TERRADUE

Advancing Earth Science

www.terradue.com

EO Application Packages hands-on

Developing Cloud-ready Earth Observation Applications

Developer Workshop, 123th OGC Member Meeting, June 2022

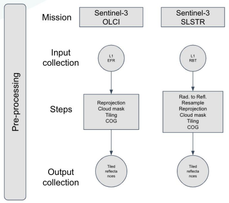
About Us

ellip
Collaborative Earth Science Services
Discover the first collaborative platform in Earth Sciences.

- Terradue is an ESA spin-off based in Rome
  - with a subsidiary in the UK (Oxford)
- Support to application builders in Earth sciences to use satellite data as their information source
  - Algorithms in different data programming models & languages
  - Producing Analysis Ready Data and Added Value Products
- Deploy & operate Earth observation data processing applications in multiple Clouds without lock-in
  - Processed near the EO data holdings
  - Fixed time period or continuous operations

Terradue
Advancing Earth Science

EO Application Package



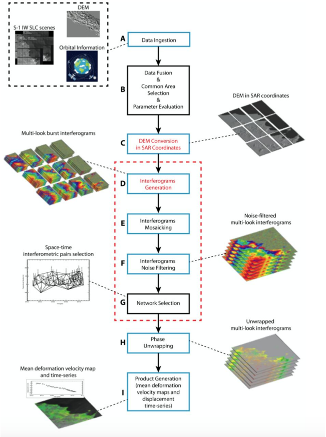
Pre-processing

Mission: Sentinel-3 OLCI | Sentinel-3 SLSTR

Input collection: L1 EFR | L1 RBT

Steps: Reprojection, Cloud mask, Tiling, COG | Rad. to Refl., Resample, Reprojection, Cloud mask, Tiling, COG

Output collection: Tiled reflectances | Tiled reflectances



S-1 IW SLC scenes
DEM
Orbital Information
DEM in SAR coordinates
Multi-look burst interferograms
Noise-filtered multi-look interferograms
Space-time interferometric pairs selection
Unwrapped multi-look interferograms
Mean deformation velocity map and time-series

A Data Ingestion

B Data Fusion & Common Area Selection & Parameter Evaluation

C DEM Conversion in SAR Coordinates

D Interferograms Generation

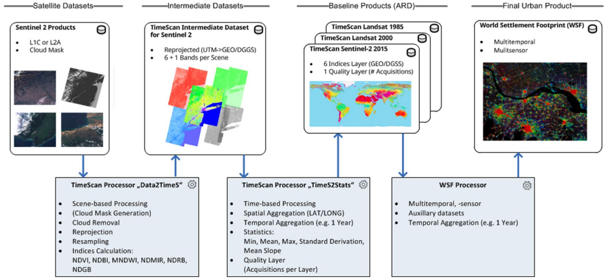
E Interferograms Mosaicking

F Interferograms Noise Filtering

G Network Selection

H Phase Unwrapping

I Product Generation (mean deformation velocity maps and displacement time-series)



Satellite Datasets

Sentinel 2 Products
- L1C or L2A
- Cloud Mask

Intermediate Datasets

TimeScan Intermediate Dataset for Sentinel 2
- Reprojected (UTM->GEO/DGGS)
- 6 + 1 Bands per Scene

Baseline Products (ARD)

TimeScan Landsat 1985

TimeScan Landsat 2000

TimeScan Sentinel-2 2015
- 6 Indices Layer (GEO/DGSS)
- 1 Quality Layer (# Acquisitions)

Final Urban Product

World Settlement Footprint (WSF)
- Multitemporal
- Mulitsensor

TimeScan Processor „Data2TimeS“
- Scene-based Processing
- (Cloud Mask Generation)
- Cloud Removal
- Reprojection
- Resampling
- Indices Calculation: NDVI, NDBI, MNDWI, NDMIR, NDRB, NDGB

TimeScan Processor „TimeS2Stats“
- Time-based Processing
- Spatial Aggregation (LAT/LONG)
- Temporal Aggregation (e.g. 1 Year)
- Statistics: Min, Mean, Max, Standard Derivation, Mean Slope
- Quality Layer (Acquisitions per Layer)

WSF Processor
- Multitemporal, -sensor
- Auxillary datasets
- Temporal Aggregation (e.g. 1 Year)

- An Earth Observation Application is set of command-line tools with numeric, textual and EO data parameters organized as a computational workflow
- An Application Package uses an explicit language that describes the input and output interface of the computational workflow and the orchestration of its command-line tools.
- The Application Package guarantees the automation, scalability, reusability, portability of the Application while also being workflow-engine and vendor neutral.

New OGC Best Practice for EO Application Packages

Open Geospatial Consortium

EO Application Package

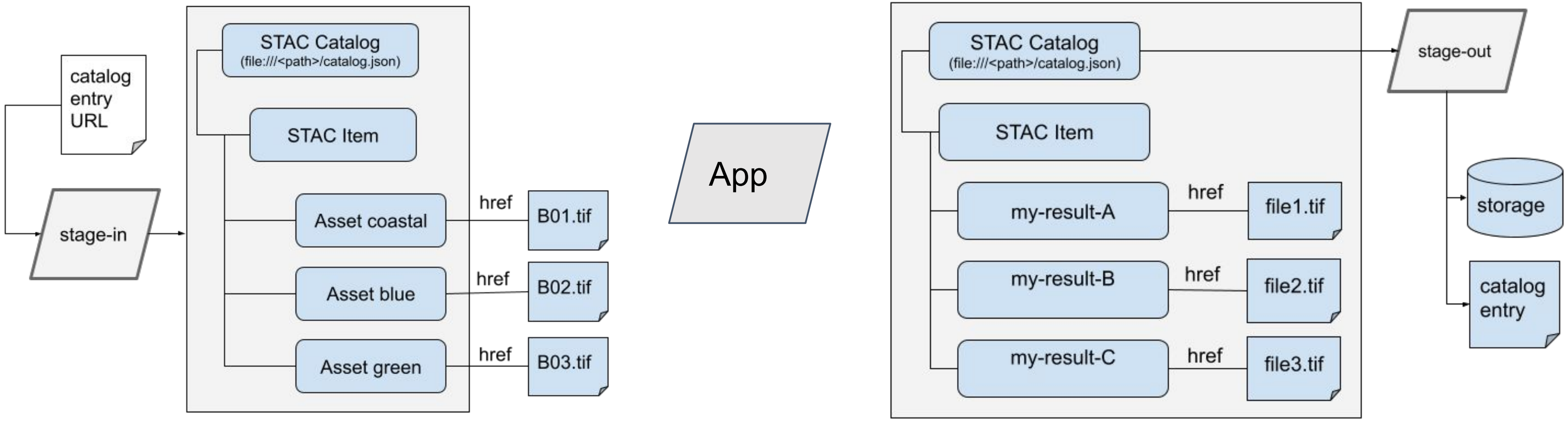
COMMON WORKFLOW LANGUAGE

The Common Workflow Language (CWL) is an ***open standard*** for describing analysis workflows and tools in a way that makes them ***portable*** and ***scalable*** across a variety of ***software and hardware environments***, from workstations to cluster, cloud, and high performance computing environments.

- The command-line tools (e.g. Python, shell script, C++) and their dependencies are containerized and registered in a container registry
- The computational workflow input and output interfaces and the orchestration of its command-line tools are described with Common Workflow Language (CWL)

EO Application Package

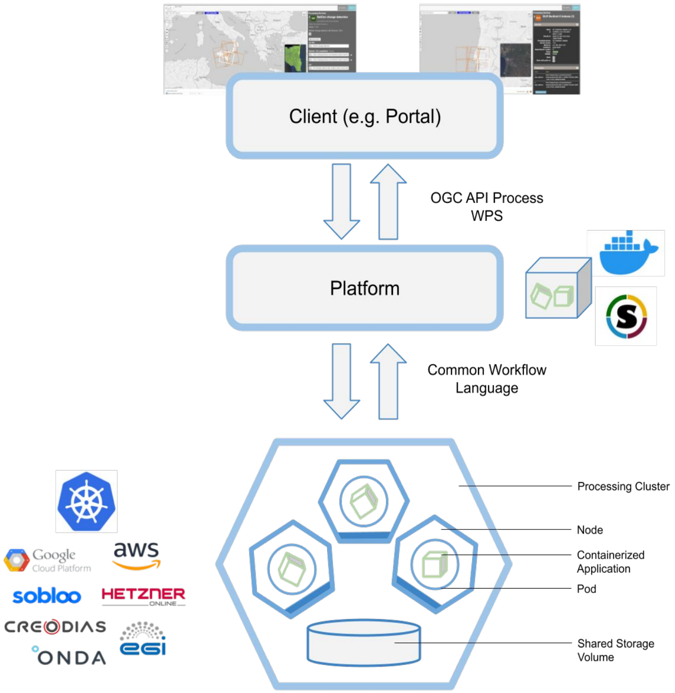
- The computational workflow data interfaces use the Spatio Temporal Asset Catalog (STAC) to describe the EO data inputs and generated results.



Stage-in

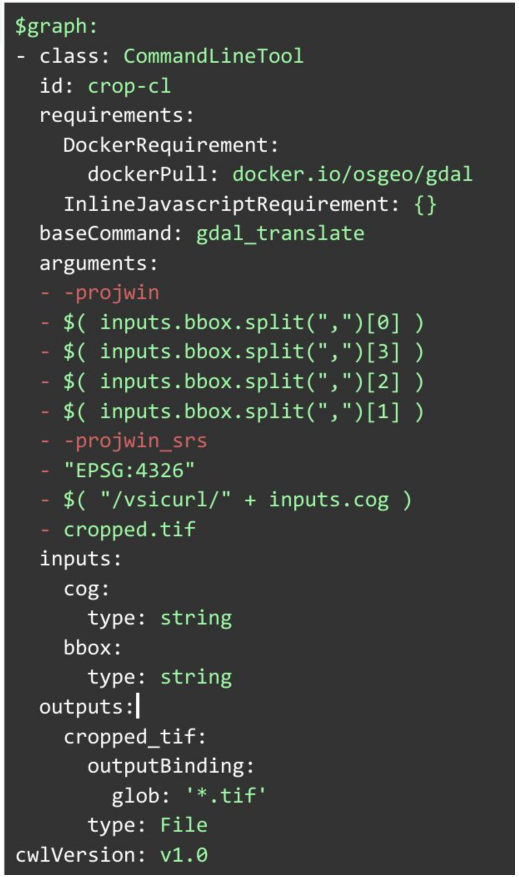
Stage-out

EO Application Package

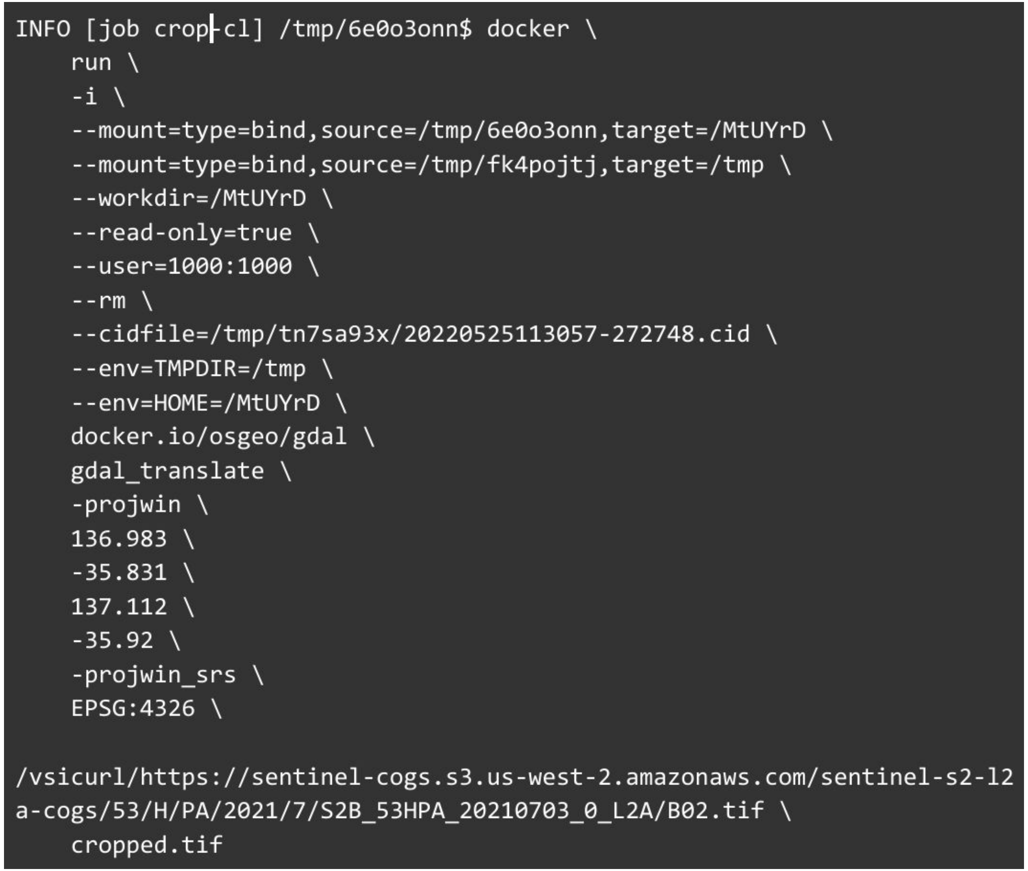


- The Platform takes the CWL application package and exposes an OGC API Processes processing service.
- The Platform provides the automation, scalability, reusability, portability by converting the OGC API Processes execution request into a CWL execution using a runner and the computing resources of the selected provider.

An example with gdal



```
$graph:
- class: CommandLineTool
  id: crop-cl
  requirements:
    DockerRequirement:
      dockerPull: docker.io/osgeo/gdal
    InlineJavascriptRequirement: {}
  baseCommand: gdal_translate
  arguments:
  - -projwin
  - $( inputs.bbox.split(",")[0] )
  - $( inputs.bbox.split(",")[3] )
  - $( inputs.bbox.split(",")[2] )
  - $( inputs.bbox.split(",")[1] )
  - -projwin_srs
  - "EPSG:4326"
  - $( "/vsicurl/" + inputs.cog )
  - cropped.tif
  inputs:
    cog:
      type: string
    bbox:
      type: string
  outputs:
    cropped_tif:
      outputBinding:
        glob: '*.tif'
      type: File
cwlVersion: v1.0
```



```
INFO [job crop-cl] /tmp/6e0o3onn$ docker \
    run \
    -i \
    --mount=type=bind,source=/tmp/6e0o3onn,target=/MtUYrD \
    --mount=type=bind,source=/tmp/fk4pojtj,target=/tmp \
    --workdir=/MtUYrD \
    --read-only=true \
    --user=1000:1000 \
    --rm \
    --cidfile=/tmp/tn7sa93x/20220525113057-272748.cid \
    --env=TMPDIR=/tmp \
    --env=HOME=/MtUYrD \
    docker.io/osgeo/gdal \
    gdal_translate \
    -projwin \
    136.983 \
    -35.831 \
    137.112 \
    -35.92 \
    -projwin_srs \
    EPSG:4326 \

/vsicurl/https://sentinel-cogs.s3.us-west-2.amazonaws.com/sentinel-s2-l2
a-cogs/53/H/PA/2021/7/S2B_53HPA_20210703_0_L2A/B02.tif \
    cropped.tif
```

TERRADUE
Advancing Earth Science

Skills and tooling

Skills

- YAML
- Containers (docker files, docker build, tags, etc.)

Tooling

- A container engine: docker or podman
- A CWL runner: cwltool, calrissian (k8s)
- An IDE: VS Code or Theia/Coder (in the Cloud)
- An object storage (S3)
- Access to a container registry (e.g. docker.io, quay.io Gitlab, Github)
- Access to Continuous Integration service (e.g. Gitlab CI, Github Actions, Jenkins, etc.)
- Access to a Package Registry (e.g. Gitlab, Github, Artifactory)

Hands-on on Binder

https://github.com/Terradue/ogc-eo-application-package-hands-on

Additional resources:

- CWL specification https://www.commonwl.org/v1.2/
- OGC Best Practice for Earth Observation Application Package

  https://docs.ogc.org/bp/20-089r1.html
- https://cwl-for-eo.github.io/guide/ (work in progress)
- CWL runners:
  - https://github.com/common-workflow-language/cwltool
  - https://github.com/Duke-GCB/calrissian
- From zero to CWL on kubernetes https://github.com/Terradue/calrissian-session

TERRADUE

Looking forward hearing from you!

https://www.terradue.com

Fabrice Brito

fabrice.brito@terradue.com