UCAP: A FRAMEWORK FOR ACCELERATOR CONTROLS DATA PROCESSING @ CERN

L. Cseppentő and M. Büttner, CERN, Geneva, Switzerland

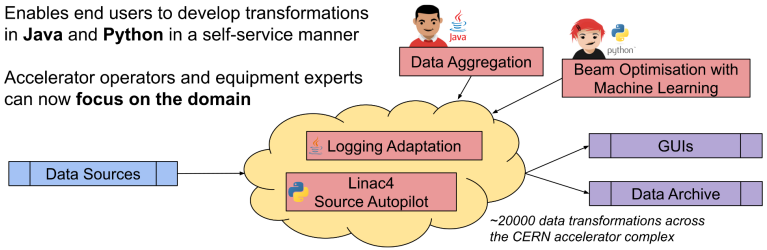
Overview

UCAP = Unified Controls Acquisition and Processing

Tackles **"Acquisition - Transformation - Publishing/Presentation"** use cases

Enables end users to develop transformations in **Java** and **Python** in a self-service manner

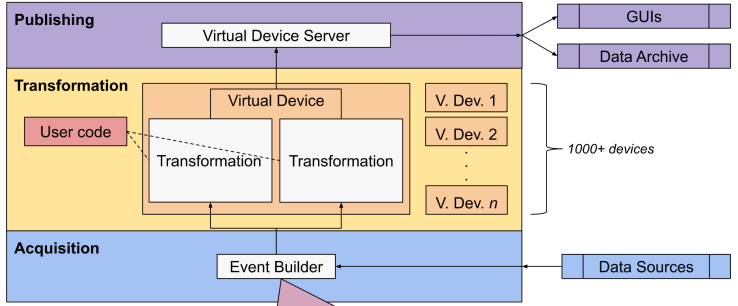
Accelerator operators and equipment experts can now **focus on the domain**



Similar Problems - Common Solution

Architecture

Generic Framework for Online Data Processing



Groups Several Input Signals

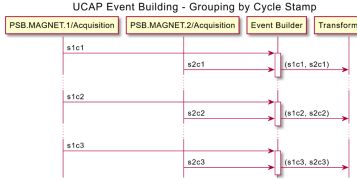
Event Building

Acquiring and grouping inputs from several data sources

Declarative configuration

10 **reusable** data acquisition algorithms

Handles errors such as: source is not available, signal is late




```
{
  "type": "GroupTriggeredCycleStampGrouped",
  "triggerGroup": {
    "subscriptions": [
      {
        "parameter": "PSB.MAGNET.1/Acquisition",
        "selector": "PSB.USER.ALL"
      },
      {
        "parameter": "PSB.MAGNET.2/Acquisition",
        "selector": "PSB.USER.ALL"
      }
    ],
    "timeoutMs": 1000
  },
  "bufferedSubscriptions": [
    {
      "parameter": "LSA.DEVICE/Setting"
    }
  ]
}
```

This definition will make UCAP to synthesise an event builder grouping data from the `PSB.MAGNET.1/Acquisition` and `PSB.MAGNET.2/Acquisition` parameters by cycle stamp.

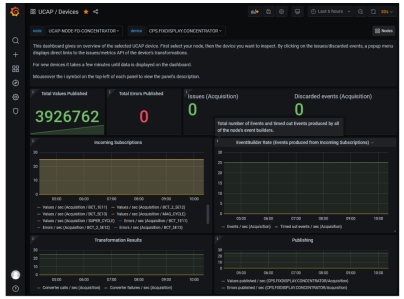
The `timeoutMs` field specifies maximum how long the event builder should wait for data from all devices — if data is missing, a partial event is fired containing only the inputs which arrived on time.

The `bufferedSubscription` part is an optional place to specify other inputs which are needed by the transformation, such as LSA settings.

User Experience

Scalable Self-Service Model

Rich CLI



Automagically Populated Dashboards

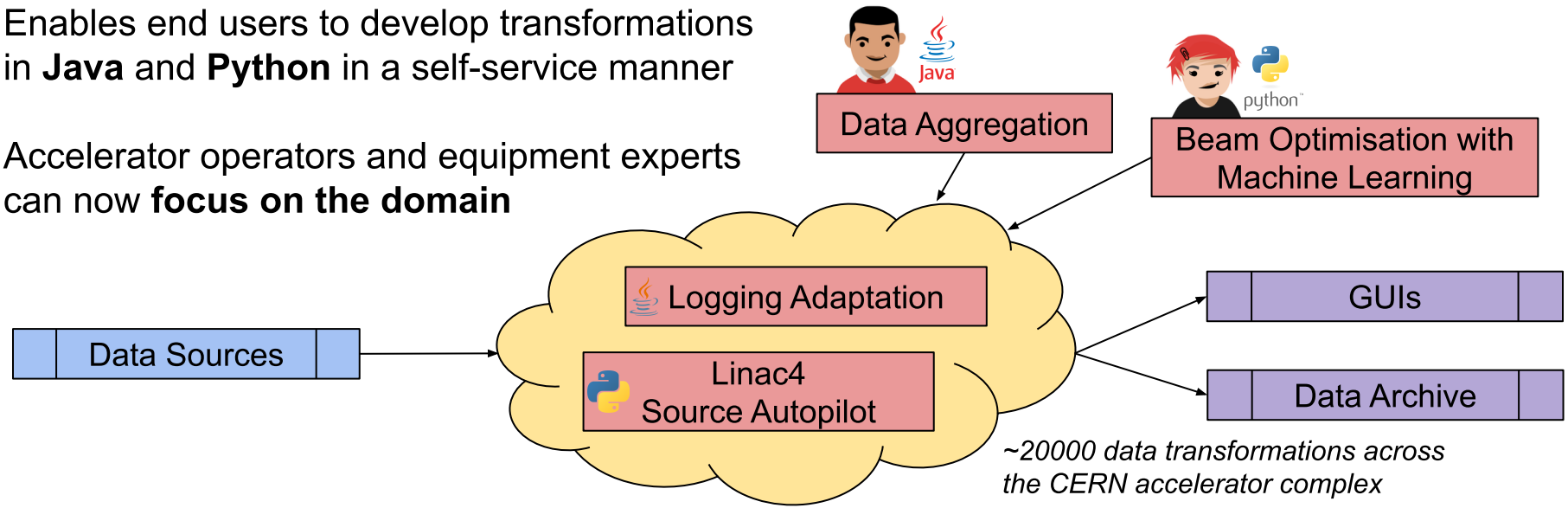
Overview

UCAP = Unified Controls Acquisition and Processing

Tackles **“Acquisition - Transformation - Publishing/Presentation”** use cases

Enables end users to develop transformations in **Java** and **Python** in a self-service manner

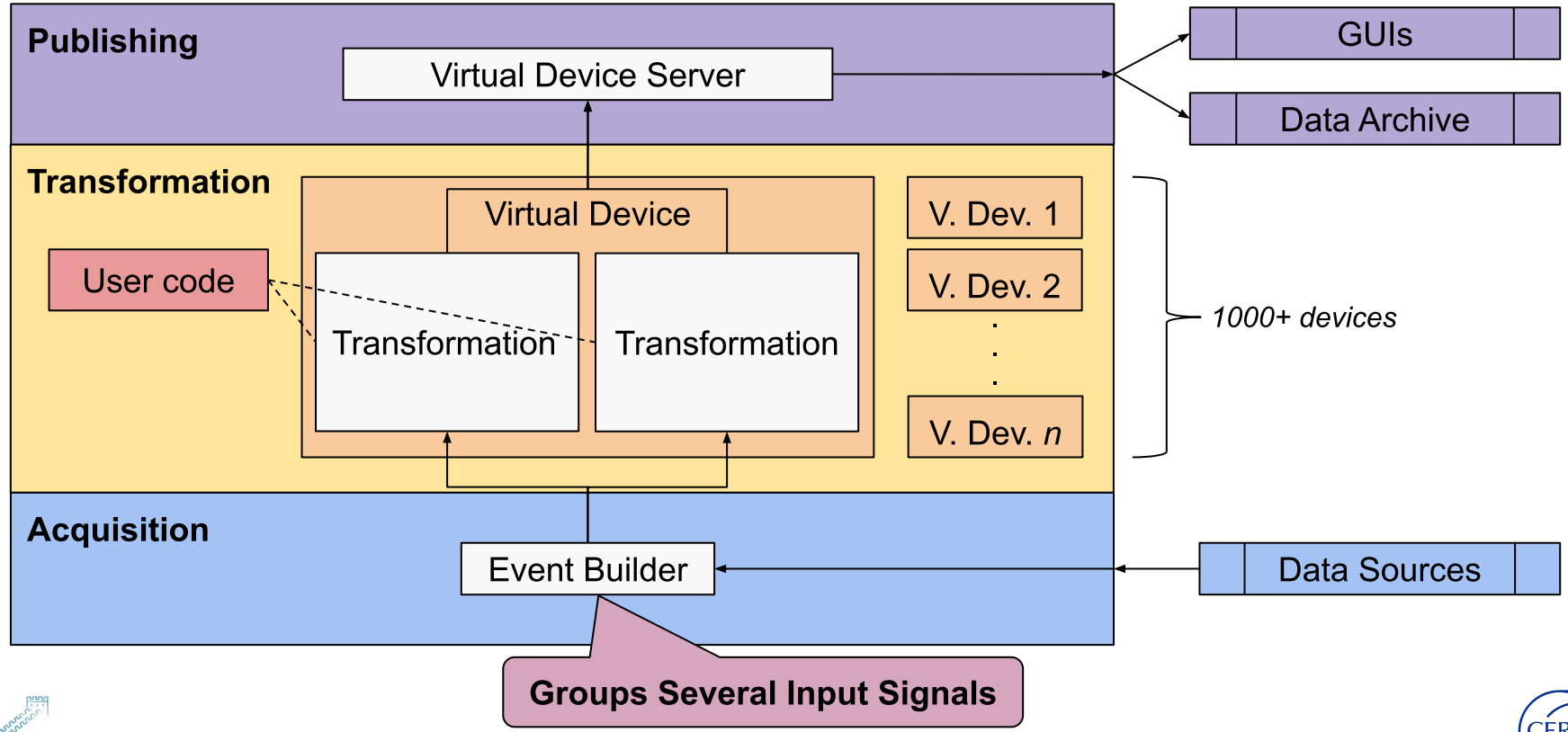
Accelerator operators and equipment experts can now **focus on the domain**



Similar Problems - Common Solution

Architecture

Generic Framework for Online Data Processing



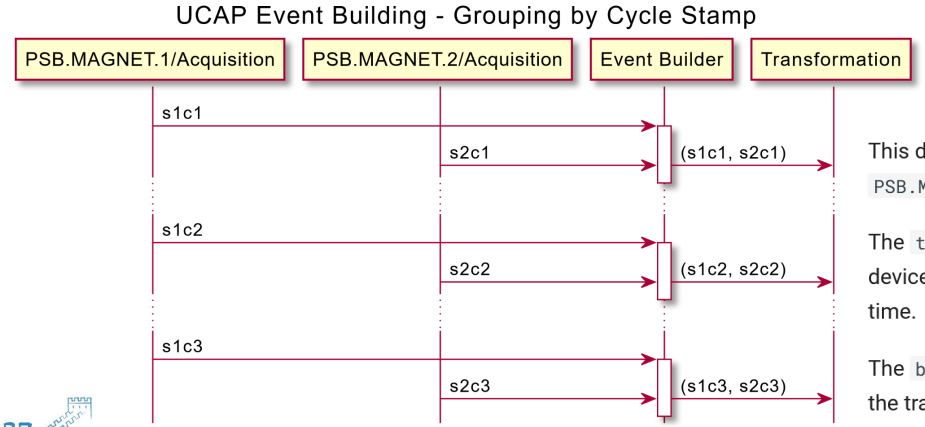
Event Building

Acquiring and grouping inputs from several data sources

Declarative configuration

10 **reusable** data acquisition algorithms

Handles errors such as: source is not available, signal is late




```
{
  "type": "GroupTriggeredCycleStampGrouped",
  "triggerGroup": {
    "subscriptions": [
      {
        "parameter": "PSB.MAGNET.1/Acquisition",
        "selector": "PSB.USER.ALL"
      },
      {
        "parameter": "PSB.MAGNET.2/Acquisition",
        "selector": "PSB.USER.ALL"
      }
    ],
    "timeoutMs": 1000
  },
  "bufferedSubscriptions": [
    {
      "parameter": "LSA.DEVICE/Setting"
    }
  ]
}
```

This definition will make UCAP to synthesise an event builder grouping data from the `PSB.MAGNET.1/Acquisition` and `PSB.MAGNET.2/Acquisition` parameters by cycle stamp.

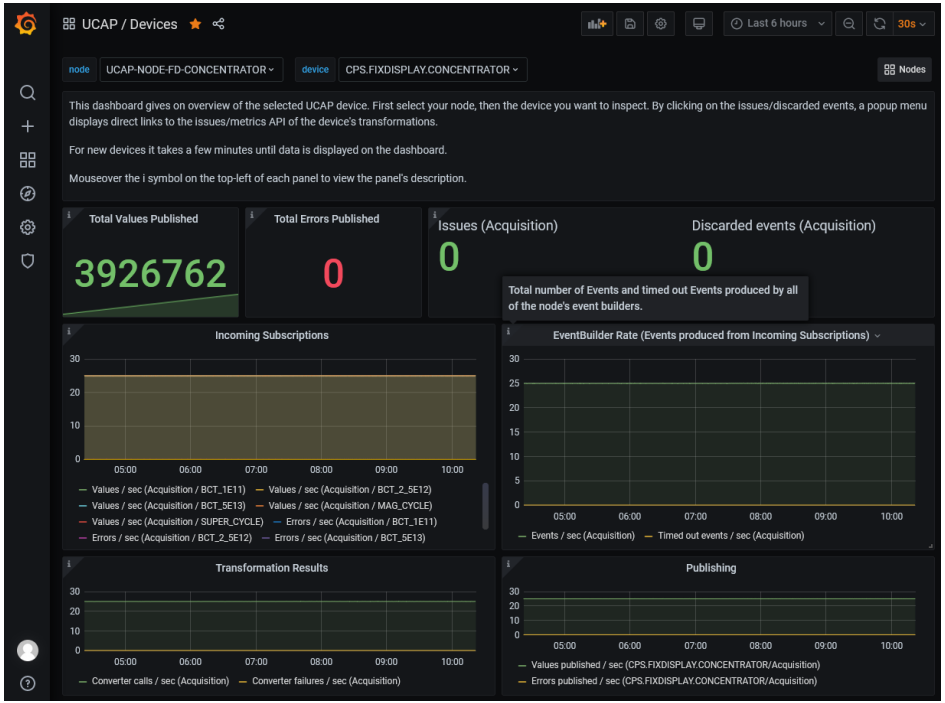
The `timeoutMs` field specifies maximum how long the event builder should wait for data from all devices — if data is missing, a partial event is fired containing only the inputs which arrived on time.

The `bufferedSubscription` part is an optional place to specify other inputs which are needed by the transformation, such as LSA settings.

User Experience

Scalable Self-Service Model

```
$ ucap-cli
      ############################################################
    ##                                                          ##
  ##                                                              ##
##        ###      ###  ############  ############  ############        ##
##        ###      ###  ###           ###           ##        ##        ##
##        ###      ###  ###           ############  ##        ##        ##
##        ###      ###  ###           ###      ###  ##        ##        ##
##        ############  ############  ############  ############        ##
##                                                  ##                  ##
  ##                                              ##                ##
    ##                                                          ##
      ############################################################
INFO: Initialising UCAP-CLI...
INFO: RBAC authentication...
Hello there... LAJOS CSEPPENTO (lcseppen)
INFO: Initialising terminal...
ucap-cli> help
  System:
    exit                        exit from app/script
    help                        command help
  Node:
    node-discover               Discovers UCAP nodes (note: already done o
    node-list                   Lists UCAP nodes (alias: nl)
    node-select                 Selects a UCAP node to work with (alias: n
    node-update                 Updates a node
  Device:
    device-create               Creates/updates devices from JSON files
    device-delete               Deletes devices from the selected node
    device-download             Downloads device definition JSON files to
    device-list                 Lists/filters devices on selected node (al
    device-reload               Reloads devices on the selected node
    device-show                 Shows detailed information for a device (a
    device-start                Starts devices on the selected node
    device-stop                 Stops devices on the selected node
    subscribe-japc              Subscribes to a device-property with JAPC
    subscribe-ucap              Subscribes to UCAP device-properties on th
```



Rich CLI

Automagically Populated Dashboards