CELESTIA

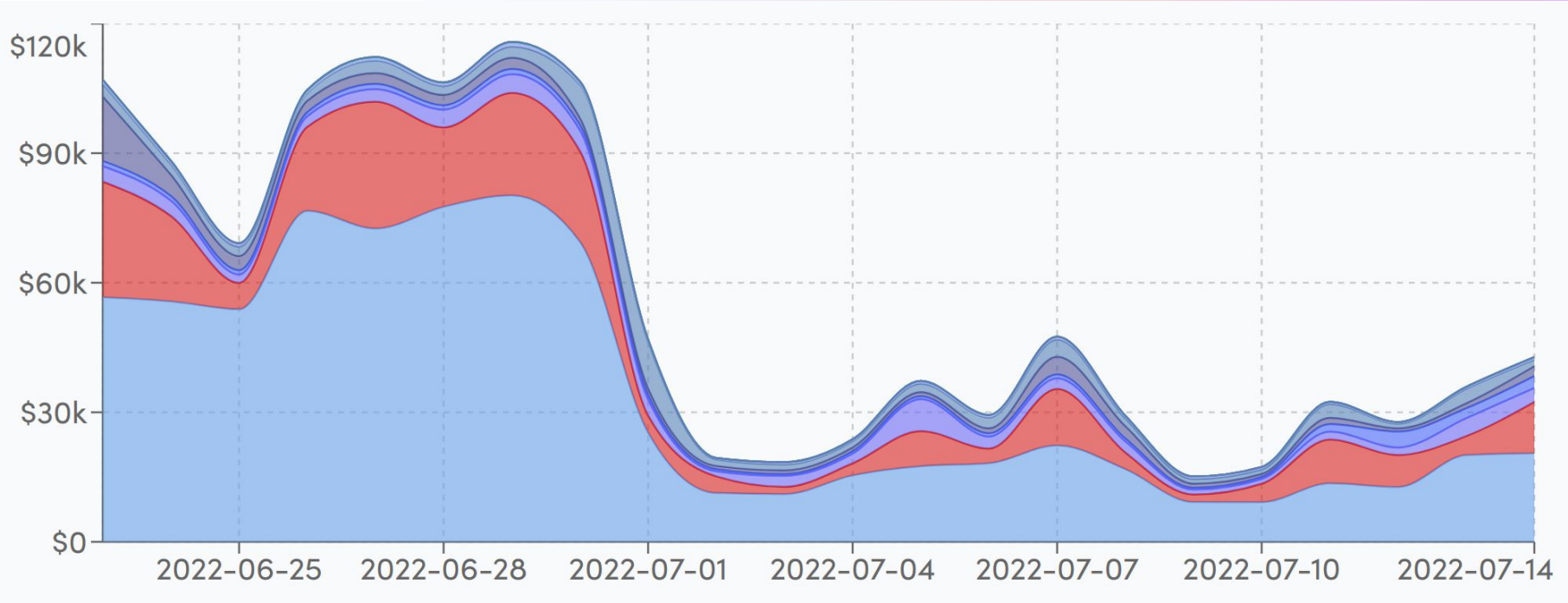
The First Modular Blockchain Network

twitter: **@evansforbes (zkFART)**
github: **@evan-forbes**

Celestiums: Scaling Ethereum L2s by using Celestia for Data Availability

The problem

Rollup transactions and security can be expensive.



Picture from L2Fees.info

Why is calldata so expensive?

- All full nodes have to download calldata to maintain full security
- Competing for gas against million dollar arbitrages, NFT mints, etc

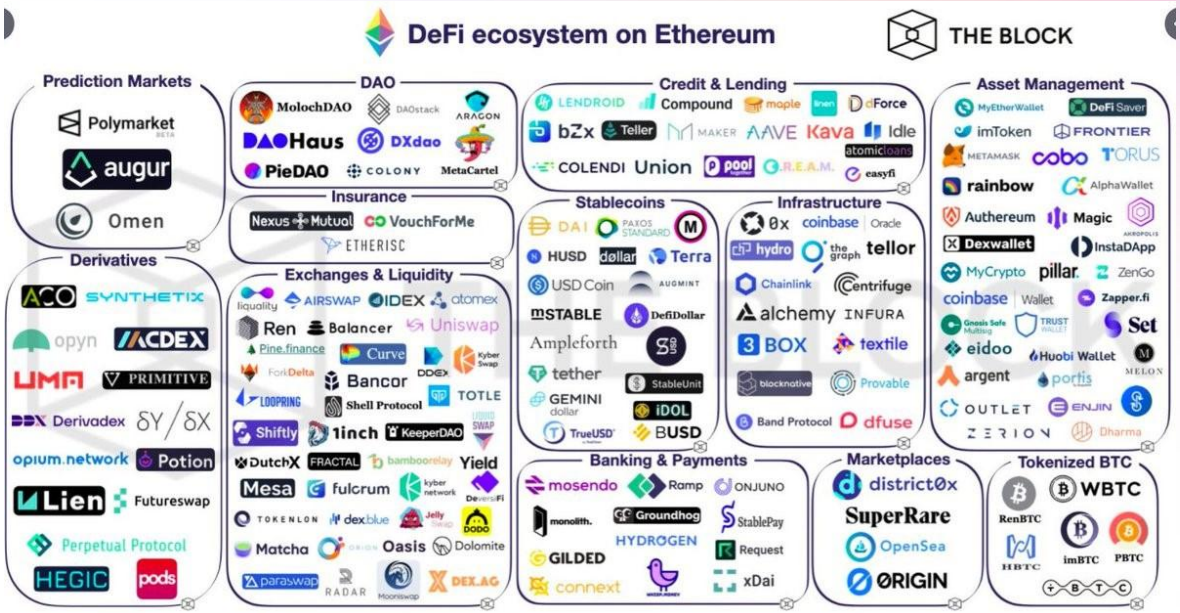


Image credit: The Block

Celestiums: the problem

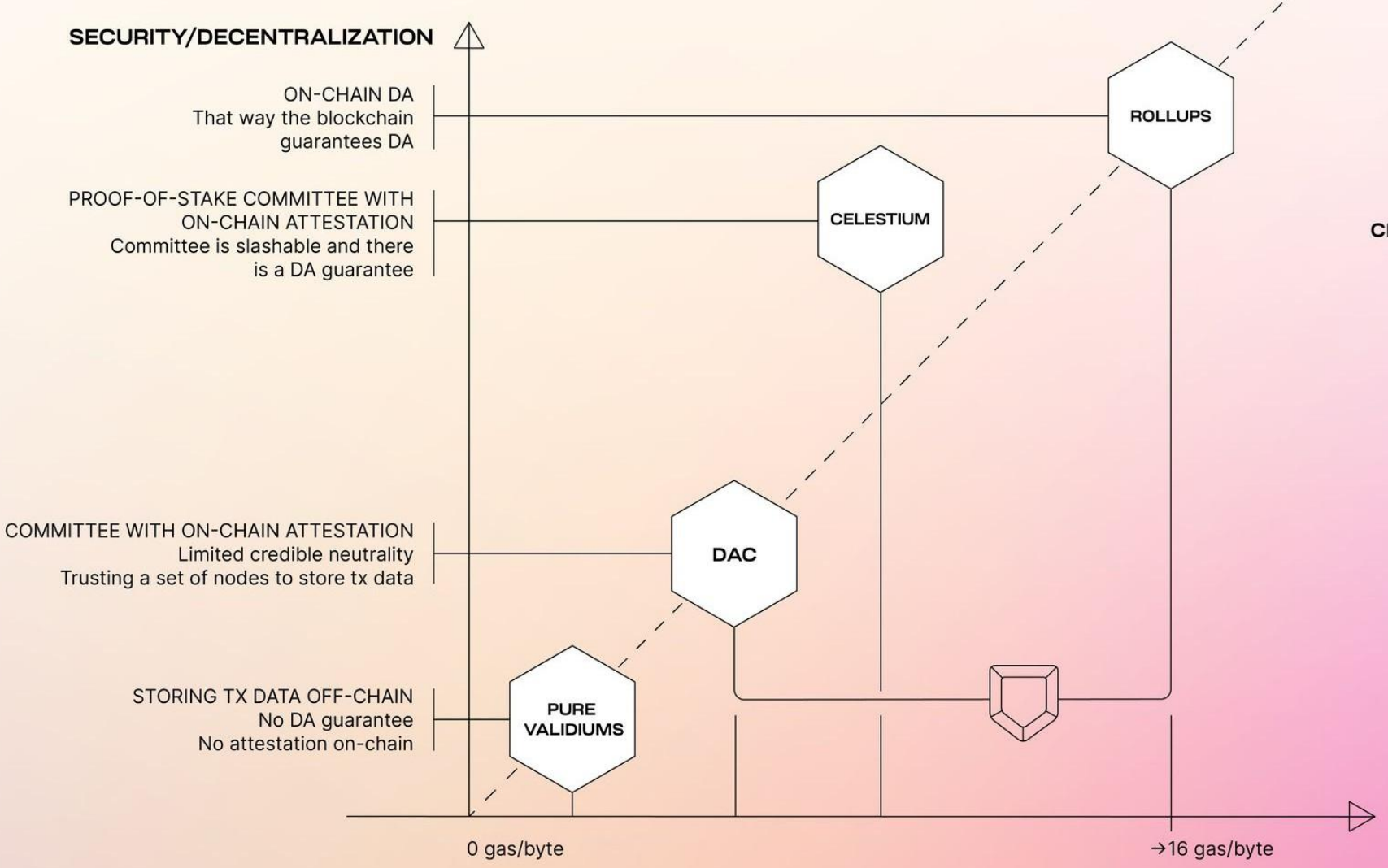
What do you do if your project can’t afford these costs?

Trade Offs: Rollups vs DACs

- **Rollups: Posting L2 data directly on Ethereum**
  - Secure
  - Can be expensive
  - Seamless “out of the box” integration
- **DACs: Using a data availability committee**
  - Insecure
  - Cheaper and more predictable price
  - Have to build yourself

Celestiums!

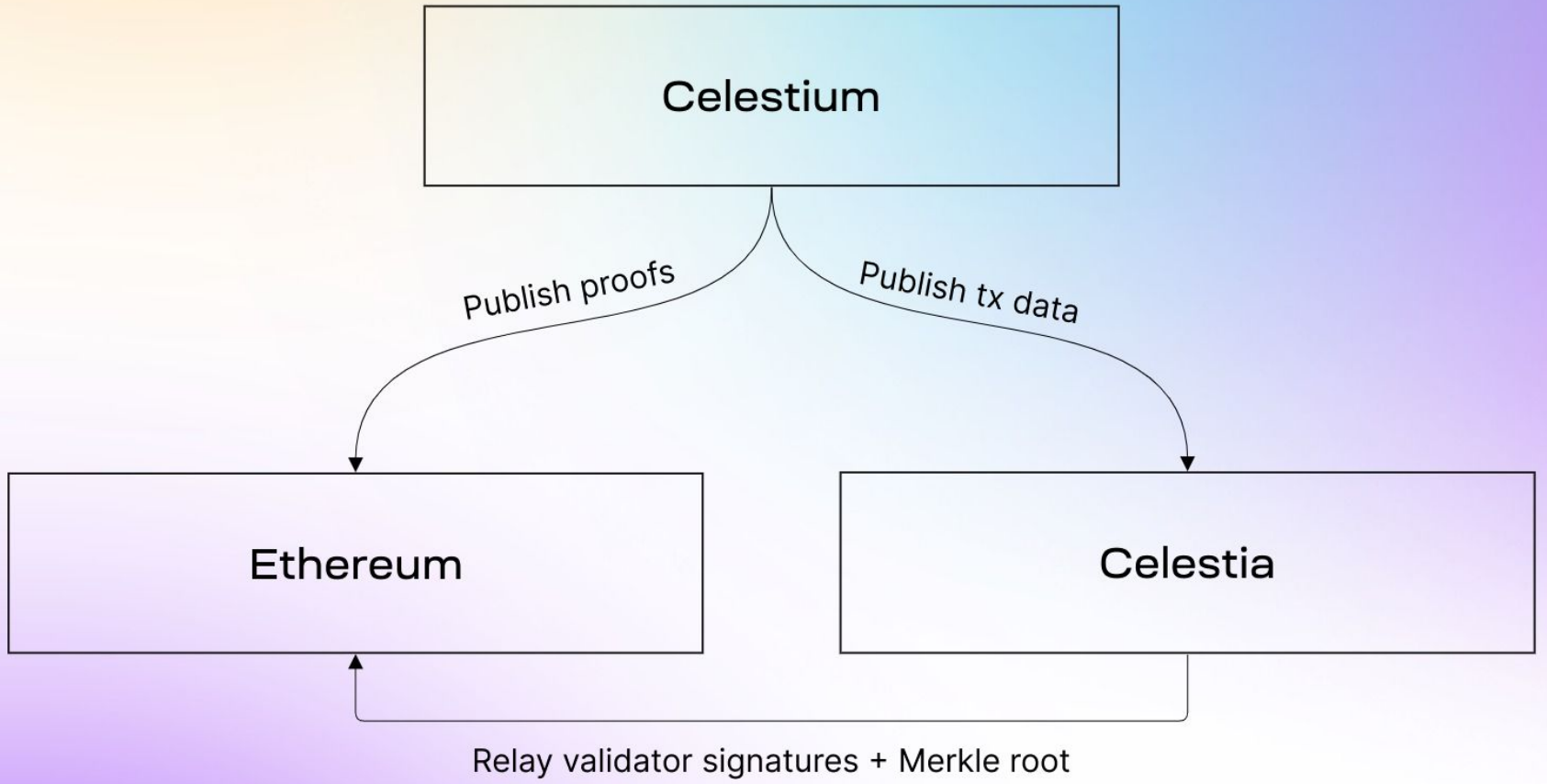
- **Celestiums: Posting L2 data on a more scalable data availability layer**
  - Great security
  - Significantly cheaper price
  - It’s built for you


SECURITY/DECENTRALIZATION
ON-CHAIN DA
That way the blockchain
guarantees DA
PROOF-OF-STAKE COMMITTEE WITH
ON-CHAIN ATTESTATION
Committee is slashable and there
is a DA guarantee
COMMITTEE WITH ON-CHAIN ATTESTATION
Limited credible neutrality
Trusting a set of nodes to store tx data
STORING TX DATA OFF-CHAIN
No DA guarantee
No attestation on-chain
ROLLUPS
CELESTIUM
DAC
PURE
VALIDIUMS
0 gas/byte
→16 gas/byte

More in depth cost analysis info at our blog

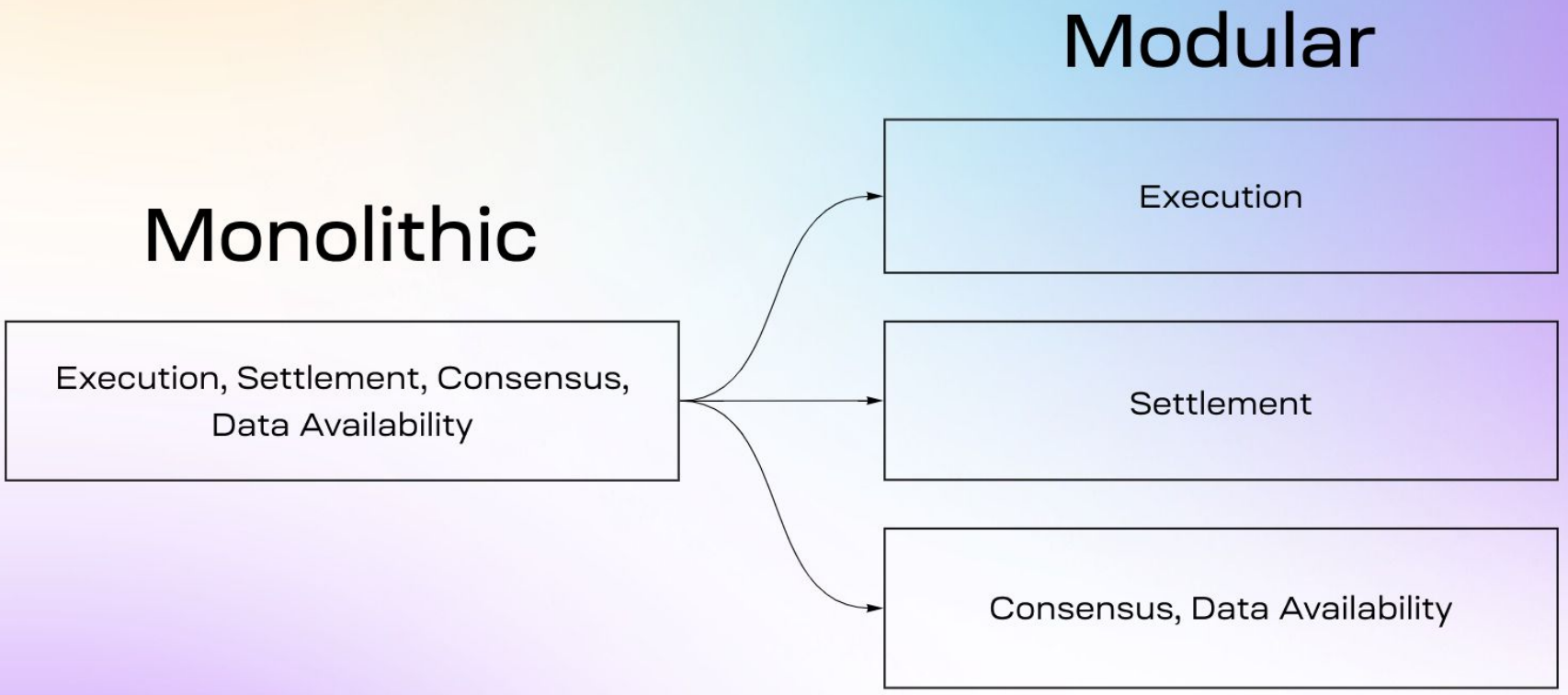
https://blog.celestia.org/ethereum-off-chain-data-availability-landscape/

Shout out Aditi and John


Celestium
Publish proofs
Publish tx data
Ethereum
Celestia
Relay validator signatures + Merkle root

Why use Celestia for Data Availability?

No execution environment!


Modular
Monolithic
Execution, Settlement, Consensus, Data Availability
Execution
Settlement
Consensus, Data Availability

No execution environment!

Spend money on bandwidth, not running a giant state db

No execution environment!

Don’t compete with Million Dollar arbs on the L1

Basically Celestia's entire proprietary state machine 👀 🫵

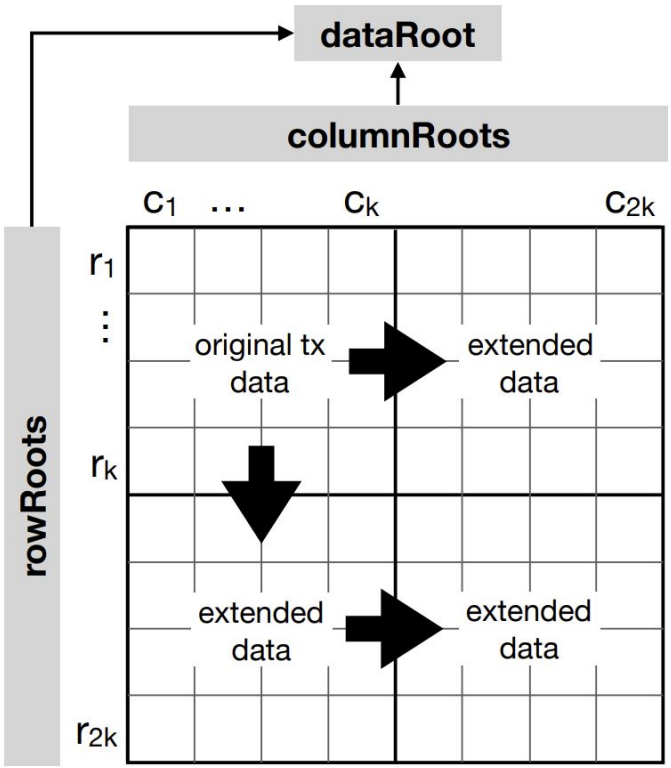
```go
func (k Keeper) PayForData(goCtx context.Context, msg *types.MsgPayForData) (*types.MsgPayForDataResponse, error) {
	ctx := sdk.UnwrapSDKContext(goCtx)

	ctx.GasMeter().ConsumeGas(msg.MessageSize, payForDataGasDescriptor)

	ctx.EventManager().EmitEvent(
		types.NewPayForDataEvent(sdk.AccAddress(msg.Signer).String(), msg.GetMessageSize()),
	)

	return &types.MsgPayForDataResponse{}, nil
}
```

Intro to Celestia: Trust Minimized Light Clients



- LCs convince themselves that data is available
- LCs listen for fraud proofs
- Essentially the same security as a full node!
- Contribute security to the network

Image credit:

Mustafa Al-Bassam, Alberto Sonnino, and Vitalik Buterin. Fraud and Data Availability Proofs https://arxiv.org/pdf/1809.09044.pdf

Data downloaded by LCs increases O(squareroot(n)) with block size

More light clients means larger blocks are still safe

Sample the data yourself. 💪

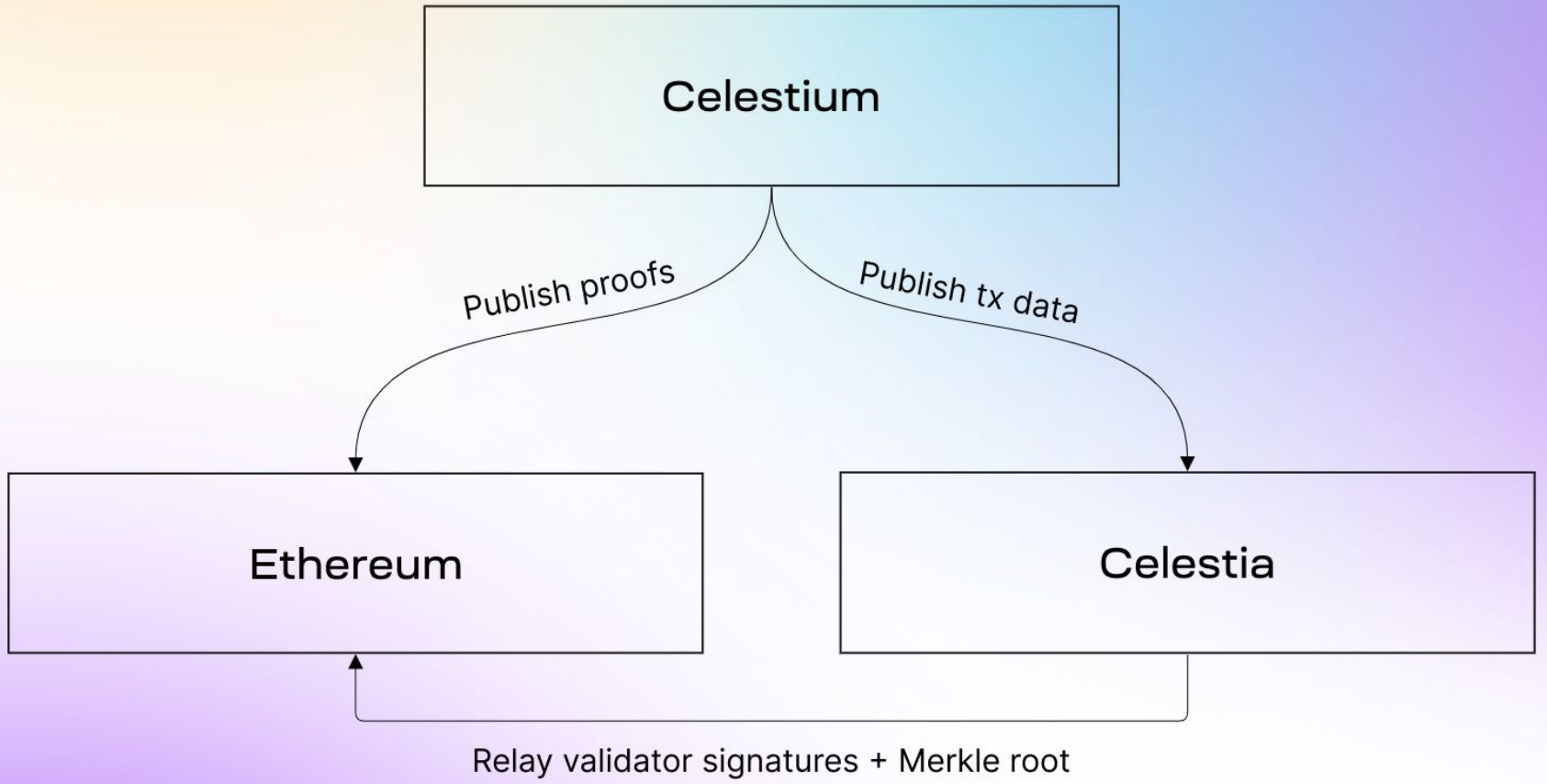
or

Download the only data you want directly

Celestia: other cool things

- First modular blockchain
- Sovereign Rollups
  - Scalability and trust assumptions of rollup && the sovereignty of a cosmos chain
- No need to spin up a validator set, just use an API
- Mustafa Al-Bassam

How can we prove that some data was posted to Celestia via the EVM?


Celestium
Publish proofs
Publish tx data
Ethereum
Celestia
Relay validator signatures + Merkle root

THE QUANTUM GRAVITY BRIDGE

QGB: Light Client Relay

Trusts the validator set

Doesn’t verify state

Does verify the commit

QGB: Data availability committee

IBC: Multisig bridge

Comparing IBC: Multisig Bridge

IBC ✅ ✨	Multisig Bridge 🛑 💀
Light client based	Completely trusting a few private keys
Crypto-economically secure	LOL (mostly reputation)
Permissionless	Permissioned
Same security as the trusted chain	The weakest link between two chains
Safely and securely transfers billions ($USD) of value	Loses billions per year in “hacks”

Comparing QGB vs DAC

QGB ✅ ✨	DAC 🛑 💀
Light client based	Completely trusting a few private keys
Crypto-economically secure	LOL (mostly reputation)
Permissionless	Permissioned
Same security as the trusted chain	The weakest link between two chains
TBD	TBD

QGB Optimizations

- Batching
  - Data Commitments
  - Validator set updates
- Requires only a single relayer to be live
- All block data on Celestia is committed to!
- Improved relayer gas usage
  - Doesn’t increase when the Celestia block size increases
- One to many design

QGB Slashing

Types:

- Liveness
- Incorrect signatures
- Equivocation

How to use the QGB: Proving inclusion

```
/// @dev see "./IDAOracle.sol"
function verifyAttestation(
    uint256 _tupleRootNonce,
    DataRootTuple memory _tuple,
    BinaryMerkleProof memory _proof
) external view override returns (bool) {
    // Tuple must have been committed before.
    if (_tupleRootNonce > state_eventNonce) {
        return false;
    }

    // Load the tuple root at the given index from storage.
    bytes32 root = state_dataRootTupleRoots[_tupleRootNonce];

    // Verify the proof.
    bool isProofValid = BinaryMerkleTree.verify(root, _proof, abi.encode(_tuple));

    return isProofValid;
}
```

How to use the QGB: Posting Data

```
/// @dev see "./IDAOracle.sol"
function verifyAttestation(
    uint256 _tupleRootNonce,
    DataRootTuple memory _tuple,
    BinaryMerkleProof memory _proof
) external view override returns (bool) {
    // Tuple must have been committed before.
    if (_tupleRootNonce > state_eventNonce) {
        return false;
    }

    // Load the tuple root at the given index from storage.
    bytes32 root = state_dataRootTupleRoots[_tupleRootNonce];

    // Verify the proof.
    bool isProofValid = BinaryMerkleTree.verify(root, _proof, abi.encode(_tuple));

    return isProofValid;
}
```

Conclusion

Celestia is a *really* scalable, trust minimized, and modular data availability layer 🧱

Now we can use Celestia for Data availability on Ethereum in a secure, cheap, and easy way 🤓

We finally have a new data availability solution for L2s other than DACs 🙌

Coming to a testnet near you!!

We’re hiring!

- Cosmos-sdk engineers
- Golang engineers
- Database engineers
- Devops engineers

Q/A

CELESTIA

https://blog.celestia.org