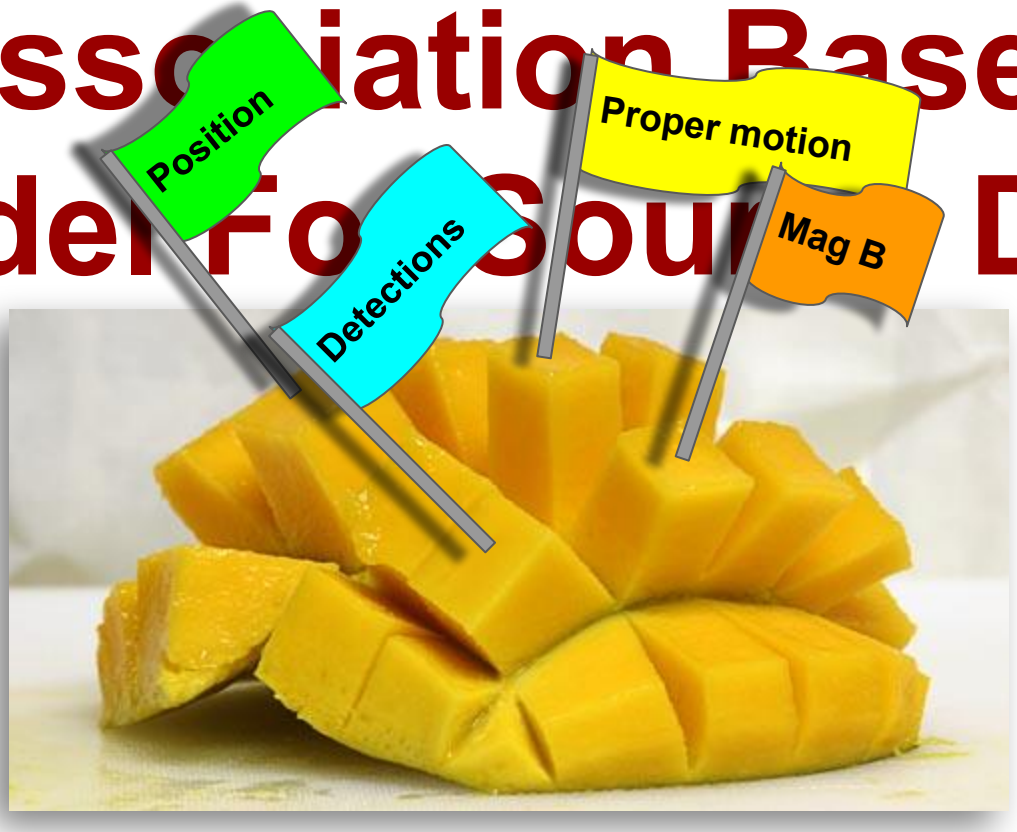
A Component and Association Based Model For Gou Data



L. Michel F. Bonnarel Gilles Landais Mireille Louys
in collaboration with CDS and others VO members

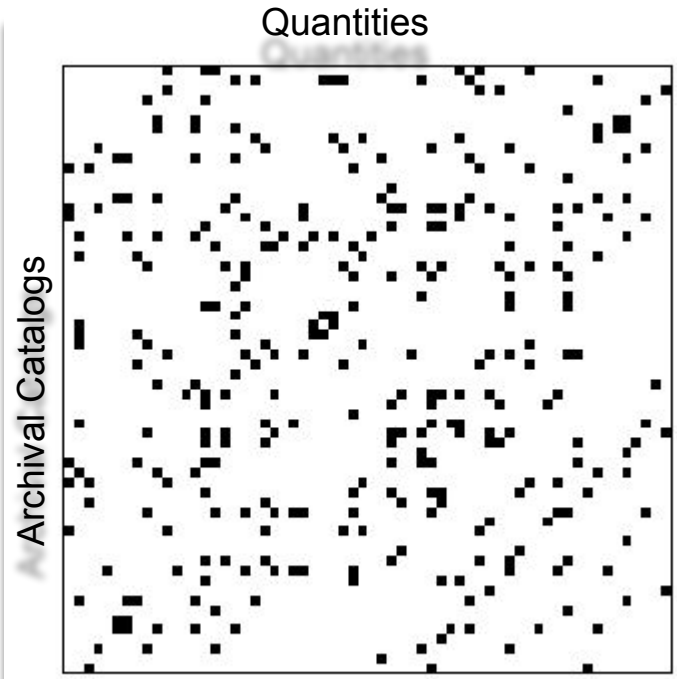
Session Roadmap

- **Model Proposal**
- **Data Annotation Process**
- **Python Proof of Concept**
  - Model building
  - Concrete example of an annotation process
  - Client implementation
  - Client interoperability

Describing a Source: an Endless Job

- **Lots of possible sets of parameters**
  - Lots of different parameters
  - Different roles for the same types of parameters
- **Lots of source types**
  - Stars, extended objects, orbiting stars, complex shaped object…..
- **Lots of possible associated data**
  - Time series, spectra, SEDs, multiple detections, cross-match….

Motivation for a Source Model



A model that would pretend to support the most used quantities would have to tackle with to **2 serious issues**:

- An **unavoidable complexity** because people using it will have to deal with many quantities useless for their particular use case.
- **No agreement** on what the most used quantities are.

2 options

- Explicitly limiting the scope of the model
- **Bypassing the hurdle**

A Suitable Example: The XMM Catalogue

Parameters

- Raw position with errors
- Corrected position with errors
- Detection flags
- Fluxes with errors in multiple Energy bands
- Hardness Ratio with errors in multiple Energy bands

Associated data

- Multiple Detections
- Xmatched sources
- Spectra
- Time series
- SEDs

How to make a distinction between raw and corrected position ?

How to interpret the flag values?

How to define the energy bands?

How to recognize an HR?

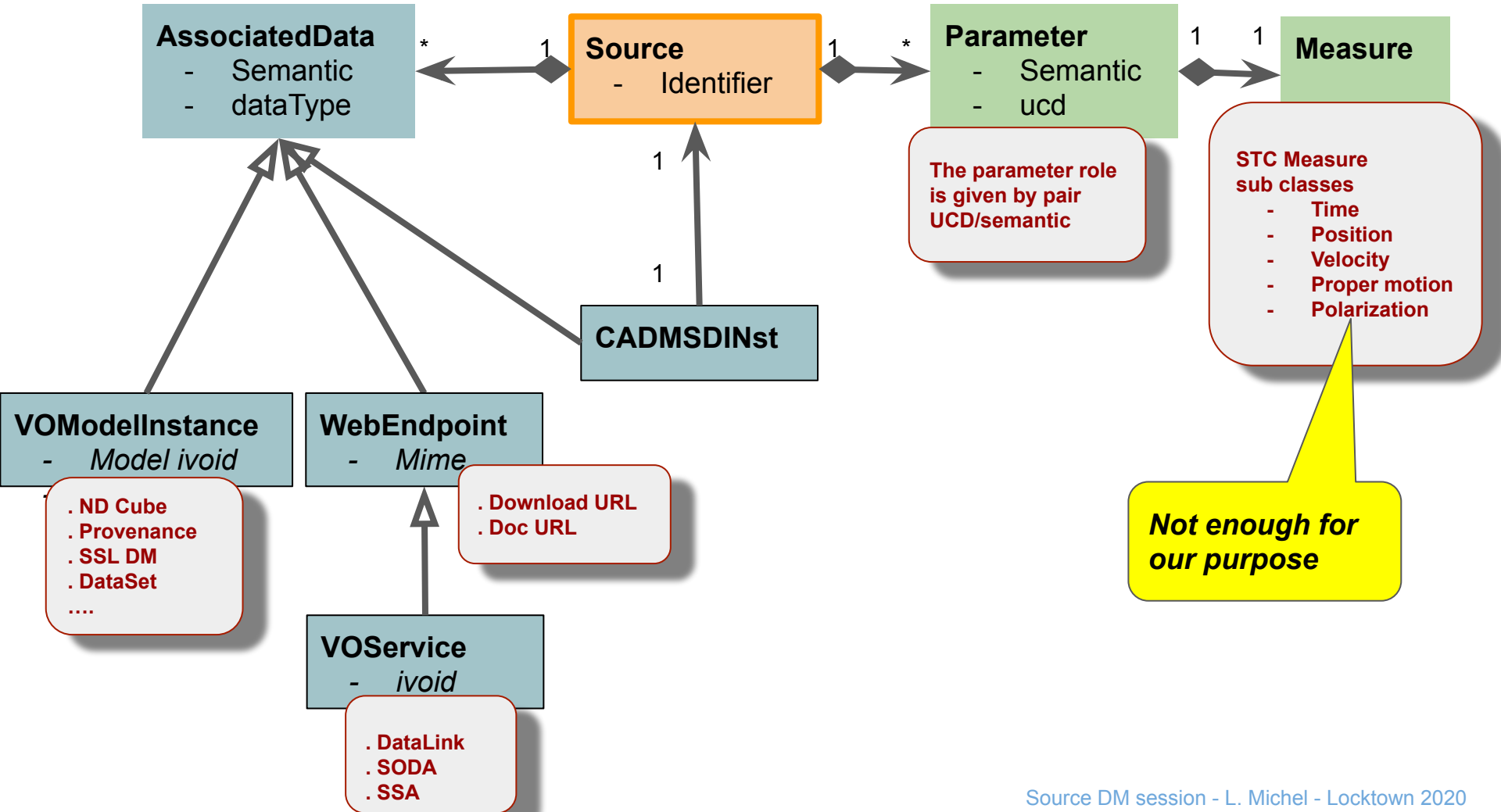
All of those are associated data, but how to know their type

NOTE: The workflow presented here is focused on the parameters

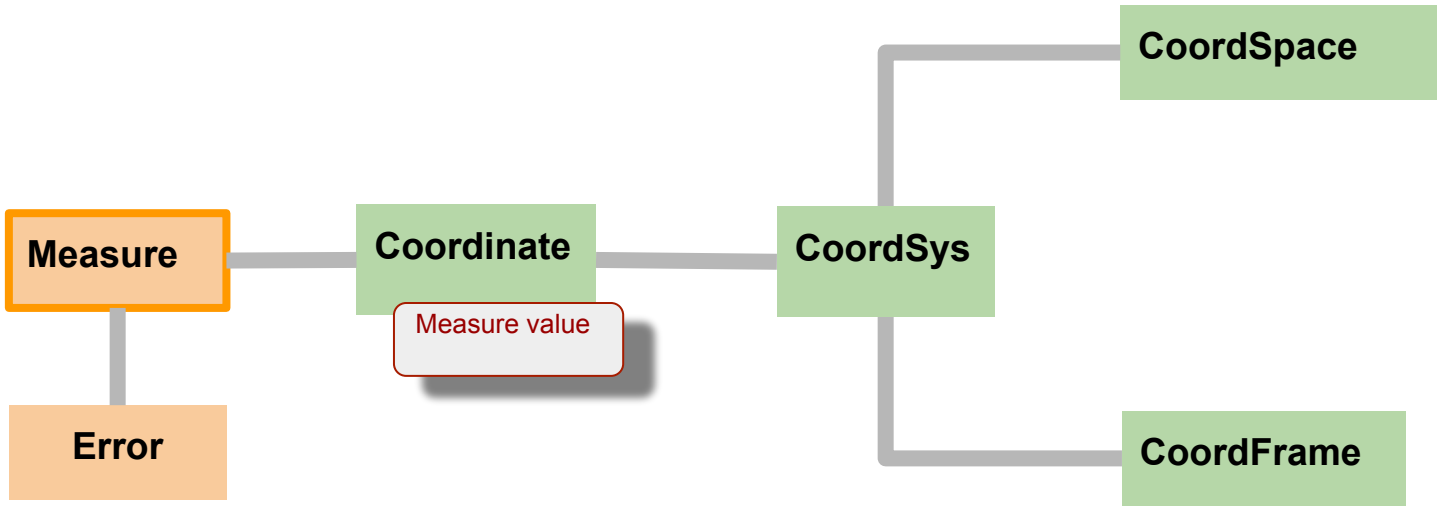
Model Requirements

- **Support of a huge diversity of quantities**
  - We got it with Measure/Coordinates
- **Extendable to new quantities without breaking the model**
  - Extension of Measure/Coordinates
- **Capability of describing the role of each quantity**
  - UCD + semantic tags
- **Embedding similar quantities with different roles**
  - Same UCD
  - Different semantic tags

CAB-MSD at a Glance



STC (simplified) Pattern

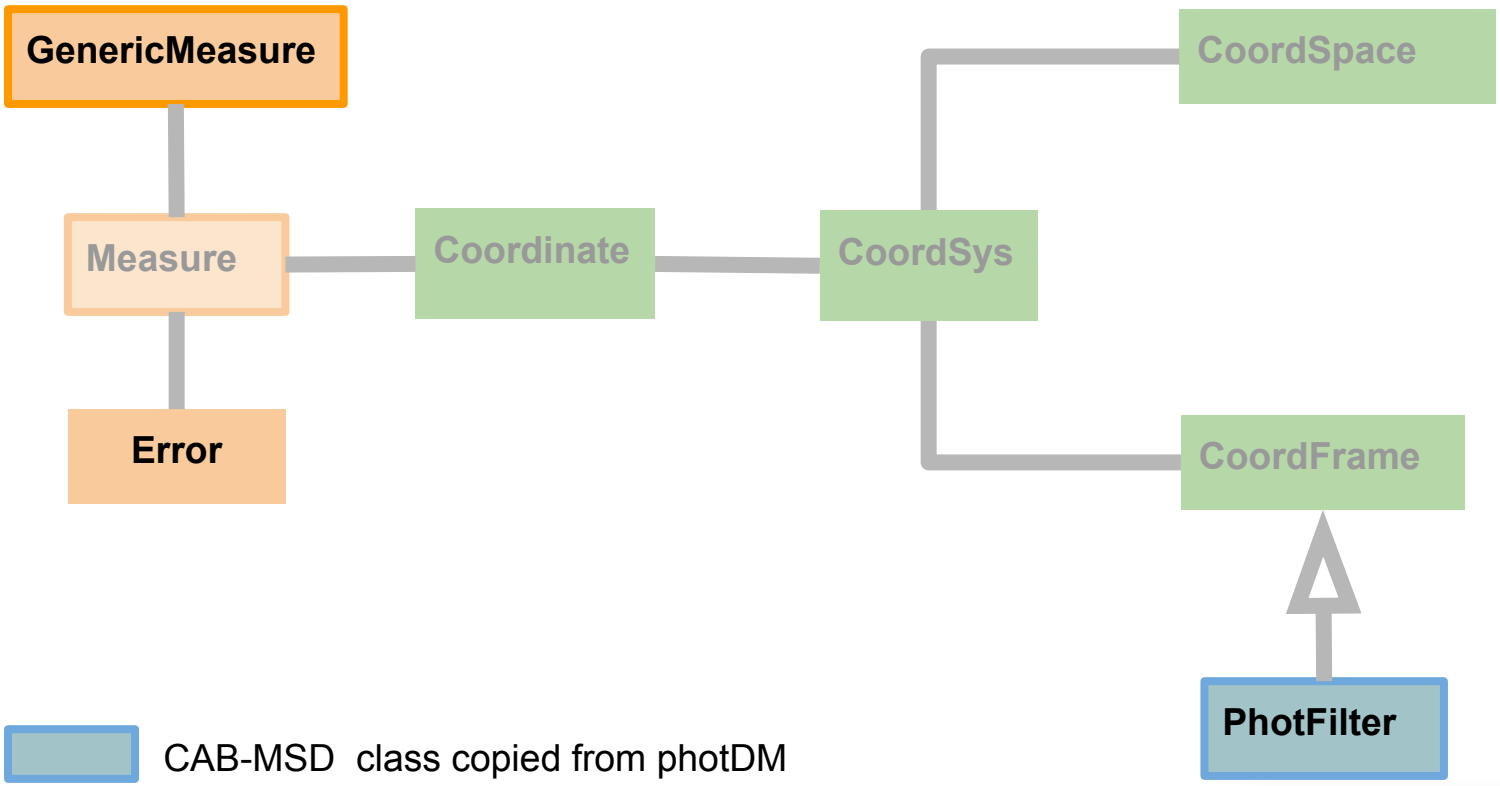


NOTE: Measure natively defined in STC are used as much as possible

Grey links are logical links
The detail of the Meas/Coord is hidden.

General relation of the Meas/Coords objects.

STC Extension: Luminosity



CAB-MSD class copied from photDM

STC class

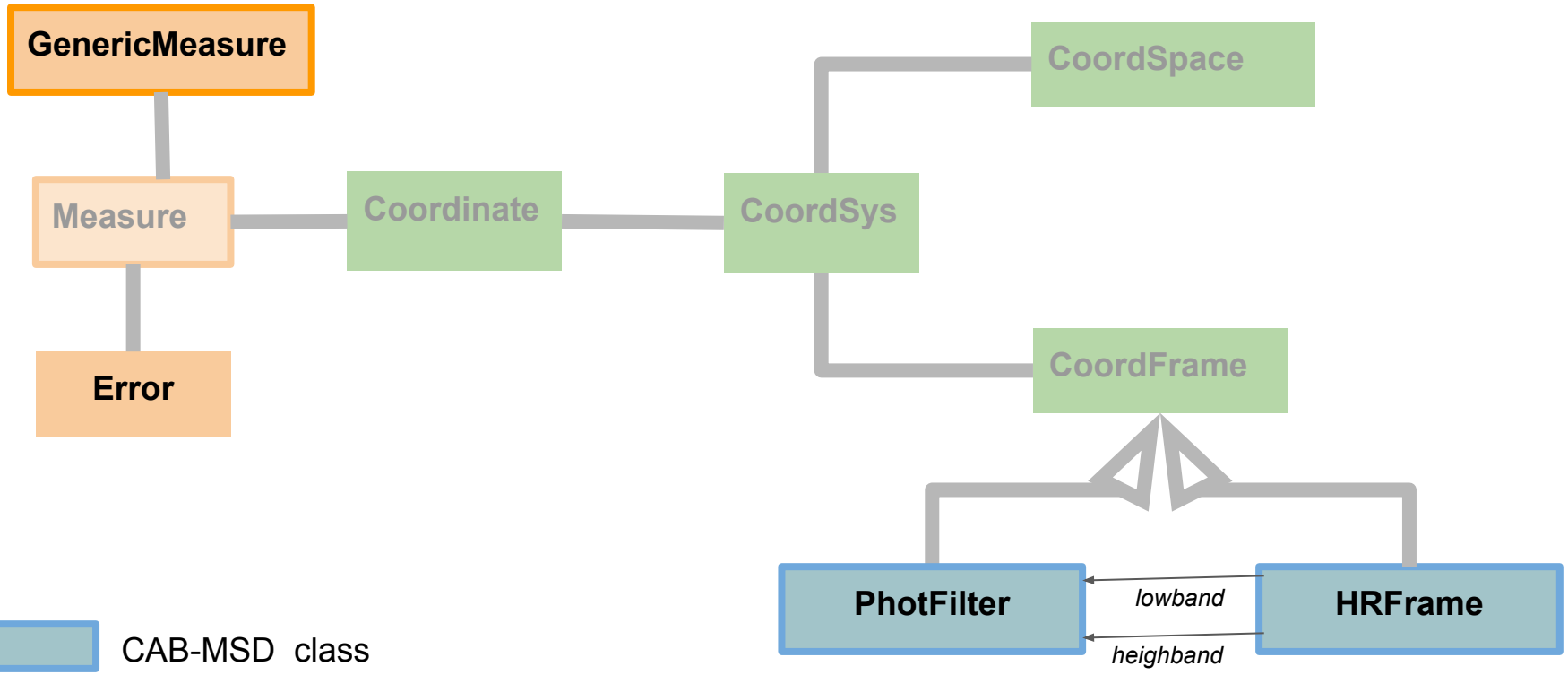
General relation of the Meas/Coords objects.

CABMSD Extension of the Meas/Coords objects.

Grey links are logical links
The detail of the Meas/Coord is hidden.

STC Extension: Hardness Ratio

$$HR = \frac{F_{heighband} - F_{lowband}}{F_{heighband} + F_{lowband}}$$



CAB-MSD class

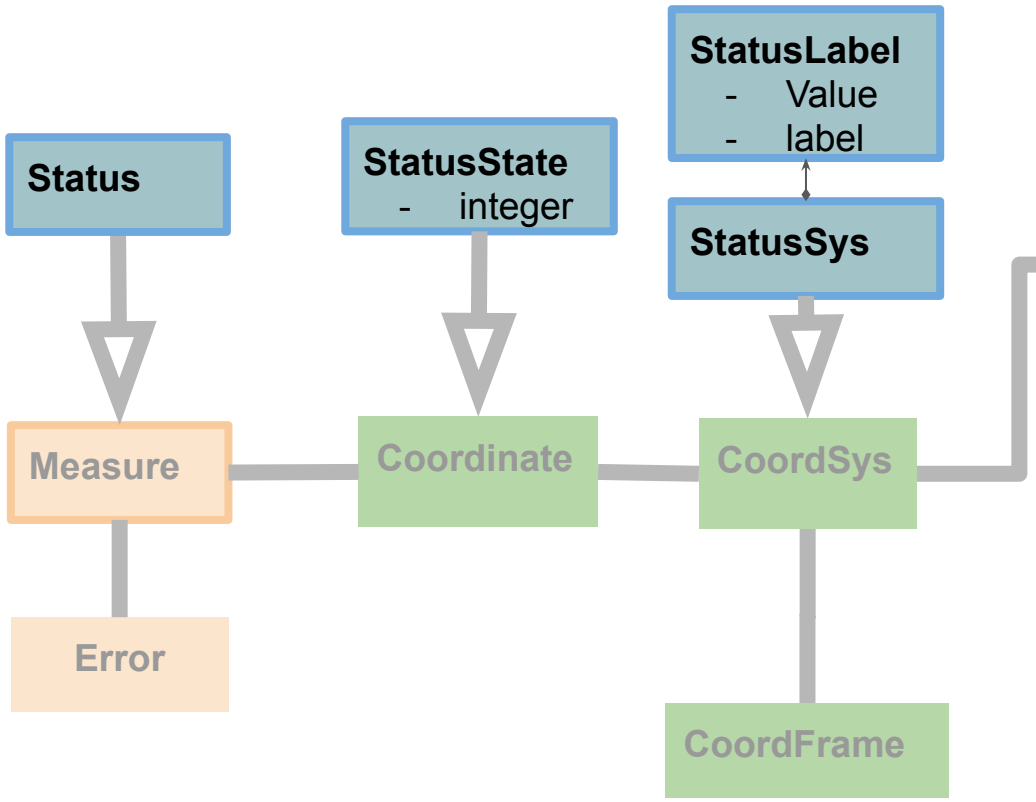
STC class

General relation of the Meas/Coords objects.

CABMSD Extension of the Meas/Coords objects.

Grey links are logical links
The detail of the Meas/Coord is hidden.

STC extension: Flag




```
XMM detection status example
     0: good
     1: Sources parameters may
        be affected
     2: possibly spurious
     3: located in an area
        where spurious
        detection may occur
```

CAB-MSD class

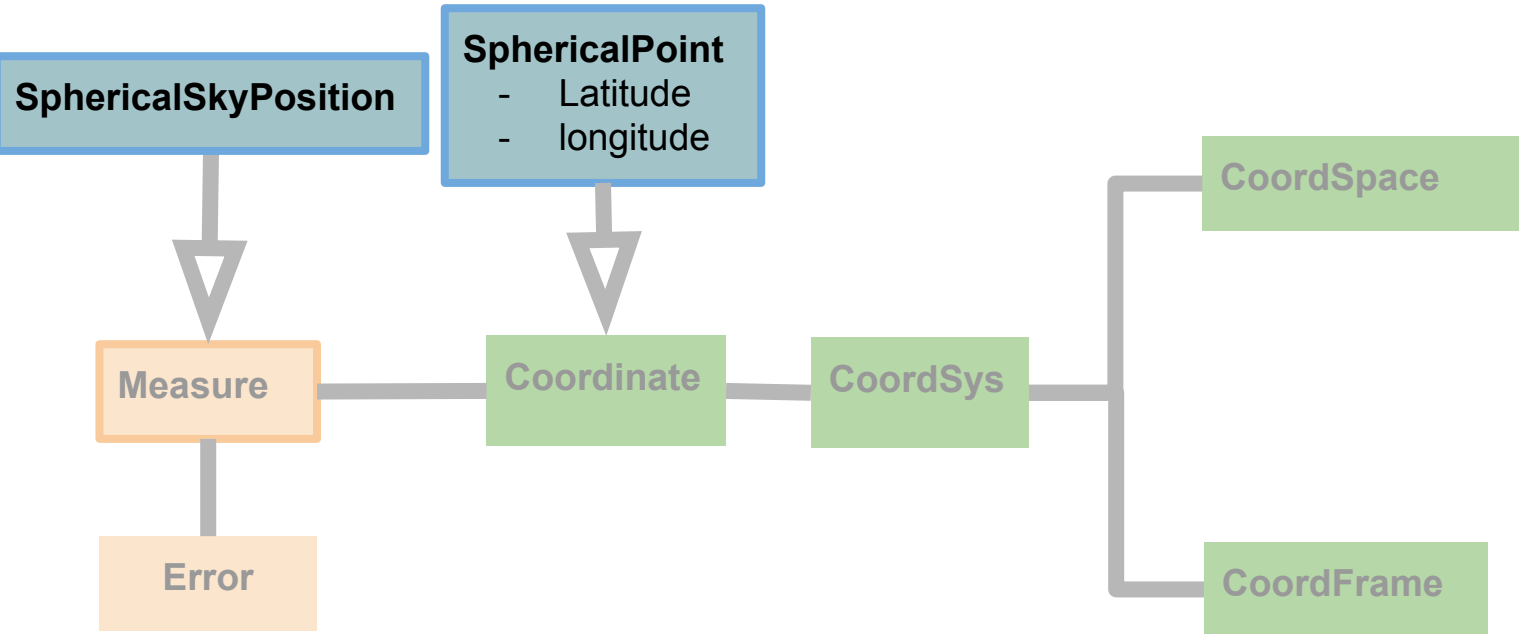
STC class

General relation of the Meas/Coords objects.

CABMSD Extension of the Meas/Coords objects.

Grey links are logical links
The detail of the Meas/Coord is hidden.

STC Extension: Sky Position Shortcut



CAB-MSD class

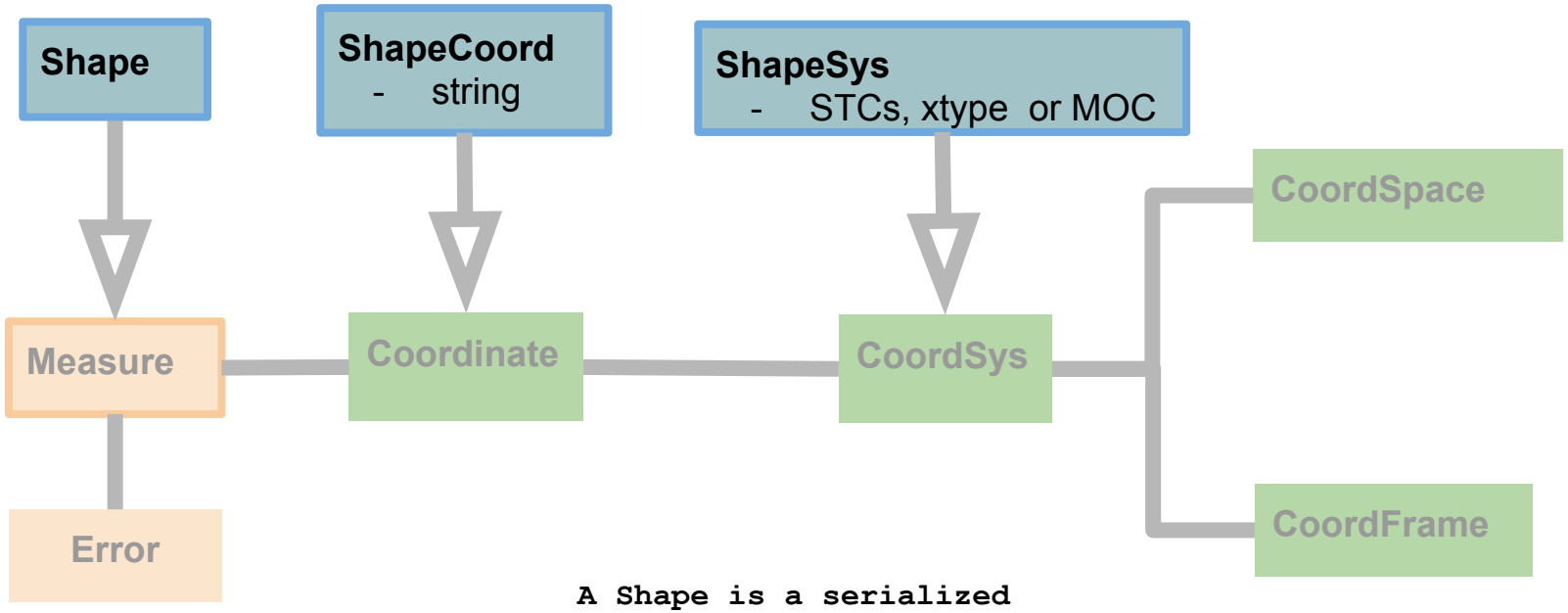
STC class

General relation of the Meas/Coords objects.

CABMSD Extension of the Meas/Coords objects.

Grey links are logical links
The detail of the Meas/Coord is hidden.

STC extension: Shape



A Shape is a serialized form of a region.

It does not intend to denote a polygon

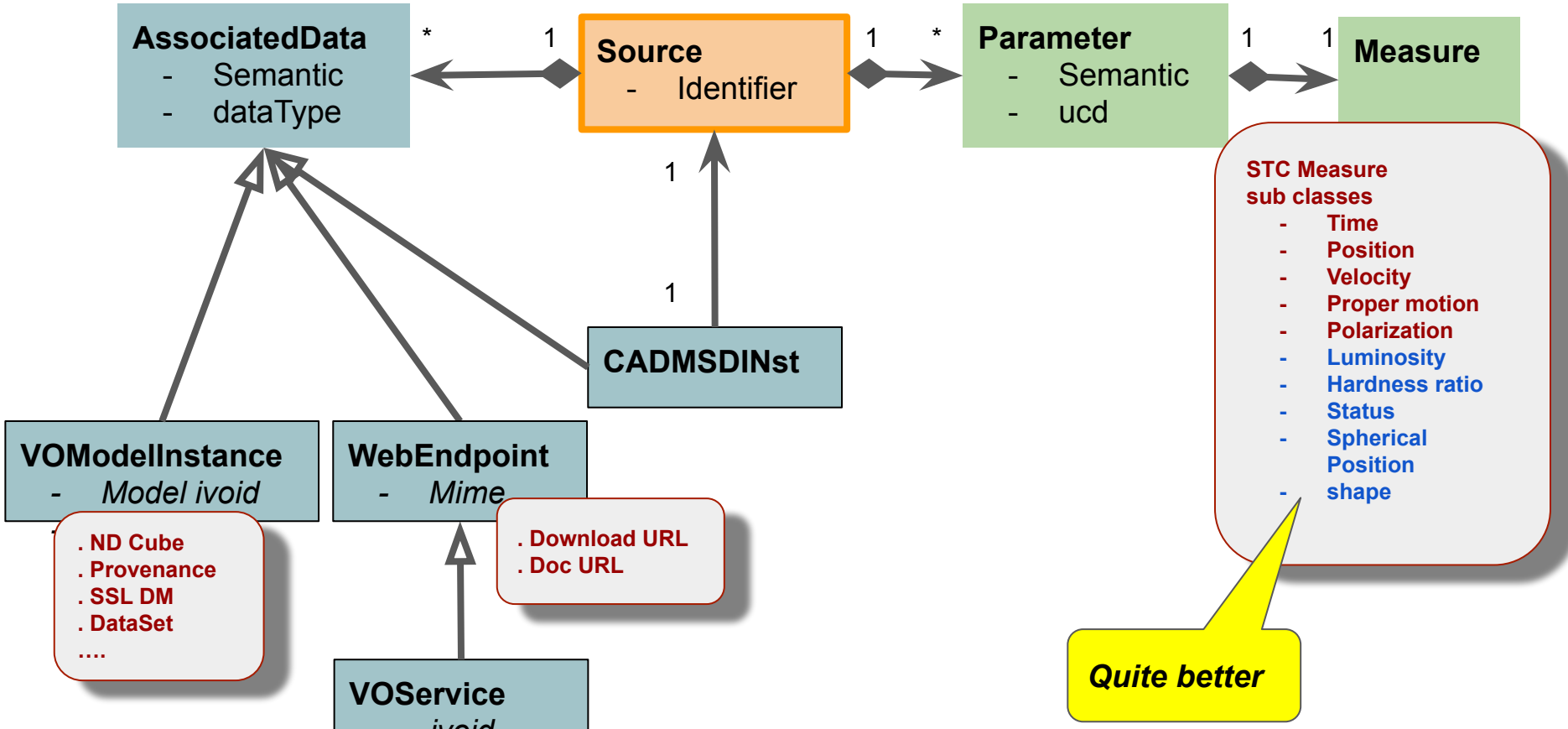
CAB-MSD class

Grey links are logical links
The detail of the Meas/Coord is hidden.

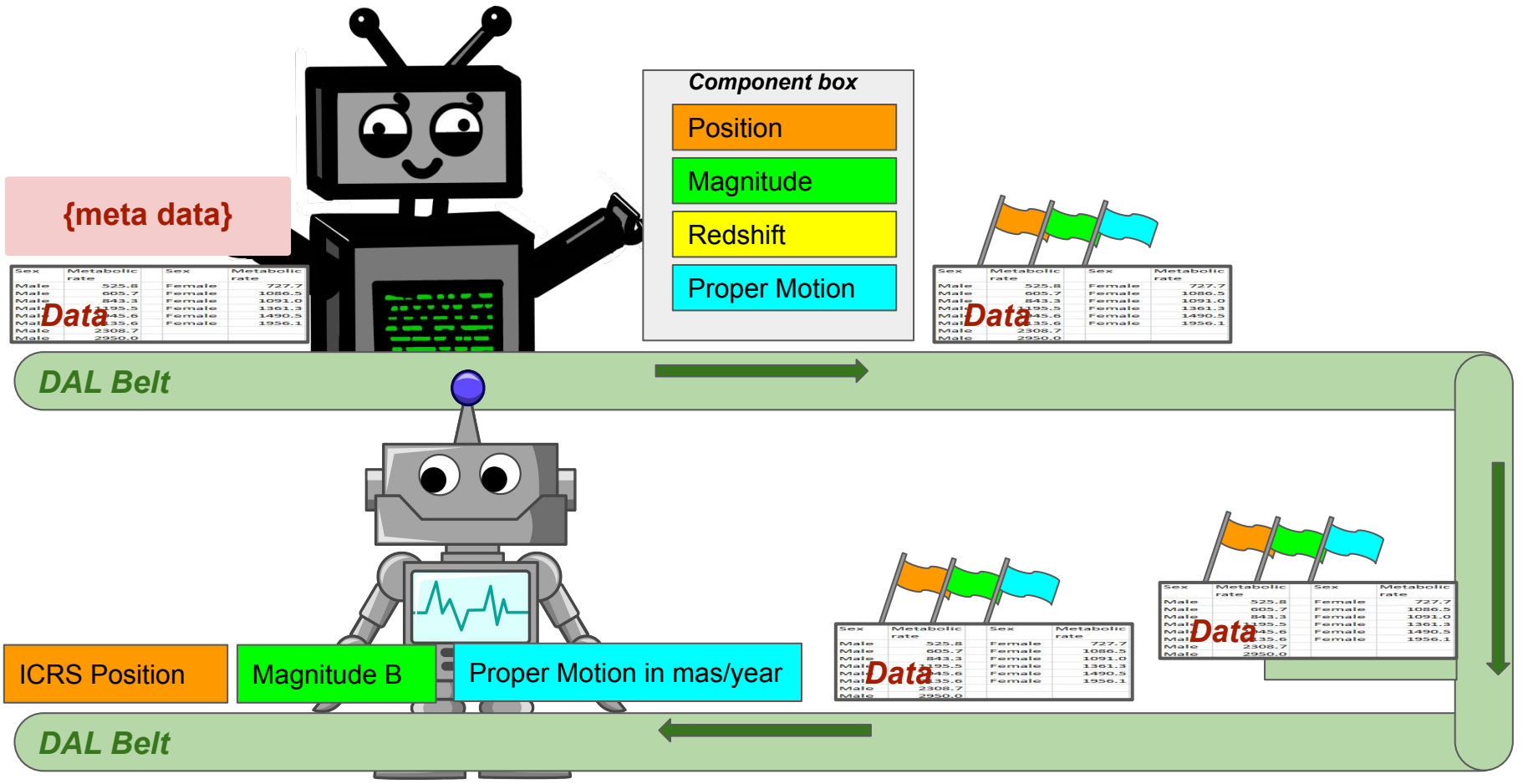
General relation of the Meas/Coords objects.

CABMSD Extension of the Meas/Coords objects.

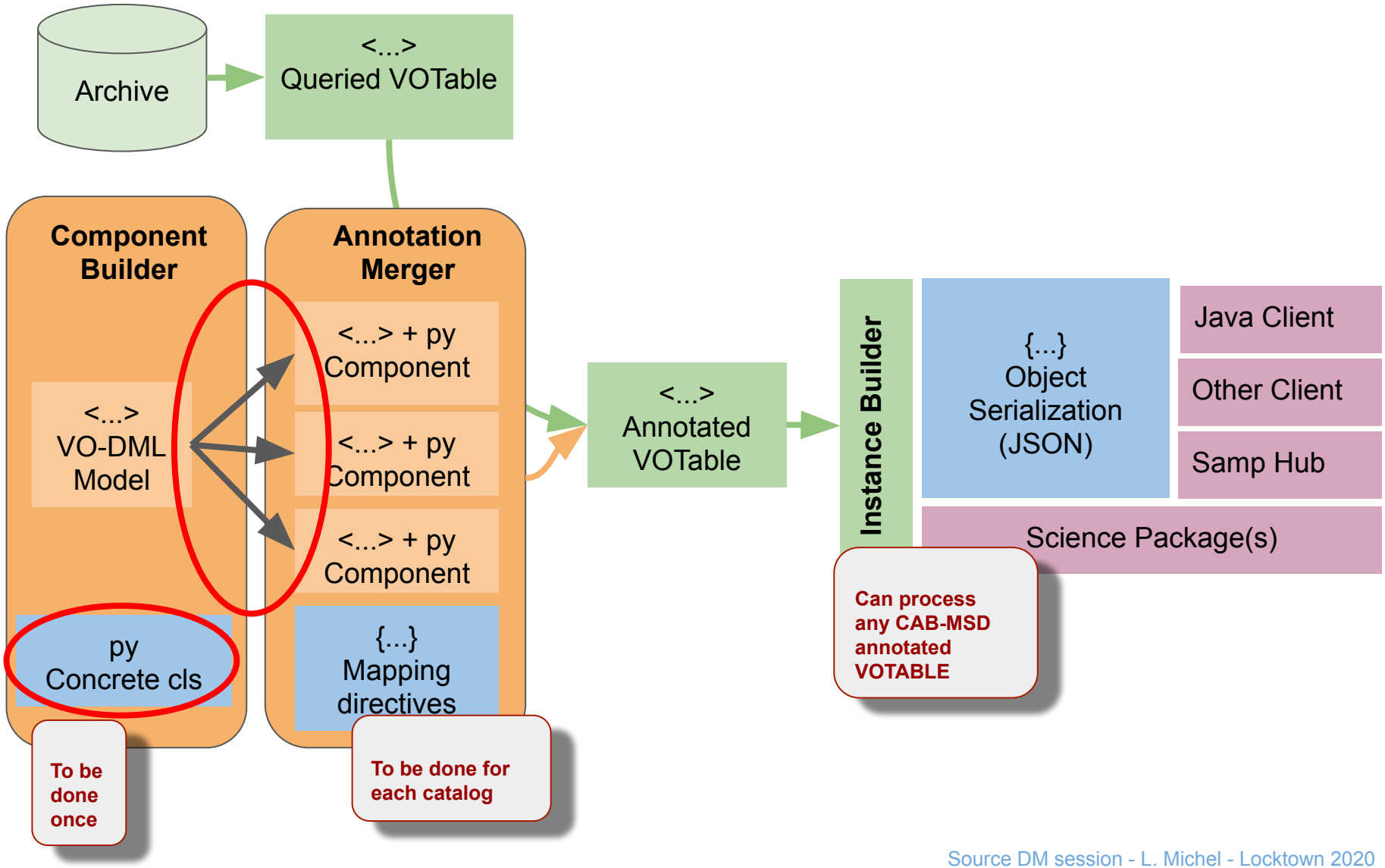
CAB-MSD with Extended STC



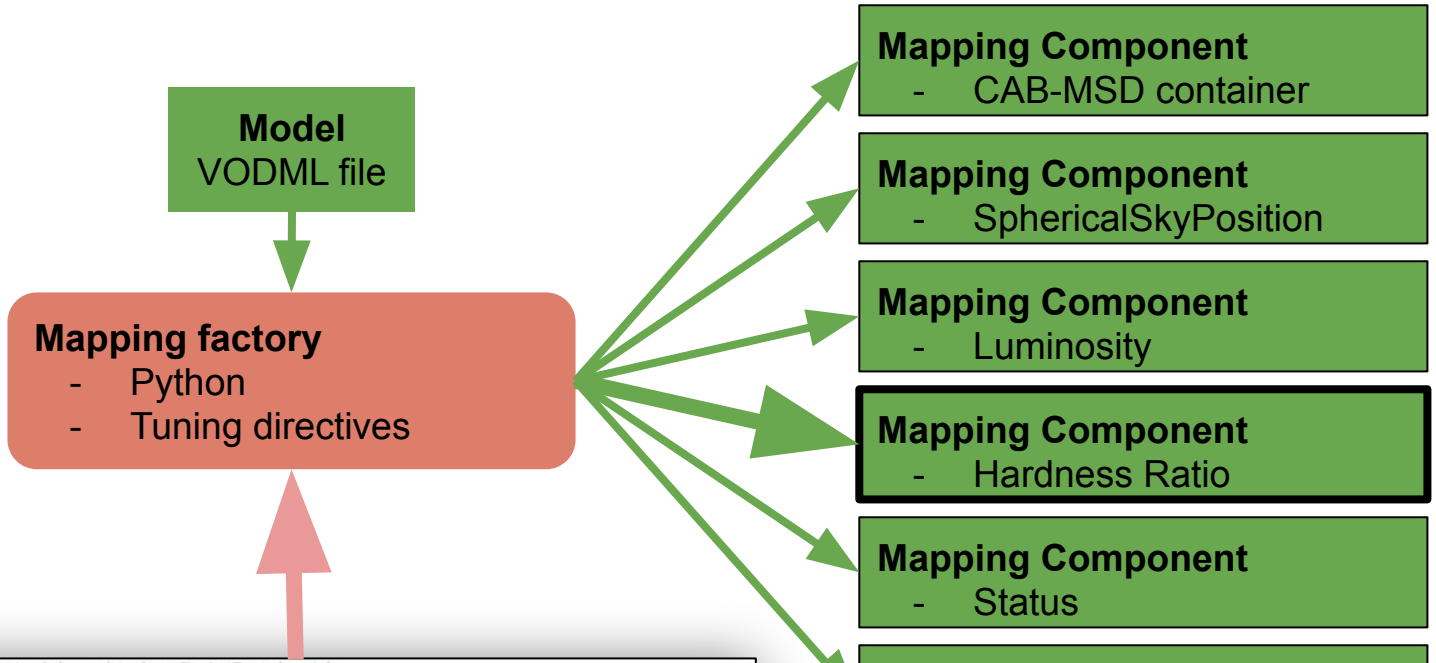
The Robotic Picture of the Process



The Archive DB case: Step #1



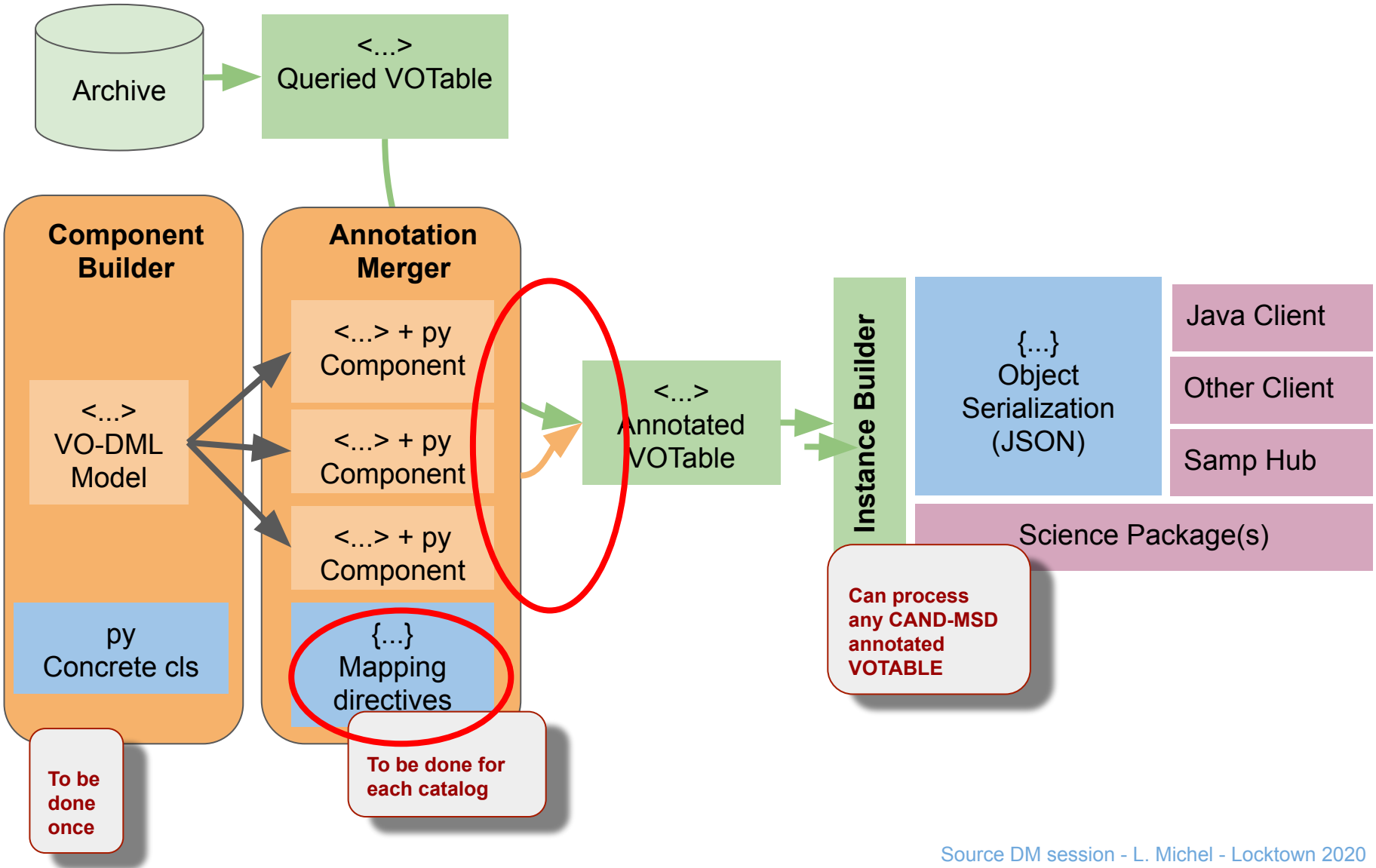
Step #1: Building Mapping Component




```
#root_object_id = 'cube:DataProduct'
mapping_generator.root_object_id = 'meas:GenericMeasure'
# set the concrete object types
mapping_generator.concrete_classes = {
    "coords:Coordinate.coordSys": ["coords:PhysicalCoordSys"],
    "coords:PhysicalCoordSys.frame": ["cab_msd:HRFrame"]
    }
# set the concrete data types
mapping_generator.concrete_types={
    "meas:Error.statError": "meas:Symmetrical",
    "meas:Error.sysError": "meas:Symmetrical"
}
```

- Must be done once
- The mapping components are templates that can be reused for all data sets
- They can (must) be refined by hand

The Archive DB case: Step #2



Step #2: VOTable Mapping

Mapping Components

mapping_components
- cab_msd.mapping.json
- cab_msd.mapping.xml
- position.mapping.xml
- status.mapping.xml

Mapping Setup

```
"parameters": [
    {
        "measure": "position",
        "ucd": "pos;meta.main",
        "semantic": "corrected",
        "identifier": "namesaada",
        "frame": {
            "frame": "ICRS",
            "equinox": null
        },
        "position": {
            "longitude": "_ra_146",
            "latitude": "_dec_147",
            "unit": "deg"
        },
        "errors": {
            "systematic": {
                "value": "_syserrcc_152",
                "unit": "arcsec"
            },
            "random": {
                "value": "_poserr_148",
                "unit": "arcsec"
            }
        }
    },
    {
        "measure": "status",
        "ucd": "meta.code.error",
        "semantic": "computed",
        "identifier": "namesaada",
        "status": {
            "value": "_sum_flag_394"
        }
```

Searched VOTable
- README.md
- xmm_detections.xml

Component processor
- SphericalSkyPosition

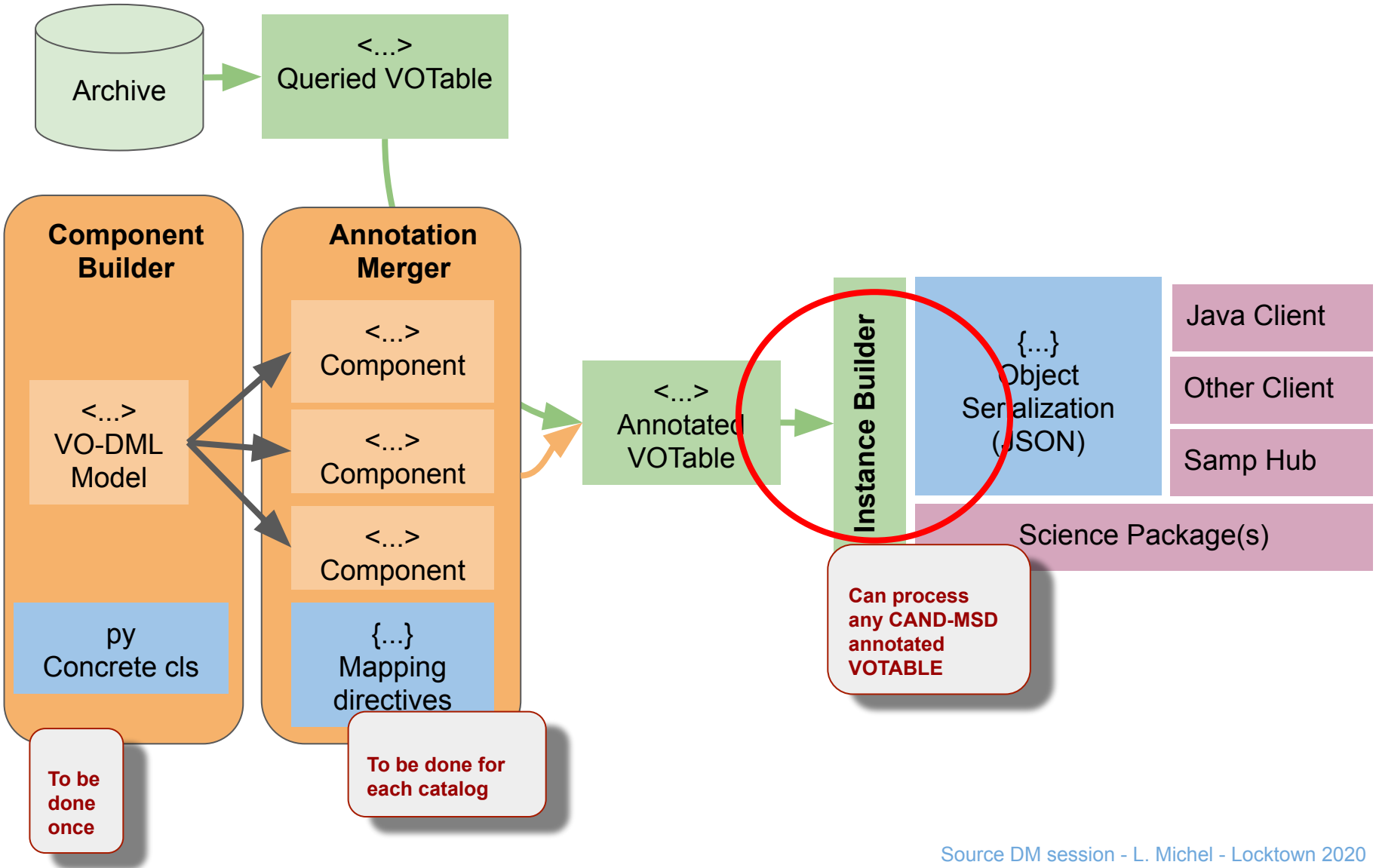
Component processor
- Status

Annotation merger

Annotated VOTable
- README.md
- xmm_detections.annot.xml

The only resource that must be set up by hand in the process.
The model complexity is hidden by the component processor

The Archive DB case: The Client



Building a Model Instance from the Table

```
instance_from_votable = InstanceFromVotable(votable_path)
instance = instance_from_votable.build_instance(resolve_refs=True)

print("=== Mapping of the columns")
print(instance.get_flatten_data_head())
#print(instance.get_data_subset_keys())
print("=== First row: flatten mode")
while True:
    inst = instance._get_next_flatten_row()
    if inst != None:
        print(DictUtils.get_pretty_json(inst))
        break
    else:
        break
```

```
=== Mapping of the columns
['cab_msd:STCSphericalPoint.longitude(ivoa:RealQuantity.value) [col#46 _ra_146]',
 'cab_msd:STCSphericalPoint.latitude(ivoa:RealQuantity.value) [col#47 _dec_147]',
 'meas:Symmetrical.radius(ivoa:RealQuantity.value) [col#48 _poserr_148]',
 'meas:Symmetrical.radius(ivoa:RealQuantity.value) [col#52 _syserrcc_152]',
 'cab_msd:STCStatus.coord(cab_msd:STCStatusState.status) [col#294
_sum_flag_394]',
 'None(cab_msd:Source.identifier) [col#1 namesaada]']
=== First row: flatten mode
[
  340.91055060369,
  -17.071667101891,
  "1",
  "0",
  "0",
  "4XMM J224338.5-170418"
```

Building a Model Instance from the Table

```
instance_from_votable = InstanceFromVotable(votable_path)
instance = instance_from_votable.build_instance(resolve_refs=True)

print("=== Mapping of the columns")
print(instance.get_flatten_data_head())
print("=== Read row: instance mode")
while True:
    inst = instance._get_next_row_instance()
    if inst != None:
        print(DictUtils.get_pretty_json(inst))
    else:
        break
```


```
"cab_msd:Source.parameters": [
  {
    "@dmrole": "cab_msd:Source.parameters",
    "@dmtype": "cab_msd:Parameter",
    "cab_msd:Parameter.measure": {
      "@dmtype": "cab_msd:STCSphericalSkyPosition",
      "cab_msd:STCSphericalSkyPosition.coord": {
        "@dmtype": "cab_msd:STCSphericalPoint",
        "cab_msd:STCSphericalPoint.latitude": {
          "@dmtype": "ivoa:RealQuantity",
          "ivoa:Quantity.unit": {
            "@dmtype": "ivoa:Unit",
            "@value": "deg"
          },
          "ivoa:RealQuantity.value": {
            "@dmtype": "ivoa:real",
            "@ref": "_dec_147",
            "@value": -27.720584349377
          }
        },
        "cab_msd:STCSphericalPoint.longitude": {
          "@dmtype": "ivoa:RealQuantity",
          "ivoa:Quantity.unit": {
            "@dmtype": "ivoa:Unit",
            "@value": "deg"
          },
          "ivoa:RealQuantity.value": {
            "@dmtype": "ivoa:real",
            "@ref": "_ra_146",
            "@value": 52.616760006165
          }
        },
        "coords:Coordinate.coordSys": {..
      },
      "meas:Measure.error": {..
    },
    "cab_msd:Parameter.semantic": {
      "@dmtype": "ivoa:string",
      "@value": "corrected"
    },
    "cab_msd:Parameter.ucd": {
      "@dmtype": "ivoa:string",
      "@value": "pos;meta.main"
    }
  },
```


This JSON serialization of the model instance is perfectly interoperable

- Can be exchanged by different software
- Can be sent by SAMP

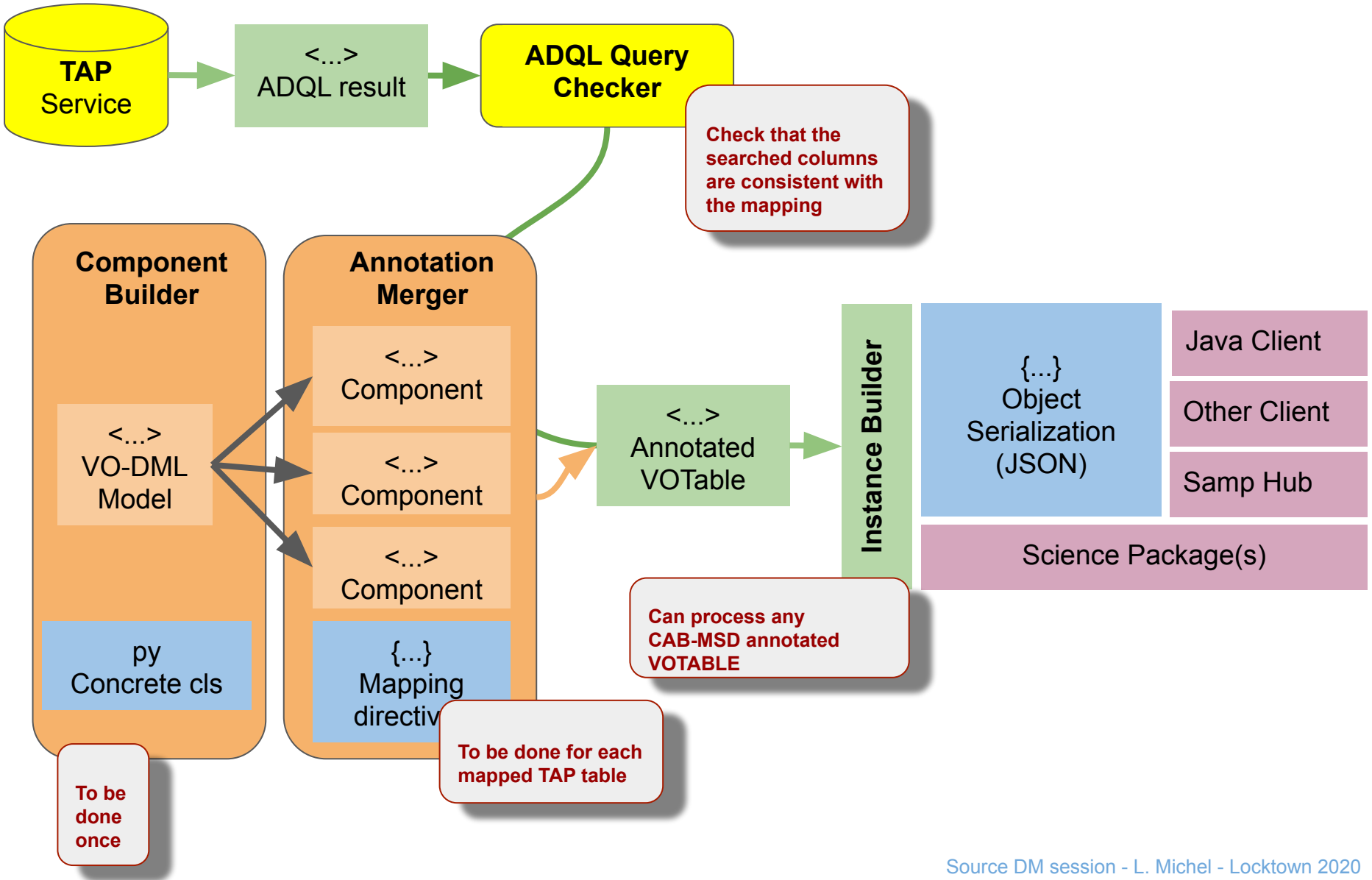
Building a Model Instance from the Table

```
{
  "@dmrole": "cab_msd:Source.parameters",
  "@dmtype": "cab_msd:Parameter",
  "cab_msd:Parameter.measure": {
    "@dmtype": "cab_msd:STCStatus",
    "cab_msd:STCStatus.coord": {
      "@dmtype": "cab_msd:STCStatusState",
      "cab_msd:STCStatusState.status": {
        "@dmtype": "ivoa:integer",
        "@ref": "_sum_flag_394",
        "@value": "0"
      },
      "coords:Coordinate.coordSys": {
    }
  },
  "cab_msd:Parameter.semantic": {
    "@dmtype": "ivoa:string",
    "@value": "computed"
  },
  "cab_msd:Parameter.ucd": {
    "@dmtype": "ivoa:string",
    "@value": "meta.code.error"
  }
}
```

meta.code.qual

```
"@dmtype": "cab_msd:STCStatusSys",
"cab_msd:STCStatusSys.statusLabel": [
  {
    "cab_msd:STCStatusSys.statusLabel": [
      {
        "@dmrole": "cab_msd:STCStatusSys.statusLabel",
        "@dmtype": "cab_msd:StatusLabel",
        "cab_msd:StatusLabel.label": {
          "@dmtype": "ivoa:string",
          "@value": "good"
        },
        "cab_msd:StatusLabel.value": {
          "@dmtype": "ivoa:integer",
          "@value": "0"
        }
      },
      {
        "@dmrole": "cab_msd:STCStatusSys.statusLabel",
        "@dmtype": "cab_msd:StatusLabel",
        "cab_msd:StatusLabel.label": {
          "@dmtype": "ivoa:string",
          "@value": "source parameters may be affected"
        },
        "cab_msd:StatusLabel.value": {
          "@dmtype": "ivoa:integer",
          "@value": "1"
        }
      },
      {
        "@dmrole": "cab_msd:STCStatusSys.statusLabel",
        "@dmtype": "cab_msd:StatusLabel",
        "cab_msd:StatusLabel.label": {
          "@dmtype": "ivoa:string",
          "@value": "possibly spurious"
        },
        "cab_msd:StatusLabel.value": {
          "@dmtype": "ivoa:integer",
          "@value": "2"
        }
      },
      {
        "@dmrole": "cab_msd:STCStatusSys.statusLabel",
        "@dmtype": "cab_msd:StatusLabel",
        "cab_msd:StatusLabel.label": {
          "@dmtype": "ivoa:string",
          "@value": "located in a area where spurious detection may occur"
        },
        "cab_msd:StatusLabel.value": {
          "@dmtype": "ivoa:integer",
          "@value": "3"
        }
```

The TAP case


TAP
Service
<...>
ADQL result
ADQL Query
Checker
Check that the searched columns are consistent with the mapping
Component
Builder
<...>
VO-DML
Model
py
Concrete cls
To be done once
Annotation
Merger
<...>
Component
<...>
Component
<...>
Component
{...}
Mapping
directiv
To be done for each mapped TAP table
<...>
Annotated
VOTable
Instance Builder
{...}
Object
Serialization
(JSON)
Java Client
Other Client
Samp Hub
Science Package(s)
Can process any CAB-MSD annotated VOTABLE

Data annotation : “Don’t be evil” (Larry Page)

- **Shy Annotations**
  - **#1:** Able to be ignored
    - Do not break working things
    - The parser implementation shouldn’t alter the existing code
    - The annotation implementation shouldn’t alter the original data
  - **#2**: Easy to use as template
    - **The mapping structure must be independent of the data structure**
  - **#3:** Parser helper: Can be used at different levels
    - Provide a clear indication on the VOTable content
    - Can be used at different levels
      - Just get the meta data
      - Get everything through the model

Mapping Syntax Sample

```
<TEMPLATES>
  <ARRAY dmrole="root">
    <INSTANCE dmrole="root" dmtype="cab_msd:Source">
  </ARRAY>
</TEMPLATES>
```

This is a table of CAB-MSD instances.
The instance inner structure is folded

```
<INSTANCE ID="StatusSys" dmrole="" dmtype="cab_msd:STCStatusSys">
  <COMPOSITION size="5" dmrole="cab_msd:STCStatusSys.statusLabel">
    <INSTANCE dmrole="cab_msd:STCStatusSys.statusLabel" dmtype="cab_msd:StatusLabel">
    <INSTANCE dmrole="cab_msd:STCStatusSys.statusLabel" dmtype="cab_msd:StatusLabel">
    <INSTANCE dmrole="cab_msd:STCStatusSys.statusLabel" dmtype="cab_msd:StatusLabel">
    <INSTANCE dmrole="cab_msd:STCStatusSys.statusLabel" dmtype="cab_msd:StatusLabel">
    <INSTANCE dmrole="cab_msd:STCStatusSys.statusLabel" dmtype="cab_msd:StatusLabel">
  </COMPOSITION>
</INSTANCE>
```

A Coord system for a status.
All of the 5 possible labels are folded.

```
<INSTANCE ID="SpaceCoordFrame" dmrole="">
  <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtype="coords:StdRefLocation">
    <VALUE dmrole="coords:StdRefLocation.position" dmtype="ivoa:string" value="NoSet"/>
  </INSTANCE>
  <VALUE dmrole="coords:SpaceFrame.spaceRefFrame" dmtype="ivoa:string" value="ICRS"/>
  <VALUE dmrole="coords:SpaceFrame.equinox" dmtype="coords:Epoch" value="NoSet"/>
</INSTANCE>
```

ICRS space frame

```
<COMPOSITION size="-1" dmrole="cab_msd:Source.parameters">
  <INSTANCE dmrole="cab_msd:Source.parameters" dmtype="cab_msd:Parameter">
    <VALUE dmrole="cab_msd:Parameter.semantic" dmtype="ivoa:string" value="corrected"/>
    <VALUE dmrole="cab_msd:Parameter.ucd" dmtype="ivoa:string" value="pos;meta.main"/>
    <INSTANCE dmrole="cab_msd:Parameter.measure" dmtype="cab_msd:STCSphericalSkyPosition">
  </INSTANCE>
```

A source parameter.
The measure is folded

Moving Ahead: Roadmap

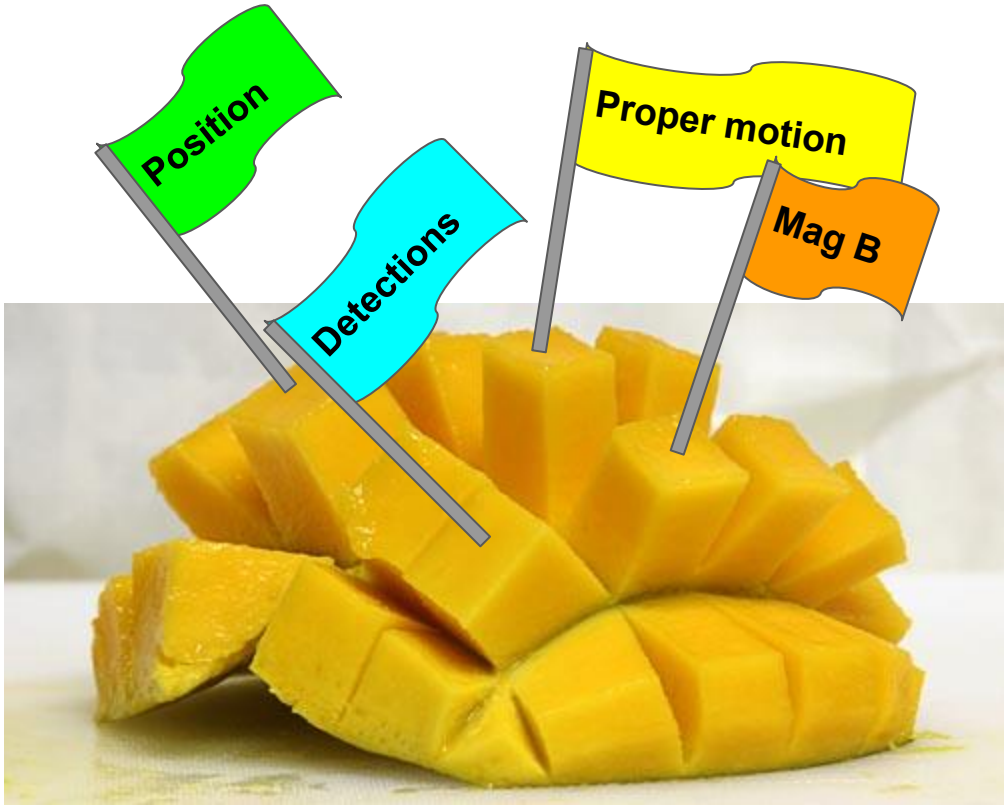
- **Attract the attention of the community for CAB-MSD**
- **Model**
  - Distribute collected use cases over imported model components and CABMSD classes
  - Continuing the WD
- **Data annotation**
  - Refining the schema of the mapping syntax
  - Continuing the WD
  - Thinking about an annotation process for TAP services
- **Client side**
  - AstroPy/PyVO interface
  - Multi language interoperability ?

References

- **Model Working Draft**
  - https://github.com/ivoa-std/CAB-MSD
- **Mapping Working Draft**
  - kind of a catch-all of things.
  - https://github.com/lmichel/vodml-lite-mapping/tree/master/doc
- **Python Workflow**
  - https://github.com/lmichel/vodml-lite-mapping/tree/master/python_workflow
  - Includes a Jupyter notebook
- **Wiki Page**
  - https://wiki.ivoa.net/twiki/bin/view/IVOA/SourceCatalogs

Don’t read beyond that point

The Model: The Fruity Version



Source Data Model

- Everything is well packed in a model instance

Model for Source Data

- All available data can be discovered and retrieved in the model instance

Why Doing a Source Model?

- **For the Science**
  - Make sure that data, scientifically relevant for a particular use case, are well described.
- **To make sure that data sent by a service will be properly understood by the clients.**
  - With a system of unambiguous annotations
  - By enabling clients to understand these annotations
  - By making sure that clients could take advantage of these annotations

The model design is not a goal in itself, it is just a (powerfull) tool