Decision Making: Applications in Management and Engineering
Vol. 5, Issue 1, 2022, pp. 1-26.
ISSN: 2560-6018
eISSN: 2620-0104
DOI: https://doi.org/10.31181/dmame0326022022a

SUPPORTING FARMING SMART DOCUMENTATION SYSTEM BY MODULAR BLOCKCHAIN SOLUTIONS

Andi Arniaty Arsyad[1], Irawan Widi Widayat[1] and Mario Köppen[1*]

[1] Departement of Computer Science and Systems Engineering, Graduate School of Creative Informatics, Kyushu Institute of Technology, Japan

Received: 6 January 2022;
Accepted: 25 February 2022;
Available online: 26 February 2022.

Original scientific paper

Abstract: *For more than a decade, various farm-specific models have been developed by collaborating and integrating sensing technologies as a step toward successful data-farm documentation and effective decision-making. However, the stored and gathered data continues to rely on cloud infrastructure or centralized platform control, which is particularly vulnerable to threats such as data tampering, data distortion, confidentiality, and manipulation, which caused the farm product data difficult to trace to its provenance. The objective study in this paper proposes a farm transaction model by demonstrating a flow of farm transaction simulation implicated by Modular Block Chain(MBC) sensing instrument with an array of sensors, controllers, networking hardware, computing equipment, and internal memory functions to enhance data integrity and security farm object. Based on the proposed model, a proof-of-concept experimental system called Encapsulating Block Mesh (EBM) integrates blockchain technology with the specific application case of cocoa production has been implemented. Results have shown that farm objects represented by MBC take turn recording information on the process of generating, transacting, and consuming a farm product and encrypting it into a block was validated and linked in the EBM with the hash of transaction data that connected to each cocoa farm object in a simulation environment. The findings from this study are twofold: the approach has been shown to be feasible and effective, but also capable to be expanded to other stages of the food supply chain, such as manufacturing, supermarkets, food consumption (consumers), and recycling in real-world environments.*

Key words: *Smart Farming, Blockchain, IoT, Smart Documentation, Virtual World.*

* Corresponding author.
E-mail addresses: arsyad.andi-arniaty825@mail.kyutech.jp (A. A. Arsyad), widayat.irawan-widi389@mail.kyutech.jp (I. W. Widayat), mkoeppen@ieee.org (M. Köppen).

1. Introduction

These days, a lot of development of tools and applications helping the farmer to handle the various aspect of farming implicated by sensing technology such as data processing, water management, field monitoring, soil condition monitoring, crop yield analysis, and disease management (Sreekantha, D.K. and Kavya, A.M., 2017). With these smart technologies, farmers can become strategic and efficient in their daily farm-related tasks and responsibilities. Centralizes, manages and optimizes farm production activities and operations, automates the recording and storage of farm data, monitors and analyzes farm activities and consumption, and tracks business expenses and farm budgets (Kutter, T et.al., 2011). However, complex food supply chains characterized by production, distribution, transportation, processing, retail, and food consumption are increasingly exposed to a wide range of risks (Chammem, N et.al., 2018), including contamination, domino effects, resource depletion, difficulty in the following origination, and quality disputes, create this technology ineffective in terms to the security data stored which is automatically vulnerable to distortion, manipulation, and so on. Simultaneously, farming as the vital component of the overall goal agri-food chain causes such complexities to be triggered, somehow guiding back to the production level.

A few reported food safety risks headlines, such as horse DNA (The Telegraph, 2013) were detected in various food products worldwide, which took some time to trace back to where the horse meat came into contact with the other meat products. Other reports address the effect of China's covid-19 crisis and coronavirus shutdowns affecting American farmers (Legal Insurrection, 2020). Farmers across the country have a surplus of produce, milk, eggs and so on that they are dumping, letting rot, or plowing under due to drastically reduced demand, which has also caused global problems in other parts of Michigan, forcing the shutdown of their restaurants, resulting in a domino effect on their profit (G.M.America, 2020). The supply chain is already jeopardized as a result of these challenges. This indicates that even if they are not a link in the pandemic chain, interaction has already occurred in this supply chain. In other words, it isn't easy to trace back any source of the origins by natural influences or different environments to any of the challenges described. Another point of view is that, while modern technology provides enormous agricultural benefits, its adoption and utilization in rural areas are limited. This is due to a lack of education among farmers and the high cost of maintaining this technology.

Therefore, it is essential to establish a farm transaction model that incorporates a sensing instrument to face the challenge of tracking the origin of farm products by integrating unique design security systems with a proper encryption mechanism to ensure the tracking back of product provenance while maintaining data confidentiality within these tools. So that captured event data is traceable and distributed in a secure, comprehensive manner, making it nearly impossible for attackers to infiltrate all supply chain nodes.

In this study, we propose a farm transaction model by demonstrating a flow of farm transaction simulation representing a 'versatile smart instrument' with an array of sensors, controllers, networking hardware, computing equipment, and internal memory functions to enhance data integrity and security farm object. Each farm object is linked in a secure block system, ensuring that farming data is monitored, stored safely, integrated, and is difficult to manipulate due to the encryption provided by a hash value in each object. For this purpose, We specially design a unique blockchain concept in the form of blockchain modularity in which farm objects take turns recording information on the process of generating, transacting, and consuming a farm product into a block. The block contains a record of every transaction ever made and provides a hash value of its contents, including the previous block's hash. Each block is then encapsulated and linked to the previous farm transaction block and keeps the hash of that block. Any modification to any block in that chain will break the chain later on by having an invalid hash value. Based on the proposed model, a proof-of-concept experimental system called Encapsulating Block Mesh (EBM) integrates blockchain technology with the specific application case of cocoa production has been implemented and validated on the simulation environment. Subsequently, the block contents data are later transmitted to the central terminal as the corresponding entry of the events log, which monitors the overall activity of transaction events.

The novelty of this study is developing and implementing sensing instruments in special farming transaction models with the integrity of the special security design using the blockchain concept. Offer farm management for farmers to monitor their farms with a detailed level of security, high accuracy, and can be traced with legality valid information from each farm object in real-time. The block security system is interconnected so that it can be ensured that the farming data collected can be monitored, stored securely, integrated, and difficult to manipulate because it is encapsulated in encrypted blocks in each farm object.

The remaining part of this paper is structured as follows. Section 2 presents reviews of published literature. Section 3, the system architecture, including a brief the blockchain transaction validation model, followed the proposed farm transaction model, the actual simulation operation of the proposed EBM in detail, as well as how the concept of blockchain is suitable in the context of our research. Section 4 contains the result and discussion. Finally, in Section 5, the conclusions are presented.

2. Related Work and Motivation

For over a decade now, sensor and sensing technology has been integrated into the supply chain for the smart farming practices. Citrus fruit production (Lee & Ehsani, 2015), UAVs for vineyards (Candiago et.al., 2015), and using multi-purpose satellite systems to enhance cotton cultivation (Huang & Thomson, 2015) are just a few examples of sensor deployments for specialty crops. In the instance of cow health (Helwatkar et.al., 2014) discovered a number of prevalent disorders in dairy cattle that may be detected using non-invasive, low-cost sensor technology. There are more advanced sensor platforms available, such as camera systems that detect back position (Viazzi et.al., 2014) and ingestible tablets for heart rate assessment (Warren et.al., 2008). Caja et al. (2016) and Rutten et al. (2013) examined the literature in terms of the documented usage of sensors for managing the health of dairy herds in agri-food sectors such as dairy farms. Sensors that monitor arbitrary features of a cow or aggregate sensor data to offer information such as estrus predominant.

On sensor networks, specifically Wireless Sensor Networks (WSNs), have seen widespread use in agriculture (Ojha et.al., 2015)(Abbasi et.al., 2014) and the food business (Wang et.al., 2015)(Garcia et.al., 2009). Application domains include crop management (Juul et.al., 2015), phenotypic assessment (Greenwood et.al., 2014), rustle prevention (Nkwari et.al., 2014), and greenhouse management (Srbinovska et.al., 2015) are only a few examples of application domains. In applications such as irrigation control, wireless sensor and actuator networks (WSANs) are gaining traction (Nikolidakis et.al., 2015) (Chikankar et.al., 2015). Moosense (Sarangi et.al., 2014) is a WSN that uses both ground-based and animal-mounted sensors to control a variety of animal characteristics such as ambient environment factors and nutrition intake (customized food auger and fluid kiosk). Gonzalez et.al. (2014) illustrated the potential of a heterogeneous WSN in providing data in real-time to help in the analysis of animal behavior and allow effective herd management.

Each of these evaluations offers farmers a variety of farming models with integrated technology for targeted, effective decision making and the option to monitor their farms in real-time with an unparalleled degree of detail. However, none expressly address whether the obtained information is securely stored, distributed, and traceable, including models developed to simulate such circumstances. All indicate the effectiveness of sensor and sensing technology usability to detect and monitor the physical, chemical, or biological property quantities and characteristics of various farming products while disregarding the requirement for data privacy, confidentiality, and integrity. Those few studies with model approach and technique used are outlined in Table 1.

Motivated by the above, our basic idea here is to implement a farm-specific model, i.e., a farm transaction model based on "the bucket principle." Since the applied concept gives a visual representation of a bucket-based transaction that occurs from interactions with other farm objects and following processes in the farm food life-cycle from field to consumer implicated by sensing instrument. We believe in an integrative approach that can mediate the actual constraints of farm operation information recorded such as security, durability, integrity, and traceability by improvising the utilization of a one-of-a-kind blockchain plan.

3. System Architecture

This section first introduced the proposed farm transaction model, which encompasses the key aspect implicated in the cycle farm transaction, such as physical assets and objects that we named 'the bucket principle,' complemented by a brief overview of blockchain transaction. A bucket-based transaction was first presented with the “Satyr Farm” farm simulation game operating in the OpenSimulator-driven simulation environment. The system model is then presented with the specific application case of cocoa production, followed by described Encapsulation Block Mesh (EBM) concept and subsequently the farm simulation operation of the proposed EBM described in detail.

3.1 The Bucket Principle

In other blockchain applications, the validity of a transaction is usually a rather straightforward task. Consider bitcoin transactions as shown in Figure 1. Alice wants to transfer an amount of n bitcoins to Bob, where she has a wallet of m bitcoins. So, the only condition here is that while other circumstances of the transaction do not matter: like the true identity of either Alice or Bob, their location, day of time the transaction took place or the weather.

The story is different for such transactions in the farming and general food supply chain environment. We may point out a number of aspects that make such transactions appear different. Assume a simplified model where a farmer brings the apple harvest from some trees to a storage site.

Table 1. Summary of related works with the main objective of the model.

Authors	Main objective of the model	Technique used
Lee & Ehsani (2015)	Defines sensing systems, including a yield mapping system that uses fruit recognition disease detection sensors that are carried by ground- and aerial-based platforms in the citrus fruit production.	NIR and Raman spectroscopy

Authors	Main objective of the model	Technique used
Candiago et.al (2015)	Demonstrates high-resolution Unmanned Aerial Vehicles (UAV)-based remote sensing and photogrammetric techniques applied in the agriculture framework to collect multispectral images.	UAV data and photogrammetric techniques
Huang & Thomson (2015)	Discusses using multi-purpose satellite systems to enhance cotton cultivation, including growth monitoring, insect Control, yield Prediction.	Remote sensing
Helwatkar et.al. (2014)	Identifies specific diseases common in dairy animals and the development of the next generation of health monitoring systems which can be identified through non-invasive, low-cost sensor technology.	Non-invasive sensor
Viazzi et.al. (2014)	Evaluate a two-dimensional and three-dimensional camera system to measure dairy cows' back posture automatically.	Three-dimensional camera
Warren et.al. (2008)	Design a pill that can remain in an animal's reticulum and electrocardiographic techniques to ascertain and automate heart rate determination.	Electrocardiographic pill
Caja et al. (2016)	Reviews the literature in terms of the documented usage of sensors for managing the health of dairy herds in agri-food sectors such as dairy farms that are expected to produce dramatic changes in traditional dairy farming systems.	Sensor devices
Rutten et al. (2013)	Provides an overview of the published sensor systems for dairy health management, including techniques that measure something about the cow activity, interpretations that summarize changes in the sensor data (e.g., increase in movement), to produce information about the cow's status (e.g., estrus), integration of information where sensor information is supplemented with other information (e.g., economic information) to produce advice (e.g., whether to inseminate a cow or not), and the farmer makes a decision, or the sensor system makes the decision autonomously.	Sensor systems
Ojha et.al. (2015)	Evaluate the network and node architectures of WSNs, the associated factors, and classification according to different applications, including the various available wireless sensor nodes and the different communication techniques followed by these nodes.	Wireless Sensor Networks (WSNs)
Abbasi et.al. (2014)	Evaluate the need for wireless sensors in Agriculture, WSN technology, and their applications in different aspects of agriculture and existing system frameworks in the agriculture domain.	Wireless Sensor Networks (WSNs)
Wang et.al. (2015)	Designs and implements reconfigurable, low data rate, cost-efficient, and low-power WSN nodes and developed a real-time monitoring system for perishable food supply chain management, including the environmental parameters the state of motion of perishable food.	Wireless Sensor Networks (WSNs)
Garcia et.al. (2009)		WSN and RFID
Juul et.al. (2015)		Wireless Sensor Networks (WSNs)
Greenwood et.al. (2014)		Wireless Sensor Networks (WSNs)

Authors	Main objective of the model	Technique used
Nkwari et.al. (2014)	Evaluate the standards and the numerous applications that utilize Wireless Sensor Technologies (WST) in the agriculture and food industry and classify them into appropriate categories.	Wireless sensor node with a GPS
Srbinovska et.al. (2015)	Introduces a system comprised of a WSN and a user interface that presents the measurements to the user in an accessible way. The system helps farmers avoid losses and achieve a more consistent quality of crops by monitoring environmental variables such as temperature and humidity during long storage periods.	Wireless Sensor Networks (WSNs)
Nikolidakis et.al. (2015)	Discusses issues underlying the need for new and novel phenotyping methods and the establishment of the WSN and pasture intake research platforms to predict feed intake and feed efficiency of individual grazing animals.	Wireless sensor and actuator networks (WSANs)
Chikankar et.al. (2015)	Determines the effectiveness of using the continuous-time Markov process to determine if a cow is being stolen or not and determine anomalies in behavior that could indicate the presence of the thieves. A wireless sensor node with GPS was designed to sense the position and speed of a cow.	Wireless Sensor Networks (WSNs)
Sarangi et.al. (2014)	Presents the development of a WSN application for precision agriculture, which is deployed in a pepper vegetable greenhouse to achieve scientific cultivation and lower management costs from environmental monitoring, including the temperature, humidity, and illumination.	Wireless Sensor Networks (WSNs)
Gonzalez et.al. (2014)	Presents an integrated architecture based on WSNs, for automated irrigation management to achieve effective and prompt irrigation of parcels with excellent energy efficiency due to the utilization of a novel routing protocol ECHERP.	Wireless Sensor Networks (WSNs)
	Describes an effective irrigation management system for the container-grown crops that effectively utilize water resources for agriculture and crop growth monitoring using GSM and ZigBee technology.	
	Presents Moosense, a WSN that uses ground-based and animal-mounted sensors to control various animal characteristics such as ambient environment factors and nutrition intake (customized food auger and fluid kiosk).	
	Presents potential of a heterogeneous WSN in providing data in real-time to help in the analysis of animal behavior and allow effective herd management with sufficient frequency to increase understanding of animal biology and improve productivity.	

- The transaction involves a physical object, subject to physical and subsequent features, weight, volume, freshness etc.

- Identity of the agent’s matter. For example, the agents need to be spatially close to the “wallets” (that appear to be physical assets as well, trees, plants, soil, container etc.).
- The transactions are inevitably linked to other transactions, foregoing that transaction, accompanying it, or, with some delay “unlocking” it.
- The transaction can be sensed and recorded, e.g., by a video camera or subject witness.
- The transaction has quality means on their own. In our example, the apples have been carried by hand, or within a large container lowering apple quality by inflicting damages. In a similar way, the transaction includes aging.
- Generally, such transactions need energy to be maintained, due to the physical nature of its constituents. At the same time, it needs human cognition and intervention to become subject of documentation and recording.
- A classical transaction e.g., of a bitcoin transfer can only be valid or invalid. Farming transactions appear valid to some degree as the outcome of the transaction at the goal site can vary even having same starting point, while there can be unknown, lost, or manipulated circumstances at the origin.
- The transactions expire in some sense. At one point in time, all products made from an apple have been eaten by a living being. There is no primary need to log transactions forever if there is nothing to do or to conclude from the information in a ledger anymore.
- Farming doesn't happen on a terminal, so the interfacing among physical "not connected" objects need to be organized.

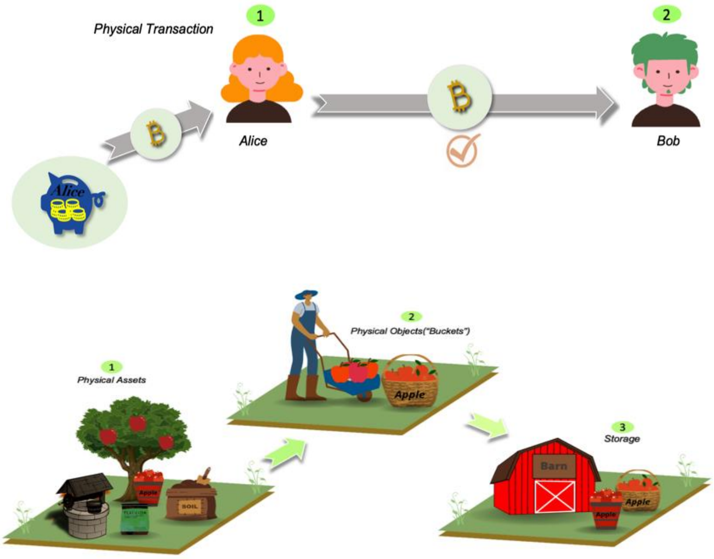


Figure 1. Classical Alice-Bob bitcoin transaction vs farm transaction.

Already by number, those aspects may put into question if blockchain is a suitable concept here at all, compared to the surveying of a number of virtual wallets. Of course, it can be argued that this just increases the number of data that has to be stored within a block and nothing else. This can indeed account for a number of aspects, but not all. Most at all, the mutual influence of transactions, and the locality aspect.

In that regard, and to overcome those problems, we propose a transaction model called “bucket principle” that reflects the transactions in a farming (and subsequent food supply chain) in a way that makes it accessible to the integration in a specially designed blockchain. The principle can be stated as:

Bucket Principle: *All transactions in farming and the subsequent food supply chain can be modeled by the means of transport of an item S from site A to site B within abstract bucket B. The bucket alone can maintain all documentation tasks while logging related events: creation or filling of a bucket, picking up or dropping of a bucket, replenishing, emptying, or consumption of a bucket's content.*

To reconsider above example the apple harvest will be filled into a basket, for example. Now, this can be the proverbial wicker basket, and it is brought by hand to a nearby storage site, keeping it cool until the storage site is filled up to load a truck. Many other such farm activities would refer to real-world buckets. But it is also an abstraction: pruning a tree, the farmer will have to bring tools for pruning to the tree. For watering, one might to have to bring a hose nearby, or dig a ditch to guide water flow it where the bucket then is the channel leading to the tree.

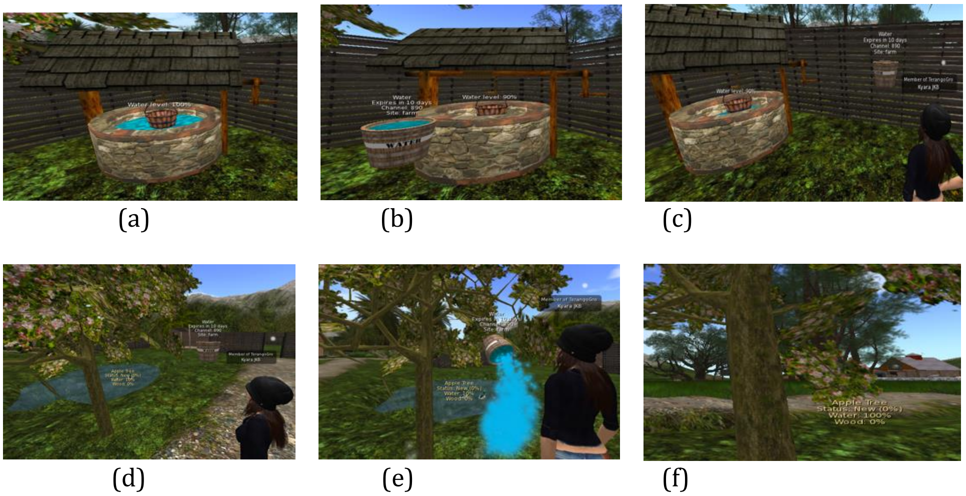
The Bucket Principle has several implications. First of all, it can be seen as a “smart tool” operating in a communication environment like edge computing or an IoT infrastructure, which can have sensors, controllers, networking hardware, computing facilities, and internal memory. We agree that the focus in “smart farming" so far was on the sensor equipment, and not much engineering efforts have been put into “Tools IoT” so far. But the argumentation so far gives a good hint that such developments are overdue, and obviously feasible. On the bottom line, even carrying items by hand can become an abstract bucket transaction by using a specially designed data glove.

A bucket has content, that is at the same time a symbolic name and a physical entity. While the latter is obvious, the former can cause documentation problems. Receiving site B might call the content of the bucket differently than the originating site A according to documentation and registration needs. It implies the need for a bucket (in conjunction with its filling) to have a unique id, thus it needs a digital identity.

It also has to be added that the action commonly referred to as transport, e.g., by a cargo ship, isn't such a bucket here. That transport has to be modeled as a *remote storage*: there is a bucket that brought the content of a storage that became subject of being moved to another location. Arriving there, buckets will be filled with content of the same storage, just at a different location.

However, while not applying to transport in classical meaning, it extends to many other subsequent transactions of farm items: cooking, placing items in a supermarket shelf or market booth, consuming like eating, after all, yes. While here is not the place to consider the universality of the bucket principle, as we are primarily interested in its implications for the design of a monitoring blockchain.

The Bucket Principle, up to our best knowledge, was initially introduced along the “Satyr Farm” farm simulation game (Satyr Farm, 2020) running in the OpenSimulator driven simulation environments (OpenSimulator, 2020) and that was introduced in the OpenSimulator driven hypergrid around 2018. The simulation provides a visual cue of a bucket-based transaction that is the result of interactions with farm objects. For example, as shown in Figure 2, to water a tree the avatar mediating user control in the simulation interacts with a well that renders a 3D model of a water bucket and that starts following the avatar as it moves to the tree to water. There, the water is replenished to the tree, and the bucket itself dismissed. The whole Satyr Farm follows this method in all transactions. In subsequent developments, we could demonstrate that the same principle allows extension of the farm to the various stages of the food supply chain, including factory, supermarket, and food consumption and recycling.



(a) (b) (c)

(d) (e) (f)

Figure 2. A bucket-based transaction in the used simulation environment.

3.2 Encapsulating Block Mesh (EBM)

The Bucket Principle explanation aforementioned gives the base for a smart farming documentation system that integrates blockchain technology. We will point out three major practical requirements:

- Concurrency: A farming transaction does not happen in isolation but under the concurrent ongoing of other transactions that are either a condition for the proper fulfilment of current transaction, or an influencing factor, up to issues of resource sharing and replenishment.
- Locality: There is no reasonable method to link farm items over long distances, especially among different, maybe even competing farms. The documentation should be done close to the related sites and only based on Near Field Communication in a technical way.
- Sparsity: We can't assume to have blockchain recording all over a farm. This is accompanied by the circumstance that several factors affecting trust and quality of farm products are not single-point events but spread over some area and time. Hardware for recording and documenting can be damaged, stolen, or being tampered with. So, the means of validating a blockchain has to account for gaps. The degree of trust into the validation has to be related by the depth to which the events proceeding the current transaction can be safely traced back.

Taking those three requirements into account, we propose the following concept of "Encapsulated Block Mesh" extending the cocoa production blockchain concept.

3.2.1. Cocoa Production Transaction Model

We selected the cocoa production model, specifically locally cocoa processing, as cocoa processing has a number and unique stages and aspects, quite a straightforward chain, and a well-defined structure. The steps involved in primary cocoa processing are also the corresponding variables in the chain, including planting, harvesting, fermentation, drying, and bagging/storage (Guda & Gadhe, 2017) (Saltini, et.al., 2013).

In the system model as shown in Figure 3, we consider that each cocoa farm object, including well, tree, fermenter, drier, and bagging in the storage, has a built-in sensing device called MBC, where the buckets, as the result of finishing a processing stage, processes events such as event logging and documentation hence enabling any farm item to receive messages and links to other farm items. All data from the buckets are later transmitted to the central terminal/cloud storage as the corresponding entry of the events log, which monitors the overall activity of the buckets.

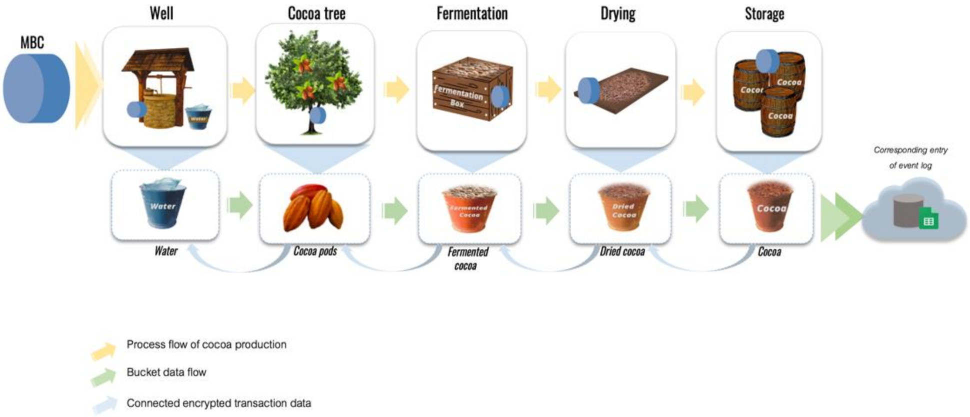


Figure 3. Cocoa production transaction.

3.2.2. Modular Blockchain (MBC)

As previously stated, a bucket-based transaction has various implications, such as a "smart tool" operating in a communication environment such as IoT infrastructure and including sensors, controllers, networking hardware, and internal memory. It is referred to as MBC. In this scenario, MBC serves as a smart sensor integrated on a farm object to record, collect, and store essential farm product information which encapsulating it in a strong cryptographic proof for data authenticity and integrity with a block-generated contract within.

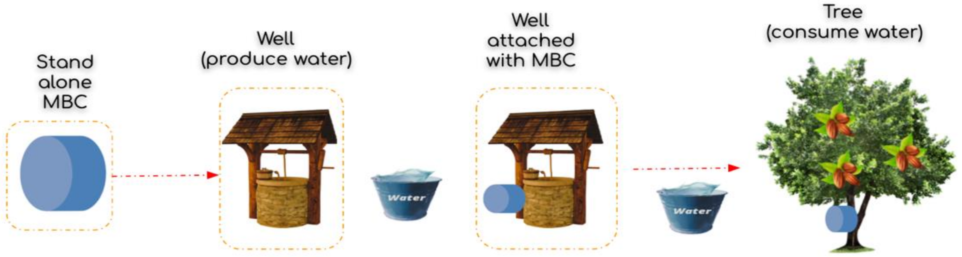


Figure 4. A block-generated contract within MBC.

Figure 4 depicts a contract produced by a block within MBC. Initially, a stand-alone MBC was used, which was later integrated into the farm object (well). Furthermore, the MBC-connected well will generate a bucket of water, which will carry a record log of events, which will subsequently be transferred and consumed to the next farm object, i.e., a tree. There are two sorts of blocks created based on the events that occur.

- A bucket of water generated by a well has information stored in blocks. As described in Algorithm 1, the previous block, hash 1, is the information generated in each transaction. An entry "NOPE" indicates that no specific information is stored here for the case of produce events. The block event was unrelated to consuming a bucket, referred by hash 2, and hash 3 is the information generated each time the farm object produces a bucket. Next, create a new block, and it is classified as a produced event.
- In the delivery of the bucket of water, the bucket will carry the amount of water and confirmation of the content transfer. As described in Algorithm 2, the hash of the very first block is generated in each transaction as hash 1. Hash 2 is content transfer originating from the bucket of produce event, and hash 3 is information generated each time the farm object consumes a bucket. The farm object, particularly the tree, will absorb the water, triggering MBC to form a new block. It is categorized as a consumed event.

Next, the pseudo-codes of blocks that are generated based on the events that transpired.

Algorithm 1 MBC Produce Event

```
function mbcProduces(Block):
    Get last previous block, $B_{last} = Block(last)$
    $Hash_1 = Hash(B_{last})$
    $Hash_2 = “NOPE”$
    $Hash_3 = Hash(eventinformation)$
    $B_{new} = [Hash_1, Hash_2, Hash_3]$
    return $B_{new}$
```

Algorithm 2 MBC Consume Event

```
function mbcConsume(Block):
    Get last previous block, $B_{last} = Block(last)$
    $Hash_1 = Hash(B_{last})$
    $Hash_2 = Hash(mbcProduce())$
    $Hash_3 = Hash(eventinformation)$
    $B_{new} = [Hash_1, Hash_2, Hash_3]$
    return $B_{new}$
```

Since our model is primarily concerned with documenting all farm operations in nearby areas, we consider MBC compatible with the NFC tag and solely interact with the NFC. NFC serves as a gateway, allowing access to data stored in the cloud. The MBC mesh's purpose is to secure the event log recordings and does not require internet connectivity. Hence, there is a requirement to employ NFC as a peer to transfer information and store data on tags to prevent MBC from being hacked. So, if the end-user wants to access sensor data from outside, it has to ask NFC to get the required data.

3.3 Implementation Principles

To put the EBM into action, we simulate local cocoa processing activities, including the primary cocoa bean processing chain, especially harvesting, fermentation, drying, and storage. This activity process represents MBC which collects, stores, manages key product information of each farm product. The following illustrates the MBC are linked in the EBM in Figure 5.

- Harvesting. This process starts with flowers and ends with cocoa beans growing in pods. The cocoa tree consumes water during its growing period and eventually generates cocoa pods as the produce event.
- Fermentation. The fermentation process should begin after pod shattering. The box fermentation collects cocoa beans per pod and finishes with the beans being equally fermented. This process is categorized as consuming and producing events.
- Drying. After fermentation, the fermented cocoa seeds must be dried, and the fermented seeds must be spread on trays exposed to sunlight. The drying plate receives a bucket of fermented cocoa, and the process ends with a bucket of dried cocoa. This process is classified as consuming and producing events.
- Storage. The dried beans are packed into sacks for storage in a warehouse. The storage is obtained in a bucket of dried cocoa and then transferred to various locations. The final stage of cocoa processing is categorized as a consume event.

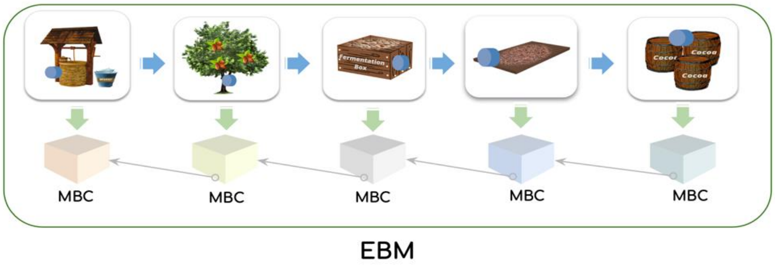


Figure 5. Block Mesh.

3.4 Farm Simulation

In this section, we have implemented the EBM for cocoa production on simulation platform as depicted in Figure 6 followed by a detailed explanation. An open simulator environment was employed to illustrate and demonstrate the farm transaction by Encapsulating Block Mesh (EBM). A cloud-hosted instance of the OpenSimulator server (version 0.9.1). This OpenSimulator provides an appropriate research environment by supporting several frameworks such as server-client architecture, grid architecture, avatar-based control, concurrency, and scripting support. It became feasible to develop an experimental framework for conducting simulations that can be evaluated, analyzed, and enhanced through multi-institutional collaboration inside the so-called hypergrid connecting the various server simulations globally (Delp et.al., 2007).

Experimental Environment

Cocoa processing involves several vital aspects before it is transformed into a cocoa product. In this experiment, we confine the modeling of the cocoa production process to watering, harvesting, fermenting, drying, and bagging. The following scenarios are considered.

- Watering the cocoa plant. At this point, the well serves as a water supplier as one of the vital components to ensure healthy growth and cocoa yield. This stage, which represents the first activity of the cocoa plant growth process, is classified as a produce event in which the well generates water consumed by the cocoa tree.
- Harvesting. Which begins with flowering and finishes with cocoa beans growing in pods. This stage is divided into two parts: consume events and produce events, in which the cocoa tree consumes water during its growth phase and ends up by generating cocoa pods.
- Fermentation. After pod breakage, fermentation should take place. This step is also divided into consume and produce events, in which the box fermentation receives cocoa beans per pod and finishes with the beans being evenly fermented.
- Drying. The fermented cocoa seeds must be dried after fermentation, and the fermented seeds must be spread on trays under sun exposure before they are shipped to the storage in the warehouse. This step is divided between consume and produce events. The drying plate receives a bucket of fermented cocoa, and the process culminates in producing a bucket of dried cocoa.
- Bagging/Storage. The dried beans are now packed into barrels/sacks for storage in a warehouse. This step is classified as a consume event in which the storage is obtained in a bucket of dried cocoa and then transferred to various locations.

EBM-specific parameters and settings will be described in Table 2 as follow.

Table 2. Application-specific parameters and settings of simulation.

Process of Event	**Description**
1. Produce Event	Farm object generates a bucket.
• Code 96	The code used by the farm object to send to MBC as a create event.
• UUID's bucket	Initial information in the form of the identity of the farm bucket received by MBC.
• Hash of former block	The hash of the very first block is generated in each transaction.
• Hash of "NOPE"	NOPE signifies the block event was not related to consuming a bucket.
• Hash of create event	The hash that is generated each time the farm object produces a bucket.
2. Consume Event	Farm object consumes a bucket.
• Bucket of content transfer	Content transfer originating from the bucket of produce event.
• Code 97	The code used by the farm object to send to MBC as a consume event.
• Hash of former block	The hash of the very first block is generated in each transaction.
• Hash of origin block	The origin identifier of previous blocks.
• Hash of consume event	The hash that is generated each time the farm object consumes a bucket

The following EBM for cocoa processing will be described in detail as shown in Figure 6.

- After connecting to MBC, each farm item, including well, cocoa tree, fermentation, drying, and storage, is assigned a unique channel number and block count. The

MBC of each farm object contains a block component structure that includes a former block hash, hash block origin, and event hash.

- The well will generate blocks, which will subsequently be carried and absorbed by the cocoa tree. The previous hash block, produced from a bucket of water, is then used as the identifier or original hash block for the following event object.
- The cocoa tree, which is the next consumption event, will receive several buckets of water and be consumed. Each tree has transaction records for tree watering, one of which results from the cocoa harvesting stage, specifically cocoa pods. The block contains hash data from water activities, while the other block contains hash data from cocoa pods. All blocks have an identical block component format, consisting of a hash of the previous block of transactions, the origin of the block hash as an identifier of the origin of the previous block from Well's MBC, and a new hash event for consuming events already consumed by the cocoa tree. In event fermentation, the hash of the previous block of the resulting cocoa pod becomes the identifier or block hash of origin for the next event object.
- At the fermentation stage, as in the following bucket, the cocoa pod provides information on the transfer of event data content from the tree, transmitted to the fermentation stage. Each fermentation box comprises a transaction records block in which the cacao pod is poured into the fermentation box while another block of fermented cocoa is generated. The former block hash of each generated fermented cocoa becomes a further identity for the object in the drying event.
- Similar to the process in the fermentation stage, in the drying stage, each drier plate comprises a record of transactions where a fermented cocoa bucket as a result of a fermentation process is poured into the drier while other blocks are generated dried cocoa. The former block hash of each generated dried cocoa becomes the identifier for the object in the final storage of the cocoa processing journey.
- Storage as the cocoa processing series' final generates has the same block component structure as the previous stages of cocoa processing. The transaction records in each block contain event log data from a bucket of dried cocoa that will be utilized as a secondary identification for the event's next step.

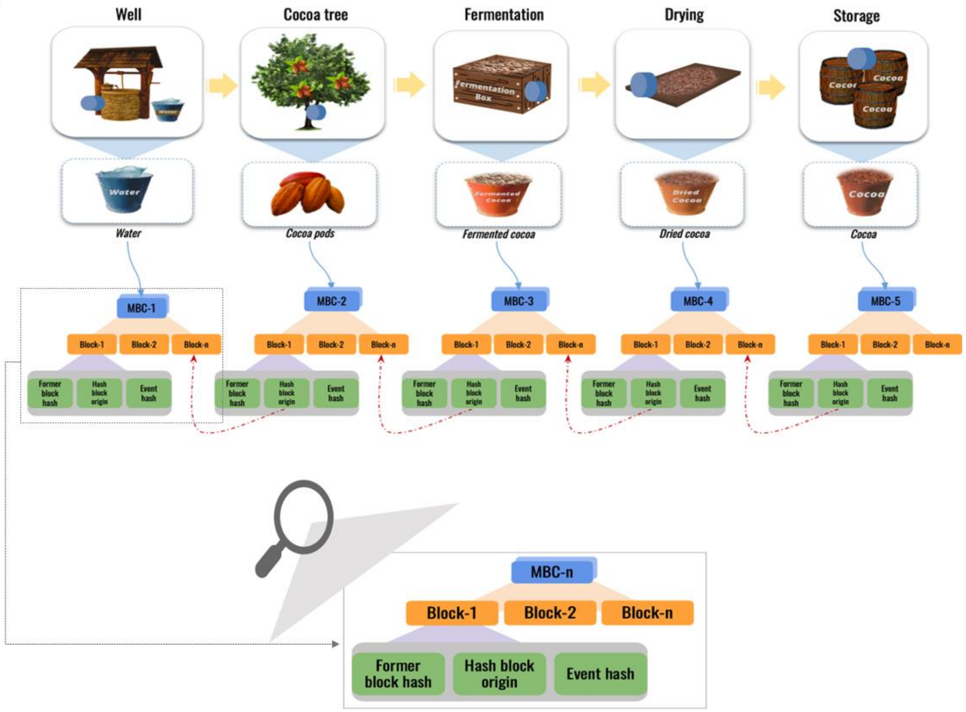


Figure 6. EBM for Cocoa Production.

3.5 MBC Event Log

In this section, we constructed gateway communication between MBC to the cloud storage, specifically transferring the farming transaction data recorded on MBC until it is stored in cloud storage. In this case, we are using Google cloud.

Each MBC will record and keep every transaction that occurs. Therefore, we construct a storage method that takes advantage of cloud storage, in this case, Google sheets as storage media. Every transaction performed by MBC will be broadcast to a certain communication channel specified in the script. To listen to every transaction that occurs on each MBC, we improve the communication gateway in the form of a script placed in a farming simulation. This script's function is to record and report every transaction from each MBC's farming simulation to the Google cloud. Some parameters such as *key_id* and *HTTP_Requests* function are needed to construct the communication between farming simulation and Google Spreadsheet.

However, like other cloud storage media, this is a plain external log that is mostly unprotected. Hence the event log protection derives from its connection to the reliably EBM.

4. Result and Discussion

4.1 Result

We now have a few records log of transactions from MBC's. Since our transaction model runs in parallel from one transaction to the next, thus necessitating authentication and validation of the farm product data as the block updates each time and proceeds to the next block. We then investigate the certain MBC of the block on the farm object.

4.1.1. Validation

Figure 7 illustrates the detailed block of MBC, with each block identified and tracked through the operational procedure of the proposed EBM.

The following is an overview of the transaction validation procedure for cocoa processing.

- A bucket of water is produced by the well that initiates the cocoa production process. A bucket of water with the bucket's information content distributes it to MBC. The contents are the bucket's UUID, the token, and the event. From here, the transaction data is hashed into blocks, and MBC generates a block and then proceeds to send back the new block to the water bucket, and it becomes the first block of the bucket. The validation process occurs if the transaction data is traced by tracing the identity of the hash. Here, between the existing hash block origin and the previous block hash of MBC. If the hash values fully comply, the transaction is considered valid.
- Assume that the cocoa tree requires transaction data from a bucket of water at an early stage. The bucket of water then transports a certain amount of water and other content transfer information to the cocoa tree for consumption. The cocoa tree then provides information comprising hash origin and consumption events to MBC, following which transaction data is hashed into blocks until MBC creates new blocks. If both hash values match, the transaction is considered valid. In such situation, the validation procedure is carried out by tracing the identity of the hash, which matches the previous hash block of the bucket of water and the origin of the hash block in the cocoa tree.
- Regarding transaction validation during the fermentation stage, drying to storage, as aforementioned, it appears to follow the same workflow in terms of producing blocks in each transaction. Produce and consume events are the two sorts of blocks that are created. Validation is accomplished during transaction block tracing by finding and matching the identity of the hash between the previous hash block of the cocoa tree and the hash block origin at the fermentation level. If the hash values of the two transactions are matched, the transaction is considered valid. Something similar will occur at the drying and storage levels.

4.1.2. Investigate the MBC's

The investigate step is taken to monitor the block on the harvested cocoa, transporting it to the fermentation box to obtain fermented cocoa, which is subsequently transported to the drier plate to get dried cocoa. The distribution process finishes with warehouse storage.

- Investigation 1:

Inquire about the origin of the dried cocoa in the warehouse storage, in this case, randomly selected storage rack at block 2. (Table A.16)

MBC Block 2:
former block hash: `be12382f98fd9240ed80fe1a322175a9`
hash block at origin: `d9e1e110f5cf22ed97f4a05ccd15a932`
event hash: `fe7ccaa75cebbf540c5c9517731d4f7e`

As can be seen, the hash of the origin block is:
d9e1e110f5cf22ed97f4a05ccd15a932.

Now double-check with the drier:

After calculating the hash, the result of the retrieve hash block origin is d9e1e110f5cf22ed97f4a05ccd15a932 which corresponds to the second entry in block 2 of drier plate two. It signifies that the dried cocoa in the warehouse storage originated the dried plate two and allowed the UUID of the MBC of the drier plate to be verified.

- Investigation 2:

After double-checking the origins of the dried cocoa, the next step is to inquire about the origins of the fermented cocoa in drier plate two block 1. (Table A.13)

MBC Block 1:
former block hash: fff7a973d18eac54302e41ce70530816
hash block at origin: 3616c124ebd63803f088017de4b85c55
event hash: f5551d1423cc272844822b938b020d74

As can be seen, the hash of the origin block is:
3616c124ebd63803f088017de4b85c55.

Double-check with the fermentation box:

After calculating the hash, the result of the retrieve hash block origin is 3616c124ebd63803f088017de4b85c55 which corresponds to the first entry in the block 1 of drier plate two. It indicates that the fermented cocoa in the drier plate originated in fermentation box four block 2 and allowed the UUID of the MBC of the fermentation box to be verified.

- Investigation 3

Inquire where the cocoa pod originated from in fermentation box four block 1. (Table A.11)

MBC Block 1:
former block hash: fff7a973d18eac54302e41ce70530816
hash block at origin: 017a6b68a82992770678410 85d79b803
event hash: 31a73da207677efd5f1603b9ef1e7d39

As can be seen, the hash of the origin block is:
017a6b68a8299277067841085d79b803.

Double-check with the cocoa tree:

After calculating the hash, the result of the retrieve hash block origin is 017a6b68a8299277067841085d79b803 same as the first entry in the block 1 of fermentation box four. It indicates that the cocoa pod in the fermentation box four originated in harvest cocoa of tree 1 and allowed the UUID of the MBC of the cocoa tree to be verified.

- Investigation 4:

Since the well has 30 blocks while we got the last block traced from the tree 1, the tree has been watered in between. Thus, we can read the "former block hash" directly. Now we investigate the former block hash of the bucket of water that was watered the cocoa tree 1. (Table A.2)

MBC Block 1:

former block hash: fff7a973d18eac54302e41ce70530816
hash block at origin: f77bc30541130cb9d10e6afc4ebd9ccf
event hash: 327e5bcd1e522ff90d3da725c9d6454c

As can be seen, the hash of the origin block is:
f77bc30541130cb9d10e6afc4ebd9ccf.

Double-check with the well:

As a result of the calculating hash, hash block origin is
f77bc30541130cb9d10e6afc4ebd9ccf,
which is the same as the former block hash of the well. The UUID of the MBC of the well can be verified.

To put it another way, we have securely identified one source of the water that watered the tree, finally leading to the examined cocoa beans in warehouse storage.

4.1.3. Corresponding Entry of Event Log

The cocoa processing encapsulating block mesh is used as the foundation for capturing transaction data of cocoa processing in order to ensure data integration, safety, and traceability of the overall activity of transaction events. The cocoa processing encapsulating block mesh method's final output can subsequently be saved on the Google cloud storage.
Figure 8 shows a screenshot of the output page of the corresponding event of the data log.

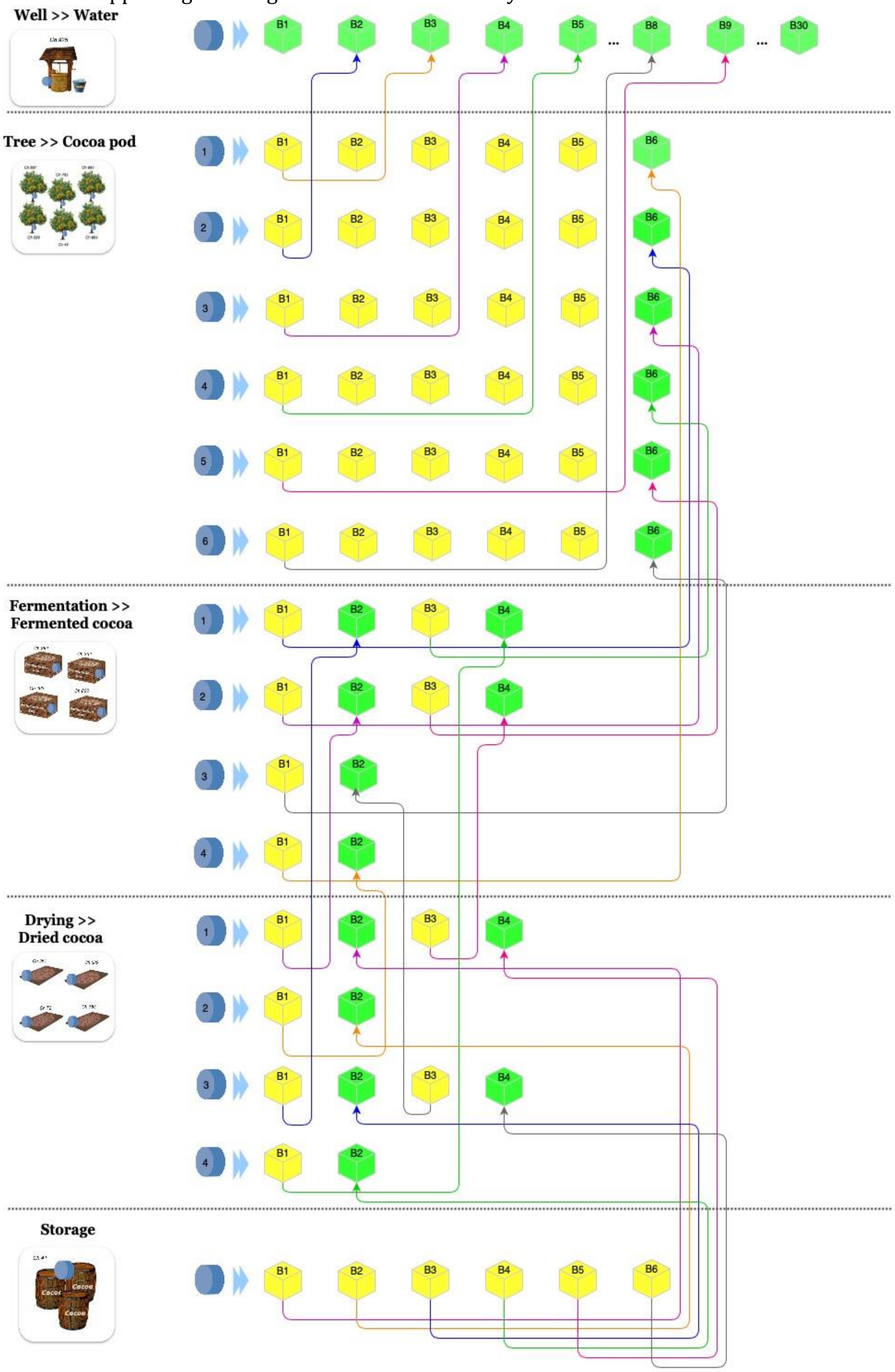
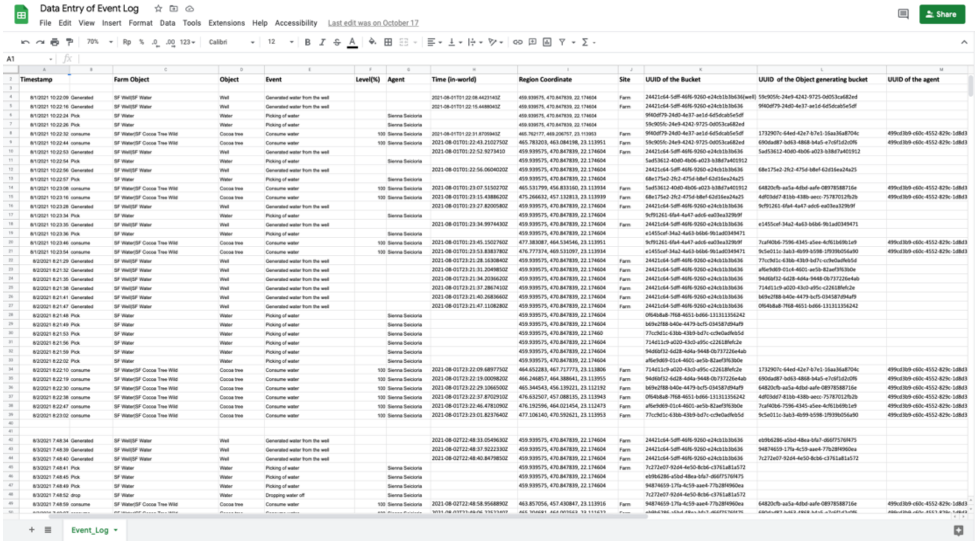


Figure 7. Detailed traced block of the MBC



	A	B	C	D	E	F	G	H	I	J	K	L	M
2	Timestamp		Farm Object	Object	Event	Level(%)	Agent	Time (in-world)	Region Coordinate	Site	UUID of the Bucket	UUID of the Object generating bucket	UUID of the agent
3													
4	8/1/2021 10:22:09	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:22:08.4423140Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636(well)	59c905fc-24e9-4242-9725-0d053ca682ed	
5	8/1/2021 10:22:16	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:22:15.4488040Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	9f40df79-24d0-4e37-ae1d-6d5dcab5e5df	
6	8/1/2021 10:22:24	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		9f40df79-24d0-4e37-ae1d-6d5dcab5e5df		
7	8/1/2021 10:22:26	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		59c905fc-24e9-4242-9725-0d053ca682ed		
8	8/1/2021 10:22:32	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:22:31.8705940Z	465.762177, 469.206757, 23.113953	Farm	9f40df79-24d0-4e37-ae1d-6d5dcab5e5df	1732907c-64ed-42e7-b7e1-16aa36a8704c	499cd3b9-c60c-4552-829c-1d8d3
9	8/1/2021 10:22:44	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:22:43.2102750Z	465.783203, 463.084198, 23.113951	Farm	59c905fc-24e9-4242-9725-0d053ca682ed	690dad87-bd63-4868-b4a5-e7c6f1d2c0f6	499cd3b9-c60c-4552-829c-1d8d3
10	8/1/2021 10:22:53	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:22:52.9273410	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	5ad53612-40d0-4b06-a023-b38d7a401912	
11	8/1/2021 10:22:54	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		5ad53612-40d0-4b06-a023-b38d7a401912		
12	8/1/2021 10:22:56	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:22:56.0604020Z	459.939575, 470.847839, 22.174604		24421c64-5dff-46f6-9260-e24cb1b3b636	68e175e2-2fc2-475d-b8ef-62d16ea24a25	
13	8/1/2021 10:22:57	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		68e175e2-2fc2-475d-b8ef-62d16ea24a25		
14	8/1/2021 10:23:08	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:23:07.5150270Z	465.531799, 456.833160, 23.113934	Farm	5ad53612-40d0-4b06-a023-b38d7a401912	64820cfb-aa5a-4dbd-aafe-08978588716e	499cd3b9-c60c-4552-829c-1d8d3
15	8/1/2021 10:23:16	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:23:15.4388620Z	475.266632, 457.132813, 23.113939	Farm	68e175e2-2fc2-475d-b8ef-62d16ea24a25	4df03dd7-81bb-438b-aecc-75787012fb2b	499cd3b9-c60c-4552-829c-1d8d3
16	8/1/2021 10:23:28	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:23:27.8200580Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	9cf91261-6fa4-4a47-adc6-ea03ea329b9f	
17	8/1/2021 10:23:34	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		9cf91261-6fa4-4a47-adc6-ea03ea329b9f		
18	8/1/2021 10:23:35	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T01:23:34.9974430Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	e1455cef-34a2-4a63-b6b6-9b1ad0349471	
19	8/1/2021 10:23:36	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		e1455cef-34a2-4a63-b6b6-9b1ad0349471		
20	8/1/2021 10:23:46	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:23:45.1502760Z	477.383087, 464.534546, 23.113951		9cf91261-6fa4-4a47-adc6-ea03ea329b9f	7caf40b6-7596-4345-a5ee-4cf61b69b1e9	499cd3b9-c60c-4552-829c-1d8d3
21	8/1/2021 10:23:54	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T01:23:53.8383780Z	476.777374, 469.531097, 23.113934		e1455cef-34a2-4a63-b6b6-9b1ad0349471	9c5e011c-3ab3-4b99-b598-1f939b056a90	499cd3b9-c60c-4552-829c-1d8d3
22	8/2/2021 8:21:29	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:28.1630840Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	77cc9d1c-63bb-43b9-bd7c-cc9e0adfeb5d	
23	8/2/2021 8:21:32	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:31.2049850Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	af6e9d69-01c4-4601-ae5b-82aef3f63b0e	
24	8/2/2021 8:21:35	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:34.2036620Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	94d6bf32-6d28-4d4a-9448-0b737226e4ab	
25	8/2/2021 8:21:38	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:37.2867410Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	714d11c9-a020-43c0-a95c-c22618fefc2e	
26	8/2/2021 8:21:41	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:40.2683660Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	b69e2f88-b40e-4479-bcf5-034587d94af9	
27	8/2/2021 8:21:47	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-01T23:21:47.1108280Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	0f64b8a8-7f68-4651-bd66-131311356242	
28	8/2/2021 8:21:48	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		0f64b8a8-7f68-4651-bd66-131311356242		
29	8/2/2021 8:21:49	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		b69e2f88-b40e-4479-bcf5-034587d94af9		
30	8/2/2021 8:21:53	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.17460		77cc9d1c-63bb-43b9-bd7c-cc9e0adfeb5d		
31	8/2/2021 8:21:56	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		714d11c9-a020-43c0-a95c-c22618fefc2e		
32	8/2/2021 8:21:59	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		94d6bf32-6d28-4d4a-9448-0b737226e4ab		
33	8/2/2021 8:22:02	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		af6e9d69-01c4-4601-ae5b-82aef3f63b0e		
34	8/2/2021 8:22:10	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:22:09.6897750Z	464.652283, 467.717773, 23.113806	Farm	714d11c9-a020-43c0-a95c-c22618fefc2e	1732907c-64ed-42e7-b7e1-16aa36a8704c	499cd3b9-c60c-4552-829c-1d8d3
35	8/2/2021 8:22:19	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:22:19.0009820Z	466.246857, 464.388641, 23.113955	Farm	94d6bf32-6d28-4d4a-9448-0b737226e4ab	690dad87-bd63-4868-b4a5-e7c6f1d2c0f6	499cd3b9-c60c-4552-829c-1d8d3
36	8/2/2021 8:22:30	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:22:29.1066500Z	465.344543, 456.139221, 23.112192	Farm	b69e2f88-b40e-4479-bcf5-034587d94af9	64820cfb-aa5a-4dbd-aafe-08978588716e	499cd3b9-c60c-4552-829c-1d8d3
37	8/2/2021 8:22:38	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:22:37.8702910Z	476.632507, 457.088135, 23.113943	Farm	0f64b8a8-7f68-4651-bd66-131311356242	4df03dd7-81bb-438b-aecc-75787012fb2b	499cd3b9-c60c-4552-829c-1d8d3
38	8/2/2021 8:22:47	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:22:46.4781090Z	476.192596, 464.021454, 23.112473	Farm	af6e9d69-01c4-4601-ae5b-82aef3f63b0e	7caf40b6-7596-4345-a5ee-4cf61b69b1e9	499cd3b9-c60c-4552-829c-1d8d3
39	8/2/2021 8:23:02	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-01T23:23:01.8237640Z	477.106140, 470.592621, 23.113953	Farm	77cc9d1c-63bb-43b9-bd7c-cc9e0adfeb5d	9c5e011c-3ab3-4b99-b598-1f939b056a90	499cd3b9-c60c-4552-829c-1d8d3
40													
41													
42	8/3/2021 7:48:34	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-02T22:48:33.0549630Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	eb9b6286-a5bd-48ea-bfa7-d66f7576f475	
43	8/3/2021 7:48:39	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-02T22:48:37.9222330Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	94874659-17fa-4c59-aae4-77b28f4960ea	
44	8/3/2021 7:48:40	Generated	SF Well|SF Water	Well	Generated water from the well			2021-08-02T22:48:40.8479850Z	459.939575, 470.847839, 22.174604	Farm	24421c64-5dff-46f6-9260-e24cb1b3b636	7c272e07-92d4-4e50-8cb6-c3761a81a572	
45	8/3/2021 7:48:41	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604	Farm	7c272e07-92d4-4e50-8cb6-c3761a81a572		
46	8/3/2021 7:48:45	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		eb9b6286-a5bd-48ea-bfa7-d66f7576f475		
47	8/3/2021 7:48:49	Pick	SF Water	Water	Picking of water		Sienna Seicioria		459.939575, 470.847839, 22.174604		94874659-17fa-4c59-aae4-77b28f4960ea		
48	8/3/2021 7:48:52	drop	SF Water	Water	Dropping water off		Sienna Seicioria				7c272e07-92d4-4e50-8cb6-c3761a81a572		
49	8/3/2021 7:48:59	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	2021-08-02T22:48:58.9568890Z	463.857056, 457.430847, 23.113916	Farm	94874659-17fa-4c59-aae4-77b28f4960ea	64820cfb-aa5a-4dbd-aafe-08978588716e	499cd3b9-c60c-4552-829c-1d8d3
50	[illegible]	consume	SF Water|SF Cocoa Tree Wild	Cocoa tree	Consume water	100	Sienna Seicioria	[illegible]	[illegible]	Farm	[illegible]	[illegible]	[illegible]

Figure 8. Corresponding entry of event log

4.2 Discussion

Our proposed EBM has been evaluated in terms of its secure data documentation. In particular, the protection of records data in the event log and the degree of trust of transaction validation have been analyzed by encapsulating the blocks for all farm entities involved in the process.

Our farm model works in parallel from one transaction to the next as a condition for effectively fulfilling ongoing circumstances and transactions. It demonstrates that it is valid to utilize the value of one entity to integrate another entity at a different goal site even if it has the same starting point. Hence the sustainability, resilience of relatedness, and reliability can be proven of the whole product tracing while avoiding manipulation of data-related decision-making.

Considering the possibility that MBC may be damaged, stolen, or tampered with, it may be assumed that there is no guarantee that the blockchain can record throughout a farm. However, authenticating blockchain related to the depth to which the events preceding the current transaction can be reliably tracked back is of great concern here for the trustworthiness of this method.

In other words, the hash value match and relatedness blocks in each event where an event that continues the current transaction occurs and wherever its area is a benchmark for the legitimacy of a transaction in the block.

After simulating EBM, the interfacing among physical objects unconnected has been constructed and organized. It is evident that the documentation of farming transaction records is more structured, complete, secured, and protected, and each farm product is identified, integrated, and verified. It implies that the proposed blockchain approach EBM corresponds to the conditions of a farming transaction.

5. Conclusion

We have proposed the Encapsulating Block Mesh (EBM) for cocoa production by integrating a unique designed blockchain and applying the principle of a bucket-based transaction implicated by MBC's sensing instruments.

The transaction documentation model is recorded at each level of cocoa processing and will be connected to events that occur on other farm objects chained together in blocks using a strong cryptographic method. First and foremost, the overall practical system requirements have been defined. Then, we introduced a protocol that allows each cocoa farm stage to validate and authenticate product farm transaction data through chained hash values in the blocks. A remarkable feature of the proposed EBM is protecting the recorded data in the event log to eliminate data tampering and distortion, allowing data to be monitored safely and transparently. The claim procedure validation and MBC analysis demonstrated the validity of our EBM.

From the finding of this study, we may refer to a realistic portrayal of a flow of farm transaction model in which farm objects take turns recording information on the process of producing, transacting, and consuming a farm product into a block. Each block is then encapsulated, linked, and verified with the previous farm transaction block. This remarkable simulation model is appropriate for implementing agricultural system documentation in the real-world environment. The simulation approach is also expected to address farming documentation issues at several levels of the food supply chain, including factory, supermarket, and food consumption. Farm management may use this simulation to manage or develop a documentation system employing a specifically built security of sensing instrument.

There are some open problems which are attractive to be explored further. The first relates to the method used, which is still constrained in the simulation environment, and the data distribution system, which is still at the farm level. So, for future works related to EBM, it will be highly fascinating to examine how the same principle allows agriculture to be expanded to other stages of the food supply chain, such as manufacturing, supermarkets, food consumption (consumers), and recycling in real-world environments.

Author Contributions: Author Contributions: Conceptualization, A.A.A, I.W.W, and M.K.; methodology, M.K.; software, M.K. and I.W.W.; validation, A.A.A and M.K.; formal analysis, A.A.A and M.K.; investigation, A.A.A and M.K.; resources, A.A.A. and M.K.; data curation, A.A.A. and M.K.; writing—original draft preparation, A.A.A; writing-review and editing, M.K.; visualization, A.A.A.; supervision, M.K.; All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors of this article would like to thank Kyushu Institute of Technology for their financial and educational support.

Conflicts of Interest*:* The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Table A.1: Well

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	NOPE	e57dd29e50f7b4d4676fcbd360a2a3f8
2	2b686a7c8601b901ab8dfd8a9d255ac8	NOPE	657d125c2693b2718350bd556b493ffb
3	f77bc30541130cb9d10e6afc4ebd9ccf	NOPE	6823b54bfff5969f432f9787de991fab
4	e5e130b88e093499fb8f6aaa300400b6	NOPE	46be480f59381b4b16c571f4e0e8a64b
5	4c760f7a1b0ca3542fe5d2aba8ad017e	NOPE	872604e1f8d8ffa837468202c27fbd53
6	127621e1abf18c792c6c1f566fba243f	NOPE	529813506deec4e77493f8651d04ed13
7	b45be60147a7cc2daa34cef71942bd6d	NOPE	9c9a63fb5699170175e9f19015e81a04
8	ab6b314704fca1ec8ec2eeba7946d225	NOPE	28886af72c1ffafb52e36dcf83da9aa3
9	f9f14141fd090b38ce86cee8cd10208e	NOPE	23134c7962420edd64b9000c6237b20b
10	6336806085e3b1b398b9d795a78ac28a	NOPE	e025b3ea1c595c599e3e9542755fbf30
11	c304a81c184818778e12e2a465743572	NOPE	3c6316bb026e20f4b497ebcd186c6f1c
12	2a418e361721c0c35722dc3f44d97fd5	NOPE	18dac09d1ce818f3e80bb939c312c94e
13	21ca390e3a5260596028af93927dbf6f	NOPE	ceb55a4f9cf6f36a4d6533ab3af2f344
14	758e14ecd188acaa352ee73fb565dd33	NOPE	3caa7e7eedebbe619f7516c88232e08d
15	be1c438246d6471521a9fdf4b799c3ea	NOPE	5463688b1f6f766cd0335b49c950b27f
16	7538340e74dcc686bcbe33c5b0a1e6ec	NOPE	2b19ec927e2076d03b51806987453316
17	f49444fe520d3aa73a2a3849caf36232	NOPE	33a0c93dcd5ec8585384bec6c1b7ed01
18	c60b5ea0d579c1aeba7312393d61b8f9	NOPE	6f6c9b7fec611acf13ae1bed3f0e10a7
19	64b09997c23c5651a761a90d5e6a3273	NOPE	ca848ccd11bc9ddb90a3a6b82979e595
20	88e875c13fc12794f273e657729238c4	NOPE	3f4ce43619d24e580c251b7389f9d8dc
21	0c632c4845b349fa993f5878a3575737	NOPE	2f0f870ff0f2826c6e8fe572db69f490
22	22cf3686af45b3e5a6093f040e178b01	NOPE	847968257d0adb1184188e5d286ebda3
23	998a8b241021d888f03f09383878688c	NOPE	62a8e5a40acd2862a4340ba5596d0701
24	54840d45afffa4332dd35a080b5746d5	NOPE	5efbbcd965a092fc43d05644849c427e
25	8559095dbb33dd22fb15675d000023a1	NOPE	24b2d8beaced5aecebadbad1f393077a
26	98b010bd5c2641257a8f06c443e7d1cd	NOPE	e9937b640838f082970b67b05dc094e3
27	63d8ab9a53527ff864f7951d4c2ffc5e	NOPE	66d91d3bba59deb5a021c18b80ca7923
28	330a7177f55f2cbe10e58bdc4720a855	NOPE	2005017c54b4092c4335cff78abcfe27
29	408519e6139b07849f2585b2c29b3421	NOPE	279afb0ecb2bf9223d94e23b60172d5b
30	2db1e2cc196f3b78d6d0bc6a8ba02678	NOPE	38000344f4d52878b1d4396cc92f7807

Table A.2: Tree 1

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	f77bc30541130cb9d10e6afc4ebd9ccf	e57dd29e50f7b4d4676fcbd360a2a3f8
2	dd2a481807268e16e510fd1dc587ce20	c304a81c184818778e12e2a465743572	657d125c2693b2718350bd556b493ffb
3	3e55384c6d35f3d864ba3713c8100a15	7538340e74dcc686bcbe33c5b0a1e6ec	6823b54bfff5969f432f9787de991fab
4	f2aca1105308dfa11a898473997699ee	998a8b241021d888f03f09383878688c	46be480f59381b4b16c571f4e0e8a64b
5	15c54c6970ede71b5959429a4b11c648	330a7177f55f2cbe10e58bdc4720a855	872604e1f8d8ffa837468202c27fbd53
6	ff682f6161312e1117ff16422dc46dca	NOPE	529813506deec4e77493f8651d04ed13

Table A.3: Tree 2

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	2b686a7c8601b901ab8dfd8a9d255ac8	deb41d87667b21468f04b5acb79fa085
2	31636d37495e8277c13c95148de7dc93	6336806085e3b1b398b9d795a78ac28a	3e78a147bb315dc706af478f6025fe1e
3	6af327fca1139f5f2073b8ad599e33fa	758e14ecd188acaa352ee73fb565dd33	90a171f92058f05a92d0c93ae41c5477
4	254828ec0945efa3fa6959554ad39d8c	54840d45afffa4332dd35a080b5746d5	9c62cfa31101717ddb20edf09e7e38e3
5	899e5b9f4034bac0502e53e13226c32f	63d8ab9a53527ff864f7951d4c2ffc5e	fe25d67eb432f707874cf689bd7d4151
6	38a8e1d31f7df1715b54176d8af01189	NOPE	ad6b11736be1b88503a7c1d5f230ba24

Table A.4: Tree 3

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	e5e130b88e093499fb8f6aaa300400b6	91507f9d2feec89e4347f8ef4e10c439
2	ab638335c76fde17260bf6775b03f3fd	2a418e361721c0c35722dc3f44d97fd5	2054e6caeb0f20aada1fb8e8e2611e56
3	6da1e0f9b74cd56c7e21b00e803f2ce3	be1c438246d6471521a9fdf4b799c3ea	1b61ed771e63a3c04f9d9477403e74a4
4	b17a82d1231ff9a3be8330e82358e94a	8559095dbb33dd22fb15675d000023a1	c0e31da9fee418c18a739c57753d3f00
5	def5decbbdc59498e886390c6e1520be	98b010bd5c2641257a8f06c443e7d1cd	f444853e0ac54ef077e5034b18b7069d
6	5a6b0916feb3b5edcd89da71516c047d	NOPE	4f2367931d670fe0cfeaf11403882e25

Table A.5: Tree 4

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	4c760f7a1b0ca3542fe5d2aba8ad017e	a1f915bd02b4055a1ed6e9cccbad809f
2	7c33a136fee6f09a4bf4978bd9b5e2f2	21ca390e3a5260596028af93927dbf6f	166c09a6f0bfa0d8f897090a54d42dac
3	2f5522a8e7313839f87d12fc0d24df94	c60b5ea0d579c1aeba7312393d61b8f9	cc790b20420e7634da462be40bd3dc10
4	192fc2038e5db537bdba7aa44a3263cd	88e875c13fc12794f273e657729238c4	2430e43fb02616071bb7a4114a2c5f1b
5	9f5e77abc5d3a3a04385ee03cd429da5	8f25e89ca0b897e1e099af7c25270214	582cf172fa340fa4af335caf4ace54bc

6	33ecc21f27310dbc28013d8699013a38	NOPE	456b456115b5e7ef06f61e8d8f78f7f6

Table A.6: Tree 5

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	127621e1abf18c792c6c1f566fba243f	285dda67e2a5a44c5a25a029f174ffc5
2	b1845125e31771b19cd3ffa99681d2c3	f9f14141fd090b38ce86cee8cd10208e	cc8a3008f4b185c9ce004f723a12f360
3	efab4637d548cdfd70b8bd13bf2f9afc	f49444fe520d3aa73a2a3849caf36232	e475c880dcc561805d764c9ed9ce5416
4	f2455f916dd1ff690259ce913b559e6c	0c632c4845b349fa993f5878a3575737	51e0e1133246d68e266668032c5cd724
5	f55a38001f7a682903c91eededcfa5c1	408519e6139b07849f2585b2c29b3421	f6967fb451bc928d2f3d7f99003410d5
6	4ccdb0cf56cd364cf7d36f078df75e64	NOPE	9b55bbaffbd3bdc1d69c17ccf3b8bfb4

Table A.7: Tree 6

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	b45be60147a7cc2daa34cef71942bd6d	764a40463057bd5278365150e12d348f
2	1cc89f7df6b8ec667320359a78ab5d28	ab6b314704fca1ec8ec2eeba7946d225	9d2016c662ff5d31b8e428080835fd16
3	ede95a11abb3fe5d0971772f326b458a	64b09997c23c5651a761a90d5e6a3273	b9967b72fdd5ac39f84049831cdb57fb
4	74bbe8e39abd392d2e16c3b19cf3bf89	22cf3686af45b3e5a6093f040e178b01	3001b9c6ecf6e0def4a871591ba055d9
5	1c5b07bbd20c3fd8501cdbbee048a2ea	2db1e2cc196f3b78d6d0bc6a8ba02678	f0eb3fb96d8786882a7fa5319a351672
6	de4445c42470314ff6aaf4ff6eca5905	NOPE	9b28d36a31d6e0204912267338f6a986

Table A.8: Fermentation Box 1

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	8cd261c40ebb9cefaae17fa3795a1435	3c047473a8ec1b43e25589da561c4fa4
2	341f7c950afc7fb316f3372da690d597	NOPE	b7907fbf5bbc22f0e4f0a82a8731bc95
3	85807726a37a81e239354c085dc58911	210f9fdcf44974c6b8118b856265b72c	3b9999e23c9a189c52c4b40f36b247d6
4	10691e511c294ab745d4d2d898b0bb9a	NOPE	26d60839f9d28bb3889109bf60b4ac3b

Table A.9: Fermentation Box 2

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	bae871823b83c1f96f8946db58f79e26	58a572742e9594aa74a87b595a3bda7f
2	a859101fbe8c23fcdc9e4e5e65b282f4	NOPE	857e15fd48c1a236737b24bd5ecfddca
3	2629b590e584815f46c1515c25aaea4a	3c6652bdb2402277f2624f0b6146e377	b1f500fc24300657d057a0038548cb08
4	f9c04d83be42e64782df33f196a37159	NOPE	2822d2b574bfbcc6a4eeddf7e994abff

Table A.10: Fermentation Box 3

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	028ba03293f3a90bc88e25e356e766c8	604f99b6cb01c2149bfdc0331dd3c9ed
2	1d5a18141d76c51bc63a2d3433ed16f3	NOPE	7bf950c7903e0b5bbc3f4b455a609a07

Table A.11: Fermentation Box 4

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	017a6b68a8299277067841085d79b803	31a73da207677efd5f1603b9ef1e7d39
2	b466b91076cfc90f39f9e88905141f84	NOPE	3ab5ce6471ea7bcd39154d434054a138

Table A.12: Drier plate 1

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	2629b590e584815f46c1515c25aaea4a	cd829eeb4a05265ed7c4e5bae10bb79f
2	432fed641dab4733ba24bf81e9665023	NOPE	e7d0dd071eca2c95df9235167ae9b3f2
3	8e37264acd37c4ad2f19a28c7ce32ef7	9ff389b611e36a118a135c876a1ec5b6	8fe6762da16a69f2bc4241e18ddec6ce
4	a8a4be40a5339d596c32172f0d9091c8	NOPE	ea9a2ff0bb42cd6d3f6aad3e414902ba

Table A.13: Drier plate 2

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	3616c124ebd63803f088017de4b85c55	f5551d1423cc272844822b938b020d74
2	f20d259f742a16387f98a587344c94f6	NOPE	9efdbf2ce2df9d88ab457d9267d46371

Table A.14: Drier plate 3

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	85807726a37a81e239354c085dc58911	5b824e46784dbaf44e5f16b9e130dbcd
2	08d7037538bf163c48e6a5b1b0234e1a	NOPE	2cd56eeae9afeba963a9690b0f78d601
3	54e7154d726af7b521ce456085f93477	bd3c7c82e1d2324ffc4d120a1594d9d5	0a1ca99f12a25f270c747d22fe40f1c2
4	d65b6b98ee4b8297ed48fd018c1f50bc	NOPE	3bbd9af9881033ac963452790c56472e

Table A.15: Drier plate 4

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	1c95778108c10289c60a5867ca48e37d	997c0c644449bfc37fa32e9afa9f2d5d
2	50e1ffae45052a9206023e6b0dacd2c3	NOPE	57e9a39524d427a2c56f4931ef789063

Table A.16: Storage

Block	Former block hash	Hash block origin	Event hash
1	fff7a973d18eac54302e41ce70530816	8e37264acd37c4ad2f19a28c7ce32ef7	1b29eb80901752ba79e10a98501491c8
2	be12382f98fd9240ed80fe1a322175a9	d9e1e110f5cf22ed97f4a05ccd15a932	fe7ccaa75cebbf540c5c9517731d4f7e
3	ec2fb067490be6e3bb004d0528c4b1bf	54e7154d726af7b521ce456085f93477	9aeeb41a3d8d9e26f65271c3bfba6894
4	95301626934335f5e76aa2d10f715744	f935e4a5a03e75007688326cc2241264	5c380d674e523499c94d31610cd7831e
5	d73fabb43416561382cbe55a5db95d8f	603a189b5a8dd9c8400de137c19e3968	26ca7014e41a97bc894376c9ef63e75a
6	951957d29fbf48930f3a481e0bef1365	7035e45e5e31c51d1183049f6dadfd38	23609fd5eb455a4a2b67480b76787bc8

References

Abbasi, A. Z., Islam, N., Shaikh, Z.A. (2014). A review of wireless sensors and networks' applications in agriculture, Computer Standards & Interfaces, 36(2), 263–270.

Caja, G., Castro-Costa, A., & Knight, C.H. (2016). Engineering to support wellbeing of dairy animals, Journal of Dairy Research, 83(2), 136–147.

Candiago, S., Remondino, F., De Giglio, M., Dubbini, M., & Gattelli, M. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from uav images, Remote Sensing, 7(4), 4026–4047.

Chammem, N., Issaoui, M., De Almeida, A.I.D. & Delgado, A.M. (2018). Food crises and food safety incidents in European Union, United States, and Maghreb Area: current risk communication strategies and new approaches. Journal of AOAC International, 101(4), 923-938.

Chikankar, P. B., Mehetre, D., & Das, S. (2015). An automatic irrigation system using zigbee in wireless sensor network, in 2015 International Conference on Pervasive Computing (ICPC). IEEE, 1–5.

Delp, S. L., Anderson, F. C. Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). Opensim: open-source software to create and analyze dynamic simulations of movement, IEEE transactions on biomedical engineering, 54(11), 1940–1950.

González, L., Bishop-Hurley, G., Henry, D., & Charmley, E. (2014). Wireless sensor networks to study, monitor and manage cattle in grazing systems, Animal Production Science, 54(10), 1687–1693.

Good Morning America (GMA). Farmers explain why shutting down restaurants creates domino effect on their profits.(2020). Available:

Greenwood, P. L., Valencia, P., Overs, L., Paull, D.R., & Purvis, I.W. (2014). New ways of measuring intake, efficiency and behaviour of grazing livestock, Animal Production Science, 54(10), 1796–1804.

Guda P. & Gadhe, S. (2017). Primary processing of cocoa, International Journal of Agricultural Science and Research, 7(2), 457–462.

Helwatkar, A., Riordan, D., & Walsh, J. (2014). Sensor technology for animal health monitoring. International Journal on Smart Sensing & Intelligent Systems, 7(5), 1-6.

https://www.goodmorningamerica.com/living/video/farmers-explain-shutting-restaurants-creates-domino-effect-profits-70795278

Huang, Y., & Thomson, S.J. (2015). Remote sensing for cotton farming, Cotton, 57, 439–64.

Juul, J. P. Green, O., & Jacobsen, R. H. (2015). Deployment of wireless sensor networks in crop storages, Wireless Personal Communications, 81(4), 1437–1454.

Kutter, T., Tiemann, S., Siebert, R., & Fountas, S. (2011). The role of communication and co-operation in the adoption of precision farming. Precision Agriculture, 12(1), 2-17.

Lee, W. S., & Ehsani, R. (2015). Sensing systems for precision agriculture in Florida, Computers and Electronics in Agriculture, 112, 2–9.

Legal Insurrection. Wuhan coronavirus shutdowns hitting america's farms.(2020). Available: https://legalinsurrection.com/2020/04/wuhan-coronavirus-shutdowns-hitting-americas-farms/

Nikolidakis, S. A., Kandris, D., Vergados, D.D., & Douligeris, C. (2015). Energy efficient automated control of irrigation in agriculture by using wireless sensor networks, Computers and Electronics in Agriculture, 113, 154–163.

Nkwari, P. K. M., Rimer, S., & Paul, B. (2014). Cattle monitoring system using wireless sensor network in order to prevent cattle rustling, in 2014 IST-Africa Conference Proceedings. IEEE, 1–10.

Ojha, T., Misra, S., & Raghuwanshi, N.S. (2015). Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges, Computers and Electronics in Agriculture, 118, 66–84.

OpenSimulator. An open source multi-platform, multi-user 3D application server. (2020). Available: http://opensimulator.org/wiki/Main_Page

Ruiz-Garcia, L., Lunadei, L., Barreiro, P., & Robla, I. (2009). A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends, Sensors, 9(6), 4728–4750.

Rutten, C. J., Velthuis, A., Steeneveld, W., & Hogeveen, H. (2013). Invited review: Sensors to support health management on dairy farms, Journal of Dairy Science, 96(4), 1928–1952.

Saltini, R., Akkerman, R., & Frosch, S. (2013). Optimizing chocolate production through traceability: A review of the influence of farming practices on cocoa bean quality, Food Control, 29(1), 167–187.

Sarangi,S., Bisht, A., Rao, V., Kar, S., Mohanty, T.K., & Ruhil, A. P. (2014). Development of a wireless sensor network for animal management: Experiences with moosense, in 2014 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 1–6.

Satyr Farm. A Farming and Roleplay system for opensimulator. (2020). Available: https://satyrfarm.github.io

Srbinovska, M., Gavrovski, C., Dimcev, C., Krkoleva, A., & Borozan, V. (2015). Environmental parameters monitoring in precision agriculture using wireless sensor networks, Journal of Cleaner Production, 88, 297–307.

Sreekantha, D.K., & Kavya, A.M. (2017). Agricultural crop monitoring using IOT-a study, 11th International Conference on Intelligent Systems and Control (ISCO), IEEE, 134-139.

The Telegraph Reporters. Horse meat scandal: timeline. (2013). Available: https://www.telegraph.co.uk/foodanddrink/9857136/Horse-meat-scandal-timeline.html

Viazzi, S., Bahr, C., Van Hertem, T., Schlageter-Tello, A., Romanini, C., Halachmi, I., Lokhorst, C., & Berckmans, D. (2014). Comparison of a three-dimensional and two-dimensional camera system for automated measurement of back posture in dairy cows, Computers and Electronics in Agriculture, 100, 139–147.

Wang, J., Wang, H., He, J., Li, L., Shen, M., Tan, X., Min, H., & Zheng, L. (2015). Wireless sensor network for real-time perishable food supply chain management, Computers and Electronics in Agriculture, 110, 196–207.

Warren, S., Martinez, A., Sobering, T., & Andresen, D. (2008). Electrocardiographic pill for cattle heart rate determination, in 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 4852–4855.