

Final Project Report – LLM Council Local Deployment

Distributed Multi-LLM Reasoning with Local Inference

Sacha CROCHET
Eliott COUTAZ
Julien DE VOS
Mathieu COWAN
Adrien DE MAILLY NESLE

ESILV – A5

Academic Year 2025–2026

Contents

1	Git Repository	2
2	Context	2
3	Original Repository Description	3
4	Project Objectives	3
5	Project Contributions	4
6	Backend Architecture and Models	4
6.1	Overall Architecture	4
6.2	Council Models and Technical Choices	4
6.3	Chairman Model	5
7	Dockerized Implementation	5
8	Solution Evaluation	5
9	Generative AI Usage Statement	6
10	Future Work and Optimizations	6
11	Conclusion	7

1 Git Repository

The source code of the *LLM Council Local Deployment* project is available at the following URL:

<https://github.com/matcoc0/LLM-Council-Local-Deployment>

The Git repository contains the complete implementation of the project, including back-end services, a frontend demonstration interface, configuration files, and documentation. The main elements of the repository are:

- **README.md** Provides a global overview of the project, installation instructions, environment configuration, and steps to run the LLM Council demonstration.
- **Backend source code** Python files implementing the core logic of the LLM Council, including:
 - orchestration of the three-stage workflow (first opinions, review and ranking, chairman synthesis),
 - REST API endpoints enabling communication between distributed services,
 - configuration of local LLM models and service URLs.
- **Frontend application** A lightweight user interface designed to demonstrate the behavior of the LLM Council, allowing inspection of individual model responses, review rankings, and the final synthesized answer.
- **Configuration files** Environment and configuration files defining model identifiers, service endpoints, and execution parameters required for local deployment.
- **Documentation** Additional README files describing the system architecture, design choices, and usage instructions.

2 Context

Large Language Models (LLMs) are commonly used in isolation, often through proprietary cloud-based services. While this approach is easy to deploy, it presents several limitations, including dependence on external infrastructures, limited transparency regarding intermediate reasoning steps, and a lack of diversity in generated responses.

The *LLM Council* concept, initially proposed by Andrej Karpathy, addresses these limitations by introducing a collaborative architecture. Multiple language models independently answer a user query, subsequently review and rank the responses produced by other models, and finally, a dedicated model referred to as the *Chairman* synthesizes all information into a single final answer.

The original implementation relied on cloud-based APIs. The objective of this project is to refactor and extend this approach in order to design a **fully local, distributed, and autonomous LLM Council**.

3 Original Repository Description

The original repository provided an initial implementation of the LLM Council concept, including the logical stages of the workflow (multiple response generation, cross-review, and final synthesis). However, this implementation suffered from several limitations:

- Dependence on cloud-based services (OpenRouter)
- Lack of a truly distributed deployment
- Weak separation of model responsibilities
- Architecture not well suited for local execution

As a result, the initial project primarily served as a proof of concept, without guarantees of reproducibility or independence from external services.

4 Project Objectives

The main objectives of this project are:

- Remove all dependencies on cloud-based APIs
- Deploy all LLMs locally
- Design a distributed architecture based on REST communications
- Clearly separate the roles of Council models and the Chairman model
- Implement a complete three-stage workflow:
 - First opinions
 - Review and ranking
 - Chairman synthesis
- Provide an interface allowing users to inspect intermediate outputs

5 Project Contributions

The main contributions of this project are:

- Complete refactoring of the backend to support local LLM execution
- Implementation of a modular and distributed architecture
- Development of an orchestrator managing the entire Council workflow
- Explicit separation between Council LLMs and the Chairman LLM
- Development of a demonstration frontend to visualize all stages
- Clear and structured documentation facilitating setup and demonstration

6 Backend Architecture and Models

6.1 Overall Architecture

The backend relies on a modular architecture composed of multiple independent services communicating through REST APIs:

- Council LLMs: generation of initial responses and participation in the review stage
- Chairman LLM: final synthesis only
- Flask API: orchestration of the different workflow stages
- Storage of conversations and intermediate outputs

This organization enables distributed execution across multiple machines and facilitates future system extensions.

[Global backend architecture]

6.2 Council Models and Technical Choices

The models composing the *LLM Council* were selected to ensure diversity of reasoning while remaining compatible with local execution on limited hardware resources.

- **LLaMA 3.2 (1B)**
A lightweight model favoring fast inference. It is used to produce concise initial responses, contributing to low overall system latency.

- **Gemma 3 (4B)**

A mid-sized model providing more structured reasoning capabilities. It produces more detailed responses while maintaining a reasonable computational cost.

- **Qwen 2.5 (1.5B)**

A compact model with solid general-purpose performance. It contributes to diversity in phrasing and reasoning approaches.

Each Council LLM runs as an independent service exposing a REST API.

6.3 Chairman Model

- **DeepSeek-R1 (7B)**

A model dedicated exclusively to final synthesis. Its stronger reasoning capabilities make it suitable for aggregating responses and rankings produced by the Council. It is deployed as a separate service, in accordance with project requirements.

7 Dockerized Implementation

The project is designed to be executed reproducibly through a clear separation of components. Configuration relies on environment variables and independent services, facilitating local deployment and controlled demonstrations.

This approach enables:

- simplified service startup
- clear dependency isolation
- future extension to real multi-machine deployments

[Running services and containers]

8 Solution Evaluation

The evaluation of the solution is based on a complete functional demonstration:

- Generation of initial responses by Council LLMs
- Anonymous review and ranking phase
- Final synthesis produced by the Chairman
- Visualization of intermediate outputs through the frontend

The system satisfies the following criteria:

- Fully functional end-to-end workflow
- Clear and modular architecture
- Effective separation of responsibilities
- Usable and clear documentation

[Complete workflow execution in the user interface]

9 Generative AI Usage Statement

Generative AI tools were used in this project in a limited and transparent manner.

- **Frontend development:** Generative AI was used to assist in the creation of the frontend interface, whose sole purpose is to demonstrate and visualize the backend system.
- **Documentation support:** Generative AI was used to help structure and optimize README files and technical documentation.

All generated content was systematically reviewed, validated, and adjusted by the team. All backend logic, architectural decisions, orchestration mechanisms, and optimizations were designed and implemented by the project team.

10 Future Work and Optimizations

Several improvements can be considered:

- Deployment across multiple physical machines
- Advanced monitoring (latency, model availability)
- Performance dashboards
- Graphical visualization of the Council workflow
- Dynamic management of Council models
- Improved ranking and synthesis strategies

11 Conclusion

This project demonstrates the feasibility of a fully local, distributed, and transparent *LLM Council*. By removing all cloud dependencies and implementing a modular architecture, the proposed solution enables greater reasoning diversity, full reproducibility, and improved understanding of internal mechanisms.

Beyond its functional aspects, this project represents a concrete experience in the design and deployment of distributed artificial intelligence systems.