Ayman Ibne Hakim, ID- 2103364042
Chowdhury Nafis Faiyaz, ID- 1931841642

Description of file storage:

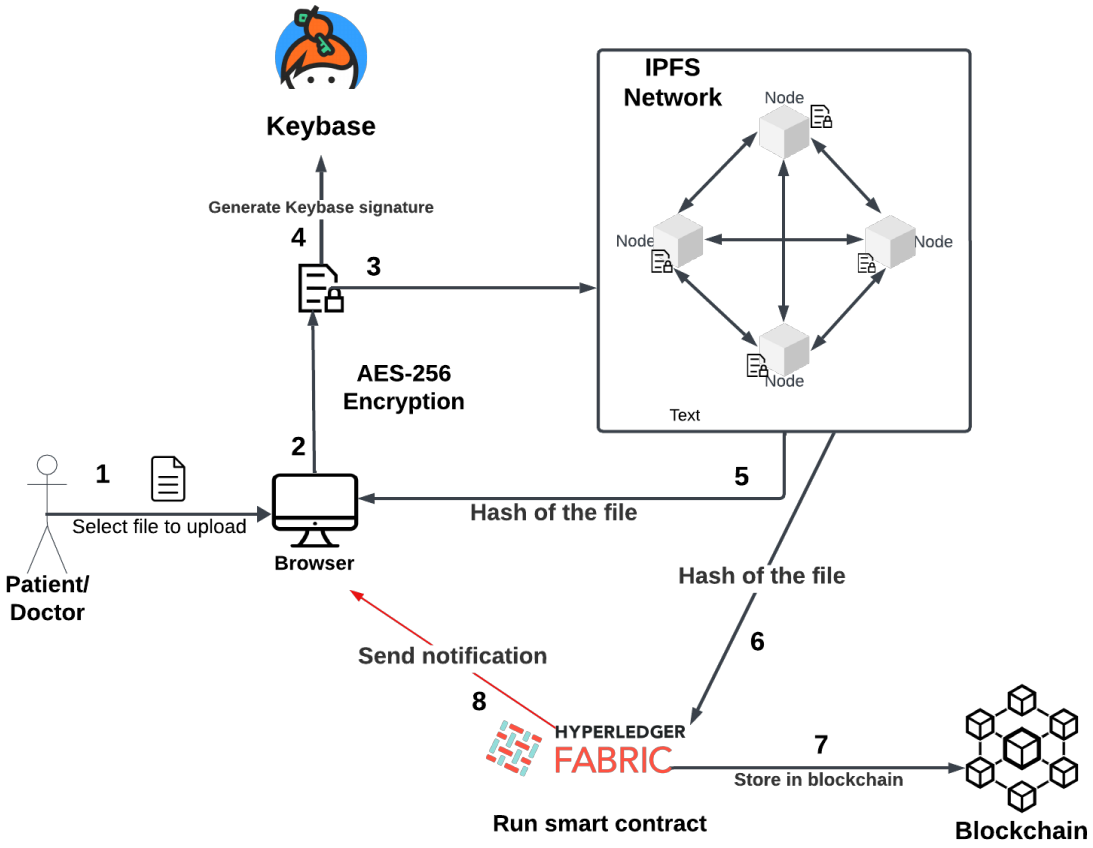
We will be using Hyperledger Fabric to create and manage our private blockchain network, IPFS to store files, and Keybase to store and share the encryption key of the files.

Hyperledger Fabric is a distributed ledger technology (DLT) platform that is designed to support a wide range of enterprise applications. It is a permissioned blockchain, which means that only authorized nodes can participate in the network.

IPFS (InterPlanetary File System) is a distributed file system that allows users to store and share files in a decentralized way. IPFS uses a peer-to-peer network to store files.

Keybase is a platform that allows users to verify their online identities and share encrypted files and messages with others. Keybase does this by linking users' online identities to a public key. After encrypting the file, the encryption key will be stored in Keybase and doctors will be able to request for the encryption key from the patient.

User uploads file:

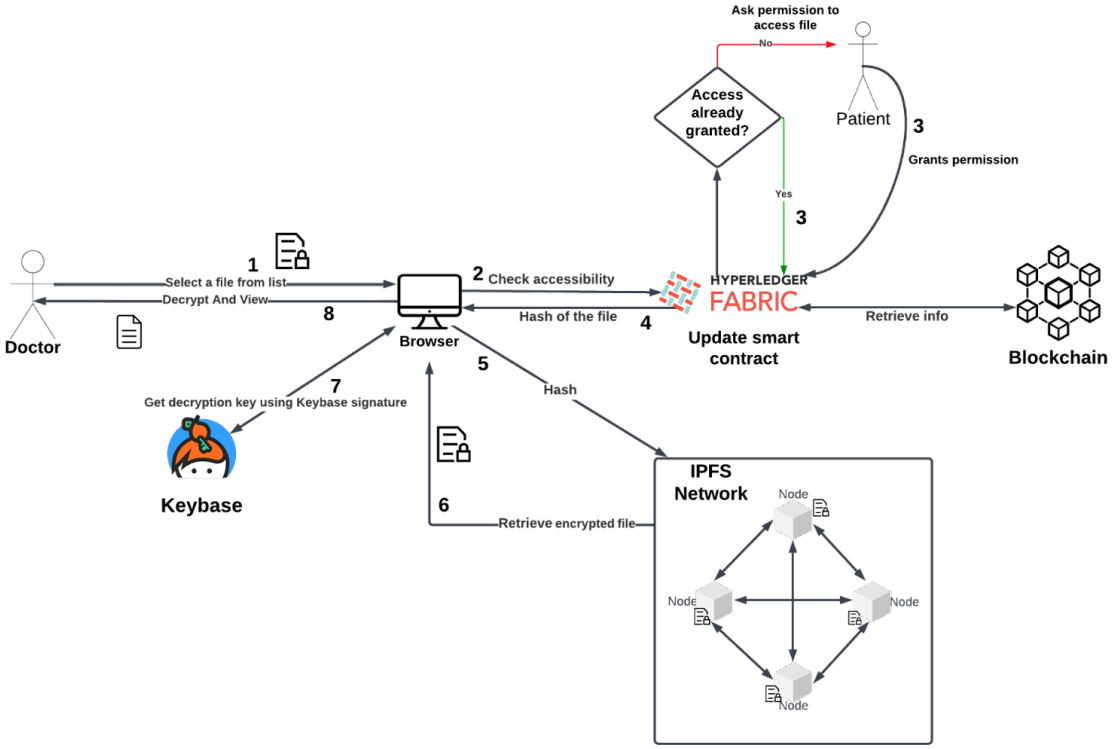


When the patient/doctor selects a file and clicks upload, the browser takes the file and generates a random 256-bit AES encryption key and uses it to encrypt the file. The browser then generates a Keybase signature for the encrypted file and stores the signature in the file's metadata. The encrypted file is then sent to the IPFS where the file is broken down into smaller chunks and each chunk is stored on a different node. The IPFS generates a hash of the file using SHA-256 cryptographic algorithm and sends the hash to the smart contract which is run on Hyperledger Fabric on a private blockchain. Upon receiving the hash of the file, the smart contract creates

a new transaction that contains the hash of the file. The transaction is then submitted to the blockchain network. The blockchain network then validates the transaction and adds it to the blockchain. The smart contract then returns the transaction ID to the client application. Similarly, another smart contract will run that stores the files metadata. The smart contract will create a new transaction that contains the file's metadata. The smart contract which contains the list of all the patient's files' metadata validates the transaction and adds the new file's metadata to the list and finally returns the transaction ID to the client.

**If it is the doctor who is creating the file for the patient, the ownership will be transferred to the patient and the doctor will need the patient's approval to view the file.

Doctor tries to view file:



When the doctor clicks on a 'View Document' button, the browser sends a request to the smart contract to request permission to view the file. The request includes the hash of the file and the doctor's address. To confirm that the doctor has permission to view the file, the smart contract validates the request. The smart contract also verifies that the file exists in IPFS. If the request is valid, the smart contract sends a notification to the owner of the file. If the request is rejected, a message will be shown on the screen stating access denied. If the owner grants permission, the owner signs a transaction that grants the doctor access to the file. The smart contract updates the access control list for the file to grant permission to the doctor. Using the hash, the encrypted file is retrieved from the IPFS. Using the Keybase signature, the decryption key is retrieved and used to decrypt the file. Finally, the decrypted file is viewed on the screen.