Science & Technology Facilities Council | UK Research and Innovation | NERC SCIENCE OF THE ENVIRONMENT

Robust and Reliable WPS for climate simulations – nothing clever, just test, test, test!

Ag Stephens (STFC CEDA) – 02/12/2019

IS-ENES3 ESGF Virtual Workshop on Compute and Analytics

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

Experience of WPS – the operations

1. It is easy to put up a processing web service that works with *some* data.

2. Scientists are cautious about using processing services. Why?
   - Transparency
   - Robustness
   - Trust

3. Complex operations are hard to validate.

Experience of WPS – the community

1. Unlike other parts of ESGF (e.g. Data/Index nodes), we have entertained divergent solutions. What is the impact?

2. There is a finite resource for development and maintenance (globally). It would be great if we could converge on a single solution.

Experience of WPS – the code

1. Each codebase starts off looking very tidy.

2. The data (e.g. CMIP5) is *dirty*: in spite of CMOR, it is actually heterogeneous.

3. The codebase tends towards complexity:
   - If model == XYZ: fix the time axis
   - If variable == ABC: fix the units
   - Etc, etc…

Proposed solution

1. Python codebase, build on PyWPS, Birdhouse, Xarray and any existing libraries/tools.

2. Begin with basic operations:
   - Subsetting, Averaging and (some) regridding.

3. Focus on *robustness*, not *functionality*.

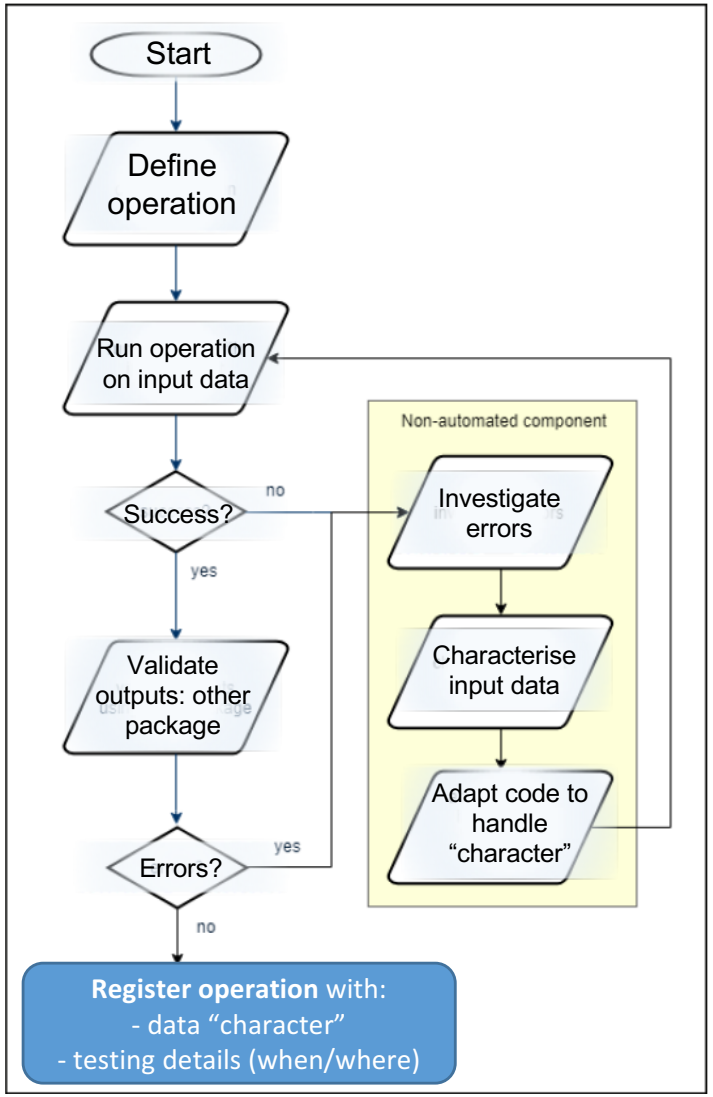
Focus on robustness, how?

For each project/activity (e.g. CMIP5, CORDEX):

- Test *all* operations against a large, representative sample of the available data.
- Validate the outputs in a comprehensive way.
- Most importantly, “characterise” the data sets in a public register.

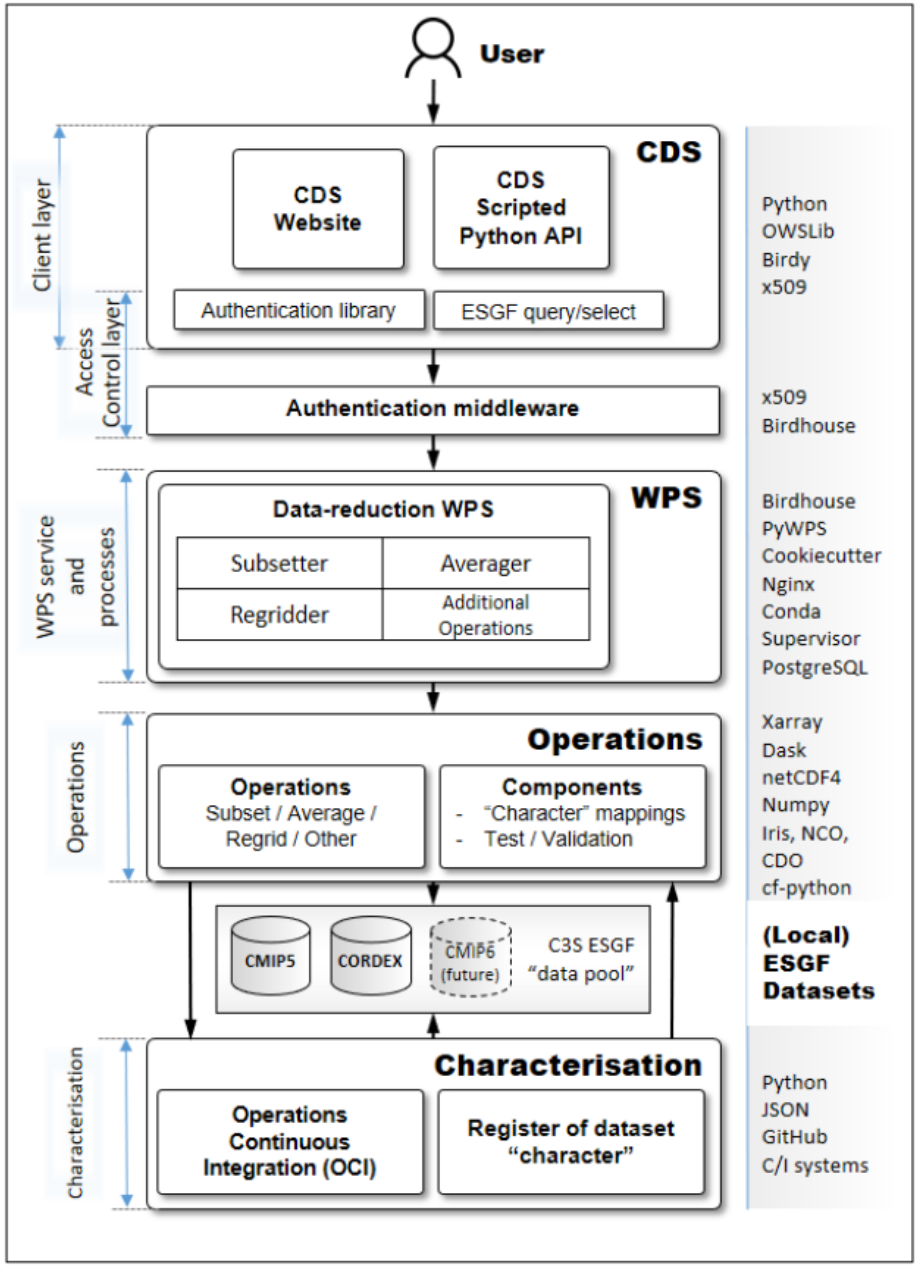
Characterise the data?

- Record the variations in “character” in a public register.
- Code for each character is agnostic/reusable.
- Keeps the code clean.
- Register can be shared across the community (scientists and developers).

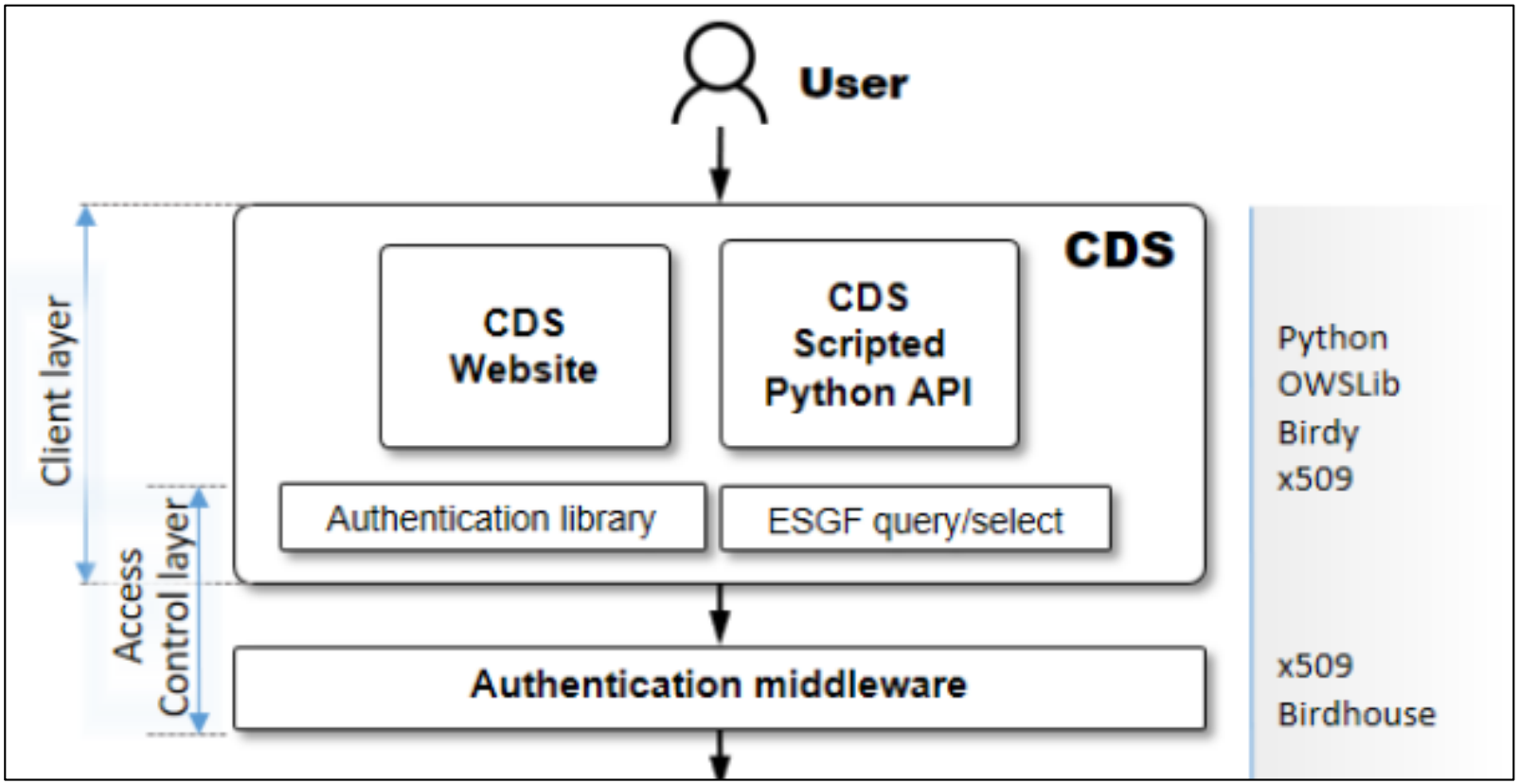


C3S Proposal

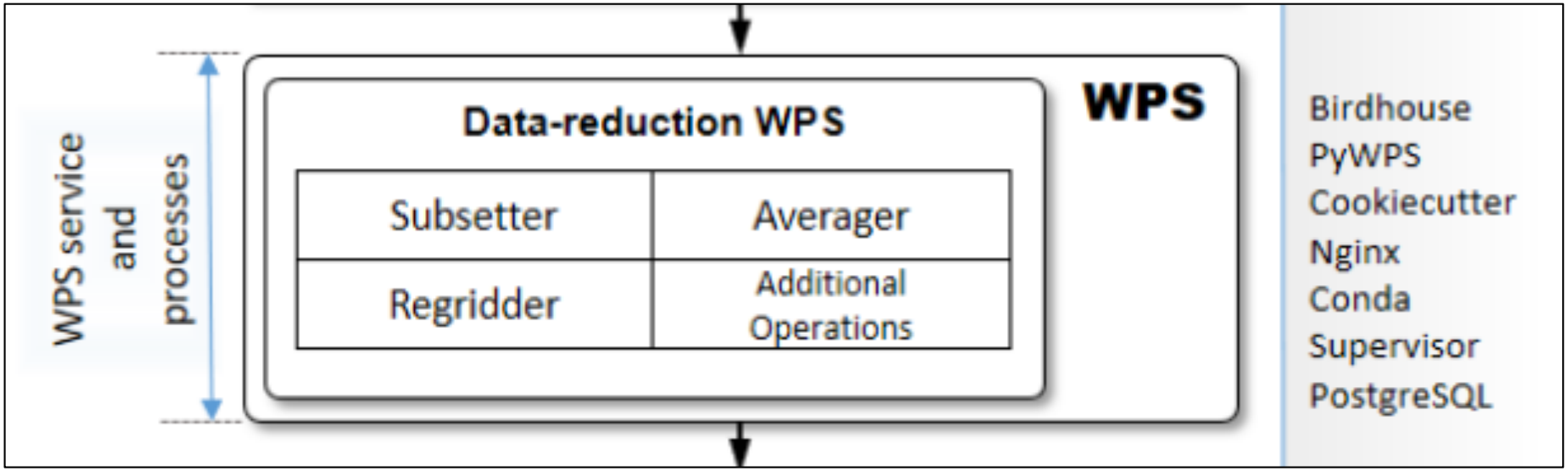
- STFC CEDA / DKRZ
- Input from Ouranos/CRIM (Canada)
- Layered approach



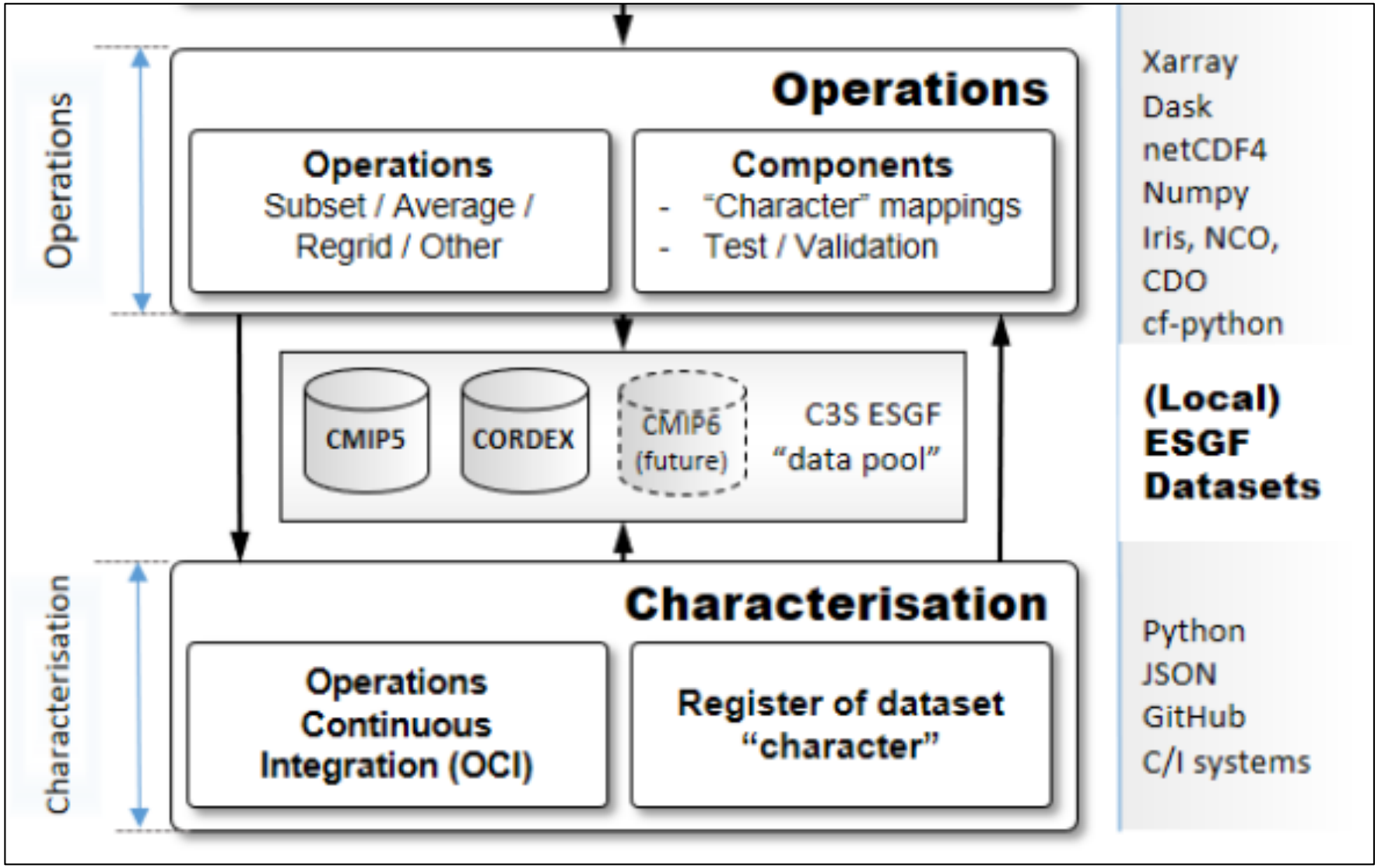
Client Layer



WPS Layer


WPS service and processes
Data-reduction WPS
Subsetter
Averager
Regridder
Additional Operations
WPS
Birdhouse
PyWPS
Cookiecutter
Nginx
Conda
Supervisor
PostgreSQL

Operations and Characterisation Interactions



Next steps

- Project funded (I hope).
- By mid-2020:
  - Design and initial implementation
- By end of 2020:
  - Subsetting and averaging processes deployed
- By mid-2021:
  - Regridding and additional processes deployed

Your thoughts/input?

- Do you disagree with this rationale/approach?
- Would you like to collaborate on any aspects?
- What should the “characterisation” register look like?

Thanks for listening