Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Infrastructuur en Milieu

DARE http://project-dare.eu

Active provenance for Data-Intensive workflows: engaging users and developers

Alessandro Spinuso, Malcom Atkinson, Federica Magnoni

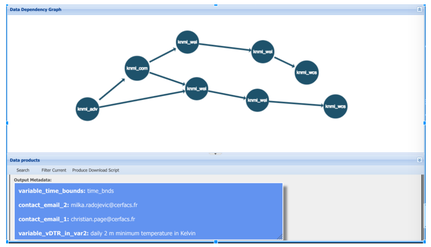
BC2DC workshop - eScience 2019

What's in this talk…

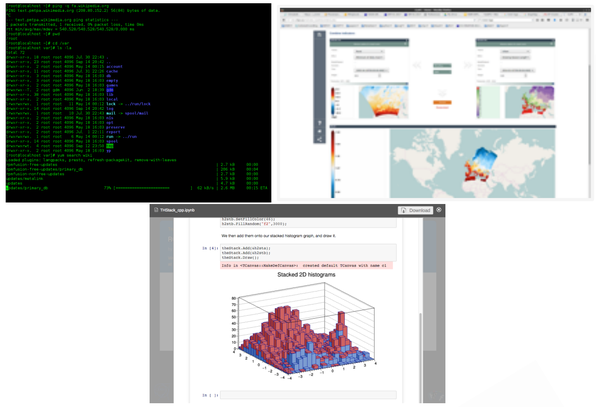
- **Research Cycles and Provenance Challenges**
- **Provenance Model:** S-PROV, Data-Intensive Workflows
- ***Active* Provenance Capturing:** Types and Configuration
- **Evaluation Use Case and Tooling**
- **Conclusions**

The Research Cycle(s)

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



Validate / Monitor / Explore



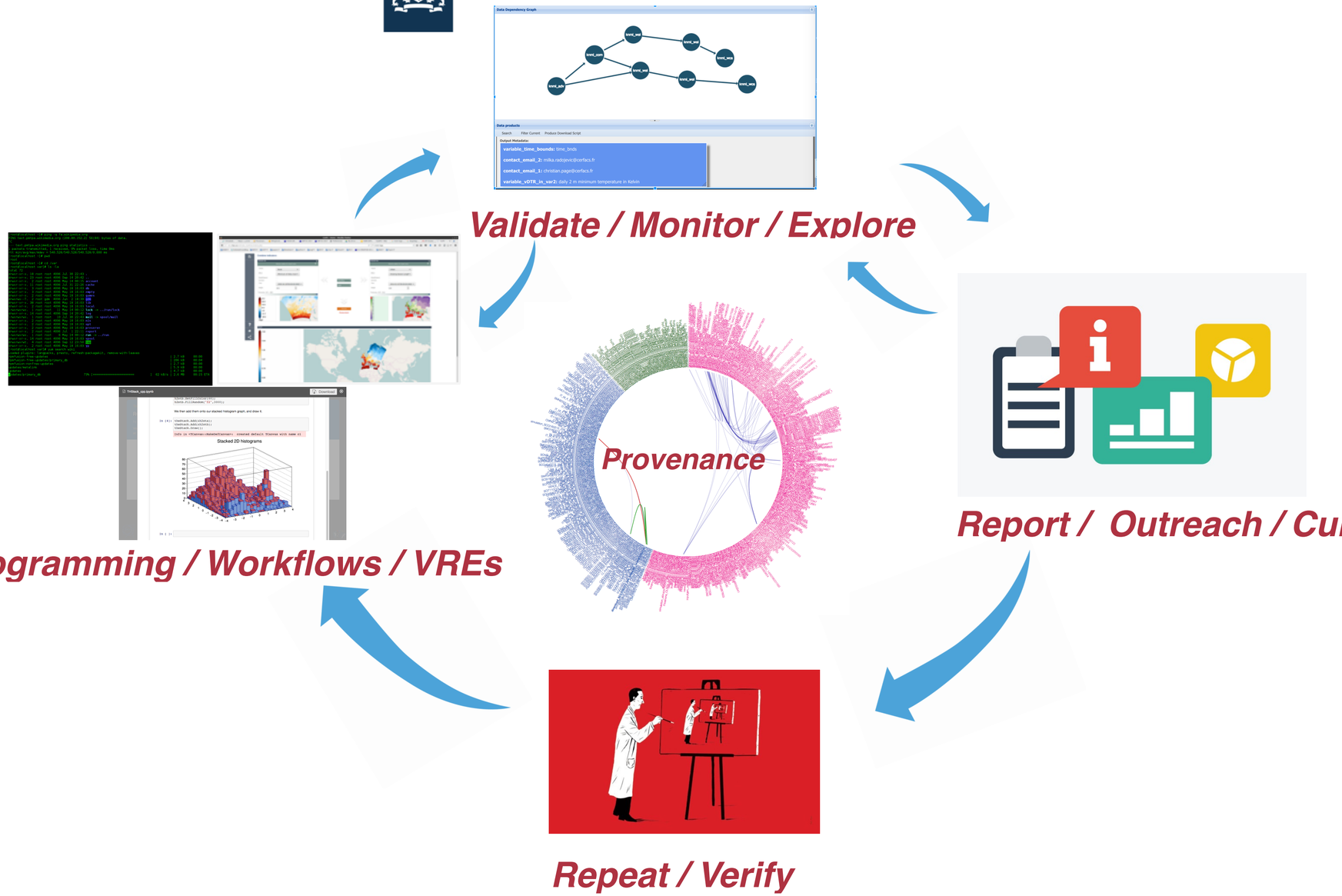
Programming / Workflows / VREs

Report / Outreach / Curate

Repeat / Verify

The Research Cycle(s)

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



Data Lineage
What and Challenges

What

- **Data's origins,** what happens to it and where it **moves over time**
- **It may include technical metadata:** quality test results, reference values.
- **Ability to trace errors** back to the root cause.
- **Its scope** determines the **volume of metadata required**.
- **Integrated** in **workflow systems** to trace the data flow/movement via various changes.

Challenges

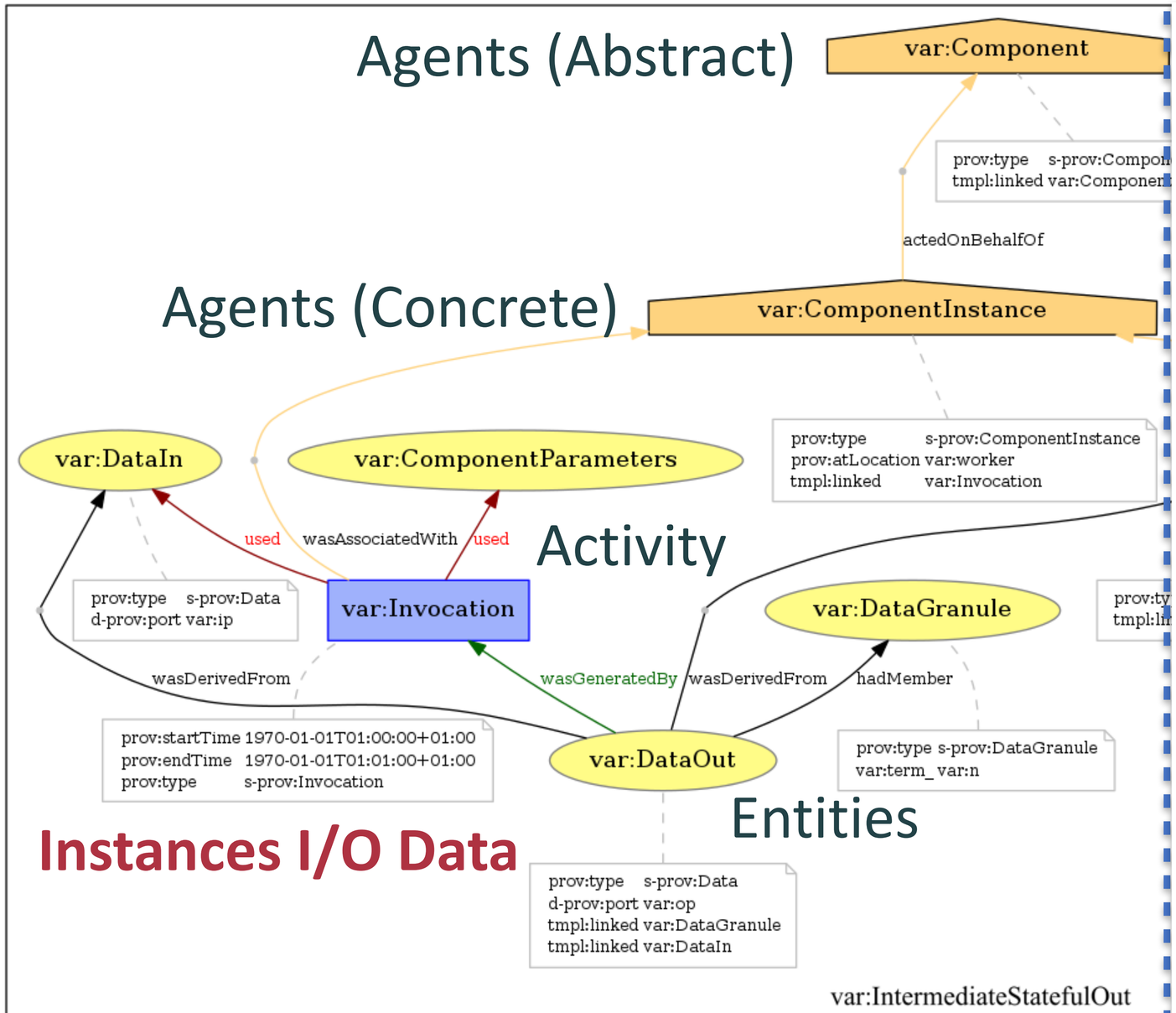
- **Relevance and Granularity:** In Data-Intensive workflows, provenance information is sometimes **too coarse or too detailed. How about domain properties?**

- **Precision:** Lack in precision in describing data derivations could make **traceability of results and understanding** of the method's behaviour **ineffective**.

Data-Intensive Workflow

Model for Lineage and Stateful Patterns (S-PROV, built on ProvONE)

http://project-dare.eu



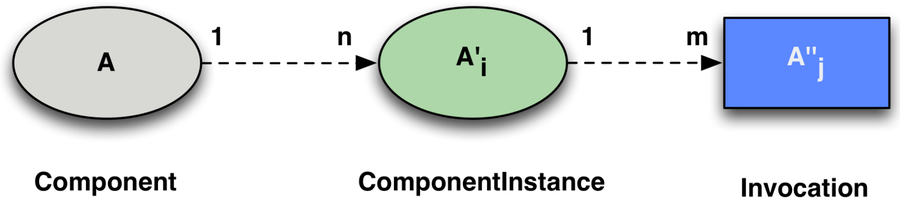
Components

- Define the structural elements in a workflow spec.
- Associated to **Programs in a particular WorkflowExecution**

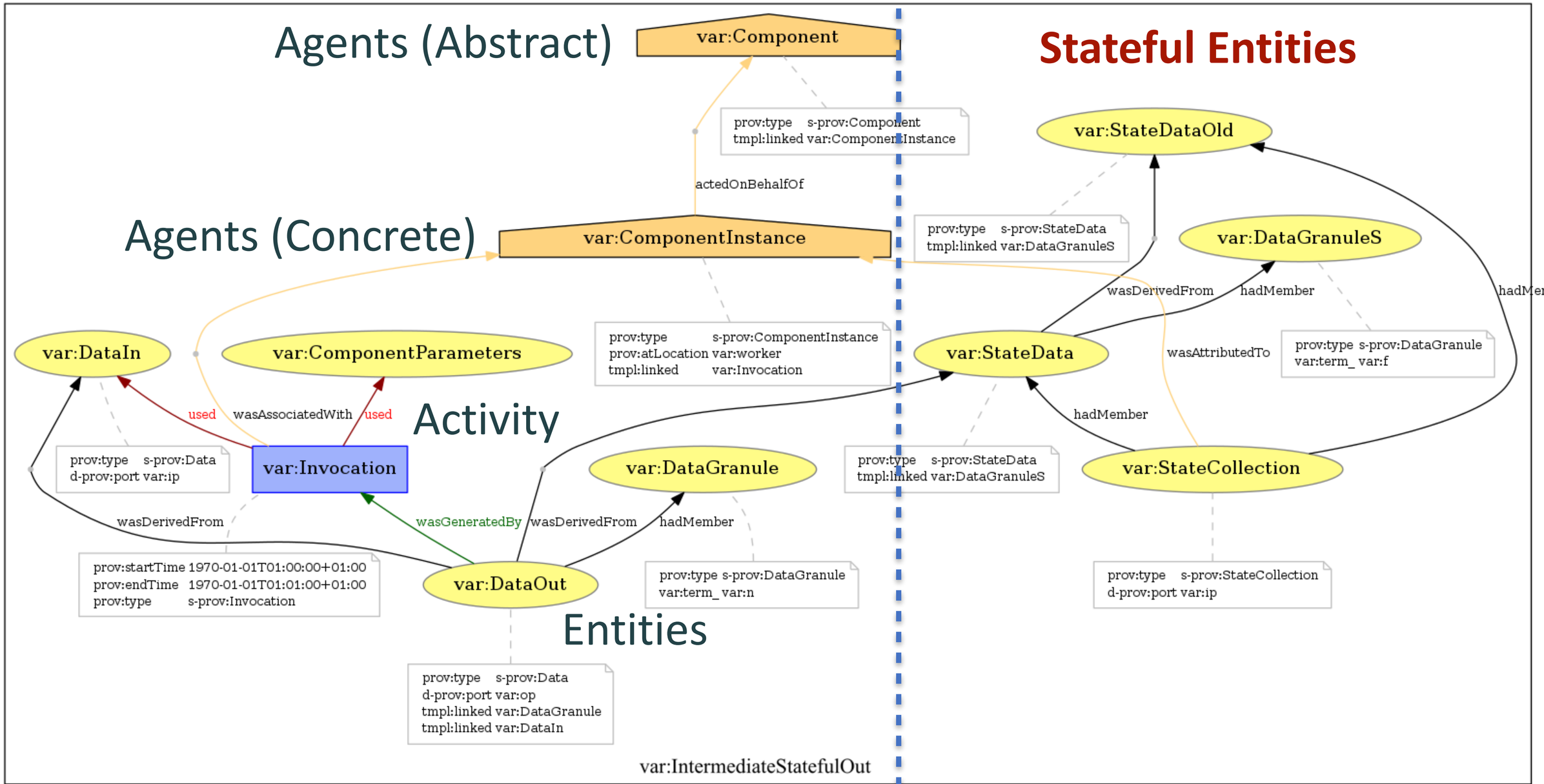
Component Instances

- Self-contained, concurrently interacting
- **Execute Programs** on behalf of a **Component**
- **Iterate on incoming data**
- **Can Change Dynamically**
- **Access Internal State -** Accumulations, Grouping…

DI Workflow Components and distributed instances

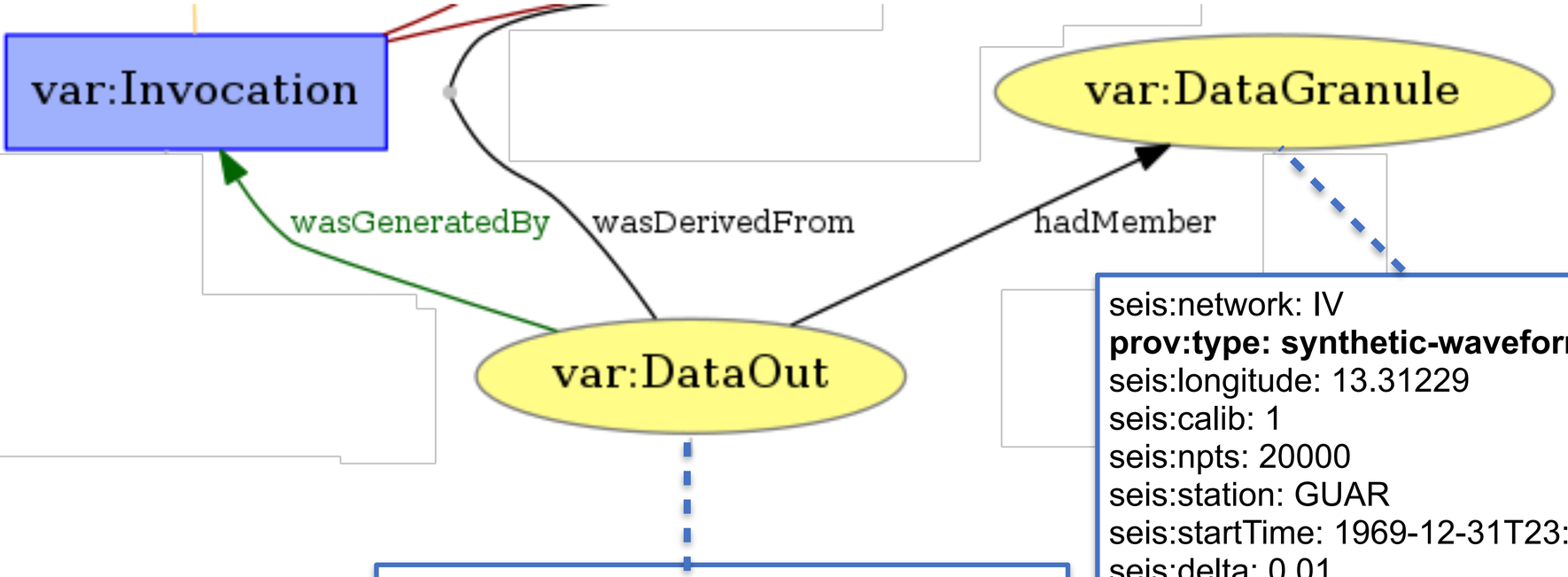


Workflow's Lineage and Stateful Behaviours (S-PROV)



http://project-dare.eu

Contextual Metadata
Data Collections and Granules



location:file://IV.GUAR.HXE.acceleration.png
format:image/png
port:output
id:063-29599-adde4f14-7c01-11e

Common Baseline
For output data collections

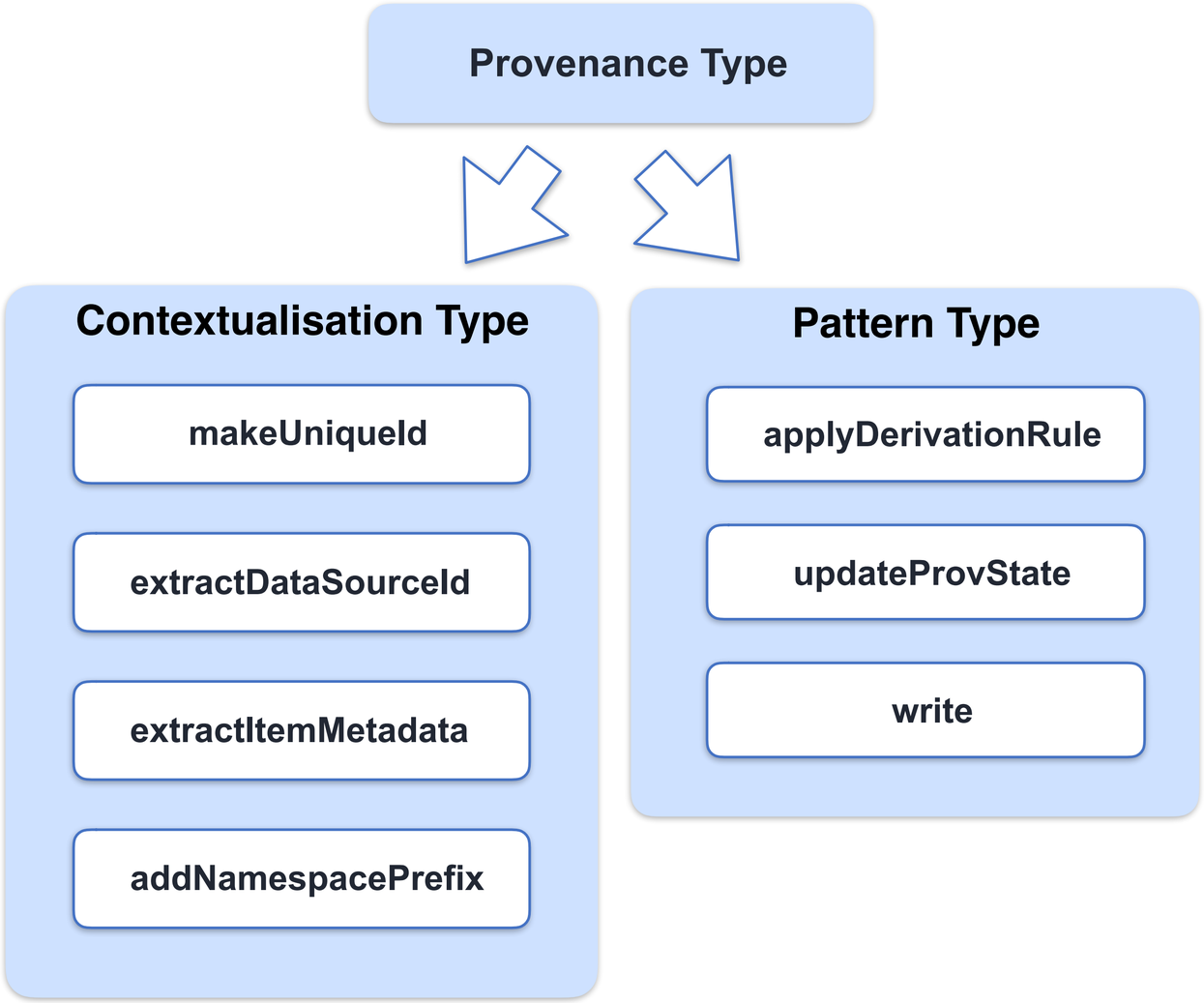
seis:network: IV
prov:type: synthetic-waveform
seis:longitude: 13.31229
seis:calib: 1
seis:npts: 20000
seis:station: GUAR
seis:startTime: 1969-12-31T23:59:57.997502Z
seis:delta: 0.01
seis:latitude: 41.7945
seis:sampling_rate: 100
seis:endTime: 1970-01-01T00:03:17.987502Z
sensor_type: acceleration (user defined)
seis:channel: HXE

Domain properties
User's Context
(e.g. Seismology)

Provenance Type

Augments the behaviour of the WF Components with the capabilities that deliver provenance data.

Same component can be extended with different types depending from requirements (Tailoring)



Provenance Type

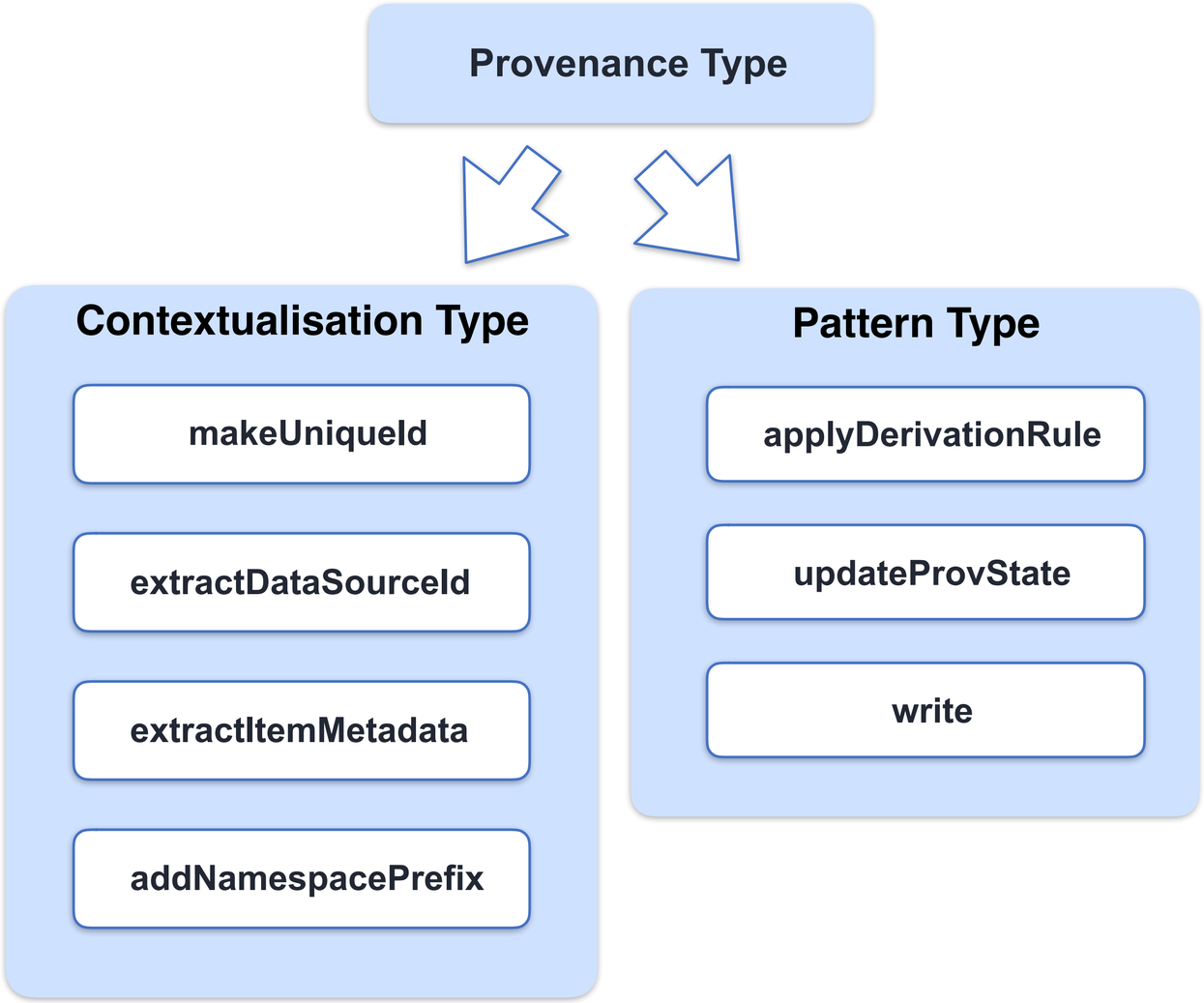
Augments the behaviour of the WF Components with the capabilities that deliver provenance data.

Same component can be extended with different types depending from requirements (Tailoring)

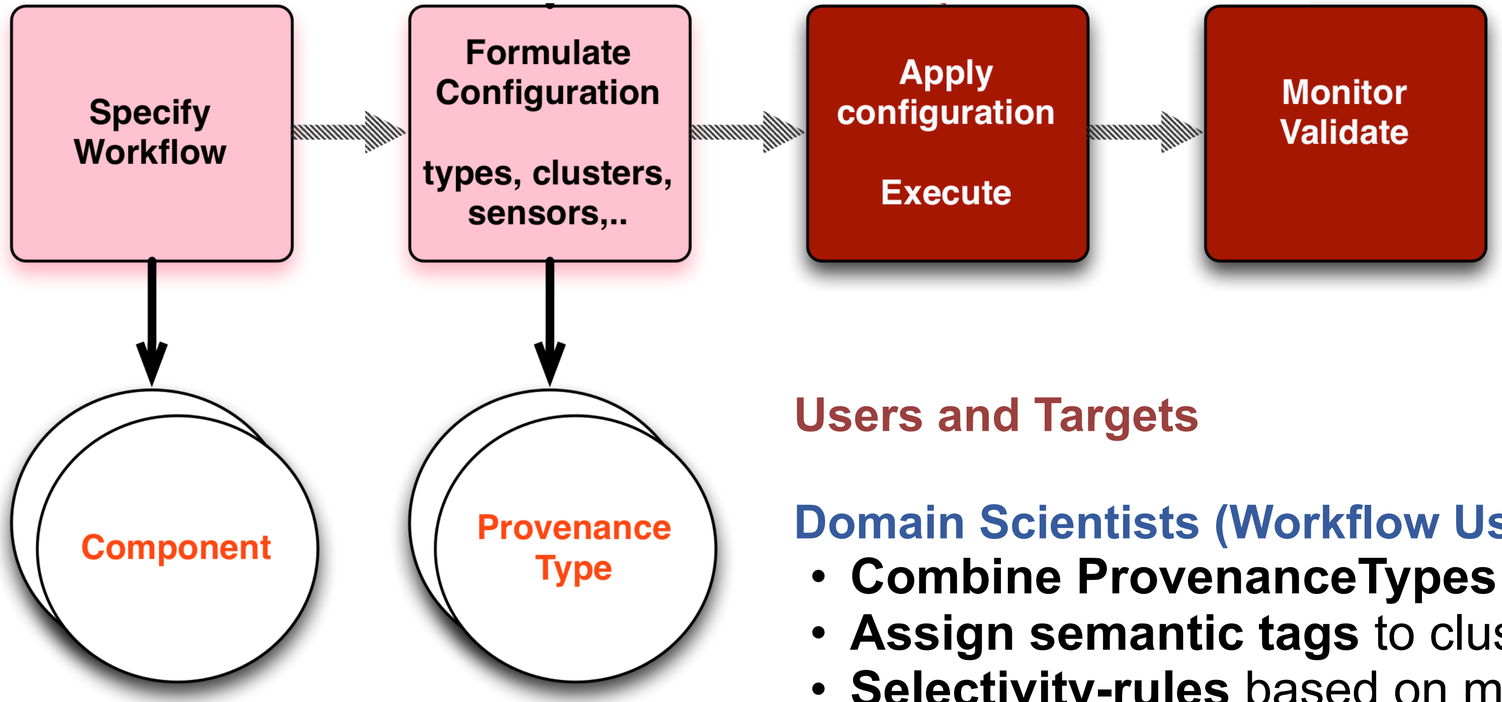
Users and Targets

Research Developers

- ***Contextualisation types*** to extract domain specific data properties
- ***Patterns types*** to capture Complex I/O and *stateful* behaviours (Lineage Precision)
- **Ad-hoc inline metadata injection** for application specific metadata



Provenance Configuration



Users and Targets

Domain Scientists (Workflow Users)

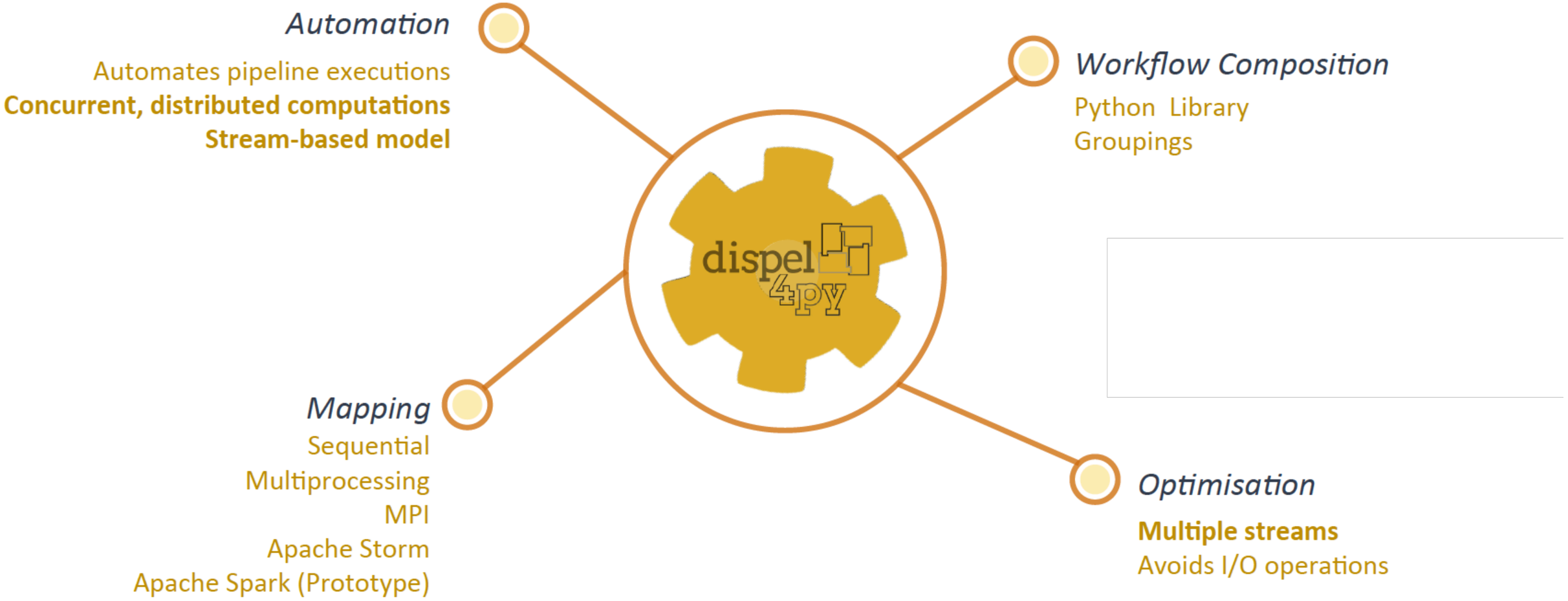
- **Combine ProvenanceTypes** incrementally.
- **Assign semantic tags** to clusters of Components and Workflows.
- **Selectivity-rules** based on metadata value ranges to narrow the focus of the lineage onto relevant situations.

System Managers

- **Tune the impact of provenance** on the infrastructure (real-time systems)
- **Mining for resource planning**

Data-Intensive Workflow

DARE http://project-dare.eu



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

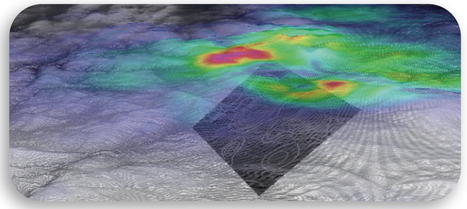
https://gitlab.com/project-dare/dispel4py

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

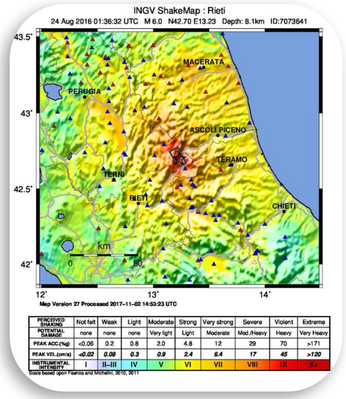
Gather **observed data**

Waveform Preprocessing

Peak Ground Motion Parameters

Compare/integrate synthetic and observed ground motion data

Store data, metadata, provenance



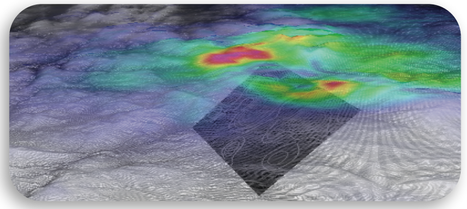
Test Case:
Seismic Rapid Assessment

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**

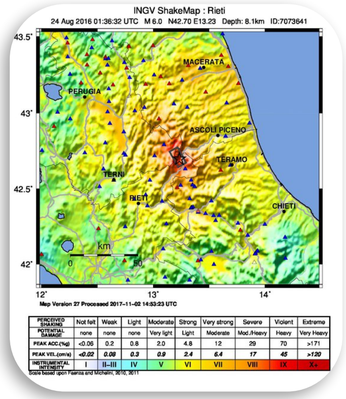
Waveform Preprocessing

Peak Ground Motion Parameters

Data Analysis

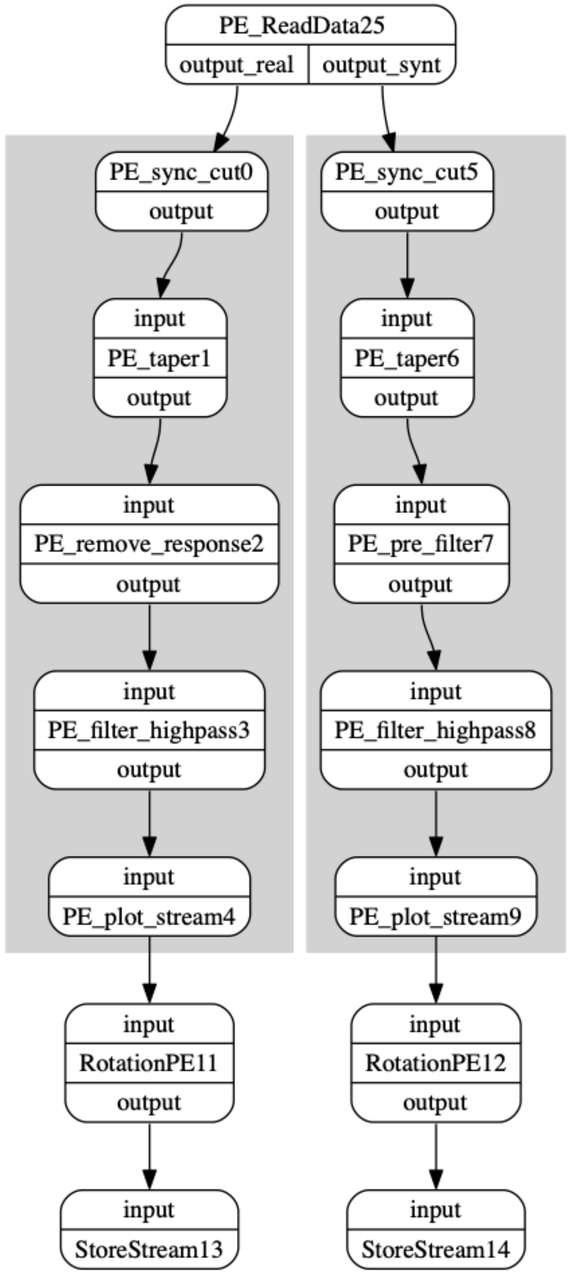
Compare/integrate synthetic and observed ground motion data

Store data, metadata, provenance



Specify Workflow

Waveform Preprocessing



pipeline JSON Description (eg. from file)

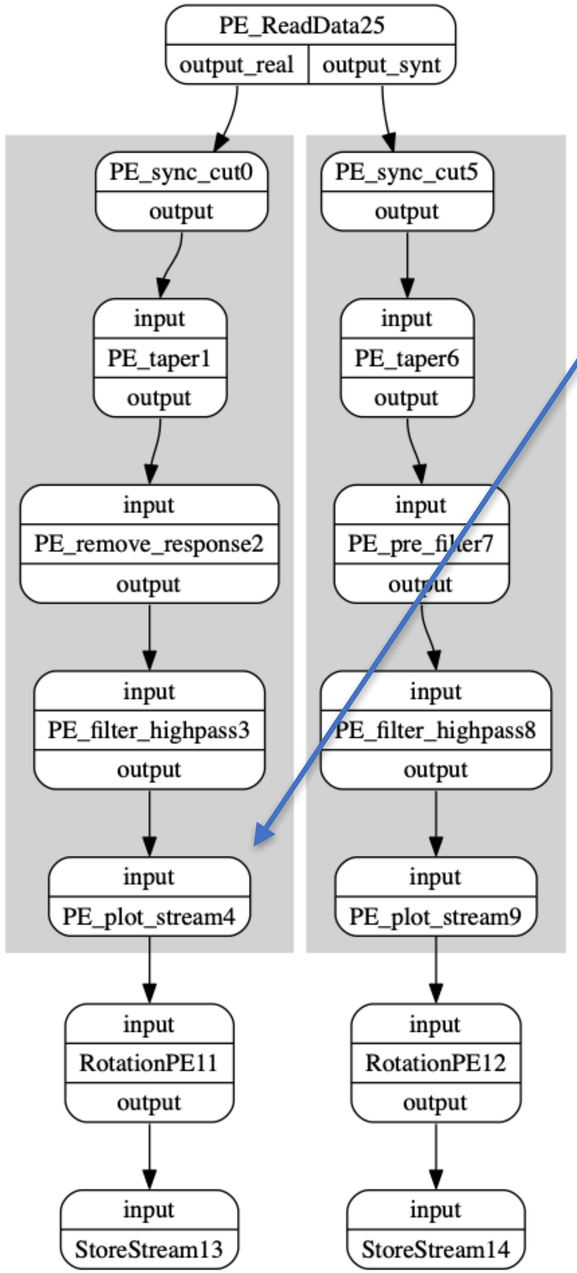
Manual Extensions

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)
```

Inline metadata injection

Waveform Preprocessing



pipeline JSON Description (eg. from file)

Manual Extensions

Functions encoded in Python
User Defined Metadata injection into Lineage traces

```
def plot_stream(stream, output_dir, tag):
    stats = stream[0].stats
    filename = "%s.%s.%s.%s.png" %
        (stats['network'], stats['station'],
         stats['channel'], tag)

    path = os.environ['STAGED_DATA'] + '/' + output_dir
    dest = os.path.join(path, filename)
    stream.plot(outfile=dest)

    prov = {'location': "file://" + socket.gethostname()+"/"+dest,
            'format': 'image/png',
            'metadata': {'origin': tag}}

    return {'_d4p_prov': prov, '_d4p_data': stream}
```

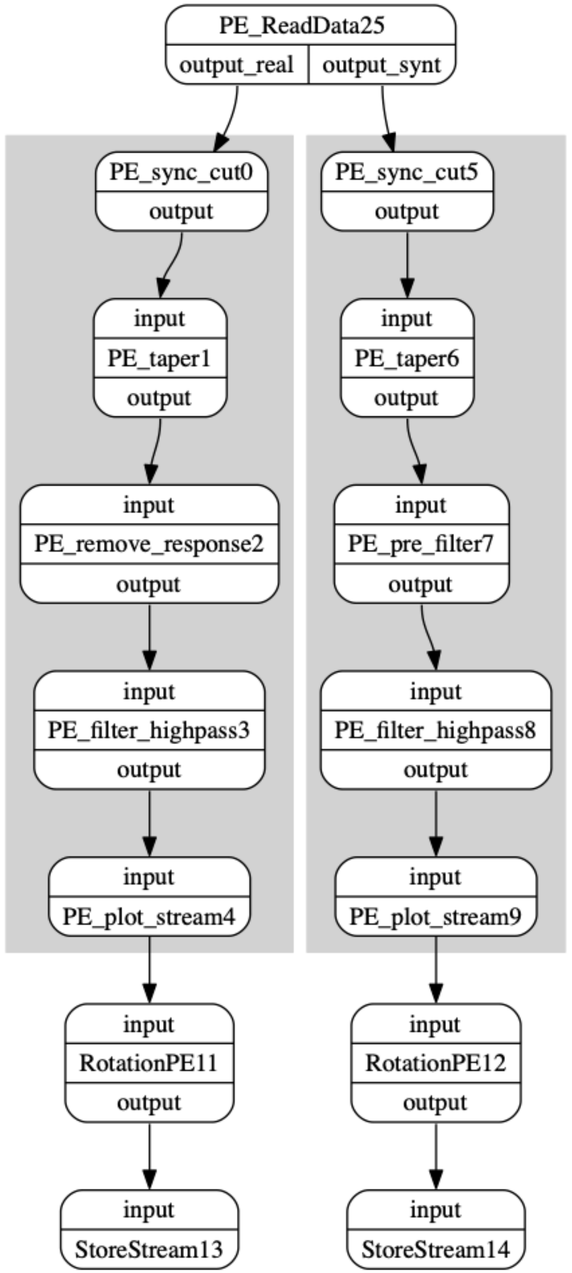
dispel4py

User Defined Metadata

Configuration - Contextualisation Types

DARE http://project-dare.eu

Waveform Preprocessing



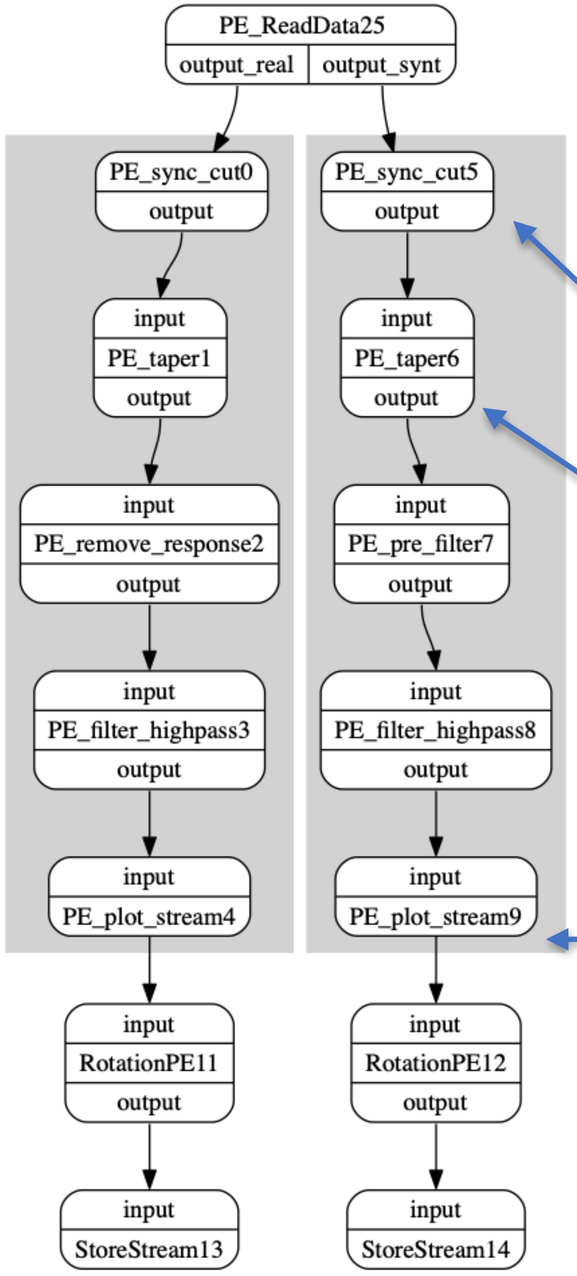
Configuration Profile in JSON with Provenance Types

```
{
    'provone:User': "aspinuso",
    's-prov:description' : "provdemo",
    's-prov:workflowName': "waveform preprocessing pipeline",
    's-prov:workflowType': "seis:preprocessing",
    's-prov:WFExecutionInputs':  [{...}],
    's-prov:save-mode'    : 'service',
    's-prov:WFExecutionInputs':  [{...}],
    # defines the Provenance Types and Provenance Clusters for the Workflow's Components
     's-prov:componentsType' :
            {'s-prov:componentsType' :
                 {'PE_ReadData':       { 's-prov:type':['SeismoType'],
                                         's-prov:prov-cluster':'seis:DataHandler'},

                  'PE_taper':          { 's-prov:type':['SeismoType'],
                                         's-prov:prov-cluster':'seis:Processor'},

                  'PE_remove_response':{ 's-prov:type':['SeismoType'],
                                         's-prov:prov-cluster':'seis:Processor'},

                  'PE_plot_stream':     {'s-prov:type':['SeismoType']
                                        's-prov:prov-cluster':'seis:Processor'},

                  'StoreStream':       {'s-prov:type':['SeismoType'],
                                        's-prov:prov-cluster':'seis:DataHandler'}}}
```

Configuration - Contextualisation Types

DARE http://project-dare.eu

Waveform Preprocessing



Configuration Profile in JSON with Provenance Types

```
{
    'provone:User': "aspinuso",
    's-prov:description' : "provdemo",
    's-prov:workflowName': "waveform preprocessing pipeline",
    's-prov:workflowType': "seis:preprocessing",
    's-prov:WFExecutionInputs':  [{...}],
    's-prov:save-mode'    : 'service',
    's-prov:WFExecutionInputs':  [{...}],
   # defines the Provenance Types and Provenance Clusters for the Workflow's Components
     's-prov:componentsType' :
              { 's-prov:componentsType' :
                   {'PE_ReadData':       { 's-prov:type':['SeismoType'],
                                           's-prov:prov-cluster':'seis:DataHandler'},

                    'PE_taper':          { 's-prov:type':['SeismoType'],
                                           's-prov:prov-cluster':'seis:Processor'},

                    'PE_remove_response':{ 's-prov:type':['SeismoType'],
                                           's-prov:prov-cluster':'seis:Processor'},

                    'PE_plot_stream':     {'s-prov:type':['SeismoType']
                                           's-prov:prov-cluster':'seis:Processor'}
```

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

delta: 0.01

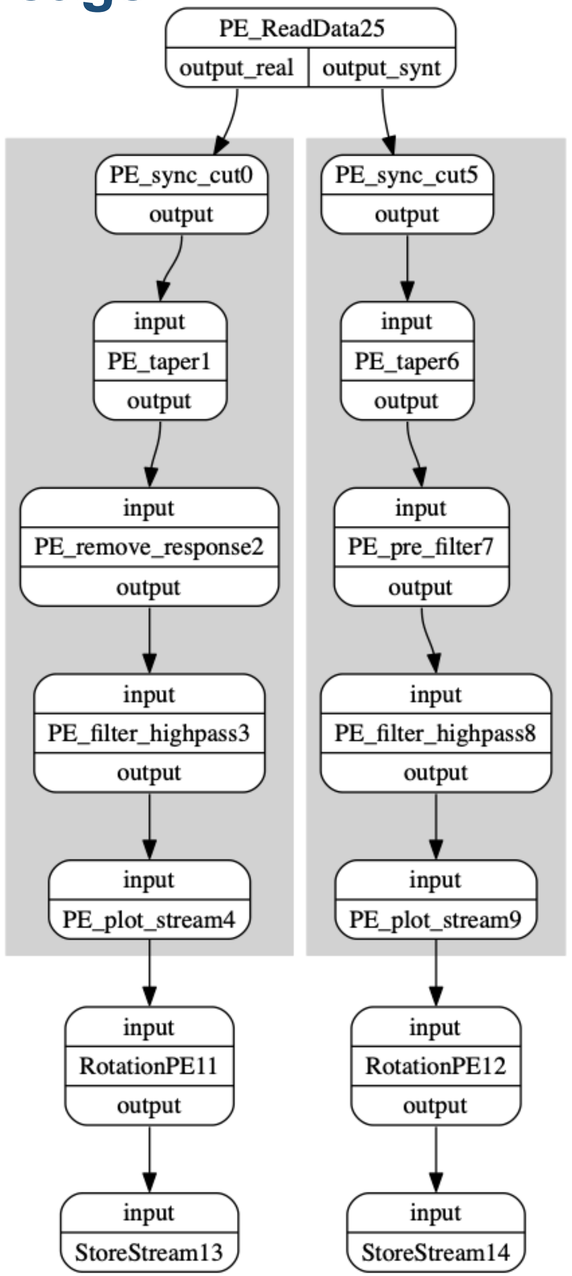
calib: 1

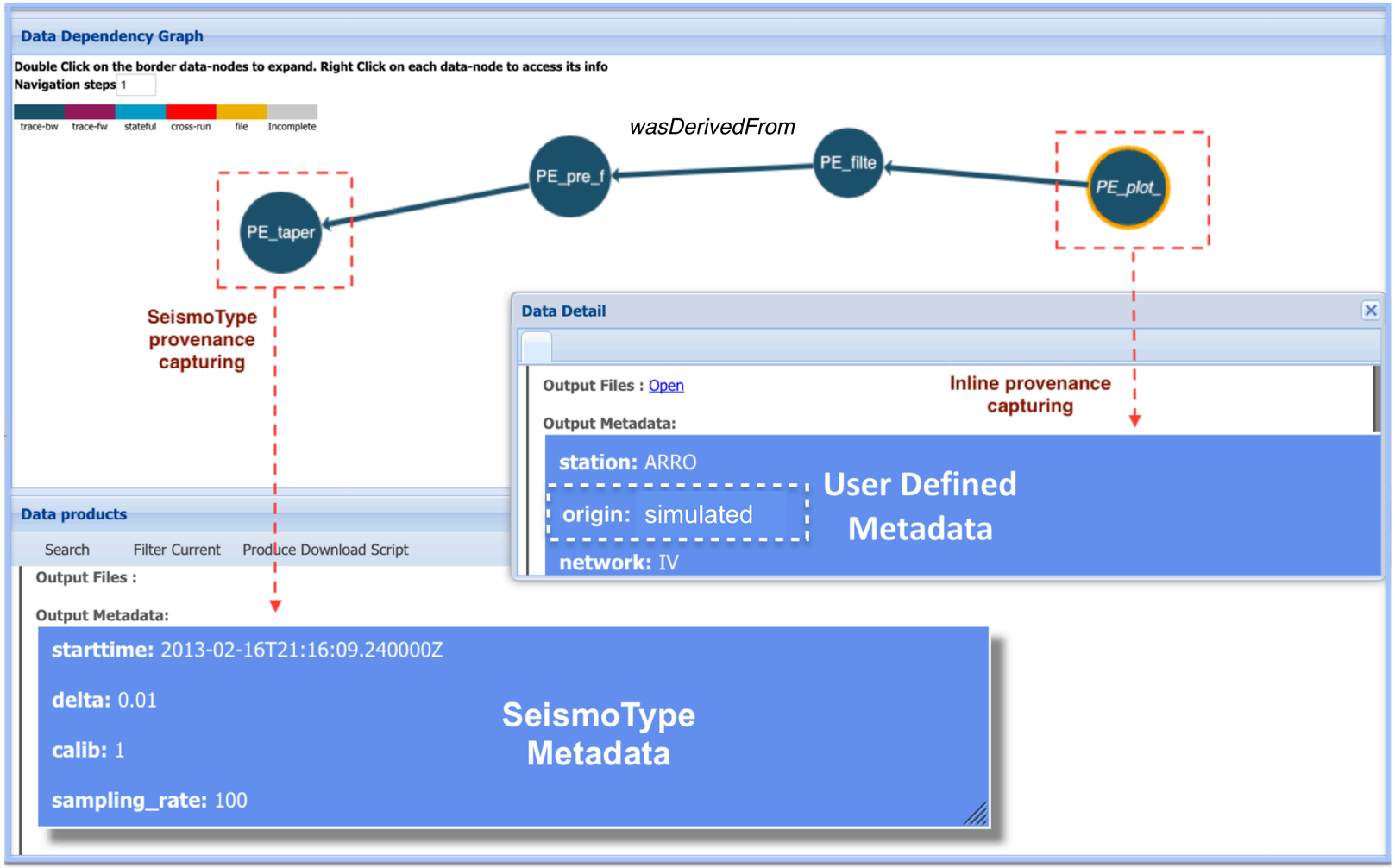
sampling_rate: 100

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

Monitor, search and analyse results through lineage

Waveform Preprocessing



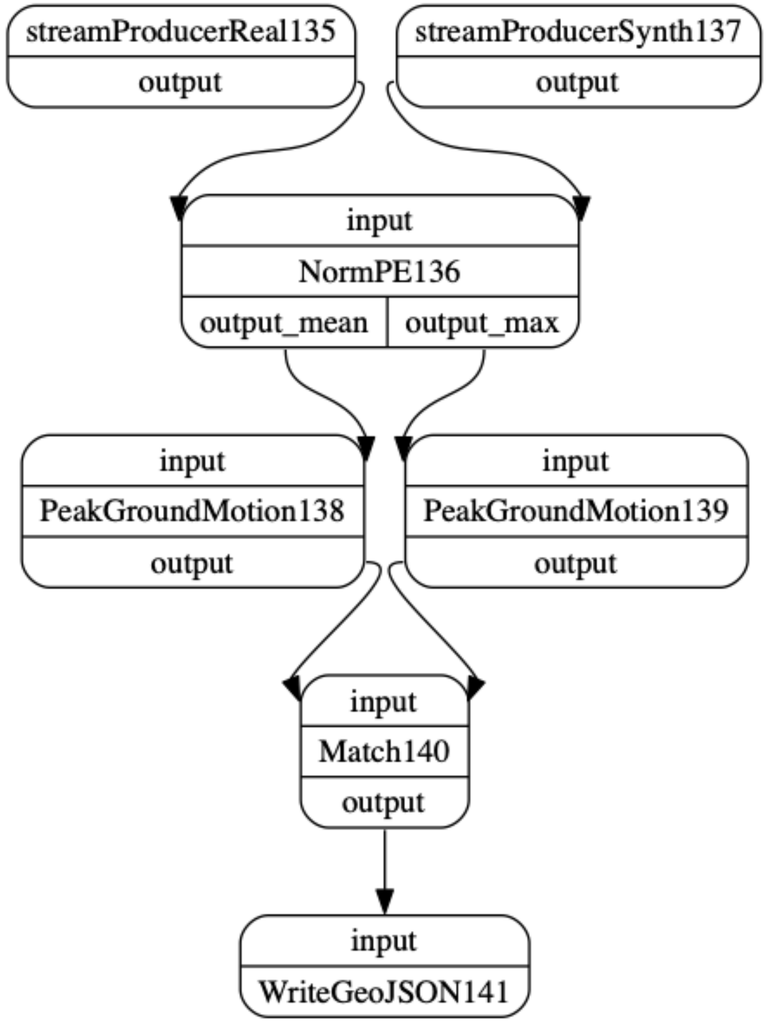


S-ProvFlow:https://gitlab.com/project-dare/s-ProvFlow

Configuration - Lineage Pattern Types

DARE http://project-dare.eu

Peak Ground Motion

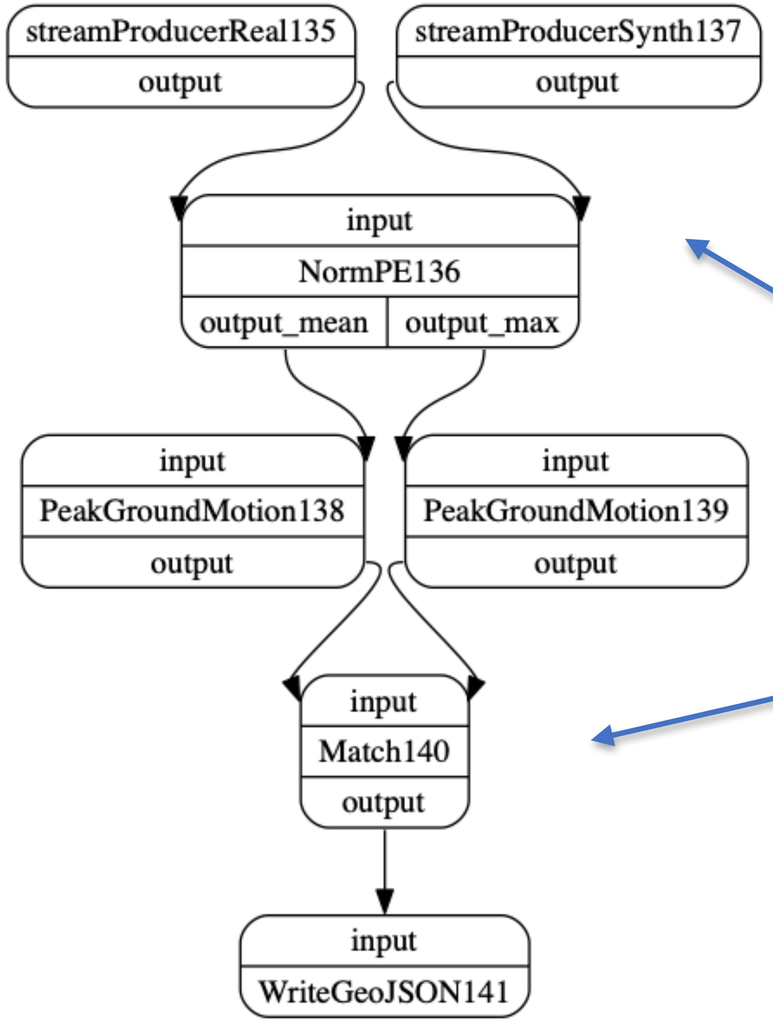



```
{
   'provone:User': "aspinuso",
   's-prov:description' : "provdemo",
   's-prov:workflowName': "Peak Ground Motion Workflow",
   's-prov:workflowType': "seis:PGMParameters",
   's-prov:save-mode'   : 'service'          ,
   's-prov:WFExecutionInputs':  [{...}],
   # defines the Provenance Types and Provenance Clusters for the Workflow's Components
    's-prov:componentsType' :
            {'s-prov:componentsType' :
                 {'streamProducerReal':{'s-prov:type':['SeismoType'],
                                        's-prov:prov-cluster':'seis:DataHandler'},

                  'streamProducerSynth':{'s-prov:type':'[SeismoType'],
                                         's-prov:prov-cluster':'seis:DataHandler'},

                  'Match':              {'s-prov:type':['ASTGrouped'],
                                         's-prov:prov-cluster':'seis:DataHandler'}}}
```

Configuration - Lineage Pattern Types

DARE http://project-dare.eu

Peak Ground Motion



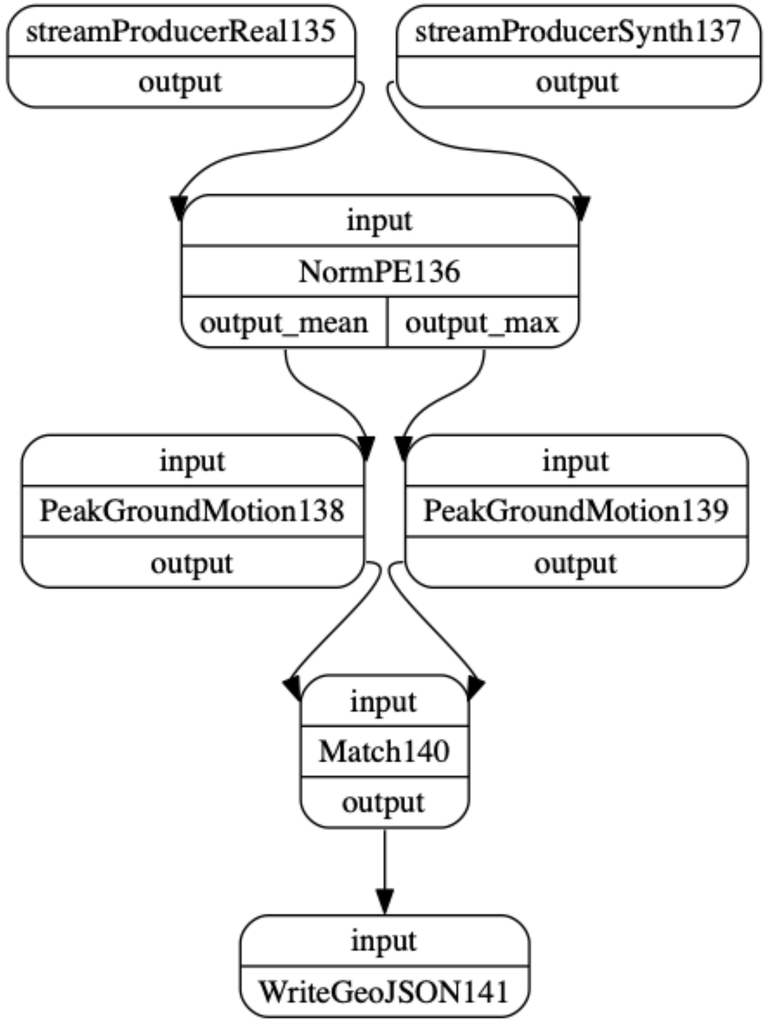

```
{
    'provone:User': "aspinuso",
    's-prov:description' : "provdemo",
    's-prov:workflowName': "Peak Ground Motion Workflow",
    's-prov:workflowType': "seis:PGMParameters",
    's-prov:save-mode'   : 'service'          ,
    's-prov:WFExecutionInputs':  [{...}],
   # defines the Provenance Types and Provenance Clusters for the Workflow's Components
     's-prov:componentsType' :
             {'s-prov:componentsType' :
                 {'streamProducerReal':{'s-prov:type':['SeismoType'],
                                        's-prov:prov-cluster':'seis:DataHandler'},

                  'streamProducerSynth':{'s-prov:type':'[SeismoType'],
                                         's-prov:prov-cluster':'seis:DataHandler'},

                  'Match':              {'s-prov:type':['ASTGrouped'],
                                         's-prov:prov-cluster' : 'seis:DataHandler'}}}
```

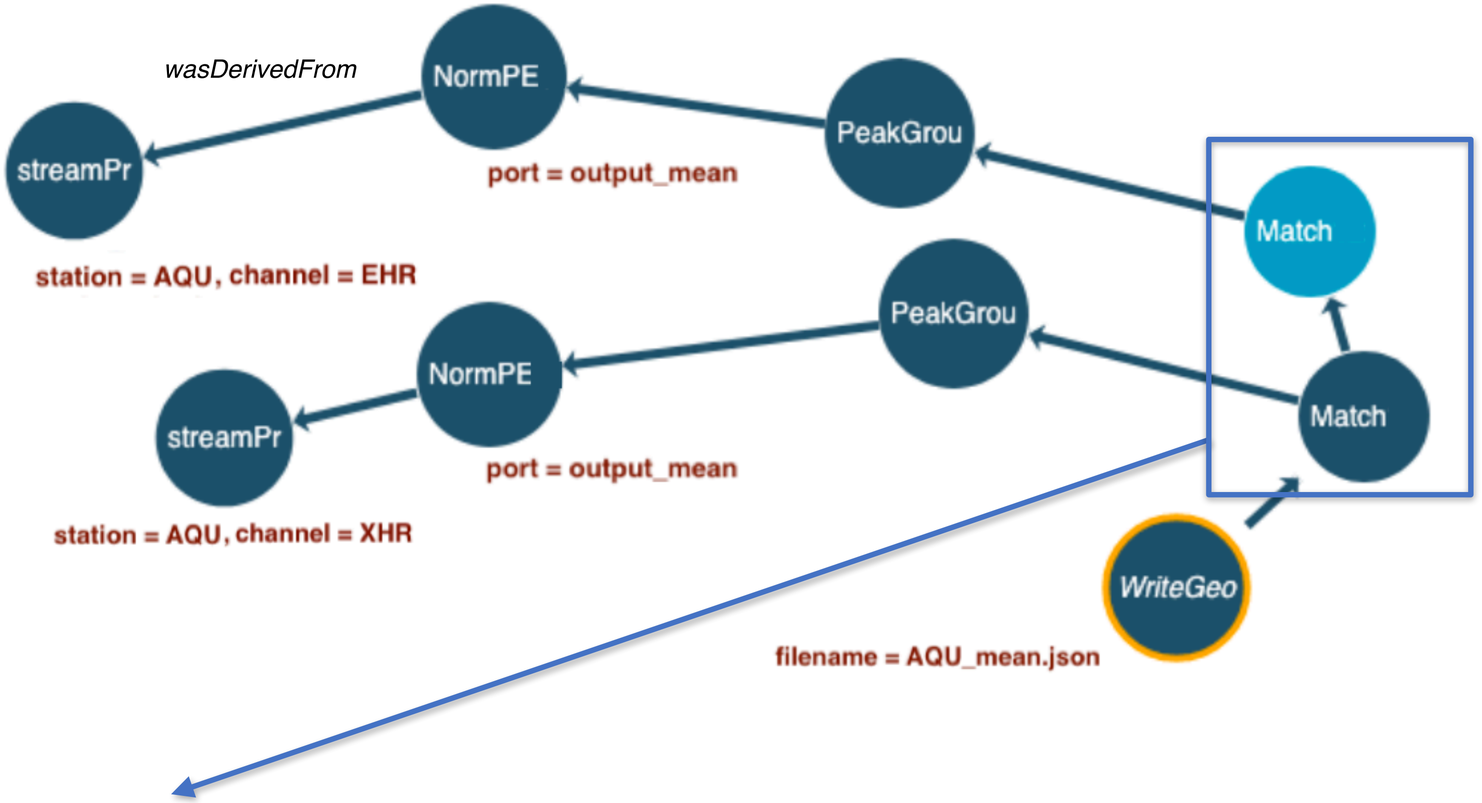
***ProvenanceType* for a Stateful Lineage Pattern**

Its instances receive and combines data with specific "metadata"values eg. station codes

Lineage Precision in stateful operators

Peak Ground Motion





StateDerivation, between the **Match PE output and data preserved in its internal state.**

Linking executions and semantic tagging
Exploring the Experiments' space

DARE http://project-dare.eu

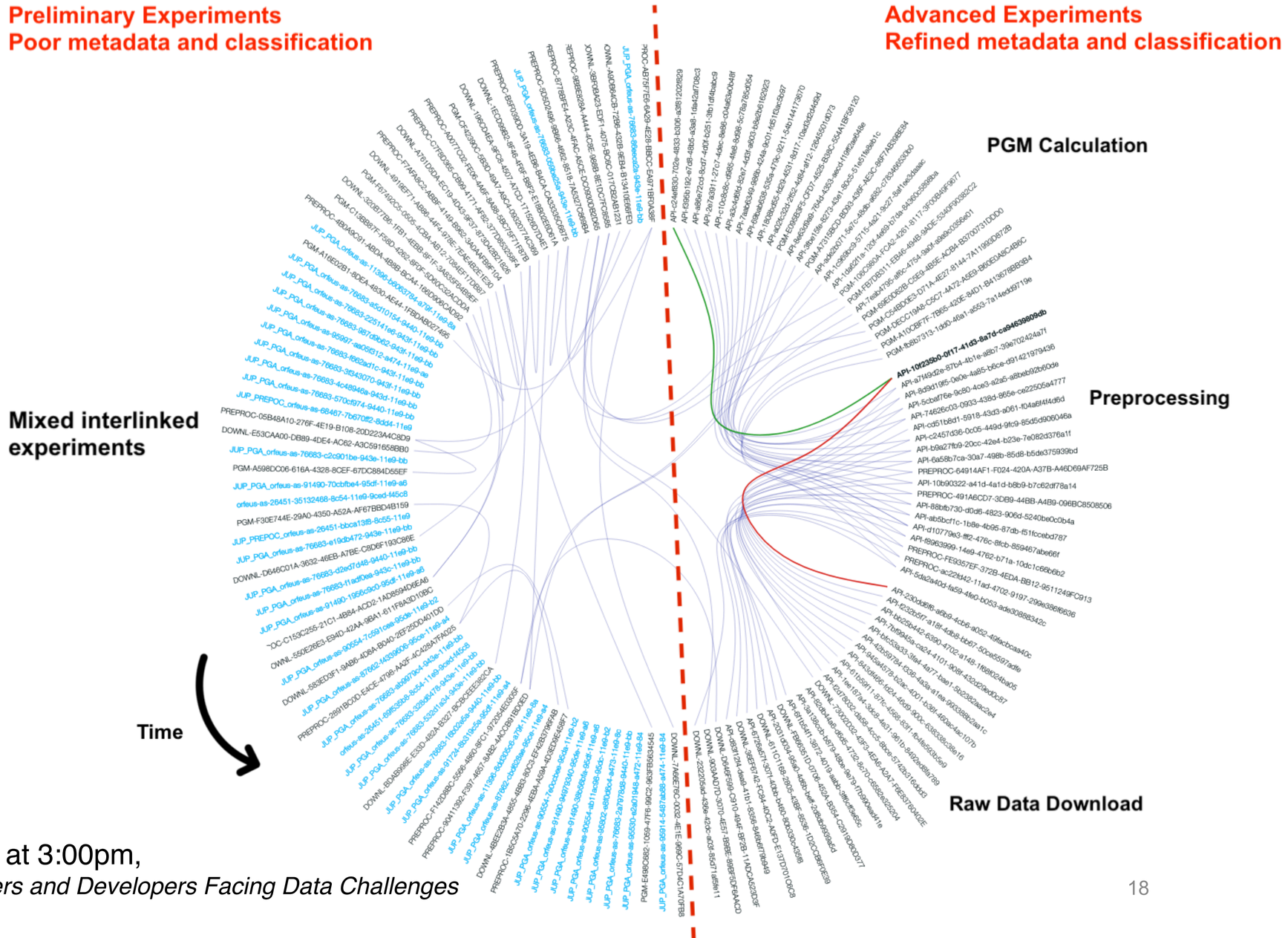
Visual analytics of data reuse between the workflows of the RA use case

Runs selected among those using the same station codes. (Contextual metadata)

Vertices are workflows execution ids **colour-coded by user**.

Edges represent data flows. Red and green edges for data input and output, respectively.

Right half: Facilitating better descriptions yields a improved management of the results



More on **Wednesday**, September 25 - Session at 3:00pm,
M. Atkinson et al. *Comprehensible Control for Researchers and Developers Facing Data Challenges*

Conclusions & Future Work

- **Balanced automation and *Active* human contribution** in provenance capturing in Data-Intensive workflows
- **A conceptual design** based on reusable and combinable *Provenance Types* that lead to the Provenance Configuration
- **Provenance model** S-PROV, that accommodates complex lineage patterns.
- **Improved utility of the traces,** through versatile and *Active* participation of the domain experts and developers yielding
- **Services and tools** developed around our framework to demonstrate its effects (S-ProvFlow)
- **Coming Next!** Combination of model and tools to tackle the challenges of *lineage exploration for steering actions.*

Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Infrastructuur en Milieu

DARE http://project-dare.eu

Thanks!