Medical Distributed Ledger Technology

Programming with Advanced Computer Languages - Python
May 24th, 2019
Team Snakes: Benjamin Cook, Dominik Eitljörg, Johanna Husson, Christopher Newson, Dean van der Merwe.

Description

This project creates a distributed ledger technology (DLT) to store patient data. The intended users are physicians who need to access and modify patients' files during or after a medical consultation, to register diagnoses and prescribed medications. Pharmacists are also potential users, to register medications bought by a given patient. The blockchain architecture ensures traceability of all actions undertaken on a patient file by alerting the user when a file has been tampered with.

A patient file is stored as a dictionary. A hashing algorithm, which assigns a cryptographic key to its input (here, the input being a dictionary), ensures the integrity of patient data because: 1- It is impossible to have the same hash for two different inputs; 2- The exact same patient dictionary produces the same hash; 3- Conversely, any change to the dictionary modifies the hash.

A blockchain (BC) is automatically created when the program is started. The blockchain is only stored as long as the program is running. Every block includes an index number, the previous block's hash and the hash of the patient data. Whenever the user changes his data, a new block is created with a new hash. Hashes of every action are saved in a list which can be visualized by the user.

When starting the program, the user is prompted to either add a new patient file or load an existing file. In the latter case, the code checks the hashes from the previous blocks and uses this to verify whether the data was once created within the program or if it was created / altered by some third party.

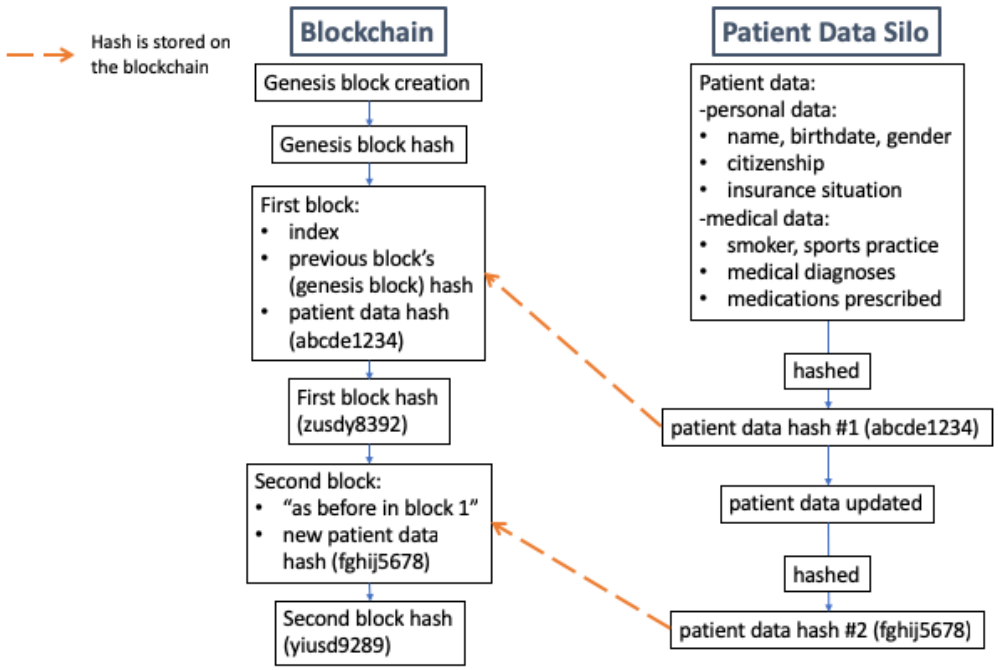

Figure 1: Visualisation of our blockchain technology for patient data.
The patient data itself is not stored on the blockchain, only the hash identifying it is stored. Therefore a user cannot see the content of a patient file, but can only see the resulting hash to verify data has not been tampered with.

Instructions

1. Run the program to create a blockchain:

You are asked to enter the first patient's information. A new block is created.
You can later add new patient files (Option 1). Every new file creates a new block on the BC.
Alternatively, you can load an existing patient file to modify it (Option 2). In case of modification, a new hash is created, and a new block is added to the blockchain.
Finally, you can visualize the list of hashes stored on the blockchain (Option 3).

2. Test the ability of the program to identify a corrupted file:

After having run the program a first time and entered at least one patient dictionary (let's call it abcde1234) in it, log out of the program (Option 4). Your patient file (abcde1234) is stored locally on your computer.
Start a new program and try to load the patient file you had created on the previous program (abcde1234). Receive a message "*File couldn't be verified with existing blockchain entries*": the new hash does not correspond to any hash loaded on your current blockchain.

3. Terminate the program (Option 4)

The blockchain is lost when the program is terminated.

Requirements

None, all libraries come installed with Python.