

# Dynamic Consent Management for Clinical Trials via Private Blockchain Technology

Giuseppe Albanese · Jean-Paul  
Calbimonte · Michael Schumacher ·  
Davide Calvaresi

Received: date / Accepted: date

**Abstract** Clinical Trials (CTs) are essential for the advancement of medical research, paving the way for the development and adoption of new treatments, and contributing to the evolution of healthcare. An essential factor for the success of a CT is the appropriate management of its participants and their personal data. According to the current regulations, collecting and using personal data from participants must comply with rigorous standards. Therefore, healthcare institutes need to obtain freely given, specific, informed, and unambiguous *consent* before being able to collect the data. Some of the major limitations of the current technological solutions are the lack of control over the granularity of consent grants, as well as the difficulty of handling dynamic changes of consent over time. In this paper, we present SCoDES, an approach for trusted and decentralized management of dynamic consent in clinical trials, based on blockchain technology (BCT). The usage of blockchain provides a set of features that allow maintaining consent information with trust guarantees while avoiding the need for a dedicated or centralized third trusted party. We provide a full implementation of SCoDES, made available as a self-contained infrastructure, with the possibility to interact with external services, and using Hyperledger as a blockchain framework.

**Keywords** Dynamic Consent Management, · Trust · Blockchain · Clinical Trials

---

HES-SO  
Sierre,  
Switzerland  
E-mail: name.surname@hevs.ch

## 1 Introduction

Clinical Trials (CTs) play a fundamental role in the advancement of medical research, for example, through data collection and analysis regarding *safety* and *effectiveness* of new drugs and (bio)medical devices (Pocock, 2013). These studies pave the way for the development and adoption of new treatments and therefore have a profound impact on the evolution of healthcare. One of the most critical factors for the success of a CT is the appropriate management of its participants and their personal data. Every step of a given study involves sharing, validating, monitoring, and accessing a considerable amount of sensitive data. To handle this kind of information, it is necessary to guarantee reproducibility, transparency, privacy, inviolability, and consent on the data. According to the GDPR (EU, 2019; Anjomshoae et al., 2019), collecting and using personal data from participants must comply with rigorous standards. Therefore, healthcare institutes need to obtain freely given, specific, informed, and unambiguous *consent* before being able to collect the data (Association, 2013).

Currently, a large number of healthcare and research institutions still rely on paper-based management of the consent (Kaye et al., 2015; Calvaresi et al., 2017). The process of migrating the consent management from paper to digital is still ongoing and characterized by several open challenges (Atasoy et al., 2018). Some of the significant limitations of the current technological solutions are the lack of control over the granularity of consent grants, the difficulty of handling dynamic changes of consent over time (Compert et al., 2018), and still open ethical concerns when binding such data with intelligent systems (Calvaresi et al., 2019a). Handling the consent in CTs demands to ensure proper management of information in a distributed manner while guaranteeing high levels of security and trust (Lorell et al., 2015). Whereas in paper-based systems trust is mainly dependent on the human factor, in the digitized approach, distributed systems and underlying technologies are the ones subject to strict scrutiny. In this scope, some of the key aspects required for trusted and secure management of consent in CTs include: *(i)* certified authentication for access to data and resources, *(ii)* decentralized and trusted consent data access across different healthcare institutions and participants, *(iii)* verifiable, unfalsifiable and accountable registry of operations, transactions and consent authorizations, *(iv)* secure and consistent modification of dynamic consent, and *(v)* participant-centered control over her own consent preferences, approvals and revocations.

In this paper, we propose an approach for trusted and decentralized management of dynamic consent in clinical trials, based on blockchain technology (BCT). The usage of blockchain –an emergent technology for decentralized sharing and management of an immutable and transparent append-only registry (Nakamoto, 2008)– provides a set of features that allow maintaining consent information with trust guarantees while avoiding the need for a dedicated or centralized third trusted party (TTP) (Agbo et al., 2019). The proposed design leverages several characteristics inherent to BCT to manage consent

information. Using cryptographic primitives, and possibly relying on specific membership mechanisms and consensus protocols (Cachin and Vukolić, 2017), BCT can be used to store sensitive information (e.g., consent) ensuring transparency and verification of the shared data even among distributed intelligent and autonomous systems (Calvaresi et al., 2018, 2019b).

The blockchain can execute a predetermined set of tasks (named smart contracts) that operate on the registry replicating the required actions in every peer. The stored data are digitally signed transactions (broadcasted by the participants) grouped into chronologically timestamped blocks. Two adjacent blocks are connected by a unique identifier, which is obtained by hashing the content of the antecedent block and stored in the subsequent (see Figure 1). Thus, potential alteration of the block’s content can be easily verified: by hashing it again and comparing the identifiers of the two subsequent blocks. Moreover, the blockchain is replicated and maintained by every peer, making it easy to spot any malicious attempt to tamper the registry.

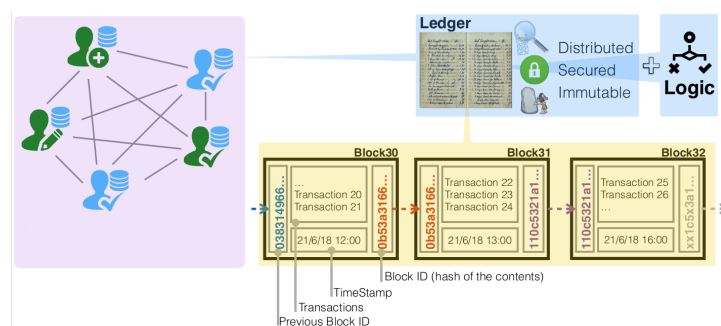


Fig. 1 BCT main components.

### Contribution:

This paper presents SCoDES, an approach and a fully implemented system for consent management in clinical trials, based on BCT. In particular, the formulation and management of the patient consent are *dynamic*, i.e., automatically generated according to the features of a given trial; and *reliable*, leveraging the qualities of the underlying blockchain infrastructure.

Moreover, the system provides full control to the patient over her consent preferences, including the possibility of *accepting/rejecting* requests of consent at different levels of granularity, and of smoothly *revoking* a prior consent, keeping track of the associated data. All operations involving the actors of a given CT are safely stored on the distributed ledger, through a private permissioned blockchain<sup>1</sup>.

<sup>1</sup>A blockchain is permissioned if the identities of the users and rights to participate in the consensus (writing to the ledger and/or validating the transactions) are controlled by a membership service.

Furthermore, the system has been implemented and made available as a self-contained infrastructure. Nevertheless, it has been designed with the possibility to interact with external services/components. For example, it provides integration with existing third-party CT software, such as REDCap (Research Electronic Data Capture), through a dedicated API. The underlying concept is that the blockchain module of the system can be decoupled from its use case. Indeed, the presented system can operate as an external consent management service extending already existing CT management systems. Summarizing, the most relevant features of BCT implemented in SCoDES include:

- *Trust management*: trust is not centralized on a single actor, but it is distributed among the peers/participants of the network (in this case health-care institutions and research facilities);
- *Immutability*: the ledger cannot be modified, nor any transaction stored in it can be deleted. Every modification is recorded and it leads to the creation of a new record (keeping the complete history);
- *Authentication*: access to the resources is granted through the authentication process managed by a certification authority;
- *Consistency*: updates of the world state (all the resources of a network with their corresponding attributes and respective values) are spread across the network and committed by each node
- *Integrity*: transactions are visible but not mutable due to the cryptographic nature of the ledger

The rest of the paper is structured as follows. Section 2 describes current structures, management, use, and implementations of existing Clinical Data Management Systems (CDMSs). Section 3 introduces the system design and modeling approach. Section 4 details the prototype (architecture and the technologies), motivating the strategical choices. Section 5 proposes the studied use cases and alternative approaches. Finally, Section 6 concludes the paper discussing and summarizing the objectives reached and open challenges.

## 2 State of the art

Since the emergence of blockchain technologies, the potential domains of application experienced fast growth. According to the study proposed by Casino et al. (Casino et al., 2018), a possible classification is proposed in Table 1. Although the domains are heterogeneous, there are consistent overlaps among the applications. For example, in the health domain, electronic health records (EHR) can get patients’ biological values from wearable and distributed sensors – henceforth, creating potential overlaps with the domain of IoT. For example, envisioning such direction, Rantos et al. (Rantos et al., 2018) proposed a blockchain-enabled framework to address primary emerging privacy needs in the IoT ecosystem (reconciling patient, home environment, and e-health services). Below, we explore existing BCT-based works in the health and the biomedical domain, focusing on consent management, clinical trial manage-



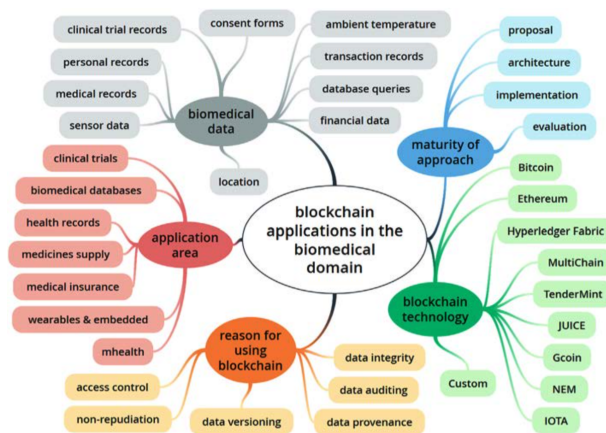
ment systems (CTMS), and the usage of blockchain for ensuring privacy and trust in such a domain.

Domain	Applications	Domain	Applications
Health	electronic health records, consent	IoT	eBusiness, distributed device management
Privacy & Security	Anonymization, secure storage	Education	reputation, certification management
Integrity verification	counterfeit, insurance, intellectual property	Financial	cryptocurrencies, marketplace prediction
Governance	identity management, proof of existence, notary & law, public administration	Business and Industry	energy sector, supply chain
Data Management	human resources, data distribution		

**Table 1** Main Domain - Applications cassification

### 2.1 Blockchain and Biomedical applications

In the health domain, biomedical data-types involved in BCT-based applications are mostly medical electronic health records (EHR), personal health records, consent forms, drug information, environmental data, location, or medical evidence data (Drosatos and Kaldoudi, 2019). Drosatos and Kaldoudi (Drosatos and Kaldoudi, 2019) proposed an extensive study (in the form of a systematic literature review) about the application of BCT in the biomedical domain. Figure 2 shows how the authors schematize the distribution of the analyzed papers. Recently, EHR have gathered the most of the attention. However,



**Fig. 2** BCT in the Biomedical domain (Drosatos and Kaldoudi, 2019)

systems employing wearable and embedded sensors and supporting clinical trials are quickly advancing from the theoretical analysis to the early adoption of BCT (Drosatos and Kaldoudi, 2019). Concerning clinical trials, the

main focus of this paper, the underlying systems have noticeably evolved in the past years, covering different aspects of the data management cycle. Nevertheless, the current systems mostly rely on conventional databases coupled with practices based on paper-work, not satisfying crucial trust requirements (e.g., accountability, immutability) (Casino et al., 2018) and raising concerns about the consent management (Drosatos and Kaldoudi, 2019). Ange (Angeletti et al., 2017) employed the blockchain to preserve data privacy and integrity of the patients from the evaluation of his/her eligibility to the actual inclusion in a given trial. benchoufi (Benchoufi et al., 2017) proposed a mechanism to ensure non-repudiation and versioning of trial consent forms. Nugent et al. (Nugent et al., 2016) proposed to use a private blockchain to store all the data from clinical trials, guaranteeing its compliance with the protocol and data integrity.

## 2.2 Clinical trials & consent management

To manage clinical trials efficiently, digital solutions and standard methodologies need to be adopted and implemented (Friedman et al., 2010). Even if many healthcare organizations still rely on custom and heterogeneous CTMS (often incompatible with each other), the collaboration among different healthcare institutions is increasingly important. Biotechnological and pharmaceutical industries use such software to maintain, manage, plan, execute, report and interact with the participants, and track deadlines and milestones. Current CTMSs encompass the underlying business process. Therefore, they do not necessarily share the same characteristics or architectures. CTMSs have to reflect the structure and processes of CTs accurately. The adoption of Electronic Data Capture (EDC) and other technologies increased the number of medical applications. Current implementations provide overlapping functionalities, therefore, introducing redundancies and inefficiency. CTMS producers have to cope with **eliminating data discrepancies** and **reconciling** the activities across systems.

Besides the *clinical* data collected during the CT, the *patient consent* is the other sensitive piece of information that a CTMS should be able to manage (Davis et al., 1998). Patient consent can be defined as a set of policies allowing participants and patients to determine what health information they are willing to permit their various care providers/researchers to access (Neisse et al., 2015). It enables patients and participants to affirm their participation in e-health initiatives and to establish consent directives to determine who will have access to their protected health information (PHI), for what purpose and under what circumstances, enforcing consumer, organizational, and jurisdictional privacy policies (Mulder and Tudorica, 2019). As in the case of CTMSs, there are different Consent-Management Platforms (CMP). CTMSs need a consent-management platform to *(i)* process personal data, *(ii)* automate the data analysis, and *(iii)* allow data transfer between organizations. A CMP should document *(i)* who gave consent, *(ii)* when the consent was given,

(*iii*) what the user consented to, and (*iv*) the consent status (e.g., amended or withdrawn). Based on these requirements, it is possible to implement a solution by using standard off-the-shelf software or building a tailored application based on specific needs.

### 2.3 Clinical trial management software

Choosing a standard software (commercial or open-source) brings all the advantages of well-established policies and workflows for data acquisition and management. Compatibility is ensured by using standardized data interchange formats such as proposed by the Clinical Data Interchange Standard Consortium (CDISC<sup>2</sup>). In addition, with the growth of internet connectivity, many players are choosing web-based solutions to manage this type of data. Off-the-shelf systems are also particularly suited for a distributed context in which multiple distant trial sites are involved. Applications can also be delivered with a rental fee through a Software as a Service (SaaS) model. The main disadvantage of using these approaches is that customers may not be willing to rely on external cloud storage and data management for keeping privacy.

Considering the different approaches for CTMS, existing standard and custom implementations are described below. REDCap (Harris et al., 2009; University, 2019) (Research Electronic Data Capture) is a free cross-platform electronic data capture (EDC)<sup>3</sup> system for designing clinical and translational research databases (based on metadata). Such a system can replace the traditional paper-based data collection practices fastening the time to market for drugs and medical devices, therefore widely adopted by pharmaceutical companies and contract research organizations. REDCap is built upon a lightweight PHP stack, relatively easy to deploy and maintain. Moreover, it implements security and privacy on different levels through access control rules, authentication, and several filters. The creator of a project (e.g., a CT) can set user privileges to define what resources are accessible to whom with a customizable granularity. The most common controls focus on limiting access to functionalities such as exporting/importing data, modifying surveys, running reports, and viewing logging records. To protect the data stored in REDCap, the application uses several techniques to filter, sanitize, and validate data. The application also provides a few methods to prevent common attacks such as Cross-Site-Scripting (XSS) and SQL injection. However, this layer of security does not cover protection on data storage. Setting the web server, database server, and securing the communication are responsibilities of the partner institution installing REDCap.

OpenClinica is an open-source clinical data management system (CDMS) providing EDC features coupled with electronic Case Report Form (eCRF) functionalities. It is a web-based solution (offering both free and enterprise

---

<sup>2</sup>CDISC: <https://www.cdisc.org/standards>

<sup>3</sup>An EDC is a computerized system designed for the collection of clinical data in electronic format for use mainly in human clinical trials.

versions). OpenClinica provides a modular structure for setting up the study, submitting data, monitoring and extracting the data. Moreover, its compliance with CDISC Operational Data Model (ODM) for data interchange is a key feature motivating a broad adoption (LLC, 2019). OpenClinica uses a PostgreSQL database that mirrors the structure and nomenclature of the CDISC ODM standard. The customer demands the definition of data protection and security policies.

Phoenix CTMS is a custom Java web application developed by the University of Graz (Krenn, 2014). It implements a full set of EDC capabilities with input form composition and scripting and elaborated user requirements (e.g., web calendars and document management). Moreover, it is compliant with regulations such as *the Good Clinical Practice, Data Privacy Act* through the implementation of data security measures including subject de-identification and application-level encryption of data at rest, audit trail and digital signatures, configurable user privileges, and host-based access restriction.

#### 2.4 Blockchain and CTMS consent management

Although implementations and architectures of the software solutions mentioned above may differ, they offer similar features typical of a CTMS. However, concerning the key concept of *consent management*, they treat it in a rather static and traditional manner. In particular, the consent related to a set of sensitive data is implemented through *standard database fields* representing status (e.g., unopened, viewed, and signed) reflecting the conditions of actual (physical) documents **still part of the clinical trial initialization process**. Besides the well-known challenges characterizing the management of the consent for the actual medical data, relating the consent to the research-data (e.g., aggregated and post-analysis data) produced by elaborating the medical data still requires further investigations (Jahankhani and Kendzierskyj, 2019).

Moreover, non-conformance of consent, the dangers of selective reporting, bias, and misconduct leading to more severe implications can affect CTs in any of their phases. To this end, BCT are *claimed* to be able to play a crucial role in CTs. Nonetheless, there is a **lack of practical applications**. BlockTrial is an early-stage implementation of BCT for CT (Maslove DM, 2018). It uses a web-based interface allowing users to run trials-related Smart Contracts on an Ethereum network. The platform enables participants to grant researchers access to their data and allows researchers to submit queries for data that are stored off-chain. Both participants and researchers behave as nodes in BlockTrial, each one with its specific Smart Contract. The network in this project is implemented using a private version of the Ethereum blockchain, a widely used solution that also provides a well-established framework. The prototype presented in this paper overcomes these limitations by employing a private blockchain infrastructure implemented through the Hyperledger framework that does not involve the use of any cryptocurrency, as detailed in the next section.

Other blockchain-based platforms for trusted information sharing includes the Korean TIP platform (Lee and Yoon, 2019), although it is not focused on consent management policies. Similarly, the Peer-to-peer file sharing system in (Vimal and Srivatsa, 2019), manages distributed resources shared by decentralized entities using blockchain, but lacks higher level access control and is limited to the capabilities of a distributed file system. Beyond blockchain-based approaches, other prototypes such as the mobile based data acquisition application by Beierle et al. (Beierle et al., 2019) explored ways of protecting data privacy, showing what types of data a participant is willing to share with researchers. It is in fact crucial to understand the dynamics of data sharing, which may also include the patients' next of kin, informal caregivers, etc. (Jaschinski and Allouch, 2019), which should be considered in any data consent and sharing system.

### 3 System Design

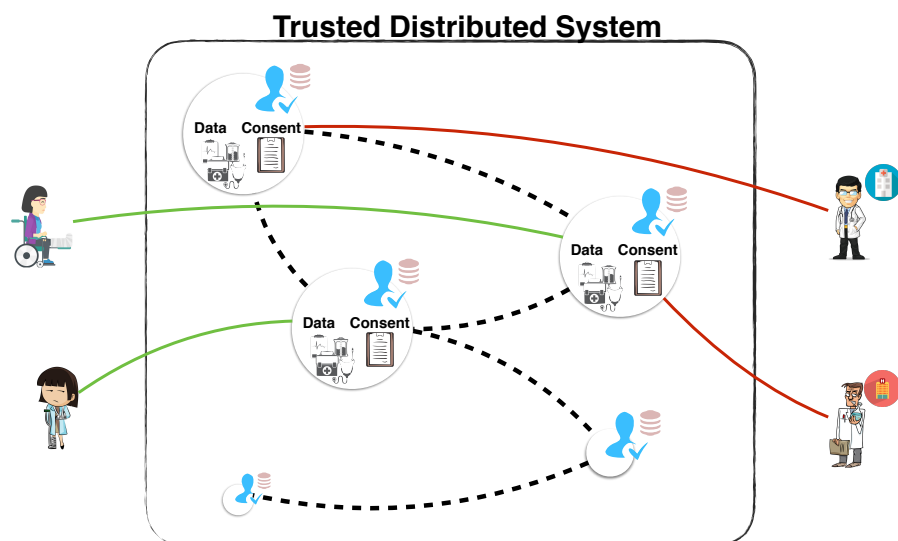
The design of SCoDES considers the management of consent for clinical studies involving several main players: healthcare institutes (e.g., hospitals or clinical research labs), private industries (e.g., pharmaceutical), universities (e.g., medical and data science labs), and regular participants. Therefore, the user groups (UG) of the system can be classified as follows:

- UG1** Investigators - initiating the CT or observing its evolution/results, and interacting with the participants (e.g., demanding for the data consent).
- UG2** Participants - tested subjects/patients: who can see the details of the clinical trials and take part in them by giving the consent to data treatment;
- UG3** Observers - accessing the data regarding given CTs for analytical purposes (e.g., industrial representatives and scientific researchers).

Such users must be able to interact (e.g., creating CT and recording/reading data) supported by reliable and secure consent management mechanisms (see Figure 3).

Elaborating on the still unmet needs presented in Section 2, the main objective set for SCoDES is *to enable the dynamic management of the patient consent via BCT*. Such a high-level objective can be decomposed in several structured system goals:

- G1** to improve the current management of consent for CTs;
- G2** to allow data-transfer between medical platforms, enabling to provide the information required in a consistent form ensuring the patient confidentiality;
- G3** to enforce the trust by ensuring the authenticity and integrity of participants' data and consent.
- G4** to ensure traceability, patient confidentiality, and transparency in the management of the patient data and consent, by employing a shared distributed ledger (e.g., BCT).



**Fig. 3** SCoDES project: Logical components and interactions.

Goals	Requirements
G1	<ul style="list-style-type: none"> <li>● R1.1: Paperless consent management</li> <li>● R1.2: Easy data insertion and access</li> </ul>
G2	<ul style="list-style-type: none"> <li>● R2.1 Integration between consent management and data capture systems</li> <li>● R2.2 Extension of existing solutions with CMS as a service</li> <li>● R2.3 CMS supporting the development of new solutions</li> </ul>
G3	<ul style="list-style-type: none"> <li>● R3.1 Certification based applications</li> <li>● R3.2 No need of Trusted Third Parties</li> <li>● R3.3 Guarantee of data authenticity</li> <li>● R3.4 Guarantee of data integrity</li> <li>● R3.5 Guaranteed data source</li> </ul>
G4	<ul style="list-style-type: none"> <li>● R4.1 Consent traceability</li> <li>● R4.2 Access control on medical data and digital consent</li> <li>● R4.3 Seamless integration between CTMSs and CMSs</li> <li>● R4.4 Tamper-proof data storing and monitoring</li> <li>● R4.5 Distributed reconciling of CTMSs and CMSs</li> </ul>

**Table 2** Project goals and requirements

To fully satisfy such goals, the set of formal requirements formalized in Table 2 must be met.

Therefore, the developed prototype –named SCoDES– has been designed to enable the creation and management of CTs, user profiles, and patient consent (via BCT). Moreover, it provides APIs to connect third-party electronic data capture software (e.g., REDCap (University, 2019)). Figure 4 represents the components and design of SCoDES schematically.

The implementation of SCoDES is characterized by classical client-server architecture. The client (front-end) consists of a web application that allows the users of the system to access and visualize data. The server (back-end) con-

sists of a custom REpresentational State Transfer (REST) server providing Application Programming Interfaces (APIs) to communicate with the blockchain and perform the needed transactions. The web application exchanges data with an underlying blockchain (composed of a data model, chaincode, and a set of access rules). The data model contains the design and definition of the assets (Crosby et al., 2016), the participants, the transactions, and the events that are part of the system. The chaincode (also known as smart contract - the naming depends on the technology used) contains the definitions of the transactions and their implementation. Finally, access to the resources is defined through a set of access control rules.

From a functional perspective, the users (patient(s), investigator(s), researcher(s)) can access the system via a web application, which communicates with a blockchain infrastructure (multi-peer deployable in both single-/multi-machine configurations).

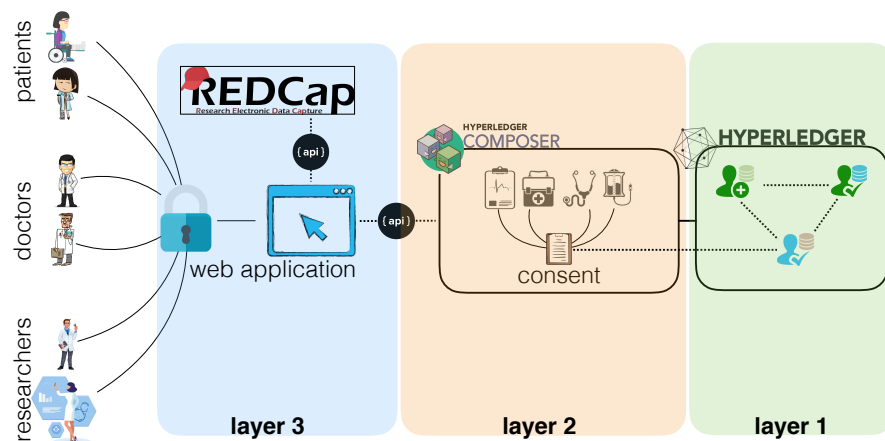


Fig. 4 Design of the SCoDES architecture.

## Layer 1

The first step of the design process included studying and modeling the underlying distributed ledger technology (DLT). As introduced in Section 1, a blockchain can be considered as a secure and decentralized append-only data-store of ordered records (grouped blocks). Each block contains timestamped transactions and is linked to a previous block via a unique identifier (generated by hashing the content of the antecedent block) (Yaga et al., 2019). Therefore, the BCT has been considered as a means to record the entire history of the transactions (concerning the consent management) that occurred in a given CT.

---

*Blockchain Infrastructure*

The Blockchain Network running in the prototype has been implemented with the Hyperledger framework<sup>4</sup>. To facilitate the understanding of the underlying blockchain layer, the next subsection quickly summarizes the features of Hyperledger Fabric that are more relevant for the SCoDES project.

*Hyperledger Fabric*

Hyperledger Fabric is an open-source enterprise-grade permissioned distributed ledger technology (DLT) platform (IBM, 2019). Its main characteristic is the *permissioned* nature (the identities can be disclosed among trusted entities, which are the only authorized to access the data). The *private* nature of Hyperledger Fabric fits well with the complex structure of a CTMS. Such a framework enables the modeling of all the entities involved in the process of data and consent management. For instance, in the SCoDES project, a peer node has been used to represent a healthcare institution, therefore connecting to it a private channel to carry out transactions.

The basic strategical key features for software dealing with CTs and digital consent are:

- Authentication: known identities,
- Access Control: role assigned restricts the actions,
- Transaction Validation: a subset of participants' checks and (if valid) signs the transactions to be endorsed.

Participants decide who and how the block validation takes place. Since the validation process involves a subset of peers that are part of the network, the administrator(s) can decide which nodes have this capability (who) and what kind of consensus algorithm<sup>5</sup> they should use (how). Hyperledger Fabric's components can be analyzed from different perspectives:

Logic: The basic elements defining the business logic of a Hyperledger Fabric network are:

- *Assets*: they represent the digitized values (JSON or binary files). They can be either tangible (sensors measurements, biometrics) or intangible (consent on data treatment). In a DLT, a transaction expresses the change of assets' state.
- *Chaincode*: it defines the assets' structure and the business logic for the transactions (e.g., in the SCoDES project, it defines the structure of the trials and the transactions used to manipulate them).

---

<sup>4</sup>Hyperledger is an open-source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration, hosted by The Linux Foundation, including leaders in finance, banking, Internet of Things, supply chains, manufacturing and Technology(IBM, 2019)

<sup>5</sup>A consensus algorithm defines the mechanisms ruling agreement among several peers about the correctness and security of a given transaction.



- *Ledger*: it records all of the assets transactions and changes (e.g., if a participant accepts a contract, this action is recorded on the ledger).

Privacy: Hyperledger Fabric is a permissioned network (no anonymous access). The components of the authenticated infrastructure are:

- *Members*: legally separate or independent entities (e.g., *network administrators*, investigators, participants, and researchers). Depending on their authority, members may be able to use a membership service providers (MSP) to create participants and infrastructure component identities within their organization.
- *Membership Service Provider(s)*: they can be one or more per network. Such components handle the authentication process (via a Public Key Infrastructure - PKI). Each organization has a Certification Authority (CA) that provides X509 certificates to identify each participant (used to handle identity and validate the transactions).
- *Certification Authority*: entity handling (issuance and revocation) of the certificates.

Connection: A Hyperledger network is composed of three types of nodes (peer-to-peer endpoints distributing and syncing the ledger) connected through channels.

- *Client*: node(s) initiating transactions (certificate required).
- *Peer*: node(s) keeping the ledger synced (certificate required).
- *Orderer*: node(s) backbone of the communication. They are responsible for the distribution and order of the transactions.
- *Channel*: members can participate in multiple Hyperledger blockchain networks. The transaction in each network is isolated, and this is made possible thanks to the channels. The peers connected with a given channel can receive all the transactions broadcasted on it. Each channel is dedicated to an independent ledger.

## Layer 2

The middle layer provides a higher-level interface towards the underlying network. This interface consists of a lightweight server communicating with a blockchain infrastructure built with the help of Hyperledger Composer, a set of tools facilitating the creation and management of business networks<sup>6</sup> running on a Hyperledger Fabric environment.

### *Hyperledger Composer*

While Fabric allows defining the concrete components of the network (i.e., peers, Certification Authorities (CAs) and Orderers), Hyperledger Composer allows to abstract the process of defining the actual network. To build a

---

<sup>6</sup>Business network refers to a blockchain application developed with Hyperledger Composer.

blockchain application, Hyperledger Composer requires four essential elements: model, chaincode, access control rules, and queries.

**Model:** the model consists of one or more Concerto files (CTO) containing a high-level object-oriented description of the domain model. In the SCoDES project, there is a single model file defining the structure and the transactions related to the assets of the network (i.e., contracts and trials). CTOs are composed of:

- A single namespace (i.e., a declarative region) containing all the resource declarations.
- A set of resource definitions, encompassing assets, transactions, participants, and events
- import declarations (import resources from other namespaces - optional)

**Chaincode:** The transaction logic is contained in at least one JavaScript file. Such scripts may contain transaction processor functions implementing the model and including decorators and metadata.

A transaction processor function is the logical operation of a transaction defined in a model file. For example, in this project, a transaction processor function to give consent for a clinical trial uses JavaScript to change the status of a contract from *issued* (i.e., newly created) to *accepted*. Through transaction processors, it is also possible to emit events once the transaction is completed.

**Access Control Rules:** to determine which users/roles are permitted to create, read, update, or delete elements in a network is allowed by ACL rules. Hyperledger Composer differentiates between access control for resources within a business network (business access control) and access control for network administrative changes (network access control). Business access control and network access control are both defined in the access control file (.acl) for a business network. Network access control uses the system namespace, which is implicitly extended by all resources in a business network, and grants or denies access to specific actions. For example, in the SCoDES prototype, the *admin* user has the system-level right to enroll and validate new participants in the network.

**Queries:** the queries can be used to get information about the blockchain world-state. Dynamic queries are used in the back-end of the SCoDES website to retrieve specific resources. For example, getting a list of contracts with a precise status.

### Layer 3

The presentation layer has been designed primarily considering the goal G1 (to facilitate and improve the use of a CTMS, even for inexperienced users). Hence, the application's user interface (UI) has been designed as a website (based on *React.js*), familiar to most users, and simple to learn for beginners. The rationale of this choice are (i) *React.js* grants ease of use, flexibility,

and modularity, and *(ii)* it speeds up the developing process (it is written in JavaScript, a programming language used throughout the whole project). The website interface has been designed with a typical dashboard layout, which allows the users to keep track and explore their data concisely and straightforwardly.

The next section provides the technical details concerning the implementation of the presented design.

## 4 System Implementation

The implementation of SCoDES follows the three-layered architecture shown in Figure 4. A web interface (layer 3) was chosen as the main interface for the end-users, given its cross-device compatibility, usability, and appeal. The logic (layer 2) is composed of a custom REST server that provides APIs to communicate with the underlying blockchain infrastructure. The blockchain infrastructure (layer 1) provides mechanisms for data access, exposition, and persistence. A detailed description of these layers follows.

### 4.1 Web Interface

According to the goals formalized in Section 3, to satisfy the diverse and remarkably heterogeneous classes of users (e.g., in terms of age, gender, geo-location, background), the most effective way is to provide them a cross-device website. The front-end has been developed using React JavaScript library (Inc., 2019), which uses a modular and flexible declarative approach facilitating the design. Each view is composed of several elements, which can be managed (e.g., their state) and rendered independently. Initially meant to develop single-page applications (SPA), React can also be used to realize but multi-pages (e.g., by using it natively or as React-Router). The SCoDES interface consists of a landing page with a short introduction to the SCoDES project and a login of the system.

Interfaces and functionalities have been realized according to the user role (see Table 3). In particular, there are: a *dashboard*, in which the users can see a summary of a set of data (user-role dependent); a *trials* page in which all available trials are listed; a *contracts* page that shows the consent status; and the patient *profile* displaying his/her medical data (in this case stored on REDCap).

#### 4.1.1 Dashboard

The dashboard page summarizes the most significant system information according to the user class. In particular:

Administrator : the dashboard displays a chart about the transactions committed since the deployment of the network (organized by the number of

transactions per hour of the day). The listed transactions are divided by type (according to the asset involved). Below the chart, there is a table containing all the transactions characterized by id, transaction type, and timestamp (sorted by submission time). Finally, the page ends with three smaller tables listing all the participants of the network organized in doctors, researchers, and participants (see Figure 5).

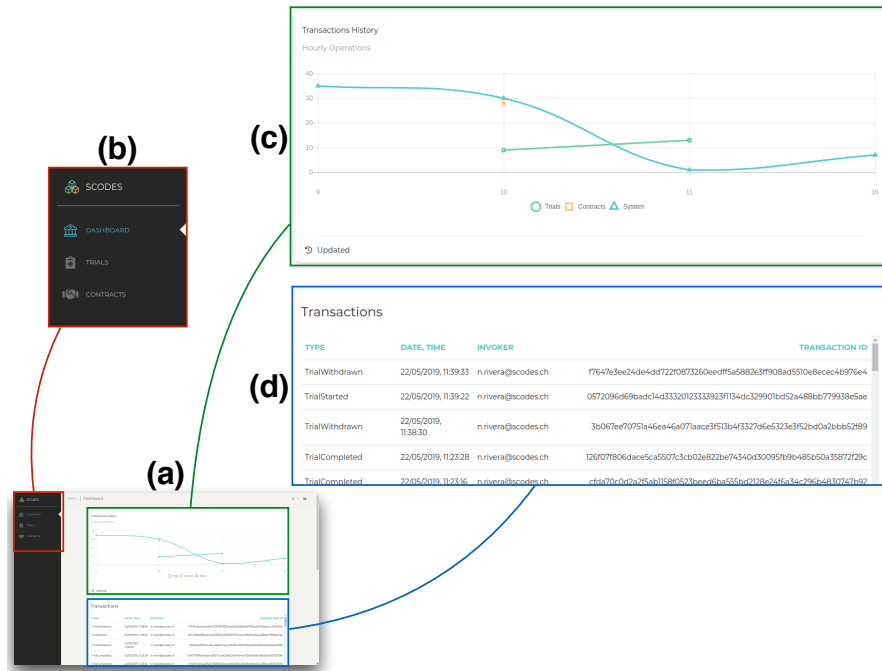


Fig. 5 Administrator Dashboard

Doctor : the dashboard summarizes trials and contracts. In the first row, four plots are showing, respectively, the number of ongoing trials, contracts (specifying the active ones), patients, and revoked contracts. In the second row, there are two charts: the former is a pie-chart indicating the number of trials based on their status (e.g., completed, started, stopped, and withdrawn ones), the latter shows the trend of the number of patients and trials in the network (see Figure 6).

Participant : the dashboard recaps a patient's personal anagraphic and biometric data, and shows the participants of the network supervising/handling and observing the data. The biometric data section also provides a mock-up structure for displaying data potentially acquirable from wearable devices. (see Figure 7).

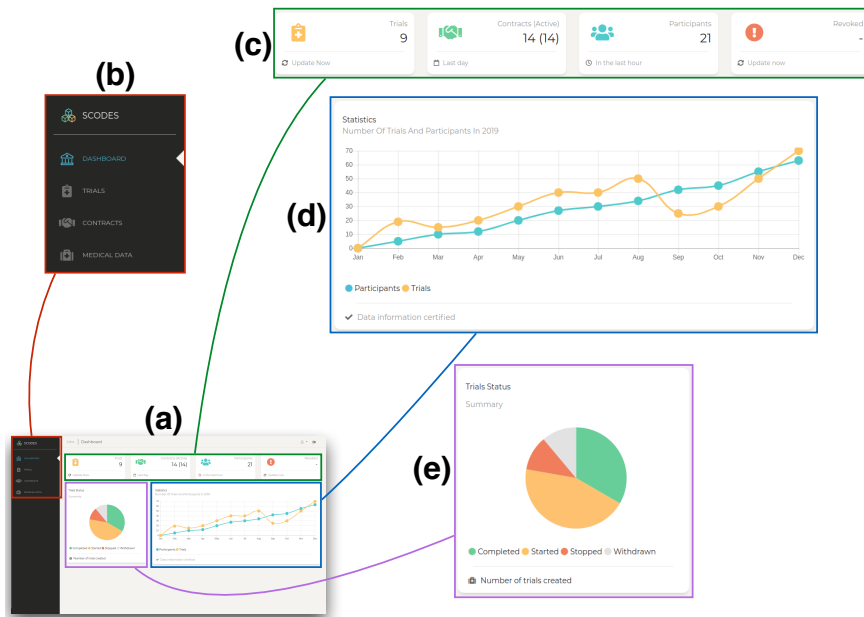


Fig. 6 Doctor Dashboard

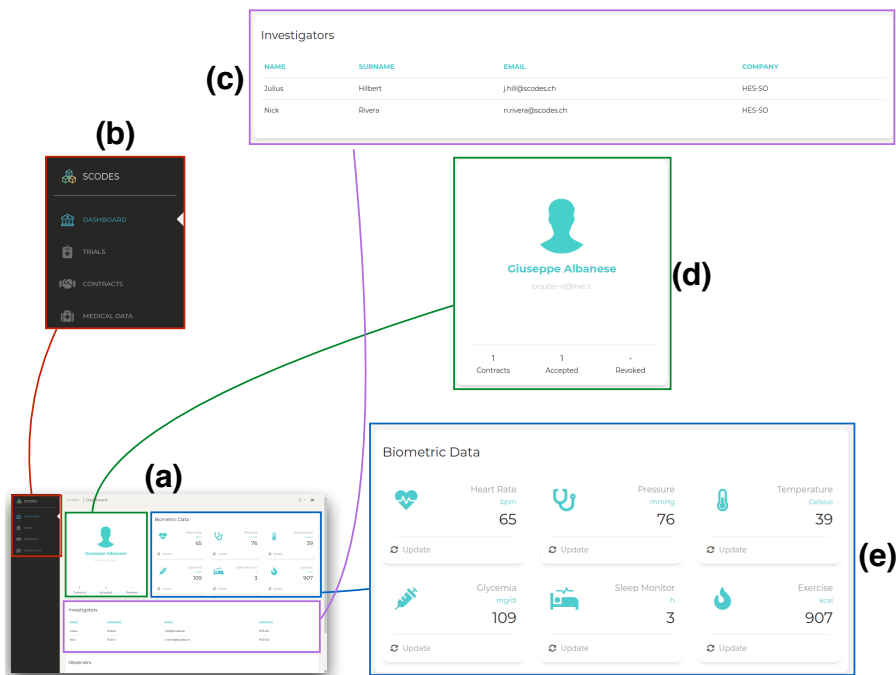


Fig. 7 Participant Dashboard

Researcher : the researcher dashboard enables to have an overview of data related to a specific clinical trial. This page loads all the data related to trials that are being observed by the logged-in researcher and to which participants have given consent (see Figure 8).

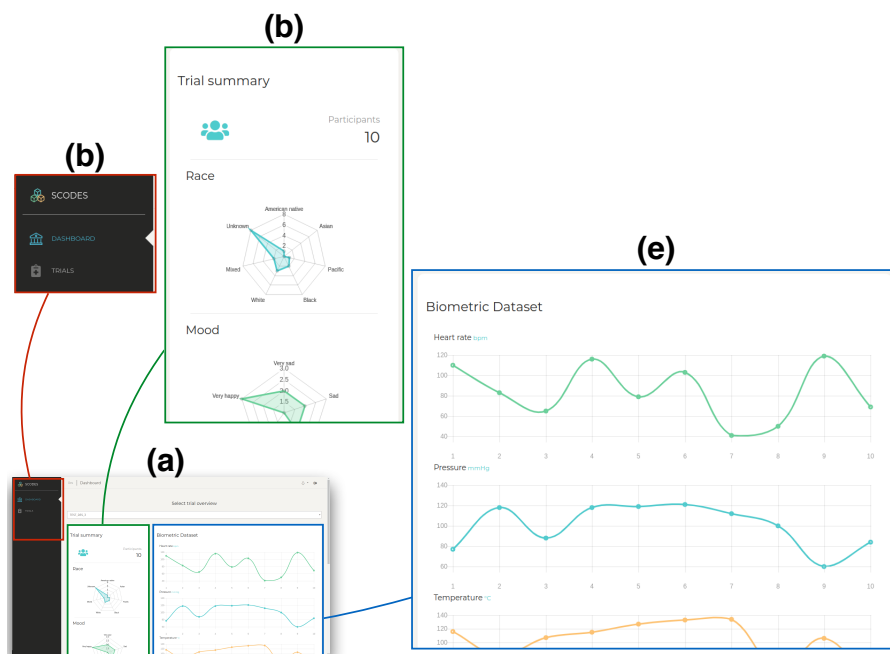


Fig. 8 Researcher Dashboard

#### 4.1.2 Trials

The trials page consists of a table reporting all the clinical trials on the network. The table provides information about the trial id, name, list of supervisors, (the doctors conducting the given trial), short description, duration of the trial, and its status. There is also a filter (drop-down menu) that allows selecting the visualized trials by their status. All the users can click on each row of the table to open a modal window containing a more extensive description of the selected trial (including all the phases and measured values involved). Each phase can be expanded to see deeper details like a brief description, starting and ending time, data-sets involved, and observers. The participants, unlike the supervisors, can visualize the data concerning the trials, but without any decisional power. The observers can be doctors and/or researchers. Doctors have additional functions on this page. They can create new trials through a dynamic modal that allows to add an arbitrary number of phases and data-sets and change the status of a trial. The latter function

has been implemented purely for demonstrative purposes, as doctors should not arbitrarily change the status of a trial. Instead, it should change due to events regarding the natural evolution of the trial itself.

#### 4.1.3 Contracts

The contracts page is similar to the trials page. It displays a table containing the contracts with characterized by id, demander, recipient, the referred trial and their status (the view is possible to be filtered). The newly issued contracts, namely the contracts that have not been accepted nor rejected, are highlighted in light blue.

On this page, the doctors can issue new contracts to possible participants to register the consent to access their medical data. Patients instead can click on a contract to open a modal containing the details of the related trial. Such a piece of information might be discriminant to decide whether to accept it or not. In this way, the participant can **at any time** grant or deny the access to his/her medical data to the supervisors and the observers of the trial.

#### 4.1.4 Medical Data

The medical-data page is available for doctors to manage the biometric data related to a given CT, and for patients, to insert here their biometric data (which are stored in REDCap).

## 4.2 REST Server

An advantage of the layered structure is its modularity. A REST server is a natural choice to decouple the system components and to comply with the principle of separation of concerns (Richardson and Ruby, 2008). The REST server realized for this prototype exposes an interface for enabling the front-end to manage the data. It has been implemented using *Node.js runtime* and *Express* framework. The functions offered by this layer are coded in three controllers: resources, transactions, and network.

### 4.2.1 Resources Controller

The resources controller manages the assets of the network. It contains methods to retrieve participants, trials, contracts, transactions, and events. These methods implement a standard process to access data stored on the blockchain:

- Connect to the network through the user's credentials<sup>7</sup>
- Perform the query
- Send the result/error

---

<sup>7</sup> Every operation that needs to communicate with the blockchain network needs to be authenticated. To avoid redundancy, this step is omitted in the next descriptions.

The connection is essential to perform requests. To access data, each participant needs to be recognized and authenticated. This process requires an identification artifact called *Business Network Card*. Such an object provides all the information needed to get connected to the blockchain network, and it is created by the administrator when a new identity is issued within the system. The queries to retrieve data are written in a *bespoke query language* defined by Hyperledger Composer, and can be contained in a single *.qry* File or built dynamically using Composer’s runtime API (see Listing 1). Finally, if the demand has been successful, the server replies with the data (JSON format), otherwise with the occurred error and its brief explanation.

```

1  async function getTrialsByStatus(req, res) {
2      const bnCon = new BusinessNetworkConnection();
3      const { cardName, status } = req.body;
4      try {
5          await bnCon.connect(cardName);
6          const query = bnCon.buildQuery('SELECT org.hevs.scodes
7      .clinical.Trial WHERE (status == _$status)');
8          const result = await bnCon.query(query, { status:
9      status });
10         res.status(200).json(result);
11     } catch (error) {
12         console.log(error)
13         res.status(500).send(error.toString());
14     }
15 }

```

**Listing 1** Dynamic query example

#### 4.2.2 Transactions Controller

The transactions controller implements methods to call transactions on the blockchain. It is important to note that transactions are not implemented here. This module exposes an interface to the chaincode (where the actual transactions are implemented). This controller allows to set up the parameters for calling transactions according to this procedure:

- Connect to the network through user’s credentials<sup>7</sup>
- Receive parameters from HTTP requests
- Create a transaction object using the Hyperledger Composer API
- Start the transaction submission through the chaincode

The possible transactions are: create trial, update trial status, create contract, and update contract.

#### 4.2.3 Network Controller

The network controller implements functions related to the communication with the blockchain network. The most important ones are the ping (to check



the network status) and the event listener (to catch events generated from transactions submission). The event listener allows sending events to the Front-End through web sockets to display different notifications. This feature enables the users to know if any relevant change occurred to network assets. For example, participants may notice that a new contract has been issued, thus checking the contract list they can decide whether to accept or reject it.

### 4.3 Blockchain implementation

The prototype blockchain network has been implemented following the scenario proposed in Figure 9. It involves an organization (e.g., a healthcare institution/hospital) managing clinical trials and demanding for a distributed blockchain network to handle the CTs participants' consent. To do so, we have realized the prototype using the following data definition model (Listing 2 shows a snippet of the CTO file containing the definition):

Assets: In the studied use case, the assets are represented by:

- Trial: contains the details about a clinical trial<sup>8</sup> It consists of id, description, list of its supervisors, status, and a list of phases. Each phase is characterized by description, starting and ending time, list of data types to be collected during the trial, and list of observers.
- Contract: represents the official agreement bounding the participant (and his/her data) with a clinical trial (and its supervisors/observers). It contains information about the demander (the one that proposes the trial), the recipient, the id of the trial, and a status indicating if the recipient has given consent.

Users: The categories of the users are doctor, researcher, patient, or admin.

Transactions: the model definition file also contains the declaration of all the transactions. These are used to change the state of the assets (e.g., update the status of a contract from *issued* to *accepted*), and to create participants.

---

<sup>8</sup>For the purposes of the study, a custom definition of the trial has been used.

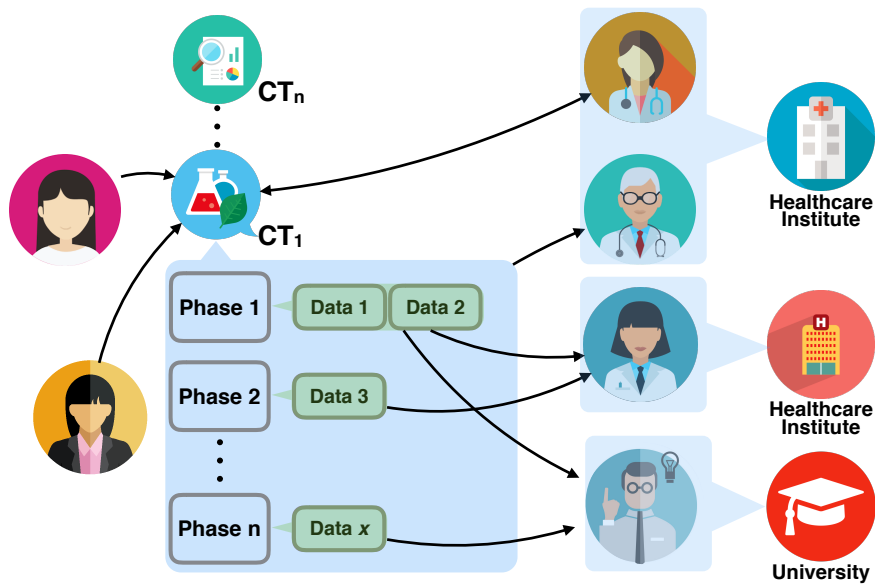


Fig. 9 Schematic representation of the implemented scenario.

```

1  /**
2   * The contract on a clinical trial
3   */
4  asset Contract identified by contractId {
5   o String contractId
6   o ContractStatus status
7   -> ScodesParticipant demander
8   -> ScodesParticipant recipient
9   -> Trial trial
10 }
11
12 /**
13  * A Clinical Trial
14  */
15 asset Trial identified by trialId {
16  o String trialId
17  o String duration
18  o String description
19  -> ScodesParticipant[] supervisors
20  o TrialStatus status
21  o Phase[] phases
22 }
23

```

Listing 2 Piece of code from the model definition file

#### 4.4 Integration with REDCap

REDCap has been widely used by the research community to design, realize, and test electronic data collection platforms to acquire/produce relevant (e.g., medical) research information. Such a platform has been chosen as database/world-state given its well-established position, robustness, involvement, and relevance in several other health-related projects. REDCap functionalities have been integrated into the prototype through JavaScript API provided by REDCap-Tools<sup>9</sup>. A subset of the data structure representing a patient’s medical data (see Section 4.3) has been mirrored in a new project on REDCap. In turn, two functions have been added to the API interface of the website to import/export data to/from REDCap. Such functions are used respectively by Patients and Doctors:

**Patients:** to insert medical data related to a trial to which they give consent.

Once they give consent to a trial, they can edit a form with their biometric data and send it to REDCap through the import function.

**Doctors:** to visualize the data related to the patients who gave their consent.

It worth to recall that doctors can only see data relative to trials that they are supervising. Moreover, if the patient revokes his/her consent, the doctors will no longer be able to access the data of such a participant.

Role	Actions
Administrator	Initialize REDCap project Create participants View all transactions View all participants View all assets
Doctor	Create/View trials Create/View contracts View participants View medical data (with consent) from REDCap
Patient	Accept/Revoke contracts (give/revoke consent) Send medical data to REDCap
Researcher	View medical data (with consent) from REDCap

**Table 3** Users types and related actions.

## 5 System Validation and Discussion

This paper proposed the integration and combination of BCT (Hyperledger Composer and Fabric) and web technologies (React, REST Server, and REDCap) to dynamically handle participants’ informed consent in CTs (require-

<sup>9</sup>REDCap-Tools is a non-official organization that provides several interfaces and project using REDCap, helping developers to exploit REDCap’s advanced functions to their full potential (Burns et al., 2019).

ment R1.1). Figure 10 shows the required interactions among the users and the system components to test a generic use-case scenario.

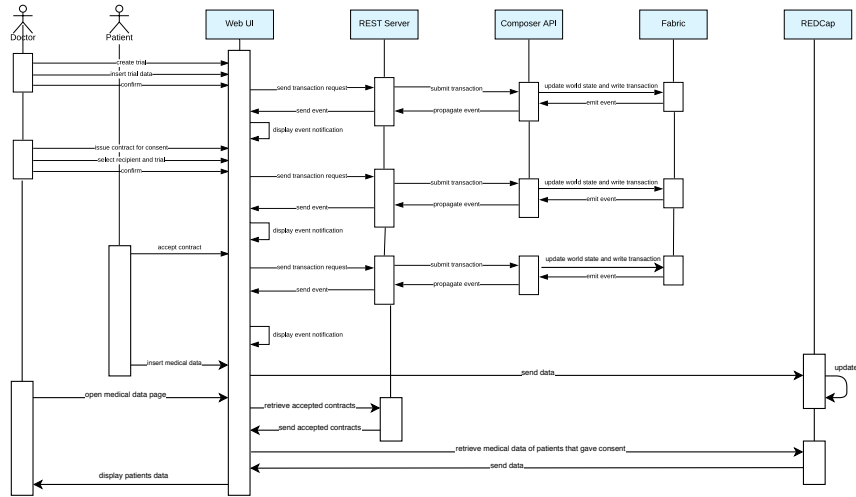


Fig. 10 Use case scenario sequence diagram

The test involves the following steps:

**Trial creation:** a doctor creates a new CT through the dedicated form in the Trials page. By doing so, the doctor can specify the characteristics of the trial (as described in Section 4). The information composing the Trial goes through all the layers of the network until it is stored in the world-state DB of the blockchain (requirements R3.1 and R3.2), and the respective transaction is submitted on the ledger, guaranteeing authenticity, integrity and trust (requirements R3.3, R3.4, and R3.5). In turn, the Fabric network emits an event that is propagated from the REST Server to the Web UI via a web socket. The Web UI then shows a pop-up notification that informs the user that the transaction has been successfully submitted. If a user receives a notification when he/she is not logged in, he/she can still visualize it by clicking on the bell icon on the top right corner).

**Contract issue:** to access and treat participants' data, an investigator needs to issue a contract to all of them. To do so, investigators can use the dedicated form on the Contracts page. A contract consists of a demander (the doctor issuing the contract), a recipient (the participant), and a CT. Both participants and trials are selected from a list of existing profiles and trials loaded from the blockchain network. Once the details are confirmed and the form submitted, the transaction follows the same workflow described above.

**Consent:** a participant can accept or reject to give the consent demanded by the contracts on the Contracts page. In such a page, the participant has listed all the contracts in which he appears as a recipient (this access rule is defined in the business network model). Thus, by clicking on the row related to a given contract, a view containing a summary of the CT involved in the contract pops up, giving the possibility to accept or revoke it.

**Data retrieval:** if the participant accepts the contract (giving the consent - requirements R4.1 and R4.2 ), he/she can insert the required medical data (requirement R1.2). In the Medical Data page, the participant can fill a form containing biometric data (currently using synthetic data). When submitted, this data are sent directly to REDCap and stored as a record (requirement R2.1). When a doctor goes to his Medical Data page, the system retrieves (from REDCap - requirement R4.3) and displays the data relative to all the patients that accepted the doctor's contracts (possible to filter). In the future, if a patient revokes the consent relative to a particular trial, the doctor is no longer able to see data from that patient on the specific trial (requirement R4.4).

Basic scenarios similar to the one presented above have been used to show how some information go through all the layers of the network, thus highlighting the most relevant interactions. Moreover, during the execution of those scenarios (with *alpha* and *beta* testers), it has been recorded participants' feedback on the single functionalities and experience with the overall presented prototype.

The high modularity of the presented prototype can ease the element substitution (e.g., in the case of constrained technological requirements) and the interaction with a broad set of third-party tools. The implementation of a modular CMS increases the possibility of supporting the development of new solutions exploiting different technologies (R2.3). Among the interesting future integrations, we could list:

**IoT (Internet of Things):** one of the most interesting integrations for the prototype presented is with IoT systems. In particular, Healthcare or more in general, Ambient Assisted Living (AAL) domain might require the use of wearable devices and sensors (e.g., to monitor patients in their daily routine). The data collected from those distributed devices could be combined with a blockchain infrastructure to enhance traceability and privacy. Moreover, automatizing secure and tamper-proof data sharing would facilitate doctors and researchers in collecting and elaborating the medical information (e.g., producing statistics via smart contracts).

**Blockchain as a Service (BaaS):** as described in Section 2, several CTMS and EDC software offer somewhat similar functionalities (yet neglecting the consent management). Nevertheless, the system presented in this paper shows that it is possible to improve consent management, leveraging on new technologies like the blockchain. However, the implementation of a blockchain infrastructure from scratch can be a demanding task, especially for complex systems and big companies. To overcome this obstacle, a feasible solution would be to provide blockchain features and functionalities as a service (requirement R2.2). For instance, the SCoDES prototype has been realized as a whole web application, integrating every aspect (currently in a simplified way) concerning the clinical trial workflow. An alternative approach could be to develop a blockchain service, providing a *consent management module* possible to be *plugged* into any existing or future platform with a lightweight integration process. In this way, it would be possible to implement a standalone CMS service and integrate it with an existing or tailored CTMS (requirement R4.5).

**Multi-Agent Systems (MAS):** a multi-agent system is a set of intelligent agents distributed in an environment, interacting with each other within or cross-organization. An agent is an, even partially, autonomous entity, represented by a program, a robot, or even a human being. In the specific use case, measuring devices and sensors could be programmed as agents communicating with each other and with the blockchain. In this way, it would be possible to exploit all the advantages of intelligent agents for handling sensitive data and also performing dynamic evaluations without forcing the test subjects to manage their data manually.

Analyzing the prototype, design- and implementation-wise, it has been possible to map the requirements presented in Section 3 on the implemented features.

Yet, many challenges still need to be addressed. For example, do we *really* need blockchain for CTs? Besides the advantages shown in managing the digital

consensus via BCT, the overall answer is not straightforward. A deeper analysis highlights multi-level concerns:

Cross-country laws: CTs and processes involving the use of sensitive data are strictly regulated, often involving public institutions and regulations (which may differ from one country to another). To this end, the challenge consists of validating the blockchain-based approach in a way that could support or even substitute all the bureaucracy that is behind. Yet, how to adopt/adapt the blockchain in cross-country scenarios (where data constraints can differ remarkably) is still an open challenge.

BCT/standards: currently, a plethora of blockchain frameworks (infrastructure) are competing to hit the market. However, there is a lack of standardization in integrating BCT in business scenarios. Hence, the typical designing phases that are encountered in traditional software engineering and development are not clearly defined. Thus, reducing the efficiency of developing a blockchain infrastructure also leads to possible higher costs (money/time). Healthcare companies or research institutes might instead choose to implement their system relying on more traditional and consolidated frameworks (e.g., conventional database with access control) to reach *comparable* results. Finally, the ability to choose a blockchain technology over another also depends on the interoperability capabilities of the platform. As shown in the prototype presented in this paper, it is possible to integrate already existent platforms to cooperate with the system (e.g., REDCap). However, the scale of a proof of concept is not comparable with the structure of a real CTMS. To study all the intersection points (third-party systems - BCT) and to prove that every part of the software and interactions complies with regulations such as GDPR (General Data Protection Regulation) and HIPAA (Health Insurance Portability and Accountability Act) standard might be overwhelming.

## 6 Conclusions

In this paper, we have presented a full implementation of a consent management application for clinical trials, based on blockchain technology. The resulting proof-of-concept addresses the following challenges related to consent management:

- Improve current procedures for managing consent while ensuring confidentiality;
- Enable sensitive data sharing between medical platforms as electronic data capture software (e.g., REDCap);
- Ensure trust in the data sources identities;
- Ensure traceability and integrity of data through a tamper-proof system.

This study shows that the use of blockchain technology within a network of researchers and healthcare institutions/practitioners can facilitate both the

process and collection of patient data and ensure the level of anonymity required for clinical trials and the reproducibility of the research. Thanks to the BCT, the control data sharing for scientific research is simplified, while privacy is ensured in cross-source.<sup>10</sup>

Finally, as described in Section 5, this study identifies as further developments the integration of BCT for managing medical data gathered through IoT technologies in combination with intelligent multi-agent networks as well as the implementation of trusted services through the paradigm of BaaS.

**Acknowledgements** The authors want to acknowledge the SCoDES project supporting this study.

### Conflict of interest

The authors declare that they have no conflict of interest.

### References

- Agbo CC, Mahmoud QH, Eklund JM (2019) Blockchain technology in healthcare: A systematic review. In: Healthcare, MDPI, Basel, vol 7, p 56
- Angeletti F, Chatzigiannakis I, Vitaletti A (2017) The role of blockchain and iot in recruiting participants for digital clinical trials. In: 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, pp 1–5
- Anjomshoae S, Najjar A, Calvaresi D, Främpling K (2019) Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, ACM, New York, NY, USA, pp 1078–1088
- Association WM (2013) Wma declaration of helsinki - ethical principles for medical research involving human subjects. <https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-research-involving-human-subjects/>
- Atasoy H, Greenwood BN, McCullough JS (2018) The digitization of patient care: A review of the effects of electronic health records on health care quality and utilization. *Annual review of public health* 40:487–500
- Beierle F, Tran VT, Allemand M, Neff P, Schlee W, Probst T, Zimmermann J, Pryss R (2019) What data are smartphone users willing to share with researchers? *Journal of Ambient Intelligence and Humanized Computing* pp 1–13
- Benchoufi M, Porcher R, Ravaud P (2017) Blockchain protocols in clinical trials: Transparency and traceability of consent. *F1000Research* 6

---

<sup>10</sup>Data coming from different platforms concerning the same user.



- Burns S, Beasley W, Zhu H (2019) Redcap-tools. <http://redcap-tools.github.io/>
- Cachin C, Vukolić M (2017) Blockchains consensus protocols in the wild. arXiv CoRR abs/1707.01873:1707.01873, URL <http://arxiv.org/abs/1707.01873>, 1707.01873
- Calvaresi D, Cesarini D, Sernani P, Marinoni M, Dragoni AF, Sturm A (2017) Exploring the ambient assisted living domain: a systematic review. *Journal of Ambient Intelligence and Humanized Computing* 8(2):239–257
- Calvaresi D, Dubovitskaya A, Calbimonte JP, Taveter K, Schumacher M (2018) Multi-agent systems and blockchain: Results from a systematic literature review. In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, Cham, pp 110–126
- Calvaresi D, Calbimonte JP, Dubovitskaya A, Mattioli V, Piguët JG, Schumacher M (2019a) The good, the bad, and the ethical implications of bridging blockchain and multi-agent systems. *Information* 10(12):363
- Calvaresi D, Mualla Y, Najjar A, Galland S, Schumacher M (2019b) Explainable multi-agent systems through blockchain technology. In: *Proc. of Explainable, Transparent Autonomous Agents and Multi-Agent Systems. EXTRAAMAS 2019*, Springer, Cham, vol 11763, pp 41–58
- Casino F, Dasaklis TK, Patsakis C (2018) A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telematics and Informatics*
- Compert C, Luinetti M, Portier B (2018) Blockchain and gdpr
- Crosby M, Pattanayak P, Verma S, Kalyanaraman V, et al. (2016) Blockchain technology: Beyond bitcoin. *Applied Innovation* 2(6-10):71
- Davis TC, Berkel HJ, Holcombe RF, Pramanik S, Divers SG (1998) Informed consent for clinical trials: a comparative study of standard versus simplified forms. *JNCI: Journal of the National Cancer Institute* 90(9):668–674
- Drosatos G, Kaldoudi E (2019) Blockchain applications in the biomedical domain: a scoping review. *Computational and structural biotechnology journal*
- EU (2019) Eugdpr. <https://eugdpr.org/>
- Friedman LM, Furberg C, DeMets DL, Reboussin DM, Granger CB, et al. (2010) *Fundamentals of clinical trials*, vol 4. Springer, Switzerland
- Harris PA, Taylor R, Thielke R, Payne J, Gonzalez N, Conde JG (2009) Research electronic data capture (redcap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics* 42(2):377–381
- IBM (2019) Hyperledger framework. <https://www.hyperledger.org/about>, accessed: 2019-04-30
- Inc F (2019) React - a javascript library for building user interfaces. <https://reactjs.org/>
- Jahankhani H, Kendzierskyj S (2019) Digital transformation of healthcare. In: Jahankhani H, Kendzierskyj S, Jamal A, Epiphaniou G, Al-Khateeb H (eds) *Blockchain and Clinical Trial: Securing Patient Data*, Springer International Publishing, Cham, pp 31–52, DOI 10.1007/978-3-030-11289-9\\_2, URL [https://doi.org/10.1007/978-3-030-11289-9\\_2](https://doi.org/10.1007/978-3-030-11289-9_2)

- Jaschinski C, Allouch SB (2019) Listening to the ones who care: exploring the perceptions of informal caregivers towards ambient assisted living applications. *Journal of ambient intelligence and humanized computing* 10(2):761–778
- Kaye J, Whitley EA, Lund D, Morrison M, Teare H, Melham K (2015) Dynamic consent: a patient interface for twenty-first century research networks. *European Journal of Human Genetics* 23(2):141
- Krenn R (2014) Design and development of a web-based clinical trial management system. PhD thesis, Graz University of Technology, DOI 10.13140/2.1.4306.2723
- Lee E, Yoon Y (2019) Trusted information project platform based on blockchain for sharing strategy. *Journal of Ambient Intelligence and Humanized Computing* pp 1–11
- LLC O (2019) Openclinica reference guide. <https://docs.openclinica.com/>
- Lorell BH, Mikita JS, Anderson A, Hallinan ZP, Forrest A (2015) Informed consent in clinical research: Consensus recommendations for reform identified by an expert interview panel. *Clinical trials* 12(6):692–695
- Maslove DM BKMP Klein J (2018) Using blockchain technology to manage clinical trials data: A proof-of-concept study. *JMIR Med Inform* 6:e11949, DOI 10.2196/11949
- Mulder T, Tudorica M (2019) Privacy policies, cross-border health data and the gdpr. *Information & Communications Technology Law* 28:261–274
- Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system
- Neisse R, Baldini G, Steri G, Miyake Y, Kiyomoto S, Biswas AR (2015) An agent-based framework for informed consent in the internet of things. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), IEEE, NJ, USA, pp 789–794
- Nugent T, Upton D, Cimpoesu M (2016) Improving data transparency in clinical trials using blockchain smart contracts. *F1000Research* 5
- Pocock SJ (2013) *Clinical trials: a practical approach*. John Wiley & Sons, Chichester, UK
- Rantos K, Drosatos G, Demertzis K, Ilioudis C, Papanikolaou A, Kritsas A (2018) Advocate: A consent management platform for personal data processing in the iot using blockchain technology. In: *International Conference on Security for Information Technology and Communications*, Springer, pp 300–313
- Richardson L, Ruby S (2008) *RESTful web services*. " O'Reilly Media, Inc.", CA, USA
- University V (2019) Redcap - research electronic data capture. <https://www.project-redcap.org/software/>
- Vimal S, Srivatsa S (2019) A new cluster p2p file sharing system based on ipfs and blockchain technology. *Journal of Ambient Intelligence and Humanized Computing* pp 1–7
- Yaga D, Mell P, Roby N, Scarfone K (2019) Blockchain technology overview. arXiv CoRR abs/1906.11078:1906.11078, URL <http://arxiv.org/abs/1906.11078>