INFRASTRUCTURE FOR HIGH-FIDELITY TESTING IN HPC FACILITIES

Ryan Prout

Intro/Bio

- Group Lead at Oak Ridge Leadership Computing Facility (OLCF) – Software Services Development Group
- Software deployment initiative lead in the Exascale Computing Project (ECP)
- Historically a support engineer in the Oak Ridge Computing Facility (OLCF)
- Interested in developing/providing tools and infrastructure for future workflow systems

Name: Ryan Prout

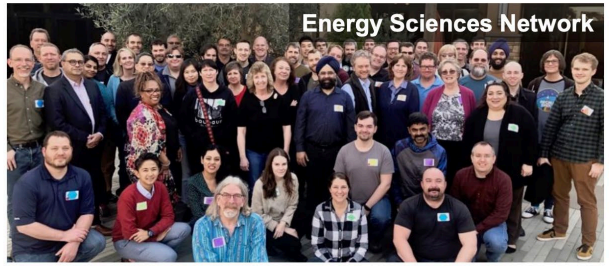
Email: proutrc@ornl.gov

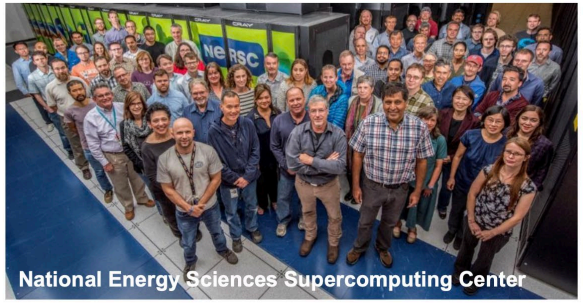
Goals of this webinar

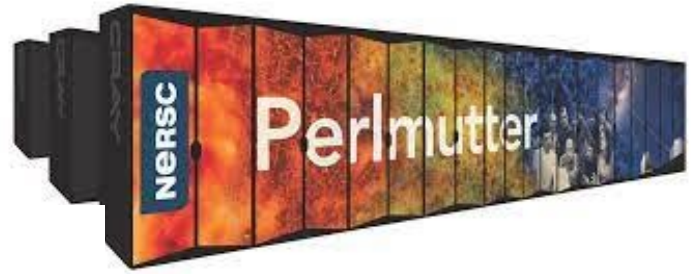
- Build momentum in creating common CI services across Advanced Scientific Computing Research (ASCR) facilities - drawing links to the Department of Energy (DOE) Integrated Research Infrastructure (IRI) initiatives
- Boost the experience for scientific software testing, integration, and deployment at HPC facilities
- Review how ongoing goals in ECP Software Deployment efforts are pushing Continuous Integration (CI) services forward in HPC environments
- Look at how providing high-fidelity environments, for testing and integration at HPC facilities, can boost scientific software development
- **Generate ongoing discussion about how to support DevOps and Research Software Engineering efforts at HPC facilities**

What are the ASCR Facilities?

The people of the ASCR Facilities: Providing high performance Research Computing, Data, and Networking for DOE and the Nation







Facilities

The High Performance Computing and Network Facilities subprogram supports the operations of forefront computational and networking user facilities to meet critical mission needs. ASCR operates three high performance computing (HPC) user facilities: the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory (LBNL), which provides high performance computing resources and large-scale storage to a broad range of SC researchers; and the two Leadership Computing Facilities (LCFs) at Oak Ridge National Laboratory (ORNL) and Argonne National Laboratory (ANL), which provide leading-edge high performance computing capability to the U.S. research and industrial communities. ASCR's high performance network user facility, the Energy Sciences Network (ESnet), operated by LBNL, delivers highly reliable data transport capabilities optimized for the requirements of large-scale science.

What is the Exascale Computing Project (ECP)?

Application Integration at the Facilities

Application integration helps ensure that Exascale Computing Project (ECP) applications are properly prepared for the forthcoming US Department of Energy (DOE) exascale systems. It also ensures that the DOE HPC facilities (hereafter referred to as Facilities) are aware of ongoing application development efforts and can contribute their experience and expertise in porting applications to the Facility-specific architectures.

Details +

Facility Resource Utilization

Producing an exascale-capable environment requires access to high-performance computing (HPC) resources for the purpose of application and software testing and development. Facility Resource Utilization is a collaboration between the Exascale Computing Project (ECP) and US Department of Energy (DOE) HPC facilities to ensure effective access to current production systems and pre-exascale and exascale computers.

Details +

Hardware Evaluation

Hardware Evaluation (HE) staff provide independent architectural analyses of exascale hardware components—even those that have not yet materialized—by using a variety of analysis tools and approaches. Their findings inform system design, highlight opportunities and challenges for scientific applications, and provide understanding of observed performance and efficiency from a hardware and software perspective.

Details +

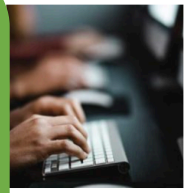
PathForward

PathForward seeks to prepare the US industry for exascale system procurements and generally improve US competitiveness in the worldwide computing market. Evidenced by CORAL2 offerings, PathForward successfully influences US high-performance computing (HPC) vendors to maintain focus on designing and building much more powerful and balanced computers for US Department of Energy simulation workloads.

Details +

Software Deployment at the Facilities

Through close partnership with ECP code teams, DOE HPC Facilities, and vendors, the Software Deployment team deploys and integrates an exascale software stack and deploys a software integration and testing capability at the Facilities to support continuous integration with site environments, including container technologies and software development kits.



Training and Productivity

For applications to take full advantage of exascale hardware and software, a robust developer training and productivity program keeps application and software team members, staff, and other stakeholders abreast of emerging technologies and key technologies of importance to ECP. These projects are done in close collaboration among the computing facilities, vendors, and the ECP community.

Strategic partnership between two DOE Organizations (Office of Science and NNSA), formed in 2016, to accelerate research, development, acquisition, and deployment projects to deliver exascale computing capabilities to the DOE laboratories.

https://www.exascaleproject.org/

What is Integrated Research Infrastructure (IRI)?

DOE's Integrated Research Infrastructure (IRI) Vision:
To empower researchers to meld DOE's world-class research tools, infrastructure, and user facilities seamlessly and securely in novel ways to radically accelerate discovery and innovation

IRI Practice Areas (6)

User experience practice will ensure relentless attention to user perspectives and needs through requirements gathering, user-centric (co)-design, continuous feedback, and other means.

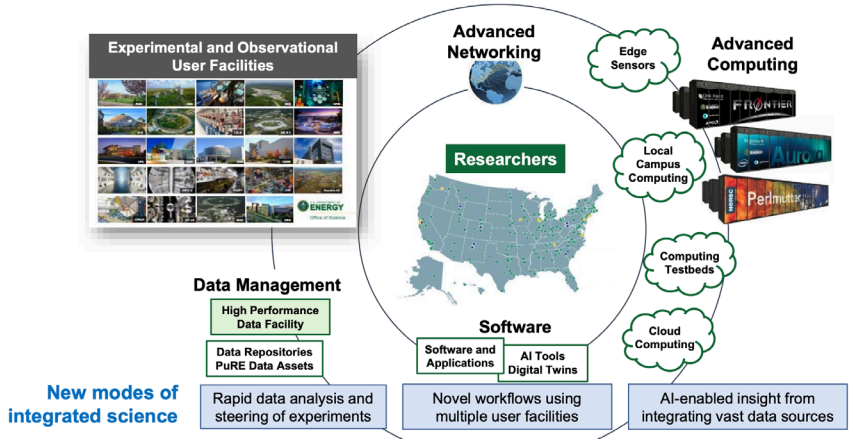
Resource co-operations practice is focused on creating new modes of cooperation, collaboration, co-scheduling, and joint planning across facilities and DOE programs.

Cybersecurity and federated access practice is focused on creating novel solutions that enable seamless scientific collaboration within a secure and trusted IRI ecosystem.

Workflows, interfaces, and automation practice is focused on creating novel solutions that facilitate the dynamic assembly of components across facilities into end-to-end IRI pipelines.

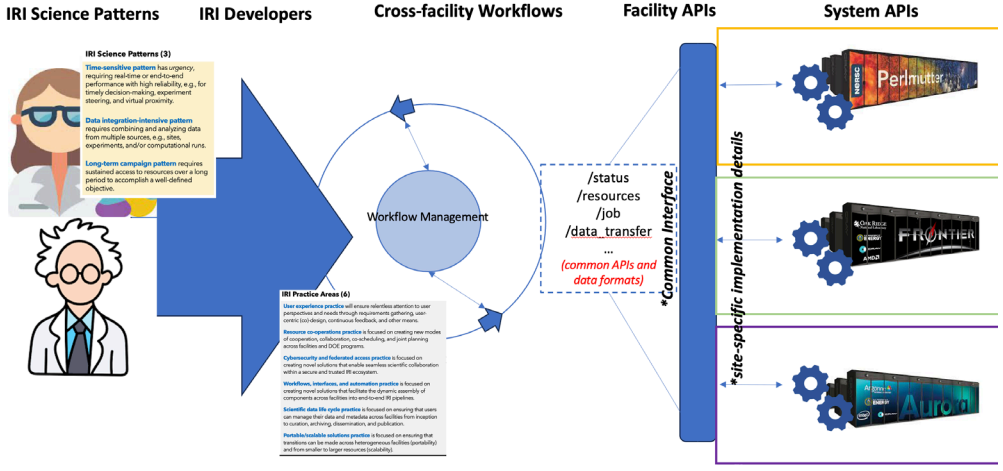
Scientific data life cycle practice is focused on ensuring that users can manage their data and metadata across facilities from inception to curation, archiving, dissemination, and publication.

Portable/scalable solutions practice is focused on ensuring that transitions can be made across heterogeneous facilities (portability) and from smaller to larger resources (scalability).



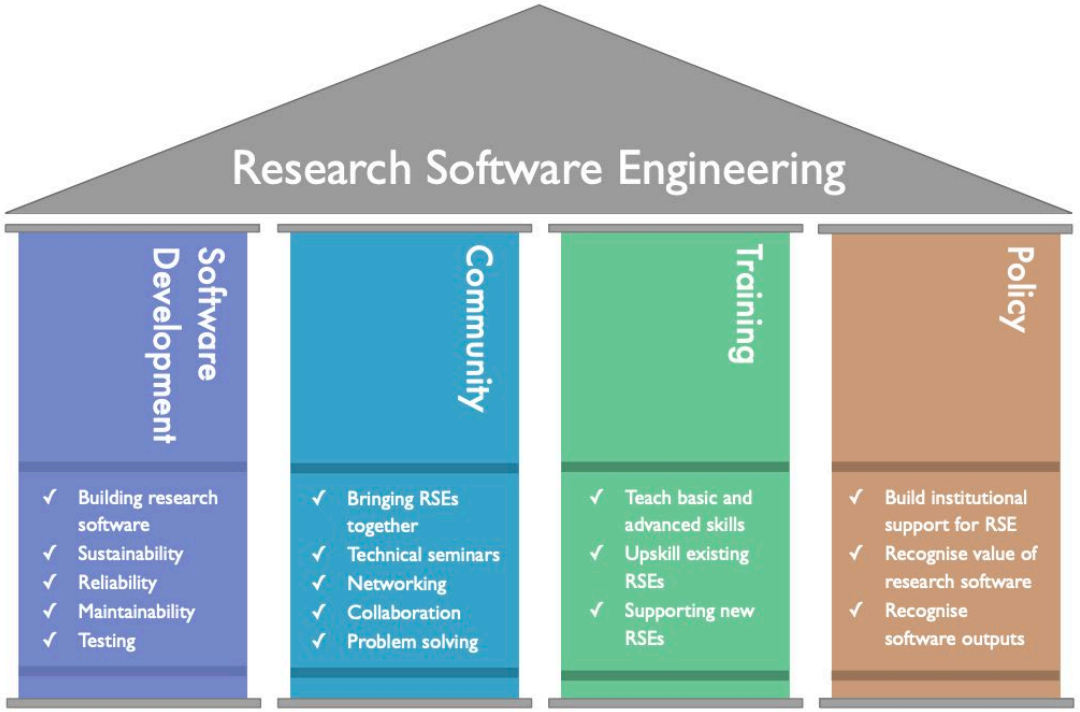
Common interfaces/services across labs and systems

Facility APIs – Providing Systems to Workflows



https://science.osti.gov/-/media/ascr/ascac/pdf/meetings/202306/Brown_IRI_ASCAC_2023206.pdf

What is Research Software Engineering?



https://arxiv.org/pdf/2002.01035.pdf

What do RSEs do?

Fundamentally, RSEs build software to support scientific research. They generally don't have research questions of their own – they develop the computer tools to help other people to do cool things. They might add features to existing software, clear out bugs or build something from scratch. But they don't just sit in front of a computer and write code. They have to be good communicators who can embed themselves in a team.

What else do RSEs do?

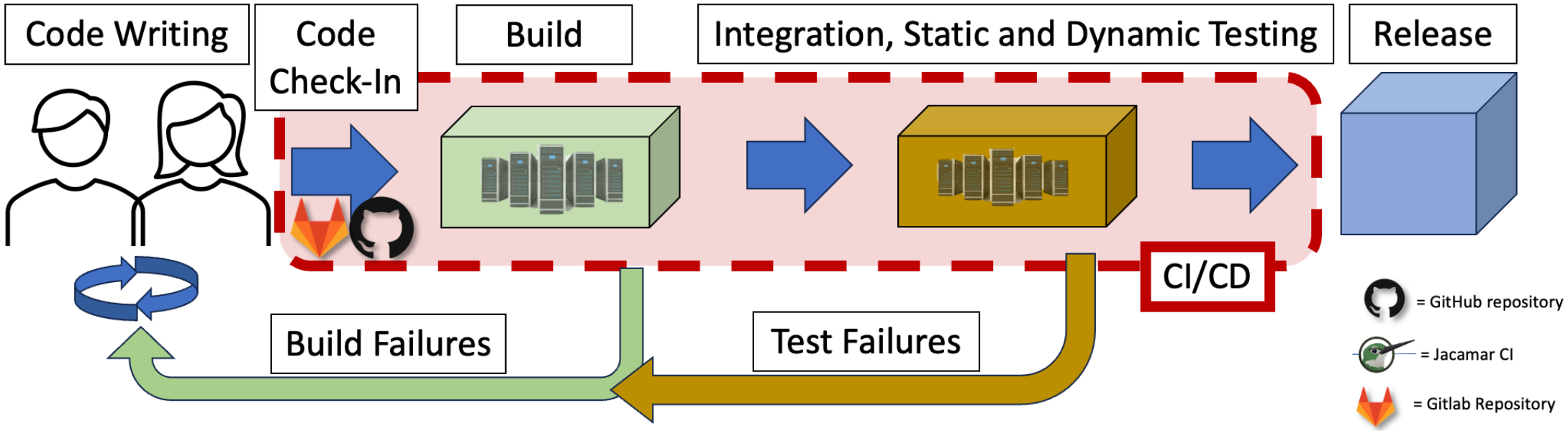
A big part of the job is raising awareness about the importance of quality software. An RSE might train a postdoc or graduate student to develop software on their own. Or they might run a seminar on good software practices. In theory, training 50 people could be more impactful than working on a single project. In practice, it's often hard for RSEs to find the time for teaching, mentorship and advocacy because they're so busy supporting research.

https://www.nature.com/articles/d41586-022-01516-2

- **Enabling RSE practices is important for the future and sustaining scientific software**
- **CI infrastructure is vital to the Software Development pillar**
- **Exascale Computing Project is a great example of embracing these principles within a long running development project**
- **RSE pillars feed future Software Sustainability initiatives: https://www.hpcwire.com/off-the-wire/doe-ascr-selects-seed-collaborations-for-software-sustainability/**

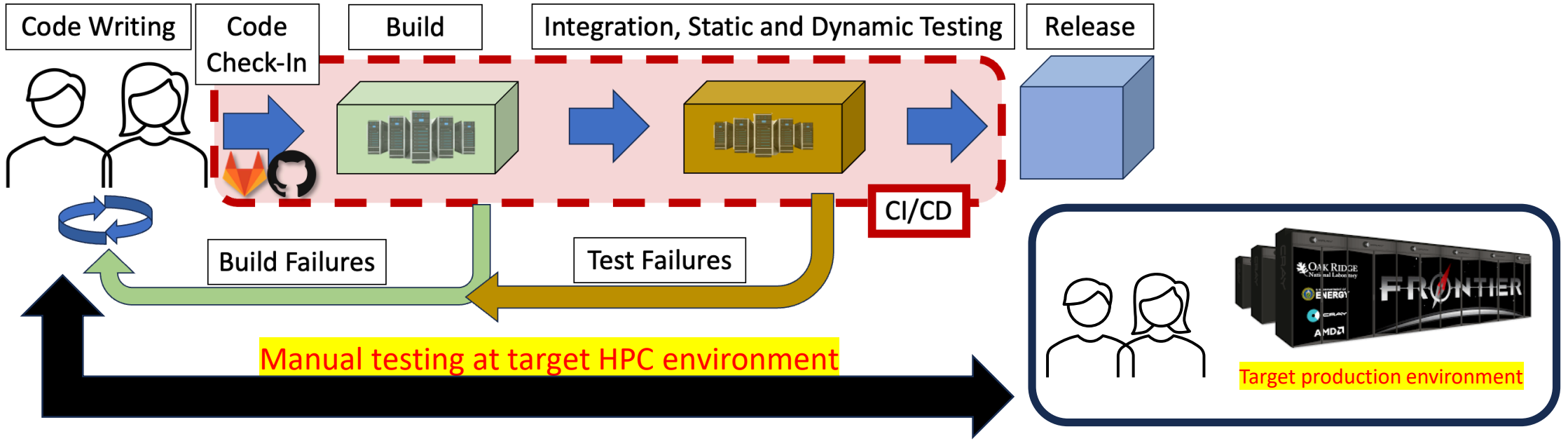
Defining Continuous Integration (CI) and Continuous Deployment (CD)

- Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project
- Continuous Deployment (CD) is the step beyond Continuous Integration in which you automate the deployment of an application to the production environment



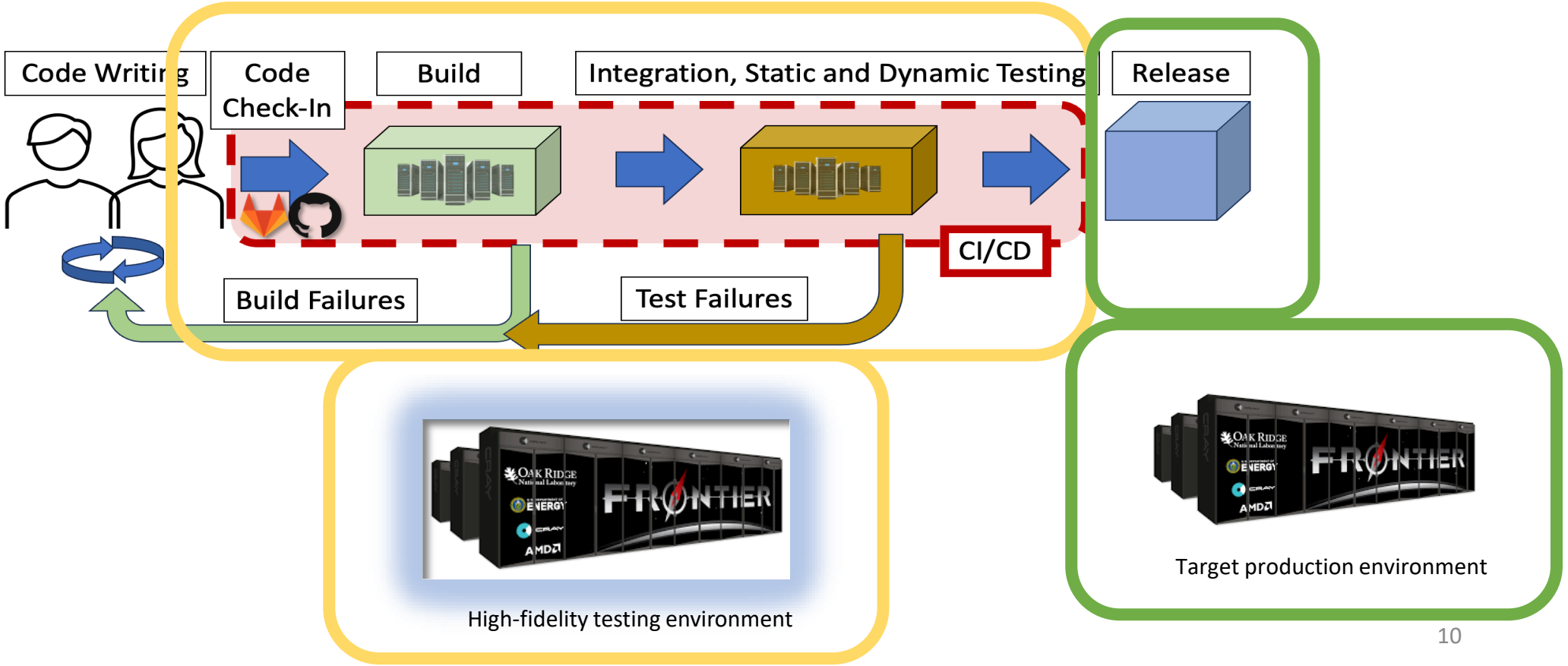
Some challenges of adopting CI for HPC applications

- Cloud-based CI is valuable for some testing, but it doesn't cover everything
- The environment of the target HPC system is hard/impossible to replicate in the cloud
- Testing integration with the HPC system often requires access to system-level packages
- You can't test/confirm performance outside of the HPC environment
- Testing against the HPC system is decoupled from the CI processes

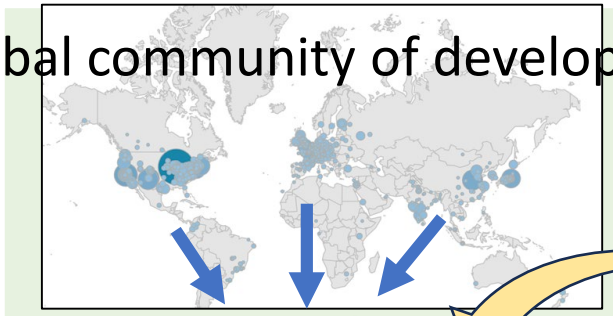


What do we mean by high-fidelity testing infrastructure?

- It is currently difficult for research software engineers to automatically test, integrate, and deploy software to remotely managed computing facilities
- If facilities provided high-fidelity environments, robust CI services could be enabled on top



What a full DevOps workflow might look like for scientific applications targeting HPC facilities



Central/Public Repository

Public SaaS/Cloud-based CI Systems

Continuous Integration Testing

Continuous Deployment

Target production environment provided by the facility

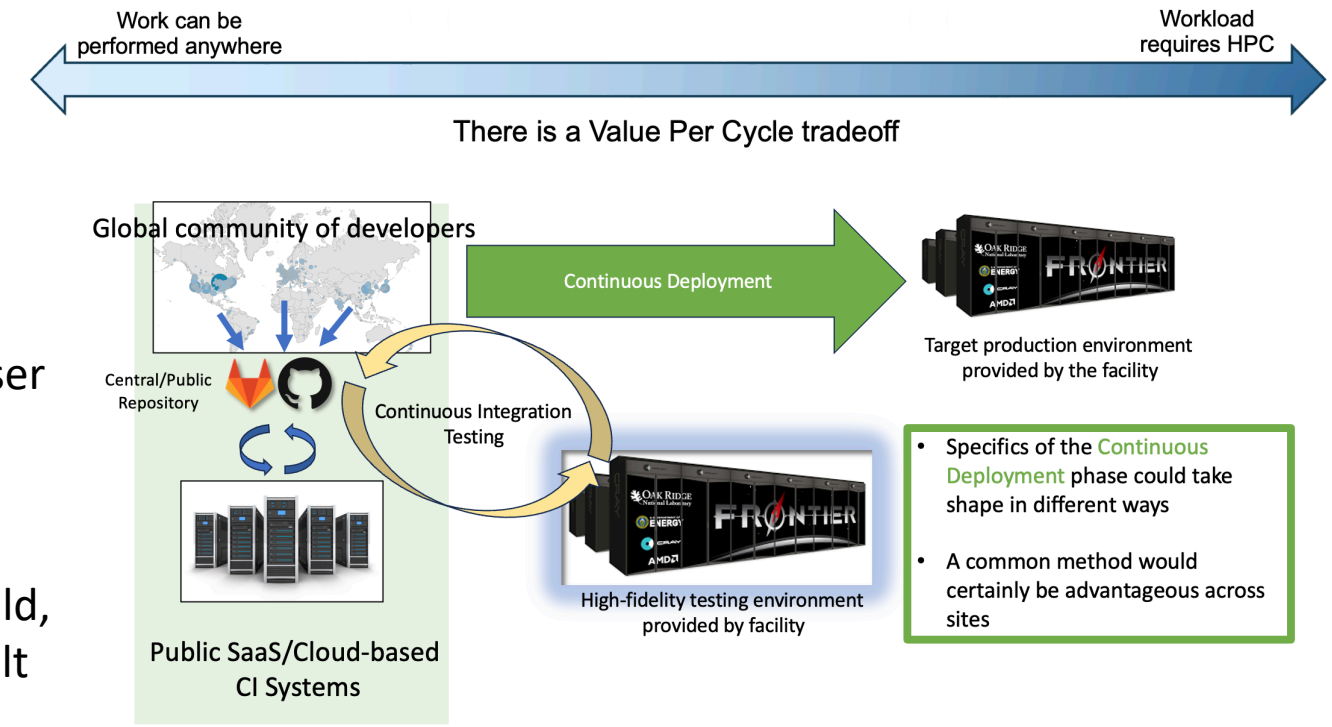
High-fidelity testing environment provided by facility

- Specifics of the Continuous Deployment phase could take shape in different ways
- A common method would certainly be advantageous across sites

Why can’t we do this today – challenges

SECURITY

Resource Utilization



- Triggering automated workflows from external systems is not a widely solved solution in ASCR facilities yet
- ASCR facilities provide computing resources to a global user base and the software is hosted outside the facility boundaries
- Traditionally, staff at the ASCR facilities manually help build, install, integrate software (the DevOps workflow is difficult to integrate with the facility)

Let’s look at where ECP comes in..

Why is DevOps important to the Exascale Computing Project (ECP)?

- ECP is focused on delivering an exascale ecosystem: applications, systems software, hardware technologies, and architectures to ***diverse platforms***

EXASCALE COMPUTING PROJECT

System Specifications

Partition	# of nodes	CPU	GPU	NIC
GPU	1536	1x AMD EPYC 7763	4x NVIDIA A100 (40GB)	4x HPE Slingshot 11
	256	1x AMD EPYC 7763	4x NVIDIA A100 (80GB)	4x HPE Slingshot 11
CPU	3072	2x AMD EPYC 7763	-	1x HPE Slingshot 11
Login	40	1x AMD EPYC 7713	1x NVIDIA A100 (40GB)	-

Specifications and Features

Compute Node:
1 64-core AMD "Optimized 3rd Gen EPYC" CPU
4 AMD Instinct MI250X GPUs
GPU Architecture:
AMD Instinct MI250X GPUs, each feature 2 Graphics Compute Dies (GCDs) for a total of 8 GCDs per node
System Interconnect:
4-port HPE Slingshot 200 Gbps (25 GB/s) NICs providing a node-injection bandwidth of 800 Gbps (100 GB/s)

Aurora System Specifications

Compute Node
2 Intel Xeon CPU Max Series processors: 64GB HBM on each, 512GB DDR5 each; 6 Intel Data Center GPU Max Series, 128GB HBM on each, RAMBO cache on each; Unified Memory Architecture; 8 SlingShot 11 fabric endpoints

CPU-GPU Interconnect
CPU-GPU: PCIe; GPU-GPU: Xe Link

Key initiative in ECP was streamlining software testing and deployment

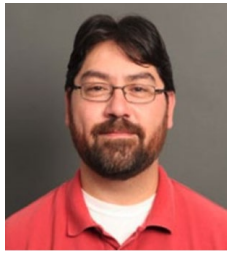
- ***Ongoing goal*** to enable testing and integration workflows, through Gitlab CI, ***directly on the system***
- **Software developers targeting these facilities have difficulty bridging development and deployment workflows to the systems. We want to help remedy that.**
- ***Security and resource constraints*** require continual discussion, with each facility, especially post-ECP

Software Deployment at the Facilities

Through close partnership with ECP code teams, DOE HPC Facilities, and vendors, the Software Deployment team deploys and integrates an exascale software stack and deploys a software integration and testing capability at the Facilities to support continuous integration with site environments, including container technologies and software development kits.

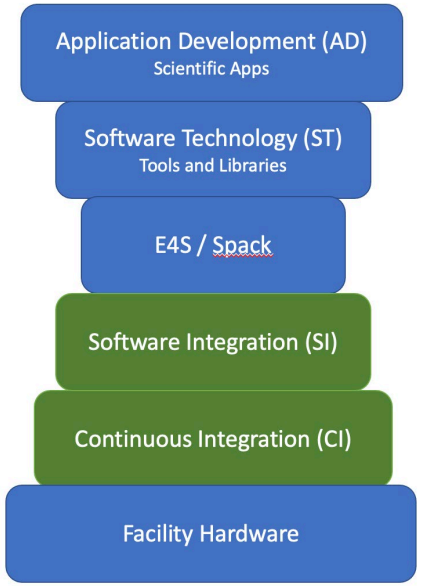
ECP Software Deployment Teams

Team	PI	Short Description/Objective
Software Integration	Shahzeb Siddiqui (LBL)	Build/Test/Deploy ECP products at facilities
Continuous Integration	Paul Bryant (ORNL)	Develop and Deploy ECP CI infrastructure



Combined Purpose:
Provide infrastructure and support for integrating software at facilities (enable DevOps at OLCF, ALCF, and NERSC)

Tools and Infrastructure:
Spack
E4S
Gitlab
Jacamar CI

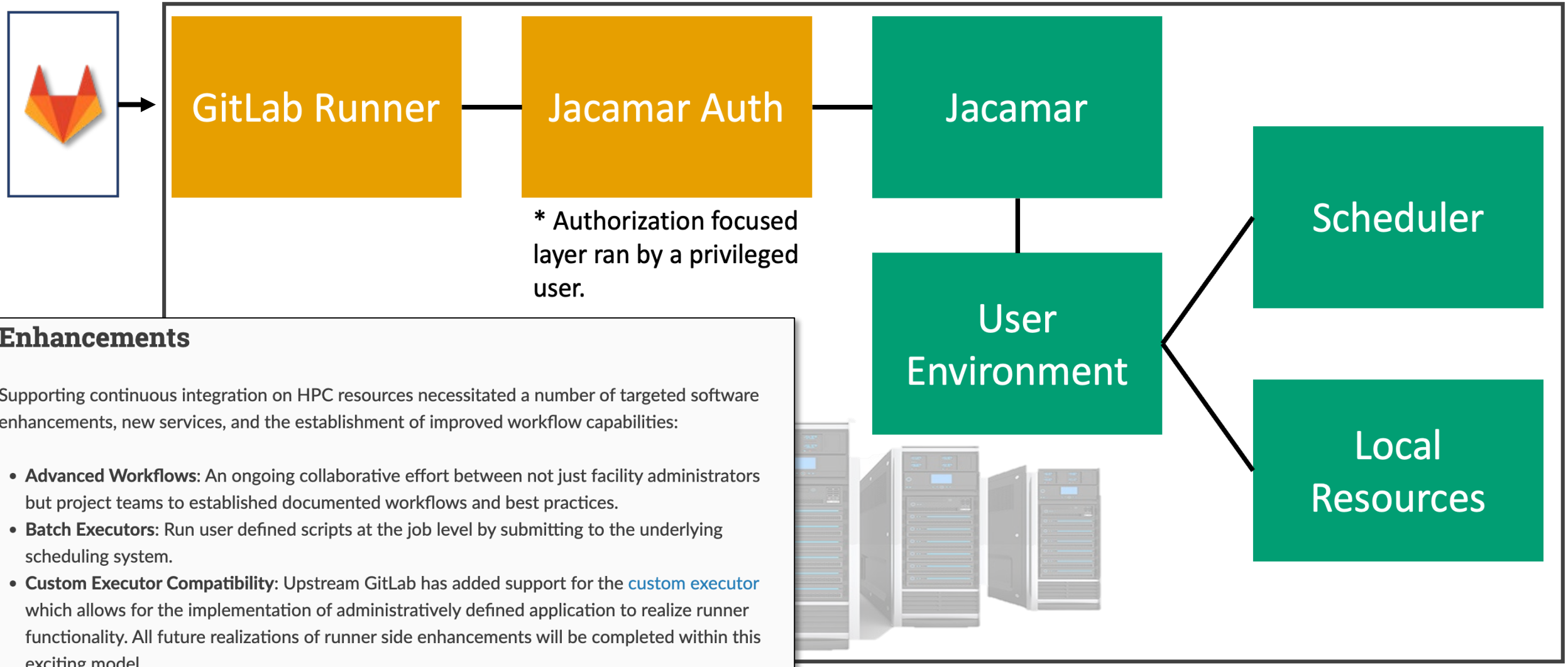


Jacamar CI – Enabling CI at ASCR Facilities for ECP

Primary goal is to provide access to powerful scientific test resources managed by facility expertise for the purposes of CI/CD workflows. While observing and meeting facility specific requirements.

https://ecp-ci.gitlab.io/docs/admin.html#jacamar-ci

How does Jacamar do this?



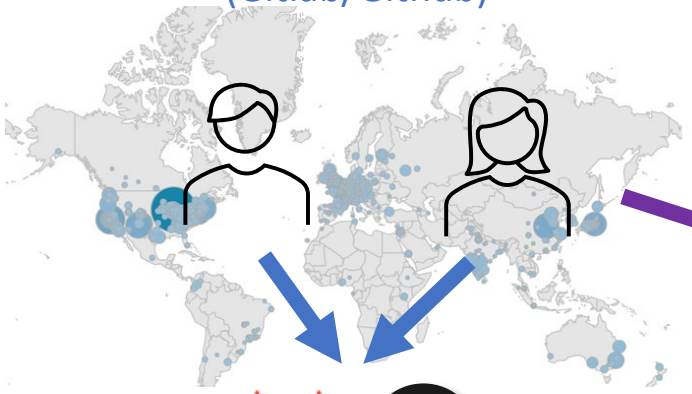
Enhancements

Supporting continuous integration on HPC resources necessitated a number of targeted software enhancements, new services, and the establishment of improved workflow capabilities:

- **Advanced Workflows**: An ongoing collaborative effort between not just facility administrators but project teams to established documented workflows and best practices.
- **Batch Executors**: Run user defined scripts at the job level by submitting to the underlying scheduling system.
- **Custom Executor Compatibility**: Upstream GitLab has added support for the custom executor which allows for the implementation of administratively defined application to realize runner functionality. All future realizations of runner side enhancements will be completed within this exciting model.
- **Downscoping Permissions**: Targeting the validated user responsible for triggering the CI pipeline and dropping permissions prior to job execution to that of their system local account.

ECP Software Testing and Deployment (WIP)

1. Developers utilize PaaS DevOps Platforms to host their central code base (Gitlab/Github)

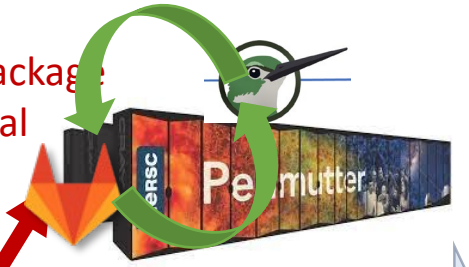


2. Developers setup integration and testing pipelines via PaaS CI mechanisms

3. Developer work with E4S team to integrate with target systems

E4S

4. E4S team mirrors package recipes to site-local repositories



4. E4S team tests packages for release on target systems via Jacamar CI (schedulable pipelines)

https://dashboard.e4s.io/

Facility Deployments - Summary

System	Deployment	Spack Details
Polaris	E4S 23.05, PrgEnv-gnu	/soft/ecp/ParaTools/E4S/23.05/PrgEnv-gnu/spack /soft/ecp/ParaTools/E4S/23.05/PrgEnv-gnu/spack.yaml /soft/ecp/ParaTools/E4S/23.05/PrgEnv-gnu/module-use.sh
Frontier	E4S 23.05, PrgEnv-gnu	/sw/frontier/ums/ums002/E4S/23.05/PrgEnv-gnu/spack /sw/frontier/ums/ums002/E4S/23.05/PrgEnv-gnu/spack.yaml /sw/frontier/ums/ums002/E4S/23.05/PrgEnv-gnu/module-use.sh
Sunspot	E4S 23.05, oneAPI	/lus/gila/projects/CSC250STPM01_CNDA/E4S/23.05/spack /lus/gila/projects/CSC250STPM01_CNDA/E4S/23.05/spack.yaml /lus/gila/projects/CSC250STPM01_CNDA/E4S/23.05/module-use.sh

5. Issues reported back to developers as needed

E4S = Extreme Scale Scientific Software Stack

= GitHub repository

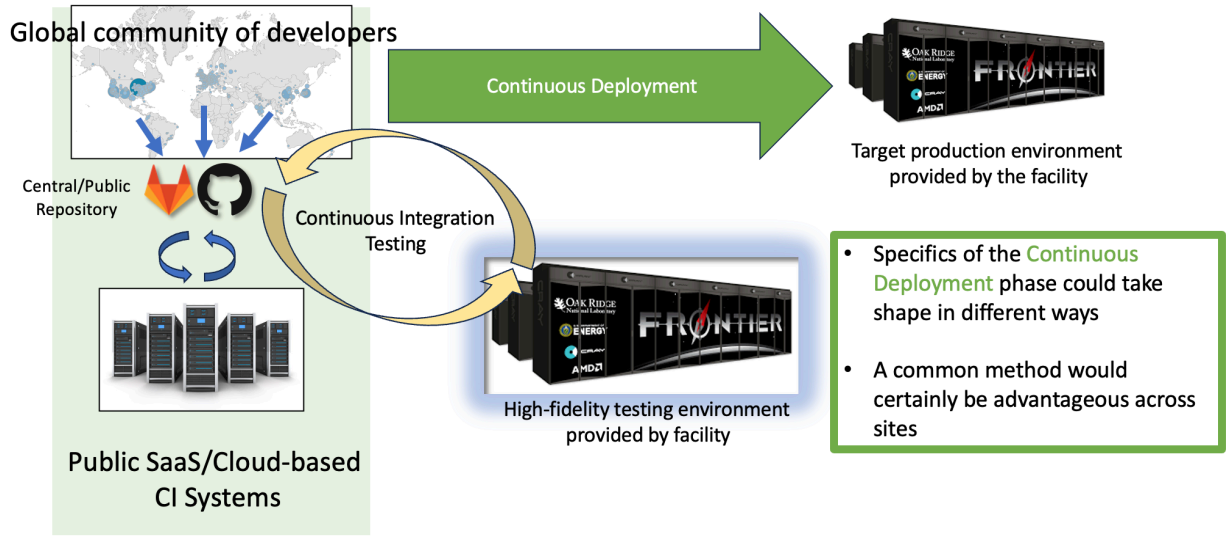
= Jacamar CI

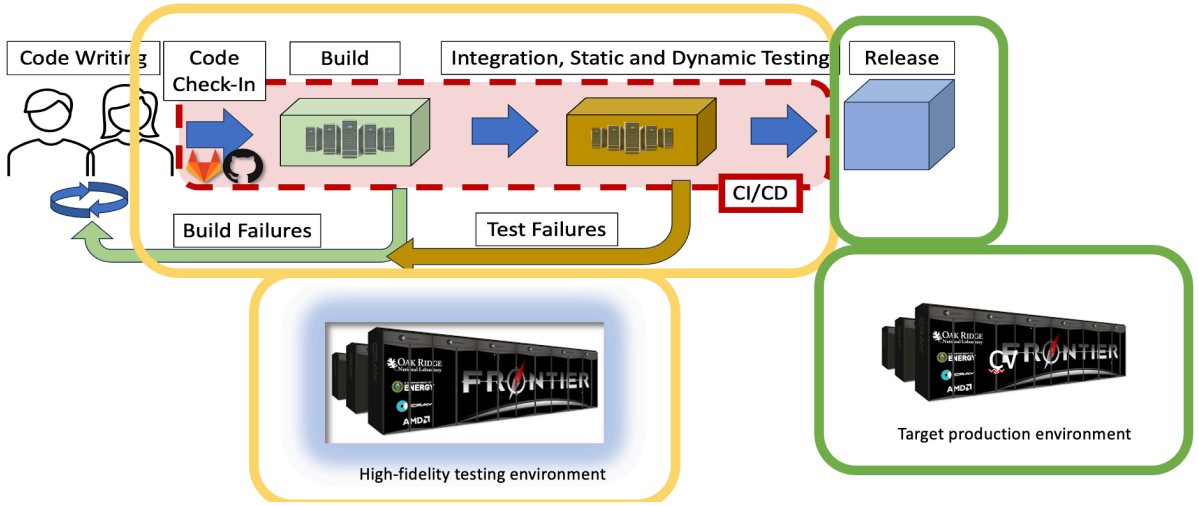
= Gitlab Repository

Why would high-fidelity infrastructure help?

- Enabling open test infrastructure, decoupled from the production system, allows us to work towards providing CI workflows comparable to the cloud (*truly continuous integration – something we have not achieved in ECP*)

- This would be the first step in providing industry standard CI services, directly on facility systems, to facility users



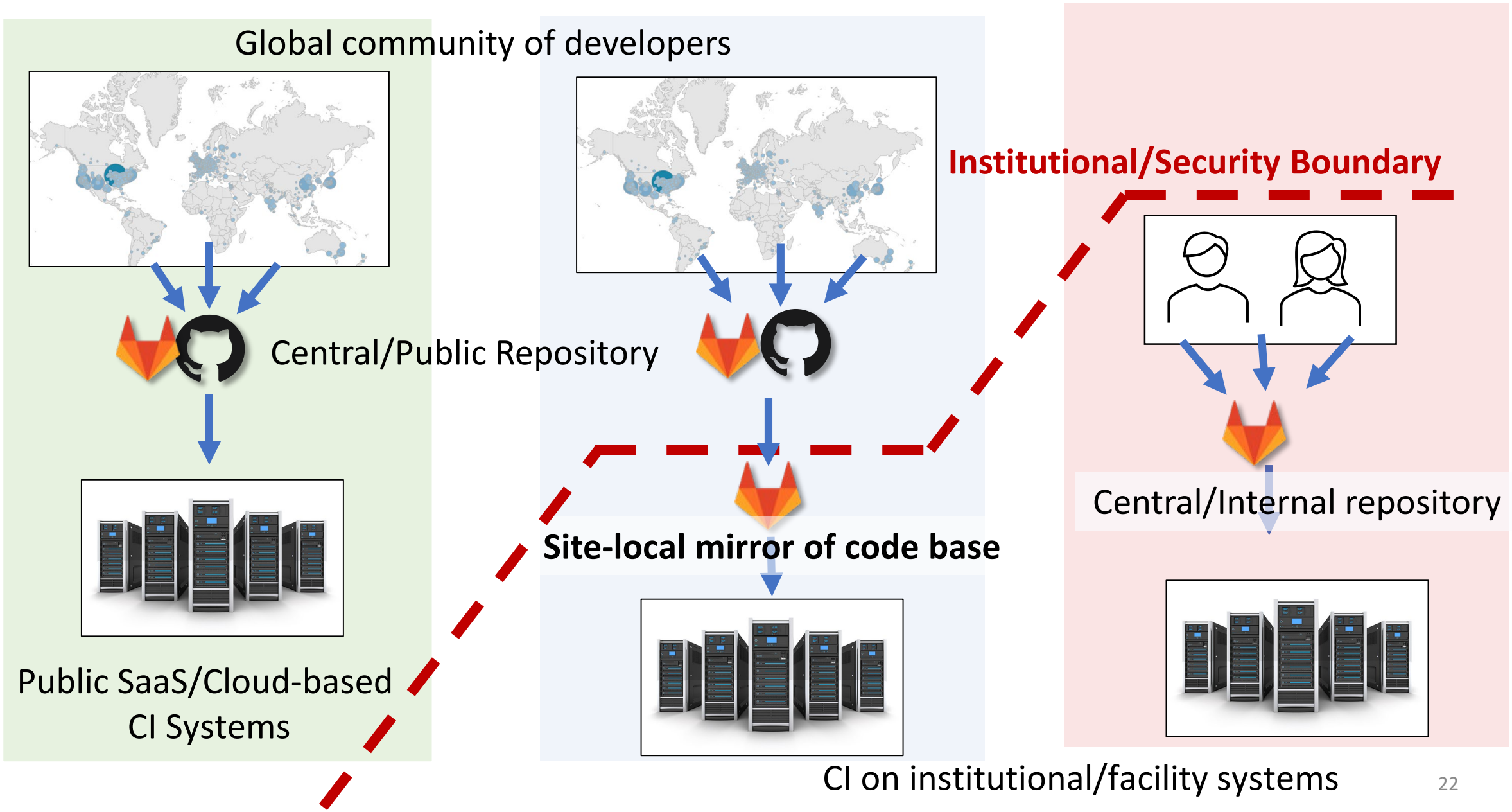


- The testing infrastructure must stay inline with target environment (AKA high-fidelity)

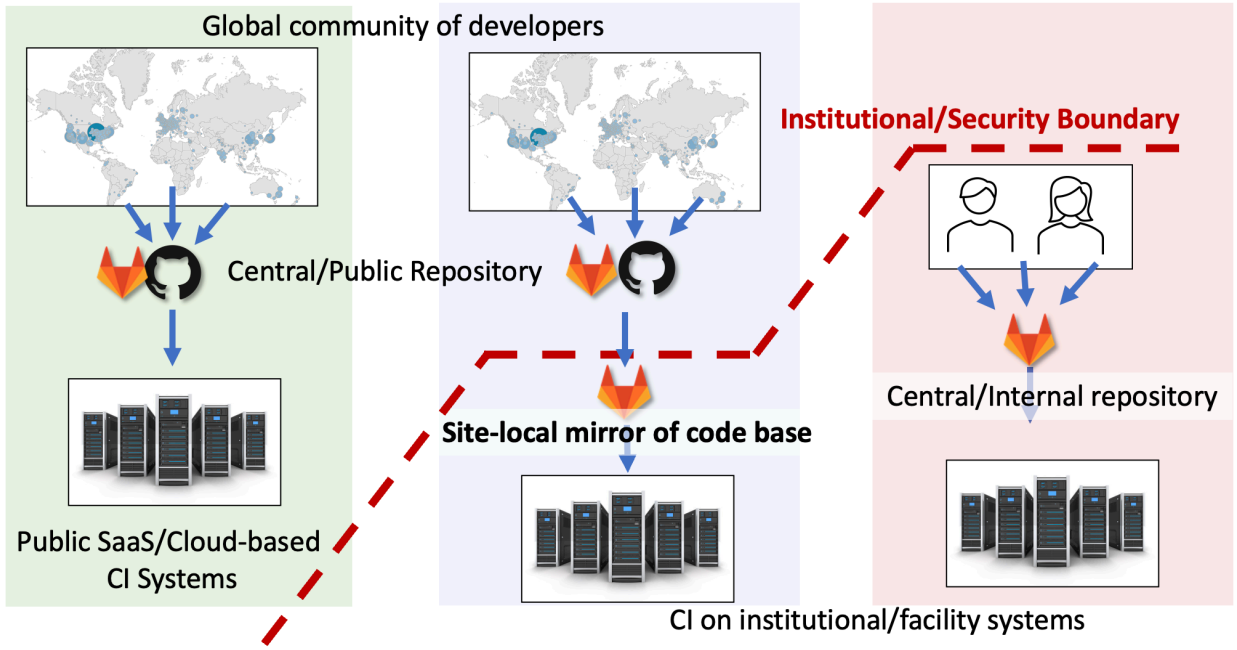
- Each facility would provide this high-fidelity testing infrastructure to relative projects or distribution efforts like E4S

Let’s take a step back and look at different CI scenarios relative to HPC environments..

Common CI Scenarios



Security Perspectives



Scenario ‘A’

- Infrastructure and processes are in total control by the code owners
- Include standard practices to protect your code base, understand dependencies, and testing implementation on SaaS systems

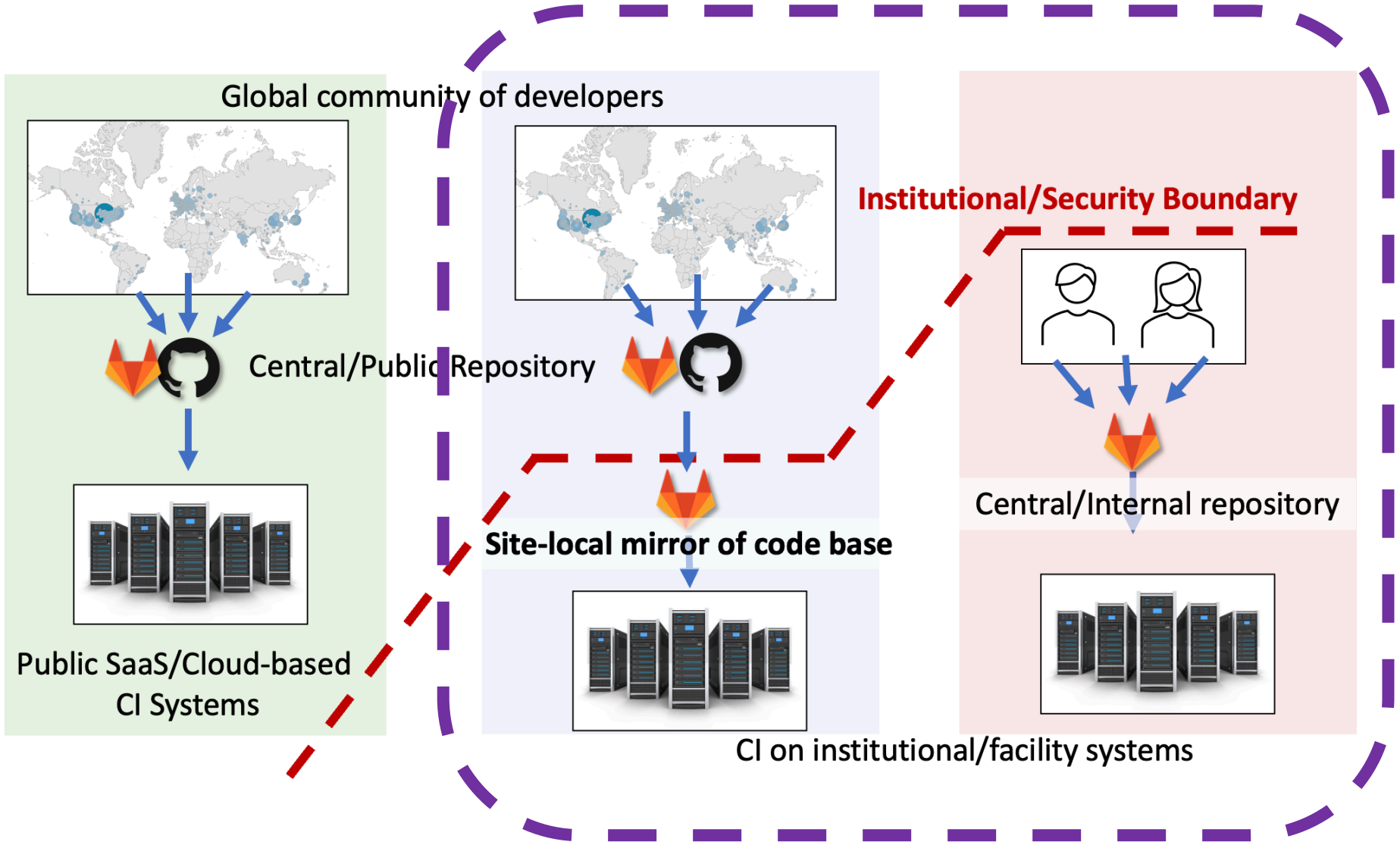
Scenario ‘B’

- Infrastructure ownership is split between code owners and targeted facility
- Include standard practices in your upstream code base
- Inside facility is an adapted testing and integration infrastructure (based on security requirements surrounding their system)

Scenario ‘C’

- Institution owns and provides everything to internal developers and code bases
- CI workflows/triggers are limited to internal developers

Review where does Jacamar CI come in?



- Extends the security model and user controls of a facility managed GitLab server instance throughout the CI/CD pipeline
- Ideal for multi-tenant HPC systems
- Other options might exist in the future, but Jacamar may be sufficient in some cases

Why is Post ECP important…

Post ECP – Carrying forward a legacy

- We need to keep the momentum and attention on making software development for these systems easier
- ECP is exemplary in how it brought the facilities together, bringing new ideas, developments, and achievements (the Integrated Research Infrastructure initiative may be next)
- The emphasis on Research Software Engineering and DevOps, within the scientific software development processes, is particularly important and should feed sustainability initiatives of the future

What we are talking about fits within the ‘IRI Practice Areas’

IRI Practice Areas (6)

User experience practice will ensure relentless attention to user perspectives and needs through requirements gathering, user-centric (co)-design, continuous feedback, and other means.

Resource co-operations practice is focused on creating new modes of cooperation, collaboration, co-scheduling, and joint planning across facilities and DOE programs.

Cybersecurity and federated access practice is focused on creating novel solutions that enable seamless scientific collaboration within a secure and trusted IRI ecosystem.

Workflows, interfaces, and automation practice is focused on creating novel solutions that facilitate the dynamic assembly of components across facilities into end-to-end IRI pipelines.

Scientific data life cycle practice is focused on ensuring that users can manage their data and metadata across facilities from inception to curation, archiving, dissemination, and publication.

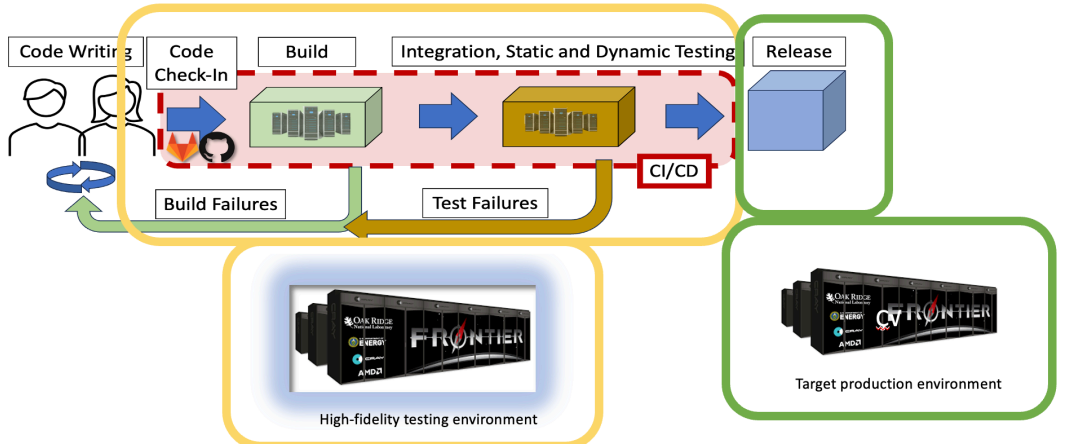
Portable/scalable solutions practice is focused on ensuring that transitions can be made across heterogeneous facilities (portability) and from smaller to larger resources (scalability).

Prioritizing and enabling DevOps and Continuous Integration services at facilities would help create:

- “Novel solutions that enable seamless scientific collaboration within a secure and trusted IRI ecosystem:
- “New modes of cooperation, collaboration, co-scheduling, and joint planning across facilities and DOE programs”

In addition, it would ensure:

- ”That users can manage their data and metadate across facilities from inception to curation, archiving, dissimination, and publication
- “That transitions can be made across heterogeneous facilities and from smaller to larger resources”



Summary

- ASCR Facilities provide unique and powerful systems that will require robust integration mechanisms for software development and deployment
- ECP is a wonderful example showcasing the ability to bring the ASCR facilities together in support of scientific software developments for the future
- The IRI and software sustainability initiatives are on the horizon and robust CI capabilities will be important for both
- Full Continuous Integration services for these unique systems and environments at ASCR facilities is still not solved and requires prioritization at the facility level
- Tools and ideas presented here are not only for ASCR systems. They are applicable to any institution wanting to enable CI services for their users

References/Resources

- Exascale Computing Project Continuous Integration Documentation: https://ecp-ci.gitlab.io/
- Better Scientific Software: https://bssw.io/
- IRI Presentation from Ben Brown (ASCR Facilities Director): https://science.osti.gov/-/media/ascr/ascac/pdf/meetings/202306/Brown_IRI_ASCAC_2023206.pdf
- Gitlab and Github docs:
  - https://docs.gitlab.com/ee/ci/introduction/
  - https://resources.github.com/ci-cd/
- Research Software Engineering: https://arxiv.org/pdf/2002.01035.pdf
- ResearchOps paper: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7140503
- Software Sustainability Seedling Projects: https://www.hpcwire.com/off-the-wire/doe-ascr-selects-seed-collaborations-for-software-sustainability/