Training Provenance Tracing in ENVRI RI's
A Provenance Demonstrator

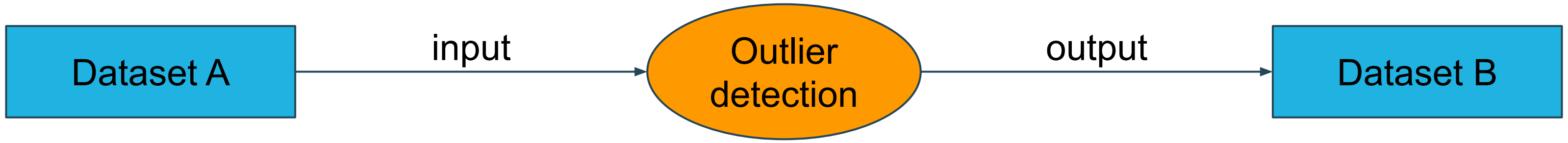
Markus Stocker[iD]
TIB - Leibniz Information Centre for Science and Technology

December 9, 2020

ENVRI-FAIR has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 824068

Introduction

- Whenever an agent performs an activity on an artefact there is provenance
- Agent can be a person or a machine
- An activity can be many things, physical or computational
- Artefact is typically data but can be a physical object (e.g., sample)
- Example

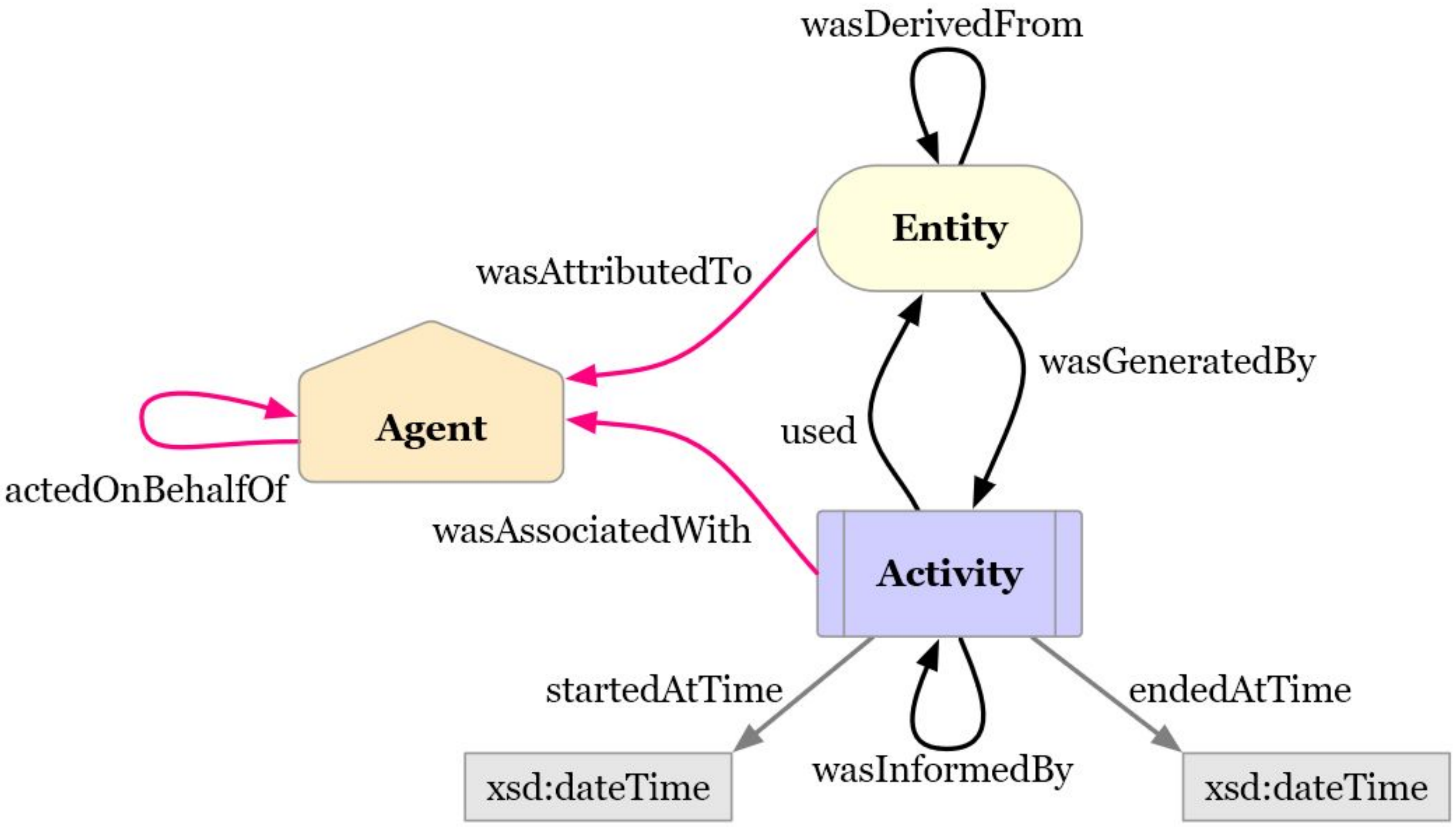


Introduction

- Provenance metadata can be encoded using various technologies
- One is using the W3C recommended Provenance Ontology (PROV-O)
- This typically (but not necessarily) also implies RDF and triple stores
- For PROV-O there is also reasonable tool support, e.g. in Python

https://www.w3.org/TR/prov-o/
https://pypi.org/project/prov/

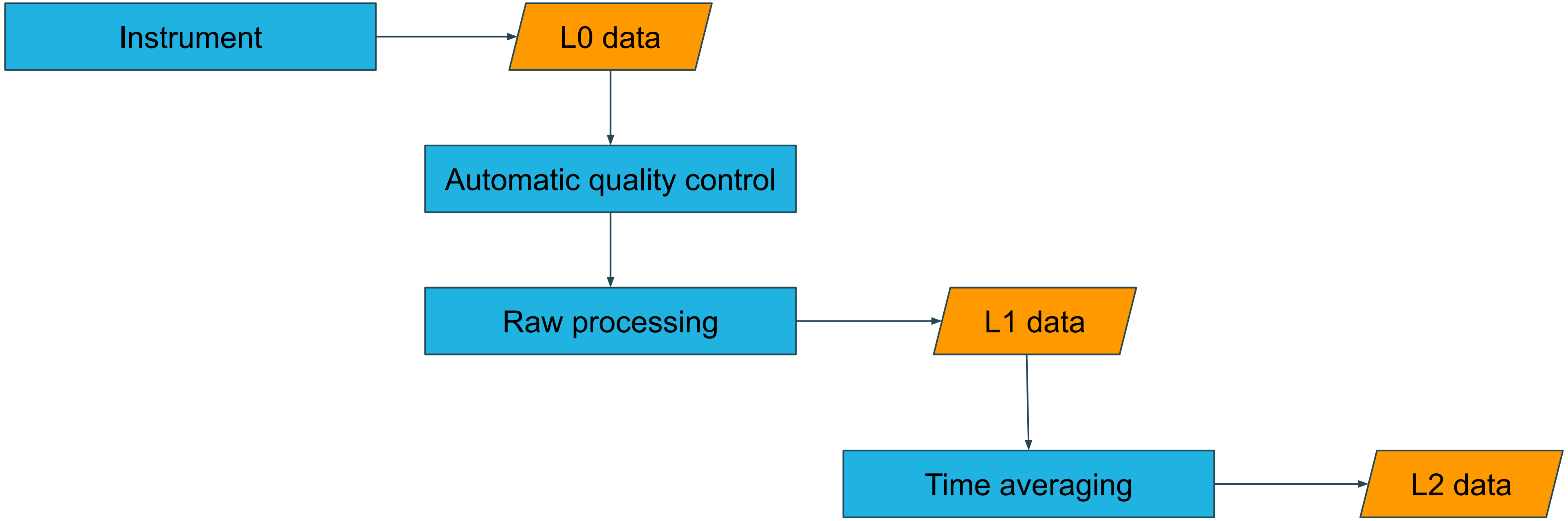
PROV-O core model



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

Demonstrator overview

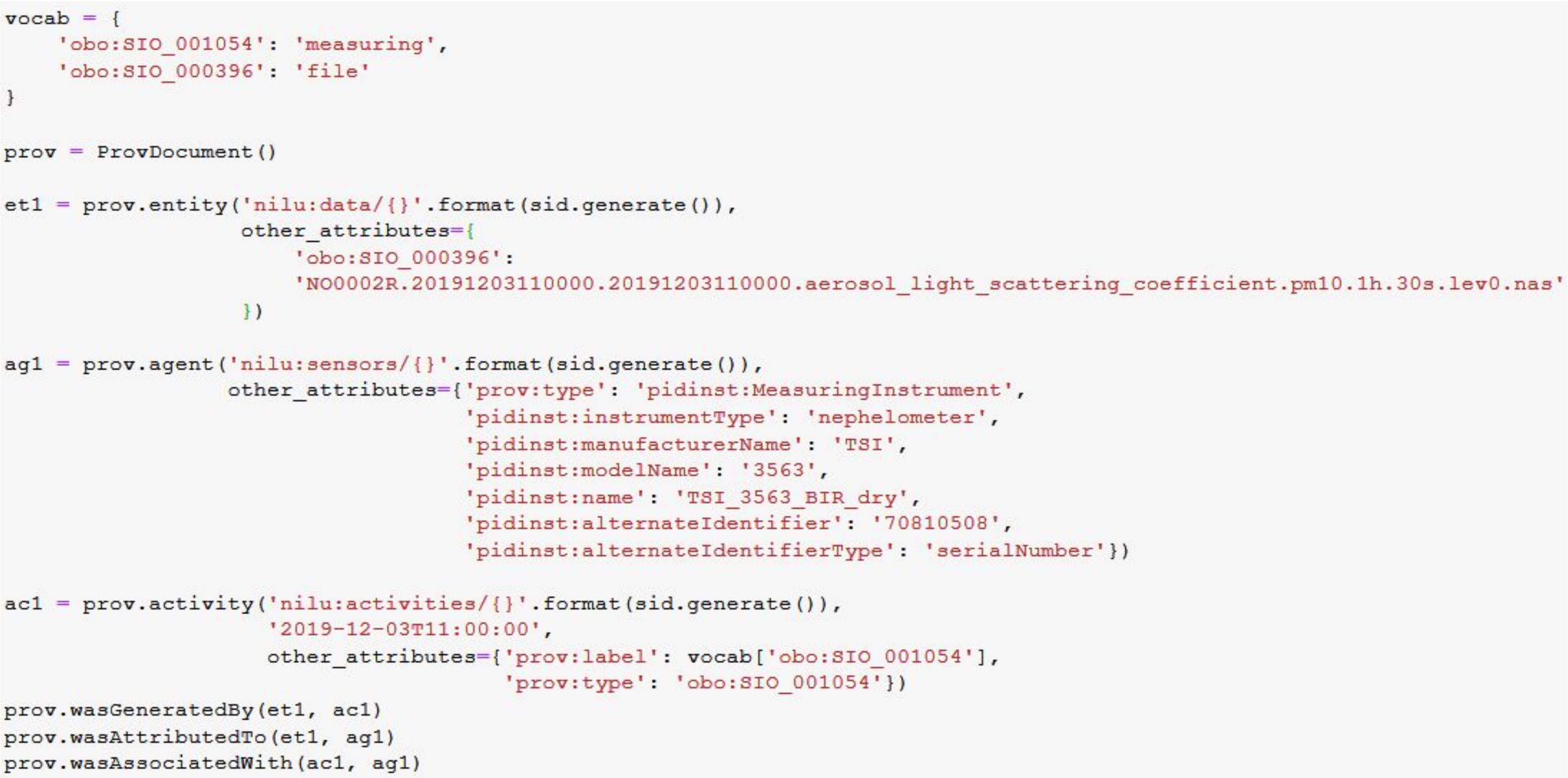
- Processing of online observations at NILU
- Detailed description at https://bit.ly/2FA9Ve4



Demonstrator implementation

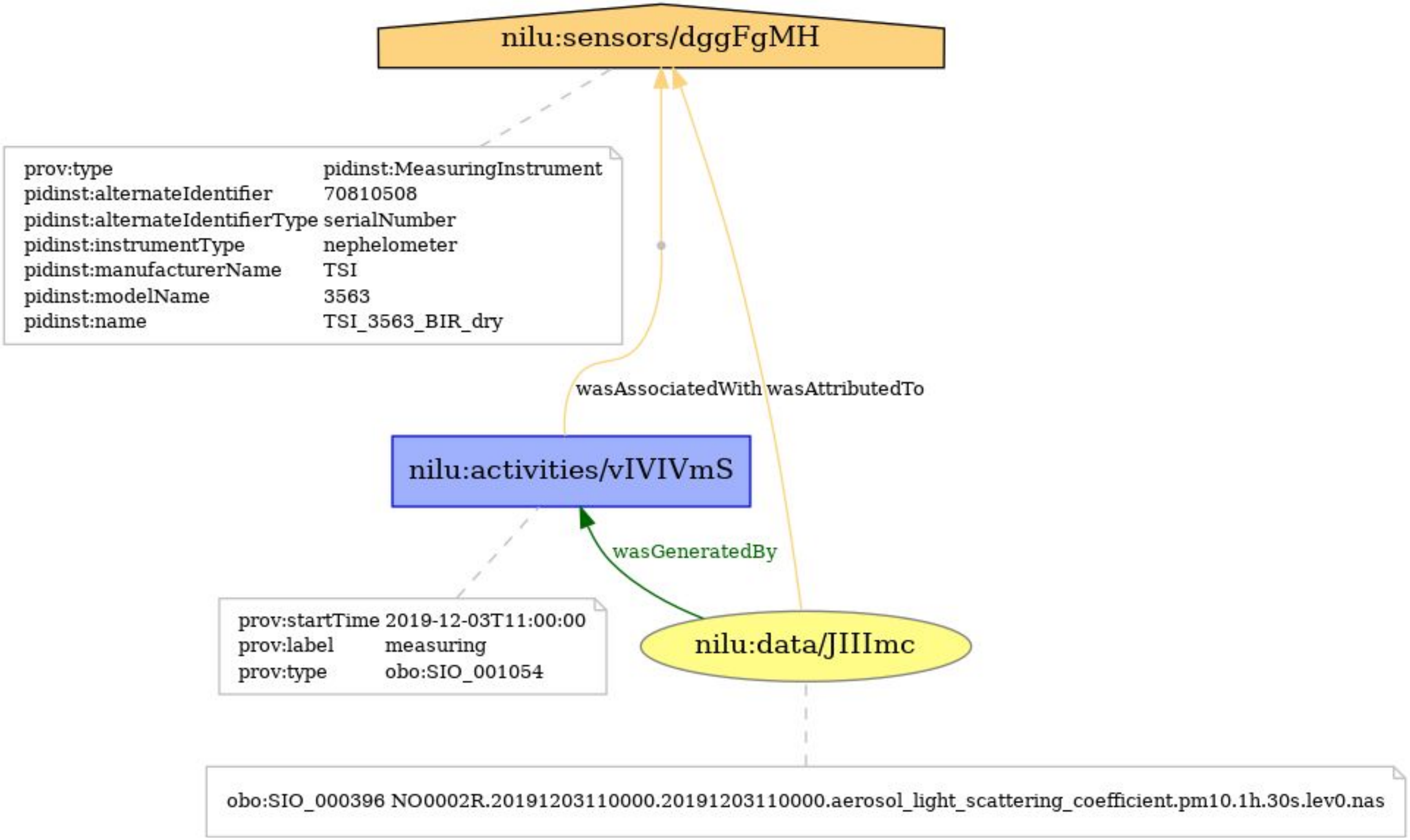
- We use Python and its `prov` package to implement the demonstrator
- The demonstrator is implemented in Jupyter
- The code is available https://github.com/envri-fair/provenance-demonstrator
- We use mybinder for the online execution of the demonstrator
- https://mybinder.org/v2/gh/envri-fair/provenance-demonstrator.git/HEAD

Demonstrator code snippet



```
vocab = {
    'obo:SIO_001054': 'measuring',
    'obo:SIO_000396': 'file'
}

prov = ProvDocument()

et1 = prov.entity('nilu:data/{}'.format(sid.generate()),
                  other_attributes={
                      'obo:SIO_000396':
                      'NO0002R.20191203110000.20191203110000.aerosol_light_scattering_coefficient.pm10.1h.30s.lev0.nas'
                  })

ag1 = prov.agent('nilu:sensors/{}'.format(sid.generate()),
                 other_attributes={'prov:type': 'pidinst:MeasuringInstrument',
                                   'pidinst:instrumentType': 'nephelometer',
                                   'pidinst:manufacturerName': 'TSI',
                                   'pidinst:modelName': '3563',
                                   'pidinst:name': 'TSI_3563_BIR_dry',
                                   'pidinst:alternateIdentifier': '70810508',
                                   'pidinst:alternateIdentifierType': 'serialNumber'})

ac1 = prov.activity('nilu:activities/{}'.format(sid.generate()),
                    '2019-12-03T11:00:00',
                    other_attributes={'prov:label': vocab['obo:SIO_001054'],
                                      'prov:type': 'obo:SIO_001054'})
prov.wasGeneratedBy(et1, ac1)
prov.wasAttributedTo(et1, ag1)
prov.wasAssociatedWith(ac1, ag1)
```

Demonstrator result



nilu:sensors/dggFgMH

prov:type	pidinst:MeasuringInstrument
pidinst:alternateIdentifier	70810508
pidinst:alternateIdentifierType	serialNumber
pidinst:instrumentType	nephelometer
pidinst:manufacturerName	TSI
pidinst:modelName	3563
pidinst:name	TSI_3563_BIR_dry

wasAssociatedWith wasAttributedTo

nilu:activities/vIVIVmS

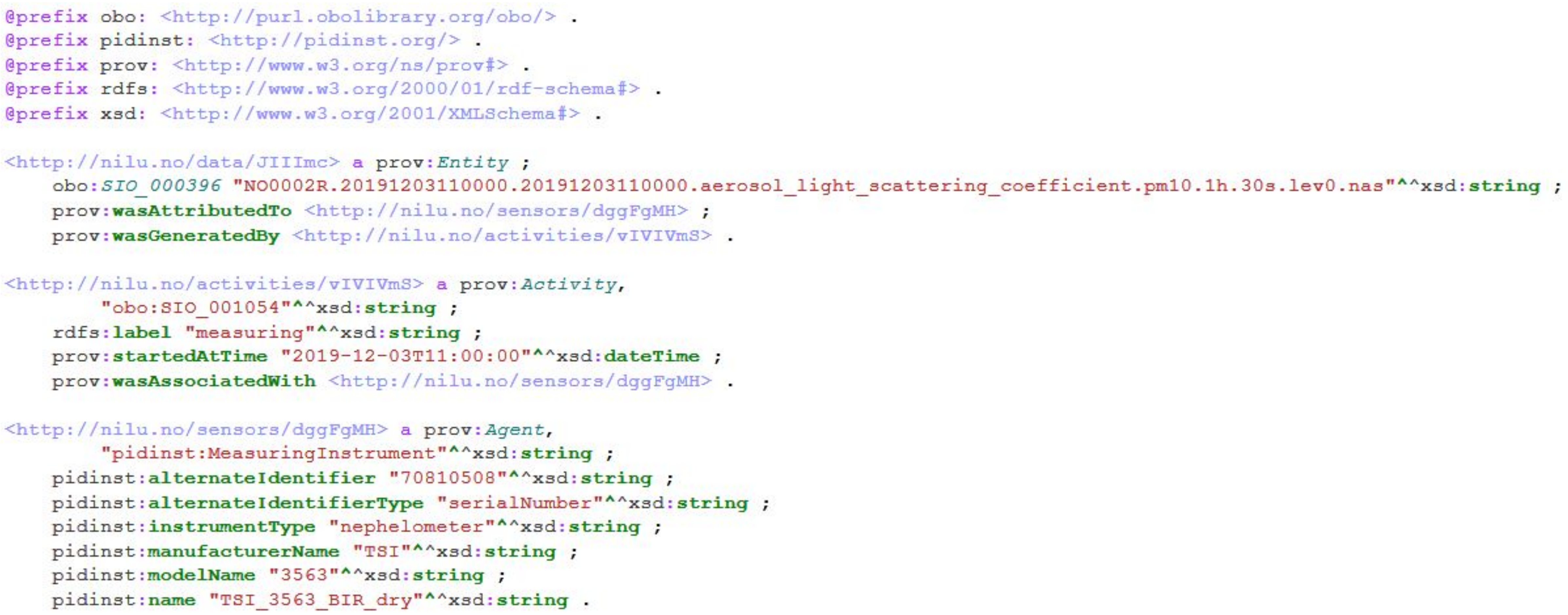
wasGeneratedBy

prov:startTime	2019-12-03T11:00:00
prov:label	measuring
prov:type	obo:SIO_001054

nilu:data/JIIImc

obo:SIO_000396	NO0002R.20191203110000.20191203110000.aerosol_light_scattering_coefficient.pm10.1h.30s.lev0.nas

Demonstrator result



```
@prefix obo: <http://purl.obolibrary.org/obo/> .
@prefix pidinst: <http://pidinst.org/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://nilu.no/data/JIIImc> a prov:Entity ;
    obo:SIO_000396 "NO0002R.20191203110000.20191203110000.aerosol_light_scattering_coefficient.pm10.1h.30s.lev0.nas"^^xsd:string ;
    prov:wasAttributedTo <http://nilu.no/sensors/dggFgMH> ;
    prov:wasGeneratedBy <http://nilu.no/activities/vIVIVmS> .

<http://nilu.no/activities/vIVIVmS> a prov:Activity,
        "obo:SIO_001054"^^xsd:string ;
    rdfs:label "measuring"^^xsd:string ;
    prov:startedAtTime "2019-12-03T11:00:00"^^xsd:dateTime ;
    prov:wasAssociatedWith <http://nilu.no/sensors/dggFgMH> .

<http://nilu.no/sensors/dggFgMH> a prov:Agent,
        "pidinst:MeasuringInstrument"^^xsd:string ;
    pidinst:alternateIdentifier "70810508"^^xsd:string ;
    pidinst:alternateIdentifierType "serialNumber"^^xsd:string ;
    pidinst:instrumentType "nephelometer"^^xsd:string ;
    pidinst:manufacturerName "TSI"^^xsd:string ;
    pidinst:modelName "3563"^^xsd:string ;
    pidinst:name "TSI_3563_BIR_dry"^^xsd:string .
```

Live demo

https://mybinder.org/v2/gh/envri-fair/provenance-demonstrator.git/HEAD

Discussion

- Provenance metadata in RDF can be loaded into a triple store
- This metadata can then be served on the Web using a SPARQL endpoint
- Other schemes allow for encoding provenance metadata (e.g., CERIF)

Discussion

- Provenance is generated not just by activities executed within an RI
- It is also generated when RI data are used (e.g., by research communities)
- Should/can such provenance be captured?
- A concrete example: ENVRIplus Aerosol Science Demonstrator
- https://github.com/markusstocker/lecce-summer-school