Law, Science and Technology
MSCA ITN EJD n. 814177

Mirko Zichichi

Smart contracts and Proof of Location in Smart Cities

Outline

- Preliminaries
  - Authentication
- Introduction to Distributed Systems
  - Distributed systems
  - Blockchain
- Smart Contracts
  - Decentralized computing
- Proof of Location in Smart Cities
  - FOAM
  - Zero Knowledge PoL

RIoE
Rights of Internet of Everything

Some preliminary notions

- Hash functions
- Digital signature
- Timestamp

RIoE
Rights of Internet of Everything

Cryptographic **Hash** function

$$f(x) = y$$

Cryptographic **Hash** function

$$f(x) = y$$

One-way

Practically infeasible to invert:

$$f^{-1}(y) = ?$$

Cryptographic **Hash** function

$$f(x) = y$$

$$f(string) = digest$$

One-way

Practically infeasible to invert:

$$f^{-1}(y) = ?$$

Cryptographic **Hash** function

$$f(x) = y$$

$$f(string) = digest$$

$$SHA256(moat) = 98E2...16F3$$

SHA256

admitted by **eIDAS** regulation

Cryptographic **Hash** function

$$f(x) = y$$

$$f(string) = digest$$

$$SHA256(moat) = 98E2...16F3$$

$$SHA256(maot) = E671...A9C9$$

SHA256

admitted by **eIDAS** regulation

Cryptographic **Hash** function

$$f(x) = y$$

$$f(string) = digest$$

$$SHA256(moat) = 98E2...16F3$$

$$SHA256(maot) = E671...A9C9$$

digest length is always the same
(256 bit in the case of SHA256)

SHA256

admitted by
eIDAS
regulation

Cryptographic **Hash** function

$$f(x) = y$$

$$f(string) = digest$$

$$SHA256(moat) = 98E2...16F3$$

$$SHA256(maot) = E671...A9C9$$

digest length is always the same
(256 bit in the case of SHA256)

Collision free : the probability of two inputs with the same digest is very small

SHA256

admitted by **eIDAS** regulation

Digital Signature

Scheme for verifying the **authenticity** of digital messages (documents).

Digital Signature

Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

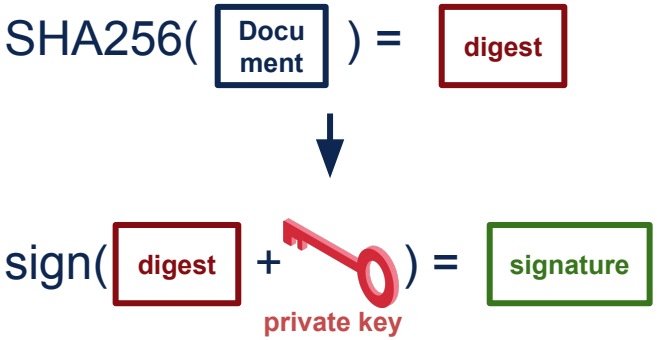
Digital Signature

Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

Digital Signature

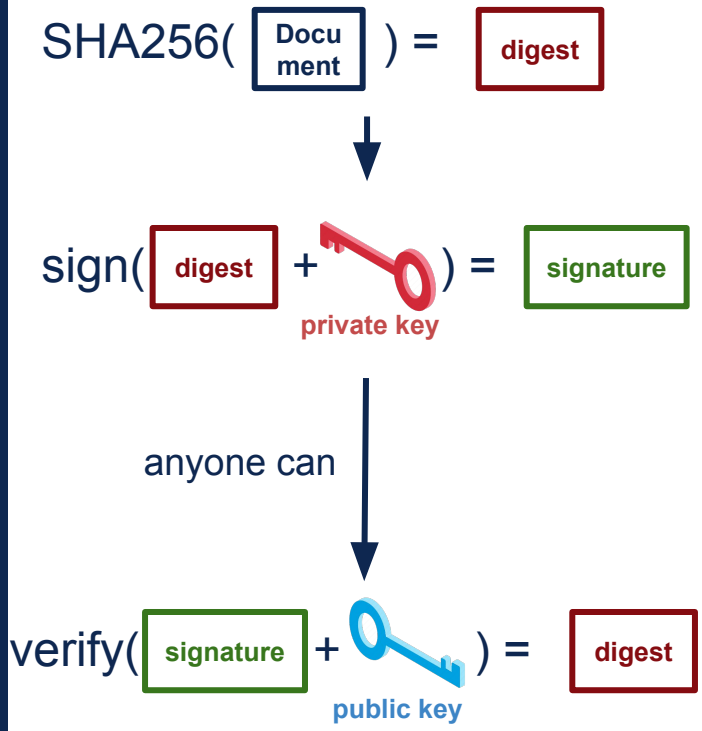


Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

Digital Signature

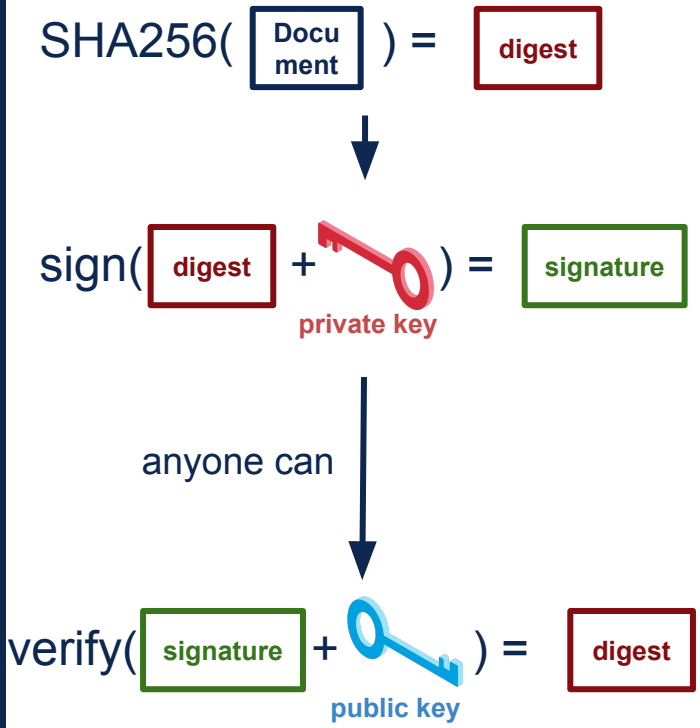


Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

Digital Signature



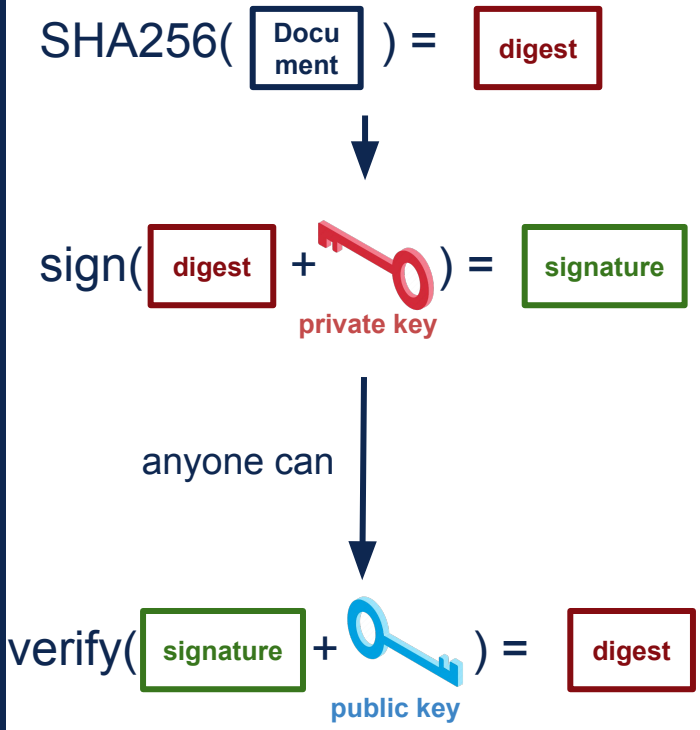
Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

Integrity: grants that the message was not altered in transit (digest)

Digital Signature



Scheme for verifying the **authenticity** of digital messages (documents).

Employs **Asymmetric Cryptography:**

- public key (associated to a pub certificate)
- private key

Integrity: grants that the message was not altered in transit (digest)
Authentication: A valid digital signature gives a recipient very strong reason to believe that the message was created by a known sender.
When can you associate the pub key to an identity (e.g. the sender)?

eIDAS recognizes 3 e-signature types

Electronic signatures

eIDAS sets a foundation for all electronic signatures by asserting that no signature can be denied legal admissibility solely because it's in electronic form e.g.:
Signing an e-mail with your personal name or entering a PIN code

eIDAS recognizes 3 e-signature types

Electronic signatures	Advanced Electronic Signatures (AdES)
eIDAS sets a foundation for all electronic signatures by asserting that no signature can be denied legal admissibility solely because it's in electronic form e.g.: Signing an e-mail with your personal name or entering a PIN code	With AdES, signatures must be uniquely linked to, and capable of identifying, the signer. Signers create their signature using data solely under their control and the final document is tamper-evident. ←**Digital Signatures** (XAdES, PAdES, CAdES, Associated Signature Container Baseline Profile without qualified Certificate, Graphometric signature, biometric signature, etc.)

eIDAS recognizes 3 e-signature types

Electronic signatures	Advanced Electronic Signatures (AdES)	Qualified Electronic Signatures (QES)
eIDAS sets a foundation for all electronic signatures by asserting that no signature can be denied legal admissibility solely because it's in electronic form e.g.: Signing an e-mail with your personal name or entering a PIN code	With AdES, signatures must be uniquely linked to, and capable of identifying, the signer. Signers create their signature using data solely under their control and the final document is tamper-evident. ⟵**Digital Signatures** (XAdES, PAdES, CAdES, Associated Signature Container Baseline Profile without qualified Certificate, Graphometric signature, biometric signature, etc.)	QES is a stricter form of AdES. Same legal value as handwritten signatures. It requires signers to use certificate-based digital ID issued by a qualified **EU Trust Service Provider (TSP)**, along with a **qualified signature creation device (QSCD)** e.g.: XAdES, PAdES, CAdES with Qualified Certificate and secure device: smart card, USB token, or mobile with a one-time passcode

Electronic Time Stamp eIDAS

Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time

Electronic Time Stamp eIDAS

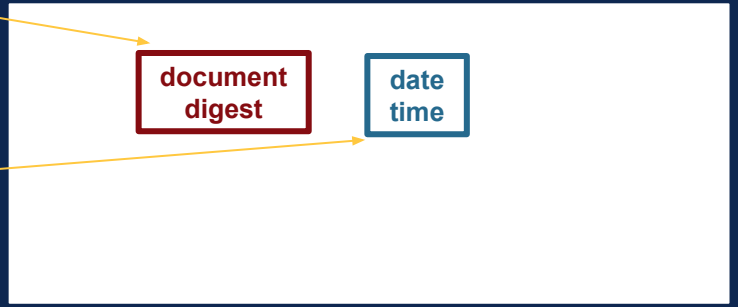
Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time

- links the date and time with the data so that the possibility of modifying the data without being detected is reasonably eliminated

Electronic Time Stamp eIDAS

Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time

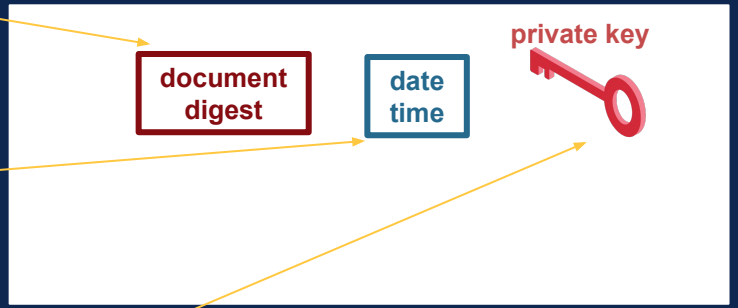
- links the date and time with the data so that the possibility of modifying the data without being detected is reasonably eliminated
- is based on a temporary information source linked to Coordinated Universal Time



Electronic Time Stamp eIDAS

Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time

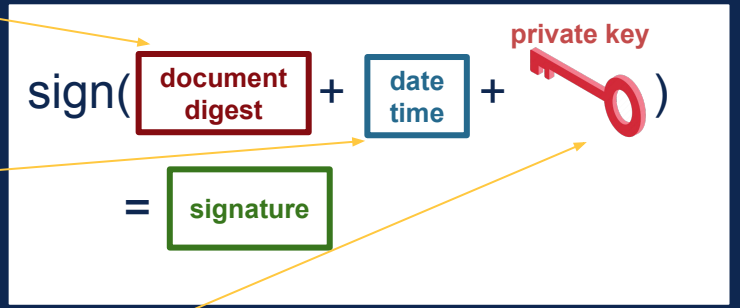
- links the date and time with the data so that the possibility of modifying the data without being detected is reasonably eliminated
- is based on a temporary information source linked to Coordinated Universal Time
- has been signed using an **AdES** or stamped with an advanced electronic stamp of the **TSP** or by any equivalent method



Electronic Time Stamp eIDAS

Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time

- links the date and time with the data so that the possibility of modifying the data without being detected is reasonably eliminated
- is based on a temporary information source linked to Coordinated Universal Time
- has been signed using an **AdES** or stamped with an advanced electronic stamp of the **TSP** or by any equivalent method



Legal validity of the qualified electronic time stamp

- Art. 41 of eIDAS
- A qualified electronic time stamp shall enjoy the presumption of the accuracy of the date and the time it indicates and the integrity of the data to which the date and time are bound.

In Italy

- The digital document signed with qualified digital signature(s) should be closed by a qualified time stamp in order to fix the date when the qualified certificate(s) of the digital signature(s) is/are valid.
- Over the time the qualified certificates expire and it is not possible to validate the validity of the certificates in the moment of the signature.

Digitally signed and time stamped Document

Digitally signed and time stamped Document

SHA256(Docu ment) = digest

sign(digest +) = signature

Alice Priv

Docu ment | signature

Alice's Certificate

Digitally signed and time stamped Document

SHA256(Docu ment) = digest

↓

sign(digest + ) = signature

Alice Priv

↓

SHA256(Docu ment signature) = digest 2

Alice's Certificate

Digitally signed and time stamped Document

SHA256(Docu ment) = digest

sign(digest + Alice Priv) = signature

SHA256(Docu ment | signature) = digest 2

Alice's Certificate

\+

date time

Request signature to TSP

Digitally signed and time stamped Document

SHA256(Docu ment) = digest

sign(digest + Alice Priv) = signature

SHA256(Docu ment + signature) = digest 2 + date time

Alice's Certificate

sign(digest 2 + date time + TSP Priv) = signature 2

Request signature to TSP

Digitally signed and time stamped Document

SHA256(Docu ment) = digest

sign(digest + Alice Priv) = signature

SHA256(Docu ment + signature) = digest 2 + date time

Alice's Certificate

sign(digest 2 + date time + TSP Priv) = signature 2

Request signature to TSP

Docu ment	signature	Alice's Certificate
date time	signature 2	TSP's Certificate

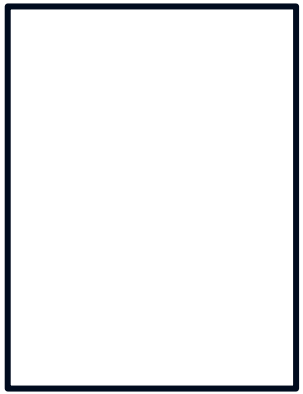
Introduction to Distributed Systems

and Blockchains

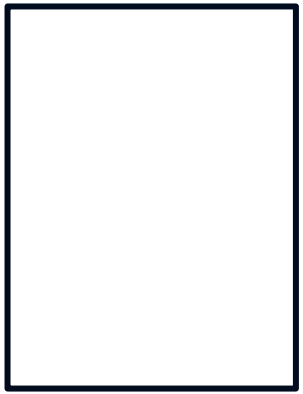
Client/Server Architecture

A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture



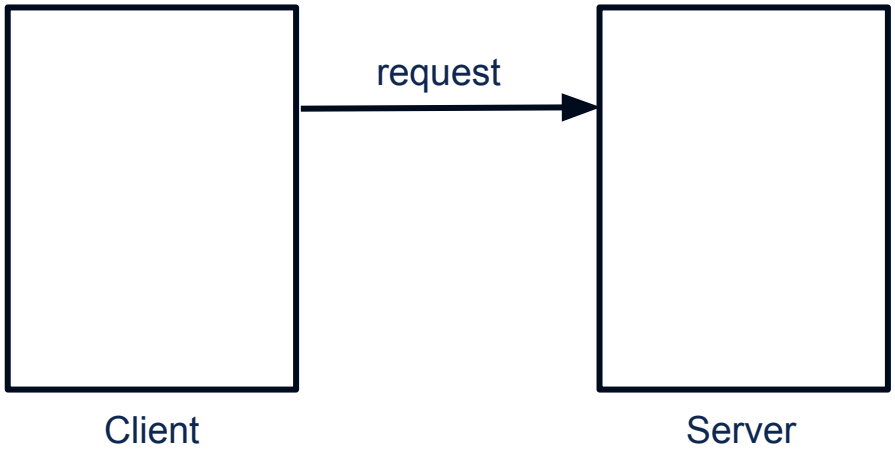
Client



Server

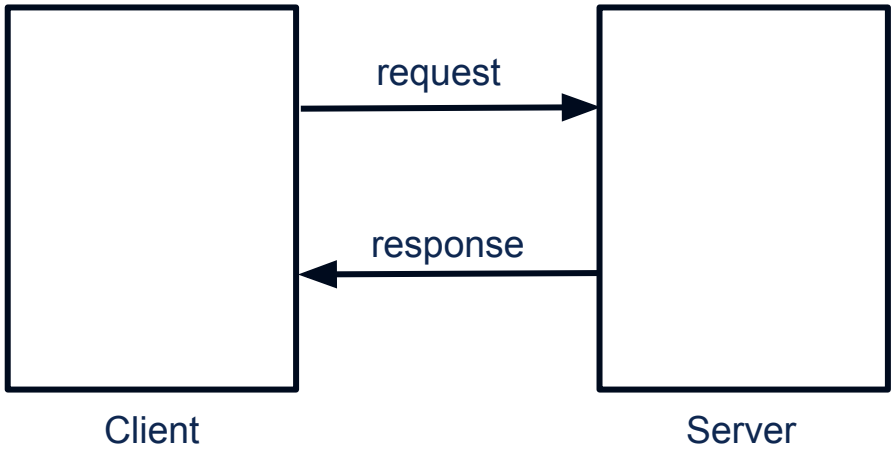
A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture



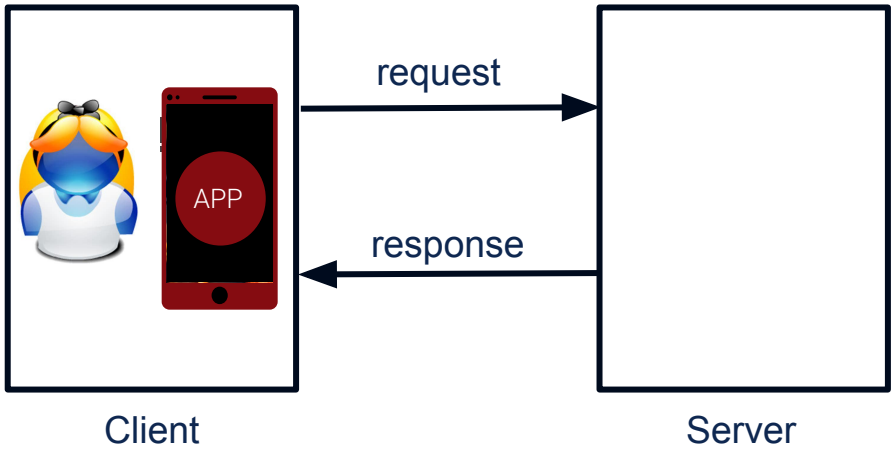
A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture



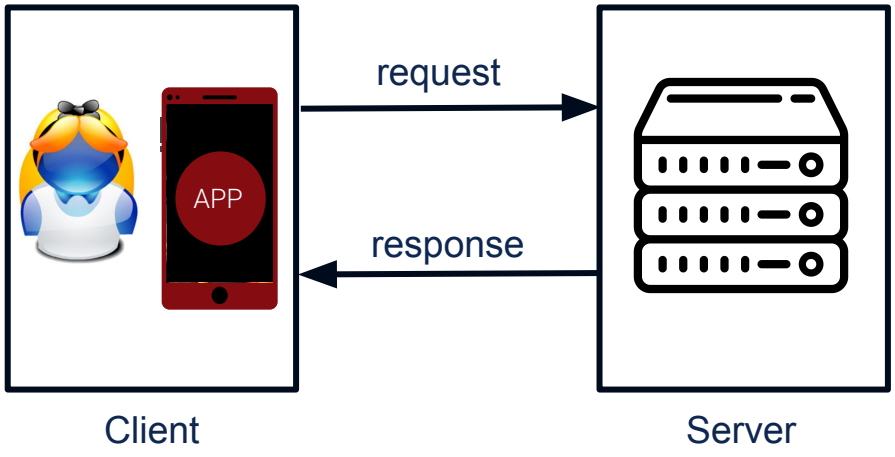
A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture



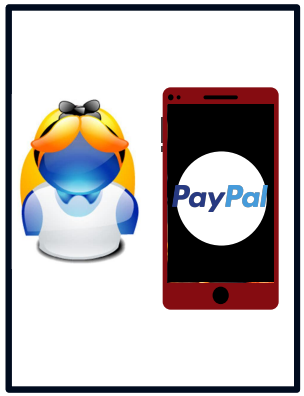
A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture

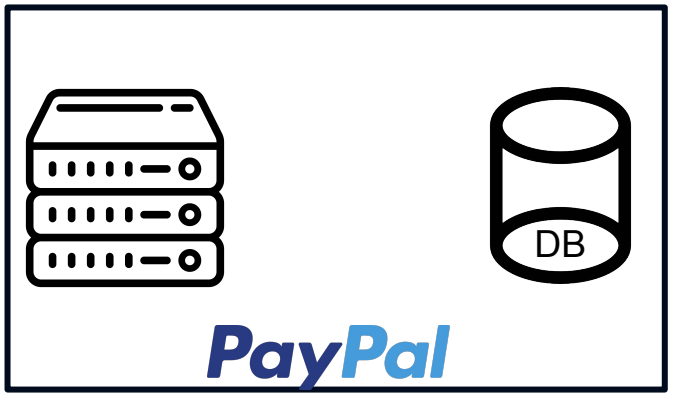


A system architecture is the conceptual **model** that defines the **structure**, **behavior**, and more **views** of a system

Client/Server Architecture Example



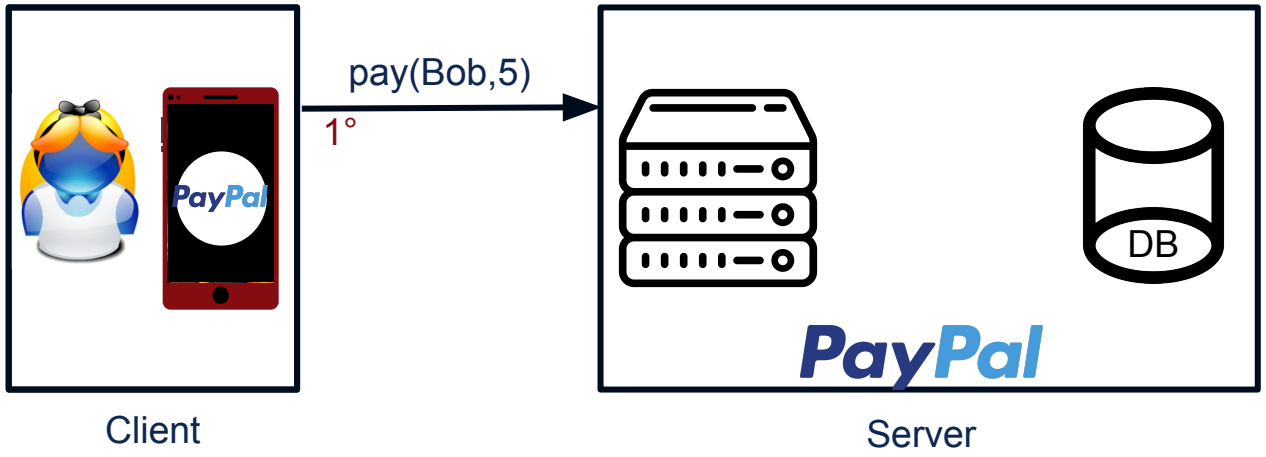
Client



Server

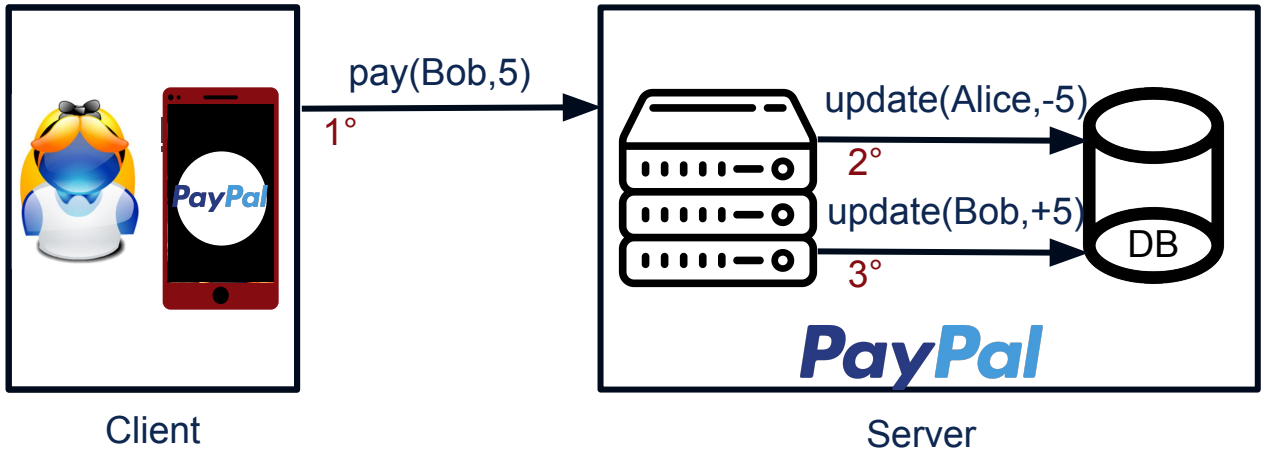
Alice
pays
Bob
5 euros

Client/Server Architecture Example



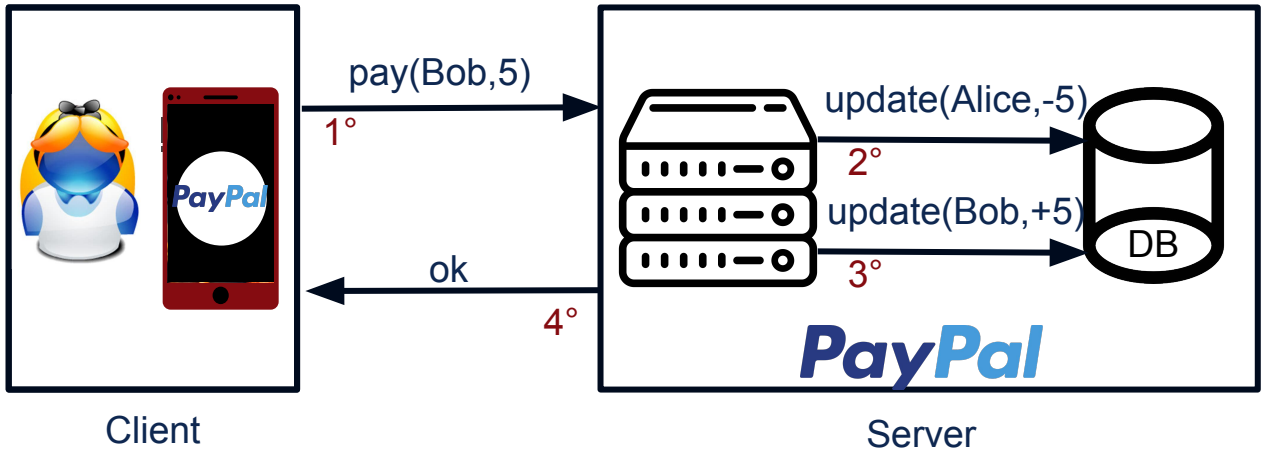
Alice
pays
Bob
5 euros

Client/Server Architecture Example



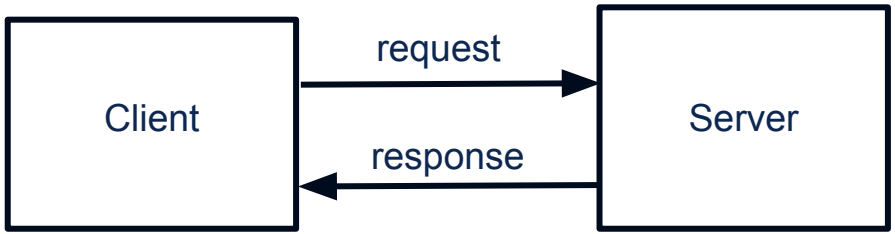
Alice
pays
Bob
5 euros

Client/Server Architecture Example

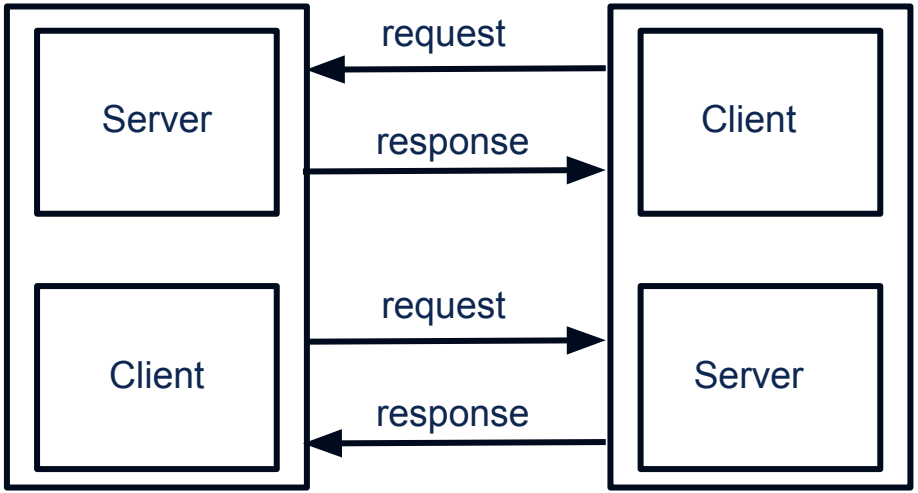


Alice
pays
Bob
5 euros

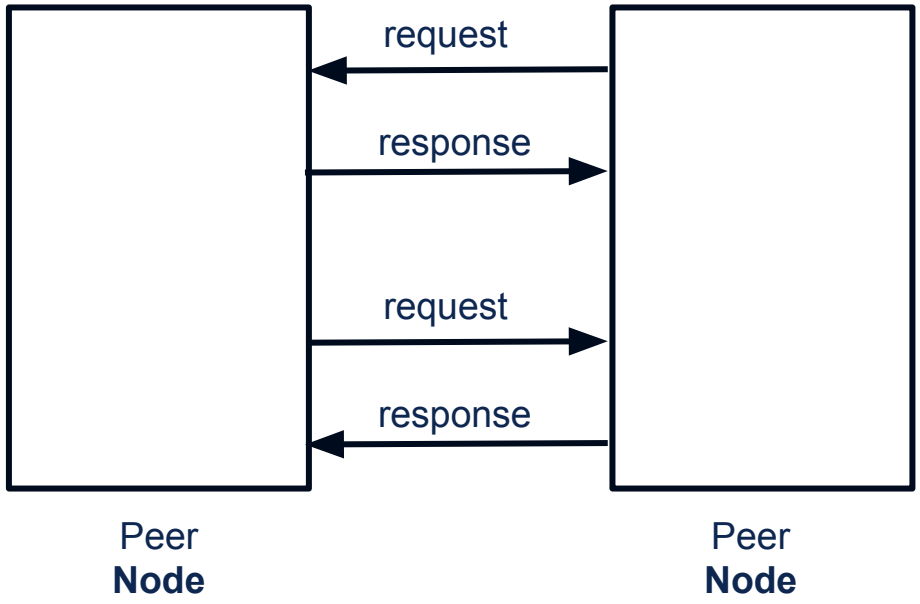
Client/Server Architecture



Peers are
Client and Server
simultaneously

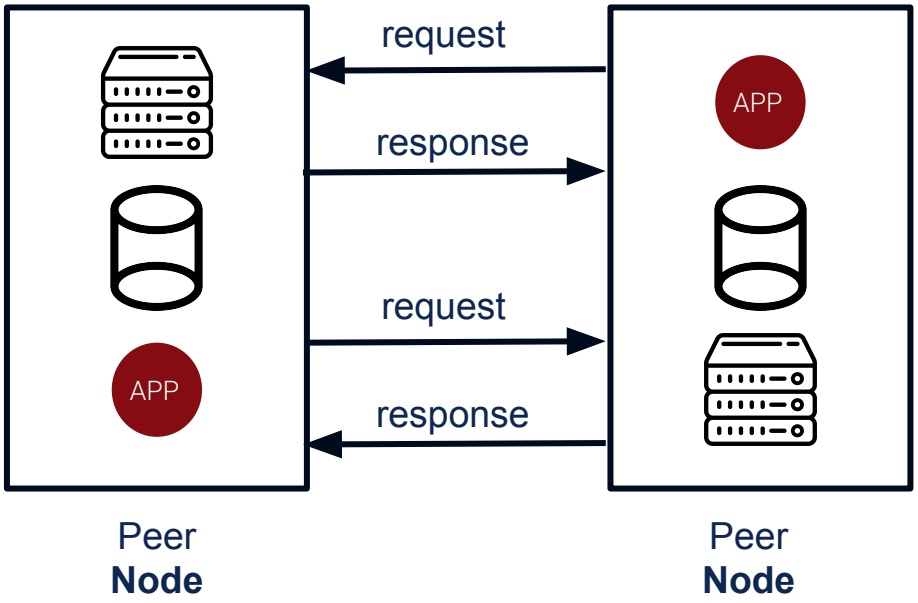


Peer to Peer (P2P) Architecture



Peers are
Client and Server
simultaneously

Peer to Peer (P2P) Architecture



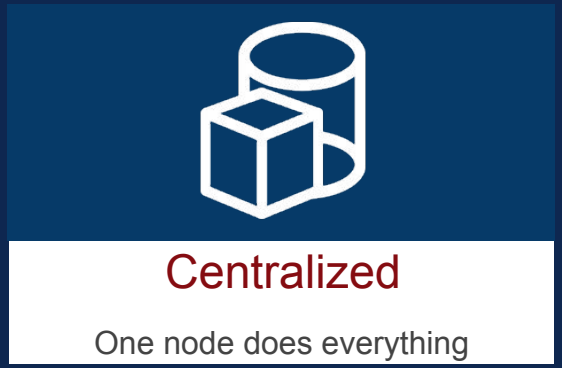
Peers are
Client and Server
simultaneously

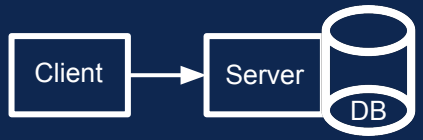
Distributed Systems

Systems that don’t share memory (or clock) but connect and relay information over a communication medium. The different nodes in distributed system have their own memory, OS and local resources.

Distributed Systems

Systems that don’t share memory (or clock) but connect and relay information over a communication medium. The different nodes in distributed system have their own memory, OS and local resources.



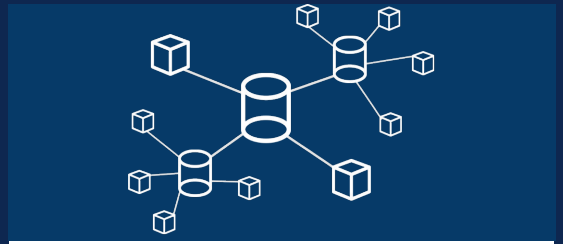


Distributed Systems

Systems that don’t share memory (or clock) but connect and relay information over a communication medium. The different nodes in distributed system have their own memory, OS and local resources.

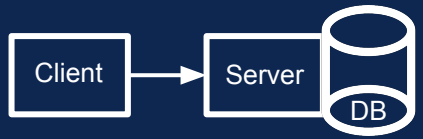
Centralized

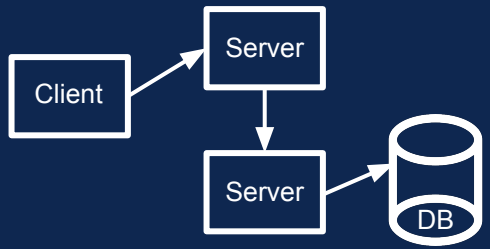
One node does everything



Semi-centralized

A node distribute work to sub-nodes



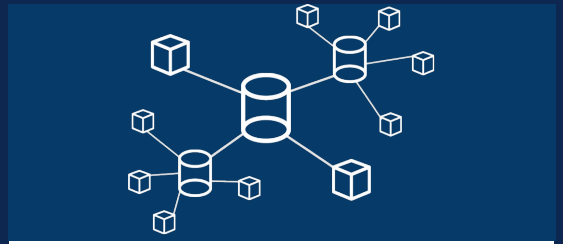


Distributed Systems

Systems that don’t share memory (or clock) but connect and relay information over a communication medium. The different nodes in distributed system have their own memory, OS and local resources.

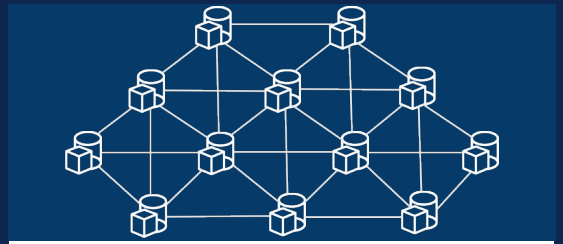
Centralized

One node does everything

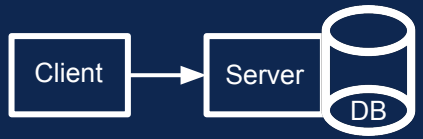


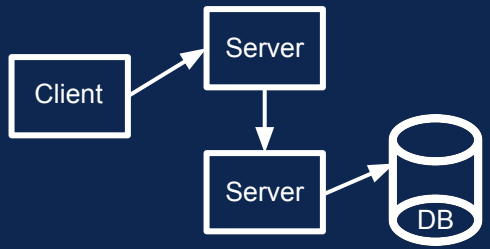
Semi-centralized

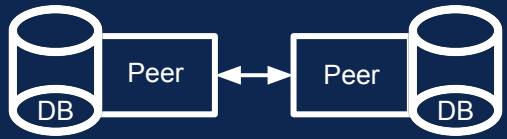
A node distribute work to sub-nodes



Distributed system Non-centralized (P2P)

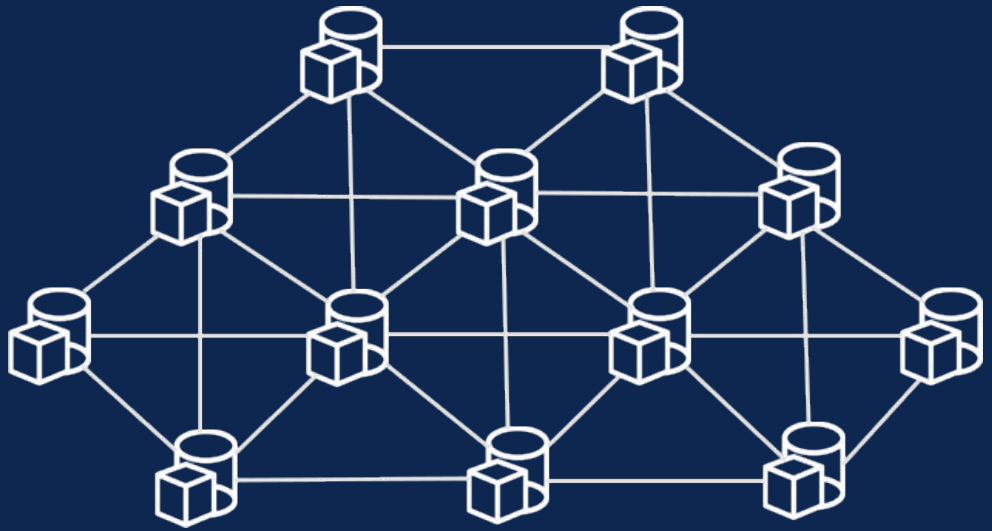






BLOCKCHAIN

A Distributed System based on a P2P network



Blockchain

- It is a technology which is part of the realm of the DLT: **Distributed Ledger Technologies**

Blockchain

- It is a technology which is part of the realm of the DLT: **Distributed Ledger Technologies**
- A ledger is distributed among nodes in a network, that update their local copy following a unique consensus mechanism

Blockchain

- It is a technology which is part of the realm of the DLT: **Distributed Ledger Technologies**
- A ledger is distributed among nodes in a network, that update their local copy following a unique consensus mechanism
- A blockchain is a DLT where the ledger takes the form of a set of block (relatively) chronologically ordered

Blockchain

we can distinguish between

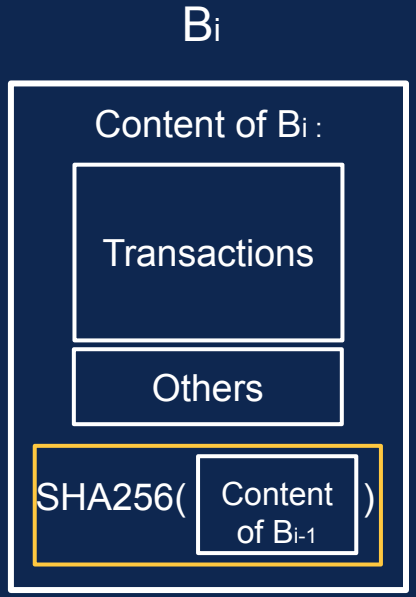
Blockchain

we can distinguish between

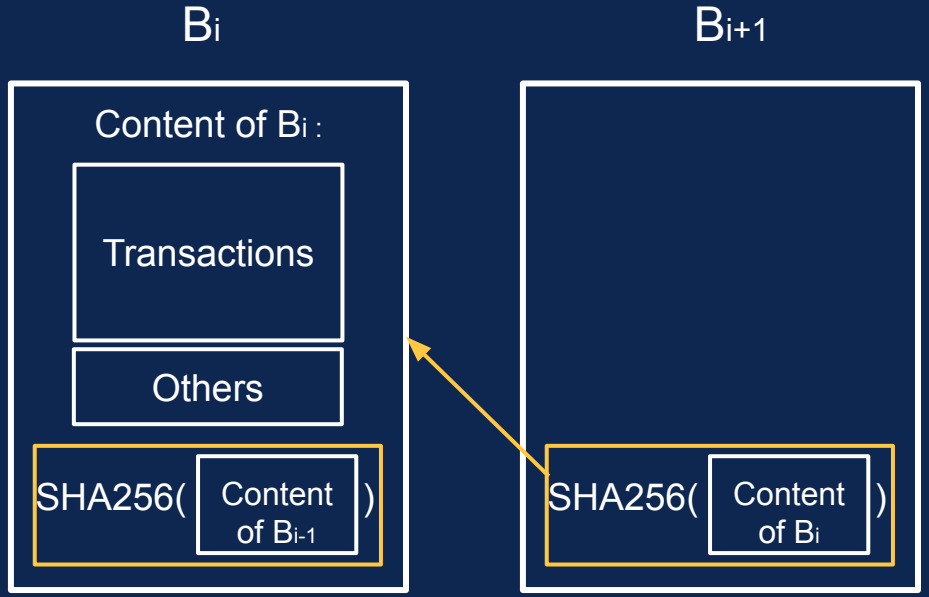
- How the ledger is structured: **chain of blocks**

Blocks

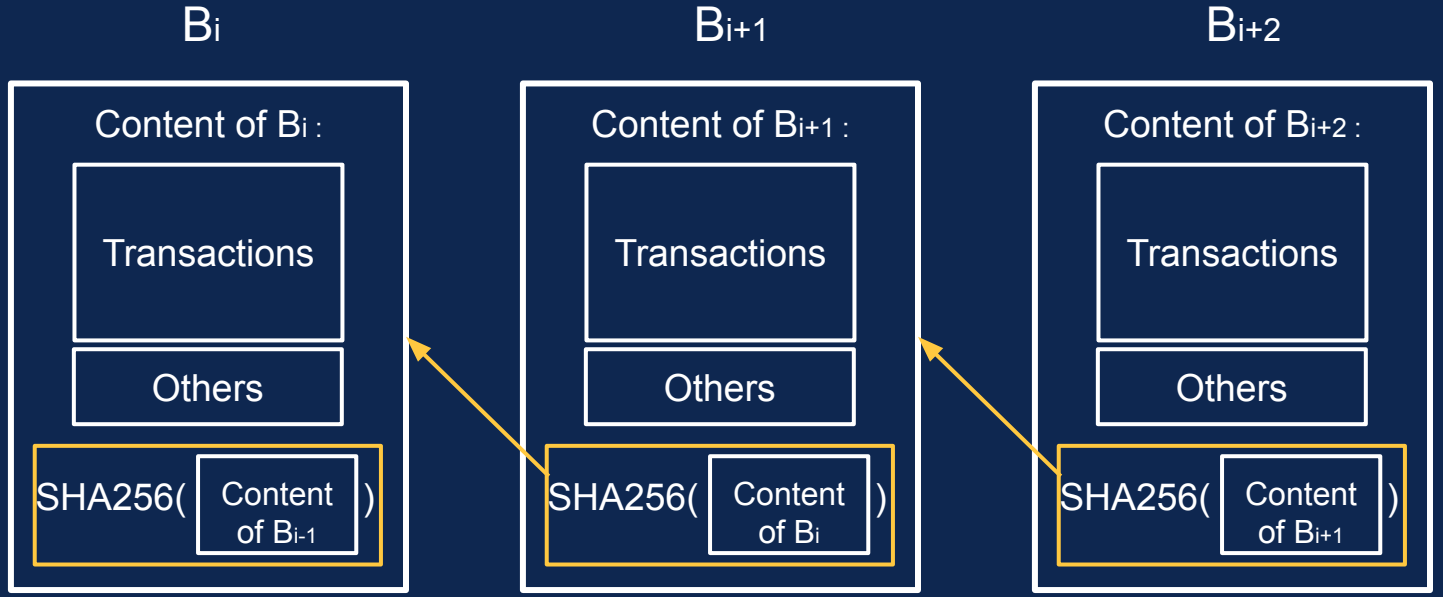
Blocks



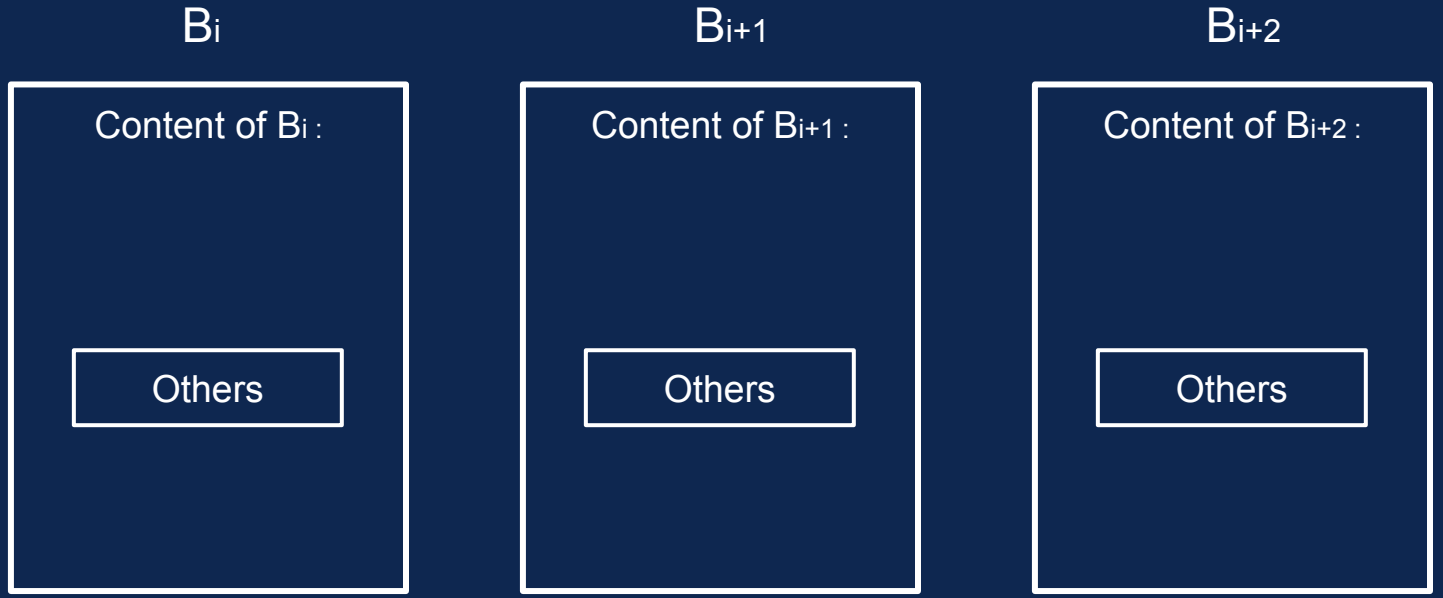
Blocks


B_i
B_{i+1}
Content of B_i :
Transactions
Others
SHA256(Content of B_{i-1})
SHA256(Content of B_i)

Blocks


B_i
Content of B_i :
Transactions
Others
SHA256(Content of B_{i-1})
B_{i+1}
Content of B_{i+1} :
Transactions
Others
SHA256(Content of B_i)
B_{i+2}
Content of B_{i+2} :
Transactions
Others
SHA256(Content of B_{i+1})

Blocks

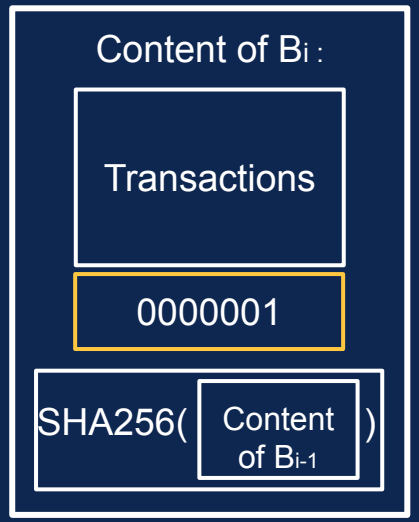

B_i
Content of B_i :
Others
B_{i+1}
Content of B_{i+1} :
Others
B_{i+2}
Content of B_{i+2} :
Others

What is Mining?

To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'

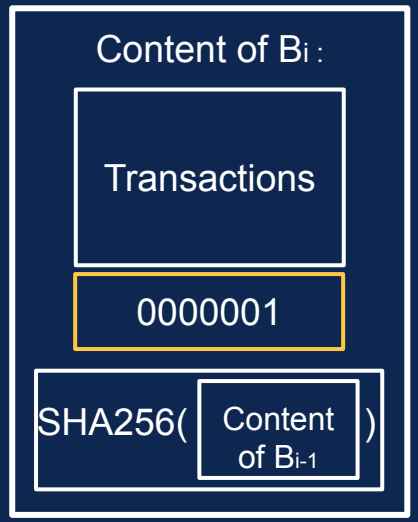
What is Mining?

To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'

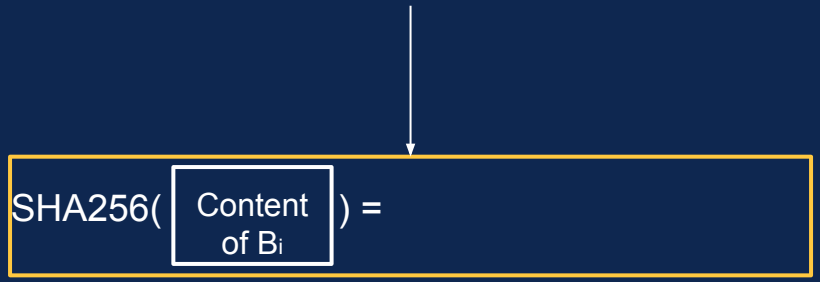


What is Mining?

To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'

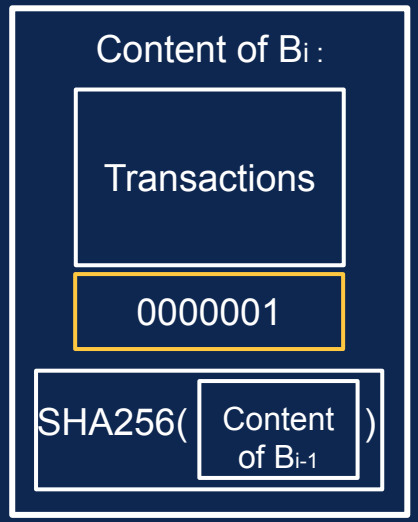


Puzzle:
this digest must start with
9 zeros

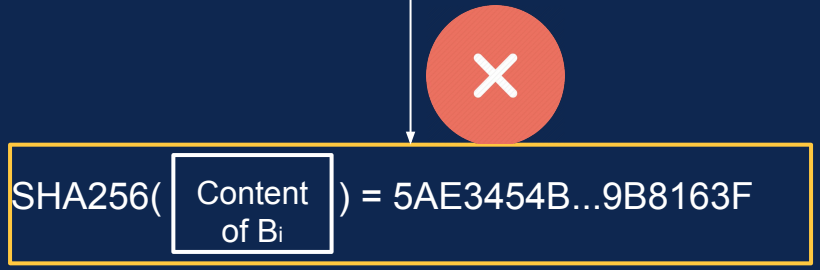


What is Mining?

To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'

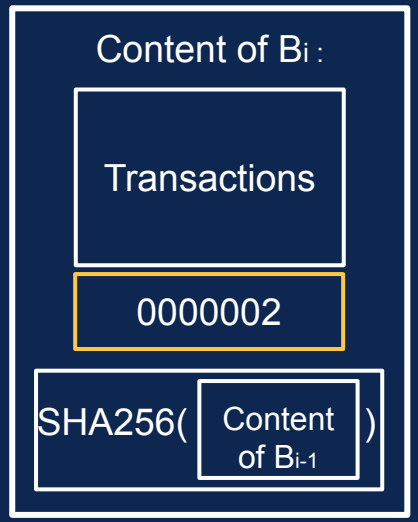


Puzzle:
this digest must start with
9 zeros

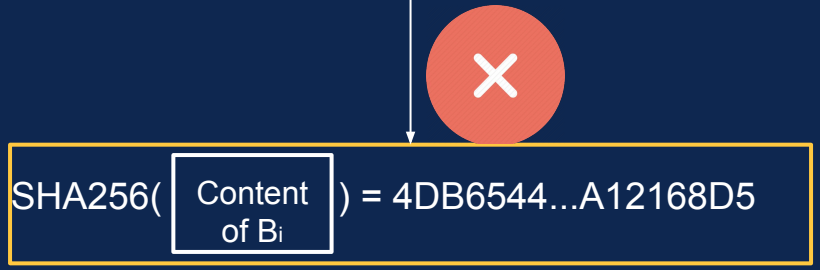


What is Mining?

To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'

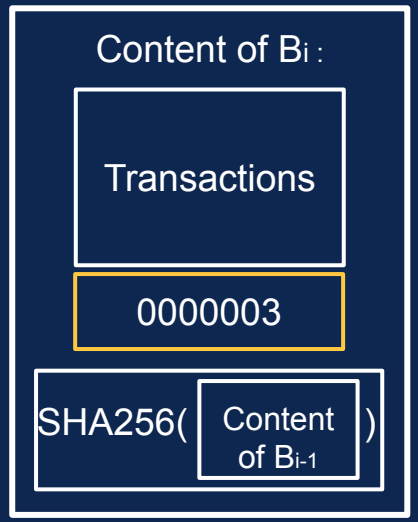


Puzzle:
this digest must start with
9 zeros

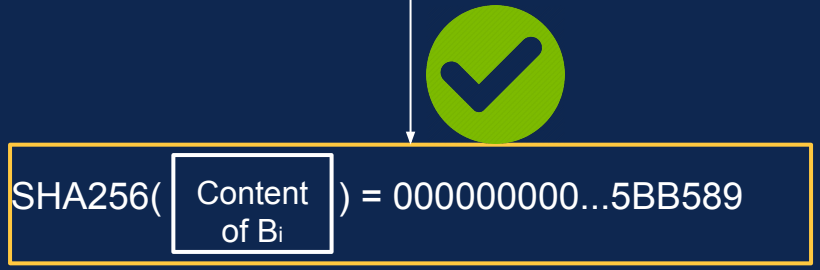


What is Mining?

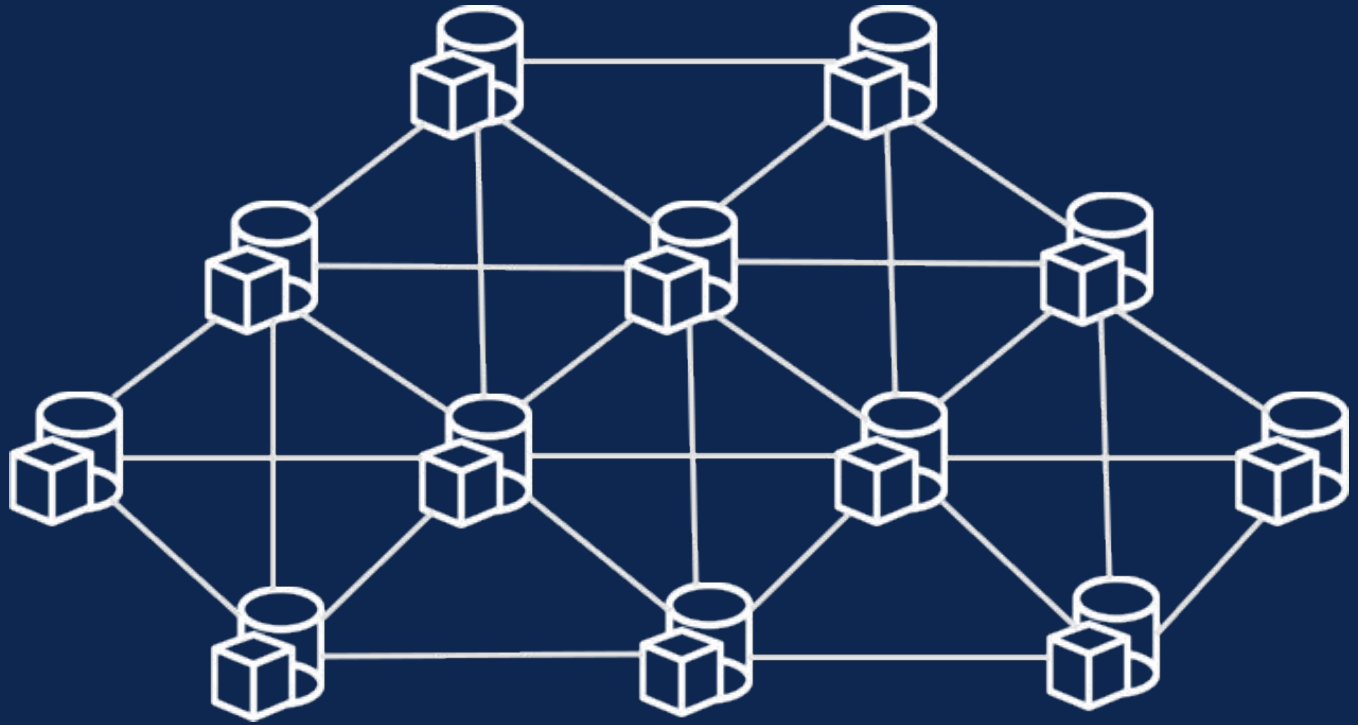
To solve a cryptographical puzzle, e.g. finding the 'needle in a haystack'



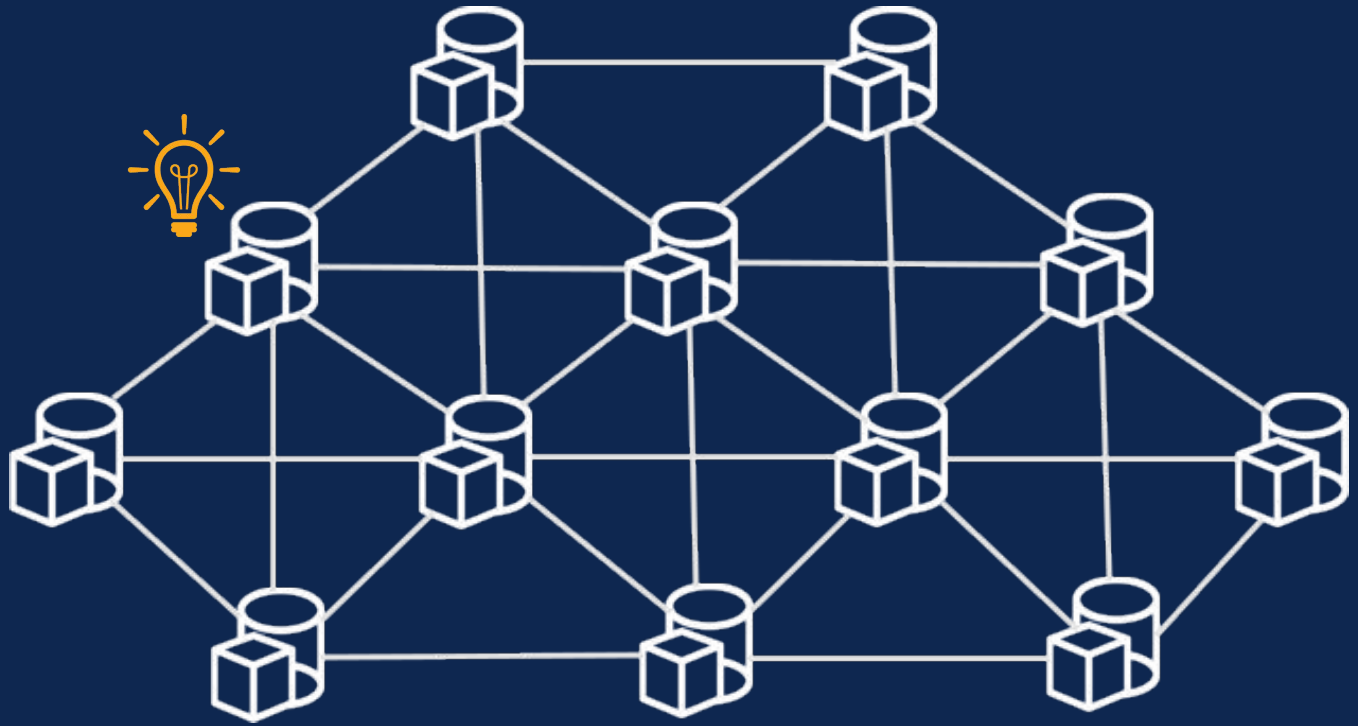
Puzzle:
this digest must start with
9 zeros



Consensus: Proof of Work and propagation

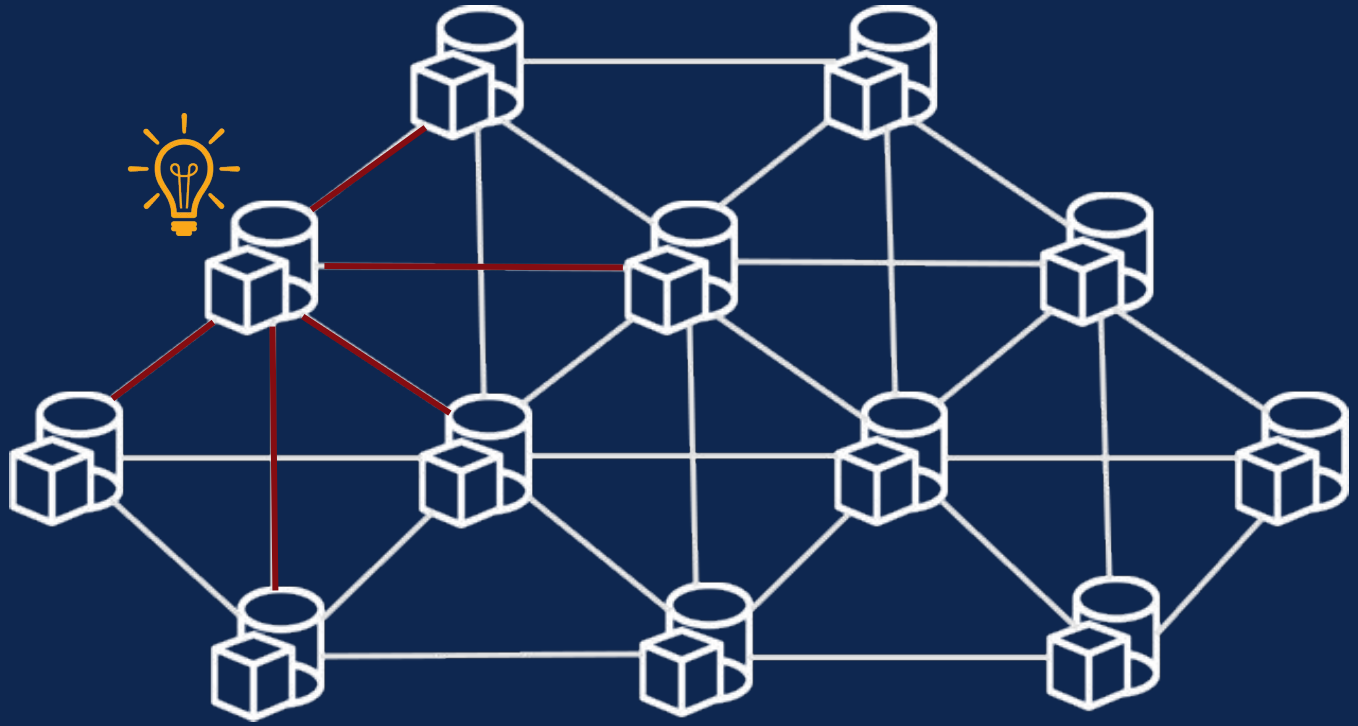


Consensus: Proof of Work and propagation



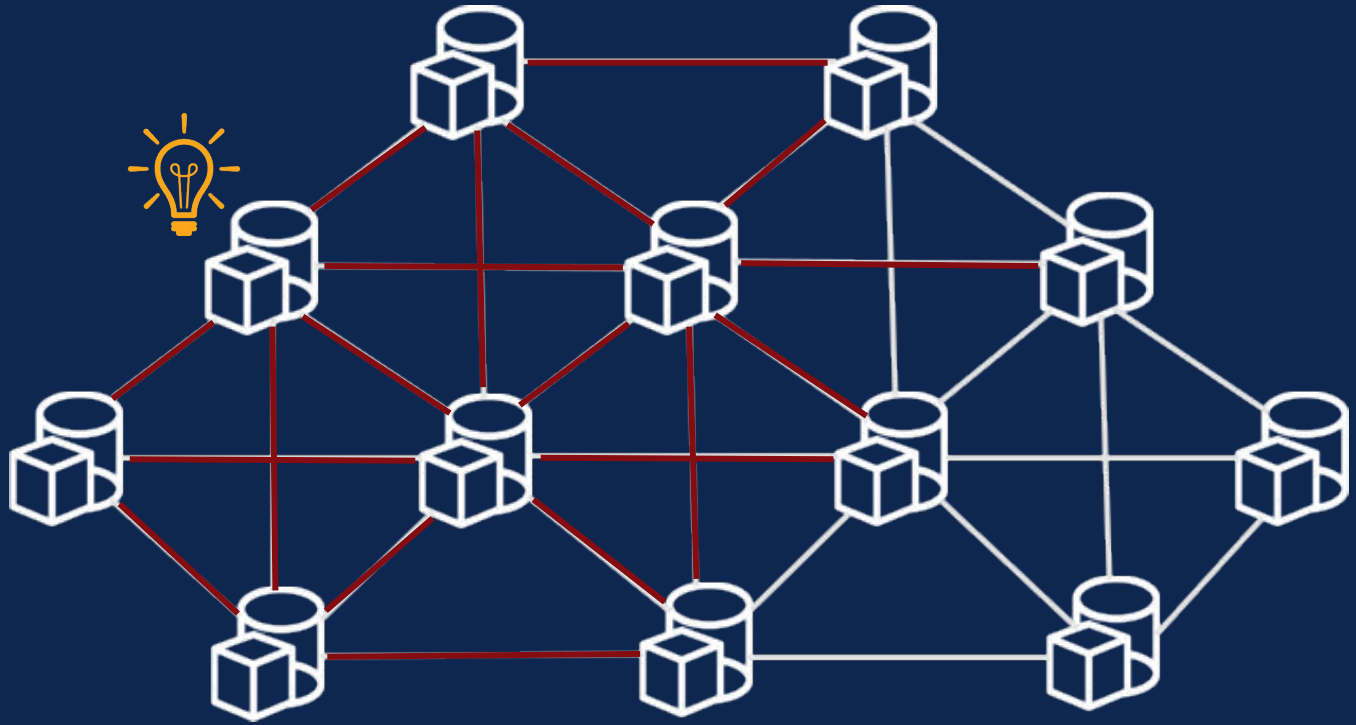
1. Node solves PoW

Consensus: Proof of Work and propagation



1. Node solves PoW
2. Broadcasts block to neighbors

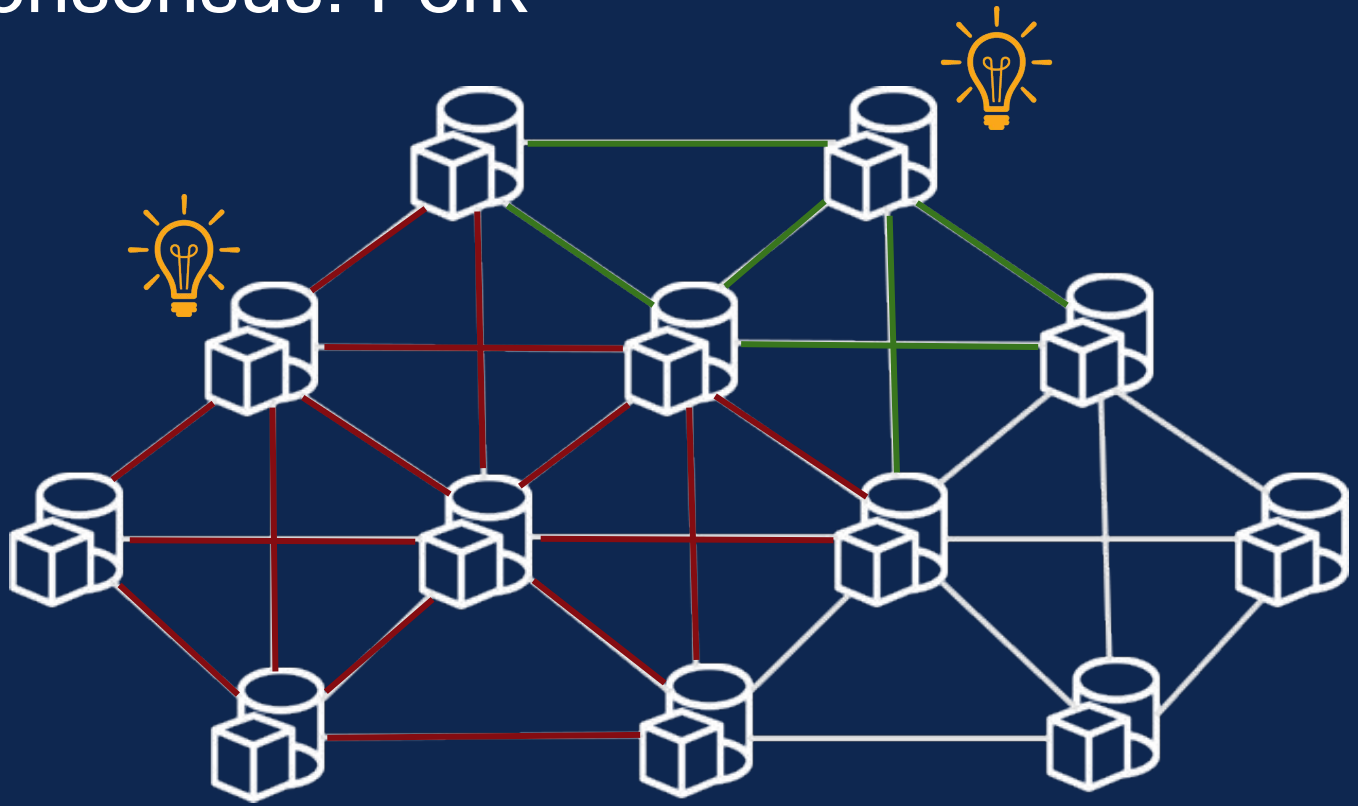
Consensus: Proof of Work and propagation



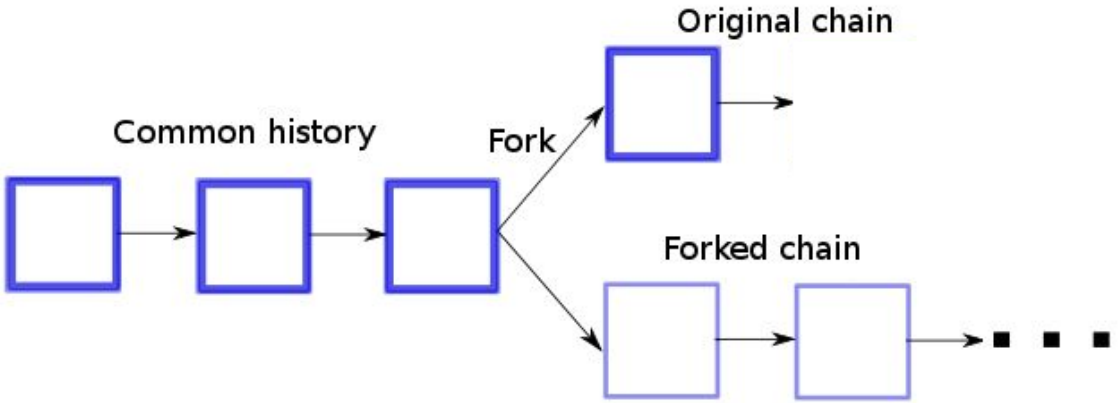
1. Node solves PoW
2. Broadcasts block to neighbors
3. Nodes check that:
- PoW is valid
- Block content is valid

then they broadcast to their neighbors

Consensus: Fork



Consensus: Fork



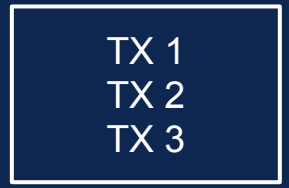
Blockchain

we can distinguish between

- How the ledger is structured: **chain of blocks**
- What to write in the ledger: **transactions**

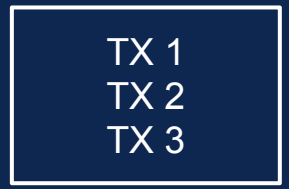
Transactions

- If the ledger maintains the state of the system, a transaction is the operation that alters this state



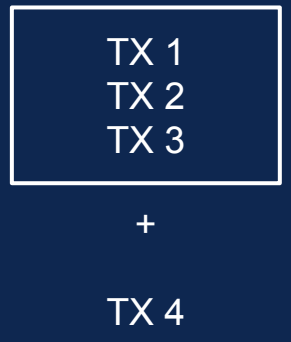
Transactions

- If the ledger maintains the state of the system, a transaction is the operation that alters this state
- The state of the system at a certain time (snapshot) is a list of transactions



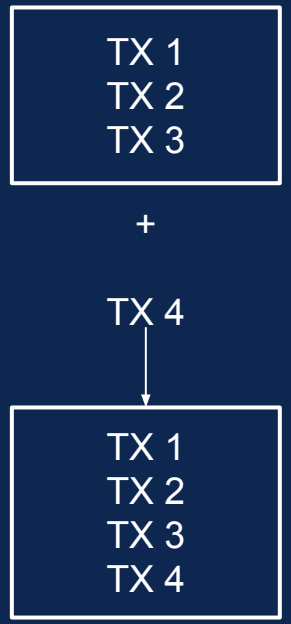
Transactions

- If the ledger maintains the state of the system, a transaction is the operation that alters this state
- The state of the system at a certain time (snapshot) is a list of transactions
- A new transaction refers to a previous one and updates the state of the system

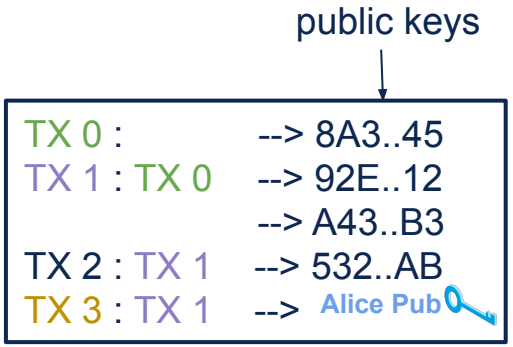


Transactions

- If the ledger maintains the state of the system, a transaction is the operation that alters this state
- The state of the system at a certain time (snapshot) is a list of transactions
- A new transaction refers to a previous one and updates the state of the system
- A valid transaction is signed using the digital signature of the account that holds the previous one



Transactions



Wallet

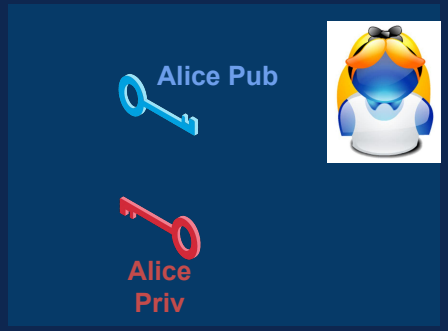
Transactions

public keys

TX 0 :	--> 8A3..45
TX 1 : TX 0	--> 92E..12
	--> A43..B3
TX 2 : TX 1	--> 532..AB
TX 3 : TX 1	--> Alice Pub

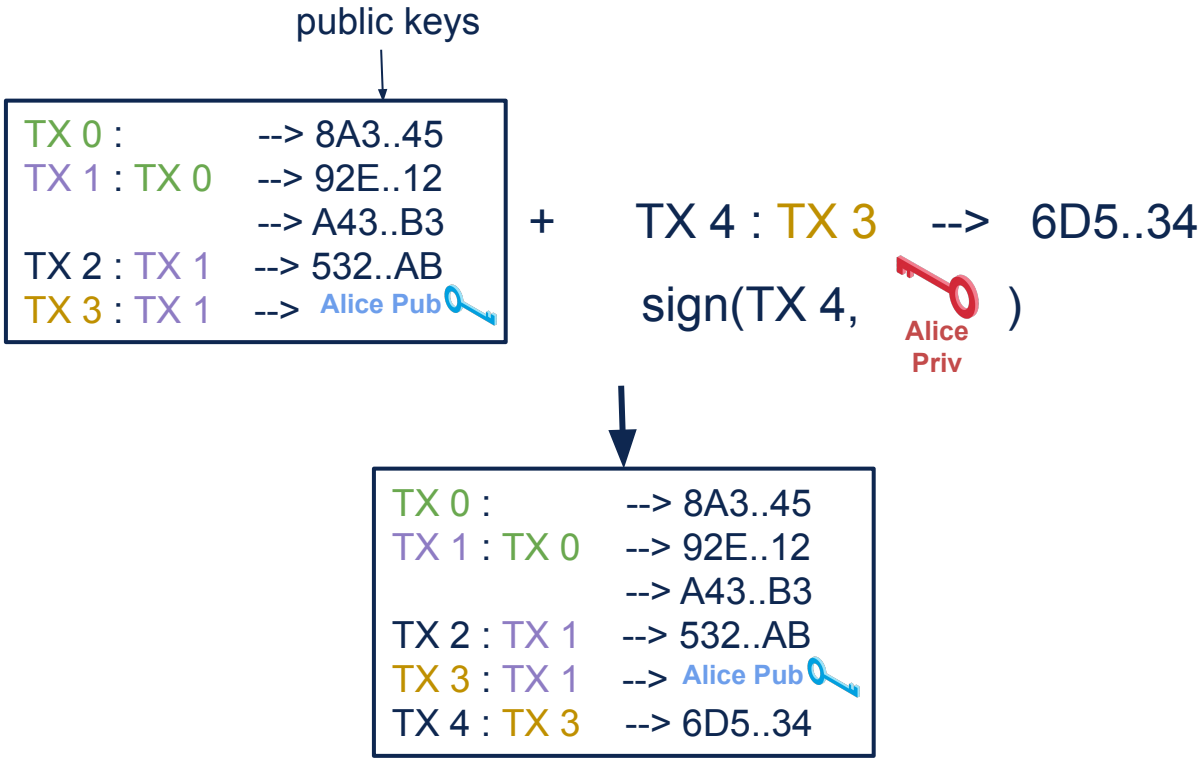
+ TX 4 : TX 3 --> 6D5..34

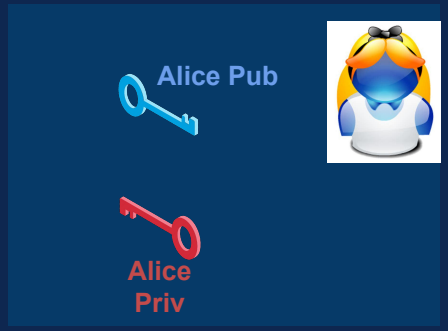
sign(TX 4, Alice Priv)



Wallet

Transactions



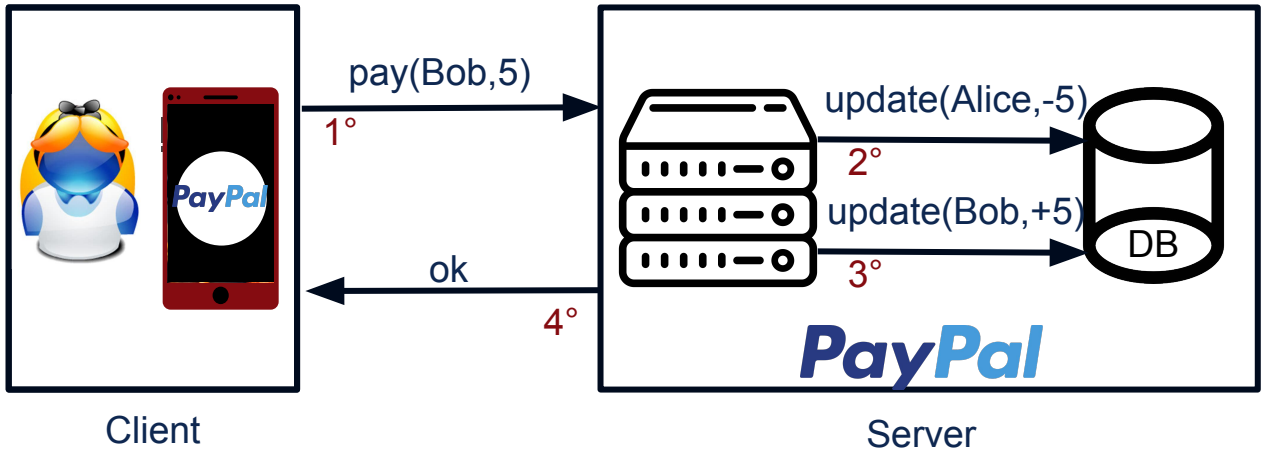


In Italy

- «the storage of a digital document using technology based on DLT produces legal effects of the time stamping of the article 41 of the regulation EU n. 919/2014….»
- «La memorizzazione di un documento informatico attraverso l'uso di tecnologie basate su registri distribuiti produce gli effetti giuridici della validazione temporale elettronica di cui all'articolo 41 del regolamento (UE) n. 910/2014 del Parlamento europeo e del Consiglio, del 23 luglio 2014.»

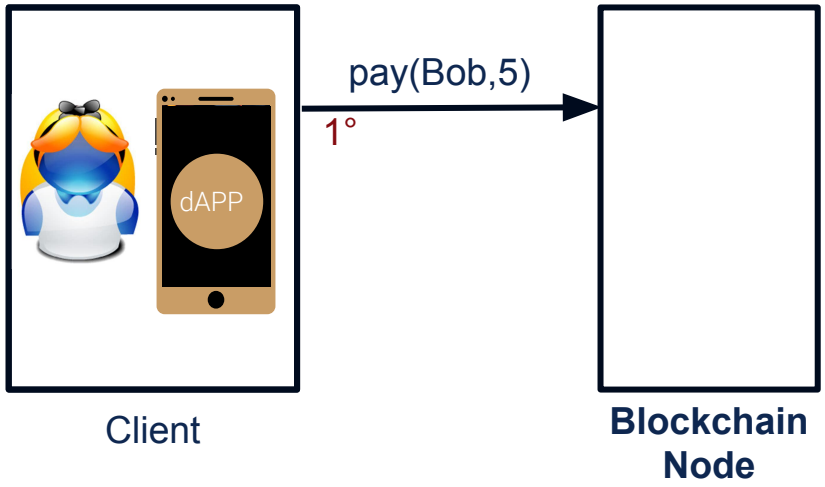
Example comparison

Alice
pays
Bob
5 euros



Example comparison

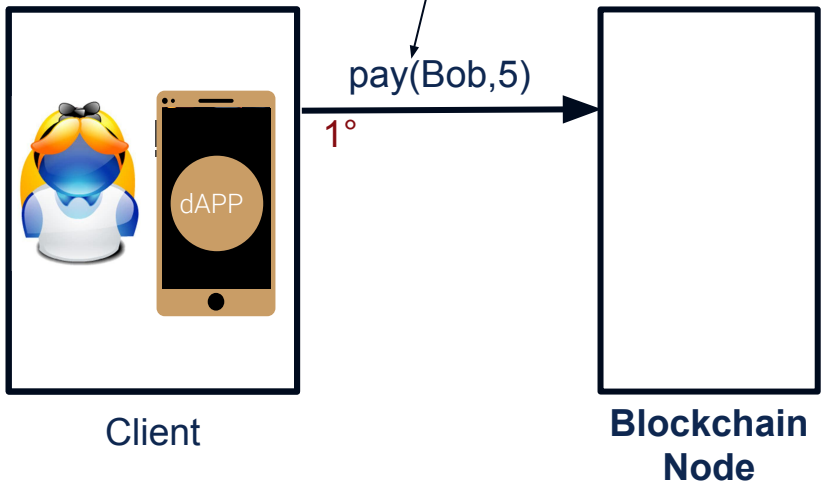
Alice
pays
Bob
5 bitcoin



Example comparison

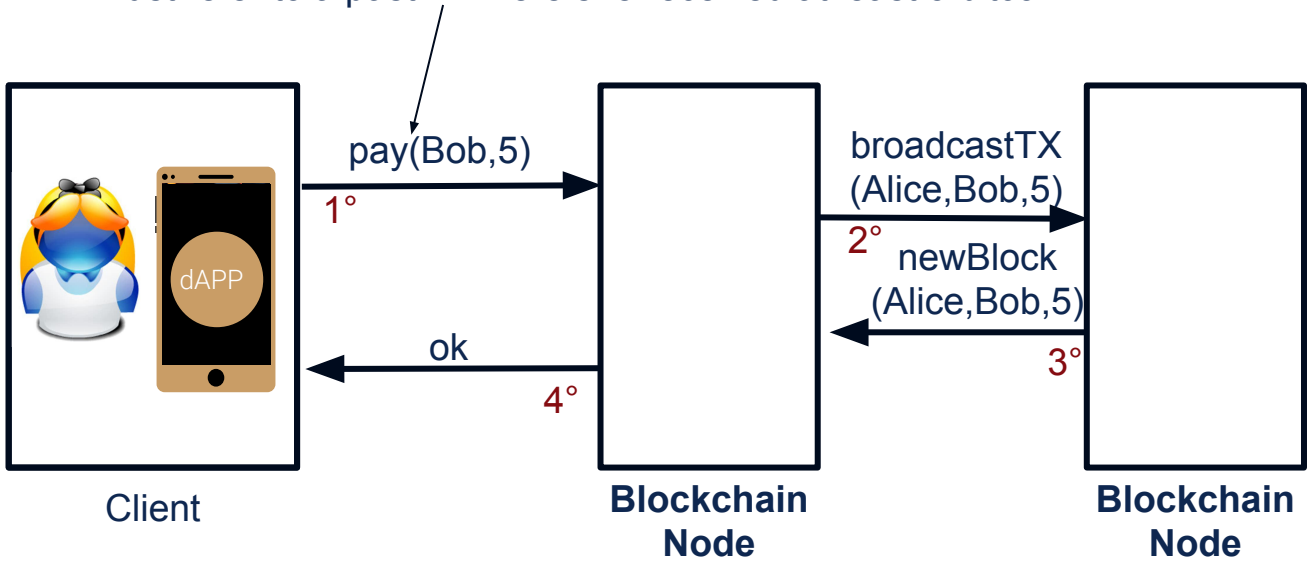
Alice
pays
Bob
5 bitcoin

It means to issue a new transaction in the blockchain, hence Alice must refer to a past TX were she received at least 5 bitcoin



Example comparison

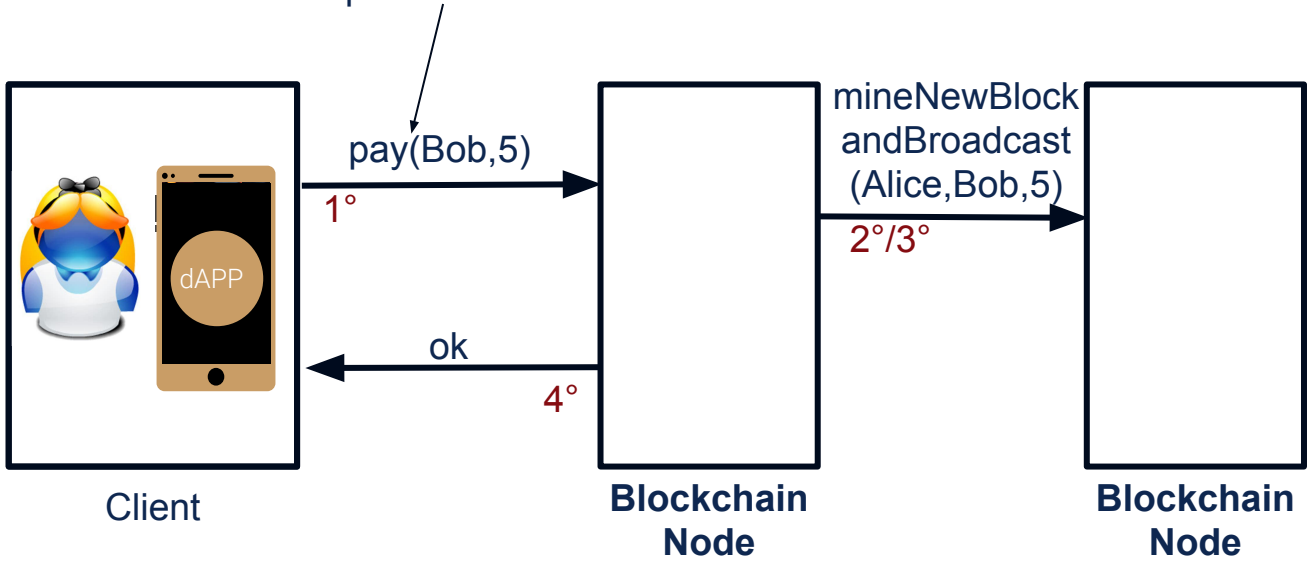
It means to issue a new transaction in the blockchain, hence Alice must refer to a past TX were she received at least 5 bitcoin



Alice
pays
Bob
5 bitcoin

Example comparison

It means to issue a new transaction in the blockchain, hence Alice must refer to a past TX were she received at least 5 bitcoin



Alice
pays
Bob
5 bitcoin

Smart Contracts

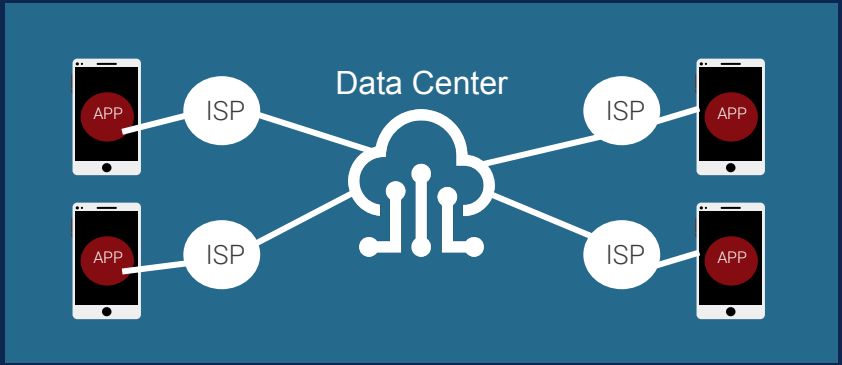
Ethereum

A Smart Contract is (simply) a program

that is executed by all the nodes in a blockchain network

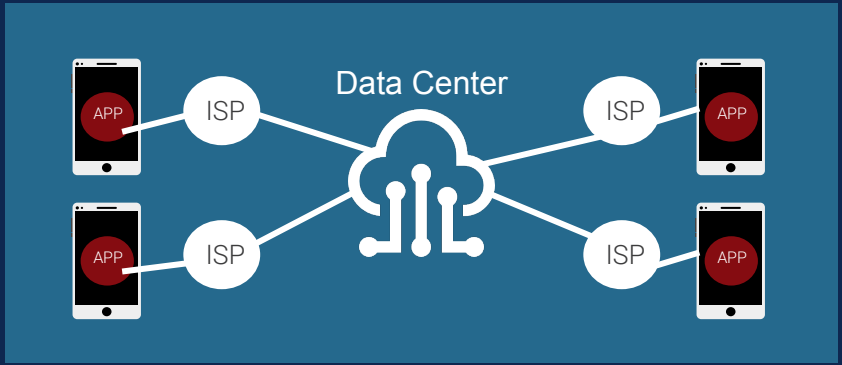
```
/**
* @dev Log a vote for a challenge
* @param challengeID The challenge position in the list
* @param inFavour Boolean value indicating if in favour or not
*/
function vote(uint256 challengeID, bool inFavour)
    public
    onlyEligible(challengeID)
    notExecuted(challengeID)
{
    Challenge storage chall = _challenges[challengeID];
    Vote storage v = chall.votes[msg.sender];
    require(!v.voted, "Voting: already voted");

    _vote(challengeID, inFavour);

    emit Voted(challengeID, msg.sender, inFavour);
}
```

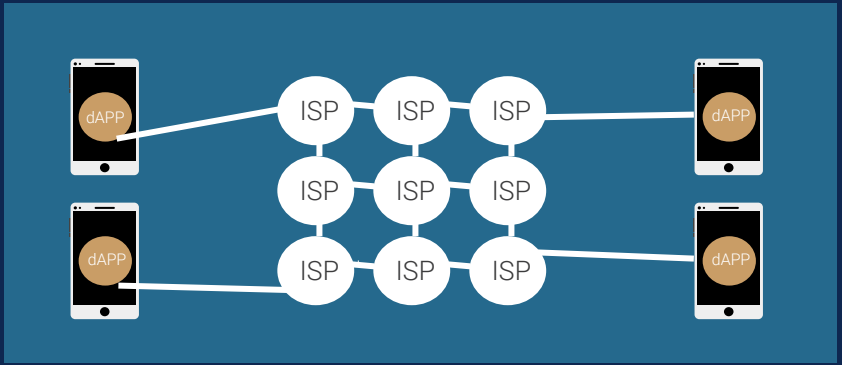
Decentralized computing



Traditional apps make requests that are processed by one or a few servers

Decentralized computing



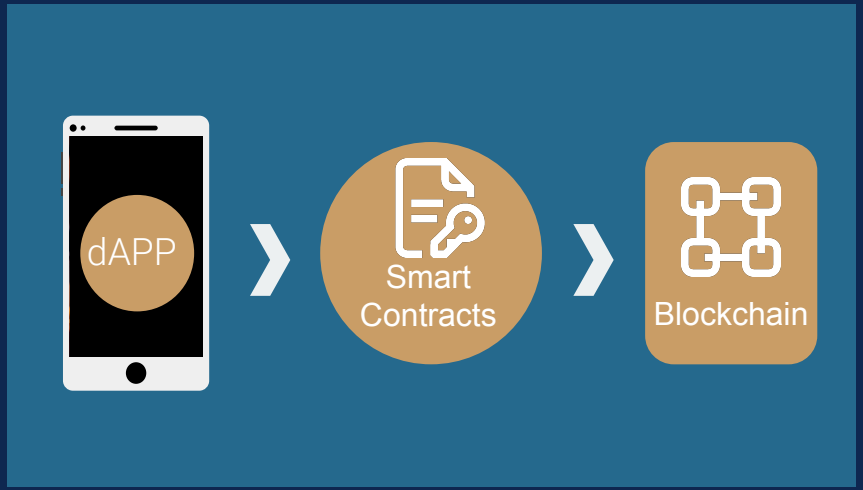


Traditional apps make requests that are processed by one or a few servers

dApps make requests that are processed by all the nodes in the blockchain network (Ethereum)

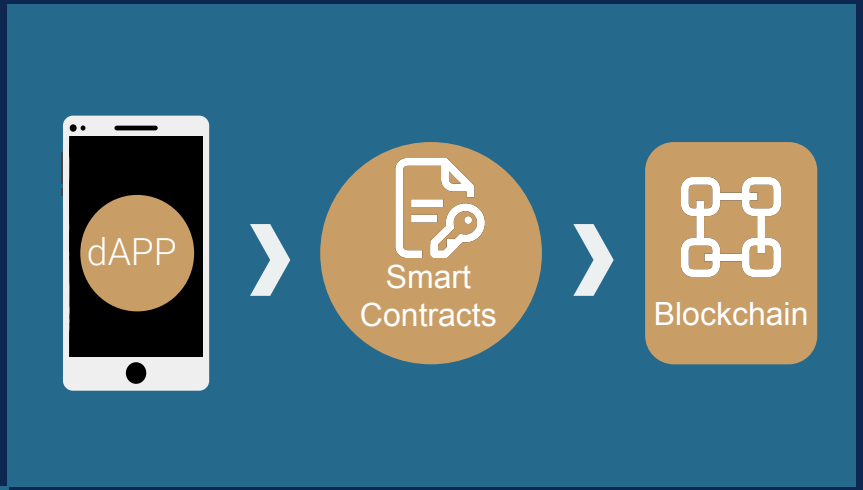
Decentralized Applications

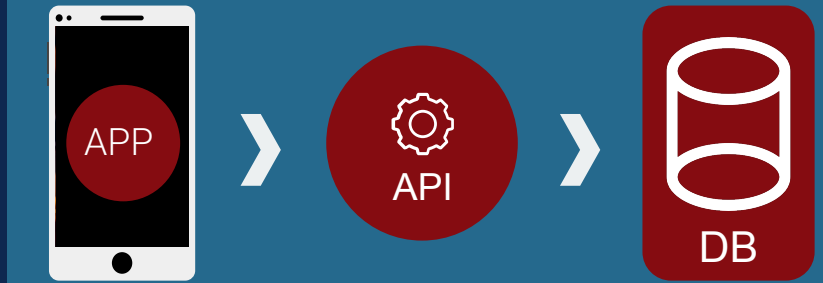
Blockchain-based user-facing interfaces which connect the end user to the technology through a combination of underlying Smart Contracts



Decentralized Applications

Blockchain-based user-facing interfaces which connect the end user to the technology through a combination of underlying Smart Contracts





The relationship between dApps, Smart Contracts and the Blockchain is similar to traditional web applications. Client/server app interacts with a particular server to access its database.

Similarly, dApps use Smart Contracts in order to connect to the particular Blockchain upon which they are based (e.g. Ethereum).

Trust without a third party

1. If you have N independent nodes in a network and the majority (⅔ + 1) of them follows the same “consensus mechanism”

Trust without a third party

1. If you have N independent nodes in a network and the majority (⅔ + 1) of them follows the same “consensus mechanism”
2. If you trust the consensus mechanism (including also the source code it is built upon -> open source)

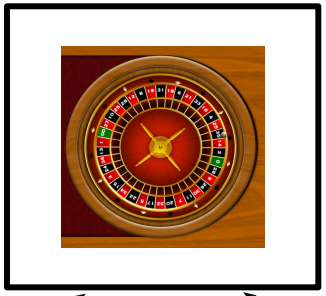
Trust without a third party

1. If you have N independent nodes in a network and the majority (⅔ + 1) of them follows the same “consensus mechanism”
2. If you trust the consensus mechanism (including also the source code it is built upon -> open source)

then

you can trust the correct execution of the program

Practical example why you can trust a smart contract



Server

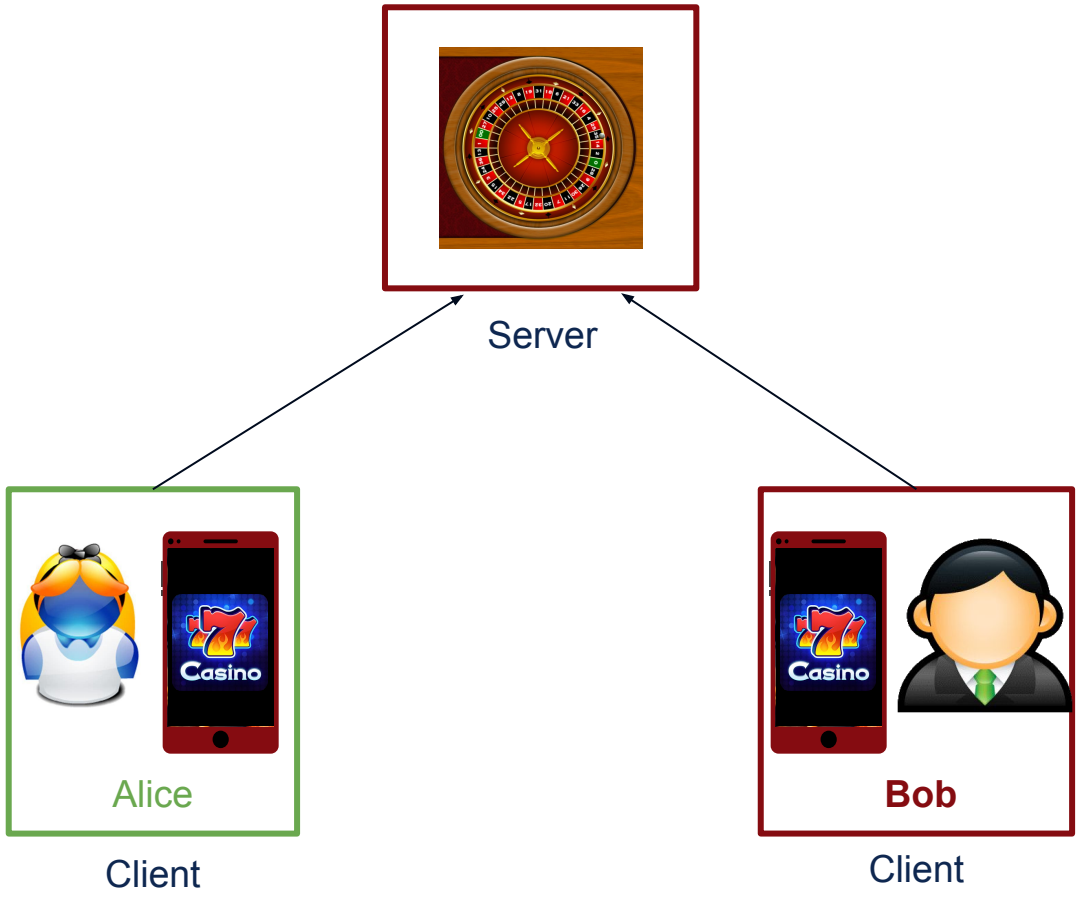
Alice

Client

Bob

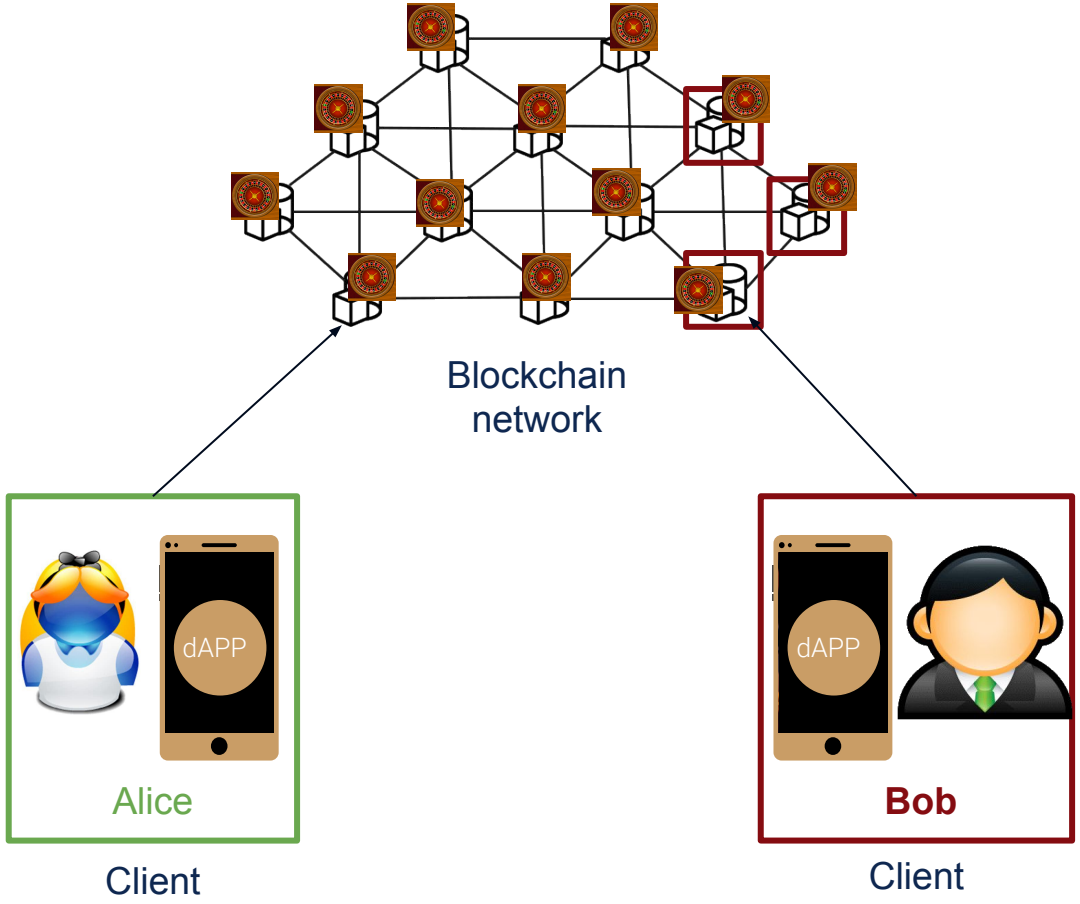
Client

Practical example why you can trust a smart contract



Practical example why you can trust a smart contract

Collusion



Practical example why you can trust a smart contract

The majority is honest

Practical example why you can not trust a smart contract

Practical example why you can not trust a smart contract

- When a Smart Contract has some bugs
  - -> assess the programming code before using it
  - -> use standards and common libraries (OpenZeppelin)

Practical example why you can not trust a smart contract

- When a Smart Contract has some bugs
  - -> assess the programming code before using it
  - -> use standards and common libraries (OpenZeppelin)

- When the blockchain code has some bugs
  - The DAO attack, 3.6 million ether robbed (~ 70$ million)
  - Force the entire network to revert transactions in the ledger (difficult)

Ethereum Smart Contracts

- Allow to easily maintain data structures in the blockchain

Ethereum Smart Contracts

- Allow to easily maintain data structures in the blockchain
- A new transaction refers to a previous one and updates the state of the system

Ethereum Smart Contracts

- Allow to easily maintain data structures in the blockchain
- A new transaction refers to a previous one and updates the state of the system
  - In this case the state of the system considers not only monetary transactions, but also data structures in smart contracts

Ethereum Smart Contracts

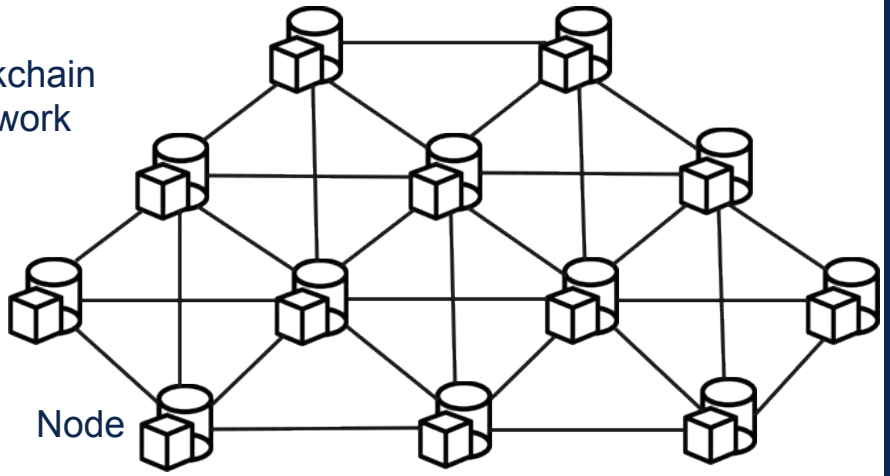
- Allow to easily maintain data structures in the blockchain
- A new transaction refers to a previous one and updates the state of the system
  - In this case the state of the system considers not only monetary transactions, but also data structures in smart contracts
  - The previous one refers to a transaction that holds the contract (machine) code

Ethereum Smart Contracts

- Allow to easily maintain data structures in the blockchain

- A new transaction refers to a previous one and updates the state of the system
  - In this case the state of the system considers not only monetary transactions, but also data structures in smart contracts
  - The previous one refers to a transaction that holds the contract (machine) code
  - The new transaction indicate a piece of code to execute in the contract

Blockchain network

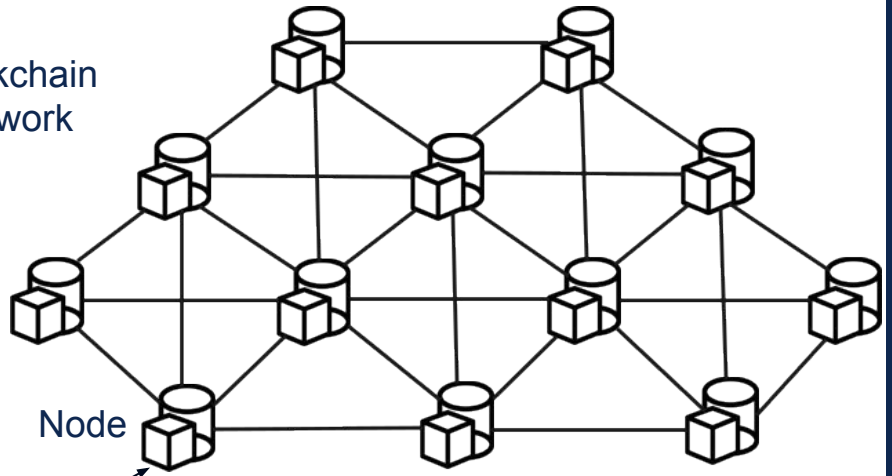


Node

Wallet

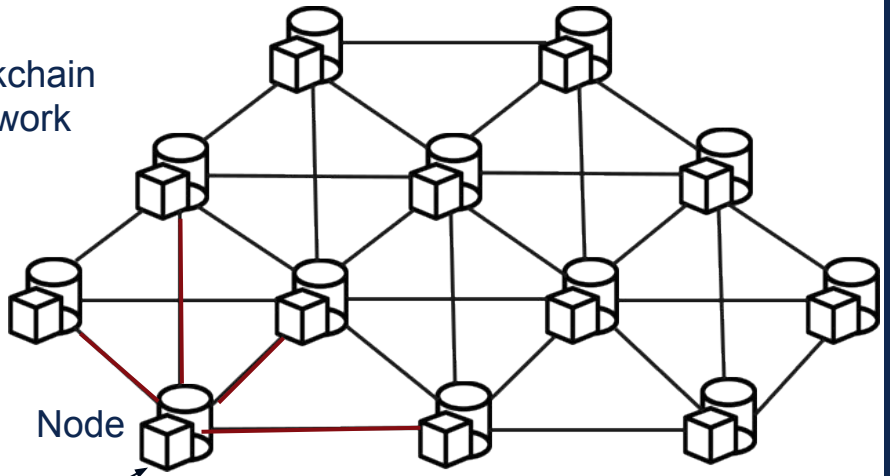
Example: A voting operation in a Smart Contract

Example: A voting operation in a Smart Contract




```
execute(
        SmartContractAddress: VotingContract,
        methodToExecute: vote,
        parameters: challengeID, true
)
```

Example: A voting operation in a Smart Contract




```
execute(
        SmartContractAddress: VotingContract,
        methodToExecute: vote,
        parameters: challengeID, true
)
```

Example: A voting operation in a Smart Contract

```
function vote(Challenge challenge, bool inFavour) public {
    if (inFavour) {
        challenge.inFavour.add(msg.sender); // Alice
    } else {
        challenge.against.add(msg.sender); // Alice
    }
}
```

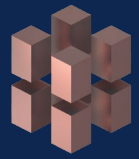
Proof Of Location

in Smart Cities

RIoE

Rights of Internet of Everything

+FOAM



FOAM is an open protocol for a decentralized geospatial data market

FOAM

FOAM is an open protocol for a decentralized geospatial data market

Crypto-Spatial Coordinate (CSC)

A standard for location in Ethereum

Smart Contracts (SC)

FOAM

FOAM is an open protocol for a decentralized geospatial data market

Crypto-Spatial Coordinate (CSC)

A standard for location in Ethereum

Smart Contracts (SC)

Spatial Index and Visualizer (SIV)

A **blockchain explorer** that enables users to engage and act with spatial data

FOAM

FOAM is an open protocol for a decentralized geospatial data market

Crypto-Spatial Coordinate (CSC)

A standard for location in Ethereum **Smart Contracts (SC)**

Spatial Index and Visualizer (SIV)

A **blockchain explorer** that enables users to engage and act with spatial data

Proof of Location (POL)

Consensus on whether an event or agent is verifiably at a certain point in **time and space**

CRYPTO-SPATIAL COORDINATES (CSC)

CSC allow any SC to make an immutable claim to a specific location, using:

- The location **geohash**

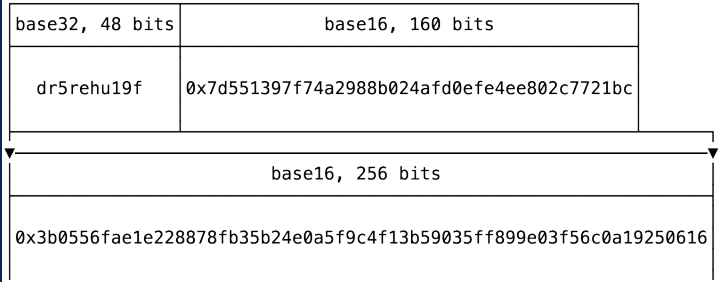
CRYPTO-SPATIAL COORDINATES (CSC)

base32, 48 bits	base16, 160 bits
dr5rehu19f	0x7d551397f74a2988b024afd0efe4ee802c7721bc

CSC allow any SC to make an immutable claim to a specific location, using:

- The location **geohash**
- A corresponding Ethereum address

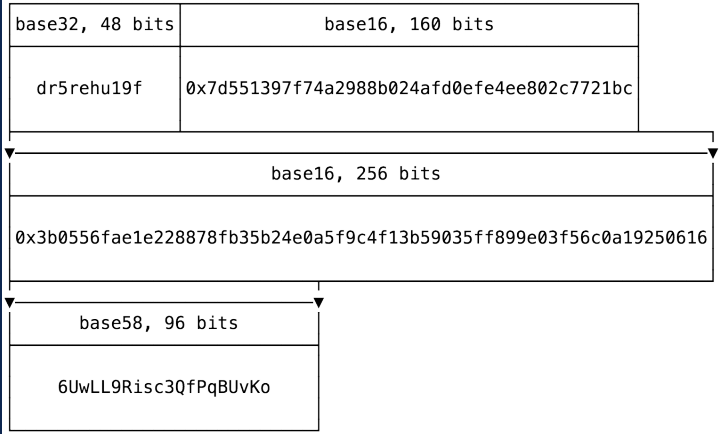
CRYPTO-SPATIAL COORDINATES (CSC)



CSC allow any SC to make an immutable claim to a specific location, using:

- The location **geohash**
- A corresponding Ethereum address

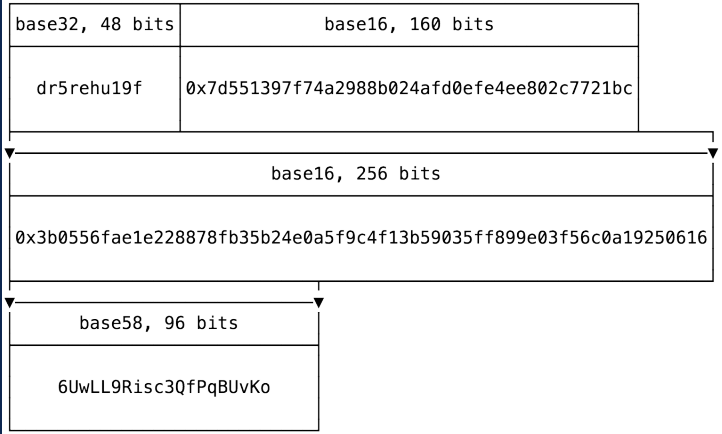
CRYPTO-SPATIAL COORDINATES (CSC)



CSC allow any SC to make an immutable claim to a specific location, using:

- The location **geohash**
- A corresponding Ethereum address

CRYPTO-SPATIAL COORDINATES (CSC)



CSC allow any SC to make an immutable claim to a specific location, using:

- The location **geohash**
- A corresponding Ethereum address

A **registry SC** takes a CSC and decode its location and Ethereum address
Any user can verify if a CSC is where it claims to be by visiting the location

The resolution is one square meter -> 500 trillion unique locations

SPATIAL INDEX AND VISUALIZER (SIV)

A **General Purpose Visual Blockchain Explorer**
front-end interface to visualize SC on a map.

SPATIAL INDEX AND VISUALIZER (SIV)

A **General Purpose Visual Blockchain Explorer**
front-end interface to visualize SC on a map.

NAVIGATE CSCs

Spatially-specific CSC-referenced SC are displayed directly in the application

SPATIAL INDEX AND VISUALIZER (SIV)

A **General Purpose Visual Blockchain Explorer** front-end interface to visualize SC on a map.

NAVIGATE CSCs

Spatially-specific CSC-referenced SC are displayed directly in the application

DEPLOY A SC

A SC is directly deployed using an Ethereum Wallet

SPATIAL INDEX AND VISUALIZER (SIV)

A **General Purpose Visual Blockchain Explorer** front-end interface to visualize SC on a map.

NAVIGATE CSCs

Spatially-specific CSC-referenced SC are displayed directly in the application

VISUALIZE NEW CSC

New CSC-referenced SCs are automatically shown

DEPLOY A SC

A SC is directly deployed using an Ethereum Wallet

FOAM
PROOF OF LOCATION

Provides the **framework and infrastructure** to support a decentralized, censorship resistant **alternative to GPS.**

FOAM
PROOF OF LOCATION

Provides the **framework and infrastructure** to support a decentralized, censorship resistant **alternative to GPS.**

Dynamic PoI

- Provide **consensus** on whether an event or agent is verifiably at a certain point in **time and space**

FOAM
PROOF OF LOCATION

Provides the **framework and infrastructure** to support a decentralized, censorship resistant **alternative to GPS.**

Dynamic PoI

- Provide **consensus** on whether an event or agent is verifiably at a certain point in **time and space**

Static PoL

- Used for Geographic Points of Interest (**PoI**)
- **Token Curated Registries** (TCRs) are a crypto-economic model for curating human readable lists of POI

Static PoL

CONSUMERS
Utilize registry

CANDIDATES
Want to be on registry

CARTOGRAPHERS
Curate registry

Static PoL

1. Candidates submit a **FOAM Token deposit** in order to add a PoI to the registry
   They **wait out an initial challenge**

Static PoL

1. Candidates submit a **FOAM Token deposit** in order to add a PoI to the registry They **wait out an initial challenge**

2. If honest and reputable, the PoI will become part of the list

+

2. If a Cartographer feels that the proposed POI will **degrade the quality** of the TCR, he issues a challenge, by submitting an equal amount of tokens:

Static PoL

1. Candidates submit a **FOAM Token deposit** in order to add a PoI to the registry
   They **wait out an initial challenge**

2. If honest and reputable, the PoI will become part of the list

+

2. If a Cartographer feels that the proposed POI will **degrade the quality** of the TCR, he issues a challenge, by submitting an equal amount of tokens:

   a. This initiates a **voting period** among Cartographers
      They have the ability to verify PoI in person and vote

CONSUMERS
Utilize registry

CANDIDATES
Want to be on registry

CARTOGRAPHERS
Curate registry

Static PoL

1. Candidates submit a **FOAM Token deposit** in order to add a PoI to the registry
   They **wait out an initial challenge**

2. If honest and reputable, the PoI will become part of the list

+

2. If a Cartographer feels that the proposed POI will **degrade the quality** of the TCR, he issues a challenge, by submitting an equal amount of tokens:

   a. This initiates a **voting period** among Cartographers
      They have the ability to verify PoI in person and vote

   b. If the challenging Cartographer succeeds, the Candidate's **deposit is distributed to the winning** Cartographers as a reward for helping to curate TCR

+

CONSUMERS
Utilize registry

CANDIDATES
Want to be on registry

CARTOGRAPHERS
Curate registry

Static PoL

1. Candidates submit a **FOAM Token deposit** in order to add a PoI to the registry
   They **wait out an initial challenge**

2. If honest and reputable, the PoI will become part of the list

+

2. If a Cartographer feels that the proposed POI will **degrade the quality** of the TCR, he issues a challenge, by submitting an equal amount of tokens:

   a. This initiates a **voting period** among Cartographers
      They have the ability to verify PoI in person and vote

   b. If the challenging Cartographer succeeds, the Candidate's **deposit is distributed to the winning** Cartographers as a reward for helping to curate TCR

+

   b. If the challenge is unsuccessful, a percentage of the loser **Cartographer's deposit is forfeited** to the Candidate whose PoI was affirmed and to the winning Cartographers

CONSUMERS
Utilize registry

CANDIDATES
Want to be on registry

CARTOGRAPHERS
Curate registry

Dynamic PoL (IoT guys!)

- FOAM PoL implementation is based on **LPWAN**, a new class of radio highly promising for **IoT**

Low
Power
Wide
Area
Networks

Dynamic PoL (IoT guys!)

- FOAM PoL implementation is based on **LPWAN**, a new class of radio highly promising for **IoT**

- LPWAN can offer the **low power** and **longer battery life** of bluetooth with the **range of cellular**, and access to the unlicensed radio spectrum

Low
Power
Wide
Area
Networks

Dynamic PoL (IoT guys!)

- FOAM PoL implementation is based on **LPWAN**, a new class of radio highly promising for **IoT**
- LPWAN can offer the **low power** and **longer battery life** of bluetooth with the **range of cellular**, and access to the unlicensed radio spectrum
- Trade-off are **low data rate** and **higher latency**
- One of the most promising new radios, **LoRa**, can travel 5–15km at 150 MHz and 1 GHz bands and provide bidirectional communication

Low
Power
Wide
Area
Networks

+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

2. ZONE FORMATION

Authorities start to establish a Zone and pledge to offer **location services** that are enforced by a SC

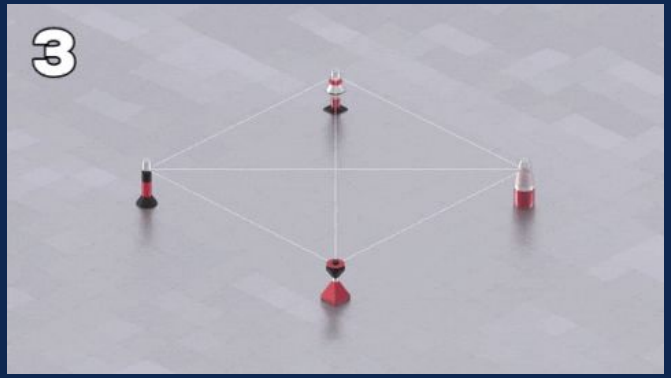
+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

2. ZONE FORMATION

Authorities start to establish a Zone and pledge to offer **location services** that are enforced by a SC



3. CLOCK SYNC

Anchors and Authorities send messages until a **consensus** can be formed on the precise time. This allows to calculate location and to determine the **network geometry**

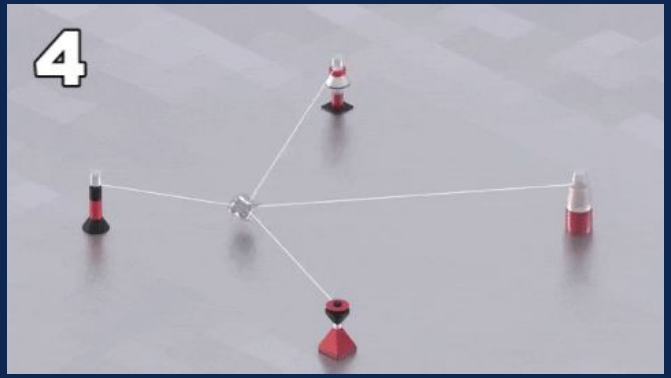
+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

2. ZONE FORMATION

Authorities start to establish a Zone and pledge to offer **location services** that are enforced by a SC



3. CLOCK SYNC

Anchors and Authorities send messages until a **consensus** can be formed on the precise time. This allows to calculate location and to determine the **network geometry**

4. TRIANGULATION

Zones can provide **Presence Claims** for Customers for a transaction fee.

Customer sends a broadcast message to the Zone and Authorities validate his presence through **Time Difference Of Arrival**

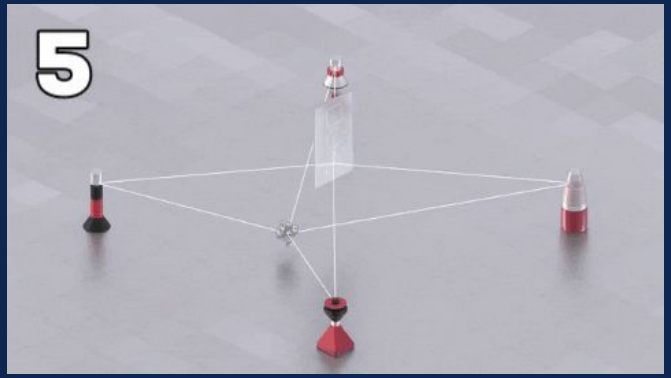
+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

2. ZONE FORMATION

Authorities start to establish a Zone and pledge to offer **location services** that are enforced by a SC



3. CLOCK SYNC

Anchors and Authorities send messages until a **consensus** can be formed on the precise time. This allows to calculate location and to determine the **network geometry**

4. TRIANGULATION

Zones can provide **Presence Claims** for Customers for a transaction fee.

Customer sends a broadcast message to the Zone and Authorities validate his presence through **Time Difference Of Arrival**

5. VERIFIERS

Zones reach consensus through a **local blockchain**.
Verifiers incentivized to check this Zone blockchain for **frauds**

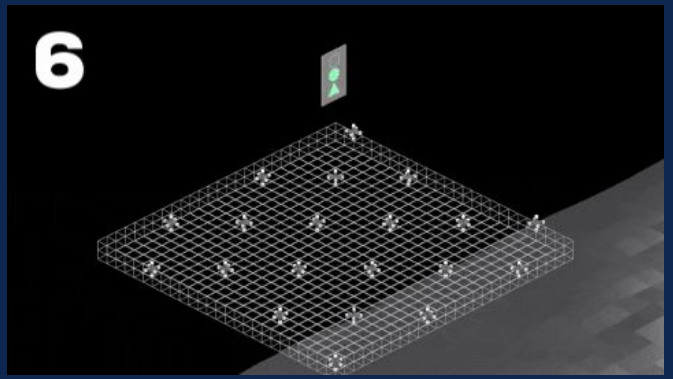
+ Dynamic PoL

1. ANCHORS & AUTHORITIES

Independent Radio **beacons** (Anchors) and Radio **gateways** (Authorities) start participating in the network by deposing some FOAM Tokens

2. ZONE FORMATION

Authorities start to establish a Zone and pledge to offer **location services** that are enforced by a SC



3. CLOCK SYNC

Anchors and Authorities send messages until a **consensus** can be formed on the precise time. This allows to calculate location and to determine the **network geometry**

4. TRIANGULATION

Zones can provide **Presence Claims** for Customers for a transaction fee.

Customer sends a broadcast message to the Zone and Authorities validate his presence through **Time Difference Of Arrival**

5. VERIFIERS

Zones reach consensus through a **local blockchain**.
Verifiers incentivized to check this Zone blockchain for **frauds**

6. PROOF OF LOCATION

Verifiers send Verified or Fraud Proofs to the Ethereum Blockchain and the PoL certificates are created

Platin

Platin is a decentralized, incentivized and **privacy preserving** location credentials service

Secure Verification of Location Claims

User location detection is secured through the use of different mechanisms

Zero Knowledge Proof of Location

User data is protected using the Zero Knowledge Proof protocol

Secure Verification of Location Claims

1. Sensor Fusion

- Location-relevant sensors: GPS, Bluetooth, WiFi, cellular network and accelerometers

Secure Verification of Location Claims

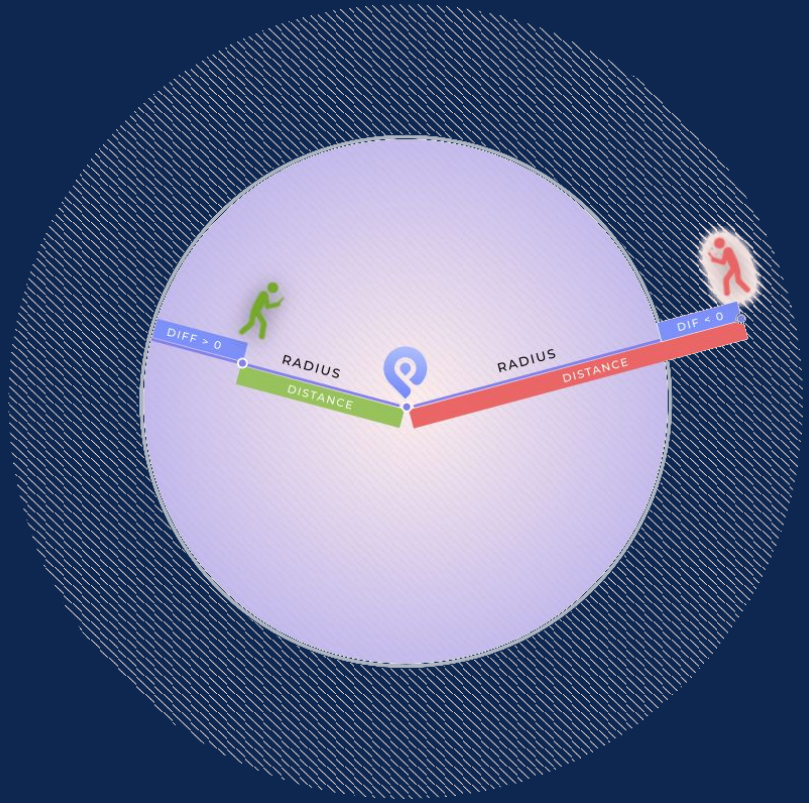
1. Sensor Fusion
   - Location-relevant sensors: GPS, Bluetooth, WiFi, cellular network and accelerometers
2. Behavior Over Time
   - Tracks user behavior over longer periods of time and builds up users' reputation scores
   - AI technologies enable to secure both sensor fusion and behavior over time (XAIN Technologies)

Secure Verification of Location Claims

1. Sensor Fusion
   - Location-relevant sensors: GPS, Bluetooth, WiFi, cellular network and accelerometers
2. Behavior Over Time
   - Tracks user behavior over longer periods of time and builds up users' reputation scores
   - AI technologies enable to secure both sensor fusion and behavior over time (XAIN Technologies)
3. P2P Witnessing
   - Users will be able to act as witnesses for each others' locations through the use of short-range communication
   - Users' efforts in verifying other's location claims will be rewarded

Zero Knowledge Proof of Location

Protocol that allows a **Verifier** to test whether position committed by a **Prover** is inside or outside the radius of a service area, without revealing exact location



Zero Knowledge Proof of Location [1\2]

Sphere Equation

Prover

Verifier

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$$

$A(x, y, z)$ Alice's position

$O(x_0, y_0, z_0)$ Reference point's position

Zero Knowledge Proof of Location [1\2]

Sphere Equation

Prover

Verifier

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 < d^2$$

$A(x, y, z)$ Alice's position

$O(x_0, y_0, z_0)$ Reference point's position

Zero Knowledge Proof of Location [1\2]

Sphere Equation

Prover

Verifier

$$d^2 - (x - x_0)^2 - (y - y_0)^2 - (z - z_0)^2 > 0$$

$A(x, y, z)$ Alice's position

$O(x_0, y_0, z_0)$ Reference point's position

Zero Knowledge Proof of Location [2\2]

Lagrange Theorem

Prover

Verifier

$$d^2 - (x - x_0)^2 - (y - y_0)^2 - (z - z_0)^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2$$

No tuple exists for negative differences (points outside the range)

$A(x, y, z)$

Alice's position

$O(x_0, y_0, z_0)$

Reference point's position

Discrete logarithm to hide the secret

Prover

Verifier

Let *p* and *q* be primes, $n = pq$. Then for some (properly chosen) positive $g < n$, the function:

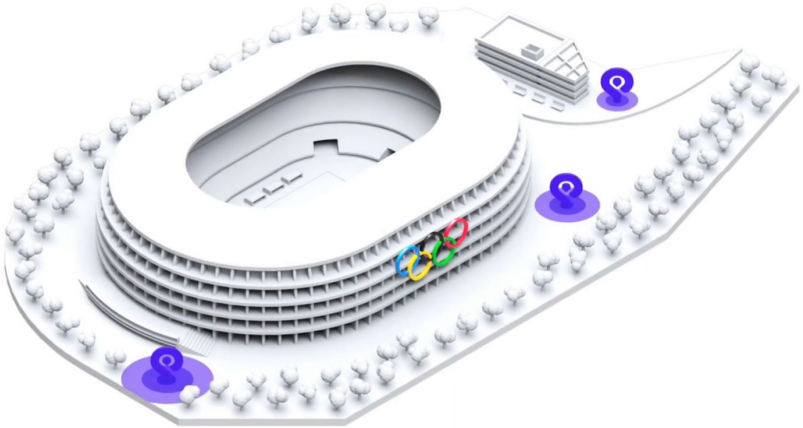
$$f(u) = g^u \pmod{n}$$

is a **one-way** function if *p* and *q* are unknown.

Hence Alice only needs to report values $g^x \pmod{n}$, $g^y \pmod{n}$ and $g^z \pmod{n}$ in order to let Bob verify the distance, without knowing $A(x, y, z)$, through the equality:

$$g^{d^2-(x-x_0)^2-(y-y_0)^2-(z-z_0)^2} = g^{a_1^2+a_2^2+a_3^2+a_4^2}$$

PLATIN
Use case



Tokyo Olympics 2020

Questions?

I ask you:

Could Proof of Location be used as evidence?