Eve: A Blockchain-Based Protocol for Matching Markets

Abstract

We propose a protocol for coordinating market matching without relying on centralized intermediaries. Eve uses blockchain transaction memos to implement an event-sourcing pattern, where market participants express bidding intents with constraints, and matches occur when sufficient compatible bids exist. The protocol enables threshold coordination where events only proceed when predetermined requirements are met, such as minimum capacity or funding thresholds. By encoding commands in transaction memos and deriving system state through deterministic processing, Eve provides complete transparency, auditability, and reply capability while minimizing on-chain computation. This approach creates efficient two-sided markets that can operate across various domains, from event ticketing to telescope time allocation, with minimal trust requirements and maximal coordination efficiency.

Protocol first approach

1. Introduction

Organizing on the internet has come to rely on large organizers selling tickets to standard event formats like conferences, concerts, and conventions or through highly visible campaigns like, to give examples of campaigns from the organizer's and the audience's perspectives, crowdsourcing and reaching funding through social media and getting an organizer to fund an event because a mass expression of audience interest was clearly visible online. While these strategies already involve putting down money to establish confidence and trust, these systems still suffer from inherent weaknesses of the trust based centralized model. Traditionally, decentralized event commitments are not common, as trust in organizers or centralized platforms remains necessary to handle payments, validate attendance pledges, and ensure sufficient collective interest to proceed. The dependency on intermediaries increases overhead costs, limiting the viability of smaller, informal gatherings and reducing spontaneity and fluidity.

The rise of blockchain technology has enabled new forms of coordination without centralized control, but current implementations rely on complex smart contracts that require advanced planning and are expensive to execute complex computation on-chain. Moreover, most blockchain-based market systems focus on simple auction mechanisms rather than multi-dimensional matching with constraints.

What is needed is a protocol that leverages the blockchain's immutability and ordering properties while keeping complex computation off-chain, enabling transparent, efficient, and flexible market matching. The protocol should support expressing complex preferences and constraints, operate with minimal on-chain footprint, and provide deterministic outcomes that can be independently verified.

In this paper, we propose Eve, a blockchain-based protocol for market matching that addresses these challenges through an event-sourcing architecture. Eve enables users to express bidding intents with optional constraints directly in transaction memos. These intents are processed deterministically to find valid matches that satisfy all participant constraints. When matches occur, resources are allocated according to predetermined rules, creating an efficient coordination mechanism without requiring trust in a central authority.

2. System Overview

Eve's bidding architecture is based on the event-sourcing pattern, where all system state is derived from an ordered sequence of immutable market actions. In Eve, these market actions are encoded in blockchain transaction memos, creating a canonical, tamper-proof log of all market actions.

2.1 Key Design Principles

1. **Command-Query Responsibility Segregation (CQRS)**: Eve separates write operations (commands) from read operations (queries). Commands modify the market actions log through blockchain transactions, while queries run against blockchain derived read models off-chain.

2. **Event Sourcing**: All system state is derived by processing the event sequence. There is no direct state modification, only event emission.

3. **Deterministic Processing**: Any node can reconstruct an identical state by processing events using deterministic rules. There are no out-of-band or non-deterministic inputs.

4. **Novel Minimalist Blockchain Integration**: Transaction memos encode commands, transaction amounts represent bid values, block height and transaction sequence provide strict ordering, and transaction senders authenticate commands.

5. **Smart Contract Agnosticism**: Eve uses native blockchain capabilities (ordering, immutability) rather than complex smart contracts, externalizing business logic to deterministic processors that can evolve over time. The use of Eve does not preclude the use of smart contracts, though, and can be extended to utilize smart contracts in the future.

System Components

The Eve protocol consists of the following core components:

1. **Transaction Memos**: Structured commands that express market intents and constraints are encoded in a human-readable format in the transaction memo field.
2. **Market Action Processor:** A deterministic engine that processes events in sequence to derive system state.
3. **Matching Algorithm:** Rules for finding valid matches between Bid and Match commands based on constraints.
4. **Verification Mechanism**: Methods to independently verify system state and match outcomes.

3. Core Protocol Elements

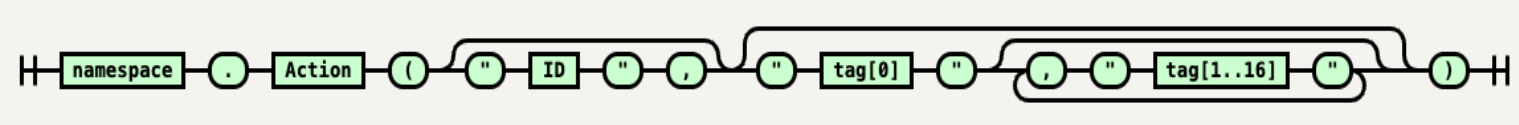
3.0 Action Metamodel

The Action Metamodel describes a flexible syntax structure for invoking actions within a given namespace. The general format is:

```
Unset
namespace.Action("ID", tags...)
```

where:

- **namespace** represents the context or domain in which the action is invoked.
- **Action** specifies the method or operation to be executed.
- **ID** (optional) uniquely identifies the target resource or object. It must be provided for existing resources but can be omitted for actions creating new resources.
- **tags** (optional) are additional key-value parameters that provide further context or metadata related to the action. NOTE: Only the 16 most recent key values associated with a given ID are stored/indexed.



3.1 Market Action Types

Eve defines four types of Market Action Types that can be expressed through transaction memos:

1. **Bid**: Represents a bidding intent for participating in a market (e.g. bidding to attend an event or bidding for utilization of a constrained resource, such as a telescope) with a maximum price the bidder is willing to pay and additional constraints, such as location and dates.
2. **Match:** Represents a market creator's matching proposal with minimum requirements such as location, dates, minimum capacity and price threshold.
3. **Intent:** A non-binding declaration of intent from the market maker that signals future market creation possibilities.
4. **System:** Reserved for protocol-level operations defined by the governing parameters.

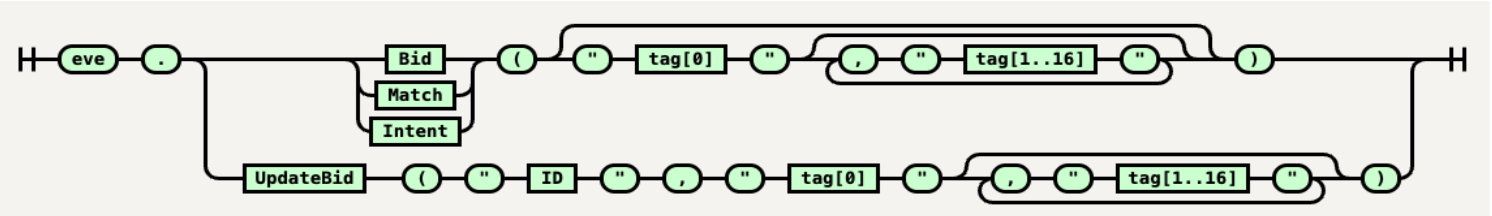
3.2 Memo Field Operations

Commands in Eve follow a standardized syntax:

```
eve.<action>(<parameters>)
```

Where:

- <action>: represents function-call like commands such as: Bid, Match, Intent, UpdateBid, UpdateIntent, RevokeBid, RevokeIntent
- <parameters>: are key-value pairs that define constraints and properties



Examples:

```
eve.Bid("location:virtual", "dates:2025-10-01...2025-10-10")
eve.UpdateBid("42AB9335", "location:San Francisco")
eve.Match("location:virtual", "dates:2025-10-02", "min_capacity:100",
"min_bid_amount:50")
eve.Intent("location:virtual", "dates:2025-10-02", "min_capacity:100",
"min_bid_amount:50")
```

3.3 Bid Mechanics

A bid represents a user's intent to participate in a market with constraints. The maximum bid amount is specified by the message value, while constraints are encoded in the memo field of the transaction.

Key aspects of bids include:

- Required parameters (such as location and dates)
- Optional parameters which may represent any number of additional constraints
- Bid ID, derived from the first eight bytes of the initial transaction creating the bid
- A bid may be marked as a "pledge", which pledges the full maximum bid amount to the matching event, instead of refunding the difference between the minimum bid requirement and the bidder's maximum bid
- Support for upda tes and revocations

For instance, a bid submitted to the even organizer, Build the Future (BTF), DAO address could look as follows:

```
sender:    000001
receiver:  000002
amount:    50 PHOTON
memo:      eve.Bid("location:virtual", "dates:2025-10-01...2025-10-10")
```

In this transaction, the sender is stating that they would like to attend a virtual event anytime between the dates of Oct 1-10 with a maximum bid of 50 PHOTON.

3.4 Match Mechanics

A match represents a market creator's proposal matching a set of bids. Matching includes requirements like minimum capacity and minimum bid amount and must match the constraints set by the bidder on constraints such as location and date.

Key aspects of matches include:

- Required parameters include location, dates, minimum capacity, minimum bid amount, etc
- Optional Parameters include maximum capacity and funding goal
- A Match is successful if and only if a compatible set of bids exist at the time of submission
- Earliest-wins rule for competing matches

4. State Transitions and Market Clearing

4.1 State Derivation

Eve's bid and match state are not stored directly on-chain, but rather derived by processing all market actions in sequence. The reproducible state consists of:

1. Active bids with their constraints and amounts
2. Successfully completed bid matches
3. Revoked bids
4. Failed matches

Any node can reconstruct the complete state by processing all market action events from the beginning of the chain, or from a trusted checkpoint.

4.2 Market Clearing Rules

When a match command is submitted, it is evaluated against all active bids to determine if a successful match can occur. A match is successful when:

1. Sufficient bids exist that satisfy the match constraints
2. The total value of compatible bids meets or exceeds the minimum capacity * minimum bid requirements and/or the funding goal set by the match.
3. Each compatible bid's maximum amount is greater than or equal to the minimum bid requirement

If these conditions are met, the match is executed:

- Matching bids are removed from the active pool
- Any excess bid amount is refunded to the bidders
- The match is recorded as successful

If multiple bids satisfy a match, but exceed matching thresholds, the earliest-created match that satisfies all constraints wins.

4.3 Price Discovery

Price discovery in Eve occurs through transparency of market demand prior to match creation. Event organizers or resource allocators can query the pool of open bids to understand what the market will bear before proposing a match:

1. **Market Visibility**: Potential matchers can analyze the distribution of maximum bid amounts (max_bid) across all open bids that meet their criteria
2. **Aggregate Demand Assessment**: By examining bid constraints and maximum prices, matchers can identify optimal pricing points that maximize both participation and revenue.
3. **Pledge Aggregation**: The system allows matchers to see potential "pledge totals" = the sum of committed funds from users who have marked their bids as pledges rather than standard bids.

This approach enables matchers to make informed decisions about minimum bid amounts when creating matches, setting prices that reflect actual market conditions. When a match is created, the `min_bid_amount` serves as the clearing price for all standard bids, while pledges contribute their full `max_bid` amount regardless of the clearing price, serving as a form of sponsorship or premium support.

5. Applications

5.1 Event Coordination

The initial use-case application of Eve is coordinating a pre-sale market for event attendance. In this scenario:

- Event organizers express intent of location, venue, dates, and minimum attendance requirements.
- Potential attendees create bids with maximum prices and preferences
- Event matches only occur when sufficient interest exists
- Prices adjust based on supply and demand

This solves the classic coordination problem of event planning, where organizers need to commit to venues before knowing attendance, and attendees want to commit only if the event is certain to happen.

5.2 Resource Allocation

Eve can be extended to other domains involving resource allocation in a two-sided market:

- Telescope time scheduling
- Computer resource allocation
- Community funding of public goods

In each case, the protocol enables efficient coordination without requiring trust in a third-party.

6. Security Considerations

6.1 Trust Model

Eve embraces a trust-but-verify model by making all operations transparent and independently verifiable from blockchain data alone. The security model relies on:

1. Blockchain security for transaction ordering and immutability
2. Deterministic processing for state derivation
3. Public verifiability of all operations
4. Trust in the custodial account: Bidders must trust the eve account that receives their funds. This account could potentially:
    - Withhold matching indefinitely and retain funds
    - Refuse to issue refunds for unsuccessful matches
    - Misappropriate or misuse the deposited funds

This custodial risk represents the primary trust requirement in the system. Real-world implementations should mitigate this through escrow smart contracts, multi-signature arrangements, time-locked releases, or decentralized treasury governance.

Apart from the custodial trust requirement, users need not trust market creators or other participants, as matching rules are enforced by the protocol and verifiable by anyone.

6.2 Potential Attacks

Potential attack vectors include:

- Front-running of bids
- Malicious market creators who never fulfill obligations

TBD: we need more here

7. Implementation Considerations

7.1 Blockchain Requirements

While initially designed for AtomOne, Eve's minimalist requirements make it appropriate for a wide range of blockchains.

Blockchains must support:

- Arbitrary memo fields in transactions
- Token Transfers
- Reliable Transaction ordering

The protocol is blockchain agnostic and can be implemented on any platform meeting these requirements.

7.2 Impact of Decentralized Implementation

matching systems traditionally rely on centralized platforms that act as trusted intermediaries between participants. While functional, these systems suffer from several limitations: they introduce single points of failure, impose arbitrary rules and fee structures, lack transparency in their matching algorithms, and create data silos that prevent interoperability. Additionally, centralized systems often struggle with coordination problems where resources are allocated inefficiently due to incomplete information about participant preferences.

Scaling Considerations

Off-chain processing of market actions enables significant scaling advantages:

- On-chain footprint limited to transaction memos
- Complex matching logic executed off-chain
- State queries handled by optimized read models

This approach allows for the protocol to scale to thousands of simultaneous markets with minimal blockchain load.

8. Conclusion

Eve represents a novel approach to market matching that leverages a minimalist approach to taking advantage of the strength of blockchain technology. By encoding market actions as events in transaction memos and deriving state through deterministic processing, Eve enables transparent, efficient coordination without requiring complex smart contracts or third parties.

The protocol's flexibility allows it to be applied across various domains, from event ticketing to resource allocation, creating new possibilities for decentralized coordination. Eve's event-sourcing architecture provides complete auditability and replay capabilities, ensuring that all market operations are transparent and verifiable.

As blockchain technology continues to evolve, protocols like Eve that combine on-chain and off-chain components will play an increasingly important role in enabling efficient, low-trust coordination at scale.

9. Risks

The possibility of unreliable participants forces organizers to impose additional verification steps, collecting more private data from attendees than strictly required. A certain level of non-attendance or false commitments is accepted as unavoidable.

References

TBD