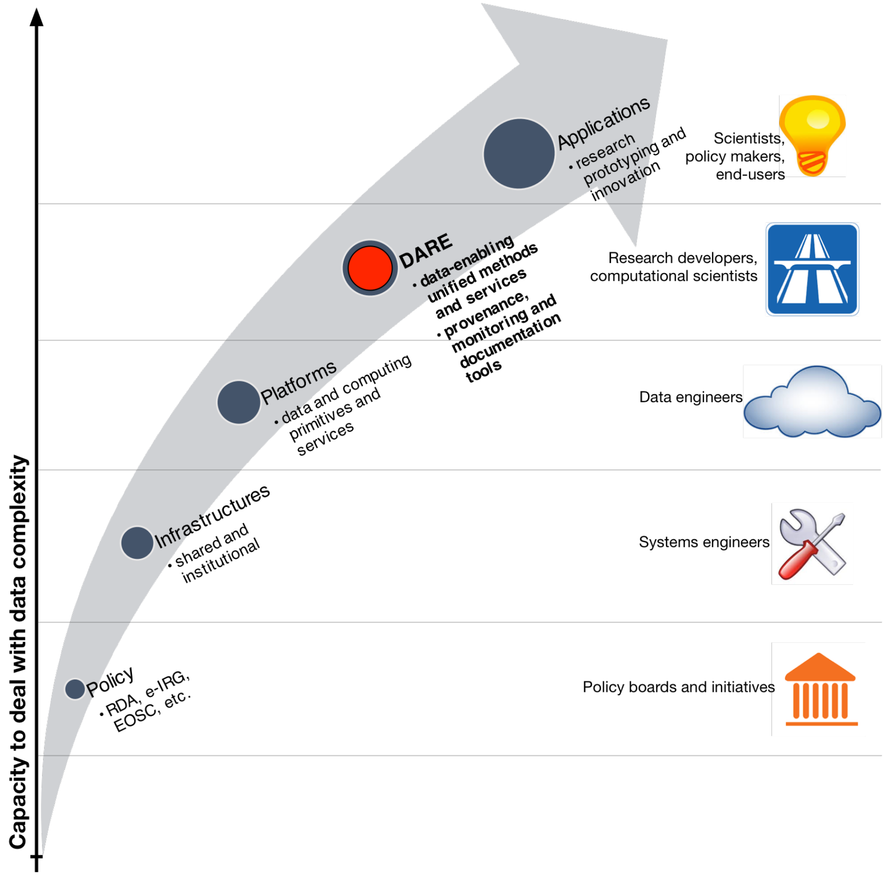
Delivering Agile Research Excellence on European E-Infrastructures

DARE

Integrating solutions for Data-Intensive and Reproducible Science

Alessandro Spinuso (KNMI) & The DARE Team

DARE
Delivering Agile Research Excellence on European e-Infrastructures



Working environment for professionals wrestling with challenges involving complexity of methods and data

Domain Expert - Computational Scientists - Research Developers

- **Mapping from abstract methods to concrete** applications executed by different enactments seamlessly
- **Validation and Traceability of runs and products: diagnose, monitor, repeat**
- **Organisation of campaigns** reusing data and methods from **multiple runs**
-

Accelerate productivity of expert teams.

Requirements Collected from EPOS/IS-ENES

DARE

EPOS
EUROPEANPLATEOBSERVINGSYSTEM

is-enes
INFRASTRUCTURE FOR THE EUROPEAN NETWORK
FOR EARTH SYSTEM MODELLING

Requirements Collected from EPOS/IS-ENES

- **Control over** heterogeneous applications (HPC simulation, data processing).
- **Develop new atomic functions adopting community libraries.**
- **Access to distributed data sources** to gather input data products.

Requirements Collected from EPOS/IS-ENES

- **Control over** heterogeneous applications (HPC simulation, data processing).
- **Develop new atomic functions adopting community libraries.**
- **Access to distributed data sources** to gather input data products.

- **Abstract** workflow pipelines from implementation.
- **Process large data volumes,** near(er) to the data storage.
- **Exploit Research infrastructures and e-infrastructures services (ESGF, EOSC).**

Requirements Collected from EPOS/IS-ENES

- **Control over** heterogeneous applications (HPC simulation, data processing).
- **Develop new atomic functions adopting community libraries.**
- **Access to distributed data sources** to gather input data products.

- **Find, Repeat, Reuse and Trace execution's outcomes.**
- **Integration within existing community portals, web-services.**
- **Automatic allocation of resources for large computations.**

- **Abstract** workflow pipelines from implementation.
- **Process large data volumes,** near(er) to the data storage.
- **Exploit Research infrastructures and e-infrastructures services (ESGF, EOSC).**

DARE Unified Actions and Remote APIs

User Facing Tools

Integration in Development environments

jupyter

Invoked on demand

climate4impact

VERCE

OGC

Open Geospatial Consortium, Inc.

DARE Unified Actions and Remote APIs

User Facing Tools

Integration in Development environments

jupyter

Invoked on demand

climate4impact

VERCE

OGC

Open Geospatial Consortium, Inc.

Unified Actions

- Manage Working Sessions
- Validate, Repeat, Reuse Results
- Configure Run
- Execute Applications and Workflows
- Compose Register Workflows

DARE Unified Actions and Remote APIs

User Facing Tools

Integration in Development environments

jupyter

Invoked on demand

climate4impact

VERCE

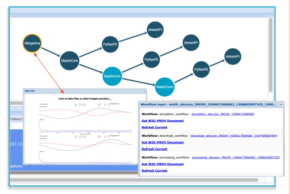
OGC® Open Geospatial Consortium, Inc.

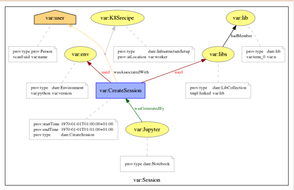
Unified Actions

- Manage Working Sessions
- Validate, Repeat, Reuse Results
- Configure Run
- Execute Applications and Workflows
- Compose Register Workflows

Services (Provenance, Execution and Catalogues)

Provenance Services

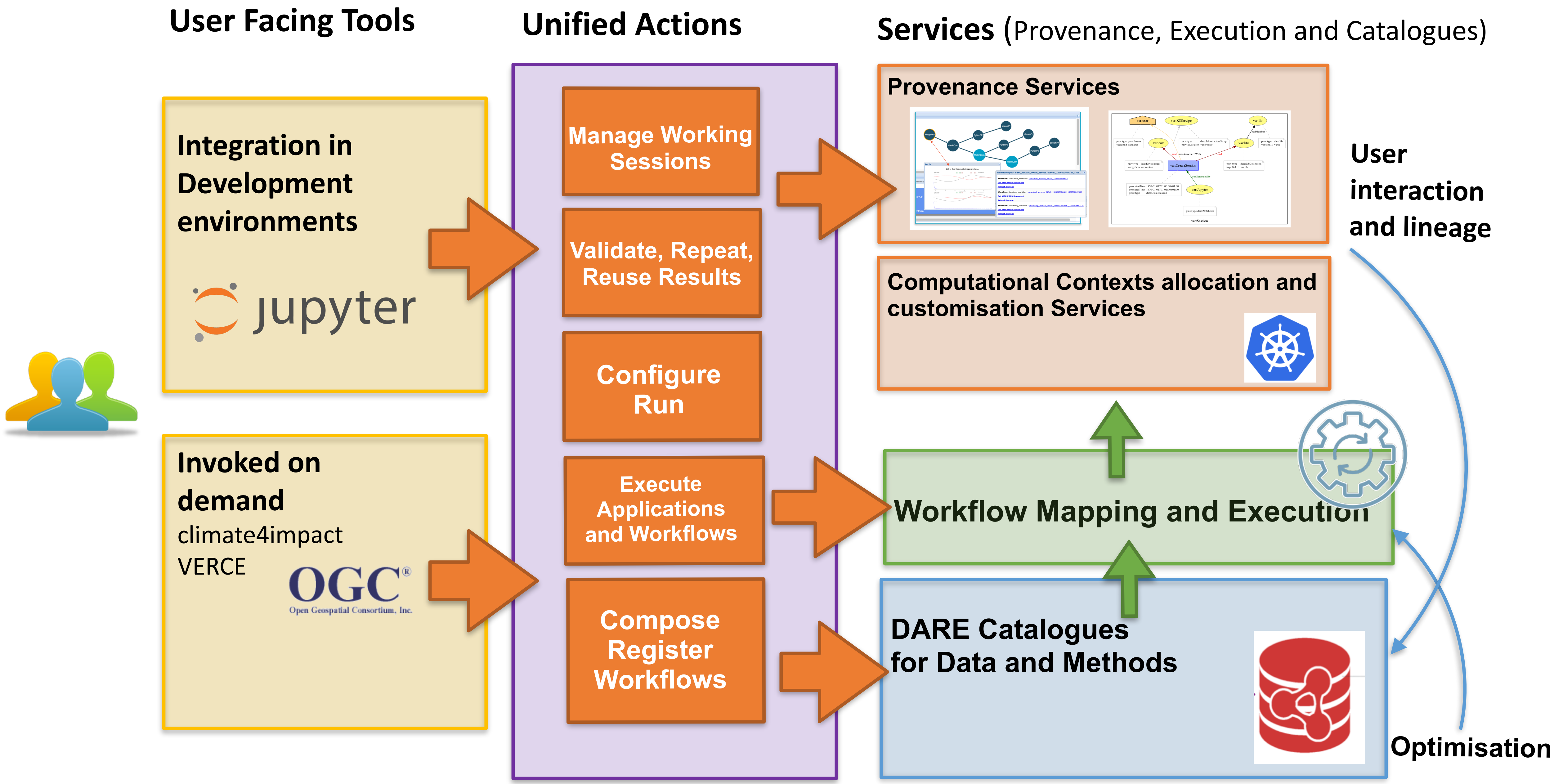




Computational Contexts allocation and customisation Services

Workflow Mapping and Execution

DARE Unified Actions and Remote APIs



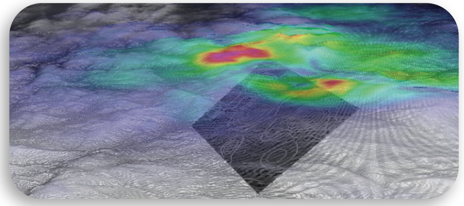
Develop, Describe and Run Workflows

Test Case: Rapid Assessment Workflow Decomposition

Rapid Ground Motion Assessment (RA)

Reusable Tasks running at different scale.

May require human monitoring and intervention



Choose/upload **seismic wavespeed** & **mesh**

Choose/upload **seismic source** (point or fault)

run waveform simulation

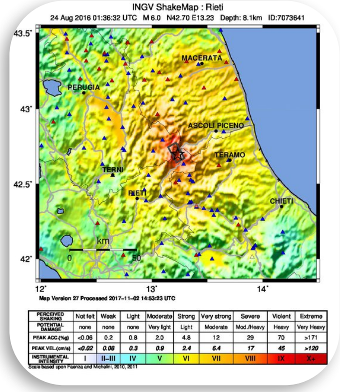
Gather **observed data**

Get pre-processed synt and data

Get ground motion parameters

Compare/integrate synthetic and observed ground motion data

Store data, metadata, provenance



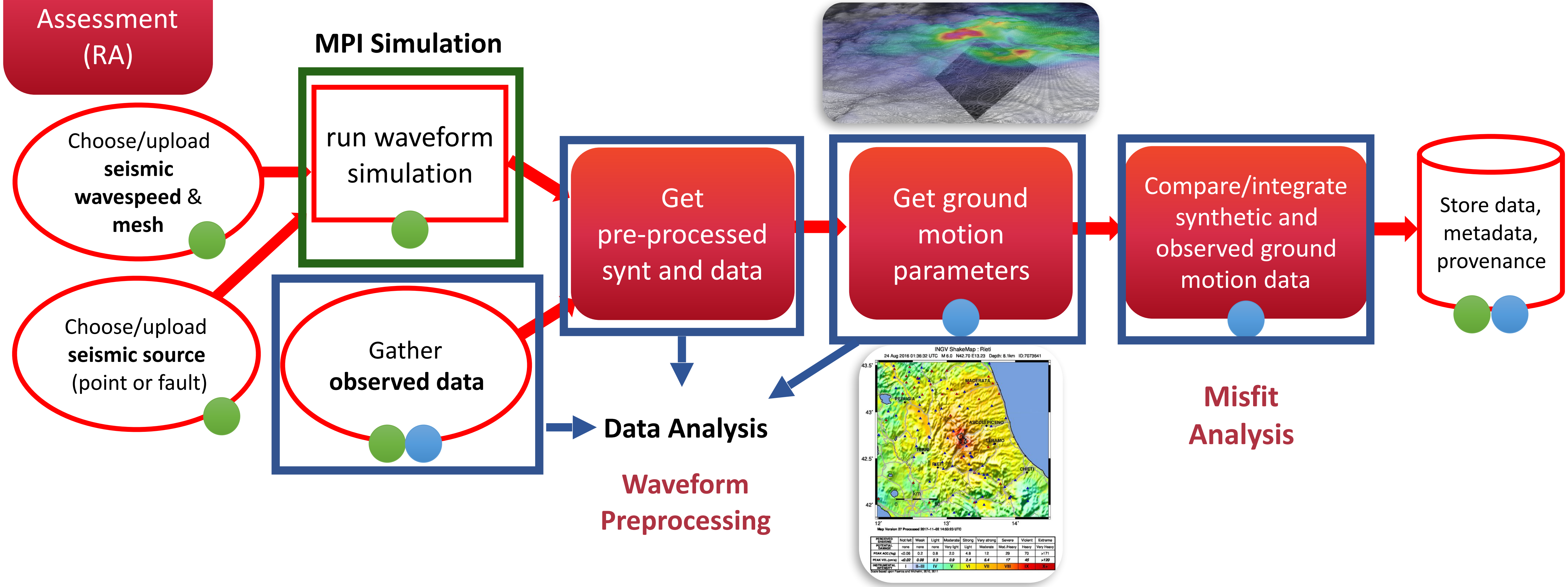
Develop, Describe and Run Workflows

Test Case: Rapid Assessment Workflow Decomposition

Rapid Ground Motion Assessment (RA)

Reusable Tasks running at different scale.

May require human monitoring and intervention

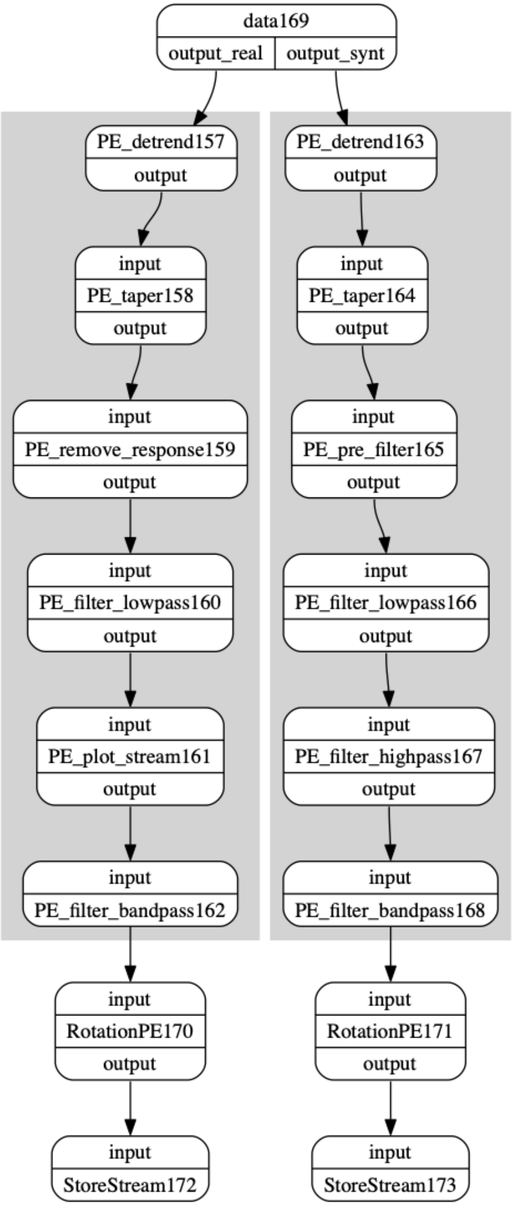


Workflow development

Waveform Preprocessing

pipeline JSON Description (eg. from file)

Manual Extensions

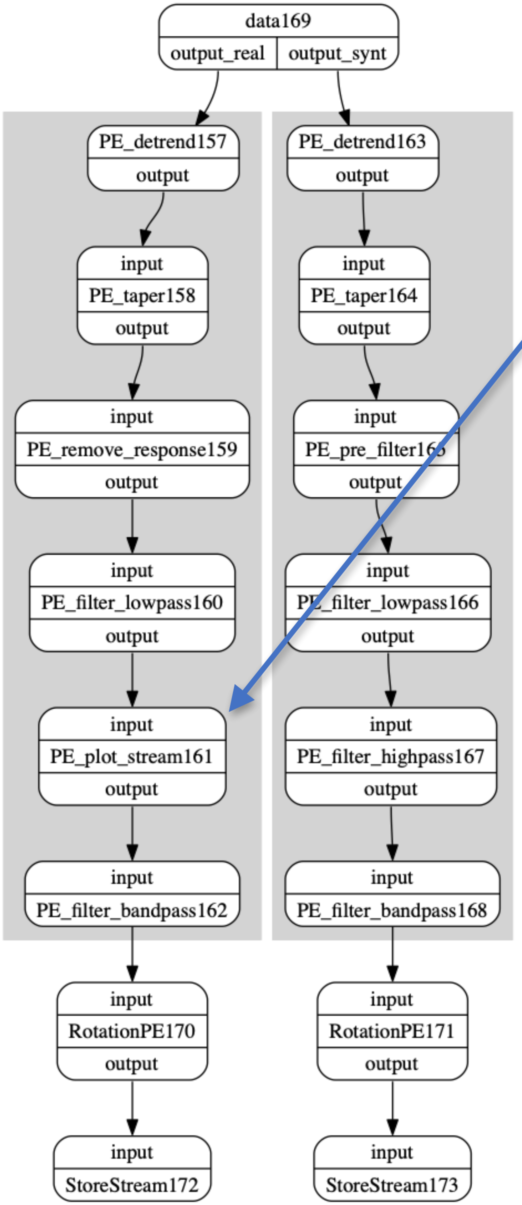


Workflow encoded in Python

```python
def buildWorkflow():
    real_preprocess = create_processing_chain(proc['data_processing'])
    synt_preprocess = create_processing_chain(proc['synthetics_processing'])
    print(real_preprocess)
    graph = WorkflowGraph()
    read = ReadDataPE()
    read.name = 'data'
    read.output_units = proc['output_units']
    rotate_real = RotationPE('data')
    rotate_synt = RotationPE('synth')
    store_real = StoreStream('data')
    store_synt = StoreStream('synth')
    graph.connect(read, 'output_real', real_preprocess, 'input')
    graph.connect(read, 'output_synt', synt_preprocess, 'input')
    if proc['rotate_to_ZRT']:
        graph.connect(real_preprocess, 'output', rotate_real, 'input')
        graph.connect(synt_preprocess, 'output', rotate_synt, 'input')
        graph.connect(rotate_real, 'output', store_real, 'input')
        graph.connect(rotate_synt, 'output', store_synt, 'input')
    else:
        graph.connect(real_preprocess, 'output', store_real, 'input')
        graph.connect(synt_preprocess, 'output', store_synt, 'input')

    return graph


graph=buildWorkflow()

from dispel4py.visualisation import display
display(graph)
```

Provenance Configuration for lineage usability

Waveform Preprocessing



pipeline JSON Description (eg. from file)

Manual Extensions

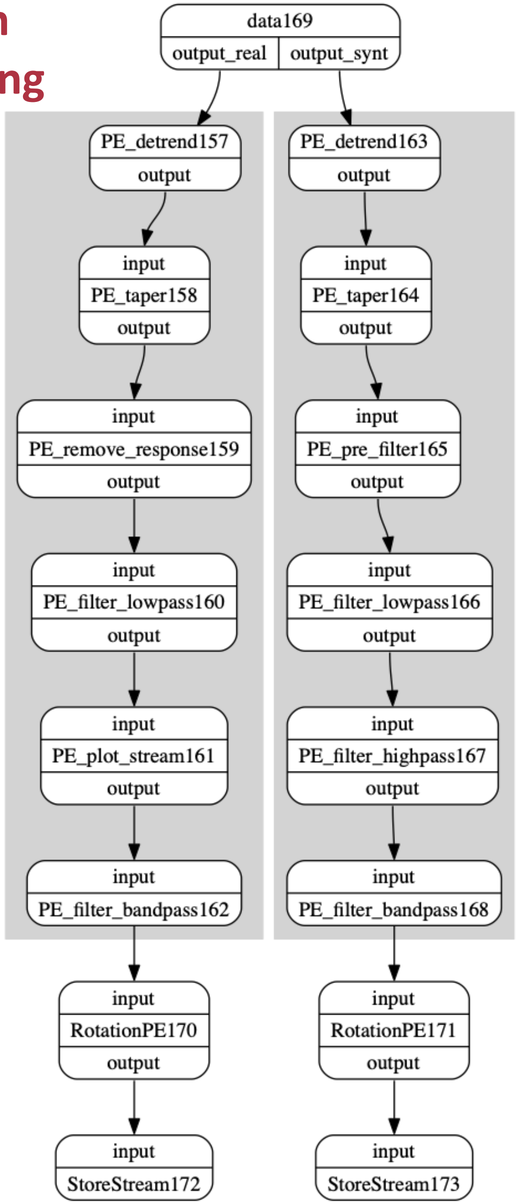
Functions encoded in Python with User Defined Metadata

```
def plot_stream(stream, output_dir, source, tag):
    stats = stream[0].stats
    filename = source + "-%s.%s.%s.%s.png" % (
        stats['network'], stats['station'], stats['channel'], tag)
    path = os.environ['STAGED_DATA'] + '/' + output_dir

    if not os.path.exists(path):
        try:
            os.makedirs(path)
        except:
            pass

    dest = os.path.join(path, filename)
    stream.plot(outfile=dest)
    #return stream
    prov = {'location': "file://" + socket.gethostname() + "/" + dest,
            'format': 'image/png',
            'metadata': {'myterm': tag}}
    return {'_d4p_prov': prov, '_d4p_data': stream}
```

User Defined Metadata

Provenance Configuration for lineage usability

Waveform Preprocessing

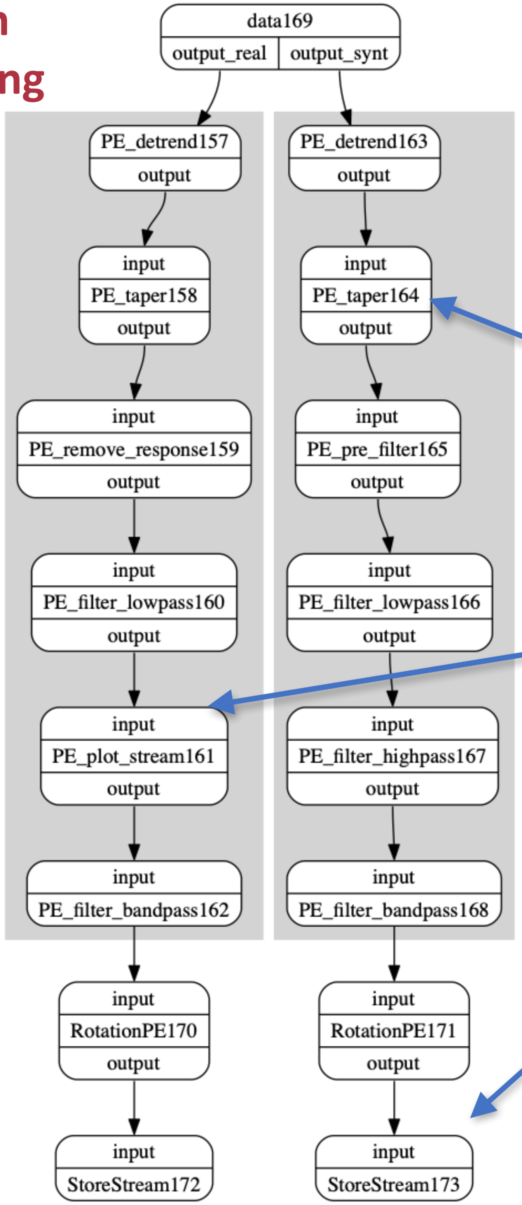


Configuration Profile in JSON with Provenance Types

```
{
        'provone:User': "aspinuso",
        's-prov:description' : "provdemo demokritos",
        's-prov:workflowName': "demo_epos",
        's-prov:workflowType': "seis:preprocess",
        's-prov:workflowId'  : "workflow process",
        's-prov:save-mode'   : 'service'         ,
 # defines the Provenance Types and Provenance Clusters for the Workflow Components
        's-prov:componentsType' :
                        {'PE_taper': {'s-prov:type':(SeismoPE,),
                                      's-prov:prov-cluster':'seis:Processor'},
                         'PE_plot_stream':    {'s-prov:prov-cluster':'seis:Visualisation',
                                               's-prov:type':(SeismoPE,)},
                         'StoreStream':    {'s-prov:prov-cluster':'seis:DataHandler',
                                            's-prov:type':(SeismoPE,)}
                         }}
```

Provenance Configuration for lineage usability

Waveform Preprocessing



Configuration Profile in JSON with Provenance Types

```
{
      'provone:User': "aspinuso",
      's-prov:description' : "provdemo demokritos",
      's-prov:workflowName': "demo_epos",
      's-prov:workflowType': "seis:preprocess",
      's-prov:workflowId'  : "workflow process",
      's-prov:save-mode'   : 'service'         ,
 # defines the Provenance Types and Provenance Clusters for the Workflow Components
      's-prov:componentsType' :
                        {'PE_taper': {'s-prov:type':(SeismoPE,),
                                      's-prov:prov-cluster':'seis:Processor'},
                         'PE_plot_stream':    {'s-prov:prov-cluster':'seis:Visualisation',
                                               's-prov:type':(SeismoPE,)},
                         'StoreStream':    {'s-prov:prov-cluster':'seis:DataHandler',
                                               's-prov:type':(SeismoPE,)}
                         }}
```

starttime: 2013-02-16T21:16:09.240000Z

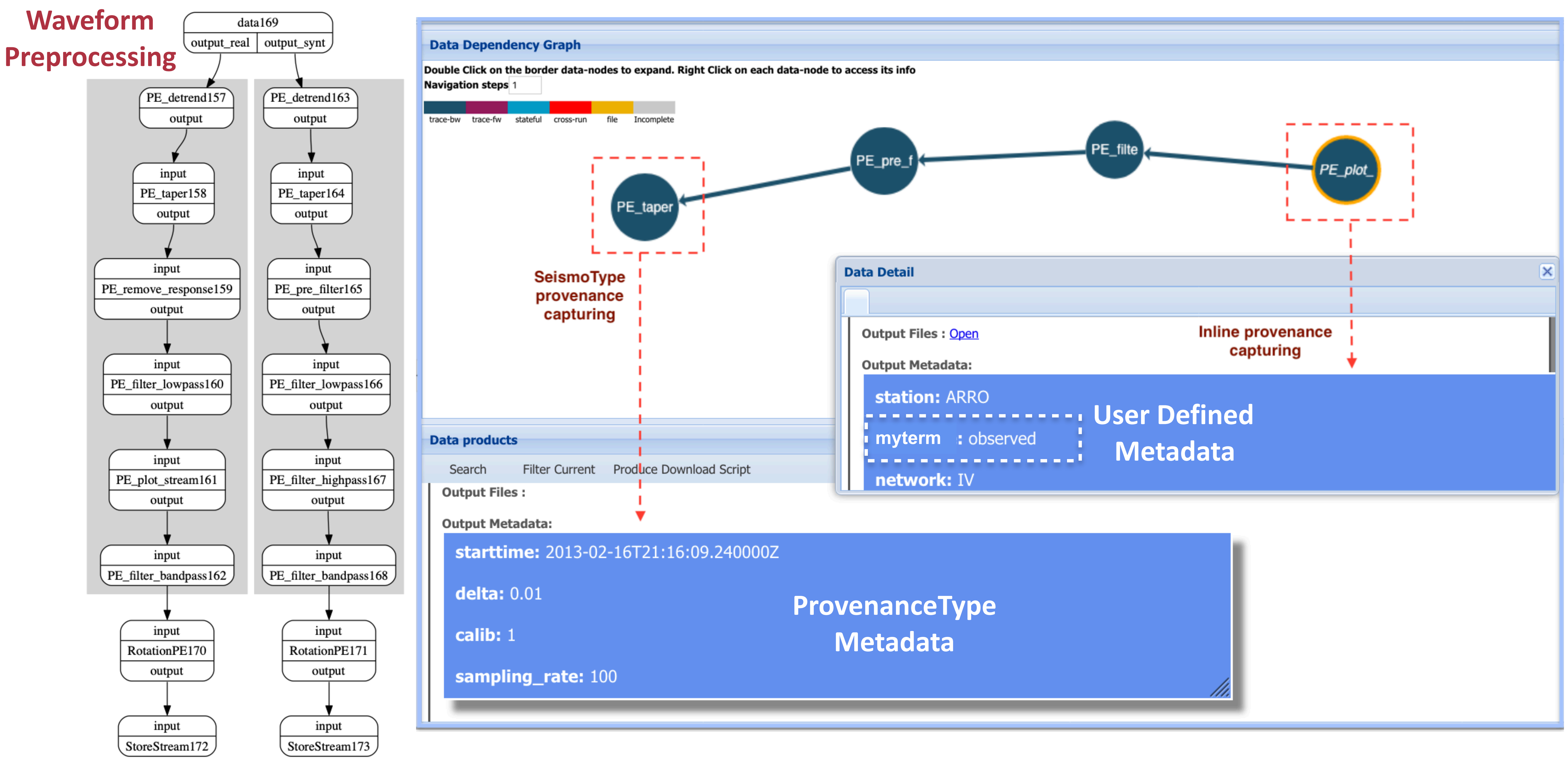
delta: 0.01

calib: 1

sampling_rate: 100

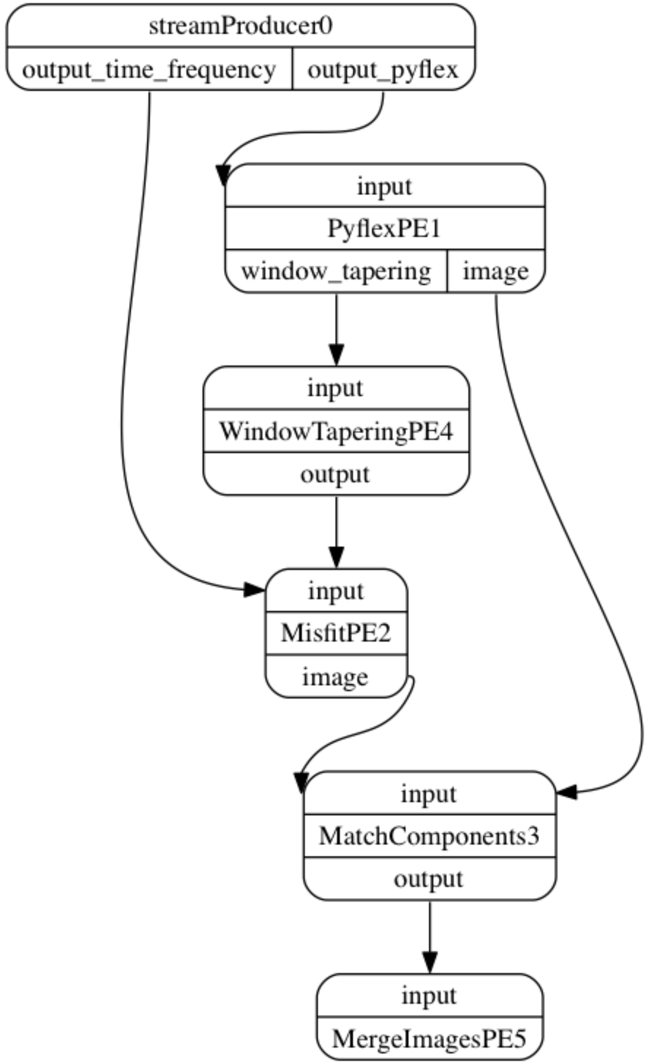
ProvenanceType **for outputs' Metadata Contextualisation and Lineage Patterns**

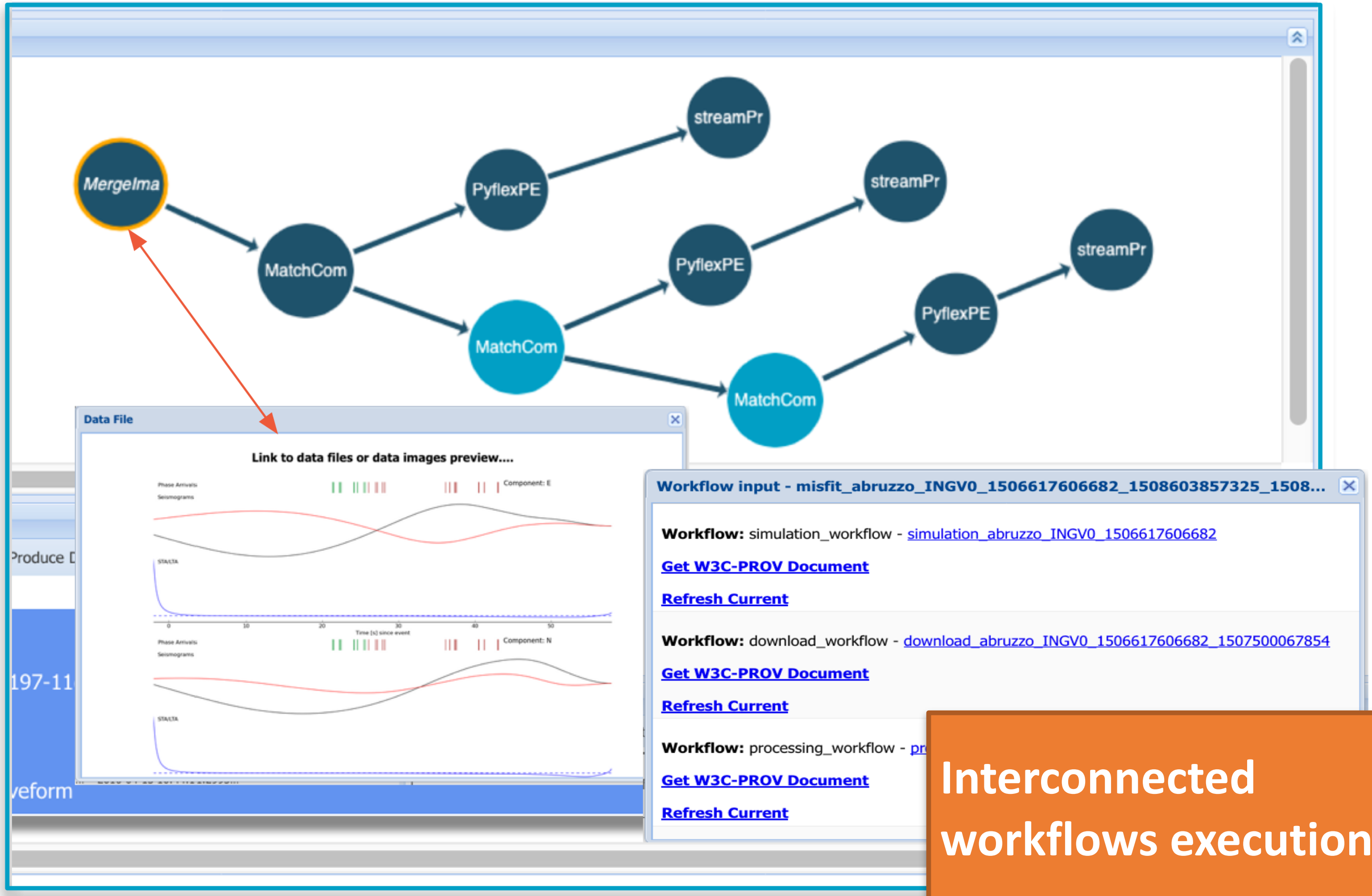
Monitor, search and analyse results through lineage



Monitor, search and analyse results through lineage

Misfit Analysis

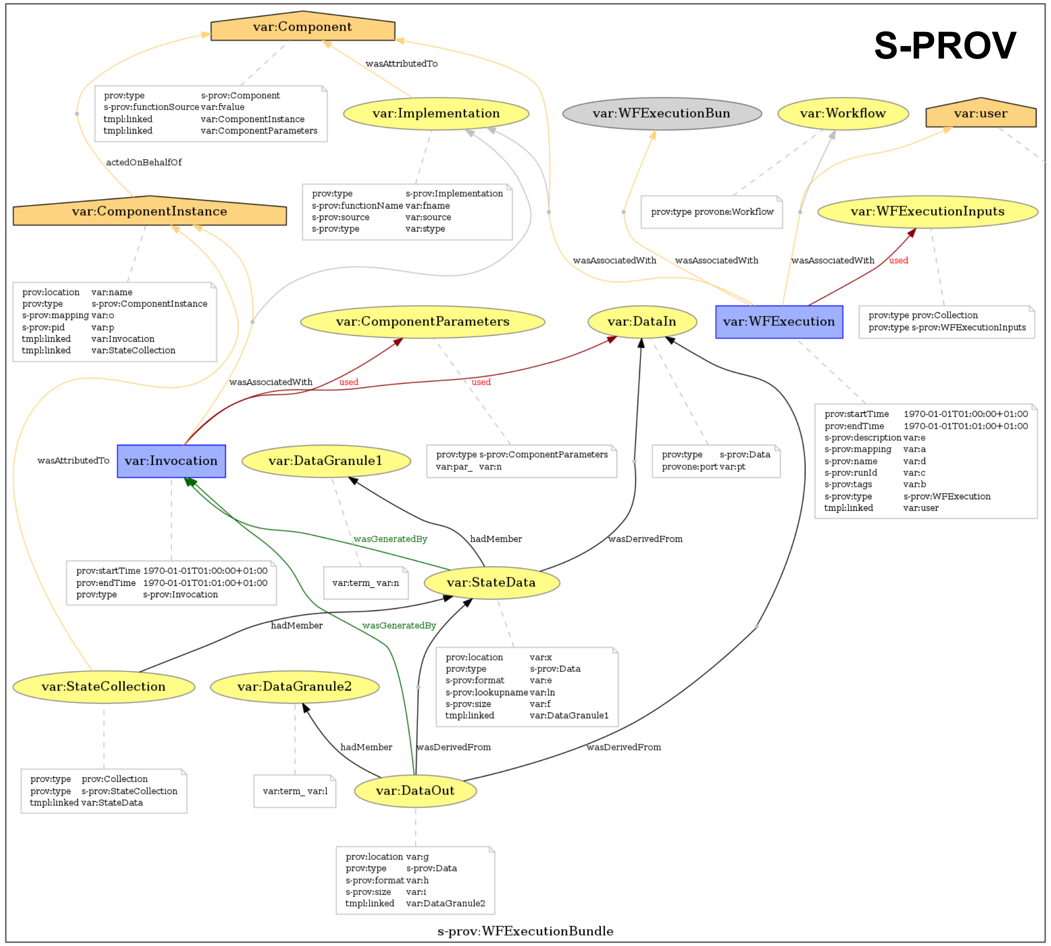




Interconnected workflows executions

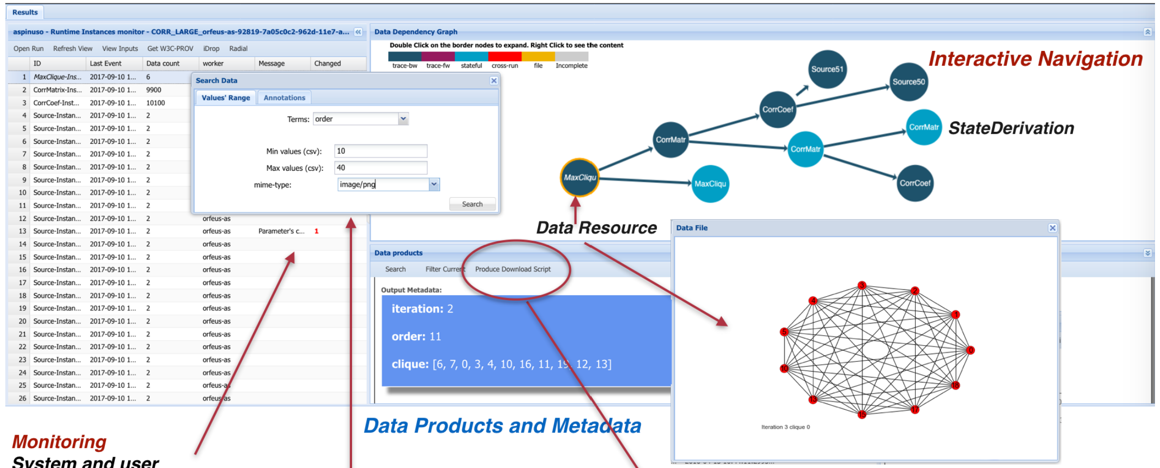
Large Scale Lineage Representation, Management, Exploitation

Provenance Model, Services and Tools combining system, domain and user-driven information about the computation



Processes Runtime Monitor

Data Dependency Graph



Monitoring
System and user messages, changes, data counter, worker

Data Products and Metadata

Discovery and Filter
Search current run for data or filter the current resultset on ancestors' properties

Preview and Staging
- *Access to concrete data resources for preview or download*
- *Produce download script for the whole resultset. Allow large transfers with delegated authorisation.*

S-ProvFlow

A. Spinuso: Active Provenance for Data-Intensive research: https://www.era.lib.ed.ac.uk/handle/1842/33181
S-ProvFlow: https://github.com/KNMI/s-provenance

S-ProvFlow Lineage API

lineage

- GET /data
- POST /data/filterOnAncestor
- GET /data/{data_id}
- GET /data/{data_id}/derivedData
- GET /data/{data_id}/wasDerivedFrom
- GET /instances/{instid}
- GET /invocations/{invocid}
- GET /terms

export

- GET /data/{data_id}/export
- GET /workflowexecutions/{run_id}/export

summaries

- GET /summaries/collaborative
- GET /summaries/workflowexecution

discovery

- GET /workflowexecutions
- DELETE /workflowexecutions/{runid}
- GET /workflowexecutions/{runid}

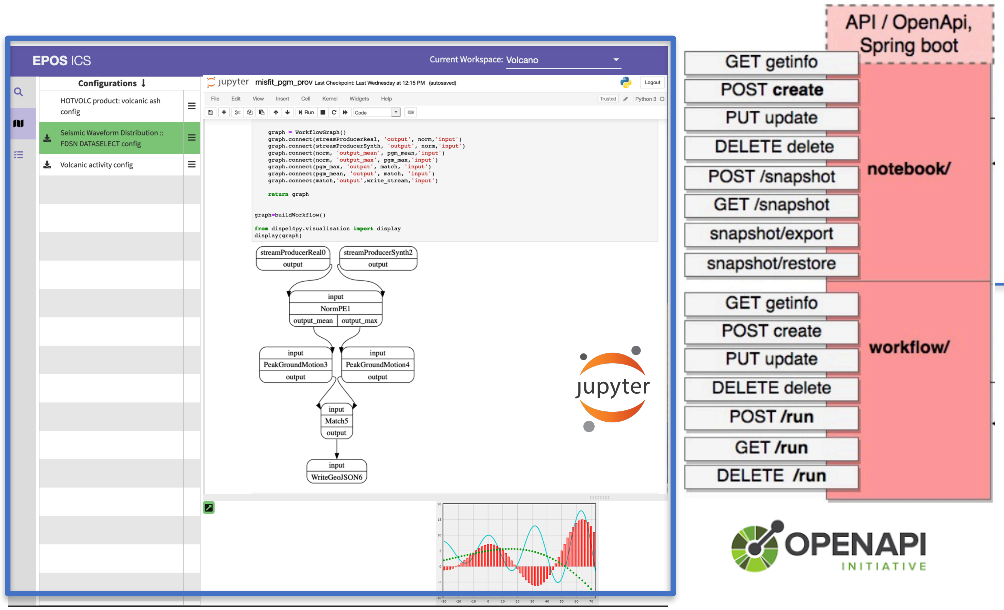
acquisition

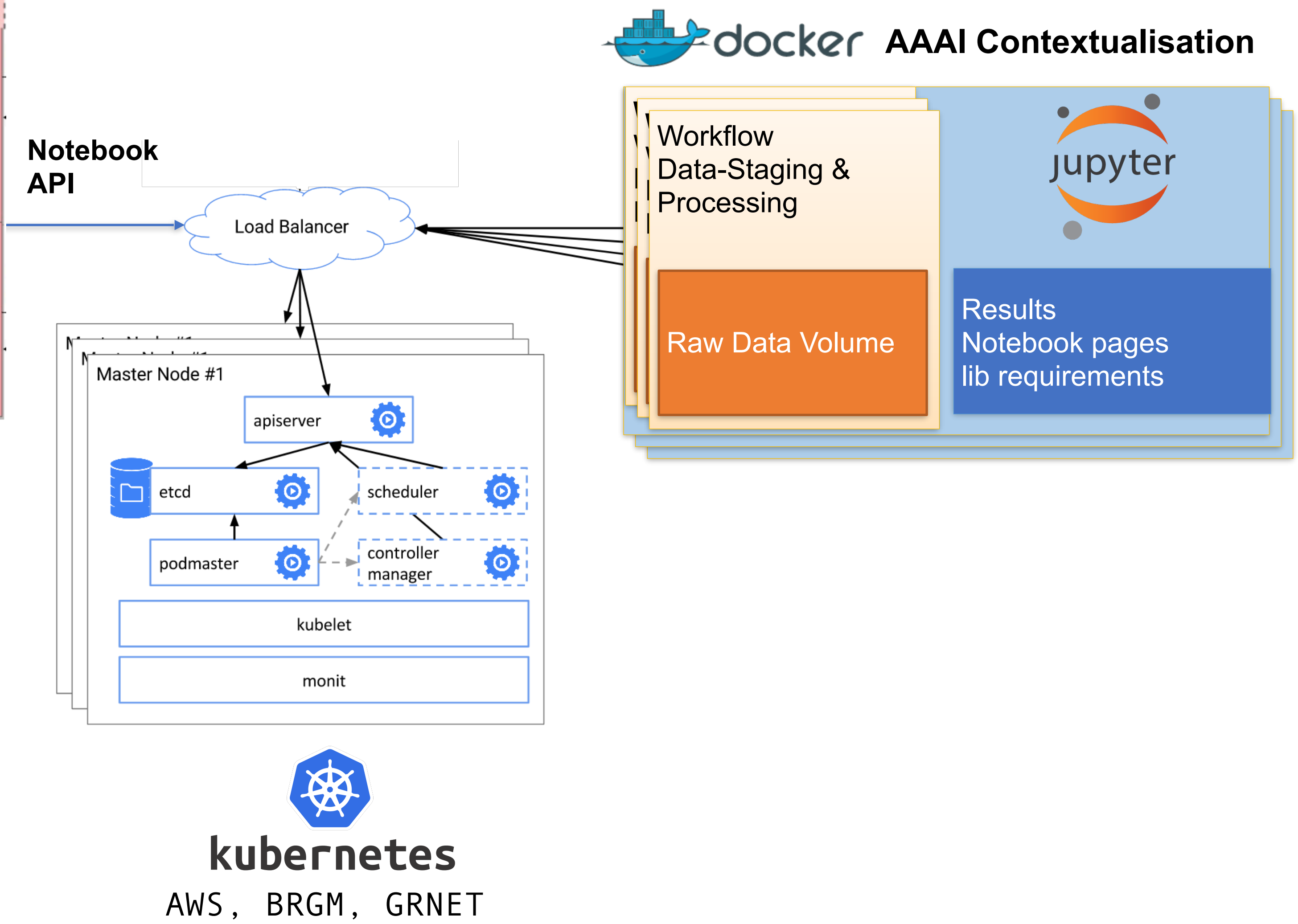
- POST /workflowexecutions/insert
- POST /workflowexecutions/{runid}/delete
- POST /workflowexecutions/{runid}/edit

monitor

- GET /workflowexecutions/{runid}/showactivity

Working Sessions as Notebooks Instances

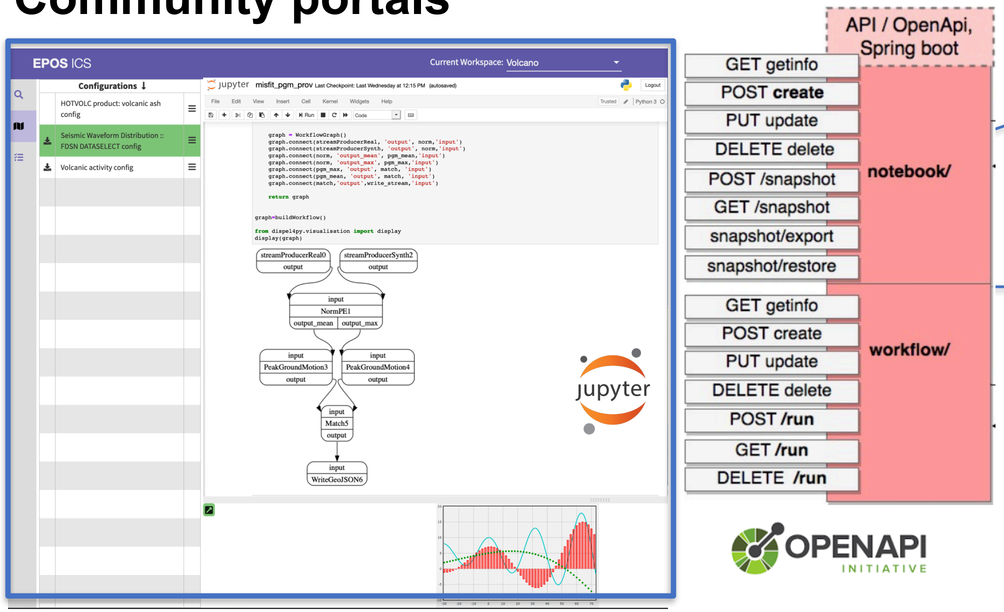


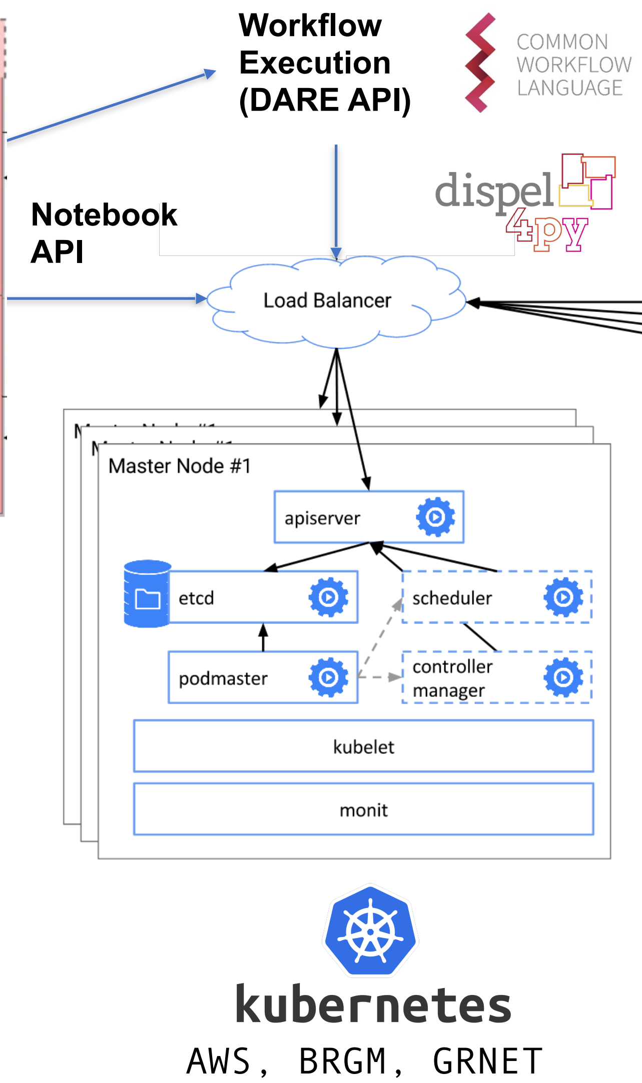


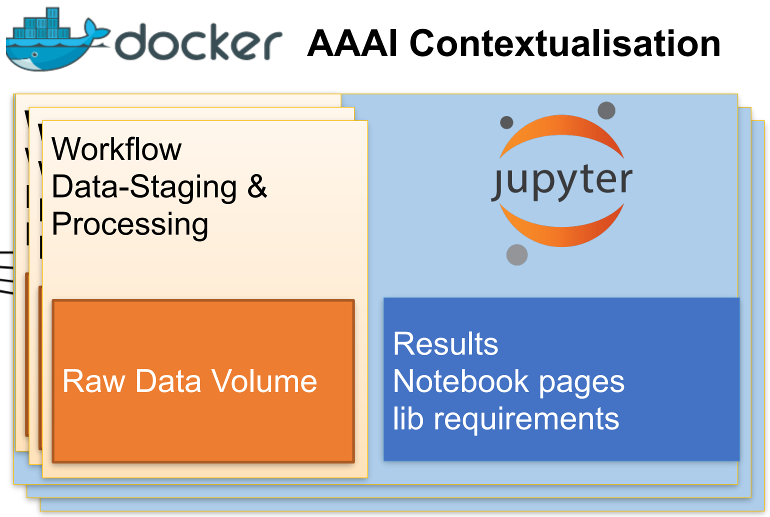
- **Web API** to facilitate **integration and flexible deployment (data proximity)**
  - **Staging and Pre-processing (CWL)**
    - Data staging history
  - Automated resources preparation
  - Local Execution
  - **Traceable Updates of Working Sessions**

Working Sessions as Notebooks Instances

Community portals







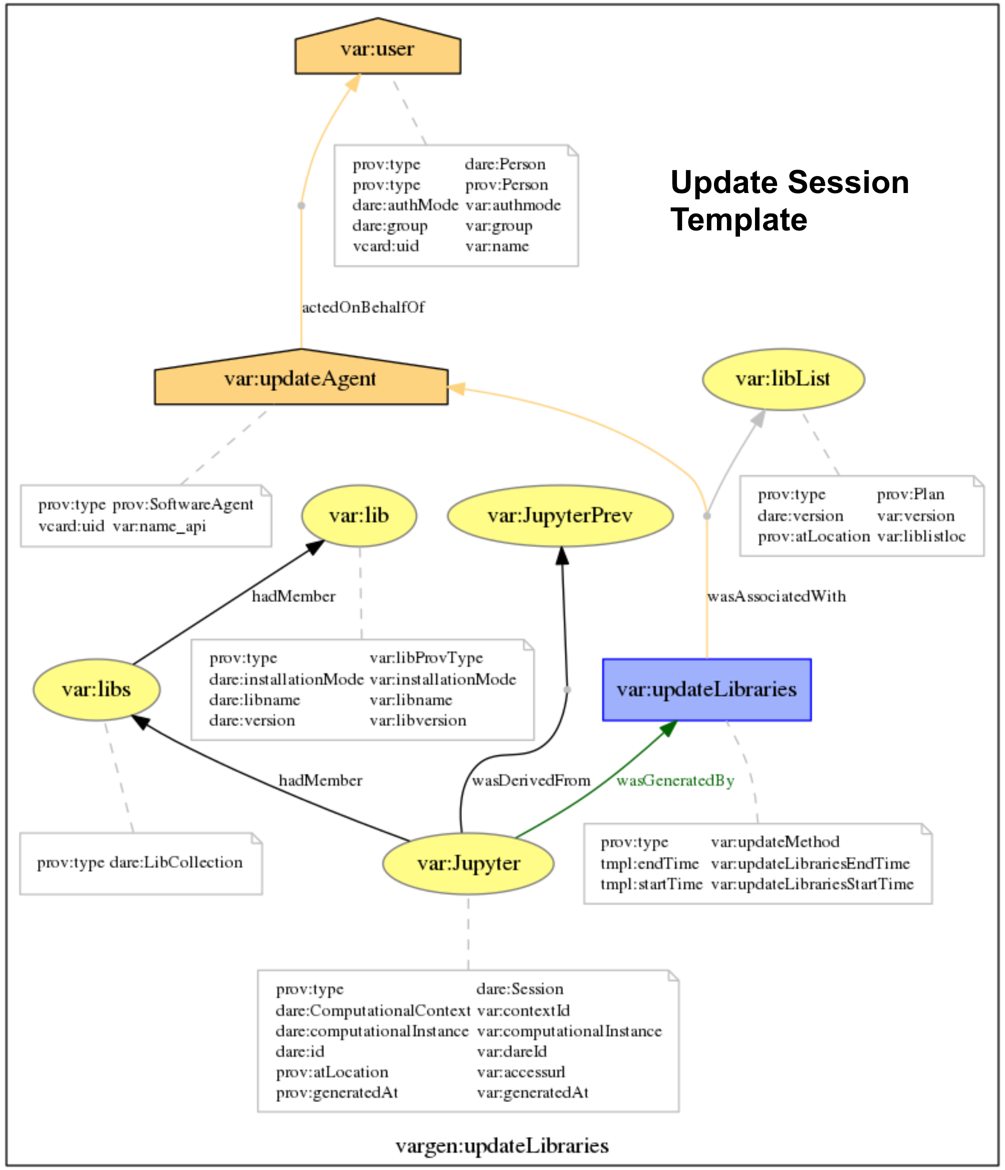
- **Web API** to facilitate **integration and flexible deployment (data proximity)**
  - **Staging and Pre-processing (CWL)**
    - Data staging history
  - Automated resources preparation
  - Local Execution
  - **Traceable Updates of Working Sessions**

-Large workflows execution on containerised resources

-Combine different workflow languages (**CWL**, **dispel4py**)

-Capture and analyse lineage

Trace Notebook Creation and Updates



PROV Templates

Model provenance of the **creation** and interactive **update** of a **Working Sessions**

- Distinction between user's choices and system's concerns
- Reproduce technical setups (.. also onto computational nodes)

Why PROV Templates (as a service)

- **Templates foster discussions** on provenance relationships involving heterogeneous agents and resources.
- **Modelling of usable** and **re-usable** provenance scenarios (tailoring vs generalisation)
- **Remove the burden** to **hardcode provenance editing** (expansion tools/services)
- **Bindings in applications costs less** than building graphs (weight-shift to the service)

Luc Moreau et al. A Templating System to Generate Provenance
https://eprints.soton.ac.uk/405025/1/provtemplate.pdf
ProvenaceTemplate Catalogue
https://github.com/EnvriPlus-PROV/ProvTemplateCatalog
https://envriplus-provenance.test.fedcloud.eu/

Thanks!