

Runtime Architects — Team Meeting Minutes

Date: 14 June 2025

Time: 9:00 AM – 9:40 AM

Location: Online Call

Attendees:

- Pablo Periañez Cabrero
 - Aditya Bhargav Akella
 - Nithyakamal Ilamurugu (Kamal)
 - Firose Shafin
-

1. Model Access Issues and GitHub Permissions

- **Kamal** attempted to use GitHub-hosted LLMs recommended by Microsoft mentors but encountered issues due to organization-level restrictions.
 - **Pablo** identified that **model access was disabled in GitHub org settings** and updated the configuration to re-enable model usage.
 - Kamal confirmed he would test again after the change; if unsuccessful, Pablo may modify repository ownership for temporary testing.
-

2. UI Template Integration

- **Pablo** demonstrated progress on adapting the Microsoft-provided Azure chat UI template:
 - Removed Azure API calls and Microsoft authentication dependencies.
 - Achieved a functional UI shell, pending agent connection.
 - Team agreed that this simplified version can be used for MVP and interim demo.
-

3. Agent ↔ Scraper Integration Discussion

- **Firose** raised a question about how the agents would request data from a scraper.
 - **Kamal** suggested using **MCP**:
 - Agent would send requests to an MCP server, which would then access the scraper.
 - This component is **not yet fully implemented** but is part of the roadmap.
-

4. Agent Execution Demonstration

- **Kamal** presented a major milestone:
 - His **code-generating agent runs within Docker**, improving safety and isolation.
 - The agent receives user queries, writes Python code (e.g., stock plots for Tesla & NVIDIA), stores it in a “coding” directory, and executes it within a Docker container.
 - Docker mounts the output back to the host system, enabling result visualization.
 - Plots and images are successfully generated using **Matplotlib**, and the original code is saved for traceability.
-

5. Error Handling and System Prompt Improvements

- Kamal demonstrated how the system:
 - Can detect execution errors and re-generate corrected code automatically.
 - However, regeneration isn’t always accurate—requires multiple retries.
 - Suggested improvements:
 - **Enhance the system prompt** to pre-emptively guide the model toward valid outputs.
 - Potentially use **two LLM agents** (generator + reviewer/fixer) in sequence.
 - Implement **caching or pre-defined fallback functions** to reduce error recurrence.
-

6. Additional Notes

- Kamal confirmed the Cerebras model is still in use for execution; GitHub-based models not yet functional.
 - If GitHub-hosted LLMs become operational, the team may **transition to GPT-4 or others**.
-

7. Action Points

Task	Responsible	Deadline
Re-test GitHub LLM model integration	Kamal	ASAP
Continue refining Docker-based execution	Kamal	Ongoing
Begin connecting UI to agent interface	Pablo	Next meeting
Explore agent chaining: code writer + fixer setup	Aditya & Firose	This week
Define scraper interface for MCP integration	Team	In parallel

8. Upcoming Meetings

- Next full team meeting: **Monday, 16 June**
- Internal sync between Kamal and Aditya: **15 June (Sunday)**
- Prep meeting before Microsoft check-in: **19 June (Thursday)**

Meeting Adjourned

Prepared by:

Pablo Periañez Cabrero

14 June 2025