UNICA UNIVERSITÀ DEGLI STUDI DI CAGLIARI

UNICA IRIS Institutional Research Information System

Automatic Generation of a Blockchain-based Drug Supply Chain Management System

Lodovica Marchesi
Dept. of Mathematics
and Computer Science
University of Cagliari, Italy
lodovica.marchesi@unica.it

***Abstract*—Blockchain technology can improve supply chain management and delivery systems for medical drugs, in several ways. It can improve trust, data integrity, transparency, certify the ownership of drug lots, and track all the phases of the life-cycle of a drug in a non-alterable way. This paper presents a blockchain-based methodology for automatically developing a DApp for drug supply chain management, starting from a description of the specific supply chain involved. It make use of the Python Django framework and is performed by building configurable SCs. Additionally, we describe a simple case study to show how our approach works.**

***Index Terms*—Drug Management, Smart Contract, Blockchain, DApp, Supply Chain Traceability**

I. Introduction

Current supply chain management systems for medical drugs and medical devices are efficient, robust and proven. Blockchain technology cannot improve their efficiency, productivity, robustness – it can, however, improve trust. Present systems are managed by one or a few organizations, data and software executables are controlled by human administrators, and there is no guarantee that data cannot be changed, maybe through direct database access, or that software updates do not introduce bugs.

Blockchain technology, public and permissioned, can improve trust in a number of ways:

- the ledger is managed by many organizations, using a consensus mechanism, and cannot be altered without the consent of all stakeholders;
- the recorded information is immutable through cryptography;
- the software running on the blockchain (smart contracts) is transparent and cannot be altered – it can be substituted with upgraded software in a transparent and shared way;
- the identity of actors sending transactions can be guaranteed through cryptography; in this way, signed transactions can be considered equivalent to signed documents, and can have strong contractual value;
- the recorded information can be made accessible to everyone, or to selected actors, making the system totally transparent.

In applications where trust is critical, adding blockchain registrations can substantially improve and extend the usefulness of the system, at relatively low cost.

This paper presents a general-purpose approach for the drug supply chain management, by proposing a blockchain-based methodology to facilitate and make more efficient the development of such applications.

For specific domains, such as drug management and traceability, a general system has been developed that can be customized for most real supply chains, automatically generating the specific applications (database schema, smart contracts, apps). It is based on general smart contracts and apps interacting with the same smart contracts, which are configured, starting from the description of the specific system to be managed through .csv files. A case study about a simple drug traceability system for a producer to hospital wards is described, to show how this approach works.

Blockchains, even the permissioned ones, are much slower than modern transactional systems, and blockchain software development is more costly, so in our approach we limit blockchain usage for the task in which it excels – trust. All other tasks and computations are performed off-chain by a traditional information system using Python language and a relational database.

a) Contributions: Starting from the study of the domain this work mainly aims to provide the reader with an easy way to develop blockchain applications for drug management, by building configurable blocks to be assembled together, so as not to start from scratch every time.

Our main contributions are the following:

1) A model that supports:
   - transparent and immutable certification of origin and quality;
   - certifications and third-party interventions identified with certainty;
   - management of transformations, manufacturing/storage sites and shipments;
   - secure management of trades and payments.
2) A simple model, extendable to real cases and configurable by choosing only the useful options (e.g., without payment management, or without localization management).
3) The automatic generation of the system (DB, GUI, SC) starting from the specifications, made in a formal way

with a tabular language, or extension of XML or Json.
4) An approach that greatly reduces the implementations costs, and safely build upon a proven framework which was thoroughly security checked.
5) Finally, the proposed approach is evaluated using a case study.

b) Structure of the paper: The remainder of the paper is organized as follows. Section II deals with the background and related work. Section III shows the proposed methodology, highlighting the DApp system architecture III-A, the entities involved III-B and the events managed III-C in such a system; the smart contracts III-D used and the UI III-E necessary to interact with the system. Section IV evaluates the approach by presenting a case study on a simple traceability system for a producer to hospital wards. Finally, Section V draws some conclusions.

II. Background and Related Work

We recall that a decentralized application, also known as DApp, is an application executed by a network of computing nodes, with no supervisor node. In the context of digital ledger technology (DLT), a DApp is stored and executed on the DLT network. In this way, it is transparent, uneditable, deterministic, and decentralized. A DApp is made up of Smart Contracts (SCs), small programs running on every node of the DLT. Typically, it also uses the services of some server components to store information that is unsuitable to be recorded on a DLT, and of a User Interface (UI) that allows users to interact with the DApp. We refer to [1] for a detailed description of DApp architecture and components.

DApps have been developed and used in several fields [2]; here we focus on supply chains for drug management and delivery.

The main purposes of a drug management system are as follows:

- to document in a transparent and secure way all events relevant to drug production and delivery;
- to allow supervisory authorities to audit and certify all stages of production;
- to allow both recordings by human users, and automatic recordings performed by IoT devices;
- to keep track of the drug locations and shipments, possibly including measures of the storage and transport environments;
- to keep track of the actual and past owners of the products.

Current supply chain management systems for medical drugs and medical devices are efficient, robust and proven. DLT technology cannot improve their efficiency, productivity, or robustness. However, it can improve trust because:

- present systems are based on a centralized client-server architecture, run by a single organization;
- data and software programs are controlled by human administrators, so there is no guarantee that data cannot be changed, maybe through direct database access, or that software updates do not introduce bugs;
- the architecture is more vulnerable to Denial of Service attacks.

DLT is decentralized and tamper-proof, as are the programs running on it. They are able to transact between mutually untrusted parties.

In this context, the scholarly literature on the adoption of DLT in the drug supply chain is beginning to emerge. In particular, since 2016 [3], a lot of research has been done on the use of blockchain technology for drug traceability and management systems. For recent reviews of the literature, see [4] and [5].

In 2018, Clauson et al. discussed the challenges and opportunities of using DLT in the health supply chain, and reported also a list of selected companies exploring blockchain for health supply chain management [6].

In the same year, Tseng et al. [7], and Huang et al. [8] proposed two systems, both based on proprietary, permissioned blockchains for Drug supply chain management and traceability. The former system was based on Gcoin Blockchain, a blockchain using the consortium Proof-of-Work approach, which looks no longer used today. The latter system, named "Drugledger" was based on a blockchain developed by the authors which used an expanded UTXO-based transaction model tailored to the drug supply chain [8]. Both works present technical and practical ideas on how the blockchain system could be designed for drug traceability and regulation, but these systems were not actually used in practice.

In 2019, Jamil et al. proposed a drug supply chain management using Hyperledger Fabric DLT to handle secure drug supply chain records [9]. This work focuses on the management of the drug delivery supply chain within a smart hospital and does not deal with drug manufacturing and distribution outside the hospital. It enables hospital personnel and patients to manage a complete, personal drug life cycle in a secure and accountable way. The system was designed and implemented, and its performance was measured to assess its suitability. However, it was still a proof-of-concept application.

More recently, Musamih et al. [10] investigated the challenge of drug traceability within pharmaceutical supply chains, especially focusing to protect against counterfeit drugs. They developed and evaluated a solution based on the public Ethereum platform. They also performed a cost analysis, though based on April 2020 gas cost data. These cost are are well known to be very volatile.

Liu et al. presented a unified five-layer Blockchain and Internet of Things-based platform, providing a decentralized tracking solution for drug supply chains [11]. They used Hyperledger Fabric DLT, and verified the feasibility and efficiency of theit platform using real data from collaborating firms.

Panda and Satapathy [12] designed a system based on Ethereum for the tracking of the medical supply chain. The system was implemented and tested on the Kovan Ethereum test-net. An analysis of its gas consumption was performed, as well as an analysis of its security level. The system was then compared with three systems described in other papers.

Uddin et al. proposed and discussed two DLT architectures, Hyperledger Fabric and Besu, to meet the critical requirements for drug traceability [13]. They then compared these solutions, together with another one based on the public Ethereum blockchain, with respect to key factors such as maturity, ease of development, transaction execution issues, network management, security, integration with third-party services. They were able to identify and discuss several future research challenges that may hinder the successful deployment of blockchain solutions in the drug supply chain.

Based on these research papers, we can assess that the use of a DLT to manage a drug traceability system guarantees:

- Data integrity, immutability and transparency of data and programs recorded on the DLT.
- Identity of the actors, human or IoT, writing data on the DLT.
- Availability and reliability of DLT data, due to their redundancy.
- No centralized authority running the system, so no single point of failure and reduced DDoS attack surface.
- The ownership of drug lots is certified and cannot be forged.
- All the steps of the life-cycle of a drug are recorded, and can be easily monitored and audited,

To the best of our knowledge, and by comparing our work to the others that deal with the use of DApps and blockchain to track the provenance and ownership of drugs, and to prevent frauds, we found out that our is the first paper that proposes the use of a system which can be automatically generated from a description of the specific supply chain involved.

III. Methodology

We studied the drug management production and delivery domain, and we found that most supply chains are similar. The supply chain starts with entering into the system a lot of packed drugs, or of raw materials which are transformed into packed drugs. The packed drugs are then delivered to intermediate distributors, or to end customers. Delivery may be carried out directly or through a forwarder, and can be traced and certified, as it must follow precise guidelines to ensure, for example, the temperature for the correct storage of drugs. The end user is typically a hospital ward or a private consumer who buys medicine from a pharmacy. Events relevant to the process can occur in each of these steps.

Using analysis techniques and object-oriented design, we performed the analysis of the drug management production and delivery domain in order to:

- Determine the actors, concepts, and entities that emerge and recur in this type of systems.
- Determine the relationships between these entities.
- Determine the events of interest for tracking and for the generation of new products (transformations, splitting and merging).
- Design configurable and modular SCs/building blocks, representing the identified entities and relationships, which can then be represented on the blockchain.
- Automatically map the user interface needed to interact with the system, describing the entities, events, and transformations.

In this section we present our approach and we show how the traceability system is then generated automatically, saving time and costs. The entire software process was carried on following the agile approach ABCDE (Agile BlockChain DApp Engineering) [14].

A. Architecture

As a blockchain technology we choose Ethereum, and specifically its permissioned versions such as Hyperledger Besu, because it is the most proven and used DLT. In the future, however, our approach could be extended to other DLTs.

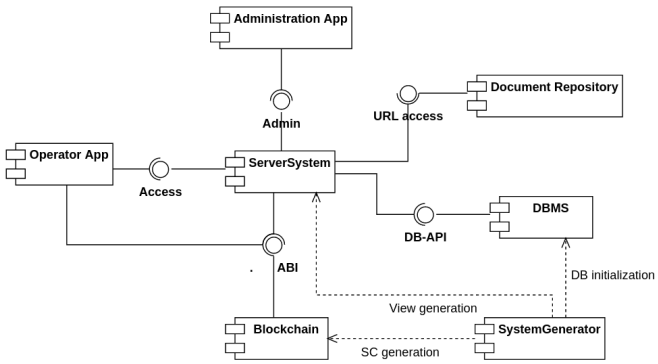

Fig. 1. The overall architecture of the proposed system.

The DApp system architecture follows the guidelines described in [1], as shown in the UML component diagram of Fig. 1. The central component is the server system, written in Python language using the Django framework [15], [16]. This system stores the needed information on a relational database, provides the GUI for the app used by operators, and manages the upload and download of files to and from a document repository. The certifications of all relevant events and transformations are performed using the blockchain. The system generator is a program which reads the system configuration, and automatically generates the smart contracts, populates the database and customizes the user interface.

A more detailed view of the architecture using a UML deployment and component diagram, highlighting the Django components, is shown in fig. 2.

The Django engine and its modules are contained in the Main server. The modules of our system are models and views, written in Python. They are shown in the figure in blue.

The models describe the data structure of the general entities of the system, and the related functions. Upon initial system generation, Django stores these data in the database of the system running in the Database server, and prepare tables to contain the system data at runtime.

The views provide user interface functionality, using HTML templates and CSS files to display information on the mobile

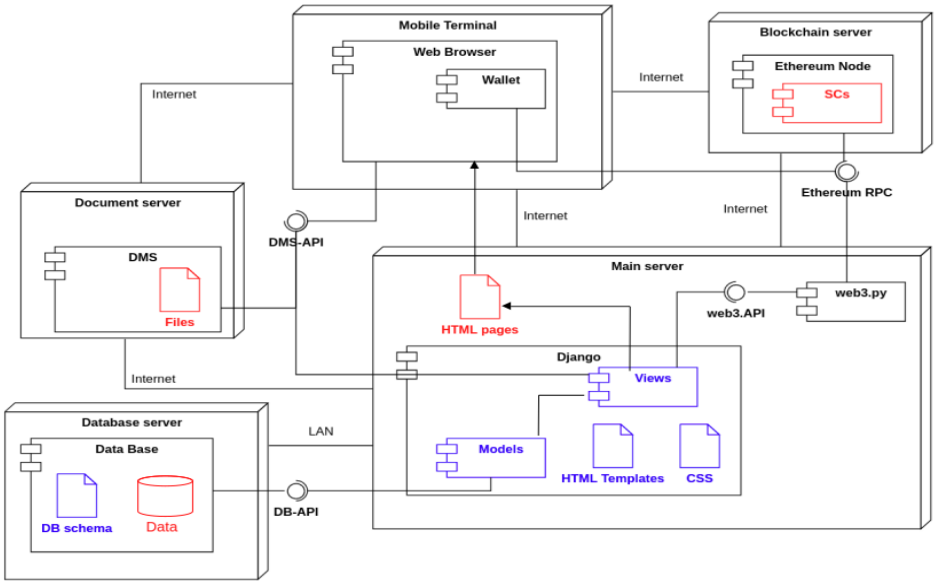

Fig. 2. The software architecture of the proposed system. In blue the components and artifacts common to all supply chains, in red the data that configure and manage a specific implementation.

terminals of the operators, and to get their input, in responsive way. The views are also provided of functions to send transactions to the blockchain, via the web3.py standard interface, and to upload and get documents to and from the Document server.

The generation of a system able to manage a specific supply chain is performed by our system generator which reads the supply chain configuration systems and produces three types of output: (i) the SQL code to populate the database with the supply chain data (organizations, operators, locations, types of products and events); (ii) the Python code to create the smart contracts implementing the DLT part of the system; (iii) The HTML pages and styles to customize the system for specific customers.

The generated SQL and Python code are then executed, and the supply system is created. The related code and data are shown in red in the figure.

During the real operation of the system, the operators input data, which are written in the database and in the blockchain, and upload documents to the document server, recording in the blockchain the link and the hash signature of the document. Also these data are shown in red in the figure.

The mobile terminals of the operators interact with the system through a web browser. The browser also runs as a plugin an Ethereum wallet able to manage the private key of the operator's address and to send transactions to the blockchain from this address.

B. Entities of the system

1) Actors: We define the key actors of a drug management and delivery system. Each actor, individual or device, is identified by a unique address able to send transactions and queries to the blockchain. The actor owns the private key associated with the address, thus being the only person able to send messages from that address. The actors of a drug supply chain are:

- **Operator**: worker in a firm.
- **Auditor**: auditor belonging to a Drug Administration Organization, such as FDA.
- **WarehouseWorker**: worker in a warehouse.
- **CarrierWorker**: worker in a forwarder firm.
- **Apothecary**: pharmacist in charge of drug delivery to hospital wards.
- **HealthWorker**: physician or nurse in charge of giving drugs to patients.
- **IoT Device**: An IoT device connected to the Web, able to send transaction to the system regarding relevant measurements of physical quantities, such as temperature, humidity, etc.

2) Organizations: System-relevant organizations to which the actors belong. They are:

- **IngredientSupplier**: firm producing/delivering raw material to a Manufacturer.
- **Manufacturer**: firm producing drugs.
- **Repackager**: firm repackaging drugs.
- **Distributor**: firm distributing drugs to Pharmacies or to other Distributors.
- **Forwarder**: firm in charge to securely deliver drugs.
- **Pharmacy**: firm/department receiving drugs and selling them to patients, or delivering them to hospital wards.
- **HospitalWard**: hospital ward giving drugs to hospital patients.
- **FinancialFirm**: company guaranteeing fiat money transfers between firms.

- **Public Authority**: public body entitled to audit drug production and management.

3) Other Entities: Other entities involved in a drug management and delivery system are:

- **Location**: physical location of an Organization. An Organization can have one or more Locations.
- **Shipment**: transfer of products from a Location to another, carried on by a Forwarder.
- **RawProduct**: lot of raw material (useful for tracing the industrial process).
- **PackagedDrug**: one ore more packaging of a drug.

Each entity is also associated to a Smart Contract.

C. EventTypes and Events

An event is an entity that occurs over time. The main events managed by the system are:

- **Registration**: initial recording of raw material bought by a Manufacturer from an IngredientSupplier or an external firm (not part of the system).
- **Production**: transformation from raw materials to other raw materials, or to packaged products.
- **Addition**: adding a product lot to another of the same type.
- **Splitting**: creation of a product lot from another of same type.
- **LocationTransfer**: a product (lot) is directly transferred to another location (no tracked shipment).
- **Dispatch**: a product (lot) is dispatched from a location to a shipment.
- **Delivery**: a product [lot] is delivered to a location through a shipment.
- **Conferral**: changing the ownership of a product with no sale (for instance, from an hospital pharmacy to a ward of the same hospital).
- **Sale**: offer to change the ownership of a product through an exchange for money.
- **Purchase**: accepting the sale offer and terms by a buyer.
- **PaymentConfirmation**: confirmation that payment is made, followed by actual ownership transfer.
- **Description**: adding generic data, or a document, to a product; the recording of physical parameters by an IoT device is a kind of description.
- **Tagging**: the official tag with lot id and possibly price is added to drug package or to all packages of a lot; a photo of the package is taken and recorded.
- **Analysis**: the result of chemical, physical or microbial analysis of a lot are recorded.
- **Certification**: adding a certificate to a product by an auditor.
- **State change**: changing the state of a product.

D. Smart Contracts

After defining the architecture of the system and identifying the actors, entities and events related to the process, the system SCs are developed.

We will have a SC for each organization (`Organization`), a SC for each user (`User`) and one for each product (`Product`). The `Organization` contains a mapping with links to all its users/employes and all its products.

A `Product` may correspond to a lot of packed drugs (e.g. lots produced on a certain day and registered), or to raw materials to be transformed into packed drugs.

Each `Product` contains the mapping of events that concern it, registered by a certain User. Each `Event` contains a list of data (encoded in a byte string called ”parameters”), to flexibly attribute data to events.

Each `Product` in the supply chain is also associated with its actual Location. The data of each location are stored in a smart contract `Location`. The `Organization` can have one or more locations so the `Location` knows which organization it belongs to.

A QR code on the final product (packed drug) allows to retrieve the information related to its history. This is accomplished by navigating all the events in (reverse) registration order with their relevant data, including links to documents and their possible hash digest.

For each product, you can also access to the `Organization` that produced it.

In order to optimize the code, we use OpenZeppelin libraries [17] - a toolkit to develop, compile, update, deploy and interact with Smart Contracts. Each SC created derives from `Ownable`, a contract module which allows you to define an owner address of the smart contract and to control that certain transactions can only be performed from this address, thus providing a basic access control mechanism.

Following the ABCDE methodology, we took into account some best practices to avoid vulnerabilities. In particular we systematically applied security and gas saving patterns and best practices [18].

The Smart Contracts used in our system are shown in the UML class diagram in Figure 3.

E. User Interface

The entities and the events of a drug supply chain, as defined in the sections above, are standard and are able to cover most of production processes. Consequently, also the applications enabling their input, editing (when allowed), and retrieving, are standard, and their user interface (UI) can be automatically generated starting from the description of the system.

More precisely, once the events of a specific production process are defined, with their data, constraints and authorizations, the views of our Django system, and the related HTML templates, dynamically generate the HTML code sent to the browser enabling the operator to create and edit the events. Clearly, the style and the appearance of the UI can be customized, but the data input, with all proper checks, does not require further programmer's intervention. An example of the system GUI is shown in the next section, in Fig. 5.

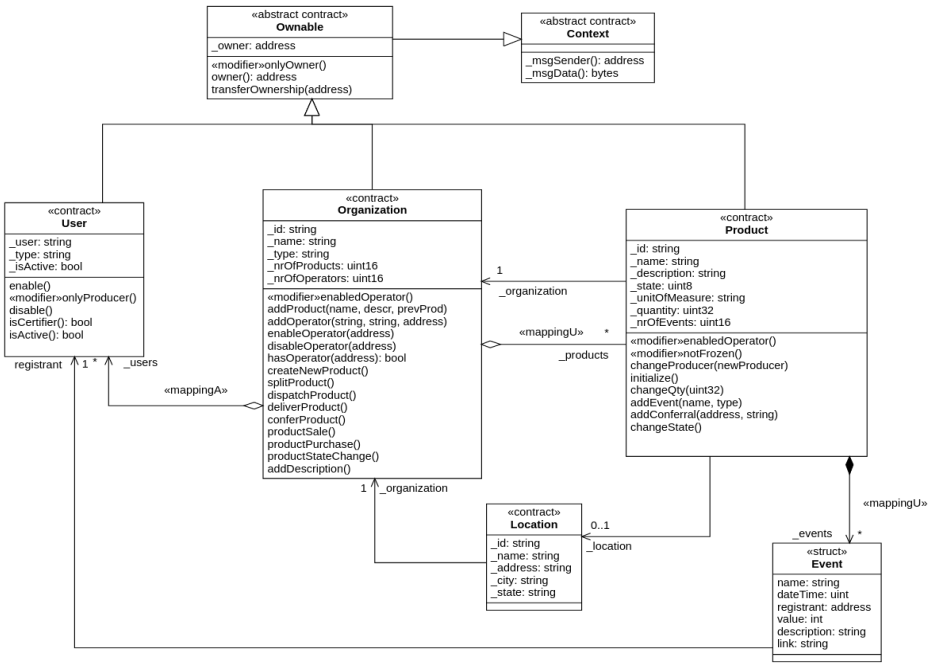

Fig. 3. UML class diagram representing the Smart Contracts used in our system, with stereotypes following ABCDE methodology [14].

IV. Case Study: A Simple Drug Traceability System From a Producer to Hospital Wards

The object of the case study is a simplified DLT drug traceability system able to track drug lots, from the producer, named PharmaCompany1, to a distributor, named DrugDistributor1, and from the distributor to an hospital. A simplified representation of the supply chain is reported in Fig. 4.

The producer produces two types of medicine: an antibiotic and a vaccine, which must be kept below -30°C (-22°F). A forwarder company, named DrugForwarder1, delivers drug lots from one location to another. The distributor and the hospital pharmacy, as well as the forwarder's trucks, are equipped with IoT temperature sensors that record the temperature and write it on the DLT at pre-set time intervals.

The hospital, in addition to the pharmacy, has two wards that will eventually receive drug lots through internal conferral. A government agency, in this case the FDA, audits the whole process, producing periodic certificates, which are also registered on the DLT.

All organizations involved have employees who manage the supply chain and input events in the system.

The supply chain is specified by .csv files, which were then used to automatically generate database, smart contracts and GUI, using Djiango Python framework.

For the sake of brevity, here we report just some snippets of these.csv files, which include their header with the names of the columns.

Among others, we have .csv files for:

- The list of actors. For each actor id, name and type are specified. Note the last actor type, which represents a IoT device:

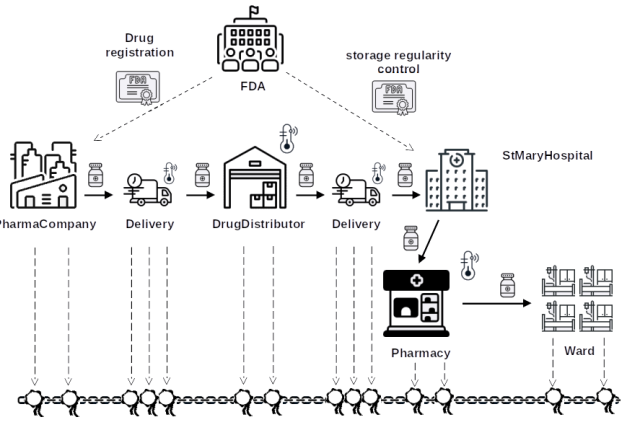

Fig. 4. A typical drug supply chain.

```
id,name,type
OP,Firm Operator,OP
CF,Drug Certifier,AU
CW,Carrier Worker,CW
AP,Apothecary,HE
PH,Physician,HE
NS,Nurse,HE
TS,Temperature sensor,DV
```

- The list of the organizations involved in the process. Note that each department or ward of the hospital that is the final beneficiary of the supply chain is reported as different organization.

```
id,name,type
MA1,PharmaCompany1,MF
DD1,DrugDistributor1
FW1,DrugForwarder1,FW
```

```
FDA,Food&DrugAdministration,GA
HPH,StMaryHospPharmacy,PH
HW1,StMaryHospER,HW
HW2,StMaryHospOrthop,HW
```

- The locations of the organizations involved in the process. Only the relevant locations are reported, for which the actual address of the drugs needs to be recorded. Many organizations do not need a location:

```
id,name,org,address,city,state
L1,MainHQ,MA1,12 Union St,New York,NY
L2,Distrib-3,DD1,4 Scotch Rd,Trenton,NJ
L3,MedicalCtr,HPH,11 Oak Rd,Edgely,PA
```

- For each type of product managed by the production process, we registered its id, name, description, type ("RW" for Raw material, "DR" for Packaged Drug) and and unit of measure:

```
id,name,description,type,unitOfMeas
DR1,CovVax,Covid Vaccine,DR,Dose
DR2,Lovoflox,Levofloxacin antibiotic,DR,Box
```

- The lists of authorized operators, and of IoT devices that can send transactions to the DLT are also needed. Both have the DLT address which the corresponding transactions are sent from. Here we report only snippets of the .csv files. The IoT devices file is reported first, followed by the human users file:

```
id,type,organiz.,SCaddress
TE1,DV,FW1,0xA4bF310ca755e3C2BE2785d05F...
TE4,DV,DD1,0xC63e181e305B34363DC182d79C...
TE5m,DV,HPH,0x90903104617914931E2802B7a...
. . .

id,type,org.,name,surname,email,Scaddress
UM1,OP,MA1,Joe,Ross,jros@phacmp.com,0x...
AU1,CF,FDA,Ann,Zhang,annz@fda.gov,0x...
UF1,CW,FW1,Tina,Cao,caotn@drfwr.com,0x...
AP1,AP,HPH,Mary,Lee,mlee@stmhp.com,0x...
DR1,PH,HW1,Eva,Kant,ekant@stmhp.com,0x...
. . .
```

- The list of relevant event types. Events are always associated with a product. For each event we specify id, name, type (DE for documentation, TR for transformation, LC for shipment-related events), description and a further help string to ease the input of the event; the latter information is not reported:

```
id,name,type,description,help
REG,Registration,TR,Registr. of a new lot,
SPL,Split,TR,Creation of a lot from another
DSP,Dispatch,LC,Shipment of a lot,
DVY,Delivery,LC,Delivering to a location,
CNF,Conferral,DE,Change of ownership ,
SAL,Sale,DE,Sale of a lot for money,
PCH,Purchase,DE,Acceptance of a sale,
DES,Description,DE,Adding a description,
CER,Certification,DE,Adding a certificate,
TER,TempRecord,DE,Recording a temperature,
SCH,StateChange,DE,Changing a lot state,
```

Transformation events cause products to be transformed into other products, or divided into sub-lots. In the shown example, however, the drug lots are directly produced by the initial manufacturer, and then delivered to the distributor, and from it to the hospital. Thus, the only transformation events are the initial registration of a newly produced lot and possible splitting of a large lot of drug packaging into smaller lots. Many events deal with shipping lots using a forwarder company, and with lot conferral among departments of the same hospital. The system has also other input files, specifying constraints such as which type of actor can record which type of event, and others.

The input files were read by the DApp generator and the SQL statements were created to populate the database schema used by Django, as well as the Solidity code to manage the Smart Contracts shown in Fig. 3.

After that, the system administrator created the users' profiles in Django and linked them to the corresponding profiles of authorized operators created in the database, to allow their use of Django framework managing the DApp system.

The DApp GUI is dynamically generated by Django when human operators ask to input a new transaction to the system. Figure 5 shows two screenshots of the application for the input of events, automatically generated.

After these steps, the authorized operators, and the IoT devices, were enabled to send transactions to the system, creating events and managing the drug supply chain.

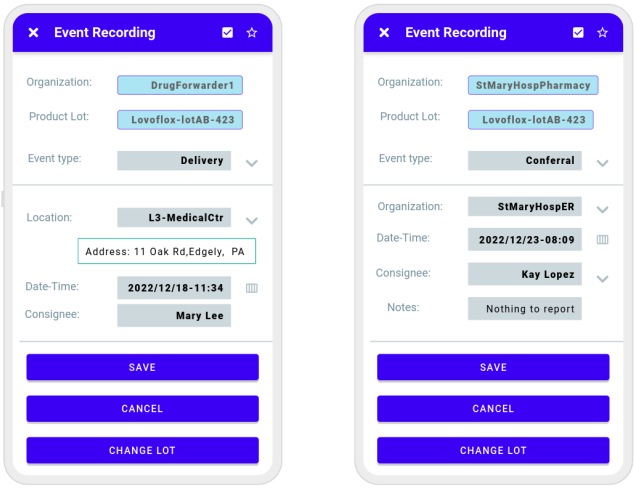

Fig. 5. Recording Delivery and Conferral events.

V. CONCLUSIONS AND FUTURE WORKS

Traceability systems are very important to ensure compliance with production and storage standards, as well as the non-entry of tampered drugs into the supply chain. Moreover, thay can certify the possession of the drugs by the legitimate owners, which is important in the case of expensive drugs. Systems based on DLT and smart contracts for monitoring, integrated with IoT devices, allow a traceability system to occur without a third party which usually is responsible for controlling the drug supply chain, enabling data transparency and indication of the origin of the lot of packaged drug, as well as allowing the end customer to control the quality and genuineness of the purchased drug.

Specifically, the advantages of automatic drug supply chain management using a blockchain are:

- The end customer can be sure of origin, production process, and quality of the product purchased.
- The Authority in charge of production and delivery control is facilitated, and can reduce on-site inspections.
- The Manufacturer can certify in a simple and non-falsifiable way all the steps of a production.
- The Forwarder can control and certify that all shipment parameter are compliant to regulations.
- Development time and cost are reduced, while maintaining a high level of security and trust.
- Blockchain system can also manage contractual transactions and payments; in this case, the system could also be extended to payments, and the transfer of ownership of a lot could also occur together with a token or cryptocurrency transfer.

To the best of our knowledge, this is the first attempt to automatically develop custom DApps for drug supply chain management. It was performed by using a state-of-the-art framework, such as Django, and building configurable SCs to be assembled together. Moreover, we actually built a proof-of-concept system to demonstrate the effectiveness of the approach. Note that this flexible approach may lead to high gas cost in creating and updating SCs on Ethereum blockchain. For this reason, we advise to use a publicly accessible permissioned blockchain as DLT. Future work may include explicit management of payments, using a stable-coin pegged to a traditional fiat currency and the extension to other DLTs.

ACKNOWLEDGMENTS

This work was partially funded by the PRIN 2020 - project, financed by the Italian Ministry of University and Research (MUR), CUP: F73C22000430001, and by the "Analysis of innovative Blockchain technologies: Libra, Bitcoin and Ethereum and technological, economical and social comparison among these different blockchain technologies" project, funded by Fondazione Di Sardegna, oct-2020 to oct 2022, CUP: F72F20000190007.

REFERENCES

[1] L. Marchesi, M. Marchesi, R. Tonelli, and M. I. Lunesu, "A blockchain architecture for industrial applications," *Blockchain: Research and Applications*, p. 100088, 2022.

[2] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[3] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom)*. IEEE, 2016, pp. 1–3.

[4] A. Sharma, S. Kaur, and M. Singh, "A comprehensive review on blockchain and internet of things in healthcare," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 10, p. e4333, 2021.

[5] A. Ghadge, M. Bourlakis, S. Kamble, and S. Seuring, "Blockchain implementation in pharmaceutical supply chains: A review and conceptual framework," *International Journal of Production Research*, pp. 1–19, 2022.

[6] K. A. Clauson, E. A. Breeden, C. Davidson, and T. K. Mackey, "Leveraging blockchain technology to enhance supply chain management in healthcare:: An exploration of challenges and opportunities in the health supply chain," *Blockchain in healthcare today*, 2018.

[7] J.-H. Tseng, Y.-C. Liao, B. Chong, and S.-w. Liao, "Governance on the drug supply chain via gcoin blockchain," *International journal of environmental research and public health*, vol. 15, no. 6, p. 1055, 2018.

[8] Y. Huang, J. Wu, and C. Long, "Drugledger: A practical blockchain system for drug traceability and regulation," in *2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. IEEE, 2018, pp. 1137–1144.

[9] F. Jamil, L. Hang, K. Kim, and D. Kim, "A novel medical blockchain model for drug supply chain integrity management in a smart hospital," *Electronics*, vol. 8, no. 5, p. 505, 2019.

[10] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, "A blockchain-based approach for drug traceability in healthcare supply chain," *IEEE access*, vol. 9, pp. 9728–9743, 2021.

[11] X. Liu, A. V. Barenji, Z. Li, B. Montreuil, and G. Q. Huang, "Blockchain-based smart tracking and tracing platform for drug supply chain," *Computers & Industrial Engineering*, vol. 161, p. 107669, 2021.

[12] S. K. Panda and S. C. Satapathy, "Drug traceability and transparency in medical supply chain using blockchain for easing the process and creating trust between stakeholders and consumers," *Personal and Ubiquitous Computing*, pp. 1–17, 2021.

[13] M. Uddin, K. Salah, R. Jayaraman, S. Pesic, and S. Ellahham, "Blockchain for drug traceability: Architectures and open challenges," *Health Informatics Journal*, vol. 27, no. 2, p. 14604582211011228, 2021.

[14] L. Marchesi, M. Marchesi, and R. Tonelli, "Abcde–agile block chain dapp engineering," *Blockchain: Research and Applications*, vol. 1, no. 1-2, p. 100002, 2020.

[15] D. S. Foundation, "Django Project website," 2023. [Online]. Available: https://www.djangoproject.com/

[16] S. Dauzon, A. Bendoraitis, and A. Ravindran, *Django: web development with Python*. Packt Publishing Ltd, 2016.

[17] OpenZeppelin, "Openzeppelin: Contracts," 2020. [Online]. Available: https://github.com/OpenZeppelin/openzeppelin-contracts

[18] L. Marchesi, M. Marchesi, G. Destefanis, G. Barabino, and D. Tigano, "Design patterns for gas optimization in ethereum," in *2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, 2018, pp. 9–15.