on creating systems capable of performing tasks that typically require human intelligence, such as natural language processing, decision-making, and image recognition.

API (Application Programming Interface): A set of protocols and tools for building and integrating software applications, enabling communication between different systems.

GDPR (General Data Protection Regulation): A legal framework established by the European Union to protect individuals' personal data and privacy rights.

LLM (Large Language Model): A machine learning model trained on extensive datasets to understand and generate human language.

MFA (Multi-Factor Authentication): A security method requiring multiple forms of verification, such as passwords and codes sent to mobile devices, to access systems.

Prompt Management: The process of designing, optimizing, and validating prompts used to interact with AI models, ensuring effective communication and high-quality outputs.

Encryption: The process of converting data into a secure format to protect it from unauthorized access.

API Key: A unique identifier used to authenticate and authorize requests made to an API.

Agent Coordination: A mechanism within multi-agent systems that manages the collaboration and communication among agents to perform distributed tasks efficiently.

7 References

Ergun, O. (n.d.). *Understanding high-level design: An introduction for beginners*. Orhan Ergun. Retrieved December 27, 2024, from https://orhanergun.net/understanding-high-level-design-an-introduction-for-beginners

GDPR Regulation Guidelines. (2016). European Parliament. Retrieved December 27, 2024, from https://gdpr.eu/what-is-gdpr/