Market Microstructure in Digital Finance

When you trade, who sets the price—and who profits?

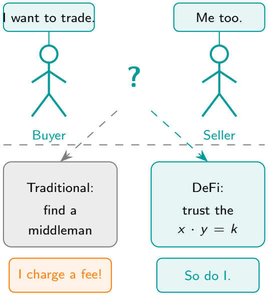
Economics of Digital Finance

BSc Course

What happens when there is no middleman to set the price?

In traditional markets, a buyer walks up to a trading floor and a specialist (a designated person whose job is to match buyers with sellers) sets the price. But what if there is no specialist, no floor, no human at all—just a mathematical formula running on a network of computers?

- Traditional markets: human intermediaries match orders (a buy request paired with a sell request)
- A new approach: replace all intermediaries with a single equation
- The equation always gives you a price—but is it a fair one?



Key Insight

Both systems solve the same problem—connecting buyers with sellers—but they make fundamentally different trade-offs between human judgment and mathematical certainty.

Market microstructure (the study of how trading mechanisms work at a detailed level) explains why these design choices matter for every trader.

Have you ever been the last one to know?

Imagine you are at an auction. You bid on an item, feeling confident about your price. But someone in the room already knows what the item will sell for tomorrow—and they quietly outbid you by just enough to win. You only find out later that the price was never really fair to begin with.

Think about a time you made a purchase and later realized someone else had better information. How did that affect the price you paid?

This feeling—of trading against someone with an advantage—is the core problem of market microstructure. When some participants know more than others, prices become skewed. The technical term is information asymmetry (when one side of a transaction knows more than the other). Understanding who knows what, and when, is the key to understanding why some trading systems are fairer than others.

Reflection

Every market mechanism is ultimately a response to the same question: how do you set a fair price when participants have unequal information?

Information asymmetry shapes every trading mechanism ever designed, from ancient bazaars to modern algorithms.

How do the two trading architectures actually differ?

Two dominant architectures exist for matching trades. An order book (a list of all pending buy and sell orders, ranked by price) relies on participants actively posting offers. An automated market maker, or AMM (a smart contract—self-executing code on a blockchain—that uses a formula to set prices), replaces human market makers entirely.

Dimension	Order Book	AMM
Price setter	Traders posting orders	Mathematical formula
Liquidity source	Professional market makers	Anyone who deposits tokens
Execution	Match buy with sell	Trade against a pool (shared token reserve)
Transparency	Partial (some orders hidden)	Full (formula is public code)
Capital need	Active management	Passive deposit
Vulnerability	Spoofing (fake orders)	Value extraction by bots

Key difference: an order book is a *negotiation*—prices emerge from competing offers. An AMM is a *computation*—the price is whatever the formula says, with no negotiation.

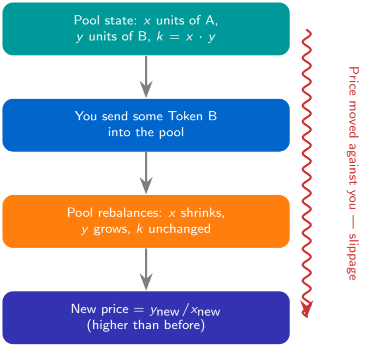
Each architecture has a different failure mode: order books fail when market makers withdraw (liquidity vanishes), while AMMs always have liquidity but the price may be stale (outdated relative to other markets).

Key Insight

Order books let the market negotiate prices; automated market makers let mathematics dictate them—neither approach is strictly superior.

The choice of trading architecture determines who bears risk, who earns fees, and how quickly prices reflect new information.

What happens inside the formula when you press swap?



The most common AMM formula is the constant product rule: $x \cdot y = k$, where x and y are the quantities of two tokens in a shared pool, and k is a constant that never changes. The price at any moment is simply y/x.

1. **Before the trade:** Pool holds x units of Token A and y units of Token B. Product: $x \cdot y = k$.
2. **You buy Token A:** You send some Token B into the pool. Token B reserve grows; Token A reserve shrinks—but k stays the same.
3. **New price:** Because there is now less Token A and more Token B, the ratio y/x rises. Token A is more expensive.
4. **Slippage** (the gap between expected and actual price): The bigger your trade relative to the pool, the more the price moves against you.

Key Insight

The constant product formula guarantees a price for any trade size, but larger trades pay progressively worse prices—the pool protects itself by making big trades expensive.

Slippage is not a bug—it is the mechanism by which the formula discourages trades that would drain the pool.

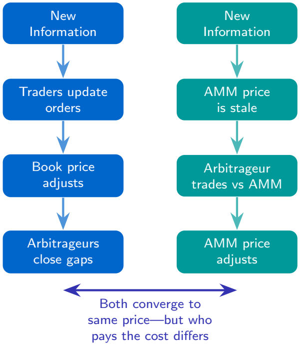
How does each system keep prices honest?

Order Book Architecture

- **Price discovery** (the process by which markets determine the correct price): Traders submit limit orders (offers to buy or sell at a specific price) that reveal their valuation
- **Correction mechanism:** If the price drifts, arbitrageurs (traders who profit by exploiting price differences between markets) buy cheap and sell expensive, pushing the price back
- **Speed advantage:** Professional firms with fast computers react to new information in milliseconds
- **Weakness:** If professionals withdraw, the book empties and prices become unreliable

AMM Architecture

- **Price discovery:** The formula sets a price, but it only updates when someone trades—it cannot react to external news on its own
- **Correction:** Arbitrageurs compare AMM price to other markets and trade until they match
- **Always-on liquidity:** The formula never refuses a trade, but the price may lag behind reality
- **Weakness:** The gap between AMM price and true price is profit for arbitrageurs—a cost to liquidity providers



Key Insight

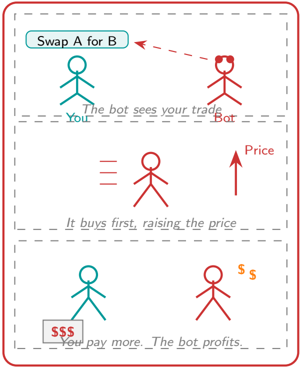
Order books react to information through human decisions; AMMs react through arbitrage trades—both reach the same price, but the cost falls on different participants.

Price discovery is never free—the question is whether the cost falls on professional market makers or passive liquidity providers.

On a blockchain, pending transactions sit in a public waiting area (the mempool—a queue of unconfirmed transactions visible to everyone). A bot (an automated program) can see your trade and insert its own trades around yours—before your transaction is confirmed.

- **Step 1—Front-run:** The bot buys the token you want, pushing the price up
- **Step 2—Your trade:** You buy at the inflated price, pushing it up further
- **Step 3—Back-run:** The bot sells at the higher price, pocketing the difference
- **Result:** You paid more than you should have; the bot extracted the difference as profit

This is a **sandwich attack** (a front-run and back-run wrapped around a victim's trade), a form of **MEV** (maximal extractable value—profit from reordering or inserting transactions).

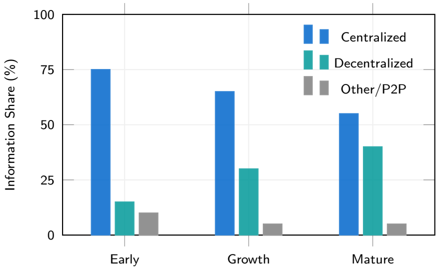


Key Insight

Transaction transparency—normally a virtue—becomes a vulnerability when others can act on your intentions before you can.

Maximal extractable value turns the public nature of blockchain transactions into a systematic cost for ordinary traders.

Where does the price actually come from?



When the same asset trades on multiple venues (different exchanges or pools), which venue sets the “real” price? Researchers measure this using information share (the proportion of price discovery—finding the true price—contributed by each venue).

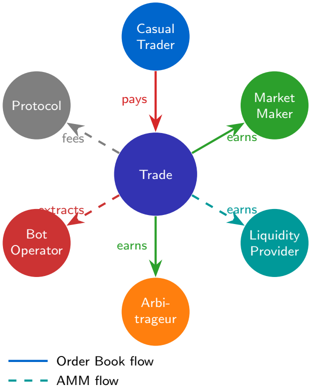
- Centralized venues tend to lead price discovery because professional traders with fast connections concentrate there
- Decentralized venues tend to follow, adjusting through arbitrage with a lag
- The gap between venues represents a cost: someone profits from closing it
- As decentralized infrastructure matures, the balance may shift

Key Insight

Price discovery is not a property of any single venue—it is distributed across all markets, with the balance shifting as infrastructure and participation evolve.

Information share measures where new price information first appears; it shifts over time as market structures mature.

Who profits and who pays in each architecture?



Every trading architecture creates winners and losers. The same person can be a winner in one system and a loser in another. Understanding who bears the costs is essential for evaluating whether a system is fair.

- **Casual traders:** Pay spreads in order books; pay slippage plus MEV in AMMs
- **Professional market makers:** Earn spreads in order books; have no role in AMMs
- **Liquidity providers** (depositors): Earn fees but suffer impermanent loss (a loss when the price ratio of deposited tokens changes) in AMMs
- **Arbitrageurs:** Profit from price discrepancies in both
- **Bot operators:** Extract MEV from AMMs; limited role in order books (front-running is regulated)
- **Protocol operators:** Earn listing fees or protocol fees

Key Insight

Automated market makers do not eliminate intermediaries—they replace professional market makers with passive depositors and shift extraction from regulated spreads to unregulated bot profits.

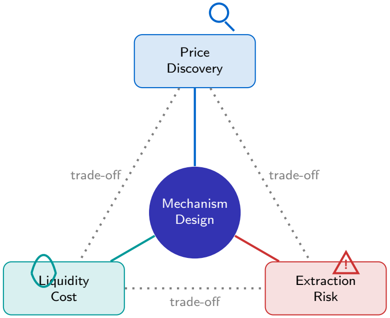
No trading architecture is neutral—each embeds a different answer to the question of who should bear the cost of providing liquidity.

Three questions to evaluate any trading mechanism

Whether you are a regulator, a developer, or a trader, you can evaluate ANY trading mechanism—past, present, or future—by asking three questions:

1. **Who sets the price, and how fast does it update?** (Measures price discovery quality: faster and more independent sources produce better prices)
2. **Where does liquidity come from, and what does it cost the provider?** (Measures sustainability: if providers lose money, liquidity will eventually dry up)
3. **Who can extract value, and is that extraction visible?** (Measures fairness: hidden extraction is more dangerous than visible fees)

Apply these three questions to any system. A good mechanism scores well on all three; most real systems trade off one dimension against another. There is no perfect architecture—only informed choices.



Key Insight

A trading mechanism that scores perfectly on price discovery, liquidity cost, and extraction risk has never existed—every design is a deliberate trade-off among the three.

Use these three questions to evaluate any trading system you encounter—from traditional stock exchanges to the newest decentralized protocols.

Your Challenge: Design a fairer market

You are designing a new trading mechanism for a digital asset. Your goal: minimize the cost imposed on casual traders while still attracting enough liquidity to function.

- **Constraint 1:** You must choose either an order book or an AMM as your base architecture (explain your choice using the three evaluation questions from the previous slide)
- **Constraint 2:** You must propose at least one mechanism to reduce value extraction by bots (describe how it works and what trade-off it introduces)
- **Constraint 3:** You must explain who provides liquidity in your design, what they earn, and what they risk
- **Deliverable:** A one-page design document with: (a) architecture choice and rationale, (b) extraction mitigation mechanism, (c) liquidity provider economics, (d) one weakness your design does NOT solve

There is no right answer. The goal is to recognize that every design choice helps someone and hurts someone else—and to make that trade-off explicit.

Challenge

The test of understanding is not whether you can describe how a market works, but whether you can design one that works differently—and explain what you sacrificed.

Market design is applied economics—the constraints are mathematical, but the choices are human.