ESRF | The European Synchrotron

daiquiri

web based UI framework for data acquisition and beamline control

Stu Fisher
ESRF
Software Engineer / BCU (UI Coordinator)

What is daiquiri?

Provides a modular UI framework for acquisition and beamline control

Does not provide a scan engine

- Actors / Scan data interface

Does not provide a controls system

- Thin hardware layer

Connected via interfaces

Daiquiri Projects

daiquiri

python server

flask rest
socketio

daiquiri-ui

javascript ui

react
redux

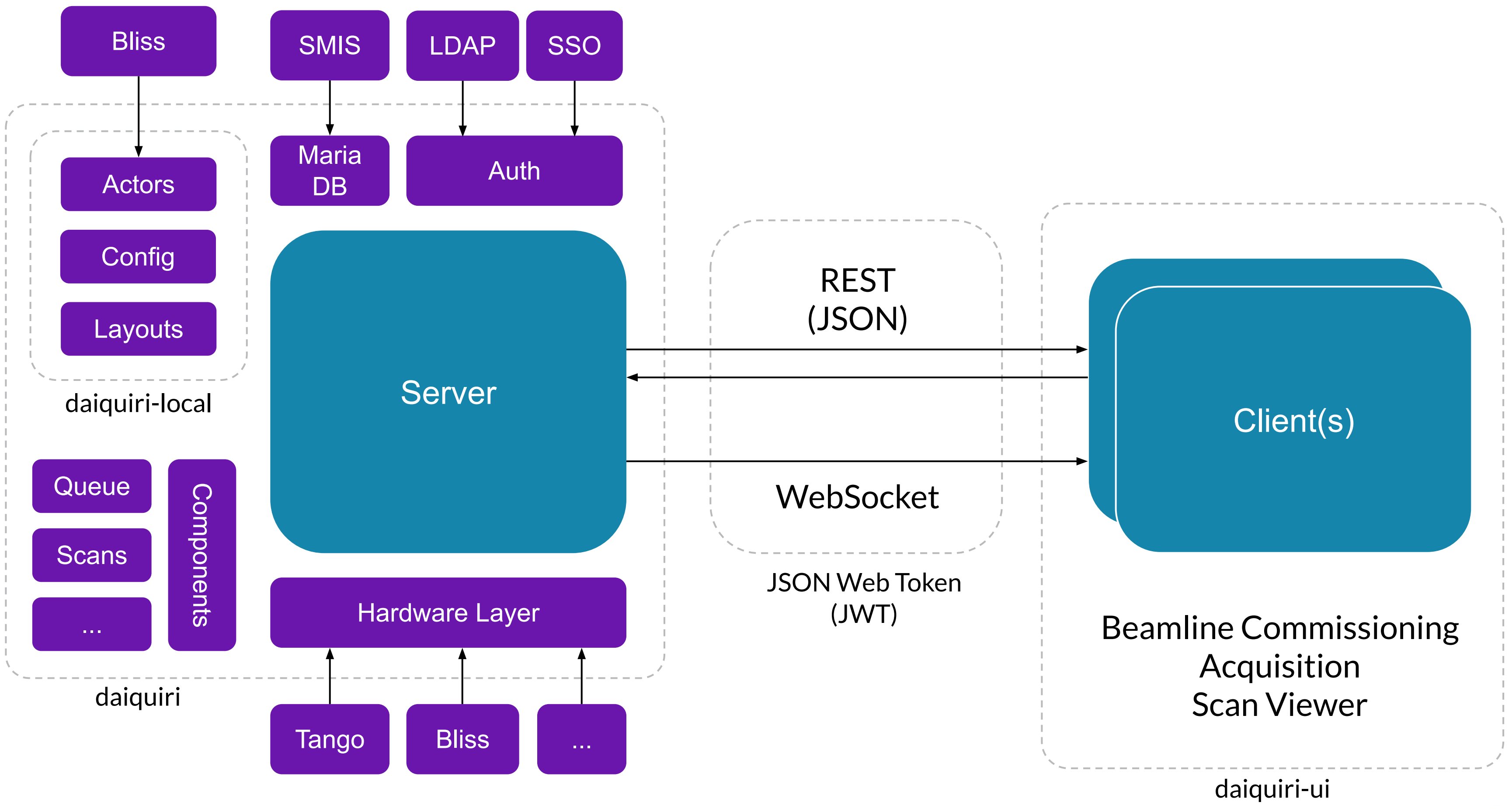
daiquiri-local

local beamline specific implementation

cookiecutter project

wrapper scans
config files

Architecture



Steal as many ideas as possible from: MXCuBE/3 (qt/web), GDA (rcp), SynchWeb (web)

Why?

UI is completely decoupled

Server REST resources can be consumed by other client

  Web, Command line, ...

Platform independent (dependencies)

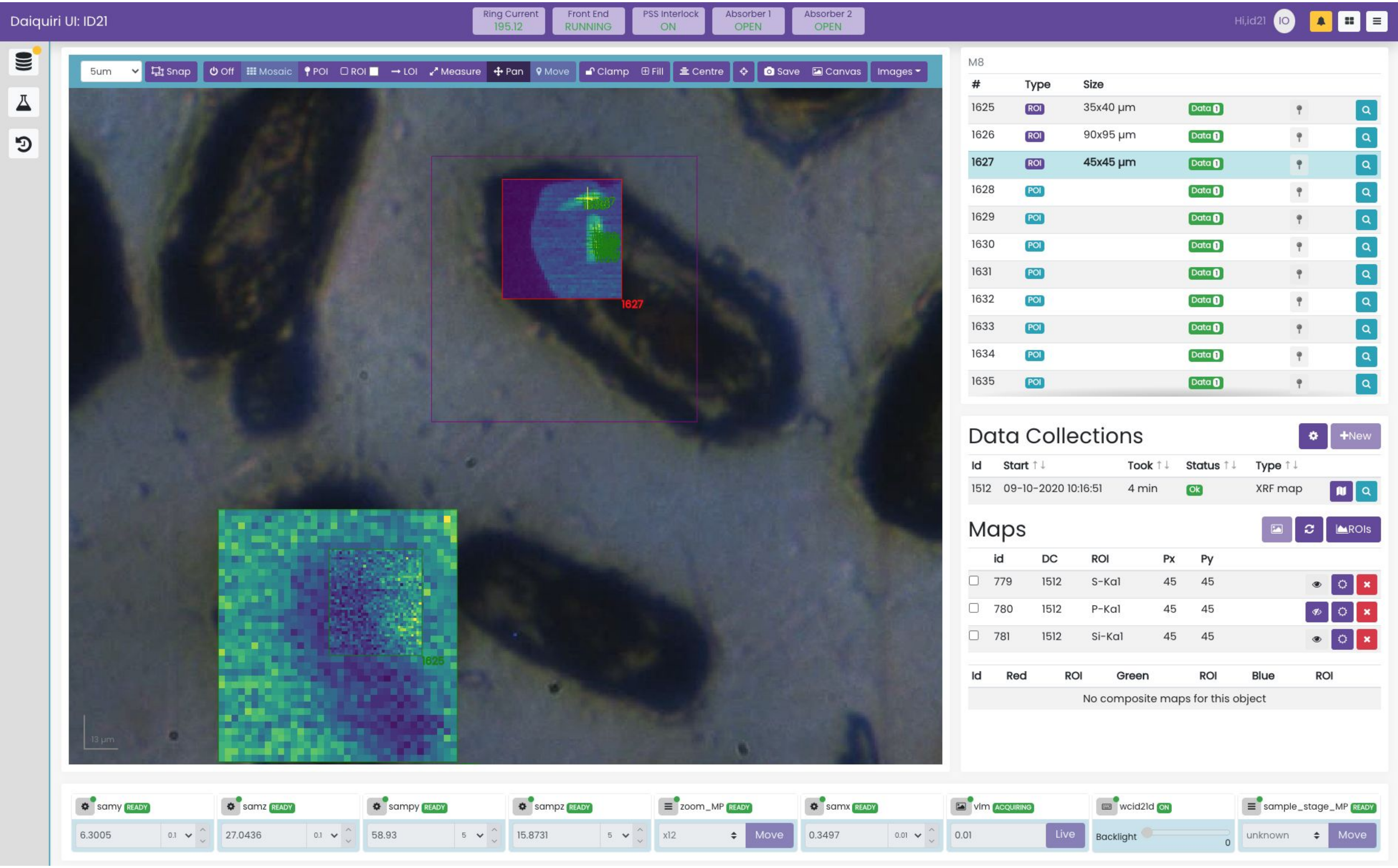
The server / client api is well documented

If a client crashes, everything is stored on the server

  If server crashes most information is persisted to db

Remote access (and monitoring).

These technologies are designed to be responsive with high latency (c.f. vnc, nx, guacamole)



M8

#	Type	Size			
1625	ROI	35x40 µm	Data 1		
1626	ROI	90x95 µm	Data 1		
1627	ROI	45x45 µm	Data 1		
1628	POI		Data 1		
1629	POI		Data 1		
1630	POI		Data 1		
1631	POI		Data 1		
1632	POI		Data 1		
1633	POI		Data 1		
1634	POI		Data 1		
1635	POI		Data 1		

Data Collections

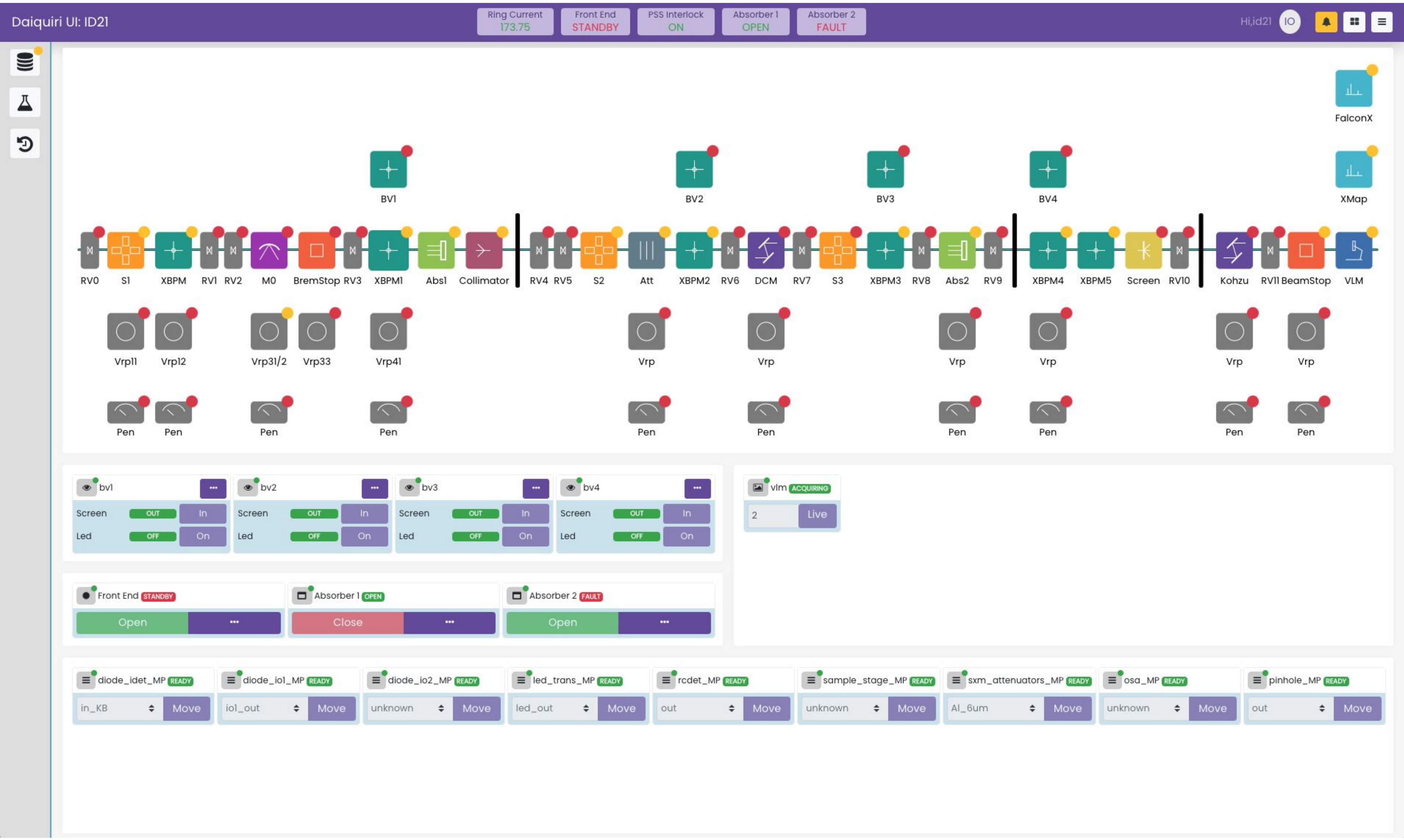
Id	Start	Took	Status	Type
1512	09-10-2020 10:16:51	4 min	Ok	XRF map

Maps

	id	DC	ROI	Px	Py
☐	779	1512	S-Kal	45	45
☐	780	1512	P-Kal	45	45
☐	781	1512	Si-Kal	45	45

Id	Red	ROI	Green	ROI	Blue	ROI
No composite maps for this object						

samy	READY	6.3005	0.1
samz	READY	27.0436	0.1
sampy	READY	58.93	5
sampz	READY	15.8731	5
zoom_MP	READY	x12	Move
samx	READY	0.3497	0.01
vlm	ACQUIRING	0.01	Live
wcid21d	ON	Backlight	0
sample_stage_MP	READY	unknown	Move



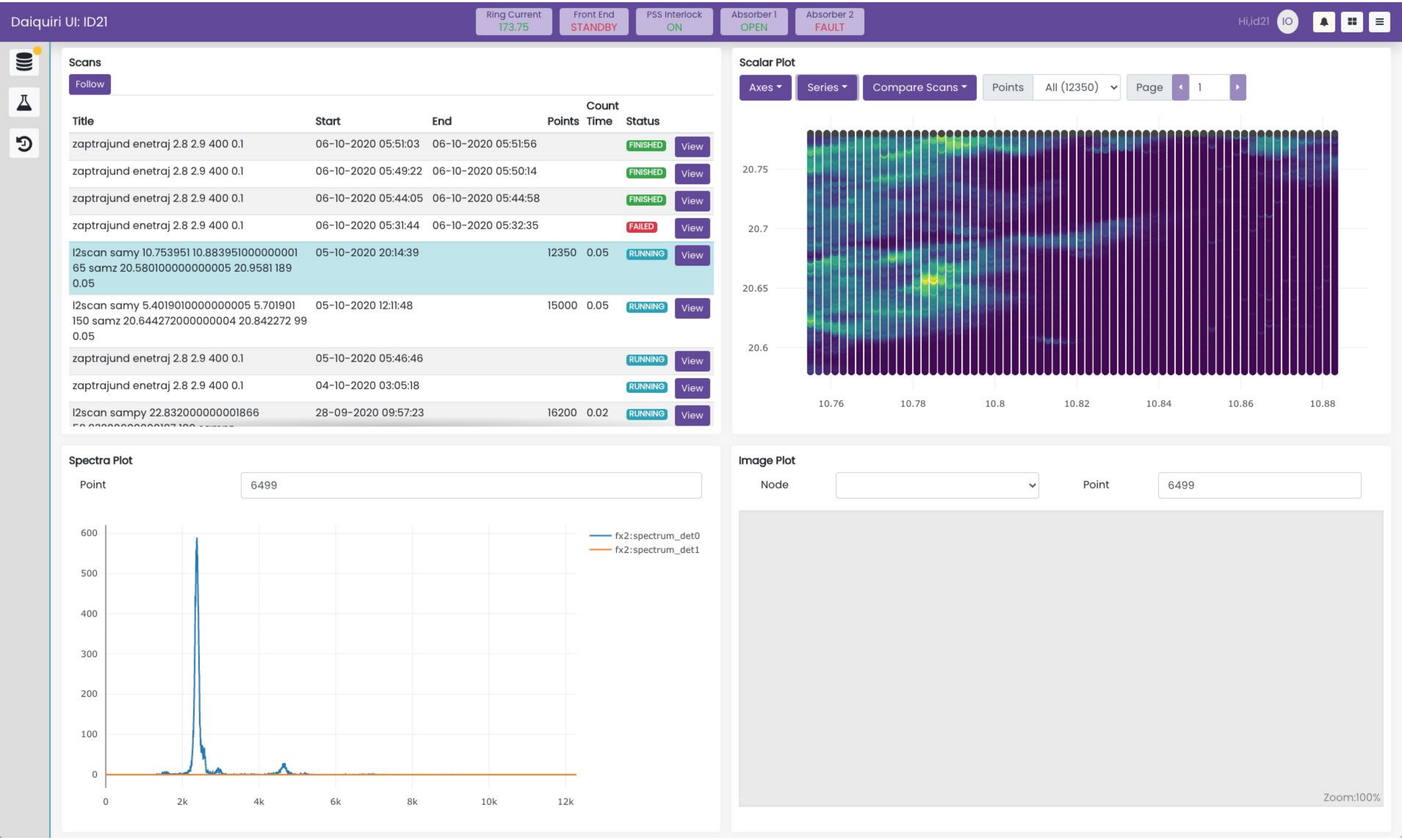
Ring Current 173.75
Front End STANDBY
PSS Interlock ON
Absorber 1 OPEN
Absorber 2 FAULT

	bv1	bv2	bv3	bv4
Screen	OUT	OUT	OUT	OUT
Led	OFF	OFF	OFF	OFF

vlm ACQUIRING: 2

Front End STANDBY
Absorber 1 OPEN
Absorber 2 FAULT

diode_idet_MP READY	in_KB
diode_io1_MP READY	io1_out
diode_io2_MP READY	unknown
led_trans_MP READY	led_out
rcdet_MP READY	out
sample_stage_MP READY	unknown
sxm_attenuators_MP READY	Al_6um
osa_MP READY	unknown
pinhole_MP READY	out


Daiquiri UI: ID21

Ring Current 173.75 | Front End STANDBY | PSS Interlock ON | Absorber 1 OPEN | Absorber 2 FAULT

Hi,id21 IO

Scans

Follow

Title	Start	End	Points	Count Time	Status	
zaptrajund enetraj 2.8 2.9 400 0.1	06-10-2020 05:51:03	06-10-2020 05:51:56			FINISHED	View
zaptrajund enetraj 2.8 2.9 400 0.1	06-10-2020 05:49:22	06-10-2020 05:50:14			FINISHED	View
zaptrajund enetraj 2.8 2.9 400 0.1	06-10-2020 05:44:05	06-10-2020 05:44:58			FINISHED	View
zaptrajund enetraj 2.8 2.9 400 0.1	06-10-2020 05:31:44	06-10-2020 05:32:35			FAILED	View
l2scan samy 10.753951 10.88395100000000165 samz 20.580100000000005 20.9581 189 0.05	05-10-2020 20:14:39		12350	0.05	RUNNING	View
l2scan samy 5.4019010000000005 5.701901 150 samz 20.644272000000004 20.842272 99 0.05	05-10-2020 12:11:48		15000	0.05	RUNNING	View
zaptrajund enetraj 2.8 2.9 400 0.1	05-10-2020 05:46:46				RUNNING	View
zaptrajund enetraj 2.8 2.9 400 0.1	04-10-2020 03:05:18				RUNNING	View
l2scan sampy 22.832000000001866	28-09-2020 09:57:23		16200	0.02	RUNNING	View

Scalar Plot

Axes | Series | Compare Scans | Points: All (12350) | Page: 1

Spectra Plot

Point: 6499

Image Plot

Node:

Point: 6499

Zoom:100%

Daiquiri (python)

REST / SocketIO / gevent (Bliss)

- flask-restful
- python-socketio

Input and output marshalling (validation) + shared schema

- marshmallow, marshmallow-jsonschema

Automated API documentation

- flask-apispec

Component architecture

- Load components relevant to a beamline
- Scans, hardware, 2d view

Interfaces

- Scan engine and hardware components

Concepts

Authentication / authorisation

- **Know who is logged in and whether to elevate privileges**
  - Limit access to specific hardware, scans, layouts to staff
- **Because a session is selected can automatically enforce data policy**

Multiple sessions can be logged in

- **Only one session can control the beamline at a time**
  - System of control request / response. Staff can always take control
- **Session mirroring**

Queue

- **Automated control of the beamline (e.g. overnight)**

Metadata

- **User office information**
- **Redis/Bliss data is transient**

Daiquiri UI (javascript es6)

Generic javascript client for daiquiri

Data acquisition and real time monitoring

- **Dynamic layout renderer**
- **Common panels**
  - Queue, Samples, History, Monitoring
  - Sessions, Logging, Chat

react / redux

react-bootstrap

fabricjs

react-jsonschema-form

sass

Layout

Defined in yaml

Layout:

- **row, col, container**
- **tab, panel**

Components (chunk and lazy load):

- **hardware**
- **synoptic**
- **console (xtermjs)**
- **file editor (acejs)**
- **twod**
- **scantable**
- **scanplot0,1,2d**
- **…**

Templating

```yaml
name: Simple Layout
description: A simple layout
contents:
  - type: row
    contents:
      - type: col
        contents:
          - type: component
            title: Scans
            component: scantable

  - type: row
    contents:
      - type: col
        contents:
          - type: component
            component: hardware
            title: Diffractometer2
            options:
              ids:
                - id: omega
                  step: 90
                  steps: [45,90,180]
```

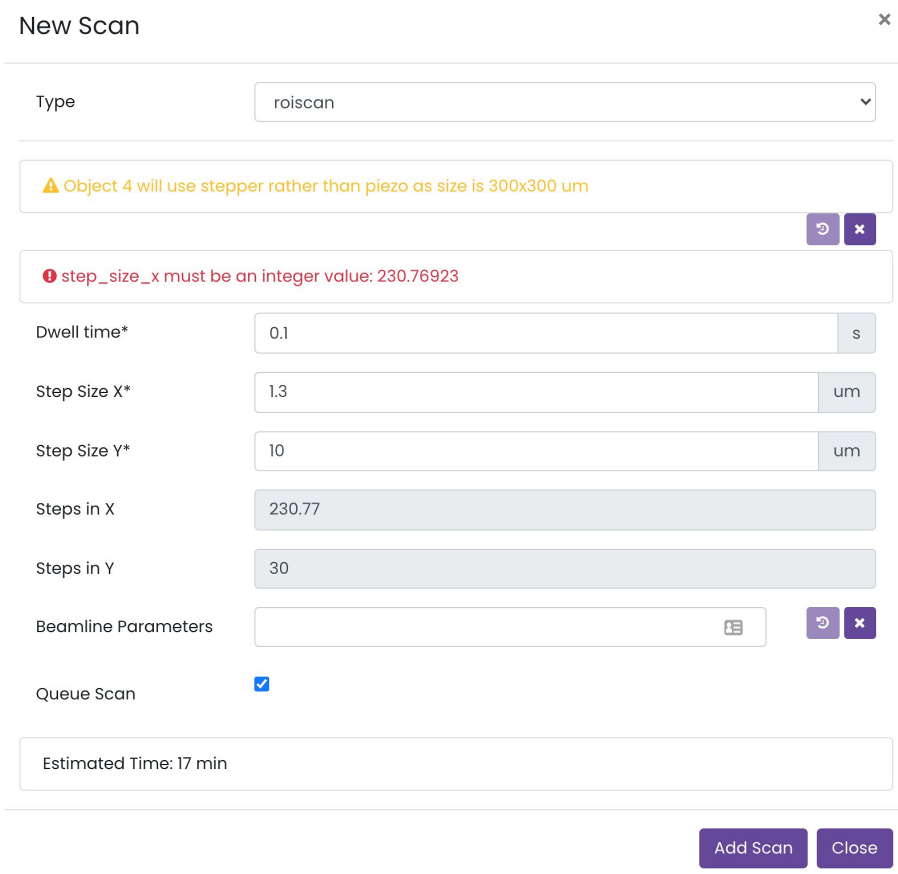
The European Synchrotron | ESRF

Schemas and Forms

Asynchronous validation, calculation, warning

Automatically reloaded

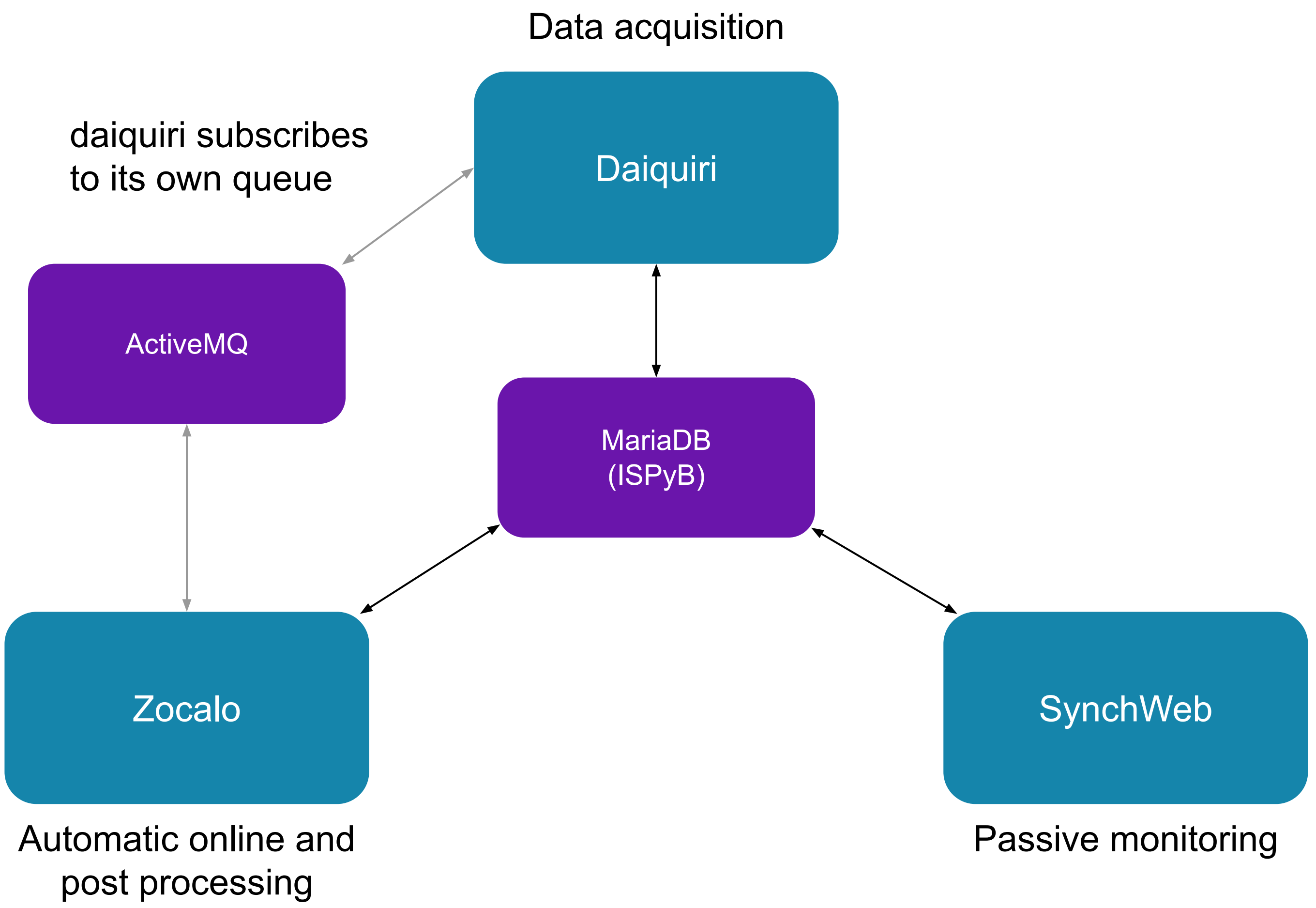
```
class ExampleSchema(ComponentActorSchema):
    motor = OneOf(["robz", "roby"], required=True, title="Motor")
    motor_start = fields.Float(required=True, title="Start Position")
    motor_end = fields.Float(required=True, title="End Position")

    @validates_schema
    def schema_validate(self, data, **kwargs):
        raise ValidationError("Invalid!")

    def warnings(self, data, **kwargs):
        return {"warning1": "Object will use stepper"}

class ExampleActor(ComponentActor):
    schema = ExampleSchema
    name = "example"

    def method(self, **kwargs):
        ...
```


New Scan

Type	roiscan	
Object 4 will use stepper rather than piezo as size is 300x300 um		
step_size_x must be an integer value: 230.76923		
Dwell time*	0.1	s
Step Size X*	1.3	um
Step Size Y*	10	um
Steps in X	230.77	
Steps in Y	30	
Beamline Parameters		
Queue Scan	✓	

Estimated Time: 17 min

Global Infrastructure



https://github.com/DiamondLightSource/python-zocalo
https://github.com/DiamondLightSource/SynchWeb

Docker

Containerised project for demo, local development, testing, etc

daiquiri-docker

- daiquiri/ui, bliss, nexus writer, lima simulator, tango dummy
- https://gitlab.esrf.fr/ui/daiquiri-docker

daiquiri-docker-testdb

- Pre-populated mariadb with a session, beamline, user, and test data
- https://gitlab.esrf.fr/ui/daiquiri-docker-testdb

https://hub.docker.com/u/esrfbcu

Status

Deployed on:

* id21 - xrf mapping + spectroscopy

* bm29 - biosaxs (custom frontend BSXCuBE3)

bm23 (commissioning) - spectroscopy

id26 (monitoring, commissioning) - spectroscopy

id13 (commissioning) - mapping

bm05 (monitoring, commissioning) - industry / tomography

Deploying to:

bm18 - tomography (very large to small scale)

id24 - spectroscopy

id27 - diffraction, extreme conditions

Links

General Information

- https://ui.gitlab-pages.esrf.fr/daiquiri-landing/about

Source

- https://gitlab.esrf.fr/ui/daiquiri
- https://gitlab.esrf.fr/ui/daiquiri-ui

Documentation

- https://ui.gitlab-pages.esrf.fr/daiquiri
- https://ui.gitlab-pages.esrf.fr/daiquiri-ui

Other Projects

- https://gitlab.esrf.fr/ui

Reference: https://doi.org/10.1107/S1600577521009851

Acknowledgements

UI Group

- Marcus Oscarsson
- Valentin Valls

Data Analysis (DAU):

- Wout de Nolf

Controls Unit (BCU):

- Jens Meyer, Andy Gotz, and many others

MXCuBE/3

- https://github.com/mxcube/mxcube3

GDA

- Jacob Filik (DLS)