AIGENTLAYER Whitepaper

Abstract

AIGENTLAYER is an autonomous AI agent designed to simplify and enhance user interaction with the EigenLayer protocol, a restaking framework on Ethereum. By translating natural language commands into on-chain actions, AIGENTLAYER removes complexity from restaking, making it accessible to a broader range of users. Built atop the open-source ai16z Eliza framework, integrated with GaiaNet's LLM services for natural language understanding, and architected for modular extensibility, AIGENTLAYER aspires to become a fully featured, community-driven solution.

This whitepaper outlines the project's core vision, technical architecture, roadmap, use cases, and a sustainable business model that builds on open-source principles while offering service-oriented value to users who prefer managed solutions.

Introduction

Restaking on EigenLayer can provide users with opportunities to earn additional rewards and participate in securing multiple services. However, the complexity of dealing with command-line tools, contracts, and blockchain nuances can deter less technical participants. AIGENTLAYER addresses this gap, creating a conversational interface that lets anyone issue commands like "Deposit 0.1 ETH into EigenLayer" or "Claim my rewards" without understanding the underlying processes.

Key Objectives:

1. **Accessibility:** Simplify the restaking experience on EigenLayer through natural language interactions.

2. **Modularity:** Leverage a flexible architecture (ai16z Eliza framework, TypeScript-based) for easy integration and community contributions.
3. **Extensibility:** Enhance capabilities over time, moving from basic deposit features to advanced strategies, automated decisions, and integration with data sources like EigenExplorer.
4. **Open-Source & Business Model Integration:** Provide a free and open platform, while offering optional paid services for those who require managed deployments, customization, or ongoing maintenance.

Background

EigenLayer

EigenLayer enables users to "restake" already-staked tokens (like ETH) to secure additional services. By doing so, participants can earn extra rewards and contribute to network security. While powerful, interacting directly with smart contracts and developer tools can be intimidating.

GaiaNet

GaiaNet is a decentralized AI ecosystem that provides robust LLM infrastructure. By leveraging GaiaNet’s scalable and reliable language processing capabilities, AIGENTLAYER ensures that user requests are understood accurately and executed consistently—even as more users and features are introduced.

ai16z Eliza Framework

AIGENTLAYER builds upon the ai16z Eliza framework, a flexible architecture supporting autonomous agents. Written in TypeScript, Node.js, and using a modular monorepo structure, Eliza provides a strong foundation for defining custom actions, providers, evaluators, and more.

Current Status and Core Features

Version v0 (Demo):

- **Deposit ETH:** The agent currently supports depositing specified amounts of ETH into EigenLayer via natural language requests.
- **Fetch Staked Balances:** Users can ask, “What’s my staked ETH balance?” and receive an immediate, human-readable answer.

These features serve as a proof-of-concept, demonstrating the feasibility of a conversational interface for EigenLayer interactions. They are the stepping stones for more complex operations and automation.

Roadmap

AIGENTLAYER’s development follows a staged approach, ensuring incremental improvements while validating each new feature set.

v0 (Current Demo)

- **Features:**
  - Deposit ETH into EigenLayer
  - Fetch and display staked balances
- **Goal:** Prove that natural language commands can trigger on-chain actions and retrieve data seamlessly.

v1 (MVP)

- **Features:**
  - **Claim Rewards:** Users can ask the agent to claim staking rewards from EigenLayer.
  - **Withdraw & Redelegate:** Easily move staked assets, withdrawing from certain services or redelegating elsewhere via a simple command.

    - **EigenExplorer Integration:** Pull real-time data (validator performance, APYs) to inform user decisions.
- **Goal:** Establish the core functionality for full restaking management and basic data-driven insights.

v2 (Enhanced Functionality)

- **Features:**
    - **Advanced Evaluators:** Implement market condition and risk evaluators. For example, if network conditions worsen or returns drop, the agent can suggest adjustments.
    - **Multiple User/Wallet Support:** Allow multiple users to run their own AIGENTLAYER instances or connect multiple wallets to a single agent instance.
    - **Community-Driven Extensions:** Encourage contributions of new actions, providers, and data sources.
- **Goal:** Transform AIGENTLAYER from a simple command executor into a smart assistant capable of guiding users toward better restaking decisions.

v3 (Strategic Intelligence)

- **Features:**
    - **Automated Strategies:** Instead of waiting for user commands, the agent can periodically analyze conditions and suggest or execute restaking moves based on predefined user policies.
    - **Integration with Other Protocols:** Expand beyond EigenLayer, interacting with other staking or DeFi services, provided they align with the user’s preferences and risk appetite.
    - **Custom Fine-Tuning:** Fine-tune the language model with specialized domain knowledge for even more accurate and context-aware responses.
- **Goal:** Establish AIGENTLAYER as an end-to-end autonomous portfolio manager, guiding users toward optimal restaking strategies in a multi-protocol ecosystem.

Technical Architecture

Core Components

1. **Actions:**
   Encapsulated logic for performing specific tasks. Current actions include depositing ETH and fetching balances. Future actions will handle claiming rewards, withdrawing, redelegating, and querying external APIs like EigenExplorer.
2. **Providers:**
   Manage external connections.
   - **Ethereum Provider:** Handles RPC calls to Ethereum testnets/mainnet and signs transactions.
   - **GaiaNet LLM Provider:** Communicates with GaiaNet APIs to interpret user input and generate human-like responses.
3. **Evaluators:**
   Determine how to respond to user prompts.
   - **Language Understanding Evaluator:** Maps user statements ("Deposit 0.1 ETH") to the correct action.
   - **Future Evaluators:** Incorporate market data, risk assessments, and time-based triggers.

Integration with EigenLayer

AIGENTLAYER connects to EigenLayer contracts via Ethereum RPC endpoints (e.g., testnets for demos, mainnet for production). It uses well-tested libraries (like ethers.js) to interact securely with the blockchain. Users only need to supply a funded wallet; the agent handles all transaction crafting and broadcasting.

Security Considerations

- **Key Management:** Private keys are stored in `.env` files for demos. Production setups should use secure vaults, hardware wallets, or managed key services.

- **Transaction Safety Checks:** Before executing a transaction, the agent can confirm balances, simulate calls, and prompt the user for final approval on high-value actions.

Business Model

AIGENTLAYER is open-source, enabling anyone to download, run, and modify the code without cost. This approach ensures broad community input, transparency, and trust. However, not all users will want the overhead of self-hosting, configuring environments, or maintaining instances. This creates opportunities for a service-oriented business model on top of the open-source foundation.

Service Offerings:

1. **Managed Hosting & Deployment:**
   For users who lack technical expertise or time, we can offer hosted solutions. They pay a monthly fee, and we handle server maintenance, updates, security, and uptime guarantees.
2. **Customization & Integrations:**
   Some teams may require custom evaluators, specialized providers, or integrations with proprietary data sources. We can offer consulting and development services at a fixed or hourly rate.
3. **Premium Support & Training:**
   Educate larger organizations or communities on how to leverage AIGENTLAYER effectively. Offer dedicated support channels, priority bug fixes, and feature requests.
4. **White-Label Solutions:**
   DAO treasuries, staking pools, or DeFi platforms might want to brand their own version of the agent. Offer white-label services where AIGENTLAYER runs under their branding, seamlessly integrated into their ecosystem.

Value Proposition:

- **For Non-Technical Users:** Peace of mind. They get the benefits of EigenLayer without the technical complexity.
- **For Developers & Startups:** A starting point and reference implementation they can fork or build upon. If they hit a roadblock or need advanced features, paid services are available.

This hybrid open-source plus services model ensures the community can thrive while also generating revenue streams that fund further development, marketing, and community-building initiatives.

Use Cases

1. **Individual Stakers:**
   A non-technical user who has ETH and wants to restake it on EigenLayer. Instead of learning complex tools, they just say, "AIGENT, deposit my 0.5 ETH," and periodically, "AIGENT, claim my rewards."
2. **DAOs and Communities:**
   A DAO managing treasury funds might restake assets into EigenLayer. AIGENTLAYER can help the DAO quickly respond to changes. "AIGENT, how are our staking rewards this month?" or "AIGENT, redelegate 10% of our stake to a new service."
3. **Service Providers:**
   Third-party staking providers can use AIGENTLAYER as a customer-facing interface. Their clients never deal with raw blockchain calls; they interact with a friendly assistant while the provider handles the complexity behind the scenes.

Open Source and Community Involvement

AIGENTLAYER is licensed under a permissive open-source license (e.g., MIT) to encourage community growth. The community can:

- Contribute new features and evaluators.
- Improve documentation and onboarding materials.
- Propose integrations with additional protocols or services.
- Run their own instances, spin up test environments, or offer specialized versions.

Community governance models can emerge where token holders or active contributors vote on new features, ensuring the platform evolves to meet the needs of its user base.

Scalability and Future Vision

As user adoption increases, AIGENTLAYER's modular architecture allows for horizontal scaling. Multiple agent instances can handle requests in parallel. GaiaNet's infrastructure ensures that language queries remain responsive. Over time, the agent could become a one-stop interface for multi-chain staking, liquidity provision, or yield farming—a universal portal to the decentralized finance ecosystem.

With advanced evaluators, the agent may learn patterns, optimize returns, and proactively suggest adjustments. It could alert users when network fees spike, or when it's optimal to claim rewards. The long-term vision is a self-improving, data-driven assistant guiding users through the evolving crypto landscape.

Conclusion

AIGENTLAYER represents a shift in how people interact with complex protocols like EigenLayer. By blending natural language processing, modular architecture, and a community-driven open-source ethos, it democratizes restaking. Users benefit from a friendly, conversational interface, while developers and businesses can build, customize, and monetize around it.

With a clear roadmap—from the current demo phase (v0), through the MVP (v1), and onward to fully autonomous strategies (v3 and beyond)—AIGENTLAYER positions itself to become a cornerstone tool for both newcomers and experienced participants in the evolving DeFi ecosystem.

References and Resources

- **EigenLayer Docs:** https://docs.eigenlayer.xyz/
- **GaiaNet Docs:** https://www.gaianet.ai/docs
- **ai16z Eliza Framework:** https://ai16z.github.io/eliza/docs/intro/
- **AIGENTLAYER GitHub Repository:** https://github.com/worksgoodcompany/aigentlayer-demo

This whitepaper provides a comprehensive overview and blueprint for AIGENTLAYER's development, positioning it as both an immediate simplification of EigenLayer restaking and a strategic platform for future growth, automation, and community-driven innovation.