DeFi Vaults & The Future of Onchain Trading

A Deep Dive into ERC-4626, ERC-7540, ERC-7575 and CLOB DEX

Agenda

Part 1: DeFi Vaults

- ERC-4626 Fundamentals
- Vault Extensions: ERC-7540 & ERC-7575

Part 2: CLOB DEX

- CLOB Fundamentals
- Real Implementations

What is a DeFi Vault?

Traditional finance

Funding

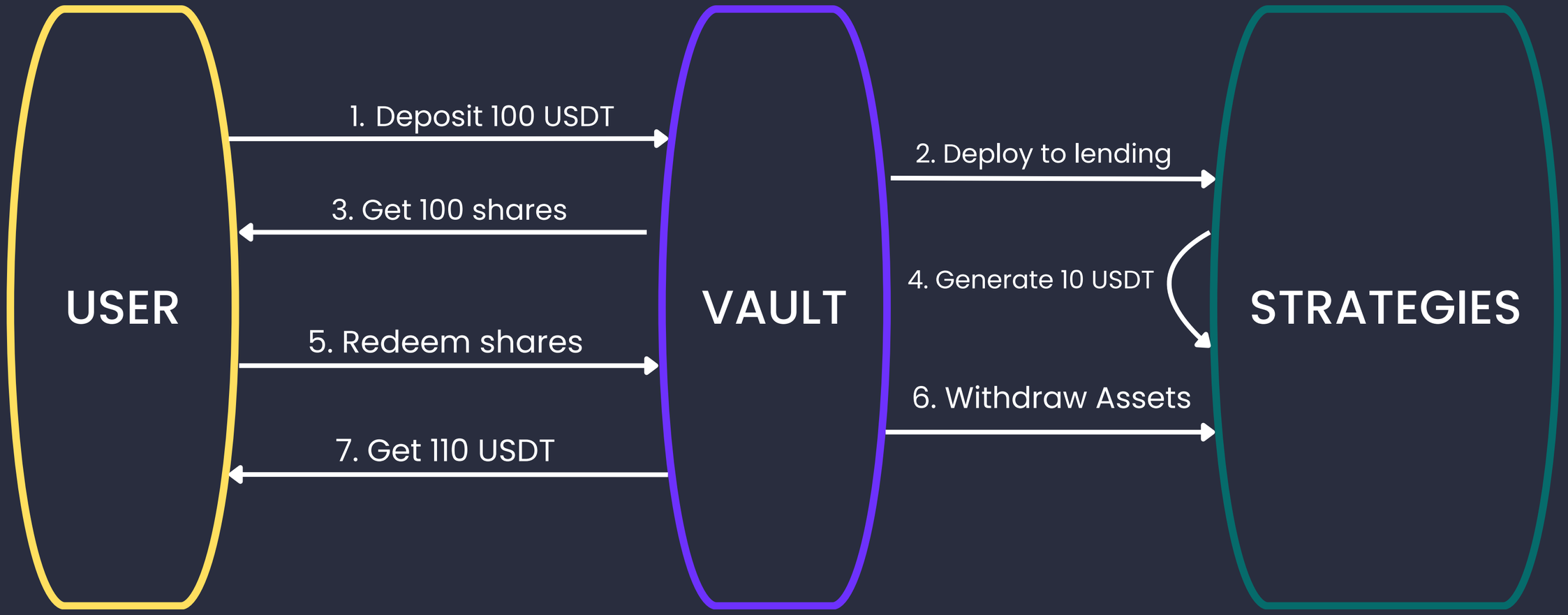
- Manual asset management
- Access: Limited

DeFi Vault

Smart contract
(Automated)

- Automated asset management
- Access: 24/7
- Transparent operations
- Share-based ownership

How Vaults Work: The Flow



Your deposit assets → Get shares → Vault earns yield → Redeem for more assets

PART 1: DeFi Vault Standards

ERC-4626 Fundamentals

The Problem Before ERC-4626

Every Protocol = Different Interface

- Yearn: deposit() → yTokens
- Aave: supply() → aTokens
- Compound: mint() → cTokens
- Curve: add_liquidity() → LP tokens

Result: Developers build custom adaptors for EACH vault

= Time + Cost + Errors

The Solution: ERC-4626 (Finalized 2022)

ONE-Standard Interface

- deposit(assets) → shares
- redeem(shares) → assets
- totalAssets()
- convertToShares()

Build one → Work with all vaults!

Simple ERC-4626 Vault Contract

solidity

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts/token/ERC20/extensions/ERC4626.sol";

contract MyVault is ERC4626 {
    constructor(
        IERC20 _asset,
        string memory _name,
        string memory _symbol
    ) ERC4626(_asset) ERC20(_name, _symbol) {}

    // Inherits everything:
    // - deposit()  ← Users deposit assets
    // - redeem()   ← Users get assets back
    // - mint()
    // - withdraw()
    // - All view functions
}
```

OpenZeppelin handles all the complexity!

Simple ERC-4626 Vault Contract

Category	Key Functions	Description
Info	`asset()`, `totalAssets()`, `convertToShares()`, `convertToAssets()`	View underlying asset, total vault balance, conversion rate
Deposit	`deposit(assets, receiver)`, `mint(shares, receiver)`	Deposit assets → mint shares
Withdraw	`withdraw(assets, receiver, owner)`, `redeem(shares, receiver, owner)`	Burn shares → withdraw assets
Preview	`previewDeposit()`, `previewMint()`, `previewWithdraw()`, `previewRedeem()`	Simulate conversions before executing
Limits	`maxDeposit()`, `maxWithdraw()` (optional)	Check user limits

ERC-4626: Share Calculation Formula

How shares are calculated:

$$\text{Shares} = \frac{\text{assets} \times \text{totalShares}}{\text{totalAssets}}$$

ERC-4626: Share Calculation Formula

DAY 1: Initial Deposit

Vault: 1,000 USDT, 1,000 shares
Rate 1 USDT = 1 share

Your deposit: 100 USDT
Calculation:

$$\frac{(100 \times 1{,}000)}{1{,}000} = 100 \text{ shares}$$

You received
100 shares

Vault has 1,100 USDT
total shares is 1,100

DAY 365: After earning 10%

Vault: 1,210 USDT, 1,100 shares
Rate 1.1 USDT = 1 share

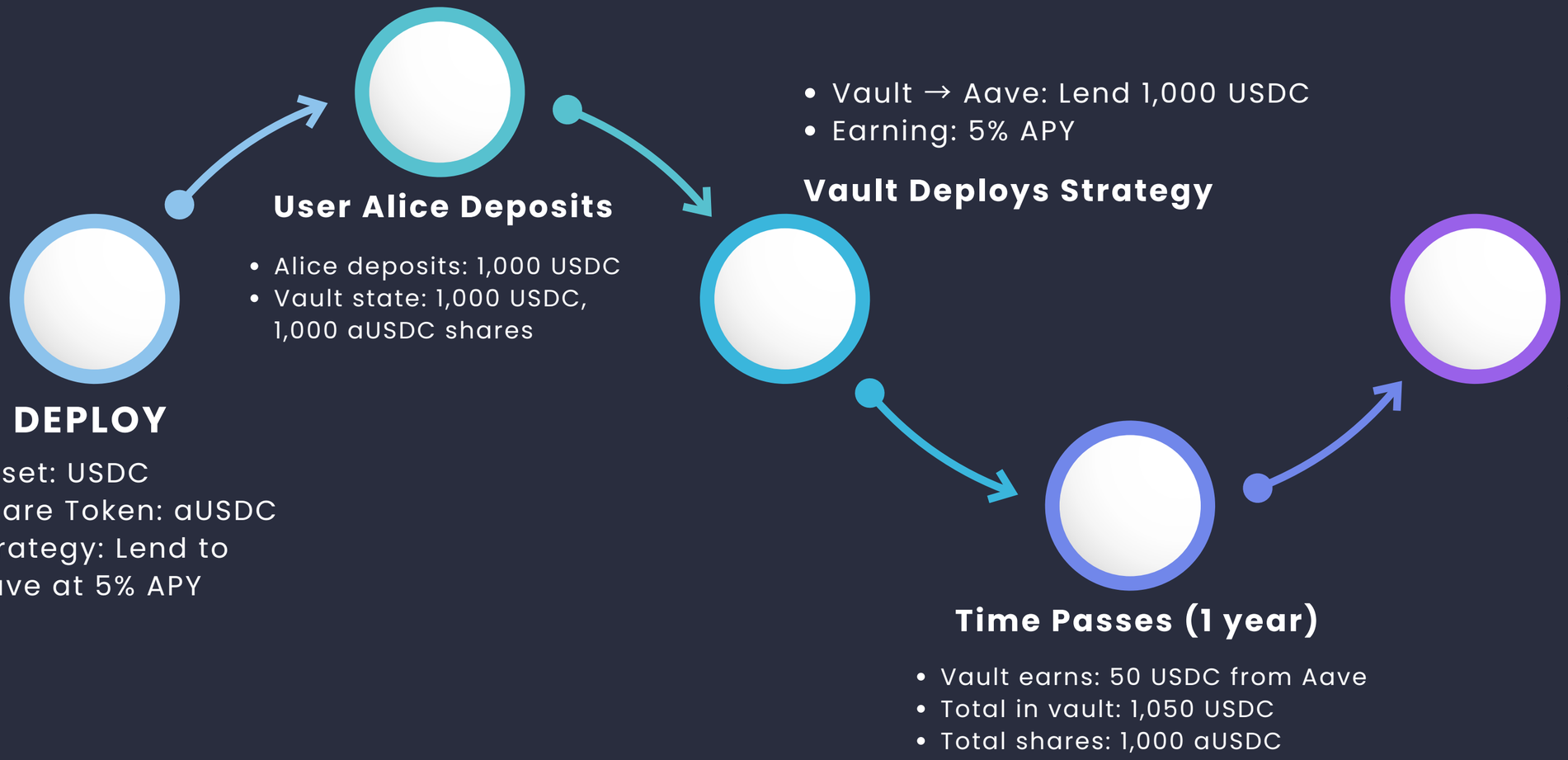
Your withdraw: 100 shares
Calculation:

$$\frac{(100 \times 1{,}210)}{1{,}100} = 110 \text{ USDT}$$

You received
110 USDT

your profit = 10 USDT (10% APY) 🎉

ERC-4626 in Action: Complete Example



DEPLOY

- Asset: USDC
- Share Token: aUSDC
- Strategy: Lend to Aave at 5% APY

User Alice Deposits

- Alice deposits: 1,000 USDC
- Vault state: 1,000 USDC, 1,000 aUSDC shares

Vault Deploys Strategy

- Vault → Aave: Lend 1,000 USDC
- Earning: 5% APY

Time Passes (1 year)

- Vault earns: 50 USDC from Aave
- Total in vault: 1,050 USDC
- Total shares: 1,000 aUSDC

Alice Redeems

- Alice redeems: 1,000 aUSDC
- Calculation

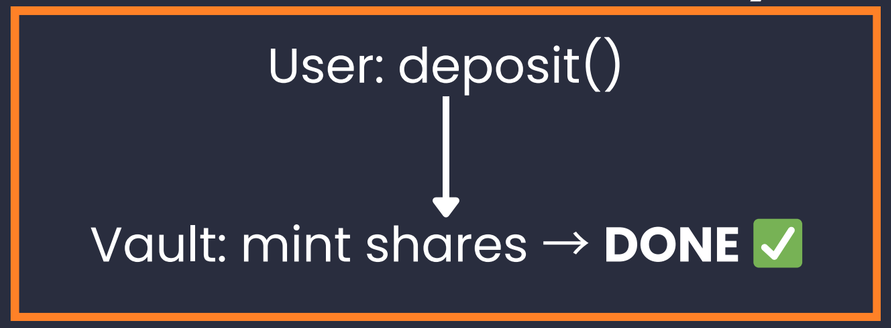
$$\frac{(1{,}000 \times 1{,}050)}{1{,}000} = 1{,}050$$

- Alice receives: 1,050 USDC
- Profit: 50 USDC (5% return)

Vault Extensions

ERC-4626 Limitation: Must be INSTANT

ERC-4626: Atomic only



Works great for

- Lending protocols (instant)
- Liquidity pools (instant)
- Simple Strategies (instant)

Doesn’t work for

- Real-world assets (days/weeks)
- Cross-chain bridges (minutes/hours)
- Liquid staking (7-21 day unbonding)
- Compliance workflows (KYC/AML delays)

The gap: Many important use causes need TIME!

ERC-7540 Solution

ERC-7540: Real-World Example

DAY 1 - User Request

Bob: requestDeposit(100,000 USDC)
→ RequestId: #15
→ 100k USDC locked in contract

Status: Pending

DAY 2-7 - Off-chain processing

- Legal verification
- Property acquisition
- Title transfer
- Compliance checks

Status: Pending

DAY 8 - Ready to claim

claimableDepositRequest(#15)
→ Returns: 100,000 shares

Status: Claimable

DAY 8 - Bob claims

Bob: claim(#15)
→ Receives 100,00 property vault shares
→ Can now trade, transfer, or hold

Status: Claimed

Use case: Tokenized Real Estate Vault

ERC-7540: The Request Lifecycle

Flow	Stage	Function	Token Movement	State
Deposit	1	`requestDeposit()`	User → Vault (assets)	Pending
	2	`processDeposit()`	Prepare shares	Claimable
	3	`claim()`	Vault → User (shares)	Complete
Withdrawal	1	`requestRedeem()`	User → Vault (shares)	Pending
	2	`processRedeem()`	Prepare assets	Claimable
	3	`claim()`	Vault → User (assets)	Complete

CONCLUSION

VAULT STANDARDS

ERC - 4626 : Base standard

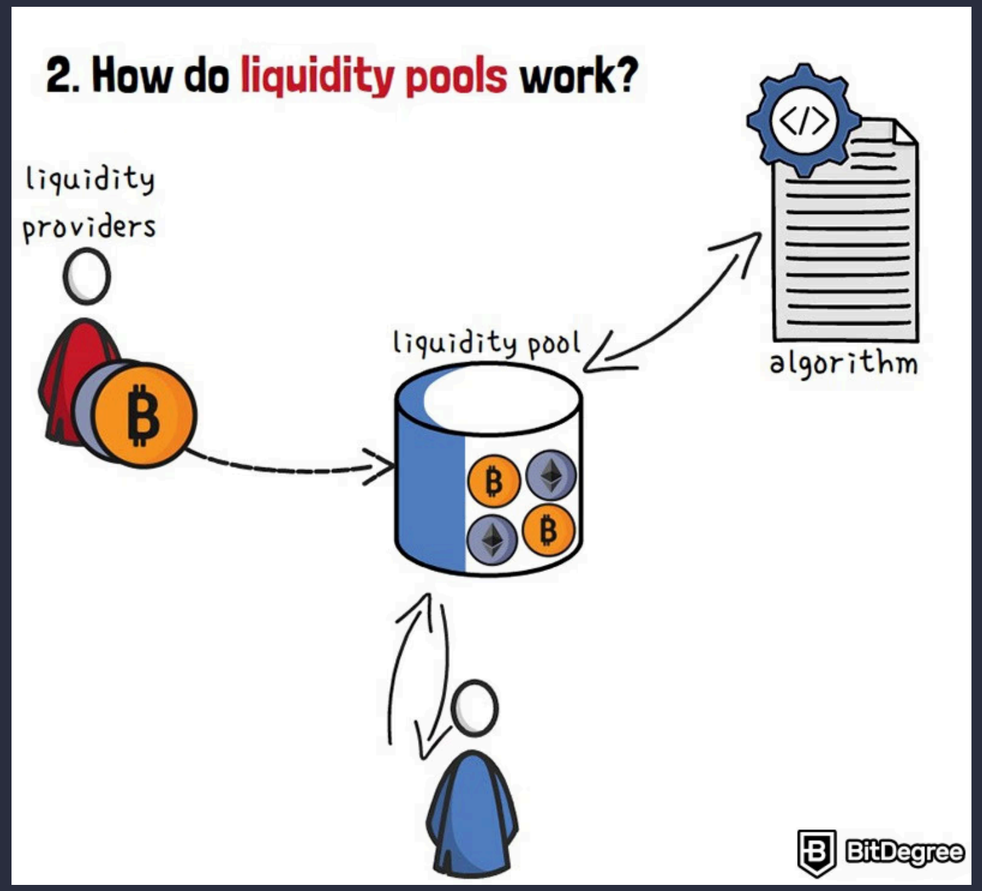
- Problem: Fragmented vault interfaces
- Solution: Unified API for all vaults
- Status: ✅ Widely adopted (Yearn, Balancer, Morpho)

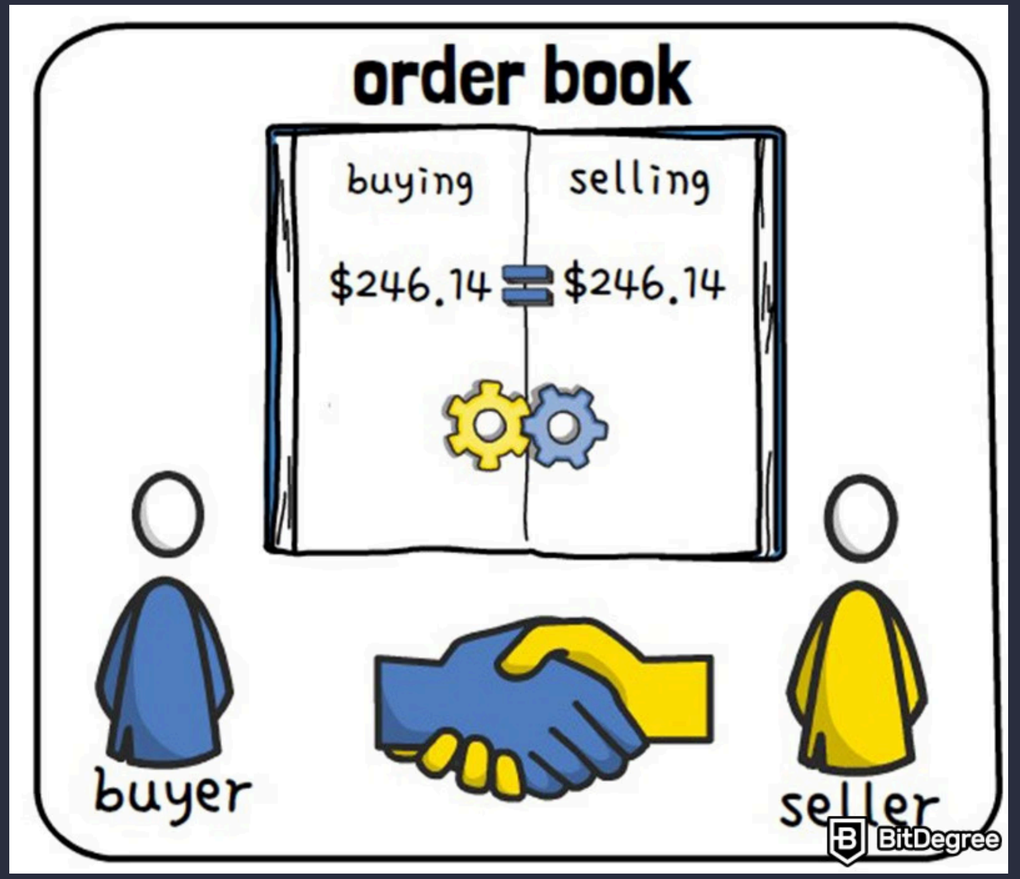
ERC - 7540 : Asynce extension

- Problem: can’t handle time delays
- Solution: Request → Pending → Claimable flow
- Status: ✅ Growing (RWAs, bridges, staking)

PART 2: CLOB DEX (Central Limit Order Book)

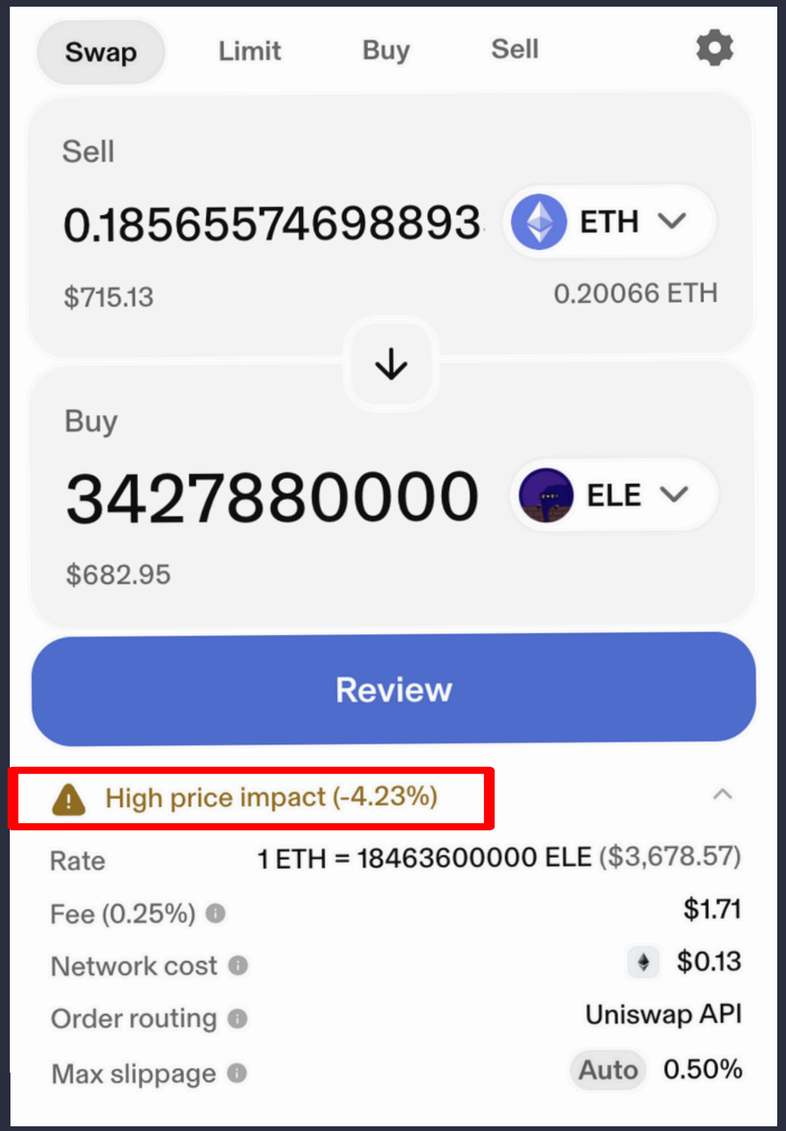
What is CLOB DEX?





https://www.bitdegree.org/crypto/learn/what-is-liquidity-pool-in-crypto

Problem of AMM



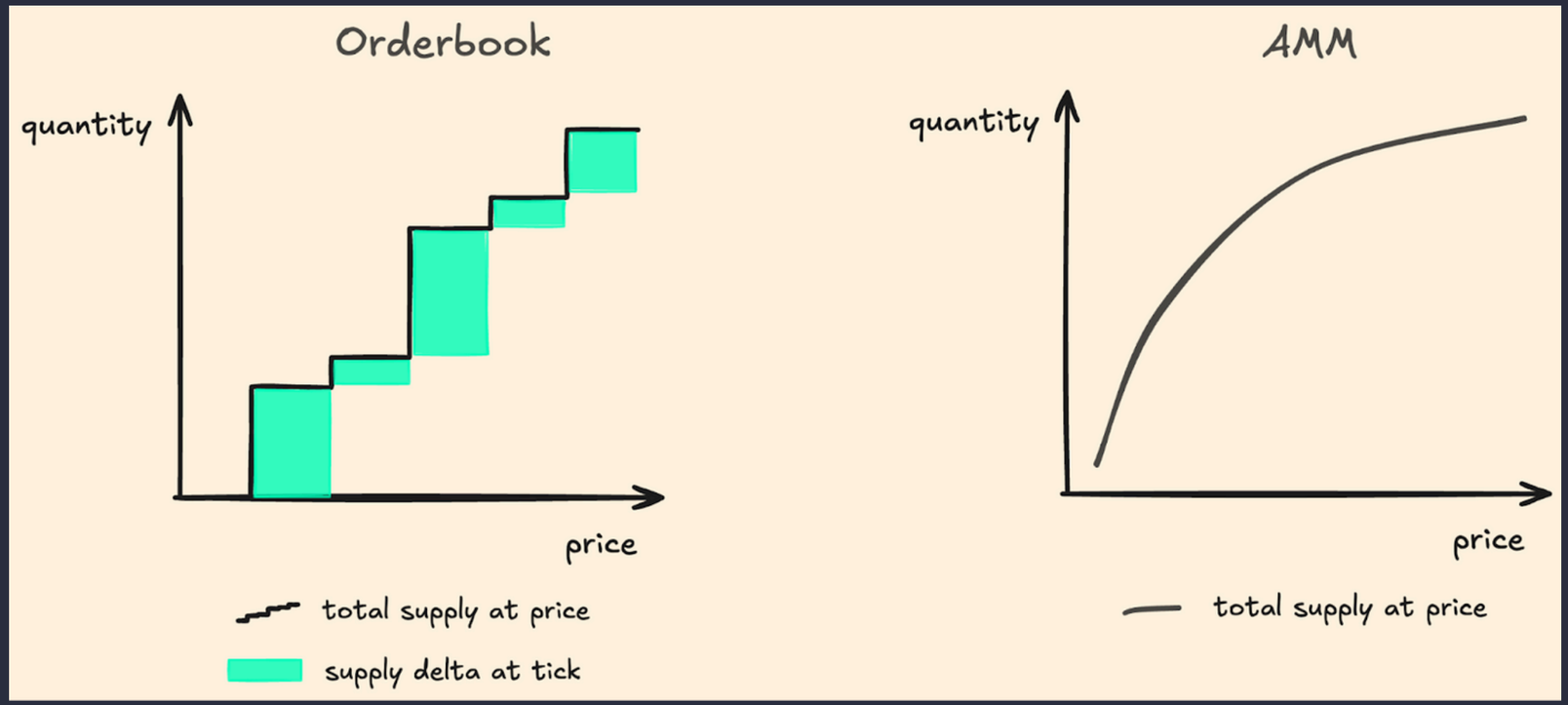
- High price impact
- No advanced trade feature

Problem: Slippage on Large Trades

Order Book | Trades

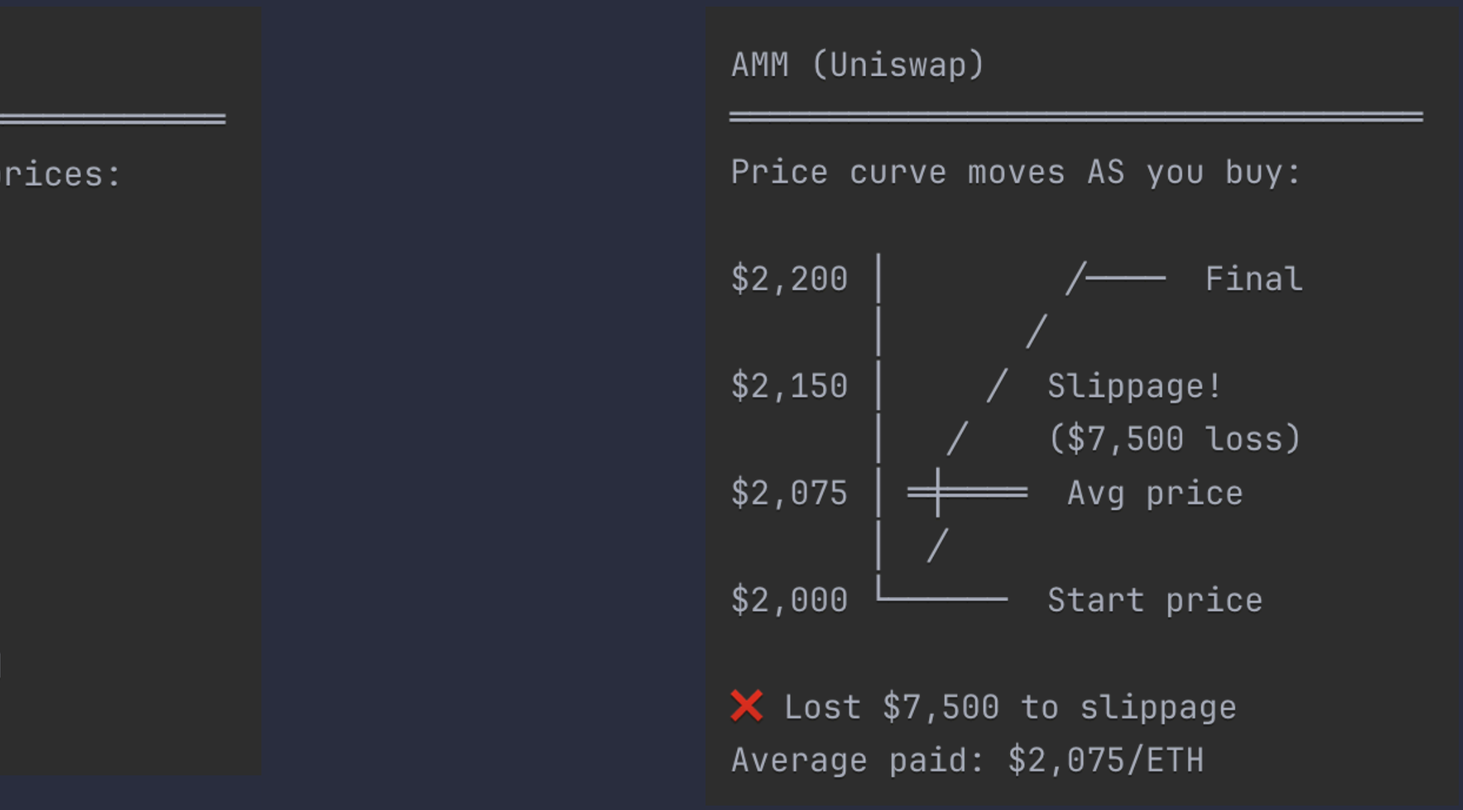
0.1 | ETH

Price	Size (ETH)	Total (ETH)
3,864.9	1.8331	575.9368
3,864.8	44.5606	574.1037
3,864.7	21.7373	529.5431
3,864.6	135.0704	507.8058
3,864.5	98.3747	372.7354
3,864.4	95.9352	274.3607
3,864.3	77.2835	178.4255
3,864.2	3.9081	101.1420
3,864.1	10.4344	97.2339
3,864.0	5.9364	86.7995
3,863.9	80.8631	80.8631
Spread	0.1	0.003%
3,863.8	111.2189	111.2189
3,863.7	21.3826	132.6015
3,863.6	6.1323	138.7338
3,863.5	226.5156	365.2494
3,863.4	18.1205	383.3699
3,863.3	119.7568	503.1267
3,863.2	82.5575	585.6842
3,863.1	89.5271	675.2113
3,863.0	65.4648	740.6761
3,862.9	19.6479	760.3240
3,862.8	148.7117	909.0357

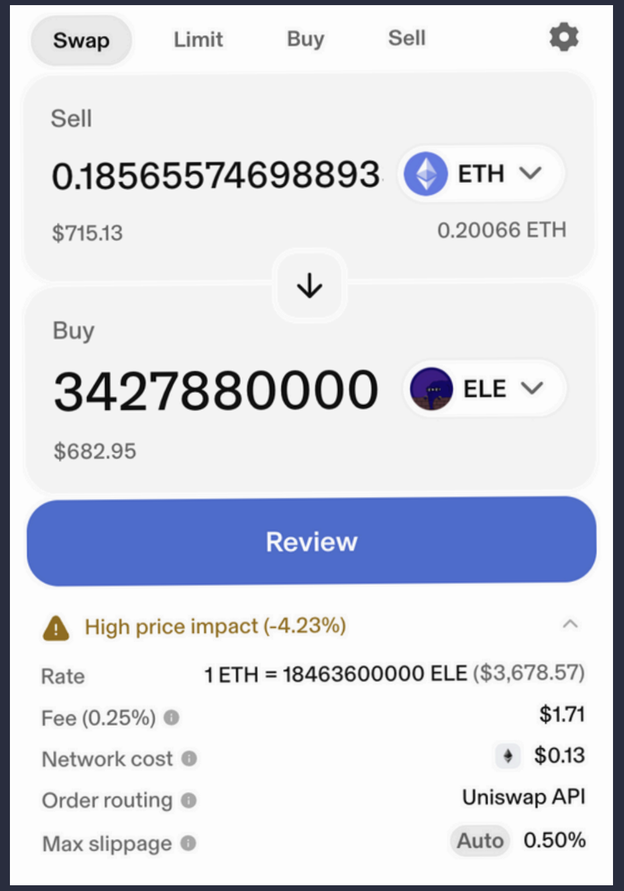


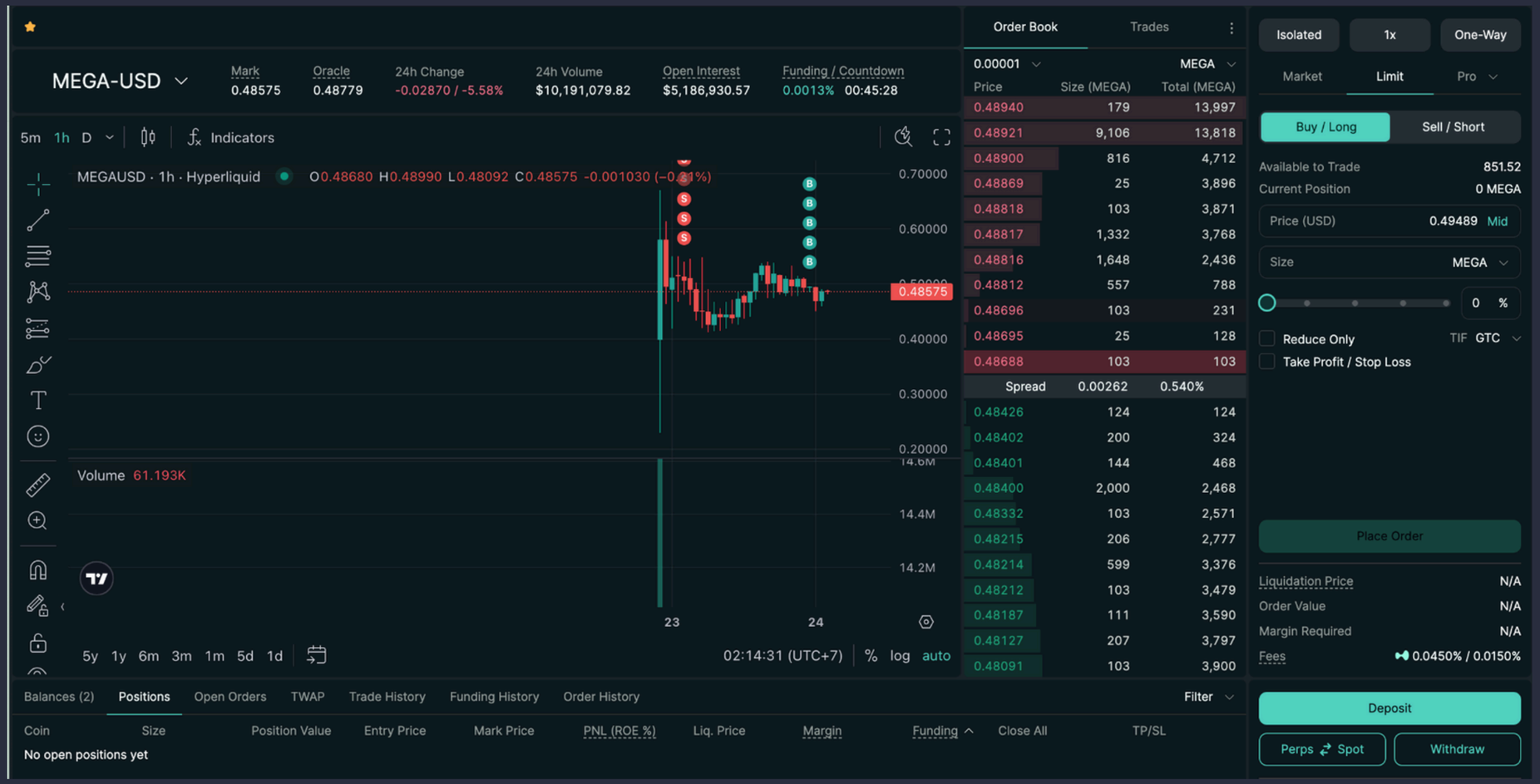
Problem: Slippage on Large Trades

```
CLOB (dYdX/Hyperliquid)
═══════════════════════

Match with FIXED order prices:

$2,010 ├── 20 ETH
$2,007 ├── 30 ETH
$2,005 ├── 25 ETH
$2,003 ├── 15 ETH
$2,000 ├── 10 ETH

✅ Minimal slippage
Average paid: $2,005/ETH
Saved $7,000 vs AMM! 🎉
```



Problem: Advanced Trading Features

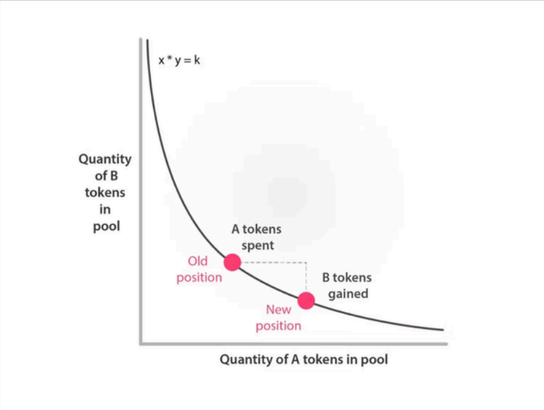




Feature Comparison

AMM DEX (Uniswap)

Liquidity pool



- ✅ Instant swaps
- ✅ Simple UX
- ✅ Always liquid
- ❌ High slippage
- ❌ No limit orders

CLOB Dex (Hyperliquid)

Order Book

0.00001		MEGA
Price	Size (MEGA)	Total (MEGA)
0.48900	816	8,759
0.48869	25	7,943
0.48818	3,289	7,918
0.48817	1,532	4,629
0.48816	1,848	3,097
0.48815	200	1,249
0.48722	50	1,049
0.48600	339	999
0.48592	103	660
0.48588	144	557
0.48587	413	413
Spread	0.00042	0.086%
0.48545	103	103
0.48543	205	308
0.48503	174	482
0.48453	103	585

- ✅ Limit orders
- ✅ Price control
- ✅ Professional tools
- ✅ No slippage (at limit)
- ❌ More complex

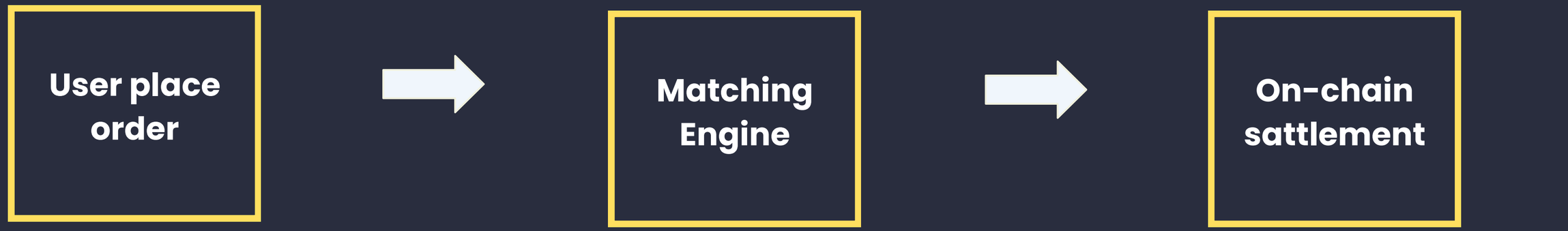
CLOB DEX: Brings order book model to blockchain

AMM vs CLOB: Core Differences

Technical Comparison:

Feature	AMM DEX	CLOB DEX
Order Type	Market only	Limit + Market
Liquidity	Pool formula	User orders
Price Discovery	x*y=k curve	Order matching
Slippage	Always	Only market orders
Professional Tools	Limited	Full suite
Gas Costs	Lower	Higher
UX	Simple	Advanced
Best For	Retail, simple swaps	Traders, leverage

How CLOB DEX Works



Buy 10 ETH

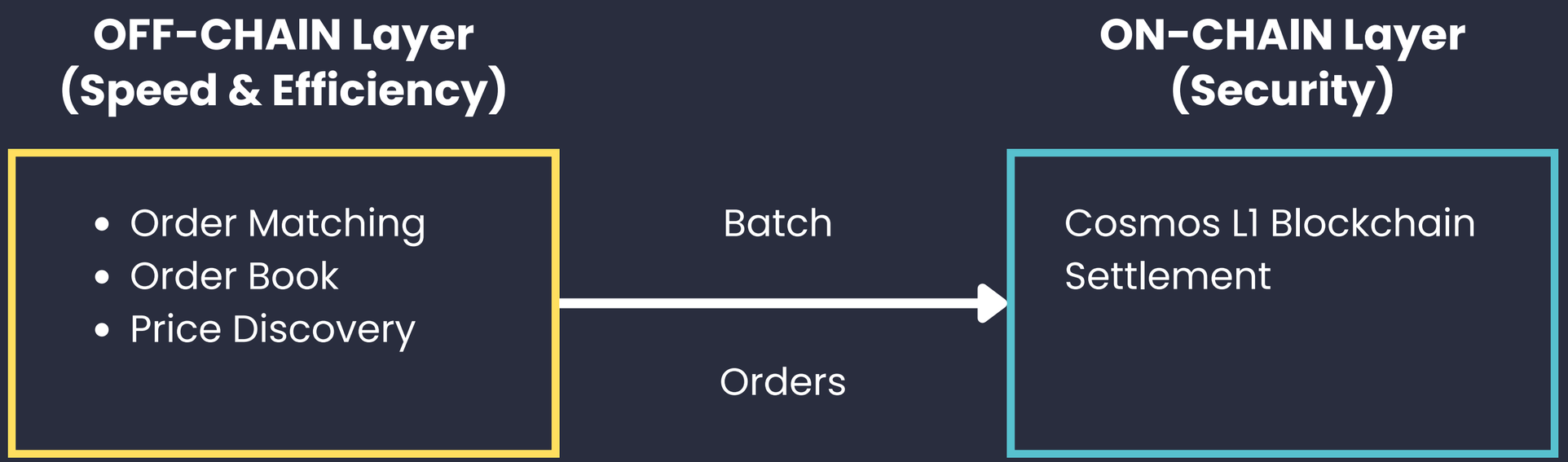
Order book update

- Buyer receives 10 ETH
- Seller receive payment

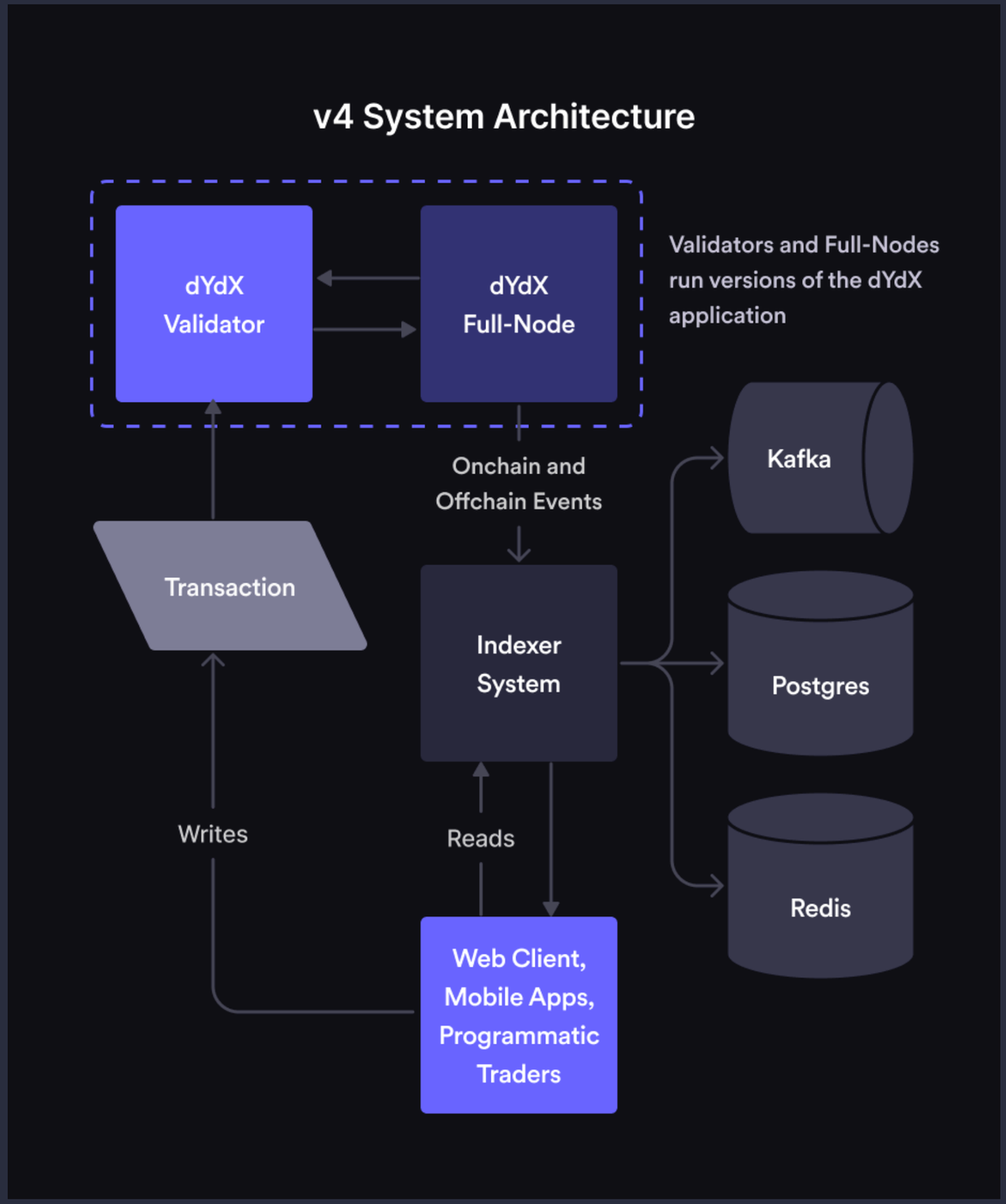
Architecture

dYdX v4

Architecture: Hybrid Approach



dYdX v4

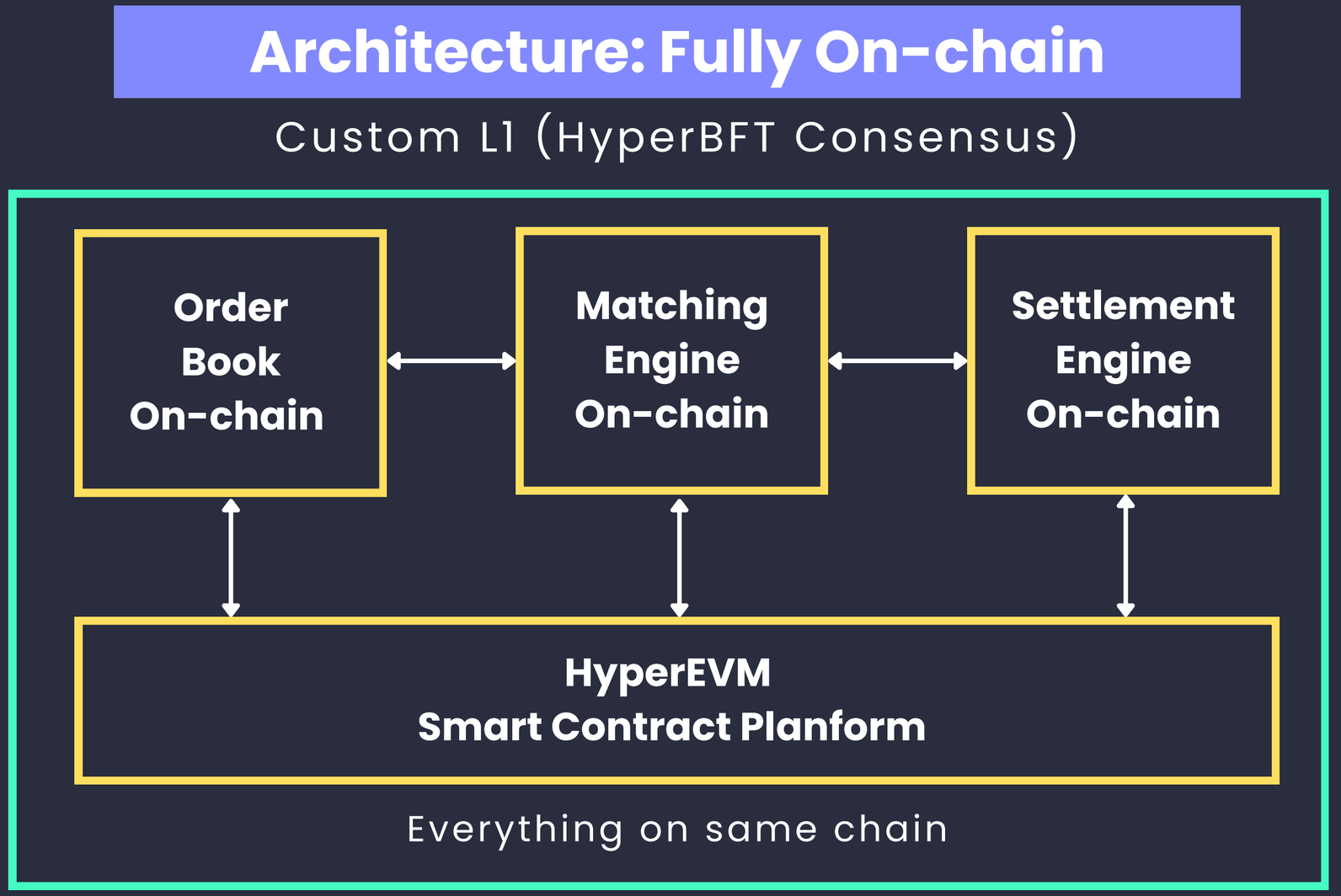

v4 System Architecture
dYdX Validator
dYdX Full-Node
Validators and Full-Nodes run versions of the dYdX application
Onchain and Offchain Events
Transaction
Indexer System
Kafka
Postgres
Redis
Writes
Reads
Web Client, Mobile Apps, Programmatic Traders

dYdX v4

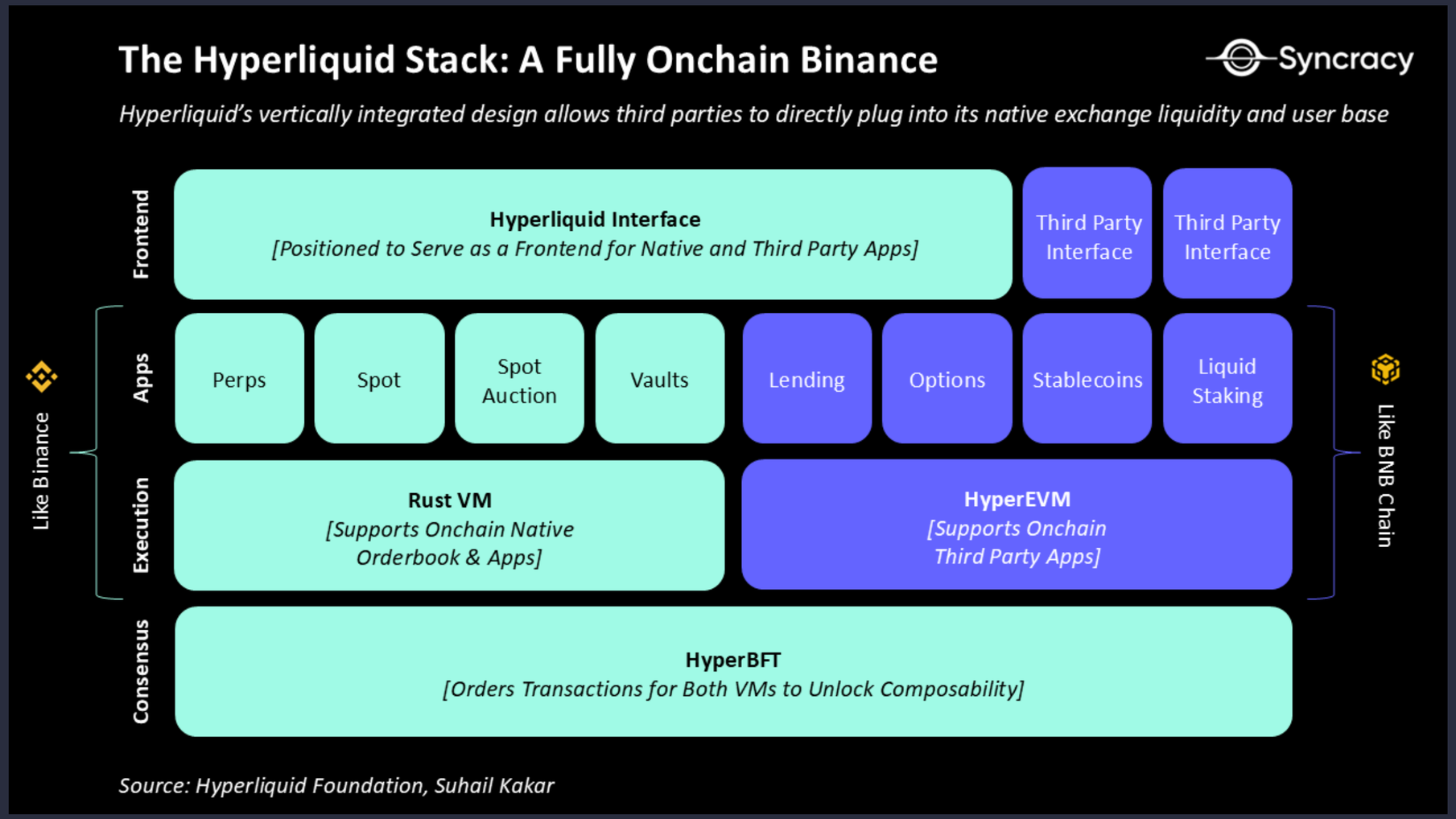
Version	Network	Core Tech	Trading Type	Order Book	Settlement	Custody
v1	Ethereum	Smart Contracts	Margin	Off-chain	On-chain (ETH)	Non-custodial
v2	L2 (StarkEx)	ZK-Rollup	Perpetuals	Off-chain	Proof → ETH	Non-custodial
v3	L2 (StarkEx v2)	ZK-Rollup	Perpetuals (Cross-margin)	Off-chain	Proof → ETH	Non-custodial
v4	Cosmos	Cosmos SDK + CometBFT	Perpetuals	Validator nodes	Native chain	Non-custodial

Hyperliquid

Architecture: Fully On-chain



Hyperliquid



https://pintu.co.id/en/academy/post/what-is-hyperliquid-hype#hype-coin-potential-as-an-investment

Hyperliquid

HyperCore is the fully on-chain trading engine of the Hyperliquid decentralized perpetuals exchange. It stores the complete state of every order across all asset pairs, handles order matching, cancellations, and all trading logic — fully on-chain, with zero off-chain order book or hidden matching.

Architecture Showdown: dYdX vs Hyperliquid

dYdX v4: Hybrid Model	Hyperliquid: Pure On-Chain
Off-chain matching → On-chain settlement	Everthing on custom L1 blockchain
PROS ✅ Proven scalable (1T+ volume) ✅ Very fast (10-50ms matching) ✅ First mover advantage	**PROS** ✅ Fully decentralized ✅ Complete transparency ✅ Higher throughput (100k ops/sec) ✅ Native smart contracts (HyperEVM)
CONS ❌ Some centralization (off-chain) ❌ Trust in matching engine ❌ Less transparent	**CONS** ❌ More complex infrastructure ❌ Newer (less battle-tested) ❌ Custom chain risk

Architecture Showdown

Imperator.co

	dYdX V4	Hyperliquid	ApeX	Vertex
Blockchain	dYdX Chain	Hyperliquid L1	StarkEx	Arbitrum
Off/On-Chain	Off + On-Chain	On-Chain	Off + On-Chain	Off + On-Chain
TPS	2,000	20,000	10,000	10,000 - 15,000
Block Latency	~1.1 sec	0.2 - 0.9 sec	0.1 - 0.2 sec	0.01 - 0.03 sec
Trading Volume Market Share	9.05%	41.45%	8.8%	7.5%

CONCLUSION

CLOB DEX

Evolution: AMM → CLOB dominance

- 2020-2022: Uniswap AMM rules (95% share)
- 2023-2024: dYdX pioneers CLOB (~40% share)
- 2025: Hyperliquid dominates (71% share)

Key Benefits

✅ Minimal slippage on large trades
✅ Professional trading tools
✅ CEX-like experience, DEX security

Questions?

Thank you

Vault

ERC-4626 Spec: https://eips.ethereum.org/EIPS/eip-4626
ERC-7540 Spec: https://eips.ethereum.org/EIPS/eip-7540
OpenZeppelin Docs: https://docs.openzeppelin.com/contracts/5.x/erc4626

https://dev.to/ernestothagreat/how-to-build-a-liquidity-pool-smart-contract-using-erc-4626-2im8

Clob

https://www.imperator.co/resources/blog/what-is-dydx-blockchain-presentation

https://rocknblock.io/blog/how-does-hyperliquid-work-a-technical-deep-dive#hyperliquid-s-architecture-the-custom-l1-blockchain