Report

Soychain

A sustainable production and trustful supply chain of soybeans

Radek Zenkl (zenklr@student.ethz.ch), Julian Neff (neffj@ethz.ch),
Pascal Küng (paskueng@ethz.ch), David Baur(davbaur@ethz.ch),
Remo Kellenberger(remok@ethz.ch), Jasper Shi(shijasper@hotmail.com)

All members contributed equally to this report

Introduction

There is a growing trend in our society for organic products. People begin to care about the origin and sustainability of the products they buy. In Switzerland the market share of organic products rose from 4.6% in 2007 to 9% in 2017. 26.6% of all eggs, 23.1% of vegetables and 22.1% of bread sold in 2017 in Switzerland was organic[1].
Organic sales in the U.S. totaled around $49.4 billion in 2017 which is an increase of around 6.4% from the previous year. The market share in the U.S. of organic products has now reached 5% and it has developed to the fastest growing sector of the U.S. food industry[2].

In the last couple of years one can also observe a growing trend in companies that sell conventional food as organic. As some examples, in May of 2017 a shipment of 36 million pounds of soybeans sailed from Ukraine to California. The soybeans were labeled as ordinary soybeans when they started their journey and were therefore also treated with pesticides but by arrival the documents and labels changed to organic soybeans. Leading to a $4 million profit to the fraud company. The fraud only got detected when already 21 million pounds of beans already were transported to their respective buyer[3].

In another case a farmer in Missouri tried to sell more than $142.4 million worth of falsely marketed organic goods between 2010 and 2017, by buying conventional goods from farmers and then mix them together with organic goods or even sell them directly to the customer as organic[4].

How can we be sure that we are really buying organic food? With today's system of how we handle the production and transport of food it is nearly impossible to ensure trust and traceability. End Users only have a label of a company that promises them that the food is

---

[1]https://www.swissinfo.ch/eng/business/agriculture_demand-for-organic-food-grows-strongly-in-switzerland/44088596

[2]https://ota.com/resources/market-analysis

[3]https://www.washingtonpost.com/business/economy/the-labels-said-organic-but-these-massive-
imports-of-corn-and-soybeans-werent/2017/05/12/6d165984-2b76-11e7-a616-d7c8a68c1a66_story.html?utm_term=.689127e3b187

[4]https://apnews.com/db6dcad19f6d405ea331ca4be38fd12f

organic and these labels are heavily dependent on strong relations with the customer. But with this growing trend in committing fraud by companies people cannot trust Labels anymore.

Even if there are no such intentions by a company itself, small unintentional mistakes done during the production can also lead to such fraud and can destroy the relationship to the publicity.

This leads to a growing desire in our society for a way of ensuring trust in organic end-products and also ensure traceability of the whole supply chain. But until now, this hasn't been possible in such a way that selling the goods afterwards is still profitable and sustainable.

If this trend in committing fraud continues then, the trend in buying organic food will pass and people will go back to more cheaper conventional food.

Following the United Nations Sustainability Development Goals we need to create a new model of how our supply chain works. This will be done with a truthful system, that detects losses and frauds and that cannot be manipulated by farmers, traders or other third party.

Furthermore with such a truthful system, resources will be used more wisely, because people will be able to trace all informations about their product. Farmers will take more care about the harvest and storage of their product. Traders want to ensure that no food will get destroyed or mixed during the transport. This will lead to less food waste.

Therefore companies also have a desire to proof the origin of the product. If a company is able to proof, people can be sure that their food is organic and are more likely to buy from this company than from other companies that are not able to proof organic food.

The use of Blockchain and IoT gives us now an opportunity to solve this problem with a system where frauds will be detected and safety of data is provided during the whole supply-chain. Our task is to develop a concept on how to integrate blockchain technology and IoT devices in supply chain systems. In particular, we have to develop an idea and a prototype, which ensures a sustainable production and trustful supply chain of soybeans.

Challenge

The Challenge is built around a specific scenario given by Peterson Control Unions representative Bogdan Weishaupt. There are three farms located in Ukraine. Each one has multiple fields growing soybeans. Farms can have warehouses, trucks and they can be certified as organic soybean sellers. In this specific scenario is a uncertified farmers. Each certified and uncertified farm has a contract with a trader who buys the beans from the farmers. The farmers either deliver the beans to the Trader or the trader uses his trucks to pick up the soybeans. This trader has a warehouse in Ukraine to accumulate the material before the export. He also has one contract with a logistics company for national and international transport with whom he transports the soybeans to switzerland and another contract with the final buyer in Switzerland to whom he sells the beans.

We need to build a system with Blockchain that ensures full traceability of organic certified soybeans produced in Ukraine and transported to Switzerland.

The system should also ensure that the volumes are recorded for all units during each harvest of a field and that the harvested beans can be traced back by the buyer to this field. Furthermore, we need to ensure that farmers can only sell what they harvested and the trader only sells certified organic beans.
The whole process from harvesting to the transport to Switzerland consists of the following steps and for each one we need to ensure that no fraud or mistake did happen:

- Harvesting the soybeans (Done by the farmer)
- Transport to warehouse (Done by the farmer with his trucks)
- Transport to traders warehouse (Done by either the farmer or the trader)
- Cleaning and drying of soybeans at the traders warehouse (Done by the trader)
- Packaging into Big-Bags with a capacity of one ton (Done by the trader)
- Transport to Switzerland to the buyer (Done by the trader)
- Processing of soybeans (Done by the final buyer)

In most steps, a possible fraud or mistake can happen which we need to detect.

For the farmer during the harvesting, he needs to proof that the beans are harvested without the use of pesticides and that they don't get mixed with beans from other fields to ensure traceability. Until now this has been a particular big problem in developing countries, because there was simply no profitable and sustainable way to do that before the time of blockchain and IoT. The challenge in this stage is to ensure the farmer has no chance to trade anything different from what is growing on his field. If for example the farmer owns only a 60 ha field of certified soybeans, it must be proven that everything he trades has its origin on this field.

In the warehouse and the transport to the warehouse, he needs to ensure that the certified beans are professionally stored until the trader collects them or he delivers the goods to him. Warehouses need to fulfill certain standards like hygiene and room temperature which a trader is now able to proof. Every truck and its corresponding load arriving or leaving at the warehouse must be registered in the blockchain. It’s essential for the confidence of the system that this registering process happens to be independent from any participating party. Our solution should notice if some farmer or trader tries to store or release any soybeans not triggered by a corresponding valid transaction in the system.

The trader has to clean and dry the beans and then package them into big-bags in a way that they don't get mixed and are still being traceable by the end customer when he delivers them. These big-bags are a problem for its own and will be described later in the presentation of the idea.

He also needs to be able to show the journey the beans made when they arrive in switzerland. With full traceability of the route for the customer. A difficulty could be that each big-bag could make his own unique way through the supply chain. Therefore we need to track and log the entire route for each big-bag independently.

All these conditions need to be fulfilled by our system. Such that the customer has the possibility to track all these informations of the product he bought. Most likely with a QR code on the packaging of the product. In that way, the customer has all the information available on his smartphone and can track the origin back of his product.

Solution

Presentation of our idea

Our idea is based on a hyperledger concept. The overseeing company would give access to the hyperledger blockchain to its registered farmers. The farmers would have to enter all their harvested crops to the blockchain. Once the crops enter the blockchain, their whole process chain can be tracked from the farmer to the customer.
After this the crops can be processed, traded or stored, but their total volume (respectively mass) cannot change anymore. For the sake of simple trading, the volume of single soy units can change as any of the participants want. This means that for example during processing and packaging of the soybeans the previous soybean unit can be split into any desired sub-units. There is no loss of information during this process because one can always track which soy bag came from which bag.
This also contributes to the overall transparency of the system. The customer can see from which farmer and which field the soybeans originate from. Additionally, the behaviour of the hyperledger users is predefined. Farmers cannot do more than selling their crops to the trader. The trader can only sell as much of the beans as he bought from the farmers. The Terms of the hyperledger are strictly based on the capabilities of farmers and the trader. Crop production of the farmers would be limited by the size of their field, so that only a certain amount of soy can enter the blockchain each year. However this raises the question, whether the harvested crops come from the same field, as the farmer specifies. This separate challenge will be discussed later on.

For this to function, one has to make sure that the data entered to the hyperledger is actually true. This means that for example when hyperledger says that one batch comes from field A, that it actually comes from field A when entering the hyperledger. This is a challenge that cannot be solved by hyperledger itself. Although it is not implemented in our solution, using a combination of IoT devices can solve this. Combining different sensors together and enabling them to enter the blockchain as well would add additional party to the whole process.

This could look like as following: When a trader buys or sells a specific amount of soybeans an independent volumetric sensor can automatically confirm the given amount and approve the transaction in the hyperledger. To determine whether the soybeans come from the given field a gps can be mounted onto the trucks that transport the soybeans from the field to the farmer or the trader. The fused readings from the gps and volumetric sensor guarantee that the stated origin is correct.

One challenge that our proposed solution does not cover is the possible fraud case, when the trader can sell certified beans to multiple customers. Since the other customer does not use hyperledger, respectively does not enter our hyperledger, there is no way of telling how much of the certified beans get sold elsewhere. The Trader can purchase uncertified beans and sell these to the customer with hyperledger as if they were certified. As long as the trader sells the same volume of the real certified soybeans to a different customer there is for us no way to tell as long as the trader sells the same amount as on the blockchain. However, if this fraud is discovered by the customer the exact moment of the fraud is traceable. Regretfully this problem cannot be sold without occasional checks of the quality of the products.

Design

As stated before, we used the Hyperledger framework instead of the Ethereum blockchain for our solution. We will look at the main differences between the two and the key features of the Hyperledger, which led to this decision.

The Hyperledger project itself is a collection of different open source blockchains and blockchain tools. It was developed by the Linux foundation and is being supported by different tech companies such as IBM, intel and more. The main difference between the Hyperledger blockchain and the Ethereum blockchain, lies in its application. While both are distributive ledger systems, the Hyperledger blockchains have been developed for companies to be used for a special application. This is the reason, why the whole project consists of many different tools and blockchains, such as Composer (which we used), Fabric, Sabretooth etc. It is not one big blockchain, on which all the smart contracts are deployed, and on which is publicly available to all other users, as it is the case in Ethereum, but it is a highly customized solution for a single business. Also, Ethereum is a delocalized network, so all the power over what gets written onto the blockchain lies with all the participants of the system, whereas in the Hyperledger the power over the system in the end is held by a single entity, which in this case would be the company running the Hyperledger. Furthermore, the Ethereum blockchain is based on a cryptocurrency, so all transactions and actions performed on the blockchain will cost Ether, on the other hand the Hyperledger is not. One other big important difference is privacy. In Ethereum all the network nodes have access to the ledger and therefore anyone on the blockchain, if business or customer can look up all transactions. With Hyperledger there exists an access control mechanism such that only nodes with the necessary permissions can access and more importantly interact with certain data. The access control mechanism for our solution is explained in more detail furtheron.

So, in conclusion the Hyperledger offers more privacy and a more fine-tuned access control for the companies which use them, a more specialized system which can be customized to their needs and remove the need for an underlying cryptocurrency.

For our solution we choose to work with Hyperledger Composer, which offers an online development environment which is based on Java Script and gives us the possibility to run the developed solution on Hyperledger Fabric. Also it is based on Assets, Participants and Transactions, which handle the asset movement between all the participation parties. This framework suits are given problem very well. For the access control there is also a file provided to regulate which involved parties can change what on the distributed ledger.

The Solution is implemented in three files: The Model file, in which we describe the participants, the assets and the transactions. A Java Script file, in which we can implement the actual logic of the transactions and an access control file described above.

In the following we want to briefly explain how each file is implemented and what different functionalities it serves.

We used two assets: fields and soybeans. The fields obviously not be traded, but they can be owned by farmers, so they also constitute an asset of some sort. The soybeans are described by an id for which we will use a string, by an owner which is one must be one of the participants, and integer which stores the amount represented by a single asset. And lastly a field id, which is used to determine with what quality the soybeans have been grown. The field in turn is also owned by someone, but in this case only by farmers, and it is identified by a string. It also has an integer referencing its crops production limit. This will be important in the implementation of the transaction, since we want to check, whether a farmer declares more soybeans as harvested from one field, than the amount given by the field capacity.

For the participants we used an abstract class trading partner which is extended by the participant farmer, trader and buyer. The abstract class contains an id string as to identify the involved party. The farmer has some additional information stored in Booleans such as whether he has a warehouse or if he uses subcontractors for transportation etc.

For the transactions we most importantly need a trading transaction, which allows participants to move crops around. Here we need the id of both involved trading parties and the amount of soybeans traded. For the farmers to harvest the beans there is a separate harvest transaction. This is used, since we want some additional functionality in this process which differs from the regular trading transaction, such as checking whether a field capacity has been exceeded. Here the field id is needed, the amount which is to be harvested and a name for the harvested beans.

Additionally, to the above transactions we also implemented a split and merge transaction. This is important since we do not want to impose, that all of a certain type of soybeans has to be harvested or traded in a single transaction. As an example if Farmer A produces certified crops and has already 20 units harvested in his warehouse and now wants to harvest 10 more units from the same field and store it in the same field, the merge function is called and will merge the two assets, which represent the same soybeans to a single asset with 30 units. A similar mechanism takes place if this Farmer now wants to trade only 12 of the 30 units stored in his warehouse. For this the split function is called and will create two new assets one with 12 units and the other with the remaining 18 units of soybeans. Now the new asset with 12 units can be traded.

All of the above described transaction functions are implemented in the Java Script file. To guarantee a bug free system it is important that each function checks whether a given transaction is even possible; for example, that a farmer can not trade more soybeans than he has in his warehouse. These cases are tested within each function and will produce and error message if a trade would violate the rules of our system.

Apart from the above described safety checks, the trading function essentially only changes the current owner of the soybeans and it also checks whether a split or merge needs to be made, in order to successfully complete the transaction.

As discussed above the harvest function checks the field capacity and will then create a new asset of the given soybean type with the given amount.

In addition to the logic of the trading and harvesting, it also it needs to be made sure that only the entitled participants can access these execute these functions, so that no third party can change the state of the soybeans when it is not allowed. These permissions are defined in a separate file where for each participant and each resource it is specified under what condition it is allowed to execute a certain operation. We will briefly explain the most important rules in the following paragraph.

Firstly, there is a rule which enables each trading partner to see its own commodities. Trading partners include farmers, traders as well as buyers. The rule allows a read operation in case the participant is the owner of the soybean asset. Also, to enable trade between all participants there is another rule which allows them to execute the trade function. It also checks that the trading partner is the owner of the soy he specifies in the trade function, so that only the owned commodity can be traded. Next, there is a rule which denies access to other trading partners trade histories, since obviously each participant should only see its own history. As already mentioned, we have a seperate harvest function which farmers execute when they harvest soybeans. For that, there exists a rule which only allows farmers to execute the harvest function. The soybean commodity gets created when farmers harvest their fields, which is why there exists a rule to allow farmers to create soybean when the harvest function is executed. Further, there are two rules which allow farmers to view their own field and another one to update it in case the harvest function gets executed, since then the amount of soybean on the field changes. Finally, we have a rule to enable each trading partner to view its own soybean assets since every participant should of course know its own commodities.

The frontend implementation is a simple skeleton interface based on angular. Since we didn’t spend time on improving the frontend, it only supports basic functionality. One could log in as either farmer, trader, buyer or admin. Dependent on the current users there are different functions available to use. If for example one is logged in as a farmer, it’s possible access the harvesting and trading functionality in the web app.

Evaluation

The included frontend demo, even though not completely up to date, shows a good overview over the current functionality of our project. In the following we discuss how the previously explained functions currently work.

When the harvest function is executed, a new soybean commodity is created and the respective farmer gets assigned as owner. A farmer is only able to harvest from his own field and the implemented limit allows him only to harvest what grows on his field because the amount gets decreased each time the harvest function gets called. Each harvest gets saved as “batch” with an ID and the amount of harvested soybeans.

The trade function allows trading between the farmers and traders. If the amount to be traded exactly matches the amount in a batch, the owner simply gets updated to the new trading partner. Since this is usually not the case, a batch gets split up into two batches with the respective amount to be traded and the amount that is left over. If, at a later step, soybeans of the same batch are again owned by the same participant, the batches get merged back together. This is explicitly shown in the demo, though this step is implemented automatically in the trade function. It corresponds to the real world scenario, where a trading partner sells only a fraction of some soybeans and therefore needs to split a load. If a trading partner receives in two different transactions bags of the exact same type of soybean, it’s not necessary for him to distinguish the loads and therefore he locally merges them in his warehouse.

The current frontend does not include an authentication system for the participants. For this reason, it looks the same of all participants and for each harvest/trade information has to be added which would automatically be filled if we could identify the user. At this stage the frontend is just designed for testing purposes of the backend and thus only provides us with minimal functionality.

Unfortunately we haven’t got time to properly test our backend implementation. Therefore we weren’t able to locate and fix any bugs. As mentioned before, the basic functions like harvesting or trading are currently working and passed some simple tests.

Conclusion / Outlook

We believe our project provides a good proof of concept to further develop on. It implements the basic functionality which is required to show how a possible solution could look like. However it’s important to point out, that due to the fact that we used the hyperledger framework, it requires an independent institution like for example Peterson Control Union which runs the business network. The institution has to make sure that field sizes, warehouse capacities and certifications of the participating farmer and trader are integrated correctly into the system and match the reality. This leads to another possibility to cheat, since human interaction is needed to do this tasks. It’s hard to prevent this fraud because of the use of the hyperledger framework. The business networks are not completely decentralized and thus some participating parties have more rights than others. Nevertheless, it makes more sense to use business networks based on the hyperledger composer framework instead of an implementation on completely decentralized blockchains like ethereum for this specific use case. In further development and optimization of the project one could think about implementing the supply chain system entirely on the ethereum blockchain for example. However due to the fact that we didn’t thought about this scenario during the process of designing our solution, we can’t argue about the feasibility of this idea.

As a next step, we would develop a more user-friendly front-end and provide an authentication system for all the farmers, traders and buyers. This would enable us to show a different interface to the different kinds of participants and make the functions easier to use for the end-user.

Furthermore, we would also integrate IoT devices into our project. Even though we were not able to do this in the scope of this project, we thought about how a possible IoT integration would look like. Weight sensors and Vol-Tech scanners at the farms and in the trucks could provide the hyperledger system directly with real-life data. Also the trucks used for transpotation could be fitted with gps sensors. The data obtained by the IoT devices could then be used to trigger functions and thus prove correctness of transactions or alert the supervisory institution that someone is trying to cheat. This would further increase the difficulty level for cheating and increase independence. In addition each unit bag of soybean could be fitted with a QR code and/or a GPS tracker. A QR code which would then show the asset on the hyperledger blockchain would provide traceability for local staff at each step of the way. The important thought here is that the QR codes are somehow manufactured into the packaging such that if one was to exchange the content of the bag it would destroy the QR code. Thus binding the QR code to the product itself to increase security. One could further improve traceability by adding these QR codes to the final packings in the shop, such that customers are able to scan the QR codes. The codes serve as a digital label to prove certification. As a nice side effect, customers get information about the origin of their product by scanning the label. GPS trackers which periodically upload their position to the hyperledger would of course further increase traceability for all participants involved. All in all, our current solution paired with the described IoT functionalities would lead to a near tamper-proof supply chain in the soybean industry. Another possibility would be to generalize our solution to not only serve as a tool for the soybean supply chain. Instead it could be adapted to support other industries, which are similar to the soybean industry as well.

One major downside of all these additional sensors and mechanisms shall also be discussed briefly. If one of the nodes connected to the Hyperledger is not equipped with the necessary hardware which includes all IoT devices and all QR-tag Hardware, the whole traceability would be lost. Since it then would be possible for potential cheaters to exchange produce at this point in the chain without anyone realizing. So if a standard in hardware is set by the company running the Hyperledger all participating nodes will have to obey this standard. This makes it difficult to realize such a system, especially if there are many different nodes in the system. To make this work a compromise between safety and feasibility needs to be found. This is highly dependent on the kind of enterprise and on the likeliness of someone cheating.

The previously mentioned additional features require adaptations in the backend and frontend as well as the development of a second application, which implements an interface to interact with the added QR codes.