Lecture 01:
A Primer on Blockchain

Nick Zoghb

BLOCKCHAIN
AT BERKELEY

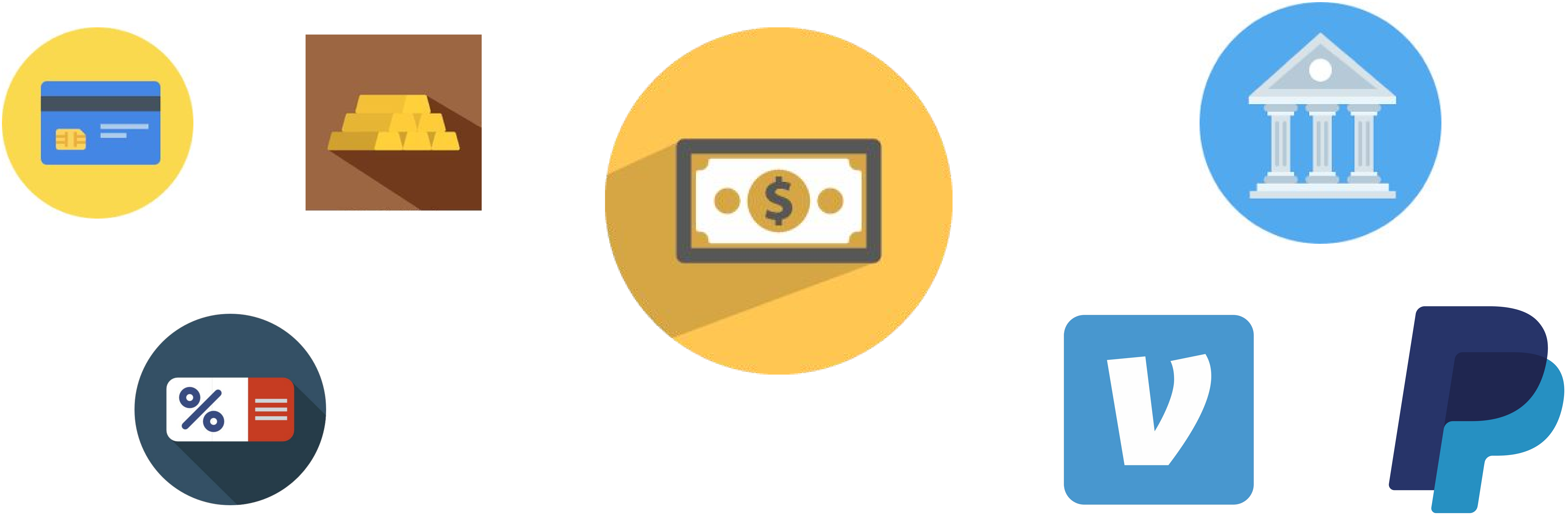
LECTURE OUTLINE

1. ECONOMIC CONTEXT
2. INTRO TO BLOCKCHAIN

1 ECONOMIC CONTEXT

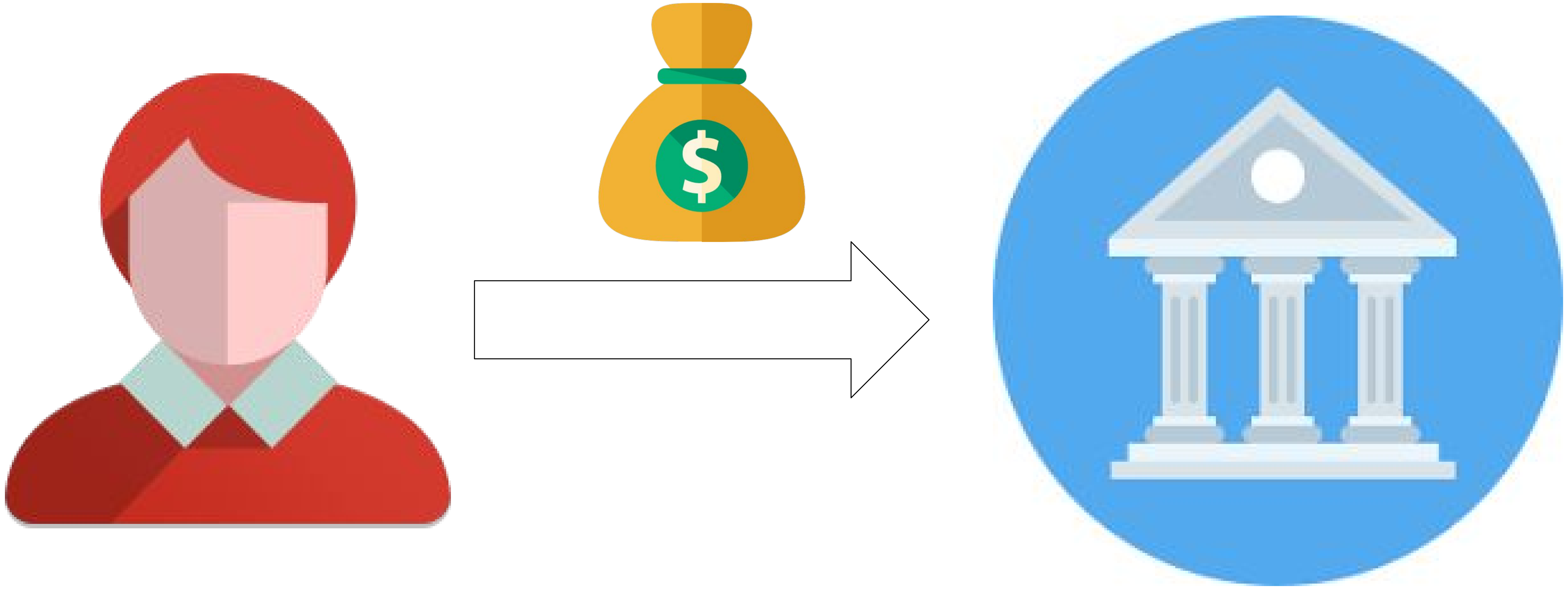
THE NATURE OF MONEY

WHAT IS IT ACTUALLY THOUGH



A CENTRAL POINT OF FAILURE

WHAT COULD POSSIBLY GO WRONG?



A CENTRAL POINT OF FAILURE

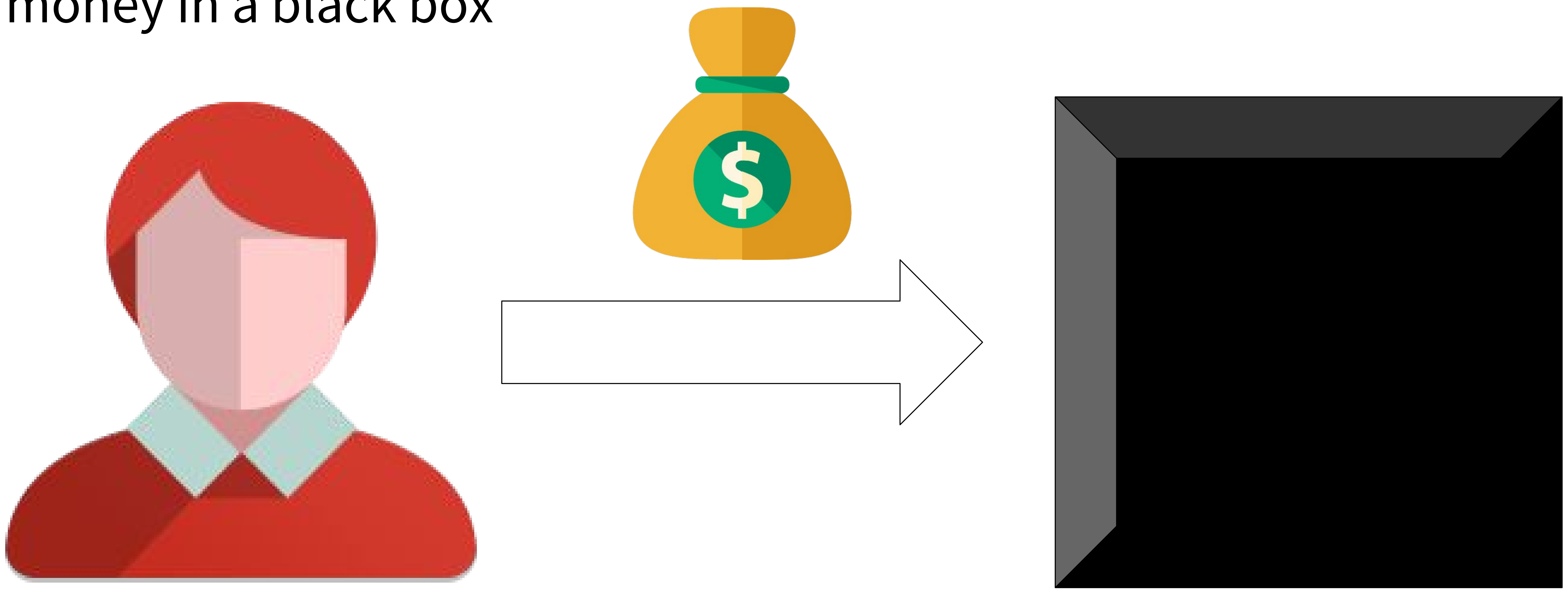
WHAT COULD POSSIBLY GO WRONG?

1. Frozen assets
2. Seized assets
3. Transaction fees
4. Hidden fees
5. Robbery
6. Hacking
7. Financial loss
8. Fraud
9. Identity theft
10. Breach of privacy

NO OPEN ENVIRONMENT

TOSSING MONEY INTO THE VOID

- Flow is abstracted away as it settles in the background from the giant entanglement of banks and payment networks - tons of middlemen
- Akin to putting money in a black box



MONEY DEFINITIONS

WHAT MAKES MONEY?

Properties of Money

- Storage of value
- A unit of account - you use it to compare the value of other items
- A medium of exchange - it's relatively easy to move around

Characteristics of Money

- Durability, portability, divisibility, uniformity, limited supply, and acceptability

MONEY DEFINITIONS

WHAT MAKES MONEY?

Types of Money

- Representative - a certificate or token that can be exchanged for the underlying commodity
- Fiat - money that does not have intrinsic value and does not represent an asset in a vault somewhere

See this post for a more in-depth explanation.

FIAT CURRENCIES

BACKED BY NOTHING

- Fiat; currencies which is valued based on faith (usually in the economy relative to other currencies) and has no commodity backing
  - Bitcoin functioning more like digital cash, not credit
  - Bitcoin is very much like fiat currency, but instead of using banks as intermediaries, you use a network of computers

- What else is fiat?

FIAT CURRENCIES

BACKED BY NOTHING

So...  = 

??????

- Not quite!
  - Fiat currency is also legal tender; backed by central government, credit, bonds, pay taxes with it
  - Cryptocurrencies no backed by government; no single entity can enact monetary policy

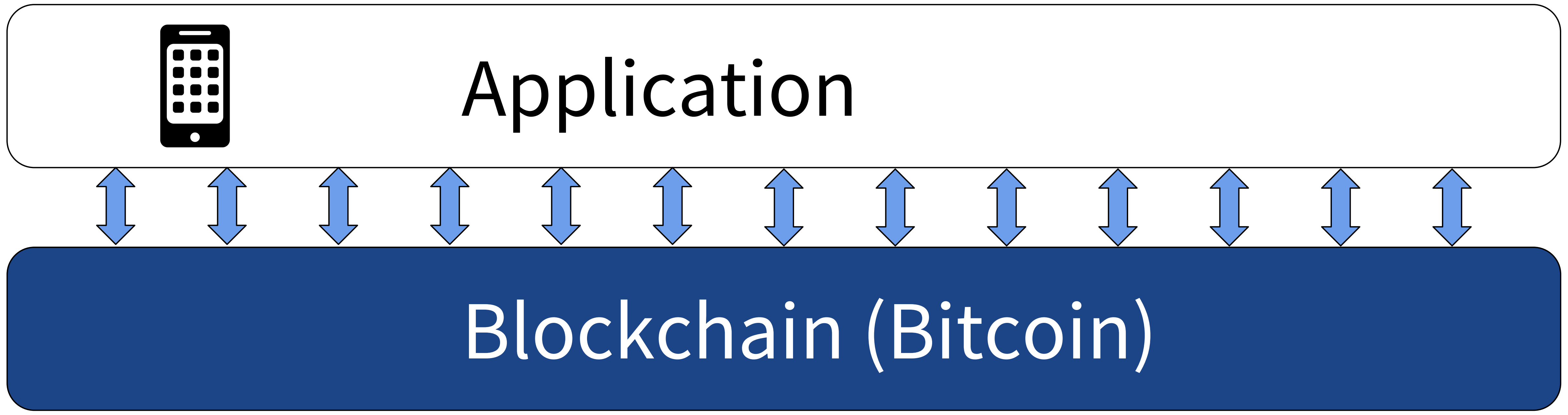
2 THE BLOCKCHAIN PROPOSAL

CRYPTOCURRENCIES AS A USE CASE

BLOCKCHAIN IS SO MUCH MORE

What does Bitcoin provide?

- Decentralized, open source, p2p cash protocol
- Solves the *Byzantine Generals' Problem*; a problem in distributed computing where several actors must work to agree on the state of a system
  - Over an unreliable network
  - Without a central trusted authority

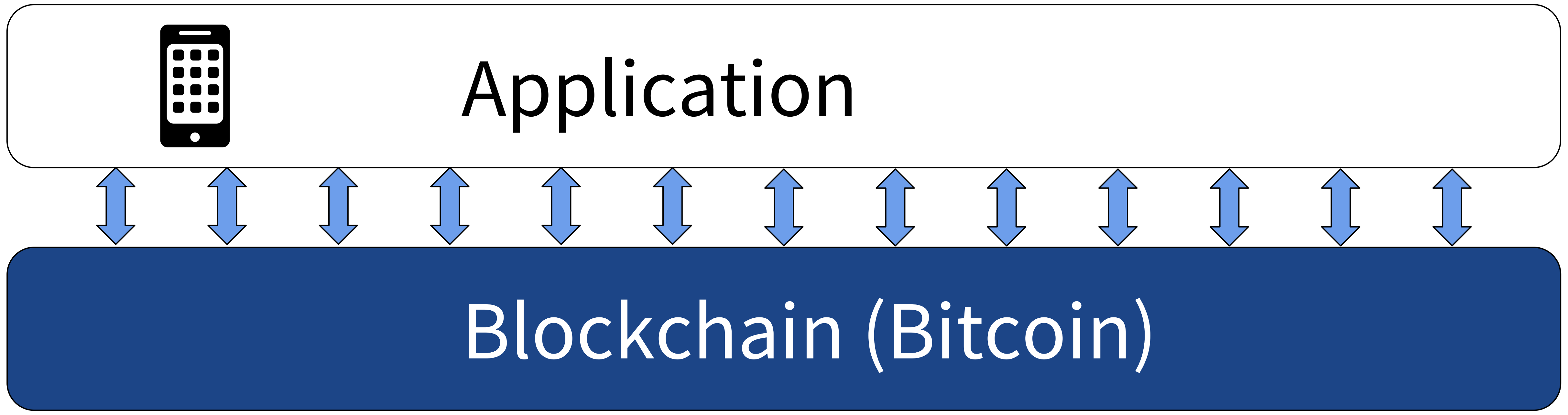


CRYPTOCURRENCIES AS A USE CASE

BLOCKCHAIN IS SO MUCH MORE

- It also avoids a lot of the pitfalls associated with banks:
  - Hacking; hash power requirement thwarts adversarial attempts at subverting the network
  - Fraud; privacy of network along with public addresses maintains accountability
  - Frozen assets; only those in control of private keys are able to transact with their own “assets”

See this article for an in-depth explanation.



BITCOIN DECOMPOSITION

A MENTAL MODEL

- Transactions
- Identity

- Application
- Semantic
- Propagation
- Mining
- Consensus

BITCOIN: IDENTITY

A MENTAL MODEL

Transactions

Identity

Let's start here

Application

Semantic

Propagation

Mining

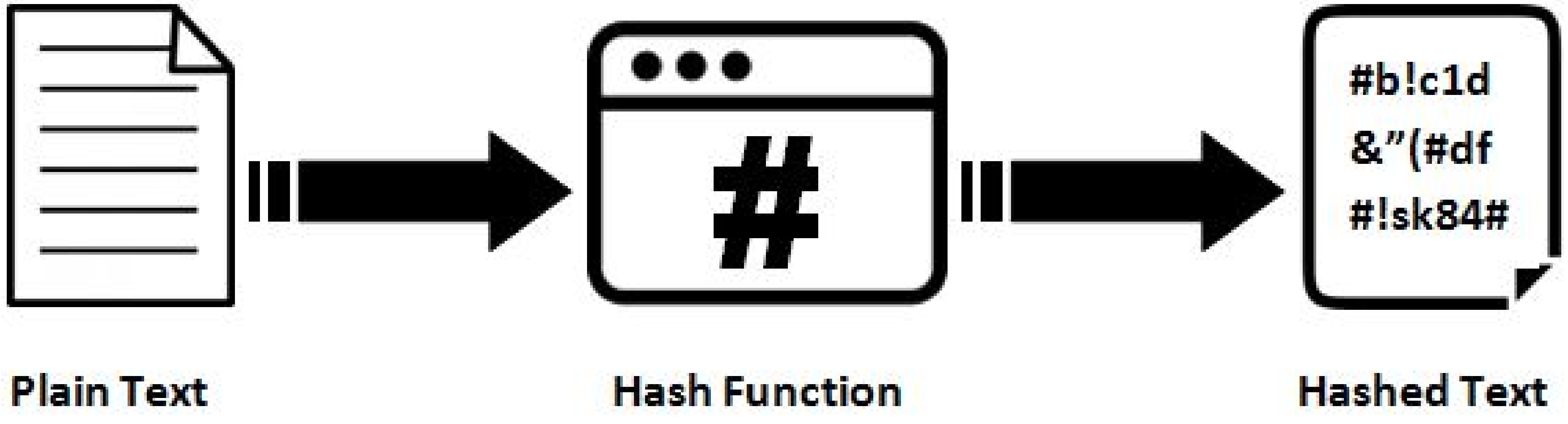
Consensus

2.1 IDENTITY

CRYPTOGRAPHIC PRIMITIVES

HASH FUNCTIONS

- A hash function produces a deterministic fixed-size, random looking output which is called a hash
  - Used in encryption
  - Used to detect tampering of data



Properties of hash functions

- Collision Resistance; a hash function is said to be collision resistant if it is infeasible to find two different input values where the output of those values is the same
- Preimage Resistance; given the output, can't find input of the hash function
  - Second pre-image resistance?

IDENTITY

PUBLIC AND PRIVATE KEYS

```
18E14A7B6A307F426A94F8
114701E7C8E774E7F9A47E
2C2035DB29A206321725
```

```
0450863AD64A87AE8A2FE83C1AF1A8403C
B53F53E486D8511DAD8A04887E5B23522C
D470243453A299FA9E77237716103ABC11
A1DF38855ED6F2EE187E9C582BA6
```

```
16UwLL9Risc3QfPqBUvKof
HmBQ7wMtjvM
```

256 bits

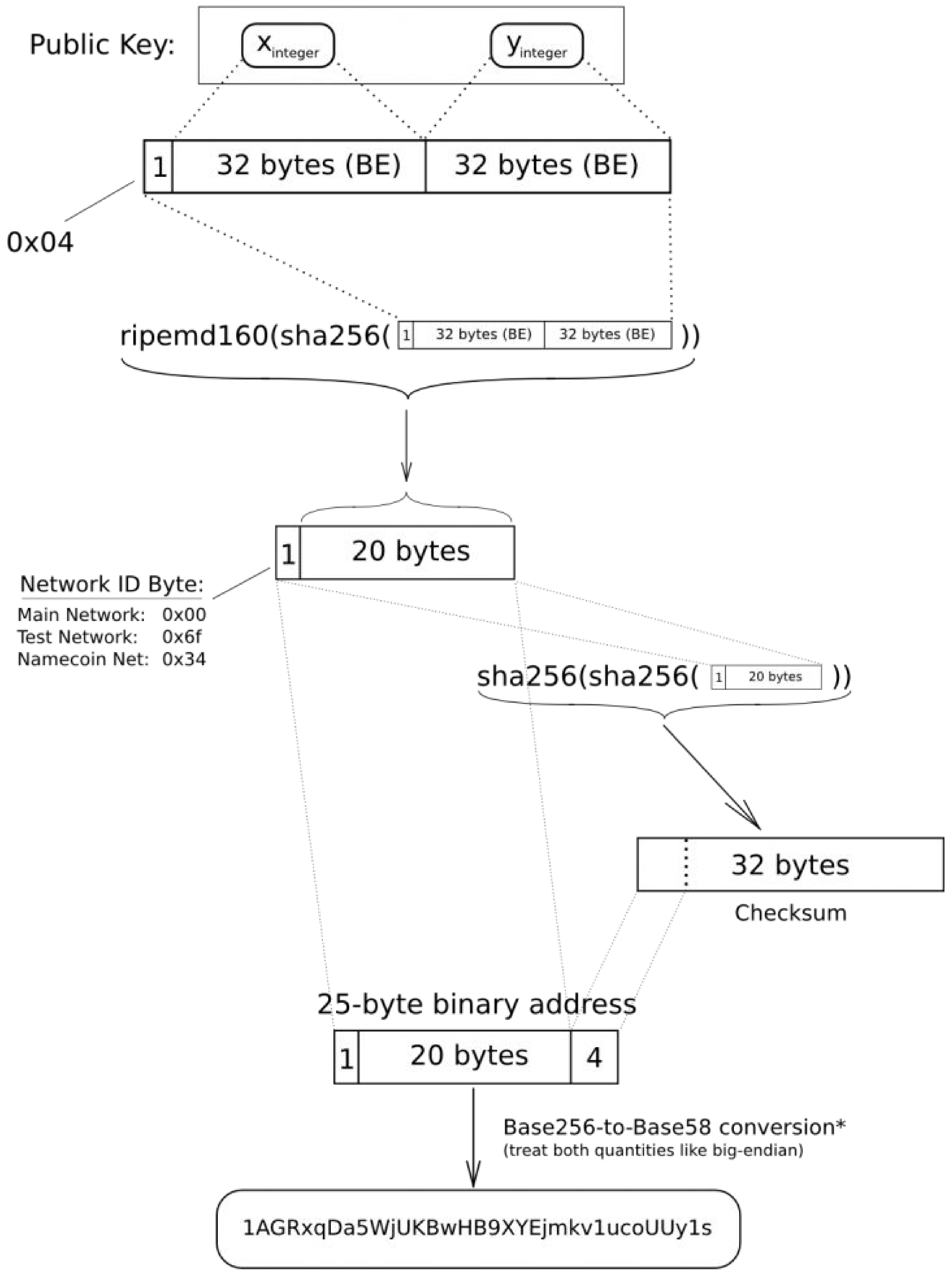
ECDSA

520 bits

Many more steps

160 bits

Elliptic-Curve Public Key to BTC Address conversion

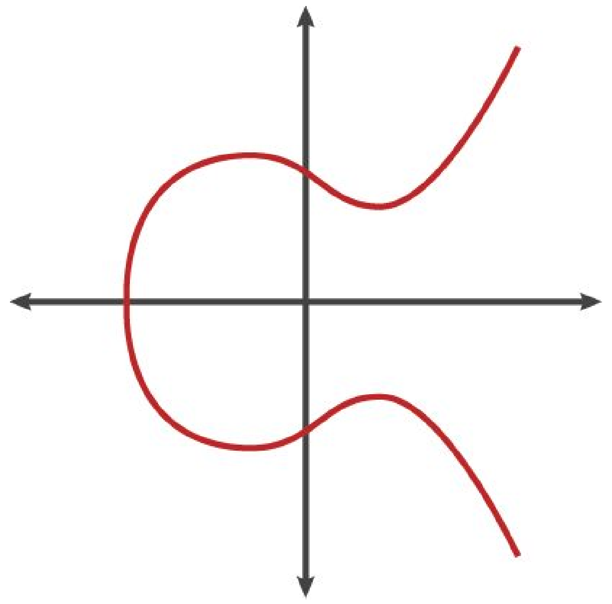


Many more steps

AUTHOR: NICK ZOGHB

CRYPTOGRAPHIC PRIMITIVES

DIGITAL SIGNATURES



- Digital analog to a handwritten signature on paper
- Bitcoin allows you to create as many addresses as you want, and use a new one for every transaction

Properties of Digital Signatures

- Only you can make your signature, and others can only verify that it's valid
  - If I sign a message with my secret key and someone tries to validate that with my public key, the signature must validate correctly
- Unforgeability; computationally infeasible to try to forge a signature

IDENTITY

SECURITY: HIDDEN IN PLAIN SIGHT

“What if someone guesses my private key?!”

- Bitcoin is hidden in the large amount of public keys
  - 2^{160} (1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,97) possible addresses
- Even in the best-case scenario where every person in the world owns a private key, the chance of guessing correctly is 0.0000000000000000000000000000000000000000000000000000000065634881018717779152936274157283036740481602769715738%

IDENTITY

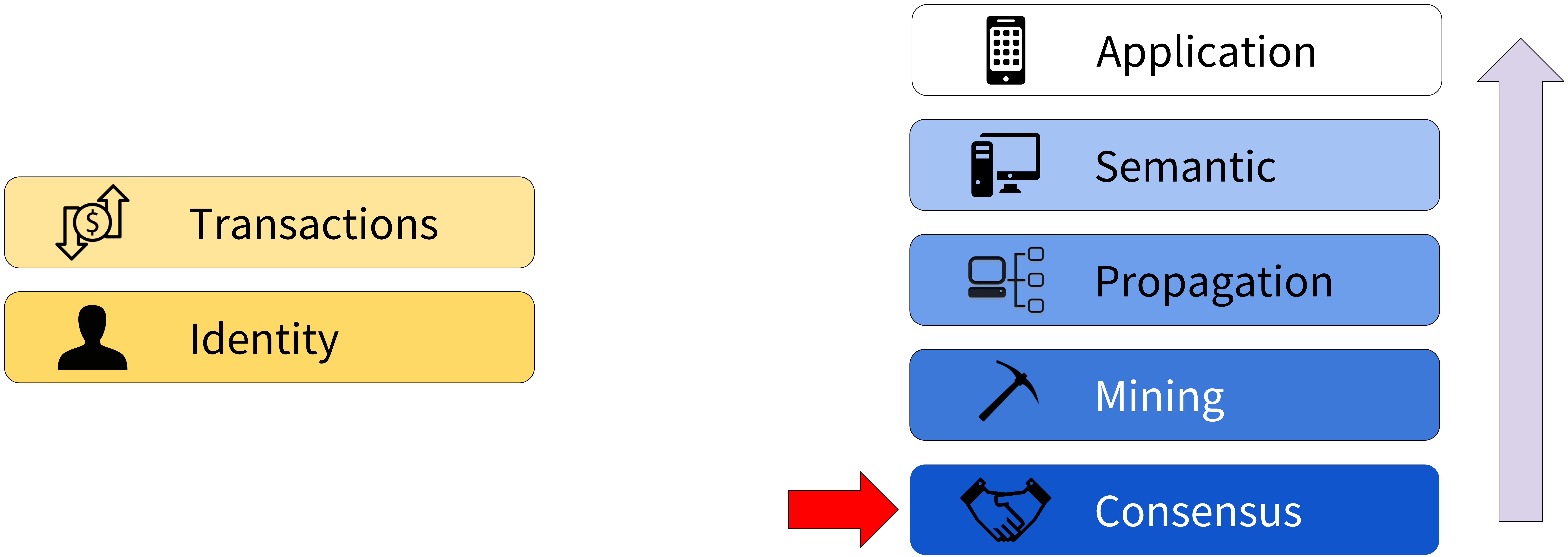
SECURITY: HIDDEN IN PLAIN SIGHT

“What if someone guesses my private key?!”

- Practically impossible for anyone to overlap
  - For reference:
    - Grains of sand on earth: 2^{63}
    - With 2^{63} Earths, each with 2^{63} grains of sand: 2^{126} total grains of sand
    - 2^{126} is only 0.0000000058% of 2^{160}
  - Population of world: 7.5 billion in April 2017
    - Every person could have about 2^{127} addresses all to themselves

BITCOIN: CONSENSUS

A MENTAL MODEL

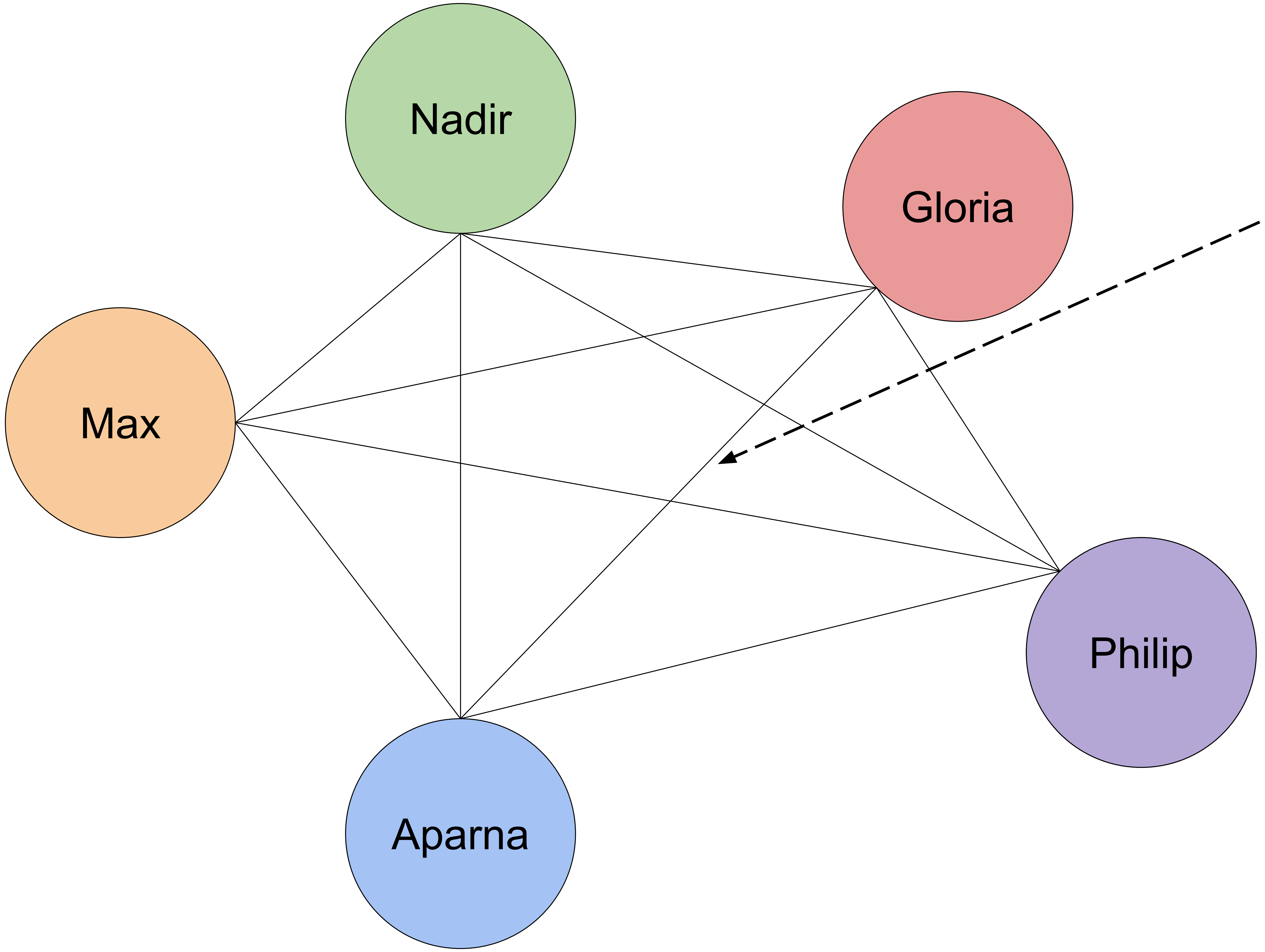


2.2 CONSENSUS

RECORD-KEEPING

DISTRIBUTED DATABASES

Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2



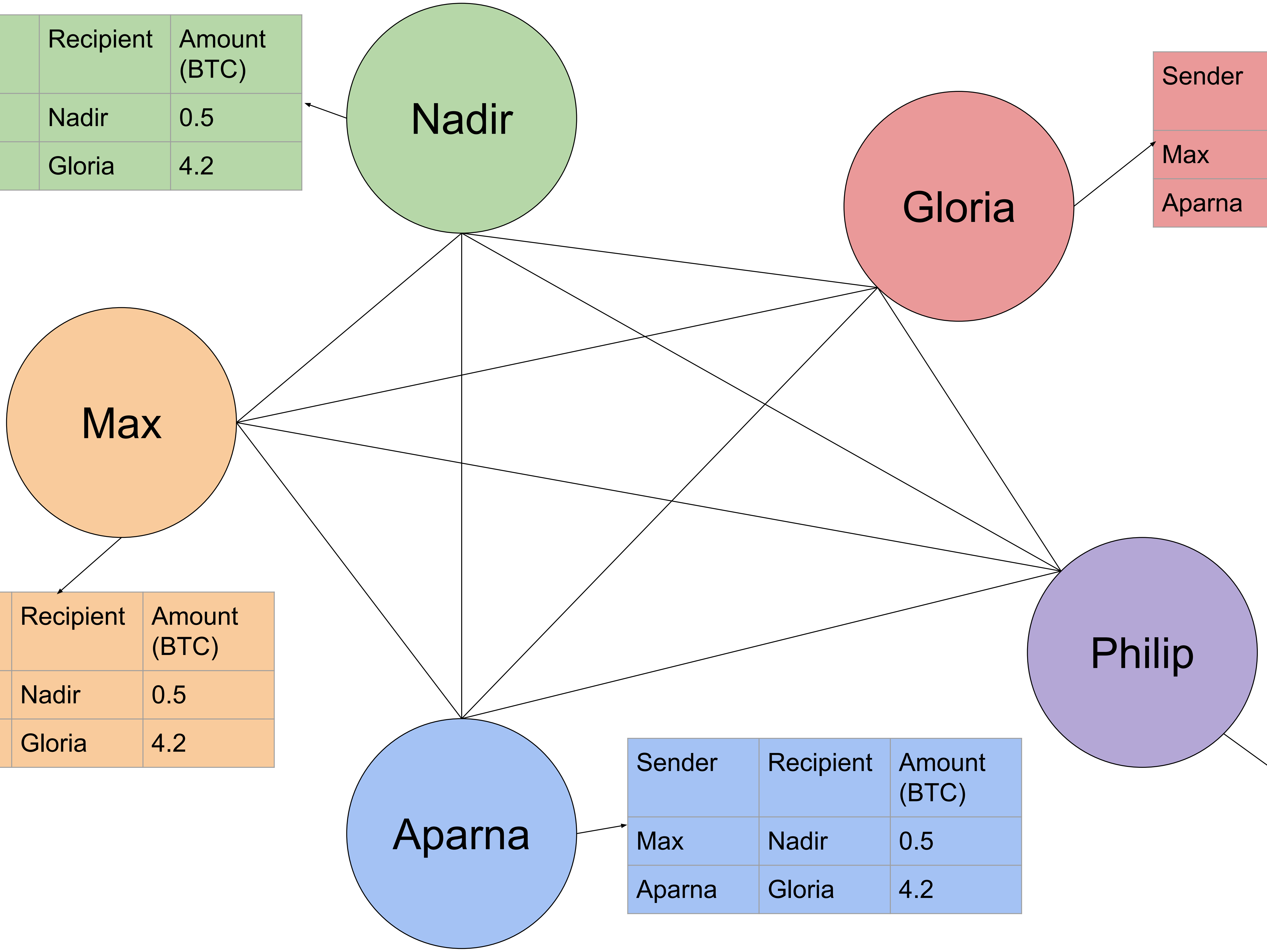
We want to represent identities and transactions, how do we keep track of this ledger?

⇒ With a *distributed database*

RECORD-KEEPING

EVERYONE'S THE BANK

Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2



Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2

Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2

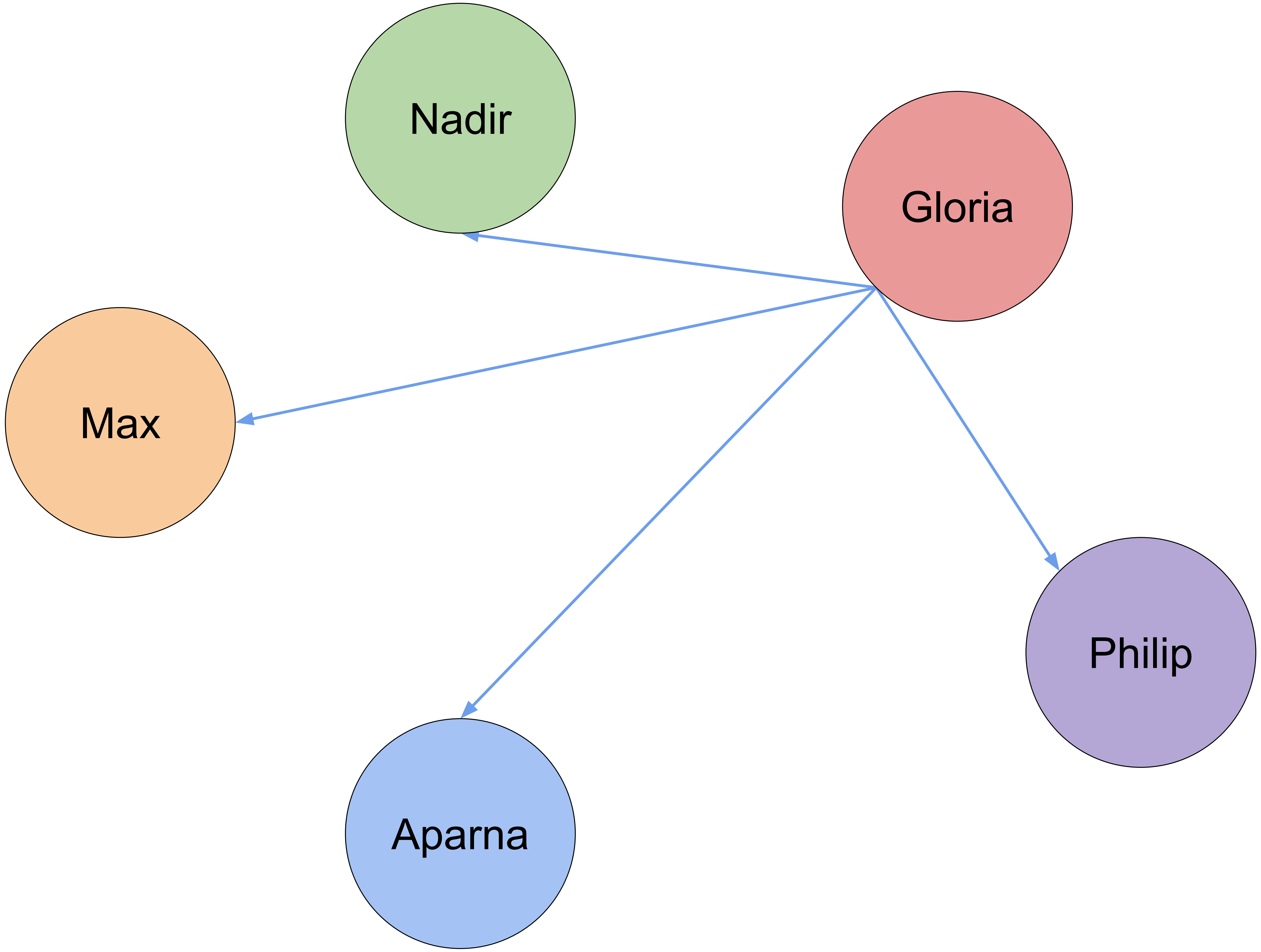
Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2

Sender	Recipient	Amount (BTC)
Max	Nadir	0.5
Aparna	Gloria	4.2

Everyone stores the ledger

CONSENSUS

STAYING ON THE SAME PAGE

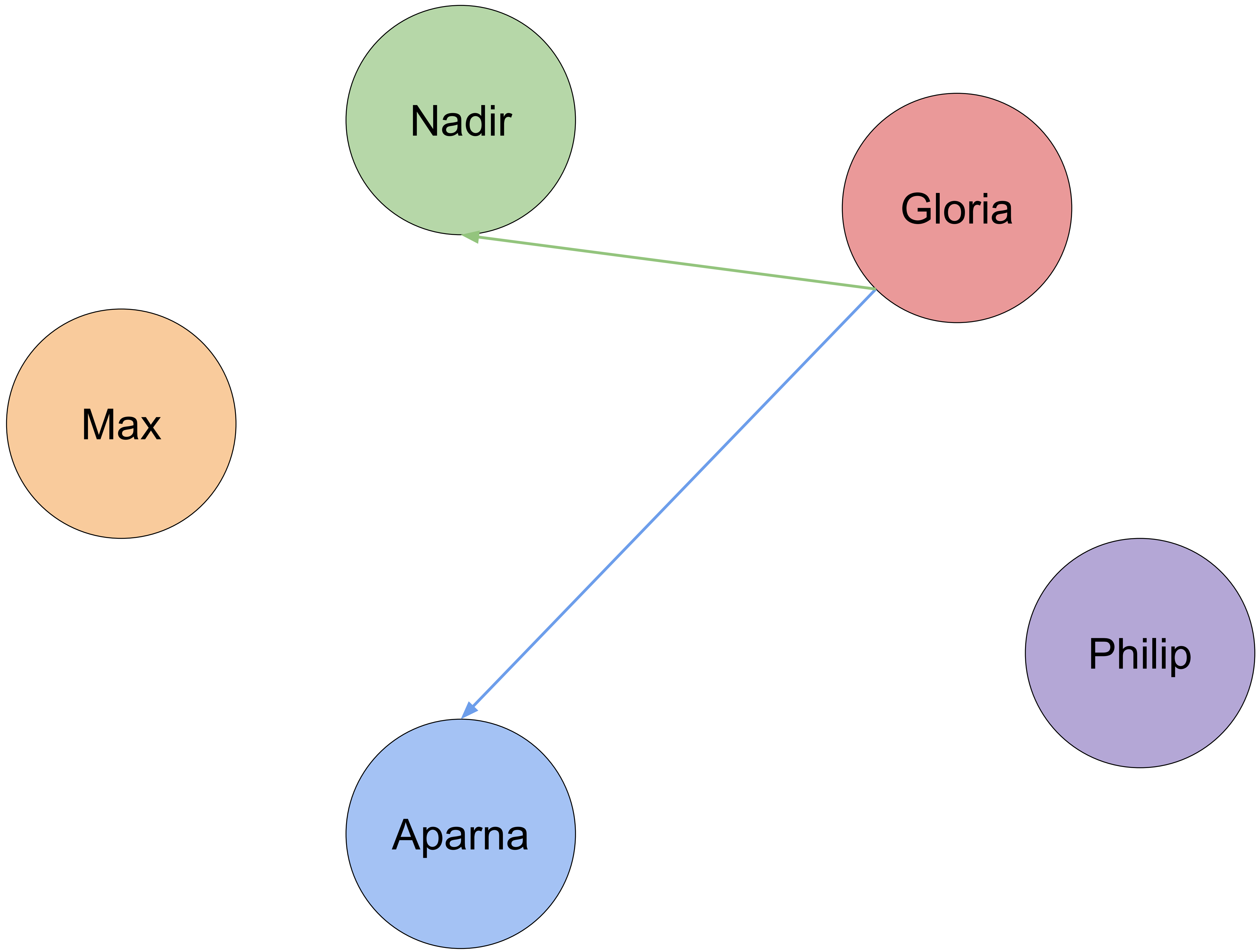


Everyone accepts valid transactions as they come around without "discussion"

How do we ensure no one's cheating if we make decisions alone?

CONSENSUS

DOUBLE SPEND ATTACK

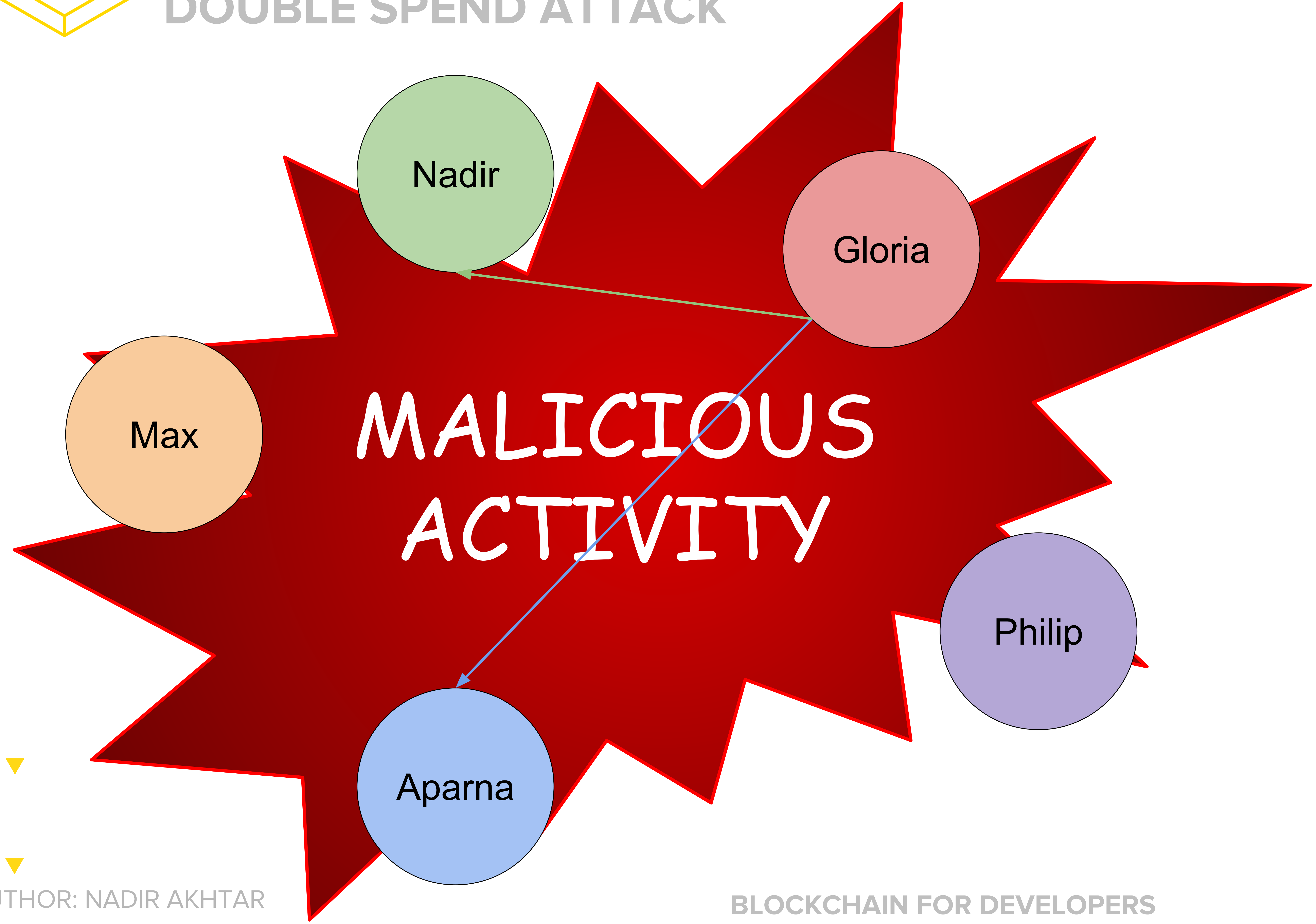


Gloria promises 10 BTC to Aparna in one transaction, and she promises 10 BTC to Nadir in another - but she only has 10 BTC total!

- Gloria is performing a *double spend* attack

CONSENSUS

DOUBLE SPEND ATTACK

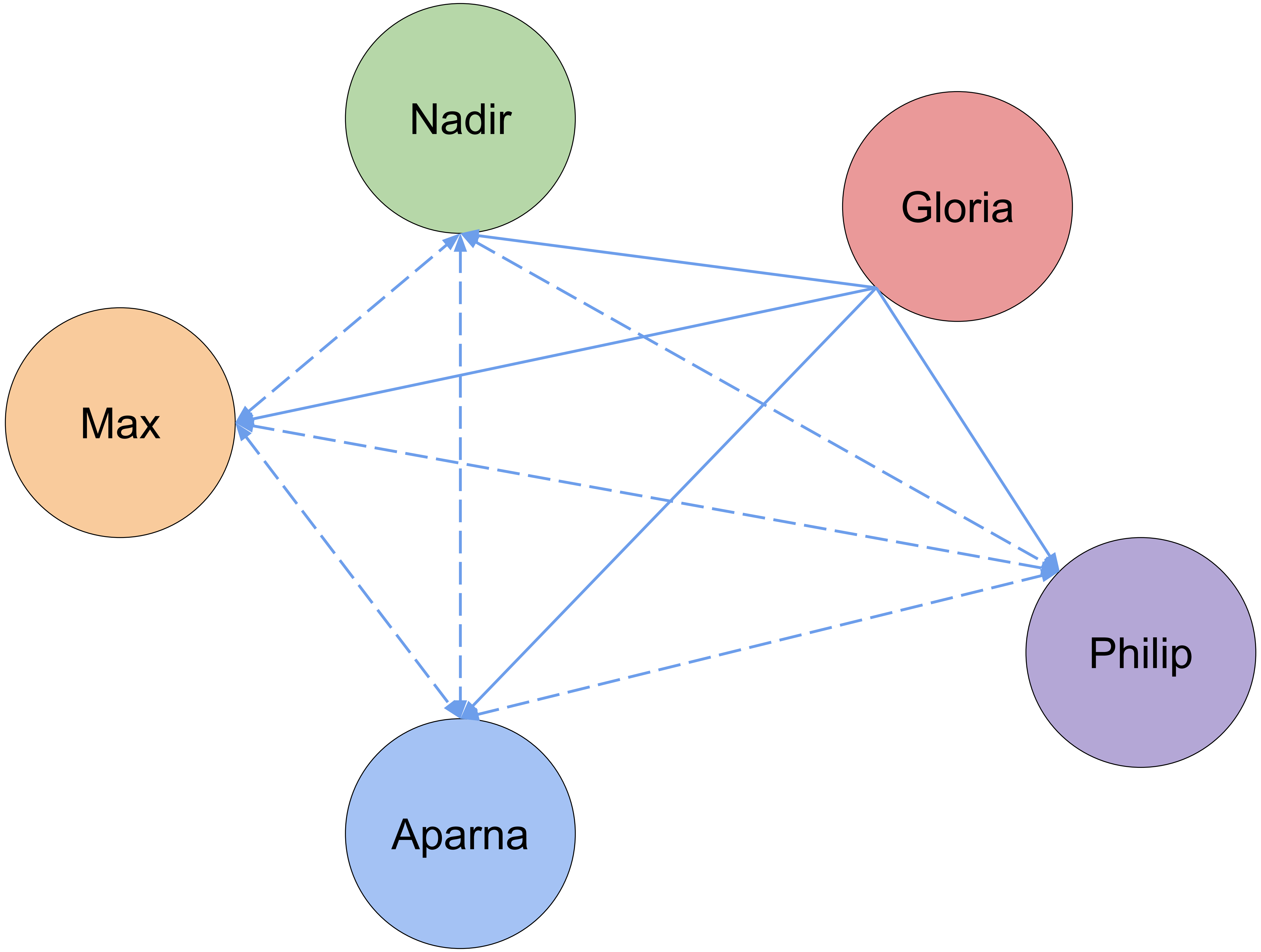


Gloria promises 10 BTC to Aparna in one transaction, and she promises 10 BTC to Nadir in another - but she only has 10 BTC total!

- Gloria is performing a *double spend* attack

CONSENSUS

PEER VALIDATION

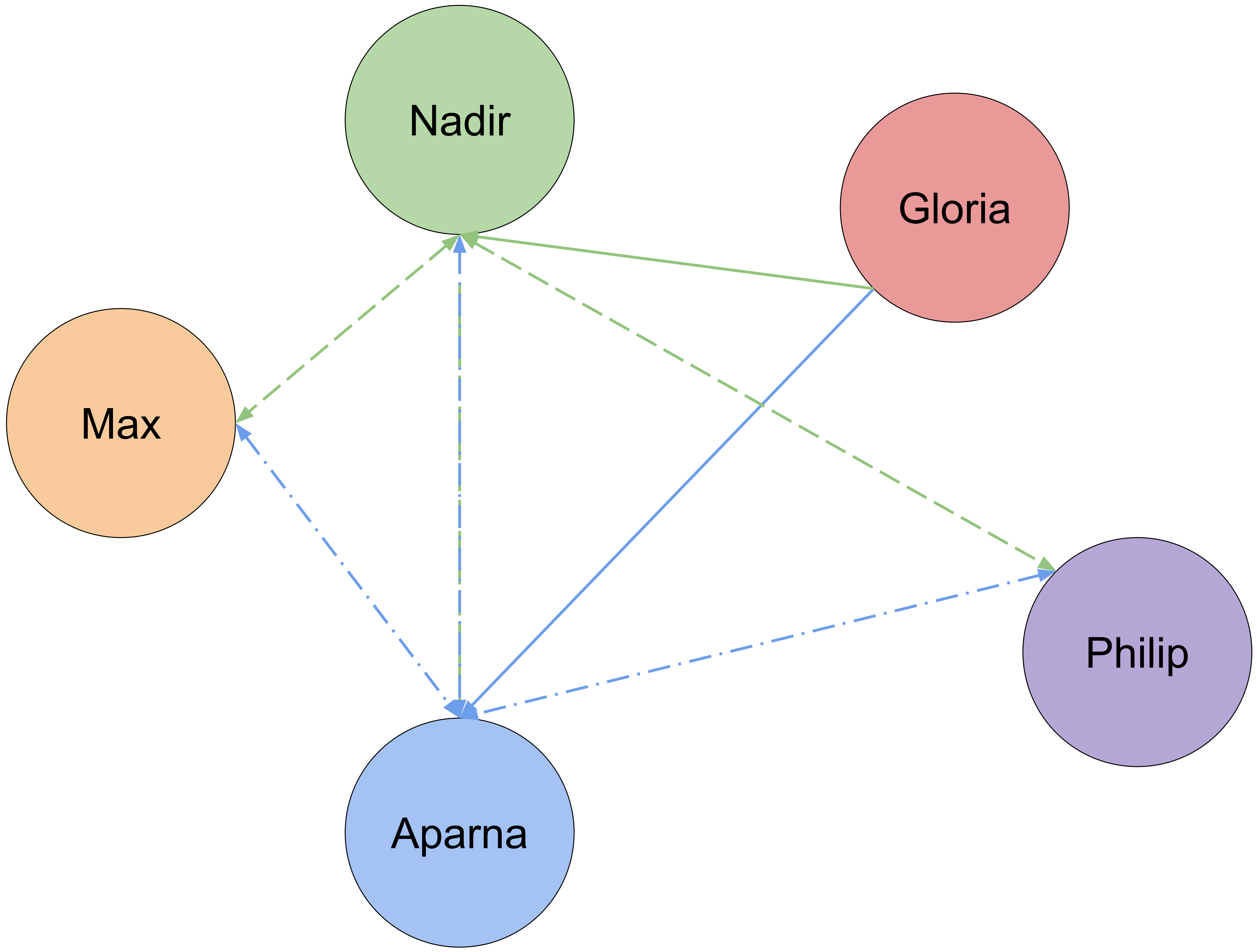


Instead of siloed decisions:

- The proposer submits a transaction to everyone else
- Peers cast votes
- Only valid after a certain number of votes

CONSENSUS

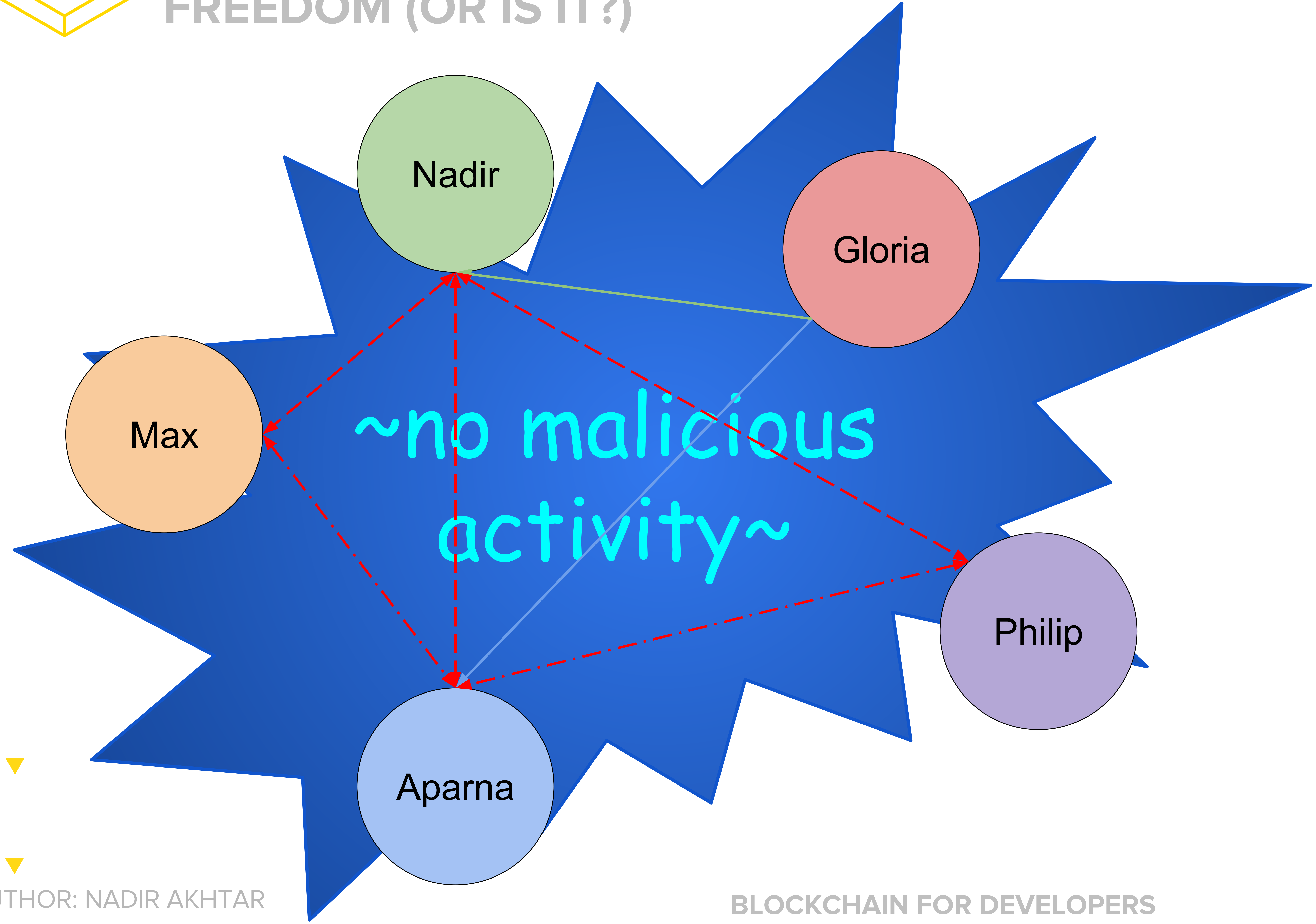
REJECTING THE DOUBLE SPEND



Now, when Gloria attempts to double spend, she will be rejected by observing peers

CONSENSUS

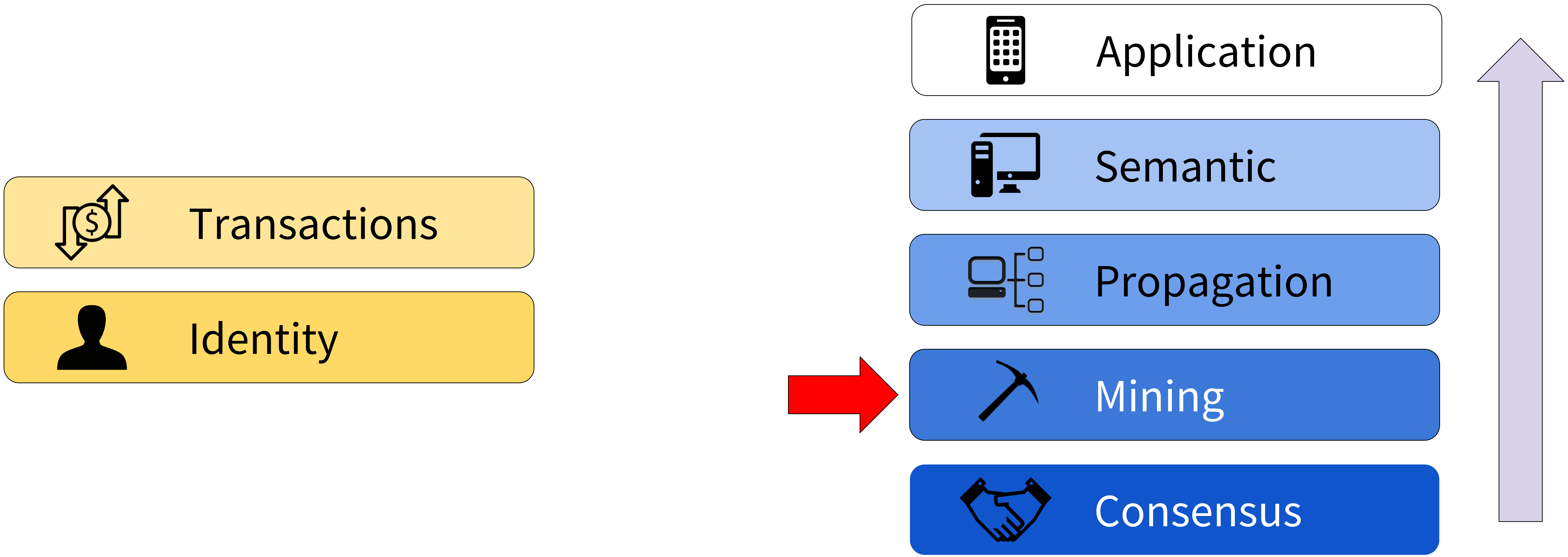
FREEDOM (OR IS IT?)



Peers vote “no” on Gloria’s proposal, as they notice multiple transactions trying to spend the same funds

BITCOIN: MINING

A MENTAL MODEL



2.3 MINING

WHAT A MINER DOES

MINERS VERIFY TRANSACTIONS

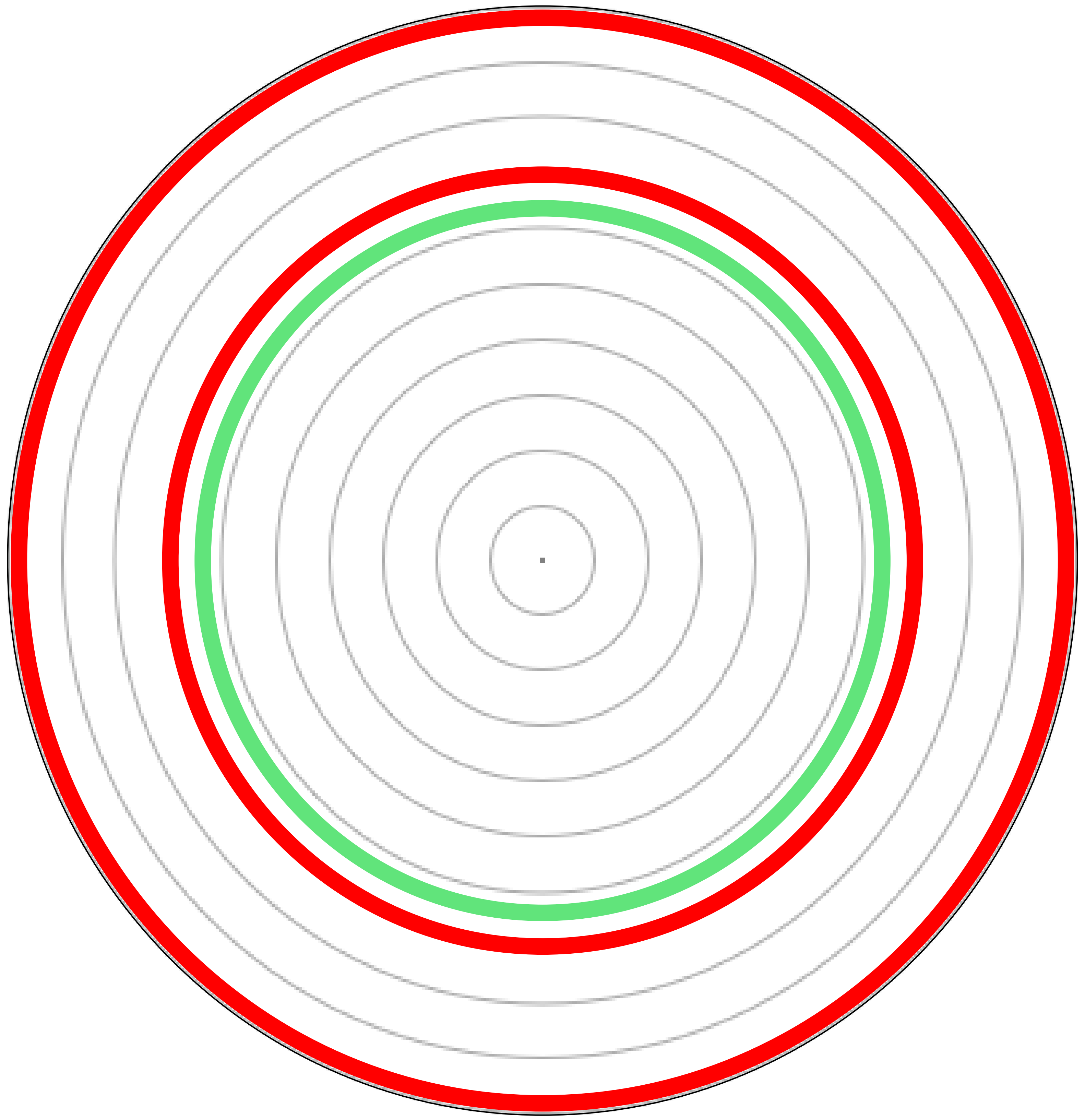
A Bitcoin miner must:

1. Download the entire Bitcoin blockchain to store the entire transaction history
2. Verify incoming transactions by checking signatures and confirming the existence of valid bitcoins
3. Create a block using collected valid transactions
4. Find a valid nonce to create a valid block header (the proof of work)
5. Hope that your block is accepted by other nodes and not defeated by a competitor block

BLOCK DIFFICULTY
DARTBOARD ANALOGY

Mining is like throwing darts at a target while blindfolded

- Equal likelihood of hitting any ring
- Faster throwers ⇒ more hits / second
- Target: within green ring
- Difficulty inversely proportional to green ring size
    - Green ring adjusts depending on average time to produce valid result
- If people get better at throwing darts, green circle needs to get smaller



H(nonce || prev_hash || merkle_root) < target)

WHAT A MINER DOES

RULES FOR THE PUZZLE

$\boxed{H(nonce \,||\, prev\_hash \,||\, merkle\_root) < target}$

Hash puzzles need to be:

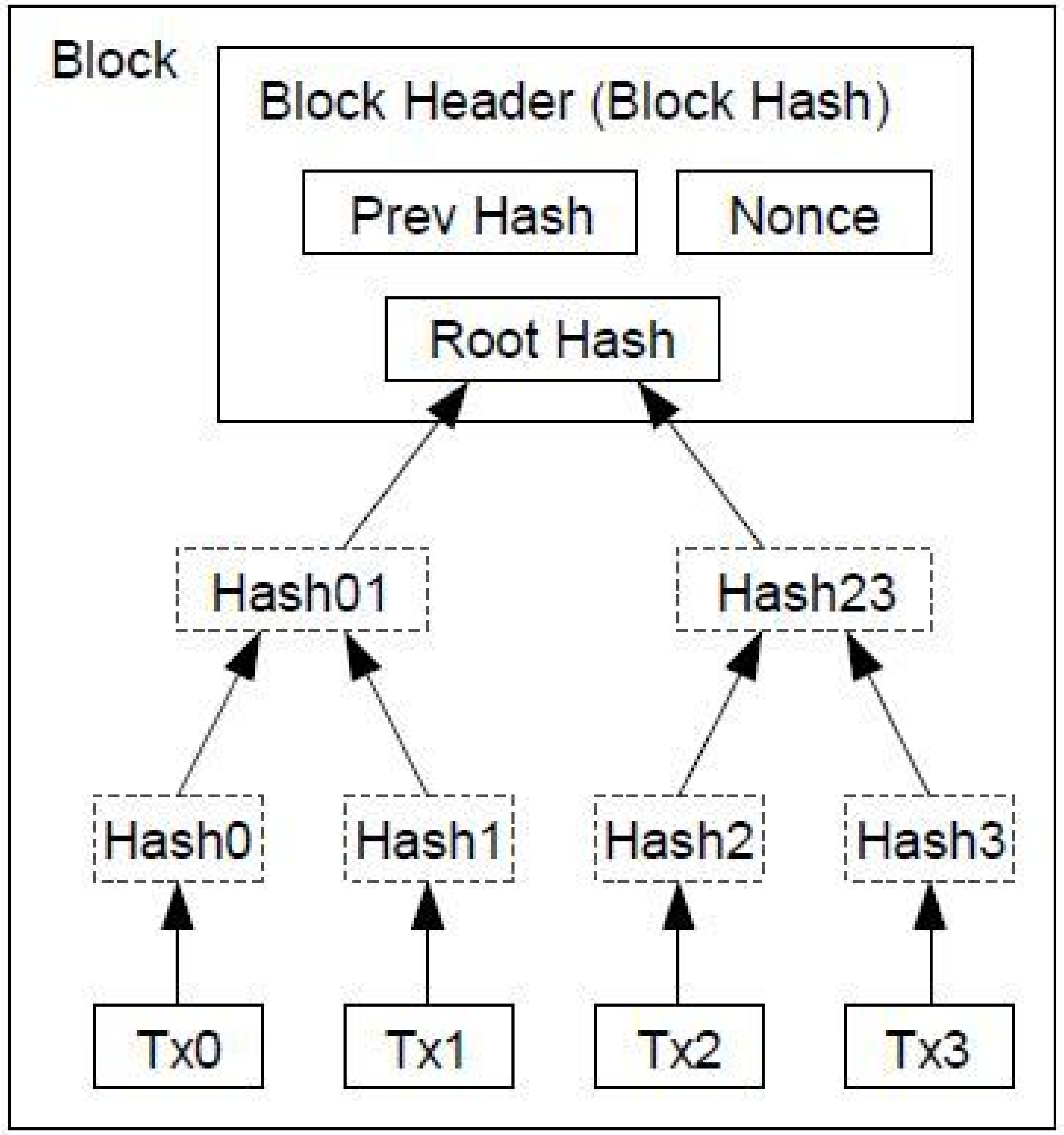
- **Computationally difficult**
  - If finding the proof-of-work requires little work, what's the point?
  - That's why we blindfold the dart-throwers
- **Parameterizable (variable) cost**
  - Allows for adjustments with global hashrate increases
- **Easily verifiable**
  - Should not be a need for a central authority to verify nonce validity; instead, other miners can rehash the nonce to verify validity

PUTTING IT ALL TOGETHER

THE PROCESS FOR CREATING A BLOCK

When miners try to compute a block, they pick all transactions they want to have added in the block, plus one *coinbase transaction* (contains the block reward) to their address

- Miners may include any transaction they want to form a *merkle tree* of transactions, from which we then take the *merkle root* of and reference that into the *block's header*
- For a block to be accepted by the network, it needs to contain only valid transactions: that means inputs that aren't spent, inputs that have a valid amount, valid signatures, etc.



Check out the block hashing algorithm!

PUTTING IT ALL TOGETHER

THE PROCESS FOR CREATING A BLOCK

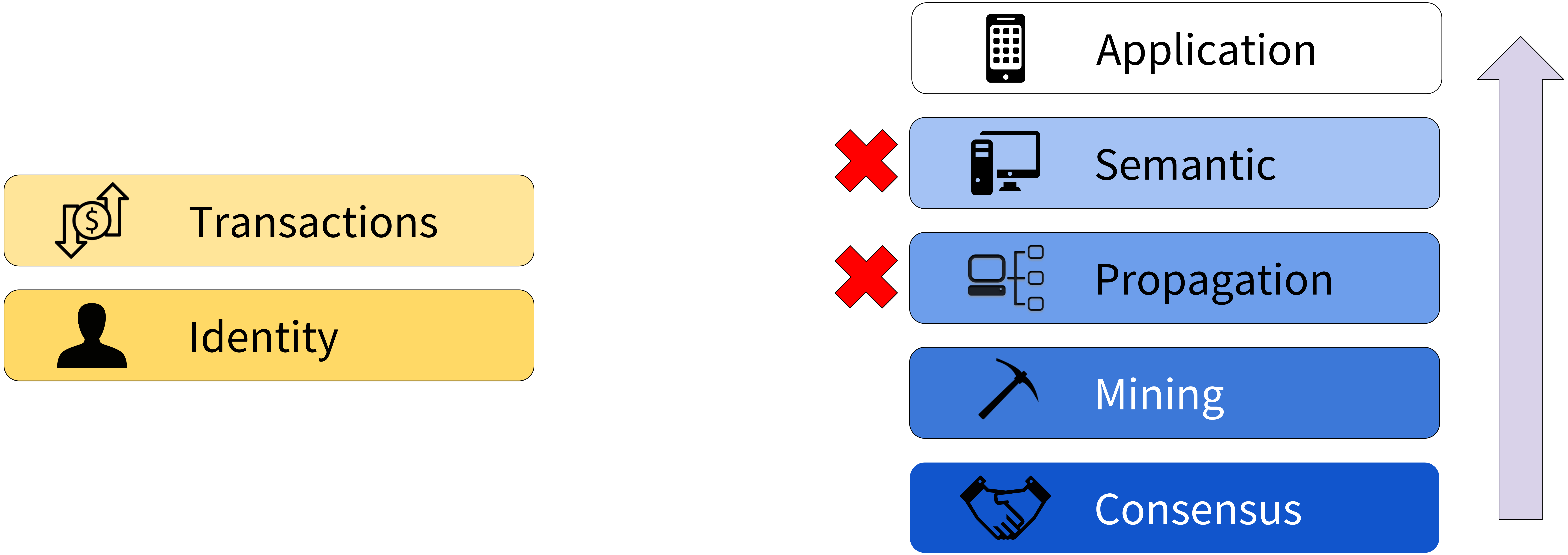
- Once the merkle root is validated, the *block header* is constructed
  - Block header: An 80-byte header belonging to a single block which is hashed repeatedly to create proof of work.
    - It contains Version, Hash of the previous block header, Merkle root, Timestamp, Bits (a representation of the network difficulty), Nonce (incremented when mining)
- Mining puzzle:

$$H(nonce \,||\, prev\_hash \,||\, merkle\_root) < target$$

  - Solve by incrementing nonce, until a hash less than the target (the difficulty threshold adjusted every 2016 blocks, referred to as nBits in code) is hit

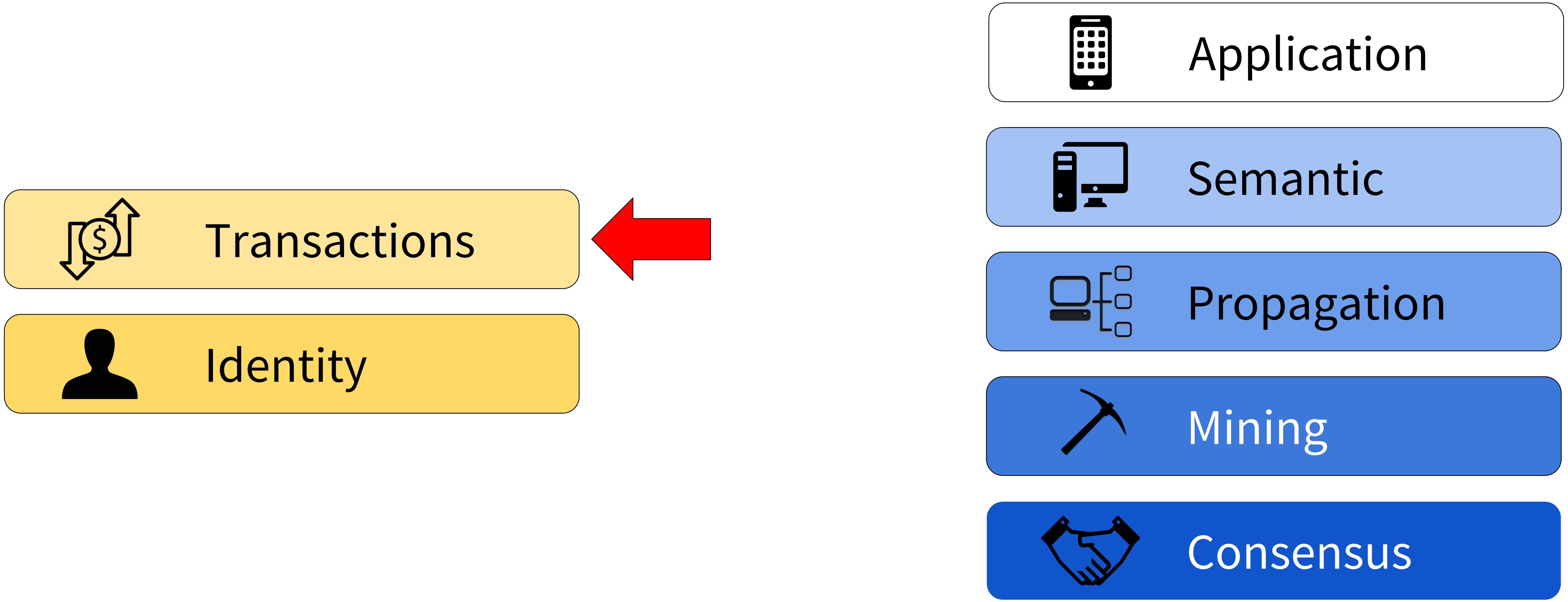
BITCOIN: PROPAGATION, SEMANTIC

A MENTAL MODEL



BITCOIN: TRANSACTIONS

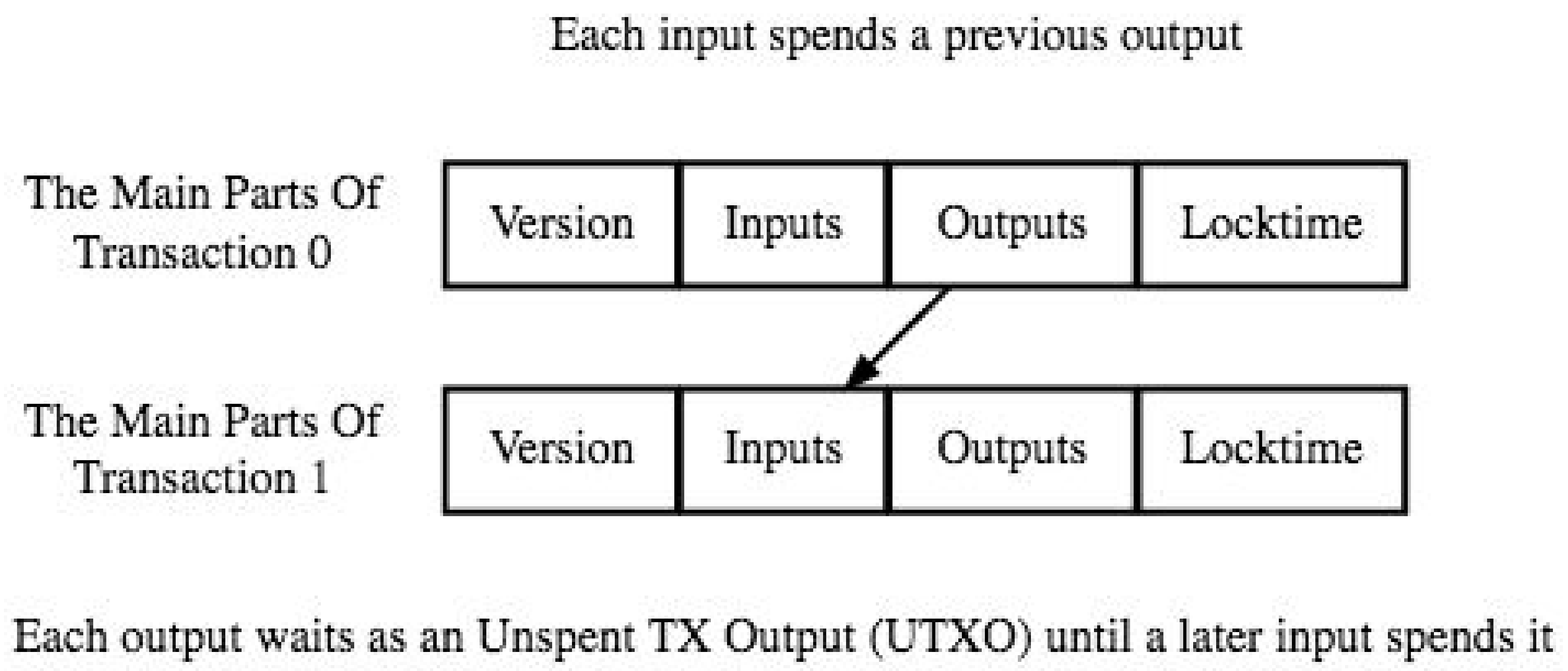
A MENTAL MODEL



2.4 TRANSACTIONS

TRANSACTIONS

SIMPLIFIED



What makes a transaction valid?

- Proof of ownership (a signature)
- Available funds
- No other transactions using the same funds

Instead of accounts like one might expect, Bitcoin uses an Unspent Transaction Output (UTXO) model to ensure that funds are used only once.

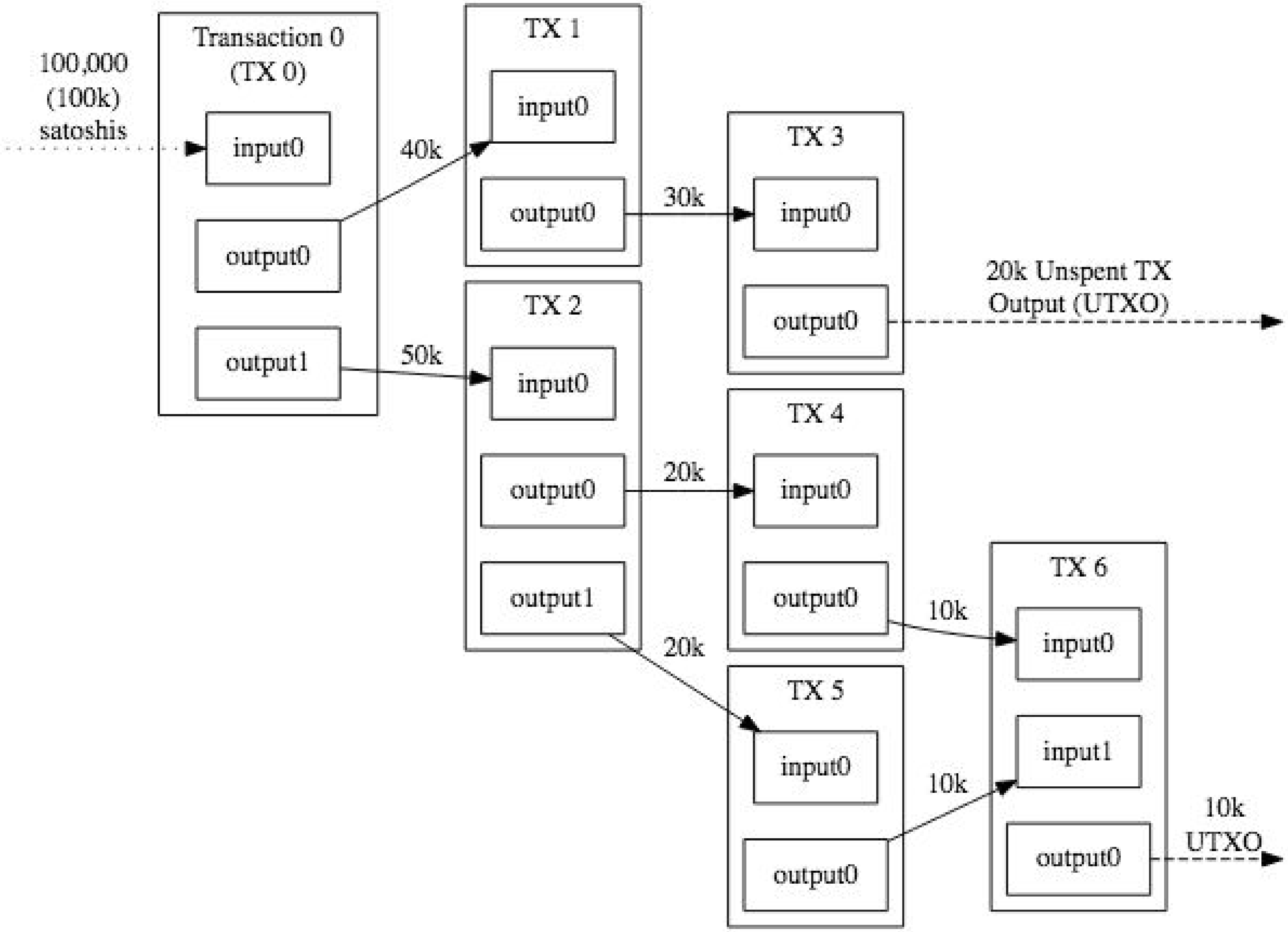
TRANSACTIONS

UTXO (UNSPENT TRANSACTION OUTPUT) MODEL

- UTXO; an unspent transaction output
- In an accepted transaction in a valid blockchain payment system (such as Bitcoin), only unspent outputs can be used as inputs to a transaction
- When a transaction takes place, inputs are deleted and outputs are created as new UTXOs that may then be consumed in future transactions

At a fundamental level, transactions:

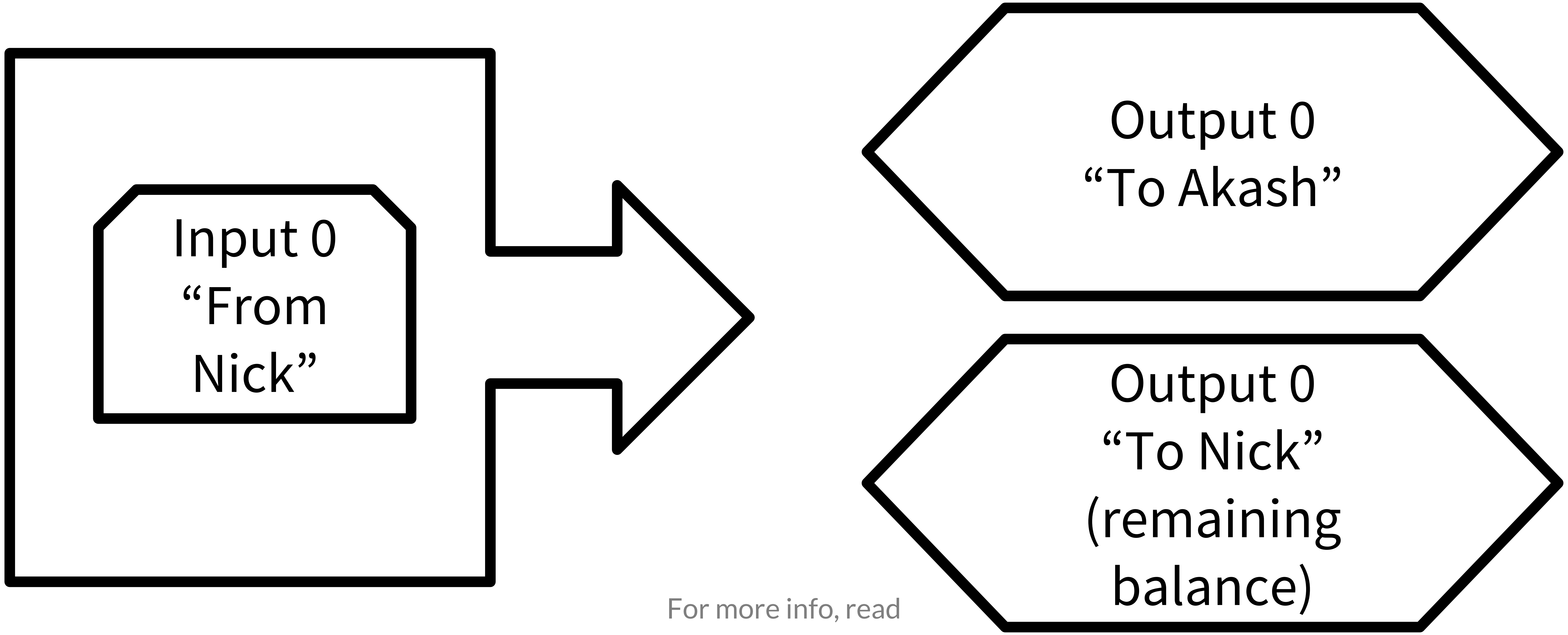
  - Map input addresses to output addresses
  - Typical tx: one input, two outputs
  - Contains signature of owner of funds



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

TRANSACTIONS

COMMON TRANSACTION

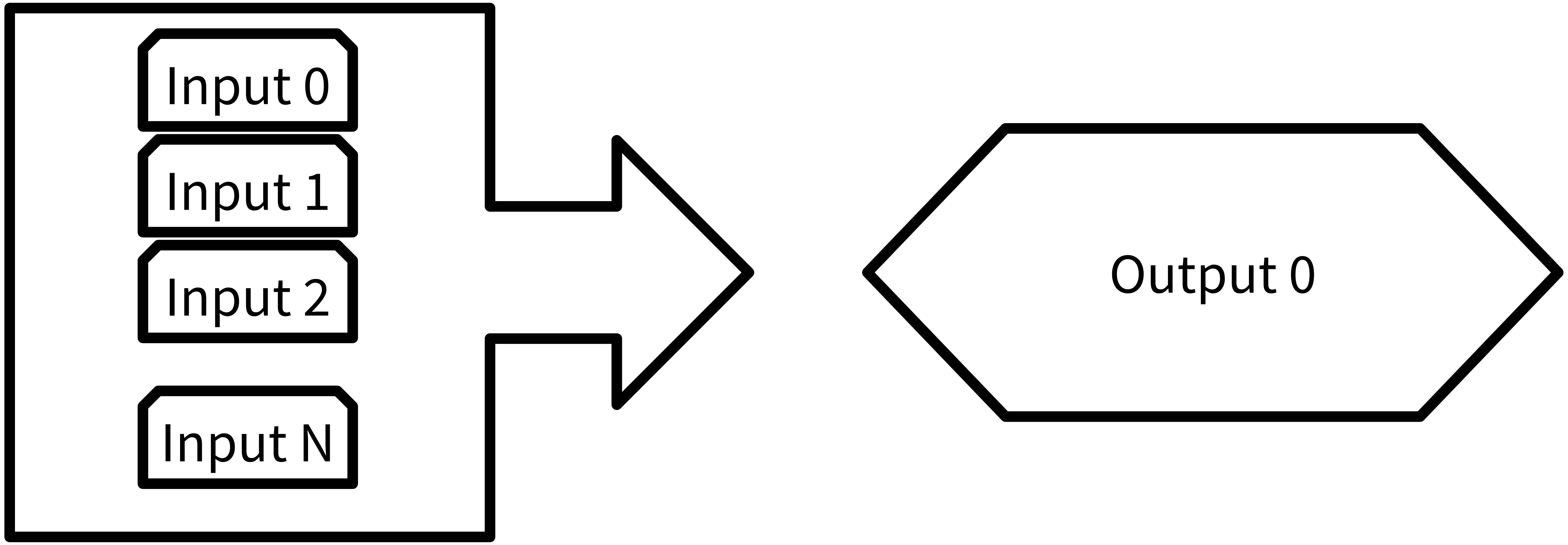


For more info, read

Mastering Bitcoin

TRANSACTIONS

AGGREGATING TRANSACTION

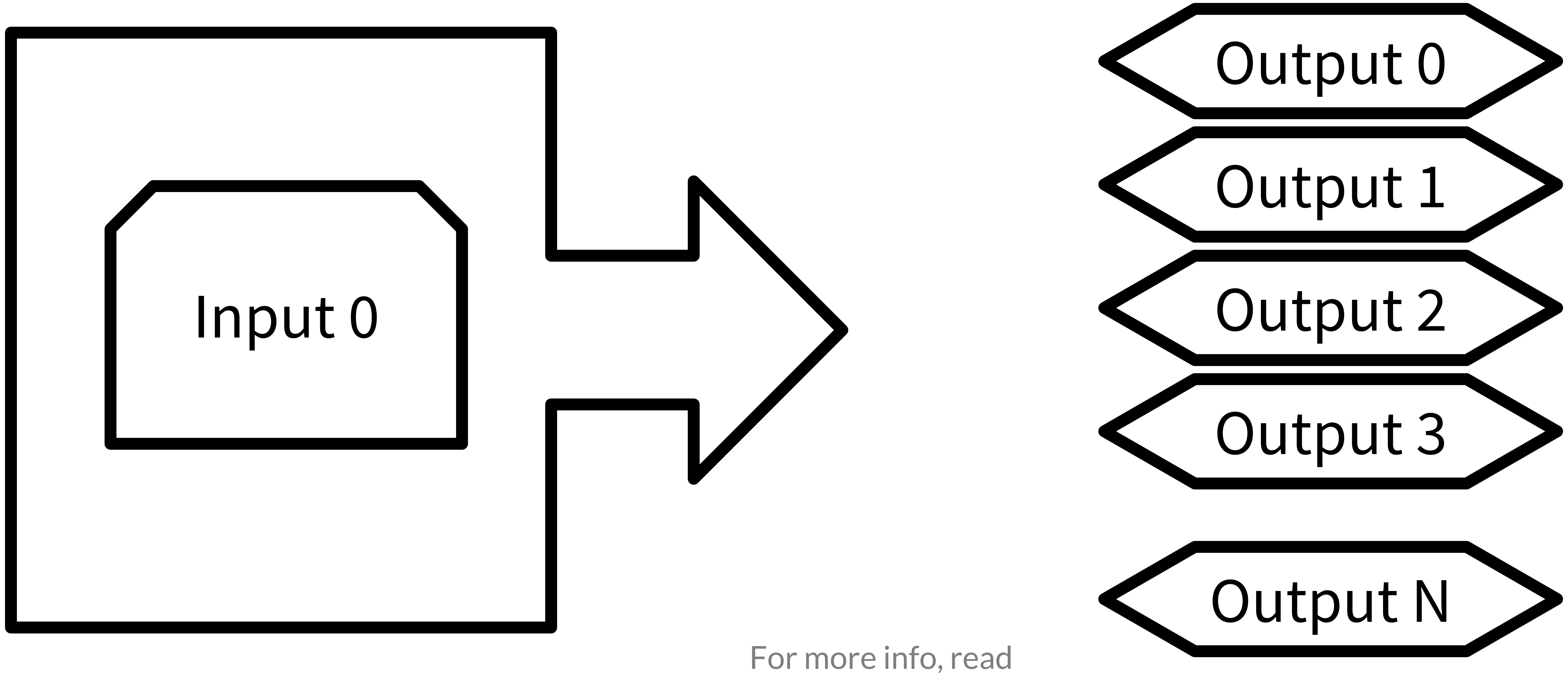


For more info, read

Mastering Bitcoin

TRANSACTIONS

DISTRIBUTING TRANSACTION



For more info, read

Mastering Bitcoin

TRANSACTIONS

UTXO (UNSPENT TRANSACTION OUTPUT) MODEL

- In Bitcoin, a UTXO is the amount that is transferred to a Bitcoin address (along with information required to unlock the output amount) during a transaction.
- Received amounts (UTXOs) are used individually during a transaction and new outputs are created:
  - One for the receiver and, if applicable, one for the amount that is left over (change output)
  - The amount sent to the recipient becomes a new UTXO in the recipient's address
    - The change output becomes a new UTXO in the sender's address that may be used in a future transaction (divisibility of Bitcoin: smallest unit is a *satoshi*, 10^{-8th} BTC)

TRANSACTIONS
THE REAL DEAL

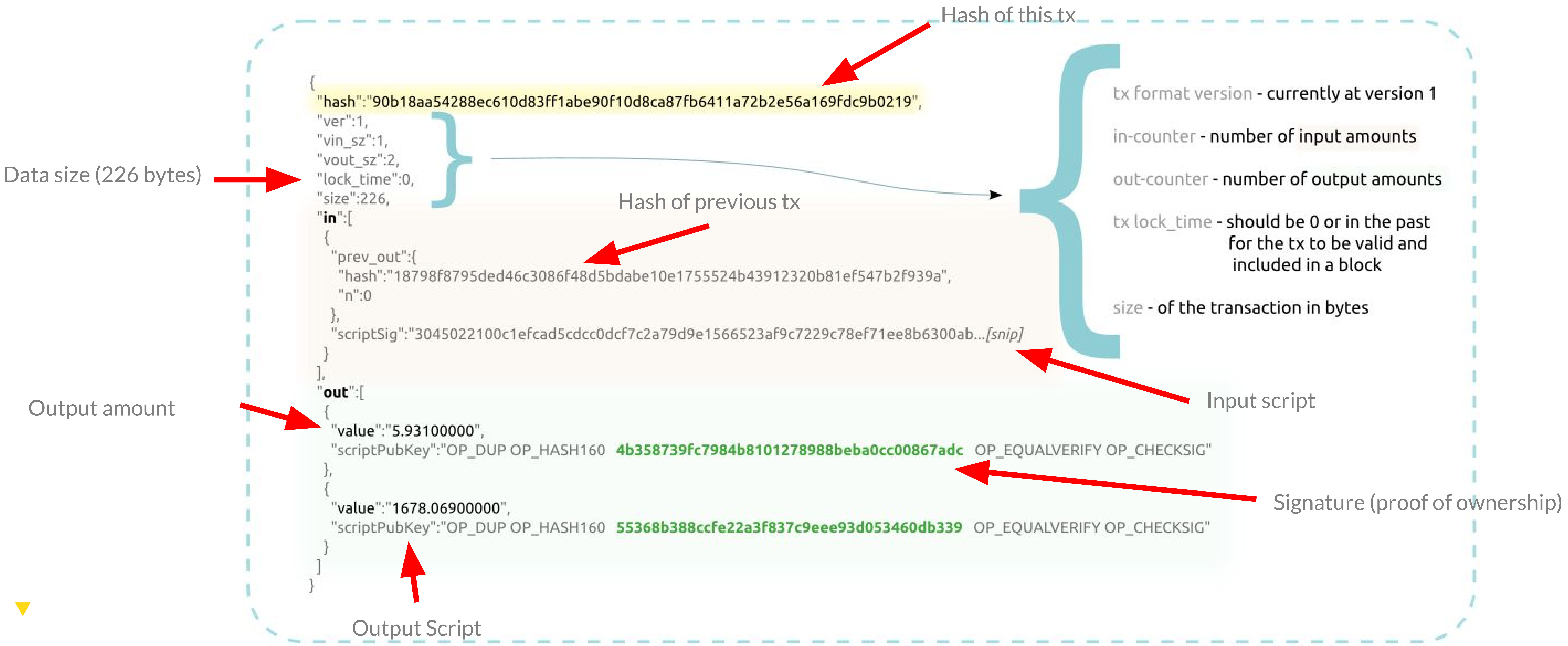
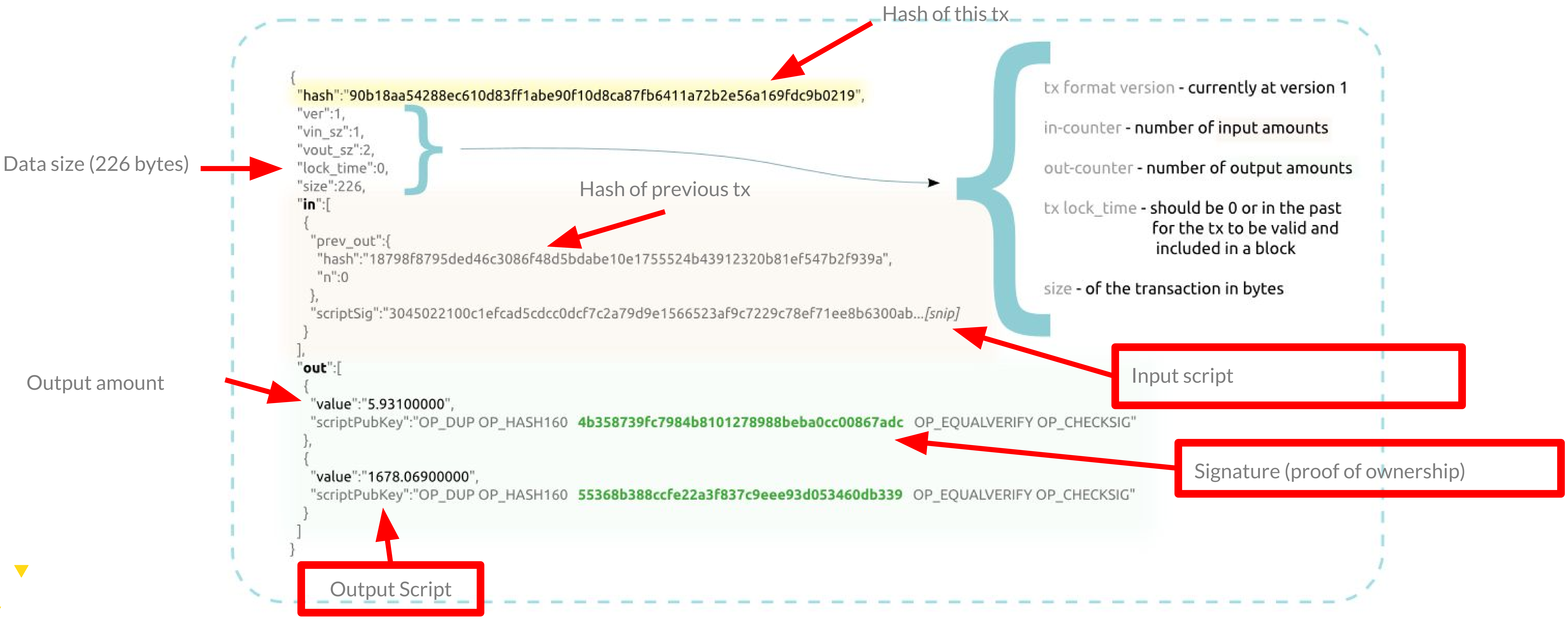


image by Venzen <venzen@mail.bihthai.net> 2014 CC SA
conditions of reuse: http://sofala.bihthai.net/works/txinout.htm

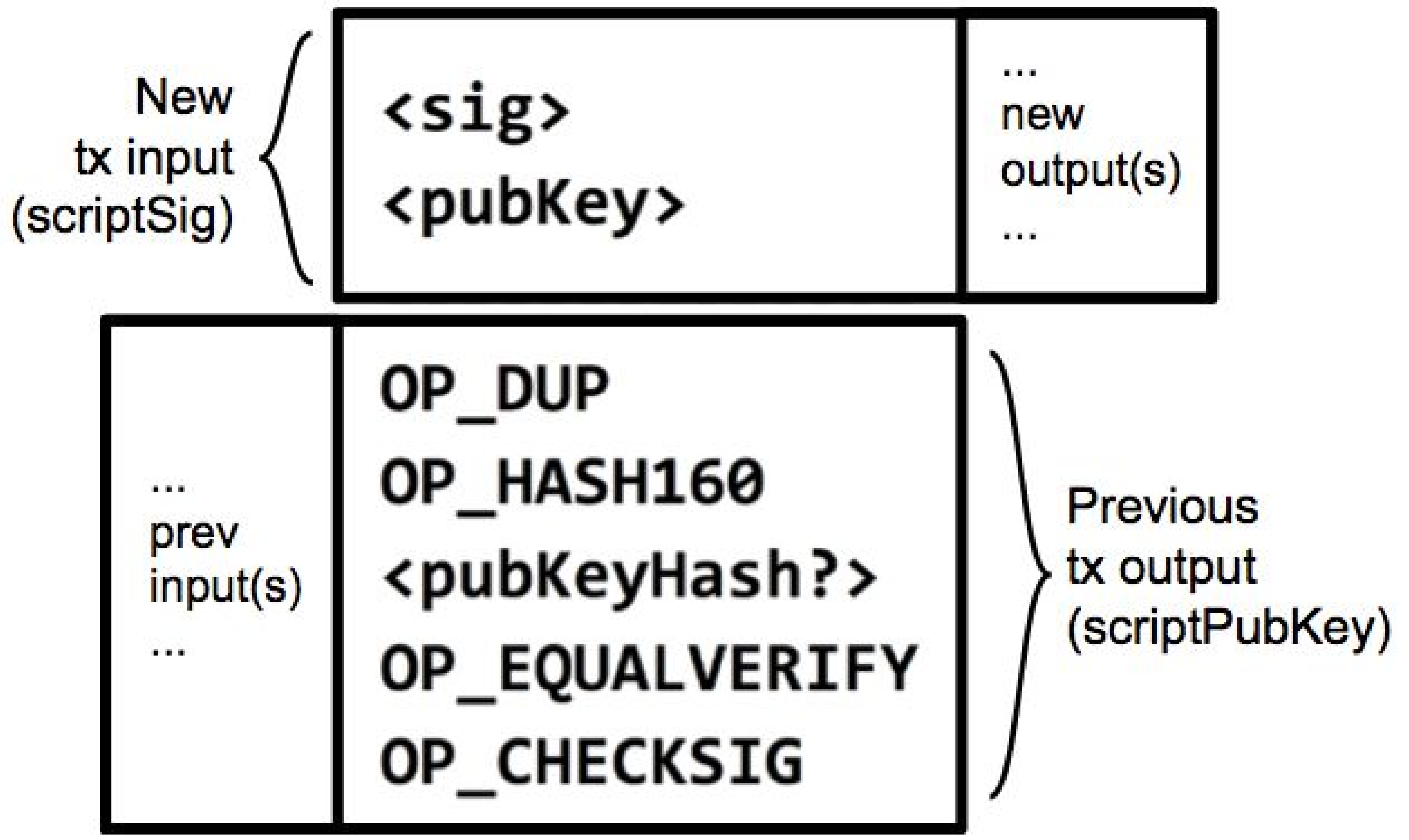
TRANSACTIONS

THE REAL DEAL



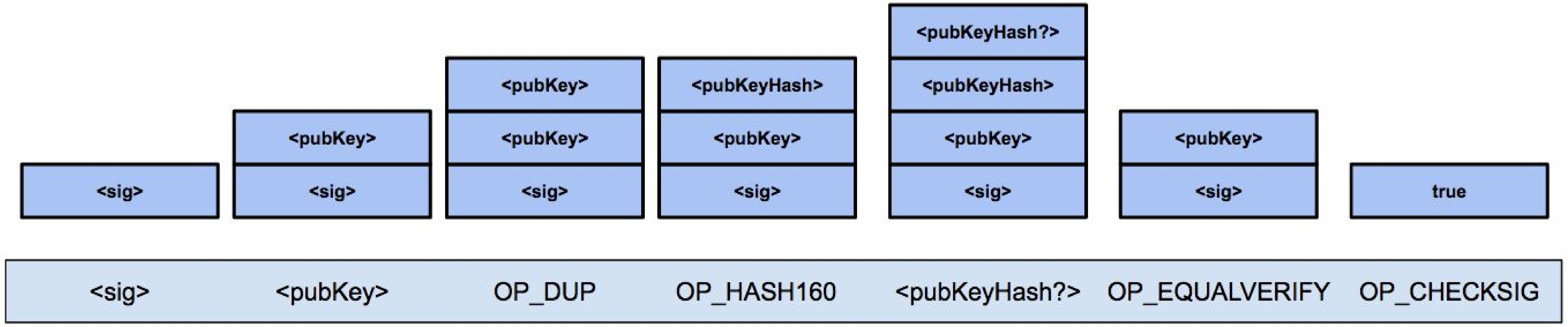
BITCOIN SCRIPT

HOW TRANSACTIONS REALLY WORK



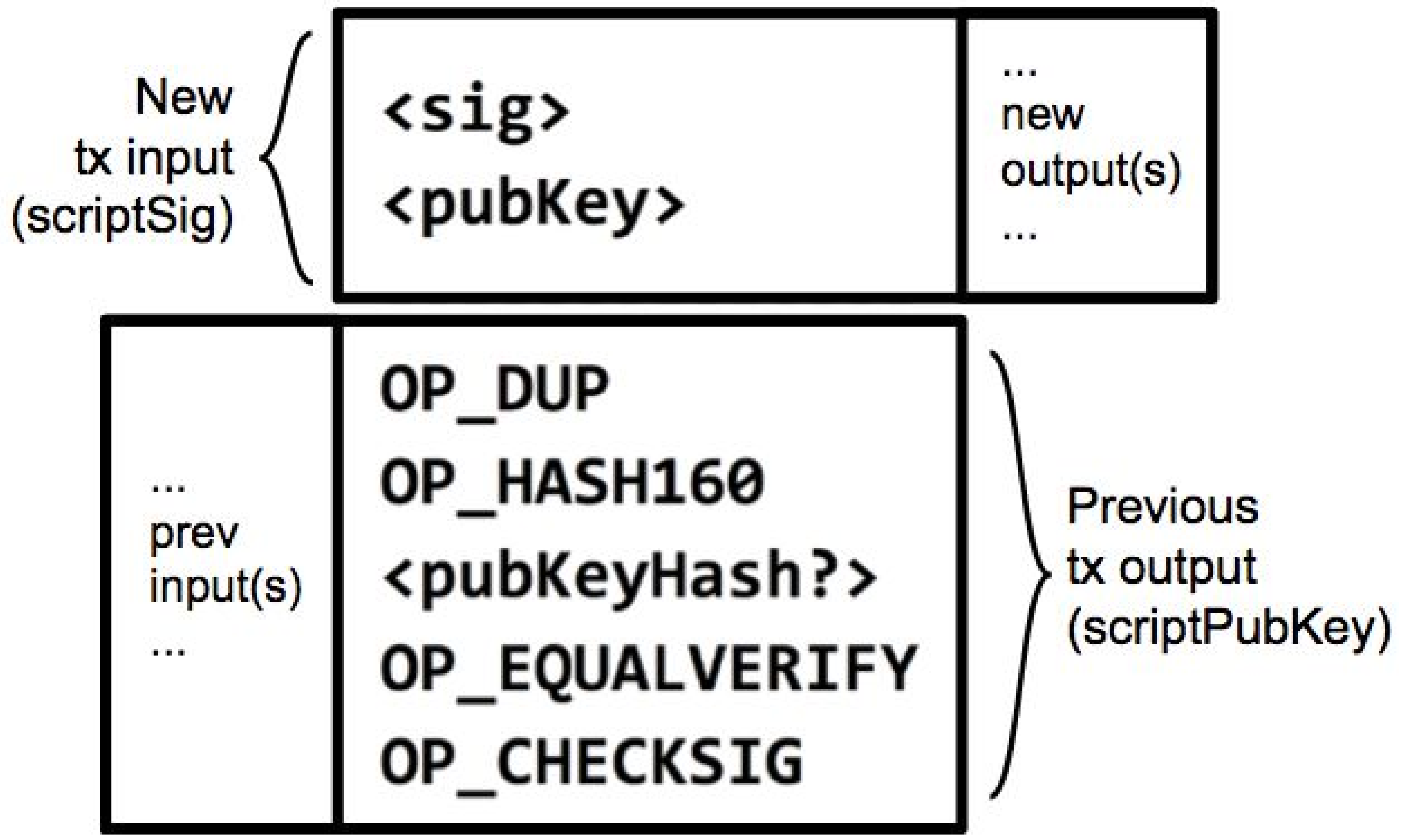
Language built specifically for Bitcoin called *Script*

- Stack based
- Native support for cryptography
- Simple - not turing complete (no loops)



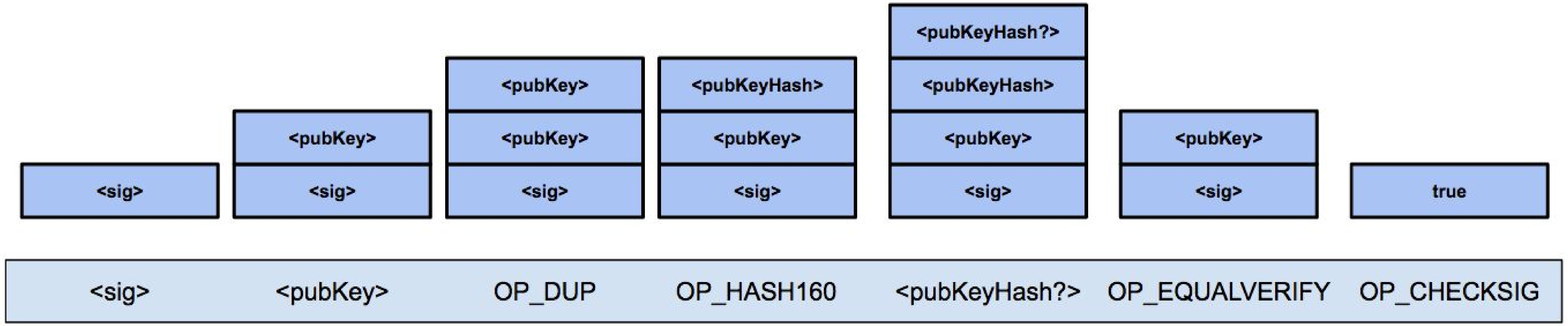
BITCOIN SCRIPT

HOW TRANSACTIONS REALLY WORK



- Output says: "This amount can be redeemed by
- 1) the **<pubKey>** that hashes to address **<pubKeyHash?>**
- 2) plus a **<sig>** from the owner of that **<pubKey>**
- ...that will make this script evaluate to **true**."

Read the Princeton's Bitcoin and Cryptocurrency Technologies for more information.



BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION

<sig>
<pubKey>

...
new
output(s)
...

...
prev
input(s)
...

OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG

<sig>

<sig>

AUTHOR: GLORIA ZHAO

BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION

<sig>
<pubKey>

...
new
output(s)
...

...
prev
input(s)
...

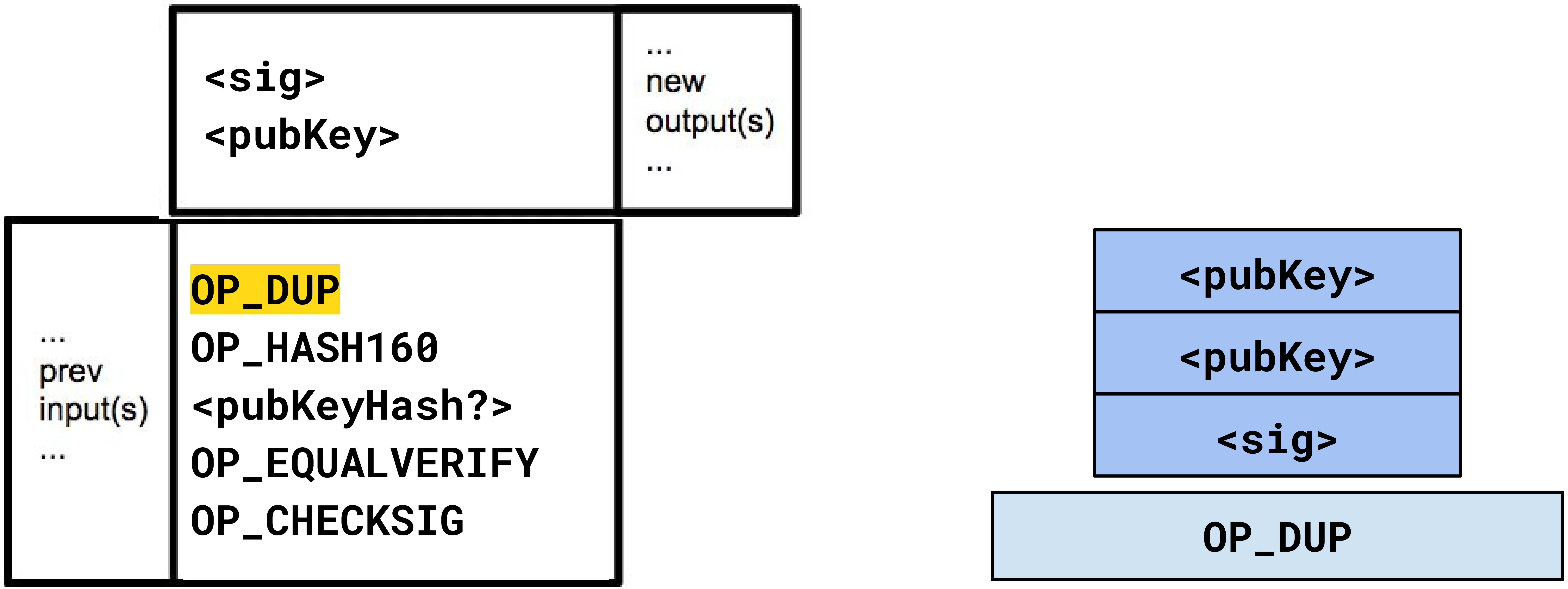
OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG

<pubKey>
<sig>
<pubKey>

AUTHOR: GLORIA ZHAO

BITCOIN SCRIPT

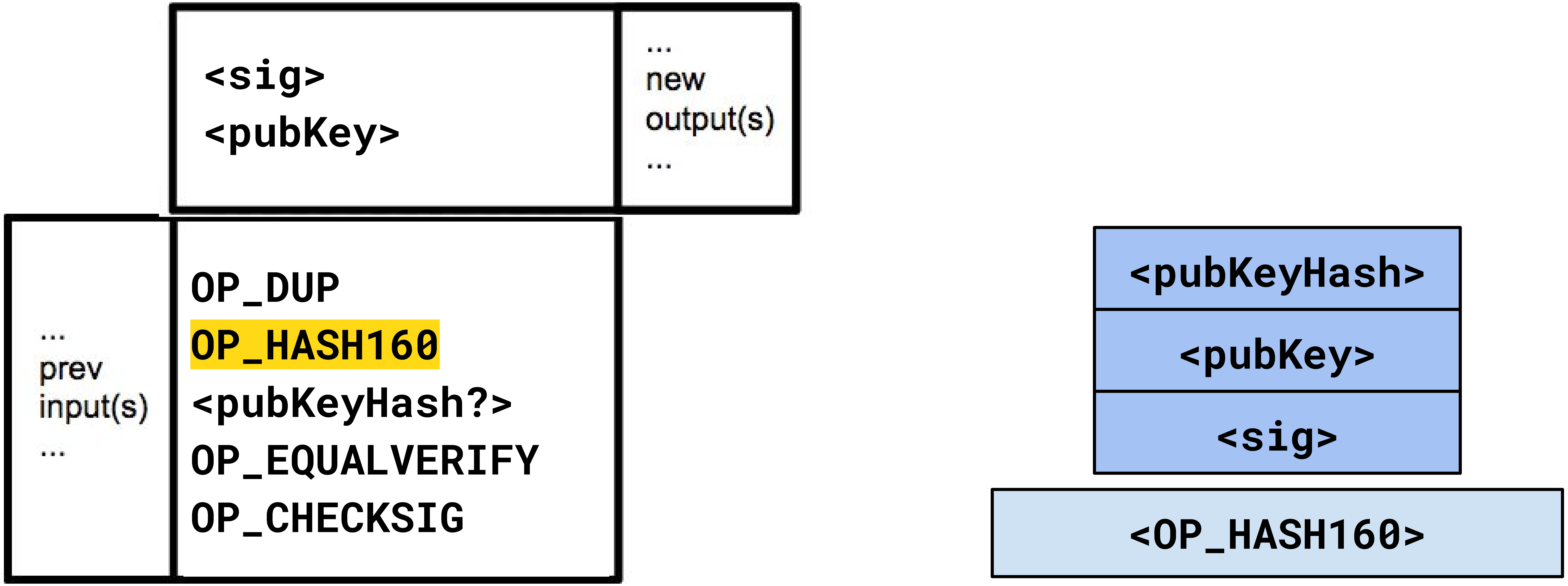
P2PKH EXAMPLE EXECUTION



AUTHOR: GLORIA ZHAO

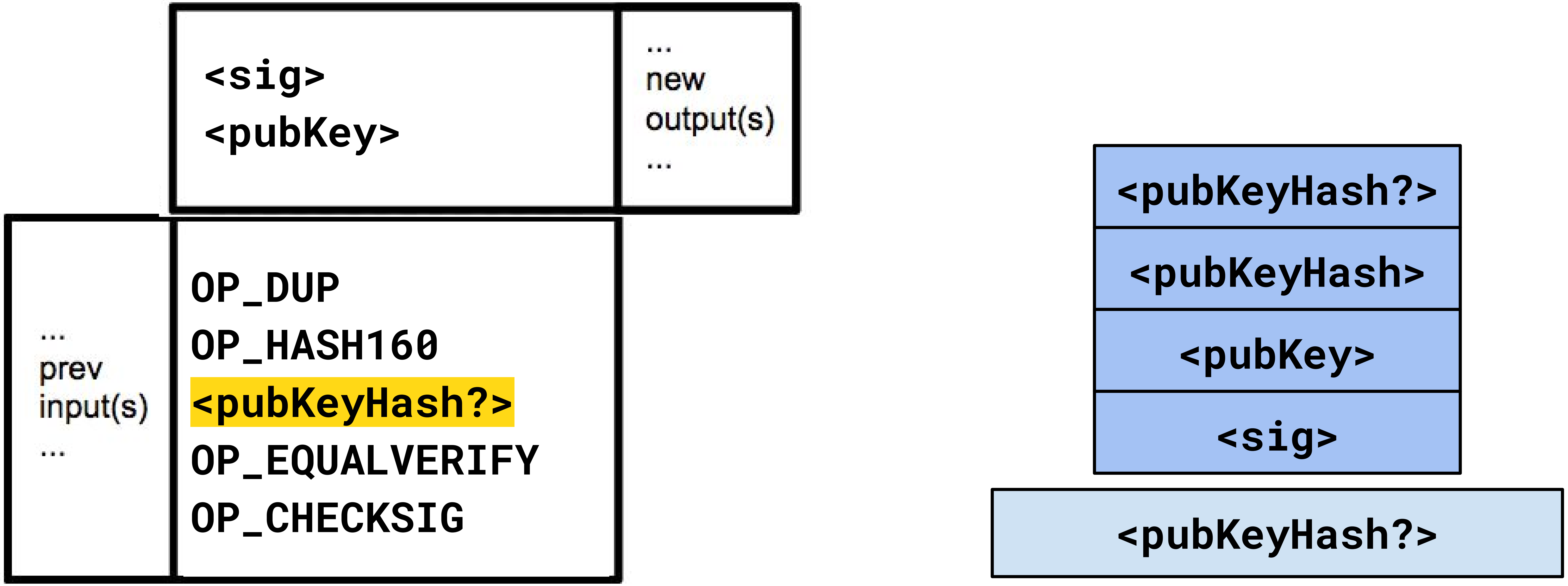
BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION



BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION



BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION

<sig>
<pubKey>

...
new
output(s)
...

...
prev
input(s)
...

OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG

<pubKey>
<sig>

OP_EQUALVERIFY

BITCOIN SCRIPT

P2PKH EXAMPLE EXECUTION

<sig>
<pubKey>

...
new
output(s)
...

...
prev
input(s)
...

OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG

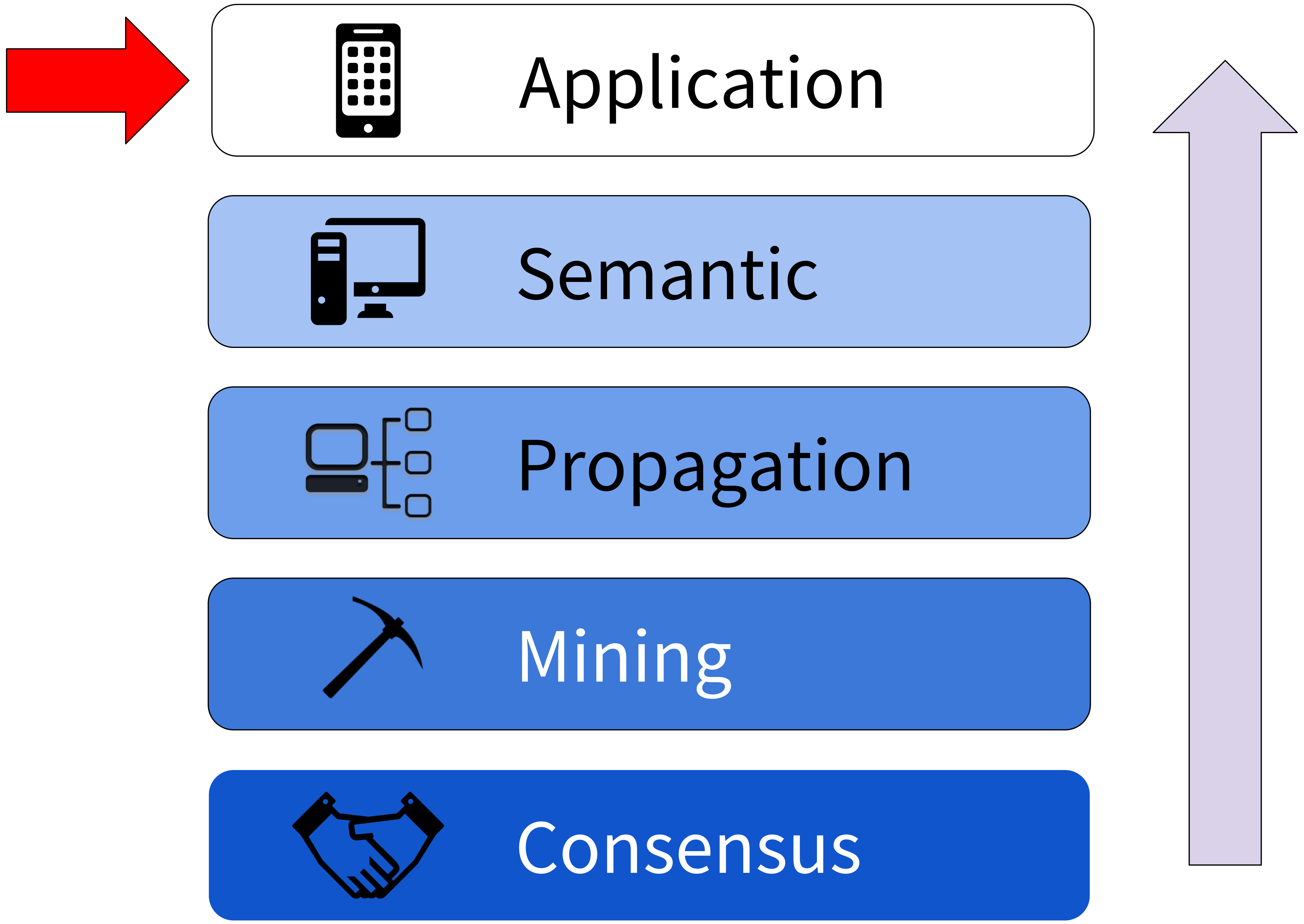
true

OP_CHECKSIG

AUTHOR: GLORIA ZHAO

BITCOIN: APPLICATION
WHAT'S NEXT?

- Transactions
- Identity



QUESTIONS?

SEE YOU NEXT TIME

Ethereum Mechanics

Smart Contracts

Account Model

Applications

Gas

Ethereum Virtual Machine

Solidity