Project Plan - Rework and Refactor the WSV storage

GOALS

Make World State View (WSV) backend modular and add new in-process WSV backend provider.

PROJECT SCOPE

HL Iroha currently uses a separate relational database (PostgreSQL) to store its state. That requires another process to start – so it takes resources and is not very efficient. This project aims to switch from a separate database to a module that would allow storing the state within the same process.

Also, the project aims to not only make the architecture neat but also on improving performance positively. The newly selected database must provide good in-memory cache with disk backed persistent state that can be reliably saved when the block is committed. Also, the database must able to hold big volumes of data which is larger than the RAM can hold.

DELIVERABLES

- ☑ Research on what key-value storage solution would work the best for storing WSV
- ☑ Approve the researched solution
- □ Refactor the code to decouple the storage and allow an easier way of changing it
- □ Execute the approved solution
- □ Research and analyse the performance affected by the decoupled storage.
- □ Create a configuration parameter to allow backward compatibility
- □ Documentation
- □ Testing

MILESTONES

- ☑ 20 Jun 2020 Complete the Research of finding the best storage solution and also get it approved
- ☑ 13 Sep 2020 Implement RocksDB block Storage.
- □ 30 Sep 2020 Write tests once Implemented solution is approved.
- □ 12 Oct 2020 Code Refactoring and Performace Analysis
- □ 30 Oct 2020 Code Review and Testing
- □ 07 Nov 2020 Documentation and discuss the future plans

EVALUATION CRITERIA

1. Refactored code allowing connection to different storage solutions
2. New storage solution embedded into the program itself
3. Analysis of the performance before and after the change