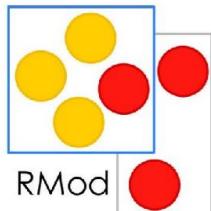


# Blockchain & Smart Contracts Crash Course

Santiago Bragagnolo ([santiago.bragagnolo@inria.fr](mailto:santiago.bragagnolo@inria.fr))



# About Me

- Santiago Bragagnolo([santiago.bragagnolo@inria.fr](mailto:santiago.bragagnolo@inria.fr))
- Engineer at Inria Lille (France, 2015 – 2020)
- Starting Ph.D. at Inria early 2020
- Working on blockchain since 2016
- Working in engineering since 2002

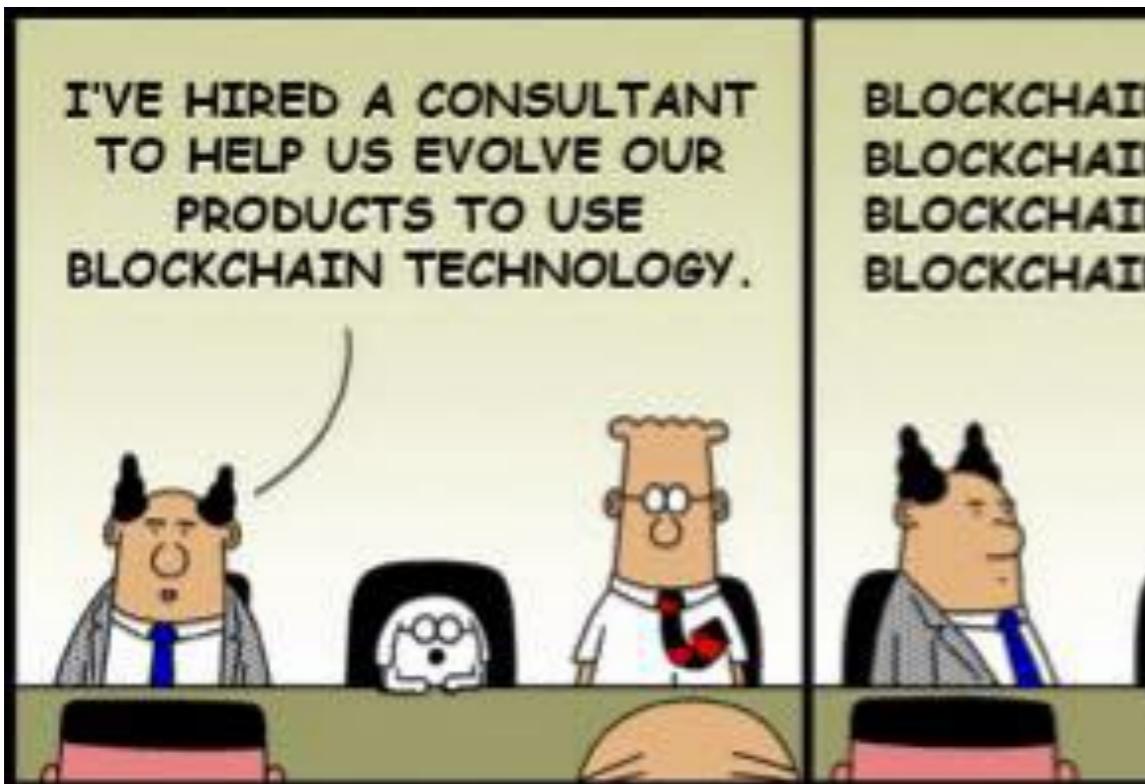
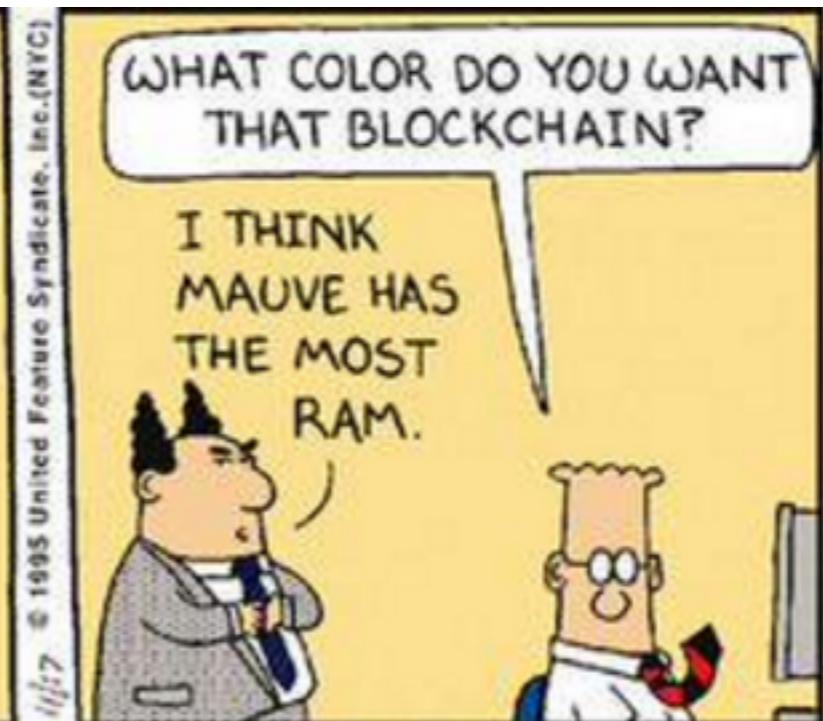
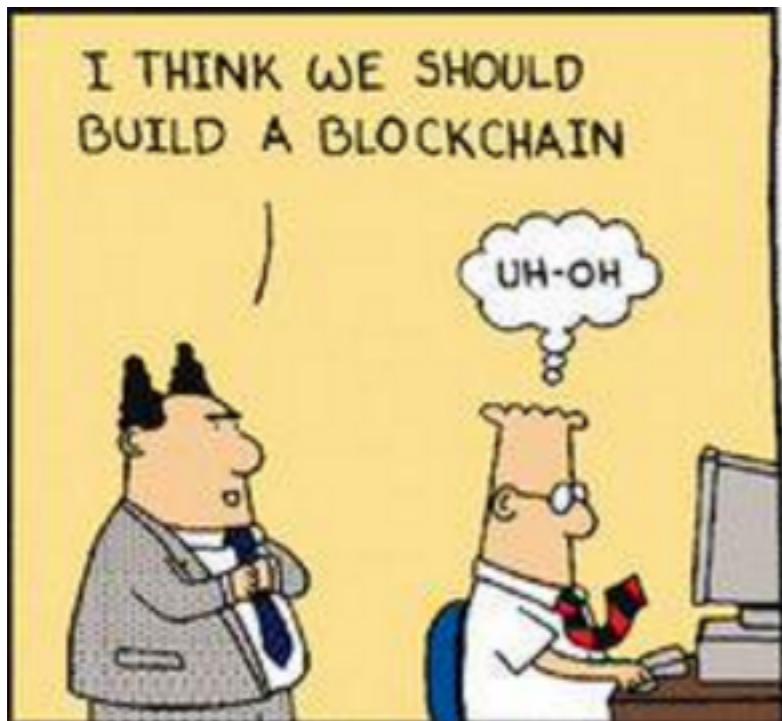
# This Tutorial at Polytech

- **1st part** (14:00 — 15:45): Presentation on Blockchain, Smart Contracts, RMOD team, etc.
- **2nd part** (16:00 — 18:00): Hands-on practices in the lab with Smart Contracts.
- This tutorial is supposed to be understandable even if a person has almost no previous knowledge on this topic.

# Summary (1st part)

- Introduction
- General Structure
- Mining
- Smart Contracts
- Some contributions from our team

# Introduction



# What is Blockchain?

 *bitcoin*

 ETHEREUM

**Examples of Blockchain Technologies**

# What is Blockchain?

## From an User's Perspective

- An open distributed Ledger



# What is Blockchain?

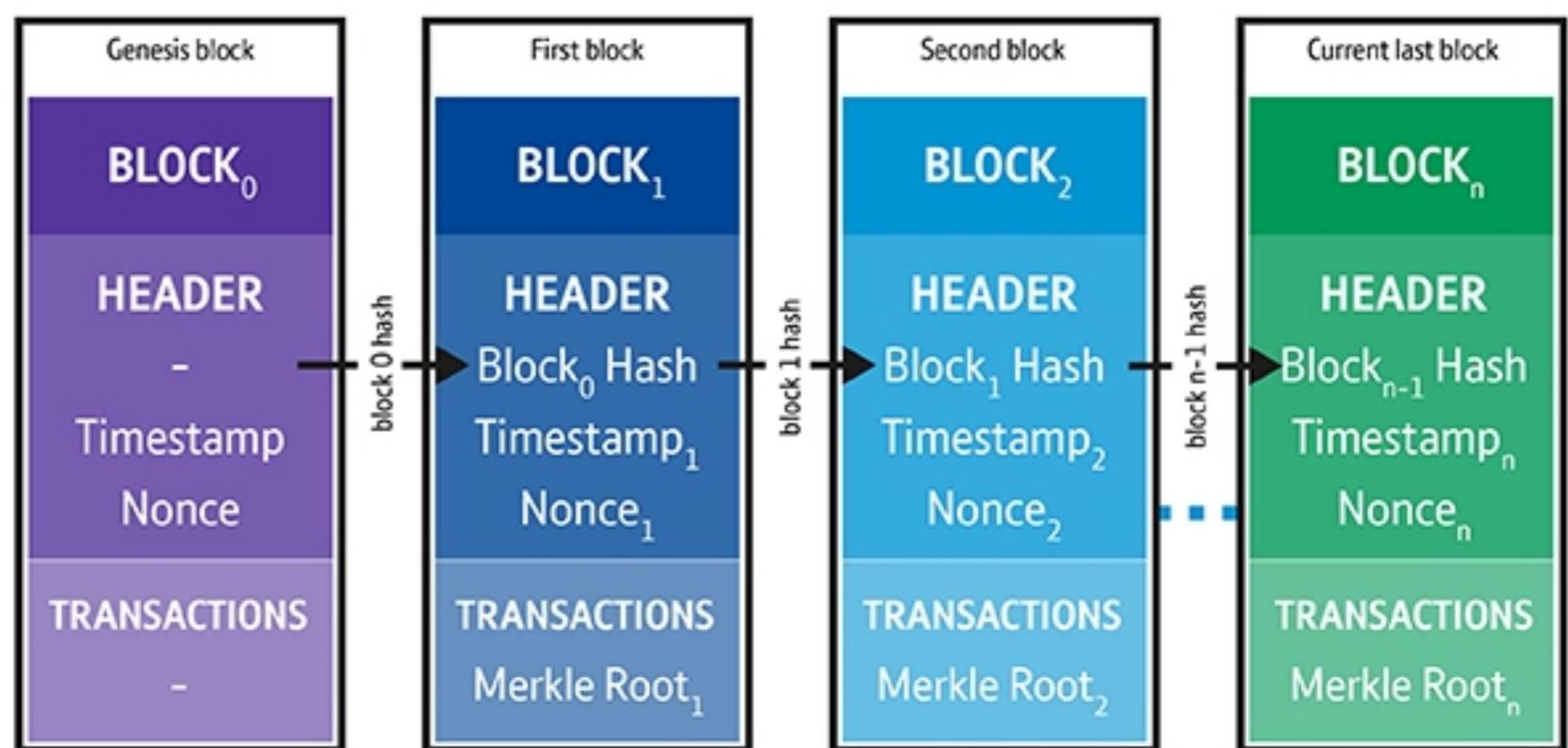
## From an User's Perspective

- An open distributed Ledger
  - Ledger - Records transactions (BitCoin, Ether).
  - Open - anyone can participate
  - Distributed - shared and synchronized among several “people”

# What is Blockchain?

## From a Developer's Perspective

- Roughly, an append-only distributed transactional database



# What is Blockchain?

## From a Developer's Perspective

- Roughly, an append-only distributed transactional database
  - Database
  - Transaction
  - Append-only
  - Distributed

# Structure

# Account

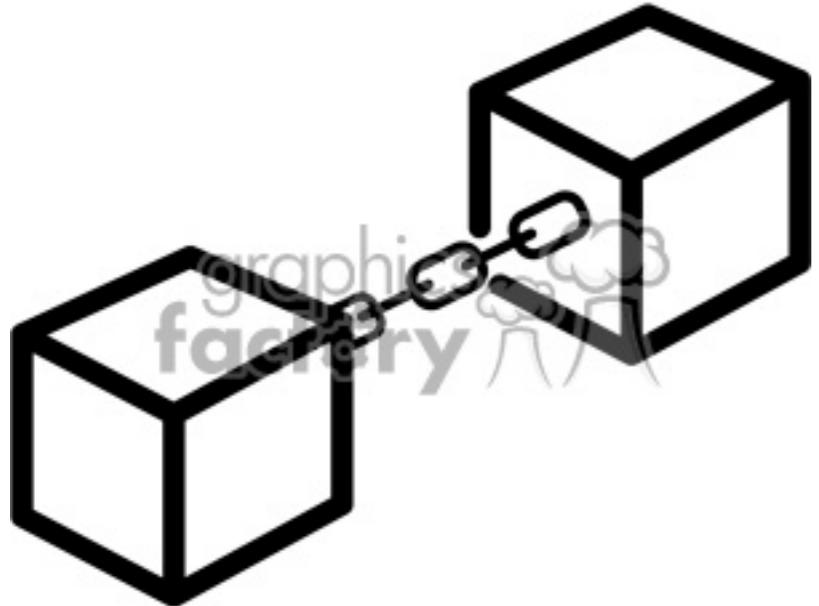
- We need an account to interact with the blockchain
- Roughly, an account's id/address is a public key and whoever holds the private key owns it.
- For example see: <<https://etherscan.io/accounts>>

# Transactions

A transaction is an operation that changes something on the blockchain.

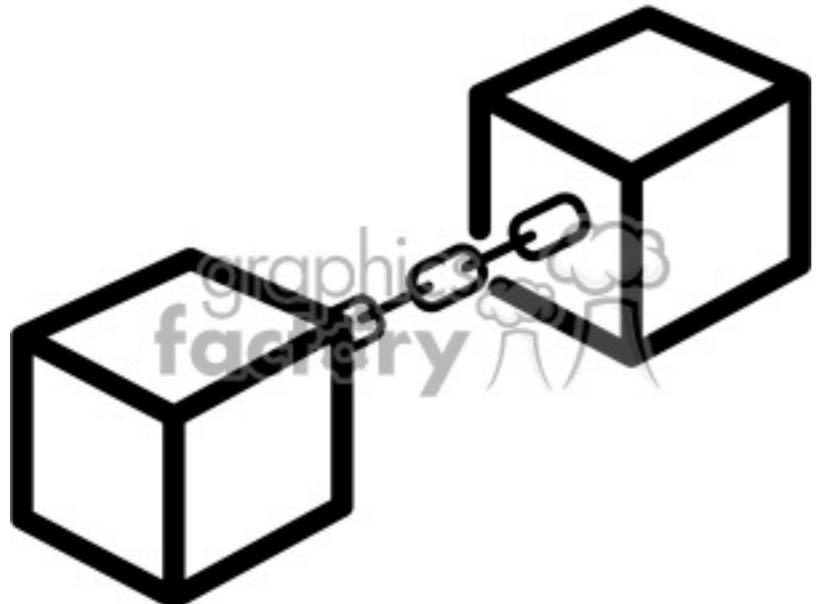
The information on transactions is public to everyone on the blockchain.

T.hash	From	To	Value
0x01...	0xFA...	0x9E...	25
0x22...	0x9E...	0x2F...	0
0xC3...	0x9E...	0x77...	7



# Blocks

- A block is storage unit in the blockchain database
  - Transactions are bundled into a block
  - Is linked to the previous block
  - It contains a timestamp



# Blocks

T.hash	From	To	Value
0x01...	0xFA...	0x9E...	25
0x22...	0x9E...	0x2F...	0
0xC3...	0x9E...	0x77...	7

**Block 245452**

<b>Hash</b> 0xd015f8981f292616e74...3b249b9a9240																
<b>TimeStamp</b> 2019-10-01T09:13:49.663753+02:00																
<b>Previous block</b> 0xf8bc8ede2727cce01b4...bda50490bd90																
<b>Nonce</b> 0x3afc82ca9063c17e																
<b>Transactions</b>																
<table><thead><tr><th>T.hash</th><th>From</th><th>To</th><th>Value</th></tr></thead><tbody><tr><td>0x01...</td><td>0xFA...</td><td>0x9E...</td><td>25</td></tr><tr><td>0x22...</td><td>0x9E...</td><td>0x2F...</td><td>0</td></tr><tr><td>0xC3...</td><td>0x9E...</td><td>0x77...</td><td>7</td></tr></tbody></table>	T.hash	From	To	Value	0x01...	0xFA...	0x9E...	25	0x22...	0x9E...	0x2F...	0	0xC3...	0x9E...	0x77...	7
T.hash	From	To	Value													
0x01...	0xFA...	0x9E...	25													
0x22...	0x9E...	0x2F...	0													
0xC3...	0x9E...	0x77...	7													

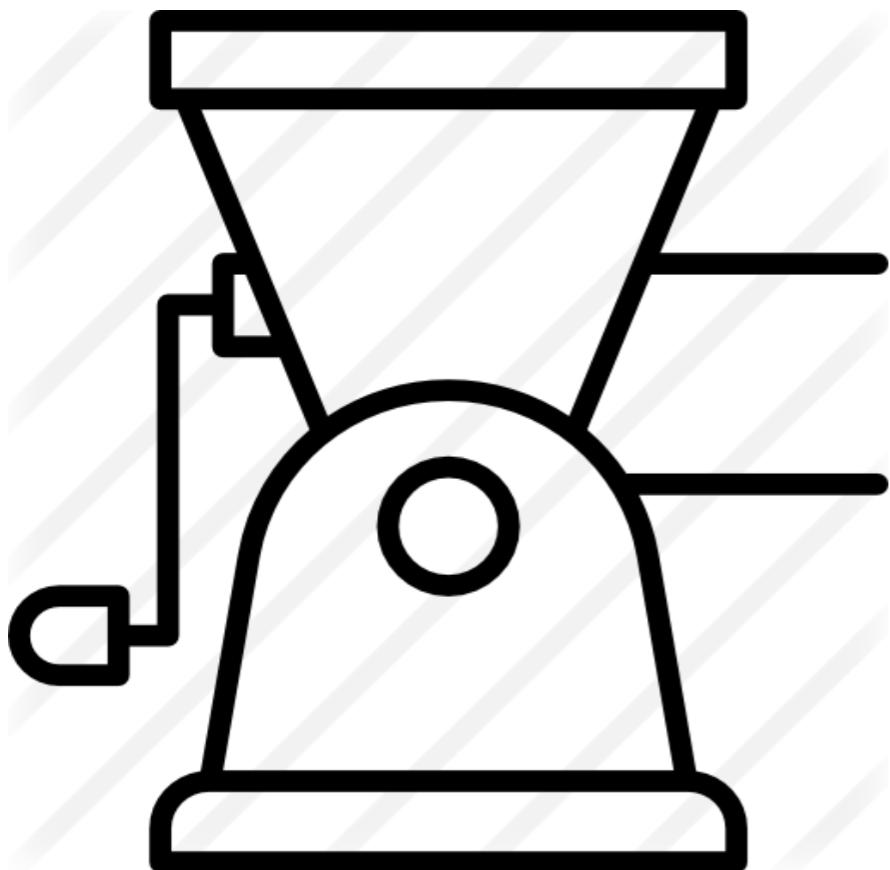
# How are the Blocks Linked?

**Previous block**  
0xf8bc8ede2727cce01b4...bda50490bd90

**TimeStamp**  
2019-10-01T09:13:49.663753+02:00

**Nonce**  
0x3afc82ca9063c17e

T.hash	From	To	Value
0x01...	0xFA...	0x9E...	25
0x22...	0x9E...	0x2F...	0
0xC3...	0x9E...	0x77...	7



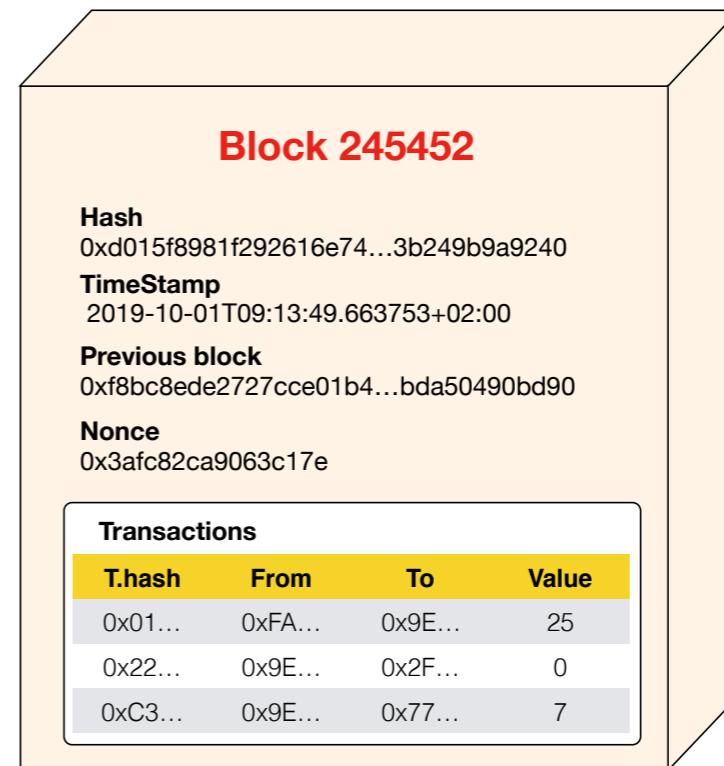
## HASH

0xd015f8981f292616e74...3b249b9a9240

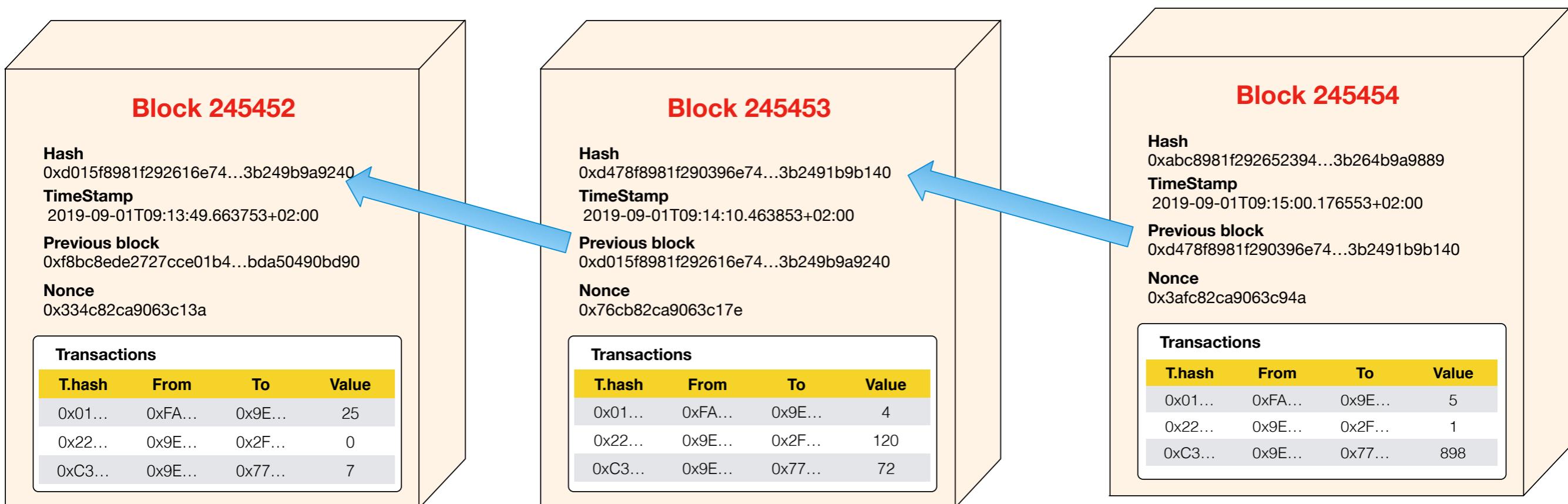


# How are the Blocks Linked?

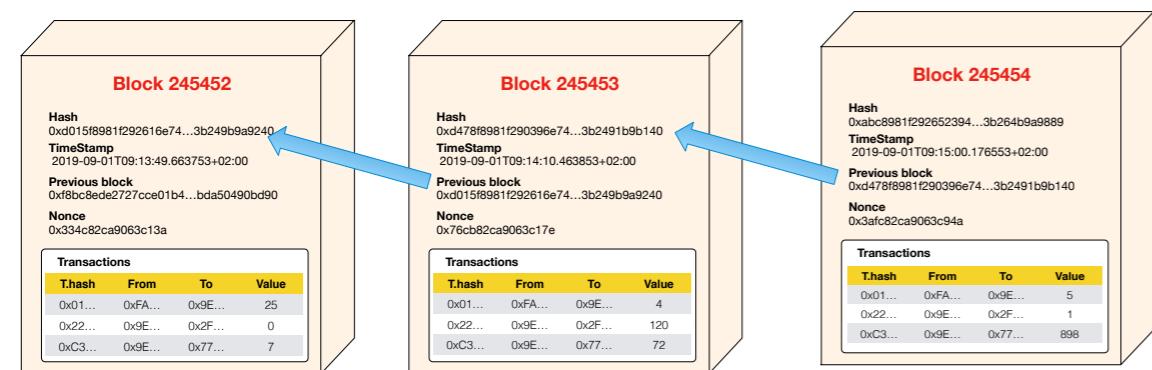
- The blocks are linked by a **Hash** number computed over the body of the block
  - Previous block's hash
  - The timestamp
  - The transactions
  - The nonce



# Linking Blocks Example



# Linking Blocks Example



## TimeStamp

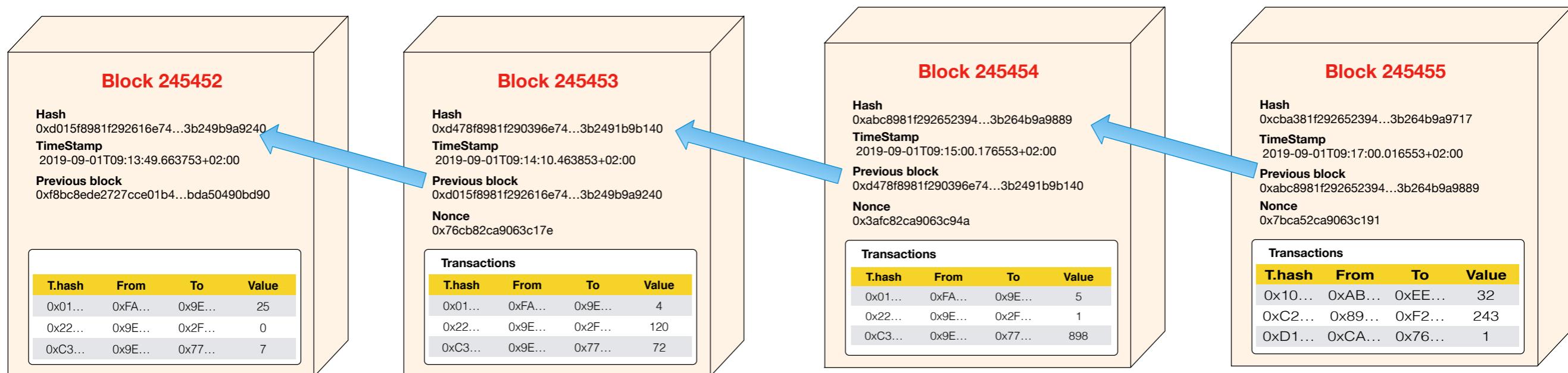
2019-09-01T09:17:00.016553+02:00

## Previous block

0xabc8981f292652394...3b264b9a9889

T.hash	From	To	Value
0x10...	0xAB...	0xEE...	32
0xC2...	0x89...	0xF2...	243
0xD1...	0xCA...	0x76...	1

# Linking Blocks Example

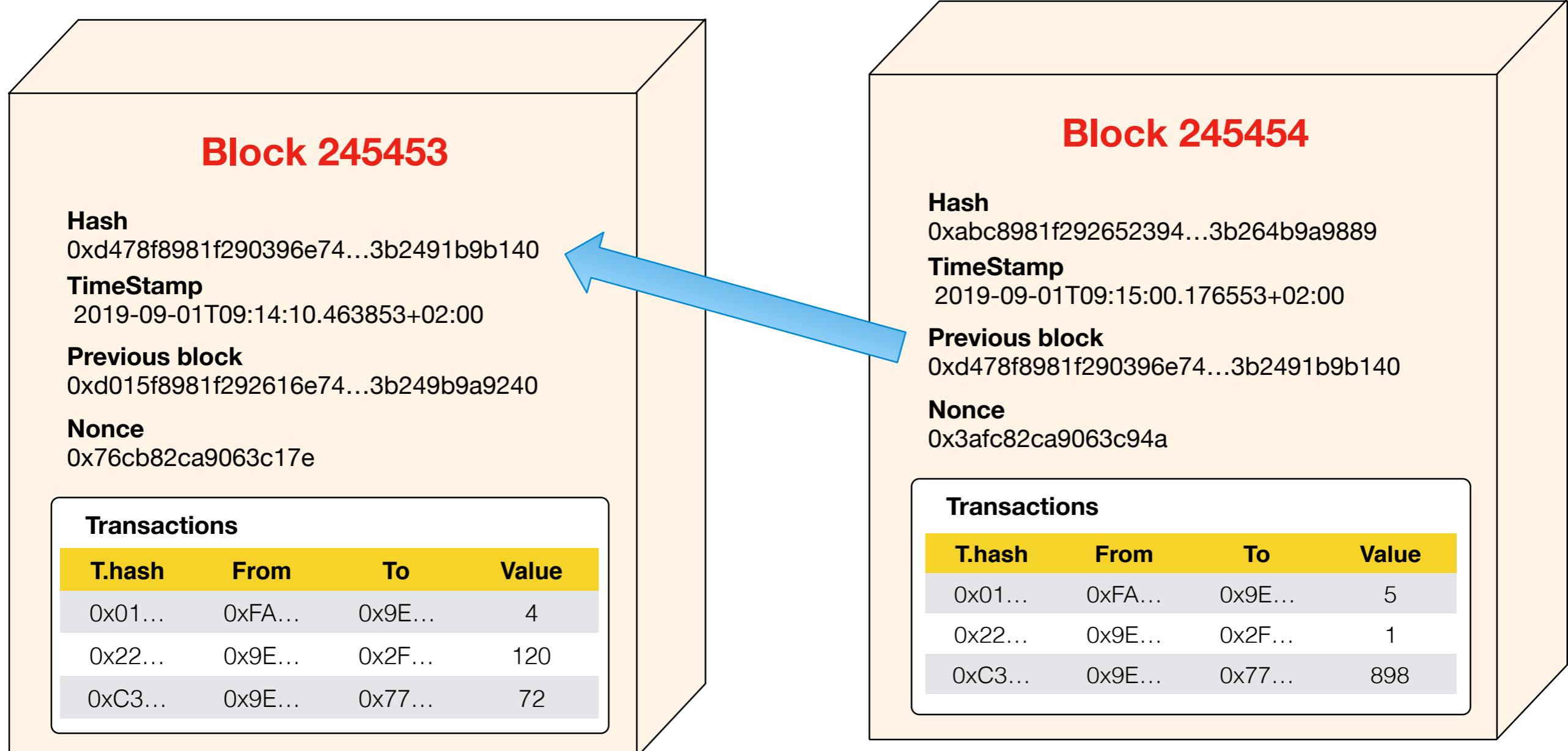




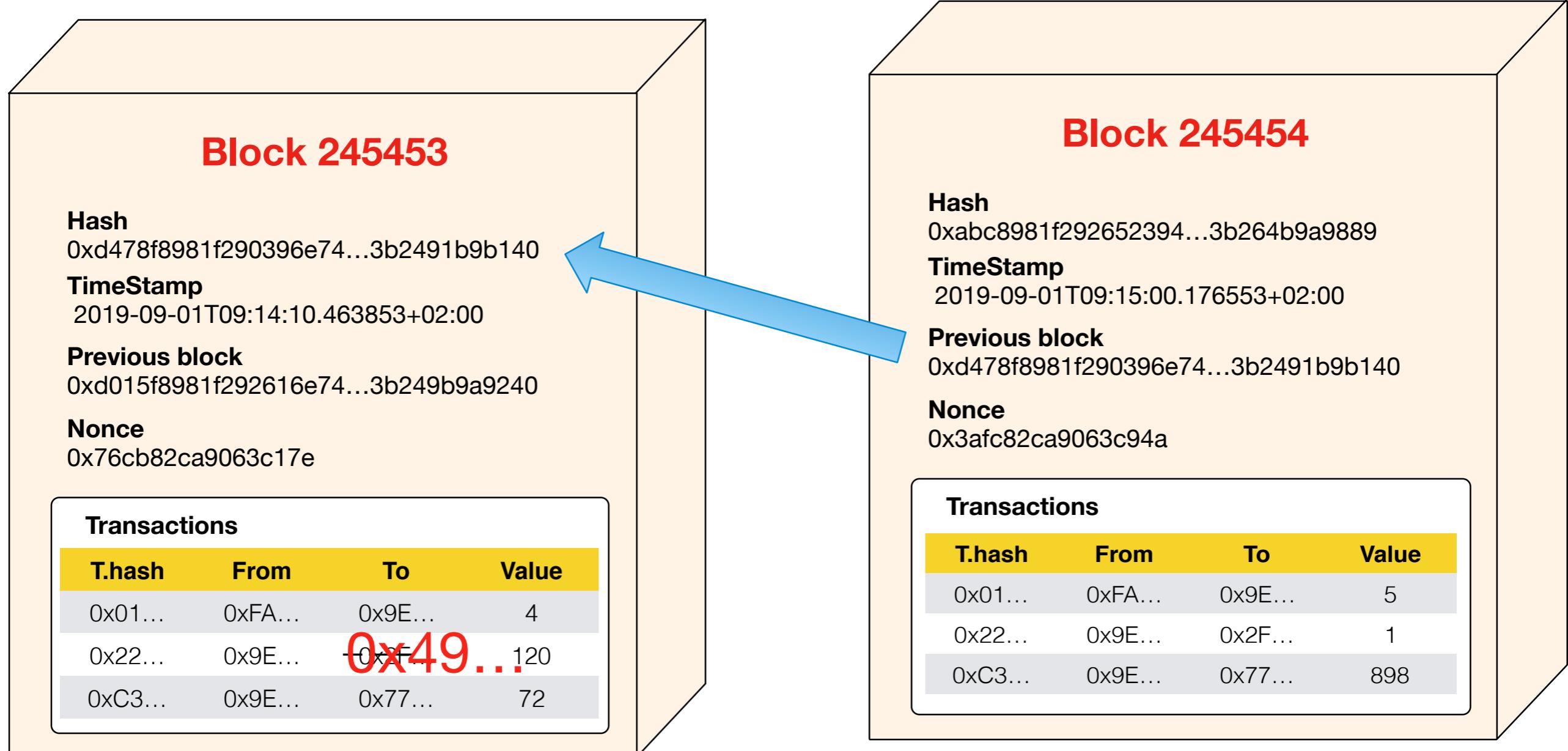
# What if... Someone Tampers the Blocks?

- For manually rollback an error
- For modify a transaction with malicious intentions

# Tampering Blocks



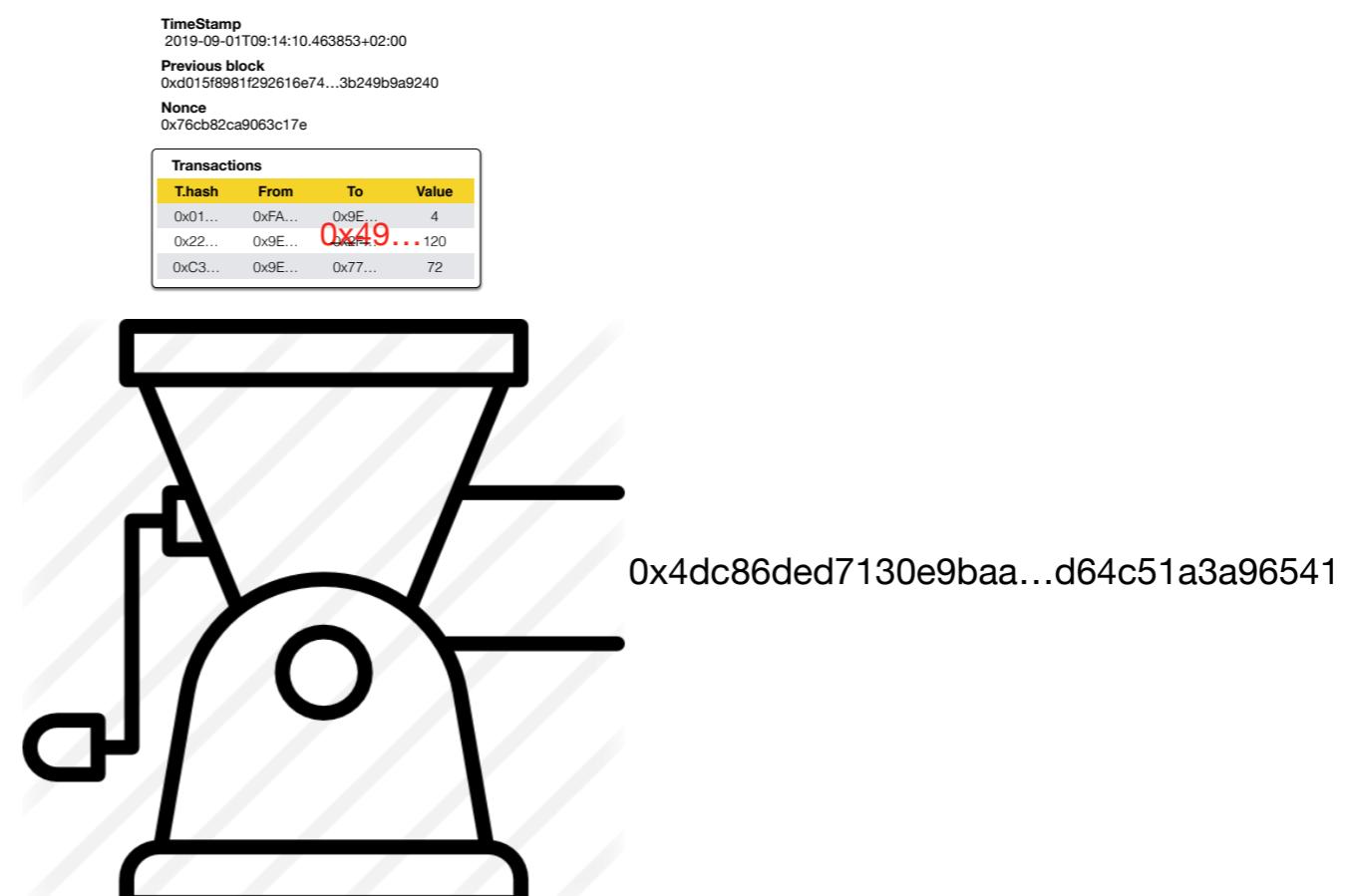
# Tampering Blocks



# Verification Process

**Block 245453**

<b>Hash</b>	0xd478f8981f290396e74...3b2491b9b140	X														
<b>TimeStamp</b>	2019-09-01T09:14:10.463853+02:00															
<b>Previous block</b>	0xd015f8981f292616e74...3b249b9a9240															
<b>Nonce</b>	0x76cb82ca9063c17e															
<b>Transactions</b>																
<table><thead><tr><th>T.hash</th><th>From</th><th>To</th><th>Value</th></tr></thead><tbody><tr><td>0x01...</td><td>0xFA...</td><td>0x9E...</td><td>4</td></tr><tr><td>0x22...</td><td>0x9E...</td><td>0x2F...</td><td>120</td></tr><tr><td>0xC3...</td><td>0x9E...</td><td>0x77...</td><td>72</td></tr></tbody></table>	T.hash	From	To	Value	0x01...	0xFA...	0x9E...	4	0x22...	0x9E...	0x2F...	120	0xC3...	0x9E...	0x77...	72
T.hash	From	To	Value													
0x01...	0xFA...	0x9E...	4													
0x22...	0x9E...	0x2F...	120													
0xC3...	0x9E...	0x77...	72													



# Verification Process

Block 245453			
<b>Hash</b>	0x4dc86ded7130e9baa...d64c51a3a96541	✓	
<b>TimeStamp</b>	2019-09-01T09:14:10.463853+02:00		
<b>Previous block</b>	0xd015f8981f292616e74...3b249b9a9240		
<b>Nonce</b>	0x76cb82ca9063c17e		
<b>Transactions</b>			
<b>T.hash</b>	<b>From</b>	<b>To</b>	<b>Value</b>
0x01...	0xFA...	0x9E...	4
0x22...	0x9E...	0x2F...	120
0xC3...	0x9E...	0x77...	72

TimeStamp  
2019-09-01T09:14:10.463853+02:00

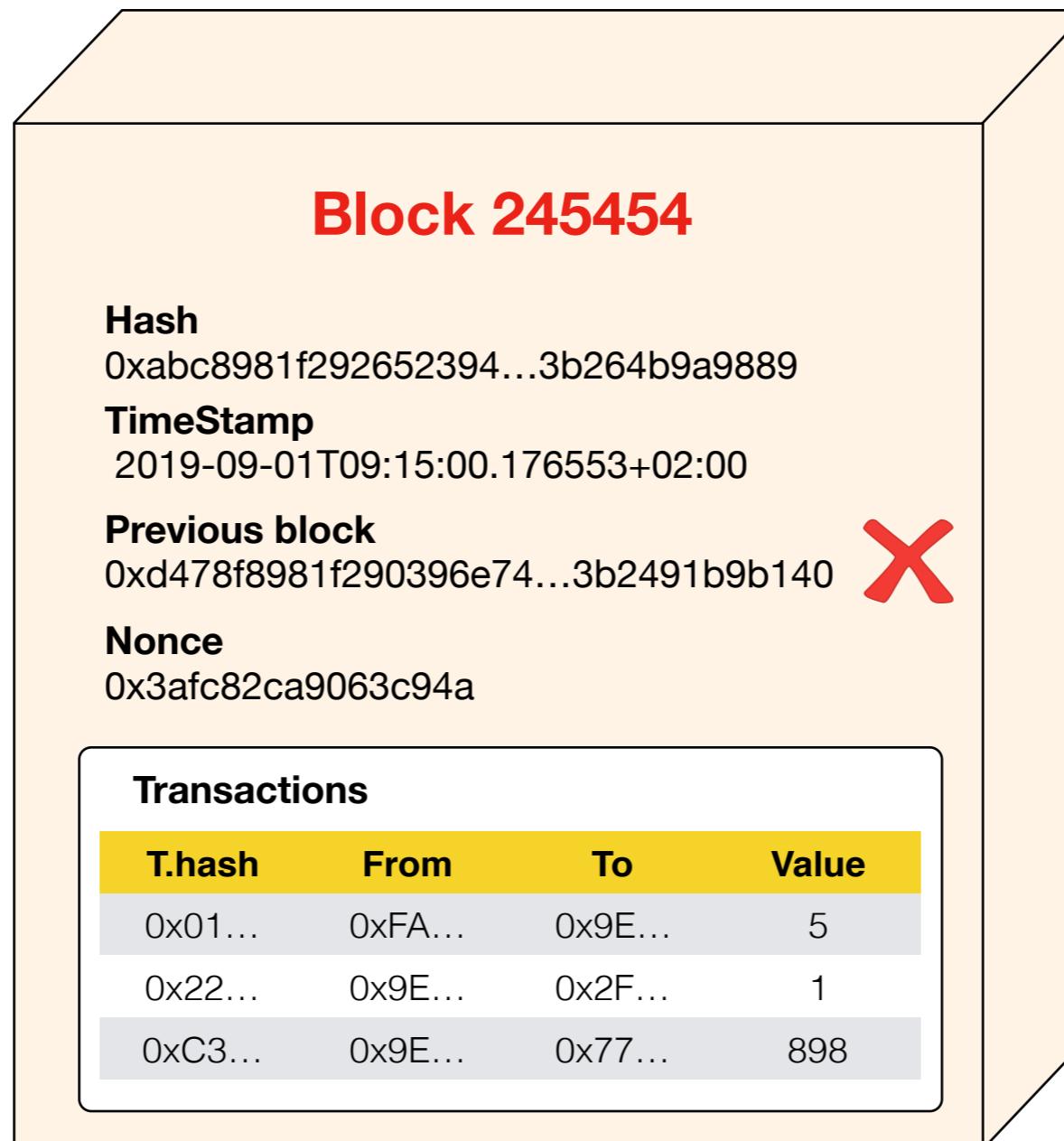
Previous block  
0xd015f8981f292616e74...3b249b9a9240  
Nonce  
0x76cb82ca9063c17e

Transactions			
T.hash	From	To	Value
0x01...	0xFA...	0x9E...	4
0x22...	0x9E...	0x2F...	120
0xC3...	0x9E...	0x77...	72



0x4dc86ded7130e9baa...d64c51a3a96541

# Verification Process



Cool but...  
what is it good for?

Ensure traceability

# Trustless Trust



The transactions/blocks are verified by every node (trustless).

This ensures the approved transactions/block are legit (promoting trust).

And allows entities to interact without a trusted 3rd party.

# Mining

# Mining

- To Mine is the process of creating a valid block and adding it to the blockchain
- A Miner is someone who mines the blockchain.
- A miner that creates a valid block, earns the right of acquiring a parametrisable amount of cryptocurrency



# **What is a valid block?**

# **What is a valid block?**

One that is accepted by the  
consensus algorithm of the network

# Consensus algorithms

- Verify / accept modifications
- Ensure that the miner has something to lose

# Consensus algorithms

## Verify / accept modifications

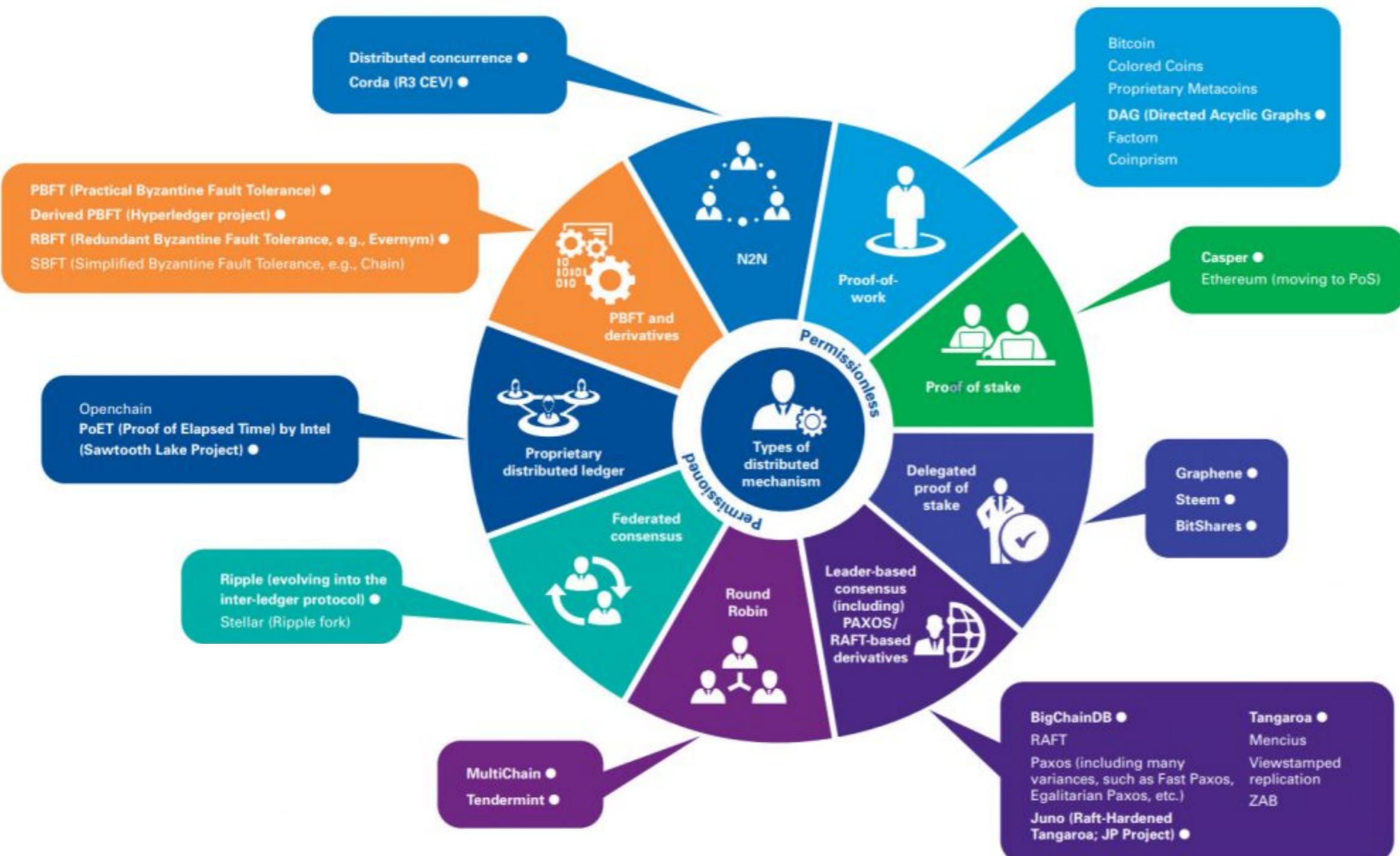
- Verify block hashes
- Verify non-double spent
- Verify transactions consistency

# Consensus algorithms

Ensure that the miner has something to lose

- Ensure that the miners do not complot for modifying history
- Ensure that the miners do not have parasitic behaviours

# Consensus algorithms



# Proof of work



- Requires the resolution of a crypto-puzzle
- Ensures the economic engagement of the miners
- Super high energy consumption

# Proof of work: The puzzle



**Hash**

0xabc8981f292652394...3b264b9a9889

**TimeStamp**

2019-09-01T09:15:00.176553+02:00

**Previous block**

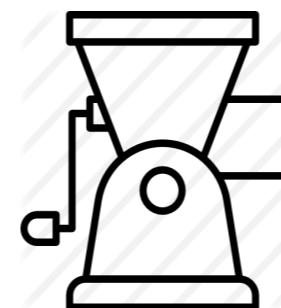
0xd478f8981f290396e74...3b2491b9b140

**Nonce**

0x3afc82ca9063c949

**Transactions**

T.hash	From	To	Value
0x01...	0xFA...	0x9E...	5
0x22...	0x9E...	0x2F...	1
0xC3...	0x9E...	0x77...	898



0x03bcaced7130e9baa...d64c51a3a96541



**Hash**

0xabc8981f292652394...3b264b9a9889

**TimeStamp**

2019-09-01T09:15:00.176553+02:00

**Previous block**

0xd478f8981f290396e74...3b2491b9b140

**Nonce**

0x3afc82ca9063c94a

**Transactions**

T.hash	From	To	Value
0x01...	0xFA...	0x9E...	5
0x22...	0x9E...	0x2F...	1
0xC3...	0x9E...	0x77...	898



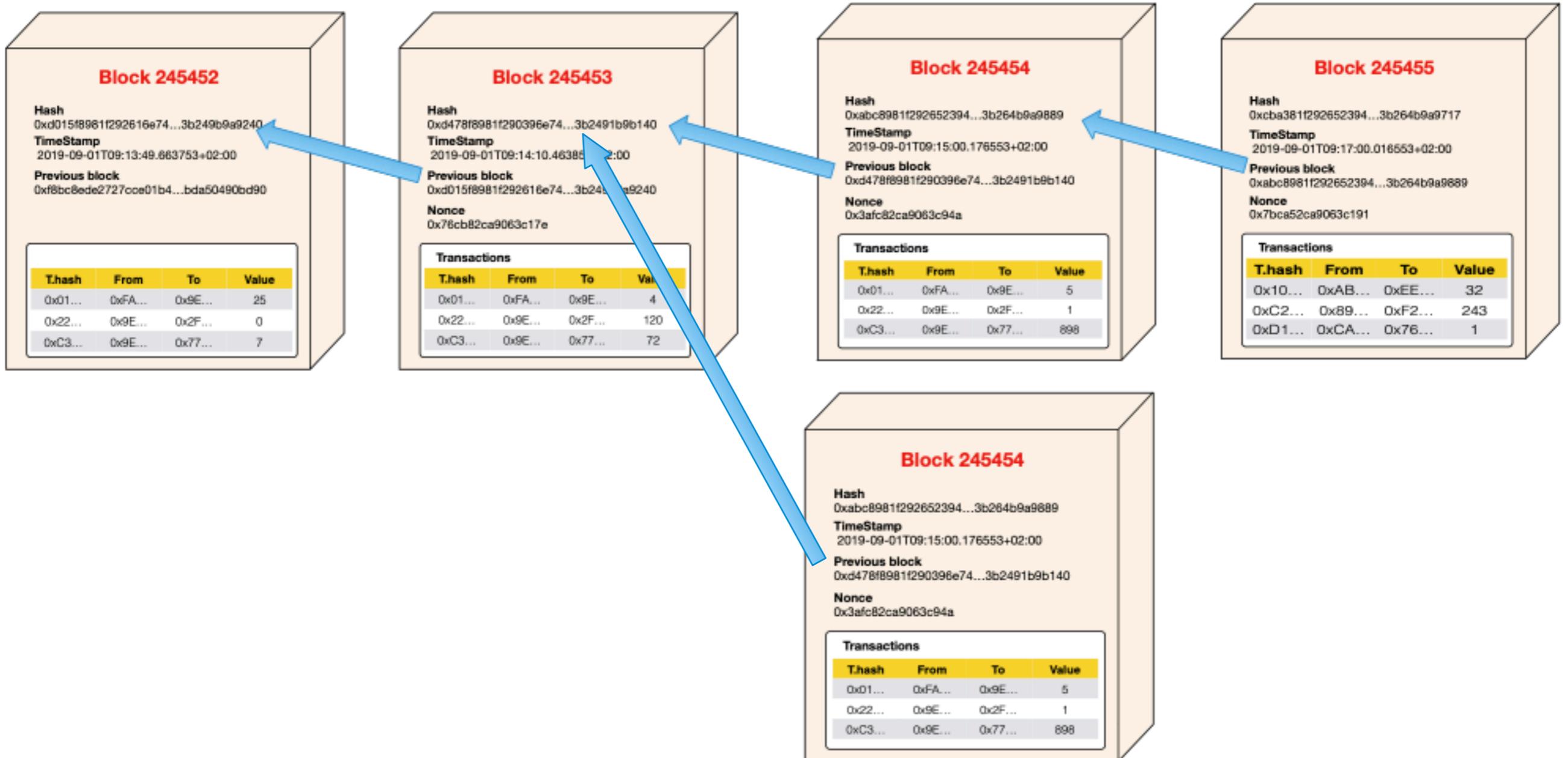
0x000000ed7130e9baa...d64c51a3a96541



# **What is a valid blockchain?**

The longest chain

# The longest chain



# The longest chain

- Ensures that tampering is near impossible
- Chooses the chain that most of the users is using for

# Brief Blockchain

# Brief

- P2P Open Network
- Transactional Append only database
- Administrated and surveilled by an engaged community
- Rules defined by a consensus algorithm

# Smart Contracts

# Smart Contracts

[Fritz Henglein, Smart Contracts..., 2017]

- “Smart Contracts are neither Smart nor Contracts”
  - Smart Contracts are self-executing programs coded in complex Turing-complete language.
  - Rules & Actions intermixed (not a contract)
  - Low-level programs hard to analyze (not smart)





# Smart Contracts

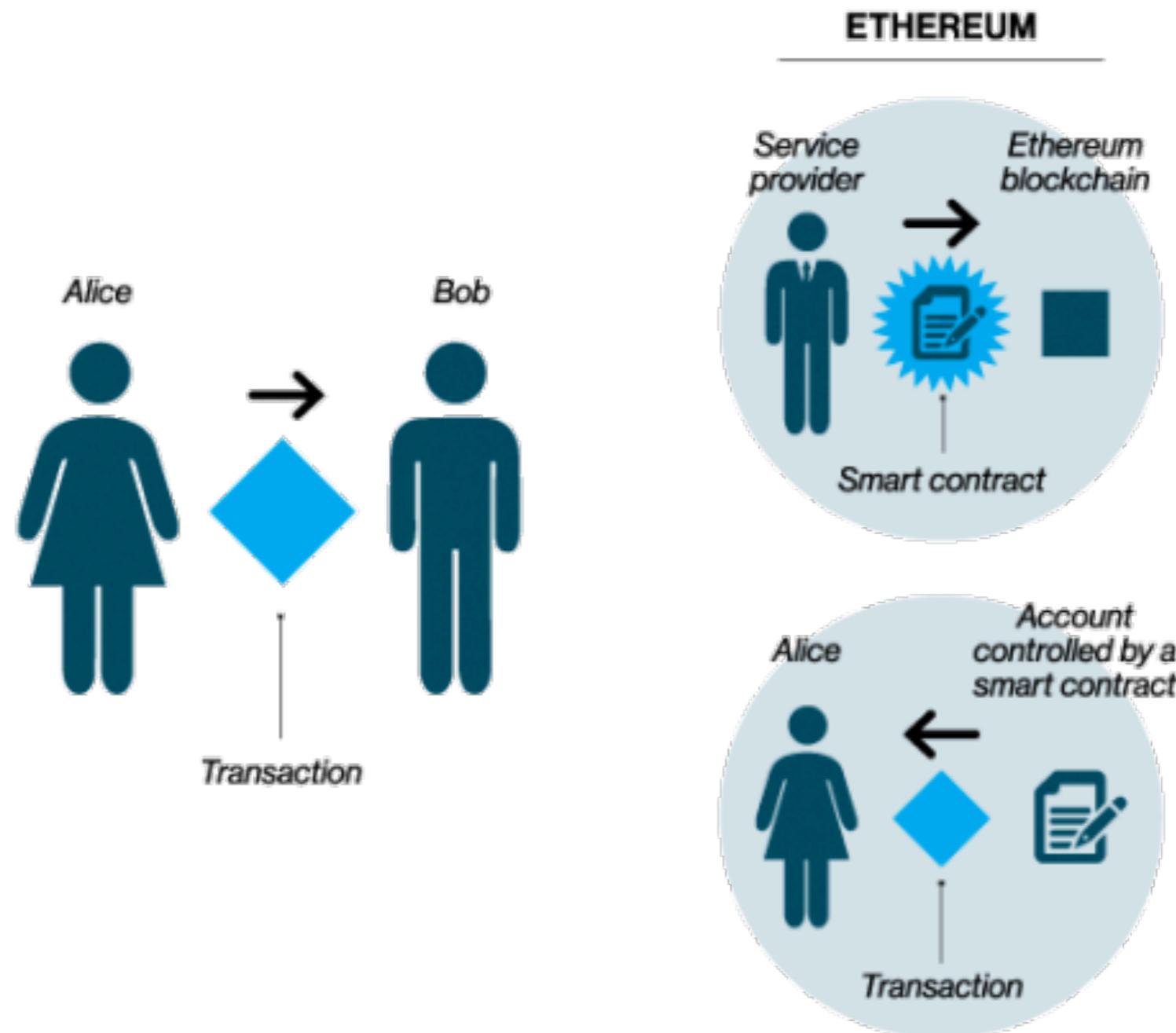
[Ethereum Doc, 2018]

- Contracts should not be seen as something that should be “fulfilled” or “complied with”
- They are more like “autonomous agents” that live inside the blockchain
  - Executing a specific piece of code when “poked” by a message or transaction
  - Have direct control over their cryptocurrency and attributes to store their permanent state.

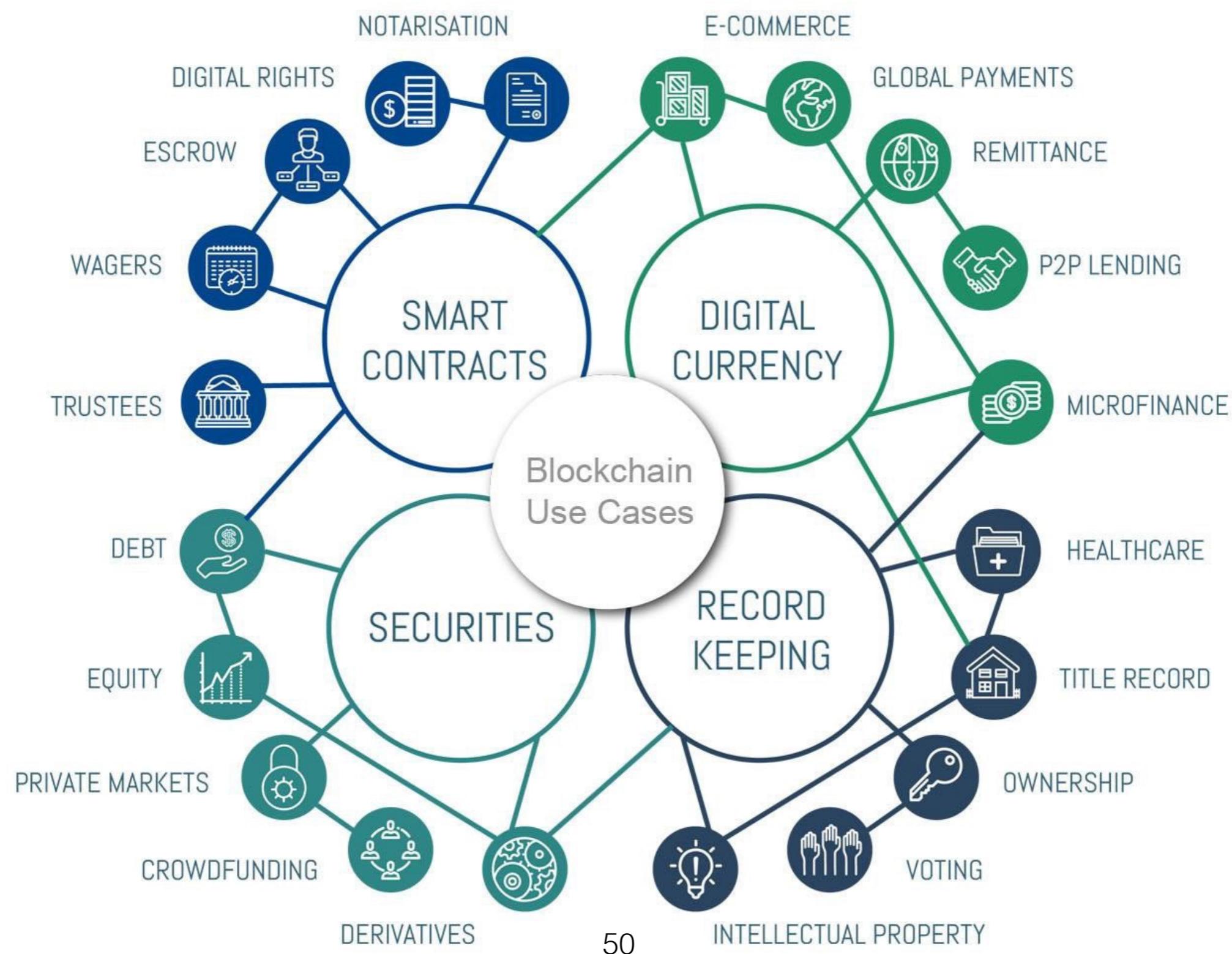
# Contract Ex: Pair Storage

```
1 pragma solidity^0.4.24;
2 /**
3  * @title Simple Pair Storage contract
4  * @author Henrique
5  */
6 contract SimplePairStorage{
7     int private data1;
8     uint data2;
9
10    function setData(int _data1, uint _data2) public {
11        data1 = _data1;
12        data2 = _data2;
13    }
14
15    function getData() public view returns(int,uint){
16        return (data1,data2);
17    }
18 } //end of contract
```

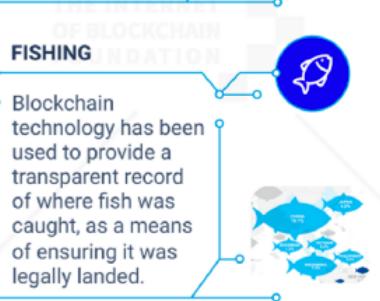
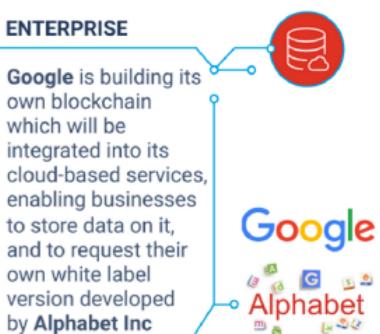
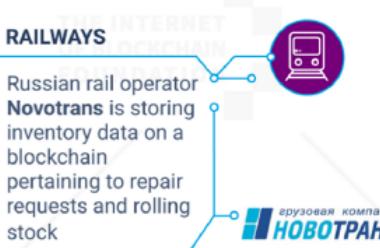
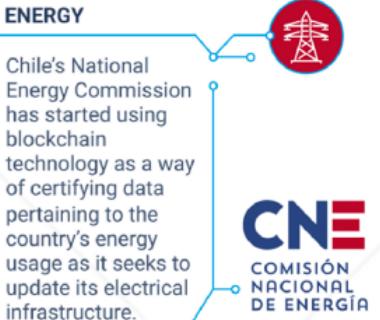
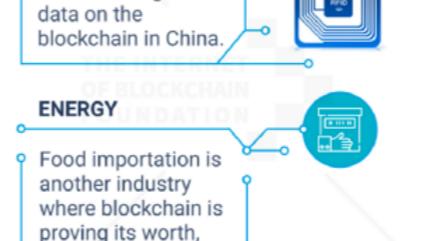
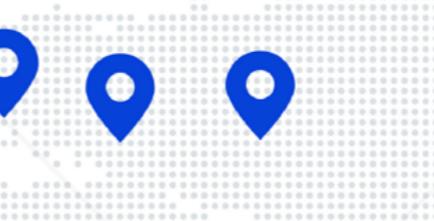
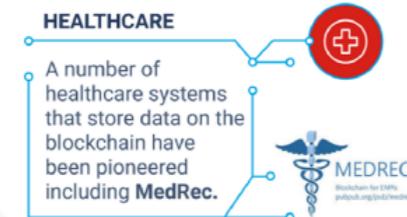
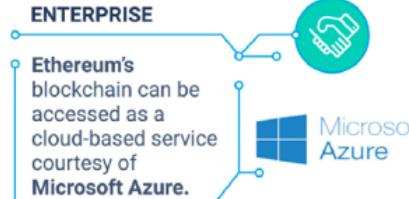
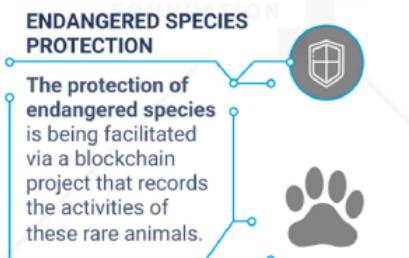
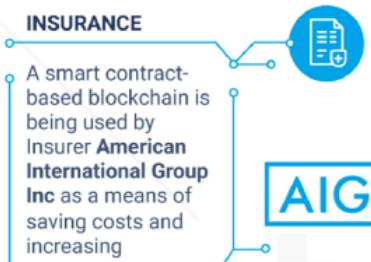
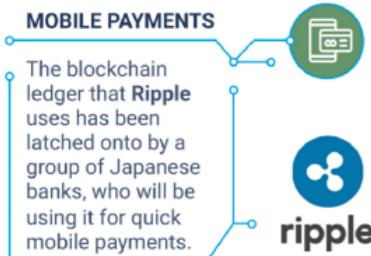
# Diagram Contract Interaction



# Blockchain Use Cases

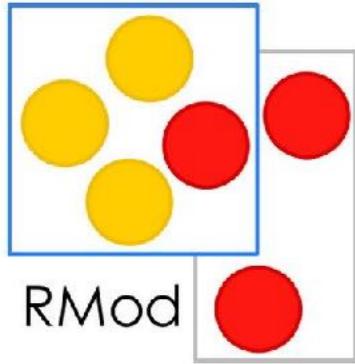


# 50+ BLOCKCHAIN REAL WORLD USES CASES





# Blockchain Research



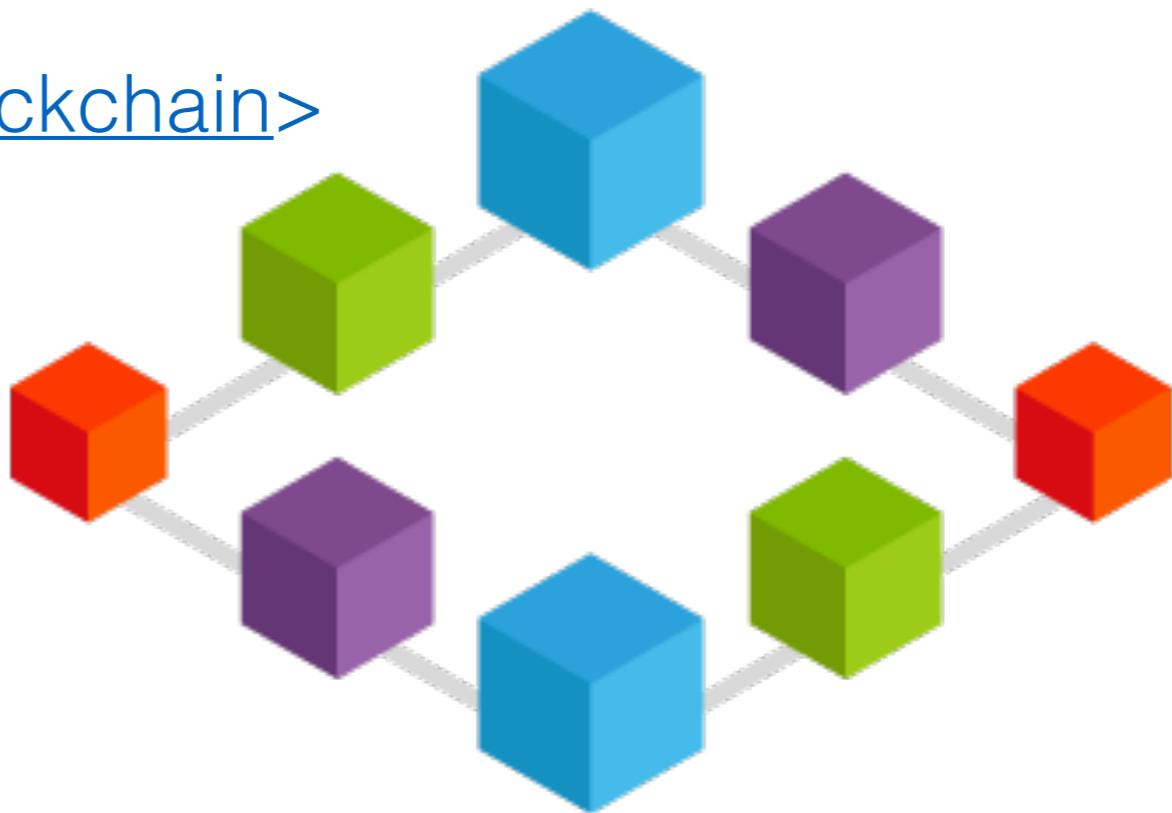
# Rmod

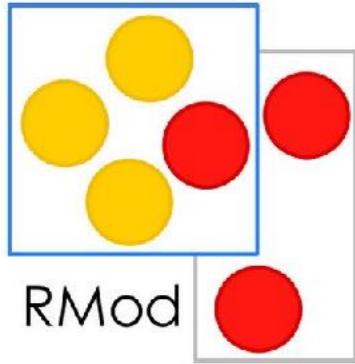
We welcome any research idea you may have for us.

We are also open to industrial and academic partnerships to advance blockchain research.

[<https://github.com/RMODINRIA-Blockchain>](https://github.com/RMODINRIA-Blockchain)

[<https://rmod.inria.fr/web>](https://rmod.inria.fr/web)





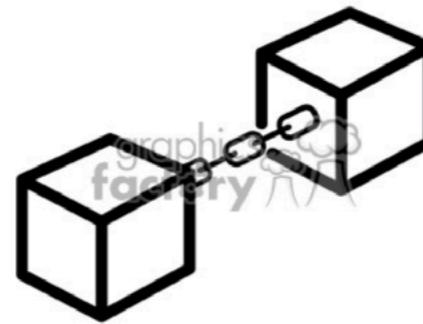
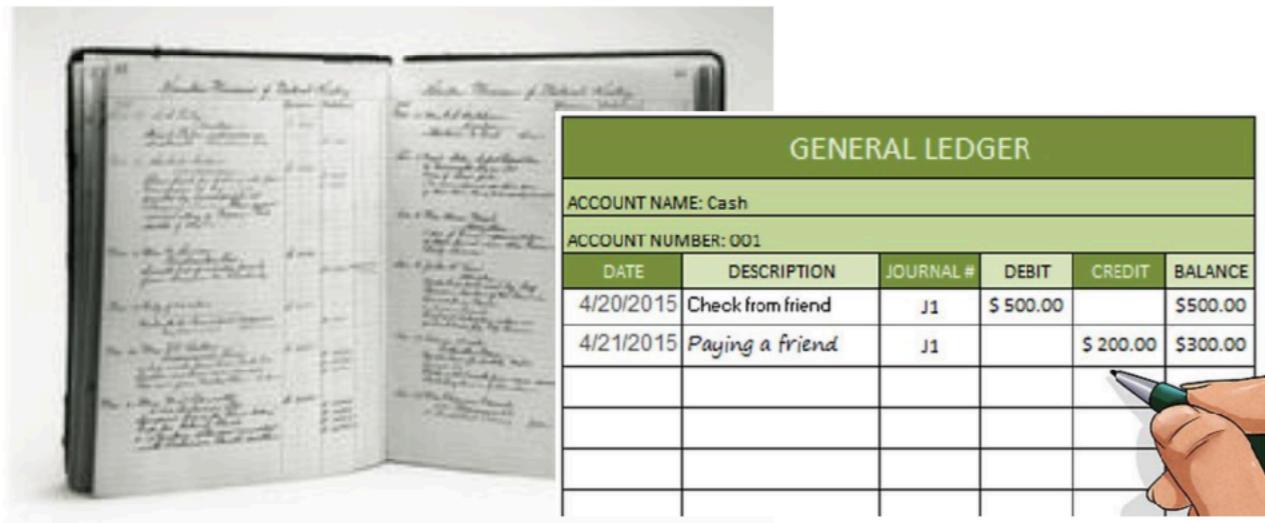
# Rmod

- SmartInspect: Contract internal state inspector
- EQL: query language for Ethereum blockchain
- SmartMetrics: Metric suite for contracts
- SolVis: Visualization tool for Solidity contracts
- BcModel: Modeling tools
- Business process Orchestration

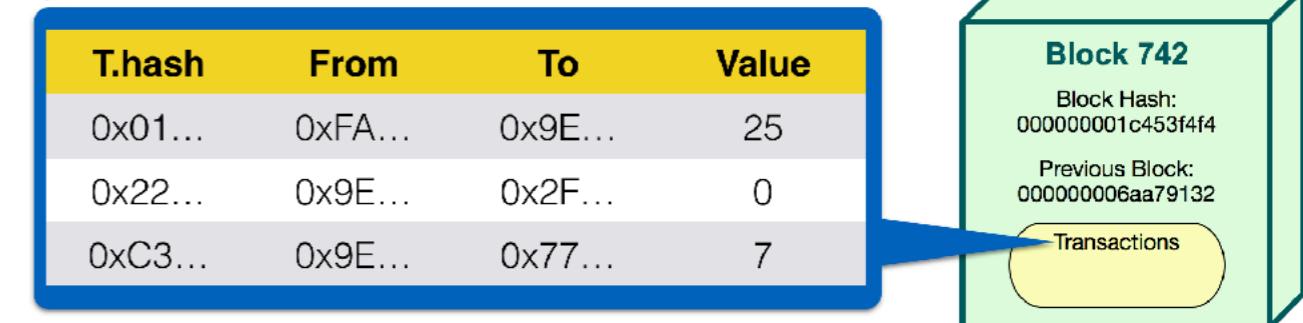
# What is Blockchain?

## From an User's Perspective

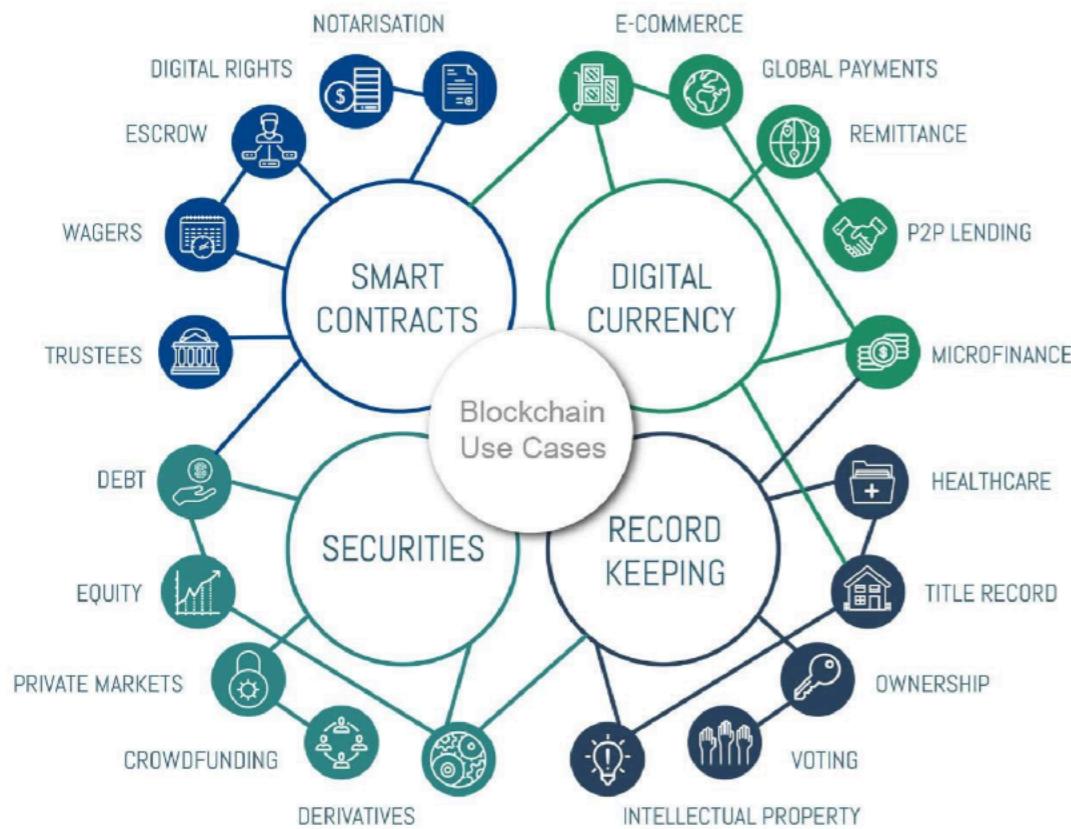
- An open distributed Ledger that promotes privacy



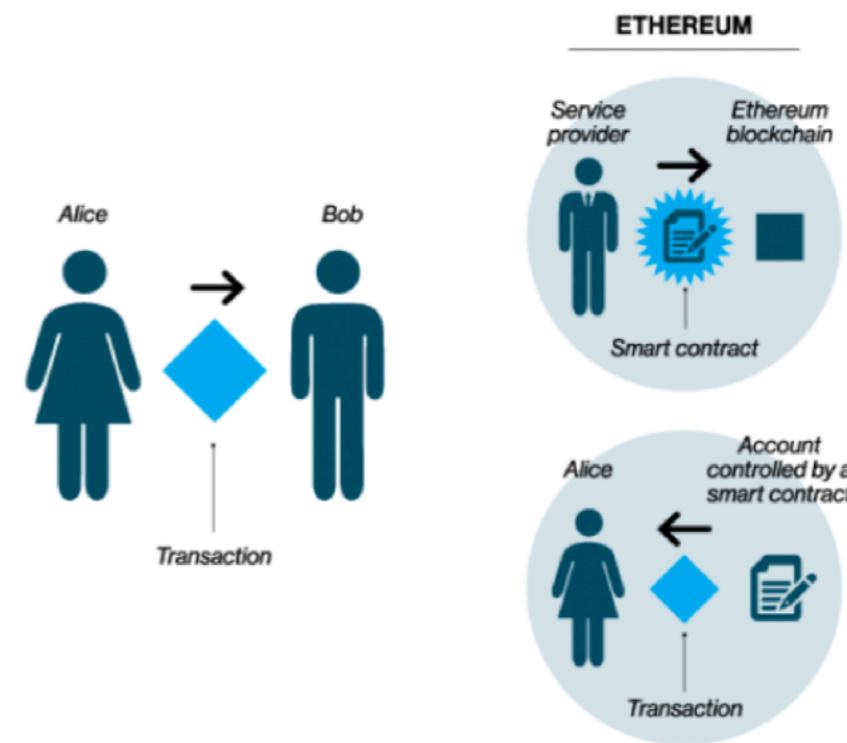
## Blocks



## Blockchain Use Cases



## Diagram Contract Interaction



## Thanks... questions?

Break :)