Why?
Why OpenDEX?

1. Most **existing DEXes are built on Ethereum** -> synthetic custodial assets like WBTC.

2. Most **existing DEXes settle on-chain** -> slow & expensive; only large trades economically make sense.

3. Orderbook DEXes suffer from **low liquidity**, AMM DEXes from **impermanent loss**

4. While aggregators help to access distributed liquidity, they introduce their own centralization & underlying **liquidity still stays fragmented**
How?
How is OpenDEX different?

1. Only use native assets, trades settle cross-chain

2. Settle on layer2: lightning & connext networks

3. Orderbook based DEX, but with AMM-inspired incentive scheme for liquidity providers

4. Establish a **DEX protocol standard** for payment channel DEXes to avoid liquidity fragmentation
How does it work?

OpenDEX "BOLD" Protocol (simplified):

t0

Alice

lnd

lnd

connext

connext

Bob
How does it work?

OpenDEX "BOLD" Protocol (simplified):

Sell 1 BTC for 50k USDT
How does it work?

OpenDEX "BOLD" Protocol (simplified):

Buy 1 BTC for 50k USDT

Sell 1 BTC for 50k USDT
How does it work?

OpenDEX "BOLD" Protocol (simplified):

Buy 1 BTC for 50k USDT

Sell 1 BTC for 50k USDT
OpenDEX "BOLD" Protocol (simplified):

1. Alice creates secret \( r_{\text{preimage}} \) and sets up HTLC using \( r_{\text{hash}} \) about 50k USDT to Bob.
How does it work?

OpenDEX "BOLD" Protocol (simplified):

1. Alice creates secret `r_preimage` and sets up HTLC using `r_hash` about 50k USDT to Bob

2. Bob checks validity of incoming HTLC and sets up HTLC about 1 BTC to Alice using the same `r_hash`
OpenDEX "BOLD" Protocol (simplified):

1. Alice creates secret `$r_preimage$` and sets up HTLC using `$r_hash$` about 50k USDT to Bob

2. Bob checks validity of incoming HTLC and sets up HTLC about 1 BTC to Alice using the same `$r_hash$`

3. Alice reveals `$r_preimage$` to settle the incoming BTC HTLC and Bob uses `$r_preimage$` to settle USDT HTLC
How does it work?

opendex.network/bold
What else does OpenDEX do differently?

Decentralized Orderbook:

- Orderbook locally aggregates orders from peers
- Orders are matched locally
What else does OpenDEX do differently?

Decentralized Orderbook:
- Orderbook locally aggregates orders from peers
- Orders are matched locally

Privacy by design:
- Tor by default
What else does OpenDEX do differently?

Decentralized Orderbook:
- Orderbook locally aggregates orders from peers
- Orders are matched locally

Privacy by design:
- Tor by default

Solve the liquidity bottleneck:
- AMM-inspired financial incentives for liquidity providers.
What else does OpenDEX do differently?

Decentralized Orderbook:
- Orderbook locally aggregates orders from peers
- Orders are matched locally

Privacy by design:
- Tor by default

Solve the liquidity bottleneck:
- AMM-inspired financial incentives for liquidity providers.
- With ARBITRAGE.
A High-Speed Liquidity Network

Binance
Kraken
Bitfinex

openexd (Liquidity Provider)
openexd (Day Trader)
openexd (Liquidity Consumer)
openexd (Other)

OpenDEX Network
A High-Speed Liquidity Network

- Binance
- Kraken
- Bitfinex

opendexd (Liquidity Provider)

OpenDEX Network

opendexd (Day Trader)

Boltz.exchange
Service Provider
Day Trader

opendexd (Liquidity Consumer)

opendexd (Other)

end Users
A High-Speed Liquidity Network

- Binance
- Kraken
- Bitfinex

OpenDEX Network

- opendexd (Liquidity Provider)
- opendexd (Day Trader)
- opendexd (Liquidity Consumer)
- opendexd (Other)

- Boltz.exchange
- Service Provider
- Day Trader

end Users

Liquidity
Challenges

- Local reputation scores
- Rebalancing of channels
- Scaling the p2p layer
- UX, UX, UX.

Join us @ opendex.network
Thank you!