BLOCKCHAIN FOR SECURELY SHARING MEDICAL INFORMATION

By Prabha Pamula
D17B / 51

Introduction

In the healthcare sector, the secure sharing of patient data is not just a necessity but a fundamental requirement for delivering high-quality care.

Healthcare, as we know it, relies heavily on the ability to access and exchange medical information efficiently and securely.

This data encompasses a patient's medical history, treatment plans, test results, medications, and more. This data is important for effective healthcare delivery.

Challenges of existing systems

Data Breaches:

- Data breaches are a significant challenge in healthcare.
- Breaches can occur due to various reasons, including cyberattacks, insider threats, and accidental exposures.

Lack of Interoperability:

- Healthcare providers often use a variety of EHR systems with different data formats and standards.
- Lack of interoperability leads to data barriers, making it difficult for healthcare professionals to access complete patient histories and coordinate care.

Fragmented Data Storage:

- Patient data is often distributed among hospitals and other healthcare entities.
- Healthcare providers may struggle to obtain a complete picture of a patient's medical history, which can impact the quality and timeliness of care.

Blockchain

Blockchain technology is a decentralized and immutable digital ledger system that records transactions across multiple computers, ensuring security and transparency. In a blockchain system, information is grouped into "blocks," linked chronologically to form a "chain." Every block in the chain contains a set of transactions linked to the previous block using cryptography, making it difficult to change data in any block without changing all the subsequent blocks.

Blockchain is a healthcare data security game-changer. It offers decentralization, eliminating central control. Immutability prevents data alteration, and transparency, coupled with cryptographic techniques, ensures robust data security

Benefits of using Blockchain for sharing medical records

- **Enhanced Security :**
  Blockchain employs robust cryptographic techniques to secure healthcare data. These techniques ensure that patient information is protected from unauthorized access, tampering, and breaches.

- **Decentralization :**
  Blockchain’s decentralized architecture that allows multiple participants to access the same dataset. This eliminates the need for a central authority to store data and reduces the risk of a single point of failure, thus enhancing data availability.

- **Immutability :**
  Immutability in blockchain ensures that once data is recorded, it cannot be altered or deleted. This feature prevents data tampering and unauthorized changes to patient records.

- **Transparency :**
  Blockchain provides transparency by recording all data transactions on an immutable ledger. The ledger is accessible to authorized parties for viewing historical data transactions.

  Blockchain creates a secure ledger of patients' health records that allows the smooth sharing of information across different healthcare providers while maintaining patient consent and control. This reduces the need for patients to repeatedly share their medical records and brings more speed and efficiency to the healthcare system.

Technologies

- **Ethereum :**
  Ethereum plays a crucial role in healthcare applications.
  The smart contracts, are self-executing agreements with predefined rules. These contracts automate and enforce agreements related to patient consent, data sharing, and billing.
  It can be employed in applications ranging from Electronic Health Records (EHRs) to supply chain management for pharmaceuticals.

- **Hyperledger Fabric :**
  Hyperledger Fabric is an open-source platform for building distributed ledger solutions. Healthcare institutions demand robust security and compliance. Hyperledger Fabric provides the tools to create private, permissioned blockchains that adhere to these requirements. It allows for the creation of private, consortium blockchains, ensuring data confidentiality.

Examples

- **MediBloc :**
  The MediBloc is using blockchain technology to enable patients to control their own medical data and allow access to healthcare providers. The platform also facilitates interoperability among healthcare institutions by enabling seamless data exchange between different healthcare providers.

- **BurstIQ :**
  BurstIQ provides a platform to help healthcare companies manage their patient data safely and securely. They leverage blockchain technology to offer a secure, decentralized environment for storing, sharing, and managing sensitive health data. By utilizing blockchain's immutability and transparency, BurstIQ ensures the integrity and privacy of patient information and empowers patients to have greater control over their data. This ensures that only authorized healthcare providers can access the data.

Conclusion

The transformative potential of blockchain for healthcare is undeniable. By leveraging the power of blockchain, healthcare organizations can protect sensitive medical data, aid faster data exchange, streamline operations, and provide better healthcare services to patients. Applying blockchain can also save significant time and money for healthcare providers and patients.

Experiment 6 : Smart Contract 1

```
1   // SPDX-License-Identifier: GPL-3.0
2
3   pragma solidity >=0.7.0 <0.9.0;
4
5   contract PatientRecord {
6       address public owner;
7       string public patientName;
8       string public medicalRecord;
9
10      mapping(address => bool) public authorizedProviders;
11
12      event MedicalRecordUpdated(string newMedicalRecord);
13
14      constructor(string memory _patientName) {    infinite gas 727400 gas
15          owner = msg.sender;
16          patientName = _patientName;
17          authorizedProviders[msg.sender] = true; // The owner (patient) is initially authorized
18      }
19
20      modifier onlyOwner() {
21          require(msg.sender == owner, "Only the owner can perform this action");
22          _;
23      }
24
25      modifier onlyAuthorized() {
26          require(authorizedProviders[msg.sender], "Not authorized to perform this action");
27          _;
```

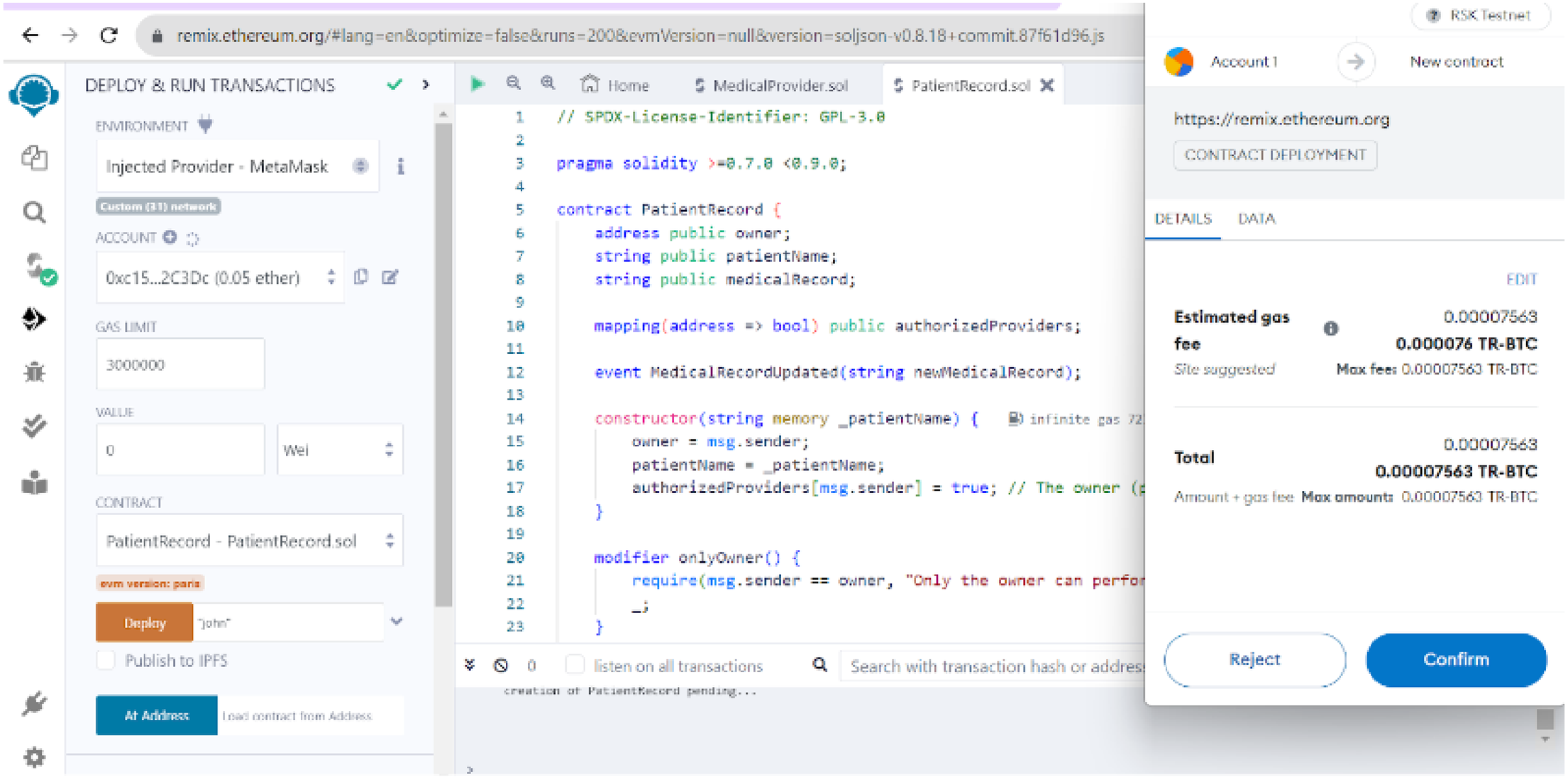
Experiment 6 : Smart Contract 2

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

import "./PatientRecord.sol";

contract MedicalProvider {
    address public owner;
    string public providerName;


    event MedicalRecordUpdateAttempt(address patientContract, string newRecord);

    constructor(string memory _providerName) {    infinite gas 1643800 gas
        owner = msg.sender;
        providerName = _providerName;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can perform this action");
        _;
    }

    function createPatientRecord(string memory patientName) public returns (address) {    infinite gas
        PatientRecord newPatientRecord = new PatientRecord(patientName);
        return address(newPatientRecord);
    }


  function updateRecord(address patientContract, string memory newRecord) public {    infinite gas
      PatientRecord patient = PatientRecord(patientContract);
      require(patient.authorizedProviders(address(this)), "Not authorized to update this patient's record");
      patient.updateMedicalRecord(newRecord);
      emit MedicalRecordUpdateAttempt(patientContract, newRecord); // Log the attempt event
  }

  function viewRecord(address patientContract) public view returns (string memory) {    infinite gas
      PatientRecord patient = PatientRecord(patientContract);
      require(patient.authorizedProviders(address(this)), "Not authorized to view this patient's record");
```

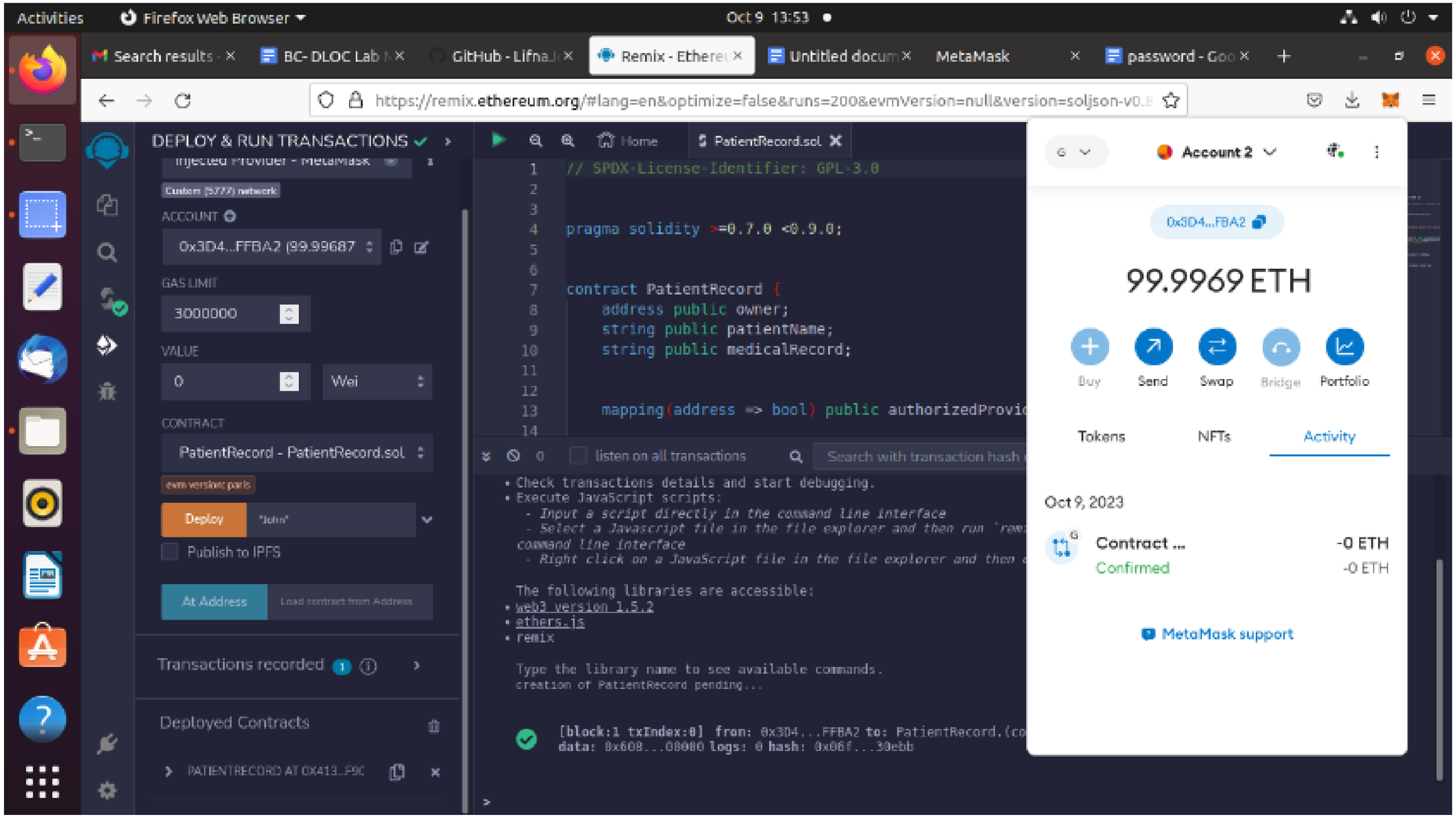
Experiment 7 : Integration with Metamask

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity >=0.7.0 <0.9.0;
4
5  import "./PatientRecord.sol";
6
7  contract MedicalProvider {
8      address public owner;
9      string public providerName;
10
11
12     event MedicalRecordUpdateAttempt(address patientContract, string newRecord);
13
14     constructor(string memory _providerName) {    infinite gas 1643800 gas
15         owner = msg.sender;
16         providerName = _providerName;
17     }
18
19     modifier onlyOwner() {
20         require(msg.sender == owner, "Only the owner can perform this action");
21         _;
22     }
23
24     function createPatientRecord(string memory patientName) public returns (address) {    infinite gas
25         PatientRecord newPatientRecord = new PatientRecord(patientName);
26         return address(newPatientRecord);
27     }
28
29     function updateRecord(address patientContract, string memory newRecord) public {    infinite gas
30         PatientRecord patient = PatientRecord(patientContract);
31         require(patient.authorizedProviders(address(this)), "Not authorized to update this patient's record");
32         patient.updateMedicalRecord(newRecord);
33         emit MedicalRecordUpdateAttempt(patientContract, newRecord); // Log the attempt event
34     }
35
36     function viewRecord(address patientContract) public view returns (string memory) {    infinite gas
37         PatientRecord patient = PatientRecord(patientContract);
38         require(patient.authorizedProviders(address(this)), "Not authorized to view this patient's record");
```

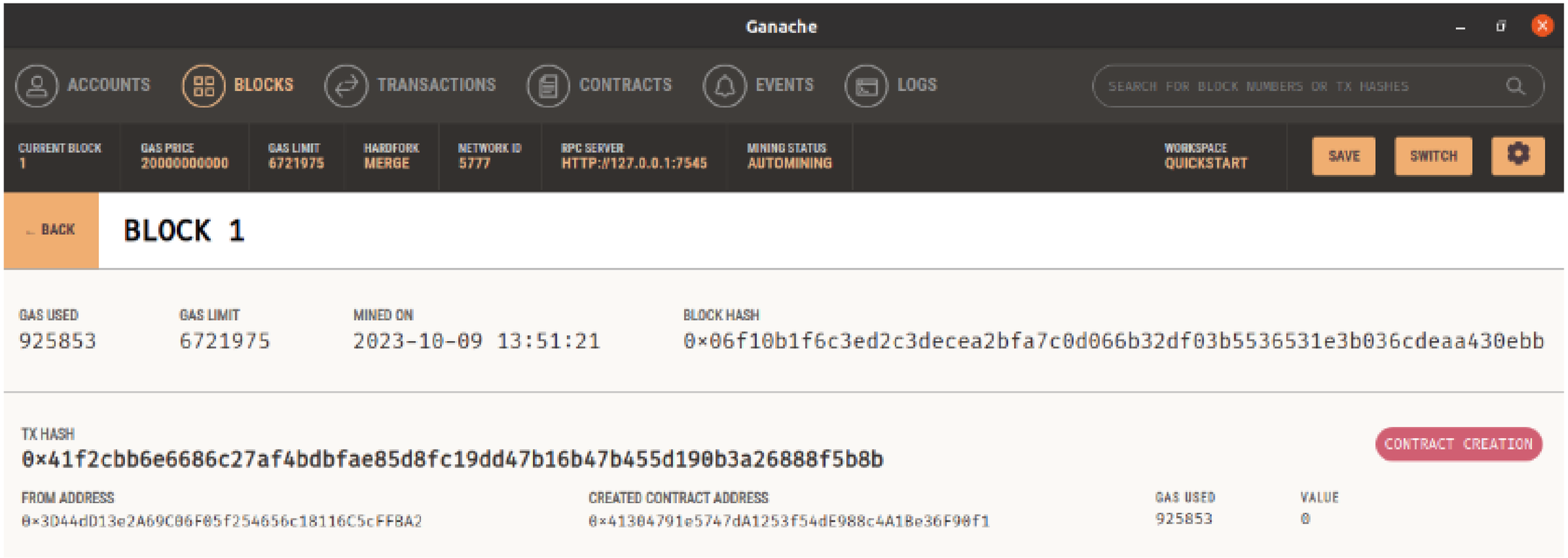
Experiment 7 : Integration with Metamask



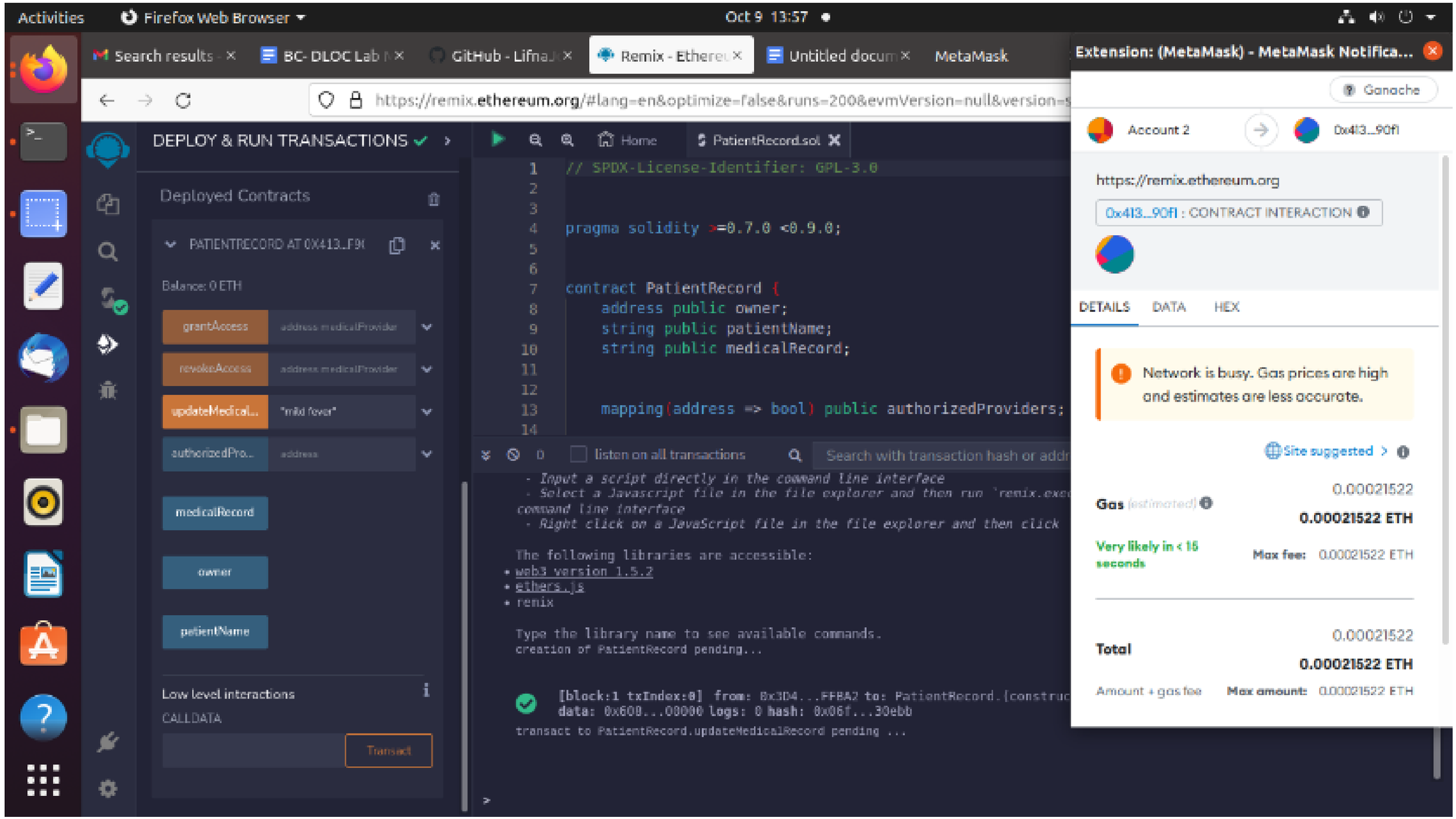
Experiment 8 : Deploying with Ganache Account



Experiment 8 : Deploying with Ganache Account



Experiment 8 : Deploying with Ganache Account



References

MediBlocks: secure exchanging of electronic health records (EHRs) using trust-based blockchain network with privacy concerns

Secure decentralized electronic health records sharing system based on blockchains

Other References :
https://www.turing.com/resources/blockchain-for-healthcare

https://www.mosmedicalrecordreview.com/blog/blockchain-technology-ensure-secure-ehr-patient-data-sharing/