# Complete Current Implementation

**Toolchain & Configuration**

This appendix serves as a centralized log of all external tools, services, and configurations required for the Symposium project.

**1. Core Language**

* **Tool:** Python
* **Version:** 3.13
* **Purpose:** The primary programming language for the entire project.
* **Source:** [https://www.python.org/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.python.org%2F)
* **Location:** System-wide installation.

**2. Development Environment**

* **Tool:** Visual Studio Code (VS Code)
* **Purpose:** The official Integrated Development Environment (IDE) for the project. Used for all code editing, debugging, and terminal operations.
* **Source:** [https://code.visualstudio.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fcode.visualstudio.com%2F)
* **Location:** Standard application installation.
* **Configuration:** The project is managed via a Symposium.code-workspace file located in the Github directory.

**3. Version Control**

* **Tool:** Git
* **Purpose:** Manages the source code history and versioning.
* **Source:** [https://git-scm.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgit-scm.com%2F)
* **Location:** System-wide installation.
* **Remote:** [https://github.com/gotonickpappas/Symposium](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgithub.com%2Fgotonickpappas%2FSymposium)

**4. Cloud Sync Utility**

* **Tool:** rclone
* **Version:** v1.71.0
* **Purpose:** Syncs the local project directory to a shared Google Drive folder for AI peer review, automatically excluding sensitive files and unnecessary directories.
* **Source:** [https://rclone.org/](https://www.google.com/url?sa=E&q=https%3A%2F%2Frclone.org%2F)
* **Location:** C:\Users\gotoz\Tools\rclone\rclone.exe (Added to system PATH).
* **Configuration:** A remote named gdrive was configured via the rclone config command to securely connect to the Director's Google Drive account.

**5. API Keys & Services**

* **Purpose:** Provide access to the various Large Language Models that act as the "raw talent" (Tier 1) for the Symposium.
* **Location:** All API keys are stored locally in the .env file in the project root, which is explicitly excluded from all repositories and cloud syncs via .gitignore.
* **Services:**
  + **Google AI (Gemini):** [https://ai.google.dev/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fai.google.dev%2F)
  + **Anthropic (Claude):** [https://www.anthropic.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.anthropic.com%2F)
  + **Groq:** [https://groq.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgroq.com%2F)
  + **Mistral AI:** [https://mistral.ai/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmistral.ai%2F)
  + **Cerebras:** [https://www.cerebras.net/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.cerebras.net%2F)

**LangSmith:** [https://www.langchain.com/langsmith](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.langchain.com%2Flangsmith)

**Project state**  
The project is contained entirely within the C:\...\Github\Symposium\ folder:

/Symposium

├── .git/

├── symposium\_env/

├── src/

│ ├── models/

│ │ ├── tools/

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── calculator.py *# Fast arithmetic tool*

│ │ │ └── python\_executor.py *# Whitelist-based Python execution*

│ │ ├── \_\_init\_\_.py

│ │ ├── agent.py *# Agent class, ModelProvider enum, factory logic*

│ │ ├── symposium.py *# Agent registry and initialization*

│ │ └── workflow.py *# LangGraph workflow creation with provider quirks*

│ ├── controllers/

│ │ ├── \_\_init\_\_.py

│ │ └── orchestrator.py *# Message routing, agent selection logic*

│ ├── views/

│ │ ├── \_\_init\_\_.py

│ │ └── cli.py *# Testing interface*

│ └── symposium\_manager\_v2.py *# Clean entry point using modular MVC architecture*

├── .env *# API Key storage*

├── .gitignore

├── publish.bat *# Automated GitHub + Google Drive publishing*

├── requirements.txt

└── run.bat *# Convenience script*

**Core Implementation Status:**

1. All five agents (Gemini, Claude, Groq\_Assistant, Mistral\_Assistant, Cerebras\_Assistant) operational
2. Dual tool system: calculator (fast arithmetic) + python\_executor (complex calculations with whitelist security)
3. Provider-specific quirks preserved and contained within workflow.py
4. Clean 15-line entry point replacing 200+ line monolith
5. Zero functionality regression, complete architectural improvement

Immediate Next Step

**[CRITICAL] Determine the next step of the Yak Shaving:** We should probably figure out how to proceed with a way where we can all three of us see the UI conceptually wireframe or otherwise likely using Claude’s tool, and translation to JSON for Gemini. At this point it seems we need at least SOME UI to proceed.

**Test Python Executor:** Verify whitelist-based Python execution with financial calculations using Decimal precision. The infrastructure is complete for complex task routing between fast calculator and secure Python execution.

**Next Expansion:** Google search tool integration for research capabilities.