**Performance Evaluation and Metrics**

**Group 5: Thuan Nhan, Dallas Mitchell, Julian Phan, Tony Wright**

**EtherScan for our smart contract:  
A screenshot of a computer

Description automatically generated**

**Dataset:**

"Accessing real-life electronic health records presents significant challenges due to privacy and ethical considerations. Even when such information is available, employing real patient data in a class project, especially one where code might be shared with individuals not bound by the same data handling responsibilities, is typically not advisable. As a result, we have chosen to use an artificial dataset for our project.

The dataset selected is Synthea, an open-source software tool that creates synthetic but realistic (non-real) patient health records. Synthea skillfully simulates the medical histories of synthetic patients, generating data that closely mirrors that found in real patient electronic health records (EHRs) in terms of format, structure, and content. This includes comprehensive records that encompass each patient's entire lifespan, covering aspects such as demographics, allergies, medications, procedures, and various medical encounters including outpatient and emergency visits. For this project, we utilized the '1K Sample Synthetic Patient Records, CSV' from <https://synthea.mitre.org/downloads>.

Due to the complexity of Synthea's CSV files, which function akin to a real database with multiple tables (including medications, providers, suppliers, payers, etc.), we have streamlined our storage solution on IPFS by merging this information into a single table. This process involved joining multiple tables to focus our dataset on key elements such as patient names, health providers, insurance providers, and visit descriptions. The resultant table comprises 15 features, including 'Id', 'FIRST', 'LAST', 'HEALTHCARE\_COVERAGE', 'START', 'STOP', 'PATIENT', 'ORGANIZATION', 'PROVIDER', 'PAYER', 'DESCRIPTION', 'Insurance\_name', 'Insurance\_ownership', 'Provider Name', and 'SPECIALITY'." These features are chosen since we believe it is the bare minimum representation of EHR that represents all 3 entity which are the patient, caregiver and the insurance provider.

**Performance Metrics:**

While we can also look at IPFS performance, from our testing, the query process is essentially instantaneous and not varied too much from run to run. Thus the only necessary measurable performance metrics can be gather from smart contract performance. Due to the design of the smart contracts, since we only have 3 operations on the smart contract which is view, addPermission and revokePermission, the only real performance metric we can really look at is the transaction time and cost of each operation. Out of the 3 functions, the view function doesn’t change the state of the network thus it has no transaction fee and practiacally no transaction time. That leave us with addPermission and revokePermission.

**Results:**

**A comparison of a diagram

Description automatically generated with medium confidence**

| **Metric** | **Value** |
| --- | --- |
| Add Permission Average Time (seconds) | 26.916994 |
| Add Permission Average Cost (Ether) | 0.000005 |
| Revoke Permission Average Time (seconds) | 23.600297 |
| Revoke Permission Average Cost (Ether) | 0.000017 |

To gather the result, we constantly add and revoke the same wallet 30 times in a loop. Our result shows that in average revoke permission cost less time while having slightly higher cost compared to add permission.

When we look at costwise in the boxplot, add permission have a lot less variance from run while revoke permission cost varies a lot more. This might be due to the fact that when you revoke, you have to search for the slot to revoke the permission from thus the calculation might need slightly more overhead compared to adding where there is not much calculation needed and you only need to add at the end of the list.

For transaction time, add permission have much more variance and much higher outlier, some even get as high as 120 seconds. This might be due to the fact that when you are requesting more storage for the smart contract information, the network response would varies more while it just generally easier for network to free up space.