StarkWare

Name: Jiatian Wang

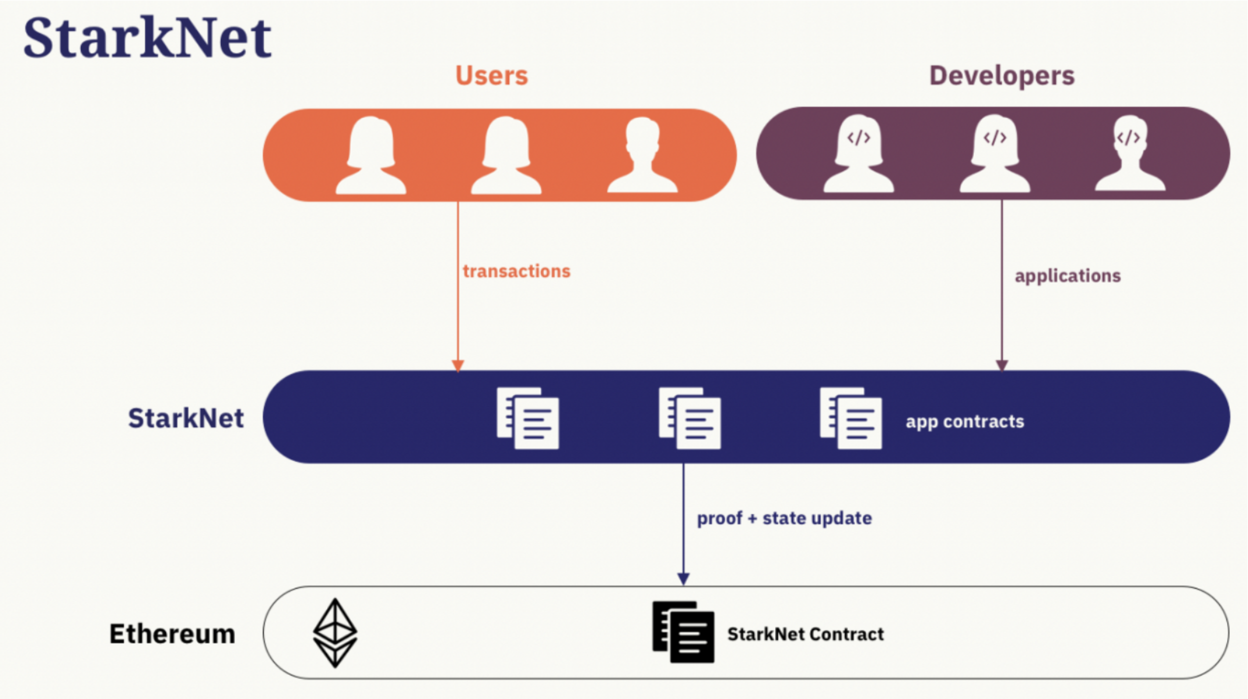
DiscordAccount: kartin#7394

StudyGroup: `CryptographicReserach`

Assignment: `zkSNARKs Application Survey Report`

Repository: https://github.com/kartinW/zkcamp/blob/master/{kartin}-{StarkWare-jiatianwang}-220710.pdf

StarkNet is a permissionless decentralized ZK-Rollup that supports independent deployment of smart contracts. Any developer can write and deploy their smart contract permissionlessly. StarkNet also supports composability.



StarkEx is a permissioned tailor-made scaling engine, designed by StarkWare to fit the specific needs of apps. Language: **Cairo**, **Shared Prover.**

Both StarkNet and StarkEx provide scalability and L1 security by using STARK-based validity proofs, and both are designed to support general computation, allowing any use case to be scaled.

Customers: dydx, immutableX,

Data Availability Committee: infura

Comparing with SNARK, STARK is “no trusted setup needed” and “post-quantum secured”

ZK-Stark paper: https://starkware.co/wp-content/uploads/2022/05/STARK-paper.pdf

ZKSTARK Steps:

1: Batching

user txs are batched off-chain by the operator, and sent to the StarkEx service

2: Validating & Updating

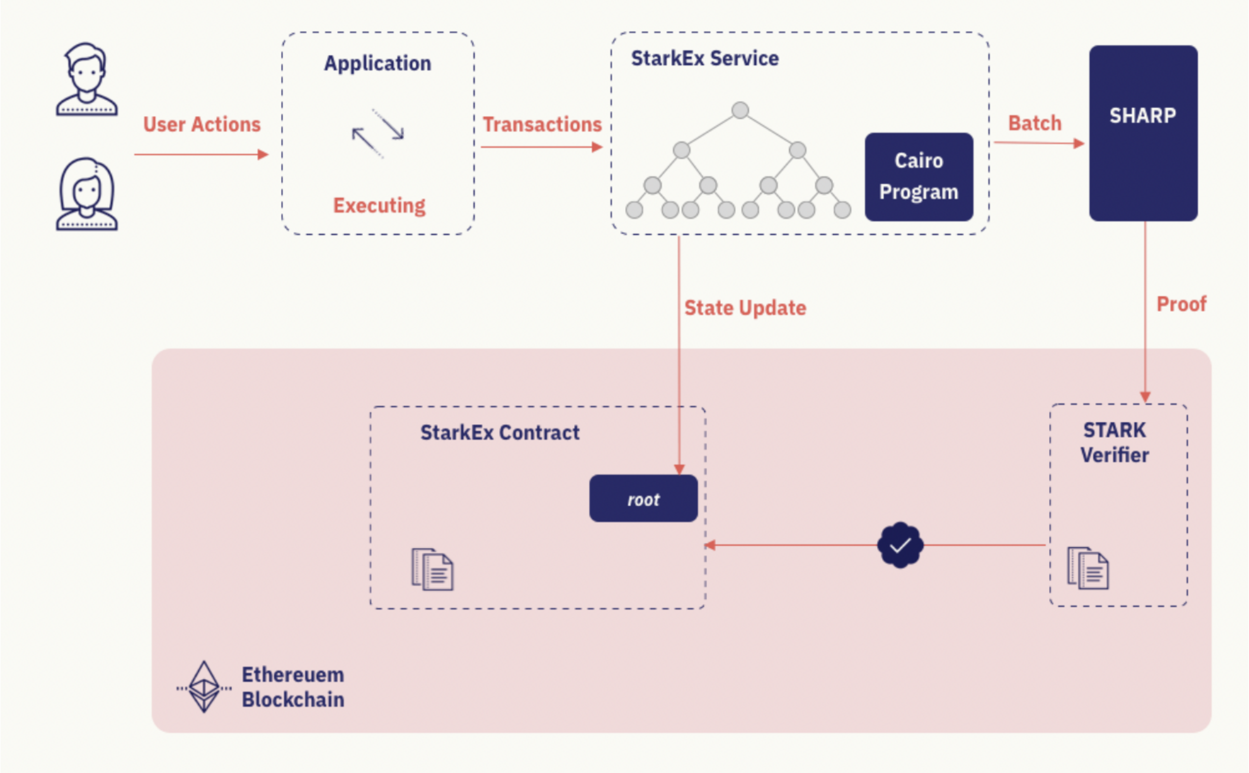
The StarkEx service validates the txs in the batch and the relevant balances are updated

3: Generating a proof

The StarkEx service generates a STARK proof, attesting to the validity of the txs in the batch, and sends the proof on-chain

4: on-chain verification

an on-chain verifier smart contract receives the STARK proof. Once the proof is verified, a commitment to the new balance states are stored on-chain



Lowest Gas fee record: 315 gas/tx