NEXT GENERATION WATER MODEL DATASTREAM

JORDAN LASER

ZACH WILLS

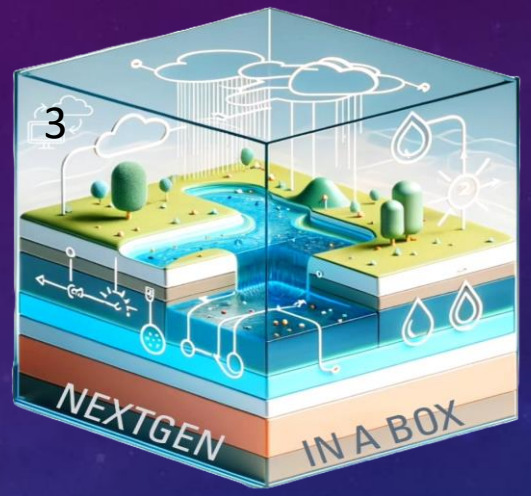
NELS FRAZIER

SPRING 2024

NEXTGEN Water Modeling Framework Datastream Conceptual Model

1 Prepare configuration and data inputs

2 Validation



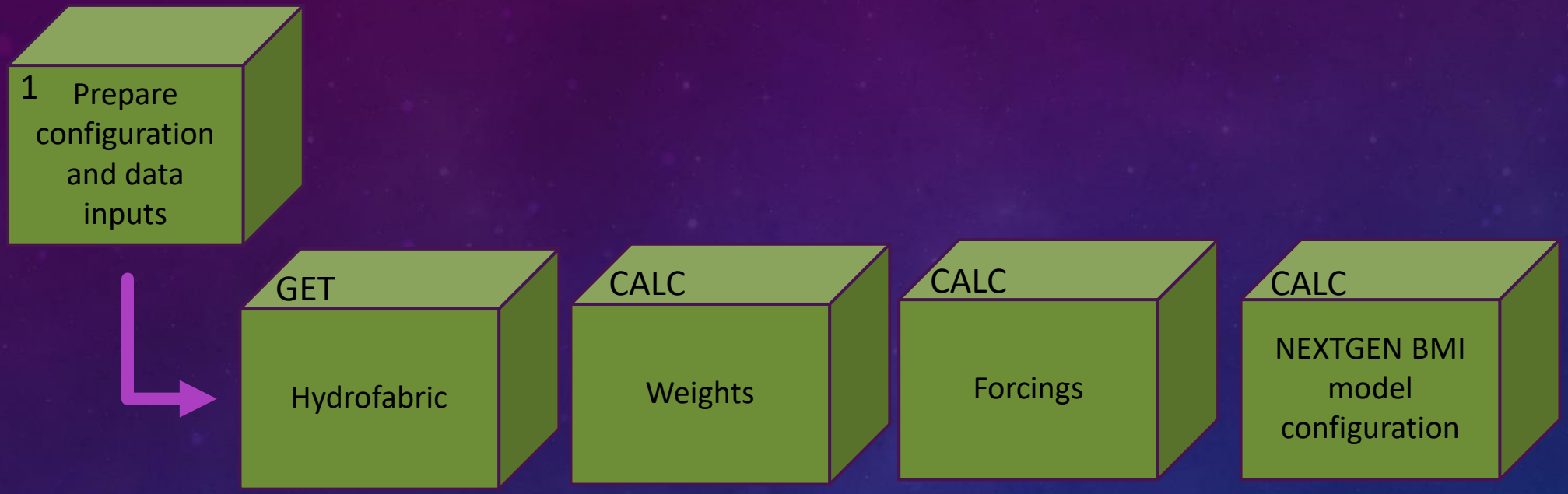
4 Versioning

NGEN-DATASTREAM

https://github.com/CIROH-UA/ngen-datastream/tree/main

ngen-datastream refers to the software chain that builds valid ngen-run input packages, executes NEXTGEN with NGIAB, and versions the entire run. NEXTGEN is a framework that coordinates hydrologic model configuration, parameterization, and execution across spatial domains. NGIAB consists of the NEXTGEN docker container pre-built for the user's architecture as well as software that aids users in properly running the container.

NEXTGEN Water Modeling Framework Datastream Conceptual Model Breakdown



Required for

Defining spatial domain and NEXTGEN BMI model configuration

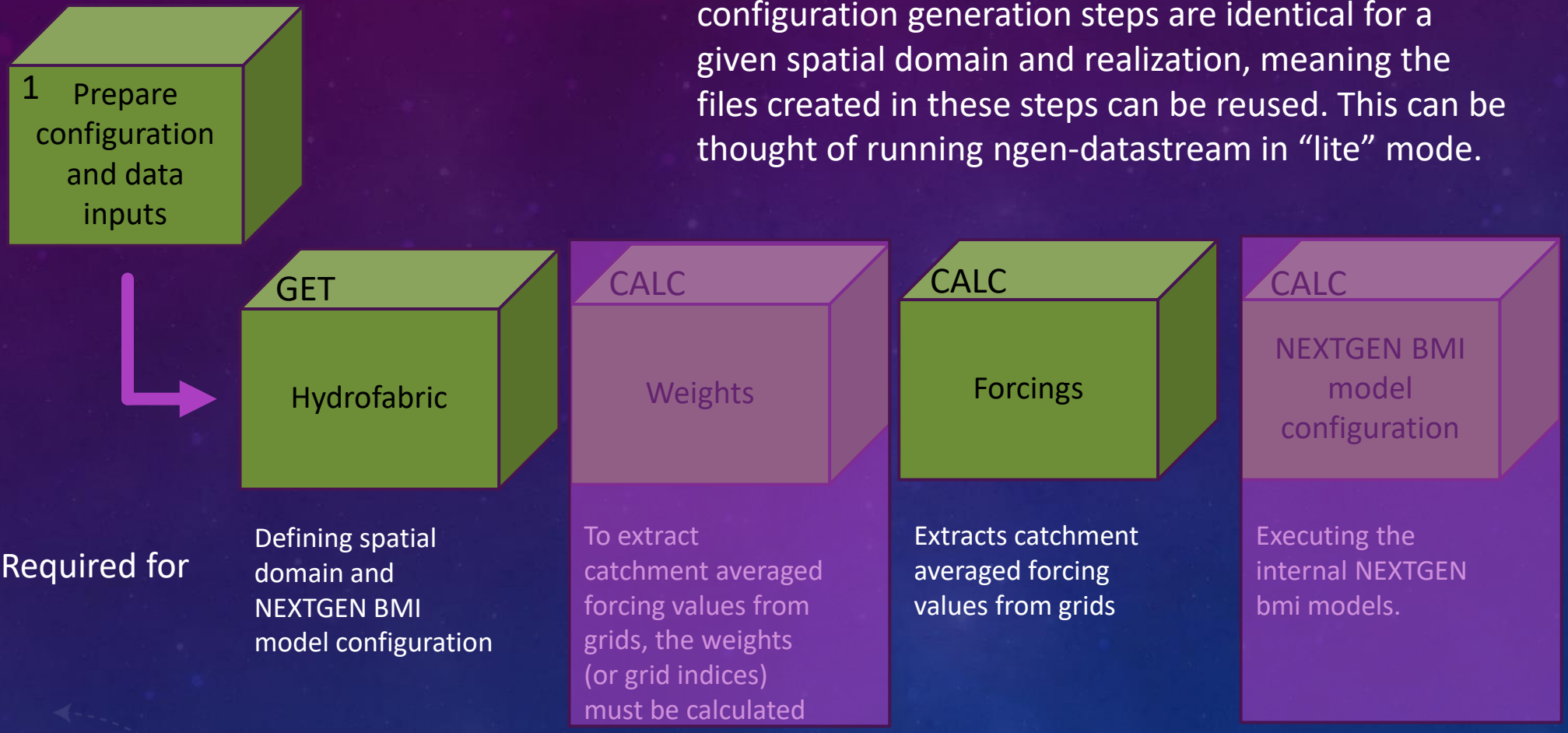
To extract catchment averaged forcing values from grids, the weights (or grid indices) must be calculated

Extracts catchment averaged forcing values from grids

Executing the internal NEXTGEN bmi models.

NEXTGEN Water Modeling Framework Datastream Conceptual Model Breakdown

The calculations in the weights and BMI model configuration generation steps are identical for a given spatial domain and realization, meaning the files created in these steps can be reused. This can be thought of running ngen-datastream in “lite” mode.

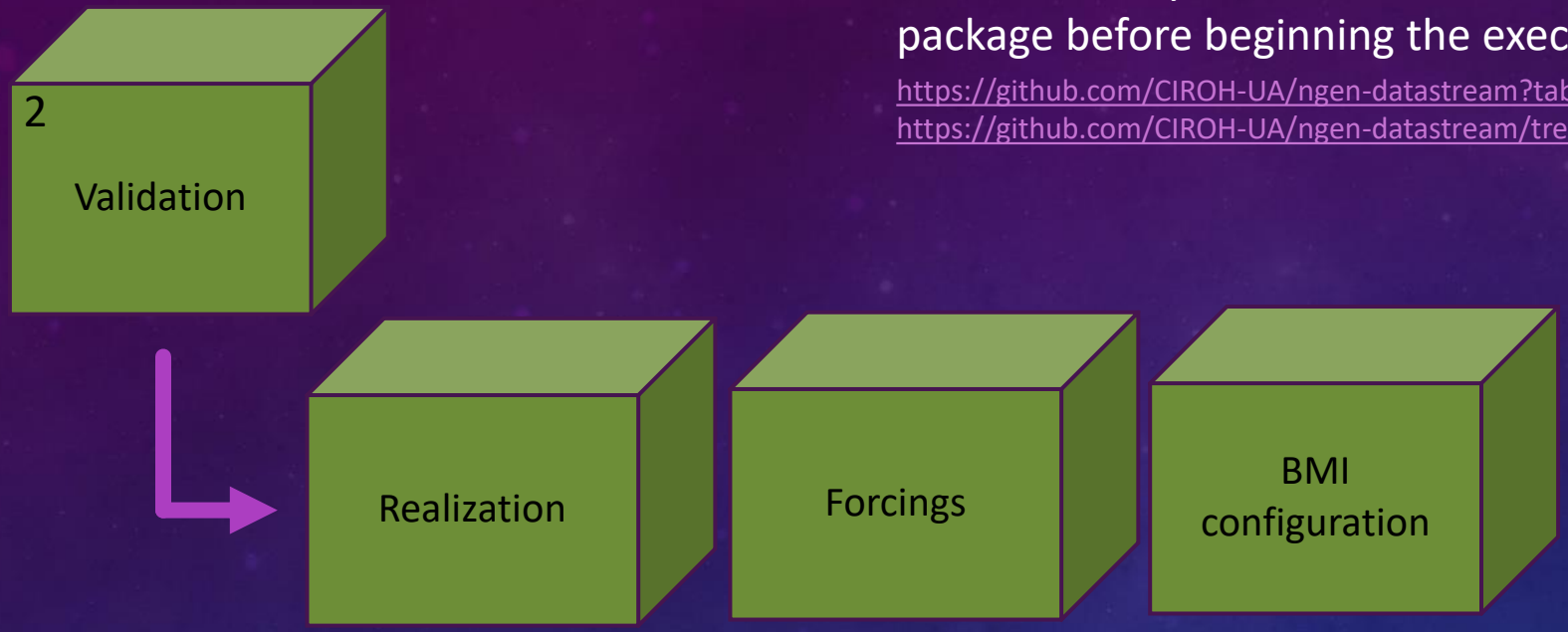


NEXTGEN Water Modeling Framework Datastream Conceptual Model Breakdown

TLDR: The value of the validation step is to notify the user of any errors in the NEXTGEN run package before beginning the execution

https://github.com/CIROH-UA/ngen-datastream?tab=readme-ov-file#ngen-run
https://github.com/CIROH-UA/ngen-datastream/tree/main/python#run_validatorpy

```
ngen-run/
├── config/
├── forcings/
├── metadata/
└── outputs/
```



Required for

Ensures the user has supplied a valid realization file to configure NEXTGEN

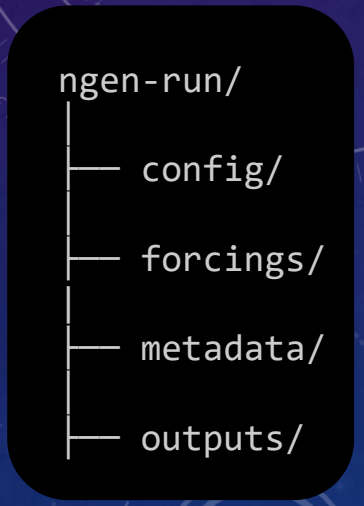
Ensures a forcing file exists for each catchment in the hydrofabric and for each time step specified in the realization

Ensures all BMI model configuration files exist.

For example, t-route requires a single configuration YAMl, whereas PET and CFE require a configuration file for each catchment

Input explanations for NGIAB/NEXTGEN

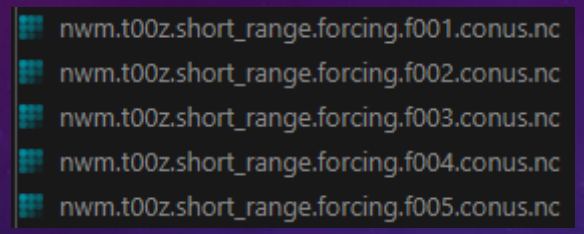
	Configuration	Hydrofabric	Forcings
Physical Explanation	Deciding how to model hydrologic data is not easy. There's a ton of water models with different versions, configurations, and parameterizations that makes running and interoperation difficult. To add to the complexity, the performance of these models has been found to depend strongly on the environment, implying some models work better than others depending on the spatial domain. NextGen gives users control over modeling this complexity.	The hydrofabric is the spatial data that describes the physical location of things like the borders and topology of a catchment, nexus points, and flowpaths.	Forcings are the physical variables needed to "force" NextGen from one time step to the next. Forcings variables include temperature, precipitation, and solar longwave and shortwave intensities. These files are per catchment time series of these variables.
Computer Science Explanation	Here is where the user supplies NextGen with the information it needs to execute a "run". This includes information like the time range, model selection, model configuration and parameterization. It is possible to specify the model selections on a per catchment basis. This allows the user to run different models over different areas and can dramatically increase the complexity of this file. The goal of NGIAB is to dynamically produce this file via a GUI. e.x. ngen-run/config/realization.json	This data is most readily available in .geopkg per VPU. Users often subset these .geopkg's into smaller domains with tools like hfsubset. NextGen only accepts .geojson format, so users will then need to conver their .geopkg to .geojson with tools like og2og2 e.x. ngen-run/config/catchments.geojson	More on how to create these files later. e.x. data_dir/forcings/cat-12.csv

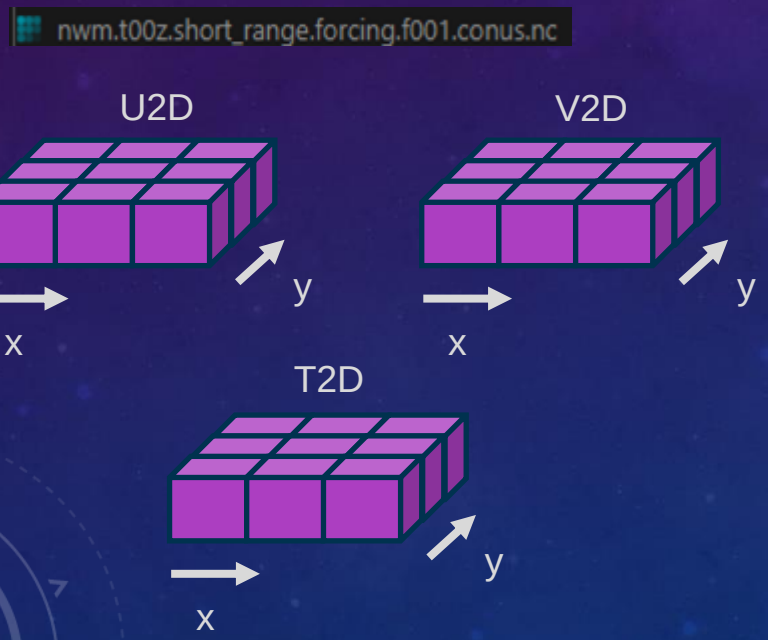


Why do we need a forcingprocessor?

NWM output forcing data

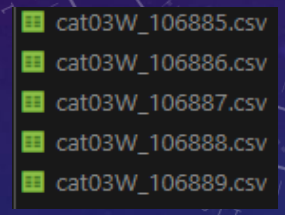
- Gridded
- CONUS wide
- One file for each forecast time

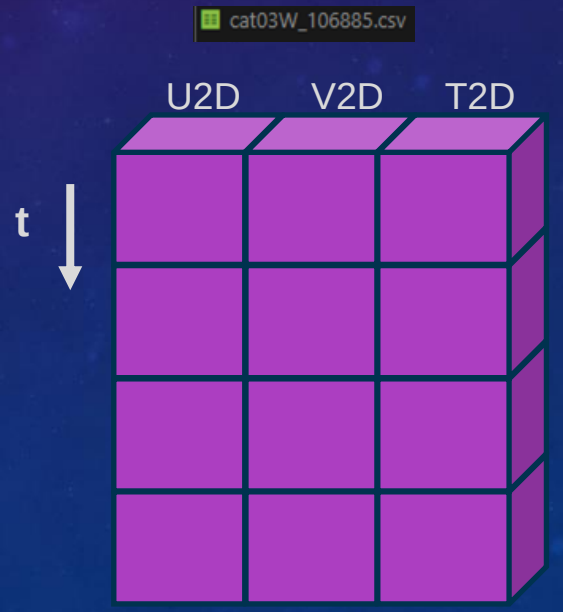


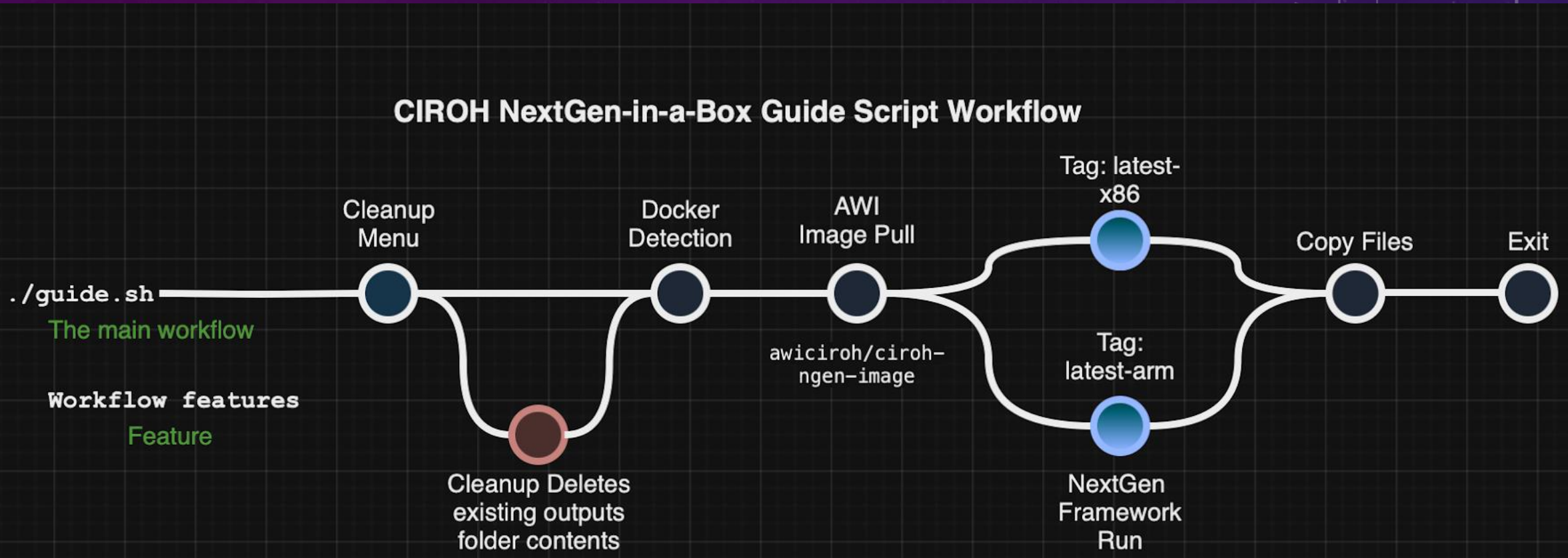


NextGen input forcing data

- Non-Gridded
- Time Series within each file
- One file for each catchment






CIROH NextGen-in-a-Box Guide Script Workflow
./guide.sh
The main workflow
Workflow features
Feature
Cleanup Menu
Cleanup Deletes existing outputs folder contents
Docker Detection
AWI Image Pull
awiciroh/ciroh-ngen-image
Tag: latest-x86
Tag: latest-arm
NextGen Framework Run
Copy Files
Exit

TERMS

- NGIAB – Next Generation National Water Model in a Box
- NGEN - Next Generation National Water Model
- Catchment – geographic area characterized by a single location, a nexus, where all precipitation in the area runs off through. A drainage basin.
- Nexus – the singular point where water flows into or out of a catchment. Often a point along a river.
- Subsetting – To reduce a large geopackage (many catchments) down to a smaller geopackage (fewer catchments) . In effect, this is choosing the domain over which ngen will run.
- Hashing – SHA256 algorithm applied to files to generate a unique id for a file. Useful for preserving and distinguishing unique inputs.
- Validation – Ensuring the ngen input directory data_dir has been constructed properly. Properly meaning that NextGen will not crash and will generate output data.

LINKS

- DATASTREAM https://github.com/CIROH-UA/ngen-datastream/tree/main
- FORCINGPROCESSOR https://github.com/CIROH-UA/ngen-datastream/tree/main/forcingprocessor
- REALIZATION GENERATION AND NGEN-RUN FOLDER VALIDATION https://github.com/NOAA-OWP/ngen-cal
- HYDROFABRIC SUBSETTING https://github.com/LynkerIntel/hfsubset
- HASHING/VERSIONING https://github.com/aaraney/ht
- NGIAB https://github.com/CIROH-UA/NGIAB-CloudInfra
- https://docs.ciroh.org/
- https://docs.ciroh.org/docs/products/tools/nextgeninabox/ngiab-intro
- https://github.com/NOAA-OWP/ngen/wiki
- https://mikejohnson51.github.io/hyAggregate/
- https://ciroh.ua.edu/