Instrumenting Accountability in MAS with Blockchain

Fernando Gomes Papi [UFSC]
Jomi Fred Hübner [UFSC]
Maiquel de Brito [IFRS]

[UFSC] Federal University of Santa Catarina - Brazil
[IFRS] Federal Institute of Education, Science and Technology - Brazil

October, 2017

Summary

1. Introduction
2. Formulation of Hypothesis
3. The Blockchain
4. Smart Contracts
5. Proposed Models
6. Example Implementation
7. Results and Drawbacks
8. Bibliography

Introduction

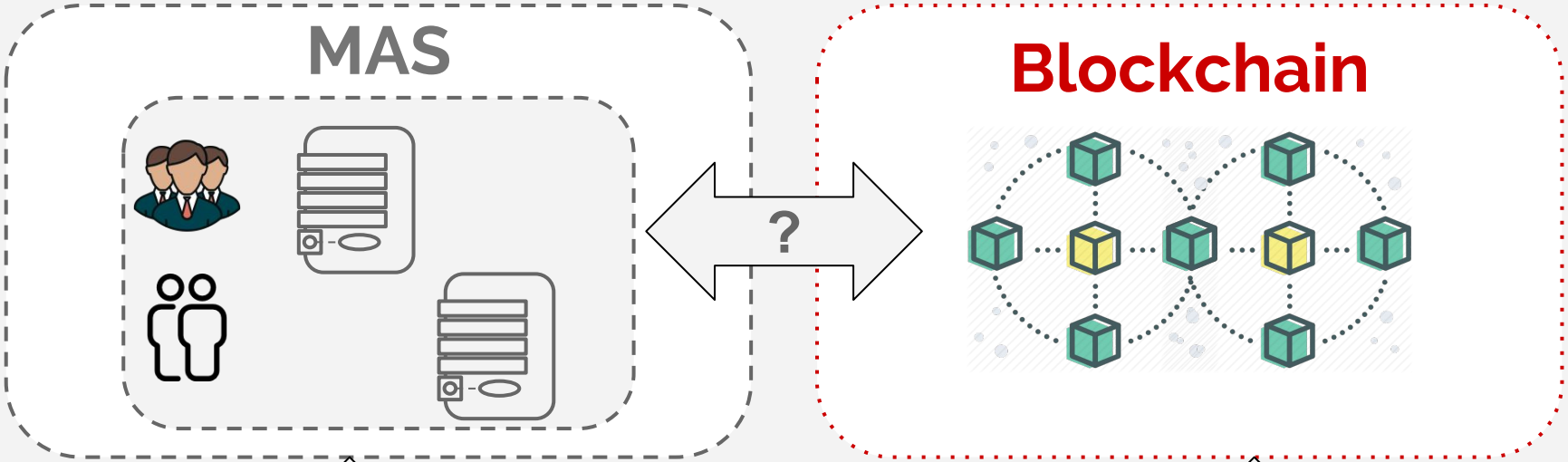
- We are entering the era of *computation as interaction*
  - Bots answer millions of human inquiries
  - Bots help us write Wikipedia
  - Bots trade stocks
  - Bots drive humans around
- Multi Agent Systems deal with decentralized autonomous computational entities
  - Design of agents
  - Design of their environment
  - Design of their goals
  - Design of a global goal
  - Design of their learning, coordination and planning
  - Design of their regulation

Introduction

- Accountability is “the acknowledgment and assumption of responsibility for decisions and actions that an individual, or an organization, has towards another party.” [6]
- Agents should be responsible for their actions and commitments. Agents are responsible for agreements they put themselves through.
- How can we provide agents reliable tools so that they can be accountable for their actions?
- Blockchains can offer agents the possibility of **trustless exchange**, **data** that is **consistent** and **timestamped** and complete **transparency** and **immutability**.

Hypothesis

Blockchain can be a powerful provider of tools for MAS accountability



- Model and Simulate Complex Systems
- Exchange Information
- Solve Distributed Problems with Global Goals
- Simplify Decentralized Computing

Examples:

- Supply Chain Management
- Transaction of Medical Records
- Economic Modeling and Planning

The Blockchain

- The technology behind the **Bitcoin** - Satoshi Nakamoto, 2009[1]
- In the context of the Bitcoin: A distributed ledger of every single transaction of bitcoins, verified by cryptographic functions (Hash Pointers of blocks of transactions)
- In a general context: A distributed database of any kind of **computational effort**, verified by cryptographic functions (Hash Pointers of blocks of generic data)

A Bitcoin Transaction[2]

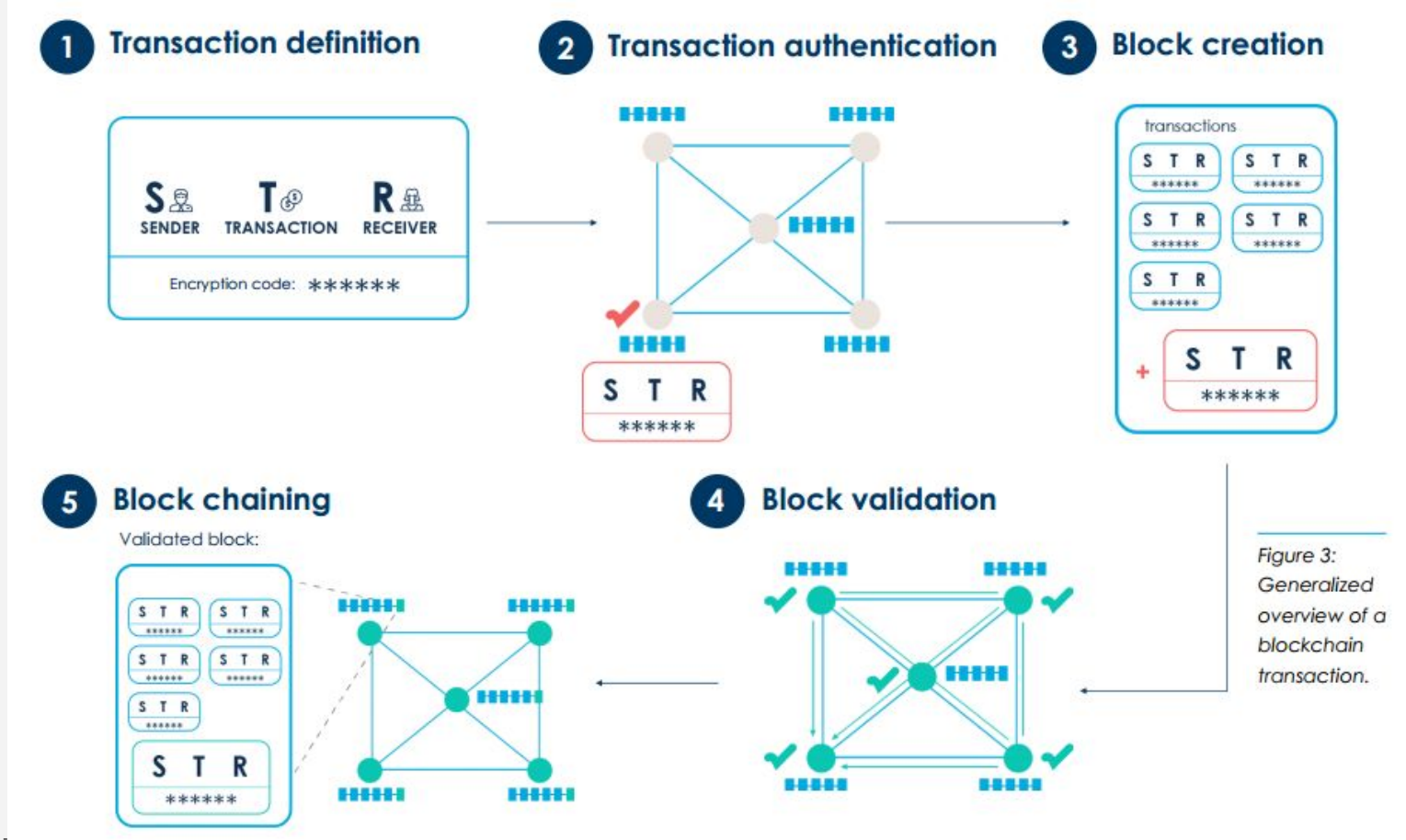
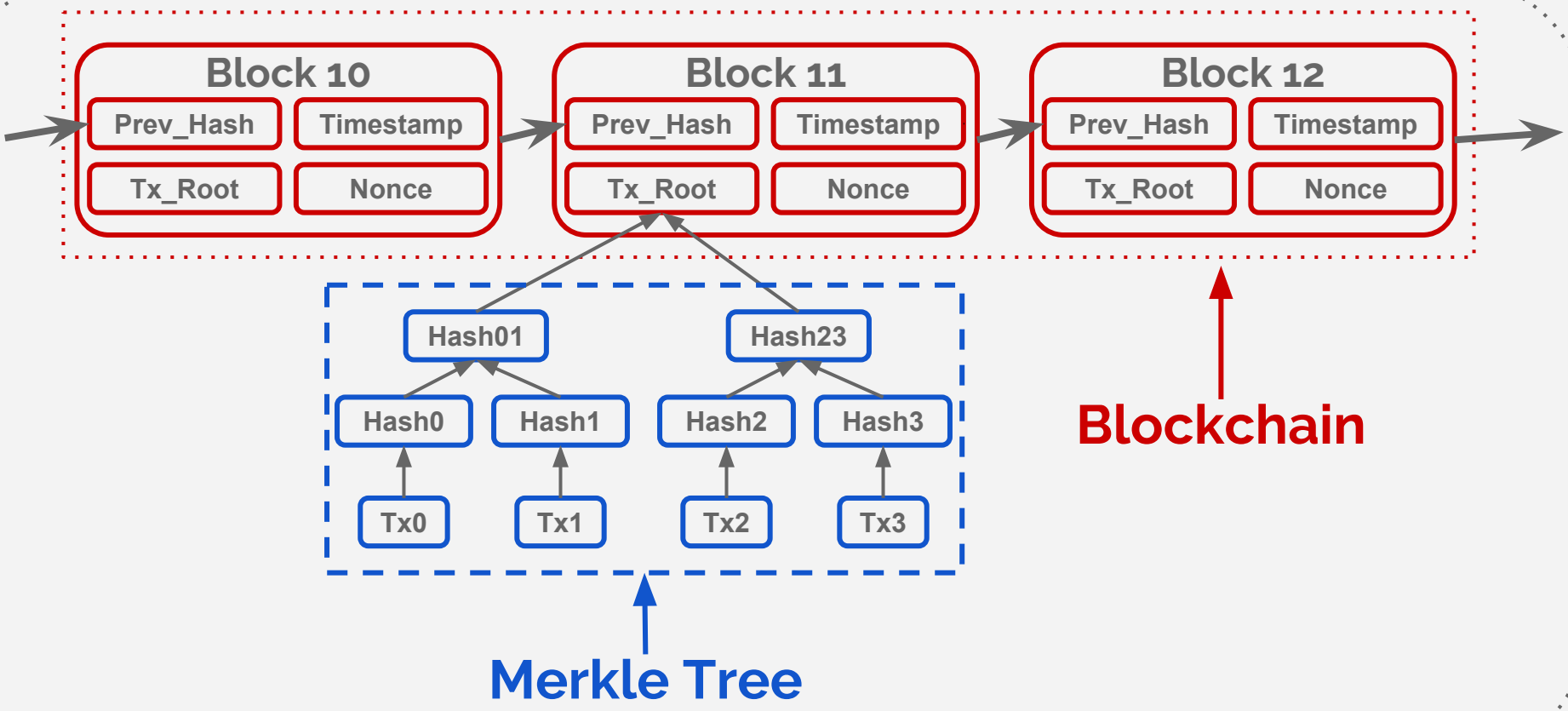
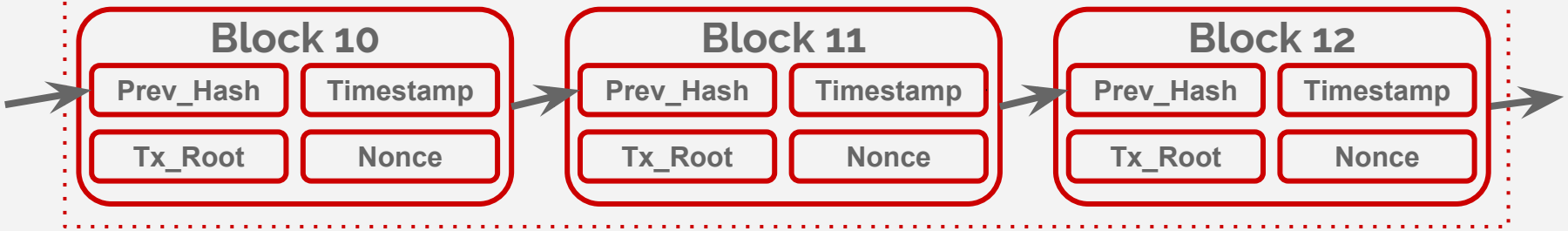


Figure 3: Generalized overview of a blockchain transaction.

Blockchain Data Structure



Why are blockchains fraud-proof?



Modifying a transaction in Block 10 will mean changing the Hash of Block 10

If the Hash of Block 10 changes, so will the Hash of Block 11

If the hash of Block 11 changes, so will the Hash of Block 12... and so on...

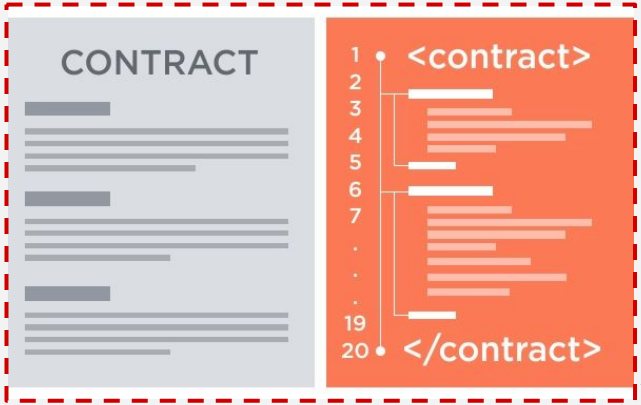
- To propagate a fraud in a Block, the attacker needs control of 51% of the network
- This means an unfeasible amount of computing power
- Blockchains are considered statistically fraud-proof
- A Transaction is normally considered safe after 6 new Blocks have been appended

From Transactions to Contracts

- “Miners” have to solve hard puzzles to **verify transactions** and **include blocks in the blockchain** to receive a reward (in bitcoin)
- Dedicated GPU’s “**waste**” huge amounts of power to solve the puzzle. Brute force search is the only method.
- Puzzle: Finding a number (**nonce**) that has a Hash with some predefined property (example: starts with five zeros)
- Instead of wasting energy, nodes could use the energy to make useful **computation of predetermined functions**

Smart Contracts

- Whenever two parts agree on a contract, it can be signed and executed through a blockchain
- Ethereum provides a platform for **Smart Contracts**[3]
- Smart Contracts are:
  - **Automatically Executed Code**
  - Reliable, Everlasting, Decentralized
  - Immutable
  - Cryptographically Secure
  - Easily verifiable from outside *agents*



The Blockchain - Key Takeaways

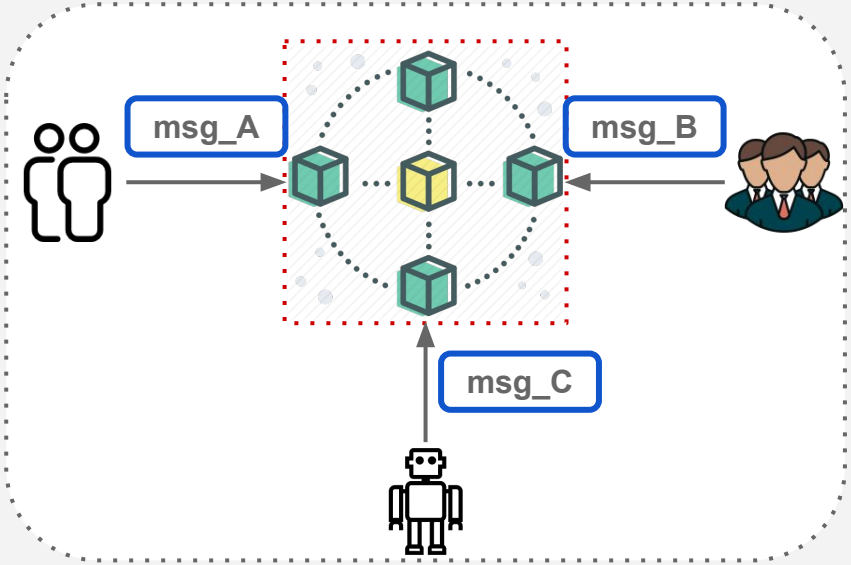
- Interactions on a Blockchain are fraud-proof
- The Ethereum Blockchain provides a programming language for deploying Smart Contracts
- Smart Contracts are pieces of code that will auto-execute when conditions are met
- Refine the Hypothesis:
    - Artifacts of the MAS could be powered by a Blockchain, providing secure, fraud-proof operations for agents

Blockchain Integrated MAS

- What is the best abstraction and model for a Blockchain in a MAS?

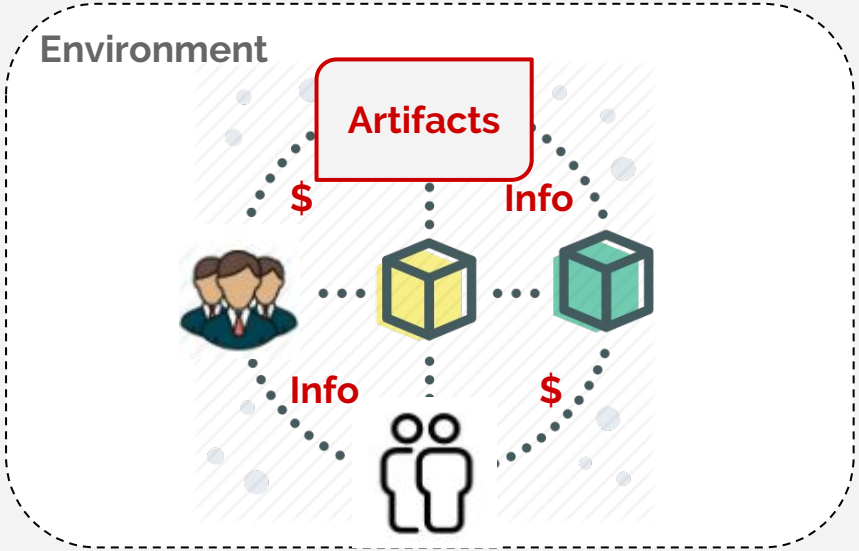
- 4 proposed models:
  - Blockchain as a means of communication
  - Blockchain as a generic Environment
  - Blockchain as a single Artifact in the Environment
  - Blockchain instrumenting application Artifacts

Blockchain as a means of communication



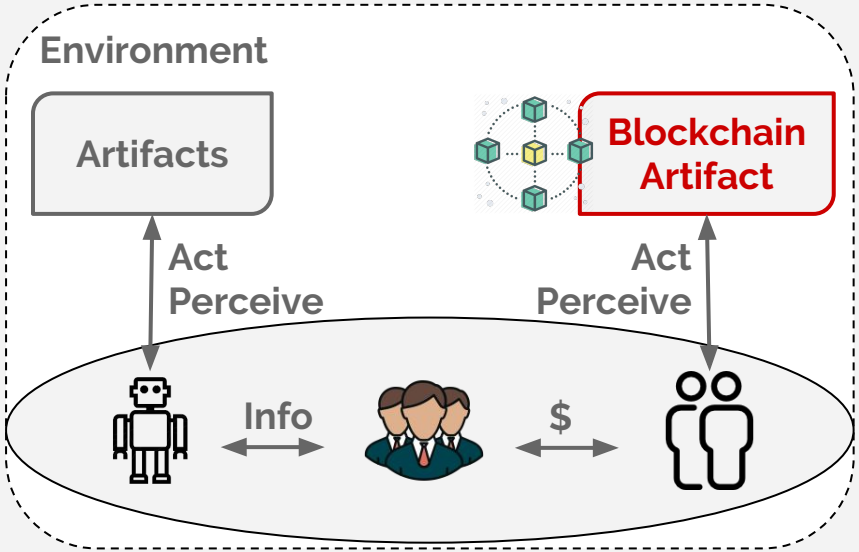
- Blockchain used as a **logger of messages** between agents
- Provides a safe register of all messages exchange: **traceability of commitments**
- No agent will be able to state that it didn't send (or receive) a determined message
- However: Traceability does not guarantee **accountability**[4]
- **Sub-utilizes** the capabilities of a blockchain

Blockchain as a generic Environment



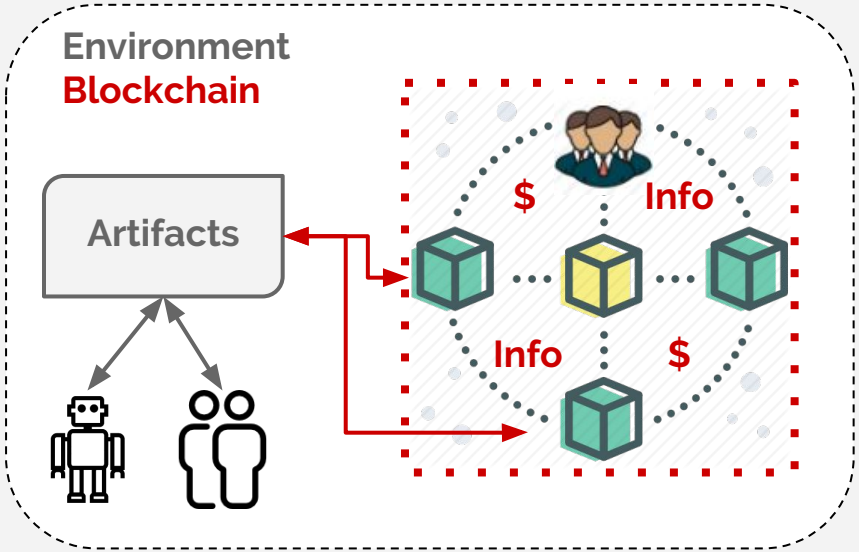
- **All Artifacts** of the Environment are implemented in the Blockchain
- Agents have **direct access** to Artifacts and other agents through the network
- **Easier** to implement, **simpler** conceptual model
- Unrealistic computational effort: **Every** interaction between agents would be registered in the Blockchain

Blockchain as an Artifact in the Environment



- A node of Blockchain executing behind **one** Artifact
- Agents can interact with it by **reading and writing** data to it
- Only common **transactions** are available
- Suitable for **simple applications** and handling of assets' **transactions** among agents (for example, payments)
- No added complexity, but **limited usage**

Blockchain instrumenting application Artifacts

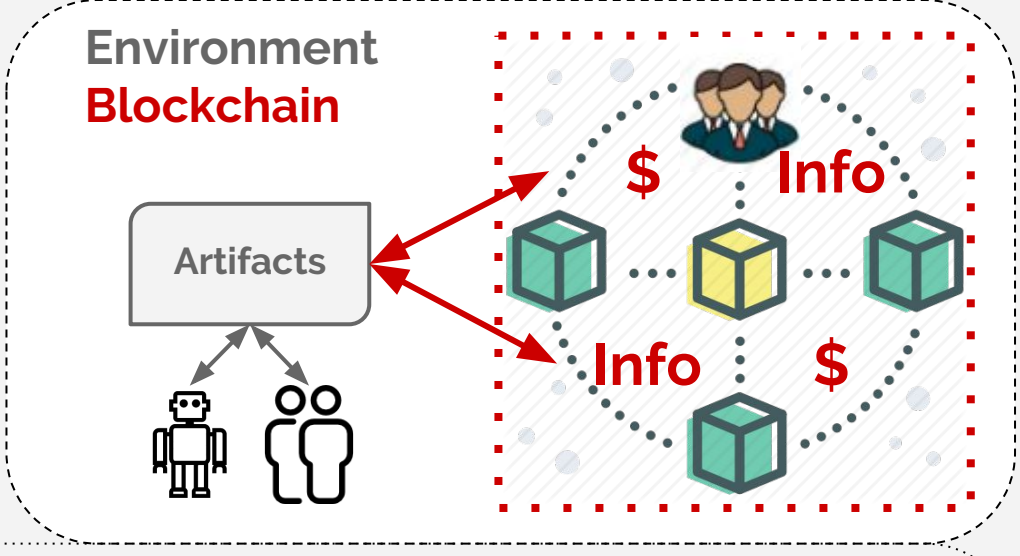


- **Each Artifact** interfaces a desirable **Smart Contract** in the Blockchain
- **The code of the designed Artifact is executed by the network**
- There could be Artifacts that execute locally, for simpler tasks
- **Provides more tools for accountability**
- Extra complexity added

Selected Model

Blockchain

instrumenting application Artifacts



- Enough flexibility to include both **on-chain** and **off-chain** Artifacts
- Able to provide essential **tools for accountability** among agents
- Encapsulates all other discussed models, while remaining flexible: Artifact for **message logging**, Artifact for **asset transactions**, Artifacts for **Smart Contracts** executing **accountability** tools, etc.

Example - Building a House

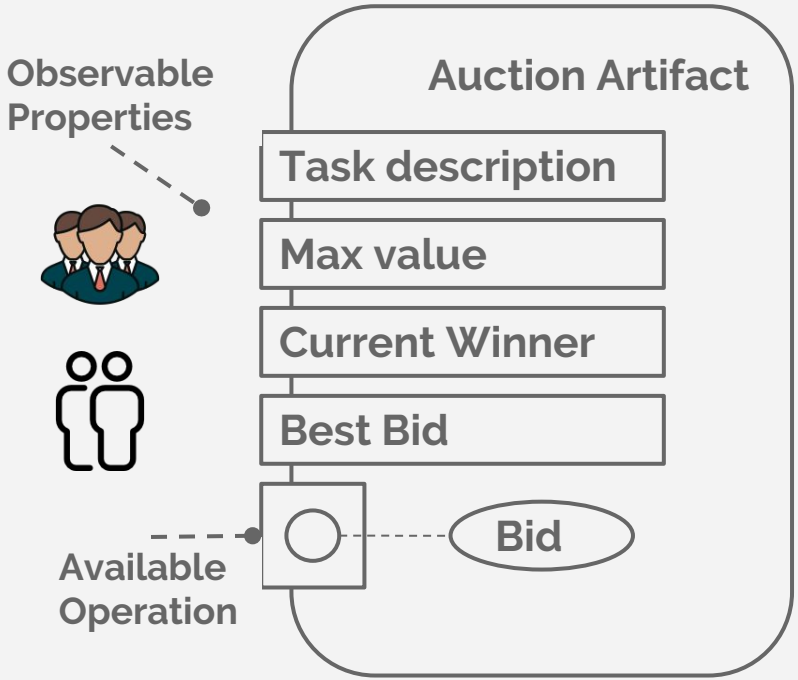
Scenario:

- Giacomo wants to **build a house**. He **knows the tasks** to be completed (Site Preparation, Walls, Floors, Roof, Windows, Doors, Plumbing, Electrical System, Exterior Painting, Interior Painting) and **how much he can pay** for each task.
- Giacomo creates **Auction Artifacts** for companies to **bid on the task**.
- After the auction closes, the **hired company must execute** the service
- Overview of the system:
  - Agents: **Giacomo** (House Owner), **Companies** (Contractors that will bid on tasks)
  - Artifacts: **Auctions** for each task to be contracted
  - Organisation: **coordination** and **cooperation** in the execution of the global workflow

Original MAS

- Artifacts were created with the CaRTagO language, defined as:

- Artifacts are **executed locally**
- Artifacts last **as long as the execution**
- The system executes in a few seconds
- The final product is a simulation house constructed for Giacomo



MAS + Blockchain

- Artifacts are coded and deployed in the Blockchain network as **Smart Contracts**
- A corresponding artifact will be created in the MAS in order to **interface the interaction between agents and Smart Contracts**, providing the same Observable Properties and Operations as before
- Smart Contracts are created and **executed in a decentralized network**
- Smart Contracts will last **as long as the Blockchain exists** (theoretically, forever)
- New instances of the MAS can use the **exact same** Smart Contract
- The system runs in **several minutes**

```
function CreateAuction( string _task,
                        uint256 _maxValue,
                        uint256 _currentBid,
                        string _currentWinner) public returns (uint auctionId) {

    auctionCount++;
    auctionList[_task].task = _task;
    auctionList[_task].maxValue = _maxValue;
    auctionList[_task].currentBid = _currentBid;
    auctionList[_task].currentWinner = _currentWinner;

    return auctionId;
}

function placeBid(string task, uint bidValue, string bidder) public {
    Auction storage a = auctionList[task];
    if (a.currentBid > bidValue){
        a.currentBid = bidValue ;
        a.currentWinner = bidder;
    }

}

function getCurrentWinnerbyAuctionID(string task) public view returns (string winner) {
    Auction storage a = auctionList[task];
    return a.currentWinner;
}

function getCurrentBidbyAuctionID(string task) public view returns (uint curretBid) {
    Auction storage a = auctionList[task];
    return a.currentBid;
}
```

The whole code for this Smart Contract is available at:

github.com/FerPapi

Bid (Operation)

Current Winner
Best Bid
(Observable Properties)

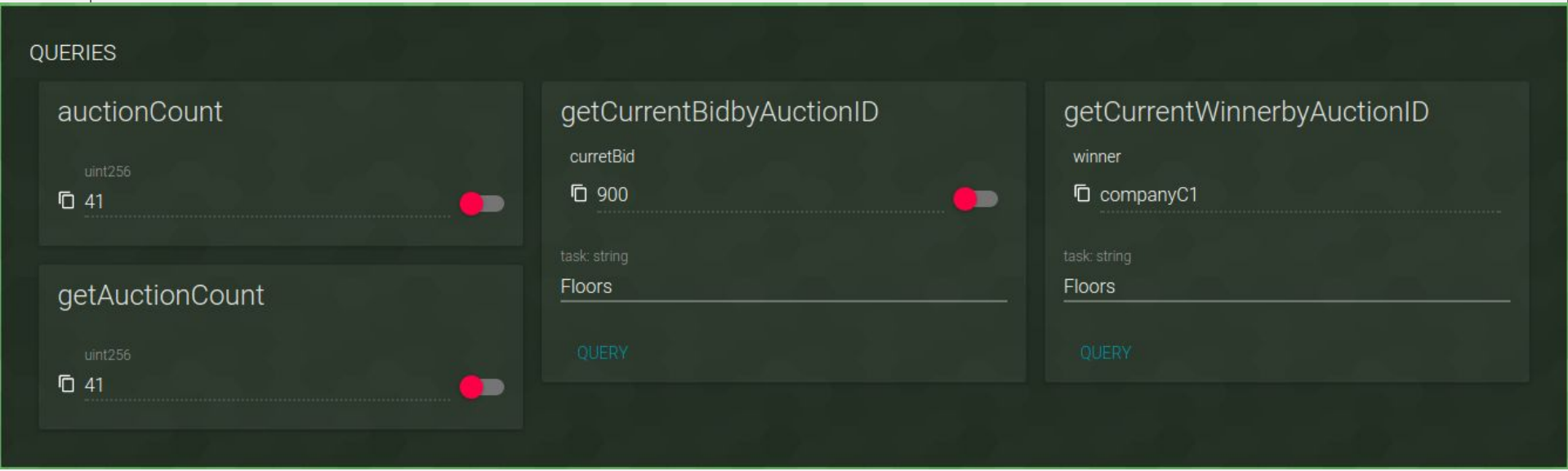
Example - Preliminary Results

- The Auction Smart Contract was deployed and tested with the MAS **successfully**
- The systems **gets the house built** for Giacomo in the simulation, and all the auction winners could be forever checked on the Blockchain
- The Blockchain technology poses as a great **potential addition** to MAS
- The token that runs in the Ethereum Network, the **Ether**, is evaluated at about **US$ 300**. This means that MAS can bring solutions to real life problems when using the Ethereum Blockchain

Example - Preliminary Results

External User Interface Explorer for this contract:

- Running the Parity client -- *parity.io*
- The Testnet Kovan was used
- Anyone with access to the Network can check the contract
- The address / QR Code for this contract is available at the end of this presentation



Example - Preliminary Results

Limitations and Drawbacks

- The Network runs quite **slowly**
- The technology is extremely new and **still evolving**
- **Synchronization problems** occur frequently, especially because the Network runs orders of magnitude slower than the local execution of the MAS
- Due to limitations of time, it still wasn't possible to demonstrate a use case of Blockchain regarding **accountability** issues. However, there are strong evidences that this technology can be very useful in this scenario

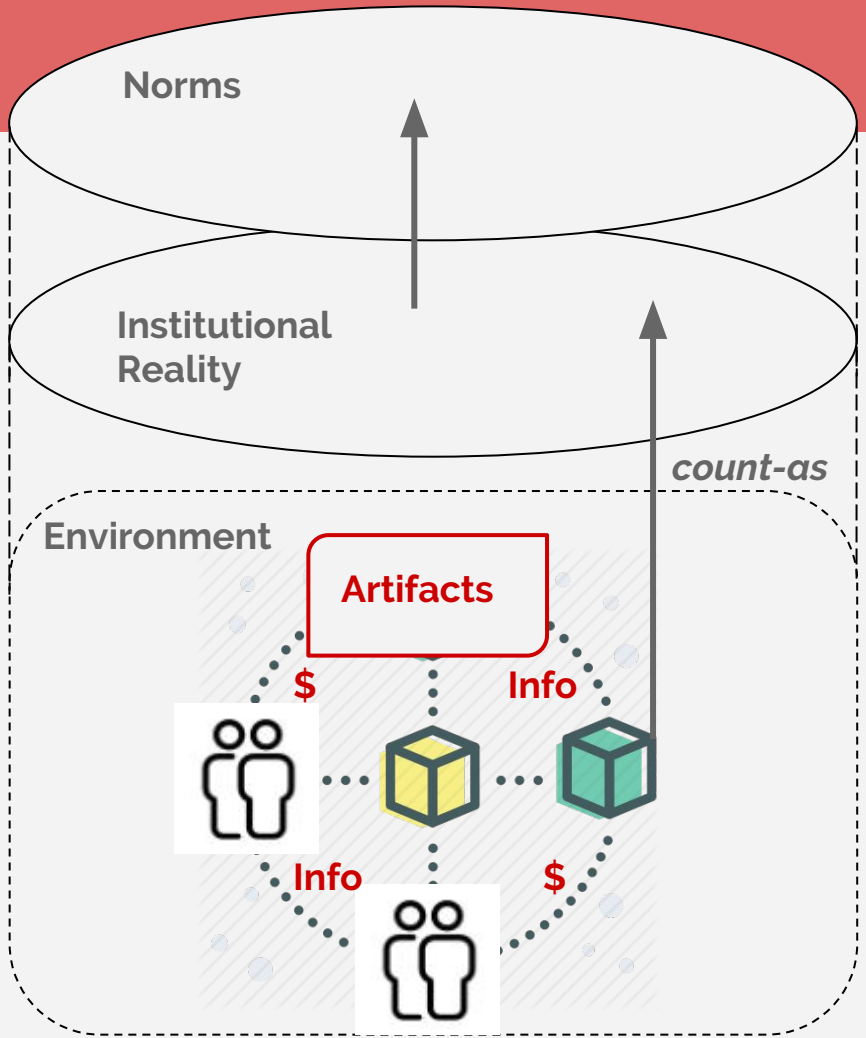
Example - Preliminary Results

Applications of the Blockchain in Computational Accountability

- The Blockchain can provide **trustful traceability** of messages, commitments, transactions, etc.
- The Blockchain can automate the **verification of commitments** and completion of tasks
- With more complex Smart Contracts, it is possible to **assign roles**, **delegate tasks**, **provide authorization**, check for **proof of membership**, check **available funds**, etc.
- Smart Contracts can carry the **execution of penalties** according to predefined rules
- Smart Contracts can automate the **payment and transaction of assets** between agents, in both **real life and simulation** scenarios

Future Works

- Make a robust and fail proof integration of Blockchain and MAS
- Combine the Blockchain model with the Situated Artificial Institution Model[5]
- Create a more complex application scenario: A Supply Chain simulation in MAS



Bibliography

[1] - Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system.* 2008

[2] - Froystad, P; Holm, J. *Blockchain: powering the internet of value*. 2016

[3] - Vitalik Buterin et al. *Ethereum white paper*. 2013.

[4] - Chopra, Amit K and Singh, Munindar P. *The Thing Itself Speaks*

[5] - Maiquel de Brito et al. *A model of institutional reality supporting the regulation in artificial institutions*, 2016

[6] - Matteo Badoni, Cristina Baroglio, Katherine M. May, Roberto Micalizion, Stefano Tedeschi. Computational Accountability. 2016

Instrumenting Accountability in MAS with Blockchain

Questions?

Fernando Gomes Papi
Jomi Fred Hübner
Maiquel de Brito

Address for the example Smart Contract:

0x040185003FE21AFD615De97330Dc61C2C8707f19

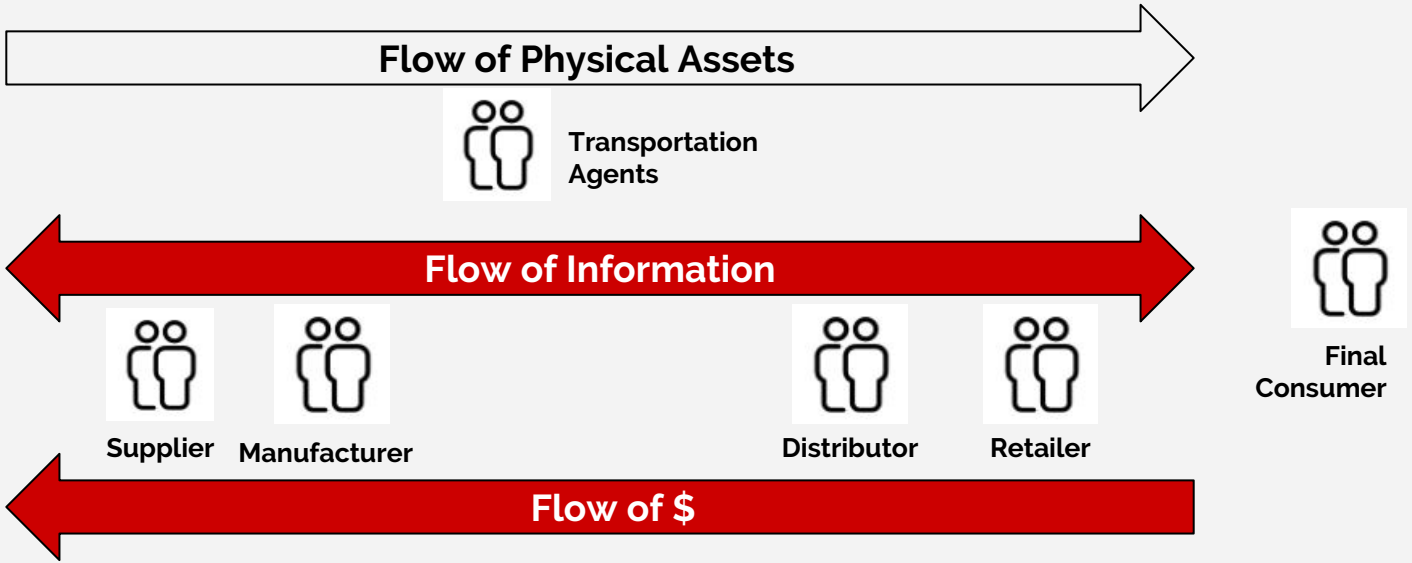
Network: Kovan Tesnet

Client: Parity (*parity.io*)

Code available:

github.com/FerPapi

Blockchain Integrated MAS Supply Chain Management

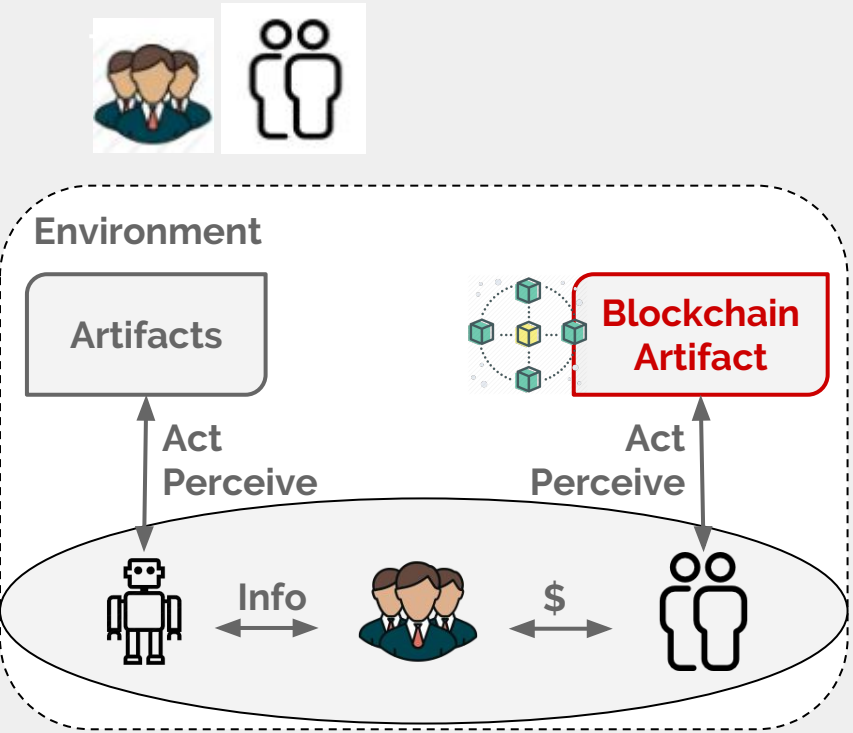


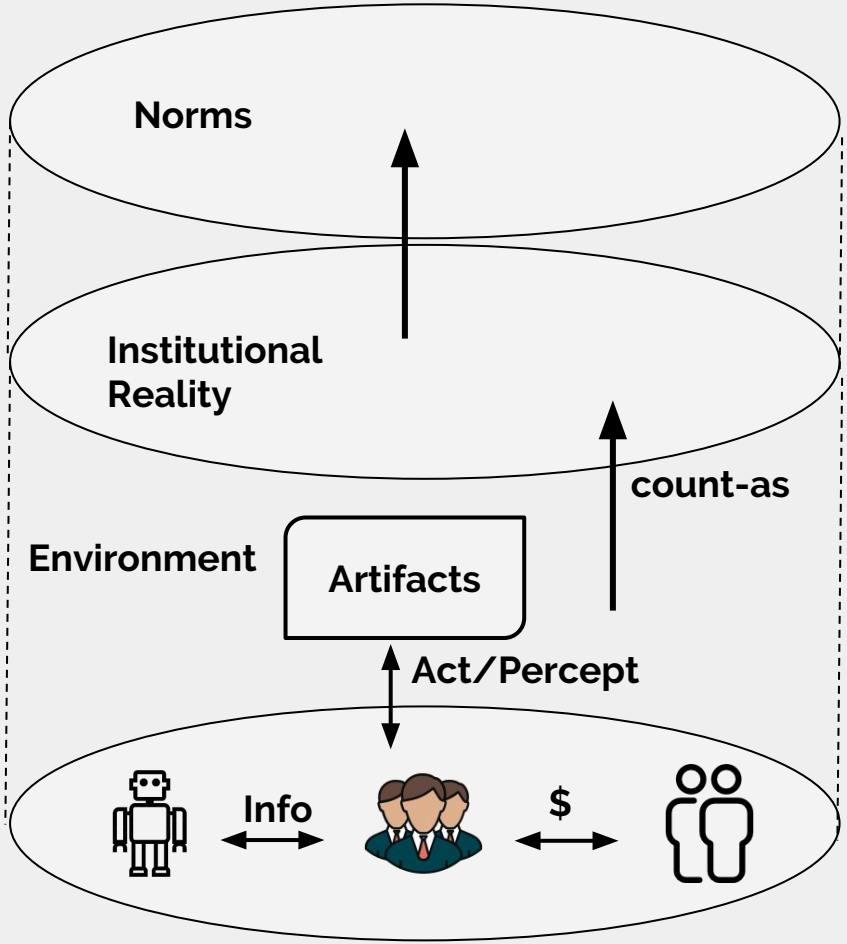
- Instantaneous Flow of Information
- Third-party Independent Flow of $
- Distributed Consensus

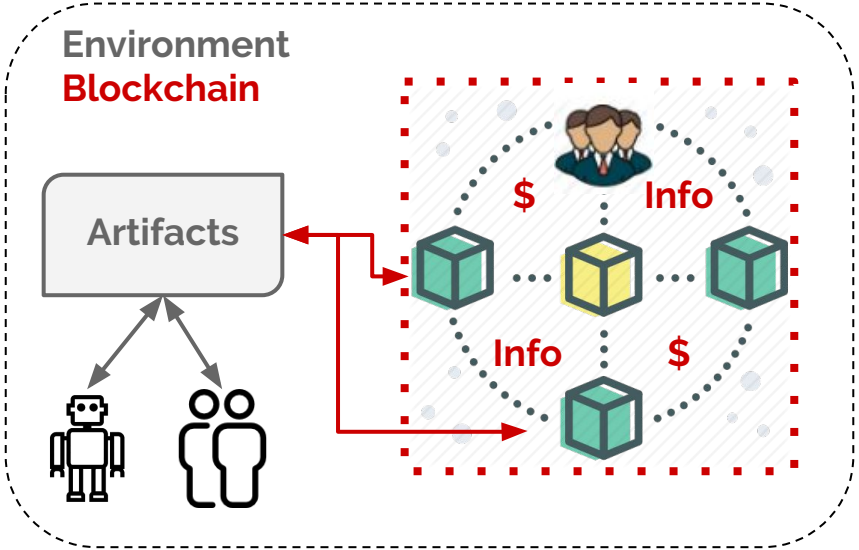
Blockchain Powered

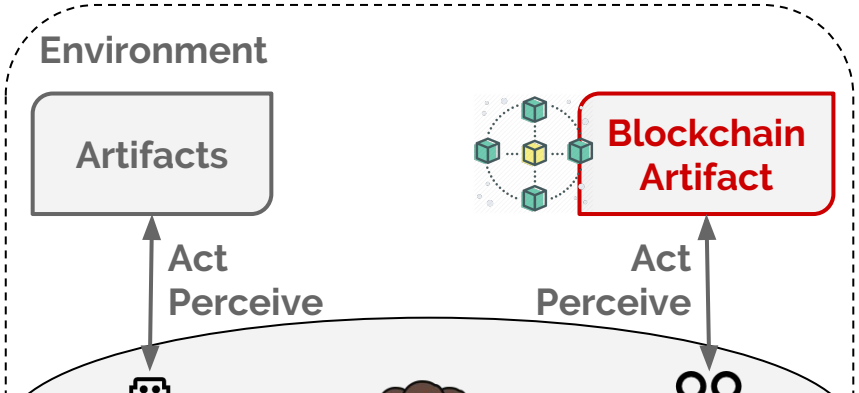
SCM/MAS+Blockchain

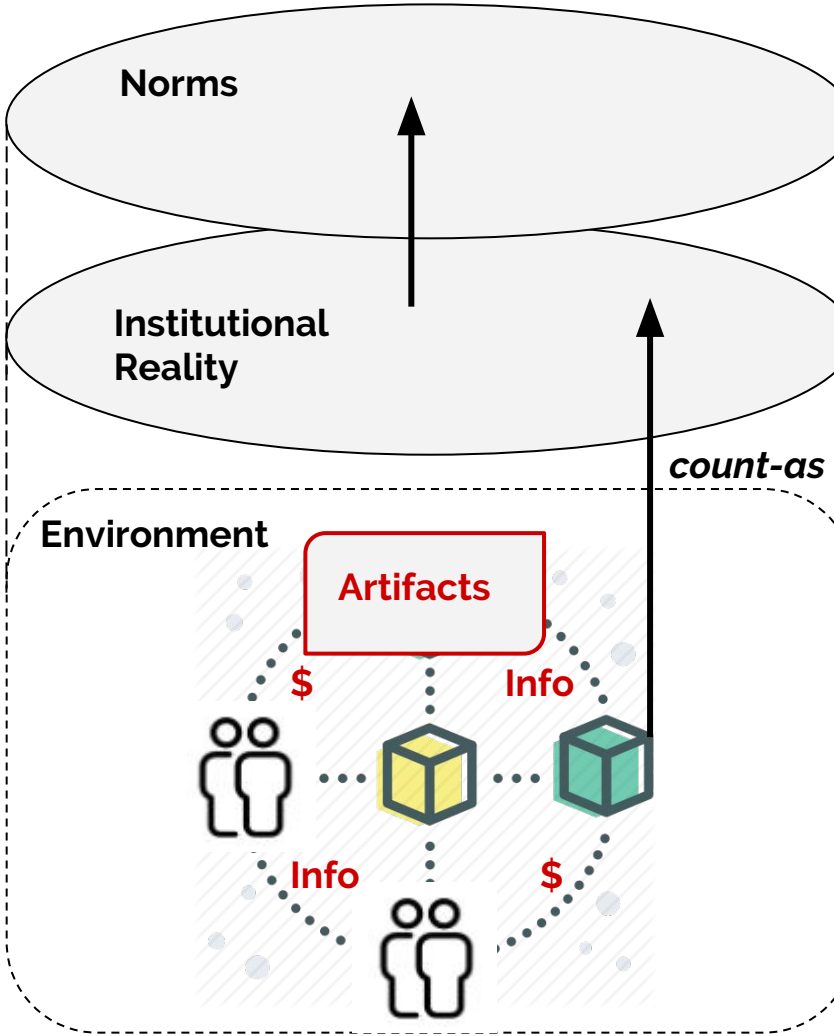
- SCM has an important problem, with extensive research literature and many MAS propositions
- The Bullwhip Effect is the amplification of variance of orders from demand to supply
- Many terrible effects, such as overproduction, price fluctuations, production scheduling, creating overall economic inefficiency
- Hypothesis: This MAS Model can help mitigate Bullwhip Effects along the SCM

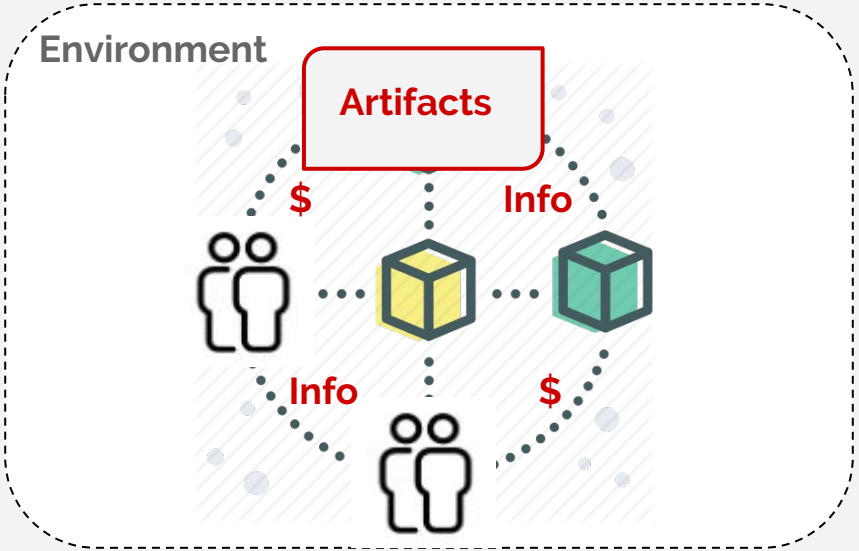

Environment
Artifacts
Blockchain Artifact
Act
Perceive
Act
Perceive
Info
$


Norms
Institutional Reality
count-as
Environment
Artifacts
Act/Percept
Info
$


Environment
Blockchain
Artifacts
$
Info
Info
$


Environment
Artifacts
Act
Perceive
Blockchain
Artifact
Act
Perceive


Norms
Institutional
Reality
count-as
Environment
Artifacts
$
Info
Info
$


Environment
Artifacts
$
Info
Info
$