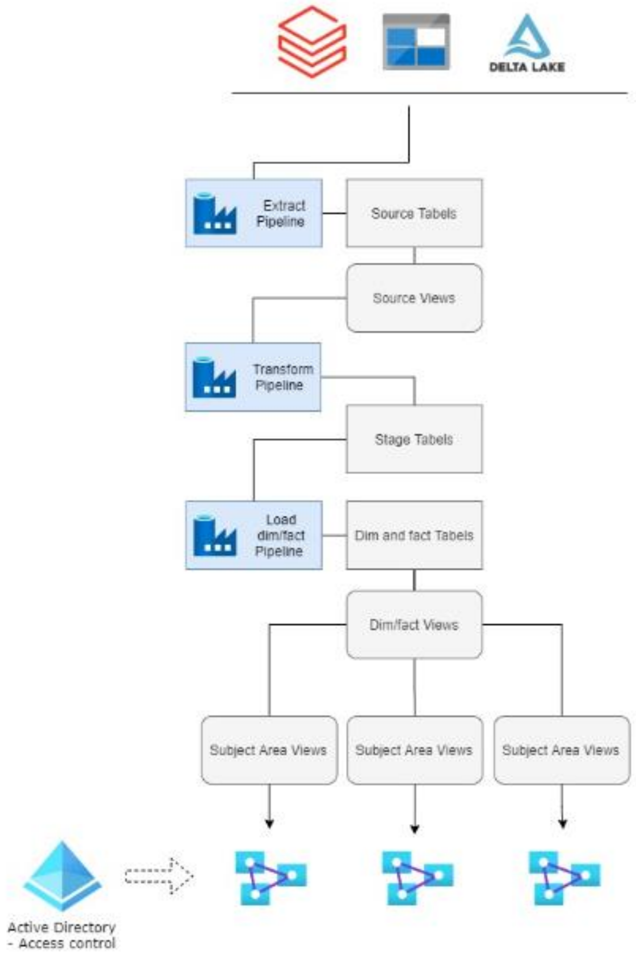
Principles of NuuData

NuuData

NuuData is a data warehouse solution that helps the business manage and analyze large volumes of data from multiple sources, to increase the level of informed decision making.



Extract

Source Tables

- Data is always pulled from Deltalake (NuuDL) into NuuData.
- NuuDL stores history hence NuuData will inherit history from the Datalake.
- No logic is added to the extract tables.

Source Views

- Auto generated by framework
- Is used for standard filtering, e.g., IsCurrent = 1
- Can be used to implement simple modifications to the source, i.e., renaming and format changes.

Transform

Stage Tables

- Used to stage the dimensional model with the correct keys between dimensions and facts.
- Framework generated SP, but logic done by developer.
- Tables are truncated in each run.
- Entity and attribute names must comply with the guideline since it is crucial for the framework to work as intended, e.g., avoid reserved keywords like %ID etc.

Load

Dim/fact/bridge Tables

- Auto generated by framework
- No modifications are done here – all logic is placed in the transform layer

Dim/fact/bridge Views

- Auto generated by framework
- No modifications are done here.

Principles of NuuData

Subject Area/Mart

The intension is to publish data for analytics data usages. Data is published in a unified reusable format customized to the analytics usage. Data are organised according to the area of business usage – hence ensuring an ease of GDPR compliance and access control to data.

The characteristics of the Mart Layer

- Data is organised based on Subject Areas specified by the business.
- To gain easy overview of which data is used in each solution.
- Consists of Dimensional Models based on dim/fact/bridge views.
- Follow NuuData naming convention.
- Changes to logic can be applied to meet the specific business requirements.

Access

Access data in NuuData

- Azure Active Directory User Groups are added to Analysis Service tabular models.

Our Framework

The intension by using the Best Practice Framework (BPF) is to generate standardized, simplified and less error prone solutions.

We use the Framework to automate creation of Azure Data Factory and SQL Server objects through PowerShell scripts.
The objective of the framework is to simplify, standardize and automate otherwise tedious and error prone work for the developer. Furthermore, the meta data, which is created in the development process, shall ensure that additional extensions to the solutions can be easily incorporated in future.

Modification to objects that are created and maintained by the framework should not be manually modified afterwards.