CELESTIA

Celestia and The Role of Data Availability

Celestia-Enabled Modular Architecture

By Jake Kennis and Seth Zhuo

Disclosure: The authors of this content and members of Nansen may be participating or invested in some of the protocols or tokens mentioned herein. The foregoing statement acts as a disclosure of potential conflicts of interest and is not a recommendation to purchase or invest in any token or participate in any protocol. Nansen does not recommend any particular course of action in relation to any token or protocol. The content herein is meant purely for educational and informational purposes only and should not be relied upon as financial, investment, legal, tax or any other professional or other advice. None of the content and information herein is presented to induce or to attempt to induce any reader or other person to buy, sell or hold any token or participate in any protocol or enter into, or offer to enter into, any agreement for or with a view to buying or selling any token or participating in any protocol. Statements made herein (including statements of opinion, if any) are wholly generic and not tailored to take into account the personal needs and unique circumstances of any reader or any other person. Readers are strongly urged to exercise caution and have regard to their own personal needs and circumstances before making any decision to buy or sell any token or participate in any protocol. Observations and views expressed herein may be changed by Nansen at any time without notice. Nansen accepts no liability whatsoever for any losses or liabilities arising from the use of or reliance on any of this content.

TLDR; *Celestia is a pluggable data availability layer for any chain to compose on. It flips the paradigm of combining execution, consensus and data availability into a single layer 1. The modular approach allows devs to focus strictly on the application layer and not the consensus layer. The team is extremely diverse with experience from scaling blockchains, Eth 2.0, Tendermint, Harmony and much more.*

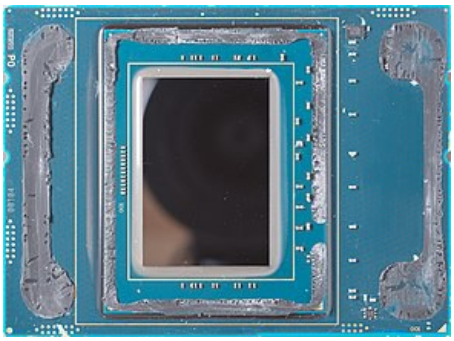
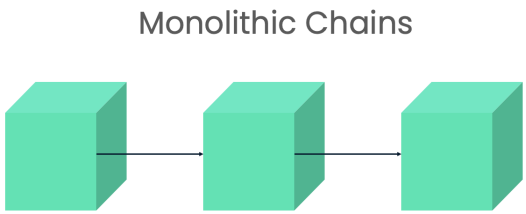
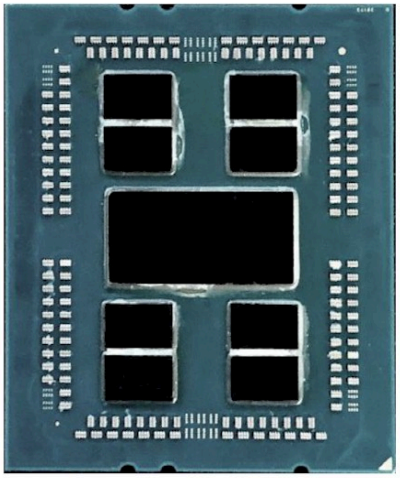
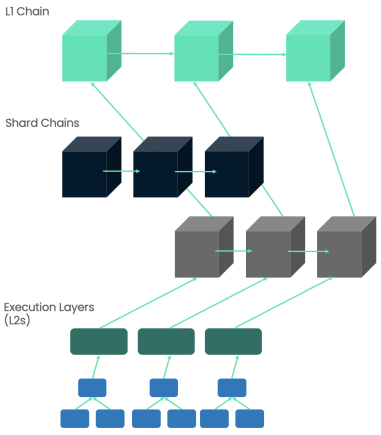
Celestia design and future roadmap

- ***Key Features:*** *data availability proofs, erasure coding, limited execution environment, light client security guarantees similar to that of full node,*
- ***EVM rollups:*** *would send attestations to Ethereum mainnet - cheaper than posting the data as these are just signatures attesting to the data posted on to Celestia*
- ***Devnet:*** *launched on December 14th, 2021 and mainnet is expected to launch in Q4 of 2022*

Introduction

Current State of Blockchains: Monolithic vs Modular

The Modular Revolution for CPU processors provides a strong reference point and direction of the trend of modularization in blockchain. Similar to CPU Monolithic Dies, a Monolithic Blockchain is limited by the processing power of a single point (CPU die/Validator Node). Currently, Solana and BSC are good examples of Monolithic Blockchains. The Scalability Trilemma describes scalability, decentralization, and security as trade-offs faced by monolithic blockchains, and is well discussed by Vitalik Buterin. In the above examples, Solana/BSC have sacrificed decentralization in favor of scalability because they increased the hardware requirements to spin up a validator.

	CPU Processors*	Blockchain
Monolithic	 Monolithic Dies in CPUs are used for the longest time. In 2019, Intel's very best was a Monolithic Die CPU: Xeon Platinum, a 28 core CPU costing $10,000	 Monolithic Blockchains are blockchains that try to do all three things: consensus, execution, and data availability, all on the L1. Examples are the POW/PoS blockchains of Eth 1.0/Solana/BSC.
Modular	 In 2019, Intel's very best chip was completely obliterated by AMD's Modular Die CPU: a 64 core CPU (3X performance), costing $4,000 (<50% costs). CPU Dies have gone in the direction of modularity since.	 The modular blockchain uses the three components of the monolithic blockchain currently on L1 and separates them to varying degrees. Shards, L2 rollups, and DA layers are examples of modular blockchain designs.

Source: Polynya

Innovations in the space of Modular Blockchains have unlocked new possibilities. Modular blockchain designs include ZK/Optimistic rollups, sharding, and most recently, specialized data availability (DA) layers sitting at the bottom layer of the modular stack. Celestia pioneers the DA

layer of the stack which is something that every blockchain needs. We will further discuss DA and the network effects that Celestia can create in this report.

What is data availability and what will it allow?

Celestia reimagines the way decentralized app development is carried out and will significantly lower the barrier to entry for new blockchains. In line with the Cosmos vision, they will enable a world that can support millions of chains. With Celestia, they are taking a fundamentally different design approach for building blockchains. It does this by building out a very modular bottom layer that allows developers to deploy blockchains on top of; hence, anyone can spin up a rollup on top of them and can focus strictly on app development while inheriting shared security through Celestia.

To build out the most basic blockchain, it needs mainly 2 things - transaction ordering and data availability. From those two things, you can build out any application on top of that where execution takes place. This is the design of Celestia to be "lazy" (formerly known as Lazy Ledger) and they do these two core things in a scalable way. Unlike Ethereum where the entire stack includes an execution environment, consensus and data availability all in one, Celestia decouples execution from the state machine - they only order transactions and provide DA for blockchains that compose on top of it with a very limited execution environment. Removing an execution environment will allow Celestia to scale much more than its scaling counterparts such as Interchain Security via the Cosmos Hub or Polkadot's Relay chain model. State execution is expensive and if these models are to become wildly successful, it will become too expensive for long tail chains and rollups to take part in these shared security models. We will dive deeper into the design space of shared security models and their tradeoffs.

The big question you may be wondering is why do we need a generalized data availability solution in the first place? Let us first define 'data availability' and what it means in the context of scaling blockchains. Then, we will discuss Celestia and why its solution can provide a more scalable approach than others today.

Data availability

Any given blockchain has nodes that validate the network. These can be broken down into 2 categories - full nodes and light clients. To understand Celestia, let's do a quick recap of some of these core concepts.

- **Full Nodes**

    - Otherwise known as validating nodes, these nodes require a lot of resources because they download and validate every transaction on the blockchain. Full nodes offer a lot of security guarantees because they can't be tricked into accepting blocks containing invalid transactions.
- **Light Clients**
    - Given Full Nodes are resource intensive, some participants will choose to run a light client. The key distinction here is that light clients do not download or validate any of the transactions. Rather, they just download the block header and assume that the block contains valid transactions. Hence, Light Clients do not offer the same security guarantees as Full Nodes.

Due to fraud proofs, light clients can indirectly check that all the transactions in the blocks are valid. The full nodes can send over a fraud proof to the light clients if a block contains an invalid transaction, instead of the light clients checking the transactions themselves. The issue at hand is that a full node is required to know the transaction data for that block in order to generate a fraud proof. Therefore, if a block producer publishes the block header but not the transaction data, the full nodes won't be able to check if the transactions are valid and generate fraud proofs if they're not valid.

Given that we need a block producer to publish all of the data in their blocks, is there a way to actually enforce this? The answer circles back to the use case of light clients - enabling a way for light clients to check if the transaction data for a block was actually published to the network. On the surface, this would appear to defeat the purpose of a light client but Celestia can offer a solution to this problem. We will cover Celestia's light node design further in the report. Below, we highlight the tradeoffs of the scaling solutions and will later dive into the design of Celestia.

Scalability Solution	Design	Scalability	Tradeoff
Increasing the block size	Bitcoin has a small block size because it makes it easier to run full nodes on a laptop.	BTC has an artificial block limit to keep the blockchain small and let it be easier to run a full node. If block size is increased to increase TPS, it will be harder for an individual to run a full node.	If the block size was increased, it'd be easier for block producers to insert invalid txs that light clients would accept as valid.
Sharding	Split the blockchain into multiple shards (blockchains) to increase throughput of a chain.	Block producers are split up in the network so that the processing power is split into different shards to process only some of the transactions.	A full node in a sharded blockchain can run a full node for one or a few shards and run a light client for the rest. B/c block producers are split into different shards, it is more likely that block producers in a given shard can become malicious and accept invalid transactions.
Optimistic Rollups	Have block producers that can transfer assets to and from other chains. They post the rollup's blocks onto Ethereum as a data availability layer.	Rollups use Ethereum as the data availability layer which competes with other smart contracts for gas fees. This forces higher fees to post these transactions and could cause scalability concerns given the demand for blockspace.	Use fraud proofs to detect invalid transactions. The blocks are put onto Ethereum for data availability. Innocent until proven guilty model. The rollup still needs data availability to post the state of the blockchain
ZK Rollups	Have block producers that can transfer assets to and from other chains. Similar to Optimistic rollups, they post the rollup's block onto Ethereum as a data availability layer.	Rollups use Ethereum as the data availability layer which competes with other smart contracts for gas fees. This forces higher fees to post these transactions and could cause scalability concerns given the demand for blockspace.	Use validity proofs to detect invalid transactions. The validity proof itself doesn't need data availability. Guilty until proven innocent model. The rollup still needs data availability to post the state of the blockchain.

Enter Celestia

Celestia is a layer 1 blockchain built out on the Cosmos SDK and secured via Tendermint PoS consensus. They do not have an execution environment that allows the consensus to only be responsible for ordering transactions and guaranteeing their data availability ("Lazy"). Note, because it is a PoS chain, they will need some state on their chain and hence there will be some execution happening in a very limited way. However, Celestia addresses a key component of any given blockchain and reimplements DA as a modular stack as opposed to an all-in-one model. Vitalik's recent paper 'Endgame' comes to the same conclusion that Celestia is trying to build:

"So what's the result? Block production is centralized, block validation is trustless and highly decentralized, and censorship is still prevented."

This recent paper by Vitalik embodies the idea that it is expensive for block producers to produce blocks (centralized) but it'll be cheap for end users to run full nodes to verify the block producers are behaving honestly (decentralized).

The recent developments of L2s and the success of app-specific blockchains such as Osmosis have given confirmation of a multi chain world playing an important role moving forward. Given enough demand for block space, many of these monolithic layer 1s will face the same issue as Ethereum today. Yet, there has not been a scalable way to resolve the data availability problem without sacrificing some level of decentralization. Although some may argue for decentralization across block production and block validation, it appears that this uniform measure will take a more modular approach instead - centralized block production and censorship resistant, decentralized block validation.

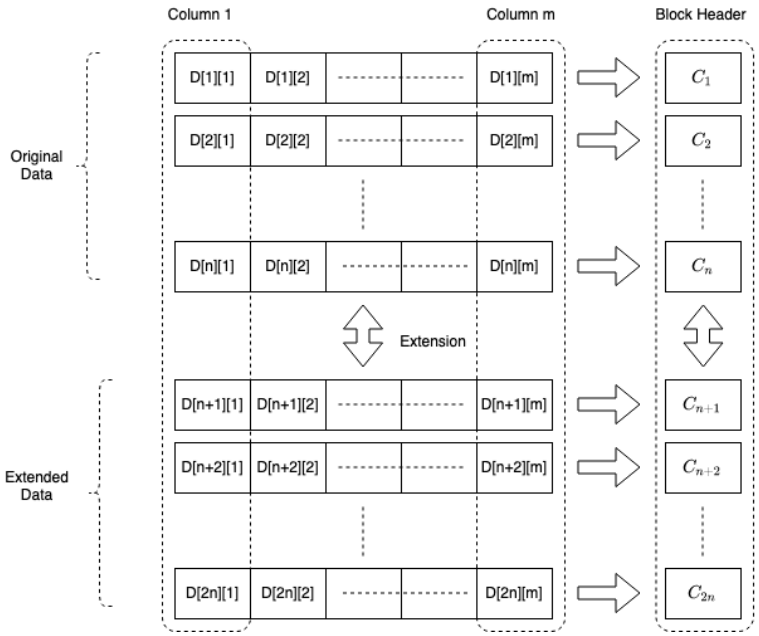
Again, Celestia is focusing on solving 1 problem - data availability. It feels like a 0 to 1 as the execution environment can scale endlessly (via Rollups) or sharding while the consensus layer (Celestia) remains scalable through some of its core primitives mentioned later in the report. A pluggable data availability solution is the missing piece in the Cosmos architecture and layer 2s more broadly. So how exactly do they enable a pluggable data availability layer?

How does Celestia work?

To resolve the tradeoffs mentioned in the scaling solutions above, Celestia is able to reduce the problem of block verification to data availability verification. It takes the data and orders it and makes sure the data is available by dispersing it through the network. This availability is guaranteed through a technique called data availability proofs. These proofs are a sampling technique that only requires each client to sample a very small number of random chunks from each block in the chain. Similar to peer-to-peer file sharing like BitTorrent, this will scale with the number of clients in the network that are sampling the entire blockchain. In short, the more clients, the more scalability you get due to a bigger block size with better security guarantees. In their data availability proofs they use a concept called *erasure coding*, which is a block encoding scheme that empowers the light nodes to verify the block data with a high probability that the rest of the block has been published.

Erasure coding lets us expand any n length of data into 2n length of data in such a way that any n out of 2n are sufficient to reconstruct the original piece of data.

Suppose a block producer is forced to erasure code the transactions tx1, tx2, …, txn. The block producer then wants to hide a single transaction. To do so, the block producer would have to hide n+1 data length of data, as any n are sufficient to construct the entire transaction set. A constant number of queries give the light client very high confidence that the underlying data is indeed available.



Source: Polygon

In short, Celestia is able to ensure with statistical certainty that the entire block has been published to Celestia. If the data is wrongly encoded, the network will be notified via a data availability fraud proof.

Other Technical Features of Celestia's MVP

Namespaced Merkle Tree (NMT)

Celestia implemented an NMT library which is a Merkle Tree sorted by namespaces. This is a key component as it makes Celestia rollup agnostic. How? Any rollup on Celestia can download data that is relevant to their chain and ignore the data for other rollups that are composed on Celestia. In order to integrate this, Celestia replaced Tendermint's Merkle tree with their own NMT.

2-Dimensional Reed Solomon Merkle Tree (rsmt2d)

As mentioned above, Celestia uses erasure encoding to ensure with statistical certainty that the entire block has been published to Celestia. For their MVP in June, they implemented a special encoding scheme called **rsmt2d**. This technique encodes the block data into a square which gets erasure-coded into a larger square with parity data. This encoding mechanism is integrated with the NMT to compute row and column Merkle roots from the encoded block. In order to do this, they had to modify the Tendermint block header to commit to these row and column roots.

IPFS Components

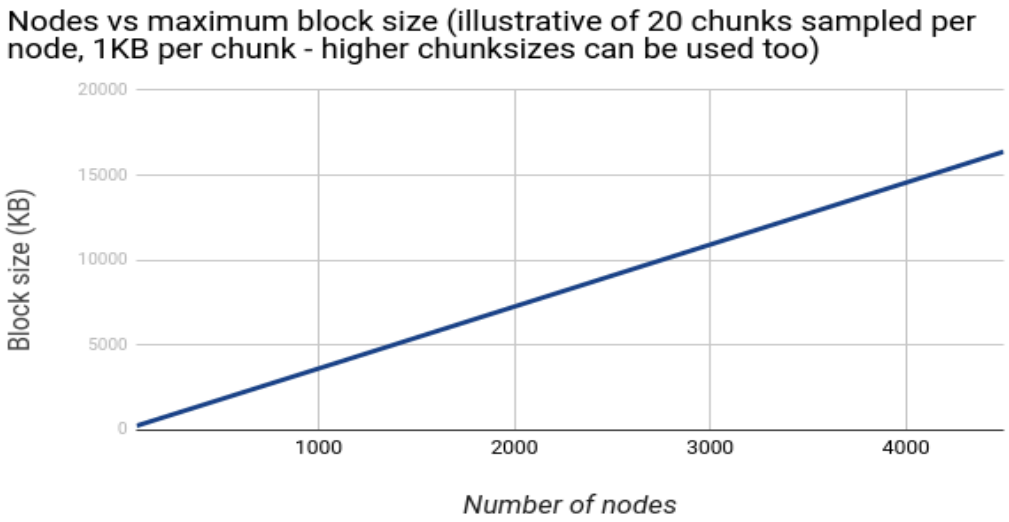
Celestia has made an architectural decision to build an augmented RPC-based Tendermint light client with an added possibility to do Data Availability Sampling (DAS). They have modified IPFS and coded an IPLD plugin to enable sampling over a P2P network, through the creation of a library for light clients and other node types to validate block availability. IPLD is a data model of the content-addressable web that is able to create a unified information space for all hash-linked data. An overview of the ADRs is posted here: MVP Light Client , How IPLD is used, How DA works .

Celestia Light Nodes

In order to make Celestia unbiased to a given rollup or chain, Celestia has built out a scalable node network. Celestia refers to their light clients as 'Light nodes' because they are more secure than typical light clients and they help scale and secure the network. Celestia's Light nodes allow individuals to participate as non-consensus nodes without the overhead of full nodes. Each light node requests a random sample of block data through a process called data availability sampling. This ensures that light nodes can verify the block data without downloading the entire block. However, there is a minimum number of Light nodes required for this data sampling to work.

In short, **light clients usually work on an honest majority assumption but now operate on a minimum number of nodes assumption**. This model would only need 1 honest full node or more to work. If there is misbehavior, it is easy to hard fork against a malicious full node.

The data throughput of the main chain increases with the number of non-consensus nodes. These nodes can increase the throughput and security of the network because they are not producing blocks. The dilemma faced by current layer 1 implementations is that things like sharding focus on scaling block production rather than block verification. However, you need both to scale the network. The diagram below shows the scalability of Celestia - the greater number of light nodes, the higher capacity for data. Scale in numbers...



Source: Celestia

Celestia is able to support a large number of rollups or blockchains. Celestia takes the data from the chains using it as a DA layer and stores them into a Merkle tree. Every chain on top of Celestia has its own namespace and the data is Merkelized and sorted per namespace. The final result is one short Merkle root but each rollup will have its own subtree. This design allows for many blockchains to use Celestia with a focus on rollups and sidechains.

If the node providers of an existing chain already have the data availability, why would they need Celestia? Celestia gives nearly the same security guarantees for a light client as a full node does, which has many implications. It does this through the probabilistic sampling-based techniques that were mentioned above. To reiterate, the Light nodes randomly sample the block and with more samples, the better the assurance. Hence, chains can take advantage of Celestia's scale and consensus and focus their development on the actual application/execution layer.

Takeaway

Celestia light clients have stronger security guarantees and engage in processes that secure the network, giving them nearly the same security as full nodes and take an active role in securing the network. These light clients are called 'light nodes' in Celestia because they are more secure than traditional light clients. The more light clients, the larger the block size can be (capacity of data) while ensuring security and decentralization. **Note, there must be a minimum number of light nodes for this security model to work in order to ensure that the original block data is recoverable from all of the samples they take individually.**

Total Addressable Market

Celestia will focus on application-specific blockchains and rollups. In just 8 months following IBC launch, 25 chains worth over $60 billion have integrated IBC with over 5 million transactions. Given the success of app-specific chains built out on the Cosmos SDK such as Osmosis, where does that leave room for vertical scaling via rollups? Celestia would provide an easier onramp for developers to deploy rollups without having to worry about security. Today, if you want to create an app-specific blockchain on Cosmos, you need to validate the network but it takes a lot of work to do so. The other solution out there is Interchain Security but you would have to pay for validators which creates a higher barrier to entry. Of course, it is always possible that Celestia implements Interchain Security to enshrine an optimized settlement layer but this is strictly hypothetical.

Ultimately, bigger chains will have an advantage for using Interchain Staking via the Hub and it doesn't leave much room for long tail chains. If the multi-chain vision plays out and we have thousands/millions of blockchains, then it seems unlikely for all of those chains to be secured via the hub through Interchain Security given those validators will be strained for capacity to support all of them. The main difference between Celestia and Interchain Security is that Celestia is aiming to scale to millions of chains with a particular focus on the long tail with a low barrier to deployment. Rather than having your own validator set, these apps can now use Celestia as their consensus network.

One might ask, why does an app need to be its own specific chain? In short, it opens up a new design space, enables self sovereignty, runtime customization and performance. For instance, Osmosis is its own layer 1 blockchain but is able to vote on the amount of gas paid or its measure of security, it is not left up to another set of governance. Further, layer 1s are able to explore new design frameworks such as Superfluid Staking which is not possible if you are not your own blockchain. This new staking framework would allow LPs to earn more yield on their capital - swapping fees, liquidity incentives and the staking rewards for securing the network, all in one.

This feature is to highlight the structural advantages of building out app-specific blockchains. However, app-specific blockchains are missing a core component - a scalable data availability layer that provides a level of shared security.

Rollups

Layer 2s use Ethereum as a consensus and data availability layer. Zk-rollups such as zkSync and Starkware move the transaction computation off of Ethereum mainnet but the data availability remains on Ethereum and the transaction data is written on the mainnet through calldata. The bottleneck to rollup scalability is the data capacity of the layer 1. Ethereum was not designed to be solely a general purpose data availability layer - it combines the execution layer, data availability and consensus into one. Eth 2.0 introduces data shards which will only process data availability and not execution but it is relatively far away in its roadmap.

Celestia does not have an on-chain smart contract execution environment and they are not a sharded chain. Rather, they take the approach of sharding on the execution layer, which are the layers above Celestia. If you want to implement Ethereum execution sharding you can do so on top of Celestia. In the same regard, if you have a really popular Cosmos Zone where the single chain is not enough to process all of the transactions, you can split that Zone into multiple Zones as rollups that communicate with each other. In short, **you can have execution sharding but that is not enshrined on the Celestia base layer**. This is similar to the Cosmos vision of having horizontal execution sharding but with a scalable shared security layer.

Starkware is resolving the data availability issue though a volition system. Users would have the ability to choose between a rollup settlement or a validium settlement. With validium settlements, the offchain data availability is guaranteed by a centralized DAC that is made up of reputable entities such as Consensys, Infura and more. However, this is rather centralized and that is the tradeoff for cheaper transactions. ZK-rollups, in their current form, use a single aggregator. An aggregator is equivalent to a block producer. In short, the data availability situation with rollups boils down to two methods - through closed committees that attest to the data availability (validium) or the data ends up on Ethereum which becomes very expensive.

Celestia's main chain is focused strictly on data availability whereas Ethereum will always have hundreds or thousands of smart contracts that are deployed on it and competing for gas. This makes Celestia an ideal place for all types of rollups to use as a DA layer - regardless if it is an optimistic rollup or a zk-rollup, they both need DA. The rollups using Celestia can be broken down into two categories - **Ethereum-based rollups** and **rollups composed on Celestia** itself. For

Ethereum rollups, they would use Celestia as a DA layer and Celestia would attest that to Ethereum. The flow is as follows: Ethereum Rollup → Celestia →Ethereum mainnet. At no point does Celestia attest the block data back to the L2, which makes it rollup agnostic. Celestia will only post the attestations onto Ethereum and optimize the gas costs by batching the attestations. Note, Celestia will not be posting the data, but rather a bunch of signatures and merkle roots which is a lot cheaper than the data itself. If $ETH and gas prices continue trending upwards, then at some point everything will be expensive which poses risk. There are some optimizations that can be employed in a world where gas prices continue to rise such as further batching, threshold signatures, or fine-tuning other parameters. Regardless, Celestia posting signatures is a much cheaper option than rollups posting the data itself.

There are some differences between how rollups use Ethereum and how they'd use Celestia. The main difference is that using Celestia, the rollups would only post the data and the validity proofs for a ZK-rollup. The data and validity proofs are locked into Celestia but they are **not verifying the correctness of the data**. The verification is happening somewhere else, taken up by the rollup and can happen locally - the validity proofs would need to be verified elsewhere by the rollups. On the other hand, if a Zk-rollup uses Ethereum, Ethereum has an execution environment where they have smart contracts to validate the validity proofs on-chain.

Given the relationship between Celestia and potential rollups, let's better understand the connector agents between them. Connector agents are everywhere - in Polkadot, you have the collator, in IBC you have relayer and ZK-rollups have agents between them and Ethereum. Agents are running clients on both sides. However, Celestia does not have a two-way bridge to the rollup, so you do not need a client of both chains running; rather, only the rollup, likely the sequencer, will run a client of the Celestia chain. These sequencers will take the rollups data and submit the transactions to Celestia and pay for the inclusion of the data. Again, this design makes Celestia agnostic to any rollup solution because it doesn't know what the data means that the rollups are giving it.

For an Ethereum rollup that uses Celestia as a DA solution, the validators of the Celestia chain would submit attestations to Ethereum on behalf of the data sent to them from the rollups but they would not have direct interaction with the rollups themselves. This communication between Celestia and Ethereum would be done through the **Quantum Gravity Bridge**, which is essentially a relayer from Celestia to any EVM-compatible chain. The Ethereum rollup clients would read the attestation that Celestia posts on to Ethereum and verify the signatures of the validators. Note, rollups in their current form are Ethereum focused but they can exist anywhere. Rollups launching directly on Celestia are not just theoretical, but we are actually seeing these rollups today. Cevmos, an EVM rollup on top of the Evmos settlement layer, has already planned to integrate

Celestia as a consensus layer. We will cover this more later in the report but it shows the use case of rollups outside of Ethereum.

Rollups that deploy directly on Celestia do not need to be built on the Cosmos SDK. In fact, the Cosmos SDK makes it hard to make the state fraud proof (Optimistic roll up style) provable because the Cosmos SDK is not like the EVM. Rather, the Cosmos SDK is very general and not well defined, whereas the EVM is specific, well defined and has a constrained execution environment. This makes it difficult to enable fraud proofs in its current form. For now, Celestia is exploring the use of the EVM, in particular Arbitrum's VM, for the default execution environment over the Cosmos SDK. There are many reasons for this path but it boils down to Arbitrum because it works in its current form. Eventually, they will introduce other execution environments such as Substrate and Zk-rollups. As of now, the optimistic rollups are more mature and you can easily deploy something whereas Zk rollups are still in development. Celestia is able to and plans on supporting many execution environments, with its go-to market being optimistic rollups. Hence, down the line, Celestia can even provide shared security for Substrate chains or even other Cosmos layer 1 chains themselves!

Celestia vs Polkadot vs Interchain Security

Project	Design	Scalability	Security
Polkadot	The relay chain has all of the parachains locked into it and the relay chain is the primary source of consensus.	The Relay chain has a limited number of execution spots. They are auctioning off the parachain slots because there is only a limited number of parachains that can connect to it. If it becomes successful, parachains will become far too expensive which is a problem given it limits new experimentation and limits the devs from capturing value for themselves. State execution is expensive and that is why they limited it to 100 execution slots.	Shared security model
Ethereum	Shared security model for all apps that deploy on it.	Etheruem is facing very high gas costs and is having issues scaling at the base layer. Layer 2s are currently scaling the network but ultimately use Ethereum as the consensus and data availability layer which is expensive. Rollups have to compete for gas with non-rollup transactions (any existing smart contract), resulting in higher fees and limited scale	Shared Security
Interchain Security	Shared security via the Cosmos Hub is similar to the Relay chain on Polkadot and provides shared security. The validators of the Hub need to do the execution as well as the state - a 2 node system. Note, Interchain Security can be implemented by any Cosmos chain given it is open source software.	Can provide an opt-in method of shared security to sovereign Cosmos Zones by 'borrowing' security from the Hub. This method requires you to pay for validators which creates a higher barrier to entry and it is unclear if this will be able to scale to thousands or millions of chains. Will likely be complimentary with Celestia.	Shared Security via the Hub.
Celestia	The Celestia main chain has multiple sub chains locked into Celestia that enjoy shared security. It solves 1 problem, data availability, really well whereas other chains have fundamental limitations to them.	Celestia decouples execution from consensus, so it is not limited in the number of chains it can support - it can have an infinite number of execution slots on Celestia. Can provide permissionless deployment, especially to the long tail of chains. Celestia uses Tendermint so it will be limited between 100-200 validators so Celestia will be similarly limited. It doesn't matter how expensive a validator is or how many validators there are, as long as you have the ability to hard fork and socially coordinate - those validators can't do anything bad. The limited number of validators doesn't have fundamental impact on Celestia	Shared Security

Layer 0 Protocols - Polkadot

There are multiple layer 0 approaches to building out scalable blockchain frameworks. The two most well-known are Cosmos and Polkadot. We will first cover Polkadot and then cover Cosmos.

Polkadot differs from Cosmos in its security model. If you want to connect a Parachain to the Relay chain, there is a limited number of execution slots on the Polkadot Relay chain. This limited number of slots is due to its security model - they all share the same level of security and the Relay chain is responsible for all of the execution. The shared security of the model forces all of the Polkadot main chain validators to validate all of the parachains, which can become expensive. Hence, we are seeing auctions for these parachain slots because there is only a limited number of parachains that can connect to it. If it becomes successful, parachains will become far too expensive which is a problem given it limits new experimentation and limits the devs from capturing value for themselves. In short, **the enshrined shared security of Polkadot proves to be a bottleneck for true scalability.**

Cosmos Security Assumptions

Cosmos takes a different approach, where the individual blockchain decides on its own level of security. Polkadot's design would be akin to Cosmos forcing all Zones to natively integrate Interchain Security by default. However, this is not the case and Cosmos takes a horizontal approach to scalability by making it more modular. In short, Cosmos gives the chain the option to opt into a shared security model (via Interchain Security) whereas Polkadot comes with Shared Security by default. Of course, missing shared security comes with tradeoffs. Some of the main trade-offs include varying levels of security across defi-focused Zones. This varying level of security becomes an issue when multiple Zones are interoperable and a given Zone cannot secure the TVL on chain. When building out a composable financial ecosystem, you want to ensure that the funds are safe. Although this is not an issue for all chains, it will be a barrier to entry for the long tail of chains that cannot meet the security requirements to secure their own chain - this is where Celestia comes to play. The improved security of Celestia's light clients means that chains built on Celestia have much stronger security guarantees for interoperability standards such as IBC. Thus, if successful, it will allow application-specific chains to have shared security while remaining sovereign blockchains.

Cosmos Framework

To better understand the modular stack Celestia is pushing forward, we will reference the Cosmos framework that they have already laid out. Cosmos itself is not a blockchain but rather a framework for building out independent and sovereign blockchains. They have taken a modular approach where they build out the essential tooling that chains will need. Some of the key innovations they have built out include some of the following:

- Tendermint for chains to easier deploy PoS
- The Cosmos SDK to easily build out applications
- IBC for generalized interoperability between Zones
- Interchain Security as a variation of opt-in Shared Security

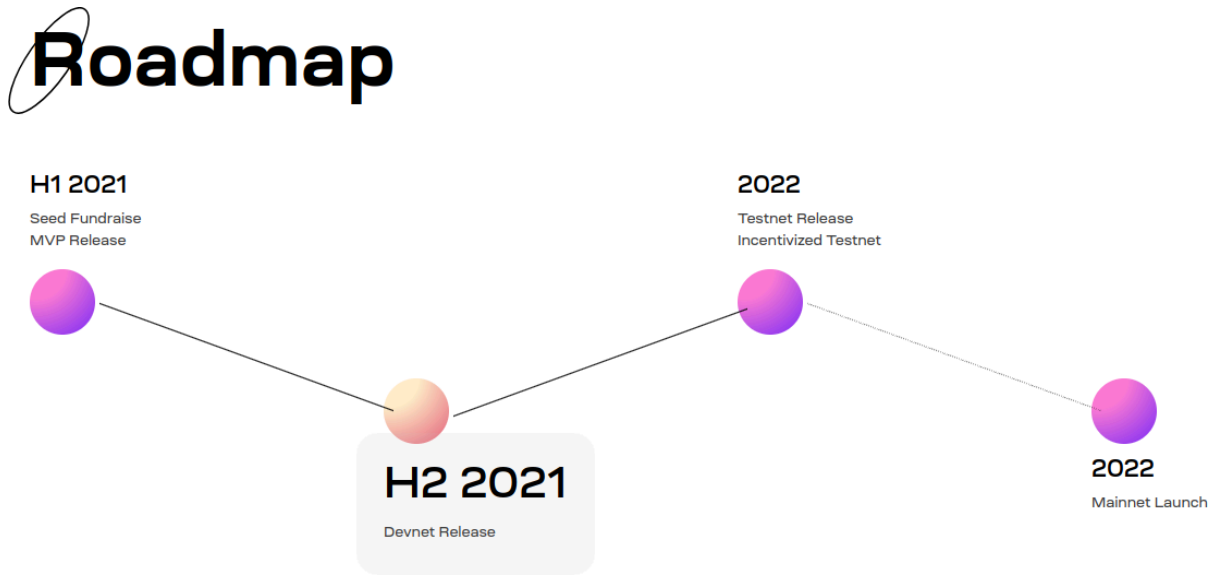
The key difference between Cosmos and any other layer 1 or 0, is that they give each chain an option to adopt or modify the key tools they have built out. In short, you can say that Cosmos defeats the notion of the fat protocol thesis and enshrines itself to be the key infrastructure for sovereign and independent blockchains to coexist in a composable way. However, in order for Cosmos to realize its true vision of millions of sovereign blockchains, Celestia will provide the necessary DA layer for Zones to compose on top of.

Celestia Transaction Fees

Transaction fees will be used to only price the data that blockchains/rollups choose to post to Celestia. This design is a lot simpler and cheaper because they are only pricing a single resource and not executing transactions (which becomes expensive). Again, data availability on Ethereum is competing for transaction fees with other smart contracts currently using the network. Celestia simplifies this bottleneck by only ordering the data and checking for data availability.

Celestia's blockspace will inherently provide greater capacity and lower fees for blockchains to compose on top of it. Fees will initially be dependent on the size of the data being published to Celestia. They also plan on implementing an EIP-1559 fee burn mechanism to offset protocol inflation.

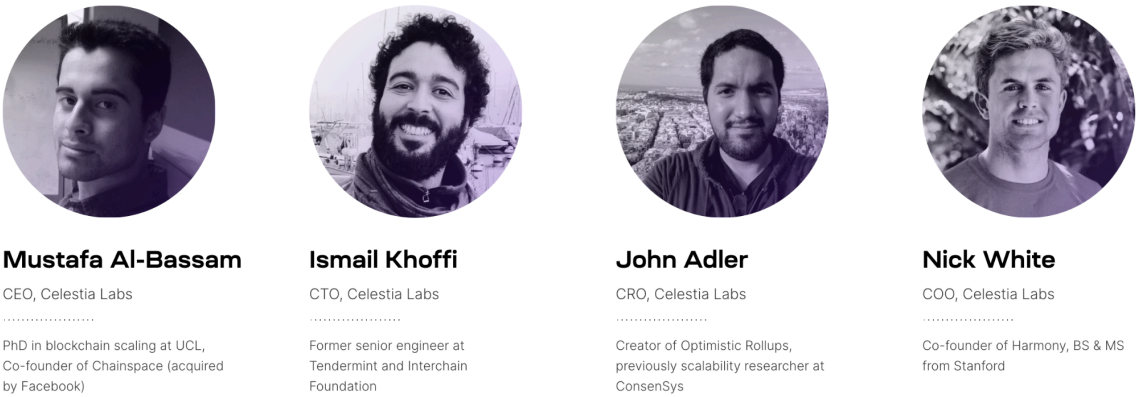
Roadmap



Source: Celestia

The Celestia Team

The Celestia team has deep experience building and scaling blockchains and have come from projects from Ethereum, Cosmos, and Harmony. Mustafa Al-Bassam was a co-founder of Chainspace and leads a team of industry heavyweights. John Adler for instance, was the first to write about how roll ups could work in 2018. Ismail Khoffi and Nick White both too have stellar technical experience, with Nick White being a co-founder of Harmony with BS & MS from Stanford, and Ismail as a former senior engineer at Tendermint & Interchain Foundation.



Source: Celestia

Celestia Devnet Release

Celestia's Devnet was launched on December 14th, 2021. The Devnet features three core components: Celestia-node, Celestia-app and Optimint.

How to get involved

If you are interested in a running a node, whitelisting is now open on Celestia for participation:

Testnet whitelist waitlist is now open here.
Developer's Beta waitlist is now open here.

Celestia X Evmos Rollup Partnerships: Cevmos (short for Celestia/Evmosos/Cosmos)

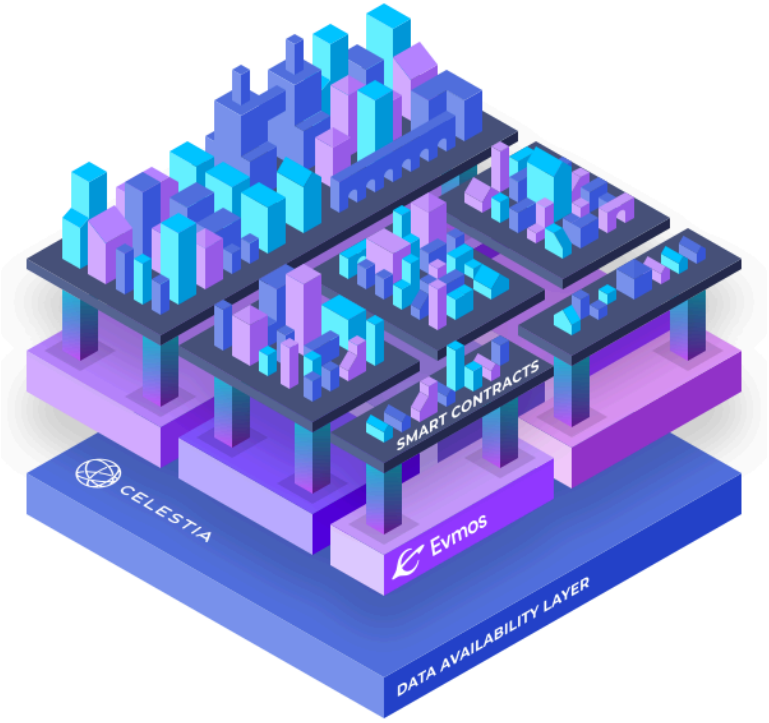
What is Evmos

Previously known as Ethermint, Evmos brings the EVM to Cosmos. It was launched as an application-agnostic chain that is EVM-compatible by default and interoperable with the greater Cosmos ecosystem via IBC. Evmos aims to be the EVM Hub of Cosmos and reduces friction with the deployment of smart contracts, and communication, within the Cosmos ecosystem. Their vision of the future is one of highly secure, fast finality, EVM-based chains that provide interoperability and greater composability for smart contracts in the Interchain. In recent developments, Evmos has just deployed its ERC-20 module that allows the permissionless smart contracts on Evmos to use their generated tokens across Cosmos applications. There was no prior way of using these tokens across the Interchain - seamlessly converting between ERC-20 tokens and Cosmos native tokens and vice versa, without bridging.

On Ethereum, fast finality has been gaining popularity this year. Vitalik Buterin has also written about the need for a Tendermint-like consensus model for the Ethereum beacon chain here. Consequently, rollup solutions have also been picking up traction. These tie in with the Evmos vision, and with Evmos's partnership with Celestia, Celestia is poised to be used as a DA layer for non-Ethereum rollups. The reason why we highlight this partnership is because it disproves the notion that rollups will only exist on Ethereum. Rollups will be widely used and many of them will need a shared DA layer such as Celestia.

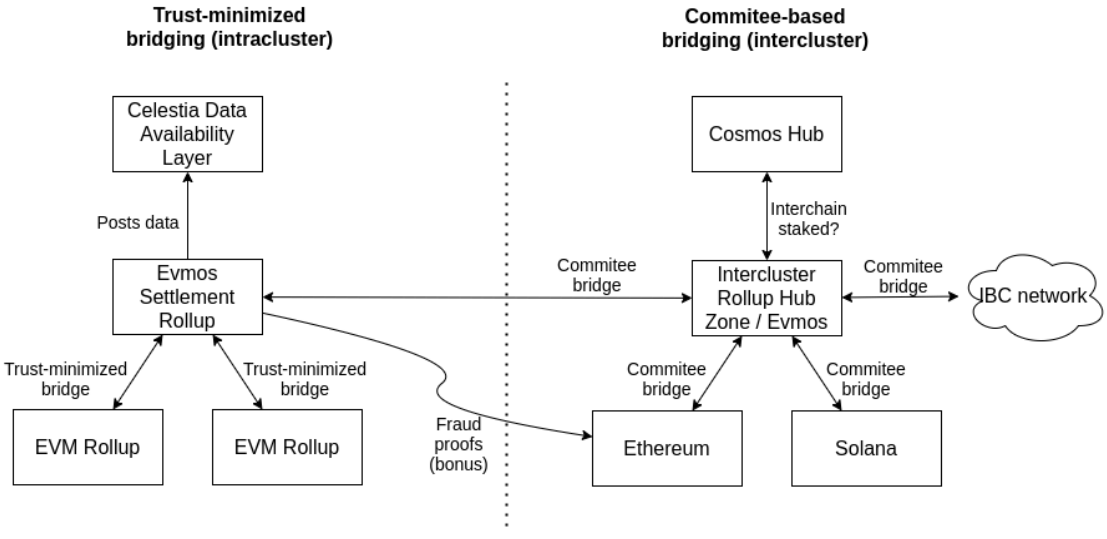
Celestia X Evmos

Celestia has partnered with Evmos to build an optimal settlement layer for EVM rollups. With rollups, transactions are done off-chain with reduced transaction costs. Transaction data are then broadcasted back on-chain onto the consensus and data availability layer for security. As such, rollups have the ability to vastly improve scalability of their respective L1s.



Source: Celestia

Celestia's partnership with Evmos results in an open, modular stack for EVM-based applications. The stack uses Celestia rollups, and an optimized settlement chain for rollups that is based on Evmos. This optimized settlement chain will be implemented by using Optimistic Tendermint instead of the Tendermint Core consensus engine used on Cosmos chains.



Source: Celestia

The settlement chain based on Evmos uses Celestia as the DA layer to provide a fully EVM-equivalent stack for interoperable smart contracts on Cosmos and the EVM ecosystem. A benefit of the Cevmos stack is that the chain will be optimized for roll ups unlike Ethereum. As a result, rollups won't have to compete for gas with non-rollup transactions. The $EVMOS token will be used for security and gas, as new chains built with Cevmos will be connected via IBC to the Evmos Hub.

Closing Thoughts

Celestia is tackling a very prevalent problem across all blockchains - it is prohibitive for developers to scale exponentially on current L1s. On Ethereum, transaction throughput is constrained by gas fees. On Polkadot's Relay Chain to Parachain model, the number of execution chains is limited by design and parachains, if successful, will be wildly expensive. On Cosmos, opt-in validator costs are expensive for interchain security integration for reasons similar to Polkadot. These limitations are due to the coupling of execution, consensus and data availability into a single layer. By decoupling them, Celestia is able to provide permissionless, low-barrier deployments with an infinite number of execution slots. This unlocks the ability to include long tail chains, and given the scalability advantages of DA layers, they may well become the "end game" described by Vitalik Buterin.

Value Accrual in a Modular Stack

Given a modular blockchain stack becomes prevalent, which layer should accrue the most value? We are unsure as of now, but let's dive into why the different levels of the stack may have varying network effects and go from there. The 3 layers in the stack include the Execution layer, Settlement layer and the Data Availability layer. You can have many execution layers such as rollups, volitions, and volidiums on a single Settlement layer. The Settlement layer is simply an execution chain that these rollups have a trust minimized bridge with and it is the method of communication across roll ups using the same Settlement layer. Hence, it is another execution layer specialized in verifying rollup state and custody of assets which poses a limit to what the settlement layer can process.

Data availability layers such as Celestia have the most scalability across any other layers of the stack, which will bring the strongest network effects given its shared security. Although this is just speculation, this horizontal scaling ability of DA layers leads us to believe that the distribution of layers across the stack is as follows: **Execution layer > Settlement Layer > DA Layer**. If this type

of distribution plays out, then it would lead us to believe that lower down the stack you go, the higher the network effects and the higher the value capture based on the scale. In short, the Settlement layer will have network effects but DA layers can have even more network effects through its shared security and horizontal scaling.