Reverse Auction Mechanism for Oasis Protocol with Staking Feature

This document outlines the solution architecture for implementing a reverse auction mechanism on the Oasis Protocol.

A reverse auction is where sellers compete to offer the lowest price for goods or services requested by buyers. The Oasis Protocol,

known for its privacy, security, and scalability, serves as the foundation for this decentralized auction system.

This document will cover core components, detailed implementation steps, and architecture diagrams to guide the development.

Table of Contents

1. Reverse Auction Overview

A reverse auction is a type of auction where multiple sellers compete by bidding lower prices for a contract issued by a buyer. The seller offering the lowest price wins the auction and provides the requested goods or services.

In this context, the Oasis Protocol, which supports decentralized applications (dApps) with privacy and scalability, is used to ensure secure bidding, confidentiality of bids, and efficient transactions on the blockchain.

2. Solution Architecture

The reverse auction mechanism will rely on smart contracts deployed on the Oasis blockchain. These contracts will handle various auction-related processes:

- **Auction Creation**: Buyers initiate auctions with details such as a description of the service or goods, maximum acceptable price, and auction duration.
- **Bid Submission**: Sellers submit their bids, which are securely handled using Oasis Protocol's privacy features.
- **Bid Validation**: The smart contract automatically validates bids to ensure they meet the buyer's requirements.
- **Auction Conclusion**: After the auction period ends, the contract identifies the lowest valid bid.
- **Payment**: The smart contract handles payments from the buyer to the winning seller once the goods or services are delivered.

2.1 Smart Contract Components

The auction workflow consists of several key steps, illustrated in the following diagram:

1. **Auction Creation**: A buyer submits a request via a smart contract.
2. **Bid Submission**: Multiple sellers submit bids to the contract.
3. **Bid Evaluation**: The contract evaluates the bids based on the lowest price and validity.
4. **Auction Conclusion**: The lowest bid is selected.
5. **Settlement**: Payment is transferred after successful delivery.

[Insert Architecture Diagram Here]

3. Implementation Steps

- **Auction Creation**: Buyers post their auction request (goods/services, max price, terms, etc.).
- **Bid Submission**: Sellers submit confidential bids through a smart contract.
- **Bid Validation**: The smart contract ensures that bids meet the set requirements.
- **Auction Conclusion**: The contract selects the lowest valid bid at the end of the auction.
- **Settlement**: Upon successful delivery of goods/services, the contract releases the buyer's payment to the seller.

4. Oasis Protocol Features Utilized

- **Confidential Smart Contracts**: Oasis Protocol allows for secure and private bid submissions, preventing price manipulation during the auction.

- **ParaTimes**: These help with scalability by segregating computation, ensuring that high-throughput auctions remain efficient.
- **Decentralized Finance Integration**: Confidential payments and settlements can be handled securely through Oasis, providing trustless finality.

5. Security and Fraud Prevention

- **Privacy Protection**: Use confidential computing to protect bid details until the auction concludes.
- **Immutable Ledger**: Every action within the auction (bids, validations, winner selection) is stored immutably on the Oasis blockchain.
- **Dispute Resolution**: Implement mechanisms for handling disputes between buyers and sellers, with automatic arbitration if required.

6. Enhancements and Variations

- **Reputation System**: Build a reputation system where buyers and sellers rate each other after successful auctions.
- **Multi-Round Auctions**: Implement dynamic auctions with multiple bidding rounds to add competition intensity.
- **Token Economy**: Use tokens for rewards, payments, and fees, integrating with the Oasis ecosystem.

7. Technical Implementation Outline

- **Smart Contracts**: Develop smart contracts using Rust or Solidity compatible with the Oasis

ParaTime for handling auctions.

- **Front-End Interface**: Design an interface for buyers and sellers to create, manage, and participate in auctions.

- **Wallet Integration**: Ensure that users can connect their wallets for secure payments and auction participation.

8. Conclusion

The reverse auction mechanism on Oasis Protocol offers a robust, secure, and privacy-preserving solution for decentralized procurement.

By leveraging Oasis' advanced features, this system ensures trustless, efficient, and private auctions, opening new possibilities for
decentralized commerce.

9. Buyer Staking and Yield Generation Feature

9. Buyer Staking and Yield Generation Feature

This feature introduces a staking mechanism for buyers, where they must stake a specified amount of tokens before placing an auction offer. The staked amount is greater than or equal to the final price offered by the winning seller, and any excess is transferred to a liquidity pool where it generates yield. The excess amount, along with the yield, can be claimed by the buyer after a lock-in period.

Key Elements:

1. **Buyer Staking Requirement**: Buyers must stake a specified amount of tokens, ensuring they

are committed to the auction. The staked amount must meet or exceed the potential cost of the service or product they intend to purchase.

2. **Excess Amount to Liquidity Pool**: If the staked amount exceeds the final price offered by the seller, the excess amount is automatically transferred to a liquidity pool. This pool generates yield for the buyer over time.

3. **Yield Generation and Lock-in Period**: The excess funds in the liquidity pool earn a yield during a predefined lock-in period. Once the lock-in period expires, buyers can claim the excess amount and the yield they have earned.

Workflow Changes:
1. **Auction Creation**: The buyer must stake a specific amount of tokens in a smart contract before creating an auction.

2. **Excess Fund Handling**: When the auction ends and the seller is chosen, the smart contract calculates the excess amount (if any) and transfers it to the liquidity pool.

3. **Yield and Claim**: After the lock-in period, the buyer can trigger a claim function to retrieve the excess funds and any yield earned from the liquidity pool.

This feature adds a financial benefit for buyers and increases their commitment to the auction, while integrating a DeFi yield mechanism.