Builder Agent 2.0 Technical Specification

Overview

Builder Agent 2.0 will automate crypto trading and liquidity pool management, leveraging advanced AI strategies, decentralized governance, and blockchain integration to optimize financial outcomes for Cuttlefish Labs projects.

Goals

- Execute AI-driven crypto trading strategies.
- Deploy and manage liquidity pools for optimized yield.
- Integrate DAO governance for transparency and decentralization.
- Maintain robust risk management and security.

Architecture

1. Crypto Trading Module

- **Predictive Analytics:**
- Use machine learning models trained on market data, sentiment analysis, and technical indicators.

- Real-time monitoring of Twitter and other platforms for sentiment signals.

- **Trade Execution:**

- API integration with Binance, Ethereum blockchain, and other exchanges.
- Real-time balance and risk checks before executing trades.

2. Liquidity Pool Management Module

- **Automated Pool Creation:**
- Smart contracts for dynamic pool deployment and management.

- Integration with ERC20 and ERC721 standards for asset tokenization.

- **Yield Optimization:**

- Automated strategies for liquidity rebalancing and yield farming.
- Monitoring liquidity and rewards performance.

3. AI & Orchestration Module

- **AI Planning & Task Management:**
- Structured AI workflows: Plan, Execute, Verify, Refine.

- Utilize OpenAI's LangChain for decision-making processes.

- **Multi-agent Coordination:**

- NLP-driven task parsing and execution.
- TrustGraph integration for peer evaluation and reputation management.

4. Decentralized Governance (DAO) Module

- **On-Chain Voting & Proposals:**
- ERC20 token-based voting.

- Transparent proposal submissions and evaluations.

- **Norm Evolution:**

- Stakeholder-driven updates and adaptations.

Technology Stack

- **Languages:** TypeScript, Python, Solidity
- **Blockchain:** Ethereum, ERC20, ERC721
- **AI Framework:** OpenAI, LangChain
- **Frameworks/Libraries:** React, Ethers.js, Web3.js, Tweepy, Binance API

Security and Risk Management

- Input sanitization and structured logging.
- Real-time monitoring and automated risk alerts.
- Escrow mechanisms for fund management.

Implementation Roadmap

- **Phase 1:** Development of core trading and liquidity management logic.
- **Phase 2:** AI integration and orchestration.
- **Phase 3:** DAO integration, governance, and security audit.
- **Phase 4:** Deployment, monitoring, and iterative improvements.

Deliverables

- Functional Crypto Trading Agent.
- Automated Liquidity Pool Manager.
- Integrated DAO governance interface.
- AI-driven orchestration and decision-making platform.

Next Steps

- Confirm the detailed module-specific requirements.
- Begin phase-wise development according to the outlined roadmap.