Jul 16, 2021

Protocol configuration in VOSSIA

N.F. Calabria

INAF - IA2 group

Summary

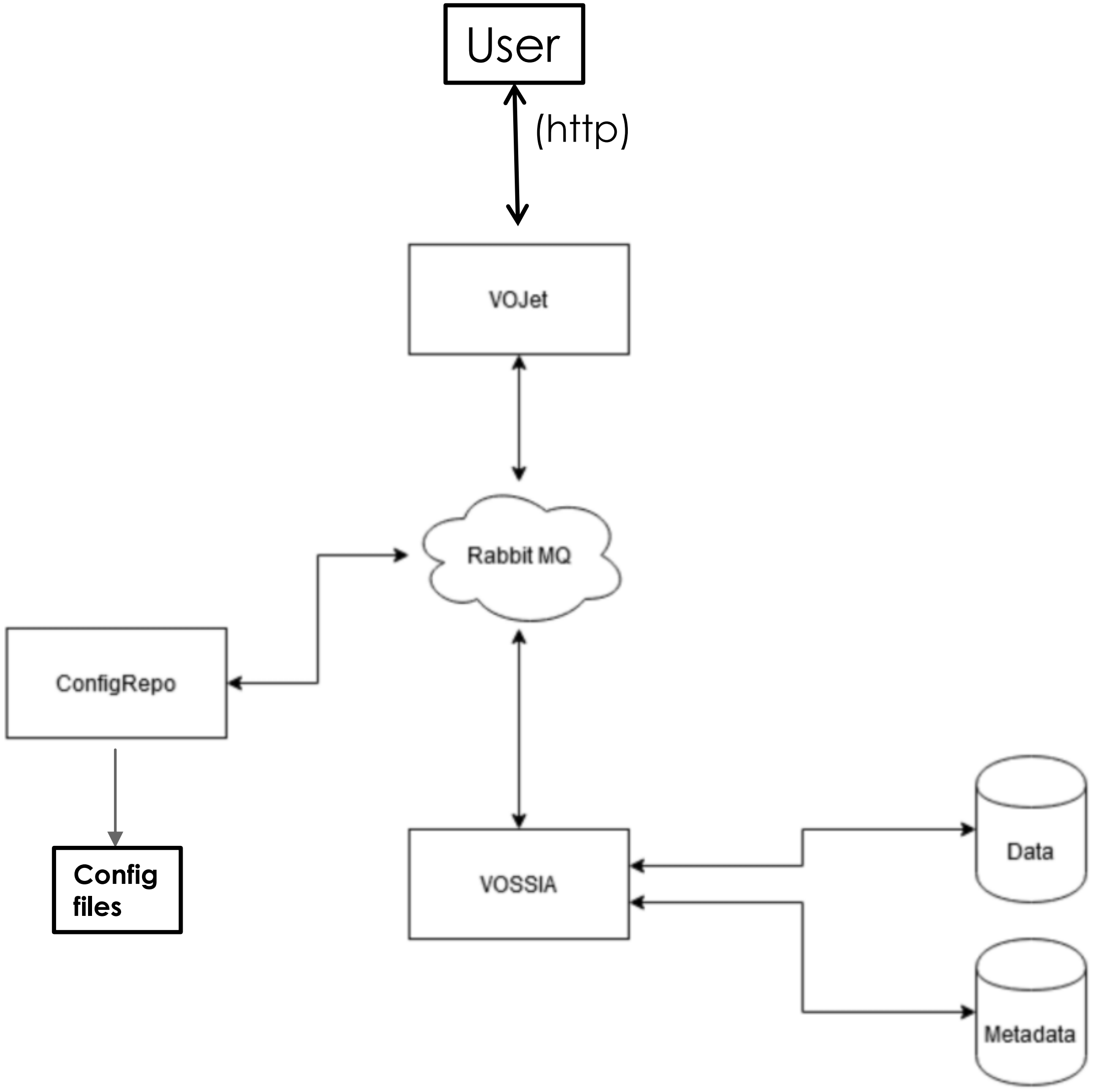
- Introducing VOSSIA 1.0
- Milestones
- VOSSIA 2.0
- Configuring exposed services
- Request example
- Perspectives and plans

Introducing VOSSIA 1.0

VOSSIA 1.0 is a web application developed at INAF-OATs IA2 to allow easy publishing of VO-related resources according to the standards defined in the Simple Image Access Protocol V2.0 (SIAP) recommendation.

VOSSIA 1.0 architecture is modular and relies on the Advanced Message Queuing Protocol (AMQP) implementation RabbitMQ for communication between modules.

VOSSIA 1.0 exposes its services via the VOJet 1.0 web interface, a separate module developed at INAF-OATs IA2.



Milestones

- Before 2017: RabbitMQ-based architecture defined by Francesco Cepparo for his Master Thesis
  «Sviluppo di un sistema distribuito modulare e scalabile per la pubblicazione di servizi VO»
  Università degli studi di Udine, corso di laurea Magistrale in Informatica.
  Relatore: I. Scagnetto, Correlatore: M. Molinaro. AA 2014/15

- May 2017: I took over development of a Simple Cone Search service with RabbitMQ distributed architecture

- June 2018: I reported status at ASTERICS European Data Provider Forum (Heidelberg)
  VOSSIA 1.0, SIAP service based on RabbitMQ architecture, was in development.

- July 2018:
  «Italian center for astronomical archives publishing solution: modular and distributed»
  Software and Cyberinfrastructure for Astronomy, SPIE Astronomical Telescopes + Instrumentation
  Austin, Texas, US. (M.Molinaro, N.F. Calabria, R. Butora, S. Zorba, R. Smareglia)

- April 2019: VOSSIA 1.0 reaches current status: includes IA2 A&A, static availability, capabilities and DataLink descriptor. VOSSIA 2.0, Spring based, development starts.

VOSSIA 1.0 is used internally at IA2 and currently unreleased to the public.
VOSSIA 2.0 is the topic of this talk

VOSSIA 2.0

Main feature:
VOSSIA 2.0 allows the exposure of data according to recommended VO and custom VO-like protocols. Goal is flexibility in exposing data already existing in a datacenter (e.g. custom or missing columns) by trading standard compliance off if needed at configuration level.

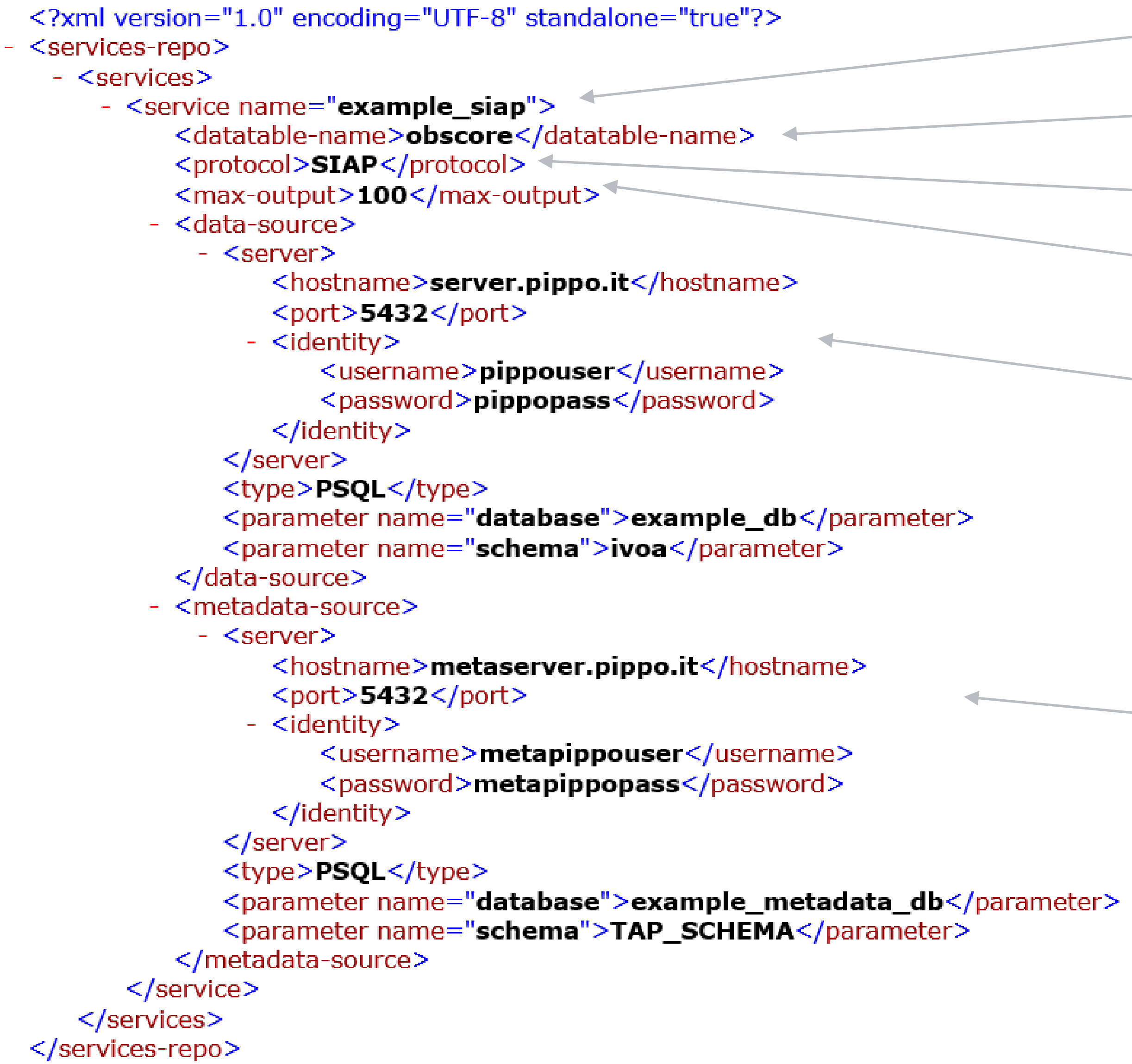
Requirements:

- Java JDK 15
- Centos 7
- PostgreSQL with pgSphere extension

We didn’t test other platforms, but there are no obvious limitations…

Code is meant to be extended to other databases for both data and metadata easily, with or without geometric extensions like pgSphere.

Configuring exposed services: services.xml file



```
  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <services-repo>
  - <services>
    - <service name="example_siap">
        <datatable-name>obscore</datatable-name>
        <protocol>SIAP</protocol>
        <max-output>100</max-output>
      - <data-source>
        - <server>
            <hostname>server.pippo.it</hostname>
            <port>5432</port>
          - <identity>
              <username>pippouser</username>
              <password>pippopass</password>
            </identity>
          </server>
          <type>PSQL</type>
          <parameter name="database">example_db</parameter>
          <parameter name="schema">ivoa</parameter>
        </data-source>
      - <metadata-source>
        - <server>
            <hostname>metaserver.pippo.it</hostname>
            <port>5432</port>
          - <identity>
              <username>metapippouser</username>
              <password>metapippopass</password>
            </identity>
          </server>
          <type>PSQL</type>
          <parameter name="database">example_metadata_db</parameter>
          <parameter name="schema">TAP_SCHEMA</parameter>
        </metadata-source>
      </service>
    </services>
  </services-repo>
```

Name of the service is its unique identifier in requests

One data source table per service policy

Protocol identifier (next slide)

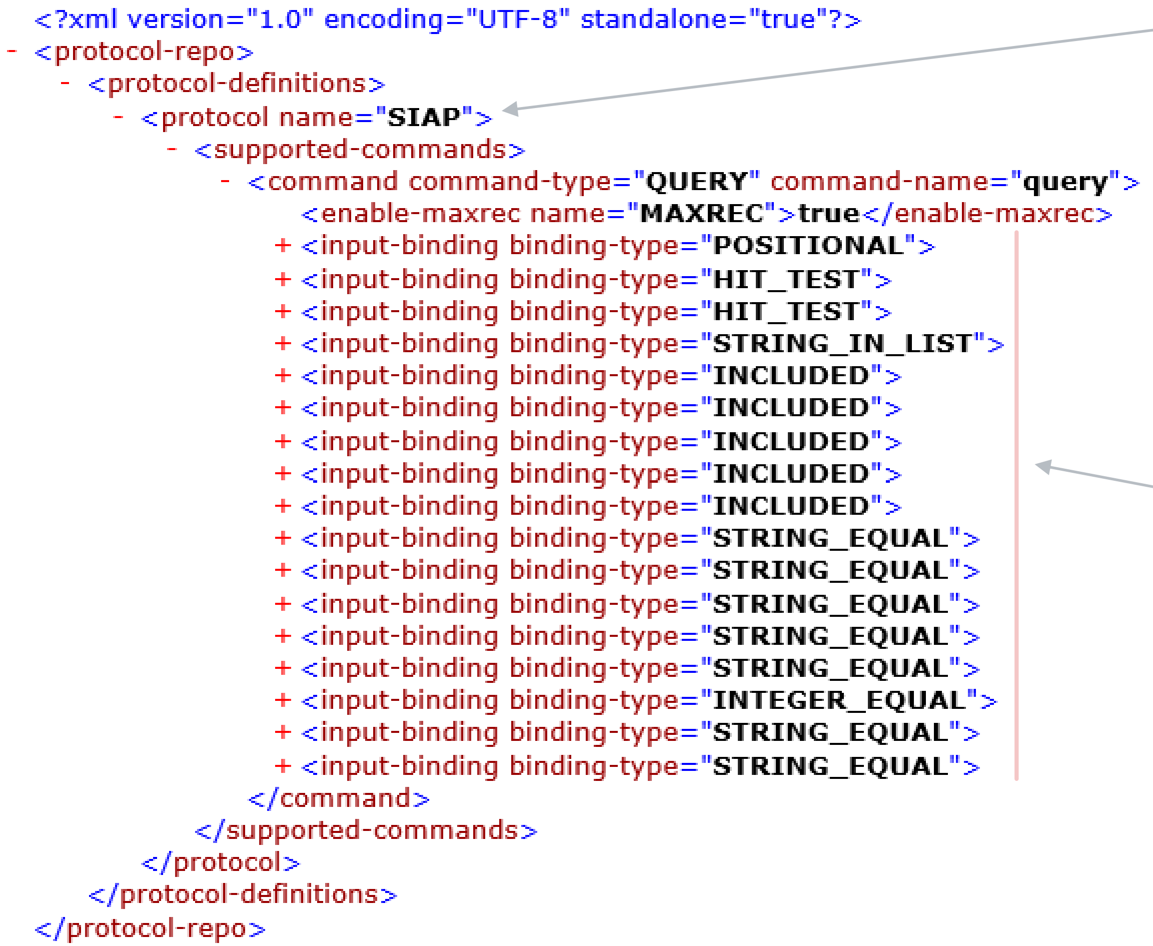
MAXREC override

Data source credentials
It must contain datatable-name table

Metadata source credentials
It must be a valid TAP_SCHEMA containing metadata for datatable-name table (just TAP_SCHEMA, no need to set up a tap service)

We use TASMAN by S. Zorba (internal tool) to generate TAP_SCHEMAs for existing tables

Configuring protocols: descriptors.xml



```
  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <protocol-repo>
  - <protocol-definitions>
    - <protocol name="SIAP">
      - <supported-commands>
        - <command command-type="QUERY" command-name="query">
            <enable-maxrec name="MAXREC">true</enable-maxrec>
          + <input-binding binding-type="POSITIONAL">
          + <input-binding binding-type="HIT_TEST">
          + <input-binding binding-type="HIT_TEST">
          + <input-binding binding-type="STRING_IN_LIST">
          + <input-binding binding-type="INCLUDED">
          + <input-binding binding-type="INCLUDED">
          + <input-binding binding-type="INCLUDED">
          + <input-binding binding-type="INCLUDED">
          + <input-binding binding-type="INCLUDED">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="INTEGER_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          + <input-binding binding-type="STRING_EQUAL">
          </command>
        </supported-commands>
      </protocol>
    </protocol-definitions>
  </protocol-repo>
```

Protocol identifier used in services.xml file

This is one endpoint of the service
e.g. query, capabilities, availability…
backed by a class for its command-type

Input bindings define how query parameters are:

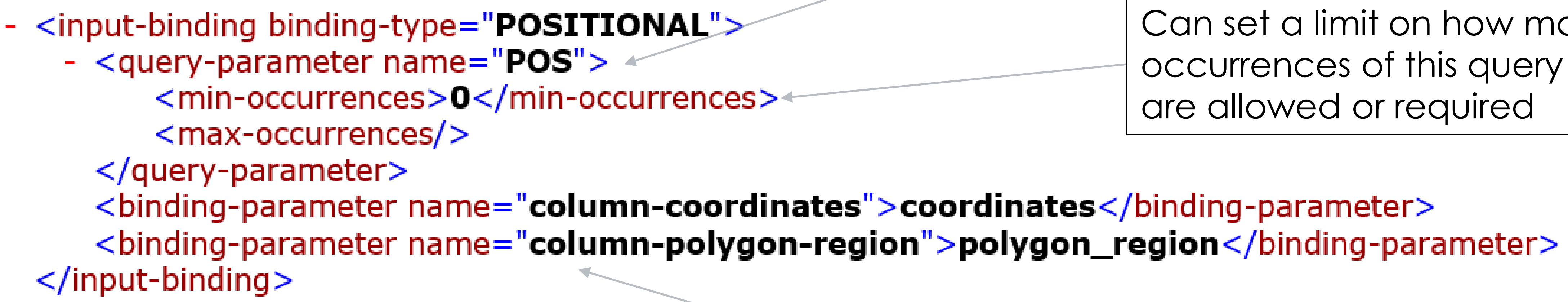
- Parsed
- Validated
- Mapped to a query clause

Let’s expand them in the next slides, one per binding-type.

Configuring protocols: input binding types

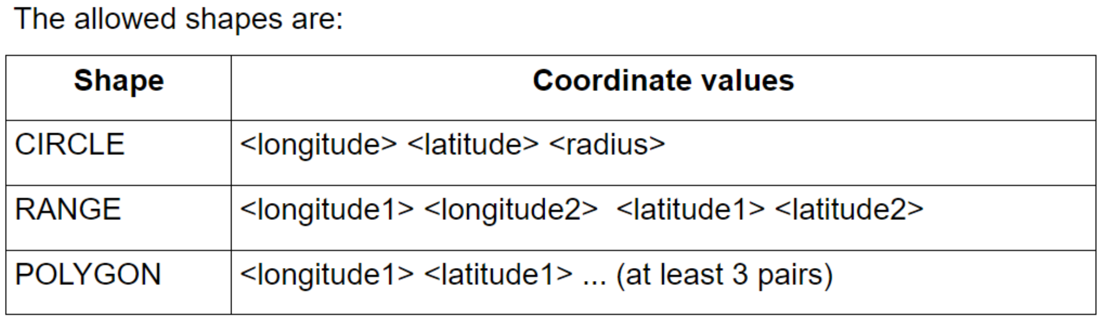
This input binding is triggered by query parameter «POS» occurrences in query string

Can set a limit on how many occurrences of this query parameter are allowed or required



```
- <input-binding binding-type="POSITIONAL">
  - <query-parameter name="POS">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
    </query-parameter>
    <binding-parameter name="column-coordinates">coordinates</binding-parameter>
    <binding-parameter name="column-polygon-region">polygon_region</binding-parameter>
  </input-binding>
```

Binding parameters are binding type specific

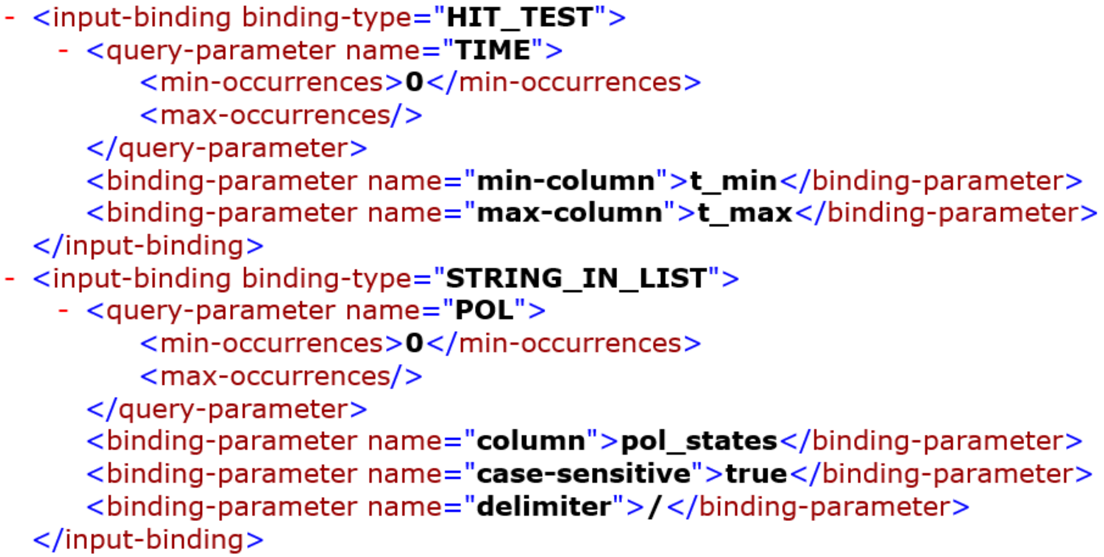


The allowed shapes are:

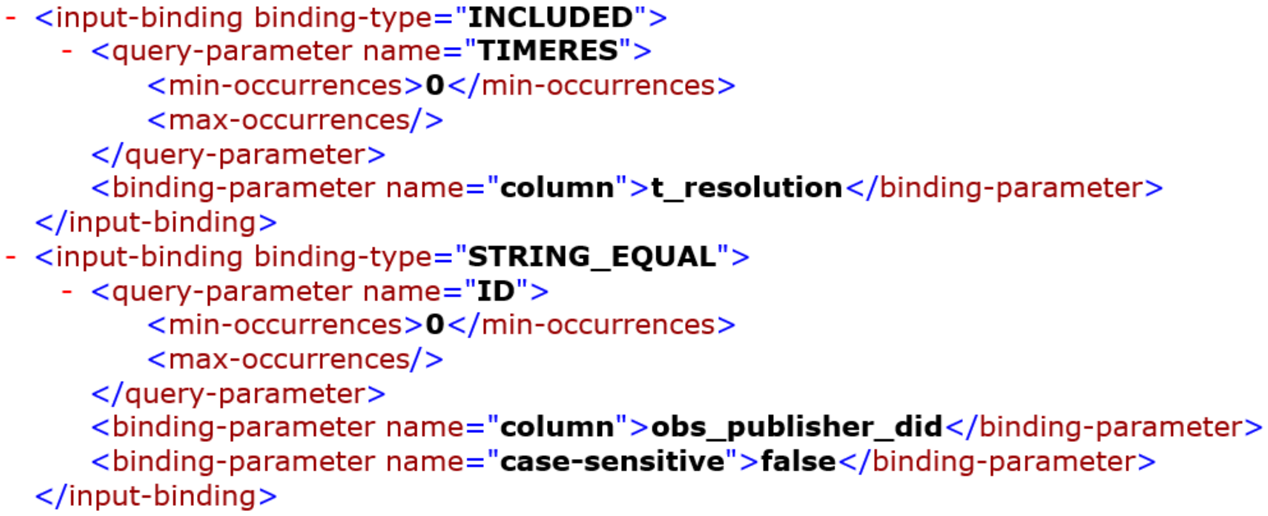
Shape	Coordinate values
CIRCLE	<longitude> <latitude> <radius>
RANGE	<longitude1> <longitude2> <latitude1> <latitude2>
POLYGON	<longitude1> <latitude1> ... (at least 3 pairs)

From IVOA SIAP v2 recommendation:
Input is validated according to this table.
This can be changed by creating a custom class for a custom POSITIONAL-like binding type

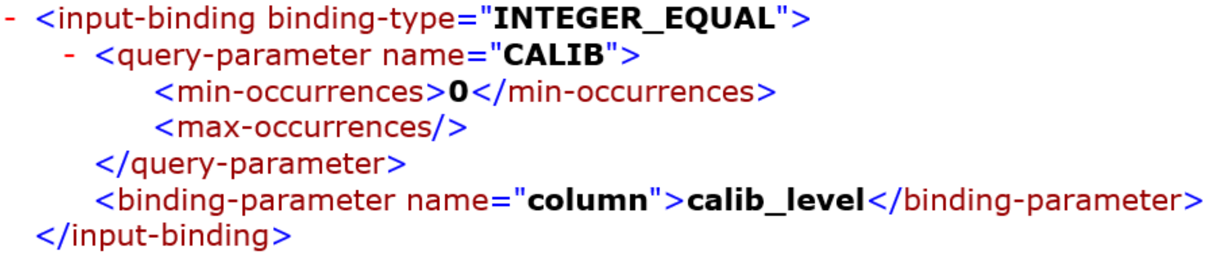
Configuring protocols: input binding types



```
- <input-binding binding-type="HIT_TEST">
   - <query-parameter name="TIME">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
     </query-parameter>
     <binding-parameter name="min-column">t_min</binding-parameter>
     <binding-parameter name="max-column">t_max</binding-parameter>
  </input-binding>
- <input-binding binding-type="STRING_IN_LIST">
   - <query-parameter name="POL">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
     </query-parameter>
     <binding-parameter name="column">pol_states</binding-parameter>
     <binding-parameter name="case-sensitive">true</binding-parameter>
     <binding-parameter name="delimiter">/</binding-parameter>
  </input-binding>
```

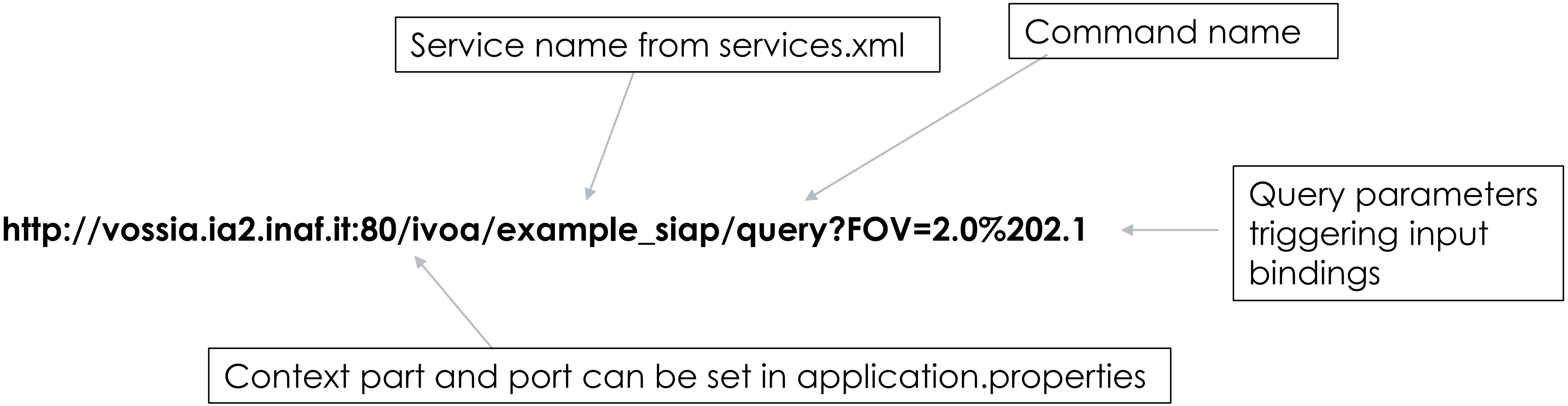


```
- <input-binding binding-type="INCLUDED">
   - <query-parameter name="TIMERES">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
     </query-parameter>
     <binding-parameter name="column">t_resolution</binding-parameter>
  </input-binding>
- <input-binding binding-type="STRING_EQUAL">
   - <query-parameter name="ID">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
     </query-parameter>
     <binding-parameter name="column">obs_publisher_did</binding-parameter>
     <binding-parameter name="case-sensitive">false</binding-parameter>
  </input-binding>
```



```
- <input-binding binding-type="INTEGER_EQUAL">
   - <query-parameter name="CALIB">
        <min-occurrences>0</min-occurrences>
        <max-occurrences/>
     </query-parameter>
     <binding-parameter name="column">calib_level</binding-parameter>
  </input-binding>
```

Request example and output



Service name from services.xml

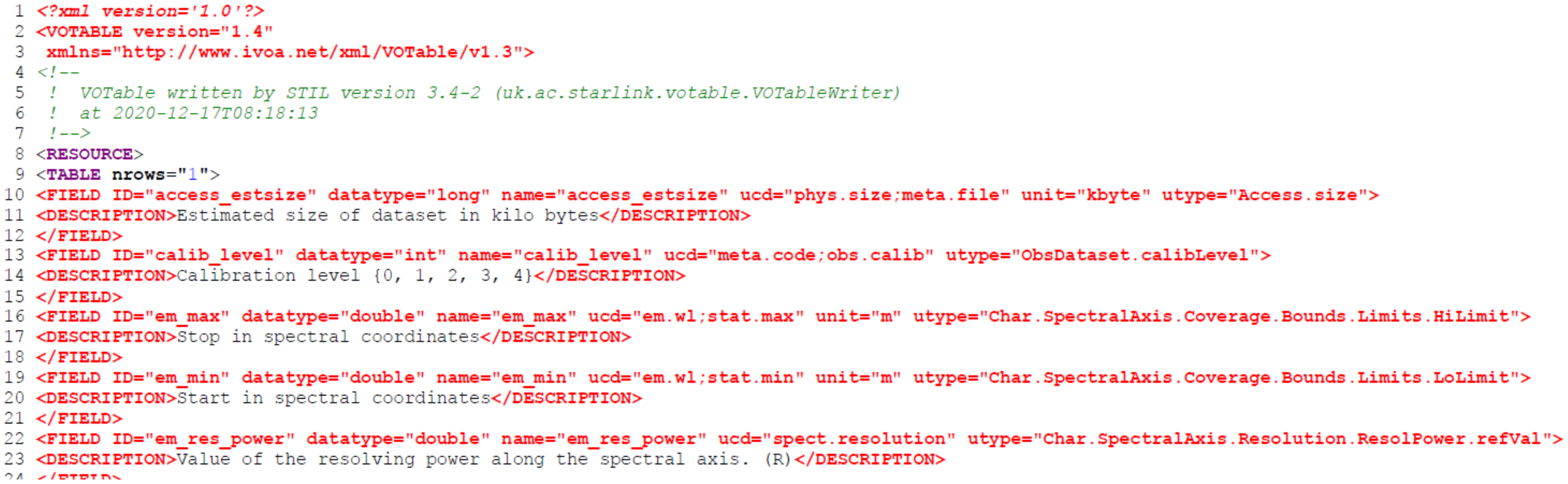
Command name

http://vossia.ia2.inaf.it:80/ivoa/example_siap/query?FOV=2.0%202.1

Query parameters triggering input bindings

Context part and port can be set in application.properties

Output is a VOTable generated by STIL (http://www.star.bris.ac.uk/~mbt/stil/)



```
<?xml version='1.0'?>
<VOTABLE version="1.4"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.3">
<!--
 !  VOTable written by STIL version 3.4-2 (uk.ac.starlink.votable.VOTableWriter)
 !  at 2020-12-17T08:18:13
 !-->
<RESOURCE>
<TABLE nrows="1">
<FIELD ID="access_estsize" datatype="long" name="access_estsize" ucd="phys.size;meta.file" unit="kbyte" utype="Access.size">
<DESCRIPTION>Estimated size of dataset in kilo bytes</DESCRIPTION>
</FIELD>
<FIELD ID="calib_level" datatype="int" name="calib_level" ucd="meta.code;obs.calib" utype="ObsDataset.calibLevel">
<DESCRIPTION>Calibration level {0, 1, 2, 3, 4}</DESCRIPTION>
</FIELD>
<FIELD ID="em_max" datatype="double" name="em_max" ucd="em.wl;stat.max" unit="m" utype="Char.SpectralAxis.Coverage.Bounds.Limits.HiLimit">
<DESCRIPTION>Stop in spectral coordinates</DESCRIPTION>
</FIELD>
<FIELD ID="em_min" datatype="double" name="em_min" ucd="em.wl;stat.min" unit="m" utype="Char.SpectralAxis.Coverage.Bounds.Limits.LoLimit">
<DESCRIPTION>Start in spectral coordinates</DESCRIPTION>
</FIELD>
<FIELD ID="em_res_power" datatype="double" name="em_res_power" ucd="spect.resolution" utype="Char.SpectralAxis.Resolution.ResolPower.refVal">
<DESCRIPTION>Value of the resolving power along the spectral axis. (R)</DESCRIPTION>
</FIELD>
```

…

Perspectives and plans

- Add more configurations:
  Parameter groups, COOSYS, TIMESYS, include DataLink descriptors
- Capabilities and availability
- Collect use-cases to release VOSSIA 2.0 with a nice set of base classes for input bindings in addition to the ones defined in IVOA Recommendations
- Plan to report VOSSIA 2.0 status + hands on session at ESCAPE Data Provider Forum (Nov. 2021)
- Goal is to release VOSSIA 2.0 in the next year (2022)
  Code will be available as soon as it reaches an acceptable maturity