VIDEO LINK

- https://youtu.be/HOcNrt4C9Ew

CONCRETE ARCHITECTURE OF GEMINI CLI

GROUP #8 - THE TEAMM8TES

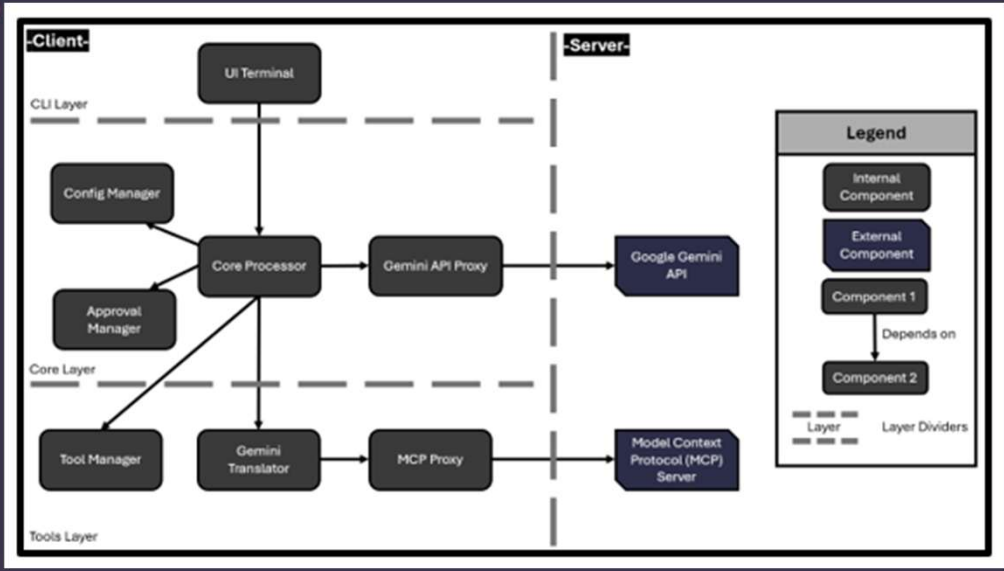
TEAM MEMBERS

- **Group Leader:**
  - Christian Fiorino | Conceptual Architecture Revamp, Second-Level Concrete Architecture, Inner Subsystem Description, Derivation Process, Prompt Engineer, AI Report
- **Presenters:**
  - Ananya Kollipara | Conceptual Architecture Revamp
  - Lillie Amos | High-Level System Concrete Architecture, High-Level System Reflexion Analysis, Introduction, Conclusion
- **Group Members:**
  - Arlen Smith | High-Level System Concrete Architecture, Lessons Learned
  - Maia Turner | High-Level System Concrete Architecture, High-Level System Description
  - Vivian Webster | Second-Level Concrete Architecture, Use Cases, Inner Subsystem Reflexion Analysis, Abstract

AGENDA

1. Conceptual Architecture Breakdown
2. High-Level Concrete Architecture
3. Second-Level Concrete Architecture
4. High-Level Reflexion Analysis
5. Second-Level Reflexion Analysis
6. Derivation Process
7. Use Cases
8. Lessons & Limitations
9. AI Report
10. Conclusion

CONCEPTUAL ARCHITECTURE BREAKDOWN

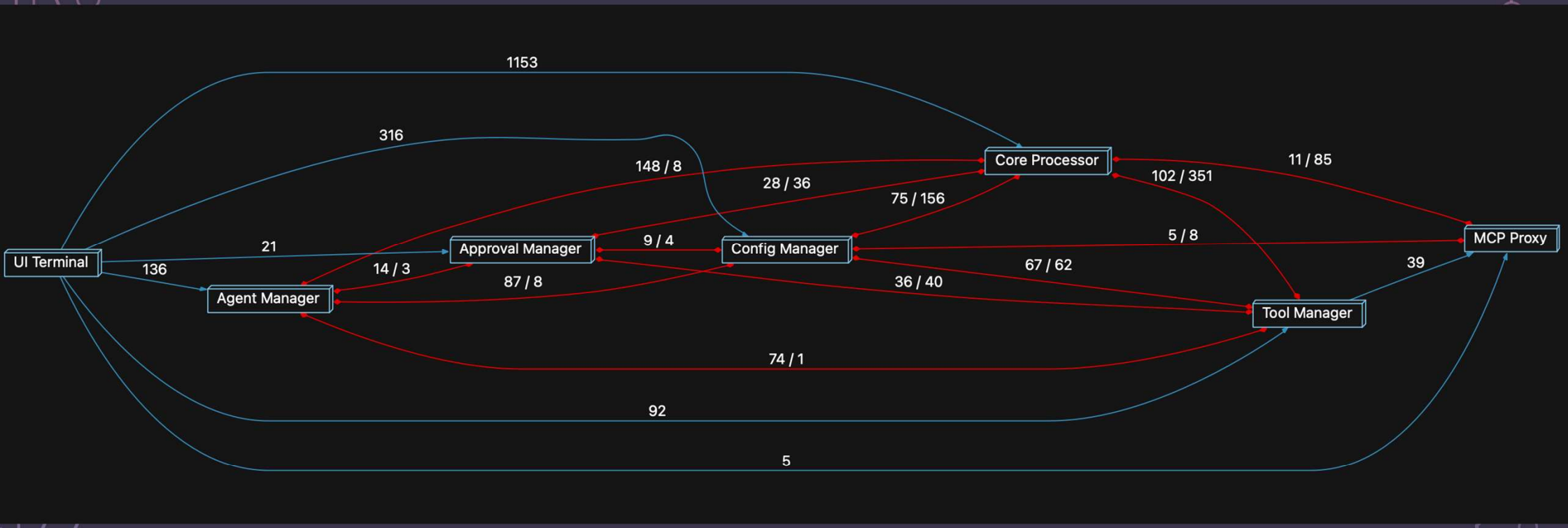


A2 Updated Conceptual Architecture
of
Layered & Client-Server

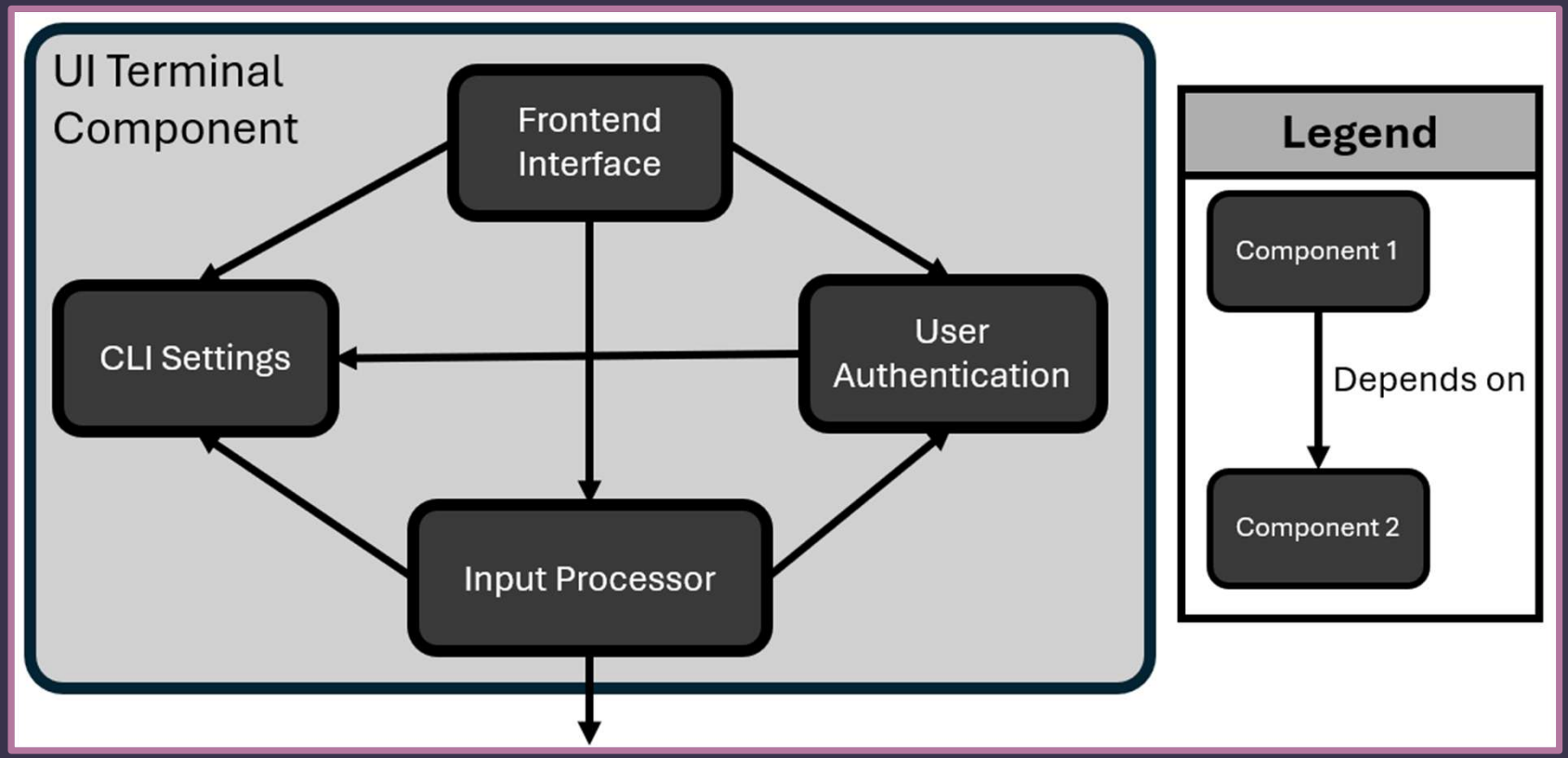
- Components merged together to simplify the processes as it was too low level:
  - All original subcomponents of the CLI as [UI terminal]
  - All tools within the tool layer as [Tool Manger]
  - [Gemini AI] & [Google Search Engine] as [Google Gemini API]

- Created [Core Processor] component to be dependent on the Tools layer components
  - [Gemini Translator] and [MCP Proxy] have been moved to the Tools Layer as they are optional tools that may be used by the CLI.
  - Created [Approval Manager] as we believe it will be used by [Core Processor]
  - Removed the [AI Agent Manager] component as the [Core Processor] component will handle the AI task delegations.
  - Added Gemini API Proxy for connected communication to [Google Gemini API]

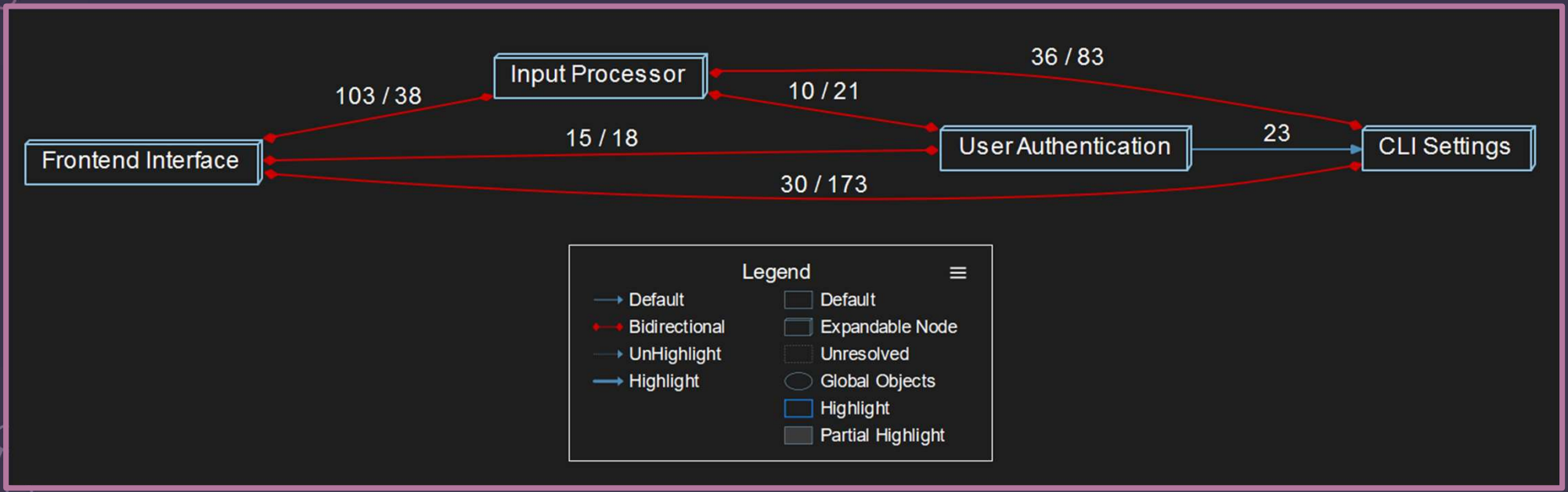
HIGH-LEVEL CONCRETE ARCHITECTURE



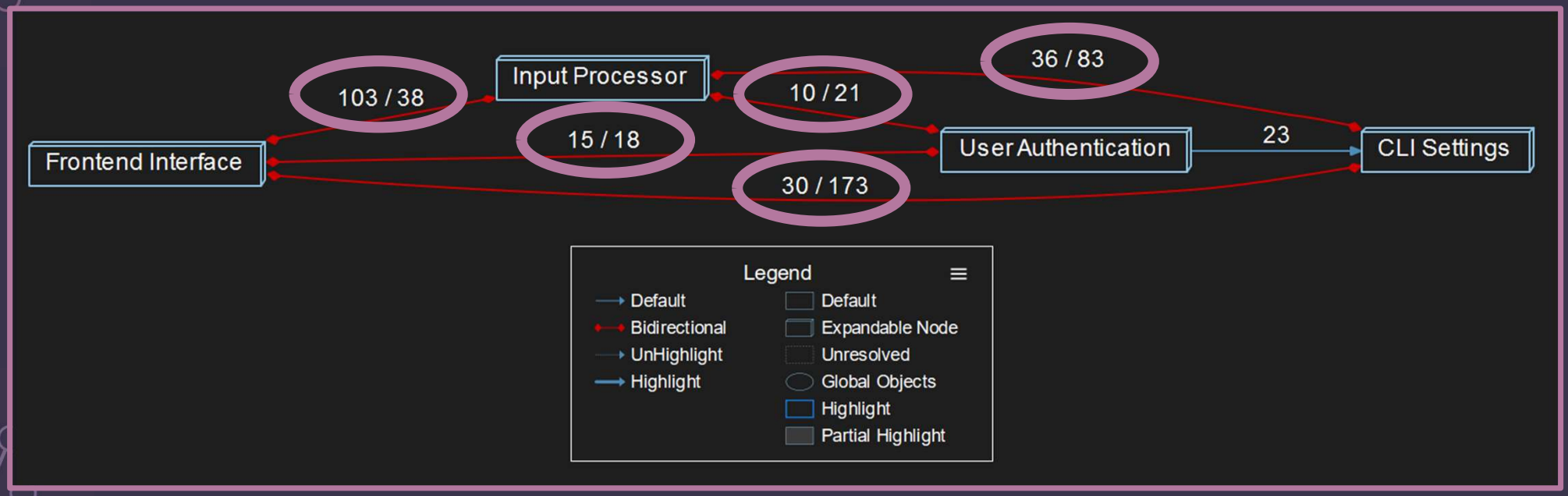
SECOND-LEVEL CONCEPTUAL ARCHITECTURE



SECOND-LEVEL CONCRETE ARCHITECTURE



SECOND-LEVEL CONCRETE ARCHITECTURE

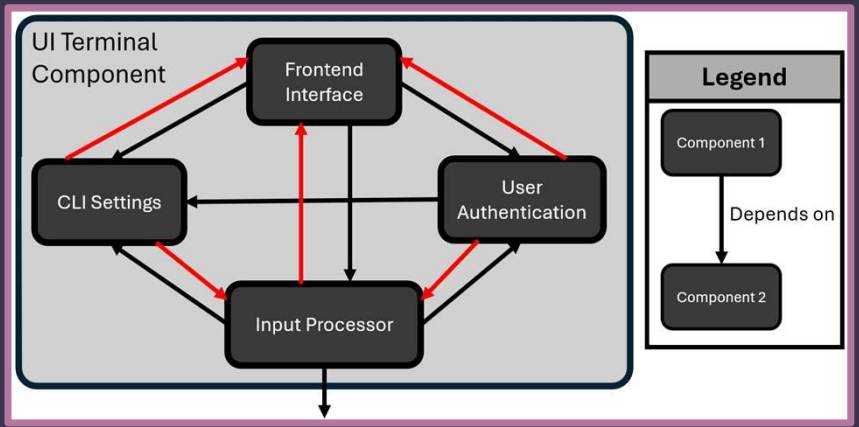


HIGH-LEVEL REFLEXION ANALYSIS

- Many bidirectional dependencies were found in the concrete architecture, diverging from our conceptual architecture.
- Primary divergence example is the dependency between the “Agent Manager” and “Tool Manager” components.
- Using GitHub blame, found that agents are able to be wrapped and exposed as tools, explaining the dependency.

SECOND-LEVEL REFLEXION ANALYSIS

- All subcomponents now reliant on Frontend Interface
- Authentication and Settings now reliant on Input
- Components much more interlinked than expected conceptually

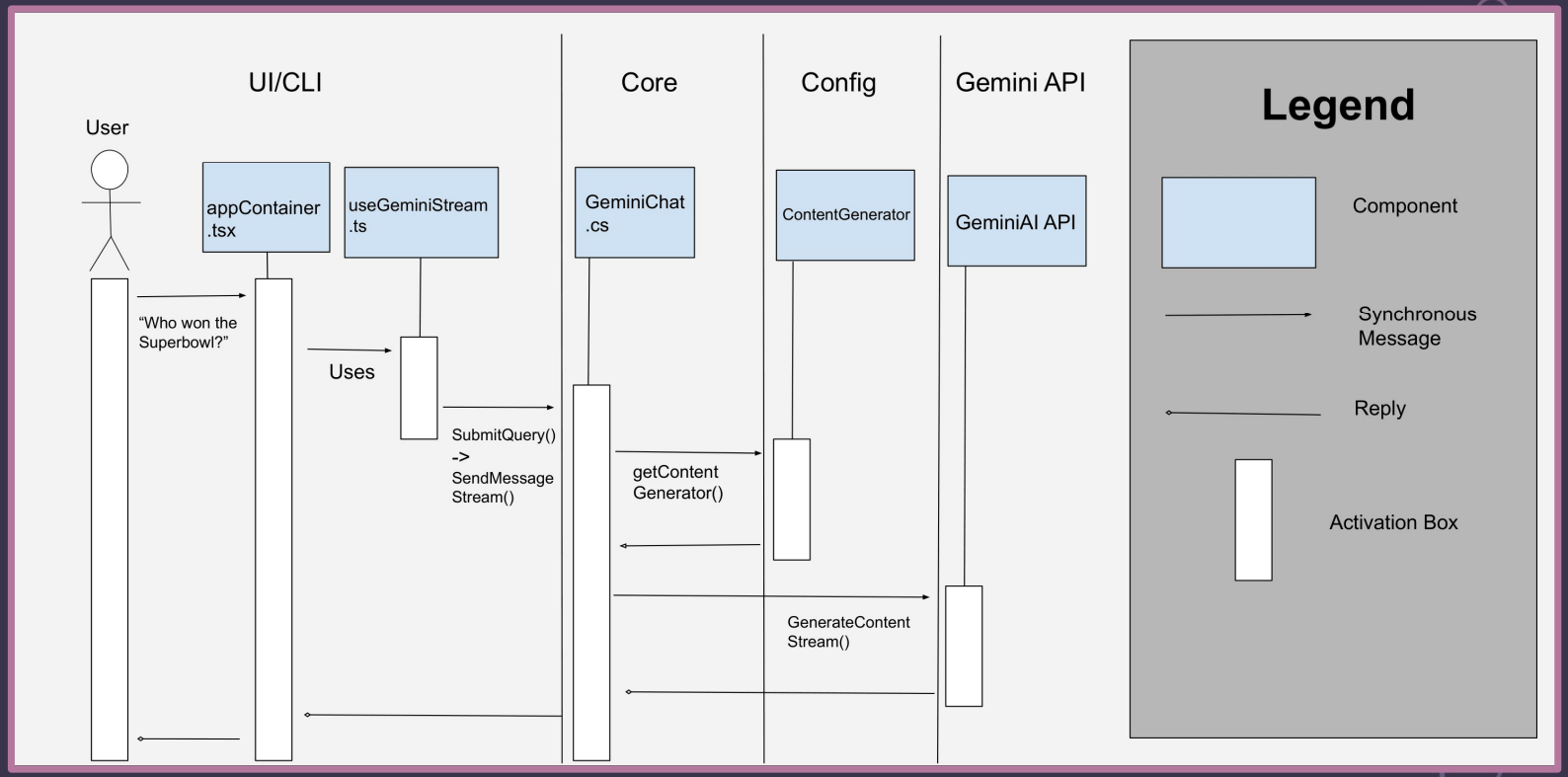


DERIVATION PROCESS

- Started with updating the Conceptual Architecture based on the global feedback version found on OnQ.
- Divided into subgroups to work on the Concrete Architecture based on each layer of the Conceptual Architecture.
- Assigned code files using *Understand*, and considered alternative Concrete Architectures such as deciding where to place files related to the UI themes.
- Used the finalized Concrete Architecture to perform the reflexion analyses and updated our Conceptual Architecture accordingly.

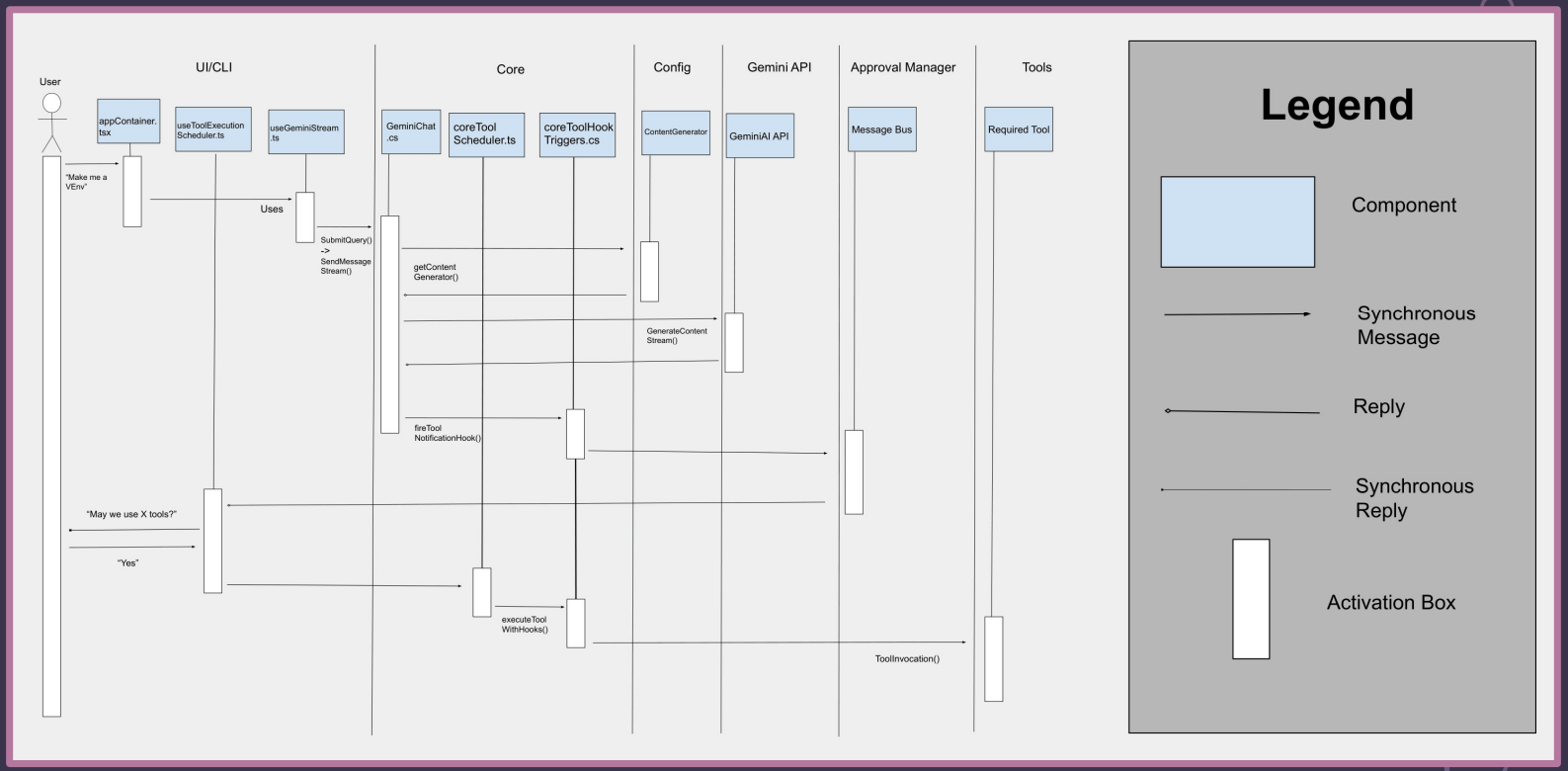
USE CASE #1

The user wants to know who won the Superbowl



USE CASE #2

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



LESSONS & LIMITATIONS

- Architecture vs Implementation - We learned to use the Understand tool as a basis for comprehending a more abstract architecture rather than deriving an architecture directly from the system's implementation.
- The Value of Workload Division – Dividing up tasks in our group allowed us to ensure that our group had a well-rounded understanding of both architectures we derived more quickly.

- One Key Limitation – Gemini CLI's official documentation was recently restructured, making it more difficult to review prior conclusions about the system.

AI REPORT

- Utilized OpenAI’s ChatGPT-5.3 model as our virtual member.
- LLM was assigned the tasks of creating a PowerPoint presentation and evaluating our report based on the rubric.
- Prompting strategy involved establishing a persona, providing it relevant information such as a tailor-made version of our report, and cross-referencing the LLM’s response with the assignment rubric to ensure accuracy.
- LLM did not provide a large impact on group dynamics and was unable to complete one of its task, its contribution rating being (5%).
- For the final assignment, we plan to integrate the LLM earlier into the process by having it work on tasks directly related to Gemini CLI’s architecture.

CONCLUSION

- Process of mapping files using *Understand* provided a better understanding of the Gemini CLI architecture.
- The concrete architecture will serve as the backbone for completing the final report.