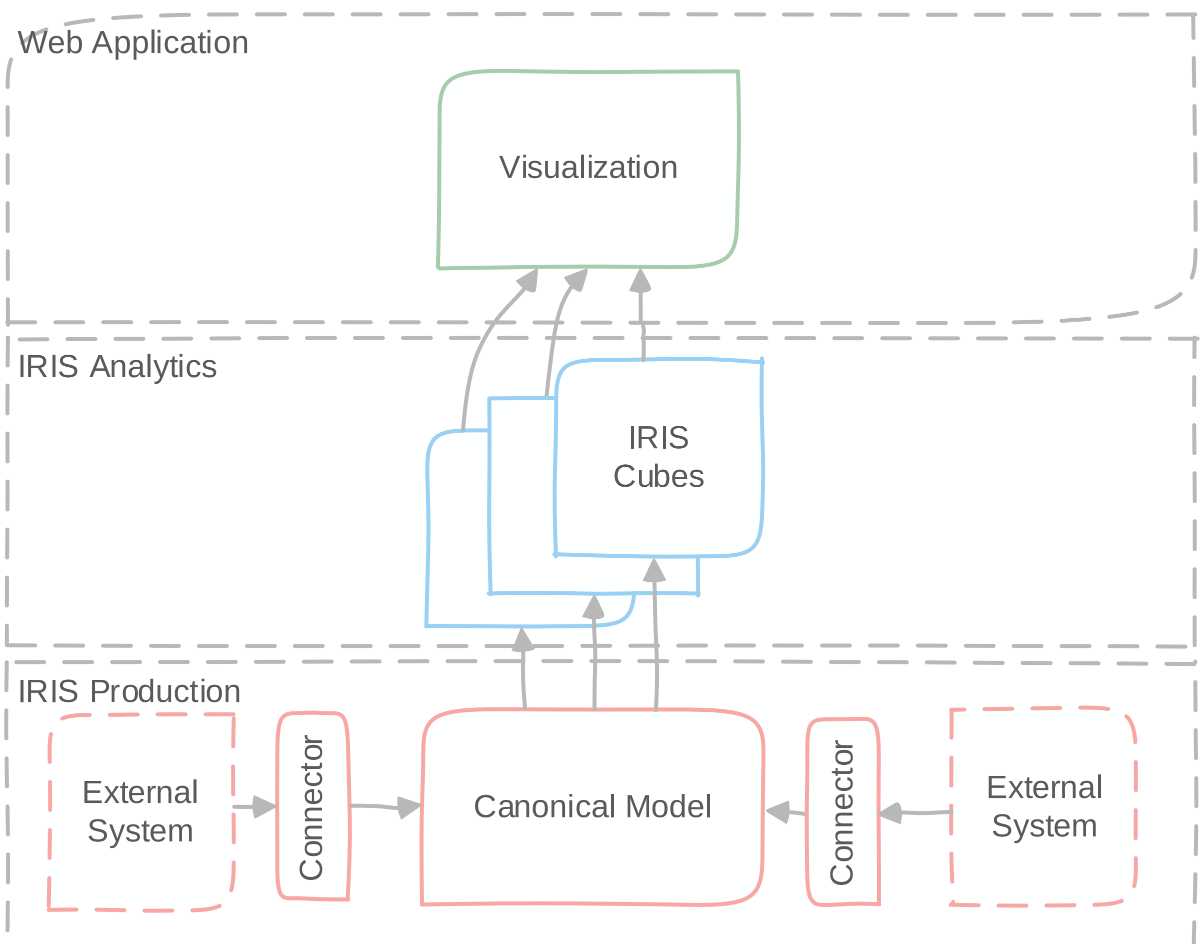
ASP Mining - Analytics Description

1 INTRODUCTION

The scope of this document is to describe the logical steps and behaviour of the data processing behind the sample ASP Mining Starter pack and the included components, it's not a source code documentation (which is available in other documents and the source code itself), but a higher level technical description of what the software does.

The ASP Mining Starter Pack has 3 layers:

1. IRIS Production: integrates mining systems data and transform it into IRIS Mining Canonical Model.
2. IRIS Analytics: performs calculations on top of the Canonical Model to generate higher value KPIs.
3. Visualization: using IRIS as a backend, implements a web-based application using standard web tools to navigate and visualize the IRIS Analytics results.

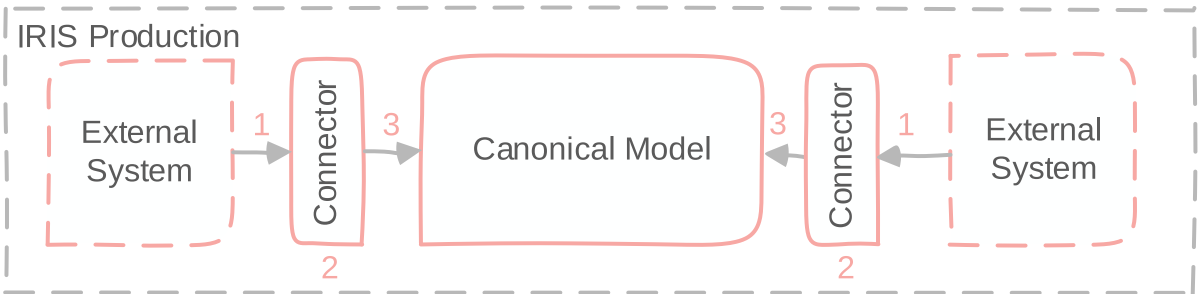


2 IRIS Production: Connectors and Data Flow

The IRIS production consists of one or more Connectors that follow the pattern:

1. External System Connection
2. Service for data collection and transformation to Canonical Model
3. Operation for storage of new data into Canonical Model

The connectors also include a Task to keep updated the relevant cubes that are used with them.



One of the main tasks in the data transformations performed by the included connectors and any other connector, is the transformation **of the Timestamps** in the data source systems, which are not always standard types, but local timestamps relatives to a shift or to other non-standard DB time datatypes.

Since the Canonical Model uses only IRIS TimeDate types, this step is crucial for being able to navigate across different sections of the model with consistency and allow proper “drill through” in the cubes.

2.1 MineCare Connector

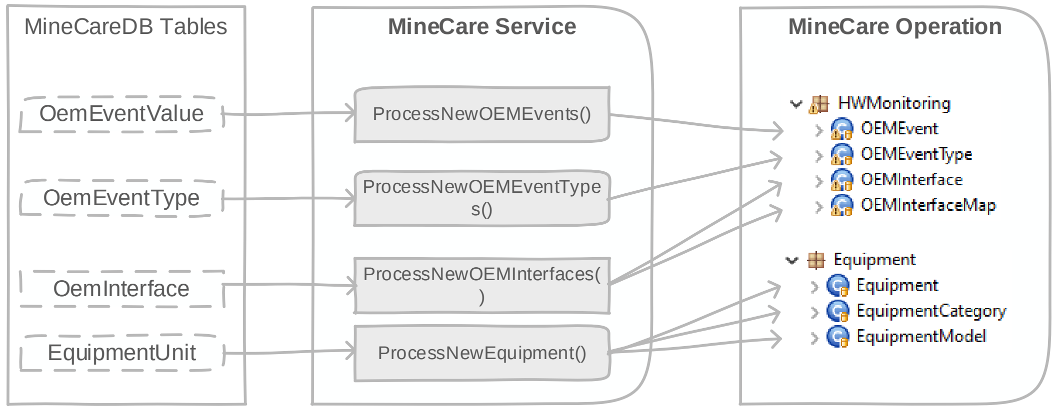
2.1.1 External System Connection

The External System connection of this connector if done using a JDBC connection to the MineCare database which runs on SQL Server (v2019+), the connector uses a standard IRIS SQL Inbound Adapter to perform this connection.

2.1.2 Service for data Collection and transformation to Canonical Model

The Service of the MineCare connector brings information about multiple health signs of the mining equipment specified by the client into the MineCare software. This data is mainly used to track sensor related events such as: overspeed, valve pressures out of range, fuel levels, temperature of internal components, and many others.

Is important to note, that that the raw sensor data is not being collected by the connector, but the events that this data generates (i.e: one or more sensor values are out of a determined range), is the data that is being collected and saved in IRIS.



Source Table	Data obtained	Data Usage
OemEventValue	Numeric/Text value of an event	Analysis and Plotting
OemEventType	Type or category of the event	Filtering and categorization
OemInterface	Phyisical interface that made the reading in the truck	Link between Sensor and Equipment
EquipmentUnit	Equipment metadata (capacity, category, model)	Equipment Information

2.1.3 Storage of new data into Canonical Model data

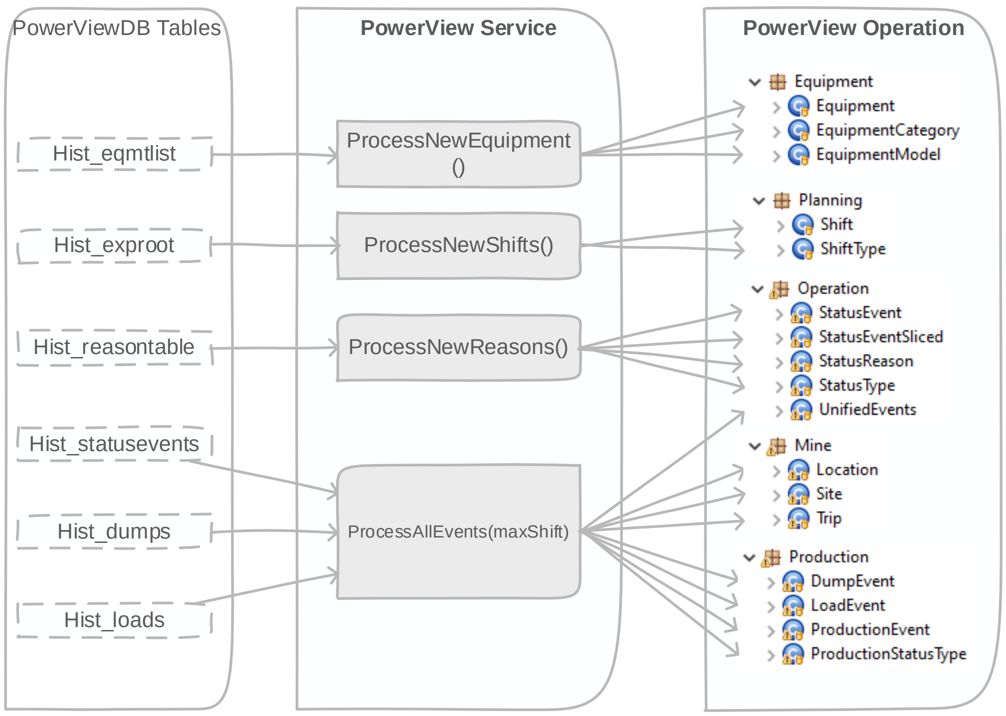
The Operation in the MineCare connector saves data to the following Canonical Model Classes

Equipment.Equipment
Equipment.EquipmentCategory
Equipment.EquipmentModel
HWMonitoring.OEMEvent
HWMonitoring.OEMEventType
HWMonitoring.OEMInterface
HWMonitoring.OEMInterfaceMap

2.2 PowerView Connector

2.2.1 External System Connection

The External System connection of this connector if done using a JDBC connection to the PowerView database which runs on SQL Server (v2019+), the connector uses a standard IRIS SQL Inbound Adapter to perform this connection.



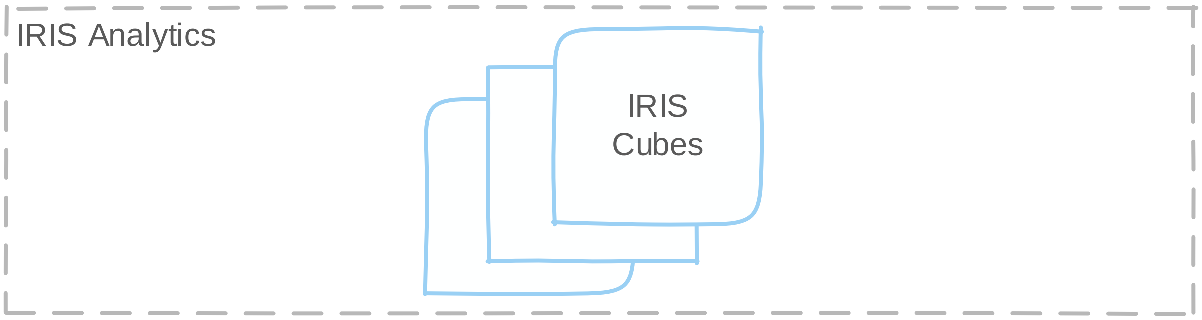
Source Table	Data obtained	Data Usage
Hist_eqmtlist	Equipment metadata (capacity, category, model)	Equipment Information
Hist_exproot	Shifts information	Shift Type and Start Time
Hist_reasontable	Reasons for Status Event changes	Details about events
Hist_statusevents	Status events history for every equipment	Events Information
Hist_dumps	History of material Dumps	
Hist_loads	History of material Loads	

2.3 Canonical Model

The canonical model details can be found in the Canonical Data Model.pdf file.

3 IRIS Analytics: DeepSee Cubes

The DeepSee cubes consolidate raw data and processed data from the Canonical Model, making it available to be accessed from the FrontEnd to the end users. These cubes are updated continuously by Scheduled Tasks.



The cubes provided by the Starter Pack are:

Cube Name	Contents	Usage
Equipment Cube	Equipment Information	General Information
OEMEvents Cube	OEM Events	Hardware Maintenance KPIs
Production Events Cube	Material Dump and Loads	Production KPIs
Status Event Cube	Equipment State Changes only	State changes trazability
Status Events Sliced Cube	Equipmente State changes with interpolation over time	State changes KPIs (daily analysis)
Unified Events Cube	Production Events combined with Status Events	OEE and OEE related KPIs: Utilization, Capacity Performance, Time Performance

3.1 Unified Events Cube

The “Unified Events Cube” is based in the Unified Events Class, which could be considered as the Central class of the model, from where you can reach any element. This cube joins Status Events (e.g: operative, delay, etc.) with Production Events (i.e. dumps and loads of material).

3.2 OEE CALCULATION

The Overall Equipment Efficiency KPI is a very known KPI used in many industries, since it indicates how efficiently the process equipment is being used.

For the ASP Mining starterpack, a the OEE for the Open Pit Mine is calculated like:

$$OEE = Utilization\ x\ Capacity\ Performance\ x\ Cycle\ Time\ Performance$$

Where:

- Utilization: Percentage of time that the equipment was operative
- Capacity Performance: How many Tons was the equipment loaded with vs its nominal capacity.
- Cycle Time Performance: How fast the Load/Dump cycle is being completed for a given source/destination location vs previous same trips.

These 3 KPIs are conveniently precalculated in the UnifiedEventsCube to make it easier for the frontend to access the results. If any of the performance KPIs are not possible to calculate for a given type of equipment in the fleet, it could be considered as a 1 to don't impact the KPI calculation.

Since the highest impact in the mine production depends on the transport equipment (trucks) we will focus the analysis on them.

3.2.1 Utilization

The equipment Utilization is calculated based on the time the equipment stays in state Operative vs other states. This state is defined in the StatusType Class of the canonical model. To calculate the total utilization, the StatusEventsSlicedCube and the UnifiedEventsCube can be used.

For more detail in the analysis the StatusEventsSlicedCube is preferred, since this cube provides 5 minute windows with all the details related to StatusEvents, where the time of all states inside this window (Operative, Delay, etc.) is calculated as PartialTime, allowing fast calculation and comparison of time durations for any given time range.

$$Utilization = \frac{Operative\ Time}{Total\ Time}$$

In the cube, the time is calculated like:

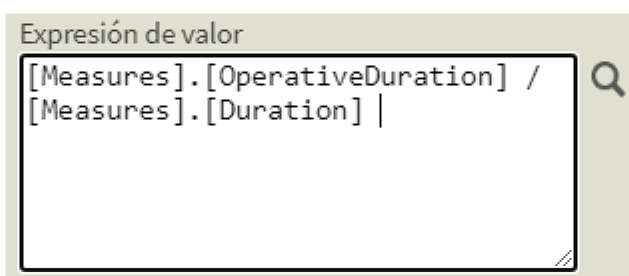


Figure 1. Cube implementation of Utilization

3.2.2 Capacity Performance

The Capacity Performance KPI indicates how many tons of material is being loaded to a truck vs its maximum capacity.

$$Capacity\ Performance = \frac{Measured\ Tons}{Truck\ Capacity}$$

The cube implementation looks like:

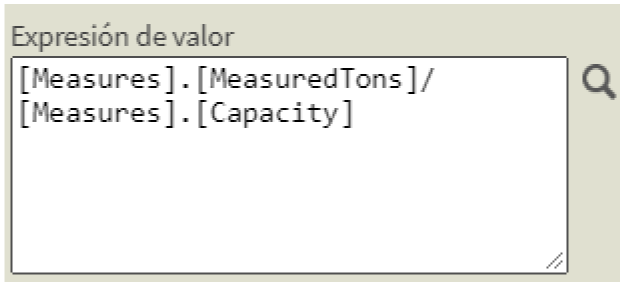


Figure 2. Cube implementation of Capacity Performance

3.2.3 Time Performance

A **Trip** is a set of a **Origin Location** and a **Destination Location** within the mine. The mining Cycle in the open pit mine consists of 2 trips: the first one is the truck from a loading site to a dumping site, and the second one is the return of the truck to a loading site.

The **Time Performance** KPI is calculated for each trip a truck performs. A **Trip** is a unique set of **Origin Location** and a **Destination Location** within the mine. The class *ASPMining.CanonicalModel.Mine.Trip* holds the data for all possible trips and keeps the statistics for it.

The “Trip Time” is the time a truck took to complete one trip while in “Operative State”. The “Trip Reference Time” is read from the **Trip** table.

$$Time\ Performance = \frac{Trip\ Time}{Trip\ Reference\ Time}$$

The cube implementation has a cap of 100% for this value to avoid errors created by changes in the destination of the truck. The cube measure looks like:

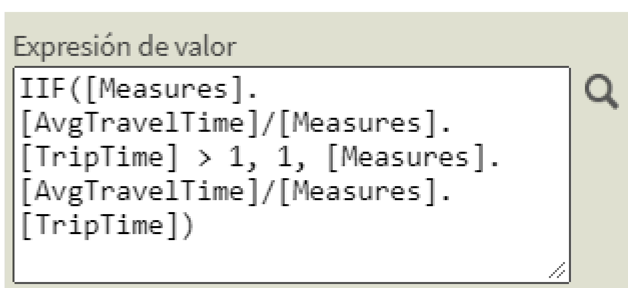


Figure 3. Cube Implementation of Time Performance

4 VISUALIZATION: WEB APP

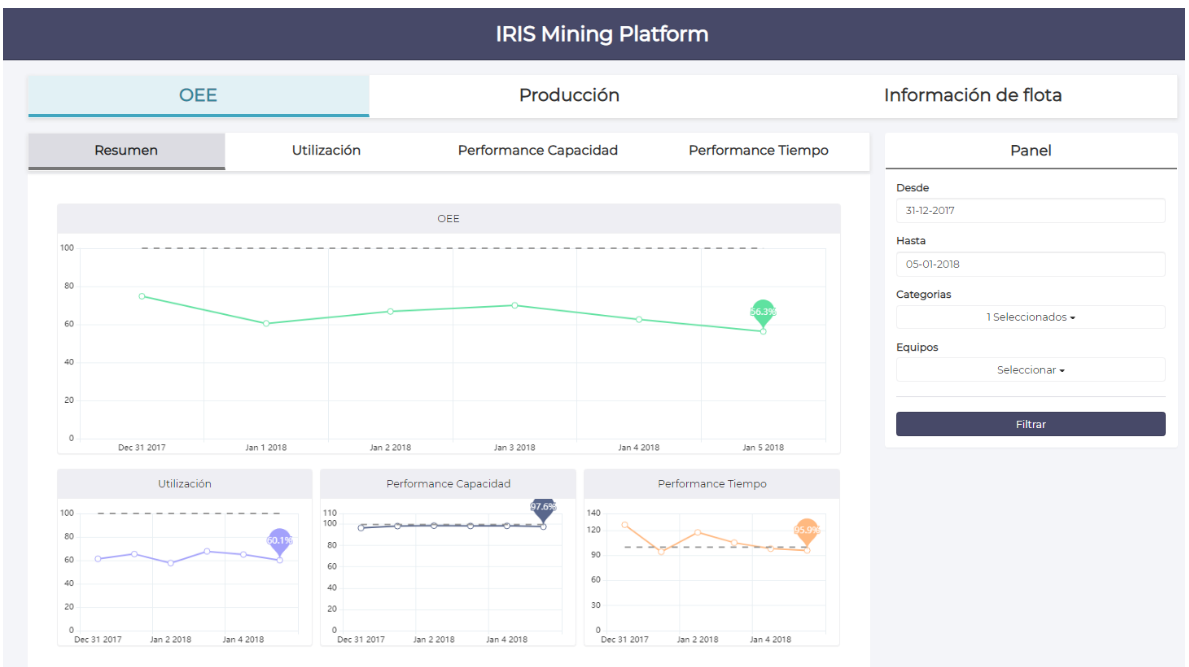
The Visualization WebApp provided, is a samples application based on AngularJS, currently using 2 libraries for graphics: Chart.js and ECharts.js, and implements multilanguage support with angular-translate library.

The JavaScript code performs queries to the IRIS cubes using MDX queries through the MDX2JSON libraries.

The sample project tree has the following structure:

- OEE
  - Summary
  - OEE
  - Utilization
  - Capacity Performance
  - Time Performance
- Production (production cycle information for trucks)
- Fleet Information

Example of the visualizations made on JavaScript showing the resulting graphics of the OEE.



The reference project could be found inside the folder FrontEnd on the root of the ASP Mining project.