Data Models:

Modularity, levels, endorsement; Architecture and TCG workflow

Mark Cresitello-Dittmar

2025-06-05

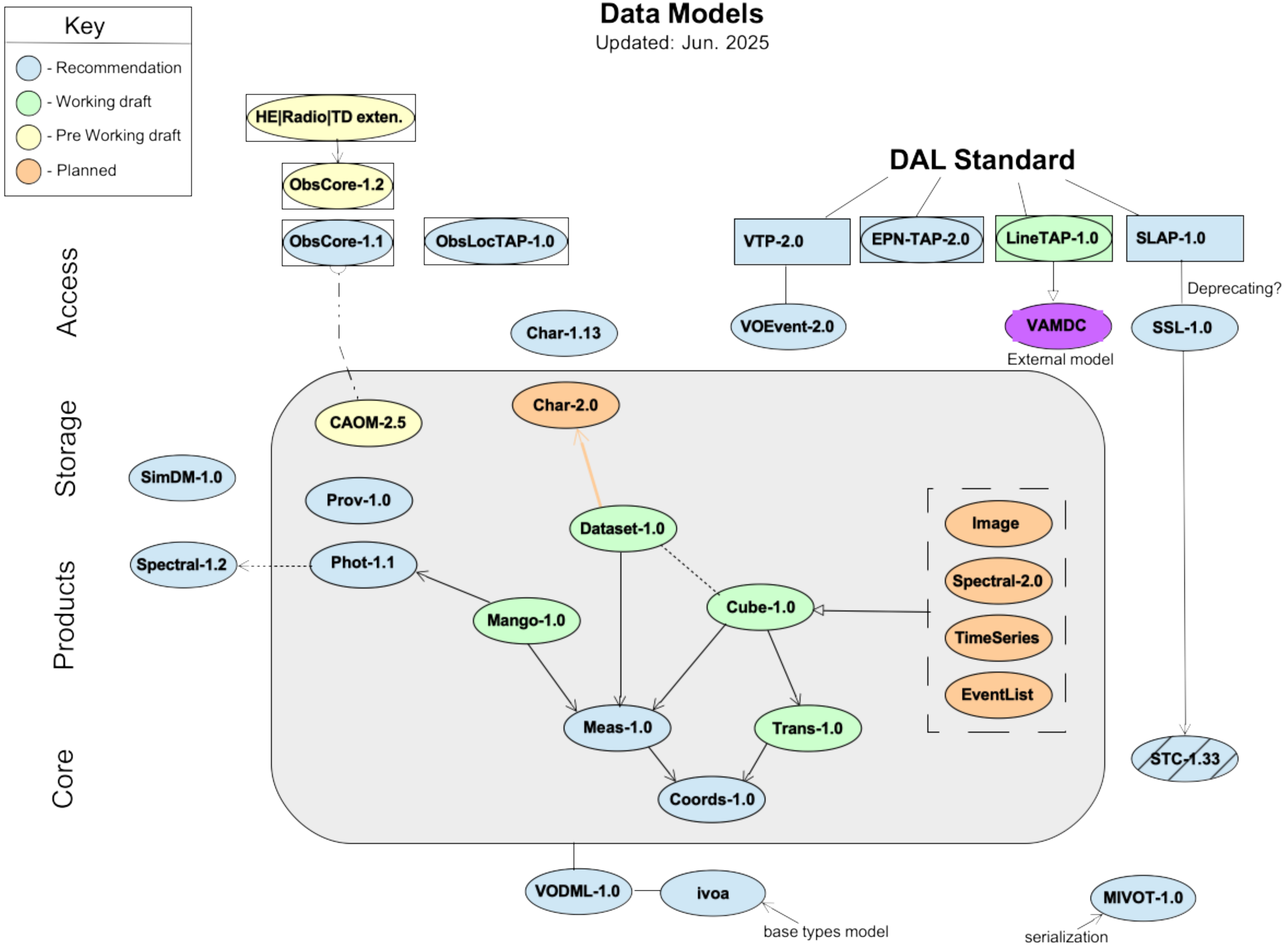
College Park, MD, USA

Back Story

- This is more-or-less a follow-on from earlier presentations
  - Sydney: https://wiki.ivoa.net/internal/IVOA/InterOpMay2024DM/session_intro.pdf
    - Introduction to the issues arising from projects like CAOM integration, MANGO implementations
  - Malta: https://wiki.ivoa.net/internal/IVOA/InterOpNov2024DM/modelOverlap_Nov2024.pdf
    - Highlights where our models overlap in content, with a focus on the mechanism for ensuring consistency, (a hard problem).
- This talk/discussion is more about the system Architecture.
  - What are the expectations on Data Models from our various users?
  - How are these organized within our Standards?
  - Project management: cross-group engagement

Data Models

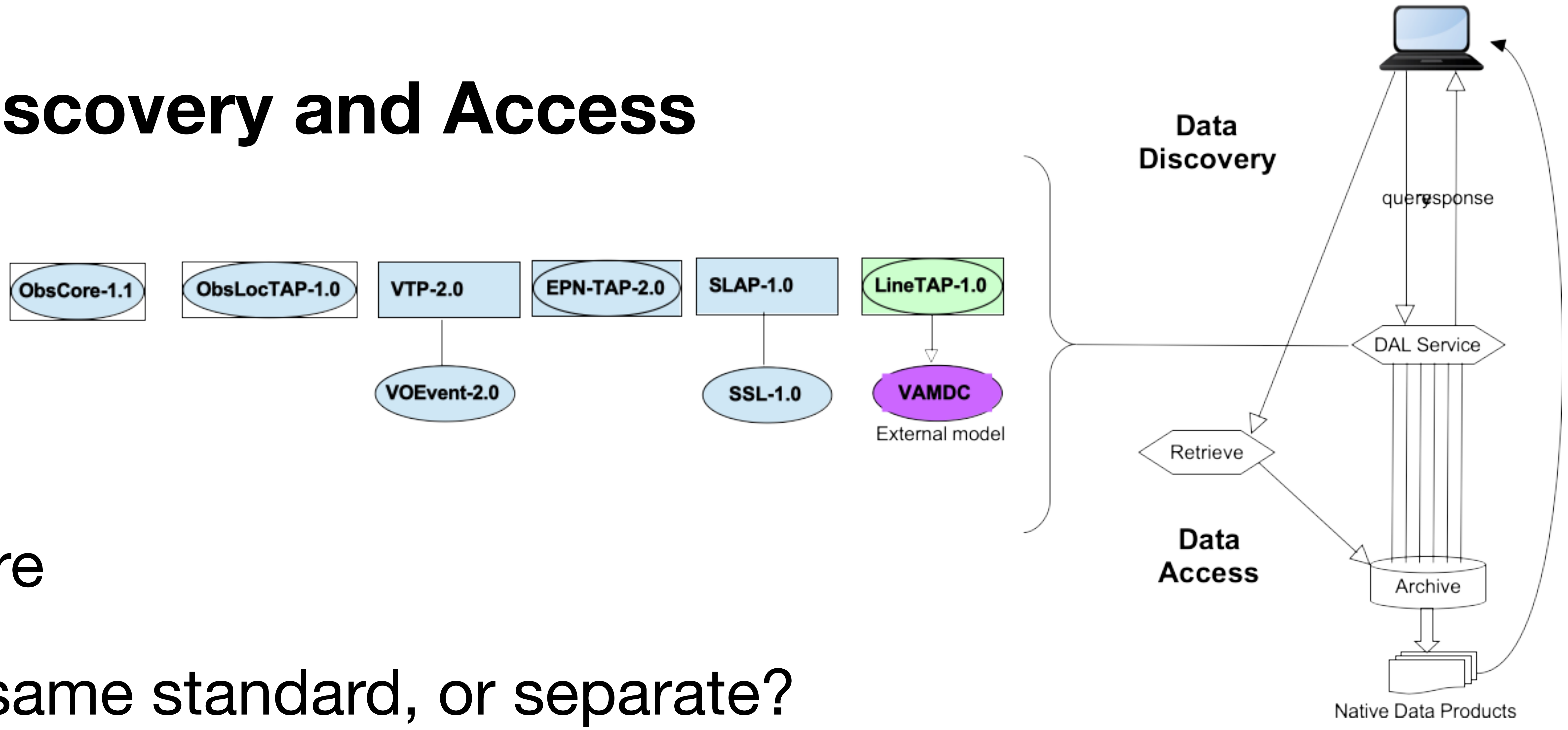
Updated: Jun. 2025



INTERFACES

Interfaces

Model driven Data Discovery and Access



- Simple Workflow
- Well exercised
- Inconsistent Architecture
  - Model & Protocol in same standard, or separate?
    - VOEvent-2.0 separated the model from unpopular protocol
    - LineTap-1.0 planning on keeping both in one standard
  - Who owns the standard… DAL or DM?

Interfaces

Different angle

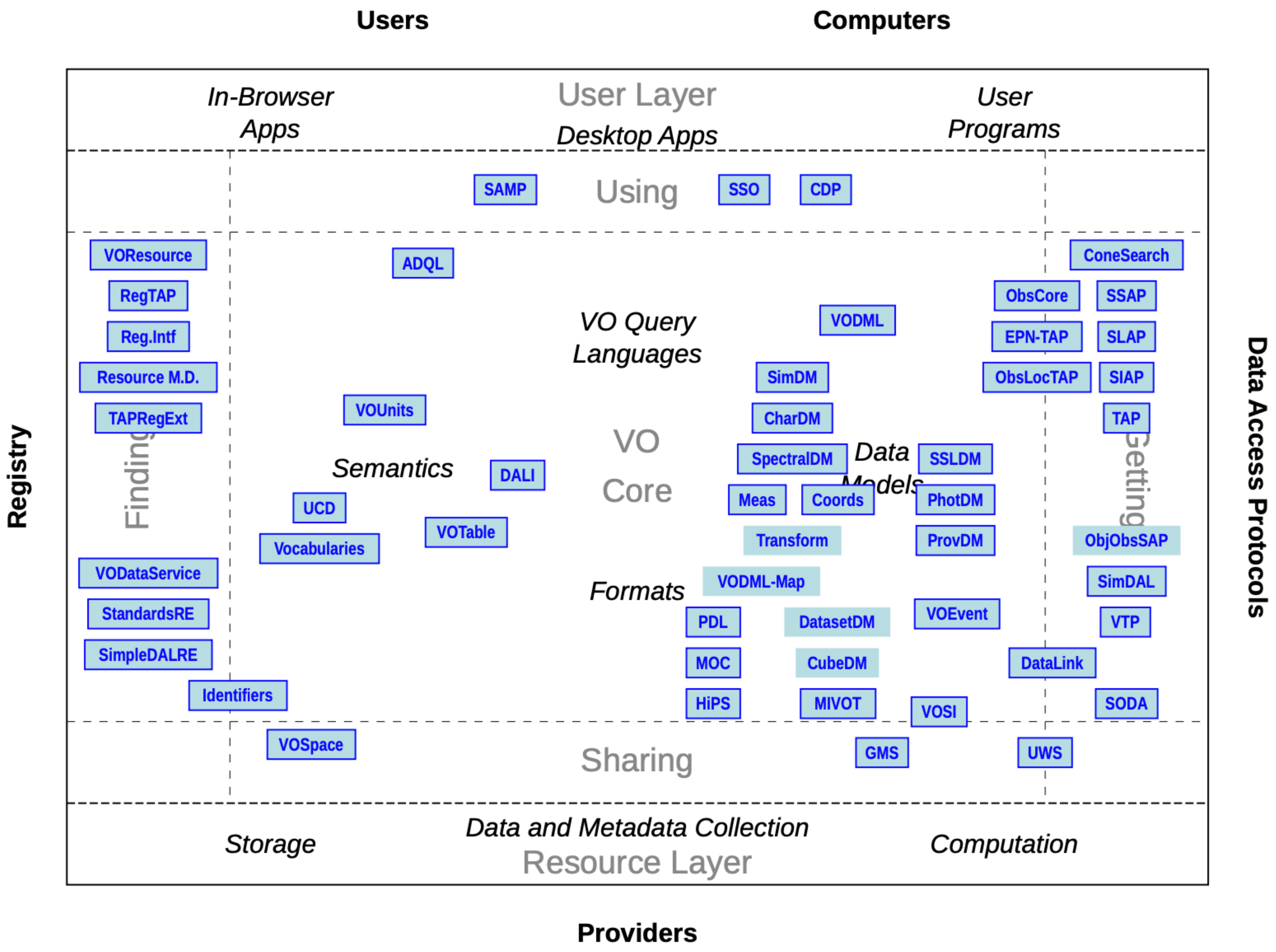
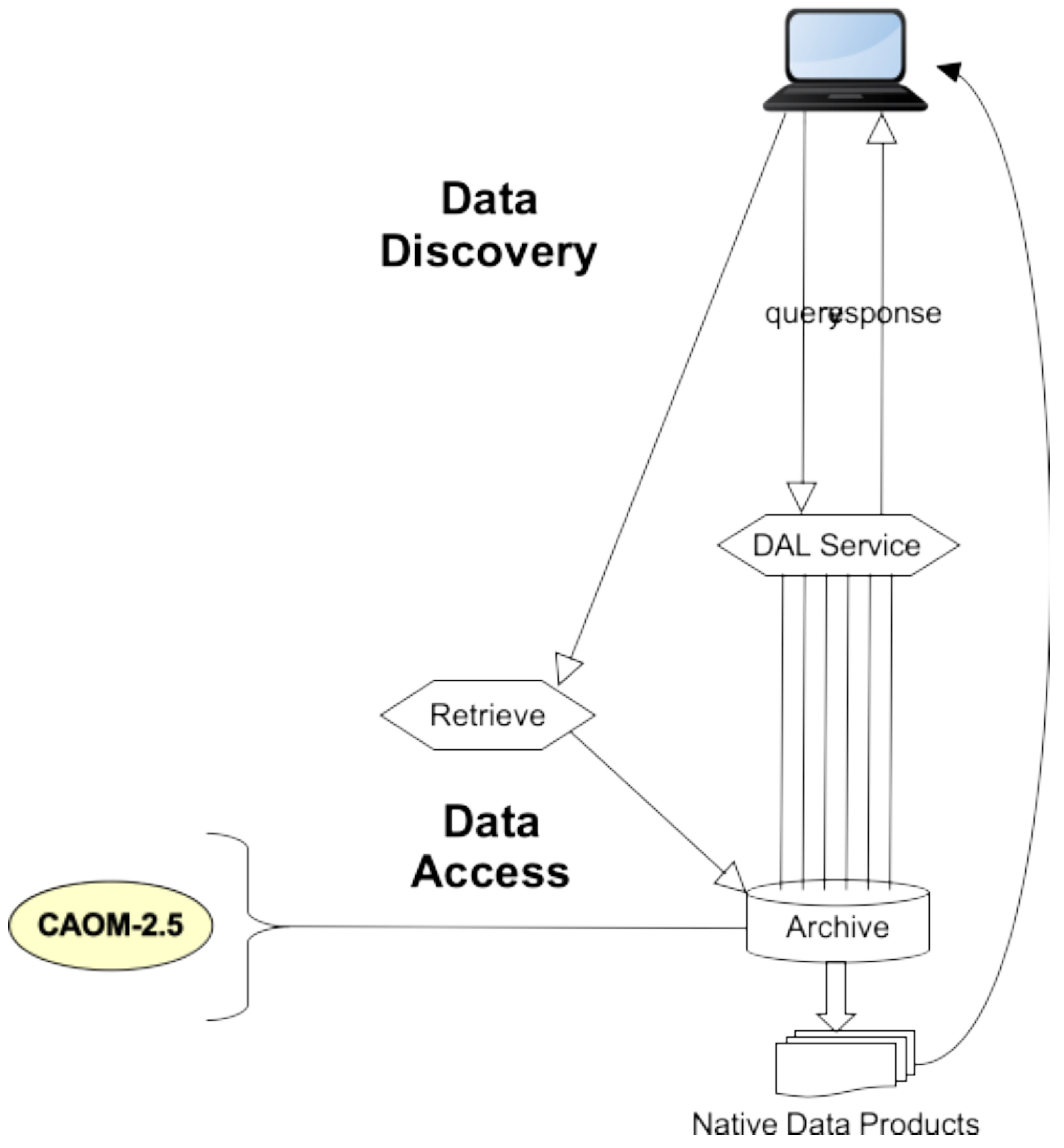


Figure 3: IVOA Architecture Level 2

Interfaces

Model driven Archive Design

- New type of model for IVOA
- Common archive design facilitates code sharing
- Improves compatibility with DAL protocols (ObsCore)



Interface

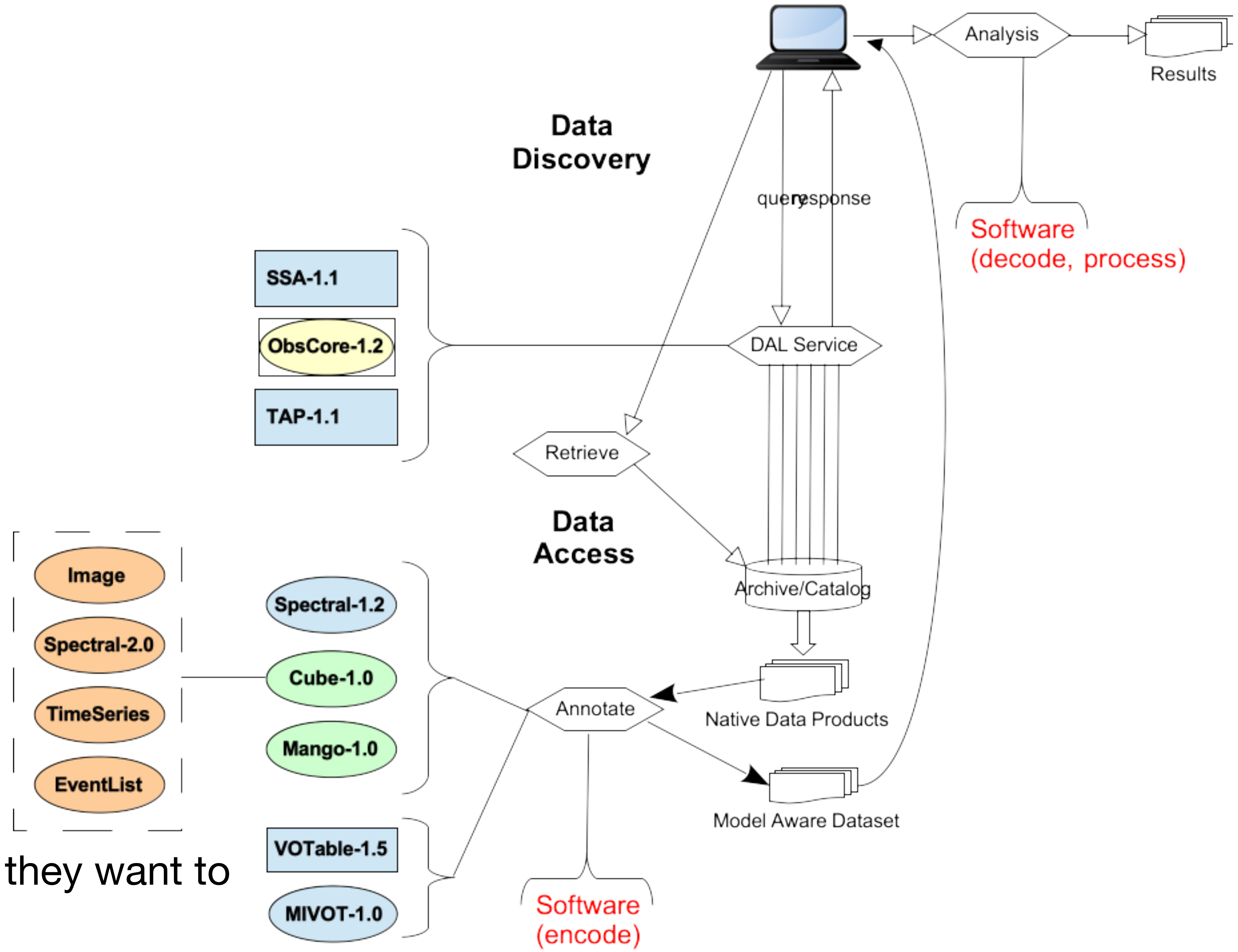
Data Analysis - serializations form interface between DM and APPS

- HiPS3D - applying HiPS to cubic data (APPS session - P. Fernique)
  - Describes 'avalanche of cubic data', and "IVOA has to invent solutions to extend our tools and standards to this evolution" << eg: Cube model
  - HiPS serialization format, for use by applications, should be backed by a data model.
  - For this particular workflow, the serialization is specialized enough to be separate from the model. It is an implied model behind the scenes.
- Some workflows are loosely tied to models via interpretation of VOTables with UCDs. UCDs are basically tags mapping to undisclosed 'hinted' model components. Enough for one to know what the element is, but not how it relates to others.
- Other applications/workflows (SPLAT/Firefly, Catalog analysis) support truly model driven analysis.. via direct mapping of data elements to model components.
  - The EpochPosition object enables organizing several table columns into a complex object with very specific roles on each element. With that object, the software can easily interpret the data to perform the required operation.
- The dependency on the underlying model is different in each case, but the model should always be there. Mapping Serializations to models (especially VODML compliant models) means instances can be easily absorbed/transformed/compared with instances in other formats.

Interfaces

Model driven Data Analysis

- Spectrum paved the way (version 1.0 in 2007)
    - simple VOTable with UTypes
    - Works fine for simple spectra
- Newer efforts with VODML models
    - Software support for this workflow done internally
    - Needs more cross-group engagement
- Requirements/constraints at Analysis end may effect what they want to see from the Annotation phase. e.g. EpochPosition
    - Different threads will have different requirements.
    - Infinite number of analysis threads
    - Where and how is this accommodated? Model, Annotation, Software
    - Where is this documented? Which Standards?



Levels

Model Overlap / “Local Models”

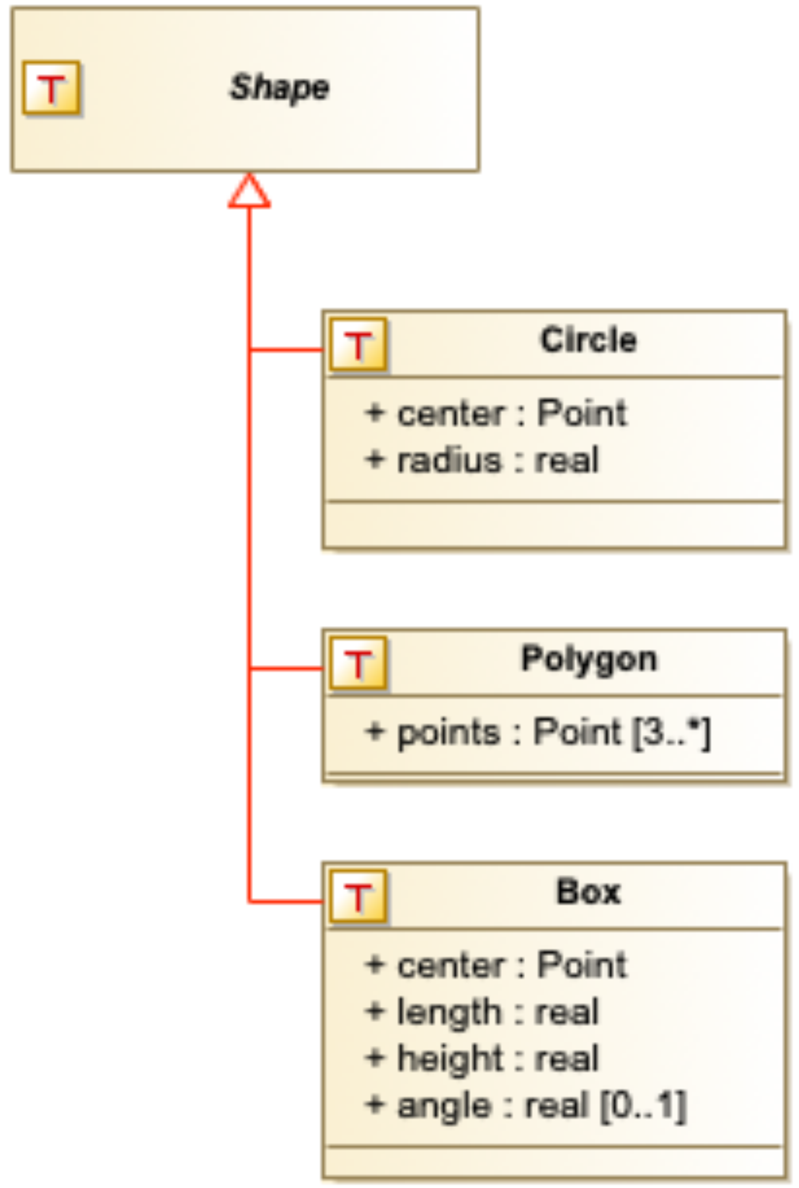
Levels

- Relates to Interfaces topic
  - Different clients and use cases impose different constraints which can **allow** simpler ‘local model’ derived from a more general ‘conceptual model’.
    - NOTE: the more general model does not always exist in IVOA landscape (e.g. Observation)
- Relates to Organization topic
  - Need a consistent approach for handling these.
- Secondary topic: Handling of Representations..
  - Different ways of instantiating/exporting the exact same instance
  - Can be accommodated at different levels (model, annotation, software)

Cases:

Shape

CAOM/DALI

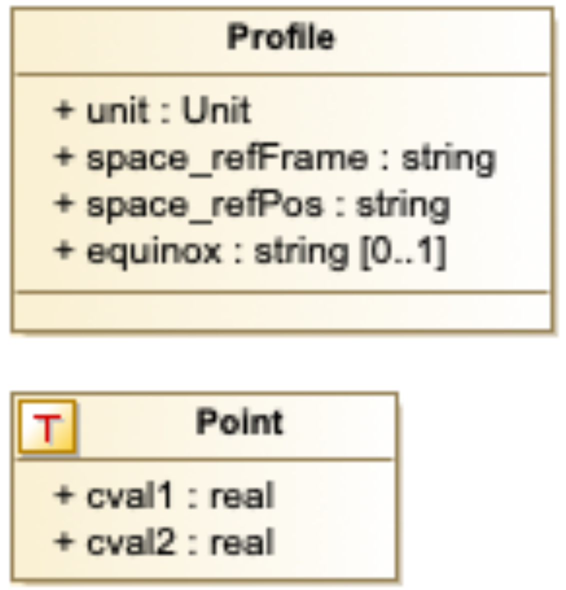


T Shape

T Circle
+ center : Point
+ radius : real

T Polygon
+ points : Point [3..*]

T Box
+ center : Point
+ length : real
+ height : real
+ angle : real [0..1]



Profile
+ unit : Unit
+ space_refFrame : string
+ space_refPos : string
+ equinox : string [0..1]

T Point
+ cval1 : real
+ cval2 : real

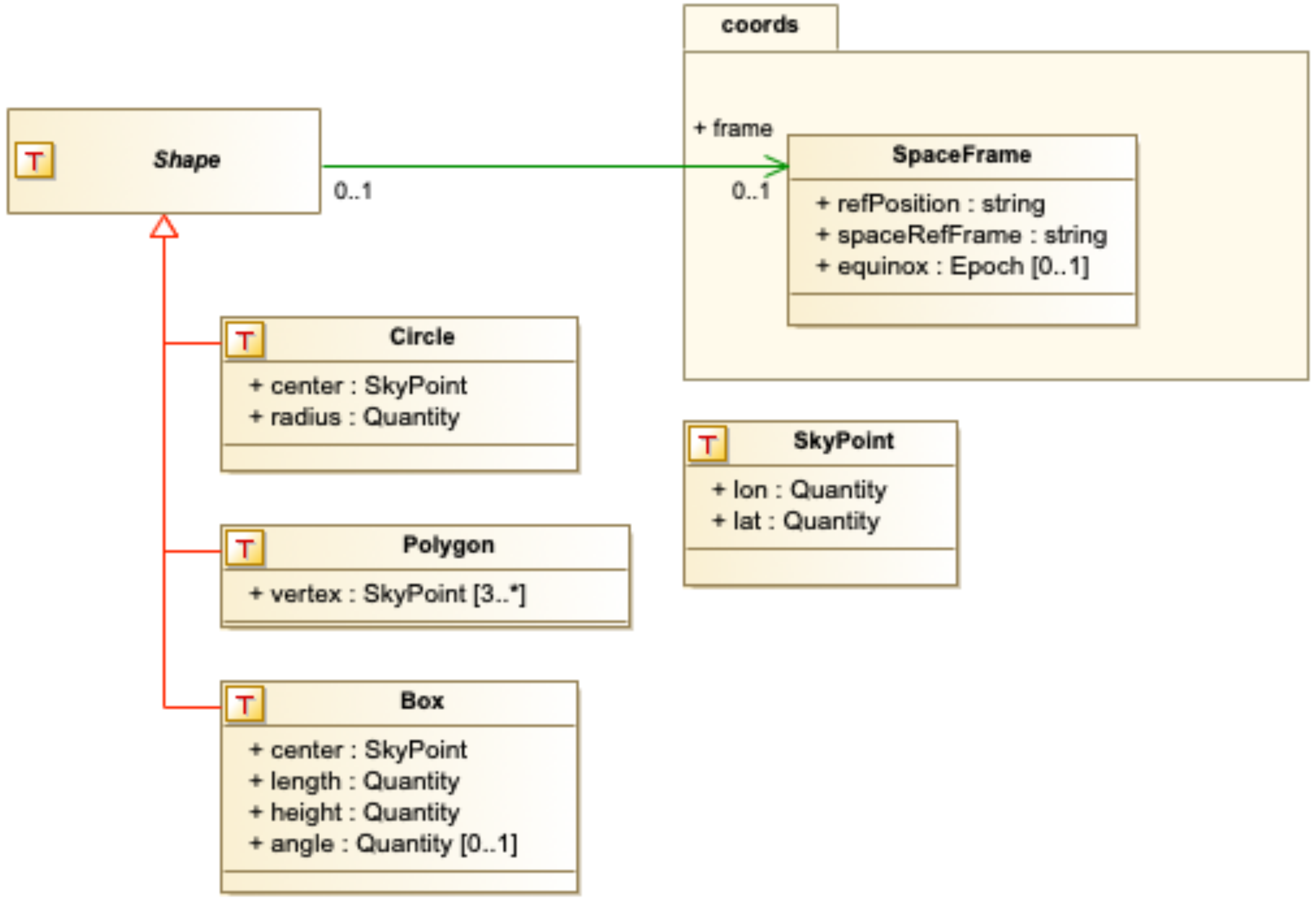
Soft constraints

- constant for all instances
- may vary at table/provider level

Hard constraints

- Spherical coord. space (unit sphere)
- Standard reference position
- (cval1,cval2) map to (lon,lat)

STC-Region-ish



T Shape

0..1

coords

+ frame

0..1

SpaceFrame
+ refPosition : string
+ spaceRefFrame : string
+ equinox : Epoch [0..1]

T Circle
+ center : SkyPoint
+ radius : Quantity

T SkyPoint
+ lon : Quantity
+ lat : Quantity

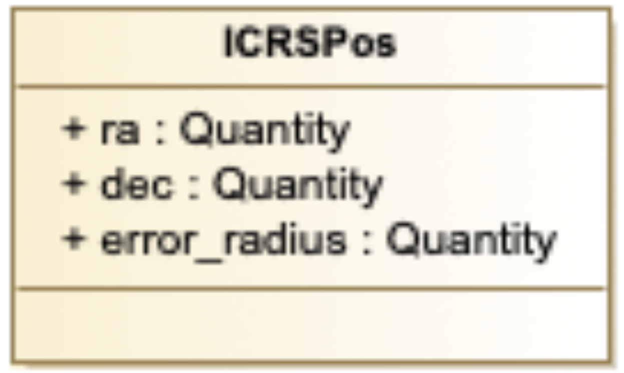
T Polygon
+ vertex : SkyPoint [3..*]

T Box
+ center : SkyPoint
+ length : Quantity
+ height : Quantity
+ angle : Quantity [0..1]

Cases:

Position

DAL level



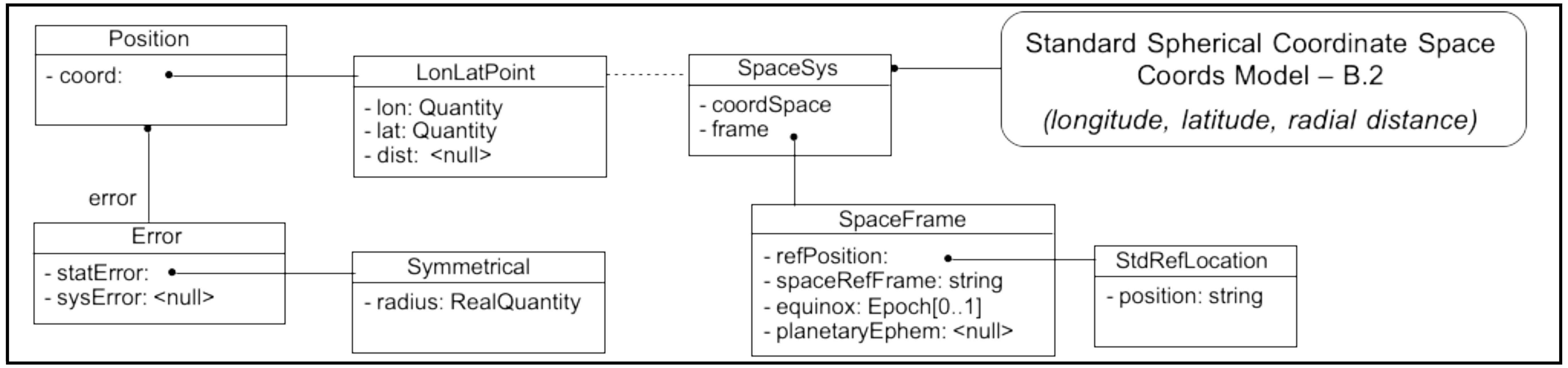
ICRSPos

+ ra : Quantity
+ dec : Quantity
+ error_radius : Quantity

Highly constrained

- Spherical coord. space (unit sphere)
- Fixed reference frame (ICRS)
- Fixed reference position (TOPOCENTER)
- Statistical error circle

Meas/Coords level



Position
- coord:

error

Error
- statError:
- sysError: <null>

LonLatPoint
- lon: Quantity
- lat: Quantity
- dist: <null>

Symmetrical
- radius: RealQuantity

SpaceSys
- coordSpace
- frame

Standard Spherical Coordinate Space Coords Model – B.2

(longitude, latitude, radial distance)

SpaceFrame
- refPosition:
- spaceRefFrame: string
- equinox: Epoch[0..1]
- planetaryEphem: <null>

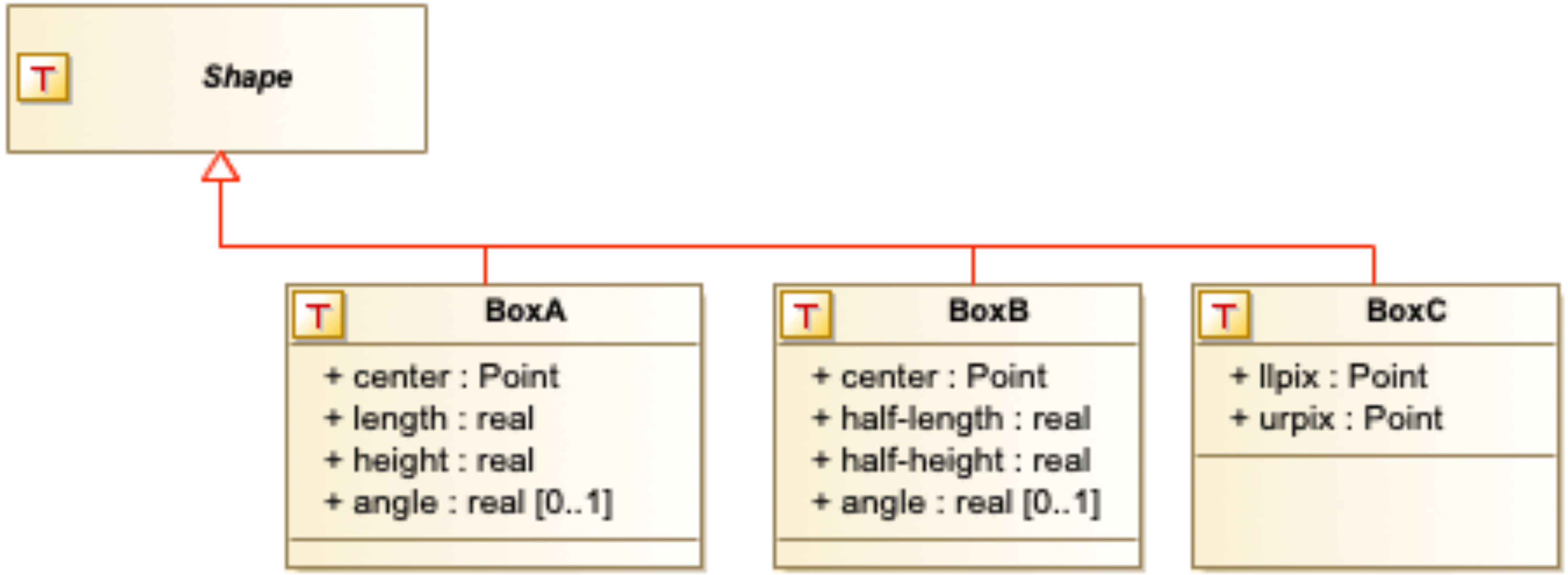
StdRefLocation
- position: string

Full Description

Cases:

Multiple API

- Objects where there are >1 way to construct the very same instance

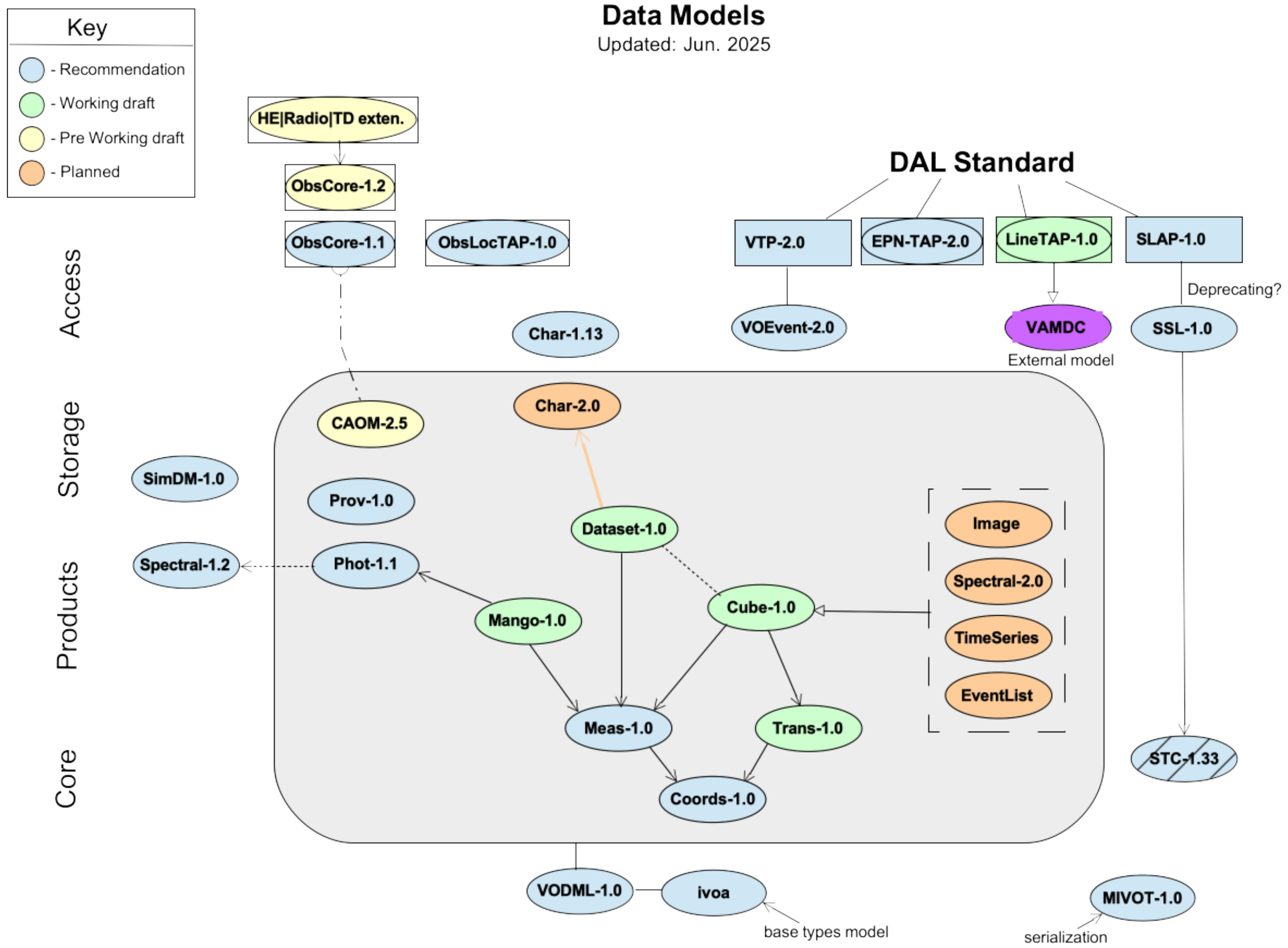


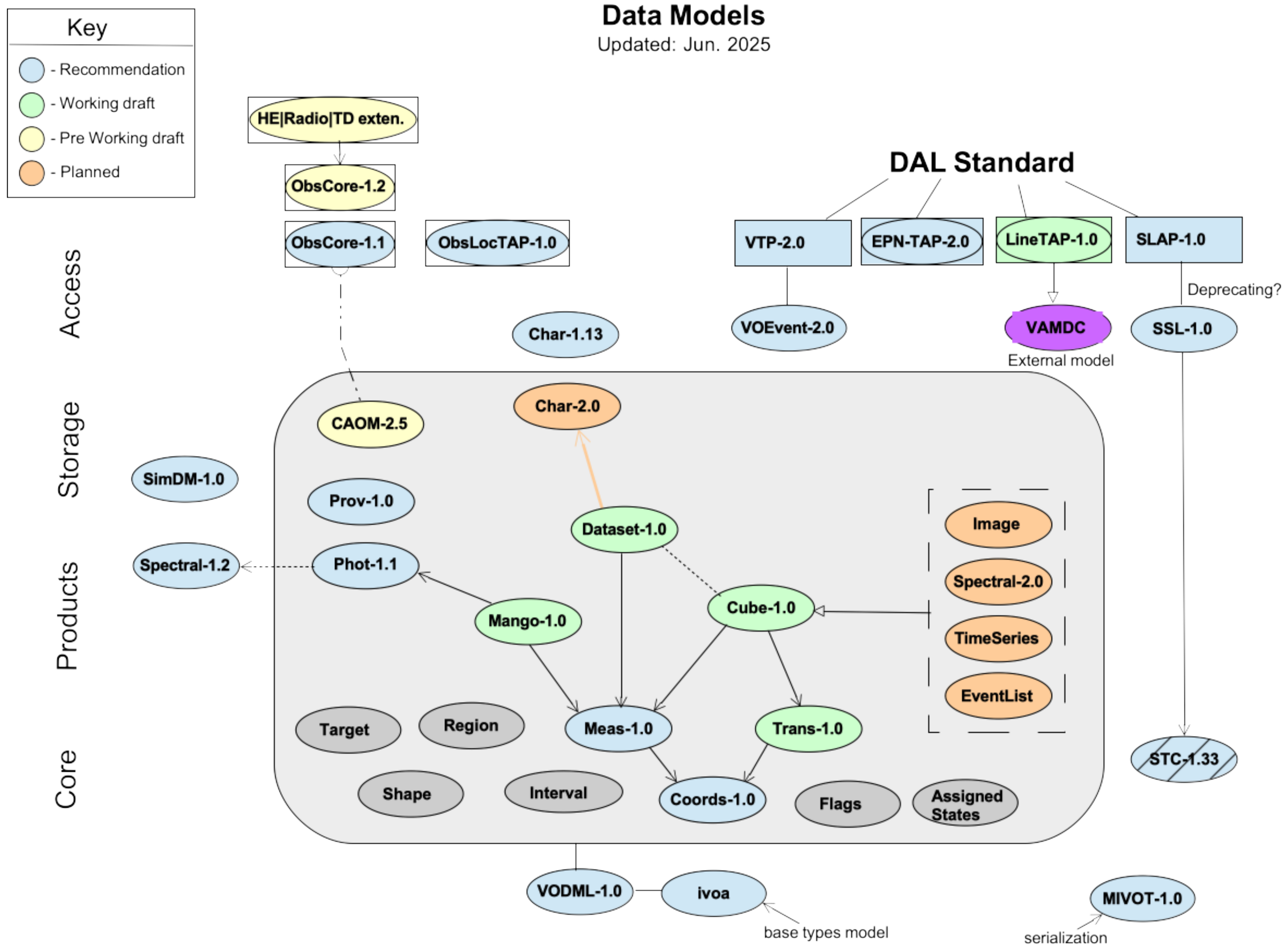
Modularity

Modularity

Atomic ‘building block’ models?

- VO-DML facilitates the reuse of modeled concepts. It is fundamental to its design. This encourages the development of small, atomic models that can be re-used in various contexts.





Modularity

Atomic ‘building block’ models

- This will lead to a build-up of these ‘brick’ models which are not generally useful for someone looking for the ‘building’ model.
  - How would we deal with this in the Docs and Standards repo? (for example)
- If we want to keep the simple constrained models with their full counterparts, the Atomic design is the way to go. Keep each standard related to 1 concept.
- If we have atomic models: when use cases involve a modeled concept, that model MUST be used.
  - If the requirements don’t quite fit, an update to the atomic model is needed
- This may be mostly a DM Internal thing, but there are consequences for standards review process.
  - Any given project is more likely to involve >1 model standard in the review process, which is not the typical review flow.
  - Having 1 model for any given concept can be hard to maintain as the number of use cases increases.
  - More likelihood for a breaking change?

Discussion Questions:

- **Interfaces:**
  - Is it a goal of the IVOA (long or short term) to facilitate model driven data analysis?
    - e.g. Slice a nDimensional cube to make Spectrum or TimeSeries
    - e.g. Analysis on Catalog data (epoch propagation == tip of the iceberg)
  - Rules for interfaces.. when to combine in 1 standard, when to separate.
  - Are we in the software business?
    - Serialization std gets I/O software; but are we responsible for creating processing software (pyvo)? Is DM responsible for this?
- **Levels:**
  - Keep concepts together (same standard)? Or distributed with usage?
    - What about ‘generally useful’ constrained objects? (ICRSPOS)
    - If separate, how would one know a particular constrained version exists? (EpochPosition)
      - Could another model re-use JUST that object? (Without importing all of MANGO?)
  - Do we want to relate child models to their progenitor(s)? formally or informally?
- **Modularity:**
  - Atomic ‘reusable building-block’ models or more independent/complete models?
    - How to organize for external view? (D&S page, Architecture diagram)

TCG Approach to Projects

- A common DM project request: “Update the Spectrum model”, “We need a model for N-Dimensional Data”, “We need a Catalog model”
    - But they aren’t linked with a broader workflow.
        - For Spectrum: had to kill the project because we couldn’t get implementation support.. even from existing applications like SPLAT / Firefly.
        - For Cube: once DAL component was done, the project was declared complete. I hesitate to start into the RFC for this model without support, and answers to these questions.
        - For MANGO: no project outside DM for exercising the model.
- We (TCG) need to either:
    - Accept that “implementation of DM” is NOT an implemented science workflow (as agreed in 2016)
    - and/or insist that DM projects MUST be a component of a broader use case the IVOA wants to enable and includes dependencies with other working groups.