Written by Benjamin Funk

Revisiting the Exchange Landscape

As quickly as its discovery, MEV has cemented itself as an unshakeable part of public blockchains. With this, an acceptance of MEV as a part of the exchange process for crypto assets has followed and led to an explosion of new products and protocols focused on minimizing and mitigating its adverse effects. Whether these efforts are enshrined, aligned, offchain, or onchain, they've brought together teams of brilliant people and a massive capital infusion to improve the "MEV stack." However, before mindlessly climbing the trees of Ethereum's dark forest, I'd like to return to the forest floor and reexamine the ground we're building on.

To get a glimpse of the future of exchange for crypto assets, we need to revisit the critical design challenges we face today. This piece represents the first of two articles exploring the path for exchange design in crypto, touching on the intersection between market microstructure and distributed systems to explore the current opportunities, challenges, and paths forward in creating the crypto asset exchanges of the future.

Where Are We Today?

Execution quality

— how closely traders buy and sell an asset at a price reflective of the "true market price" — is generally a function of the liquidity offered by market makers. These market makers play a crucial role as exchange intermediaries and are compensated for taking on the risk of matching both sides over time. The exchange component responsible for interfacing with market makers and traders is an order-matching engine

— effectively a digital system that pairs buy and sell orders according to specific rules.

Architecturally, an order-matching system's effectiveness at matching users' trades is a function of the balance between the degrees of freedom and constraints they impose on the market makers submitting orders to the matching engine. Exchanges that design their order-matching systems in a way that enables market makers to update their quotes efficiently and reliably attract the most competing market makers and liquidity, leading to the highest execution quality for traders.

Despite having paved the way for permissionless trust-minimized exchange, the execution quality obtained through AMMs

has yet to catch up with offchain alternatives. Even before incorporating front-running risksuser fees in DeFi today are substantially higher than in traditional finance (tradfi). The deepest Uniswap pool averages around 0.05%, or 5 basis points (bps), before gas, while the average markouts of retail orders across traditional exchanges average around 0.007%, or 0.7 bps. That's almost 10x in terms of performance.

This poor quality of execution on AMMs stems from the fact that the environments they create don't effectively attract sufficient, high-quality market makers to facilitate the exchange between buyers and sellers in the first place.

Over the past year especially, order flow aggregators

ranging from exchanges like Uniswap to wallets like MetaMask have become more opinionated in designing order-matching systems that improve upon execution quality. Some of these aggregators are building out internal solutions to improve execution quality. At the same time, an array of traveling merchants are attempting to sell them software – from account abstraction SDKs to intent solutions to OFAs – that promise to solve their problems for them. The main force driving this trend is the inability of AMMs to provide users with high-quality trade execution.

A Brief Overview of the AMM Problem

These challenges of classic AMMs have already been discussed at length, so I'll keep this short. On AMMs, the market makers

(liquidity providers) who take on liquidity risk must openly declare their market-making strategies. These strategies, recorded on blockchains, specify how each trade influences the prices they offer for their assets. However, since blockchains update slowly, these market makers cannot adjust their prices quickly enough to avoid getting sniped by arbitrageurs. Consequently, market makers are disincentivized to participate in AMMs, and onchain order execution suffers.

One argument made in favor of AMMs is that passive market-making should be able to compete in terms of offering good prices to users compared to professional market-making because competition between professional tradfi market makers stems from latency wars to get to the top of the queue for execution priority. Consequently, the efforts made by professional market makers to win this competition don't necessarily lead to better execution quality for traders.

If the only market maker in town for ETH were an AMM that takes money from liquidity providers, it wouldn't be surprising to see more or less the same prices offered as if professional market makers were given that monopoly. Nevertheless, this correct observation around the realities of market making today can only be applied to good use in a vacuum, as the existence of faster, more expressive exchanges for professional market makers leads to passive liquidity providers on

AMMs committing to a losing strategy anyway (through the means described earlier in A Brief Overview of the AMM Problem)

.

Hooks & TEEs – The Limits to Futuristic AMMs

Innovative designs, such as LVR (loss-versus-rebalancing)

-reducing hooks through dynamic fees, are emerging to deal with adverse selection by helping market makers forecast future trades and reprice their inventory. Nevertheless, there remains a critical barrier to their adoption by market makers, which is that AMMs force market makers to publicly commit to a strategy

, leading to frontrunning issues.

Generally, frontrunning

occurs because adversaries know a market maker's order before it gets executed

. But when a market maker publicly commits to a market-making strategy, they tell the market the sequence of trades they will make given a set of inputs. This makes it possible for an adversary to front-run the market maker before they even place an order

. The more sophisticated the market-making strategy, the greater the surface area for attack vectors exists.

For the same reason that it would suck to play poker while telling everyone what you'll bid on, tradfi market makers like Citadel make their employees sign NDAs to keep their market-making strategies private. Keeping these strategies confidential is so essential to these firms that they don't let their former employees work for a competitor for a year or two after they leave. They may even pay their full-time salary and benefits for them to just sit at home throughout this time.

Privacy solutions like TEEs

(such as SGX) combined with tools such as Uniswap's hooks

could offer a way for AMMs to incorporate highly sophisticated strategies similar to those used by sophisticated high-frequency trading market makers while keeping these strategies hidden from the public. Despite these potential improvements, the challenge with this approach is that to stay competitive, sophisticated market makers would need to update their market-making algorithms constantly.

For example, a liquidity pool creator on UniswapV4 implementing their market-making strategy using SGX-based hooks would have to regularly change their algorithms running inside SGX to stay competitive as market conditions change. Moreover, the confidentiality of these algorithms doesn't guarantee protection against an adversary inferring and exploiting them, which would be another factor pushing liquidity managers to update their algorithms running inside SGX.

This creates a tradeoff: you can either have an exchange where liquidity providers can publicly verify the strategy before they put in money, or you can have customizability for adapting to changing market conditions. Consequently, the value proposition of AMMs as trust-minimized vehicles that define strict rules around managing assets would need to be reconsidered.

Does Your Order Matcher Know Too Much, Too Little, or Just Enough?

Because of these challenges, we're seeing a partial move away from the AMM model and the re-emergence of order books

and Request for Quote (RFQ)

systems in crypto in an attempt to invite the virtuous cycle of market makers, liquidity, and high-quality execution that comes with it.

User-facing order flow aggregators, from wallets to dapps to exchanges, vary in their roles, incentives, and responsibilities. They make money by 1) providing the front-end that captures trader attention and trust or 2) creating the exchange that facilitates trades. Focusing on the latter, we need to understand the tradeoffs and challenges in pairing different architectures for order-matching with varying systems of markets and assets. The long-term competitiveness of exchanges depends on how effectively Party A and B can be matched, primarily downstream of the constraints (or lack thereof) placed upon the market-making intermediaries across these venues. One of the primary ways exchanges differ in this regard is in the choice of an RFQ or an order book.

Information Asymmetry in RFQs and Order Books

Stepping away from the blockchain context and looking at these systems in a vacuum, evidence across academia and industry overwhelmingly favors order books over RFQ systems for superior order execution. Order books enable efficient price discovery and reduced spreads for users by creating a much more dynamic equilibrium of supply and demand.

We can see this by breaking down the stakeholders in matching a trade:

1. The Buyer (Party A)

2. The Seller (Party B)

3. The Market Maker – the intermediary facilitating the interaction

The Buyer (Party A)

The Seller (Party B)

The Market Maker – the intermediary facilitating the interaction

In an order book system, price intentions are declared publicly by all parties involved. Users post their orders directly, and market makers compete to execute them. If Party A wants to buy 1 ETH for up to $10,000, and Party B wishes to sell 1 ETH for no less than $11,000, the spread between these two prices is visibly $1,000. With this information in the open, participants can make decisions based on real-time order depth and liquidity. If market makers or other participants place quotes that don't align with current market conditions, their orders will remain unfilled until they adjust.

Conversely, in an RFQ system, Parties A and B request a quote based on the amount of the asset they're eyeing without being able to specify their price limits

. When the market maker receives this request, they are incentivized to widen their prices, anticipating that Parties A and B may tolerate some slippage.

In some situations, RFQs allow traders to better control the dissemination of information: what

to show, who

to show it to, and when

to do so to limit adverse market reactions during large block trades in illiquid markets. In these situations, RFQs can be more effective than dark pools as they enable traders to outsource order execution to a professional market maker who takes a cut in return for ensuring the trader doesn’t screw up their order execution.

Generally, market makers in RFQs are positioned to make more than in order books because they don’t need to commit liquidity until Parties A and B specify how much of an asset they’d like. Without the pricing pressure from a transparent order book, Parties A and B will likely incur higher costs in an RFQ system, benefiting the market makers at the traders’ expense.

In light of these differences, we should be cautious to accept the narrative that the difference between order books and RFQs is insignificant enough to accept RFQs as the path forward for crypto exchange. Many participants in the crypto industry allude to the existence of zero fees in Robinhood’s RFQ system and the current dominance of RFQs in the bond market as evidence for their legitimacy. But we shouldn’t forget that these markets are characterized by uncompetitive behavior antithetical to crypto’s purpose.

Looking at Robinhood, for example, it is

true that market makers like Citadel only get retail order flow if they are improving upon national best bid and offer (NBBO)

across the numerous order book venues where equities are traded. However, if those Robinhood users collectively sent their trade on the NASDAQ, the spreads they would pay would decrease because Citadel would have to compete with everyone else.

We should not rely on evidence from oligopolistic industries to justify the existence of the order-matching systems they utilize.

Opaque market structures, such as the bond market (controlled by JPM, Citi, and BofA), benefit people with more information. It should go without saying that when concentrated entities control a significant portion of a market, they have the information, leverage, and incentive to resist changes to that market’s structure that could affect their dominance.

Despite this, it’s clear that we’ve made strides towards improving how RFQs work as an industry. For example, RFQ systems in traditional markets are characterized by high-touch processes and inefficiency. A typical RFQ-based interaction between counterparties for derivatives on commodities will force Party A, Party B, and their market makers to set initial and variation margins. The financial contract is outlined through back and forths with legal contracts on legacy corporate ticket systems and manual, error-prone communications at expiry. This complex process, compounded by T+2 settlement periods, creates challenges across effective validation, reconciliation, and risk management, the negative results of which are all passed on to the end users. There’s a lot of room for improvement here that crypto can play a role in.

Within crypto RFQ development, we’ve seen some fast-paced improvements as well. In many crypto RFQ systems, market makers aren’t required to commit to liquidity to match against it beforehand and only have to improve against the AMM price from the previous block.

On the surface, it may seem that if the market maker decides not to improve that price, the worst possible price for the trader should be the same as if they had gone through an AMM directly. However, by looking at the example of a buy order, we can see that things aren’t so simple.

The market scenario in which an RFQ provider is likely to route the order to an AMM (instead of filling it themselves) is when the AMM price is lower than the offchain price. Why would an arbitrageur sell their assets to this RFQ trader at the price of the last block when they can sell it for more money on Binance? Consequently, the swapper gets routed to the AMM, where they must compete against specialized arbitrageurs to get to the top of a block. If they can get to the top of the block, the swapper can get the price they were originally quoted, but the swapper won’t win that battle.

In UniswapX’s RFQ implementation, the price at which user orders get filled is a function of competition between fillers which are not only able to, but forced to, compete based on the speed of on/offchain data ingestion, analysis, and order submission. Should a filler decide not to fill a trade they won after this offchain competition, the price they had previously committed to is used to parameterize an onchain Dutch auction. Getting back to the example of a buy order routed onchain because it’s unattractive to a filler (they can sell for more offchain), the swapper’s price will likely be better from a well-parameterized Dutch auction than from an RFQ where they have little chance of getting to the top of the block.

MafiaEV or MonarchEV? The Information Asymmetry Tradeoff in Onchain Order Books

So, if order books are better than RFQs, let’s put that onchain and call it a day! In both theory and practice, it’s not that simple.

An onchain order book is defined as a platform where:

1. Users post orders onchain
2. Order execution is prioritized according to the orders with the best prices and earliest submission times
3. Consensus or a leader selection algorithm is utilized for censorship resistance

Users post orders onchain

Order execution is prioritized according to the orders with the best prices and earliest submission times

Consensus or a leader selection algorithm is utilized for censorship resistance

There are some fantastic attempts at creating performant onchain order books designed to be competitive with their offchain counterparts. This is often achieved through operating in environments with cheap compute to reduce the costs of placing orders onchain and achieve faster block times, both of which reduce LVR. Even if these characteristics can be obtained such that onchain order books can compete with onchain AMMs, critical challenges still emerge from inherent blockchain constraints that challenge onchain order books vying for liquidity and volume from those available offchain.

Onchain order books don’t have a uniform architecture and will all look different depending on the chain they’re built on. But in all cases, the basic flow is similar — a retail user submits an order, the order goes through the consensus mechanism where the sequence of orders is decided, and then the order appears onchain.

"Goes through the consensus mechanism"

is where all of the games can be played that put onchain order books at a structural disadvantage to competing order flow aggregators using offchain order books.

Onchain order books can choose one of two systems to determine the state of an order book:

1. Multiple leaders provide input into the sequence of orders
2. A single leader decides the sequence of orders

Multiple leaders provide input into the sequence of orders

A single leader decides the sequence of orders

Either way, onchain order books will encounter one type of MEV – LVR

, resulting from multiple leaders, or tx reordering

, resulting from a single leader. These different types of MEV are well suited to@sxysun1’s framing in this talk, which classified MafiaEV and MonarchEV as two of three different types of MEV. MafiaEV

denotes the extractable value achieved through coordinated strategies among network participants exploiting information asymmetries. In contrast, MonarchEV

encapsulates the value extractable through centralized, authoritative control within blockchain protocols, particularly by entities with decisive power over transaction sequencing and state finalization, such as block builders.

MafiaEV: Designs with Multiple Leaders

In blockchain-based order book systems using multi-leader consensus, latency arises from three key technical aspects: conflict resolution, network delays, and transaction processing. Multiple leaders processing transactions simultaneously leads to conflicts, requiring time-consuming consensus rounds. Geographically dispersed nodes introduce significant network latency. Additionally, each node's independent validation and ledger state replications add processing time.

Regardless of the precise details of the consensus mechanism, onchain order books that multiple leaders update must deal with this MafiaEV stemming from adversaries taking advantage of market makers' inability to update how they distribute liquidity across the order book quickly. While the absolute latency a market maker experiences when interacting with an order book is important, it’s critical to emphasize that an exchange’s survival depends more on its latency relative to other exchanges

.

Suppose the fastest onchain order-matching engine Y takes 10 seconds to trade, but offchain order book X takes half a second. In that case, price discovery will happen offchain, and all arbitrage will be from the offchain order book to the other exchanges. Suppose the onchain order book Y lowers latency to half a second, but Coinbase takes 10 milliseconds. In that case, onchain order book Y’s prices will be stale, as will its liquidity and user uptake.

Block times, costs, and the latency to submit and cancel quotes can certainly be reduced, and consensus and network layers can be innovated and push the boundaries such that the relative latency between onchain order books gets close to their offchain counterparts.

Still, we must also consider latency guarantees across time and order type. On any order book, if order cancellations are slower than order submissions (or vice versa), market makers lack guarantees around how they will be able to handle various market conditions. While the latencies they bear in submitting orders might be suitable, market makers can’t rely on that information to infer how quickly they could cancel those quotes in the future if they become stale. In an onchain order book, consensus mechanisms’ unpredictable latency magnifies this problem.

On top of this, participants depend on the block builder to not order trade requests in such a way that is highly beneficial to them at some point. In fact, should an onchain order book attract significant volume, block producers specialize in capturing the generated MEV. This could have a centralizing force on the underlying blockchain, potentially harming its value proposition as a credibly neutral settlement layer.

MonarchEV: Designs with a Single Leader

Onchain exchanges will likely remove consensus from as many parts of the order-matching process to combat this latency issue. One of the simplest solutions to this issue is to grant a single leader the ability to decide the order sequence.

These single-leader venues experienceMonarchEV. In this context, MonarchEV originates from temporary monopolies granted to single market makers in permissionless environments, allowing them to reorder transactions.

Teams like dYdX aim to counter this by requiring market makers to put up collateral before granting them these monopolies, keeping them in check. However, this requirement increases market makers’ necessary capital costs and, more importantly, increases the venue’s risks of mispricing the collateral required to keep block producers in check. This ultimately creates scalability issues for the venue as asset variety, volume, and volatility grow.

How an exchange sets the slashing costs

also becomes a significant challenge. Slash too little, and manipulation becomes profitable, even considering the slashed stake. Slash too much, and more capital is put at risk, making benign failures (i.e., misconfiguration) more costly. If an exchange wants to figure out the "right" amount to slash, they would have to do something akin to an auction, and then they’re back to latency problems.

Implementing SGX or threshold encryption

can also constrain the power the monopolist(s) holds over transaction ordering. Still, these implementations can only guarantee that, for a given set of transactions, the leader commits to not reordering them, inserting their own, etc. However, they cannot guarantee that every transaction is included fairly, so it still doesn't mitigate the problem – it just invalidates a subset of attacks.

Rollup-ing the Exchange

One way to work around the difficulties that front-running poses is to design a system that holds a check on the single operator of the order-matching system. This could be achieved by forcing the operator to commit to rules around issuing order receipts to users upon trade submission and posting trade history to a data availability (DA)

layer

.

An exciting approach worth highlighting here is to turn the exchange into a rollup like LayerN has done. By rollup-ing the exchange, the order-matching system could execute offchain while keeping its operator in check through a verifiable proof

on a DA layer. At a high level, this system could guarantee market participants that if the sequencer orders trade in such a way that violates the rules of the matching engine, traders could submit a fraud-proof and rely on a DA layer filled with trade history to do so. This also means that the exchange's throughput would be limited to the performance of the underlying DA layer.

Combined with a leader selection algorithm (automated or governance-based) that can replace a censoring sequencer, this exchange model could maintain the censorship resistance required for permissionless market creation while freeing itself from the constraints of a consensus-based orderbook. Moreover, exchange rollups could improve the security model for mitigating censorship and front-running from an honest majority assumption to an honest minority assumption through fraud or validity proofs.

However, minor latency manipulations by the exchange operator would be undetectable by fraud proofs. Consequently, affected market makers wouldn't be able to discern if the latency issues they are experiencing result from uniformly distributed network issues or targeted actions on the part of a misaligned sequencer. While all participants experience some variance in latency, a consistent disadvantage of a few milliseconds can critically impact a market maker's survival. For this reason, exchanges underpinned by these single-sequencer order-matching systems might struggle to gain adoption from market makers who expect regulations and reputation to give them guarantees around these risks.

It's important to note that, unfortunately, SGX doesn't solve this. Yes, if the bits of information containing orders could be sent directly to the order-matching engine running inside SGX, market participants could get guarantees that latencies were applied impartially. However, these packets of information that contain trade orders don't go directly from users to the enclave. They rely on some untrusted computer, like a router, to deal with communication between them. For this reason, it's always possible for the sequencer to manipulate the timing around when orders are seen by the order-matching engine running in an SGX enclave.

What About Auctions?

One of the exciting solutions to the MafiaEV vs. MonarchEV tradeoff that occurs in onchain order books is to combine a batch auction

with sufficiently low latency, solving for MafiaEV, which uses encryption, solving MonarchEV.

In contrast to continuous trading systems like orderbooks, where transactions are processed sequentially and immediately as they occur, batch auctions operate by accumulating a series of buy and sell orders within a predetermined time frame. At the conclusion of this interval, the collected orders are executed simultaneously at the same clearing price.

Noteworthy developments have

been made in enhancing the efficiency of batch auctions through privacy. For instance, In Penumbra's sealed-bid batch auction implementation, orders are first encrypted, and block builders commit to including these encrypted orders within a block. Only then are these orders decrypted and executed through a batch auction.

However, batch auctions struggle with real-time price discovery, largely due to the time needed to integrate new market information. This delay, inherent in their interval-based execution, contrasts with the continuous, immediate processing of onchain order books, which better suit high-frequency traders' need for quick liquidity injection.

When market consensus on an asset's value changes rapidly, batch auctions can't keep up, leading to a mismatch between the real-time market valuation and the batch auction price before the next interval begins. High-frequency traders (HFTs)

who, in part, capitalize on short-term price differences, find this delay unappealing. As a result, they shy away from these platforms, which potentially leads to reduced liquidity and slows integration of new price information into the market. While the positive outcome is that latency arbitrage becomes less profitable, traders looking for more immediate prices also shy away from placing orders in these batch auctions.

This phenomenon is backed up by empirical research on transitioning from batch to continuous trading conducted on the Taiwan Stock Exchange. The study found that continuous trading significantly enhanced price efficiency for medium and small cap stocks, indicating the importance of the market's ability to quickly integrate new information. Notably, this increase in trading activity was not

attributable to latency arbitrage, suggesting that the improvements in price efficiency resulted from the incorporation of continuous trading.

Despite this, the questions around batch auctions' relative merits and drawbacks seem far from reaching a close, at least in academia. While batch auctions might only take over tradfi markets if they receive support in the form of regulatory pressures against HFTs, they could become an integral part of the solution for onchain exchange due to their attractive properties in eliminating sandwich attacks and reducing gas costs.

Wen Part II?

This article aimed to lay out the challenges and opportunities around crypto exchange and MEV that we are currently presented with, including the shortcomings of AMMs, the arrival of order books and RFQs in crypto, and the design space for their implementation off and onchain. When looking at onchain order books, these tradeoffs can be framed through the lens of MafiaEV and MonarchEV. At a higher level, it seems that any attempts to make systems for onchain exchange more sophisticated lead us to a battle between efficiency and integrity.

In Part II, we further explore the opportunities, challenges, and implications of rapidly emerging primitives across cryptography and systems design, from intents to OFAs and net-new financial products. From this point, we'll hopefully be able to paint a clearer picture of how the future pipelines of onchain value might shape up.

We're excited to see teams tackle these difficult design challenges. If you're working at the cutting edge of these open questions, please reach out!

Special thanks to @soumyab8, @Autoparallel, @0xjepsen, @tylerinternet, @katiewav, @mountainwaterpi, @willkantaros, @AshAEgan, and @DannySursock for their feedback and insights on Part I, and the many more who also helped with Part II (coming soon).

I also want to shout out the@thebellcurvepod and @MikeIppolito_, @danrobinson, and @hasufl. Grateful to be getting a front seat in learning from thoughtful conversations between all the fantastic hosts and guests on the podcast throughout the seasons.

Disclaimer:

This post is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. You should consult your own advisers as to legal, business, tax, and other related matters concerning any investment or legal matters. Certain information contained in here has been obtained from third-party sources, including from portfolio companies of funds managed by Archetype. This post reflects the current opinions of the authors and is not made on behalf of Archetype or its affiliates and does not necessarily reflect the opinions of Archetype, its affiliates or individuals associated with Archetype. The opinions reflected herein are subject to change without being updated.