Private Blockchain

Ethereum, Corda, Hyperledger Fabric

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

What's the plan for this lecture?

1.
What is private blockchain?

Problems of public blockchain?

Energy consuming and slow
(consensus mechanism)

Availability and transparency of data: all data are replicated at every node and available to every node
(even online to everybody!)

Based on “anonymity” of the nodes
(we know addresses, but not the identity of the owners)

Marco Comuzzi mcomuzzi@unistac.kr

Private blockchain

A blockchain system in which the nodes belong to a private Internet network
(= nodes must be admitted to this network)

Private blockchain

The nodes must be vetted before being admitted to a private blockchain, therefore…

The identity of the nodes must be known.

It is possible to segregate the information stored in the ledger among different groups of nodes (based on their identity) as required by the specific business settings.

The consensus mechanism need not be based on highly resource-consuming algorithms (off- or on-chain), but, exploiting the usually limited size of the network and the fact the identity of the nodes is known, it can rely on more lightweight solutions that would speed up the overall system performance while not compromising its security.

Marco Comuzzi mcomuzzi@unistac.kr

2.
Private instance of Ethereum

Private Ethereum

The Ethereum client can be downloaded on the nodes of a private network of computers

Nodes of a (private) LAN

Nodes of a VPN

The result is a private blockchain

Marco Comuzzi mcomuzzi@unistac.kr

Private Ethereum: configuration

Ether on a private network does not have value in fiat currency

Nodes can be assigned Ether when the private network is created
Gas price can be kept low

Consensus mechanism can be configured to execute much more quickly

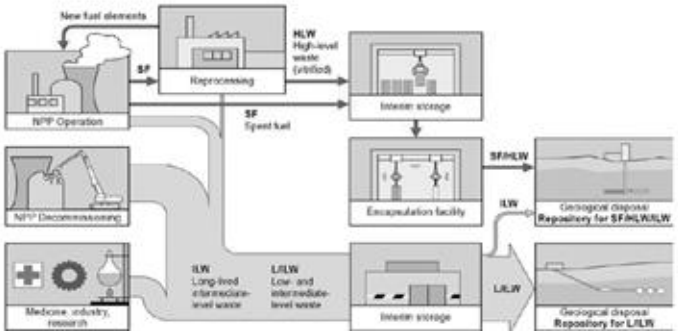
Example: see next week's talk

Marco Comuzzi mcomuzzi@unistac.kr

Next week's talk

Speaker: two UNIST undergrad students ☺

Blockchain for radioactive waste management



Marco Comuzzi mcomuzzi@unistac.kr

3.
Corda

What is Corda?

A framework for developing private blockchain systems [Corda Systems]

Open source since 2016

Gained traction in the financial industry

https://www.r3.com/products/corda/

Limitations of public blockchain

Same data replicated at all nodes. Only "some" data are relevant for a node.

All the data in the ledger are public, can be accessed by all nodes (and anybody online…)

OK if nodes are anonymous, but not if identity of nodes is known

Marco Comuzzi mcomuzzi@unistac.kr

Corda

A framework to create private blockchain systems

Ledger created on a "need-to-know" basis

Generalize the Bitcoin's UTXO mechanism

Marco Comuzzi mcomuzzi@unistac.kr

Corda system: data structure

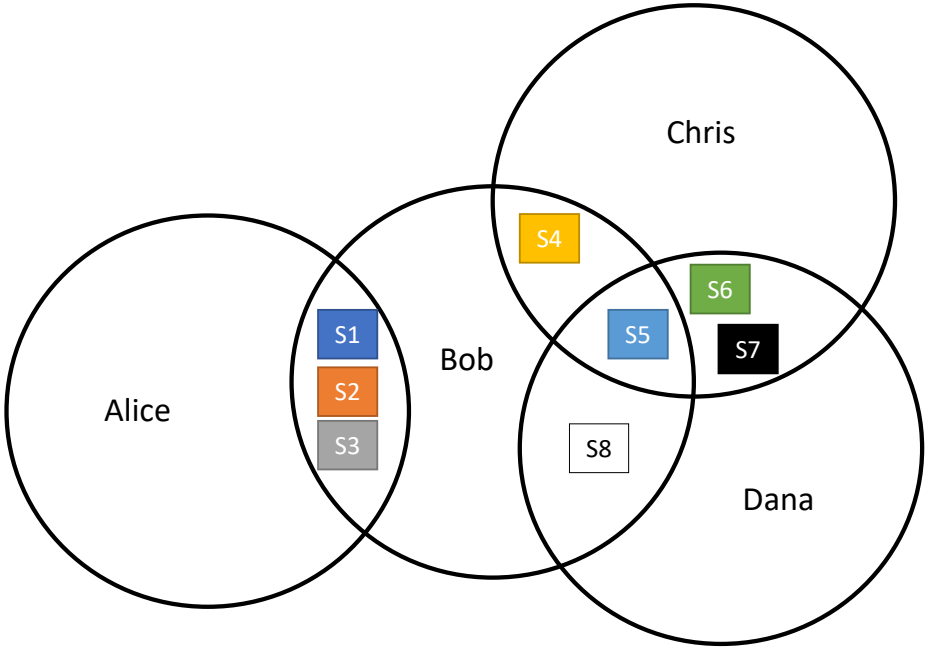
Ledger contains "states"

A "state" captures a fact shared by two or more nodes

Ex. "**Alice** owes 10$ to **Bob**",
"**Chris** is owner of car (**reg**. N.: XYZ) insured by **ACME** Ltd."

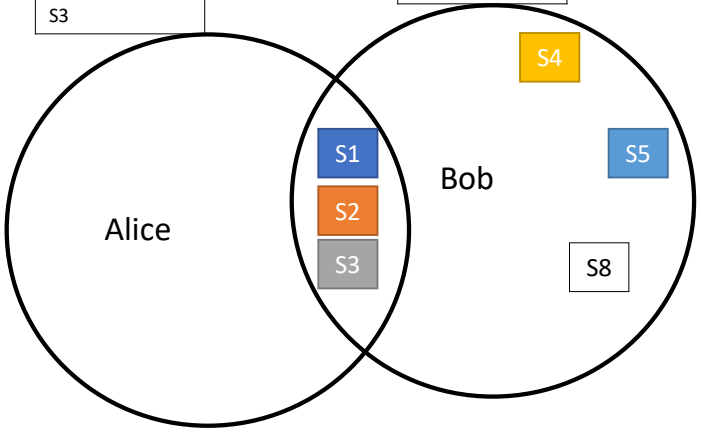
The ledger at a node contains only the facts relevant for that node ("need-to-know" basis)

Marco Comuzzi mcomuzzi@unistac.kr



Alice's Ledger
S1
S2
S3

Bob's Ledger
S1
S2
S3
S4
S5
S8



Corda system: transactions

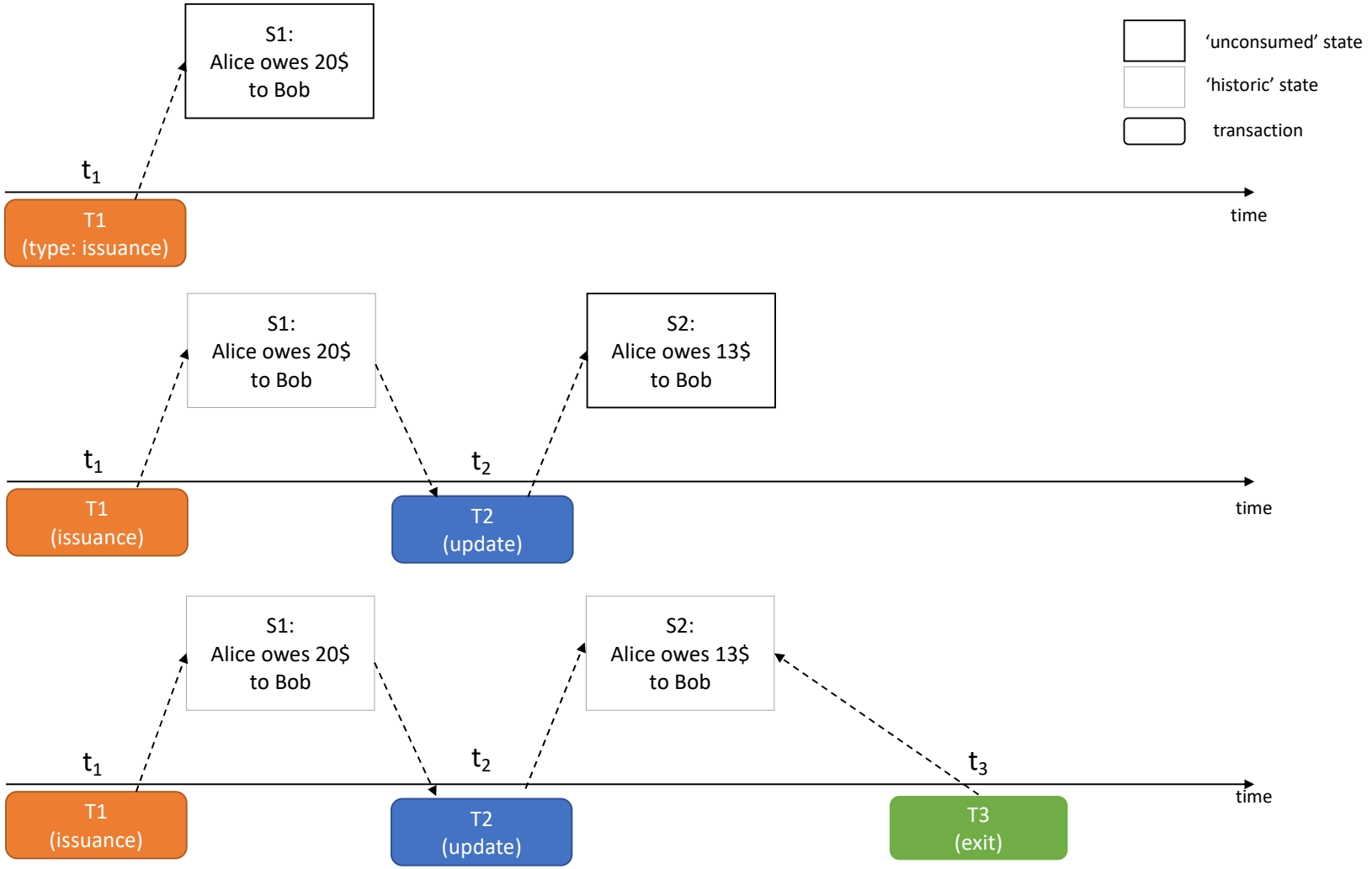
Transactions consume states and produce new ones.
Tx are digitally signed by the originator

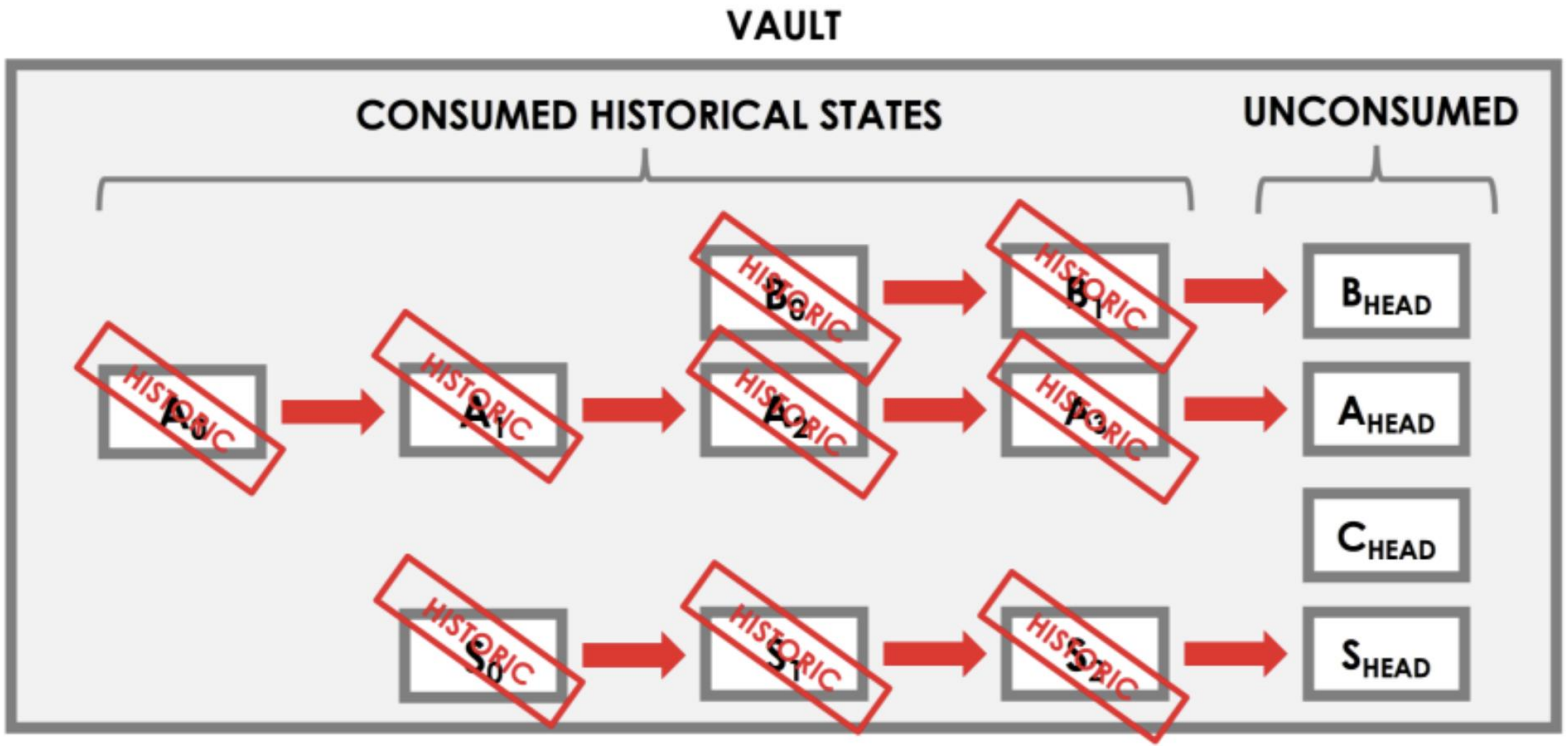
States consumed by transactions become "historic"
Transactions can only use unconsumed (= non historic) states as input

Issuance: produces new state(s), no input
Update: consume states and produce new ones
Exit: consume state(s), no output

Transactions in Corda extend the UTXO mechanism to generic "states"

Marco Comuzzi mcomuzzi@unistac.kr


S1:
Alice owes 20$
to Bob
'unconsumed' state
'historic' state
transaction
t_1
T1
(type: issuance)
time
S1:
Alice owes 20$
to Bob
S2:
Alice owes 13$
to Bob
t_1
t_2
T1
(issuance)
T2
(update)
time
S1:
Alice owes 20$
to Bob
S2:
Alice owes 13$
to Bob
t_1
t_2
t_3
T1
(issuance)
T2
(update)
T3
(exit)
time



Marco Comuzzi mcomuzzi@unistac.kr

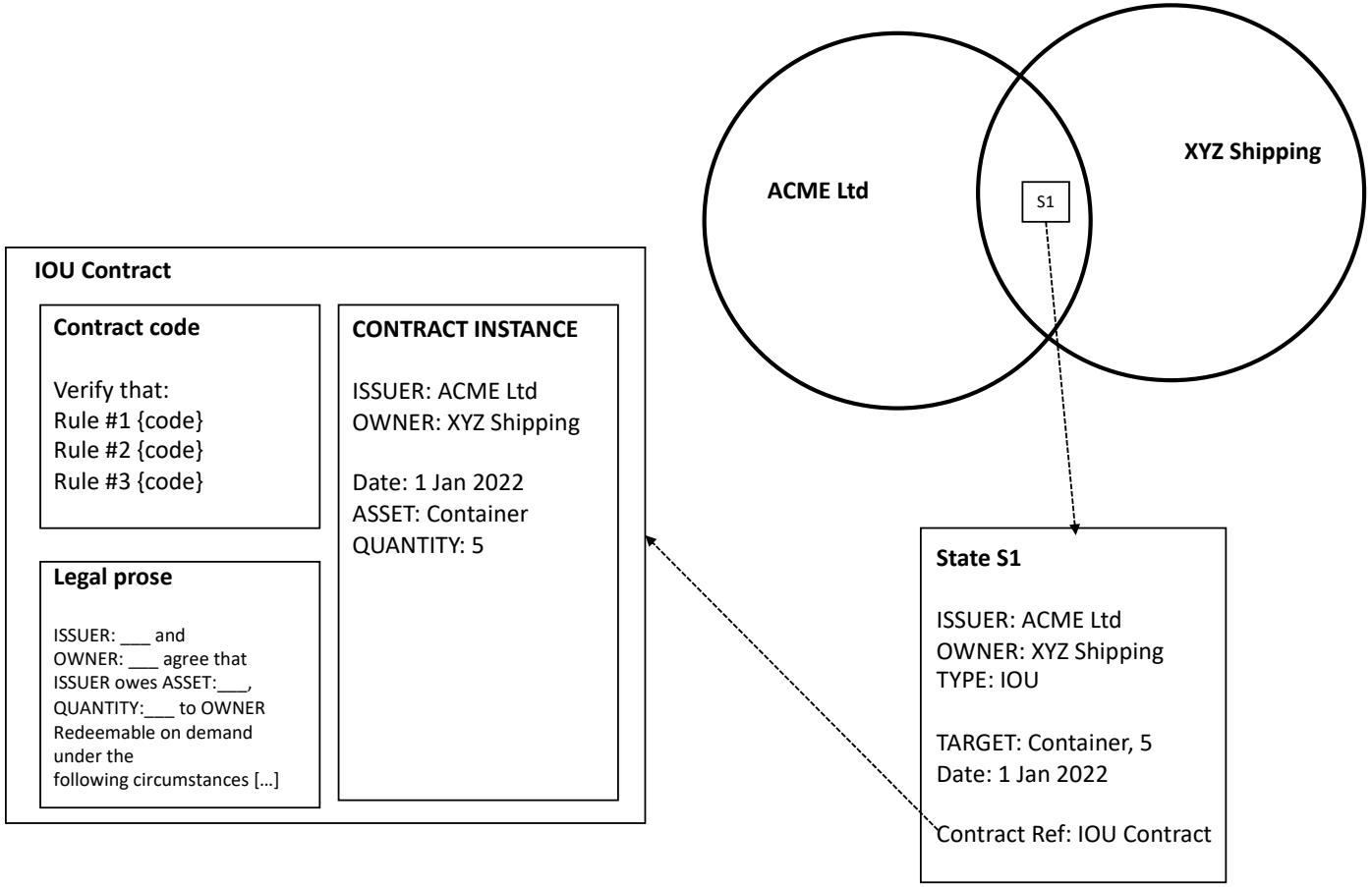
Corda “contract”

A contract in Corda specifies the validation rules of a transaction or transaction type

Contracts can be associated with “legal prose”
(which in some countries can be used in a court)

Corda contracts are not smart contracts
(no business logic is actually executed)

Helpful and used in fintech applications


ACME Ltd
XYZ Shipping
S1
IOU Contract
Contract code
Verify that:
Rule #1 {code}
Rule #2 {code}
Rule #3 {code}
Legal prose
ISSUER: ___ and
OWNER: ___ agree that
ISSUER owes ASSET:___,
QUANTITY:___ to OWNER
Redeemable on demand
under the
following circumstances […]
CONTRACT INSTANCE
ISSUER: ACME Ltd
OWNER: XYZ Shipping
Date: 1 Jan 2022
ASSET: Container
QUANTITY: 5
State S1
ISSUER: ACME Ltd
OWNER: XYZ Shipping
TYPE: IOU
TARGET: Container, 5
Date: 1 Jan 2022
Contract Ref: IOU Contract

Consensus mechanism in Corda

Why a consensus mechanism?

We still need to fix a “double spending” problem

”States” could be double spent if nodes process transactions in a different order

Also, because of the “need-to-know”, not all transactions may reach all the nodes

Marco Comuzzi mcomuzzi@unistac.kr

Double spending a state: example

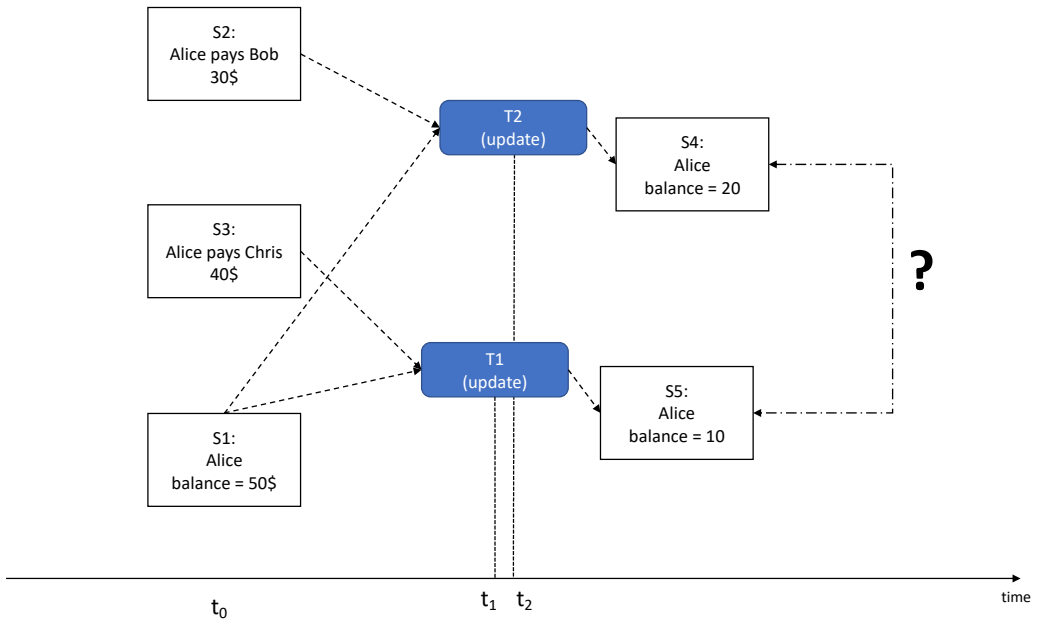
S3 is known to Alice and Chris

S2 known to Alice and Bob

S1 known to Alice (and a bank)

Alice may manage to consume S1 “twice”, creating inconsistent output states

(in this example, money is also spent twice!)



Marco Comuzzi mcomuzzi@unistac.kr

Validation and Consensus in Corda

When receiving a new transaction, a node must check that:

1. The transaction is valid
2. The state updates produced by the transaction are unique and consistent

Marco Comuzzi mcomuzzi@unistac.kr

Transaction validation in Corda

A node verifies that a transaction:

Is digitally signed

Is well-formed

Complies with all the contracts that it references

It consumes states that are not “historic”

Marco Comuzzi mcomuzzi@unistac.kr

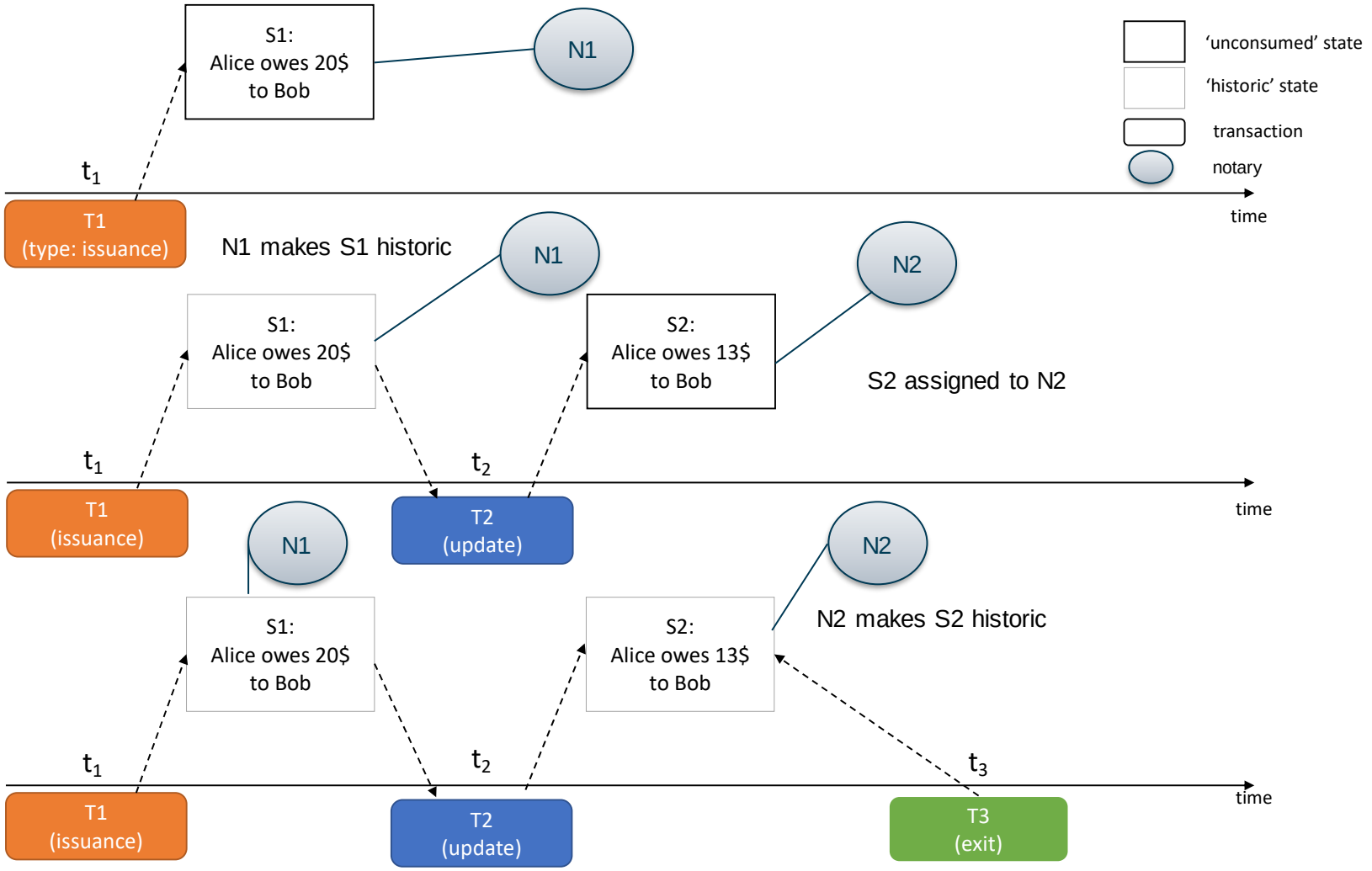
Consistent state updates in Corda

Achieved using “notary” nodes

Simple nodes that only monitor states
(do not even see the content of states, but only if they are historic or not)

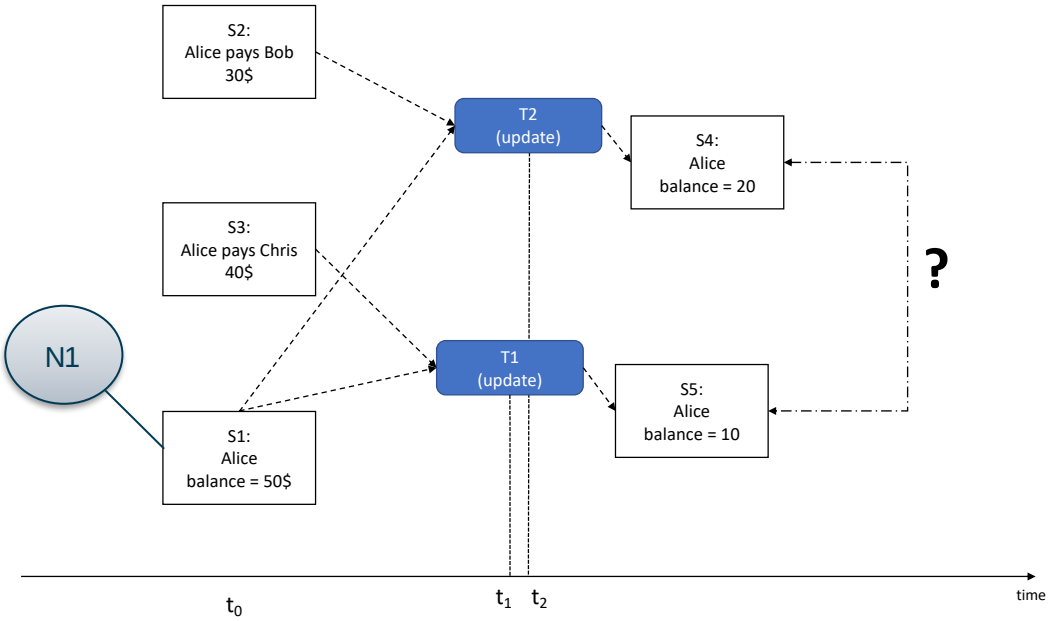
When a state is used as input of a transaction, the notary turns it into “historic” immediately

Nodes must check with the notary if they can use a state as transaction input


S1:
Alice owes 20$
to Bob
N1
'unconsumed' state
'historic' state
transaction
notary
t_1
T1
(type: issuance)
time
N1 makes S1 historic
N1
N2
S1:
Alice owes 20$
to Bob
S2:
Alice owes 13$
to Bob
S2 assigned to N2
t_1
t_2
T1
(issuance)
T2
(update)
time
N1
N2
S1:
Alice owes 20$
to Bob
S2:
Alice owes 13$
to Bob
N2 makes S2 historic
t_1
t_2
t_3
T1
(issuance)
T2
(update)
T3
(exit)
time

After T1 is executed, N1 will make S1 historic, so it cannot be used by T2

T2 will never be validated



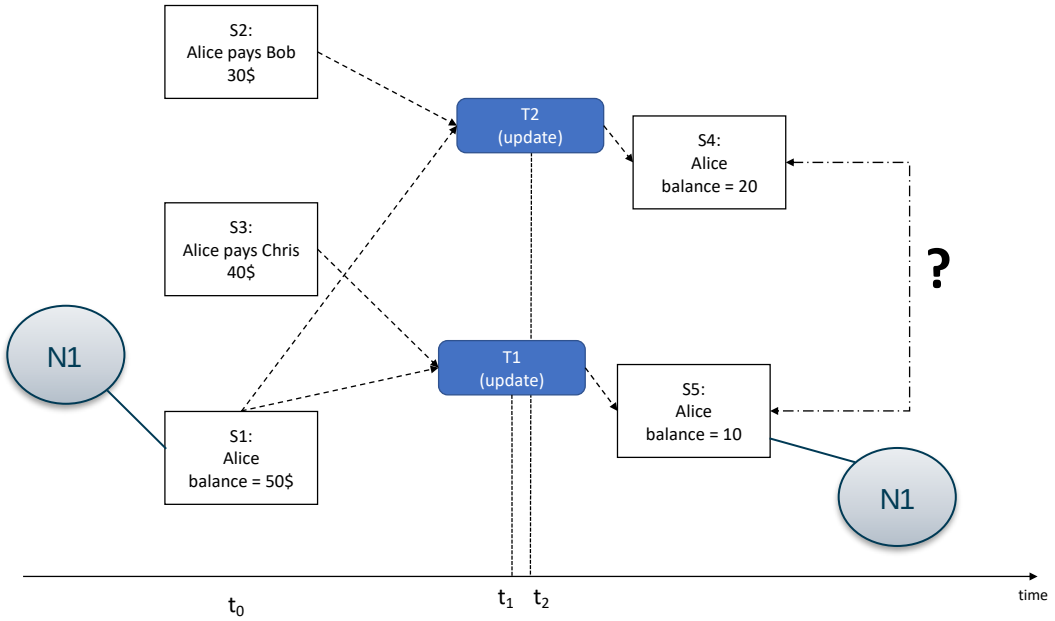
Marco Comuzzi mcomuzzi@unistac.kr

Problem with single notaries

Single notaries can be a "central authority" or a "single point of failure"

Let's assume that N1 is assigned to both S1 and S5:

N1 may collude with Alice to delay the state S1 change to "historic", so that also T2 can be validated, and then turn S5 to historic immediately, so that only S4 is valid (higher balance for Alice!)



Marco Comuzzi mcomuzzi@unistac.kr

Network of notaries

Instead of a single notary, use a network of notaries

The network of notaries runs a consensus mechanism among its nodes to agree on state changes

Consensus in private blockchain: see next lecture (probably next week)

4.
Hyperledger Fabric

HLF

A framework for developing private blockchain networks

Developed by the Hyperledger Foundation since 2016

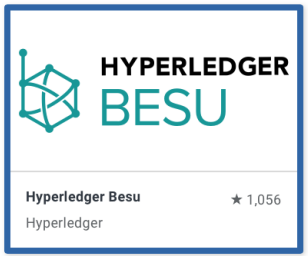
Foundational principles:

- "Need-to-know" ledger (like Corda)
- Smart contracts (unlike Corda)

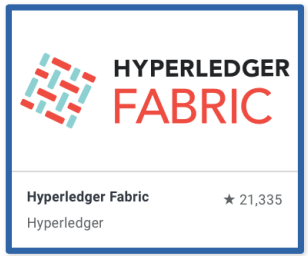
Marco Comuzzi mcomuzzi@unistac.kr

Graduated Hyperledger Projects (6)

Hyperledger Aries ★ 1,604
Hyperledger



Hyperledger Besu ★ 1,056
Hyperledger



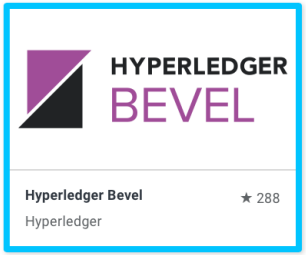
Hyperledger Fabric ★ 21,335
Hyperledger

Hyperledger Indy ★ 1,656
Hyperledger

Hyperledger Iroha ★ 693
Hyperledger

Hyperledger Sawtooth ★ 1,761
Hyperledger

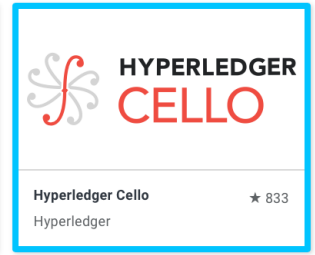
Incubating Hyperledger Projects (9)



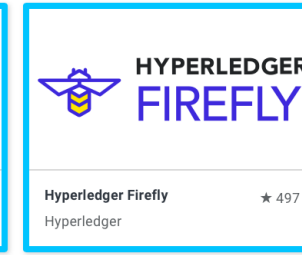
Hyperledger Bevel ★ 288
Hyperledger

Hyperledger Cacti ★ 242
Hyperledger

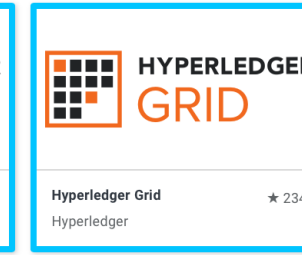
Hyperledger Caliper ★ 624
Hyperledger



Hyperledger Cello ★ 833
Hyperledger

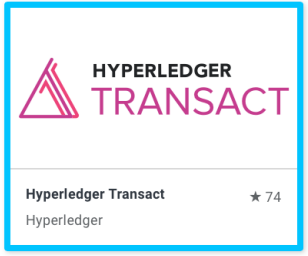


Hyperledger Firefly ★ 497
Hyperledger

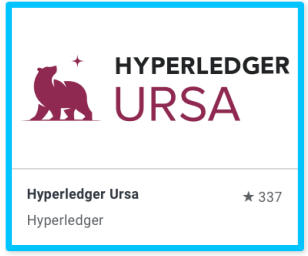


Hyperledger Grid ★ 234
Hyperledger

Hyperledger Solang ★ 894
Hyperledger



Hyperledger Transact ★ 74
Hyperledger



Hyperledger Ursa ★ 337
Hyperledger

Data in HLF

Information is represented by business objects, similarly to Enterprise Systems (ERP)

Business objects have attributes
<key, value>

The values represent the “state” of a business object

Ex.

```
Invoice(customer VAT number, items purchased,
quantity, price, discount, paid?)
```


Marco Comuzzi mcomuzzi@unistac.kr

Transactions in HLF

Transactions in HLF can:
create new business objects,
modify the state (value) of existing ones
delete objects.
(like Corda)

HLF includes smart contracts, called “chaincode”
(like Ethereum)

A transaction must always invoke one chaincode
(unlike Ethereum)

Marco Comuzzi mcomuzzi@unistac.kr

Ledger in HLF

World state

Key-value database that maintains the most up-to-date value of every business object.

Objective: quick access to business object values

Blockchain

Ordered collection of transactions grouped into cryptographically linked blocks

Objective: maintain the history of when/how/by whom business objects have been modified

Marco Comuzzi mcomuzzi@unistac.kr

“Need-to-know” basis in HLF

Channels

Single ledger, disconnected from others

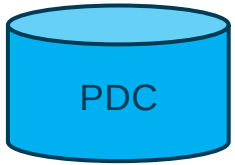
Has it own participants (nodes), world state and blockchain.

Transactions submitted to one channel are not visible on other channels

Private Data Collection

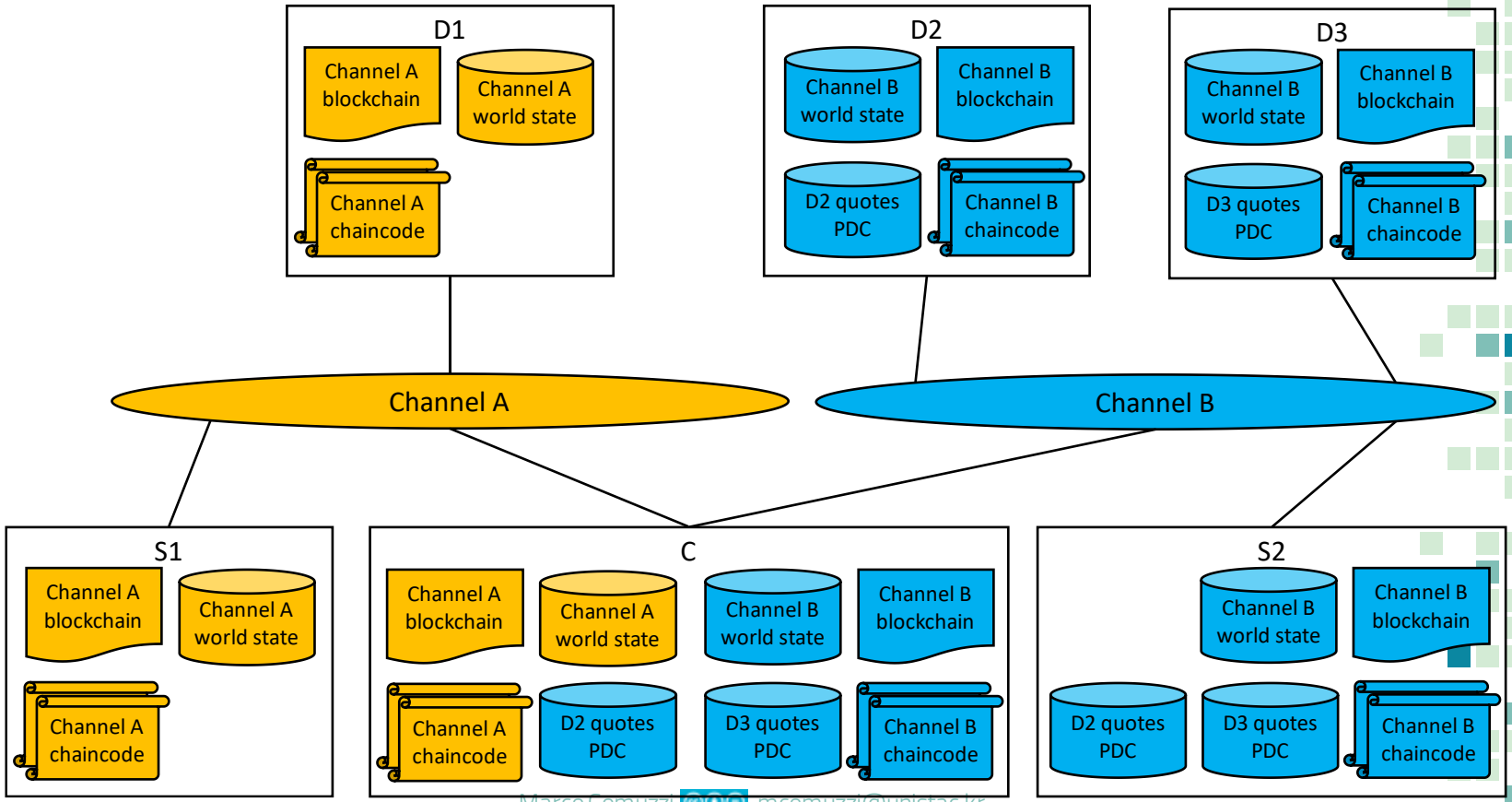
Collection of business objects in a side database replicated at each node using “off-chain” mechanisms (hash digests)

Address limitations of channels



Marco Comuzzi mcomuzzi@unistac.kr

HLF: Example


D1
Channel A blockchain
Channel A world state
Channel A chaincode
D2
Channel B world state
Channel B blockchain
D2 quotes PDC
Channel B chaincode
D3
Channel B world state
Channel B blockchain
D3 quotes PDC
Channel B chaincode
Channel A
Channel B
S1
Channel A blockchain
Channel A world state
Channel A chaincode
C
Channel A blockchain
Channel A world state
Channel B world state
Channel B blockchain
Channel A chaincode
D2 quotes PDC
D3 quotes PDC
Channel B chaincode
S2
Channel B world state
Channel B blockchain
D2 quotes PDC
D3 quotes PDC
Channel B chaincode

Transaction lifecycle and consensus mechanism in Ethereum

Ethereum consensus is **"validate-order-execute"**: Transactions are…

Validated during gossiping

Ordered into blocks by validators (miners)

Executed by the nodes when a new block is received

Transaction execution at each node must lead to the same state updates

>> No random logic allowed in smart contracts

Marco Comuzzi mcomuzzi@unistac.kr

Transaction lifecycle and consensus mechanism in HLF

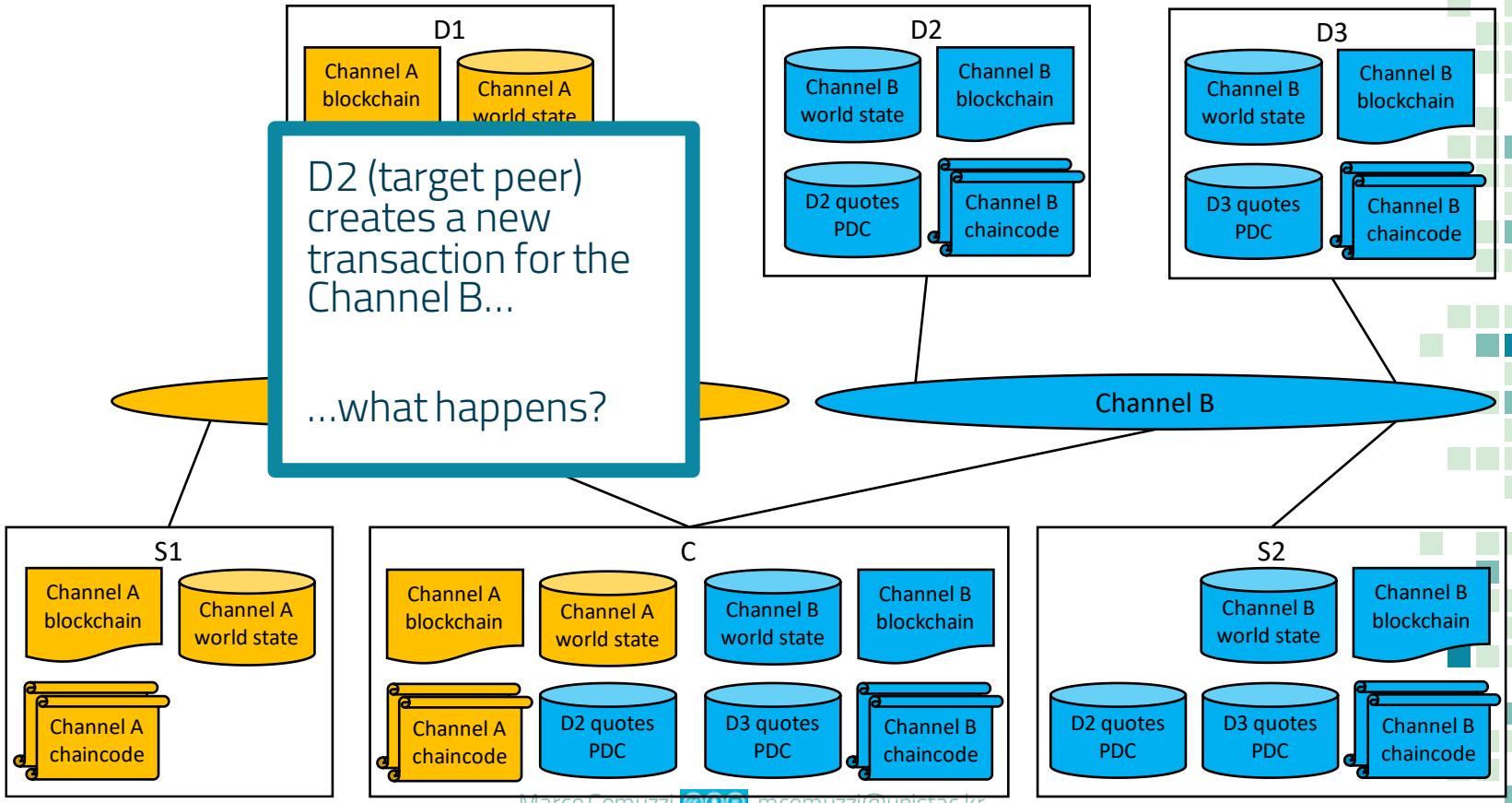
In HLF consensus is **“execute-order-validate”**

Transactions are first executed by a subset of the nodes (endorsing peers), then ordered into blocks (by ordering peers)

Blocks are distributed and the transactions in them are validate by every node. Validation implies that all the state changes entailed by the transactions are recorded in the world state.

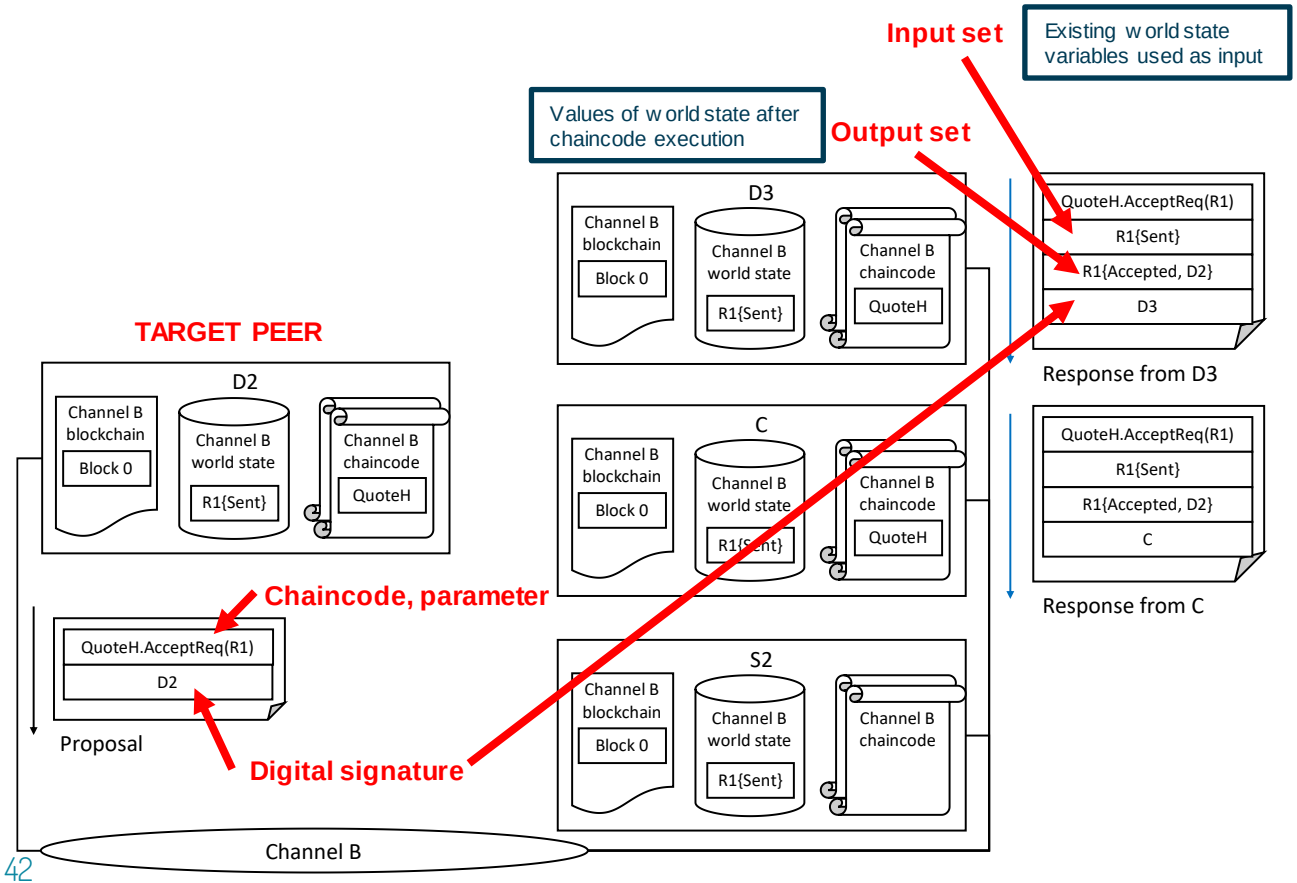
This avoids the creation of “forks” in the blockchain and, consequently, the need for PoW or PoS

Transaction lifecycle in HLF: Example



Marco Comuzzi mcomuzzi@unistac.kr

Execute-Order-Validate



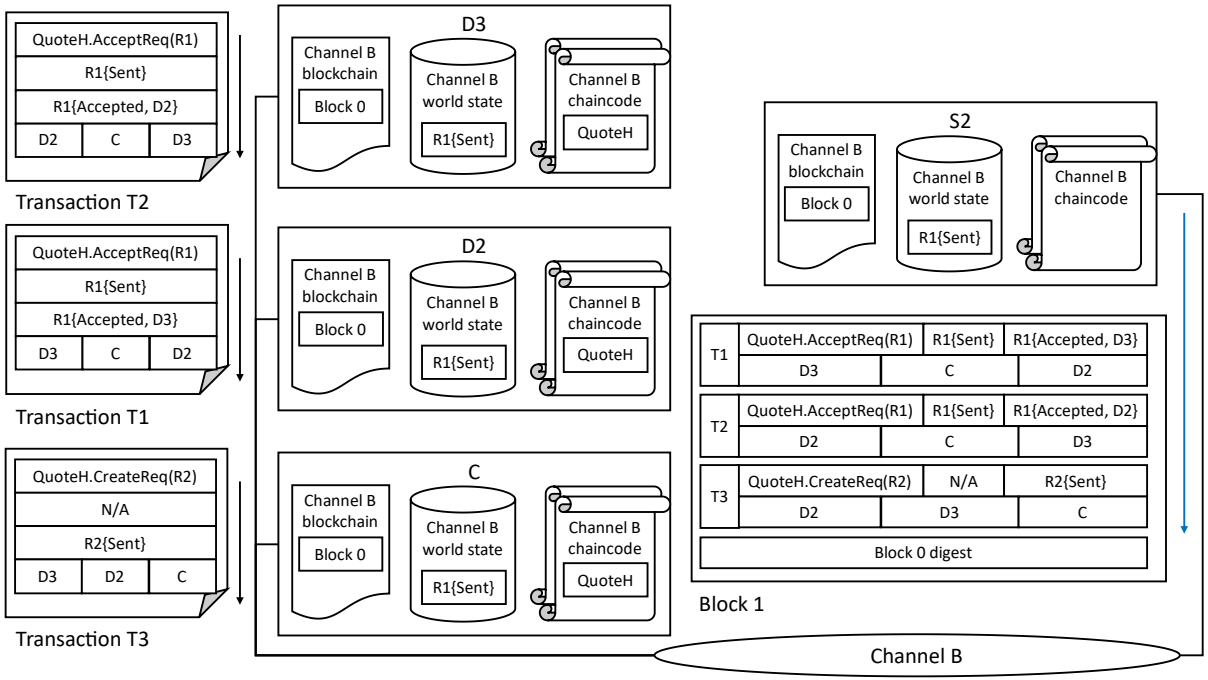
A target peer create a transaction proposal, signs it, and sends it to "endorsing peers"

Proposal contains: which chaincode to invoke and related parameters

Endorsing peers **execute** the transaction, calculating input and output set. If execution makes sense, endorsing peers return a proposal response to the target peer

Marco Comuzzi mcomuzzi@unistac.kr

Execute-Order-Validate



If target peer receives enough equal endorsements (based on the policy), they send a transaction to the ordering service(s).

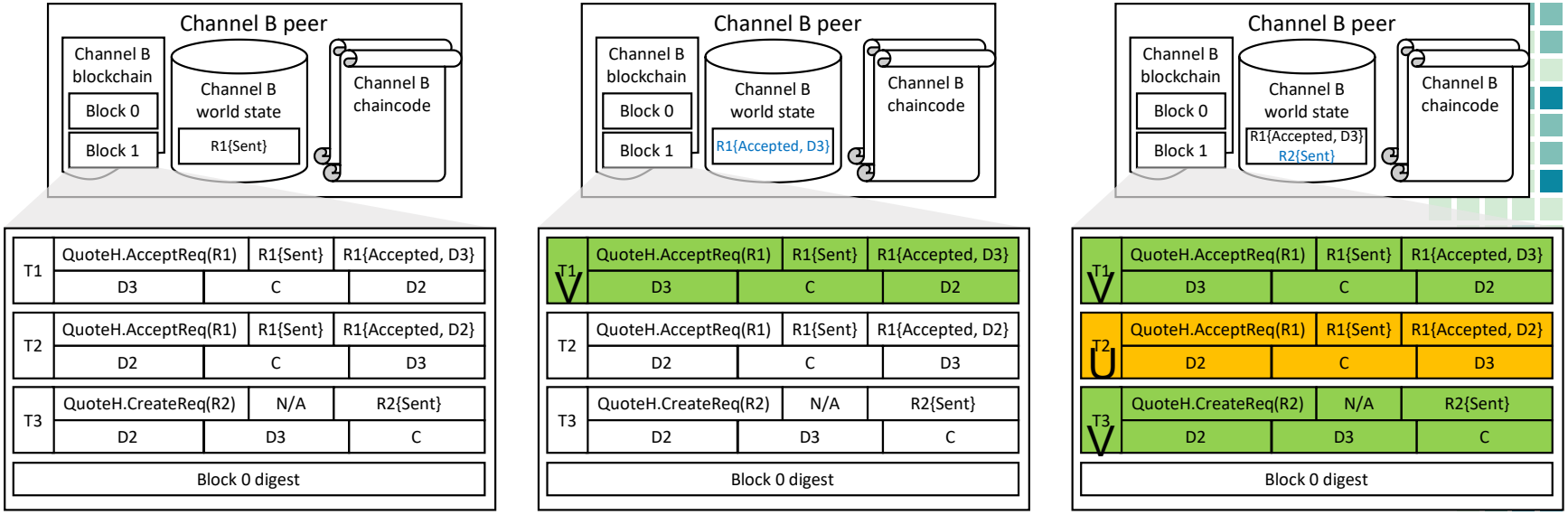
D3, D2, and C are target peers with an endorsed transaction proposal

S2 is the ordering service peer, assembling a new block

There can be multiple ordering services (peers), executing a consensus mechanism to create a new block

Marco Comuzzi mcomuzzi@unistac.kr

Execute-Order-Validate



A peer that receives a new block validates the transaction in the order in which they appear, using the input/output sets (chaincode is not needed!)

Transactions are valid if they are coherent with the world state.

Validated transaction update the "world state". Invalid transaction are recorded in the blockchain (as invalid), but do not modify the world state.

Updates are consistent for all peers in a channel >> consensus!

In the example, T2 is not valid because is inconsistent after the world state update entailed by T1

Transaction lifecycle and consensus mechanism in HLF: summing up

Chaincode in HLF can be written in any language that endorsing peers can execute (go, Java, Javascript, …), not all peers need all chaincode

There is even no requirement for "determinism" of chaincode. Endorsing peers may obtain a different output set for a transaction, the endorsing policy determines if and which response will be sent to the ordering services

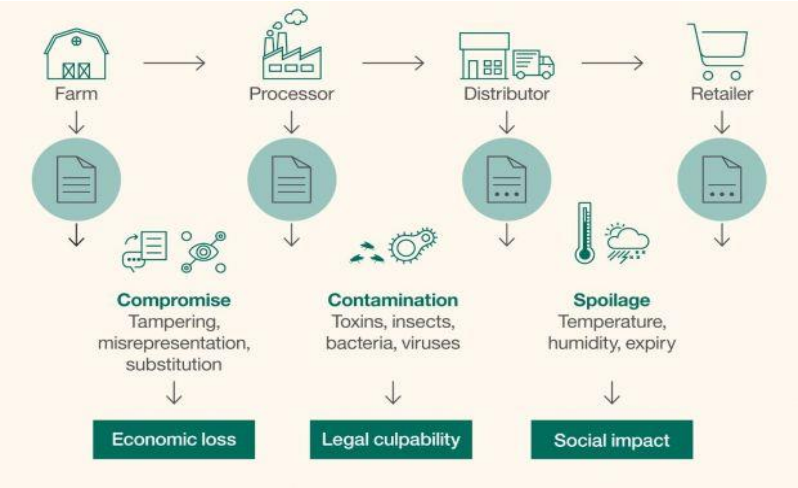
The ordering services may execute a consensus mechanism to agree on new blocks (see next lectures)

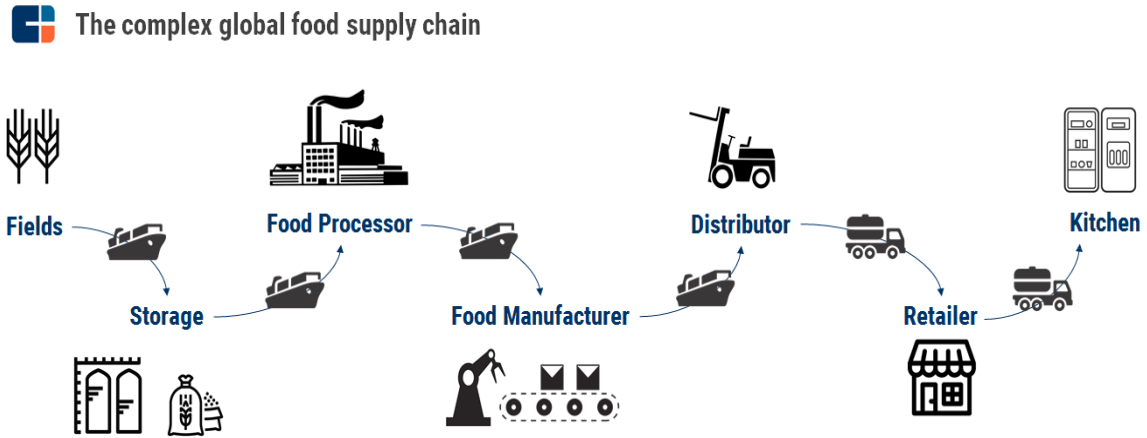
All peers receive the same new blocks: These may still contain invalid transactions, but they guarantee consistent state updates (= all peers will agree also on which transactions are invalid)

Marco Comuzzi mcomuzzi@unistac.kr

5.
HLF-based Real World Applications

IBM FoodTrust


Farm
Processor
Distributor
Retailer
Compromise
Tampering, misrepresentation, substitution
Contamination
Toxins, insects, bacteria, viruses
Spoilage
Temperature, humidity, expiry
Economic loss
Legal culpability
Social impact


The complex global food supply chain
Fields
Storage
Food Processor
Food Manufacturer
Distributor
Retailer
Kitchen
CBINSIGHTS

IBM FoodTrust

A private blockchain system to track and share information about food and produce across the supply chain

Fully implemented on Hyperledger Fabric

In production from October 2018

Walmart as early adopter

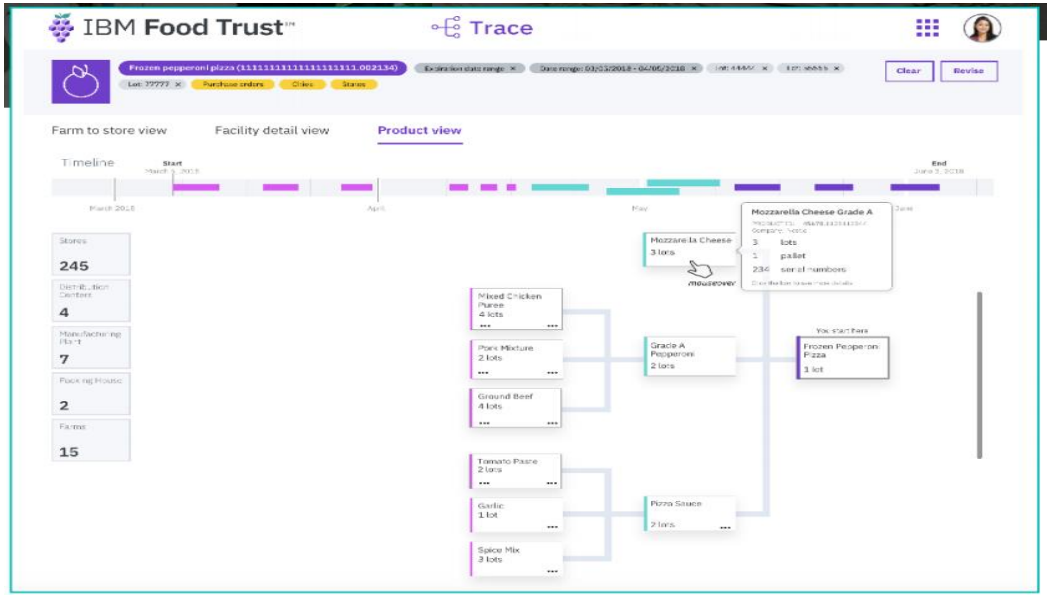
https://www.youtube.com/watch?v=QWijITDHLMQ

Provenance information

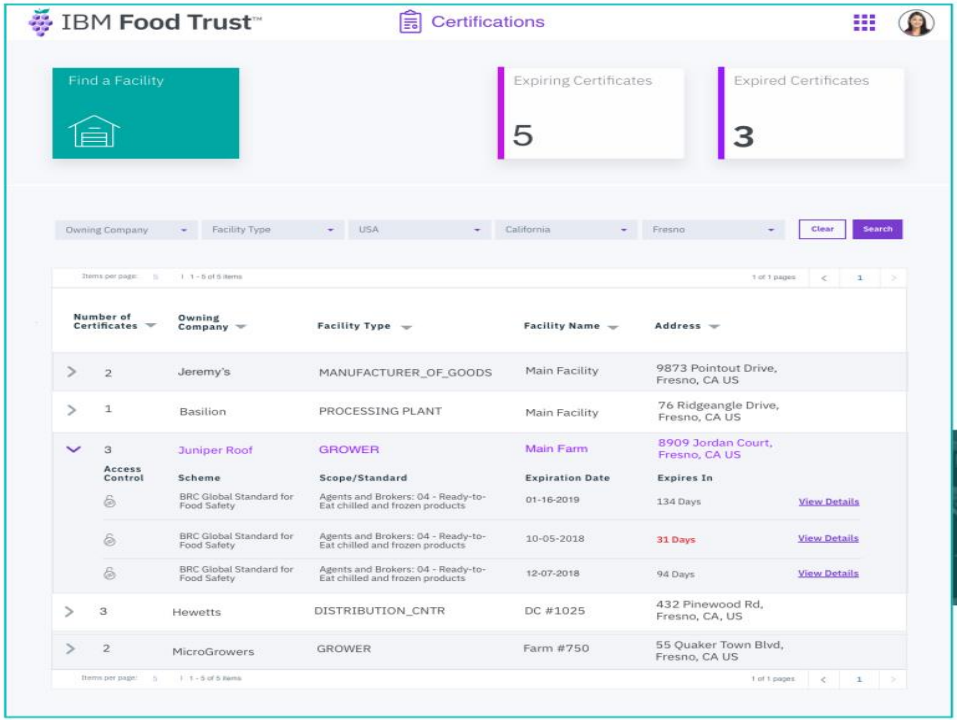
- Documents

IoT-generated information

- Temperatures
- Humidity levels
- ...



Safely, immutable store and share certification information



Trade Lens (IBM – Maersk)

Issues with Global Shipping Industry

Inconsistent data

"Re-keying" of containers across supply chain

Blind spots

Across organisations and geographic boundaries

Document-based management (often manual)

Bill of Lading, packing list, insurance policies, orders, sanitary certificates, …

Heavy reliance on P2P messaging

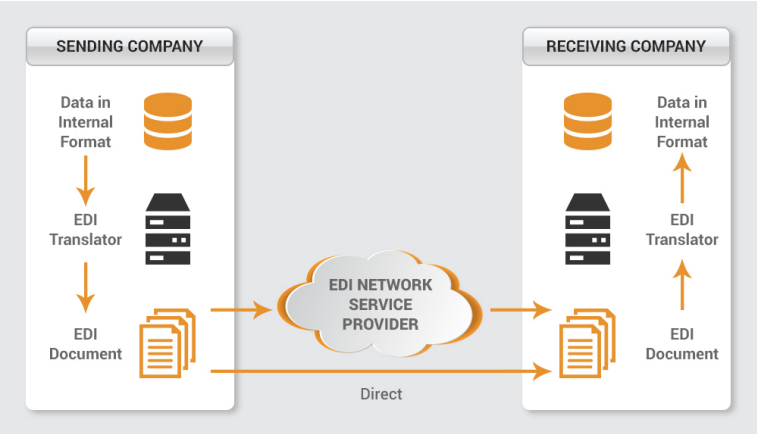
"A shipment of avocados from Mombasa to Rotterdam entails > 200 communications among > 30 parties"

Inefficient clearance processes

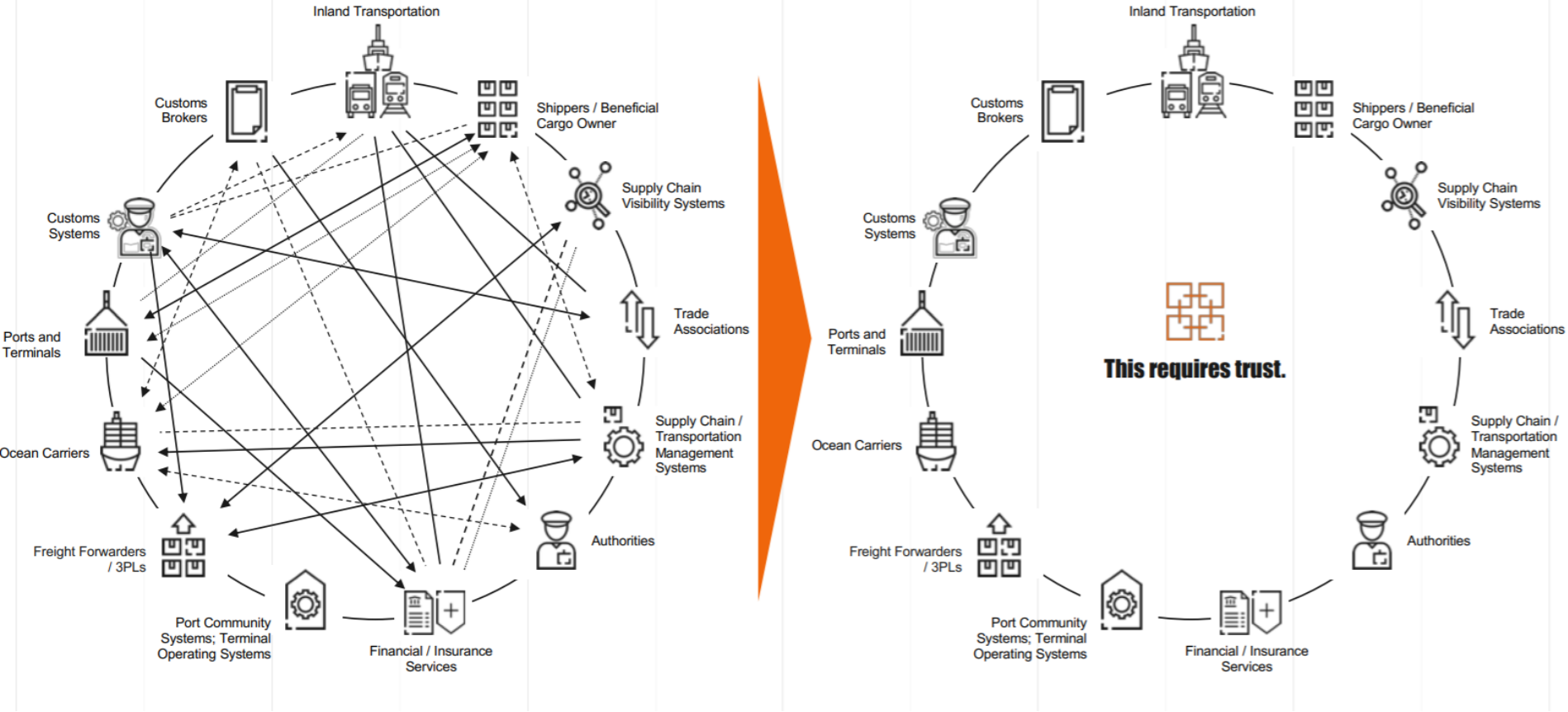
Open to fraud

Issues with Global Shipping Industry

Point to point exchange of electronic documents



SHIPPER-CENTRIC MODEL TO NETWORK MODEL



THANKS!

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

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

You can find me at:

@dr_bsad

mcomuzzi@unist.ac.kr