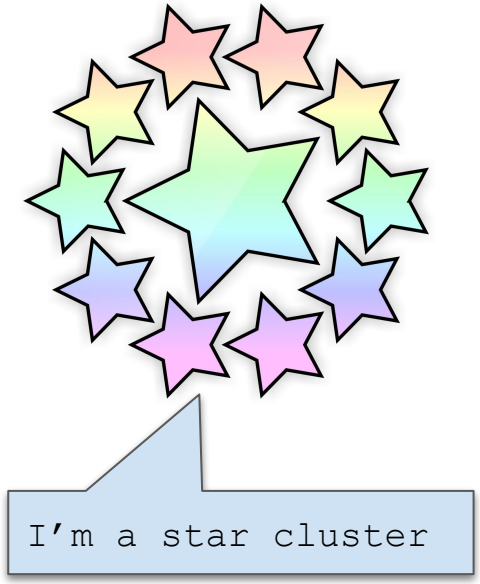
TAP and the Data Models

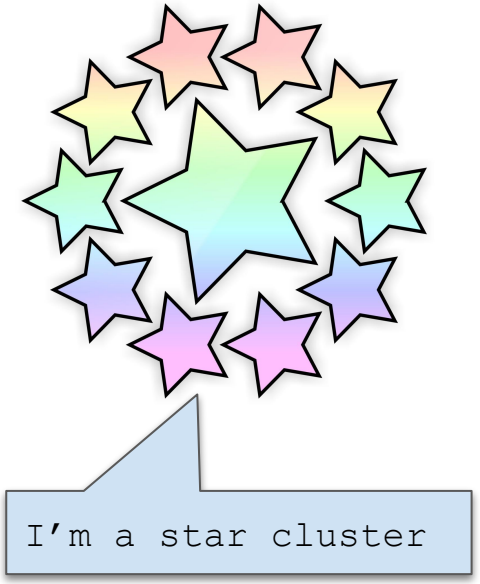
ADASS XXXI

28/09/2021

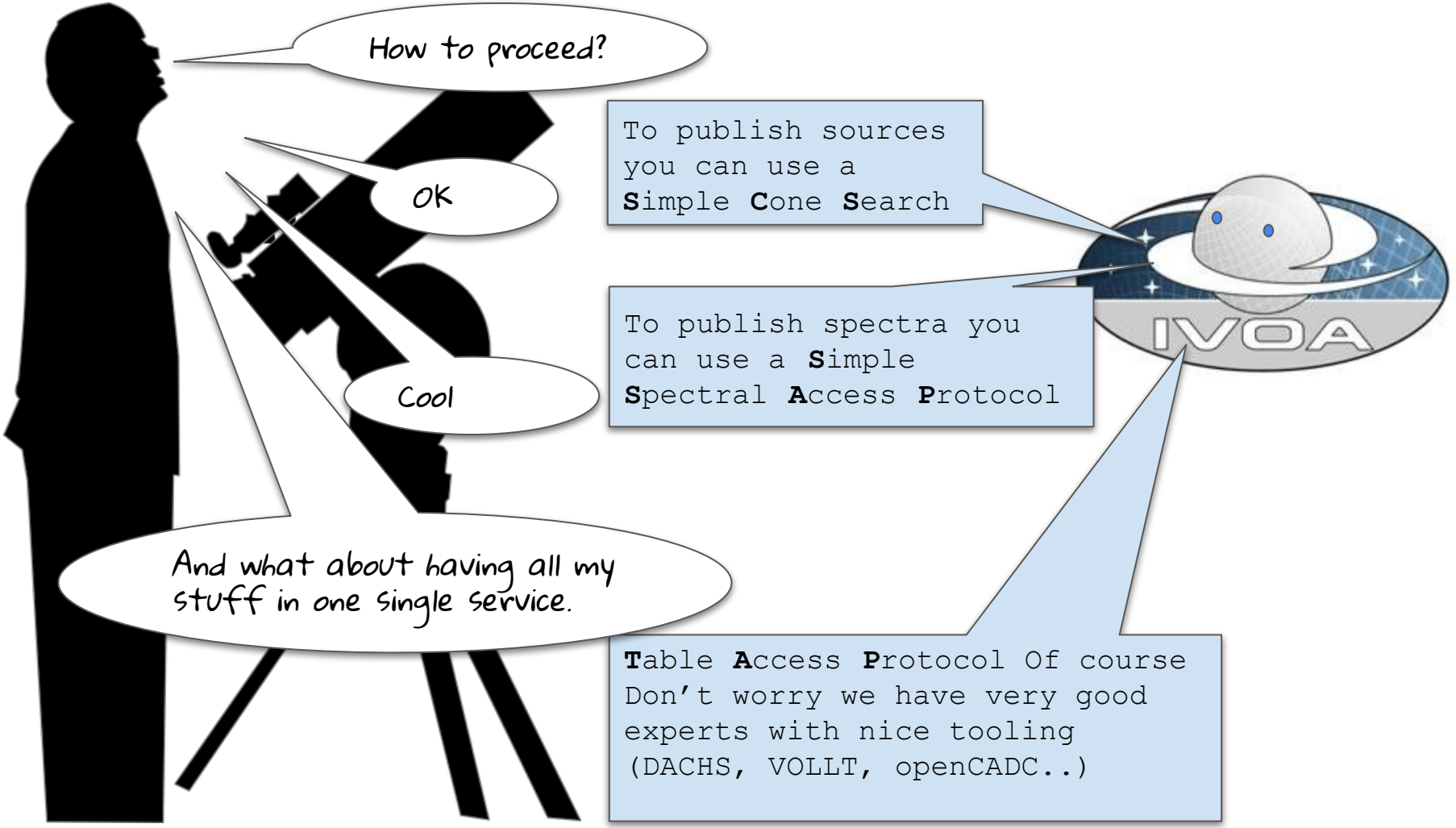
Laurent Michel - François Bonnarel - Mireille Louys - Dave Morris


I’m a star cluster


I'm an astronomer


I’m a star cluster


I'm an astronomer
Night after night, I study this cluster
I have observed each of its stars many times
I acquired many spectra during these observations
I want' to publish this data in the VO
I'm a star cluster


How to proceed?
OK
Cool
And what about having all my stuff in one single service.
To publish sources you can use a **S**imple **C**one **S**earch
To publish spectra you can use a **S**imple **S**pectral **A**ccess **P**rotocol
IVOA
Table **A**ccess **P**rotocol Of course Don't worry we have very good experts with nice tooling (DACHS, VOLLT, openCADC..)

TAP Basics

- **Make rational DBs interoperable**
  - TAP does not care of the way data is stored:
    - Put it in relational tables and that’s it
  - TAP just makes it discoverable and searchable
    - Interoperable in others words

- **Built on 3 pillars**
  - **Access: UWS**
    - UWS: REST API, sync or async
  - **Query language: ADQL**
    - ADQL : derived from an SQL subset with some astro-specific functions
  - **Query result format: VOTable**
    - at least

TAP_SCHEMA

Schemas DB desc
- Name

Tables DB desc
- Name
- Location

Columns DB desc
- Name
- Location
- type

Joins DB desc
- Source
- target
- keys

Data Tables

sources

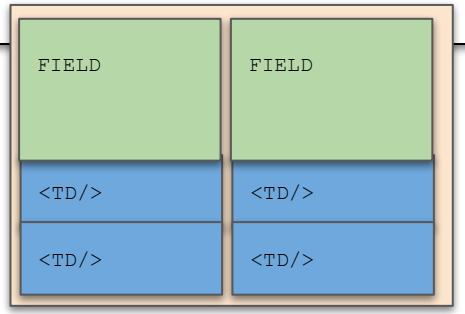
detection

spectra

Interoperability Keys:

- The TAP_SCHEMA provides a standard description of the database content
- Client can explore the TAP_SCHEMA with standard ADQL queries to discover the DB content and the way to retrieve data

```
SELECT * FROM sources WHERE sources.magV > 19
```



TAP_SCHEMA

Schemas DB desc - Name	**VO Meta-data** - Description - UType
Tables DB desc - Name - Location	**VO Meta-data** - Description - UType
Columns DB desc - Name - Location - type	**VO Meta-data** - UCD / UType - XType - Unit
Joins DB desc - Source - target - keys	**VO Meta-data** - Description - UType

Data Tables

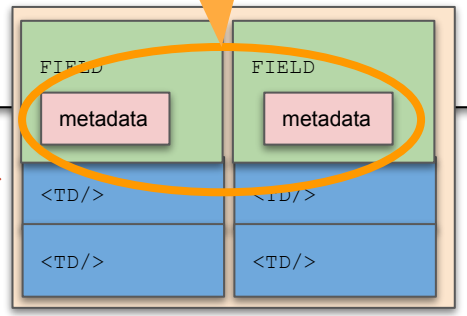
- sources
- detection
- spectra

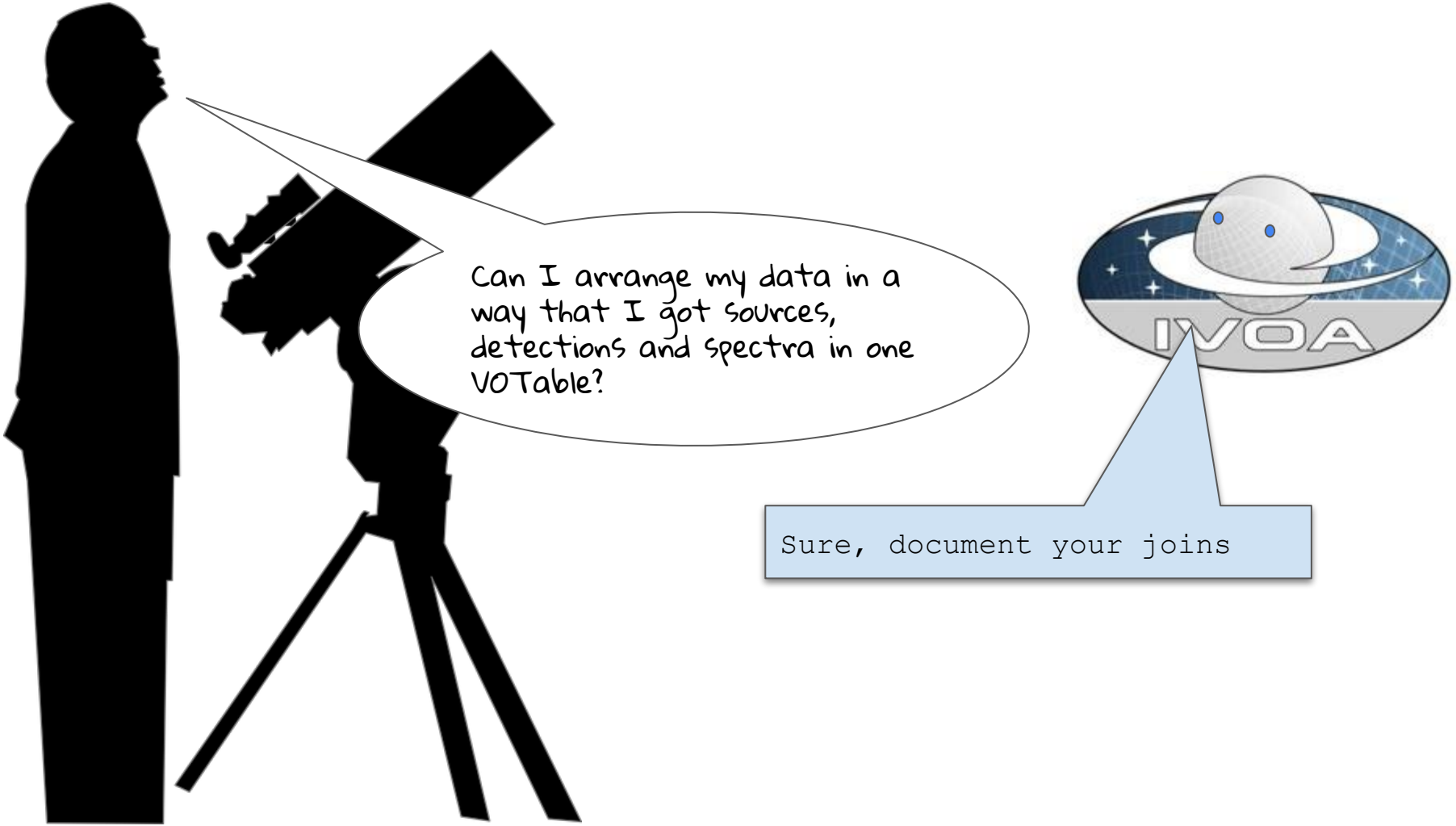
Interoperability Keys:

Having standard metadata allows clients to properly understand retrieved data without consideration on their origin (up to a certain extend)

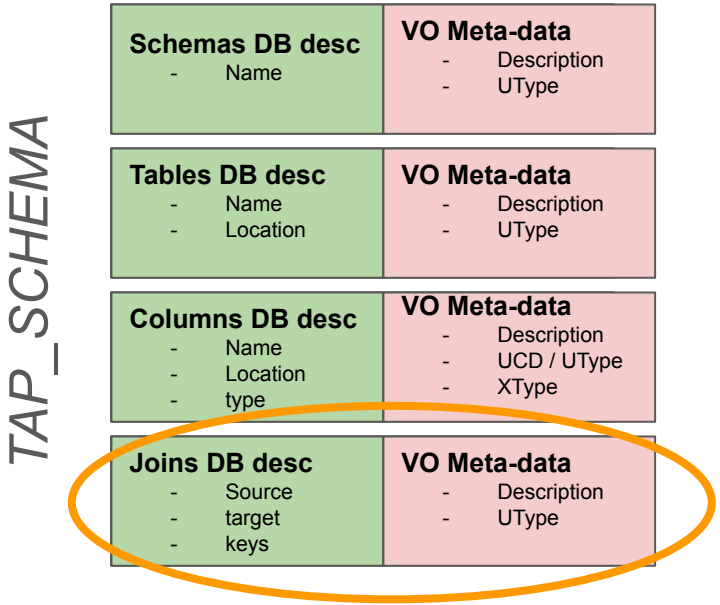
- Helps for generic client code
- Facilitate dataset comparisons

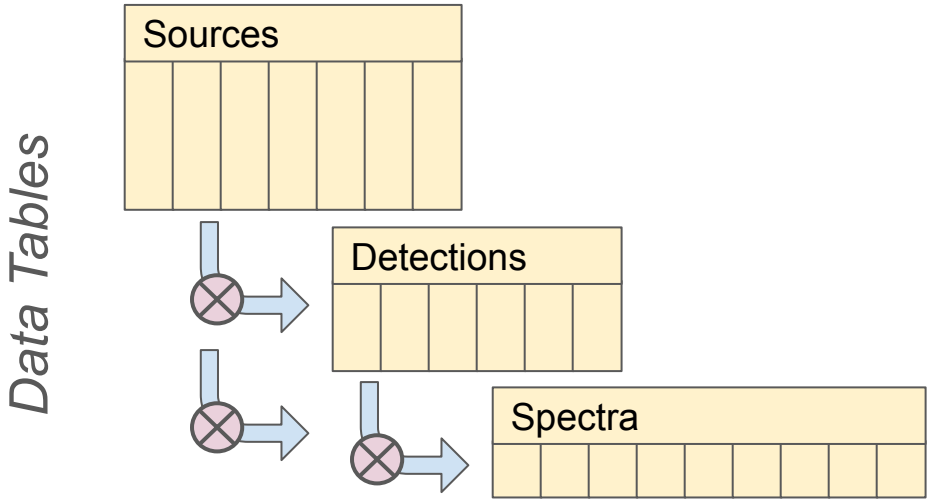
```
SELECT * FROM sources WHERE sources.magV > 19
```



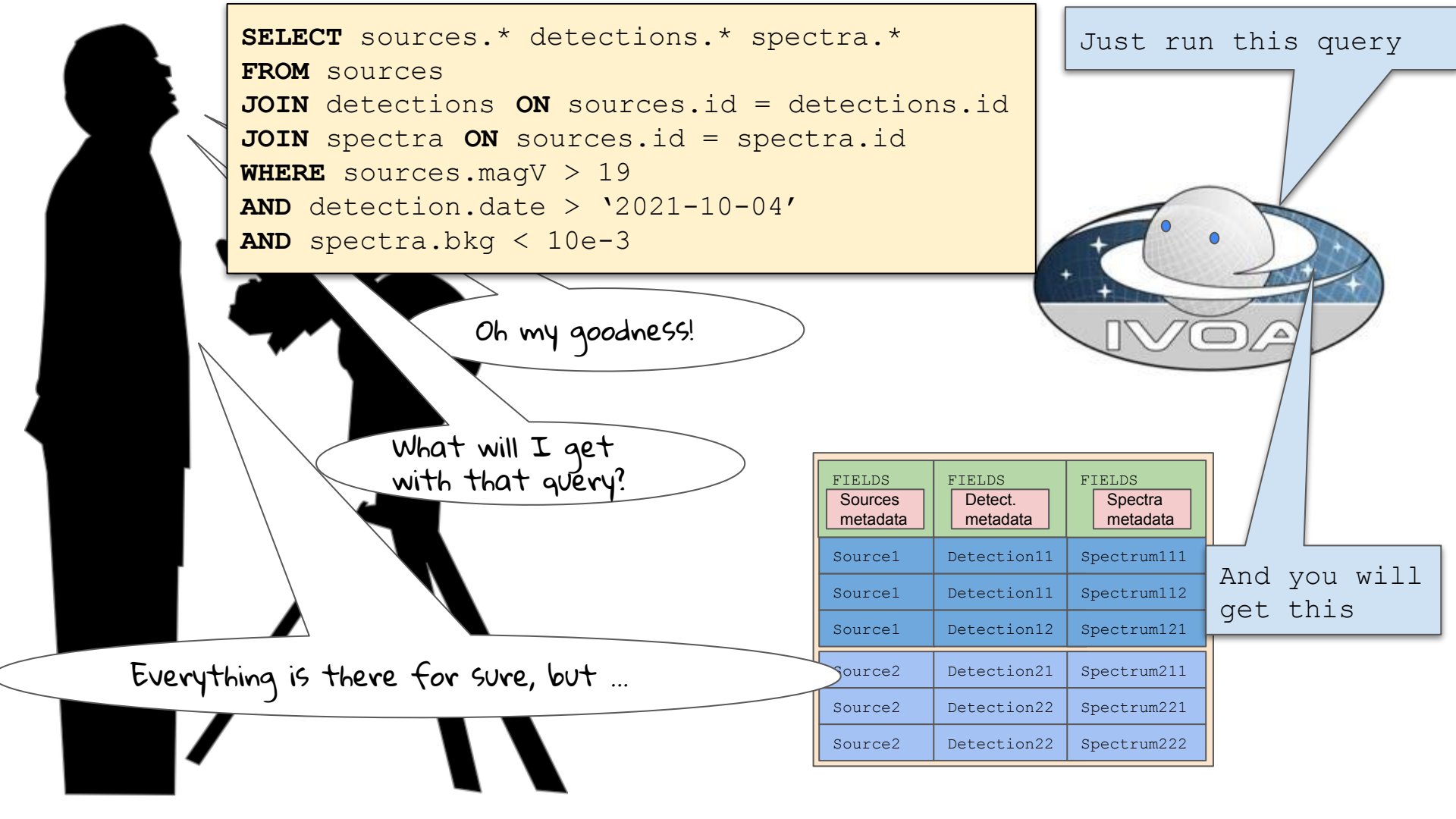

Can I arrange my data in a way that I got sources, detections and spectra in one VOTable?
IVOA
Sure, document your joins

TAP Level 2



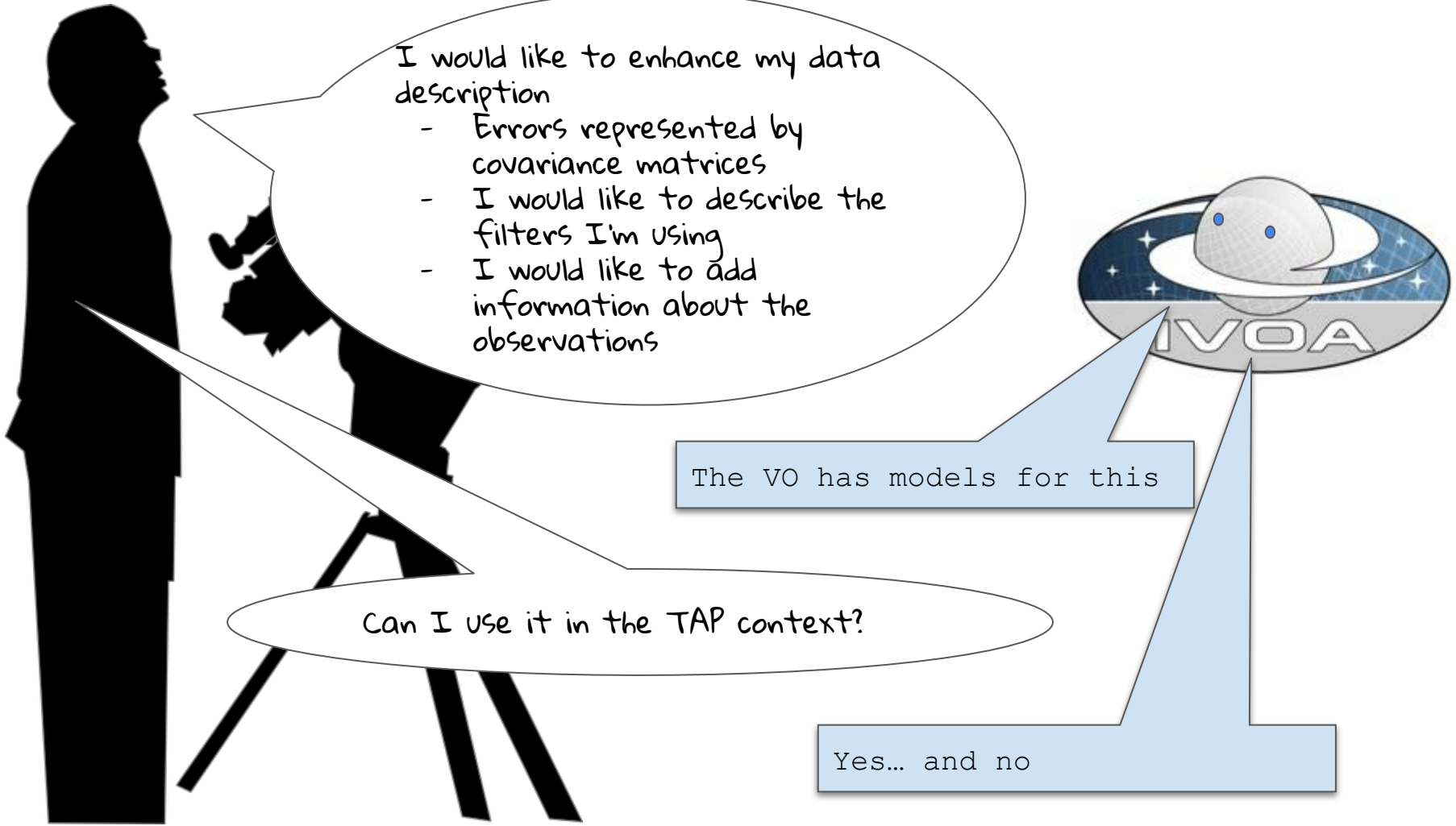


- The TAP_SCHEMA can suggest and document scientifically relevant table joins


SELECT sources.* detections.* spectra.*
FROM sources
JOIN detections ON sources.id = detections.id
JOIN spectra ON sources.id = spectra.id
WHERE sources.magV > 19
AND detection.date > '2021-10-04'
AND spectra.bkg < 10e-3
Just run this query
Oh my goodness!
What will I get with that query?
Everything is there for sure, but ...
And you will get this
FIELDS Sources metadata
FIELDS Detect. metadata
FIELDS Spectra metadata
Source1 Detection11 Spectrum111
Source1 Detection11 Spectrum112
Source1 Detection12 Spectrum121
Source2 Detection21 Spectrum211
Source2 Detection22 Spectrum221
Source2 Detection22 Spectrum222

How to Deal with Connected Data

- TAP almost support models...
  - Standard meta-data
  - 1-N relationship (table joins)
    - can be served by a Datalink service
- … But not completely
  - See after
- … Or not easily
  - Queries difficult to set-up
  - Output difficult to process
    - This can be worked around at client level
    - Tap-complex: a JavaScript *Join-Walker*


I would like to enhance my data description
- Errors represented by covariance matrices
- I would like to describe the filters I'm using
- I would like to add information about the observations
IVOA
The VO has models for this
Can I use it in the TAP context?
Yes… and no

TAP Level 2

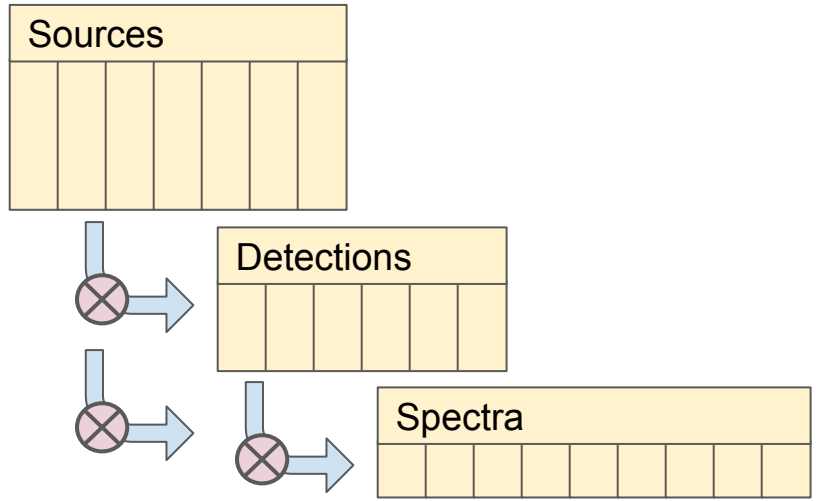
Modeles meta-data
- Complex errors
- Column grouping
- Filters
- Frames
- Autorship
- …..

This information does not come along with the data, It results from a curation work (e.g. paper reading)

TAP_SCHEMA

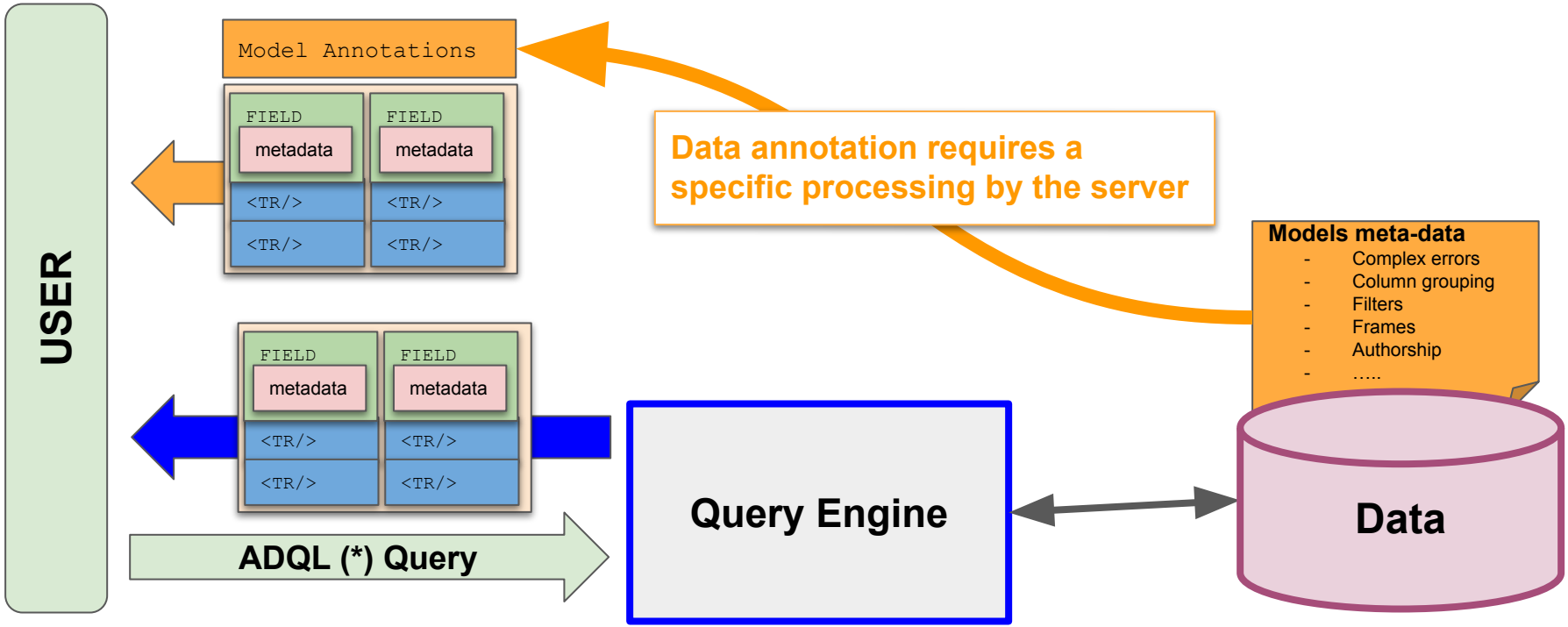
Schemas DB desc - Name	**VO Meta-data** - Description - UType
Tables DB desc - Name - Location	**VO Meta-data** - Description - UType
Columns DB desc - Name - Location - type	**VO Meta-data** - Description - UCD / UType - XType
Joins DB desc - Source - target - keys	**VO Meta-data** - Description - UType

Data Tables



- The TAP_SCHEMA cannot support some **model related meta-data**

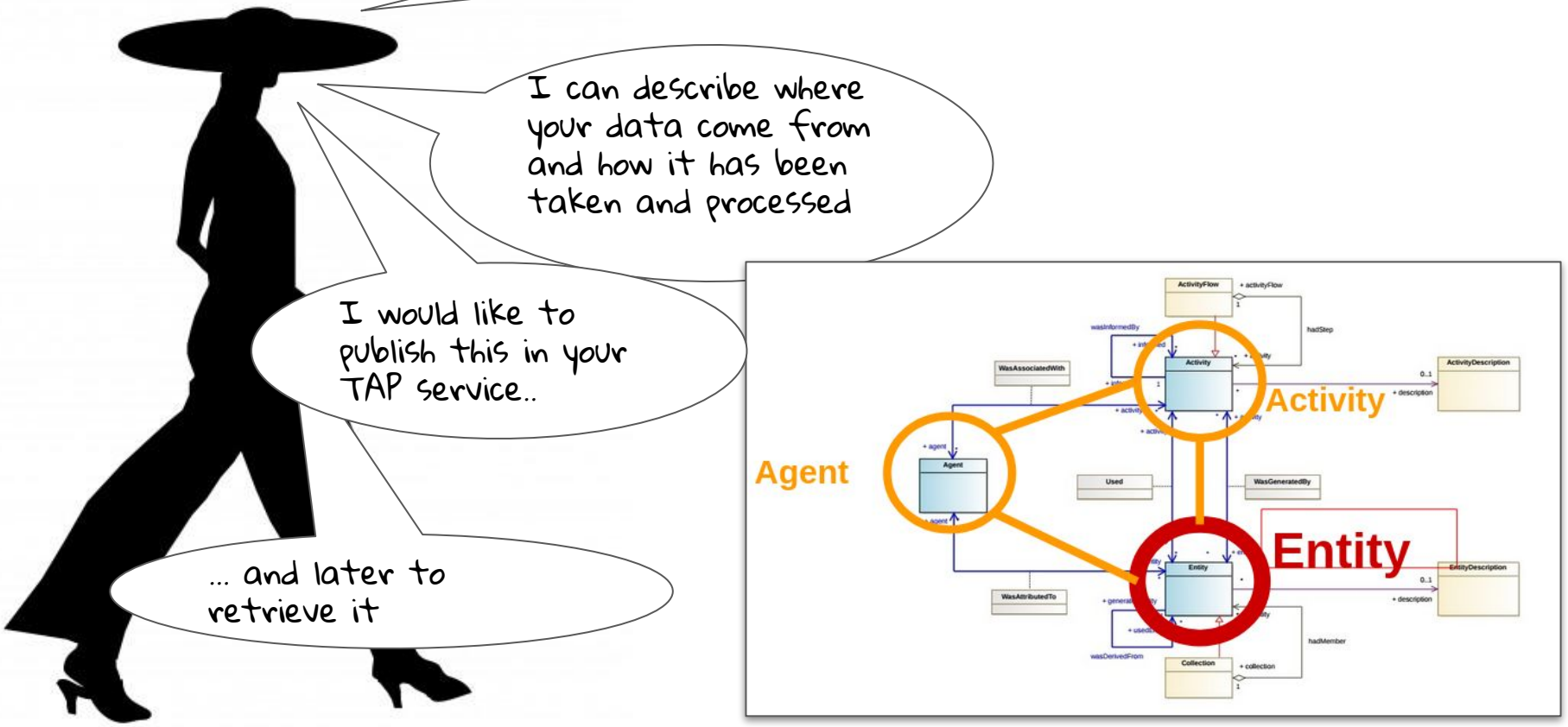
Data Annotation



(*) Must be able to notify the server whether searched data has be annotated or not

Data Annotation

- **Three Questions**
  - **CURATION: How to give the service the items it needs to annotate data**
    - Curation issues are out the scope of any standards (hand-made work so far)
  - **QUERY PROCESSING: How to enable a service to annotate data**
    - Mireille: model annotation on the fly based on VODML mapping
    - Judith: model annotation based on <GROUPS>
  - **USER REQUEST: How to tell a service to annotate searched data**
    - Part #1 of Laurent's talk (on behalf of Dave) talk coming now

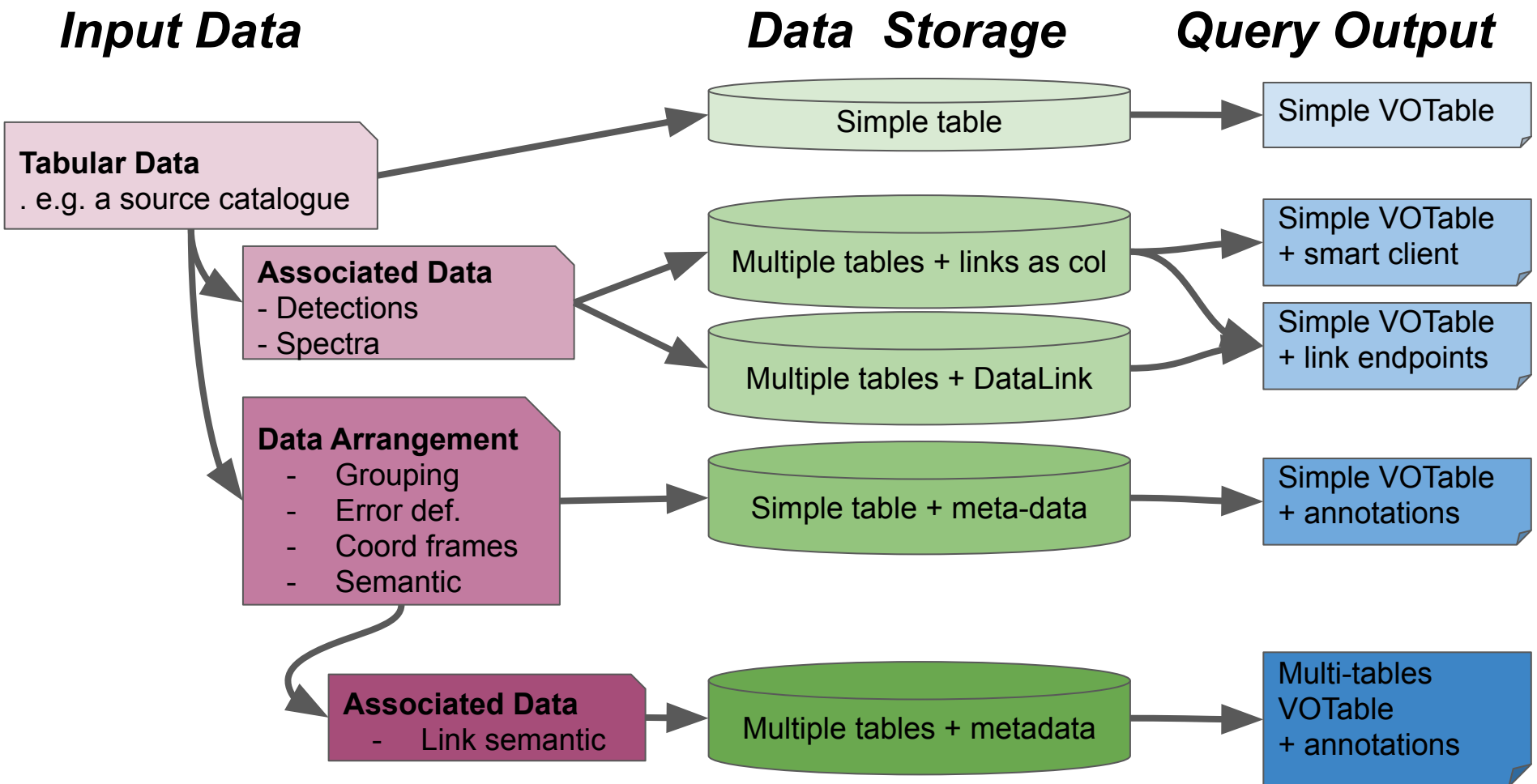

I'm Miss Provenance
I can describe where your data come from and how it has been taken and processed
I would like to publish this in your TAP service..
... and later to retrieve it
Agent
Activity
Entity
ActivityFlow
Activity
Agent
Entity
Used
WasGeneratedBy
WasAssociatedWith
WasAttributedTo
Collection
ActivityDescription
EntityDescription
wasInformedBy
hadStep
wasDerivedFrom
hadMember

How to deal with mapped data

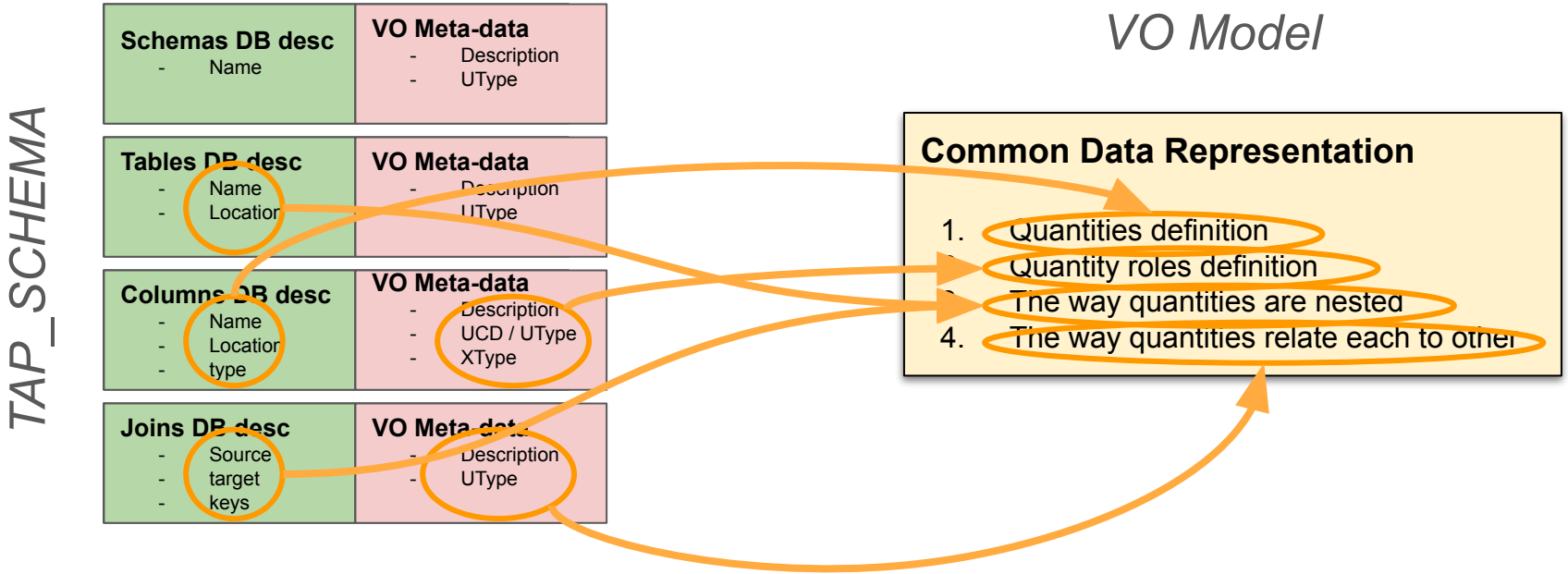
- **A case very close to an object relational mapping (ORM) pattern:**
  - We start from an existing model e.g. Provenance Data Model
  - We have data matching this model

- **Three Questions**

  - **SERVICE SETUP: How to enable a TAP service to store model instances**
    - ORM rules
    - TAP_SCHEMA setup
  - **DATA INGESTION: How to import model instances in that service**
  - **François's talk coming now**
  - **DATA RETRIEVAL: How to retrieve model instances from that service**
    - Part #2 of Laurent's talk (on behalf of Dave) talk coming now


Input Data
Data Storage
Query Output
Tabular Data
. e.g. a source catalogue
Associated Data
- Detections
- Spectra
Data Arrangement
- Grouping
- Error def.
- Coord frames
- Semantic
Associated Data
- Link semantic
Simple table
Multiple tables + links as col
Multiple tables + DataLink
Simple table + meta-data
Multiple tables + metadata
Simple VOTable
Simple VOTable
+ smart client
Simple VOTable
+ link endpoints
Simple VOTable
+ annotations
Multi-tables
VOTable
+ annotations

TAP has (almost) anything to host models instances



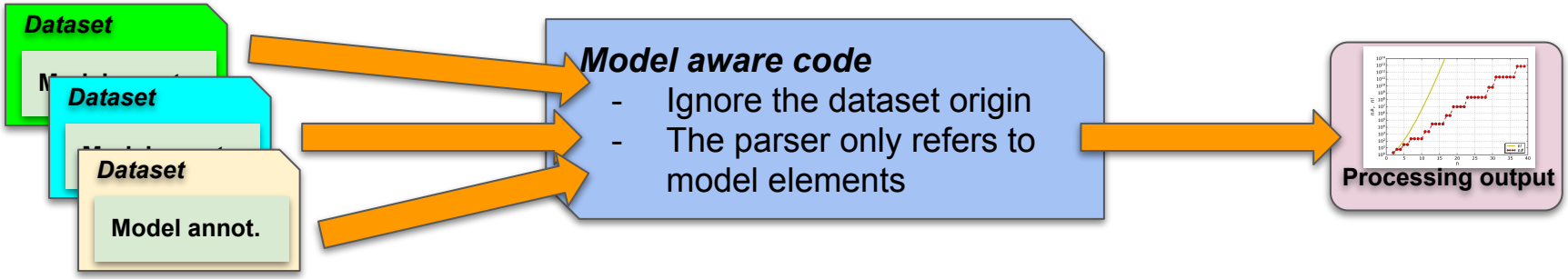
Adding VO Meta-data

- **TAP_SCHEMA Meta-data**
  - **`Description`**: Textual description
  - **`UCD`**: Standard VO vocabulary telling the quantities stored in the columns
  - **`UType`**:
    - **Columns**: Bind data columns with data model leaves
    - **Table**: Data-model mapped on the table
  - **`XType`**: Tells how to interpret data columns (e.g. string as STC polygon)
  - **`Unit`**: Unit in VO standard format

Interoperability Key:
Having standard metadata allows clients to properly understand retrieved data without consideration on their origin (up to a certain extend)

- Helps for generic client code
- Facilitate dataset comparisons

Modeled meta-data => VO Model

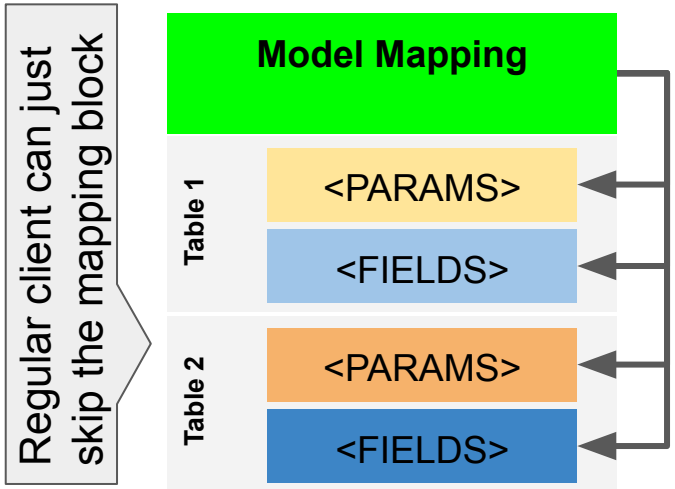


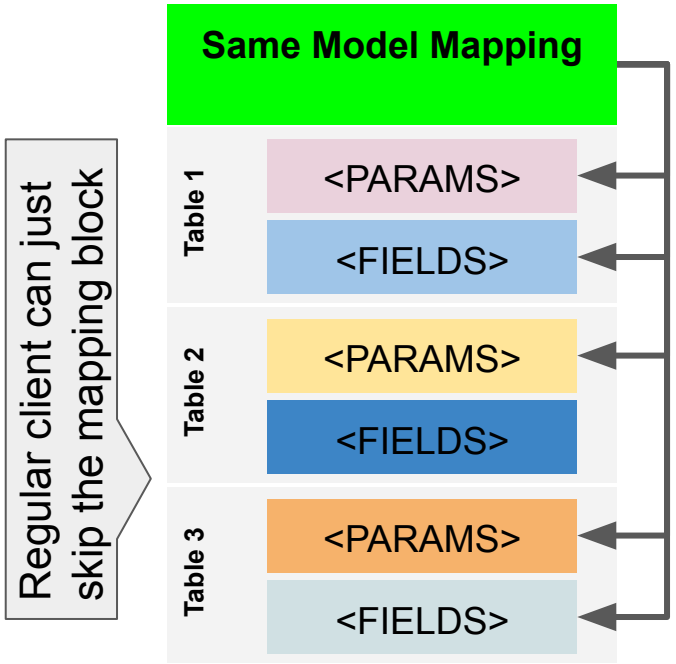
Modeling data <> Making Data models
Building data descriptions that can be shared by different stakeholders and that are independent from any particular data provider

Common Data Representation

1. Quantities definition
2. Quantity roles definition
3. The way quantities are nested
4. The way quantities relate each to other

- A model-aware client access data through the mapping block.
- The same code can understand and process an annotated VOTable whatever either its origin or the way data are arranged





TAP and the models

- **What TAP can do**
  - Bind tables with models (UTypes)
  - Bind columns with model leaves (UTypes)
  - Report this information in VOTables in a shareable way.

- **What TAP cannot do yet**
  - Store extra flat meta-data (e.g. frames, authorship…)
    - It is still possible to create specific tables for this though.
  - Store the way legacy data must be map yo a model
    - when this mapping involves more than one column
    - when this mapping involves more than one table
  - Build VOTables able to map data on a model
    - Using multiple tables
    - Set model annotations inside the VOTable
  - Tell the query engine to return such VOTables