# Assignment 7

### ****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.*

**[ANSWER]**

Yes.

There are 3 major ways to scale a blockchain: increase block size, sharding and rollups. Together, these approaches will end up increasing the chain’s size exponentially. It’s not technically and economically feasible to store and verify all transaction data in one node or client. It must be stored in distributed manner and is always available to all kinds of clients. As separation of concerns, I believe a dedicate layer (or another block chain) to handle the data availability (DA) is a good design. Such pluggable and modular design could eventually give us unlimited scalability.

I especially like the use case that Celestia works as transaction storage for rollup solution. Technically, I feel it can be a more scalable and extensible solution for zkSync than zkPorter (zkSync’s own chain).

I’m enlightened to realize that Celestia is startup and developer friendly. It eliminates the needs to build and incentivize validators’ community which is too costly to a small project. This work can be outsourced to Celestia through its own validator network.

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, such kind of core blockchain capabilities. Then, gives L1 chains the flexibility to implement its own execution, scalability solution and tokenomics. It’s unclear to me that how much great advantage Celestia can gain in this fast-moving field. Or it can focus on supporting rollup solution.

### ****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?*

**[ANSWER]**

Advantage

1. zk-Snarks require a trusted setup phase whereas zk-Starks use publicly verifiable randomness to create trustlessly verifiable computation systems.
2. zk-Starks are more scalable in terms of computational speed and size when compared to zk-Snarks.
3. zk-Snarks are vulnerable to attacks from quantum computers due to the cryptography they use. zk-Starks are currently quantum-resistant.

Disadvantage

1. The proof size of zk-Starks (> 10KB) is much bigger than zk-Snarks (~0.2 KB), It consumes more gas to verify in smart contract environment and is also limited in mobile use case (that’s the reason Celo choose zk-Snarks).
2. Since zk-Starks is younger. The community and tool chain for zk-Snark is more developed than zk-Starks. To highlight, there is no solid implementation for recursion (although possible) which is quite important for bridge and light client project.

I feel zk-Starks is technically superior but for my current project, I would still choose zk-Snarks over zk-Starks as my underline ZKP technology to reduce risk.

### ****Question 3**:**

*Write in brief ( 1- 2 line for each) about the polygon’s product stack. Refer this* [*Polygons ZK Product Overview*](https://messari.io/article/polygon-a-multi-sided-approach-to-zk-scaling)

**[ANSWER]**

### ****Question 4:****

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

**[ANSWER]**

### ****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*](https://talk.harmony.one/c/funding-proposals/zkdao/84)*.*

**[ANSWER]**