CI for ASAP Applications: Using Github Actions for Rapid Development

Haniye Kashgarani

University of Wyoming

Mentors: Supreeth Suresh, Cena Brown

August 1st, 2023

Motivation

What is CI:

- Continuous Integration (CI) refers to the process of automatically building and testing code every time a team member commits changes to version control system.

Life without CI:

- Integration Hell
- Late Bug Detection
- Slow Feedback Loop
- Unclear Understanding of Project State
- Slow Delivery

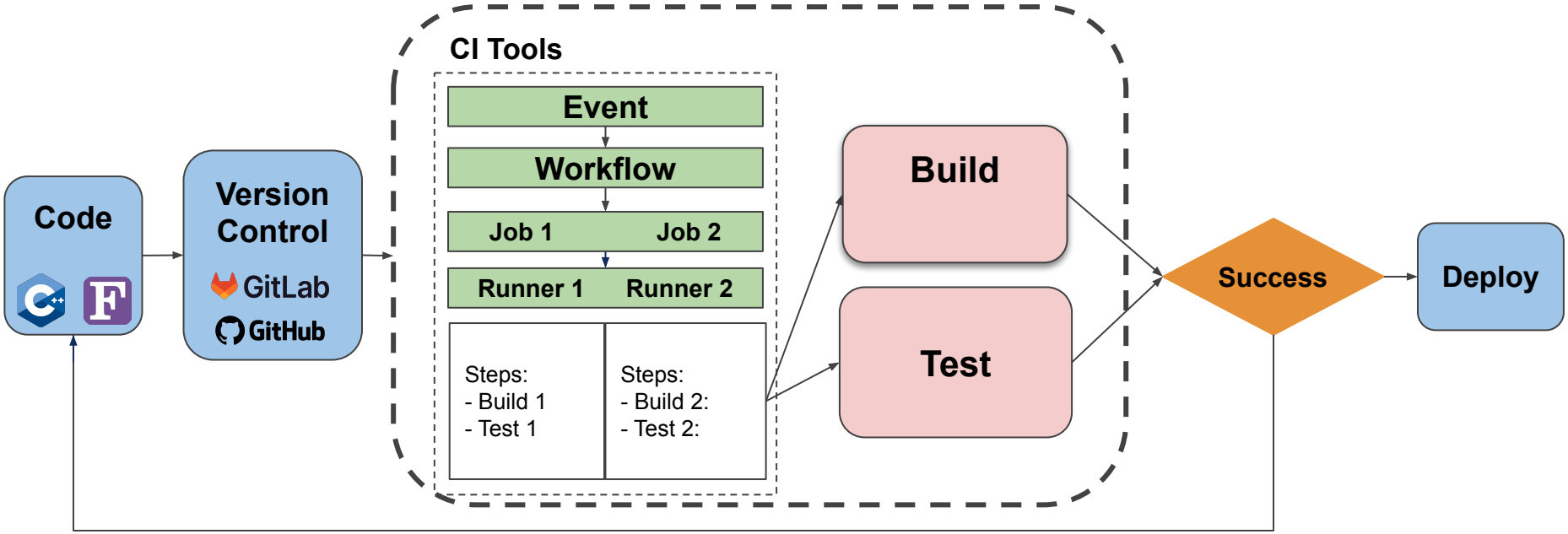
CI Benefits for Parallel Applications

Application Scalability And Performance Group: Leverage the parallel power of GPUs and CPUs with MPI communication for higher performance.

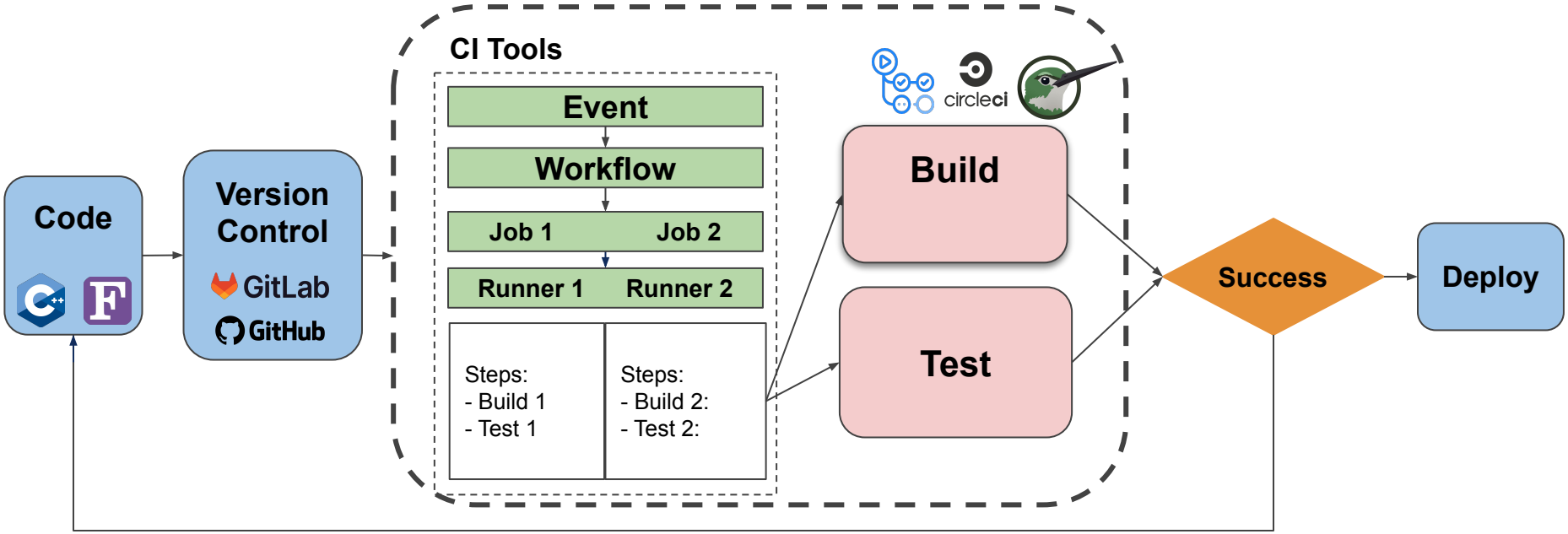
CI for CUDA and MPI applications:

- Dependency management
- Library versions: Dealing with CUDA and OpenMPI versions
- GPU compatibility
- Frequent automated testing
- Early detection of bottlenecks
- Speedy Development

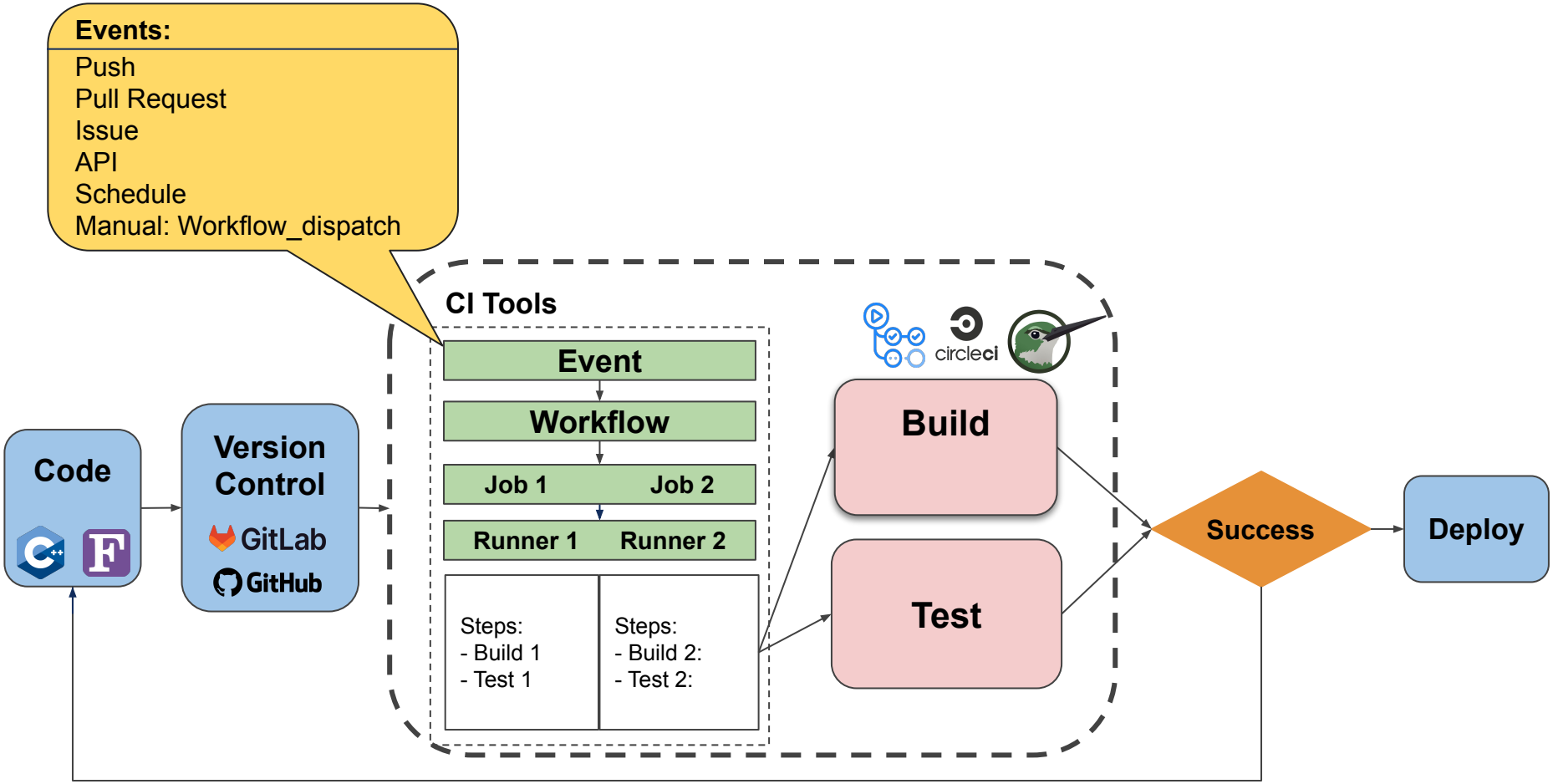
CI Pipeline


CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
Code
Version Control
GitLab
GitHub
Build
Test
Success
Deploy

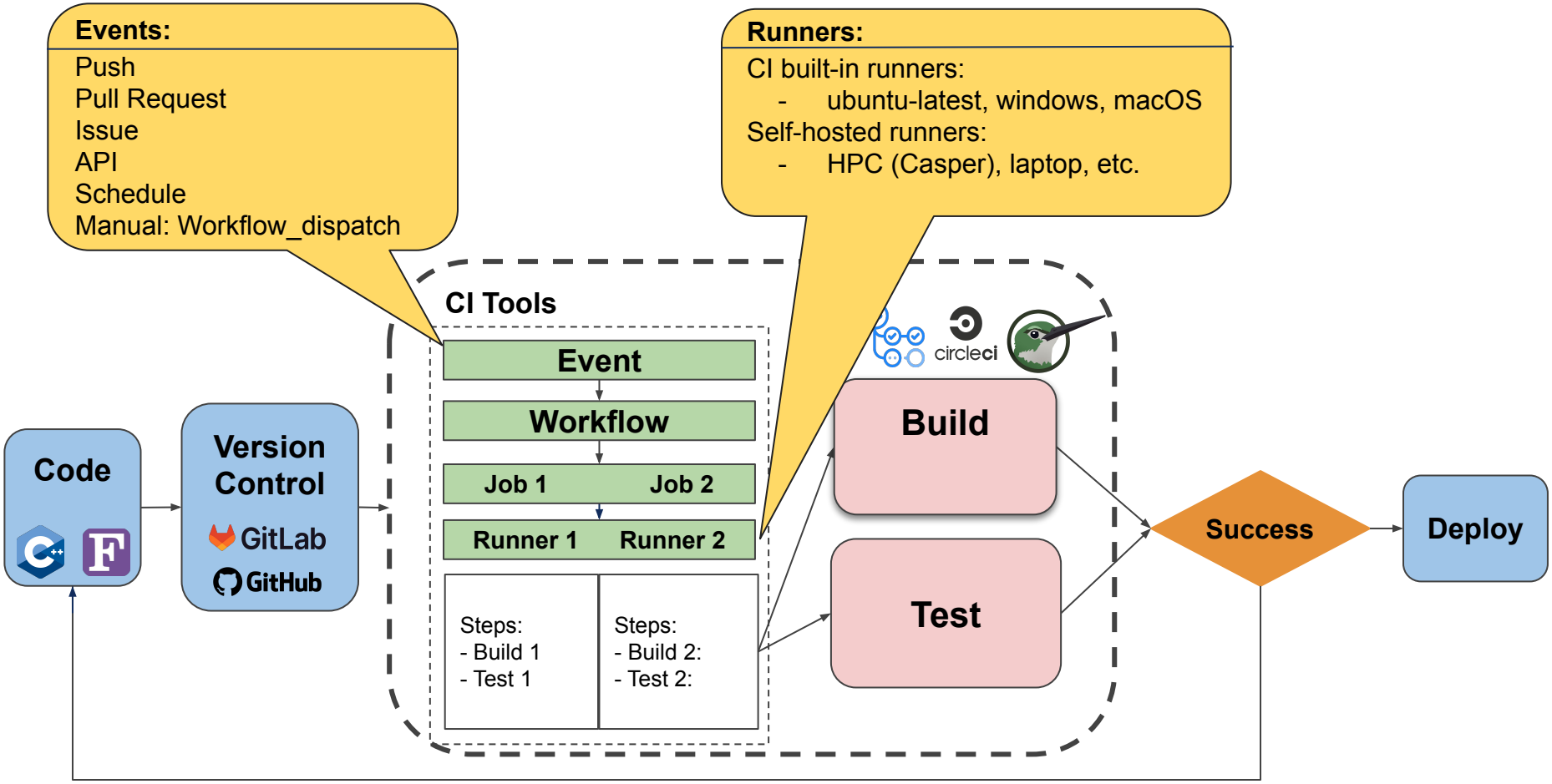
CI Pipeline


CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Code
Version Control
GitLab
GitHub
Build
Test
Success
Deploy

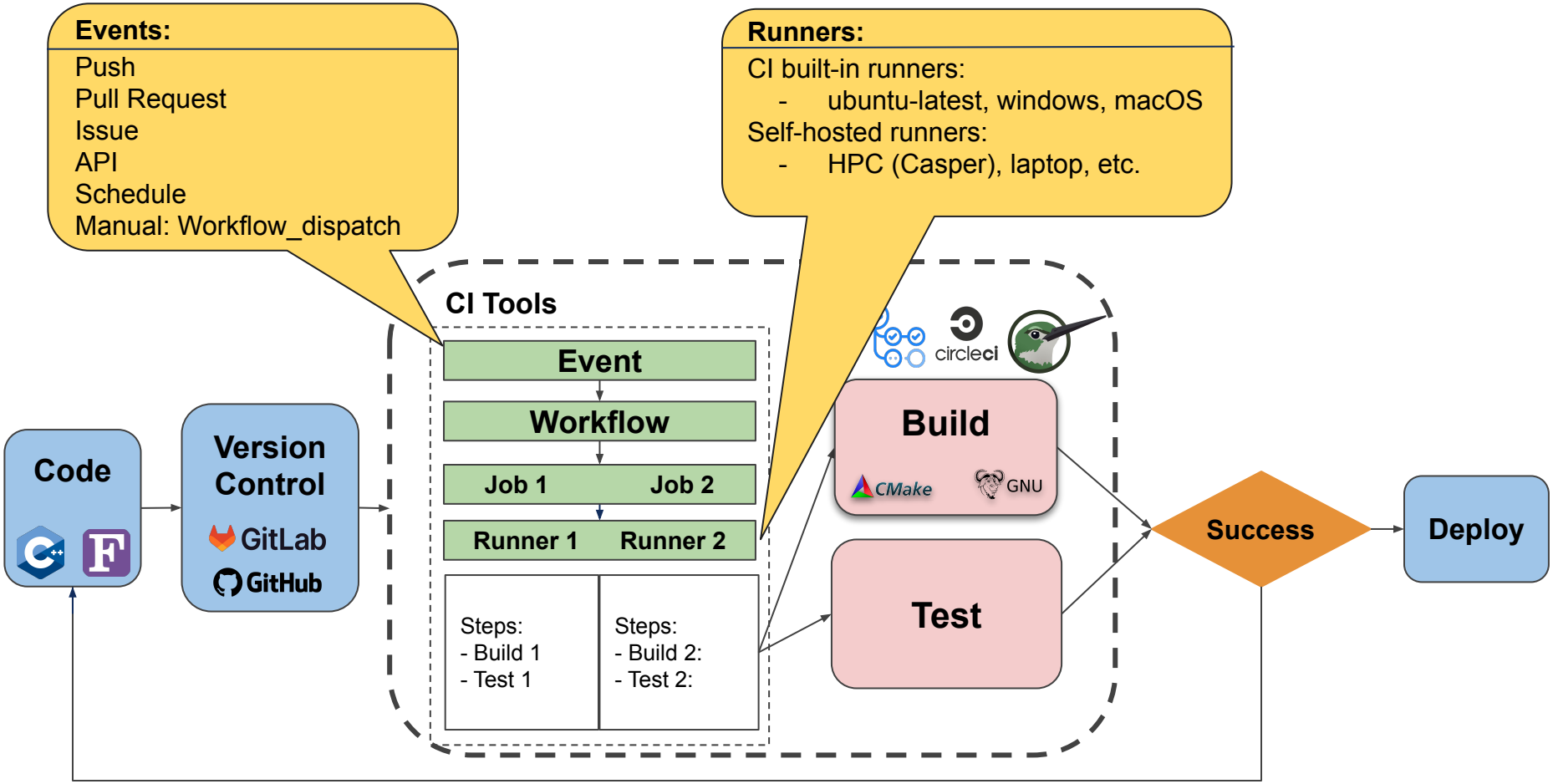
CI Pipeline


Events:
Push
Pull Request
Issue
API
Schedule
Manual: Workflow_dispatch
CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Code
Version Control
GitLab
GitHub
Build
Test
Success
Deploy

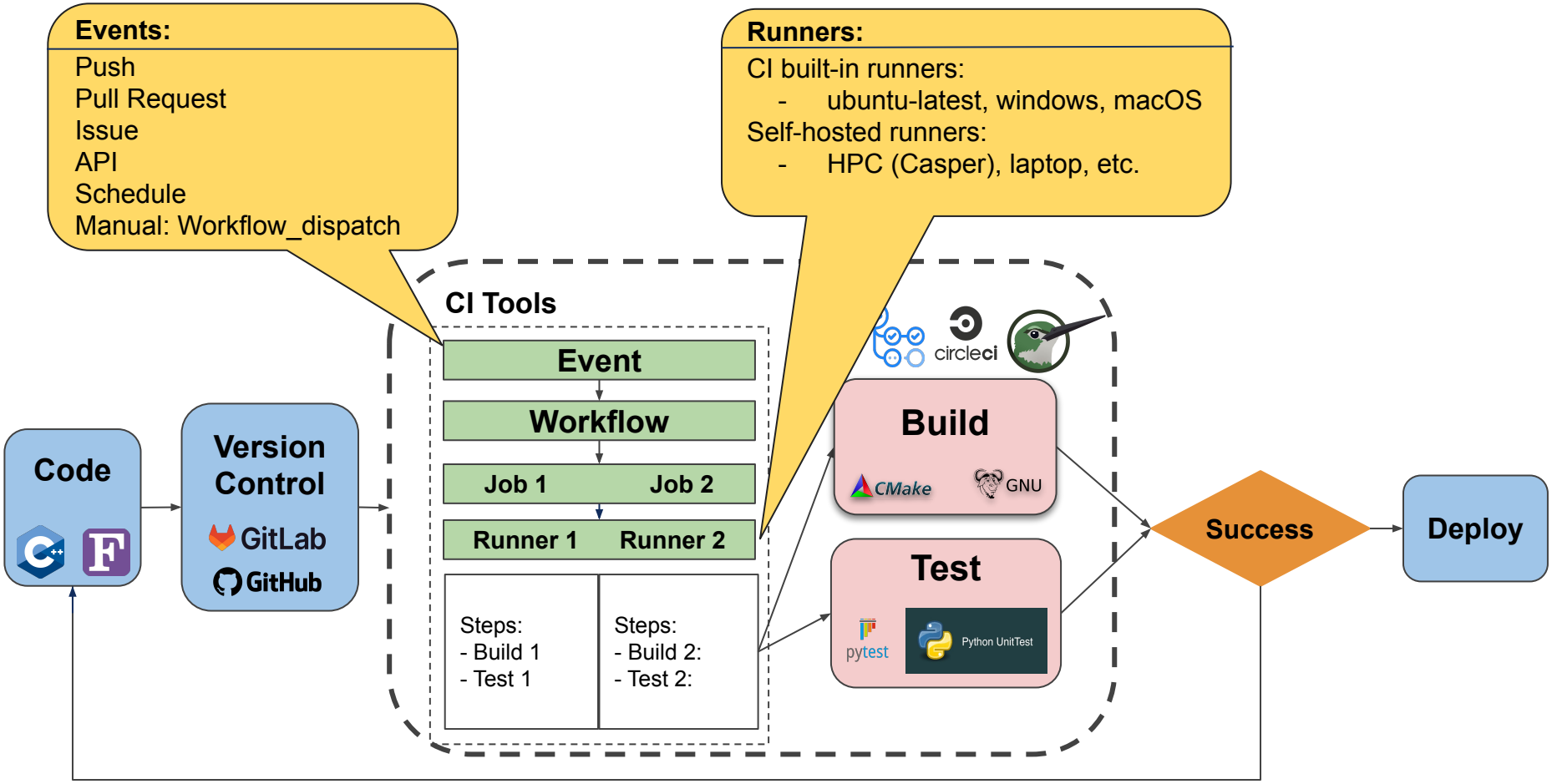
CI Pipeline


Events:
Push
Pull Request
Issue
API
Schedule
Manual: Workflow_dispatch
Runners:
CI built-in runners:
- ubuntu-latest, windows, macOS
Self-hosted runners:
- HPC (Casper), laptop, etc.
CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Code
Version Control
GitLab
GitHub
Build
Test
Success
Deploy

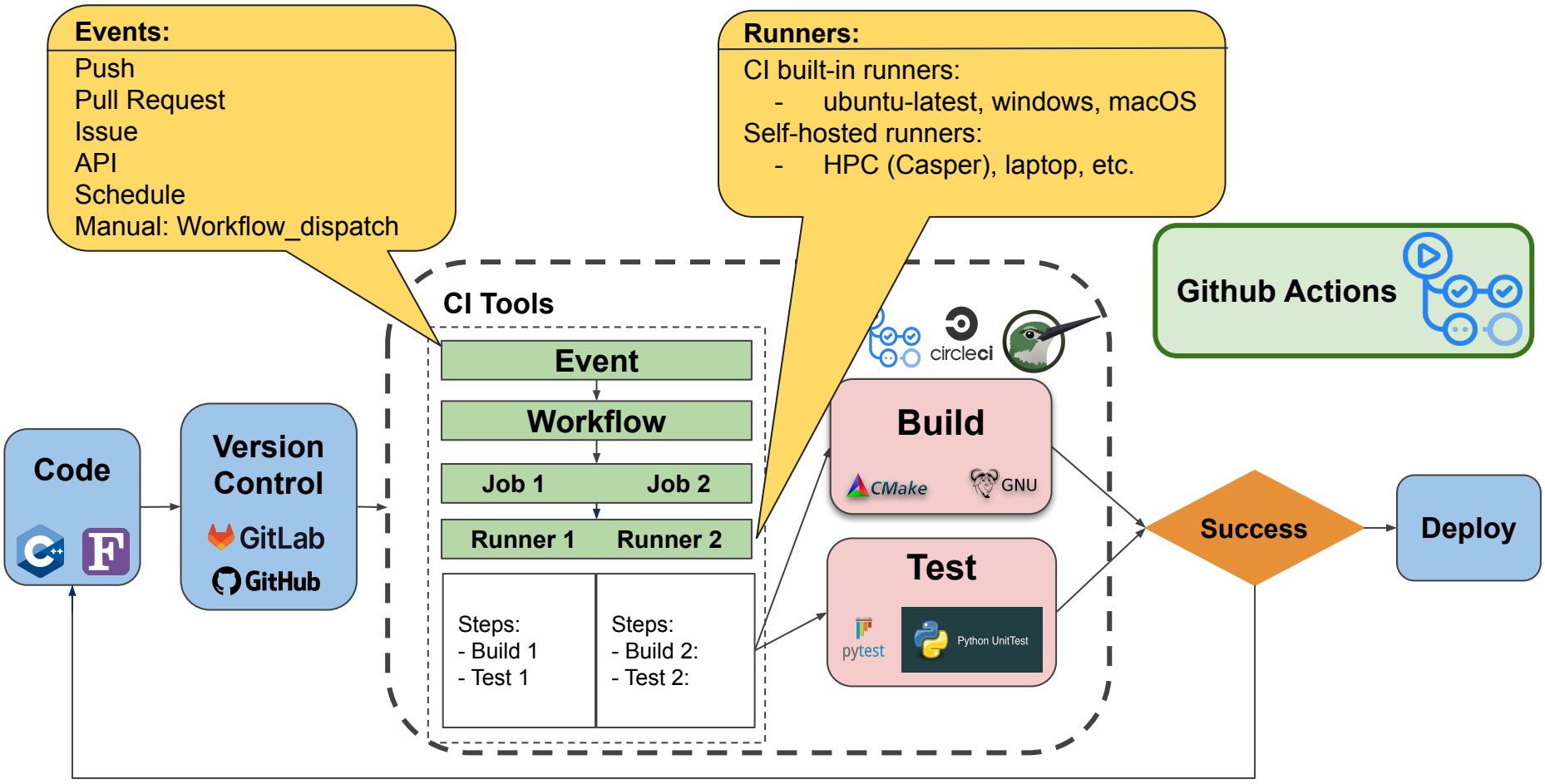
CI Pipeline


Events:
Push
Pull Request
Issue
API
Schedule
Manual: Workflow_dispatch
Runners:
CI built-in runners:
- ubuntu-latest, windows, macOS
Self-hosted runners:
- HPC (Casper), laptop, etc.
CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Build
CMake
GNU
Test
Code
Version Control
GitLab
GitHub
Success
Deploy

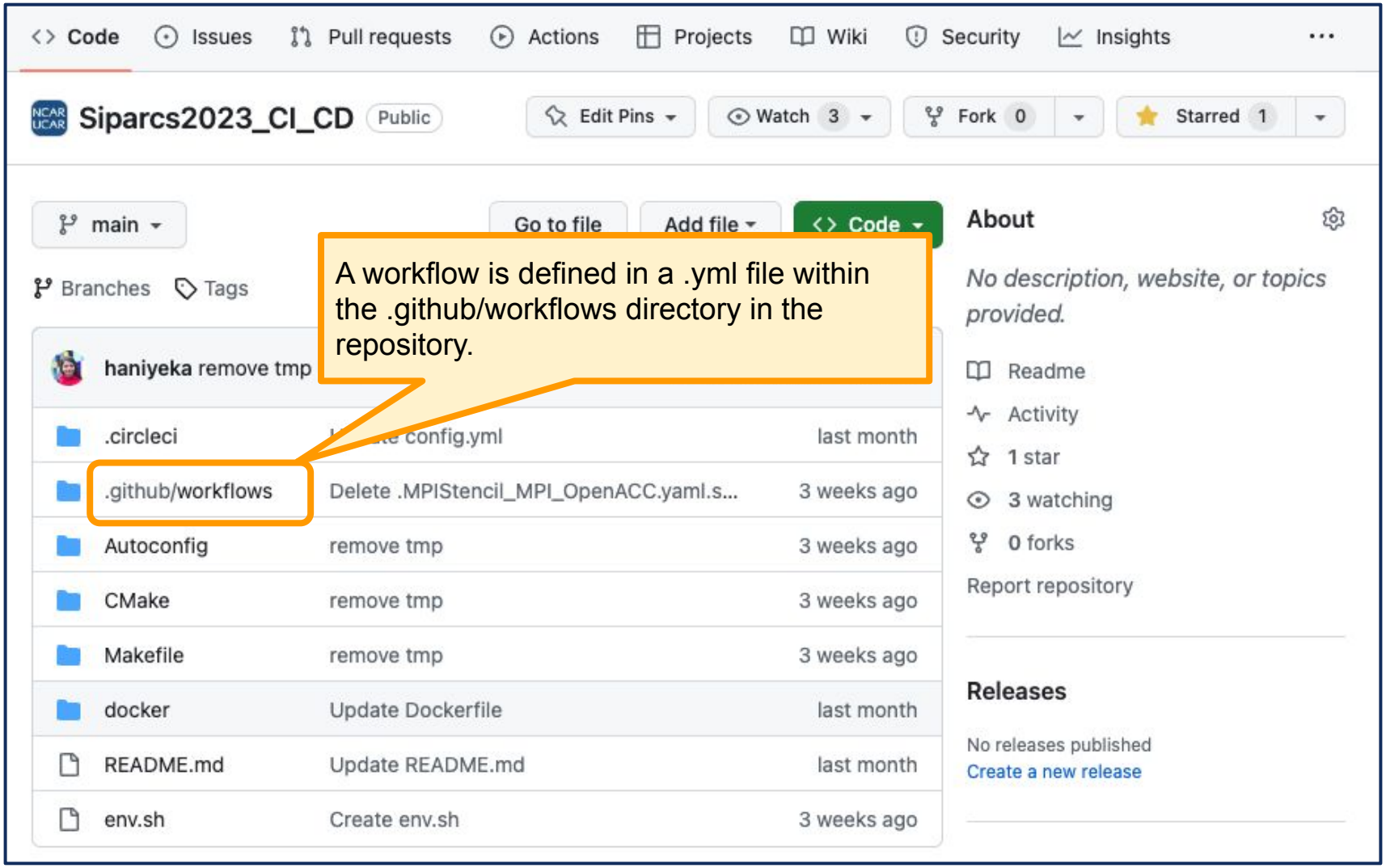
CI Pipeline


Events:
Push
Pull Request
Issue
API
Schedule
Manual: Workflow_dispatch
Runners:
CI built-in runners:
- ubuntu-latest, windows, macOS
Self-hosted runners:
- HPC (Casper), laptop, etc.
CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Build
CMake
GNU
Test
pytest
Python UnitTest
Code
Version Control
GitLab
GitHub
Success
Deploy

CI Pipeline


Events:
Push
Pull Request
Issue
API
Schedule
Manual: Workflow_dispatch
Runners:
CI built-in runners:
- ubuntu-latest, windows, macOS
Self-hosted runners:
- HPC (Casper), laptop, etc.
Github Actions
CI Tools
Event
Workflow
Job 1
Job 2
Runner 1
Runner 2
Steps:
- Build 1
- Test 1
Steps:
- Build 2:
- Test 2:
circleci
Build
CMake
GNU
Test
pytest
Python UnitTest
Code
Version Control
GitLab
GitHub
Success
Deploy

Github Actions



Github Actions

Events

Workflow files can define one or more jobs

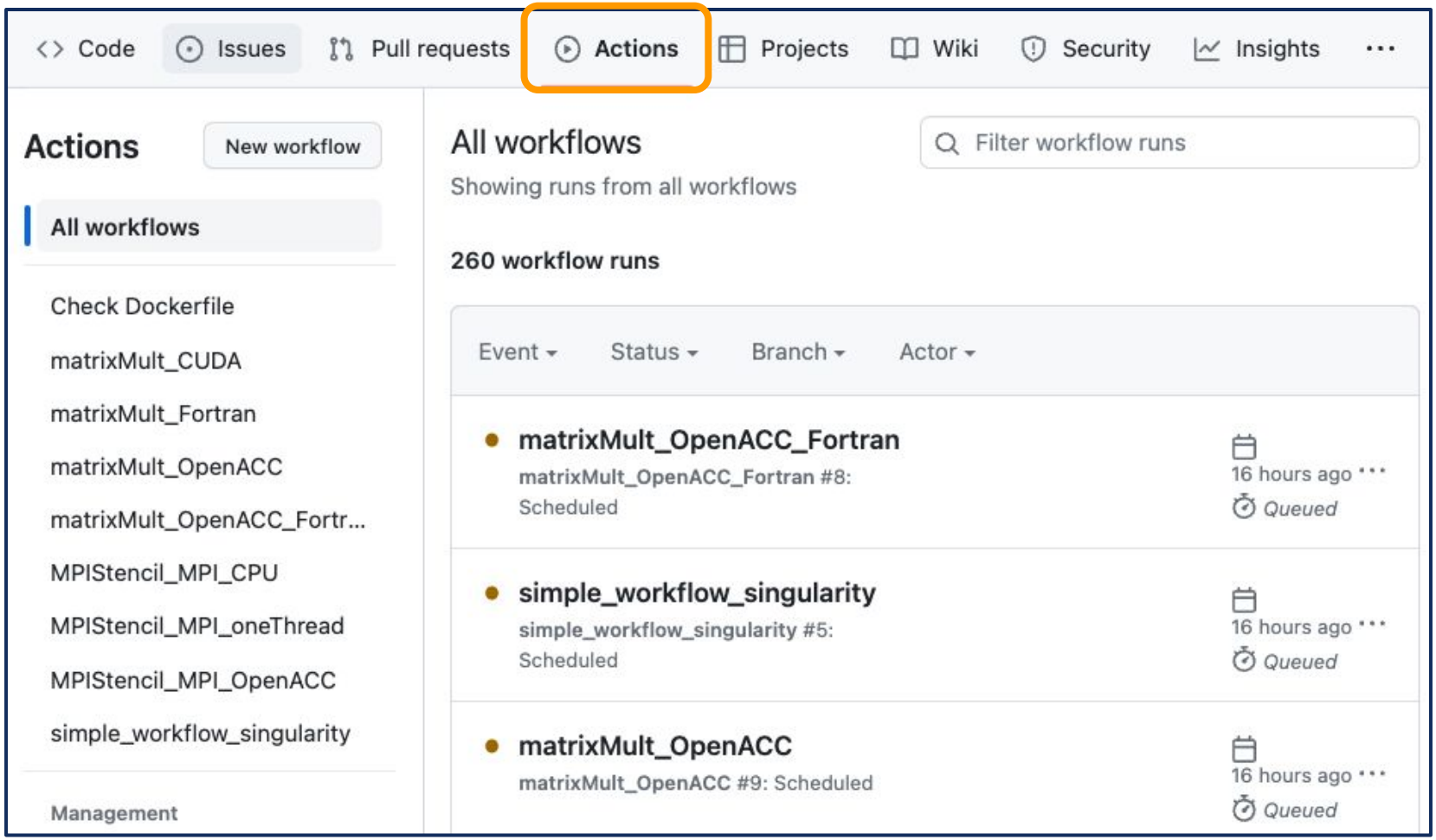
Runner Label

```
container:
  image: node:14.16
```

Each job consists of a set of steps that perform individual tasks such as running a script or a command.

```
1   name: matrixMult_Fortran_Autoconfig
2
3   on:
4     push:
5       branches:
6         - main
7     pull_request:
8       branches:
9         - main
10    workflow_dispatch:
11
12  jobs:
13    Makefile:
14      runs-on: ubuntu-latest
15      steps:
16      - uses: actions/checkout@v3
17
18      - name: Navigate to directory and autoconfig
19        run: |
20          cd Autoconfig/matrixMult/FORTRAN/
21          pwd
22          autoconf
23          ./configure
24          make
```

Github Actions



Self-hosted Runners

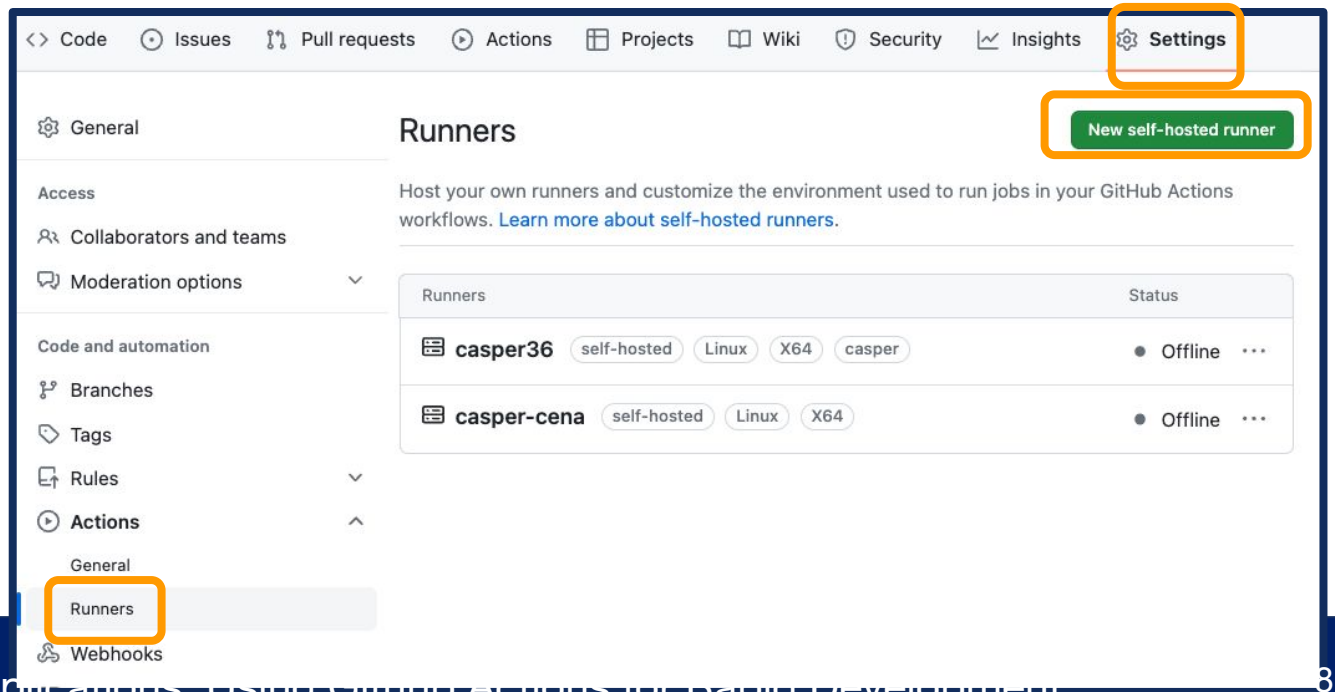
Built-in runners: No GPU software and hardware support, no MPI communication support

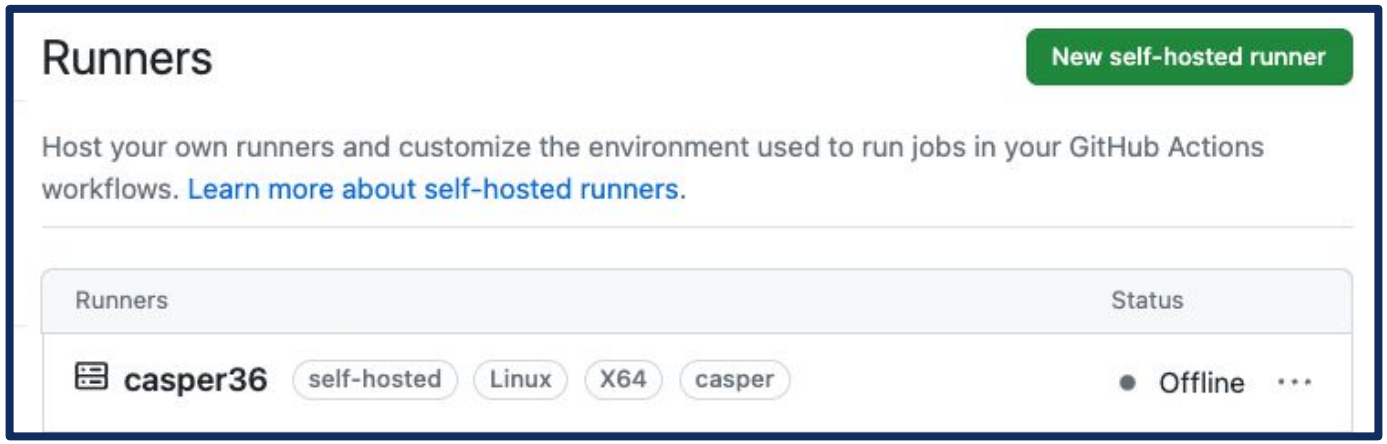
Self-hosted runners:

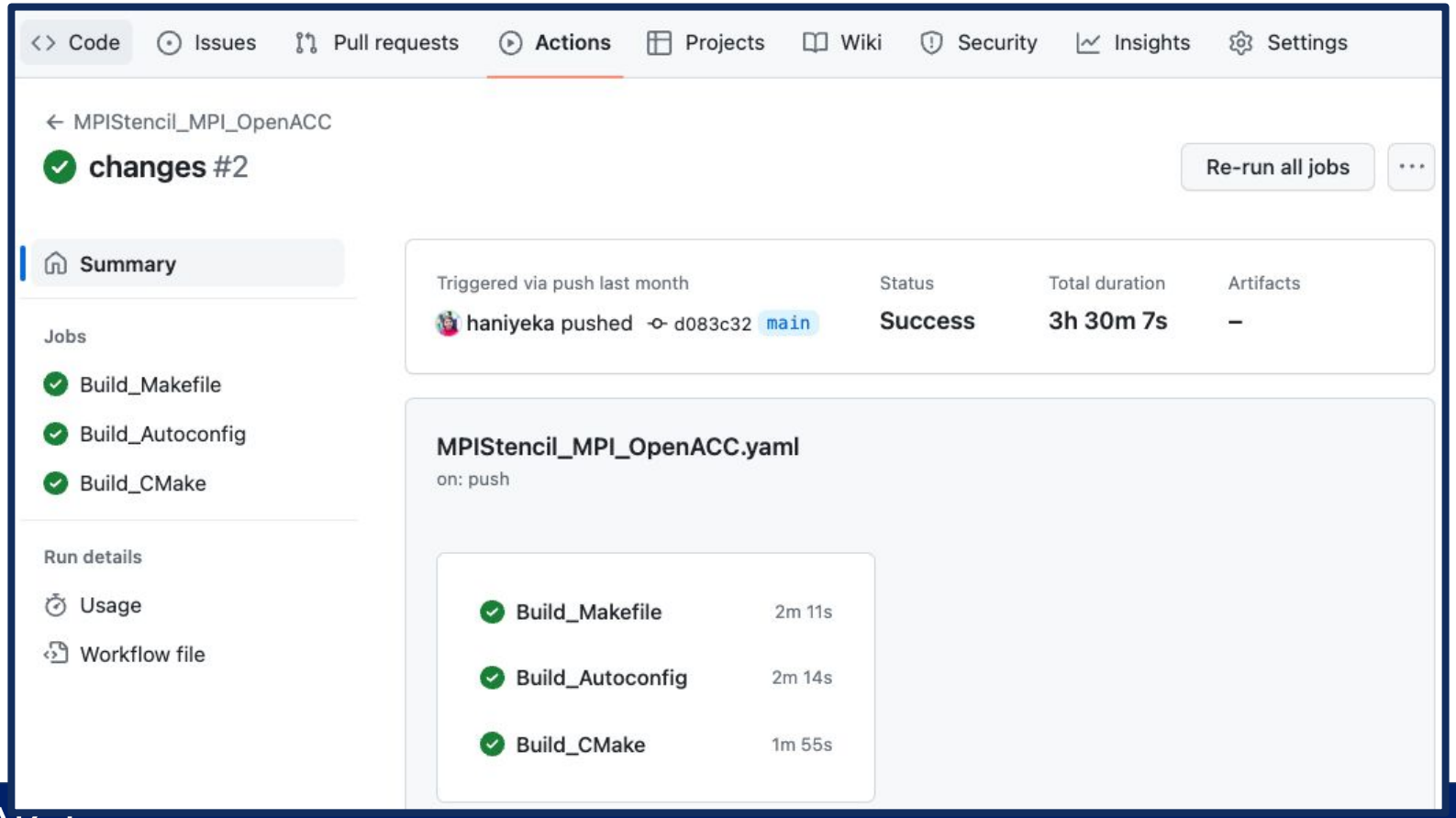
- Login to HPC: `ssh username@casper.ucar.edu`
- Follow step-by-step instructions provided in the repository's settings
- Prepare your environment and Run the provided **run.sh** by github

```
execcasper -A NTDD0005 -l walltime=05:00:00 -l
select=1:ncpus=1:mpiprocs=1:ngpus=1:mem=50GB -l gpu_type=v100
```

- Add labels to yml file: `casper, self-hosted, Linux, X64`
- Trigger an event
  - **Push**
  - **Pull Request**



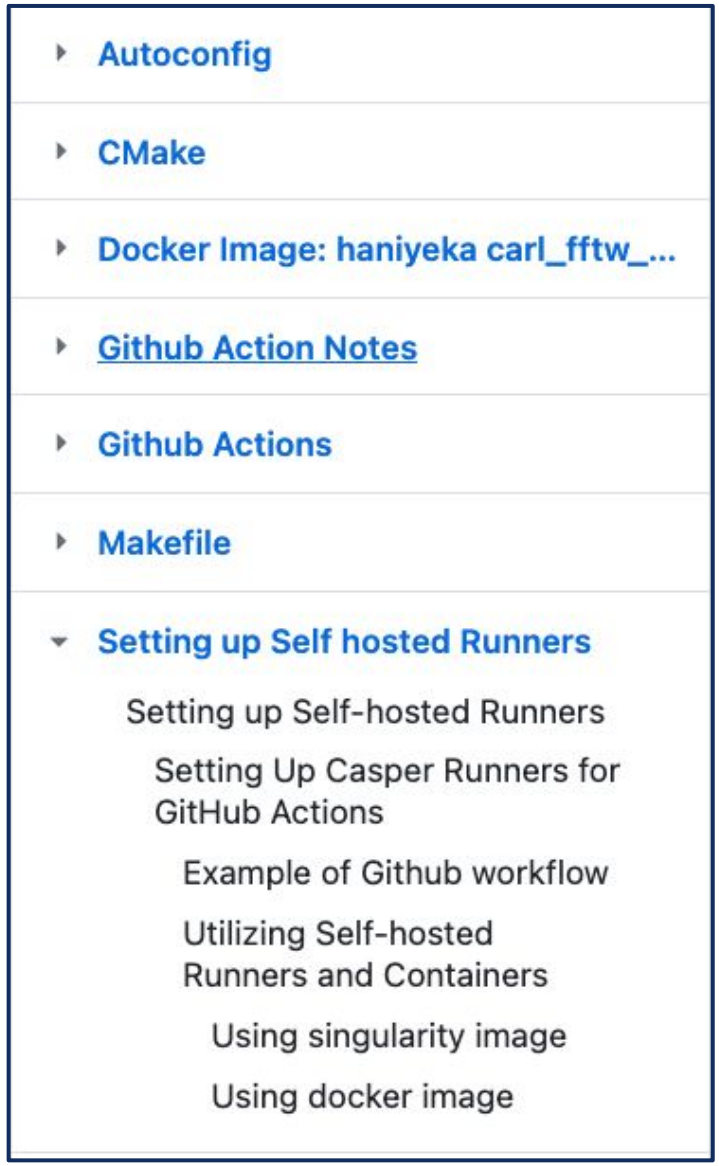

Runners
New self-hosted runner
Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Learn more about self-hosted runners.
Runners
Status
casper36
self-hosted
Linux
X64
casper
Offline


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
MPIStencil_MPI_OpenACC
changes #2
Re-run all jobs
Summary
Jobs
Build_Makefile
Build_Autoconfig
Build_CMake
Run details
Usage
Workflow file
Triggered via push last month
haniyeka pushed d083c32 main
Status
Success
Total duration
3h 30m 7s
Artifacts
–
MPIStencil_MPI_OpenACC.yaml
on: push
Build_Makefile 2m 11s
Build_Autoconfig 2m 14s
Build_CMake 1m 55s

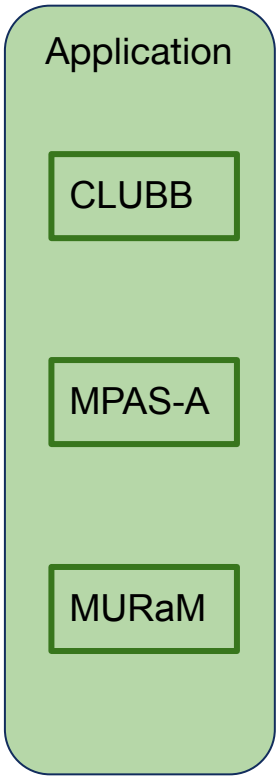
Documentation and Examples

https://github.com/NCAR/Siparcs2023_CI_CD/

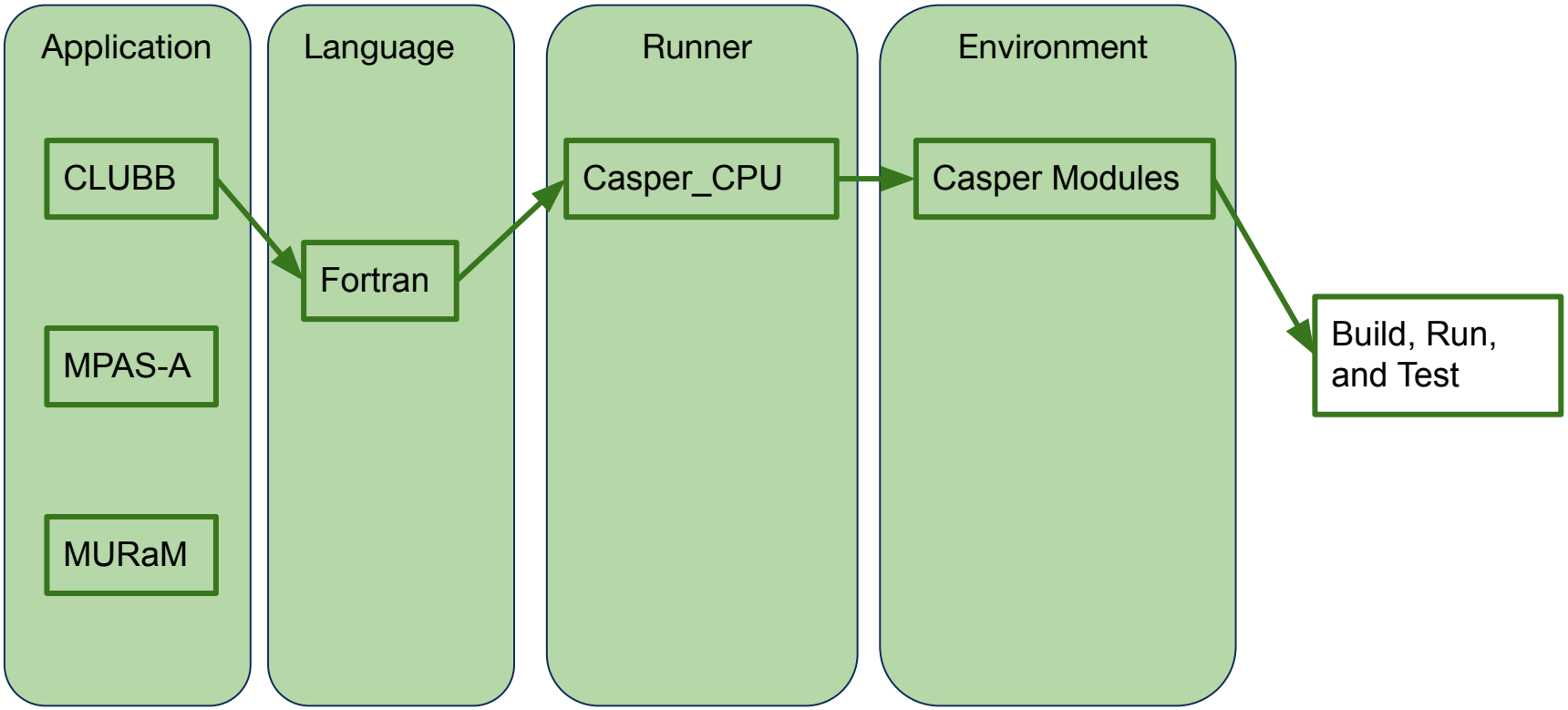
- DockerCheck.yaml
- MPIStencil_MPI_CPU.yaml
- MPIStencil_MPI_OpenACC.yaml
- MPIStencil_MPI_oneThread.yaml
- matrixMult_CUDA.yaml
- matrixMult_Fortran.yaml
- matrixMult_OpenACC.yaml
- matrixMult_OpenACC_FORTRAN.yaml
- simple_workflow_docker.yaml



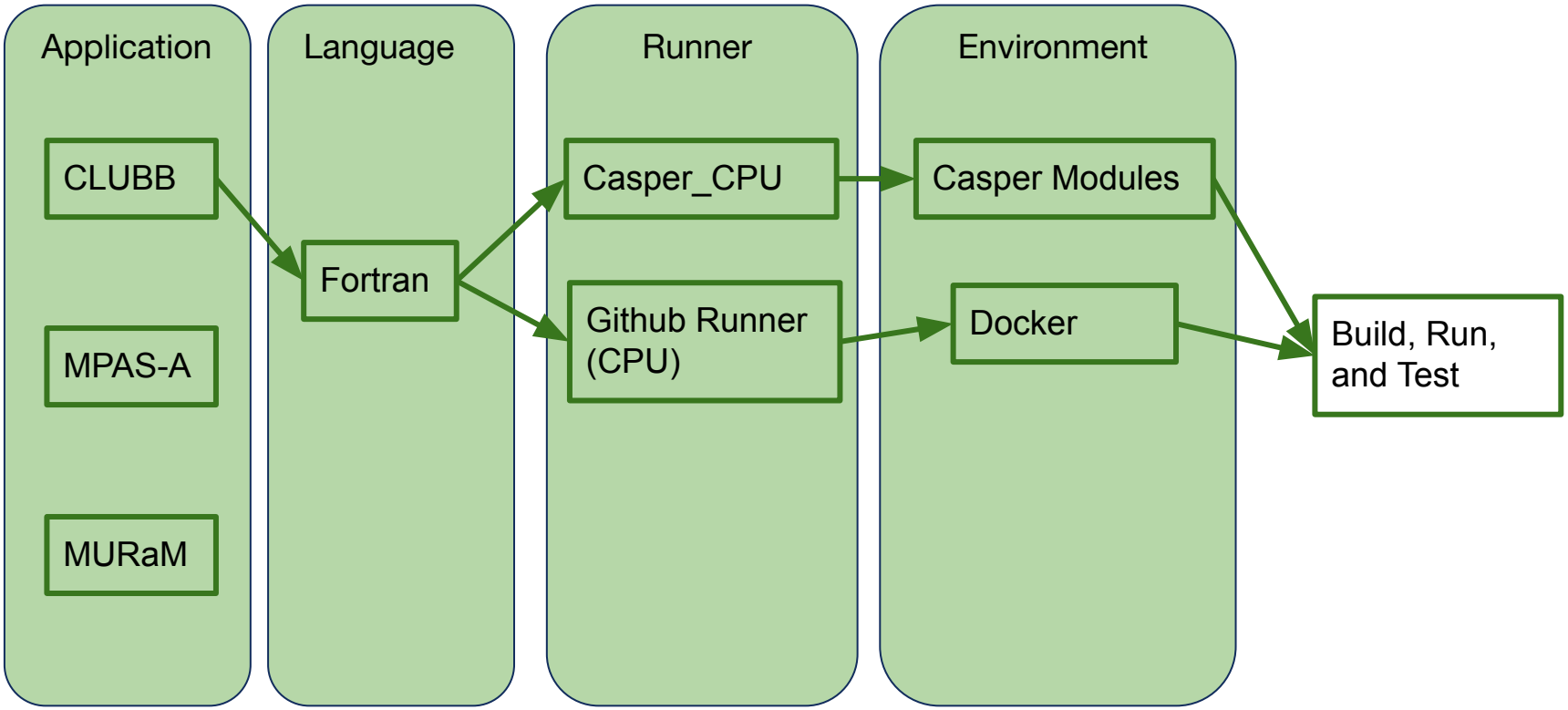
ASAP Applications



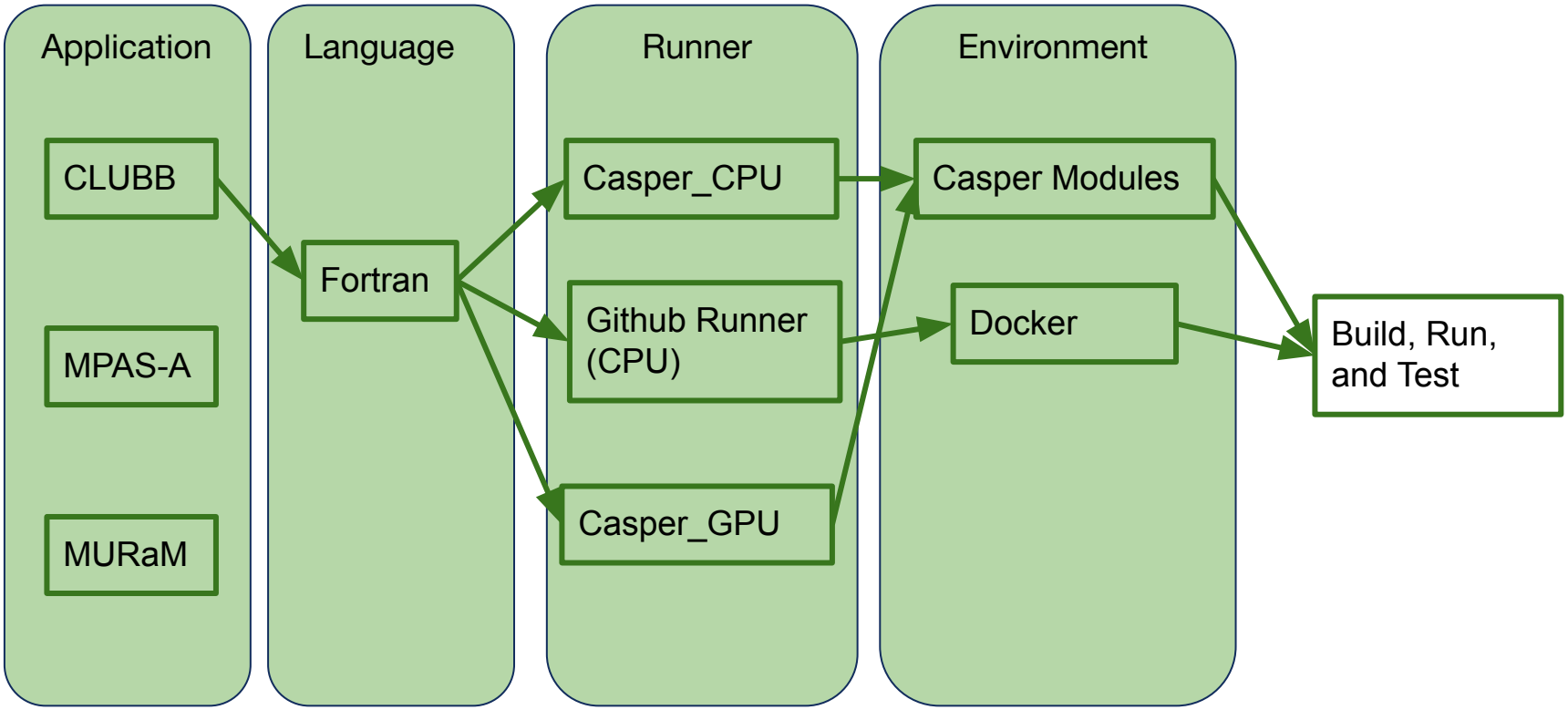
ASAP Applications



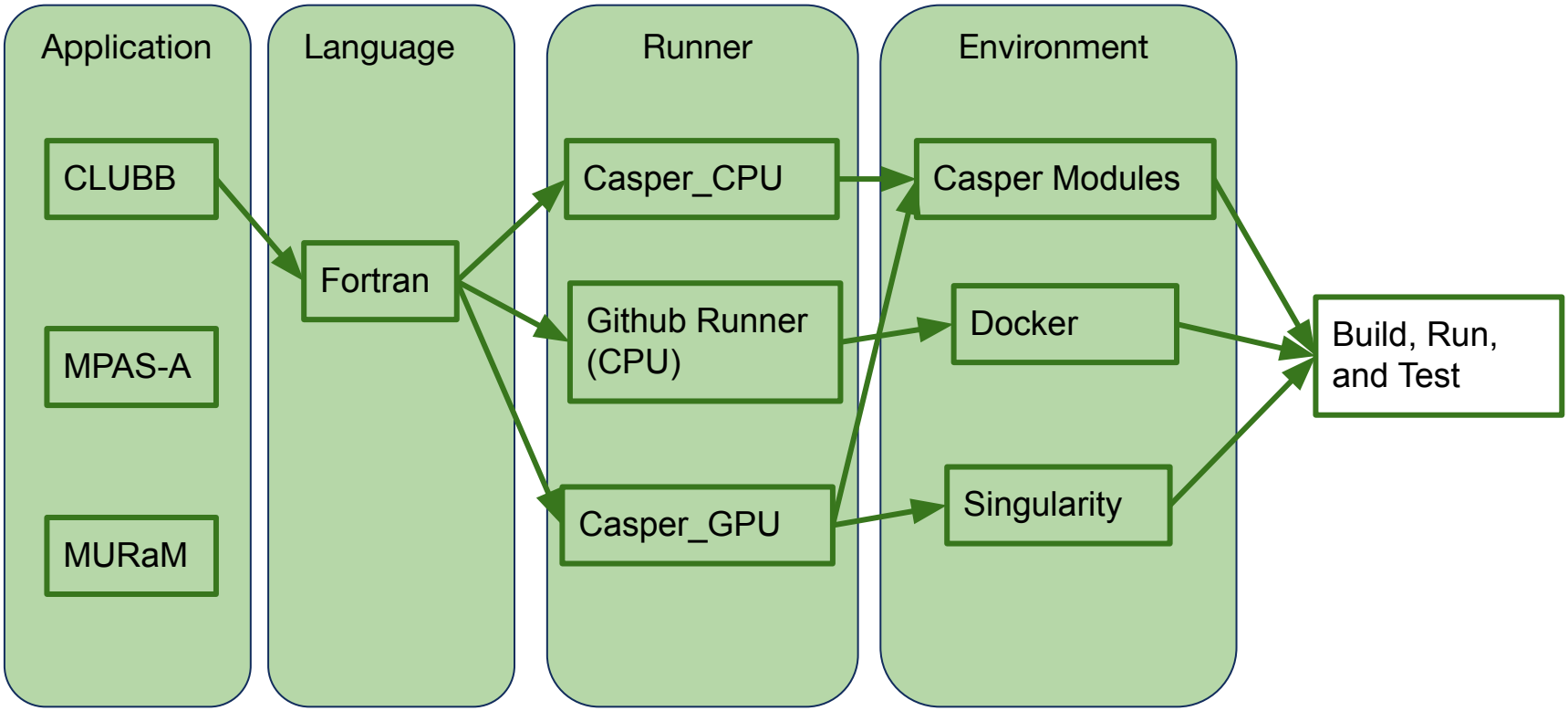
ASAP Applications



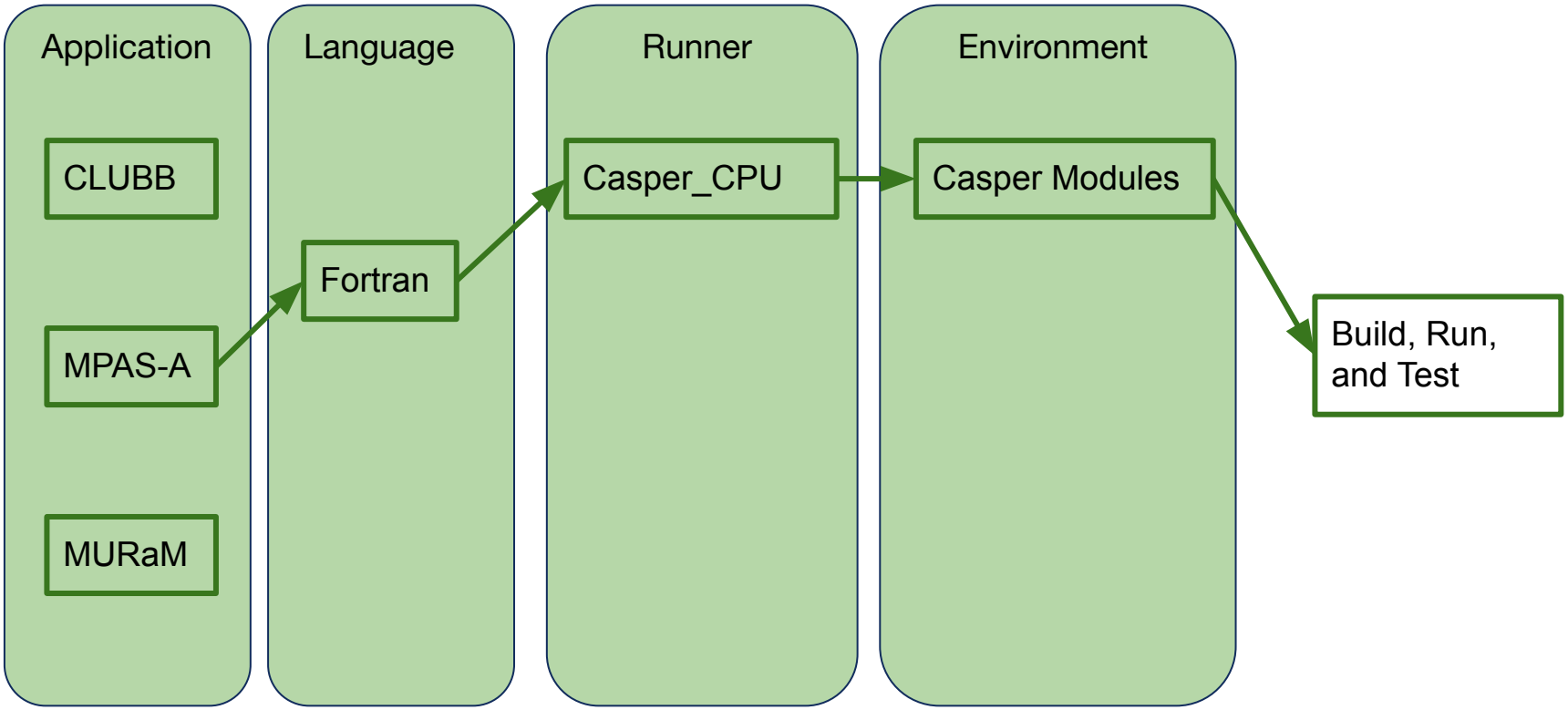
ASAP Applications



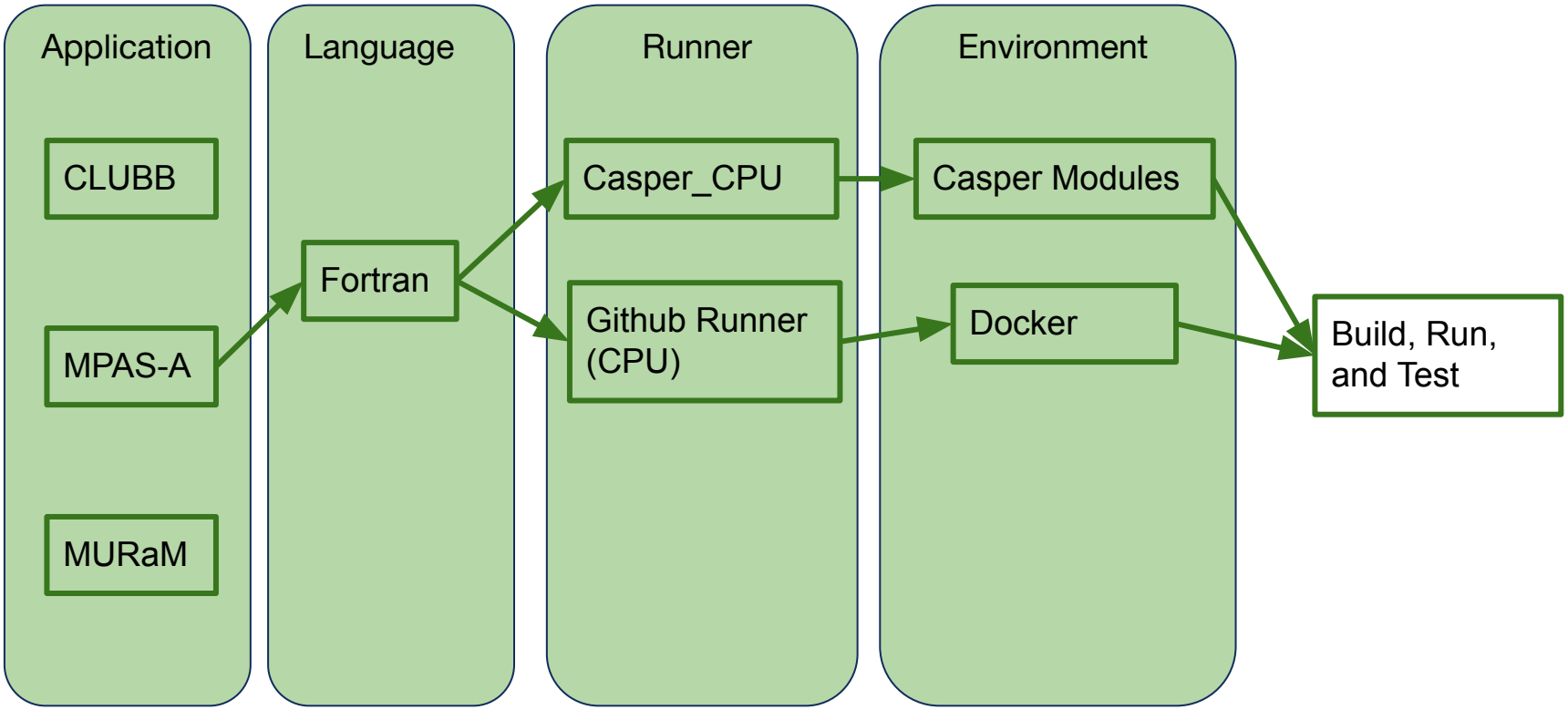
ASAP Applications



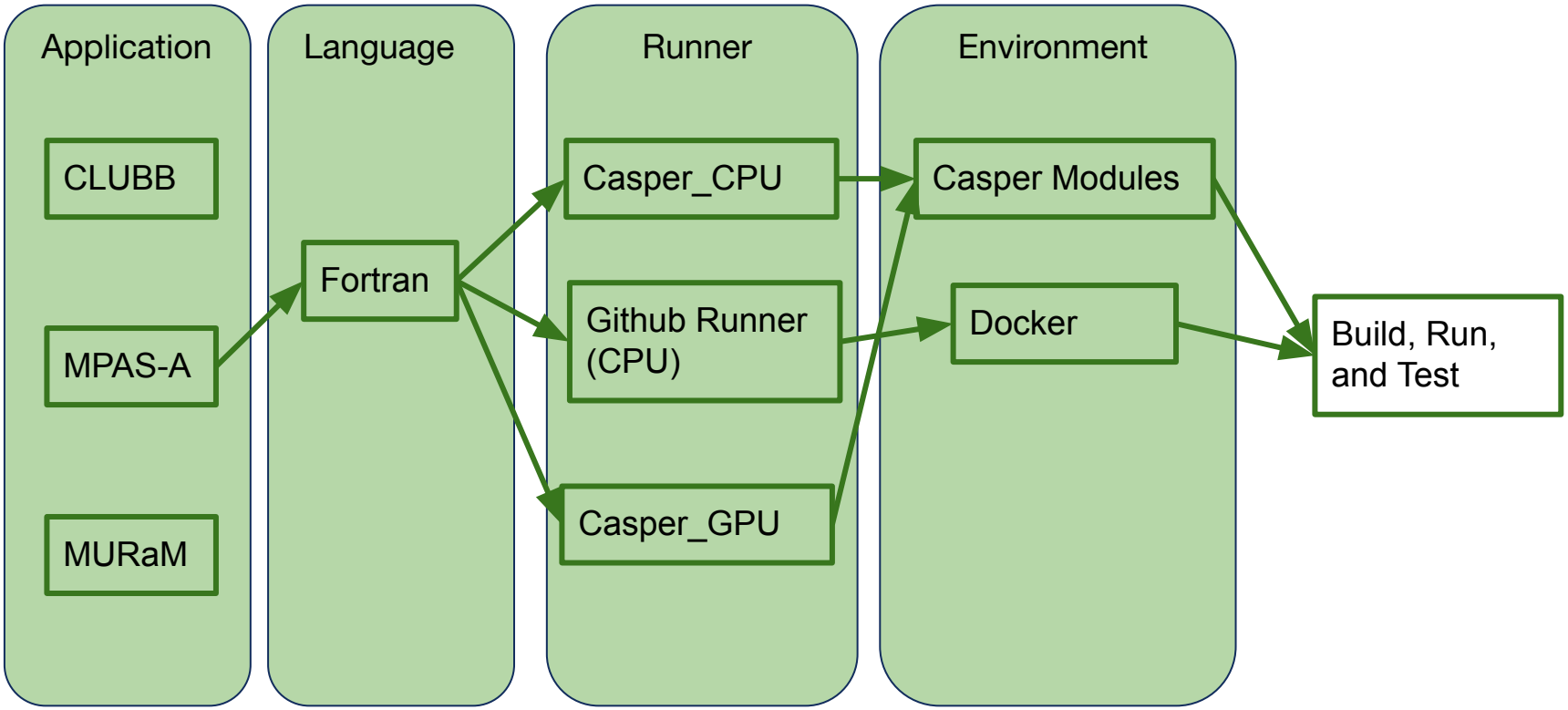
ASAP Applications



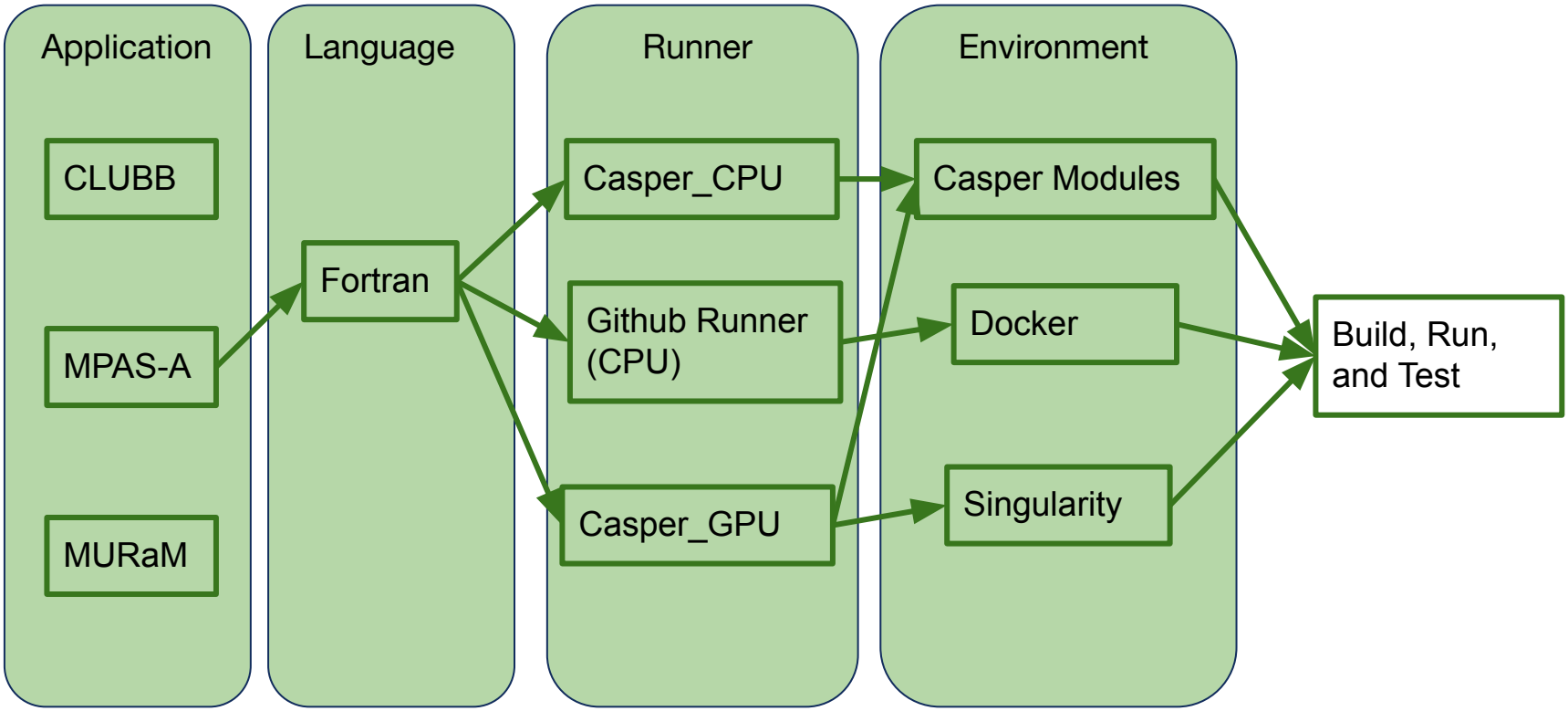
ASAP Applications



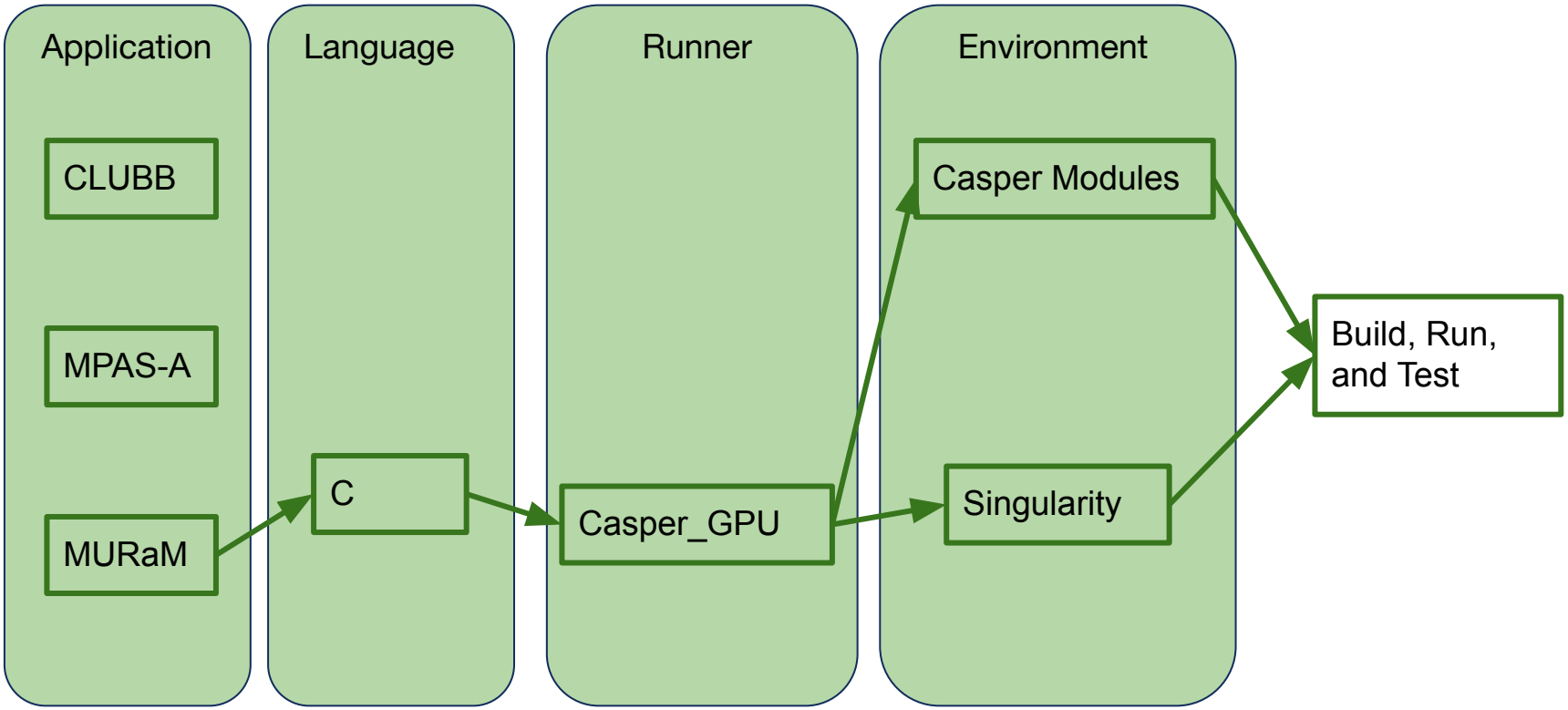
ASAP Applications



ASAP Applications



ASAP Applications



Summary of Achieved Goals

- Documentation and Examples

- Focus on implementing CI for multiple ASAP applications
  - CLUBB
  - MURaM
  - MPAS-A

Thank you!

Mentors: Supreeth Suresh, Cena Brown

Carl Ponder - NVIDIA

ASAP Team!

SIParCS Team!

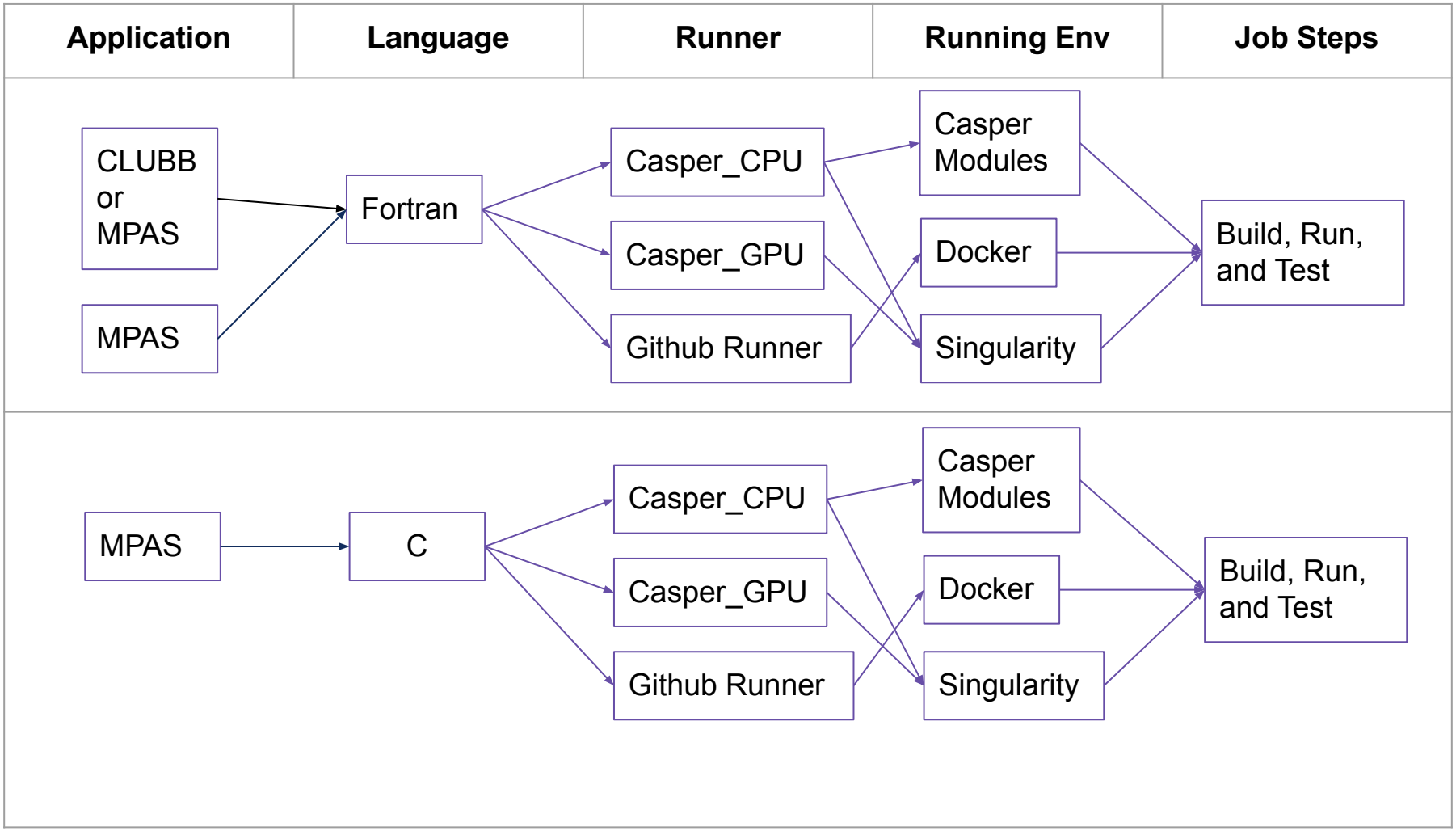
SIParCS 2023 Interns!

Questions?

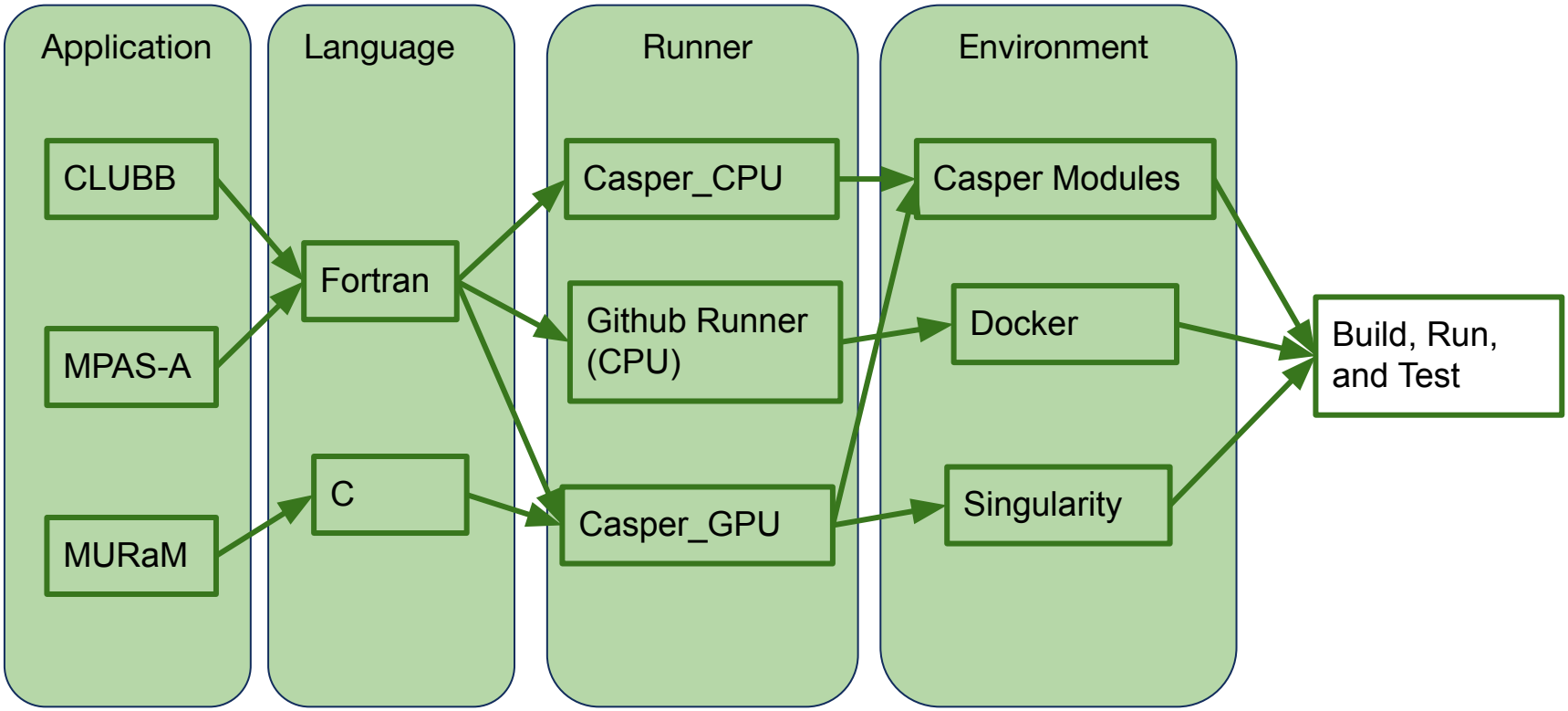
CI/CD Examples and Documentations

hkashgar@uwyo.edu

Workflows


Application
Language
Runner
Running Env
Job Steps
CLUBB or MPAS
MPAS
Fortran
Casper_CPU
Casper_GPU
Github Runner
Casper Modules
Docker
Singularity
Build, Run, and Test
MPAS
C
Casper_CPU
Casper_GPU
Github Runner
Casper Modules
Docker
Singularity
Build, Run, and Test

ASAP Applications



Application	Language	Runner	Environment
CLUBB	Fortran	Github Runner	Docker Container
		Casper CPU	Casper Modules
		Casper GPU	Singularity Container
			Casper Modules
MPAS-A	Fortran	Github Runner	Docker Container
		Casper CPU	Casper Modules
		Casper GPU	Singularity Container
			Casper Modules
MURaM	C	Casper GPU	Singularity Container
			Casper Modules