Grant Title:

Development of an Open-Source Tool for Verifying MEV Blocker Transactions

Author:

- Name: Artem
- GitHub: Mefiseru, hayashidevs
- Email: f92388556@duck.com
- Discord: hayashidev.april

About You:

I am a Python developer with six years of experience, currently working on an anonymous P2P messenger using Web3 technology, set to be publicly available within this week. My expertise lies in blockchain interaction and Django-based web applications.

Grant Category:

Developer tools (SDK)

Grant Description:

This project aims to build an open-source tool that verifies MEV Blocker transactions, maximizes user refunds, and flags rule violations. Utilizing Python and Django, the tool will ensure secure and efficient verification of transactions while providing real-time monitoring and alerts through a Django-based interface.

Grant Goals and Impact:

The primary goal is to develop a reliable and efficient tool to verify MEV Blocker transactions. This will maximize user refunds and provide real-time monitoring and alerts, enhancing the transparency and security of the CoW Protocol ecosystem. The successful execution of this project will benefit users by ensuring fair transaction processes and increasing trust in the protocol.

Milestones:

Milestone

Payment (xDAI)

Payment (COW Tokens)

Setup and Initial Development

200 xDAI

820.67 COW

Transaction Identification

850 xDAI

3487.14 COW

State Management and Verification

1050 xDAI

4308.24 COW

Monitoring Interface and Alerts

600 xDAI

2462.01 COW

Dockerizing

400 xDAI

1641.34 COW

Testing and Deployment

100 xDAI

410.34 COW

Total

3200 xDAI

13128.17 COW

Milestone Descriptions:

Milestone 1: Setup and Initial Development

- Tasks:
- Set up the Django project structure.
- Implement basic blockchain interaction using web3.py.
- Set up the Django project structure.
- Implement basic blockchain interaction using web3.py.
- Outcomes:
- A functional Django project with initial blockchain interaction capabilities.
- A functional Django project with initial blockchain interaction capabilities.

Milestone 2: Transaction Identification and Simulation

- Tasks:
- Fetch block data and identify MEV Blocker transactions.
- Track related bundles within the specified time frame.
- Simulate bundles using web3.py and sort by payment value.
- Fetch block data and identify MEV Blocker transactions.
- Track related bundles within the specified time frame.
- Simulate bundles using web3.py and sort by payment value.
- Outcomes:
- Accurate identification and simulation of MEV Blocker transactions.
- Accurate identification and simulation of MEV Blocker transactions.

Milestone 3: State Management and Verification

- Tasks:
- Record the initial state using web3.py.
- Simulate backrun transactions and update the state.
- Verify the inclusion of transaction hashes and ensure refunds.
- Record the initial state using web3.py.
- Simulate backrun transactions and update the state.
- Verify the inclusion of transaction hashes and ensure refunds.
- Outcomes:
- Effective state management and transaction verification.

- Effective state management and transaction verification.

Milestone 4: Monitoring Interface and Alerts

- Tasks:
- Set up Django models and views for monitoring.
- Implement real-time alerts using Django Channels.
- Set up Django models and views for monitoring.
- Implement real-time alerts using Django Channels.
- Outcomes:
- A functional monitoring interface with real-time alert capabilities.
- A functional monitoring interface with real-time alert capabilities.

Milestone 5: Dockerizing

- Tasks:
- Create a docker-compose file with all required services and environment variables.
- Create a docker-compose file with all required services and environment variables.
- Outcomes:
- A fully Dockerized application ensuring easy deployment and scalability.
- A fully Dockerized application ensuring easy deployment and scalability.

Milestone 6: Testing and Deployment

- Tasks:
- Conduct comprehensive testing of all components.
- Prepare detailed documentation.
- Deploy the tool in a production environment.
- Conduct comprehensive testing of all components.
- Prepare detailed documentation.
- Deploy the tool in a production environment.
- Outcomes:
- A thoroughly tested and documented tool ready for production use.
- A thoroughly tested and documented tool ready for production use.

Funding Request:

The total funding requested is 3200 xDAI, justified by the scope and complexity of the project, as well as the anticipated impact on the CoW Protocol ecosystem.

I would prefer to receive the funding in either xDAI or COW tokens equivalent.

Budget Breakdown:

Category

Amount (xDAI)

Amount (COW Tokens)

Development

2000 xDAI

8206.71 COW

Testing

500 xDAI

2051.68 COW

Deployment

300 xDAI

1231.01 COW

Documentation

200 xDAI

820.67 COW

Miscellaneous

200 xDAI

820.67 COW

Total

3200 xDAI

13128.17 COW

Gnosis Chain Address (to receive the grant):

0xC6C985637eabC798c232dDAd3afFde525242dFd7

Other Information:

This project builds on my previous work in blockchain and Django, ensuring a high level of reliability and performance. I have attached relevant documents and previous work samples for reference.

Referral:

Olga Fetisova.

Terms and Conditions:

By submitting this grant application, I acknowledge and agree to be bound by the CoW DAO Participation Agreement and the CoW Grant Terms and Conditions.