# TAP services integration at IA2 data center

#### Pietro Apollo INAF - Astronomical Observatory of Trieste









- IA2TAP:
  - TAP implementation
  - 2+1+1 services
- TapSchemaManager:
  - a supporting web application for managing TAP\_SCHEMA schemata
- Summary



# IA2TAP: general overview

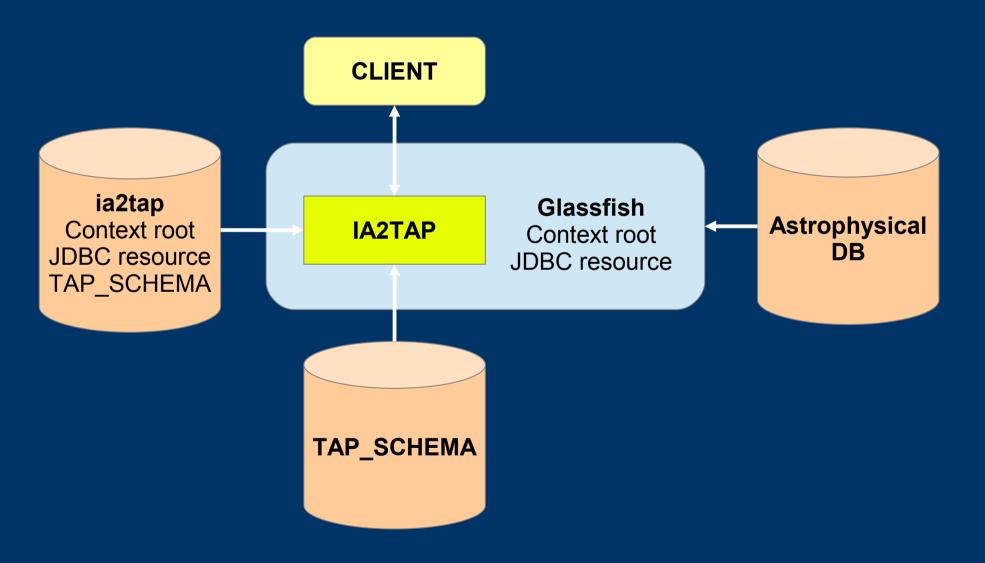


- IA2TAP is a web application developed at the INAF OATs IA2:
  - Coded in Java (JDK 1.6)
  - Runs on Glassfish web container (tested on version 3.1)
  - Uses a MySQL DBMS (tested on version 5.5):
    - to store its internal configuration such as service descriptions, job details, logging
    - to connect to the astrophysical archives
  - It has been developed on top of the openCADC libraries, to fulfill the main TAP requisites



#### IA2TAP block diagram







#### Dynamic endpoints



- IA2TAP implementation builds the endpoints (capabilities, availability) on the base of the configuration, instead of using XML configuration files
- The tables endpoint is dynamically built on the base of the metadata exposed by TAP\_SCHEMA







- The openCADC libraries have been customized to allow multiple TAP services running side by side on the same web server
- Multiple deployments of a unique WAR package are used to expose on the Glassfish web container different TAP services





# IA2TAP: multiple JDBC resources

- Customised implementation of openCADC libraries:
  - every TAP service:
    - extracts its own astrophysical data from the schemata available through a dedicated JDBC connection (provided by Glassfish)
    - loads the dedicated JDBC resource (connection pool) from the Glassfish web container as specified by *ia2tap* internal configuration



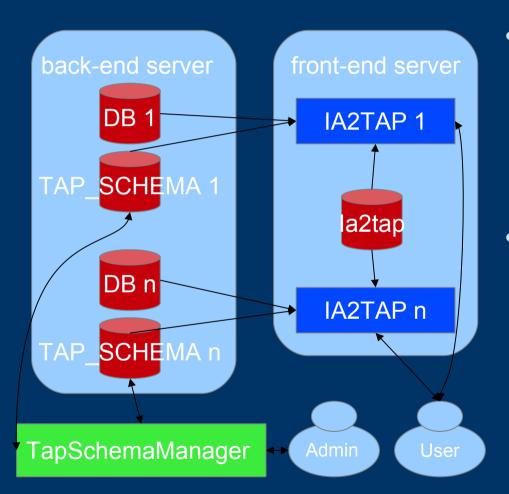


- Flexible TAP\_SCHEMA naming clashes with the TAP specification:
  - each service uses a different TAP\_SCHEMA schema:
    - i.e. a schema not named TAP\_SCHEMA that acts like the TAP\_SCHEMA
    - the schema name is stored in the internal configuration
- This IA2TAP feature joined with the recommended fixed schema name forces an ADQL modification at query level to interrogate the proper custom\_TAP\_SCHEMA
  - letting TAP\_SCHEMA be an optional element of the TAP specification or allowing custom naming can solve this problem



#### TAP services architecture at IA2: diagram





- IA2TAP implementation allows deployment of tablesets served by MySQL servers running on servers, different from the one hosting the web application
- IA2 TAP services architecture separates back-end and frontend on different servers



#### **Deployed services**



- IA2 registered TAP services:
  - Catalogue of WGE photometric redshifts for SDSS candidate quasars and galaxies
  - NASA dust catalogue (EPN-TAP);
- Custom internal TAP service for Asiago data
- Relational Registry Schema TAP interface:
  - Under development, not fully compliant



#### TapSchemaManager overview



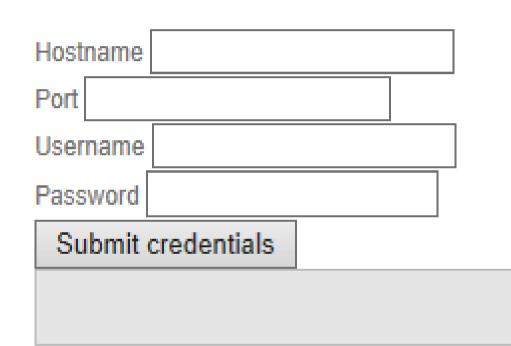
- TapSchemaManager is a supporting web application
  - coded in Java
  - using the Wicket framework
- TapSchemaManager has a graphical interface so that a user can manage the TAP\_SCHEMA schemata in a visual and flexible way

### TapSchemaManager usage workflow Login Selects databases and TAP SCHEMA YES NO New TAP\_SCHEMA? New TAP\_SCHEMA Existing TAP\_SCHEMA Gets dataset metadata Gets dataset metadata and and merges with partially fills TAP\_SCHEMA TAP SCHEMA metadata Edits TAP\_SCHEMA Updates TAP\_SCHEMA (deletes and re-inserts)



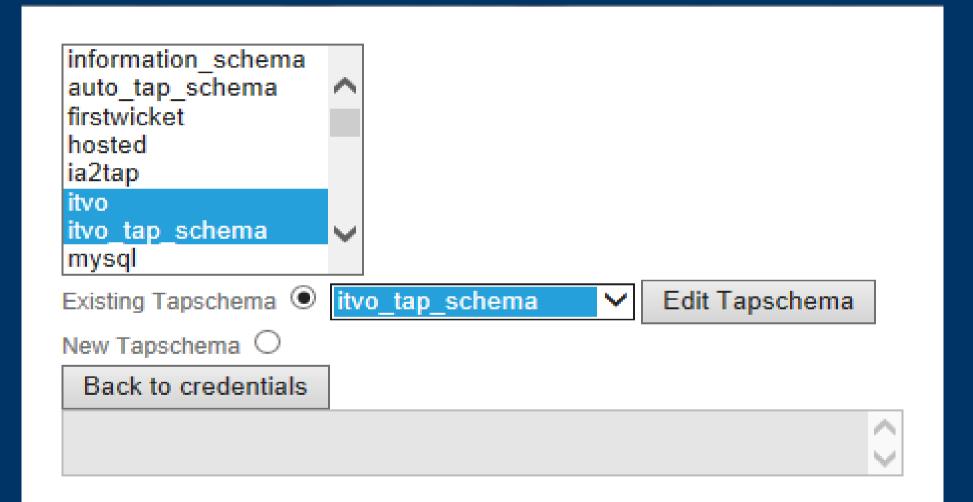
#### Login screenshot















#### Metadata management page screenshot

Ito_tap_schema         ito           hv_lap_schema ukype				chema												
vo_tap_schema description     Bedect itvo_tap_schema keys     ittoo      ittoo		itvo_ta	ap_sch	itvo												
vo_tap_schema description     Bedect itvo_tap_schema keys     ittoo      ittoo	itv	o tap s	chema	a utvpe												
wo_tap_schema.keys       ivo_tap_schema.key_colums       wo_tap_schema.telse       ivo_tap_schema.telse         wo_tap_schema.keys       description       descrip								_								
Itvo_tap_schema.keys utype         vo_tap_schema.keys description         velect itvo_tap_schema.keys         elect       k         k       table_name         column_name       utype         ucd       unit         description       adql:CHAR         itvo_tap_schema.keys       form_table         itvo_tap_schema.keys       description         d       itvo_tap_schema.keys	elect i	tvo_tap_	_schen	na 🗹												
Itvo_tap_schema.keys utype         io_tap_schema.keys description         elect itvo_tap_schema.keys description         elect itvo_tap_schema.keys description         id       itvo_tap_schema.keys keyID         id       itvo_tap_schema.keys key_Id         id       itvo_tap_schema.																
in lap_schema.keys     elect itvo_tap_schema.keys     titvo_tap_schema.keys     tittap_schema.keys     tittap_schema.keys     tittap_schema.keys     tittap_schema.keys <td< td=""><td>[</td><td>itvo_ta</td><td>ap_sch</td><td>nema.keys itvo_tap</td><td>_schema.schemas</td><td>itvo_tap_s</td><td>chema.key_column</td><td>s itvo_tap_sch</td><td>nema.tables</td><td>itvo_tap_schem</td><td>.columns</td><td></td><td></td><td></td><td></td><td></td></td<>	[	itvo_ta	ap_sch	nema.keys itvo_tap	_schema.schemas	itvo_tap_s	chema.key_column	s itvo_tap_sch	nema.tables	itvo_tap_schem	.columns					
o tap_schema keys description ieter tivo_tap_schema keys teter tivo_t																
lect itvo_tap_schema keys   iet itvo_tap_schema keys     ittoo tap_schema keys     ittoo tap_schema keys     ittoo tap_schema keys     ittoo tap_schema keys																
kt       kt       table_name       column_name       utype       ucd       unit       description       datatype       size       principal       ndexed       std         d       kt       fk       itvo_tap_schema.keys       from_table																
i ik itvo_tap_schema keys from_table     i ik     i itvo_tap_schema keys   i icto_tap_schema keys   i icto_tap_sch	lect	tvo_tap_	_schen	na.keys ⊻												
Image: Control of the select schemata																
Image: Control of the select schemata	elect	pk	fk	table_name	column_	_name	utype	ucd	unit	descript	on datatype	size	principal	indexed	std	
Image: Control of the select schemata		pk	fk fk		-	_name	utype	ucd	unit	descript					std	
Image: Character of the select schemata       Image: Character of the select schemata         Image: Character of the select schemata       Image: Character of the select schemata	2	pk	fk fk	itvo_tap_schema.ke	/s from_table	_name	utype	ucd	unit	descript	adql:CHAR	128	0	0	std 1 1	
Image: Character of the select schemata	2	pk	fk fk	itvo_tap_schema.key	rs from_table rs description	_name	utype	ucd	unit		adql:CHAR adql:CHAR	128 512	0	0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
lect all table columns	2	pk	fk fk	itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table description keyID	_name	utype	ucd	unit	descript	adql:CHAR adql:CHAR adql:BIGINT	128 512 0		0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Jpdate all Back to select schemata	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	pk	fk fk	itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table rs description rs keyID rs utype	_name	utype	ucd	unit		adql:CHAR adql:CHAR adql:BIGINT adql:CHAR	128       512       0       512		0 0 0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
				itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table rs description rs keyID rs utype rs key_id	_name	utype	ucd	unit	descript	adql:CHAR adql:CHAR adql:BIGINT adql:CHAR adql:CHAR	128           512           0           512           64		0 0 0 0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
			fk	itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table rs description rs keyID rs utype rs key_id	_name	utype		unit		adql:CHAR adql:CHAR adql:BIGINT adql:CHAR adql:CHAR	128           512           0           512           64		0 0 0 0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	Image: state	III table c	fk column	itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table rs description rs keyID rs utype rs key_id	_name	utype		unit		adql:CHAR adql:CHAR adql:BIGINT adql:CHAR adql:CHAR	128           512           0           512           64		0 0 0 0	std 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	<ul> <li>Image: second second</li></ul>	III table c	fk column	itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key itvo_tap_schema.key	rs from_table rs description rs keyID rs utype rs key_id		utype		unit		adql:CHAR adql:CHAR adql:BIGINT adql:CHAR adql:CHAR	128           512           0           512           64		0 0 0 0	1 1 1 1 1 1 1	





- A better dynamic generation of the results of the TAP
- A more exhaustive logging system (e.g. statistics and more information about the execution of jobs)
- Implementation of the ADQL geometrical functions using mysql\_sphere library
- Integration of IA2TAP internal configuration with the VO-Dance one
- Integration of TapSchemaManager into an administration suite for both the VO-Dance and IA2TAP service publishers
- Guided identification of UCDs, UTYPEs and UNITs to fill the created TAP\_SCHEMA





# Thank you for your attention Pietro Apollo (and the IA2 team)