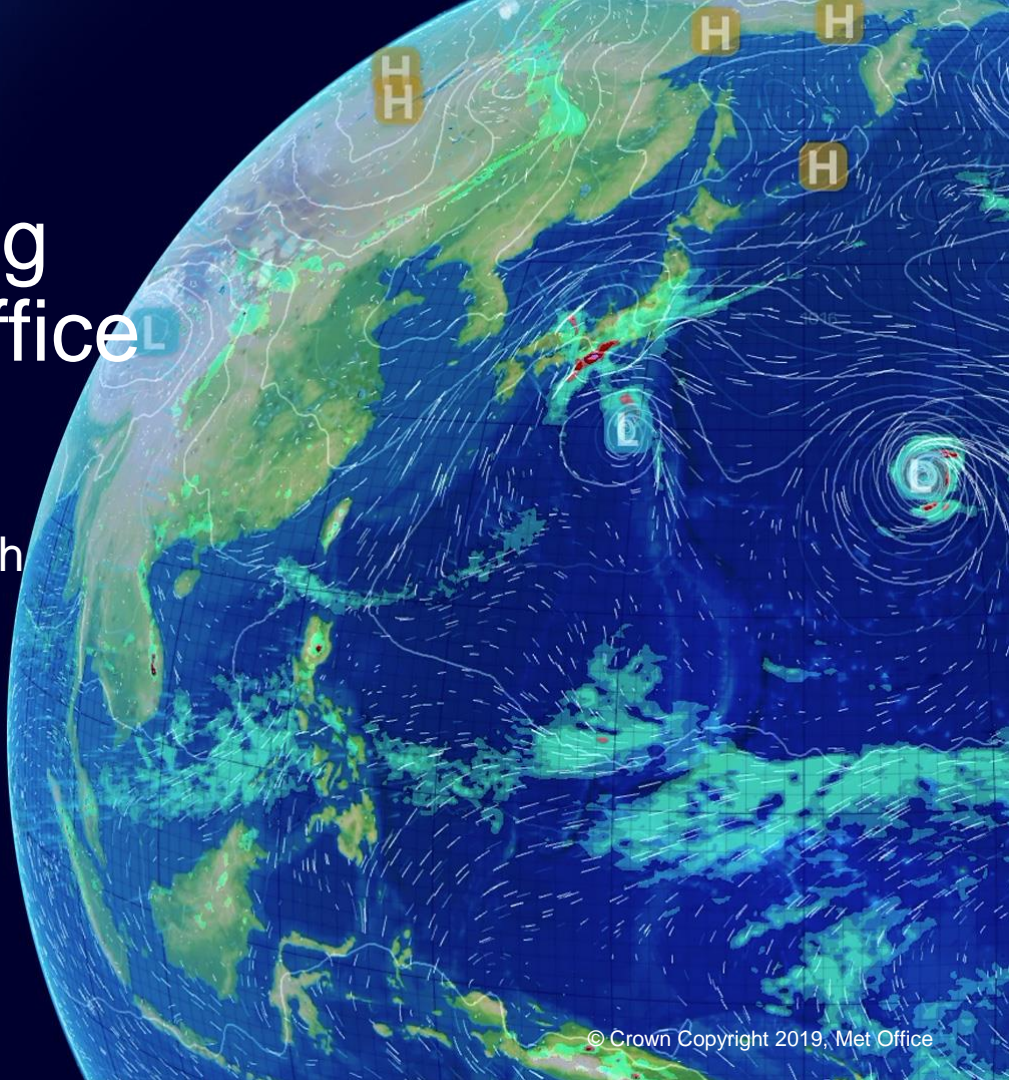


CMIP6 post-processing workflow at the Met Office

By Stephen Haddad

Building reproducible workflows for earth
sciences

15 October 2019



Introduction to CMIP6

WCRP



World Climate Research Programme

- Coupled Model Intercomparison Project – 6th phase
- Gather data in a standard format for Climate Researchers
- Analysis and research based on standard multi-model ensemble

CMIP6 Overall Workflow



Define Experiments



Run Simulations



Produce standardised data



Publish Data



Use of data for analysis and
reference



Design requirements

- Reproducible
- Flexible / Responsive
- Efficient
- Extendable / Portable
- Traceable / Transparent
- Standards Compliant

Inline vs Post- processing

Process with simulation run

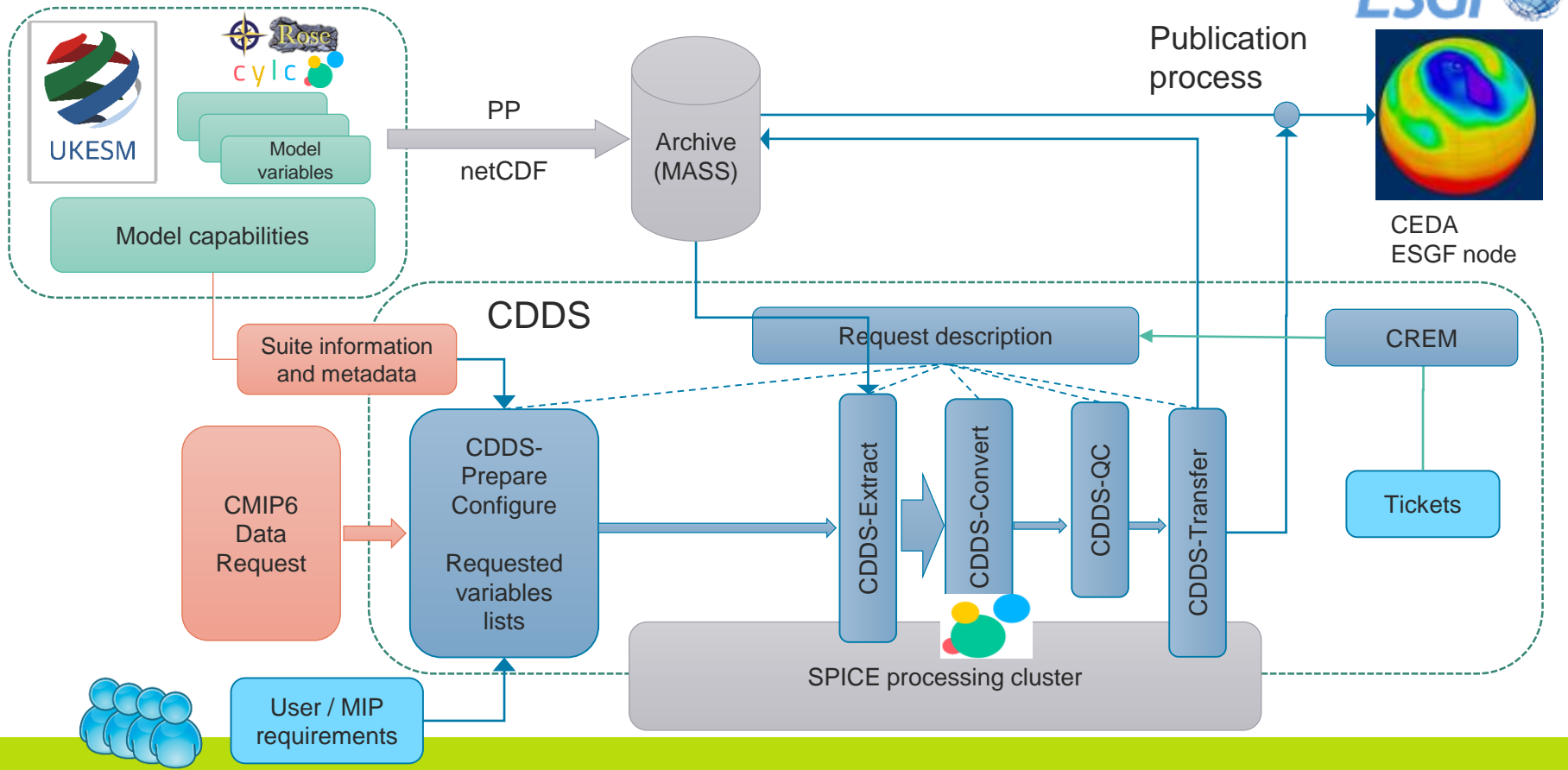
- Only store data for CMIP
- Requires finalised definition before simulation run

Archive output and process later

- Simulation can run before experiments being finalised
- Decouples simulation and data production code
- Store more data “just in case”
- 2 Store and 1 retrieve vs 1 store operation
- Multiple passes to produce data as requirements defined.

Met Office Climate Data Dissemination System (CDDS)

Hadley Centre



Reproducibility

Computing environment

- MO central conda plus additions

Versioning of all aspects

- Source code
- Configuration data

Workflow documentation

- Operational procedure
- Tickets for processing, audit trail

Performance

I/O bound
process

Extracting from archive key bottleneck
Use optimised storage

Inherently
parallel

How to split the processing into tasks?
Year? MIP? Domain? Output
frequency?

Parallel
processing
at what
level?

Workflow management tool – Rose
Python – multiprocessing, dask

Pros and cons

+ Responsive to evolving requirements

+ Scientist approval

- Data Extraction bottleneck

- Scientist bottleneck

- Manual intervention

Beyond CMIP6



Inline and Post-processing



Future CMIP
project
management

Regular
releases?



Create common CMIP/ climate
research workflow



Changing platforms (e.g. cloud)