Koninklijk Nederlands
Meteorologisch Instituut
*Ministerie van Infrastructuur en Milieu*

DARE http://project-dare.eu

EPOS EUROPEANPLATEOBSERVINGSYSTEM https://www.epos-ip.org

# Integrating solutions for Data-Intensive and Reproducible Science


Alessandro Spinuso @ RDWD
**R&D Observation and Data Technologies**

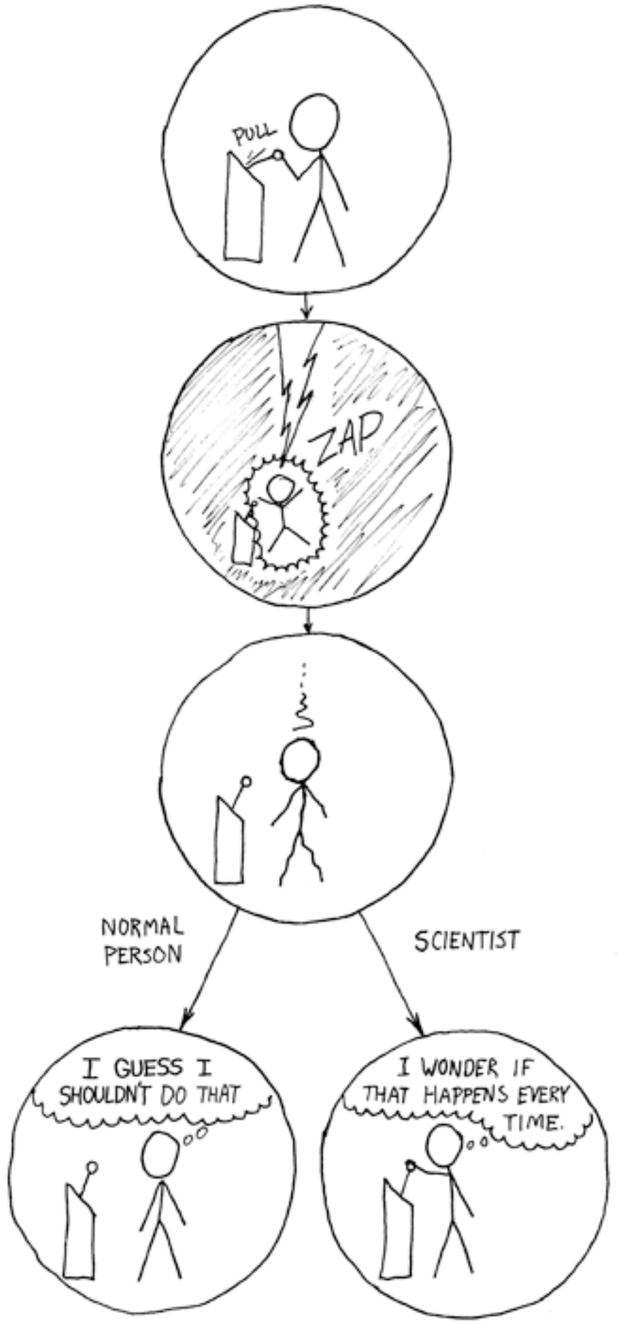
spinuso@knmi.nl


Jupyter Meetup 13 June 2019

# What’s in this talk…

- **Reproducible Science and Research Cycles**
- **Notebooks Working Session and Use Cases**
- **Notebook Service and Provenance Recordings**
- **Workflow Execution Service and Lineage Management**

# Reproducible Science ?

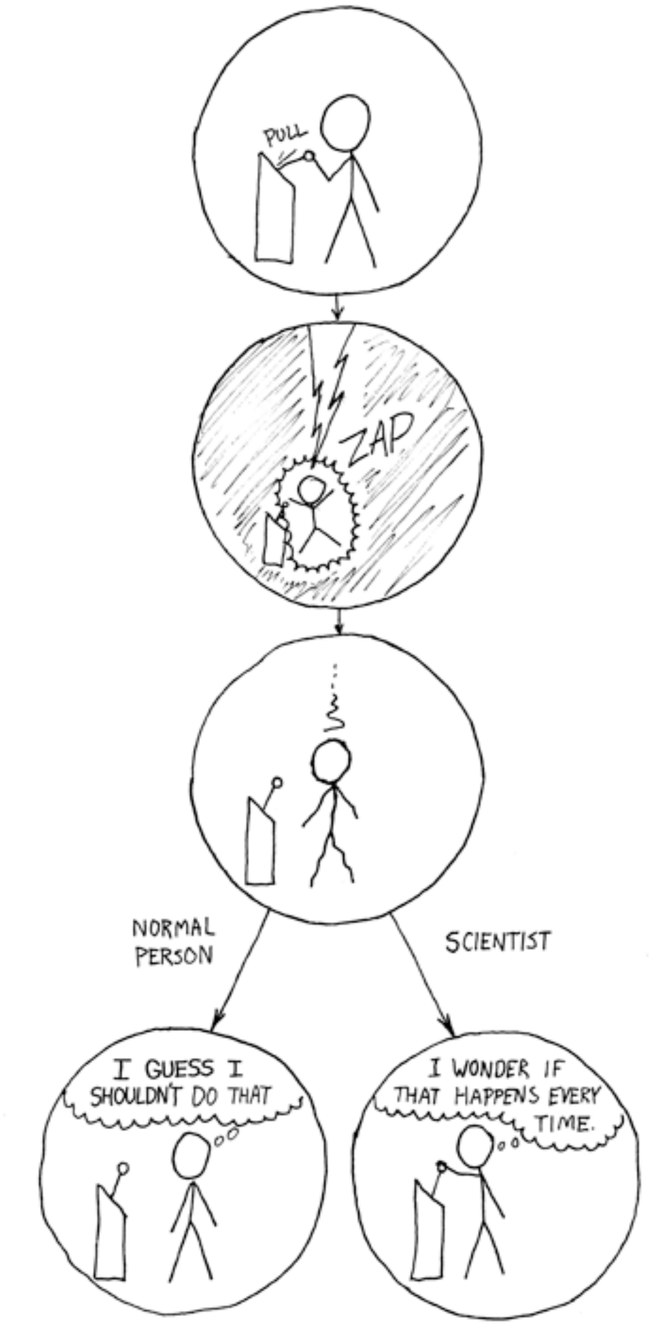


# *Reproducible Science ?*

## What does this suggest?

Scientists aren't normal..

Reproducibility is masochism..



# *Reproducible Science ?*

**What does this suggest?**

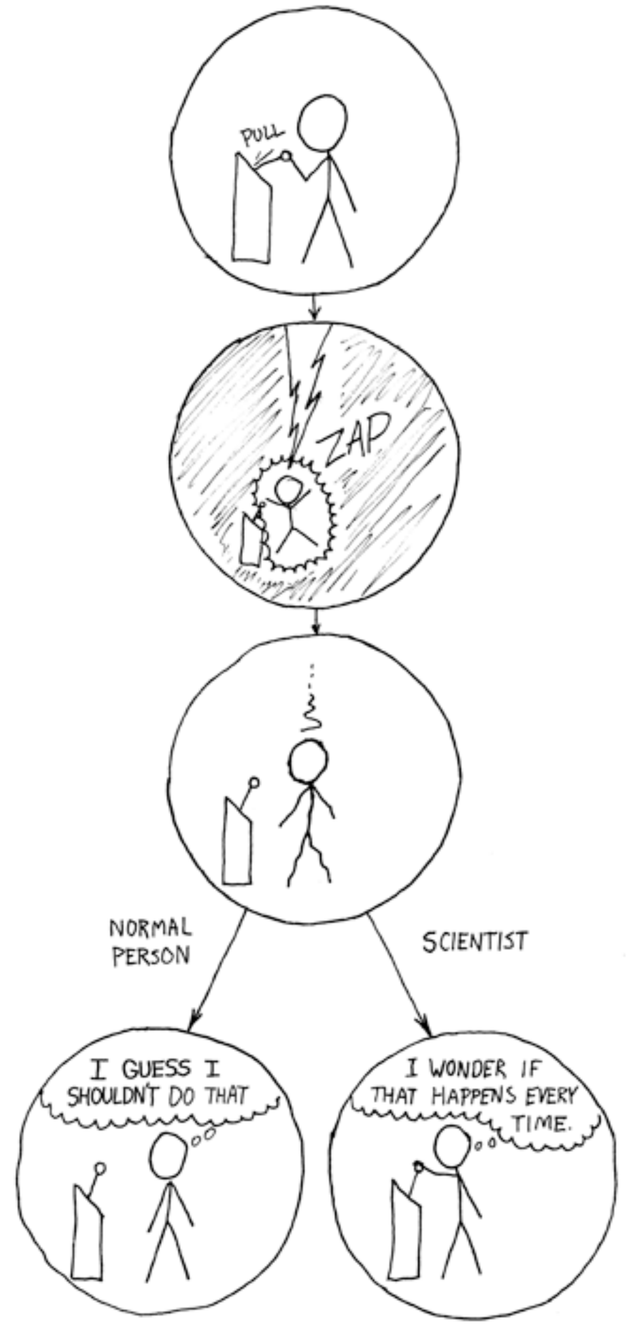
Scientists aren't normal..

Reproducibility is masochism..

**Reproducibility is fundamental but difficult to achieve.**

**Not always convenient/ possible..**

**We need more than just rerun**



# ANY PROBLEM WITH THE SCIENTIFIC METHOD??

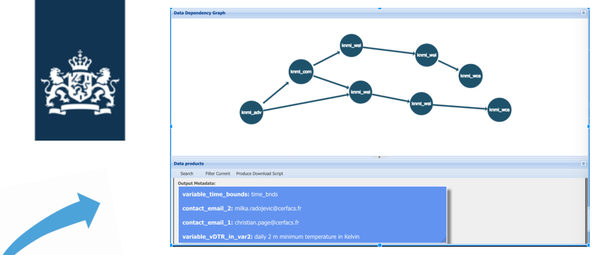

25 May 2016

- Questionnaire on reproducibility filled by 1500 scientists
- > 70% of researchers have tried and failed to reproduce another scientist's experiments
- > 50% have failed to reproduce their own experiments
  - Chemistry: 90% (60%)
  - Biology: 80% (60%)
  - Physics and engineering: 70% (50%)
  - Medicine: 70% (60%)
  - Earth and environment science: 60% (40%)

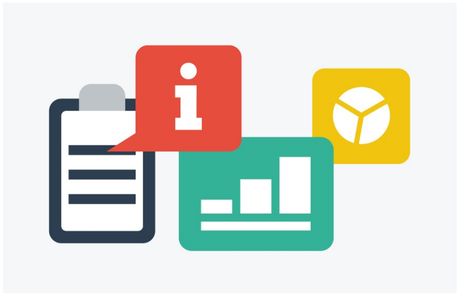
SKA-Link Project

# The Research Cycle(s)

- **Long-running research campaigns** conducted by groups of researchers
- **A variety of tools and working environments** involving different expertise
- **Execution of Multiple experiments** with many stages
- **Incremental maturity** of methods and definitions of properties and metadata

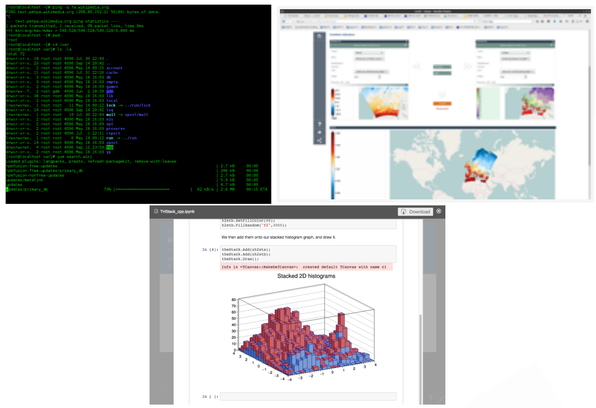



*Validate / Monitor / Explore*



*Report / Outreach / Curate*

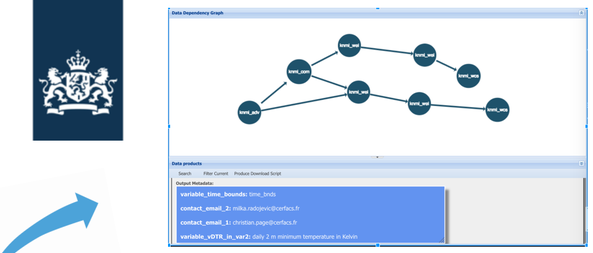
*Repeat / Verify*



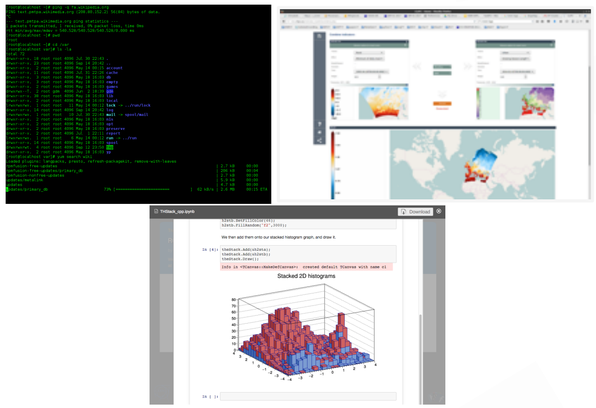

*Programming / Workflows / VREs*

# The Research Cycle(s)

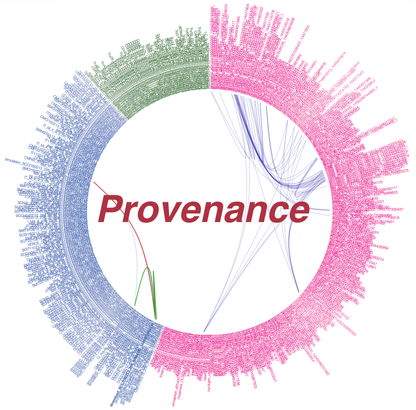
- **Long-running research campaigns** conducted by groups of researchers
- **A variety of tools and working environments** involving different expertise
- **Execution of Multiple experiments** with many stages
- **Incremental maturity** of methods and definitions of properties and metadata



*Validate / Monitor / Explore*



*Programming / Workflows / VREs*




*Report / Outreach / Curate*

*Repeat / Verify*

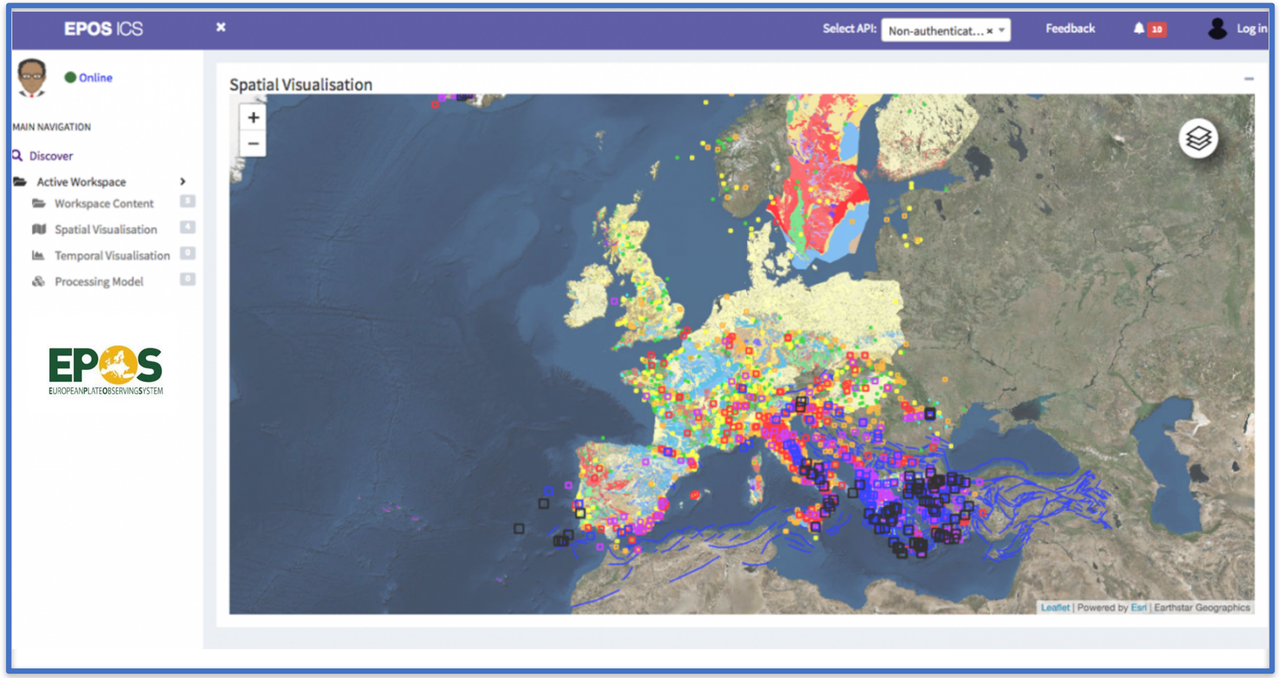
**Data Provenance: Information about processes' lineage and interactions between peers, methods and systems leading to scientific results**

# From Data Discovery to Working Sessions

https://www.epos-ip.org

**Objective: Extend Community Portals with focussed & reproducible workspaces**
*(Raw Data, Software, Scalable Workflow Executions, Results)*

Distributed vs Centralised Data Availability



# From Data Discovery to Working Sessions

https://www.epos-ip.org

**Objective: Extend Community Portals with focussed & reproducible workspaces**
*(Raw Data, Software, Scalable Workflow Executions, Results)*

Distributed vs Centralised Data Availability

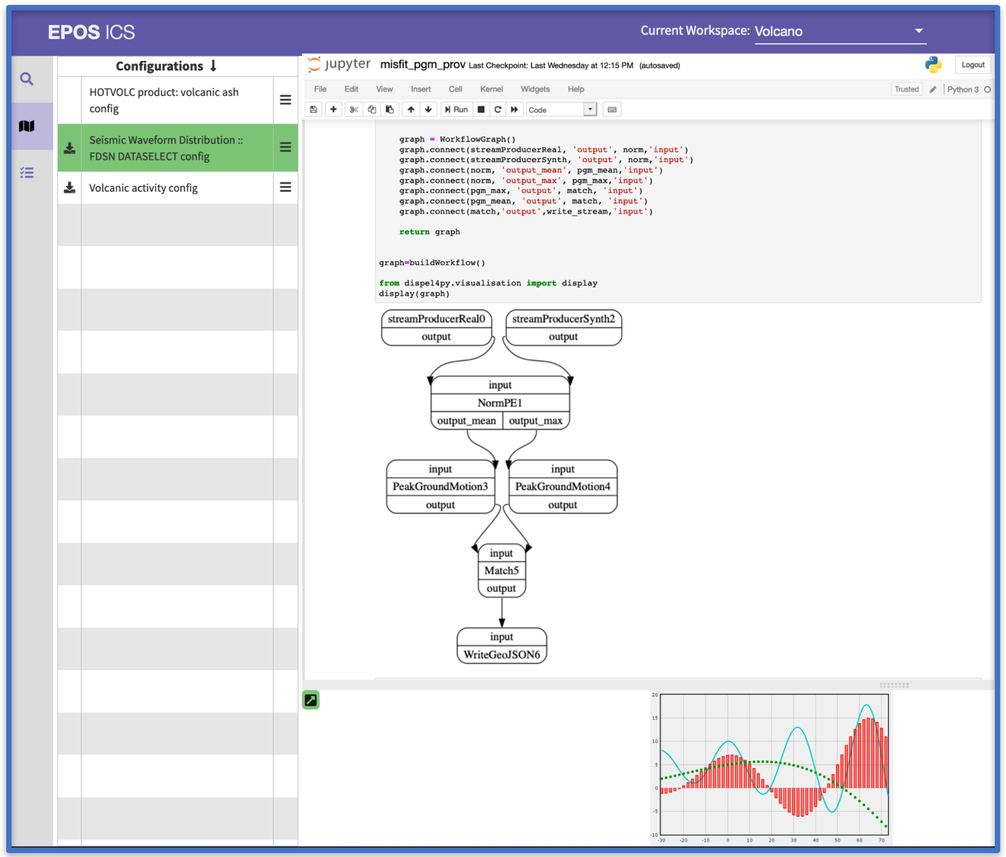
# From Data Discovery to Working Sessions

https://www.epos-ip.org

**Objective: Extend Community Portals with focussed & reproducible workspaces**
*(Raw Data, Software, Scalable Workflow Executions, Results)*

Distributed vs Centralised Data Availability


Documented Analysis
(Focussed Workspaces, Workflows)



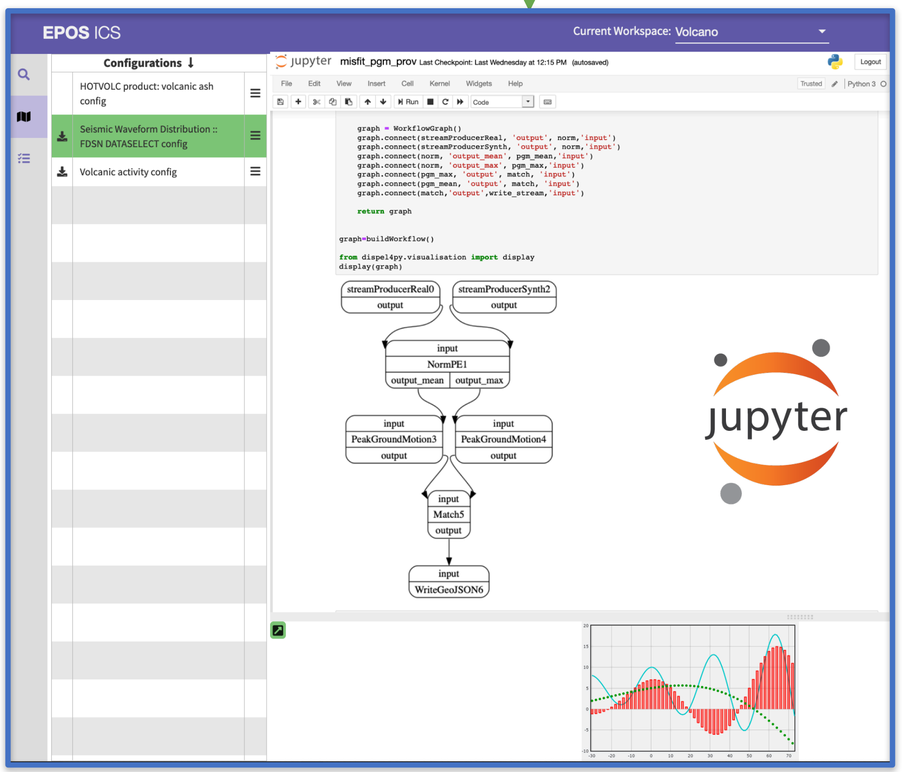
# Working Session Use Cases

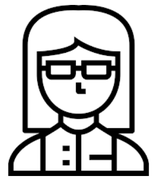
https://www.epos-ip.org

Data Resource A

Data Resource B

**Workflow for Data-Staging/Mounting/ Preprocessing**



**a researcher wants to**

- **stage/access raw data** from dedicated computational resources.
- **develop/apply custom and reusable methods** for processing and visualisation.
- **be informed about libraries** that fit the selected data and use them.
- **update the raw data** staged in the computational environment
- **keep old versions of original data when changed** (Reproducibility)
- **customise and restore the state of their computational environment** (Reproducibility)
- **share all of the above** with co-workers

# Setting up a Working Session
# Notebook API

https://www.epos-ip.org

## notebook — Setup the Computational Context / Locality

| | | |
|---|---|---|
| POST | `/notebook` | Deploys a notebook and creates a data and working directory. |
| PUT | `/notebook/{notebookId}` | Update Notebook libraries |
| GET | `/notebook/{notebookId}` | Retrieves the running notebook for the given notebookId. |
| DELETE | `/notebook/{notebookId}` | Deletes notebook for the given notebookId. |

## workflow — Data Staging / Data Preparation / Processing

| | | |
|---|---|---|
| POST | `/workflow/{workflowId}/run/` | Runs the specified workflow on a work space (notebook) |
| GET | `/workflow/{workflowId}/run/{runId}/` | Return the status of the workflow given by the given runId. |

# Setting up a Working Session Notebook API

https://www.epos-ip.org

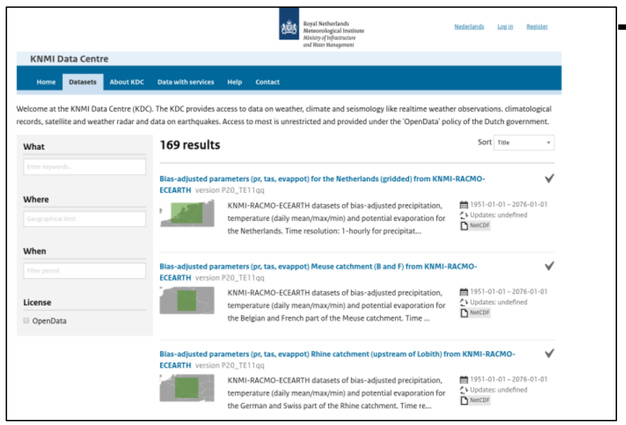
**Web API** to facilitate **integration as Service**

- **Create Session with initial library requirements**

- **Staging Workflow and Pre-processing**
  - Data staging history
    - Read-only and extensible raw data
  - **New:** Workflows at Scale

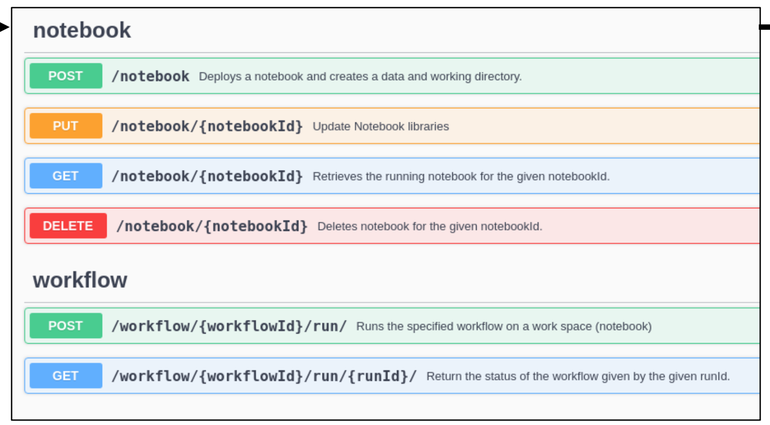
- **Library Updates** handled through the API and terminal

- **New:** **On demand snapshots** (in progress, Git/Binder)



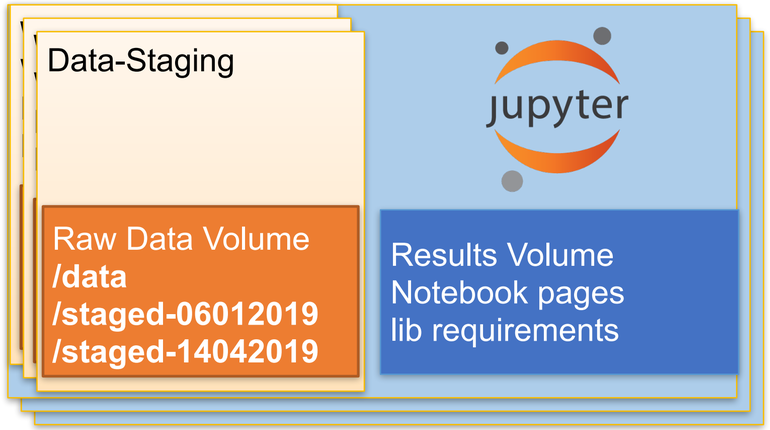

KDC KNMI DataCentrum
The data access point for all KNMI Data

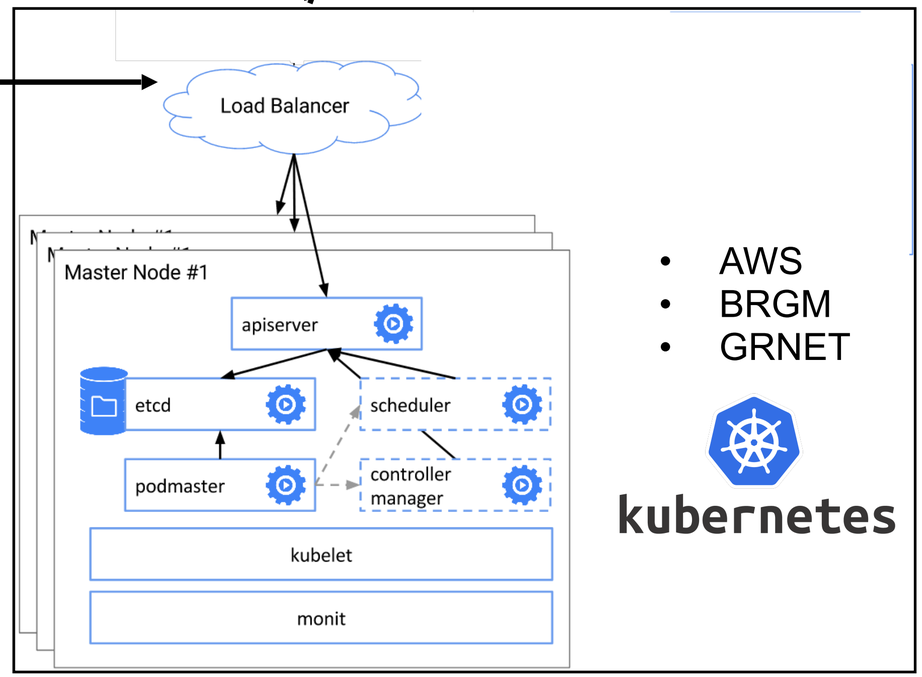
OPENAPI INITIATIVE




COMMON WORKFLOW LANGUAGE

docker



# Setting up a Working Session Notebook API

https://www.epos-ip.org

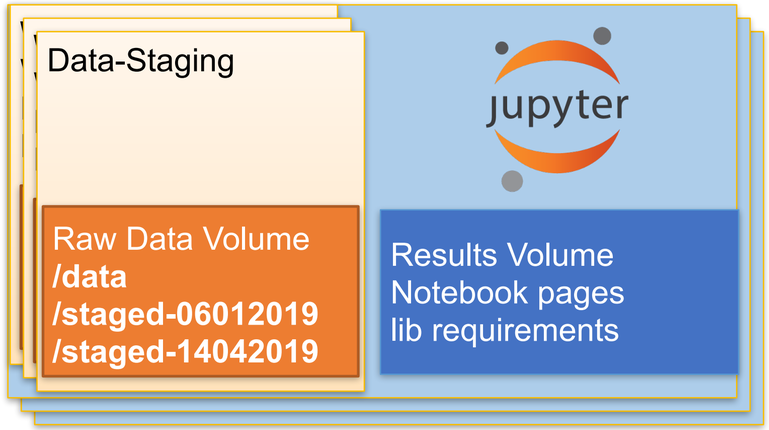
**Web API** to facilitate **integration as Service**

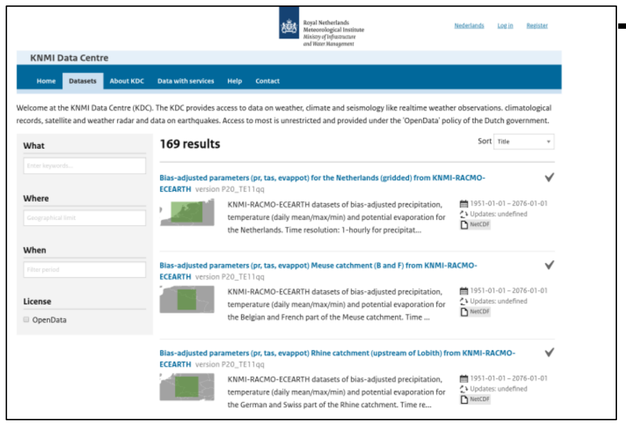
- **Create Session with initial library requirements**

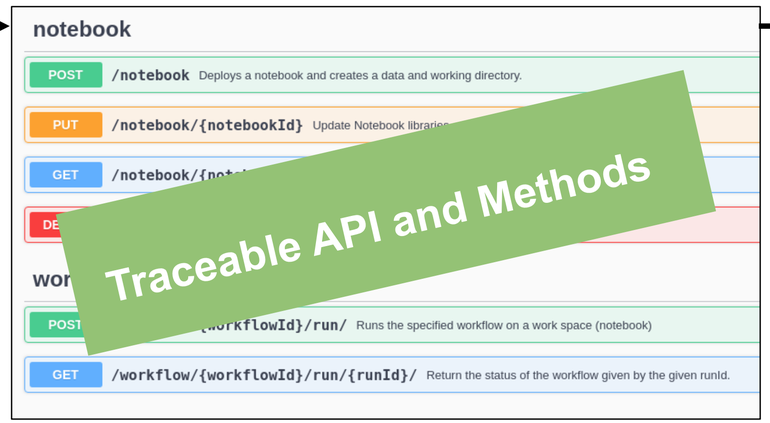
- **Staging Workflow and Pre-processing**
  - Data staging history
    - Read-only and extensible raw data
  - **New:** Workflows at Scale

- **Library Updates** handled through the API and terminal

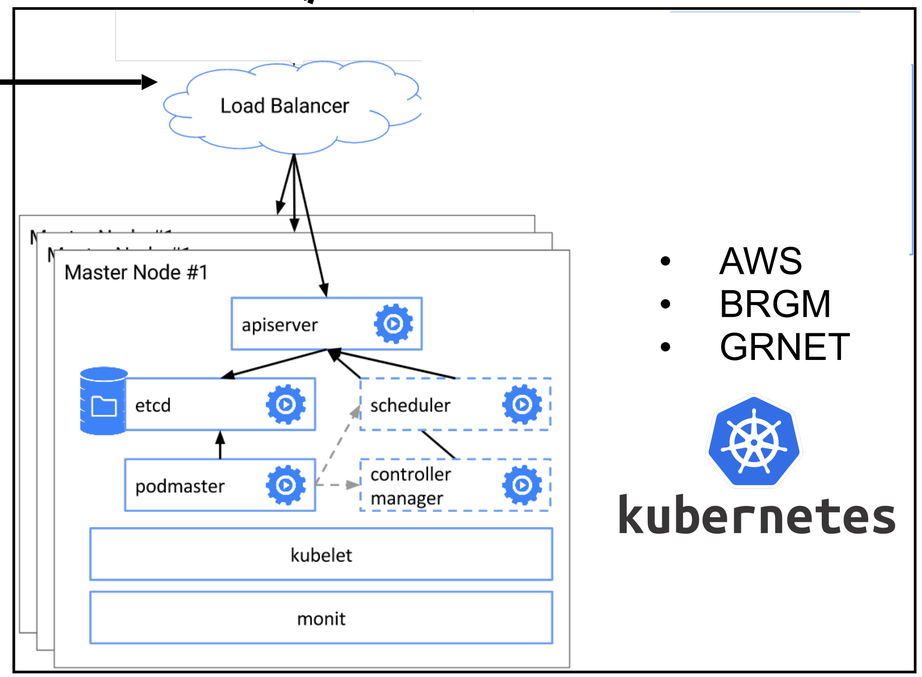
- **New:** **On demand snapshots** (in progress, Git/Binder)




Traceable API and Methods



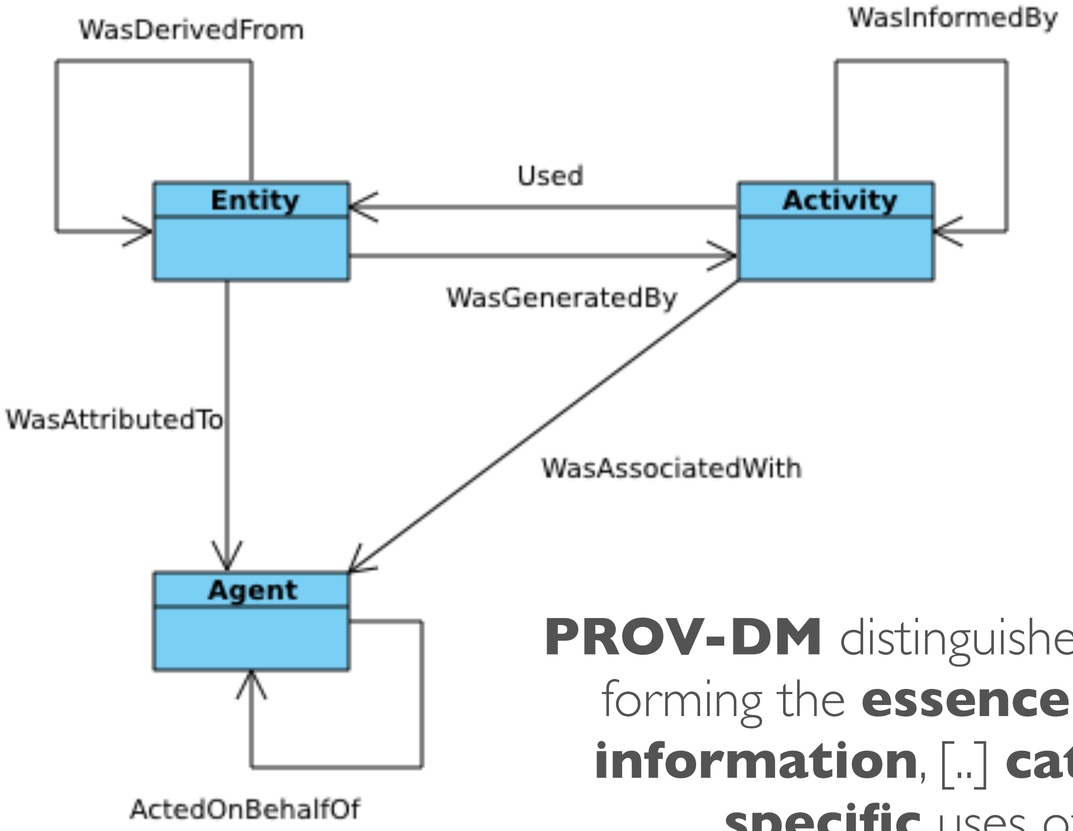

- AWS
- BRGM
- GRNET

kubernetes

KDC KNMI DataCentrum
The data access point for all KNMI Data

# Provenance Model

**Data-Lineage** → **Knowledge**




W3C PROV

**PROV-DM** distinguishes **core structures**, forming the **essence of provenance information**, [..] **catering for more specific** uses of provenance.

**Enactment** ← **Machinery & Data**

# Provenance Templating Techniques

- **Foster discussions** on provenance relationships involving heterogeneous systems and agents.
- **Modelling of usable** and **re-usable** provenance scenarios

  (tailoring vs generalisation)
- **Remove the burden** to **hardcode provenance editing**

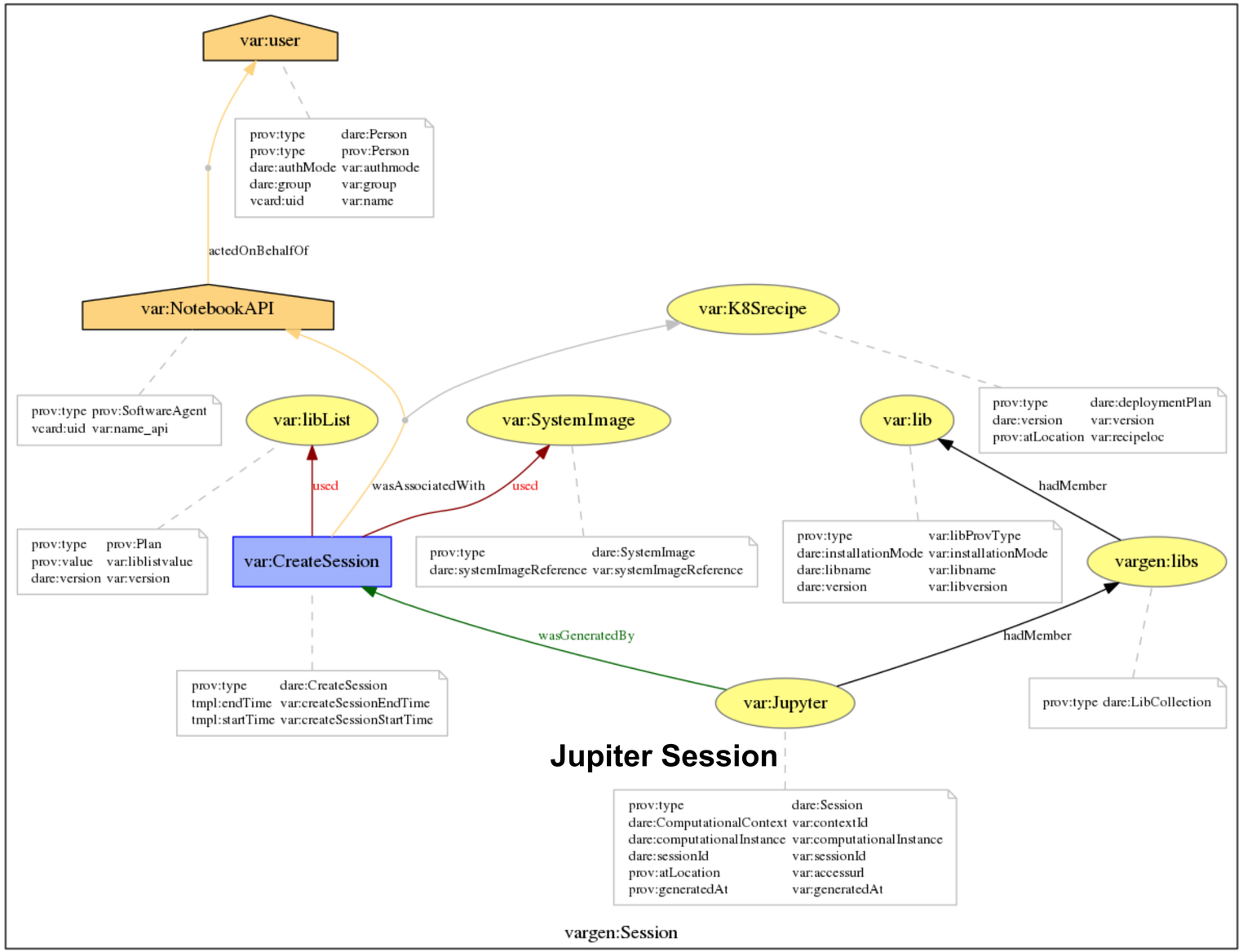
  (expansion tools/services)
- **Bindings in applications** can directly map to updated Provenance Templates

  (Decoupling and weight-shift)

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/

# “Create” Notebook Template

https://www.epos-ip.org



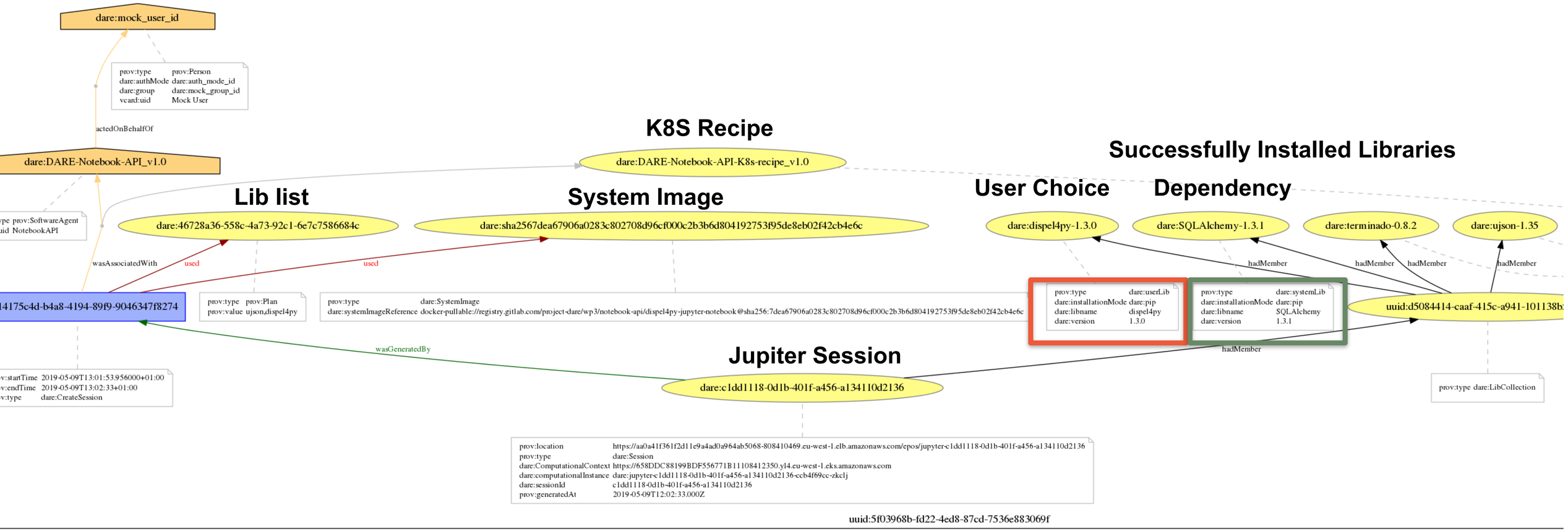

**PROV Templates to capture provenance of** the **creation** and **interactive update** of a **Notebook Working Session**

**Distinction between user’s and system’s concerns** to **recall choices and reproduce technical setups**.

Example available at:
https://openprovenance.org/store/documents/904

# “Create” Notebook Instance

https://www.epos-ip.org




https://openprovenance.org/store/documents/916

# “Update” Notebook Template

https://openprovenance.org/store/documents/914

Copy to Clipboard

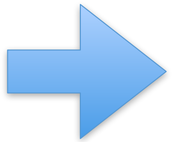
```
document
  prefix vargen <http://openprovenance.org/vargen#>
  prefix s-prov <http://s-prov/ns/#>
  prefix pre_0 <http://www.w3.org/2001/XMLSchema>
  prefix dare <http://project-dare.eu/ns#>
  prefix d-prov <http://d-prov.org/#>
  prefix vcard <http://www.w3.org/2006/vcard/ns#>
  prefix var <http://openprovenance.org/var#>
  prefix tmpl <http://openprovenance.org/tmpl#>
  prefix dcterms <http://purl.org/dc/terms/>
  prefix foaf <http://xmlns.com/foaf/0.1/>

  bundle vargen:Session
    prefix vargen <http://openprovenance.org/vargen#>
    prefix dare <http://project-dare.eu/ns#>
    prefix vcard <http://www.w3.org/2006/vcard/ns#>
    prefix var <http://openprovenance.org/var#>
    prefix tmpl <http://openprovenance.org/tmpl#>
    prefix dcterms <http://purl.org/dc/terms/>

    entity(var:SystemImage, [dare:systemImageReference='var:systemImageReference', prov:type='dare:S
    entity(var:Jupyter, [prov:generatedAt='var:generatedAt', prov:atLocation='var:accessurl', dare:s
    entity(var:K8Srecipe, [prov:atLocation='var:recipeloc', dare:version='var:version', prov:type='d
    entity(var:lib, [dare:libname='var:libname', dare:version='var:libversion', dare:installationMod
    entity(var:libList, [dare:version='var:version', prov:type='prov:Plan', prov:value='var:liblistv
    entity(vargen:libs, [prov:type='dare:LibCollection'])
    used(var:CreateSession, var:SystemImage, -)
    used(var:CreateSession, var:libList, -)
    wasAssociatedWith(var:CreateSession, var:NotebookAPI, var:K8Srecipe)
    activity(var:CreateSession, -, -, [tmpl:startTime='var:createSessionStartTime', prov:type='dare:
    actedOnBehalfOf(var:NotebookAPI, var:user, -)
    agent(var:user, [vcard:uid='var:name', dare:authMode='var:authmode', dare:group='var:group', pro
    agent(var:NotebookAPI, [vcard:uid='var:name_api', prov:type='prov:SoftwareAgent'])
    wasGeneratedBy(var:Jupyter, var:CreateSession, -)
    hadMember(var:Jupyter, vargen:libs)
    hadMember(vargen:libs, var:lib)
  endBundle
endDocument
```

**PROV-N**

https://www.w3.org/TR/prov-n/



# “Update” Notebook Template

Copy to Clipboard

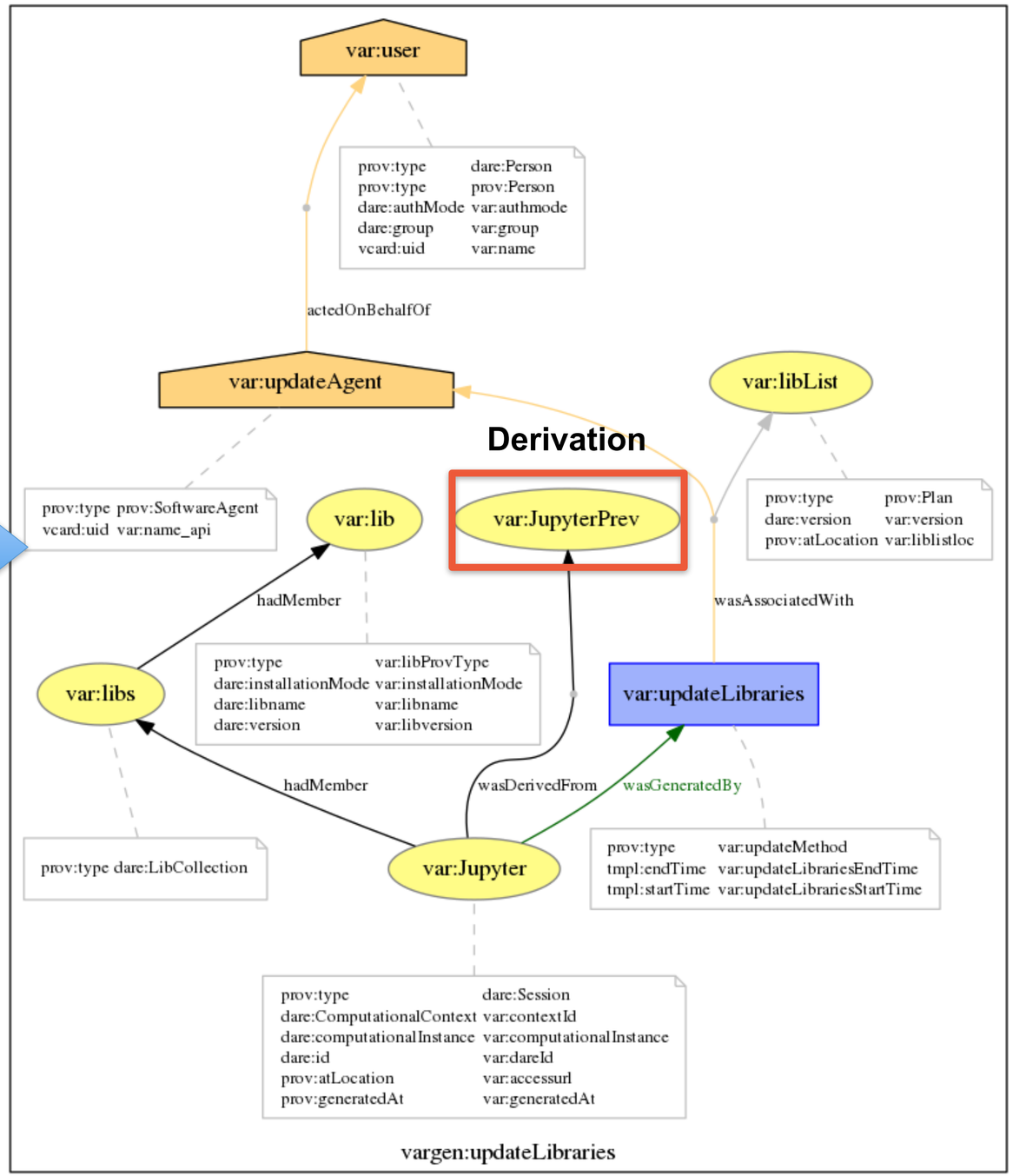
```
document
  prefix vargen <http://openprovenance.org/vargen#>
  prefix s-prov <http://s-prov/ns/#>
  prefix pre_0 <http://www.w3.org/2001/XMLSchema>
  prefix dare <http://project-dare.eu/ns#>
  prefix d-prov <http://d-prov.org/#>
  prefix vcard <http://www.w3.org/2006/vcard/ns#>
  prefix var <http://openprovenance.org/var#>
  prefix tmpl <http://openprovenance.org/tmpl#>
  prefix dcterms <http://purl.org/dc/terms/>
  prefix foaf <http://xmlns.com/foaf/0.1/>

  bundle vargen:Session
    prefix vargen <http://openprovenance.org/vargen#>
    prefix dare <http://project-dare.eu/ns#>
    prefix vcard <http://www.w3.org/2006/vcard/ns#>
    prefix var <http://openprovenance.org/var#>
    prefix tmpl <http://openprovenance.org/tmpl#>
    prefix dcterms <http://purl.org/dc/terms/>

    entity(var:SystemImage, [dare:systemImageReference='var:systemImageReference', prov:type='dare:S
    entity(var:Jupyter, [prov:generatedAt='var:generatedAt', prov:atLocation='var:accessurl', dare:s
    entity(var:K8Srecipe, [prov:atLocation='var:recipeloc', dare:version='var:version', prov:type='d
    entity(var:lib, [dare:libname='var:libname', dare:version='var:libversion', dare:installationMod
    entity(var:libList, [dare:version='var:version', prov:type='prov:Plan', prov:value='var:liblistv
    entity(vargen:libs, [prov:type='dare:LibCollection'])
    used(var:CreateSession, var:SystemImage, -)
    used(var:CreateSession, var:libList, -)
    wasAssociatedWith(var:CreateSession, var:NotebookAPI, var:K8Srecipe)
    activity(var:CreateSession, -, -, [tmpl:startTime='var:createSessionStartTime', prov:type='dare:
    actedOnBehalfOf(var:NotebookAPI, var:user, -)
    agent(var:user, [vcard:uid='var:name', dare:authMode='var:authmode', dare:group='var:group', pro
    agent(var:NotebookAPI, [vcard:uid='var:name_api', prov:type='prov:SoftwareAgent'])
    wasGeneratedBy(var:Jupyter, var:CreateSession, -)
    hadMember(var:Jupyter, vargen:libs)
    hadMember(vargen:libs, var:lib)
  endBundle
endDocument
```

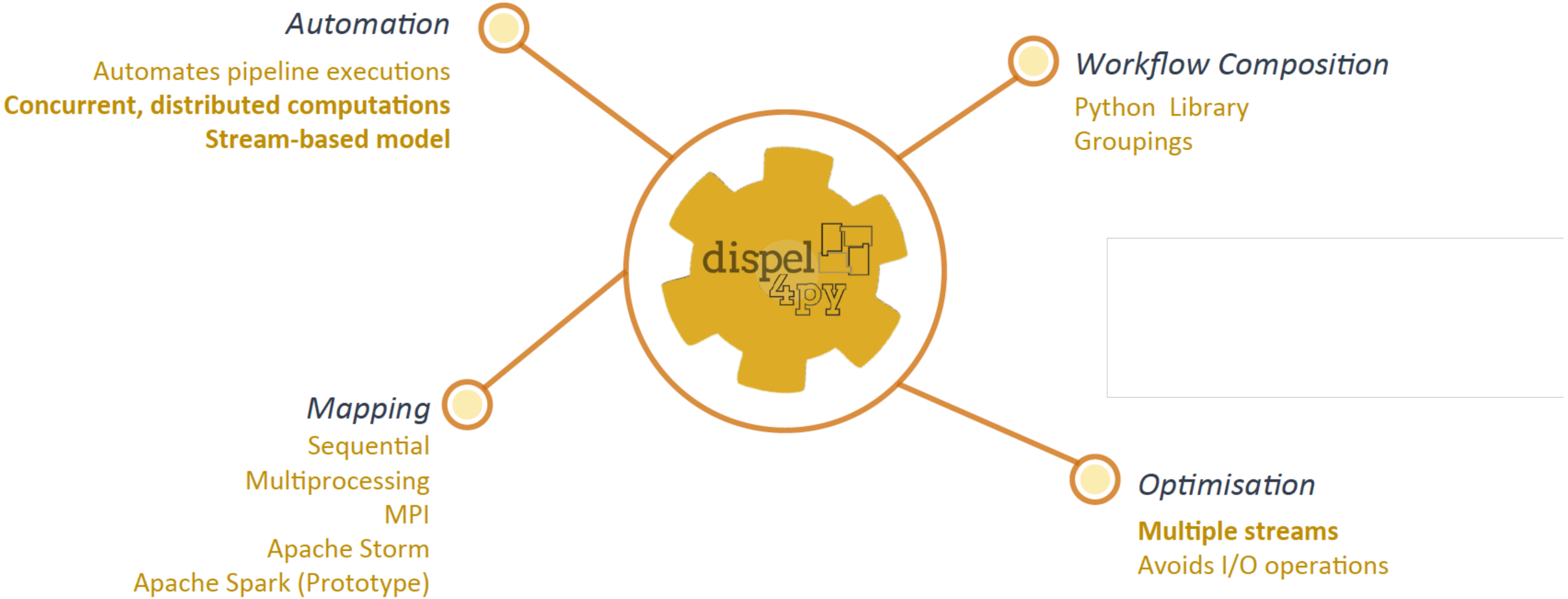
**PROV-N**
https://www.w3.org/TR/prov-n/

https://openprovenance.org/store/documents/914



# Workflows execution and lineage management

DARE http://project-dare.eu




**Key-features**: Automatic parallelisation/mappings, concurrent & stream-based, configurable provenance

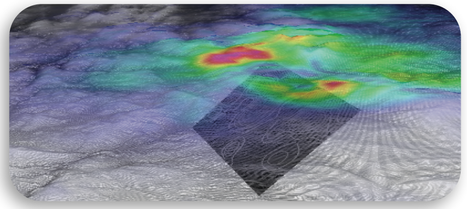
https://github.com/dispel4py/dispel4py

# Test Case:
# Seismic Rapid Assessment

http://project-dare.eu

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

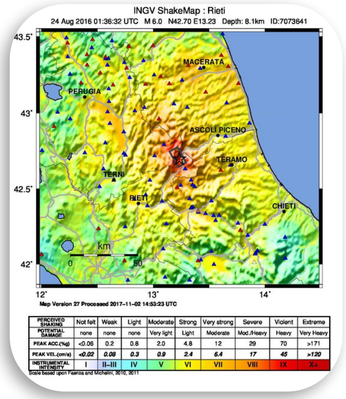
Gather **observed data**

**Waveform Preprocessing**

**Peak Round Motion Parameters**

Compare/integrate synthetic and observed ground motion data

Store data, metadata, provenance



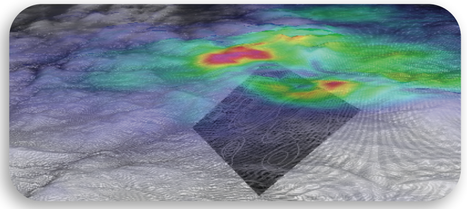
# Test Case:
# Seismic Rapid Assessment

http://project-dare.eu

Rapid Ground Motion Assessment (RA)

**Reusable Tasks running at different scale.**
**May require human monitoring and intervention**

**MPI Simulation**



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

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

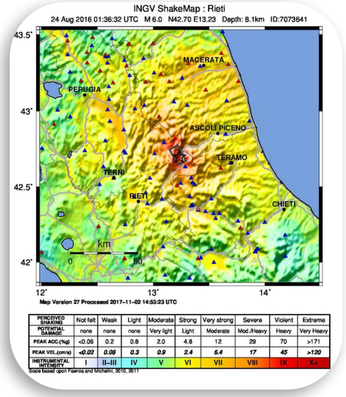
run waveform simulation

Gather **observed data**

**Waveform Preprocessing**

**Peak Round Motion Parameters**

**Data Analysis**



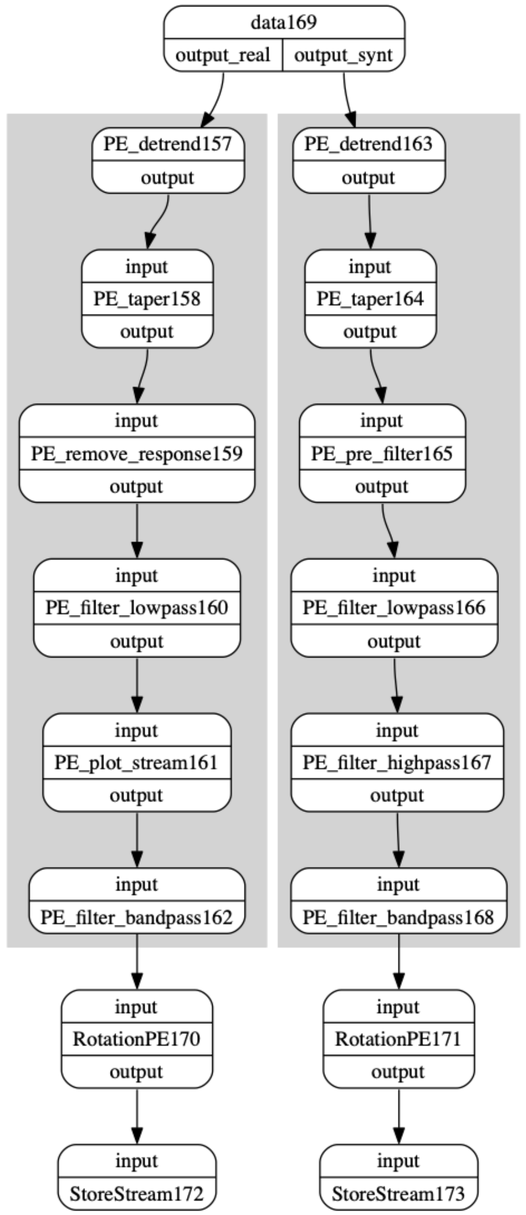

Compare/integrate synthetic and observed ground motion data

Store data, metadata, provenance

# Workflow development

DARE http://project-dare.eu

**Waveform Preprocessing**




pipeline JSON Description (eg. from file)

Manual Extensions

## Workflow encoded in 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)
```

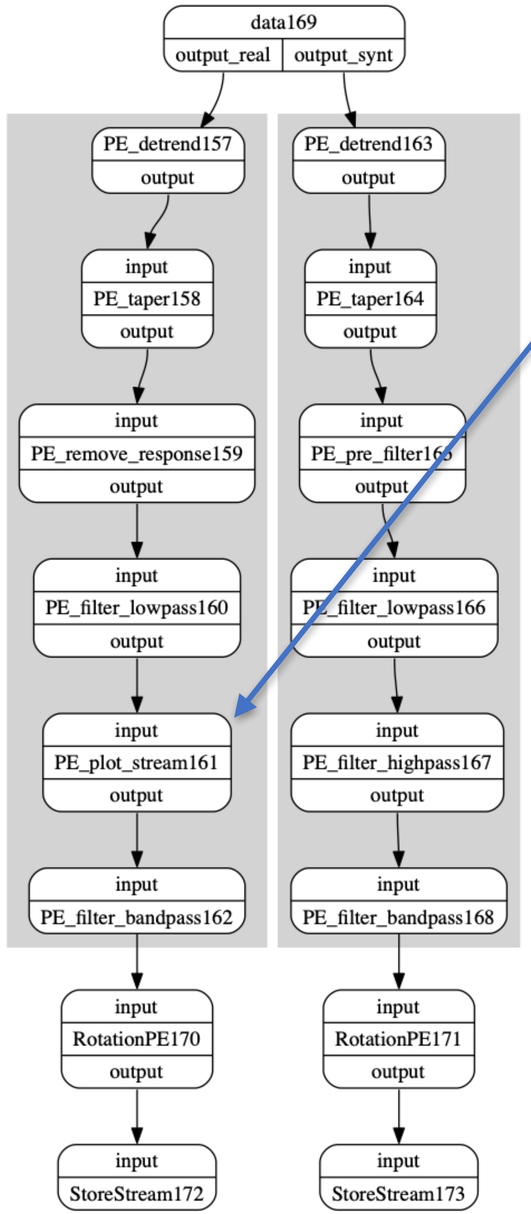
dispel4py

# Provenance Configuration for lineage usability



http://project-dare.eu

**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}
```

dispel4py

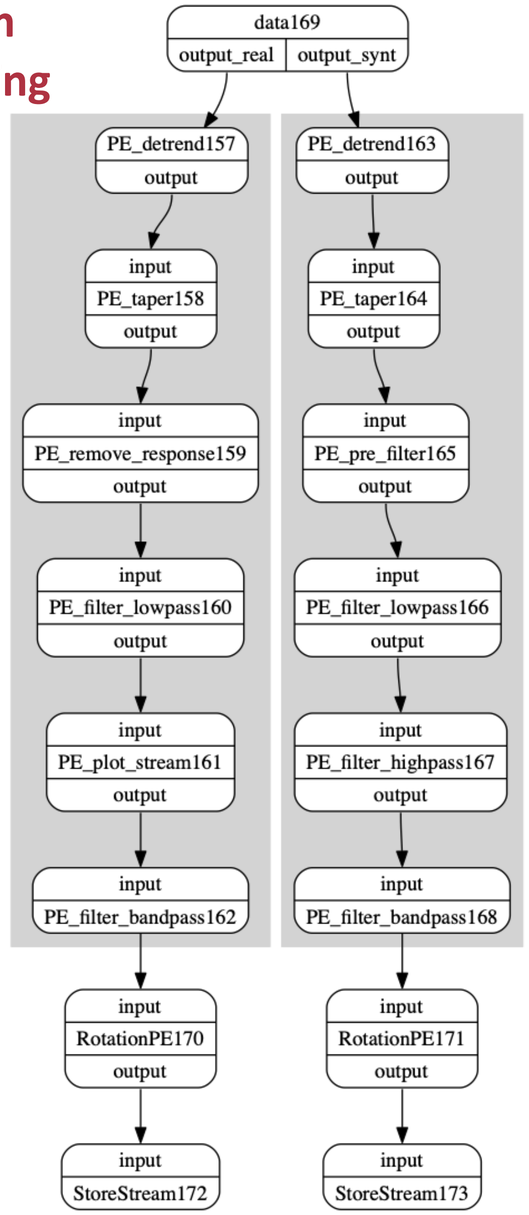
**User Defined Metadata**

# Provenance Configuration for lineage usability



DARE http://project-dare.eu

**Waveform Preprocessing**



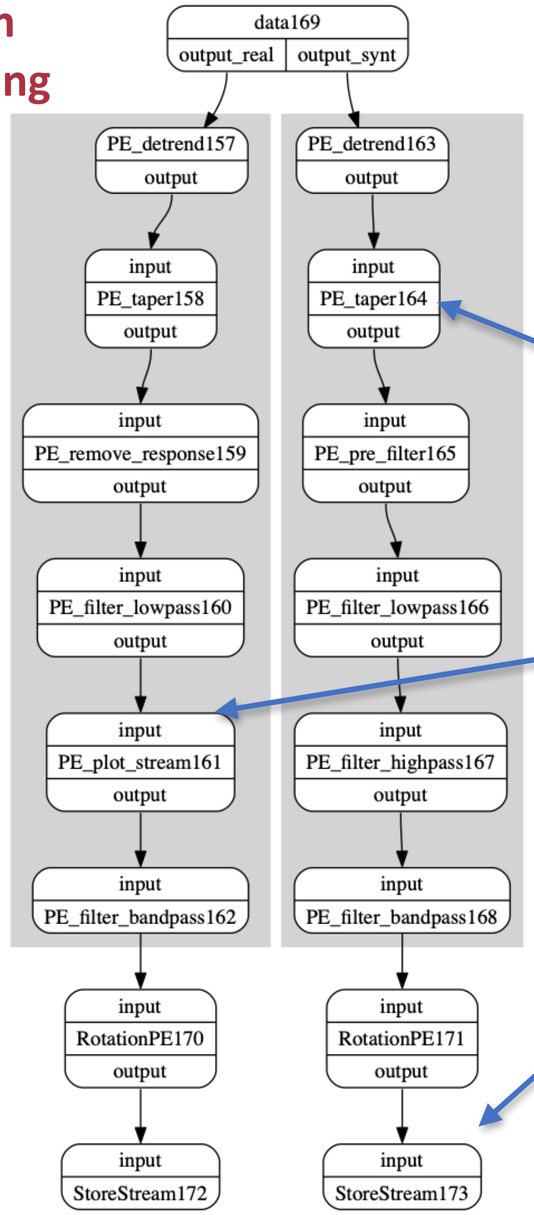

## Configuration Profile in JSON with Provenance Types

```
{
     '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

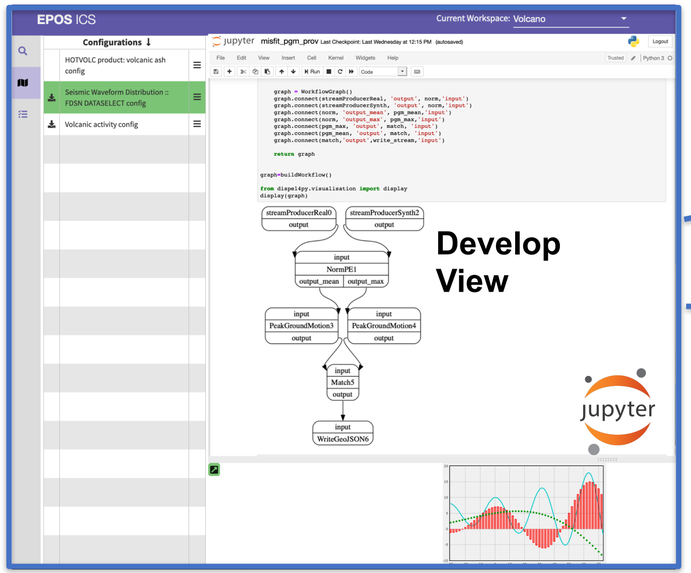
**calib:** 1

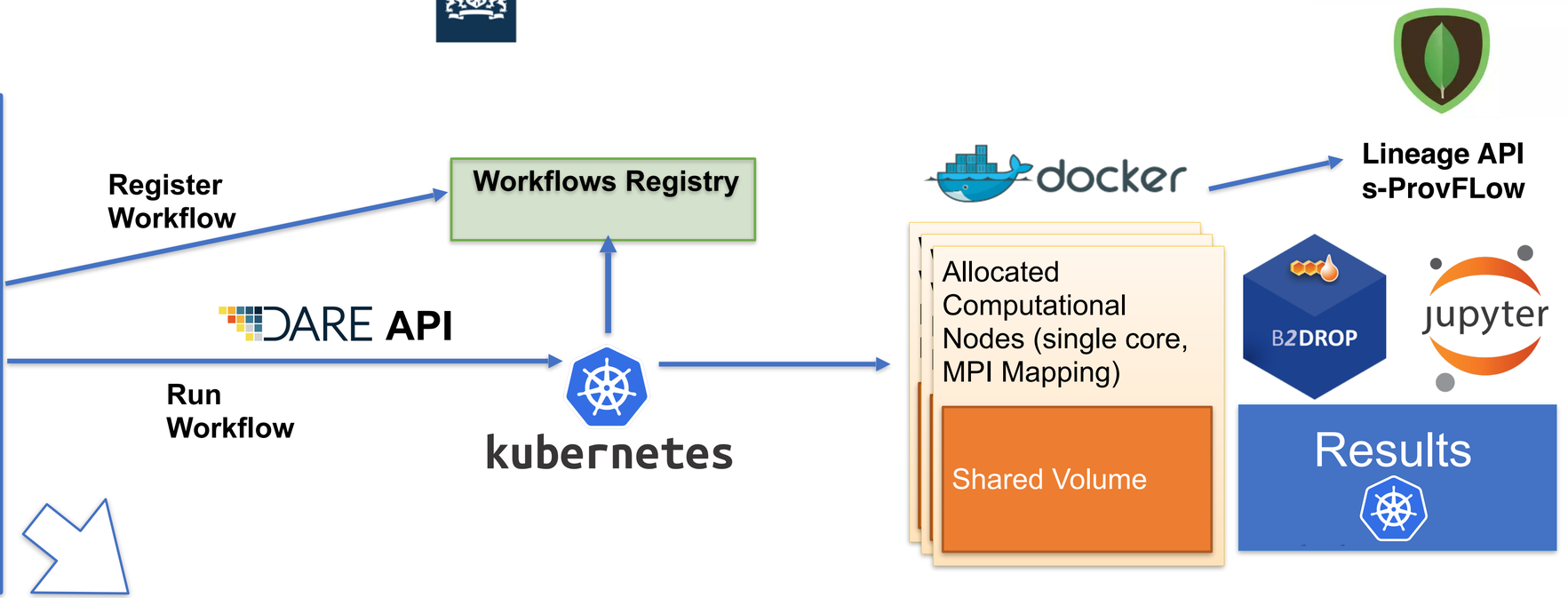
**sampling_rate:** 100

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

# Workflow execution

http://project-dare.eu




- *Develop & Register Workflows*
- *Scalable Workflow execution on containerised resources*
- *Lineage Capturing and visualisation*

**Register dispel4py workflow**

```
In [ ]: # Local code
impl_id = F.create_peimpl_temp(desc="", code=In[2],
                               parent_sig=pe_url, pckg="test_impl",
                               name="waveform_preprocessing", workspace=workspace_url,
                               clone="", creds=creds)

print impl_id
```

**Execute registered workflow**

```
In [ ]: F.submit_d4p(impl_id=impl_id, pckg="test", workspace_id=workspace_id, pe_name="waveform_preprocessing",
                    token=F.auth(), creds=creds, n_nodes=6, no_processes=6, iterations=1)
```

# Workflow execution

DARE http://project-dare.eu

## Submit Run and Specify Libraries

In [ ]:
```
F.submit_d4p(impl_id=impl_id, pckg="test", workspace_id=workspace_id, pe_name="waveform_preprocessing",
             token=F.auth(),
             creds=creds,
             reqs='https://gitlab.com/project-dare/dare-api/raw/master/examples/jupyter/requirements.txt',
             n_nodes=6, no_processes=6, iterations=1)
```

## Upload / Download / Export (To/From Cluster and External Repo b2drop)

In [ ]:
```
os.system('zip -r input.zip input.json')
F.upload(token=F.auth(), path='d4p-input', local_path='input.zip', creds=creds)
```

In [ ]:
```
resp = F.myfiles(token=F.auth(), creds=creds)
F.files_pretty_print(json.loads(resp))
```

In [ ]:
```
resp = F._list(path="/home/mpiuser/sfs/uploads/Th1s4sY0urT0k3Nn_d4p-input", creds=creds)
F._list_pretty_print(json.loads(resp))
```

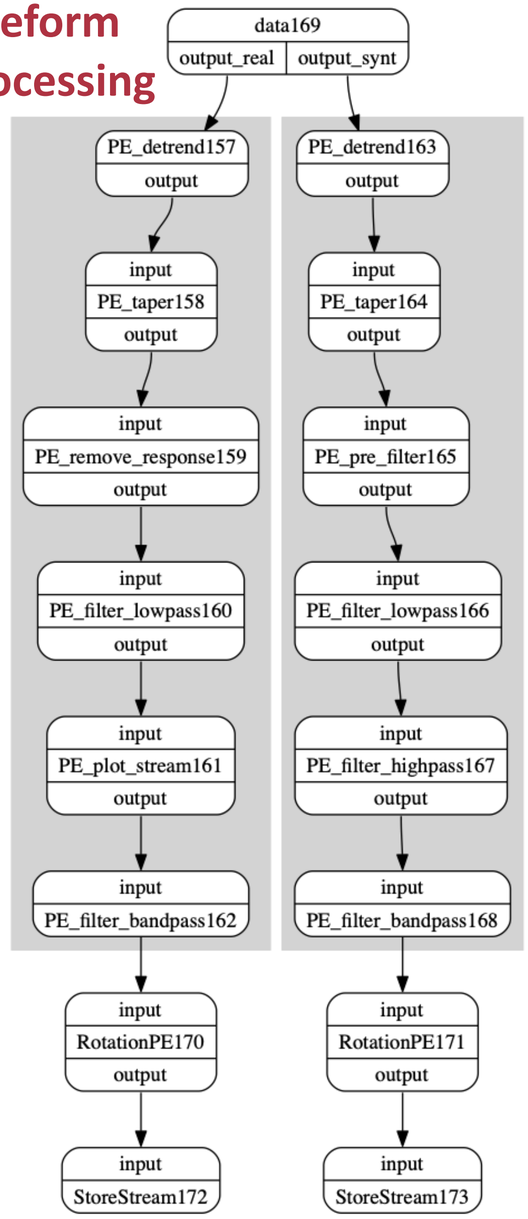
In [ ]:
```
F.download(path="home/mpiuser/sfs/d4p/Th1s4sY0urT0k3Nn_d4p-input/input.json", creds=creds, local_path='123.json')
```

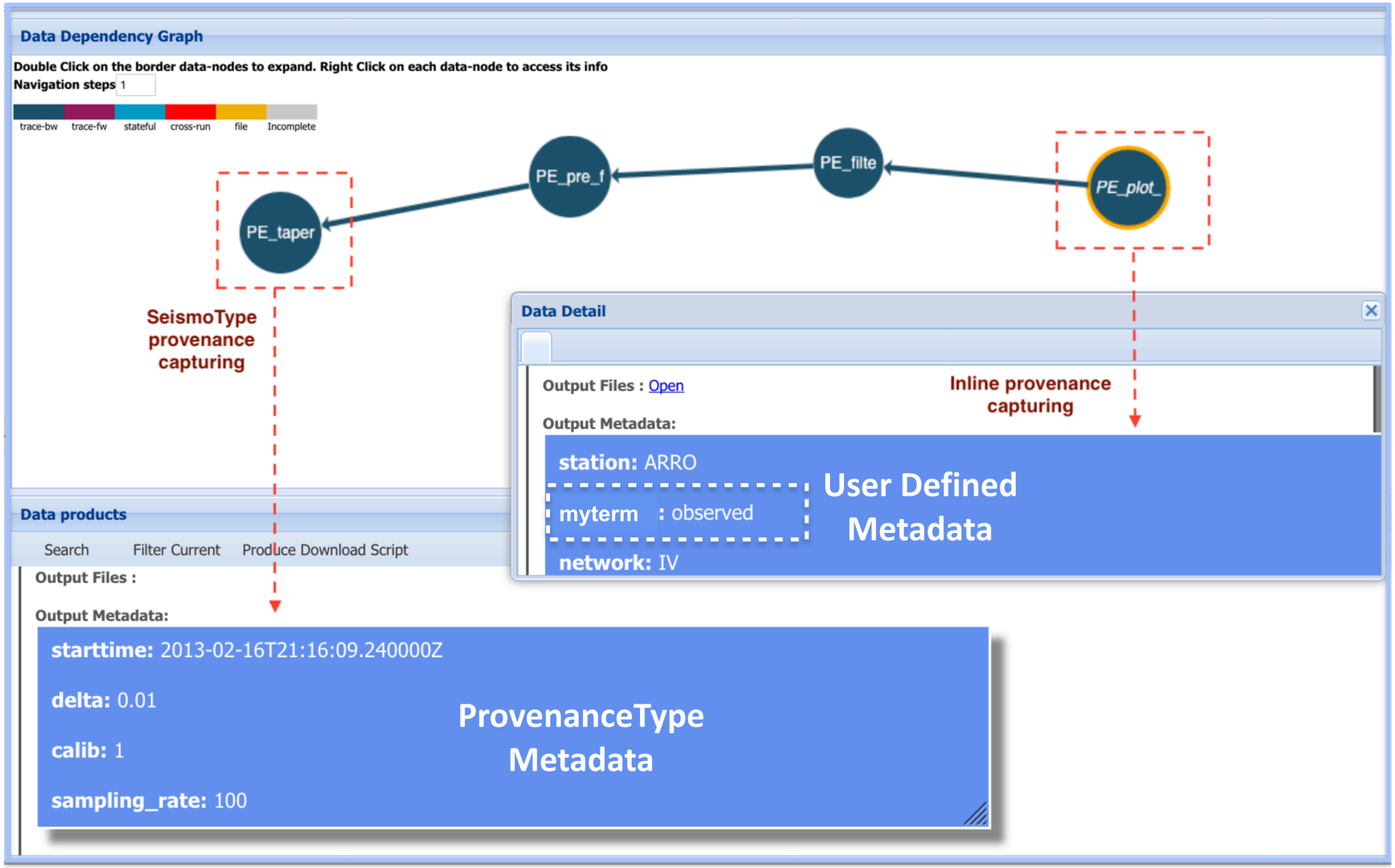
In [ ]:
```
F.send2drop(token=F.auth(), creds=creds, path="/home/mpiuser/sfs/Th1s4sY0urT0k3Nn_d4p-input/input.json")
```

# Monitor, search and analyse results through lineage



**Waveform Preprocessing**

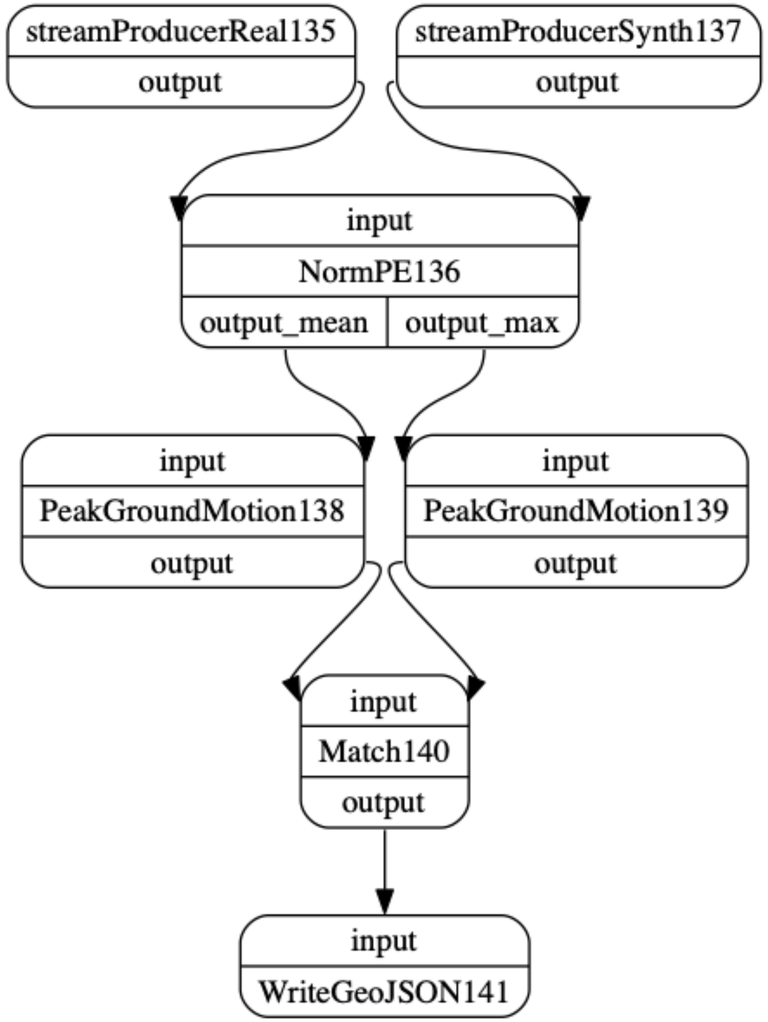


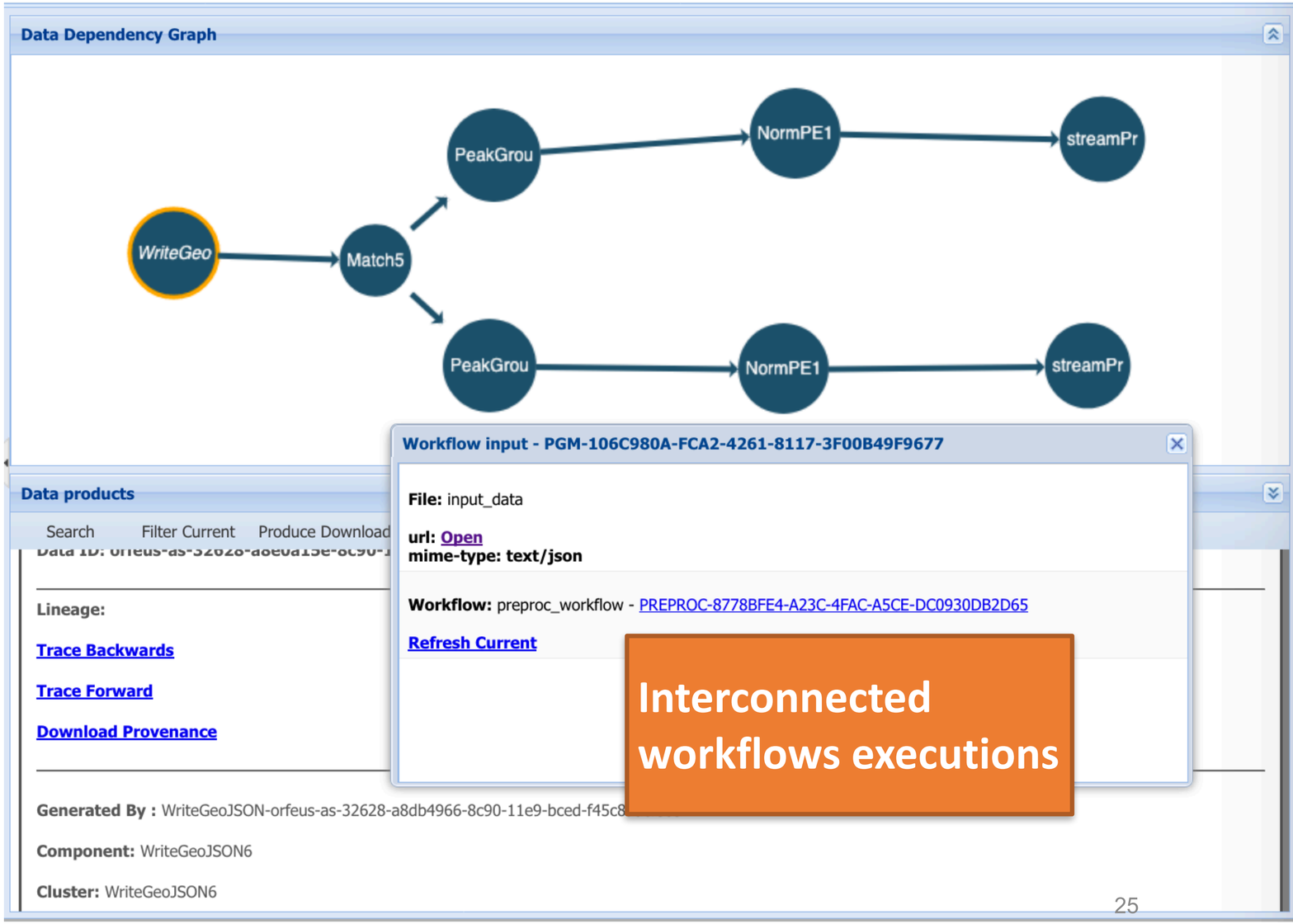



# Monitor, search and analyse results through lineage

DARE http://project-dare.eu

## Peak Ground Motion

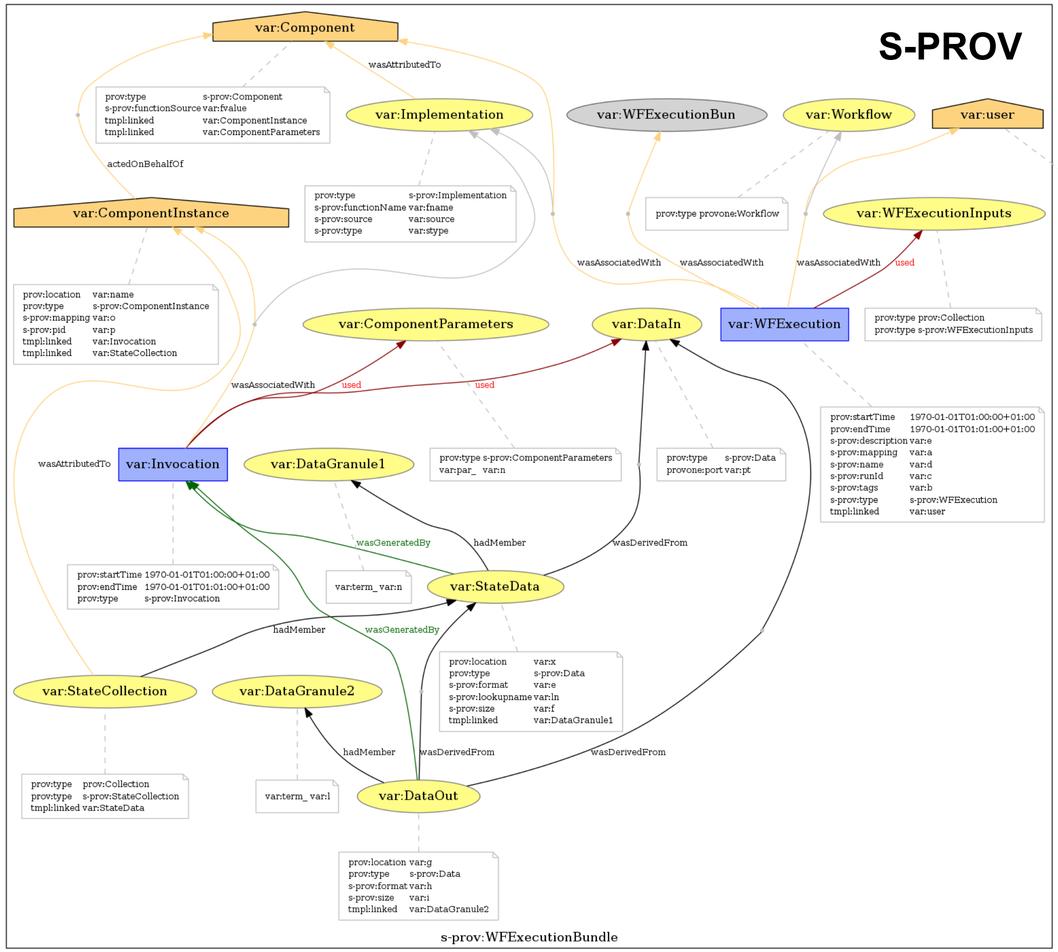


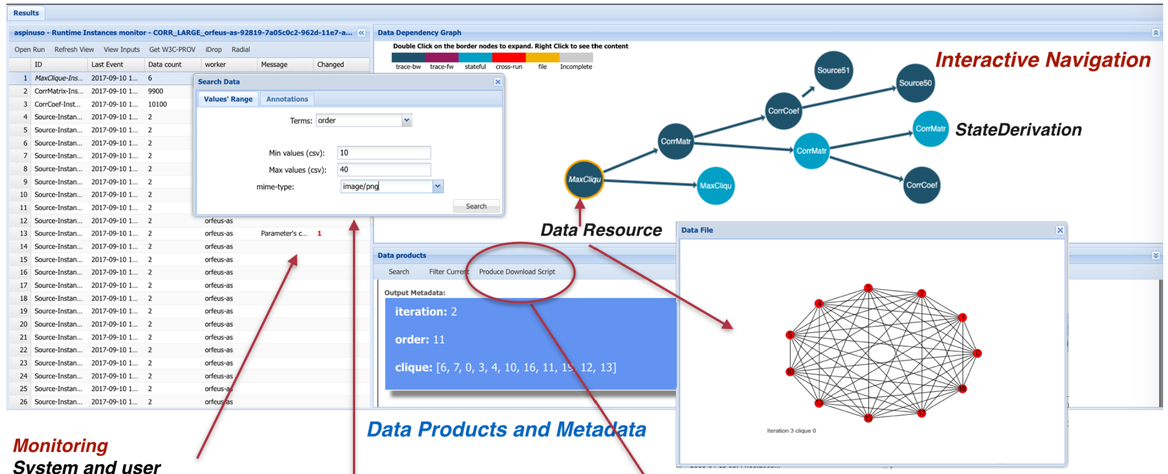



# Large Scale Lineage Representation, Management, Exploitation

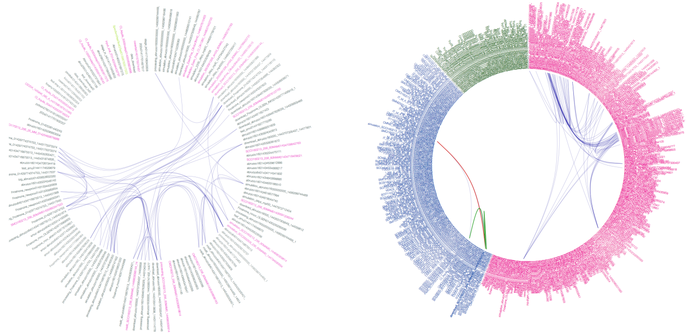
DARE http://project-dare.eu

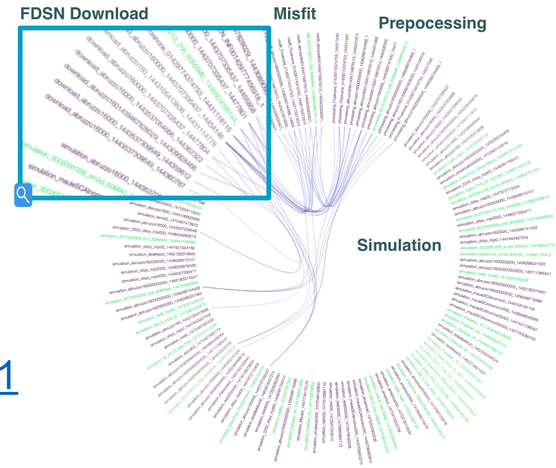
## Provenance Model, Services and Tools combining system, domain and user-driven information about the computation




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

DARE http://project-dare.eu

## 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 |

# Conclusion: Integrated Platform

(Reproducible Unified Actions)

DARE http://project-dare.eu

## User Facing Tools

**Integrated Development environment**

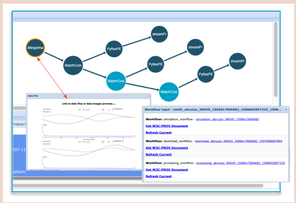
jupyter

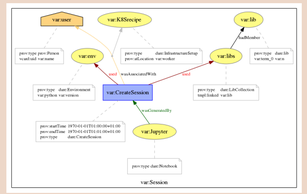
## Unified Actions

**Manage Working Sessions**

## Services (Provenance, Execution and Catalogues)

**Provenance Services**





**Computational Contexts allocation and customisation Services**

**Working Session and Lineage**

# Conclusion: Integrated Platform

(Reproducible Unified Actions)

DARE http://project-dare.eu

## User Facing Tools

**Integrated Development environment**

jupyter

## Unified Actions

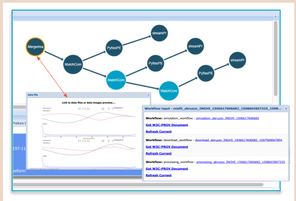
**Manage Working Sessions**

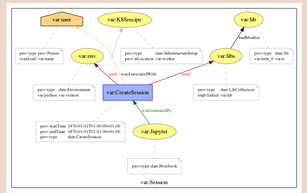
**Configure Run's**

**Compose Register Workflows**

## Services (Provenance, Execution and Catalogues)

**Provenance Services**





**Working Session and Lineage**

**Computational Contexts allocation and customisation Services**

**Catalogues for Data and Methods**

# Conclusion: Integrated Platform

(Reproducible Unified Actions)

DARE http://project-dare.eu

## User Facing Tools

**Integrated Development environment**

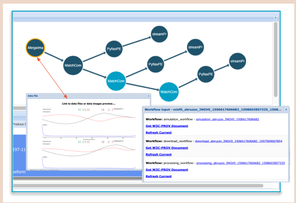
jupyter

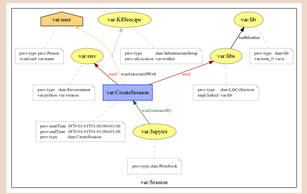
## Unified Actions

- Manage Working Sessions
- Execute and Monitor Runs
- Configure Run's
- Compose Register Workflows

## Services (Provenance, Execution and Catalogues)

**Provenance Services**





**Working Session and Lineage**

**Computational Contexts allocation and customisation Services**

**Workflow Mapping and Execution**

**Catalogues for Data and Methods**

# Conclusion: Integrated Platform

(Reproducible Unified Actions)

DARE http://project-dare.eu

## User Facing Tools

**Integrated Development environment**

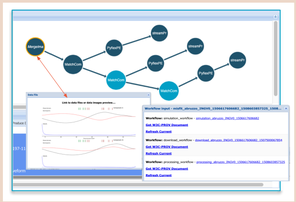
jupyter

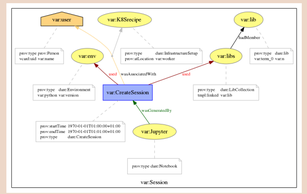
## Unified Actions

- Manage Working Sessions
- Validate Results, Reproduce
- Execute and Monitor Runs
- Configure Run's
- Compose Register Workflows

## Services (Provenance, Execution and Catalogues)

**Provenance Services**





**Computational Contexts allocation and customisation Services**

**Workflow Mapping and Execution**

**Catalogues for Data and Methods**

**Working Session and Lineage**

**Optimisation**

Koninklijk Nederlands
Meteorologisch Instituut
*Ministerie van Infrastructuur en Milieu*

DARE http://project-dare.eu

EPOS EUROPEANPLATEOBSERVINGSYSTEM https://www.epos-ip.org

# Thanks!

Jupyter Meetup 13 June 2019