An open-source Blockchain as a Service solution for life science applications

Barbara Martelli, Davide Salomoni, Alessandro Costantini, Luca Giommi, Ana Velimirović
The Problem

• On 25th May 2018 a new European Union (EU) law came into effect: the General Data Protection Regulation (GDPR) with the aim of protecting EU individuals' privacy
  • EU individuals = whoever is on EU territory, not only EU citizens
• Many INFN life science research activities involve personal data processing related to healthcare patients and need to comply with the new norm
  • To address this issue, we have built EPIC Cloud (Enhanced Privacy and Compliance Cloud): an ISO 27001 27017 27018 certified partition of INFN Cloud adopting technical and organizational security measures that make it fit for processing personal data
• Among the rights of individuals, there are the rights To Be Informed, To Erasure and To Restrict Processing of personal data
  • We need to add functionalities to enable patients who provide personal data stored in EPIC Cloud to exercise their rights
The solution from an organizational process point of view

• To address the Data Sovereignty issue, the most frequent solution is to ask Data Subjects (in our case patients) to fill and sign a form named “Informed Consent” clearly stating
  • Purposes of data processing
  • Retention period
  • Who is the Data Controller and who is the Data Processor

• Any patient has the right to withdraw his consent at any time
  • In this case Data Controller and Data Processor must delete all his data immediately
The (partial) solution from a technical point of view

• To manage the Informed Consent workflow and lifecycle, the exploitation of a Consent Management System (CMS) is frequently proposed in literature
  • A CMS should enable data subjects to establish control over their data, giving access permission and audit the use of their personal data, withdrawing permissions and deleting their data
• However, even state-of-the-art CMS still suffer of lack of transparency
  • It is necessary to trust the CMS provider (usually a private company) for the effective deletion of data and compliance to GDPR
  • Trust is often based on the adoption of certification mechanisms (like ISO/IEC 27001), which foresee a third-party independent audit performed on a yearly basis
How to add transparency and trustworthiness to CMSs - The Blockchain

A way of enhancing transparency and trustworthiness to current CMS systems is the exploitation of blockchain technologies

- Distributed ledgers implemented as concatenation of data blocks in a growing chain of immutable elements
- The current value of all ledger values is called the World State (WS)

The chain is maintained by several actors exploiting a combination of

- cryptographic techniques
- consensus algorithms
- peer-to-peer communications
- game theory
## Two main open-source blockchain platforms

<table>
<thead>
<tr>
<th>Ethereum</th>
<th>Hyperledger Fabric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> permissionless</td>
<td><strong>Access:</strong> permissioned</td>
</tr>
<tr>
<td><strong>Deployment:</strong> public</td>
<td><strong>Deployment:</strong> private</td>
</tr>
<tr>
<td><strong>Consensus algorithm:</strong> Proof of Stake (PoS)</td>
<td><strong>Consensus algorithm:</strong> Practical Bizantine Fault Tolerance (PBFT) or dynamic (Sawtooth)</td>
</tr>
<tr>
<td><strong>Development language:</strong> Solidity</td>
<td><strong>Development language:</strong> golang or JavaScript + Hyperledger Composer</td>
</tr>
<tr>
<td><strong>Distributed App (DApp):</strong> Smart Contract</td>
<td><strong>Distributed App (Dapp):</strong> chaincode</td>
</tr>
</tbody>
</table>

Both FOSS backed by an international foundation
Ethereum and Hyperledger Fabric which is the best in our case?

We choose a mix of them: Hyperledger BESU
Hyperledger BESU: an Ethereum permissioned blockchain

• While Ethereum architecture (PoS, smart contracts, Solidity) is the most advanced, in our use case the public Ethereum blockchain (mainnet) has some drawbacks:
  • Users can be anonymous, while we need to identify, authenticate and give appropriate roles to all subjects accessing personal data
  • Performance are low and transactions can wait for several minutes to get committed -> eventual consistency
  • Transactions come with a cost: the coin is Ether (ETH) and you need conventional money to get it

• To overcome these limits, we are going to exploit Hyperledger BESU: a permissioned version of Ethereum blockchain that can be deployed in private environments
What is a DApp

- A DApp (Distributed Application) is an application built on a decentralized network that combines a smart contract and a frontend user interface. It is:
  - **Decentralized** – when DApps operate on Ethereum no one person or group has control, as it is an open public decentralized platform
  - **Deterministic** - DApps perform the same function irrespective of the environment in which they get executed
  - **Turing complete** - DApps can perform any action given the required resources
  - **Isolated** - DApps are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract has a bug, it won’t hamper the normal functioning of the blockchain network

- On Ethereum and Hyperledger BESU DApps are written in the **Solidity** language
- Solidity is **object-oriented, statically-typed and high-level**, with syntax influenced by JavaScript and C++

https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application
What is a Smart Contract

• A smart contract is **code** that lives on the blockchain. It contains some **business logic** and a limited amount of **data**. The business logic is executed:
  • if specific criteria are met by data stored in the blockchain
  • If Participants in the blockchain run the smart contract

• You can think of it as a DApp's backend
  • It's a **collection of code (its functions) and data (its state)** that resides at a specific address on the blockchain

• Once smart contracts are deployed on the blockchain network, **you can't change them**
First step:
make available a DApp
development environment on
INFN Cloud
Ethereum environment for developers: scaffold-eth

User

UI

Ant Design

Smart Contract

Solidity

React

Ethers

Blockchain
(single node)

Hardhat

Block n-1
- Header
- Previous Block Address
- Timestamp
- Nonce
- Merkel Root

Block n
- Header
- Previous Block Address
- Timestamp
- Nonce
- Merkel Root

Block n+1
- Header
- Previous Block Address
- Timestamp
- Nonce
- Merkel Root
Deploy Scaffold-eth via INFN Cloud

• Scaffold-eth containerization
  • To be portable and reusable
  • Contracts on github repo: every time a contract is created or modified, it will be automatically updated in Scaffold

• Scaffold and Nginx integration
  • Scaffold-eth and Nginx have been integrated in a docker-compose file
  • Nginx is taking care of the correct proxy redirections. It will provide also TLS termination as a further step

• Scaffold-eth deployment customization
  • Tosca template has been properly customized in order to deploy scaffold-eth application within the INFN Cloud infrastructure
Scaffold-eth “as a service” tool on INFN Cloud

Scaffold-eth can be deployed on-demand using the INFN Cloud dashboard and related graphical environment.
Deployment setup

Scaffold-eth

- CPUs
- RAM

Scaffold-eth Customization

Scaffold-eth Default values

- Project Name
- Env variables
- Service ports

Scaffold-eth Deployment created
Access Scaffold deployment

code:

```python
endpoint: http://[redacted]:8080
	node_creds

ssh_login: claudadm

ssh_private_key:

node_ip: [redacted]
```

Diagram:

- Click on the "Debug Contracts" tab.
- View the contract details, including purpose and setPurpose.
- Contract address: 0x7f...0512

Overall, the page provides instructions on how to access and interact with a Scaffold deployment, including connection details and contract interaction options.
Second step: define the CMS architecture and the Informed Consent workflow
Consent Management System

- A CMS supports users in controlling who can access their personal data and audit who has accessed their personal data by adding access control and transparency.
- Ideally a CMS shouldn’t be controlled by any individual/company, that’s why often blockchain approaches are considered for its implementation.
- A good practice in CMS is to decouple the consent layer from the data management layer.
- A CMS must be tamper-proof and auditable.
**Actors**

**Patient**

*GDPR Role: Data Subject*
- Give data to Hospital
- Sign the Informed Consent
- Revoke Consent
- Audit the CMS

**Hospital**

*GDPR Role: Data Controller*
- Collect patients’ data
- Appoint INFN Cloud as Data Processor
- Data Consumer: analyse Patients’ data exploiting the Life Science Datalake

**Hospital Ethic Committee**

*Role: Watchdog*
- Assign Roles on CMS and datalake
- Audit CMS

**INFN Cloud**

*GDPR Role: Data Processor*
- Get data from hospitals and manage them
- Develop and manage the CMS
- Develop and manage the life science datalake
Informed Consent Workflow

• Patients grant or withdraw consent to data consumers (doctors, researchers, insurance companies, ...) acting in particular roles

• Watchdogs (Ethic Committee) assign and revoke data consumers’ roles

• Given their roles, data consumers request permission to access data

• The CMS must determine whether such permission can be granted, based on patients consent

• A CMS must be tamper-proof and auditable
Informed consent smart contracts

The Solidity code for a basic consent management smart contract is here: https://github.com/bmartell/sc-dcms

The code is based on [3] We added Individual-oriented word state (IWS) as suggested by [1]
Conclusions (1/2)
The Big Picture: toward an open-source genomic datalake for research

INFN Cloud technologies are already the basis for several research projects at national and European level

- Harmony Alliance
- Health Big Data
- Several national projects funded under the EU Recovery and Resilience Plan

We believe that blockchain technologies will be of paramount importance to enhance transparency, auditability and trust at various architectural levels
Conclusions (2/2)
INFN Cloud Roadmap to exploit Blockchain technologies in life science applications

1. **BCDEaaS**: on-demand deployment of Blockchain Development Environments based on Ethereum and Scaffold-eth -> Done!

2. **BCaaS**: on-demand deployment of Blockchain general purpose, permissioned blockchain environments based on Hyperledger BESU -> work in progress

3. **BC-CMSaaS**: on-demand deployment of blockchain-based Consent Management Systems built on top of the BCaaS functionality -> work in progress

4. **BC-GIMSaaS**: Genomic Information Management System built on top of the BCaaS functionality and exploiting the BC-CMS to manage patient consent -> future work

ISGC 2023
Main References


• https://github.com/scaffold-eth/scaffold-eth
• https://docs.scaffoldeth.io/scaffold-eth/
• https://www.cloud.infn.it/