A blockchain-based model for fungible assets and secure transformation processes traceability: an application to agro-food supply chain

Nicolas Herbaut[1], Etienne Baumgartner[2], Mehdi Chebbah[2], Chadi Grolleau-Raoux[2], Hugo Marques[2], Marieme Sow[2], Quentin Tambone[2], and Paul-Cesar Toux[2]

[1] Centre de Rercherche en Informatique, Université Paris 1 Panthéon-Sorbonne, Paris 75013, France
`nicolas.herbaut@univ-paris1.fr`

[2] MIAGE Sorbonne, Université Paris 1 Panthéon-Sorbonne, Paris 75013, France
`first.last@etu.univ-paris1.fr`

Abstract. Blockchain has been proposed to support supply chain use cases due to its numerous properties such as transparency, immutability, and auditability. For simple scenarios, such as asset property transfer, these solutions work well. Supply chain actors exchange their goods in a traceable manner and use smart contracts to enforce specific business rules. However, in cases where exchanged components are further used in manufacturing processes to produce refined items, which then serve as the basis for other products, maintaining traceability through these transformations is a challenge. Tracking and transforming physical goods also present trust issues concerning the actual existence of the goods and the conditions of the transformation processes, such as temperature or precise location. In this paper, we present BC24, a smart supply chain solution that uses a privacy-aware consortium blockchain and RFID to support asset, process, and environmental traceability along a supply chain. By leveraging standard token contracts ERC-1155, our solution is adaptable to a wide range of supply chain use cases and allows for chain interoperability. The duality of cyber and physical tracking is ensured by a secure NFC protocol. We validate our model through a real industrial use case in a, agro-food supply chain specialized in meat processing.

Keywords: blockchain; supplychain; IOT; ERC-1155 token; RFID; Agro-food

1 Introduction

The agriculture industry is under pressure due to the growing global population and the rising demand for safe, high-quality agri-food products. The food supply chain has become more globalized, with dependency on imported food increasing by 50% between 2006 and 2020. This has heightened consumer concerns about food safety and quality [1].

Today's supply chain management systems rely on centralized authorities for information transfer and sharing, leading to non-transparency, monopolistic control, and asymmetric information distribution. These systems are vulnerable to fraud, corruption, data falsification, and single points of failure [2]. Thus, often leading government authorities to respond to food scandals in order to maintain consumer trust. Ensuring high-quality and safe food products is essential for consumer health and market competitiveness. This quality insurrance is achieved through a more reliable agri-food supply chain processes. Ensuring traceability provides a comprehensive view of product harvesting, processing, and distribution phases [3].

To improve food safety and traceability and increase consumer trust, Blockchain Technologies offers a tamperproof, reliable, fraud-resistant, and trustworthy peer-to-peer network platform. Real-time risk point detection using blockchain can reduce food fraud and contamination while strengthening recall mechanisms for affected product batches [4]. This information allows consumers to reconstruct a complete history of a product's life cycle transactions, ensuring transparency and reliability [3].

Despite the promising concept of blockchain-based agro-food traceability, several issues remain unresolved, and numerous opportunities for improvement have emerged due to recent advancements in blockchain for information systems. These developments enable blockchain-based solutions to transcend mere digitization of ledger-based traceability, that is the norm today.

Firstly, while point-to-point traceability is beneficial, it often falls short in supporting transformation processes that merge resources. Ingredient sourcing should be traceable through all stages of food processing up to the distribution phase, ensuring trust from ingredients to their processing. Consequently, reliable agro-food traceability systems must verify not only the origin of ingredients but also the integrity of the production process and the final product. For example, an organic pancake restaurant should be able to trace the milk, eggs, and flour used in its dishes, advancing beyond simple atomic traceability (one item equals one certificate) to more flexible traceability capabilities. Secondly, despite cyber-level traceability and blockchain-backed food certificates, fraud can still occur in the physical world. For instance, an organic pancake restaurant might misuse organic certificates while using low-cost ingredients, claiming to maintain high traceability standards. Finally, the production process itself could be fraudulent by failing to adhere to legal regulations (e.g., maintaining required temperature levels) or misrepresenting the protected geographical indication for otherwise compliant products. For instance, the organic pancake restaurant might falsely claim to serve "Honey of Provence" while actually using a product from "Honey Laundering" [5].

In this paper, we propose BC24: a comprehensive model to ensure continuous agro-food traceability from resources to processes, and from the cyber to the physical world. In BC24, resources are represented through an extended token smart contract that enhances ERC-1155, supporting both fungible and non-fungible assets. This model also incorporates a configurable state machine engine that enforces critical domain business rules related to the creation, production, and transfer of assets. BC24 propose a device to support physical world traceability via local communication technology. This effectively and securely binds cyber resources on the blockchain with physical resources on physical token, managing their lifecycle jointly. Additionally, we propose the integration of sensor measurements into the traceability information and enforce specific business rules defined in the state machine through a dedicated state machine specification. In our evaluation, we demonstrate a real-world implementation of our model for an agro-food startup, showcasing how our approach effectively addresses meat-processing traceability issues.

The remainder of the paper is structured as follows: Section 2 illustrates the problem we aim to solve and establishes the requirements for a solution. Section 3 provides background information on related technologies. Section 4 presents our solution. Section 5 details our case study with a real-world implementation. Section 6 briefly compares our solution to related works. Finally, we conclude in Section 7.

2 Example, Problem Illustration, and Requirements

2.1 The Beef meat traceability problem

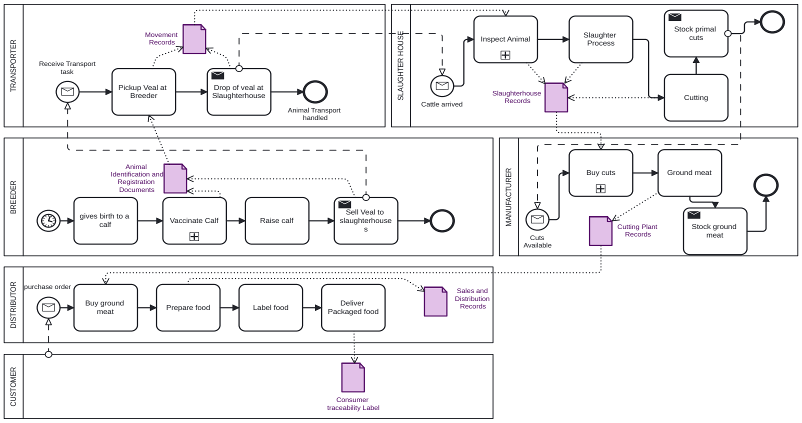


Fig. 1. A multi-party process for the end user food traceability documents

Figure 1 presents a Business Process Model and Notation (BPMN) collaboration diagram depicting a supply chain within the meat sourcing and processing sector[3], specifically focusing on the production of processed food containing

[3] adapted from french regulation for beef meat https://www.economie.gouv.fr/dgccrf/tracabilite-de-la-viande-bovine

meat. A new process instance is initiated when a customer purchases packaged food from a distributor. Given that the entire process is not executed on demand, with meat ingredients sourced beforehand, the diagram also illustrates the primary upstream operations: raising cattle, transporting it to a slaughterhouse, producing primal cuts, producing cut derivatives, and ultimately delivering the product along with *Consumer traceability documents* documents to the final client.

It is important to note that traceability information is transmitted and accumulated from one stakeholder to another. For example, a copy of *Animal Identification and Registration Documents* are provided to the transporter, that hand them over to the slaughethouse along with its own *Movement records*. The claimed legal target is to be able, for the final client, to trace back its product to the animal, as demonstrated in Table 1.

Table 1: Extract of the regulatory *Consumer traceability document* for beef meat in France.

Information	**Provenance**	**Purpose**
Use-by Date	Distributor	Consumer Safety
Batch number	Manufacturer	ID of meat cuts from several carcasses
Slaughter location	Slaughterhouse	Livestock slaughter location
Cut location	Manufacturer	Country in which the carcass was cut
Breeding country	Breeder	Origin country of the cattle

Malpractice incentives play a significant role; for example, the agromafia's influence in Italy is estimated to generate a revenue of €25 billion annually [6], indicating that fraud could be endemic. Despite the extensive measures in place, several scandals have exposed vulnerabilities. In 2013, horse meat was found in ground beef and distributed throughout Europe [7]. This issue was traced back to the French manufacturer Spanghero, which had relabeled horse meat as beef in its ground meat production. EU inspectors had to rely on DNA testing of the final product to confirm the fraud.

Even though new "food integrity" measures were implemented by the EU administration based on eight pillars—consumers first, zero tolerance, intelligence gathering, laboratory services, audit, government support, leadership, and crisis management [8]—the approach to preventing fraud remains largely based on a posteriori verification, document integrity, and trust in manufacturing processes. For instance, DNA testing would prove ineffective in preventing the practice of producing 1 kg of organic-branded ground meat by mixing 500 g of low-cost meat with 500 g of genuine organic meat. This issue can be summarized as a lack of continuity between the cyber world and the physical world in production processes.

To summarize, the three main goals that need to be achieved to regain customer trust in meat processing are data integrity, ensuring that data is not tampered with; cyberphysical continuity, effectively linking information artifacts to physical products; and processing flexibility, allowing stakeholders to continue using their existing processes.

These aspects lead to the series of requirements we will discuss next.

2.2 Requirements

Table 2: Goals and Requirements of meat processing traceability.

	Goal	Requirement
R1	Data Integrity	Information artifact must be permanent
R2	Data Integrity	Information artifact must be non-repudiable
R3	Data Integrity	Information artifact for process inputs must be propagated to outputs
R4	Data Integrity	Processes themselves must be traced and prevent abuse
R5	Cyberphysical continuity	Material cannot be processed without a physical token
R6	Cyberphysical continuity	Processes must generate a new physical token attached to their output
R7	Processing flexibility	Process must support using only a fraction of a certified input
R8	Processing flexibility	Selected inputs for processes can be non-certified

First, any information added to the record must possess integrity and non-repudiability properties (**R1** and **R2**); this is referred to as certified information for certified products used in certified processes. This prevents the forging and altering of records. Second, upstream traceability information on certified raw materials should be automatically propagated during processing and attached to the resulting certified product (**R3**). For instance, when processing raw materials, any relevant records must be attached to the resulting products. As these products are further processed, this information will remain attached, along with new information such as the responsible manufacturer's details. Additionally, thresholds and safeguards should be applied to ensure that processing a batch cannot yield more output than input (**R4**). This prevents stakeholders from injecting unsourced material during the certified manufacturing process. To trace transport or processing, a physical token representing the traceability information should be used to confirm the actual presence of the certified raw material (**R5**). No certified process can occur without verifying the presence of the physical token. The output of the process should be accompanied by a new physical token, which is initialized from the application of a given certified process to its input (**R6**). The underlying physical materials could be used entirely or partially in a certified process, depending on the parameters (**R7**). Any remaining certified material after certified processing could be used in another certified process. Process execution should support using base materials that do not require any certification (**R8**). For example, if only certain ingredients are certified, the process would prevent manufacturing more than the allowable quantity, thereby preventing the addition of unsourced material to produce more final products.

3 Background

Blockchain technology, first conceptualized by an anonymous entity known as Satoshi Nakamoto in 2008 [9], is a decentralized ledger that records transactions across multiple computers to ensure security, transparency, and immutability. Blocks containing transaction data are cryptographically linked, forming a chain that resists tampering and centralized control. This robust platform supports various applications beyond cryptocurrencies, including supply chain management [10], healthcare [11], and digital identity verification [12].

Building on blockchain, Non-Fungible Tokens (NFTs) emerged as unique digital assets authenticated through the blockchain [13]. Unlike fungible cryptocurrencies, NFTs represent distinct items and ownership. Each NFT contains metadata that ensures its individuality and provenance, making them suitable for representing digital art, collectibles, real estate, and other unique assets. The ERC-1155 token standard represents an evolution of ERC-721, offering enhanced flexibility and efficiency [14]. ERC-1155 allows both fungible and non-fungible tokens within a single contract, addressing limitations of ERC-20 and ERC-721. This reduces transaction costs and complexity, making ERC-1155 valuable for applications like gaming, where a variety of assets need management.

Radio-Frequency Identification (RFID) and Near Field Communication (NFC) enable wireless data transfer between devices. RFID, used in inventory management and asset tracking, identifies and tracks tags attached to objects. NFC, a subset of RFID, operates within a shorter range and is used for secure transactions and access control.

NFC's security features include encryption and secure channels, ensuring data transfer cannot be intercepted or altered. This security makes NFC suitable for integration with blockchain and NFTs, providing a robust method for verifying the authenticity and provenance of physical items linked to digital tokens. Combining NFC's security with blockchain's immutability establishes a secure, end-to-end traceability system, enhancing trust in applications like supply chain management and digital identity verification.

In summary, blockchain's decentralized and secure architecture, combined with the uniqueness of NFTs and the advanced capabilities of ERC-1155, provides a powerful framework for developing innovative applications across various industries. The integration of these technologies promises to enhance digital asset management, improve transactional transparency, and foster the growth of decentralized ecosystems.

4 BC24 solution

In this section, we present the BC24 model, which consist of a Distributed Cyberphyisical System comprised of:

- The BC24 Network that supports the entire traceability solution and allows different actors to interact with the overall system. 4.1
- Several Field Nodes deployed on stakeholder premices that supports physical token management, process interaction and also participate in the consensus protocol described in section 4.2
- A Smart Contract that manages the artifact information traceability using a configurable state machine engine 4.3

4.1 BC24 System Overview

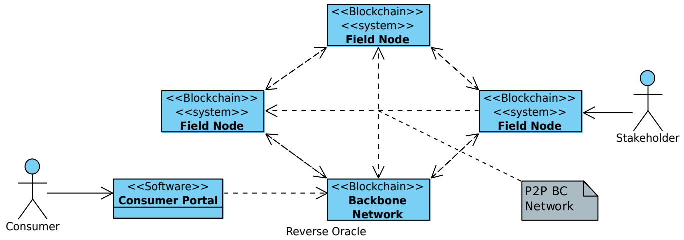


Fig. 2. BC 24 Network Block Diagram

Figure 2 presents a SysML Block diagram illustrating the overall system architecture. Given that blockchain is a decentralized system, it encompasses numerous redundant functions. The `Backbone Network` is a blockchain network maintained by a consortium of stakeholders. This network consists of several core nodes running validators responsible for processing transactions and ensuring the liveness of the consensus algorithm.

Built upon this `Backbone Network`, each supply chain `Stakeholder` is equipped with a `Field Node` that not only runs a blockchain node validator but also allows stakeholders to interact with the process and its resource traceability features. Each field node is equiped with an unique cryptographic identification, that enables the secure interaction with the distributed system.

For supply chain participants requiring only data visualization, such as `Consumer` or auditors, a `Consumer Portal` application is available. This application is integrated with the blockchain network via an interface and retrieves and displays the corresponding artifact information.

The subsequent sections will delve into the field node, exploring its capabilities and design in greater detail.

4.2 BC24 Field Node

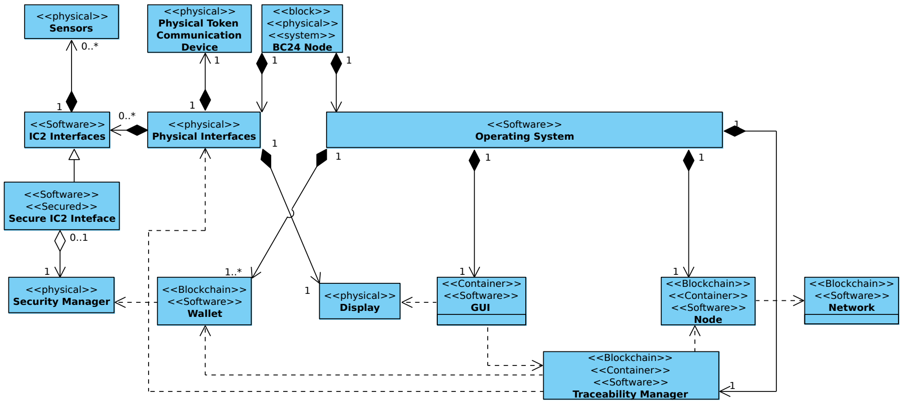


Fig. 3. BC 24 Field Node Internal Block Diagram

Field nodes facilitate various interactions for stakeholders within the system. Figure 3 presents an internal block diagram of the system, comprising a hardware component supporting `Physical Interfaces` and a software component supporting the `Operating System`.

The `Wallet` software securely accesses private keys used to sign blockchain transactions through the `Security Manager` and interacts with the traceability smart contracts via the `Traceability Manager` (See 4.3). The `Traceability Manager` functions as a backend for the `GUI` application. Through the `GUI`, stakeholders can register new resources or process their owned resources. Additionally, a `Secure Interface` is used for secure storage of cryptographic materials, managed by the `Security Manager`.

When a resource is processed, specific sensor data may be added depending on the process. For example, a transporter must accurately record the exact pickup and delivery locations for any given cattle transport. The `Traceability Manager` is also involved during the processing of physical resources. A physical token must be presented to the `Physical Token Communication Device` which subsequently asserts the possession of the provided underlying physical product. Similarly, when creating new products by processing other suitable resources, the Communication Device module writes the corresponding reference information of the created virtual resource on a newly provided physical token.

These interactions are managed by the `GUI` module throug the physical `Display`, which prompts users to present existing physical tokens for reading or new physical tokens for initialization.

Having detailed the field nodes, we will now explore the digital aspect of the traceability system stored on the blockchain.

4.3 BC24 Tracing contract

To store traceability data in a secure and non-repudiable way, it is stored in a smart contract deployed on the blockchain (as per requirements **R1** and **R2**). The smart contract is designed for storing all the data collected from the field node and provides additional key operation such as resource creation, resource processing and resource transfer along the way.

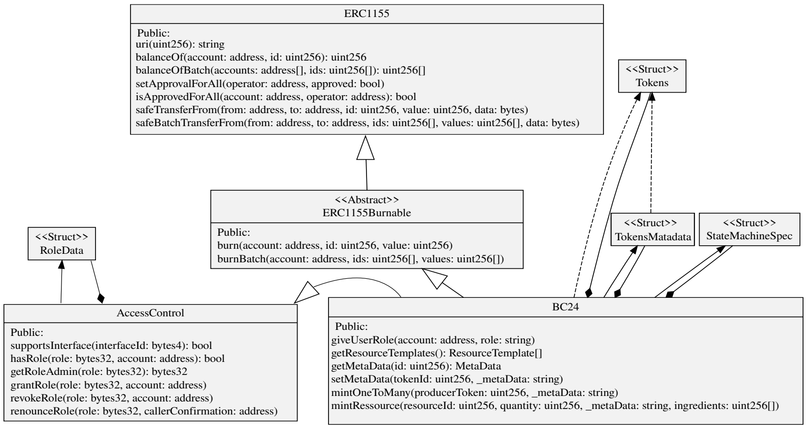

ERC1155
Public:
uri(uint256): string
balanceOf(account: address, id: uint256): uint256
balanceOfBatch(accounts: address[], ids: uint256[]): uint256[]
setApprovalForAll(operator: address, approved: bool)
isApprovedForAll(account: address, operator: address): bool
safeTransferFrom(from: address, to: address, id: uint256, value: uint256, data: bytes)
safeBatchTransferFrom(from: address, to: address, ids: uint256[], values: uint256[], data: bytes)
<<Struct>>
Tokens
<<Abstract>>
ERC1155Burnable
Public:
burn(account: address, id: uint256, value: uint256)
burnBatch(account: address, ids: uint256[], values: uint256[])
<<Struct>>
RoleData
<<Struct>>
TokensMatadata
<<Struct>>
StateMachineSpec
AccessControl
Public:
supportsInterface(interfaceId: bytes4): bool
hasRole(role: bytes32, account: address): bool
getRoleAdmin(role: bytes32): bytes32
grantRole(role: bytes32, account: address)
revokeRole(role: bytes32, account: address)
renounceRole(role: bytes32, callerConfirmation: address)
BC24
Public:
giveUserRole(account: address, role: string)
getResourceTemplates(): ResourceTemplate[]
getMetaData(id: uint256): MetaData
setMetaData(tokenId: uint256, _metaData: string)
mintOneToMany(producerToken: uint256, _metaData: string)
mintRessource(resourceId: uint256, quantity: uint256, _metaData: string, ingredients: uint256[])

Fig. 4. BC 24 Contract Class Diagram

4.3.1 Design The class diagram of the BC24 tracing smart contract, shown in Figure 4, utilizes the ERC-1155 standard for token management and runs on the Ethereum Blockchain.

This setting allows optimal gas spending and the `ERC1155` standard is optimal because of its flexibility regarding token representation. Said tokens can be either non-fungible, representing unique respectively unitary assets (such as cattle), or fungible tokens, which are essentially interchangeable and therefore well suited to represent non-unitairy resources (such as a beef cuts).

We extended the main functionality of the `ERC1155` standard to support the metadata management and its traceability even for fungible resources. This features as well as any other functionality is created with access clearance in mind. Each stakeholder in our system assumes a specific role, granting access to certain processes. This concern is reflected by the `AccessControl` class, which is extended by our BC24 contract.

We defined the following three structures to support traceability:

- `Token`: Represents the resources on the blockchain, which can be processed or transferred from one stakeholder to another.

- TokenMetadata: Represents the metadata for each token and contains the traceability information.
- StateMachineSpec: This parameter is passed to the contract constructor and describes the certified process used in the system to manage the creation of (output) tokens from (input) tokens. It also manages guard conditions on the process execution, described in Section 4.3.2, such as how many tokens of resource A and B have to be used to produce a given quantity of resource C.

Regarding the BC24 contract implementation itself, in addition to the standard implementation inherited from ERC1155 and ERC1155Burnable, we propose the mintResource function, which can be called in two ways to create new certified resources:

- If the resources do not require any existing resource for creation (such as cattle), then its creation (called minting for blockchain tokens) is done ex-nihilo. The stakeholder allocates a new token along with new metadata.
- If the resources need other certified input resources, then their creation is done through a certified process defined in the StateMachineSpec structure. In this case, the certified resources provided as input are burnt with ERC1155Burnable(this means permanently marked as used on the blockchain). The corresponding physical tokens are disabled, and the BC24 contract mints new resources, assigning the stakeholder as their owner. Every metadata attached to any input resource used in the process is appended to the newly minted token's metadata, assuring downstream traceability. Potential leftover resources not used in the process remain owned and usable by the stakeholder for further processing or transfer to another stakeholder, thanks to its corresponding physical tokens that remain usable.

As mentioned before, the processes of resource creation are defined by the StateMachineSpec Struct. In the following section, we describe the state machine engine configured by the StateMachineSpec instance passed at construction time, as well as the StateMachineSpec class diagram.

4.3.2 Configurable State Machine

First, the state machine is initialized at contract construction time. Given the significant impact of the state machine specification on operations, we developed a formal verification tool to ensure its consistency by verifying that every resource can be effectively created either directly or through processing. Once the state machine is configured, the contract is ready to be called via the safeTransferFrom function, transitioning the state machine to the Transferring State, or through the mintResource function, transitioning to the Clearance state.

Both states have preconditions to ensure the operation is legal according to the state machine definition. For the Transferring State, the hasAvailableResources precondition specifies that to transfer an asset to a new stakeholder, the caller of the safeTransferFrom function must own sufficient corresponding tokens. For the Clearance state, two preconditions must be met for

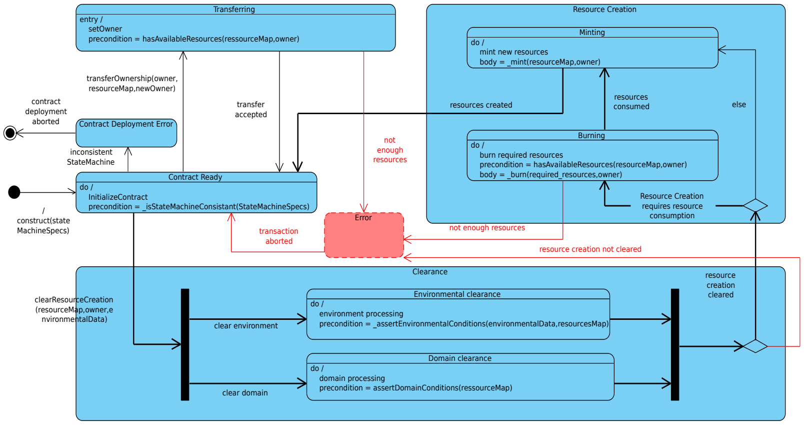

Transferring
entry /
setOwner
precondition = hasAvailableResources(ressourceMap,owner)
transferOwnership(owner, resourceMap,newOwner)
transfer accepted
contract deployment aborted
Contract Deployment Error
inconsistent StateMachine
Contract Ready
do /
InitializeContract
precondition = _isStateMachineConsistant(StateMachineSpecs)
/ construct(state MachineSpecs)
not enough resources
Error
transaction aborted
Resource Creation
Minting
do /
mint new resources
body = _mint(resourceMap,owner)
resources created
resources consumed
else
Burning
do /
burn required resources
precondition = hasAvailableResources(resourceMap,owner)
body = _burn(required_resources,owner)
Resource Creation requires resource consumption
not enough resources
resource creation not cleared
Clearance
clearResourceCreation (resourceMap,owner,environmentalData)
Environmental clearance
do /
environment processing
precondition = _assertEnvironmentalConditions(environmentalData,resourcesMap)
clear environment
Domain clearance
do /
domain processing
precondition = assertDomainConditions(ressourceMap)
clear domain
resource creation cleared

Fig. 5. BC 24 Contract State Machine Diagram

the process to proceed: `Environmental Clearance` and `Domain Clearance`, encapsulated by two substates of `Clearance`. The state machine specifications can impose guard conditions based on sensor data, passed as metadata. For example, if a slaughterhouse is registered in France, the GPS coordinates reported by the system should confirm that the cattle is located in France. If the GPS coordinates are unavailable or indicate a location outside France, the environmental clearance would fail, causing an error and aborting the transaction. Similarly, for `Domain Clearance`, if there is a discrepancy between the formally stated process requirements and the metadata, a similar error would be raised. For instance, if the state machine expects a vaccination date for the cattle and it is not present in the metadata.

Once the `Clearance` state is reached without errors, the system transitions to the `Resource Creation` state. As mentioned in section 4.3.1, two scenarios are possible based on the creation requirements of the selected resource. If other resources are needed, the system transitions to the `Burning` substate, where the corresponding resource tokens are permanently burnt after checking the `hasAvailableResources` precondition. Then, the system transitions to the `Minting` state, where new tokens are created in quantities corresponding to the input resource and state machine configuration, marking the function caller as the owner. During this step a reference of the burnt resources is added in the newly created token metadata.

Whenever a token is minted, the owner will initialize and assign the corresponding virtual resource token reference to a physical token. Likewise, when a token is burnt, the corresponding physical token is not usable by anyone.

If no errors occur, the state machine returns to the `Contract Ready` state and can receive new orders.

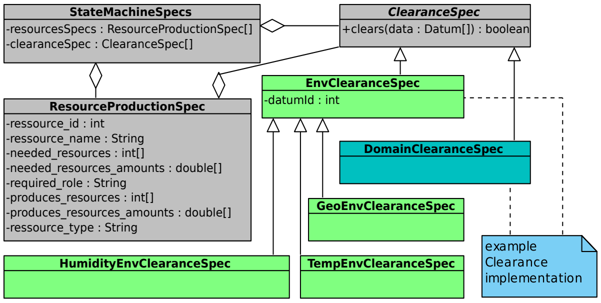


Fig. 6. BC 24 StateMachine Specifications Class Diagram

4.3.3 BC 24 State machine specification We now describe the state machine specfications, Figure 6, used to configure the creation and processing of resources. It consists of a set of `ResourceProductionSpec` and `ClearanceSpec` structures. The `ClearanceSpec` defines a series of clearance preconditions for the process execution, while the `ResourceProductionSpec` specifies the required input and output resources, their respective quantities for each process, and the applicable clearance preconditions.

For example, the French AOP Maine-Anjou is a Protected Designation of Origin (PDO) that limits the PDO to a list of 603 authorized municipalities for processing and imposes various conditions on the animal breed and carcass weight[4]. The process specification can define that to produce 1 kg of organic pot-au-feu meat from AOP Maine-Anjou, the manufacturer must use organic primal cuts of beef (specified in the needed_resources attribute) aged no more than 10 years (specified in a domain clearance and obtained through the cattle metadata retrieved from the primal cuts metadata) that were born, raised, and slaughtered in the PDO (specified in an environmental clearance based on GPS data at each step). Additionally, the carcass must weigh more than 380 kg (domain clearance on the carcass metadata), and 100 kg of carcass can yield up to 15 kg of pot-au-feu primal cuts (specified in the produces_resources_amounts attribute).

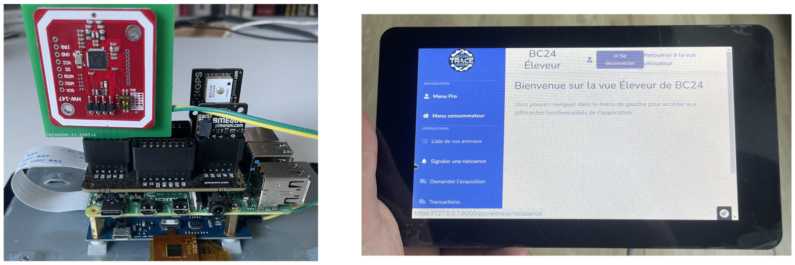
Now that we have described the BC24 system, we move to the case study.

5 Case study

5.1 Trace SAS Prototype

We implemented the BC24 model for a startup company, Trace SAS, incubated in the MIAGE department of Université Paris 1 Panthéon-Sorbonne. This company aims to sell meat traceability solutions for every stakeholder in the supply chain.

[4] This specification is derived from *Décret n° 2011-536 du 16 mai 2011 relatif à l'appellation d'origine contrôlée Maine-Anjou.*



Bienvenue sur la vue Éleveur de BC24

Vous pouvez naviguer dans le menu de gauche pour accéder aux différentes fonctionnalités de l'application.

Fig. 7. BC24 prototype implementation for Trace SAS

The prototype[5] developed, as shown in Figure 7, utilizes a low-cost Raspberry Pi 4 with 8GB of RAM running the Debian Bookworm OS. A touch screen is connected for easy interaction, along with a series of sensors on the I2C Interface: a BME680 module for temperature and humidity level readings and a PA1010D module for GPS location readings and time synchronization.

We used NFC technology as the physical device communication module, and Mifare tags as physical tokens. To this end, a PN532 module was included for reading and writing of NFC chips. Additionally, the Security Manager relies on a ZYMKEY4 security module for secure hardware storage of cryptographic materials. This module ensures device integrity, preventing the execution of altered software and access to the file system via the Raspberry Pi SD card, making it an optimal choice for unattended IoT devices.

For the blockchain layer, we deployed a backbone of Hyperledger Besu, an EVM-compatible blockchain solution that offers a permissioned blockchain deployment running the QBFT proof of authority consensus protocol with a block-period of 4s. The network was deployed on a Kubernetes cluster of virtualized nodes hosted by a cloud provider across several availability zones. For simplicity, the gas price was set to zero, ensuring that all data and metadata storage happens on-chain. We extended classes provided by the OpenZeppelin library[6].

The GUI was developed in PHP and serves as the embedded application on the Pi, as well as a consumer portal.

5.2 BC24 Requirements analysis

We illustrate requirements coverage through the usage description of the Trace SAS system, provided as a sales speech to potential investors. First, a breeder uses their field node to register the birth of a new animal. They tap on the GUI, enter information, and submit a form. A transaction for the new animal is signed from the built-in wallet stored on the Pi. Once the transaction is completed and the ownership of the animal along with its metadata is stored both on the Backbone Hyperledger Besu network and on the local Besu node (**R1**,

[5] Source code and demo https://github.com/bc24-miage-dev

[6] https://www.openzeppelin.com/.

R2), the breeder receives confirmation and is prompted to initialize a new NFC tag, which they present to the NFC module (**R6**). A month later, the animal receives preventive treatment. The breeder scans the NFC tag, which opens the animal's page (**R5**), allowing them to specify the treatment received. On-chain, the metadata is updated through a new transaction that only the breeder can perform.

A few months later, the animal is sold to a slaughterhouse. The animal and its NFC tag are sent to the slaughterhouse. The operator scans the NFC tag on their field node, transferring ownership of the animal to the slaughterhouse via the pre-approved transaction stored in the NFC tag. The operator slaughters the animal and taps the "slaughter" button on the animal's page. The application requests a new NFC tag to be associated with the carcass. The operator scans the newly initialized NFC tag, bringing them to the carcass page, where they can see all the animal's information, including vaccination history (**R3**). They then proceed to cut the carcass. They scan the NFC tag of the carcass, click on "cut," and enter information about the cuts. If they mistype the number of legs, specifying three legs, an error appears upon submission of the cut transaction, indicating that the information is incorrect. The Trace SAS smart contract domain clearance fails to validate the transformation from a carcass to three legs (**R4**). The operator corrects the form and resubmits, generating NFC tags for the produced cuts.

Later, a butcher receives a cut of meat. To apply the "Viande Française" label, the butcher scans the NFC tag corresponding to the cut and verifies that the animal was bred and slaughtered in France, complying with regulations. Finally, the butcher prepares roast beef sandwiches to sell to customers. Each sandwich contains a slice of 100g of roasted beef among other ingredients (**R7**, **R8**) and is accompanied by its own NFC tag, which the customer can scan with their mobile phone to see the origin of the meat.

From the previous scenario, we can see that every requirement is covered but the Trace SAS implementation.

5.3 Performance testing

We carried out stress tests using Locust[7] on the operations involving blockchain data access and modification. Table 3 shows that the read operations show acceptable delays of less that a second to retreive the data for a given token. Write operations (creation and transfer) are impacted by the block creation time, and show an overhead below 2s, which is also acceptable for production and may benefit from fine-tuning of the besu deployment.

Metric / Operation	**get metadata**	**create resource**	**transfer resource**	**get all resources list**
median response time (ms)	590	5000	4800	610
Average payload size (b)	639	578	249	6329

Table 3. Stress test of the most common blockchain operations

[7] https://locust.io/

6 Related work

A few other proposal were made with related technologies in the litterature. proposes using ERC-1155 tokens for different industries such as agro-food [15], automotive [16] however, those proposal only track transfert and metadata and not processes.

Other reasearch dissued using RFID technologie for agroo-food traceability [17], however they do not use blockchain for guarantee data integrity, but instead a centralized cloud database own by a trusted thrid party. Our proposal differes in the sense that it does not completely rely on a trust third party for data validation, but only the identification of the stakeholders participating in the consensus algorithm, as imposed by the permissionned blockchain model.

Other proposal suggested using both blockchain and RFID [18], but only manage one feature of the food (hallal or not), without taking into account other proceses.

7 Conclusion

This paper presented BC24, a blockchain-based model that integrates fungible asset tracking and secure transformation process traceability, applied to the agro-food supply chain. Our solution leverages a privacy-aware consortium blockchain and RFID technology to ensure continuous traceability from raw materials to finished products. By extending the ERC-1155 token standard, BC24 effectively manages both fungible and non-fungible assets, providing a flexible framework adaptable to various supply chain scenarios.

Our evaluation, demonstrated through a real-world implementation for a meat processing startup, showcases the model's capability to address critical traceability challenges. The secure integration of sensor data and physical token verification ensures the integrity of both the cyber and physical aspects of the supply chain. The implementation showed acceptable performance metrics for both read and write operations, confirming its feasibility for production environments.

The BC24 model not only addresses the immediate needs of the agro-food supply chain but also provides a robust foundation for other industries requiring stringent traceability. Future work will focus on expanding the model's applicability, optimizing the underlying blockchain infrastructure, and further refining the integration of more secure NFC technologies, such as Myfare Desfire to enhance real-time data accuracy and process automation.

Reference

[1] T. Bosona and G. Gebresenbet, “The role of blockchain technology in promoting traceability systems in agri-food production and supply chains,” *Sensors*, vol. 23, no. 11, p. 5342, 2023.

[2] F. Tian, “An agri-food supply chain traceability system for china based on RFID & blockchain technology,” in *2016 13th international conference on service systems and service management (ICSSSM)*, 2016, pp. 1–6.

[3] A. Marchese and O. Tomarchio, “A blockchain-based system for agri-food supply chain traceability management,” *SN Computer Science*, vol. 3, no. 4, p. 279, 2022.

[4] A. Patel, M. Brahmbhatt, A. Bariya, J. Nayak, and V. Singh, “Blockchain technology in food safety and traceability concern to livestock products,” *Heliyon*, vol. 9, no. 6, 2023.

[5] D. C. Hall, “Managing fraud in food supply chains: The case of honey laundering,” *Sustainability*, vol. 15, no. 19, p. 14374, 2023.

[6] F. Fanizza and M. Omizzolo, *Caporalato: An authentic agromafia*. Mimesis, 2019.

[7] L. Agnoli, R. Capitello, M. De Salvo, A. Longo, and M. Boeri, “Food fraud and consumers’ choices in the wake of the horsemeat scandal,” *British Food Journal*, vol. 118, no. 8, pp. 1898–1913, 2016.

[8] S. Brooks, C. T. Elliott, M. Spence, C. Walsh, and M. Dean, “Four years post-horsegate: An update of measures and actions put in place following the horsemeat incident of 2013,” *npj Science of Food*, vol. 1, no. 1, p. 5, 2017.

[9] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[10] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, “Blockchain technology and its relationships to sustainable supply chain management,” *International journal of production research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[11] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, “Blockchain technology in healthcare: A systematic review,” in *Healthcare*, 2019, vol. 7, p. 56.

[12] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, S. Bhatia, A. Mashat, A. Kumar, and M. Kumar, “Self-sovereign identity solution for blockchain-based land registry system: A comparison,” *Mobile Information Systems*, vol. 2022, no. 1, p. 8930472, 2022.

[13] M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, and A. Baronchelli, “Mapping the NFT revolution: Market trends, trade networks, and visual features,” *Scientific reports*, vol. 11, no. 1, p. 20902, 2021.

[14] M. Di Angelo and G. Salzer, “Tokens, types, and standards: Identification and utilization in ethereum,” in *2020 IEEE international conference on decentralized applications and infrastructures (DAPPS)*, 2020, pp. 1–10.

[15] R. B. Dos Santos, R. P. Pantoni, and N. M. Torrisi, “Blockchain tokens for agri-food supply chain.”

[16] M. Kuhn, F. Funk, and J. Franke, “Blockchain architecture for automotive traceability,” *Procedia Cirp*, vol. 97, pp. 390–395, 2021.

[17] D. Pigini and M. Conti, “NFC-based traceability in the food chain,” *Sustainability*, vol. 9, no. 10, p. 1910, 2017.

[18] N. N. Ahamed, R. Vignesh, and T. Alam, "Tracking and tracing the halal food supply chain management using blockchain, RFID, and QR code," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48987–49012, 2024.