Bilkent University

Department of Computer Engineering

Senior Design Project

Project Short-name: So FarM So Good

High Level Design Report

Giray Baha Kezer, Fazilet Simge Er, Melih Ünsal, Kaan Atakan Öztürk

Supervisor: Prof. Dr. Halil Altay Güvenir

Jury Members: Prof. Dr. Özcan Öztürk, Prof. Dr. Uğur Güdükbay

Innovation Expert: Kerem Erikçi

December 31, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491.

TABLE OF CONTENTS

1 Introduction

1.1 Purpose of the System

In Turkey, farmers, especially the ones without large businesses, have many problems and this situation causes our production in agriculture to decrease day by day. However, we cannot put ourselves at risk for agriculture because agriculture constitutes around 13% of the country's exports. We also see that agricultural areas and the number of farmers has decreased for the last 5 years. So the problems of agriculture should be resolved one by one and we plan to solve some problems of agriculture by So FarM So Good. This web based platform is going to combine small farmers so that they will be able to sell their goods to the appropriate companies by the virtual cooperative we are going to establish. With our virtual cooperative, the farmers will be able to combine their crops and the cooperative is going to sell the crops and pay to the farmers proportional to the amount of products the farmers put into with more secure payment system which uses the new trend: Blockchain

1.2 Design Goals

1.2.1 Usability

- The target customer of the platform is farmers, which are the most important part of agriculture. Hence, the platform should have ease of use for them. Farmers demand more ease of use for this technological move. Thus, the features of the platform and the user interface should be easy to understand and use.

1.2.2 Security

- The platform should ensure the security of data of users and private information about companies by blockchain system, taking KVKK into consideration.

1.2.3 Scalability

- The platform should be scalable enough to handle the huge number of users and data processing work.

1.2.4 Robustness

- The platform should be robust. Whatever the size of the coming data, the platform should handle it.

- The platform should be able to handle internet connection problems and errors. These problems should not affect the transactions and put any of the users into trouble.

1.2.5 Extensibility

- The platform should support easy integrations for possible future features.

1.3 Definitions, Acronyms, and Abbreviations

- **KVKK:** Kişisel Verileri Koruma Kanunu (Law on the protection of personal data in Turkey)
- **P2P:** Peer to Peer

1.4 Overview

Our platform consists of different subsystems and it will be a web application written in Angular 4-5. Blockchain will be our platform's core.The platform will be developed using Docker to standardize operations and seamlessly move the platform. A database will also be set up to store data. Architectural overview of the proposed system are examined in Section 3.1 .

2 Current Software Architecture

There are several platforms which provide similar services to the customers. The following two are the most similar platforms which has some features we are planning our platform to have.

2.1 Agrivi

Agrivi is a data-driven farm management software which helps people manage every activity on their farm. It gives the people the opportunity to plan, monitor and analyze their farms from planting to harvesting with only a few clicks. Agrivi's core features are farm management, powerful analytics, advanced pest detection algorithms and best practice knowledge [1].

2.2 WeFarm4

Wefarm4 is a cooperation among farmers with the sub-branches of WeFarm4future, WeFarm4planet, WeFarm4Community and WeFarm4Growth. They're aiming to work together, create long-term benefits and empower farmers through cooperatives [2].

Above mentioned applications are established to support to the farmers. However, our projects different than the current ones. Establishing the online cooperatives between farmers is a significant feature that does not exist those. Moreover, we also intend to solve quota problem for the sugar beet farmers which is also new for the current system. On the other hand, our software will be web application. However, entire above mentioned softwares have mobile applications. The project also aims to contribute economy of the Turkey by controlling sugar beet distribution and holding the data in Blockchain system.

3 Proposed Software Architecture

3.1 Overview

SoFarmSoGood will use Peer to Peer Architecture instead of Microservices or Layered architecture. Among the most important is the fact that P2P networks offer greater security than traditional client-server arrangements. The distribution of blockchains over large numbers of nodes renders them virtually immune to the Denial-of-Service (DoS) attacks that plague numerous systems. Likewise, because a majority of nodes must establish consensus before data is added to a blockchain, it's almost impossible for an attacker to alter the data. This is especially true for big networks like the one of Bitcoin. Beyond security, the use of P2P architecture in cryptocurrency blockchains also renders them resistant to censorship by central authorities. Unlike standard bank accounts, cryptocurrency wallets cannot be frozen or drained by governments. This resistance also extends to censorship efforts by private payment processing and content platforms.[3]

3.2 Subsystem Decomposition

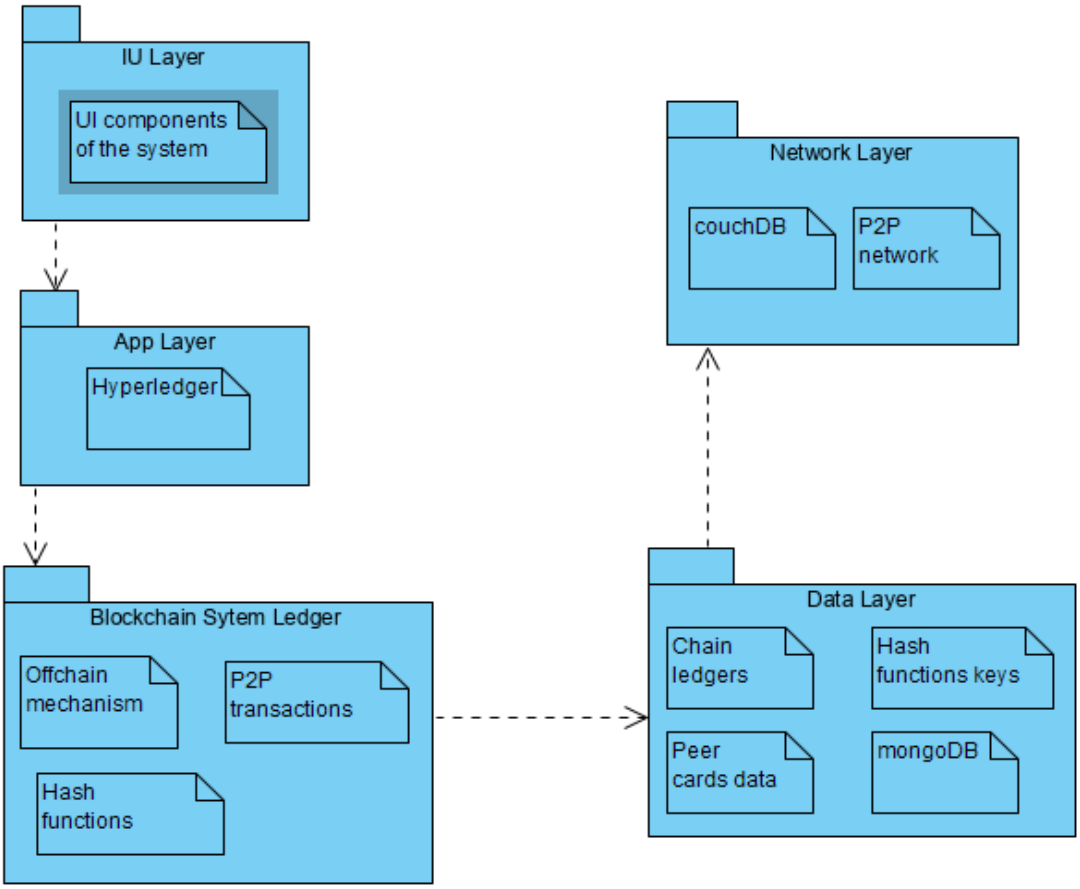

Figure 1:Subsystem Decomposition

3.3 Hardware/Software Mapping

- System performance should be both fast and reliable enough to hold transactions securely.
- Web based website of SoFarmSoGood will run on any browser.
- By the nature of blockchain technologies, all the data will be hold on every computer connected to the network.

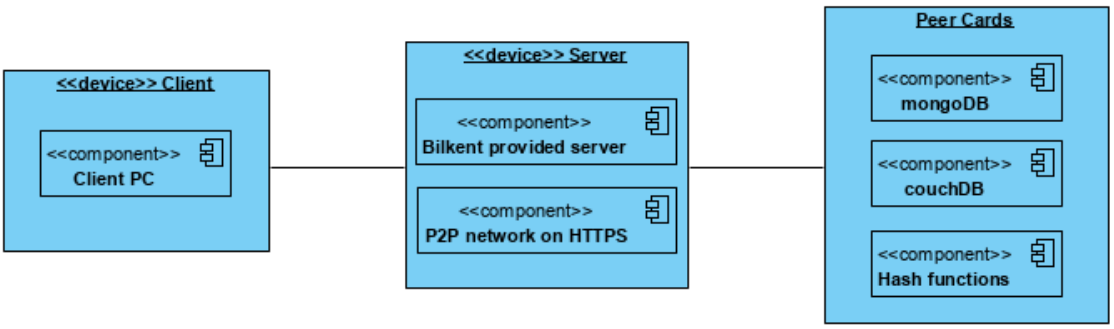

Figure 2: Client-Server Node Diagram for Hardware Software Mapping

3.4 Persistent Data Management

SoFarmSoGood is a decentralized blockchain platform that makes farmers together related their product amount. Platform uses their harvest rates mostly in the case of being above to the quota limits of product companies. In the sugar beet harvest we have an example of Konya Şeker Company as the biggest one in that scope. Platform makes virtual cooperatives among them and makes them as a one producer to the market. We will support Blockchain architecture and Blockchain composer as Hyperledger Composer.

Blockchain is based on P2P network and distributed network technology is used to validate transactions. In order to store and manage data properly, Blockchain uses **Distributed Ledger Technology (DLT).** This technology serves as a decentralized database between multiple nodes. Transactions are stored in the ledger as a series of blocks where each block pointing to the preceding one. A Blockchain based storage system stores data by creating data segments and constructing unique hash functions for each data shard. Data management will be easier by using Blockchain because this technology offers more transparency than a traditional cloud service. Moreover, it also provides performance advantages as users can see the data where it is stored. Blockchain is extremely secure because of hash functions which generates 64 bit private keys. Those hash functions are almost impossible to be hacked. Then it generally changes this key with some different periods. Namely, possibility of facing malwares is very low. In this structure the data is spread out. DLT also generates redundant copies in order to prevent data loss from all sort of events.

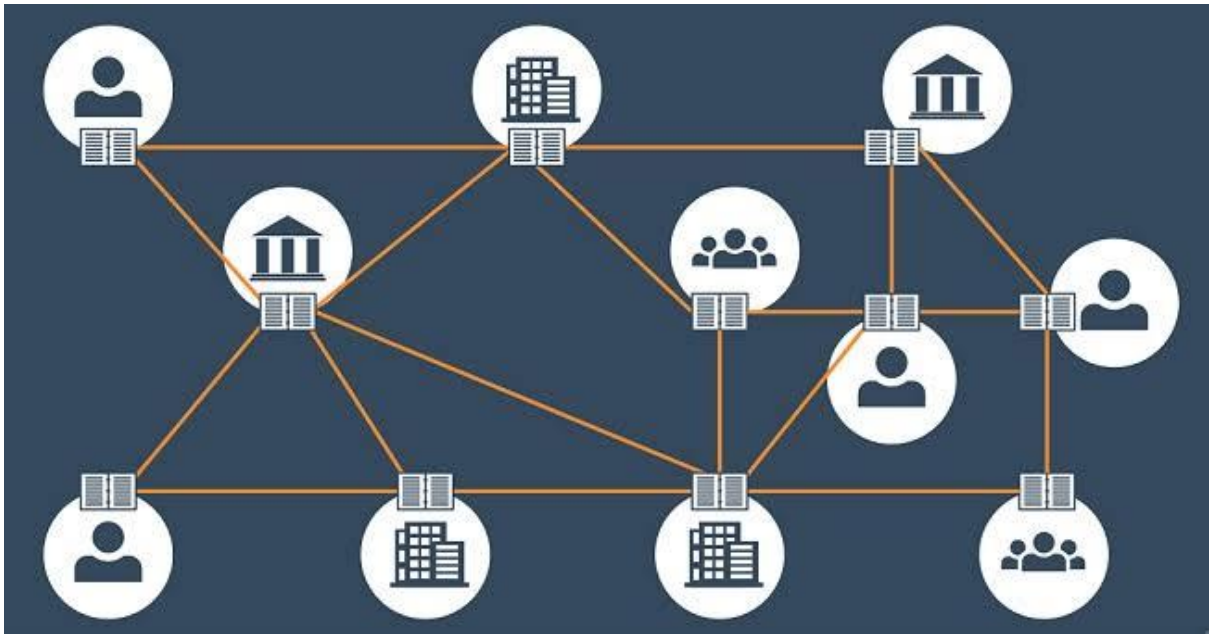

Figure 3: LT Network

3.5 Access Control and Security

Blockchain technology has verified to be in corruptible and mostly transparent. This technology provides users end to end security which **prevents** third parties from accessing the data. Moreover, it provides encryption because of decentralized structure. Thanks to the distributed structure, Blockchain eliminates the possibility of human based errors and protects against siber attacks. In addition to the various advantages of **regular** access control technology might be considered the entire data is stored at a server. Because of that storage, the all data is centralized and processed at a server. This situation basically means users might gain unauthorized access to the data and control operations. When your company takes security assistance from third party companies as outsourcing, you have to put lots of trust to them.

However, implementation of **Blockchain access control**, the above-mentioned threads will disappear. The entire information is distributed within a network of nodes and it is not stored exactly one server. Users of the Blockchain should be able to choose which personal info or data to share in their network.

These days many startups utilize Blockchain access control to empower their security requirements. Their main aim is resisting to the security threads by gaining control of the data by each end user. In brief, distributed technology like the blockchain access control will bring new level of cyber security. Private key permissions and encryption methods are essential for the decentralized technology and provides safer business environment.

3.6 Global Software Control

Since Blockchain decentralized architecture is applied, P2P network is the core of the all software control. This network consists of many computers, but in a way that the data can not be altered without the consensus of whole network. Since, decentralized network is dependent

on the P2P that is also known as crypto cycles and their laying out and propagation as the base foundational layer.

The structure of the Blockchain technology can be represented by a list of blocks with transactions in a particular order. These lists can be stored as txt file or in a form of simple database. This data structure includes two vital data structure: pointers and linked lists.
Pointers: They are variables that keep information about the location of another variable. Moreover, linked lists a sequence of blocks where each block has specific data and links to the successor block with using a pointer.

3.7 Boundary Conditions

3.7.1 Initialization

First of all, the user must sign up to the platform either as a company or a producer. They must provide their email address and a unique password in the course of the sign up process. After signing up, the user can login to the application from the login page, using the email and the password they've provided during the sign-up process.

3.7.2 Termination

The logged in user can log out by clicking their icon on the right top of the page, and then clicking *Logout* button. The user icon will be available to the user in every page, except the pages which are a part of the transaction process.

3.7.3 Failure

There are two different situations that might cause a system failure. These can be labeled as internet disconnection and database failure.
The application will be a web application; therefore, it will require internet connection throughout the usage of the application. Thus, disconnection from internet at any time will cause a failure in the system.
The data which will be shown to the user and used in the transactions will be taken from the database. Hence, a problem with the database or a problem with connecting to the database will cause a failure as well.

4 Subsystem Services

4.1 Application Logic

Application logic is sustained on blockchain process. It continues among hyperledger fabric and its composer, transactions goes on distributed ledgers and stored in peer cards of actors.

4.2 Data Management

Transaction data is stored transaction records in blockchain process. CouchDB is used for docker container; transaction data and network via blockchain and docker container is stored in couchDB. CouchDB is also used for communication via computers and also get and post methods inside. MongoDB mostly used for storing data of peer cards related their assets and system front end side. In our case it would be Angular JS related with frontend.

4.3 Blockchain Communication

Blockchain communicates via P2P distributed network among peer cards. SoFarmSoGood uses basic REST operations over HTTPs.

4.4 Blockchain Features

Ledgers, peer cards for actors of system to hold their assets and keys inside are holded inside our blockchain system. System will be formed with distributed ledgers as nodes, we will have genesis block as emergency hammer for ledger chains, data and relations. Nodes will be in communication as into P2P network via our system. Transactions will be recorded into secure way via blockchain natural features.

4.5 Hyperledger

For the SoFarmSoGood system Hyperledger Fabric Framework will be used. This framework could be used for decentralized and distributed mechanism blockchain systems. It also supports structures in block, private and public ones.

4.6 Off-Chain Computing

The transactions of harvest among farmers and companies holded by peer cards will be sustained with off-chain computing instead of on-chain computing. On-chain in some cases could limit the capacity of whole blockchain process that is why off-chain would be more appropriate for our process. Also process could be sustained in privately not in publicly, so off-chain is the best option for computing for our blockchain system.

4.7 P2P Transactions

P2P servers transactions will be used for transferring data into transactions between different actors, holded by peer cards in system. The process will be maintained via internet usage, P2P farmers and company transactions is one of the most important features of our blockchain

process. Transactions will be done as peer to peer transactions and blockchain through its nature, will secure the transaction data in chain.

4.8 Blockchain Structure

The chain process will be sustained via connected/chained ledgers. Chain itself sustains the data among the actors in system. The chain will be secured thanks to hash functions of blockchain.

4.9 Hash Functions

Hashes are core features of blockchain technology. These are used for data storage inside chains and process works like that; input is taken by hash functions and generate some output data inside and secure inside. Then after some time hash functions generate new ones as private keys and holds inside and makes same chains with same ledgers via new keys. Thus, hash functions provides almost impossible to hacked system to users and developers, it means they stores related data inside in total secure way.

5 New Knowledge Acquired and Learning Strategies Used

Blockchain is one of the new growing technology in practice of data. There are new growing research market for that technology so even we have smaller scope of blockchain project, we also need to learn plenty of things. Some of these plenty of things can be exemplified such as being decentralized blockchain in ledgers, what are these ledgers, how they can be bind them and what is their relation among them? Also we researched about genesis block as first node of our blockchain relation. We also learned about composers and we picked Hyperledger Fabric and we learned about ledger connections as off chain and on chain. Then we researched about blockchain elements' affairs and types as commodity, asset and actors. We searched about our composer playground thus, transaction affairs among actors. Then lastly, we additionally searched about our project's related front end side and we picked as Angular language to bind it with our blockchain. We need storage for our peer card, transaction and properties data inside blockchain relation, so we learned about mongoDB and couchDB and their assigned works by chain system. We added Docker container into our research side for connection again. Lastly we learned about RestAPI for relation. In detailed information can be found below.

Decentralized and distributed blockchain models are blockchain edited systems for some different usage, decentralized model is a model for coming from the center to leaves but there is no similar hierarchy inside distributed model. Distributed model is likely non-hierarchical model so that there is no need for center, governor for process. It is kind of having your data again in your hand mentality not like decentralized model. That is why, we picked distributed modelling after our related research for our blockchain ledgers process in terms of data relation. For that we need to learn about ledgers beforehand, we did some research about ledgers either. Ledgers can be seen as an account pages shows our accounts balance, information inside. We inspect relation of that ledgers connection among them in distributed modelling. These ledgers can be controlled and bind them into a relation they are binded without root in distributed model. In addition, genesis block is a first block of blockchain, which is the first block in any blockchain-based protocol. Foundation on which additional blocks are sequentially added to form a chain of blocks, resulting in the term, blockchain being controlled.[4] The second block could be iterated to the top of genesis block and chain is distributed in that way with binding on the top of consecutive blocks beginning from genesis. We searched about composers and picked Hyperledger Fabric via this virtual environment, we could model our program and work that task after that process, this task could be integrated with RestAPI to work into environment. Also we learned chain design mechanism; on-chain and off-chain. We would prefer off-chain mechanism related with our research about it, which does not require transactions inside the chain process it is progressed outside the chain inside db feature. We did related research about mongoDB and off-chain mechanism process for peer cards' transactions in record. MongoDB is also one the new learned things for our blockchain process, which offers us strong querying capability and cheap storage for large amount of data as other traditional databases. However, if you create your usage into distributed process for mongoDB you would take also cheap comparison pros inside blockchain data and easy redundancy ability for large amount data in process. [5] These were also other things that we learned from related storing research. Additionally, we searched about Docker container as a toolkit for our project reports. Docker would give us reduction of time about testing or deploying on error-prone steps automation and also quicker application testing via users if we could appropriately use it. We did some related research about Docker either and learned that information. The last thing, we could write here as an blockchain related element, is RestAPI. Lastly RestAPI is communication connection between client and server for data process of our blockchain, which was the result of our research about it. Then we searched about our project front end features except for back end side and we have picked Angular framework of JS language. We inspected its integration with blockchain, its pros and cons for conducting a project with it. To summarize even in a short period of researching, it was totally beneficial for us.

6 Glossary

Decentralized: The concept of a shared network of dispersed computers (or nodes) that can process transactions without a centrally located, third-party intermediary. [6]

Hyperledger: Started by the Linux Foundation, Hyperledger is an umbrella project of open source blockchains [6]

Hyperledger Composer: Hyperledger Composer is Blockchain Application Development framework which simplify the blockchain application development on Hyperledger Fabric[6]

Hyperledger Fabric: Hyperledger project hosted by Linux which hosts smart contracts called chaincode.[6]

Genesis Block
The initial block within a blockchain.[6]

On-chain governance
A system for managing and implementing changes to a cryptocurrency blockchain [6]

7 References

[1] “Agrivi”. [Online]. **https://www.agrivi.com/en**. [Accessed: Dec 30, 2019].

[2] “WeFarm4”. [Online]. **https://wefarm4.eu/cooperation-among-farmers-2/**. [Accessed: Dec 30, 2019].

[3] “Peer-to-Peer Networks Explained,” *Binance Academy*, 30-Dec-2019. [Online]. Available: https://www.binance.vision/blockchain/peer-to-peer-networks-explained. [Accessed: 30-Dec-2019].

[4] Mycryptopedia. (2019). *Genesis Block Explained - Mycryotpedia*. [online] Available at: https://www.mycryptopedia.com/genesis-block-explained/ [Accessed 30 Dec. 2019].

[5] Malcoded.com. (2019). *Storing Data on the Blockchain: The Developers Guide*. [online] Available at: https://malcoded.com/posts/storing-data-blockchain/ [Accessed 30 Dec. 2019].

[6] Blockchain Training Alliance. (2019). *Global Glossary of Blockchain Terms 2.0 in 5 Languages*. [online] Available at: https://blockchaintrainingalliance.com/pages/glossary-of-blockchain-terms [Accessed 30 Dec. 2019]