



Analysis of On-Chain AI Agent Frameworks: Challenges and Current Implementations

jistro.eth

Last update: January 11, 2025

ABSTRACT

This paper examines three blockchain AI agent frameworks: Autonomie, G.A.M.E., and ChainGPT. We evaluate these frameworks based on their blockchain integration capabilities and AI implementation approaches. Our analysis reveals that current implementations primarily operate by wrapping existing AI models, with blockchain interactions limited to response execution rather than AI processing. While solutions like ChainGPT show promise through rollup technology, significant challenges remain in achieving true on-chain AI processing, particularly regarding model storage and computational requirements. These findings highlight the current state and limitations of blockchain-based AI agent architectures.

Keywords: EVM, Blockchain, Artificial Intelligence

CONTENTS

| | | |
|---|--------------------------|---|
| 1 | AI Agent Criteria | 1 |
| 2 | Autonomie | 1 |
| 3 | G.A.M.E. | 2 |
| 4 | ChainGPT | 2 |
| 5 | Conclusion | 2 |
| | References | 3 |

1 AI AGENT CRITERIA

Before to analyze every on-chain agent framework the following criteria have been established:

- Capability to interact with EVM chains or potential for implementation
 - The framework must be capable of interacting with smart contracts
 - Must support basic or complete blockchain operations, including state reading and transaction sending
 - (Optional) The agent must be capable of operating fully on-chain

This criterion ensures that the AI agent frameworks selected for analysis maintain direct blockchain integration capabilities.

2 AUTONOME

This framework consists of two main products:

- Eliza: Distinct from the 1964 chatbot, this is a versatile AI tool offering connectors for Discord, Twitter, and Telegram. It supports multiple language models (including Llama, Grok, OpenAI,

and Anthropic), provides document processing capabilities, and features an extensible system for custom actions.

- Agentkit: A developer tool integrated within the Coinbase Developer Platform (CDP) that enables the creation of autonomous AI agents for blockchain network interaction and automated crypto-powered applications.

Based on the established criteria, Agentkit emerges as the primary candidate. According to the documentation, the framework's operation involves two key steps:

First, developers must decide whether to execute predictions using a Trusted Execution Environment (TEE). A TEE provides a segregated and encrypted area of memory and CPU where data cannot be read or modified by external code. Only authorized code can manipulate data within the TEE [1].

Following this decision, Autonome requires both CDP API credentials and an OpenAI API key for operation. Once these configuration steps are completed, the framework generates a chatbot capable of interacting with and executing transactions on the blockchain.

3 G.A.M.E.

Developed by Virtuals Protocol, this framework enables the creation of AI agents capable of interacting with messages, 3D environments, and on-chain data through an ERC-6551 wallet. A notable feature of the protocol is the "Immutable Contribution Vault" (ICV), which provides several key functionalities [2]:

- Secure storage for user-uploaded custom models and datasets on the blockchain
- A Model Enrichment Pipeline that enhances AI models with new data, secured through cryptographic proofs
- Voice and Text Data Repositories that maintain data integrity and provenance through decentralized blockchain storage

However, upon thorough examination of the documentation and protocol data, no contract address for the ICV implementation could be located, raising questions about its current deployment status.

The deployment process of an agent involves several steps:

- Creation of a token, which functions similarly to holding stock but represents belief in the agent's capabilities
- Definition of character variables through the G.A.M.E. engine's prompting system
- Configuration of response and action parameters based on user input

The framework utilizes META's Llama as its AI model, with all AI executions occurring off-chain, which represents a significant limitation in terms of full blockchain integration.

4 CHAINGPT

This hackathon project leverages a Cartesi rollup as its backend infrastructure, utilizing Stanford's Alpaca Large Language Model to process responses within the rollup environment. The system's security architecture is built upon multiple validator nodes, which can be operated on standard computing hardware by individual participants. The key features of the system include:

- Reproducible results across validator nodes, ensuring response consistency
- Blockchain-based validation for settling disputes regarding chatbot behavior
- Cost-efficient consensus mechanism for result verification
- Comprehensive audit trail of all ChainGPT interactions

Among the analyzed frameworks, this implementation most closely aligns with the requirements for true on-chain AI agents, offering both decentralized execution and blockchain-based result validation.

5 CONCLUSION

After analyzing various agent frameworks, a clear pattern emerges. Current implementations primarily operate by wrapping existing AI models, either utilizing their existing infrastructure or replicating it for agent-specific applications. The blockchain interaction is largely limited to the execution of agent responses rather than the AI processing itself. More specifically:

- Most frameworks rely on established AI models rather than developing blockchain-native solutions
- On-chain interactions are primarily restricted to response execution
- True decentralized AI processing remains a significant challenge for the following reasons:
 - The storage requirements for model weights are too large for general-use blockchains (L1 or L2), exceeding typical block size limitations
 - On-chain execution requires substantial computational resources, making it impractical for current blockchain architectures
- The infrastructure predominantly remains off-chain, with blockchain integration serving mainly as a transaction layer

REFERENCES

- [1] vinfnet, “Trusted execution environment (TEE),” Oct. 2023. [Online]. Available: <https://learn.microsoft.com/en-us/azure/confidential-computing/trusted-execution-environment>
- [2] “Virtuals Protocol.” [Online]. Available: <https://www.virtuals.io/protocol>