



Federated Numerical Weather Prediction Workflows with MPAS



Christopher W. Harrop & Naureen Bharwani

ParslFest - Sep 26-27, 2024



Our Collaborators

This work is funded by the NOAA Software Engineering for Novel Architectures (SENA) Program*

In collaboration with U.S. Department of Energy, University of Chicago, & Parallel Works (funded by DOE Small Business Innovation Research)

Special thanks to:

- Eric Schnepf (NOAA/GSL/ITS/RDHPACS)
- Stefan Gary (Parallel Works)
- Kyle Chard, Yadu Babuji, and Ben Clifford (University of Chicago / ANL)

*NOAA cooperative agreement NA22OAR4320151

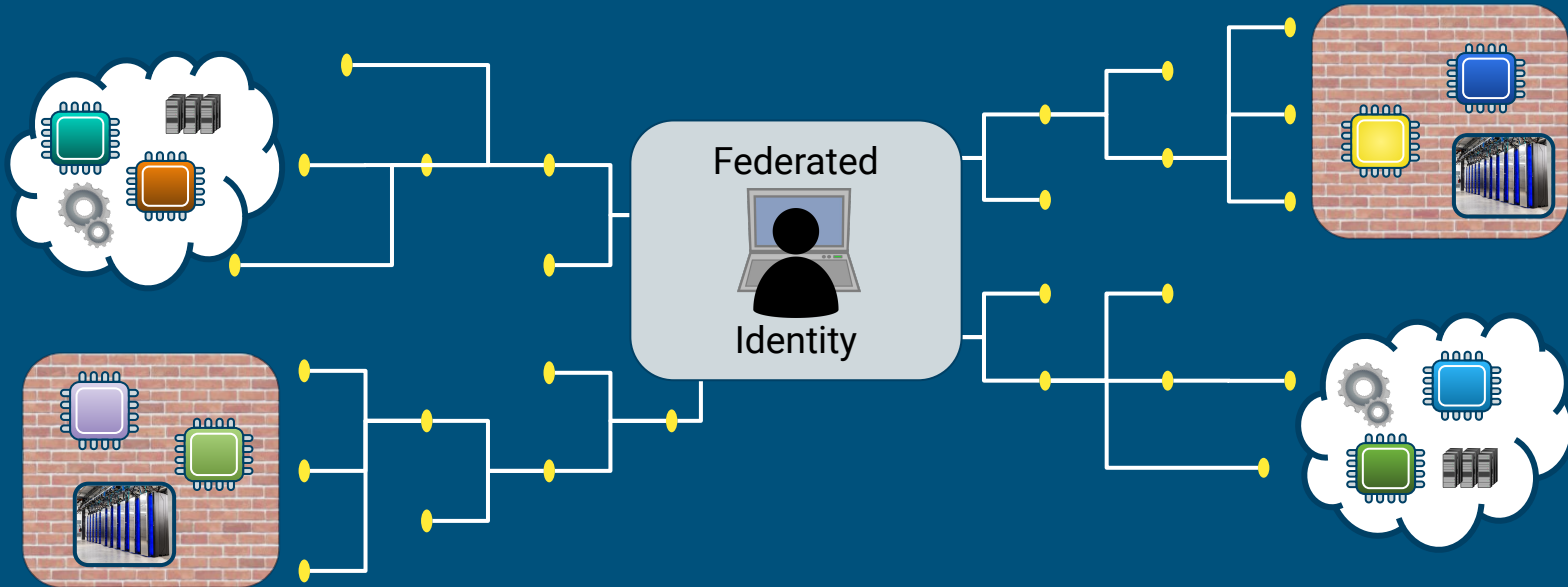
Emerging Technologies + NWP

We face two main workflow challenges:

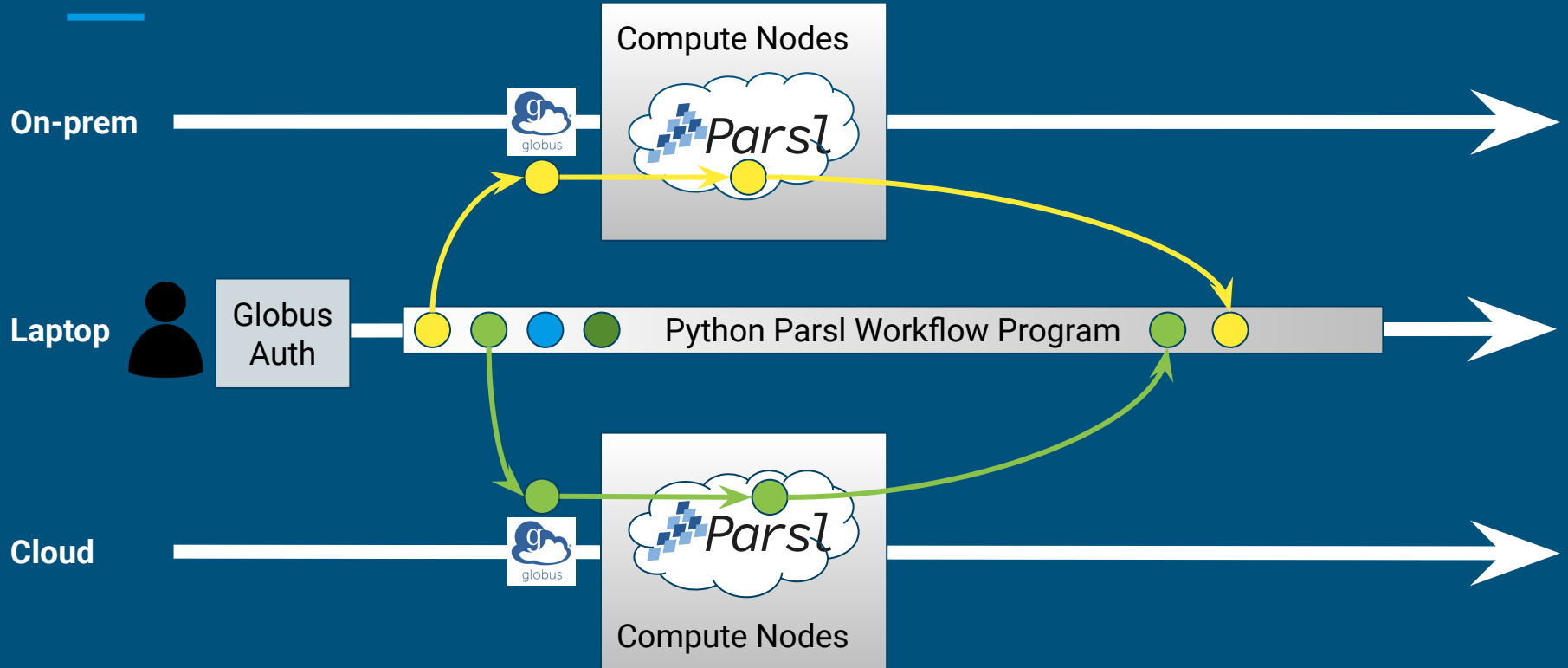
- **Scale** of problems we want to investigate (we need > 1M cores)
 - No single environment is large enough to contain the computations we want to run
 - < 4 Km global simulations, digital twins & ML training data
 - 100+ ensemble members
- **Diversity** of emerging technologies, applications, and compute environments
 - No single environment has all the resources we require
 - CPUs, GPUs, TPUs, Quantum processors & Intel, AMD, NVIDIA, ARM
 - Proprietary data stores, Data lakes, NVRAM, test systems

Diversification & Scale → Federated Workflows

Our Vision: Diversified **federated** workflows to enable NWP HPC/ML Research



Federated Compute & Automation Services



Collaboration → Key Developments for NOAA

- RDHPCS / Globus teams working toward mutual plan for security to enable use of Globus Compute Endpoints on NOAA on-prem (**ongoing, long-term**)
 - Direct support for MPI applications in Parsl (**done ✓**)
 - New MPIExecutor
 - Direct support for MPI applications in Globus Compute (**done ✓**)
 - New GlobusMPIEngine
- Removal of Flux & associated complexities from Chiltepin stack (**done ✓**)

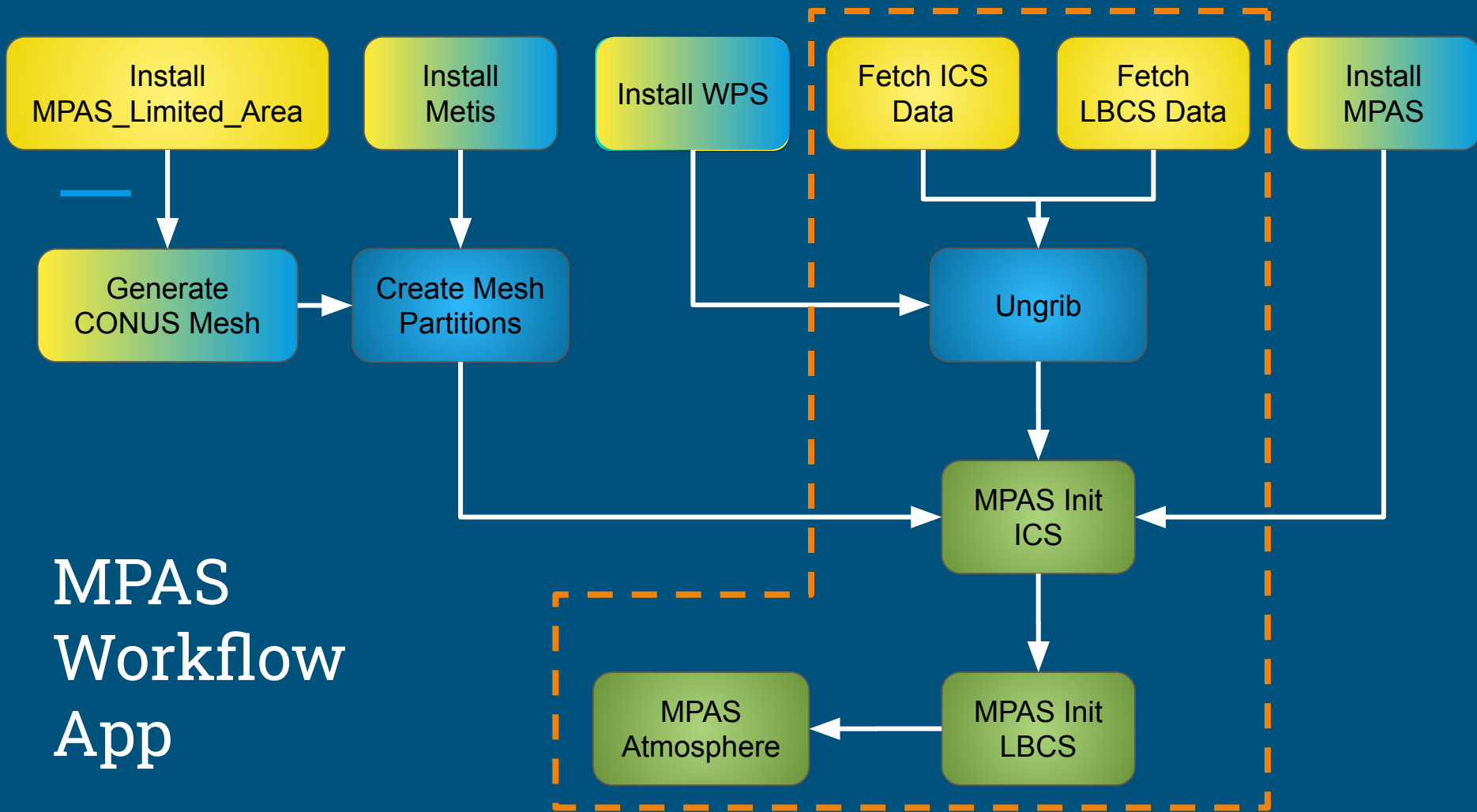
Introduction to *Chiltepin*

Chiltepin → A Python Library for Building Federated NWP Workflows

- Creation of NWP workflow “endpoints”
- Configuration of resource pools on multiple systems
- Encapsulation of NWP task configuration & execution
- Wrap uwtools as federated Parsl & Globus Compute workflow tasks
- Workflows are Python programs

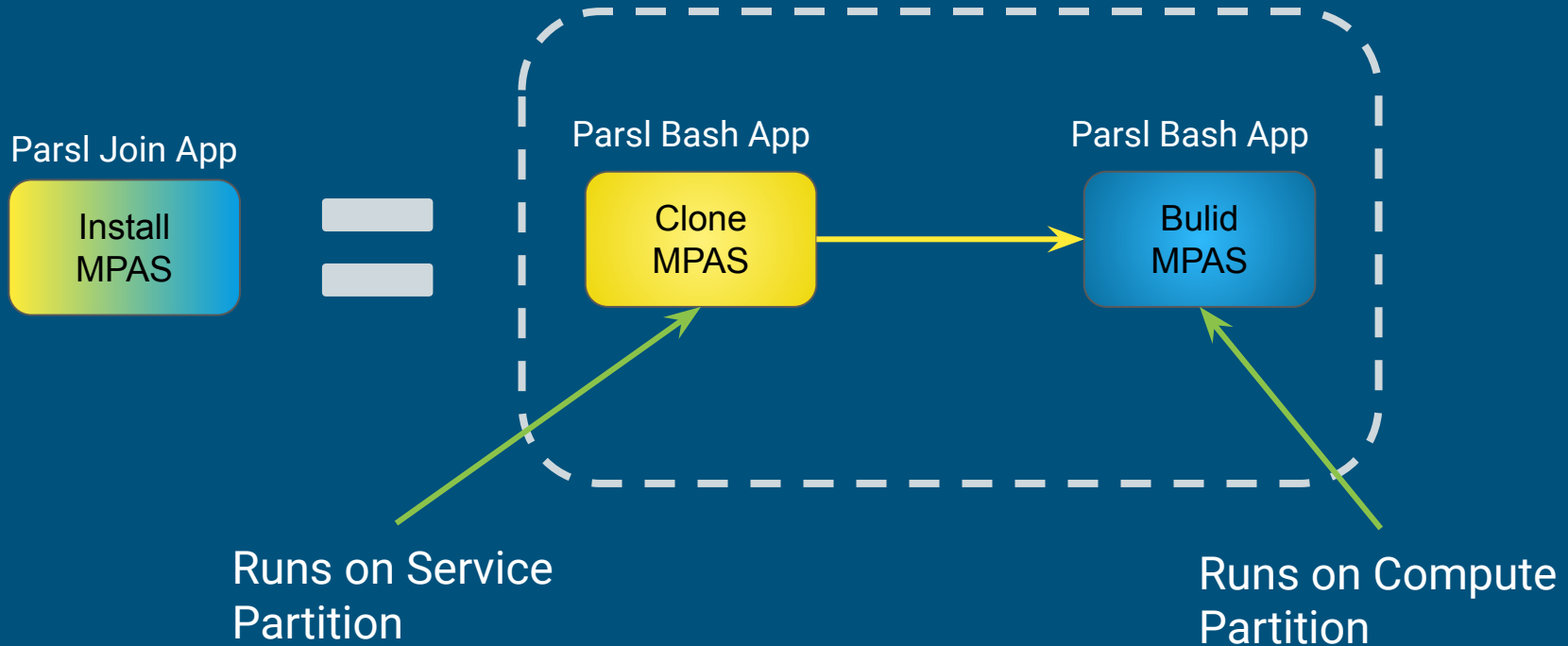
uwtools → A Python Package to automate common NWP workflow tasks

- Configuration management
- Multi-stage drivers for UFS model components
 - Provisioning of run environment
 - Execution of binaries



MPAS Workflow App

Workflow Tasks May Also Be Mini-Workflows



Chilstepin Workflows

Run ungrib

```
ungrib = wps.ungrib(config, cycle_iso, ...)
```

Create initial conditions

```
mpas_init_ics = mpas.mpas_init(config, cycle_iso, "create_ics", ungrib, ...)
```

Workflow futures inject dependencies

Create boundary conditions

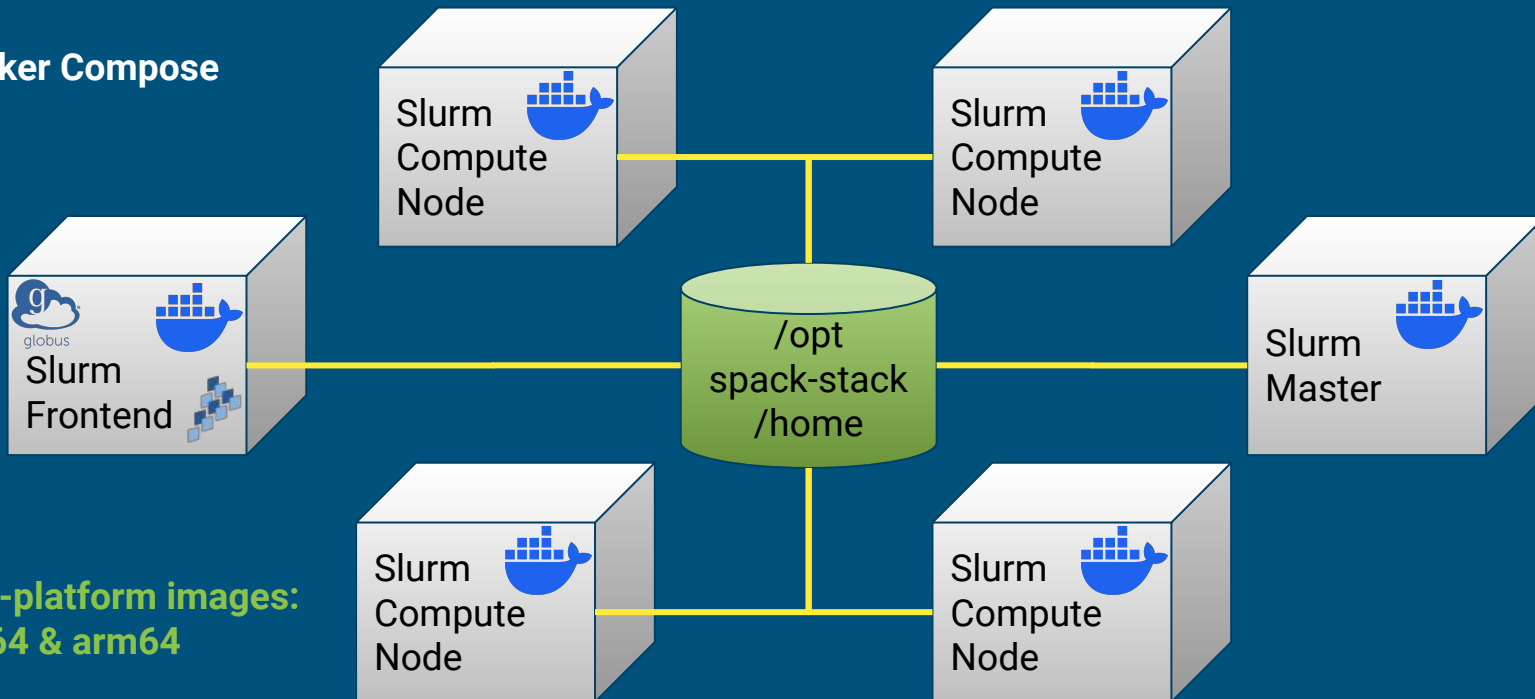
```
mpas_init_lbc = mpas.mpas_init(config, cycle_iso, "create_lbc", mpas_init_ics, ...)
```

Run the forecast

```
mpas_forecast = mpas.mpas_forecast(config, cycle_iso, mpas_init_lbc, ...)
```

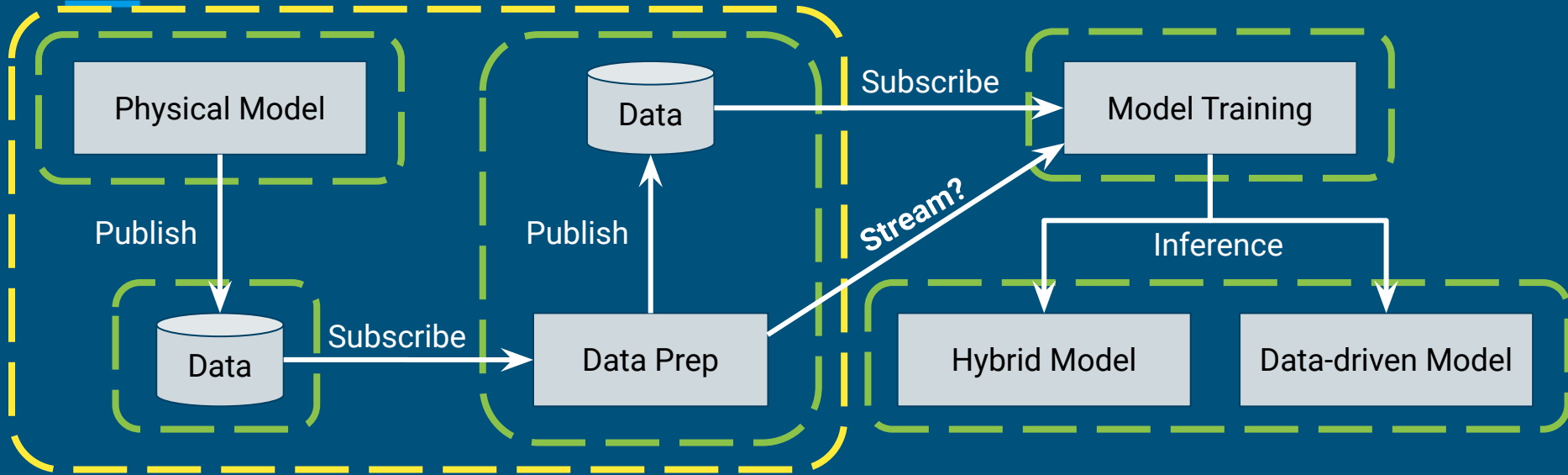
Chiltepin CI Testing With Containers

Docker Compose



Multi-platform images:
amd64 & arm64

Looking Ahead - ML/AI



- Globus Flows?
- Move data to where it's needed
- Deploy compute on optimal resources (e.g. NVIDIA H100s)



Questions / Discussion

