SAND2019-10881C

Citadel Data Frames: An Architecture for Managing Legacy Data

Stephen Jackson

Citadel is a custom code framework designed to quickly roll out multiple data storage applications at Sandia National Laboratories. Citadel helps projects by providing features common to all data system applications. These include: enabling CRUD operations, tracking changes to the data (via a user accessible provenance database), parsing common data types into Parquet-Avro files, opening RESTful API's for third party tools to interact with the data (see posters on WASP), and establishing custom access controls to data on a granular level.

Currently the two implementations of Citadel are the SEDS and DataSEA projects. SEDS focuses on replacing a large legacy data system. It has hard-coded data structures and rules for accessing data. DataSEA focuses on quickly rolling out smaller data systems that allow the users to determine their own data structures and custom rules for storing, accessing, and manipulating data.

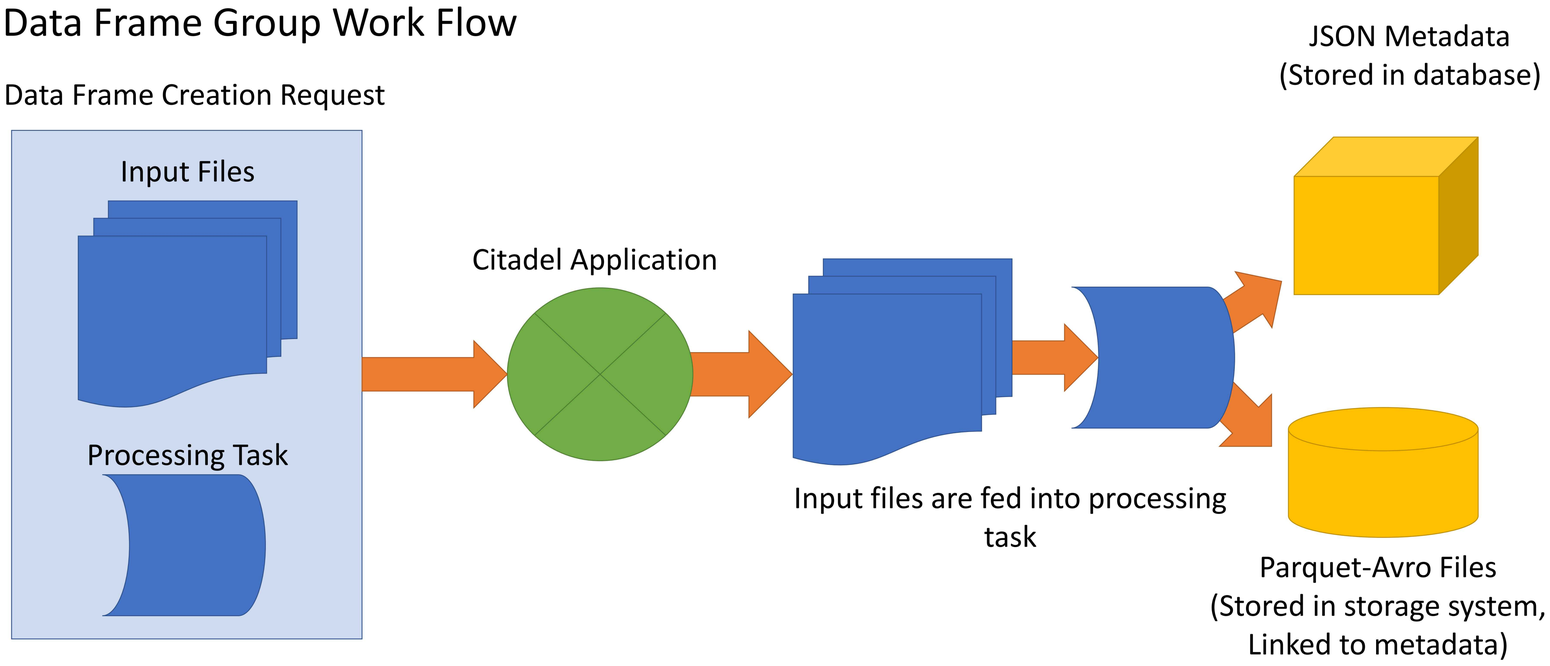
Citadel based data systems needs to ingest data from a variety of legacy sources. Legacy data sources are typically mixed bundles of metadata or data in problematic formats. For example:

- Metadata/Data was stored in one-off formats that were not easily parse-able.
- Binary blobs grouped many channels of timeseries data into one file, meaning users had to download lots of wasted data.
- Data was stored in ASCII meaning long, slow downloads for some datasets.

Our Data Frames architecture:

- Is compatible with modern tools like Python and MATLAB
- Uses binary formats for efficiently transferring data.
- Enables users to only pull data relevant to them.

Data Frame Group Work Flow

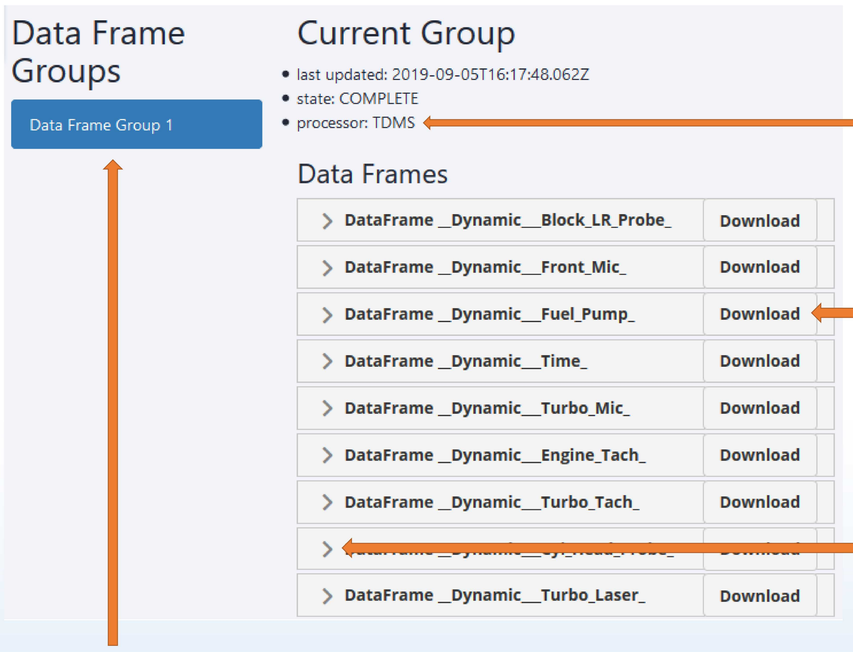


Customers and developers can create processing tasks, isolated units managed by the Citadel application that responsible for translating a customer data format into Parquet-Avro. The Citadel application sandboxes the processing task, allowing it to run with the specific set of dependencies needed for the data.

Processing tasks produce Data Frame Groups, which are a collection of Data Frames. Each Data Frame is a table of data. For data sources capturing many channels of data from an experiment, each channel can be its own Data Frame.

The system records the exact task used to process a set of input data. When coupled with good code management processes allows exact knowledge of what code was used to transform data.

Data Frames are processed asynchronously, meaning they can be offloaded to a collection of worker nodes. Users can watch for their processing task to be finished in the user interface:

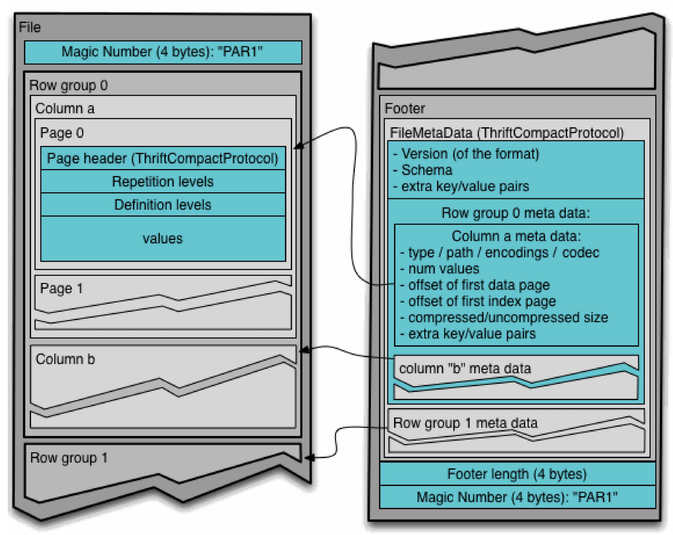


The processor used to extract data from the input files.

The user can download individual frames of data (typically channels)

Users can expand these panels to view metadata for the channel.

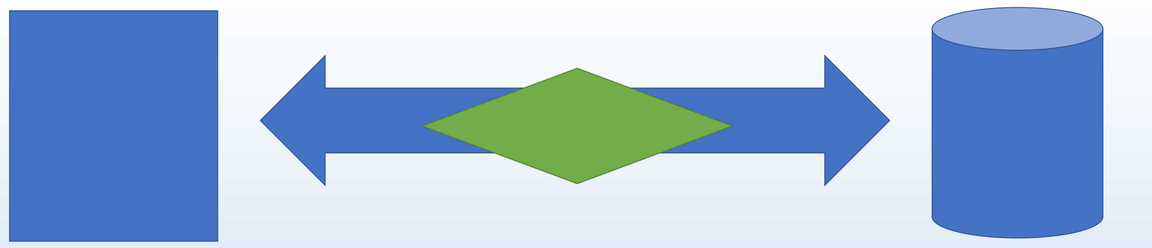
Data Frame Groups are a collection of Data Frames. Data Frame Groups are attached to records



Parquet is columnar data storage format. It stores tables of data, but stores them on disk by column (rather than by row). By storing them by column, many operations on series of data are much more efficient.

Parquet is a binary format, meaning it can efficiently store numeric values.

We use Parquet-Avro, because Avro is a self-describing format for the contents of the Parquet file, making it self contained.



Parquet has its own representation it stores in disk

Object model converters transform parquet data into a user friendly format

We use Avro, because Avro is self-describing.

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. _____