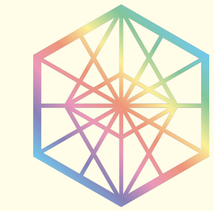
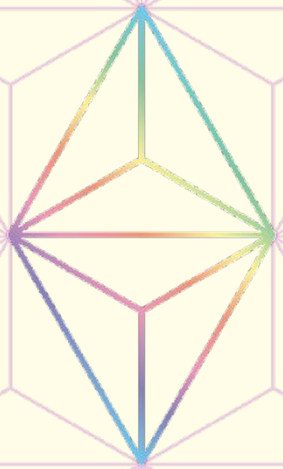


Sharding and Data Availability Sampling

Protocol Study Group 2024



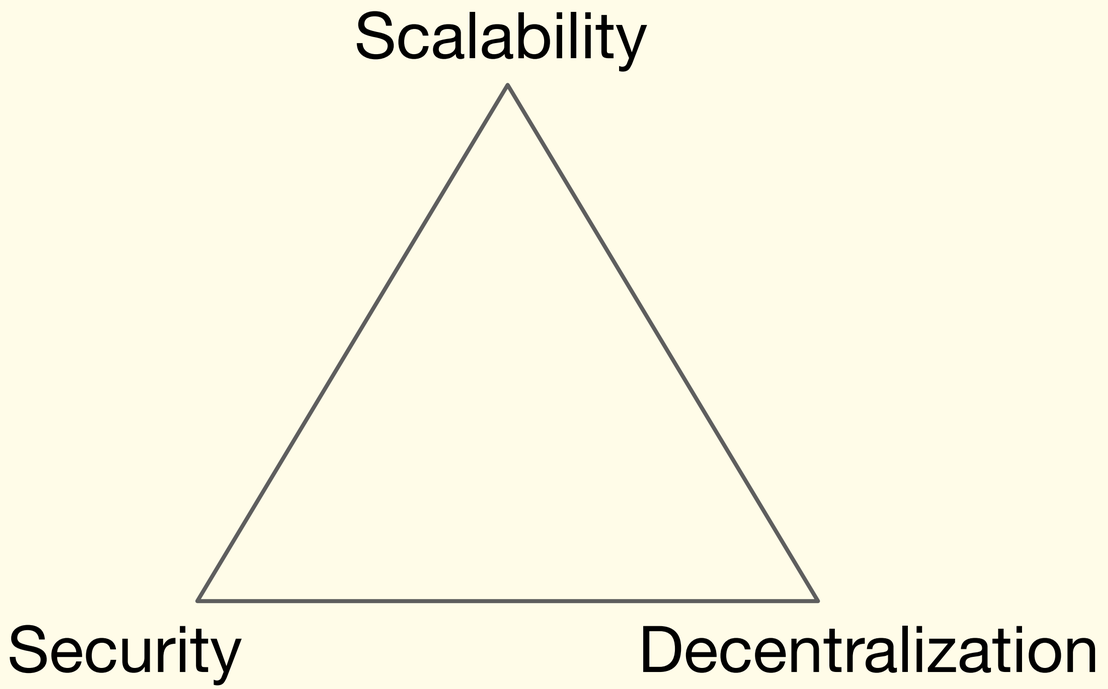
Dankrad Feist
Ethereum Foundation (Consensus Research)

Outline

- Blockchain scalability
- The data availability problem
- Data availability sampling
- Danksharding
- EIP-4844

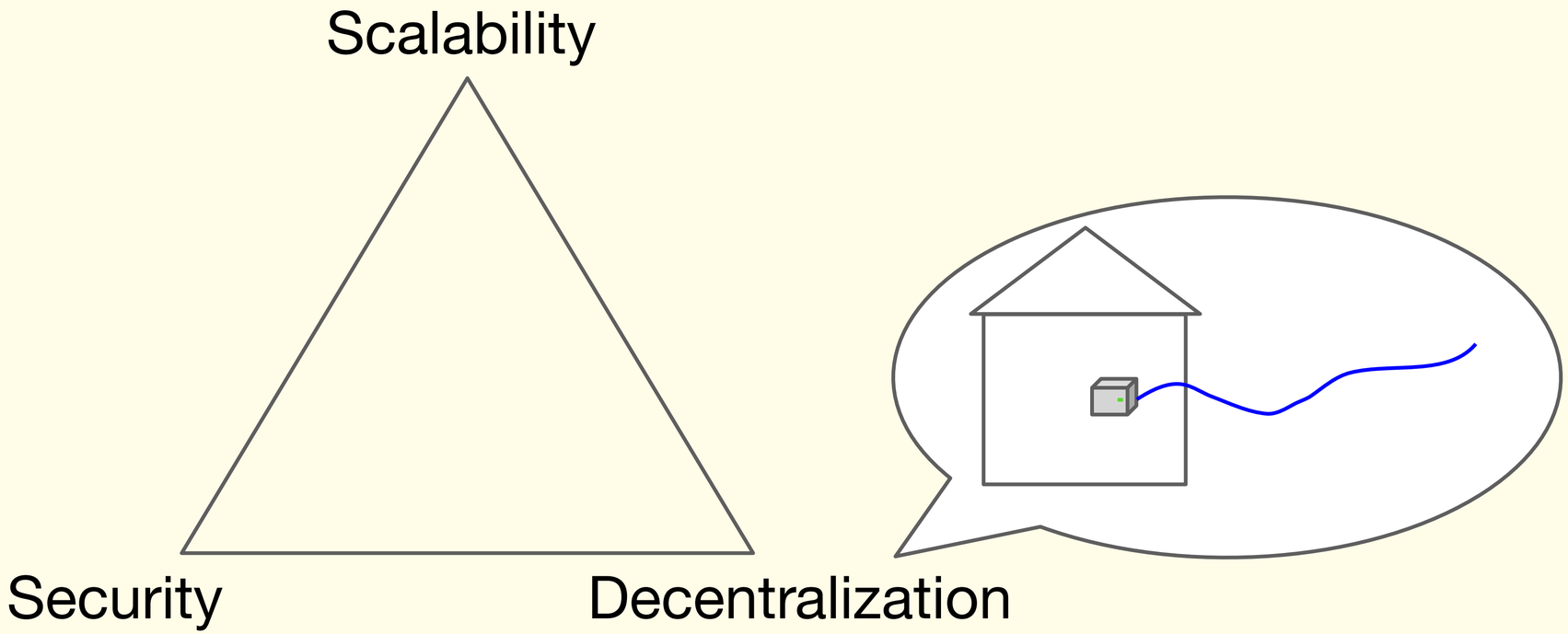
The blockchain scalability trilemma

- It is difficult to design a blockchain that provides scalability, security and decentralization

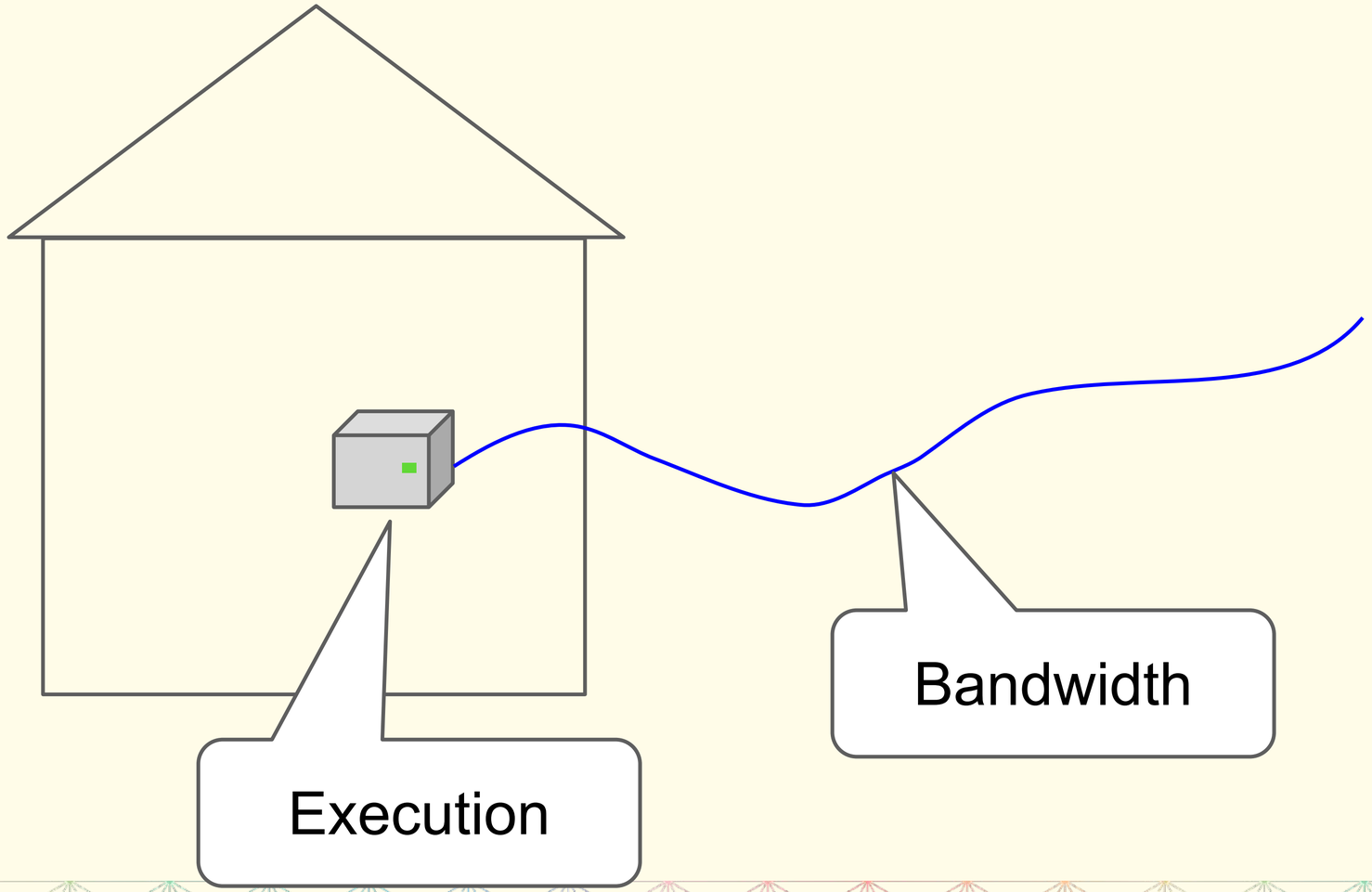


The blockchain scalability trilemma

- It is difficult to design a blockchain that provides scalability, security and decentralization

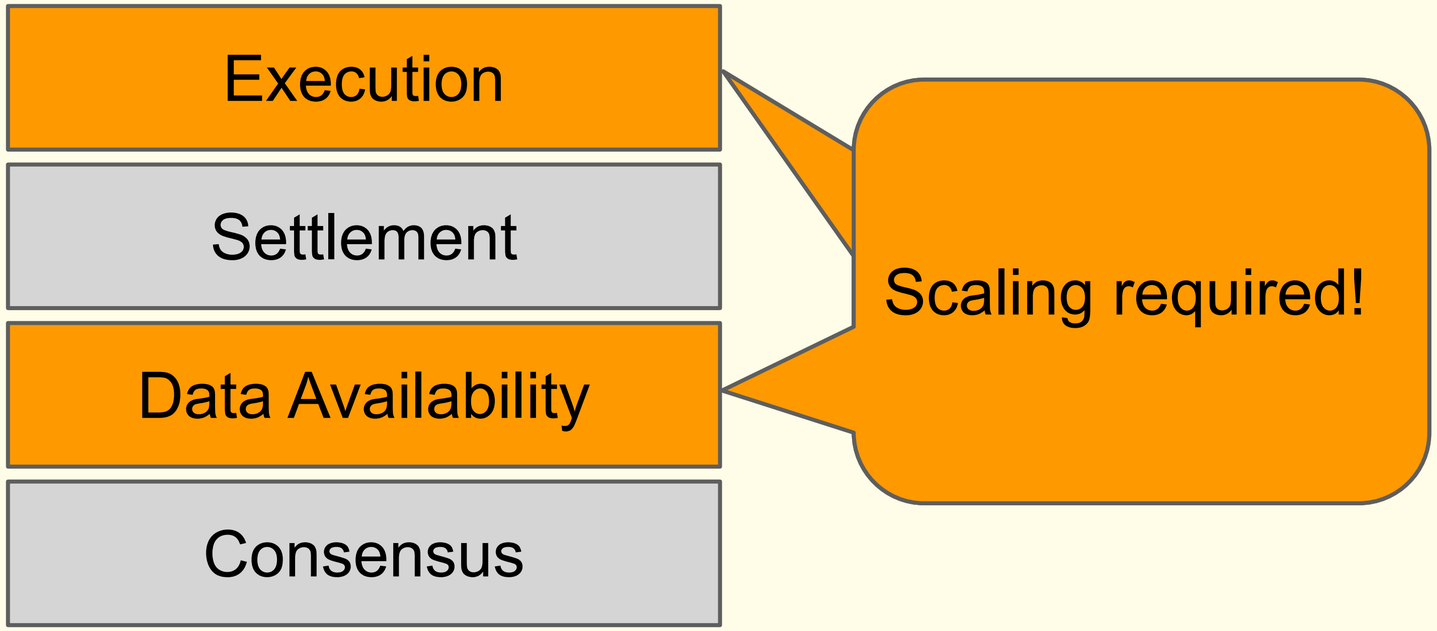


The blockchain scalability trilemma

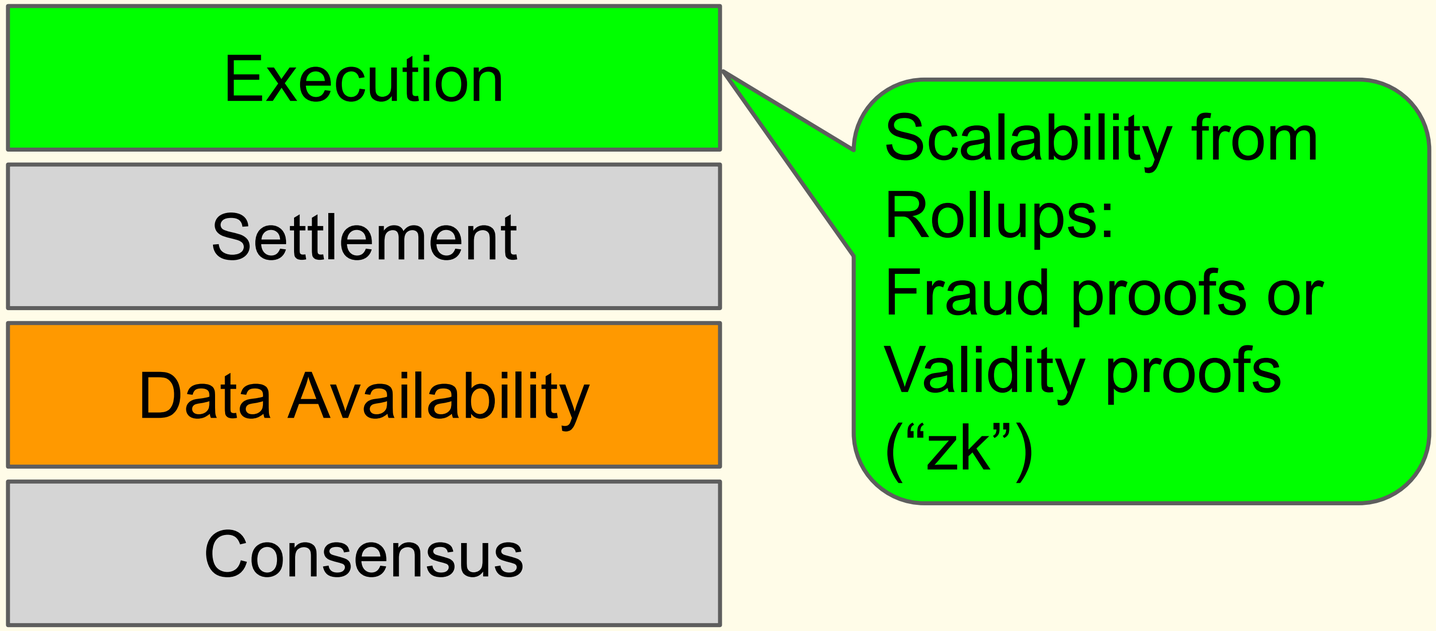


The blockchain stack

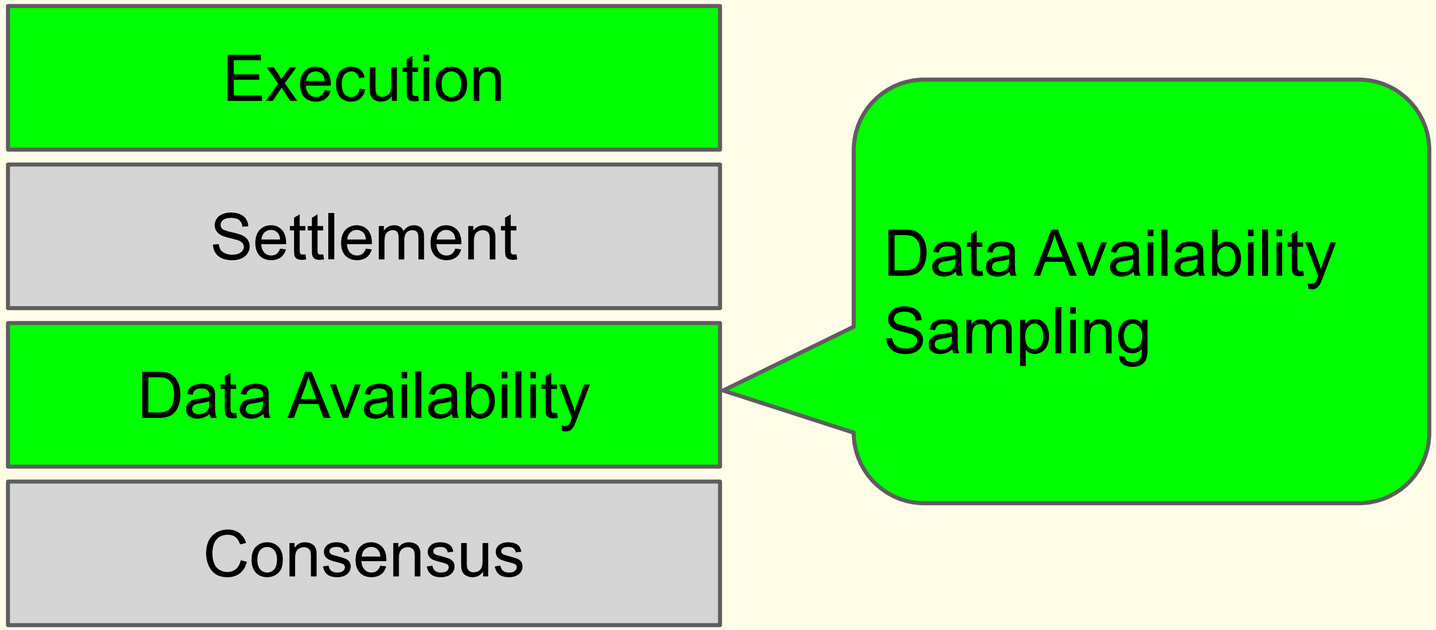
The blockchain stack



The blockchain stack



The blockchain stack



The data availability problem

Definition:

Data availability means that no network participant, including 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)
- 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.

Data availability
= Assurance data was not withheld
= Assurance data was published

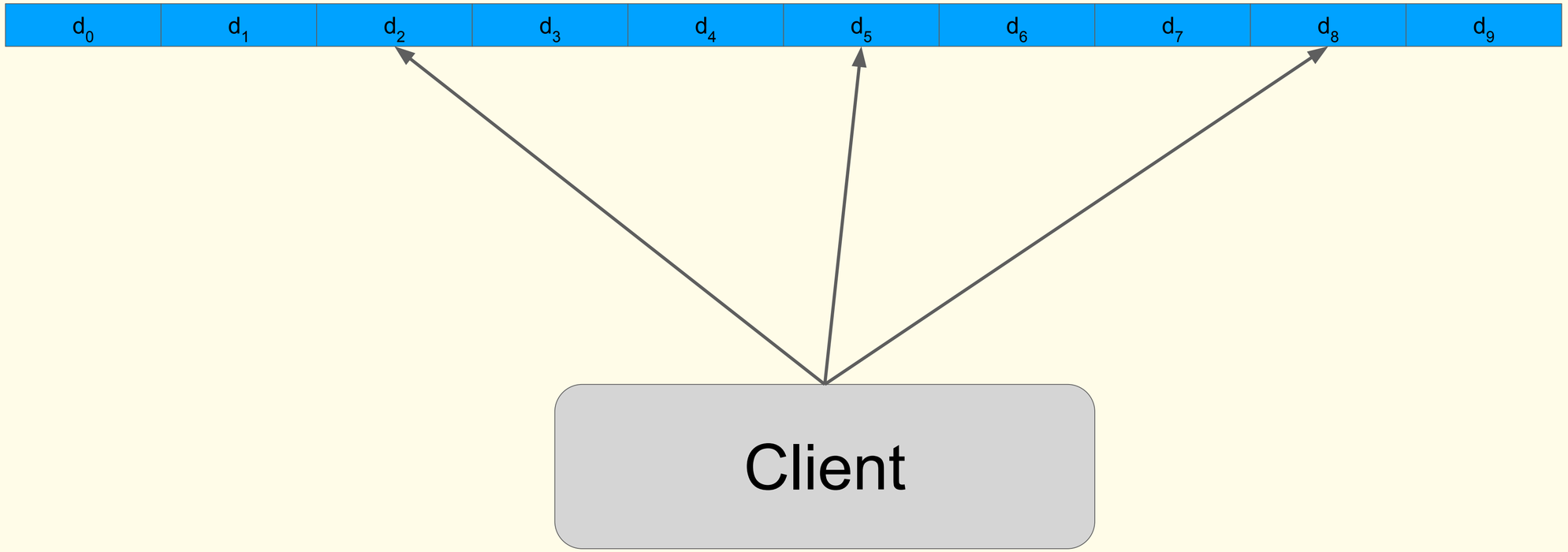
Data availability
≠ Data storage
≠ Continued availability

The data availability problem

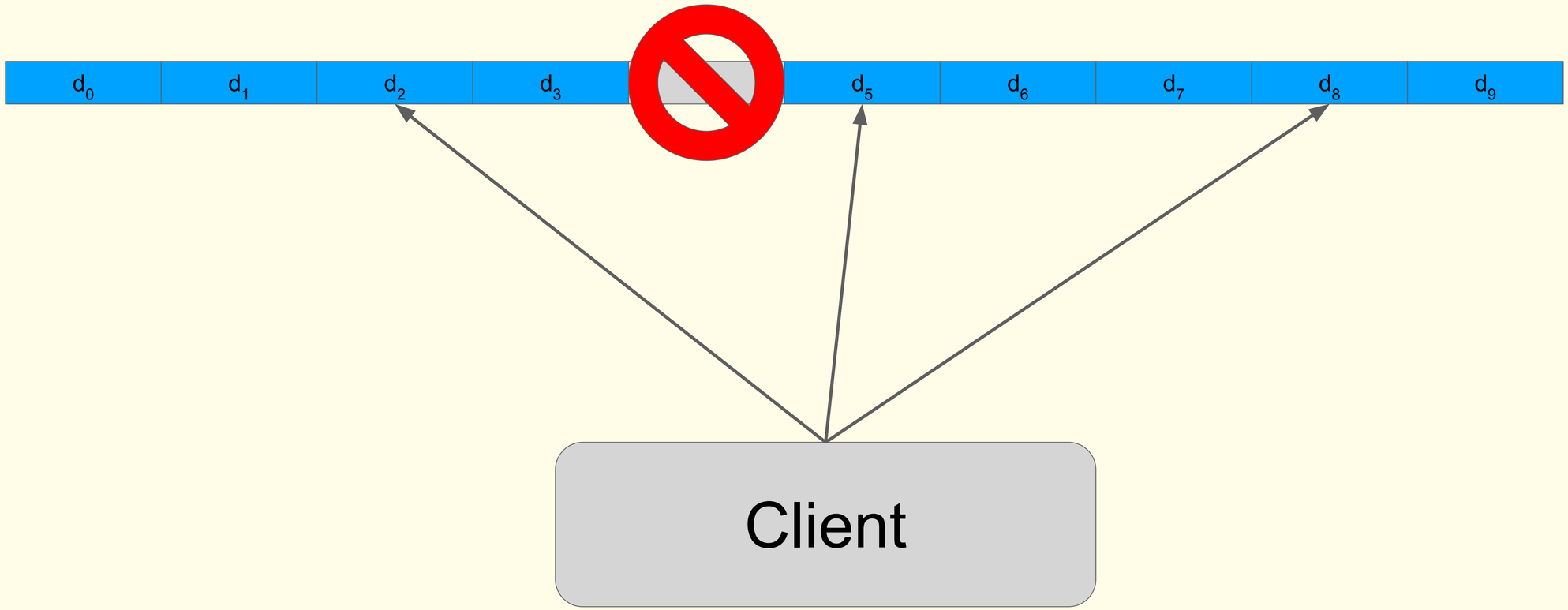
- This sounds like an unimportant detail. Is it really that important?
- Let’s look at our two scalable execution options:
  - Optimistic rollups (using fraud proofs):
    - Any missing data could be fraudulent, e.g. a state change printing 1 trillion Ether. All data needs to be available unconditionally or fraud proofs cannot be constructed
  - ZKRollups (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

Data availability sampling



Data availability sampling



Random sampling is not enough

- Let’s say the light client samples 10% of the data
- The block producer hides a single unavailable transaction
- Probability of catching this one transaction = 10%
- Sampling 10% is already way too much and 10% probability of catching it way too low
- We need a way to amplify this method, so that with a small number of samples we can already be very sure that most of the data is available

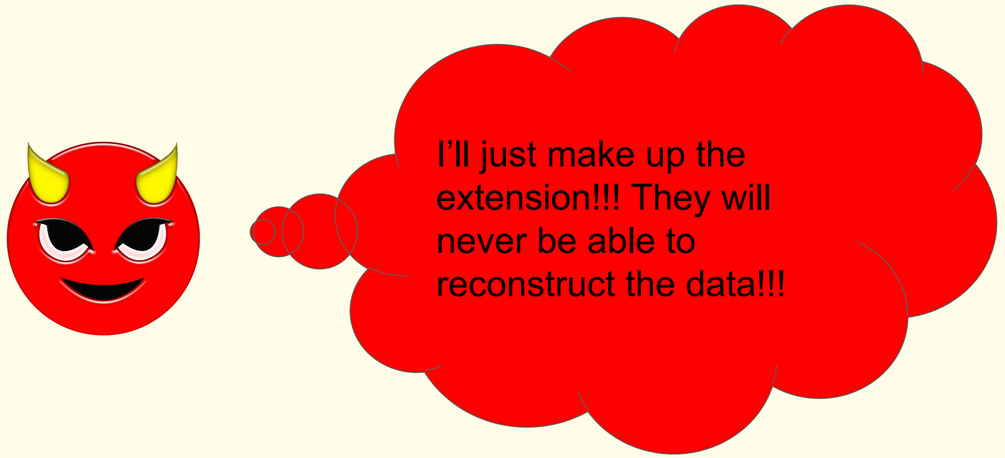
Erasure coding

Original data

Polynomial extension (degree 3)

- Extend the data using a Reed-Solomon code (polynomial interpolation)
- E.g. at coding rate r=0.5, it means any 50% of the blocks (d_0 to e_4) are sufficient to reconstruct the whole data
- Now sampling becomes efficient
  - E.g. query 30 random blocks; if all are available, probability that more than 50% not available is 2^{-30}
- But: We need to ensure that the encoding is correct

Ensuring validity of the encoding



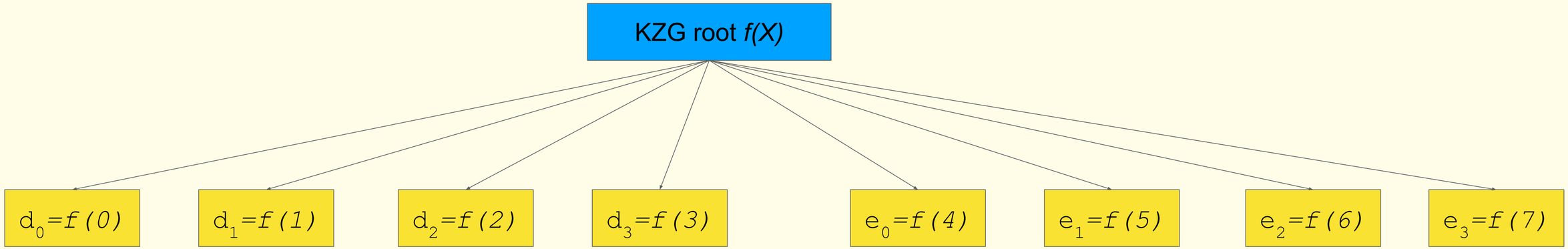
Original data

d_0	d_1	d_2	d_3

Polynomial extension (degree 3)

e_0	e_1	e_2	e_3

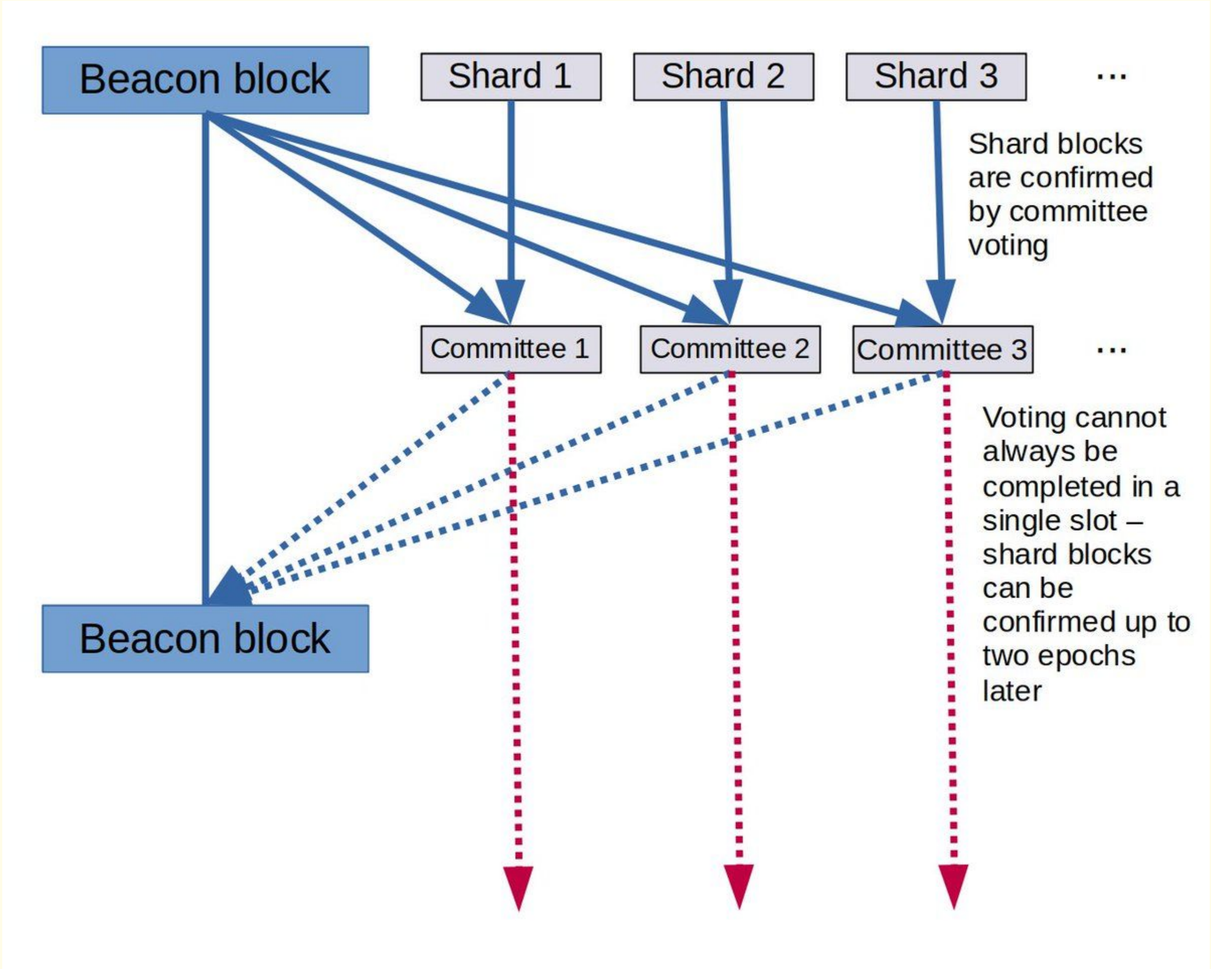
KZG Commitments as Data Availability Roots



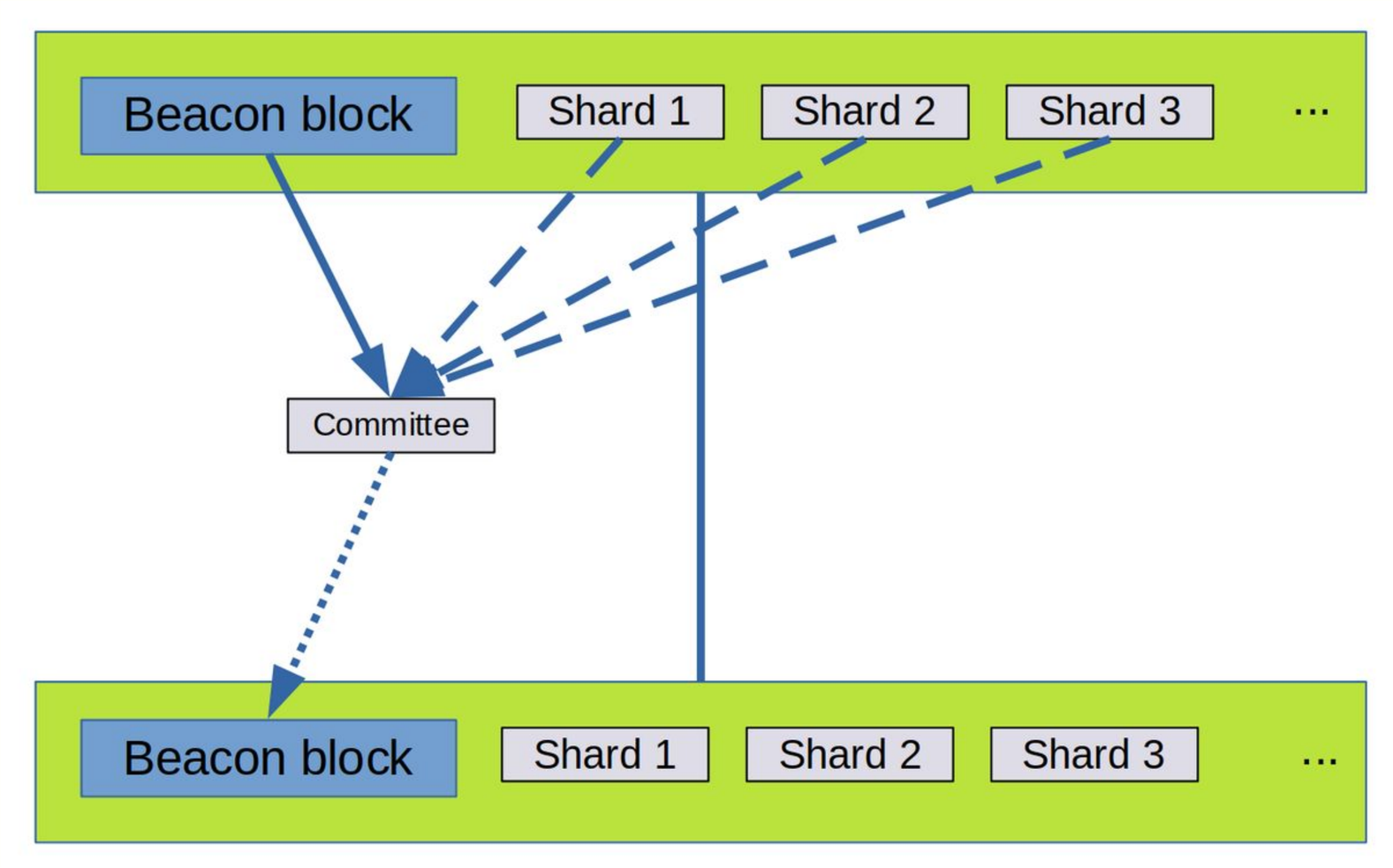
- Think of the “KZG root” as something similar to a Merkle root
- The difference is that it commits to a “polynomial”:
  - All points are guaranteed to be on the same polynomial
  - Merkle root (vector commitment, not poly commitment) cannot guarantee this

Danksharding

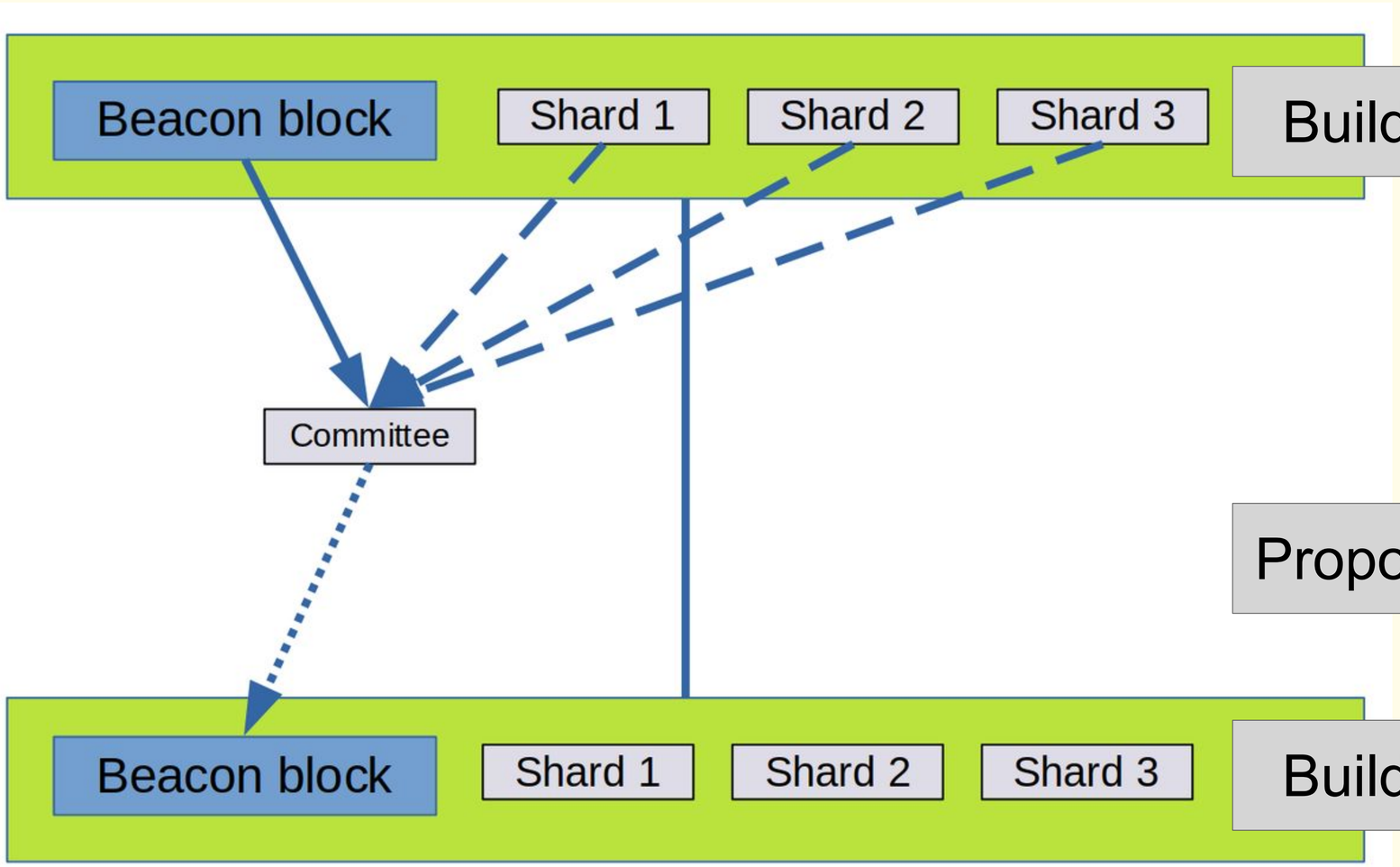
Separate shard proposals



Danksharding



Danksharding



ProposerBlock

Proposer

BuilderBlock

Builder

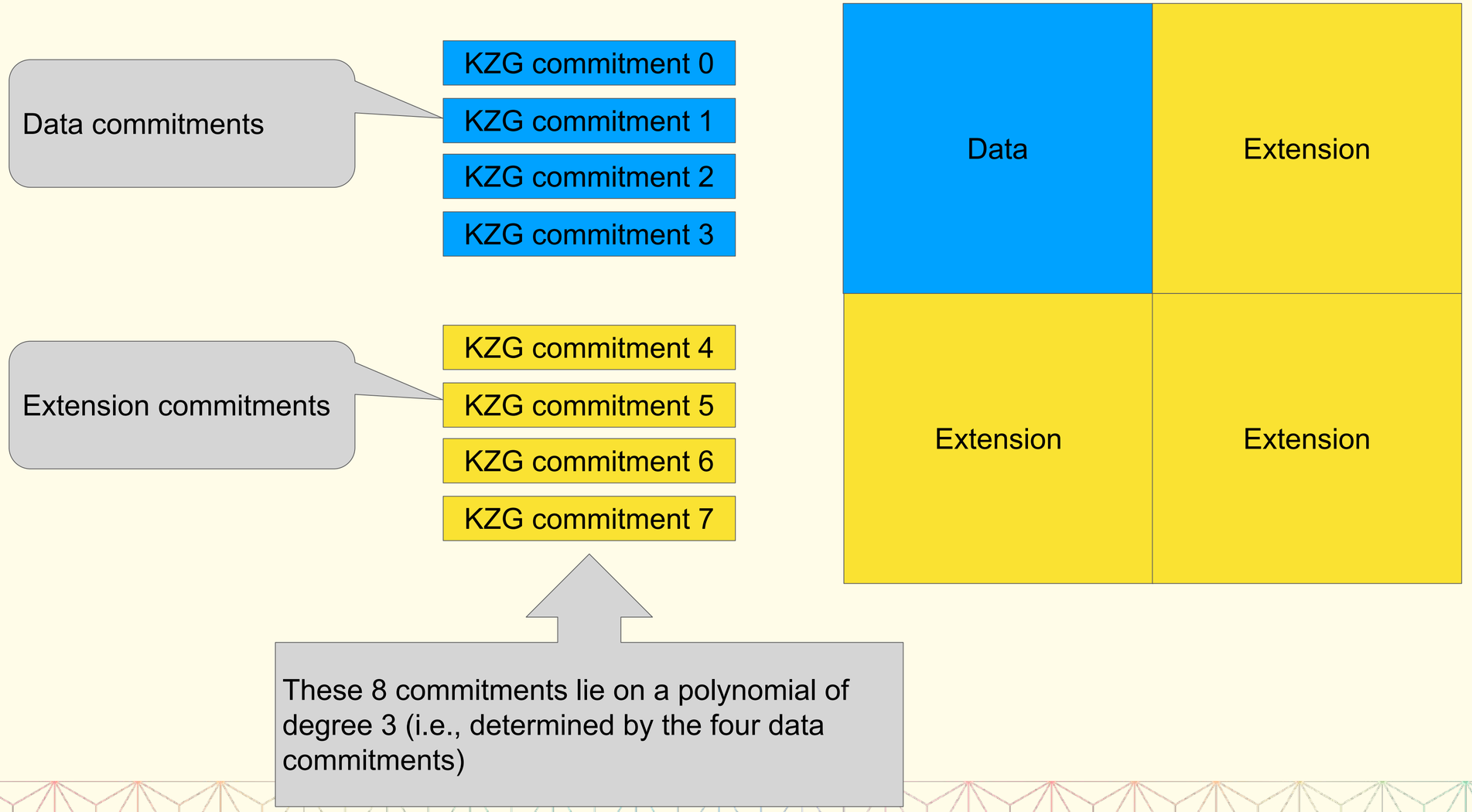
ProposerBlock

Proposer

BuilderBlock

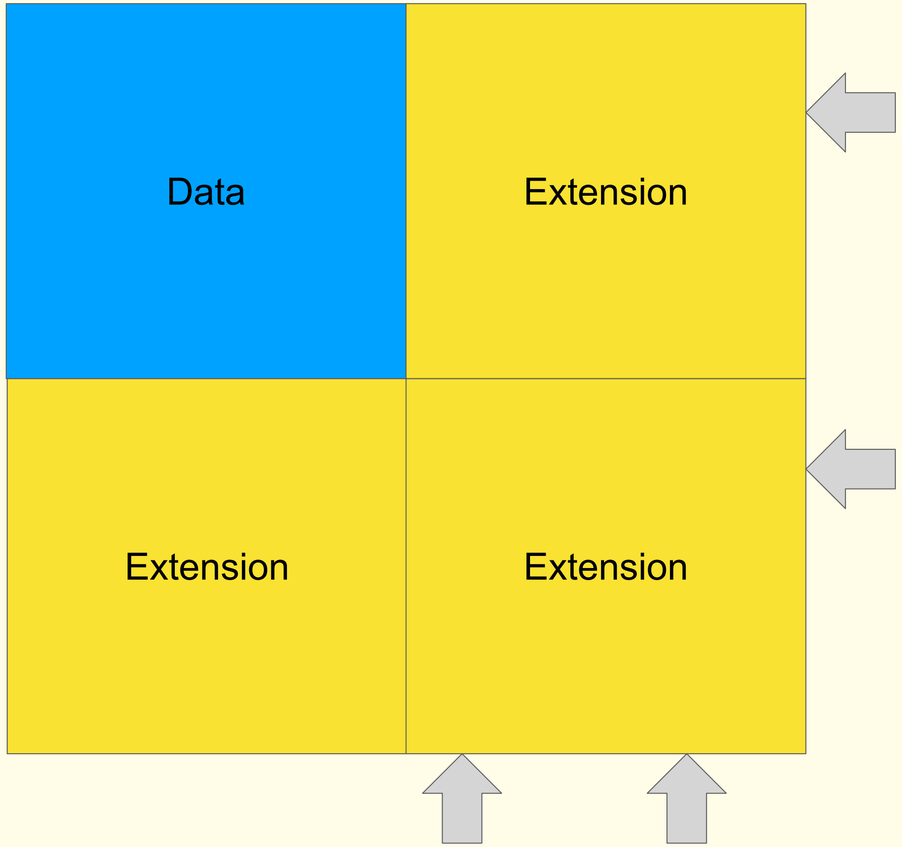
Builder

KZG 2d scheme



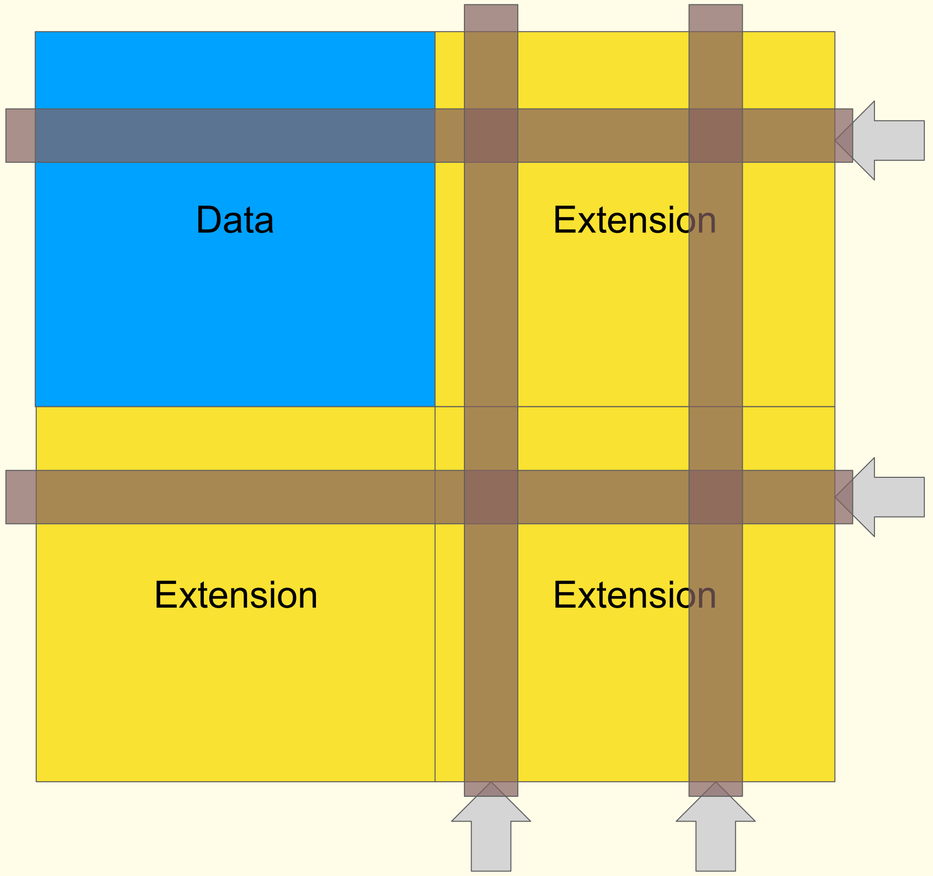
Danksharding honest majority validation

- Each validator picks $s = 2$ random rows and columns
- Only attest if the assigned row/column are available for the entire epoch
- An unavailable block (<75% available) cannot get more than $2^{-2s} = 1/16$ attestations



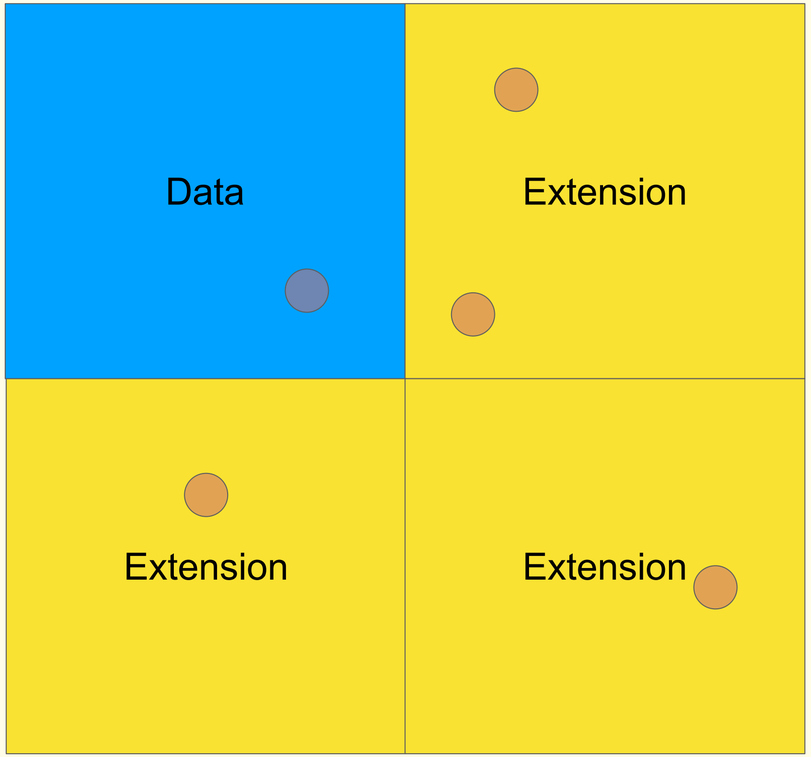
Danksharding reconstruction

- Each validator should reconstruct any incomplete rows/columns they encounter
- 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)



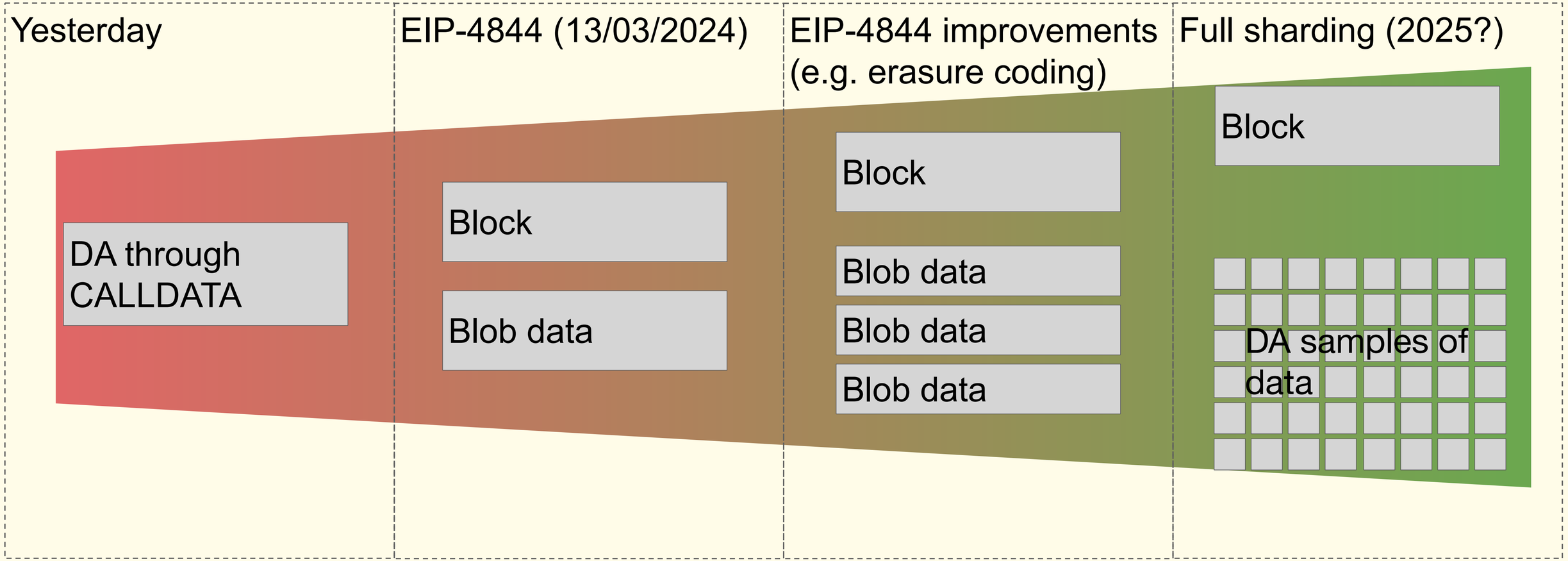
Danksharding DA sampling (malicious majority safety)

- Future upgrade
- Each full node checks 75 random samples on the square
- This ensures the probability for an unavailable block passing is $< 2^{-30}$
- Bandwidth 75*512 B / 16s = 2.5 kb/s

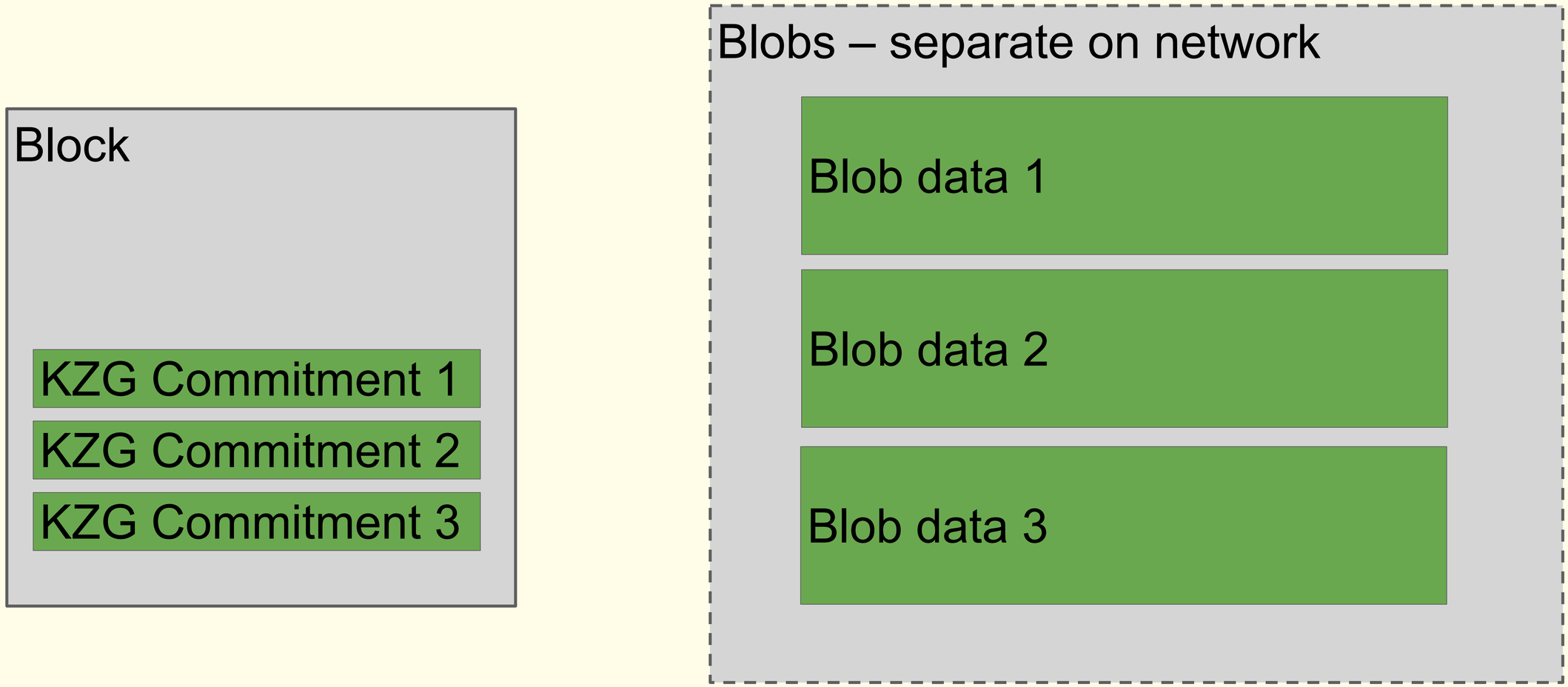


EIP-4844

Ethereum’s data availability roadmap



EIP-4844 design – data blobs



EIP-4844 (“Proto-Danksharding”)

- 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
  - Uses KZG commitments so that erasure coding can be used (required for DAS, beneficial for networking improvements)
  - Almost all further work can be done without consensus changes just through networking upgrades
  - Rollups will also not have to upgrade again to benefit – a single update from CALLDATA to blobs is enough

Thank you