Week 6 Research Track - Sharding & DAS

Guest speaker

- Dankrad Feist, Ethereum Research

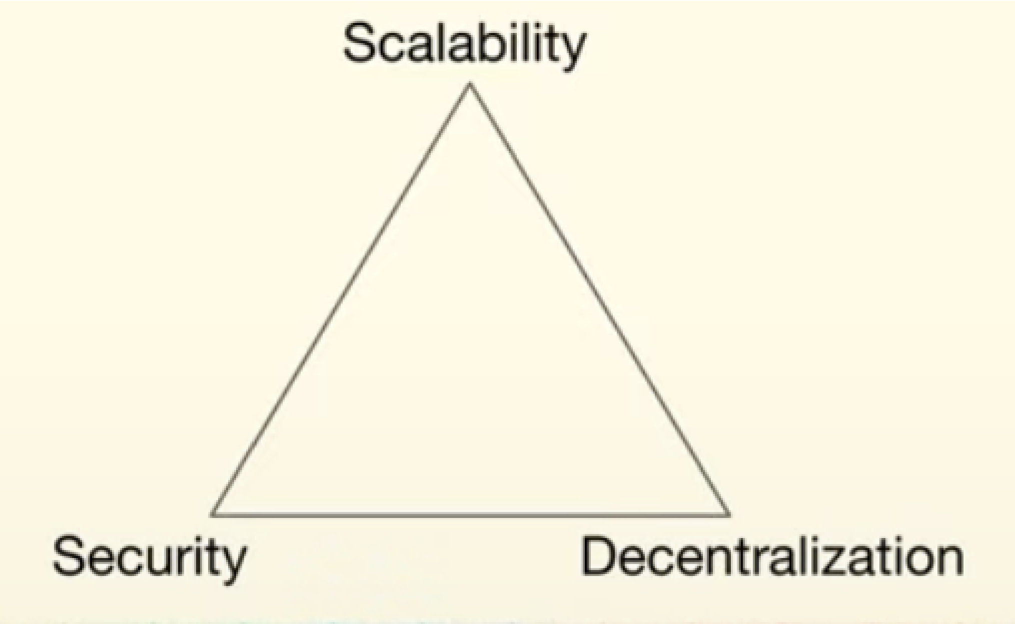
Summary notes

- Edited by Chloe Zhu
- Online version: https://ab9jvcjkej.feishu.cn/docx/Ipdvd6RQqoDrskx8dpscXUO4nIf

Blockchain scalability

The Trilemma problem:

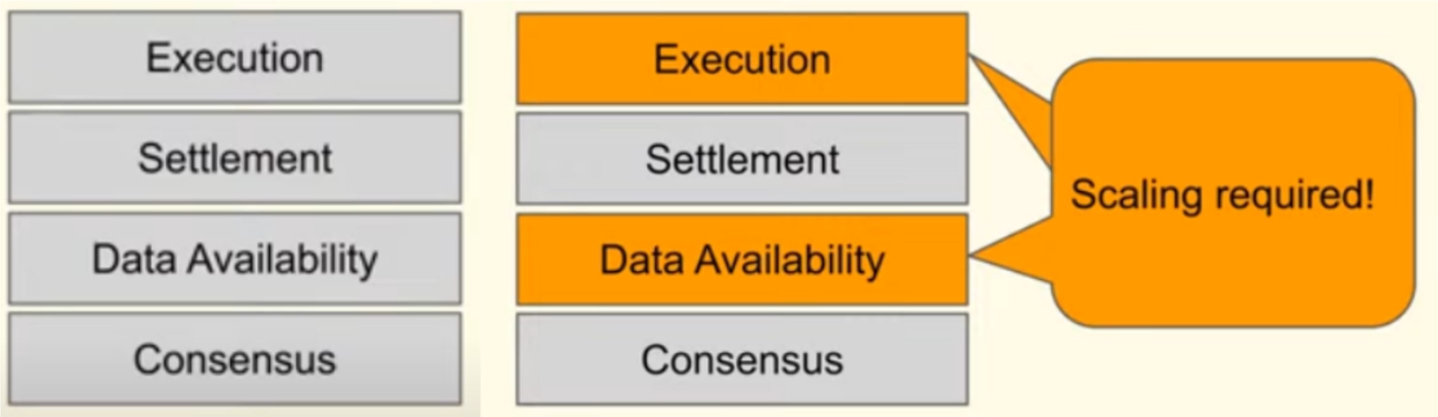
- It's difficult to design a blockchain that provides scalability, security & decentralization. However, it doesn't say that it's impossible.



Decentralization

- Two limitation factors
  - Execution: constrained by the current computation hardware
  - Bandwidth: constrained by the internet connection

Blockchain stack



- Execution:
    - Decide which transactions are valid and execute
    - Dependent on the number of transactions, which needs scaling
- Settlement:
    - Root layer of all
    - Independent of the number of transactions
- Data availability (DA):
    - Make sure everything that's part of the blockchain is available to everyone
    - Dependent on the number of transactions, which needs scaling
- Consensus:
    - Define the canonical chain
    - Independent of the number of transactions

Scalling attempts

- Execution: Rollups with fraud proofs or validity proofs
- DA: Data availability sampling (DAS)

The DA problem

Definition of the DA problem

- DA means that no network participant, incl. a colluding supermajority of full nodes, has the ability to withhold data
- Current blockchains:
    - All full nodes download all the data (impossible to withhold data)
- But how to make this scalable?
    - Scalable means that the work required should be less than downloading the full blocks

    - E.g. Constant or a logarithmic amount of work

What DA is about

- DA = Assurance data was not withheld = Assuance data was published

What DA is not about

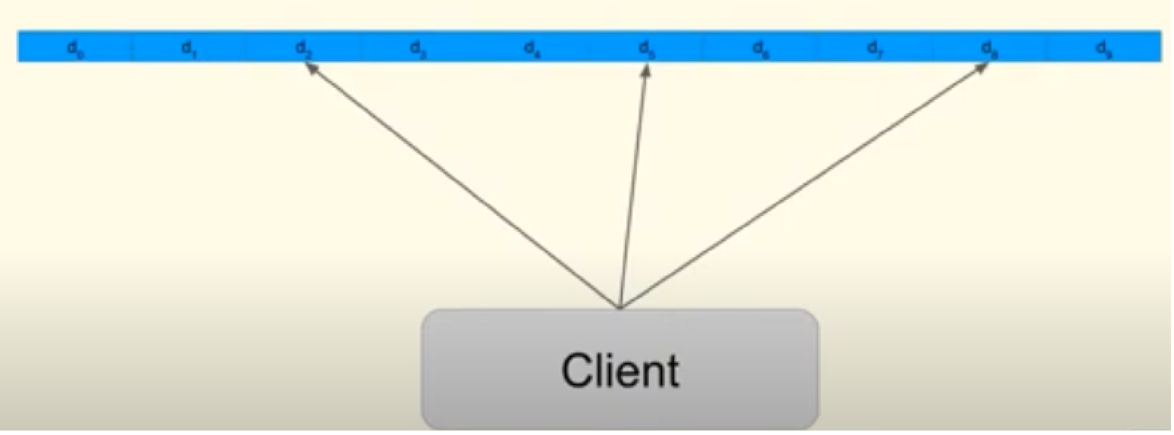
- DA =/ Data storage =/ Continued availability

Is the DA problem really that important?

- Two scalable execution options
    - OP rollups using fraud proofs
        - Any missing data could be fraudulent. e.g. A state change printing 1 trillion ETH
        - All data needs to be available unconditionally or fraud proofs cannot be constructed.
    - ZK rollups using validity proofs
        - Missing data can contain an update to your account
        - If you don't know how to access your account (missing witness), you will lose access.

Data Availability Sampling

The idea



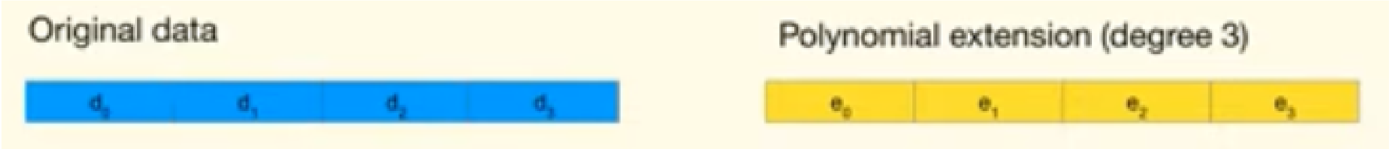
- We could chunk the data into pieces, and instead of downloading all the data, we could just select some random bits of the pieces and download them. And at the same time, we hope that all of the data that we request are available

The problem

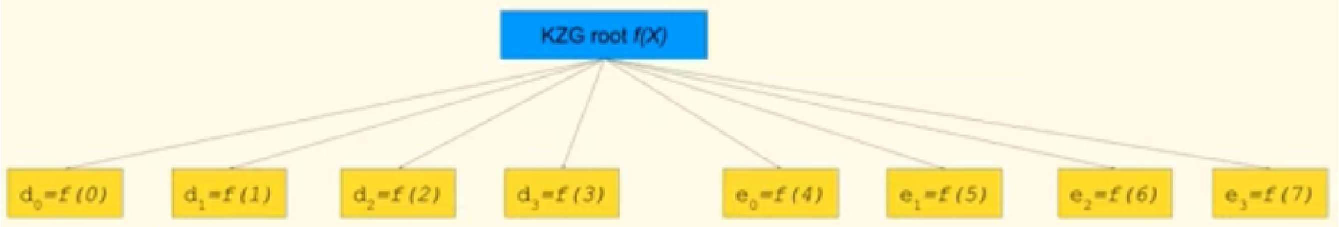
- Even if a tiny bit of data missing could be a huge problem. Thus, random sampling is not enough.

- Need a way to amplify the method, so that with a small number of samples we can already be sure that most of the data is available.

Erasure coding



- Data
  - We don't take the original data, but extend the data using Reed-Solomon code (polynomial interpolation)
  - Eg. At coding rate r=0.5, it means any 50% of the blocks (d0 to e4) are sufficient to reconstruct the whole data
- Sampling becomes efficient
  - Eg. If we query 30 random blocks and they are all available. The probability that more than 50% not available is 2^(-30), which is pretty low.
- Ensuring validity of the encoding
  - Problem: The encoding process could get wrong and the attacker could make up random points.
  - Solution: KZG commitments as DA roots

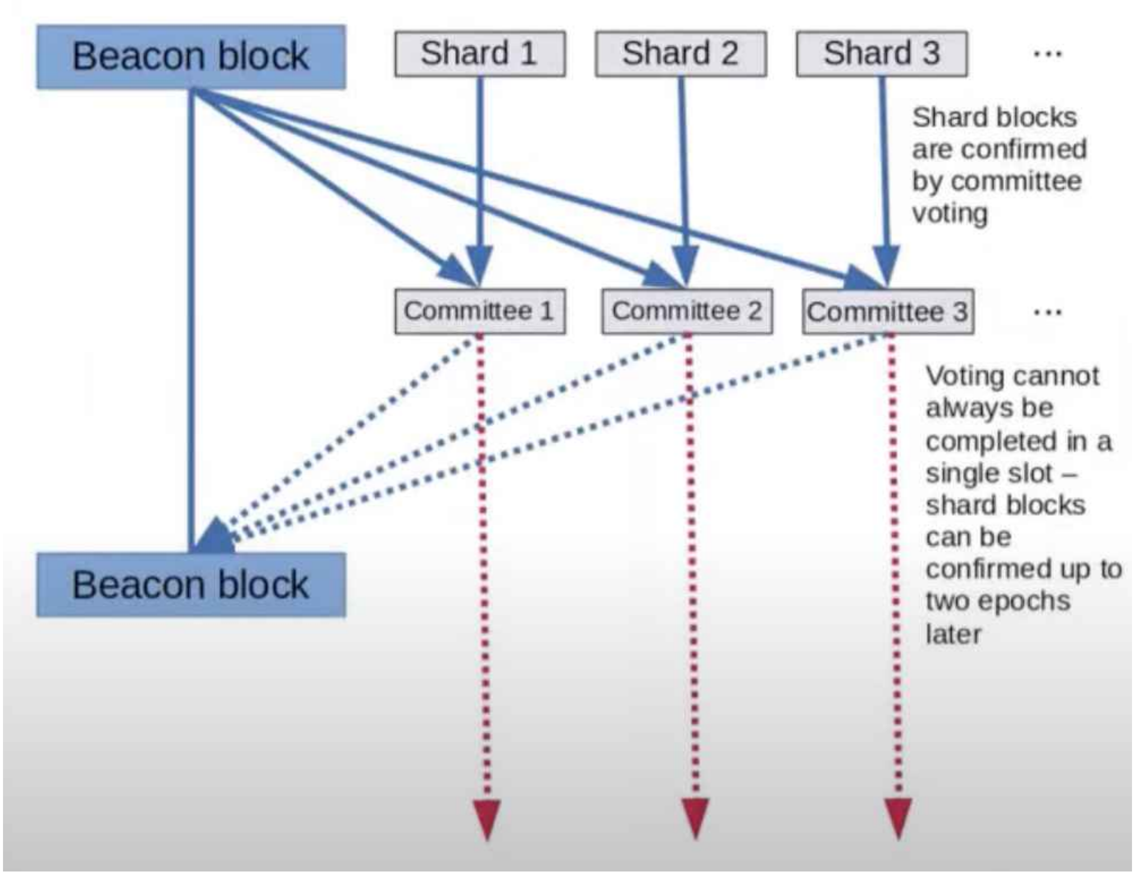
    - All points of KZG commitment are guaranteed to be on the same polynomial. Thus, it's impossible for anyone to commit to an invalid route.

Danksharding

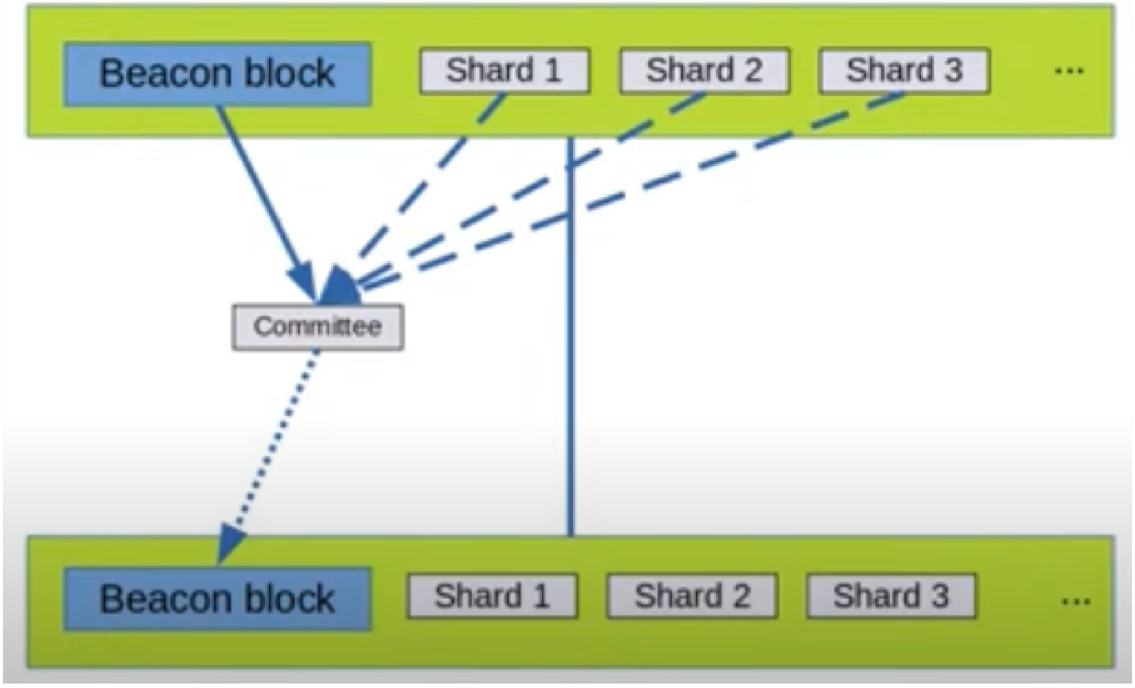
Separate shard proposals

- The original sharding proposal
  - Apart from the beacon chain, there will be 1,024 shards all running in parallel and similar to the beacon chain.
  - Each of the shard will have a proposal and a committee that votes on whether that proposal shown up and what block they proposed.

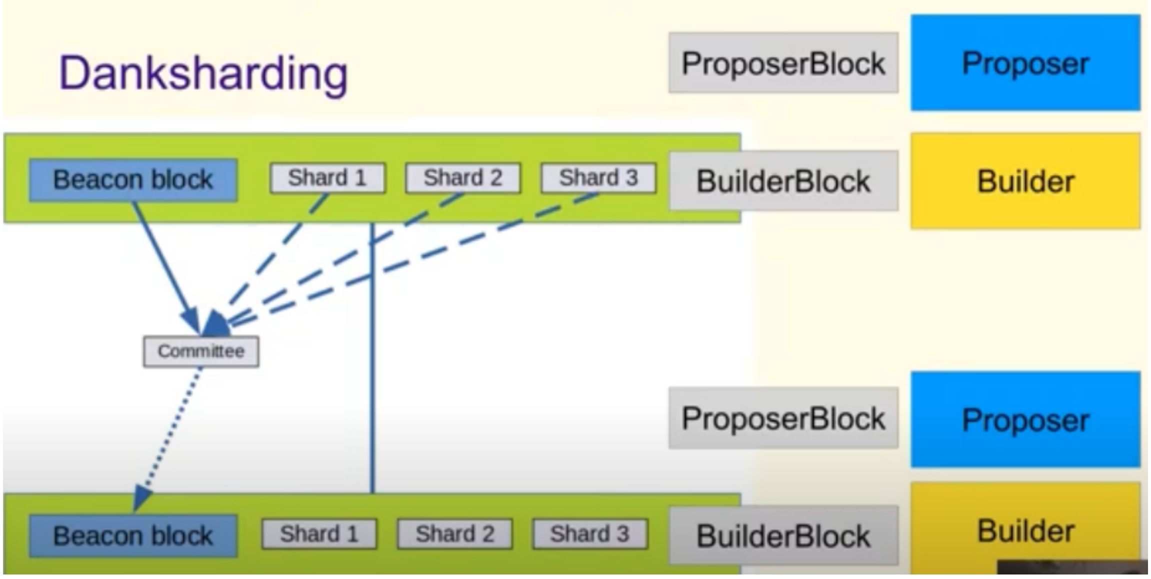
    - The problem: Each proposal can do something bad and each of the committee could be split into eg. 50/50. Then it's hard to decide the validity and may lead to longer time of finality and other issues.


Beacon block
Shard 1
Shard 2
Shard 3
...
Shard blocks are confirmed by committee voting
Committee 1
Committee 2
Committee 3
...
Voting cannot always be completed in a single slot – shard blocks can be confirmed up to two epochs later
Beacon block

- Alternative proposal
    - Let one proposal create all the shards and the beacon block. Then we only have one big committee to check the validity. If the proposer fails to publish the data for one of the shards, we can let the block fail.



   - In the proposer-builder separation (PBS), first there will be a proposer block, where a proposer selects a builder. Then the main block would be built by a builder, who has more hardware resources, and is able to construct the beacon block & shard blocks at the same time.



Danksharding

ProposerBlock

Proposer

Beacon block

Shard 1

Shard 2

Shard 3

BuilderBlock

Builder

Committee

ProposerBlock

Proposer

Beacon block

Shard 1

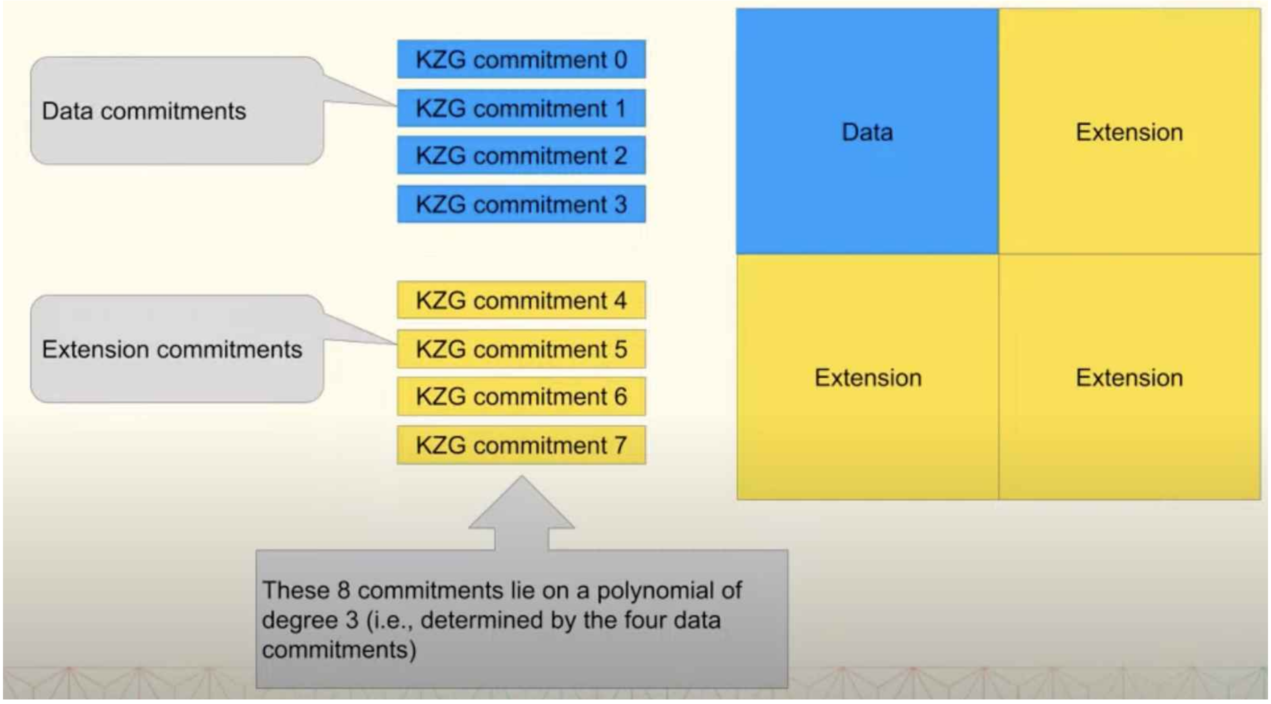
Shard 2

Shard 3

BuilderBlock

Builder

KZG 2-dimension scheme



Data commitments

KZG commitment 0

KZG commitment 1

KZG commitment 2

KZG commitment 3

Extension commitments

KZG commitment 4

KZG commitment 5

KZG commitment 6

KZG commitment 7

These 8 commitments lie on a polynomial of degree 3 (i.e., determined by the four data commitments)

Data

Extension

Extension

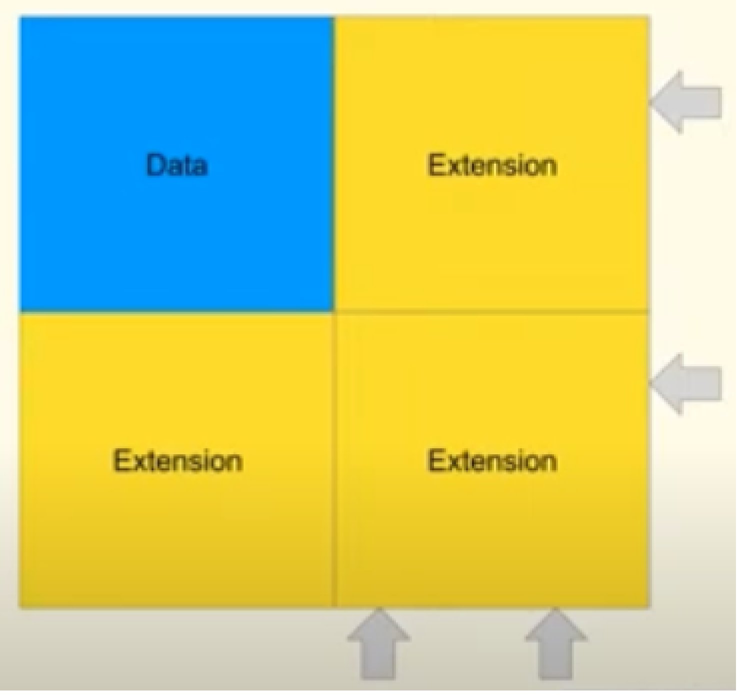
Extension

- Horizontal: KZG commitment for the shards, aka blobs
- Vertical: Each data cell will also be extended vertically
   - Since the KZG commitment is linear, it's possible to compute the additional KZG commitment by adding polynomial extension as well.
- Feature: Can reconstruct the data if the 3 quarters of the whole square are available

Two layers of validation & DAS

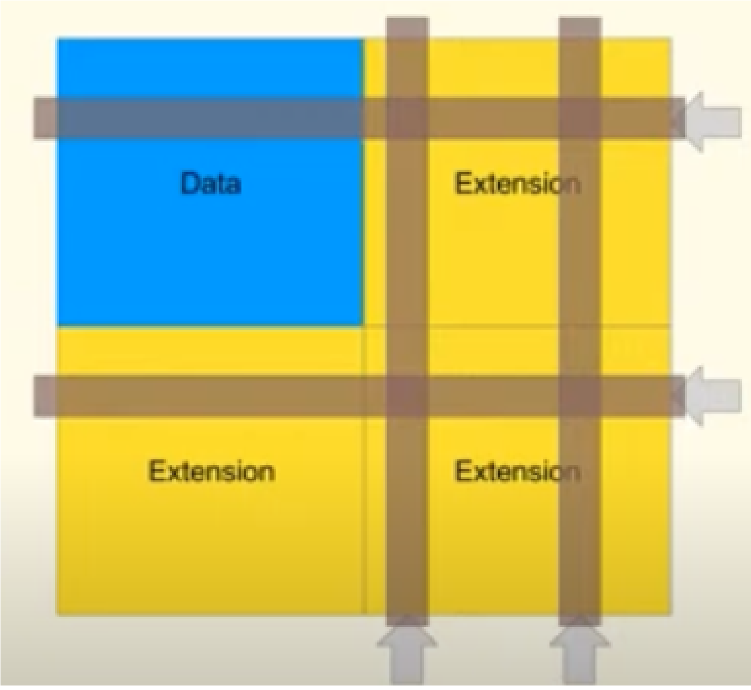
Danksharding honest majority valdiation

- Each validator picks s=2 random rows and columns, to validate that the builder has done the work correctly
- Only attest if the assigned row/ colum are available for the entire epoch
- An unavailable block (<75% available) cannot get more than 2^(-2s) = 1/16 attestations, i.e. the unavailable block will be reorged/ ignored



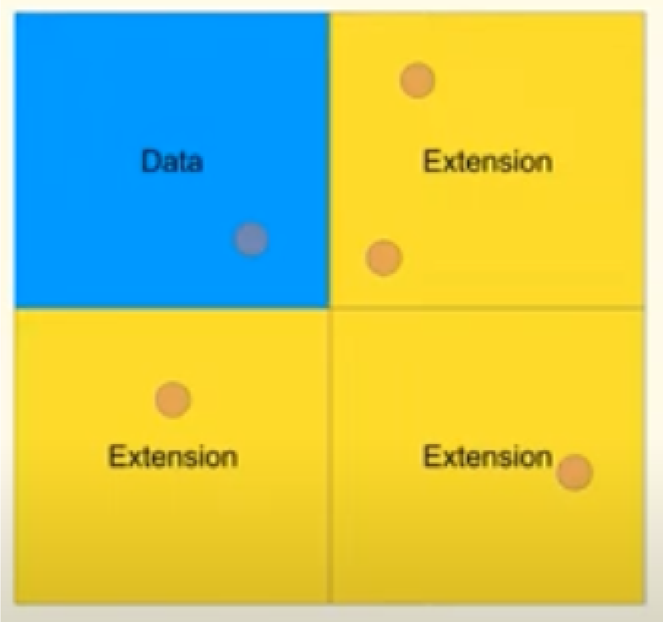
Danksharding reconstruction

- Each validator should reconstruct any incomplete rows/ columns they encounter
  - Since each row/ column individually is a polynomial, they can be individually reconstructed
  - The reconstruction can also be done in a distributed way
- While doing so, they should transfer missing samples to the orthogonal lines
- Each validator can transfer 4 missing samples between rows/ columns
  - Ca. 55,000 online validators guarantee full reconstruction



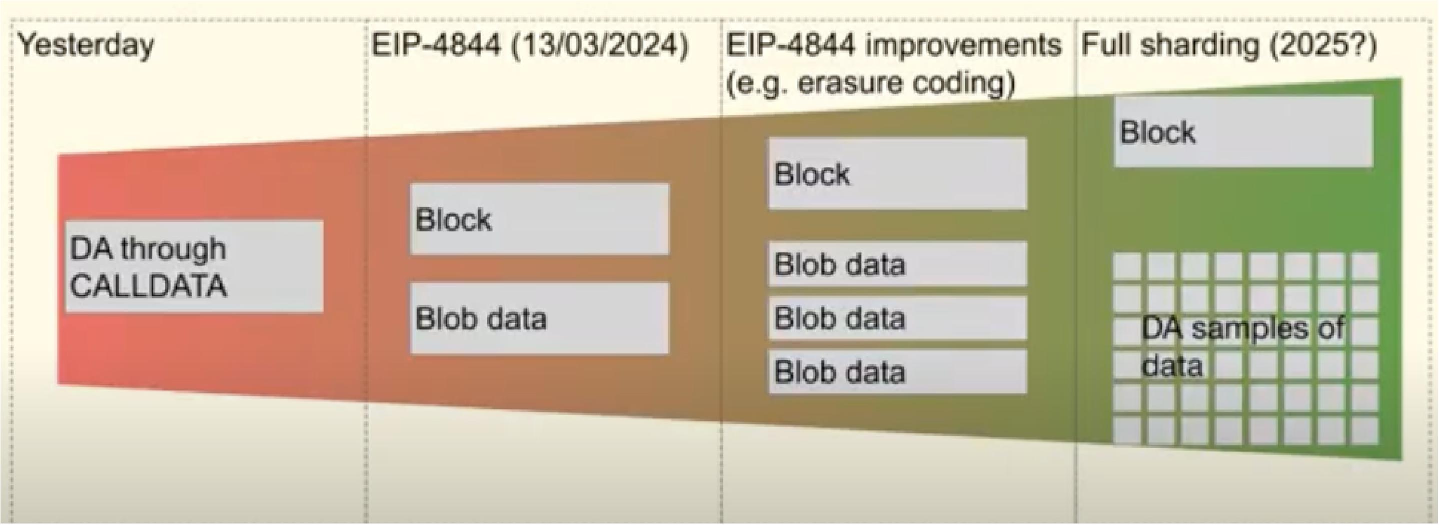
Danksharding DA sampling (malicious majority safety)

- DAS is planned for future upgrade
- Each full node will check 75 random samples on the square
  - This ensures the probability for an unavailable block passing is < 2^(-30)
  - Bandwidth 75* 512B/ 16s = 2.5 kb/s



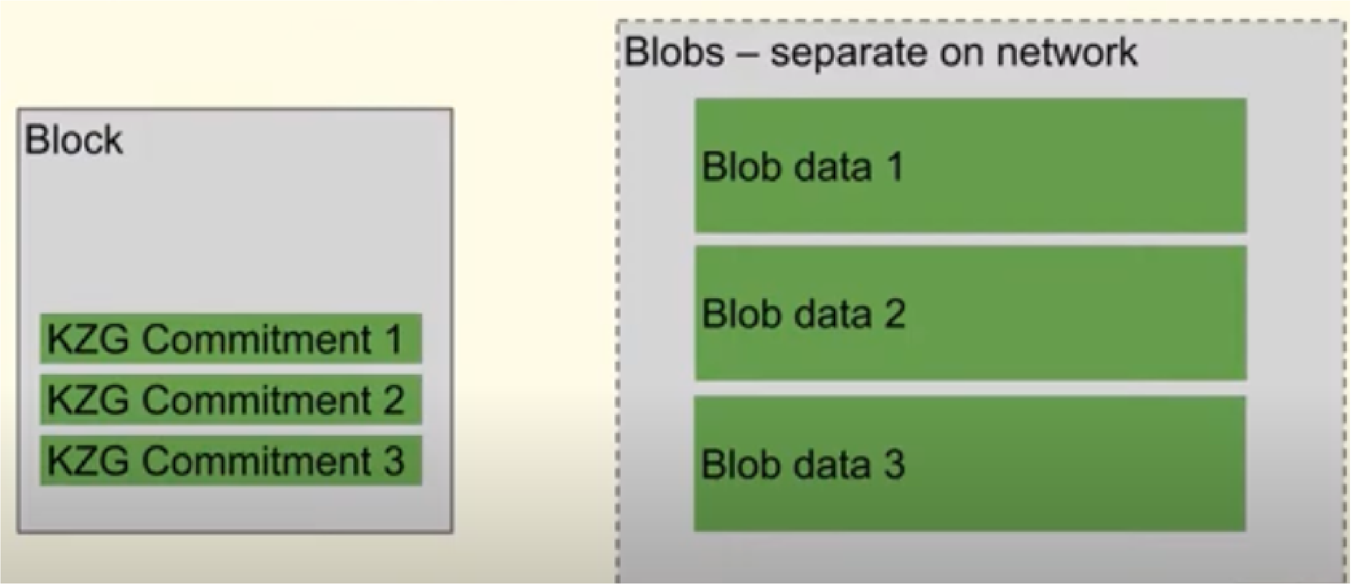
EIP-4844

DA roadmap



4844 design - data blobs

- KZG commitment is added into each block. And the actual data, i.e. the blob data, is separately distributed on the network and work in parallel.



4844 (Proto-Danksharding) details

- Extension of Ethereum to add data blobs to the protocol
  - Blobs are priced independently from execution (new gas type)
  - Blobs are not required to compute state updates, and will only be stored for a short period
- Construction is designed with future upgrades in mind
  - Use KZG commitments so that erasure coding can be used (which is required for DAS, beneficial for networking improvements)
  - Almost all further work can be done just through networking upgrades without consensus changes
  - Rollups won't have to upgrade again to benefit - a single upgrade from CALLDATA to blobs is enough

Q&A

- What are the current challenges after 4844?
  - peerDAS: A conservative version of DAS that can offer additional scaling
  - Full sharding: The end game, with further research needed on security assumption, DHT, etc.
- About the 2d KZG: Where does the 75% of data coming from?
  - If you want 75% of the data to be available, you need 0.75^N = 2^(-30), then you get 75
- Are there any drawbacks of proto-danksharding?
  - Bandwidth increase
- Any suggested interesting research topics that people can start working on?
  - Economics: What could staking be like in the future?
- Thoughts on staking maxi?
  - It doesn't seem to be ideal that all the staking grabbed by the LST. And the question would be how can the stakers help Ethereum? It's still not super clear for now, apart from being a pure money game.
  - As of these days, it would be better to just keep staking more limited. The most dangerous scenario is that the whole protocol becomes ETH staked and it would be hard to reverse (regarding incentive, governance etc.).
- What's Dankrad's feeling on when a protocol has 'enough' economic security?
  - Fairly minimalist on economic security
- What's the most misunderstood aspect of sharding?
  - One misunderstood point is that sharding with rollups means that there will be no more composability.
  - What's cool is that within rollups, you can have massive ecosystem with full composability to some extent if you're willing to integrate with the base layer proposals.
- What are some specific open R&D problems in DA and DAS that people can contribute to?
  - The biggest open question: How to build a robust DHT?
  - The security assumption of DHT is pretty hard right now. It can be broken down even with relatively minor fractions of malicious nodes.
- Execution sharding future roadmap?
  - Execution sharding landscape has changed a lot in the last 5 years.
  - Future roadmap could be increase the gas limit of the execution layer, and put the execution layer onto the data layer