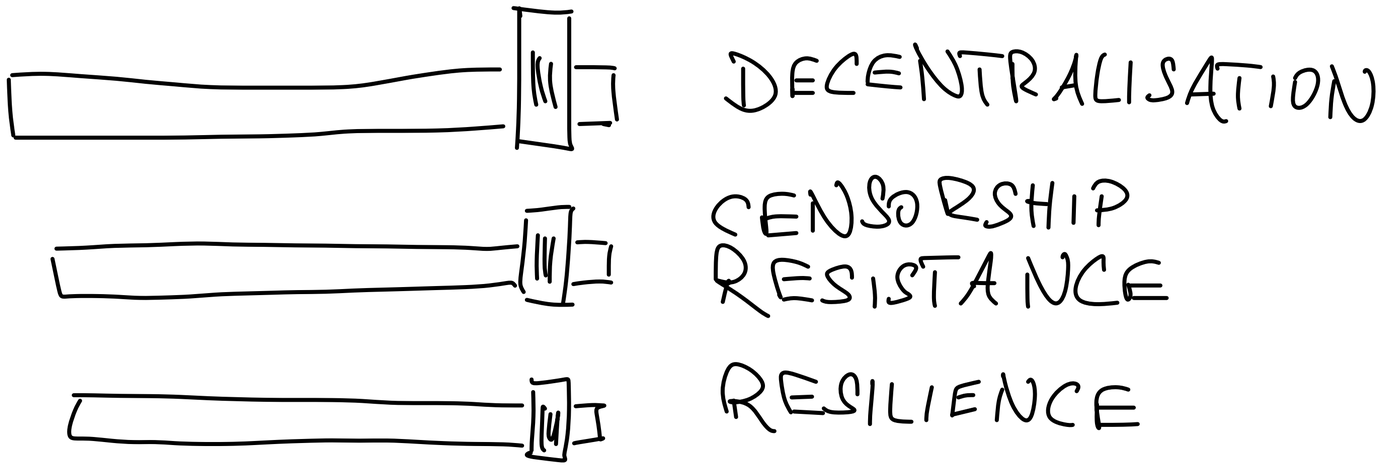
Mina Foundation

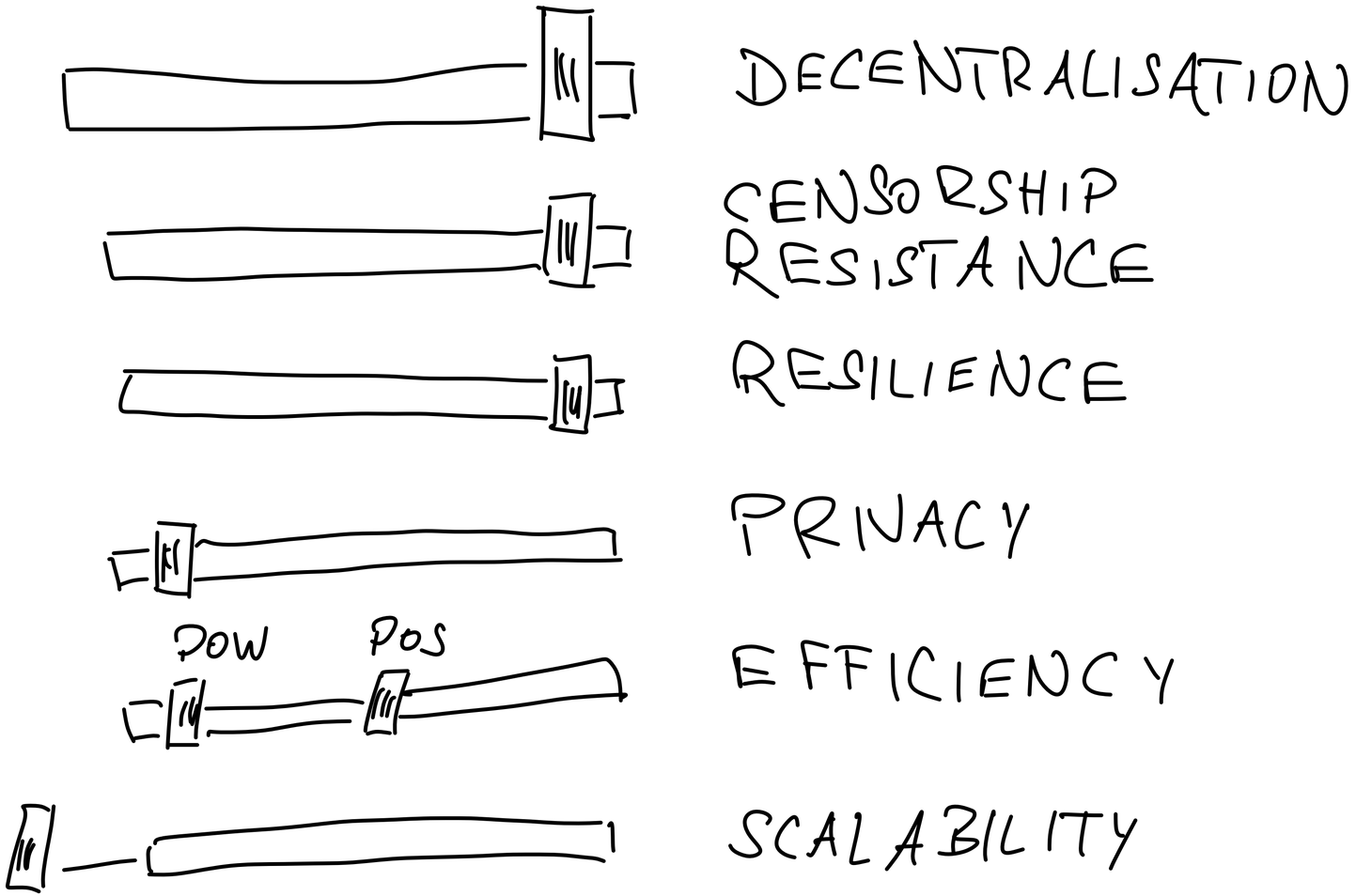
Zero-Knowledge Proofs for Trust, Privacy, and Scalability

Zero Knowledge in Blockchain

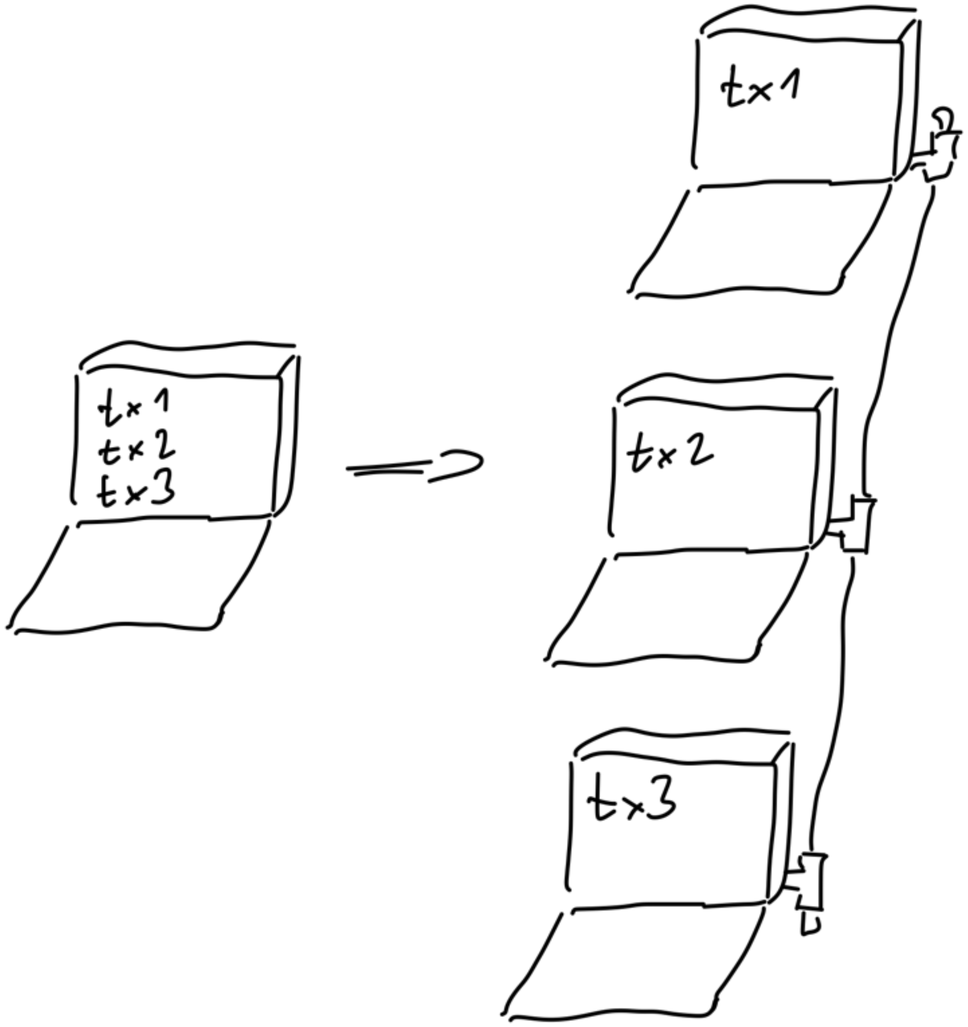
Blockchains



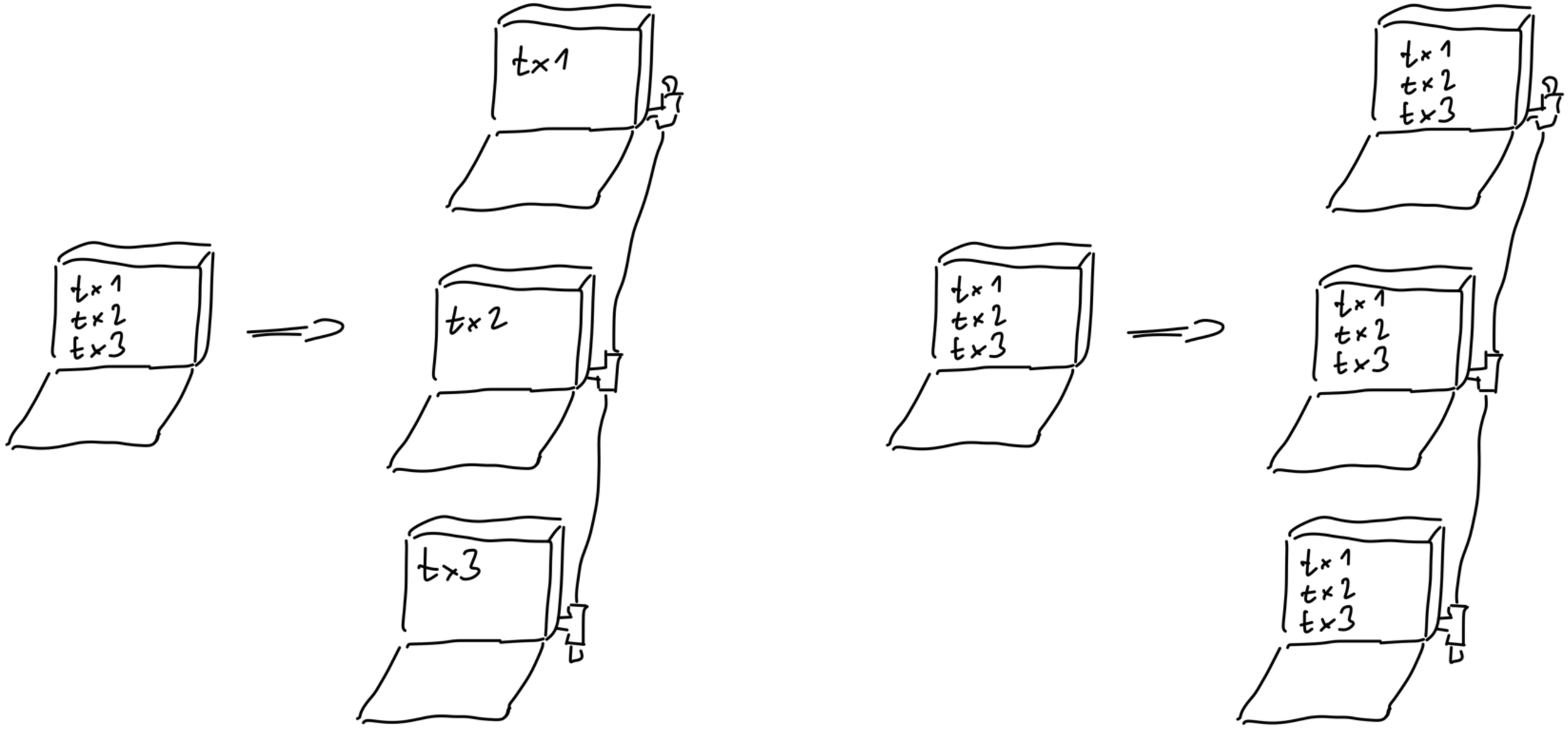
Blockchains



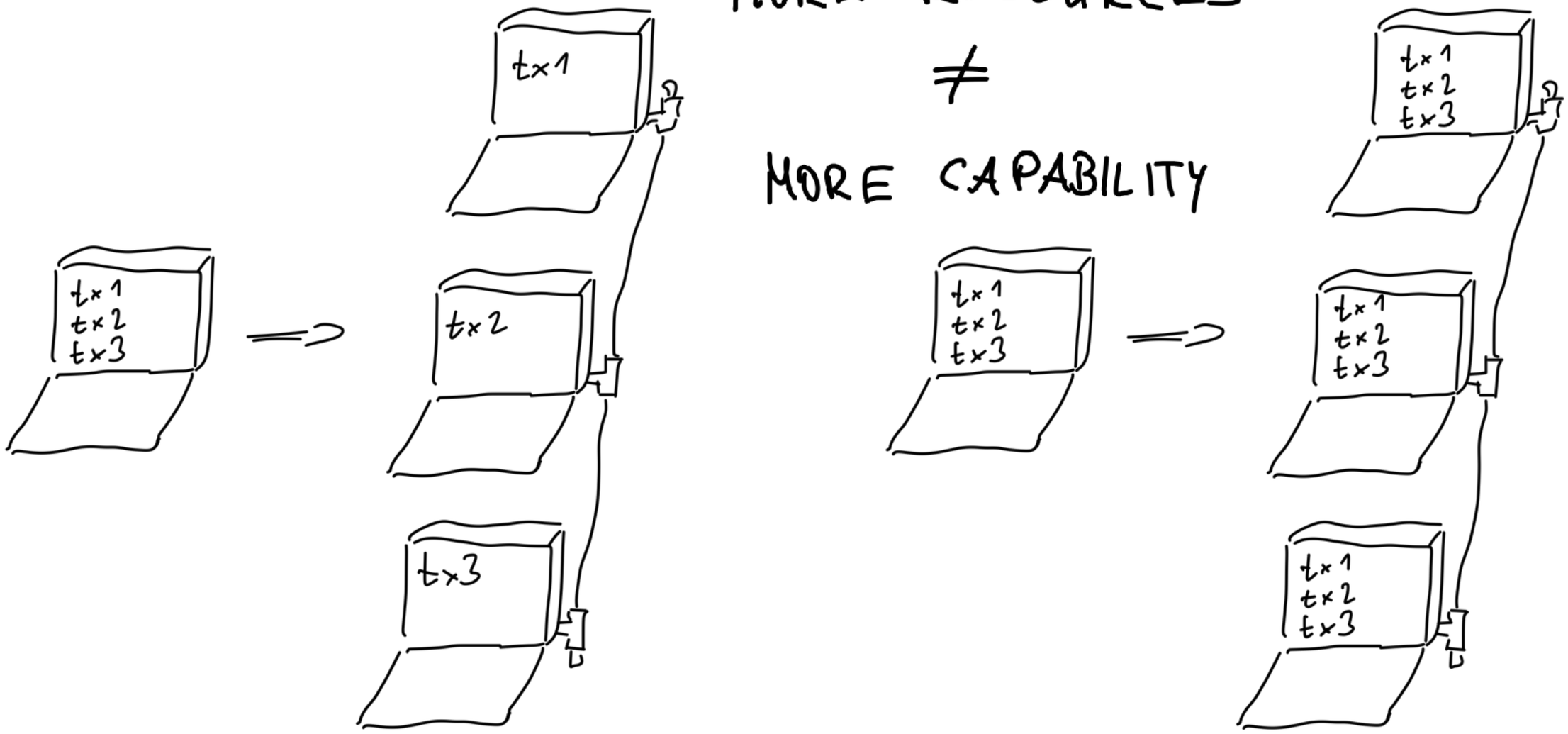
Scalability



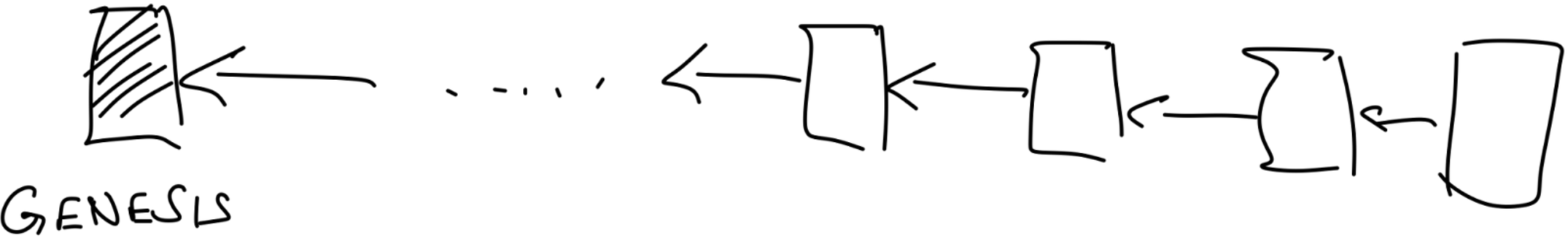
Scalability



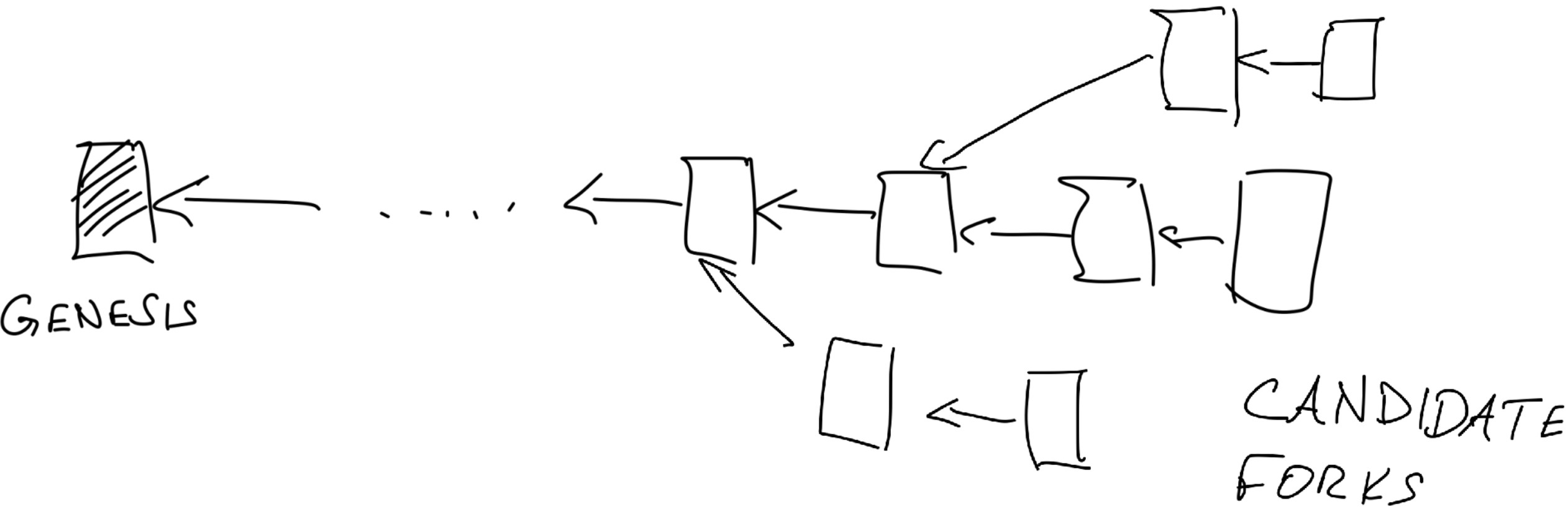
Scalability

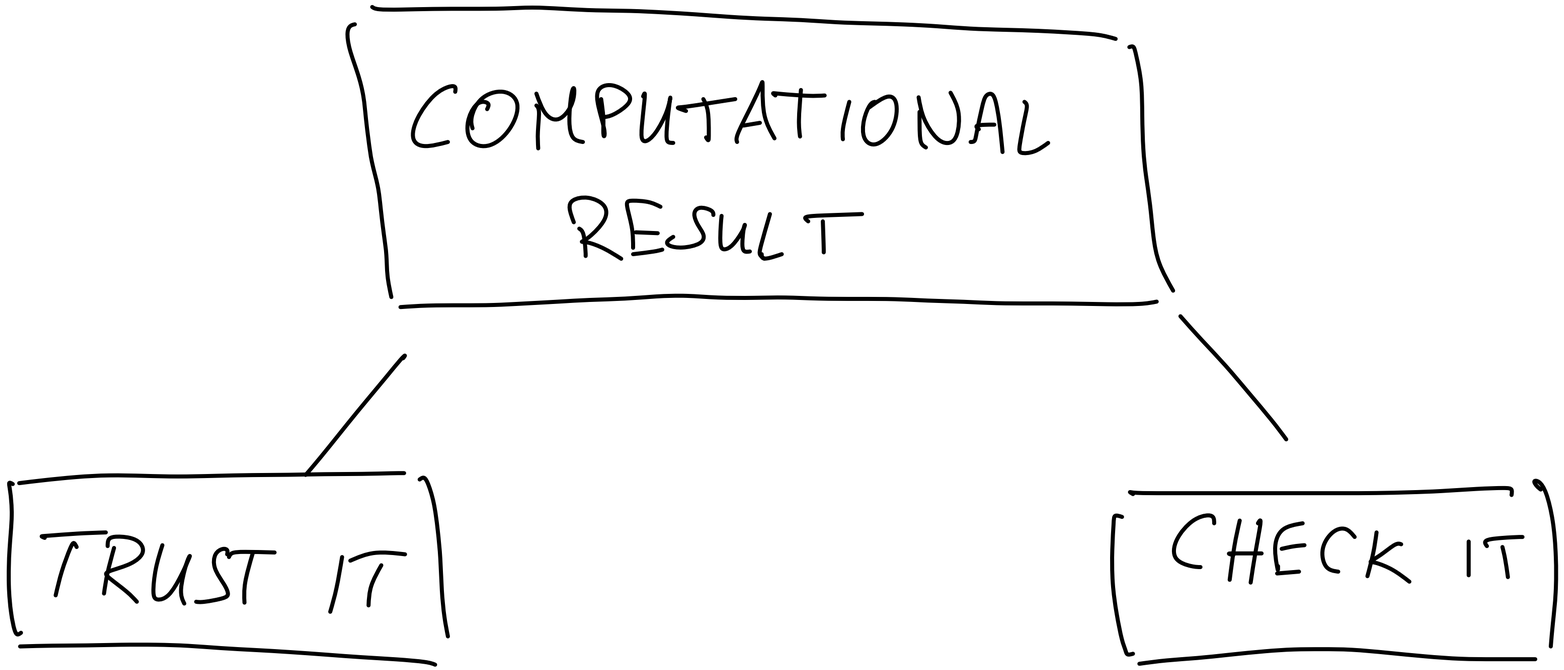


Joining the Network



Joining the Network




COMPUTATIONAL RESULT
TRUST IT
CHECK IT

zk

SNARK

ZK ZERO KNOWLEDGE

S SUCCINCT

N NON-INTERACTIVE

AR ARGUMENT of

K KNOWLEDGE

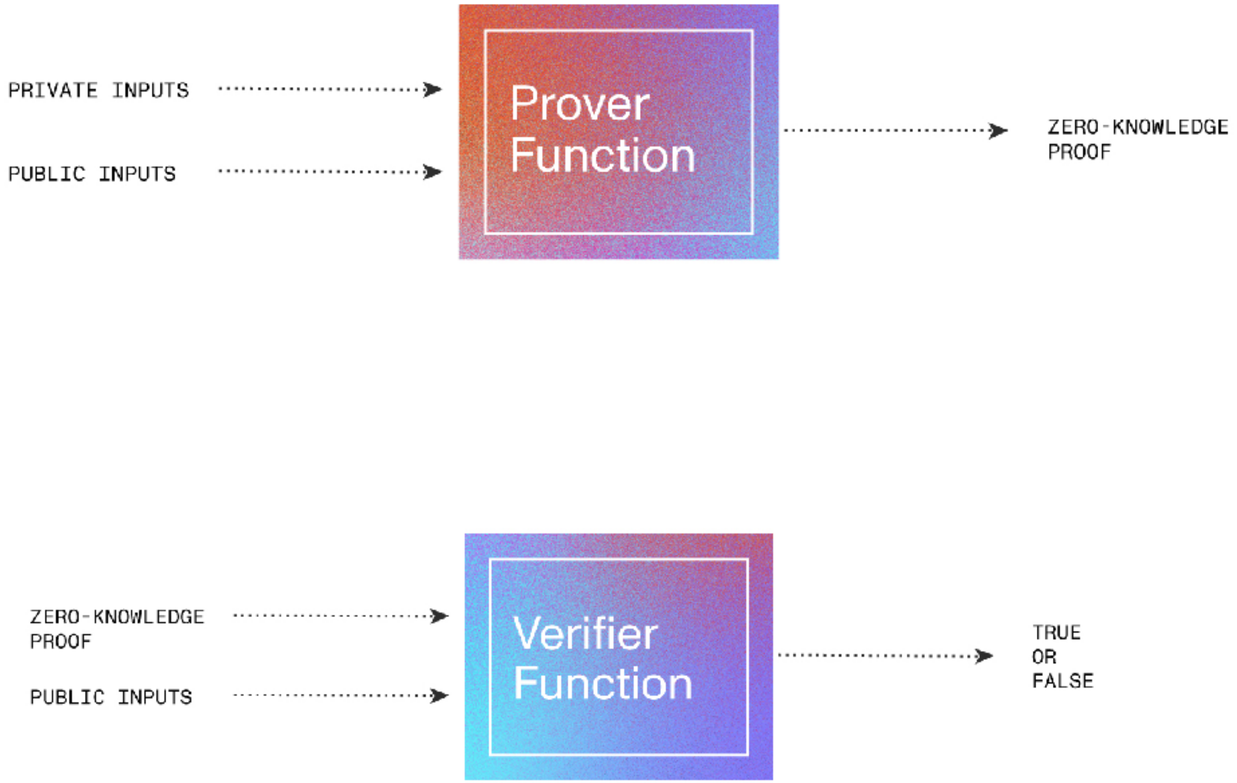
ZK SNARK

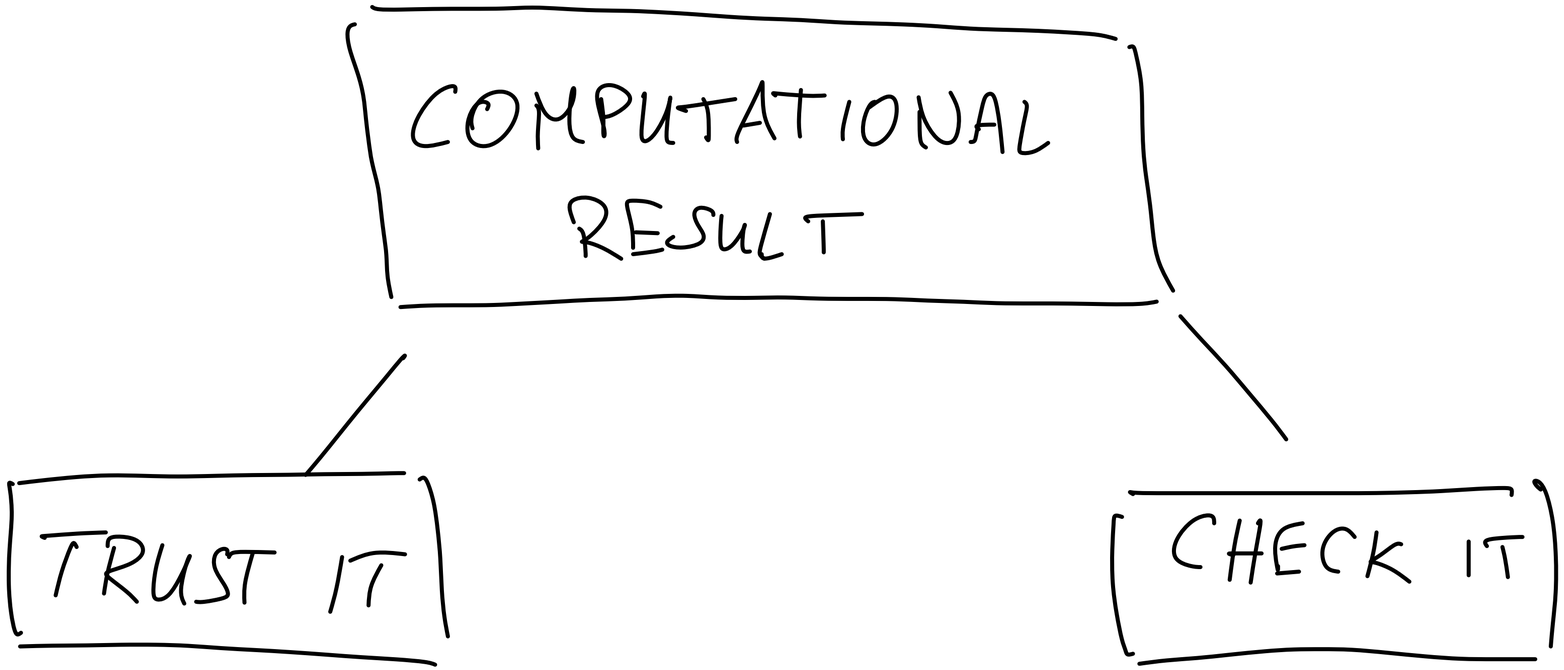
ZERO KNOWLEDGE

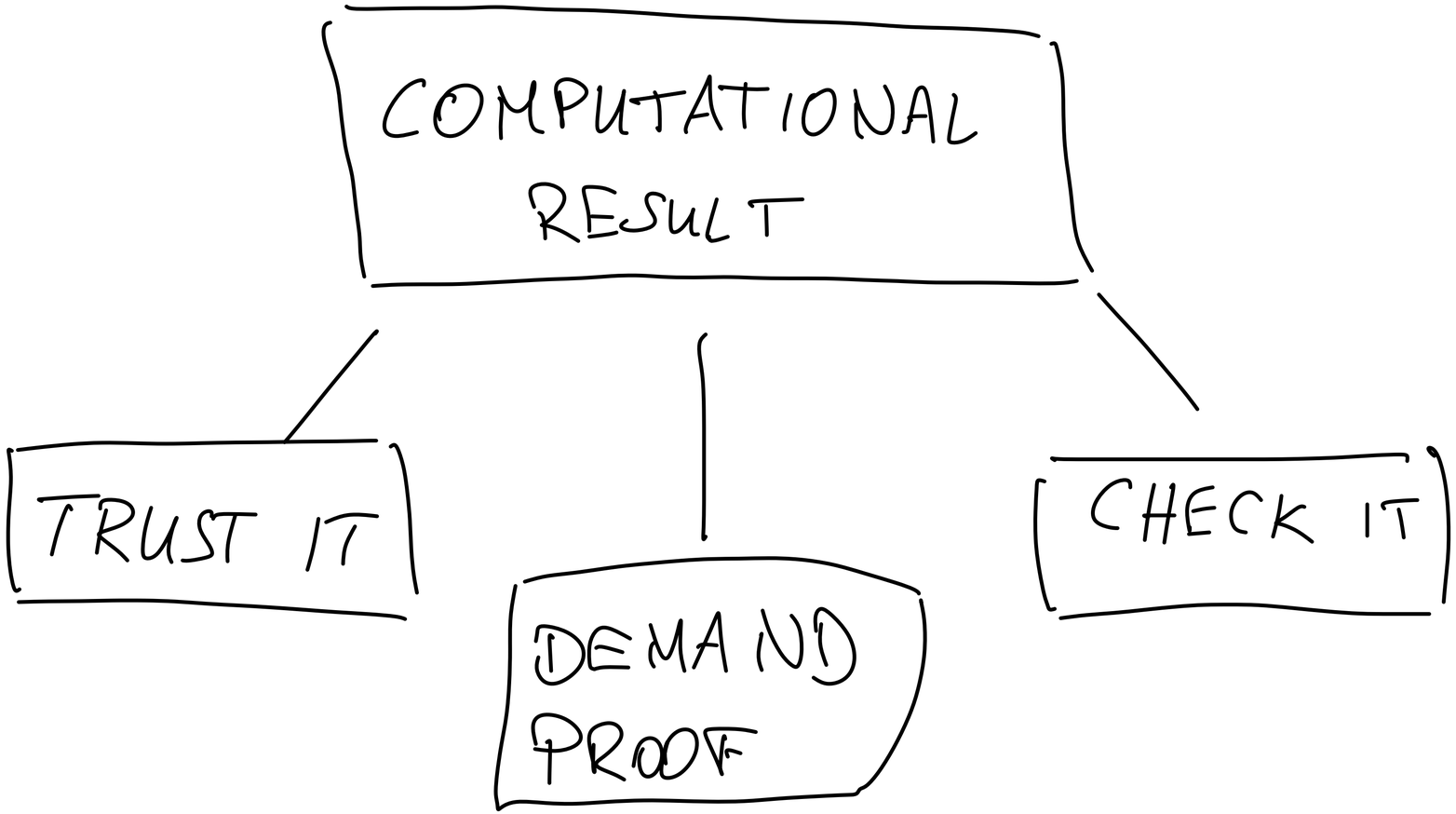
SUCCINCT

NON-INTERACTIVE

ARGUMENT of KNOWLEDGE

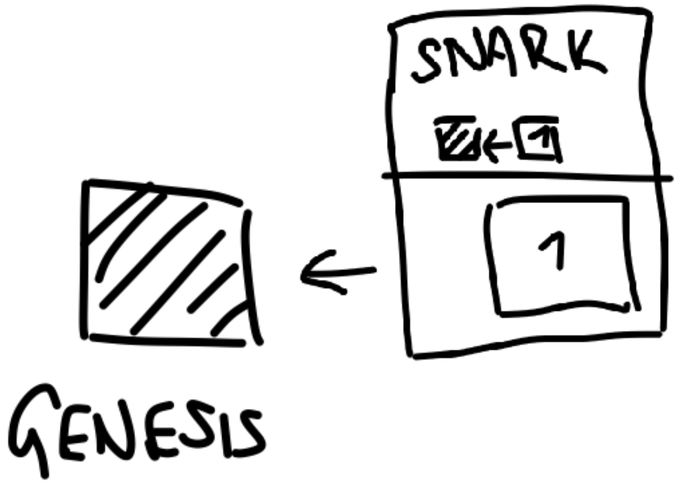



COMPUTATIONAL RESULT
TRUST IT
CHECK IT

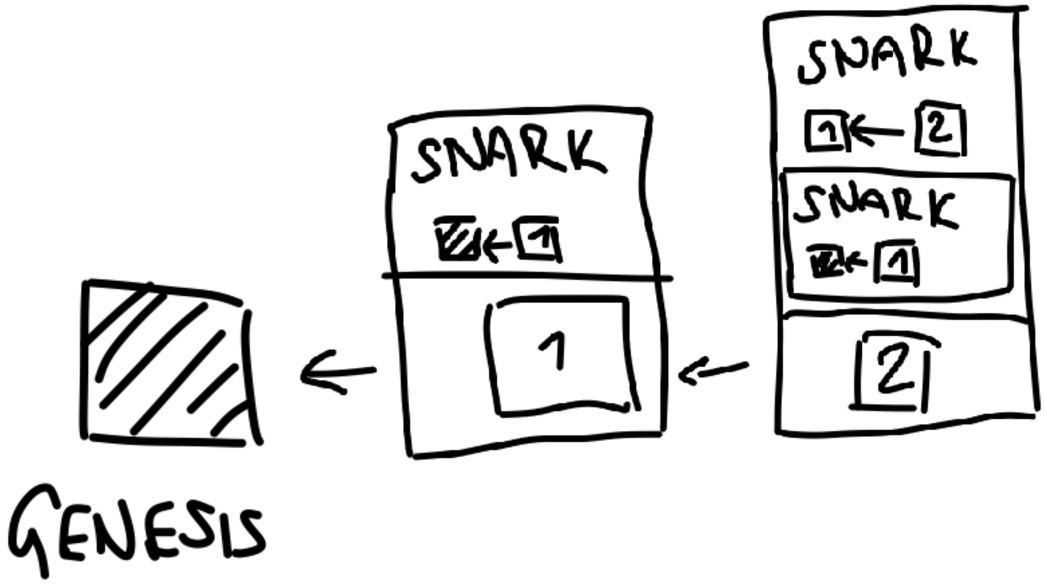

COMPUTATIONAL RESULT
TRUST IT
DEMAND PROOF
CHECK IT

Succinct Blockchain: Ouroboros Samasika

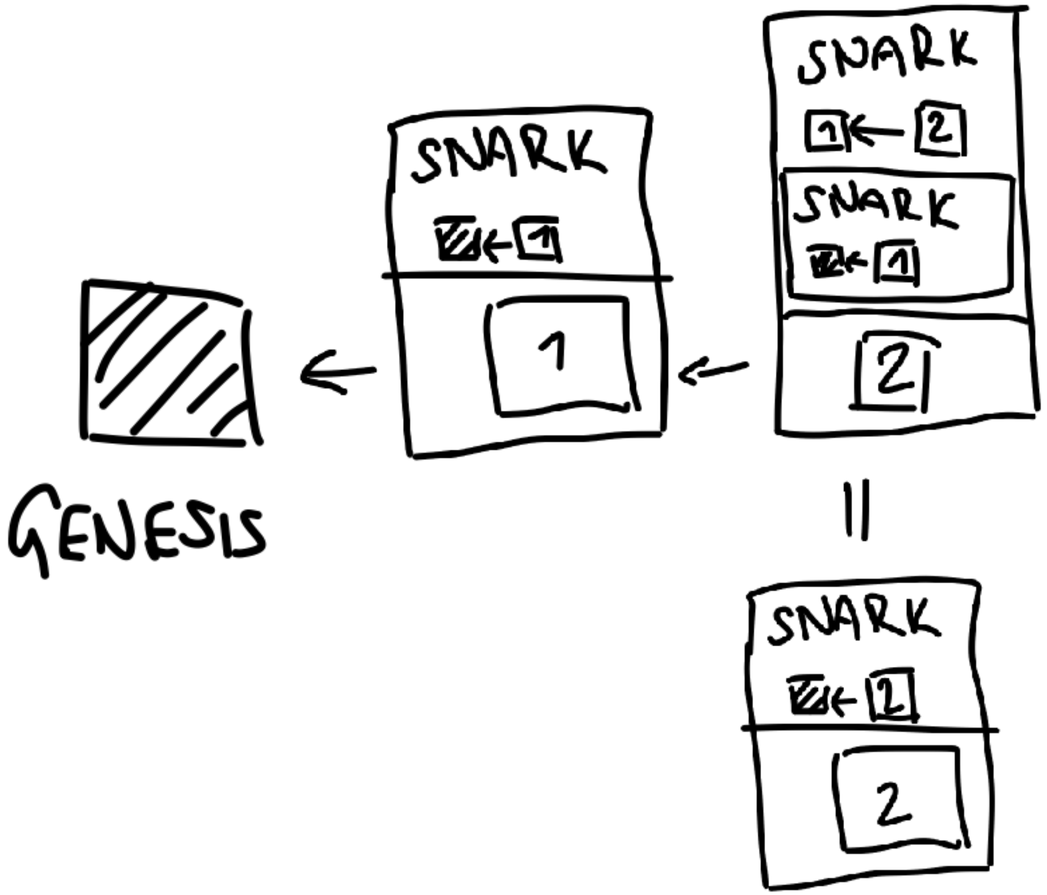
Succinct Blockchain: Ouroboros Samasika



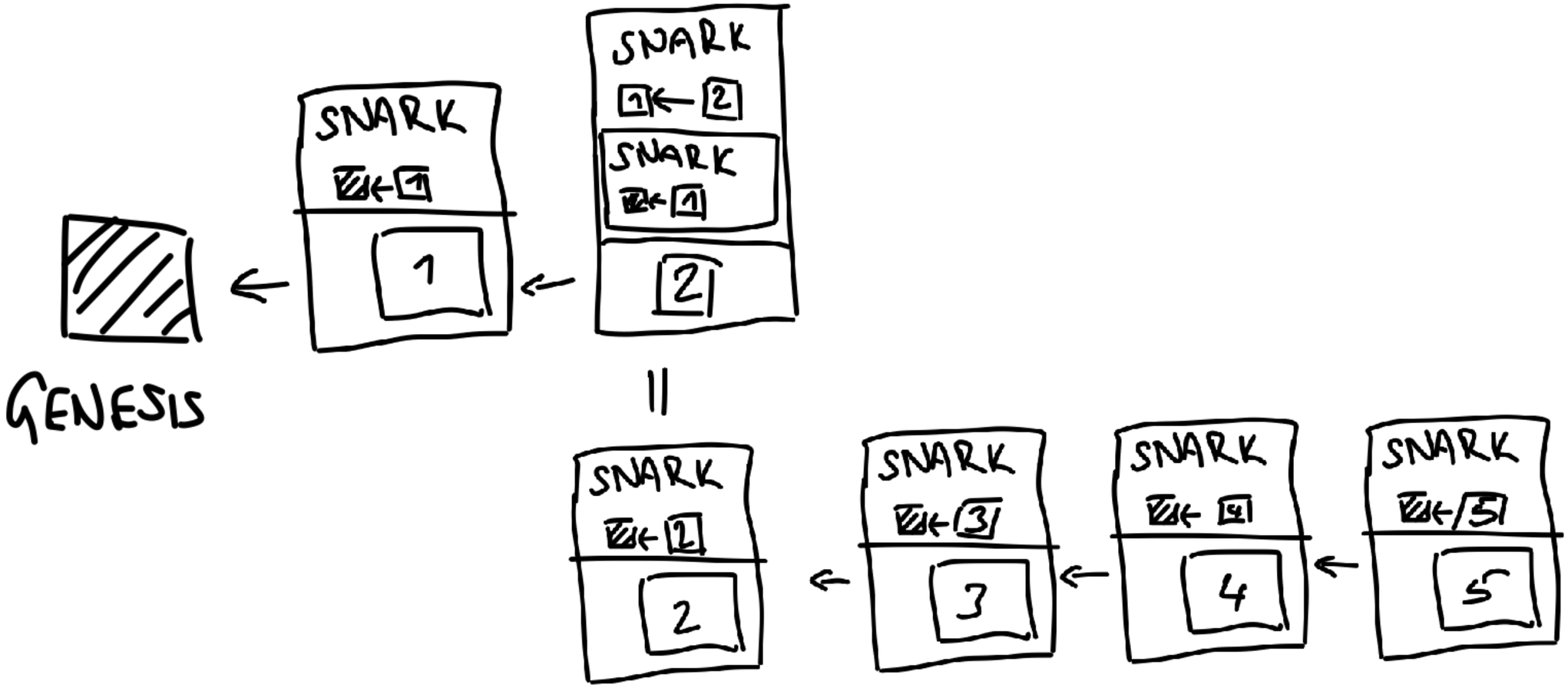
Succinct Blockchain: Ouroboros Samasika



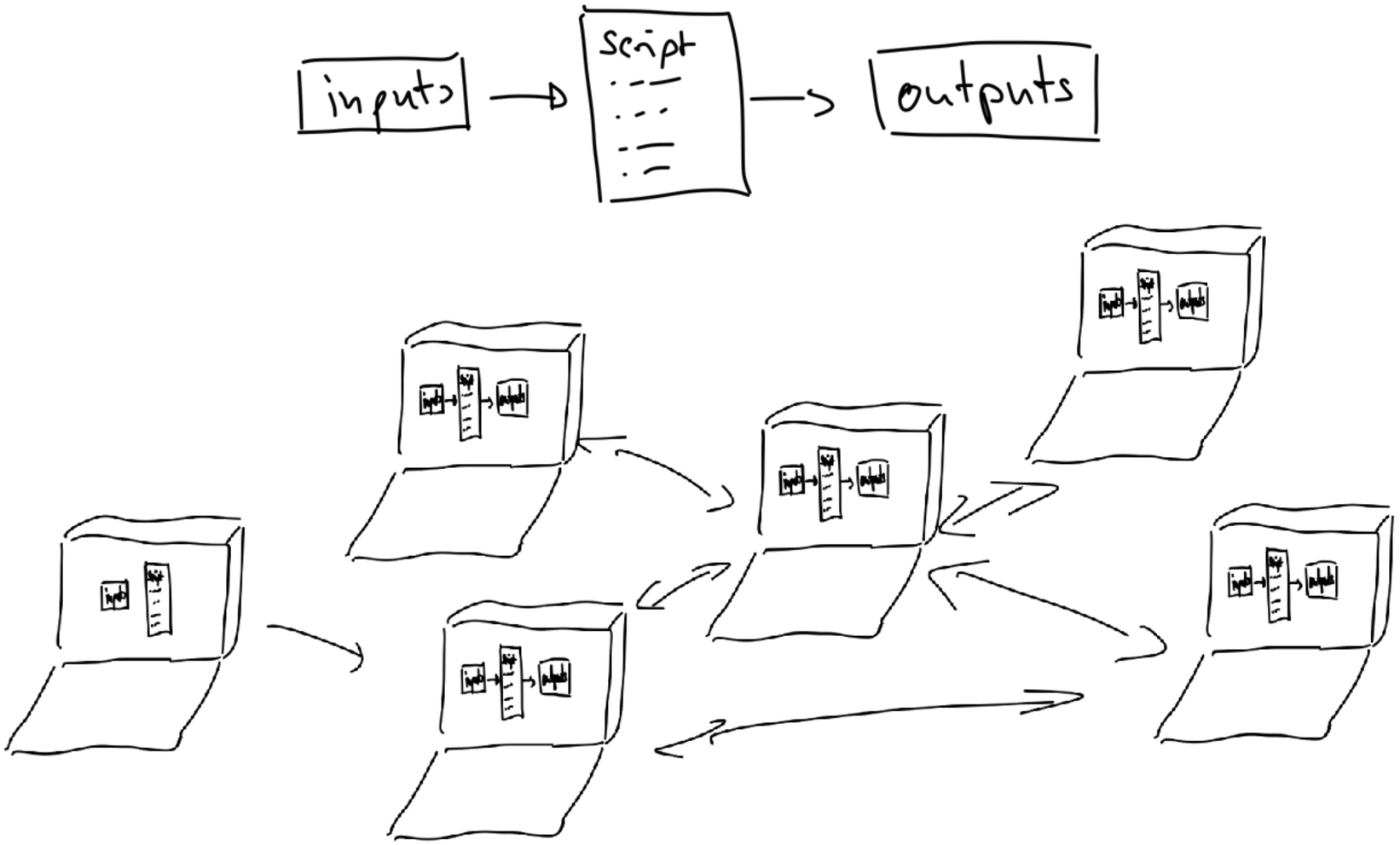
Succinct Blockchain: Ouroboros Samasika

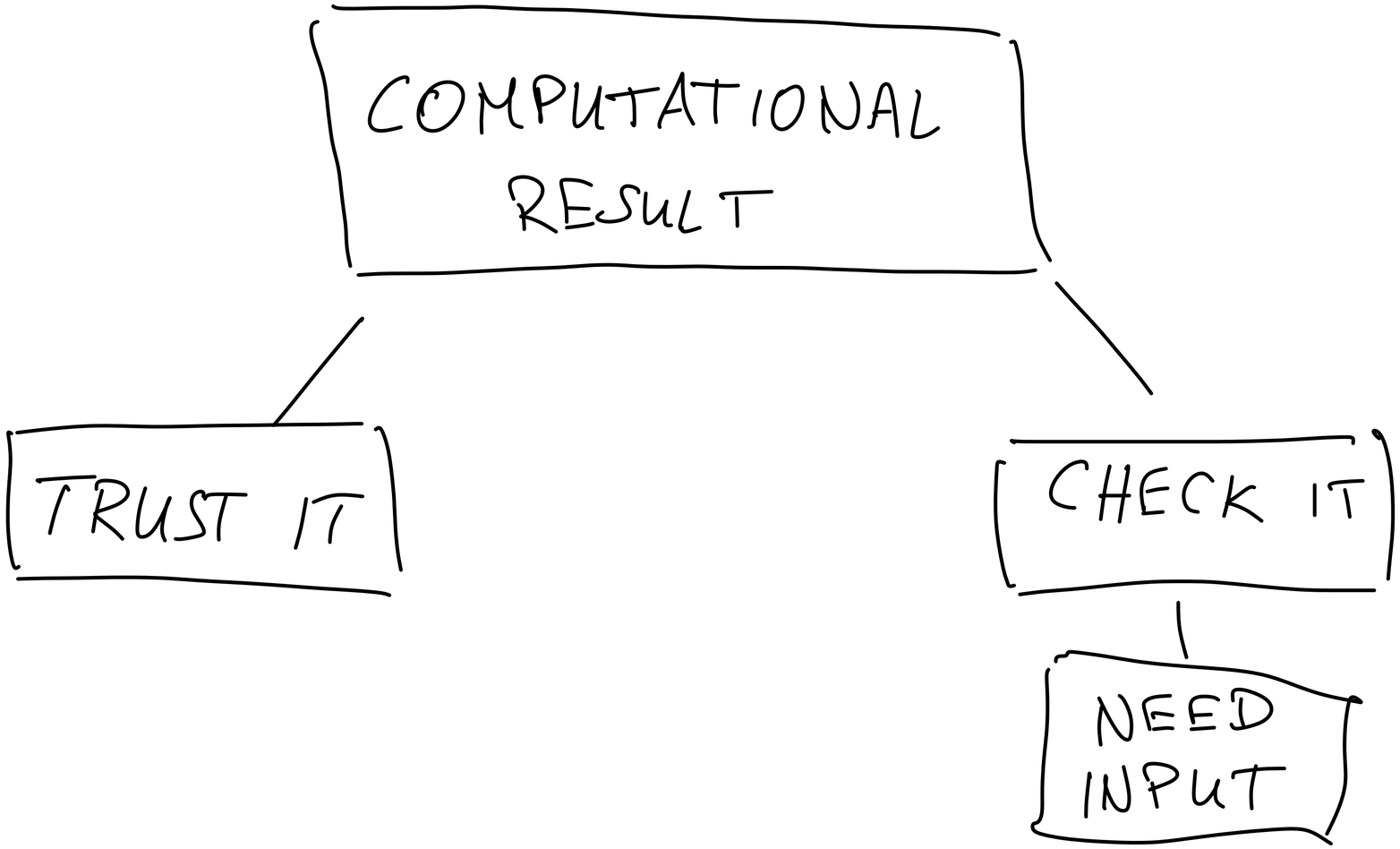


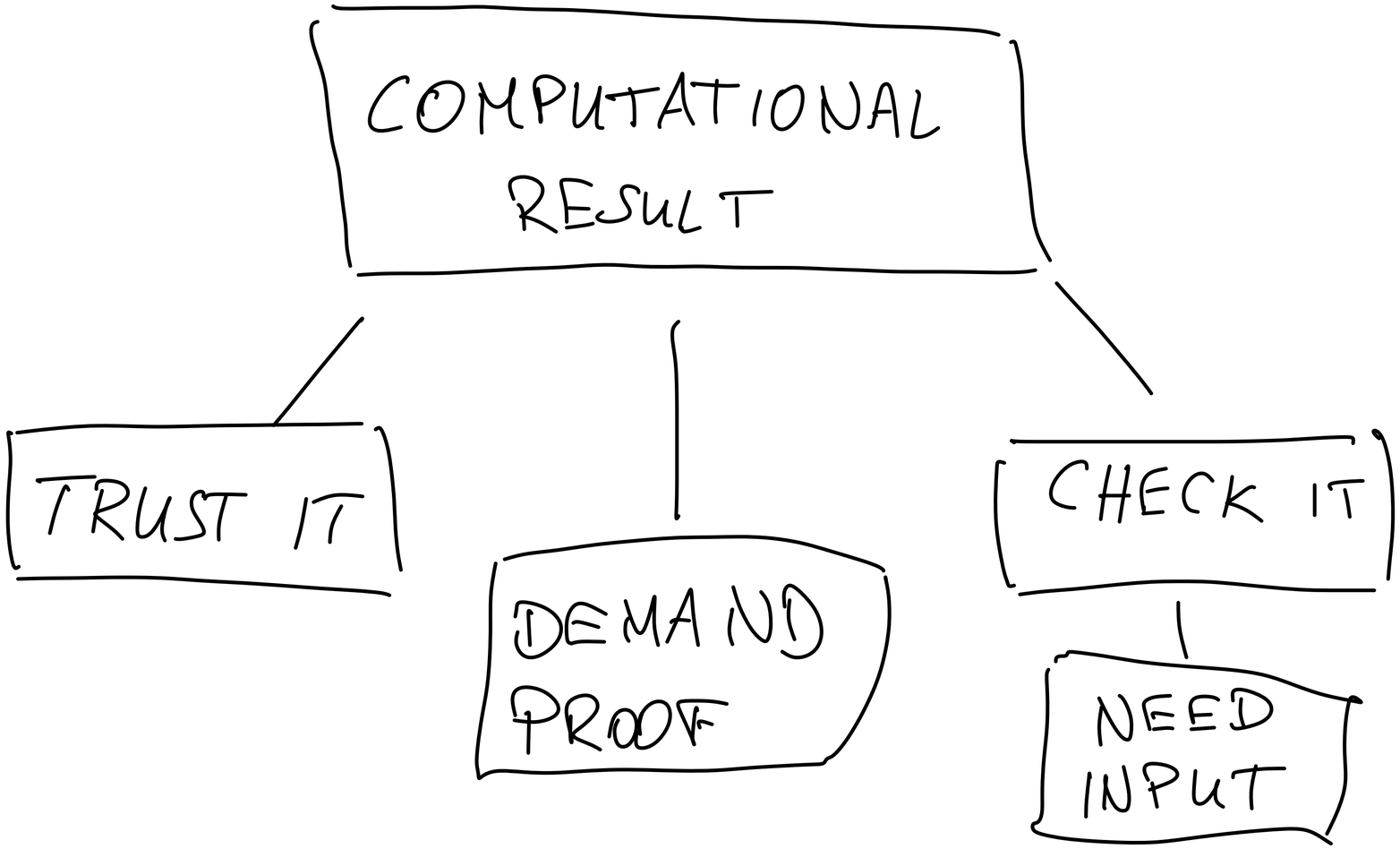
Succinct Blockchain: Ouroboros Samasika



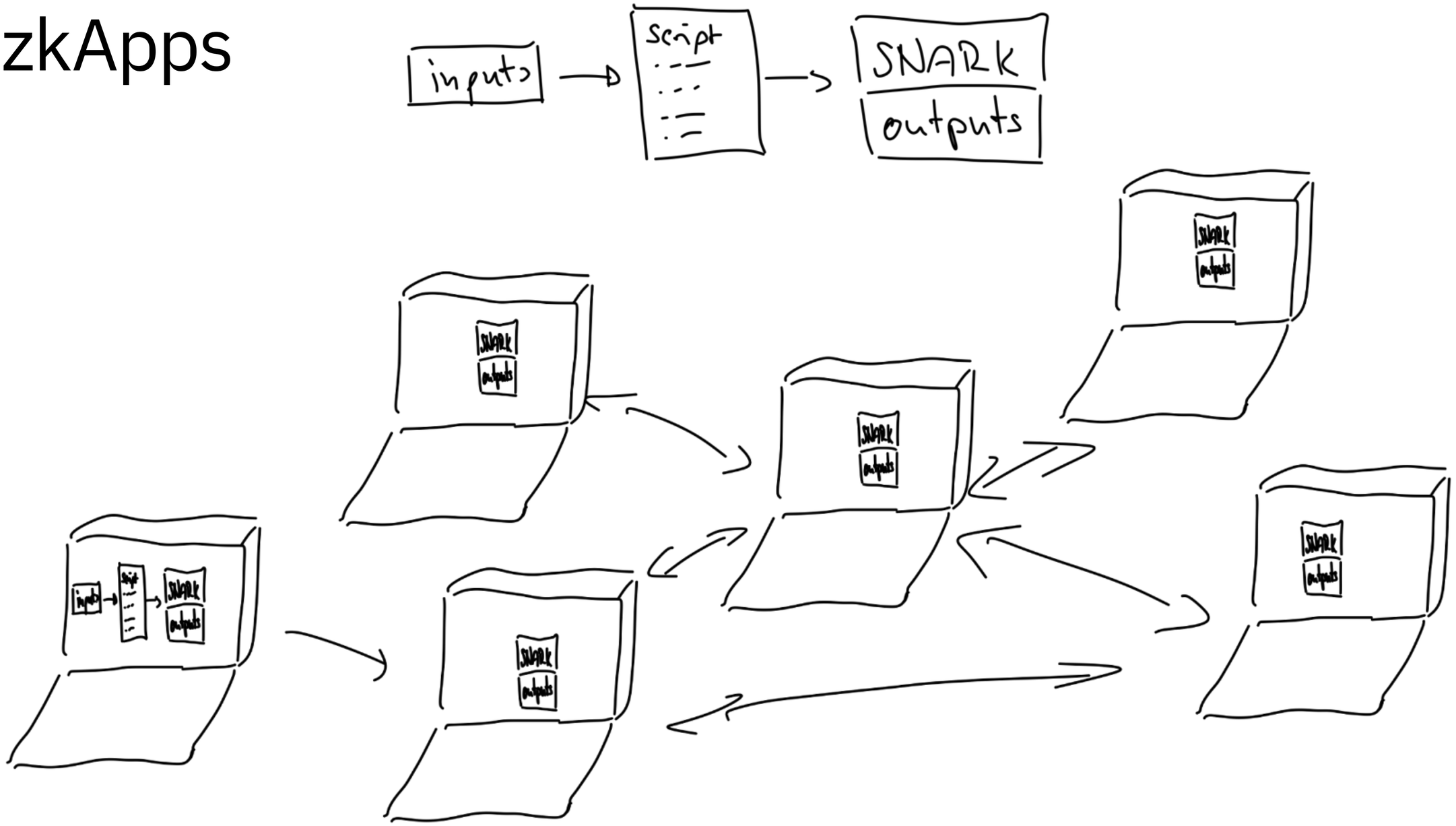
Privacy




COMPUTATIONAL RESULT
TRUST IT
CHECK IT
NEED INPUT

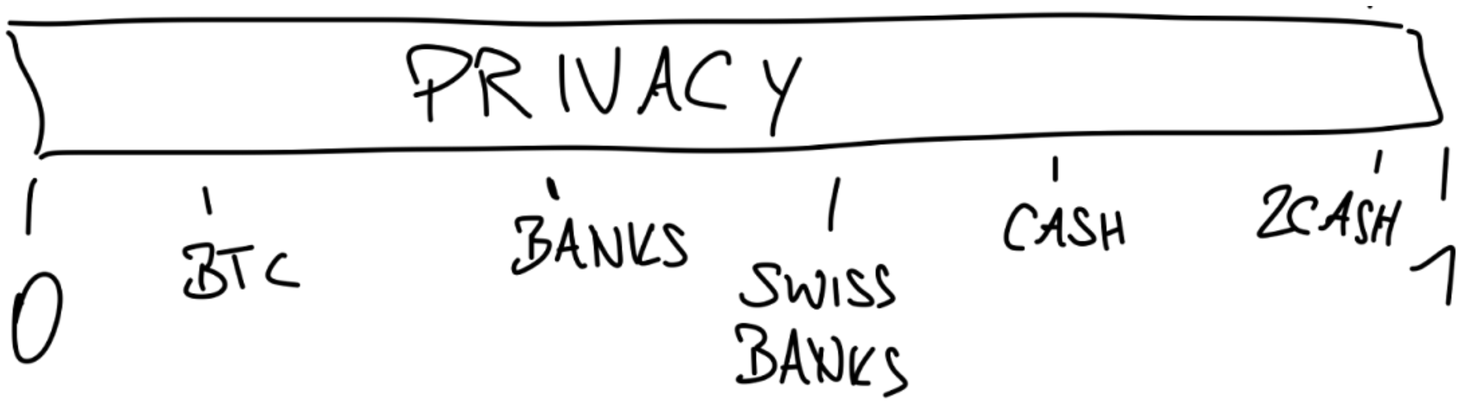

COMPUTATIONAL RESULT
TRUST IT
DEMAND PROOF
CHECK IT
NEED INPUT

zkApps

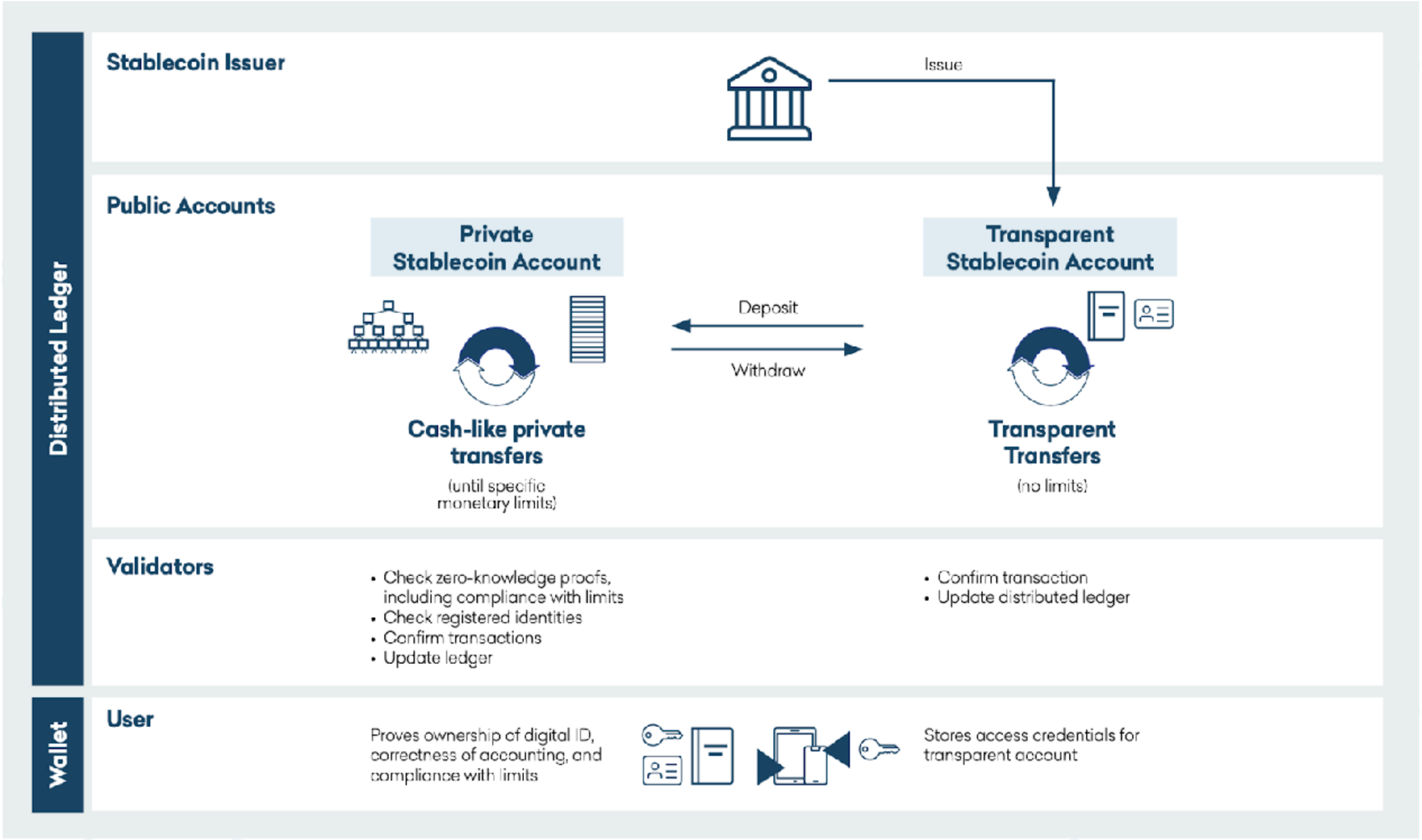


Degrees of Privacy

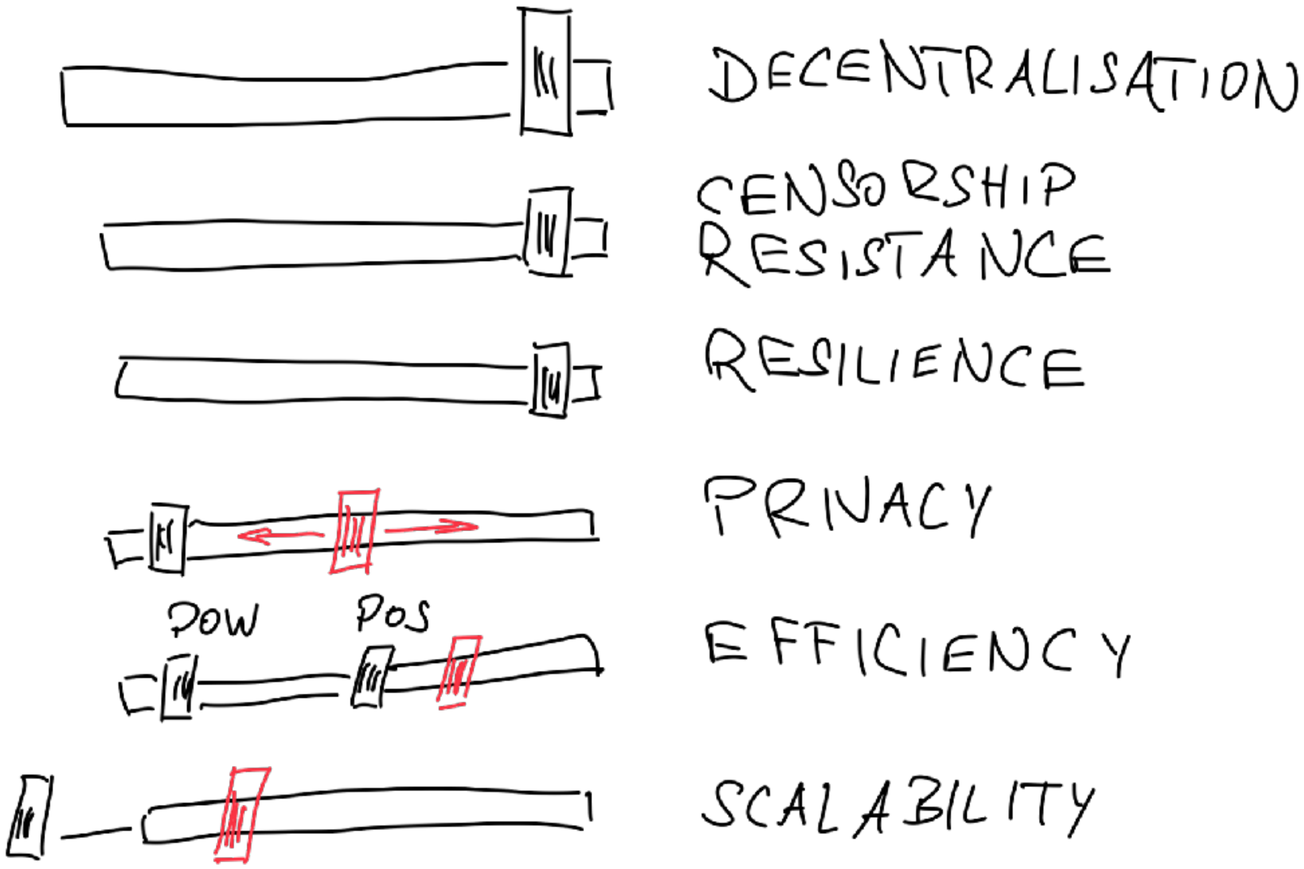
Subjective!



Programmable Privacy



Source: based on Gross et al. (2021); applied to a decentralized stablecoin system.


DECENTRALISATION
CENSORSHIP RESISTANCE
RESILIENCE
PRIVACY
POW
POS
EFFICIENCY
SCALABILITY

Zero Knowledge Beyond Blockchain

Disinformation in Social Media

Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f

Disinformation in Social Media

Das Foto ist von der Loveparade vor 21 Jahren

31.08.2020, 16:09 (CEST)

Wie nach den Demonstrationen gegen die Corona-Maßnahmen der Bundesregierung Anfang August in Berlin werden auch nach den Kundgebungen am 29. August sehr hohe und von den offiziellen Polizei-Angaben abweichende Teilnehmerzahlen verbreitet. Manche User verwenden bei ihren Posts rund um die Demo auch Fotos, die nicht von dieser sind, so wie in diesem Facebook-Beitrag.

dpa-factchecking.com

Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f

Disinformation in Social Media

Das Foto ist von der Loveparade vor 21 Jahren

31.08.2020, 16:09 (CEST)

Wie nach den Demonstrationen gegen die Corona-Maßnahmen der Bundesregierung Anfang August in Berlin werden auch nach den Kundgebungen am 29. August sehr hohe und von den offiziellen Polizei-Angaben abweichende Teilnehmerzahlen verbreitet. Manche User verwenden bei ihren Posts rund um die Demo auch Fotos, die nicht von dieser sind, so wie in diesem Facebook-Beitrag.

dpa-factchecking.com

Kameraperspektive

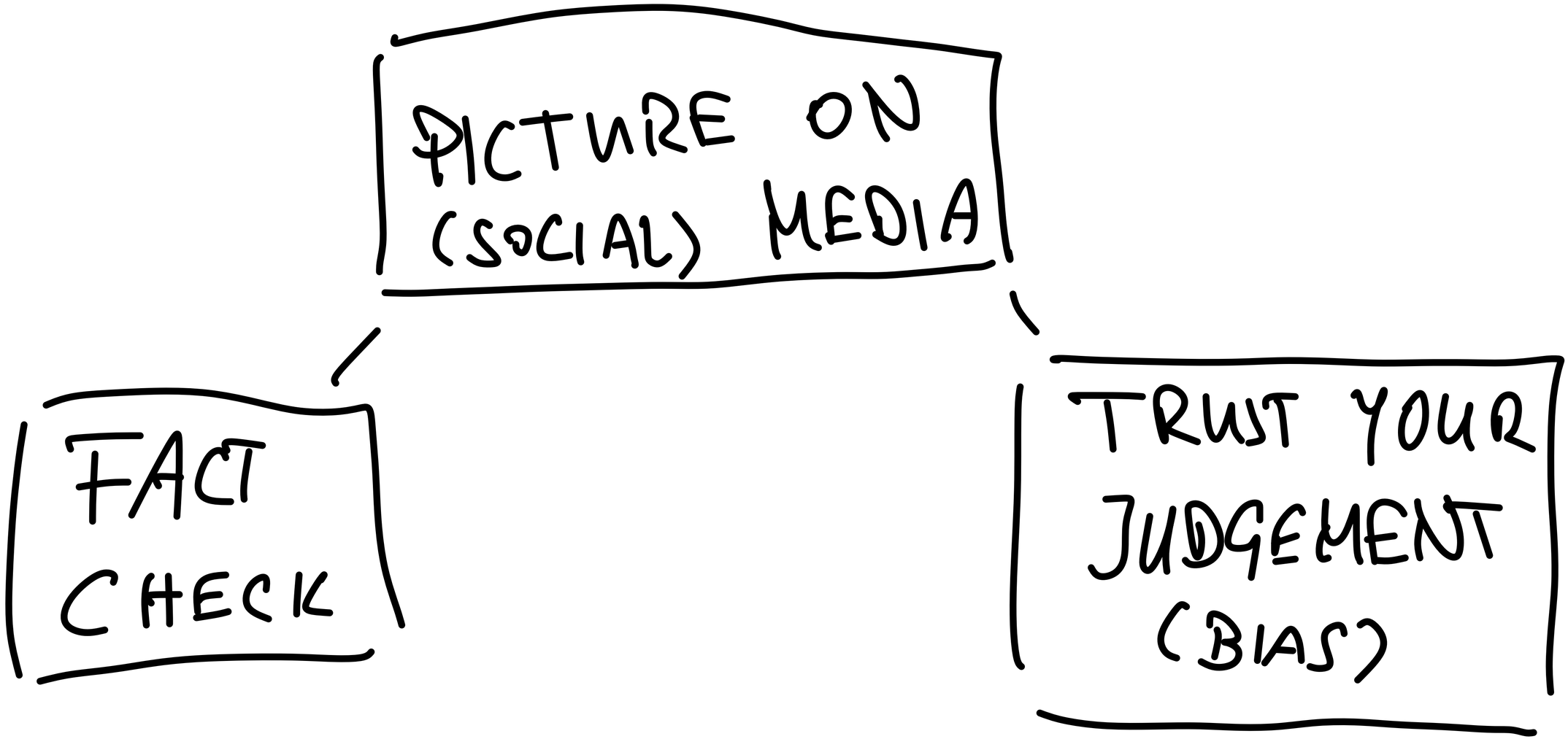
Keine Manipulation: Kleine Alster durch Demo-Teilnehmer verdeckt

22.01.2024, 15:33 (CET), letztes Update: 22.01.2024, 16:09 (CET)

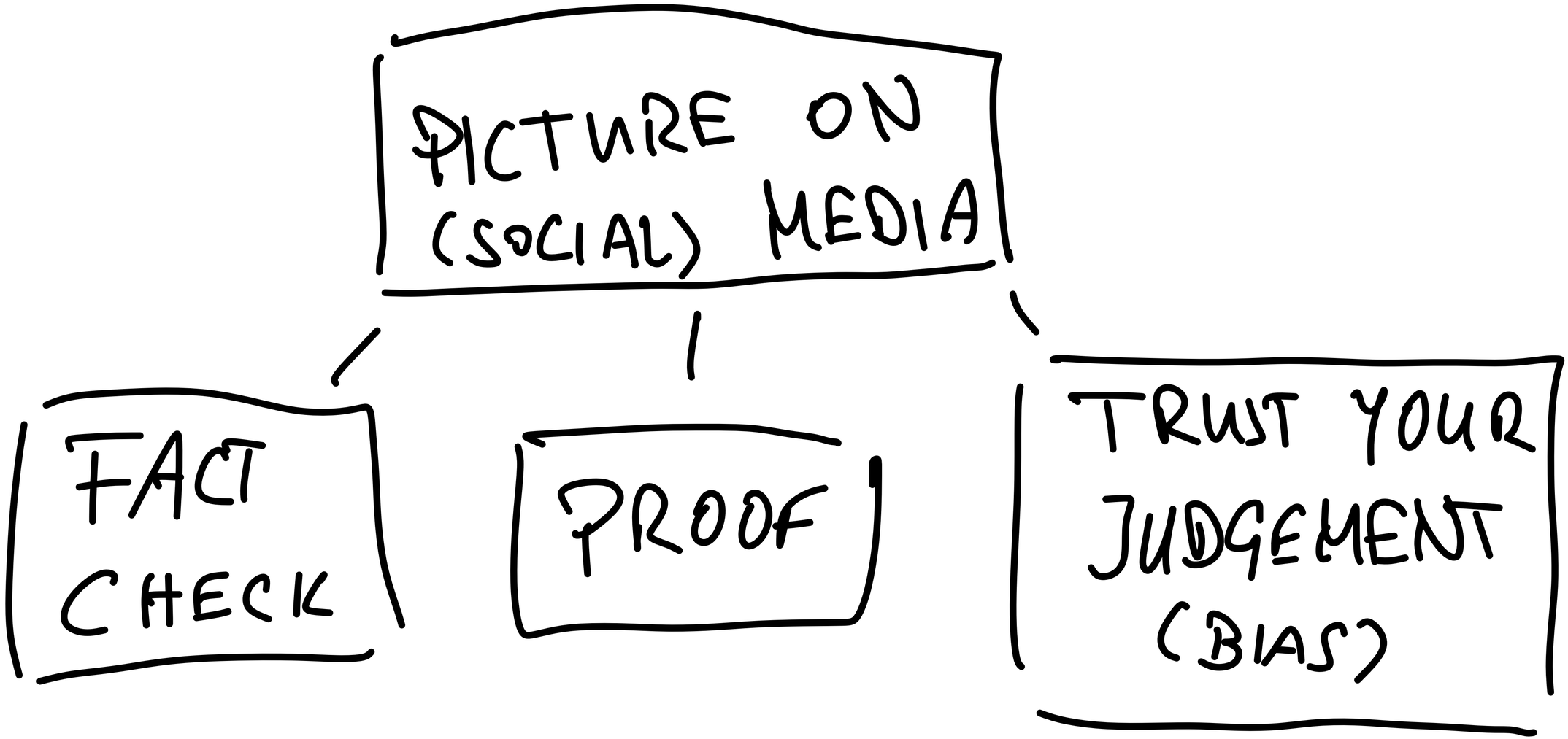
Keine KI, keine Fälschung: Dieses Bild von einer Demonstration in Hamburg gegen rechtsextreme Umtriebe ist echt.
Foto: Jonas Walzberg/dpa

AfD-Politiker Björn Höcke unterstellt dem ZDF die Manipulation eines Bildes. Doch nur weil bestimmte Details auf einem Foto nicht zu erkennen sind, heißt das nicht, dass diese nicht existieren.

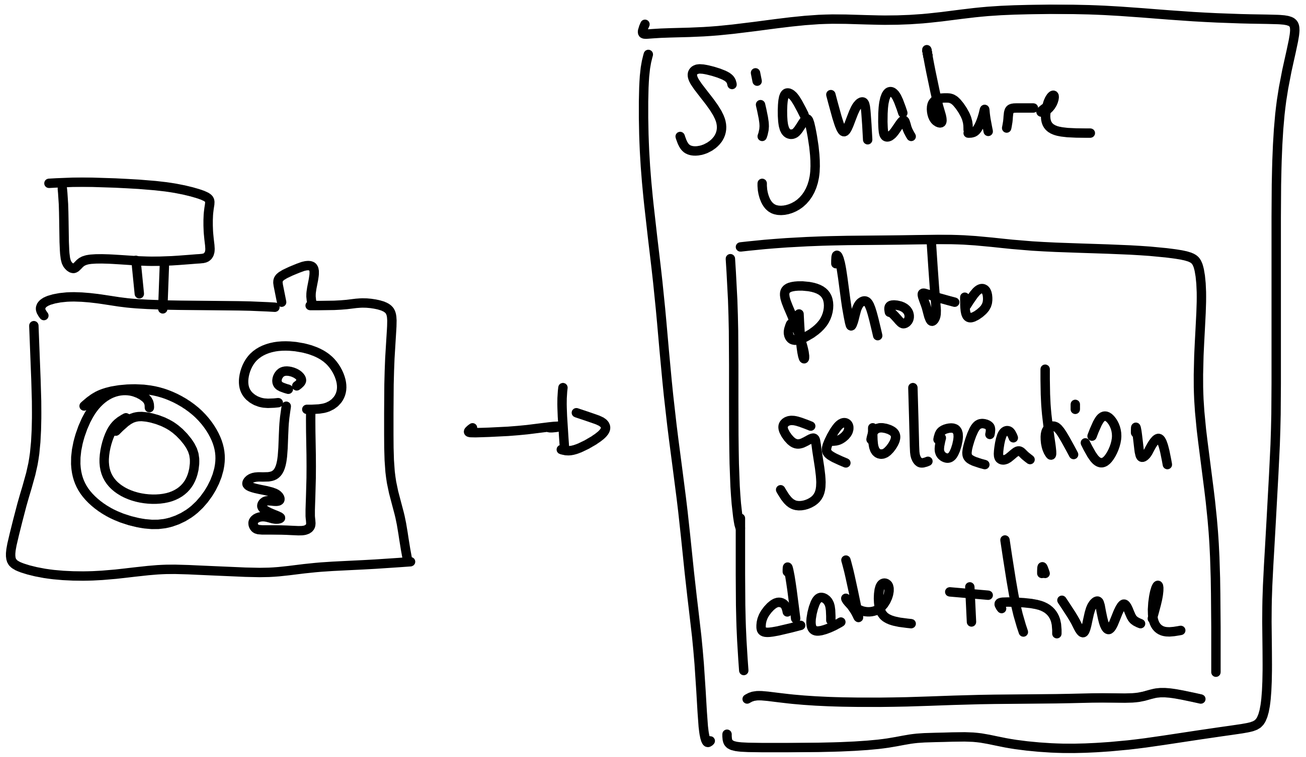
Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f



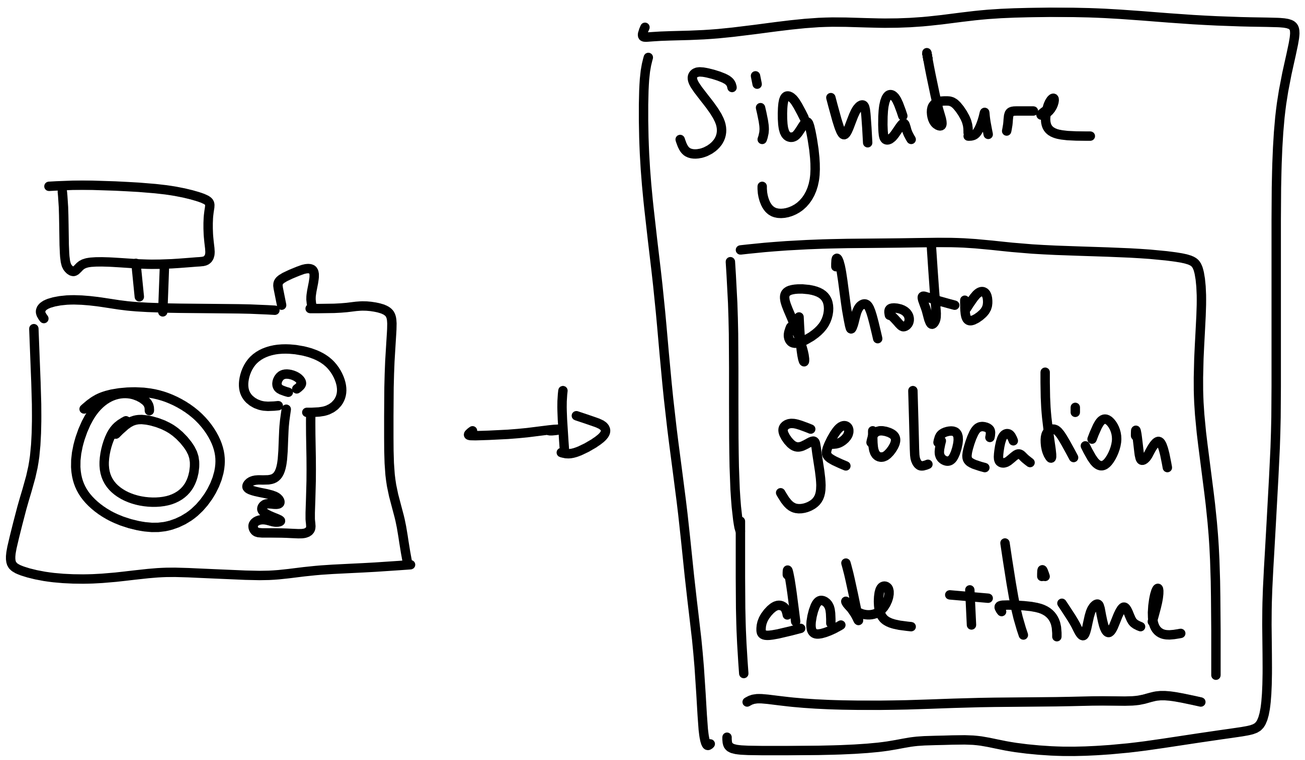
Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f



Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f



Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f



Problem:
publish processed picture
-> signature check fails

Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f

Publisher

proof:

∃ original image & signature

∃ series of permissible edits that result in published picture

consumer (browser)

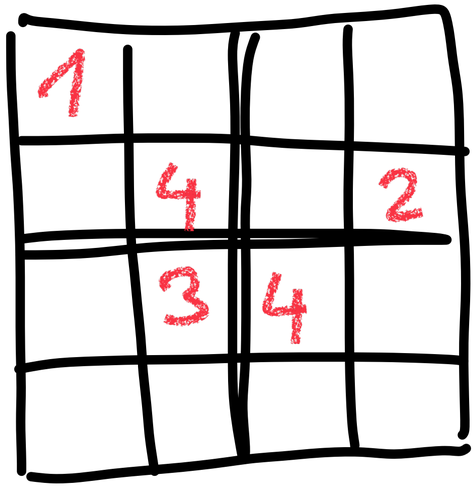
verify proof

show picture

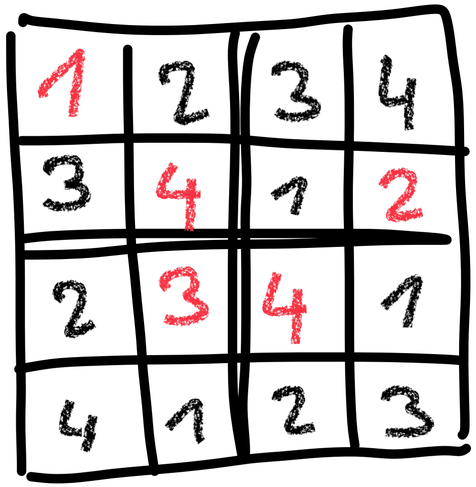
Trisha Datta, Dan Boneh: Using ZK Proofs to Fight Disinformation
https://medium.com/@boneh/using-zk-proofs-to-fight-disinformation-17e7d57fe52f

So, How Does It Work?

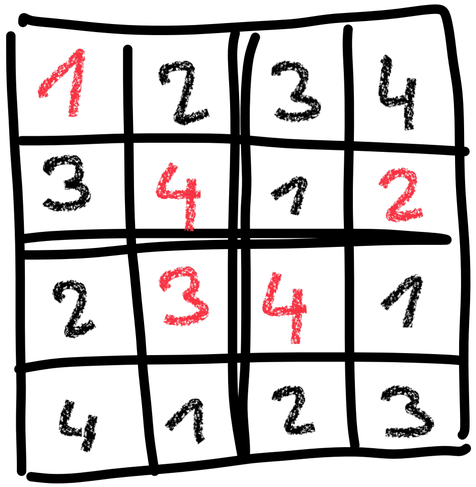
Interactive Example: Sudoku

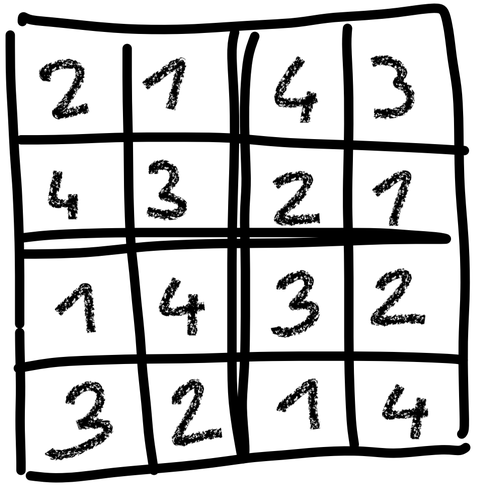


Interactive Example: Sudoku



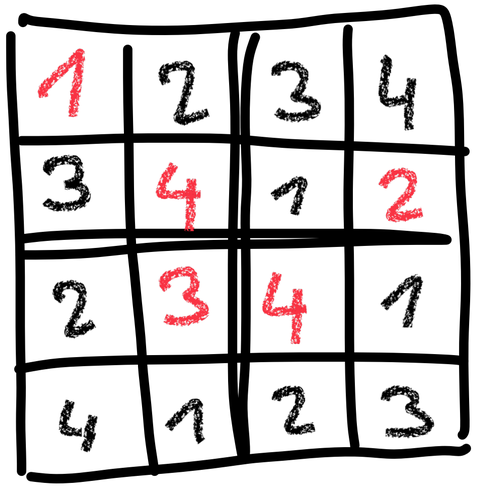
Interactive Example: Sudoku

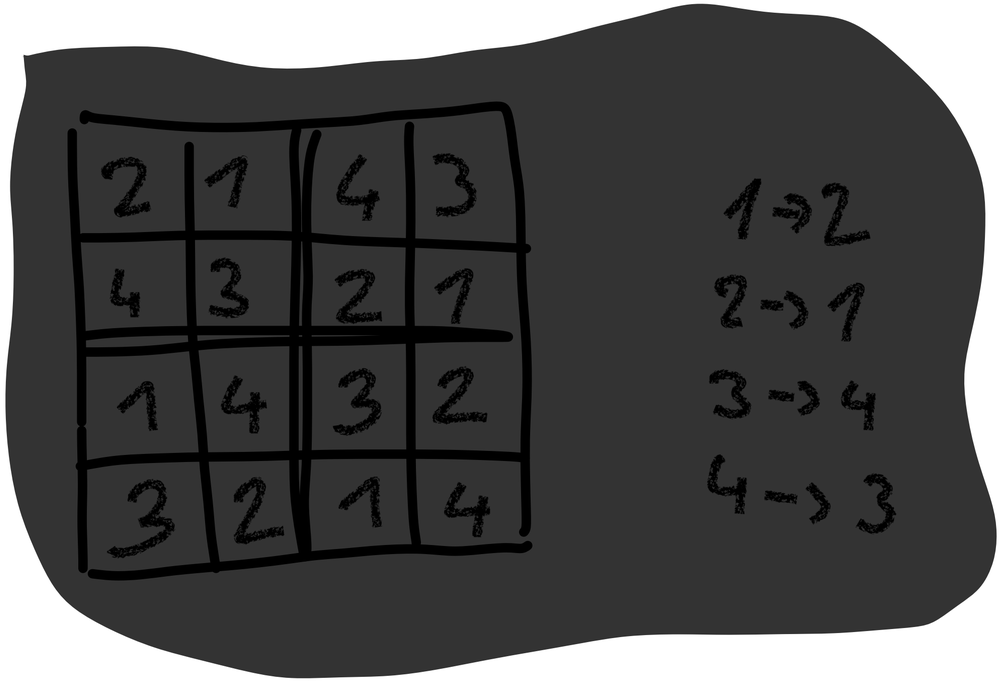




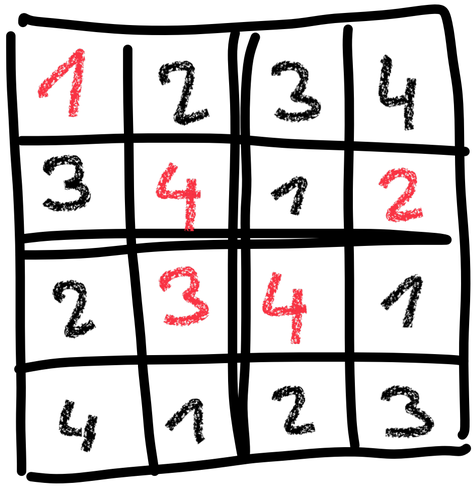
1 → 2
2 → 1
3 → 4
4 → 3

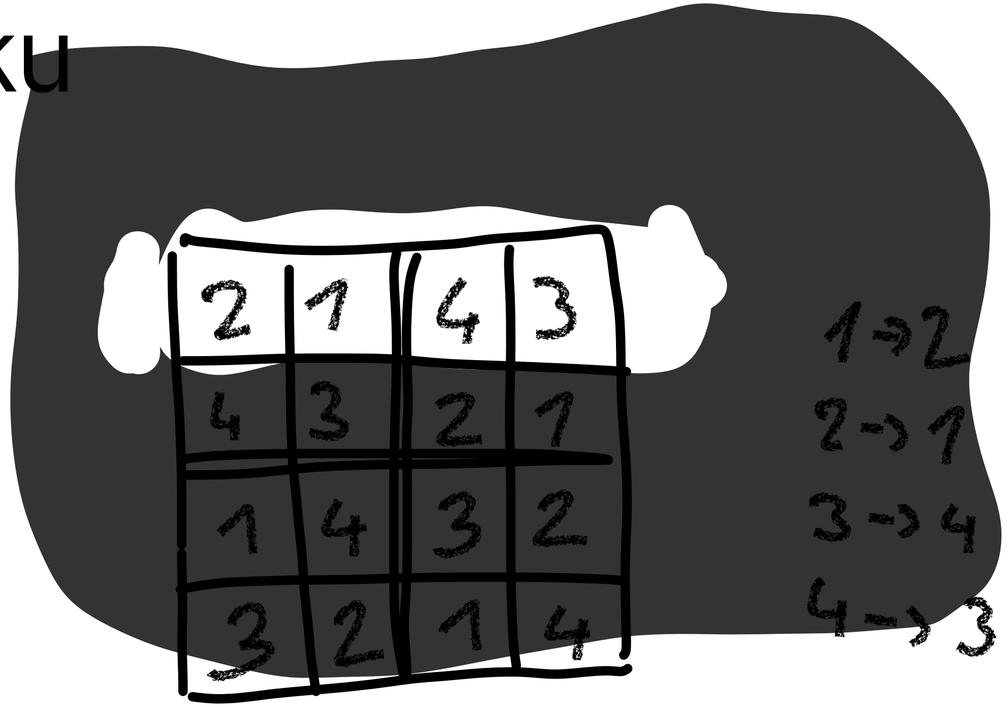
Interactive Example: Sudoku



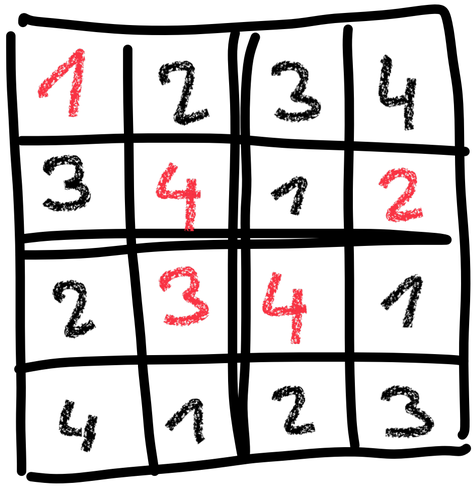


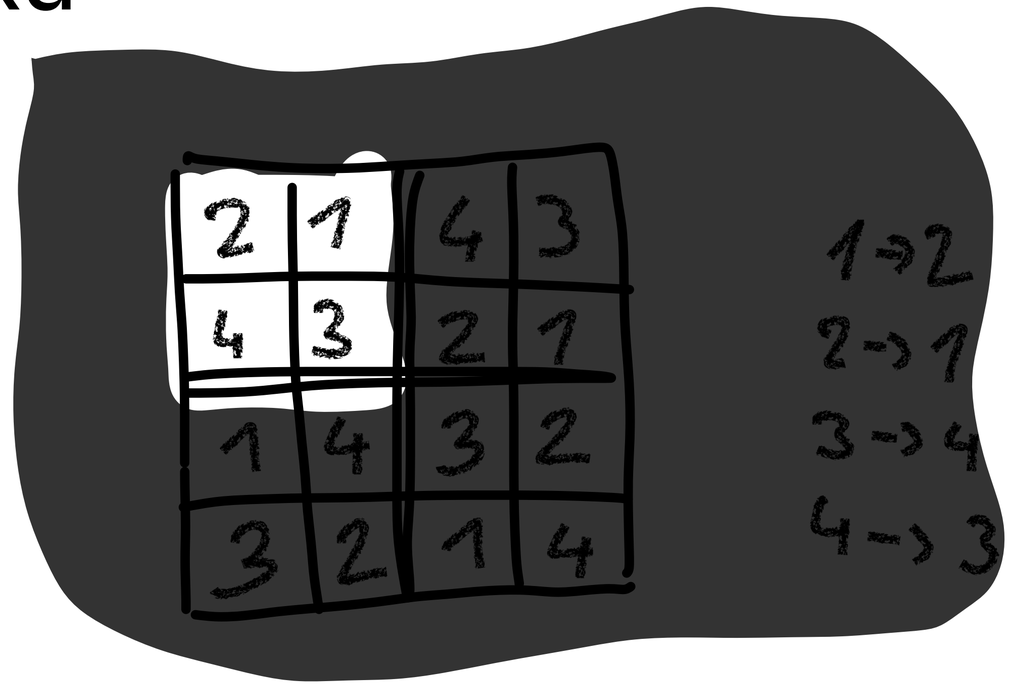
Interactive Example: Sudoku



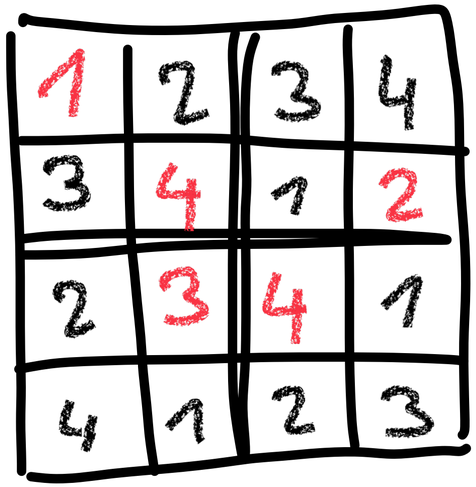


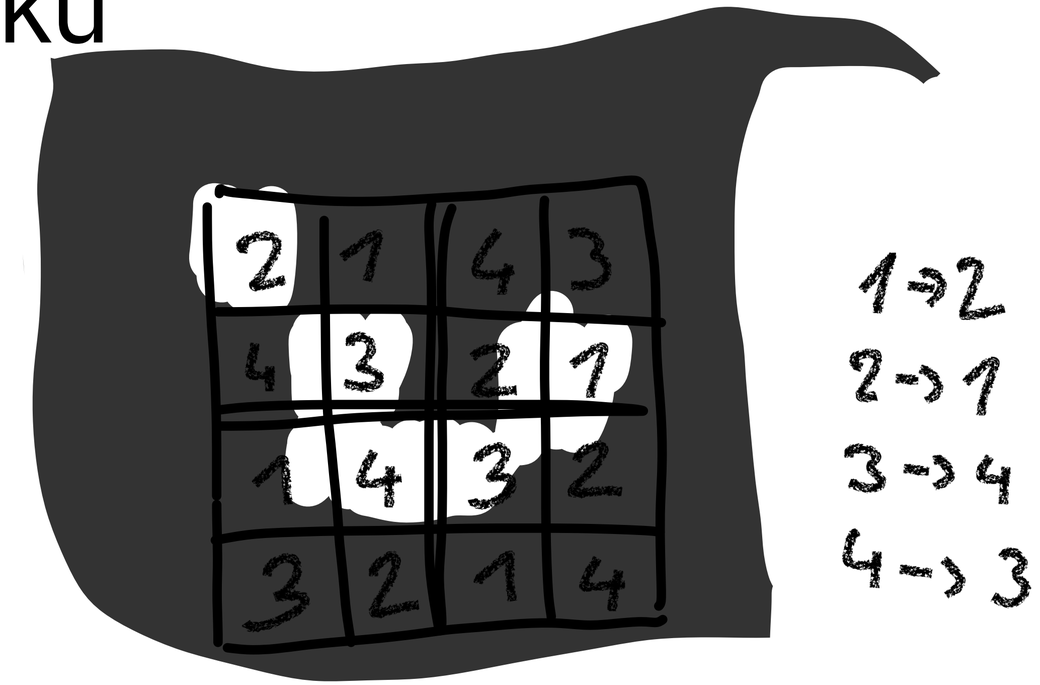
Interactive Example: Sudoku



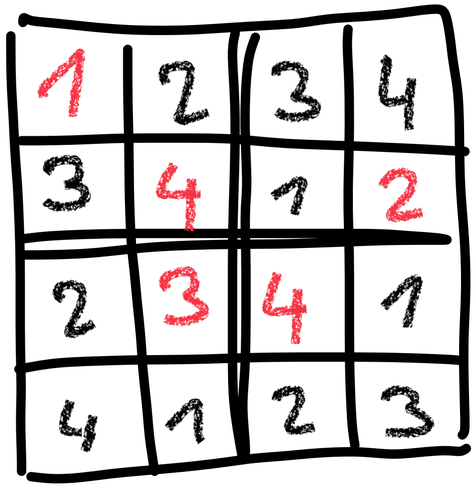


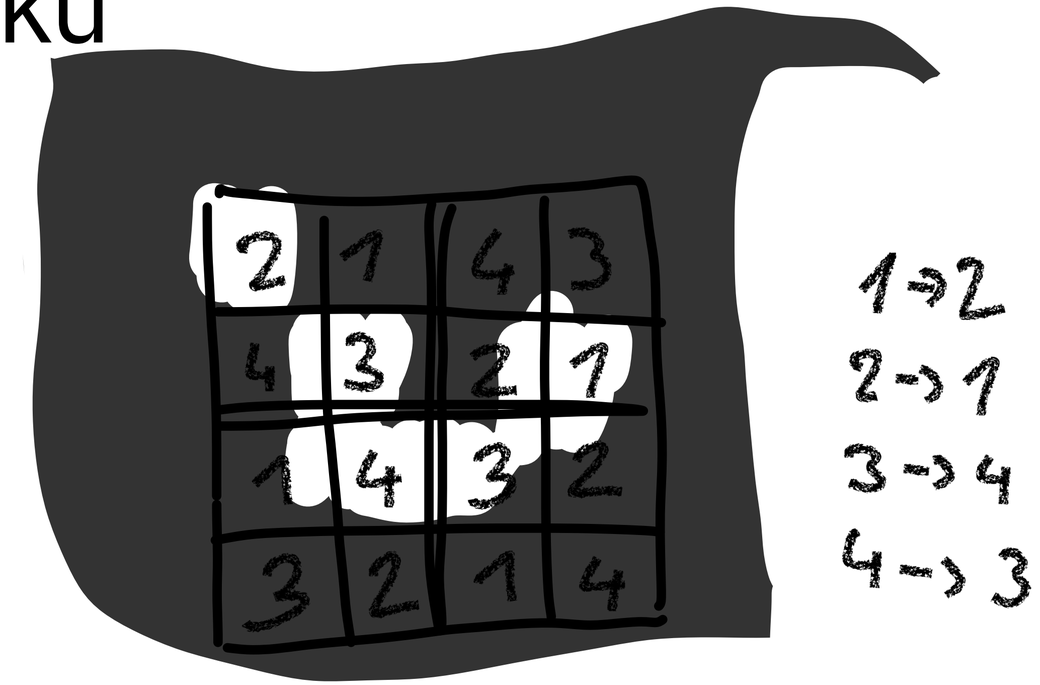
Interactive Example: Sudoku





Interactive Example: Sudoku





iterate with new permutation

zk

SNARK

ZK ZERO KNOWLEDGE

S SUCCINCT

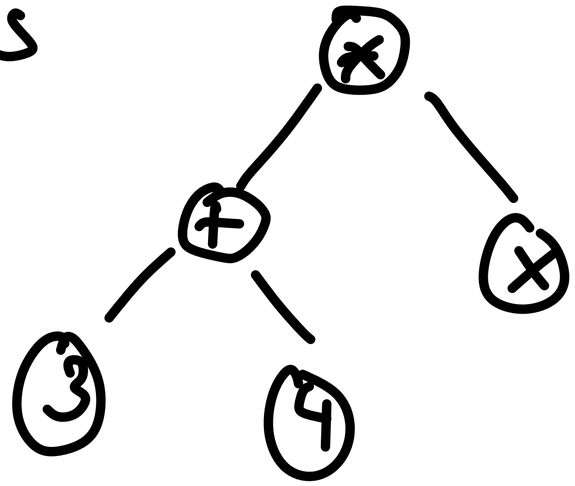
N NON-INTERACTIVE

AR ARGUMENT of

K KNOWLEDGE

SNARK Ingredients

- polynomials over finite fields
- elliptic curve cryptography
- commitment schemes
- arithmetic circuits



o1js: TypeScript DSL for Provable Programs

- write program in TypeScript, using o1js library
- mostly abstracts away circuits, polynomials, etc.
  → some constraints (static circuit size, control flow, ...)
- useful analogy: hardware design

```
class Sudoku extends Struct({
  value: Provable.Array(Provable.Array(Field, 9), 9),
}) {
  static from(value: number[][]) {
    return new Sudoku({ value: value.map((row) => row.map(Field)) });
  }

  hash() {
    return Poseidon.hash(this.value.flat());
  }
}
```

```
class SudokuZkApp extends SmartContract {
  @state(Field) sudokuHash = State<Field>();
  @state(Bool) isSolved = State<Bool>();

  @method submitSolution(sudokuInstance: Sudoku, solutionInstance: Sudoku) {
    let sudoku = sudokuInstance.value;
    let solution = solutionInstance.value;

    // first, we check that the passed solution is a valid sudoku
    // ...

    // next, we check that the solution extends the initial sudoku
    for (let i = 0; i < 9; i++) {
      for (let j = 0; j < 9; j++) {
        let cell = sudoku[i][j];
        let solutionCell = solution[i][j];
        // either the sudoku has nothing in it (indicated by a cell value of 0),
        // or it is equal to the solution
        Bool.or(cell.equals(0), cell.equals(solutionCell)).assertTrue(
          `solution cell (${i + 1},${j + 1}) matches the original sudoku`
        );
      }
    }

    // finally, we check that the sudoku is the one that was originally deployed
    let sudokuHash = this.sudokuHash.getAndAssertEquals();

    sudokuInstance
      .hash()
      .assertEquals(sudokuHash, 'sudoku matches the one committed on-chain');

    // all checks passed => the sudoku is solved!
    this.isSolved.set(Bool(true));
  }
```

Thank You!

More Information: https://docs.minaprotocol.com/