TARGET SOLUTION PROPOSAL

How to ingest data with DP from Data SuperMarket UI

This series of slide represent a proposal on how to ingest data into Kafka using the UI, at the moment the UI has a slightly different user experience, to implement this flow some work will be required.

The main difference here respect the current implementation are the following:

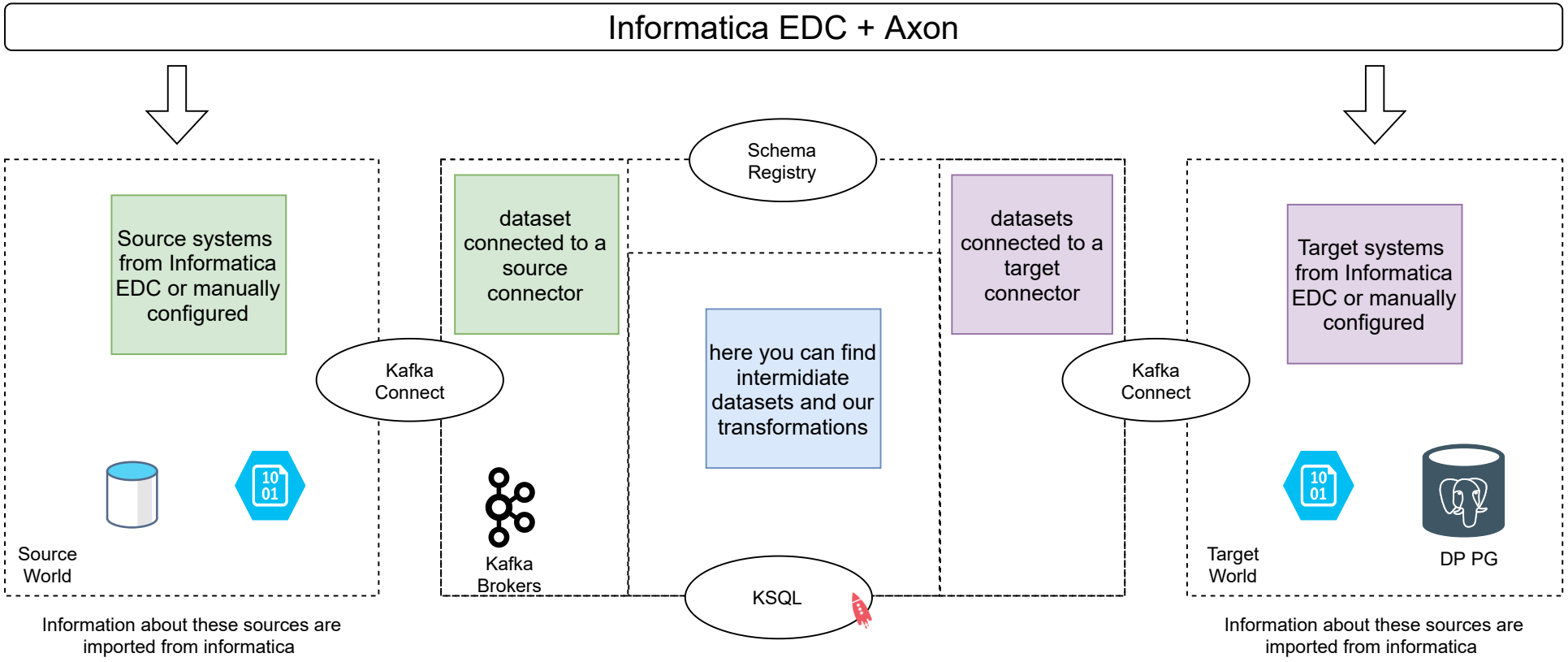
- Data avaialable in Kafka is represented as a dataset object. Dataset may be connected to: source connector, target connector or transformation (e.g. intermidiate result)

- Everything is deployed together (dataset with their related connector and transformation)

This pattern is based on the Azure Data Factory user experience, you can find how we would be able to implement use cases in this solution, with some video example captured from Azure DF.

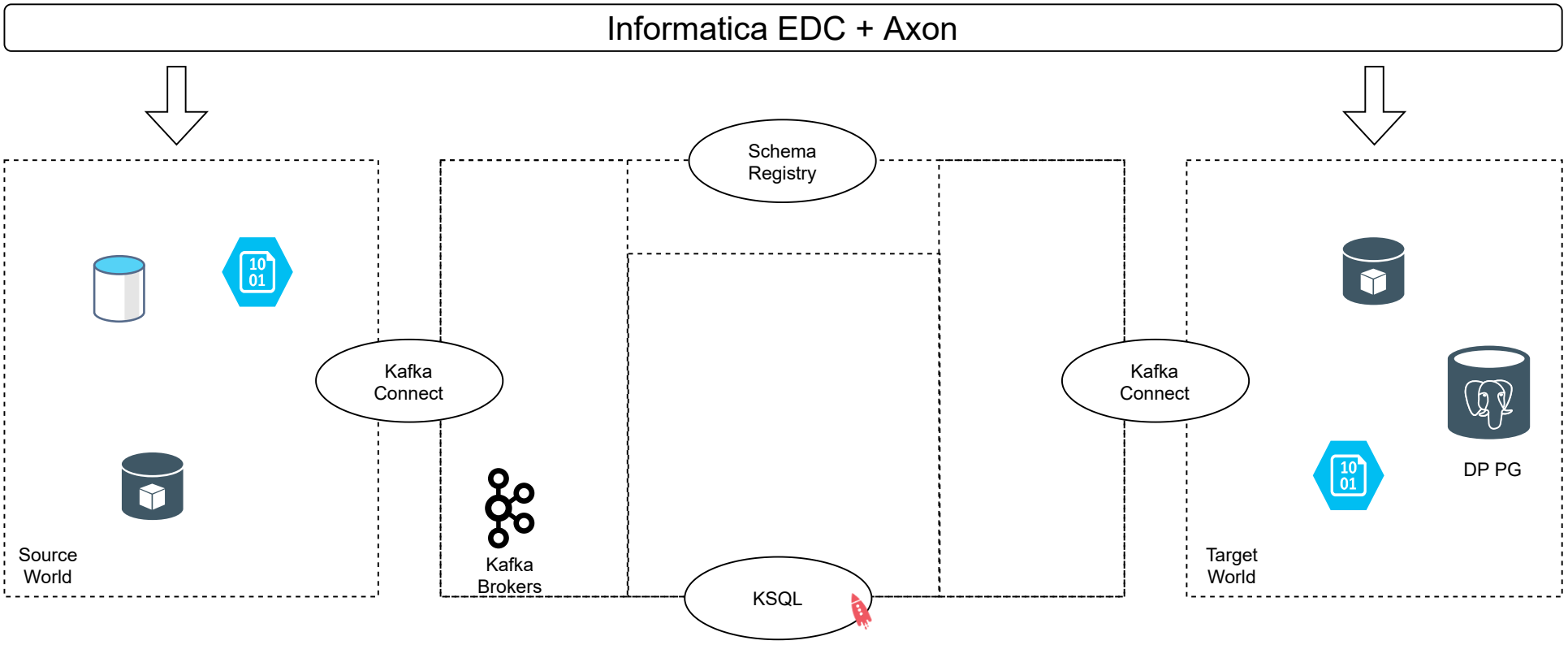
Ingestion Architecture Layout

Before jumping into the STEPS I would like to describe the canvas where we are going to place all objects



STEP 0:

Source and target table metainformation are imported from informatica

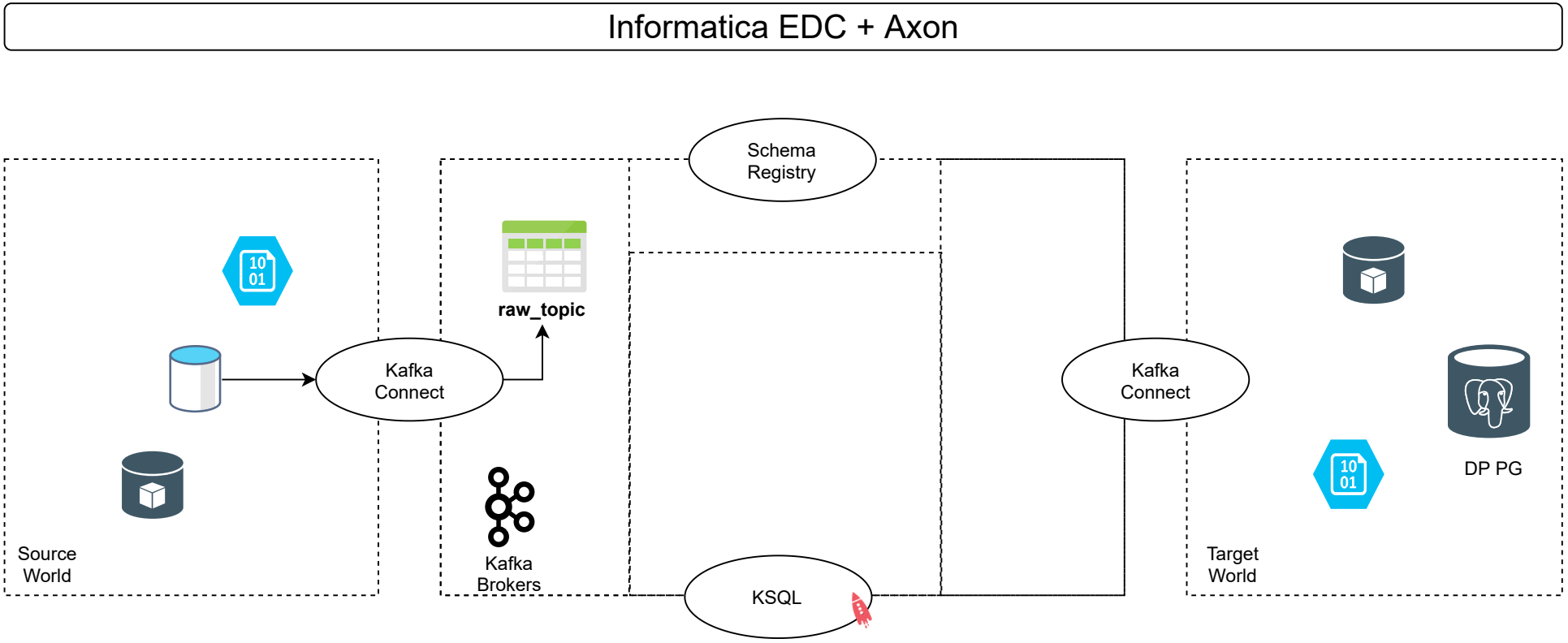


OPEN POINTS:

1) Import from informatica N.A. yet

STEP 1:

User define a source connector and its properties which results into a source dataset (raw_topic)



OPEN POINTS:

1) What is the appropriate naming for raw_topic?

3) Connector creation is not clear at all in current super market UI

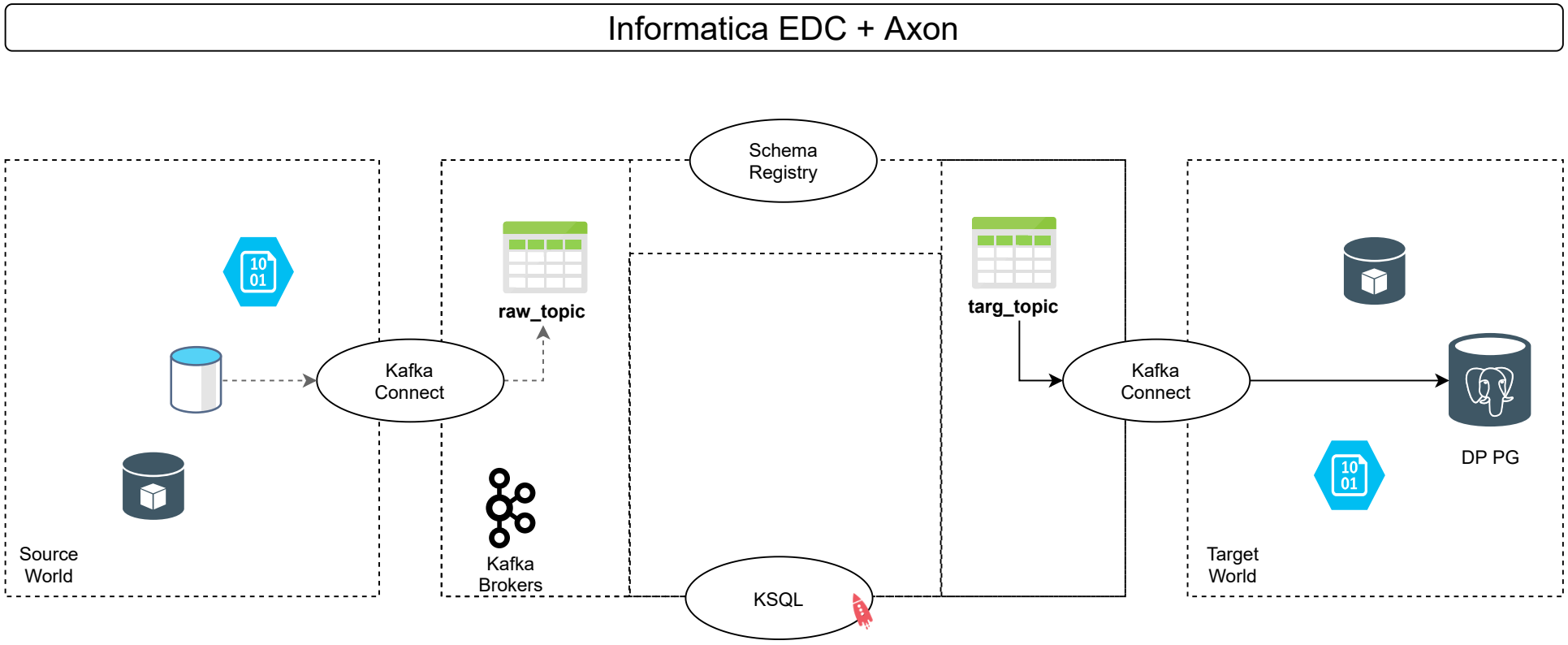
NOTES:

1) SInce we are creating the source dataset together with the connector, we can choose whatever naming we want (we are the right context)

VIDEO EXAMPLE FROM DATAFACTORY https://drive.google.com/file/d/1LG5mi3Tao_-RQj4MEYxG4L2_BHl0xYHq/view?usp=sharing

STEP 2:

User define a target connector and its properties which results into a target dataset (targ_topic)



OPEN POINTS:

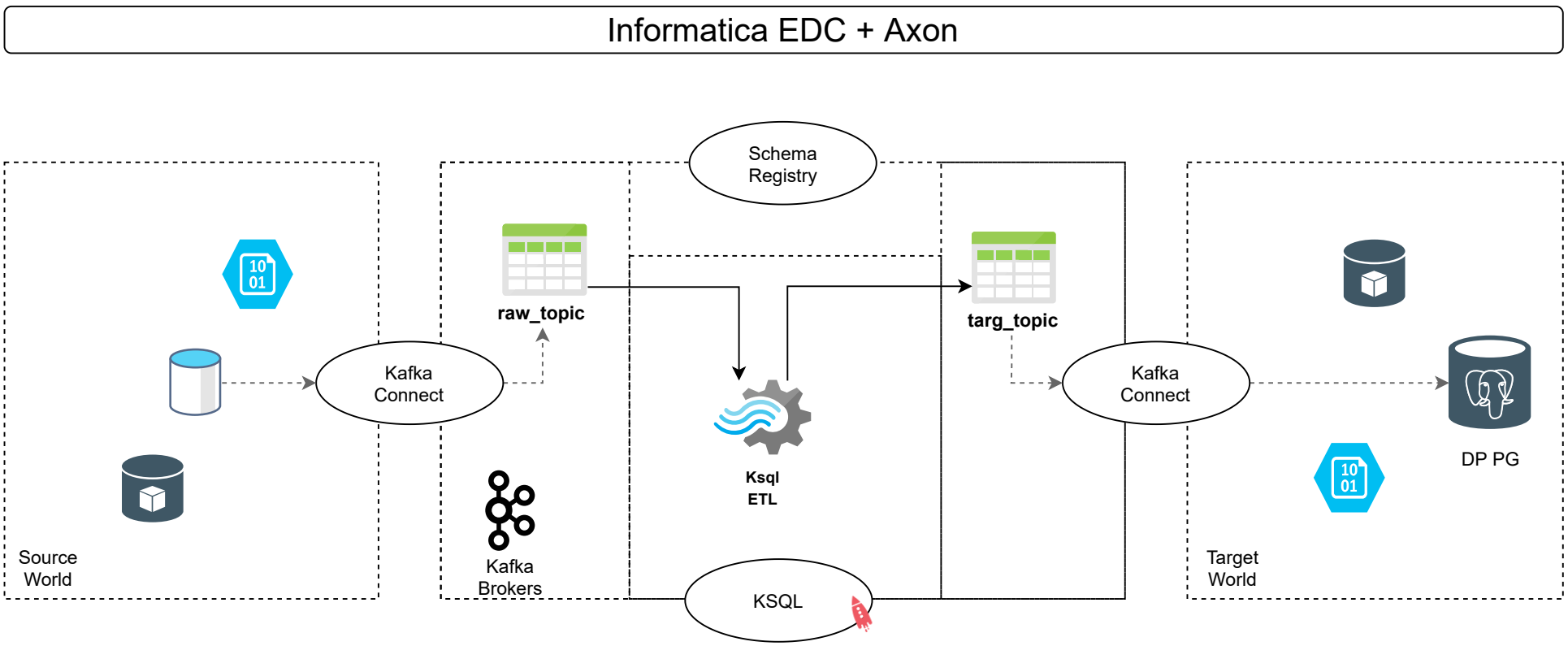
1) Target connector creation not available yet in UI

2) What is the appropriate naming for targ_topic

VIDEO EXAMPLE FROM DATAFACTORY https://drive.google.com/file/d/1X8lKuhfhOml_JYs85UhuBa9FZJD_jLUG/view?usp=sharing

STEP 3:

User defines a trasnsformation from raw_topic to target_topic



OPEN POINTS:

1) The current UI canvas represent it in a different way (only transformation is displayed)

2) At the moment there are no rule for transformation, we may even be able to create a transformation writing into the source topic

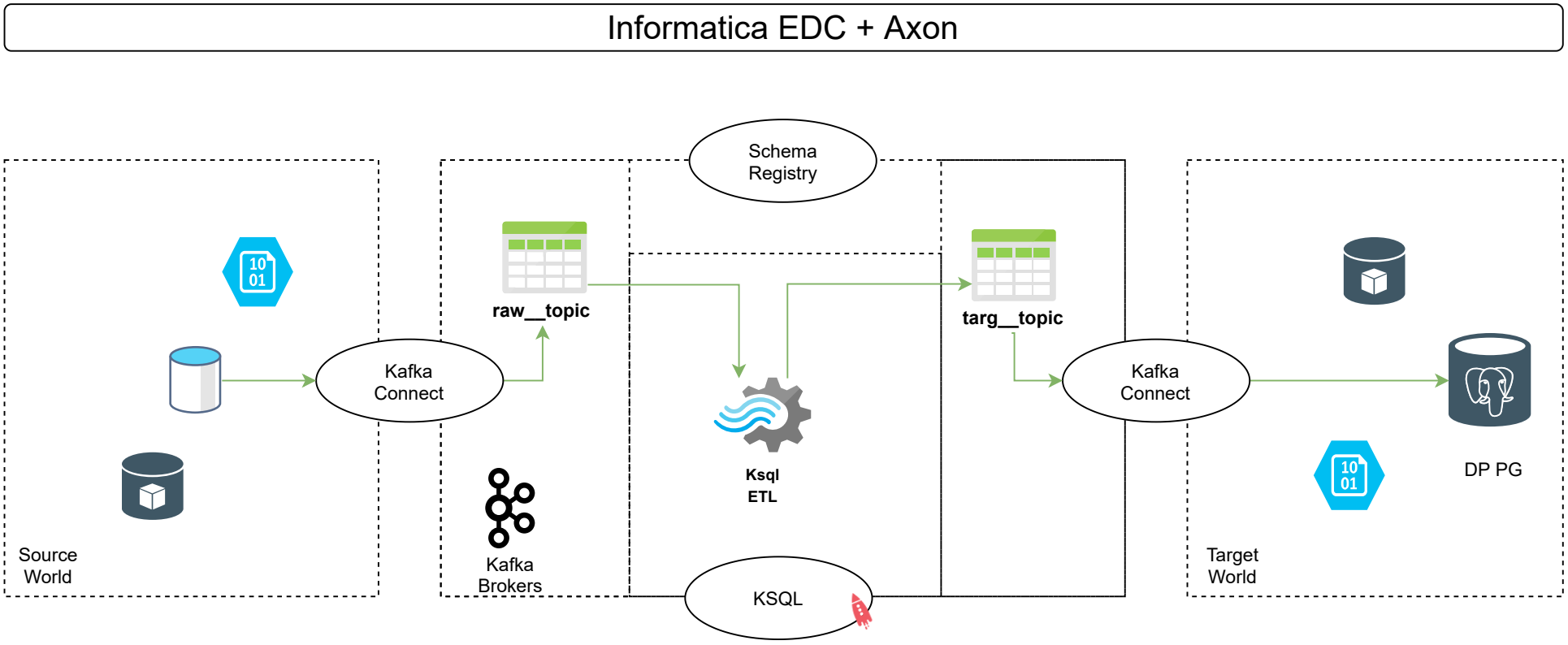
NOTES:

1) Ksql ETL element shoul be a subclass of a generic box, another box would CDC or DQ or 2 Ksql ETL box in case of intermidiate table (see next Transformation section)

VIDEO EXAMPLE FROM DATAFACTORY https://drive.google.com/file/d/1srnwSBqsFA54L6Rby6CGX8SPM-WDku2W/view?usp=sharing

STEP 4:

User can save the data flow and deploy in the related environment



OPEN POINTS:

1) No way to represent the entire data-flow canvas in the same object atm, we can only represent a transformation we will need an obect wrapping everything together

ETL BOXES COMPONENTS

How we can concatenate different transformation in the UI canvas

Here I would like to explain the scability of this system with the creation of different ETL boxes.

A ETL box, is represented with an icon in the canvas, it has input and outputs datasets. We can concatenate multiple ETL boxes with the help of intermidiate datasets

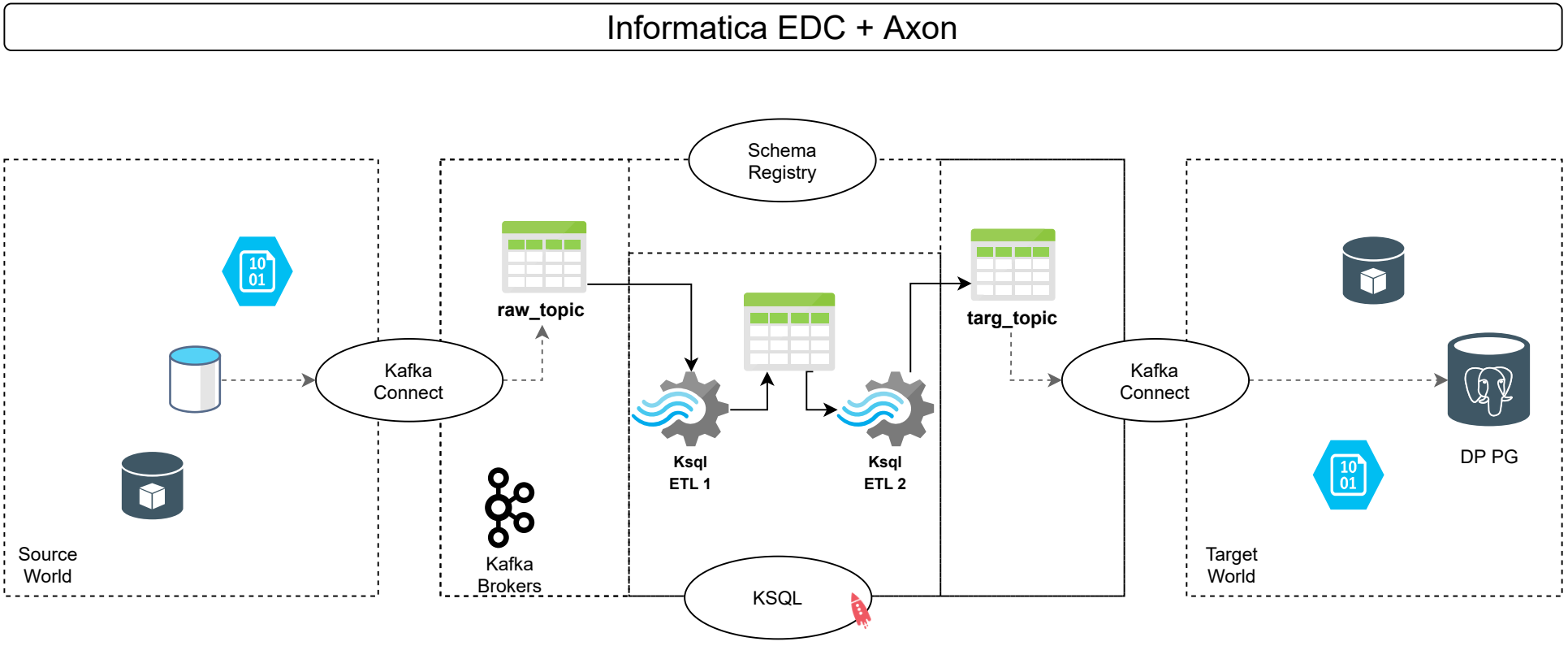
Currently we only support KSQL transformation has a ETL box, in the future we may think about introducing:

1) CDC: this box takes as input one dataset only and outputs a new dataset with only updated or new row

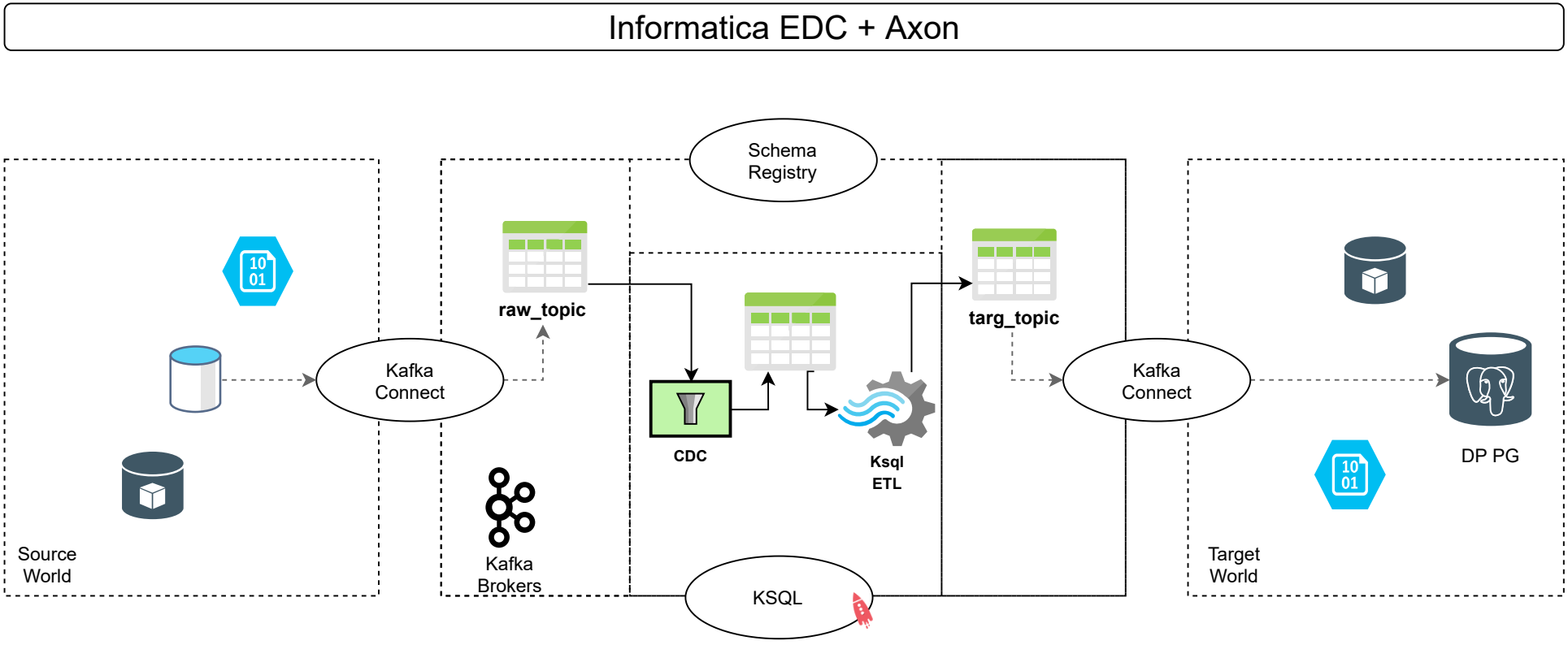
2) Spark ETL box: this box takes as input one or more dataset and as outputs one or more datasets

3) .... whataver you want e.g. containers batch job, ...

Multiple KSQL transformation one after the other



Custom transformation boxes like CDC



Informatica EDC + Axon