ComeHere: exploiting Ethereum for secure sharing of health-care data

M. Franceschi[1]✉, D. Morelli[1,4], D. Plans[1,4], A. Brown[1], J. Collomosse[2], L. Coutts[2], and L. Ricci[3]

[1] University of Surrey/Center for Digital Economy, Guildford, UK
mfranceschi94@gmail.com, {d.morelli, d.plans, alan.w.brown}@surrey.ac.uk
[2] University of Surrey/Center for Vision, Speech and Signal Processing, Guildford, UK {j.collomosse, l.v.coutts}@surrey.ac.uk
[3] University of Pisa/Department of Computer Science, Pisa, Italy
laura.ricci@unipi.it
[4] BioBeats Group LTD, London, UK {davide, david}@biobeats.com

Abstract. The problem of protecting sensitive data like medical records, and enabling the access only to authorized entities is currently a challenge. Current solutions often require trusting some centralized entity which is in charge of managing the data. The disruptive technology of blockchains may offer the possibility to change the current scenario and give to the users the control on their personal data.
In this paper we propose ComeHere, a system able to store medical records and to exploit the blockchain technology to control and track the access right transfer on the blockchain. The paper shows the current status of the project, presents a preliminary proof-of-concept implementation and discusses the future improvements of the system, and some critical issues which are still open.

Keywords: Ethereum · Healthcare · Blockchain

1 Introduction

The huge amount of personal data produced by different networked services, (social networks, health care services, selling services,...), are currently scattered among numerous data servers owned by different companies. The end users do not really have the control over their data and have to trust several entities which manage them, hoping they maintain the privacy of users' data. However, their trust is, in many cases, misplaced, because these entities handle users' data in a hardly controllable and verifiable way. Logging events generated by users to keep track of how data is used may be a solution, but in general, it is not implemented, and, in any case, no guarantee against log tampering is given.

A nagging problem is that of sensitive data such as medical records which may be scattered between numerous servers and encoded in different ways. When a user moves from a healthcare entity to another one, often he or she has to ask

for their own data. Furthermore, research groups need to gather lots of medical data from a large number of individuals to support their researches and patients have to trust those groups to maintain their privacy and data secure. However, the tools to guarantee that only the proper entities access data and that gathered data remains anonymous are often inadequate.

Medical data pose several challenges, some of the most important are summarized in [2]. These data are sensitive and should be handled with the highest security standards. In particular, amongst other requirements, users should be able to delete their data; data should be exempt from tampering; only allowed entities should be able to access and process users data; all additions, deletions, and modifications of data should be logged, and all actors participating in the data handling system should be accountable for their actions. A system designed to fulfill security requirements would only allow access to a restricted set of agents, requiring explicit user consent to grant access to new agents. However, there are several cases in which requiring explicit consent could be impractical or would be detrimental to users best interests, e.g. to allow access to medical doctors in an emergency, or to allow researchers to develop drugs that the user might need in the future.

The blockchain technology [1] has recently been proposed as a solution to many of these problems. At a high level, the blockchain is a publicly accessible, append-only, tamper-free, distributed and replicated ledger of the same type of data. Using the blockchain technology and its characteristic to support tracking of medical data is a novel research field. Even without completely eliminating a centralized trusted entity, the blockchain technology could help to define a public and trusted log, storing how users data is accessed and shared, enforcing accountability for data access. Of course, control over data has to be guaranteed to their real owners. Data may be encrypted and shared on the blockchain so there is no way for people except who knows the right decryption key to access the data. Furthermore sharing data on the blockchain must be done carefully because, as already mentioned, it's an append-only data structure so it's impossible to remove data from it.

The ComeHere project exploits the blockchain technology to create a shared, trusted log of all the actions made over the data. This log is guaranteed to be tamper-proof and the creation of its record is delegated to trusted code running on the blockchain therefore not modifiable and publicly audible by every party that use the system. The difference between a traditional storage system and ComeHere is that the way data is shared, requested, saved etc. is publicly audible and impossible to tamper.

The structure of the paper is the following: Section 2 presents a brief introduction to blockchain technology, in particular Ethereum, and analyzes some existing proposals close to our system. Section 3 describes the general structure and architecture of ComeHere focusing, in particular, on the role of each smart contract. Section 5 shows how the system is implemented and the necessary steps to execute each action available to the users. Some preliminary results are

presented in Section 6, and finally, Section 7 concludes the paper pointing out some open problems and future directions that ComeHere will take.

2 Background and Related Work

2.1 The Ethereum Blockchain

Blockchain technology allows creating an immutable, distributed and secure database of arbitrary data. A distributed consensus protocol gives the rules on how to manage and update the database (in particular decides what valid data is to add). The first usage of the blockchain technology was as a support for the Bitcoin cryptocurrency [6], in this scenario the database is used to record the transaction between entities. ComeHere exploits the Ethereum blockchain [7] which takes the idea of the Bitcoin blockchain and adds the possibility to execute distributed applications defined through a Turing complete programming language. The protocol has *ether* as currency. Entities in the blockchain, masked with a pseudonym (addresses), can exchange value between them by sending transactions to the network which are validated and update the global state. To validate a transaction, miners in the network have to execute the Ethereum consensus algorithm called *Ethash* which is based on PoW (i.e. Proof-of-Work). The main characteristic of the Ethereum protocol is the possibility to use the blockchain not only to store value but also code. Users can send transactions carrying executable Turing complete code, the *smart contracts*. Transactions can create smart contracts which deploy their code and link that to a public address. Any new transaction sent to this address triggers the execution of one of the functions inside the deployed smart contract. Transactions carry the function parameters needed for the execution. To validate such transactions the code has to be executed, and this can possibly update the global state which can be seen as the state the EVM, i.e. the Ethereum Virtual Machine, a virtual machine that runs all the code stored inside the blockchain. Miners actually execute the code inside the smart contract and they are rewarded for this: to each instruction of the EVM it's assigned a price proportional to the difficulty of that instruction. This price is called *gas*. So a transaction will have a total gas cost which is the sum of gas of every single instruction that has to be executed. Each transaction specifies two parameters *gas limit* and *gas price*, the former is the maximum amount of gas the transaction is allowed to consume, the latter is the amount that the user creating the transaction is willing to pay for each gas consumed.

2.2 Blockchain and Access Control

The use of blockchain for giving permission and access to data has been recently proposed, in particular for sensible records like health care data. The main reason why we are trying to use this technology instead of standard and consolidated ones is that the blockchain gives audibility and removes the need for trusting who maintains personal data. The code is publicly readable and checkable, hosted in

the blockchain, we don't need anymore to trust closed systems that promise functionalities we don't have any control over. A general approach to the use of blockchain as an access control is presented in [3, 4] where they implement the standard XACML directly in the blockchain. In [5] the authors implement a typology of smart contract that pairs each patient to a medical provider and stores pointers to data (as SQL queries) of the patient in the provider's database. [8] proposes a system where private blockchains are used to directly store the data encrypted and HDGs (i.e. Healthcare Data Gateway) which are off chain software are used as gatekeeper for the access to such data.

3 The ComeHere System: general architecture

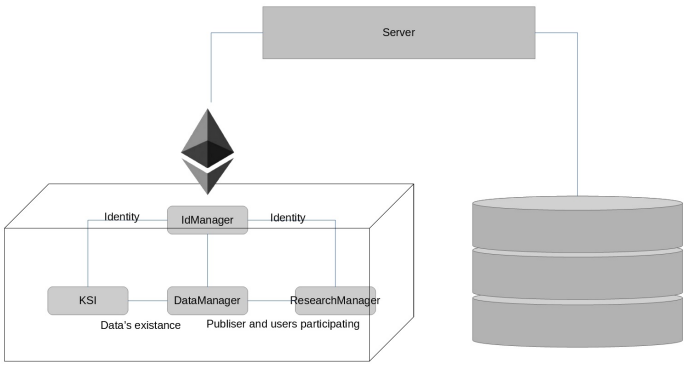

Fig. 1. Representation of the interaction and the structure of the ComeHere system

This paper proposes the ComeHere system, the Figure 1 shows an high level representation of the system architecture. The system includes a trusted centralized server that maintains personal and sensitive healthcare data and exploits the Ethereum public blockchain to create a public audible log of the history of the data managed by the system. This log is made using 4 smart contracts communicating with each other, the server queries the smart contracts to give access to data and the rights to access any data can be verified by anyone checking the smart contract.

The aim of the system is to transform the provision of personal healthcare by commodifying and brokering personal healthcare data (e.g. from WBS, mobile devices or the IoT) to healthcare providers, enabling them to optimize preventative healthcare and helping to achieve a more efficient healthcare system. The system establishes the technical, logistic and socio-economic feasibility of the use of a public blockchain as an access-control manager to personal healthcare data. In particular, ComeHere aims to change the present paradigm where healthcare providers have to find people to base their study on. This system offers to

healthcare providers an already established pool of people which will be informed about the researches the providers are carrying on. Who is willing to participate decides to opt-in giving the authorization to share personal and anonymous data to them.

There are two types of actors in the system: *institutions* which use the system to gather data to support their research and *participants* which take part in researches and create data for them.

The system is formed by a centralized server and a blockchain where four smart contracts have been deployed. The functions of the smart contracts are the following ones:

- *IdManager*. Maintains the digital identity of the registered users
- *ResearchManager*. Maintains public information about published researches and the list of participants of each research
- *KSI*. Keyless Signature Infrastructure, maintains information about submitted data and stores the hash of all the data submitted as an anti-tamper proof
- *DataManager*. Maintains authorization to store and retrieve data, and implements the real log of the access and storage of all the data managed by ComeHere.

Every user is publicly identified by their Ethereum address. The personal information which links an address to a person or institution is maintained securely in the centralized server. The registration is done sending personal data to the server that stores it in its database, and a message signed with their Ethereum private key which proves they own the address. The server may request additional information from the user and the registration is completed when the address is saved on the *IdManager*. After the registration, only the Ethereum address is required to interact with the system.

Institutions can publish researches by sending a transaction to the *ResearchManager* smart contract with all the needed information that are saved directly on the blockchain. The server maintains in its database a link to the published research. Participants can request the list of the researches to the server which gathers all the information from the *ResearchManager* and sends them to the participant. If a participant decides to opt in a research, he/she sends a transaction to the *ResearchManager* which logs that he/she is now participating in that research.

Participants are requested to publish data for researches. Data is stored in the server database, while the blockchain is used to guarantee their correct usage. First, the hash of the data is sent to the *KSI smart contract*. This proves that data won't be changed when submitted and after it is saved. Then data is submitted to the server which checks the identity of the submitter and that the hash of the data exists. The server logs on the blockchain the reception of the data. At this point, the participant submits an authorization to the blockchain that requests to save his data. The server sets in the blockchain a unique id that identifies the data in its server and adds this id to the list of the research data's id.

To retrieve the data submitted for their researches, institutions send a transaction to the *DataManager* to log and authorize the request to retrieve data. Then, they send the request to the server which asks the *DataManager* to check if the authorization has been submitted, if the authorization is found, the server retrieves the list of data's id from the blockchain and sends those data, saved in its database, to the institution.

4 ComeHere: the Smart Contracts

Each smart contract defines a set of public functions that can be called by every registered user and some private functions reserved to the server which is the only one that has the right to use them. Examples of private functions are those setting the unique ID of some data or registering new users in the system. Those functions can't be public but are a prerogative of the server.

4.1 Keyless Signature Infrastructure

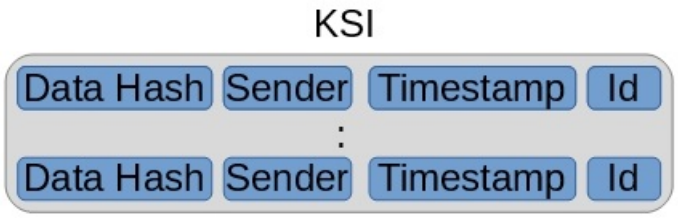

Fig. 2. Data Stored in the KSI smart contract

This smart contract allows registered users to publish the hash of their data, those hash are also timestamped and the sender address is logged. The hash is used as an anti-tamper proof for the data. The centralized server can invoke two more functions of the smart contract which permit to log the event corresponding to the reception of the data and to set the data unique id when it is saved.

4.2 IdManager

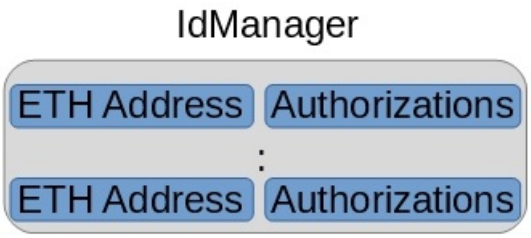

Fig. 3. Data Stored in the IdManager smart contract

The smart contract saves the registered users of the system with their roles and the permissions set during the registration process. Only the server has the rights to register new users. In particular, institutions that retrieve data first need to be identified by the system. The smart contract permits also to check the role of every registered user. No personal information is stored in the blockchain: users are identified by their address and the server maintains a link between the address and the personal information.

4.3 ResearchManager

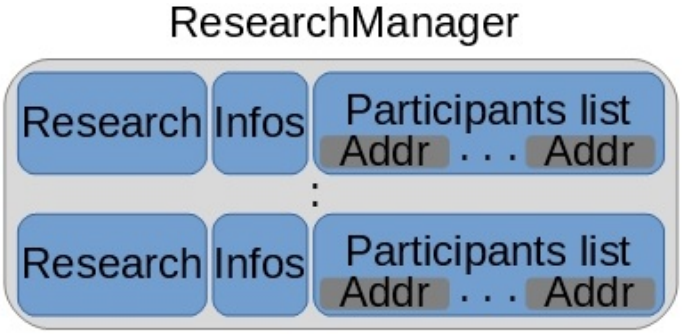

Fig. 4. ResearchManager smart contract is used to publish researches. It stores research's public info and the Ethereum address of each participant is added to the Participants list

The smart contract allows registered institutions to publish new researches. The data they provide is the research period and an URL where the user may find further information about the research. Participants can utilize the smart contract to participate in published researches.

4.4 DataManager

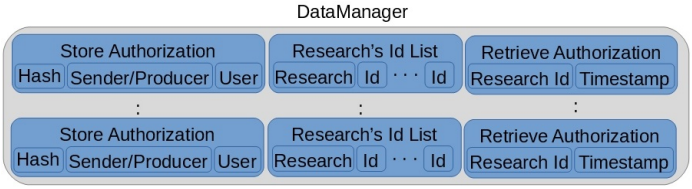

Fig. 5. DataManager has three different type of data structures: Store Authorization used to store submitted data, Research's Id List which maintains for each research the list of data Id relatives to that research and Retrieve Authorization used to log when a retrieve request is made.

The smart contract logs requests to store and retrieve data from the server database. Participants send a request to save data submitting the hash and the project id of the data. The smart contract checks if the user is authorized to perform this action and if the server has previously received the data. Institutions can submit authorization to retrieve data for their researches, the smart contract checks if the institution has the rights to send the authorization, in particular if the research has started and has not finished yet. The server uses this smart contract to set the unique ID of the data and also adds that ID to the list of research's data. This smart contract is also used when an institution asks the server to get the data for its research. The smart contract first checks the existence of an authorization submitted within a day and retrieves the list of ids of data to send to the institution.

5 ComeHere: Implementation

The server is implemented as a simple HTTP server which already gives all the functionalities described, but it can also be used by sending requests directly to it, without using a browser. The server doesn't need to implement any registration or cryptographic protocol to identify the user that is interacting with it, because the blockchain provides all the security and identification needed. The interaction between the server and the blockchain is realized by exploiting the standard*web3.js* library and *truffle-contract*[5] library to instantiate and use the smart contracts. The smart contracts are written using Solidity. Solidity is a contract-oriented language which is compiled into a bytecode executable by the Ethereum Virtual Machine.

The server monitors in real time when a new research is published and a new request to save data is received. The identification of a user requires the server to request from that user a signed message with its Ethereum address private key

[5] https://github.com/trufflesuite/truffle-contract

which proves its identity and the signature is directly controlled by the smart contracts code as an additional trust check.

The website pages provided by the server use *Metamask* plugin[6] which injects its own implementation of the standard *web3.js* library to interact with the smart contracts and Ethereum blockchain.

As already mentioned, even if the server does not expose a proper API at the moment, it's already possible to send requests directly from the application without standard authentication protocols. This is possible because each request made to the server will be authenticated using the blockchain smart contracts.

5.1 Registration

The registration is made by the server sending a transaction to the IdManager smart contract. The address of the user to register is sent in the transaction along with a message signed with that address's private key and parameters chosen by the server to set the permissions of the new user (e.g. possible permissions are the ability to create or retrieve data and to publish researches). The smart contract checks that the signed message is valid and registers the user in the system.

5.2 Publishing Research

A registered institution with the authorization of publishing researches can send a transaction to the ResearchManager smart contract to publish a new research. The smart contract checks the authorization of the user, then stores all the information in the public blockchain. The smart contract also fires an event that informs that a new research has been published and the server stores the id of the new research in its database. Other applications can also receive the event and act accordingly, for example, a notification server can wait for those events and push a notification on the mobile phone of a user.

5.3 Participating in a Research

A registered user can send a transaction to the ResearchManager smart contract to opt-in a research. The smart contract checks that the user is registered in the system and then adds its public address to the research's list of participants.

5.4 Submitting Data For Research

A user submits data to the system by sending to the *KSI smart contract* a transaction with the hash of the data. This contract first checks if the user is registered and has the authorization to submit data. If this condition is true, it stores the hash of the data with its timestamp and the address of the user.

[6] https://metamask.io

Afterwards, the user sends the data to the server with its hash signed with its private key.

The server again checks that the user is registered and authorized, then verifies the signature and if all these conditions are satisfied, it sends a transaction to the KSI smart contract to log that it received the data, then it sends a positive response to the user. The user, at this point, can send an authorization request to save its data to the *DataManager smart contract* calling the method *submitAuthorization*. This method includes the hash of the data to save and the id of the research for which data is saved. The smart contract checks that the user of the data is participating in the research he is submitting data for and that the sender of the data is also the owner of it. Then checks that the hash of the data exists and has been sent by the same user who sent this message and that has been received by the server. In this case, it stores the new authorization and fires an event that requests to save the sent data. The server waits for these events and it is the only authorized to submit a transaction to the *DataManager smart contract* which adds the Id of the data to the research's id list and sets the same Id on the *KSI smart contract*.

We plan to extend the ComeHere system to let users called *Data Producers* to submit data on behalf of other participants and this is the reason for the complexity of the procedure to submit data. If users submit data for themselves, it is enough to publish the hash of data to the *KSI smart contract* and then save it to the database. In the more general case, where Data Producers submit data on behalf of other users, they have to send the authorization request and the end user decides to authorize to save the data or not.

5.5 Retrieving Data For Research

When an institution wants to retrieve the data that has been previously submitted for its research, it needs first to submit an authorization request to the *DataManager smart contract*, the authorization is used as a timestamped retrieval requests to be stored on the log. The institutions send a transaction which includes the ID of the research to retrieve data for and the *DataManager* checks that the sender is also who published the research and that the research has not been yet concluded. Note that each authorization is timestamped and is valid for a day. Then the institution sends the request to the server which requires it to sign a message with its personal Ethereum address private key to prove its identity, then ask the DataManager to retrieve the research's list of data Id. The DataManager checks again that authorization not yet expired exists, and, in this case, it sends to the server the list of data's id to send to the institution. In this case, the authorizations to retrieve data are just a log of the number and time of the requests because when a user decides to participate in a research it gives the consent to give the institution its data.

As a future work, ComeHere will also permit institutions, medical staff or others entities called *Data Retriever* to request data from a user. In this case, the authorization request is stored on the blockchain and the owner of the data decides to give the authorization to access its data.

6 Experiments

We conducted a preliminary set of experiments on a 'proof of concept' system including the server and a private Ethereum network running on our machines, with just a node which validates all the incoming transactions. Even if in our test every transaction is instantly mined, we can evaluate the average time required to execute an operation like submitting new data to be at least 50 seconds in a real Ethereum network, where a new block is mined approximately every 10 seconds.

We created 50 users that submitted random data each day for 9 days for a single project. The simulation showed that with a fixed gas price of 5 gwei for transaction (the standard gas price swings between 2 and 5 gwei in this moment) a total of 0.376352 ether was spent by the users which is equivalent to 283,85$ (at an ethereum price of 755,35$[7]) each user spent 0.00752832 ether which is 5,68$. This may be an issue with the utilization of this system because people won't willingly spend their own money to give data for free for researches. Including a reward system for users or using another type of blockchain, for instance, a permissioned one could resolve this problem.

Over the time of the simulation, the amount of space occupied in the blockchain global state was 259.2 kbyte, not counting the space occupied by each block mined during the process. We need to take in consideration the space that this system occupies because blockchain space is expensive. The blockchain is replicated in every node of the network so 259.2 kbyte are replicated in thousands of others machines. We plan to solve this problem by defining a hierarchical solution including user's node which does not store an entire copy of the blockchain but connects to full nodes which maintain a full copy of the blockchain. Users' nodes can send transactions and interact with the smart contracts using full nodes.

7 Conclusions and Future Work

In this paper we presented ComeHere, a work in progress system to maintain medical data secure and give permissions over them through the use of the blockchain technology. The current implementation is a system where users publish their own medical data. In a real scenario, users never create medical and biometrical data for themselves but hospitals, medical staffs, care services create data for them. In the future, we plan to modify the system so that these third parties could create, submit and also retrieve data for users. The system offers a trusted environment where institutions may gather data for their research projects. The system gives the possibility to store and retrieve data to third parties so giving users a unique place where they can safely store and manage all their data. The user approves requests from third parties to submit or retrieve their data. The blockchain makes this possible, by registering the user's decision in a tamper-free distributed ledger. We plan to develop a full version of the server

[7] https://coinmarketcap.com/currencies/ethereum/

and to implement a real API. A mobile app will be created to interact with the system and also to enable interaction with other healthcare applications.

Some real-world problems have to be addressed before making ComeHere a publicly available system. For example, in every cryptocurrency, there is no way to recover a lost private key and who lose it automatically lose all his funds. Even if losing funds is already a big loss, it's not even comparable to losing access to all your medical history. A system must be designed to address this problem. The other problem that we need to address is the fact that we can keep track of data while it's inside ComeHere environment but when a user gives access to some data to third parties actually trusting those third parties and trusting how his data will be managed, we lose the traceability over that piece of data. After these and others minor problems have been addressed we think that ComeHere will be a really successful and groundbreaking platform that uses one of the most discussed and researched technology of this time.

Acknowledgements This research was partly sponsored by the United Kingdom's Engineering and Physical Sciences Research Council, grant number EP/P03196X/1, under the project: Co-operative Models for Evidence-based Healthcare Redistribution (CoMEHeRe), and also by BioBeats Group LTD, a UK company.

References

1. Bashir, I.: Mastering blockchain: Distributed ledgers, decentralization and smart contracts explained (2017)
2. Bujari, A., Furini, M., Mandreoli, F., Martoglia, R., Montangero, M., Ronzani, D.: Standards, security and business models: key challenges for the iot scenario. Mobile Networks and Applications **23**(1), 147–154 (2018)
3. Di Francesco Maesa, D., Mori, P., Ricci, L.: Blockchain based access control. In: DisCoTec 2017 12th International Federated Conference on Distributed Computing Techniques. vol. 10320 (2017)
4. Di Francesco Maesa, D., Mori, P., Ricci, L.: Blockchain based access control services. In: IEEE International Symposium on Recent Advances on Blockchain and Its Applications, Halifax, Canada (2018)
5. Ekblaw, A., Azaria, A., Halamka, J.D., Lippman, A.: A case study for blockchain in healthcare:"medrec" prototype for electronic health records and medical research data. In: Proceedings of IEEE Open & Big Data Conference. vol. 13, p. 13 (2016)
6. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
7. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151**, 1–32 (2014)
8. Yue, X., Wang, H., Jin, D., Li, M., Jiang, W.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. Journal of medical systems **40**(10), 218 (2016)