

Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

VWright Group

Integrations

Queue Server

Conclusions

Acknowledgments

yaq - Yet Another Acquisition

A modular approach to spectroscopy software and instrumentation

Kyle F. Sunden

University of Wisconsin–Madison

2022-10-10



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

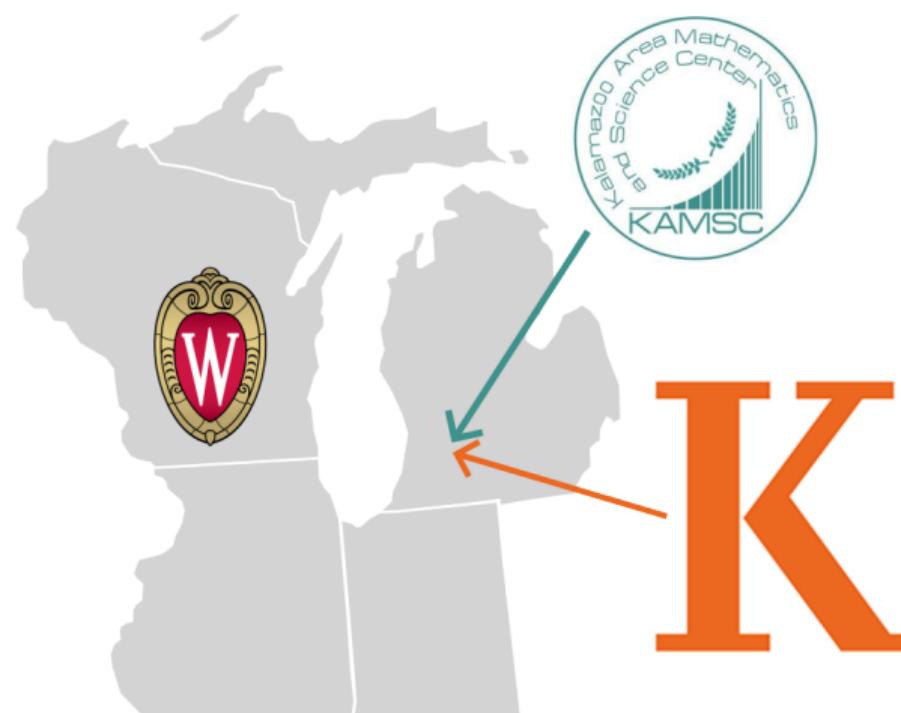
Vright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

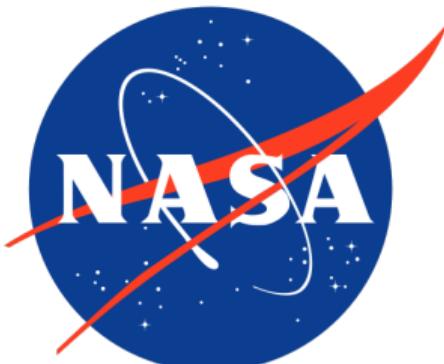
Queue Server

Conclusions

Acknowledgments



matplotlib



NUMFOCUS
OPEN CODE = BETTER SCIENCE

Coherent Multidimensional Spectroscopy (CMDS)

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

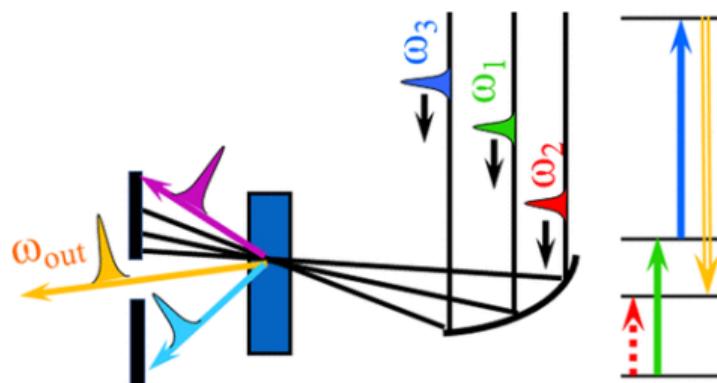
VVright Group

Integrations

Queue Server

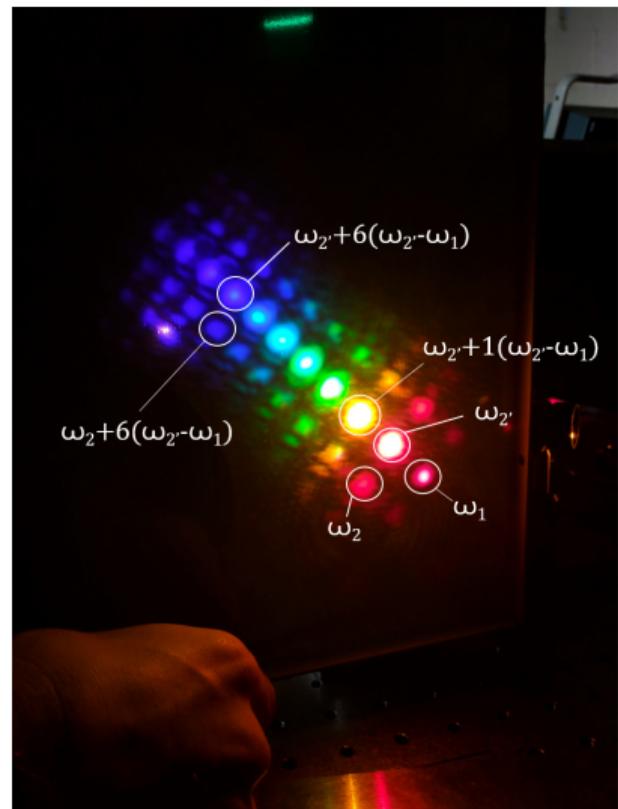
Conclusions

Acknowledgments



Source:

[10.1021/acs.analchem.0c01662](https://doi.org/10.1021/acs.analchem.0c01662)



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

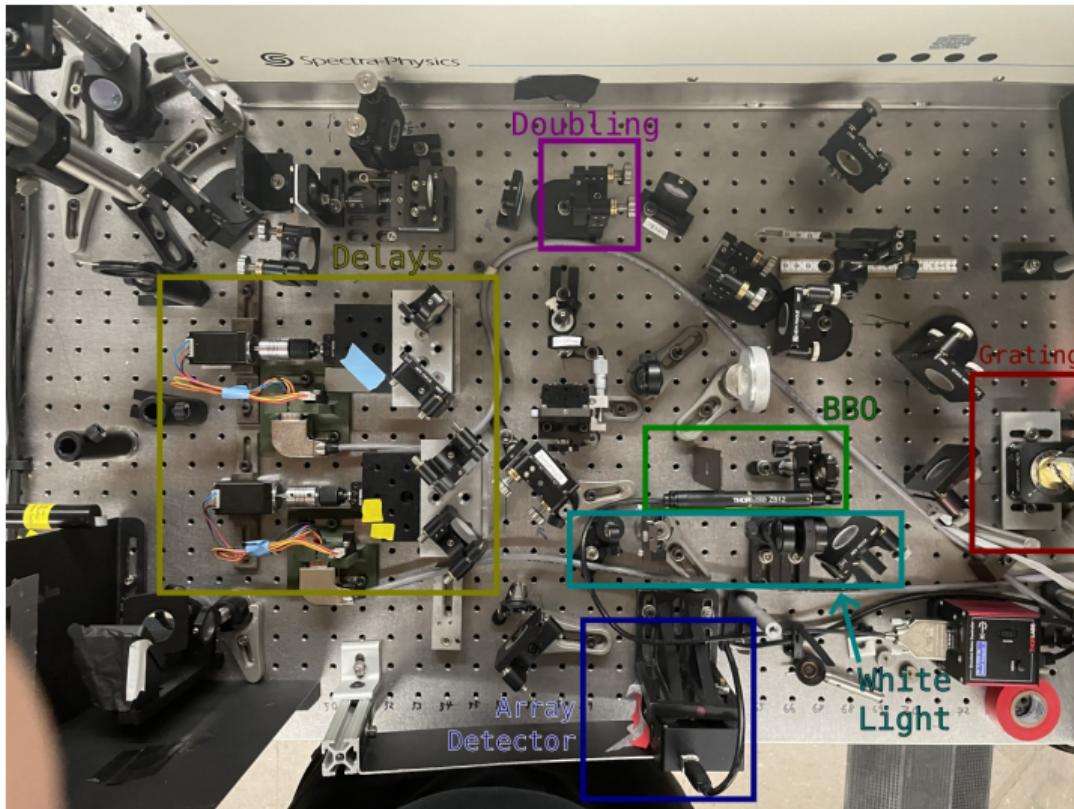
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

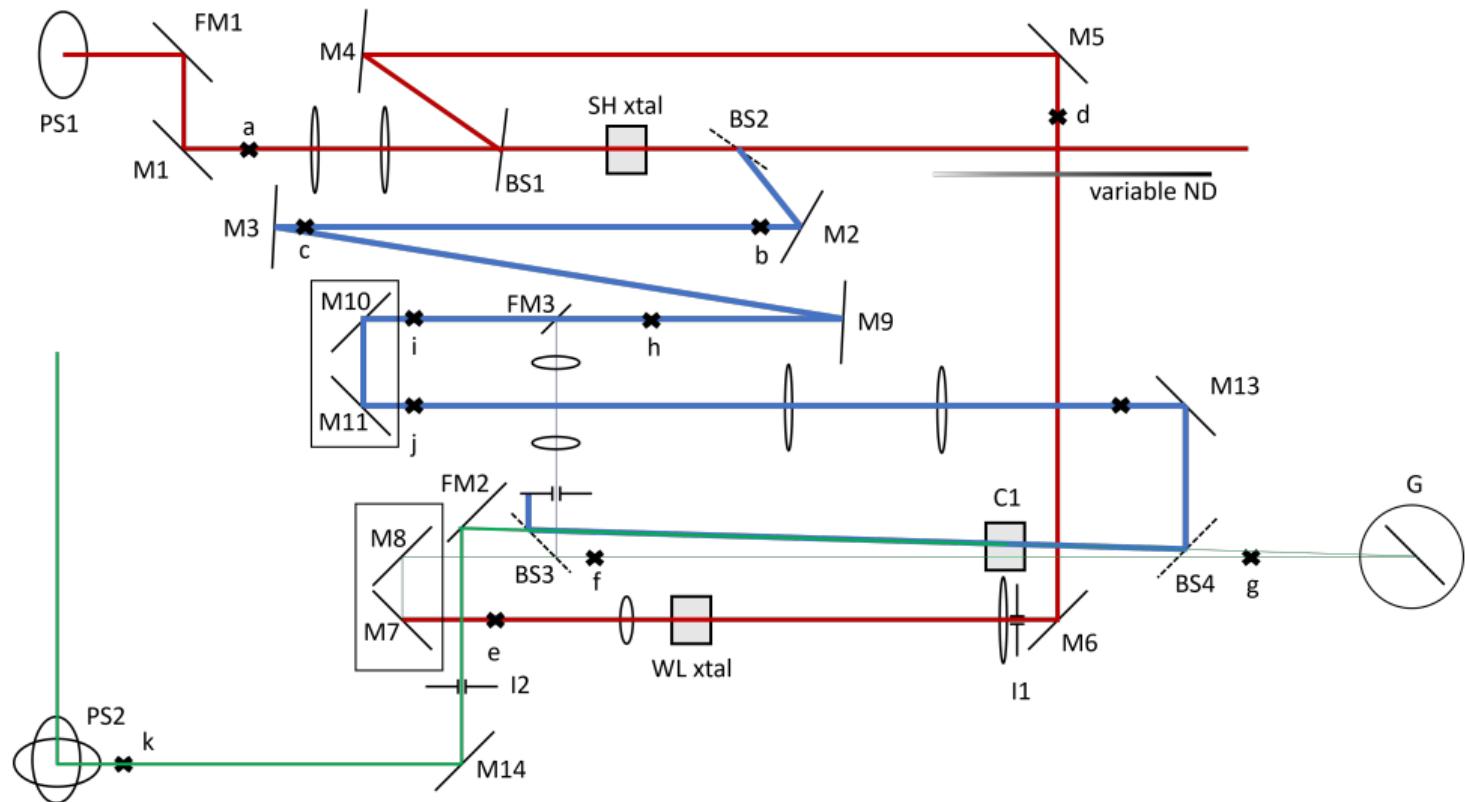
VWright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments

A single application which does:

- ▶ Direct hardware communication
- ▶ Engineering tasks to prepare for experiments
- ▶ Experimental orchestration
- ▶ Graphical User Interface
- ▶ Data recording
- ▶ Data work-up



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments

Problems:

- ▶ Systems are tightly coupled
- ▶ All hardware “speak” different “languages”
- ▶ Adding or changing hardware is arduous
- ▶ Often limited to one instrument/experiment
- ▶ Bugs unrelated to measurement cause measurement to stop



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

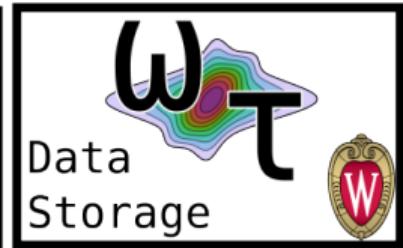
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

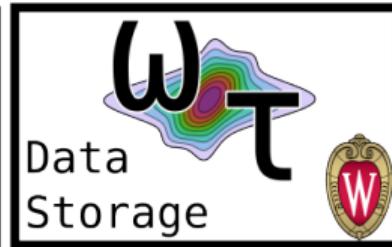
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

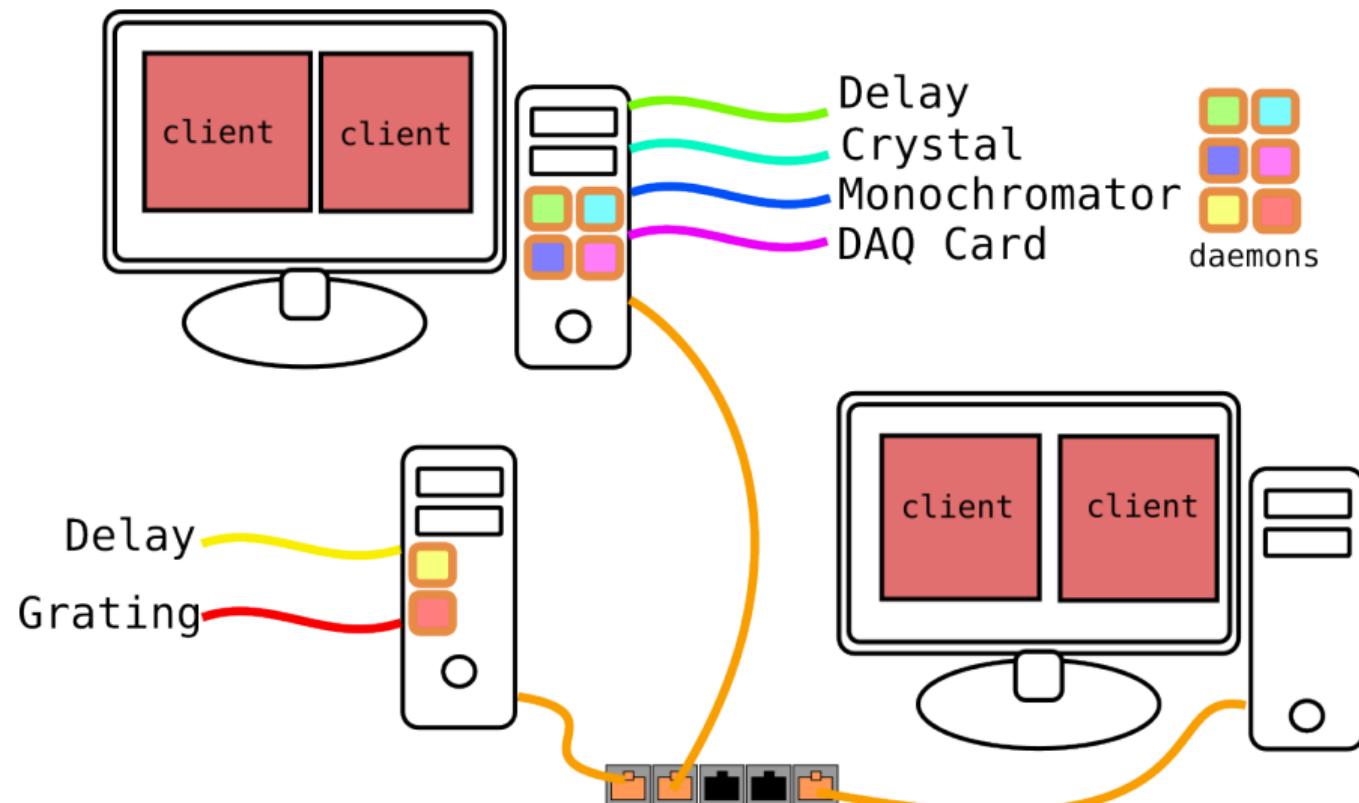
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Daemon: small background application which performs a single task

Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Vright Group

Integrations

Queue Server

Conclusions

Acknowledgments

Each yaq daemon provides a protocol:

- ▶ Self Describing
- ▶ Built on Apache Avro Remote Procedure Call
- ▶ Shared Traits for similar functionality



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments

We provide a generic client written for Python: yaqc

yaqc dynamically generates python methods from the protocol

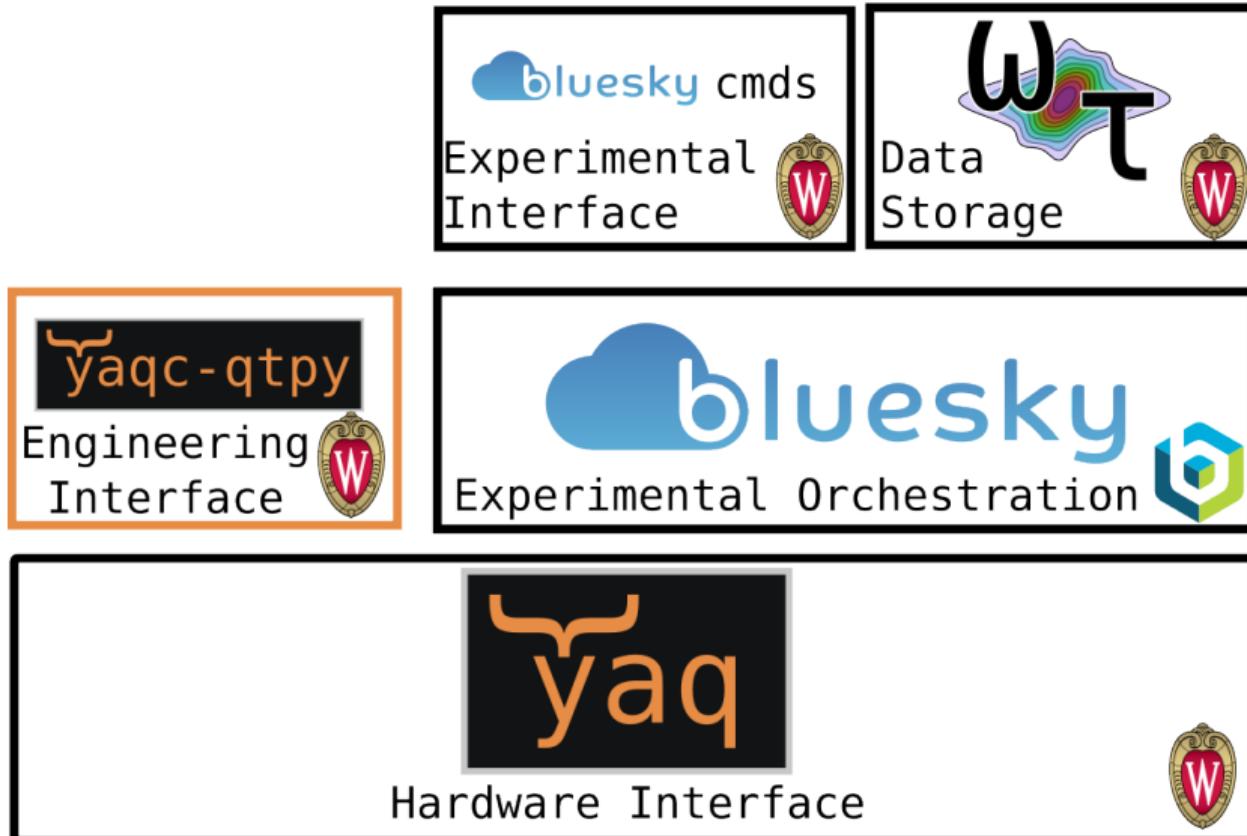
Simple Python scripts are an easy entry to using yaq



Intro

Clients

bluesky



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

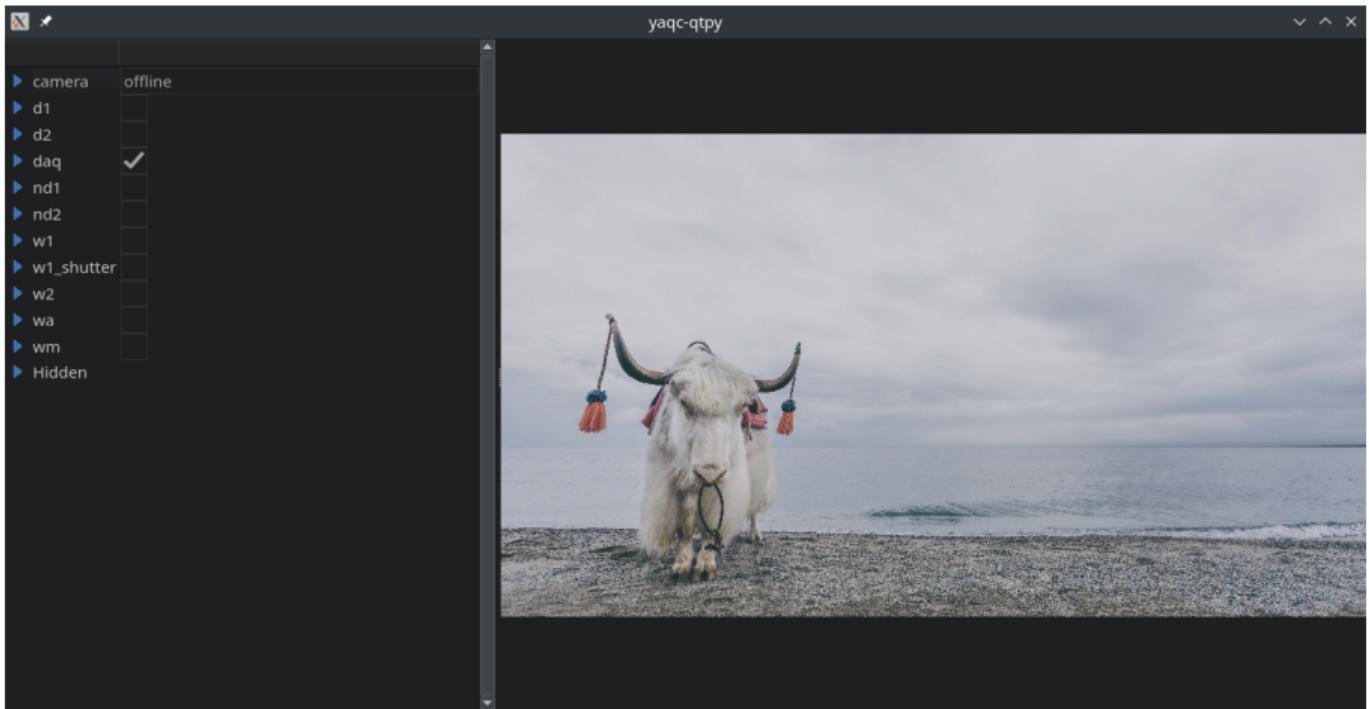
Vright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



The screenshot shows a sidebar interface for managing acquisition parameters. On the left, a tree view lists components: camera, d1, d2, and daq. Under d1, there are three sub-components: host:port, destination, and position. Under daq, there is one sub-component: host:port. To the right of the tree view, there are two sections for d1 and daq, each containing input fields and dropdown menus. The d1 section has fields for host:port (127.0.0.1:38401), destination (10.000000), and position (10.000000). The daq section has a host:port field (127.0.0.1:38999). Each section includes a "view advanced menu" button at the bottom.

Component	Sub-Component	Value	Unit
d1	host:port	127.0.0.1:38401	
	destination	10.000000	ps
	position	10.000000	ps
daq	host:port	127.0.0.1:38999	

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

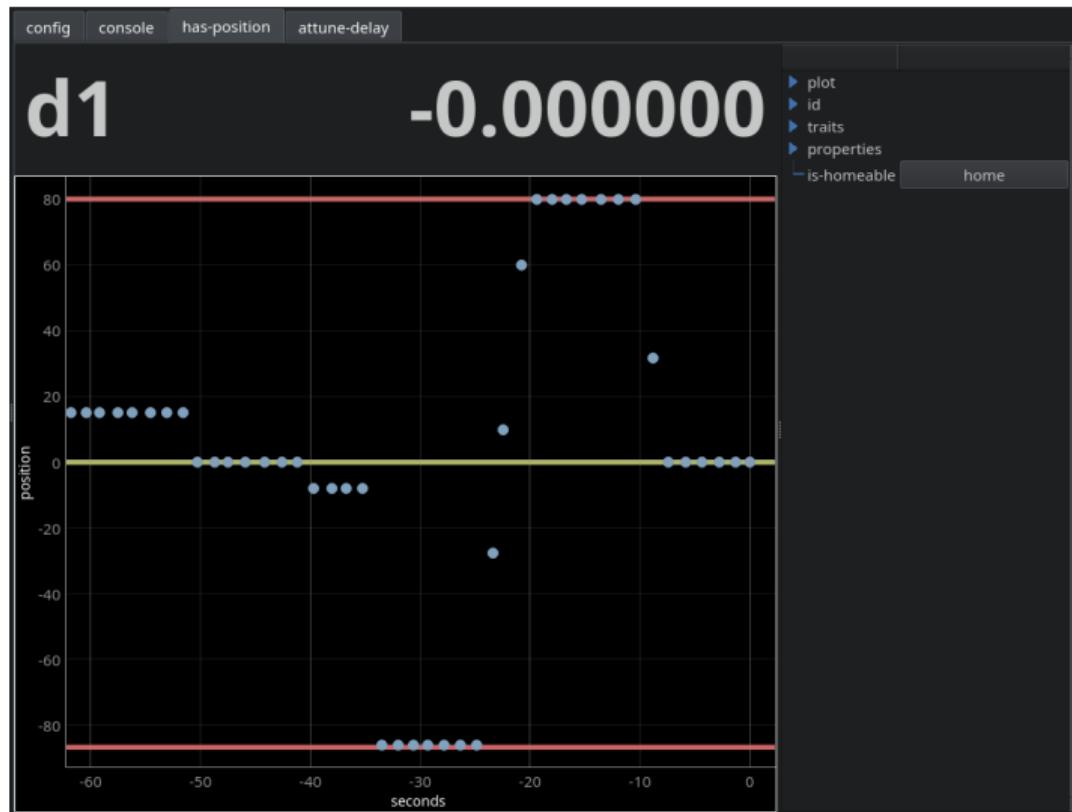
Vright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

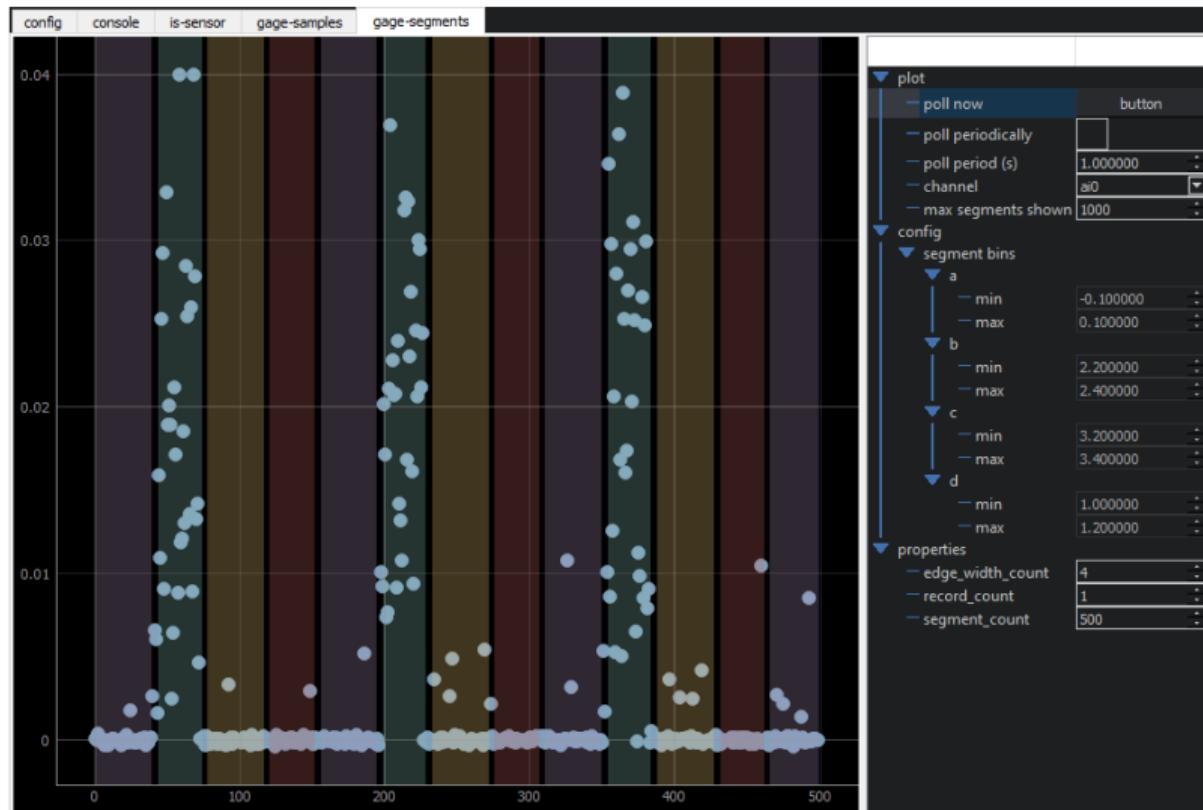
V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

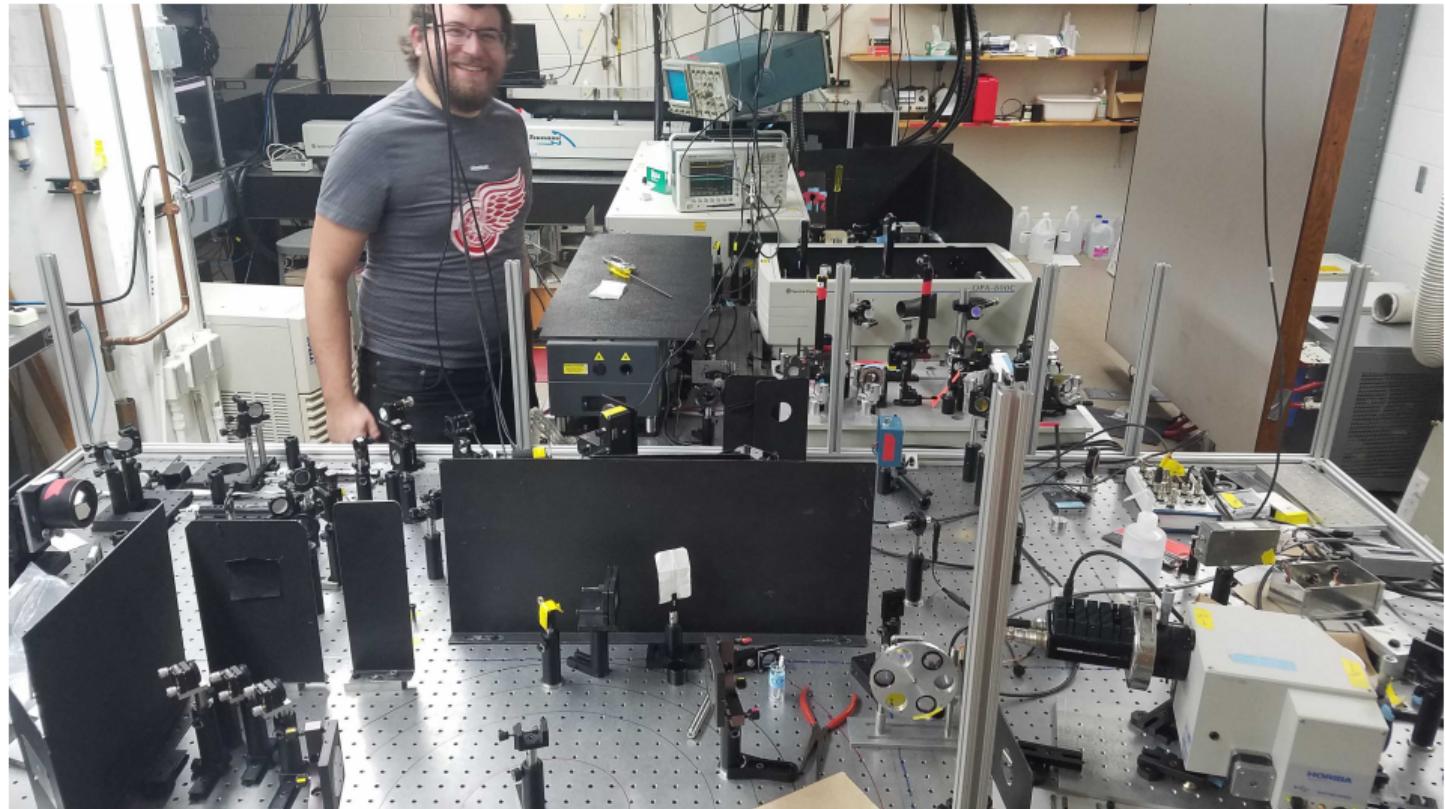
Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

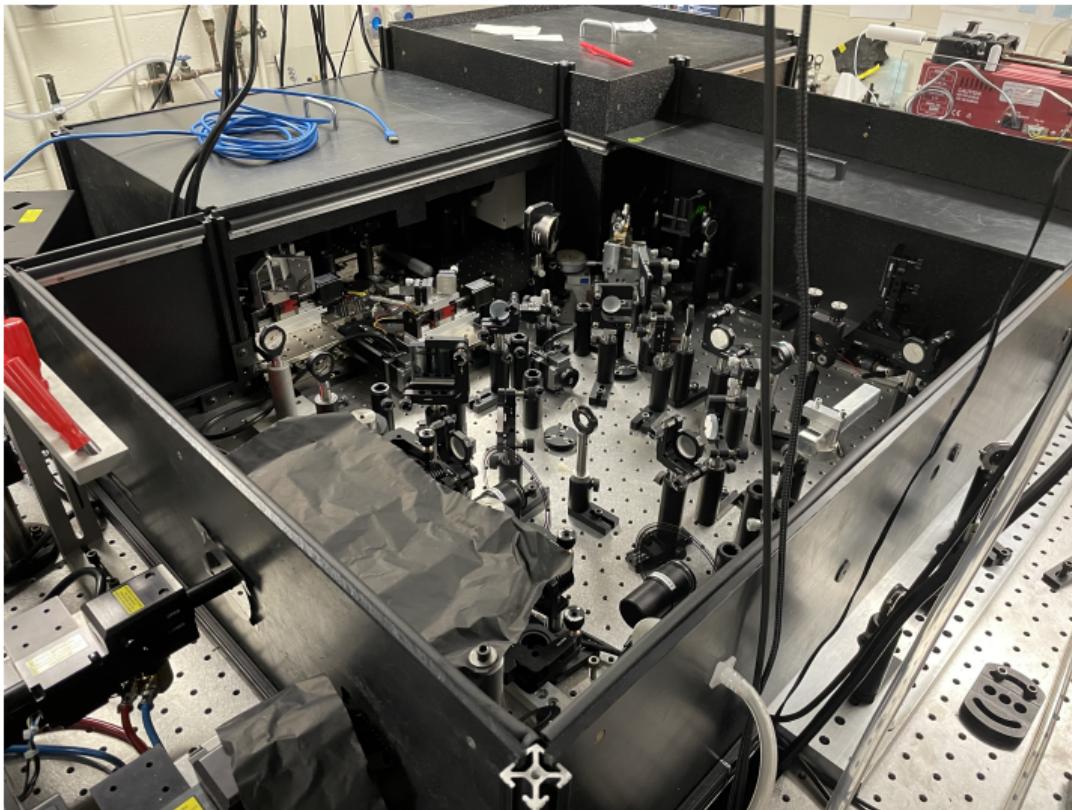
V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Outside of the Wright Group

Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

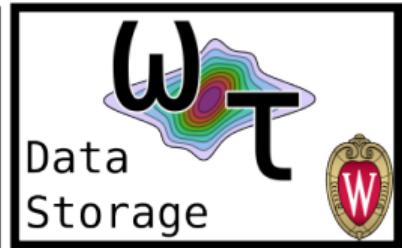
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

VWright Group

Integrations

Queue Server

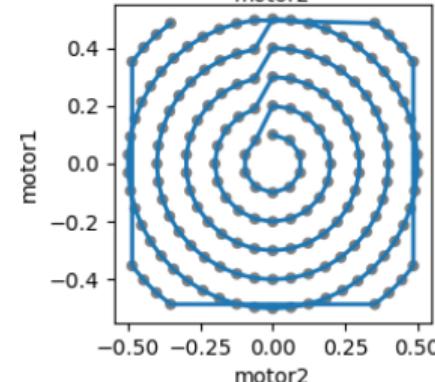
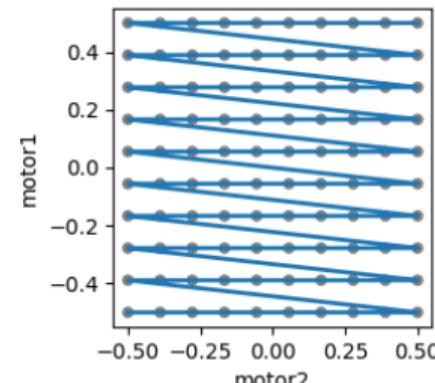
Conclusions

Acknowledgments



Steps for collecting a single point:

- ▶ set motors to correct positions
- ▶ wait for motion to complete
- ▶ trigger reading of sensors
- ▶ wait for sensors complete measurement
- ▶ read sensors and motors
- ▶ save data



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Wright Group
Integrations

Queue Server

Conclusions

Acknowledgments



Translate yaq into something that Bluesky can use.

bluesky / yaqc-bluesky (Public)

Code Issues 9 Pull requests 1 Discussions Actions Wiki Security Insights

master 1 branch 10 tags Go to file Add file Code

untag update which versions of python blu... 9a25ea6 9 days ago 145 commits

.github/workflows update which versions of python bluesky-git i... 9 days ago

binder refactor with one class per trait (#12) 2 years ago

tests Fixes for cameras and mappings (#77) 6 months ago

yaqc_bluesky stop forcing measure at sensor connect time ... 3 months ago

.gitignore Commit from cookiecutter 3 years ago

.pre-commit-config.y... no busy in read (#52) 2 years ago

CHANGELOG.md stop forcing measure at sensor connect time ... 3 months ago

LICENSE flit (#16) 2 years ago

README.md v2021.1.0 2 years ago

bluesky_protocols.nv test against bluesky_protocols 14 months ago

About

A bluesky interface to the yaq instrument control framework.

yaq.fyi/

bluesky yaq

Readme

BSD-3-Clause license

6 stars

3 watching

3 forks

Releases 9

2022.4.1 Latest on Apr 26

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

VWright Group
Integrations

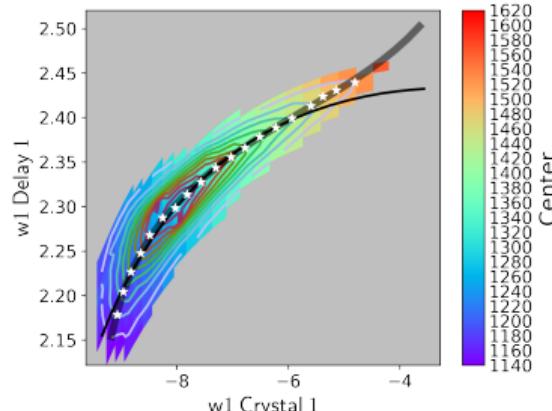
Queue Server

Conclusions

Acknowledgments



- ▶ Versions of built-in plans
 - ▶ Units support
 - ▶ “Constants”
- ▶ Plans for performing tuning operations



Modular Approach: WrightTools

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Wright Group
Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

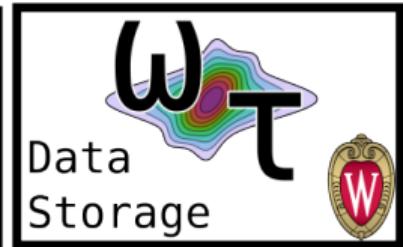
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

bluesky



Search or jump to... / Pull requests Issues Marketplace Explore

bluesky / bluesky-queueserver Public Edit Pins Unwatch 6 Fork 16 Starred 7

Code Issues 16 Pull requests Actions Projects 1 Wiki Security Insights

main 1 branch 17 tags Go to file Add file Code

 dmjav REL: v0.0.17 ✓ b074de3 7 days ago 1,489 commits

.github yml formating 17 days ago

bluesky_queueserver TST: generate new list of plans and devices 13 days ago

docs DOC: add missing 'executing_task' manager ... 7 days ago

.codecov.yml INT: run cookiecutter 3 years ago

.coveragerc CI: don't include tests and docs in coverage 2 years ago

.flake8 ENH: removed code for HTTP server 14 months ago

.gitattributes INT: run cookiecutter 3 years ago

.gitignore Add http server tests (4 for now) 2 years ago

AUTHORS.rst Fix Labraroty; sync LICENSE with bluesky/bl... 2 years ago

CONTRIBUTING.rst Update linting command in CONTRIBUTING.rst 2 years ago

LICENSE DOC: installation instructions 16 months ago

About

Server for queueing plans

[blueskyproject.io/bluesky-queueserver/](#)

hacktoberfest bluesky bluesky-queueserver bluesky-webclient hacktoberfest2021

Readme BSD-3-Clause license

7 stars 6 watching 16 forks

Releases

 Release v0.0.17 Latest 7 days ago

Modular Approach: Experimental GUI

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

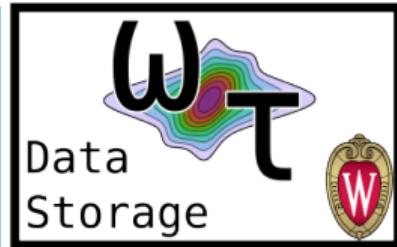
VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sunden

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Vvright Group

Integrations

Queue Server

Conclusions

Acknowledgments



bluesky-cmds 2022.7.0+fix_issues_39_40 00:05:55

Index	Type	Status	Description	REMOVE	LOAD
2	run_intensity	queued	[[{"daq": "w1", "delay_2": 3.0, "npts": 31, "spectrometer": {"device": "wm", "method": "static", "center": 200.0}}]		
1	run_intensity	queued	[[{"daq": "w1", "delay_2": 3.0, "npts": 31, "spectrometer": {"device": "wm", "method": "track"}}]		
0	run_intensity	queued	[[{"daq": "w1", "delay_2": 3.0, "npts": 31, "spectrometer": {"device": "wm", "method": "scan", "width": -350.0, "npts": 33}}]		
	run_intensity	RUNNING	[[{"daq": "w1", "delay_2": 4.0, "npts": 41, "spectrometer": {"device": "wm", "method": "track"}}]		
	run_intensity	completed	[[{"daq": "w1", "crystal_1": 3.0, "npts": 31, "spectrometer": {"device": "wm", "method": "zero"}}]		
	run_intensity	completed	[[{"wa": "w1", "crystal_1": 1, "npts": 11, "spectrometer": None}]]		
	run_intensity	completed	[[{"wa": "w1", "krystal_1": 1, "npts": 11, "spectrometer": None}]]		
	grid_scan_wp	completed	[[{"daq": "d0", "d0": -1.0, "d1": 0.61, "ps": "d2", "d2": -1.0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 25}]]		
	grid_scan_wp	completed	[[{"daq": "d0", "d0": -1.0, "d1": 0.61, "ps": "d2", "d2": -1.0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 25}]]		
	grid_scan_wp	completed	[[{"daq": "w1", "d0": -1.0, "d1": 0.11, "ps": "d2", "d2": 0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 11}]]		
	grid_scan_wp	failed	[[{"daq": "d0", "d0": -1.0, "d1": 0.61, "ps": "d2", "d2": -1.0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 25}]]		
	grid_scan_wp	completed	[[{"daq": "w1", "d0": -1.0, "d1": 0.11, "ps": "d2", "d2": 0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 11}]]		
	grid_scan_wp	unknown	[[{"daq": "d0", "d0": -1.0, "d1": 0.61, "ps": "d2", "d2": -1.0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 25}]]		
	grid_scan_wp	stopped	[[{"daq": "w1", "d0": -1.0, "d1": 0.61, "ps": "d2", "d2": -1.0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 25}]]		
	grid_scan_wp	completed	[[{"daq": "w1", "d0": -1.0, "d1": 0.11, "ps": "d2", "d2": 0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 11}]]		
	grid_scan_wp	failed	[[{"daq": "w1", "d0": -1.0, "d1": 0.11, "ps": "d2", "d2": 0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 11}]]		
	grid_scan_wp	failed	[[{"daq": "w1", "d0": -1.0, "d1": 0.11, "ps": "d2", "d2": 0, "ps": "d1", "d1": 0, "ps": "d0", "ps": "d1", "npts": 11}]]		
	count	completed	[[{"daq": "w1"}]]		
	count	completed	[[{"daq": "w1"}]]		

Kyle F. Sundén

Intro



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

VVright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Wright Group

- ▶ Dan Kohler
- ▶ Skye Kain
- ▶ Jonathan Handali
- ▶ Darien Morrow
- ▶ Nathan Neff-Mallon
- ▶ Kyle Czech
- ▶ Natalia Spitha
- ▶ Emily Kaufman
- ▶ David Lafayette
- ▶ Chris Roy
- ▶ Jason Scheeler
- ▶ Kelson Oram
- ▶ Kent Meyer
- ▶ Peter Cruz Parrilla
- ▶ Ryan McDonnell
- ▶ James Harris
- ▶ (and more...)

Bluesky Community

- ▶ Tom Caswell
- ▶ Dan Allen
- ▶ Dmitri Gavrilov
- ▶ Pete Jemian
- ▶ Zach Lenz
- ▶ Maksim Rakitin
- ▶ Dylan McReynolds
- ▶ Tom Cobb
- ▶ Callum Forrester
- ▶ Clinton Roy
- ▶ Abby Giles
- ▶ Juan Marulanda
- ▶ Juliane Reinhardt
- ▶ Ken Lauer
- ▶ Rober Tang-Kong
- ▶ Antoine Wojdyla
- ▶ (and more...)

Committee

- ▶ John Wright
- ▶ Bob Hamers
- ▶ Blaise Thompson
- ▶ Clark Landis
- ▶ Etienne Garand

UW-Madison Chemistry
Department

- ▶ Rob McClain
- ▶ Steve Myers

Open Source Maintainers

Friends and family

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

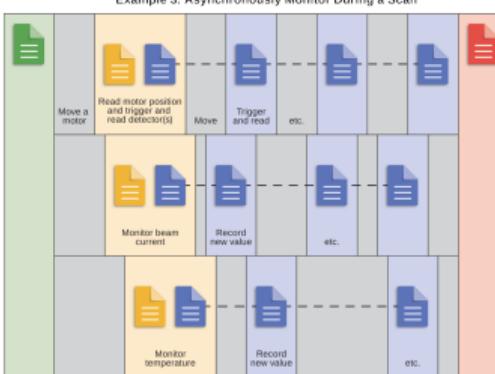
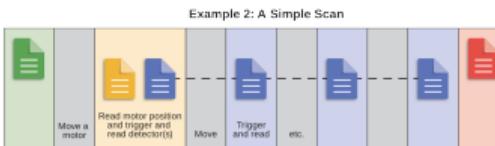
Community

V Wright Group
Integrations

Queue Server

Conclusions

Acknowledgments



Run Start: Metadata about this run, including everything we know in advance: time, type of experiment, sample info., etc.

Event Descriptor: Metadata about the readings in the event (units, precision, etc.) and the relevant hardware

Event: Readings and timestamps

Run Stop: Additional metadata known at the end: what time it completed and its exit status (success, aborted, failed)

Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

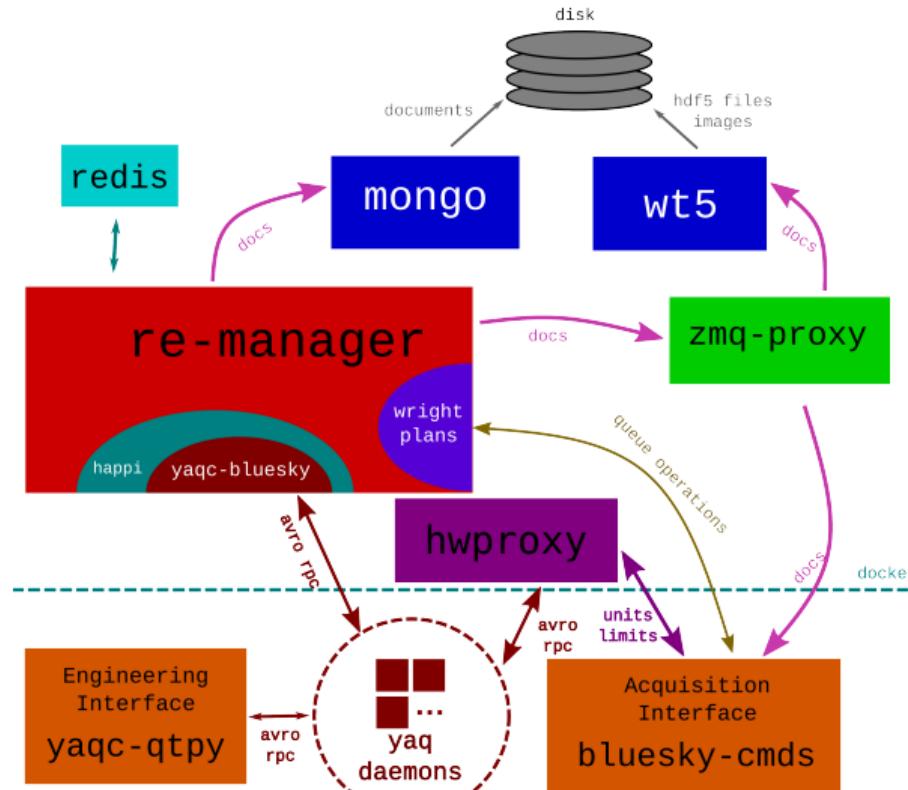
V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments



Kyle F. Sundén

Intro

Motivation

yaq

Implementation

Clients

Users of yaq

bluesky

Community

V Wright Group

Integrations

Queue Server

Conclusions

Acknowledgments

