HDBPP Timescale Progress

Project Summary

Completed

Evaluation Project – initial testing and investigation.

New TimescaleDb Insertion Library – libhdbpp-timescale

Cluster Development and Deployment – 3 node cluster running at ESRF

TTL Service – deployed via Docker

Health Check Device Server – depends on Health Reporting Service

Reorder Chunks service – deployed via Docker

Health Reporting Service – deployed via Docker

Project Consolidation – see hdbpp-timescale-project, unified build

Decimation requirements

TimescaleDb Features Evaluation (Compression/Aggregation)

Web Reporting Prototype - depends on Health Reporting Service

Aggregate Views (Scalar/Array) – 12 month summary views

Current

Libhbdpp API Update – Ready for merge

Enum Support (libhbdpp-timescale) – Ready for merge

Batch Event Insertion (libhbdpp-timescale) – Ready for merge

ES/CM to libhbdpp-timescale Direct Linking – Ready for merge

User Python Data API – In progress (will be completed later)

Future

Decimation – Plotting

Decimation - Control

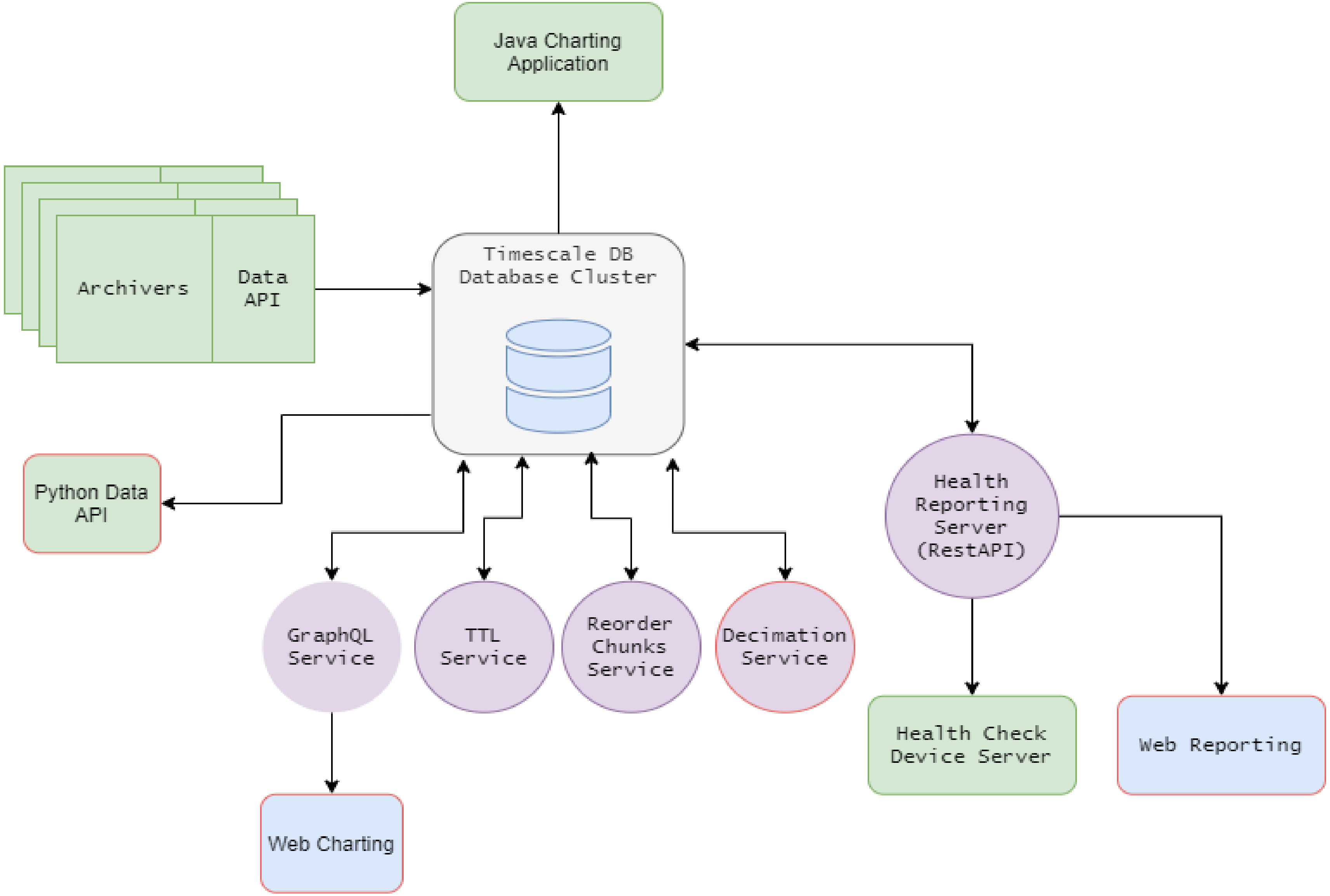
Cassandra Data Import

Table Compression Deployment

Experimental

Prototype charting solution

Component Overview



libhdbpp/es/cm Changes Summary

libhbdpp

- Renamed API functions
- Support for batching events + feature support function.
- Clang integration and usage
- Updated cmake build
- Updated docs
- Moved public headers to /include/hdb++
- Pass by reference, not value
- Hdbpp Namespace
- Split into class based files
- Derive HdbClient from AbstractDB to enforce API
- Moved HdbCmdData to hdbpp_es

es/cm

- Clang integration and usage
- New cmake build systems
- Improved docs
- Support changes to libhbdpp
- Moved add attribute functionality from CM to ES, no need to link to libhbdpp now (test system possible without CM).
- Modernise code and remove warnings.

Questions

- I will support the project for the next few months.
- I can be contacted at chedburgh@gmail.com for issues with the project.