Modeling Carne PRI supply chain with the *-Chain Platform

Stefano Bistarelli[1], Francesco Faloci[2,3,*], Marino Miculan[4], Paolo Mori[3] and Carlo Taticchi[1]

[1]Università degli Studi di Perugia
[2]Università di Camerino; Italy
[3]Istituto di Informatica e Telematica Consiglio Nazionale delle Ricerche Italy
[4]Università degli Studi di Udine

Abstract

Certification of the origin of products and control over the production chain are fundamental activities that can make goods competitive. Therefore, it is of great interest to develop platforms that enable domain experts to build supply chain management systems with easy-to-use systems, providing integration with structures that can guarantee the certification of a certain good. This paper shows how the *-chain framework solves this problem. In particular, we represent the use case: "Pezzata Rossa" - Carne PRI supply chain. *-chain generates the related blockchain-based traceability system. The framework's tools generate a set of solidity smart contracts implementing the system and three web interfaces to interact with them.

Keywords

Supply Chain, Blockchain, Distributed Ledger Technology, Domain Specific Graphical Language, Smart Contracts, Automatic Smart Contract Generation, Carne PRI

1. Introduction

Selling high-quality goods in today's highly competitive marketplace requires producers to provide potential customers with added value, such as certifying their quality or defining their origin. The presence of trusted information raises the quality of a product, and information about the production chains can drive consumers' decisions towards more expensive products. Strict protocols and regulations, such as Protected Designations of Origin (PDOs), are being established to ensure that production chains have high-quality goods as end products. Among European countries, Italy is the one with the highest number of PDO products[1]; for instance, thirty-nine PDOs have been defined for olive oil, spread over the various regions, thirty-nine others for cheese and nineteen for meat products, each defining several constraints on the production process. We consider the case of the *Italian Pezzata Rossa*, or PRI, a bovine breed developed in Friuli in 1870 for work, milk, and meat. The breed's meat is highly regarded

DLT 2023: 5th Distributed Ledger Technology Workshop
*Corresponding author.
stefano.bistarelli@unipg.it (S. Bistarelli); francesco.faloci@unicam.it (F. Faloci); marino.miculan@uniud.it (M. Miculan); paolo.mori@iit.cnr.it (P. Mori); carlo.taticchi@unipg.it (C. Taticchi)

[1]Web portal on geographical indications: https://www.tmdn.org/giview.

for its quality and fine marbling. A labelling system known as "La Rossa Pezzata del Friuli Venezia Giulia" has been recently established to ensure that the end consumer receives correctly processed meat of a guaranteed origin. This labelling system is subject to several European and Italian regulations and decrees, including EP Regulation 1760/2000, EC Regulation 1825/2000, and the Ministerial Decree of 16.01.2015. The "La Rossa Pezzata del Friuli Venezia Giulia" label is only applied to meat sold to end consumers in all supply chain stages that have been thoroughly documented and registered in accordance with the specification in [1]. This specification also outlines the information that must be provided at each supply chain step, known as *Carne PRI*, to maintain a clear link between each animal and the derived meat. To guarantee that all these constraints are respected, it is necessary to put in place some mechanisms to check how the production process is carried on, to compare the observed data with the regulations in law of that specific product, and then to integrate the result also taking into account transport and intermediate sales information. Supply Chain Management Systems [2] (SCMSs) are software systems that track production processes, from raw materials, through intermediate and half-processed products, to the final goods. Through an SCMS, a user controls and analyses the current state of an asset. Blockchain Technology (BT) for implementing SCMSs provides several advantages [3, 4]. First, blockchains natively provide *auditability* because all transactions registered on the blockchain will be available forever and without the possibility of being cancelled or altered. Moreover, blockchains also offer trusted support for data processing. Keeping track of supply chain tracking information in blockchain systems makes it easier to monitor supply chain phases because the data observed for each phase is stored permanently and immutably. However, designing and developing a blockchain-based SCMS is nontrivial and requires expertise concerning supply chain principles and blockchain technology [5, 6, 7].

In [8, 9], we presented *-chain, a framework that allows goods producers to design the SCMSs representing their production processes through a simple and user-friendly graphical interface based on a graphical language and to automatically obtain from this graphical representation the set of smart contracts implementing the related SCMSs on the blockchain, as well as a set of web-based graphical interfaces that can be used for SCMSs management, both from administrators, production process participants, and customers sides. In addition, we have already tested the application of the *-chain framework in [10], where we exploited the tool for the use case PDO Olive Oil.

In this paper, we focus on the Carne PRI supply chain. We show how the *-chain framework can be exploited to build a blockchain-based SCMS representing a production process for "Pezzata Rossa" certification. We focus on how such a process can be represented through our graphical platform and then used by the *-chain framework for producing a blockchain-based implementation of the SCMS.

2. Background

Distributed Ledger Technology (DLT) refers to systems and protocols dealing with immutable network data. It is more commonly known as Blockchain Technology (BT), given its potential across industries and financial sectors. DLT refers specifically to the technological infrastructure and protocols that allow the simultaneous access, validation and updating of records exploiting distributed ledgers. BT, however, is not limited to cryptocurrencies. BT can have applications in non-financial sectors, such as traceability problems and workflow organization. A smart contract is a self-executing contract (script) with the terms of the agreement between two actors, generally a buyer and a seller, directly written into lines of code. One of the most popular coding languages for describing smart contracts is Solidity [2], widely used for Ethereum [3] systems. The *-chain framework, then, consists of various tools based on the Domain-Specific Graphical Language (DSGL) we developed to represent the flow of an SC and the tracking of an asset. This model is a fundamental part of the platform and the web graphical user interfaces and follows a more asset-centric point of view instead of recording the notification of the execution of a production phase. The language includes objects and operations; objects can be assets or containers, while operations can be of the following types: asset_create, asset_update, asset_transform, asset_pack, asset_unpack, monitor, sell and give_control.

3. The Carne PRI supply chain representation

3.1. Carne PRI flowchart

In the following, we describe the process that defines the production of the "Pezzata Rossa" certification according to the Carne PRI denomination. All process phases are illustrated in Figure1: each flowchart block identifies a phase, while the colours identify the actors performing that phase. Optional phases are represented with a dashed border. First of all, the *farmer* must identify the cattle by assigning a particular ID (*ID Assignment*), applied as a label to the animal's ear. The animal can then be either introduced into the herd (*Animal Introduction*) or purchased (Animal Purchasing). Both these options bring to the breeding phase (*Breeding*). During breeding in the barn, the *veterinarian* may administer antibiotics (*Antibiotics*). At this stage, there may also be an initial awarding of the *ANAPRI* Carne PRI Certification (*ANAPRI Certification*). After a four-month rearing period, the cow is ready for slaughter. Before transport, the specialised *veterinarian* evaluates the animal (*Cow Evaluation*). Then, the cow is transported to the slaughterhouse by one of the accredited *deliver* (*Cow Transport*). After the delivery, the cow undergoes a first analysis (*Ante Mortem Inspection*). Then, after the approval of the *veterinarian*, the *butcher* proceeds to slaughter (*Slaughter*). The carcass must rest for 48 hours before being processed, passing the second evaluation (*Post Mortem Inspection*). Then, the technician gives the carcass stamps and codes delivered to ANAPRI and the ID Mark of the animal (*Carcass Evaluation*). If the butcher's shop is not equipped to handle the carcass properly, the meat is left in the blast chiller (*Carcass Maturing*) and is only then transported to a specialised butcher's shop (*Carcass Transport*). In the specialised butcher's shop, the butcher cuts up the carcass, dividing it into homogeneous parts for processing (*Sectioning*). If the meat has not been placed in the blast chiller, it is left on rest for about 48 hours (*Meat Maturation*). The chopped meat is then transported to the sale point (*Meat Transport*), where other butchers will either grinds (*Grinding*) or cut (*Portioning*) the pieces of meat. Each preparation is then packaged (*Packaging*) to be ready for sale (Selling).

[2]Solidity project: https://docs.soliditylang.org/en/v0.8.4/.
[3]Ethereum project: https://ethereum.org/en/.

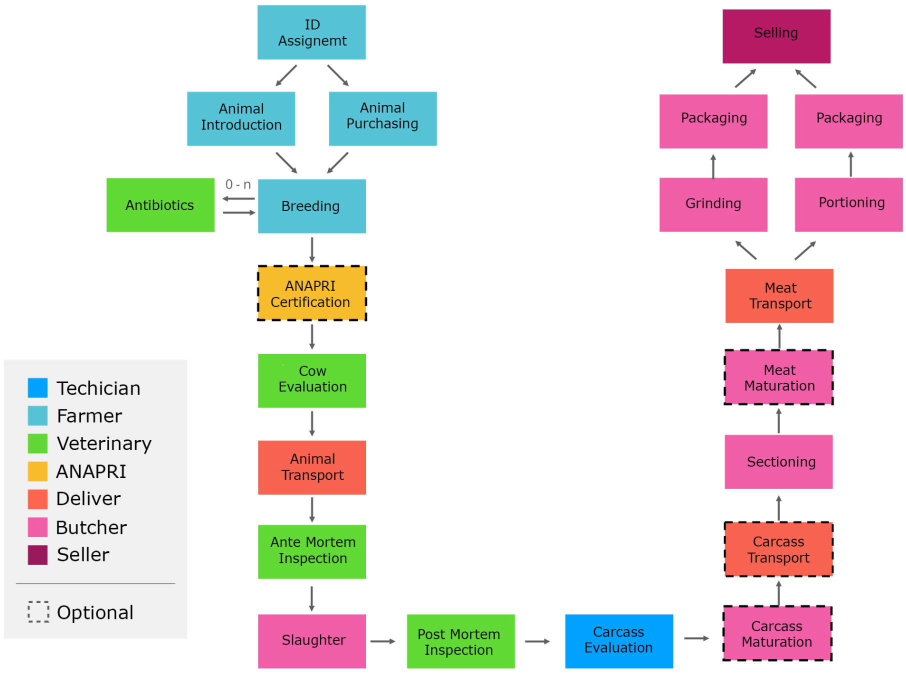

Figure 1: The Carne PRI schema.

3.2. Carne PRI graphical representation with *-chain

The *Operations* are the elements of our DSGL aimed at representing the activities the production process performs on the assets to obtain the final goods. In the following, we only illustrate the operations used for representing the Carne PRI production process. For a detailed description of all DSGL operations in the *-chain framework, refer to [8, 9].

asset_create creates a new asset in the SCMS. The supply chain participant who invokes the *asset_create* operation will be the owner and the controller of such asset.

asset_update updates the properties (e.g. position, weight and colour) of an asset.

asset_transform represents the execution of a process which modifies an asset's features and traits, leading to the creation of a new and distinct asset.

asset_pack represents the process of putting any kind of asset into a container. This operation does not change the original asset and can be iterated to store containers into other containers.

asset_unpack represents the unloading of assets from a container. Consumable containers are destroyed by the unpack operation, while non-consumable ones become empty.

monitor checks information on a given asset in a specific period of time and for a specific number of iterations. Relevant information concerning the asset is recorded and then stored.

sell changes the value of the *owner* property of an asset and represents the transfer of the ownership of the asset from one supply chain participant to another. This operation can only be executed by the owner of the asset that is being sold.

give_control is a transaction operation between two supply chain participants and represents the transfer of an asset's control. Hence, this operation changes the controller property of the asset.

Each of the operations above have been implemented within the DSGL representation of the *-chain framework. Each block in the diagram of Figure 1 is translated into a set of operations. The phases ID Assignment, Animal Introduction, Animal Purchasing, Antibiotics and Breeding are shown in Figure 2. The ANAPRI Certification, Cow Evaluation and Cow Transport phases are shown in Figure 3. The Ante Mortem Inspection and the Slaughter phases are shown in Figure 4. The Phases of Post Mortem Evaluation and Carcass Evaluation are shown in Figure 5. The Carcass Maturing and Carcass Transport phases are shown in Figure 6 Sectioning, Meat maturation, and Meat Transport phases are shown in Figure 7 The Grinding, Portioning, Packaging and Selling phases are shown in Figure 8

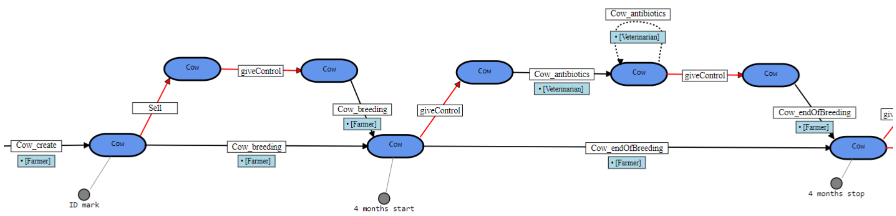

Figure 2: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases ID Assignment, Animal Introduction, Animal Purchasing, Antibiotic Administration and Breeding.

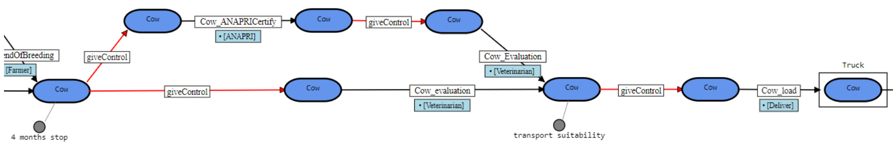

Figure 3: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases ANAPRI Certification, Cow Evaluation and Cow Transport.

3.3. Auto-generated Smart contract

Once the design is complete, the *-chain framework translates the model into smart contract code: each object (e.g. asset or container) represents a different smart contract. Each smart contract

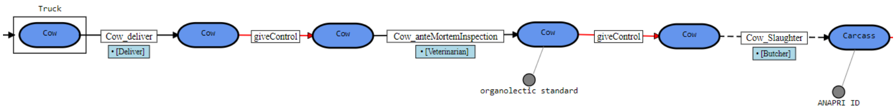

Figure 4: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases Ante Mortem Inspection and Slaughter.

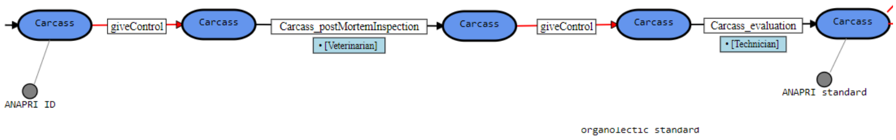

Figure 5: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases Post Mortem Evaluation and Carcass Evaluation.

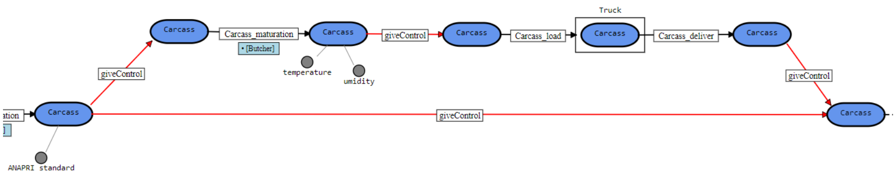

Figure 6: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the optional phases Carcass Maturing and Carcass Transport.

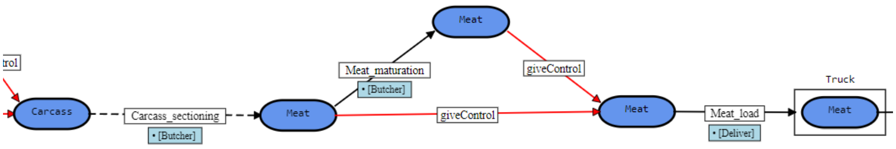

Figure 7: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases Sectioning, Meat Maturation and Meat Transport.

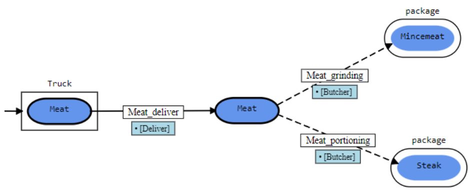

Figure 8: Snippet of Carne PRI Supply chain translation of the Figure 1 schema representing the phases Grinding, Portioning, Packaging and Selling.

contains a data structure representing the history of all the objects of its type, all the operations defined on the schema involving the object, plus the creation, destruction and view operations.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.8.0;
import "@openzeppelin/contracts/access/AccessControl.sol";
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract contract_Cow is ERC20, AccessControl {
    bytes32 public constant Farmer = keccak256("Farmer");
    bytes32 public constant Veterinarian = keccak256("Veterinarian");
    bytes32 public constant ANAPRI = keccak256("ANAPRI");
    bytes32 public constant Deliver = keccak256("Deliver");
    bytes32 public constant Butcher = keccak256("Butcher");

    enum asset_states {Initialized, Sell_ed, Cow_breeding_ed, giveControl_ed, Cow_antibiotics_ed, Cow_endOfBreeding_ed, Cow_ANAPRICertify_ed, Cow_Evaluation_ed, Cow_evaluation_ed, Cow_load_ed, Cow_unload_ed,  Destroied}

    struct asset_Cow_history {      //properties
        string position;
        string ID_mark;
        string _4_months_start;
        string _4_months_stop;
        string transport_suitability;
        asset_states state_of_Cow; //actual state
    }

    struct asset_Cow_struct{
        asset_Cow_history[] Cow;
        uint32 ID;
    }
    asset_Cow_struct[] public store_Cow_s; // MAIN STORAGE

    function Sell(uint _ID, string memory position) public {...}
    function Cow_breeding(uint _ID, string memory position) public {...}
    function Cow_antibiotics(uint _ID, string memory position) public {...}
    function Cow_endOfBreeding(uint _ID, string memory position) public {...}
    function Cow_ANAPRICertify(uint _ID, string memory position) public {...}
    function giveControl(uint _ID, string memory position) public {...}
    function Cow_Evaluation(uint _ID, string memory position) public {...}
    function Cow_evaluation(uint _ID, string memory position) public {...}
    function Cow_load(uint _ID, string memory position) public {...}
    function Cow_unload(uint _ID, string memory position) public {...}
    function create_Cow(uint256 _ID, string memory position, string memory ID_mark, string memory _4_months_start, string memory _4_months_stop, string memory transport_suitability) public {...}
    function view_Cow(uint256 _ID) public view returns(asset_Cow_history[] memory) {...}
    // end of Cow's contract
constructor()ERC20("name","SYM") {}
}
```

Figure 9: A snippet of the generated Carne PRI SC smart contract.

Figure 9 shows the auto-generated Carne PRI SCMS Smart contract. The generated code contains seven contracts, one each defined asset and container in the graphical representation: Cow, Carcass, Meat, Mincemeat and Steak for the assets; and Truck and Package for the containers. Each contract derived from an asset contains a data structure and some functions. The data structure stores the history of each instantiated asset (e.g. "asset_Cow_history"). For each type of asset, its historical data is stored. Each function is a reference of a declared operation in the graphical representation. The control check operations on Roles and Constraints are automatically encoded within each function.

3.4. CarnePRI SCMS costs

Asset	Deployment Cost	USD
Cow	6026082	11.29
Carcass	4959541	9.29
Meat	3710175	6.95
Mincemeat	2609557	4.89
Steak	2609407	4.89
Truck	2618557	4.90
Package	2609544	4.89
erc20	1246301	2.33
Total	26389314	49.43

Table 1
Deployment costs of CarnePRI SCMS smart contracts

We present a more detailed analysis, describing in detail the costs of translation of the model into an automated SCMS. We want to provide quantitative proof of the costs for both the management and the use case, via the ethereum gas cost need for deploying the smart contracts, and for executing a full assets' life circle on the SCMS. Costs are evaluated via the execution of smart contracts through the Remix platform [4]. In the Table 1 we show the deploying cost of CarnePRI SCMS. The cost are espressed in GEWI unit. Wei is the smallest denomination of ether, the cryptocurrency token on the Ethereum network. Currently, 1 GWEI is equal to 0.000002 USD. Using the current live conversion rate. Technically, GWEI is a denomination of ETH – each GWEI is equal to 0.000000001 ETH.

To better characterize the example, we try to estimate the total cost -with respect to the blockchain- of the product's life path, through the various assets, controls, transformations and sales. We calculate at least one execution for each step, taking the longest path in the paths the SC provides. The total cost for a single product it is maximum of 749100 (GWEI), equivalent to 1.49 USD

4. Generated Interfaces

In the following, we show the participant interfaces with its components. The participant interface is one of the auto-generated interfaces using the framework tools: all the auto-generated interfaces are described in detail in [9]. The participant interface manages the assets, allowing registered users to declare which operations to perform. The interface dynamically makes available the functions of the graphical representation, highlighting only those operations possible, in that given asset, for that given user, in the current instance of time.

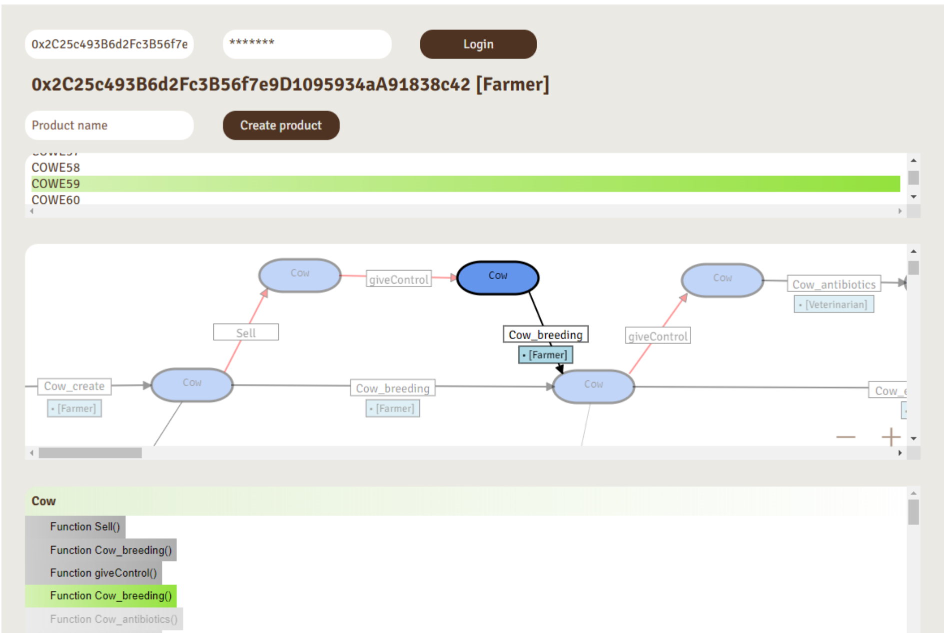

Figure 10: The participant web interface.

[4]https://remix.ethereum.org/

The participant interface presents four different navigation sections. Figure 10 show the edited participant interface for the Carne PRI SCMS. The user logs in to the service in the upper section, retrieving all the information associated with his blockchain address. The second-top section has an input field and a list panel. The input field allows the participant to enter the name or an identifier of the product they want to create. The panel below, then, shows a list of all assets the participant has the right to manage. The main section provides a copy of the supply chain diagram, displaying all the elements as transparent blocks. However, the element representing the state of the asset has no transparency. Finally, below the supply chain diagram is a list of all operations. The operations are grouped into subsets according to the assets. Each item in the list is represented with a transparent label. If the state in which the asset is located has performed the operation, its associated label becomes dark grey, and transparency is removed. All the possible operations following the state in which the asset is located are displayed in green.

5. Conclusion and Future work

In this paper, we apply the *-chain platform to a real use case of the "Pezzata Rossa" – Carne PRI supply chain. By translating the workflow, we show how the framework can represent the supply chain in a DSGL schema. Hence, we show the smart contract automatically produced by the framework. In the final step, we provide the Participant web interface of the Carne PRI supply chain. We demonstrate how easy it is to design the scheme following the guidelines of "Pezzata Rossa" regulation law, enhancing the product's value by the surveyed SCMS and specifying roles and constraints of the different phases. We plan to develop new tools for the platform to translate the graphic representation into different smart contract code languages, e.g. *teal* for Algorand system. We could also add more functions to the framework, allowing users to customize the graphical aspect of the auto-generated interfaces and particular types of outputs. Finally, we want to develop macros of the existing DSGL operations to ease drawing complex diagrams using preset functions.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] ANAPRI, Disciplinare di etichettatura delle carni bovine etichettate dalla Società Agricola Cooperativa "La Rossa Pezzata del Friuli Venezia Giulia", 2018.

[2] H. Stadtler, C. Kilger, Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies, 4th ed., Springer Publishing Company, Incorporated, 2008. doi:https://doi.org/10.1007/978-3-540-74512-9.

[3] N. Trautmann, C. Fündeling, Supply chain management and advanced planning in the process industries, in: K. Waldmann, U. M. Stocker (Eds.), Operations Research, Proceedings

2006, Selected Papers of the Annual International Conference of the German Operations ResearchSociety (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SVOR), Karlsruhe, Germany, September 6-8, 2006, 2006, pp. 503–508. doi:10.1007/978-3-540-69995-8_80.

[4] M. Nakasumi, Information sharing for supply chain management based on block chain technology, in: 2017 IEEE 19th Conference on Business Informatics (CBI), volume 01, 2017, pp. 140–149. doi:10.1109/CBI.2017.56.

[5] F. Corradini, A. Marcelletti, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Chorchain: A model-driven framework for choreography-based systems using blockchain, in: A. Marrella, D. T. Dupré (Eds.), Proceedings of the 1st Italian Forum on Business Process Management co-located with the 19th International Conference of Business Process Management (BPM 2021), Rome, Italy, September 10th, 2021, volume 2952 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 26–32. URL: http://ceur-ws.org/Vol-2952/paper_294a.pdf.

[6] F. Milani, L. García-Bañuelos, S. Filipova, M. Markovska, Modelling blockchain-based business processes: a comparative analysis of BPMN vs CMMN, Bus. Process. Manag. J. 27 (2021) 638–657. URL: https://doi.org/10.1108/BPMJ-06-2020-0263. doi:10.1108/BPMJ-06-2020-0263.

[7] K. Salah, N. Nizamuddin, R. Jayaraman, M. Omar, Blockchain-based soybean traceability in agricultural supply chain, IEEE Access 7 (2019) 73295–73305. doi:10.1109/ACCESS.2019.2918000.

[8] S. Bistarelli, F. Faloci, P. Mori, Towards a graphical dsl for tracing supply chains on blockchain, in: (To appear in) Euro-Par 2021: Parallel Processing Workshops - Euro-Par 2021 International Workshops, Online Event, September 01-03, 2021, Lecture Notes in Computer Science, Springer, 2021, pp. –.

[9] S. Bistarelli, F. Faloci, P. Mori, *.chain: automatic coding of smart contracts and user interfaces for supply chains, in: Third International Conference on Blockchain Computing and Applications, BCCA 2021, Tartu, Estonia, November 15-17, 2021, IEEE, 2021, pp. 164–171. URL: https://doi.org/10.1109/BCCA53669.2021.9656987. doi:10.1109/BCCA53669.2021.9656987.

[10] S. Bistarelli, F. Faloci, P. Mori, C. Taticchi, Olive oil as case study for the *-chain platform, in: M. Pizzonia, A. Vitaletti (Eds.), Proceedings of the 4th Workshop on Distributed Ledger Technology co-located with the Italian Conference on Cybersecurity 2022 (ITASEC 2022), Rome, Italy, June 20, 2022, volume 3166 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 94–102. URL: http://ceur-ws.org/Vol-3166/paper07.pdf.