Blockchain: Research and Applications 3 (2022) 100049

Contents lists available at ScienceDirect

Blockchain: Research and Applications

journal homepage: www.journals.elsevier.com/blockchain-research-and-applications

Using blockchain and semantic web technologies for the implementation of smart contracts between individuals and health insurance organizations

Efthymios Chondrogiannis *, Vassiliki Andronikou, Efstathios Karanastasis, Antonis Litke, Theodora Varvarigou

National Technical University of Athens, 9 Heroon Politechniou Str, 15773, Athens, Greece

ARTICLE INFO

Keywords:
Blockchain
Smart contracts
Insurance organizations
Health standards
Semantic web

ABSTRACT

Blockchains and smart contracts are gaining momentum as enabling technologies for a wide set of applications where data distribution and sharing among decentralized infrastructures is required. In this work, we present a distributed application developed using blockchain technologies that allows individuals and health insurance organizations to come into agreement during the implementation of the healthcare insurance policies in each contract. For this purpose, health standards and semantic web technologies were used for the formal expression of both the insured individual's data and contract terms. Accordingly, a fine-grained data access policy was applied for evaluating contract terms on the basis of relevant data captured in healthcare settings. A prototype was implemented involving the development of several different smart contracts for the Ethereum platform as well as the necessary visual environment for accessing them. The developed system validates various features related to blockchain and smart contract features that are briefly discussed in this work, part of which can be mitigated or resolved through the use of a private permissioned blockchain. The application of well-established techniques for potential malfunctions of external services could also boost the security of the system and prevent it from potential attacks.

1. Introduction

Bitcoin revolutionized cryptocurrencies by allowing financial transactions between unreliable users to take place without the need of intermediaries (e.g., central banks), as Nakamoto initially presented in 2008 [1]. Bitcoin is based on a publicly available distributed ledger, known as blockchain (due to its data structure), that contains all the transactions already completed, which are protected using public key cryptography [2] and consensus algorithms [3]. It soon became apparent that the technologies being used (including blockchain data structure and consensus algorithms), could be applied not only to store and protect data, such as the amount of money of each user, but also programs that could be executed from the blockchain network nodes [4] without anyone being able to alter their code (Blockchain 2.0 era). The most well-known platform, which supports the storage and execution of programs (aka smart contracts) using blockchain technologies is Ethereum, which was launched in 2013 [5] and soon claimed a significant cryptocurrency market share. Subsequently, various distributed applications (DApps) that do not belong strictly to the financial domain were developed, taking into account the advantages provided by these technologies in order to cover the unmet needs in each particular domain.

Blockchain [6] is an update-only data source that records all the transactions that are already completed and verified by the rest of the network nodes. The data recorded in the blockchain are publicly available and hence anyone can further examine the information in each block (transparency) and even participate in the blockchain update without being able to alter the transactions recorded so far (immutability). The data linked with each account can be modified only by the respective

* Corresponding author.
E-mail addresses: chondrog@mail.ntua.gr (E. Chondrogiannis), vandro@mail.ntua.gr (V. Andronikou), ekaranas@mail.ntua.gr (E. Karanastasis), litke@mail.ntua.gr (A. Litke), dora@telecom.ntua.gr (T. Varvarigou).

Production and Hosting by Elsevier on behalf of KeAi

https://doi.org/10.1016/j.bcra.2021.100049
Received 21 July 2021; Received in revised form 19 October 2021; Accepted 7 December 2021

user (i.e., real persons or software entities, aka smart contracts) who has the right to transfer an appropriate amount of money to another user and/or change the platforms' state (in the case of smart contracts execution). The unique features provided by the blockchain technologies enable secure communication among distributed entities despite the fact that there is no central authority and have also been adapted for relevant usage in the health domain [7]. In particular, this technology has enabled individuals to control the data produced during their visit to a healthcare entity as well as the secondary use of such data [8,9]. Being able to access the whole medical history of individuals can be beneficial since healthcare providers can provide more safe and tailored therapies based on it. Nevertheless, for many unforeseen reasons, an injury or adverse event may occur, especially in the individuals participating in a clinical study due to the nature of the research itself.

There are some compelling ethical reasons regarding the compensation of injured subjects, including injuries having occurred in the clinical research field [10]. However, the provision of compensation to the individuals[1] participating in a clinical study is still optional in some countries around the world, such as in the United States of America (USA) [11,12]. In many countries (such as the Netherlands), the compensation of individuals for research-related injuries is based on the presence of a medical error and hence the compensation is being tried in court. Countries that follow a no-fault approach (such as New Zealand) provide more direct compensation to eligible individuals [13]. Despite the fact that no-fault systems provide compensation to a greater number of individuals compared to tort-based systems, the individuals participating in clinical trials may still not receive any compensation even in the case of a severe injury [14]. Making provision for some additional private health insurance that could cover a wider range of research-related injuries and assure the immediate healthcare and/or financial compensation of injured individuals, without them having to prove any proof of cause or negligence other than the documentation of the injury itself, could alleviate the aforementioned issues. At the same time, secure sharing and reuse of knowledge are essential. Blockchain technologies provide the means for efficient collaboration among relevant parties (i.e., individuals, healthcare and/or clinical research institutes, and insurance organizations) and the direct and fast compensation of eligible entities without the presence of any dispute.

The computer-based processing of individual's data (e.g., for analyzing and verifying the healthcare data regarding an insurance claim) is another challenging issue due to the plethora of heterogeneity issues across independently designed data sources as well as the semantic distances if they come from different domains, such as healthcare, clinical research, health insurance, etc. For the interoperable representation and exchange of individual's data, various standards have been published so far by international standards developing organizations such as the Health Level 7 (HL7)[2], the Food and Drug Administration (FDA)[3], and World Health Organization (WHO)[4]. Nevertheless, the mismatches among overlapping standards in combination with their poor adoption within the healthcare ecosystem hamper the communication with and impactful processing of the underlying data sources. Moreover, the complex semantics in the biomedical domain (including the ones used in the healthcare and the clinical research domains) pose an additional obstacle for the in-depth analysis of individual's data. Semantic web technologies, including OWL ontologies [15], offer great opportunities towards addressing these issues both in healthcare and clinical research, but also in collaborating fields, such as healthcare insurance. In particular, they can enable and further enrich the development and management of reference models for different aspects of the process, such as the claims, policy terms, and exclusions, as well as the interlinking with data of great interest (such as healthcare data) and their intelligent analysis for faster, more robust, trustful, and reliable service offering.

In this work, we present a distributed application (DApp) developed using blockchain and semantic web technologies that enable individuals to sign a contract with a Health Insurance Organization (HIO) regarding their health status (e.g., potential manifestations of an adverse effect) and have their claims directly and trustworthily validated. For ensuring the integrity of the signed contracts, they will be stored in the blockchain whereas the insured data will be stored off-blockchain. Nevertheless, a fine-grained data access policy would be followed taking into account the semantics of insured data so that the HIO can further process only the one being necessary for the evaluation of contract terms. For the machine-processable representation of both insured data and contract terms, international health standards and codifications in combination with semantic web technologies are used, while the evaluation of contract terms takes place through the use of the appropriate smart contracts and services.

The document is structured as follows. In Section 2 the blockchain and smart contracts along with research studies, injury compensation policies, health standards, and semantic web technologies are briefly described. In Section 3 the distributed application developed is presented along with the approach followed for the representation and storage of both health data and contract terms. The smart contracts developed are presented in Section 4, including a prototype and an example of use. Relevant systems in this field are presented in Section 5, along with the limitations of the approach followed and the technologies used. Finally, in Section 6 we will summarize the main points of this work.

2. Background

2.1. Ethereum and smart contracts

Ethereum is a cryptocurrency and a platform (i.e., a transaction-based state machine) that allows users to develop Distributed Applications (DApps) using blockchain technology [16]. Both Ethereum and Bitcoin rely on the presence of a public distributed ledger (aka blockchain) which is constantly being updated with new transactions. However, the blockchain data structure (i.e., the data recorded within each block), and especially the data recorded within each transaction, is slightly different, since the Ethereum platform can be used not only to store the amount of money that each user account has at their disposal (aka Externally Owned Account), but also "programs" (aka smart contracts) that are totally controlled by the Ethereum platform. These programs are expressed in a predefined low-level code (aka Bytecode[5]) that can be executed by the Ethereum Virtual Machine (EVM) of each node. The smart contracts are often written in a high-level platform-specific programming language, such as Solidity[6] and the Bytecode is being automatically produced by the compiler along with the interface of the smart contract's methods (aka Application Binary Interface (ABI)). For the execution of each method (including the initialization), the user should provide an adequate amount of money; otherwise, the execution would fail and the given amount of the user's money would be lost.

Ethereum (aka Ethereum 1.0) relies on the Proof of Work (PoW) algorithm for the blockchain update (such as Bitcoin). However, the difficulty of the problem has been properly set so that a new block can be added every about 12 s. Due to the significant amount of time and energy being necessary for the execution of this consensus algorithm (also see Section 5.2) in the future releases of this system a different approach

[1] In the clinical research field, an individual (either a patient or a healthy volunteer) participating in a clinical study is often called a human subject. This definition is available at https://www.nia.nih.gov/research/dgcg/nia-glossary-clinical-research-terms.

[2] Health Level Seven International (HL7), available at http://www.hl7.org/.

[3] U.S. Food and Drug Administration (FDA), available at https://www.fda.gov/.

[4] World Health Organization (WHO), available at https://www.who.int/.

[5] Ethereum Virtual Machine (EVM) Opcodes, available at https://ethervm.io/.

[6] Solidity, available at https://docs.soliditylang.org/.

would be used that is based on the Proof of Stake (PoS) algorithm (Ethereum 2.0). According to this consensus algorithm, the more money the users have invested in the system, the less they want to hurt it since they are going to have greater losses. The node that would be responsible for updating the blockchain depends on the amount of money it has invested in the network, and hence, the more money a user has, the higher that user's probability of selection is. When the new block has been successfully added to the blockchain, the user will get back their own money (i.e., the one offered during the blockchain update selection process) as well as a reward, which is often driven by the amount of money (i.e., gas) offered by each transaction. In this way, the network nodes do not spend their time or energy in order to solve a difficult puzzle (such as in the case of PoW), but (only when selected) they focus on testing the validity of existing transactions and preparing the new block.

It should be noted that the PoS algorithm cannot properly deal with conflicts if the participants may continue to expand the two chains (in order to maximize their profit) since they have nothing to lose. This problem is known as "nothing at stake" and can be solved by enabling users to participate in only one of the available chains. Also, there are several variations of the aforementioned process. For example, the blockchain update can involve several users who own a significant amount of the network's money, to first build "their" candidate block and then vote among themselves which block should be added, until there is a consensus (Byzantine fault tolerance PoS). Another approach takes into account the age of the money held by each user when selecting the node that will be responsible for adding the new block to the chain (coin age PoS). This approach allows users with the most money to create blocks more often, but it does not allow them to dominate over other nodes.

2.2. Clinical trials and research-related injuries compensation

Clinical Trials (CTs) typically follow successful animal testing (pre-clinical testing) and provide the means for introducing a new chemical entity (NCE) to the market. The NCE is initially administered to a limited number of often healthy volunteers (phase 1 CTs) in order to study its safety (e.g., determine maximum tolerated dosage). The efficacy of the NCE is actually measured in separate CTs (i.e., phase 2 and 3 CTs) which typically enroll a larger group of individuals, while the true capabilities of the NCE in the real world are measured in phase 4 CTs [17]. Detecting the appropriate individuals for potential participation in the CTs is a rather difficult process, due to the research risk (especially in phase 1 CTs) and strict criteria that they should often adhere to (especially for phase 2, 3, and 4 CTs). The pharmaceutical companies and organizations involved with deriving candidate individuals for participating in CTs provide several benefits, including monetary ones (especially in phase 1 CTs) [18] and opportunities to access new treatments, among others.

Research-related injuries are injuries that occurr to individuals as a result of their participation in a clinical study and may range from minor harm to major or catastrophic injuries [10]. It is generally accepted that the researchers and institutes should minimize the results of an injury through the provision of medical care and/or financial compensation to the individuals (principles of beneficence and justice). However, there are significant differences in the standards, policies, and regulations used across countries for the compensation of individuals for research-related injuries [11,12]. In the USA research institutes are not obliged to provide compensation to the injured research individuals. The compensation of injured individuals is also optional in Brazil and China. On the other hand, in many European Union (EU) countries, provision should be made for indemnity or compensation of an injury or death attributed to a clinical trial. The same also applies to Russia and South Africa. Due to the absence of strict national regulations for the conduction of a clinical trial (CT) and especially the compensation of research-related injuries, each organization can follow its own rules. The only requirement is that researchers should inform their individuals about the presence of any compensation plan in the informed consent and also not use exculpatory language to describe those plans.

The approach that is typically being followed in many countries and research institutes around the world (including the USA, Brazil, Russia, China, and many EU countries) ensures that the individuals would be compensated in case of an injury that occurs as a result of their participation in the CT. In the fault-based approaches, the claimants must prove that the research-related injury occurred due to some negligence, despite the fact that they successfully carried out all their duties. However, it is often difficult to prove the allegations due to the lack of evidence (cause of injury and negligence) and hence the chances to prevail in court are quite limited (less than 30% based on relevant data from medical malpractice cases) [10]. No-fault systems (applied in India and to some extent in South Arabia along with some EU countries) eliminate the requirement of providing negligence [13]. The claimants must only prove that the injury was caused by their participation in the trial. Hence, more injured individuals may be compensated under no-fault than tort systems and the eligible individuals are compensated in a uniform manner for both economic and non-economic losses. The study of data collected from the primary care of the no-fault medical system of New Zealand during the first 4 years following the respective 2005 legislative reforms indicated that 2/3 of the incoming injury claims were accepted [14] with the overall acceptance rate for major, serious, and sentinel injuries being above 90%. Nevertheless, even in a no-fault compensation system, only subsets of medical injuries are eligible for compensation, while the individuals are obliged to abandon their right to seek redress for their injuries through the legal system.

2.3. Health standards and semantic web technologies

Data representation and exchange is a rather challenging topic in the health domain due to the presence of many different human expressions for the same biomedical terms and their complex semantics. For the interoperable representation and exchange of health data in the healthcare and clinical research fields, international standards developing organizations have published several standards with the most widely acclaimed being the ones published by the Health Level Seven (HL7) and the Clinical Data Interchange Standards Consortium (CDISC)[7]. These organizations have made an attempt to standardize different aspects of this field by publishing (a) detailed descriptions of the terms of specific domains of interest (i.e., open/closed sets of terms, classifications or treasures), such as the International Classification of Diseases (ICD)[8] and the Chemical Entities of Biological Interest (ChEBI) [19]; (b) Reference Information Models that could be used for the organization of the data stored or exchanged among relevant entities, such as the HL7 Reference Information Model (RIM) [20] and the CDISC Biomedical Research Integrated Domain Group (BRIDG) model [21]; (c) functional profiles that focus on the functionality that an entity should provide; and (d) message exchange standards for the health domain, such as the HL7 overall, the CDISC Operational Data Model (ODM) [22], and the Digital Imaging and Communications in Medicine (DICOM aka ISO 12052) standard [23]. Nevertheless, the poor adaption of these standards by the different organizations in combination with their complexity and the plethora of mismatches among overlapping competitive standards (even previous versions of standards published by the same organization) limit their benefits.

Semantic web technologies such as RDFS and OWL [15] provide the means for the meaningful description of a domain of interest in terms of classes, properties, and relations among them (aka an ontology [24] in the computer science field) so that it can be directly consumed by a software agent. Despite the fact that ontologies intend to provide an

[7] Clinical Data Interchange Standards Consortium (CDISC), available at https://www.cdisc.org/.

[8] World Health Organization (WHO)—International Classification of Diseases (ICD), available at https://www.who.int/standards/classifications/classification-of-diseases.

explicit specification of a shared conceptualization, the developed ontologies are always less complete and formal than what would be desirable in theory since it is not possible in practice to share the whole conceptualization [25]. Subsequently, a considerable amount of different partially overlapping ontologies have been published so far about the same domains, which have significant differences among them, such as the amount and granularity of the terms included, the axis of classifications used, the organization of their terms, and the ontology development paradigm followed [26]. This prevents the seamless combination and/or use of individual knowledge sources. Nevertheless, the standardization of the language used for the expression of ontologies (OWL-2[9] is a W3C recommendation since December 2012) prompted the development of several algorithms and tools that could be used for bridging the gap between these ontologies and therefore facilitate the interaction among relevant entities. Also, software systems can further process the axioms specified and infer additional information about the elements used for capturing individual data (e.g., detect all pharmaceutical drugs or chemical substances that belong to a particular category) and hence boost the search process.

3. Approach followed

3.1. Distributed application

In this section, we present the distributed application developed using blockchain and semantic web technologies that enable individuals to sign one or more contracts with a HIO by paying the necessary amount of money and directly being compensated when the contract terms are met. In the latter case, the HIO would be responsible for automatically providing the respective amount of money in the healthcare entities for the conduction of relevant tests and therapies, and in some exceptional cases, it could also provide financial compensation directly to the individuals themselves.

A health insurance contract constitutes an agreement between the insured person and the healthcare insurance organization regarding coverage of medical expenses. It specifies the terms for deductibles, co-payments, and co-insurance. It also encapsulates parameters such as duration of health insurance coverage, waiting times, conditions for contract modification, renewal, and termination, benefits (including covered medical services and treatments, threshold levels for payments), and the exclusions of cover, among others. In the rest of this document, we will use the term health contract to refer to the aforementioned agreement with the contract terms so that we can easily distinguish it from a smart contract which stands for a piece of code that can be executed by the blockchain network nodes. Nevertheless, there is a strong relation between them since smart contracts provide the means for capturing the contract terms in such a way so that they can be linked with individuals and healthcare entities for the automatic evaluation of the conditions specified and the direct transfer of money among such entities. For this purpose, the health contract conditions are internally represented in a machine-processable way using semantic web technologies and health standards (analytically described in Section 3.2).

The presented system provides the minimum functionality required for supporting the three different types of users involved in this process, that is, HIOs, healthcare provision entities, and individuals (i.e., potentially insured persons). In particular, it enables HIOs to develop a health contract and upload it to the blockchain platform so that it can be then selected by individuals. The latter can go through the available contracts and sign the one best fitting them, as follows, by paying the corresponding amount of money through their own account. Both individuals and HIOs can examine the contract terms and hence decide if the insured individual should be compensated or not. For this purpose, the sensitive insured data recorded by each healthcare entity is stored off-blockchain but the links to them along with the data access policy are securely stored in the blockchain, so that the individuals have total control over their own data and the applications that allowed to further process them. In this work, particular focus has been given to the overall system architecture and especially to the methodology followed for the expression, singing, and evaluation of contract terms. For the sake of simplicity, we have assumed that there is only one HIO and the insured data come from only one healthcare entity. Nevertheless, the same approach can be also applied for supporting more than one HIO and healthcare entity.

Fig. 1 presents the architecture of the system developed. The system consists of five different smart contracts that enable the different users to control the data linked with their account as well as participate in the healthcare insurance contract management on the basis of their role (including specification, negotiation, signing, and term enforcement). For the interaction of the users with the smart contracts deployed in the blockchain platform, a Graphical User Interface (GUI) was developed. The GUI prompts the users to log in to their account (when necessary for the blockchain update) and accordingly invokes the appropriate smart contract method for retrieving the current blockchain status or updating it. In the latter case, the users should also accept or reject the transactions having taken place (e.g., money transfer for signing a new health contract) by also providing the necessary amount of money to cover the transaction's execution cost. Regarding the storage and filtering of insured data, two independent web services were developed that process that data in the data source and update the blockchain with the appropriate indexes (e.g., data pointers) and events (e.g., individual financial compensation). Both smart contracts and services are described in Section 4 in detail.

The GUI has been developed using a JavaScript framework (in our case, React[10]) and hence it is being initially transferred to the client side (when the user downloads a web page) so that it can be used for the direct interaction of the user with the smart contracts. The health contracts are stored on-blockchain whereas the insured data are off-blockchain. Nevertheless, both of them are tightly linked with the models developed that explicitly specify the meaning of terms (especially the ones residing in the Reference Model) including those ones used for the expression of the conditions that the insured data should satisfy (specified in the Eligibility Criteria Model), and will be described in the following paragraphs in depth. Regarding the reimbursement of the beneficiaries, a separate document was developed (Compensation Plan) using the Reference Model terms that specified (a) the amount of money that the user should receive and/or (b) the laboratory examinations (or generally treatments) that could perform at no cost, when diagnosed with a particular condition and will be also presented in the following sections.

3.2. Data representation and storage

3.2.1. Insured data

For the interoperable representation of the insured data and contract terms, we based our work on existing models and international coding systems. More precisely, the OWL ontology, which we have already developed in the HarmonicSS project[11] for recording the data of individuals with Sjogren's syndrome, was used (Fig. 2a). According to this Reference Model (the design of which was driven by CDISC and HL7 Standards), an individual can be linked with one or more data types, such as Demographic Characteristics, Laboratory Tests, Medical Problems, and Medications. For each one of these data types, we have also defined the parameters of particular interest as well as the terminology allowed for each one of them. To broaden the scope of this model, we used international coding systems to record diseases, pharmaceutical drugs, and

[9] OWL 2 Web Ontology Language, available at https://www.w3.org/TR/owl2-overview/.

[10] React—A JavaScript library for building user interfaces, available at https://reactjs.org/.

[11] HarmonicSS project, available at https://www.harmonicss.eu/.

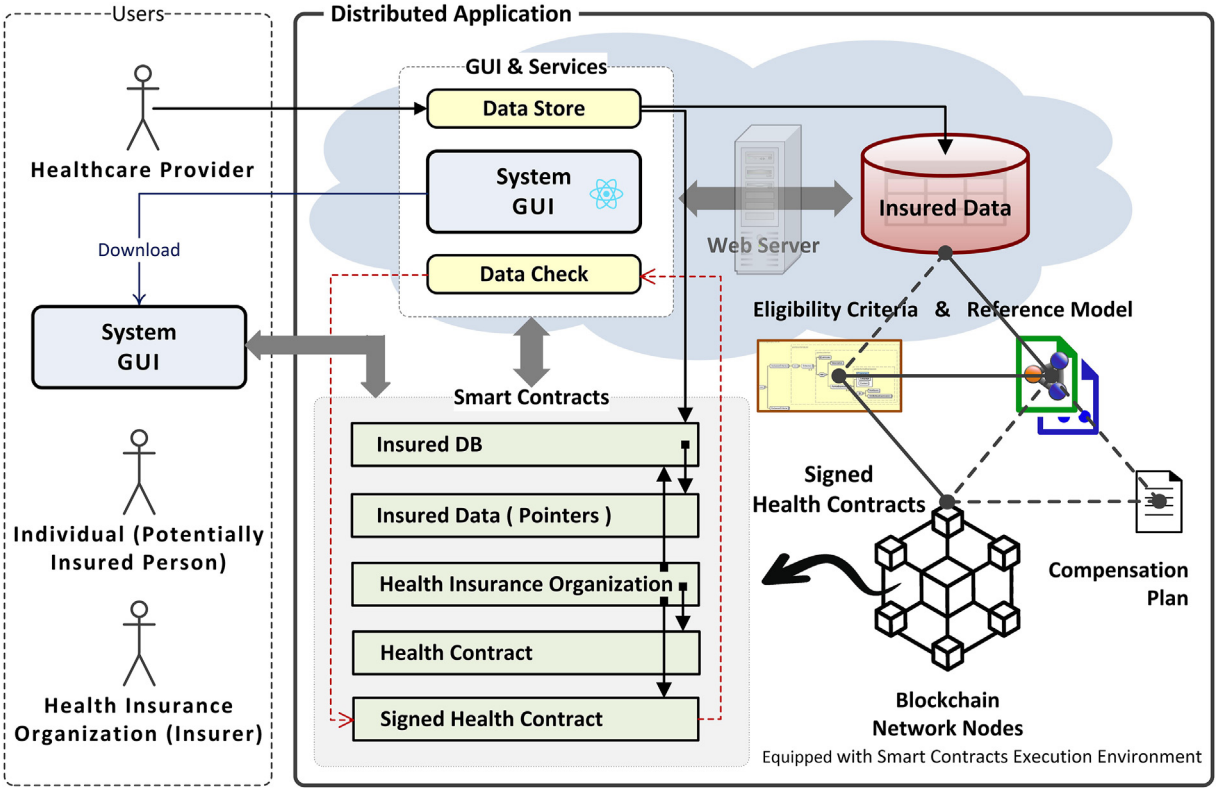

Fig. 1. System architecture.

laboratory tests. In particular, we have used the 10th version of the ICD, the database of ChEBI as well as the Database of Logical Observation Identifiers Names and Codes (LOINC). For each one of these systems, we created an ontological representation of its terms (if not already available) and then included them in our Reference Model (Fig. 2b), in order to have gathered all the information required for the expression of insured data in a single OWL ontology.

Insured data can be easily expressed using the terms of the above Reference Model terms. For example, using the elements existing in the upper part of the above figure (i.e., Person, Unique ID, Diagnosis, Disorder Code, and Date of Onset) along with the appropriate Domain Term (e.g., ICD-10 for disorders), we can express that an individual with a Unique ID "X" has been diagnosed with a particular condition, such as Diabetes Mellitus Type 2 with renal complications (ICD-10 code: E11.2) in February 2018. The insured data collected within each healthcare entity, such as the diagnoses made, the drugs prescribed, etc., were formally expressed using the aforementioned terms and accordingly stored in a separate data source without taking down in the data source the unique ID of the individuals they belong to. All the RDF statements introduced for expressing the insured data of a particular data type (e.g., all the statements about a specific drug prescription including the pharmaceutical drug and the period of administration) were collected, formally expressed using RDF/XML[12], ordered based on the statement characters used and accordingly stored in the data source in a single table row, along with the hash of the above data (Unique ID) and the Data Type they belong to (OWL class). The link between the individual and the particular data recorded in the healthcare entity (i.e., unique id/data hash) was stored in the blockchain along with the broader category they belong to (i.e., the Data Type) and the list of users having access to them so that we can provide fine-grained data access to the respective services based on the type of the data being necessary for the evaluation of the contract terms (e.g., if there is no condition about a laboratory examination completed, there is no need for the service to access the respective insured data).

In the current version of this system, particular focus has been given to the formal expression of insured data and contract terms and the relations among them rather than the security and privacy of the data which have been already described so far [8] (a brief description is provided in Section 5.1). Hence, the operation and validation of the system were based on simulated insured data stored in a central relational database. Ideally, the insured data should be both encrypted and scattered among several entities of a peer-to-peer system (other than blockchain nodes) for security and privacy reasons.

3.2.2. *Health insurance contract terms*

For the precise and accurate description of the contract terms and especially the conditions covered by the contract, we have repurposed the model we have developed for the expression of eligibility criteria (aka inclusion/exclusion criteria) for recruitment purposes in a clinical trial [27]. This model allows for the specification of several inclusion/exclusion criteria which belong to different semantic classes and the formal expression for each one of them using the elements in the Reference Model so that it can be further processed by a software agent. In the context of expressing the terms of a health insurance contract, each health-relevant term constitutes a series of restrictions on insured data parameters, which in turn means that they can be expressed through the elements of the Eligibility Criteria Model and the Reference Model terms. The model is so rich that it can support personalized healthcare insurance contracts, going beyond the simple segmentation-based offerings and meeting the pure individual needs of the insured person. Towards this direction, the model allows for the expression of terms at the level of individual treatments and disorders with flexible temporal constraints. For example, using the two aforementioned models we can specify that the insured individuals will be compensated for Ischemic Heart Disease (presented in Fig. 3) on the condition that they have followed the suggested by the clinical expert treatment with a particular pharmaceutical

[12] W3C RDF 1.1 XML Syntax, available at https://www.w3.org/TR/rdf-syntax-grammar/.

drug (e.g., Nitroglycerin). The conditions that the insured data should satisfy were expressed using JSON (in contrast with XML which is supported in the original version of the Eligibility Criteria Model) so that they can be directly expressed using the Reference Model elements and then translated into the appropriate SPARQL query [28] to be applied to the respective insured data (Section 4.2.1).

The health insurance contracts developed by the HIO were also linked with two different entities, that is, (a) the amount of money that the user should pay and (b) the period of time/validity, both of which were stored in the blockchain along with the contract terms (in Fig. 3, the period of time also included in the formal expression of medical conditions covered). The Eligibility Criteria Model and Reference Model ontology that were used for the formal expression of the health contract terms were stored off-blockchain. However, their unique ID (i.e., the hash of the respective ontologies and schemas) was also included in the blockchain so that we can confirm that the two models (i.e., Eligibility Criteria Schema and Reference Model) based on which data are expressed have not changed. The laboratory examinations and/or therapies covered for each particular condition, along with the amount of money that the HIO could provide to the respective healthcare provider in each case, were stored in a separate file named Compensation Plan. In this file, for each medical condition or disorder, we have recorded the laboratory examinations or therapies covered by the HIO as well as the amount of money that the insured individual deserves (if any). For the formal description of this file, we have used the elements specified in the Reference Model so that we can accordingly examine the tests performed in each particular healthcare entity or the pharmaceutical drugs administered (assuming that they will be provided at the healthcare entity site) and deposit the corresponding amount of money in the healthcare entity's account. Also, in case of diagnosis with some particular conditions, the HIO was committed to financially compensating the respective individual with a predefined amount of money. Generally speaking, an HIO could either provide a fixed amount of money or regularly credit the user account with a predefined amount of money. However, in this work, only the first scenario is supported and hence the eligible user would receive the corresponding amount of money only once, which will be practically transferred to their account when contract terms were met. The hash of this document was also stored in blockchain for verifying that it would not be tampered with.

4. Prototype & background mechanisms

4.1. Smart contracts & external services

4.1.1. Smart contracts

The current section provides the details of the developed smart contracts, their interactions among them, and the external services implemented for dealing with insured data. Fig. 4 presents an overview of the five smart contracts developed. For each one of them, the available methods along with their scope (e.g., public or private/permissioned) are presented. The arrows among them indicate the other smart contracts used for accomplishing their purpose, whereas the symbol existing in the upper right corner of each smart contract (if any) indicates the storage of cryptocurrencies in this smart contract (apart from other data). As mentioned before, the cryptocurrencies in each one of them are totally controlled by the blockchain platform (in our work, Ethereum) and the user can deposit additional money or withdraw them through the appropriate methods developed that are available only to the respective users (e.g., the administrator of an HIO).

4.1.1.1. Insured DB and insured data. In order to deal with the insured data, two different smart contracts were developed. The first smart contract is called Insured DB (IDB-SC). This contract is deployed only once during the initialization of the system (using the account of the administrator of the healthcare entity) and it is used by the insured individual or signed health contract to locate the actual data as well as the Reference Model based on which the data is expressed. The location of the insured data is recorded by the healthcare entity by creating a new instance of another smart contract, called Insured Data, that contains the data unique ID (i.e., hash value), the OWL class of the data, as well as a list with addresses of the users (e.g., the addresses of the particular individual and the HIO smart contract) having access to these data.

4.1.1.2. Health insurance organization (HIO) & health contract terms. For recording the health insurance contracts available by the HIO two different smart contracts were also developed. The HIO smart contract (HIO-SC) is deployed only once (using the account of the administrator of this HIO) and is used to record the basic parameters of the HIO, such as its name and physical address, as well as the address of the user who is responsible for this organization. New Health Insurance Contracts can be developed and deployed by the administrator of this organization. These contracts contain all the necessary parameters, including the formal expression of health insurance contract terms using the technologies described in the previous section. The latter, among others, include the policy holder, the insurance plan (including coverage, compensations, exclusions, formulary), insurance premium, and duration information. In the HIO-SC, the administrator can also (a) deposit the appropriate amount of money that is necessary for the correct function of the HIO and/or (b) withdraw it (or part of it) at any time. The amount of money existing in the HIO-SC should be greater than the amount of money that the HIO should provide to an individual or healthcare entity when health contract conditions are satisfied, and hence, it depends on several parameters, such as the fame of the organization, the number of insured individuals, and the probability that the signed contract terms would be met in the near future among others. Nevertheless, in this work we have assumed that the number of cryptocurrencies residing in the HIC-SC should be greater than a predefined threshold (e.g., 20 ether).

Through the HIO-SC, individuals can identify the available health contracts and sign the appropriate one. In the latter case, the following five steps are taking place: (a) The individual invokes the appropriate HIO-SC method by providing the address of the selected health insurance contract along with an adequate amount of money based on the cost of the particular health insurance contract. (b) The HIO-SC ensures the end user that the organization has an adequate amount of money (in our work, it should be greater than a threshold) and accordingly “accepts” the incoming amount of money. (c) The HIO-SC creates another smart contract with the specific health insurance contract terms based on their formal description along with the particular health insurance contract parameters (e.g., the address of the insured individual and the specific date the contract were signed). (d) Access is granted to the respective insured data to the new smart contract taking into account the contract terms and especially the type of data being necessary for the evaluation of the contract terms. (e) Finally, it records the address of the end user along with the address of the new smart contract and the amount of money provided by the insured individual.

4.1.1.3. Selected & signed health insurance contract. The information about the health contracts signed by an individual is stored in another smart contract, as already mentioned. In this smart contract, there is information about the user's address, the terms of the health contract, the specific duration as well as two additional variables that indicate whether the contract terms were met or not along with the amount that the person or the healthcare entity should retrieve. On condition that the signed health contract has not expired yet, the smart contract uses an external service for detecting the appropriate user data and accordingly examines the contract terms by providing the smart contract address, the IDs of the insured data, and the XML document with the contract terms. The external service is being asynchronously called using the Provable framework[13] and the outcome of this process (i.e., the fact that the

[13] The provable blockchain oracle for modern DApps, available at https://provable.xyz/.

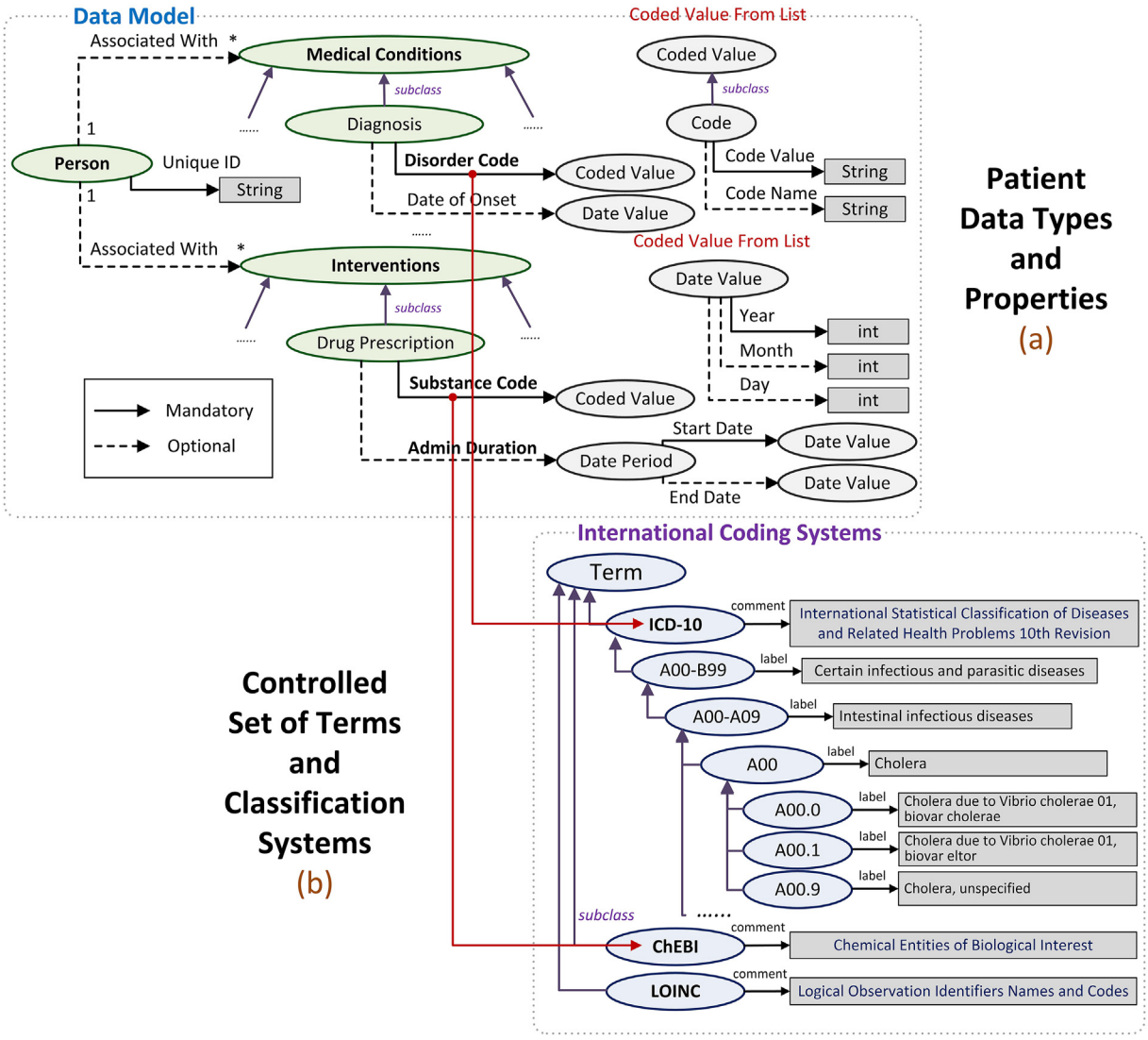

Fig. 2. Reference Model linked with International Coding Systems (a) Data Structure for capturing insured data, (b) Terminology used in each particular domain.

individual satisfies the given conditions or not) is recorded in the blockchain so that it can be accordingly used for the compensation of the individual. In the latter case, the HIO-SC would be responsible for transferring the appropriate amount of money to the insured individual address and/or healthcare entity administrator address (e.g., for covering the cost of a laboratory examination when contract terms are met). It should be noted that the system also enables an HIO to cancel an existing signed contract (on the condition that the contract terms are not satisfied yet) or even delete/destroy an available contract. In the latter case, the system should compensate all the individuals who have already signed this health contract by providing their money along with an additional amount of cryptocurrencies (penalty).

4.1.2. *External services*

In our work, two external services were developed. The first one is being used by the healthcare entity to record the insured data produced during their visit, in the appropriate format and accordingly update the blockchain, whereas the second service is being used for the evaluation of the health contract terms.

4.1.2.1. Data record. This web service is being used by the administrator of a health entity for updating the relational database with the insured data produced during their visit. During this process, the insured data are expressed using the Reference Model ontology terms and accordingly stored in the database along with the class of the data and their unique id (i.e., hash). The unique ID of the insured individual data (i.e., hash) is stored in the blockchain in the respective list so that the respective data can be identified only by its owner or the entities they are allowed to. For this purpose, the web service should already know the address of the insured individual in the blockchain. Alternatively, it should maintain a map with the user address and their local database identifier.

In this work, we have assumed that the data produced by a healthcare entity are already expressed using the Reference Model terms. In a real case scenario, the data produced by the existing systems of a healthcare entity are often stored in a different format. In this case, the tools and systems we have developed in the past [29,30] can be used for the expression of insured data using the Reference Model terms following a semi-automatic process. Using these systems and tools, we can specify the correspondence between the terms of the Local Data Model and the Reference Model terms so that we can accordingly use them for automatically expressing insured data in the appropriate format.

4.1.2.2. Data check. This service (aka oracle) is responsible for the evaluation of the contract terms based on the data provided by the user.

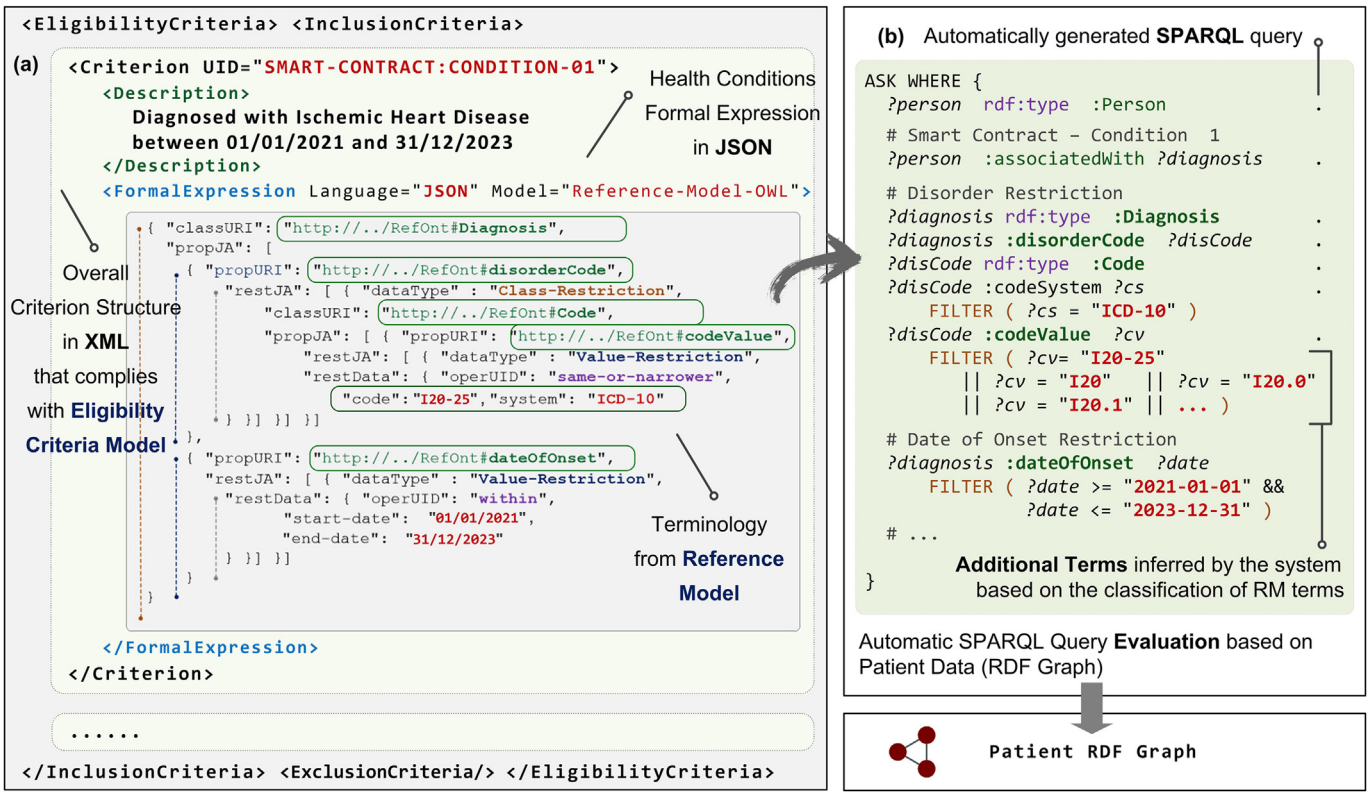

Fig. 3. Health Contract Conditions formally expressed in two different formats (a) Health Contract Conditions formally expressed using the Eligibility Criteria Model and Reference ontology Terms; (b) SPARQL query automatically generated for Health Contract Conditions evaluation purposes.

For this purpose, it initially uses the IDs (hashes) for locating the appropriate insured data and accordingly creates an RDF graph based on the statements specified. The document with the contract terms and, in particular, the formal expression of the conditions that the insured data should satisfy is further processed by the system so that they can be accordingly used to filter the insured data. During this process, the system automatically prepares the corresponding ASK SPARQL query based on the JSON data specified and the Reference Model terms (Fig. 3b) and accordingly applies to the real insured data (i.e., RDF Graph). Since the terminology used for the expression of contract terms may be different than the ones used in the data source, the system takes into account the meaning of terms specified and enriches the generated SPARQL query with additional terms based on the classification of domain terms (e.g., ICD-10 for diseases) in the Reference Model (Fig. 2b).

The outcome of this web service is being recorded by the Provable framework in the blockchain. This framework regularly examines for new requests (raised by existing smart contracts), invokes the predefined web service with the appropriate parameters (i.e., smart contract address, list of hashes, health contract data), and finally provides the results back to the blockchain system by invoking the relevant smart contract method that records the events that occurred. In the current version, we only record the outcome of the process (i.e., the address of the smart contract and a boolean value) along with the amount of money that the caller should receive (either the insured individual or the healthcare entity administrator) based on the laboratory examination performed or therapy being followed (according to the compensation plan). Nevertheless, additional information could be also recorded, such as the date when the process was executed. Also, symmetric key encryption algorithms could be used for securing the messages exchange. Both of them have been deliberately omitted in the current version of the system for reducing its complexity. In Section 5.2.4, there is a description of the approach we can follow for further protecting this service and the whole system as well from potential malicious users.

4.2. Implementation details and example of use

4.2.1. System preparation: infrastructure, components deployment and DApp initialization

For the implementation of the presented system, the Ganache tool was initially used for creating a local Ethereum platform. This tool is part of the Truffle suite[14] and it comes with 10 different user accounts (by default), each of which has 100 “fake” ethers. In our work, we have assumed that the first account belongs to the administrator of the HIO, the second account belongs to the administrator of the healthcare entity, and the third one belongd to the Insured Individual (Fig. 5). The five smart contracts were expressed in Solidity and accordingly compiled and deployed to the local Ethereum platform using the tools provided by the Truffle suite. In particular, the HIO-SC was deployed using the first account and the IDB-SC was deployed using the second account so that the critical methods of such contracts can be used only by the respective entities (accounts). Regarding the implementation of external services, the Apache Jena[15] framework was used for constructing the RDF graph based on the particular insurance data and for the evaluation of health contract terms (using SPARQL). The GUI was developed using the React framework and the web3 library[16] was used for the interaction of the web application with the smart contracts deployed in the blockchain. For enabling web users to connect to the blockchain using their own credentials and accordingly to use the platform for signing a new health contract the Google Chrome web browser was used and in particular, the MetaMask extension[17] (i.e., a Chrome plugin), which allows users to

[14] Truffle Suite, available at https://www.trufflesuite.com/.
[15] Apache Jena, available at https://jena.apache.org/.
[16] Ethereum JavaScript API (web3.js), available at https://web3js.readthedocs.io/.
[17] MetaMask, available at https://metamask.io/.

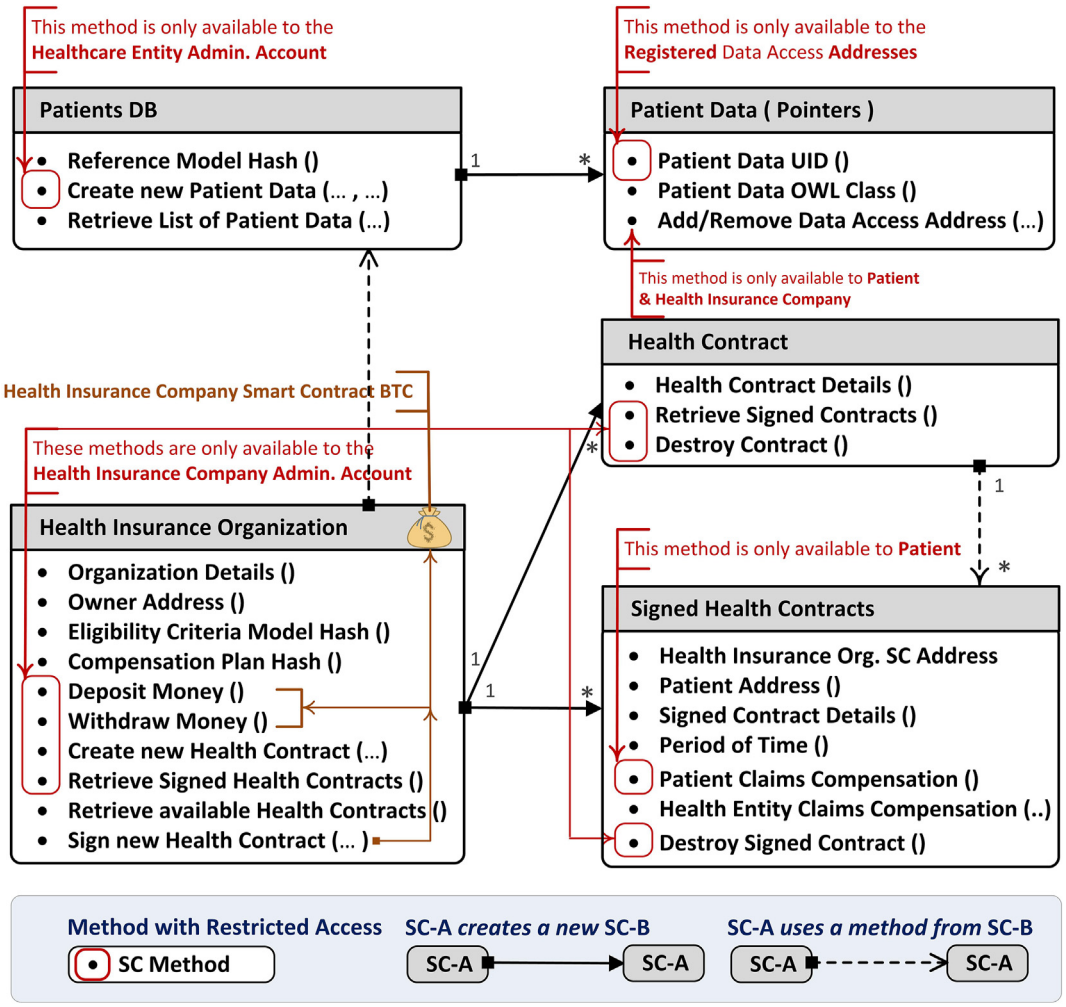

Fig. 4. Smart Contracts (SCs), Internal Methods and Relation among them.

manage their existing blockchain accounts and accept or reject the transactions being performed by the system, on behalf of them.

For the proper functioning of the system, an initialization phase is required, according to which the healthcare entity should update its systems, so that the data recorded for each insured patient during his/her visit is also provided to the platform developed as part of this work, by calling the corresponding web service (i.e., Data Store). In particular, the user's public address should be provided (alternatively, it could be another identifier, which would facilitate locating the individual's account) as well as the respective insured data which should be expressed using the Reference Model terms. Then, the system will not only store the provided data in the relational database, but it will also locate and update the user account on the blockchain platform, with the index of the inserted data (i.e., hash) along with the broader category that they belong to (i.e., Reference Model OWL class URI). For testing purposes, we have developed a desktop application that provides simulated data to the system through the use of the aforementioned service.

During the system initialization, it's also recommended for the administrator of the HIO to define the available health contracts so that they can be accordingly selected by the individuals. For this purpose, the user should specify the health contract terms using the Reference Model terms (i.e., JSON string with OWL ontology terms) as well as the duration of the contract along with their cost, package them in an XML document and then upload it to the blockchain using the GUI. In the current work, each one of the health contracts supported by the system has been manually expressed. Nevertheless, since the formal expression of a criterion is a rather complicated process that presumes additional knowledge of both semantic web and IT technologies (which the HIO representative would most probably not be trained upon) another system can be used which we developed in the past [31] for Visual Query Formulation based on the ontological representation of the data (i.e., Reference Model). This system automatically generates a JSON structure containing the conditions expressed based on the OWL ontology terms (and then uses them for producing the corresponding SPARQL query) and hence it is ideal for our purpose.

4.2.2. *Health contract signing and individual compensation*

Through the GUI developed the administrator of the HIO has logged in to the system using his/her account and transferred an adequate amount of money in the HIO-SC (in our case, 30 ether) which is necessary for the proper functioning of the organization (i.e., guarantee the health contracts signed by the beneficiaries). Through the system's web page, individuals can examine the available health contracts and sign the desired one, paying the required amount of money. It should be noted that an individual can see the amount of money available to the HIO (i.e., smart contract) for testing purposes, but he/she cannot deposit or withdraw any amount of money to/from this account. The system has been additionally set up in such a way that the available contracts can be selected/signed by any user, except the HIO and healthcare entity administrators. For signing a new health contract, the individual logged in

to the system using the third account and provided the necessary amount of money for the contract, along with the appropriate amount of money for covering the transaction cost.

Fig. 5 presents the state of the system after the execution of two consecutive steps, i.e., (a) the beneficiary selects the first contract, which costs 8 ether, and confirms the transaction for the transfer of his/her money to the HIO; and (b) the system (both GUI and blockchain data) updates the amount of money available for each account, as well as the current status of the system presented in the GUI. It should be noted that the amount of money available to the HIO administrator is a bit less than 70 ether. This has happened since the administrator has already transferred 30 ether to the corresponding smart contract (as mentioned at the beginning of this section) and also deployed the two health contracts, paying the transaction cost. For signing the first health contract, the individual should give their consent for the transfer of the appropriate amount of money (i.e., 8 ether) by also covering the blockchain transaction cost. After that, the amount of money available in their beneficiary's account (i.e., the third account) is a bit less than 92 ether, as expected. Also, the amount of money that was initially available to the HIO-SC (i.e., 30 ether) has been increased to 38 ether.

For testing that the beneficiary would be able to get the appropriate amount of money (if any) when the contract terms are satisfied or that the cost of the laboratory examinations in a healthcare entity is covered by the HIO, the data residing in the data source (including blockchain data) should be updated by the healthcare entity. For this purpose, the developed desktop application was used (using the second account that belongs to the administrator of the healthcare entity) with the appropriate individual's data, which indicate the current status of the individual and can be accordingly used by an oracle to verify that the contract conditions are met. Then the beneficiary used the GUI for requesting their financial compensation and the system correctly detected the fact that the beneficiary had the right to get the appropriate amount of money, which was accordingly transferred to their account. After this, the individual requested their compensation again, but nothing happened as expected. Regarding the second scenario, i.e., the coverage of laboratory examinations or therapies, a similar process was followed. However, in this case (a) the process was initiated by the healthcare entity (for testing purposes, by invoking the appropriate smart contract method using the Remix IDE[18]) when the individual performed a laboratory examination; and (b) the respective amount of money was deposited into the account of the healthcare entity's administrator.

The presented prototype indicated that the proposed solution for dealing with the manifestation of undesirable health conditions such as research-related injuries and the direct compensation of eligible individuals is feasible. The blockchain platform (in our case, Ethereum) in combination with the implemented smart contracts and services enables transparent, secure, and fast communication among the three main entities, while the semantic web technologies in combination with widely acceptable health standards in this field further facilitate the interaction among these entities. Nevertheless, there is still room for improvement, in areas such as the GUI, the implementation of smart contracts (e.g., the execution cost of smart contract methods should be reduced), the interaction with external services (e.g., encryption of exchanged data), and others. Regarding the scalability, latency, and throughput of the presented system, it highly depends on the particular blockchain platform used (Section 5.2.3), its number of users, and their amount of activity. In our work, since the Ganache tool was locally used for our experiments with the active involvement of a limited amount of users (less than ten), new blocks are instantly mined, which is far from reality. In the following, section we initially present relevant systems in this field and then discuss the limitations and deficiencies of the prototype implemented, especially focusing on the technologies used and the approach that can be followed for securing the system from potential malfunctions of the external services.

[18] Remix—Ethereum IDE, available at https://remix.ethereum.org/.

5. Related work and discussion

5.1. Relevant systems

5.1.1. Personal data management

Private and public organizations collect and store a significant amount of personal data (including sensitive personal data), which the associated persons have neither enough knowledge about nor any kind of control over how it is being used. In Ref. [8], the authors presented an approach based on blockchain technologies that allows people to have complete control over their data and allow third parties to access it. For this purpose, blockchain technology is used to control access to user data, which, however, is stored outside the blockchain (off-blockchain). In particular, user data are encrypted by using symmetric key encryption and then scattered around the nodes of a peer-to-peer system (which may be different than the blockchain network nodes), while the blockchain contains only a pointer (i.e., the hash of the data) to them. Another entity (e.g., an organization/service) that would like to access the data should first consult the blockchain to detect the "link" to the data. This way, the persons can have complete control over their own data and the entities using them, while they can change their data access policy at any time.

Patient data produced during a person's lifecycle (e.g., laboratory examinations performed, drugs prescribed, diagnoses made, etc.) are also scattered across various organizations and institutions. The collection and analysis of the patients' history can be beneficial for providing better health services to the patients themselves as well as in-depth study of the factors affecting a specific disease, and hence, enriching the available knowledge about the problem and improving the health policies being followed. In Ref. [9], the authors presented a system (MedRec) for organizing patient data and controlling access to it, taking into account blockchain technologies. The development of this system has been influenced by the work [8] and has been implemented using cutting-edge technologies such as Ethereum and smart contracts. In particular, for their system, the authors implemented three different smart contracts, which allow users to record a unique ID (such as their name or social security number), the description of their own personal data, and the users/entities that can access them. User data collected by different organizations are stored in separate databases (e.g., relational databases), while in the blockchain resides only the SQL query that provides access to the respective patient data, which are available to a specific network user with the given address. Additionally, the summary of the user data is recorded in the blockchain for security reasons. The user can control the data accessible to an entity by modifying the SQL query (and the summary of the data as well). Nevertheless, the systems can still locally store the retrieved data and hence use them for their purpose, even if the data access policy has been changed and hence they should not be allowed to use the respective data anymore.

5.1.2. Health data monitoring

Remote patient monitoring is constantly gaining ground. For this purpose, the patient is often equipped with wearable or implanted electronic devices, which monitor their condition in real time and transmit their measurements to a master device, such as a mobile phone, while in some cases, they can even take certain actions (e.g., try to change the patient's behavior) with the ultimate goal being the avoidance of dangerous or life-threatening situations. In Ref. [32], the authors presented a system based on blockchain technologies, for the effective management of patient data stemming from sensors and the provision of an immediate response for triggering the necessary actions, when required. For this purpose, patient data is stored in separate databases, while a blockchain records only the events that take place and not the sensitive personal data of patients, so that it is compatible with the provisions of the Health Insurance Portability and Accountability Act (HIPAA).

For the needs of this system, a private blockchain has been used, in which the chain can be updated from selected nodes. Also, two different

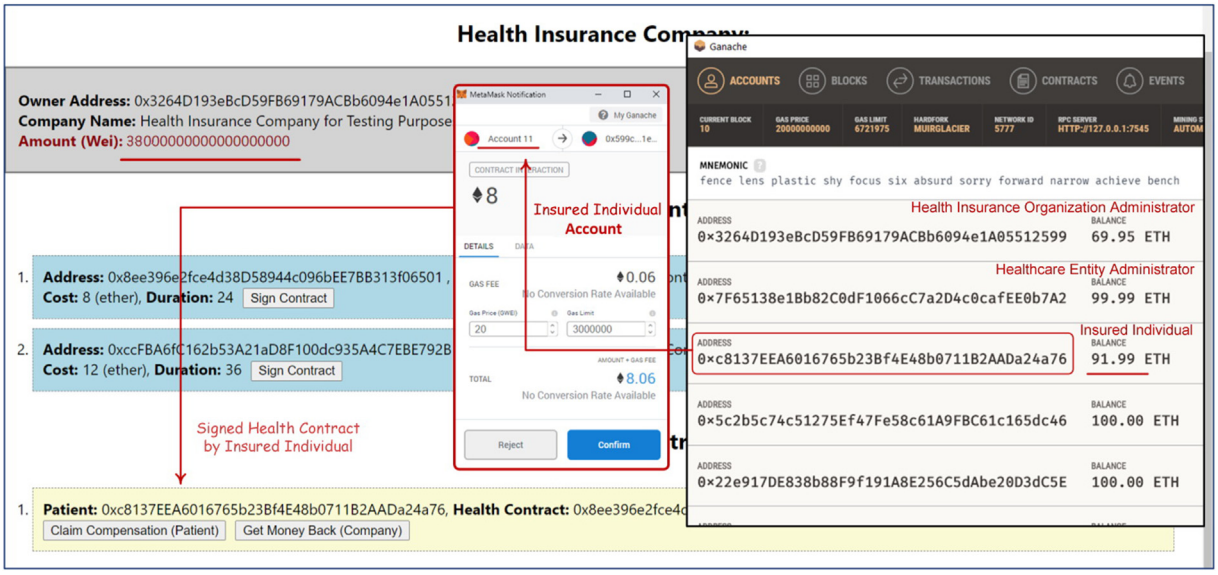

Fig. 5. A screenshot of the GUI of the DApp after signing of a health contract along with the accounts balances.

smart contracts have been implemented, which were written in the Solidity language and can be executed by the Ethereum platform. The first contract acts as the main device's point of contact with the blockchain and it is common to all users. This in turn undertakes to create the necessary smart contracts (based on the factory pattern [33]), depending on the type of user data. These smart contracts are responsible for checking the user's status and accordingly taking the necessary steps. For this purpose, the patient's data (stemming from the sensors, including their normal values) are provided by the service deployed in the main device to the appropriate smart contract (as a parameter in the respective method) which is responsible for examining these values. In case their value is abnormal, this fact is recorded in the blockchain and then the appropriate clinical experts are to be informed. The actions to be performed (either automatically or after contacting the clinical experts) are also recorded in the blockchain. However, the given user data are stored in a separate data source. In this way, the system ensures the users that the important events would be properly handled. However, it is tightly linked with the service running on the master device.

5.1.3. Other systems

The aforementioned systems are distributed applications running on top of existing publicly available blockchain platforms (such as Ethereum), and hence the sensitive patient data are deliberately stored off-blockchain (such as in our work) while they can be further protected through encryption and distribution around the nodes of a peer-to-peer system (in our case, this is part of future work). Alternatively, the data could be recorded on the blocks of a private blockchain on the condition that only the respective users have access to them. This approach was applied by the Medicalchain[19] through the development of a private blockchain platform using Hyperledger (briefly described in Section 5.2.3) along with the appropriate blockchain applications. This approach does not depend on external services since the data are directly incorporated in the blockchain. On the other hand, the patient data are exposed to entities that may try to harm the system by exploiting the limitations of the technologies being used. Another category of systems encompasses the ones that have been developed (often entirely) from scratch, by taking into account the existing blockchain principles, such as the system presented in Ref. [34]. The whole approach applied by these systems is rather interesting since they make an attempt to overcome some of the blockchain limitations. Nevertheless, they are often quite complicated since they need to implement several components, including the blockchain, consensus algorithms, and smart contracts.

5.2. Discussion on limitations and vulnerabilities

5.2.1. Blockchain limitations

In the above systems (especially the ones described in Sections 5.1.1, 5.1.2 and the system that we implemented), the transactions taking place are recorded in the blockchain. All users of the network can keep a copy of the blockchain and consequently access the data recorded in each block, as well as participate in the blockchain update. Given the absence of a central authority to control the system and the transactions taking place, the user's trust in the system stems from the system architecture, the data encryption scheme, and consent algorithms used. The above ensure that the user's transactions recorded in the blockchain are publicly available (transparency) and cannot be changed (immutability), and that money cannot be spent twice (double-spending problem). However, under certain circumstances [35] the above restrictions can be detoured putting the whole system into danger.

Two of the most well-known consent algorithms are the PoW and PoS. In both cases, users (a single one or a synergy among them) can alter or damage the blockchain on the condition that they hold more than 51% of the computing power or currency of the network, respectively. The private key that is being used by each of the users for securing their transactions can be detected by malicious software (based on the data already available in the blockchain, such as the digital signatures) when less than enough randomness has been used during its creation, and hence control over the users' data can be taken. Also, due to the significant amount of time required for the introduction of a new block with completed transactions (especially in the case of PoW), malicious users may attempt and manage to spend their money more than once during that time. Regarding the privacy of user data, anyone could analyze the transactions completed and locate the money or data linked with each account and, in exceptional cases, even reach the real users, since blockchain ensures pseudonymity, but not unlinkability to real entities. However, these kinds of problems can be mitigated through the use of more than one address for the same person. Last but not least, the systems can be used for carrying out illegal transactions, such as retrieving money from a Ransomware attack.

[19] Medicalchain, available at https://medicalchain.com/.

5.2.2. Smart contracts vulnerabilities

Ethereum smart contracts are programs executed from the Ethereum network nodes which result in changing the state of the platform. These programs are expressed in a low-level language known as Bytecode, which is a Turing-complete programming language supported by the EVM. However, smart contracts are often written in a high-level language, such as Solidity, and the "executable" Bytecode (and its interface, aka ABI) is produced during their compilation process. The fact that Solidity is a very expressive language (e.g., it supports branching and iteration) allows developers to write quite complex contracts to meet their needs. Developers can also create and use other contracts, despite the fact that they may have been developed by another IT expert. The freedom provided through this language in the expression of contracts and the execution of their methods may put the users' money and/or data stored in the smart contracts in danger. More precisely, potential intruders can create other smart contracts that invoke the methods of the existing ones and take advantage of their vulnerabilities to steal money and/or data from the accounts of the users coming into contact with such DApps.

The authors of Ref. [36] have organized the security vulnerabilities of smart contracts into three broad categories, depending on the level at which they are located. The first category includes those issues coming from the expressivity provided by the Solidity language. Solidity provides three different ways in which another smart contract can be called, each of which behaves in a slight different way when exceptions occur. Also, the execution of the programs requires a considerable amount of money (aka gas/fuel), which may cause a program to fail when not enough has been provided. In addition, poor design of a method can lead to incorrect system operation in the event of a method recall/re-entrance, while the private data of a smart contract (i.e., those that must be kept secret, even for a short time) can in some cases be detected. The second category includes those issues linked with the execution of the Bytecode by the EVM. The internal stack used for the storage and execution of the codes is limited in size and hence, when exceeded, may cause a method to fail which may in turn cause problems in the data recorded by the smart contract. Code weaknesses detected cannot be easily fixed (smart contracts recorded in the blockchain cannot be removed or replaced by new ones), especially when the system has not been properly designed so that the replacement of some of its components is feasible. In such cases, it may be necessary for the whole system to be rolled back to a previous stable state and the necessary changes in the used data structures and algorithms (including the smart contracts) to be made (aka hard fork). The third category includes the weaknesses related to the updates of the blockchain. Since the exact time when a block is included in the blockchain is different than the time when the method/code starts being executed, the smart contract cannot know the exact state of the system at that time, which may lead to several problems. Also, for producing random numbers and/or calculating time intervals (when necessary) the smart contracts are often based on the timestamp of the last accepted block, which can be influenced by potential intruders. The above is just a short introduction to smart contract deficiencies. We prompt the users to read the aforementioned paper where concrete code examples that simulate widely known attacks (e.g., the DAO attack) can be found.

5.2.3. Blockchain platform architecture limitations

Bitcoin and Ethereum are based on the order-execute architecture for transaction management [37]. According to this model, tasks are sorted and then performed serially, one after the other, when creating a new block. Consequently, the serial execution of the individual functions of smart contracts is directly affected by the execution of the remaining transactions. This can be easily understood if we consider the existence of a poorly designed smart contract, which "enters" into an endless loop. In the case of Ethereum, this problem is addressed through the detailed costing of the individual orders, so that the continuous execution of a program, like the one mentioned before, is automatically interrupted when all available money (gas) is spent. In this manner, the system is protected from such problems (intentional or not). However, this approach significantly increases the total amount of time required for blockchain updates, thus affecting the system throughput.

Moreover, both the Bitcoin and Ethereum platforms are currently based on the PoW algorithm. This consensus algorithm requires the consumption of a significant amount of energy (especially in the case of Bitcoin), which is practically unnecessary since only the effort (and hence the energy spent) of the node that will manage to solve the problem first will be actually spent on purpose, while the remaining nodes will eventually abandon this effort. Also, the significant amount of computing power required for solving such a difficult problem on time, along with the fact that the problem can be easily parallelized and hence split among different entities, resulted in the introduction of groups of miners (aka mining pools) which work together and share their profits (in case of success). Consequently, the chances of independent network nodes winning the process are quite limited, while the collaboration among a large number of nodes and/or other mining pools increases the chance of getting control over the network and the blockchain with undesirable effects.

Also, the way that these systems work, and in particular the achievement of consensus, is based on the fact that all network nodes should have access to the transaction data, which limits the applications that can be developed by using these platforms, since in some cases, only certain entities should have access to the respective data (confidentiality). For this purpose, encryption algorithms or zero-knowledge proof methods can be used, which further increase the complexity of the systems and require additional calculations and, therefore, computational resources. In addition, the programs in the blockchain must be deterministic (i.e., always produce the same result), so that the independent execution of the programs from the different network nodes always leads the system to the same state. To avoid writing non-deterministic programs (for example, directly or indirectly call a function that returns a random number), these systems do not allow the usage of a general-purpose language, such as C or Java. Instead, the users should express their smart contracts in a language that is specific to each system (e.g., for Ethereum, the use of the Solidity language is recommended), which requires additional effort from the IT experts, while the programs developed may, in some cases, behave differently than their developers expected (in comparison to when using general purpose programming languages).

The above issues led to the creation of a different model for the execution of transactions and reaching consensus, known as execute-order-validate, which "breaks" the process into three independent steps (i.e., execution, classification, and control) that can be performed by different entities. This approach enabled the parallel execution of transactions and then their classification, taking into account the interdependencies detected during their execution. This methodology has been used in the Hyperledger Fabric platform [37], which provides some freedom during the configuration of the blockchain platform (in comparison with Bitcoin or Ethereum). Using this platform, a system can be created in which the nodes participating in the blockchain updating process must have the appropriate permission (aka permissioned blockchains), in contrast with the systems that we have examined so far, where any node can participate in this process (aka permissionless blockchains). These systems are ideal for those cases in which there is a requirement for the participation of certified organizations, which have a common purpose (e.g., healthcare entities and HIOs), but do not trust each other completely. The fact that the users participating in the blockchain updating process are limited, enables the administration to "relax" the consensus algorithms used with a direct impact on the amount of time, computing resources, and energy required for the blockchain updates. Last but not least, it should be mentioned that this platform also allows users to write their own smart contracts, which are called chaincode. However, these programs must be written in a standard, general-purpose programming language, such as Go[20] (a functional programming language designed by Google), and hence the errors that occur during the

[20] The Go Programming Language, available at https://golang.org/.

expression of a program are limited in comparison with a less well-known platform-specific language.

5.2.4. Externals services (aka oracles) and external data sources

Smart contracts are based on the data already recorded in the blockchain. Oracles provide the means for introducing additional information in the blockchain (e.g., the current exchange rate of a currency) so that it can be accordingly used by smart contracts. On the other hand, when using oracles the proper functioning of smart contracts is highly dependent on the correct functioning of these external services. Hence, the protection of the external services from widely known attacks (e.g., Denial-of-Service Attack, Man-in-the-Middle Attack) is essential. Also, it is necessary to develop the appropriate blockchain code that further processes the data provided by the respective external web services for detecting potential service malfunctions on time (e.g., service being temporarily unavailable, provided data seems to be fake, etc.).

External services are critical for the correct functioning of a distributed application, such as the one presented in this work, which can be further protected through the application of existing blockchain techniques in order to reach a consensus on the data provided by external services before being accordingly introduced in the blockchain. In particular, the same service could be deployed (or even new variations that provide the same functionality could be developed) in different locations and called randomly from the blockchain network nodes, and their outcome could be included in the blockchain on the condition that the majority of nodes agreed on this. However, this approach can be implemented on the condition that the independent calls for the services using the same input would always provide the same output. This restriction is satisfied in our case since the evaluation of the contract terms is based on the data recorded in the data source. In our current approach, the data are placed in only one data source (for the sake of simplicity), and hence the system can be still manipulated by a potential intruder. Nevertheless, the encryption and distribution of data across different nodes in conjunction with the presence of several replicas can limit the possibility of system malfunction.

The insured data should be also expressed in a machine-processable way (e.g., stored in a relational or graph database such as Neo4j[21] or GraphDB[22]) so that they can be further examined by software agents. Nevertheless, for being able to communicate with the data source (as well as any other different data sources), the insured data should be expressed using the Reference Model terms presented in Section 3.2. For this purpose, tools and mechanisms can be used for mapping the existing terminology with the one specified in the Reference Model and then automatically express the insured data in the appropriate format, as already mentioned in Section 4.1.2. This is feasible, but in case the meaning of terms is slightly different than the one specified in the Reference Model, there would be some information loss during this process, which may also affect the validity of the response provided by the service. Also, a large portion of insured data is often partially structured or even unstructured which makes their analysis and expression by using the Reference Model terms rather difficult. In this case, the application of machine learning techniques (e.g., text analysis using word-embeddings, image processing, etc.) could alleviate these problems to some extent and help retrieve the data needed. Nevertheless, the outcome of such techniques is not always correct. Hence, for the proper functioning of the services, it should be made sure that the corresponding data are formally expressed in the respective data sources in such a way that they can be mapped to the terms of the reference model. In this manner, the system will be able to guarantee that any events of interest (if present) are correctly identified by the respective entities (i.e., healthcare entities) so that the contractual obligations can be inspected and respected by both individuals and HIOs.

[21] Neo4j Graph Data Platform, available at https://neo4j.com/.

[22] GraphDB, available at https://graphdb.ontotext.com/.

6. Conclusion and next steps

In this work, we presented a distributed application which is based on blockchain and semantic web technologies and allows users to come into agreement with an HIO, regarding their health status and their compensation in case of an injury. Health contracts signed on this basis are stored in the blockchain so that their conditions cannot be changed. Nevertheless, the evaluation of contract terms based on the data collected by the healthcare entities is being performed by an external service (aka oracle) that can retrieve and examine the data linked with each individual. The user has still the total control over their data since the data access policy is being stored in the blockchain while the semantic representation of insured data using health standards and semantic web technologies enables the systems to grant temporal access only to the respective insured data (i.e., the ones being necessary for the evaluation of the health contract conditions) taking into account the meaning of health contract terms. The outcome of this service is also stored in the blockchain so that the individuals or healthcare institutes can be directly compensated by the HIO when conditions are met. For this purpose, in the presented prototype, the system allocates an adequate amount of money that depends on the number of individuals who have signed a health contract so that the HIO can directly compensate the insured individuals when contract terms are met. For increasing the security and privacy of insured data, the interaction of the smart contract with the external services should be further protected using state-of-the-art security and privacy mechanisms. Also, additional information could be recorded about each external service call, for detecting potential malfunctions and threats (e.g., fraud detection). The application of the core ideas of blockchain could be also used for reaching consensus among independent developed services that provide the same functionality, before the blockchain update. Last but not least, for reducing potential intruders a permissioned blockchain could be used in future releases of this system for verifying the identity of the users coming in contact with the system, and hence reducing the chance of malicious actions.

Declaration of competing interest

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

References

[1] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, Available online: https://bitcoin.org/bitcoin.pdf, 2008. (Accessed 19 July 2021).

[2] W. Diffie, M. Hellman, New directions in cryptography, IEEE Trans. Inf. Theor. 22 (6) (1976) 644–654, https://doi.org/10.1109/TIT.1976.1055638.

[3] M. Castro, B. Liskov, Practical Byzantine fault tolerance, in: Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI'99); 22–25 Feb 1999; New Orleans, LA, USA, USENIX Association, Vancouver, BC, Canada, 1999, pp. 173–186.

[4] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, NIST Interagency/Internal Report (NISTIR), https://doi.org/10.6028/NIST.IR.8202, 2018.

[5] V. Buterin, Ethereum white paper, GitHub Repository 1 (2013) 22–23.

[6] Z.B. Zheng, S.A. Xie, H.N. Dai, et al., An overview of blockchain technology: architecture, consensus, and future trends, in: 2017 IEEE international congress on big data (BigData congress); 25–30 Jun 2017; Honolulu, HI, USA, IEEE, Piscataway, NJ, USA, 2017, pp. 557–564, https://doi.org/10.1109/BigDataCongress.2017.85.

[7] T.T. Kuo, H.E. Kim, L. Ohno-Machado, Blockchain distributed ledger technologies for biomedical and health care applications, J. Am. Med. Inf. Assoc. 24 (6) (2017) 1211–1220, https://doi.org/10.1093/jamia/ocx068.

[8] G. Zyskind, O. Nathan, A.S. Pentland, Decentralizing privacy: using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops; 21–22 May 2015; San Jose, CA, USA, IEEE, Piscataway, NJ, USA, 2015, pp. 180–184, https://doi.org/10.1109/spw.2015.27.

[9] A. Azaria, A. Ekblaw, T. Vieira, et al., Medrec: using blockchain for medical data access and permission management, in: 2016 IEEE 2nd International Conference on Open and Big Data (OBD); 22–24 Aug 2016; Vienna, Austria, IEEE, Piscataway, NJ, USA, 2016, pp. 25–30, https://doi.org/10.1109/obd.2016.11.

[10] D.B. Resnik, Compensation for research-related injuries, ethical and legal issues, J. Leg. Med. 27 (3) (2006) 263–287, https://doi.org/10.1080/01947640600870866.

[11] S. Gainotti, C. Petrini, Insurance policies for clinical trials in the United States and in some European countries, J. Clin. Res. Bioeth. 1 (1) (2010), 101, https://doi.org/10.4172/2155-9627.1000101.
[12] G.R. Chingarande, K. Moodley, Disparate compensation policies for research related injury in an era of multinational trials: a case study of Brazil, Russia, India, China and South Africa, BMC Med. Ethics 19 (2018), https://doi.org/10.1186/s12910-018-0244-y, 8 (2018).
[13] D.M. Studdert, T.A. Brennan, No-fault compensation for medical injuries: the prospect for error prevention, JAMA 286 (2) (2001) 217–223, https://doi.org/10.1001/jama.286.2.217.
[14] K. Wallis, D. Susan, No-fault compensation for treatment injury in New Zealand: identifying threats to patient safety in primary care, BMJ Qual. Saf. 20 (7) (2011) 587–591, https://doi.org/10.1136/bmjqs.2010.047696.
[15] B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, U. Sattler, Owl 2: the next step for OWL, J. Web Semantics 6 (4) (2008) 309–322, https://doi.org/10.1016/j.websem.2008.05.001.
[16] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper 151 (2014) 1–32.
[17] V.L. Mahan, Clinical trial phases, Int. J. Clin. Med. 5 (21) (2014) 1374–1383, https://doi.org/10.4236/ijcm.2014.521175.
[18] J.A. Fisher, L. McManus, M.M. Wood, et al., Healthy volunteers' perceptions of the benefits of their participation in phase I clinical trials, J. Empirical Res. Human Res. Ethics 13 (5) (2018) 494–510, https://doi.org/10.1177/1556264618804962.
[19] K. Degtyarenko, P. de Matos, M. Ennis, et al., ChEBI: a database and ontology for chemical entities of biological interest, Nucleic Acids Res. 36 (suppl_1) (2008) D344–D350, https://doi.org/10.1093/nar/gkm791.
[20] G. Schadow, C.N. Mead, D.M. Walker, The HL7 reference information model under scrutiny, Stud. Health Technol. Inf. 124 (2006) 151–156.
[21] D.B. Fridsma, J. Evans, S. Hastak, et al., The BRIDG project: a technical report, J. Am. Med. Inf. Assoc. 15 (2) (2008) 130–137, https://doi.org/10.1197/jamia.M2556.
[22] S. Hume, J. Aerts, S. Sarnikar, et al., Current applications and future directions for the CDISC operational data model standard: a methodological review, J. Biomed. Inf. 60 (2016) 352–362, https://doi.org/10.1016/j.jbi.2016.02.016.
[23] M. Mustra, K. Delac, M. Grgic, Overview of the DICOM standard. 2008 50th International Symposium ELMAR; 10–12 Sep 2008; Borik Zadar, Croatia, IEEE, Piscataway, NJ, USA, 2008, pp. 39–44.
[24] B. Smith, Ontology, in: G. Hurtado, O. Nudler (Eds.), The Furniture of the World, Rodopi, New York, NY, USA, 2012, pp. 47–68.
[25] N. Guarino, D. Oberle, S. Staab, What is an ontology? in: S. Staab, R. Studer (Eds.), Handbook on Ontologies Springer, Berlin, Heidelberg, Germany, 2009, pp. 1–17.
[26] M. Klein, Combining and relating ontologies: an analysis of problems and solutions, in: A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt (Eds.), Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing; 4–5 Aug 2001; Seattle, WA, USA, 2001, pp. 53–62.
[27] E. Chondrogiannis, V. Andronikou, A. Tagaris, et al., A novel semantic representation for eligibility criteria in clinical trials, J. Biomed. Inf. 69 (2017) 10–23, https://doi.org/10.1016/j.jbi.2017.03.013.
[28] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of SPARQL, ACM Trans. Database Syst. 34 (3) (2009) 1–45, https://doi.org/10.1145/1567274.1567278.
[29] E. Chondrogiannis, V. Andronikou, E. Karanastasis, et al., An intelligent ontology alignment tool dealing with complicated mismatches, International SWAT4LS Workshop; 9–11 Dec 2014; Berlin, Germany (2014).
[30] E. Chondrogiannis, E. Karanastasis, V. Andronikou, et al., Bridging the gap among cohort data using mapping scenarios, J. Adv. Inf. Technol. 12 (3) (2021) 179–188, https://doi.org/10.12720/jait.12.3.179-188.
[31] E. Chondrogiannis, V. Andronikou, E. Karanastasis, et al., A novel framework for user-friendly ontology-mediated access to relational databases, Int. J. Comput. Inform. Eng. 9 (3) (2015) 685–694.
[32] K.N. Griggs, O. Ossipova, C.P. Kohlios, et al., Healthcare blockchain system using smart contracts for secure automated remote patient monitoring, J. Med. Syst. 42 (2018), https://doi.org/10.1007/s10916-018-0982-x, 130(2018).
[33] M. Bartoletti, L. Pompianu, An empirical analysis of smart contracts: platforms, applications, and design patterns, in: M. Brenner (Ed.), Financial Cryptography and Data Security. FC 2017., Springer, Cham, Switzerland, 2017, pp. 494–509, https://doi.org/10.1007/978-3-319-70278-0_31.
[34] L.J. Zhou, L.C. Wang, Y.R. Sun, MIStore: a blockchain-based medical insurance storage system, J. Med. Syst. 42 (2018), https://doi.org/10.1007/s10916-018-0996-4, 149(2018).
[35] X.Q. Li, P. Jiang, T. Chen, et al., A survey on the security of blockchain systems, Future Generat. Comput. Syst. 107 (2020) 841–853, https://doi.org/10.1016/j.future.2017.08.020.
[36] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (SoK), in: M. Maffei, M. Ryan (Eds.), Principles of Security and Trust. POST 2017, Springer, Berlin, Heidelberg, Germany, 2017, pp. 164–186, https://doi.org/10.1007/978-3-662-54455-6_8.
[37] E. Androulaki, A. Barger, V. Bortnikov, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: EuroSys '18: Proceedings of the Thirteenth EuroSys Conference; 23–26 Apr 2018; Porto, Portugal, ACM, New York, NY, USA, 2018, pp. 1–15, https://doi.org/10.1145/3190508.3190538.