Smart Access Control Using Blockchain

A BTP report

Submitted in partial fulfillment of the

requirements for the award of the degree

of

Bachelor of Technology

Adwait Thattey (S20170010004)

Siddhant Jain (S20170010151)

Bagwan Mahammad Adam (S20170010021)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

17 December 2020

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**Smart Access Control Using Blockchain**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2020 to December 2020 under the supervision of Dr. Rajendra Prasath, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

(ADWAIT THATTEY)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Adwait Thattey has successfully completed his BTP Examination held on 17 December 2020

Signature of BTP Advisor

Date:

BTP Coordinator

Date:

Signature of BTP Panel Member:1

Date:

Signature of BTP Panel Member:2

Date:

Signature of External Examiner

Date:

---

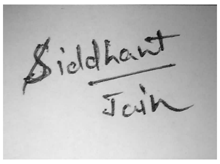
blank page

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**Smart Access Control Using Blockchain**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2020 to December 2020 under the supervision of Dr. Rajendra Prasath, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.



(SIDDHANT JAIN)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Siddhant Jain has successfully completed his BTP Examination held on 17 December 2020

Signature of BTP Advisor | BTP Coordinator

Date: | Date:

Signature of BTP Panel Member:1 | Signature of BTP Panel Member:2 | Signature of External Examiner

Date: | Date: | Date:

---

blank page

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**Smart Access Control Using Blockchain**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2020 to December 2020 under the supervision of Dr. Rajendra Prasath, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

(BAGWAN MAHAMMAD ADAM)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Bagwan Mahammad Adam has successfully completed his BTP Examination held on 17 December 2020

Signature of BTP Advisor — BTP Coordinator

Date: — Date:

Signature of BTP Panel Member:1 — Signature of BTP Panel Member:2 — Signature of External Examiner

Date: — Date: — Date:

ACKNOWLEDGEMENTS

We are immensely grateful to our BTP Mentor Dr. Rajendra Prasath for his guidance and direction throughout this project. We are thankful to our Institute, IIIT Sri City for providing us this opportunity to work on such an interesting topic as our B.Tech project. We are thankful to our mentor and the institute for supporting us and helping us in every way possible during this pandemic.

We are also thankful to countless researchers whose work gave us insights and ideas and helped us work out the solution to this problem.

We are thankful to the developers at IBM Corp and The Linux Foundation who worked on Hyperledger Fabric for maintaining the platform that we built our project upon. We are also thankful to the thousands of volunteer developers who helped maintain the project and the documentation.

Without the support of everyone mentioned above, we would not have been able to successfully complete this project in such a short time. We have learnt a lot while working on this topic and will continue to work on it even after the end of the project.

Contents

blank page

Introduction

In Information Security, Access Control is a security technique that regulates who or what can view or use resources in a computing environment. With the developments of internet and computer hardware, more and more devices are connected with each other through wireless network, making the scale of internet larger and larger. The resources or data produced by these devices often contain privacy and sensitive data, so there will be serious consequences when they are obtained illegally.

Traditional access control methods include discretionary access control (DAC), identity-based access control (IBAC), and mandatory access control (MAC), etc. But these methods are all centralized designs, which have the disadvantages of single-point failure, difficult to expand, low reliability and low throughput. Attributed based access control (ABAC) is an access control model, which controls the access between subjects and objects, according to the attributes of entries, operations and related environments.

In this project, we provide a way to implement Attribute Based Access Control using Blockchain technology with special focus on IoT data. We provide mechanisms for different organizations to authorize different users to access different devices and provide a platform on which different organizations can securly exchange data among themselves.

blank page

Background

2.1 What Is Access Control

Access control is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.

Access control systems perform identification authentication and authorization of users and entities by evaluating required login credentials that can include passwords, personal identification numbers (PINs), biometric scans, security tokens or other authentication factors.

2.2 Types of Access Control

There are different types of Access Control models based on requirements. Some of the widely used models are:

- **Mandatory access control (MAC):** This is a security model in which access rights are regulated by a central authority based on multiple levels of security. Often used in government and military environments, classifications are assigned to system resources and the operating system (OS) or security kernel.

blank page

- **Discretionary access control (DAC):** This is an access control method in which owners or administrators of the protected system, data or resource set the policies defining who or what is authorized to access the resource. Many of these systems enable administrators to limit the propagation of access rights. A common criticism of DAC systems is a lack of centralized control.

- **Rule-based access control:** This is a security model in which the system administrator defines the rules that govern access to resource objects. Often, these rules are based on conditions, such as time of day or location. It is not uncommon to use some form of both rule-based access control and RBAC to enforce access policies and procedures.

- **Role-based access control (RBAC):** This is a widely used access control mechanism that restricts access to computer resources based on individuals or groups with defined business functions – e.g., executive level, engineer level 1, etc. – rather than the identities of individual users. The role-based security model relies on a complex structure of role assignments, role authorizations and role permissions developed using role engineering to regulate employee access to systems. RBAC systems can be used to enforce MAC and DAC frameworks.

- **Attribute-based access control (ABAC):** This is a methodology that manages access rights by evaluating a set of rules, policies and relationships using the attributes of users, systems and environmental conditions.

blank page

2.3 What is Blockchain

Blockchain, sometimes referred to as Distributed Ledger Technology (DLT), makes the history of any digital asset unalterable and transparent through the use of decentralization and cryptographic hashing. This allows data to be stored globally on thousands of servers while letting anyone on the network see everyone else's entries in near real-time. That makes it difficult for one user to gain control of, the entire network.
There are various types of blockchains that implement different types of consensus algorithms and different ways of storing data. For this project, we have used Hyperledger Fabric that is explained below.

2.4 Hyperledger Fabric

To decide on what kind of blockchain would suit our purpose, we evaluated through following models 1. Karl Wutsl Model 2. Gardner Model

After validating through above models, we concluded a private-permissioned blockchain would be suitable for an access-control system in a marketplace setting. The Hyperledger fabric project launched by the Linux Foundation in 2015 is an enterprise blockchain developing platform. To counter issues such as lower transaction throughput, consistency issues, privacy issues of public blockchains like Bitcoin and Ethereum, hyperledger fabric provides a solid scalable architechture. Unlike public blockchains, Hyperledger fabric's architecture has modular components including authentication, consensus algorithm, smart contracts, data storage.

All programs (smart contracts, data storage etc.) run in the docker containers. These containers are like sandbox, thus separating application with physical resources. Since, Hyperledger fabric is a permission-ed network of nodes, only authorized parties (users and nodes) can join the blockchain network. Hyperledger fabric overcomes a lot of shortcomings of public blockchain. Below is a description of main components of hyperledger fabric.

blank page

2.4.1 Certificate Authority

Certificate Authority provides and revokes digital certificates and private keys to member nodes and users. Generally 1 CA is setup per organization.

2.4.2 Client

Client interacts with the user and peer node and indirectly with blockchain via Smart contracts deployed on the peer nodes. The client provides following operations.

- **Network Manager**: To manage network nodes (start, stop, configure)
- **Chaincode manager**: Admin interface to deploy, remove or update chaincode on peer nodes.
- **User Interface**: The interface developed on Fabric SDK, provides all sorts of business features supported by smart contracts deployed on Peers

2.4.3 Peer

Peers store blockchain. Smart contracts are deployed on Peers. Clients connect with peer to interact with blockchain. The types of peers

- **Endorser Peers**: Hyperledger fabric operates on Execute-Order-Validate flow for a singe transaction. Thus any transaction incoming from some client goes to an endorser peer first. The role of endorser peer is responsible for verifying, simulating and endorsing transactions.
- **Committing Peers**: All peers are committing peers. They receive blocks of transactions from orderer, and are responsible for validating the transaction and updating blockchain and ledger status.
- **Anchor Peers**: A set of peers from multiple orgs responsible to distribute data coming from Orderers to their organization's peers.

blank page

2.4.4 Orderer

Orderer nodes receive the simulated transactions from endorsing peers. The job of the orderer nodes is to order the incoming transactions. To eliminate the inconsistencies across orderer nodes, a consensus has to be reached among all the orderer nodes. Orderer then creates block of transactions and then decimate to commiting peers.

2.4.5 Channel

Channel defines the network. An Organization can be part of multiple channels(networks). We define policies at channel level.

2.4.6 Private Data store

To store proprietary data away from fellow organizations on the channel while providing proof of ownership, organization can store their data on the private data store. The private data store then records a transaction on blockchain consisting of an hash of the corresponding data.

2.4.7 Ledger

The ledger is devided in 2 parts.

- **World** State: Mutable Key-value pair data store. The keys are current assets on blockchain and values indicate their current values.
- **Blokchain**: An immutable transaction log that records all changes that have resulted in the formation of the current world state.

There is one ledger per channel. Each peer in the channel maintains a copy of the ledger

2.4.8 Chaincode

Chaincode or smart contracts generates transactions to interact with the blockchain. All the business logic can be defined by one or more smart contracts in particular order.

blank page

Each chaincode is bound with an endorsement policy, which determines which organization in a blockchain network should sign the transaction generated by a given smart contract for it to be considered valid.

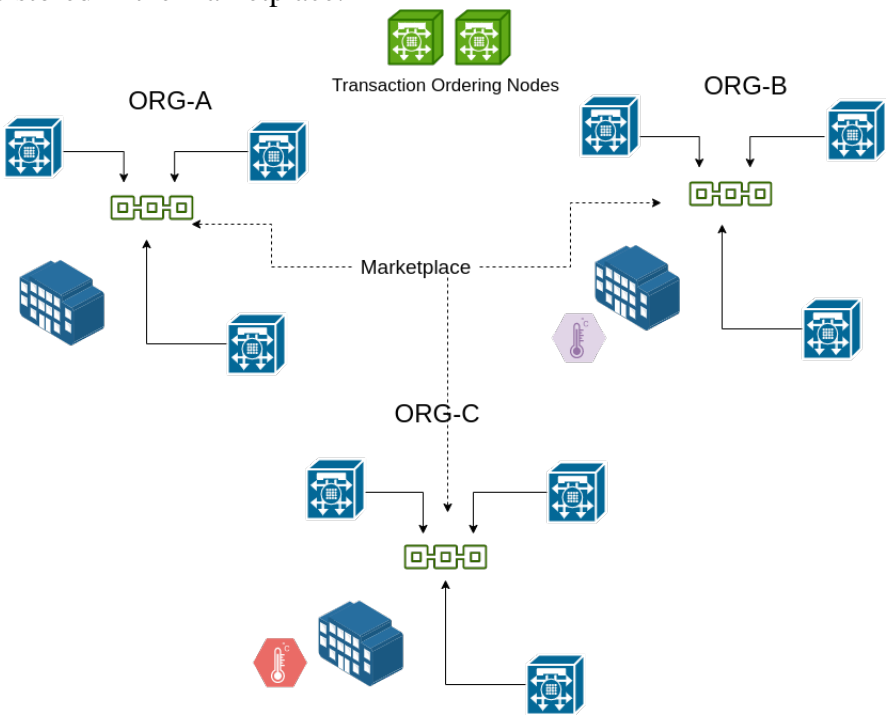
blank page

Proposed Approach

In this section we explain in detail and give a flow of how organizations can control access to their devices and share data with each other.

3.1 Decentralized Marketplace

The marketplace is the public section common to all organizations. Details about the devices are stored here. The interest tokens for devices (explained in below sections) are also stored in the marketplace.



- 1 blockchain per marketplace
- Sellers can put device data on sale

blank page

- Buyers can query Marketplace for on-sale assets
- For every Successful Trade, Receipt is generated and stored on Blockchain for auditing

3.2 Components of an organization

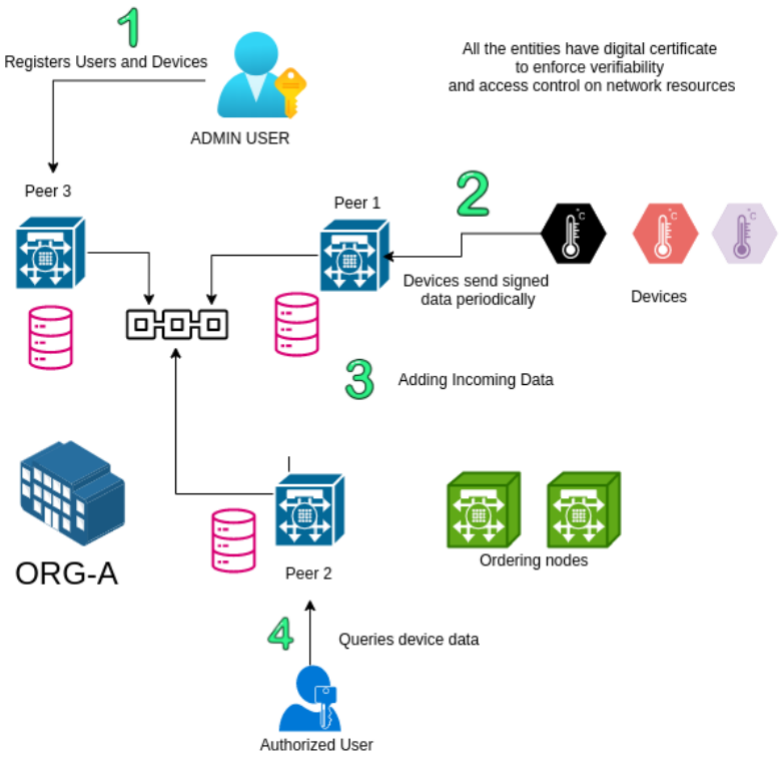
First we explain all the components that an organization will have and the places it can store data to.

- **Devices Private Data:** This is the place where private data for the devices (secret, keys etc) are stored. The data for devices is also stored here. This is a private section. The hash for this data is stored on the public ledger.
- **Trade Agreements Data:** This is the place where trade agreements with other organizations are stored. The receipts for trades are also stored here. This is a private section. The hash for this data is stored on the public ledger.
- **Shared Data section:** For every pair of organizations in the network, there exists a shared collection that is accessible by only those 2 organizations. Any data they might want to share with each other is stored here.

blank page

3.3 Intra Organization Flow

First we will take the case of a single organization and explain how an organization can store it's data and grant access to users within the organization.



Every organization has one or more admin users who are responsible for registering other non-privileged users and devices.

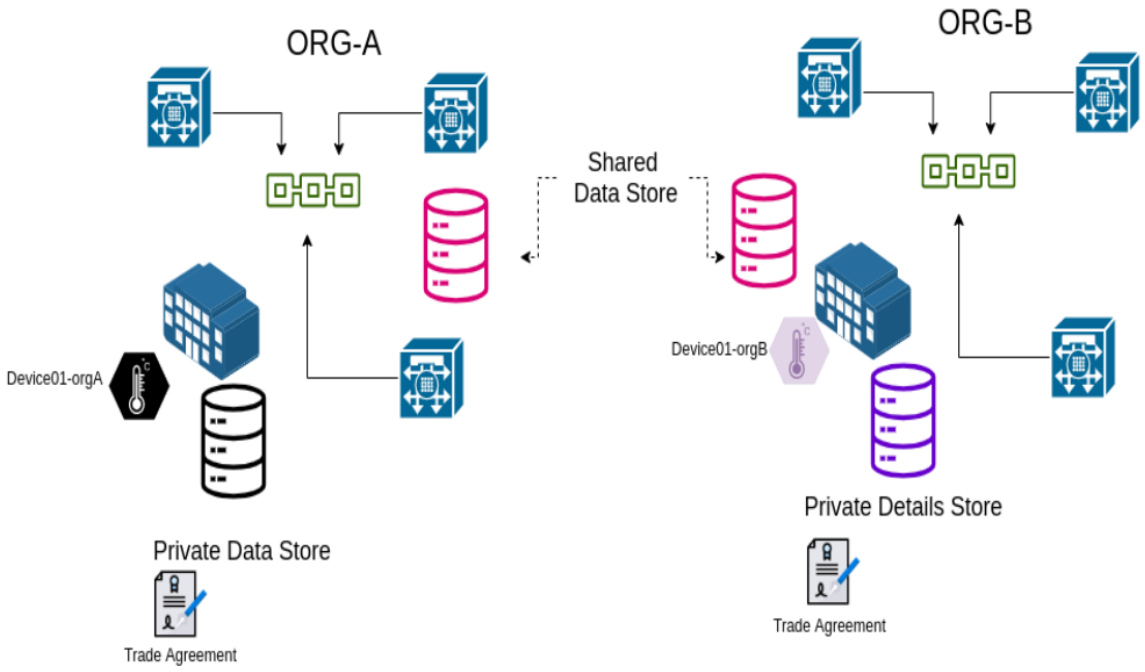
- Whenever a new device is registered, a new user is created and attached with the device. Crypto material (Certificates, keys, etc) is generated for this attached user. Unless changed by admin, only this user is permitted to add data for the device. The common details are added to the public part of block-chain ensuring everyone can see the device whereas secrets of the device are stored in private data part of the block-chain

blank page

- Every time a device adds data to the channel, it sends the certificate issued to the attached user. The data is then added to the private storage of the organization. The corresponding hash is recorded on the public ledger. The timestamp and transaction ID is recorded and an event containing the details is emitted.
- Now any authorized user can query the data of the device from the private store.
- Other organizations receive only the hash of the data. This ensures that if they get access to the data later, they can use this hash to verify that the data they received has not been tampered with.

3.4 Inter Organization Data Sharing

Now we will explain how the sharing of data between organizations will work



Whenever a new device is added, it's entry is put on the corresponding marketplace. This ensures that everyone is aware of the existence of the new device. Data sharing is a multi step process as given below

blank page

- The owner organization marks a device as ”on sale” meaning is is willing to negotiate a deal with someone for the device data
- The owner organization can add one or more trade agreements for the device having different parameters. These trade agreements are stored in the private details of the org. This is private so that others can not see the trade agreement an org makes with someone else. This is not a mandatory step. A trade agreement can be initiated by either the owner or the buyer.
- The buyer creates a trade agreement in it's own private data for the device containing all the parameters. If the buyer knows about a trade agreement on the owners end that it agree with, it can use ID and parameters of the same agreement or it can create a new trade agreement and the owner can choose to accept or reject it. The information about trade agreements created by the owner will need to be communicated to the buyer of the chain (as they are in private store). This ensures that buyer knows about an existing trade agreement only if the owner chose to tell it.
- The buyer creates an interest token for the device. This interest token is public and everyone can see it. A trade ID needs to be attached to the interest token that can correspond to a trade either at buyers end or at seller's end.
- The owner sees the Interest Token. If the corresponding trade was created by the owner, the owner can directly accept the interest token once the exchange for the price has taken place. If the trade was initiated by the buyer, the owner will need to create a corresponding trade agreement with same parameters before accepting the token. Note that the exchange for the price of the device has to take place off the chain.
- Once the buyer accepts the interest token, the hash of both trade agreements (owner and buyer) are matched to ensure that both have set the same parameters for the trade. Once verified, a trade receipt is generated containing the transaction ID of the trade and other details and an event containing this receipt is emitted to the owner and the buyer.

blank page

Implementation Details

In this section we explain how we implemented each part that we proposed above.

4.1 Code Repo Links

The entire **chain-code** for the block-chain is written in GoLang. Code can be found **here**

The code for the **Application** is written in node.js . Code can be found **here** .

The code for the **Web User Interface** is written in React. Code can be found **here** .

4.2 Collections

As explained in the last chapter, each org has multiple collections it can write data to. Some are public and some are private. Each organization has one public collection, the marketplace. Then we created 4 private collections for each organization. 1. Device Private Details, 2. Device Data, 3. Trade Agreements, 4. Access Control List.

Along with this, for each pair of organization, there exists a collection where they keep the data hat they want to share among themselves.

The number of collections running on any single node at any time are n+4, where n is the number of organizations in the network.

blank page

4.3 Access Control List

To streamline the process of adding device data to shared collections, we created access control lists. Each organization maintains an access control list having details of which device data is to be shared with which organization.

Whenever new data comes for the device, the ACL is checked and data is also added to the corresponding shared collections.

Whenever a new trade trade completes successfully, an entry is added to the ACL automatically that grants access to the buyer organization.

4.4 Consensus Algorithm

Many distributed blockchains, such as Ethereum and Bitcoin, are not permissioned, which means that any node can participate in the consensus process, wherein transactions are ordered and bundled into blocks. Because of this fact, these systems rely on probabilistic consensus algorithms which eventually guarantee ledger consistency to a high degree of probability, but which are still vulnerable to divergent ledgers (also known as a ledger "fork"), where different participants in the network have a different view of the accepted order of transactions.

Hyperledger Fabric works differently. It features a node called an orderer that does this transaction ordering, which along with other orderer nodes forms an ordering service. Because Fabric's design relies on deterministic consensus algorithms, any block validated by the peer is guaranteed to be final and correct. Ledgers cannot fork the way they do in many other distributed and permissionless blockchain networks.

Thus consensus needs to be achieved among all peers on what is the order of transactions and the ordering node must approve this order. For this, fabric provides a plugglable interface in which we can use any consensus algorithm we like. By default Hyperledger supports **Raft** and **Kafka** consensus.

In this project we have used Raft Consensus Algorithm.

blank page

4.4.1 RAFT Consensus

Raft is a crash fault tolerant (CFT) ordering service based on an implementation of Raft protocol. Raft follows a "leader and follower" model, where a leader node is elected (per channel) and its decisions are replicated by the followers. Raft ordering services should be easier to set up and manage than Kafka-based ordering services, and their design allows different organizations to contribute nodes to a distributed ordering service.
Raft is robust and can sustain loss of nodes including leader node, as long as majority of nodes are alive
Leader Election in Raft works as follows:

- Every ordering node is in one of 3 states Leader, follower, candidate. Initially as follower
- If a leader is already elected, follower receive logs from leader and replicates deterministically
- If no heartbeat message is received, this means no leader has been elected so far
- Nodes self-promote to candidate state and asks for votes from peers

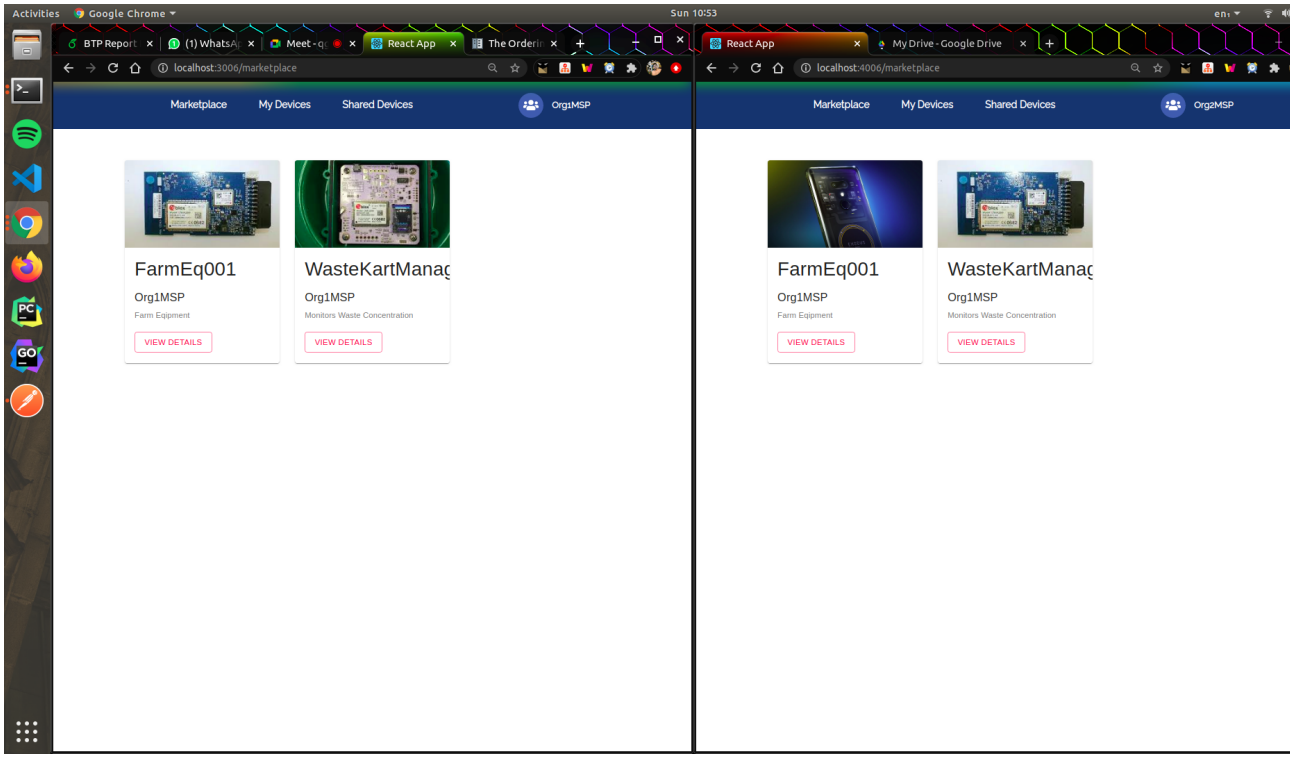
4.5 Data Sync Across Nodes

Hyperledger uses gossip-based data dissemination protocol to communicate among nodes and sync data. It perfoms 3 primary functions:

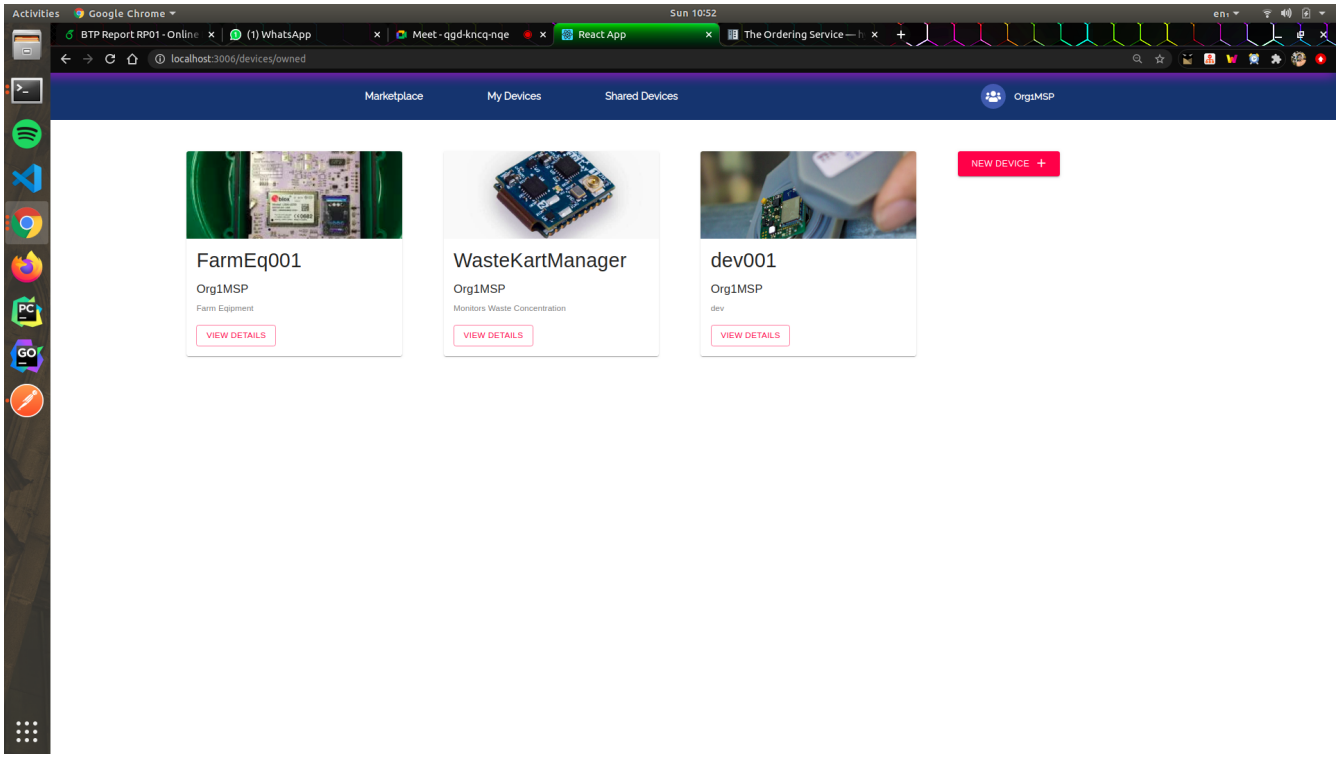
- Manages peer discovery and channel membership, by continually identifying available member peers, and eventually detecting peers that have gone offline.
- Disseminates ledger data across all peers on a channel. Any peer with data that is out of sync with the rest of the channel identifies the missing blocks and syncs itself by copying the correct data.
- Bring newly connected peers up to speed by allowing peer-to-peer state transfer update of ledger data.

Project Screenshots

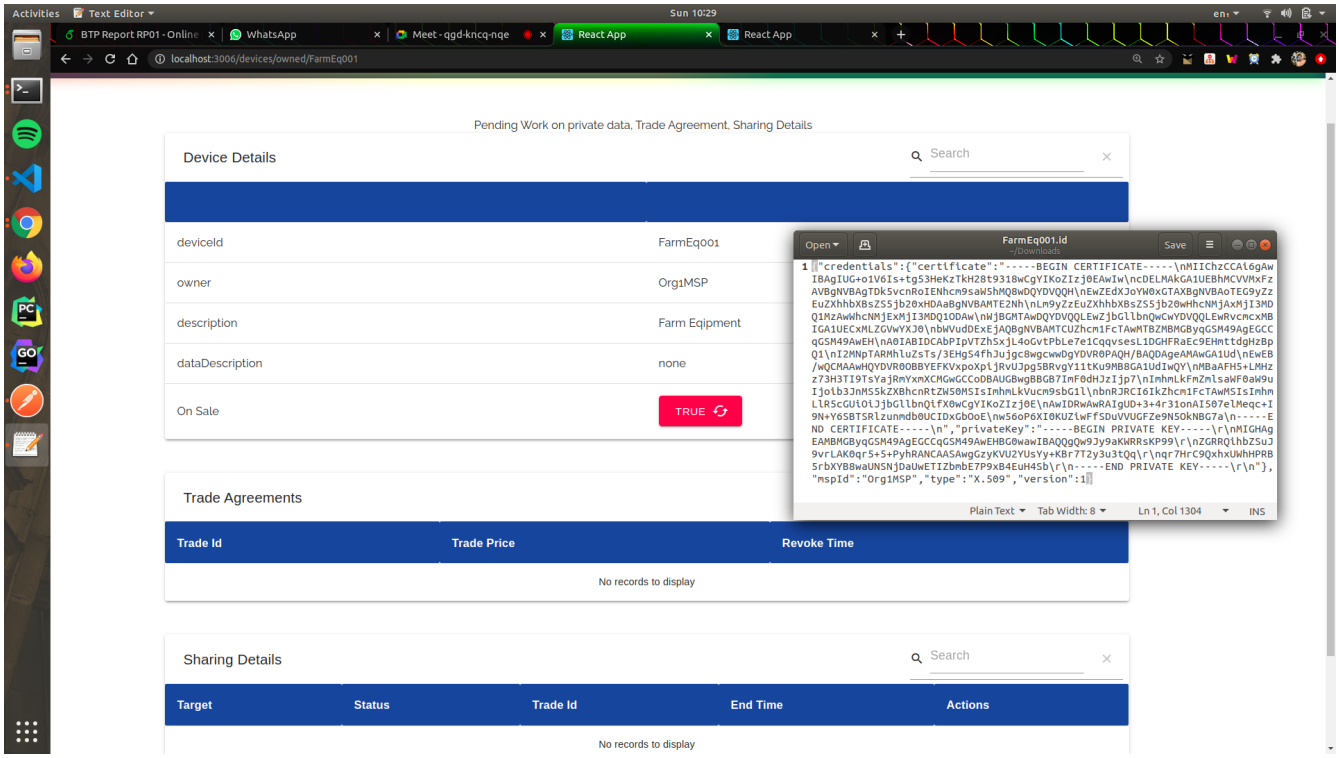
Here are some screenshots from our project.



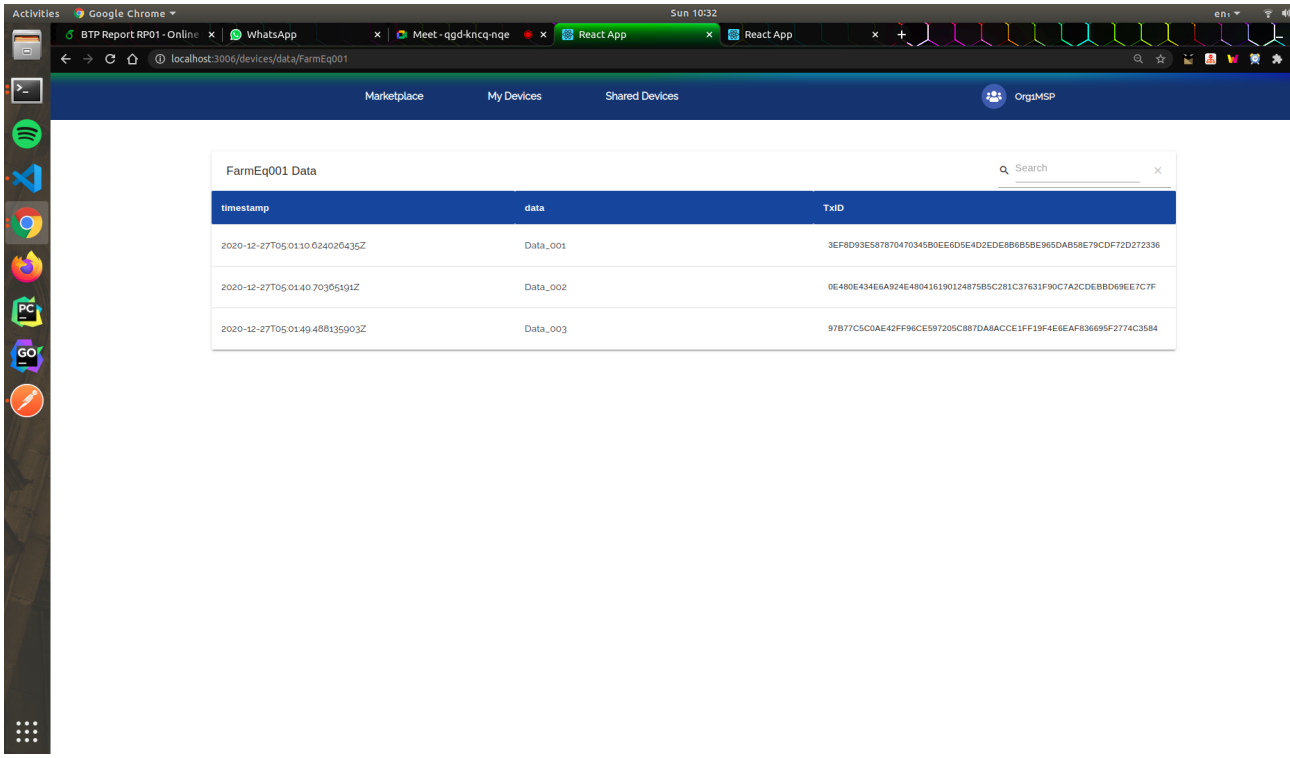
The Marketplace displaying all devices on sale.



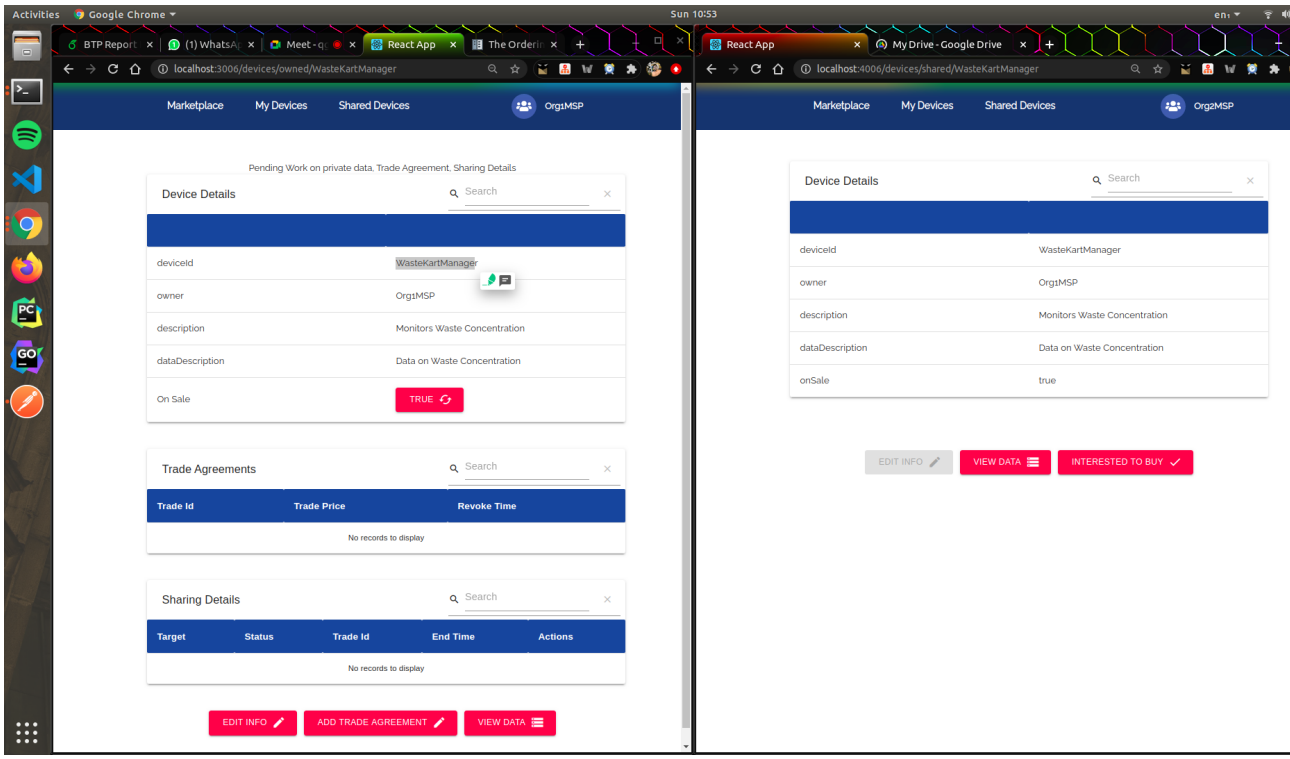
Seller Org1's all devices, two of them are on sale.



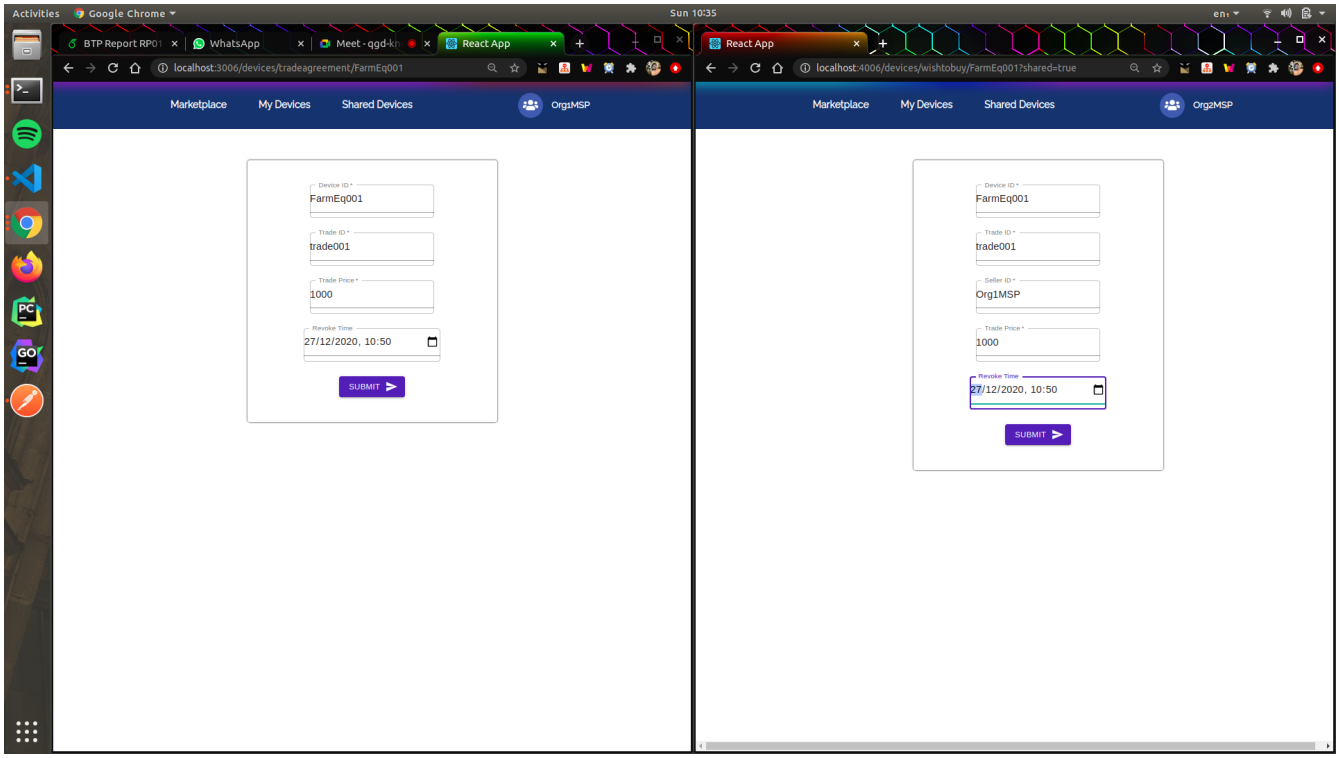
Seller Org1 creates new device. The device X.509 certificate is downloaded. Every subsequent data request send by device, will be signed with this certificate. Org1's peer node will validate and authenticate.



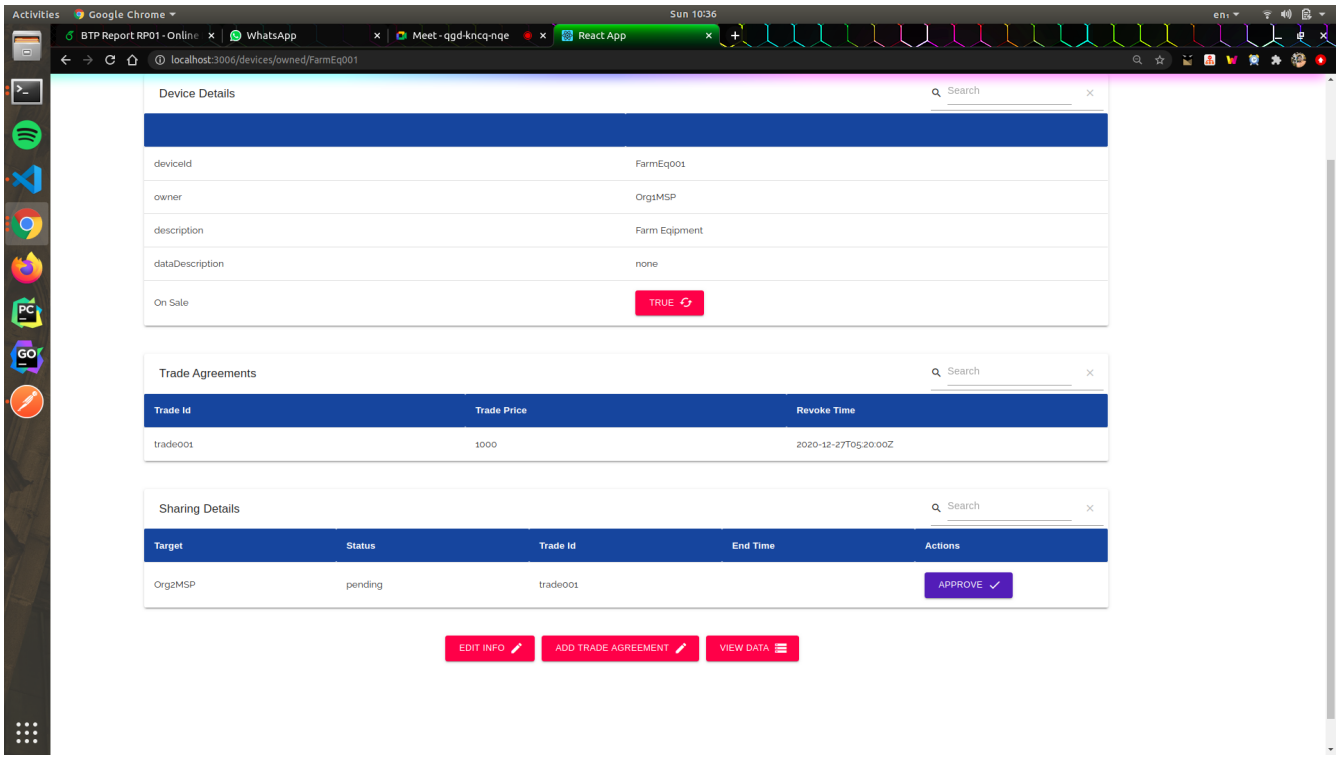
Incoming data from the device, visible to the owner



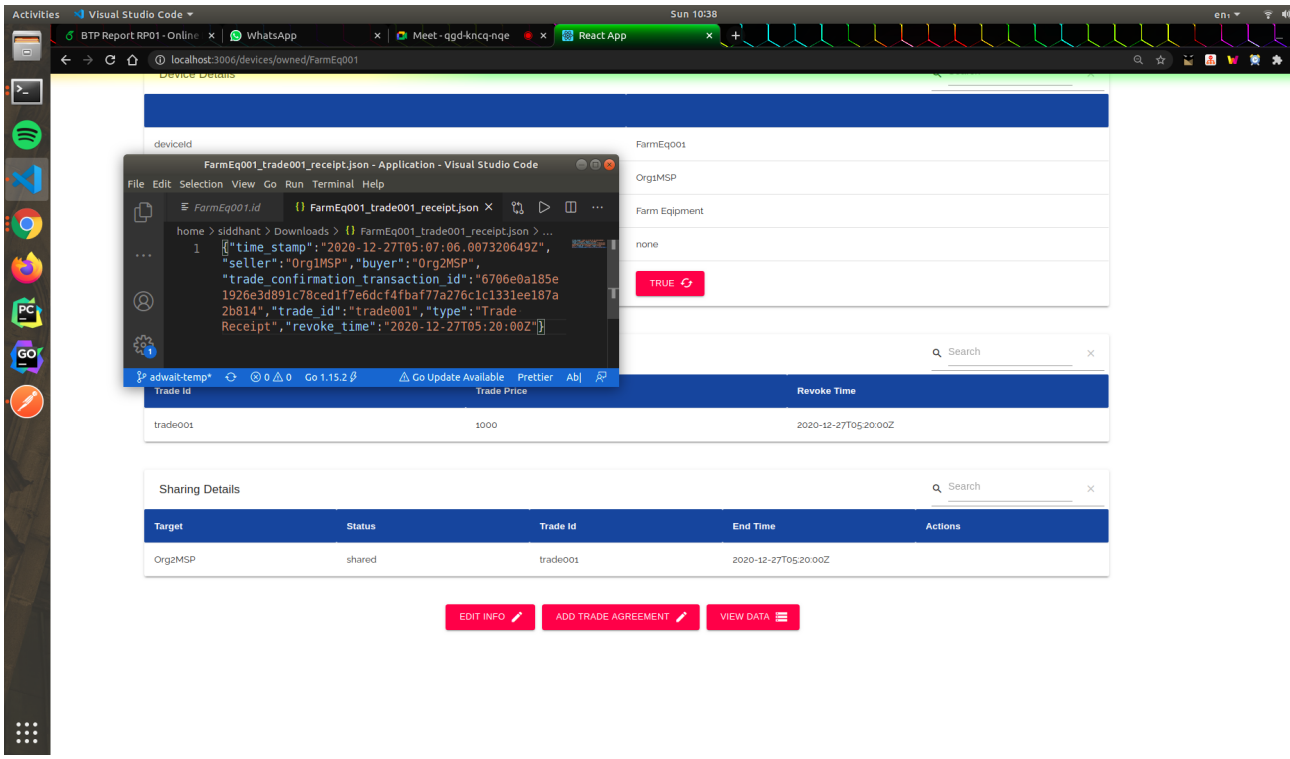
Different device details page for Seller(left) and Buyer(Right) in a Marketplace setting



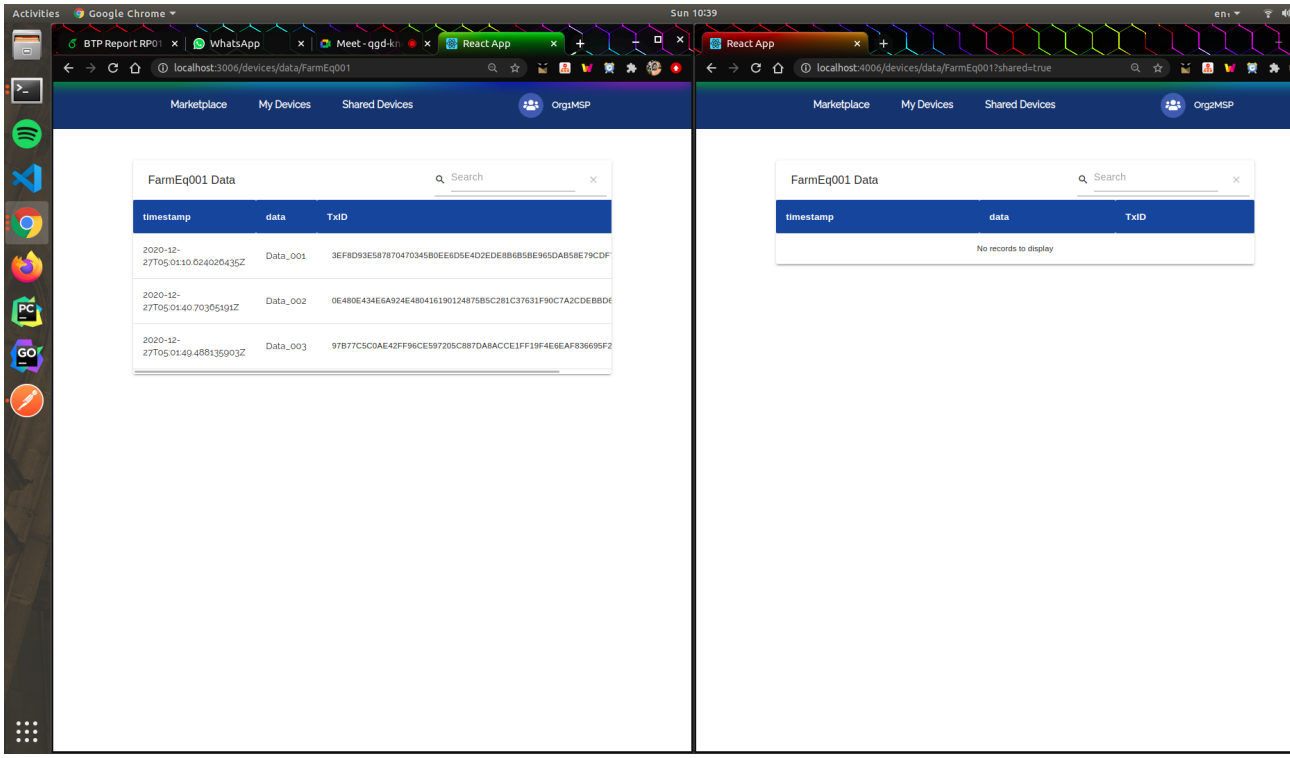
Trade Agreements generated by Seller (Org1) and Buyer(Org2).



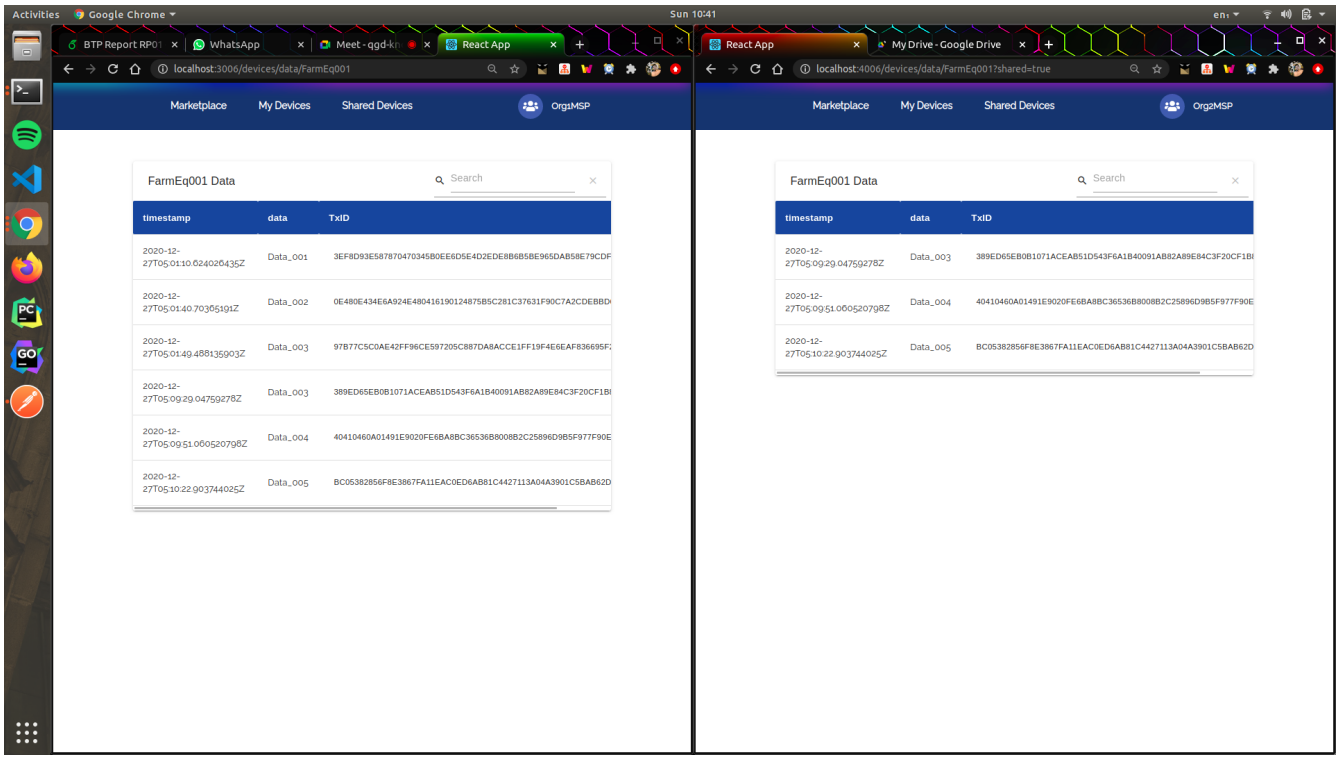
Seller can view pending and completed trade requests. If pending, seller can approve.



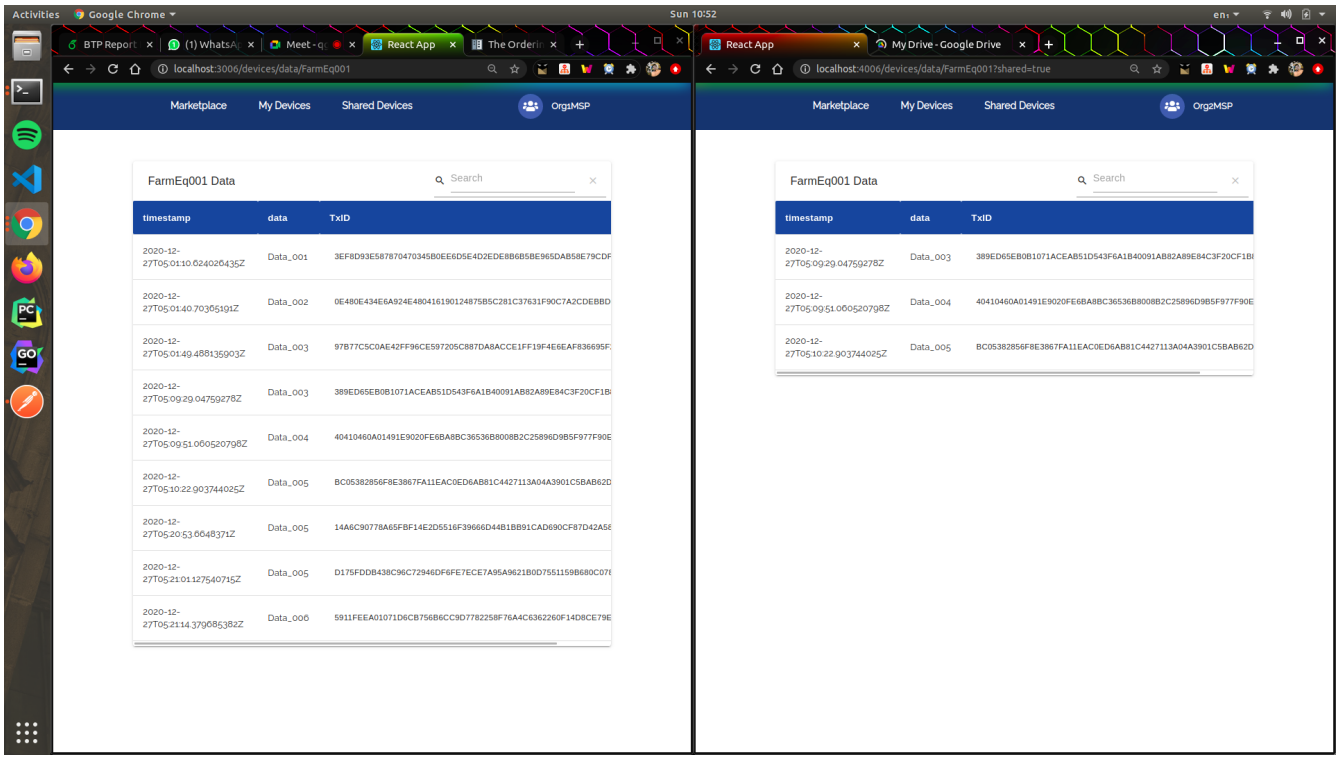
Trade Success. The Trade Agreement hashes must match. If trade is successful, receipt is generated and can be used to audit blockchain Transaction later.



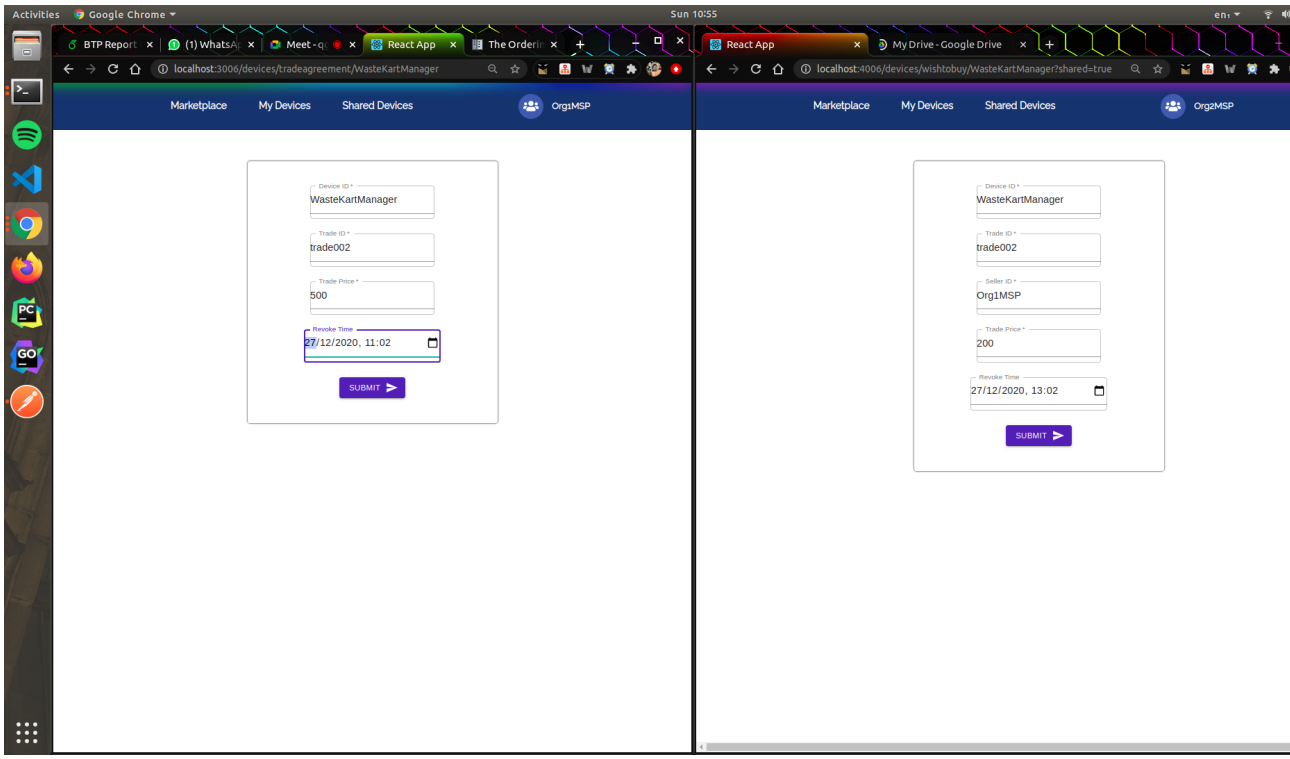
Data with Seller (left, Org1) and Buyer (right, Org2) before starting starting data share.



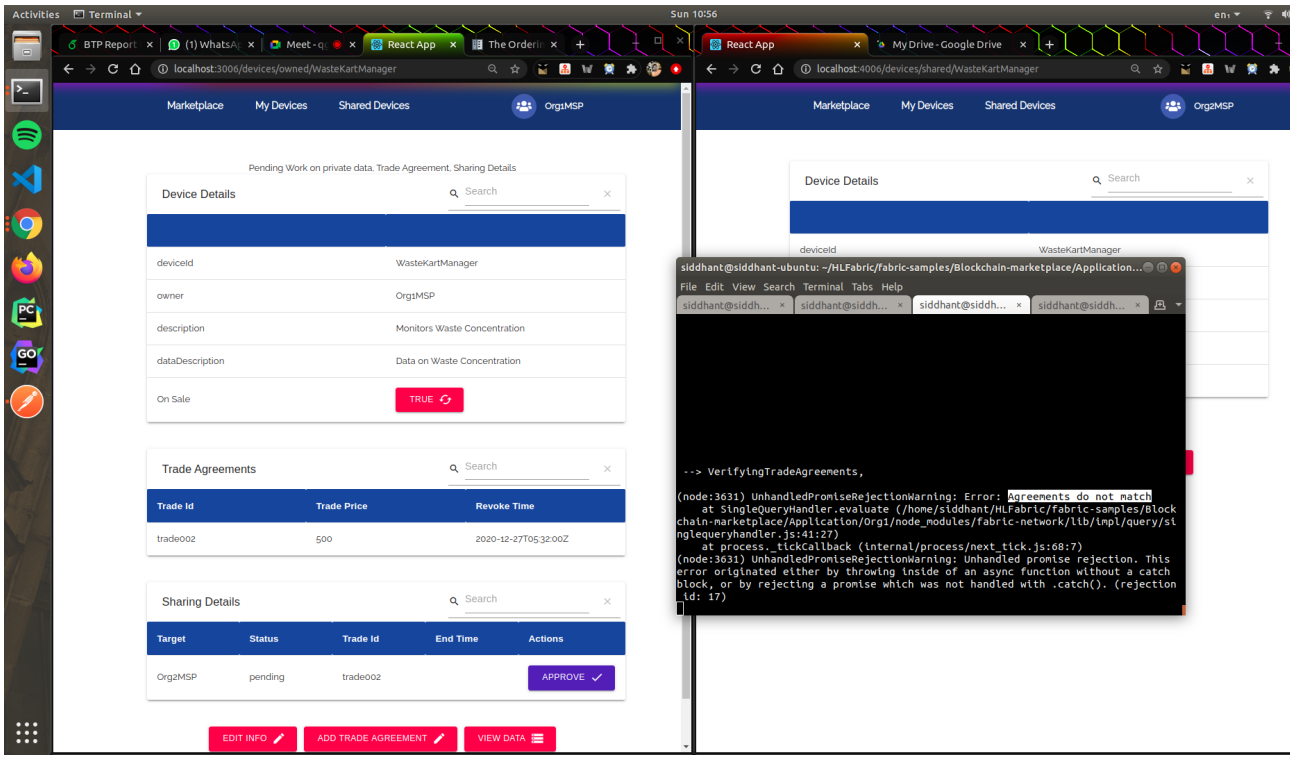
Data with Seller (left, Org1) and Buyer (right, Org2) while sharing data.



Data with Seller (left, Org1) and Buyer (right, Org2) after time limit expires. Only Seller gets new data.



A case where Trade agreements differ. The prices set by seller and buyer are different.



While approving Interest Tokens, if trade agreements do not match trade fails. Both the parties must agree on Trade Details.

Conclusion

In this project we were able to successfully present a solution for secure and fast access control using blockchain. This project proves the feasibility of the idea.

The performance of the project depends on various factors such as the transaction size, block size, network size, hardware limits etc.

The **Hyperledger Fabric Performance and Scale working group** currently works on a benchmarking framework called **Hyperledger Caliper**.

Several research papers have been published studying and testing the performance capabilities of Hyperledger Fabric. The latest **scaled Fabric to 20,000 transactions per second**

This project can be modified in a lot of ways to suit different business needs. We mention some of them below

blank page

Future Work

- More types of complex trade agreements, data types can be added that can be used in different business scenarios.
- More consensus algorithms can be explored that may give better performance.
- A richer access control system can be created that gives access only to some part of device data or past data.
- Automated scripts can be made that ease in deployment of the project.

blank page

References

1. D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable Access Control systems," Computers Security, vol. 84, pp. 93–119, Jul. 2019, doi: **https://doi.org/10.1016/j.cose.2019.03.016**

2. Do you need a Blockchain? **https://eprint.iacr.org/2017/375.pdf**

3. Raft Protocol **https://raft.github.io/raft.pdf**

4. C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," Journal of Network and Computer Applications, vol. 116, pp. 42–52, Aug. 2018, doi: **https://doi.org/10.1016/j.jnca.2018.05.005**

5. T. Sultana, A. Ghaffar, M. Azeem, Z. Abubaker, M. U. Gurmani, and N. Javaid, "Data Sharing System Integrating Access Control Based on Smart Contracts for IoT," in Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Springer International Publishing, 2019, pp. 863–874. **Link**

6. Hyperledger Fabric White paper **Link**