

Automated Workflows for Building and Testing Climate Models

Tommy Gatti (ACCESS-NRI)

Harshula Jayasuriya (ACCESS-NRI)

Aidan Heerdegen (ACCESS-NRI)





Starting with a Summary

- Climate model provenance and discoverability is paramount and automatable
- Spack gives us full build provenance, reproducibility and build caching of our software
- GitHub Actions tie this together in modular, generic workflows, allowing us to scale quickly
- Both model developers and researchers benefit from the use of these tools!
- This is applicable to a wide variety of types of software

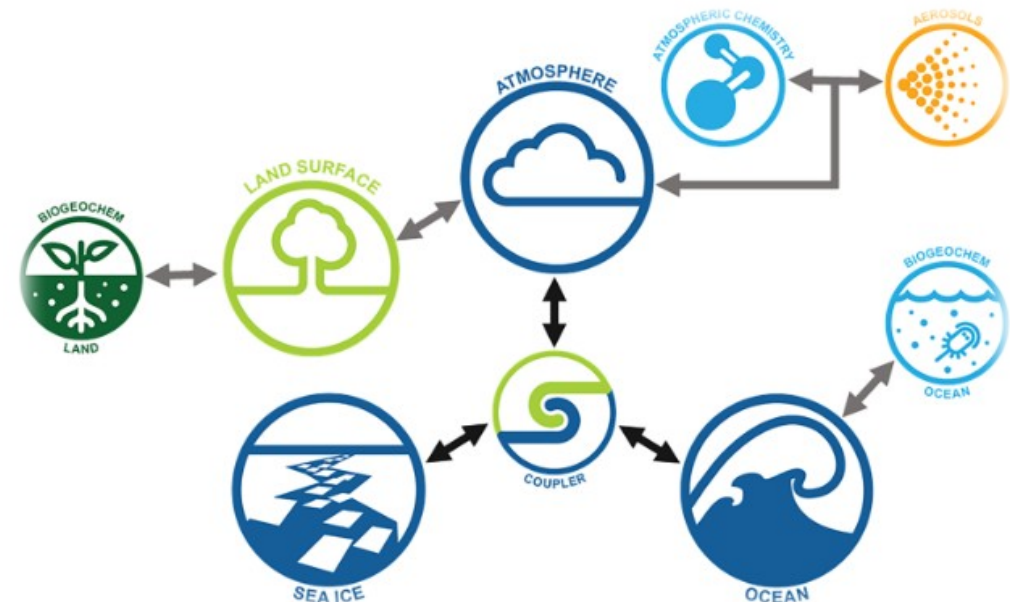
Brief Background – ACCESS-NRI

- **Australia's climate simulator (ACCESS-NRI)** is a National Research Infrastructure that supports development/research with Climate Models
- A big grab-bag of research software engineers, model developers, scientists, oh my!
- We attempt to make climate researchers lives easier
- In our team (Model Release) we look to make the model developers lives easier
- Automated building and testing, model use statistics, ease of use and discovery of models...



Brief Background – ACCESS Models

- ACCESS Models are software-based climate models linking together mathematical representations of major Earth System components
- Complex – some models total 1.5m+ lines of code, with many dependencies!
- Scale – We support 7 models currently, 10+ by the end of 2026!
- We have two main audiences for these:
 - Researchers using the models
 - Developers updating the models









The Problem of Provenance – The Bad Old Days



"Which is the *good* binary?"
- Anon., 1980(?) - 2025

The Problem of Provenance – The Bad Old Days

Bobbing for binaries...

-  Bad – Segmentation fault
-  Good – But how was it built?
-  Bad – Replaced a good binary
-  Bad – Intractable errors
-  Good – But no-one knows where it is
-  Bad – Insert your own lived experience here

The Problem of Provenance – The Bad Old Days

**Developers are human.
And, to err is human...**



The Solution with Spack + Automated Workflows

- Provenance via Spack
 - Build-from-source package manager specialising in HPC systems
 - Manifest specification of dependencies, variants, versions, compilers...
 - Installs centrally, caches components, returns lockfile for repeatable builds

```
spack:
specs:
- mpi@5.0.5
```

Spack Manifest

Simple YAML syntax
Abstract

Concretization →

```
Input spec
-----
- mpi

Concretized
-----
[+] openmpi@5.0.5%gcc@13.2.0~atomics~cuda build_system=autotools arch=linux-rocky8-x86_64
[+] ^autoconf@2.72%gcc@8.5.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^gcc-runtime@8.5.0%gcc@8.5.0 build_system=generic arch=linux-rocky8-x86_64
[e] ^glibc@2.28%gcc@8.5.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^m4@1.4.19%gcc@8.5.0 build_system=autotools patches=9dc5fbd,bfdffa7 arch=linux-rocky8-x86_64
[+] ^diffutils@3.10%gcc@8.5.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^libsigsegv@2.14%gcc@8.5.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^automake@1.16.5%gcc@8.5.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^gcc-runtime@13.2.0%gcc@13.2.0 build_system=generic arch=linux-rocky8-x86_64
[e] ^glibc@2.28%gcc@13.2.0 build_system=autotools arch=linux-rocky8-x86_64
[+] ^gmake@4.4.1%gcc@8.5.0~guile build_system=generic arch=linux-rocky8-x86_64
```

Installation →

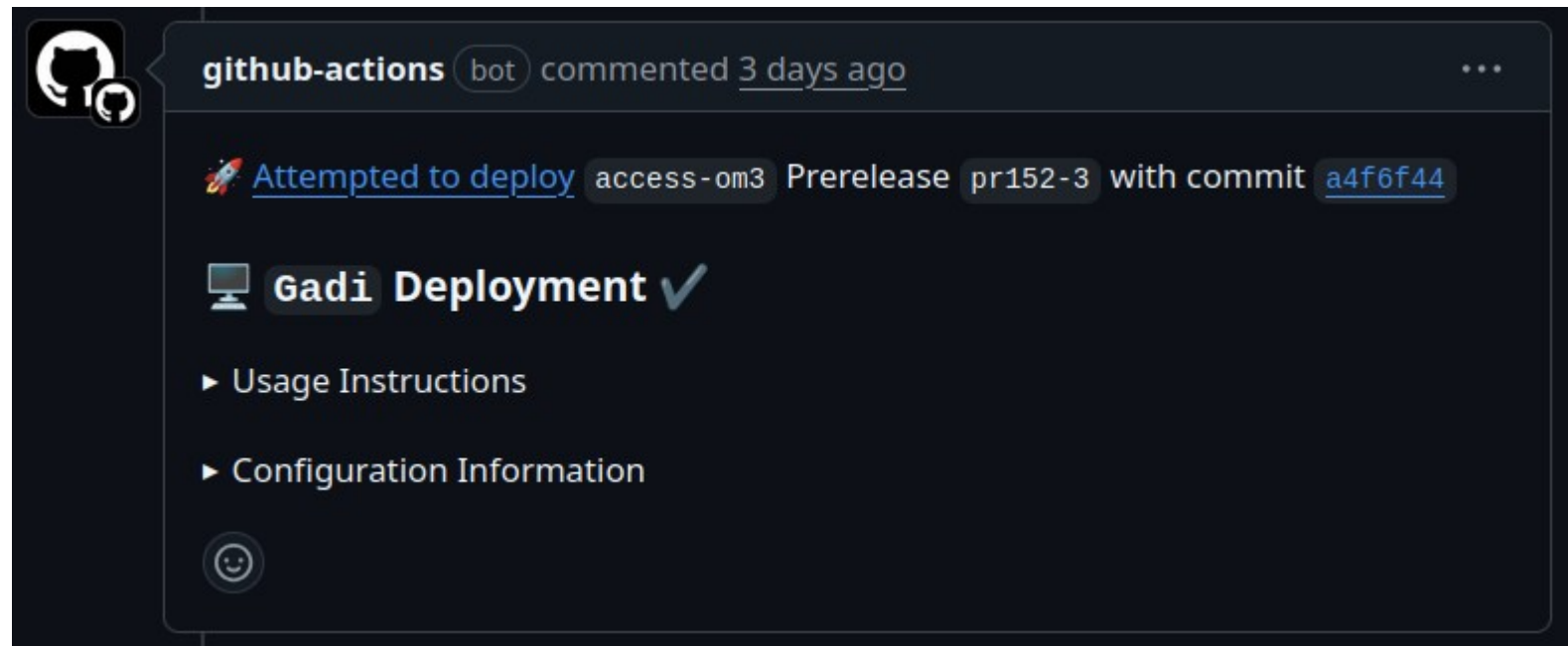
```
{ "_meta": {"file-type": "spack"}, "spack": {"version": "0.22.5", "99f79c"}, "roots": [{"hash": "mglfhd", "spec": "openmpi@5.0.5"}, {"hash": "cas6u7o", "spec": "mpi@5.0.5"}], {"name": "openmpi", "version": "5.0.5", "arch":
```

Spack Lock File







Machine Readable
Concrete
Reproducible!

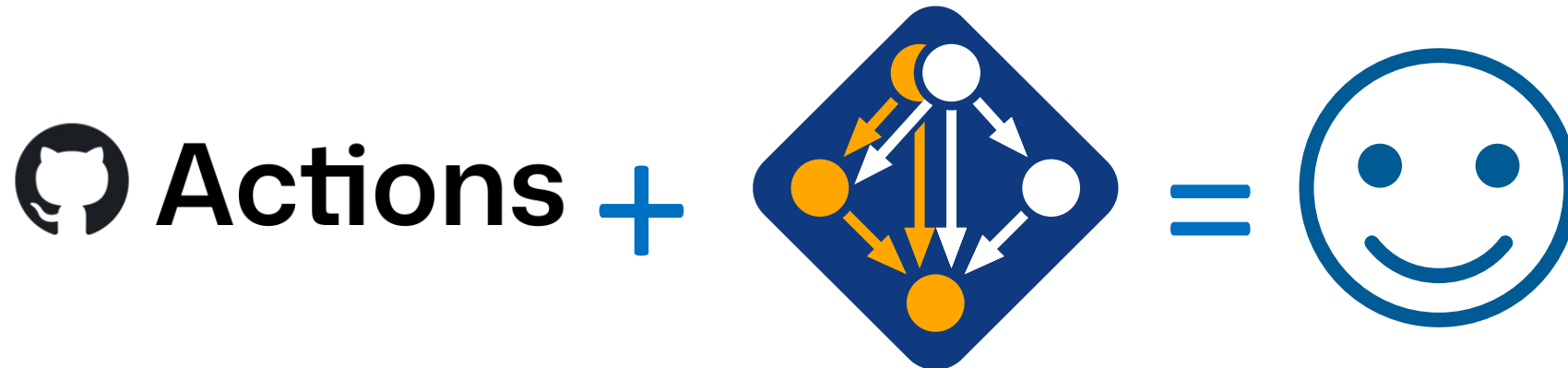
The Solution with Spack + Automated Workflows

- Provenance via Spack
- Automation via GitHub Actions
 - Allow triggering of Prerelease and Release builds from Pull Requests
 - Quick, transparent feedback cycle in the Pull Request
 - Generic, reusable workflows for our models, components, software



The Solution with Spack + Automated Workflows

- Provenance via Spack 
- Automation via GitHub Actions 
- Both led to happier...
 - Developers – Build provenance is clear, automatic, reusable 
 - Users – Successful builds are in a central location  
 - Infrastructure Maintainers – Generic workflows = no bespoke solutions 



Infrastructure Run-Through – Big Picture

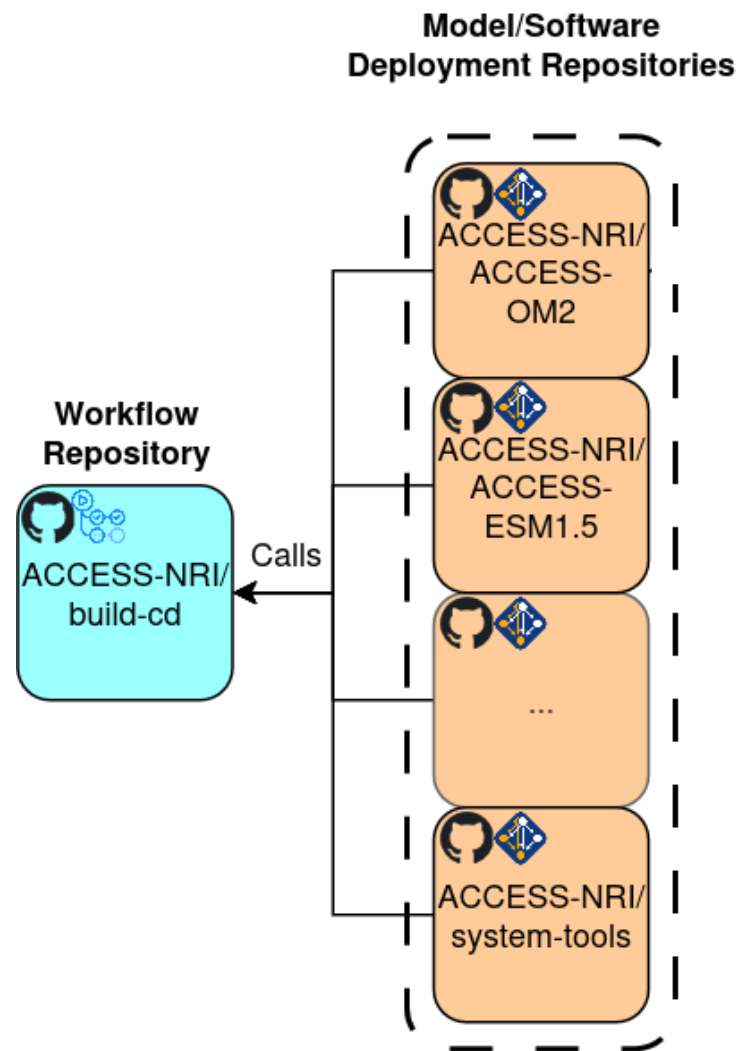
Workflow
Repository



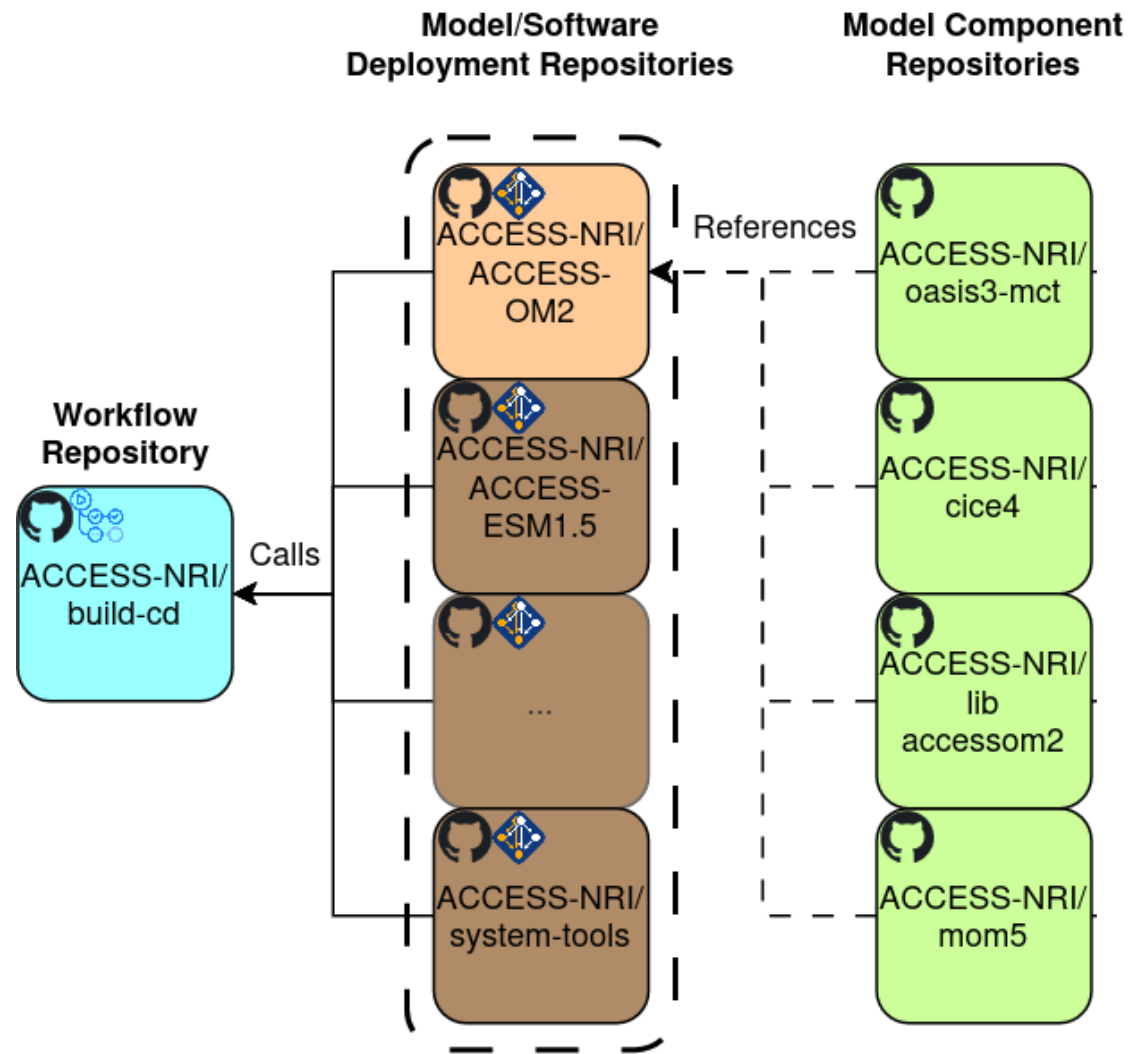
Called in GitHub Actions Workflows via:

```
uses: access-nri/build-cd/.github/workflows/ci.yml@v5 #(or v5.1)
with:
  model: access-om2
  spack-manifest-schema-version: 1-0-7
  ...
```

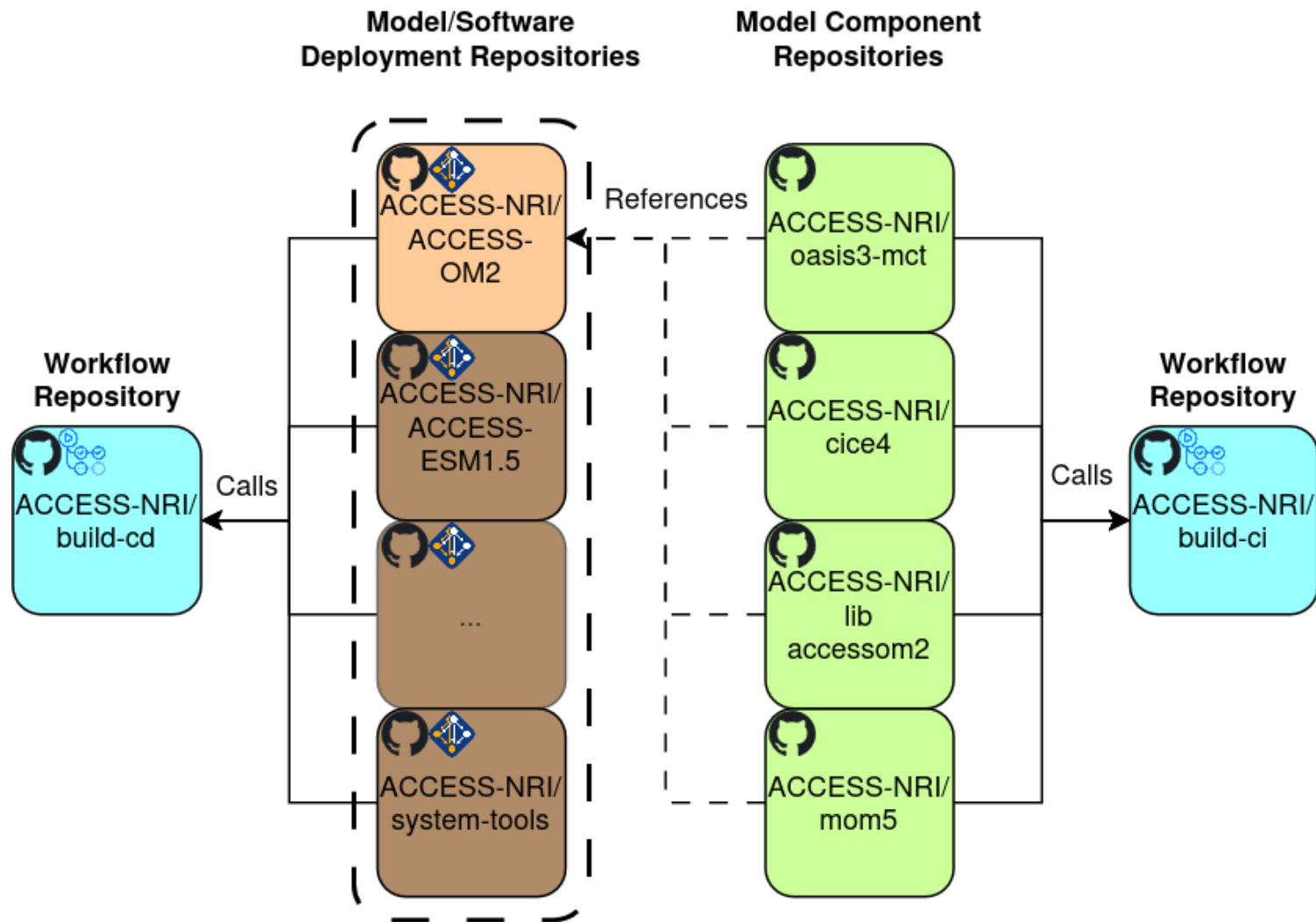
Infrastructure Run-Through – Big Picture



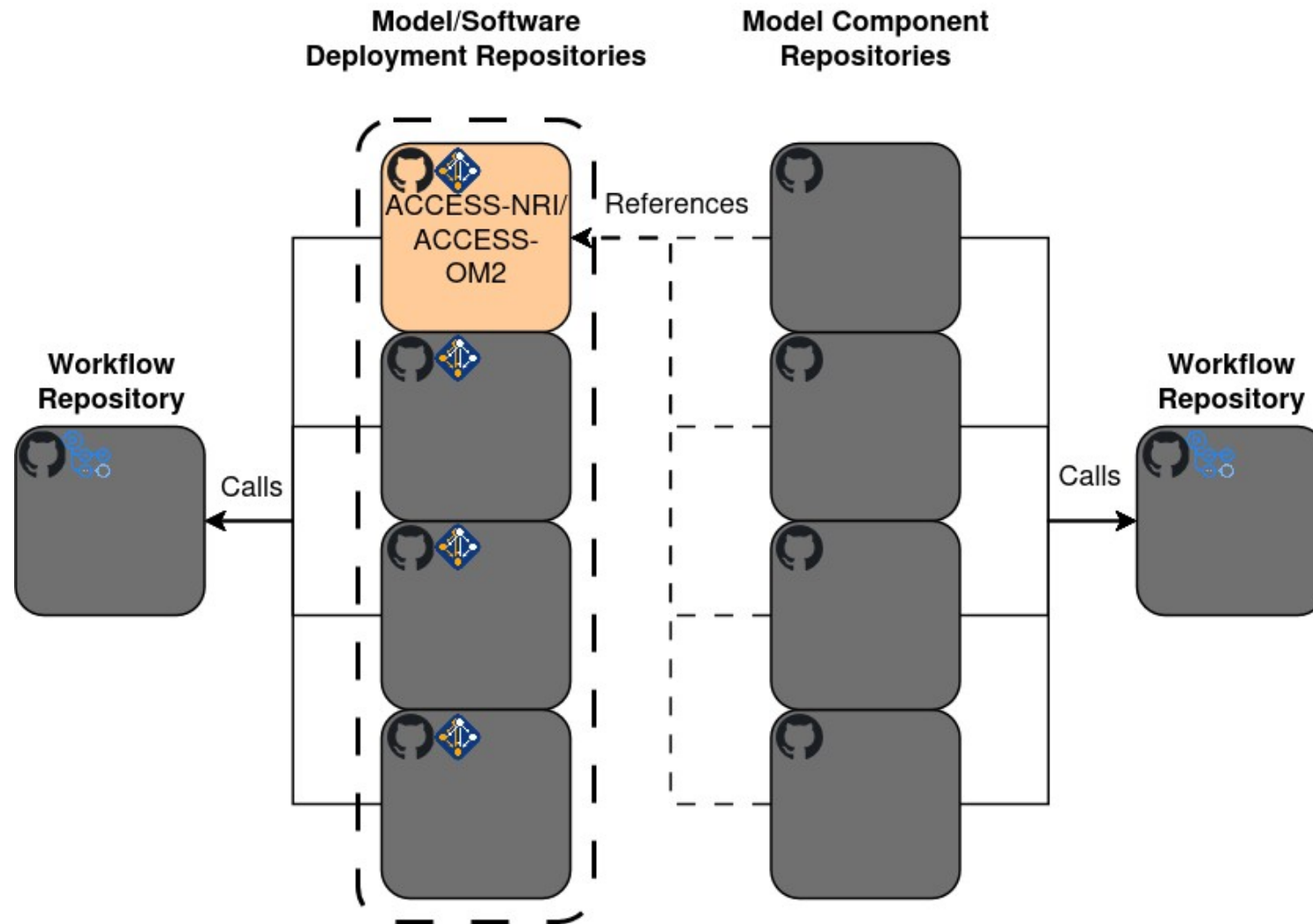
Infrastructure Run-Through – Big Picture



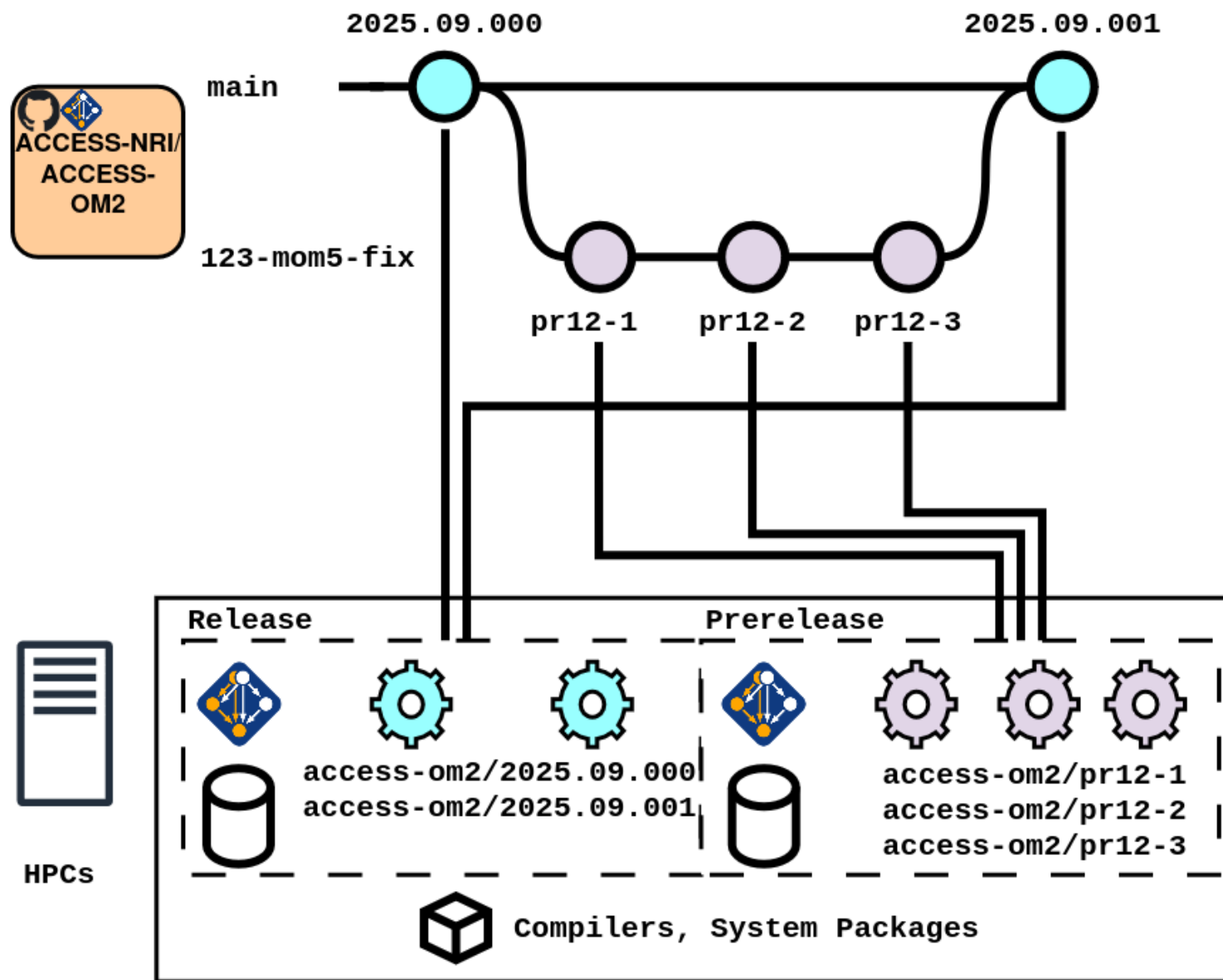
Infrastructure Run-Through – Big Picture



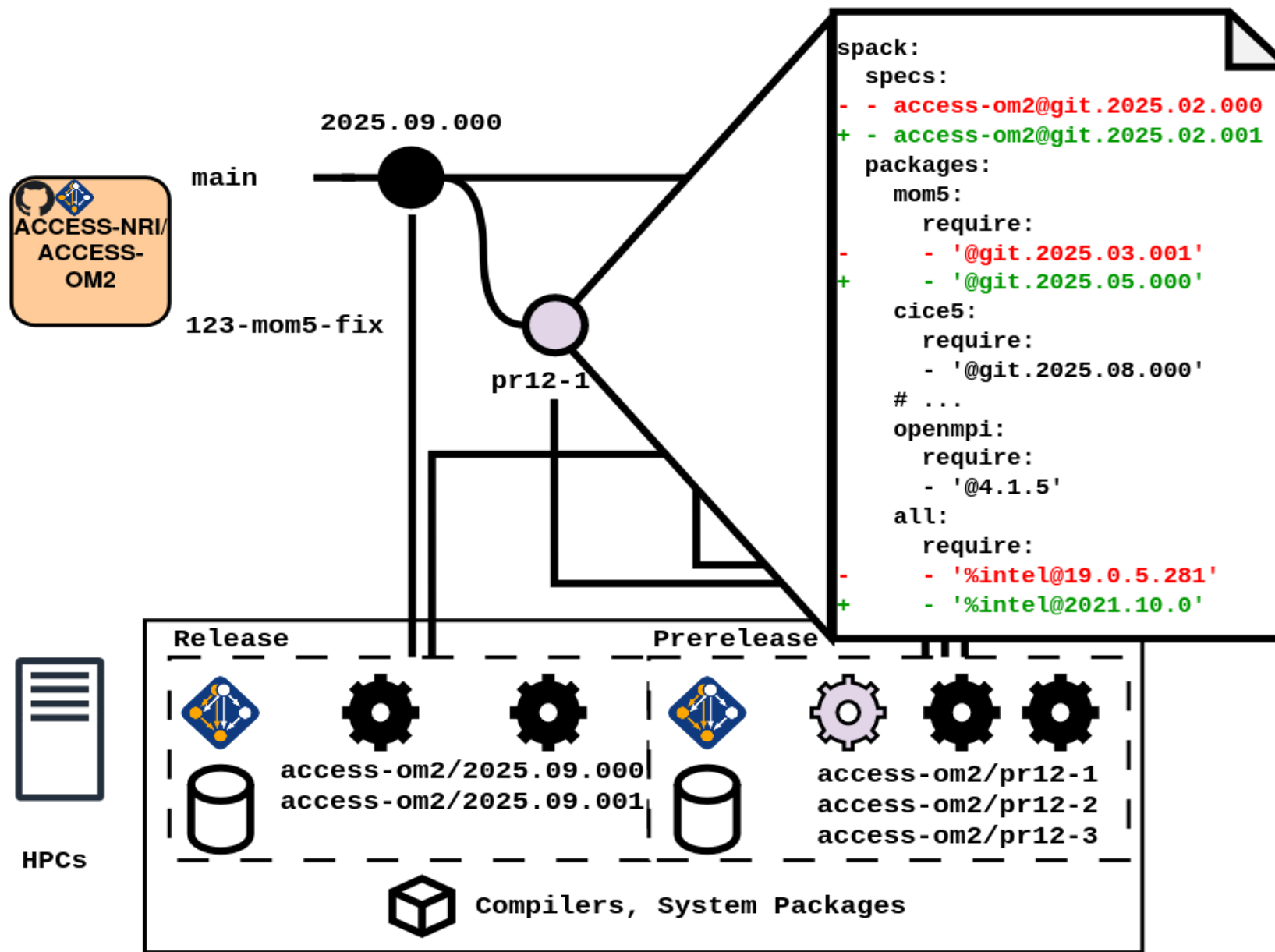
Infrastructure Run-Through – Big Picture




Infrastructure Run-Through – OM2 Repo Example



Infrastructure Run-Through – OM2 Repo Example





The User Perspective – In Their Own Words

- **Provenance** - "Others could check what was tested [in the PR] and testing was documented for the future"
- **Reduced Barrier To Entry** - "It turns 'I can't build it' into 'I can start experimenting.' The shift from struggling with toolchains to focusing on science, is where this really shines"
- **Caching** - "Spack's build caching is foundational to the ambition of completely modular, scalable deployment workflows"
- **Discoverability** - "lets others see (and use) exactly what has been done"

Final Thoughts

- Climate model provenance and discoverability is paramount and automatable
- Spack gives us full build provenance, reproducibility and build caching of our software
- GitHub Actions tie this together in modular, generic workflows, allowing us to scale quickly
- Both developers and researchers benefit from the use of these tools!
- This is applicable to a wide variety of types of software



Australia's climate simulator

A decorative horizontal bar at the bottom right of the page, consisting of several colored segments: blue, light blue, dark blue, green, and two small yellow-green dots.