

VIDEO LINK

- <https://youtu.be/ZfyJIFs9uIM>



CONCEPTUAL ARCHITECTURE OF GEMINI CLI

GROUP #8 - THE TEAMM8TES

TEAM MEMBERS

- **Group Leader:**

- Christian Fiorino | Derivation Process, Prompt Engineer, AI Report, Lessons Learned

- **Presenters:**


- Ananya Kollipara | Sequence Diagrams
- Lillie Amos | Research, Introduction, Conclusion, Architectural Analysis

- **Group Members:**

- Arlen Smith | Research, External Interfaces
- Maia Turner | Research, Architectural Analysis
- Vivian Webster | Use Cases, Abstract



AGENDA

1. Architecture Overview
 2. Components & Dependencies
 3. Subsystems
 4. Evolution
 5. Derivation Process
 6. Alternative Architectures
 7. Use Cases
 8. Lessons & Limitations
 9. AI Report
 10. Conclusion
- 



ARCHITECTURE OVERVIEW

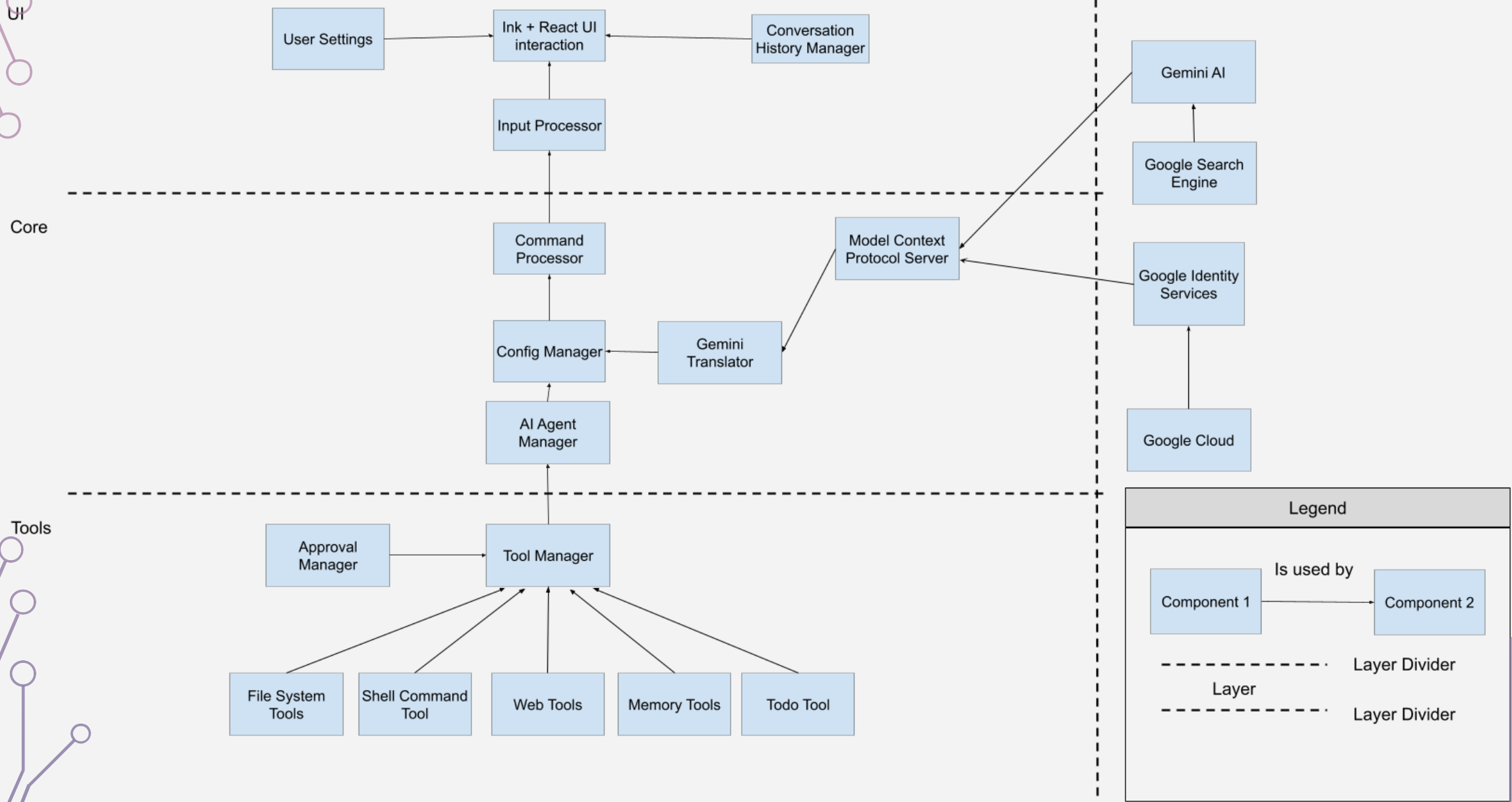
Architectural Styles:

- Server / Client
 - Server: Gemini
 - Client: Gemini CLI
- Layered
 - CLI
 - Core

-Client-

COMPONENTS & SUBSYSTEM INTERACTION

-Server-



EVOLUTION

- **Core:** Can be updated to add new features, fix bugs and support new versions of Gemini AI Models.
- **CLI:** Will stay in sync with updates to the core, prompts can be further clarified with updated information and keyboard shortcuts can be introduced.
- **Tools:** Through updates, new tools can be added, and existing tools can be edited.
- **Extensions:** Gemini CLI's capabilities can be evolved through user-created extensions found in an official Extensions gallery online.
- **Agent Skills:** Users can define specialized capabilities and domain-specific knowledge to give Gemini CLI on-demand expertise upon Gemini identifying a task matching the skill's description.

DERIVATION PROCESS

- Started with reviewing Gemini CLI Documentation and resources posted to OnQ.
- Created rough sequence and box-and-line diagrams to infer architectural styles from them.
- Inferred a **layered style** due to system components being partitioned, where each partition had certain responsibilities and relied on services from lower layers.
- Also decided on the system using a **client-server style** due to the API calls it uses to communicate servers on a network, and because the “cli” and “core” components split the system into a “client” and “server” side.

ALTERNATIVE ARCHITECTURES

Repository Style

- The Gemini LLM model could act as a repository of information.
- Allows the system to retain information from previous prompts, allowing for better responses.
- This style would require more robust infrastructure.
- Would make the system less modular, which goes against Gemini CLI's design principles.

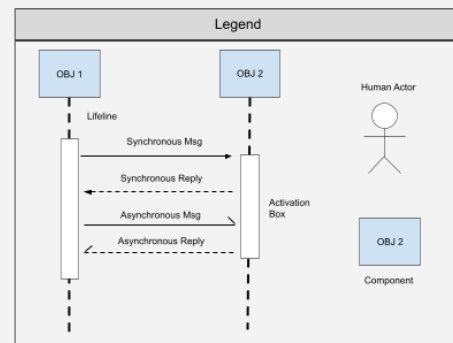
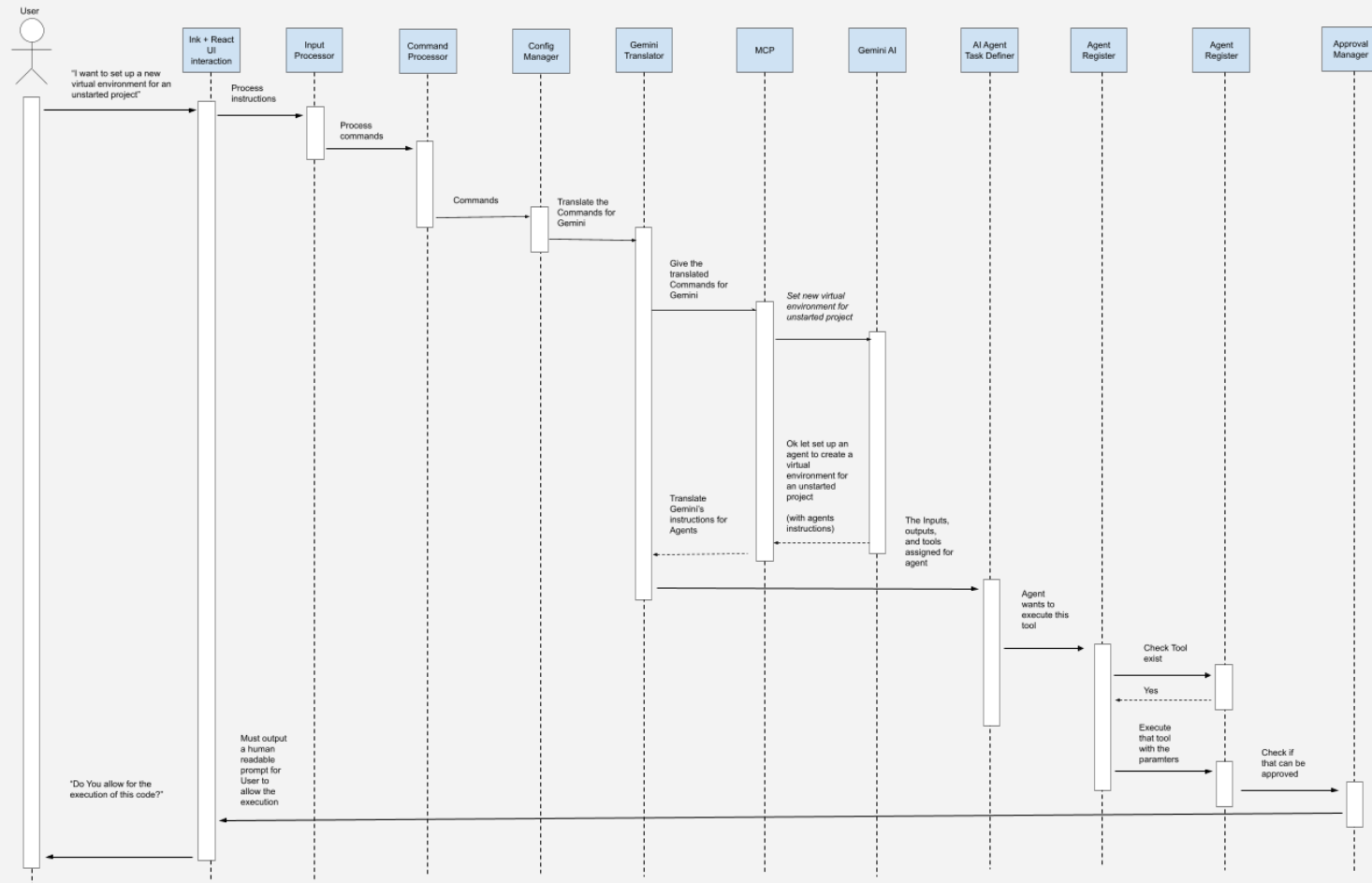
Object-Oriented Style

- Allows increased modularity as no inconsistencies occur when changing component implementations.
- Does not encompass the whole architecture of Gemini CLI, unlike the client-server and layered styles.

USE CASE #1

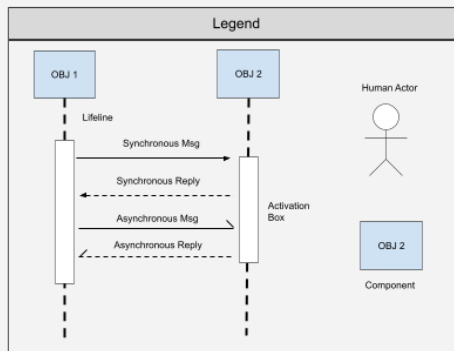
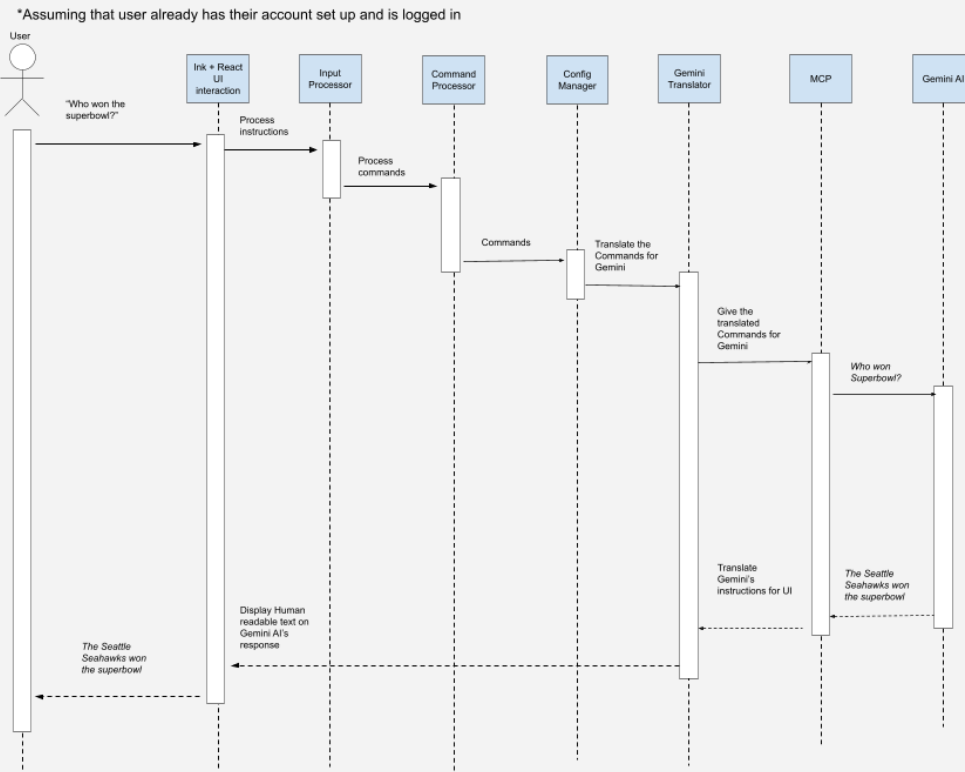
The user wants to set up a virtual environment for a new project.

*Assuming that user already has their account set up and is logged in



USE CASE #2

The user wants to research a concept using Gemini CLI



LESSONS & LIMITATIONS

- It is vital to include security in architectures that affect sensitive information such as a user's system.
- Starting projects early and setting clear soft deadlines is essential to ensuring the project is completed on time.
- Ensuring everyone has a clear and consistent understanding of the requirements allows there to be no misunderstandings with tasks that need to be completed.
- A key limitation was that a lot of non-official documentation did not provide a deeper level of detail for understanding how Gemini CLI works.

AI REPORT

- Utilized Gemini v3 Fast as our virtual member.
- LLM was assigned the task of searching for resources on the internet.
- Prompting strategy involved establishing a persona, asking for a specified output, and cross-referencing the LLM's response with its resources to ensure accuracy.
- LLM did not provide a large impact on group dynamics, its contribution rating being (10%).
- For future projects, we will allow the LLM to perform more tasks and ensure that all group members will interact with the LLM.

CONCLUSION

- Gemini CLI is a modular system that utilizes a layered and client-server style.
- Acts as a bridge between classic command line terminal and the modern LLM prompting interface.