Matchbox: Decentralized Conditional Wagering on Prediction Markets

calmxbt (https://github.com/calmdentist)

October 24, 2025

Abstract

Decentralized Prediction Markets (DPMs) offer transparent, peer-to-peer wagering but critically lack the ability to support conditional wagers like parlays. This is not a feature gap but a fundamental architectural flaw: the combinatorial explosion of ”parlay markets” makes it impossible to provide liquidity for every outcome. Matchbox solves this problem by introducing a non-custodial, decentralized automation layer. Instead of creating new, illiquid markets for conditional bets (e.g., $P(A \cap B)$), Matchbox provides a trustless ”if-then” execution router that executes a sequence of trades against existing, liquid, underlying markets.

1 The Problem: The Combinatorial Liquidity Trap

Traditional sportsbooks derive a significant majority of their revenue from parlays. For DPMs like Polymarket to compete, they must offer a similar product.

The intuitive solution—creating a new market for every parlay (e.g., a ”YES on A and YES on B” token)—is a trap. This approach is unscalable and fragments liquidity. It requires a market maker to price and provide liquidity for every *combination* of events, which is combinatorially impossible.

A secondary, unsolved problem is price volatility. In a sequential parlay, the user buys ”YES on A.” If A wins, they must then buy ”YES on B.” However, the resolution of Market A provides new information to the market, meaning the price of B (the true conditional probability, $P(B|A)$) may have changed drastically. The user has no protection from this volatility and slippage.

2 The Solution: The ”Matchbox” Protocol

Matchbox is not a new prediction market. It is a non-custodial meta-protocol that allows users to design and deploy automated, conditional workflows.

The core principle is simple: **We separate the logic from the liquidity.**

A ”Matchbox” is a user-owned smart contract vault that holds their funds and a predefined set of rules. This vault is triggered by an external, decentralized automation network (e.g., Chainlink Automation) to execute trades against existing DPMs.

2.1 Protocol Architecture

The protocol consists of three distinct layers:

- **The Design Layer (dApp):** A user-friendly web interface where a user can build their conditional sequence (the ”circuit”) without writing code. The dApp helps the user deploy their personal `Matchbox` vault.

- **The Logic Layer (Smart Contracts):**
  - `MatchboxFactory.sol`: A single factory contract that deploys new, user-owned `Matchbox` vaults.
  - `Matchbox.sol`: The user's personal, non-custodial vault. It holds the user's funds (USDC, conditional tokens) and the rule-set for their sequence. The user is the *only* one who can withdraw funds.
- **The Execution Layer (Automation):**
  - `MatchboxRouter.sol`: A single, stateless, and heavily audited utility contract. It is the only contract that talks to the underlying DPMs (e.g., Polymarket's AMM) to perform swaps.
  - **Decentralized Trigger:** An automation network (e.g., Chainlink Automation) that monitors for the resolution of an event and triggers the `Matchbox.sol` contract to execute its next step.

3 Key Innovation: The Price Constraint

The most powerful feature of Matchbox is the user's ability to define *constraints* on their sequence. This solves the $P(B|A)$ volatility problem and provides users with fixed minimum payouts.

A user can define a Matchbox as:

1. **Step 1:** Buy $100 of "YES on Market A."
2. **Step 2:** *IF* Market A resolves YES, redeem proceeds (e.g., $200).
3. **Step 3 (Constraint):** *THEN*, buy "YES on Market B" with the $200 *ONLY IF* the price of "YES on B" is < 0.50.

If the price of "YES on B" has spiked to 0.55 after Market A resolves, the `MatchboxRouter` will attempt the trade, see that the user's constraint would be violated (i.e., they would receive fewer shares than their 'minAmountOut'), and the transaction will atomically revert.

The parlay is broken, but the user's $200 in proceeds remain safe and unspent in their personal `Matchbox` vault, exactly as they intended. This transforms a "dumb parlay" into a smart, automated, and risk-managed trading strategy.

4 Competitive Landscape

Matchbox's architecture provides a clear advantage over all existing solutions by leveraging existing liquidity instead of trying to create new, fragmented markets. See Table 1: Competitive Analysis for more details.

5 Conclusion

Matchbox is the missing automation layer for the decentralized prediction market ecosystem. By separating conditional logic from market liquidity, our protocol allows for the creation of complex, scalable, and safe strategies without a central counterparty. This is the foundation for a new generation of DeFi-native conditional execution, moving far beyond simple parlays into complex, chain-agnostic strategies.

Table 1: Competitive Analysis

Platform	Liquidity Source	Scalability	Key Mechanic
Matchbox	**Underlying DPMs** (e.g., Polymarket)	**Infinite** (No new pools needed)	**Automation Router** (Non-custodial, 1-of-1 vault)
CTF Wrappers (e.g., Predict Shark)	New AMM Pool (Per-Parlay Token)	Very Poor (Fragments liquidity)	Tokenization ($P(A \cap B)$ Token)
DPM Platforms (e.g., Kalshi)	RFQ / Manual (MM provides liquidity)	Very Poor (Not on-chain)	Manual Market Making (Request for Quotation)
Sportsbooks (e.g., DraftKings)	Centralized (Bookmaker)	N/A	”The House” (Permissioned, custodial)