Blockchain

An implementation-agnostic definition

Prof. Marco Comuzzi
Department of Industrial Engineering
Ulsan National Institute of Science and Technology (UNIST)
mcomuzzi@unist.ac.kr

Plan for today

Implementation-agnostic blockchain model:

Blockchain as a Network

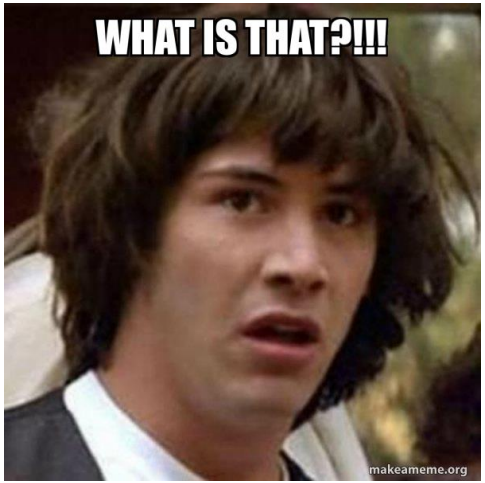
Blockchain as a Data Structure

Blockchain as a Network

BC4C: BlockChain for Chess

Marco Comuzzi mcomuzzi@unistac.kr

Implementation-agnostic??



A model of blockchain that does not depend on the specific implementation (Bitcoin, Ethereum, etc.)

Separate fundamental design concerns from implementation-specific details

Help to introduce specific systems later on in the course

Marco Comuzzi mcomuzzi@unistac.kr

Two-step definition

Blockchain

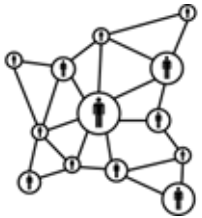
Smart contracts

Marco Comuzzi mcomuzzi@unistac.kr

OK, but … what is blockchain?

P2P Network

A set of nodes (peers) connected to each other on the Internet



Data structure

A database replicated on each node of the network

[distributed ledger]

Protocol

A set of rules for nodes to agree on the content of the database

[consensus mechanism]

Marco Comuzzi mcomuzzi@unistac.kr

1.
Blockchain as a Network

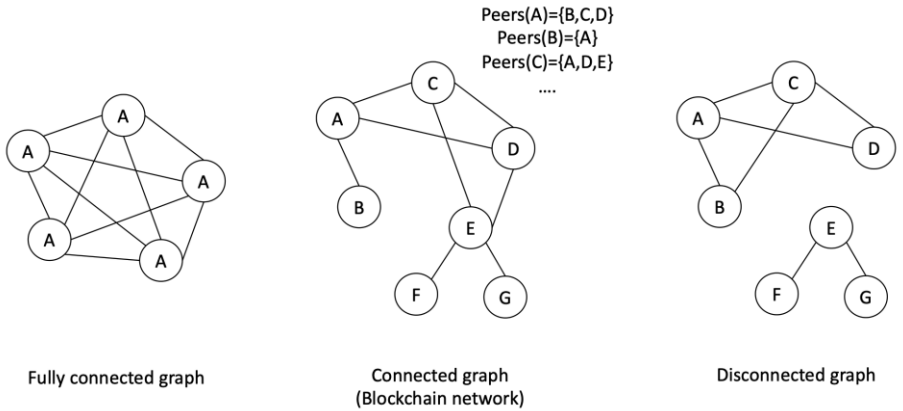
Marco Comuzzi mcomuzzi@unistac.kr

Blockchain as a P2P Network

Blockchain is a connected graph of computational nodes connected to the Internet

Each node is connected to a set of "peers"

Nodes can disconnect and reconnect to the network at their will



Marco Comuzzi mcomuzzi@unistac.kr

Nodes of a blockchain network: terminology

User

A person or organization using a blockchain. Often anonymous (e.g., in Bitcoin and other cryptocurrencies)

Client

The software application through which a user interacts with a blockchain (Web site, mobile app, cryptocurrency ATM, …)

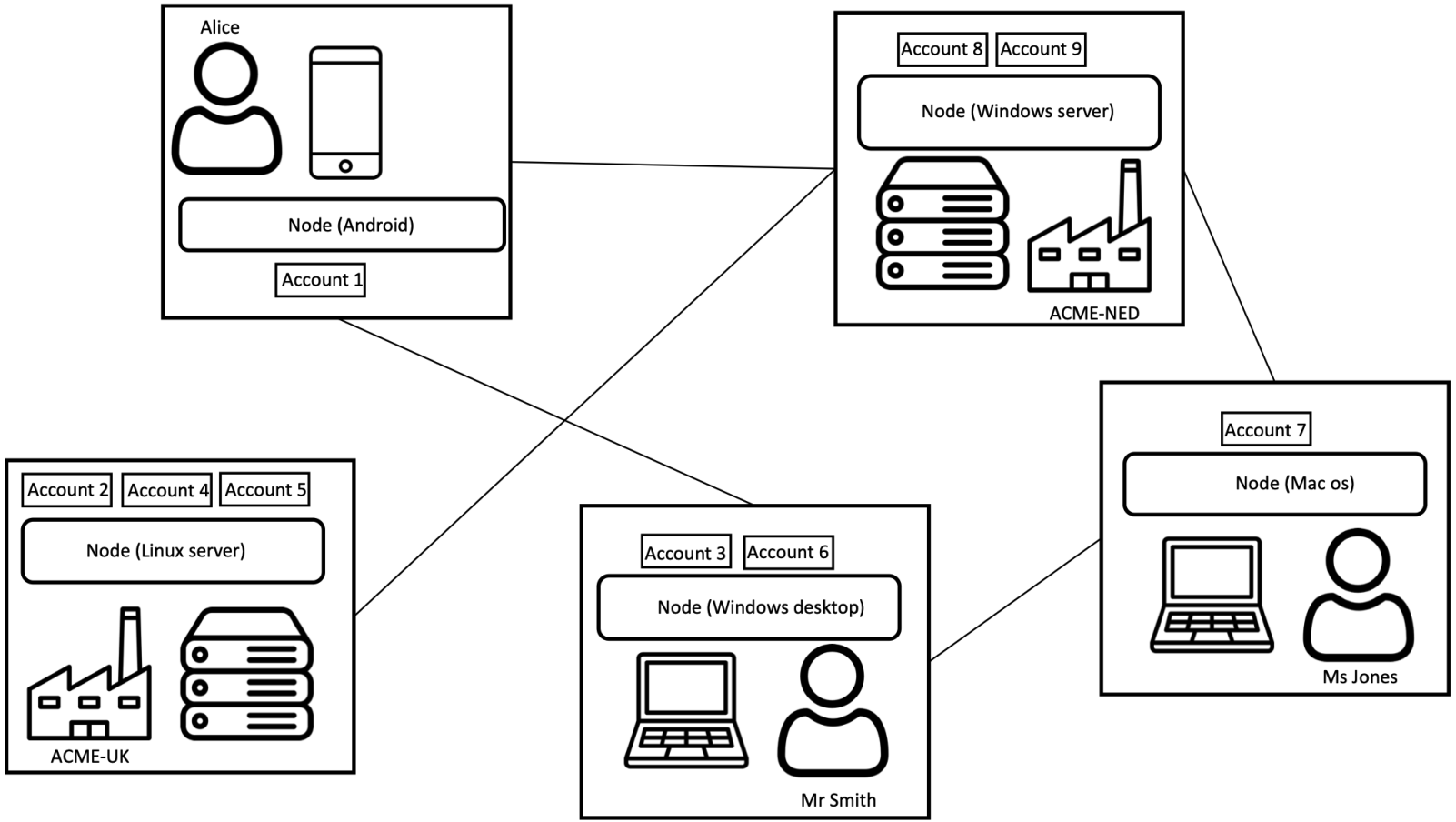
Node

Inside a client, a node is the software component implementing the application logic to interact with a blockchain

Account

A unique identifier of a user interacting with a blockchain network. Usually associated with the public/private keys to authenticate the messages sent.

Marco Comuzzi mcomuzzi@unistac.kr


Alice
Node (Android)
Account 1
Account 8
Account 9
Node (Windows server)
ACME-NED
Account 2
Account 4
Account 5
Node (Linux server)
ACME-UK
Account 3
Account 6
Node (Windows desktop)
Mr Smith
Account 7
Node (Mac os)
Ms Jones

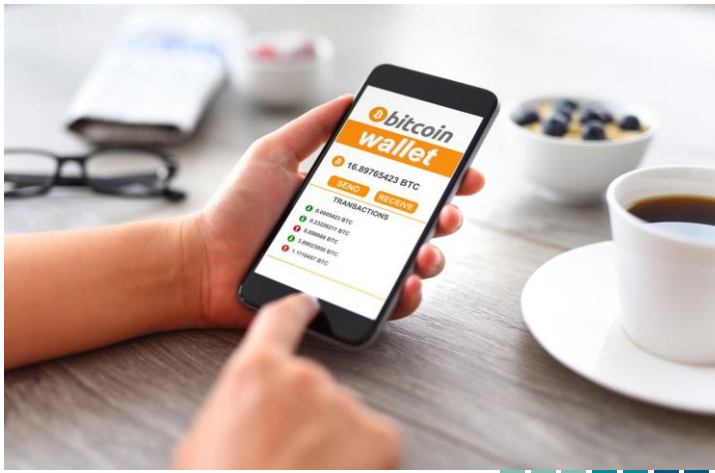
Then…what's a "wallet"?

In a cryptocurrency system, a wallet is the sw application for users to access and manage their digital assets (e.g., BTC, ETH or other Ethereum tokens).

May use multiple accounts under the hood

Realize for the user the abstraction of a physical wallet

Wallets mix-up the concepts of client, node, accounts.



Marco Comuzzi mcomuzzi@unistac.kr

Public vs. Private blockchain networks

Public blockchain

Anybody can be a user.

Users are normally anonymous.

Issues with scalability/performance.

Examples: Bitcoin or any cryptocurrency

Private blockchain

Access to the network is vetted.

Identify of nodes is known.

Typical of many common business scenarios.

Examples: later in the course

Marco Comuzzi mcomuzzi@unistac.kr

2.
Blockchain as a Data Structure

Data in a blockchain

Blockchain nodes exchange messages, usually called “transactions”, which update the “state” of the blockchain

A transaction is issued by one node (originator) and may or not be addressed to one or many other nodes

- Cryptocurrencies: from sender to receiver
- Supply chain: from originator to every node

Transaction are validated and broadcast by a node to its peers, reaching in this way all the nodes (gossiping)

Every node stores the transactions.

Marco Comuzzi mcomuzzi@unistac.kr

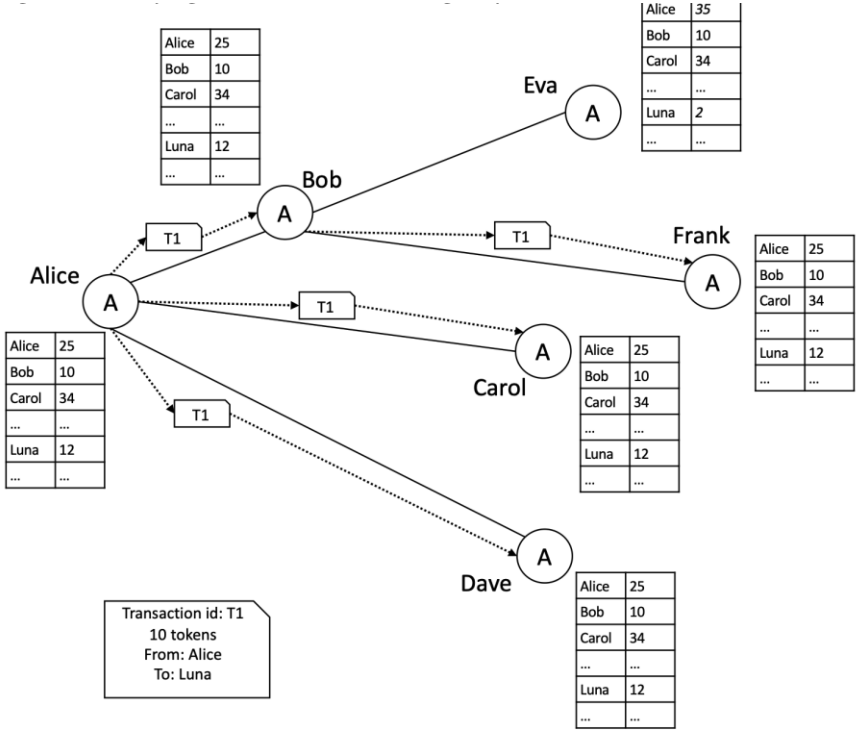
Order of "gossiped" transactions

Network delays, disconnections, node connection patterns make the gossiping unpredictable

A node N is not guaranteed that all other nodes will receive the transactions in the same order in which they are issued by N

This has major implications on the design and behaviour of a blockchain

Gossiping of transactions example [cryptocurrency]



STATE = balance of all nodes

Marco Comuzzi mcomuzzi@unistac.kr

Transactions as state updates

Transactions are not direct messages among nodes of a blockchain network

Transactions are used by nodes to signal a change of the blockchain state, which all nodes store in their local copy of the ledger

Authentication of originators

Every transaction must be digitally signed by its originator

Originators sign transactions using their private key

Any other node can verify the signature using an originator's public key

Public key is usually included in a transaction by the originator

Validation of transactions

Nodes receiving a transaction T check the following:

That T is “well-formed” (syntax and content)

That T proposed a state update coherent with the current state of the blockchain

That T is digitally signed by the originator

Marco Comuzzi mcomuzzi@unistac.kr

Transactions and Blocks (towards immutability)

Every node stores every transaction in its local ledger

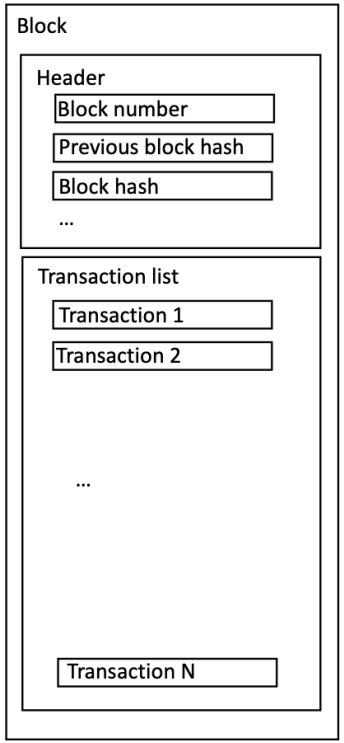
The transactions can then be used by nodes to reconstruct the current state of the ledger (sometime the state itself is also stored in the ledger)

Transactions are usually not stored individually, but in “blocks”

Blocks are not strictly necessary, a blockchain can also stored each transaction individually

Marco Comuzzi mcomuzzi@unistac.kr

Blocks

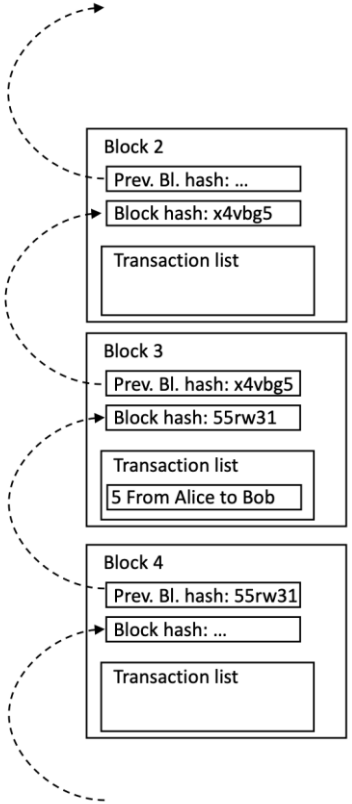


Blocks contain a list of (recently submitted) transactions and a header

Header contains “metadata” describing the block

Marco Comuzzi mcomuzzi@unistac.kr

A Chain of Blocks (or Blockchain?)



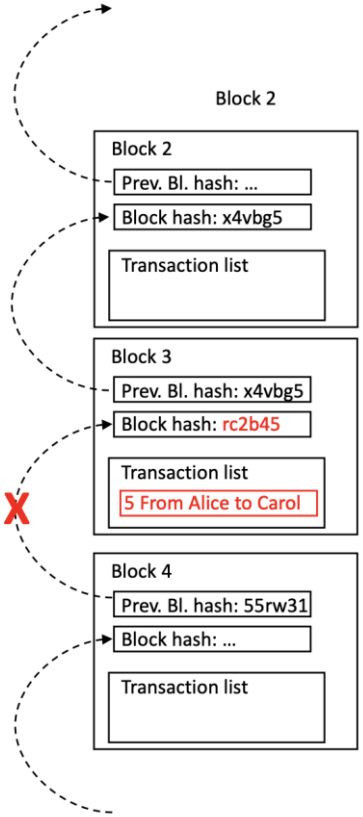
Blocks are numbered progressively

Block 0 is usually called the "Genesis" block

Block are linked by a chain of cryptographic hashes: each block contains its hash and the hash of its predecessor in the header

Marco Comuzzi mcomuzzi@unistac.kr

Immutability of a chain of blocks



Think an attacker wants to push a new transaction or modifying an existing one in block 3

Changing even one bit of Block 3 will change its hash and "break" the chain

The attacker will have to modify the content of the next blocks to match the new hash of block 3: this is computationally impossible because of the properties of cryptographic hashing

Marco Comuzzi mcomuzzi@unistac.kr

Demo

https://andersbrownworth.com/blockchain/hash

3. Blockchain as a Protocol

Why a protocol?

A blockchain protocol is usually called a “consensus mechanism”

It is devised so that nodes can agree on the content of the ledger at anytime

Content of the ledger: content and order of the transactions (or blocks) submitted until a given point in time

Marco Comuzzi mcomuzzi@unistac.kr

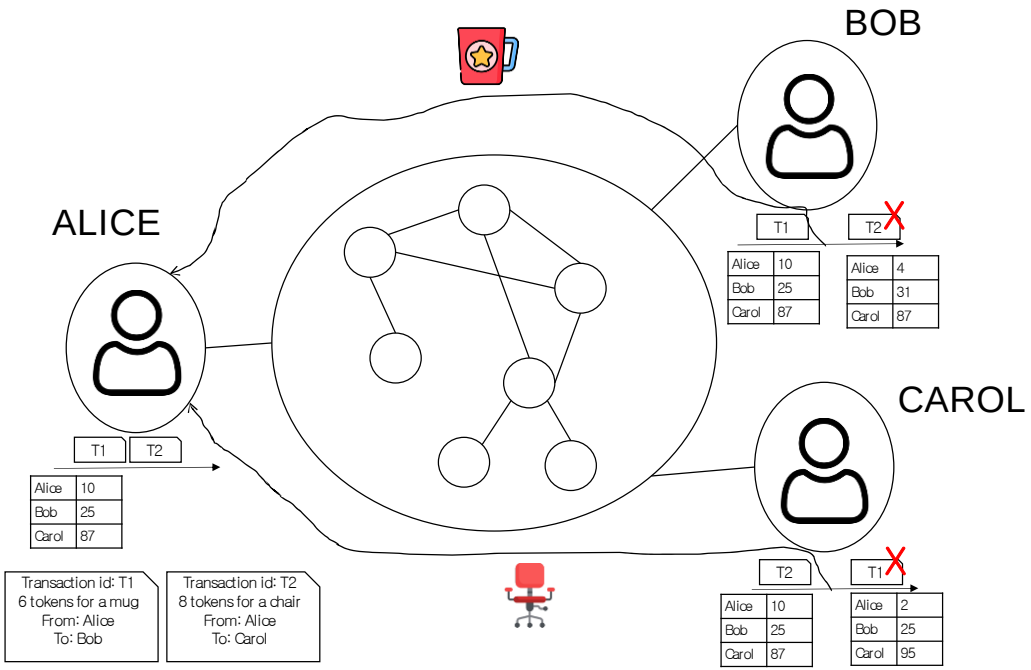
“Double spending”

Consensus protocols in blockchain solve the following fundamental problem:

“How is it possible to establish the order of transactions, particularly when issued in a short time span?”

In cryptocurrencies, this is the double spending problem: how can we prevent nodes to spend their coins twice?

Double spending in a nutshell



Alice wants to buy a mug (cost: 6 tokens) from Bob and a chair (8 tokens) from Carol

Alice has only 10 tokens, but she can issue T1 and T2 very close in time…

… if Bob receives T1 before T2 (and Carol T2 before T1), then Alice may "double spend" her tokens

Marco Comuzzi mcomuzzi@unistac.kr

4.
Using the Blockchain Definition: BC4C

Marco Comuzzi mcomuzzi@unistac.kr

Alice and Bob play chess on the Internet

They want to be able to disconnect and resume a game at any time

How can we design a system to support them?

How can such a system prevent that, while one player is offline, the other does not cheat?

Marco Comuzzi mcomuzzi@unistac.kr

BC4C: Blockchain Network

A public blockchain network

Anybody can potentially be a player, so anybody can run a node downloading the BC4C client software

Marco Comuzzi mcomuzzi@unistac.kr

BC4C: Blockchain Protocol

Rule ID	Rule Specification	Notes
1	A game involves 2 players. A new game is proposed by one player and must be accepted by the opposite player to start. When a game is accepted, an id is generated for it according to standard rules, e.g. “AliceBob3” for the 3rd game involving Alice and Bob	
2	A game starts with the standard configuration of a chess board (number/types of pieces and positions)	
3	The players of a game take turns in making moves. The player who proposed a game moves first. A move is proposed by one player and must be accepted by the opposite player. A unique id for each move can be generated following simple rules	Move validation rules can be also embedded in the transaction validation rules
4	A game terminates when a checkmate move is proposed by one player and this is approved by the counterpart.	Game termination

Marco Comuzzi mcomuzzi@unistac.kr

BC4C: Blockchain Transactions

Type	Specification	Notes
1	**`CreateNewGame [gameId, counterPlayer]`**: this type of transaction creates a new game between the originator of the transaction and another player, identified by the parameter counterPlayer.	
2	**`ConfirmGameCreation [gameId]`**: this type of transaction confirms the creation of a game. It is issued by the counter player.	
3	**`Move[gameId, moveId, piece, new position, isCheckMate]`**: This type of transaction specifies a move in a game. A move involves moving a piece to a new position on the board. A Boolean flag isCheckMate identifies whether the move leads to checkmate.	Move validation rules can be also embedded in the transaction validation rules
4	**`ConfirmMove [gameId, moveId]`**: This type of transaction is issued by a player to confirm the validity of a move issued by a counter player	Game termination

Marco Comuzzi mcomuzzi@unistac.kr

BC4C: Blockchain Ledger

Content of the ledger (transactions)

Transaction id	Transaction	Originator	Timestamp
0	Genesis of BC4C	BC4C	22-04-19 12:00:17
1	CreateNewGame[AliceCarol1, Carol]	Alice	22-04-20 09:00:03
2	CreateNewGame[BobCarol1, Carol]	Bob	22-04-20 09:00:45
3	ConfirmGameCreation[BobCarol1]	Carol	22-04-20 09:01:23
4	ConfirmGameCreation[AliceCarol1]	Carol	22-04-20 09:01:28
5	Move[BobCarol1, 1, pawn_h2, h3, false]	Bob	22-04-20 09:02:34
6	CreateNewGame[DaveAlice1, Alice]	Dave	22-04-20 09:02:48
7	Move[AliceCarol1, 1, pawn_d2, d4, false]	Alice	22-04-20 09:03:01
8	ConfirmMove[AliceCarol1, 1]	Carol	22-04-20 09:03:55
9	Move[AliceCarol, 2, pawn_a2, a3, false]	Carol	22-04-20 09:04:26

State of the System (games currently playing)

AliceCarol1

BobCarol1

A total of 9 transactions has been issued from the inception of the system.

There are currently two games being played: `AliceCarol1` and `BobCarol1`. The game `DaveAlice1` has been proposed by Dave, but not confirmed by Alice, yet.

There are no confirmed moves played in the game `BobCarol1`, whereas one move has been confirmed in the game `AliceCarol1` (see transactions 4 and 8). In the same game, Carol has proposed her own first move (transaction 9), but this has not yet been confirmed by Alice.

What's in the ledger

A list of transactions

The ledger stores the list of transactions issued from the inception of the system

The state of the ledger can be reconstructed by nodes by replaying all the transactions

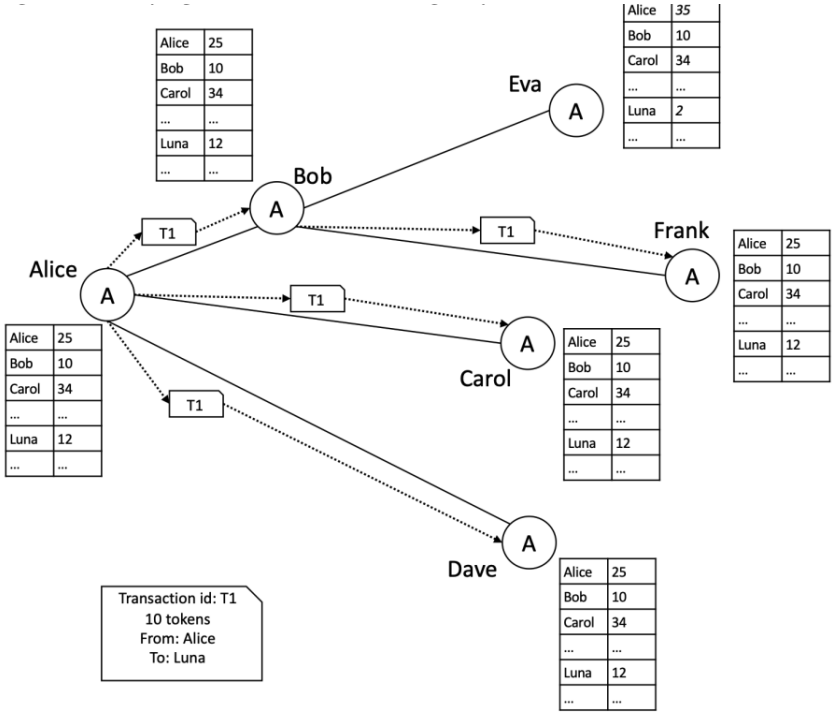
The state of the system

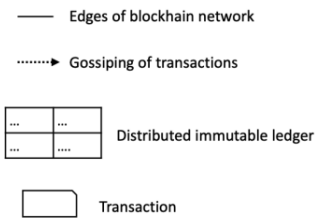
The ledger contains the value of the variables describing the state of the system, updated by every new transactions

Easier to access the blockchain state, but we lose information about individual transactions

Marco Comuzzi mcomuzzi@unistac.kr

Ledger as state of the system: cryptocurrency





The state of the system is the balance of all nodes

Transactions are transfers of cryptocurrency units between nodes

Marco Comuzzi mcomuzzi@unistac.kr

Ledger as transaction list: BC4C

Content of the ledger (transactions)

Transaction id	Transaction	Originator	Timestamp
0	Genesis of BC4C	BC4C	22-04-19 12:00:17
1	CreateNewGame[AliceCarol1, Carol]	Alice	22-04-20 09:00:03
2	CreateNewGame[BobCarol1, Carol]	Bob	22-04-20 09:00:45
3	ConfirmGameCreation[BobCarol1]	Carol	22-04-20 09:01:23
4	ConfirmGameCreation[AliceCarol1]	Carol	22-04-20 09:01:28
5	Move[BobCarol1, 1, pawn_h2, h3, false]	Bob	22-04-20 09:02:34
6	CreateNewGame[DaveAlice1, Alice]	Dave	22-04-20 09:02:48
7	Move[AliceCarol1, 1, pawn_d2, d4, false]	Alice	22-04-20 09:03:01
8	ConfirmMove[AliceCarol1, 1]	Carol	22-04-20 09:03:55
9	Move[AliceCarol, 2, pawn_a2, a3, false]	Carol	22-04-20 09:04:26

State of the System (games currently playing)

AliceCarol1

BobCarol1

The state of the system is the current state of games

Transactions are the moves of the games

THANKS!

https://sites.google.com/site/marcocomuzziphd

http://iel.unist.ac.kr/

You can find me at:

@dr_bsad

mcomuzzi@unist.ac.kr