MAM: Maker Auction Markets

Matthew L. Hooft

EqualFi Labs

E-mail: mhooft@equalfilabs.com (Matthew L. Hooft)

Date of this document: 2026-01-02.

Abstract. Maker Auction Markets (MAM) constitute a decentralized settlement paradigm in which makers commit to fully collateralized, time-parameterized Dutch price curves. These curves are immutable once published but may be cancelled without constraint, and each trade executes against exactly one curve under a complete set of on-chain user constraints. The absence of pooled liquidity, orderbooks, or privileged routing processes eliminates the state-dependent price adjustments that enable classical MEV vectors [13, 14]. Because prices evolve according to predetermined schedules rather than reactive reserve dynamics, adversarial behaviors such as sandwiching or backrunning fail to extract value from takers. This paper formally defines the MAM primitive, articulates its invariants, and demonstrates that competition among deterministic curves yields transparent, globally accessible liquidity and predictable execution semantics even in adversarial environments.

1 Introduction

Decentralized exchange designs have historically converged around three architectural families: automated market makers (AMMs), off-chain orderbooks, and solver mediated routing systems. Each approach exhibits structural vulnerabilities that arise from its method of price formation. AMMs update prices as a direct function of reserves, creating predictable reactions that enable sandwich attacks and exposing liquidity providers to impermanent loss [2, 3, 4, 5, 13, 14]. Off-chain orderbooks rely on continuous quote maintenance and incur latency discontinuities, which permit selective or conditional fills unavailable to on-chain participants [7, 6, 16]. Solver networks, including UniswapX, attempt to aggregate liquidity and improve execution but introduce privileged intermediaries who determine routing, ordering, and effective pricing [8, 9, 1, 15]. Despite their differences, these systems share a common property: price depends on mutable state and can be influenced by adversarial timing [13, 18].

Maker Auction Markets (MAM) adopt a different foundation. A maker publishes a fully collateralized Dutch curve whose price schedule is fixed from the moment of commitment and does not react to orderflow, state transitions, or adversarial interference. The curve specifies how units of inventory become eligible for purchase over time, and settlement always occurs against exactly one curve with no composition, no routing, and no cross pool aggregation. Users supply mandatory on-chain constraints, including minimum output, slippage bounds, and expiration parameters, that are enforced deterministically at settlement. Because prices evolve only according to the committed schedule and never in response to prior trades, execution becomes predictable, locally verifiable, and resistant to canonical MEV strategies [13, 14]. Once a curve is published, adversaries cannot manipulate its price, alter its inventory trajectory, or degrade user outcomes through reorderings of the transaction sequence.

This paper introduces the MAM primitive, formalizes its invariants, and analyzes its behavior under adversarial settlement assumptions. We show that deterministic curves, full collateralization, and single curve settlement provide strong correctness and user protection guarantees, enabling transparent, globally addressable liquidity without oracles, shared pools, or privileged matchmakers. MAM is intended as a minimal, mechanism driven foundation for decentralized exchange, offering predictable execution and reduced attack surface relative to reactive pricing models.

2 Preliminaries

This section defines the execution environment, actors, and notation used throughout the paper. The goal is to state the assumptions required for Maker Auction Markets without introducing implementation specific details.

2.1 Execution Environment

We assume a blockchain that provides a deterministic state transition function, a global clock, and atomic transaction execution. Transactions may be reordered or delayed by adversarial block producers. The network does not provide fairness or consistent latency. Smart contracts are able to hold balances of fungible assets and enforce arithmetic invariants over those balances.

Assets are modeled as units of a fungible token type. A token transfer reduces the sender balance and increases the recipient balance. No reliance is placed on special token behavior such as rebasing or fee on transfer.

2.2 Actors

A MAM involves three primary actors.

Makers. A maker owns inventory of a base asset and wishes to sell it for a quote asset. The maker commits this inventory to a deterministic price curve that becomes visible to all participants once created.

Swappers. A swapper seeks to exchange an input asset for an output asset. The swapper supplies constraints that limit acceptable execution outcomes, including minimum output, slippage bounds, and expiration time.

Executors. An executor submits a transaction that fills part of a curve. Executors have no privileged role and cannot influence the price or inventory of a curve. Their only power is to choose which curve to target.

2.3 Curves and Time

Time is represented as a real valued parameter derived from the blockchain clock. A Dutch curve is a function

$$C(t) = \text{price at time } t,$$

where the price interpolates between a start value and an end value over a fixed duration. The curve is created at time t_0, and the price at later times depends only on $t - t_0$.

Curves do not respond to fills or external state. They are commitments rather than dynamic pricing functions. Dutch auctions and related formats have been studied extensively in auction theory [10, 11].

2.4 Inventory and Collateralization

A maker controls an inventory balance I of an asset. A curve may sell at most I, and the mechanism ensures that the entire quantity offered on a curve is fully collateralized at creation time. Inventory is divided into two logical regions: free inventory that is available for new curves and reserved inventory committed to active curves.

The system enforces the invariant

$$\text{free} + \text{reserved} = \text{actual balance}.$$

This invariant prevents a maker from publishing a curve that cannot be settled.

2.5 User Constraints

A swapper specifies a set of constraints that must hold for a fill to be valid. Let X denote the input amount and let Y denote the output amount implied by the curve at settlement time. A fill is valid only if

$$Y \geq \text{minOut},$$

the implied price deviation does not exceed the declared slippage bound, the transaction occurs before the declared deadline, and the swapper accepts the fee asset used in settlement.

These constraints restrict which curves can be used by an executor and prevent adversarial settlement that would harm the user.

2.6 Atomic Settlement

A fill executes as a single atomic state transition. Either all balance changes and curve updates occur or none occur. No intermediate state is exposed to adversarial actors. This property removes several classes of MEV strategies that depend on manipulating intermediate transitions [13, 14].

3 Maker Auction Markets

This section defines the Maker Auction Market primitive. A MAM is characterized by deterministic maker defined price curves, full collateralization of inventory, and atomic settlement against a single curve. The mechanism is intentionally minimal and does not rely on pools, orderbooks, or routing logic.

3.1 Definition of a Maker Auction Market

A Maker Auction Market consists of a set of makers $\mathcal{M}$, a set of assets $\mathcal{A}$, and a set of Dutch curves $\mathcal{C}$. Each curve $c \in \mathcal{C}$ is created by a maker $m \in \mathcal{M}$ and represents a commitment to sell a fixed quantity of a base asset for a quote asset over a predefined price schedule.

Formally, a curve is a tuple

$$c = (m, a_{\text{base}}, a_{\text{quote}}, Q, C(t), t_0, T),$$

where Q is the total base asset committed, $C(t)$ is the price function, t_0 is the creation time, and T is the duration. At time t, the curve quotes a price $C(t - t_0)$.

A curve is immutable once created. No actor can modify Q, the price trajectory $C(t)$, or the duration T.

3.2 Deterministic Price Formation

The price of a curve is determined by a deterministic function. A standard Dutch curve interpolates linearly between a start price P_0 and an end price P_1 over duration T. For $0 \leq \tau \leq T$,

$$C(\tau) = P_0 + (P_1 - P_0)\,\frac{\tau}{T}.$$

This definition removes all state dependent pricing. The price at time t depends only on t and the curve parameters. It does not depend on order flow, reserve changes, or external market conditions.

3.3 Single Curve Settlement

A swap executes against exactly one curve. Let a swapper provide an input amount X. The mechanism selects a fill amount $f \leq Q_{\text{remaining}}$ and produces output

$$Y = f \cdot C(t - t_0).$$

Settlement is atomic. If the swap is valid, $Q_{\text{remaining}}$ is reduced by f. If any constraint fails, no state changes occur.

There is no routing across curves and no composition of multiple curves within a single settlement. This restriction eliminates cross curve price games and simplifies the analysis of adversarial behavior.

3.4 Collateralization Invariants

Let $I_m(a)$ denote the actual on-chain balance of maker m for asset a. Let $F_m(a)$ denote the free balance of a and let $R_m(a)$ denote the reserved balance committed to active curves.

The mechanism enforces the invariant

$$F_m(a) + R_m(a) = I_m(a)$$

for every maker m and asset a.

A curve that commits quantity Q of asset a_{base} must satisfy

$$F_m(a_{\text{base}}) \geq Q.$$

On creation, the mechanism moves Q from free inventory to reserved inventory. This guarantees that every active curve is fully collateralized and that a maker cannot publish a curve that cannot be settled.

3.5 Deterministic Execution Path

Curves evolve only through the passage of time and the reduction of their remaining quantity. The mechanism does not incorporate external price feeds, pool reserves, or solver logic. The state machine contains no nondeterministic transitions and no dependence on off-chain computation.

Let the system state at time t be $S(t)$. A fill produces a state transition

$$S(t) \to S'(t)$$

defined entirely by the curve parameters, the fill amount, and the user constraints. The absence of dynamic pricing or routing ensures that adversarial actors cannot influence price discovery beyond choosing which curve to target.

3.6 Commitment Based Liquidity

Liquidity in a MAM is expressed as commitments rather than reactive reserves. A maker does not quote a price conditional on inventory. Instead, the maker publishes a schedule that becomes more attractive as time progresses or less attractive in the case of ascending curves. This model shifts price discovery away from instantaneous state and toward explicit time dependent commitments.

This property creates a predictable and transparent liquidity landscape. Participants can observe how much inventory is available at each price and at what time those prices will take effect.

4 User Constraints and Protection Model

A Maker Auction Market allows users to specify strict limits on acceptable execution outcomes. These constraints are enforced entirely within the settlement mechanism and apply to every fill without exception. This section formalizes the constraint model and its role in determining the validity of a settlement.

4.1 Constraint Set

A swapper provides an input amount X of a base asset and specifies a set of constraints

$$\mathcal{C} = (\text{minOut}, \text{slippage}, \text{deadline}, \text{feeAssetAllowed}).$$

Let t denote the settlement time and let $C(t - t_0)$ denote the curve price at that time. A fill that consumes base amount f yields output

$$Y = f \cdot C(t - t_0).$$

A fill is valid only if all constraints in $\mathcal{C}$ are satisfied.

4.2 Minimum Output

The minimum output constraint requires

$$Y \geq \text{minOut}.$$

This guarantees that the swapper never receives less than a threshold chosen prior to execution. The mechanism does not permit partial satisfaction of this condition. Either the fill meets the requirement or it is rejected.

4.3 Slippage Bound

A swapper may specify a slippage tolerance expressed as a fractional bound on deviation from the reference curve price at settlement time. Let $P_{\text{eff}} = Y/f$ denote the effective execution price and let $P_{\text{ref}} = C(t - t_0)$ denote the reference curve price. The slippage condition is

$$\left| \frac{P_{\text{eff}} - P_{\text{ref}}}{P_{\text{ref}}} \right| \leq \text{slippage}.$$

This protects the swapper against deterioration in execution quality relative to the deterministic curve price.

4.4 Deadline

A fill must occur before a user specified expiration time. Let t_{deadline} denote this deadline. The constraint is

$$t \leq t_{\text{deadline}}.$$

This prevents delayed settlement in periods of adverse market conditions or network congestion.

4.5 Fee Asset Compatibility

The mechanism may require fees to be paid in either the input asset or the output asset. A swapper declares acceptable fee asset types in the field `feeAssetAllowed`. A fill is permitted only if the fee asset used during settlement belongs to this declared set.

This prevents executors from choosing a curve whose fee configuration is incompatible with the swapper's intent.

4.6 Validity Condition

Let a curve c have remaining quantity Q_{rem}. Let a fill attempt consume $f \leq \min(X, Q_{\text{rem}})$. The settlement is valid if and only if all of the following hold:

$$\begin{aligned}
&Y = f \cdot C(t - t_0),\\
&Y \geq \text{minOut},\\
&\left|\frac{Y/f - C(t - t_0)}{C(t - t_0)}\right| \leq \text{slippage},\\
&t \leq t_{\text{deadline}},\\
&\text{feeAsset} \in \text{feeAssetAllowed}.
\end{aligned}$$

If any condition fails, the settlement is rejected and the system state does not change.

4.7 Protection Against Adversarial Executors

Executors cannot deviate from the constraint set. They cannot reduce output, modify fee direction, exceed slippage limits, or schedule fills after the declared deadline. Their only degree of freedom is the choice of curve c to target. The validity checks ensure that this choice cannot harm the swapper beyond the constraint envelope chosen at the time the intent was formed.

5 Security Model

This section defines the adversarial model for Maker Auction Markets and states the safety properties that the mechanism must provide. The analysis considers malicious makers, malicious executors, malicious swappers, MEV aware block producers, and adversarial network conditions. The objective is to reason about correctness and user protection under the strongest practical assumptions [13, 14].

5.1 Adversarial Setting

The system operates in an environment where block producers may reorder, delay, or exclude transactions. The mempool is observable, and adversaries may coordinate off-chain. Front running, back running, and censorship attacks are explicitly allowed within the model [13, 18]. No trust is placed in any executor or maker, and all interactions are treated as potentially adversarial.

Assets behave as standard fungible tokens. The mechanism does not normalize or compensate for exotic behavior such as rebasing or fee on transfer. Makers bear responsibility for understanding any nonstandard asset they choose to list.

5.2 Adversarial Actors

We consider the following adversarial classes.

Malicious Makers. A malicious maker may attempt to publish misleading curves, withdraw inventory before settlement, or design curve schedules intended to harm swappers.

Malicious Executors. A malicious executor may attempt to settle a swap in a manner that violates user constraints, misprice a fill, or manipulate the fee direction used during settlement.

Malicious Swappers. A malicious swapper may attempt to manipulate input amounts, propose unfillable orders, or exploit timing conditions to disrupt settlement.

MEV Adversaries. An adversary with mempool visibility may attempt to reorder transactions to extract value through sandwiching, back running, or related price manipulation strategies [13, 14].

Adversarial Network. The network may delay messages or propagate transactions inconsistently. No delivery or ordering guarantees are assumed.

5.3 Core Safety Guarantees

A Maker Auction Market must satisfy the following safety properties.

Safety 1. Full Collateralization. A maker cannot publish a curve that exceeds available inventory. Let $I_m(a)$ denote the actual balance and let Q denote the committed amount. The mechanism enforces

$$Q \leq I_m(a),$$

and ensures that the committed amount is reserved for the lifetime of the curve.

Safety 2. User Constraint Enforcement. For any fill attempt, the mechanism verifies that all user constraints are satisfied before any state transition occurs. If any condition fails, the fill is rejected without updating balances or consuming curve quantity.

Safety 3. Immutable Pricing. Once a curve is created, its price schedule is immutable. Executed prices are determined entirely by the creation parameters and the settlement time. Adversaries cannot influence prices except by choosing which curve to target.

Safety 4. Atomic Settlement. A fill either completes entirely or has no effect on system state. No partial execution path is visible to adversarial actors, and no intermediate quantities can be manipulated.

5.4 Resistance to Malicious Makers

A malicious maker cannot create an under collateralized curve or alter a curve after creation. The reserved inventory invariant prevents overselling, and immutability ensures that execution conditions cannot be degraded after a swapper has formed an intent. A harmful maker can publish unattractive curves, but such curves are ignored by swappers and executors.

5.5 Resistance to Malicious Executors

Executors cannot modify prices, alter fee direction, or override slippage conditions. All settlement conditions are verified on-chain. An executor cannot force a swapper to accept less favorable terms than those explicitly declared. Executors have no influence over curve evolution and cannot alter the remaining quantity or price trajectory.

5.6 Resistance to Malicious Swappers

If a swapper provides insufficient input, violates deadlines, or specifies contradictory constraints, the settlement fails early and no state is updated. Swappers cannot force makers to accept unfavorable execution and cannot violate internal accounting invariants.

5.7 MEV Considerations

Because price depends only on time and not on state transitions, MEV strategies that rely on reactive pricing provide no advantage. A sandwich attack is ineffective since the curve price does not respond to prior trades [13]. Back running produces no slippage effect and does not extract value from either party. The remaining MEV surface is limited to transaction ordering competition, which does not degrade user outcomes or violate safety properties [14, 18].

5.8 Summary of Security Properties

Under the adversarial assumptions above, a Maker Auction Market provides the following guarantees:

- every curve is fully collateralized at creation time,
- users never receive output below their declared minimum,
- adversaries cannot alter curve prices after creation,
- settlement is atomic and exposes no intermediate state,
- state reactive MEV strategies are ineffective.

6 Market Structure Properties

Maker Auction Markets create a distinctive market structure that differs from automated market makers, off-chain orderbooks, and solver based routing systems. This section analyzes the economic and structural properties that emerge when multiple deterministic curves coexist and compete for order flow. The focus is on transparency, liquidity expression, and the nature of price discovery in a MAM environment.

6.1 Emergent Price Discovery

A MAM does not compute market prices from reserves and does not maintain a central limit orderbook. Instead, price discovery emerges from the interaction of multiple published curves. Each curve provides a time dependent price path. Participants observe the set of active curves

$$\{C_i(t)\}_{i\in\mathcal{I}}$$

and select the curve that best satisfies their constraints.

At any time t, the effective market price is shaped by the most competitive curve among those that remain collateralized and unfilled. As time progresses, descending curves become more attractive and may cross the valuations of swappers. The result is a transparent and continuous form of price discovery that does not depend on oracle updates, reserve movements, or solver computations.

6.2 Liquidity as Explicit Commitments

Liquidity in a MAM is expressed through explicit commitments rather than reactive reserves. For each curve, both the committed quantity Q_i and the price path $C_i(t)$ are visible to all participants. This contrasts with AMMs, where the effective price depends on the current reserve ratio, and with off-chain orderbooks, where quote visibility can be incomplete due to latency or stale updates [2, 3, 4, 5, 7, 6, 16].

The explicit nature of commitments enables participants to reason about future availability. A swapper can observe when and at what price a curve will become attractive, which yields a predictable liquidity surface and removes ambiguity about execution conditions.

6.3 Competitiveness Among Makers

Makers compete by choosing which curves to publish, when to publish them, and how to manage their Desk's inventory over time. Although curves are immutable once created, makers may cancel an auction at any moment and repost a new one. This gives them flexibility to respond to market conditions, shift inventory between the two sides of the Desk, or reorganize their liquidity as circumstances change.

Competition therefore arises from the skill with which makers design, schedule, and adjust their auction inventory rather than from continuous repricing. A maker with a well curated set of auctions can maintain tighter effective spreads, denser curve coverage, and more predictable execution quality. The mechanism rewards makers who treat liquidity as an actively managed resource instead of a static pool. This is analogous in spirit to optimal execution and inventory management problems studied in traditional market microstructure [17].

6.4 Liquidity Granularity and Curve Composition

Makers may divide their liquidity into many small auctions or consolidate it into a few larger ones. Fine grained decomposition allows makers to approximate the behavior of an orderbook, with layered limit style coverage at varying prices and times. Coarse decomposition produces broader liquidity bands with fewer individual curves but larger notional movement per fill.

Because makers may cancel and repost auctions at any cadence, they can shape their Desk's behavior with considerable precision. A Desk may combine flat auctions for limit style execution, narrow Dutch curves for price discovery, and time staggered auctions that adapt to volatility. This flexibility allows makers to encode sophisticated strategies, including dynamic rebalancing between sides of the Desk, inventory shading, iceberg style exposures, and time based replenishment.

All such behavior emerges from the same primitive. The protocol introduces no special strategy rules and no privileged execution paths. The expressiveness arises solely from how makers compose and schedule their own liquidity.

6.5 Absence of Routing and Its Effects

A MAM does not attempt to split a swap across multiple curves or route a swap through multiple paths. This design choice simplifies execution and avoids the optimization problems that arise in solver based systems [8, 9, 15]. It also prevents situations in which routing decisions shape the effective execution price or introduce implicit favoritism.

Because each transaction interacts with exactly one curve, participants understand precisely which maker they traded against and how the price was determined.

6.6 Comparison to Automated Market Makers

Automated market makers provide continuous liquidity but rely on state dependent pricing. Reserve changes caused by a trade affect subsequent trades, which leads to price reactivity and exposure to sandwich attacks [2, 3, 4, 5, 13]. MAM curves do not respond to state transitions and therefore avoid this class of behavior.

Liquidity providers in AMMs face impermanent loss when external markets move. MAM makers do not incur impermanent loss, since their exposure is determined entirely by their chosen price path. The risk profile shifts from reactive exposure to a commitment based model that resembles posting limit orders.

6.7 Comparison to Orderbooks and RFQ Systems

Traditional orderbooks rely on rapid quote updates and continuous repricing. This creates latency races and confers structural advantages on participants with superior network access or faster update capabilities [7, 6, 16]. In such environments, the visible state of the book is often an approximation of true liquidity, since stale quotes, fleeting orders, and selective cancellations introduce uncertainty about what inventory is genuinely available.

RFQ systems attempt to mitigate these issues by allowing a taker to solicit quotes from intermediaries who respond with executable prices. These intermediaries often rely on private information, proprietary routing logic, or off-chain agreements that are not observable to the taker [8, 9]. The resulting execution quality depends on the integrity and skill of the quoting party, and on the timeliness of the quote relative to market conditions.

A MAM differs from both architectures in several respects. First, a MAM does not require continuous quote maintenance. Once a curve is published, its price and quantity remain fixed for its lifetime, and execution cannot deteriorate as a result of delayed updates or strategic quote withdrawal. Second, a MAM does not rely on intermediaries to compute prices. All price information is encoded directly in immutable curves that are visible to all participants.

Although MAM does not require continuous maintenance, effective liquidity provision may involve active management of curve placement. A maker who posts a sequence of flat curves can approximate the behavior of a limit orderbook and can reproduce RFQ-like behavior by arranging discrete auctions that correspond to price levels a taker might request. This flexibility allows MAM to span the expressive space of orderbooks and RFQ systems while avoiding their dependence on mutable quotes, privileged intermediaries, and private off-chain negotiation. The essential difference is that MAM expresses these strategies through transparent and immutable commitments rather than through rapid repricing or intermediary coordination.

6.8 Predictability and Transparency

The deterministic nature of curves allows all actors to compute future price availability. This stands in contrast to mechanisms where supply, demand, or off-chain interactions determine execution quality. Because MAM curves do not change after publication, every participant observes the same liquidity landscape and can verify that settlement matched the curve parameters exactly.

This transparency contributes to fairness and simplifies external auditing. Any deviation from the stated price schedule would be detectable, and the absence of mutable pricing removes many common sources of complexity in verification.

7 Extensions and Variants

Maker Auction Markets are defined by a minimal set of rules, yet the primitive admits several extensions that preserve the core properties of determinism, full collateralization, and single curve settlement. This section outlines possible variants that expand the expressive power of the model while retaining its foundational guarantees.

7.1 Multi Curve Interaction

The base MAM model restricts each settlement to a single curve. A natural extension is to allow a swapper to specify an ordered list of curves to be consumed sequentially, subject to the same constraint set. This creates a form of piecewise execution while maintaining deterministic pricing at each step. Any such extension must ensure that each curve is individually collateralized and that the atomicity of each partial fill remains intact, so that no intermediate state can be exploited by an adversarial executor.

7.2 Batch or Interval Settlement

Another variant is batch settlement, where multiple swappers interact with a curve at the same timestamp. Because the price at time t is fixed, all fills in the batch execute at the same price. This produces a continuous time model that remains compatible with discrete batch auction clearing. The primary design requirement is to ensure that adversarial reordering within the batch does not alter execution quality or create opportunities for selective settlement [12].

7.3 Alternative Price Schedules

The basic model uses linear Dutch interpolation. Curves may instead employ piecewise linear schedules or monotonic convex or concave shapes, provided that the schedule is deterministic and specified at creation time. Makers may choose these shapes to control how liquidity becomes attractive over time. The essential requirement is that price remains a pure function of time and is not influenced by state transitions or prior fills [10, 11].

7.4 Dual Funded Curves

A maker may publish a curve that accepts input from two complementary assets. For example, the maker could simultaneously commit to buy one asset and sell another according to a shared time dependent schedule. This creates a symmetric form of auctioning in which two sides of a market are expressed simultaneously. The mechanism must still enforce full collateralization for both assets and ensure that each side of the curve behaves independently with respect to settlement.

7.5 Cross Chain Variants

A MAM can be extended to cross chain settings where the output asset resides on a different ledger. In such cases, settlement must be conditioned on a verifiable cross chain proof of payment. The deterministic nature of curves simplifies the design, since the on-chain component does not require pricing information from the remote chain. The mechanism must only verify that the fill corresponds to an authorized external condition, such as a proof of lock or proof of burn [20].

7.6 Privacy Oriented Variants

A MAM may incorporate privacy preserving techniques by obscuring the identities of makers or swappers. Provided that curve parameters remain public and verifiable, hiding counterparty identity does not affect settlement logic. Zero knowledge proofs could be used to demonstrate that a curve is properly collateralized without revealing the identity of the maker, and similar techniques could hide which swapper initiated a fill [19].

7.7 Extensions to Non Fungible Assets

Although the MAM primitive is described for fungible assets, it generalizes naturally to settings where each unit of inventory is unique. A maker may publish a curve over a sequence of non fungible items with deterministic pricing. The primary challenge is addressing partial fills. If items are indivisible, the curve becomes a schedule for a sequence of discrete auctions, and the mechanism must ensure that fills select valid items and preserve the atomicity of each auction event.

7.8 Synchronized Curves and Market Wide Events

Multiple curves may be linked through shared time parameters or synchronized release conditions. For example, a maker may require that several assets become available only after a specific timestamp. This allows coordinated supply schedules and the expression of multi asset strategies, while preserving determinism and the independence of pricing rules.

7.9 Summary

The extensions described above maintain the defining properties of a MAM: deterministic price evolution, full collateralization, and atomic settlement against a single pricing rule. These variants illustrate that the primitive can support a broad range of market structures without compromising its analytical simplicity or its security model.

8 Related Work

Maker Auction Markets relate to several strands of research in decentralized exchange design, auction theory, and automated pricing mechanisms. This section reviews the most relevant areas and contrasts them with the MAM primitive. The goal is to situate MAM within the broader literature on on-chain markets and commitment based economic mechanisms.

8.1 Automated Market Makers

Automated market makers provide liquidity through reserve based pricing functions. Canonical examples include constant product markets, constant sum markets, and stable swap functions [2, 3, 4, 5]. The price in these systems is a deterministic function of the reserves, and the reserves evolve endogenously as trades occur. This coupling between price and state creates predictable reactions that adversarial actors can exploit. Sandwich attacks, long term tracking strategies, and various forms of state reactive MEV have been analyzed extensively in the literature [13, 14].

Liquidity providers in AMMs also face impermanent loss, since their exposure is determined by the shape of the pricing curve rather than by explicit limit style commitments. These systems provide path independent execution but often require complex incentive engineering to manage the risk borne by liquidity providers.

MAMs differ from AMMs along several dimensions. First, MAMs use time based pricing schedules rather than reserve based pricing. A curve does not react to prior fills, and therefore adversarial state manipulation provides no pricing advantage. Second, makers in MAMs do not pool inventory. Each maker expresses liquidity as an explicit commitment, avoiding the interdependent risks associated with shared reserves. Third, the absence of reserve dynamics eliminates impermanent loss. The exposure of a maker is entirely determined by the price path chosen at creation time.

8.2 On-chain Orderbooks

On-chain orderbooks attempt to reproduce centralized limit orderbook mechanics within a blockchain environment. Makers post limit orders directly to the chain, and takers select which orders to fill. These systems typically suffer from latency related issues, because quotes must be updated rapidly to remain competitive [7, 6, 16]. Slow block times and heterogeneous transaction propagation lead to stale orders, quote flickering, and opportunities for selective execution by participants with better network placement.

The visibility of orderbook depth is often incomplete, since the best quotes may exist only transiently and can be cancelled before settlement. Continuous repricing introduces race conditions, and the resulting execution quality depends heavily on a participant's ability to update orders quickly.

MAMs differ in that curves do not require continuous maintenance. Once a curve is published, its price and quantity remain fixed until consumption or cancellation. This avoids the latency races inherent in orderbooks. At the same time, a sequence of flat curves can approximate orderbook behavior, allowing MAMs to reproduce many of the expressive features of a limit orderbook without inheriting its temporal fragility.

8.3 RFQ and Solver Based Protocols

Request for quote systems and solver driven routing protocols rely on off-chain actors who compute execution paths or supply prices. Examples include CoW Protocol, 1inch Fusion, and UniswapX [8, 9, 1]. These intermediaries may have private access to order flow, strategic inventory, or bilateral routing arrangements. Execution quality can vary significantly depending on the trustworthiness and sophistication of the intermediary. Many RFQ and solver based systems operate in partially opaque environments, where the taker cannot easily verify how prices were selected or which liquidity sources were considered. UniswapX in particular combines Dutch orders with an off-chain filler set that competes to route across liquidity sources [1], whereas MAM internalizes the Dutch schedule directly into on-chain settlement without delegating routing to an external solver network.

Unlike these systems, a MAM does not rely on intermediaries to form prices. All pricing information is contained in deterministic curves that are visible to all participants. Makers do not need to respond to RFQ messages, and solvers do not route across liquidity sources. Although effective use of a MAM may involve careful curve management, the computation of prices does not depend on private information or off-chain negotiation. The transparency of published curves allows takers to assess execution quality directly.

8.4 Batch Auctions

Batch auction models clear all orders submitted within a discrete interval at a single uniform price. This approach reduces several forms of adversarial extraction by eliminating the ordering effects that arise in continuous trading [12]. Batch auctions have been proposed as a mitigation for MEV and have been studied both in theory and in prototype systems. However, they require complex clearing logic, explicit coordination across participants, and careful incentive engineering to prevent undesirable strategic behavior.

MAMs do not implement batch clearing, yet they can reproduce some of its effects. If multiple fills occur at the same timestamp, each executes at the same price because the curve price at time t is fixed. Unlike batch auctions, MAMs do not coordinate across participants, do not compute a uniform clearing price, and do not require knowledge of aggregate demand. Each fill remains atomic and independent.

8.5 Commitment Based Market Mechanisms

A number of auction formats rely on commitments that cannot be altered once published. Time locked auctions, sealed bid auctions, and descending price auctions all incorporate the idea of deterministic pricing that evolves independently of external behavior [10, 11]. MAMs generalize this idea by embedding deterministic price functions directly into the exchange mechanism. The focus is not on a single auction event but on a continuous environment where many independent curves coexist and compete for order flow. This creates a persistent market structure built from overlapping commitments rather than from a centralized matching procedure.

8.6 MEV and Adversarial Analysis

There is extensive research on miner extractable value and adversarial strategies that exploit price reactivity in decentralized systems. Many studies focus on AMMs, where reactive pricing creates profitable deviations under adversarial ordering [13, 14]. Front running, back running, sandwiching, and liquidity draining strategies all rely on manipulating the state in ways that influence future prices.

Because MAMs rely on time based rather than state based pricing, these strategies provide limited advantage. A front run cannot alter the price a swapper receives, since the curve does not react to prior trades. A back run cannot generate beneficial price movement, since the curve trajectory is predetermined. The remaining MEV surface is limited to standard transaction ordering competition, which does not affect correctness or degrade user outcomes [13, 18, 14].

8.7 Summary

MAMs occupy a conceptual position between traditional auction designs and decentralized exchange protocols. They share the commitment structure of auctions while operating continuously like a market. They provide predictable execution without intermediaries and avoid the state dependent pricing that creates many vulnerabilities in prior work. By treating liquidity as a set of explicit commitments rather than a reactive state, MAMs define a distinct point in the design space of on-chain markets.

9 Conclusion

Maker Auction Markets provide a minimal and deterministic foundation for decentralized exchange. By replacing state dependent pricing with time dependent commitments, MAMs eliminate the reactive behavior that enables many adversarial strategies in automated market makers, orderbooks, and solver driven routing systems [2, 3, 4, 5, 7, 6, 8, 9, 13]. Makers publish fully collateralized curves that remain immutable once created, and users interact with these curves under strict on-chain constraints. The result is a market structure in which settlement is transparent, predictable, and resistant to manipulation.

The analysis presented in this paper shows that the core properties of MAMs yield strong safety guarantees. Makers cannot oversell inventory, executors cannot degrade user outcomes, and the price of a curve is determined solely by its creation parameters and the passage of time. These invariants support a form of price discovery driven by competition among deterministic curves rather than by reactive state transitions or privileged intermediaries.

MAMs constitute a general mechanism capable of supporting a wide range of market behaviors while preserving analytical simplicity. The primitive can be extended to batch style settlement, cross chain execution, privacy oriented designs, and richer price schedules without altering its foundational invariants. As decentralized markets continue to evolve, MAMs provide a direction for constructing exchange mechanisms that emphasize fairness, verifiability, and structural resistance to adversarial behavior.

A Formal Adversary Model

This appendix presents a formal adversary model for Maker Auction Markets. The goal is to specify the capabilities, information sets, and constraints of adversarial actors, and to provide a basis for reasoning about correctness and user protection under strong adversarial assumptions. The model follows standard practice in the analysis of decentralized exchange protocols and on-chain settlement systems [13, 14].

A.1 System Model

We assume a blockchain that provides atomic transaction execution, deterministic state transitions, and a global notion of time derived from block timestamps. Transactions may be reordered, delayed, or excluded by block producers. The mempool is public and may be monitored by adversarial actors. No fairness guarantees are assumed for message delivery or transaction inclusion.

Let S denote the global system state, including all maker balances, reserved inventory, active curves, and their remaining quantities. A transaction τ induces a state transition

$$S \xrightarrow{\tau} S'$$

if and only if τ passes all validity checks defined by the MAM mechanism.

A.2 Adversary Capabilities

The adversary controls any subset of makers, swappers, and executors, and may coordinate their actions. The adversary also influences block production through the ability to choose transaction ordering within a block.

The adversary is granted the following capabilities:

- full visibility of all pending transactions in the mempool,
- arbitrary reordering, delaying, or exclusion of transactions when acting as block producer,
- arbitrary choice of curve to target when acting as an executor,
- creation of curves with any valid deterministic price schedule when acting as a maker,
- ability to cancel its own curves at any time before settlement,
- off-chain coordination among adversarial identities.

The adversary is not granted the ability to:

- modify the parameters of an existing curve,
- settle a curve without providing the required assets,
- bypass on-chain constraint checks,
- alter the global clock or block timestamp,
- introduce nondeterministic behavior into state transitions.

A.3 Information Sets

At any time t, the adversary knows:

- the full set of active curves and their remaining quantities,
- all curve parameters, including price schedules and timestamps,
- all user submitted constraints for transactions in the mempool,
- pending fills and cancellations visible in the mempool,
- the system state observable from the blockchain.

The adversary does not know:

- private keys of honest users,
- future transactions that have not yet entered the mempool,
- future block timestamps unless the adversary controls the next block.

A.4 Adversarial Goals

The adversary may attempt to achieve any of the following:

- induce a user to receive output below their declared minimum,
- exceed a user's slippage bound,
- settle after the declared deadline,
- deplete a maker's inventory without payment,
- execute a profitable front run or back run,
- manipulate the apparent availability of liquidity,
- exploit intermediate state during settlement.

The adversary succeeds only if the mechanism's safety properties or invariants can be violated.

A.5 Correctness Under Adversarial Ordering

Let $\mathcal{A}$ denote the adversary and let σ denote an arbitrary transaction ordering chosen by $\mathcal{A}$. The mechanism satisfies correctness under adversarial ordering if:

$$\forall \sigma, \forall \tau \in \sigma, \quad \tau \text{ is applied to } S \text{ if and only if } \tau \text{ satisfies all MAM validity conditions.}$$

This property ensures that correctness is independent of ordering. A valid fill cannot be invalidated by reordering, and an invalid fill cannot be made valid through adversarial sequencing.

A.6 MEV Surface

Under this model, the only MEV surface available to the adversary is transaction ordering competition. Since curve prices do not depend on state transitions, an adversary cannot improve execution by manipulating intermediate state. Sandwiching and back running do not affect the price that a swapper receives. The structure of MAMs therefore reduces MEV opportunities to inclusion and ordering effects, none of which degrade user defined execution quality [13, 14, 18].

A.7 Summary

The formal adversary model defined in this appendix captures a broad class of behaviors, including malicious makers, swappers, executors, and block producers. Despite the strength of these assumptions, the deterministic structure of MAMs preserves correctness and user protection. Immutability of curve parameters, full collateralization, and atomic settlement limit the adversary to ordering control, and this capability does not permit violation of the mechanism's invariants.

B State Machine Specification

This appendix formalizes the Maker Auction Market as a state machine. The goal is to provide a precise definition of the system state, the set of permitted events, the transition rules for each event, and the invariants that must hold for all reachable states.

B.1 State Variables

Let the global state at time t be denoted by $S(t)$. The state consists of the following components.

- **Maker Balances:** For each maker m and asset a, the system stores the actual balance $I_m(a)$.
- **Reserved Inventory:** For each maker m and asset a, the system stores the quantity $R_m(a)$ committed to active curves.
- **Free Inventory:** For each maker m and asset a, the system stores the free quantity $F_m(a)$, where
$$F_m(a) + R_m(a) = I_m(a).$$
- **Active Curves:** Each active curve c is represented as a tuple
$$c = (m, a_{\text{base}}, a_{\text{quote}}, Q_{\text{total}}, Q_{\text{rem}}, C(t), t_0, T, \text{feeAsset}),$$
where Q_{rem} is the remaining quantity.
- **System Clock:** The protocol reads a global time value from block timestamps, denoted t.

B.2 Events

The state machine exposes four classes of events.

- **CreateCurve**(m, a$_{\text{base}}, a_{\text{quote}}, Q, C(t), t_0, T, \text{feeAsset}$)
- **CancelCurve**(m, c)

- **FillCurve**(c, f, C)
- **Transfer**(x, y, a, q) (standard token transfers)

Fills must respect the user constraint set $\mathcal{C} = (\text{minOut}, \text{slippage}, \text{deadline}, \text{feeAssetAllowed})$. Transfers behave as standard token movements and are included for completeness.

B.3 Transition Rules

Each event induces a deterministic state transition. Let $S \to S'$ denote a valid transition.

CreateCurve. Creation is valid if and only if:

$$F_m(a_{\text{base}}) \geq Q.$$

If valid, the transition is:

$$\begin{aligned} F_m(a_{\text{base}}) &\leftarrow F_m(a_{\text{base}}) - Q, \\ R_m(a_{\text{base}}) &\leftarrow R_m(a_{\text{base}}) + Q, \\ \text{ActiveCurves} &\leftarrow \text{ActiveCurves} \cup \{c\}. \end{aligned}$$

CancelCurve. Cancellation is valid if the curve belongs to the maker:

$$c.m = m.$$

If valid:

$$\begin{aligned} R_m(a_{\text{base}}) &\leftarrow R_m(a_{\text{base}}) - Q_{\text{rem}}, \\ F_m(a_{\text{base}}) &\leftarrow F_m(a_{\text{base}}) + Q_{\text{rem}}, \\ \text{ActiveCurves} &\leftarrow \text{ActiveCurves} \setminus \{c\}. \end{aligned}$$

FillCurve. A fill request with amount f is valid only if:

$$f \leq Q_{\text{rem}}, \quad t \leq \text{deadline}, \quad \text{feeAsset} \in \text{feeAssetAllowed},$$
$$Y = f \cdot C(t - t_0) \geq \text{minOut},$$
$$\left| \frac{Y/f - C(t-t_0)}{C(t-t_0)} \right| \leq \text{slippage}.$$

If valid:

$$\begin{aligned} Q_{\text{rem}} &\leftarrow Q_{\text{rem}} - f, \\ R_m(a_{\text{base}}) &\leftarrow R_m(a_{\text{base}}) - f, \\ I_m(a_{\text{quote}}) &\leftarrow I_m(a_{\text{quote}}) + Y, \\ I_{\text{swapper}}(a_{\text{quote}}) &\leftarrow I_{\text{swapper}}(a_{\text{quote}}) - Y. \end{aligned}$$

If $Q_{\text{rem}} = 0$, the curve is removed.
If any constraint fails, the event has no effect:

$$S' = S.$$

Transfer. Token transfers update balances in the standard way:

$$I_x(a) \leftarrow I_x(a) - q, \quad I_y(a) \leftarrow I_y(a) + q.$$

B.4 Invariants

For all reachable states, the following invariants hold.

Invariant 1. Full Collateralization. For every maker m and asset a,

$$F_m(a) + R_m(a) = I_m(a).$$

Invariant 2. No Overselling. For every active curve c,

$$Q_{\text{rem}} \leq Q_{\text{total}}.$$

Invariant 3. Immutability of Curve Parameters. For every active curve c,

$$(c.m, c.a_{\text{base}}, c.a_{\text{quote}}, c.Q_{\text{total}}, c.C(t), c.t_0, c.T)$$

remain constant throughout the lifetime of the curve.

Invariant 4. Atomic Execution. Every FillCurve event satisfies:

$$S' = S \quad \text{or} \quad S' = \text{ApplyFill}(S, c, f, \mathcal{C}).$$

No intermediate state is visible.

Invariant 5. Deterministic Price Evaluation. For any curve c and any time t,

$$\text{Price}(c, t) = C(t - t_0),$$

independent of all prior fills or state transitions.

B.5 Summary

This state machine specification formalizes the MAM mechanism as a deterministic process with well defined events and invariants. All transitions depend solely on curve parameters, user constraints, and the passage of time. No event introduces nondeterminism or hidden state, and the invariants guarantee that full collateralization, atomicity, and price immutability are preserved for all reachable states.

C Illustrative Examples

This appendix presents several concrete examples that illustrate how Maker Auction Markets operate in practice. The examples demonstrate curve creation, competitive price selection, user constraint enforcement, and the effects of deterministic pricing. All numbers are simplified for clarity.

C.1 Example 1: Execution Against a Single Descending Curve

Consider a maker who wishes to sell $Q = 10$ units of asset A for asset B. The maker publishes a descending Dutch curve with:

$$P_0 = 110 \quad (\text{units of } B \text{ per unit of } A),$$

$$P_1 = 90, \quad T = 1000 \text{ seconds}.$$

The price schedule is:

$$C(\tau) = 110 - 0.02\tau, \quad 0 \leq \tau \leq 1000.$$

Assume the curve is created at time $t_0 = 0$. A swapper submits an order at time $t = 300$ seconds with:

$$X = 2, \quad \text{minOut} = 180, \quad \text{slippage} = 0.01, \quad t_{\text{deadline}} = 400.$$

The effective curve price is:

$$C(300) = 110 - 0.02 \cdot 300 = 104.$$

The output is:

$$Y = 2 \cdot 104 = 208.$$

All constraints are satisfied:

$$208 \geq 180, \quad \left| \frac{104 - 104}{104} \right| = 0 \leq 0.01, \quad 300 \leq 400.$$

The fill is valid. The remaining quantity becomes $Q_{\text{rem}} = 8$.

C.2 Example 2: Competition Between Multiple Curves

Suppose two makers publish curves for the same trading pair.

- Curve c_1: descending from 105 to 95 over 600 seconds.
- Curve c_2: descending from 120 to 100 over 1200 seconds.

At time $t = 300$ seconds after the creation of both curves, the prices are:

$$C_1(300) = 105 - \frac{10}{600} \cdot 300 = 100,$$
$$C_2(300) = 120 - \frac{20}{1200} \cdot 300 = 115.$$

A swapper with a standard constraint set selects the curve offering the best price that satisfies all conditions. Since $C_1(300) = 100$ is more attractive, the executor targets c_1. The MAM mechanism does not route across curves or split the fill. The swapper trades entirely against c_1 at a transparent, deterministic price.

C.3 Example 3: Slippage and Deadline Enforcement

A curve has price schedule:

$$C(\tau) = 200 - 0.1\tau.$$

A swapper submits a fill request at time $t = 500$ seconds with:

$$f = 1, \quad \text{slippage} = 0.005, \quad t_{\text{deadline}} = 480.$$

The curve price is:

$$C(500) = 200 - 50 = 150.$$

Two constraints immediately fail:

$$500 > t_{\text{deadline}}, \quad \text{and slippage cannot be evaluated without a valid timestamp.}$$

The fill is rejected without state change. The executor cannot force execution past the deadline.

C.4 Example 4: Sequential Auction Scheduling

A maker wants to approximate a limit orderbook with discrete price levels. The maker publishes three flat curves:

$$\begin{aligned} C_1(t) &\equiv 102 \quad \text{with} \quad Q_1 = 3, \\ C_2(t) &\equiv 104 \quad \text{with} \quad Q_2 = 5, \\ C_3(t) &\equiv 106 \quad \text{with} \quad Q_3 = 2. \end{aligned}$$

A swapper seeking the best execution receives:

$$102 \text{ for the first 3 units}, \quad 104 \text{ for the next 5 units}, \quad 106 \text{ for the last 2 units}.$$

Each fill is atomic and evaluated independently. No curve affects the pricing of any other curve. This demonstrates how MAMs can emulate layered limit order strategies without requiring a mutable orderbook.

C.5 Example 5: Effect of Cancelling and Reposting

A maker publishes a curve with:

$$P_0 = 150, \quad P_1 = 130, \quad T = 600, \quad Q = 4.$$

After 200 seconds, the maker cancels the curve with $Q_{\text{rem}} = 3$. The mechanism returns the remaining quantity to free inventory. The maker then publishes a new curve at a different schedule.

This example illustrates that makers may adapt inventory and revise their strategies without undermining determinism. The cancellation does not alter any historical execution price or affect other curves.

C.6 Summary

These examples demonstrate how deterministic curves, strict constraints, and atomic settlement govern execution in a MAM. The mechanism is expressive enough to model a wide range of trading behaviors while maintaining predictable and verifiable pricing at all times.

D Limitations and Open Questions

This appendix describes several limitations of the Maker Auction Market model and outlines open questions that merit further study. Although MAMs provide a deterministic and protective mechanism for decentralized exchange, they introduce practical and theoretical challenges that require careful analysis. The goal is to identify the boundaries of the model and highlight opportunities for future research.

D.1 Infrastructure Requirements

MAMs rely on the availability of infrastructure for curve discovery and indexing. The mechanism itself does not prescribe how participants locate active curves or evaluate their relative attractiveness. In practice, users and executors require access to reliable indexing services that present the global set of curves in a timely manner. The performance and decentralization of such indexing layers remain open concerns.

D.2 Inventory Management Complexity

Although MAMs do not require continuous repricing, effective participation may involve active management of auction schedules. Makers who wish to maintain tight spreads or approximate orderbook style coverage must coordinate multiple curves, manage inventory flows, and decide when to cancel or repost auctions. The strategic complexity of these decisions may create an implicit entry cost for makers with limited resources or expertise.

D.3 Liquidity Attention Rather Than Liquidity Fragmentation

MAMs avoid liquidity fragmentation in the traditional sense, since all curves are globally visible and any swapper can target any curve. However, the model introduces what may be described as liquidity attention. Makers must ensure that their curves remain visible, competitive, and correctly parameterized. The distribution of attention across many curves may shape observed liquidity in ways that are not yet fully understood.

D.4 Gas Costs and On-chain Overheads

Curve creation, cancellation, and settlement incur on-chain costs. The frequency with which makers repost auctions and the granularity of curve decomposition directly influence gas usage. Although MAMs avoid the storage costs associated with large orderbooks, the long term cost profile of curve management requires further study, especially in environments with high demand for block space.

D.5 Extreme Market Movements

Because curves follow deterministic schedules, large and sudden movements in external market prices may render many curves temporarily unattractive. Makers retain the ability to cancel and repost auctions, but there is no adaptive pricing mechanism within the curve itself. The behavior of MAMs during periods of extreme volatility is therefore an open question, particularly regarding market resiliency and execution availability.

D.6 Fairness and Auction Timing

The determinism of price evolution eliminates state reactive MEV but does not eliminate competition for transaction ordering. In environments with heterogeneous access to block producers, certain actors may gain systematic advantages in targeting competitively priced curves. Quantifying the impact of these timing advantages and developing mitigations remain important questions [12, 13].

D.7 Cross Chain Extensions

Cross chain settlement variants require reliable proofs of execution between chains. The security of such designs depends on assumptions about finality, message passing, and proof verification. A complete analysis of cross chain MAMs will require integrating results from cross chain communication research, which remains an active area of study [20].

D.8 Non Fungible Asset Extensions

Extending MAMs to non fungible assets introduces discrete allocation challenges. If inventory cannot be subdivided, the mechanism must specify how items are selected for settlement. The effect of indivisibility on user constraints, curve shapes, and fairness is not yet fully understood. This area presents opportunities for new auction theoretic analysis [10, 11].

D.9 Strategic Maker Behavior

The ability to cancel and repost auctions enables a wide range of strategies. Makers may attempt to influence perceptions of available liquidity, structure time dependent supply schedules, or create layered auction stacks. A formal study of strategic behavior in MAMs, including equilibrium analysis and incentives for truthful curve construction, remains an open problem [17, 12].

D.10 Open Research Questions

Several questions remain open:

- How do MAMs behave in equilibrium when many makers pursue adaptive strategies.
- What indexing architectures best support decentralized and censorship resistant curve discovery.
- How should makers optimally allocate inventory across different curve shapes and time horizons.
- How robust is a MAM to extreme volatility, congestion, or coordinated adversarial pressure.
- Can MAMs be combined with privacy mechanisms without reducing verifiability or transparency [19].

D.11 Summary

Although MAMs address several vulnerabilities in existing decentralized exchange protocols, they introduce new practical and theoretical challenges. The limitations and open questions identified in this appendix suggest a broad research agenda that spans mechanism design, cryptographic infrastructure, market microstructure, and the study of adversarial environments. Continued investigation is necessary to fully understand the strengths and boundaries of the MAM model.

References

[1] Hayden Adams, Noah Zinsmeister, Mark Toda, Emily Williams, Xin Wan, Matteo Leibowitz, Will Pote, Allen Lin, Eric Zhong, Zhiyuan Yang, Riley Campbell, Alex Karys, and Dan Robinson. *UniswapX*. Uniswap Labs, July 2023.

[2] Hayden Adams, Noah Zinsmeister, and Dan Robinson. *Uniswap v2 Core*. Uniswap Labs, 2020.

[3] Hayden Adams et al. *Uniswap v3 Core*. Uniswap Labs, 2021.

[4] Fernando Martinelli and Nikolai Mushegian. *Balancer: A Non-Custodial Automated Portfolio Manager and Trading Platform*. Balancer Labs, 2019.

[5] Michael Egorov. *StableSwap: Efficient Mechanism for Stablecoin Exchange.* Curve Finance, 2019.

[6] Project Serum. *Serum: A Decentralized Exchange on Solana.* Project Serum, 2020.

[7] Will Warren and Amir Bandeali. *0x: An Open Protocol for Decentralized Exchange on the Ethereum Blockchain.* 0x Project, 2017.

[8] CoW Protocol Team. *CoW Protocol Whitepaper.* Gnosis, 2022.

[9] 1inch Network. *Fusion Mode: A Fully Off-chain RFQ-based Settlement Model.* 1inch Research, 2022.

[10] Paul Klemperer. *Auction Theory: A Guide to the Literature.* Journal of Economic Surveys, 1999.

[11] Milgrom, Paul R., and Robert J. Weber. *A Theory of Auctions and Competitive Bidding.* Econometrica, 1982.

[12] Eric Budish, Peter Cramton, and John Shim. *The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response.* Quarterly Journal of Economics, 2015.

[13] Philip Daian et al. *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges.* arXiv:1904.05234, 2019.

[14] Flashbots Research. *MEV: A Taxonomy of Attacks and Defenses.* Flashbots, 2021.

[15] Anoma Research. *Intent-centric Architectures for Decentralized Exchange.* Anoma Foundation, 2022.

[16] dYdX Foundation. *The dYdX Layer 2 Protocol.* dYdX, 2021.

[17] Robert Almgren and Neil Chriss. *Optimal Execution of Portfolio Transactions.* Journal of Risk, 2000.

[18] Dan Robinson and Georgios Konstantopoulos. *Ethereum is a Dark Forest.* Paradigm, 2020.

[19] Eli Ben-Sasson et al. *Scalable Zero Knowledge for Use in Decentralized Auctions.* StarkWare Research, 2021.

[20] James Prestwich. *Interchain Communication and Trust Models.* Summa Technologies, 2019.