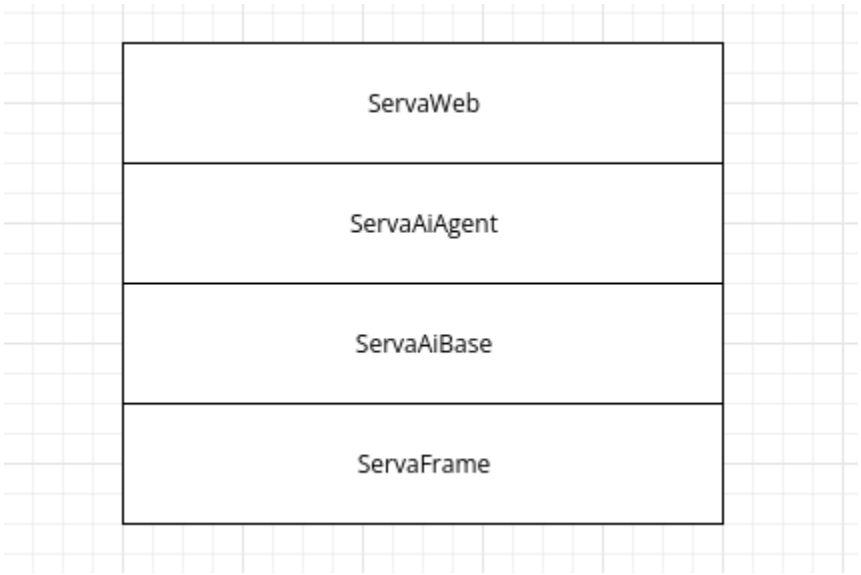
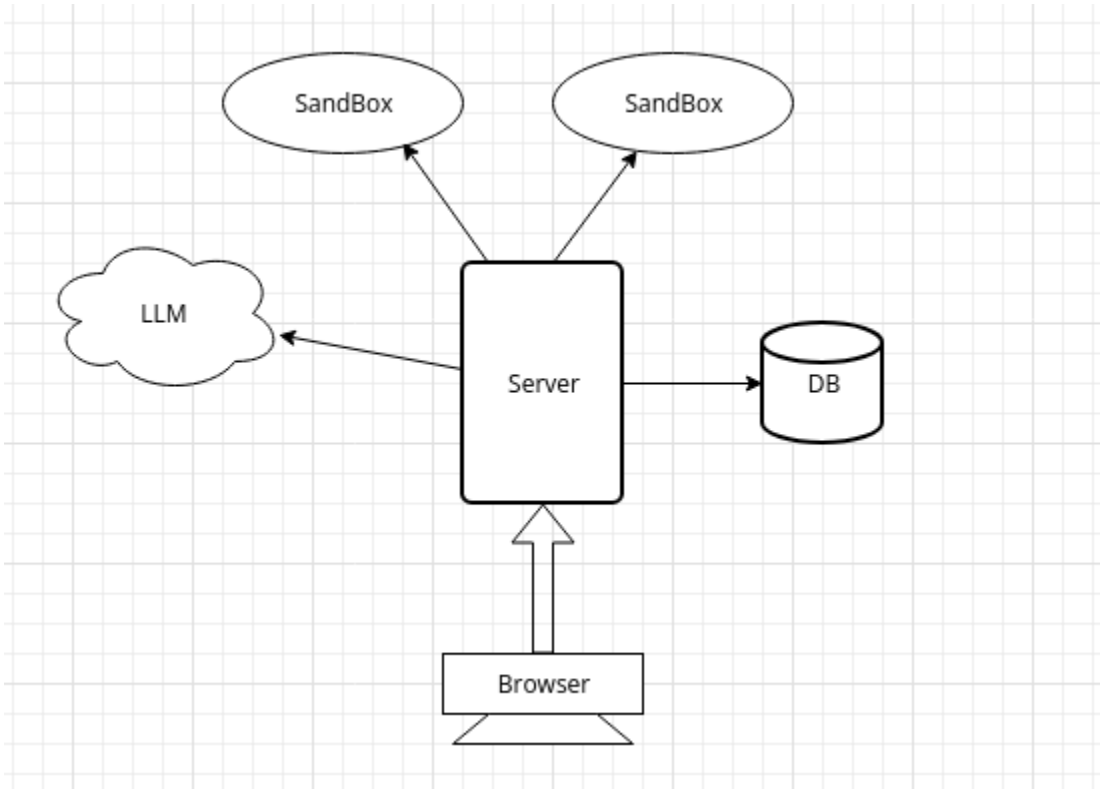
Module Architecture

Module dependents

ServerWeb — (depends on) —>ServerAiAgent — (depends on) —> ServerAiBase —> (depends on) —> ServerFrame



Deploy architecture



- **LLM**: including OpenAI GPT Models, Google Gemini Models, providing AI Interfaces via RESTful API
- **Server**: Provides core functions via RESTful API for browsers to handle requests from clients, coordinate between LLM and sandbox.
- **SandBox**: provides virtual environments for code generation, compiling, testing, error correction and packaging.
- **DB**: provides configuration and information recording

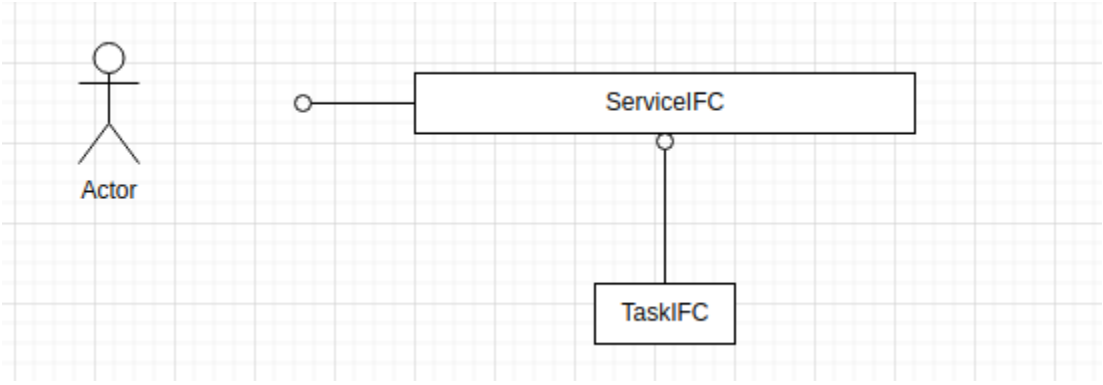
ServaFrame Design

Conceptual Framework

With servaframe, actions are categorized into two distinct dimensions: non-business logic and business logic.

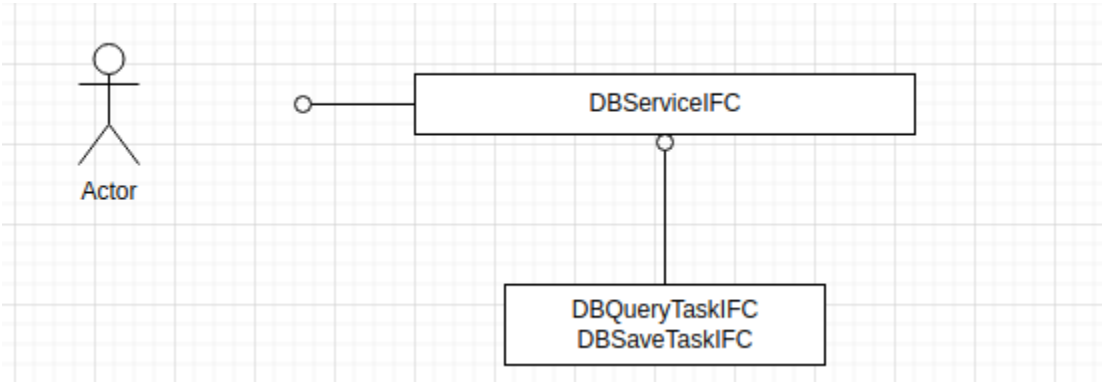
- **Non-business Logic:** This dimension is managed by the Service component, which handles infrastructure-level tasks such as database connectivity, transaction management, and resource allocation.

- **Business Logic:** This dimension is handled by the Task component, which implements the core business processes and logic specific to the application's functionality.



Component Responsibilities

- **Service:** The Service component is responsible for non-business logic tasks. For instance, DBService handles database connectivity, establishing a connection to the database, and managing the lifecycle of this connection (including commit, rollback, and closure operations).
- **Task:** The Task component provides a callback interface to the Service and is responsible for executing business logic. There are two types Task:
    - **DBQueryTask:** Responsible for business which only needs to retrieve data from the database.
    - **DBSaveTask:** Responsible for business which might save data to the database, all save actions in this task will be in one transaction.
    - **DBAutoCommitSaveTask**: Responsible for business which might save data to the database, each save action will be committed automated.

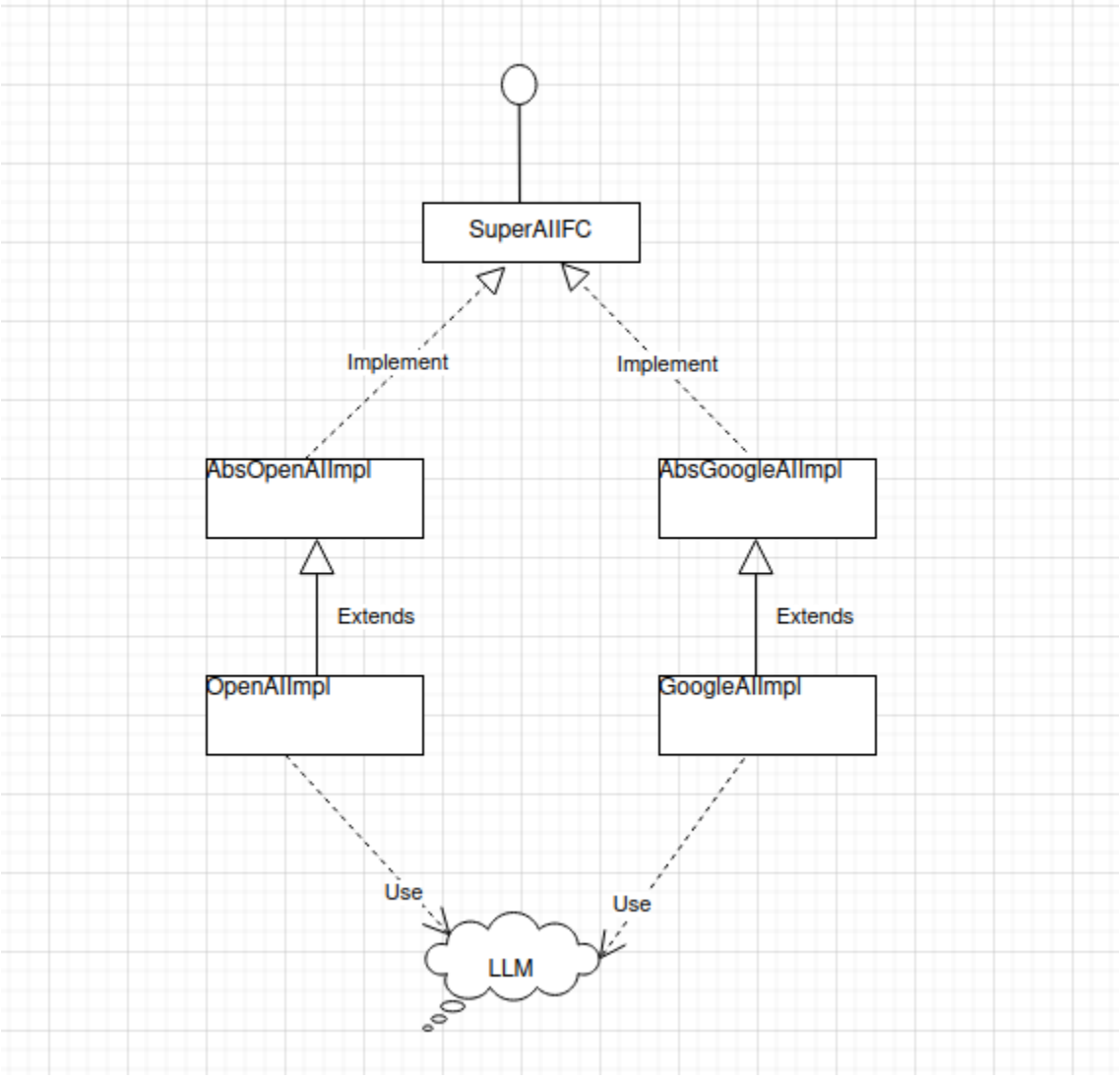


DBQueryTask/DBSaveTask/DBAutoCommitSaveTask will be injected with an instance of **DBConnection** by DBService.

By delineating responsibilities between the Service and Task components, the server framework ensures a clear separation between infrastructure management and business logic execution, leading to a more maintainable and scalable system architecture.

ServaAIBase

ServaAIBase acts as the basic layer which encapsulates LLM, and provides common interfaces via SuperAIIFC.



SuperAIIFC supports below functions:

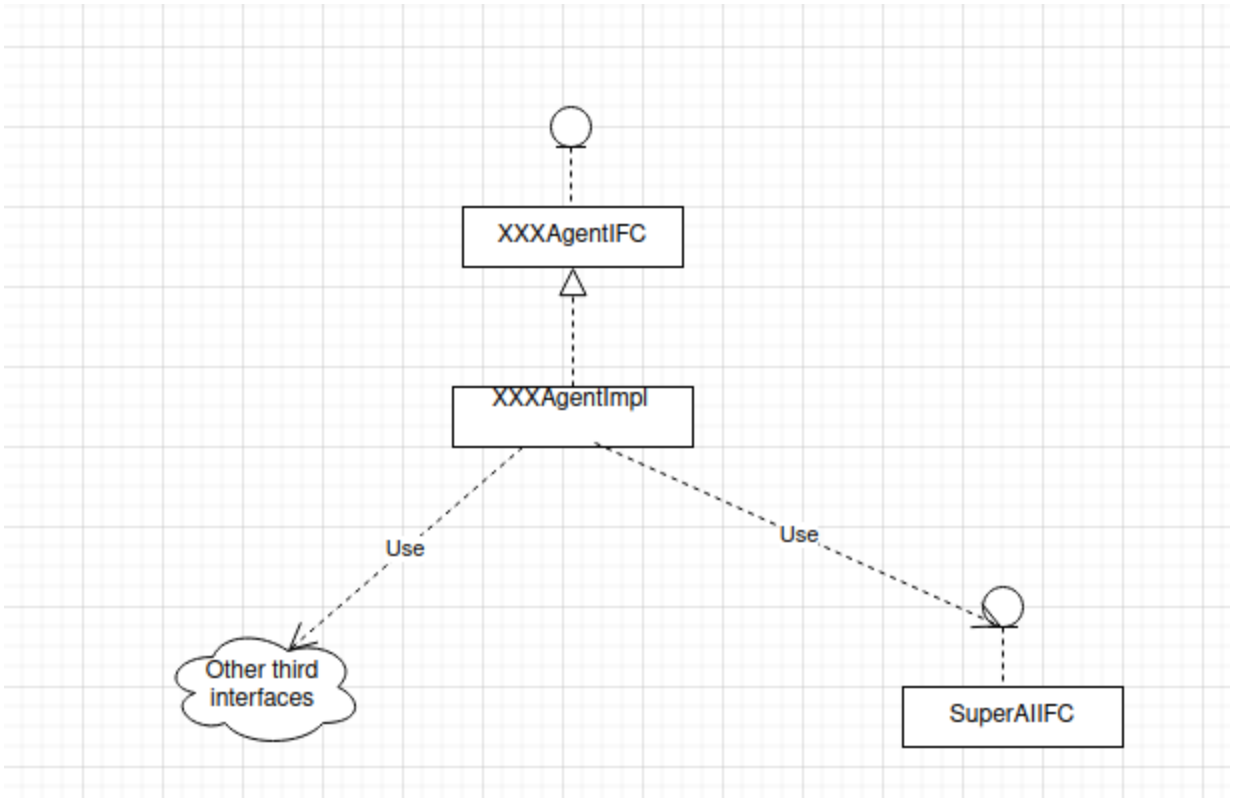
- Chat
- Chat with FunctionCall (FunctionCall is the core to support Code Generation)
- Image generation
- Vision
- Embedding
- Text/speech conversion

LLM model supports:

- OpenAI Models
    - gpt-4o-mini
    - gpt-4o
    - gpt-4-turbo-preview
    - gpt-3.5-turbo
    - text-embedding-3-large
    - text-embedding-3-small
    - dall-e-3
    - dall-e-2
    - gpt-4-vision-preview
    - tts-1
    - tts-1-hd
    - whisper-1
    - To be extended…
- Google Gemini Models
    - gemini-1.0-pro
    - gemini-1.5-pro-latest
    - embedding-001
    - gemini-pro-vision
    - To be extended…

ServaAiAgent Design

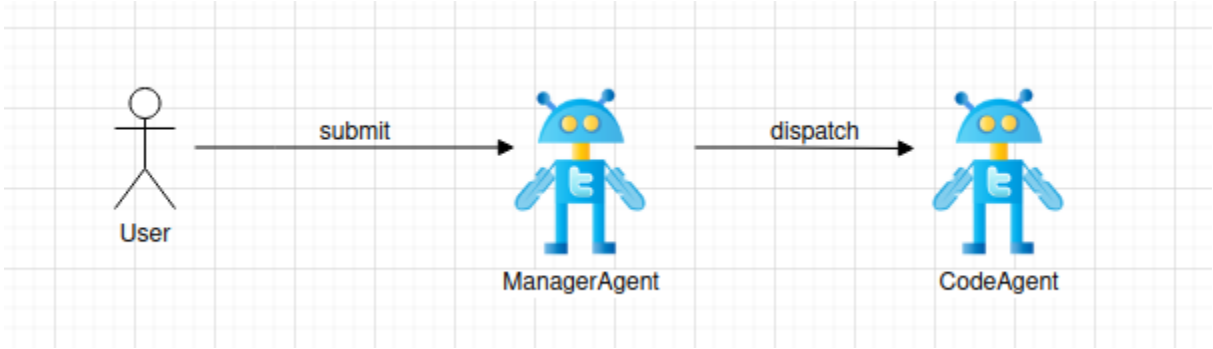
ServaAiAgent is based on ServaAiBase, it defines interfaces to support specific requirements. The implementations call the APIs of ServaAiBase to achieve the requirements.



Some Agents Description

CodeGeneration agents

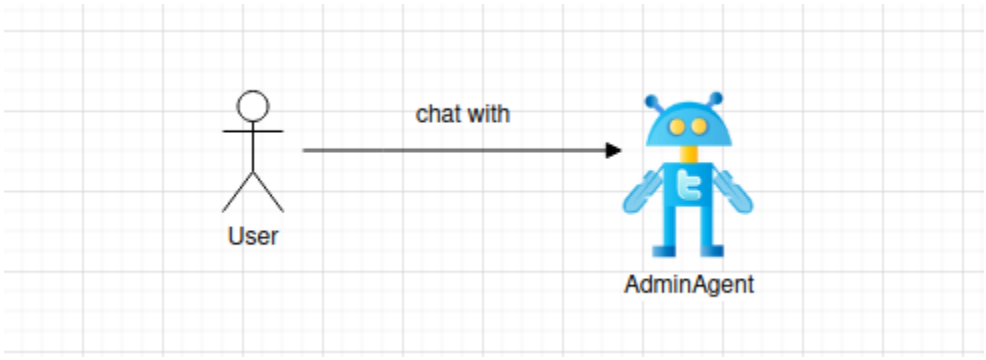
- ManagerAgent
- CoderAgent



The user submits a task, AdminAgent receives the requirement, coordinates with a suitable codeAgent, then assigns the task to the codeAgent. CodeAgent will try to generate code and compile,test, debug code in a sandbox environment. When code could run success, package this whole code in a .tar.gz file and provide a download link.

Administrator Agent

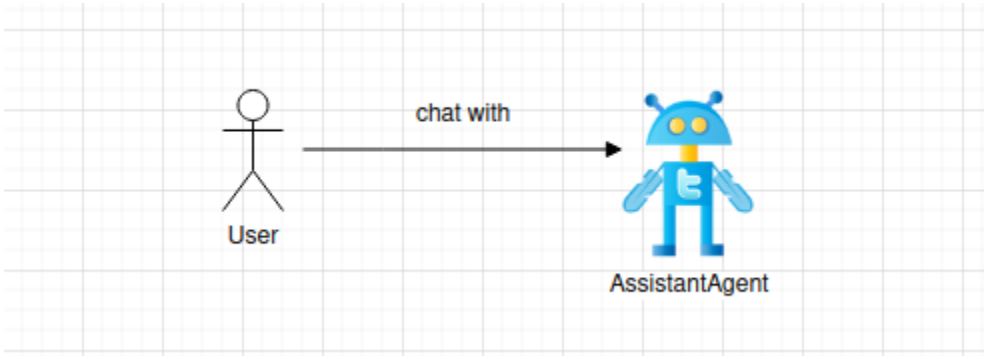
- AdminAgent



AdminAgent acts as a chatbot who provides management tasks with administrators. Only Administrator accounts have the permission to chat with it..

Assistant Agent

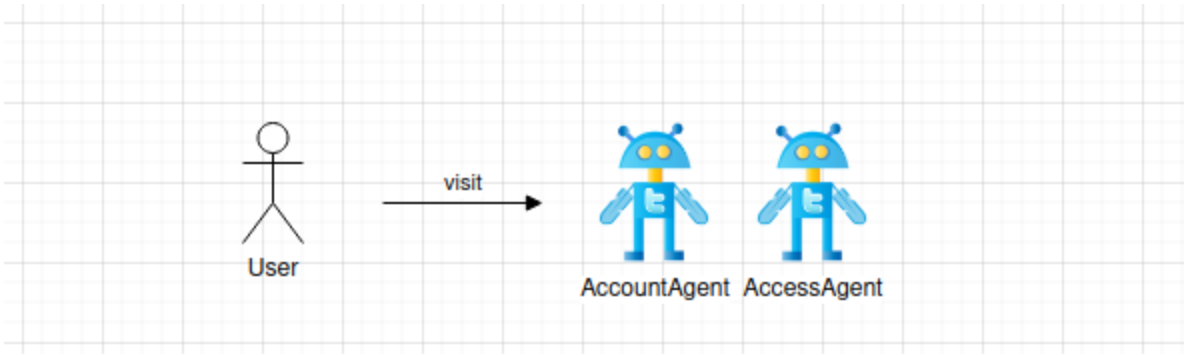
- AssistantAgent



AssistantAgent acts as a chatbot which provides assistance for normal users.

Access Control

- AccountAgent
- AccessAgent



All visit action will be controlled by the Access control module, which contains AccountAgent and AccessAgent. AccountAgent is responsible for user register and login, login session maintenance, credits related control works. AccessAgent is responsible for region limitation, IP limitation, white list/blacklist related control works.