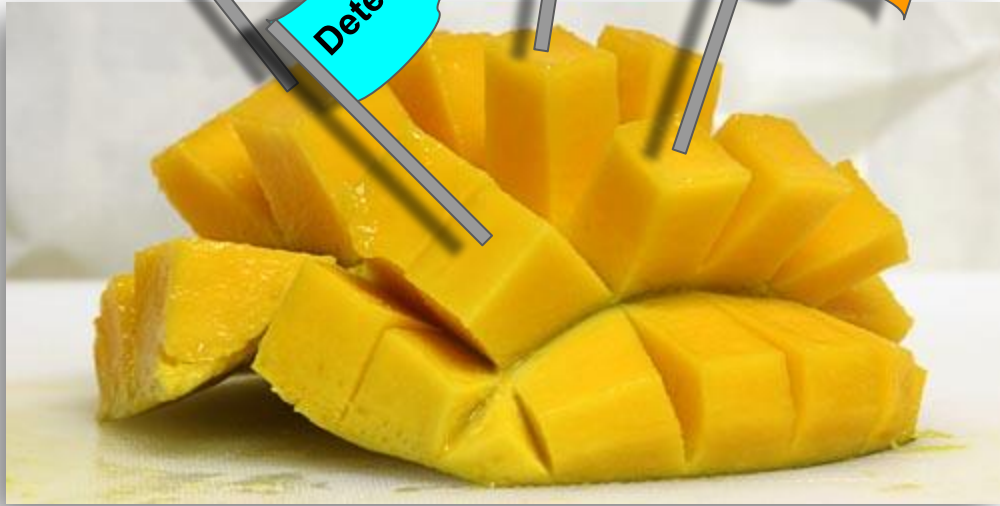


*Giving
archival catalogs
a capability
of interoperability*

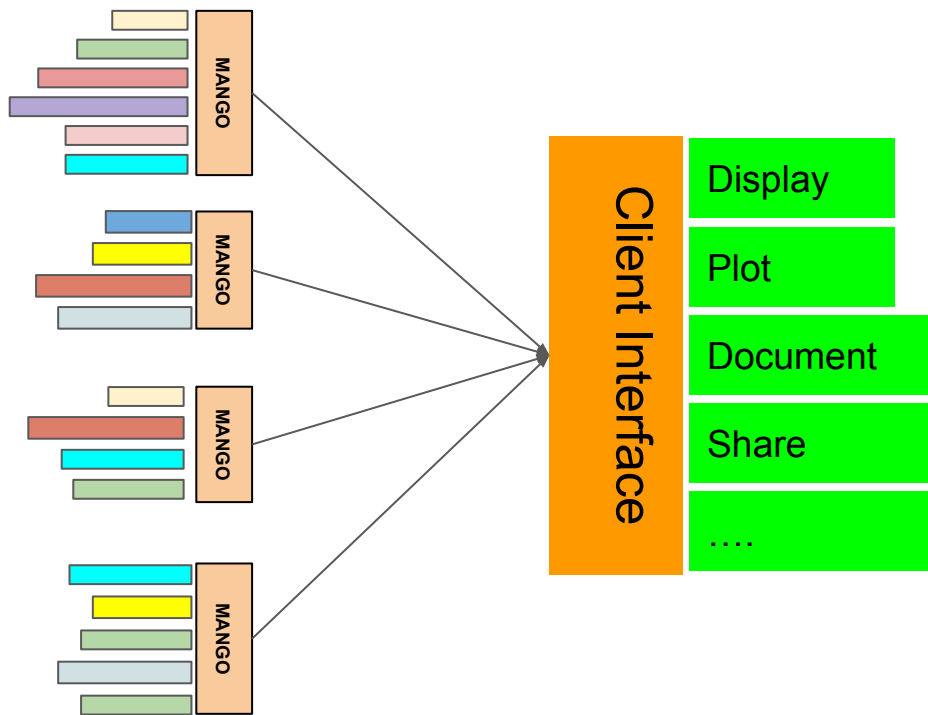


MANGO

A Model for Source Data
DM Workshop 2021

L. Michel F. Bonnarel G. Landais M. Louys M. Molinaro J. Salgado

Model for ANnotating Generic Objects



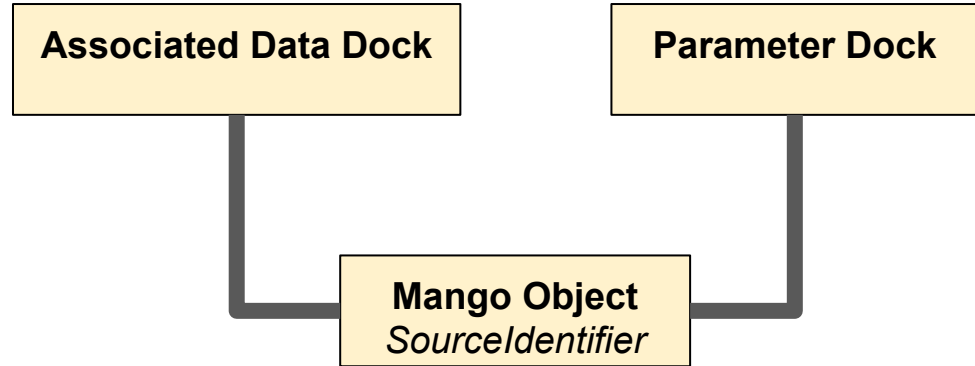
- MANGO is not for describing what sources are
- MANGO is for enhancing the description of source data that cannot be changed by the curator (archive, TAP) and that can be very heterogeneous (Vizier)

- MANGO purposes
 - **Guideline** for interpreting data attached to a source
 - **Guideline** for building **annotations**
- **Expectation**
 - Clients can get an **accurate description of parameters** whatever the data origin is.

Data Exposed by MANGO

A MANGO source is made with

- **One source identifier**
- **A set of Parameters**
 - Set of simple values (string or numerical)
 - Measurements
 - Computed parameters
 - Flags
- **A set of associated data**
 - complex data such as light curves,
 - any sort of counterparts,
 - Pointers VO services



Mango Parameters

Allow column grouping

Associated
Parameter

MANGO Parameter

Ucd
Semantic

Measure
- error

Parameter Dock

Point

ProperMotion

Time

PhotMeas

Generic Meas

Status

Label

Space Fra

Time Frame

Phot Filter

Value Enum

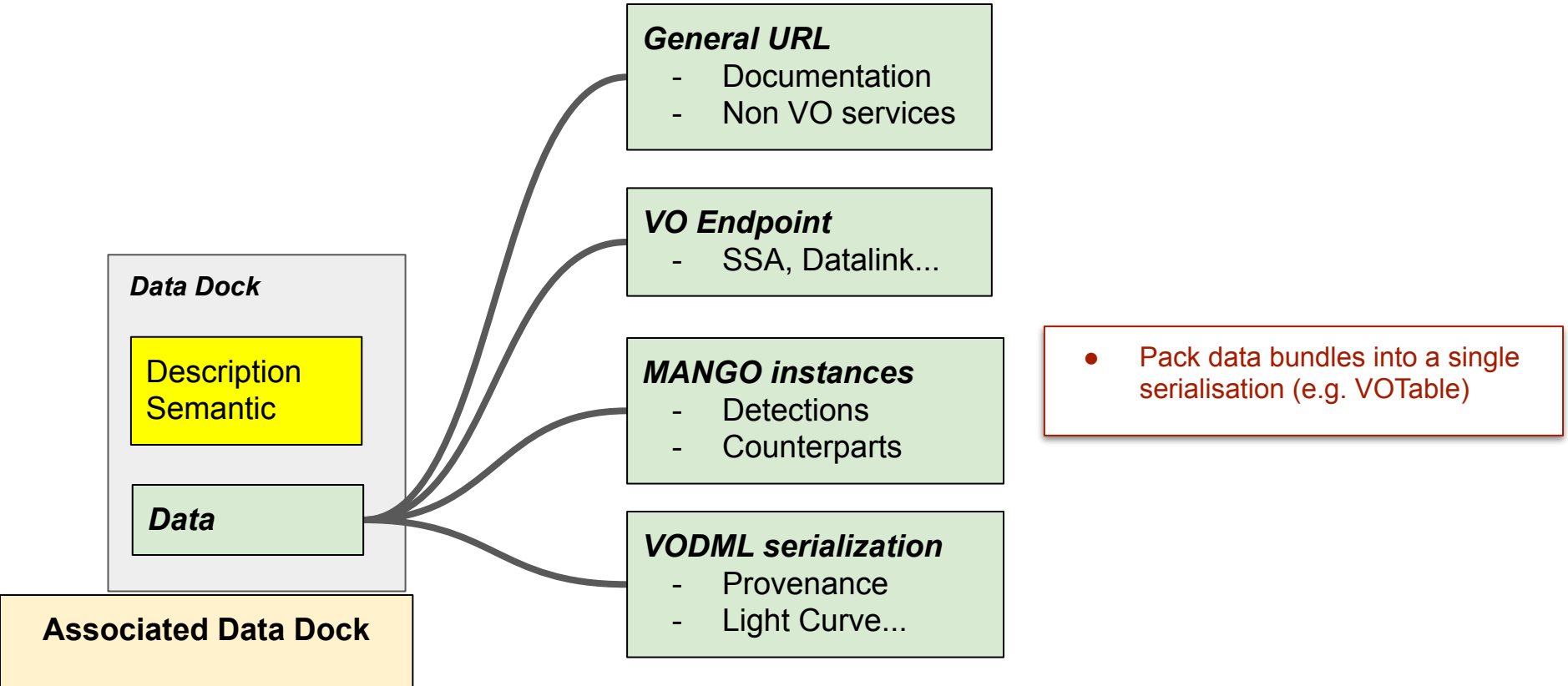
The list of per-domain
classes is still discussed

To be used for any
quantity having no
specialised class

Physical meaning
given by the
Parameter UCD

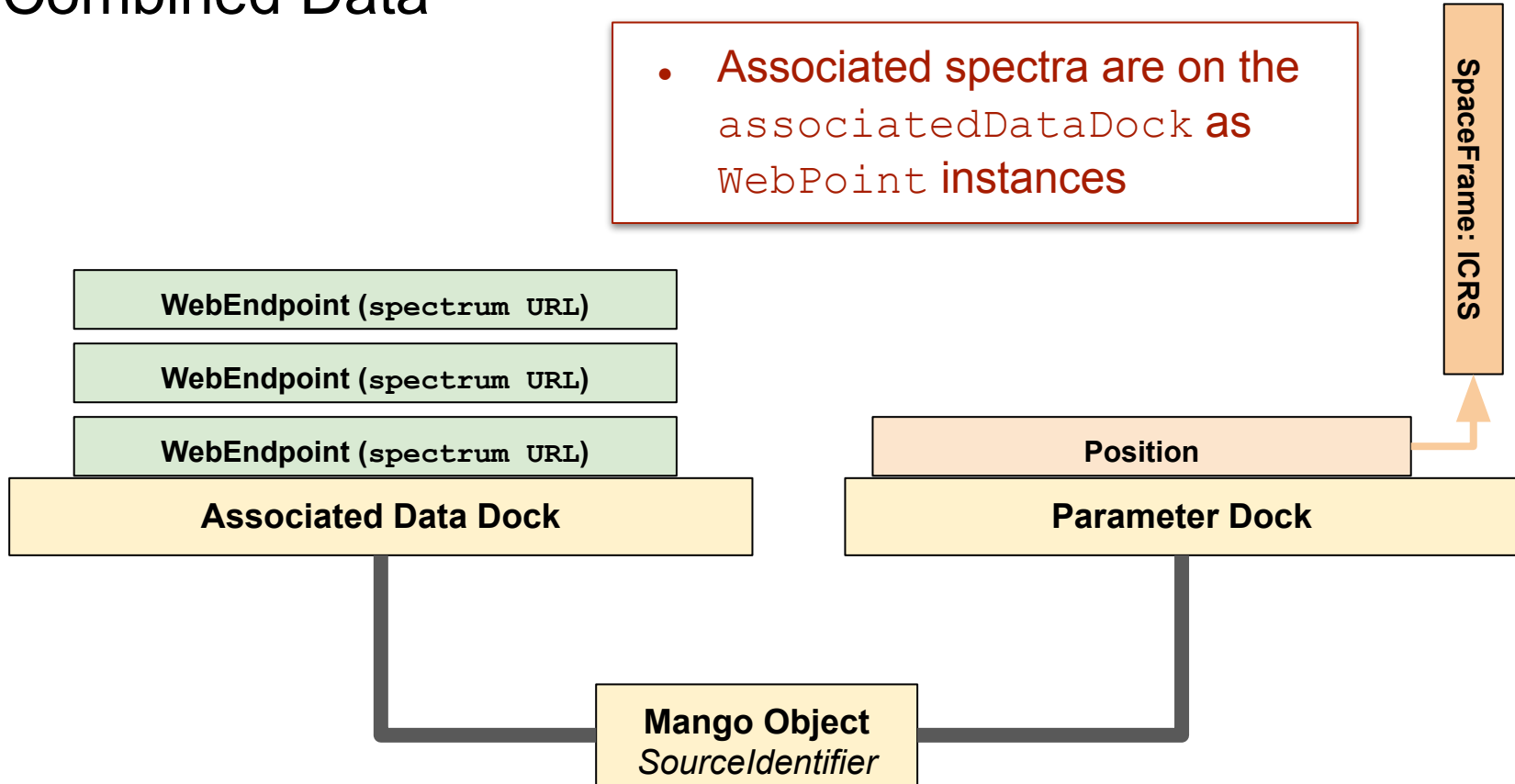
Domain related measures

Associated Data



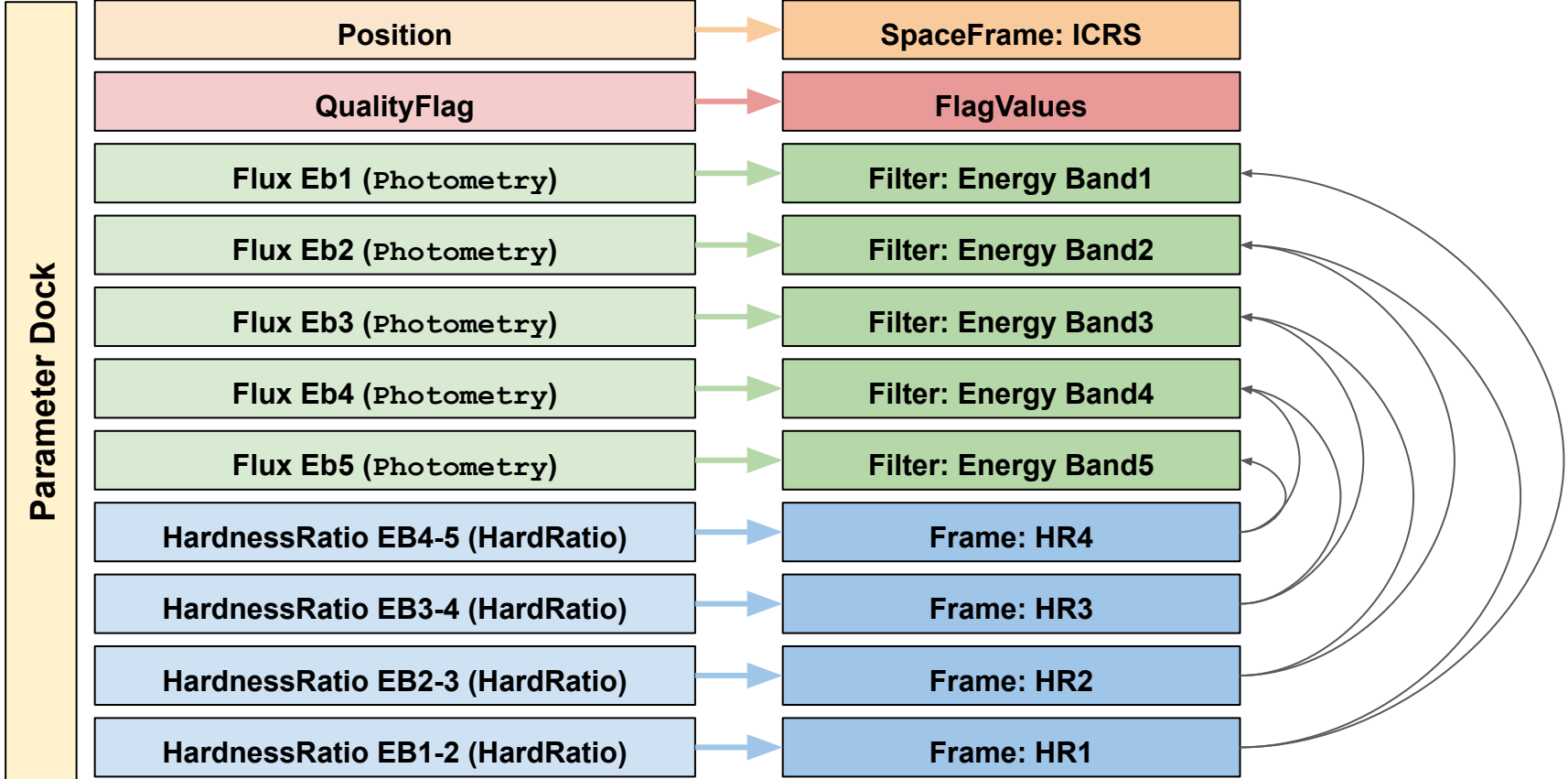
Combined Data

- Associated spectra are on the `associatedDataDock` as `WebPoint` instances

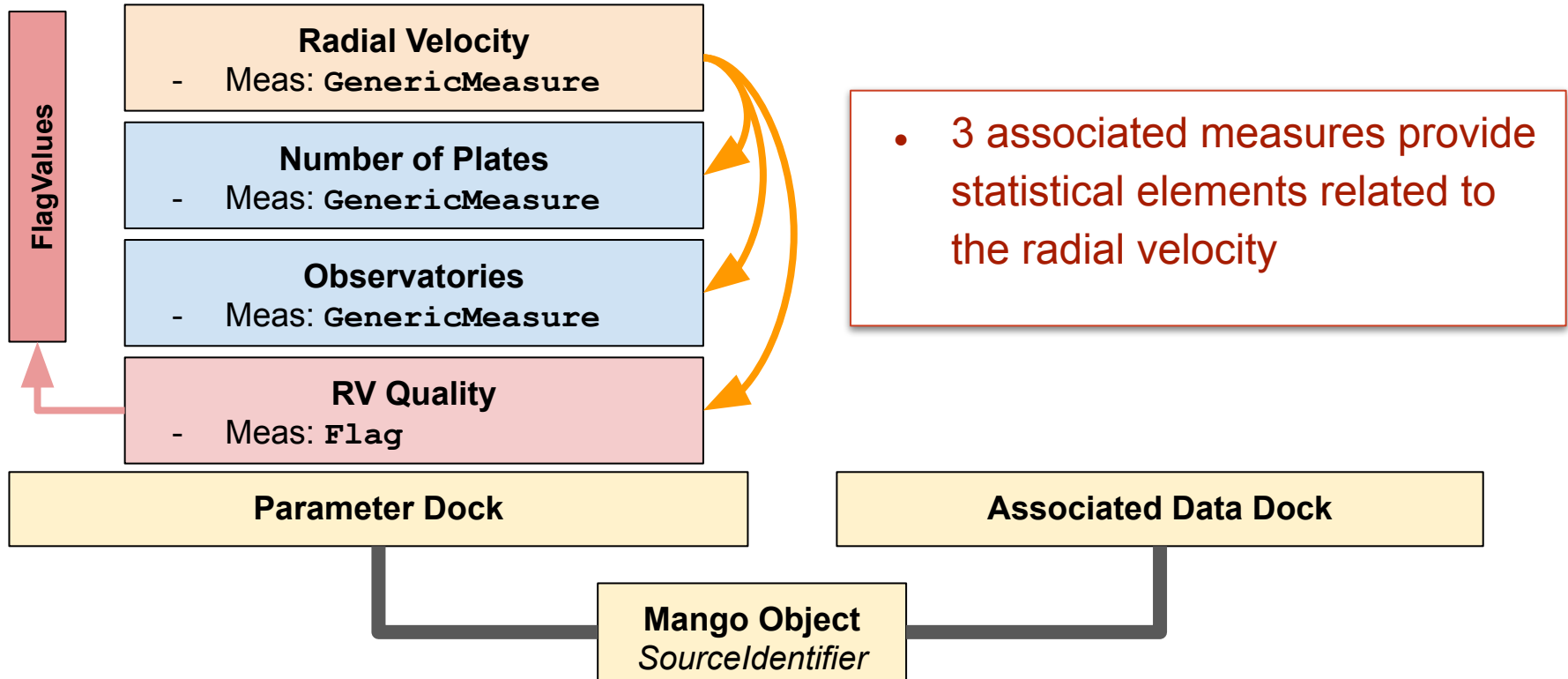


Standard Properties

Extended Coord Frames



Column Grouping



Mapping Syntax ModelInstanceInVot

- **#1** Can 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** readability
 - Compactness
 - Human readability
- **#3** Parser helper: Can be used at different levels
 - Just get the type of the VOTable content
 - Just get meta data
 - Get everything through the model

```
<TABLE_MAPPING tableref='Results'>
  <!--
  The TABLE contains one time series
  -->
  <INSTANCE dmrole="root" dmtpe="mock_ts:TimeSeries">
    <!--
    TS meta data
    -->
    <INSTANCE dmrole="mock_ts:TimeSeries.dataSet" dmtpe="mock_ts:DataSet">
      <!--
      The TS in a collection of light curves
      -->
      <COLLECTION dmrole="mock_ts:TimeSeries.lightCurves">
        <!--
        Light curve for B band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <!--
          Light curve meta data: Name + filter
          -->
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve G band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiag" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band G
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
        <!--
        Light curve for RP band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve RP band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiarp" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band R
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
        <!--
        Light curve for BP band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve BP band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiabp" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band B
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
      </COLLECTION>
    </INSTANCE>
  </TABLE_MAPPING>
```

Mapping of the data enclosed in the *Results* table

Time Series mapping

Time Series meta-data (folded)

The TS contains several light curves

Mapping of one light curve

Light curve meta-data

Photometric points mapping with a (folded) table iterator that contains a filter

- Maps both table PARAMS and table DATA
- 3 table row iterators populating the same instance

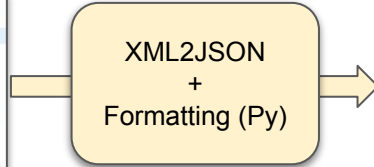
API: Proof of Concept

- **Non normative: based on dictionaries**

- Easy to process on many languages
- Model roles used as keys

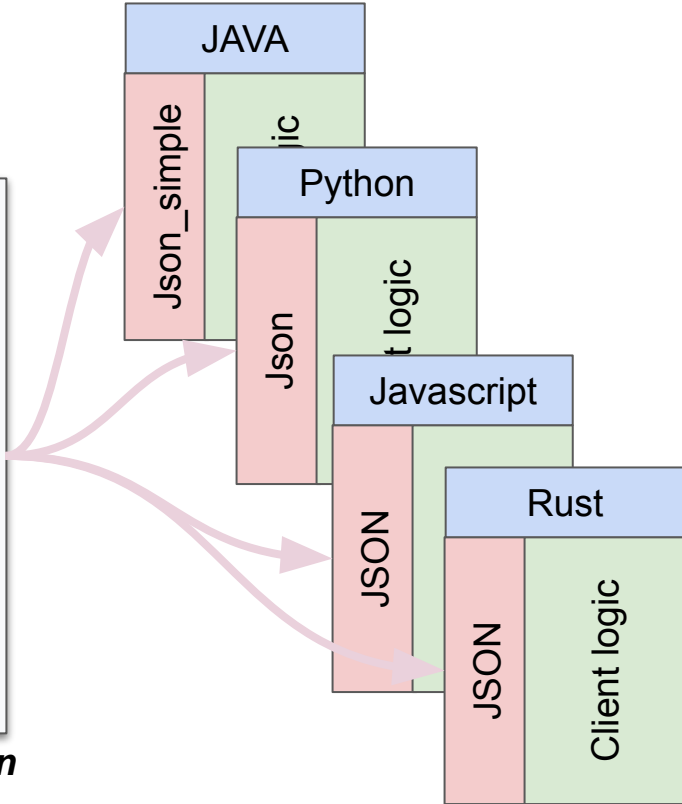
```
<INSTANCE dmrole="mango:Parameter.measure"  
  dntype="mango:Parameter">  
  <ATTRIBUTE dmrole="mango:Parameter.semantic"  
    dntype="ivoa:string" value="#position" />  
  <ATTRIBUTE dmrole="mango:Parameter.ucd"  
    dntype="ivoa:string" value="pos.eq;meta.main" />  
  <ATTRIBUTE dmrole="mango:Parameter.description"  
    dntype="ivoa:string" value="this is the position" />  
</INSTANCE dmrole="mango:Parameter.measure"  
</INSTANCE>
```

XML mapping block



```
"#1_pos": {  
  "coord_type": "mango:stcextend.LonLatPoint",  
  "coords:SpaceFrame": {  
    "@ID": "SpaceFrame_ICRS",  
    "@dntype": "coords:SpaceFrame",  
    "coords:SpaceFrame.equinox": {  
      "@dntype": "coords:Epoch",  
      "@value": "NoSet"  
    },  
    "coords:SpaceFrame.refPosition": {  
      "@dntype": "coords:StoRefLocation",  
      "coords:StoRefLocation.position": {  
        "@dntype": "ivoa:string",  
        "@value": "NoSet"  
      }  
    },  
    "coords:SpaceFrame.spaceRefFrame": {  
      "@dntype": "ivoa:string",  
      "@value": "ICRS"  
    },  
    "coosys_type": "coords:SpaceFrame",  
    "description": "Corrected position",  
    "error_type": "meas:Error",  
    "mango:stcextend.LonLatSkyPosition": {  
      "field:latitude": {  
        "id": "_dec_147",  
        "index": 1  
      },  
      "field:longitude": {  
        "id": "_ra_146",  
        "index": 0  
      },  
      "meas:Error": {  
        "field:meas:Symmetrical.radius": {  
          "id": "_poserr_148",  
          "index": 2  
        },  
        "unit": "NotSet"  
      },  
      "measure_type": "mango:stcextend.LonLatSkyPosition",  
      "semantic": "position.corrected",  
      "ucd": "pos"  
    },  
  },  
,
```

JSON serialization
Keys are DM roles



API: Output keep connected to native data

Extracting a position from MANGO annotation

```
mango_data = mango_browser.get_data(measure_type="mango:stcextend.LonLatSkyPosition")  
DictUtils.print_pretty_json(mango_data)
```

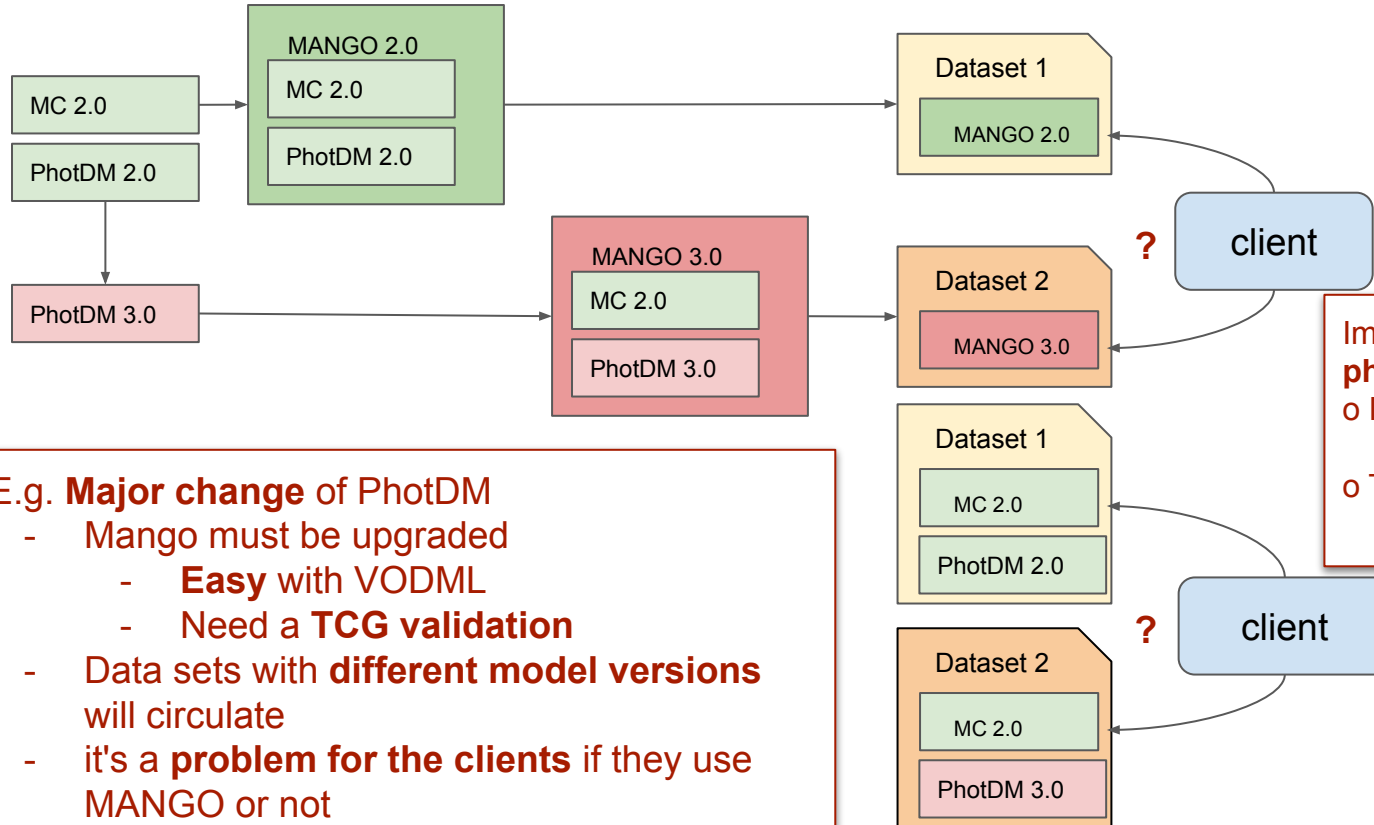
```
{  
  "data": [  
    [ 340.91055060369,  
      -17.071667101891,  
      1.50765 ]  
  ],  
  "head": [  
    "field:longitude [#1 pos]",  
    "field:latitude [#1 pos]",  
    "error: field:meas:Symmetrical.radius [#1 pos]"  
  ],  
  "selected_index": [  
    0,  
    1,  
    2  
  ]  
}
```

Sky position read (limited here to one row)

Model attribute references
Labels can be used as keys to get more information

VOTable column Indices

Impact of major model change



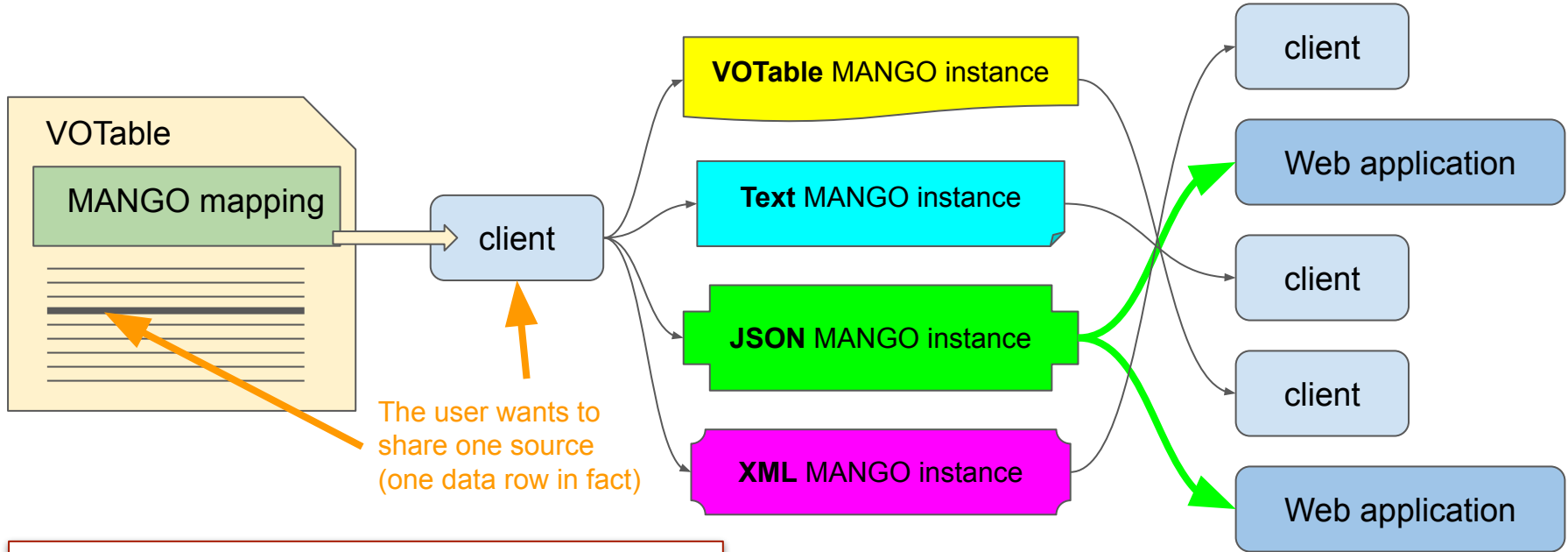
E.g. Major change of PhotDM

- Mango must be upgraded
 - **Easy** with VODML
 - Need a **TCG validation**
- Data sets with **different model versions** will circulate
- it's a **problem for the clients** if they use MANGO or not

Imported models describe **physical quantities**

- o Most **likely minor changes**
 - Adding something
- o Than major ones breaking backward compatibility

Sharing MANGO instances



- o Using an **integrated model** facilitate the data sharing (e.g.SAMP) with **different serialisation** modes
- o **Less easy** with **sparse model components**

Why Using Mango Docks?

- **Unentangled models:** All mapping elements listed at the same level

- Just provide a bag of Lego bricks



- OK as long as bricks have all the same nature (.e.g measures)
- May become confusing when brick get heterogeneous
 - Measure, join keys, columns groups, axis description ...

- **Mango docks:** Model elements located at the right placeholder in the docks

- Provide a Lego construction within which users know where are searched things



- Note really more complicated than the unentangled mode
- Comfortable with complex data patterns as long as they are taken into account by the model
- Can be exchanged with any serialization mode.

Status and Prospect

- **Mango**

- WD on GitHub
- Some feature requests risen for the workshop

- **Mapping syntax ModellInstanceInVot**

- WD on Github
- XML schema ready
- Many unit tests
- Annoter helper prototype
- A few missing features pointed out by workshop discussions

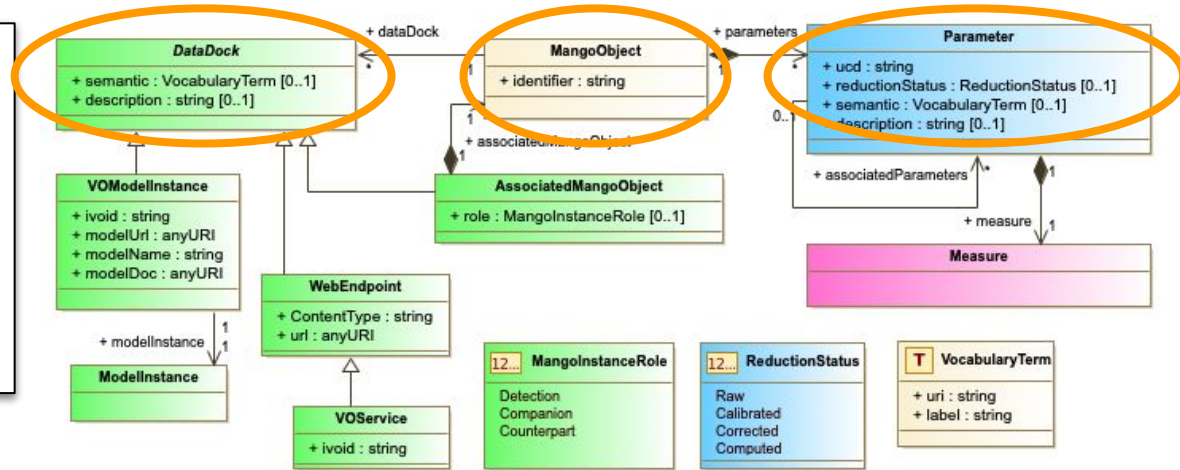
- **API**

- Just a proof of concept so far
- Works with many examples
- Needs both MANGO and mapping syntax to be adopted to be consolidated

Mango Skeleton

3 components

- **One source identifier**
- **2 Docks**
 - The content of the docks are not defined by the model
 - The model lists possible objects that can be attached to a dock

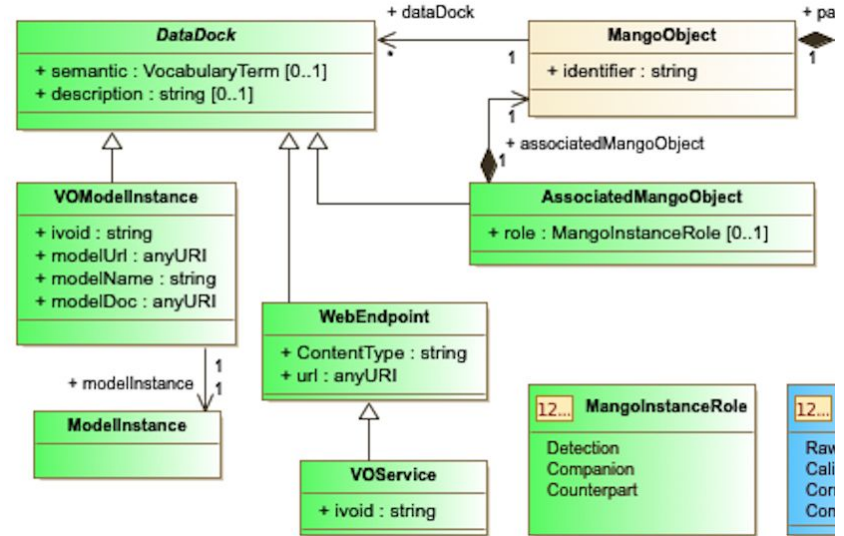


Docks are open ended data containers

- The model describes which quantities that can be dropped off on a dock
- It does not say which ones have to be there or not
- The content of the docks varies from a dataset to another
- We can have several instances of the same quantity on a dock
 - Multiple positions
 - Multiple counterparts

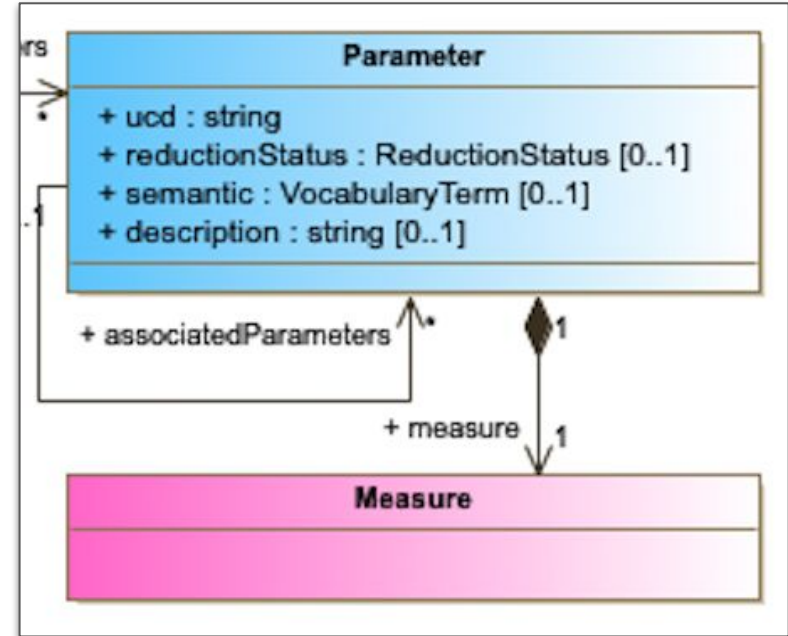
Associated Data

- **URL (VO or not)**
 - VO endpoints
 - Datalinks, SIA, SSA
 - Serving data related to that source
 - General purpose URLs
 - Documentation
 - Non VO tools
- **Other Mango Instances**
 - Source detections
 - Counterparts in other datasets
- **VODML serialized objects**
 - Light curves
 - Provenance
- **Purpose**
 - Pack data bundles on a VOTable and to retrieve them
 - Can be replaced with Datalinks



Mango Parameter

- **A MANGO parameter is an object that can be dropped on the parameter dock**
- **Parameters have 2 components**
 - A semantic block giving the role of the measure
 - UCD
 - Reduction status (Model enumeration)
 - Vocabulary entry (label + URI)
 - Text description
 - instance of class deriving from the abstract Measure class
 - See after
- **Extended usage of the UCDs**
 - UCD scope extended to complex values (values +error)
 - Example: pos;meta.main gives the role of a sky position measure, along with its values, errors and frame.

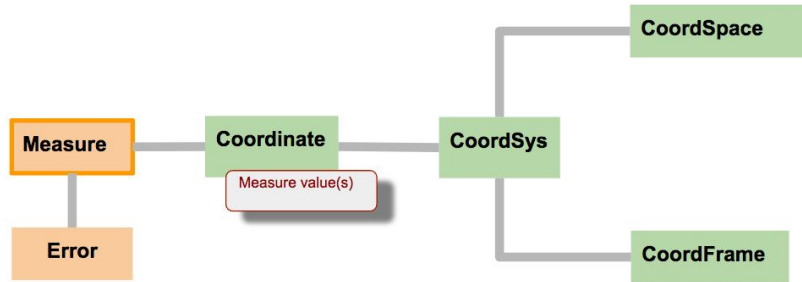


Measure

- **Broad measure taxonomy**
 - Still in expansion
 - Can be measurements or computed values or annotation
- **The use of different measurements in archival data is very variable**
 - Some are very common (**position**)
 - Some are less usual (**shape, flag**)
 - Some can be duplicated in the docks (**multipositions, multiple bands...**)
 - Some are definitely exotic ("**decimal log of surface gravity**")
- **A modeling effort weighted by that usage rate**
 - The most popular are modeled by specific classes
 - The less popular can be described by generic objects
 - No trouble with adding new quantities that have not been foreseen by the model

Measure Classes

- **All built on the Meas/Coordinates (MC) pattern**
 - `Meas = coord + error`,
 - Using the same pattern for any measure facilitates parsing



- **Using MC classes when possible ...**
 - `Meas` concrete classes are reused as they are.
 - All measure that have no dedicated classes can be handled by generic `Meas:GenericMeas`
- **... or extending MC classes**
 - Mango proposes a set of classes that are not in `Meas:Measure`
 - Some are necessary (Luminosity, shape, quality flag)
 - The others still must be discussed
- **Using a specific class when available is not mandatory**
 - A magnitude can be presented with a generic measure.
 - The semantic block allows us to properly identify the quantity.

Measurement Core Model Extension

Measure Model

Position
Time
Velocity
ProperMotion
Polarization
GenericMeasure

Measure

Mango MC Extensions

Photometry
Flag
Shape
Orbit
ObjectType
HardnessRatio
Temperature
Redshift
LonLatPosition

PhotometryCoordSys

FlagSys

ShapeSys

ObjectTypeSys

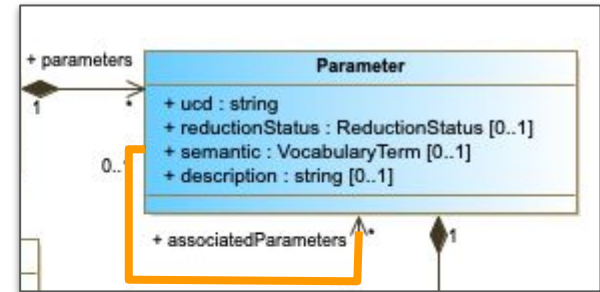
HardnessRatioSys

RedshiftSys

Some measures that are for now in the MANGO MC extension could be added to the core Measurement model depending on the general usefulness of the extended measures in other contexts.

Measure Sets: A Vizier Request

- **Vizier data need a way to group columns**
 - Grouping columns around a particular quantity
 - Vizier J/MNRAS/392/19/2slagqso
 - Redshift + quality
 - Spectroscopic type + flag
 - Gaia
 - Position + proper motion to handle complex errors (Xmatch - work in progress)
 - Such groups have no semantic but **is-related-to**
- **Likely one of the most important features to annotate Vizier data**
 - Implemented by the ***associatedParameters*** relation



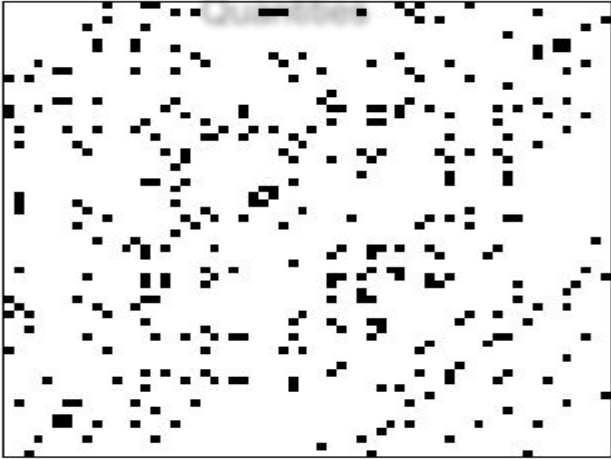
Describing a Source: an Endless Job number of use-cases

- **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 counterparts....

Motivation for a Source Model

Quantities

Archival Catalogs



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**

Other Features

Shortcuts: Model components that are parts of a standard can be folded in compact XML elements

```
<INSTANCE dmrole="coords:Point.axis1" dmtpe="ivoa:RealQuantity">  
  <ATTRIBUTE dmrole="ivoa:RealQuantity.value" dmtpe="ivoa:real" ref="RA_ICRS"/>  
  <ATTRIBUTE dmrole="ivoa:Quantity.unit" dmtpe="ivoa:Unit" value="deg"/>  
</INSTANCE>
```



```
<SC_REALQUANTITY dmrole="coords:Point.axis1"  
  ref="RA_ICRS" unit="deg" />
```

Row filtering: Only processing data with a certain field value

```
<COLLECTION dmrole="dmrole">  
  <TABLE_ROW_TEMPLATE>  
    <FILTER ref="ref" value="value"/>  
    <INSTANCE dmref="dmref" dmrole="dmrole" />  
  </TABLE_ROW_TEMPLATE>  
</COLLECTION>
```

Foreign keys: Joining data from different tables

```
<TABLE_MAPPING tableref="fgdgfddf">  
  <COLLECTION dmrole="dmrole">  
    <JOIN tableref="tableref" primary="primary" foreign="foreign">  
      <INSTANCE dmref="dmref" dmrole="dmrole" />  
    </JOIN>  
  </COLLECTION>  
</TABLE_MAPPING>
```

Row grouping: Grouping data of the same source spread over multiple rows

```
<GROUPBY ref="ref">  
  <INSTANCE dmref="dmref" dmrole="dmrole">  
    <ATTRIBUTE ref="ref" dmrole="dmrole" dmtpe="dmtpe"  
      value="dmvalue" />  
  </INSTANCE>  
</GROUPBY>
```

One mapping block per <TABLE>

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ivoa.net/xml/VOTa
  <DESCRIPTION><![CDATA[
  <COOSYS ID="ICRS" system="ICRS" />
  <RESOURCE type="results">
    <DESCRIPTION>Native Saada</DESCRIPTION>
    <INFO name="QUERY_STATUS" value="OK" />
    <TABLE name="Results">
    <TABLE name="Spectra">
  </RESOURCE>
</VOTABLE>
```

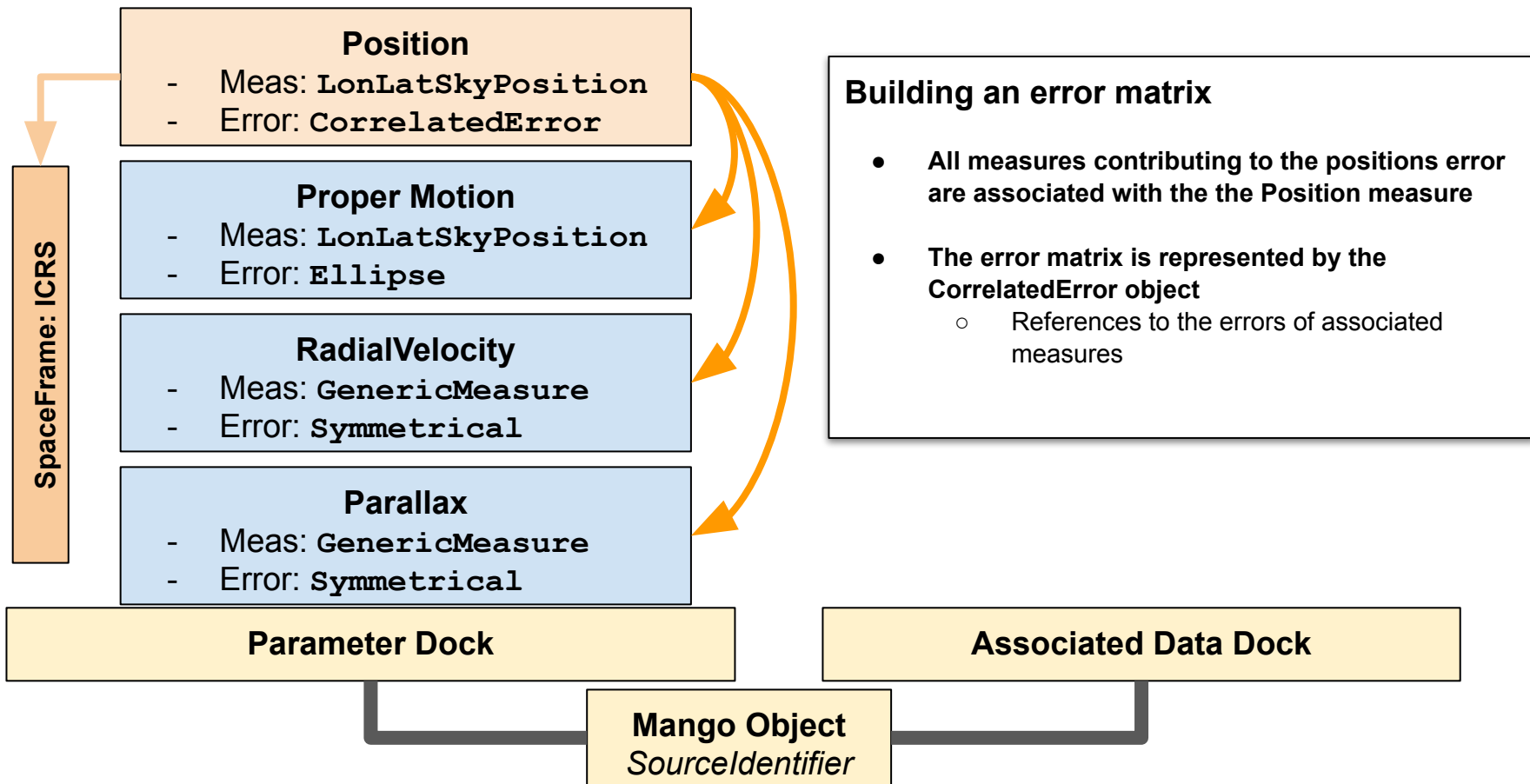
Raw VOTable (folded)

```
<MODEL_INSTANCE name="MANGO" syntax="ModelInstanceInVot" >
  <GLOBALS>
  <TABLE_MAPPING tableref="Results">
  <TABLE_MAPPING tableref="Spectra">
</MODEL_INSTANCE>
```

Mapping block (folded)

- One mapping block for each <TABLE>
- Such placeholder makes easier the mapping insertion

Precise Astrometry



Motivation for a Model for Source Data

- **What is missing?**
 - a clean representation of the **coordinate systems**
 - a clean representation of the **complex errors**
 - Support **cross-columns parameters**
 - Columns grouping
 - Complex errors (pos + pm + parallax)
 - Capacity of Gathering data from **multi-table** VOTable (sources + detection)

Motivation for a Model for Source Data

- **Catalog data are well supported by VOTables**
 - VOTable has been designed for this purpose!
- **Data column well defined**
 - UCD, format unit, utype
 - Global parameters (PARAM)
- **Possible column associations**
 - GROUPs
- **Coordinate Systems**
 - Time/space
- **Human readable information**
 - INFO
 - DESCRIPTION
- **Plenty of others features to discover within the 32 pages of the standard**

- **Anything we need to get a good interoperability level!**
- **Do we need more?**