**Motivation: Project Hamilton**

The goal of this project was to integrate Project Hamilton with Oracle Cloud Services. This instance of Project Hamilton on Oracle Cloud processes transactions sent by users and records the transaction into blockchain tables. To achieve this goal, we had to modify our instance to write to a backend that records transactions into a blockchain table.

The Project Hamilton architecture is broken up into several servers that communicate with one another via RPCs, the sentinel, controllers, and shards. The Sentinel verifies transactions submitted to it by the user. It ensures the sends owns the input tokens along with other checks. Once verified, the sentinel passes the transaction to the Coordinator to process. The coordinator utilizes 2PC with the shards to process transactions.

Our project writes to blockchain tables within Project Hamilton to store general and compliance information about processed transactions.

![Diagram of Project Hamilton Instance on Oracle Cloud](image1)

**Achievements**

1. **Building the project with OCI** to utilize Oracle Autonomous DB
2. **Successfully Sending Data from Sentinel to DB**
3. **Blockchain Table and Its Schema Design**
4. **Expense Tracker UI and Server Development**
5. **Admin UI Demo**

**Results and Conclusions**

Based on what we’ve learned, what can we do differently?

OpenCBDC by itself is a very complex system that is created around the in-memory storage and the separate system architectures. If we were to do it again, we would likely build our own CBDC, rather than implement tools on top of an existing and specialized CBDC, as it would be a lot easier to develop with the tools we want to use in mind and not retrofit an existing project to use the tools.

**Where would we extend this project?**

1. **Continue Admin UI Development**: We constructed a framework for the Admin UI, primarily focusing on the structure without retrieving data from the database for diagram presentation. To further enhance this, it is crucial to add data filtering and sorting when executing the SELECT statement and fetching the data from the backend.
2. **Benchmarking**: We would extend the entire implementation and comparing the metrics with those found in the OpenCBDC Whitepaper would definitely be good to implement. Discussions of this task were had many times during the course of this project, and our initial plan has been laid out and documented for future work on the project.