

L'IVOA (International Virtual Observatory Alliance) è una collaborazione internazionale il cui scopo è agevolare l'interoperabilità degli archivi astronomici e costruire un osservatorio virtuale integrato. Il suo operato si concentra sulla definizione di standard per raggiungere questo scopo. Uno di questi è il Table Access Protocol (TAP), un protocollo per servizi web progettato per dare accesso a dati raccolti genericamente sotto forma di tabelle. Un servizio basato sul TAP dà accesso a tableset immagazzinati in un RDBMS, arricchendoli con metadati astrofisici specifici dell'Osservatorio Virtuale (VO), permette l'interrogazione su questi tableset utilizzando delle query ADQL (Astronomical Data Query Language, un altro standard dell'IVOA) e risponde a queste interrogazioni sotto forma di una tabella che segue lo standard VOTable. I servizi TAP, così come altri servizi basati su protocolli IVOA, sono resi disponibili per la comunità attraverso tutti quegli strumenti sviluppati all'interno della comunità IVOA (per esempio TOPCAT).

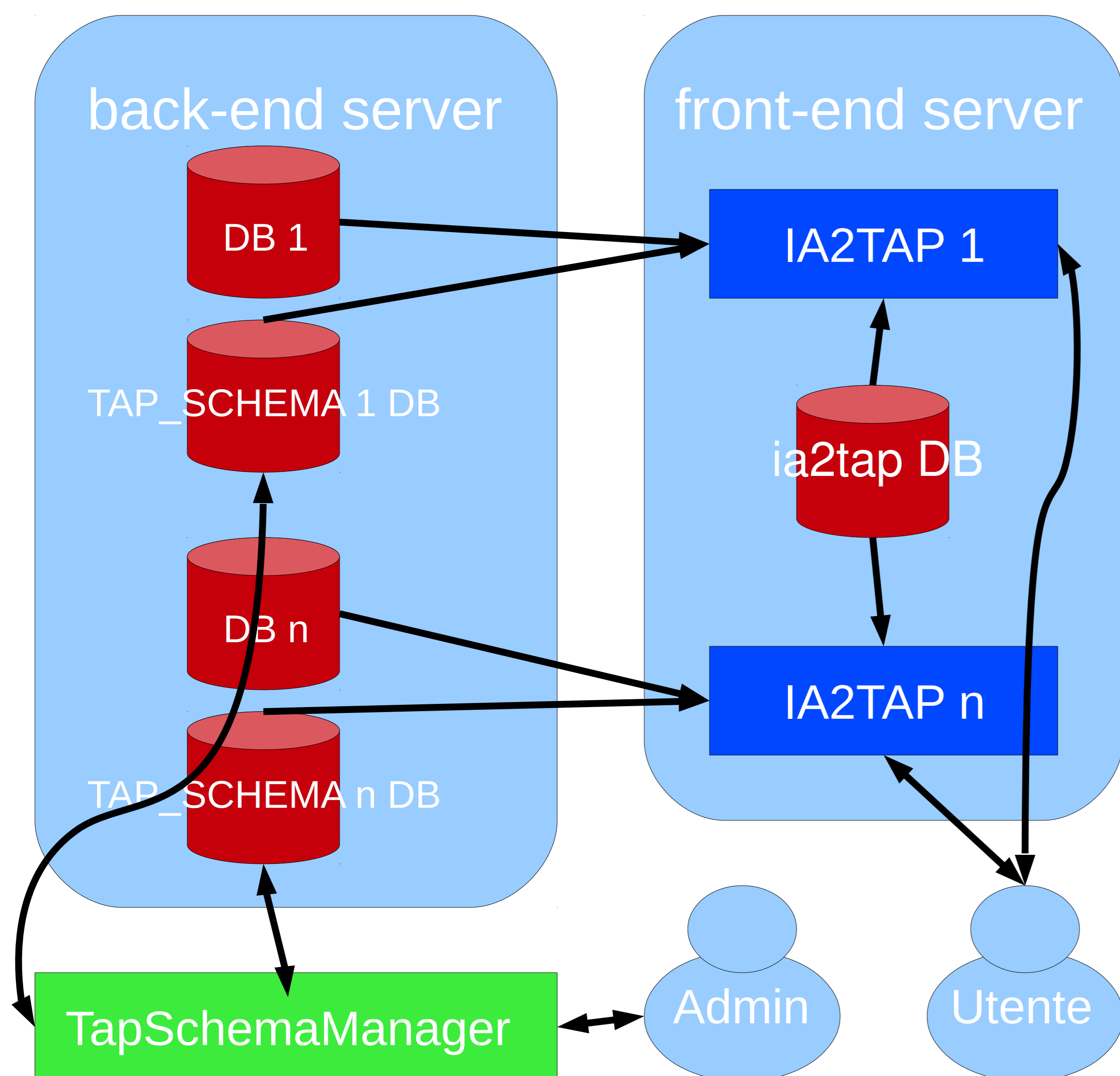
Date le complesse caratteristiche del protocollo TAP, per un ricercatore può essere abbastanza difficile implementare un servizio che ne soddisfi le specifiche ed esporre i propri dati nel mondo VO. Per facilitare questo compito, presso il centro dati INAF-OATs IA2 (Istituto Nazionale di Astrofisica - Osservatorio Astronomico di Trieste, centro Italiano di Archivi Astronomici) è stata sviluppata un'applicazione web, IA2TAP. Utilizzando quest'applicazione, un ricercatore che voglia esporre i propri tableset nel VO deve solamente: 1) configurare la connessione dell'applicazione verso l'RDBMS contenente i database (al momento è supportato solo MySQL) e 2) preparare il TAP_SCHEMA (l'information schema definito dal TAP) per ottenere il risultato voluto.

IA2TAP è un'applicazione web scritta in Java che richiede Glassfish quale web container per l'esecuzione e un server MySQL per immagazzinare la sua configurazione interna (descrizioni del servizio, logging, ecc...). È stata sviluppata sulla base di un insieme di librerie fornite dal Canadian Astronomical Data Center (CADC), le quali soddisfano i principali requisiti del TAP, e personalizzata così da permettere: 1) l'esecuzione di più servizi TAP in parallelo sullo stesso server web e 2) il deployment di tableset serviti da più server MySQL in esecuzione su server differenti da quello che ospita l'applicazione web.

A fianco di IA2TAP è stata anche sviluppata un'applicazione, TapSchemaManager, per agevolare la generazione e manutenzione dei TAP_SCHEMA necessari all'esecuzione dei servizi TAP configurati. TapSchemaManager è un'altra applicazione web scritta in Java con un'interfaccia grafica sviluppata utilizzando il framework Wicket.

Usando IA2TAP e TapSchemaManager un utente può fare il deploy dei propri tableset avvantaggiandosi delle strutture dei servizi TAP presso il centro dati IA2, o facendo il deploy di un proprio servizio TAP con i soli requisiti di un server sul quale siano in esecuzione un web container Glassfish e un server MySQL.

Futuri sviluppi di IA2TAP spaziano da una migliore generazione dinamica dei vari endpoint ausiliari del TAP ad una migliore archiviazione dei risultati, ad un sistema di logging più esaustivo e all'implementazione delle funzioni geometriche dell'ADQL. È anche pianificata l'integrazione della sua configurazione interna e gestione con l'applicazione VO-Dance (l'altro sostegno offerto da IA2 alla comunità astrofisica per la pubblicazione di servizi VO).



L'applicativo TapSchemaManager è stato sviluppato per dotare l'utente di uno strumento atto a creare e mantenere gli schemi TAP_SCHEMA, schemi necessari per la generazione di servizi TAP e che espongono i metadati dei tableset astronomici. Tale applicativo è, come IA2TAP, una web application che richiede un web container e, nell'architettura di IA2, è collocata sul server di front-end (ma può essere collocata su un qualsiasi server).

TapSchemaManager è dotata di un'interfaccia grafica implementata utilizzando il framework Wicket. Mediante questa interfaccia grafica un utente amministratore, inserendo i dati di accesso (per ora quelli di connessione al server MySQL del caso), ha a disposizione le informazioni del DBMS che ospita le risorse da esporre e che ospita o ospiterà il relativo TAP_SCHEMA. Dopo l'accesso potrà selezionare gli schemi, tabelle e colonne da esporre fra quelli a cui le sue credenziali hanno accesso ed inoltre definire un nuovo TAP_SCHEMA o selezionare uno già esistente per definirne i metadati.

L'applicazione è in grado di raccogliere tutti i metadati (informazioni su schemi, tabelle, colonne, indici, ...) delle risorse selezionate e di riportare eventuali metadati già presenti nel TAP_SCHEMA selezionato.

La visualizzazione di tali metadati avviene tramite schede annidate (una per ciascuno schema e per ciascuna tabella dello schema, queste ultime raggruppate per schema). L'utente può navigare fra tali schede, modificarne i metadati e selezionare quali colonne, tabelle o schemi esporre nel TAP_SCHEMA. Una volta soddisfatto l'utente non dovrà far altro che salvare nel TAP_SCHEMA il proprio lavoro.

TapSchemaManager è pensato come una delle componenti che andranno a formare un sistema completo di amministrazione dei servizi generati da IA2TAP.

IA2TAP espone database astrofisici e ne permette l'interrogazione (che può avvenire tramite browser web o con applicativi in grado più pratici e completi, ad esempio TOPCAT). I risultati vengono forniti sotto forma di VOTable. L'applicazione IA2TAP consiste di un pacchetto .war (Java web application package) che per essere eseguito richiede:

- Un web container Glassfish 3.1;
- Un server MySQL (v 5.5+) per la configurazione interna e i dati da esporre.

Presso IA2, dove sono ospitati più servizi TAP, ogni servizio (IA2TAP 1, ..., IA2TAP n in figura) viene generato a partire da un unico pacchetto .war che viene quindi configurato in fase di installazione sul web container. Questo sistema permette la coesistenza in parallelo di più servizi su un unico server di front-end, sul quale è presente anche lo schema di database di configurazione (ia2tap DB in figura), comune a tutti i servizi. Tale schema conserva anche i log di accesso e attività dei servizi.

Ciascun servizio TAP espone i dati astrofisici, specifici della risorsa che rappresenta, appoggiandosi ai database MySQL (DB 1, ..., DB n in figura). Questi ultimi possono essere collocati su un diverso server (ad es. quello di back-end rappresentato in figura) come nel caso dell'architettura utilizzata presso IA2. Gli schemi TAP_SCHEMA (uno per ciascun servizio TAP) che espongono i metadati del servizio sono posizionati sullo stesso server MySQL delle risorse che descrivono.

Ciascuna istanza di IA2TAP risponde alle richieste dell'utente che vengono indirizzate ad un insieme specifico di URL formato da un indirizzo URL di base univoco (es: <http://server/myTap/>) e un sotto albero di indirizzi (es: <http://server/myTap/{ramo}>) che espone alcune caratteristiche tipiche del TAP, questi rami sono:

- *availability*, che riporta la disponibilità del servizio;
- *capabilities*, descrittore delle funzionalità del servizio;
- *tables*, elenco degli schemi, tabelle e colonne esposti dal TAP_SCHEMA;
- *sync*, che risponde alle query ADQL in modo sincrono;
- *async*, per le risposte asincrone;

Le *capabilities* di ciascun servizio TAP sono generate dinamicamente in base ai dati memorizzati nelle tabelle di configurazione dello schema ia2tap.

Le query sincrone vengono eseguite immediatamente e l'applicazione che ha eseguito la chiamata deve attendere la terminazione per ottenere il risultato, pertanto il loro uso è conveniente per la richiesta di piccole quantità di dati.

Il metodo asincrono è invece utile per interrogazioni che possono richiedere tempi di esecuzione lunghi; infatti non obbliga l'utente ad attendere la risposta del servizio ma ne consente il monitoraggio e l'accesso ai risultati avviene in un qualsiasi momento dopo il termine dell'operazione.

L'impiego di un applicativo come TOPCAT per l'uso di un servizio TAP come quelli generati da IA2TAP consiste solitamente in:

- impostazione dell'URL del servizio;
- visualizzazione delle *capabilities* e degli schemi, tabelle, colonne esposte;
- esecuzione di una query ADQL sincrona o asincrona;
- visualizzazione dei risultati.



Pietro Apollo
INAF-Osservatorio Astronomico di Trieste
apollo@oats.inaf.it

Marco Molinaro
INAF-Osservatorio Astronomico di Trieste
molinaro@oats.inaf.it

Riccardo Smareglia
INAF-Osservatorio Astronomico di Trieste
smareglia@oats.inaf.it