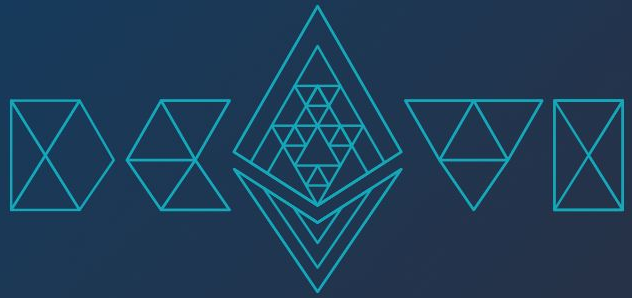


KZG Ceremony

or how I learnt to stop worrying and love trusted setups

Carl Beekhuizen

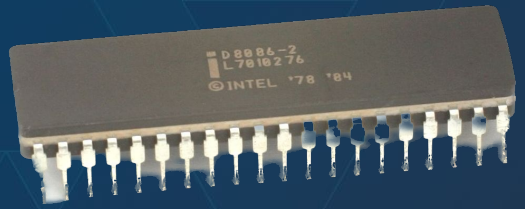
EF Researcher


THINKING BACK 5 MONTHS
7000 GWEI
BASEFEE FLASHBACKS
imgflip.com

Enter the L2s

Scaling

L2s Scale Compute

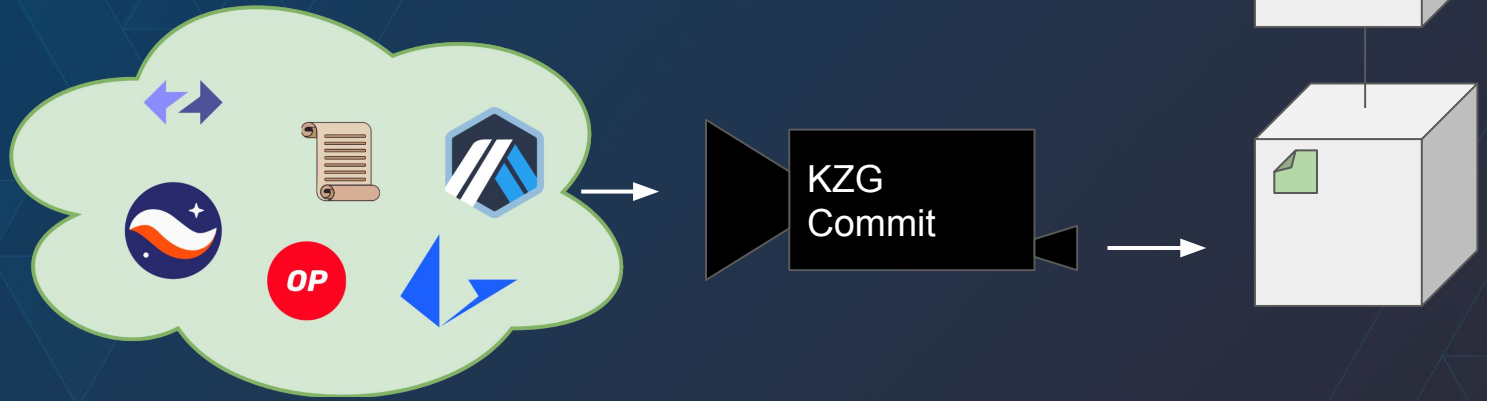


L2s

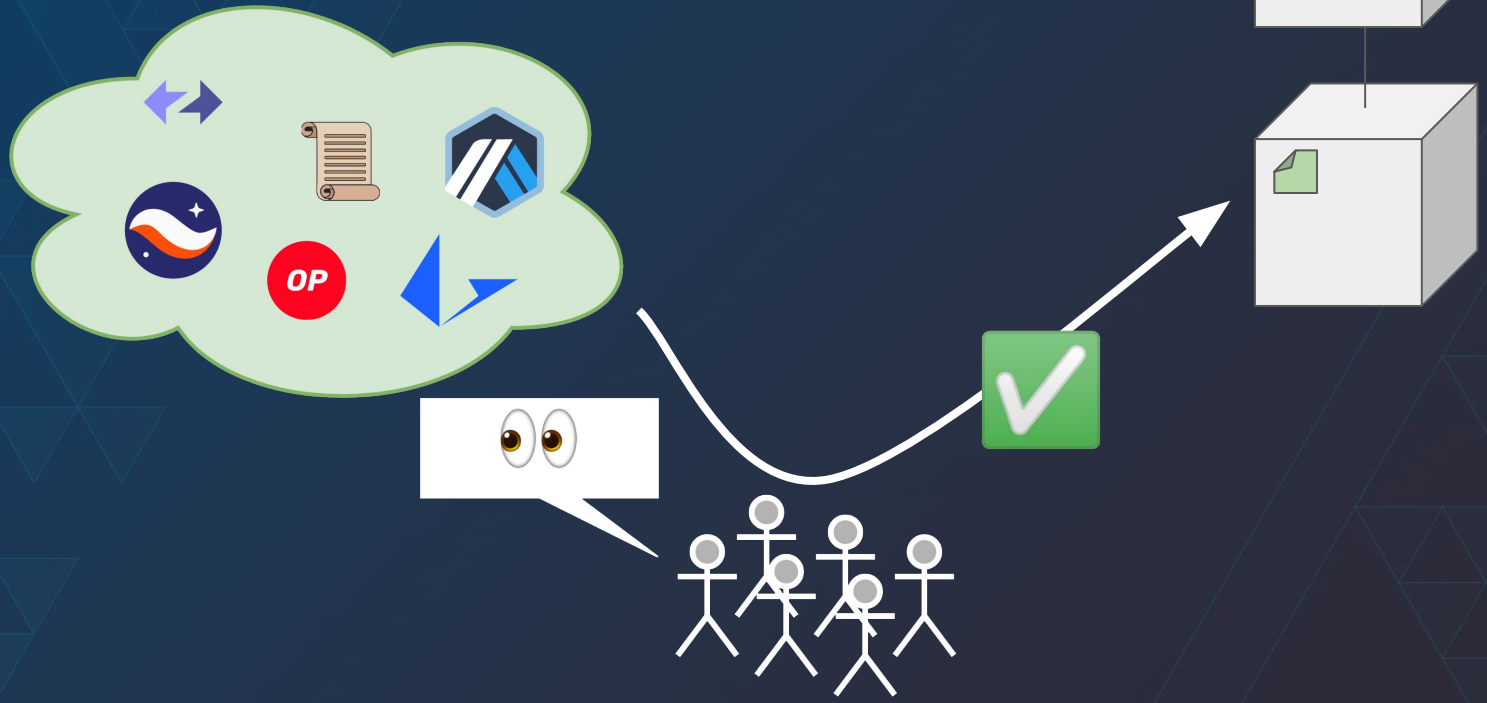
(Proto) Danksharding Scales Data Availability

EIP 4844

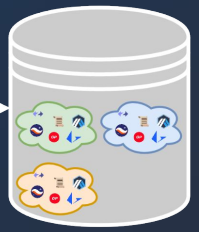
(Proto) Danksharding / EIP-4844

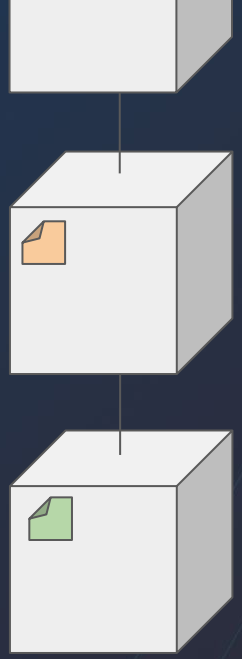


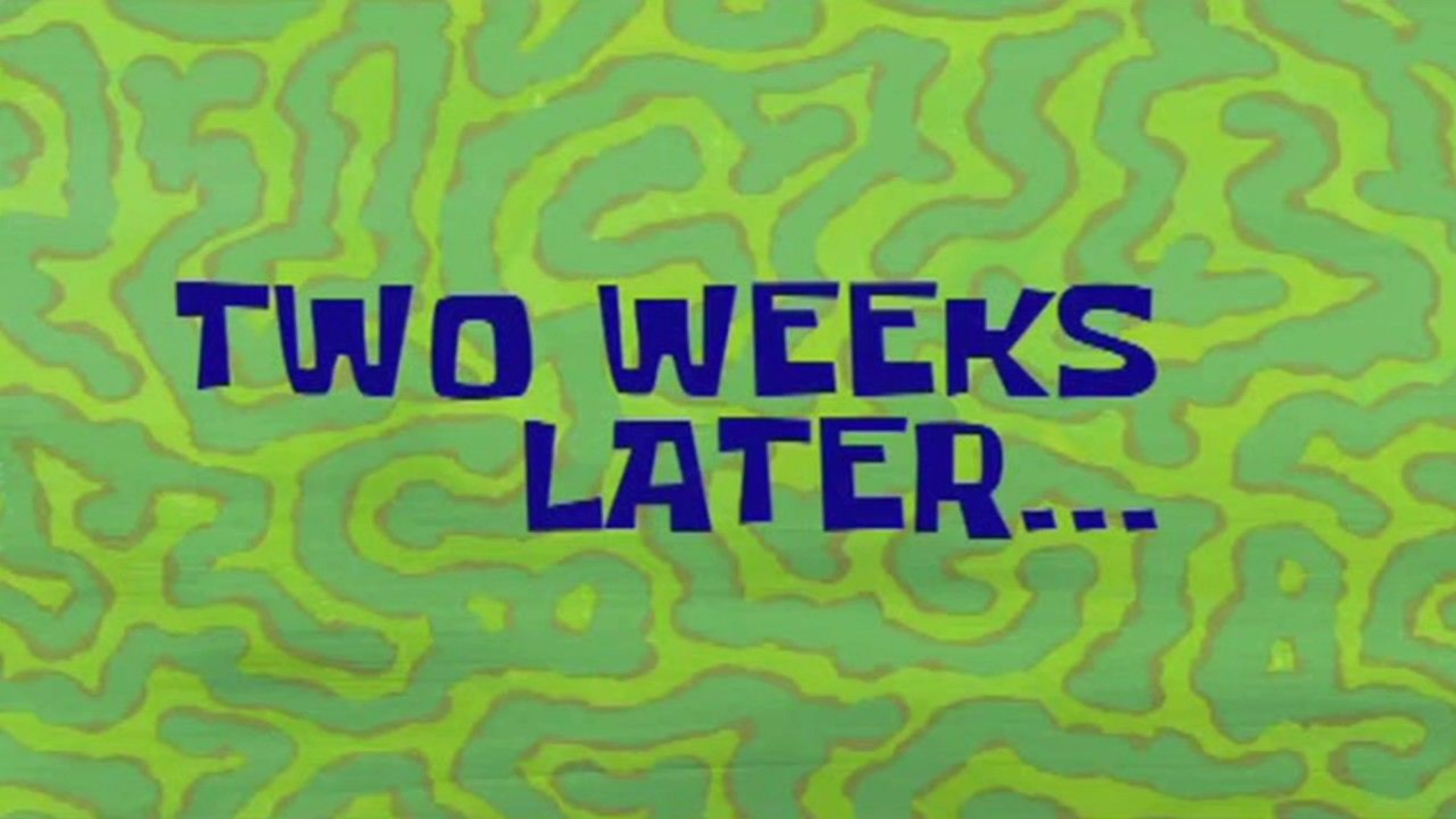
Checking availability



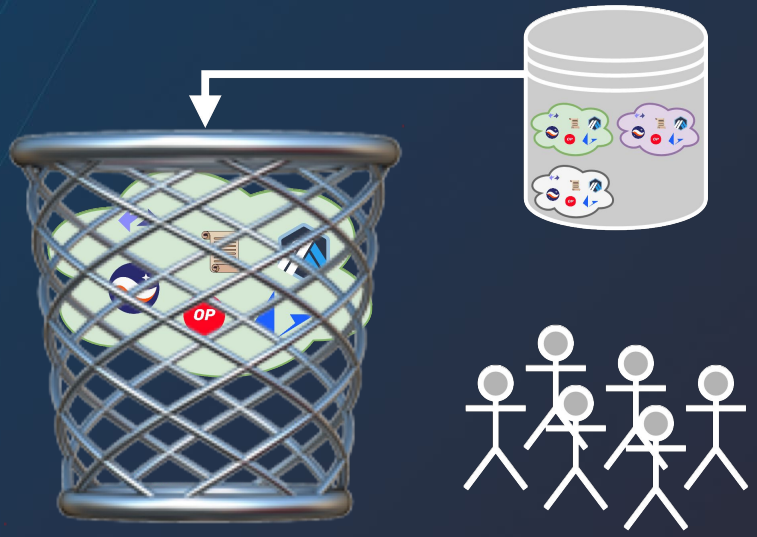
Serving the data

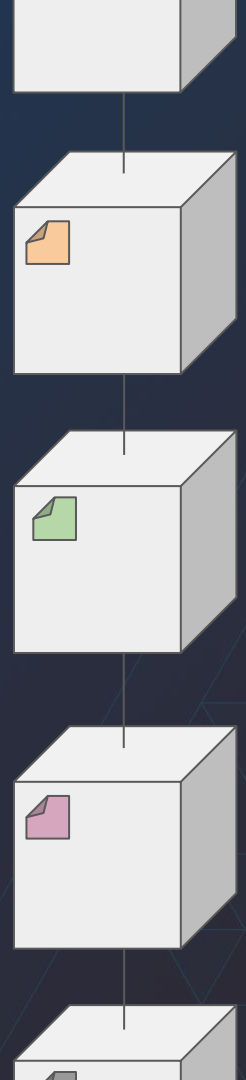





TWO WEEKS
LATER...

Throw it all away!





Why KZG?

	Merkle (SHA 256)	Merkle (Arithmetic)	**KZG**	IPA
Verification inside SNARK	Prohibitive	Doable	**Cheap**	Cheap
Proof Verification (& low deg. proofs)	Hard	SNARKs needed for easy verify	**Easy**	O(N) time
Proof generation in O(n*log(n))	Yes	Yes	**Yes**	Only with larger proof-size
Commit linearity	No	No	**Yes**	Yes
Proof linearity	No	No	**Yes**	No
Security assumption	Very safe	Immature hash functions	**Trusted setup + Quantum vuln.**	Quantum vuln.

Modified from Vitalik's post: ethresear.ch/t/13863

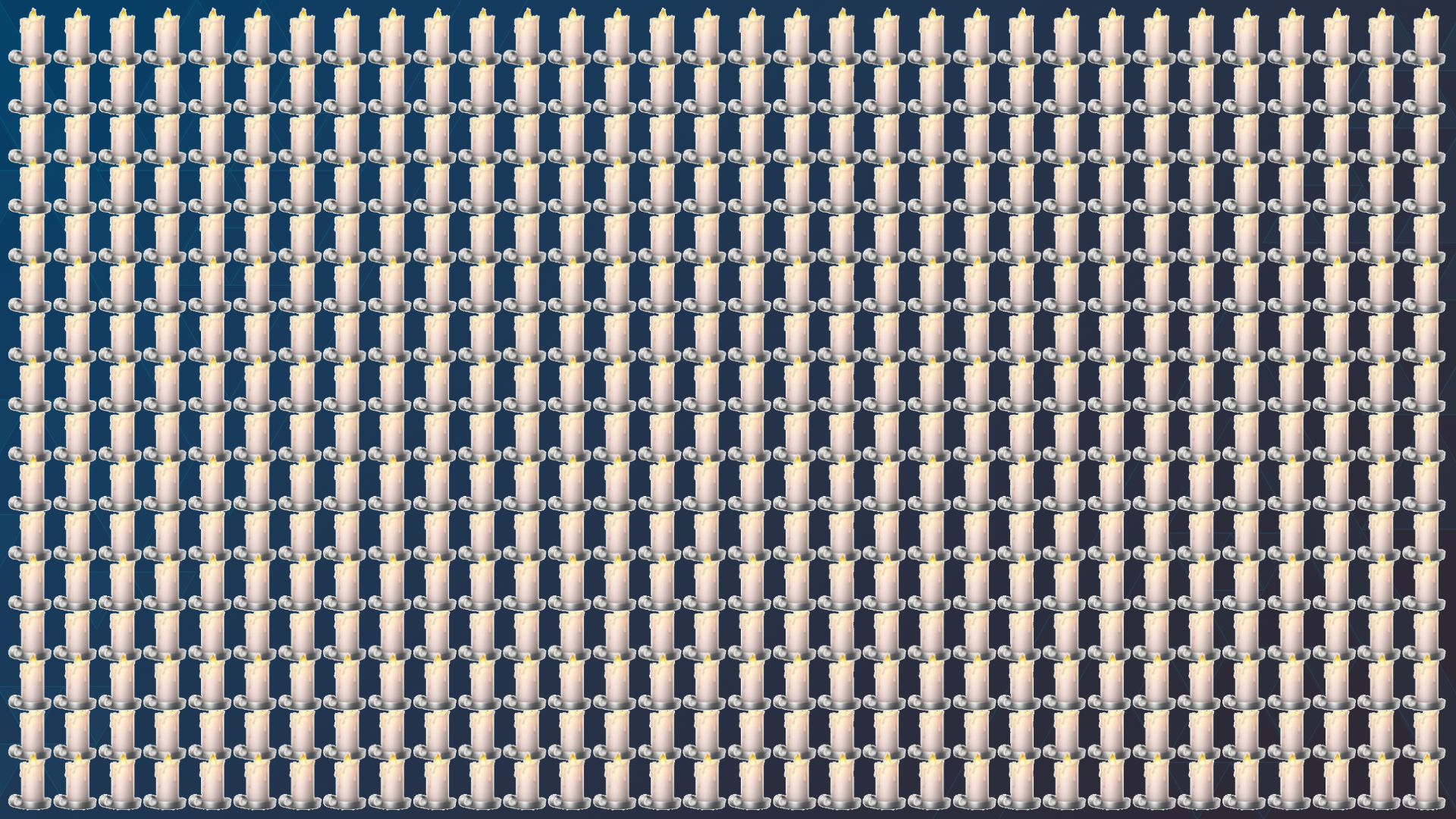
Wait, what a trusted setup?!

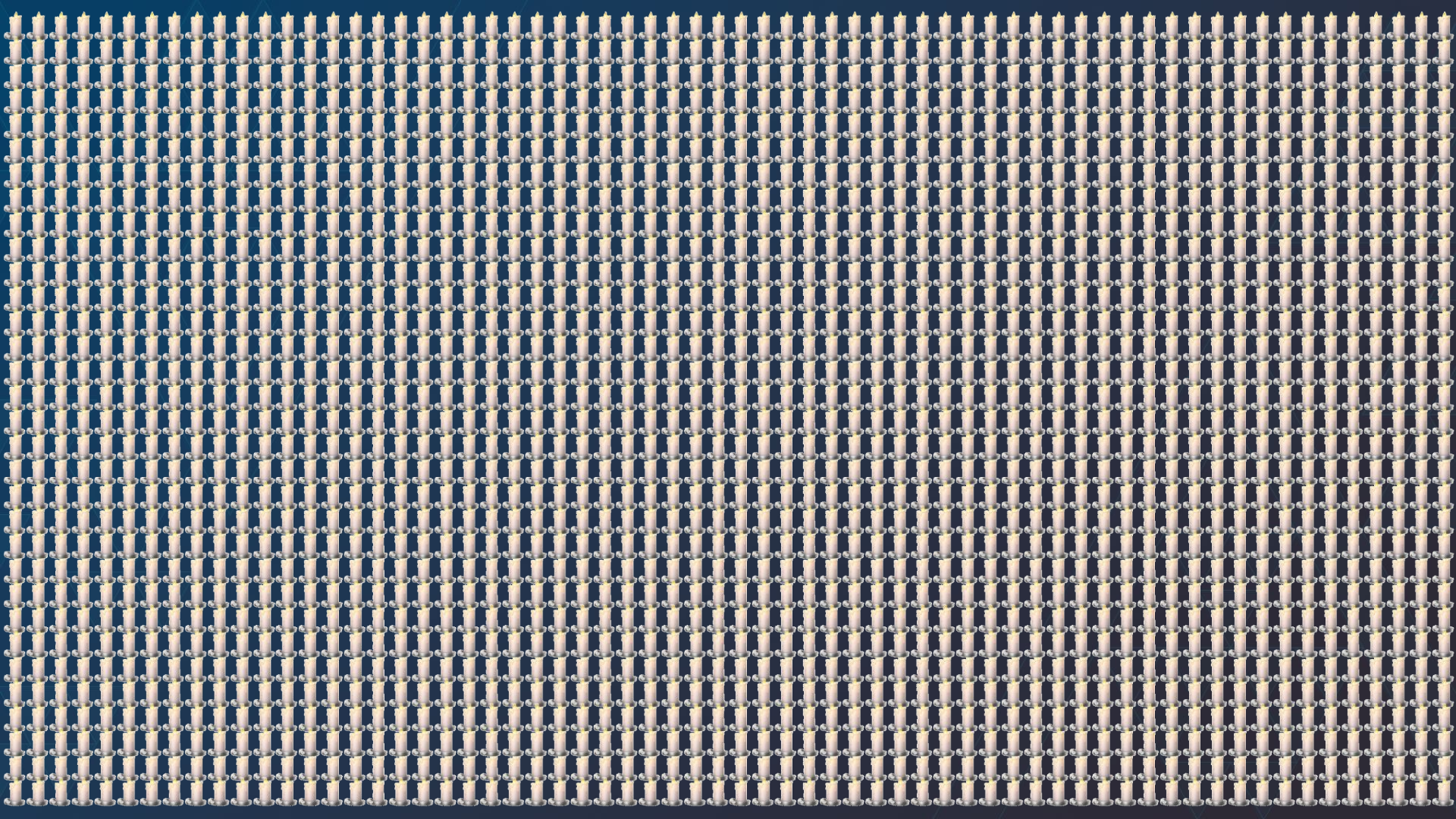
Summoning a secret

Only one honest participant required

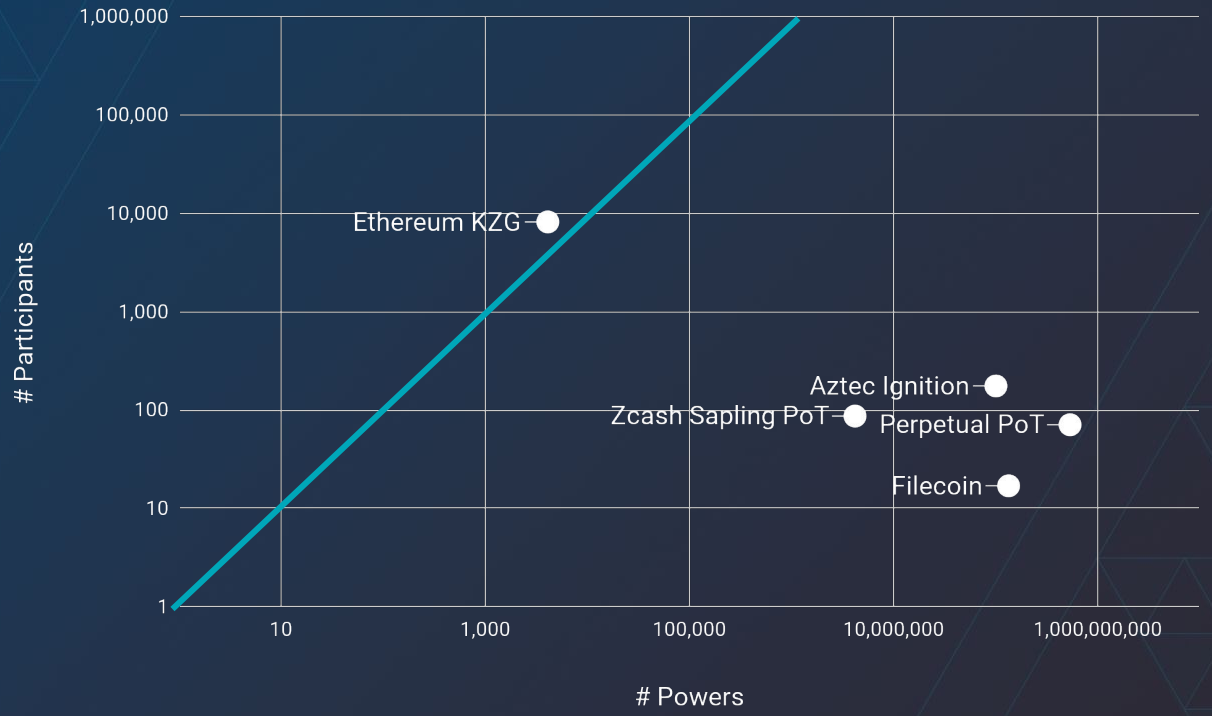
There will be bugs!

There will be malicious users





More Participants than Powers!



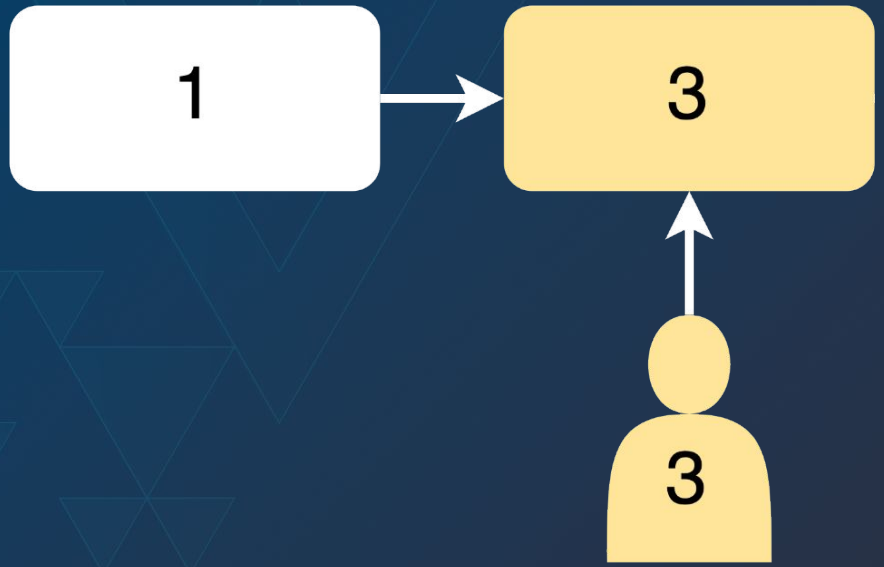
Note: Log-Log scale

Design decisions - KISS

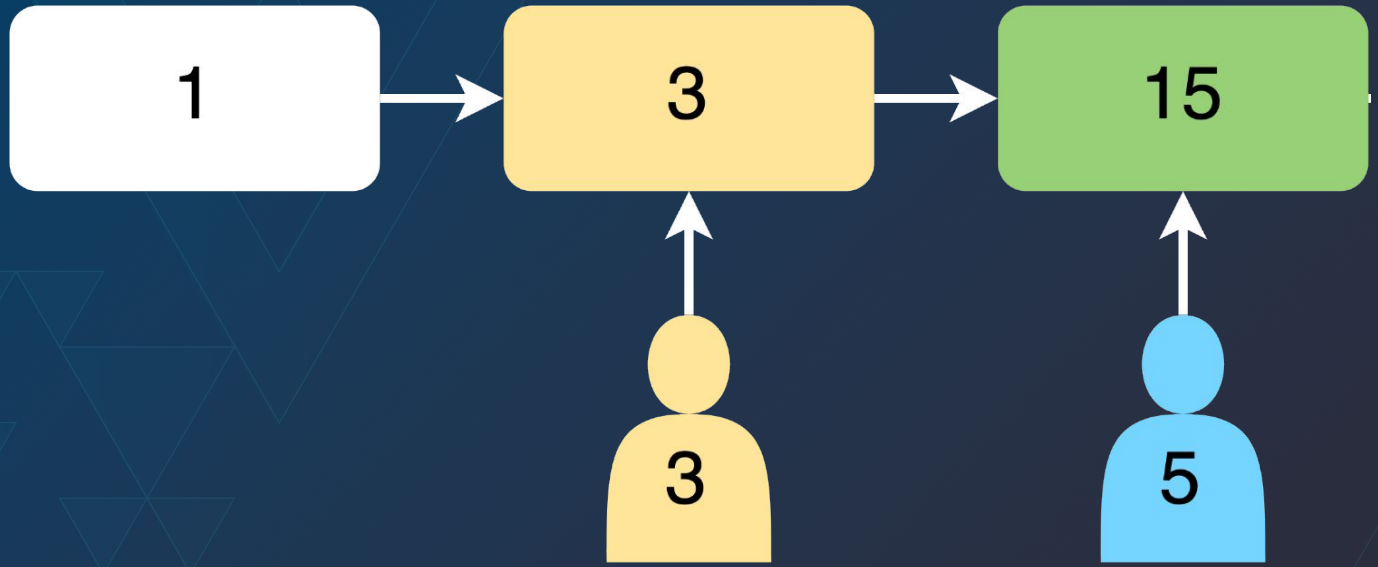
- Quick and easy ⇒ Many participants
- Simple specs ⇒ Many implementations (write your own!)
- Few powers ⇒ Low resource requirements
- Updatable setup ⇒ Future ceremonies can add to it

How does it work?

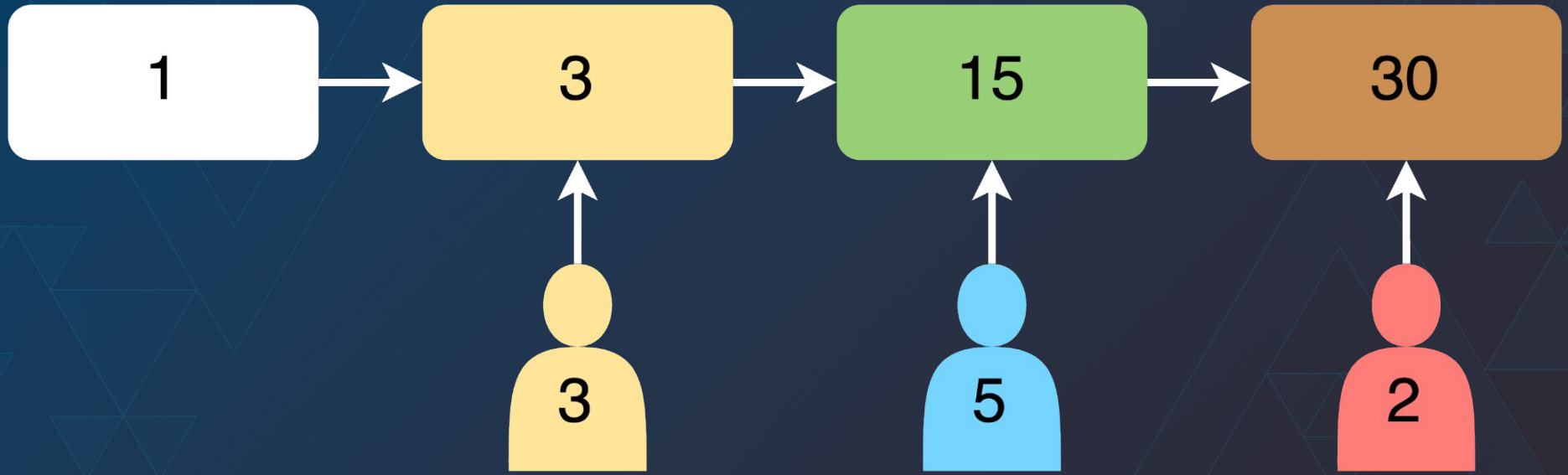
How does the ceremony work



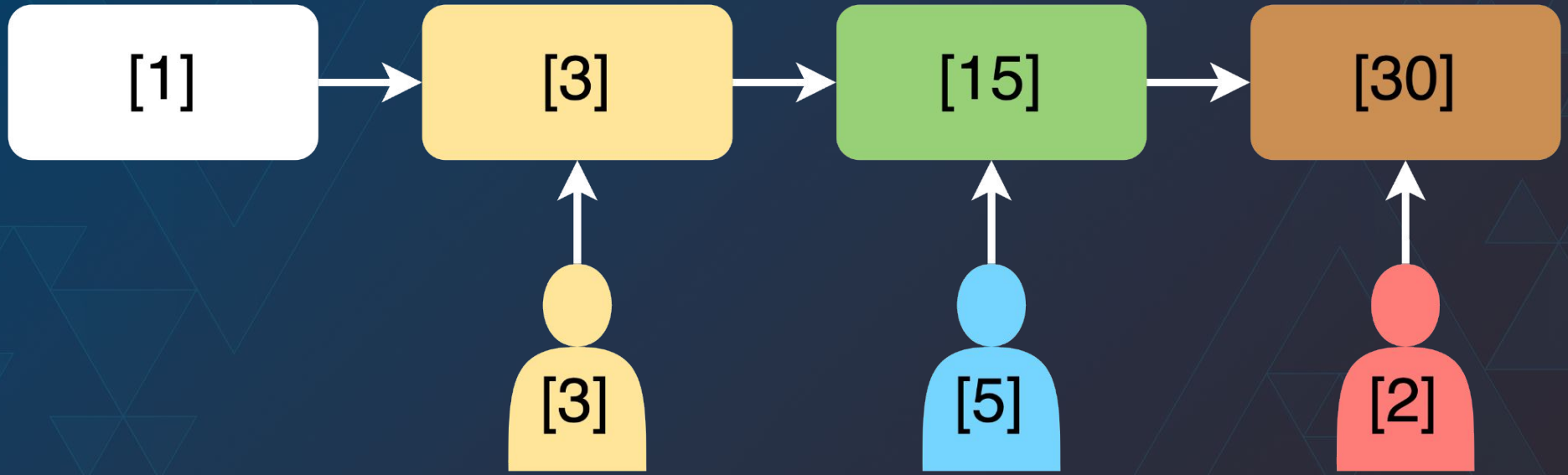
How does the ceremony work



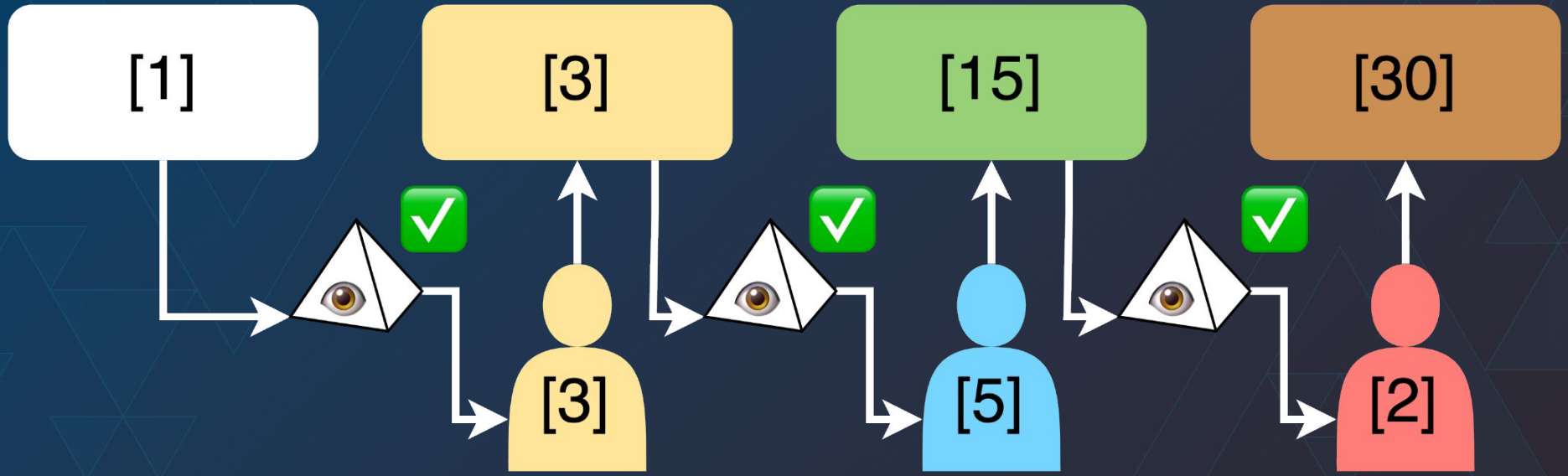
How does the ceremony work

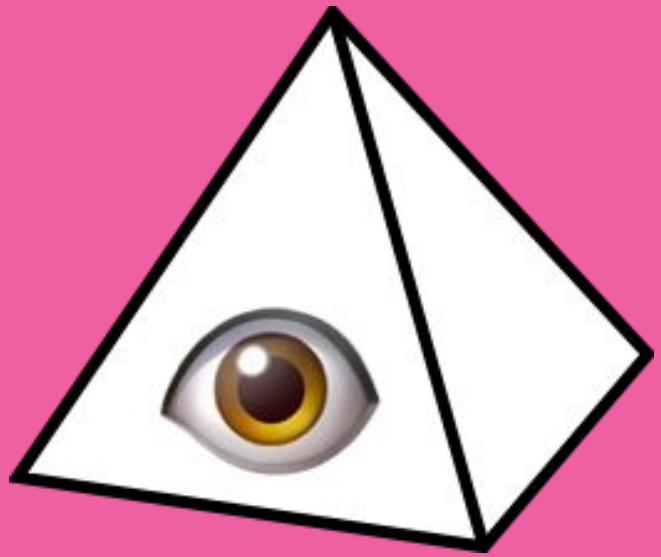


Enter the curves



Sequencer checking things





What could go wrong?

- Censor contributions
  - Lie about verification
  - Not give you a turn
- But, this is attributable!

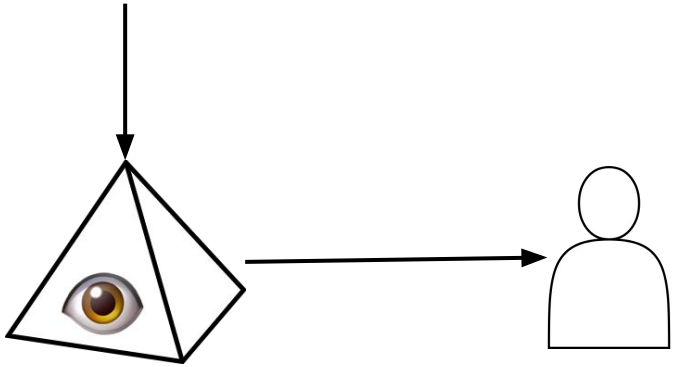
How does it *really* work?

Powers??

$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}]$$

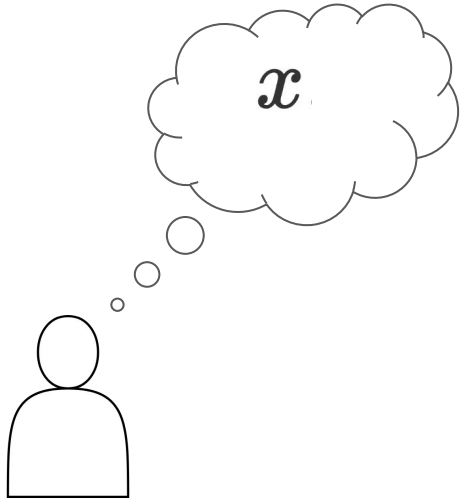
Powers??

$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}]$$



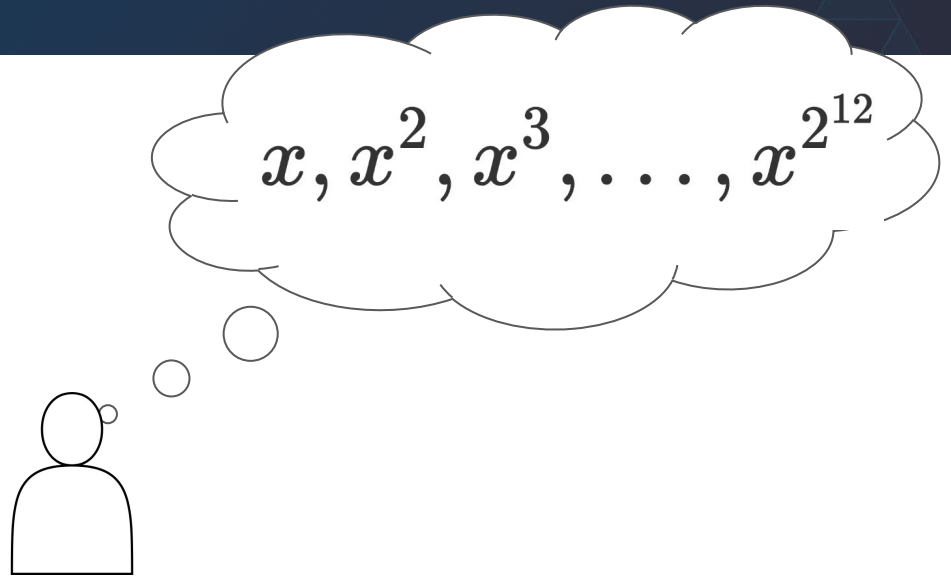
Entropy Calculation

$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}]$$



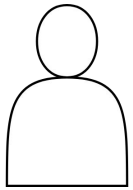
Doing a heckn' big calculate

$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}]$$



Even more math...

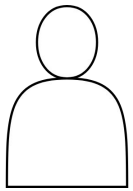
$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}] \qquad x, x^2, x^3, \ldots, x^{2^{12}}$$



$$x[s], x^2[s^2], x^3[s^3], \ldots, x^{2^{12}}[s^{2^{12}}]$$

Even more math...

$$[s], [s^2], [s^3], \ldots, [s^{2^{12}}] \qquad x, x^2, x^3, \ldots, x^{2^{12}}$$

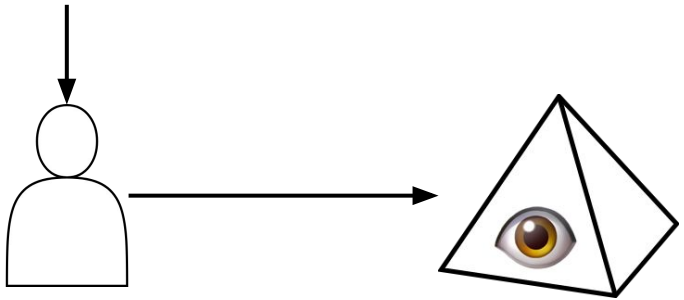


$$x[s], x^2[s^2], x^3[s^3], \ldots, x^{2^{12}}[s^{2^{12}}]$$

$$[x.\,s], [(x.\,s)^2], [(x.\,s)^3], \ldots, [(x.\,s)^{2^{12}}]$$

Returning the Powers

$$[x.\,s], [(x.\,s)^2], [(x.\,s)^3], \ldots, [(x.\,s)^{2^{12}}]$$



When can you participate?

Testnet

Next few days - Opening the ceremony up to the public to smooth out the UI.

Final Audit

7 Nov - The sequencer and the primary contributor implementation audit starts

Launch

End Nov - The ceremony goes live. Runs for at least 2 months.

EIP-4844

2023 sometime - Up to the client devs and community demand.

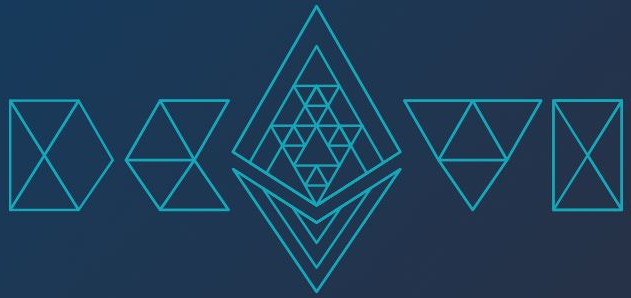
Grants!

Writing an implementation

- 👩‍💻 Roll your own crypto!
- 🛠️ Implement a client

Crazy randomness generation

- 🏜️ Vision trip
- 🌋 Collect weird entropy
- 🔥 Destroy your secret (and PC?!)



Thank you!

Carl Beekhuizen

EF Researcher

carl@ethereum.org

@CarlBeek

Verification

A contribution contains

$$[x.\,s]_1, [(x.\,s)^2]_1, [(x.\,s)^3]_1, \ldots, [(x.\,s)^{2^{12}}]_1$$

$$[s]_2 \qquad [x.\,s]_2$$

Update check

$$e([x.\,s]_1, g_1) \stackrel{?}{=} e([x]_1, [s]_2)$$

Powers check

$$\forall i \in \mathbb{Z}^+, i \leq 2^{12}: \quad e([x.\,s]_1^i, [x.\,s]_2) \stackrel{?}{=} e([(x.\,s)^{i+1}]_1, g_1)$$

Why KZG?

	Merke (SHA 256)	Merkle (Arithmetic)	**KZG**	IPA
Verification inside SNARK	Prohibitive	Doable	**Cheap**	Cheap
Proof Verification (& low deg. proofs)	Hard	SNARKs needed for easy verify	**Easy**	O(N) time
Proof generation in O(n*log(n))	Yes	Yes	**Yes**	Only with larger proof-size
Commit linearity	No	No	**Yes**	Yes
Proof linearity	No	No	**Yes**	No
Security assumption	Very safe	Immature hash functions	**Trusted setup + Quantum vuln.**	Quantum vuln.

Modified from Vitalik's post: ethresear.ch/t/13863