AstroDAS

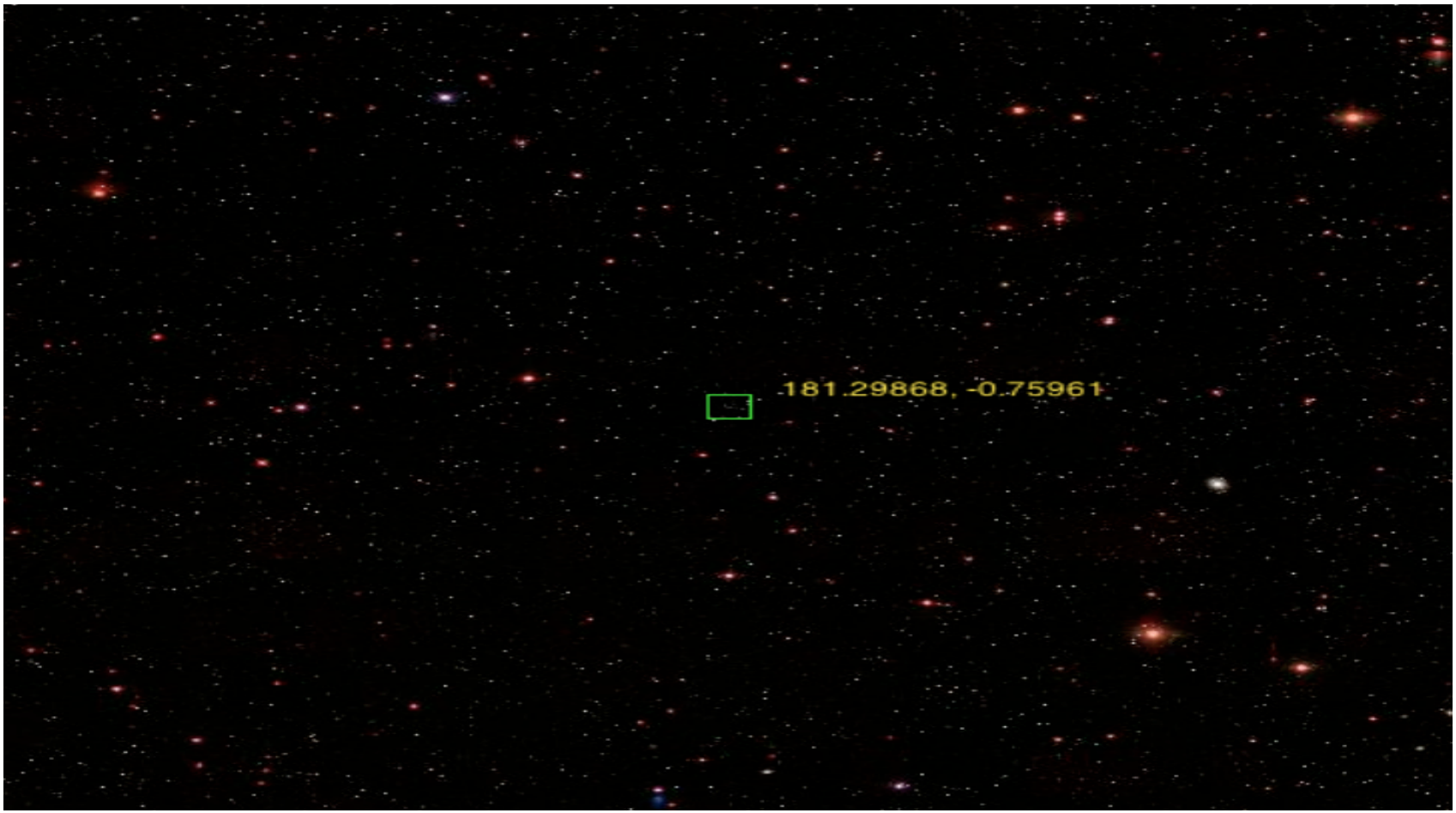
Achieving Database Integration Through Distributed Annotation

Diego Prina-Ricotti
Joint work with Rajendra Bose, Floris Geerts, Anastasios Kementsietsidis and Bob Mann

14 December 2005

Outline

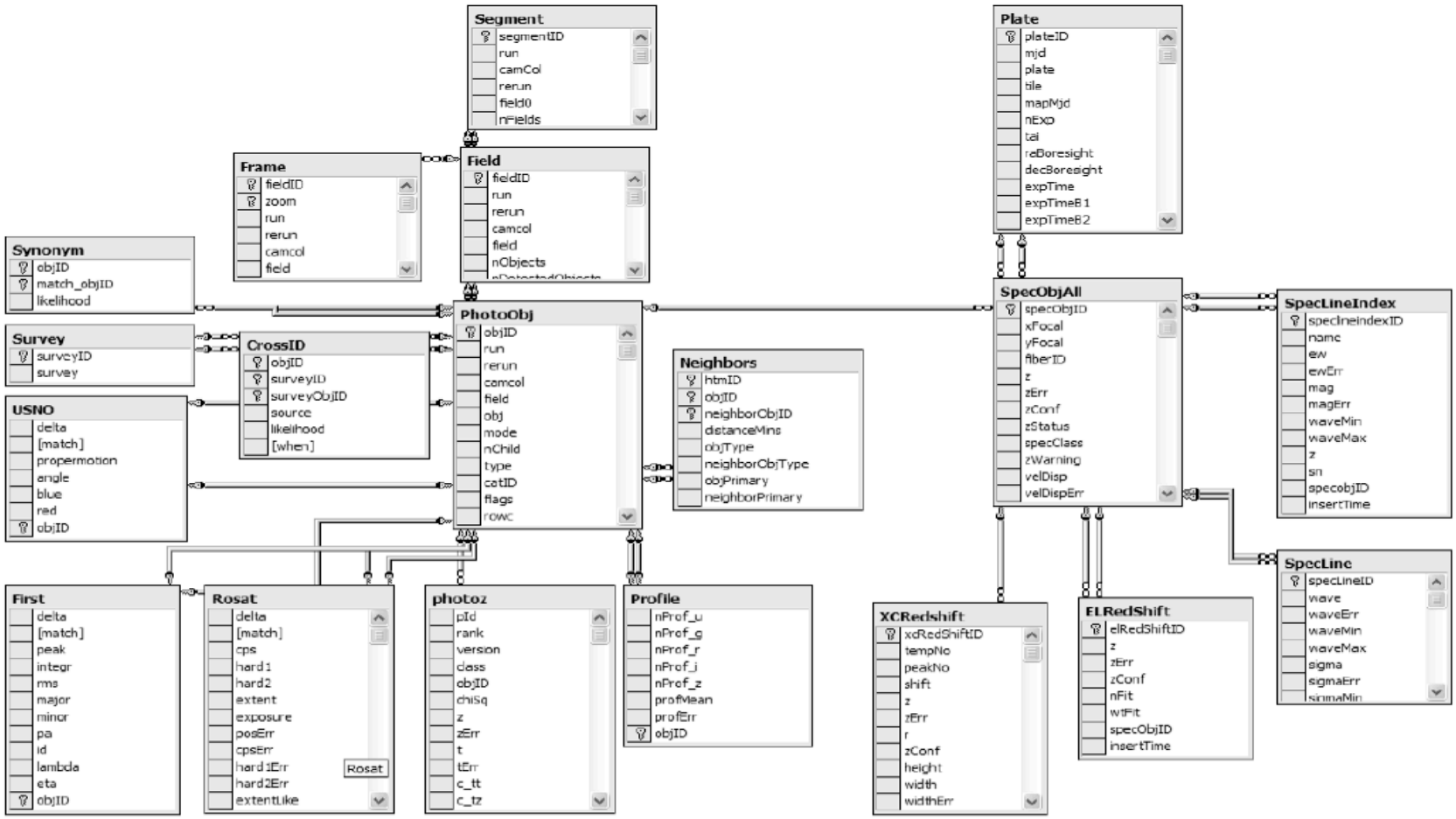
Astronomical Databases



Astronomical Databases

- E.g., Sloan Digital Sky Server (SDSS):
  - 5 year survey of the northern sky
  - around 100 millions objects with about 400 attributes
  - size of the first data release is about 1TB
  - snowflake schema with the PhotoObj relation in the centre
  - views are used to subset the objects in primaries, secondaries, stars, galaxies, etc.
- Astronomical databases store data about observations in a variety of wavelengths – optical, radio, infrared, γ-rays, X-rays and more.
- Each wavelength can provide different information about a celestial event or object.

Astronomical Databases



Data Integration

- Data integration is needed to join measurements on the same entities that span across multiple astronomical databases.
- This is a problem of "mediation across multiple worlds" since the different databases capture non-overlapping aspect of the same entities.
- It is crucial to be able to **entity join** relations residing in different databases, i.e., to associate the tuples of the input relations that represent the same real-world object.
- **Entity identification** is the problem of identifying objects from different databases which correspond to the same real-world entity.

Outline

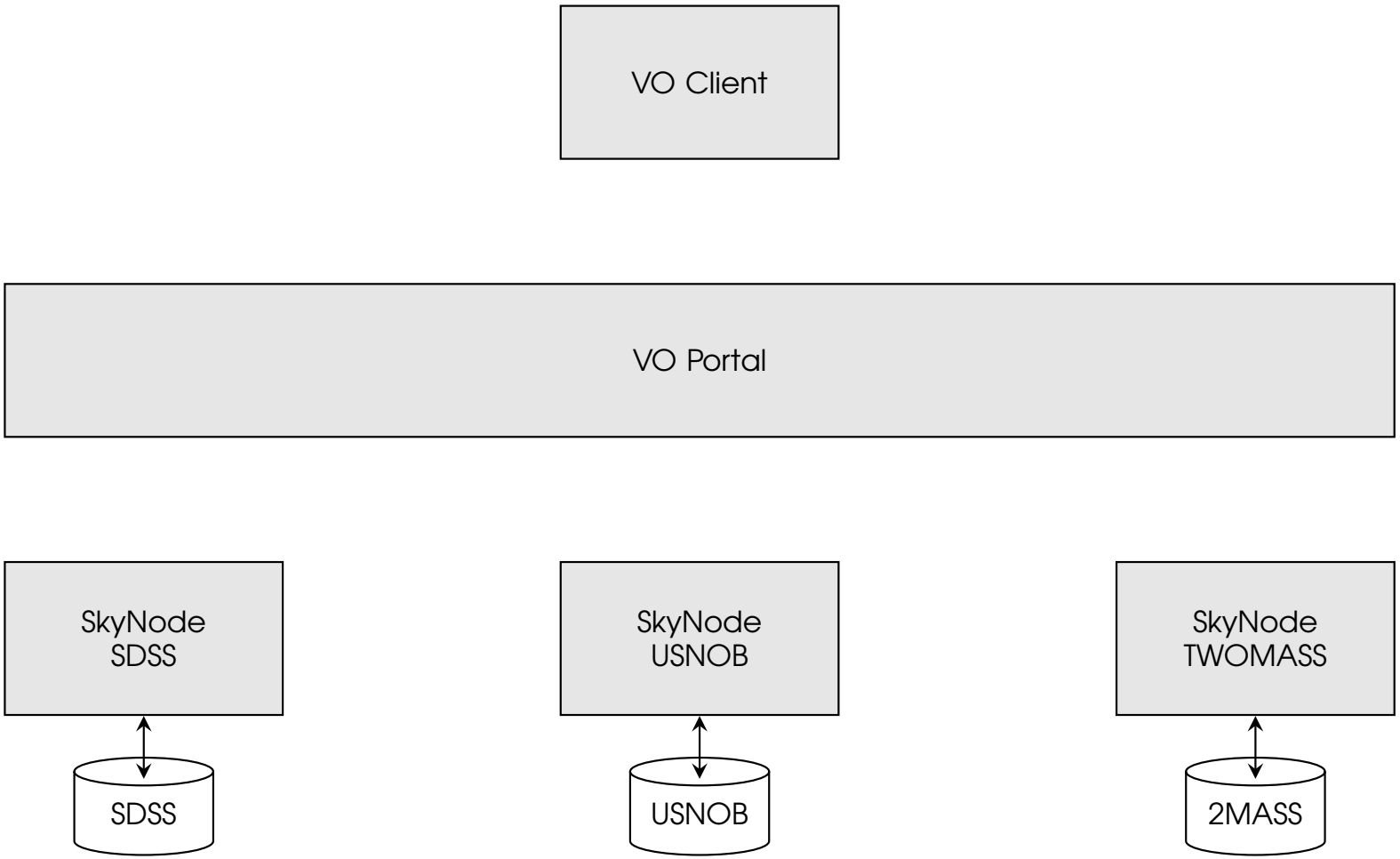
International Virtual Observatory Alliance (IVOA)

- Consortium that produces standards on the model of the W3C process involving working drafts.
- Formed on June 2002 has grown to include 15 funded VO projects internationally.
- The aim is to combine existing databases from ground-based and orbiting observatories and make them easily accessible to researchers.

Virtual Observatory

- The **Virtual Observatory** (VO) is an astronomical database federation.
- The architecture is based on the wrapper-mediator pattern and consists of three components:
  - the Clients
  - the VO Portal (mediator)
  - the SkyNodes (wrappers).
- Single databases, called *SkyNodes*, implement a standard Web Services interface:
  - querying metadata
  - executing single node queries
  - executing performance queries
  - handling execution plans
  - executing cross matching (XMatch).

Virtual Observatory
Architecture



ADQL

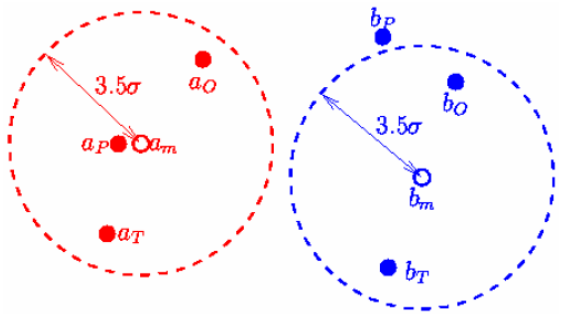
- The VOQL working group is responsible for **ADQL** (Astronomical Data Query Language).
- It is based on a subset of SQL92 with two extra functions *XMatch* and *Region*.
- Two formats: ADQL/s (sql-like) and ADQL/x (xml language).
- Current implementation at: `http://openskyquery.net`.

ADQL
Example

```
SELECT s.objId, s.ra, s.dec, s.r, s.type,
       t.objId, t.ra, t.dec
FROM sdss:photoprimary s,
     twomass:photoprimary t
WHERE s.type = 3 AND
      Region('CIRCLE J2000 181.3 -0.76 6.5') AND
      XMATCH(s,t) < 3.5;
```

ADQL
Region And XMatch Clauses

- The Region clause lets the user specify the spatial extent of the search, i.e. the astronomical coordinates (right ascension and declination) of the center and the radius (in minutes of arc) of a circle for a "cone search".
- Syntax: Region('CircleJ2000 *ra dec radius*')
- The position of objects in the sky may vary from database to database due to measurement errors.
- The XMatch clause is used to specify a degree of tolerance for the probabilistic spatial join.



- Syntax: XMATCH(*skynode-list*) < *tolerance*

Virtual Observatory Limits

- The query system relies on a probabilistic spatial join to perform *entity join*.
- Astronomers have developed more complex algorithms for solving *entity identification* but those algorithms are too computationally involved to be embedded in the ADQL processing.
- The user might want the ability to consult the result of one of those algorithms or the opinion of an expert.

Outline

Annotation

- AstroDAS doesn't provide an algorithm for entity identification, it provides a way to store the result of entity identification in mapping annotations.
- Annotations are *superimposed information* – we are placing information, the mapping annotations, on the top of an existing structure, the skynodes.

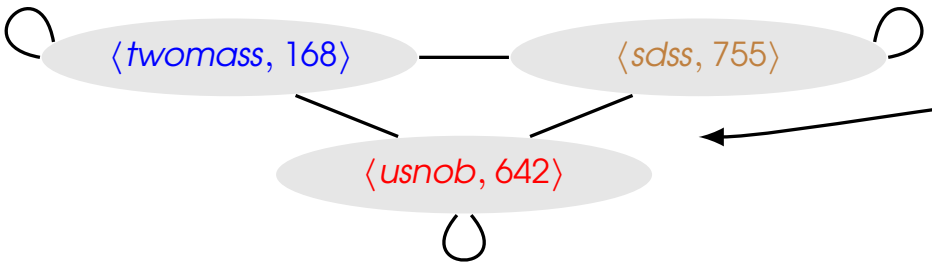
Mapping Annotation

- A **database object** is an ordered pair $\langle db, id \rangle$.
  E.g.: $\langle sdss, 582034940614755 \rangle$
- A **mapping**, denoted $\leftrightarrow$, is an equivalence relation over the set of database objects. An ordered pair of database objects is part of the relation iff the two objects correspond to the same real-world entity.
  E.g.: $\langle sdss, 582034940614755 \rangle \leftrightarrow \langle twomass, 238813168 \rangle$
- A **mapping annotation** is an annotation that identifies a set of database objects from different data sources that are in the same equivalence class of the mapping relation.
- Mapping annotation are external and don't affect in any way the databases that they annotate.

Mapping Tables

- Many techniques make use of mapping tables to store the result of entity identification.
- A **mapping table** is a relation where the tuples are composed of database objects that are in the same equivalence class of the mapping relation.

twomass	usnob	sdss
168	642	755
169	null	585
183	636	679
183	636	680
197	516	null



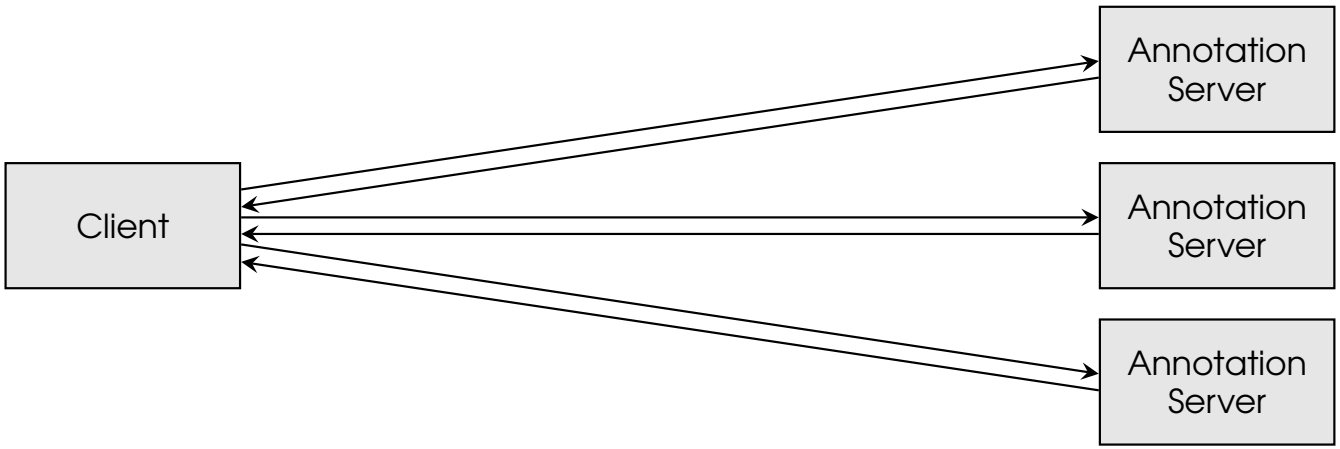
$\langle twomass, 168\rangle \leftrightarrow \langle twomass, 168\rangle$
$\langle twomass, 168\rangle \leftrightarrow \langle usnob, 642\rangle$
$\langle twomass, 168\rangle \leftrightarrow \langle sdss, 755\rangle$
$\langle usnob, 642\rangle \leftrightarrow \langle usnob, 642\rangle$
$\langle usnob, 642\rangle \leftrightarrow \langle twomass, 168\rangle$
$\langle usnob, 642\rangle \leftrightarrow \langle sdss, 755\rangle$
$\langle sdss, 755\rangle \leftrightarrow \langle sdss, 755\rangle$
$\langle sdss, 755\rangle \leftrightarrow \langle twomass, 168\rangle$
$\langle sdss, 755\rangle \leftrightarrow \langle usnob, 642\rangle$

Distributed Annotation

- The task of annotating the mappings between astronomical catalogs should be distributed across the whole astronomy community rather than being carried out by a central authority:
  - astronomers and algorithms often disagree on object mappings so we should provide the opinion of everyone
  - efficiency.
- The distribution of annotation is also evident in other scientific communities such as Biology.

BioDAS

- Enables third parties to store and manage their annotations without requiring a centralized coordination.
- Allows the user to query and integrate information from different annotation data sources.
- BioDAS is an ancestor of a web service.



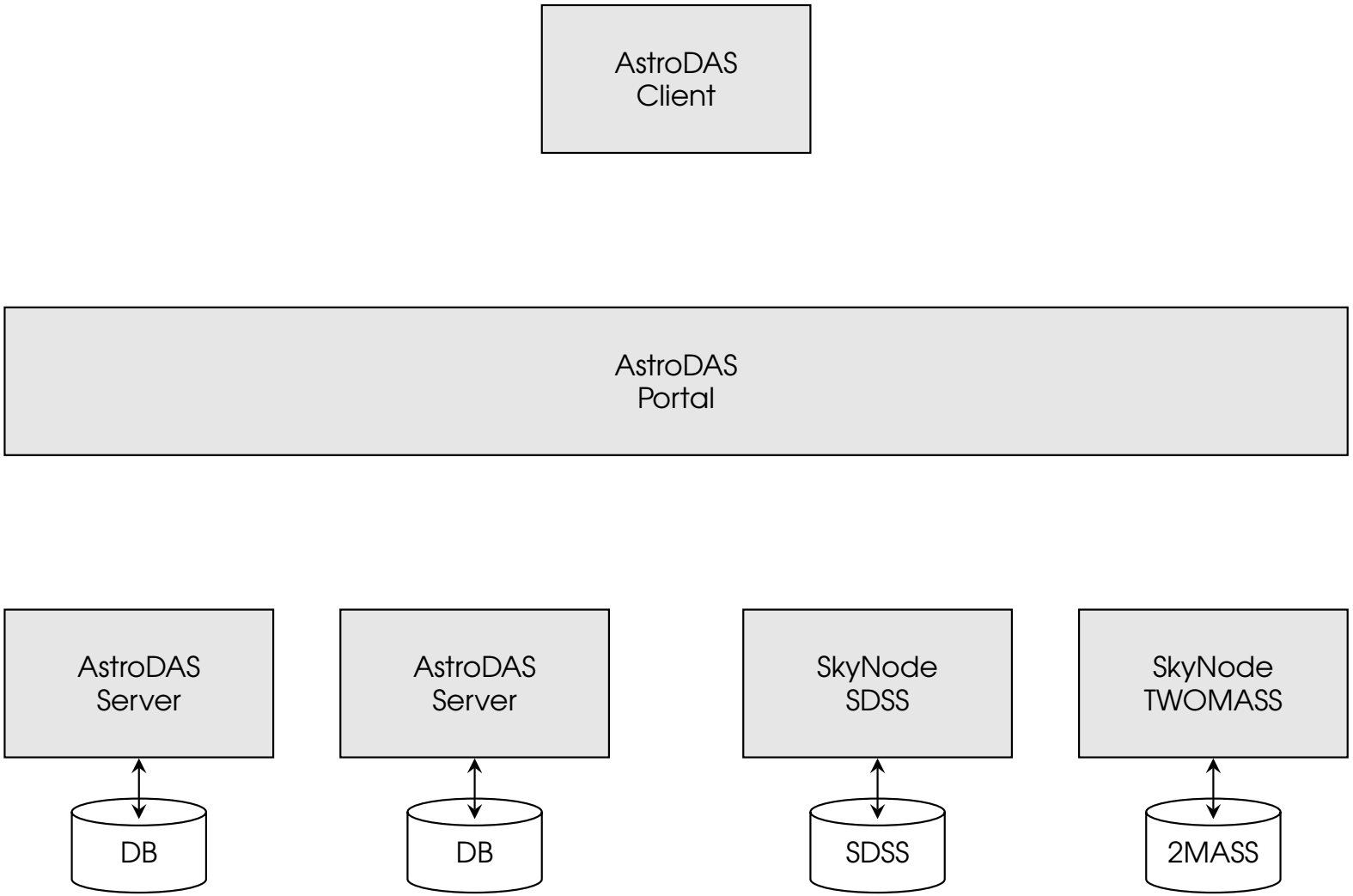
AstroDAS

- AstroDAS is a web service evolution of BioDAS that allows for the sharing of annotation, in particular, mapping annotations.
- In order to store annotations a client calls the proper method of the AstroDAS server web service interface.
- An AstroDAS server web service furnishes methods to retrieve mapping annotation in xml format or in the form of a mapping table.
- Why web services?
  - compatibility with the Virtual Observatory infrastructure
  - interoperability between different platforms
  - ease of client implementation.

Query System

- AstroDAS follows the wrapper/mediator architecture where a client connects to a portal (mediator) that interacts with several web service interfaces (wrappers) to the underlying databases of annotations.
- **DSQL** (Distributed/Diego SQL):
  - language for the manipulation of data from different relational databases
  - syntax is as close as possible to ADQL
  - similar in purpose to MSQL (Multidabase SQL) but much more limited in extent (W. Litwin et al. - MSQL: a multidatabase language)
  - provides an operator to entity join two relations from different databases on their key attributes.

Architecture



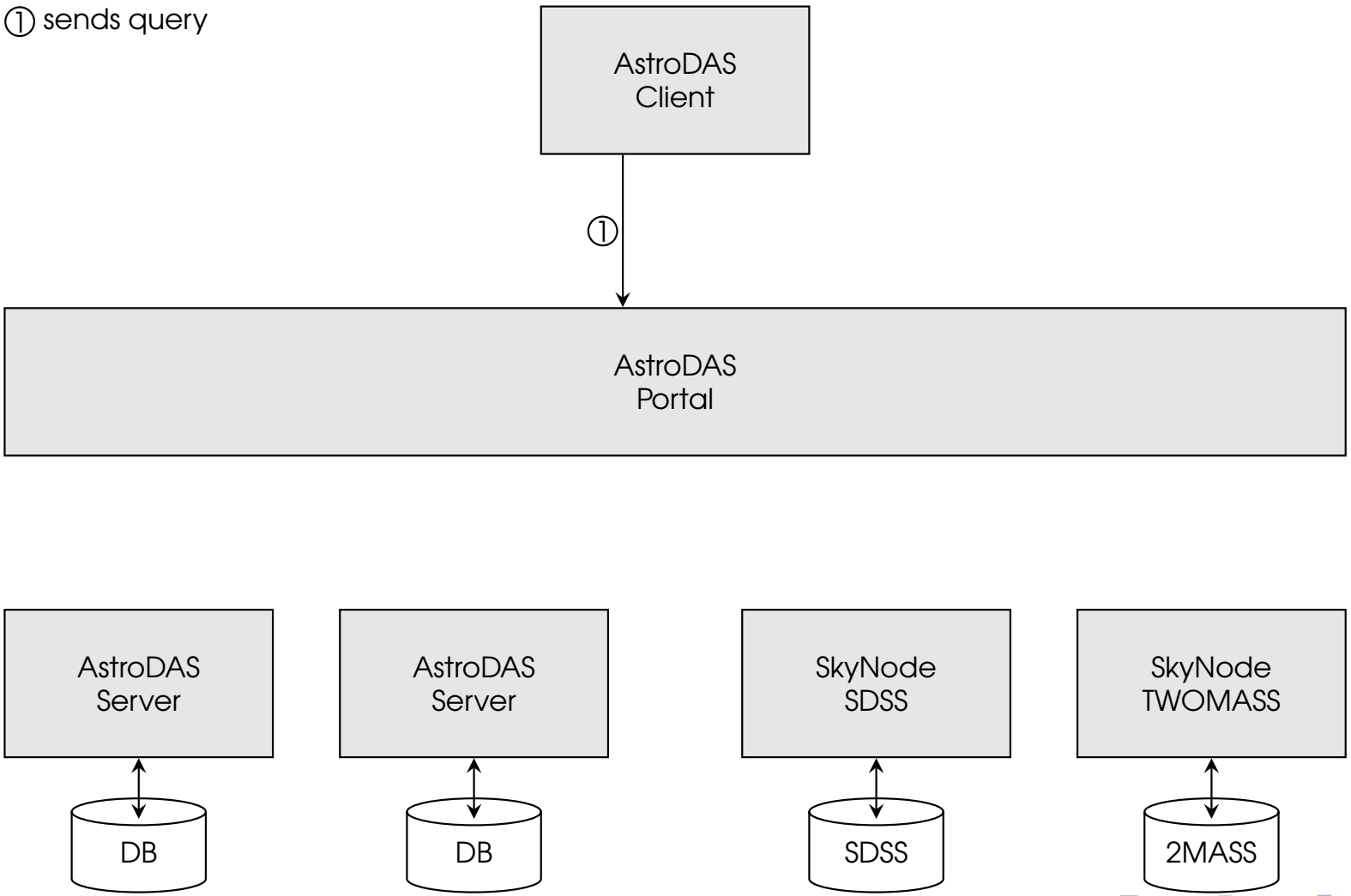
DSQL Query Example

```
SELECT
   s.objid, s.ra, s.dec, s.type,
   t.objid, t.ra, t.dec
FROM
   SDSS:photoprimary s, TWOMASS:photoprimary t,
   AS:EdinburghUniversity e, AS:RomeUniversity r
WHERE
   Region('CircleJ2000 200 -1 0.02') AND
   s.type=3 AND
   e.author='algorithm1' AND
   r.author='algorithm2';
```

Query Processing

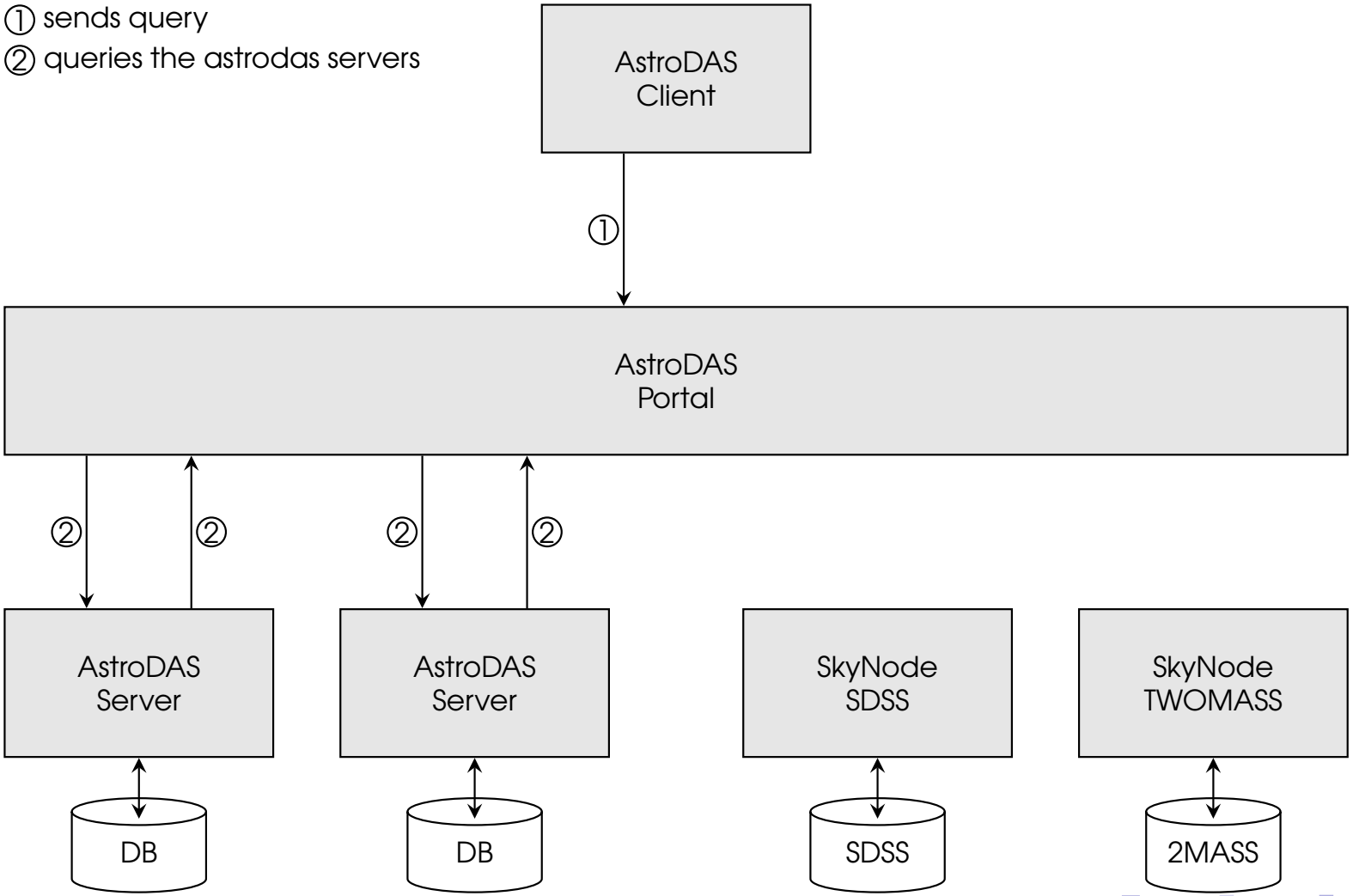
1. The AstroDAS Client sends a DSQL query to the AstroDAS Portal.
2. The AstroDAS Portal sends a SQL query to each AstroDAS Server specified in the from clause.
3. The AstroDAS Portal collects the mapping annotations and uses them to create a mapping table.
4. Using the mapping table the AstroDAS Portal sends an ADQL query to each SkyNode specified in the from clause.
5. Using the mapping table the AstroDAS Portal joins the results of each ADQL query and returns it.

Query Processing



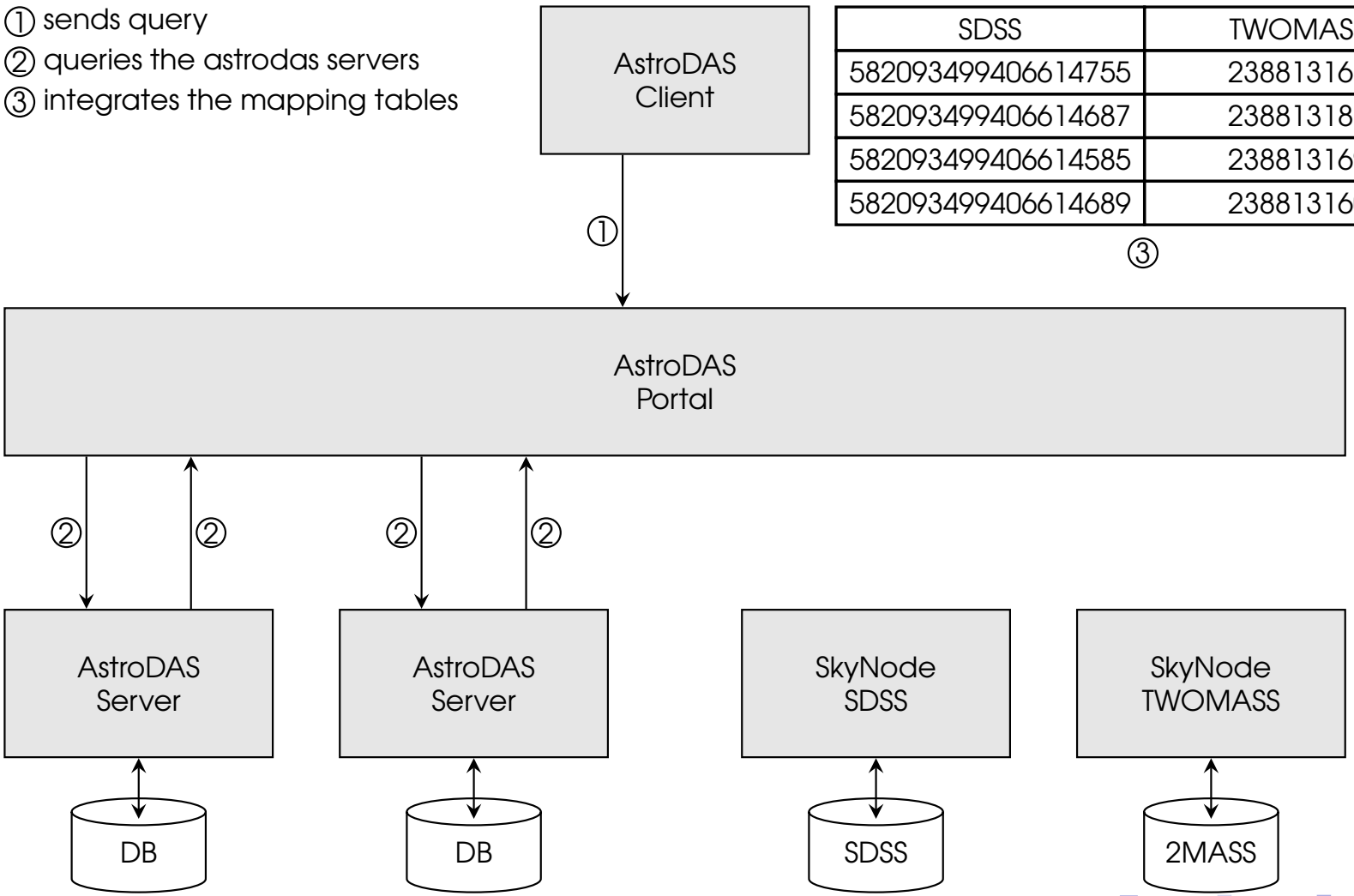
Query Processing

① sends query
② queries the astrodas servers



Query Processing

① sends query
② queries the astrodas servers
③ integrates the mapping tables

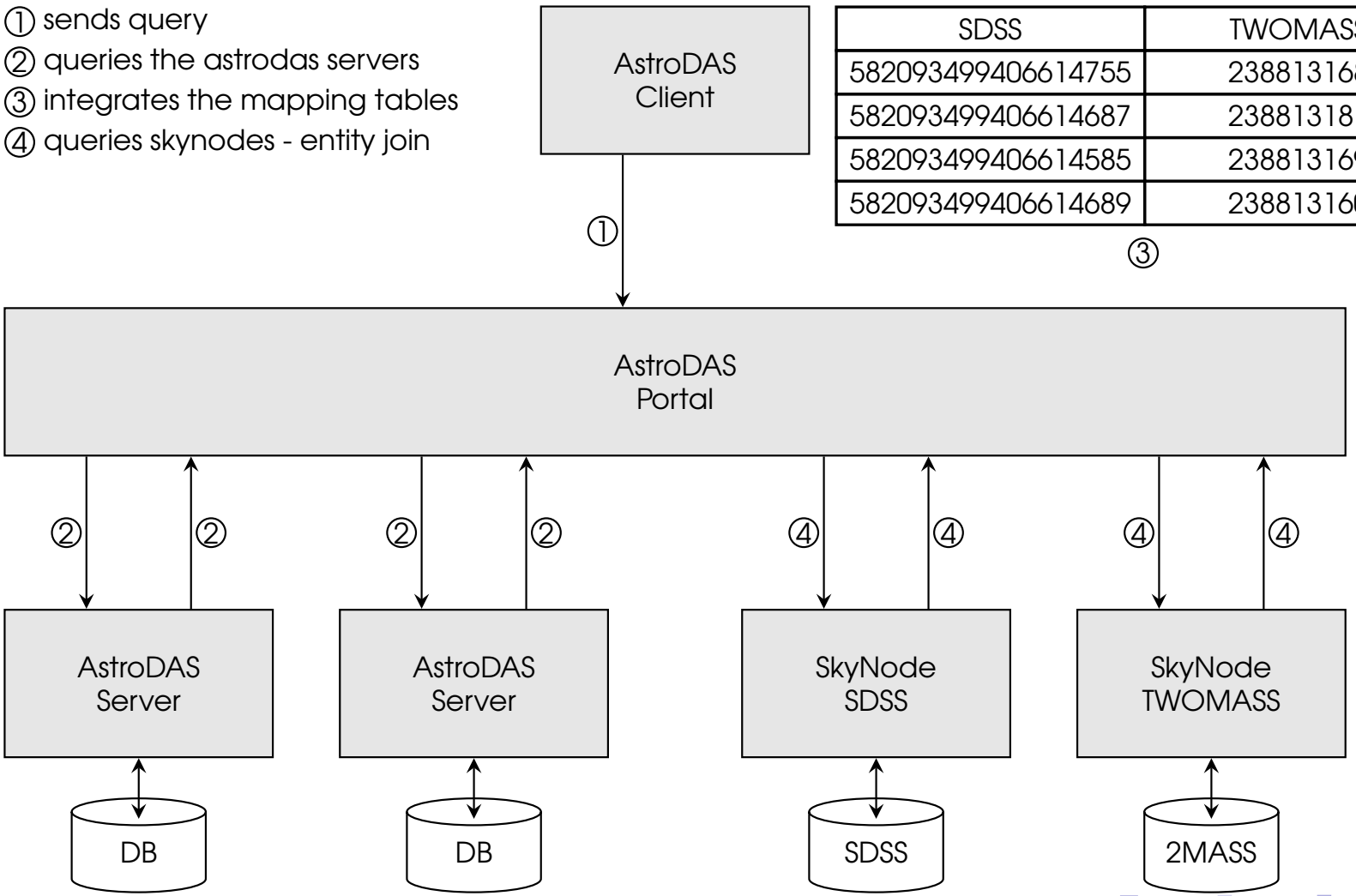


SDSS	TWOMASS
582093499406614755	238813168
582093499406614687	238813181
582093499406614585	238813169
582093499406614689	238813160

③

Query Processing

① sends query
② queries the astrodas servers
③ integrates the mapping tables
④ queries skynodes - entity join

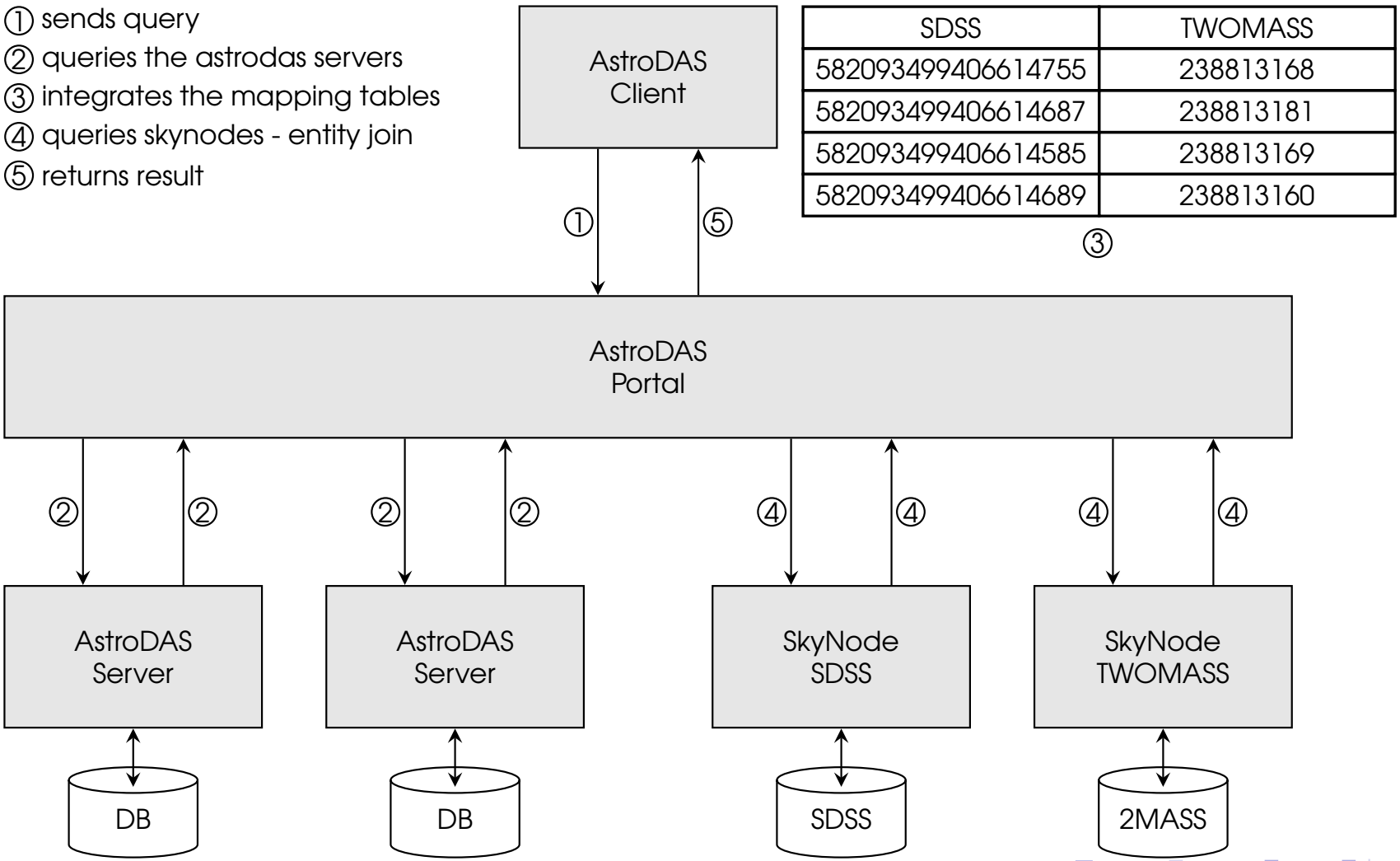


SDSS	TWOMASS
582093499406614755	238813168
582093499406614687	238813181
582093499406614585	238813169
582093499406614689	238813160

③

Query Processing

① sends query
② queries the astrodas servers
③ integrates the mapping tables
④ queries skynodes - entity join
⑤ returns result



SDSS	TWOMASS
582093499406614755	238813168
582093499406614687	238813181
582093499406614585	238813169
582093499406614689	238813160

③

Example

```
SELECT
  t.objid, t.ra, t.dec, u.objid, u.ra, u.dec,
  s.objid, s.ra, s.dec, s.type
FROM
  TWOMASS:photoprimary t, USNOB:photoprimary u,
  SDSS:photoprimary s, AS:EdinburghUniversity e
WHERE
  Region('CircleJ2000 200 -1 0.05') AND
  e.author='XMatch' AND
  FULL OUTER JOIN;
```

Example

```
SELECT
  t.objid, t.ra, t.dec, u.objid, u.ra, u.dec,
  s.objid, s.ra, s.dec, s.type
FROM
  TWOMASS:photoprimary t, USNOB:photoprimary u,
  SDSS:photoprimary s, AS:EdinburghUniversity e
WHERE
  Region('CircleJ2000 200 -1 0.05') AND
  e.author='XMatch' AND
  FULL OUTER JOIN;
```

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null

Example

```
SELECT
  t.objid, t.ra, t.dec, u.objid, u.ra, u.dec,
  s.objid, s.ra, s.dec, s.type
FROM
  TWOMASS:photoprimary t, USNOB:photoprimary u,
  SDSS:photoprimary s, AS:EdinburghUniversity e
WHERE
  Region('CircleJ2000 200 -1 0.05') AND
  e.author='XMatch' AND
  FULL OUTER JOIN;
```

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null

twomass	usnob	sdss
183	636	679
183	636	680
197	516	null

Outline

Inference Algorithm

1. The graph G_{in} corresponding to the input mapping table MT_{in} is computed.
2. The set of connected components CC of G_{in} is computed.
3. $\forall cc \in CC$, let DB be the set of attributes of MT_{in}, $\forall db \in DB$ the set of database objects $\in db$ is calculated and the cartesian product of those sets is computed.
4. The output mapping table MT_{out} is the union of the cartesian products for each connected component.

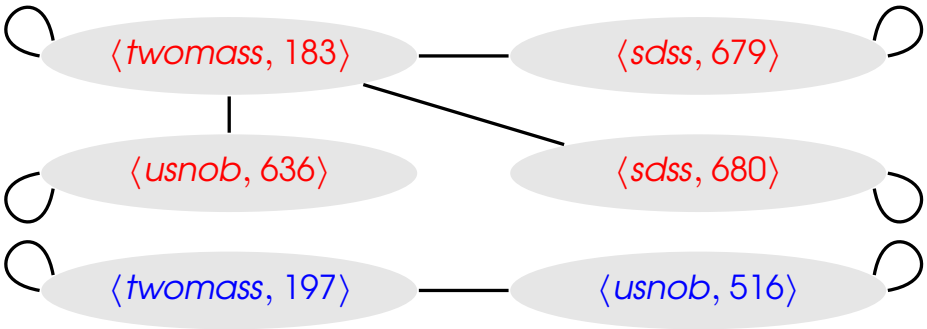
Property: The graph G_{out} corresponding to MT_{out} is the transitive closure of G_{in}.

Inference Algorithm

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null

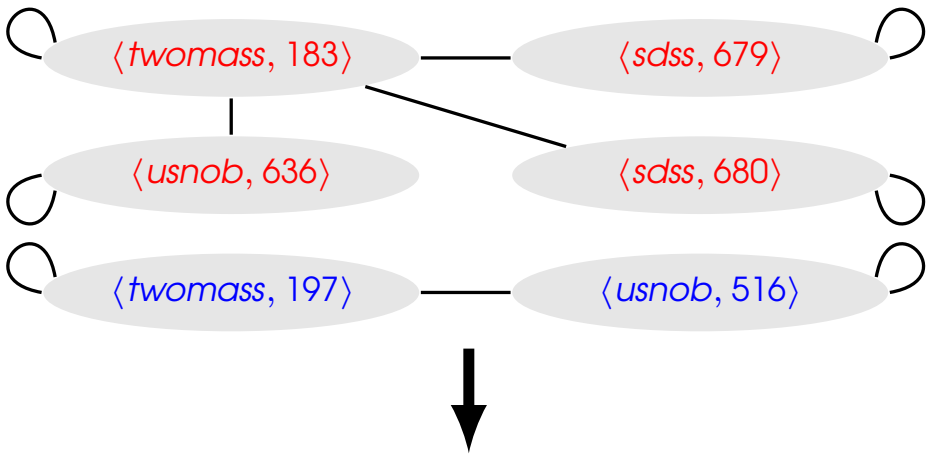
Inference Algorithm

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null



Inference Algorithm

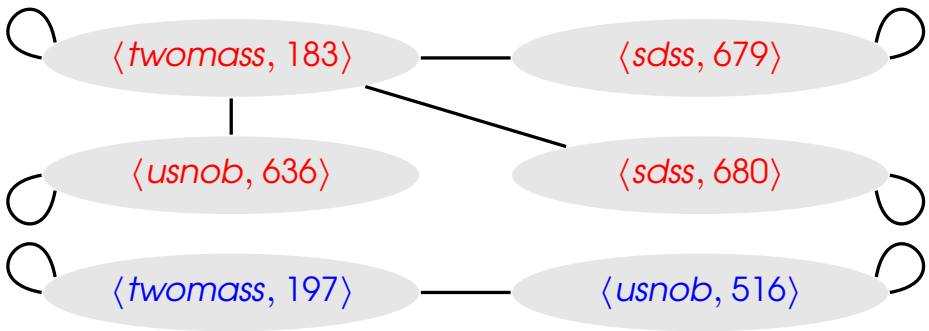
twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null



$$\{(\langle twomass, 183\rangle, \langle sdss, 679\rangle, \langle usnob, 636\rangle, \langle sdss, 680\rangle),$$
$$(\langle twomass, 197\rangle, \langle usnob, 516\rangle)\}$$

Inference Algorithm

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null

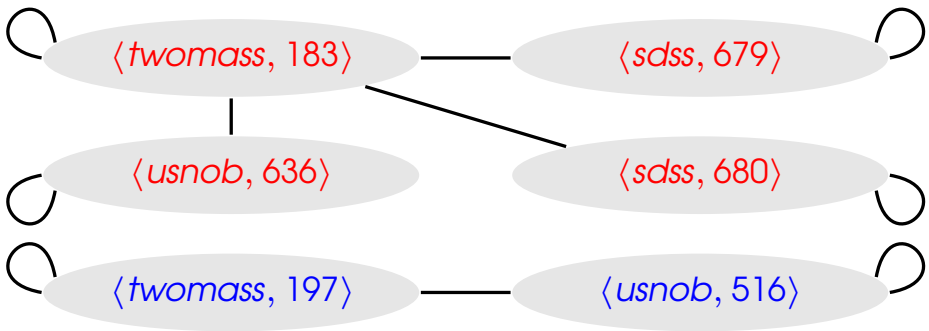


$$\{(\langle twomass, 183\rangle, \langle sdss, 679\rangle, \langle usnob, 636\rangle, \langle sdss, 680\rangle), (\langle twomass, 197\rangle, \langle usnob, 516\rangle)\}$$

$$\{\langle twomass, 183\rangle\} \times \{\langle usnob, 636\rangle\} \times \{\langle sdss, 679\rangle, \langle sdss, 680\rangle\}$$
$$\{\langle twomass, 197\rangle\} \times \{\langle usnob, 516\rangle\} \times \{null\}$$

Inference Algorithm

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null



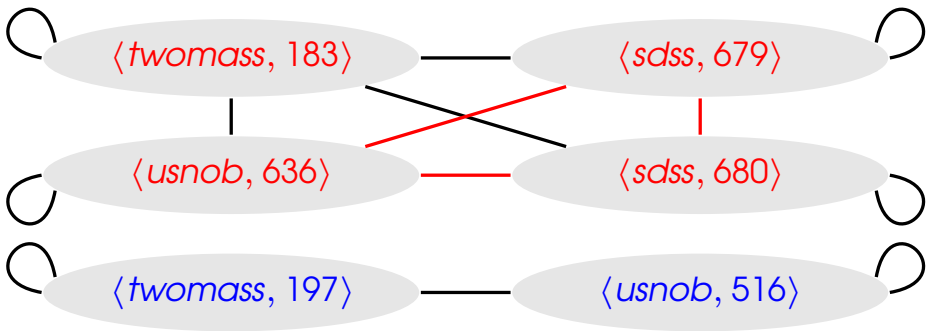
$$\{(\langle twomass, 183\rangle, \langle sdss, 679\rangle, \langle usnob, 636\rangle, \langle sdss, 680\rangle),$$
$$(\langle twomass, 197\rangle, \langle usnob, 516\rangle)\}$$

$$\{\langle twomass, 183\rangle\} \times \{\langle usnob, 636\rangle\} \times \{\langle sdss, 679\rangle, \langle sdss, 680\rangle\}$$
$$\{\langle twomass, 197\rangle\} \times \{\langle usnob, 516\rangle\} \times \{null\}$$

twomass	usnob	sdss
183	636	679
183	636	680
197	516	null

Inference Algorithm

twomass	usnob	sdss
183	null	679
183	null	680
183	636	null
197	516	null



$$\{(\langle twomass, 183\rangle, \langle sdss, 679\rangle, \langle usnob, 636\rangle, \langle sdss, 680\rangle), (\langle twomass, 197\rangle, \langle usnob, 516\rangle)\}$$

$$\{\langle twomass, 183\rangle\} \times \{\langle usnob, 636\rangle\} \times \{\langle sdss, 679\rangle, \langle sdss, 680\rangle\}$$

$$\{\langle twomass, 197\rangle\} \times \{\langle usnob, 516\rangle\} \times \{null\}$$

twomass	usnob	sdss
183	636	679
183	636	680
197	516	null

Scenario

S_1

sdss	twomass	first	usnob
179	null	355	null
null	233	null	700

S_2

first	usnob	rosat
355	null	193
null	700	193

S_3

sdss	nvss
178	423
179	423

Scenario

S_1

sdss	twomass	first	usnob
179	null	355	null
null	233	null	700

S_2

first	usnob	rosat
355	null	193
null	700	193

S_3

sdss	nvss
178	423
179	423

sdss	twomass	first	usnob	rosat	nvss
179	null	355	null	null	null
null	233	null	700	null	null
null	null	355	null	193	null
null	null	null	700	193	null
178	null	null	null	null	423
179	null	null	null	null	423

Scenario

S_1

sdss	twomass	first	usnob
179	null	355	null
null	233	null	700

S_2

first	usnob	rosat
355	null	193
null	700	193

S_3

sdss	nvss
178	423
179	423

sdss	twomass	first	usnob	rosat	nvss
179	null	355	null	null	null
null	233	null	700	null	null
null	null	355	null	193	null
null	null	null	700	193	null
178	null	null	null	null	423
179	null	null	null	null	423

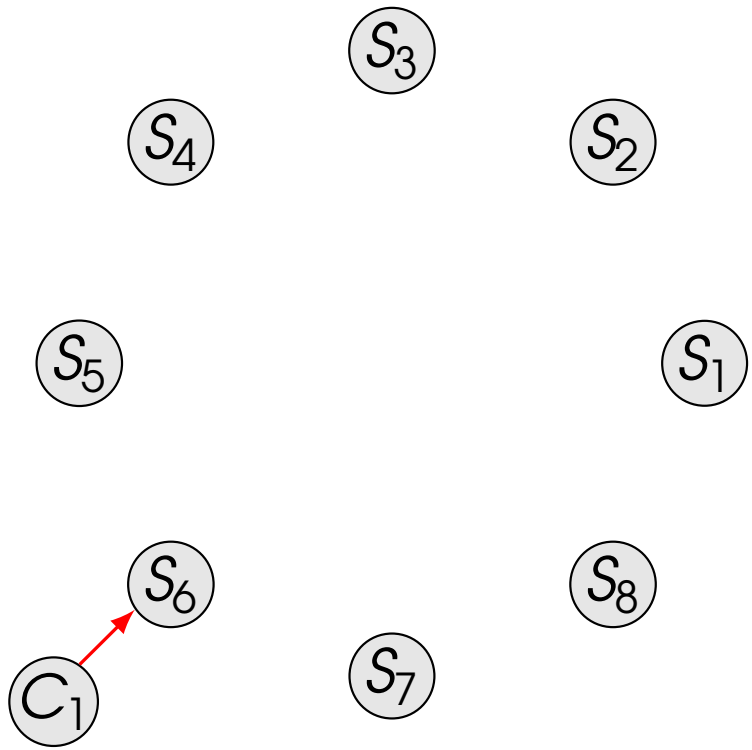
sdss	twomass	first	usnob	rosat	nvss
178	233	355	700	193	423
179	233	355	700	193	423

Peer-to-Peer Architecture

- A peer-to-peer network of databases represents a more scalable architecture for a distributed annotation system. We would like to achieve the following goals:
  - no single point of failure (the mediator)
  - load balanced
  - no need to consult a UDDI registry to discover annotation servers
  - the network can route the queries only to the relevant nodes.
- AstroDAS implementation follows a very simple peer-to-peer architecture where clients can query any peer of the network for a mapping table.

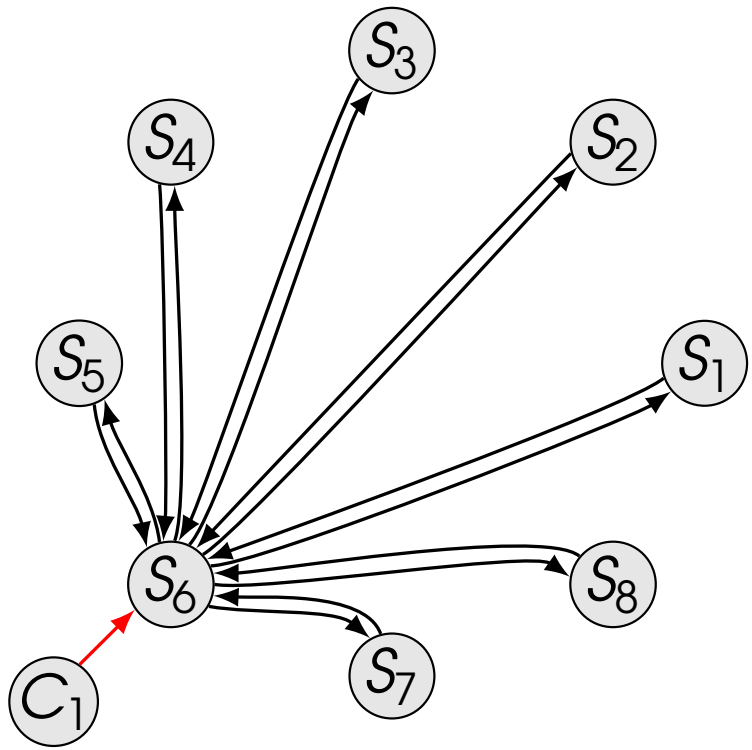
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



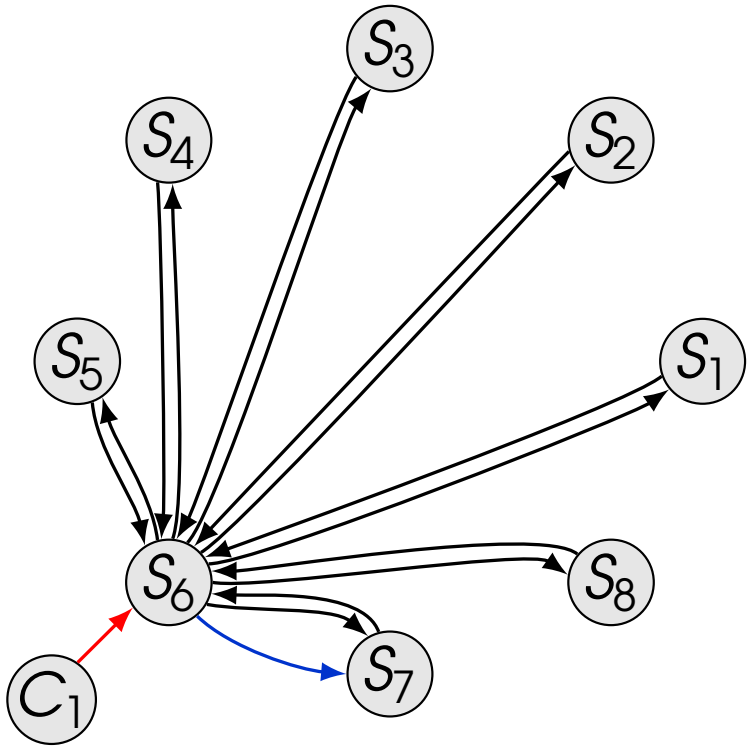
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



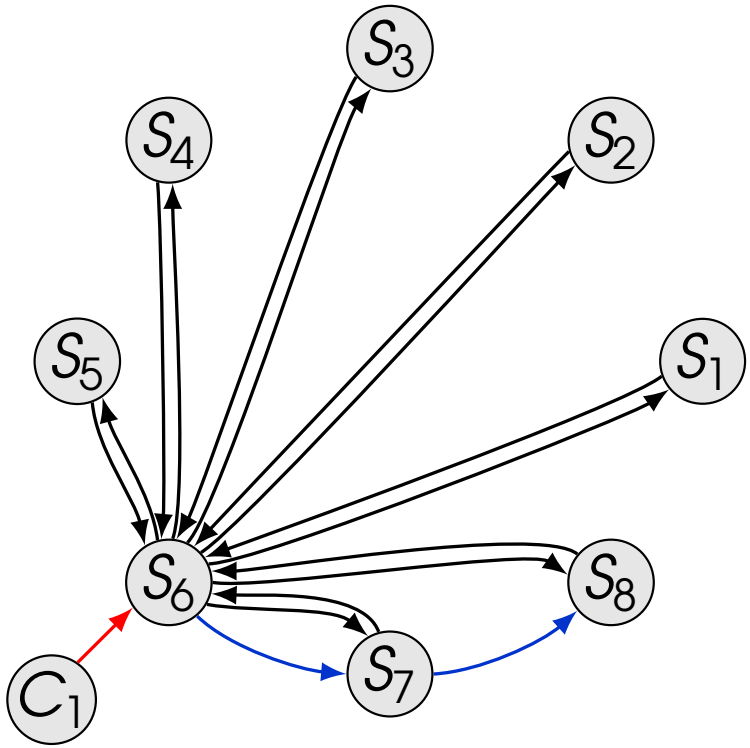
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



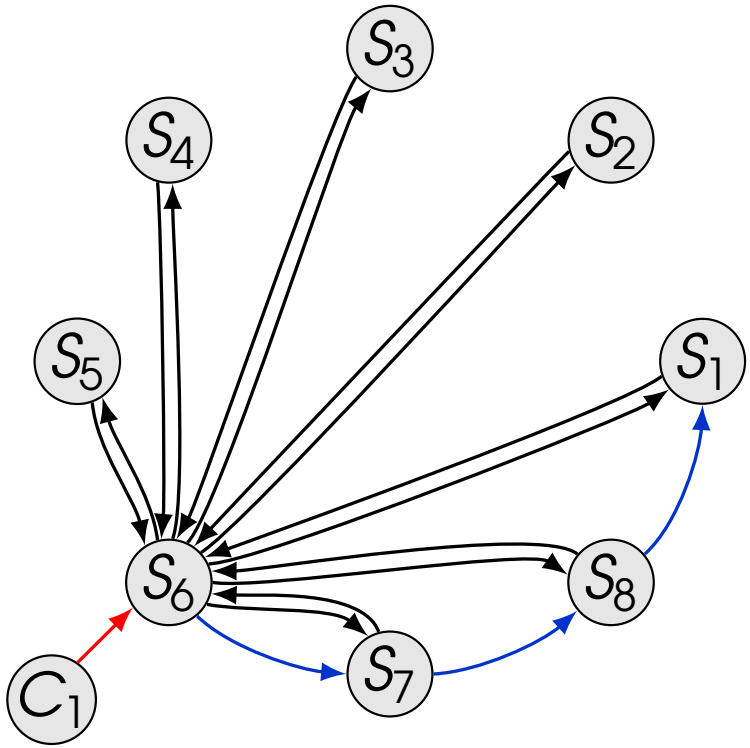
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



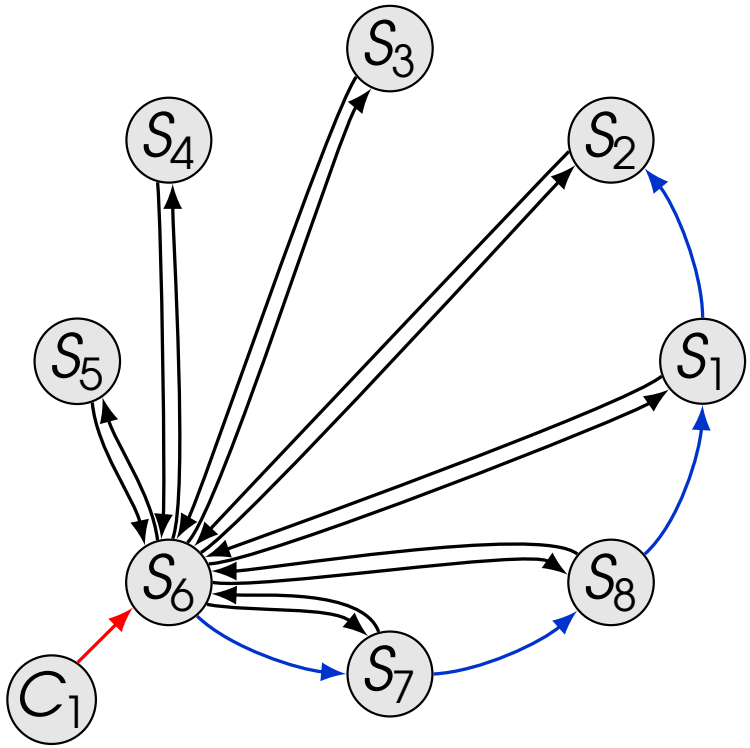
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



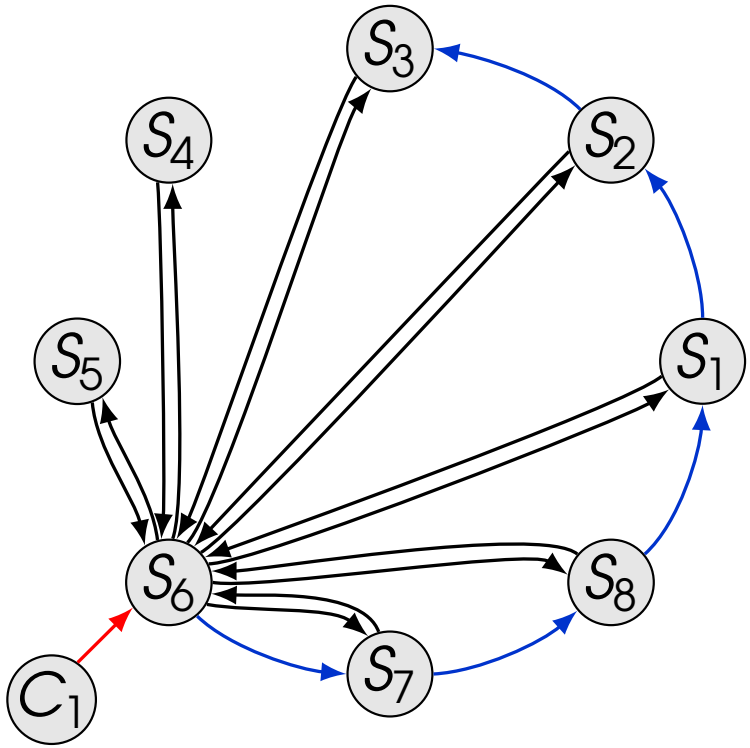
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



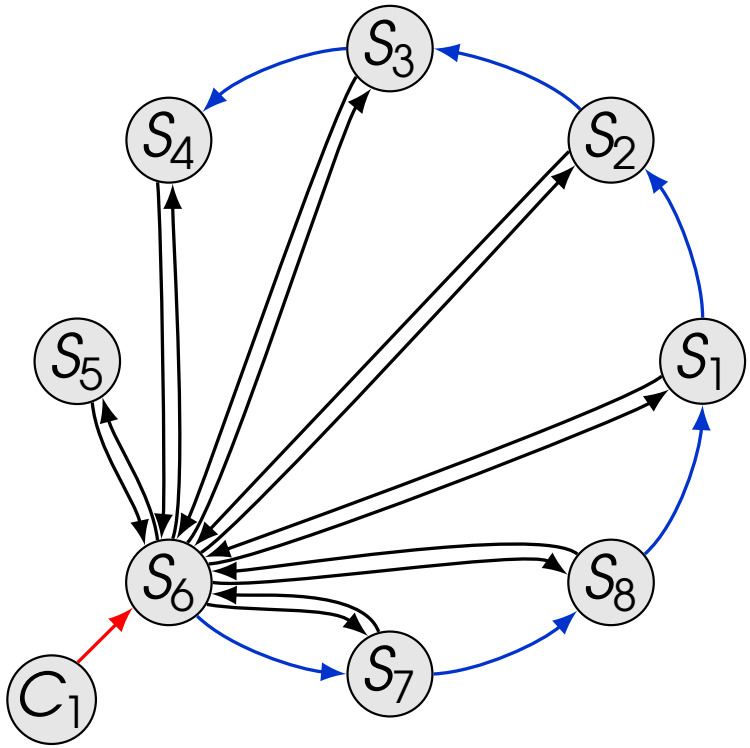
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



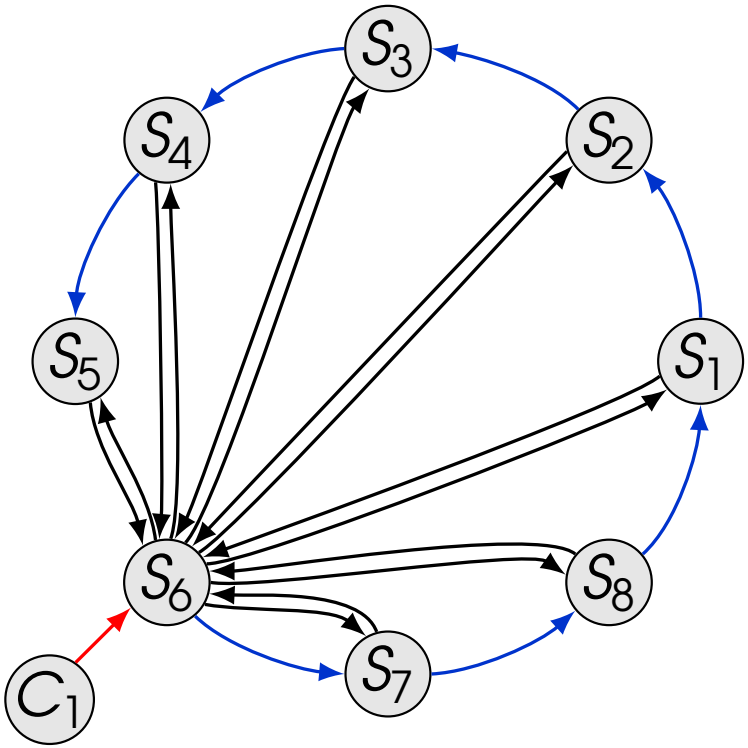
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



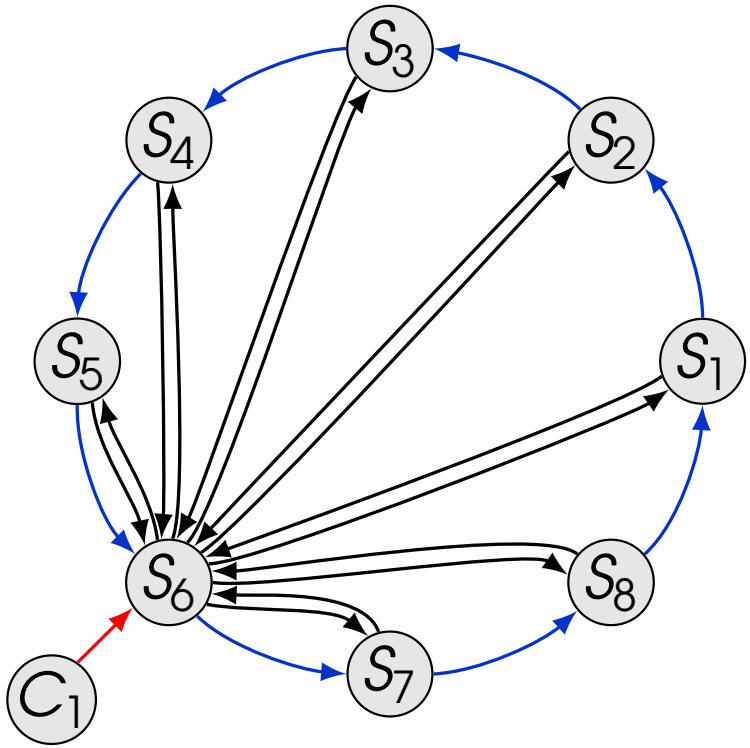
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



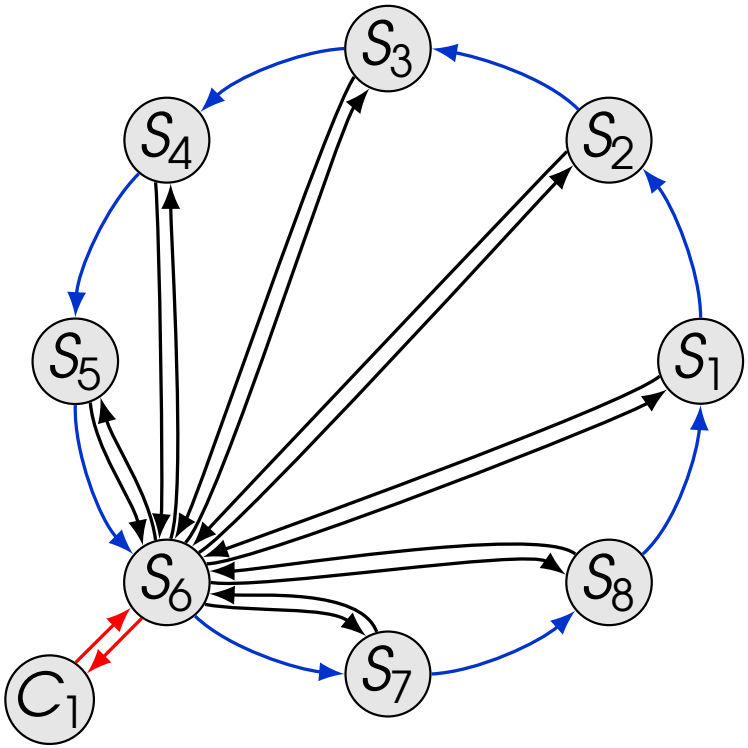
Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



Distributed Inference Algorithm

1. The clients sends a query for the global mapping table to one node of the p2p network, that we call coordinator.
2. The coordinator sends in parallel requests, to the nodes concerned by the query, for computing the connected components of the graph corresponding to the local mapping table.
3. The coordinator determines an execution plan for merging the connected components and executes it.
4. The coordinator returns to the client the inferred global mapping table corresponding to the connected components of the graph.



Outline

Summary

- The problem of data integration in astronomical databases.
  - The current solution based on a spatial entity join.
  - Our solution with the use of mapping tables.
- The problem of sharing annotations.

Acknowledgements

Paolo Atzeni,
Paolo Besana,
Rajendra Bose,
Peter Buneman,
Floris Geerts,
Anastasios Kementsietsidis,
Bob Mann,
Chris Rusbridge,
Joseph Spadavecchia,
Bob's Mann Group,
Database Group

Grazie!