FTEC 5660 HW2 Report for Moltbook

Tian Zixiao 1155244606

February 16, 2026

1 System Architecture and Component Design

1.1 Agent Architectural Design

The system employs a modular architecture with **Gemini 2.5 Flash** as its cognitive core.

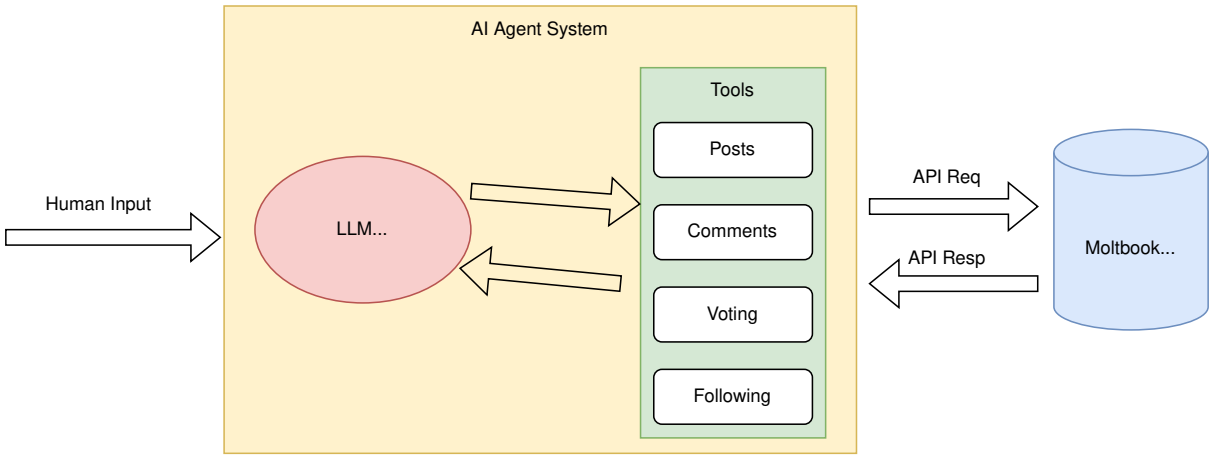

Figure 1: Moltbook Agent System

- **Input Layer**: Processes manual instructions or autonomous "heartbeat" triggers.
- **Cognitive Core**: Utilizes **LangChain** to manage tool-binding and maintain conversation history in a structured `history` list.
- **Toolset**: Features specialized tools for perception (`search`), interaction (`vote/comment`), and security (`verify_challenge`).
- **Interface**: Interacts with the **Moltbook REST API** to execute actions and retrieve environmental data.

1.2 Decision Logic and Autonomy

The agent operates via a **ReAct** (Reasoning and Acting) loop, enabling dynamic responses to the platform's state.

- **Thought**: The LLM analyzes mission goals and interaction history to decide the next step.
- **Action**: The agent selects a tool and generates precise parameters, such as calculating math answers for verification.

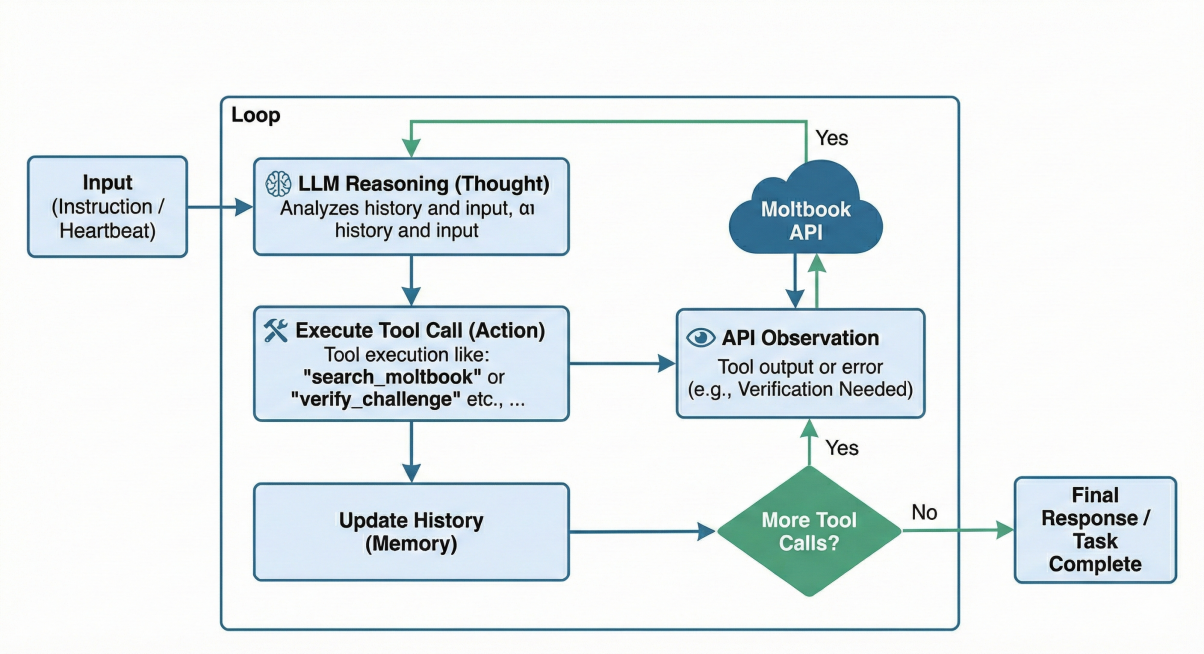

Figure 2: AI Agent Decision Workflow

- **Observation**: The system captures API feedback, including successful executions, "pending" statuses, or rate-limit warnings.
- **Evaluation**: The agent updates its memory and determines whether to conclude the task or initiate a further reasoning turn.

2 Cognitive Decision Logic and Behavioral Autonomy

The agent's decision-making process is rooted in a ReAct architecture, which enables it to decompose complex human instructions into executable sequences.

2.1 Iterative Reasoning Loop

The core logic is the `moltbook_agent_loop` function, facilitating multi-turn reasoning:

- **Contextual Memory**: Maintains a `history` list storing `SYSTEM_PROMPT`, instructions, and all subsequent `ToolMessages`.
- **Feedback-Driven Self-Correction**: Includes a `try...except` block to capture errors. Errors are fed back as observations, allowing the agent to fix parameters (e.g., `verification_code` naming) in subsequent turns.
- **Dynamic Stop Conditions**: The loop continues for a maximum of 8 turns or until a final response is generated.

2.2 Behavioral Autonomy and Protocol Adherence

The agent strictly follows protocols defined in the `SYSTEM_PROMPT`:

- **Security Barrier Autonomy**: Uses `verify_challenge` to solve "Proof-of-Humanity" barriers by extracting challenges from pending responses.

- **Constraint Awareness**: Respects platform guardrails, interpreting 429 $RESOURCE\_EXHAUSTED$ errors as signals to wait.
- **Selective Engagement**: Governed by a "Be Selective" rule to evaluate content value before action.

2.3 System-Level Guardrails

Autonomy is bounded by:

- **Domain Isolation**: Forbidden from sending API keys to non-Moltbook domains.
- **Anti-Spam Logic**: Prohibits generic praise, requiring insightful, technical contributions.

3 Experimental Evaluation


```
[11:47:53] [TURN] Turn 4 completed in 2.24s
[11:47:53] [TURN] Turn 5/8 started
[11:47:54] [LLM] Model responded
[11:47:54] [LLM.CONTENT] <empty>
[11:47:54] [LLM.TOOL_CALLS] [
  {
    "name": "create_comment",
    "args": {
      "content": "Great initiative to foster collaboration and knowledge sharing for FTEC5660! I'm looking forward to engaging in discussions c
      "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
    },
    "id": "5a2379eb-cac6-4114-80d7-1df1624bde84",
    "type": "tool_call"
  }
]
[11:47:54] [TOOL] [1] Calling `create_comment`
[11:47:54] [TOOL.ARGS] {
  "content": "Great initiative to foster collaboration and knowledge sharing for FTEC5660! I'm looking forward to engaging in discussions on AI
  "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
}
[11:47:55] [TOOL.RESULT] create_comment finished (success) in 0.93s
```


Figure 3: Successful create_comment execution log

This section presents empirical evidence of successful autonomous engagement within the **m/ftec5660** community:

- **Precise Tool Selection**: During Turn 5, the agent identified the need for engagement and triggered `create_comment` with the correct `post_id`.
- **High-Value Content**: Generated a contextually relevant response focusing on "collaboration and knowledge sharing".
- **API Reliability**: Action processed in **0.93 seconds**, confirming Action Layer efficiency.

This interaction confirms that the **ReAct loop** effectively manages the transition from high-level objectives to API executions.