Assignment 7

Please make your code (if relevant in any question) public on Github and attach the repo link in your submission.

Submission goes here.

Question 1: [**Both streams**] 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.

Answer Q1:

Yes I think data availability is the true bottleneck to scale blockchains and it's the price paid for internet's sovereignty, So I choose to argue for the must of maintaining data available at all times, also I can go further for all chain's history inclusion at nodes, whether at master nodes archiving in a decentralized/distributed and un-cofescatable way, light nodes need to retain the ability to acquire the same knowledge level required at time needed relative to their main use cases.

Question 2: [**Both streams**] 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?

Answer Q2:

Specifically in the case of Starkware's licensing dilemma I prefer zk-Snarks, even though harnessing STARKs in open source and License free context is much better, however in many cases I'd prefer SNARKs for it's ability to avoid computation complexity, ease of auditing and also preserving transparency while minimizing gas costs, but definitely I will keep an eye on every advancements in STARKs waiting for it to be market ready (open-source/developer friendly).

Question 3: [**Both streams**] Write in brief (1- 2 line for each) about the polygon's product stack. Refer this Polygons ZK Product Overview

Answer Q3:

Polygon Hermez: Decentralized and Active

Albeit without ZK EVM compatibility Hermez places a high emphasis on decentralization. It is the only active Layer-2 without a need for a centralized operator.

Polygon Zero: Need for Speed

A project for decentralized applications powered by recursive ZK Proofs, Unlike Polygon Hermez – which devotes the majority of research resources towards decentralization – Polygon Zero's target is speed.

Polygon Miden: STARKs not SNARKs

Miden's advantage lies in the creation of Miden VM, a ZK-STARK compatible with EVM. Miden VM is a general-purpose ZK Virtual Machine, which lets developers utilize the full Turing completeness capability of the platform. In addition, multi-language support is offered for developers.

Polygon Nightfall: Privacy for Enterprises

The idea behind the pairing of ZK cryptography with a fraud proof rollup originated from the need to service enterprises with a differentiated product, one that kept the privacy elements of ZK cryptography while maintaining low transaction costs.

Polygon Avail: Data Availability for Ethereum

Avail is a data availability-specific blockchain designed for standalone chains, sidechains, and other scaling technologies – meaning the entirety of the Avail chain is purposed to store Ethereum 'calldata' tracking changes to the Ethereum state machine. No smart contracts are intended to be deployed on Avail, and no applications can be built on Avail either.

Polygon Edge: Made With Developers in Mind

Polygon Edge is an open-source modular blockchain development framework built for engineers who want to create their own blockchains. The framework allows for the creation of both secured chains (Layer-2 blockchains) and standalone Ethereum sidechains.

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

Answer Q4:

Aside from the fact that I learnt a lot about zero-knowledge proofs, I learned about BFT consensus chains, about the organization techniques they undergo to insure fault-tolerance, I learned about many cryptographic primitives and their security assumptions, and about networking complexities, also about fraud proofs, and how different projects harness edged technology to compensate their trad-offs to meet their market use cases.

Question 5:

[**Stream A**] 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.

Answer Q5:

Idea 1:

I would to explore the case&methodology for porting Plumo to Harmony chain, and may solve even a little piece of the issue, like building a HVM precompile for Plumo's BLS.

Pros: I've been working for the last months on porting sikorka "old Ethereum project" to Harmony, but it's main point of weakness is the absence of a liable light client, so I thought researching Plumo would help.

Cons: An already trial to implement a blst like plumo's BLS faced an issue Due to the base point is different between Harmony and Eth2, so the project had not merged.

Idea 2:

I tried to construct a write up for an algorithm in assignmnet4, for providing a time-stamped ZK primitives, by providing access to a black pox that works as a beacon deriving Harmony's chain underlying security guarantees, I may build it as system consists of smart contract and a centralized back-end server.

Pros: I've been working for the last months on porting sikorka "old Ethereum project" to Harmony, but one of the main point of weakness related to the use case intended is the complexitiy of syncronizing it's core smart contracts across chains

Cons: I can find any similar approach on coding level

Idea 3:

I am fascinated by dark forest background culture, I can find something to contribute to an already existing approach on Harmony trying to build it's native dark forest game.

Pros: I like joining the gaming community enthusiasm, already tested a lot code base

Cons: In case of dark forest, it would be hard to add value in the realm of game real timing, light-client implementation.

[**Stream B**] Please provide an update on what you have achieved on your final project and what you plan on doing next.