ENGINE

Co-funded by the
Erasmus+ Programme
of the European Union

Blockchain and Cryptocurrency Technologies

Tackling the Magic

NICOLA DRAGONI
Professor in Secure Pervasive Computing, PhD
Head of DTU Center for Digital Security (DIGISEC)
Deputy Head of DTU Compute's PhD School
DTU Compute
Technical University of Denmark (DTU)
ndra@dtu.dk

About This Talk

Co-funded by the
Erasmus+ Programme
of the European Union

"Any sufficiently advanced technology is indistinguishable from magic"

Arthur C. Clarke, *Hazards of Prophecy: The Failure of Imagination*,

Profiles of the Future: An Inquiry into the Limits of the Possible, 1962

About This Talk

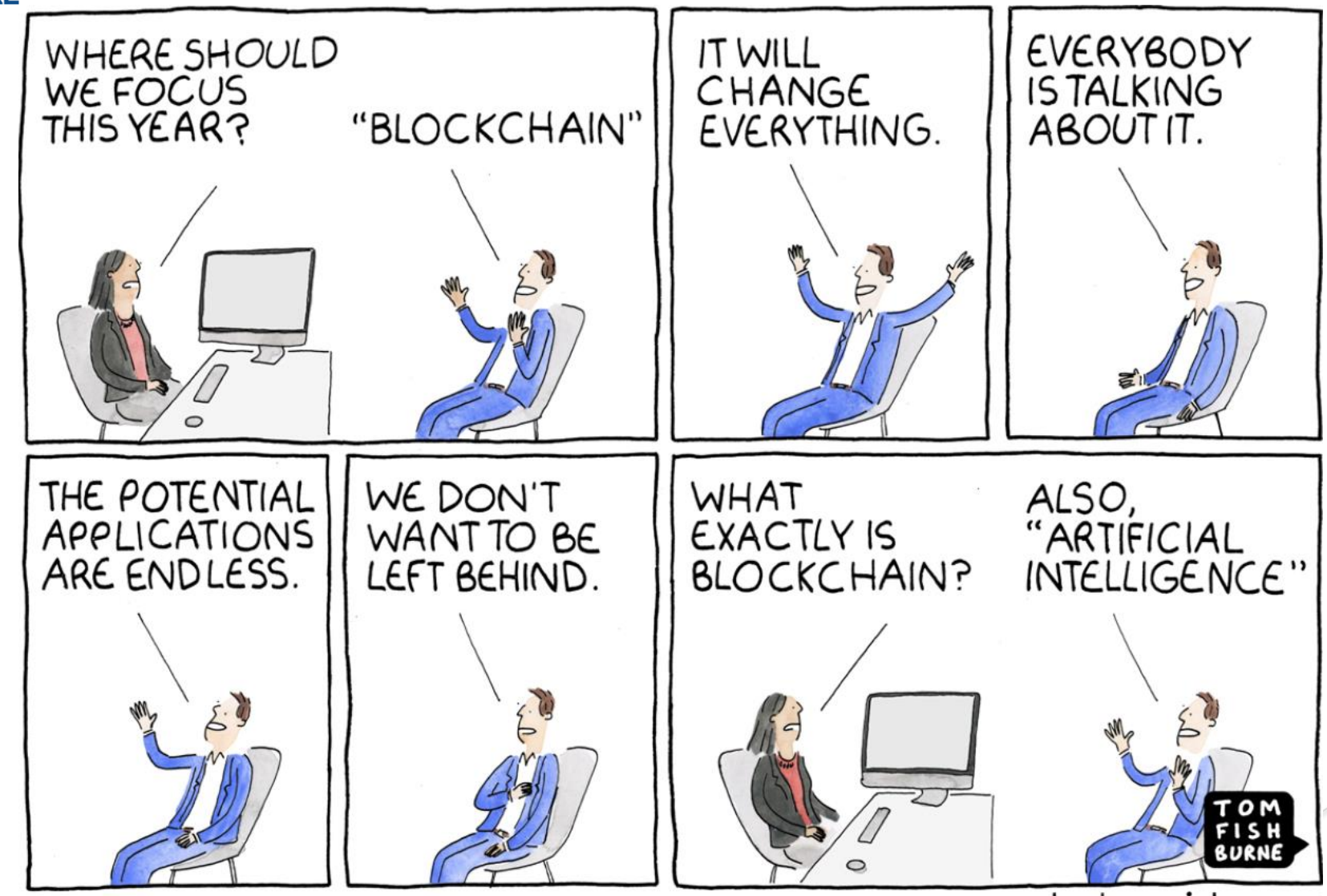
Co-funded by the
Erasmus+ Programme
of the European Union

- There has been **hype** around the use of **blockchain technology**
  - Yet the **technology is not well understood…**
- **Tendency is to apply it to every sector in every imaginable way**

The narrative goes somewhat like this:

- This industry is inefficient
- We haven't solved the issues with current technologies
- **Blockchain** is new and different
- Lo and behold, blockchain will make the industry efficient

Co-funded by the
Erasmus+ Programme
of the European Union




Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic
Blockchain is not magic

Outline

Co-funded by the
Erasmus+ Programme
of the European Union

- Blockchain is **not magic**: it will not solve all problems
- Describe the method behind the magic (i.e., how blockchain technology work)
- High-level understanding of the technology
- Examples from **cryptocurrency** domain
  - Blockchain technology is the foundation of modern cryptocurrencies
  - The first blockchain based cryptocurrency was **Bitcoin**
  - **Bitcoin** was the first of many blockchain applications

Blockchain Start-Ups

Co-funded by the
Erasmus+ Programme
of the European Union

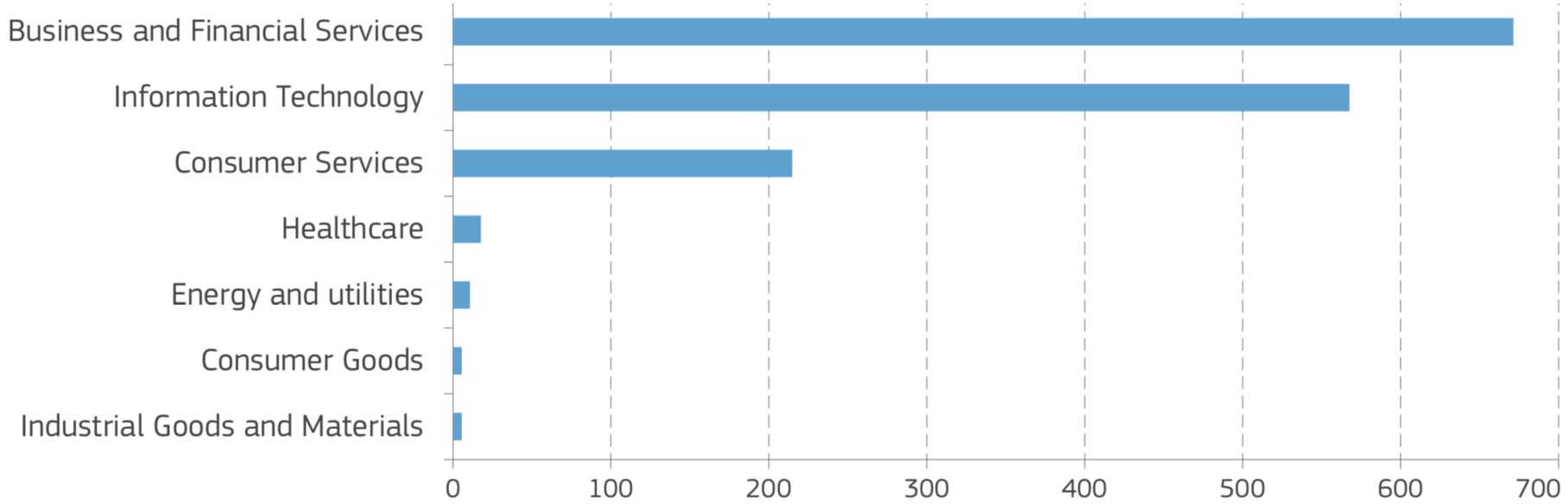


Figure 7: Blockchain start-ups across sectors of economic activity
Note: 1 211 firms of which 283 are operating in more than one sector
Source: Venture Sources - Dow Jones

Cryptocurrencies

Co-funded by the Erasmus+ Programme of the European Union

Cryptos: 10,281 Exchanges: 383 Market Cap: $1,656,396,468,509 24h Vol: $128,665,511,223 Dominance: BTC: 41.7% ETH: 18.7% ETH Gas: 19 Gwei

CoinMarketCap Cryptocurrencies Exchanges NFT Portfolio Watchlist Calendars Products Learn

4th June 2021

All Cryptocurrencies

#	Name	Price	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply
1	Bitcoin BTC Buy	$36,816.90	5.24%	1.31%	$689,860,340,308	$41,603,131,749 1,129,338 BTC	18,726,612 BTC
2	Ethereum ETH Buy	$2,659.21	6.09%	5.33%	$309,443,201,482	$32,863,457,137 12,335,889 ETH	116,155,062 ETH
3	Tether USDT Buy	$1.00	0.05%	0.00%	$62,008,044,427	$92,180,332,908 92,076,588,270 USDT	61,938,257,284 USDT

https://coinmarketcap.com

Blockchain Technology

Co-funded by the
Erasmus+ Programme
of the European Union

"Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance). New blocks are replicated across copies of the ledger within the network, and any conflicts are resolved automatically using established rules." [NISTIR 8202]

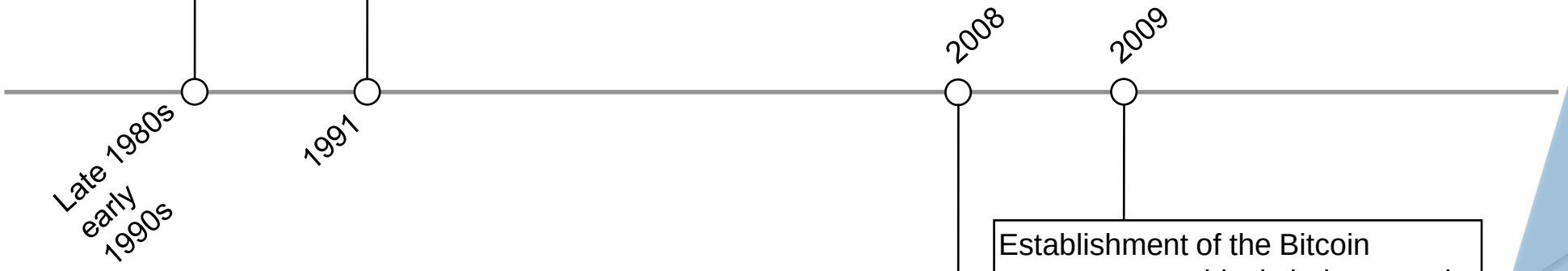
CRYPTOGRAPHY **DATA STRUCTURE** **DISTRIBUTED SYSTEMS**

A Bit of History

Co-funded by the
Erasmus+ Programme
of the European Union

Core ideas behind blockchain technology
L. Lamport developed the Paxos protocol: consensus model for reaching agreement on a result in a network of computers where the computers or network itself may be unreliable

A signed chain of information was used as an electronic ledger for digitally signing documents in a way that could easily show none of the signed documents in the collection had been changed

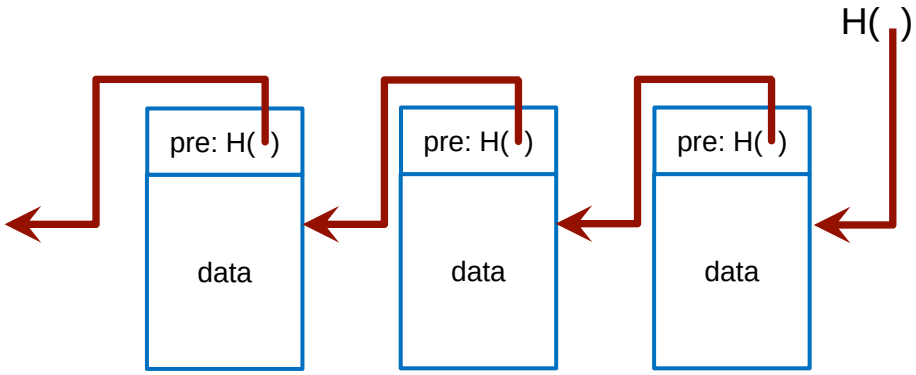


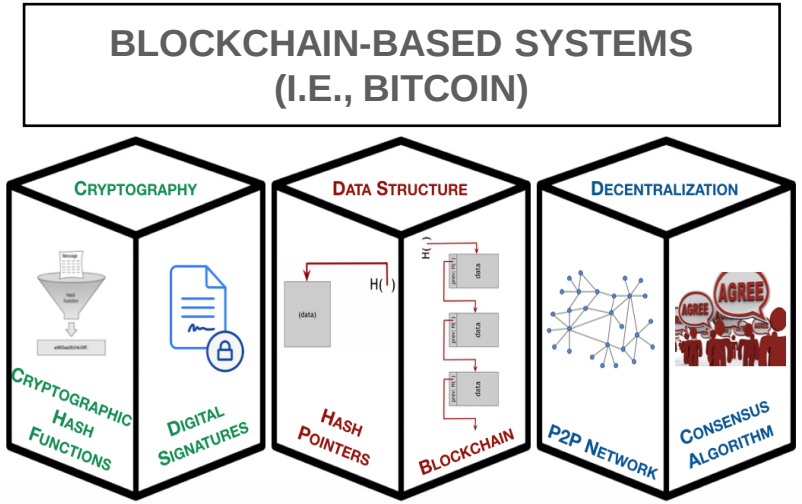
Establishment of the Bitcoin cryptocurrency blockchain network

These concepts combined and applied to **electronic cash**
- "Bitcoin: A Peer to Peer Electronic Cash System", published pseudonymously by Satoshi Nakamoto
- Nakamoto's paper contained the blueprint that most modern cryptocurrency schemes follow

Co-funded by the
Erasmus+ Programme
of the European Union

Blockchain is a data structure





Blockchain technology refers to the technologies (from the 3 domains) that are used to design and implement a system based on blockchain

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

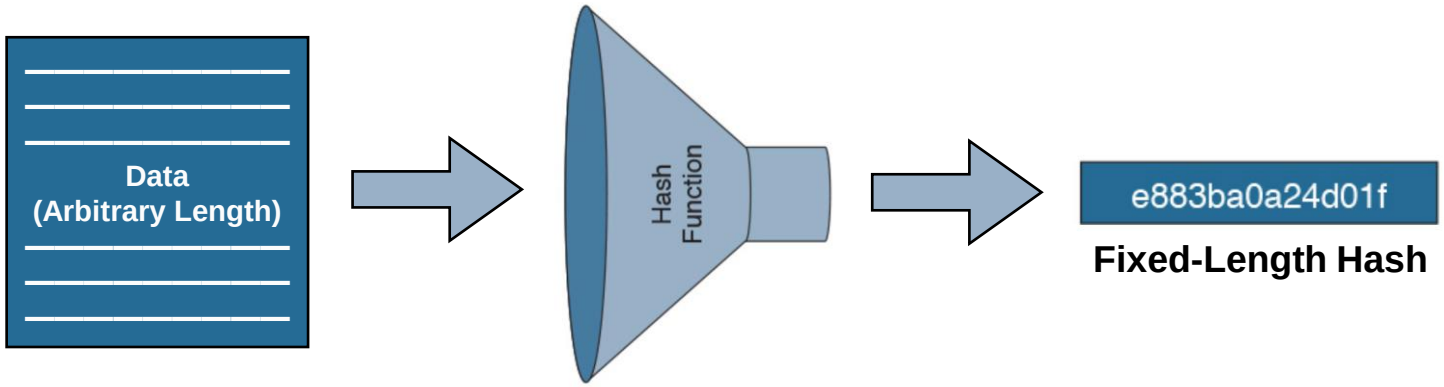
PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

General Hash Function

Co-funded by the
Erasmus+ Programme
of the European Union

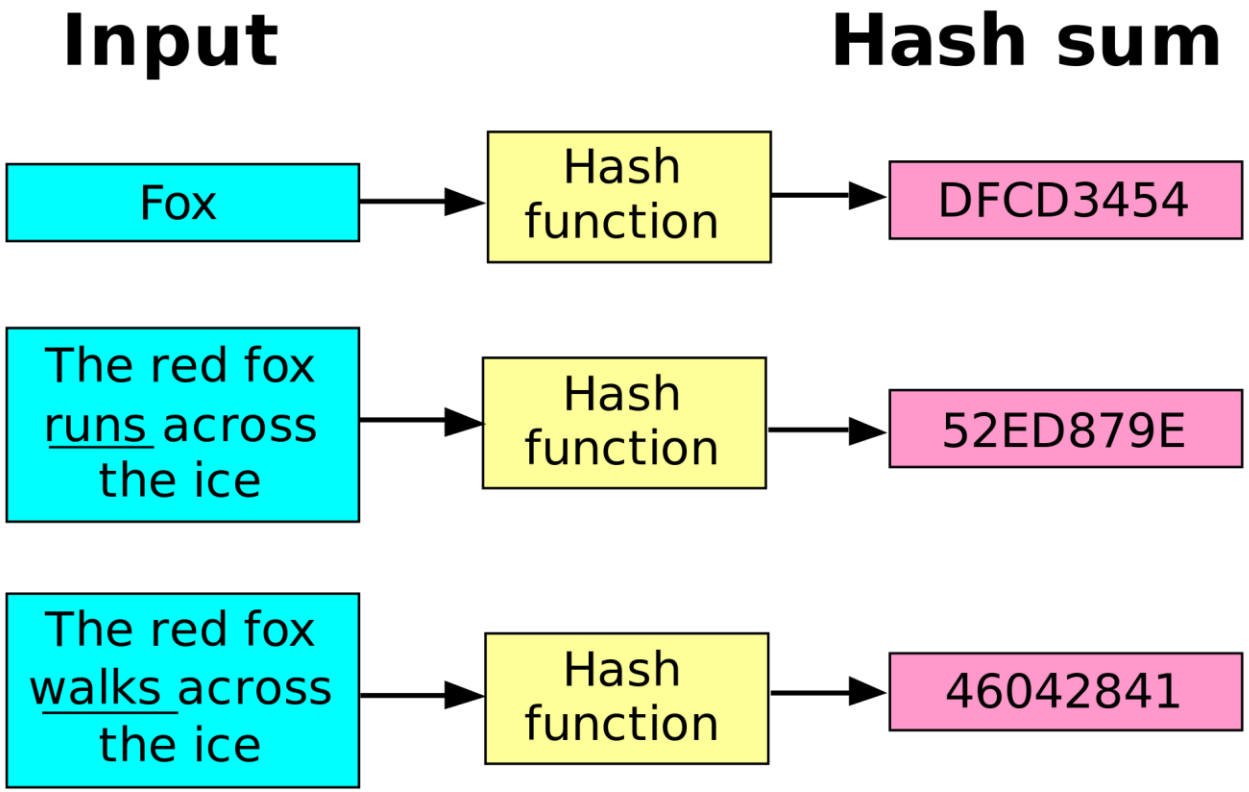
MATHEMATICAL FUNCTION WITH THE FOLLOWING **3 PROPERTIES**:

1. THE **INPUT** CAN BE ANY STRING OF **ANY SIZE**
2. IT PRODUCES A **FIXED-SIZE OUTPUT** (FOR INSTANCE, **256-BIT LONG**)
3. IT IS **EFFICIENTLY COMPUTABLE** (SAY, ***O(N)*** FOR ***N***-BIT STRING)



Co-funded by the
Erasmus+ Programme
of the European Union

General Hash Function: Examples



Co-funded by the
Erasmus+ Programme
of the European Union

Cryptographic Hash Functions

A **HASH FUNCTION** IS **CRYPTOGRAPHICALLY SECURE** IF IT SATISFIES THE FOLLOWING **3 SECURITY PROPERTIES**:

1. **COLLISION RESISTANCE**
2. **PREIMAGE RESISTANT**
3. **SECOND PREIMAGE RESISTANT**

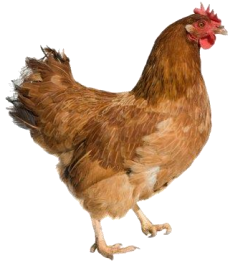
Co-funded by the
Erasmus+ Programme
of the European Union

Property 1: Collision Resistance

<u>Very hard</u> to find two different chickens that produces the same chicken nugget (by using the same machine)

Co-funded by the
Erasmus+ Programme
of the European Union

Property 2: Preimage Resistant



Given a chicken nugget, very hard to find the related chicken

Co-funded by the
Erasmus+ Programme
of the European Union

Property 3: Second Preimage Resistant

Given a chicken, very hard to find a different chicken that produces the same chicken nugget (by using the same machine)

?

Co-funded by the
Erasmus+ Programme
of the European Union

ENGINE

Cryptographic Hash Functions

Very easy

Very hard

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

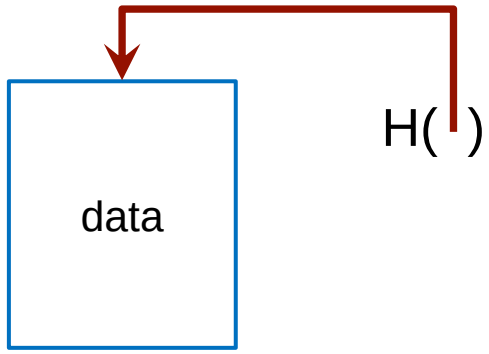
- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Hash Pointers

Co-funded by the
Erasmus+ Programme
of the European Union

- A **hash pointer** is
  - a **pointer** to **where** some information is stored **AND**
  - a **cryptographic hash** of the information
- If we have a hash pointer, we can:
  - get the **info** back
  - **verify** that the info hasn't changed



Co-funded by the
Erasmus+ Programme
of the European Union

Hash Pointers and Data Structures

KEY IDEA:

BUILD DATA STRUCTURES WITH HASH POINTERS

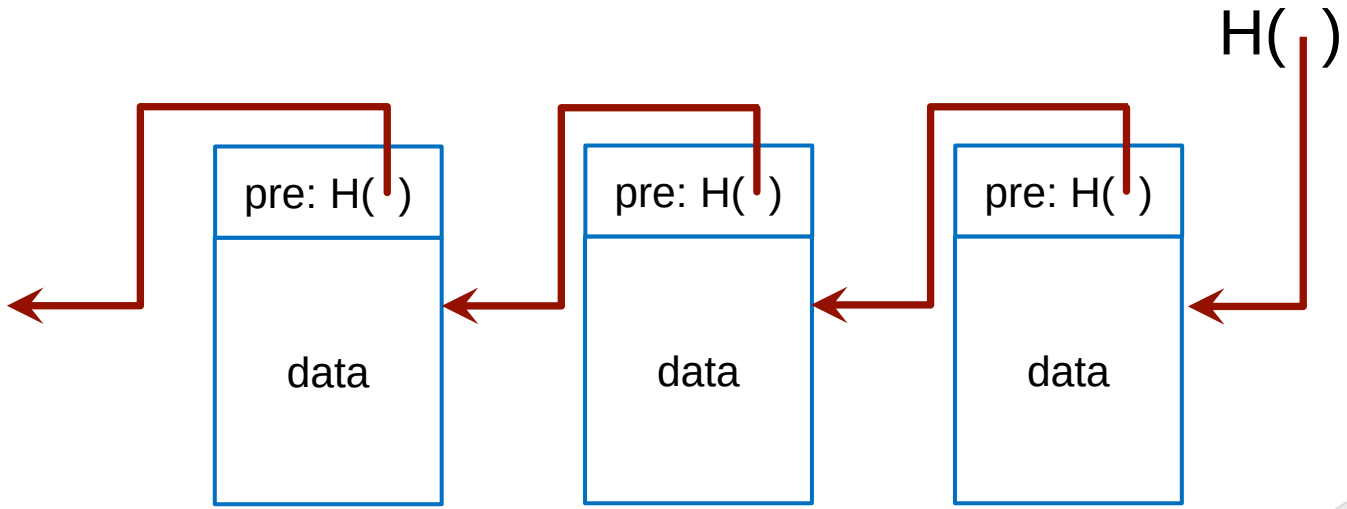
- We can **use hash pointers to build all kinds of data structures**

- Intuitively, we can take a familiar data structure that uses pointers such as a **linked list** or a **binary search tree** and **implement it with hash pointers**, instead of pointers as we normally would

Co-funded by the
Erasmus+ Programme
of the European Union

Linked List with Hash Pointers: Blockchain

- A **blockchain** is a **linked list** that is built **with hash pointers** instead of pointers
  - Each block not only tells us **where** the value of the previous block is, but it also contains a **digest of that value** that allows **to verify that the value hasn't changed**
  - The head of the list is just a **regular hash-pointer that points to the most recent data block**



Co-funded by the
Erasmus+ Programme
of the European Union

Use Case: Tamper-Evident Log

- A **log data structure** that stores a **bunch of data**, and allows us **to append data onto the end of the log**

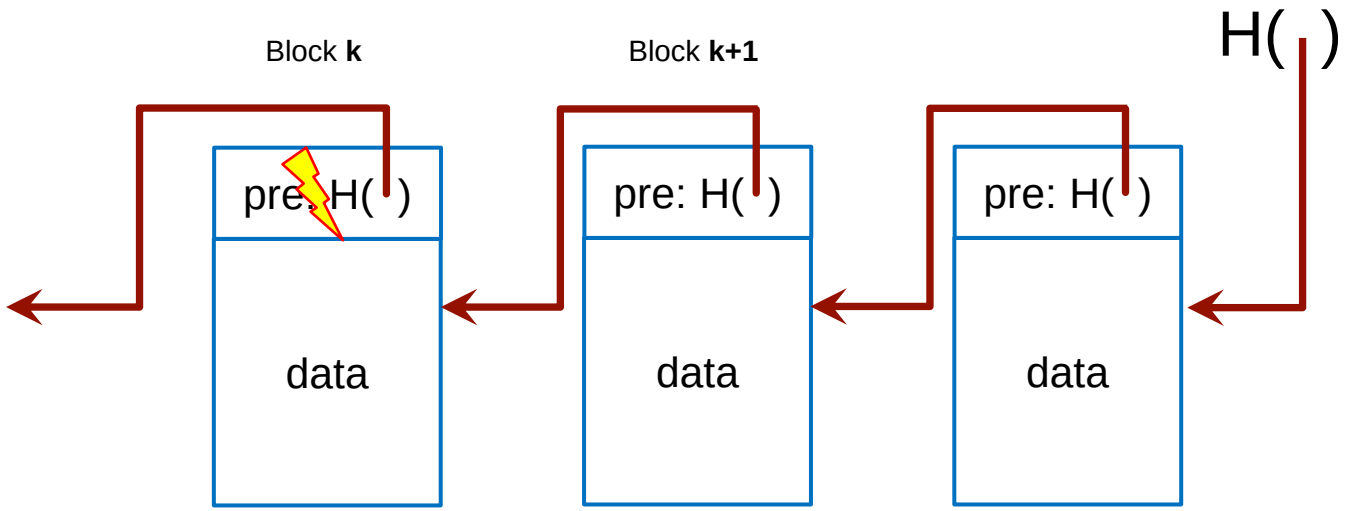
KEY PROPERTY

IF SOMEBODY ALTERS DATA THAT IS EARLIER IN THE LOG, WE DETECT IT

*… to understand why a **blockchain achieves** this **tamper-evident property**, let's see what happens if an adversary wants to tamper with data…*

Co-funded by the
Erasmus+ Programme
of the European Union

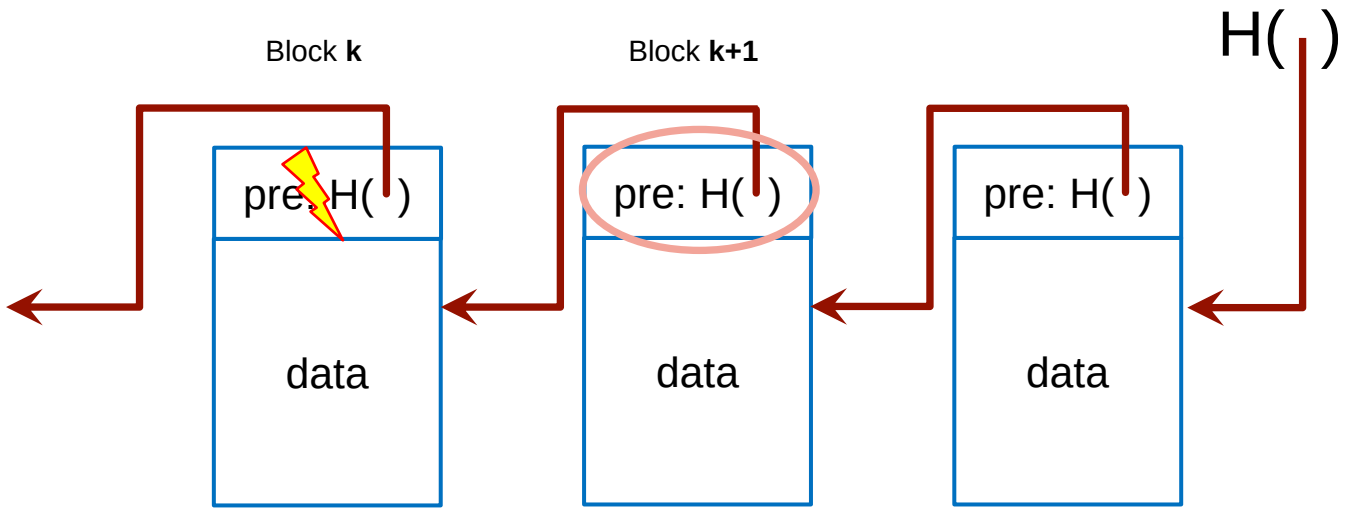
Use Case: Tamper-Evident Log



- To achieve this goal, the adversary changes the data of some block **k**

Co-funded by the
Erasmus+ Programme
of the European Union

Use Case: Tamper-Evident Log

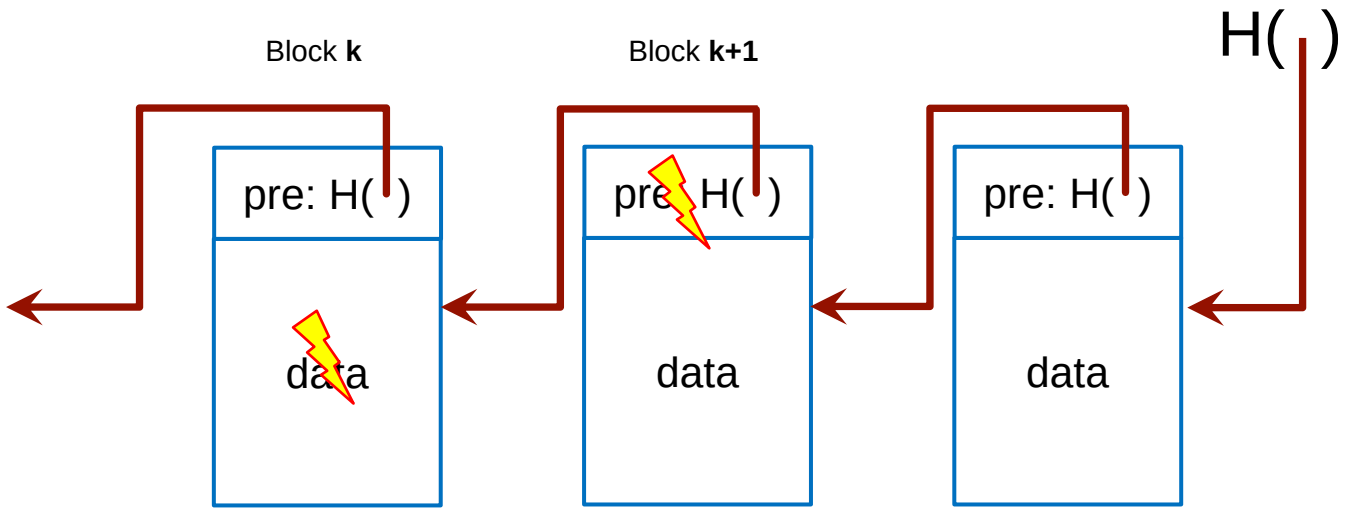


- Since the data has been changed, the hash in block **k + 1** (which is a hash of the entire block **k**) is not going to match up: so we can **detect the inconsistency**!

- **N.B.**: we are **STATISTICALLY GUARANTEED** that the new hash will not match the altered content since the **HASH FUNCTION IS COLLISION RESISTANT**

Co-funded by the
Erasmus+ Programme
of the European Union

Use Case: Tamper-Evident Log

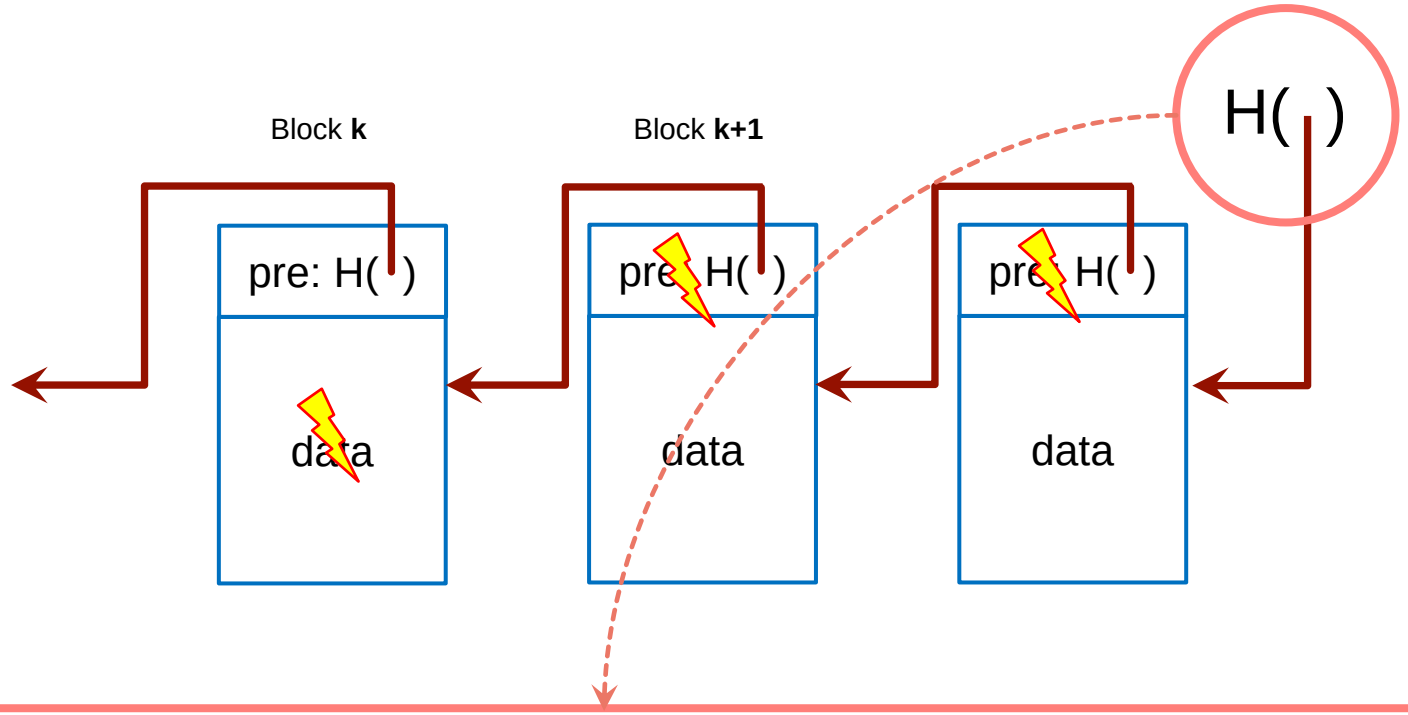


- The adversary can continue to try and cover up this change by **changing the next block's hash** as well

- The adversary can continue doing this, but this strategy will fail when he reaches the head of the list

Co-funded by the
Erasmus+ Programme
of the European Union

Use Case: Tamper-Evident Log



AS LONG AS WE STORE THE HASH POINTER AT THE HEAD OF THE LIST IN A PLACE WHERE THE ADVERSARY CANNOT CHANGE IT, THE ADVERSARY WILL BE UNABLE TO CHANGE ANY BLOCK WITHOUT BEING DETECTED

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Digital Signatures

Co-funded by the
Erasmus+ Programme
of the European Union

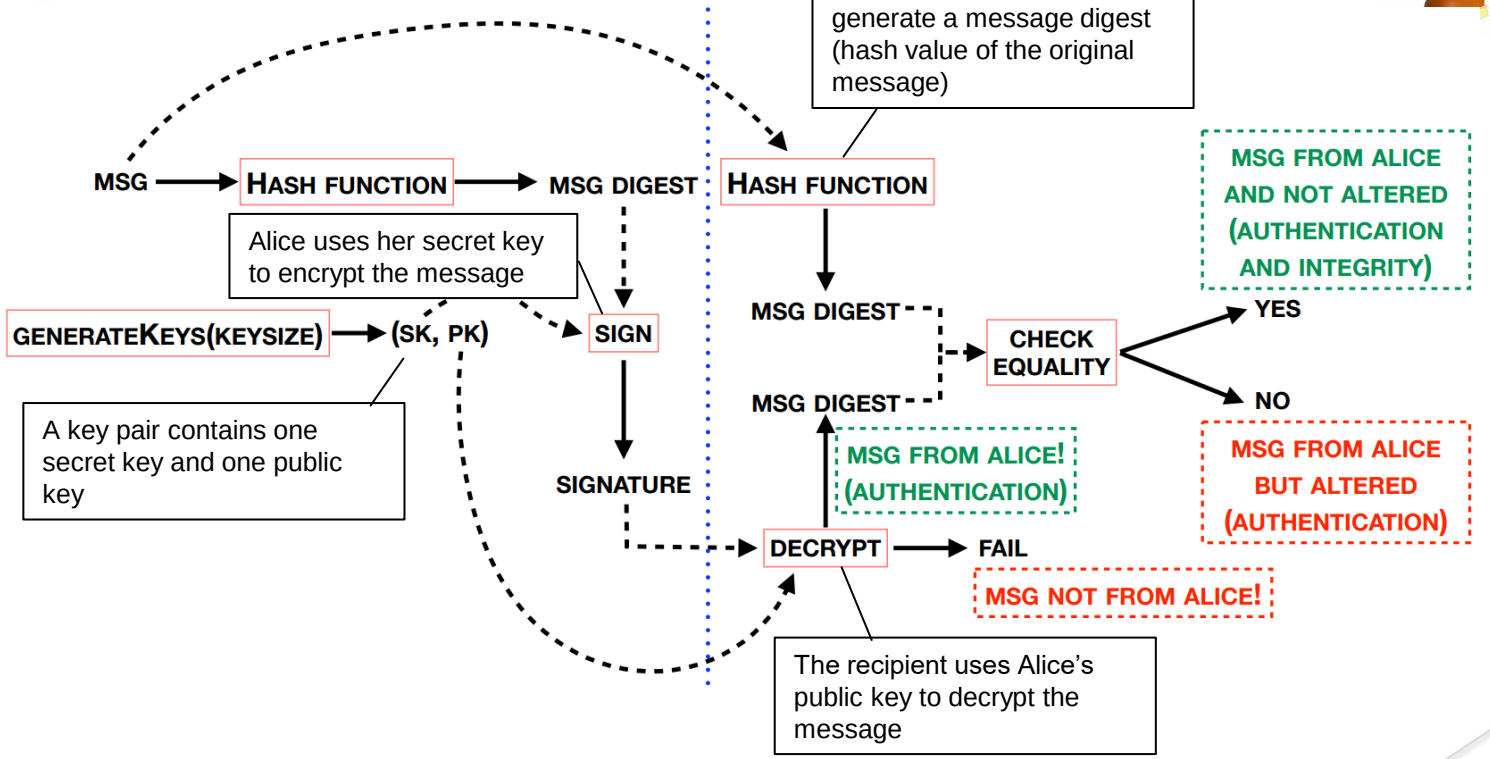
- **Digital signature**: **digital analog** to a **handwritten signature** on paper
- Meant for **authenticity** (authentication + integrity): the ability to determine that statements, policies, and permissions issued by persons or systems are genuine/authentic
- **Two key properties**:
  - only **you** can **sign with a private key**, but **anyone** who sees the signature can **verify** that it's **valid through your public key**
  - signature is **tied to a particular document**
    - i.e., a signature cannot be cut-and-pasted to another document

How can we build this in a digital form using cryptography?

Digital Signatures

Co-funded by the Erasmus+ Programme of the European Union

(MSG, SIGNATURE)



Public Key = Identity

Co-funded by the
Erasmus+ Programme
of the European Union

DECENTRALIZED IDENTITY MANAGEMENT

- No central point of coordination
- **Anybody** can make a **new identity at any time** (creating a **key-pair**)
  - *You can make as many as you want!*
- These identities are called **"addresses" in Bitcoin**

PRIVACY

- Addresses **not** directly connected to real-world identity
- But observer can link together an address's activity over time, make inferences

STORING BITCOINS IS REALLY ALL ABOUT STORING AND MANAGING BITCOIN SECRET KEYS

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

GoofyCoin

Co-funded by the
Erasmus+ Programme
of the European Union

- Only **2 rules**:
  1. A designated entity, **Goofy**, can **CREATE NEW COINS**
     - **Newly created coins belong to him**
  2. Whoever owns a coin can **transfer** it on to someone else

Co-funded by the
Erasmus+ Programme
of the European Union

How Goofy Creates New Coins

2. He then **computes the digital signature** of this string with his **secret signing key**

signed by pk_{Goofy}

CreateCoin [uniqueCoinID]

The string, together with **Goofy**'s signature, is a **coin**

1. **Goofy** generates a **unique coin ID uniqueCoinID** that he has never generated before and constructs the string **"CreateCoin [uniqueCoinID]"**

- **Anyone** can **verify** that the coin contains **Goofy**'s valid signature of a **CreateCoin** statement, and is therefore a valid coin

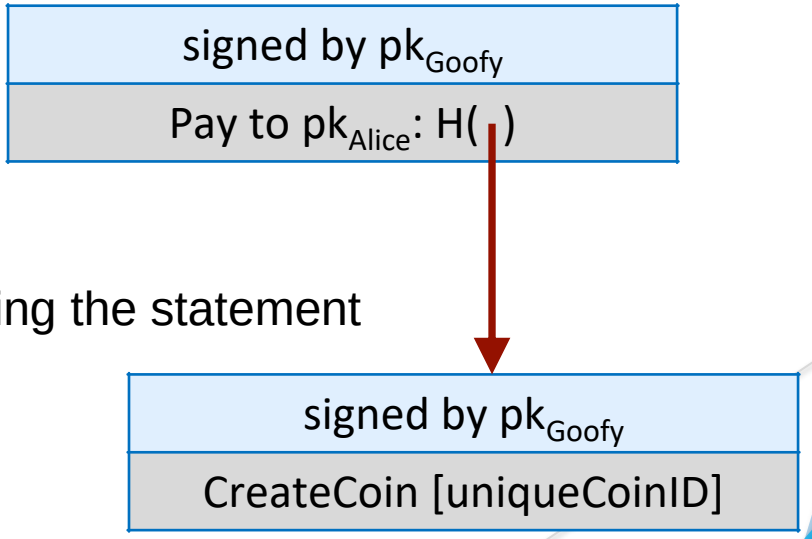
Co-funded by the
Erasmus+ Programme
of the European Union

How Goofy Can Spend the Coins

- **Goofy** wants to **transfer** a coin that he created to **Alice**

1. To do this he **creates a new statement** that says *"Pay this to Alice"*
   - *"this"* is a **hash pointer that references the coin** in question
   - *"Alice"* refers to **Alice's public key**

   (identities are just public keys)

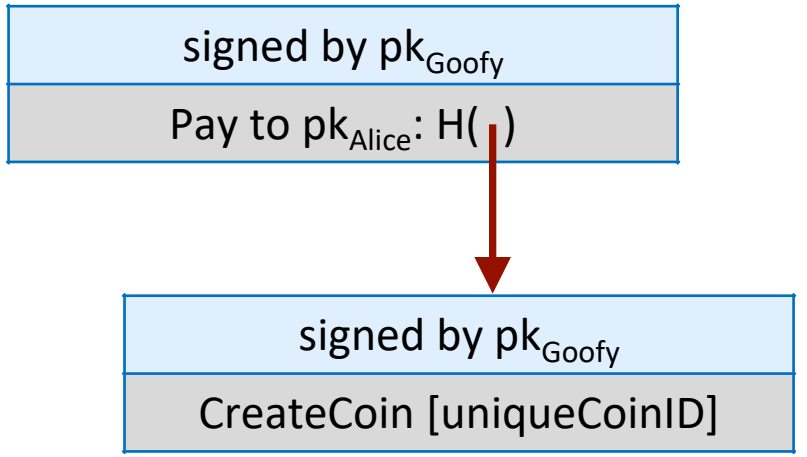


2. Finally, **Goofy signs the string** representing the statement

Co-funded by the
Erasmus+ Programme
of the European Union

Why Does Alice Own the Coin Now?

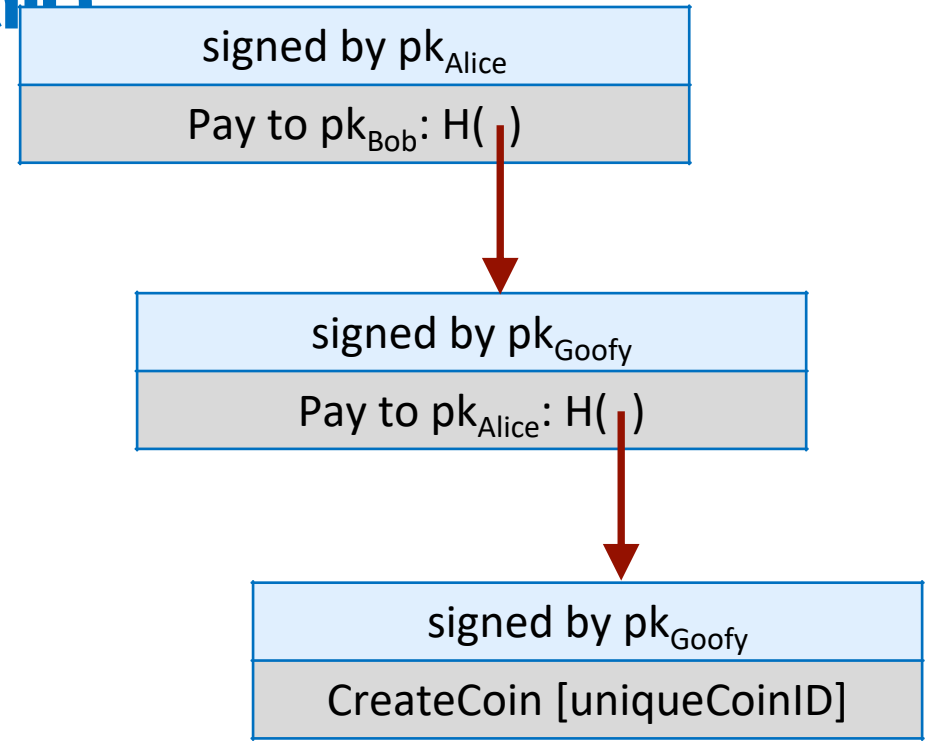
- **Alice** can **prove** to anyone that she **owns** the coin, because she can present the data structure with **Goofy**'s valid signature



The **validity and ownership of coins** are **self-evident in the system**!

Co-funded by the
Erasmus+ Programme
of the European Union

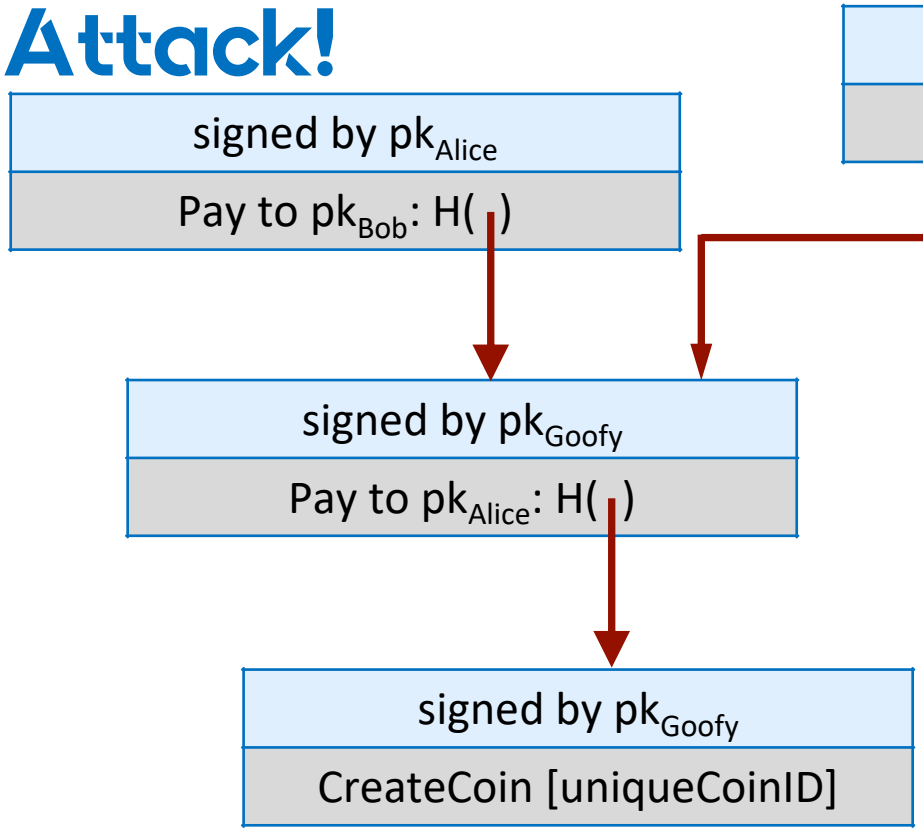
The Recipient Can Pass on the Coin Again



A coin has been created (bottom) and spent twice (middle and top)

Co-funded by the
Erasmus+ Programme
of the European Union

Security Problem: Double Spending Attack!



Bob and **Charles** both have **valid-looking claims to be the owner of this coin**!

Alice **creates** another **signed statement** that **pays the same coin to Charles**

Double-Spending

Co-funded by the
Erasmus+ Programme
of the European Union

Main **design challenge** in **all digital currencies**!

- To solve the problem, let's design another cryptocurrency: **ScroogeCoin**
- Built off of **GoofyCoin**, but a bit more complicated in terms of **data structures**

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

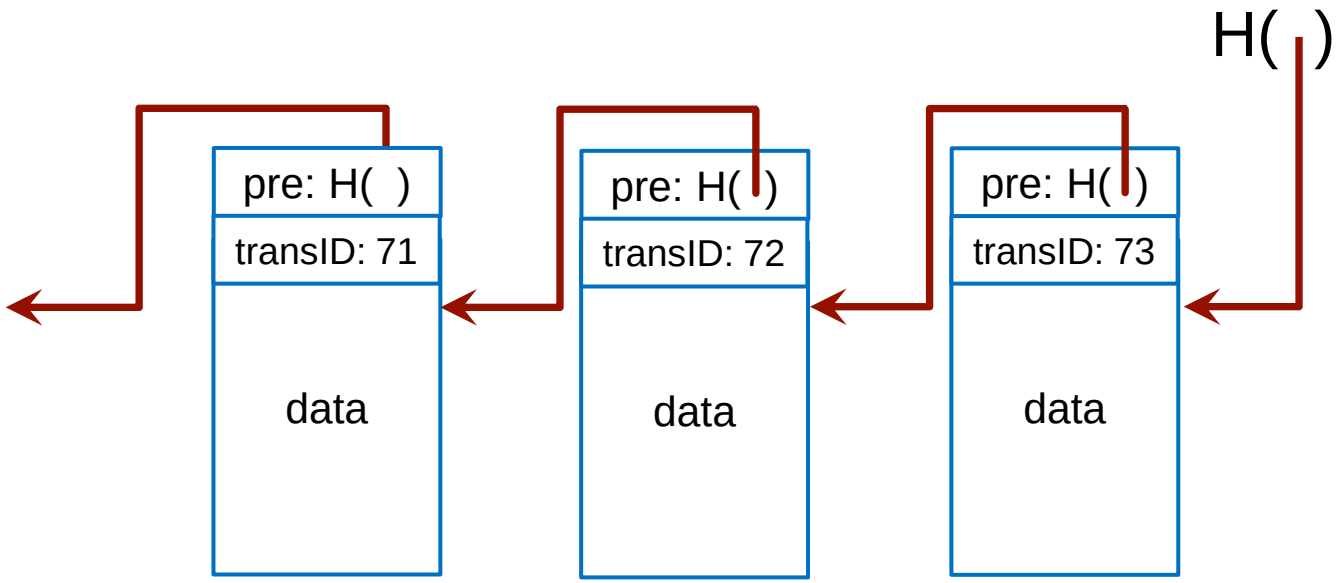
PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

ScroogeCoin

Co-funded by the
Erasmus+ Programme
of the European Union

- **KEY IDEA: RECORD TRANSACTIONS IN CENTRAL BLOCKCHAIN**

  - A designated entity, **Scrooge**, publishes an **append-only ledger** containing the **history of all the transactions that have happened**
  - The **append-only ledger** is a **blockchain** signed by **Scrooge**

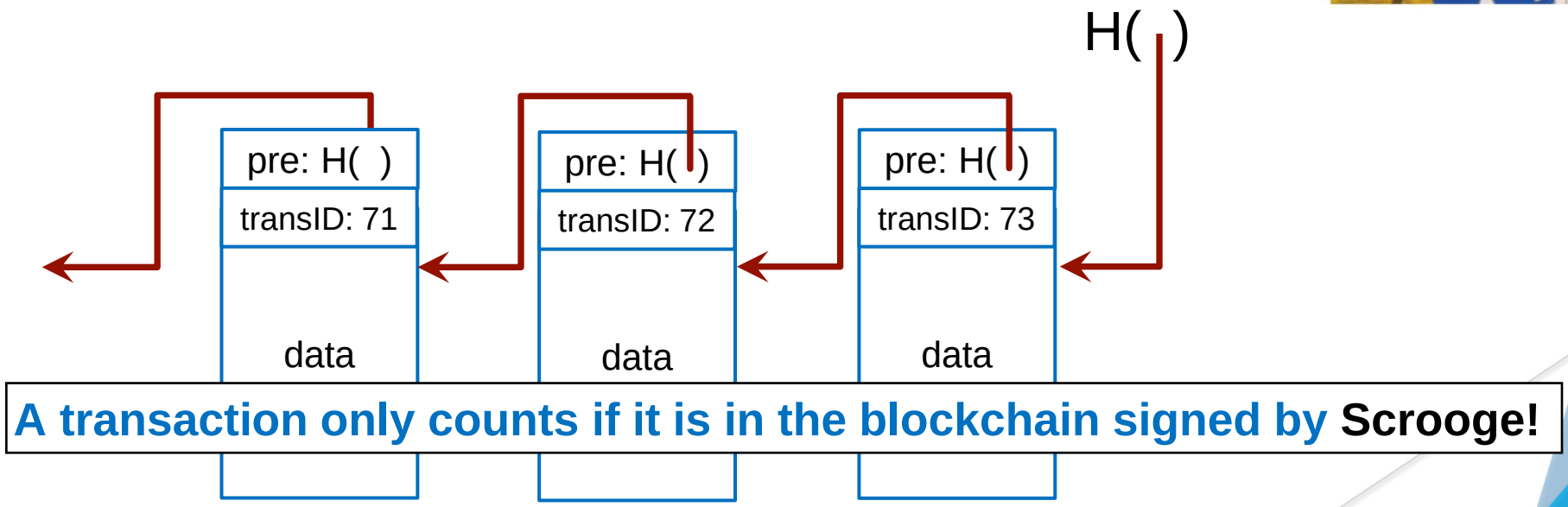


ScroogeCoin (2)

Co-funded by the
Erasmus+ Programme
of the European Union

- **KEY IDEA: RECORD TRANSACTIONS IN CENTRAL BLOCKCHAIN**

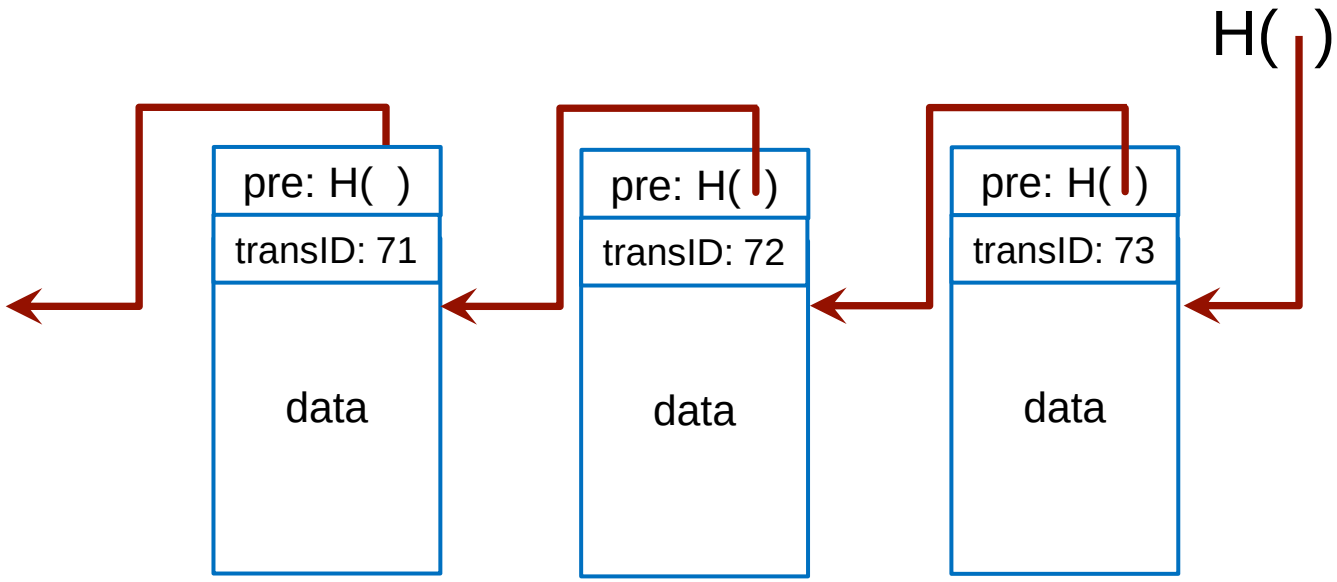
  - Each **block** has the **ID of a transaction**, the **transaction's contents**, and a **hash pointer to the previous block**
  - **Only Scrooge** can accept a transaction and append it to the blockchain



ScroogeCoin (3)

Co-funded by the
Erasmus+ Programme
of the European Union

- **KEY IDEA: RECORD TRANSACTIONS IN CENTRAL BLOCKCHAIN**
  - Scrooge **digitally signs the final hash pointer**, which binds all the data in this entire structure, and **publishes the signature along with the blockchain**

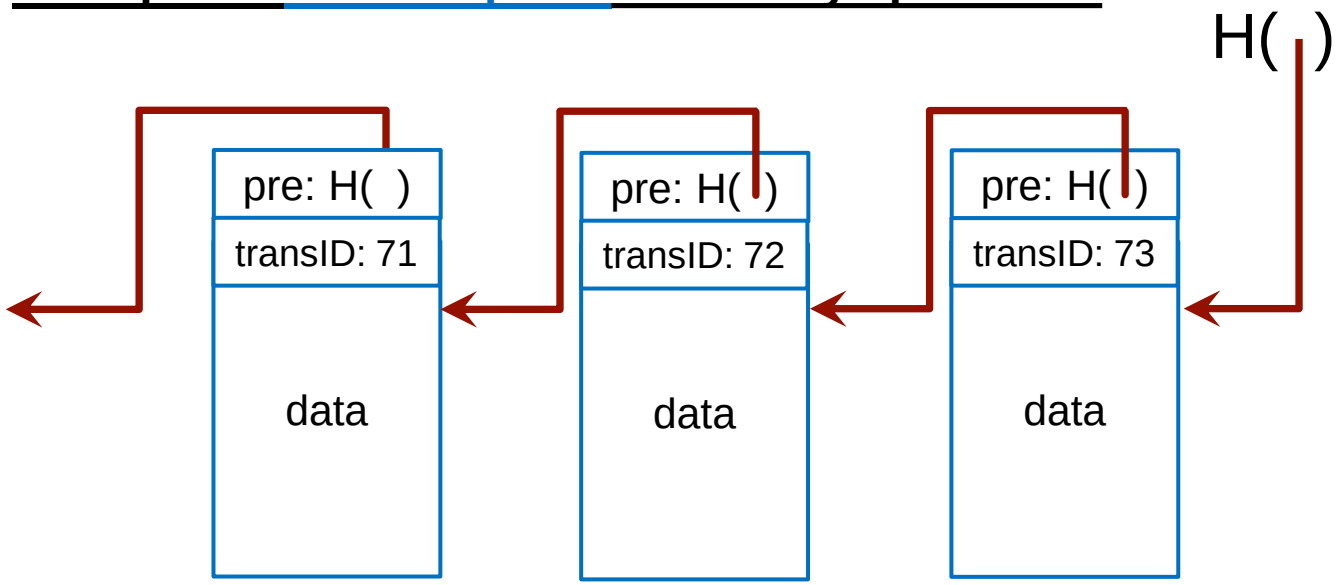


ScroogeCoin (4)

Co-funded by the
Erasmus+ Programme
of the European Union

- **KEY IDEA: RECORD TRANSACTIONS IN CENTRAL BLOCKCHAIN**

  - Anybody can **verify** that a transaction was endorsed by **Scrooge** by **checking Scrooge's signature**
  - **Scrooge** makes sure that **he doesn't endorse a transaction that attempts to double-spend an already spent coin**



Co-funded by the
Erasmus+ Programme
of the European Union

Creating New Coins in ScroogeCoin

- By means of the **CreateCoins** **transaction**
  - Like for **GoofyCoin**, but creates **multiple coins**

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

coinID 73(0)
coinID 73(1)
coinID 73(2)

We refer to coins by **coinIDs**

Each coin has a **serial number** within the transaction

Each coin also has a **value**

Each coin has a **recipient**, which is a **public key** that gets the coin when it's created

Co-funded by the
Erasmus+ Programme
of the European Union

How to Pay Coins in ScroogeCoin

- By means of the **PayCoins** **transaction**: **consumes** (and **destroys**) some coins, and **creates** new coins of the same total value

transID: 73 type:PayCoins		
consumed coinIDs: 68(1), 42(0), 72(3)		
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...
signatures		

A **transaction** is **VALID** if:

1. **Consumed coins** are **valid** (they were created in previous transactions)
2. **Consumed coins** were **not already consumed** in some previous transaction (to avoid double-spending)
3. Total value of the coins that come out of the transaction == total value of the coins that went in (only **Scrooge** can create new value)
4. The transaction is validly **signed by the owners of all of the consumed coins**

Co-funded by the
Erasmus+ Programme
of the European Union

How to Pay Coins in ScroogeCoin (2)

- By means of the **PayCoins** **transaction**: **consumes** (and **destroys**) some coins, and **creates** new coins of the same total value

transID: 73 type:PayCoins		
consumed coinIDs: 68(1), 42(0), 72(3)		
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...
signatures		

- If **valid**, **Scrooge** can **accept** the transaction
  - **Scrooge appends it** to the **blockchain**
- After that, everyone can see that this transaction has happened
  - It is only at this point that the participants can accept that the transaction has actually occurred
- Until it is published, it might be preempted by a **double-spending transaction**

Co-funded by the
Erasmus+ Programme
of the European Union

The Problem with ScroogeCoin

KEY PROBLEM IS CENTRALIZATION

Basic example of **permissioned** blockchain

CAN WE "DESCROOGIFY" THE CURRENCY

AND OPERATE WITHOUT ANY

CENTRAL, TRUSTED PARTY?

(to get permissionless blockchain)

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Co-funded by the
Erasmus+ Programme
of the European Union

Aspects of Decentralisation in Bitcoin

PEER-TO-PEER NETWORK

- **Anybody** can run a **Bitcoin** node
  - Download a **Bitcoin** **client** and run a node on your laptop or your PC

MINING

- **Open** **to anyone**
- But **requires a very high capital cost**: because of this there has been a high degree of **centralization**, or a **concentration of power** (seen as undesirable!)

UPDATE OF SOFTWARE THAT BITCOIN NODES RUN

- Numerous interoperable implementations of the protocol
- In practice: most nodes run the **reference implementation**, and its **developers are trusted by the community** and have a lot of power

Distributed Consensus

Co-funded by the
Erasmus+ Programme
of the European Union

- **Key technical problem** to solve in building a **distributed e-cash system**
  - Intuitively: the goal can be seen as **decentralizing ScroogeCoin**

> ***Distributed consensus protocol.*** There are *n* nodes that each have an input value. Some of these nodes are faulty or malicious. A distributed consensus protocol has the following two properties:
>
> - It must terminate with all honest nodes in agreement on the value
> - The value must have been generated by an honest node

- Has various applications
- Has been **studied for decades** in computer science

Consensus on What?

Co-funded by the Erasmus+ Programme of the European Union

- Given that a variety of users are broadcasting these transactions to the network, the **nodes must agree on**
  - **which transactions** were broadcast
  - **the order** in which these transactions happened
- This will result in a **single, global ledger for the system**
- Recall that in **ScroogeCoin**, for optimization, we put transactions into blocks
  - Similarly, in **Bitcoin**, **consensus** is done on a **block-by-block** basis

Co-funded by the
Erasmus+ Programme
of the European Union

Bitcoin Consensus Algorithm (Simplified)

Bitcoin consensus algorithm (simplified)

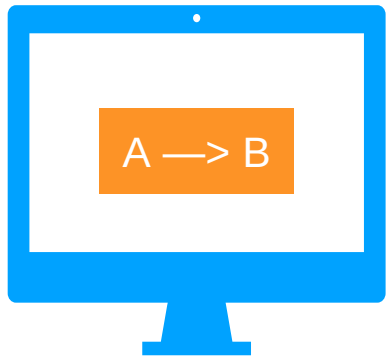
This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

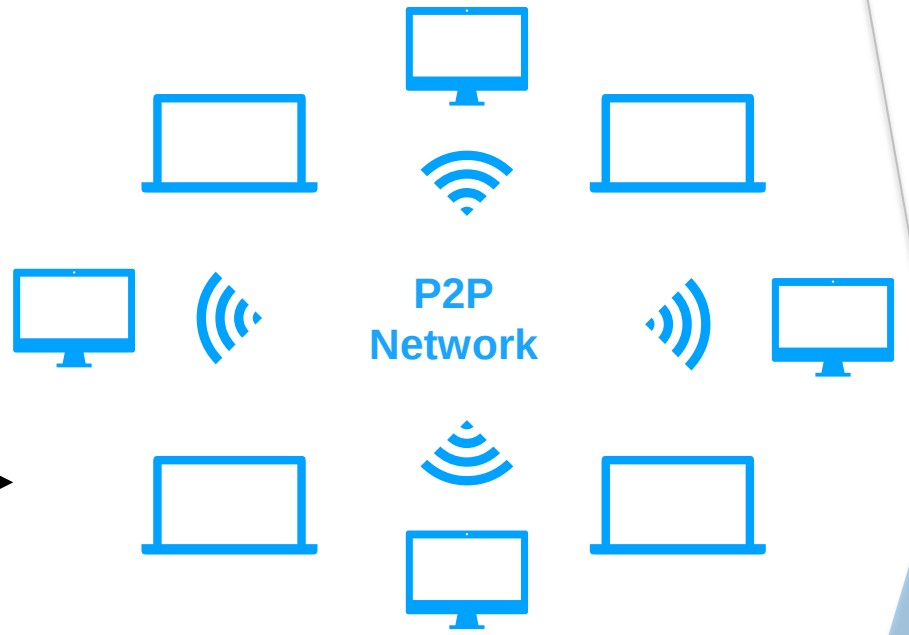
1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

Co-funded by the
Erasmus+ Programme
of the European Union

How a Blockchain-Based System Works

Someone requests a transaction (Alice pays Bob)



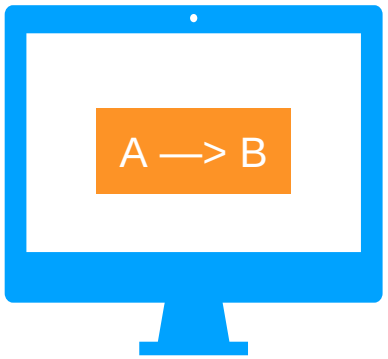


P2P network consists of connected computers known as "nodes"

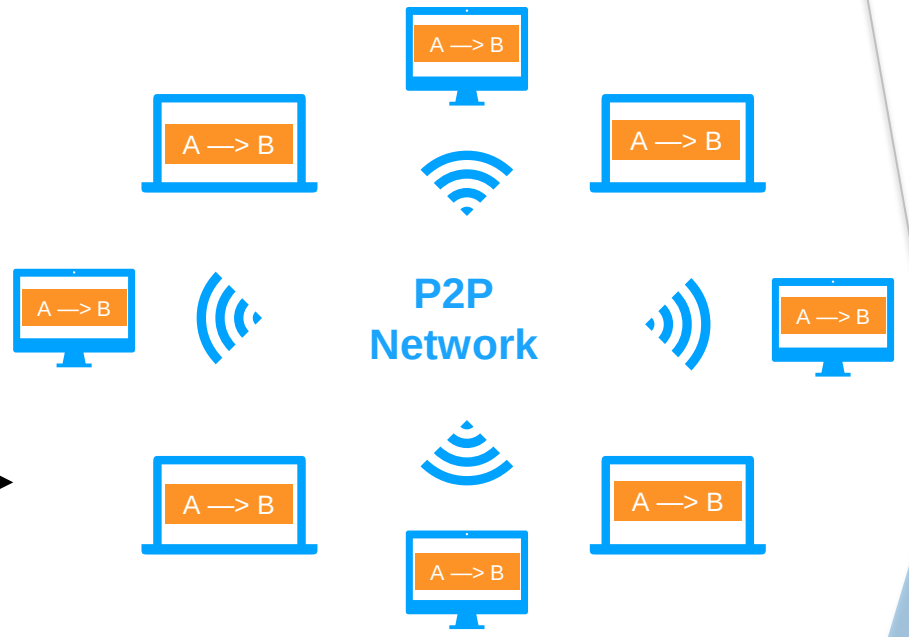
Co-funded by the
Erasmus+ Programme
of the European Union

How a Blockchain-Based System Works

Someone requests a transaction (Alice pays Bob)



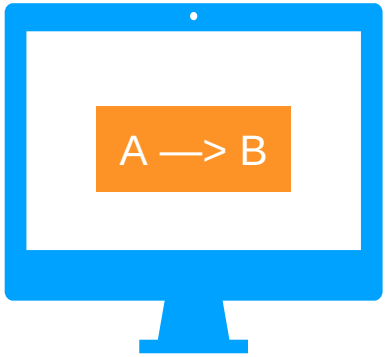
The transaction is broadcast to P2P network



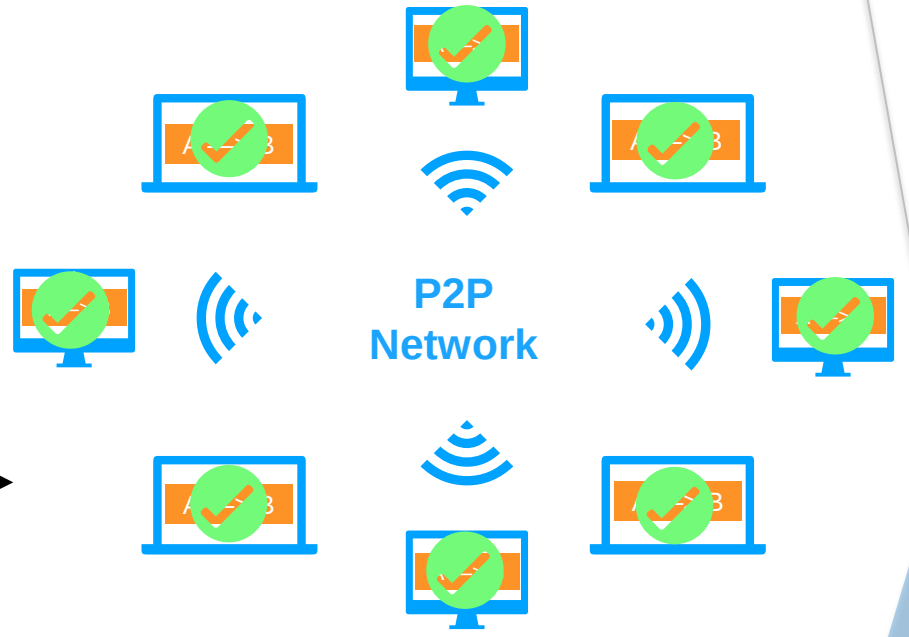
Co-funded by the
Erasmus+ Programme
of the European Union

How a Blockchain-Based System Works

Someone requests a transaction (Alice pays Bob)



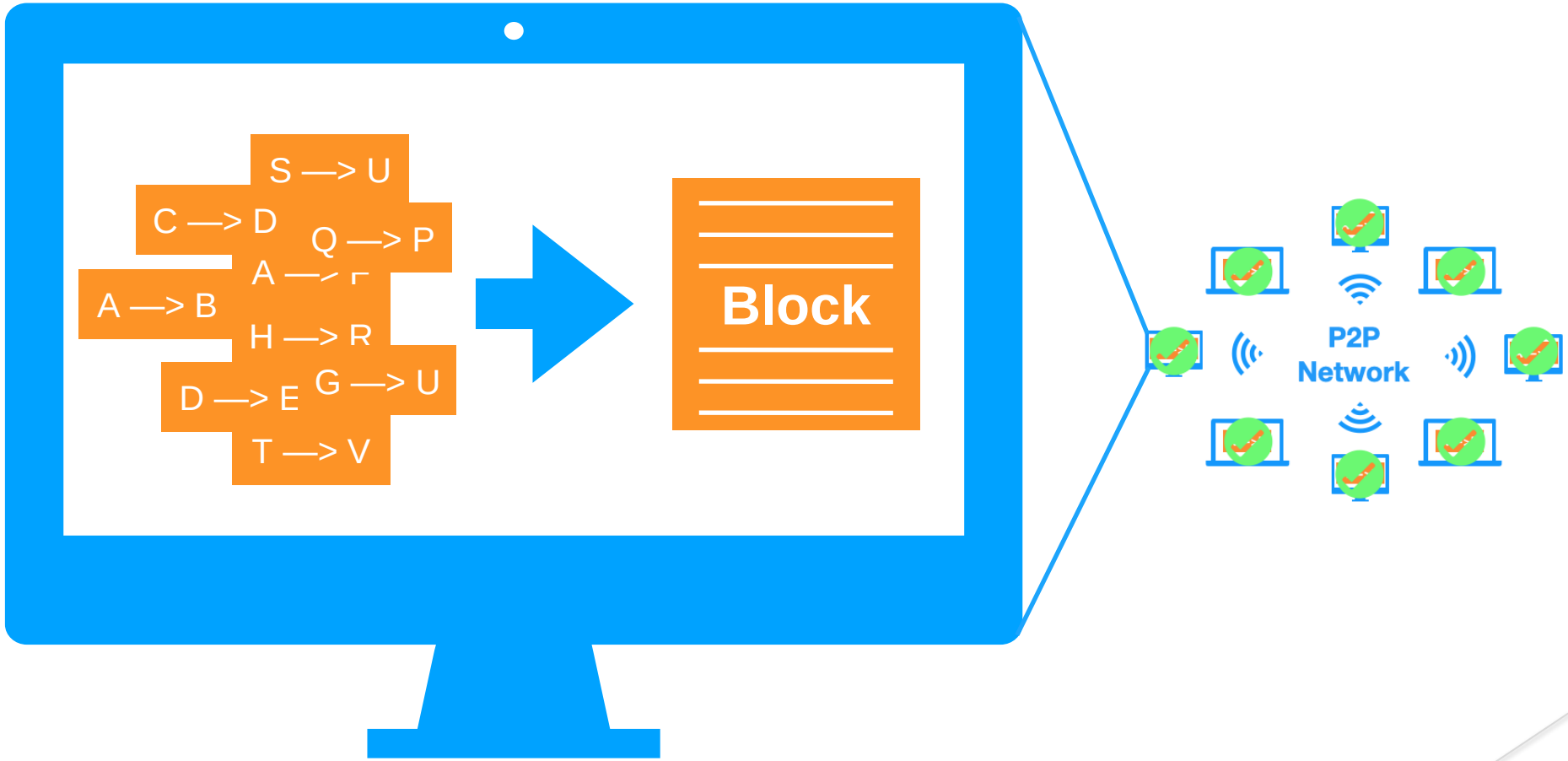
The transaction is broadcast to P2P network



VALIDATION
The network of nodes validates the transaction

Co-funded by the
Erasmus+ Programme
of the European Union

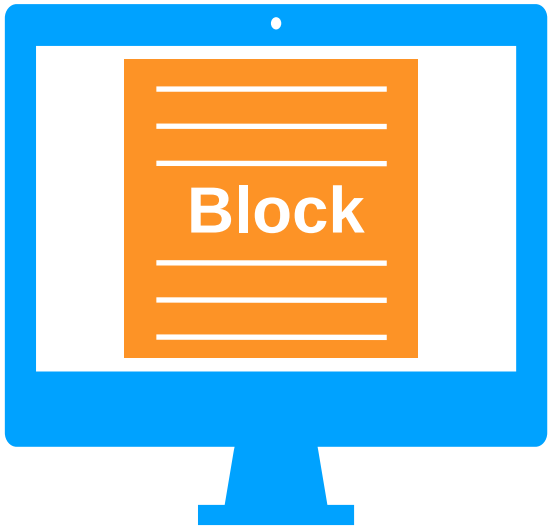
How a Blockchain-Based System Works



Once verified, the transaction is combined with other transactions to create a new block of data

Co-funded by the
Erasmus+ Programme
of the European Union

How a Blockchain-Based System Works

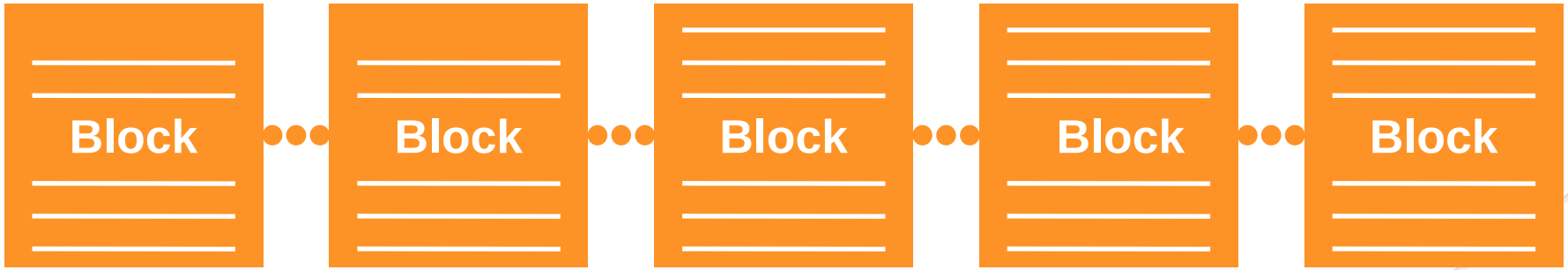


A (random) node will add the new block to the existing blockchain, in a way that is tamper evident and tamper resistant

How:

- New block sent to the P2P network
- Each node validates the block and add it to the blockchain (if valid)

The transaction is complete



Co-funded by the
Erasmus+ Programme
of the European Union

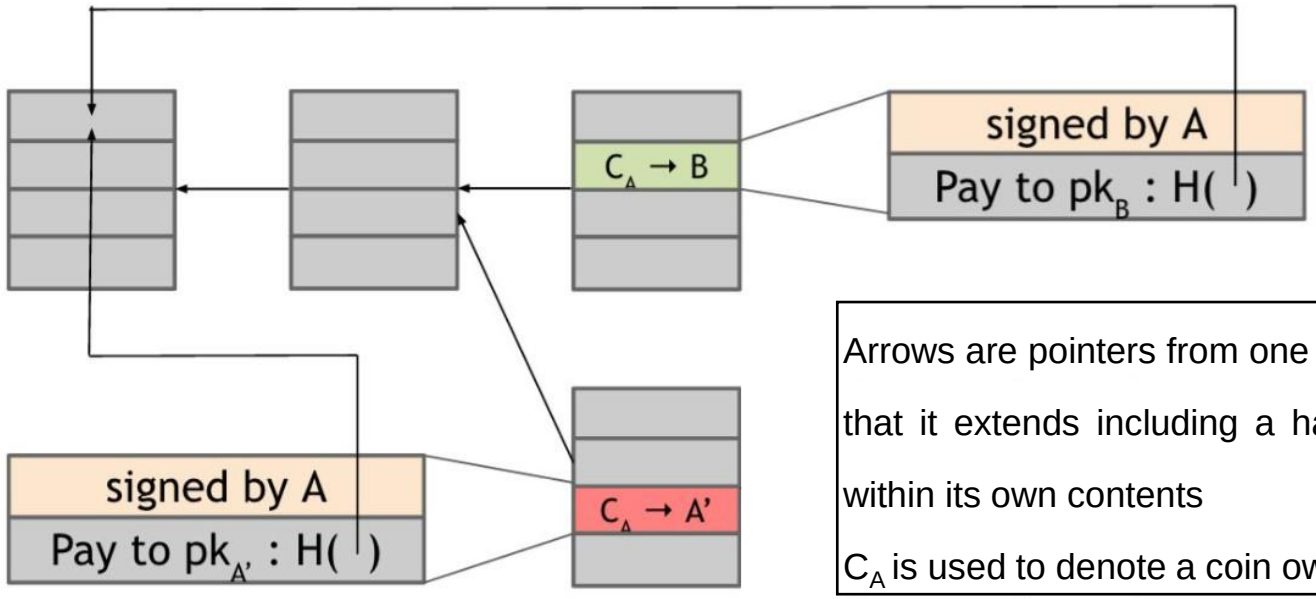
ENGINE

Is Double-Spend Attack Possible?

- For instance:
  - ***Alice*** *purchases service (i.e., buy sw) from* ***Bob*** *and pays in bitcoins*
  - ***Alice*** *creates transaction and broadcasts it to the network*
  - *Some honest node creates the next block including this transaction*
  - *So there is now a block that was created by an honest node that contains a transaction that represents a payment from* ***Alice*** *to the merchant* ***Bob***
  - *Upon seeing this transaction included in the blockchain,* ***Bob*** *concludes that* ***Alice*** *has paid him and allows* ***Alice*** *to use the service (i.e., download sw)*
  - *Later,* ***Alice*** *attempts to pay same coin to one of her accounts*

Co-funded by the
Erasmus+ Programme
of the European Union

Is Double-Spend Attack Possible? (2)



Arrows are pointers from one block to the previous block that it extends including a hash of that previous block within its own contents

C_A is used to denote a coin owned by Alice

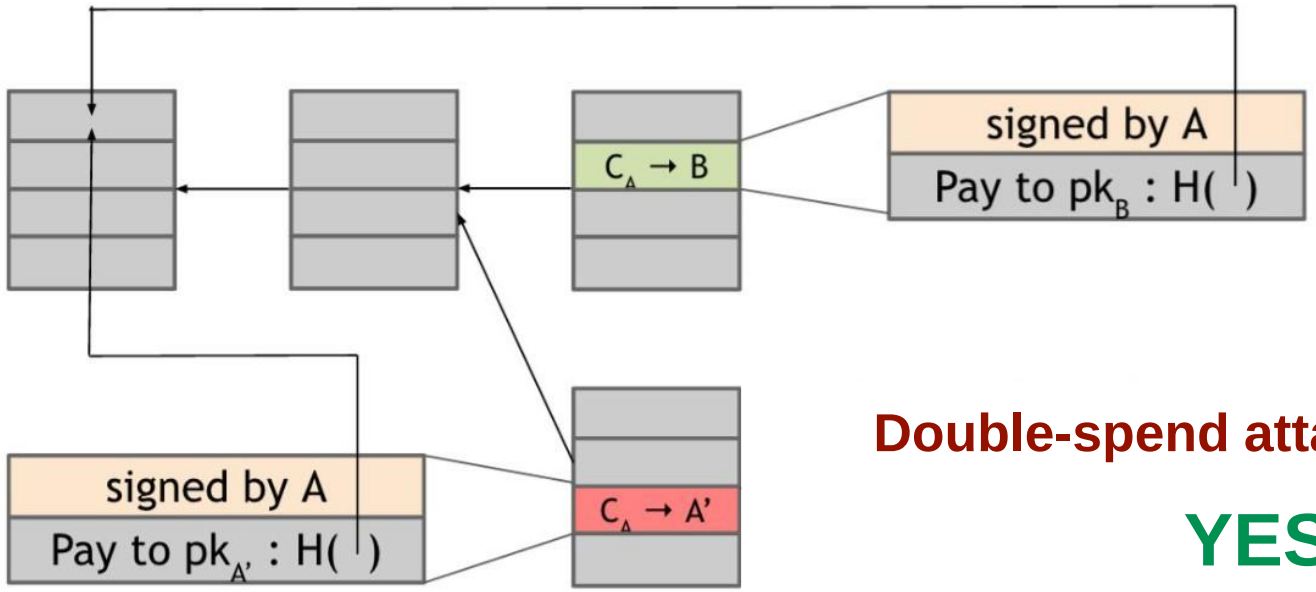
Alice creates **two transactions**:

- one in which she sends Bob **bitcoins**
- and one in which she double spends those **Bitcoins** by sending them to a different address that she controls (A')

*As they spend the same **bitcoins**, only one of these transactions can be included in the blockchain!*

Co-funded by the
Erasmus+ Programme
of the European Union

Is Double-Spend Attack Possible? (3)



Double-spend attack *possible?*

YES

What determines which block will be included in the blockchain, as they are both valid?

Honest nodes follow the policy of extending the longest valid branch, so which branch will they extend?

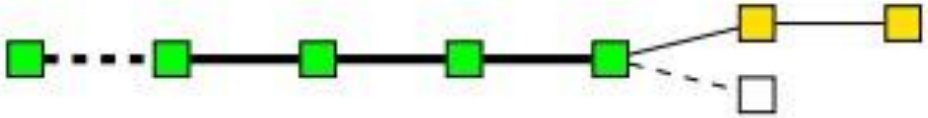
There is **no right answer**! At this point, ***the two branches are the same length*** — they only differ in the last block and both of these blocks are valid

The node that chooses the next block then may decide to build upon either one of them, and this choice will largely determine whether or not the double-spend succeeds

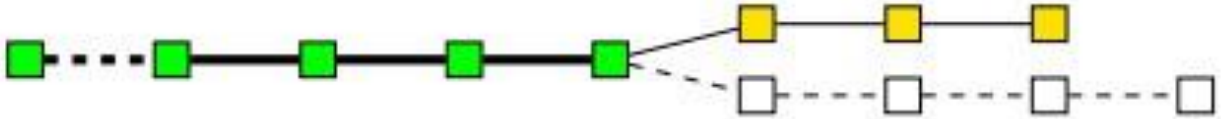
Co-funded by the
Erasmus+ Programme
of the European Union

In Summary

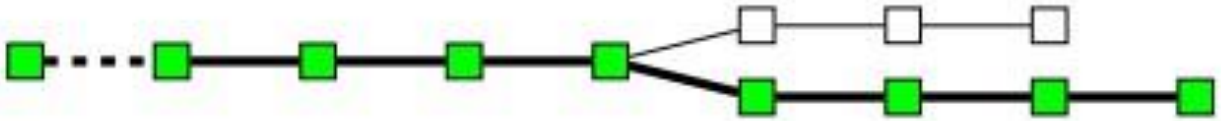
(a) Initial state of the blockchain in which all transactions are considered as valid.



(b) Honest nodes continue extending the valid chain by putting yellow blocks, while the attacker secretly starts mining a fraudulent branch.



(c) The attacker succeeds in making the fraudulent branch longer than the honest one.



(d) The attacker's branch is published and is now considered the valid one.

Co-funded by the
Erasmus+ Programme
of the European Union

Recap

PROTECTION AGAINST INVALID TRANSACTIONS:

✓ ENTIRELY **CRYPTOGRAPHIC**

✓ ENFORCED BY **CONSENSUS**

- Majority of (honest) nodes won't include the transaction in the blockchain

PROTECTION AGAINST DOUBLE-SPENDING IS PURELY BY CONSENSUS

- Transactions of a **double-spend** are **valid** from a cryptographic perspective
- Consensus determines which one will end up on the long-term consensus chain

YOU'RE NEVER 100% SURE THAT A TRANSACTION IS ON THE CONSENSUS BRANCH

- Guarantee is probabilistic

Co-funded by the
Erasmus+ Programme
of the European Union

How to Randomly Select a Node?

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block ?
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

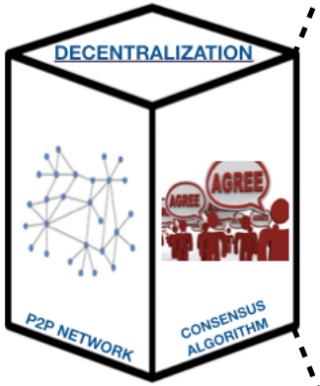
PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Consensus Models

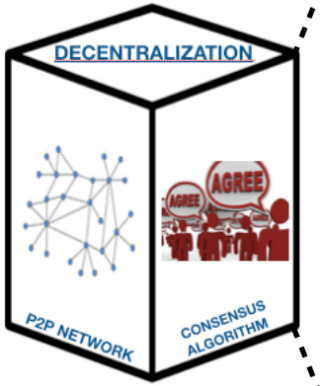
Co-funded by the Erasmus+ Programme of the European Union



Name	Goals	Advantages	Disadvantages	Domains	Implementations
Proof of work (PoW)	To provide a barrier to publishing blocks in the form of a computationally difficult puzzle to solve to enable transactions between untrusted participants.	Difficult to perform denial of service by flooding network with bad blocks. Open to anyone with hardware to solve the puzzle.	Computationally intensive (by design), power consumption, hardware arms race. Potential for 51 % attack by obtaining enough computational power.	Permissionless cryptocurrencies	Bitcoin, Ethereum, many more
Proof of stake (PoS)	To enable a less computationally intensive barrier to publishing blocks, but still enable transactions between untrusted participants.	Less computationally intensive than PoW. Open to anyone who wishes to stake cryptocurrencies. Stakeholders control the system.	Stakeholders control the system. Nothing to prevent formation of a pool of stakeholders to create a centralized power. Potential for 51 % attack by obtaining enough financial power.	Permissionless cryptocurrencies	Ethereum Casper, Krypton
Delegated PoS	To enable a more efficient consensus model through a 'liquid democracy' where participants vote (using cryptographically signed messages) to elect and revoke the rights of delegates to validate and secure the blockchain.	Elected delegates are economically incentivized to remain honest More computationally efficient than PoW	Less node diversity than PoW or pure PoS consensus implementations Greater security risk for node compromise due to constrained set of operating nodes As all delegates are 'known' there may an incentive for block producers to collude and accept bribes, compromising the security of the system	Permissionless cryptocurrencies Permissioned Systems	Bitshares, Steem, Cardano, EOS

Consensus Models

Co-funded by the Erasmus+ Programme of the European Union



Name	Goals	Advantages	Disadvantages	Domains	Implementations
Round Robin	Provide a system for publishing blocks amongst approved/trusted publishing nodes	Low computational power. Straightforward to understand.	Requires large amount of trust amongst publishing nodes.	Permissioned Systems	MultiChain
Proof of Authority/Identity	To create a centralized consensus process to minimize block creation and confirmation rate	Fast confirmation time Allows for dynamic block production rates Can be used in sidechains to blockchain networks which utilize another consensus model	Relies on the assumption that the current validating node has not been compromised Leads to centralized points of failure The reputation of a given node is subject to potential for high tail-risk as it could be compromised at any time.	Permissioned Systems, Hybrid (sidechain) Systems	Ethereum Kovan testnet, POA Chain, various permissioned systems using Parity
Proof of Elapsed Time (PoET)	To enable a more economic consensus model for blockchain networks, at the expense of deeper security guarantees associated with PoW.	Less computationally expensive than PoW	Hardware requirement to obtain time. Assumes the hardware clock used to derive time is not compromised Given speed-of-late latency limits, true time synchronicity is essentially impossible in distributed systems [13]	Permissioned Networks	Hyperledger Sawtooth
…	…	…	…	…	…

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Blockchain Architectures

Co-funded by the
Erasmus+ Programme
of the European Union

- **Permissionless blockchains** (e.g., Bitcoin)
  - Bitcoin
  - Ethereum
  - Zcash
- **Permissioned blockchains** (e.g., ScroogeCoin)
  - Hyperledger Fabric
  - Quorum

Permissionless Blockchains

Co-funded by the Erasmus+ Programme of the European Union

- **Characteristics**:
  - Participation **open** to the public
  - **Peer-to-peer** transactions
  - Typically tied to **cryptocurrency**
  - **Fully decentralized** (in principle…)
- **Challenges**:
  - **Privacy** and **scaling**

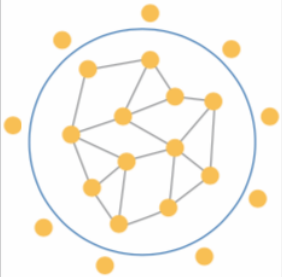
Permissionless blockchains are a disruptive technology that can dramatically change how we conduct business activities

Permissioned Blockchains

Co-funded by the Erasmus+ Programme of the European Union

- **Characteristics**:
  - Participation can be **private** and/or **controlled**
  - **Trusted** participants
  - More **efficient** than many public blockchains
  - Can support **privacy** and **confidentiality** in transaction
- **Challenges**:
  - Some level of **centralized trust** through governing authority

Permissioned blockchains may lead to cost-savings, workflow improvements, automation and improved auditing with current business processes

Blockchain type	Explanation	Example	Visualisation
Public permissionless blockchains	In these blockchain systems, everyone can participate in the blockchain's consensus mechanism. Also, everyone worldwide with an internet connection can transact and see the full transaction log.	Bitcoin, Litecoin, Ethereum	
Public permissioned blockchains	These blockchain systems allow everyone with an internet connection to transact and see the blockchain's transaction log, although only a restricted number of nodes can participate in the consensus mechanism.	Ripple, private versions of Ethereum	
Private permissioned blockchains	These blockchain systems restrict both the ability to transact and view the transaction log to only the participating nodes in the system, and the architect or owner of the blockchain system is able to determine who can participate in the blockchain system and which nodes can participate in the consensus mechanism.	Rubix, Hyperledger	
Private permissionless blockchains	These blockchain systems are restricted in who can transact and see the transaction log, although the consensus mechanism is open to anyone.	(Partially) Exonum	

Blockchain now and tomorrow: assessing multidimensional impacts of distributed ledger technologies

Agenda

Co-funded by the
Erasmus+ Programme
of the European Union

PART 1 CRYPTOGRAPHY & DATA STRUCTURES

- Cryptographic Hash Functions
- Hash Pointers and Data Structure (Blockchain)
- Digital Signatures

PART 2 LET'S DESIGN SOME (SIMPLE) DIGITAL CASH!

- GoofyCoin
- ScroogeCoin

PART 3 DECENTRALIZATION

- Distributed Consensus
- Consensus Models
- Blockchain Architectures

PART 4 BLOCKCHAIN LIMITATIONS AND MISCONCEPTIONS

Co-funded by the
Erasmus+ Programme
of the European Union

WHAT CAN I DO WITH BLOCKCHAIN?

Co-funded by the
Erasmus+ Programme
of the European Union

Demythologizing Blockchain…

- *Blockchain is secure…*
- *Blockchain is fast…*
- *Blockchain increases transparency…*
- *Blockchain is cheap…*
- *Blockchain is trustworthy…*
- *Blockchain is reliable…*
- …

It all depends on the specific implementation!

ENGINE

Co-funded by the
Erasmus+ Programme
of the European Union

BLOCKCHAIN
LIMITATIONS AND MISCONCEPTIONS

Immutability

Co-funded by the
Erasmus+ Programme
of the European Union

"…blockchain ledgers are immutable"

- **Not** strictly true
  - There are situations in which the blockchain can be modified…
- Example:
  - **51% attack** (permissionless blockchain networks)
  - attack can be mitigated in permissioned blockchain networks

Blockchain Governance

Co-funded by the Erasmus+ Programme of the European Union

"…no one controls a blockchain!"

- **Not** strictly true
- **Blockchain governance**: rules, practices and processes by which the blockchain network is directed and controlled
- Permissioned blockchains:
  - Networks are generally setup and run by an owner or consortium, which governs the blockchain network
- Permissionless blockchains:
  - Governed by blockchain network users, publishing nodes, and software developers
  - Each group has a level of control that affects the direction of the blockchain network's advancement

Blockchain Death

Co-funded by the
Erasmus+ Programme
of the European Union

- Traditional centralized systems are created and taken down constantly
  - Permissioned blockchain networks will likely not be different
- Permissionless blockchain networks are decentralized
  - There is a high chance that when a blockchain network “shuts down” it will never be fully shut down
  - There might always be some lingering blockchain nodes running

Co-funded by the
Erasmus+ Programme
of the European Union

"…blockchain is secure!"

- Makes no sense
- Blockchain is a component of a bigger system
  - Example from cryptocurrency:
    - bitcoin client, wallets, pools, …
    - storing bitcoins is a key management problem

If the whole blockchain-based system is not secure, also blockchain becomes not secure…

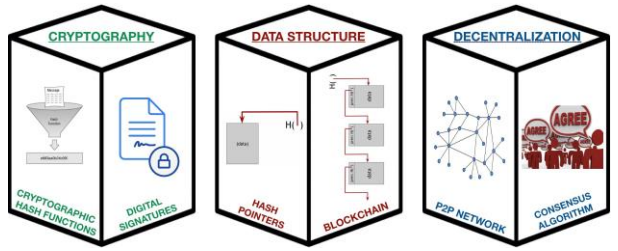
The Human Factor

Co-funded by the
Erasmus+ Programme
of the European Union

- Blockchain **cannot** enforce users to well-behave
- Particularly **true** in permissionless blockchain networks
  - They often provide a reward (e.g., bitcoin) to motivate users to act fairly
  - Some may choose to act maliciously if that provides greater rewards
- **True** also for permissioned blockchain networks
  - Administrators of the infrastructure for networks may also act maliciously
  - Example: administrators may be able to take over block production, exclude certain users from performing transactions, rewrite block history, double spend coin, delete resources, or re-route or block network connections

Key Points

Co-funded by the
Erasmus+ Programme
of the European Union



- Blockchain is **not magic**
- Blockchain is a **data structure**
- Blockchain-based systems (a.k.a. “blockchain technology”) are **specific implementations** of blockchain
- By combining different technologies, you can have **significantly different blockchain-based systems**
  - … satisfying **very different properties!**

Thank you!

PROF. NICOLA DRAGONI, PHD
Embedded Systems Engineering (ESE) Section
DTU Compute
Technical University of Denmark

ndra@dtu.dk

Further Reading

Co-funded by the Erasmus+ Programme of the European Union

- Website blockchain.info
- **Blockchain Technology Overview** - NISTIR 8202

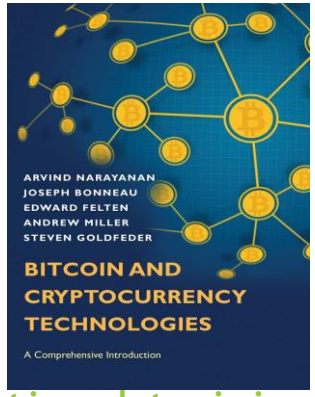
https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

- **Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction**

A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder

Princeton University Press, 2016

Warning: 300+ pages



- **Bitcoin: A Peer-to-Peer Electronic Cash System**

Satoshi Nakamoto

https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf

- **Paxos Made Simple**

Leslie Lamport

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/paxos-simple-Copy.pdf

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

Answer the following questions by marking the correct boxes.

N.B. More than one answer (== box) can be correct for a single question.

Exercise*

Co-funded by the
Erasmus+ Programme
of the European Union

What does blockchain technology may refer to?

- Cheap technology for modern IT systems
- Data structure
- Sustainable technology for modern IT systems
- Cryptography
- Decentralization

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

What property should a cryptographically secure hash function have?

- Collision resistance
- Preimage resistance
- Second collision resistance
- Second preimage resistance

Exercise*

Co-funded by the
Erasmus+ Programme
of the European Union

What is a hash pointer?

- A pointer to where some information is stored
- A cryptographic hash of the information
- Can be used to verify that some info hasn't changed
- Can be used to verify who made a transaction

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

In blockchain, blocks are linked

- Forward to next block
- Backward to the previous block
- Not linked with each other

Exercise*

Co-funded by the
Erasmus+ Programme
of the European Union

Which of these keys are required for verifying a signature?

- The secret key
- The public key
- Both the secret and the public key
- None. Keys are required only for signing; anyone can verify the signature without a key

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

If you generate numerous identities (public keys) for yourself and interact online using those different identities:

- It is essential to have a good source of randomness. Otherwise, adversaries might be able to deduce your secret key and take control of your identities
- Adversaries may be able to link your identities because public keys generated on the same computer tend to look similar
- Adversaries may be able to de-anonymize you by analyzing your activity patterns

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

Alice and Bob use ScroogeCoin. Alice owns ten coins, each under a different address (public key) and each of value 3.0. She would like to transfer coins of value 5.0 to Bob. Recall that the PayCoins transaction consumes (and destroys) some coins and creates new coins of the same total value. Alice's transfer will require, at a minimum:

- One PayCoins transaction, one new coin created, and one signature
- One PayCoins transaction, two new coins created, and two signatures
- Two PayCoins transactions, two new coins created, and four signatures
- Two PayCoins transactions, one new coin created, and two signatures

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

Which of these factors make distributed consensus hard?

- Nodes may crash
- Nodes may be taken over by malware
- Encrypted messages may be intercepted and decrypted
- There is latency on the network

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

Why is Bitcoin able to reach consensus in practice despite this being a generally difficult problem?

- Financial incentives cause participants to work together
- Only small groups of nodes have to reach consensus rather than the network having to globally reach consensus
- The order of blocks doesn’t matter for consensus
- Consensus only has to be reached over long-time scales

ENGINE

Exercise*

Co-funded by the
Erasmus+ Programme
of the European Union

What can a malicious node in a Bitcoin network do?

- Create valid transactions originating from someone else's address
- Prevent a valid transaction from getting any confirmations
- Ignore the longest valid branch rule when proposing a new block

Exercise

Co-funded by the
Erasmus+ Programme
of the European Union

A 51% attacker can potentially

- Steal coins from an existing address
- Make it unprofitable for other miners to mine
- Change the block reward
- Suppress transactions from the block chain

Exercise*

Co-funded by the
Erasmus+ Programme
of the European Union

Bitcoin is based on _____ blockchain.

- Private Permissioned
- Private Permissionless
- Public Permissioned
- Public Permissionless