ZKU Assignment 7

Name: Shyam Patel
Discord: L_guy#4297
Email: shyampkira@gmail.com

Question 1:

Celestia is set out to be the consensus and data availability layer for blockchains. Chains built on top of Celestia can concentrate on execution. Do you think data availability is the true bottleneck to scale blockchain? Argue for and against the need for the data availability layer for blockchain.

Yes, Data Availability is a bottleneck to scale blockchain. There are 3 major ways to scale a blockchain: **increasing block size**, **sharding** and **rollups**.
Rollups are a design that uses a blockchain only as a data availability layer to dump transactions, but all the actual transaction processing and computation happens on the rollup itself. This leads to an interesting insight: a blockchain doesn't actually need to do any computation, but at minimum it needs to order transactions into blocks and guarantee the data availability of transactions. A dedicated layer (or another block chain) to handle the data availability (DA) is a very good design.
Eventually such pluggable and modular design could achieve millions of TPS. The use case of Celestia (and Polygon Avail) works as transaction storage for a rollup solution. Technically, It can be designed to be a more scalable and extensible solution like zkPorter is for zkSync. However, one could argue that such design is already done in L0 block chain (Comos, Polkadot etc) where the main chain only focuses on communication and DA these core blockchain capabilities. Then, gives L1 chains the flexibility to implement its own execution, scalability solution and tokenomics.

Question 2:

Another popular zero knowledge technology in the market today is zk-STARKs. Starkware uses this technology to power dApps such as DiversiFi, ImmutableX, dYdX, etc.. List some advantages of zk-Starks over zk-Snarks. In your opinion, which one is better and why?

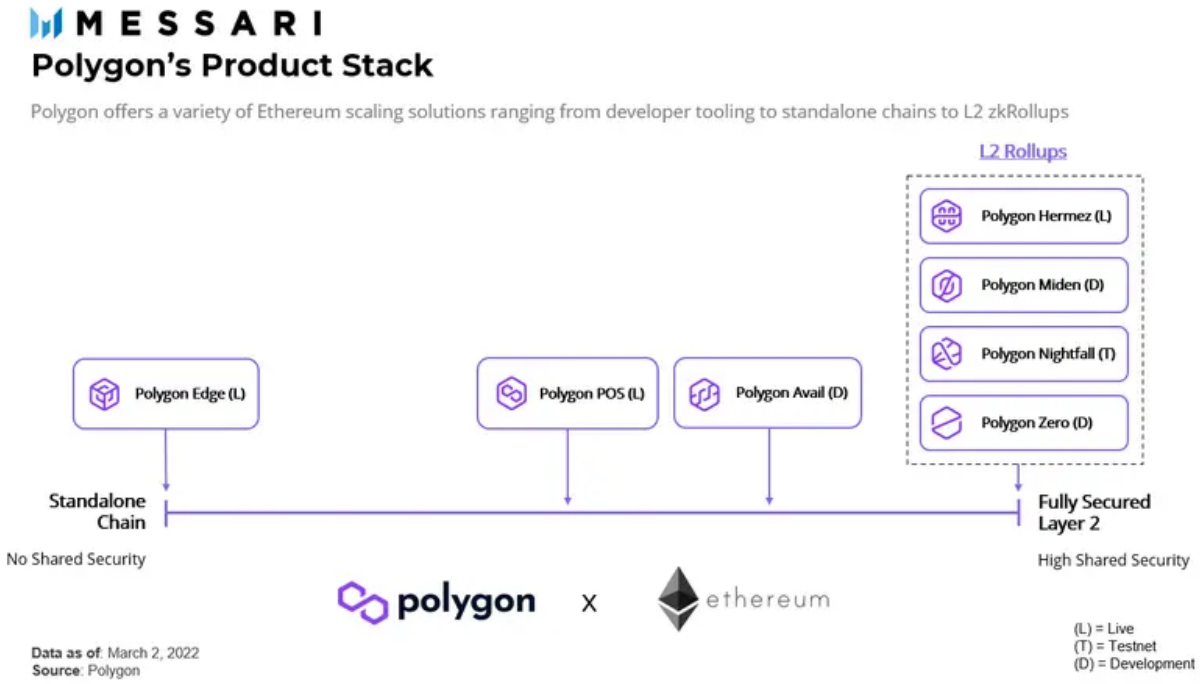
zk-Starks have several advantages over zk-Snarks:

- zk-Snarks require a trusted setup phase whereas zk-Starks uses publicly verifiable randomness to create trustlessly verifiable computation systems. Zk-Starks don't require a trusted setup but a universal and transparent setup.
- Also zk-Starks don't need to run a ceremony for every new circuit implementation, but rather one for the whole scheme.
- Thus, zk-Starks are more scalable in terms of computational speed when compared to zk-Snarks.
- zk-Snarks are vulnerable to attacks from quantum computers due to the cryptography they use. zk-Starks rely solely on Hash functions that are post-quantum secure (doesn’t use elliptic curve)

From a technological point of view zk-starks are superior but also they are newer. The community and tool chain for zk-Snarks are more developed than zk-Starks.

Question 3:

Write in brief (1- 2 lines for each) about the polygon's product stack. Refer this [Polygons ZK Product Overview]



Polygon PoS Sidechain:

A layer 2 scaling solution that achieves unprecedented transaction speed and cost savings by utilizing side-chains for transaction processing. Polygon uses its own Proof-of-Stake (PoS) blockchain and Commit Chain connectivity to help scale the Ethereum network, and seeks to solve inefficiencies that may hinder widespread adoption of blockchain technology.

Polygon Hermez (Decentralized and active, ZK Rollup):

Polygon Hermez is an open-source ZK-Rollup optimized for secure, low-cost and usable token transfers on the wings of Ethereum. A zk-Rollup L2 chain that is essentially used for business operations. The first mainnet version will only be suitable for value transfer and will not be EVM-compatible. **Key information:** 2000 tps (and rising) for 90% less gas costs.

Polygon Miden (STARKs Based, ZK Rollup):

Polygon Miden is layer 2 scaling solution for Ethereum. Miden relies on zero-knowledge technology (zk-STARKs) to "roll-up" thousands of layer 2 transactions into a single Ethereum transaction, thus, increasing throughput and reducing transaction fees.

Polygon Zero (Need for speed, Recursive SNARKs, ZK Rollup):

A zk-Rollup L2 chain that utilizes recursive SNARKs (Plonky2). What separates Polygon Zero from other ZK scaling solutions is the power of Plonky2, our groundbreaking prover system, which generates ZK proofs faster than any other existing tech. Revolutionary in its time to generate proofs (170 ms).

Polygon Nightfall (Privacy for Enterprises):

It is aimed to lower the transaction cost of ERC20, ERC721 and ERC1155 token private transfers. It uses Optimistic Rollup for lowering the costs and the privacy is attained by leveraging zero knowledge proofs. The optimistic rollup contracts are deployed on Ethereum (layer 1). Validators will roll up transactions into blocks and submit them to the Optimistic contracts. Challengers will submit fraud proofs for any invalid block to the same contracts.

Polygon Avail (DA for Ethereum):

Avail enables modular chain design where various execution environments can use Avail for data ordering and availability. Avail is agnostic to the execution layer, which can be the standalone chains or off-chain scaling solutions that use Avail. The applications hosted on these execution layers enjoy the full security of Avail. Kinda like Celestia.

Polygon Edge (Made With Developers in Mind):

An SDK made to build modular blockchains: kind of like Polka Dot's substrate, Cosmos and tendermint, or Avalanche's SDK.

Question 4:

Write in brief (at least 4 -5 lines) about your learnings throughout the course.

It has been an incredible journey studying at ZKU. Personally my learnings are best when it happens by actually doing stuff and interacting with people doing with me. Cohorts like ZKU are one of the best ways to do it. As a cherry on top of a cake we get paid for actually learning stuff.

- ZKU gave me a good understanding of the current blockchains' infrastructures, challenges, and potential solutions. I also got to start learning rust which I was planning to get started on since forever. Thanks to this cohort now I can read and write rust codes. Most importantly, It got me hooked on ZK-proofs and cryptography.
- It helped me get an understanding of ZKPs in use i.e. zk-snarks, zk-starks, metrics used to evaluate them. I also got a better understanding of SNARKs (trusted setup, power of tau, common random string, etc.), STARKs (transparency, proof size...) and other zero-knowledge mechanisms.
- I got to explore fascinating stuff like DarkForest, Modular Blockchain Design, Requirements of long term sustainable blockchain, and many other great concepts which convinced me to stay in web3 space for a long time.

Question 5:

Provide 2 - 3 ideas for your final project. Explain the pros and cons of each idea. Also, provide a draft proposal for the idea of your liking. Refer here for samples.

1. Transacting Amounts based on proofs

Goal: Crowdfunding or individual funding

Problem Statement:

Entity A has resources(money) to offer and a job/work which has some number of tasks. Entity B has the skills and time to complete those tasks and needs resources in exchange for completing the tasks. Among Entity A and B, both of them may or may not want to reveal their identities.

Intended Solution

Entity A sets checkpoints beforehand and submits the tasks that are needed to be done. Funds are then sent to a contract which will verify if entity B has completed the tasks required by entity A as mentioned by submitting proofs of the tasks. Entity B will be sent the rewards based on the completion and verification of milestones/tasks.

One more scenario for this can be that both entities are the same. To elaborate an entity wants to complete tasks but wants to ensure it gets completed correctly. Some examples for this can be 30-days coding challenge or a TODOs tasks list. User beforehand sets some goals and stake funds to the contract and will redeem it after completing milestones gradually.

2. Enhancements in DarkForest

I am not sure yet how vast and what number of things has DarkForest covered but it seems incredibly complex. I saw a proposal for minimal DF by one of the previous ZKU students. Can maybe work on a variant of MinimalDF that helps someone else to help understand the game better.

3. Provide Portings on other projects

I have explored some ideas on portings. Some of them seemed quite doable like adding polygon chain support to https://github.com/harmony-one/ethhmy-bridge.sdk. I have yet to explore how I can modify existing circuits to implement similar stuff related to it like changing plumo's circuit to work on harmony's consensus (or another blockchain's consensus) Or maybe build a ZK-based bridge for connecting chains.