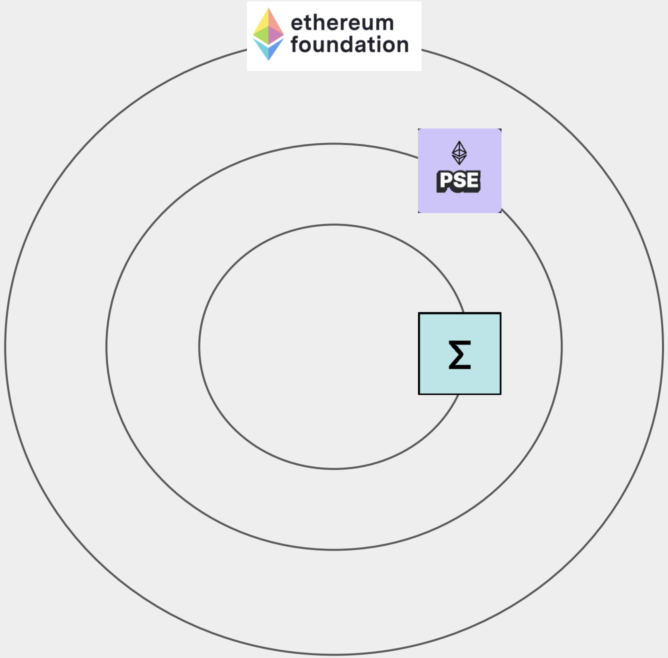
Summa

Barcelona - 15/4/2022

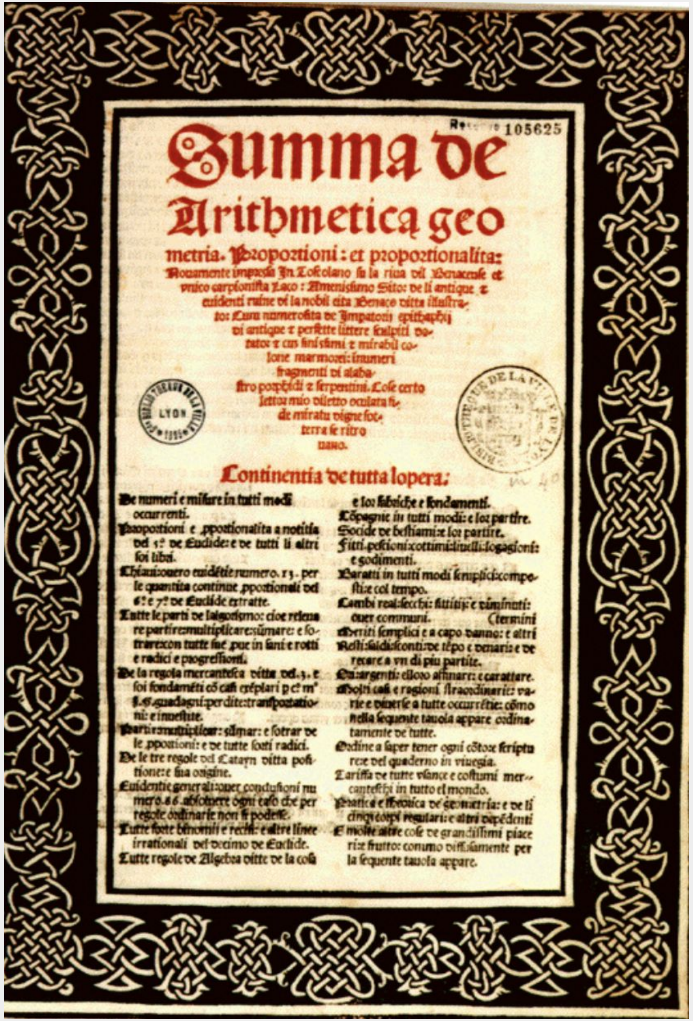
Σ

Zero-Knowledge Proof of Solvency for CEXs

enrico.eth


ethereum foundation
PSE
Σ

EVCLIDES
LI. R. LV.
BVR



Summa de Arithmetica geometria. Proportioni: et proportionalita:

Nouamente impressa In Toscolano ſu la riua dil Benacenſe et vnico carpioniſta Laco: Ameniſsimo Sito: de li antique ꝛ euidenti ruine di la nobil cita Benaco ditta illuſtrato: Cum numeroſita de Imperatori epithaphij di antique ꝛ perfette littere ſculpiti dotato ꝛ cum finiſsimi ꝛ mirabili colone marmorei: inumeri fragmenti di alabaſtro porphidi ꝛ ſerpentini. Coſe certo lettor mio diletto oculata fide miratu digne ſotterra ſe ritrouano.

Continentia de tutta lopera:

De numeri e miſure in tutti modi occurrenti.

Proportioni e ꝓportionalita a notitia del 5° de Euclide: e de tutti li altri ſoi libri.

Chiaui: ouero euidētie numero. 13. per le quantita continue ꝓportionali del 6° e 7° de Euclide extratte.

Tutte le parti de lalgoriſmo: cioe releuare partire: multiplicare: ſummare: e ſotrare con tutte ſue ꝓue in ſani e rotti e radici e progreſsioni.

De la regola mercanteſca ditta del. 3. e ſoi fondamēti cō caſi exēplari p c° m° ſ. 6. guadagni: perdite: tranſportationi: e inueſtite.

Partire: multiplicar: ſūmar: e ſotrar de le ꝓportioni: e de tutte ſorti radici.

De le tre regole del Catayn ditta poſitione: e ſua origine.

Euidentie generali: ouer concluſioni numero. 66. abſoluere ogni caſo che per regole ordinarie non ſi podeſſe.

Tutte ſorte binomij e recīſi: e altre linee irrationali del decimo de Euclide.

Tutte regole de Algebra ditte de la coſa e lor fabriche e fondamenti.

Cōpagnie in tutti modi: e lor partire.

Socide de beſtiami: e lor partire.

Fitti. peſcioni: cottimi: liuelli: logagioni: e godimenti.

Baratti in tutti modi ſemplici: compoſti: e col tempo.

Cambi real: ſecchi: fittitij: e diminuti: ouer communi. (Termini

Meriti ſemplici e a capo danno: e altri

Reſti: ſaldi: ſconti: de tēpo e denari: e de recare a vn di piu partite.

Oro: argenti: elloro affinare: e carattare.

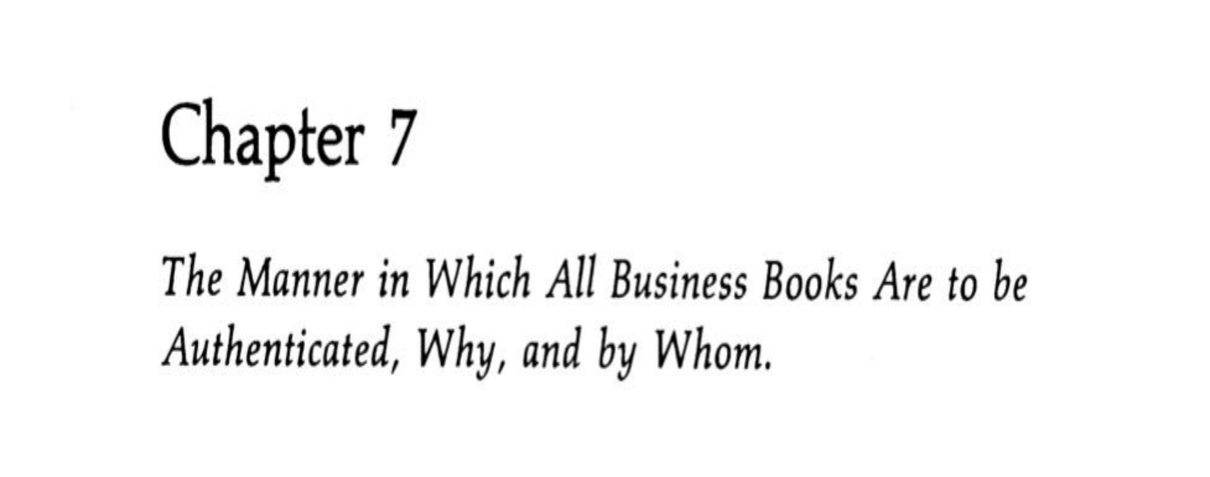
Molti caſi e ragioni ſtraordinarie: varie e diuerſe a tutte occurrētie: cōmo nella ſequente tauola appare ordinatamente de tutte.

Ordine a ſaper tener ogni cōto e ſcripture del quaderno in vinegia.

Tariffa de tutte vſance e coſtumi mercanteſchi in tutto el mondo.

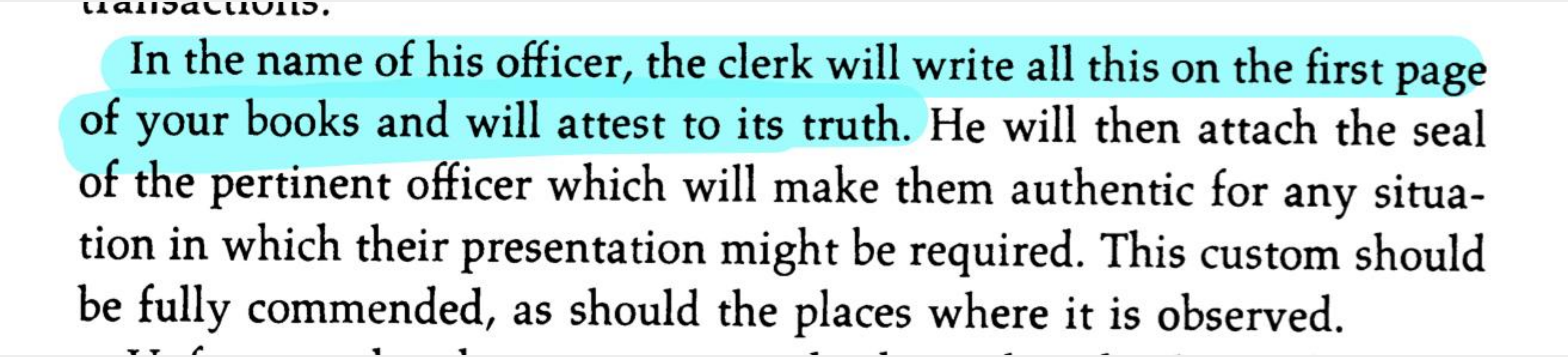
Pratica e theorica de geometria: e de li cinqꝫ corpi regulari: e altri dipēdenti

E molte altre coſe de grandiſsimi piacere e frutto: comme diffuſamente per la ſequente tauola appare.



Chapter 7

The Manner in Which All Business Books Are to be Authenticated, Why, and by Whom.



transactions.

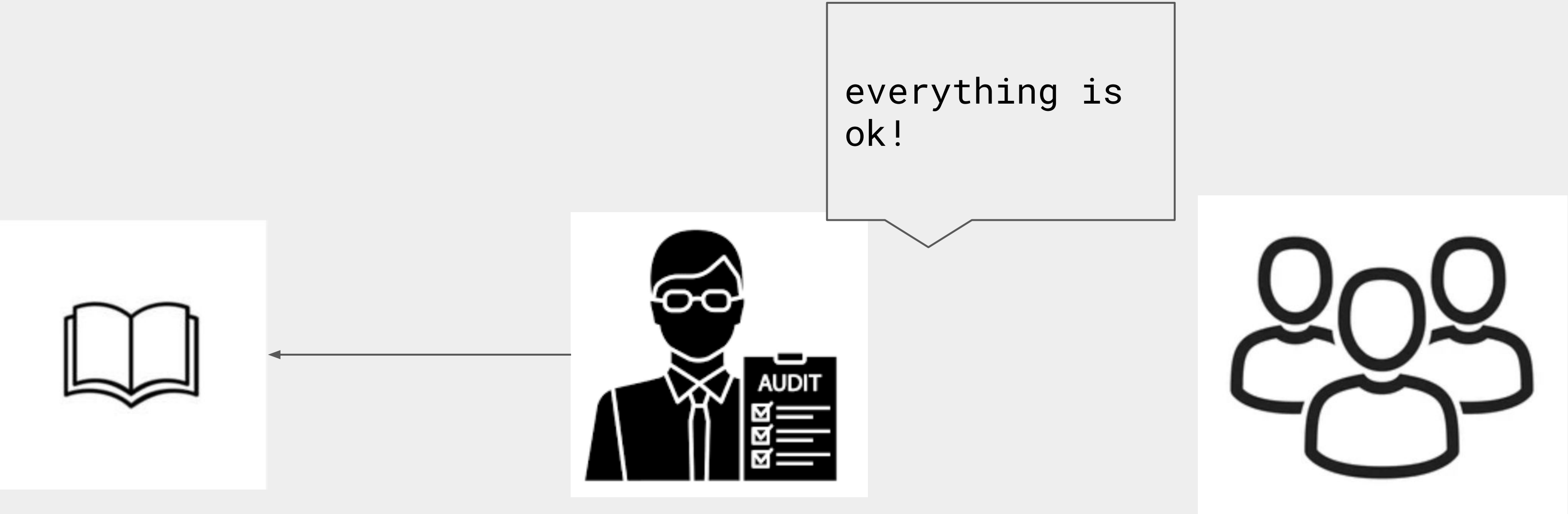
In the name of his officer, the clerk will write all this on the first page of your books and will attest to its truth. He will then attach the seal of the pertinent officer which will make them authentic for any situation in which their presentation might be required. This custom should be fully commended, as should the places where it is observed.

Book authentication

1400

2020

auditor-based book authentication

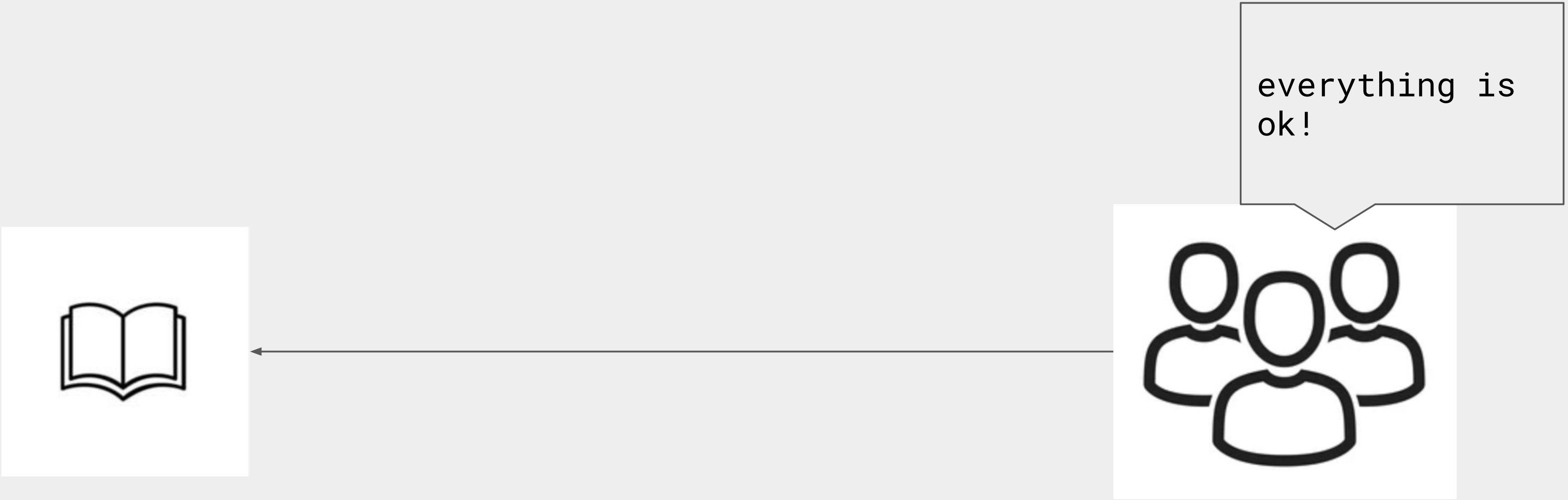


everything is ok!

...2023

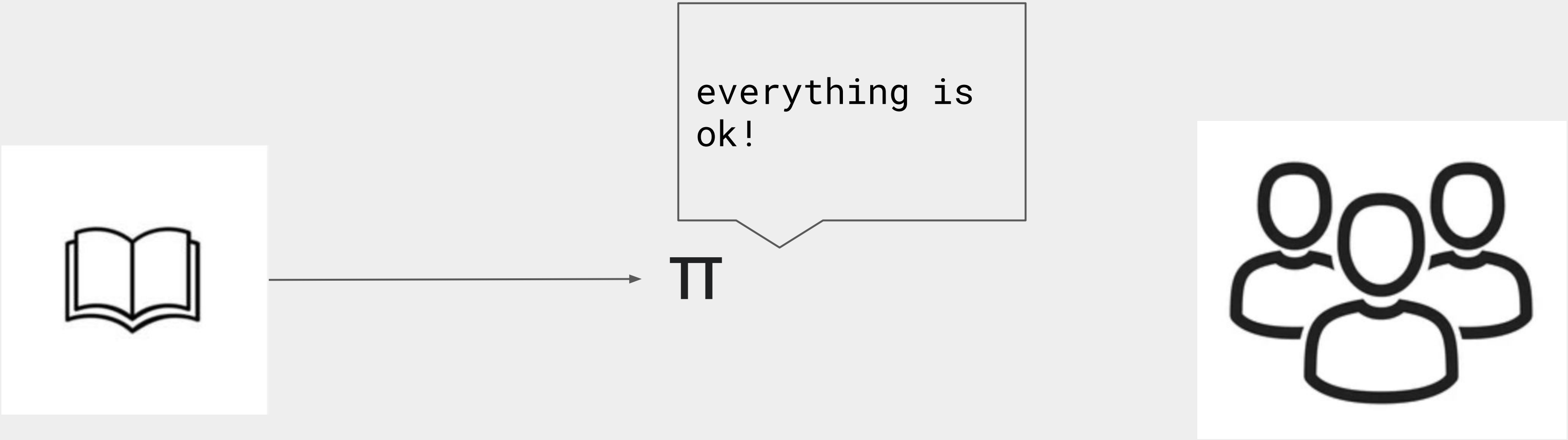
Trustless book authentication

trustless book authentication (naive approach)



everything is ok!

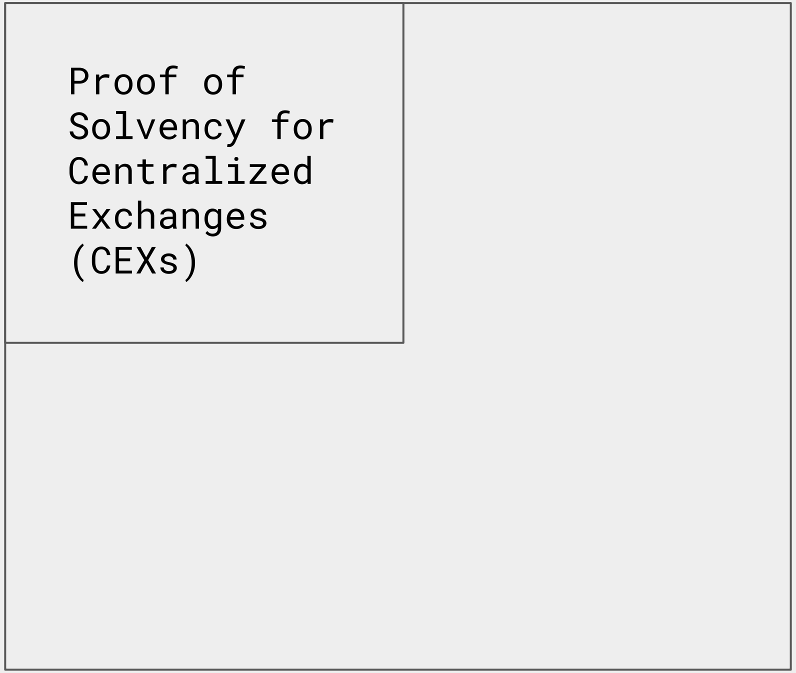
trustless book authentication
(ZK approach)



π

everything is ok!

Book Authentication



Proof of Solvency for Centralized Exchanges (CEXs)

Proof of Solvency

- Cryptographic proof that a CEX is solvent at a specific moment in time

Proof of Solvency

- Cryptographic proof that a CEX is solvent at a specific moment in time

Assets >= Liabilities

LIABILITIES

- Deposits of the users
- Denominated in ETH, BTC, USDC …
- Do not live on-chain, live in the CEX’s DB

ASSETS

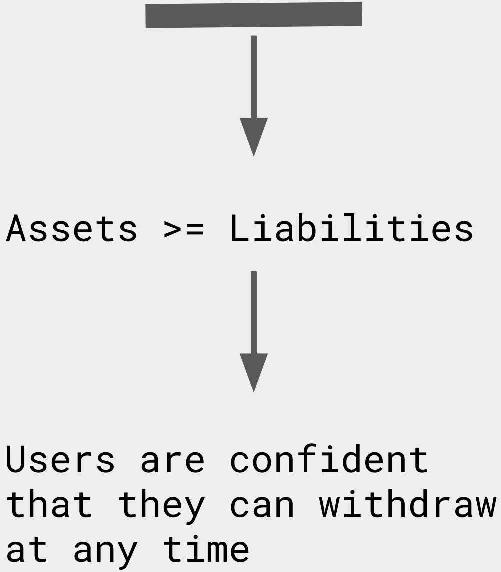
- Cryptographic assets (ETH, BTC, USDC…) controlled by the CEX
- Live on-chain
- Should map 1:1 the deposits of the users

LIABILITIES

- Deposits of the users
- Denominated in ETH, BTC, USDC …
- Do not live on-chain, live in the CEX’s DB

Proof Of Solvency

- Cryptographic proof that a CEX is solvent at a specific moment in time



Assets >= Liabilities

Users are confident
that they can withdraw
at any time

Summa: ZK Proof of Solvency

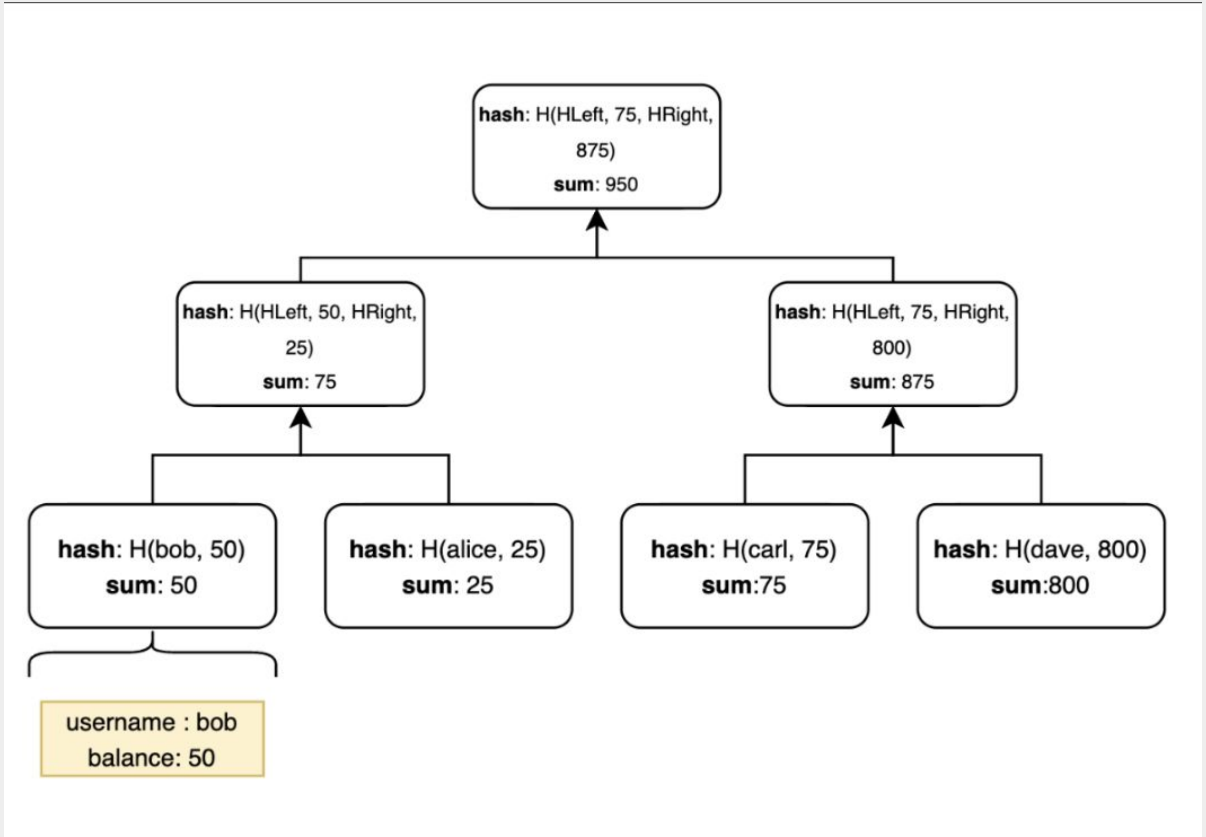
Why ZK?

ZK for computational integrity

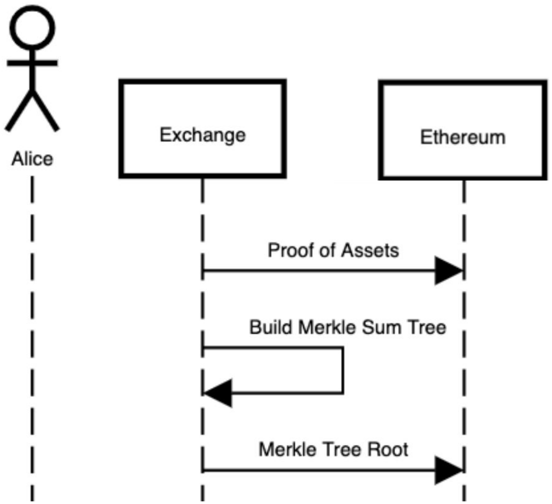
ZK for privacy

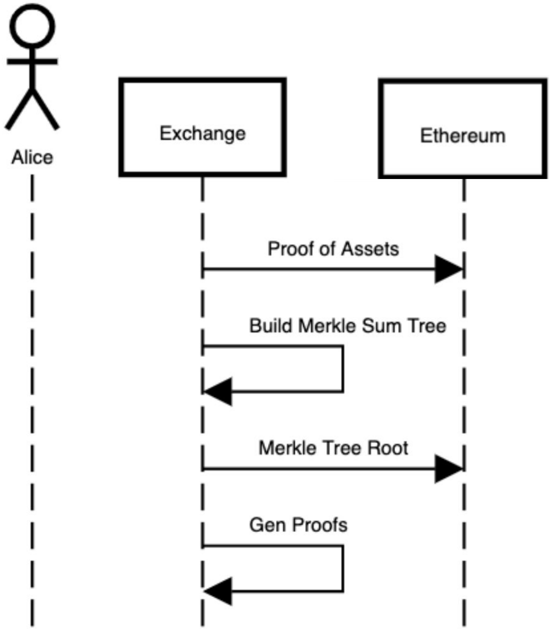
How?

Merkle Sum Tree



- The entries are the users’ data (= liabilities)
- Lives off-chain
- Only the root-hash gets published on-chain


Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root


Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root
Gen Proofs

Zk Proofs - computational integrity

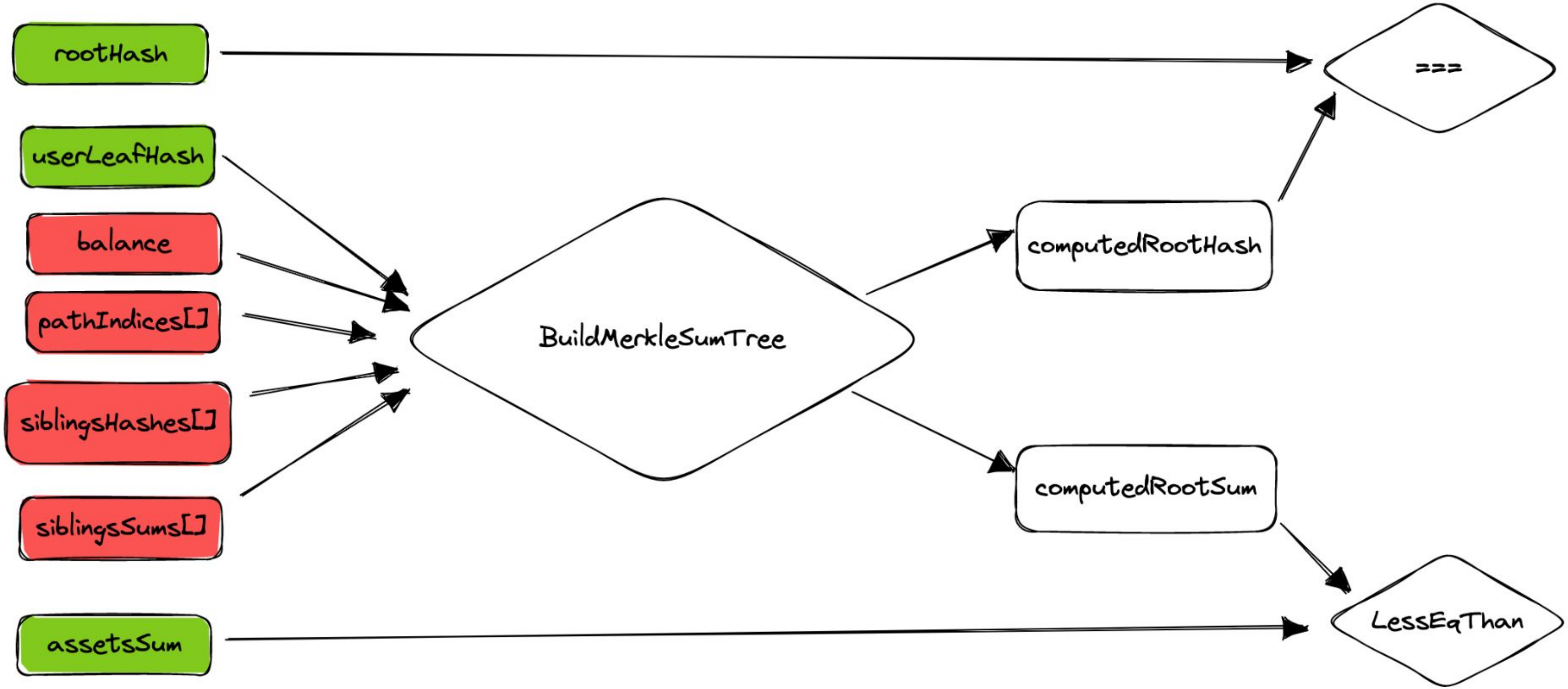
- Attest that the user is included in the Merkle Sum Tree with the correct balance
- Attest that hash of the Merkle Sum Tree matches the one committed
- Attest that sum of liabilities is Less Than the assets of the exchange (as committed in step 1)
- Attest that no sum overflow happened in the merkle sum tree computation

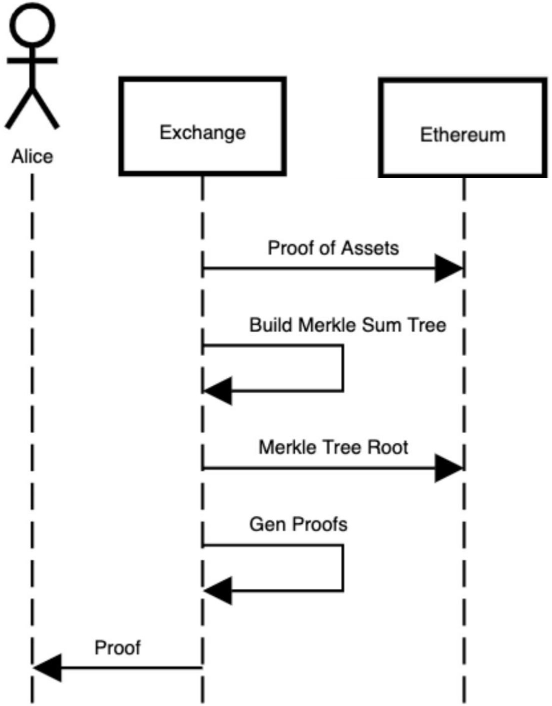
Zk Proofs - secrecy

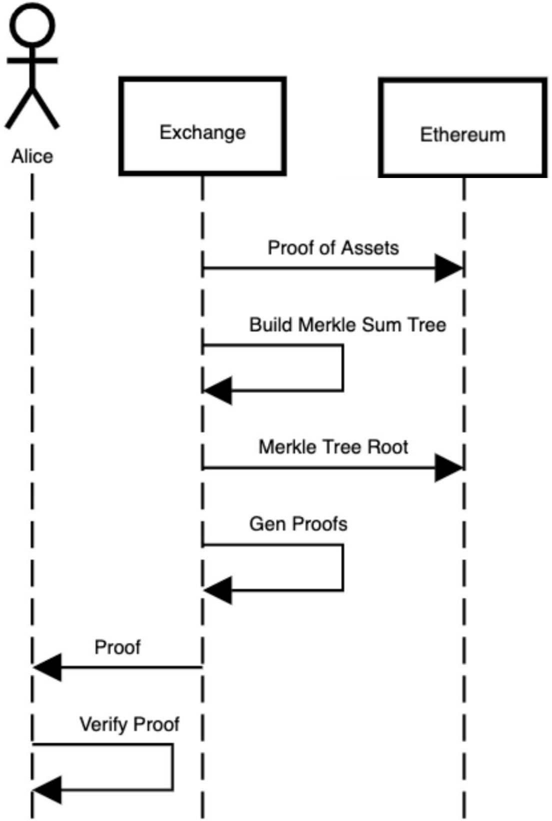
- Other users information such as their balances and usernames
- Total number of users
- Total amount of liabilities
- Total amount of assets
- The addresses of the wallets controlled by the CEX

Zk Proofs - secrecy

- Other users information such as their balances and usernames
- Total number of users
- Total amount of liabilities
- Total amount of assets (WIP)
- The addresses of the wallets controlled by the CEX (WIP)


rootHash
userLeafHash
balance
pathIndices[]
siblingsHashes[]
siblingsSums[]
assetsSum
BuildMerkleSumTree
computedRootHash
computedRootSum
===
LessEqThan

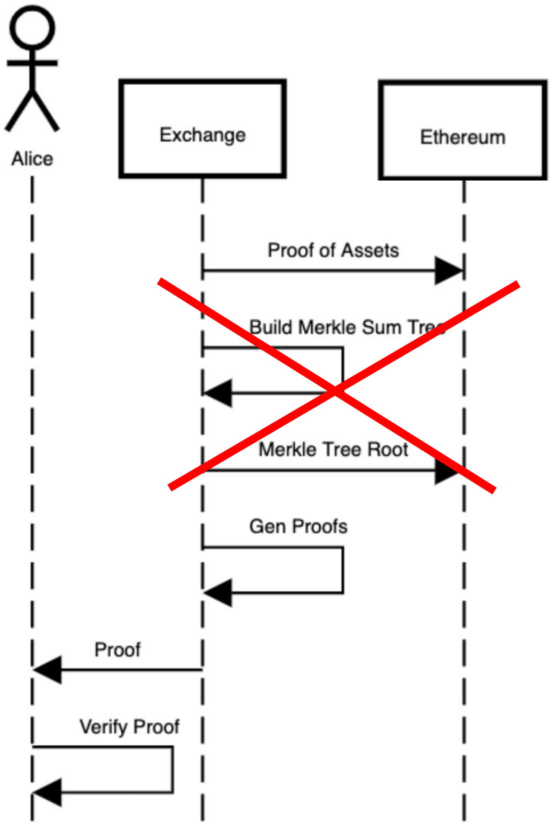

Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root
Gen Proofs
Proof


Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root
Gen Proofs
Proof
Verify Proof

Proof Verification

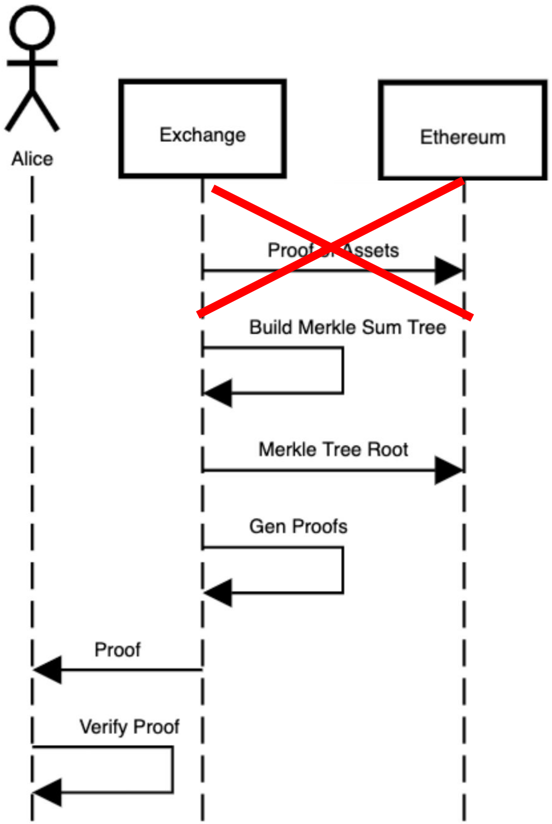
$F(\pi, \text{username}, \text{balance}, \text{assetsSum}, \text{rootHash}) = \text{yes/no}$

Next Steps


Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root
Gen Proofs
Proof
Verify Proof

KZG Polynomial Commitment

- Replace the merkle sum tree commitment with a KZG polynomial commitment
- Proving that (username, Balance) is included in that commitment


Alice
Exchange
Ethereum
Proof of Assets
Build Merkle Sum Tree
Merkle Tree Root
Gen Proofs
Proof
Verify Proof

Ethereum State Proof

- Prove that Cex own a wallet using ECDSA Signature
- Prove the balance of that wallet using account proofs from the ethereum state Trie
- Prove that this balance is >= liabilities

Open issues

- Dispute resolution
- Interactive protocol

Abstracting the protocol..

- Receive money from the users
- Have some mandate related to managing these money
- Want to be trusted by its users
- Don’t want their business information revealed to the public

- Deposit their money into an institution
- Expect some behaviour from this institution
- Don’t trust the institution

- BANK
- Insurance Companies
- Investment Funds
- Charities
- whoever has some mandate over your money..

Abstracting even more..

- Receive ~~money~~ **data** from the users
- Have some mandate related to managing these ~~money~~ **data**
- Want to be trusted by its users
- Don’t want their business information revealed to the public

- Deposit their money **data** ~~into~~ an institution
- Expect some behaviour from this institution
- Don’t trust the institution

- Social Media
- AI Companies
- whoever has some mandate over your data..

idea #2 Recursion for privacy

- Recursively verify inside a snark that:
  - an Axiom proof attesting the balance of a wallet is valid
  - the CEX controls that wallet (ECDSA signature)
  - the balance of that wallet is >= total liabilities

idea #2 Recursion for privacy

- Recursively verify inside a snark that:
  - an Axiom proof attesting the balance of a wallet is valid
  - the CEX controls that wallet (ECDSA signature)
  - the balance of that wallet is >= total liabilities

The recursed proof hides a public input from the original proof

Thank you!

∑ on github