MedRec

Your medical records.
Reimagined.

v1.0

Login as patient

username

password

Login

Create an Identity

First name

Last name

username

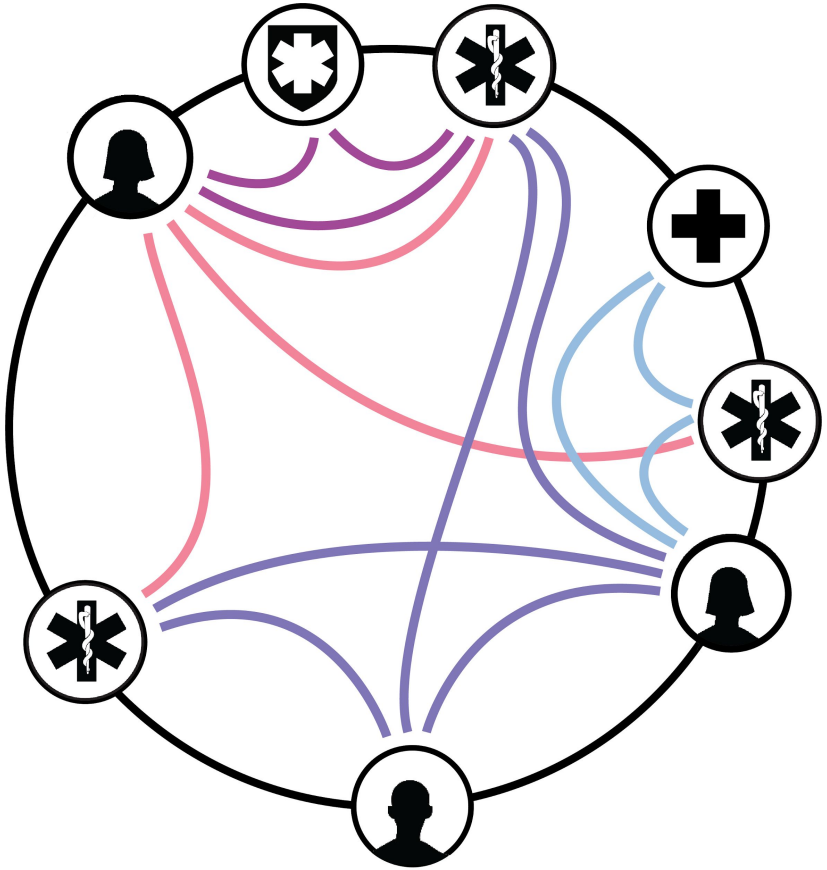
password

Create

medrec 2.0

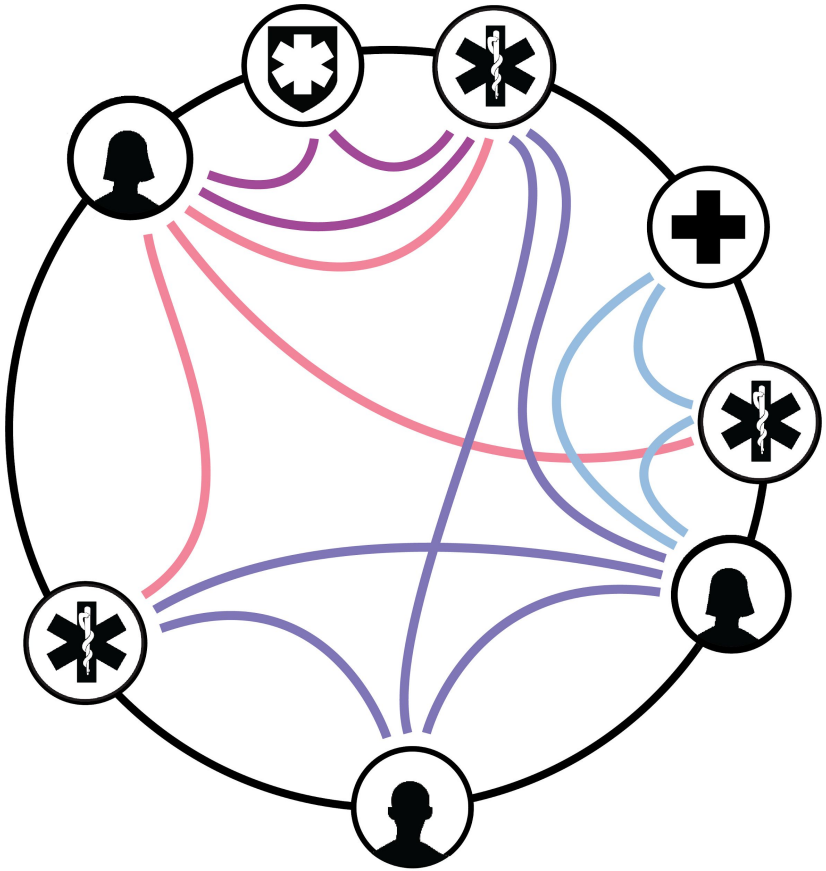
MedRec is an Ethereum-based management system to give patients control over their medical records. MedRec 2.0 is...

- Decentralized
- Built on extensible smart contracts
- Non-commercial and open
- Intended for incremental adoption
- Anonymous and private



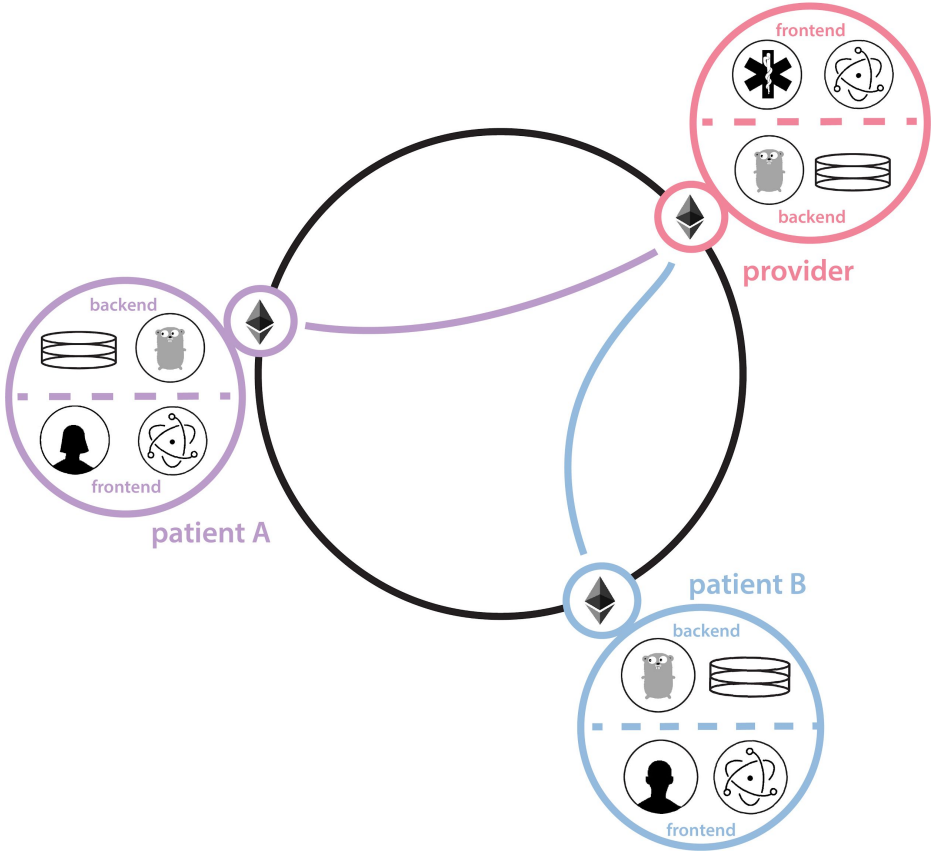
medrec is a *network*

- A smart contracts on the blockchain encode access rights
- Open contracts subscribed to and composed by patients/users
- Signers (medical providers) join the network as full nodes
- Signers add new blocks to the chain through *proof of authority*
- A node permits read access to contracts and write access when approved by a signer



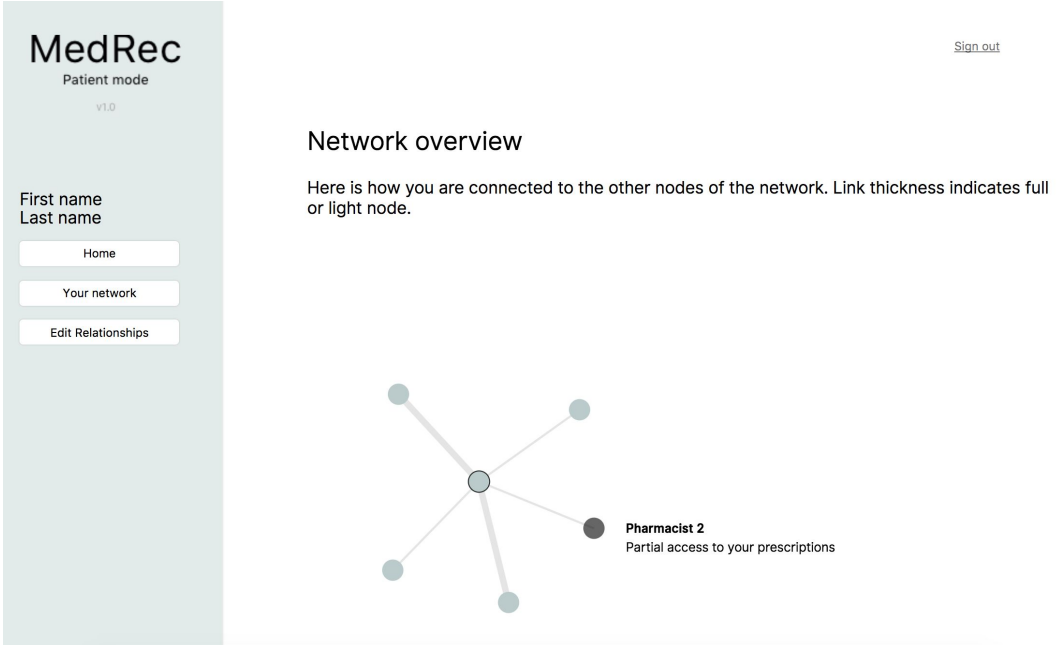
app overview

- Patients and providers interface to MedRec via an Electron desktop app
- There are two databases and database managers. The app makes calls to a Golang database manager (read+write)
- Both Golang backend and JS frontend interface with the Ethereum blockchain
- All app credentials are stored locally and locked by password



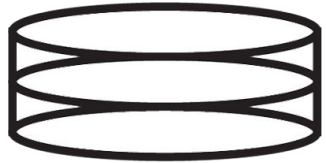
interface

- Electron executable
- authentication credentials stored locally
- different UI for provider and patient
- basic data exploration and visualization



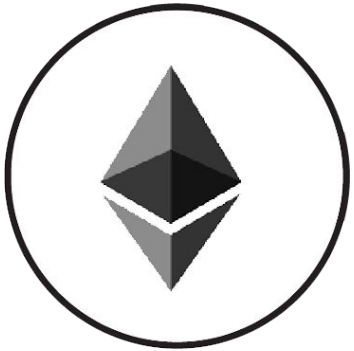
authentication

- Requests are signed using the patient's ethereum address and private key
- the provider database decodes the request, and the address of the patient
- The database returns the records associated with that patient
- Public key encryption ensures valid, unfakeable digital signatures

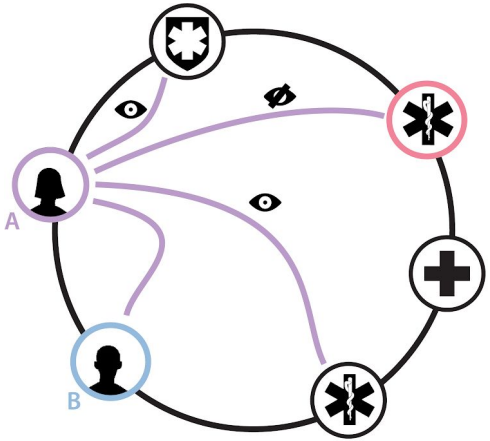


blockchain - contracts

- Smart contracts define relationships between agents in the network.
- The parameters of each relationship (what they can do) are defined by sets of viewing permissions
- For each relationship contract, a number of permissions are set about what each agent can and can not do with respect to the other
- Vision: these should be dynamic, custom, and can respond to external factors (oracles)

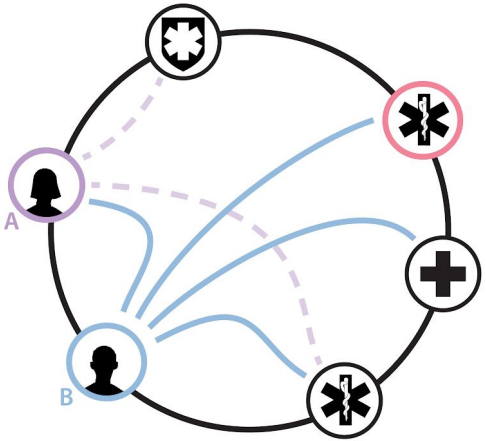


vision: modifiable smart contracts



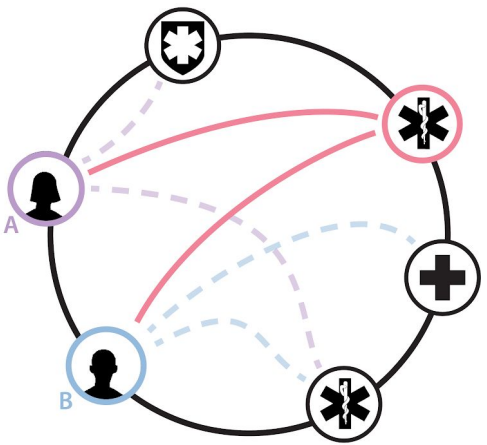
patient A view
(custom permissions)

patient A has set her viewing permissions so patient B can see some (not all) of her records



patient B view
(auto permissions)

patient B leaves his permissions as default: only ‘trusted providers’ can view his records

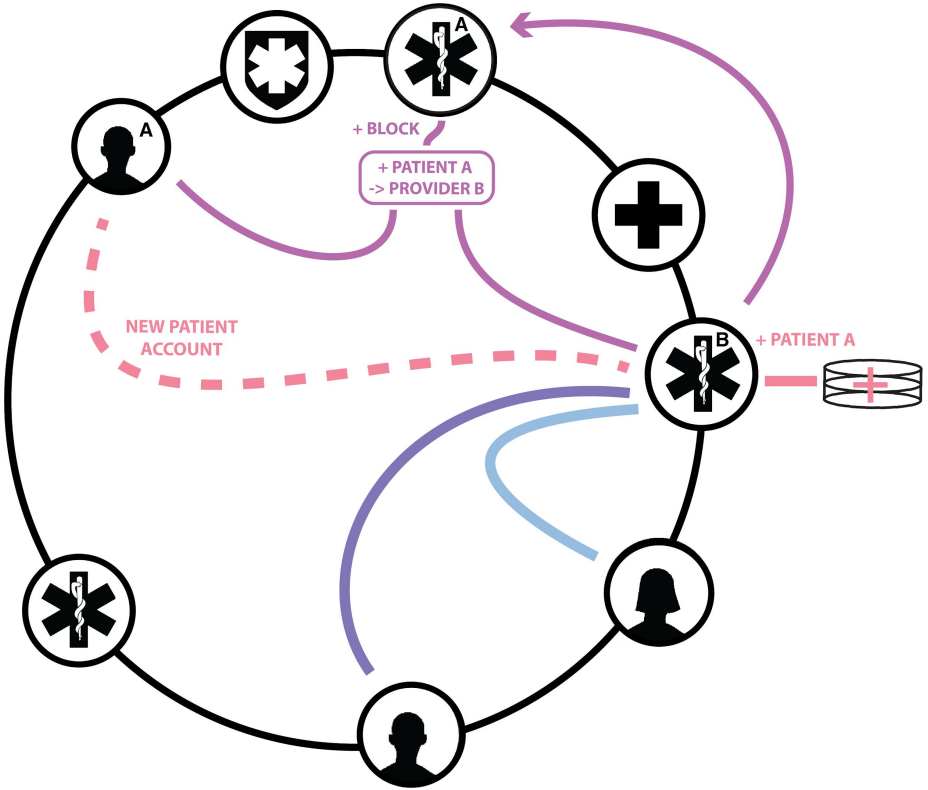


provider view

if patients A and B give permission, the provider views relationships with past providers

preserving privacy: delegate contracts

- We use a system of ‘delegate contracts’ to protect differential privacy
- For each new patient relationship, providers create a ‘delegate account’
- With a flat network, it is no longer possible to triangulate identities



the future

- Testing scalability
- Open sourcing
- Deployment
- Identity management > medical records management