Spyhce

The Usability, Security, and Privacy Limits of Blockchain

Codiax 2019

JIBE.COMPANY

Hello!

- ➔ I’m Joris van Rooij
- ➔ Software Architect at Jibe.Company
- ➔ From Eindhoven, the Netherlands

I like... **Free Software**
I’m all about... **Digital Rights**
My pet peeves are... **Operational / Information Security**

JIBE.COMPANY

Our goals

- ➔ Get to know what “blockchain” *actually* means
- ➔ Find out what its strengths are
- ➔ Uncover its weaknesses
- ➔ Learn when its application is appropriate

JIBE.COMPANY

DISCLAIMER TIME

JIBE.COMPANY

Our Agenda

1. Blockchain 101
2. Introducing PoSeID-on
3. Blockchain in PoSeID-on
4. Blockchain applicability

JIBE.COMPANY

Blockchain 101

JIBE.COMPANY

Quick High-Level Overview

JIBE.COMPANY

Distributed Systems

JIBE.COMPANY

Distributed System

A distributed system is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another.

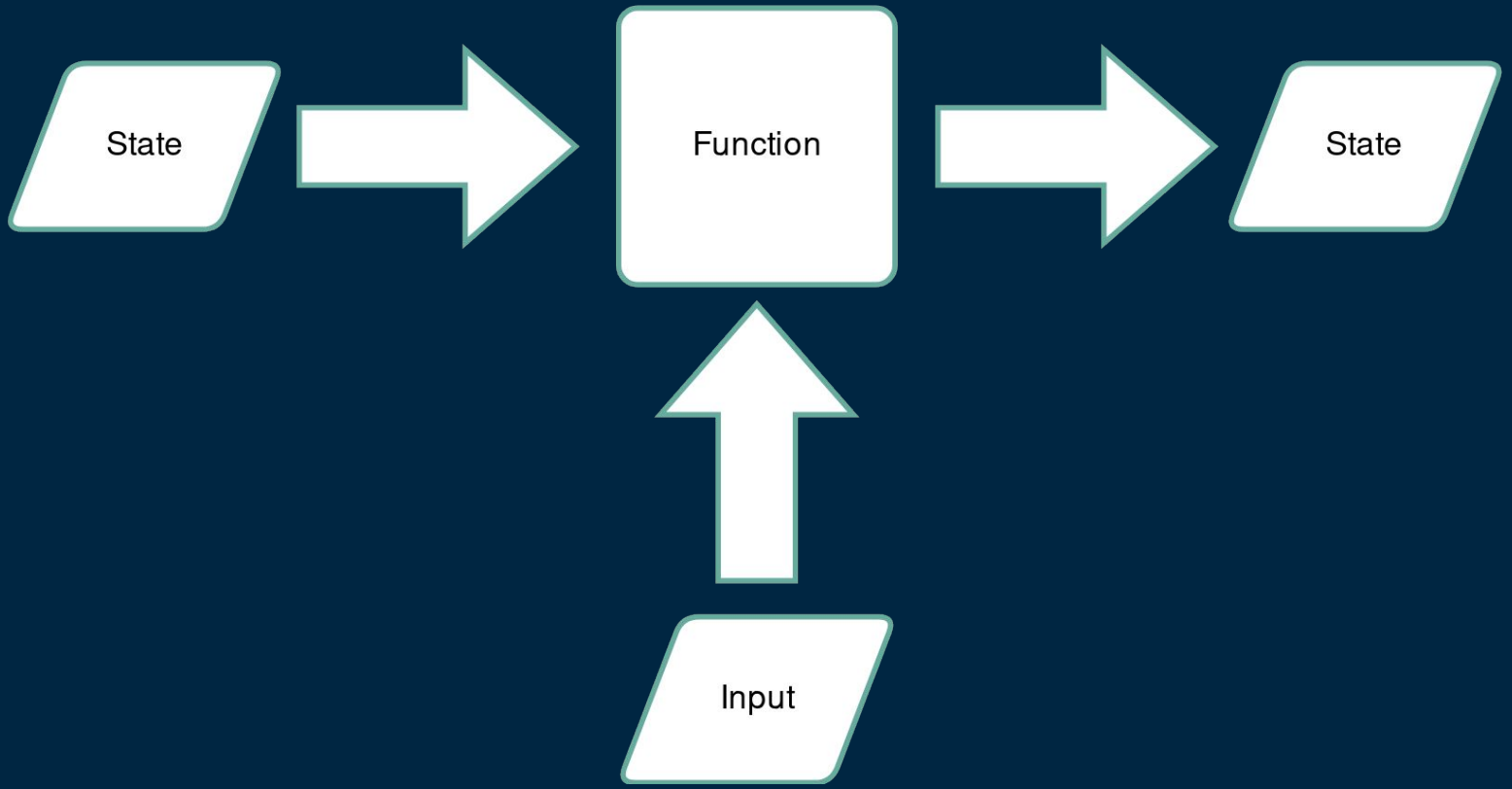
JIBE.COMPANY

Distributed System

A working distributed system is a state machine, distributed across a network on multiple machines, with consensus about the state.

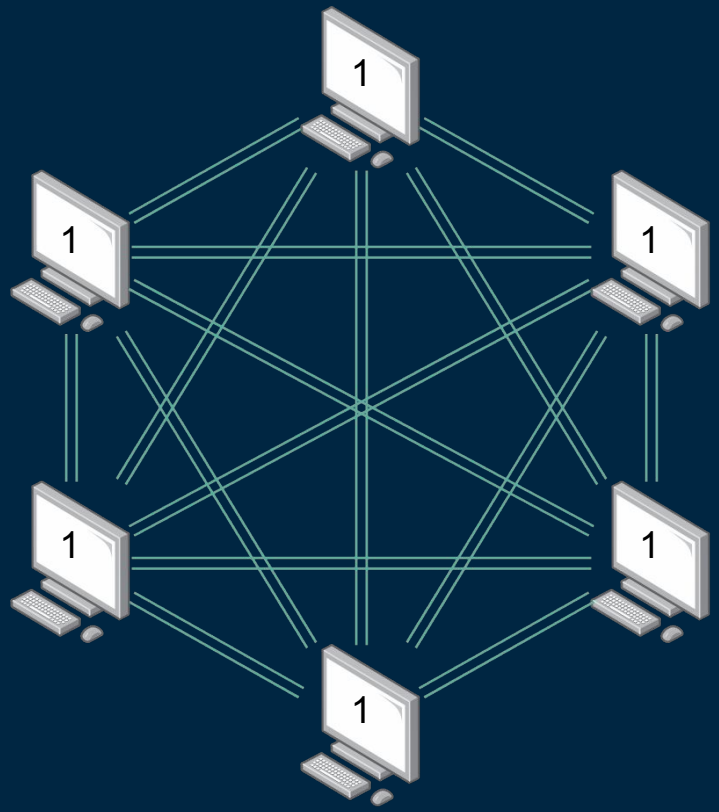
JIBE.COMPANY

State Transition Function



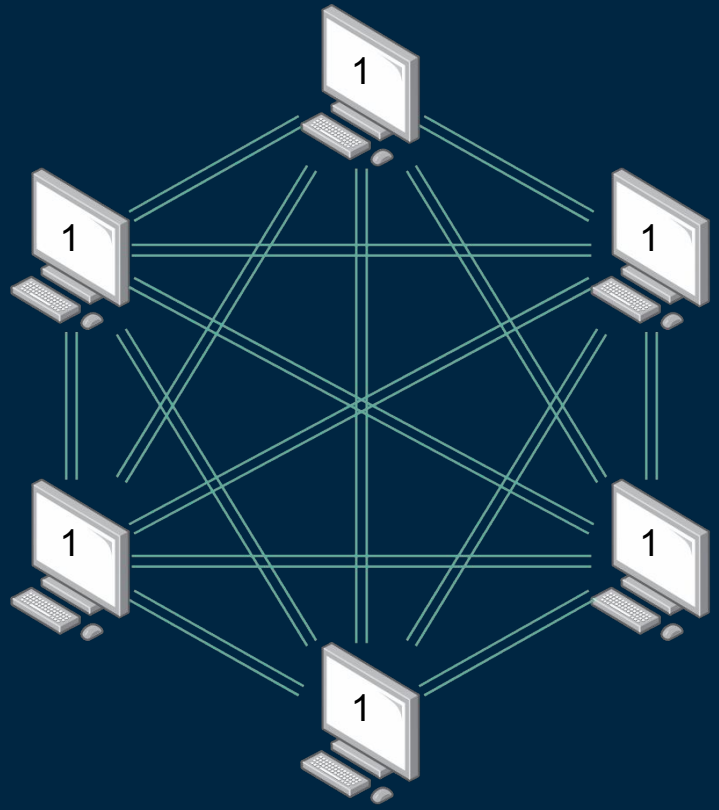
JIBE.COMPANY

Distributed System



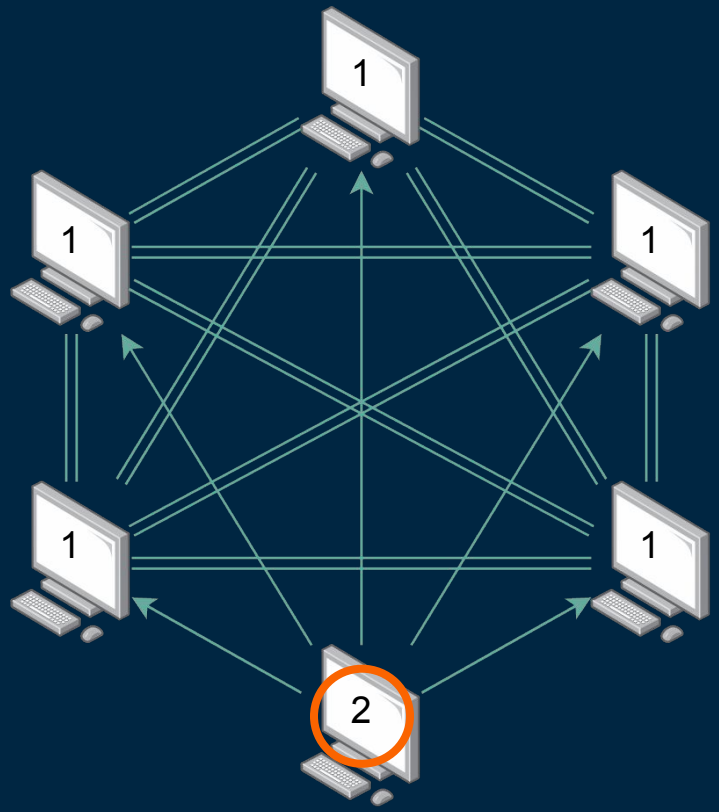
JIBE.COMPANY

Approach: Full Mesh Broadcast



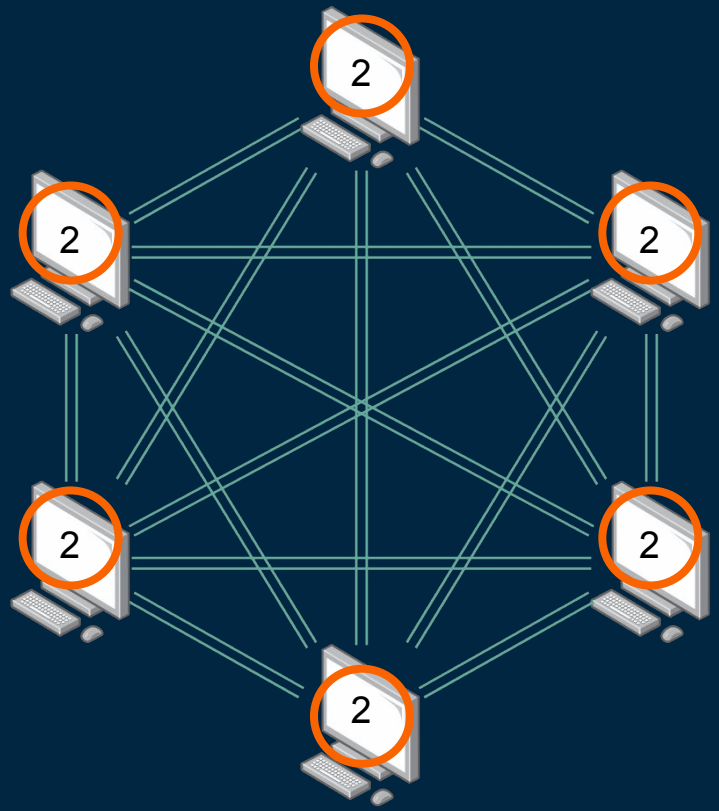
JIBE.COMPANY

Approach: Full Mesh Broadcast



JIBE.COMPANY

Approach: Full Mesh Broadcast



JIBE.COMPANY

Approach: Full Mesh Broadcast

Every computer has a direct connection to every other computer

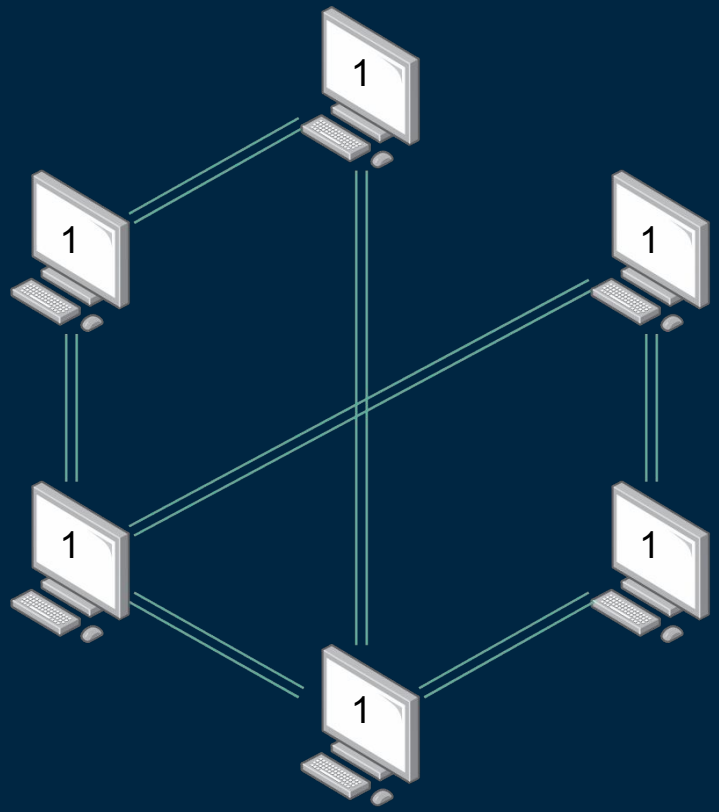
JIBE.COMPANY

Approach: Full Mesh Broadcast

That won’t scale very well

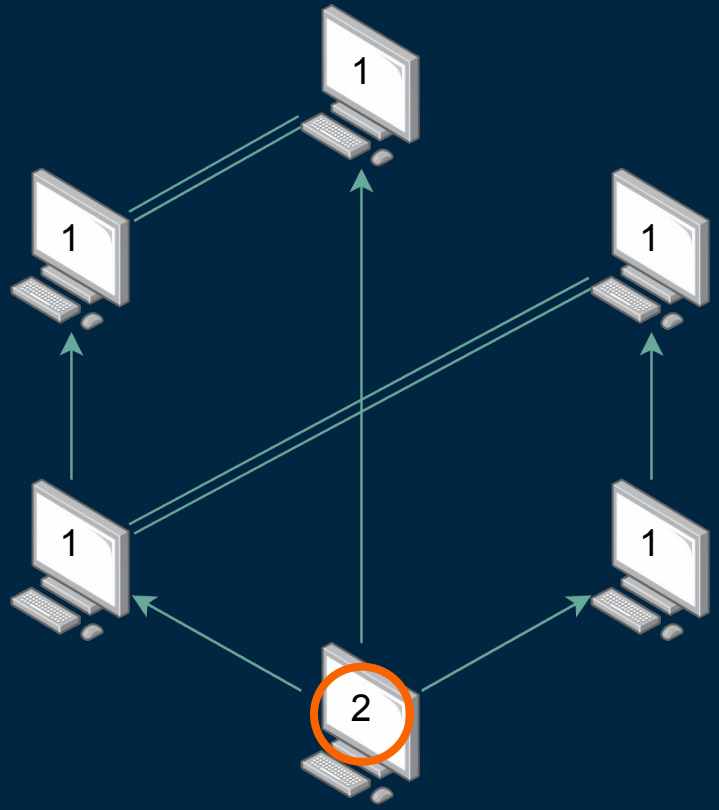
JIBE.COMPANY

Approach: Partial Mesh Broadcast



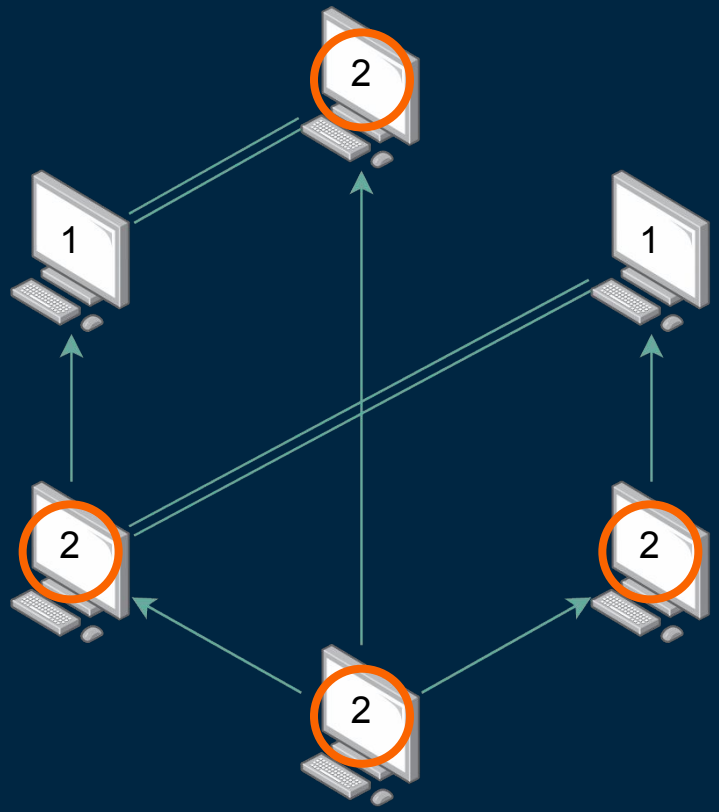
JIBE.COMPANY

Approach: Partial Mesh Broadcast



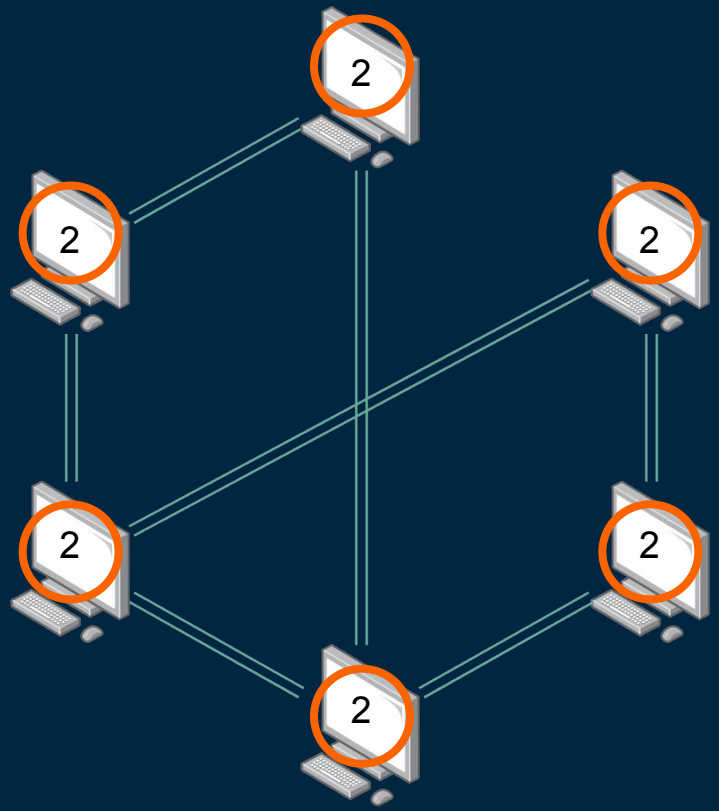
JIBE.COMPANY

Approach: Partial Mesh Broadcast



JIBE.COMPANY

Approach: Partial Mesh Broadcast



JIBE.COMPANY

Approach: Partial Mesh Broadcast

It’s better, but...

JIBE.COMPANY

Node Failures

Computers are not connected all the time

JIBE.COMPANY

Node Failures

- ➔ Network interference
- ➔ Hardware failure
- ➔ Software failure
- ➔ Your mom unplugging your computer

...etc

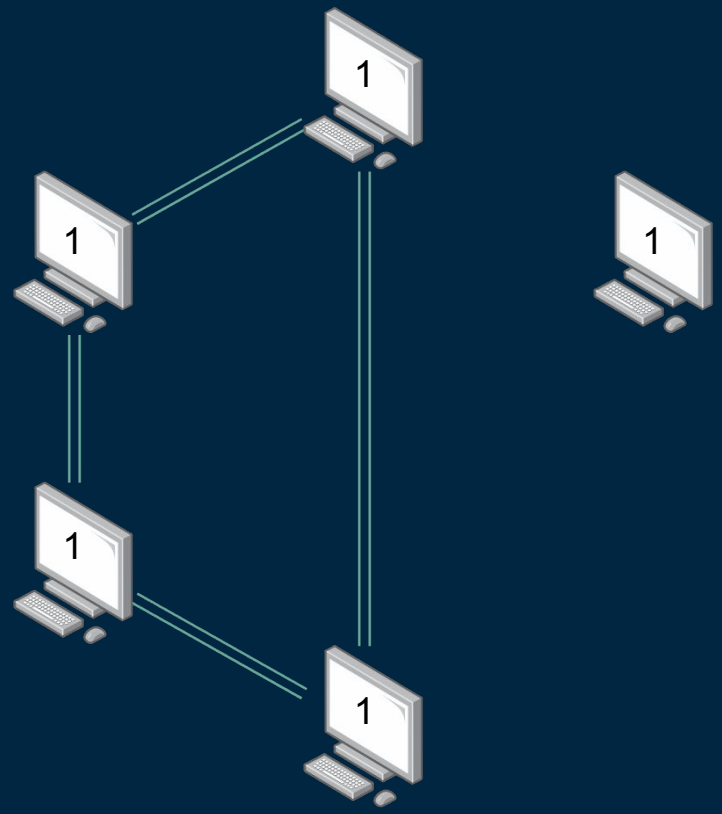
JIBE.COMPANY

```
00:00 -!- Netsplit a.irc.net <-> b.irc.net quits: usera, userb, userc
00:00 -!- Netsplit over, joins: usera, userb, userc
```

Remember these?

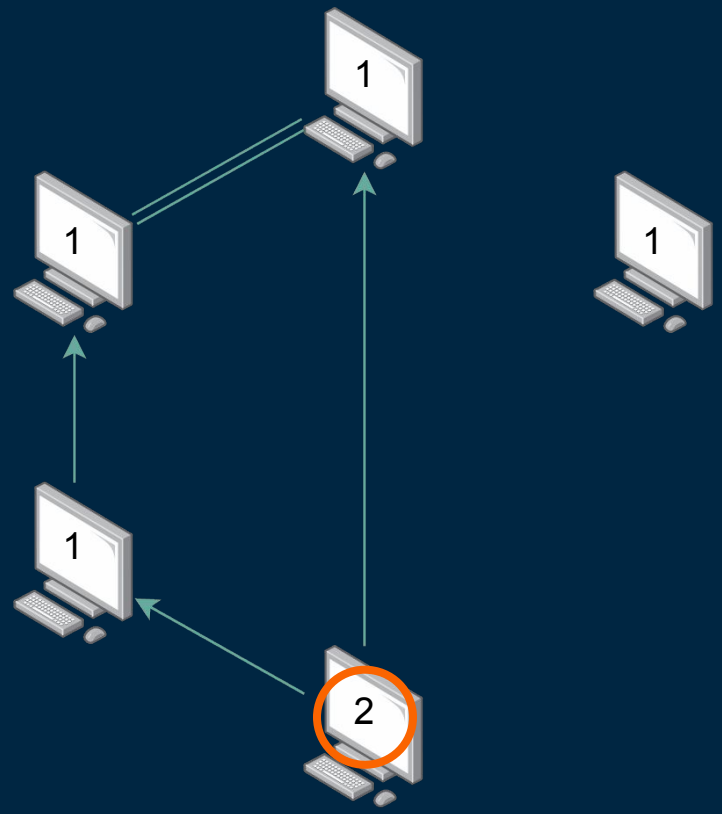
JIBE.COMPANY

Node Failures



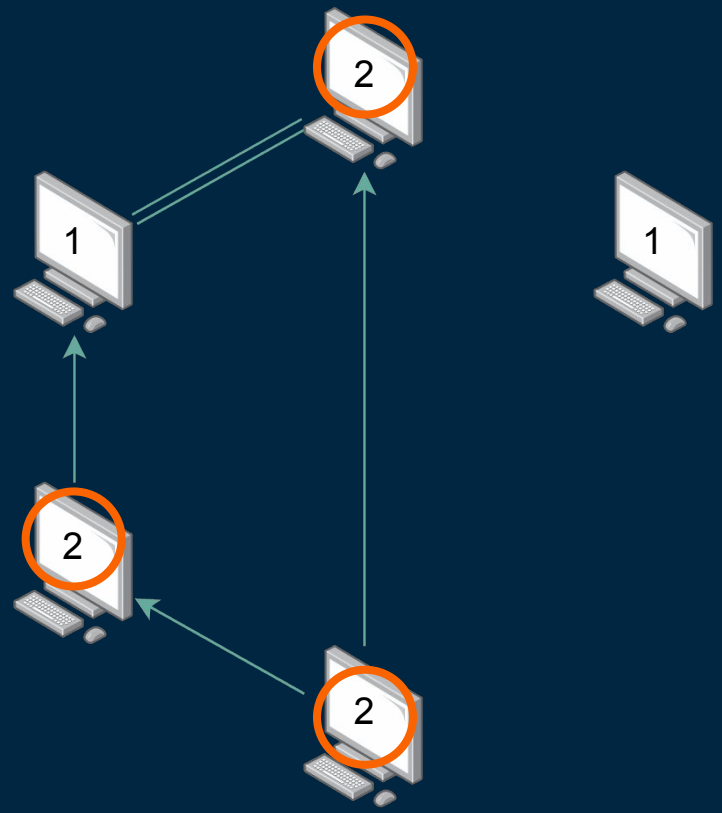
JIBE.COMPANY

Node Failures



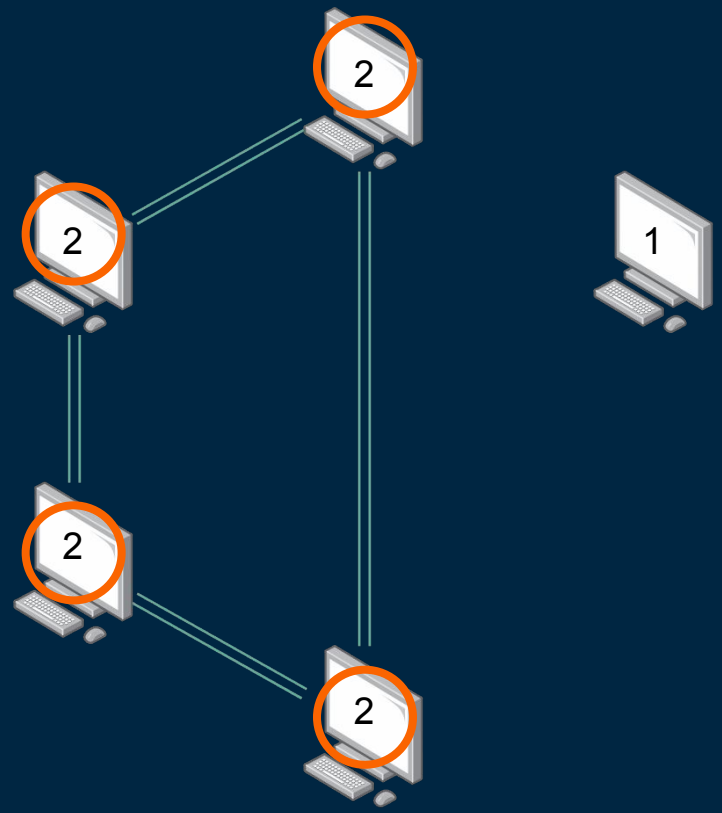
JIBE.COMPANY

Node Failures



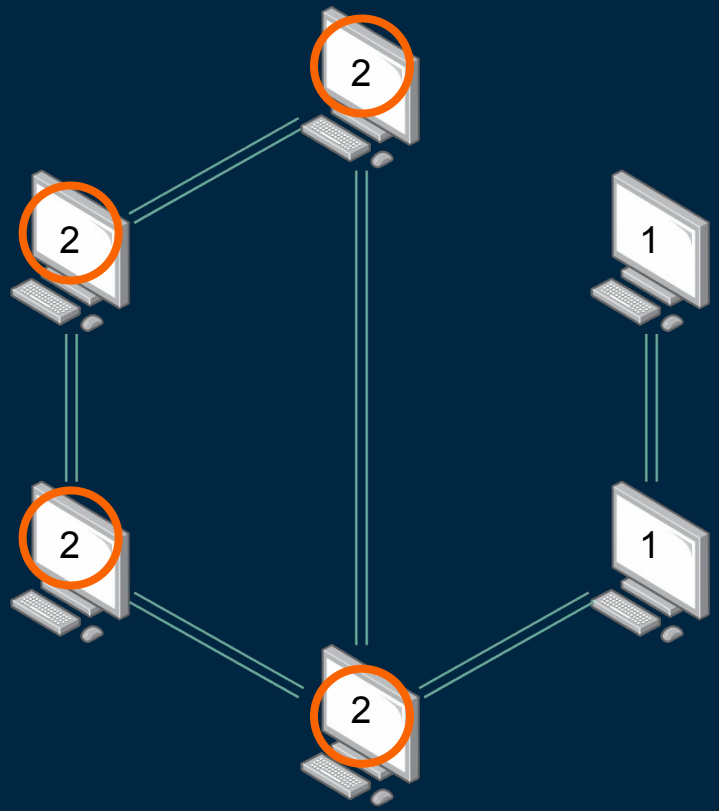
JIBE.COMPANY

Node Failures



JIBE.COMPANY

Node Failures



JIBE.COMPANY

Node Failures

This is not fault tolerant

JIBE.COMPANY

Approach: Transaction Log

We need sequential transactions; a log

JIBE.COMPANY

Approach: Transaction Log

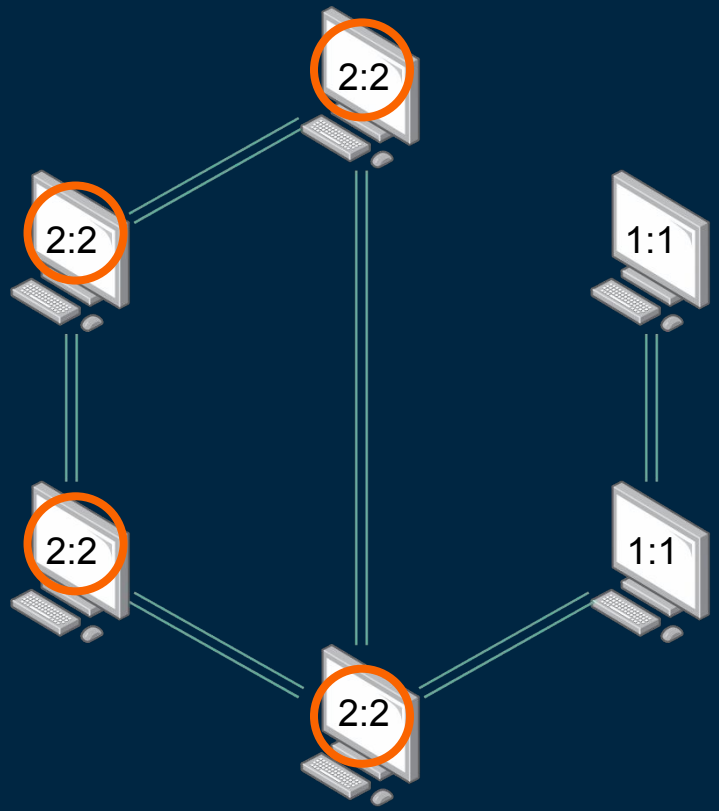
Step 1: set value to 1

Step 2: set value to 2

Step 3: set value to 42

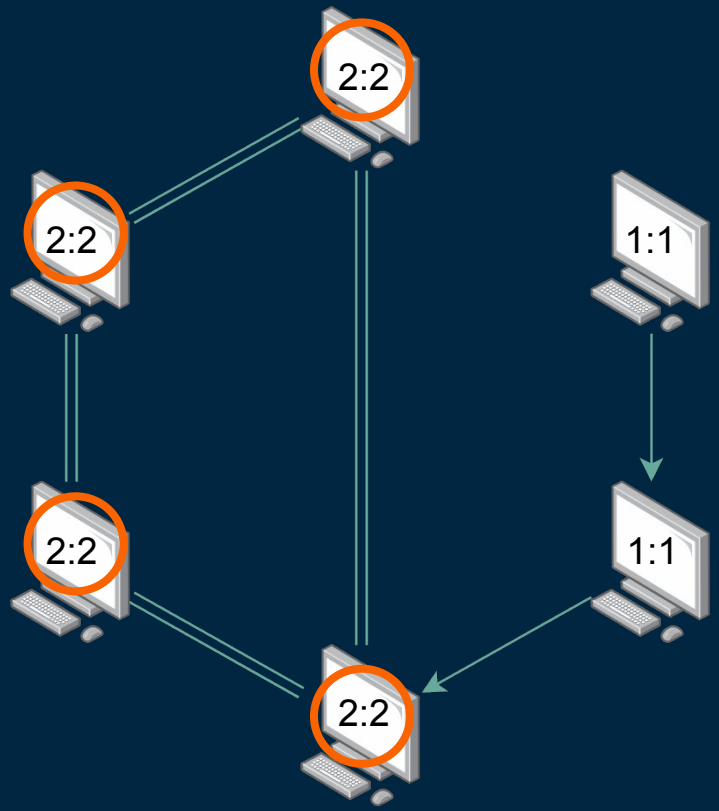
JIBE.COMPANY

Approach: Transaction Log



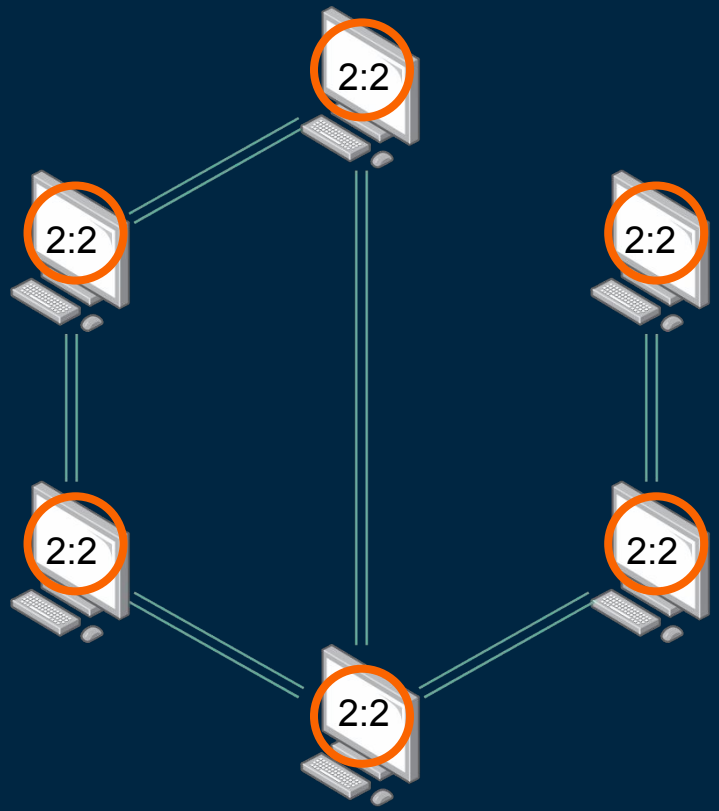
JIBE.COMPANY

Approach: Transaction Log



JIBE.COMPANY

Approach: Transaction Log

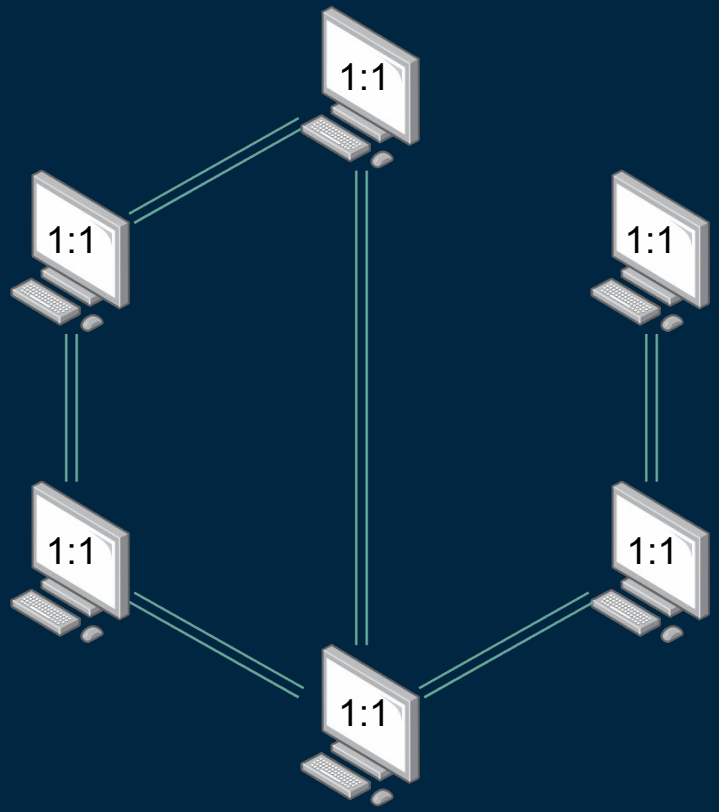


Approach: Transaction Log

It’s better, but...

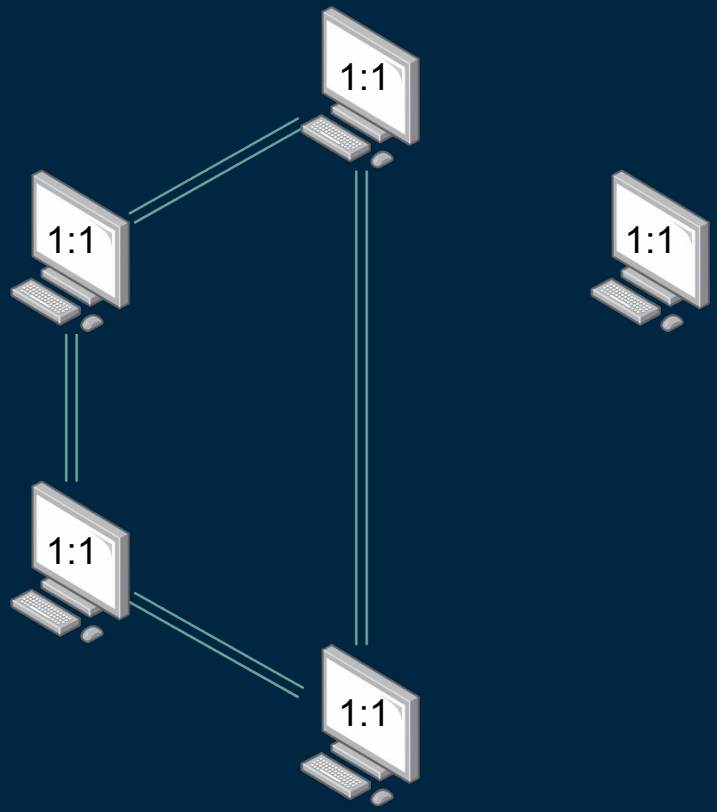
JIBE.COMPANY

Approach: Transaction Log



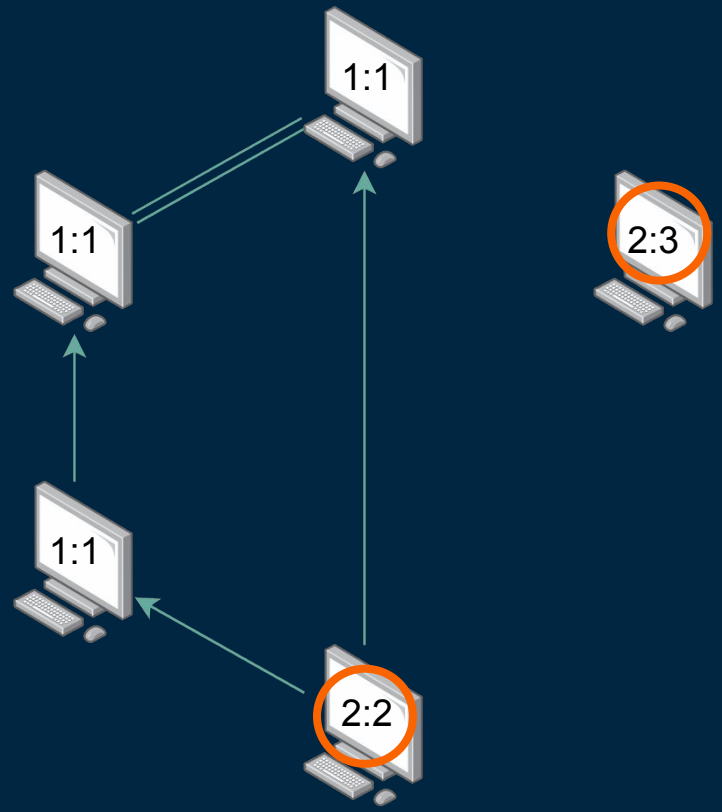
JIBE.COMPANY

Approach: Transaction Log



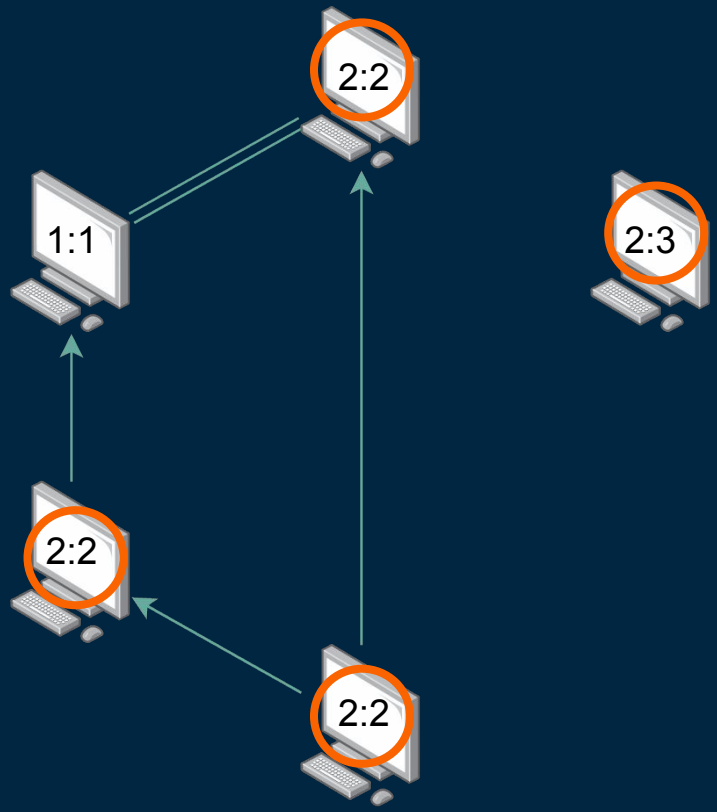
JIBE.COMPANY

Approach: Transaction Log



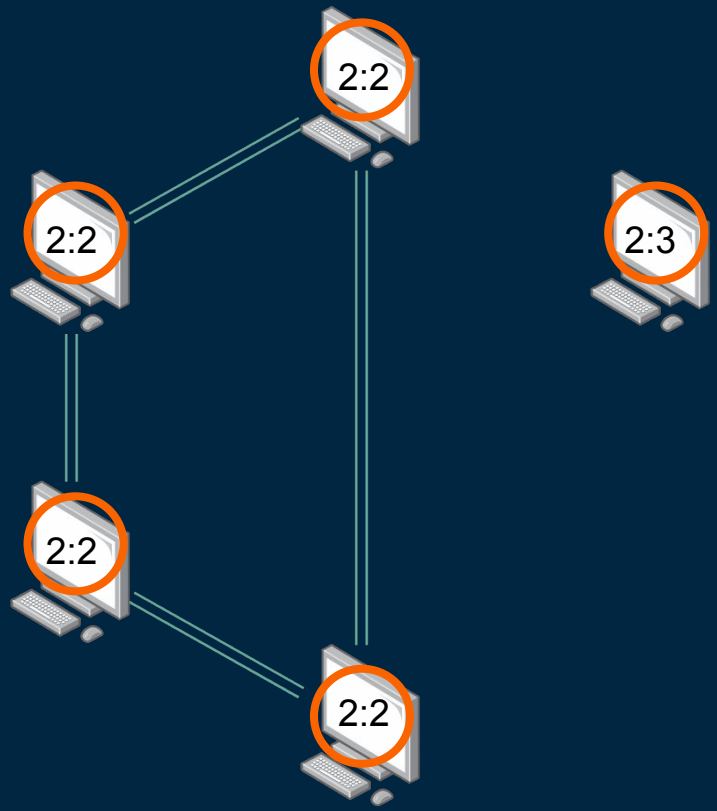
JIBE.COMPANY

Approach: Transaction Log

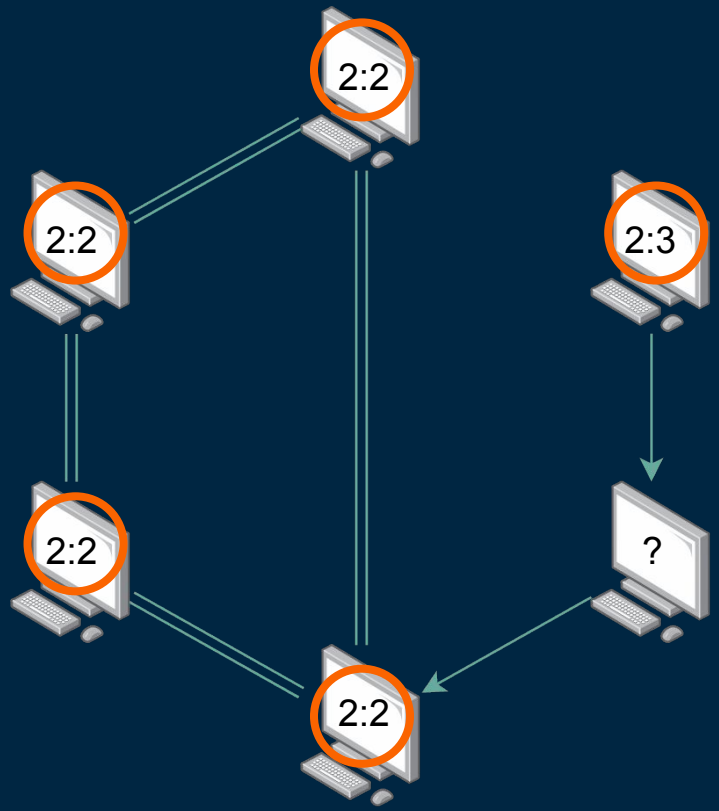


JIBE.COMPANY

Approach: Transaction Log



Approach: Transaction Log



Approach: Transaction Log

Step 1: set value to 1

Step 2: set value to 2

Step 2: set value to 3

JIBE.COMPANY

Approach: Master Node

Introduce a master with the final say in the order and validity of transactions

JIBE.COMPANY

Approach: Master Node

Step 1: set value to 1

Step 2: set value to 2

~~**Step 2:** set value to 3~~

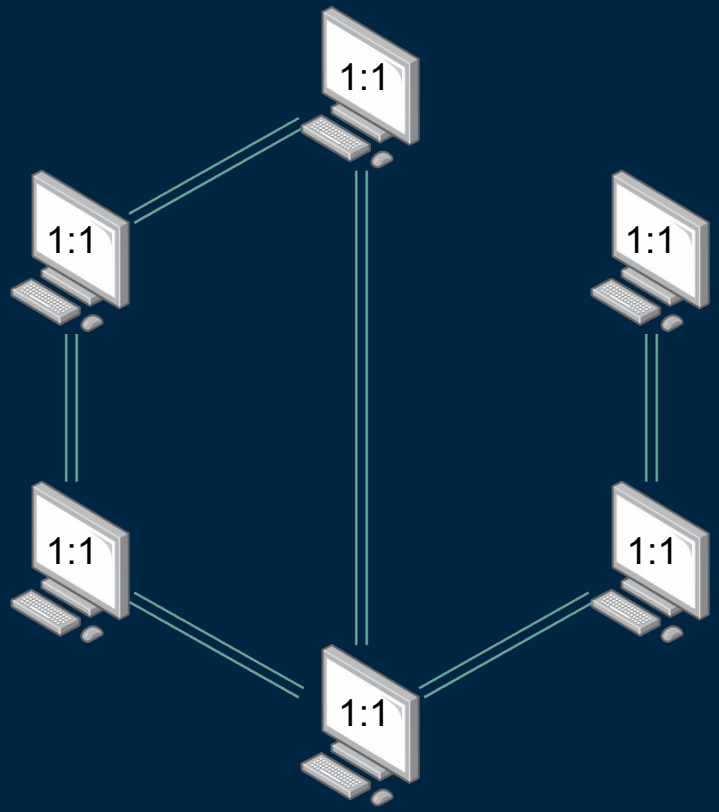
JIBE.COMPANY

Approach: Master Node

Who gets to be master?

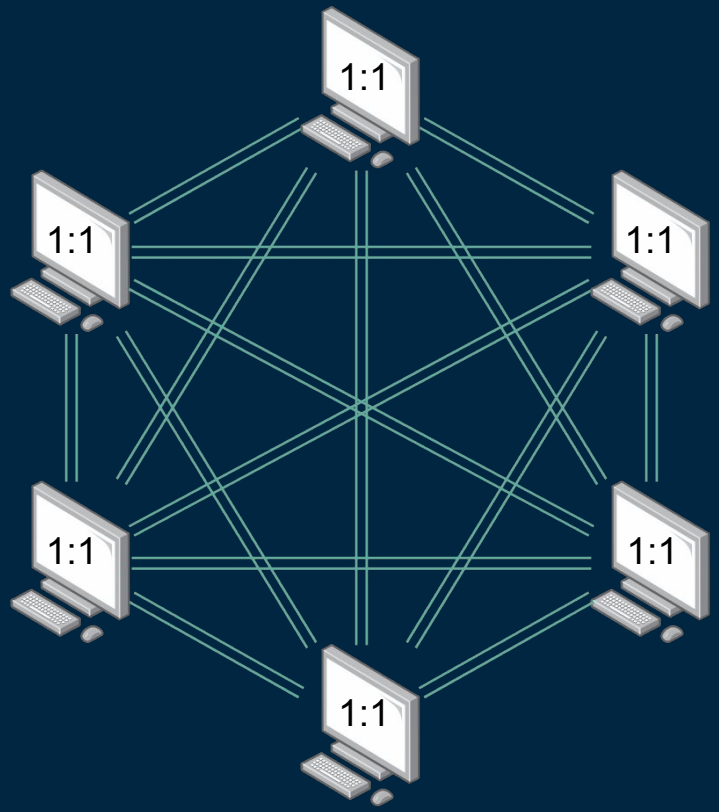
JIBE.COMPANY

Approach: Master Node



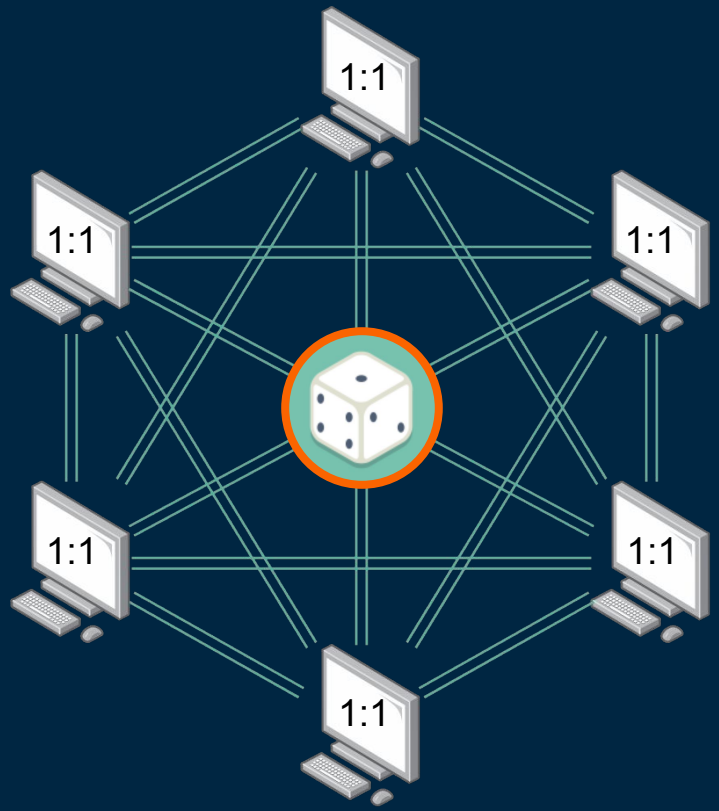
JIBE.COMPANY

Approach: Master Node

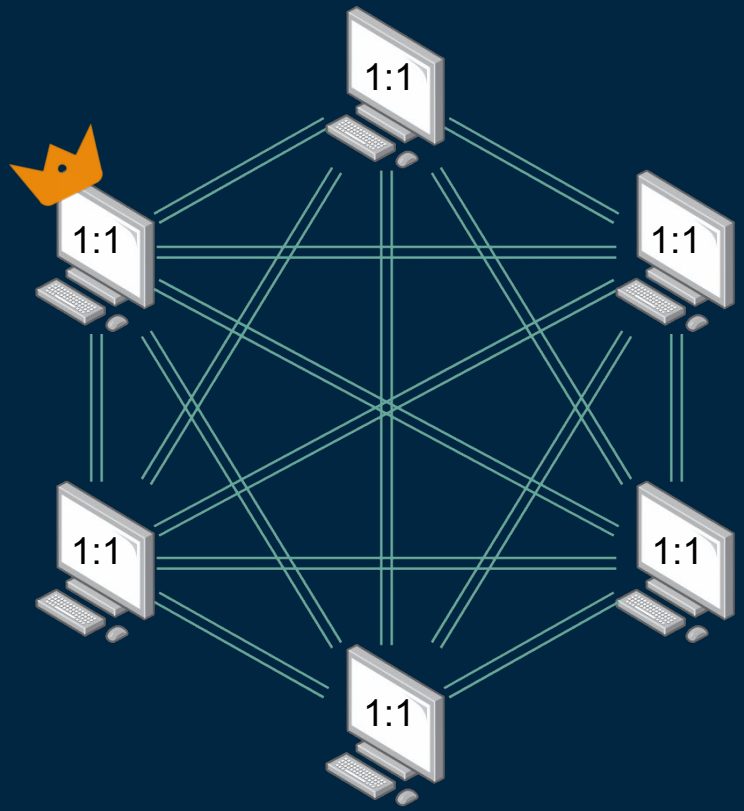


JIBE.COMPANY

Approach: Master Node



Approach: Master Node



JIBE.COMPANY

Recap

- ➔ Partial mesh broadcast
- ➔ Transaction log
- ➔ Master node

JIBE.COMPANY

Recap

It’s better, but...

JIBE.COMPANY

Recap

Everybody has to play by the rules

JIBE.COMPANY

Crash Fault Tolerance

We have only achieved crash fault tolerance

JIBE.COMPANY

Crash Fault Tolerance

This consensus algorithm is CFT

Crash Fault Tolerance

- ➔ Paxos (1989)
- ➔ Raft (2013)

JIBE.COMPANY

Byzantine Fault Tolerance

What if not everybody plays by the rules?

JIBE.COMPANY

Byzantine Fault Tolerance

- ➔ We can’t trust all messages we receive
- ➔ We can’t trust all other nodes
- ➔ We still want to reach consensus
- ➔ We still want to remain crash fault tolerant

Blockchain

JIBE.COMPANY

Blockchain

Let’s focus on Bitcoin for now

JIBE.COMPANY

Blockchain

Bitcoin is a byzantine fault tolerant distributed state machine

JIBE.COMPANY

Blockchain

Bitcoin’s consensus algorithm is BFT

JIBE.COMPANY

Step 1: Identity

Every user on the network has a private-public keypair

JIBE.COMPANY

Step 1: Identity

Every user on the network is identified by their public key; their Address

JIBE.COMPANY

Step 1: Identity

Every user on the network keeps their private key safe

JIBE.COMPANY

Step 1: Identity

A public key is a pseudonym; a “random” 256-bit number

JIBE.COMPANY

Step 1: Identity

A public key can be used to verify signatures made using the corresponding private key

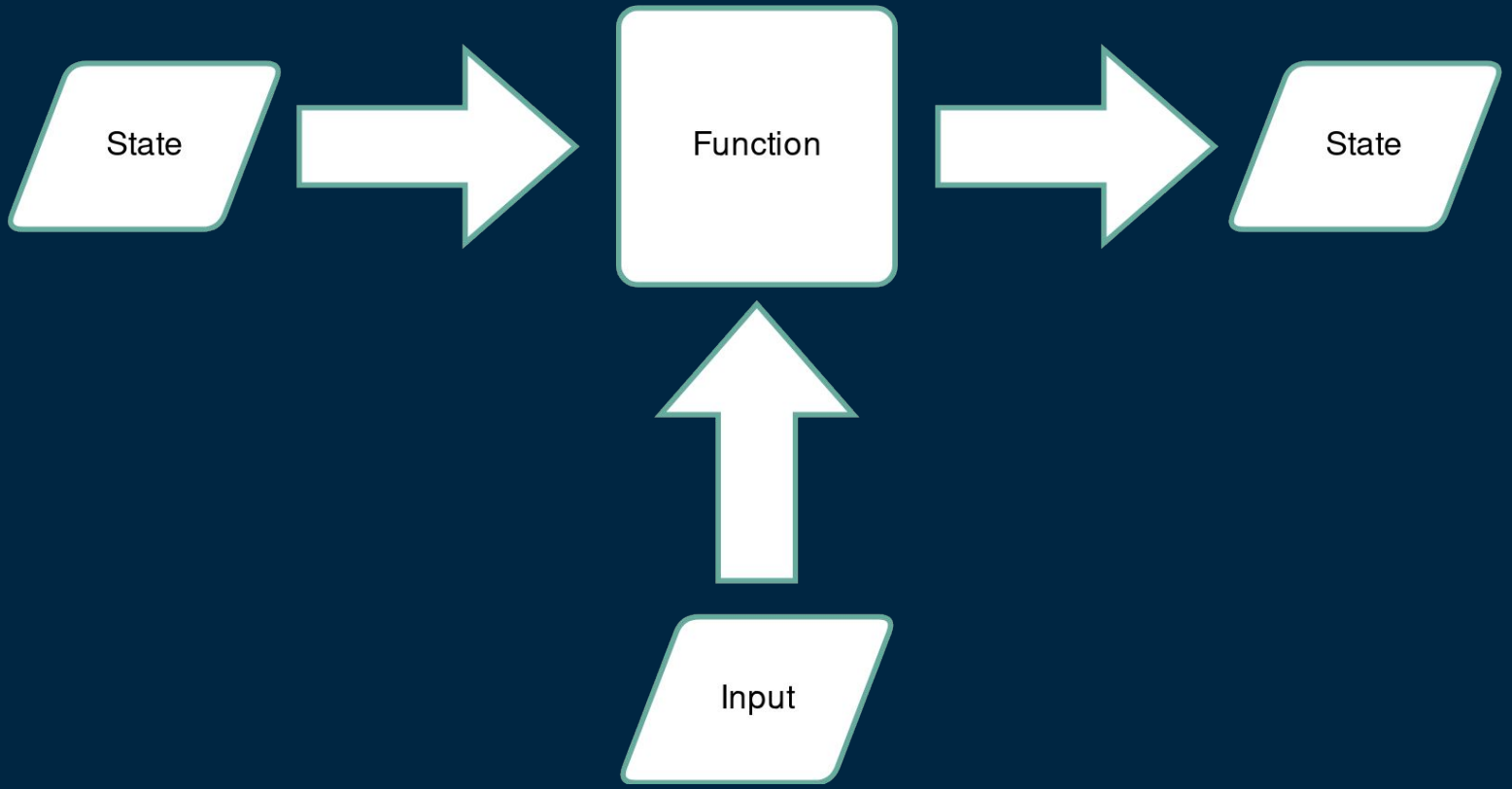
JIBE.COMPANY

Step 2: Transaction

Sign the input of the state transition function and broadcast it

JIBE.COMPANY

State Transition Function



JIBE.COMPANY

State

Bob’s Address: 2.3 BTC

Alice’s Address: 0.4 BTC

Eve’s Address: 567 BTC

...etc

JIBE.COMPANY

Step 2: Transaction

For instance, “move 1 BTC from me to Bob’s Address”

JIBE.COMPANY

Step 3: Transaction Verification

Every node receives the transaction and verifies it

JIBE.COMPANY

Step 3: Transaction Verification

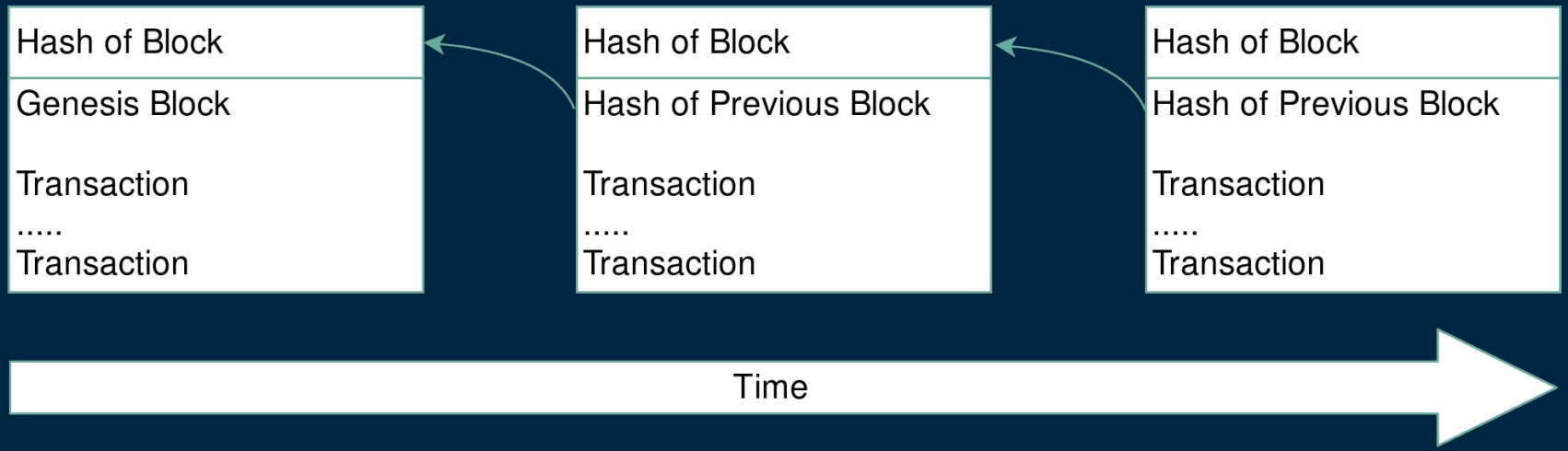
- ➔ Is the signature correct?
- ➔ Does the sender have enough bitcoin?

Step 3: Transaction Verification

Discard all unverified transactions

JIBE.COMPANY

Step 4: Block Creation



Cryptographic Hash

- ➔ **hello**
  `2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824`
- ➔ **Hello**
  `185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969`

JIBE.COMPANY

Step 4: Block Creation

The blockchain is an immutable, append-only data structure

JIBE.COMPANY

Step 4: Block Creation

Who gets to add the block to the chain? Who gets to be the master?

JIBE.COMPANY

Step 5: Proof of Work

A computationally intensive puzzle, of which the answer is easy to verify

Step 5: Proof of Work

On average, only one node comes up with the correct answer every 10 minutes

JIBE.COMPANY

Step 5: Proof of Work

That node gets a reward in BTC

Step 5: Proof of Work

- ➔ Fairly select a master node for each iteration of the blockchain
- ➔ Protect against denial-of-service
- ➔ Incentivize the stewardship of the network

Step 5: Proof of Work

- ➔ Slow transaction completion
- ➔ Huge amount of power (ab)used

JIBE.COMPANY

Step 6: Block Distribution

The puzzle winner distributes the block to the network

Step 6: Block Distribution

The nodes

- ➔ Verify the block
- ➔ Verify the puzzle answer
- ➔ Verify the transactions
- ➔ Execute the verified transactions, updating the state

Recap

```
10 IDENTITY
20 TRANSACTION
30 TRANSACTION VERIFICATION
40 BLOCK CREATION
50 PROOF OF WORK
60 BLOCK DISTRIBUTION
70 GOTO 20
```

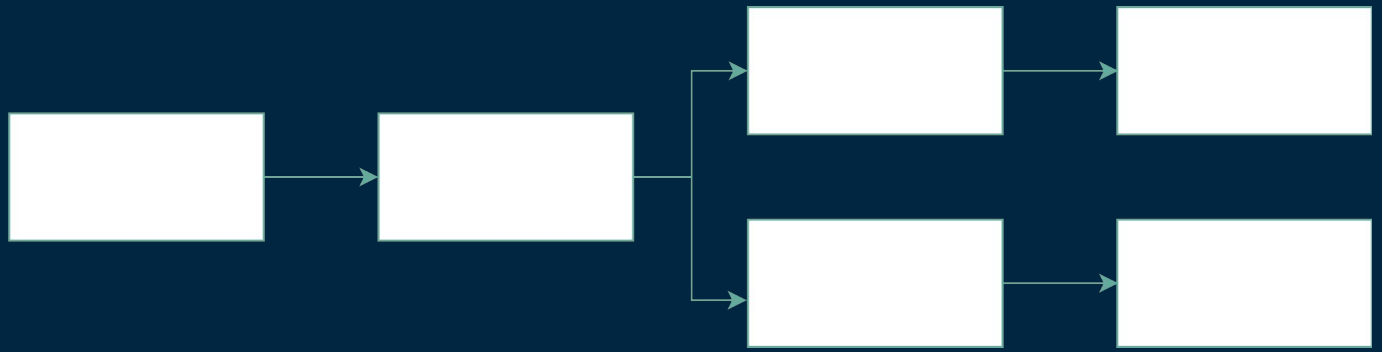
JIBE.COMPANY

Crash Fault Tolerance

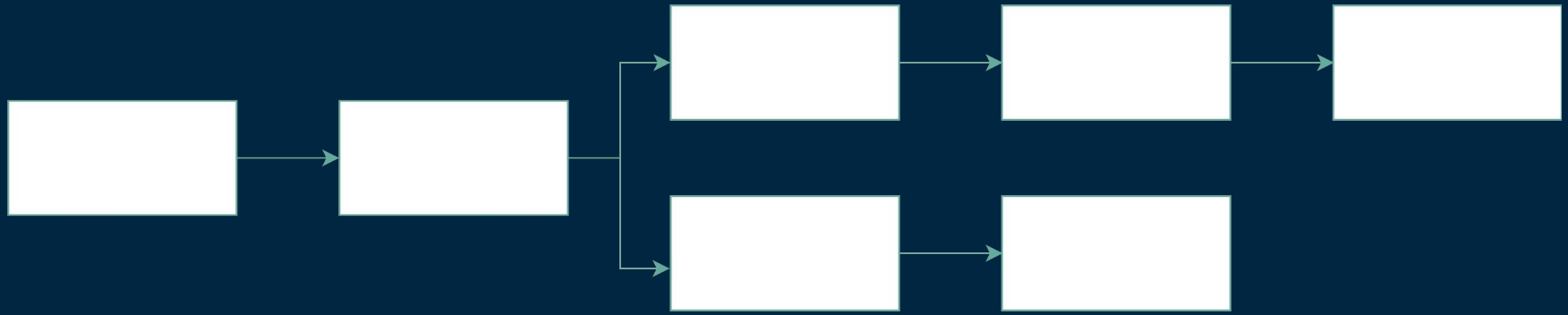
What if the network becomes split?

JIBE.COMPANY

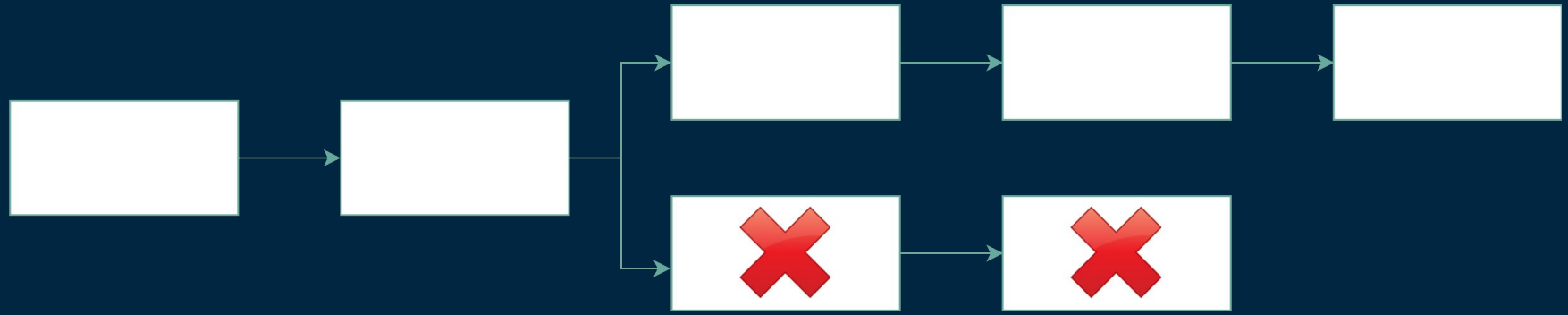
Crash Fault Tolerance



Crash Fault Tolerance



Crash Fault Tolerance



JIBE.COMPANY

Smart Contracts

JIBE.COMPANY

Smart Contracts

- ➔ User-configurable state transition functions
- ➔ Deployed using a transaction
- ➔ Have their own Address

Blockchain

- ➔ Distributed state machine
- ➔ Byzantine fault tolerant
- ➔ Immutable transaction history
- ➔ Shared transactions and state on all nodes

- ➔ Pseudonymous transparency
- ➔ Relatively slow
- ➔ Relatively expensive

JIBE.COMPANY

Case study

JIBE.COMPANY

POSEIDON

JIBE.COMPANY

PoSeID-on

Protection and control
Of
Secured
Information by means of a privacy enhanced
Dashboard

JIBE.COMPANY

PoSeID-on

- ➔ Give organizations the means to comply with the GDPR
- ➔ Give users the means to exercise their GDPR-derived rights

JIBE.COMPANY

GDPR in a Nutshell: Organization

- ➔ Only ask for personal information when you really need it
- ➔ Explain why you need that information and what you intend to do with it
- ➔ Explicitly get permission from the person to whom that information belongs
- ➔ Keep that personal information secure at all times
- ➔ Be transparent about what information you have on someone
- ➔ Don’t keep personal information if you don’t need it anymore
- ➔ Make it possible for someone to move their information to a competitor

GDPR in a Nutshell: User

- ➔ You know what personal information an organization has about you
- ➔ What they do with it and for what purpose they have it
- ➔ With your explicit permission
- ➔ You can update your information at any time
- ➔ You can request deletion of your information at any time*
- ➔ You can move your information to a competitor

PoSeID-on

PoSeID-on will develop and deliver an **innovative intrinsically scalable platform**, as an integrated and comprehensive solution aimed to **safeguard the rights of data subjects**, exploiting the cutting-edge technologies of **Smart Contracts and Blockchain**, as well as support organizations in data management and processing while **ensuring GDPR compliance**.

JIBE.COMPANY


HORIZON 2020


J I B E


accenture


SANTANDER
CIUDAD


tecnalia
Inspiring
Business


MEF
Ministero
dell'Economia
e delle Finanze


SOFTEAM


U
C
UNIVERSIDADE DE COIMBRA


mita
MALTA INFORMATION TECHNOLOGY AGENCY


PNO


e-lex
STUDIO LEGALE

PoSeID-on

https://www.poseidon-h2020.eu

JIBE.COMPANY

One-stop shop for all personal information

JIBE.COMPANY

Challenges

- ➔ Accessibility
- ➔ Security
- ➔ Scalability
- ➔ Ease of use
- ➔ Budget

JIBE.COMPANY

Accessibility

- ➔ All European citizens
- ➔ Regardless of technological literacy
- ➔ Regardless of disabilities
- ➔ Regardless of device

Accessibility

➔ Web-based Dashboard

JIBE.COMPANY

Security

- ➔ Strong authentication (eIDAS)
- ➔ Strong end-to-end encryption
- ➔ No central authority with access to all personal information

JIBE.COMPANY

Security

- ➔ No central repository with personal information
- ➔ Facilitate a secure conduit between organizations to transport information
- ➔ While managing and checking access permissions

JIBE.COMPANY

Scalability

➔ Scale from a few hundred users to hundreds of millions while staying adequately performant

JIBE.COMPANY

Ease of use

- ➔ Easy for users to understand and use
- ➔ Easy for organizations to integrate with

JIBE.COMPANY

Budget

- ➔ Limited timeframe
- ➔ Limited EU-given budget

JIBE.COMPANY

Budget

➔ Use as many pre-existing (open-source) solutions as possible

JIBE.COMPANY

Blockchain

➔ PoSeID-on is a research project

Blockchain

➔ Exchange personal information using blockchain

JIBE.COMPANY

Blockchain

- ~~Exchange personal information using blockchain~~

Blockchain

- ~~Exchange personal information using blockchain~~

- Blockchain is immutable
- Blockchain has a shared state

JIBE.COMPANY

Blockchain

➔ Exchange encrypted personal information using blockchain

JIBE.COMPANY

Blockchain

- ~~Exchange encrypted personal information using blockchain~~

JIBE.COMPANY

Blockchain

- ~~Exchange encrypted personal information using blockchain~~

- Encryption will fail over time
- Blockchain is append-only

Blockchain

➔ Exchange permissions over personal information using blockchain

JIBE.COMPANY

Blockchain

- ➔ Exchange permissions over personal information using blockchain`

- ➔ Organization A has the right to have info B from user C
- ➔ Organization D has the right to read info B from user C from organization A

Blockchain

- ➔ Exchange permissions over personal information using blockchain

- ➔ Asking for permissions
- ➔ Giving permissions
- ➔ Checking permissions
- ➔ Revoking permissions

Blockchain

➔ Smart contract for permission management

Blockchain

Person	Recipient	Sender	Data	Permission
Alice	Bank	Government	Passport details	Requested
Bob	Car insurance	Car company	Location data	Given
Eve	Hospital	-	Weight	Given

Blockchain

Exchange permissions over personal information using a smart contract

1. Bank asks Alice for access to her passport details
2. Alice gives explicit permission
3. Bank asks the government for Alice’s passport details
4. The government validates the given permission
5. The government supplies Bank with a copy of Alice’s passport

JIBE.COMPANY

Blockchain

Person	Recipient	Sender	Data	Permission
Alice	Bank	Government	Passport details	Requested
Bob	Car insurance	Car company	Location data	Given
Eve	Hospital	-	Weight	Given

JIBE.COMPANY

Blockchain

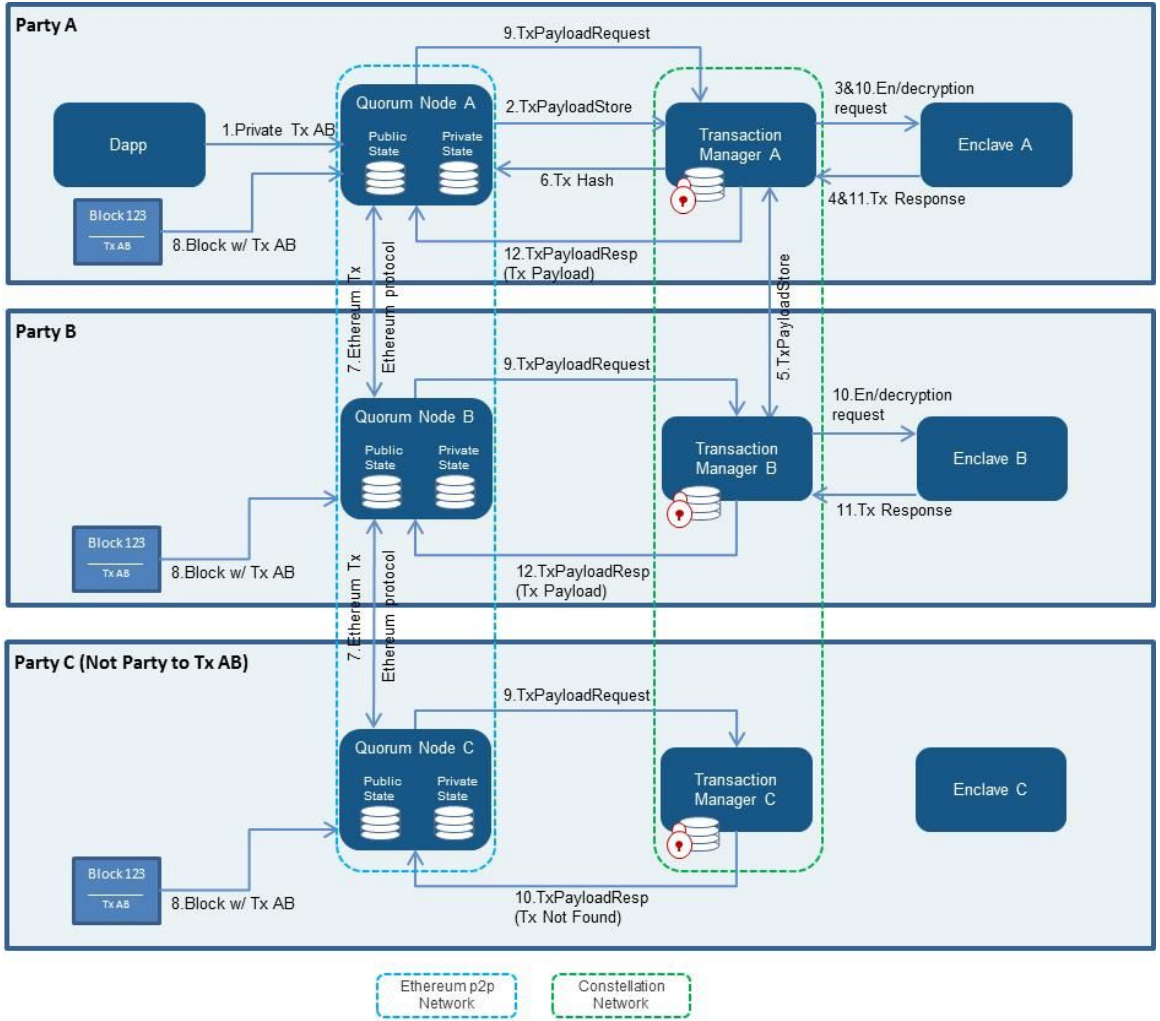
Even relationships are potentially sensitive personal information

JIBE.COMPANY

Blockchain

- ➔ Use Quorum by JP Morgan Chase
- ➔ Use burnable pseudo-identities

JIBE.COMPANY


Party A
9.TxPayloadRequest
Quorum Node A
Public State
Private State
2.TxPayloadStore
Transaction Manager A
3&10.En/decryption request
Enclave A
Dapp
1.Private Tx AB
6.Tx Hash
4&11.Tx Response
Block 123
Tx AB
8.Block w/ Tx AB
12.TxPayloadResp (Tx Payload)
7.Ethereum Tx
Ethereum protocol
5.TxPayloadStore
Party B
9.TxPayloadRequest
Quorum Node B
Public State
Private State
Transaction Manager B
10.En/decryption request
Enclave B
11.Tx Response
Block 123
Tx AB
8.Block w/ Tx AB
12.TxPayloadResp (Tx Payload)
7.Ethereum Tx
Ethereum protocol
Party C (Not Party to Tx AB)
9.TxPayloadRequest
Quorum Node C
Public State
Private State
Transaction Manager C
Enclave C
Block 123
Tx AB
8.Block w/ Tx AB
10.TxPayloadResp (Tx Not Found)
Ethereum p2p Network
Constellation Network

Burnable pseudo-identities

- ➔ Keep a list of identities (Addresses) for each user
- ➔ Use a new Address every transaction/hour/day
- ➔ Remove the association when the user wants to be forgotten

PoSeID-on Blockchain

- ➔ Only exchanges permissions using blockchain
- ➔ Uses a permissioned blockchain network
- ➔ Uses Quorum to further separate state
- ➔ Uses burnable pseudo-identities to further protect identities

PoSeID-on Blockchain

➔ There is still room for improvement

Room for Improvement

➔ A web-based dashboard introduces a central, trusted authority

JIBE.COMPANY

Room for Improvement

➔ Burnable identities still need a central, trusted, correlation database

JIBE.COMPANY

Room for Improvement

➔ PoSeID-on is a very capable solution; a big step in the right direction

JIBE.COMPANY

Blockchain Applicability

JIBE.COMPANY

Questions to ask yourself

1. Do you have a distributed state machine?
2. Does it need to be crash fault tolerant?
3. Does it need to be byzantine fault tolerant?
4. Does it have shared state on all nodes?
5. Are all parties involved capable of running a node?
6. Do all nodes have adequate processing/storage capacity?
7. Are you okay with relatively slow transactions?

Questions to ask yourself

1. Do you have a distributed state machine? ✅
2. Does it need to be crash fault tolerant? ✅
3. Does it need to be byzantine fault tolerant? ✅
4. Does it have shared state on all nodes? ❌
5. Are all parties involved capable of running a node? ❌
6. Do all nodes have adequate processing/storage capacity? ✅
7. Are you okay with relatively slow transactions? ❌

Possible Alternatives

JIBE.COMPANY

Possible Alternatives

- ➔ “Classic” authenticated encrypted message exchange
- ➔ Distributed multi-party signatures
- ➔ Distributed zero-knowledge proofs
- ➔ DHT using homomorphic encryption

Are you interested?

JIBE.COMPANY

Join our team!

Jibe.Company

Woenselsestraat 350
5623 EG Eindhoven
The Netherlands

+31 40 767 6001

hello@jibecompany.com

Spyhce

Calea Moților no. 28
Cluj Napoca

+31 77 8080 140 (NL, DE, EN)
+49 211 21070426 (DE)

hello@spyhce.com

JIBE.COMPANY