In this report, I’ve dug into the designs of pure play dark pools like Renegade and Tristero and private defi infrastructure providers such as Panther, Penumbra, Railgun, and more.

The following report dives into the following:

1. How these designs approach pre-execution privacy with native, third-party, or zkKYC’d wallets
2. Order discovery and matching mechanisms using a range of different PETs. Design choices are made across AMMs, orderbooks, batch auctions, and hybrid designs
3. Whether liquidity is sourced externally or bootstrapped internally, and the user privacy trade-offs it requires
4. Whether they aim to be OFAC compliant or not, and the methods they use to do so

Here’s a quick overview of their designs:

- Renegade, with a crossing network based design, uses a UTXO based system and will be deployed using Arbitrum’s Orbit. It uses a combination of Ultra-ponk based Zero knowledge proof system and collaborative MPC.
- Tristero with an orderbook-based design, uses an account model and a network of Intel SGX TEE nodes. It uses “salted” intents.
- Panther Protocol, an orderbook, uses a combination of a MASP and a time-based escrow contract to source liquidity externally.
- Singularity with a hybrid orderbook-AMM design, also uses a UTXO based system. It relies on a network of FHE nodes and an Ultra-plonk based zero-knowledge proof system.
- Penumbra with an CL-AMM based on a UTXO model. It uses a groth16 ZK proof system with homomorphic pedersen commitments to run order batching system against a whole host of individual concentrated liquidity pools.

[

https://distributedresearch.substack.com/p/diving-into-dark-pools

944×813 89 KB

](https://collective.flashbots.net/uploads/default/original/2X/f/f95b173dac3d6f4cb6e56e42b8500076062d49d4.png)

The use of PETs depends entirely on the dark pool design. If we’re going with a pure order book crossing network, ZK-MPC may be sufficient. However, such designs would require active mechanism design efforts to govern their relayer network.

Designs using TEEs can hit the ground running with low latency execution but would need to have additional security measures in place.

For a dark pool to be useful, it needs to have sufficient liquidity to match the demand side of order flow. So, if you’re running an orderbook, you need to have a sufficient number of counterparties or market makers or solvers to be able to sustain it. But if a dark pool is sourcing liquidity externally, it would need to rely on FHE, HE, and other cryptographic methods to be able to source that liquidity externally without compromising on user privacy.

Continue reading here