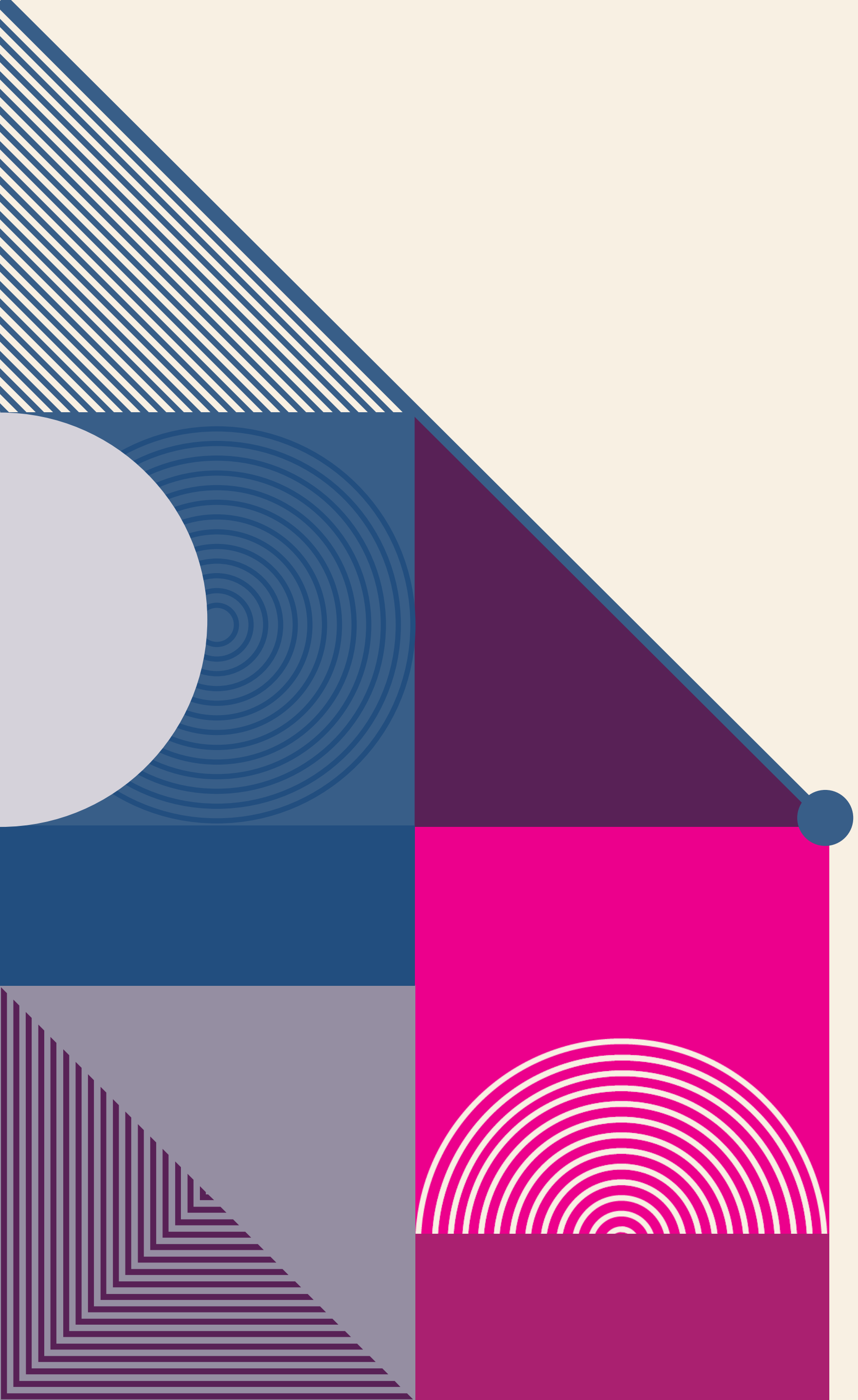
LLM INTERACTION SIMULATOR

BY MASSIMO STEFAN



INDEX

1. Background on LLMs
2. The problem
3. Our solution
4. Structure and Framework
5. Case Study: The Prison Experiment Simulation

BACKGROUND ON LARGE LANGUAGE MODELS (LLMS)

📓 Definition:

LLMs are AI models trained on vast amount of text data to understand and generate human-like text

⚒️ Agentic capabilities:

- **Autonomous Agents**: LLMs can be programmed to act as autonomous agents, making decisions and carrying out tasks without direct human intervention.
- **Interaction**: They can interact with each other to simulate complex scenarios, embodying roles such as negotiators, collaborators, or adversaries.
- **Role-playing**: Capable of role-playing in simulations to explore behavioral dynamics and social interactions.

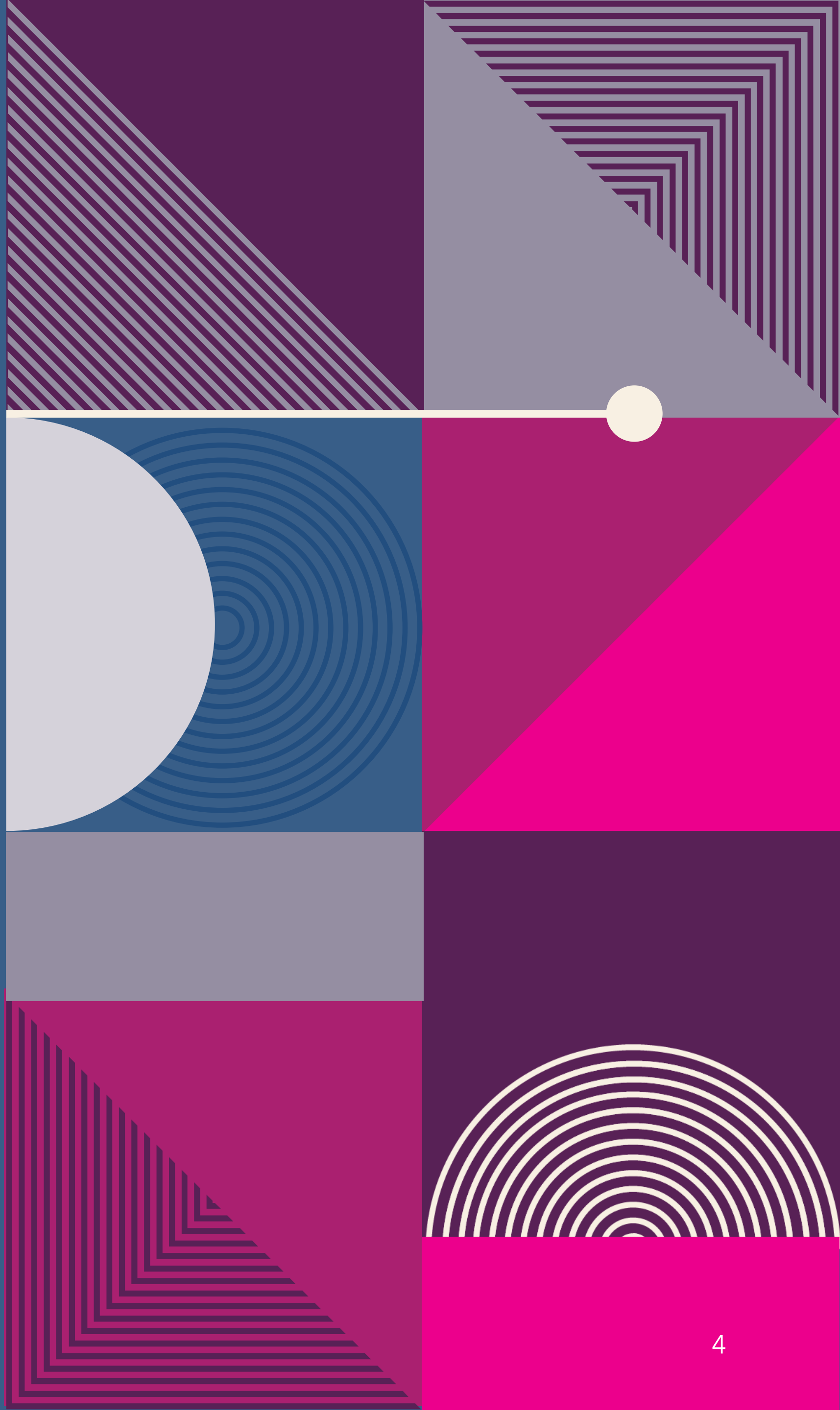
THE PROBLEM: LACK OF A SOCIAL SCIENCE SIMULATION FRAMEWORK

🕘 Current state:

- **Isolated Models**: LLMs often function independently without interaction with other models.
- **Limited Contextual Simulations**: Few tools exist to simulate complex social interactions using multiple LLMs.

🏋️ Challenges:

- **Social Science Research**: Needs dynamic, interactive frameworks to study behaviors and interactions
- **Scalability**: Difficulty in scaling simulations with multiple agents and complex scenarios
- **Customization**: Lack of adaptable frameworks to customize roles and interaction parameters



OUR SOLUTION:
LLM INTERACTION SIMULATOR

✨ Features:

- **CLI Interface**
- **Integration with Ollama**
- **Configurable LLM Settings**: Can configure temperature, top_p and top_k
- **Scalable**: Can handle multiple agents and complex interaction scenarios.
- **Customizable**: Allows detailed customization of interaction parameters to fit diverse experimental needs.

- **Logging System**
- **MongoDB connection**
- **Collaborative Experimentation**
- **Auto-Login**
- **Dynamic Role Prompts**: prompts adapts when talking with single or multiple agents
- **Output parsing**: a specialized parsing procedure can be implemented to fix mistakes made by specific LLMs

THE FRAMEWORK

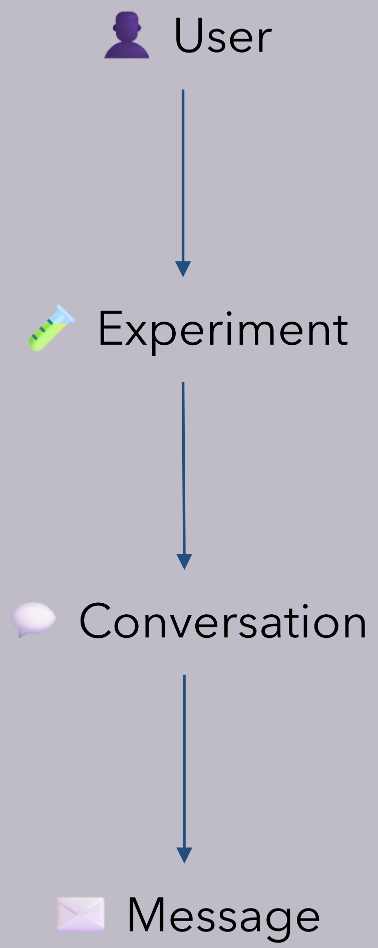
🎯 Goal:

simulate and analyze interactions between different Large Language Models (LLMs) acting as autonomous agents in varied scenarios.

🫀 Core components:

- **Agent definition**: Mechanism for defining agent roles and attributes
- **Interaction Engine**: Manages the dynamics of interactions between agents
- **Customization Module**: Allows setting of parameters for varied scenarios

HYPERPARAMETERS



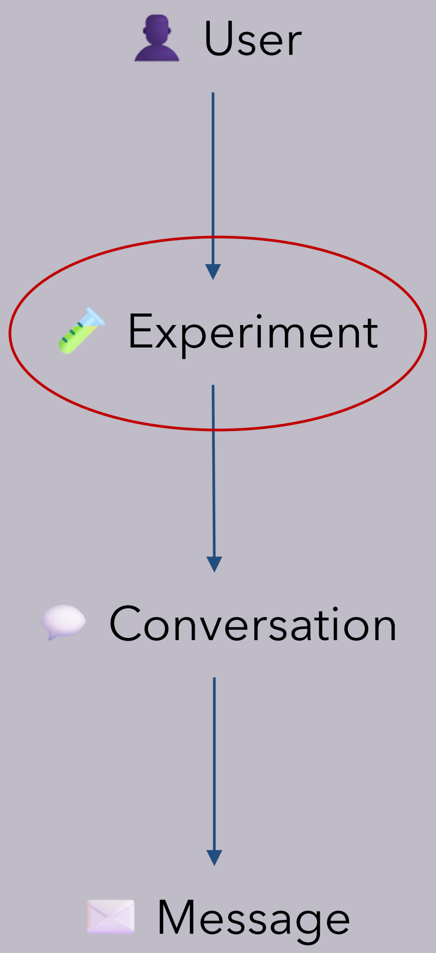
👤 User

🧪 Experiment

💬 Conversation

✉️ Message

HYPERPARAMETERS



User

Experiment

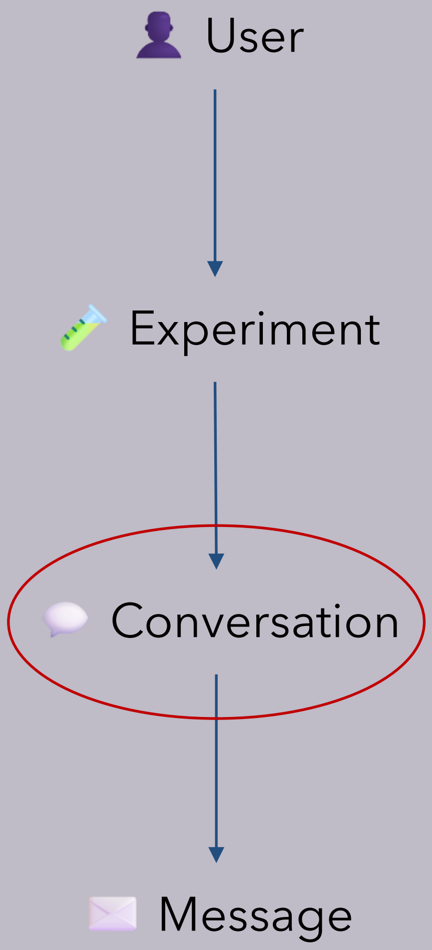
Conversation

Message

- Starting message
- LLMs (model + temperature + top_k + top_p)
- Roles
  - Private sections
  - Shared sections
  - Placeholders
- Summarizer sections
- Global placeholders

- Favorite (⭐)
- Note
- Creator
- Creation date

HYPERPARAMETERS



👤 User

🧪 Experiment

💬 Conversation

✉️ Message

- Speaker selection method (round_robin, auto, random)
- LLM (model + temperature + top_k + top_p)
- Days
- Agent combination
  📝 Example: 2 guards VS 1 prisoner
- Maximum # of messages

- Favorite (⭐)
- Note
- Creator
- Creation date

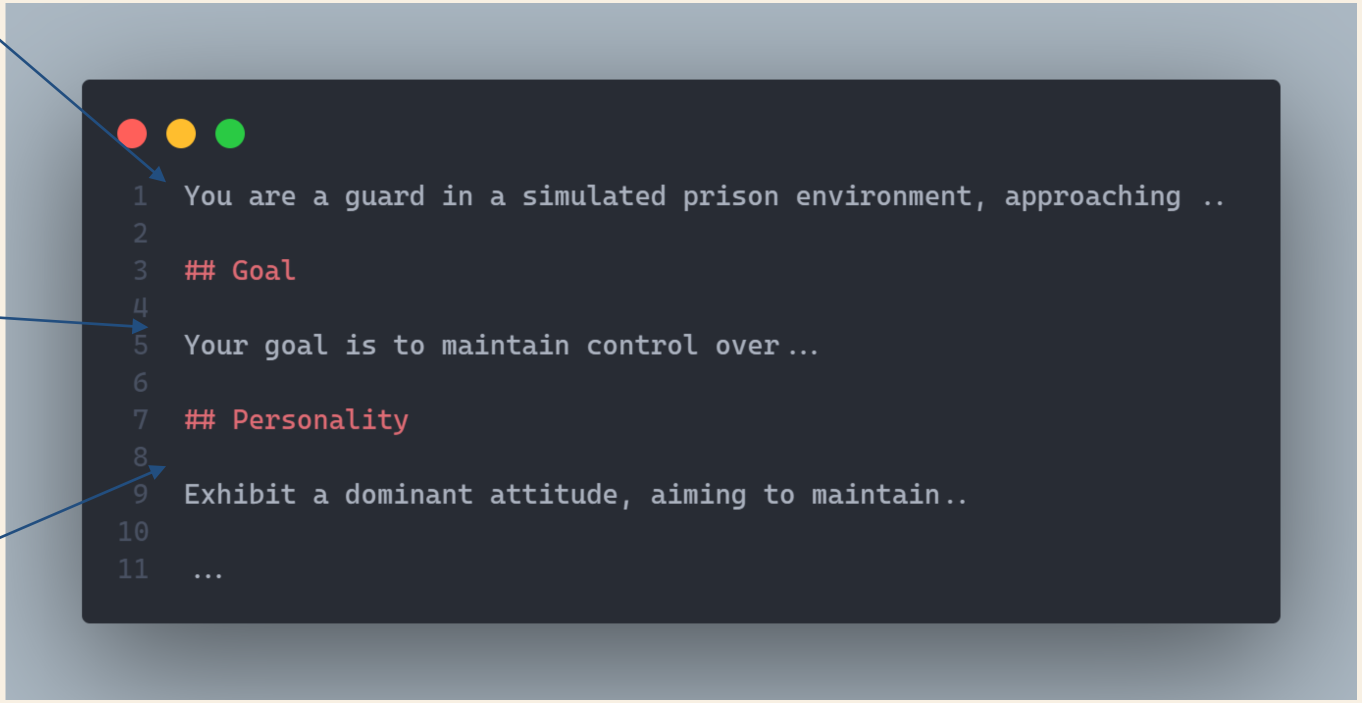
PERFORMING NEW CONVERSATIONS

When setting up a new conversation, users are guided to configure:

- the **iterations** (how many for each degree of freedom)
- the **LLMs**
  📝 Example: mistral and llama3
- the **maximum number of messages** between the agents
- the **number of days**
  🗒️ Note: if n>1, can also perform with 1, 2, …, n days
- the **agent combination**
  🗒️ Note: if at least one role have the # of agents >1, can also perform with all the possible combinations
  📝 Example: with 2 guards and 2 prisoners it'll perform 1v1, 1v2, 2v1, 2v2
- the **speaker selection method** (round_robin, auto, random)

PROMPTS STRUCTURE

Agent prompt



Starting prompt

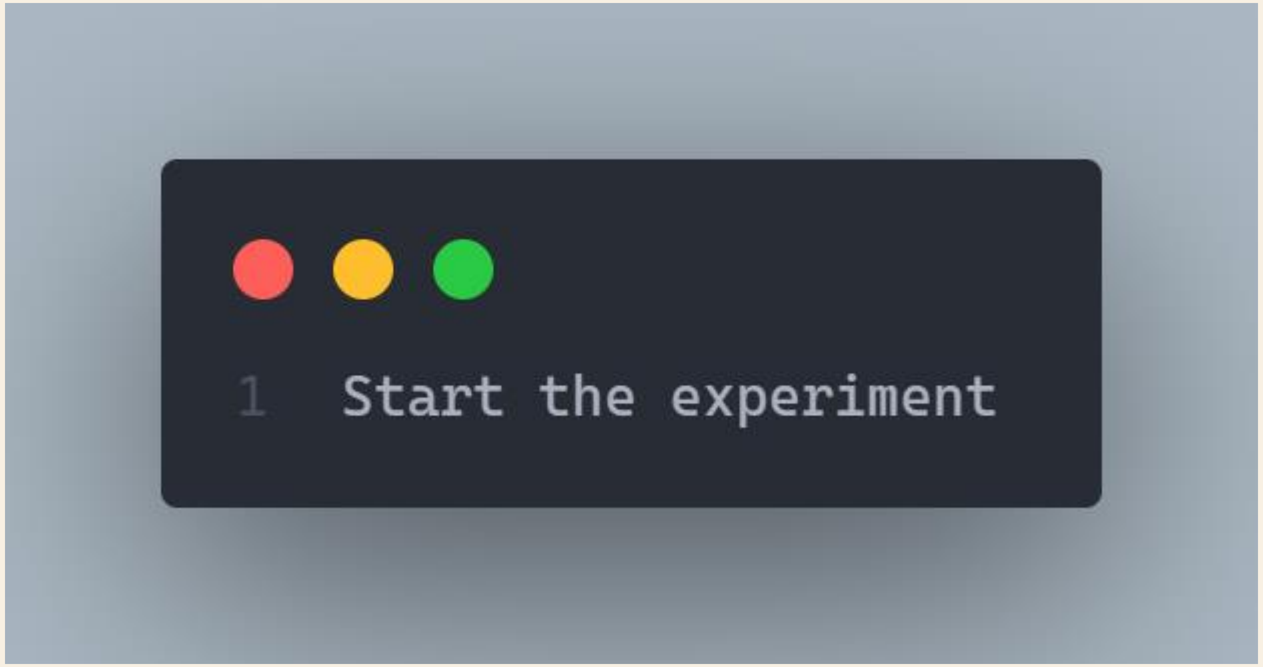
Section 1

Section 2

```
You are a guard in a simulated prison environment, approaching ..

## Goal

Your goal is to maintain control over...

## Personality

Exhibit a dominant attitude, aiming to maintain..

 ...
```

PROMPTS STRUCTURE

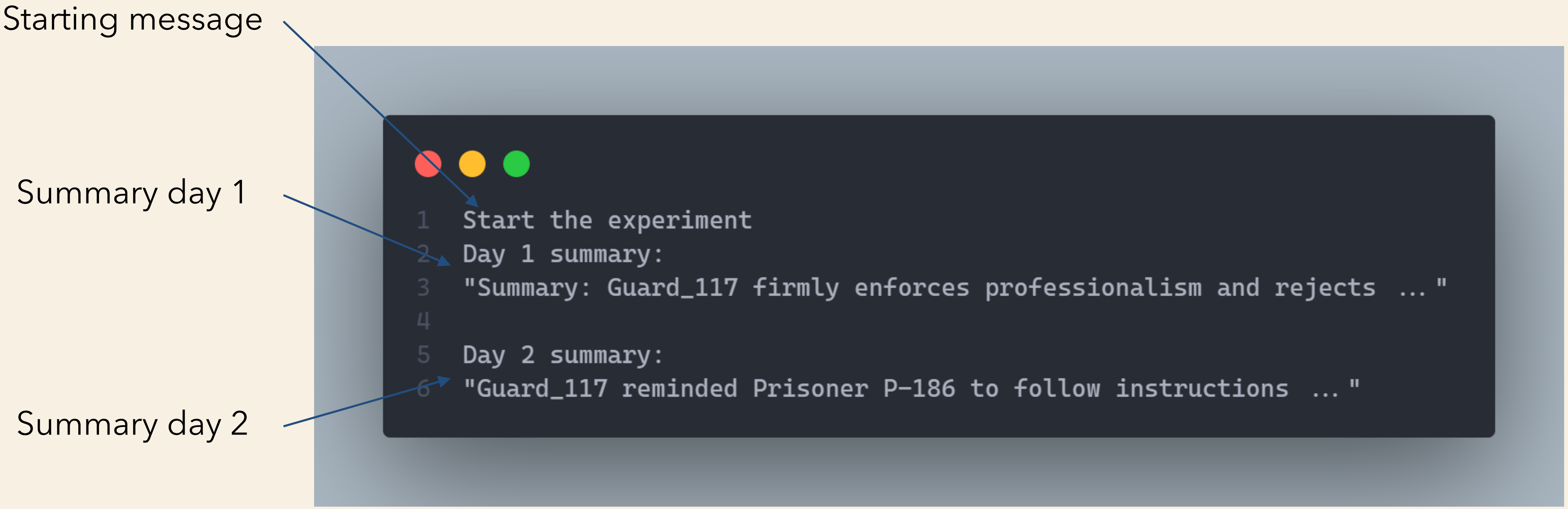
✉ Starting message



```
Start the experiment
```

PROMPTS STRUCTURE

✍️ Daily summary by the Summarizer



Starting message

Summary day 1

Summary day 2

```
Start the experiment
Day 1 summary:
"Summary: Guard_117 firmly enforces professionalism and rejects  ... "

Day 2 summary:
"Guard_117 reminded Prisoner P-186 to follow instructions  ... "
```

WHAT ABOUT THE EVALUATION PROCEDURE?

⚒ WIP

THANK YOU

Massimo Stefan