

# DAQ and controls overview

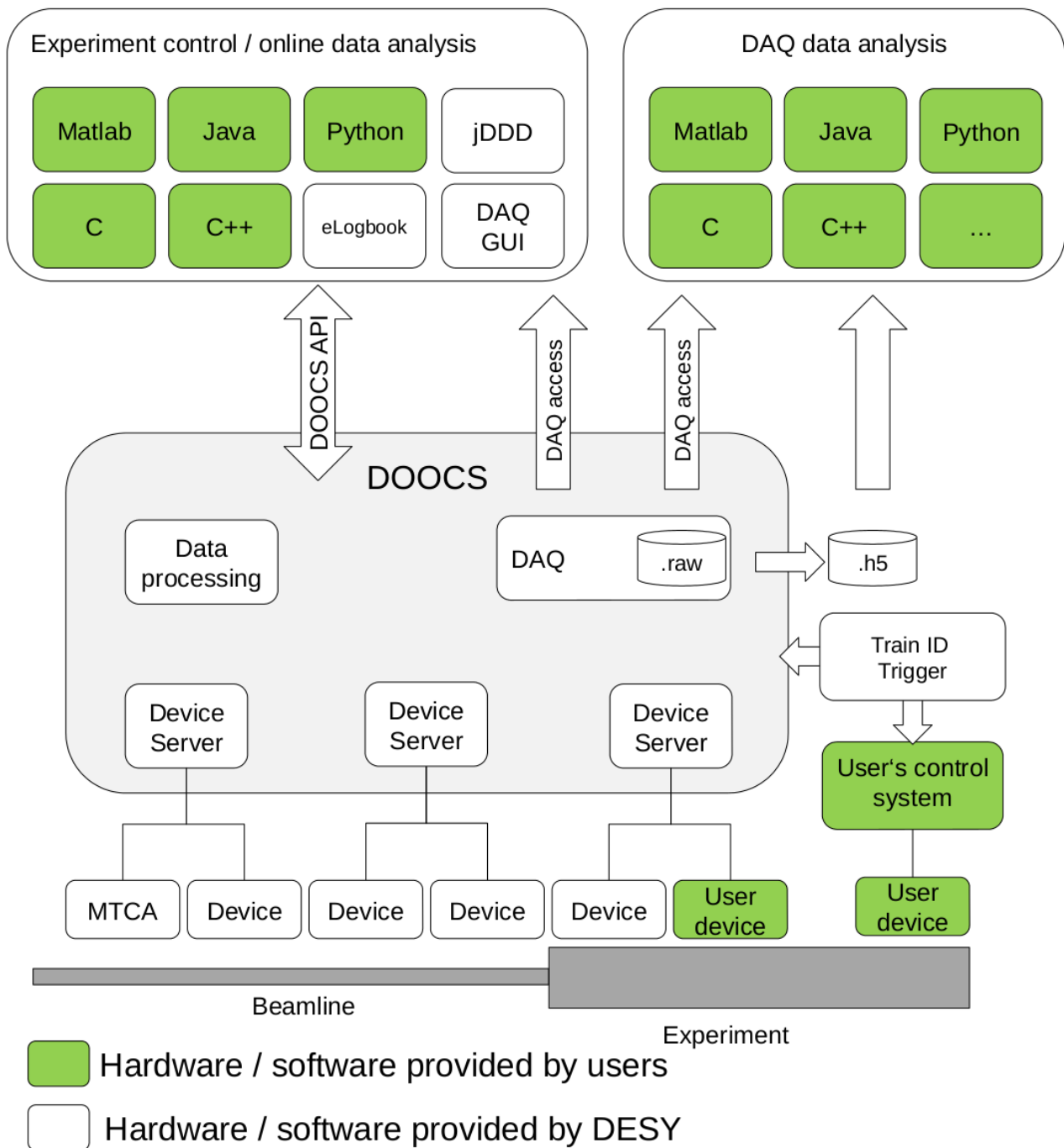
Before starting a beamtime at FLASH the users have to consider several options regarding the control and data acquisition system where the users are supported by [the controls group](#) .

## Contents

- [Contents](#)
- [Overview](#)
- [Things to consider before the beamtime](#)
- [Beamline control and photon diagnostics](#)
- [IT infrastructure](#)
- [Data access](#)
- [Online](#)
- [Data acquisition](#)
- [nearOnline](#)
  - [.raw files](#)
  - [HDF5 files](#)
- [Offline](#)

---

## Overview



The FLASH accelerator, beamlines and experiments are using the Distributed Object Oriented Control System ( [DOOCS](#) ). Beamline and experiment devices are mainly operated via graphical user interfaces created with the Java DOOCS Data Display ( [jddd](#) ) by the FLASH team or the users themselves. Further it is provided a [DOOCS client API](#) in Java, Python, Matlab, C or C++ to the users to write own software for data access, online data or device control. User-provided devices can be run in parallel or can be implemented in DOOCS. For each beamline and accelerator is a dedicated [electronic logbook](#) available in which jddd and other software can print directly.

The data acquisition system (DAQ) collects beamline and experimental data and writes them to hard disk. All data are stored along with a time stamp and, in case of train specific data, along with a (pulse) train ID. There are several ways to access beamline and experimental data stored in the DAQ during or after the beamtime be it programmatically via DAQ access libraries or after data conversion to HDF5 format. While the data transport to the user's home institution is possible, DESY hosts high power computing resources which can be used for analysis.

[top](#)

## Things to consider before the beamtime

In consultation with the user's local contact the user should check our [beamtime preparation checklist](#) and fill in the [beamtime preparation questionnaire](#).

[top](#)

## Beamline control and photon diagnostics

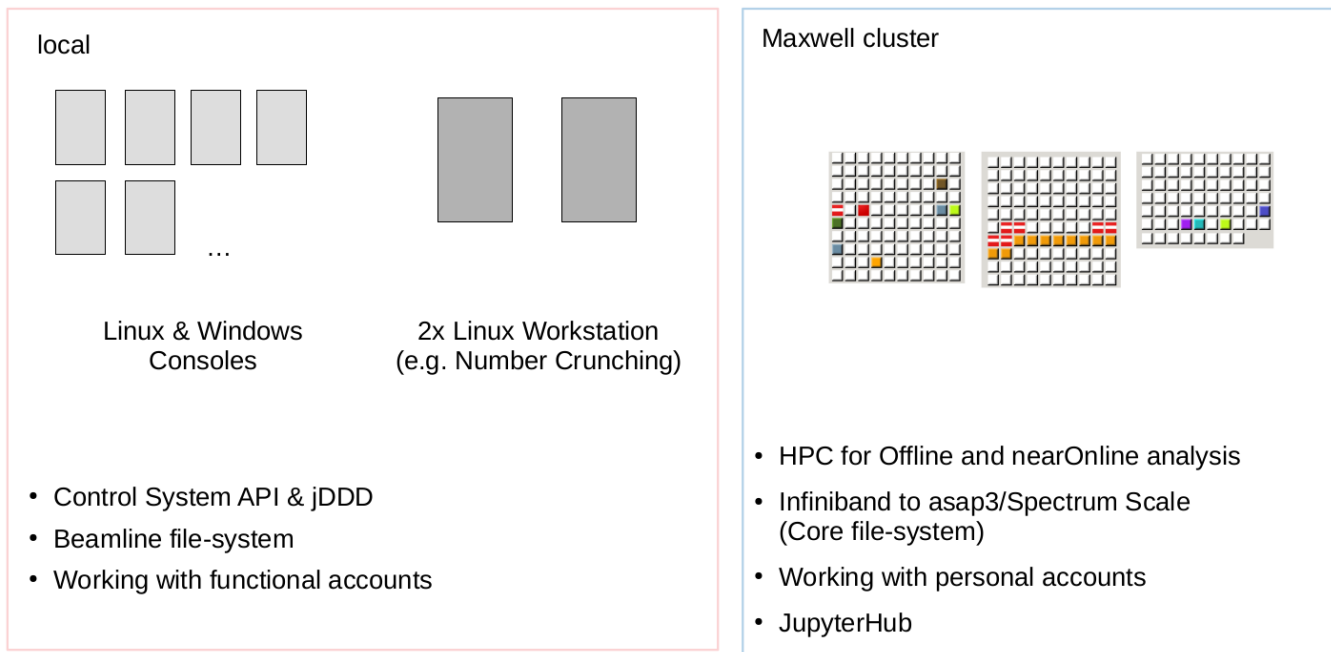
Use existing jddd panel provided by FLASH or use the jddd editor to create a new panel according to the user's own needs.



[top](#)

## IT infrastructure

During a beamtime at FLASH we have two IT infrastructures each with different purpose. In the FLASH halls you have [local machines](#) which are used with functional accounts and they have access to the [beamline file-system](#) for your current experiment. For more demanding task we could also provide workstations which can be dedicated to a single user experiment. For offline and nearOnline analysis the [Maxwell cluster](#) for high performance computing is available. On the Maxwell cluster you have to work with personal accounts as this regulates data access to the [core file-system](#).



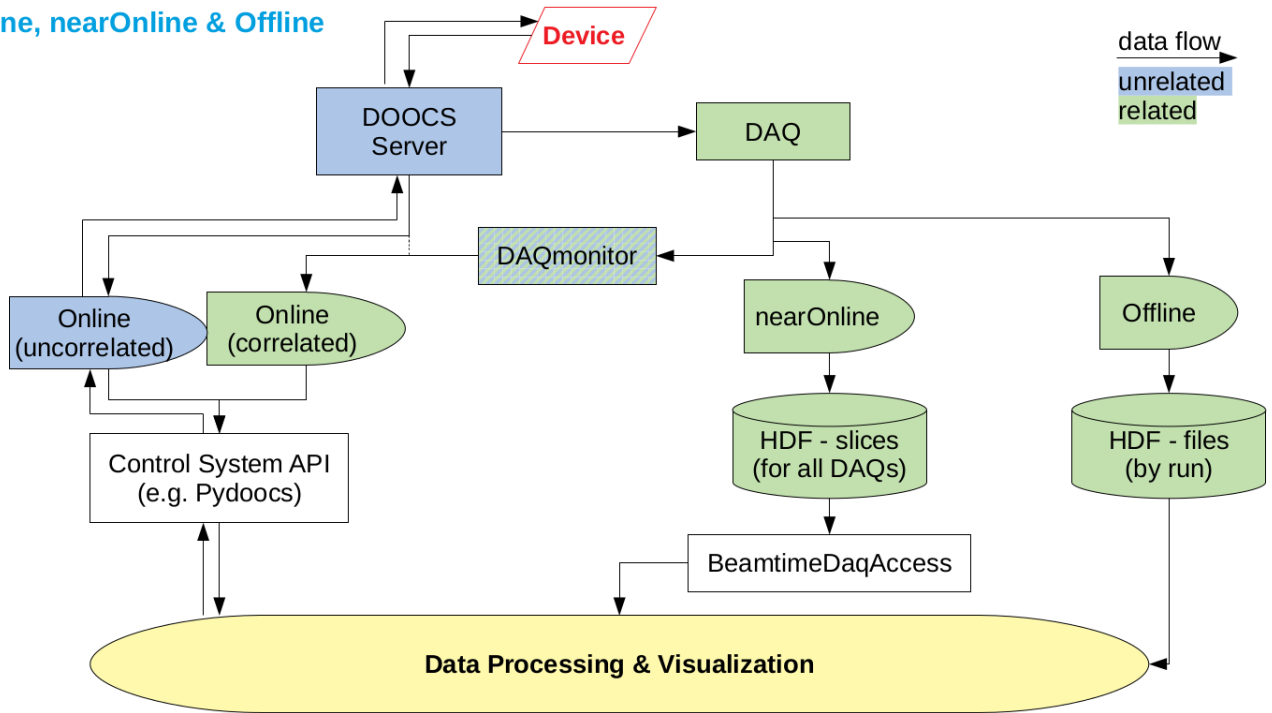
The FLASH control & DAQ system supports [several devices](#) and at each beamline are [MTCA ADCs](#) available. With the MTCA technology it is possible to [synchronize own devices in respect to the FEL](#). If the user is using his or her own DAQ system it is further possible to [receive the trainId via network](#) for the purpose of synchronization.

<a href="#">local machines</a>	<a href="#">Maxwell cluster</a>	<a href="#">MTCA ADCs</a>	<a href="#">supported devices</a>	<a href="#">supported cameras</a>	<a href="#">trigger via MTCA</a>	<a href="#">trainId via network</a>
--------------------------------	---------------------------------	---------------------------	-----------------------------------	-----------------------------------	----------------------------------	-------------------------------------

[top](#)

## Data access

## Online, nearOnline & Offline



At the Free-electron Laser Hamburg ( [FLASH](#) ) we use the Distributed Object Oriented Control System ( [DOOCS](#) ). Devices are implemented via DOOCS server and via an API ( [ONLINE](#) ) it is possible to request data directly from the DOOCS server by knowing the DOOCS address. As correlations of different physical properties are often required all data at FLASH are indexed by the [trainID](#), which identify each of FLASH's pulse trains. The during a beamtime recorded data are stored via a Data Acquisition System ( [DAQ](#) ) which sort all events from the individual DOOCS server by trainID. When requested [HDF files](#) are created after the beamtime which includes the important data from [the accelerator and its diagnostic](#) as well as the data created by the [users](#). This time scale we define as *offline* as the HDF files are converted after the beamtime is over. For synchronous data during an experiment it is possible to create shorter HDF slices via a *nearOnline* converter within a few minutes. For working with this partially incomplete HDF slices we provide an API called [BeamtimeDaqAccess](#). Reading synchronous data via an *online* API is possible via a configurable DAQ middle layer server, the [DAQmonitor](#), which feeds the correlated data back in the control system while it provides a ring buffer with 32 events in size.

online	live at 10 Hz
online via DAQmonitor	live up to 3s into the past
nearOnline	a few minutes
offline	after the beamtime

[top](#)

## Online

To monitor individual parameters online, e.g. ADCs or cameras, the use of `jddd` is recommended. For more complex tasks the users can use the DOOCS client API for [Matlab](#), [Python](#) and [Labview](#). For accessing the control system [DOOCS addresses](#) are required.

A collection (Matlab [ [TOF & camera GUIs](#) ] / [Jupyter notebooks](#) ) for common use cases is available.

[top](#)

## Data acquisition

Relevant machine data, e.g. pulse energy or arrival time, are saved in the FLASH Photon diagnostics DAQ (PBD) while experiment related parameter are saved on demand in the FLASH [User DAQs](#). In addition it is disk space available for devices and parameter outside the FLASH DAQ, which can be [synchronized via Ethernet connection](#). The for the beamtime provided storage space, the [spectrum scale](#) formerly known as ASAP3 /GPFS, is access regulated via [DOOR](#) and only registered participants have access.

The FLASH DAQ system records the data in binary `.raw` files. On request the data will also be available in the HDF5 format after a conversion during or after the beamtime. Incoming data is collected, sorted and saved into `.raw` files in chunks of 60 MB to 1 GB which corresponds to tens of seconds up to several minutes. The HDF5 files can be created *nearOnline* or *Offline*. In the *nearOnline* conversion every individual `.raw` files will be converted to a single HDF5 file to provide the fastest access possible. After the beamtime it is possible to get a HDF5 file per DAQ run which is very convenient as it contains the merged data of the User DAQ and the PBD DAQ.

While the DOOCS addresses are rather cryptic the [HDF5 file](#) is structured based on the actual location and function of the devices. A complete list is available in [DESY's Repository](#) .

<a href="#">DAQ introductions</a>	<a href="#">FLASH1 PBD</a>	<a href="#">FLASH2 PBD</a>	<a href="#">User DAQs</a>	<a href="#">User DAQs subsystems</a>	<a href="#">HDF5 files</a>
-----------------------------------	----------------------------	----------------------------	---------------------------	--------------------------------------	----------------------------

[top](#)

## nearOnline

### **.raw files**

The .raw files are only accessible from certain computers within the DESY network and in general it is not recommended to use them directly. For checking the file content and doing simple analysis, e.g. histogram, line outs, we provide the [DAQdataGUI](#). If the user already have very precise knowledge of the desired parameters and it's types it is possible [to read with Matlab from the files directly](#).

FLASH's DAQ .raw files are saved locally and with a time delay of a few minutes backedup via tape ( [dCache](#) ) by the DESY central IT.

### **HDF5 files**

As the .raw files are highly optimized for writing speed there are some issues which have to be taken care of which are also present in the nearOnline [HDF5 files](#) (e.g. missing/doubled events). For this we provide with [BeamtimeDaqAccess](#) an API which handles a lot of the occurring difficulties as knowing the file or DAQ instances. The [BeamtimeDaqAccess](#) library is written in Python, but a Matlab wrapper is available.

[top](#)

## Offline

The offline [HDF5 files](#) containing complete DAQ runs will be put after the beamtime on the spectrum scale file system. Access is granted from the within the DESY network for beamtime participants or from the outside via [Gamma portal](#) . HDF-by-run files can processed with common tools, e.g. with [Matlab](#) or [Python](#), as during the conversion a lot of discrepancies have been resolved. As .raw files are rarely used for analysis they are saved on the FLASH DAQ servers and will only be put on the spectrum scale file system on special request.

The spectrum scale file system contains the [substructure](#) : raw, processed, shared and scratch. The raw folder contains the data recorded by the experiment (user data and HDF5) and will be set read only shortly after the beamtime is over. Processed is typically used to store processed data and analysis software. Both folder are backedup by the spectrum scale file system.

[top](#)

---