

Application Programming Interface for SACLA User Experiments



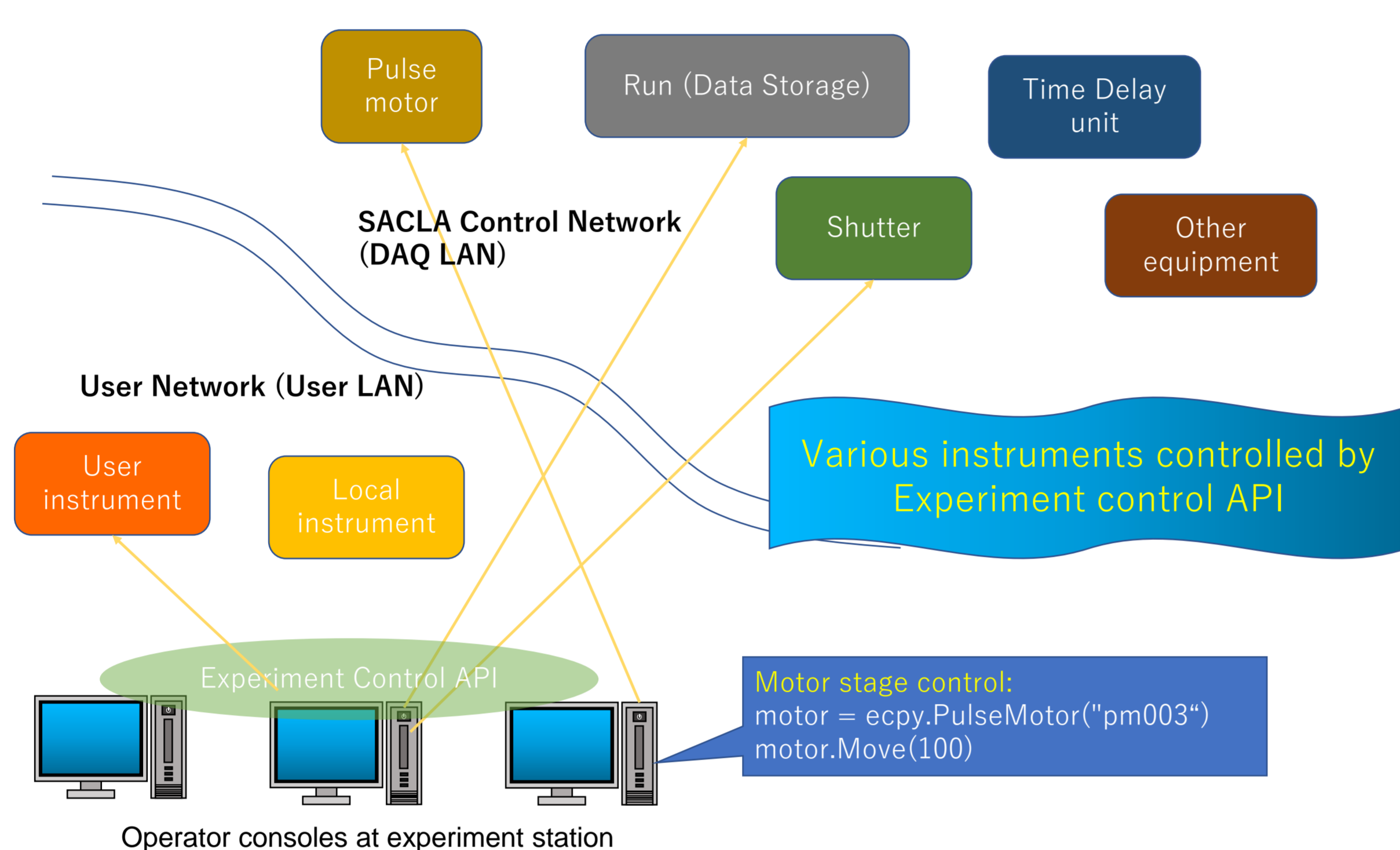
Koji Motomura
On behalf of RIKEN SPRing-8 Center (SACLA),

Overview

Application programming interfaces that can control and access to various resources of users' experiments have been developed at SACLA system. Among them, we particularly introduce Experiment control API (ecpy) and Data access API (dbpy) in this poster. "ecpy" is for controlling instruments, and "dbpy" is for accessing experiment data. They are available for all experimental users using operator consoles (opcons) at experiment stations. All APIs have Python binding interface, therefore users can simply make effective programs for the experiments.

	Experiment Control API	Data Access API
Python module	<i>ecpy</i>	<i>dbpy</i>
Language	Python3	Python3, C
Hosts	opcon	opcon, anapc, HPC
Description	Control instruments and Run (storage)	Access shot-by-shot data and Run information

Experiment Control API (ecpy)



Application programming interface for experiment control (named *ecpy*) is a Python package to control Run (a unit of data acquisition) and instruments in SACLA. This package has many classes which provide interfaces to control them. The unified interfaces (method, property) without complex procedure makes a programming easily and quickly.

```
import ecpy

run = ecpy.Run()
run.Initialize()

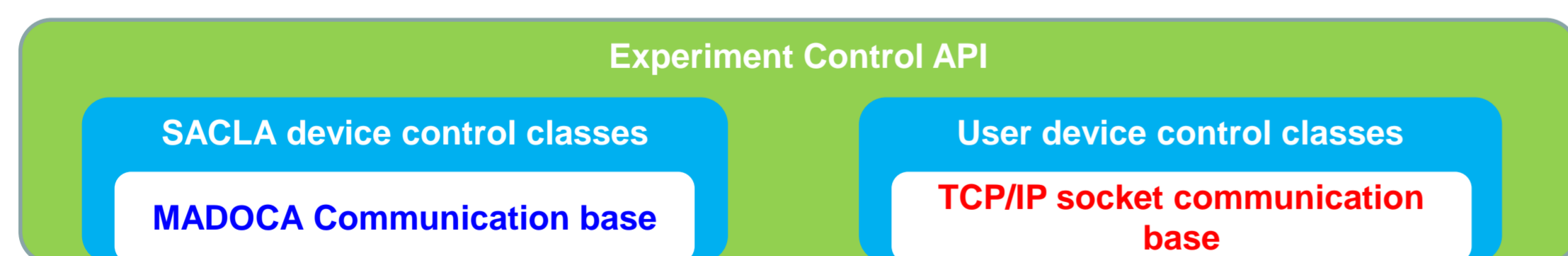
run.EnableShutter("B12Xfel")
run.DisableShutter("B12Lh1")
run.EnableStorage("MPCCD-880-2-003")
run.SetComment("Run test")
run.SetShotNumber(100)
run.SetShutterPattern("B12Xfel", "continuous", 100)

run.Start()
while run.IsRunning() == True:
    pass

run.End()
run.Finalize()
```

Easy scripting to control the experiments

*Example code: Carry out a Run with 100 shots and FEL shutter open



```
import ecpy
pm1 = ecpy.PulseMotor("b13_st4_pm040")
pm1.Move(1000)
position1 = pm1.GetPosition()

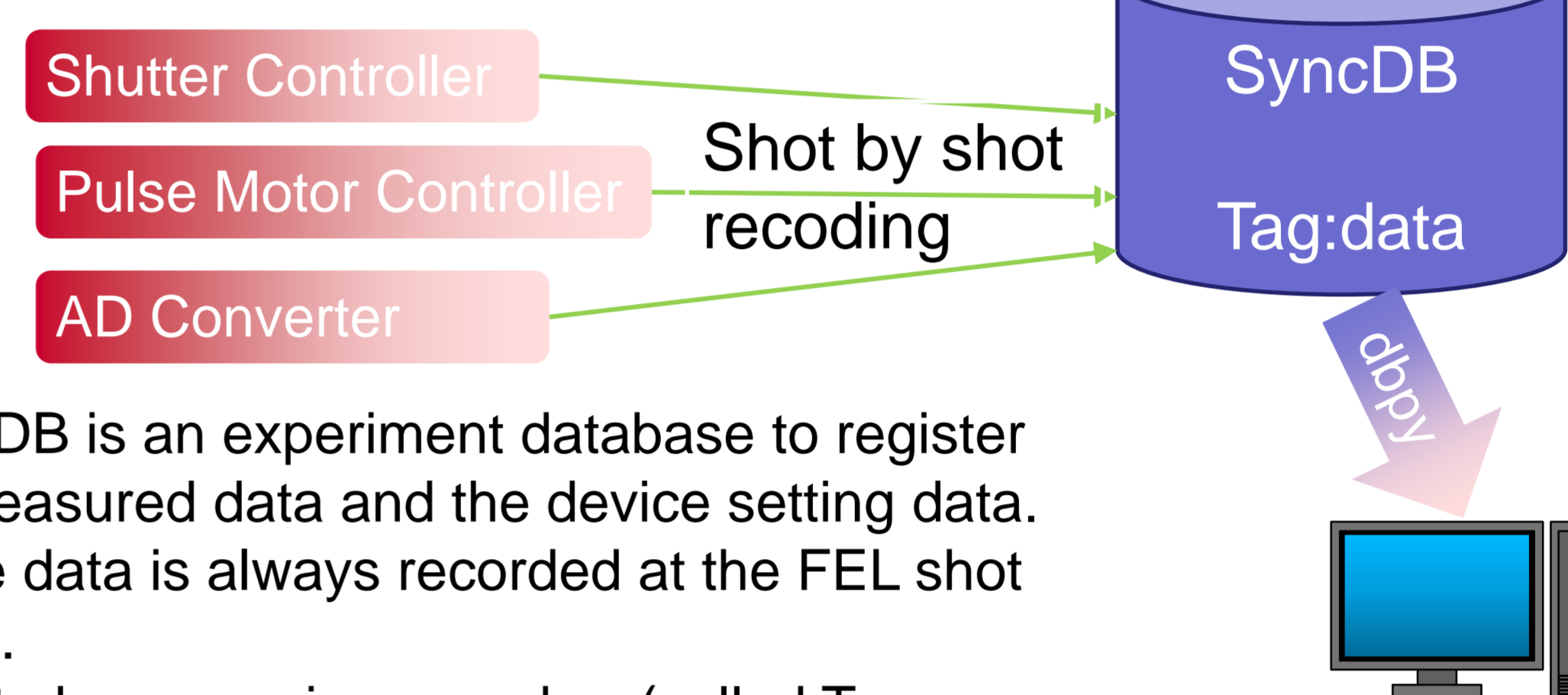
import ecpy
pm16c = ecpy.PM16C()
pm16c.Connect("hostname", port)
pm16c.Move(1, 1000)
position1 = pm16c.GetPosition(1)
```

*Example code: Pulse motor control program for both communications

Experiment Control API can communicate not only with MADOCA devices (SACLA standard control) but also with TCP/IP socket devices (local control). They are designed with similar interfaces in order to write the code with similar manner.

Data Access API (dbpy)

Device controllers



SyncDB is an experiment database to register the measured data and the device setting data. These data is always recorded at the FEL shot timing.

All data have a unique number (called Tag number with 32 + 32 bits integer) corresponding to each FEL shots. In addition to shot by shot data, Run information (number of shots, start / end tag, detector info, comments and so on) is also recorded.

Python module "dbpy" can retrieve the experiment data by using Tags or Run number.

We can get:

- Shutter state (open/close)
- FEL intensity
- FEL photon energy
- Motor position
- Run information
- Photo diode signal intensity
- ...

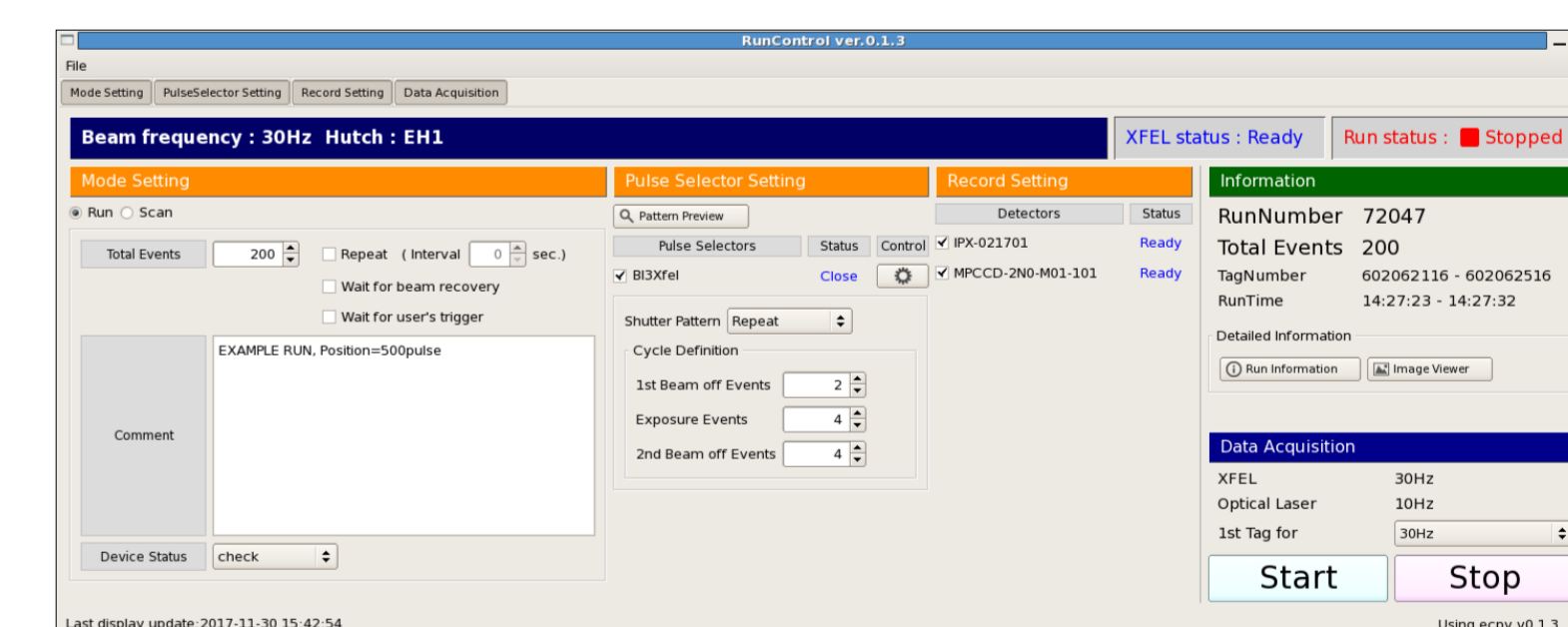
```
import dbpy
b1 = 2
run = xxxxx
tagHi = dbpy.read_hightagnumber(b1, run)
tagList = dbpy.read_taglist_byrun(b1, run)
data = dbpy.read_syncdatalist("xfel_b1_2_shutter_1_open_valid/status", tagHi, tagList)
```

*Example code: Get the list of FEL shutter state on BL2 and Run number xxxxx

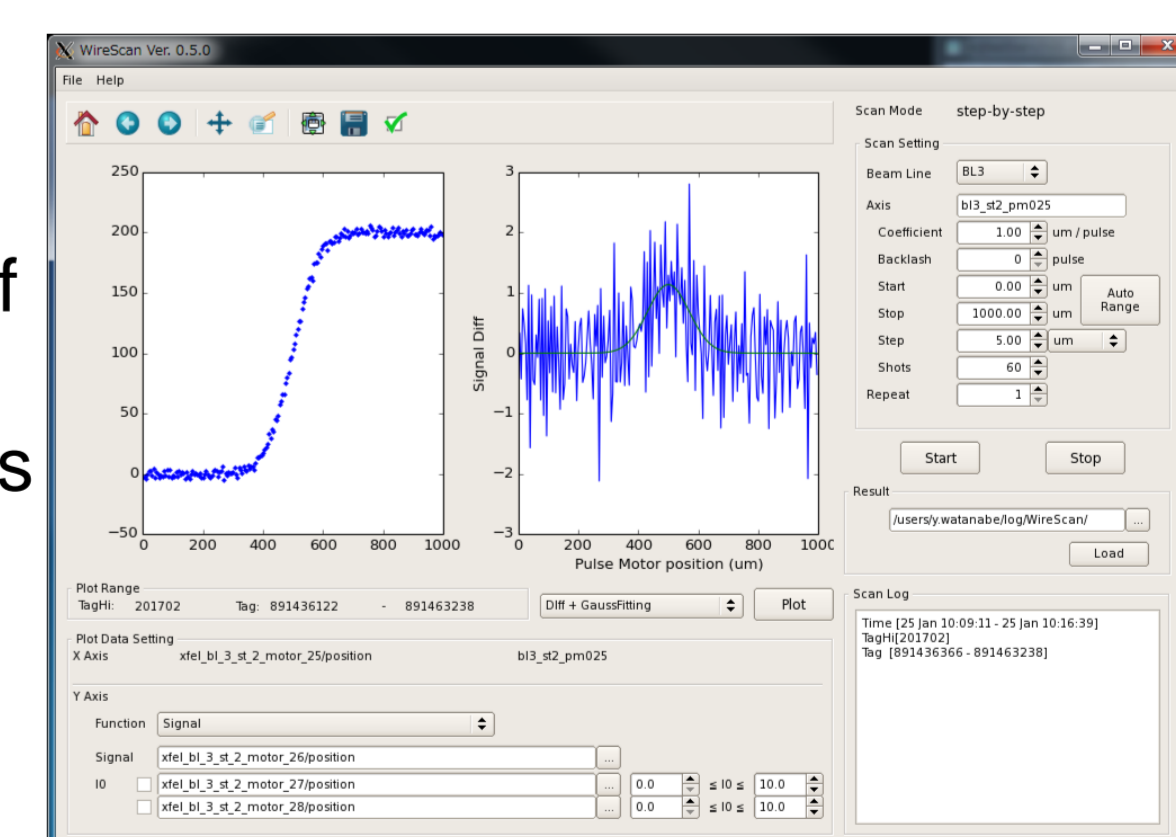
Application examples

Using APIs, we can make from small scripts to large programs for efficiency and quality of the experiment. Several applications are indicated below.

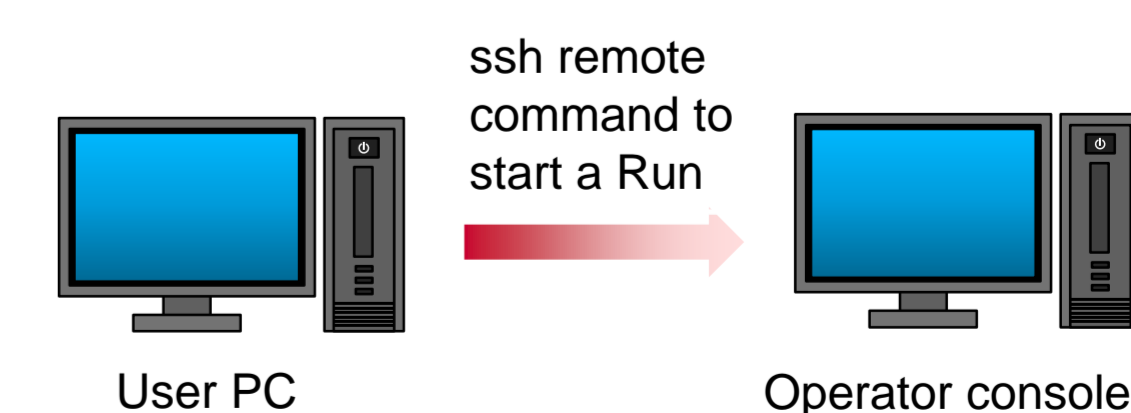
- Run control
Standard facility application to control the Run. It is possible to modify the code for the individual experiment.



- Wire Scan program
In order to measure the beam diameter of focused FEL, wire scan (knife edge method) is performed. The program scans the pulse motor and analyze photo diode instantly.



- Remote Run from user PC
When user PC is controlling the instrument or detector, the Run is started via network by remote command.



- Safety start the Run
Add routine to check the sample position for high power laser.
- The function to keep the Run waiting for external trigger
- ...

We welcome your ideas!

Reference Manual

Online help: Both Python modules can refer the online help by *help()* function on interactive mode.
ecpy: To be prepared
dbpy: <http://xhpcfef.hpc.spring8.or.jp/manuals/offline/index.html#DataAccessUserAPI>

Example codes

ecpy: [/prj/SACLATool/example](http://prj/SACLATool/example) (access from opcon)
dbpy: <http://xhpcfef.hpc.spring8.or.jp/document/examples/>