A Blockchain-based Concept and Implementation for Machine Identity and Machine-to-machine Communication

Master Thesis Defense

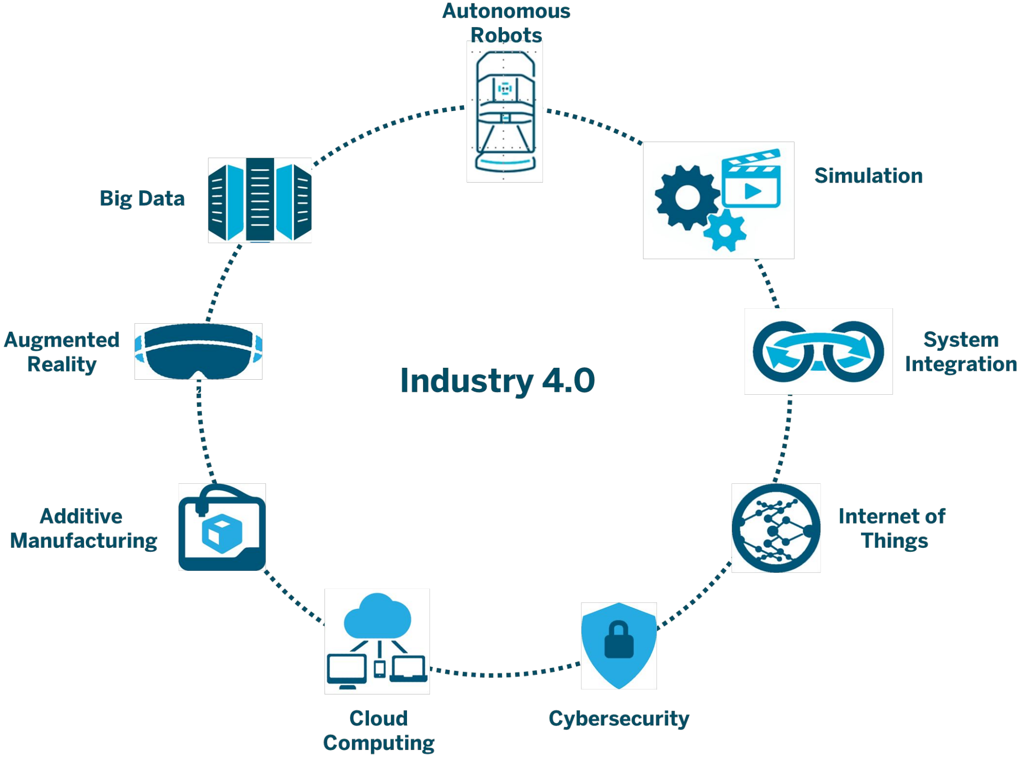
Mohammed Ghanem
Supervisor: Prof. Wolfgang Prinz

19/02/2021

Outline

1. Introduction
2. Research focus
3. Related work
4. Conceptual design & modeling
5. Implementation & prototype
6. Evaluation
7. Live demo
8. Conclusion

Industry 4.0



https://aethon.com/mobile-robots-and-industry4-0/

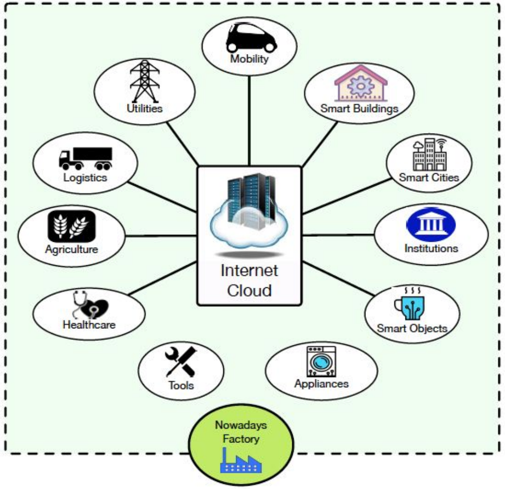
Challenges of Industry 4.0

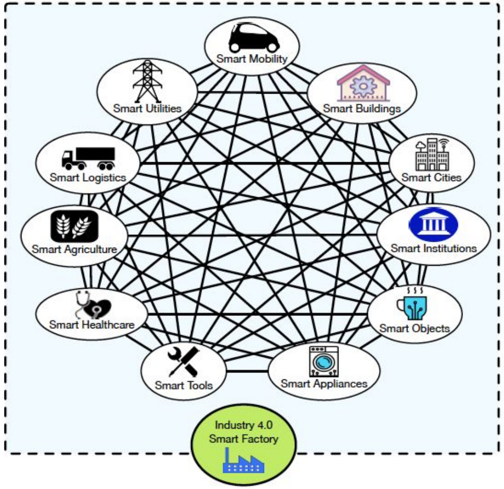
→ Centralization

→ Transparency

→ Security & privacy

→ Trustworthiness





https://ieeexplore.ieee.org/document/8678753

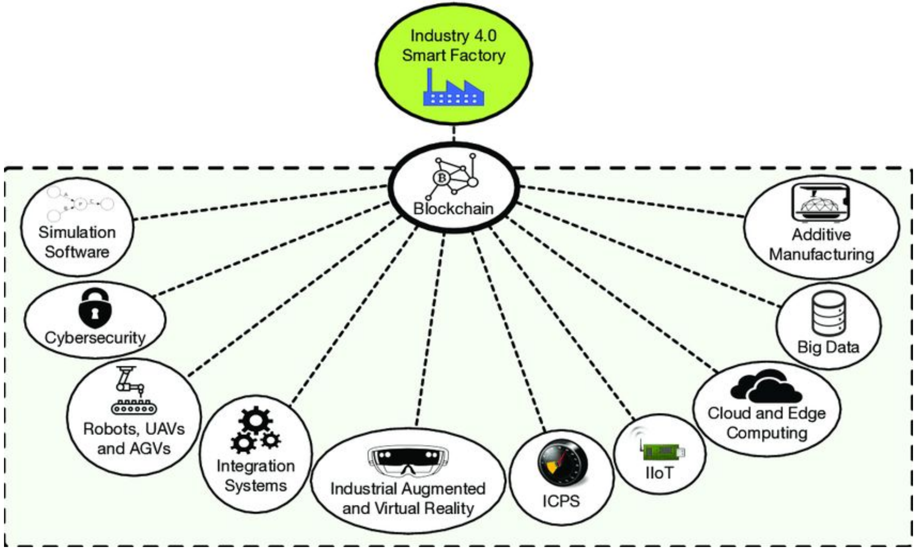
Solution

Blockchain (BC) technologies:

→ Decentralized

→ Distributed

→ Immutable



https://ieeexplore.ieee.org/document/8678753

Research Focus

Use blockchain inside factory to achieve the following:

→ Digital twins for machines and products

→ Blockchain-based digital identity for machines and products

→ Machine-to-Machine communication (M2M)

Research Questions

RQ1: How can blockchain technologies support M2M communication and digital-twins solution in the manufacturing industry?

RQ2: What are approaches for the design and development of blockchain-based solutions for M2M communication and digital identity?

RQ3: How can machines, products, and manufacturing processes be modeled on the blockchain?

RQ4: How feasible is it to utilize the blockchain for M2M communication and digital twin solutions?

Related Work

- Research Query:
  - Blockchain, Distributed Ledger, or Smart Contracts;
  - Manufacturing, Production, M2M, Factories, or Plants
- 7 research works:
  - Cloud manufacturing
  - M2M communication

ResearchGate

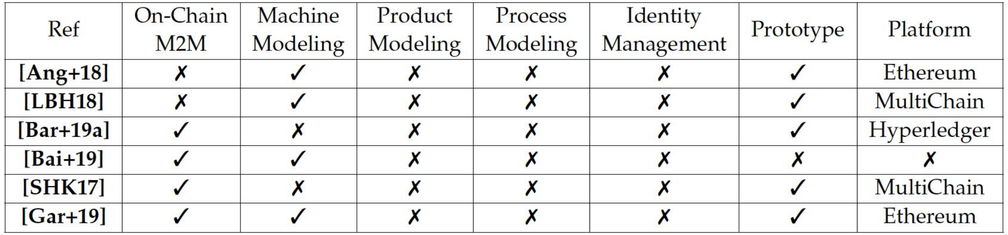
IEEE

arXiv.org

Google Scholar

Related Works Summary

- → Focused on horizontal integration
- → Ignored manufacturing processes
- → Small poor prototypes
- → Lack of identity standards



Ref	On-Chain M2M	Machine Modeling	Product Modeling	Process Modeling	Identity Management	Prototype	Platform
[Ang+18]	✗	✓	✗	✗	✗	✓	Ethereum
[LBH18]	✗	✓	✗	✗	✗	✓	MultiChain
[Bar+19a]	✓	✗	✗	✗	✗	✓	Hyperledger
[Bai+19]	✓	✓	✗	✗	✗	✗	✗
[SHK17]	✓	✗	✗	✗	✗	✓	MultiChain
[Gar+19]	✓	✓	✗	✗	✗	✓	Ethereum

Objectives

→ Build a digital twin for machines and products using the blockchain

→ Provide a blockchain-based digital identity for the digital twins

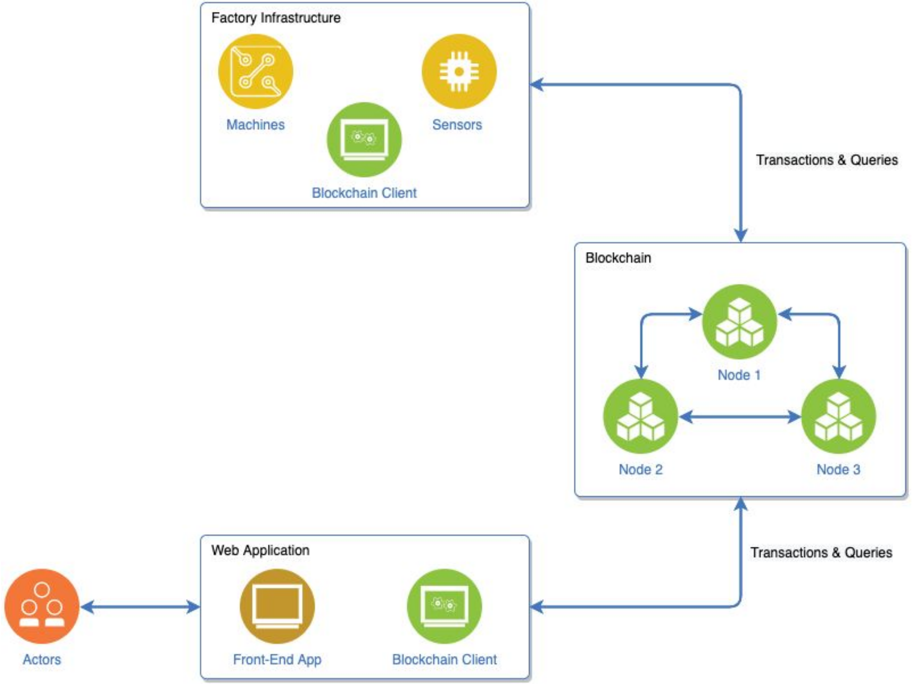
→ Model and execute the processes and their business logic using the blockchain

→ Build a user interface to interact and access the blockchain’s information

→ The design should be generic and replicable to different use cases

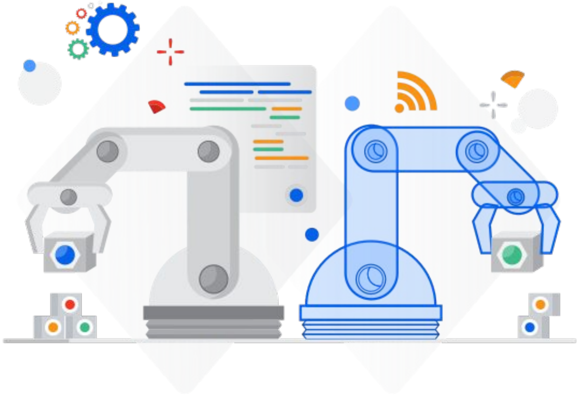
High-Level Overview

- Compounds
  - Factory Infrastructure
  - Blockchain
  - Web Application
- Actors
  - Machine
  - Process
  - Machine Owner
  - Process Owner
  - Product Owner



Machine Modeling

- → Any factory component regardless of its size
- → Multiple tasks types in multiple processes
- → One smart contract for each machine
- → The smart contract represents the machine digital twin
- → Store & validate information about the machine



https://www.xenonstack.com/digital-twin/

Machine Smart Contract

- Store information about:
  - Identity
  - Basic Information
  - Processes
  - Tasks
  - Products
  - Readings
  - Alerts
- Validate information to:
  - Check products requirements
  - Check sensor readings values

https://thenounproject.com/term/smart-contract/2199759/

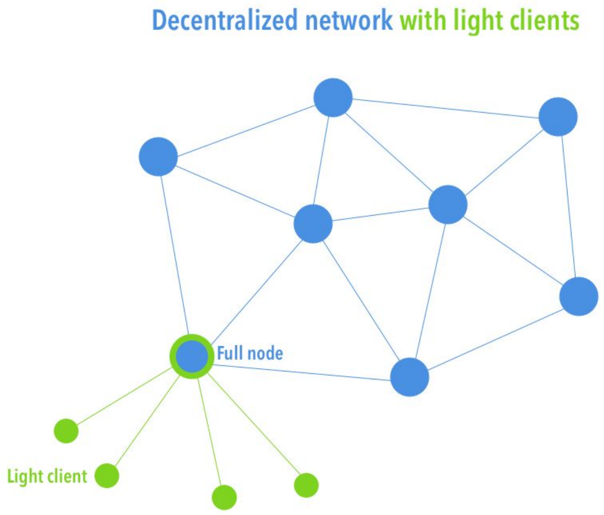
Twins Interaction

- Two directions interaction
- From the physical machine to digital twin
- From the digital twin to the machine

https://www.emersonautomationexperts.com/2019/simulation/advancing-digital-twin-technology/

From Machine to Digital Twin

- → Machine can not be a full node
- → The machine has a client runs in the gateway
- → Interact with the machine digital twin
- → Sign transactions for:
    - ○ Start task
    - ○ Finish task
    - ○ Store reading
    - ○ Store product operation



https://medium.com/metamask/metamask-labs-presents-mustekala-the-light-client-that-seeds-data-full-nodes-vs-light-clients-3bc785307ef5

From Digital Twin to Machine

- Interaction done using events
- Machine smart contract emits different types of events
- Client takes actions when receiving:
  - Task assigned event
  - Send reading event

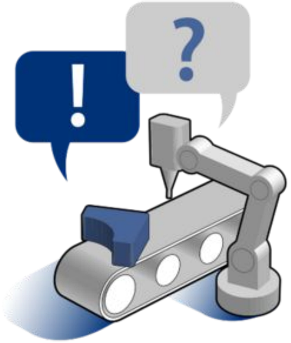
Product Modeling

- Non-compounded products
- All products are modeled within one smart contract
- The smart contract represents the products digital twins
- Store & validate information about the product:
    - Identity
    - Operations
    - Processes

https://thenounproject.com/term/smart-contract/2199759/

Manufacturing Process Modeling

- Simple fixed process with no sub-processes
- One smart contract for each process
- The contract assigns tasks to the machines’ digital twins
- Store information about:
    - Machines
    - Process instances

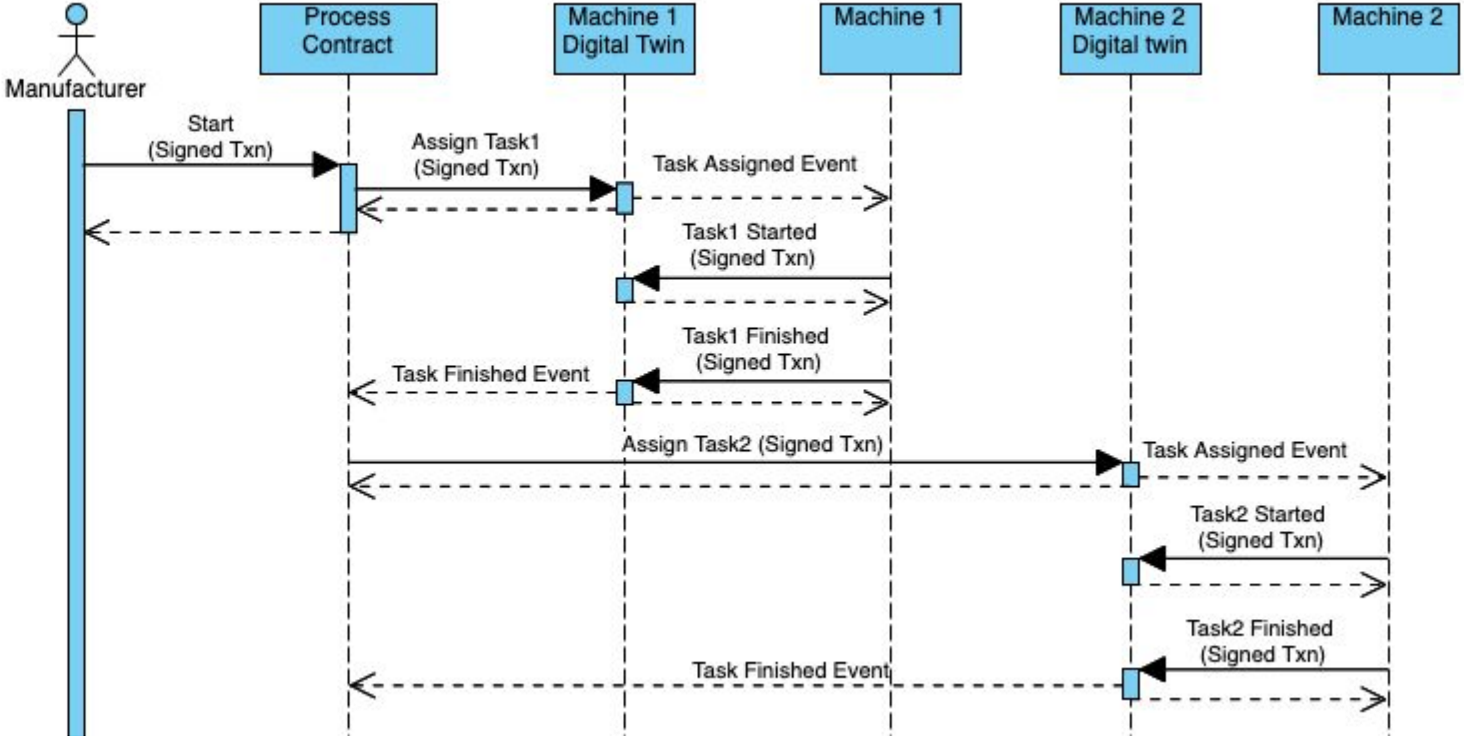


https://www.hbm.com/en/6589/machine-to-machine-communication-optimizes-production-processes/

Process Execution

- Process contract emits the following events:
  - Process started event
  - Process step started event
  - Process step finished
- Similar to the twins' interaction. Done with the help of a client runs in the gateway
- The client of the process is listening to the events emitted by the machines’ contracts, and the process contract

Execution Example

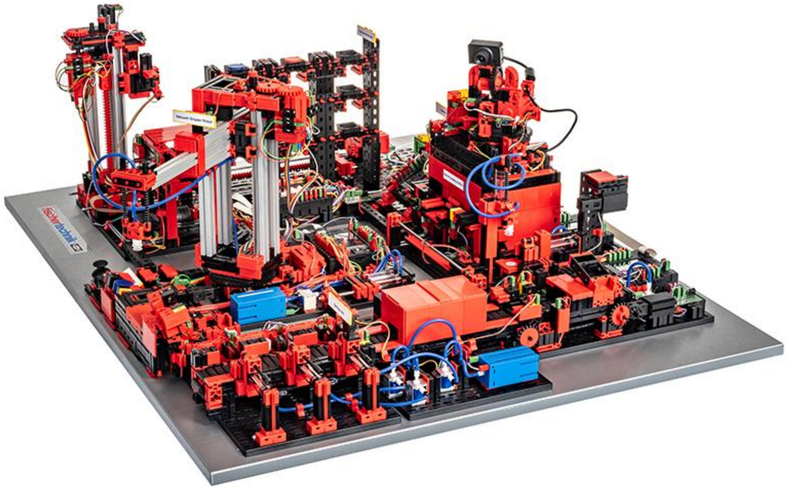


Identity Management

- Self-sovereign identity using W3C standards
- DIDs:
    - Ethr DID method developed by uPort
    - Lightweight identity using ERC-1056
    - Each machine, product has an blockchain address which is a valide DID
- VCs:
    - Machine issues credential for every product operation
    - Off-chain JSON Web Tokens (JWT) credentials using did-jwt

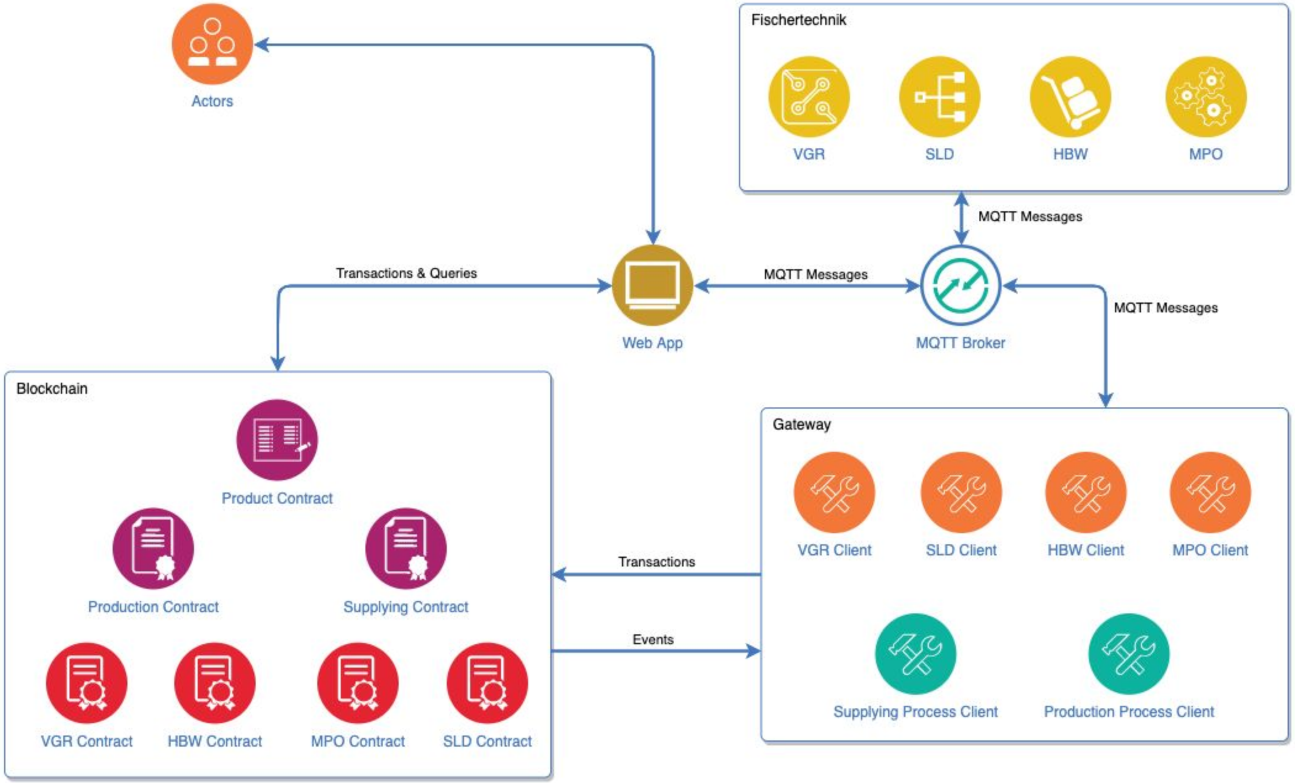
Case Study

- Fischertechnik factory simulation
- Machines:
  - **HBW**: High-Bay Warehouse
  - **VGR**: Vacuum Gripper Robot
  - **MPO**: Multi-Processing Station with Oven
  - **SLD**: Sorting Line
- Processes:
  - Supplying process
  - Production process



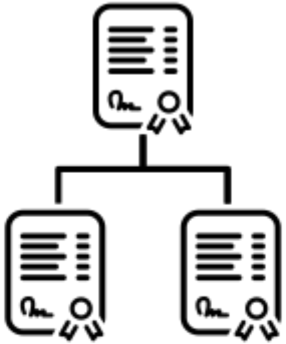
https://www.fischertechnik.de/en/products/teaching/training-models/551584-edu-training-factory-industry-4-0-9v-education

Architecture



Implementation - Smart Contracts

- Define one abstract contract for machines and processes
- No business and custom logic
- All machines and processes will have the same interface
- The subcontracts must override the certain functions



Evaluation

- The prototype is an evaluation for the design and the modeling
- The design fulfilled our objectives:
  - Build machine digital twin
  - Build product digital twin
  - Use Blockchain-based identities for twins
  - Execute the manufacturing processes over the blockchain
  - Generic design

Limitations

- Digital twins:
  - Only subset of the information in the operational phase
- Manufacturing processes:
  - Process execution order is hardcoded
  - Manually assign machines
- Technical:
  - Gateway
  - Throughput and Latency
  - Privacy

Live Demo

Conclusion

- Applicability of the blockchain in the manufacturing industry
- It is all about trust
- Data is more valuable when it is on-chain
- Feasible with assumptions and limitations
- Future Work
  - Overcoming the limitations
  - Model-driven engineering for processes modeling
  - Build services and applications e.g. warranty agreement

Appendices

Registry Modeling

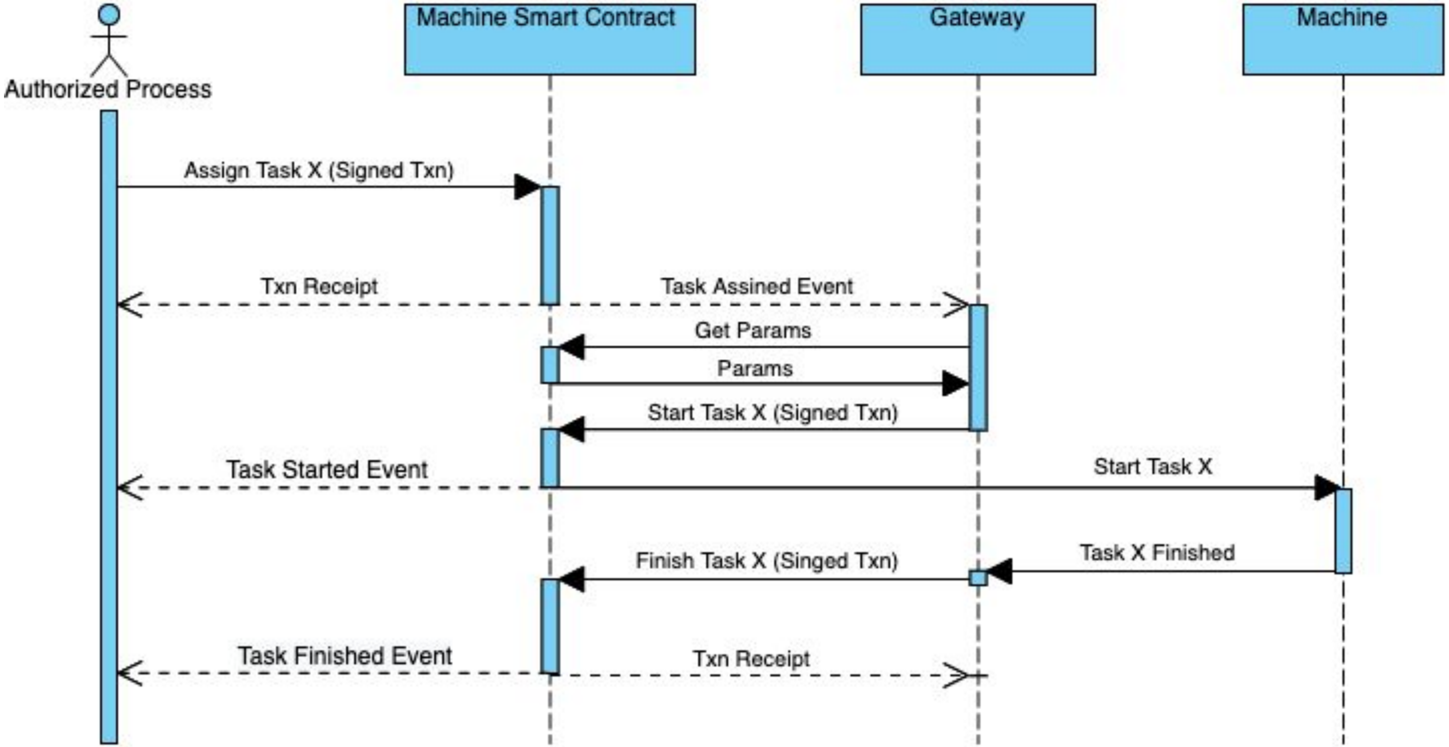
→ Added in a late stage of the thesis

→ On-chain registry

→ Maintains mapping between blockchain addresses and user-friendly names

→ Store the addresses of all the machines and processes

Interaction Example



Machine Contract Implementation

- Each subcontract contract overrides the following functions:
  - Get tasks types count
  - Get task type name
  - Get name
  - Assign task
  - Save reading
- Contains business and custom logic to validate:
  - Tasks params
  - Sensors readings values
  - Product operations

Process Contract Implementation

→ Each subcontract contract overrides the following functions:

- ○ Get number of machines
- ○ Get number of steps
- ○ Get machine number
- ○ Get step task type
- ○ Get name

→ Define function for each step and functions defined in the abstract class like:

- ○ Assign task
- ○ Authorize machine
- ○ Mark step as started

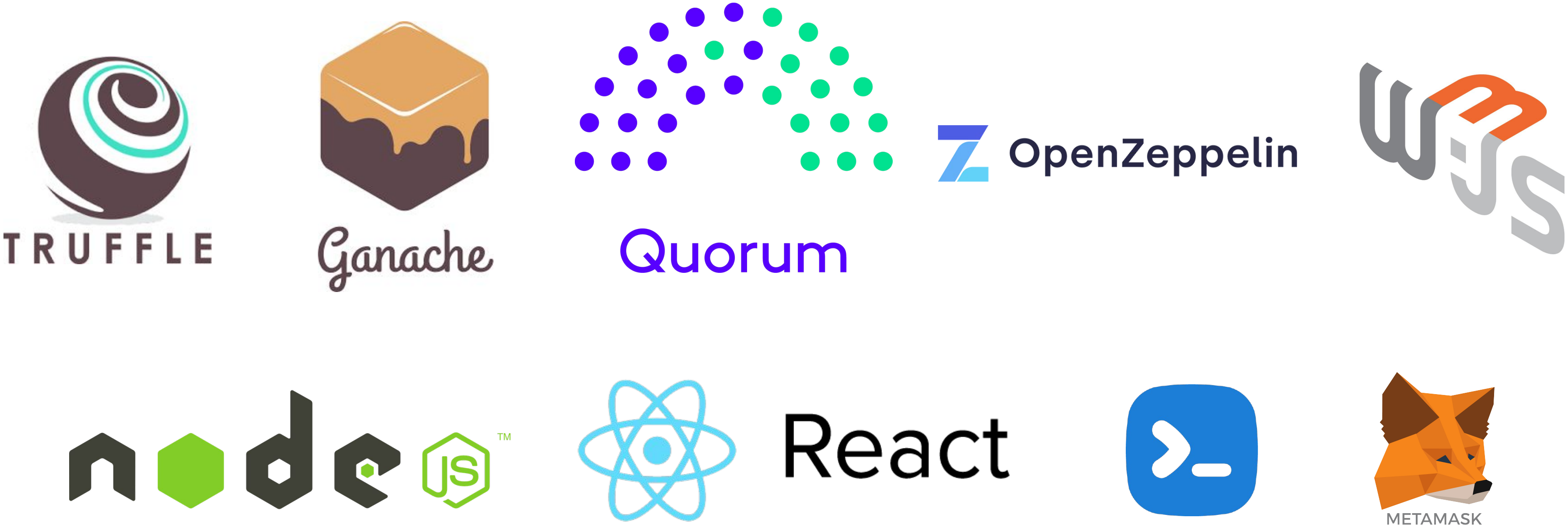
Other Contracts Implementation

- Product contract:
  - One contract for all products
  - Theoretically unlimited number of products
  - Each product is defined with a DID which is an Ethereum address
- Registry contract:
  - Map Ethereum addresses into strings
  - Automatically register the machines and processes by calling the getName function

Identity Implementation

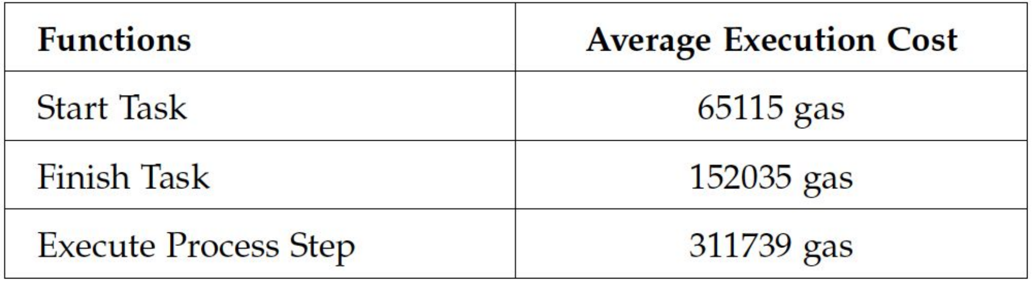
- DIDs:
  - Ethr DID method developed by uPort
  - Lightweight identity using ERC-1056
  - Each machine, product has an ethereum address which is a valide DID
- VCs:
  - JSON Web Tokens (JWT) credentials using did-jwt
  - Off-chain credentials signed by the machine

Development Tools and Frameworks



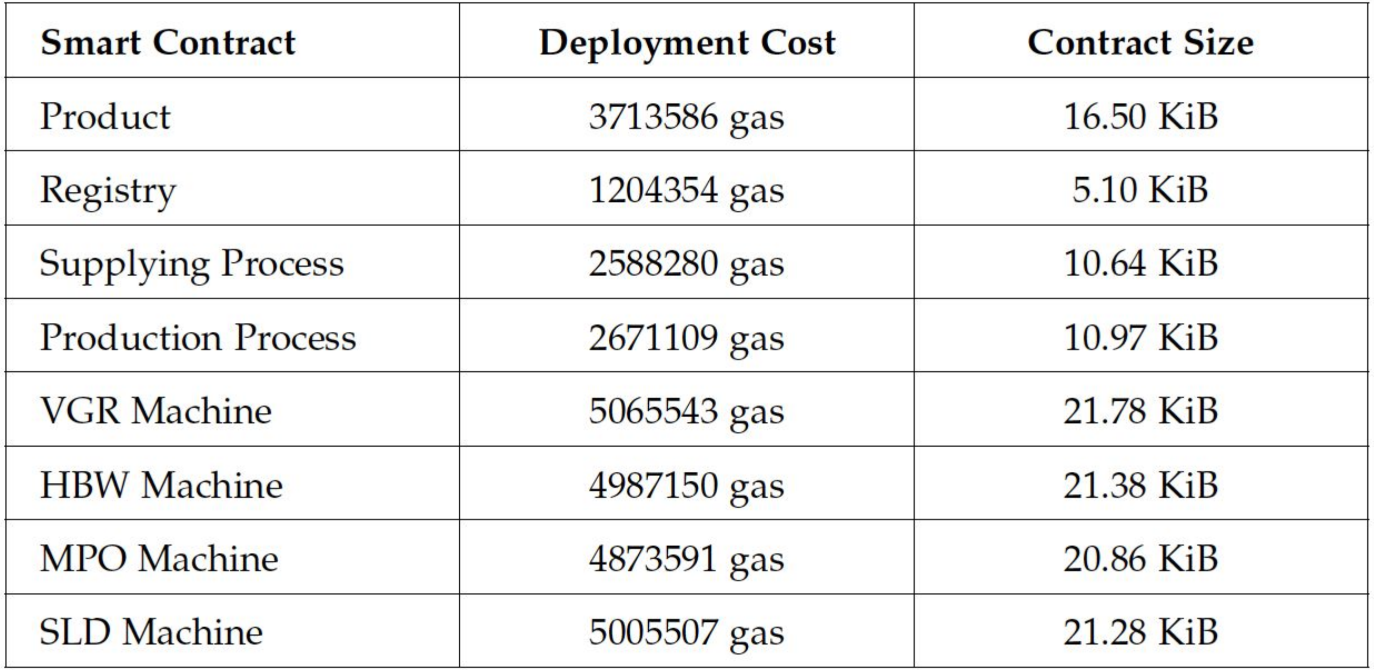
Quantitative Evaluation - Gas consumption

→ Measures execution cost in units of gas



Functions	Average Execution Cost
Start Task	65115 gas
Finish Task	152035 gas
Execute Process Step	311739 gas

Quantitative Evaluation - Contract Size



Smart Contract	Deployment Cost	Contract Size
Product	3713586 gas	16.50 KiB
Registry	1204354 gas	5.10 KiB
Supplying Process	2588280 gas	10.64 KiB
Production Process	2671109 gas	10.97 KiB
VGR Machine	5065543 gas	21.78 KiB
HBW Machine	4987150 gas	21.38 KiB
MPO Machine	4873591 gas	20.86 KiB
SLD Machine	5005507 gas	21.28 KiB