SOS 26

March 12th, 2024

Workflow Execution Interface (WEI): A Practical Framework for Integration of Diverse Scientific Instruments at Scale for Automated Scientific Experimentation

CASEY STONE
Computational Scientist
Data Science and Learning Division
Argonne National Laboratory

Rafael Vescovi, Doga Ozgulbas, Ryan Lewis, Tobias Ginsburg, Abe Stroka, Kyle Hippe, Rory Butler, Mark Hereld, Ben Blaiszik, Tom Brettin, Priyanka Setty, Arvind Ramanathan, Ian Foster

Published Work

RESEARCH-ARTICLE

Exploring Benchmarks for Self-Driving Labs using Color Matching

Authors: Tobias Ginsburg, Kyle Hippe, Ryan Lewis, Aileen Cleary, Doga Ozgulbas, Rory Butler, Casey Stone, Abraham Stroka, Rafael Vescovi, Ian Foster Authors Info & Claims

SC-W '23: Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis • November 2023 • Pages 2147–2152 • https://doi.org/10.1145/3624062.3624615

Published: 12 November 2023 Publication History

ABSTRACT

Self Driving Labs (SDLs) that combine automation of experimental procedures with autonomous decision making are gaining popularity as a means of increasing the throughput of scientific workflows. The task of identifying quantities of supplied colored pigments that match a target color, the color matching problem, provides a simple and flexible SDL test case, as it requires experiment proposal, sample creation, and sample analysis, three common components in autonomous discovery applications. We present a robotic solution to the color matching problem that allows for fully autonomous execution of a color matching protocol. Our solution leverages the WEI science factory platform to enable portability across different robotic hardware, the use of alternative optimization methods for continuous refinement, and automated publication of results for experiment tracking and post-hoc analysis.

DOI: 10.1145/3624062.3624615

ROYAL SOCIETY OF CHEMISTRY

DOI: 10.1039/D3DD00142C (Paper) *Digital Discovery*, 2023, **2**, 1980-1998

Towards a modular architecture for science factories†

Rafael Vescovi [a], Tobias Ginsburg [a], Kyle Hippe [a], Doga Ozgulbas [a], Casey Stone [a], Abraham Stroka [a], Rory Butler [a], Ben Blaiszik [ab], Tom Brettin [a], Kyle Chard [ab], Mark Hereld [ab], Arvind Ramanathan [a], Rick Stevens [ab], Aikaterini Vriza [a], Jie Xu [ab], Qingteng Zhang [a] and Ian Foster [*ab]

[a] *Argonne National Laboratory, Lemont, IL 60439, USA. E-mail: foster@anl.gov*

[b] *University of Chicago, Chicago, IL 60637, USA*

Received 31st July 2023 , Accepted 26th October 2023

First published on 7th November 2023

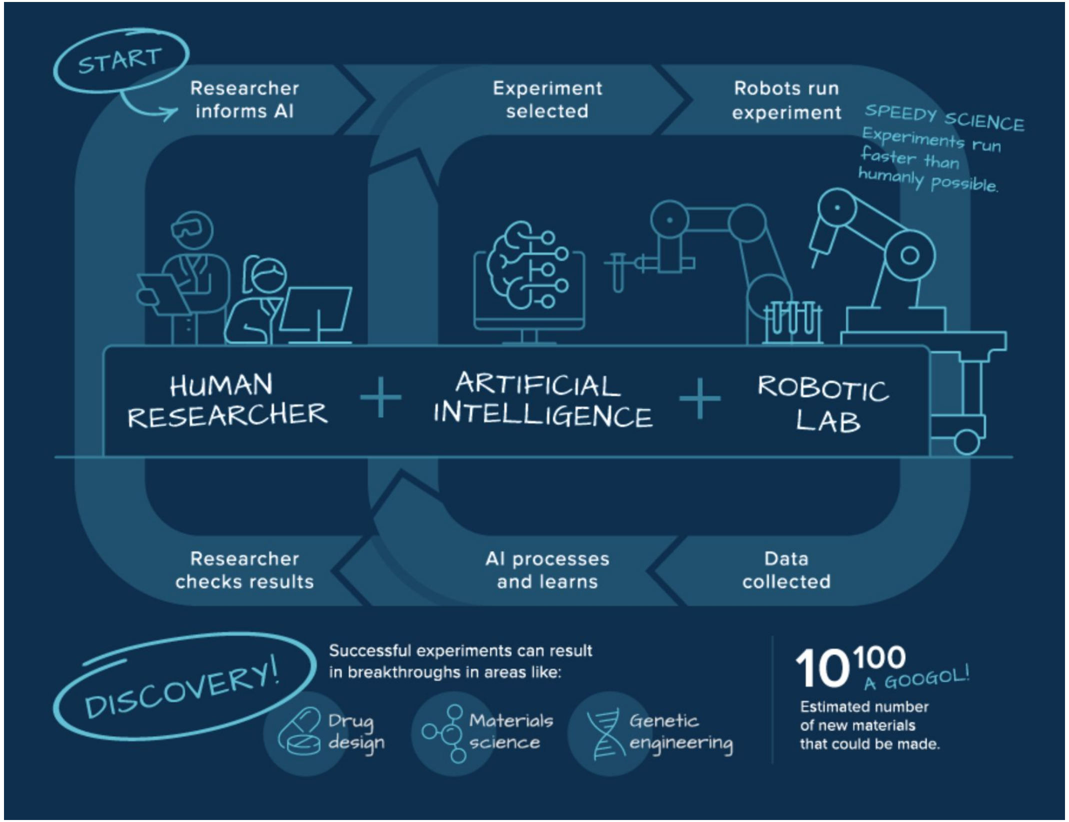
Abstract

Advances in robotic automation, high-performance computing (HPC), and artificial intelligence (AI) encourage us to conceive of *science factories*: large, general-purpose computation- and AI-enabled self-driving laboratories (SDLs) with the generality and scale needed both to tackle large discovery problems and to support thousands of scientists. Science factories require modular hardware and software that can be replicated for scale and (re)configured to support many applications. To this end, we propose a prototype modular science factory architecture in which reconfigurable *modules* encapsulating scientific instruments are linked with manipulators to form *workcells*, that can themselves be combined to form larger assemblages, and linked with distributed computing for simulation, AI model training and inference, and related tasks. *Workflows* that perform sets of actions on modules can be specified, and various *applications*, comprising workflows plus associated computational and data manipulation steps, can be run concurrently. We report on our experiences prototyping this architecture and applying it in experiments involving 15 different robotic apparatus, five applications (one in education, two in biology, two in materials), and a variety of workflows, across four laboratories. We describe the reuse of modules, workcells, and workflows in different applications, the migration of applications between workcells, and the use of digital twins, and suggest directions for future work aimed at yet more generality and scalability. Code and data are available at https://ad-sdl.github.io/wei2023 and in the ESI.

DOI: 10.1039/D3DD00142C

What is Autonomous Discovery?

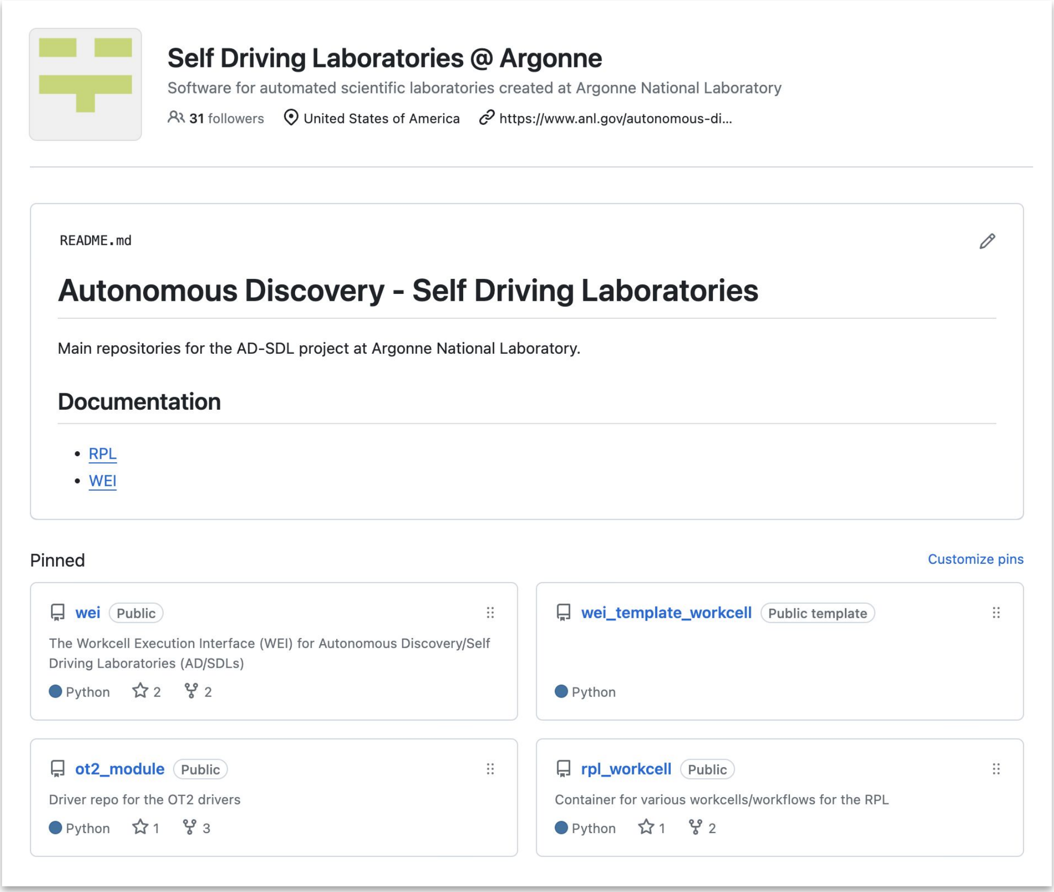
Harnessing the power of artificial intelligence — including robotics, machine learning, simulations and more — to aid in the planning, execution and analysis of scientific experiments



Workflow Execution Interface (WEI)

a robust, modular, composable, and extensible toolkit that coordinates the many instruments and services required for automated experiments and autonomous discovery in the physical sciences.

- Modular
- Open-source
- Experimental logic implemented in Python
- Integrated with Globus ecosystem


Self Driving Laboratories @ Argonne
Software for automated scientific laboratories created at Argonne National Laboratory
31 followers United States of America https://www.anl.gov/autonomous-di...

README.md

Autonomous Discovery - Self Driving Laboratories

Main repositories for the AD-SDL project at Argonne National Laboratory.

Documentation

- RPL
- WEI

Pinned | Customize pins

wei Public
The Workcell Execution Interface (WEI) for Autonomous Discovery/Self Driving Laboratories (AD/SDLs)
Python 2 2

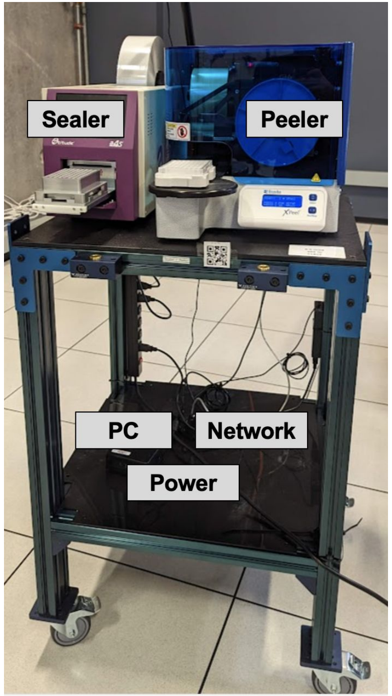
wei_template_workcell Public template
Python

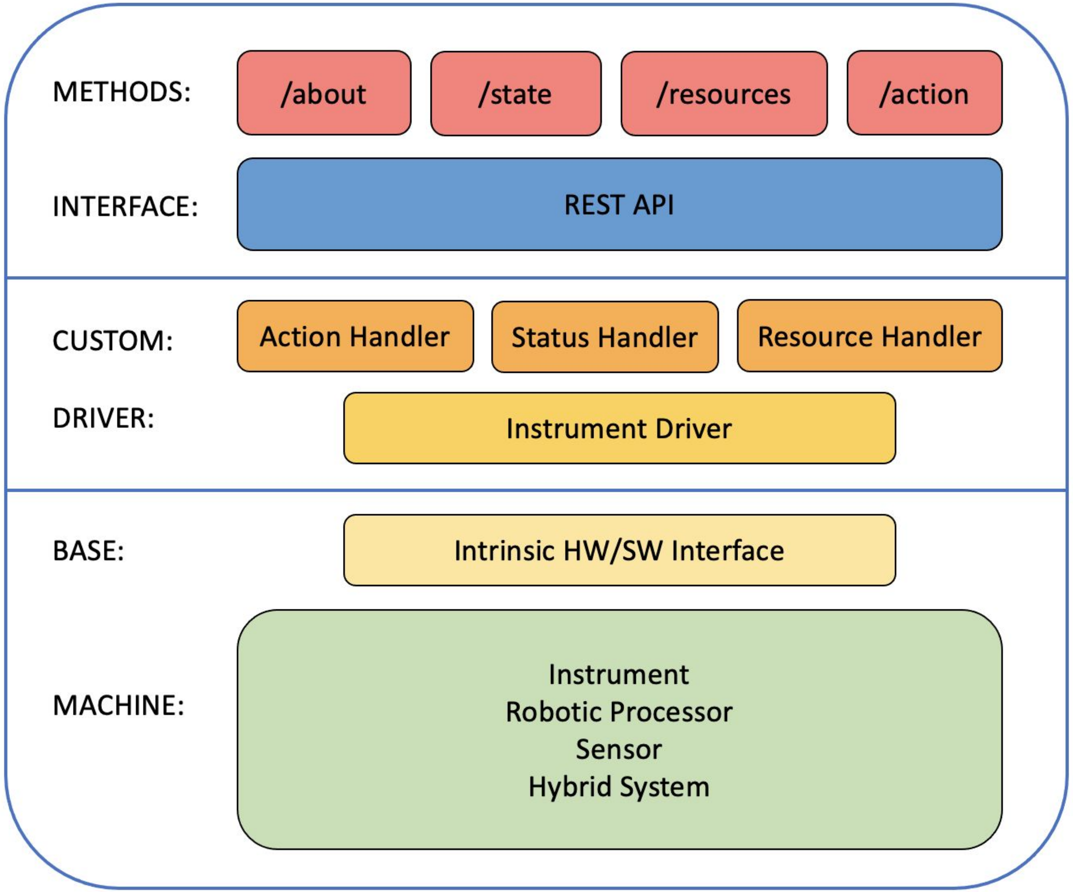
ot2_module Public
Driver repo for the OT2 drivers
Python 1 3

rpl_workcell Public
Container for various workcells/workflows for the RPL
Python 1 2

https://github.com/AD-SDL

Modules


Sealer
Peeler
PC
Network
Power


METHODS:
/about
/state
/resources
/action
INTERFACE:
REST API
CUSTOM:
Action Handler
Status Handler
Resource Handler
DRIVER:
Instrument Driver
BASE:
Intrinsic HW/SW Interface
MACHINE:
Instrument
Robotic Processor
Sensor
Hybrid System

Development of Modules

A **common API** abstracts details of the interfaces used by various devices

After experimenting with different interfaces, we converged on REST API

REST				
Python API	C# API	Python API	C# API	Python API
Ethernet	Serial, WinDLL	Ethernet	USB, WCF	Wireless

Opentrons OT-2

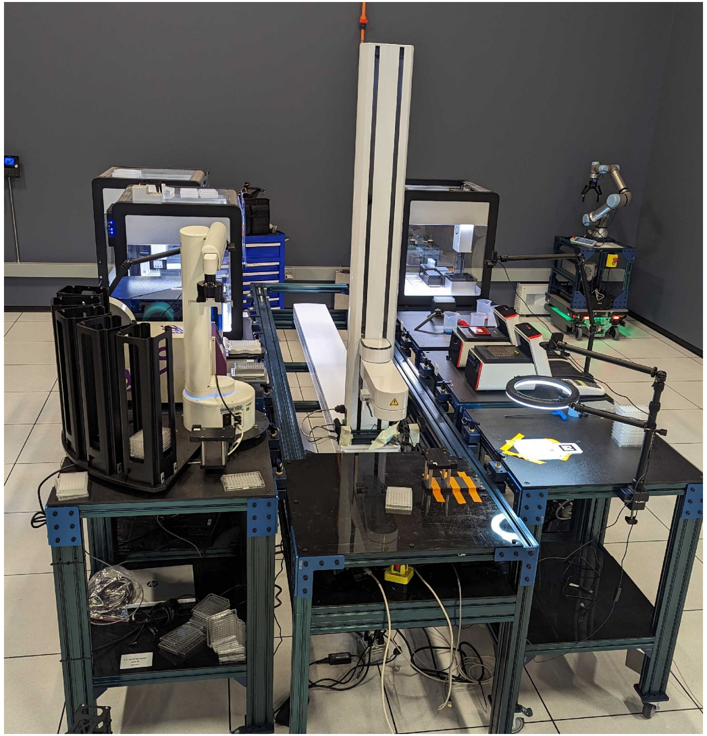
Thermocycler

ChemSpeed

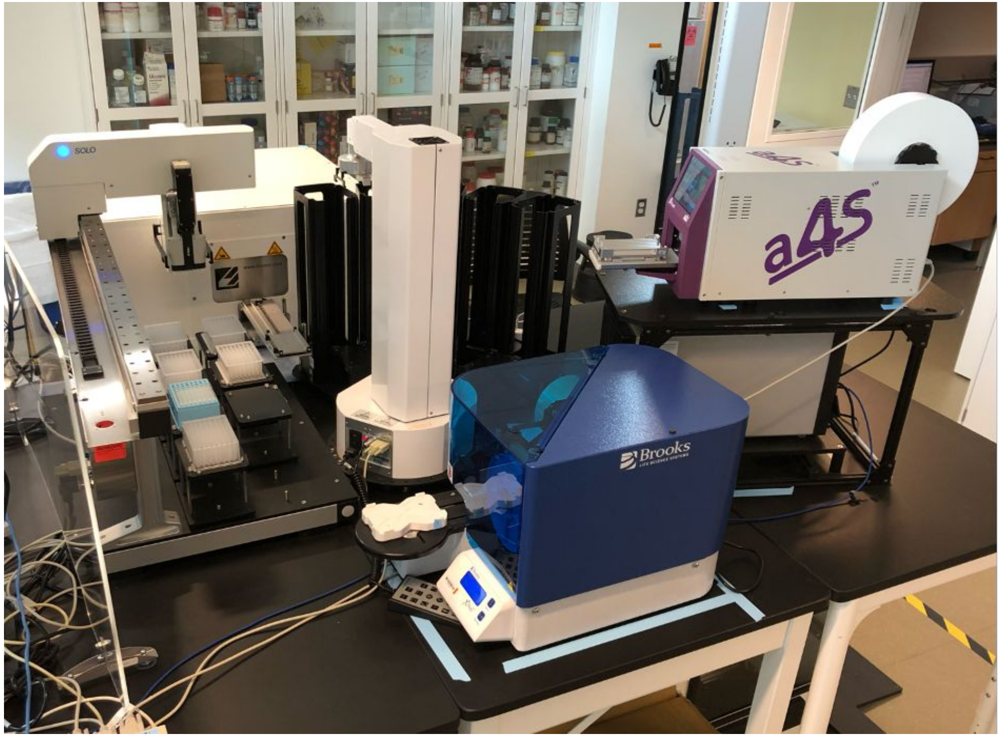
Hidex Sense

MiR 250 Mobile Base

Modules can be combined into Workcells



Cart-based Workcell in RPL



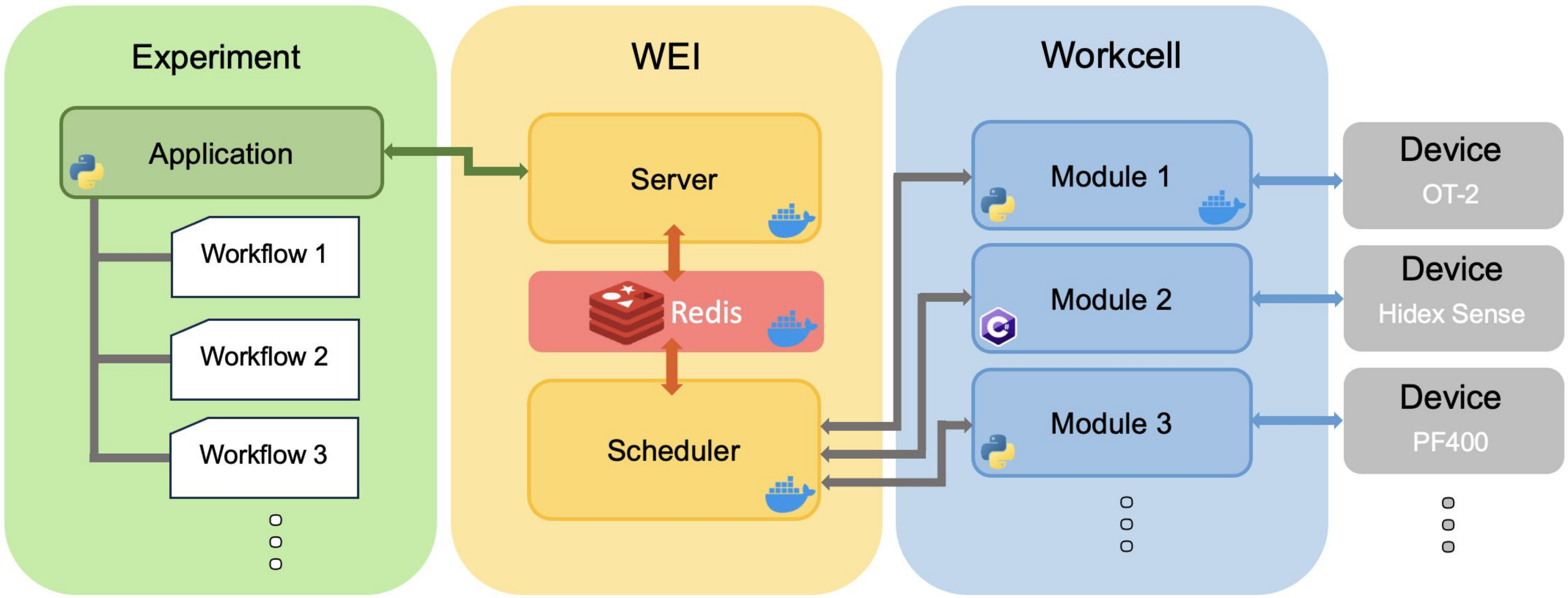
Compact Workcell in Biosciences Division

Supported Devices

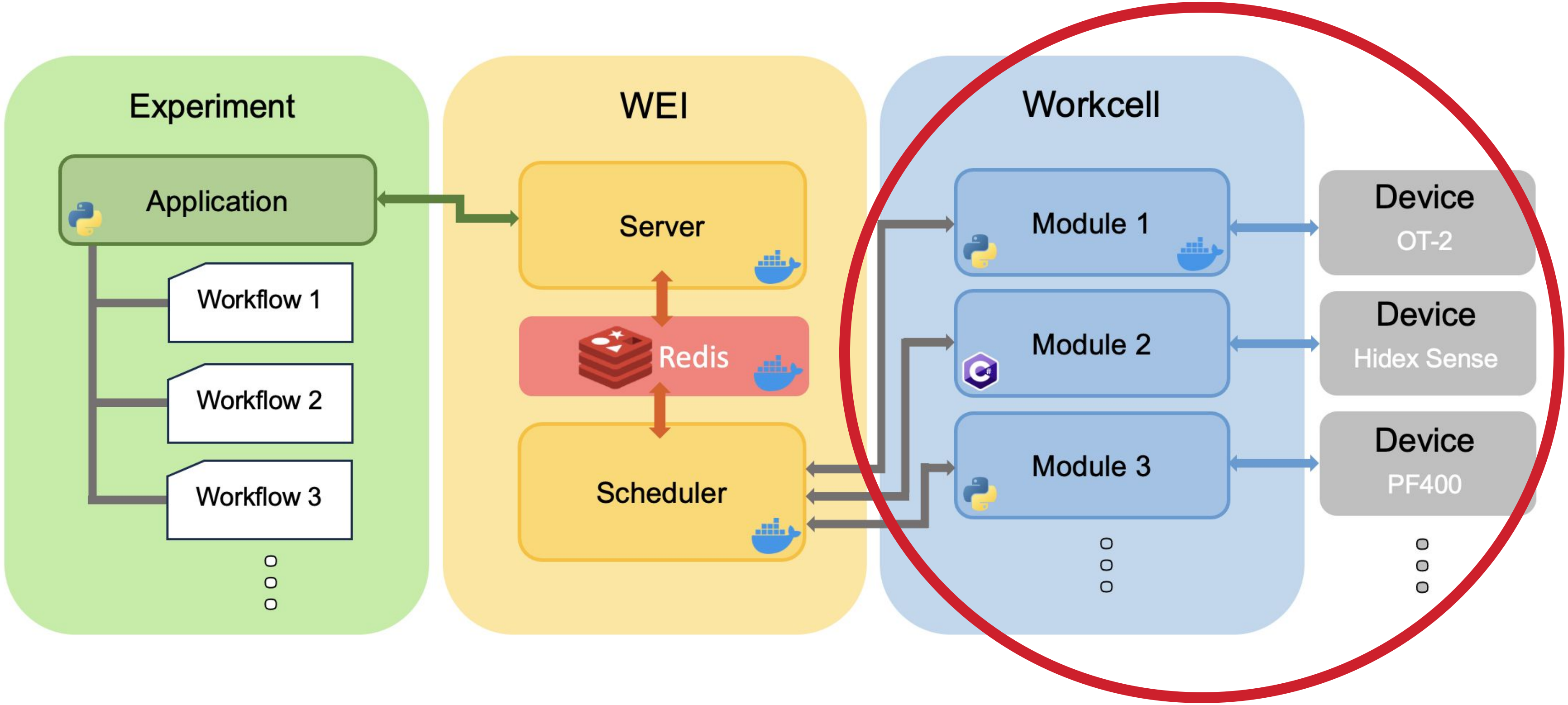
- PF400
- Hudson PlateCrane EX
- Hudson SciClops
- UR5e
- N9 Robotic Arm
- MiR250
- Hudson SOLO Liquid Handler
- Opentrons OT-2
- Chemspeed
- Hidex Sense Microplate Reader
- A4S Automated Plate Heat Sealer
- Brooks Automated Plate Seal Remover
- Biometra TRobot II Thermocycler
- HiG4 Automated Centrifuge

https://github.com/AD-SDL

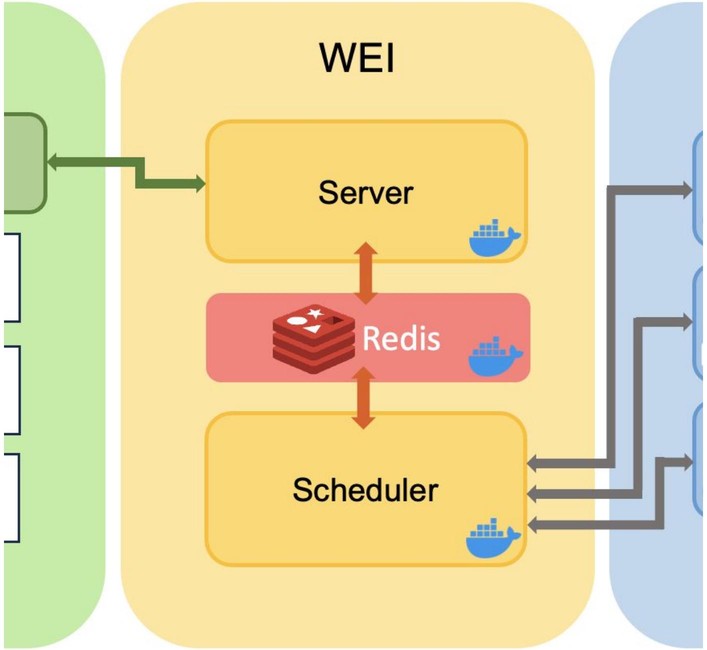
WEI Architecture



WEI Architecture

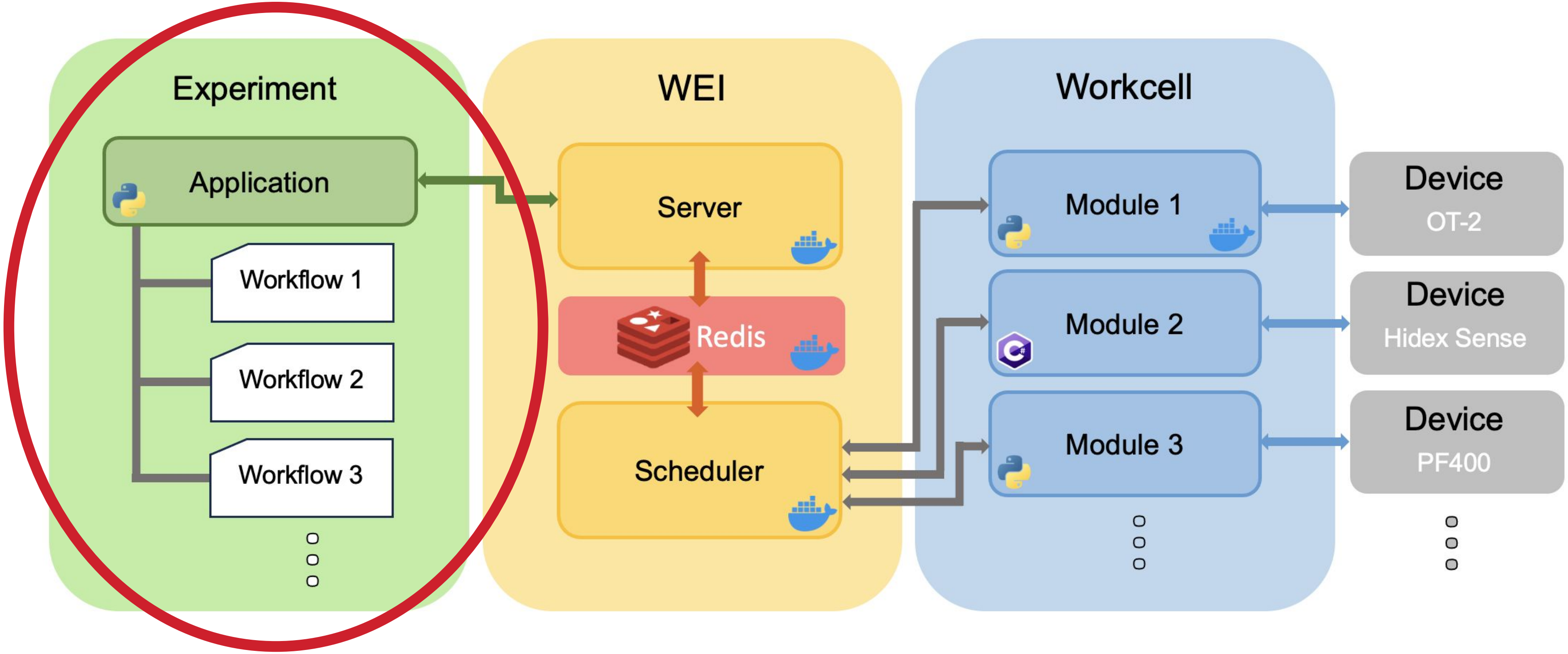


WEI Server



- REST API
- Independent of user-side experiment applications and device-side module functionality
- Robust to errors, issues, and client level problems
- Manages instruments and data
- Handles experimental logging

WEI Architecture



Workflows

A sequence of steps to be executed on a given Workcell to accomplish a scientific task

- YAML formatted
- Each step specifies an action to be performed on a specified module

Protocol Files

Defines an instrument specific set of actions


```yaml
name: Color Picker - Mix Colors - Workflow

metadata:
  author: Tobias Ginsburg, Rafael Vescovi
  info: Main workflow for the RPL Color Picker
  version: 0.1

modules:
  - name: ot2_cp_gamma
  - name: pf400
  - name: camera_module

flowdef:
  - name: Move from Camera Module to OT2
    module: pf400
    action: transfer
    args:
      source: camera_module.plate_station
      target: ot2_cp_gamma.deck2
      source_plate_rotation: narrow
      target_plate_rotation: wide
    comment: Place plate in ot2

  - name: Mix all colors
    module: ot2_cp_gamma
    action: run_protocol
    args:
      color_A_volumes: payload.color_A_volumes
      color_B_volumes: payload.color_B_volumes
      color_C_volumes: payload.color_C_volumes
      color_D_volumes: payload.color_D_volumes
      destination_wells: payload.destination_wells
      use_existing_resources: payload.use_existing_resources
    files:
      protocol: payload.config_path
    comment: Mix colors A, B, C, and D portions according to input data
```

Experiment Application

Submits one or more Workflows to WEI and encodes any logic required to manage and run the experiment

Define the Experiment object that will communicate with the WEI server

Define path to the Workflow definition YAML file

Run the Workflow

Fetch the result and save it in our local directory

```python
#!/usr/bin/env python3
""" Experiment Application """

from wei import ExperimentClient

def main() -> None:
    exp = ExperimentClient("localhost", "8000", "Example_Program")

    wf_path = "example_workflow.yaml"

    flow_info = exp.start_run(
        wf_path.resolve(),
        payload={
            "wait_time": 5,
            "file_name": "experiment_output.jpg",
        },
    )

    exp.get_file(
        flow_info["hist"]["Take Picture"]["action_msg"],
        "experiment_output.jpg",
    )

if __name__ == "__main__":
    main()
```

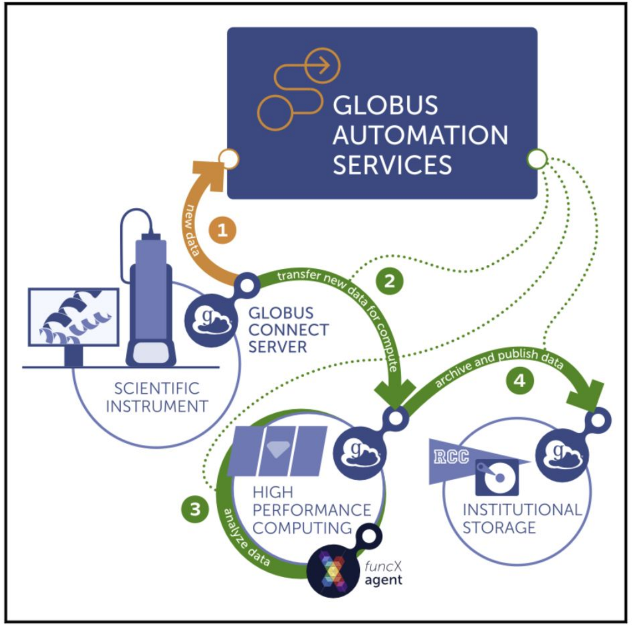
We need to link experimentation to computation and data management

- Cloud-hosted Globus services make it easy to orchestrate actions at 1000s of institutions
- Link with data storage
- Enables rapid analysis

Patterns

Linking scientific instruments and computation: Patterns, technologies, and experiences

Graphical abstract



Highlights

Authors

Rafael Vescovi, Ryan Chard, Nickolaus D. Saint, ..., Nicholas Schwarz, Kyle Chard, Ian T. Foster

Correspondence

foster@anl.gov

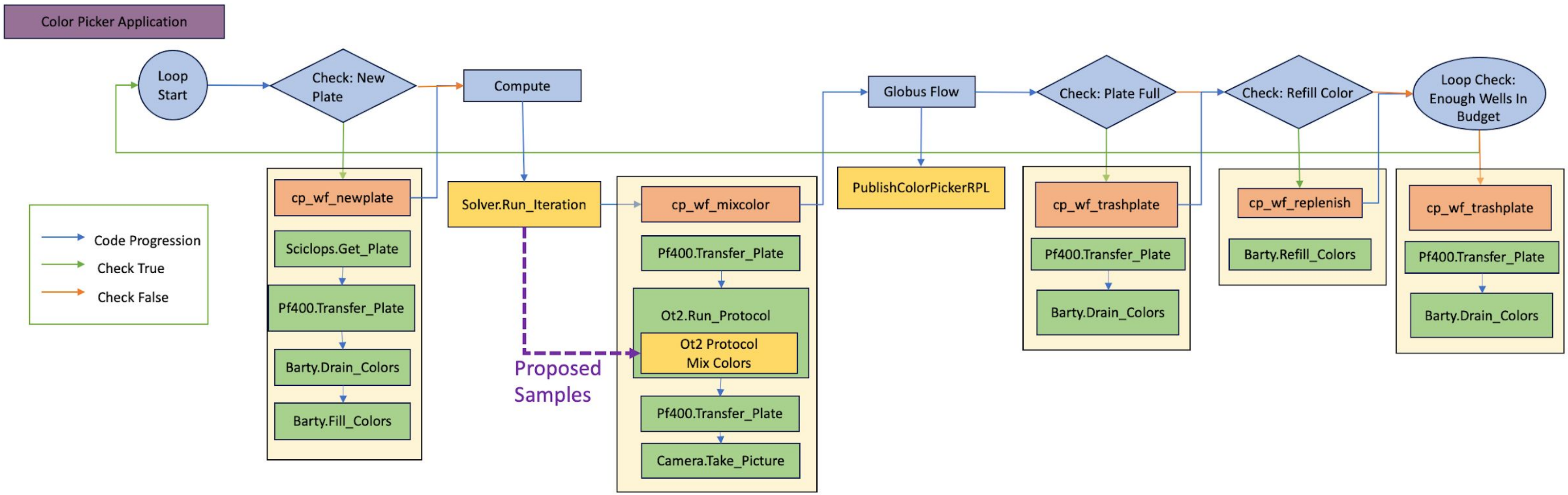
In brief

We review patterns associated with computational flows that link scientific instruments with computing, data repositories, and other resources. We describe methods for implementing such flows and present use cases in which these methods are applied to process data from five different scientific instruments, each of which engages powerful computers for data inversion, machine-learning model training, and other purposes. We also discuss implications of such methods for operators and users of scientific facilities.

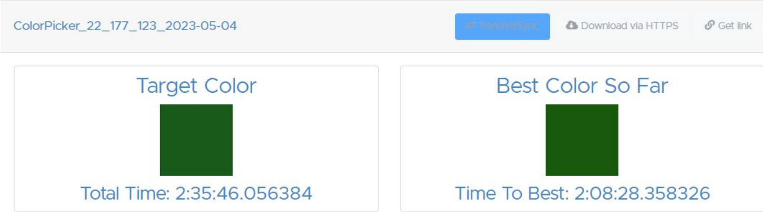
10.1016/j.patter.2022.100606

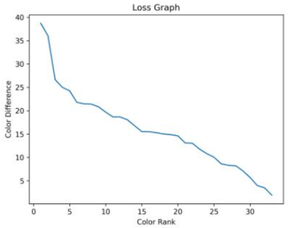
Color Picker Application

Autonomously mixes printer inks and analyzes an image of the results in a loop to recreate a target color



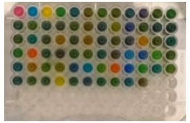
Data from closed-loop "color-picker" application are recorded automatically at ACDC.alcf.anl.gov

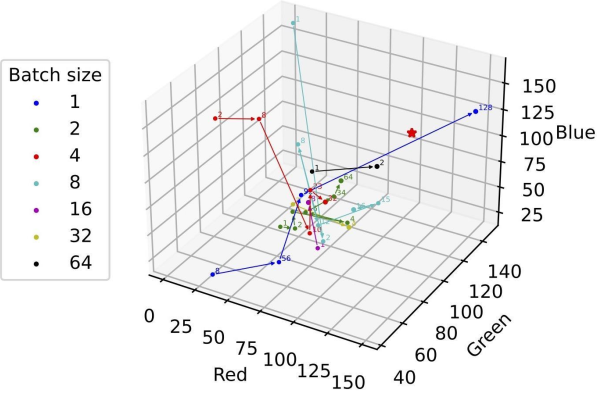




Target Color	[25, 89, 25]
Best Color So Far	[25, 89, 13]
Minimum Difference So Far	1.936329854704391
Total Plates	1
Total Iterations	33
Population Size	2

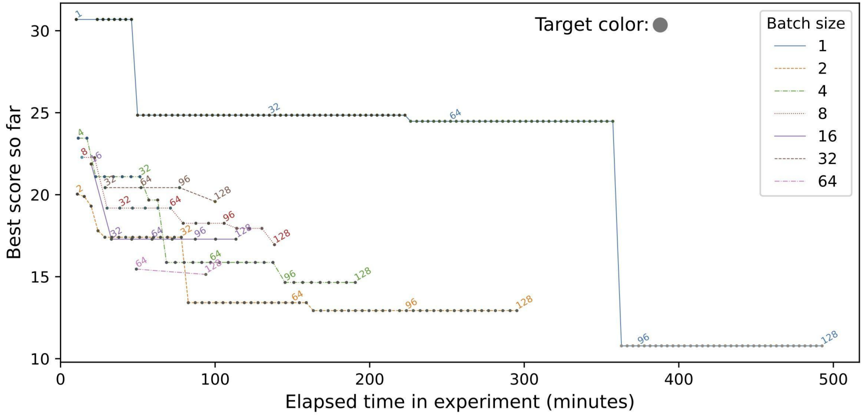
Plate 1





Analyze progress in color space

Analyze progress over time



Current Applications

Autonomous Discovery Exemplars

- Autonomous Protein and Biomimetic Design
- Design of antimicrobial peptides and small molecules
- Discovery of redoxmers for batteries and energy storage
- Development of circular plastics through waste

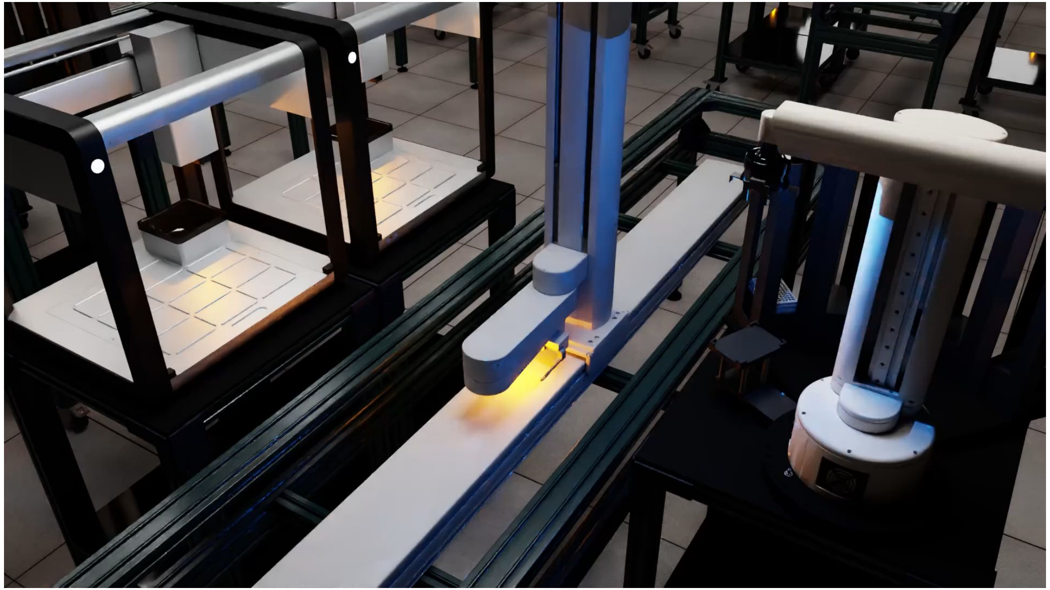
BSL-2 Workcell in Biosciences Division

Future Challenges and Opportunities

Software Development

- Coordination of multiple Workcells
- Scheduling and optimization of Applications
- Error handling and recovery
- AI generation of Application running instructions

Workforce Development!



Rafael Vescovi

Casey Stone

Doga Ozgulbas

Ryan Lewis

Tobias Ginsburg

Abe Stroka

Rory Butler

Mark Hereld

Kyle Hippe

Priyanka Setty

Gyorgy Babnigg

Ben Blaiszik

Arvind Ramanathan

Tom Brettin

Ian Foster

This material is based upon work supported by the Mobilizing the Emerging Diverse AI Talent (MEDAL) project under the Advanced Scientific Computing Research, and Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

THANK YOU

Casey Stone
Computational Scientist
Data Science and Learning Division
Argonne National Laboratory
cstone@anl.gov