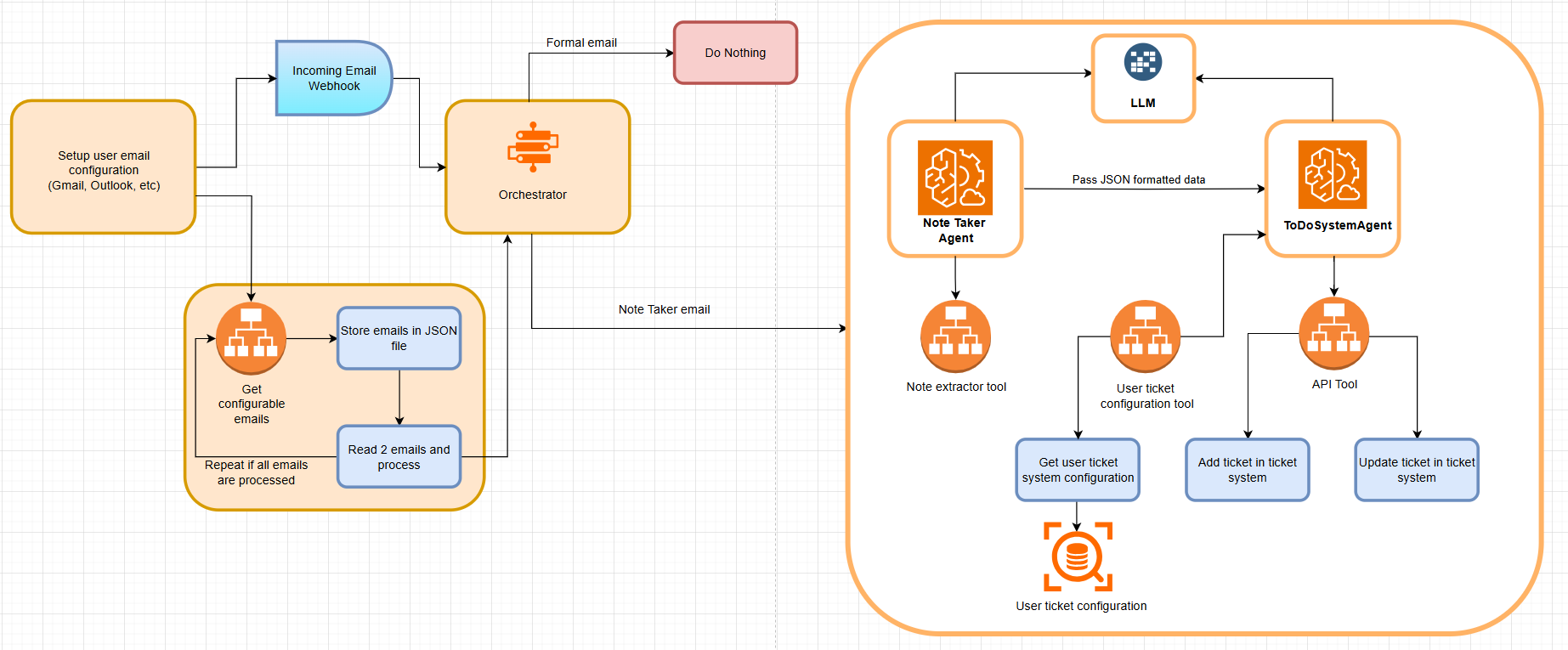
Meeting Action Item AI Agent - Technical Documentation

# 1. System Overview

The Meeting Action Item AI Agent is a Python-based intelligent system designed to process meeting notes received via email from services like Fathom, TickTick, etc. It automatically extracts actionable tasks using a large language model (LLM) via the configurable API, checks these against existing tickets from Ticket system, updates existing ones, and creates new tasks when needed.  
Targeted for developers and ticket system users, the system leverages an agentic architecture for flexibility and extensibility.



# 2. Architecture & Workflow

The system follows a modular and extensible agent-based architecture composed of:  
- \*\*Webhook Receiver\*\*: Listens for incoming meeting note emails.  
- \*\*Orchestrator\*\*: Controls the flow and coordination between agents.  
- \*\*KeyPointExtractorAgent\*\*: Extracts task-related information from email content using configurable Model and API.  
- \*\* ToDoSystemAgent \*\*: Interacts with the Ticket system to create or update tickets.  
- \*\*Tools\*\*: Specialized Python functions that support each agent in carrying out specific tasks.

Workflow Steps:  
1. An email from a notetaker service is received via webhook.  
2. The orchestrator sends the content to `KeyPointExtractorAgent`.  
3. The agent uses the configurable Model and API to extract and format actionable tasks.  
4. The orchestrator sends tasks to the `Ticket system `.  
5. The `ToDoSystemAgent` fetches existing tasks, performs semantic comparison, updates existing or creates new ’Ticket system’.

# 3. Agents and Responsibilities

\*\*KeyPointExtractorAgent\*\*  
- Identifies whether the email is from Fathom, TickTick, or other sources.  
- Extracts actionable items from the content using LLaMA3-70B.  
- Structures output into JSON.  
- Filters out duplicates using semantic comparison.  
  
\*\*JiraAgent\*\*  
- Fetches current JIRA tasks.  
- Performs task comparison.  
- Updates task status if a match is found.  
- Creates new JIRA tickets for new tasks.

# 4. Tools Description

- `extract\_email\_body`: Extracts clean content from email structure.  
- `extract\_email\_task`: Uses LLM to extract action items.  
- `fetch\_existing\_jira\_tickets`: Pulls current tickets from JIRA.  
- `create\_jira\_ticket`: Creates a new ticket in JIRA with task details.  
- `update\_jira\_task\_status`: Updates an existing ticket status or content.

# 5. Installation Guide

\*\*Requirements\*\*  
- Python 3.10+  
- Access to GROQ API (with llama3-70b-8192 model)  
- JIRA account and API token  
- FastAPI for running webhook  
  
\*\*Steps\*\*  
1. Clone the repository.  
2. Create and activate a Python virtual environment.  
3. Install dependencies using `pip install -r requirements.txt`.  
4. Create a `.env` file with the following keys:  
 - `GROQ\_API\_KEY`  
 - `JIRA\_BASE\_URL`  
 - `JIRA\_EMAIL`  
 - `JIRA\_API\_TOKEN`  
5. Run the FastAPI server using `uvicorn main:app --reload`.

# 6. Developer Guide

\*\*Project Structure\*\*  
- `agents/`: Contains KeyPointExtractorAgent and JiraAgent.  
- `tools/`: Houses all modular tool functions.  
- `orchestrator.py`: Core logic for sequencing agents and tools.  
- `email\_webhook.py`: Webhook endpoint using FastAPI.  
  
\*\*Extending the System\*\*  
- Add new tools in the `tools/` folder and register them in the orchestrator.  
- Create new agents by subclassing the Agent class and assigning them tool access.  
- Update `.env` and `requirements.txt` accordingly.

# 7. User Manual

\*\*Deployment\*\*  
- Host the FastAPI app on a server with HTTPS enabled for webhook access.  
  
\*\*Monitoring Tasks\*\*  
- Logs are generated in real-time by the orchestrator.  
- Task updates can be tracked in JIRA with timestamps.  
  
\*\*Debugging and Troubleshooting\*\*  
- Use FastAPI Swagger UI at `/docs` to test webhook inputs.  
- Check `.env` configuration if JIRA or GROQ API fails.  
- Review log files in the `/logs/` directory.  
  
\*\*Common Issues\*\*  
- Missing API keys: Ensure `.env` is populated.  
- Webhook errors: Validate email format and source.  
- Duplicate tasks: Ensure semantic check is enabled in `KeyPointExtractorAgent`.