MVP for Atomic Composability

Overview

- **Goal**: To demonstrate feasibility and effectiveness of the model in a real-world setting.
- **Strategy**: Develop an MVP focusing on core functionalities to solve atomic composability challenges in Ethereum rollups.

Objective

Validate atomic transactions across multiple Ethereum rollups.

Scope

- Select a small set of rollups for interoperability testing.
- Implement basic mechanisms: buffering, dependency handling, concurrency control, and zero-knowledge proof validations.

Key Metrics for Success

- **Transaction Success Rate**: The percentage of transactions correctly finalized across rollups.
- **Performance**: The time taken for cross-rollup transactions to be processed.
- **User Experience**: Feedback from users regarding the ease of use and understandability of the process.
- **Security**: The number of security issues identified and resolved.

Use Cases and User Stories

Critical Use Cases

- Cross-rollup token swaps.
- Multi-step contract interactions across different rollups.
- Conditional execution of transactions based on another rollup's state.

User Stories

- Articulate clear needs and expected outcomes for each use case.

Technical Architecture

Development Tools

- Frameworks like Truffle or Hardhat.
- Programming languages: Vyper for smart contracts, JavaScript/Python for off-chain components.

Infrastructure

- Deployment on testnets (Rinkeby, Ropsten) or a local blockchain (Ganache).

Building Core Components

Components

- **Buffering Mechanism**: Smart contract for transaction queuing.
- **Dependency Handler**: Functions for resolving transaction dependencies.
- **Concurrency Control**: Method to timestamp and validate transaction concurrency.
- **Zero-Knowledge Proof Integrations**: (Optional) For validating transactions confidentially.

Integration and Testing

Integration

- Combine core components to function cohesively.

Testing

- Execute and buffer transactions on rollups.
- Test dependency resolution and transaction processing.
- Validate concurrency control and ZK-proof effectiveness.

Interface and MVP Testing

Interface Development

- Simple UI for transaction submission and status tracking.

MVP Testing

- **Unit Testing**: Function correctness.
- **Integration Testing**: Component compatibility.
- **End-to-End Testing**: Real-world scenario simulation.
- **Security Testing**: Cross-chain transaction vulnerability checks.

Implementation details

Smart Contract Interaction

Core of Rollup Functionality

Rollup Smart Contracts

- Handling transactions submissions.
- Creating proofs, especially for ZK-Rollups.
- Finalizing transactions on Ethereum mainnet.

Writing Contract Interactions

Key Functions

- Submitting Transactions: Token transfers, smart contract interactions.
- Handling Proofs: Generating/verifying zero-knowledge proofs (ZK-Rollups).
- Retrieving Data: Querying the status of transactions, state roots.

Tools and Languages

- Truffle, Hardhat, Vyper for development.

SDKs and APIs

Simplifying Rollup Integration

Leveraging Tools

- SDKs/APIs for easier interaction with rollups.
- Abstracting complex transaction submissions and proof handling.

Event Monitoring

Staying Informed and Responsive

Key Events

- Transaction batching.
- State root updates.
- Proof submissions.

Decentralized System Operation

Goal

- Running the rollup system in a trustless, decentralized manner.
- Ensuring integrity across on-chain and off-chain components.

On-Chain and Off-Chain Components

Balancing Decentralization

On-Chain

- Smart contracts on Ethereum mainnet: Immutable and trustless.

Off-Chain

- Challenges in decentralizing: Oracles, relays, and infrastructure.
- Strategies for distribution and transparency.

Implementing Decentralized Components

Distributed Oracles/Relays

- Network of nodes for consensus on transaction states.

Decentralized Hosting

- Leveraging platforms like IPFS for infrastructure.

Open Participation and Governance

- Incentivizing operation and DAO for system management.

Conclusion and Future Vision

Building the Future

- Implementing a decentralized, trustless system for rollup operations.
- Aligning with blockchain's ethos to enhance Ethereum's capabilities.

Next Steps

- Continued development and community involvement.
- Expanding and refining the decentralized operation model.