

# The MIVOT Stack

-  
Laurent MICHEL on the behalf of all DMers

# Results of Previous Episodes

Mivot approach for the Epoch propagation    L. Michel  
Presentation of 4 approach to represent information of COOSYS

Tucson 2024  
Closing remarks

VOTable 1.5 Status    T. Donaldson  
Presentations of evolution and discussion on the process  
we go for VOTable 1.5 and let COOSYS to 1.6

[https://wiki.ivoa.net/internal/IVOA/InterOpNov2023CloseTCG/apps\\_cloture.pdf](https://wiki.ivoa.net/internal/IVOA/InterOpNov2023CloseTCG/apps_cloture.pdf)

- Engagement with the community

- Continuous work on PyVO:

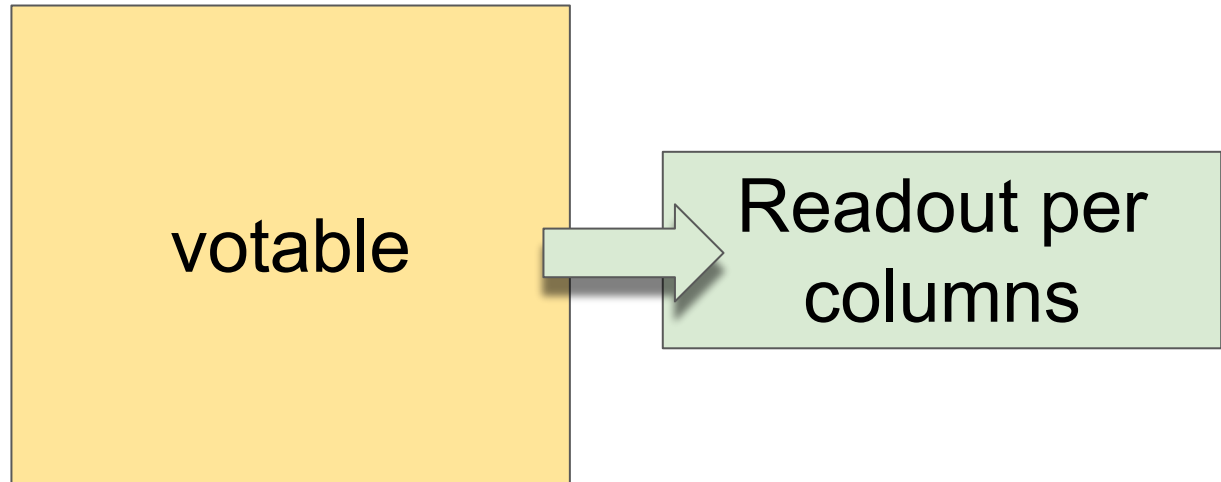
- Eg. improve the discoverability of astronomical data through the Registry in an end user client such as PyVO

- Support coordination activities to implement the **MIVOT** feature into astropy and PyVO Python packages

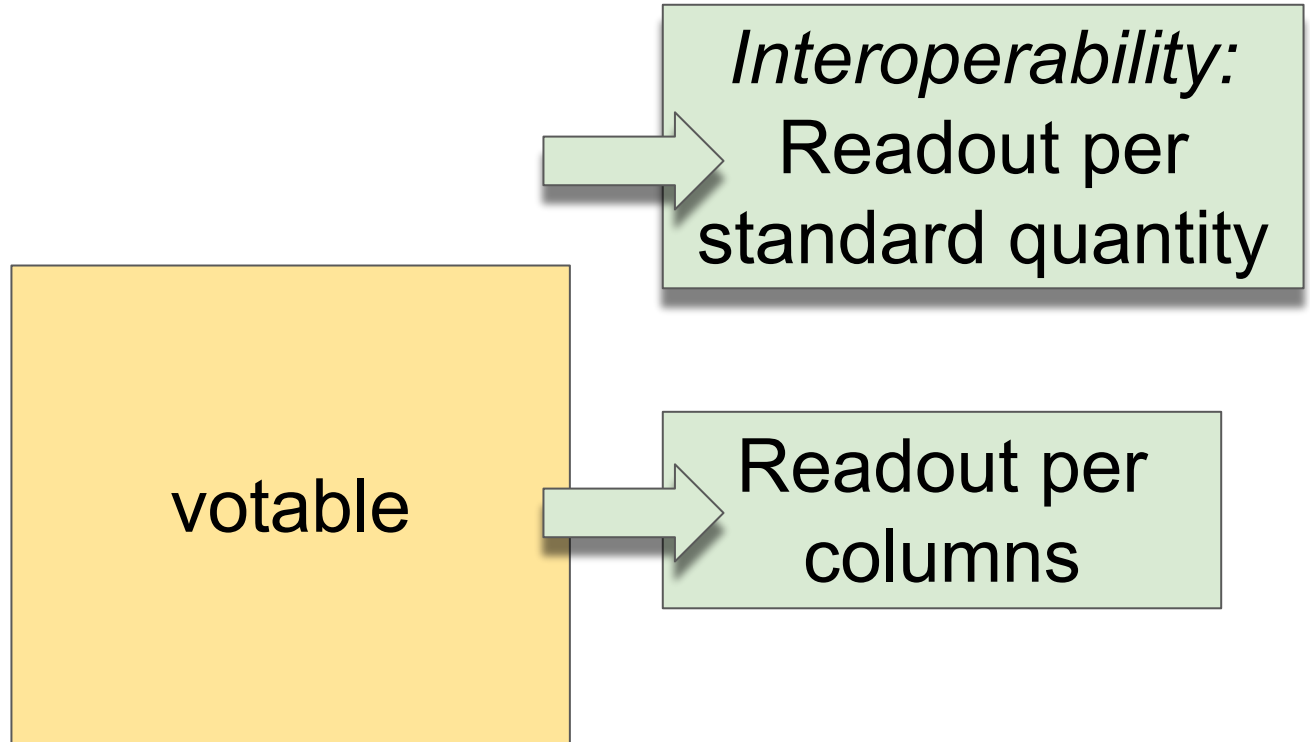
CSP  
Sydney 2024

[https://wiki.ivoa.net/internal/IVOA/InterOpMay2024/CSP-Presentation-interop-2024A\\_FC\\_2.pdf](https://wiki.ivoa.net/internal/IVOA/InterOpMay2024/CSP-Presentation-interop-2024A_FC_2.pdf)

# Mivot in one slide



# Mivot in one slide



# Mivot in one slide

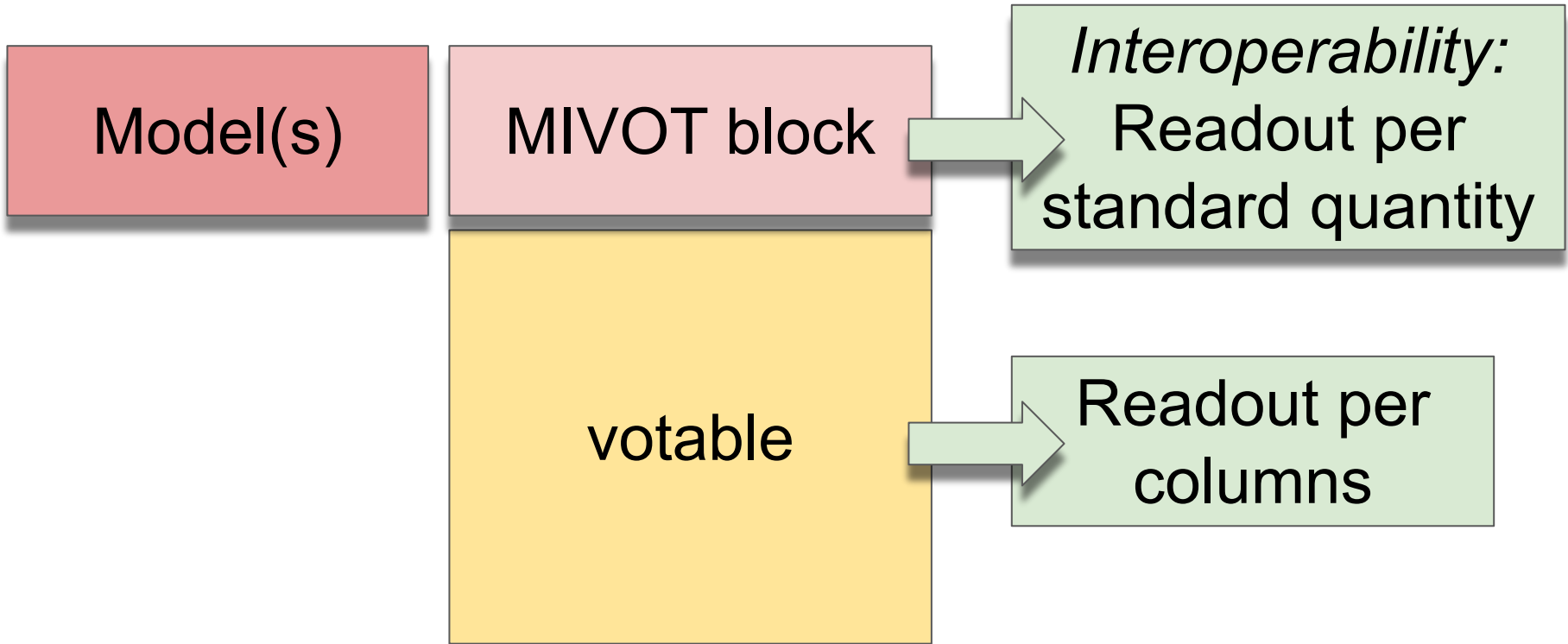
Model(s)

votable

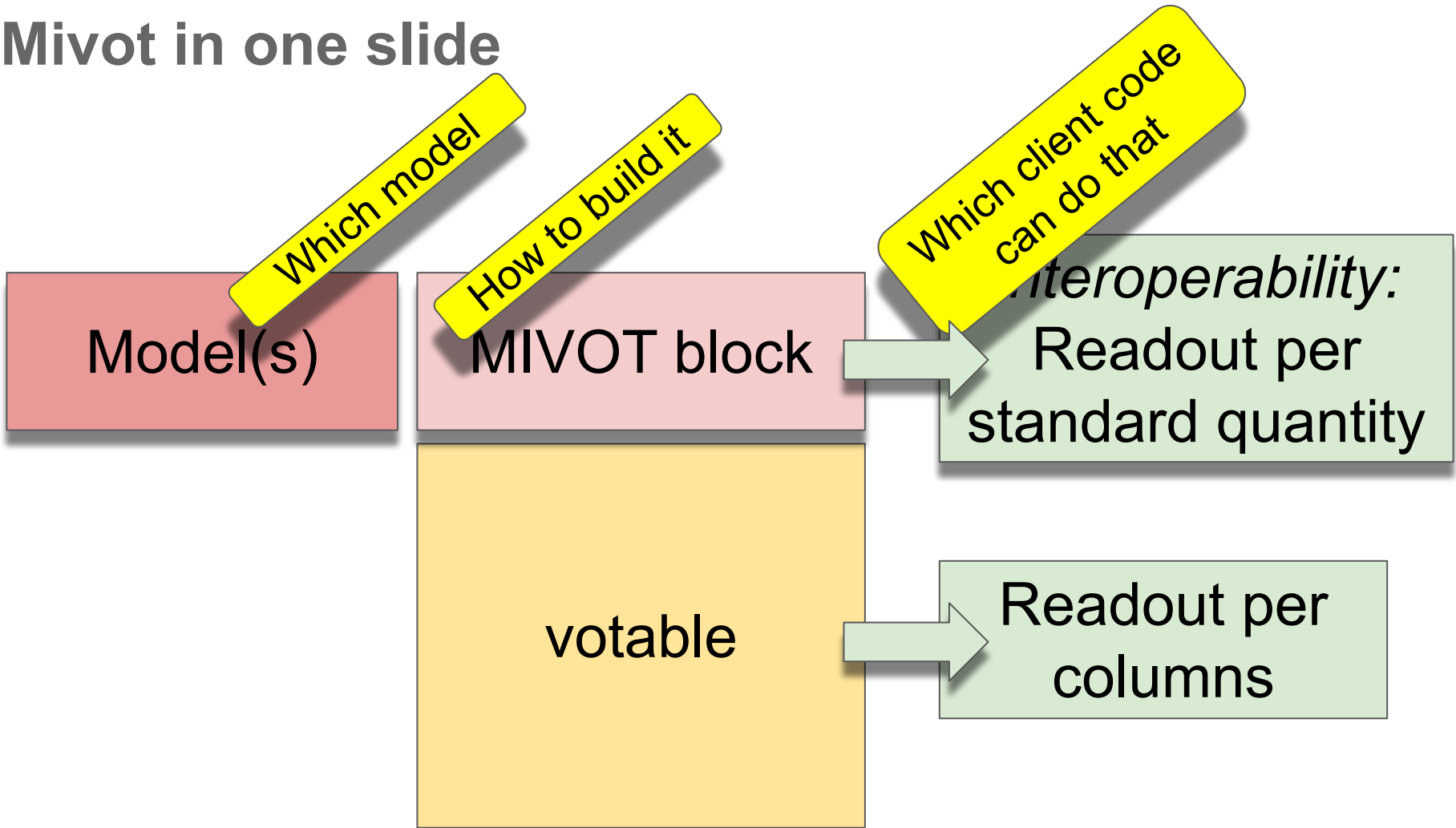
*Interoperability:*  
Readout per  
standard quantity

Readout per  
columns

# Mivot in one slide



# Mivot in one slide



# MIVOT: Mapping Block above the Data Table

- The space coordinate system is a **GLOBAL** object that can be referenced by any other MIVOT element

- Each table row can be interpreted as an instance of the class **EpochPosition** of the MANGO model

```
<VODML xmlns:dm-mapping="http://www.ivoa.net/xml/mivot">
  <REPORT status="OK">hand-made mapping</REPORT>
  <MODEL name="meas" url="https://www.ivoa.net/xml/Meas/20200908/Meas-v1.0.vo-dml.xml" />
  <MODEL name="coords" url="https://www.ivoa.net/xml/STC/20200908/Coords-v1.0.vo-dml.xml" />
  <MODEL name="mango" />
  <MODEL name="ivoa" url="https://www.ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml" />
  <GLOBALS>
    <INSTANCE dmid="SpaceFrame_ICRS" dmtpe="coords:SpaceSys">
      <INSTANCE dmrole="coords:PhysicalCoordSys.frame" dmtpe="coords:SpaceFrame">
        <INSTANCE dmrole="coords:SpaceFrame.refPosition" dmtpe="coords:StdRefLocation">
          <ATTRIBUTE dmrole="coords:StdRefLocation.position" dmtpe="ivoa:string" value="NoSet" />
        </INSTANCE>
        <ATTRIBUTE dmrole="coords:SpaceFrame.spaceRefFrame" dmtpe="ivoa:string" value="ICRS" />
      </INSTANCE>
    </INSTANCE>
  </GLOBALS>
  <TEMPLATES>
    <INSTANCE dmtpe="mango:EpochPosition">
      <REFERENCE dmrole="coords:Coordinate.coosys" dmref="SpaceFrame_ICRS"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.longitude" dmtpe="ivoa:RealQuantity" ref="pos_RA"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.latitude" dmtpe="ivoa:RealQuantity" ref="pos_DEC" />
      <ATTRIBUTE dmrole="mango:EpochPosition.pmLongitude" dmtpe="ivoa:RealQuantity" ref="pm_RA" />
      <ATTRIBUTE dmrole="mango:EpochPosition.pmLatitude" dmtpe="ivoa:RealQuantity" ref="pm_DEC"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.pmCosDeltApplied" dmtpe="ivoa:boolean" value="true"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.radialVelocity" dmtpe="ivoa:RealQuantity" ref="RV"/>
      <ATTRIBUTE dmrole="mango:EpochPosition.parallax" dmtpe="ivoa:RealQuantity" ref="PARALLAX" />
      <ATTRIBUTE dmrole="mango:EpochPosition.epoch" dmtpe="coords:Epoch" value="J2016.0" unit="year"/>
    </INSTANCE>
  </TEMPLATES>
</VODML>
```



## **MODEL**

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

## **SERVER SIDE**

L. Michel, G. Landais, G. Mantelet, I. Errami, Carlos Rodrigo (rip)

## **CLIENT SIDE**

L. Michel, F.X. Pineau, J.Abid, S. Floret, M. Marchand, M. Taylor, Astropy team

## **VALIDATOR**

L. Michel, J.Abid, M. Louys, F. Bonnarel,

## **ONLINE DOCUMENTATION**

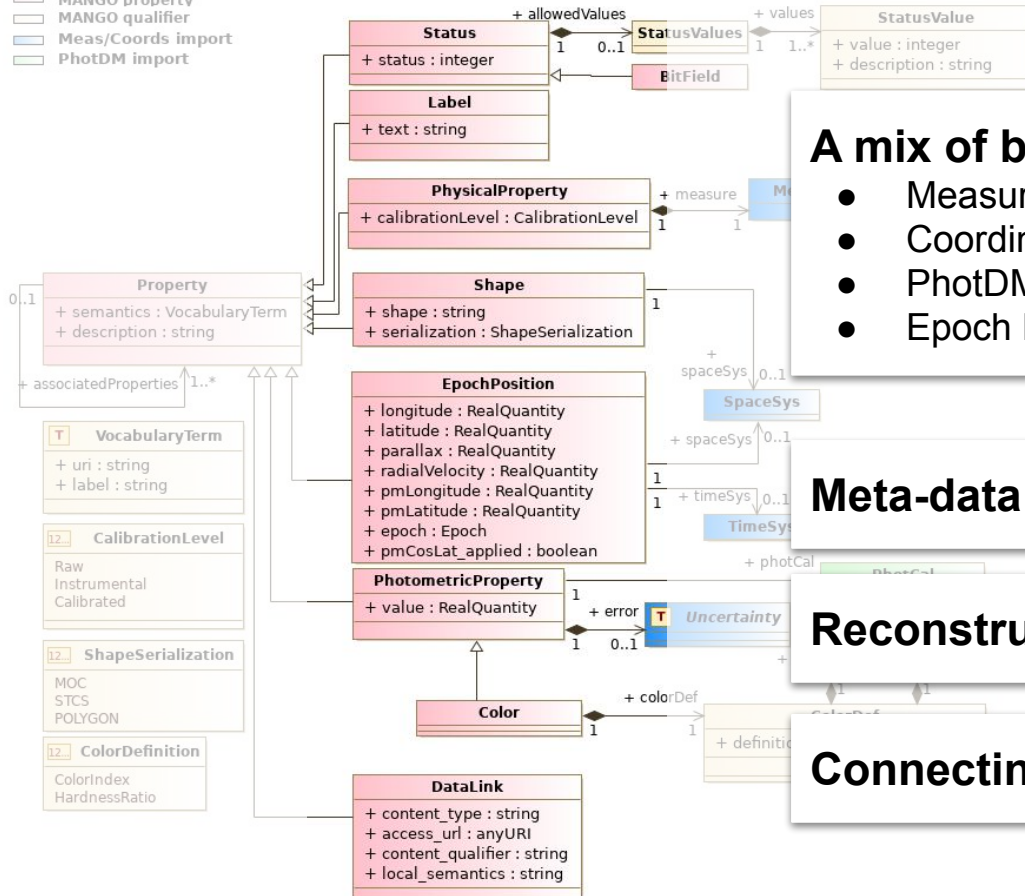
L. Michel, J.Abid, M. Louys, F. Bonnarel, D. Durand

## **MISCELLANEOUS**

L. Michel, T. Boch, P. Fernique, M. Louys, F. Bonnarel, C. Nogueira

# MODEL: MANGO: An object Container

- MANGO property
- MANGO qualifier
- Meas/Coords import
- PhotDM import



## A mix of built-in and imported classes

- Measurement
- Coordinates
- PhotDM
- Epoch Propagation

## Meta-data enhancement

## Reconstruction of multi-columns quantities

## Connecting objects to each other

## SERVER SIDE: Vizier Cone Search

### Maps query responses on the `EpochPosition` MANGO class

- Maps epoch, positions and proper motions in a single object

### Works on any Vizier catalogue

- [https://cdsarc.cds.unistra.fr/beta/viz-bin/mivotconesearch/TABLE\\_ID](https://cdsarc.cds.unistra.fr/beta/viz-bin/mivotconesearch/TABLE_ID)
  - Ex: TABLE\_ID = I/239/hip\_main

### Used to validate client code ad validator

```
% curl 'https://cds/viz-bin/mivotconesearch/I/329/urat1?RA=52.26708&DEC=59.94027&SR=0.05'
```

## SERVER SIDE: XTapDB

### Serve the 4XMMdr14 catalogue

### Maps query responses on different MANGO classes

- Maps Positions, flux, hardness ratio, detection flags in a single object
- Only maps the queried columns

URL: <https://xcatdb.unistra.fr/xtapdb/sync>

#### PARAMETERS:

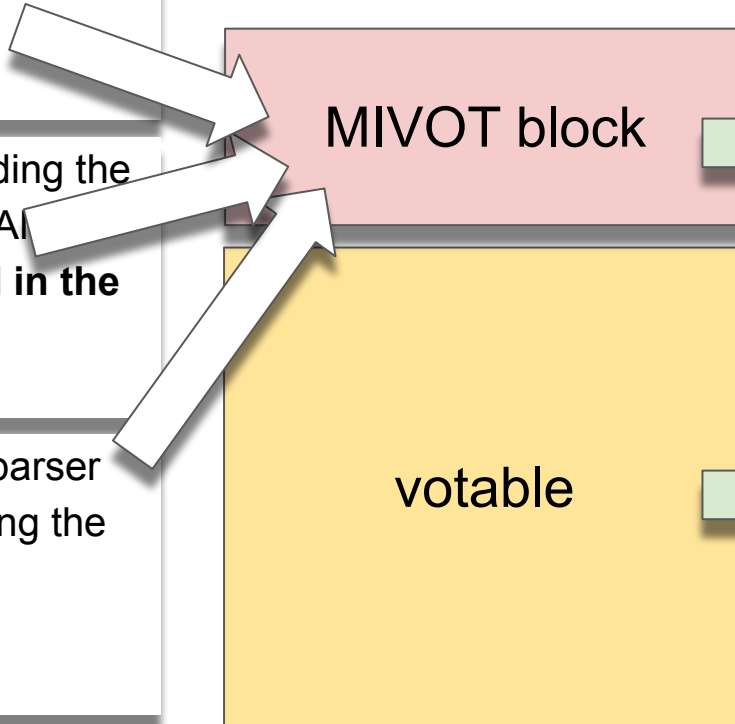
- LANG=ADQL
- REQUEST=doQuery
- **FORMAT=application/mango**
- QUERY=select top 1 \* from **mergedentry**

## SERVER SIDE: XTapDB

Resource that **associates table columns with the attributes** of the MANGO properties to which they can be mapped.

Module capable of providing the **XML snippets** for the MANGO properties to be **stacked in the query response**

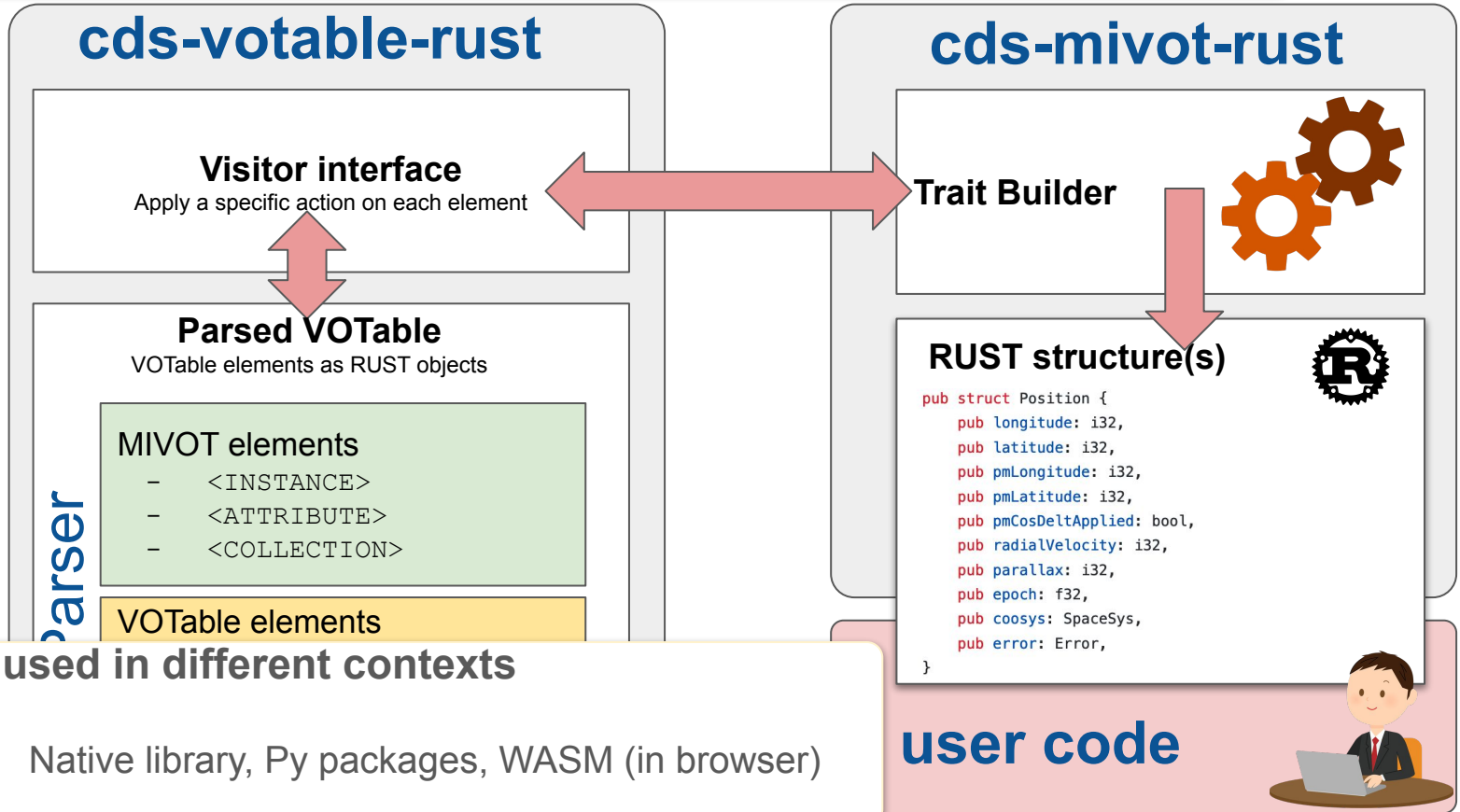
Connector to the ADQL parser that is capable of detecting the table **columns actually queried**.



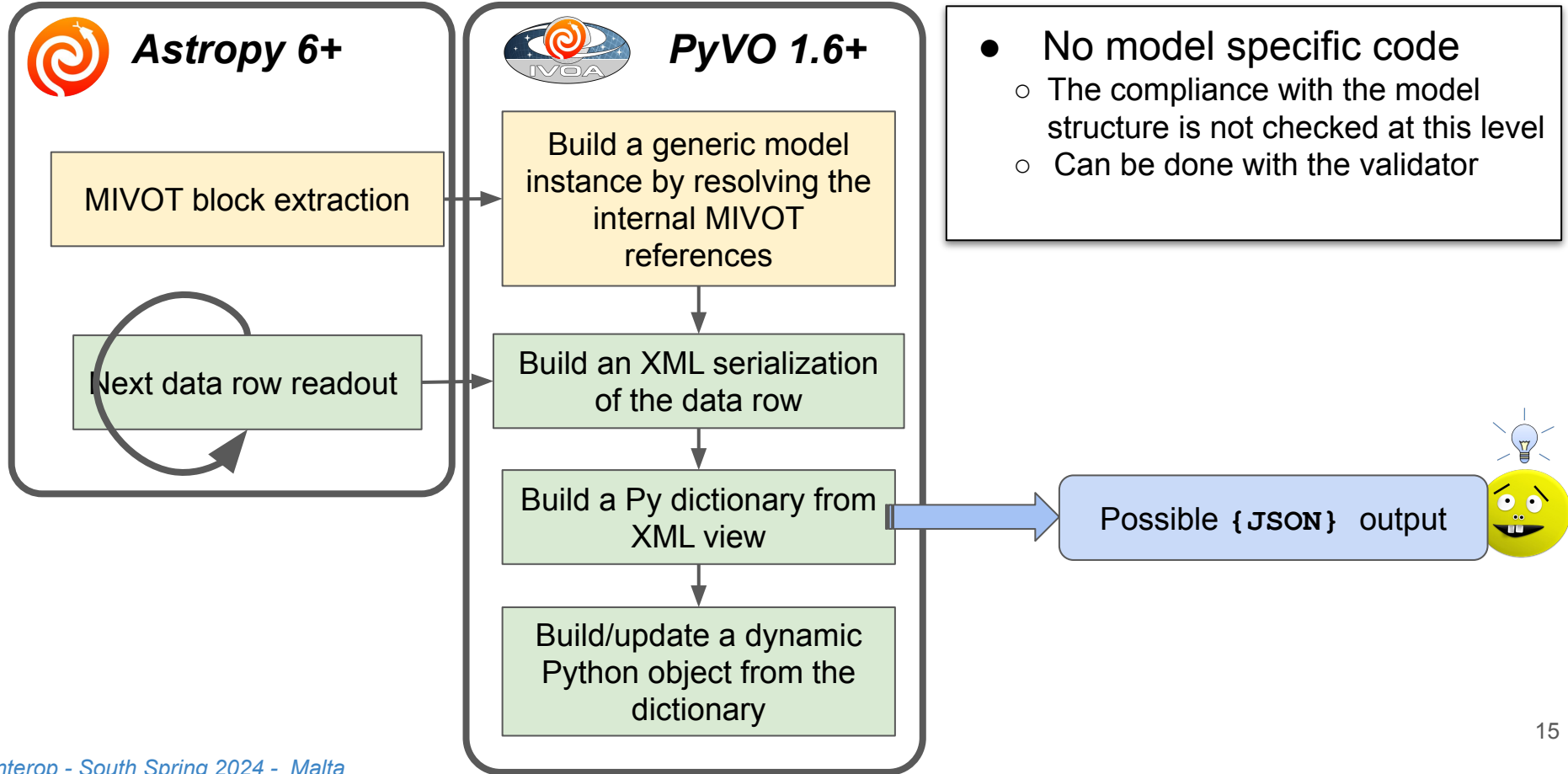
*Interoperability:*  
Readout per  
standard quantity

Readout per  
columns

# CLIENT SIDE: Rust implementation



# MIVOT: PyVO implementation,



# CLIENT SIDE: Astropy/Pyvo implementation

```
vizier_url = "https://cdsarc.cds.unistra.fr/beta/viz-bin/mivotconesearch/I/239/hip_main"
scs_srv = SCSService(vizier_url)

# init the viewer from a DAL response.
mivot_viewer = MivotViewer(
    scs_srv.search(
        pos=SkyCoord(ra=52.26708 * u.degree, dec=59.94027 * u.degree, frame="icrs"),
        radius=0.05,
    )
)

# get the reference on the MIVOT instance
mivot_object = mivot_viewer.dm_instance
while mivot_viewer.next():
    # get the space frame reference (could be done out of the loop)
    frame = mivot_object.Coordinate_coordSys.
    # get the position values and units
    ra = mivot_object.longitude.value
    ra_unit = mivot_object.longitude.unit
    dec = mivot_object.latitude.value
    # get the proper motion values and units
    pmra = mivot_object.pmLongitude.value
    pmra_unit = mivot_object.pmLongitude.unit
    pmdec = mivot_object.pmLatitude.value
    # get the epoch
    epoch = mivot_object.epoch.value
    # Print out a summary of the row
    print(f"Year {epoch}: position({frame})=[{ra} {dec} {ra_unit}] proper motion = [{pmra} {pmdec} {pmra_unit}]")

Year 1991.25: position(ICRS)=[52.26722684 59.94033461 deg] proper motion = [-0.82 -1.85 mas/yr]
```

**MIVOT instance set from SCS output**

**This implementation is model-agnostic**  
Same code for whatever model  
Model knowledge is the charge of the user



## CLIENT SIDE: Astropy/Pyvo implementation

```
from pyvo.mivot import MivotViewer

m_viewer = MivotViewer(path_to_votable)
mivot_instance = m_viewer.dm_instance
print(mivot_instance.get_SkyCoord())
<SkyCoord (ICRS): (ra, dec) in deg(52.26722684, 59.94033461)
      (pm_ra_cosdec, pm_dec) in mas / yr(-0.82, -1.85)>
```

### **Get a Skycoord on the fly in one line of Python**

The SkyCoord instance is built from all EpochPosition available parameters

# VALIDATOR

## 2 Stages validation

- Against VOTable and MIVOT XML schemas
- Against the model structure

## Class snippet generator

- XML Snippets for all classes of a model
- XML snippet for one class with constraints on the concrete classes to be instanced

## Available on Pipy

- `% pip install mivot-validator`

# VALIDATOR

🏠 mivot-validator latest

Search docs

CONTENTS:

Welcome to mivot-validator's documentation!

- Indices and tables
- mivot\_validator package
- mivot\_validator.instance\_checking package
- mivot\_validator.instance\_checking.builder package
- mivot\_validator.instance\_checking.xml\_inter package
- mivot\_validator.launchers package
- mivot\_validator.utils package
- mivot\_validator

Hot off the press! The 2024 Tidelift state of the OSS maintainer report.

Read now!

Ad by EthicalAds

🏠 / Welcome to Mivot Validator Tools [Edit on GitHub](#)

## Welcome to Mivot Validator Tools

This package has 2 purposes:

- Validation of VOTables with **MIVOT** annotations
- MIVOT serialization of model components (snippets) that can be used to build annotations

### Python scripts for validating annotated VOTables

Search projects

## mivot-validator 0.2

pip install mivot-validator

Released: Oct 3, 2023

Tools for validating VOTable annotated with MIVOT and to generate MIVOT snippets

<https://github.com/ivoa/mivot-validator>

<https://mivot-validator.readthedocs.io/en/latest/>

```
USAGE: mivot-mapping-validate [path]
Validate XML files against MIVOT schema
path: either a simple file or a directory
      all directory XML files are validated
exit status: 0 in case of success, 1 otherwise
```

🏠 Homepage

There are 2 validation levels:

- against the XML schemas (VOTable and MIVOT)
- against the model itself as it is defined in VODML

Meta

- License: MIT License (MIT License)
- Copyright (c) 2022 International

## Web Application

- Feature by feature documentation

### MIVOT-Syntax Use Cases and Examples

#### ATTRIBUTE Mapping Example in MIVOT

The ATTRIBUTE element defines either a class attribute or a collection item, both set with atomic va...

Go to page

DALI xHtml

#### COLLECTION Mapping Example in MIVOT

COLLECTION is a container element to be used each time we have to gather items of the same kind. It ...

Go to page

DALI xHtml

#### PRIMARY\_KEY Mapping Example in MIVOT

The PRIMARY\_KEY element allows to set an identification key to an INSTANCE. The primary keys are on...

Go to page

DALI xHtml

#### INSTANCE Mapping Example in MIVOT

This is a mapping example in MIVOT for the INSTANCE field. Its purpose is to defines a complex Objec...

Go to page

DALI xHtml

#### MODEL Mapping Example in MIVOT

A VOTable can provide serializations for an arbitrary number of data models. In order to declare whi...

Go to page

DALI xHtml

#### JOIN Mapping Example in MIVOT

The JOIN element populates a COLLECTION with INSTANCE elements from another collection, called herea...

Go to page

DALI xHtml

#### MIVOT-Syntax: ATTRIBUTE Mapping Example in MIVOT

The ATTRIBUTE element defines either a class attribute or a collection item, both set with atomic values. The requirements on the content (especially @ref and @drole, may differ depending on the usage.

##### Usage

1. Child of INSTANCE  
The ATTRIBUTE can be seen as a class attribute; it must have a @drole XML attribute.  
In this case, the ATTRIBUTE must be specified by:
  - \* @ref - reference to a VOTable PARAM or FIELD
  - \* @value - a literal
  - \* if both are provided; @value serves as the default if the reference cannot be resolved
  - \* if @ref cannot resolved and there is no @value, the client must consider that value as NULL in its software context (Language, inner model).
2. Child of COLLECTION  
In this case the host COLLECTION can be seen as a vector and the ATTRIBUTE as one coordinate of the vector. It must have no @drole XML attribute or an empty one.  
In this case, the ATTRIBUTE must be specified by:
  - \* @ref - reference to a VOTable PARAM or FIELD
  - \* @value - a literal
  - \* if both are provided, @value serves as the default if the reference cannot be resolved
  - \* if @ref cannot resolved and there is no @value, the client must consider that value as NULL.The definition of the NULL value depends of the software environment
3. Any case :  
The ATTRIBUTE must always have a non-empty @dtype XML attribute. This @dtype may complete missing types in the VOTable. In case of inconsistency with VOTable @dtype SHOULD supersede VOTable types.

<https://saada.unistra.fr/voexamples/show/MIVOT-Syntax>

Go to page

DALI xHtml

Go to page

DALI xHtml

will be set with the value of the Idobstime column:

## A BIT MORE ...

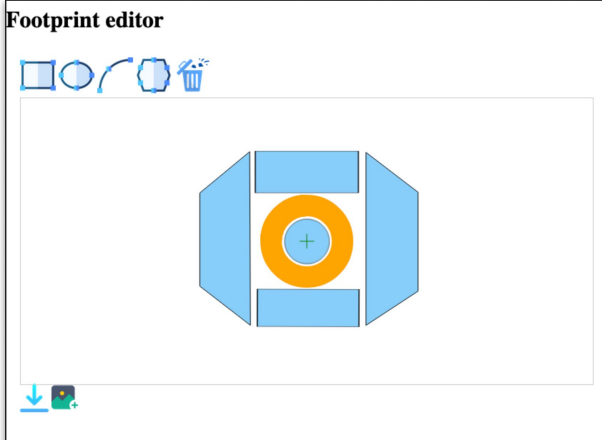
The FPS has been enable to provide MIVOT serializations of any filter

<http://svo2.cab.inta-csic.es/theory/fps/fpsmap.php?ID=SLOAN>

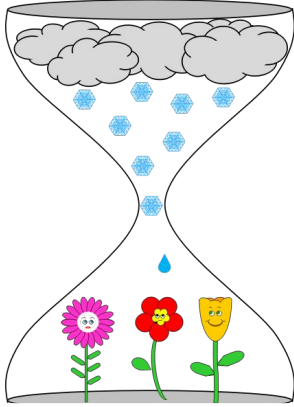
```
<!-- XML serialization of Photometry Filter -->
<dm-mapping:VODML>
  <dm-mapping:GLOBALS>
    <!-- Photometric System -->
    <dm-mapping:INSTANCE dmtypes="photdm:PhotometricSystem" dmid="_sys_">
      <dm-mapping:ATTRIBUTE dmrole="photdm:PhotometricSystem.description" dmtypes="ivoa:string" value=""/>
      <dm-mapping:ATTRIBUTE dmrole="photdm:PhotometricSystem.detectorType" dmtypes="ivoa:integer" value=""/>
      <dm-mapping:COLLECTION dmrole="photdm:PhotCal.photometryFilter">
        <dm-mapping:REFERENCE dmref="_fil_">
          <dm-mapping:COLLECTION>
            <dm-mapping:INSTANCE>
              <!-- Photometric Calibration -->
              <dm-mapping:INSTANCE dmtypes="photdm:PhotCal" dmid="_cal_">
                <dm-mapping:ATTRIBUTE dmrole="photdm:PhotCal.identifier" dmtypes="ivoa:string" value=""/>
                <!-- Magnitude System -->
                <dm-mapping:INSTANCE dmtypes="photdm:MagnitudeSystem" dmrole="photdm:PhotCal.magnitudeSystem">
                  <dm-mapping:ATTRIBUTE dmrole="photdm:MagnitudeSystem.type" dmtypes="photdm:TypeOMagSystem" value=""/>
                <dm-mapping:INSTANCE>
                  <!-- Zero Point -->
                  <!-- Filter -->
                  <dm-mapping:REFERENCE dmref="_fil_" dmrole="photdm:PhotCal.photometryFilter">
                    <dm-mapping:INSTANCE>
                      <!-- Filter -->
                      <dm-mapping:INSTANCE dmtypes="photdm:PhotometryFilter" dmid="_fil_">
                        <dm-mapping:ATTRIBUTE dmrole="photdm:PhotometryFilter.identifier" dmtypes="ivoa:string" value=""/>
                        <dm-mapping:ATTRIBUTE dmrole="photdm:PhotometryFilter.name" dmtypes="ivoa:string" value=""/>
                      </dm-mapping:INSTANCE>
                    </dm-mapping:REFERENCE>
                  </dm-mapping:INSTANCE>
                </dm-mapping:INSTANCE>
              </dm-mapping:INSTANCE>
            </dm-mapping:COLLECTION>
          </dm-mapping:REFERENCE>
        </dm-mapping:COLLECTION>
      </dm-mapping:INSTANCE>
    </dm-mapping:COLLECTION>
  </dm-mapping:GLOBALS>
</dm-mapping:VODML>
```

The combination of the FoV-DM (draft) with MIVOT allows to carry the description of instrument footprints

<https://saada.unistra.fr/fov/>



# STATUS AND PROSPECTS



**Waiting MANGO is stable**

**Hope to have new implementations**

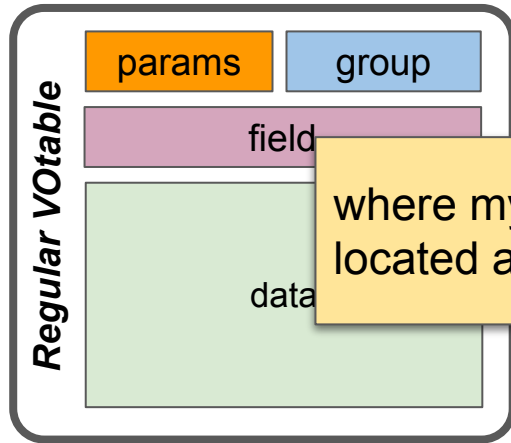
- services
- client codes

**Extend the PyVO implementation**

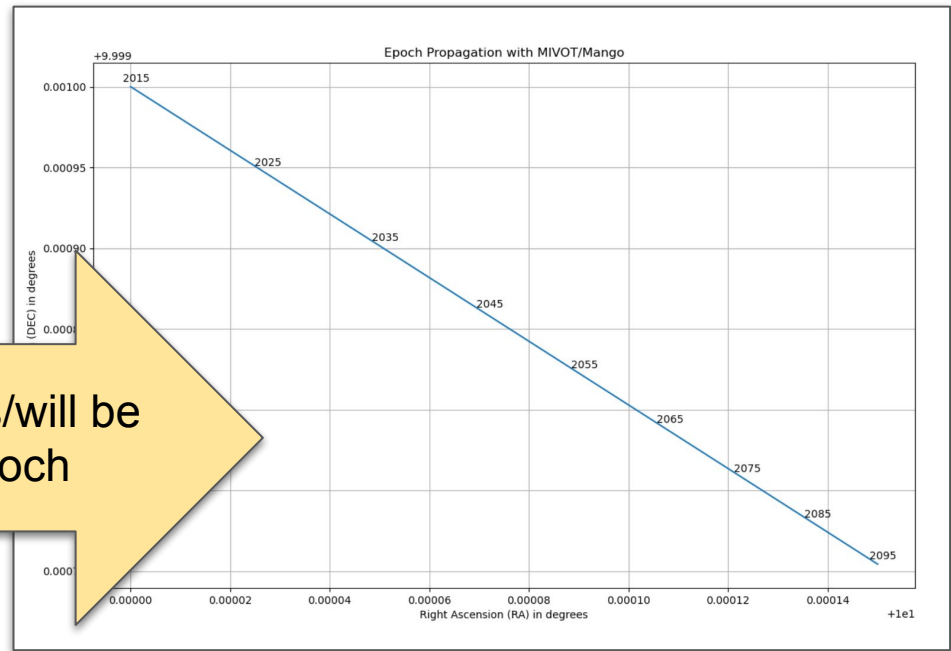
- photometry
- errors

**backup**

# The Challenge



where my object was/will be located at a given epoch



**In this VOTABLE, I have somewhere:**

- Position
- Proper motion
- Parallax
- Radial velocity
- Desired metadata

*The challenge:* **Agree** on an appropriate way to present data to make the process standard



# Legacy: Position + Proper Motion

```
<COOSYS ID="J2000" equinox="J2000" epoch="J2000" system="eq_J2000"/>
<FIELD name="pos_RA" ucd="pos.eq.ra;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pos_DEC" ucd="pos.eq.dec;meta.main" datatype="double" unit="deg" ref="J2000"/>
<FIELD name="pm_RA" ucd="pos.pm.ra;meta.main" datatype="double" unit="mas/y" ref="J2000"/>
<FIELD name="pm_DEC" ucd="pos.pm.dec;meta.main" datatype="double" unit="mas/year" ref="J2000"/>
```



- Can see with UCDs that **pos\_RA** and **pos\_DEC** do work together
- Can see with UCDs that **pm\_RA** and **pm\_DEC** do work together
- The 4 columns refer to the **COOSYS** element



- The role of the **@ref->@ID** link is implicit (**@ref** to what?)
- No clear way to see that **pos\_RA/pos\_DEC** and **pm\_RA/pm\_DEC** relate to the same quantity

# Legacy: Limitation of the COOSYS solutions



- The `@ref` semantic or **role is not well defined**
- The way fields connected to `coosys` **do interact** together is **poorly defined**
- **No** convenient way to show up **errors with correlations**
- According the the VOTable document, solution based on `coosys` are **short term solution**

Note that the `COOSYS` may be deprecated in the future in favor of a more generic way of describing the conventions used to define the positions of the objects studied in the enclosed tables.

- We **miss a model** showing how all of these **parameters do work together**.

# MIVOT: The Model Based Alternative

- 0 Keep the VOTable as delivered by the DAL engine**
  - No FIELDREF tuning
  - No GROUP to create
  
- 1 Use a VO model describing the EPOCH propagation**
  - 6 parameters
  - Errors
  - Space Frame
  - Time System
  
- 2 Write an XML serialization of that MODEL in the VOTable**
  - MIVOT syntax
  
- 3 Insert that XML piece above the TABLE**
  - Put the reference of the matching FIELDS into the model leaves

# Run the Notebook

```
% git clone git@github.com:astropy/pyvo.git
```

```
% git clone git@github.com:ivoa/dm-usecases.git
```

```
% cd dm-usecases/notebooks
```

```
% jupyter notebook
```

# MIVOT: A VO Standard

International Virtual Observatory Alliance

IVOA Documents



## Model Instances in VOTables Version 1.0

IVOA Recommendation 20 June 2023

### Interest/Working Group:

<http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel>

### Author(s):

Laurent Michel, Mark Cresitello-Dittmar, François Bonnarel, Gilles La

### Editor(s):

Laurent Michel, Mark Cresitello-Dittmar

## Abstract

Abstract: Model Instances in VOTables (MIVOT) defines a syntax to map VOTable d VOTable to the data model elements (class, attributes, types, etc.) of a standardiz missing in the table metadata. The data model elements are grouped in an indepe The MIVOT syntax allows to describe a data structure as a hierarchy of classes. It VOTable. Missing metadata can also be provided using MIVOT, for instance by con both client and server sides. The adopted design does not alter the original VOTab

## Status of this document

This document has been produced by the Data Model Working Group. It has been reviewed by IVOA Members and other interested parties, and has been reference from another document. IVOA's role in making the Recommendation is t Community.

Available formats: [pdf](#)

The standard comes with:

- An IVOA standard document
- An XML schema allowing computer to validate documents.
  - The schema has been written in XSD1.1 to support different syntax patterns depending on the local context
  - The schema is independent from the VOTable: tools not supporting MIVOT are still working on annotated VOTables

# 4 Aladin Desktop Prototype (P. Fernique)

The screenshot displays the Aladin v12.1 desktop interface. At the top, it reads "Aladin v12.1 \*\*\* BETA VERSION (based on v12.107) \*\*\*". The main window shows a star field with a central bright star. A yellow arrow points from a text box "Many stars do move" to a cluster of stars. Another yellow arrow points from a text box "If you move that cursor" to a cursor in the bottom right corner of the star field. A third yellow arrow points from a text box "Believe me!" to the same cursor. The interface includes a left sidebar with a tree view of data collections, a top menu bar with "Command", "Frame ICRS", and "Projection Aitoff", and a right sidebar with a "Welcome to Aladin" message and a list of features. At the bottom, there are various tool icons and a status bar showing coordinates and a scale of 1'.

Many stars do move

If you move that cursor

Believe me!

## 4 RUST implementation (F.X Pineau J. Abid)

```
use std::path::Path;
use crate::mivot::ModelLayer;

let mut model_layer = ModelLayer::from_file(Path::new("my-votable.xml"), true).unwrap();

model_layer.init_epoch_positions();

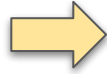
// Retrieve the EpochPosition instance from the mapping block
let epoch_positions = model_layer.get_epoch_positions_as_ref().get("EpochPosition").unwrap();

// Browse the instance
print!("{}", epoch_positions.longitude.value);
print!("{}", epoch_positions.latitude.value);
print!("{}", epoch_positions.coosys.frame.spaceRefFrame);
```

**This implementation is model-dependant**  
New/change model => New code

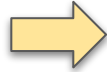
# MIVOT: Flexibility

EPOCH defined in a **<FIELD>**  
@ref to the FIELD identifier



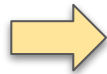
```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           ref="_EPOCH_FIELD"/>
```

EPOCH defined in a **<PARAM>**  
@ref to the PARAM identifier



```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           ref="_EPOCH_PARAM"/>
```

EPOCH defined in a non  
machine-readable element  
No @ref but a fixed @value



```
<ATTRIBUTE dmrole="tucson:Position.epoch"  
           dmtype="coords:Epoch"  
           unit="year"  
           value="J2023.88"/>
```

The structure of MIVOT block is not altered by the way the EPOCH is set in the VOTable

- Allow server code to be versatile
- Allow a same client code to process many different VOTable



# Conclusions

## **MIVOT+Mango: a seamless solution for the Epoch propagation**

- Model supporting the complex errors
- Astropy/PyVO API
- Server side implementation preserving the original VOTable
- No change in the VOTable schema

## **The same mechanism can be used for many others quantities**

- Versatile mapping syntax
- Photometric data
- Dataset meta data
- ...

"eq\_FK4"), and `epoch` specifies the epoch of the positions if necessary. Note that the `COOSYS` may be deprecated in the future in favor of a more generic way of describing the conventions used to define the positions of the objects studied in the enclosed tables.

# CLIENT SIDE: Astropy/Pyvo implementation

