

## Developers' meeting – minutes

### MXCuBE project meeting, virtual ESRF, May 2022

#### Participants :

Marcus Oscarsson Daniele de Sanctis, Michael Hellmig, Oskar Aurelius, Andrey G, Isidoro Grispo Garcia, Lelicia, Roberto Bordghes Bo Yi, Meghdad Yazdi, Rasmus Fogh, Wenming, Alexander Dillman [with apologies for misspelling)

#### Developers' discussion

MO gave a talk to summarise the development situation.

- There is work in progress on making MXCuBE-wide recentring code, based on Gleb Bourenkovs CentringMath object
- Refactoring of the queue and queue\_entry is in progress. Elements include
  - Pydantic library to specify and document parameters
  - Separation of queue entries and queue model objects into separate files
  - A facility for adding new files to the relevant directory, which will be then automatically picked up and added to the (web) user interface.
- In a similar vein work has started on an AbstractProcessing object, that would allow launching all common processing functionality (including e.g. characterisation and thumbnail generation) through a single interface that is the same at all beamlines. A result is expected before summer.
- The Braggy web-based image viewer is being considered for closer integration with MXCuBE. This work would be integrated with a general framework for calling external processes, workflows etc. The top candidate is Celery, possibly with RabbitMQ underneath.

The later discussion included the following:

A release of the current state of mxcubecore is indicated as a large part of the refactoring is successfully completed, and there is a risk that new improvements might otherwise delay the release. It is agreed to aim for a mid-June release, and that MO will make one in a pull request.

The work on the abstract diffractometer will be for a later date; the current expectation is the end of summer 2022.

Upgrading/moving up to the release at individual beamlines is agreed to be in practice a matter for each beamline. There was a proposal to recommend that beamlines upgrade during their major shutdowns, but while this would be highly desirable it was not felt

possible to try to pressure beamlines into changing their scheduling. At the same time there was no obligation for external software providers (mostly Global Phasing) to support old or ancient MXCuBE versions.

The image viewer Braggy is at the point where it is relevant to consider attaching it more firmly to MXCuBE. Several groups (RB, AG, DdS) expressed intense interest in having an image viewer that was easily connected. The technical details remain to be settled, but the discussion tended towards having Braggy run as a separate, external program, which facilitates communication to MXCuBE. Some issues were getting images from MXCuBE, coordinating access permission, viewing images as they were acquired, viewing multiple images ('thick slices') for weak images, launching the viewer from MXCuBE, displaying annotations, display within MXCuBE as opposed to within a separate window, viewing of archive images. Braggy could be set up to be used also from the Qt interface, possibly through a Hardware Object. MO, DdS would check with ESRF if there were IP issues that might complicate the release of Braggy.

For reorganising the queue and workflow handling it is considered to make Processing HardwareObject, using the Celery framework. This is a generalised task queue, which can be attached to a message broker (RabbitMQ or Redis). It allows you to set up a Python Celery client for installation on your processing computer(s), and to launch processing jobs etc. in a uniform manner on different beamlines, with (relatively) simple configuration. There was a general approval both for the practicality and for the move to a modern code framework. Peter Keller suggested standardising remote procedure calls on Celery, and using it to replace Py4J (Global phasing) XMLRPC and other systems, in order to reduce the future maintenance burden. Coordination with ISPyB would be nice, but they are likely to make their own choices.

**ACTION:** PK will try to make a prototype.

The idea of code camps were viewed favourably, both for production and as a good way to integrate new members, but concrete proposals would depend on which features we wanted to take on.

## Developers' final summary

- It is agreed that some form of integration of MXCuBE with Braggy would be good.
- It is agreed that the work on a general framework for calling external processes should proceed, and also be considered as a possible replacement for other frameworks such as XMLRPC.
- A release of the (currently stable) version of MXCuBE is planned for mid-June.

## Steering committee summary

- The Steering committee is very impressed with the results achieved in the refactoring process, the mxcube core repository, the new branch organisation, and the organisation in abstract classes, as with the work of the AbstractDiffractometer and other working Groups. The adoption of the new release across all synchrotrons

is highly encouraged. Areas of particularly high priority for future work are fully automatic data collection, workflows and SSX both collection and processing. An integrated image viewer (such as Braggy) is indeed important, but the steering committee suggests also contacting other groups with image viewers, e.g. SLS.

- The existence of two different user interfaces (Qt and Web) is not seen as a problem – and is anyway unlikely to go away in the foreseeable future. If nothing else it should have the effect of keeping core and user interface cleanly separated.
- The steering committee is aware that developers resources for MXCuBE are coming under strain at several beamlines, and appreciate that this may slow down developments. It might be a good idea to concentrate on a few of the priority goals first.
- There are four new synchrotrons that have shown interest in joining MXCuBE: IHEP, AMSTO, Arinax, and NSPS. It is agreed to welcome all four, and to make a single new collaboration agreement to cover the accession of all four. From past experience delays are expected in getting this agreement past legal departments and the people with authority to sign agreements, and it is hoped that all parties will accept digital signatures. Wenming (NSPS) warns that the process may be unusually slow given the COVID quarantine in Shanghai, but is reassured that this is as expected: the signature process has been unusually slow whenever it has been tried.
- The next meeting will be in week 49, the week of December 5, at ESRF. Barring the unforeseen it will be an in-person meeting, with support for remote participation, particularly in some developers' sections. It is noted that this may clash with the BESSY users meeting, in which case participants from BESSY will (have to) participate remotely.