MediVault

Leo Unoki, Piet Wolff, Szilard Parrag, Anna Erdi

OVERVIEW

The centralized manner of storing patients' medical records within and across countries raises several issues. The sensitive nature of the data requires privacy and secure storage, while in case of an emergency situation the access to this data might be crucial.

To solve this problem, we propose a secure storage for medical records on the blockchain, which is accessible from anywhere and gives the right to the verified actors to read and write data to the blockchain.

(Basically it would use a general "secrets vault" which would allow anybody to store/share secrets, and a group of the users (e.g. verified users == doctors) could verify secrets as well.)

FEATURES

Problem 1: Going on a vacation, the local hospital doesn't have access to your medical record (e. g. GDPR vs visiting a non-EU country).

Solution: Allow patients to post records and get it validated by doctors anywhere in the world.

Problem 2: Medical records are sensitive data, users want to control over which bit of the data to be shared. (e.g.family members sharing specific parts of their medical histories)

Solution: Users can give permissions to let doctors access the

Logic for account management - Pallet 1	Logic for sharing - Pallet 2
register_as_patient	submit_record
register_as_doctor	retrieve_records
later on: access level settings	share_record
	attest_record
	later on: message handling

CONTRIBUTION PLAN

Leo	Implementing Pallet1, review Pallet2
Szilard	Implementing Pallet1, review Pallet2
Piet	Implementing Pallet2, review Pallet1
Anna	Implementing Pallet2, review Pallet1

MILESTONES

1. Implement record keeping logic

- Create an account for patients and doctors.
- Patients can submit/retrieve encrypted medical records (only the patient can decrypt it).
- Doctors (privileged users) can submit medical records which are signed by them.

    //Storage

    Doctors = HashSet<AccountId>

    Patients = HashMap<AccountId, Vec<RecordContent>>

    enum Record {

        VERIFIED(RecordContent, some_doctors_signature),

        NON_VERIFIED(RecordContent)

    }

2. Implement sharing feature and fees

- Patients can share their records with doctors. All permitted doctors are able to decipher the whole records locally.
- Doctors can attest to non-verified records (turn the non verified to verified record).
- Justified Fees are in place for every interaction to meet the block weight limit (Benchmarking).

3. Notification message system and Access layers

Message NFTs for on-chain communication between patients and doctors.

Access control implementation.