MXCuBE 3 web application: on the way to next generation experiment control

Mikel Eguiraun - mikel.eguiranu@maxiv.lu.se
NOBUGS 16
Outline

- MXCuBE project
- MXCuBE v3
  - Technologies in use
  - User interface
  - Demo video
- Next steps & work force
MXCuBE

- Macromolecular Xtallography Customized Beamline Environment
  - Started in 2005 at ESRF
  - Beamline control and data acquisition platform for running MX experiments

- Supported by the following partners: ESRF, Soleil, MAX IV, HZB, EMBL, Global Phasing Ltd, DESY, ALBA.
- Already tested software & builtin experience (many years + many people + many beamlines)
MXCuBE - Main Features

- Customizable for each beamline/facility
- Reuse of existing code for different beamlines; common solution for users
  - same or similar hardware devices
  - same or similar experimental procedures
- Hardware mockups available, testing and developing without equipment
- Current stable version based on PyQt
- Hide the complexity of the Hardware to the user (and to the developers...) thanks to the usage of the **HardwareObjects**
MXCuBE - HardwareObjects

- Self-contained piece of software linking devices and graphical interface
  - Through the HardwareRepository
- Configured through xml files
- A HO is not only hardware! Procedures/sequences etc
- Emitting signals to hardware objects, graphical elements
- Supported protocols: Tango, EPCIS, Sardana...

```python
class MicrodiffMotor(Device):
    def init(self):
        self.position_attr = self.addChannel({"type":"exporter",
                                             "name":"position"},
                                             self.motor_name)
                                             self.motor_name)
    def getPosition(self):
        return self.position_attr.getValue()
    def move(self, absolutePosition):
        self.position_attr.setValue(absolutePosition)
```

MotorOmega.xml

MicrodiffMotor.py
MXCuBE 3

- Beamline control and data acquisition as web application
- Reuse existing HardwareObjects maintaining compatibility with v2
- Modern web technologies for user interface
- Easier integration with LIMS (*laboratory information management system*)
- Maintenance and deployment (decoupling client/server)
- Remote access in a more *natural* way
- Feedback from user community

[https://github.com/mxcube/mxcube3](https://github.com/mxcube/mxcube3)
MXCuBE 3: Current status

- Under development but first data collection during Biomax commissioning (June 2016, MAX IV)
- Sample imaging and operation
- Basic data collection strategies
  - experiment queue configuration and execution
  - standard dc, characterisation, helical
- Screen mirroring / master-slave operation
- LIMS and sample changer integration very advanced
Backend

- Python **Flask** microwebframework:
  - web server made simple
  - extensions (database, login, ...)
  - easily adaptable to your needs while scalable
- **http request API**: rest-like
  - url for each function
  - Simple to add new features without changing existing ones
- Flask **socketio** for sending HO messages
  - server-client bi-directional communication, websocket based
- Reuse the existing Hardware Objects
Frontend REACT

- User interface has been redesigned (compared to MXCuBE v2)
- Javascript/React library (Facebook)
- Only for the user interface (the V in MVC)
- UI is described as a collection of components
  - Different components programmed independently
- Reusing existing code when the layout changes
Frontend REDUX

- **Redux** application architecture/pattern
  - Predictable state container for JavaScript apps
  - Changes on the internal state in a single place
  - Unidirectional data flow
  - Clear and safe development
User Interface

- Two main operation modes:
  - Automatic:
    - multiple sample selection and configuration
    - Workflows (predefined sequence of automatic operations)
  - Manual:
    - sample by sample
    - manual operation and configuration
Sample Grid
Data Collection
Demo video
Next steps

- Continue the development
  - Finish LIMS and sample changer interfaces
  - Polish the interface (icons, buttons, minimal layout changes)
- Debugging
- Performance improvements
- Additional views
- Ready for users
MXCuBE3 People

**MAX IV:** M. Eguiraun, A. Milan-Otero, J. Nan, F. Bolmsten, M. Thunnissen, V. Hardion, D. Spruce

**ESRF:** M. Guijarro, M. Orkarsson, A. Beteva, D. de Sanctis, G. Leonard, J. Meyer, A. Gotz

Supported by:

- MXCuBE collaboration
- MAX IV MX and KITS teams
- ESRF BCU and SB teams
Thanks for your attention!