Report for the BIOTS18 hackathon at ETHZ

Team “AmBros” - Smart Logistics

Challenge “open category” of BIOTS18, challenge initiative by “Ambrosus”

Authors:

Jingxuan He	(hej@student.ethz.ch)
Laurentiu Mirica	(lmirica@student.ethz.ch)
Marko Palic	(mpalic@student.ethz.ch)
Kay Spiess	(kaspies@student.ethz.ch)
Christoph Zürcher	(zuerchch@student.ethz.ch)

ETH Zürich, 30.04.2018

Statements

Contribution Statement

All members contributed equally to this report.

Open Source Statement

The software code which is part of this report is open source and available at https://github.com/ETHBiots2018/AmBros.

Academic Environment Statement

This project report was written as part of the spring 2018 course ‘Blockchain And the Internet of Things (851-0591-01L)’ run by M. Dapp, S. Klauser, and D. Helbling.

Author and Report Background Statement

For this project, the group had only one programmer at disposal.

The original chapter “payment solutions” was omitted in the final version of the report. The purpose of the chapter was to elaborate a detailed possible way to implement a payment solution, but was omitted in order to allow a deeper insight on the main focus (max. limit of words).

The number of words was counted to 5‘098, without any tables of content, tables, picture and table descriptions or statements.

Licence Statement

Abstract

This present report makes a proposal, how a smart contract between a delivery initiator and a general logistic provider could look like. The smart contract can replace only a few parts of a conventional contract; but through the "objectivity" of IoT devices and transparency and indisputableness of blockchain data, costly law suits could be avoided.

The report gives first a general concept and then looks at a specific example of a transport of a chemical reagent called "Isocyanate" which has to be transported under very strict temperature conditions.

In a real demo code, the two requirements of temperature and delivery time were implemented in a smart contract.

Table of Content

1. Introduction

Blockchain technology unfolds its true potential in processes, where multiple parties have to trust each other. This trust, be it that a service is performed or a payment is made if certain conditions are fulfilled, can be externalized to the blockchain. The blockchain user doesn’t have to trust the other party, if he trusts the system.

An obvious use case where multiple parties are involved and easily measurable conditions have to be fulfilled are supply chains. We believe, that especially the transport of sensitive goods can profit from blockchain technology. When handling sensitive goods such as drugs or chemical products, it is important to keep various conditions like temperature or humidity in between certain boundaries. The transport of these goods is a particular sensible part and it is difficult to control, that the pre-defined conditions are not violated. If a company wants to buy goods, they have to trust the logistics company, that the conditions during the transport are maintained. Furthermore, the logistics company has to trust the client that the transport costs are paid if the transport was successful. This is where blockchain technology can create a trusted source of information and automatic execution.

2. Problem Statement

We discussed numerous different applications of smart contracts in the supply chain. If we simplify a transport process, there are at least three parties involved (under the assumption, that the transport is performed by a logistics company). The buyer wants to obtain goods from the seller. The logistics company acts as an intermediary that transports the goods.

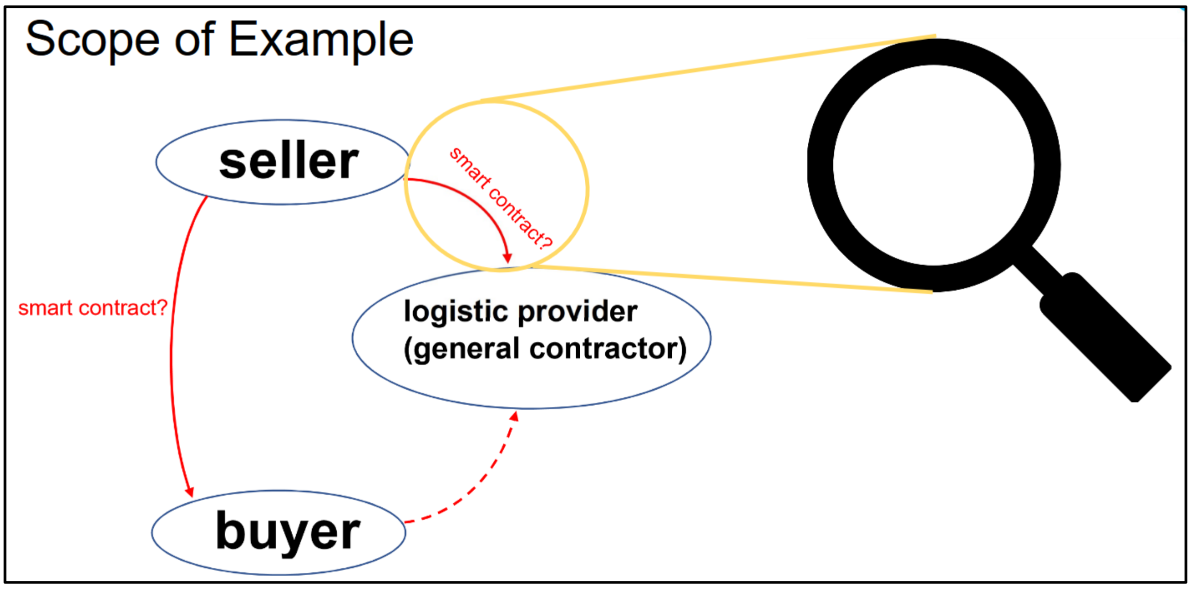


Figure 1: Simplified scheme of the involved parties in a shipment process.

An obvious use case for a smart contract is the relationship between the logistics company and the purchaser of the logistics service, which can be the buyer or seller of goods. For our project, we focused on this specific deal.

There are several factors in this process that one needs to take care of. Most importantly, the transport conditions have to be fulfilled and if that is the case, the logistics company has to be payed. In the following we will present a smart contract based solution, to simplify and improve this specific process. In our view, it makes most sense for logistics companies to provide such smart contract based solutions to profile themselves with more transparency as well as data based proof, that their transports worked well in the past.

3. Concept Proposal

Let’s look at an example: A company has sensible goods to transport. They contact the logistics company and define the conditions that have to be fulfilled during the transport. The logistics company sets up a smart contract involving these conditions as well as the penalties/ incentives that are applied when deviating from them. They also define the costs for this specific order. The customized smart contract is then presented to the transport initiating company. If the company accepts the conditions and prize, they sign the smart contract, automatically allocating the costs for the transport. The funds are now locked until the transport is over.

During the transport process, IOT sensors record the variables of interest and store them on the blockchain. As soon as the company receives the goods, the data recording is stopped. The data are read from the blockchain and the conditions are checked (automatically, off-chain). It is checked, whether the conditions during the transport were fulfilled and according to the penalties/incentives defined in the contract, the price is calculated. Then, the smart contract is automatically deployed and the funds that were locked up are allocated according to the calculated price. The logistics company receives the calculated prize and if there were any penalties, the rest of the initial payment goes back to the transport initiating company.

3.1. Process

For a better understanding, we specified in the following the different steps of the process.

1. Order initiator sends specifications to the logistics company via GUI on website
2. Logistic company creates smart contract (=offer)
3. Seller checks offer and signs smart contract with private key
    a. Transfer of appropriate shipping fees (example: ETH, in future: stable coin) into the smart contract
    b. The process triggers the formal order of picking up the goods
4. Goods get picked up
    a. handover confirmed by sender and shipper app
    b. data logging starts
    c. time and place are recorded

5. Transfer starts
    a. Temperature (and other variables) are logged in defined interval
6. Goods are delivered to destination
    a. Handover confirmed by receiver app
    b. Data logging stops
    c. Data is read off the blockchain and processed off-chain to determine whether pre-defined conditions held
    d. Final price is calculated off-chain according to conditions in smart contract
7. Smart contract is deployed according to off-chain price calculation (include hash of data to confirm that data used in calculation is the same as data on the blockchain)

3.2. IOT devices and technical setup

We specified the logistics company as the provider of the service. Therefore it is easy to include stationary IOT sensors into the means of transportation (example case: truck). A possible setup would include several sensors per truck, such that measurement errors can be detected and multiple measurements per time unit are available. Depending on the type of transport, more or fewer sensors can be activated. Furthermore, additional sensors could be attached right to the surface of the goods. The options here are manifold. We also discussed the possibility of semi-isolated sensors that are directly attached to liquid containers. In that case, only the temperature of the liquid will be measured and air temperature would not play a role. This would guarantee, that precisely the temperature of interest is measured.

The sensors are all connected to a central console that is, for example, 3G capable. In predefined time steps. The measured data can be written to the blockchain. Alternatively, also a private blockchain for the transport unit could be implemented. This could be useful in the case, where goods are shipped overseas and it is not possible to send data to the public chain for longer time periods. As soon as the ship arrives at the port and the console has access to the public chain again, the data can be sent.

For the data logging we thought of different concepts. To save unnecessary storage usage, the system could have two modes of action. If in a defined time interval no variables exceeded their limits, only a summary statistic of the variable will be sent to the blockchain. Otherwise, when a variable exceeded its limit, the whole log is sent. With this setup, a precise investigation in the case of a violation of the limits is possible. Of course, factors such as time intervals, number of sensors, how many times the sensors themselves measure the variables, how much data is sent to the blockchain etc. depends on the budget of the client and how sensible the goods are. Therefore, there is a lot of flexibility for price/service tradeoff.

Next to the measurement setup, a simple and easy-to-use app would be implemented to verify process steps 4.a) and 6.a). The app would verify that both parties are present and log time, location as well as the shipment identifier.

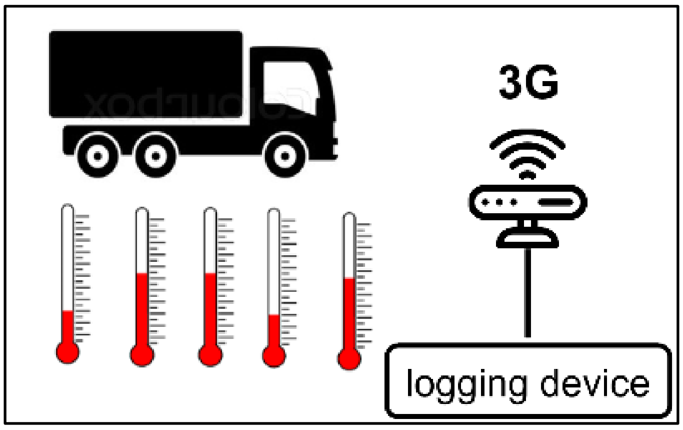


Figure 2: Five sensors are measuring the temperature, sending it to a central logging device. The logging device then sends it to the public blockchain via 3G.

4. Example Case: pMDI

4.1. Framework for Example

In this example case study, the concepts of chapter 3 are applied to a specific problem and a detailed specific solution is developed.

4.2. Problem Statement of Example Case

Polymers Diphenylmethandiisocyanate, short as pMDI and in the following just named “Isocyanate”, is a base chemical for the polyurethane industry. Global volumes of 21’000 Kt (Kilotons) of product are shipped by road trucks between production facilities and consumers. The chemical is sensitive to temperature and must not be exposed to temperatures below 5°C and above 40°C. In case these thresholds get violated, the value of the chemical is destroyed. Therefore, conditioning of the trucks is important and continuous monitoring is in the interest of seller, buyer and shipper of the goods.

This example is a simple example, because its result of the temperature control is binary. There is no “devaluation” of the goods if the environment parameter are not as agreed; it is either okay (value of temperature is kept, goods have still the full value) or the value is zero, in case the required temperature band is violated.

In this example, we want to transport 24’000 Liters of Isocyanate with a truck. The Isocyanate is stored in liquid form in 24 IBC Containers (IBC = intermediate Bulk Container).

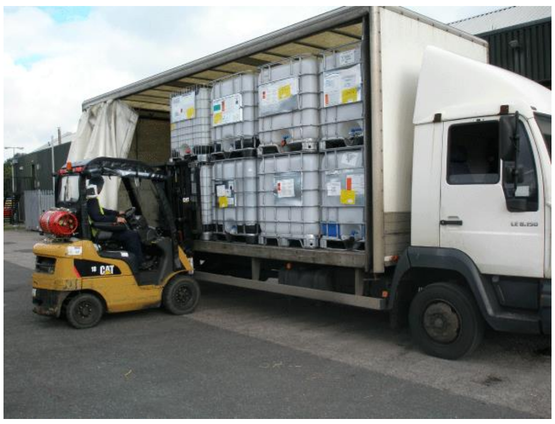


Figure 3: Truck for IBC container transport with a capacity of 24 containers;
illustration source: ibcrecycling.co.uk

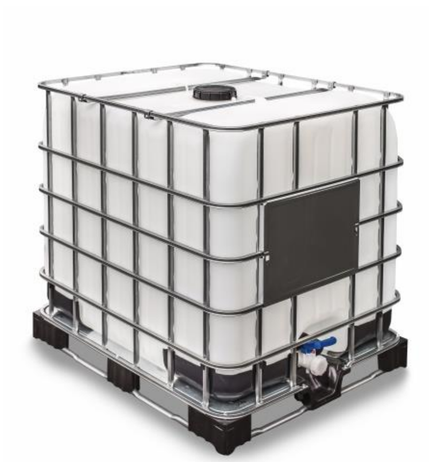


Figure 4: IBC container with a capacity of 1’000 liters;
illustration source : www.arnoldengineering.co.uk

To make it not too simple, we do not only have the requirement of the temperature band, but also of a specific delivery time. The goods shall be delivered within 48 hours. Unlike the temperature requirement (which is binary), the violation of the delivery time requirement opens the possibility for gradual impact of the payment via the smart contract.

The table below summarizes the requirements for the Isocyanate delivery which shall be covered:

Temperature requirements	Must not be lower than 5°C
	Must not be higher than 40°C
Time requirements	Delivery date must be within 48 hours

Table 1: Smart contract parameters for the blockchain application

Note that there could be a lot of other parameters like moisture, acceleration or purity. Note that only parameters can be implemented in a blockchain/IoT respectively smart contract solution, which are measureable by devices and where unambiguous data can be logged. This would be possible also for moisture and acceleration, with some more effort maybe also for purity. Other quality specifications - like the physical integrity for instance - cannot be subject of a smart contract solution. For example, scratches and nicks in IBC-container caused by a forklift can neither be determined nor logged by IoT devices. Thus, the smart contract still has to be accompanied with a conventional contract respectively the conventional contract remains the “main” contract, just some specific terms are implemented in a smart contract.

4.3. Implementation of IoT solution

4.3.1. Use of Temperature Sensors

There are several ways to measure the temperature of the liquid in the IBC Container. A possible way is to measure the temperature of the liquid directly, so sensors with a probe are a possibility, where the probe makes direct contact with the Isocyanate.

Another way is to measure the temperature indirectly. Temperature sensors could be attached outside of the IBC container, so they are measuring the environment temperature rather than directly the temperature of the Isocyanate. However, the IBC Container could equipped in a way so that the temperature sensor is not directly in contact with the liquid, but that there is only a thin wall and a thermal mass or heat pipe, so also en external temperature sensor could measure the liquid temperature with acceptable accuracy.

There is the possibility to install multiple temperature sensors in order to achieve a redundancy or equalize faulty signals. In addition, the Isocyanate has a certain thermal mass and could have a temperature gradient. The “core” of the 1’000 liters in the container could be still within the temperature limits, while the outer regions could be out of limit due to a recent sudden change of the environment temperature. In the present example, we decided to go for only one sensor per container, as there are 24 containers and the redundancy is guaranteed through its use in assemblage.

Although the binary requirement (hard temperature limit) is not negotiable, there has to be found an agreement on the following problem: there could be a short gust of wind and bring for a short time the temperature of a sensor outside of the allowed range. A strict smart contract would then automatically declare the whole shipment as “destroyed”. This problem is also present, when for instance the Isocyanate is loaded from a temperature controlled room to a temperature controlled truck. In the few minutes of loading, the temperature measuring devices may be exposed to a temperature outside of the required boundaries.

But the decisive fact is, that the Isocyanate has a thermal mass and its temperature does not fall out of range, if the environment temperature does for a short amount of time. To really get rid of this problem, a direct measurement in the liquid would be necessary as well as mixing of the liquid in order to avoid temperature gradients within the 1’000 liters.

To overcome this problem with a pragmatic solution, it was decided to define an acceptable amount of time during which the temperature can be out of range while the temperature of the goods still counts as “within limits”. In this case, this limit was set to 2’700 seconds (45 minutes) and could be subject to experimental testing (will later be an input factor for the smart contract).

The temperature shall be logged with an intervall of 1 minute on a logging device in the truck.

4.3.2. Use of GPS geo tracker

The GPS geo tracker provides the worldwide position of the containers, as far as satellite signal is available. It is possible that a satellite signal is not available during sea shipping or similar.

The availability of the geo tag has several advantages, namely in two areas: first, it allows a cause analysis of where and at which link in the supply chain a possible violation of the parameters (repeatedly) happens. In case of multiple companies involved in the supply chain process, the fallible operator cannot only be identified by the time tag, but also by the geo tag. Second, as the blockchain application requires a connection to the internet anyway, the availability of the geo position of the goods does allow a whole variety of actions controlling the supply chain process. It is possible to make decisions (refusing and rerouting goods) based on the actual geo position, even live planning to reroute goods to another customer or waste disposal facility in case the goods are "devaluated" goods.

As the position of the containers does not change so fast, it is not necessary that the GPS geo tracker signal is logged with the same frequency (1 minute) to the logging device as the temperature signal. It is sufficient if the geo tag comes with a 30 minutes interval for the "summary".

4.3.3. Time Stamp and Time Format

In any case, every data log has to be tagged with a time tag or time stamp. In this case, the "Unix Time Stamp" is the best suitable option. The Unix Time Stamp consists of a (usually) 10-digit integer number which equals the amount of seconds passed since January 1st, 1970 (UTC).

4.3.4. Inner Ring and Outer Ring

As the temperature does not change extremely fast, a sample rate of the temperature of 1 minute is completely acceptable (even lower rates imaginable). The sensors could be bluetooth sensors, which wake up every 60 seconds and send the actual temperature to a logging device on the truck respectively where ever the 24 containers are. This amount of data and this sample rate is easily managable for modern devices, as disk space of several gigabytes does not cost much anymore nowadays. The sensors and the logging device (capable of acquire GPS positions and generating time stamps) represent the "inner ring".

A little different is the situation for the "outer ring", which is everything after the logging device. There is no clue to save the whole amount of data on the blockchain or IPFS. These solutions would be extremely expensive and just generate tremendous amounts of data over the years. Thus, every 30 minutes, the logging device generates a "summary".

Via a 3G module, this summary is send via the internet to the blockchain respectively to IPFS.

As the transport of the Isocyanate happens by truck, it is assumed that at most times, a 3G connection is available.

4.3.5. Distinction of Cases

In order to save the maximum amount of data traffic, it was analyzed which data are necessary in which situation. As long as the agreed conditions are met, a record with the explanatory power of “okay” would be sufficient in principle. There might be some cases imaginable, where both parties do not want only the “interpreted status” (like “okay”), but also the raw data.

Therefore, in case the conditions are met, only the following 5 parameters are sent to the blockchain respectively IPFS every 30 minutes:

- Minimal temperature (only the lowest sensor signal, the other 23 sensors neglected)
- Maximal temperature (only the max. sensor signal, the other 23 sensors neglected)
- Average temperature (of all 24 sensor signals)
- Geo tag
- Time stamp

The parameters refer to the last 30 minutes (since the last “summary”).

In case the parameters were violated in the last 30 minutes, there is a bigger data struct saved on IPFS. It consists in total out of 26 parameters, which are:

- The absolute value of all 24 sensors
- Geo tag
- Time stamp

The setting of the IoT devices described above is shown in the schematic in the figure below.

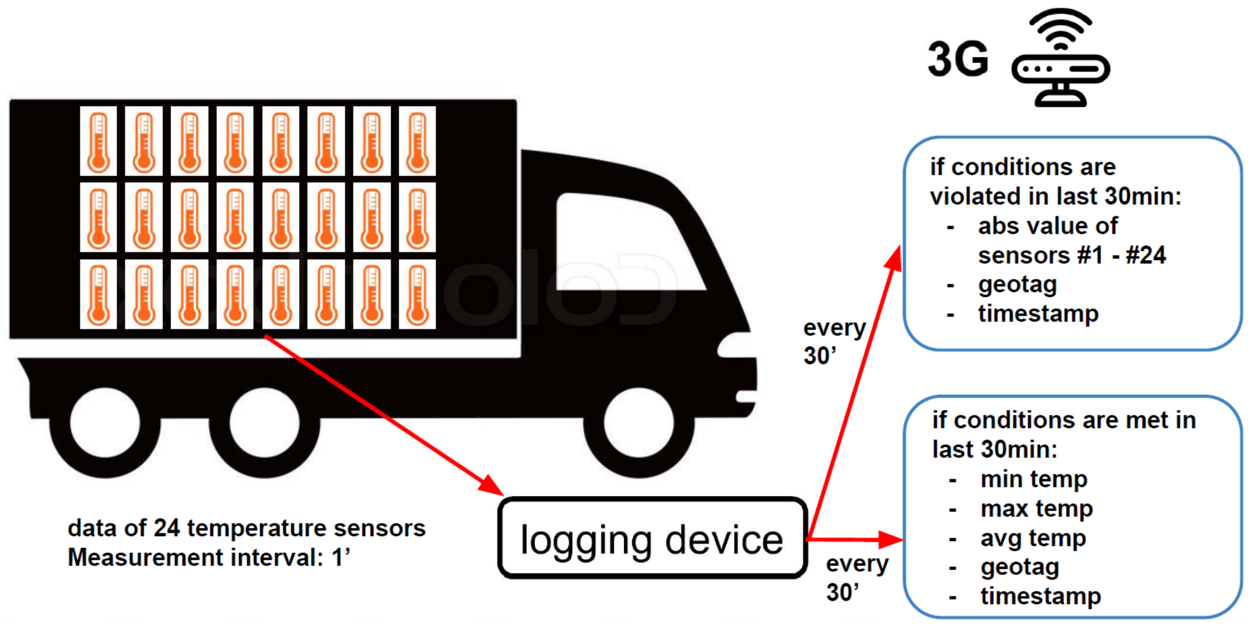

3G
data of 24 temperature sensors
Measurement interval: 1’
logging device
every 30’
every 30’
if conditions are violated in last 30min:
- abs value of sensors #1 - #24
- geotag
- timestamp
if conditions are met in last 30min:
- min temp
- max temp
- avg temp
- geotag
- timestamp

Figure 5: Schematic of the setting of the IoT devices

4.4. Implementation of the Smart Contract

For the understanding of the terms of the smart contract, it is important to recapitulate the following:

- The contract at hand is closed between the seller and the logistic provider (general contractor), and not between the seller and buyer (although there is also a smart contract solution imaginable, see Figure 1).
- The two parameters subject in the smart contract are the temperature and the delivery date.

4.4.1. Initiation of the Smart Contract

When opening a new smart contract between the seller and the logistic provider, the following parameters can be set:

- **Order Number** (no influence on smart contract, just an identification)
- **Creation Time** (the time, at which the smart contract was created in Unix time stamp format; in the present example, it is 1518652801, which equals the 15th of February 2018 at 00:00:01 UTC).

- **Pickup Time** (the time, at which the goods are picked up; this is the starting point for the 48 hours delivery time limit in our example, Unix time stamp format; equals creation time in this example).
- **Delivery Time** (the time, at which the goods have to be shipped respectively the smart contract has to be terminated, Unix time stamp format. In this example, it is 1518825601, which is the 17th of February 2018 at 00:00:01 UTC, exactly 2 days after the pickup time).
- **Shipping Price**
- **Quantity** (24 in present case)
- **Max Temperature** (upper limit of allowed temperature band, in example case 40°C)
- **Min Temperature** (lower limit of allowed temperature band, in example 5°C)
- **Tolerance Time** (time, how long a single signal is allowed to violate the temperature band; in this example, it is set to 2700 seconds which equals 45 minutes. As the data summary extraction is every 30 minutes, this means, that for 1 extraction, a violation is accepted, but when it happens two times in a row, the logistic provider fails the temperature requirement).
- **Measure Interval** (set to 60 seconds, logging device rate for the inner ring)
- **Penalty per day** (this is the penalty of the payment of the general logistic contractor; the penalty is set to 0.1 per day. This means for example, when the goods arrive two days too late, the discount is 20% of the shipping costs.
- **Max Penalty** (set to 0.5; so even if the goods arrive later than 5 days too late, the max. discount for the amount paid for the shipping service is 50%; it could even be agreed, that in that case, the shipping is refused, but this is not subject in the actual example).
- **Blockchain Sync Interval** (equals to the connection between the “inner circle” and the “outer circle”, where the summaries are extracted from the logging device; in this example, the value is set to 30 minutes.
- **Incoterm** (Here, additional predefined conditions can be written in the smart contract, if not already agreed in the conventional contract).

Now, via clicking on the “submit” button, both parties (seller and logistic provider general contractor) can accept the smart contract.

4.4.2. Temperature Parameters Conditions

As the tolerance time is set to 45 minutes and the extraction time (= blockchain sync interval) is 30 minutes, there have to be 2 consecutive extractions containing a temperature value not between 5°C and 40°C, and the delivery condition control is qualified als “failed” and the goods are judged as “destroyed”. In that case, the agreed terms in the conventional contract apply for compensation for losses suffered (not subject of the smart contract anymore).

4.4.3. Delivery Time Conditions

The Pick-up time was defined to be the 15th of February 2018, 00:00:01 UTC. If the goods are delivered before the 17th of February 2018 00:00:01 UTC, there is no penalty. If the goods are delivered between 17th of February 2018 00:00:01 UTC and 18th of February 2018 00:00:01 UTC, then the penalty of 10% is applied, and so for every further day until the 5th day of delay. In practice, the delivery time either has to be confirmed manually or it is captured by a QR code scan or a RFID capture.

4.4.4. Closing of the Smart Contract

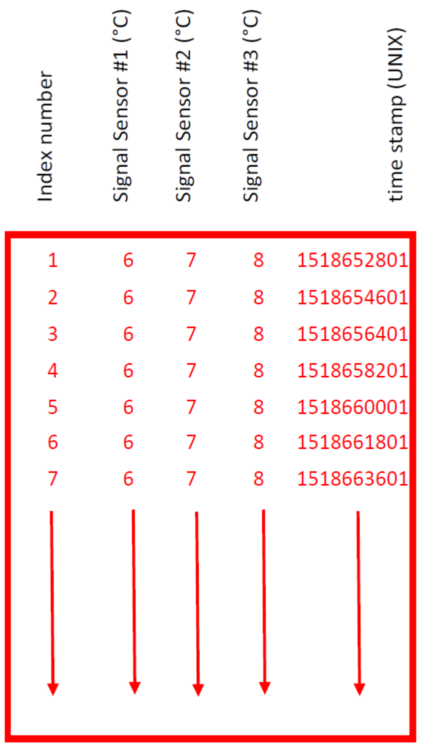
As soon as the delivery is confirmed, the time stamp of the delivery is available. Then, the smart contracts either sends back the funds “reserved” for the logistic company to the delivery initiatior, sends the funds to the logistics company or a combination of the two.

4.5. Implementation of Data for Demo Software

4.5.1. Format of Demo Software Struct

For the demo code, all the parameters and settings were implemented like described above, with one exception: the demo data structs do not contain 24 temperature sensor data, but only the data of 3 sensors.

The data demo structs have the following format:



Index number	Signal Sensor #1 (°C)	Signal Sensor #2 (°C)	Signal Sensor #3 (°C)	time stamp (UNIX)
1	6	7	8	1518652801
2	6	7	8	1518654601
3	6	7	8	1518656401
4	6	7	8	1518658201
5	6	7	8	1518660001
6	6	7	8	1518661801
7	6	7	8	1518663601

Figure 6: Format of demo software struct

4.5.2. Generated Demo Software Structs

There were two demo software structs generated, each one to fail in one of the given two parameter criteria (temperature band and delivery time).

Demo Software Struct 1: temperature range not maintained

- ➢ 100 % of funds returned to delivery initiatior
- ➢ 0% of funds to transportation

Demo Software Struct 2: delivery half a day too late

- ➢ 10 % of funds returned to delivery initiatior
- ➢ 90% of funds to transportation

5. Demo Software

5.1. Architecture

The figure below shows the overall architecture of our demo software. It includes order initiator (left in figure 7), who can start an order, pay deposit and confirm receipt of items, and logistic company (right in figure 7), who can record transportation data and receive payment according to pre-specified conditions. The blockchain in the middle serves as an intermediate for the two parties to ensure loyalty and fairness. Transportation data (like temperature, humidity and timestamp) captured by IoT sensors is stored on the blockchain. Smart contract agreed by both parties specifies transportation conditions and plays an escrow role for payments.

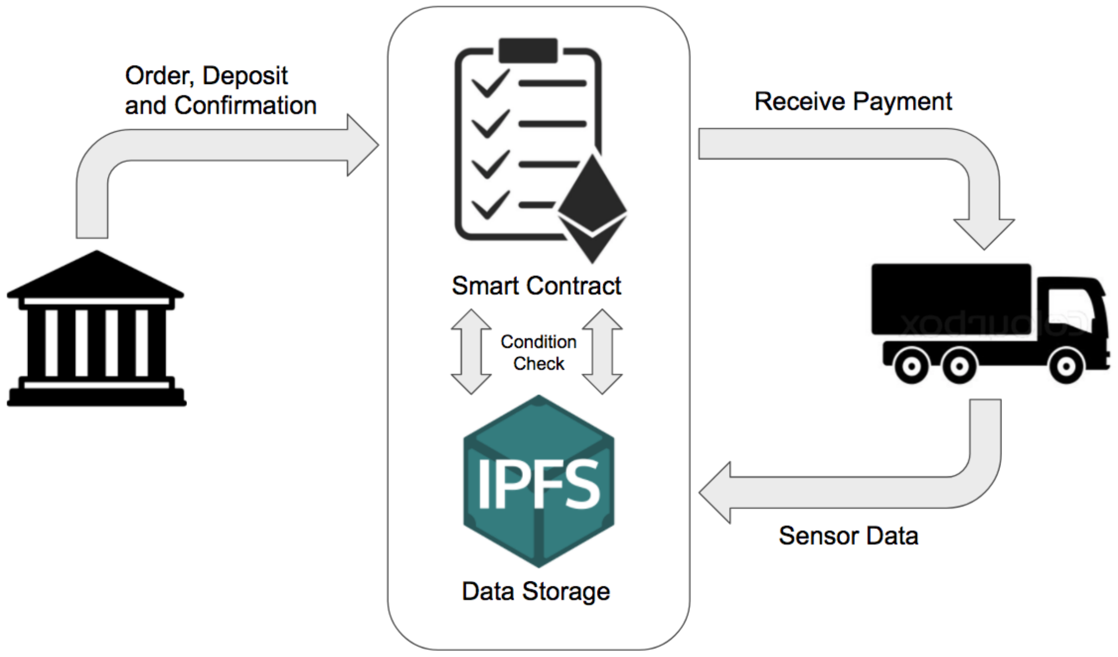


Figure 7: The architecture of our demo software

5.2. GUI

The demo software provides an interface for both parties through GUI. We illustrate design and usage of the GUI through a running example in next section. The GUI consists of four windows:

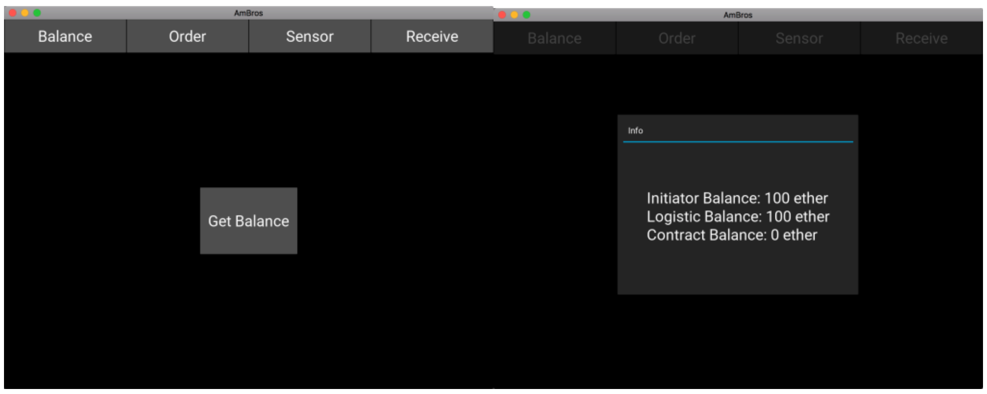
- Balance: balance window is used for querying current balance of order initiator, logistic company and smart contract.

- Order: order initiator can make orders to logistic company through the order window. Information about the shipping items and specification for shipping condition must be provided.
- Sensor: during transportation, critical data like temperature is recorded by IoT sensors and uploaded to the blockchain. We emulate data recording by data sheet files as discussed in previous sections. Our software can upload the data from files to blockchain data services.
- Receive: Once package arrives at destination, order initiator can check whether shipping conditions are met and how much to pay. Then, payment can be confirmed by order initiator and the shipping business is finished.

5.3. Running Example

In this section, we demonstrate the design and use of our demo software and its GUI through a motivating example.

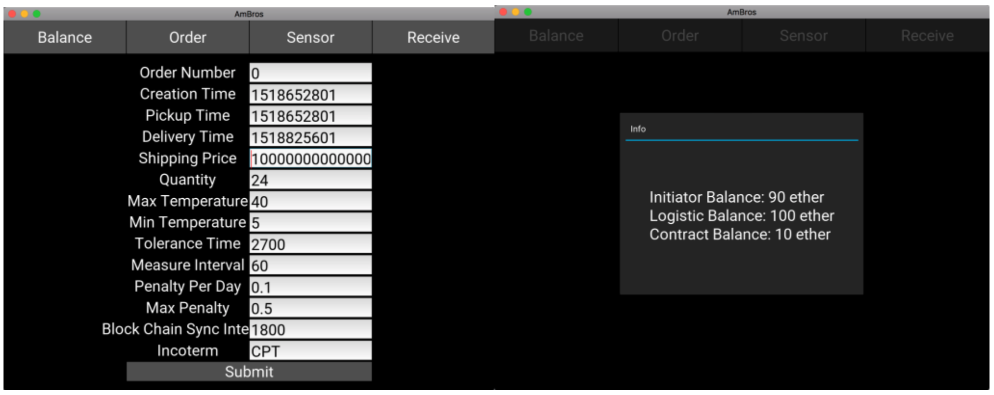
First, we query the account balance of order initiator, logistic company and smart contract. In the balance window shown in the figures below, we can click the 'Get Balance' button and it shows that their balances are 100 ether, 100 ether and 0, respectively.



Initiator Balance: 100 ether
Logistic Balance: 100 ether
Contract Balance: 0 ether

Figure 8: Balance window software

Then, order initiator makes an order with item details and shipping conditions in the order window. The initial shipping price is 10 ether (shown as amount in wei in the figure). After order initiator submits the order, we check the balance again. As shown in the right figure below, the order initiator pays 10 ether to the smart contract.



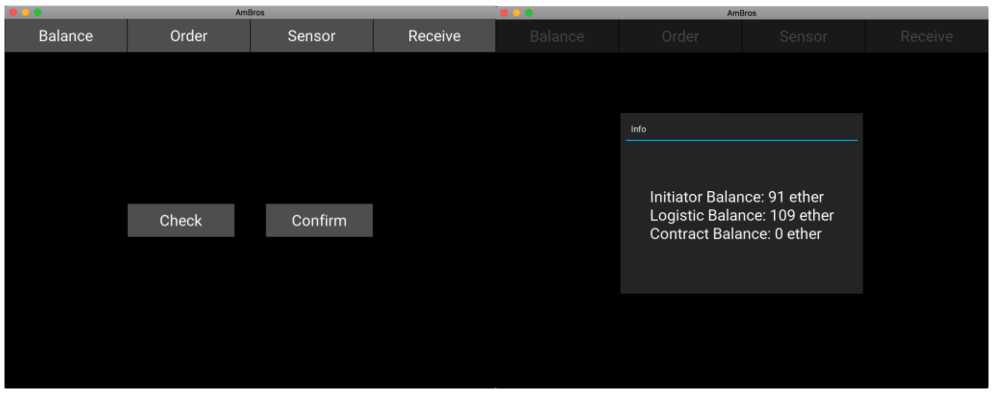
Order Number	0
Creation Time	1518652801
Pickup Time	1518652801
Delivery Time	1518825601
Shipping Price	10000000000000
Quantity	24
Max Temperature	40
Min Temperature	5
Tolerance Time	2700
Measure Interval	60
Penalty Per Day	0.1
Max Penalty	0.5
Block Chain Sync Inte	1800
Incoterm	CPT

Initiator Balance: 90 ether
Logistic Balance: 100 ether
Contract Balance: 10 ether

Figure: Order window

Next, logistic company records IoT sensor data and upload them to blockchain. As described above, we emulate the data recording step with data sheet files. In this example, we select the example 2 datasheet specified in previous sections, where the delivery is half a day late. After we select the file, the data is uploaded to blockchain.

Finally, the item is delivered. The smart contract calculates final shipping fees based on sensor data on blockchain. In the receive window shown in the left figure below, the order initiator can check the amounts by clicking 'Check' button in receive window and confirm payment by clicking 'Confirm' button. Since the delivery is half a day late, a penalty of 1 ether is introduced by the smart contract. As a result, after order initiator confirms payment, 9 ether is paid to logistic company as shipping fee and 1 ether is paid back to order initiator.



Initiator Balance: 91 ether
Logistic Balance: 109 ether
Contract Balance: 0 ether

Figure 9: Receive window

5.4. Implementation Details

We implemented this demo software in Python. The smart contract runs on a local ethereum testnet by testrpc and is accessed using web3.py. The blockchain data storage is implemented with APIs provided by the company Ambrosus. The UI is developed with kivy python library.

6. Discussion

6.1. Disruption Potential

In the industry, especially in the pharma and food industry, the transport, origin or exact list of ingredients of chemicals or food must meet strong conditions (e.g. temperature), which often cannot be verified. Therefore, it is always a dilemma when damaged goods arrive because they did not meet the required conditions, which is difficult to prove when and where the conditions were not met. To solve the dispute companies would often have to go to court, which is very expensive time consuming and might potentially ruin their partnership with their supplier. In such cases a fully automated system which can check when and where the conditions were not met would be desirable and therefore automatically trigger for example discounts or other predefined action upon such events. Therefore, a fully automated system where with the help of IoT sensors the conditions could be surveilled, would increase efficiency and reduce overall litigation cost. Consequently, disputes would be eliminated and monetary transfer is conducted automatically by smart contracts which were negotiated beforehand. Additionally, one has to think about value of the gathered data which could be used for reputation to show the quality of service and could possibly lower some insurance costs. In case of unexpected events during transport, the gathered data could be used for rerouting or discontinue the transport if it becomes clear that the predefined conditions in the smart contract cannot be met.

Another very important aspect is sustainability. To be able to grow a sustainable system, one has to be able to establish trust in this system. When the consumer is willing to buy more expensive goods to make sure that the products come from a sustainable production and sustainable harvesting, but cannot be sure whether the product is exactly what it claims to be. Labels like the “Fairtrade” label for bananas can help, but they often just improve one aspect of the whole supply chain. Also, these labels can easily be misused, since they are often just stickers on a fruit or printed on the package. If one looks at the whole life cycle from the beginning to the end and the whole supply chain of a product, it is easy to see how convenient it would be to have every step of the supply chain recorded on the blockchain. This would make it a trustworthy system in which the consumer can be sure that he or she is buying what he or she intended to buy.

6.2. Open Questions and Challenges

One of the biggest challenges is obviously the fact that smart contracts can only take the predefined and negotiated conditions into account. Any damage conflicts or other problems with the goods that were not recorded by the IoT sensors cannot be solved. The system can only be fully automated and can only give guarantees for the predefined conditions for the measured data. Therefore, it only makes sense to use these systems for transporting goods that are highly sensitive and have strong conditions to meet.

Further, to make it a trustworthy system, it would be desirable to connect the private blockchain to a public blockchain. Unfortunately, there is no possibility to guarantee a permanent connection to the public blockchain and keep everything up-to-date. Another problem is the sheer throughput of data on the blockchain, which can lead to high transaction costs and slow processing time. To tackle this problem one can make use of the IPFS protocol.

Additionally, the question of payment is I tough one, since currently no cryptocurrencies exist which have a stable value in FIAT currency. One possible idea to tackle this problem is for both parties to agree on a private token which is created under certain conditions and then bought by both parties with FIAT. Then the parties can agree on the value of the token and it can be used to trigger fully automated token transactions.

Another issue is to define specific standards for the setting of the IoT devices. In chapter 4.3, there was a debate over how “close” the temperature measurement devices shall be positioned and how long of “not acceptable deviation” does then really mean “failed”, as there could be single disturbance through cold gusts etc. or the short opening of a door.

To avoid that every delivery initiator has to elaborate all those detail for every single type of shipment with every logistics company, the industry is well advised to agree on common standards which then easily can be applied.