# Component provenance tracing through blockchain-based, trackable data exchange system for safety critical industrial supply chains.

Sangeeta Joseph, 7th December 2020, Master's Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

# Outline

Motivation

Problem Statement & Research Question

Approach

Theoretical Background

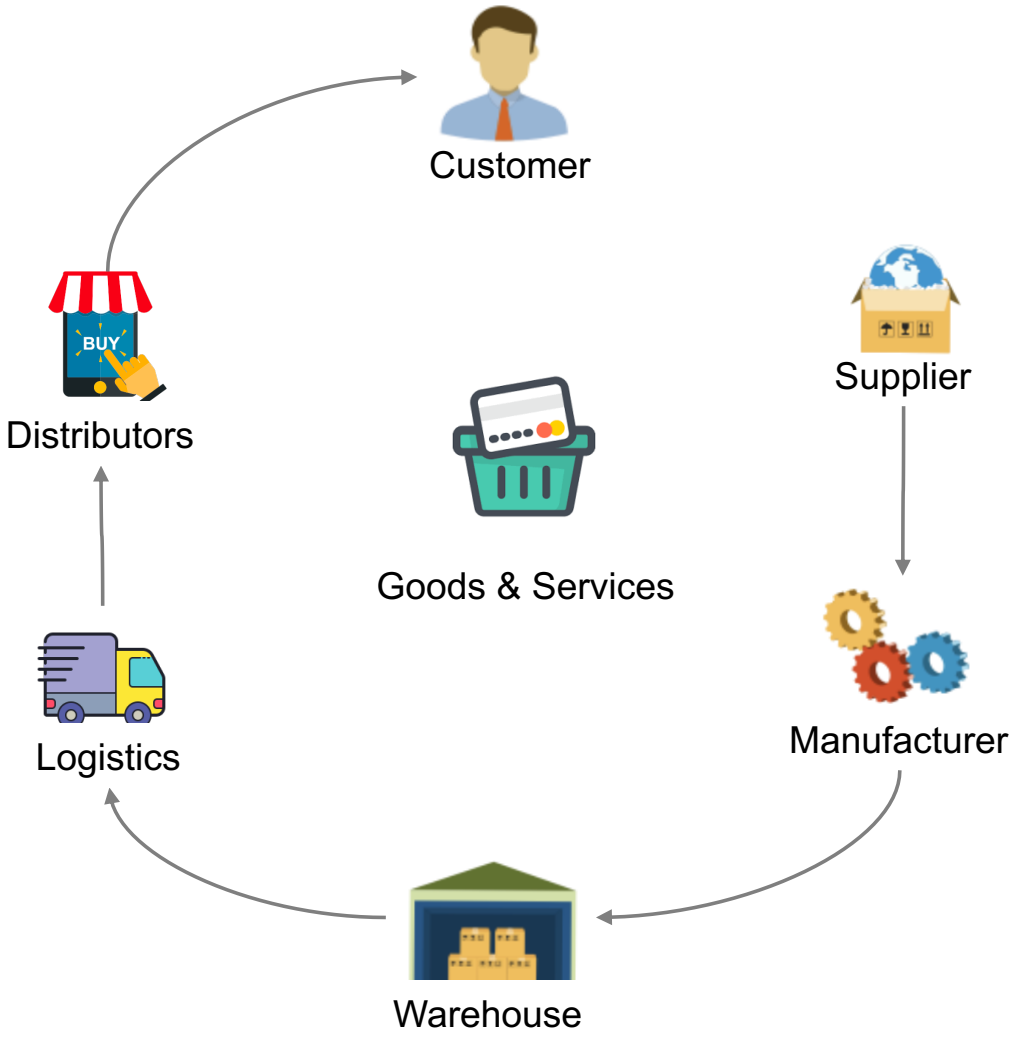
Solution

Requirements, Architecture & Implementation

Evaluation

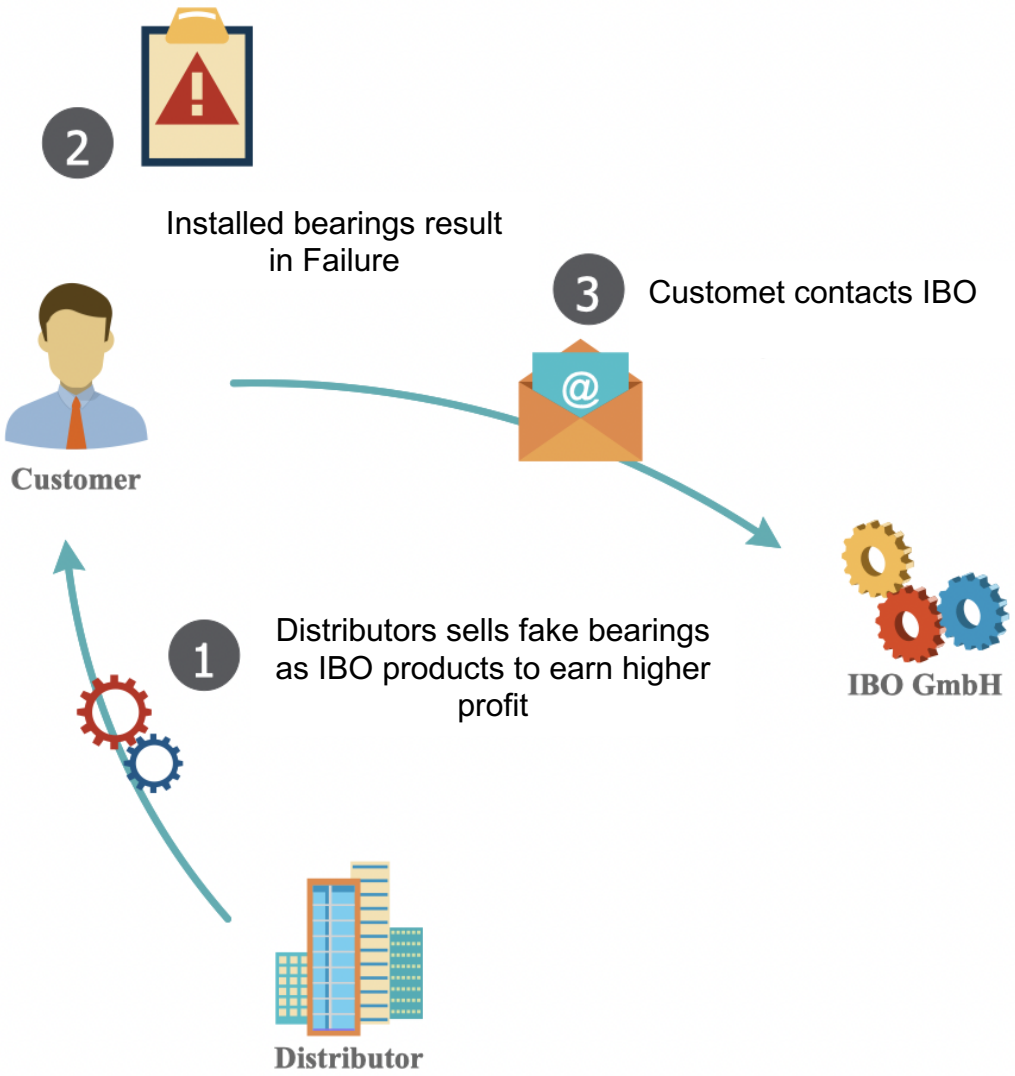
Conclusion

# Motivation



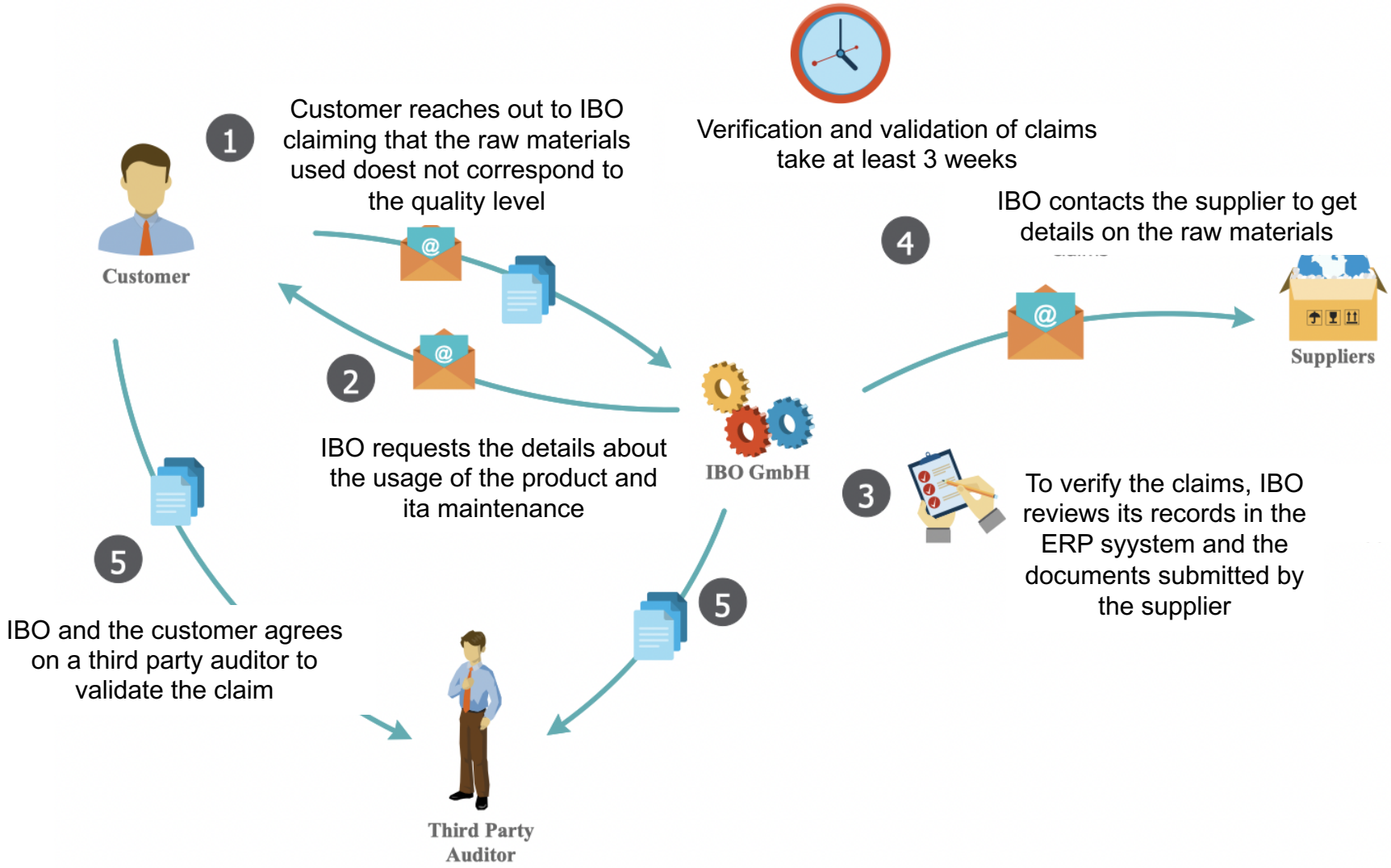

IBO GmbH is a SME, manufacturing roller bearings for safety critical domains.

# Motivation: Counterfeit




Track & trace for high-price products with ID numbers lasered in the product body as well as low-price products sold by batch.

# Motivation: Longer Audits




Transparent and trusted, fraud-proof data storage for fast and efficient audit purpose

# Problem Statement

The goal of this master's thesis is to design and implement a blockchain-based supply chain solution for SMEs in safety critical supply chains in collaboration with IBO GmbH to provide transparent data flow and product tracking without disclosing business secrets.

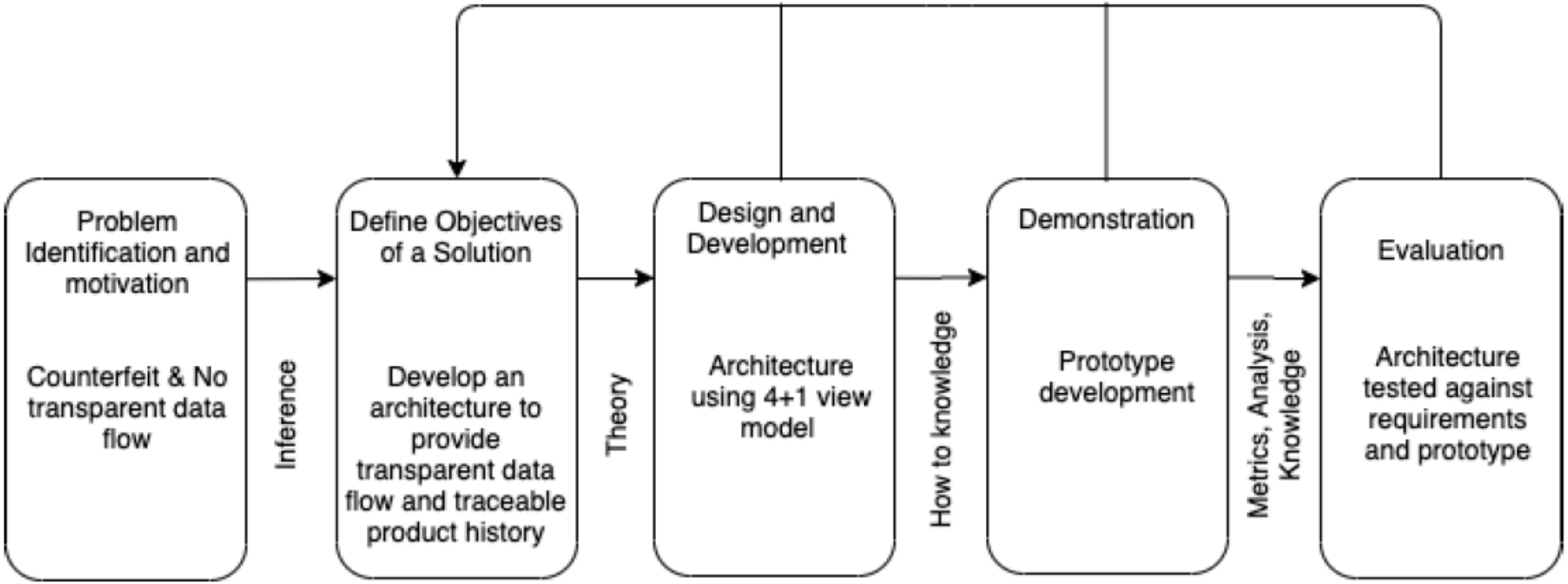
# Research Questions

**<RQ1>:** What are the requirements for blockchain-based supply chain system to reduce fraud and improve supply chain management?

**<RQ2>:** What is an architecture of a blockchain-based system for fraud reduction and supply chain management?

**<RQ3>:** What is the prototypical implementation for a Blockchain-Based Supply Chain system for fraud reduction and supply chain management?

# Design Science Research Methodology




Source: K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee. “A Design Science Research Methodology for Information Systems Research.”

# Why Blockchain?

Blockchain provides a transparent immutable tamper-proof data storage.

# System Analysis: Traditional vs Blockchain-based Supply Chain

| | Traditional Supply chain | Blockchain-based Supply chain |
|---|---|---|
| Data Storage | Each party stores its own data and the data that is shared by the other parties. | Each data record is stored on the ledger upon which the parties had previously agreed. |
| Data Sharing | Data is shared through emails, paper documents. | Any party can access the data only if granted authorization by the owner of the data. |
| Audit Time | The process can be lengthy (up to several weeks) because the validation is done on each party's data copies. | The validation can be quickly performed by accessing the ledger as the parties agree upon the data stored. |
| Product Tracking | To track each phase in the supply chain, partial data from all party members must be processed and assembled into a product history. As more members are added to the supply chain, the time to complete this process increases exponentially. | The product history can be built by traversing the product connections in the ledger. |
| Fraud | Party members can compromise their own data and update history leading to a false product history. | The data update history is stored on the ledger, which is tamper-proof by design. |
| Security | Traditional supply chains are vulnerable to data breaches and cyber attacks. | As the blockchain is replicated on all the nodes in the network, the consensus mechanism and the cryptographic techniques, cyber attacks are almost impossible. |

# Related Projects

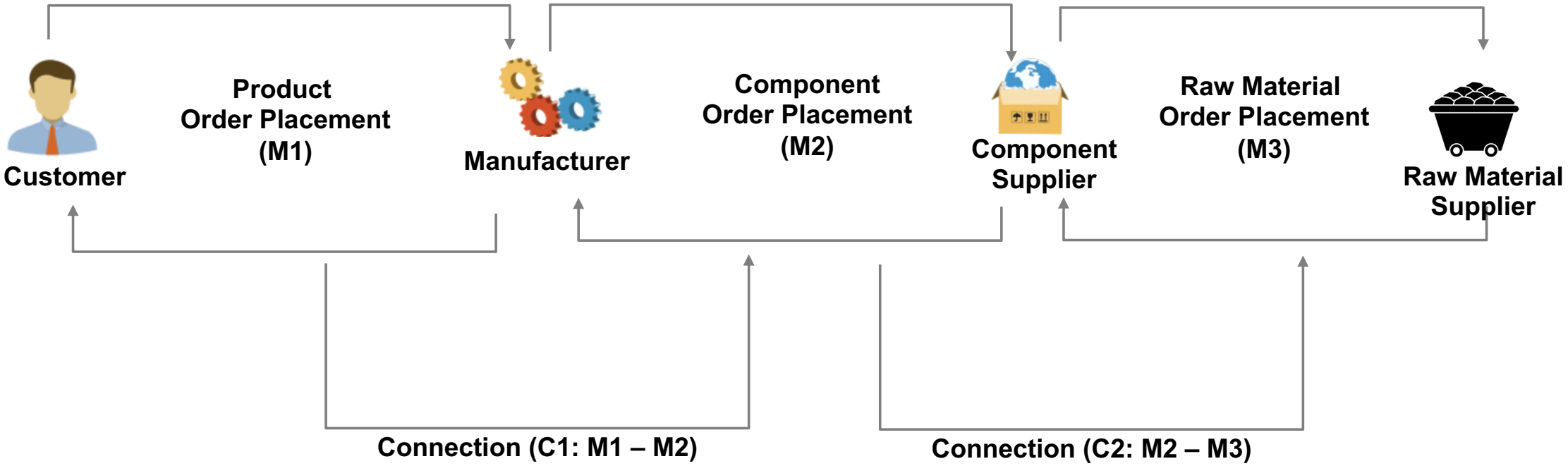
## Everledger

Everledger is an independent technology company aiming to provide blockchain-based solutions in luxury goods, wine, and diamonds. These industries are facing challenges related to authenticity and sustainability.

## Origintrail

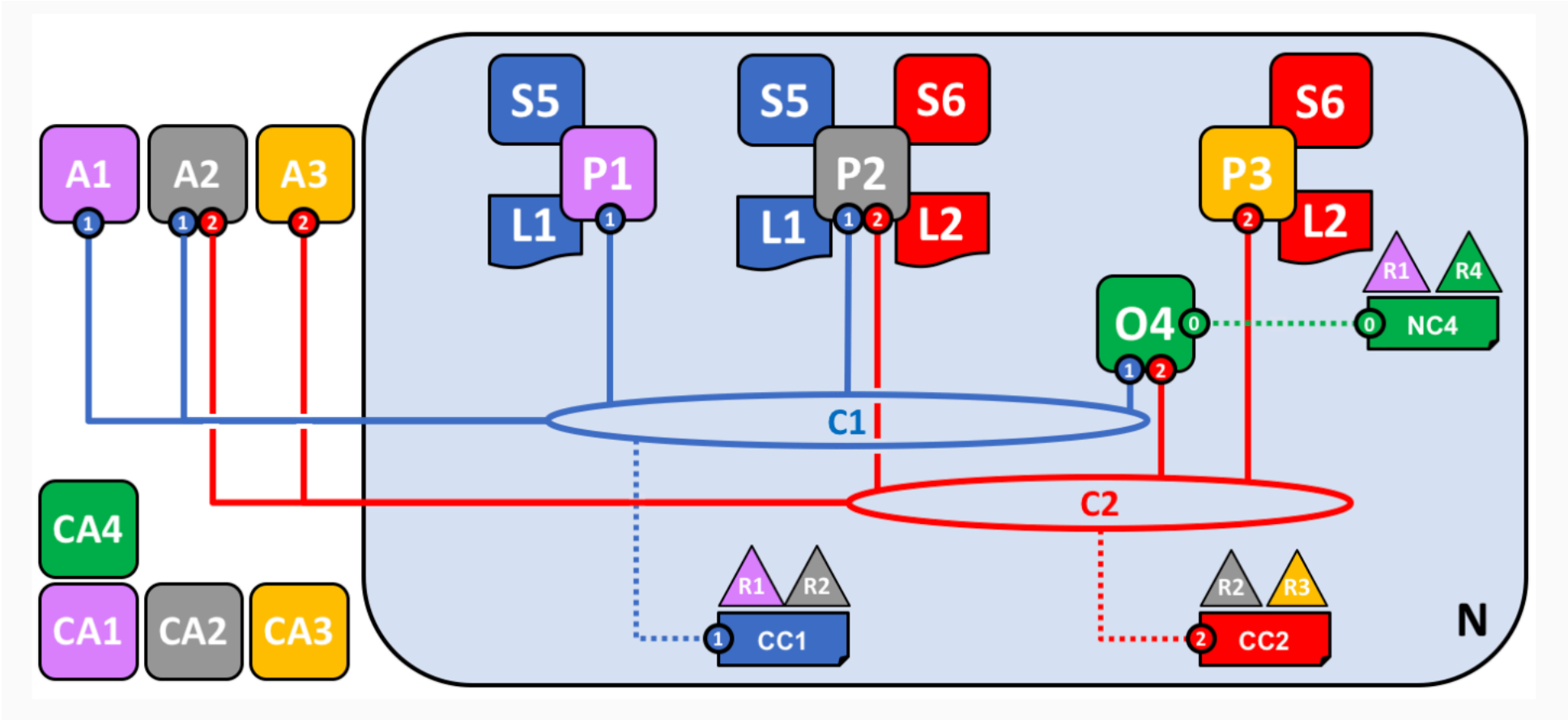
Origintrail is a blockchain-based ecosystem aimed at connecting legacy supply chain IT systems to ensure data immutability. The Origintrail protocol solves three significant issues: poor interoperability, inefficient data storage, reluctance to share data.

# Solution


Customer
Product
Order Placement
(M1)
Manufacturer
Component
Order Placement
(M2)
Component
Supplier
Raw Material
Order Placement
(M3)
Raw Material
Supplier
Connection (C1: M1 – M2)
Connection (C2: M2 – M3)

# Theoretical Background : Hyperledger Fabric

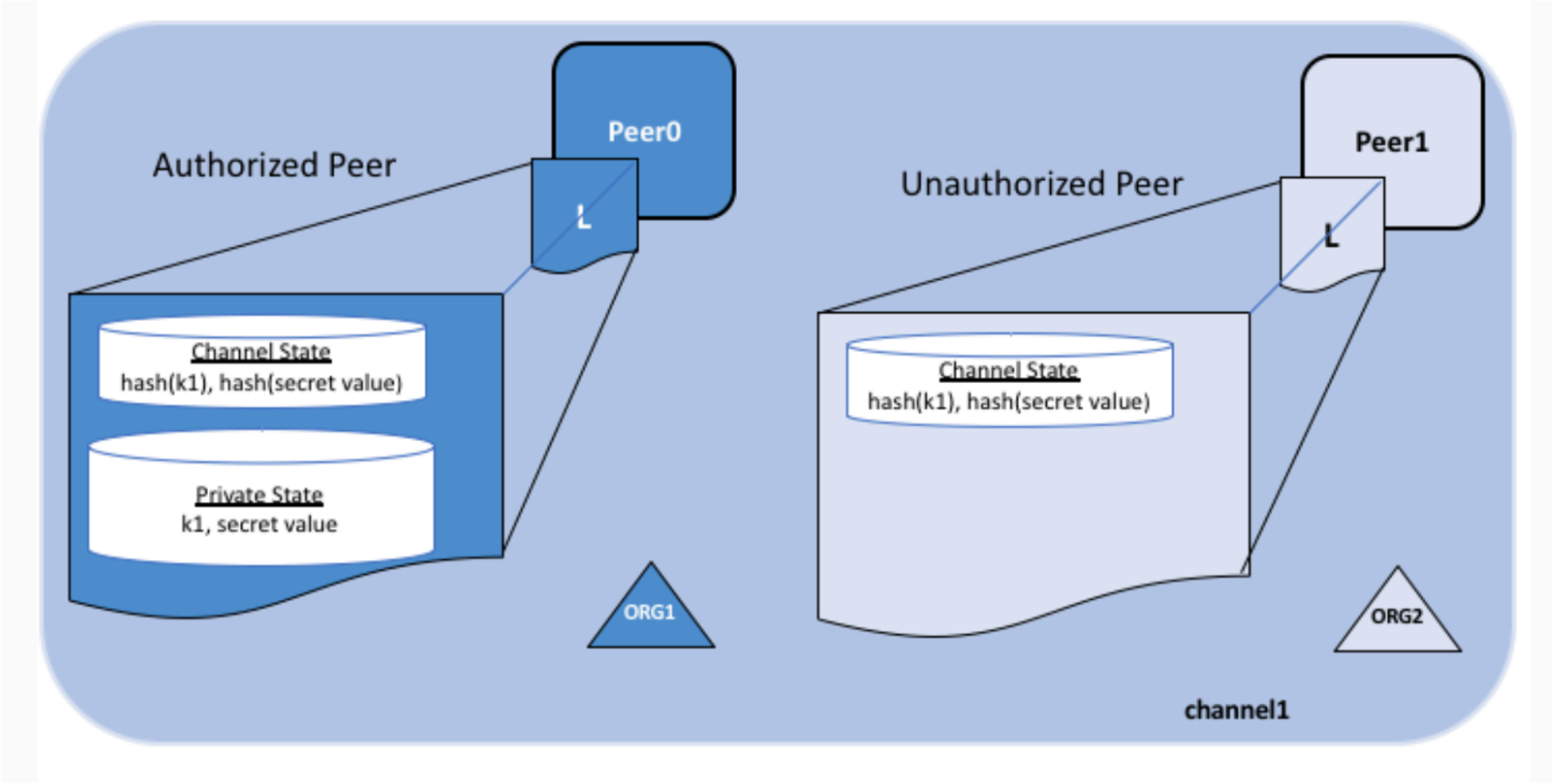
Hyperledger fabric is an enterprise-grade, open-source, Distributed Ledger Technology (DLT) platform used to develop enterprise applications leveraging the features of distributed ledgers and blockchain technology.




Source: https://hyperledger-fabric.readthedocs.io/en/release-2.2/

# Theoretical Background : Hyperledger Fabric

**Private data collections**, which allow a defined subset of peers on a channel the ability to commit, or query private data without having to create a separate channel.




Source: https://hyperledger-fabric.readthedocs.io/en/release-2.2/

# Requirements Analysis:

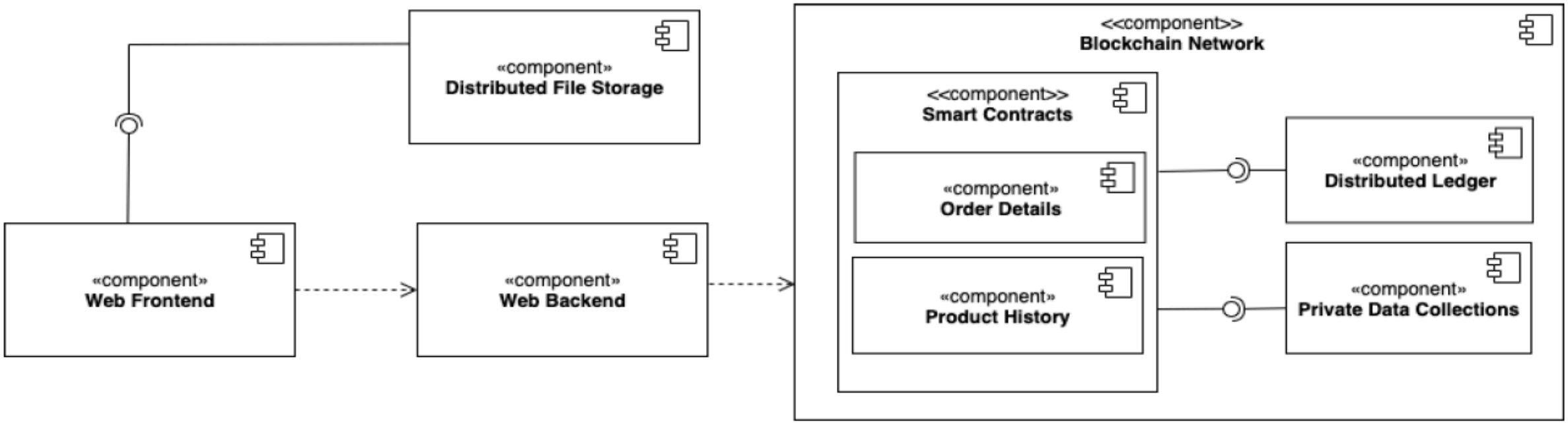
Details of the supply chain step is stored on the Blockchain and is approved by the parties involved

Trace the Product History

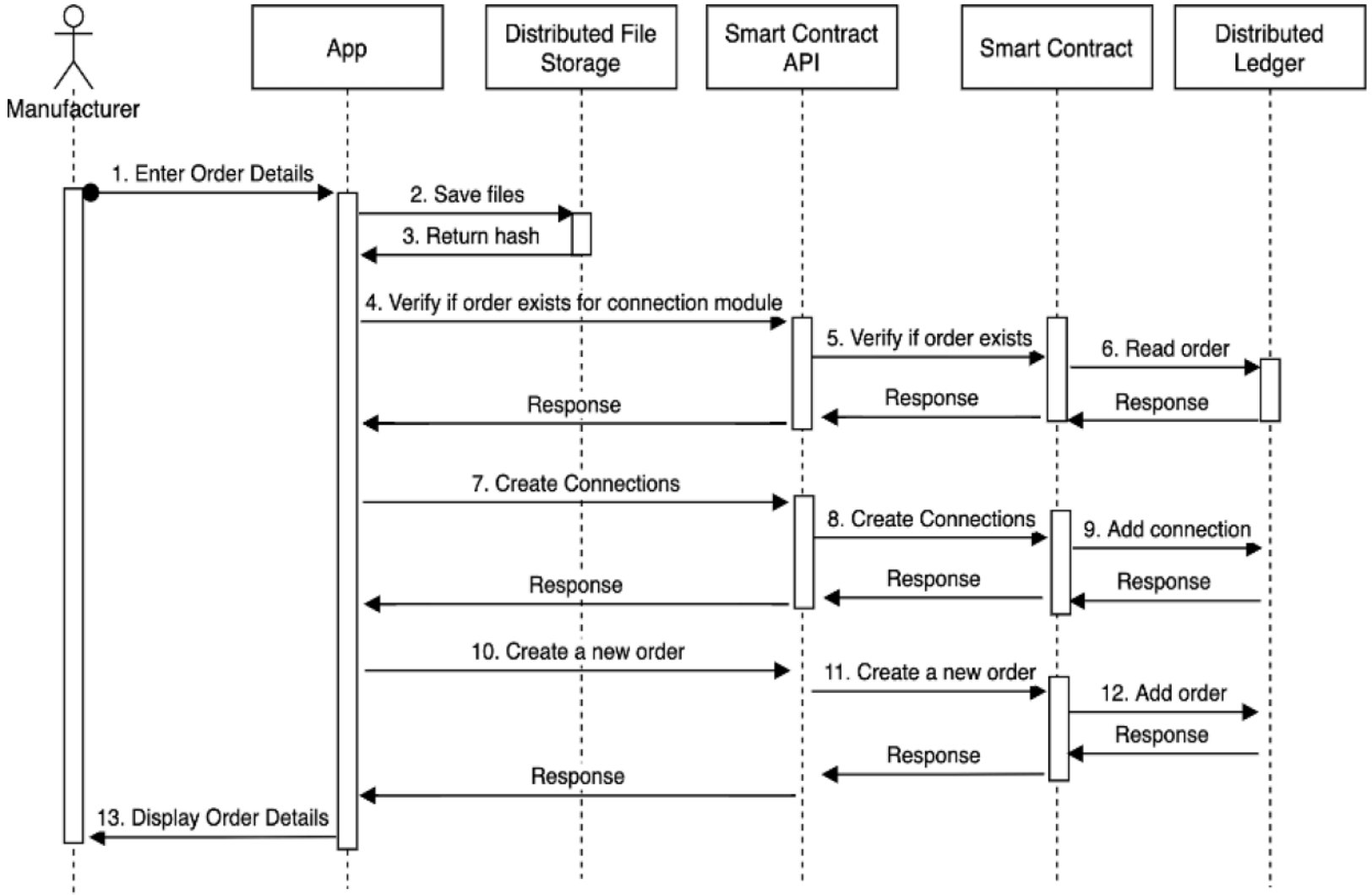
Deletion of sensitive information

Confidentiality from unauthorized parties

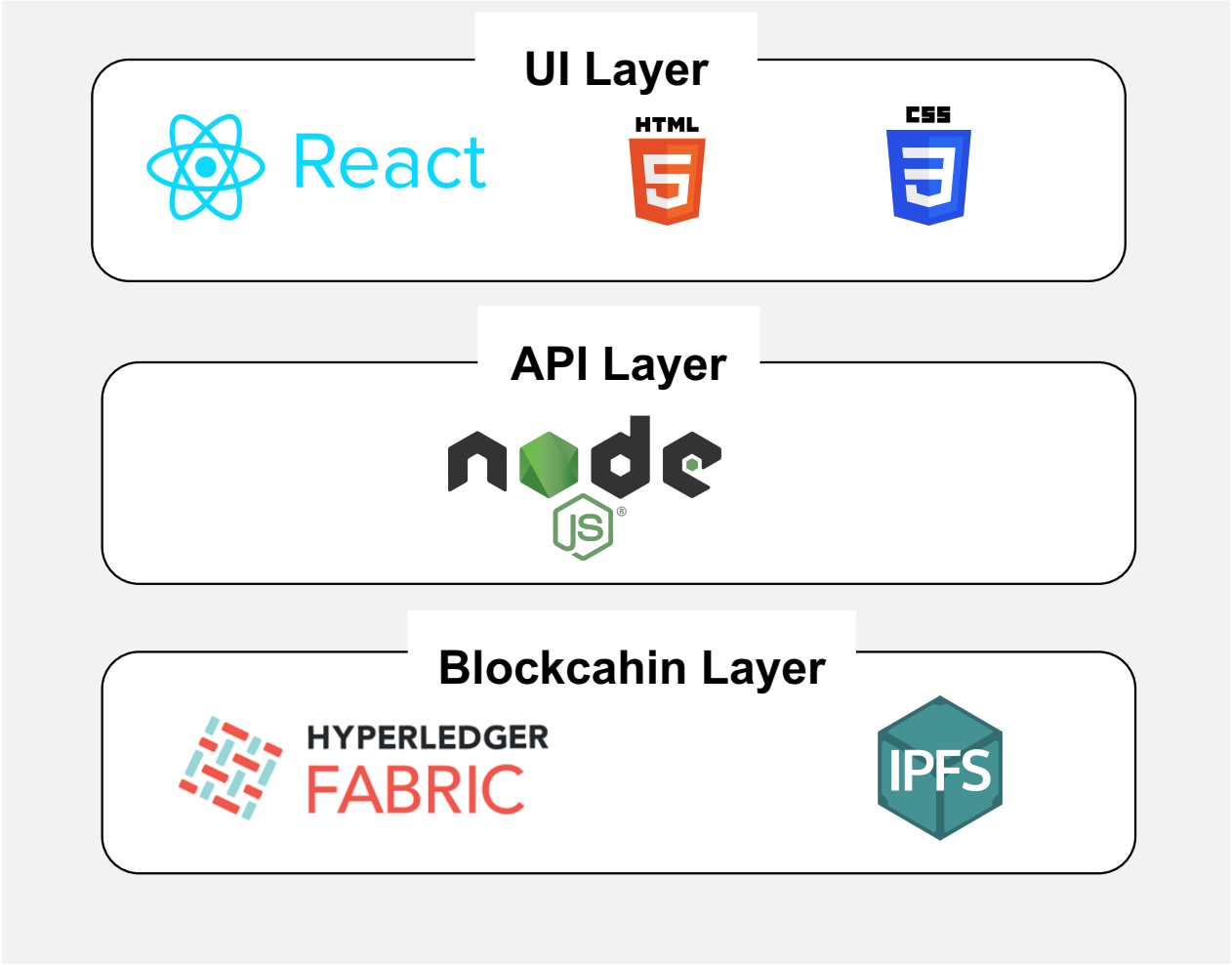
# Component Diagram

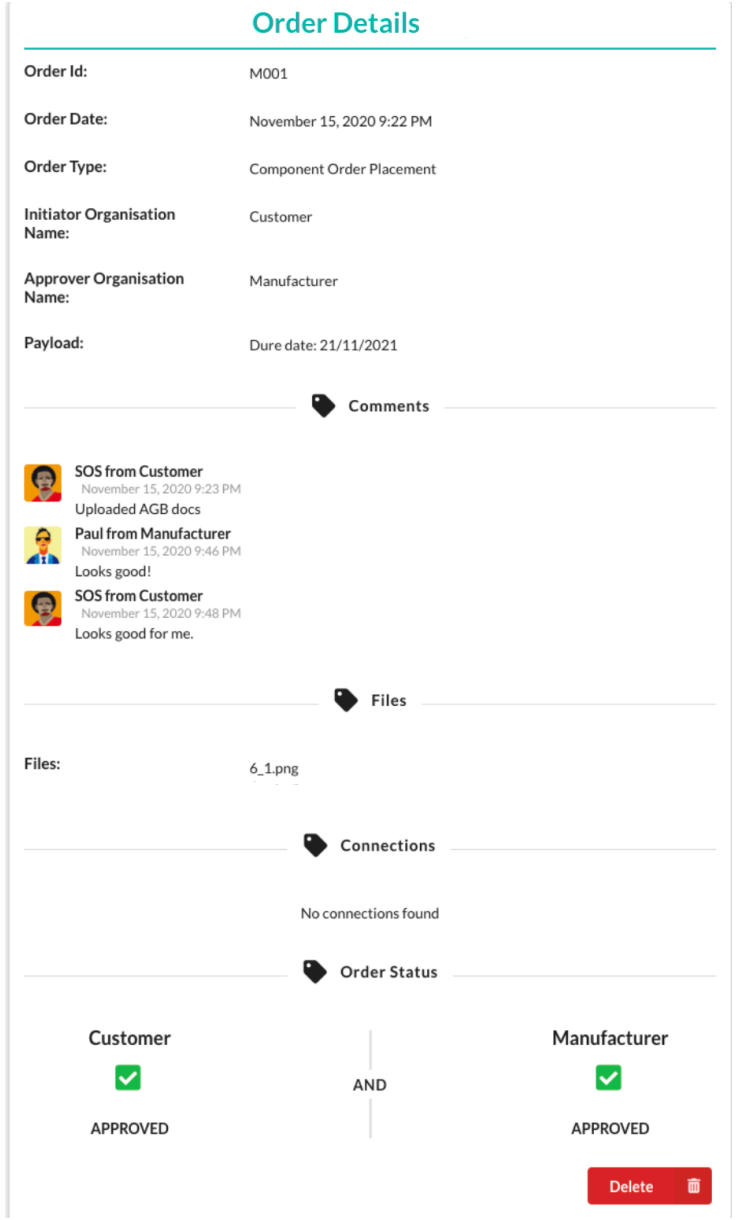


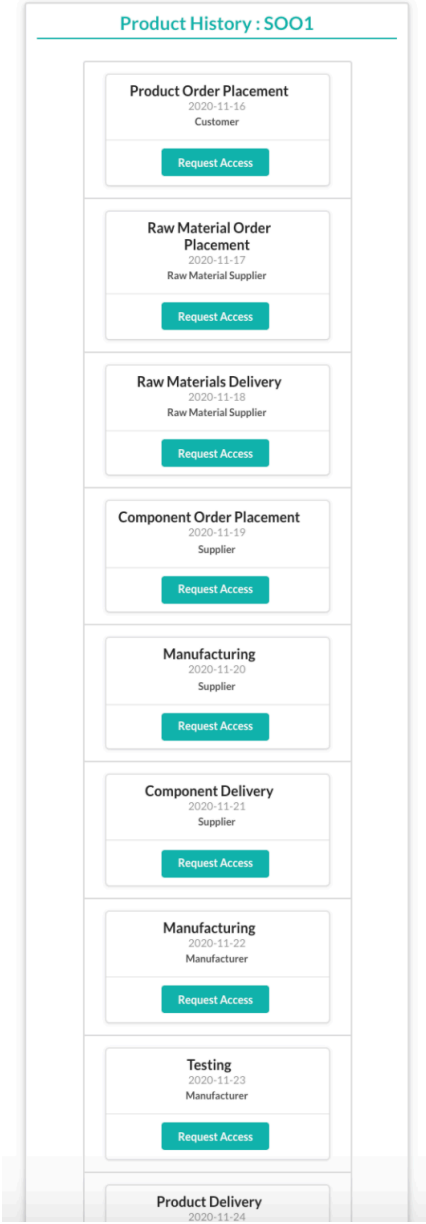
# Sequence Diagram



# Technology stack

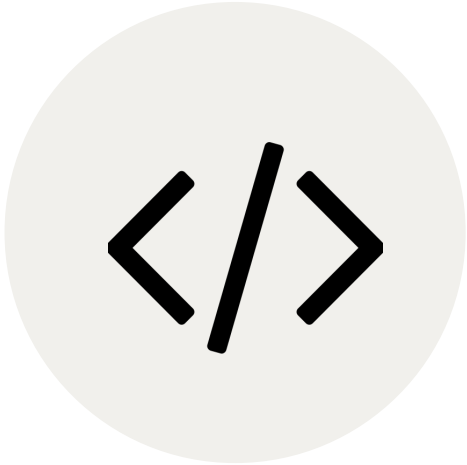



Order Details
Order Id:
M001
Order Date:
November 15, 2020 9:22 PM
Order Type:
Component Order Placement
Initiator Organisation Name:
Customer
Approver Organisation Name:
Manufacturer
Payload:
Dure date: 21/11/2021
Comments
SOS from Customer
November 15, 2020 9:23 PM
Uploaded AGB docs
Paul from Manufacturer
November 15, 2020 9:46 PM
Looks good!
SOS from Customer
November 15, 2020 9:48 PM
Looks good for me.
Files
Files:
6_1.png
Connections
No connections found
Order Status
Customer
APPROVED
AND
Manufacturer
APPROVED
Delete


Product History : SOO1
Product Order Placement
2020-11-16
Customer
Request Access
Raw Material Order Placement
2020-11-17
Raw Material Supplier
Request Access
Raw Materials Delivery
2020-11-18
Raw Material Supplier
Request Access
Component Order Placement
2020-11-19
Supplier
Request Access
Manufacturing
2020-11-20
Supplier
Request Access
Component Delivery
2020-11-21
Supplier
Request Access
Manufacturing
2020-11-22
Manufacturer
Request Access
Testing
2020-11-23
Manufacturer
Request Access
Product Delivery
2020-11-24

# Evaluation

EVALUATION BASED ON
REFERENCE ARCHTECTURE



EVALUATION BASED ON
PROTOTYPICAL IMPLEMENTATION

# Conclusion

The proposed architecture solves the two main issues of complex supply chains:

- Provides transparent data flow between any 2 parties in the supply chain, allowing for reliable audit while maintaining confidentiality
- Provides a traceable history to guarantee the authenticity of the product

This protects both manufacturer and customer from attempted fraud or counterfeit.

# Limitations

- Requirements are confined to IBO
- Using IPFS doesn´t guarantee encryption, so if any unauthorized party illicitly obtains the hash to a file in the private data, they can access it.

# Future Work

1. Gathering requirements by interviewing other SMEs and parties in the supply chain.
2. Researching more on the data that needs to be uploaded on the ledger.
3. Implementing encryption and decryption logic for private files uploaded on IPFS.


**Sangeeta Joseph**

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

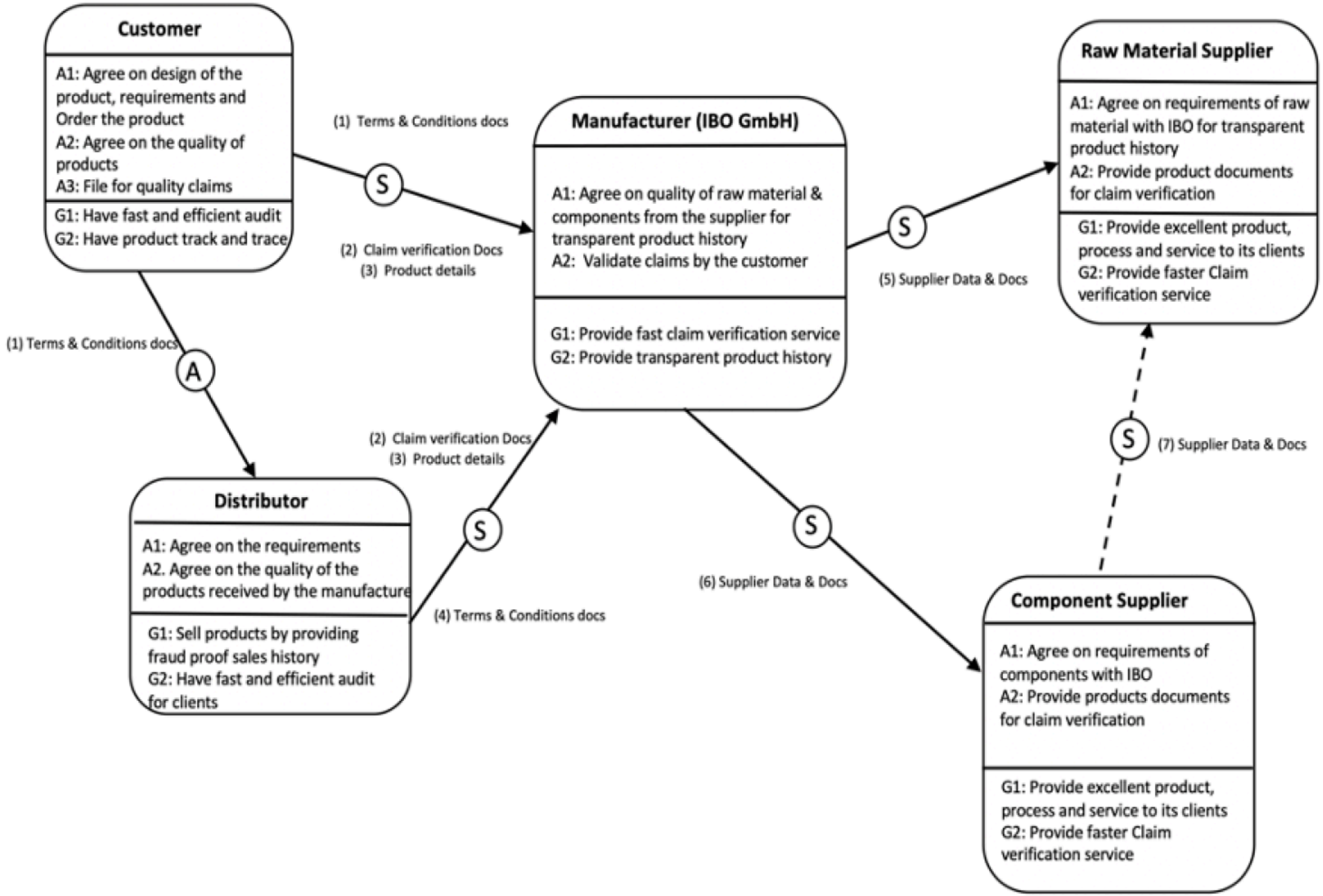
Tel +49.89.289. 17132
Fax +49.89.289.17136

sangeeta.joseph@tum.de
wwwmatthes.in.tum.de

# Comparison between Ethereum, Corda and Hyperledger Fabric

| | Ethereum | Corda | Hyperledger Fabric |
|---|---|---|---|
| Purpose | Platform to build distributed apps | Solutions for Financial Industries | Modular platform to build enterprise solutions |
| Permissions? | Public blockchain | Permissioned blockchain | Permissioned blockchain |
| Consensus | Mining based on proof-of-work | Parties involved in the transaction make the decision | Pluggable consensus mechanism |
| Transaction rate and Performance | Slow | Fast | Fast |
| Smart contract language | Solidity | Java and Kotlin | Go, Node.js and Java |
| Tokenization | Using ERC20 token standard | Using token sdk | Custom implementation |

# Stakeholder Analysis



# Requirements

| | As a... | Id | I want/need to ... |
|---|---|---|---|
| System Requirements | System | R1 | Record user actions and store them in the ledger |
| | | R2 | Maintain an immutable list of transactions in the ledger |
| | | R3 | Add mechanism to limit access to the ledger |
| | | R4 | Expose the smart contracts using APIs |
| | | R5 | Authorize users to execute smart contracts |
| Transparent data flow Requirements | Consumer | R6 | Create new order |
| | | R7 | Update order details |
| | | R8 | Tag the Producer for approval |
| | | R9 | Approve/Reject order details changes from the Producer |
| | | R10 | View a list of all orders |
| | | R11 | Restrict access to the order uploaded only to the Producer |
| | | R12 | Restrict access to the order shared only to the authorized organizations |
| | | R13 | Delete the order. |
| | | R14 | Approve order access request |
| | | R15 | View list of order access requests that need approval |
| | Producer | R16 | Approve/Reject order details agreement |
| | | R17 | Update the order details |
| | | R18 | View a list of orders that need approval |
| | Organization | R19 | Raise an access request to view the details of an order. |
| | | R20 | View the list of raised access requests. |
| Product Tracking Requirements | Organization | R21 | Access the product history using the order id. |
| | | R22 | Upload order connections |