PROCEEDINGS OF SCIENCE

An open source blockchain as a service solution for health science applications

Barbara Martelli,[a,*] **Davide Salomoni,**[a] **Alessandro Costantini,**[a] **Luca Giommi**[a] **and Ana Velimirović**[a,b]

[a] *INFN-CNAF,*
Viale Berti Pichat 6/2, Bologna, Italy

[b] *Università di Bologna, Dipartimento di Farmacia e Biotecnologie,Via Belmeloro 6, Bologna , Italy*

E-mail: barbara.martelli@cnaf.infn.it, davide.salomoni@cnaf.infn.it, alessandro.costantini@cnaf.infn.it, luca.giommi@cnaf.infn.it, ana.velimirovic@studio.unibo.it

The use of big data in the field of omics and biomedical studies is the enabling factor for finding new insights with sufficient statistical confidence. When dealing with such data, several issues have to be addressed, related to the personal identifiable information (PII) often included in datasets and subject to the European General Data Protection Regulation (GDPR)[1], which imposes particular organizational and technical measures, aimed to protect patients' privacy. In this contribution we describe our roadmap for the exploitation of blockchain solutions on the INFN Cloud [3] infrastructure and the results gained so far using blockchain technologies to overcome present limitations in trustworthiness, auditability and transparency of some state-of-the-art life science applications like Consent Management Systems. Moreover, we'll show a decentralized application (DApp) for consent granting and withdrawal having the main target of managing personal information while preserving patients' rights, thanks to the trusted, tamper-proof, traceable and accountable distributed digital ledger provided by the blockchain.

International Symposium on Grids & Clouds (ISGC) 2023 in conjunction with HEPiX Spring 2023 Workshop
BHSS. Academia Sinica
19-31 March 2023

[*] Speaker

PoS(ISGC&HEPiX2023)028

1. Introduction

On 25th May 2018 a new European Union (EU) law came into effect: the General Data Protection Regulation (GDPR) [1] with the aim of protecting privacy of EU individuals, that is whoever is on EU territory, not only EU citizens. As a research institution, the Italian National Institute of Nuclear Physics (INFN) is involved in several life science research activities requiring to analize large volumes of personal data related to healthcare patients, therefore there is the need to comply with the new GDPR norm. To address this issue, we have built EPIC Cloud [2] (Enhanced Privacy and Compliance Cloud): an ISO/IEC 27001 27017 27018 certified region of INFN Cloud [3] adopting technical and organizational security measures that make it fit for processing personal data. Among the rights of individuals protected by GDPR, there are the rights *To Be Informed*, *To Erasure* and *To Restrict Processing* of personal data. GDPR introduces a specific terminology, whose main concepts are:

- Personal Data: any information relating to an identified or identifiable natural person (Data Subject) directly or indirectly;
- Data Controller: the entity which, alone or jointly with others, determines the purposes and means of the processing of personal data;
- Data Processing: any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction;
- Data Processor: the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data;
- Informed Consent: GDPR generally prohibits processing of personal data, unless it is expressly allowed by law, or the data subject has consented to the processing. The consent must be freely given, specific, informed and unambiguous. The data subject must also be informed about his or her right to withdraw consent anytime. The withdrawal must be as easy as giving consent. In the remainder of this paper we'll use the term *informed consent* to indicate the electronic form used to inform the data subject about his rights and to register his choices about allowing or forbidding a particular processing on his data.

Data controllers and data processors are required to provide means to allow data subjects to have full control on their data: each EU citizen should ideally have a dashboard with the complete list of his datasets associated with the list of controllers and processors who have access to them, the information about the type of access (read, modify, share, delete, archive, ecc) and buttons allowing to enable or disable the consent to perform a particular kind of processing on a specific dataset, from a specific processor/controller, for a defined purpose. This requirement imposes to every organization dealing with personal data to expand its computing infrastructures with functionalities enabling patients who provide personal data to exercise their rights. From the point of view of

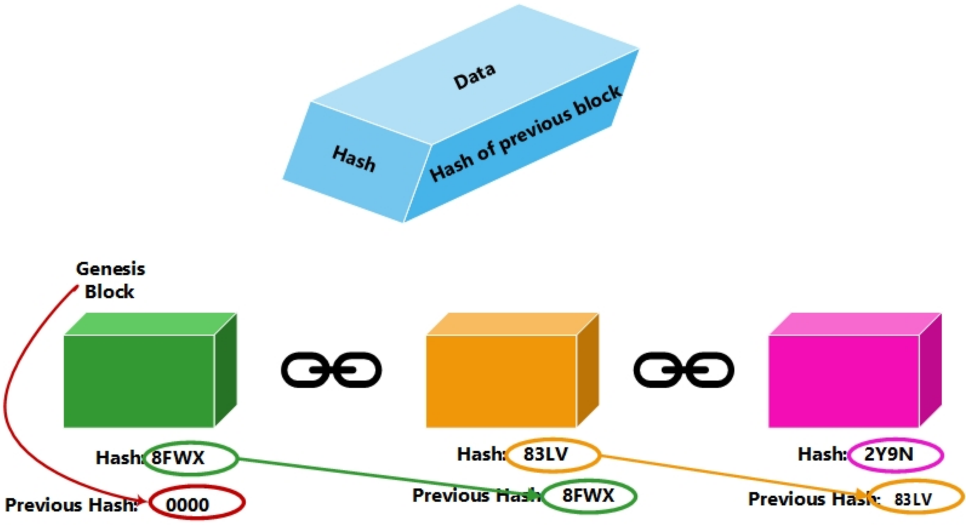


Figure 1: blockchain structure

organizational processes, a 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,
- How to withdraw the consent.

From a technical point of view, the (partial) solution is to use a Consent Management System (CMS) to correctly manage the informed consent workflow and lifecycle. A CMS enables data subjects to establish control over their data, granting access permission, auditing the use of their data, withdrawing permission and deleting their data whenever they want. However, even state-of-the-art CMS still suffer from a 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 foresees a third-party independent audit performed on a yearly basis. This is not always sufficient, because the audit guarantees the compliance at a specific point in time (once a year), and doesn’t put the power on the individual (data subject). Our proposal is to exploit blockchain technologies to enhance transparency and trustworthiness to current CMS solutions. blockchain is a set of solutions based on 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) and the chain is maintained by several actors exploiting a combination of:

- cryptographic techniques,
- consensus algorithms,

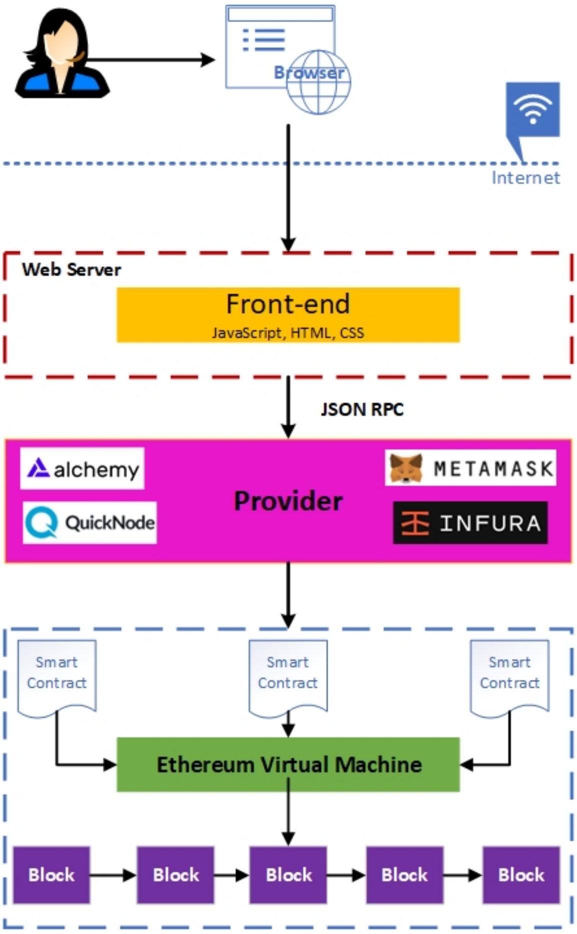


Figure 2: DApp structure

- peer-to-peer communications,
- game theory.

When a new block is added to the chain, it has to be verified by the majority of the nodes through a consensus algorithm. Each node has a full copy of the whole chain and is able to check the WS at each given time in the past. This technology is therefore not only secure, but gives us the possibility to do without a Trusted Third Party (TTP), usually a private company as a cloud provider, sharing the knowledge and the power about the WS between a potentially large number of nodes. Modern blockchain technologies include the possibility to run programs on the chain through Decentralized Applications (DApp) [4], Figure 2.

A DApp is an application built on the blockchain decentralized network that combines a smart contract and a frontend user interface. It is:

- Decentralized – when DApps operate on an open and public blockchain, no one person or group has control;
- 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 so that if the smart contract has a bug, it won't hamper the normal functioning of the blockchain network.

In this paper we'll discuss how blockchain technologies can be exploited on INFN Cloud and we'll present our roadmap for the deployment of a set of services for developers and end users.

2. Background and related work

Currently in literature there are attempts to implement blockchain-based consent management systems driven by the same motivation as our efforts, so in this section few of related works will be briefly presented. Keeping in mind that one of the main shortcomings of blockchain-based CMS solutions is lack of implementation details and performance analysis, we will focus on publications that don't have these problems.

Consentio [5] is proposed as a domain agnostic, scalable consent management system, based on a permissioned blockchain network. They attempt to make their system highly performant, through enabling higher transaction throughput and low latency. Their main approach is to separate the whole issue of personal data consent management into two planes – data layer and consent layer. Consentio deals only with the consent layer, in which it maps consent operations into blockchain transactions. After a data subject gives his consent for a data processor to access their data, the data processor is provided with a key to access the data stored in an off-chain data store, and record of this transaction is persisted on-chain. Their system is developed based on Hyperledger Fabric, or more specifically Fast Fabric, which is chosen for its improved transaction throughput. Regarding data management, they assume trust in third party data stores, due to potential legal consequences of unauthorized data releases. Another downside of Consentio is that it doesn't offer the option to data processors to directly request access from data subjects.

Another interesting solution, which isn't specific to a single sector, is a smart contract-based dynamic consent management system [6]. It was created according to design requirements specified in GDPR, since it aims to enable lawful personal data usage under said legislation. The authors propose a multi-layer, user-centric solution, which provides a systematic way for individuals to enter different types of consent agreements and manage the usage of their consented personal data. The first layer of the system is the personal data layer which provides a DApp and services for personnel data management. Then comes a middleware layer for enabling dynamic consent management by using smart contracts on top of the blockchain. At the bottom is a distributed ledger technology and secure storage layer utilizing Quorum blockchain to run smart contracts which enforce consent agreement requirements, and IPFS to provide peer-to-peer secure data storage. The performance, security and privacy of the system are demonstrated in the paper. However, the system is quite complex, thus making key management challenging. Also, due to the immutability of blockchain, it faces certain limitations with regard to Article 17 of GDPR which foresees the deletion and forgetting of user's data if they withdraw consent.

Ameyed at al. [7] propose a solution for consent management based on multiple blockchains. Similarly to other described systems, it is domain-agnostic, user-friendly and scalable. However, the key difference they envision is the multi-chain approach. Their system offers an application layer encapsulating a GUI which offers a way for data owners to set the state of consent for their personal data, and an API for communication with the blockchains. The database layer stores consent specifications and personal data separately. Finally, the multi-blockchain layer contains a separate chain for each respective data owner. The multi-chain approach was chosen due to the need to manage consents on many distinct sets of data from distinct data owners. The existence of personal chains for each respective user enables the control of each data owner's consent totally independently from others. Furthermore, single chain per user makes it possible for small servers to handle all transactions, since each user's data is stored in a separate chain. This multiple-chain blockchain was implemented by the group developing the system, and for proof-of-concept purposes they used a relatively low complexity mining proof, but this can be made more stringent in the future. However, similarly to Consentio, due to the nature of the personal data store, this system doesn't prevent illegal access to personal data, but it does offer a temper-proof mechanism for accessing the data.

BAQALC [8], or Blockchain Applied FASTQ and FASTA Lossless Compression, is a lossless compression algorithm developed with the aim of facilitating efficient transmission and storage of large amounts of DNA sequence data with the average compression ratio of roughly 12. Gursoy et al. [9] developed a novel private blockchain network for on-chain storing of personal genomic variants and reference-aligned next generation sequencing reads, also tools for rapid access and analysis of on-chain data, as well as tools for variant calling. This is a good example to illustrate the full impact the blockchain technologies could have on genomic data management. Integration of a CMS with a system for on-chain genomic data storage and analysis would migrate the entire life cycle of sensitive data to chain, and virtually eliminate all trust-based relationships.

3. Main blockchain solutions

The landscape of blockchain solutions is wide and include very diverse options, combining different deployment models, cryptographic algorithms, consensus mechanisms and game theory strategies. In the context of INFN Cloud we apply software adoption policies requiring that each software adopted is open-source, preferably free (non free options are considered only in case of lack of free solutions matching a particular set of requirements), with a wide support community, and relevant portfolio of users. Based on these policies, in a preliminary survey and evaluation phase, we restricted our choices to two solutions: Hyperledger Fabric [10] and Ethereum [11].

3.1 Hyperledger

Hyperledger is an umbrella project for the development of a number of software frameworks, tools, and libraries aimed at creating an open, global ecosystem for enterprise-grade blockchain deployments. It is backed by the Hyperledger Foundation, part of the Linux Foundation and the main focus is on vendor neutrality, opennes, and community-driven development. Among the most mature projects there is Hyperledger Fabric, a modular and customizable blockchain solution structured around the following concepts:

- Assets: enable the exchange of anything over the network;

- Chaincode: the businss logic. It is software defining an asset, and the transaction instructions for modifying the asset;
- Distributed Ledger: the sequenced, tamper-resistant, immutable record of all state transitions in the fabric. State transitions are a result of chaincode invocations submitted by participating parties. Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes. This shared ledger encodes the entire transaction history and includes SQL-like query capability for efficient auditing and dispute resolution;
- Privacy: channels and private data collections enable private and confidential multi-lateral transactions. A ledger exists in the scope of a channel. It can be shared across the entire network (assuming every participant is operating on one common channel), or it can be privatized to include only a specific set of participants;
- Security & Membership Services: only approved members can participate in the blockchain network.All transactions can be detected and traced by authorized regulators and auditors. In Hyperledger Fabric there is a transactional network where all participants have known identities. Public Key Infrastructure is used to generate cryptographic certificates which are tied to organizations, network components, and end users or client applications. Data access control can be manipulated and governed on the broader network and on channel levels;
- Consensus: the full-circle verification of the correctness of a set of transactions comprising a block. Different algorithms can be plugged-in to manage the block validation process.

3.2 Ethereum

The intent of Ethereum is to allow developers to create arbitrary consensus-based applications that have scalability, standardization, feature-completeness, ease of development and interoperability at the same time. Ethereum does this by building a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. [12]. Ethereum is a public, permissionless blockchain, meaning that anyone can act on the underlying blockchain and participate in the consensus mechanism. The main logical building blocks of Ethereum are:

- Accounts: the objects which alltogether make up the "state". Each account has a 20-byte address and state transitions are direct transfers of value and information between accounts. An Ethereum account contains four elements:
  - the nonce: a counter to ensure that each transaction is processed only once,
  - the current ether balance,
  - the contract code (if the account refers to a smart contract),
  - the storage.

  Ether is the main internal crypto-currency of Ethereum, and is used to pay transaction fees and acts as a "fuel" for executing transactions.

- Transactions: signed data packages that store a message. They represent the transition from one state to another.
- Code execution: the code in Ethereum contracts is written in a low-level bytecode language, referred to as "Ethereum virtual machine code" (EVM code). The operations have access to three types of space in which to store data:
  - stack: a last-in-first-out container to which 32-byte values can be pushed and popped (volatile);
  - memory, an infinitely expandable byte array (volatile);
  - Contract's long-term storage, a key/value store where 32-bytes keys and values (persistent).
- blockchain: a chain of blocks, connected through an hash function, containing the transaction list and the most recent state.

3.3 Comparison between Hyperledger Fabric and Ethereum

Both solution are Free and Open Source Software (FOSS) backed by an international foundation. A comparison and contrast between Hyperledger Fabric and Ethereum is given in Table 1 where Access, Deployment and Consensus algorithm are provided. In Table 2, instead, deveopment language and distributed app (DApp) is highlighted.

Block Chain	Access	Deployment	Consensus algorithm
Ethereum	permissionless	public	Proof of Stake (PoS)
Hyperledger Fabric	permissioned	private	Practical Bizantine Fault Tolerance (PBFT)

Table 1: Comparison between Ethereum and Hyperledger Fabric.

Block Chain	Development language	Distributed App (DApp)
Ethereum	Solidity	Smart Contract
Hyperledger Fabric	golang or JavaScript + Hyperledger, Composer	Chaincode

Table 2: Comparison between Ethereum and Hyperledger Fabric in terms of language and DApp.

Being a public, permissionless blockchain, Etherum offers high accessibility and trustworthiness, the most effective decentralization and full transparency and censorship resistance, while lacks of features like privacy preservation because all transactions are potentially publicly available and suffers of scalability problems (about 29 Transactions per Second - TPS [13]). On the other side, Hyperledger Fabric is private and permissioned, meaning that it is deployed in a private environment and it is fully controlled by the organization (or consortium) running it. This makes Hyperledger Fabric the best choice in case of privacy requirements (which is our case) and the most efficient solution (3k TPS with solutions in literature that can scale toward 20k TPS [14]). From the point of view of DApps and their related development language expressiveness, Solidity, the language developed by Ethereum, is the most mature, complete, and supported solution. Solidity is an object-oriented, statically-typed and high-level language, with syntax influenced by JavaScript and C++. The drawback with Ethereum DApps is that they need fuel (Ethers) to be run on the blockchain (the Ethereum Mainnet). The only way to get Ethers is to buy them with conventional money, making this choice unfeasible in our environment.

3.4 *In medio stat virtus*: Hyperledger BESU

The best solution in our environment should provide a mix of features like transparency, trustworthiness and expressive power of development language (Ethereum characteristics) with privacy preservation, security and efficient DApp execution (Hyperledger Fabric characteristics). The Hyperledger BESU project [15] is an Ethereum blockchain deployable in private permissioned environments, with an extractable EVM implementation. It includes several consensus algorithms like Proof of Stake, Proof of Work, and Proof of Authority. Its comprehensive permissioning schemes are designed specifically for use in a consortium environment. There is an inevitable trade-off between censorship resistance and TPS performance, therefore deploying an Ethereum blockchain in a private setup, with a limited number of network nodes participating in the consensus algorithm, could be more prone to 51% attacks, anyway in our opinion this risk is acceptable in our context and can be mitigated through the deployment of the network nodes in different regions of INFN Cloud, each one controlled by a different administrators group. Hyperledger BESU combines the Solidity-based DApps with the efficiency of an high speed blockchain network, due to the fact that the deployment is private and the consensus algorithm involves a limited number of clients. Users have to authenticate to the network in order to access it and permissions on datasets can be defined in order to guarantee privacy preservation.

4. On-demand deployment of Blockchain Development Environments

The role of blockchain technologies on a cloud infrastructure can be multifold: they can be deployed as a managed infrastructural service, on which users can store immutable events, or they can be deployed on-demand. Depending on the use case, in fact, the blockchain solution can be a "vanilla" deployment around which a user can build its own application, or can be a complete SaaS solution targeted to a particular application.

In such respect, as a first attempt to have an on-demand blockchain Development Environment (BCDEaaS) see Figure 3 for details, Scaffold-eth [16] has been chosen to be deployed as a cloud service via INFN Cloud capabilities [3].

Scaffold-eth is a toolkit containing all necessary state-of-the-art tools for rapid full-stack development and experimentation with DApps on an Ethereum-based blockchain platform that provides an easy way to start familiarizing, or freshening up on concepts related to the Solidity language and the Ethereum environment.

Key components of the scaffold-eth dev stack are here described:

- Hardhat [17] allows easy deployment, testing and debugging of Solidity code. It comes with the Hardhat runner and the Hardhat network built-in. Hardhat runner is a task runner designed

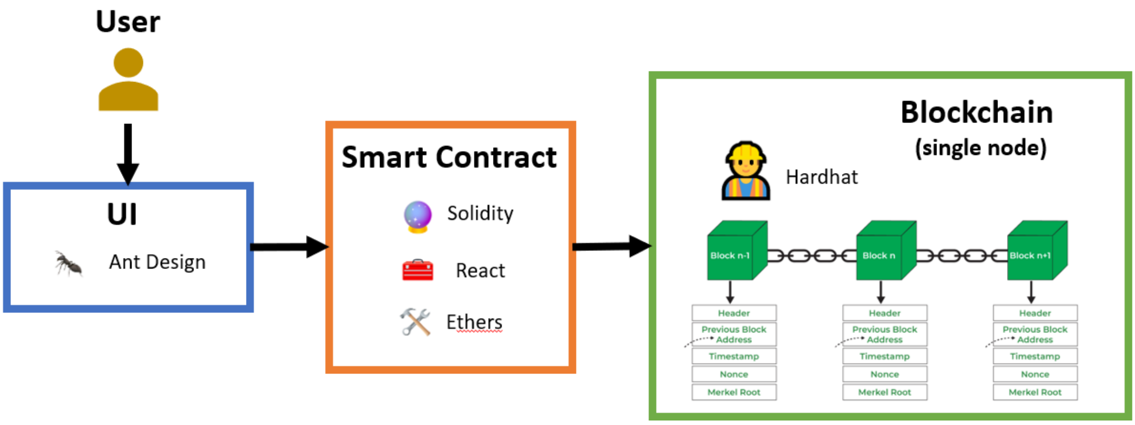


Figure 3: Scaffold-eth DApp development environment components

with concepts of tasks and plugins in focus. Each hardhat command initiated from the CLI is running some task, which can call other tasks and create workflows. Users can create their own custom tasks and plugins, which can override existing tasks and configure workflows. The Hardhat network is a local Ethereum network node designed for development, allowing contract deployment, testing, and debugging within the local machine.

- React [18] is a JavaScript library used for the frontend and User Interface (UI), which is very common in the Ethereum ecosystem.

- Ethers.js [19] is another widespread JavaScript library offering the possibility to communicate with the Ethereum blockchains from the front-end. In this toolset Ethers is quite abstracted from developers, and common react hooks, helpers, and components that already have Ethers backed are provided. However, use of Ethers can be extended beyond what is provided by default.

- Ant design [20] is a library providing many UI components that can be used for the final users.

4.1 BCDEaaS: Scaffold-eth as a cloud-enabled application

The Scaffold-eth application have been already containerized but it was almost not mature to be deployed in a Cloud environment. Starting from that, an important activity has been carried out to improve the virtualization part of the application in order to deploy and run it on the Cloud.

As a first step, a customized Docker image of scaffold-eth [21] has been created. The image has the scope to make the application fully portable and reusable in different environments. Moreover, the image is linked with a specific Github repo [22] where the contracts are stored. Every time a new contract is created or modified in the linked Github repo, it will be automatically updated in the scaffold-eth application and made available for the developer.

As an added value, a docker-compose file has been created. It contains, one hand, the needed instruction to deploy scaffold-eth and, on the other hand, the instructions to deploy an Nginx server. The need to have an Nginx server running is twofold: it properly redirects connection form and to the different Scaffold services and it also provide TLS termination.

To deploy the full Scaffold-eth stack over a Cloud environment, we made use of the facilities made available by the INFN Cloud environment. For the sake of simplicity, we perform our tests in a separated Cloud environment made available for the present purpose. In particular, we took advantage from the docker-compose run deployment [23], already available in the INFN Cloud, to deploy our modified version of the Scaffold-eth application.

Moreover, we properly customize the TOSCA template [24] used within INFN Cloud in order to have the Scaffold-eth deployment, and the related button, available via the INFN Cloud orchestrator-dashboard. Figure 4 shows a graphical overview of the INFN Cloud orcehstrator-dashboard used for tests where the Scaffold-eth deployment can be selected.

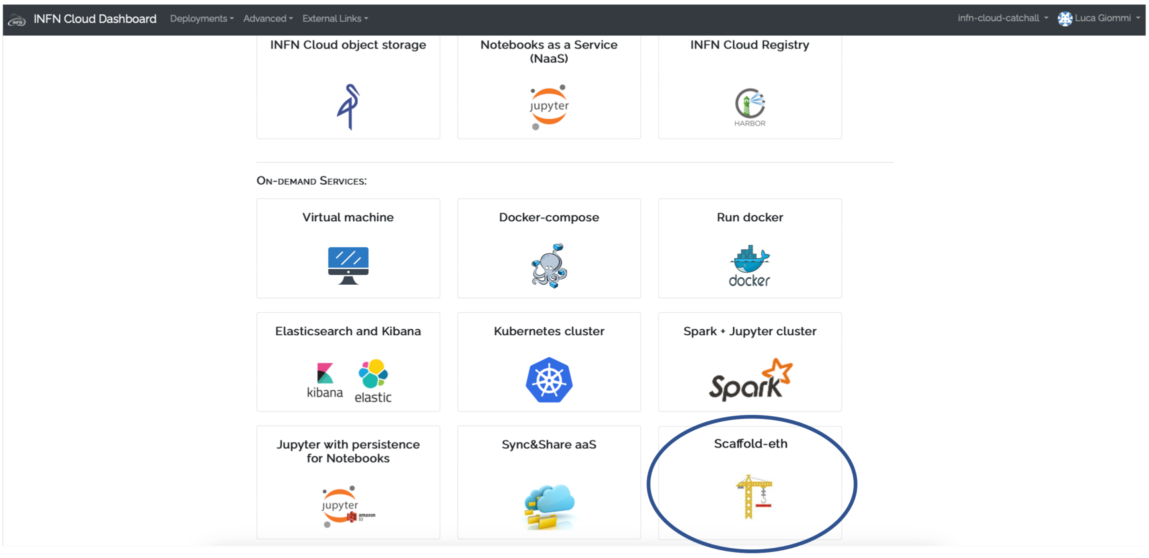
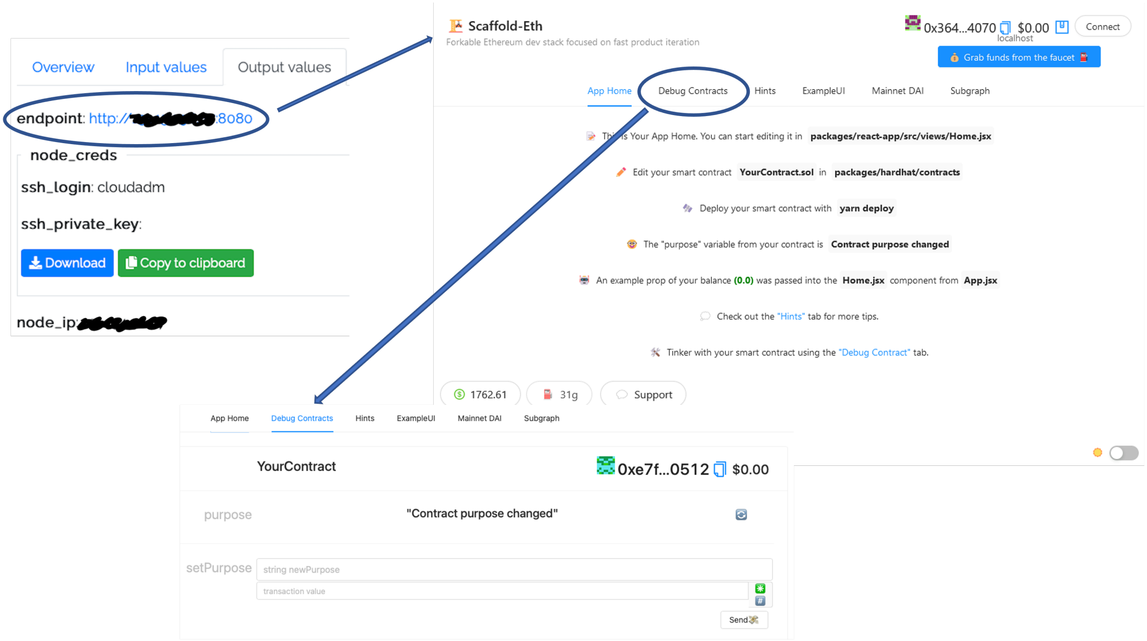


Figure 4: The Scaffold-eth customized deployment available within the INFN Cloud orchestrator-dashboard environment used for test.

Once the Scaffold-eth deployment is selected, the user is redirected to a proper configuration page where he/she is requested to provide some information aimed at setting-up (i) the resource used to host the scaffold-eth application (e.g., the number of CPUs and RAM size) (ii) the URL of the docker-compose file to be used, (iii) variable keys and related values to be used by the application during the deployment process. This information is used to populate the related TOSCA template, which is then submitted to the PaaS Orchestrator as a service deployment request used to select the service provider most suitable to deploy the application. As an outcome, the system will provide the http(s) endpoint to reach the deployed service (see Figure 5 for more details). As from the Figure, the user can access the scaffold service over http(s) protocol and start working with the Blockchain Development Environment deployed on-demand over the INFN Cloud.

5. Roadmap for blockchain exploitation over INFN Cloud

In order to provide INFN Cloud users a complete set of solutions, we have defined a roadmap, starting from the provision of less complicated services, then composing those services toward the offer of a complete set of solutions. Four steps have been identified, listed as increasing complexity:



endpoint: http://[redacted]:8080

node_creds

ssh_login: cloudadm

ssh_private_key:

node_ip: [redacted]

Scaffold-Eth
Forkable Ethereum dev stack focused on fast product iteration

0x364...4070 $0.00
localhost

This is Your App Home. You can start editing it in packages/react-app/src/views/Home.jsx

Edit your smart contract YourContract.sol in packages/hardhat/contracts

Deploy your smart contract with yarn deploy

The "purpose" variable from your contract is Contract purpose changed

An example prop of your balance (0.0) was passed into the Home.jsx component from App.jsx

Check out the "Hints" tab for more tips.

Tinker with your smart contract using the "Debug Contract" tab.

1762.61 31g

YourContract 0xe7f...0512 $0.00

purpose "Contract purpose changed"

setPurpose string newPurpose

transaction value

Figure 5: Example of output in the INFN Cloud dashboard used for test. Using the endpoint URL, the user is redirected to the Scaffold-eth application deployed on-demand in the INFN Cloud testing environment.

1. *BCDEaaS: on-demand deployment of Blockchain Development Environments*. A developer environment deployable as a service, explained in section 4.

2. *BCaaS: on-demand deployment of general purpose, multinode blockchain environments*. A multi-node blockchain deployable through the INFN Cloud dashboard and selecting the number of required nodes.

3. *BC-CMSaaS: on-demand deployment of blockchain-based Consent Management Systems built on top of the BCaaS functionality*. A Consent Management system deployable from the INFN Cloud dashboard. This is exploitable in life science research environments to manage the informed consent workflow without having to trust an external third party.

4. *BC-GIMSaaS: blockchain based Genomic Information Management System* built on top of the BCaaS functionality and exploiting the BC-CMS to manage patient consent. An efficient and trustable storage system for genomic data, integrated with the consent management system in order to ensure that researchers access only data shared by patients in the scope of a valid informed consent.

5.1 BCaaS: on-demand deployment of general purpose, multinode blockchain environments

The aim of this service is the on-demand provision of a multi-node blockchain that can subsequently be exploited by vertical applications needing an immutable distributed ledger run by a community, without any hierarchical authority. As explained in section 3.4, for this service we've choosen Hyperledger BESU as blockchain technology, because it merges the transparency and DApp expressive power of Ethereum, with the possibility of deploying the blockchain in a private, permissioned environment. The default deployment of Hyperledger BESU on INFN Cloud is depicted in Figure 6 and includes four validation nodes based on the QBFT [25] consensus protocol.

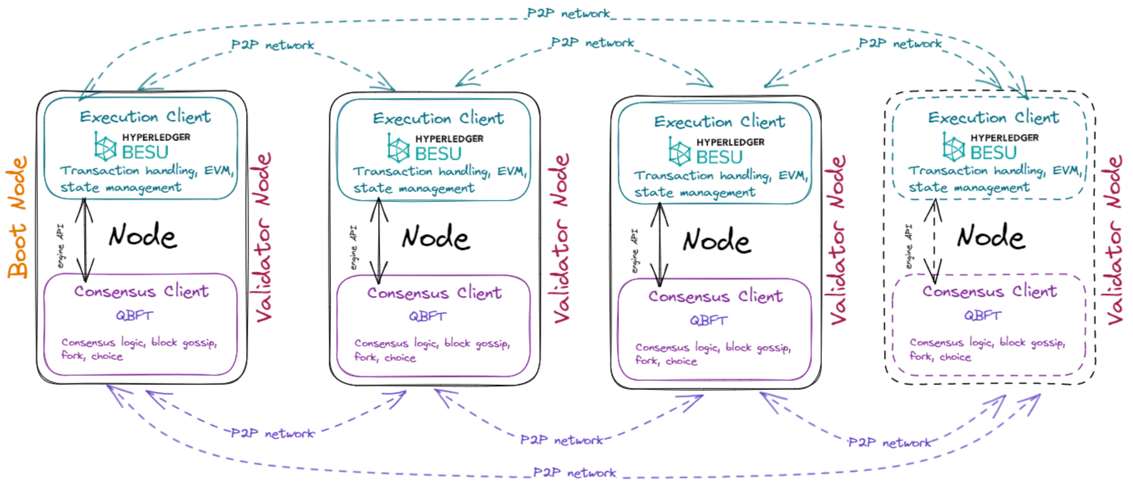


Figure 6: Architecture of the BCaaS network

QBFT consensus protocol is a Byzantine Fault-Tolerant Proof-of-Authority consensus algorithm designed for consortium use. QBFT is an evolution of the “Istanbul BFT Consensus” algorithm (IBFT) described in EIP-650 [26] that provides improvements in reliability and speed, and with as few as $\frac{2}{3}$ of validators functioning correctly at any given time it will not stall nor produce forks. In QBFT networks, approved accounts, known as validators, validate transactions and blocks. Validators take turns to create the next block. Before inserting the block onto the chain, a super-majority (greater than or equal to $\frac{2}{3}$) of validators must first sign the block. Existing validators propose and vote to add or remove validators. The default deployment system performs the following actions: deploys four validator nodes based on BESU containers, creates the genesis block for each node, starts the boot node which detects all other peers on the P2P network, starts the BESU client.

5.2 BC-CMSaaS: on-demand deployment of blockchain-based Consent Management Systems built on top of the BCaaS functionality

The BC-CMSaaS leverages the Hyperledger BESU blockchain to implement a Consent Management System (CMS) to support users in defining and auditing who can access their personal data. The blockchain is useful to add transparency, trustworhiness and TTP independence to the system. It is tamper-proof and auditable. The consent layer is decoupled by the data management layer. The actors in the consent management workflow are:

- Patients: play a data subject role. Give data to hospitals; sign the informed consent; revoke consent; audit the CMS.

- Hospitals: play a data controller role. Collect patients’ data; appoint INFN Cloud as data processor; analise Patients’ data exploiting the life science datalake.

- 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.
- Hospital ethic committees: play a whatchdog role. Assign roles on CMS and datalake; audit CMS.

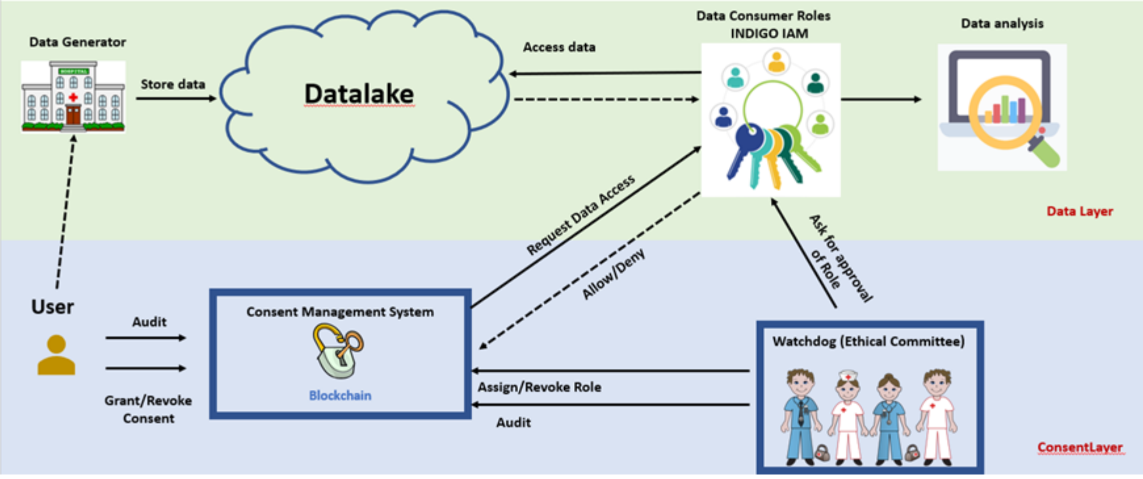


Figure 7: Consent management workflow

The consent management workflow is inspired by Consentio [5] and implements the model described in [7]: patients grant or withdraw consent to data consumers (doctors, researchers, insurance companies) playing specific roles. Watchdogs (Ethic Committee) assign and revoke data consumers' roles. Given their roles, data consumers request permission to access data stored in the datalake. The CMS determines whether such permission can be granted, based on patients consent. The consent is given and withdrawn through a smart contract inspired by [6] where we added the Individual Oriented Word State (IWS) in order to improve performance, as suggested by [5]. The smart contract data structures and variables are defined after the GConsent Onotology [27] [28], an OWL2-DL ontology for representing information associated with consent in the context of semantic web. The resulting Solidity code is publicly available on GitHub [29].

5.3 BC-GIMSaaS: blockchain based Genomic Information Management System

This service will leverage the BCaaS and BC-CMSaaS to make available a complete genomic information management system inspired by [8] and [9]. Genome data will be indexed in several blockchains in a hierarchical organization tailored on the genomic data structure. Genomes will be stored off-chain on IPFS [30] in order to guarantee good performances.

6. Conclusion and Future Work

INFN Cloud technologies are already the basis for several life science research projects at national and European level like Harmony Alliance [31], Health Big Data [32], and several national projects funded under the EU Recovery and Resilience Plan like the ICSC research center [33]. We believe that blockchain technologies will be of paramount importance to enhance transparency, auditability and trust at various architectural levels, therefore we defined an implementation and deployment roadmap based on four steps: 1. BCDEaaS, 2. BCaaS, 3. BC-CMSaaS and 4. BC-GIMSaaS. At the time of writing we have completed the first step, we have defined the architecture of the second and the third, and we are investigating the implementation options of the fourth.

References

[1] General Data Protection Regulation https://gdpr.eu/

[2] B. Martelli et al, SGSI project at CNAF, The European Physical Journal Conferences 214:08017, 23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018), published 17 settembre 2019, DOI: 10.1051/epjconf/201921408017

[3] INFN Cloud environment, https://www.cloud.infn.it/, last seen April 25th 2023.

[4] The architecture of a Web3.0 application https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application

[5] R. R. Agarwal, D. Kumar, L. Golab and S. Keshav, "Consentio: Managing Consent to Data Access using Permissioned Blockchains," 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2020, pp. 1-9, doi: 10.1109/ICBC48266.2020.9169432.

[6] Merlec MM, Lee YK, Hong S-P, In HP. A Smart Contract-Based Dynamic Consent Management System for Personal Data Usage under GDPR. Sensors. 2021; 21(23):7994. https://doi.org/10.3390/s21237994

[7] D. Ameyed, F. Jaafar, F. Charette-Migneault and M. Cheriet, "Blockchain Based Model for Consent Management and Data Transparency Assurance," 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), Hainan, China, 2021, pp. 1050-1059, doi: 10.1109/QRS-C55045.2021.00159.

[8] Lee S-J, Cho G-Y, Ikeno F, Lee T-R. BAQALC: Blockchain Applied Lossless Efficient Transmission of DNA Sequencing Data for Next Generation Medical Informatics. Applied Sciences. 2018; 8(9):1471. https://doi.org/10.3390/app8091471.

[9] Gürsoy G, Brannon CM, Ni E, Wagner S, Khanna A, Gerstein M. Storing and analyzing a genome on a blockchain. Genome Biol. 2022 Jun 29;23(1):134. doi: 10.1186/s13059-022-02699-7. PMID: 35765079; PMCID: PMC9241283.

[10] Hyperledger https://www.hyperledger.org/

[11] Ethereum https://ethereum.org/

[12] Vitalik Buterin Ethereum Whitepaper https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdfVitalikButerin2014

[13] Ethereum performance monitor https://ethtps.info/

[14] C. Gorenflo, S. Lee, L. Golab and S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second," 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea (South), 2019, pp. 455-463, doi: 10.1109/BLOC.2019.8751452.

[15] Hyperledger BESU https://www.hyperledger.org/use/besu

[16] Scaffold-eth github repo, https://github.com/scaffold-eth/scaffold-eth, last seen April 25th 2023.

[17] Hardhat development environment for Ethereum software, https://hardhat.org/docs, last seen April 25th 2023.

[18] React library for web and native user interfaces, https://react.dev/, last seen April 25th 2023.

[19] Ethers.js library for Ethereum Blockchain, https://docs.ethers.org/v5/, last seen April 25th 2023.

[20] Ant design applciation, https://ant.design/, last seen April 25th 2023.

[21] Customized Scaffold-eth image, https://hub.docker.com/r/anavel/scaffold-eth, last seen April 25th 2023.

[22] Customized Github repo for Scaffold-eth contracts, https://github.com/scaffold-eth/scaffold-eth, last seen April 25th 2023.

[23] INFN Cloud docker-compose run implemetnation guide, https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/sysadmin/compute/docker_compose.html, last seen April 25th 2023.

[24] INFN Cloud customized Tosca templates, https://baltig.infn.it/infn-cloud/tosca-templates, last seen April 25th 2023.

[25] QBFT Blockchain Consensus Algorithm Specification version 1 https://entethalliance.org/specs/qbft/

[26] EIP650 specification https://github.com/ethereum/EIPs/issues/650

[27] Pandit, H.J., Debruyne, C., O'Sullivan, D., Lewis, D. (2019). GConsent - A Consent Ontology Based on the GDPR. In: The Semantic Web. ESWC 2019. Lecture Notes in Computer Science(), vol 11503. Springer, Cham. DOI https://doi-org/10.1007/978-3-030-21348-0_18

[28] GConsent Ontology specification https://openscience.adaptcentre.ie/ontologies/GConsent/docs/ontology

[29] Customized smart contract code https://github.com/bmartell/sc-dcms

[30] Interplanetary File System https://ipfs.tech/

[31] HARMONY Alliance https://www.harmony-alliance.eu/

[32] Health Big Data project https://www.alleanzacontroilcancro.it/en/progetti/health-big-data/

[33] *Centro Nazionale di Ricerca in HPC Big Data e Quantum Computing* ICSC https://www.supercomputing-icsc.it/en/icsc-home/