# HYPERLEDGER FABRIC

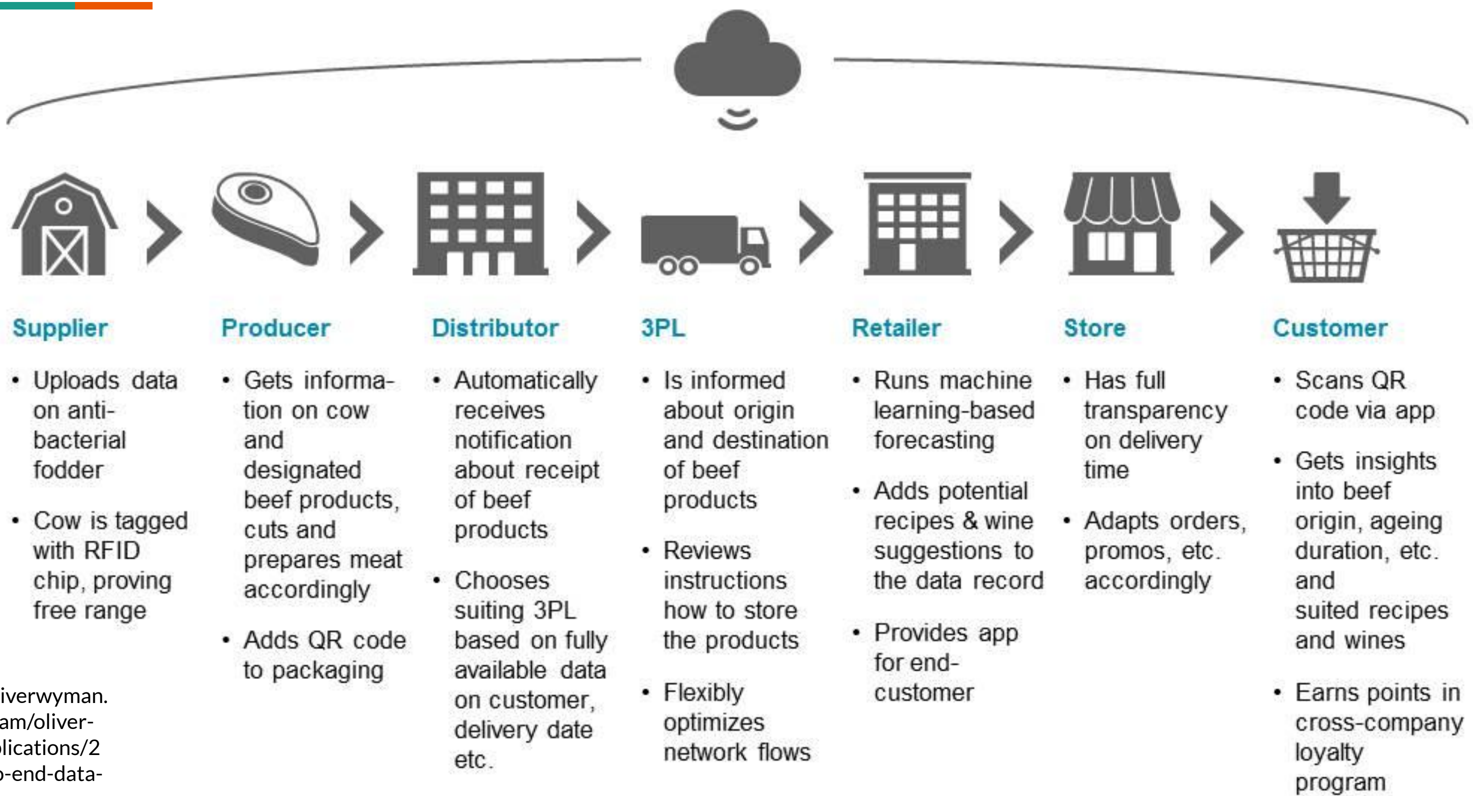
# FOOD SUPPLY CHAIN

Mitchell Dale (a1688826)

# HYPERLEDGER

- Is a **distributed** ledger platform that is comprised of **modular** components.
- Fabric differs from traditional blockchain implementations as it is **private** and **permissioned**.
- **Flexible** and hence **complex** system as the developer can control each aspect.
- **Developed** by **IBM** as a blockchain solution for **business enterprises**.

# Overview - Supplychain



**Supplier**

- Uploads data on anti-bacterial fodder
- Cow is tagged with RFID chip, proving free range

**Producer**

- Gets information on cow and designated beef products, cuts and prepares meat accordingly
- Adds QR code to packaging

**Distributor**

- Automatically receives notification about receipt of beef products
- Chooses suiting 3PL based on fully available data on customer, delivery date etc.

**3PL**

- Is informed about origin and destination of beef products
- Reviews instructions how to store the products
- Flexibly optimizes network flows

**Retailer**

- Runs machine learning-based forecasting
- Adds potential recipes & wine suggestions to the data record
- Provides app for end-customer

**Store**

- Has full transparency on delivery time
- Adapts orders, promos, etc. accordingly

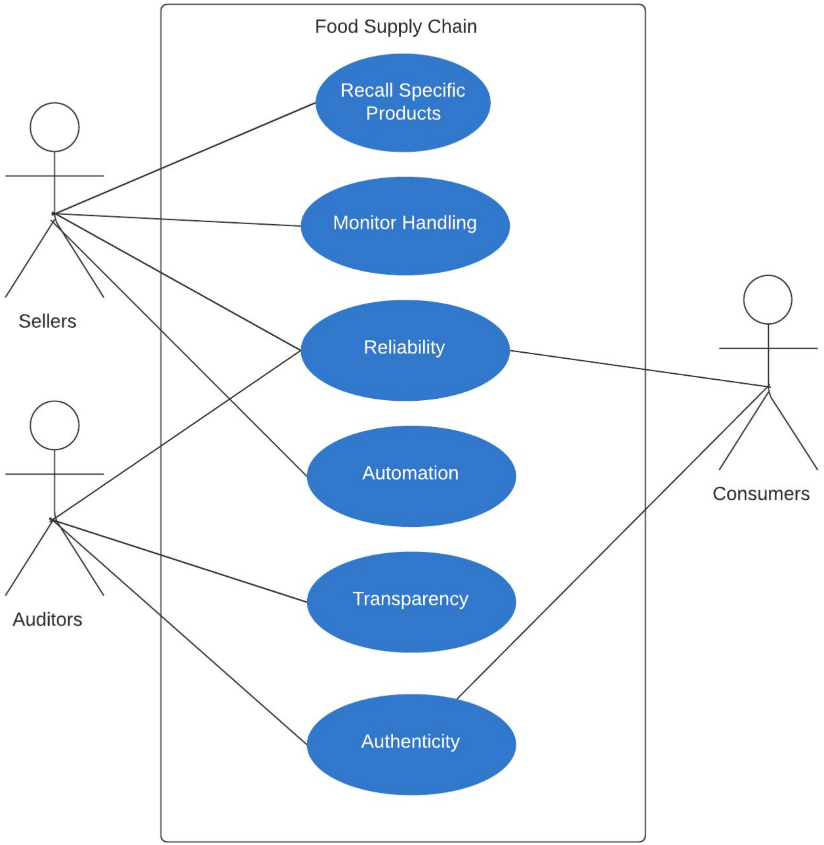
**Customer**

- Scans QR code via app
- Gets insights into beef origin, ageing duration, etc. and suited recipes and wines
- Earns points in cross-company loyalty program

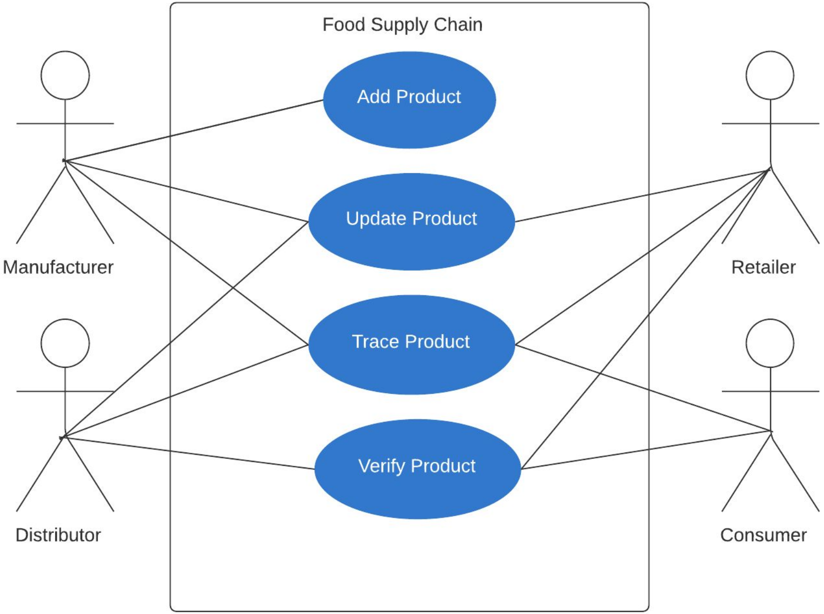
Ref: https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2017/jun/end-to-end-data-transparency.jpg

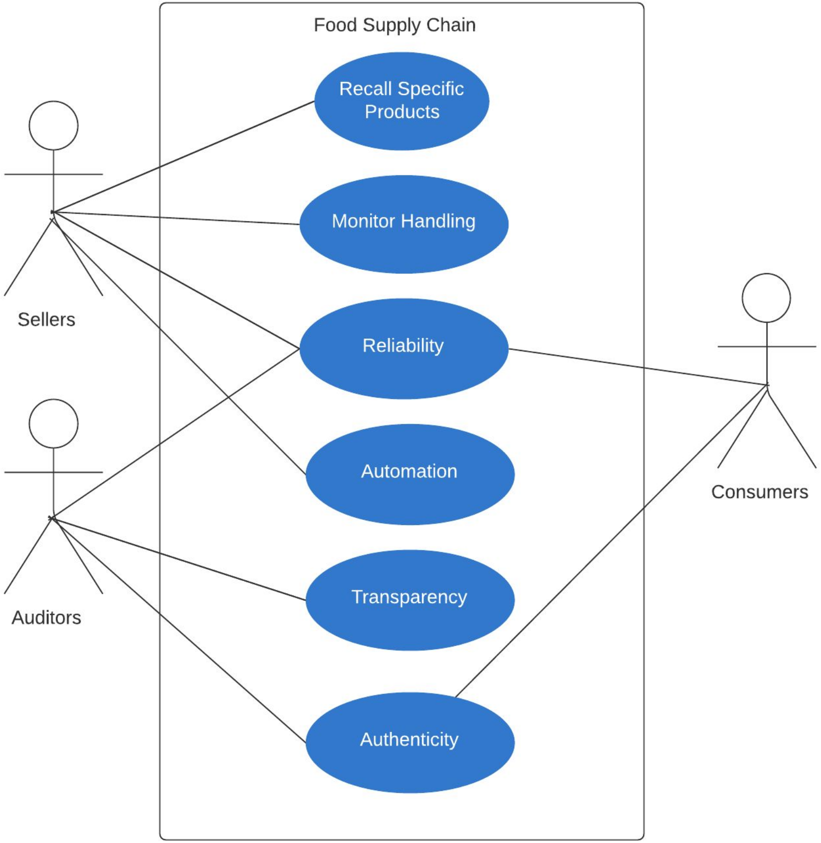
# Use Cases

- **Contaminated** food can be controlled
- Greater comfort in knowing **ethical** and **safety** origins of food
- **Distributed** in nature so no single point of failure
- **Auditable**

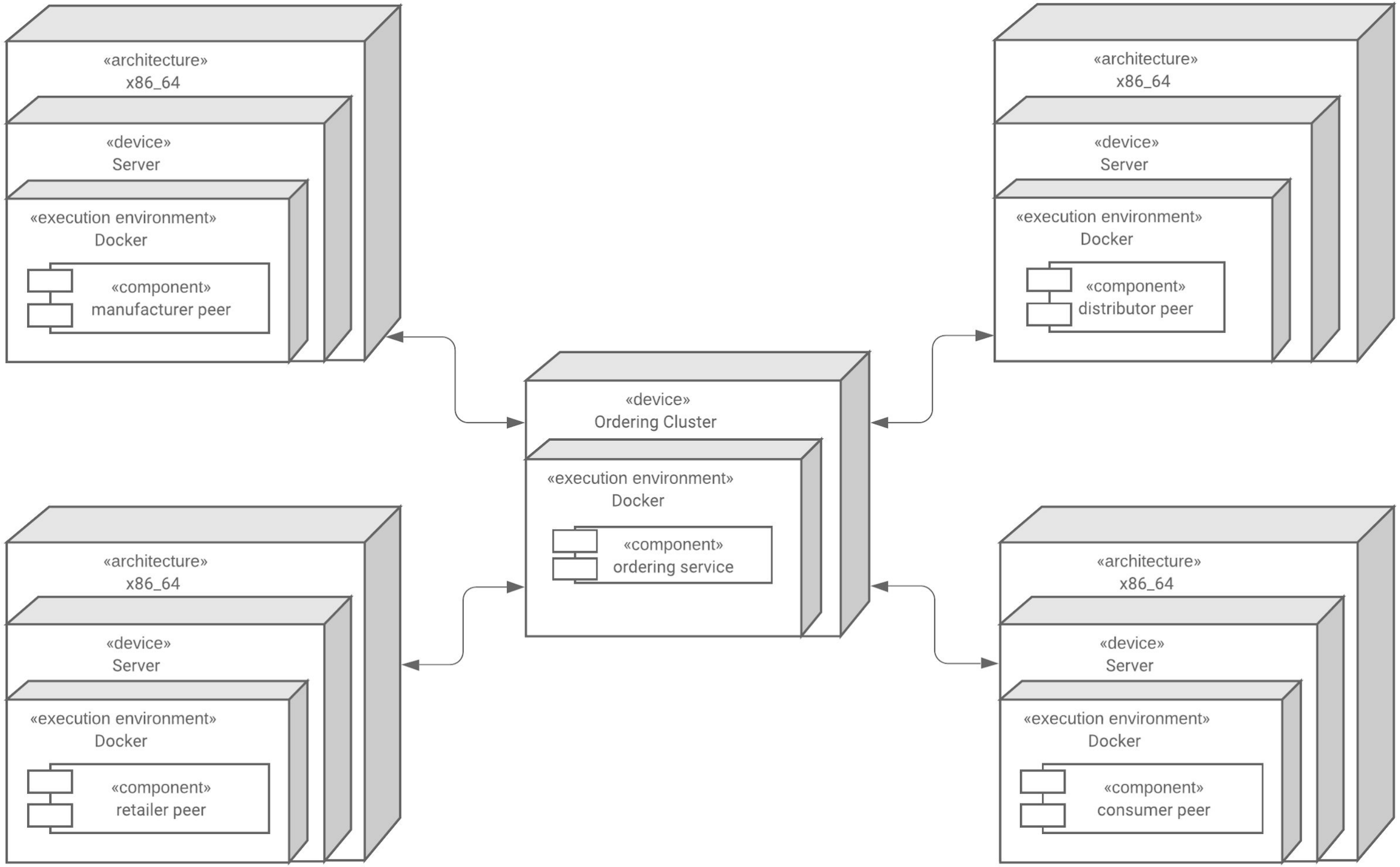


# Use Cases

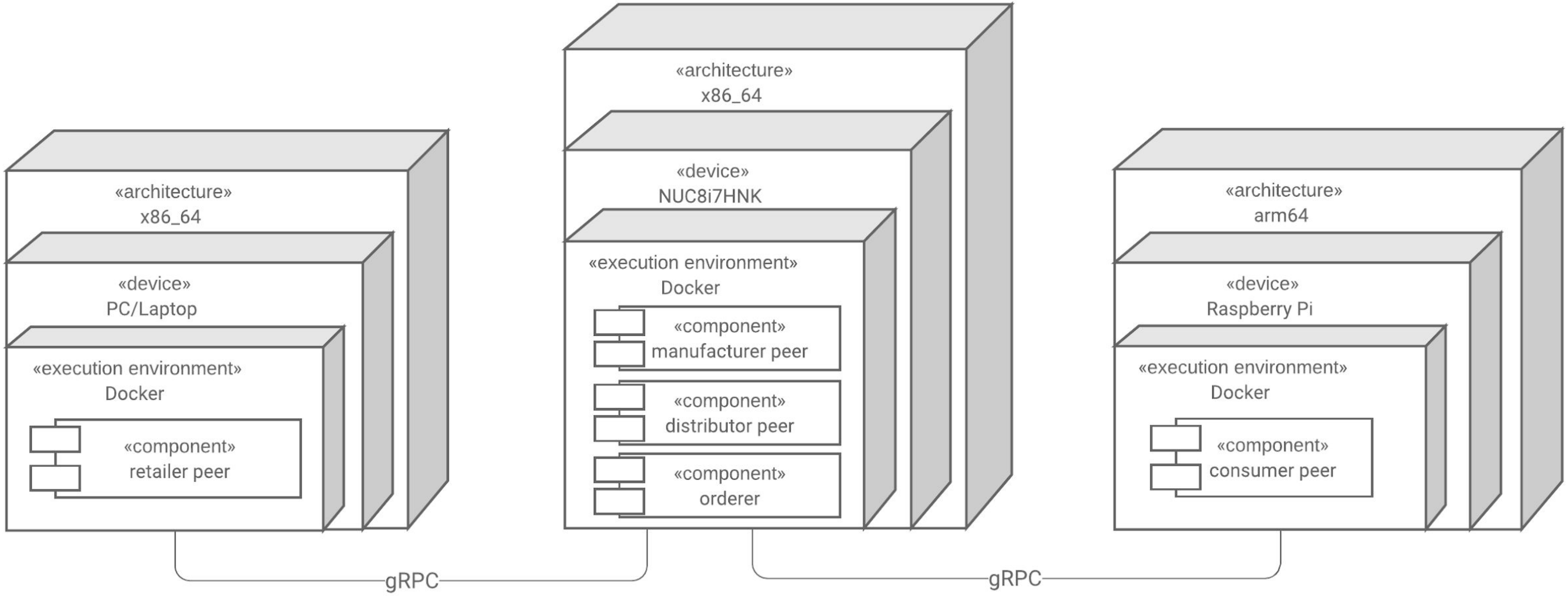

Food Supply Chain
Add Product
Update Product
Trace Product
Verify Product
Manufacturer
Distributor
Retailer
Consumer


Food Supply Chain
Recall Specific Products
Monitor Handling
Reliability
Automation
Transparency
Authenticity
Sellers
Auditors
Consumers

# Ideal


«architecture»
x86_64
«device»
Server
«execution environment»
Docker
«component»
manufacturer peer
«architecture»
x86_64
«device»
Server
«execution environment»
Docker
«component»
distributor peer
«device»
Ordering Cluster
«execution environment»
Docker
«component»
ordering service
«architecture»
x86_64
«device»
Server
«execution environment»
Docker
«component»
retailer peer
«architecture»
x86_64
«device»
Server
«execution environment»
Docker
«component»
consumer peer

# Current Deployment



# Protocol Design Space

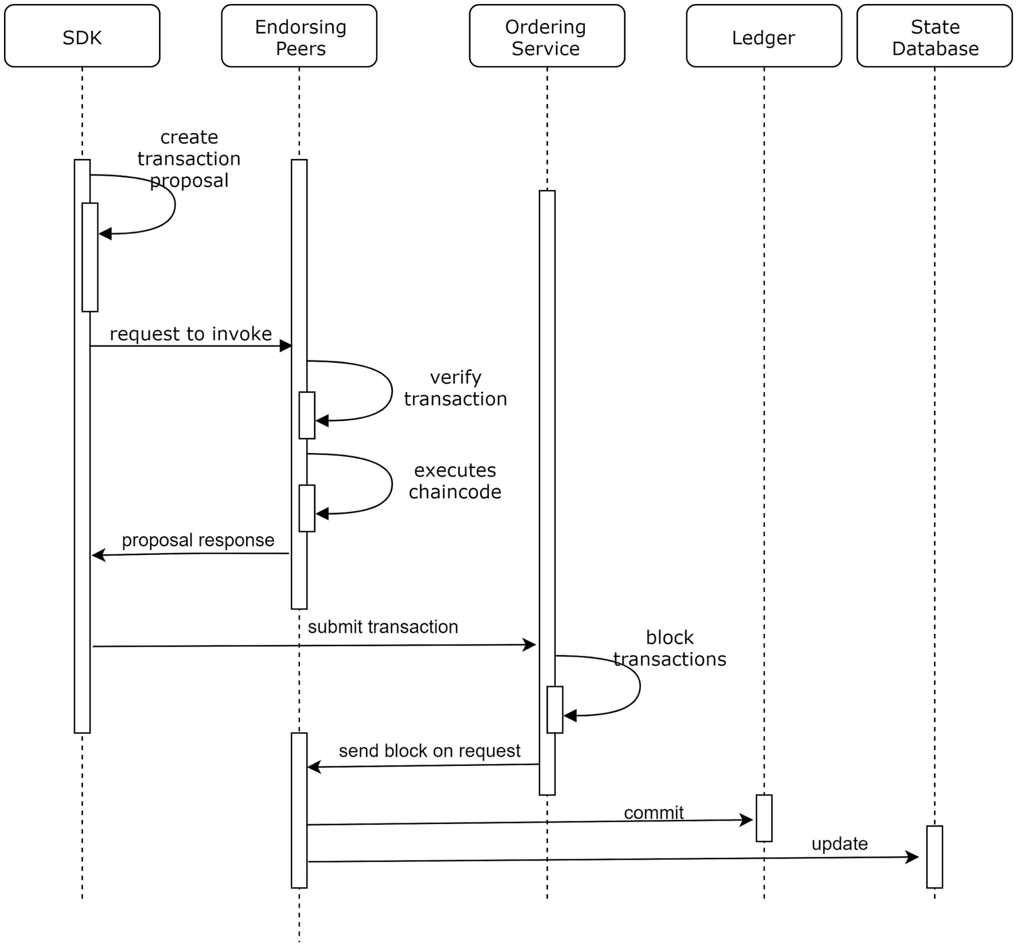
- Uses a **Versioned Key-Value Pairs** model
- Smart Contracts are **installed** on peers rather than on-chain
- Uses a **docker** virtual environment
- **Crash-tolerant** consensus protocol (RAFT)
- Identity model using **X.509 Cetificates**
- **Public** and **Private** key cryptography
- **Private Channels** for data preservation

# Network Design Decisions

- **Four** full nodes to represent the different organisations
- **One-to-one** relationship between participants and full node
- Deployed using **Networked Computers**
- **Gossip-based broadcast** between peers
- **Dynamic leader election** using gossip protocol
- Interact with blockchain using an **SDK**

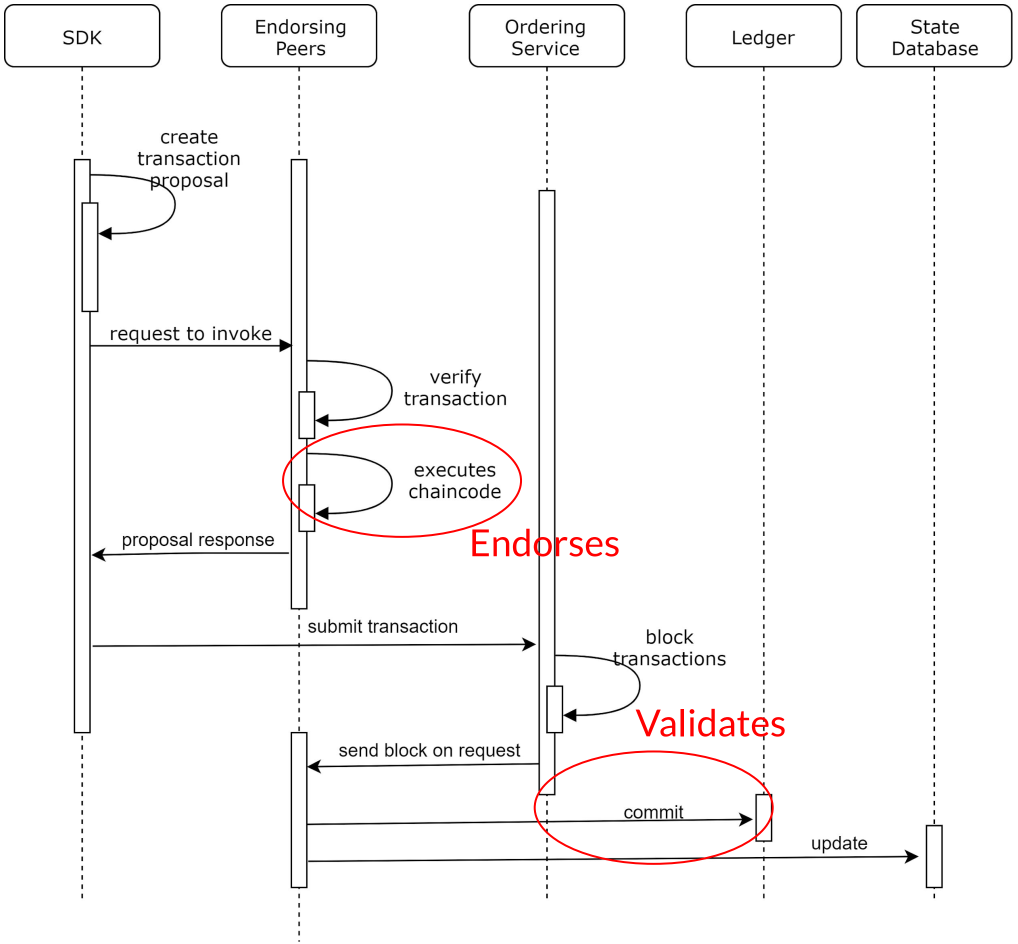
# Sequence

- Describing a **client** issuing an **update** to the blockchain.
- Clients are **authenticated** through the use of **identities**
- Identities are issued by **Membership Service Providers** and belong to **Organisations**



# Sequence

- Describing a **client** issuing an **update** to the blockchain.
- Clients are **authenticated** through the use of **identities**
- Identities are issued by **Membership Service Providers** and belong to **Organisations**



# Demonstration

# Performance Benchmarking

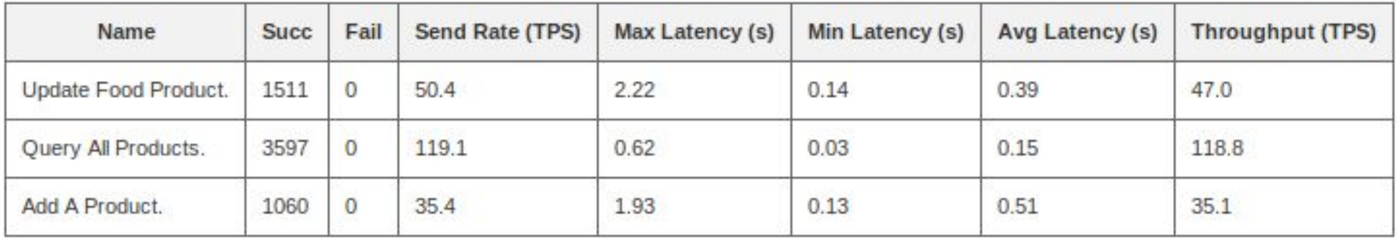
# Benchmarking Environment

- **Hyperledger Caliper** used to benchmark the blockchain network
- Using **fixed-backlog** rate controller
  - **Maximum of 5 Unfinished Transactions**
  - **Transaction timeout after 30 seconds**
- Testing performed on Network **with** and **without** Raspberry Pi

# Benchmarking Results without Pi

## Summary

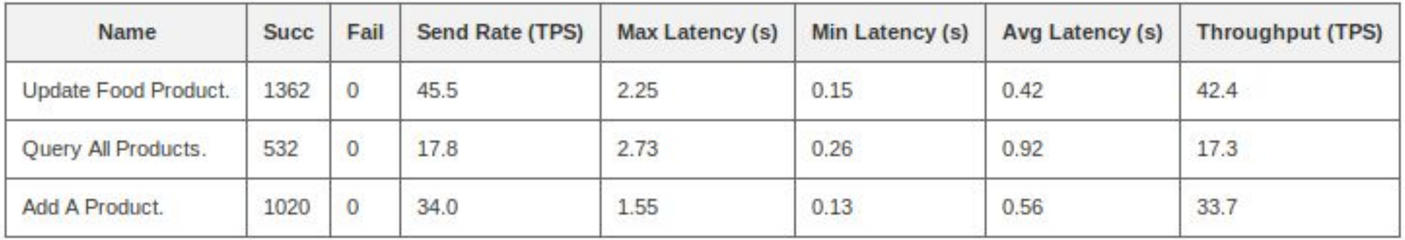
Running from Laptop



| Name | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|---|---|---|---|---|---|---|---|
| Update Food Product. | 1511 | 0 | 50.4 | 2.22 | 0.14 | 0.39 | 47.0 |
| Query All Products. | 3597 | 0 | 119.1 | 0.62 | 0.03 | 0.15 | 118.8 |
| Add A Product. | 1060 | 0 | 35.4 | 1.93 | 0.13 | 0.51 | 35.1 |

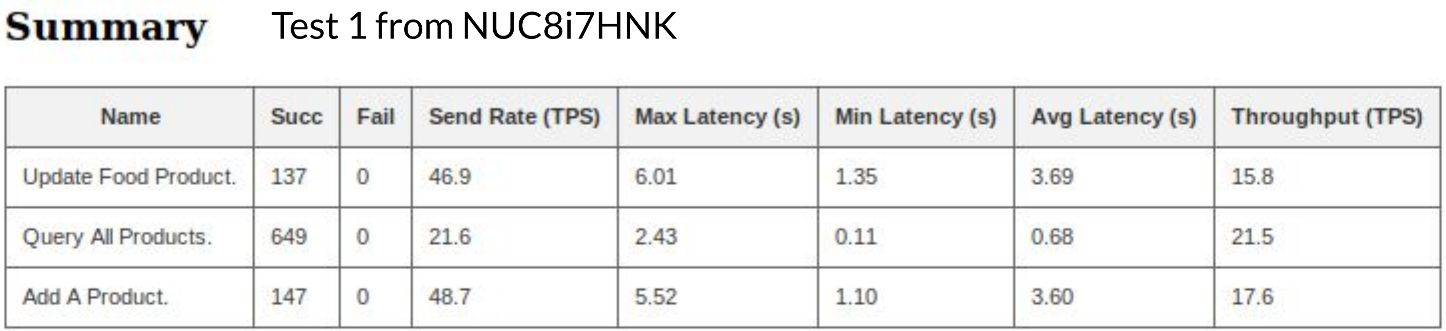
## Summary

Running from NUC8i7HNK



| Name | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|---|---|---|---|---|---|---|---|
| Update Food Product. | 1362 | 0 | 45.5 | 2.25 | 0.15 | 0.42 | 42.4 |
| Query All Products. | 532 | 0 | 17.8 | 2.73 | 0.26 | 0.92 | 17.3 |
| Add A Product. | 1020 | 0 | 34.0 | 1.55 | 0.13 | 0.56 | 33.7 |

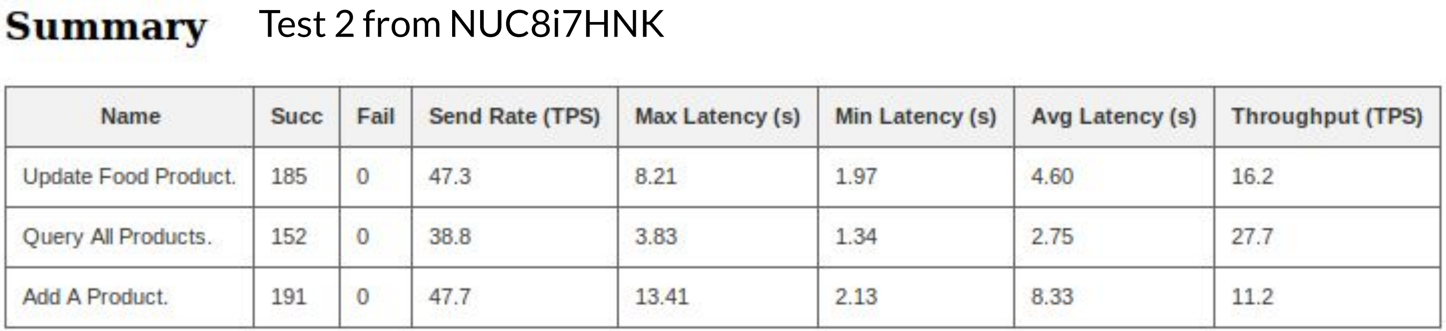
# Benchmarking Results with Pi



**Summary**

Test 1 from NUC8i7HNK

| Name | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|---|---|---|---|---|---|---|---|
| Update Food Product. | 137 | 0 | 46.9 | 6.01 | 1.35 | 3.69 | 15.8 |
| Query All Products. | 649 | 0 | 21.6 | 2.43 | 0.11 | 0.68 | 21.5 |
| Add A Product. | 147 | 0 | 48.7 | 5.52 | 1.10 | 3.60 | 17.6 |



**Summary**

Test 2 from NUC8i7HNK

| Name | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|---|---|---|---|---|---|---|---|
| Update Food Product. | 185 | 0 | 47.3 | 8.21 | 1.97 | 4.60 | 16.2 |
| Query All Products. | 152 | 0 | 38.8 | 3.83 | 1.34 | 2.75 | 27.7 |
| Add A Product. | 191 | 0 | 47.7 | 13.41 | 2.13 | 8.33 | 11.2 |