IVOA Provenance Table Access Protocol (ProvTAP)

Version 1.0

IVOA Working Draft 2019-03-22

Author(s)
    François Bonnarel, Mireille Louys, Markus Nullmeier, Kristin
    Riebe, Michèle Sanguillon, Mathieu Servillat, IVOA Data Model
    Working Group
Editor(s)
    François Bonnarel

Abstract

This document describes the ProvTAP protocol for accessing provenance information according to the IVOA ProvenanceDM standard. It defines how the elements of ProvDM are described in the TAP schema tables and provides guidelines for implemnienting with TAP 1.1.

Status of this document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/documents/.

Contents

1 Introduction

1.1 context

ProvTAP is a TAP service specialization for delivering IVOA Provenance metadata an build up serialisation documents consistently with the IVOA Provenance data model (Riebe and Servillat et al., 2017) As for any TAP

service (Dowler and Rixon et al., 2010), ProvTAP is providing ADQL query responses consisting of a single table each, gathering columns selected from various tables defined in the service's TAP schema. The default format of the Query Response is VOTable. Other formats (JSON, tsv, etc.) may also be optionally supported.

The definition of a ProvTAP service is similar to the approach used in the ObsCore specification (Louys and Bonnarel et al., 2011) where ObsCore data model plays the role of underlying model for ObsTAP services. It is also inspired by the multiple table design proposed in the the RegTAP specification (Demleitner and Harrison et al., 2013) with respect to VOResource data model.

Basically, data-model consistency for a TAP service is asserted by the datamodel element in the capability of the service. The relational view of the model is expressed via the table structure in the TAP schema and the definition of each column datatype, ucd, utype attributes in the same schema.

ProvTAP is not the only DAL service able to provide provenance metadata. The ProvSAP (**?**) specification offers a complementary method based on a search-by-parameters strategy and does not require the server to adopt the TAP architecture. ProvSAP services allow to retrieve sequences or subgraphs of Provenance metadata associated to a starting entity or activity allowing different combinations.

1.2   Role within the VO architecture

IVOA ProvTAP specification is used to define TAP services with a specific TAP schema expressing view of the IVOA provenance Data Model. It relies on DALI for general DAL definitions. It benefits from existing IVOA notations and standards like UCD, VOUnits, VOTable, VOResource and extensions of the latter such as VODataservice and TAPREgExt.

Fig. 1 shows the dependencies of this document with respect to other existing standards.

2   ProvTAP in action

Here are a few examples of ADQL queries for ProvTAP (values are taken from the test metadata presented in appendix) :

- To retrieve all activity metadata for activies sharing the same activityDescription, by looking for the activities with the same a_description value:
  `SELECT * FROM Activity WHERE`
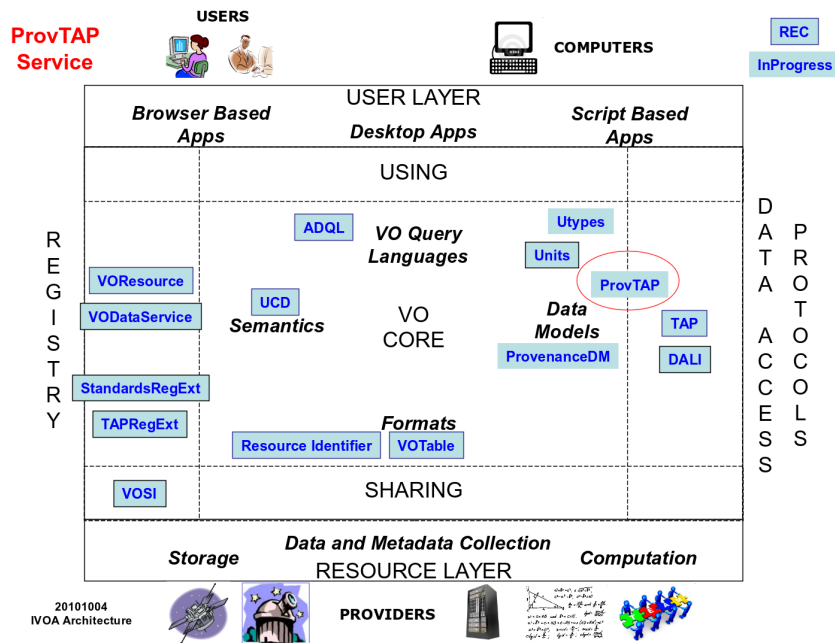 `Activity.a_description = 'hipsgen15'`

3

Figure 1: Architecture diagram for the ProvTAP specification. It is based on a view of IVOA Provenance data model an relies on TAP access technology. As such it is fully integrated in the IVOA framework

- To retrieve all activities associated with a specific agent:
  ```
  SELECT WasAssociatedWith.waw_activity_id, Activity.a_name,
         Activity.a_comment FROM WasAssociatedWith
  INNER JOIN Activity ON
  WasAssociatedWith.waw_activity_id = Activity.id
  WHERE WasAssociatedWith.waw_agent_id = 'agent_1_1'
  ```

- To retrieve all entities identifiers attributed to `curator` agents:
  ```
  SELECT WasAttributedTo.wat_entity_id FROM WasAttributedTo
  WHERE WasAttributedTo.wat_role = 'curator'
  ```

I have to modify, and enrich this section above by adding new queries

The result of the first ADQL query is the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:ex="http://www.example.com/provenance"
  xmlns:ivo="http://www.ivoa.net/documents/rer/ivo/"
  xmlns:voprov="http://www.ivoa.net/documents/dm/provdm/voprov/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd">
  <RESOURCE type="provenance">
    <DESCRIPTION>Provenance VOTable</DESCRIPTION>
    <TABLE >
```

```
<FIELD id="a_ID" name="a_id" ucd="meta.id" utype="voprov:Activity.id" arraysize="*" datatype="char"/>
<FIELD id="a_nameID" name="a_name" ucd="meta.title" utype="voprov:Activity.name" arraysize="*" datatype="ch
<FIELD id="a_startID" name="a_startTime" ucd="time.start" utype="voprov:Activity.startTime" arraysize="*" d
<FIELD id="a_endID" name="a_endTime" ucd="time.start" utype="voprov:Activity.endTime" arraysize="*" datatyp
<FIELD id="a_commentID" name="a_comment" ucd="meta.description" utype="voprov:Activity.comment" arraysize="
<FIELD id="a_descID"  name="a_description" ucd="meta.id" utype="voprov:Activity.description_id" datatype="c
<DATA>
  <TABLEDATA>
    <TR>
      <TD>act:CDS/P/HI4PI/NHI</TD>
      <TD>Generation of HI4PI NHI  HiPS</TD>
      <TD>2011-02-14T12:00</TD>
      <TD>2011-02-14T12:00</TD>
      <TD>Generation of HI4PI NHI survey (full-sky HI column density distribution) HiPS</TD>
      <TD>hipsgen15</TD>
    </TR>
  </TABLEDATA>
</DATA>
</TABLE>
<INFO name="QUERY_STATUS" value="OK"/>
</RESOURCE>
</VOTABLE>
```

let's change what is above to be consistent with the prototype and query

3   VOSI availability and capabilities

According to the DALI specification for VO services (Dowler and Demleitner et al., 2013), a provenance service implementing ProvTAP must provide a VOSI availability interface as well as a capabilities interface with entries for ProvTAP. The `standardId`s for these provenance interfaces are:

`ivo://ivoa.net/std/ProvenanceDM#ProvTAP-1.0`

The capability for a TAP service to support the Provenance DM is expressed by the dataModel element as:

```
<dataModel ivoid="ivo://ivoa.net/std/ProvenanceDM-1.0">
  ProvenanceDM-1.0
</dataModel>
```

For ProvTAP, the VOSI tables interface must also be provided.

4   VOTable serialization for provenance metadata and ProvTAP TAP schema

The table and column definitions available in the TAP schema, including their metadata (`unit`, `ucd`, `utype`, `datatype`, `description`) exactly define the VOTable mapping of the IVOA Provenance data model (see table and column definitions and metadata.

5

Below is a serialisation format mapping the Provenance data model into VOTable. This format is useful to provide full serialisation of Provenance metadata for an archive or data reduction workflow in VOTable. It can also be used to build the TAP schema of a ProvTAP service.

4.1 Tables in the VOTable serialisation or in the ProvTAP TAP schema

The following table gives the full list of tables necessary to map the IVOA Provenance data model. The contents of the 'name' and 'utype' columns of this table are normative.

Name	utype	Description	status
DataSetEntity	voprov:DataSetEntity	instances of *DataSetEntity* class	Mandatory
ValueEntity	voprov:ValueEntity	instances of *ValueEntity* class	Mandatory
ValueDescription	voprov:ValueDescription	instances of *ValueDescription* class	Mandatory
DatasetDescription	voprov:DatasetDescription	instances of *DatasetDescription* class	Mandatory
Activity	voprov:Activity	instances of Actvity class	Mandatory
ActivityDescription	voprov:ActivityDescription	instances of *ActivityDescription* class	Mandatory
Agent	voprov:Agent	instances of *Agent* class	Mandatory
Parameter	voprov:Parameter	instances of *Parameter* class	Optional
ParameterDescription	voprov:ParameterDescription	instances of *ParameterDescription*	Optional
ConfigFile	voprov:ConfigFile	instances of *ConfigFile* class	Optional
ConfigFileDescription	voprov:ConfigFileDescription	instances of *ConfigFileDescription*	Optional
Used	voprov:Used	instances of *Used* relationships	Mandatory
UsageDescription	voprov:UsageDescription	instances of *UsageDescription* relationships	Mandatory
GenerationDescription	voprov:GenerationDescription	instances of *GenerationDescription relationships*	Mandatory
WasGeneratedBy	voprov:WasGeneratedBy	instances of *WasGeneratedBy* relationships	Mandatory
WasAssociatedWith	voprov:WasAssociatedWith	instances of *WasAssociatedWith* relationships	Mandatory
WasAttributedTo	voprov:WasAttributedTo	instances of *WasAttributedTo* relationships	Mandatory
WasConfiguredBy	voprov:WasConfiguredBy	instances of *WasConfiguredBy* relationships	Optional
WasDerivedFrom	voprov:WasDerivedFrom	instances of *WasDerivedFrom* relationships	Optional
WasInformedBy	voprov:WasInformedBy	instances of *WasInformedBy* relationships	Optional

Table 1: List of tables in the TAP schema

> Above: do we add Collections ? The idea was that collection was a "type of entity". So no specific table for that ? a "collection" attribute should contain the id of the collection entity for normal entities considered as members ?

6

When the ProvTAP service is requested by generic ADQL queries involving optional tables which it doesn't include, it should not fail and the response MUST be a table headed by the queried columns names and containing no data. An example of this is a query constraining the unit or ucd of a parameter sent to a service not implementing the configuration classes.

4.2 Detailed table description

Tables 2 to 19 provide full descriptions of the FIELDS of each of the mapping tables. The contents of all these tables' columns are normative. For convenience, in all these tables an arrow ($\rightarrow$) indicates columns that contain references (foreign keys) to other tables. The column names are prefixed by a short-cut for the table name in order to ensure unique column names in queries with table joins for ProvTAP.

Name	ucd	utype	datatype
de_id	meta.id	voprov:DatasetEntity.id	char
de_name	meta.title	voprov:DatasetEntity.name	char
de_type	meta.code.class	voprov:DatasetEntity.type	char
de_rights	meta.code.class	voprov:DatasetEntity.rights	char
de_location	meta.ref.url	voprov:DatasetEntity.location	char
de_generated	time.start	voprov:DatasetEntity.generatedAtTime	char
de_invalidated	time.stop	voprov:DatasetEntity.invalidatedAtTime	char
de_comment	meta.description	voprov:DatasetEntity.comment	char
$\rightarrow$ de_collection	meta.id.parent	voprov:DatasetEntity.collection	char
$\rightarrow$ de_description	meta.id	voprov:DatasetEntity.description_id	char
$\rightarrow$ de_usedEntity	meta.id	voprov:DatasetEntity.wasDerivedFrom.usedEntity	char

Table 2: Column description for DatasetEntity table

Name	ucd	utype	datatype
ve__id	meta.id	voprov:ValueEntity.id	char
ve__name	meta.title	voprov:ValueEntity.name	char
ve__type	meta.code.class	voprov:ValueEntity.type	char OPTION
ve__rights	meta.code.class	voprov:ValueEntity.rights	char OPTION
ve__comment	meta.description	voprov:ValueEntity.comment	char
ve__location	meta.ref.url	voprov:ValueEntity.location	char
ve__generated	time.start	voprov:ValueEntity.generatedAtTime	char
ve__invalidated	time.stop	voprov:ValueEntity.invalidatedAtTime	char
ve__value	stat.value	voprov:ValueEntity.value	char
→ ve__collection	meta.id	voprov:ValueEntity.collection	char
→ ve__description	meta.id	voprov:ValueEntity.description__id	char
→ ve__usedEntity	meta.id	voprov:ValueEntity.wasDerivedFrom.usedEntity	char

Table 3: Column description for ValueEntity table. The value of type and subtype strings belong to a set of possible literal values given for instance using the VOTable OPTION element.

Above: do we also add a generic Entity table ?

Name	ucd	utype	datatype
dd__id	meta.id	voprov:DatasetDescription.id	char
dd__name	meta.title	voprov:DatasetDescription.name	char
dd__description	meta.description	voprov:DatasetDescription.description	char
dd__content	meta.description	voprov:DatasetDescription.contentType	char
dd__type	meta.code.class	voprov:DatasetDescription.type	char OPTION
dd__subtype	meta.code.class	voprov:DatsetDescription.subtype	char
dd__doculink	meta.ref.url	voprov:DatasetDescription.doculink	char

Table 4: Column description for DatasetDescription table. The value of type string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
vd_id	meta.id	voprov:VaueDescription.id	char
vd_name	meta.title	voprov:ValueDescription.name	char
vd_description	meta.description	voprov:ValueDescription.description	char
vd_type	meta.code.class	voprov:ValueDescription.type	char OPTION
vd_subtype	meta.code.class	voprov:ValueDescription.subtype	char OPTION
vd_doculink	meta.ref.url	voprov:ValueDescription.doculink	char
vd_valueType	meta	voprov:ValueDescription.valueType	char OPTION
vd_unit	meta.unit	voprov:ValueDescription.unit	char
vd_ucd	meta.ucd	voprov:ValueDescription.ucd	char
vd_utype	meta	voprov:ValueDescription.utype	char
vd_min	stat.min	voprov:ValueDescription.min	char
vd_max	stat.max	voprov:ValueDescription.max	char
vd_default	meta	voprov:ValueDescription.doculink	char
vd_options	meta	voprov:ValueDescription.doculink	char

Table 5: Column description for ValueDescription table. The value of type, subtype and valeueType strings belong to a set of possible literal values given for instance using the VOTable OPTION element.

> do you want to suppress comment in value and Dataset Descriptions ? I think it's useful to keep them. Do we also add a generic Entity Description table ?

Name	ucd	utype	datatype
a_id	meta.id	voprov:Activity.id	char
a_name	meta.title	voprov:Activity.name	char
a_startTime	time.start	voprov:Activity.startTime	char
a_endTime	time.stop	voprov:Activity.endTime	char
a_comment	meta.description	voprov:Activity.comment	char
→ a_description	meta.id	voprov:Activity.description_id	char
→ a_informant	meta.id	voprov:Activity.wasInformedBy.informant	char

Table 6: Column description for Activity table

Name	ucd	utype	datatype
ad_id	meta.id	voprov:ActivityDescription.id	char
ad_name	meta.title	voprov:ActivityDescription.name	char
ad_type	meta.code.class	voprov:ActivityDescription.type	char OPTION
ad_subtype	meta.code.class	voprov:ActivityDescription.subtype	char OPTION
ad_description	meta.description	voprov:ActivityDescription.description	char
ad_doculink	meta.ref.url	voprov:ActivityDescription.doculink	char

Table 7: Column description for ActivityDescription table. The value of type and subtype strings belong to a set of possible literal values given for instance using the VOTable OPTION element.

do you want to suppress "comment" in activity description ? I think it's useful to keep it

Name	ucd	utype	datatype
ag_id	meta.id	voprov:Agent.id	char
ag_name	meta.title	voprov:Agent.name	char
ag_type	meta.code.class	voprov:Agent.type	char OPTION
ag_address	meta.address	voprov:Agent.address	char
ag_email	meta.email	voprov:Agent.email	char
ag_affiliation	meta	voprov:Agent.affiliation	char
ag_phone	meta.	voprov:Agent.phone	char
ag_comment	meta.description	voprov:Agent.comment	char

Table 8: Column description for agent table. The value of type string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
p_activity	meta.id	voprov:Parameter.activity_id	char
p_id	meta.id	voprov:Parameter.id	char
p_name	meta.title	voprov:Parameter.name	char
p_value	stat.value	voprov:Parameter.value	param dependent
→ p_description	meta.id	voprov:Parameter.parameterDescription_id	reference to parameter description

Table 9: Column description for Parameter table

Do we really need the name here. Duplication with ParameterDescription ?

Name	ucd	utype	datatype
cf_name	meta.title	voprov:ConfigFile.name	char
cf_comment	meta.description	voprov:ConfigFile.comment	char
cf_location	meta.ref.url	voprov:ConfigFile.location	char
→ cf_description	meta.id	voprov:ConfigFile.ConfigFileDescription_id	reference to parameter description

Table 10: Column description for Configuration File

Name	ucd	utype	datatype
wcb_artefact	meta.code	voprov:WasConfiguredBy.artefactType	char
→ wcb_configfile	meta.id	voprov:WasConfiguredBy.ConfigFile_id	char
→ wcb_parameter	meta.id	voprov:WasConfiguredBy.parameter_id	char
→ wcb_activity	meta.id	voprov:WasConfiguredBy.activity_id	char

Table 11: Column description for WasConfiguredBy relationship table

Name	ucd	utype	datatype
pd_activitydescription	meta.id	voprov:ParameterDescription. activityDescription_id	char
pd_id	meta.id	voprov:ParameterDescription.id	char
pd_name	meta.title	voprov:ParameterDescription.name	param dependent
pd_description	meta.description	voprov:ParameterDescription.description	char
pd_datatype	meta	voprov:ParameterDescription.datatype	char
pd_unit	meta.unit	voprov:ParameterDescription.unit	char
pd_ucd	meta.ucd	voprov:ParameterDescription.ucd	char
pd_utype	meta	voprov:ParameterDescription.utype	char
pd_min	stat.min	voprov:ParameterDescription.min	param dependent
pd_max	stat.max	voprov:ParameterDescription.max	param dependent
pd_options	meta	voprov:ParameterDescription.options	param dependent

Table 12: Column description for ParameterDescription table

Name	ucd	utype	datatype
cfd_id	meta.id	voprov:DatasetDescription.id	char
cfd_name	meta.title	voprov:DatasetDescription.name	char
cfd_description	meta.description	voprov:DatasetDescription.description	char
cfd_content	meta.code.mime	voprov:DatasetDescription.contentType	char
cfd_comment	meta.description	voprov:DatasetDescription.comment	char
cfd_type	meta.code.class	voprov:DatasetDescription.type	char
cfd_subtype	meta.code.class	voprov:DatsetDescription.subtype	char
cfd_doculink	meta.ref.url	voprov:DatasetDescription.doculink	char

Table 13: Column description for ConfigFileDescription table

Name	ucd	utype	datatype
→ u_entity	meta.id	voprov:Used.entity_id	char
→ u_activity	meta.id	voprov:Used.activity_id	char
→ u_usedDescription_id	meta.id	voprov:Used.usedDescription_id	char
u_time	time.start	voprov:Used.time	char

Table 14: Column description for Used relationship table

> Do we still need the time there ? Do we add role which is already in usage description ?

Name	ucd	utype	datatype
→ ud_id	meta.id	voprov:UsageDescription. id	char
→ ud_entityDescription	meta.id	voprov:UsageDescription. entityDescription_id	char
→ ud_activityDescription	meta.id	voprov:UsageDescription. activityDescription_id	char
ud_role	meta.code.class	voprov:UsageDescription.role	char OPTION

Table 15: Column description for UsedDescription relationship table. The value of a role string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
→ wgb_entity	meta.id	voprov:WasGeneratedBy.entity_id	char
→ wgb_activity	meta.id	voprov:WasGeneratedBy.activity_id	char
→ wgb_generationDescription	meta.id	voprov:WasGeneratedBy.GenerationDescription_id	char
wgb_role	meta.code.class	voprov:WasGeneratedBy.role	char OPTION
wgb_time	time.stop	voprov:WasGeneratedBy.time	char

Table 16: Column description for WasGeneratedBy relationship table. The value of a role string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
→ gd_id	meta.id	voprov:GenerationDescription. id	char
→ gd_entityDescription	meta.id	voprov:GenerationDescription. entityDescription_id	char
→ gd_activityDescription	meta.id	voprov:GenerationDescription. activityDescription_id	char
gd_role	meta.code.class	voprov:GenerationDescription.role	char OPTION

Table 17: Column description for GenerationDescription relationship table. The value of a role string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
→ waw_agent	meta.id	voprov:WasAssociatedWith.agent_id	char
→ waw_activity	meta.id	voprov:WasAssoatciatedWith.activity_id	char
waw_role	meta.code.class	voprov:WasAssociatedWith.agentRole	char OPTION

Table 18: Column description for WasAssociatedWith relationship table. The value of a role string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
→ wat_entity	meta.id	voprov:WasAttributedTo.entity_id	char
→ wat_agent	meta.id	voprov:WasAttributedTo.agen_id	char
wat_role	meta.code.class	voprov:WasAttributedTo.agentRole	char OPTION

Table 19: Column description for WasAttributedTo relationship table. The value of a role string belongs to a set of possible literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype
→ wib_informant	meta.id	voprov:WasInformedBy.informant_id	char
→ wib_informed	meta.id	voprov:WasInformedBy.informed_id	char

Table 20: Column description for WasInformedBy relationships table

Name	ucd	utype	datatype
→ wdf_usedEntity	meta.id	voprov:WasDerivedFrom.usedEntity_id	char
→ wdf_generatedEntity	meta.id	voprov:WasDerivedFrom.generatedEntity_id	char

Table 21: Column description for WasDerived From relationship table

For the two table above. Do we really need tables or just reentrant pointers (called informant and generatedentity ?

Appendix A   TAP schema in xml

The ProvTAP TAP schema will be found on the IVOA xml document repository (currently available at:
https://wiki.ivoa.net/internal/IVOA/ObservationProvenanceDataModel/ProvTAP-schema-new.xml)

Appendix B   TAP service content exemple. Database dump in VOTable.

The exemple available at URL :
https://wiki.ivoa.net/internal/IVOA/ObservationProvenanceDataModel/CDS-ProvInput-VOT21032018.xml
provides information on data collections used or gathered by CDS in the context of Aladin. It starts from Schmidt
plate collections and goes up to HiPS datasets and collections. These metadata are organized below into a set
of related VOTables consistent with the ProvTAP TAP schema.

Appendix C   Changes from Previous Versions

mireille: changed stopTime everywhere in endTime and ucd "time.stop" in "time.end" to ensure compatibility

with provDM tables.

Bibliography

Demleitner, M., Harrison, P., Molinaro, M., Greene, G., Dower, T. and Perdikeas, M. (2013), 'IVOA registry relational schema', IVOA Working Draft.
http://www.ivoa.net/documents/RegTAP/

Dowler, P., Demleitner, M., Taylor, M. and Tody, D. (2013), 'Data access layer interface, version 1.0', IVOA Recommendation.
http://www.ivoa.net/documents/DALI/20131129/

Dowler, P., Rixon, G. and Tody, D. (2010), 'Table access protocol version 1.0', IVOA Recommendation.
http://www.ivoa.net/documents/TAP

Louys, M., Bonnarel, F., Schade, D., Dowler, P., Micol, A., Durand, D., Tody, D., Michel, L., Salgado, J., Chilingarian, I., Rino, B., de Dios Santander, J. and Skoda, P. (2011), 'Observation data model core components and its implementation in the Table Access Protocol, version 1.0', IVOA Recommendation.
http://www.ivoa.net/documents/ObsCore/20111028/
REC-ObsCore-v1.0-20111028.pdf

Riebe, K., Servillat, M., Bonnarel, F., Louys, M., Rothmaier, F., Sanguillon, M. and the IVOA Data Model Working Group (2017), 'IVOA provenance data model', IVOA Working Draft.
http://www.ivoa.net/documents/ProvenanceDM/