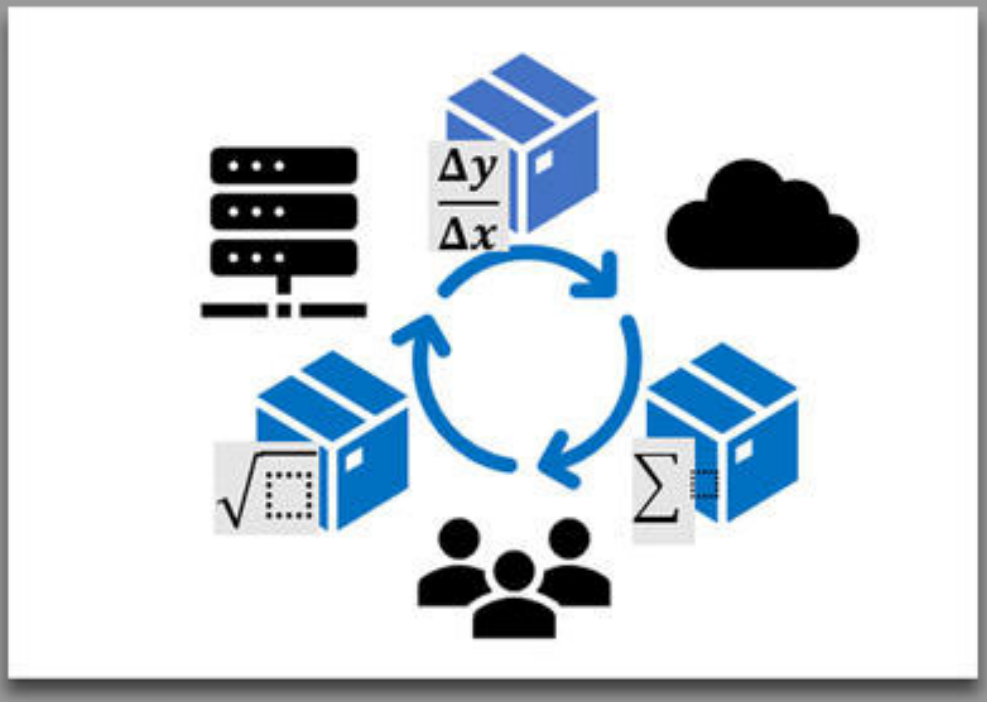
2nd International Workshop on FAIR Containerized Computational Software

December 5-7, 2023



NIST | NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Welcome to the Day One of the 2nd International Workshop on FAIR Containerized Computational Software

Co-organized by National Institute of Standards and Technology (NIST) and National Center for Advancing Translational Sciences (NCATS) at National Institutes of Health (NIH)

Peter Bajcsy
NIST

Nathan Hotaling
NCATS NIH

FAIR Containerized Computational Software

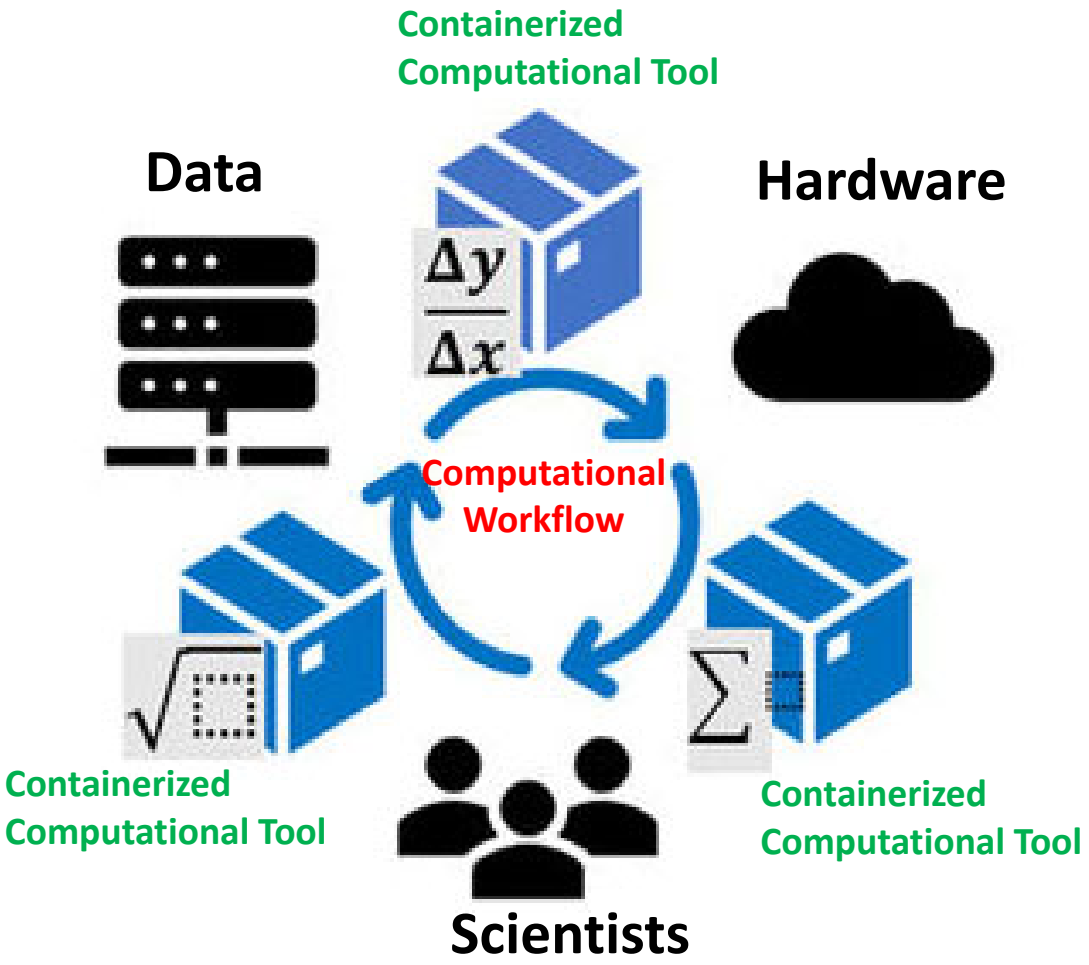
Digital assets: computational software in sciences

Containerization: packaging of software applications so that the software can run in any computational environment

FAIR: Findable, Accessible, Interoperable, and Reusable

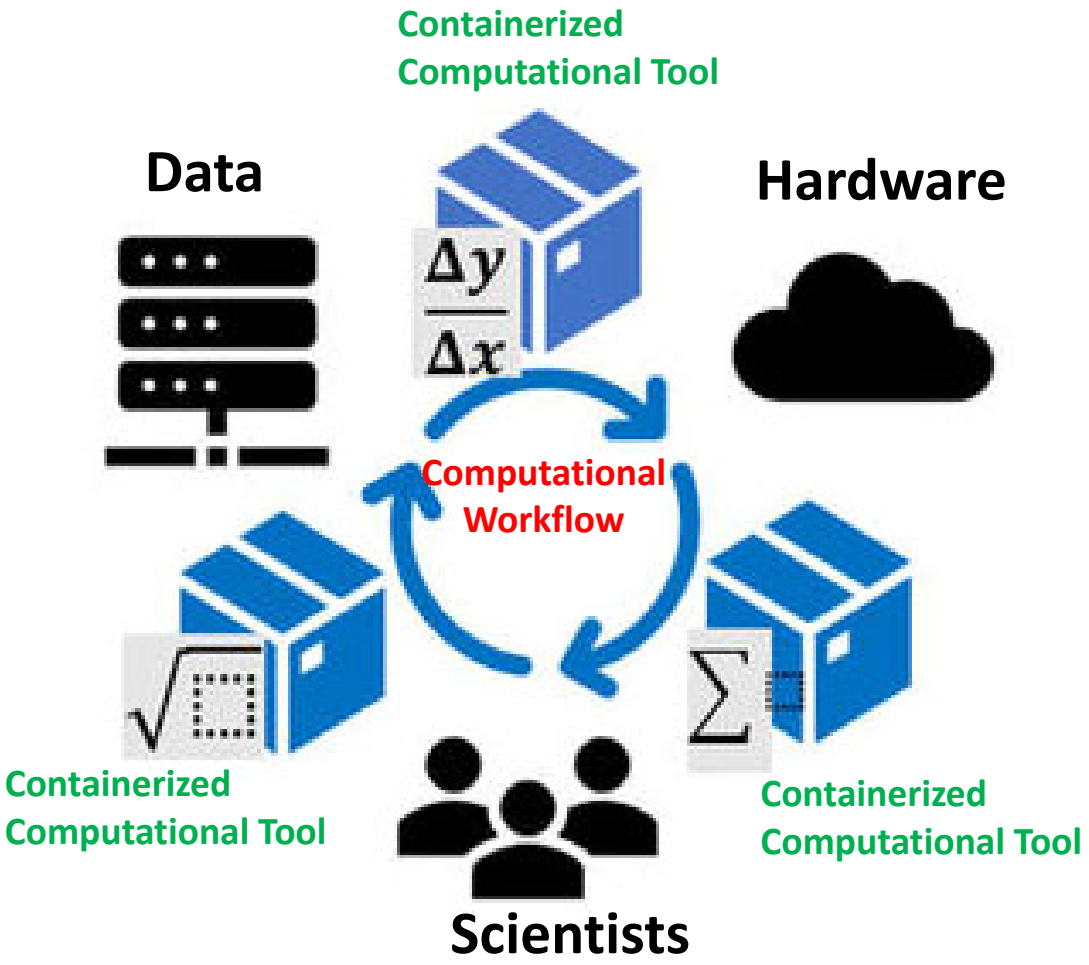
Main goal

The main goal for the workshop is to establish a community consensus on creating interoperable containerized computational tools that can be chained into scientific workflows/pipelines and executed over large image collections regardless of the cloud infrastructure components.



Approach to Forming Computational Workflows

- Create a manifest file accompanying each containerized software tool
- Specify fields in the manifest file for
  - **Inputs/Outputs**
  - **Security**
  - **GUI**
  - **Hardware Requirements**

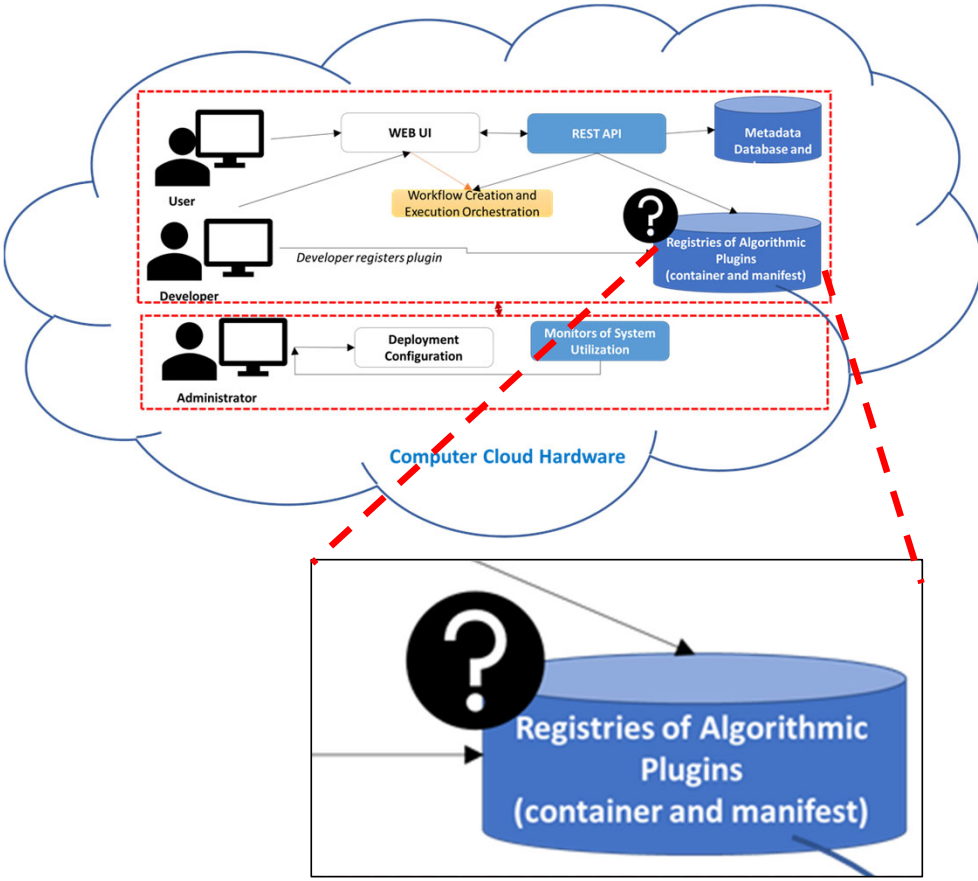


The purpose of the workshop is:

1. establish fields in a manifest file accompanying each containerized software tool (primary purpose),
2. summarize best practices for containerization of algorithms and interfaces between containerized algorithms and datasets in heterogeneous storage environments,
3. explore application programming interfaces (APIs) for finding containerized software tools and container-based workflows in registries,
4. support executions of container-based workflows in a variety of workflow engines,
5. discuss tooling and governance to support a large community of users.

Past Work

- Hosted the 1st Workshop on Interoperability of Web Computational Plugins for Large Microscopy Image, 2019, URL
- The workshop discussions focused on:
  - 1. Containerization of execution code
  - 2. Data storage
  - 3. Interoperability requirements of workflow engines for running containerized plugins
  - 4. Standard packaging of web user interface modules
  - 5. Security of container-based distribution.



Workshop Conclusions from the 1st Workshop

Table 1: Workshop conclusions ranging from practical recommendations to achieved consensus on development and future research and discussions

"Practice:" Workshop Recommendation	"Development:" Consensus Possible For	"Research:" More Discussion Needed to
Use the on-line best practices for containerization of software [1], [2].	Launching algorithms with parameters using error handling for the "run and then destroy" container scenario.	Construct complex user interfaces for collecting parameters passed to container executions.
Adopt specifications for the metadata manifest files from parallel efforts, e.g. [3], [4] .	Application programming interface (API) to access a spectrum of file formats.	Specify the access API for file formats and capture computational provenance, e.g., [7].
Store plugin manifest metadata files in GitHub and leverage existing tools for storing execution profiles and errors e.g., [5], [6].	The information in plugin manifests to include parameters and execution requirements using a predefined schema.	Define the methods that would assist in finding plugins and analyze execution log profiles.
Build UI modules that would consistently support data types found on CPU and GPU hardware platforms.	Creation of reusable UI modules for basic data types (string, int, …).	Create reusable UI modules for complex data types (arrays, …).
Follow the NIST report entitled "Application Container Security Guide" [1].	Security of plugin execution being addressed by plugin authors signing a Docker image.	Understand how to design container security scanning tools, and how to integrate them CI/CD workflows.

Between the 1st workshop and now:

- Workflow reproducibility testing between NIST and NCATS NIH
- Plugin registry deployment at NIST and NCATS NIH
- Pandemic

New needs for extending the plugin manifest specification based on test deployments

Structure of the 2nd International Workshop on FAIR Containerized Computational Software

Workshop Structure

Themes for each day:

December 5 (Day 1): Inputs/Outputs and Security for FAIR containerized computational software

December 6 (Day 2): Graphical User Interfaces for FAIR containerized computational software

December 7 (Day 3): Hardware Requirements for FAIR containerized computational software

Top level program outline (Each day):

Session 1: General session consisting of opening remarks and background introduction (one hour)

Session 2: Five breakout sessions consisting of about 15-30 people discussing specified topics (two hours)

Session 3: General session consisting of summaries from breakout sessions and closing remarks (one hour)

The Workshop Flow

- Session 1: Main Zoom room
  - After the introduction to the theme of the day, NIST conference facility staff will move participants to breakout sessions to have a uniform representation of organizations across all breakout sessions
- Session 2: Breakout Zoom room
  - Moderators and scribes will go over a set of questions/topics to be discussed. The topics map to the those posted at Federal Registry. The set of questions/topics is the same for all breakouts.
- Session 3: Main Zoom room
  - The moderators and scribes report a set of unique answers/solutions for each question/topic and the results of polls

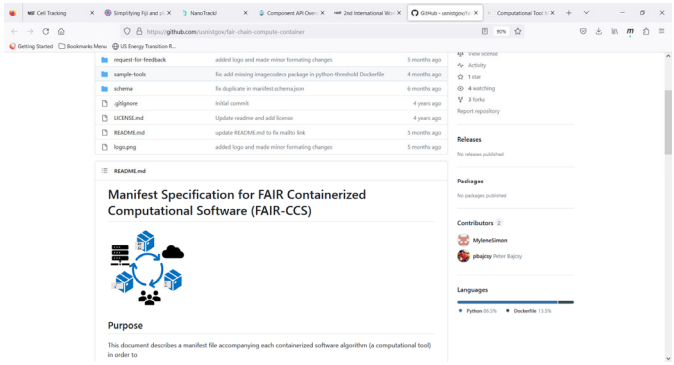
Time and Information Resources

Everyday Meeting Times:

US East Coast (11am-3pm),
US West Coast (8am-noon),
UK (4pm-8pm),
Germany (5pm-9pm),
Korea/Japan (1am-5am)

Comments on the topics outside of the workshop:
https://www.federalregister.gov/documents/2023/08/24/2023-18263/request-for-information-regarding-file-specification-for-findable-accessible-interoperable-and

Workshop event URL: https://www.nist.gov/news-events/events/2023/12/2nd-international-workshop-fair-containerized-computational-software



The GitHub repository with the current specification:
https://github.com/usnistgov/fair-chain-compute-container

Registries of manifests adhering to the current specification:
https://wipp-plugins.nist.gov/
https://wipp-registry.ci.ncats.io/

The Workshop Information

- During the workshop:
  - The workshop is not recorded.
  - The summary notes and chats from breakout sessions will be used for preparing the workshop report.
- After the workshop:
  - Peter Bajcsy (NIST) and Nathan Hotaling (NCATS NIH) will work on the workshop report.
  - The workshop report will be shared with all registered participants for feedback before it will be disseminated to the public.
  - The GitHub repository https://github.com/usnistgov/fair-chain-compute-container will be updated with the workshop report.

Additional Topics

- Use cases
- Ecosystem of tools to support seamless integration to scientific research
- Educational materials
- Benchmarking information
- Governance

December 5, 2023 (Day 1)

Theme: Inputs/Outputs and Security for FAIR containerized computational software

Agenda

- Session 1: Introduction
  - Peter Bajcsy (NIST) – workshop goals
  - Mylene Simon (NIST) – current manifest file
  - Nathan Hotaling (NIH) –development of containerized computational tools at NIH
  - Michael Bartock (NIST)– security of container platforms
- Session 2: Breakouts

Date	Breakout	Moderator	Scribe
5-Dec	1	Peter Bajcsy	Sameeul Samee
5-Dec	2	Nathan Hotaling	Derek Juba
5-Dec	3	Mylene Simon	Kevin Duerr
5-Dec	4	Nick Schaub	Pushkar Sathe
5-Dec	5	Sunny Yu	Guillaume Sousa

- Session 3: Summary