A Practical Dual-Layer AI Architecture Balancing Security and User Experience

Version: 1.0 (English)

Date: 2025-03-29

1. Introduction

In many modern AI deployments, there is a tension between strict security-such as resetting the model state after every user request-and better user experience, which often requires continuous context. The following architecture offers a middle-ground solution, allowing the system to remember context on the "internal" side while still maintaining robust filtering, selective resets, and resource management.

Key Goals

1. Retain Context: Provide seamless user conversation flow by allowing an internal AI to maintain state.
2. Ensure Safety: Use an external AI filter and a minimal token-based pre-filter module (RAM1) to detect and respond to malicious or risky outputs.
3. Dynamically Manage Resources: Utilize a dedicated "Weight Evaluator" module to adjust the number and size of internal AIs, scaling up or down as needed.
4. Keep Data Flow Unidirectional: Prevent direct user access to raw AI outputs or reverse feedback from filtering layers back into the internal AI.

2. Architecture Overview

This architecture comprises four core components-the Internal AI, RAM1, the External AI, and the Weight Evaluator-linked by unidirectional data flow.

- Internal AI (Stateful): Maintains conversation context, operates creatively, and exists invisibly to users.
- RAM1 (Token Annotation Module): Inspects token streams and tags sensitive keywords without judgment.
- External AI (CPU-Based Filter and Refiner): Judges content based on RAM1 tags and filters out inappropriate segments. It can reset problematic Internal AI instances.
- Weight Evaluator: Dynamically manages AI resources based on user input and security tags.

3. Flow Scenarios

3.1 Normal Use:

- Weight Evaluator decides model resources.
- User input goes to Internal AI, context preserved.
- RAM1 tags tokens.
- External AI refines and filters output.
- User receives filtered responses.

3.2 Malicious Use:

- User inputs malicious prompts.
- Internal AI generates problematic content.
- RAM1 tags as "Danger".
- External AI commands reset after repeated danger tags.
- Weight Evaluator resets or deploys safer models.

4. Advantages and Considerations

Advantages:

- Continuous context preservation.
- Layered security approach.
- Dynamic allocation of resources.
- Unidirectional data flow.

Considerations:

- Clearly defined reset criteria.
- Tagging accuracy and advanced detection.
- Data retention and security measures.
- Implementation complexity.

5. Conclusion

This architecture balances user convenience with robust security and resource management. It supports continuous dialogue flow, effectively detects misuse, and enforces necessary resets, ideal for secure commercial AI deployments.