Eidgenössische Technische Hochschule Zürich

BETH 2019 - Blockchain for Sustainability

REPORT

Trust In Global Supply Chain

Authors:	*Student Number:*
Luca Colagrande (colluca@ethz.ch)	18-950-477
Giovanni Lippolis (glippoli@ethz.ch)	18-946-046
Davide Menini (dmenini@ethz.ch)	18-933-192
Felix Stabel (fstabel@ethz.ch)	17-949-488
Alexandre Waibel (waibela@ethz.ch)	13-917-570

"All members contributed equally to this report"

April 30, 2019

Contents

1 Introduction: Context of Challenge

Ensuring traceability of goods from the field to the consumer is becoming more and more valuable because of the growing interest in high-quality products. A great effort is required to guarantee this property to the level of satisfaction of the customer. This is among all aspects an incredibly complex challenge as so many difficulties can arise in this frame. As an example, contaminations can occur from simple mistakes in the logistics or, even worse, from fraudulent behaviour of any actor in the supply chain (farmers or traders). To this end it is necessary to rethink the way the supply chain works, embracing new technologies as a resource to build a truthful system and counter both losses and fraud. Among these the Blockchain stands out as a potential solution for its immutability and privacy related warranties.

2 Challenge

This report is aimed at offering a potential solution to the aforementioned issues and challenges in the specific case of soybean trade, leveraging in particular the Blockchain and IoT technologies.
The goal is to ensure full traceability of organic-certified soybeans, produced in Ukraine and transported to Switzerland, along the whole supply-chain. Our solution deals with a simplified, but highly generalizable and easily scalable, scenario.
We consider three farms working independently in different contexts, all of which interact with a single trader:

- Farm 1 produces organic-certified soybeans. It has 2 fields, for a total of 180 ha. Moreover, it has its own warehouse and uses 2 trucks for transport between field and warehouse. The contract with the trader is for 540 tons, picked up from the trader in the farmer warehouse.
- Farm 2 again harvests organic-certified soybeans. It has 3 fields, for a total of 127 ha. Unlike farm 1, it does not have a warehouse or trucks of its own. Instead, it uses directly the trader warehouse and a subcontractor logistic company for transport. It has a contract with the trader for the whole harvested material.
- Farm 3 doesn't harvest organic products. It has 2 fields for a total of 150 ha and has 4 trucks to deliver the non-organic material to its own warehouse. By contract, the trader picks up the whole harvested material from there.

The trader stores both certified and non-certified soybeans in its warehouse in Ukraine, where the material gets accumulated before being exported to Switzerland. The international transport is carried out by an external logistics company.
The final buyer, which resides in Switzerland, has a contract with the trader for only organic-certified soybeans.

The supply chain is comprised of the following stages:

- Harvest: This first stage is carried out on each field by dedicated machines which gather the produce and unload it on some trucks.
- Transport to warehouse and/or trader's warehouse: The trucks carry the raw produce to a dedicated warehouse, either property of the farmer or directly to the trader's warehouse.
- Export preparation: Inside the warehouse the soybeans are cleaned and dried in preparation for being exported. They are then packed into bags to be carried outside of the facility.
- Transport to buyer factory: The bags are loaded onto trucks responsible for the transport to the final buyer.
- Processing of soybeans: Once the produce arrives at destination it further undergoes processing before being ultimately dispatched to the consumer.

We aim to achieve several objectives. Specifically, volumes must be recorded for all units, namely certified and non-certified batches. Each batch must be linked to the corresponding farm of origin, to a detail of individual fields among each farm. Each farm is enforced to sell only what it harvests. In addition, the trader is guaranteed to sell only certified soybeans to the buyer, who is provided with complete traceability over the goods.

3 Presentation of idea

In order to succeed in our achievement we need to address several pitfalls which manifest in the supply chain, all of which appear at the interfaces between the different stages.

Specifically the single most important property which needs to be guaranteed is there must not be introduction of external untracked produce in the supply chain. This requires monitoring consistency of goods at entry and exit points of consecutive stages. That means, for instance, that the goods received at the warehouse must match the sum of those sent from the farm and harvested on the fields. Also the sum of goods exiting the warehouse must match the sum of those entering the warehouse and so on. If this holds, there is no incentive to fraud for any party at stake because the total value of outgoing goods at any stage is known to the buyer from the following stage, who can monitor every other transaction the seller makes, and verify that no goods are sold for a value greater than what they are worth. This is done by registering (as we will do on the Blockchain) all goods which enter or exit any stage and verifying their equivalence. For instance, assume the trader receives 200 tons of organic and 200 tons of non-organic soybeans. It is reasonable to assume that he will not attempt to sell the non-organic produce as certified organic produce to the buyer. In fact, if he did, he would be constrained to sell the 200 tons of organic produce as non certified to satisfy the goods matching criteria which is publicly verifiable on the blockchain. If he didn't, a violation of such criteria would signal a potential fraud. Given this constraint, the trader cannot make any profit from counterfeit as the overall value of the goods he can sell is fixed by the goods he has received. Thus it is "rational", in a game-theoretical sense, to assume that he will not sell the buyer non-certified produce, as this would lead him to no monetary benefit but to an unjustified risk in terms of commercial reputation if the fraud were exposed. The same holds also for the exchanges between farmer and trader. If Farm A harvests 300 tons of organic soybeans it will not be able to sell the trader more than what it harvests if only those 300 tons are registered. Indeed registering will be automatically performed by physical IoT systems installed by Peterson which cannot be manipulated and which register any exchange of goods they can measure. For the same reason we can assume that inside the warehouses each batch will conserve its independent processing line, and no certified and non-certified produce will be mixed and/or exchanged.

The whole system works under the assumption that there cannot be any unregistered transactions, for instance with unidentified third parties. Removing this assumption to extend the previous example, the trader could sell its certified produce to a third party without registering the transaction, and buying an equivalent amount of non-certified soybeans, ending up with 400 tons of non organic soybeans. When trading to the registered buyers he can still sell half of its unsold (non-certified) produce as certified, since the outgoing volume is still at zero. This fraudulent behaviour is incentivized by the net profit he can earn through counterfeit with respect to the "fair" trading strategy and thus must be countered by preventing any irregular untracked exchange of goods.

A simple solution is provided if we assume unauthorized goods can only enter the supply chain but none can exit. Then registering outgoing and regularly incoming volumes, is a sufficient measure. This is a simpler case compared to the example made above. In this case consistency of goods between any two points in the chain is guaranteed by automatically registering and comparing the volumes of certified and non-certified produce. If a mismatch in volumes is detected this implies there has been a contamination. Suppose for instance Farm A doesn't harvest 540 tons of certified soybeans but a lower amount of 300 tons, thus saving on the production costs. If Farm A then buys 240 tons of non-certified soybeans from an external source, it can profit by selling these as counterfeit certified produce to the trader. However, since Farm A cannot register the 240 tons of incoming produce (since the source itself is not registered) a mismatch between the regular 300 tons produced and 540 tons sold will signal a contamination.

Back to the original example, if unauthorized goods can both enter and exit the supply chain, there is no way to prevent a fraud. It is therefore necessary to physically prevent any unregistered transaction at least in one of the two directions (incoming and outgoing) between any two points in the supply chain. This is achieved by installing security IoT systems which automatically register any exchange of goods. In no other way can any goods be registered which are not following a path approved and controlled by Peterson via installed security systems. We will thus have to deal with the issues of registering all batches at the endpoints of transport routes and guaranteeing consistency during transport. All trucks will have to be adequately monitored and to be even more precise, we will also assume that external contamination could be carried out directly on the fields, when the trucks are loaded by the harvesting machines. All these specific duties will be addressed properly in the next section, where we will design the autonomous IoT security systems mentioned above to fulfill such tasks.

That being said, how and why should we employ the blockchain and Internet of Things for this purpose?

Generally speaking, the blockchain provides a transparent and immutable environment for registering transactions of any kind. Basically, it is a distributed ledger that grows a chain structure made of different blocks: each block contains information about the previous block in the form of a cryptographic hash. Due to the fact that all the blocks are related, no one can be modified without the whole subsequent structure being altered. In other words, the history of the chain cannot be corrupted and every change will be noticed.
The system is not completely immune to attacks and corruption of data, but it is a safe-by-design environment, meaning its insecurities and weaknesses are measurable and thus predictable and avoidable under certain assumptions. This is why it can be used to exchange transaction data transparently and reliably without fear of counterfeit. In particular we will be making use of the Ethereum platform and an architecture developed around smart contracts. Smart contracts are essentially computer programs which are executed on the blockchain and can thus perform any computation on some input data to verify compliance to certain criteria.

In our case, we will be using Internet of Things (IoT) devices to gather data about exchanged goods that will be forwarded to the blockchain to be registered and stored. Each IoT device is associated an Externally Owned Account (EOA) uniquely identifying it in the system. In following stages in the supply chain these data will be queried and compared against to prove consistency and reliability in the whole process.
Nowadays IoT sensors are cheap and small, allowing to be easily integrated into machines, trucks and so on. They constitute an automatic and thus reliable interface to the physical world, continuously monitoring for events which need to be tracked, e.g. the counterfeit goods exchange introduced before.

We will be employing IoT devices to both track the regularly produced soybeans, and to verify that no unauthorized exchange of goods takes place. All captured transactions will then be automatically forwarded to the blockchain where they will be permanently stored and registered for future consultation, both from human operators, like directly interested parties, and the IoT devices installed by Peterson in the following stages, which compare these transactions to ensure consistency between goods throughout the supply chain. In the next section we will describe the detailed design of such IoT devices and blockchain solutions and how they are employed in each specific stage.

4 Solution design

Our solution provides several IoT modules which are dislocated along the supply chain and gather data about moving produce. These devices are connected to the blockchain and are each registered with a unique unalterable identifier (EOA) which is assigned at the moment they are installed by Peterson. No other device can steal that identity to send fake data under the same name on the blockchain. This allows only trusted sources to register goods inside the supply chain.
Starting from the entry of the supply chain, we want to register all and only product which is harvested on recognized fields, e.g. the 7 fields from farms A, B and C. To avoid any contamination already in between harvesting machine and truck to be loaded to carry the soybeans to the warehouse, we want some system to be mounted on the harvesting machine itself to automatically prove the amount which is actually harvested on the field. Thus, we want to have some kind of flow measurement device that is recording the volume of the harvested good. In the case of

soybean harvesting, the machine cuts the ripe plant into a granular shape and fills a truck that is following the harvest machine through a long tube. In this tube, a Solid Particle Mass Flow Meter can be installed. For example, Eastern Instruments' CentriFlow Meter could be used. This device is a flow sensor that accurately measures volume flows by the principle of centripetal force. It is a durable, maintenance-free and compact system that is very cost-effective and reliable. (source: [1]). Furthermore, the harvest machine will be equipped with a GPS sensor in order to record the exact geolocation of the field being harvested. Consequently, the harvested beans can be identified as either organically grown soybeans or not, by knowing their field of origin. Additionally, as the area of the specific field is known and easily verifiable by satellite images, an approximate amount of harvest good can be estimated and expected to match the amount measured by the CentriFlow Meter. Furthermore, the GPS device can make a timestamp at the beginning and the end of the harvesting process, giving us an additional information that ensures traceability. We call this combination of a volumetric flow meter and GPS device IoT Module A.

The data gathered is immediately sent to the blockchain from the module. We provide the functionality to do so through a smart contract which is named *HarvestContract*. The contract is deployed by Peterson and has his EOA internally hardcoded to allow certain function calls to be issued by Peterson only. This is achieved by declaring such functions with the modifier *onlyPeterson* which checks that the caller's EOA actually matches the hardcoded one.

At system setup, prior that any call be issued to the contract by any account, Peterson would register all actors which are authorized to take part in the supply chain. This is done by assigning them a unique ID in the system, along with their EOA address in case the actor will interact with the blockchain. This tuple is declared in the *sharedlibs* module and referred to loosely as an *"Account"*. The contract exposes the following functions which are dedicated to such purpose:

- *registerFarm*: registers a farm with a specified ID. The farmer does not need to own an Ethereum account, this function only serves the purpose to associate harvesting machines, trucks, fields and warehouses which are authorized to interact;
- *registerHarvestingMachine*: registers a harvesting machine associated with a farm. It takes as input the account of the machine and the ID of the farm it belongs to, which must have been previously registered;
- *registerFields*: registers multiple fields associated to a farm. It takes as input the geometrical and geographical data describing the field, the certification status of its crops and the ID of the farm it is associated to.
- *registerTrucks*: registers multiple trucks that are allowed to carry produce which is exchanged by a farm. It takes as input the account of each truck, their capacity and the associated farm.

Once a harvesting cycle has been completed, IoT Module A would call a function named *storeHarvestData* which receives the GPS coordinate range (including time) where the harvest has been carried out, the volume which has been harvested, the certification status of the harvested produce and the truck ID onto which the product has been loaded. This function is declared with the *onlyHarvestingMachine* identifier, such that only EOA accounts previously registered as harvesting machines in the system are allowed to execute it. Before storing the data in memory, the function checks that the truck ID is registered and associated to the same farm, that the harvest doesn't exceed the capacity of the truck and that the GPS coordinates where the harvest has been carried out match those of a field registered to the same farm and with the same certification status of the declared harvested crops. If all conditions are met then the load-truck pair is stored for further verification at arrival to its destination, otherwise an *unauthorizedHarvest* event is emitted which will permanently be visible on the blockchain recording the violation.

Trucks themselves are equipped with an IoT module (B) which serves the purpose of verifying that the load is not altered during transport, i.e. from field to warehouse or from warehouse to trader. In order to achieve this, a type of IoT sensor must be considered which can monitor the cargo at all times during the transport. A simple way to control this would be by using a truck load sensor registering the weight of the beans during the transport. However, we do not choose this solution because it is susceptible to failure. Think about a heavy rainfall during the transport or harvest and loading time, where the moisted beans could be significantly overweight. A more reliable, although more expensive IoT device that is able to monitor a truck load is an optical sensor. There are several truck load tracking systems available on the market, a deluxe version is the SC200 Cargo Sensor by Orbcomm. This device provides visibility of the truck load and sends an alert when the load status changes or when the sensor is damaged or removed. It can be combined with a solar rechargeable battery to ensure unlimited service with no maintenance needed. (source:[2])

Together with a simple GPS device, these sensors build IoT module B. If any violation according to the mentioned criteria is detected, the device will call an *invalidateLoad* function removing the previously stored load-truck pair, such that no warehouse can later accept the load from this incoming truck. The concerned load-truck pair is identified searching for the caller's EOA. This way the only device able to invalidate a load is the one mounted on the truck itself.

The contract related to the second stage of the supply chain is called *WarehouseContract*. Similarly as before, some modifiers like *onlyPeterson* and *onlyWarehouse* are needed to ensure that the functions are issued by authorized callers. First of all, Peterson has to register the authorized warehouses on the blockchain. This is easily done by calling the function *registerWarehouse*. A unique ID and an EOA address are provided to each warehouse, to identify them and allow them to interact with the blockchain, and an array of Farm IDs is passed to register which farms the warehouse can collect from. The latter data structure could be augmented to include limits on the amount of produce which can be collected from each farm and so on.

When the truck arrives at the entry point of the warehouse, we must check that its load is registered, that it comes from a previously verified source, and that the truck is associated to the same farm as the warehouse, i.e. it is unloading in the correct place. If an unidentified truck (source of counterfeit soybeans) or, equivalently, an identified truck but with a non-verified load approaches the warehouse, it must not be allowed to unload its product. Before giving access to the truck, the warehouse will therefore attempt to identify the truck and issue a call to the smart contract function *verifyTruck* passing it its ID. If the above criteria is satisfied the truck will be granted access to unload in the warehouse. Otherwise, *reportInvalidTruck* will be called emitting an *invalidTruck* event on the blockchain.

The way the warehouse could identify the truck is flexible. We could either register the truck with its license plate and then retrieve its ID from the license plate, or we could have a device (IoT module C) directly communicate with onboard module B to retrieve the ID. Both approaches are adaptable to both a human operator or an automated system (equipped with cameras and/or wireless communication capabilities) managing accesses to the warehouse. The latter approach is preferable as all calls to the blockchain would be automatically forwarded and, assuming the module is physically fixed, localized, thus guaranteeing reliability.

If the truck passes the check, the soybean batch is unloaded from the verified truck in the warehouse. Then, an additional control could be performed to check that the volume of the unloaded batch is not greater than the amount loaded on field: if the volume of the batch is higher, meaning that the truck has loaded other soybeans from an unregistered source, an *invalidVolume* event is triggered, even though we provided IoT module B as protection against such condition. If the load is safe, it gets accepted and via a call to *unloadTruck* the load is unregistered from the blockchain, preventing the truck which is now actually empty to be reused for fraudulent deliveries. At the same time a batch-warehouse pair is registered to the warehouse account invoking the function from its registered EOA. The batch data in particular inherits all the product specifics which were stored in the load data, in addition to new information such as the timestamp at its entrance in the warehouse.

Now the batch can enter the processing line, where it will undergo cleaning and drying, before being eventually packaged into bags.

When each bag is packaged it shall be also stamped with an RFID (Radio Frequency IDentification) tag. At the moment of stamping, another IoT module (D) shall connect to the blockchain and register the bag via a call to *assignRFID*, linking the assigned RFID to the product specifics propagated up to this point in the supply chain, where the produce is finally shaped into the form which will be received by the end buyer. In particular, each bag now has accumulated information about time and location of harvest and type of soybeans, field ID and farm ID, warehouse ID, timestamp at the product's entrance in the warehouse, timestamp at its packaging and so forth. The RFID stamp is meant to provide an additional means against counterfeit since it is unique upon assignment and cannot be reproduced externally. Contamination is thus prevented by registering all and only internally assigned RFIDs.

This is done by authorizing exclusively the warehouse installed IoT device to call the function by

appending the *onlyWarehouse* modifier, which provides the necessary verification.
At the exit point of the warehouse, the bags are loaded on different trucks, headed towards the trader. A proper function called *loadBagsOnTruck* takes care of this work: the loaded bags, with their RFIDs, are assigned to the corresponding truck ID to keep track of this loading. Again, this process must be allowed only upon authorization, thus the modifier *onlyWarehouse* is needed. As the bags are loaded on the trucks they are also deregistered from the pool of bags pending in the warehouse, and are instead stored as a bag-truck pair in an array of bags in transit.

The trader defines the third part of the supply chain and we provide a smart contract related to all exchanges during this stage called *TraderContract*. Here there is a new actor that has to be given access to the Blockchain, so the new modifier *onlyTrader* is needed.
The function *registerTrader*, which can be called only by Peterson, is used to register new traders if it is needed, despite the fact that in our challenge there was only one trader. It will take a set of warehouse IDs as input, to identify the warehouses which are authorized to deliver to the trader. When a truck arrives at the trader it is either destined to go to the dispatch facility or to the warehouse, which we assume to have separate entrances. The trader's warehouse is managed in the same way as any farmer's warehouse, and thus relies on the functionality provided by IoT modules C and D and *WarehouseContract* as described above.
The dispatch facility is fed with the bags coming from the trader's warehouse and those sent from the farmers' warehouses on dedicated trucks. Once the latter approach the facility's entrance, they are treated in an analogous way as trucks carrying raw soybean loads at the entry to the warehouse. Namely they are subject to identification and verification of their load. Only bags provided with registered RFIDs and whose associated warehouse ID matches one registered to that trader will be accepted. In case of any violation either an *invalidTruck* or an *unidentifiedBags* event will be raised and emitted on the blockchain. If no violation occurs, the bags are accepted and removed from the pool of bags in transit. At the same time, they are registered to the trader's facility similarly as done for batches registered to warehouses.
A special note must be made for the second farm, which doesn't have its own warehouse, and sends its produce directly to the trader's warehouse. There is absolutely no difference in the transport between field to farmer's or trader's warehouse. Also, there is no difference in the treatment which the produce undergoes inside the two warehouses, which can thus be treated and managed in the exact same manner. The only difference is that the bags packaged inside the trader's warehouse will skip the second transport stage and will therefore not be submitted to any verification stage between warehouse and dispatching facility. Once the bags reach the dispatching facility they all follow the same dispatching flow and are treated equally.

5 Evaluation and Discussion (features, bugs, etc)

It is clear that this solution is not optimal and can be improved by making a better usage of the blockchain and its resources. Indeed the blockchain allows to achieve transparency and decentralization, but at the expense of speed and computational cost. The main problem in our solution is employing the blockchain to store a large amount of data, e.g. the product specific details propagated throughout the supply chain. However, the storable amount of data on a blockchain is limited either by the protocol or by the cost of the transactions. To this end, we could use a database such as IPFS (Interplanetary File System) as primary storage for the data structures and reference them by their hashes in the transactions on the blockchain.

Moreover, the smart contracts can be enhanced by providing additional features. For instance, aside all *register* functions analogous *deregister* or *modify* functions could be implemented to make the system more customizable. A *deregisterFields* function would allow to exclude a number of specific fields from a farm, following a hypothetical reorganization of the latter. Furthermore, a *modifyTrucks* function would allow Peterson Union to successively adapt the registered truck fleet of a farmer or logistics company being currently used, as some vehicles may not be operating at all times due to maintenance or other reasons.

As a supplementary feature, any exchanges between parties, for instance through the *unloadTruck* function, could directly embed payment options to perform the payment in the native Blockchain currency. By requiring both parties' signatures we could furthermore enforce not only that e.g.

the warehouse recognizes the truck, but also that the truck approves the warehouse responsible of actually invoking *unloadTruck*.

We also want to point out that a solution design including too many IoT devices might not be the best solution for a supply chain monitoring challenge. Especially in terms of scalability, the amount of data generated by these modules must not be underestimated. A prominent drawback of the blockchain technology is its inefficiency when dealing with large volumes of transactions. A solution design with fewer monitoring devices would improve efficiency and overall cost, although conflicting with the objective of ensuring a certain security level and product traceability.

6 Conclusion

This project shows the possibilities of the blockchain and IoT technologies in the sector of supply chain management. The technical capabilities provided by these arising technologies enable to achieve a high level of automation, enhance trust and allow to revolutionize the payment system between the different actors in the products processing chain. The large amount of data volume generated remains a problem that can be mitigated by a good solution design, as previously mentioned.

With regards to the idea of sustainability, this project can be rated in different manners. On one hand, by enabling an unalterable track history of the organic-certified soybeans from the harvest field in Ukraine to the final buyer in Switzerland, trust between customer and producer is ultimately strengthened because fraudulent activities would be detected. Consequently, people are more willing to pay the higher retail price for organic-certified beans, since they have prove they really are. In this way, organic food is generally promoted and protected. On the other hand, speaking of a sustainable blockchain project would require some major improvement in the energy efficiency of the whole tracking system.

References

[1] E. INSTRUMENTS, “Centriflow solids mass flow meters.” Accessed: 29.04.2019.

[2] ORBCOMM, “Sc 200 full length cargo sensor.” Accessed: 29.04.2019.