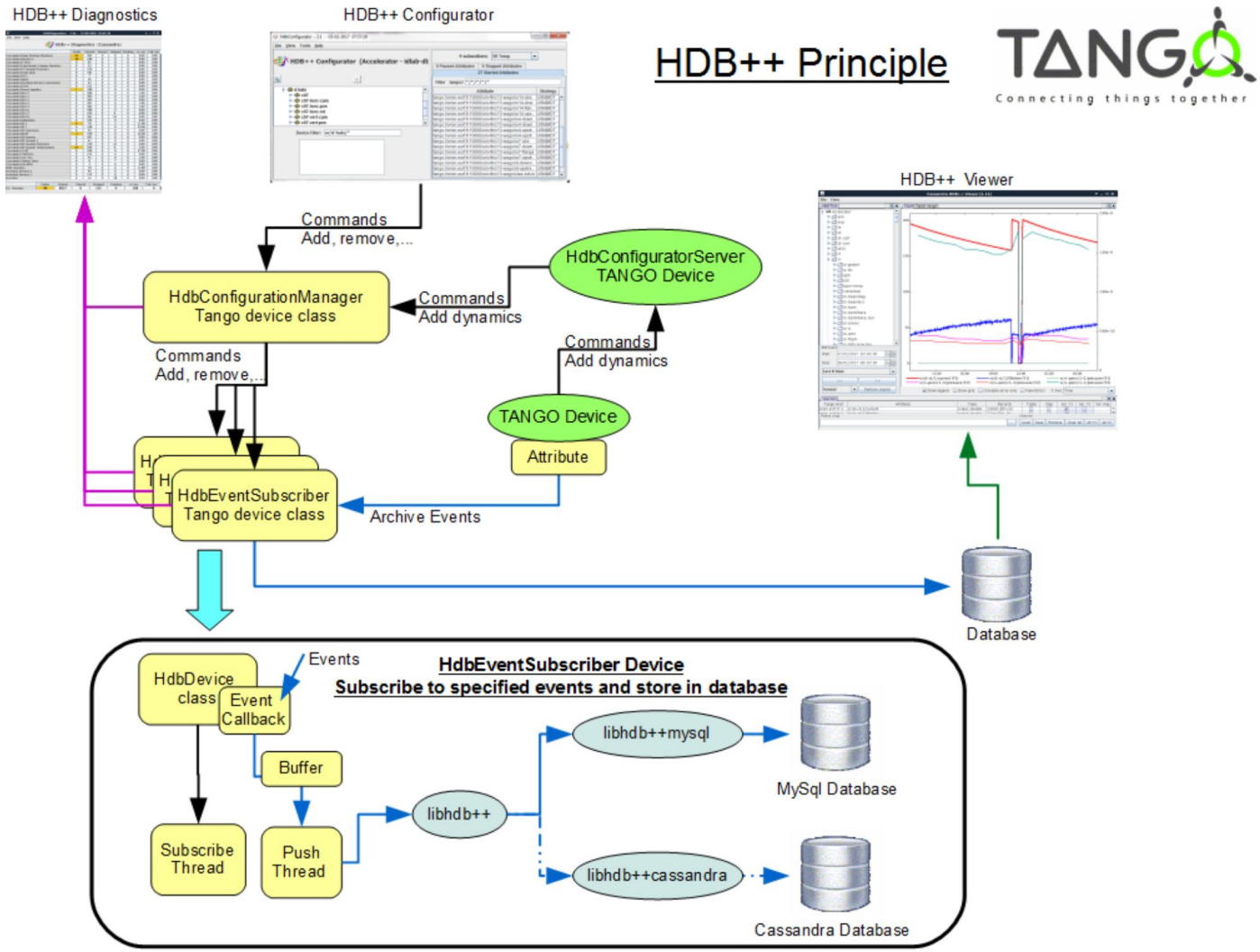
HDB++ UPDATES

Collaboration Status

HDB++ overview : the Archiving for Tango

https://tango-controls.readthedocs.io/en/latest/tools-and-extensions/archiving/HDB++.html



Database engines/flavours used on each institute

ESRF : Migrated from Cassandra to **TimeScale**

MaxIV : Planning to migrate from Cassandra to **TimeScale**

Elettra : Archiving on **MySQL** (new schema with JSON arrays) and legacy schema

ALBA : Archiving on **MariaDB** (compact schema, multi-DB, split arrays)

SKAO: likely to use **TimeScale**, using Elastic in some projects

BINP: **PostgreSQL**

JINR : **TimeScale**

<Your Institute Here> : < ? >

+ Non-standard archivers: periodic archivers, Elastic, Snaps

👍**Cmake Building from sources have been improved, so it becomes easier to download and test on your own local setup.**

Database Engine Sizes, not easily comparable

Long term maintenance/decimation is usually what takes more effort once the system is running, due to enlarging database size.

Size is affected by database engine, clustering, number of servers, partitioning policies, indexing, ..., Event frequency! (from 1 value day to 88v arrays at 20Hz), and type of data! (Shorts Vs String Arrays)

Graziano Scalamera (Elettra) working on benchmarks for different engines → they can be compared!

https://gitlab.com/tango-controls/hdbpp/hdbpp-benchmark

A rough rule of thumb:

- TimeScale for huge DB's, it requires big servers and clusters,
- MariaDB/MySQL for small-medium sized on a single host
- ~10000 attributes DBs are the "grey area" in between

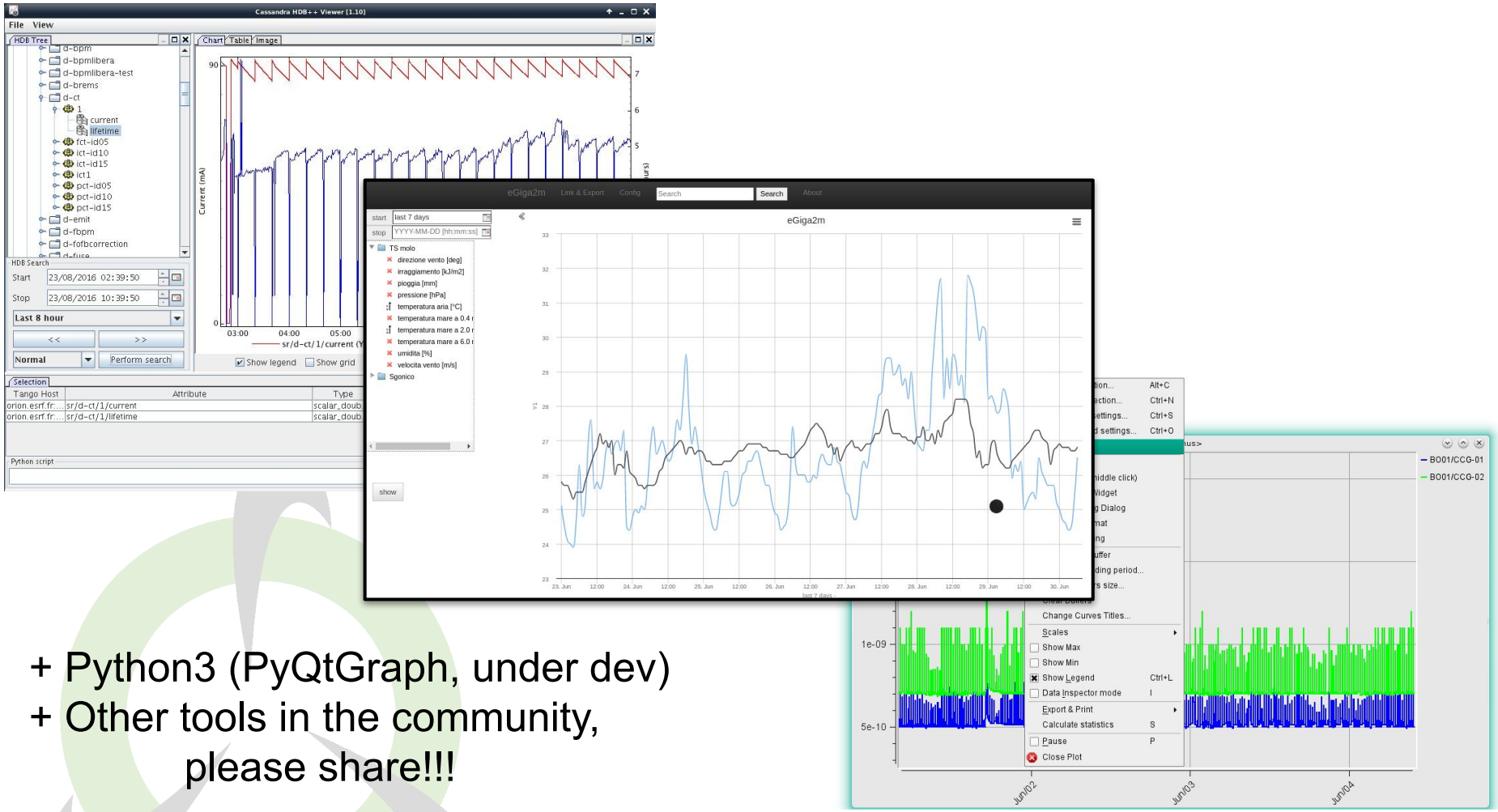
Archiving Clients, unifying on the pythonic side

Many tools

- Only HDBViewer is compatible with all DB engines.
- eGiga (web, by Elettra) widely used for MySQL
- PyTangoArchiving (MySQL/MariaDB + Taurus) have been migrated to python3, but it's still too dependent on DB engine.
- BINP also developed it's own python library for DB access (postgreSQL/mariadb).
- More schemas are on the way (ElasticSearch),

An specification for an AbstractReader and a main reader class have been developed in python3; to become the new python base tool for data extraction.

Archiving Clients, Java, Web/php, Python2 (Taurus)



+ Python3 (PyQtGraph, under dev)
+ Other tools in the community,
  please share!!!

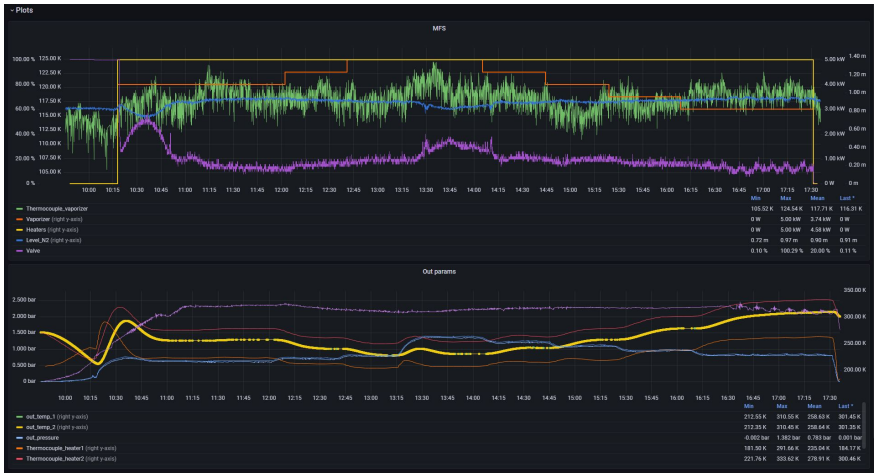
Archiving Clients (Grafana)

No coding or special libraries required, simple SQL queries used (TimeScaleDB example)

```
SELECT
  data_time AS "time",
  value_r as "temp_0"
FROM att_scalar_devdouble
WHERE
  $__timeFilter(data_time)
  and
  att_conf_id = (select att_conf_id from att_conf where
domain='fsd' and member='chip_temp' and name='temp_0')
ORDER BY 1
```

Can be used as soft real-time monitoring tool (auto-refresh option, min 5s). Extremely useful with continuous aggregates and DB cluster.

Alerting system included (Email, Slack, webhooks, etc.)



Trends

Synoptics (flowcharting plugin)
Examples:
https://play.grafana.org/dashboards/f/yIhcIRjZz/visualization-flowchart

Archiving AbstractReader

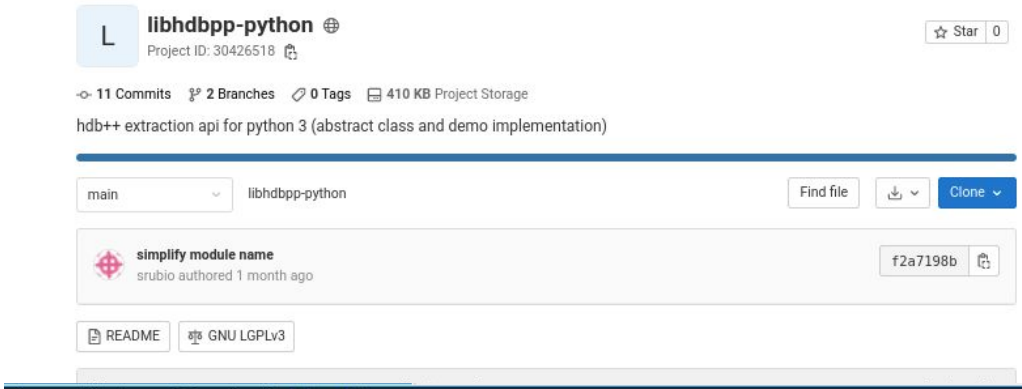
Fixed API to access any DB engine

Python chosen as the preferred language for OS science applications.

Usable from either CLI, Jupyter, web (Taranta), or GUI application.

Should be capable of mixing data from multiple databases (as PyTangoArchiving.Reader or ProxySQL does).

Capable to export/decimate data to file or load it into jupyter/ipython console.




```
class AbstractReader(object):
    """
    Subclass this class to create a PyTangoArchiving Reader for your specific DB

    e.g. TimeDBReader(AbstractReader)
    """

    def get_attributes(self, active=False, pattern=''):
        """
        Queries the database for the current list of archived attributes.
        arguments:
            active: True: only attributes currently archived
                    False: all attributes, even the one not archiving anymore
            pattern: '' :filter for attributes to retrieve
        """
        return list()

    def is_attribute_archived(self, attribute, active=False):
        """
        Returns if an attribute has values in DB.

        arguments:
            attribute: fqdn for the attribute.
            active: if true, only check for active attributes,
                    otherwise check all.
        """
        return True
```

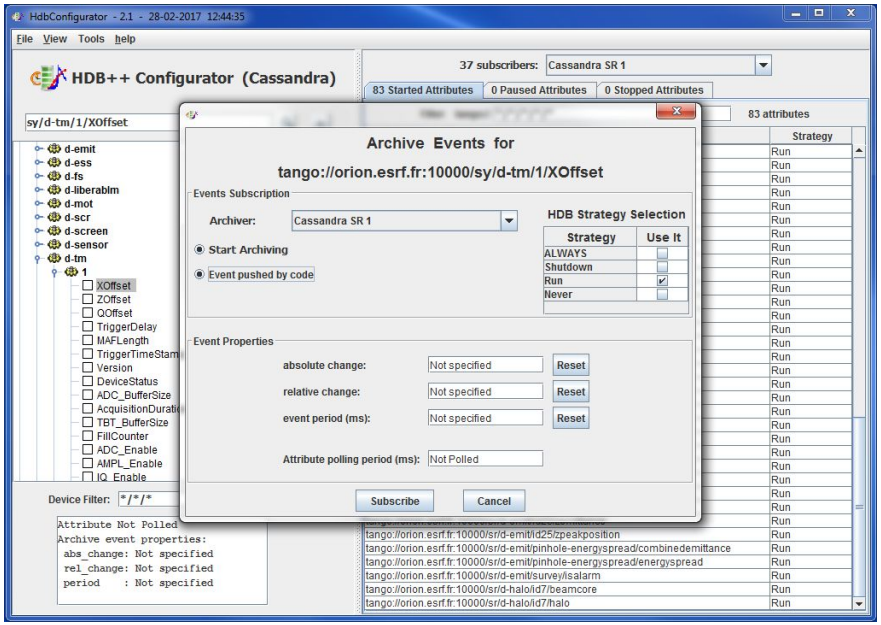
Configuring HDB++

HDBConfigurator (Java) is the most used tool.

PyTangoArchiving **(+ fandango, + panic)** already migrated to python3 (with help from solaris/**s2i**), providing many functionalities for archiving configuration.

MariaDB dependencies on extraction to be replaced by new AbstractReader.

MaxIV/SKA working on its own config tool (merge likely?).

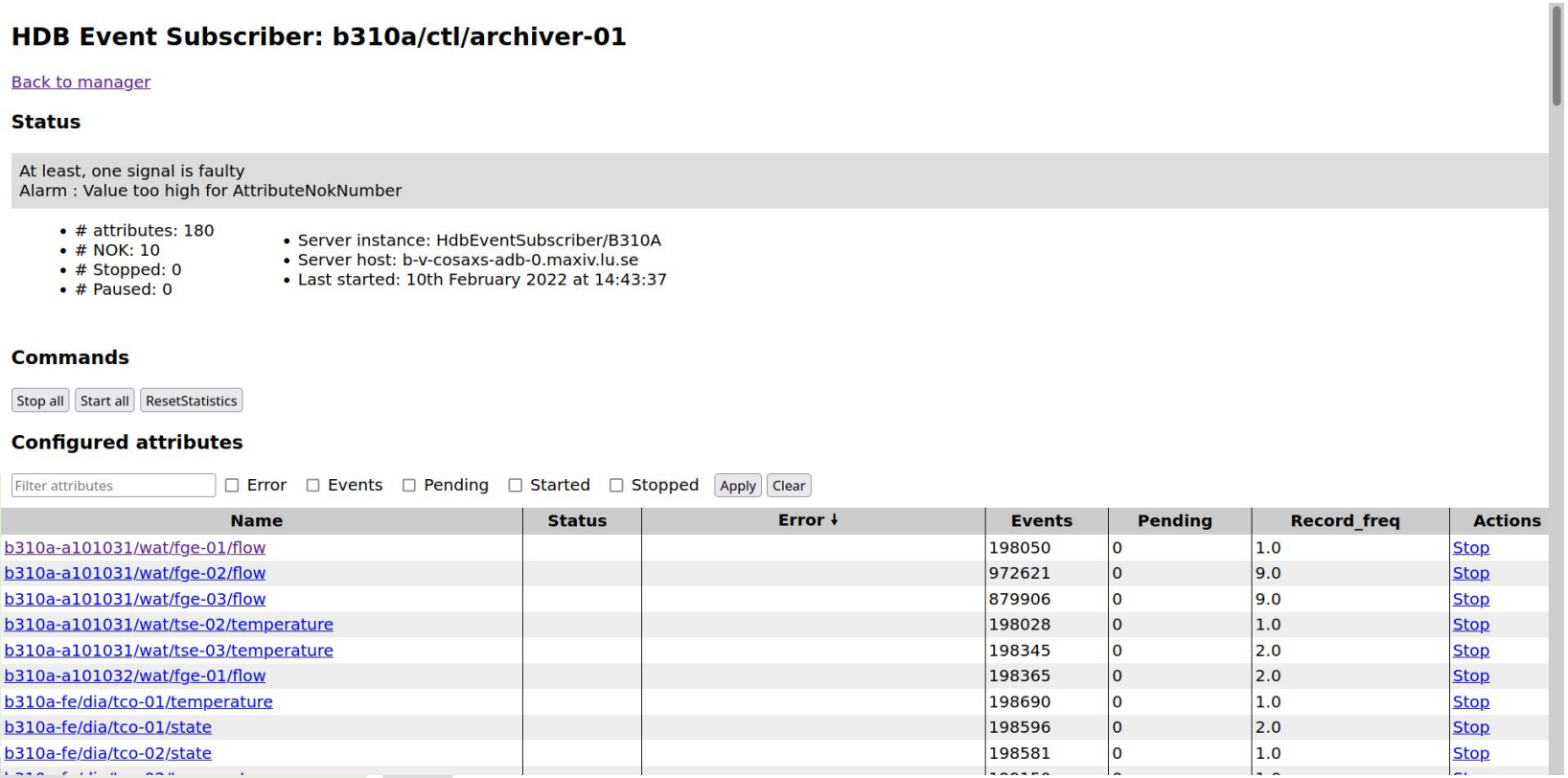



```
import PyTangoArchiving as pta, fandango as fn
api = pta.api('hdbvc')  # Choosing from pre-configured DBs
api.add_attributes(find_attributes('sr/di/dcct/*',code_event=True))
api.add_periodic_attribute('sr/vc/all/maxpressure',period=3000)
api.get_attribute_values('sr/vc/all/maxpressure', now()-100)
```

Monitoring HDB++

"Archwizard" is a web based tool for inspecting the current HDB++ configuration and investigating errors.

Developed at MAX IV -> https://gitlab.com/tango-controls/hdbpp/archwizard



HDB Event Subscriber: b310a/ctl/archiver-01

Back to manager

Status

At least, one signal is faulty
Alarm : Value too high for AttributeNokNumber

- # attributes: 180
- # NOK: 10
- # Stopped: 0
- # Paused: 0

- Server instance: HdbEventSubscriber/B310A
- Server host: b-v-cosaxs-adb-0.maxiv.lu.se
- Last started: 10th February 2022 at 14:43:37

Commands

Stop all | Start all | ResetStatistics

Configured attributes

Filter attributes ☐ Error ☐ Events ☐ Pending ☐ Started ☐ Stopped Apply Clear

Name	Status	Error ↓	Events	Pending	Record_freq	Actions
b310a-a101031/wat/fge-01/flow			198050	0	1.0	Stop
b310a-a101031/wat/fge-02/flow			972621	0	9.0	Stop
b310a-a101031/wat/fge-03/flow			879906	0	9.0	Stop
b310a-a101031/wat/tse-02/temperature			198028	0	1.0	Stop
b310a-a101031/wat/tse-03/temperature			198345	0	2.0	Stop
b310a-a101032/wat/fge-01/flow			198365	0	2.0	Stop
b310a-fe/dia/tco-01/temperature			198690	0	1.0	Stop
b310a-fe/dia/tco-01/state			198596	0	2.0	Stop
b310a-fe/dia/tco-02/state			198581	0	1.0	Stop

More Info regarding HDB++

SLACK, #hdbpp@tango-controls

Tango Forums!

Whole project migrated to gitlab!

- https://gitlab.com/tango-controls/hdbpp
- **.../hdbpp-tickets** and **.../meeting-minutes**
- regular meetings every 2 months

The HDB++ crew: Reynald Bourtembourg, Lorenzo Pivetta, Sergi Rubio, Thomas Juerges, Stuart James, Benjamin Bertrand, Damien Lacoste, Graziano Scalamera, Giacomo Strangolino, Luzio Zamboni, Johan Forsberg, Anton Joubert, Mirjam Lindberg, teams at JINR/BINP and many, many others.