

Integrating MET/METPlus in CSET

Jorge Bornemann, James Frost, Sylvia Bohnenstengel

Thanks to Phil Gill, Rob Darvel, Claire Bysouth, Rachel North

Momentum Convective Scale Workshop

Melbourne X – September - 2024

Climate, Freshwater & Ocean Science



NIWA
Taihoro Nukurangi

Requirements & Background

- NG_VER migration from VER to MET/METPlus
- Site-Specific:
 - Observations
 - Model output
- General:
 - Replicate parallel suites verification

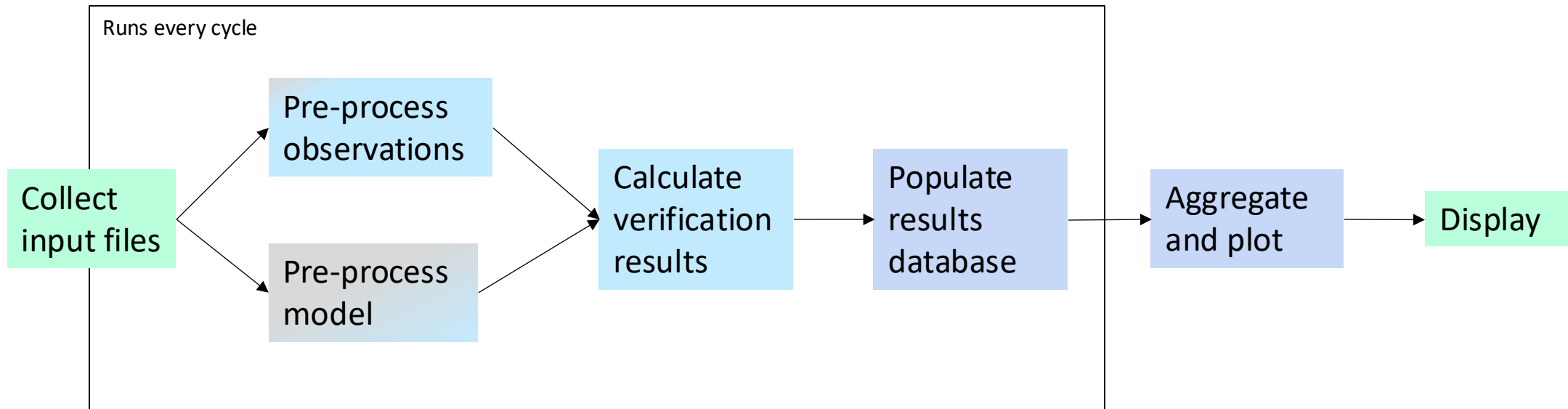
MET/METplus



- Collection of model verification tools developed at Development Testbed Center
- MET (<https://dtcenter.org/community-code/model-evaluation-tools-met>)
 - Core engine C++ (admits python bindings)
 - Highly configurable
- METplus (<https://dtcenter.org/community-code/metplus>)
 - Verification framework including **MET**, database and visualisation tools and **python wrappers**.

Workflow

CSET
METplus
VERPy
NG_VER



Observations and model fields pre-processing

- Fields files pre_processing:
 - NG_VER tool to convert files to MET-compliant netCDF files
 - Or make use of python embedding in MET statistic tool
- Observations pre-processing:
 - Use ASCII2NC MET tool to convert observations in MET-compliant netCDF files
 - Synoptic observations in ODB format
 - Use NG_VER tool to extract to ODB into ASCII files
 - Ingest ASCII files into ASCII2NC
 - Local observations (in whatever format)
 - Use python embedding to define obs retrieval process

Python embedding

- Allows calling the c++ code from python scripts.

CSET/cset-workflow/app/metplus_ascii2nc/file/niwa/ASCII2NC_python_embedding.conf

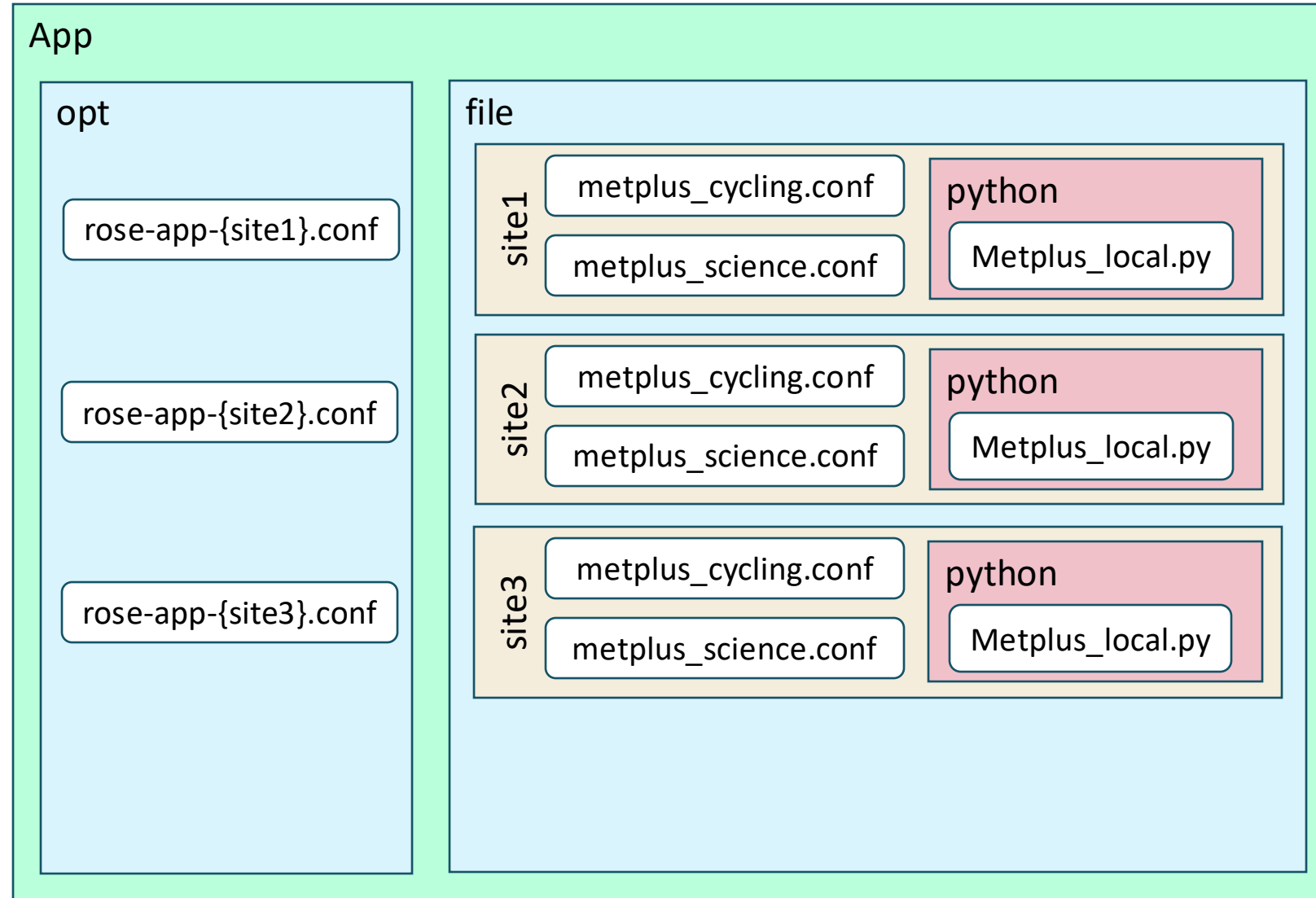
```
49 ASCII2NC_INPUT_DIR =
50 ASCII2NC_INPUT_TEMPLATE = "niwa/python/{ENV[INPUT_READ_SCRIPT]} {INPUT_BASE}/Surface.NiwaList.{init?fmt=%Y%m%dT%H} {valid?fmt=%Y%m%dT%H?shift=-10800}00 {ASCII2NC_FILE_WINDOW_BEGIN} {ASCII2NC_FILE_WINDOW_END}"
52 ASCII2NC_OUTPUT_DIR =
53
```

CSET/cset-workflow/app/metplus_ascii2nc/file/niwa/python/read_ascii_point.py

```
58 dtype={"typ": "str", "sid": "str", "vld": "str", "var": "str", "qc": "str"},
59 ).values.tolist()
60 print(f"    point_data: Data Length:\t{len(point_data)}")
61 print(f"    point_data: Data Type:\t{type(point_data)}")
62 met_point_data = convert_point_data(point_data)
63 print(f" met_point_data: Data Type:\t{type(met_point_data)}")
64 except NameError:
65     print("Can't find the input file")
66     sys.exit(1)
```

Apps file structure

- Exploit 'stackable' configuration files to separate cycling, file naming, etc... from statistics requested
- Allow to use python embedding if required



Verpy sqlite3 loader and plotting

- From VERPy-6.1:
 - SQLite3 loading from MET output files (ASCII files)
 - Can be done every cycle, allowing to housekeep the large number of files generated by MET
 - VERPy plotting routines can query the databases, aggregate results and produce png files
- VERPy also creates html pages to visualise the results, but we will try to integrate the verification plots in the CSET web page
- Assume types of statistics homogenised across the Partnership, but optional task configurations suitable to handle special cases.

Progress

- Done:
 - ASCII2NC, Point-Stat, Grid-Stat working at NIWA
 - VERPy loading working at NIWA (but outside CSET)
 - Being implemented at Met Office
- Still to do:
 - Ensemble-Stat
 - VERPy plotting
 - Integrate plots in web page
 - Onboard other Momentum Partners
 - Single programming environment for CSET and METPlus tasks

Conclusion

- MET/METplus is the tool of choice for objective verification in CSET
- METplus wrappers configured as tasks in CSET Workflow
- Rose/Cylc optional configurations, MET python embedding and METplus “stackable” configuration files provide the flexibility for site-specific requirements to be catered for, while keeping intact the structure of the workflow.
- Still apps to add for full functionality
- Momentum Partners configurations to be added
 - Please reach out to start adding config files

Thank you

Climate, Freshwater & Ocean Science

