



SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER SCIENCE



CUPChain

BLOCKCHAIN AND DISTRIBUTED LEDGER
TECHNOLOGIES

Professors:

Claudio Di Ciccio

Students:

Palma Alessio

Mancini Riccardo

Gargiulo Antonio Andrea

Gabrielli Davide

Buzo Elios

Contents

1	Preface	2
2	Background	3
2.1	The Blockchain	3
2.2	Application domain	4
3	Context	5
3.1	Aim of CUPChain	5
3.2	Why using the blockchain	6
4	Architecture	7
4.1	Front-end	7
4.2	Back-end & Database	8
4.3	Blockchain & Smart Contracts	9
4.4	Rollups	9
4.5	UML Diagrams	10
5	Implementation	14
5.1	Tools and Technologies	14
5.2	Smart Contract Development	14
5.3	Back-end Development	15
5.4	Front-end Development	16
6	Limitations	21
7	Conclusions	22
	References	23

1 Preface

This report presents our work for the “Blockchain and Distributed Ledger Technologies” course at Sapienza University of Rome, a.y. 2023/24. Our group name is CUPcakes and the project is named **CUPChain**. In its basic definition, it is a reservation system for medical examinations that will be managed by the national healthcare system. It aims to be accessible to everyone in Italy, trustable by its users and verifiable, providing no preferential treatment to anyone and making it difficult for malicious actors to skip the queue.

The team members and related main responsibilities are:

- **Davide Gabrielli:** Full-stack \ Database;
- **Antonio Andrea Gargiulo:** Back-end \ Smart contracts \ UML;
- **Riccardo Mancini:** Full-stack \ Smart contracts;
- **Alessio Palma:** Back-end \ Database \ Rollups;
- **Elios Buzo:** Back-end \ Smart contracts \ Testing.

but cooperation between all of us was of paramount importance.

In this report, we will start with an introduction to the main blockchain concepts, then move into presenting the context of our application and the rationale behind the use of the blockchain. After this first theoretical part, we will discuss the software architecture of our system making heavy use of UML diagrams, and then move to a deep dive into the implementation presenting also screenshots of the main pages of our DApp. To close the report, we are also going to discuss the known limitations to which our system might be exposed in a real-use environment and wrap up with a conclusive section.

2 Background

2.1 The Blockchain

The Blockchain is a protocol defining the rules for a decentralized and distributed ledger system that securely records transactions across a network of computers. The foundational principle of this protocol is consensus, ensuring agreement on the state of the ledger among all participants.

Bitcoin [1], introduced by the pseudonymous Satoshi Nakamoto in 2009, is widely recognized as the first successful implementation of a blockchain and it has achieved success especially as a decentralized digital currency. It uses the so-called transaction (or UTXO) model, meaning that the transaction is the first-class citizen of the system and there is no explicit concept of account balance. Bitcoin operates on the Proof of Work (PoW) mechanism: the miners compete to solve an intricate mathematical puzzle, validating transactions and adding blocks to the blockchain. The consensus is based on favoring the chain with the highest amount of cumulative work, hence energy-intensive PoW ensures the security and immutability of the ledger by making it highly computationally expensive and time-consuming to tamper with historical transactions.

Ethereum [2, 3], proposed by Vitalik Buterin in 2013 and launched in 2015, extended the capabilities of blockchain technologies beyond simple transactions. It uses the so-called account (or balance) model, putting stateful accounts at the center of the system. Ethereum introduced the concept of Contract Accounts, which are non-human-owned accounts having their behavior fully specified by a Smart Contract, which are deterministic pieces of software that can be invoked on the chain.

Initially using PoW, Ethereum underwent a transition to Proof of Stake (PoS), which was completed in 2022 with The Merge. PoS is based on attestations issued by a committee of validators, who put their cryptocurrencies at stake to obtain decision rights, and consensus in Ethereum's PoS is ensured with a two-stage process based on Gasper [4]. First, a new block can be proposed at each slot and the new head of the chain is voted through the LMD-GHOST algorithm [5] by a pool of pseudo-randomly selected validators. Then, after a block gets finalized through Casper-FFG [6] votes, it is very expensive (and subject to slashing) to try to rewrite history. PoS aims to improve energy efficiency and scalability, addressing various concerns associated with PoW such as reducing its electricity consumption and environmental impact. The emergence of smart contracts on Ethereum opened the door to Web3.0 and the development of Decentralized Applications (DApps): applications operating with a blockchain in the back-end, removing the need for centralized control.

In the historical context, blockchain's evolution from Bitcoin to Ethereum represents a response to the limitations of traditional centralized systems. Bitcoin sought to provide a peer-to-peer electronic cash system, ensuring financial transactions with-

out intermediaries. Ethereum, on the other hand, broadened the scope to enable programmable and decentralized applications, aiming for a more versatile blockchain platform.

2.2 Application domain

The Italian national healthcare system stands at a critical juncture, grappling with the challenges of decentralized administration and fragmented reservation systems. Presently, Italy’s healthcare infrastructure is characterized by regional autonomy, where each region assumes responsibility for its healthcare system within the framework of essential levels of care (LEA) outlined by the central government [7]. However, this decentralized approach has led to disparities in service delivery and operational inefficiencies [8], particularly evident in the realm of reservations. While these systems share similar functionalities, not all regions possess robust online reservation services.

The need for a unified yet decentralized reservation framework has never been more apparent and blockchain technology emerges as a transformative solution to this pressing challenge [9, 10]. By leveraging the inherent features of blockchain, such as transparency, immutability, and decentralization, Italy’s healthcare system can achieve seamless integration and equitable access to medical reservations.

Firstly, it ensures the integrity and security of patient data, safeguarding sensitive information from unauthorized access and tampering. Secondly, the decentralized but unified nature of blockchain will bring all the regional healthcare systems to the same level, eliminating the risk of some local systems being left behind. It would align with the ethos of Italy’s regional autonomy, because each region could maintain sovereignty over its availability of appointments, while fostering collaboration and interoperability across disparate systems participating in a shared network, promoting data exchange and standardization of reservation protocols. Thirdly, by facilitating peer-to-peer transactions, blockchain enables seamless coordination between healthcare providers and patients, reducing administrative overhead and streamlining the reservation process.

In conclusion, the application domain of the Italian national healthcare system reservations presents a unique opportunity for innovation and transformation. By embracing blockchain technology, Italy can surpass the limitations of fragmented reservation systems and establish a unified, decentralized framework that prioritizes accessibility, efficiency, verification and patient-centric care.

3 Context

3.1 Aim of CUPChain

The first aim of our system is to rebuild trust among the population towards the Italian public healthcare system, which is facing significant challenges. Statistics from 2021 reveal that only two-thirds of Italians trust hospitals and clinics (Figure 1), a lower score compared to other European and American countries. This distrust is exacerbated by a lack of publicly accessible data, making it difficult for the population to have confidence in the system.

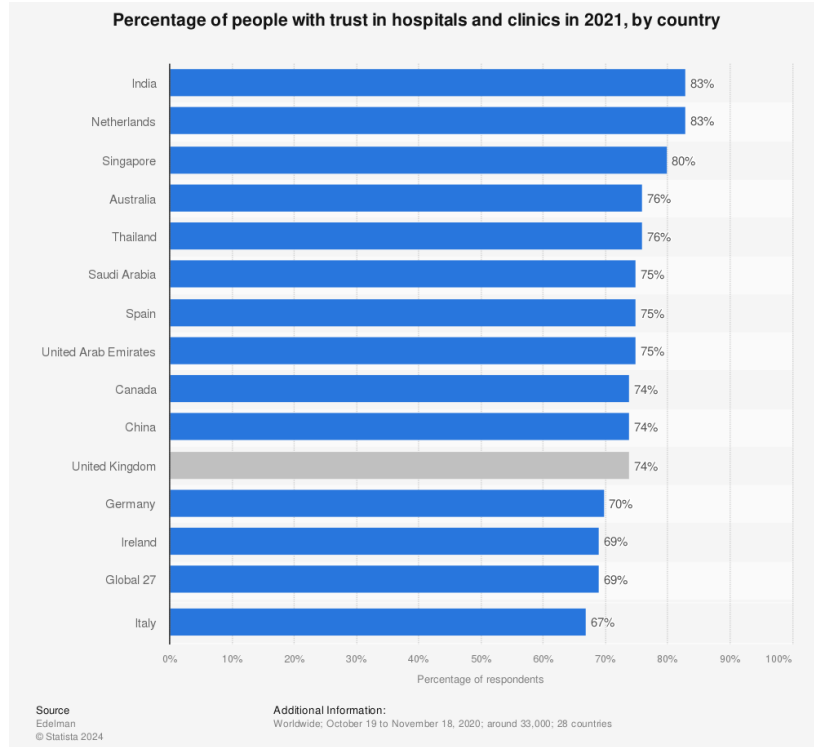


Figure 1: Trust levels towards hospitals and clinics by country 2021 [11]

This is mainly due to the long waiting times for accessing treatments [11], which can be surely alleviated by removing humans (and telephonic calls) from the reservation process. Another problem plaguing the Italian healthcare system is the fact that some doctors have been caught favoring acquaintances and their own private patients, altering the order of waiting lists [12, 13]. Addressing these specific problems and providing reliable information on the current state of the healthcare system could significantly improve public perception and alleviate concerns. That is why our objective is to substitute the C.U.P. (Centro Unico Prenotazioni, the name of regional systems for booking medical exams) with a blockchain-based solution that ensures the immutability of stored information and enables public verification, while still preserving the privacy of users.

3.2 Why using the blockchain

The decision to adopt blockchain technologies stems from the need to take care of all the problems we mentioned in the previous sections. With respect to the flow diagram presented in Figure 2, we note that:

- we *need to store states*;
- there are *multiple writers* like doctors, hospitals and patients;
- we *can't use any trusted third party*;
- *all writers are known*, since participants in our blockchain ecosystem have to be ideally identified via SPID to receive a private key, ensuring accountability;
- public *verifiability is required*, but storing all of our transactions on a public chain will be extremely costly for the collectivity.

To keep costs as low as possible and still ensure public verifiability, we propose a **private permissioned** blockchain to do all the transactions related to prescription emission and appointment booking, alongside a **public permissionless** one. This setup allows for the secure storage of sensitive data while enabling public verification through periodic rollups onto the public chain.

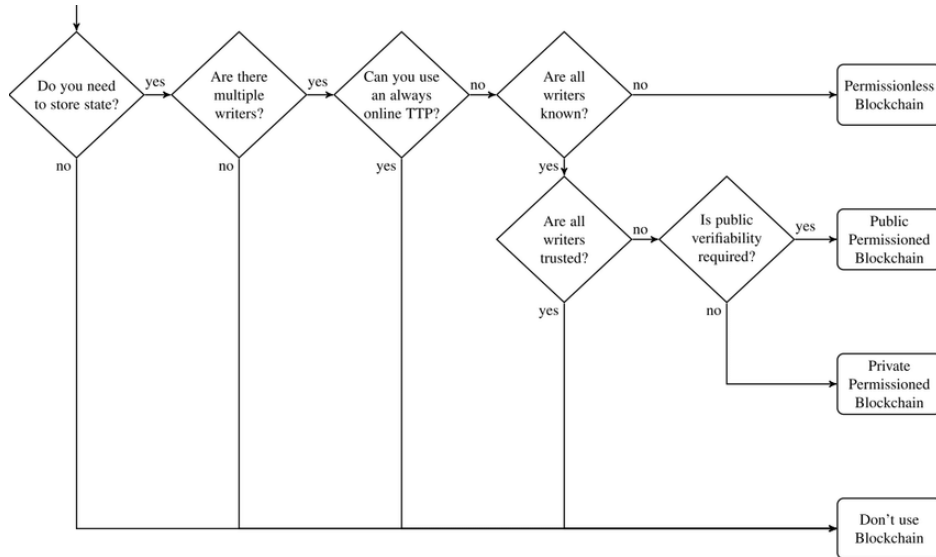


Figure 2: Why a blockchain is needed? Figure from [14]

4 Architecture

The CUPChain system architecture is divided into four main layers:

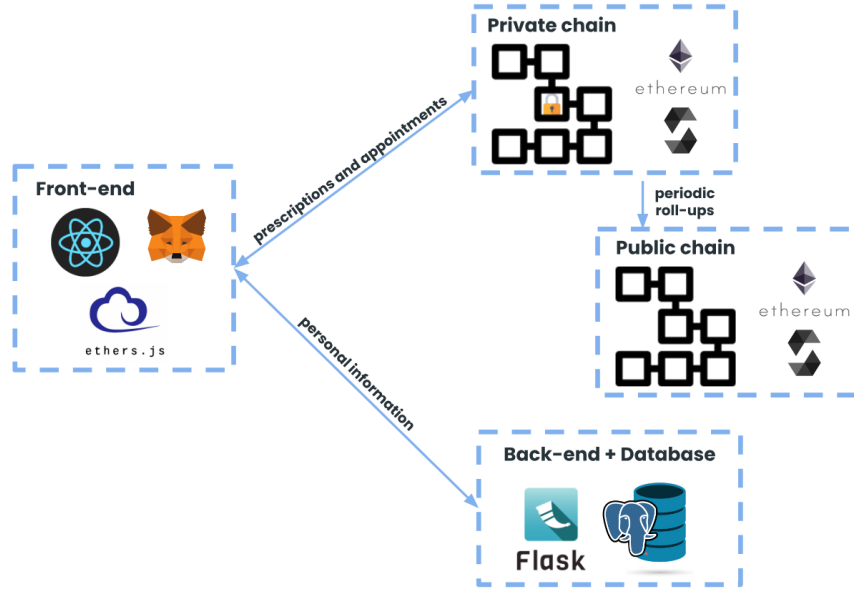


Figure 3: Overall architecture of the system

4.1 Front-end

Our front-end module uses React, prioritizing the organization of our user interface into distinct screens and reusable components, fostering modularity and code maintainability. Each screen represents a unique user view or interaction within our application, delineating specific functionalities such as appointment management or prescription creation.

Complementing our screen-based approach is the utilization of reusable components, which encapsulate common UI elements such as buttons or navigation bars. By abstracting these elements into reusable components, we ensure consistency across the application and mitigate redundancy in code.

To align with standards and best practices in public administration applications, we integrate design-react-kit from Developers Italia [15] into our front-end development workflow. This library provides a comprehensive set of pre-designed React components and styles tailored to meet the design guidelines and accessibility requirements for public sector applications. By adopting design-react-kit, we ensure that our application adheres to established design patterns and accessibility standards, fostering consistency and interoperability with other government services.

4.2 Back-end & Database

Our back-end module uses Flask, a Python-based web framework, to create a Representational State Transfer (REST) Application Programming Interface (API) following the Model View Controller (MVC) pattern. The REST APIs allow communication between the server and the client through HTTP requests. Flask is a lightweight framework that is easy to use and offers a wide range of features such as routing, debugging, and templating. The MVC pattern separates the application into three components: model, view, and controller. The model handles the data, the view handles the user interface, and the controller handles the logic that connects the two. This separation of concerns makes the code easier to maintain and modify. Overall, by using Flask and the MVC pattern, we have been able to create a robust and scalable back-end module for our application. The back-end also enables communication with the database to store data, most of which are strings, which would be costly to manage in a blockchain environment.

For the database, we have chosen PostgreSQL, a powerful open-source object-relational database system that is reliable, robust, and high-performing. The main reason for its utilization is due to reducing the cost of storing a huge amount of strings and other data that are only needed for visualization and interaction within the application. In such a way the back-end module can utilize the information stored in the database to complete the data displayed to the final user and give them complete information about their prescription, appointments, etc. In Figure 4 we can see the ER diagram of the database.

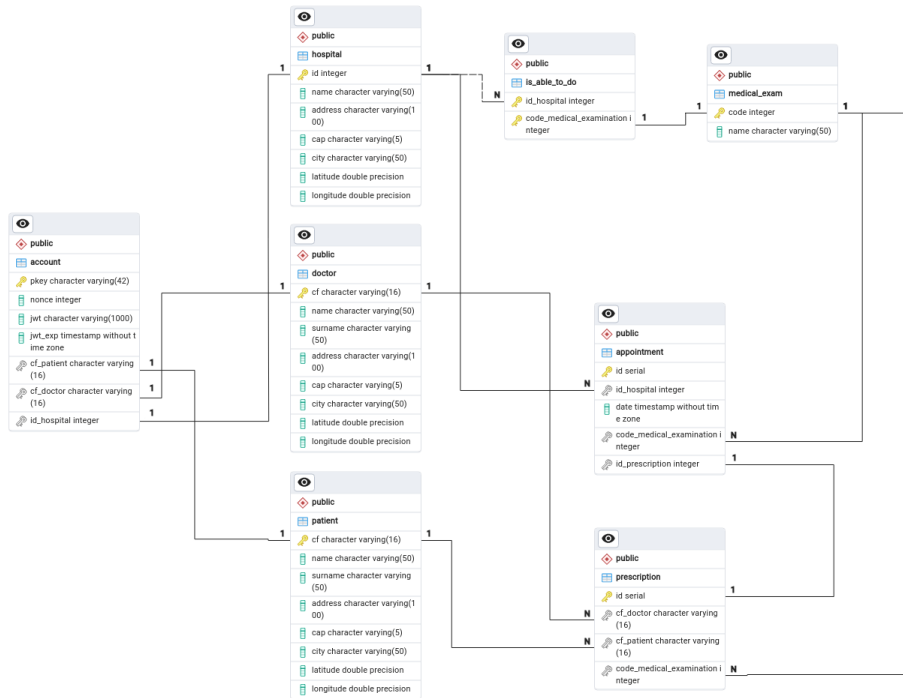


Figure 4: ER diagram of the database.

Notice that no hashing of rows is stored in the database, so one can question that those data are modifiable and no one could notice the changes, but this is not the case since all the hashes are passed to the corresponding transaction and publicly available on the blockchain. This provides full verifiability since a user can access the data and compute the hash on the fly, to perform a check. Moreover, this preserves the privacy of sensible information since only the user who is the owner of the data can access them.

4.3 Blockchain & Smart Contracts

The smart contracts handle the creation and trading of prescription and appointment tokens. Doctors will be assigned the role of “minter” for the prescriptions, while hospitals will be assigned the same role for the appointments. Through the front-end, they can interact with the PrescriptionTokens and AppointmentTokens contracts. Doctors can issue prescription tokens, while hospitals can create appointment tokens for available time slots. Patients can then use the contracts through the front-end to book and cancel appointments.

The PrescriptionTokens contract handles the booking: the owner of the prescription calls the related function, which transfers their token to the hospital; at the same time the function gives the hospital’s appointment token to the patient and saves the id of the prescription used by invoking the AppointmentTokens contract.

In the same way, the AppointmentTokens contract handles the cancellation of bookings: the patient (owner of an appointment token) calls the function that transfers his token back to the hospital. The function also retrieves the prescription used to book the appointment and employs the PrescriptionTokens contract to transfer back to the patient his prescription.

Only tokens with the same category (the type of medical exam they represent) can be exchanged. For example, a prescription token for an electrocardiogram can only be exchanged for an appointment of the same type. Moreover, it is important to note that patients are not allowed to transfer their tokens (prescriptions or appointments) directly to other users, discouraging bad usage of the application.

Tokens are also linked to the hash of their metadata saved in the back-end to ensure verifiability.

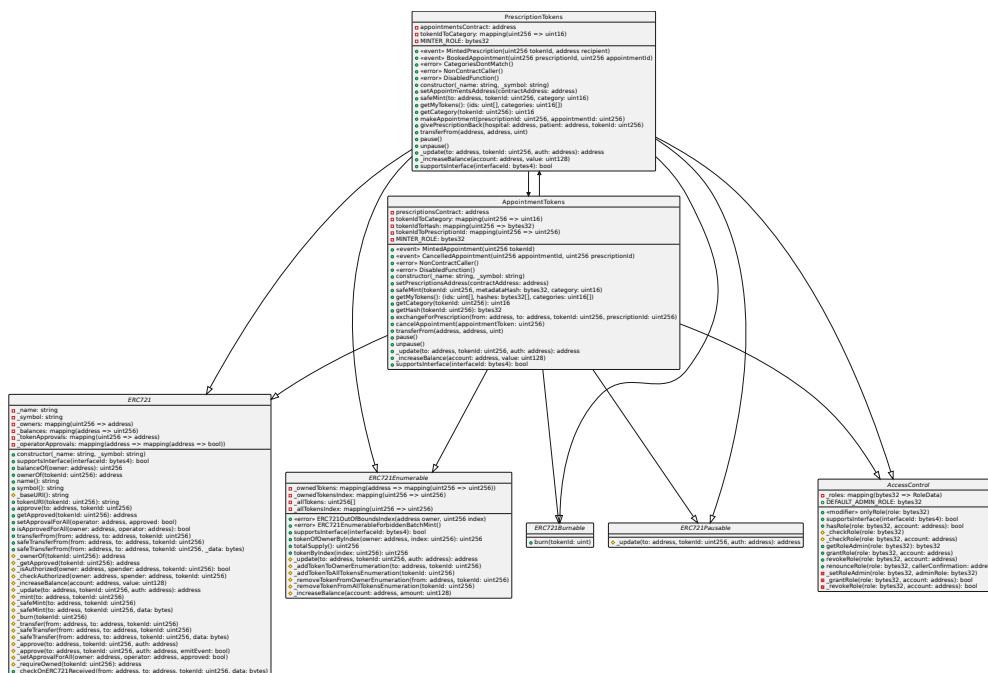
4.4 Rollups

Rollups are protocols that operate on layer 2 (L2) of the Ethereum’s network [16, 17]. They enable the scalability [18, 19] of Ethereum’s base layer (L1) by allowing more transactions to be performed per second while reducing the computational cost on the main chain. This is achieved by performing transactions off-chain.

Rollups bundle a large number of transactions from the L2 and publish them through a single transaction on the L1. Although the Ethereum chain is not responsible for executing each transaction, the data of transactions can be verified by a single transaction published on the Ethereum main chain, in the form of a proof. Thus trying to revert a single transaction in the L2 would require reverting the Ethereum chain, which is extremely costly (estimated around 4 million).

In our system rollups are crucial to ensure the verifiability of the system, since the private blockchain and the back-end system are owned and maintained by the government, institutions and medical system, therefore a centralization of power could occur. Having a proof that no one is altering data or performing malicious operations to the detriment of others is a good feature for our system and it could address the problem of trust in the actual healthcare sector.

4.5 UML Diagrams



In Figure 6 via a UML Use Case diagram are depicted the interactions between professional and non-professional users with the system. The diagram captures the primary functions of the system for each user.

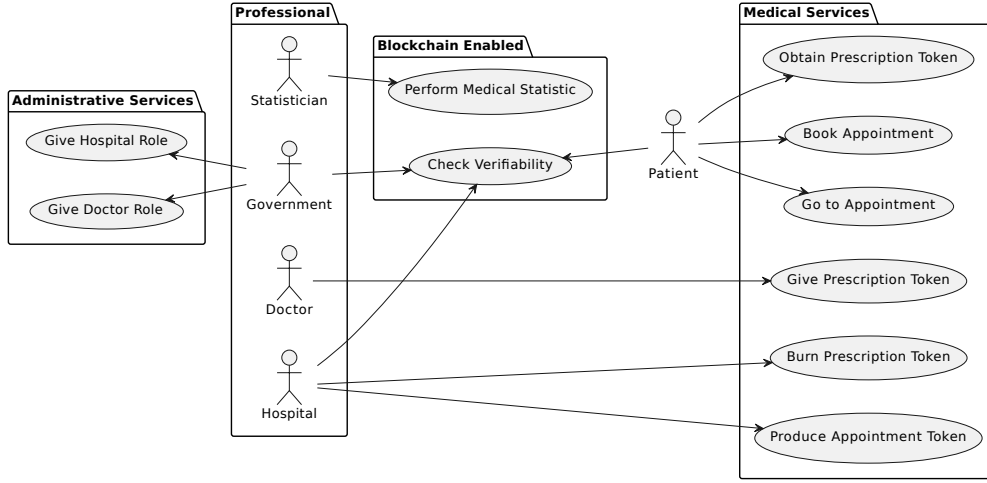


Figure 6: Use Case Diagram depicting the interactions of the system

In Figure 7 we can see the interactions of a user representing some manager of the Minister of Health (Government) granting a role to a doctor interacting with our smart contracts.

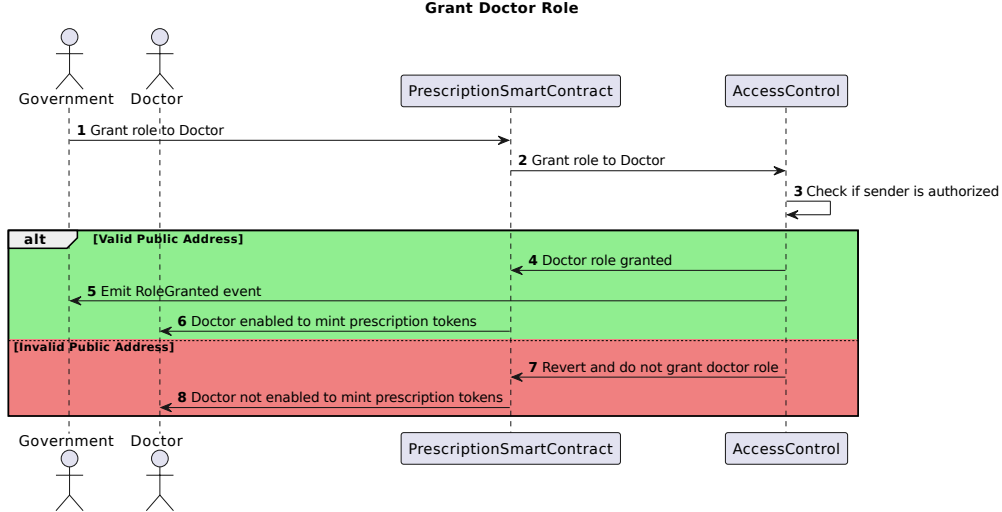


Figure 7: Sequence Diagram depicting the government granting role to a doctor.

In Figure 8 we can see the login using the Metamask wallet for our application.

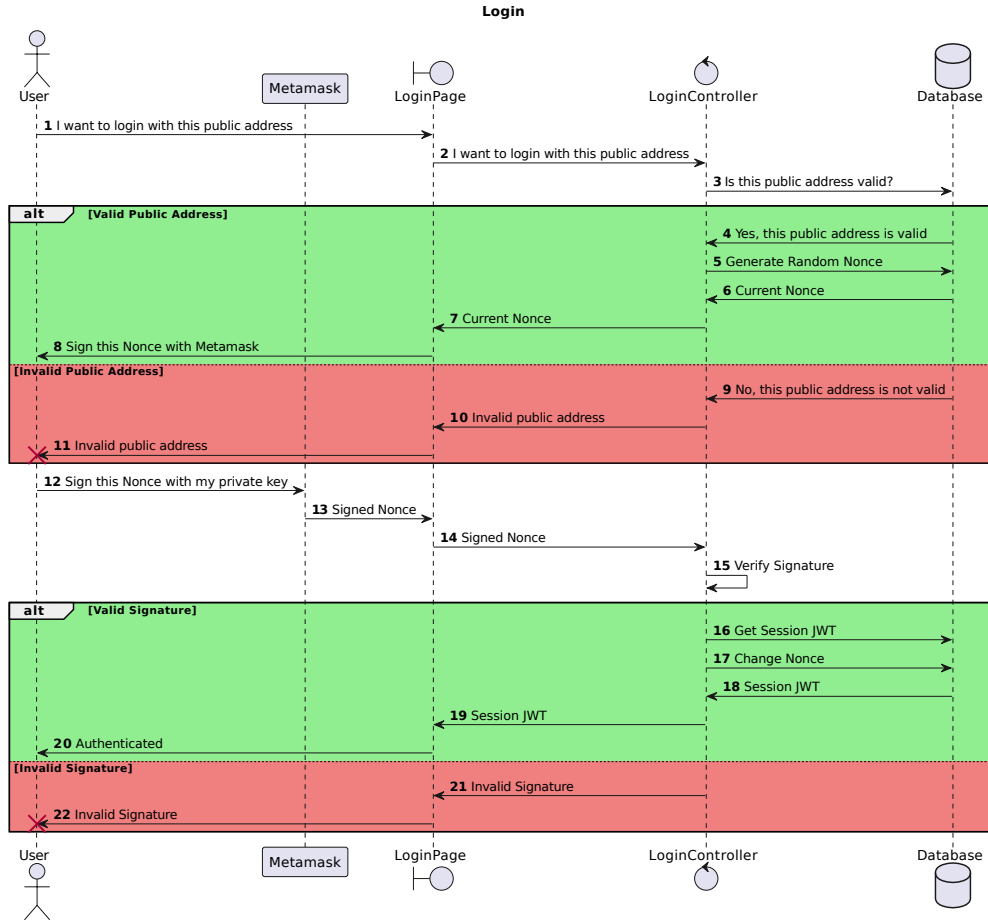


Figure 8: Sequence Diagram depicting a user login into the system.

In Figure 9 and in Figure 10 we can see the interaction between our back-end, database and smart contracts to improve the healthcare system, with a blockchain-based architecture.

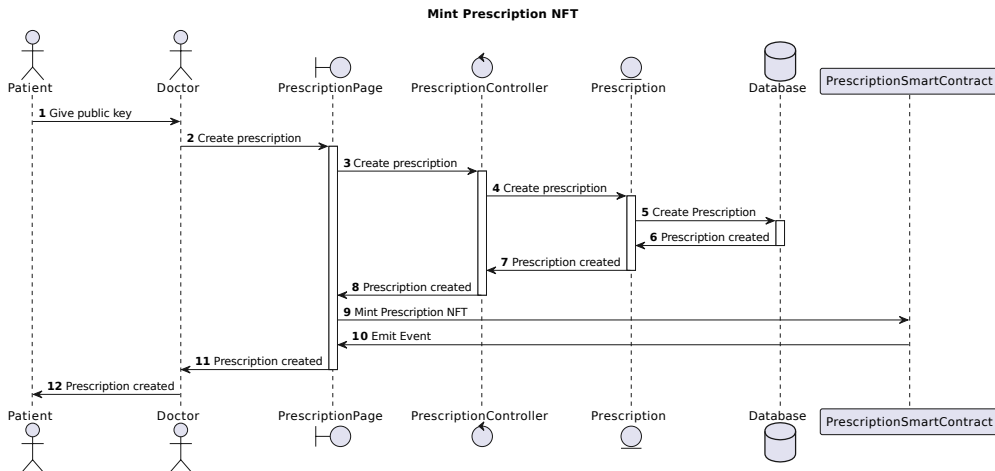


Figure 9: Sequence Diagram depicting a doctor minting a prescription for a patient.

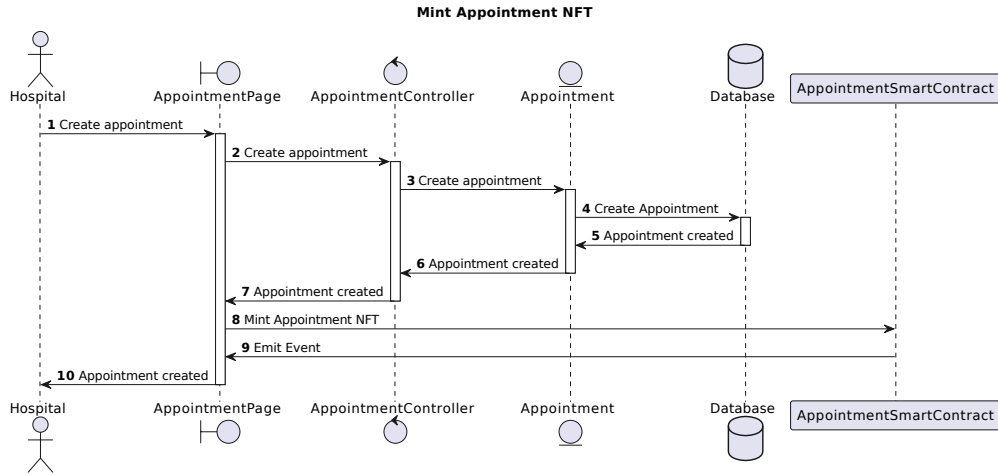


Figure 10: Sequence Diagram depicting an hospital minting an appointment.

In Figure 11 we can see a detailed interaction of a patient with our application highlighting the interchanges between smart contracts and showing how it is possible to reduce the costs of these operations by integrating communication with a database.

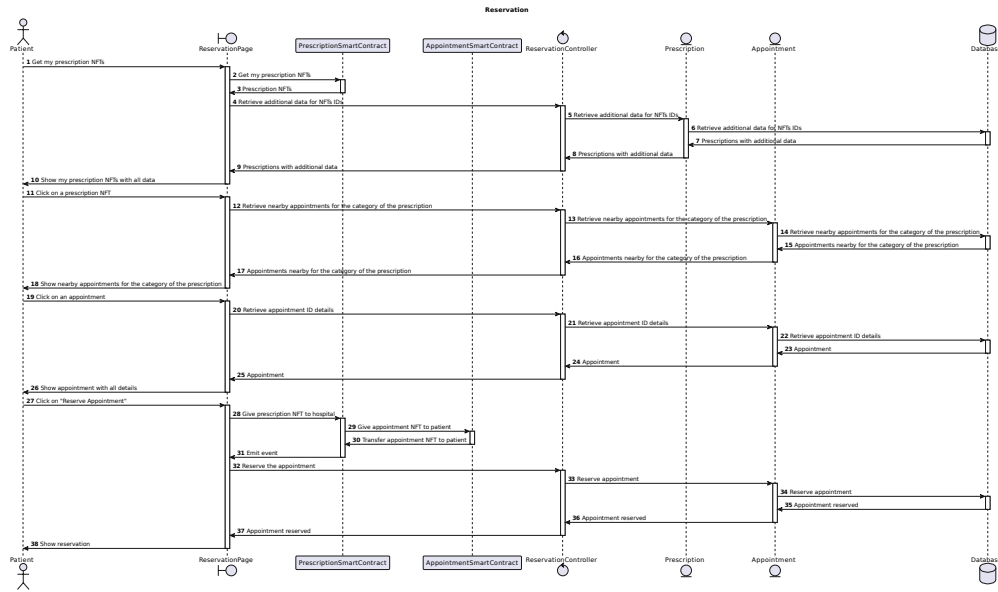


Figure 11: Sequence Diagram depicting a patient exchanging a prescription for an appointment.

5 Implementation

5.1 Tools and Technologies

Our technology stack was carefully curated to ensure optimal performance and seamless interaction within the blockchain ecosystem:

- **Solidity:** Solidity is the language used for creating smart contracts on the Ethereum blockchain. It allowed us to create and enforce rules for managing appointments in a decentralized and secure way.
- **OpenZeppelin:** To aid in the creation of Tokens compliant with the ERC721 standard [20], we used the library provided by OpenZeppelin [21].
- **React:** For the front-end development, we opted for React, a powerful JavaScript library renowned for its flexibility and scalability.
- **Flask:** For the back-end development, Flask provided the necessary tools for handling server-side operations and interacting between the database and the front-end.
- **PostgreSQL:** As our choice of the database management system, PostgreSQL offered reliable data storage capabilities, ensuring the secure management of appointment-related data, patient records, and hospital information.
- **ethers.js:** The integration with the Ethereum blockchain was facilitated through ethers.js, a JavaScript library for interacting with Ethereum smart contracts and conducting transactions. ethers.js provided the necessary utilities for securely managing wallets, contracts, and transactions.
- **Mocha and Chai:** The unit tests for ensuring the correct behavior of our smart contracts are written with the help of Mocha, a lightweight Node.js test framework, and Chai, a Test Driven Development assertion library for node.

5.2 Smart Contract Development

Our prescription and appointment tokens are Non-Fungible Tokens compliant with the ERC721 standard and implemented with the help of the smart contract library OpenZeppelin.

They inherit the Burnable, Pausable, Enumerable and AccessControl contracts, also provided by OpenZeppelin. In this way they allow owners to burn their tokens, contracts to be put on pause, to easily check which tokens are owned by which account, and to handle permissions with ease through the use of roles.

We prevented users from transferring tokens to each other by overriding the transfer functions. The only transfers that should occur are those between patients and hospitals, and they should be handled exclusively by contracts. The possibility to create new tokens (minting) is restricted to individuals with the MINTER role. This role is granted to doctors for prescriptions and to hospitals for scheduling appointments. The Ministry of Health, which is presumed to be the entity deploying the contracts, is responsible for assigning the MINTER role to these entities.

Prescription and appointment tokens are categorized based on the respective medical exam they represent, with the latter also including metadata hashes.

The PrescriptionTokens contract contains the function for booking the appointment, which, after checking that the categories of the prescription and the appointment match, transfers the caller's prescription token to the hospital that owns the appointment token, and then it calls the function residing on the AppointmentTokens contract that deals with transferring the appointment token from the hospital to the patient and saving the id of the prescription that was used.

The AppointmentTokens contract contains the function for canceling the appointments, which retrieves the id of the prescription used for the booking and gives it back to the patient through the PrescriptionTokens contract after transferring the appointment token back to the hospital.

The contracts include functions that allow the user to retrieve a list of owned tokens. Additionally, the contracts also enable users to access information about a particular token, such as its category. In the case of appointment tokens, users can retrieve the token's hash and the id of the prescription used to make the booking.

Every time a token is minted, or an appointment is booked or canceled, a corresponding event is emitted, in order to let the listeners on the front-end know that the operation has been completed successfully.

Since the self destruct functionality has been deprecated we need a way to disable, temporarily or permanently, the core functions of our contracts. The Pausable pattern, helped us to implement this behavior. By implementing two functions that set the state of the contract, PAUSE and UNPAUSE, we unlocked the use of two modifiers that specify which function can be used with respect to the state of the specific contract. In this way the owner of the blockchain can disable the minting and the exchanging of the tokens, which are critical for our system.

5.3 Back-end Development

The Flask back-end retrieves from the database, through a series of APIs, the data about doctors, hospitals and patients, along with their public addresses in the blockchain. These addresses can be used to log in via Metamask, allowing the user to perform all interactions pertaining to their role. It is also used to store the association between

each category id and its corresponding name, as well as the metadata of tokens. In particular, this metadata includes the date and time of each appointment, along with its associated hospital, category and id of the prescription used to book it (if it has been booked).

The back-end is mainly used to retrieve all these data that cannot be stored on the blockchain, be it for privacy reasons or storage cost constraints. It also helps to quickly access all available appointments for the requesting patient and retrieve the list of created prescriptions for the requesting doctor. To ensure privacy, the users only through the login are allowed to access their private information.

Rollups are implemented in a Javascript file running as a background job that listens on the private chain for the “block” event, then once every 10 finalized private chain blocks, the rollup of these last 10 finalized blocks is written on the public chain. It is implemented as a transaction between two accounts both assumed to be property of the Italian Ministry of Health, having in the message field 3 integers: the keccak256 hash of the concatenation of the hashes of the 10 private chain blocks being included in the rollup (ordered from older to newer); the block number of the first and last block being included in the rollup, to ensure complete verifiability from users of the private chain.

5.4 Front-end Development

On the login page users will be able to login using their Metamask wallet. The back-end will give the user a specific challenge to sign based on their address, the signed message will be then verified to generate a session token to allow the use of the web application (Figure 12).

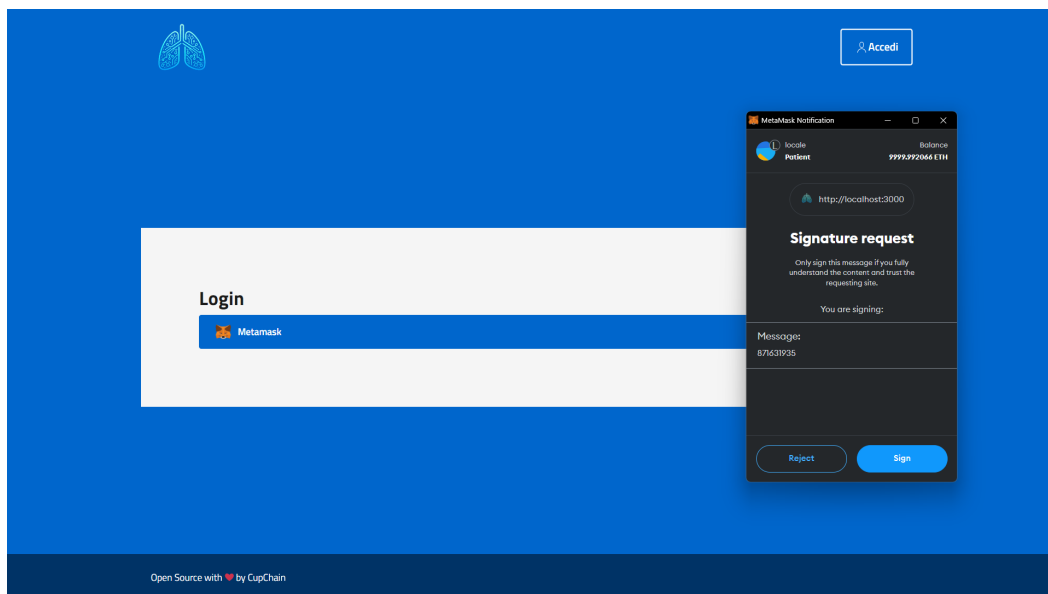


Figure 12: Login Screen

After the login, the user will be redirected to their reserved section. Patients will be forwarded to the “Reservations” page (Figure 13).

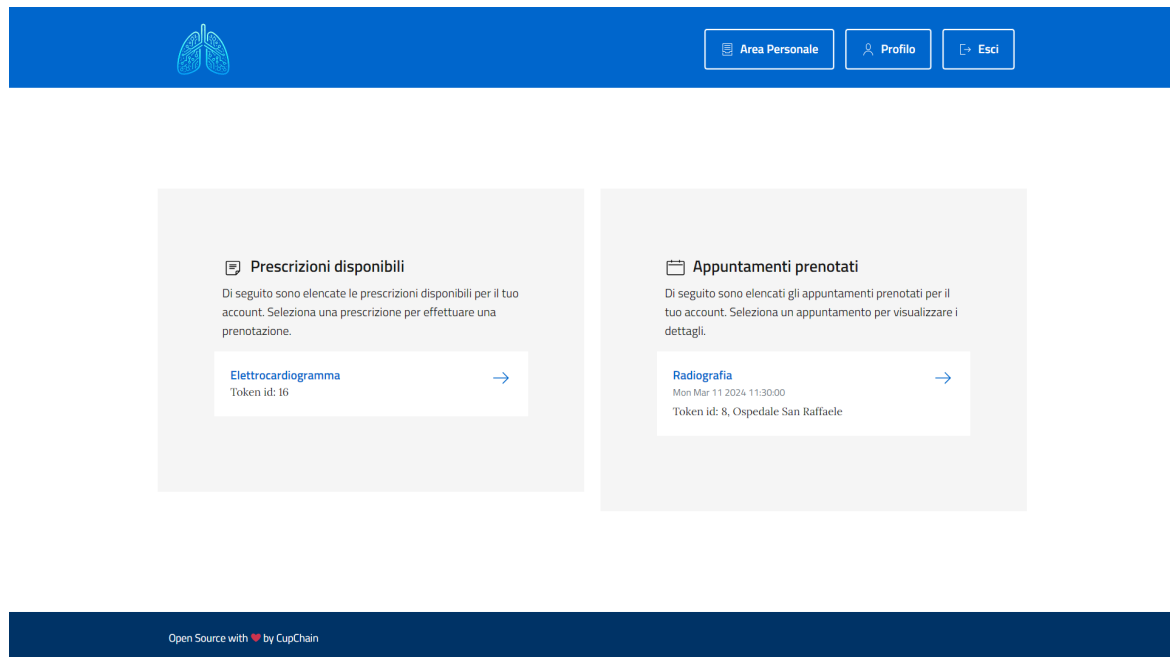


Figure 13: Patient reservations page showing the list of prescriptions and appointments

On this page, they can reserve a new appointment by using a prescription (Figure 14) or manage their already reserved appointments (Figure 16).

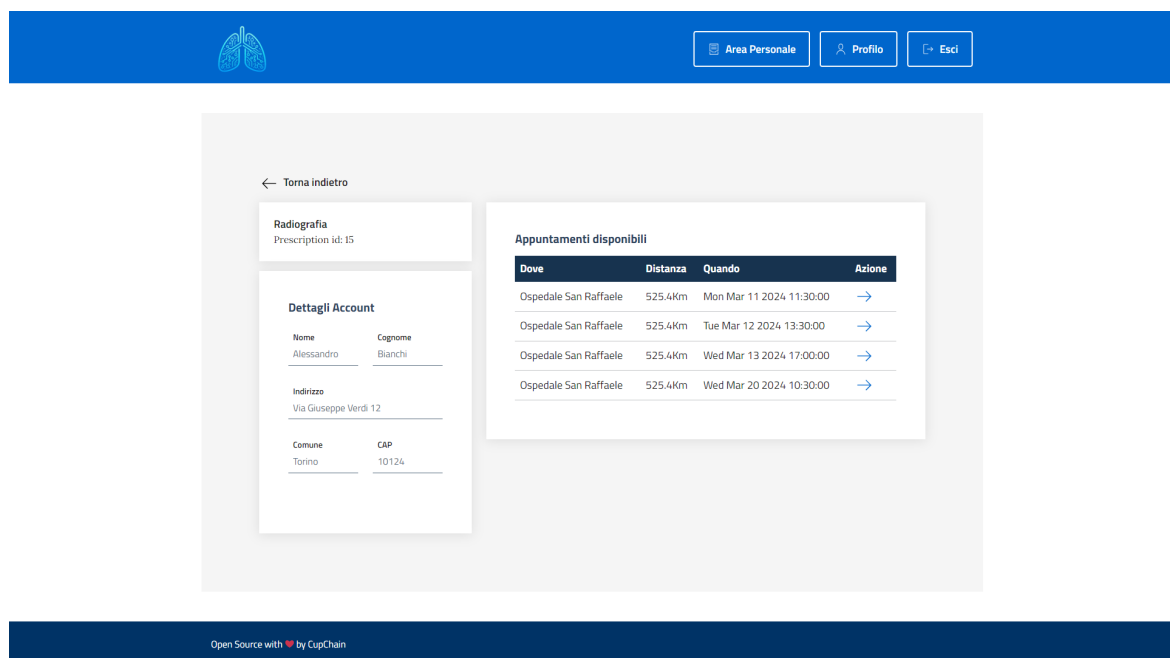


Figure 14: Hospital selection for a prescription

The patient has to confirm the appointment on a dedicated page (Figure 15) he then will be able to have a guarantee of his booking with the display of a notification derived from the event emitted by the smart contract.

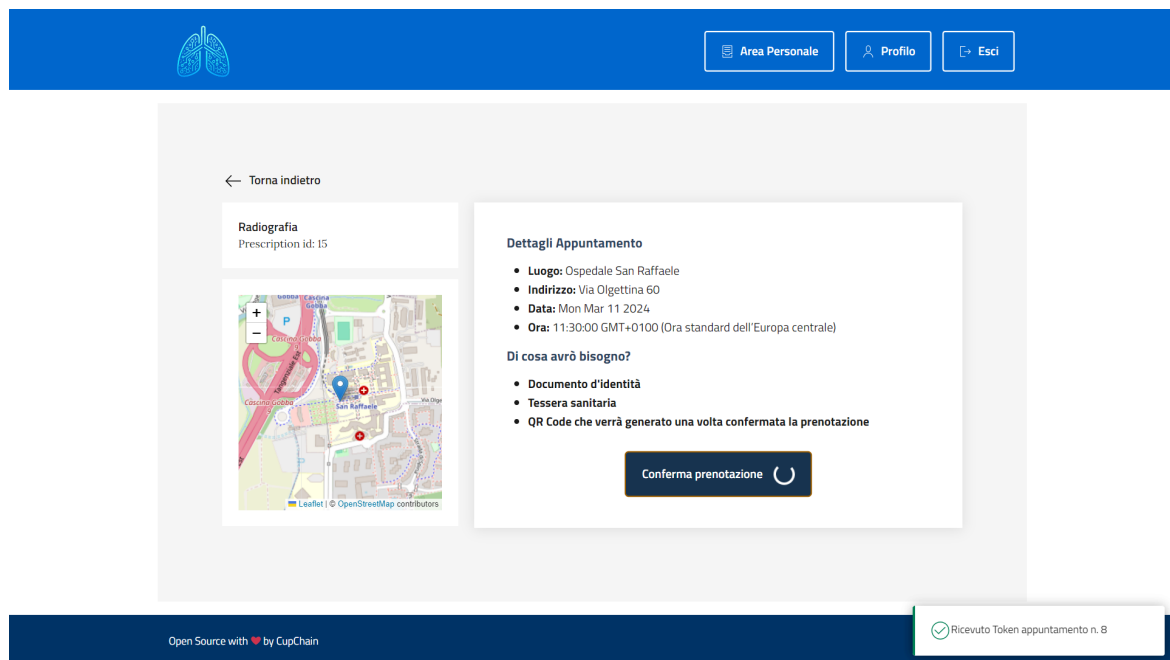


Figure 15: Confirm appointment with notification of successful exchanged token.

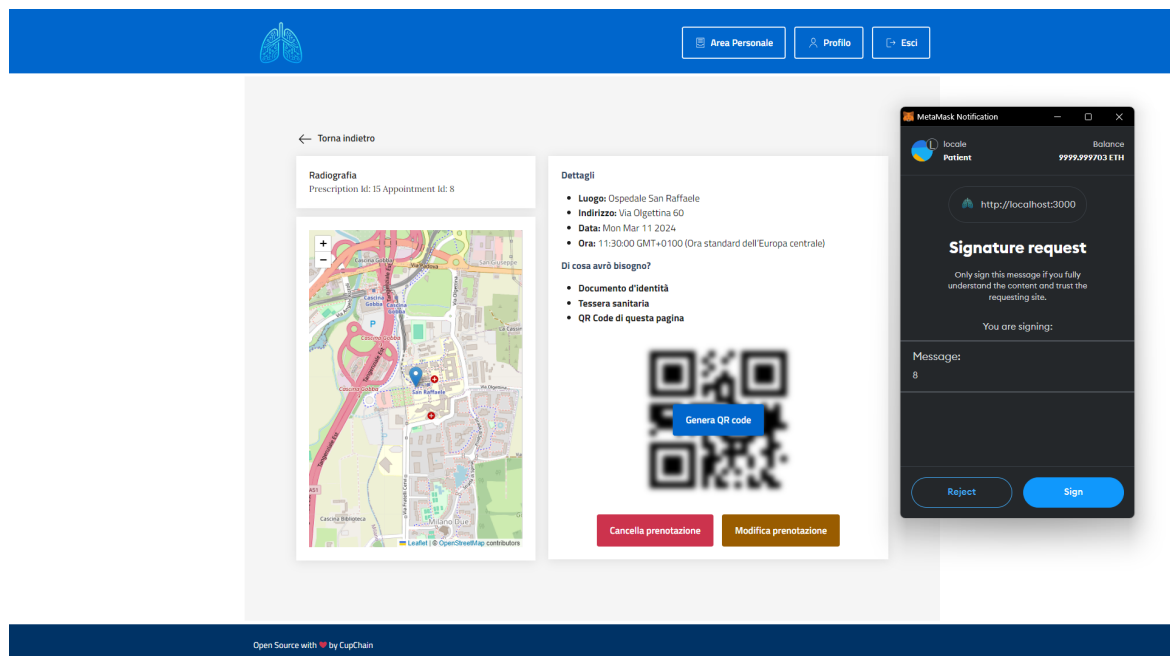
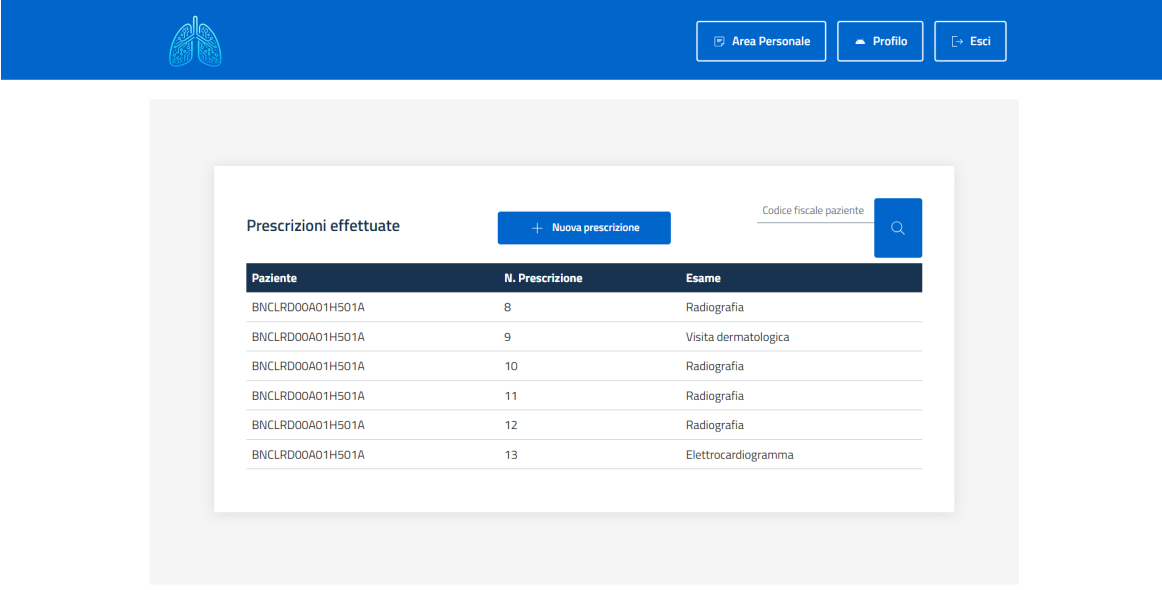


Figure 16: Appointment page showing prompt for signing with Metamask the information needed to generate the QR

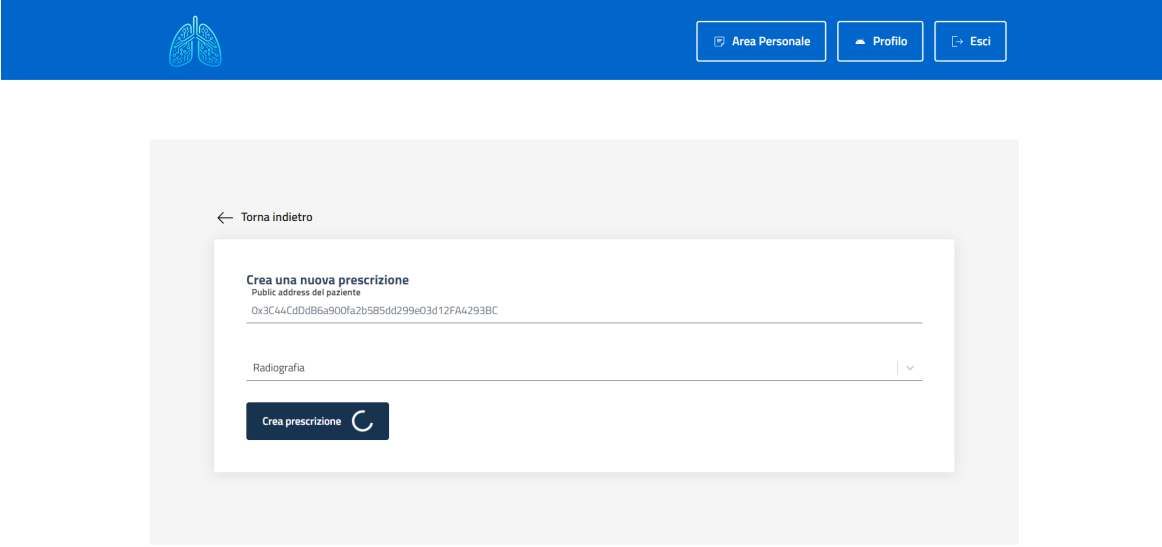
Doctors will be forwarded to the “Prescription list” page (Figure 17). Here they can see the prescription they have already done and create a new one by using the specialized form (Figure 18).



Paziente	N. Prescrizione	Esame
BNCLRD00A01H501A	8	Radiografia
BNCLRD00A01H501A	9	Visita dermatologica
BNCLRD00A01H501A	10	Radiografia
BNCLRD00A01H501A	11	Radiografia
BNCLRD00A01H501A	12	Radiografia
BNCLRD00A01H501A	13	Elettrocardiogramma

Open Source with ❤ by CupChain

Figure 17: List of prescriptions made by the doctor.



← Torna indietro

Crea una nuova prescrizione

Public address del paziente
0x3C44CdD86a900fa2b585dd299e03d12FA4293BC

Radiografia

Crea prescrizione

Open Source with ❤ by CupChain

Figure 18: New prescription creation from a doctor.

Hospitals will be forwarded to the “New appointment” page (Figure 19). On this page, they can create a new appointment by selecting the date and the type of medical exam provided.

The screenshot shows a web interface for creating a new appointment. At the top, there is a blue header bar with a white icon of lungs on the left and three buttons on the right: 'Area Personale' (with a folder icon), 'Profilo' (with a person icon), and 'Esci' (with a door icon). Below the header, the main content area is light gray. In the center, there is a white card titled 'Crea un nuovo appuntamento'. Inside the card, there are three input fields: 'Giorno della visita' with the value '20/03/2024' and a calendar icon; 'Orario della visita' with the value '09:30' and a clock icon; and 'Radiografia' with a dropdown arrow. At the bottom of the card is a blue button labeled 'Crea appuntamento'. At the very bottom of the page, there is a dark blue footer bar with the text 'Open Source with ❤️ by CupChain'.

Figure 19: New appointment creation from the hospital.

6 Limitations

Designing a public healthcare reservation system using blockchain technology offers benefits, such as greater transparency and verifiability. However, it is important to consider the limitations and challenges associated with such a system. Here are some known limitations:

1. **Transaction Speed and Scalability:** Blockchain transactions can take time to process due to the consensus mechanisms involved. In a public healthcare system where e.g. two patients would like to book the same appointment, only after the finalization of a block we would know who managed to book it.
2. **Usability and User Experience:** Cryptographic concepts such as private keys and wallet addresses can be challenging for non-technical users to understand and manage. Teaching patients and healthcare providers how to use tools such as MetaMask or handle private keys may require significant effort and ongoing support. This complexity could lead to user errors and adoption barriers.
3. **Data Privacy and Security:** While blockchain technology provides immutability and transparency, it's critical to ensure that sensitive healthcare data is appropriately protected. Our approach is safe if and only if the pseudo-anonymity is preserved, but since this is a real-world application, we have some problems since people can be identified by matching information such as doctors' public keys and locations of appointments. Useful tools like MARTSIA [22] can be used to alleviate this point.
4. **Regulatory Compliance:** Healthcare is a heavily regulated industry, and implementing a blockchain-based solution requires compliance with existing regulations and standards.
5. **Malicious Behaviours:** Operations such as slashing, penalties and whistleblowing in a blockchain environment disallow users to perform malicious operations. Actually, in our proof of concept, no one of these automatic operations can be performed against the incorrect use of the system. The system is designed to be of public utility, so making the patients pay for an operation is not the right choice. In a real scenario, finding a way to balance costs and also enable penalties could be a wise strategy.

7 Conclusions

In conclusion, we truly believe that blockchain can be applied to revolutionize public administration (PA) applications by making them more democratic, providing a secure, transparent and globally accessible alternative to traditional systems involving third parties.

With Cupchain we developed a proof-of-concept for a public healthcare reservation system using multiple advanced topics such as interaction between two blockchains (private and public chain), exchange of tokens and Wallet Signature Login.

This project could be extended in the future to create a fully functional system for use within the public healthcare sector. Exploring alternative blockchain platforms with potentially lower costs and higher throughput could be an intriguing avenue for further development. Platforms like Algorand [23], for instance, offer promising features that could enhance the scalability and cost-effectiveness of the system.

References

- [1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at <https://metzdowd.com>*, 03 2009.
- [2] Vitalik Buterin. Ethereum white paper: A next generation smart contract & decentralized application platform, 2013.
- [3] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.
- [4] Vitalik Buterin, Diego Hernandez, Thor Kamphofner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Y. Wang, and Yan X Zhang. Combining ghost and casper. *ArXiv*, abs/2003.03052, 2020.
- [5] Yonatan Sompolinsky and Aviv Zohar. Accelerating bitcoin’s transaction processing. fast money grows on trees, not chains. *IACR Cryptol. ePrint Arch.*, 2013:881, 2013.
- [6] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *ArXiv*, abs/1710.09437, 2017.
- [7] Ministero della Salute. I principi del servizio sanitario nazionale (ssn). <https://www.salute.gov.it/portale/lea/dettaglioContenutiLea.jsp?lingua=italiano&id=5073&area=Lea&menu=vuoto>, 2019.
- [8] Domenica Matranga and Laura Maniscalco. Inequality in healthcare utilization in italy: How important are barriers to access? *International Journal of Environmental Research and Public Health*, 19:1697, 02 2022.
- [9] Tim K. Mackey, Tsung-Ting Kuo, Basker Gummadi, Kevin A. Clauson, George Church, Dennis Grishin, Kamal Obbad, Robert Barkovich, and Maria Palombini. ‘fit-for-purpose?’ – challenges and opportunities for applications of blockchain technology in the future of healthcare. *BMC Medicine*, 17(1):68, Mar 2019.
- [10] Israa Abu-elezz, Asma Hassan, Anjanarani Nazeemudeen, Mowafa Househ, and Alaa Abd-alrazaq. The benefits and threats of blockchain technology in healthcare: A scoping review. *International Journal of Medical Informatics*, 142:104246, 2020.
- [11] Statista. Percentage of people with trust in hospitals and clinics in 2021, by country. <https://www.statista.com/statistics/1071631/trust-levels-towards-hospitals-clinics-in-select-countries/>, 2021.
- [12] Il Sole 24 Ore. Verifiche nas su 3.884 liste d’attesa, denunciati 26 medici. https://www.ilsole24ore.com/art/verifiche-nas-3884-liste-d-attesa-denunciati-26-medici-AF1YHYm?refresh_ce=1#commentsform, 2023.

- [13] Il Fatto Quotidiano. Liste d’attesa cambiate per favorire pazienti privati o agende chiuse per andare in ferie, blitz del nas: denunciati 26 operatori sanitari. <https://www.ilfattoquotidiano.it/2023/09/08/verifiche-dei-nas-su-oltre-3mila-liste-dattesa-in-tutta-italia-denunciati-26-operatori-sanitari-per-irregolarita-che-compromettevano-il-ssn/7284948/>, 2023.
- [14] K. Wüst and Arthur Gervais. Do you need a blockchain? *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54, 2018.
- [15] Developers Italia. design-react-kit: A react component library for italian pa applications. <https://github.com/italia/design-react-kit>, 2024.
- [16] Vitalik Buterin. A rollup-centric ethereum roadmap. <https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698>, Oct 2020.
- [17] Vitalik Buterin. An incomplete guide to rollups. <https://vitalik.eth.limo/general/2021/01/05/rollup.html>.
- [18] Vitalik Buterin. A step-by-step roadmap for scaling rollups with call-data expansion and sharding. https://notes.ethereum.org/@vbuterin/data_sharding_roadmap.
- [19] Vitalik Buterin and Dankrad Feist. Eip-4844: Shard blob transactions. <https://eips.ethereum.org/EIPS/eip-4844>, Feb 2022.
- [20] William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs. Erc-721: Non-fungible token standard. <https://eips.ethereum.org/EIPS/eip-721>, 2018.
- [21] OpenZeppelin. openzeppelin: A library for secure smart contract development. <https://docs.openzeppelin.com/contracts/5.x/>, 2024.
- [22] Edoardo Marangone, Claudio Di Ciccio, Daniele Friolo, Eugenio Nerio Nemmi, Daniele Venturi, and Ingo Weber. Martsia: Enabling data confidentiality for blockchain-based process execution. In *Enterprise Design, Operations, and Computing: 27th International Conference, EDOC 2023, Groningen, The Netherlands, October 30 – November 3, 2023, Proceedings*, page 58–76, Berlin, Heidelberg, 2023. Springer-Verlag.
- [23] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. SOSP ’17, page 51–68, New York, NY, USA, 2017. Association for Computing Machinery.