

WIR SCHAFFEN WISSEN - HEUTE FÜR MORGEN



Nikhil Biyani :: Experiment Control Software Developer :: Paul Scherrer Institut

File Writing in NICOS

7th ECP Workshop | 04th September 2018

Concept

Data Sink

NICOS Device
created only once

Processes a piece of data

Write to file
Display to user
Draw plots
...

Data Sink Handler

Python Object
created as required

Actually does the job!

prepare()
begin()
putMetainfo()
putValues()
putResults()
end()

Dataset

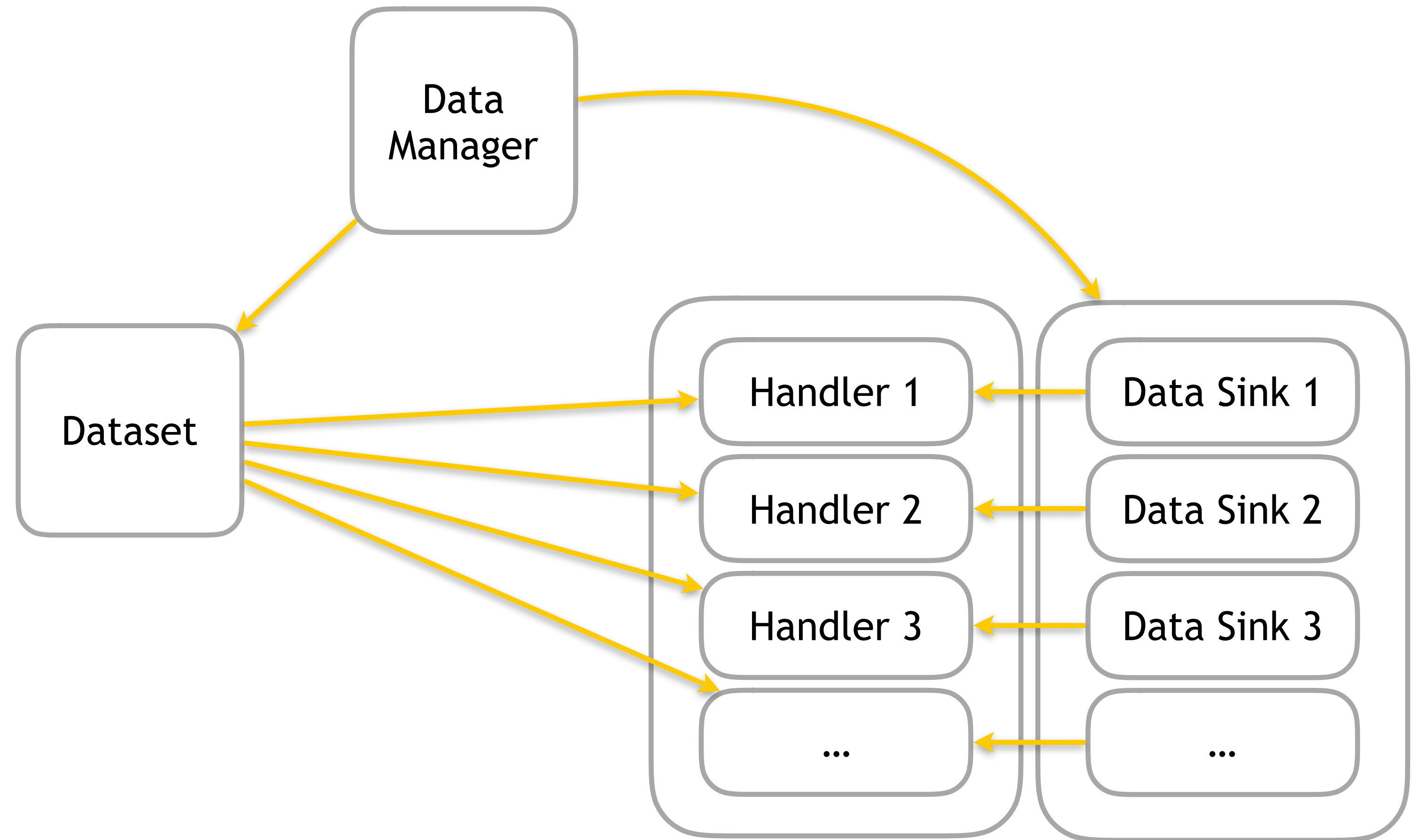
Python Object
created for each measurement

Collects data of measurement

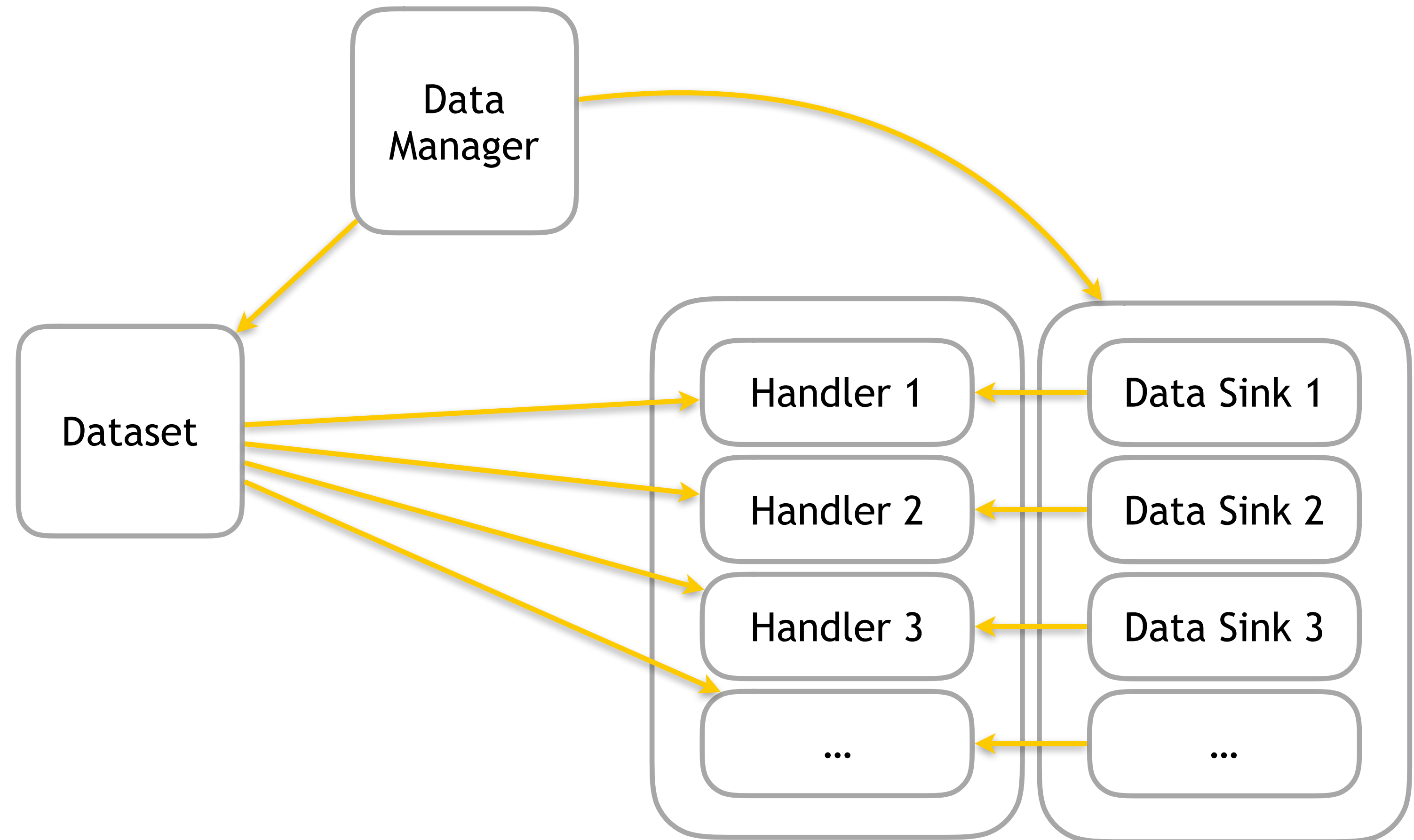
uid
counter
filename
values
metainfo
...

- Has a list of all configured sinks
- Creates a new Dataset object for each measurement
- Creates a new handler class for all configured sinks and assigns them the created dataset
- Responsible to assign counter, filename to the created dataset
- Assigns device values and metadata to the dataset
- Provides the dataset with detector data

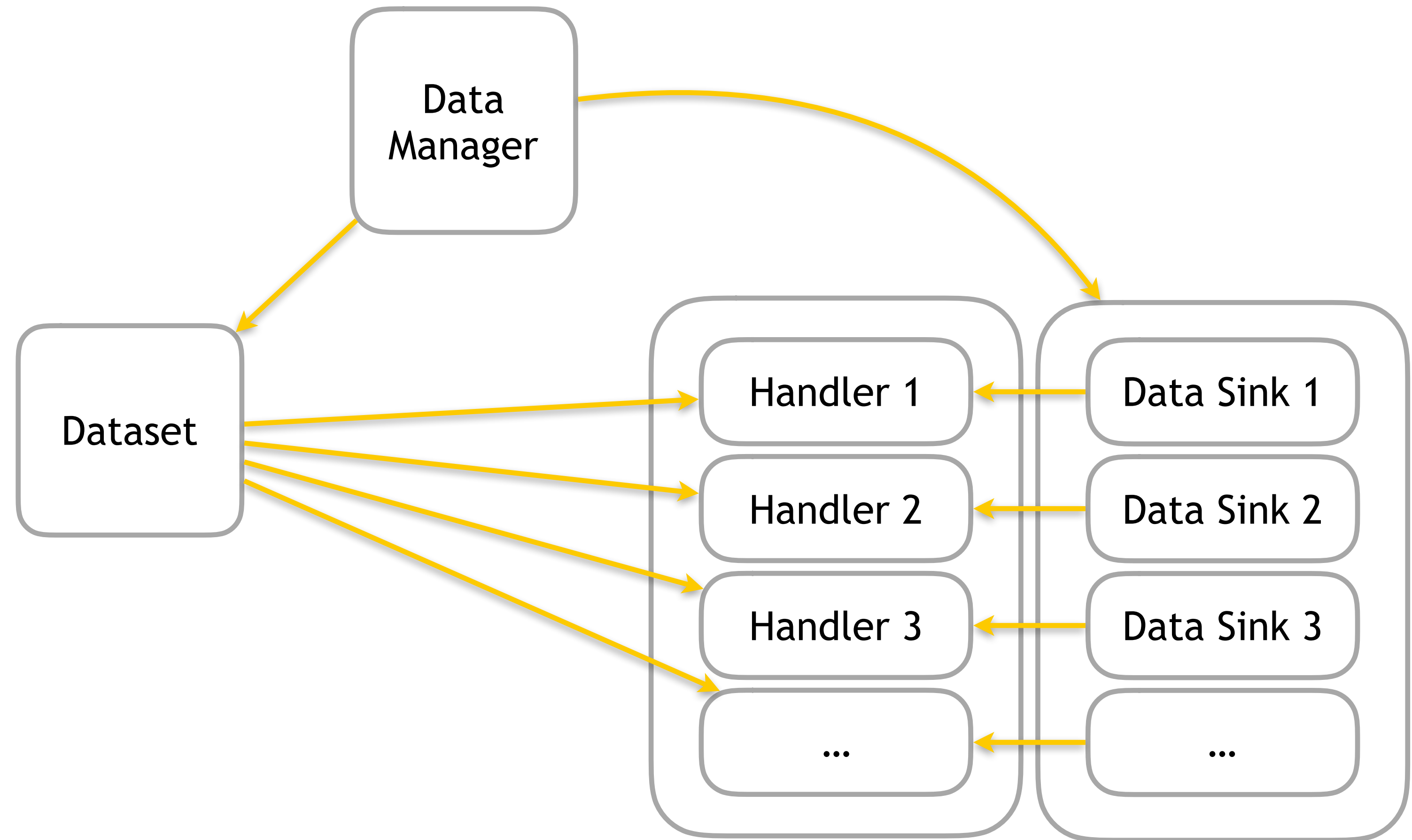
- Assigns to dataset:
 - Counter/Number
 - File name using templates
 - Start time
- Calls for each handler:
 - prepare()
 - begin()
- Writes metainfo to dataset (device and their parameters) and calls putMetainfo for each handler



- Writes results to dataset (detector data) and calls putResults for each handler after certain period of time
- Also provides the “quality” hook to the putResults method:
 - LIVE
 - INTERMEDIATE
 - FINAL
 - INTERRUPTED
- Time period determined using saveinterval, liveinterval parameters of detector



- Assigns to dataset:
 - End time
- Calls for each handler:
 - end()



Some example of data-sinks

ConsoleSink

Writes the scans results to console output

FileSink

Writes data to configured ASCII files

PNGImageSink

Writes Live data to a PNG file optionally with log10 colour scale

DaemonSink

Sends scan datasets to connected GUI clients

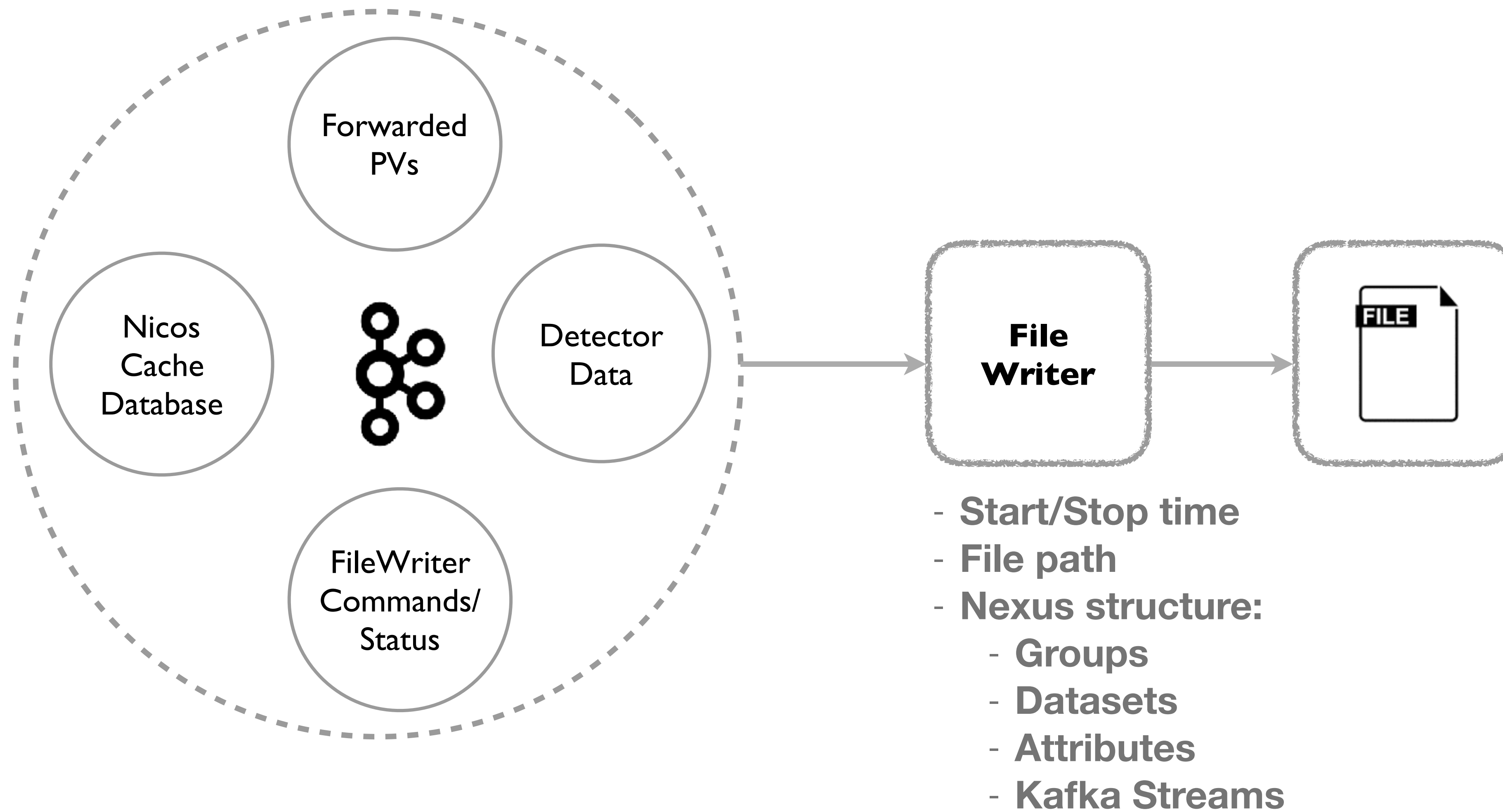
RawImageSink

Writes Raw image binary data to a file

NexusFileWriterSink

Writes Nexus files using kafka-to-nexus

Setup for File Writer



Nexus File Writer Handler

begin()

Send the FileWriter_new
command

Gather all the information
required from the nexus
template and send the JSON
command to Kafka

end()

Send the FileWriter_stop
command

Send the stop command to
Kafka and write the
information to NICOS cache



Nexus File Writer Status

- Monitors the Filewriter status topic
- Checks the status of jobs issued from NICOS
- Displays this information to users
- Additional treatment after the file has been closed by FileWriter

Providing NeXus Structure - The challenges

- Allow multiple NeXus hierarchies
- Provide properties of static devices in NeXus structure
- Some device parameters should always be streamed from Kafka
 - E.g. Counts from detectors
- Some devices that change during the file writing should be streamed from Kafka
- Event/Histogram data streaming

Python dictionary to represent hierarchy (keys mapped to one of the implemented nexus element)

NXAttribute

value
type

NXDataset

value
type
attributes

NXGroup

class
children
attributes

NXLink

path

KafkaStream

class
topic
source
....

HistogramStream

Histograms from
Kafka streams

EventStream

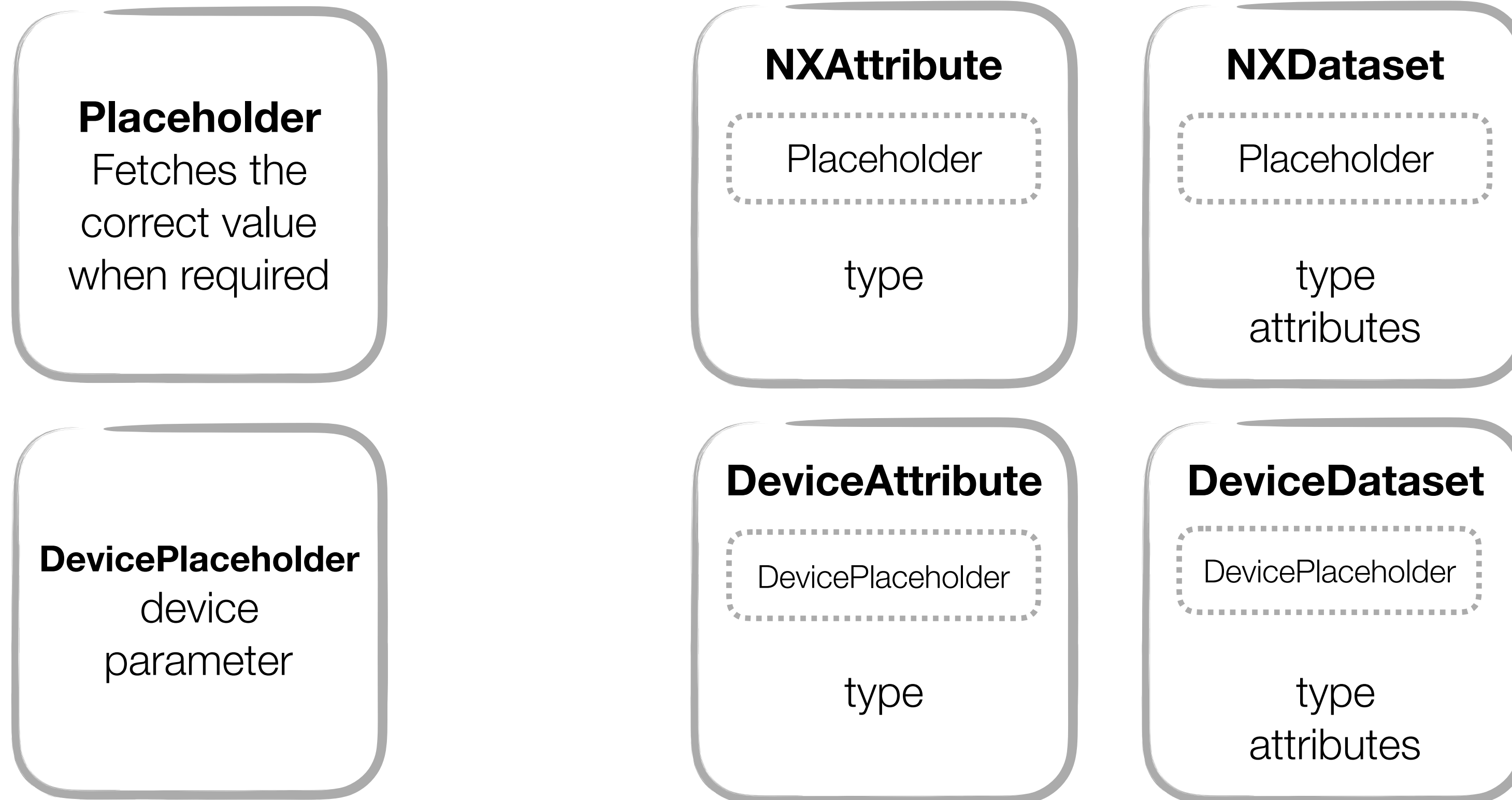
(NXevent_data)
Kafka event data
configuration

DeviceStream

(NXlog)
device
Parameter

Solution: Nexus Templates with Placeholders

Allow use of non-constant values in NeXus templates using placeholders which fetch the value just before writing the file



Example template

```
template = {
  "entry1:NXentry": {
    "sample: NXsample": {
      "distance": NXDataset(325.0, dtype=double, some_attr=32.0),
      "height": NXDataset(DeviceValuePlaceholder('dev')),
      "property": DeviceDataset('dev', 'param', unit='K'),
    },
    "INST:NXinstrument": {
      "name": NXDataset("Instrument"),
      "detector: NXdetector": {
        "data": EventStream(topic="EventTopic", source="SrcName")
      },
      "control: NXmonitor": {
        "mode": DeviceDataset('detector', 'mode', 'string'),
        "preset": DeviceDataset('detector', 'preset'),
        "monitor1": DeviceStream('c1'),
      },
    },
  },
}
```

- Limitations of the NeXus template - Devices not present:
 - Default values?
 - Modular templates?
- File Ownership?
- Update command with `FileWriter_end`?



Thanks.
Questions?