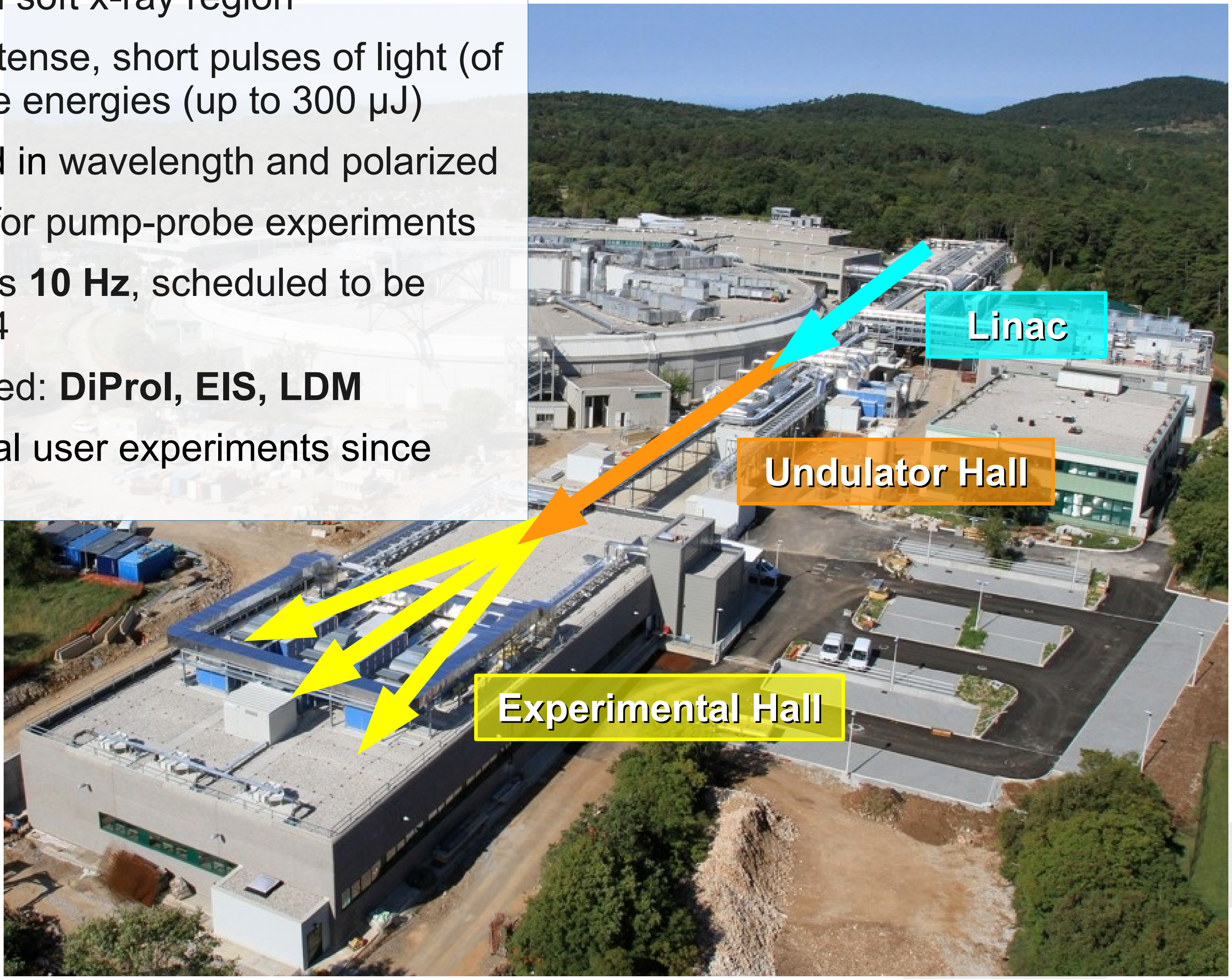
Elettra
Sincrotrone
Trieste

A COMMON SOFTWARE FRAMEWORK FOR FEL DATA ACQUISITION AND EXPERIMENT MANAGEMENT AT FERMI

R. Borghes, V. Chenda, A. Curri, G. Kourousias,
M. Lonza, G. Passos, M. Prica, R. Pugliese

FERMI overview

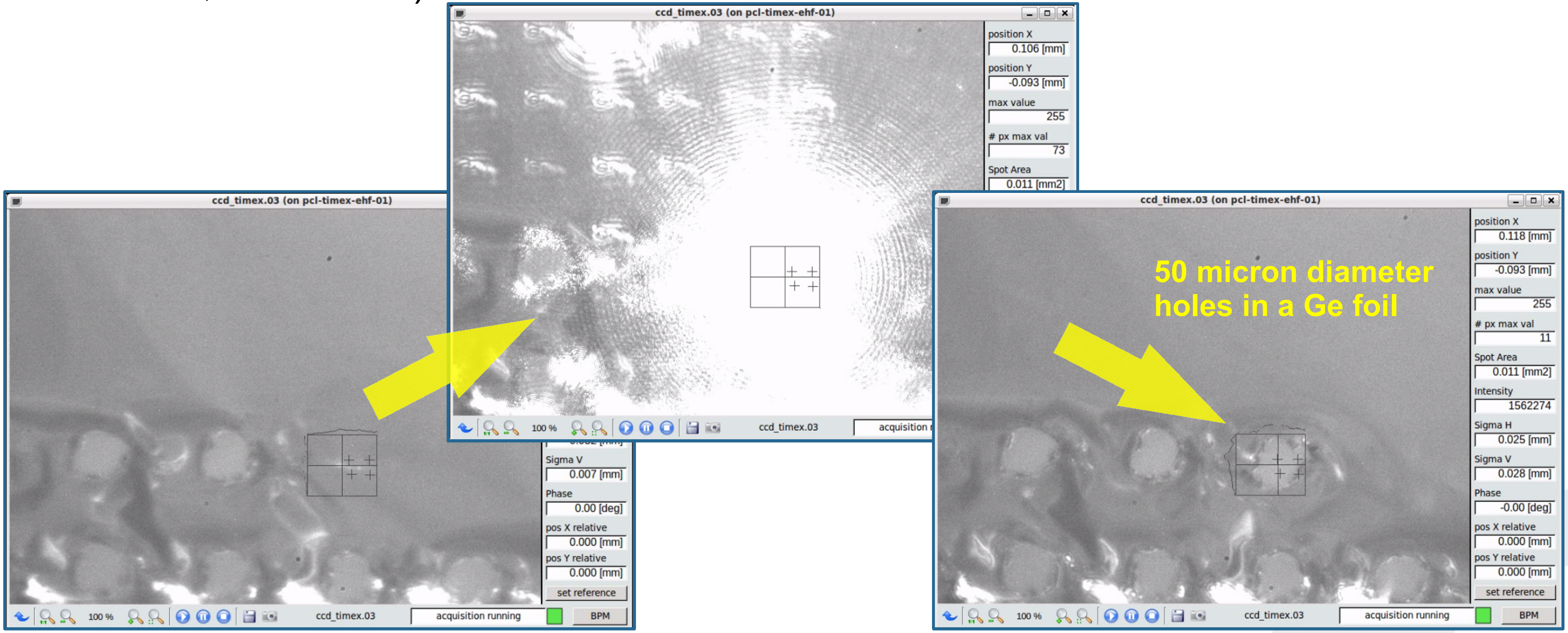
- **FERMI** is a seeded Free Electron Laser operating in the extreme ultraviolet and soft x-ray region
- the machine produces intense, short pulses of light (of the order 100 fs) and pulse energies (up to 300 μJ)
- light source can be tuned in wavelength and polarized
- a user laser is available for pump-probe experiments
- the pulse repetition rate is **10 Hz**, scheduled to be increased to 50 Hz in 2014
- three end-stations installed: **DiProI, EIS, LDM**
- **FERMI** is open to external user experiments since December 2012



Elettra
Sincrotrone
Trieste

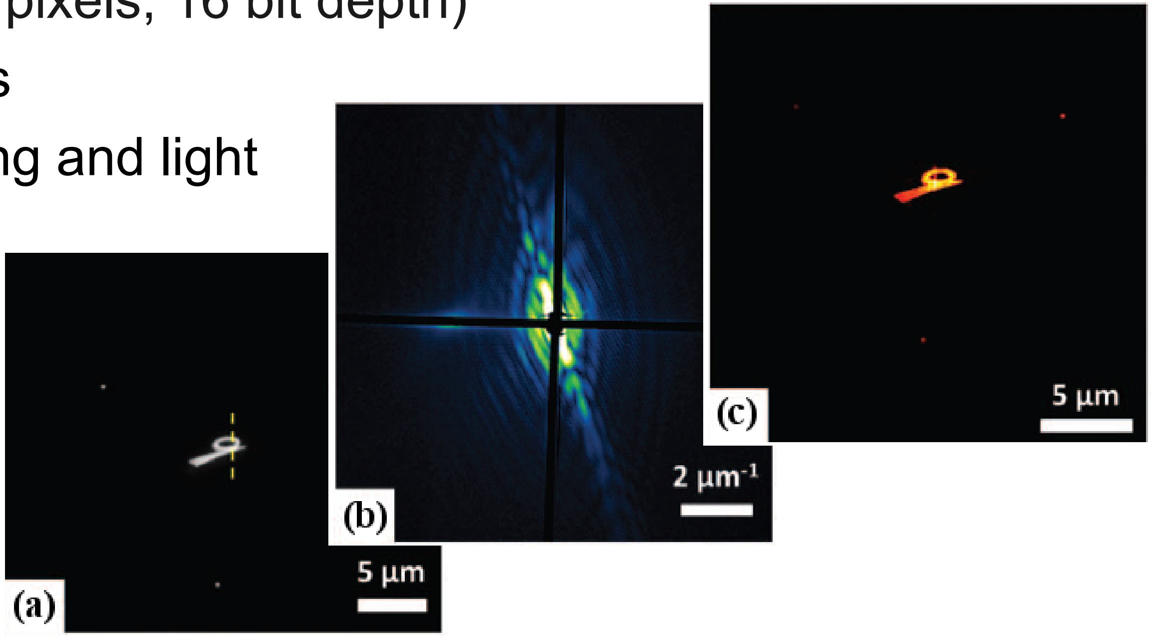
TIMEX end station

- probing fundamental properties of dense matter under extreme thermodynamic conditions
- 2D sample scan, pump & probe experiments
- heavy use of FEL wavelength tuning
- absorption and transmission detectors acquired by a CAEN digitizer (1 GS/s, 10 bits, 8 channels)



DIPROI end station

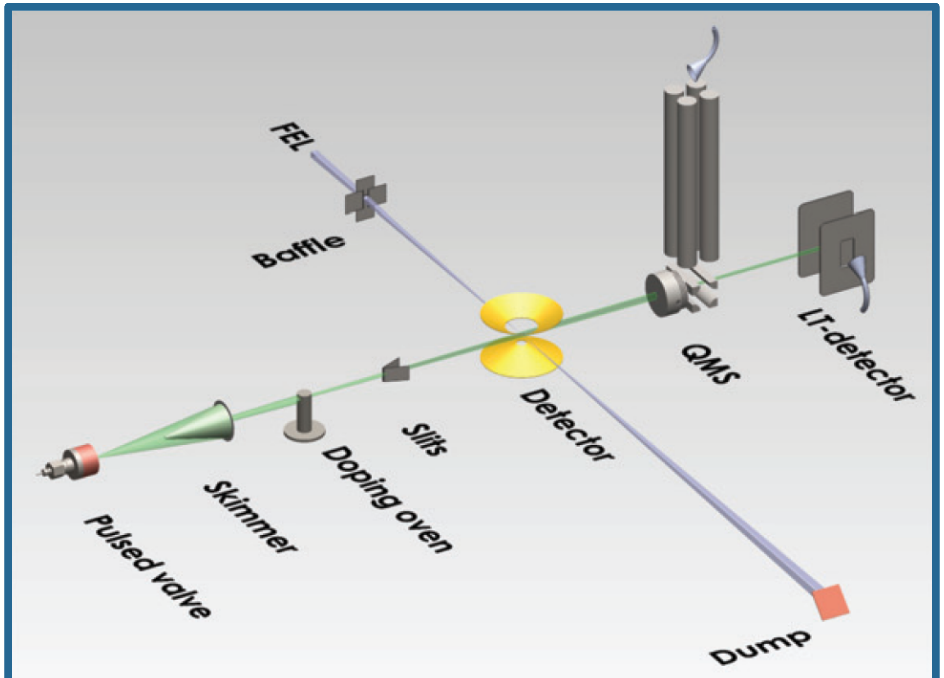
- single shot Coherent Diffraction Imaging experiments
- diffraction patterns are measured by a Princeton CCD (2048x2048 pixels, 16 bit depth)
- pump & probe experiments
- uses FEL wavelength tuning and light polarization

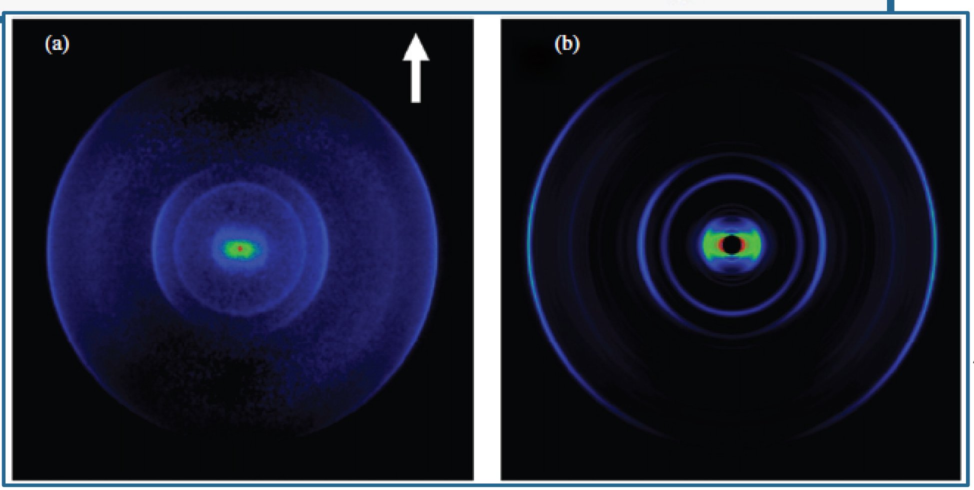


* Images from
F. Capotondi et al., "Coherent imaging using seeded free-electron laser pulses with variable polarization: First results and research opportunities", Review of Scientific Instruments, Vol. 84 - 5 (2013)

Elettra Sincrotrone Trieste

LDM end station





- Low Density Matter investigations
- a pulsed valve provides a jet of atomic, molecular and cluster targets
- pump & probe experiments
- uses FEL wavelength tuning and light polarization
- Velocity Map Imaging spectrometer based on a sCMOS Andor R Neo camera (2560x2160 pixels, 16 bit depth)
- Time Of Flight mass spectrometer based on a CAEN digitizer (VX1751, 1 GS/s, 10 bits, 8 channels)
- data throughput ~120 MB/s

* Images from:
V. Lyamayev et al., “A modular end-station for atomic, molecular, and cluster science at the low density matter beamline of FERMI”, J. Phys. B: At. Mol. Opt. Phys. Vol. 46 - 16 (2013)

Starting the design

FUNDAMENTAL REQUIREMENTS

1) Data must be **acquired and tagged** with the corresponding FEL pulse identification number (bunchnumber)

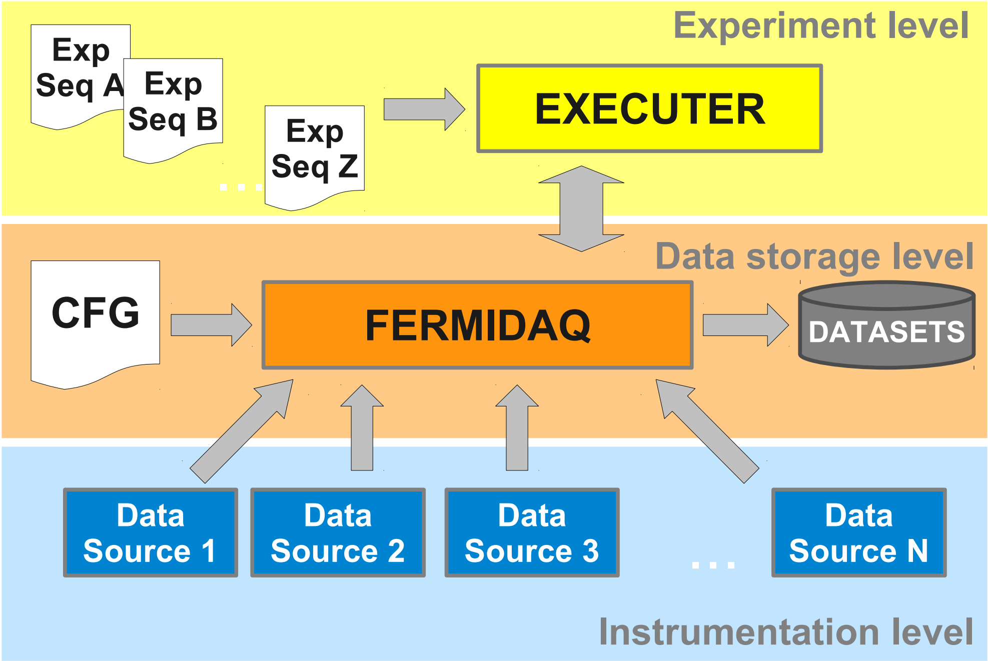
2) Number and type of data sources continuously change, the acquisition framework **should be easily configurable**

3) To fully meet the users experimental requirements the framework **should allow for easy adaptation** and implementation of new experimental procedures and sequences

4) Keep it **simple and reusable**

5) Development based on TANGO

System overview



The development has been broken up in three logical levels:

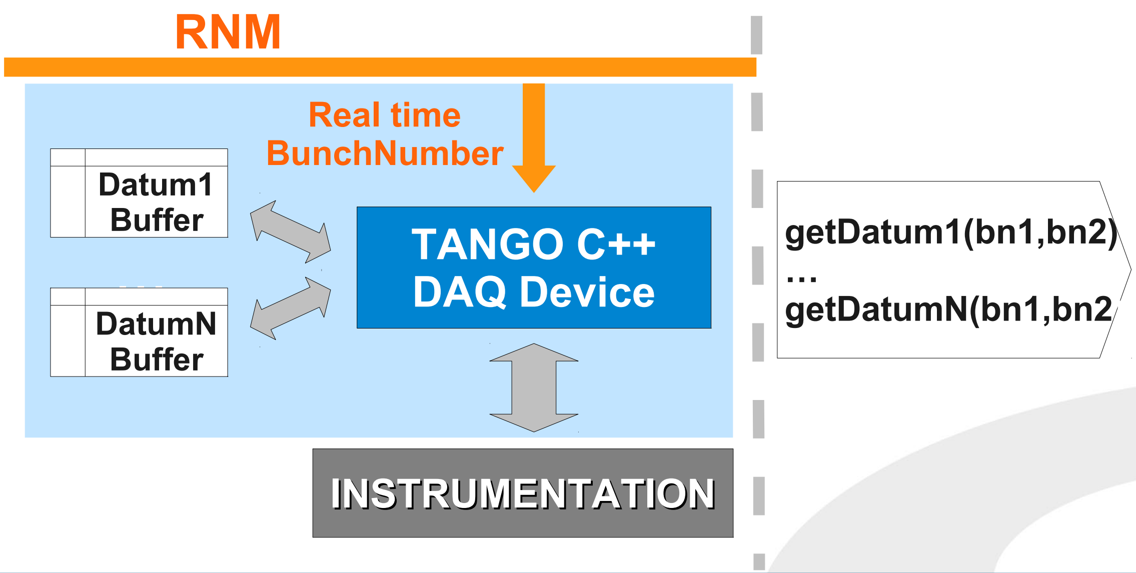
- at EXPERIMENT level there is the script engine **EXECUTER**, capable of implementing different experimental sequences in the form of Python scripts
- at DATA STORAGE level there is a single centralized, configurable software device, named **FERMIDAQ**, that organizes and stores data coming from multiple sources
- At INSTRUMENTATION level there are multiple shot-by-shot data acquisition devices, capable of buffering and exporting data tagged with the bunchnumber

Data source devices

For each FEL instrument, a C++ Tango device has been developed following few common guidelines:

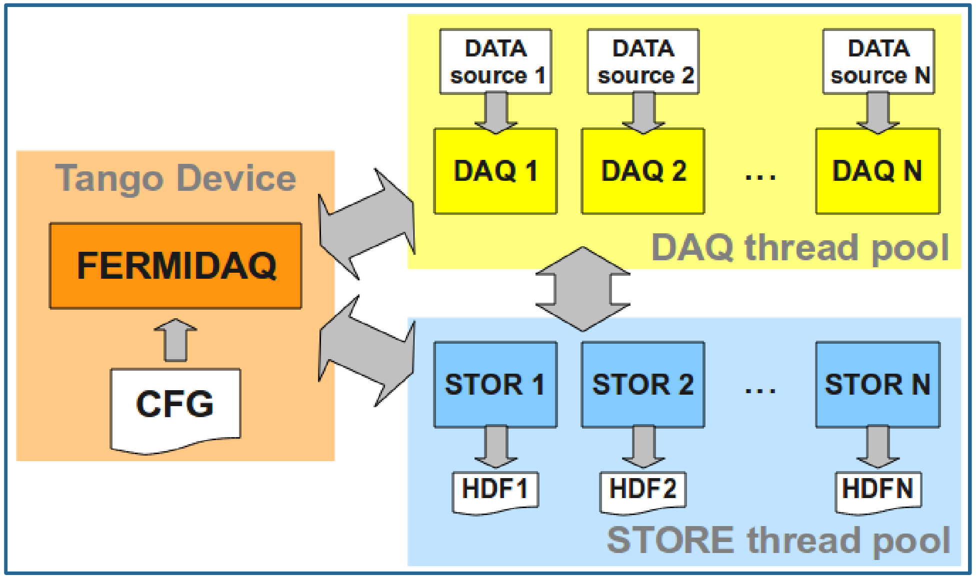
- shot-by-shot acquired data must be tagged with a bunchnumber and memorized in a local buffer
- a standard Tango command must exist for retrieving data relative to a range of bunchnumbers
- a Tango attribute containing the last acquired value must exist

- data comes from scientific and diagnostic instrumentation
- data type ranges from single floating values to 16-bit images
- the bunchnumber value is distributed using the RNM software infrastructure, which implements a data-transparent memory sharing among computers



FERMIDAQ device

Core of the data storage phase in the FEL experiment, it is a Python Tango device that continuously collects and saves shot-by-shot data and metadata.



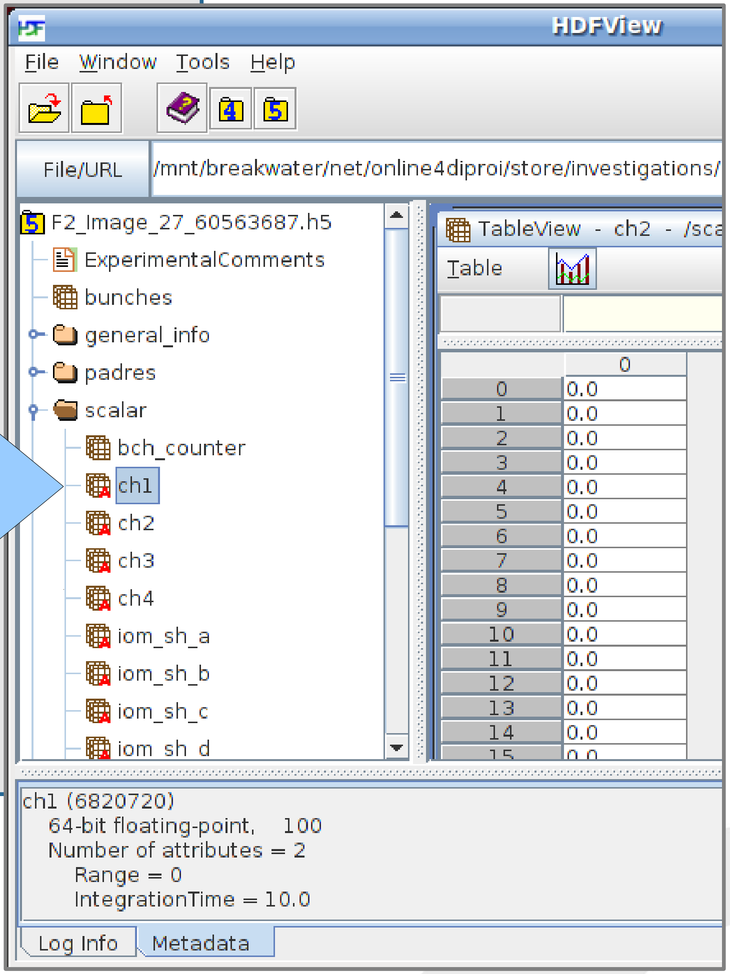
- multiple acquisition threads, one for each data source (61-71 data sources per experiment)
- data sources are defined in an XML file
- data saved in HDF5 archives
- multiple storage threads, one for each file
- after each file completion a “trigger action” can be performed (post processing)

The data storage system is based on the Gluster 3.3 filesystem with 250 TB homogeneously distributed over five Linux machines with a 10 GbE connection.

FERMIDAQ XML configuration

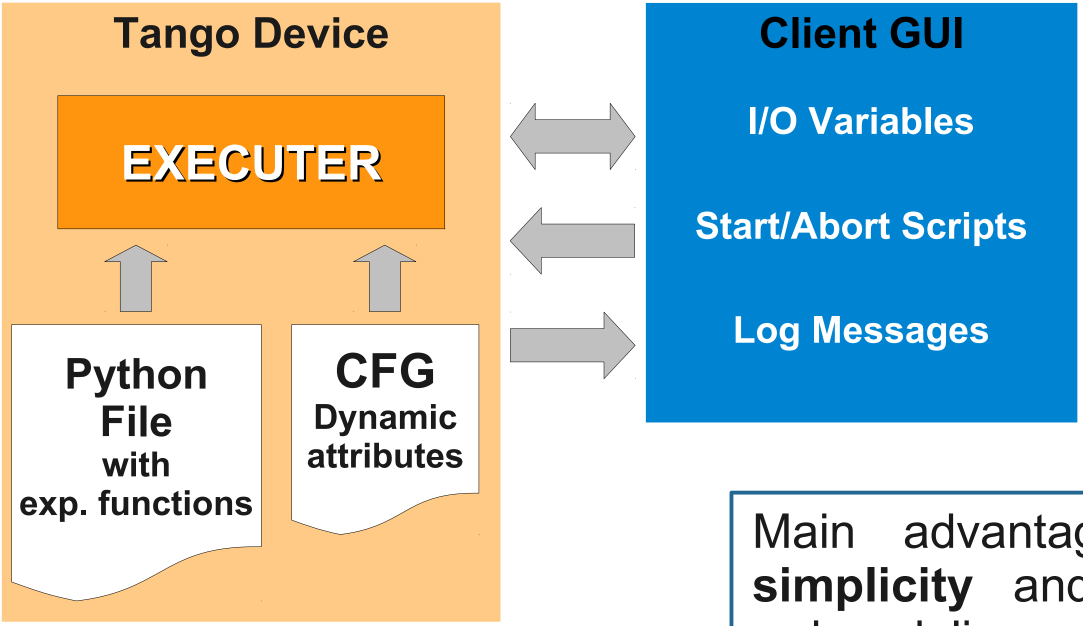
```
...
 <attribute type = 'meta'>
    <name>general_info/diproi_status/slu_delay_pos</name>
    <tango_attr>diproi/slu/corvus_dpi_02/Position1</tango_attr>
 </attribute>


 <!-- PICOAMMETER -->
 <attribute type='sync'>
    <name>scalar/ch1</name>
    <tango_attr>diproi/pam/01/CH1Data</tango_attr>
    <acq_class>ScalarAttribDaq</acq_class>
    <metadata name='IntegrationTime'>
    diproi/pam/01/IntegrationTime
    </metadata>
    <metadata name='Range'>
    diproi/pam/01/Range
    </metadata>
 </attribute>
...
```

will be saved to HDF5



EXECUTER device

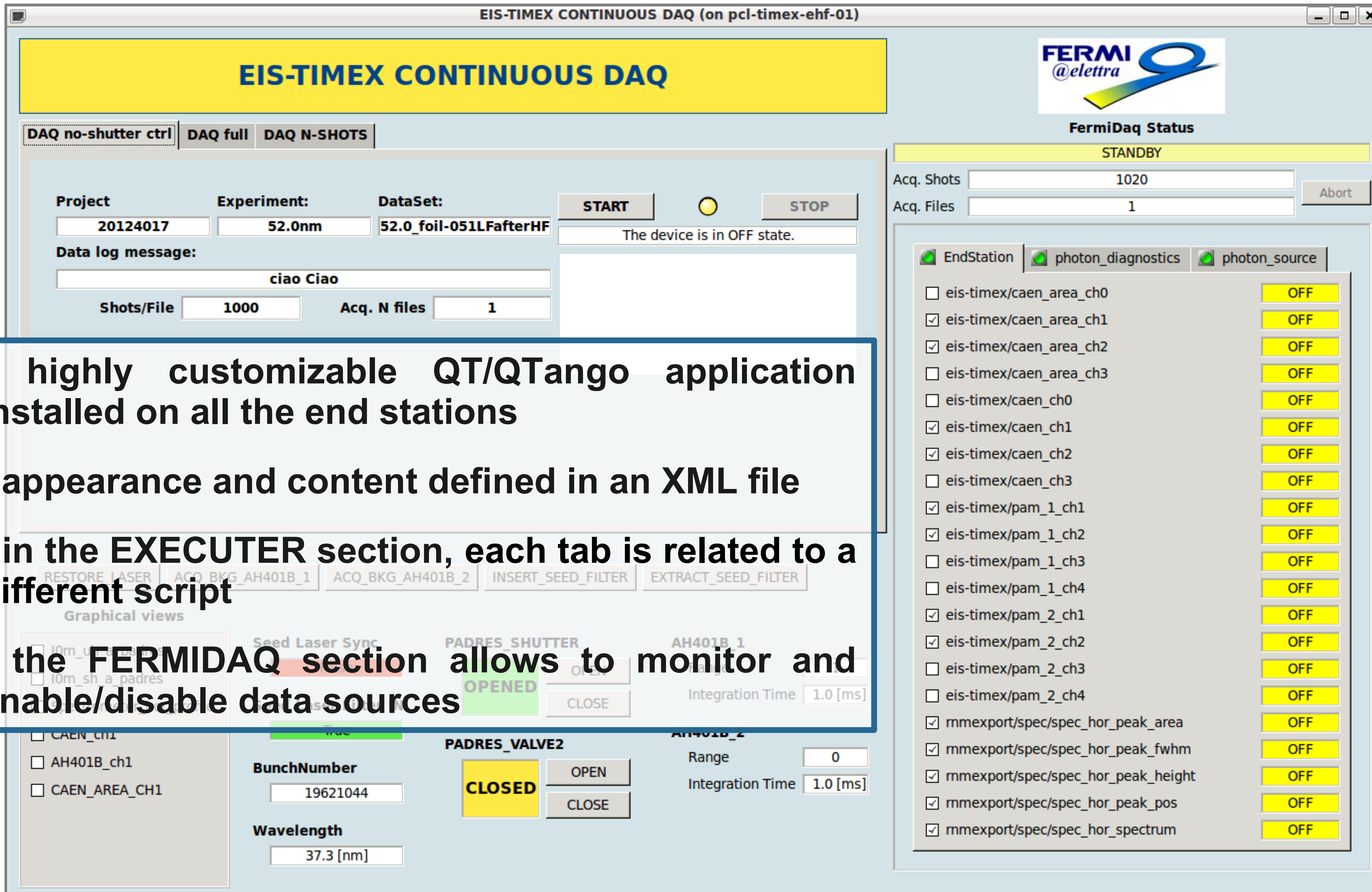
- the **EXECUTER** application is a highly flexible Python Tango device, capable of executing generic external Python scripts
- experimental scripts are written in an external file
- input variables or result viewers can be dynamically added as dynamic Tango attributes



Main advantages of such approach are the **simplicity** and the **flexibility** that allowed for reduced lines of code and significant speedup of any new developments.

Elettra
Sincrotrone
Trieste

The graphical interface



- highly customizable QT/QTango application installed on all the end stations
- appearance and content defined in an XML file
- in the EXECUTER section, each tab is related to a different script
- the FERMIDAQ section allows to monitor and enable/disable data sources

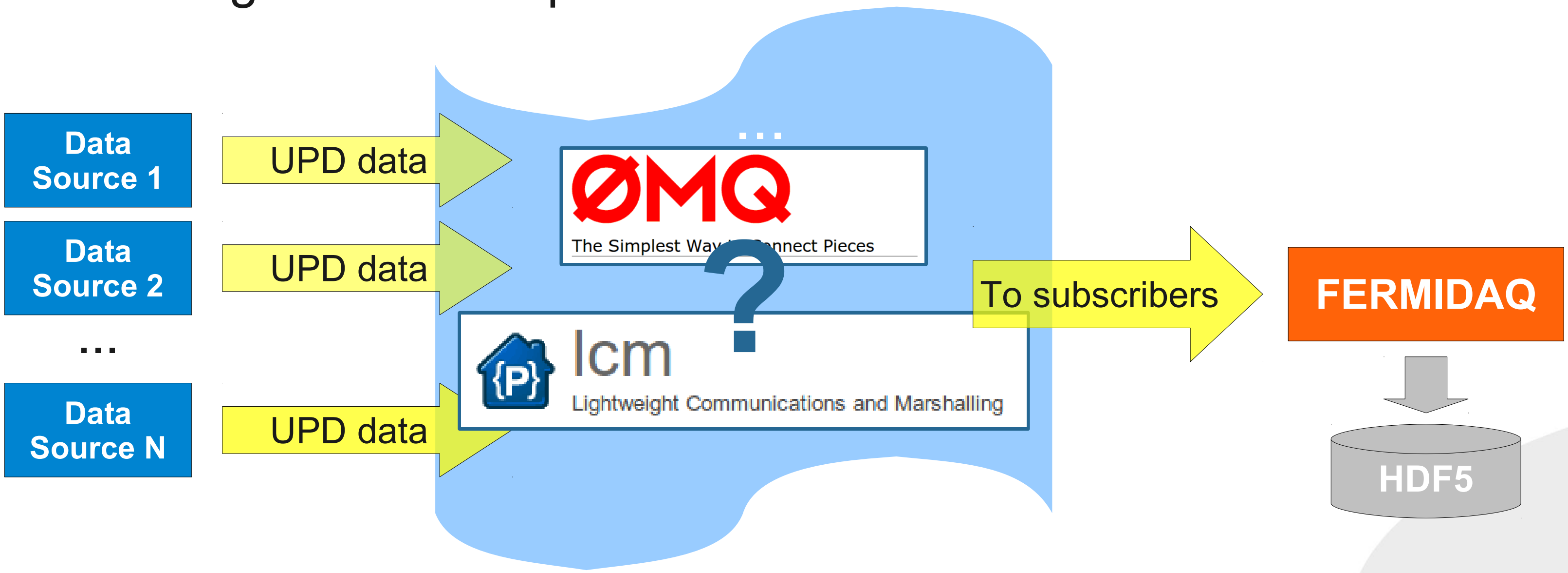
Future developments

3 new end-stations under construction:

- TIMER
- MAGNEDYN
- TERAFERMI

upgrade of the repetition rate of the facility to 50 Hz scheduled for 2014:

there is the need to minimize the communication overhead, by transforming the data acquisition devices to *active UDP sources*



DONKI TOOLS

Python applications that help users in doing *donkey works*

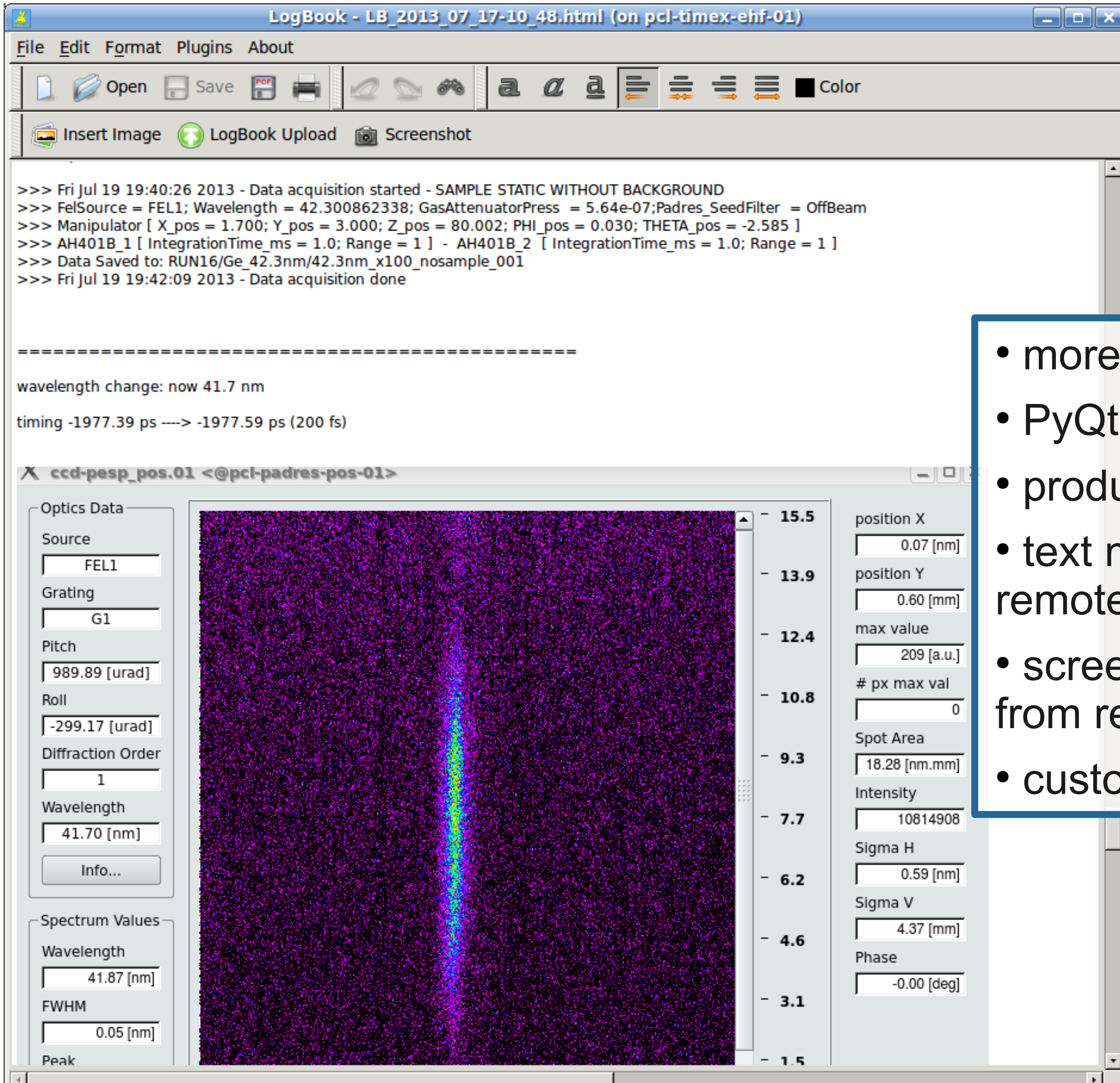
DONKILOG

DONKIMAGE

DONKIPY

Elettra
Sincrotrone
Trieste

DonkiLog

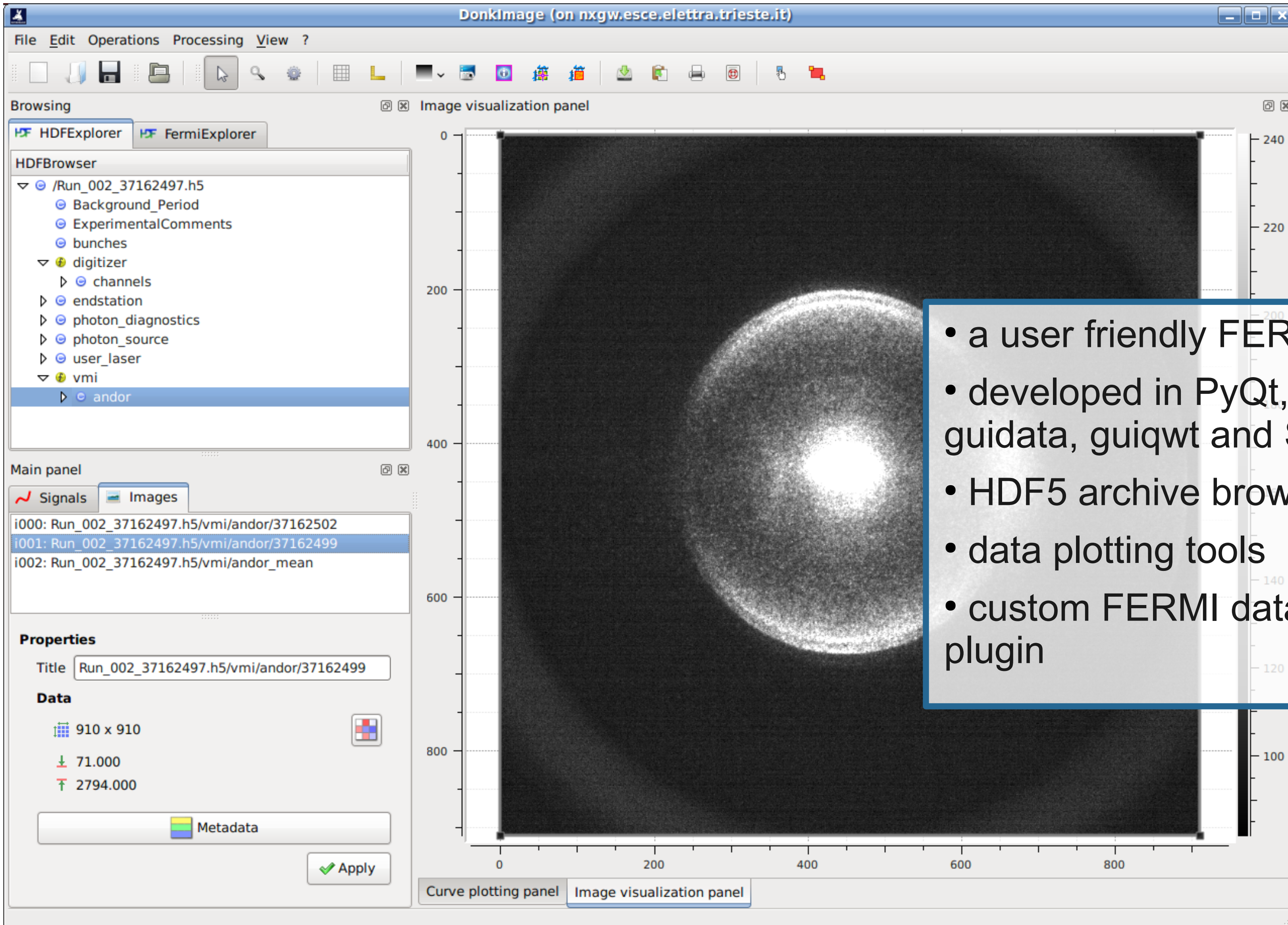


DONKILOG

- more than an electronic logbook
- PyQt based (linux and windows)
- produces HTML or PDF documents
- text messages can be added from remote (e.g. from a DAQ application)
- screenshots can be added locally or from remote
- custom plugins can be added

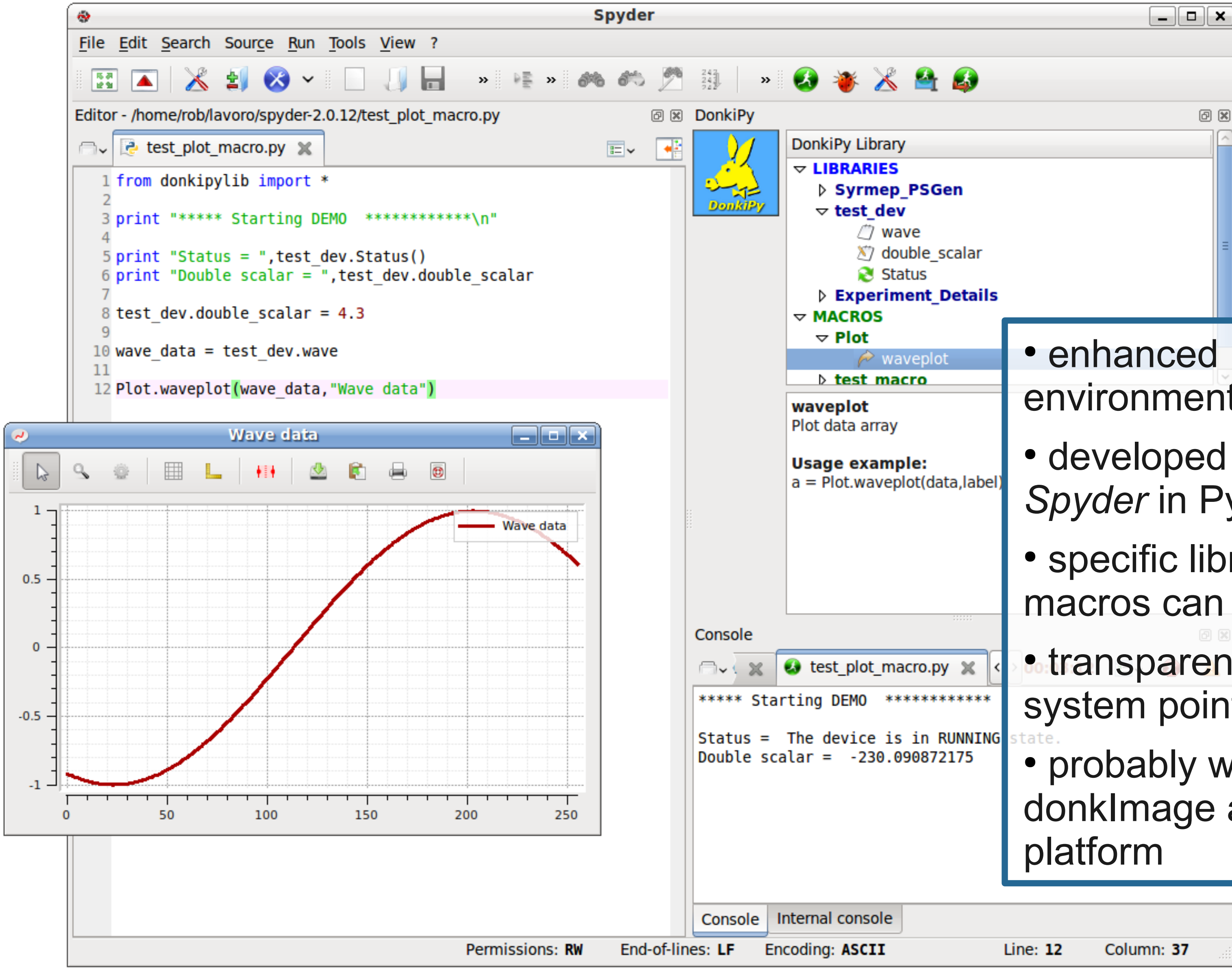
Elettra
Sincrotrone
Trieste

DonkImage



- a user friendly FERMI data viewer
- developed in PyQt, is based on guidata, guiqwt and Sift
- HDF5 archive browser
- data plotting tools
- custom FERMI dataset viewer plugin

DonkiPy



- enhanced Python scripting environment
- developed as a plugin for *Spyder* in PyQt
- specific library modules and macros can be added with docs
- transparent access to control system points
- probably will be merged with donkImage as data processing platform

Elettra
Sincrotrone
Trieste

www.elettra.eu