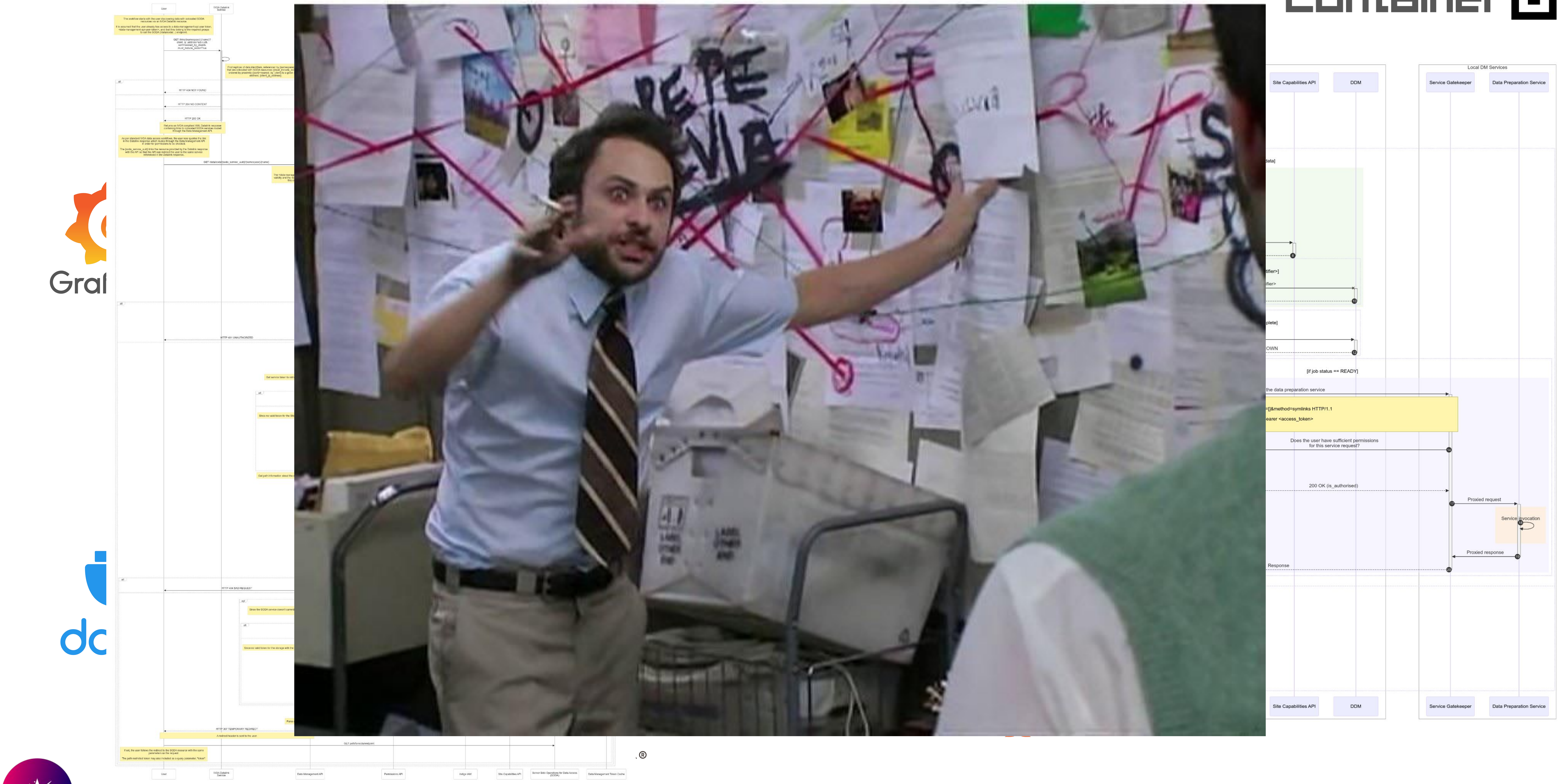
Data Management Services for SRCNet v0.1

Rob Barnsley
SKAO

How to give yourself an existential crisis


PEPE SILVIA
HTTP 307 TEMPORARY REDIRECT
A redirect header is sent to the user.
GET path/to/soda/endpoint
If set, the user follows the redirect to the SODA resource with the same parameters as the request.
The path restricted token may also included as a query parameter, "token".
User
IVOA Datalink Service
Data Management API
Permissions API
Indigo IAM
Site Capabilities API
Server Side Operations for Data Access (SODA)
Data Management Token Cache
Local DM Services
Site Capabilities API
DDM
Service Gatekeeper
Data Preparation Service
[if job status == READY]
Does the user have sufficient permissions for this service request?
200 OK (is_authorised)
Proxied request
Service invocation
Proxied response
Response

Problem reduction

SRCNet v0.1 DM

SCIENTIFIC DATA MANAGEMENT
RUCIO

+ A&A + everything else

😱

A better angle in: High level DM related flows for v0.1

- Consider three distinct stakeholders:
  - A **user** has a level of access in accordance with data policies
  - A **site administrator** has the required privileges to manage data at their node
  - A **service operator** has full access to all Rucio functionality; they will be responsible for the day to day running of Rucio and all that entails, and as such require the lowest level (root) access.

- Within the context of an SRCNet **user**, we need to think about:
  - Data discovery (what data is there?)
  - Data location (where is the data?)
  - Data access (how do I get a local copy of the data?)
  - Data staging (how can I stage data for remote work on a particular compute node?)

A better angle in: High level DM related flows for v0.1 (continued)

- Within the context of an **operator/site admin of an SRCNet node**, we need to (additionally) think about:
  - Data logistics e.g. I need N copies of the data at sites X, Y and Z
  - Data ingestion
  - Data curation e.g. update metadata for a data product

- Can we use internal Rucio systems alone for all these flows? Yes, but…
  - At time of implementation, OIDC token support was in its infancy
  - Current permissions system doesn't quite satisfy need for fine grained control over e.g. listing data and metadata
  - For consistency, want permissions handled at a layer higher than Rucio & through a centralised permissions system that is used for all services
  - Don’t want to force users (or site administrators) to have to learn the Rucio ecosystem and associated tooling

- To empower users and site administrators we have decided to abstract services via a set of APIs
  - This includes Rucio

v0.1 APIs

- **What?**

  Form a significant part of the public facing component of the SRCNet that an SRCNet stakeholder will utilise to perform actions, either directly through their REST interfaces or via command line clients

- **Why?**

  Abstract interfaces to SRCNet services allow signatures to be predetermined and technology to be switched out at a later date if required

v0.1 APIs (continued)

Data management

- Data discovery
- Data location
- Data access
- Data staging
- Data logistics
- Data curation

Site capabilities

- Listing basic attributes of sites in the SRCNet
- Listing available storages & supported storage protocols e.g. https/xroot
- Listing available compute & associated services e.g. Rucio, SI, Dask, Jupyterhub

Permissions

- Authorising access to an API’s route
- Authorising a token exchange for a particular service
- Authorising access to a service

Authentication

- Requesting tokens
- Exchanging tokens for access to different services

v0.1 APIs (continued)

Data management

- Data discovery
- Data location
- Data access
- Data staging
- Data logistics
- Data curation

Site capabilities

- Listing basic attributes of sites in the SRCNet
- Listing available storages & supported storage protocols e.g. https/xroot
- Listing available compute & associated services e.g. Rucio, SI, Dask, Jupyterhub

Permissions

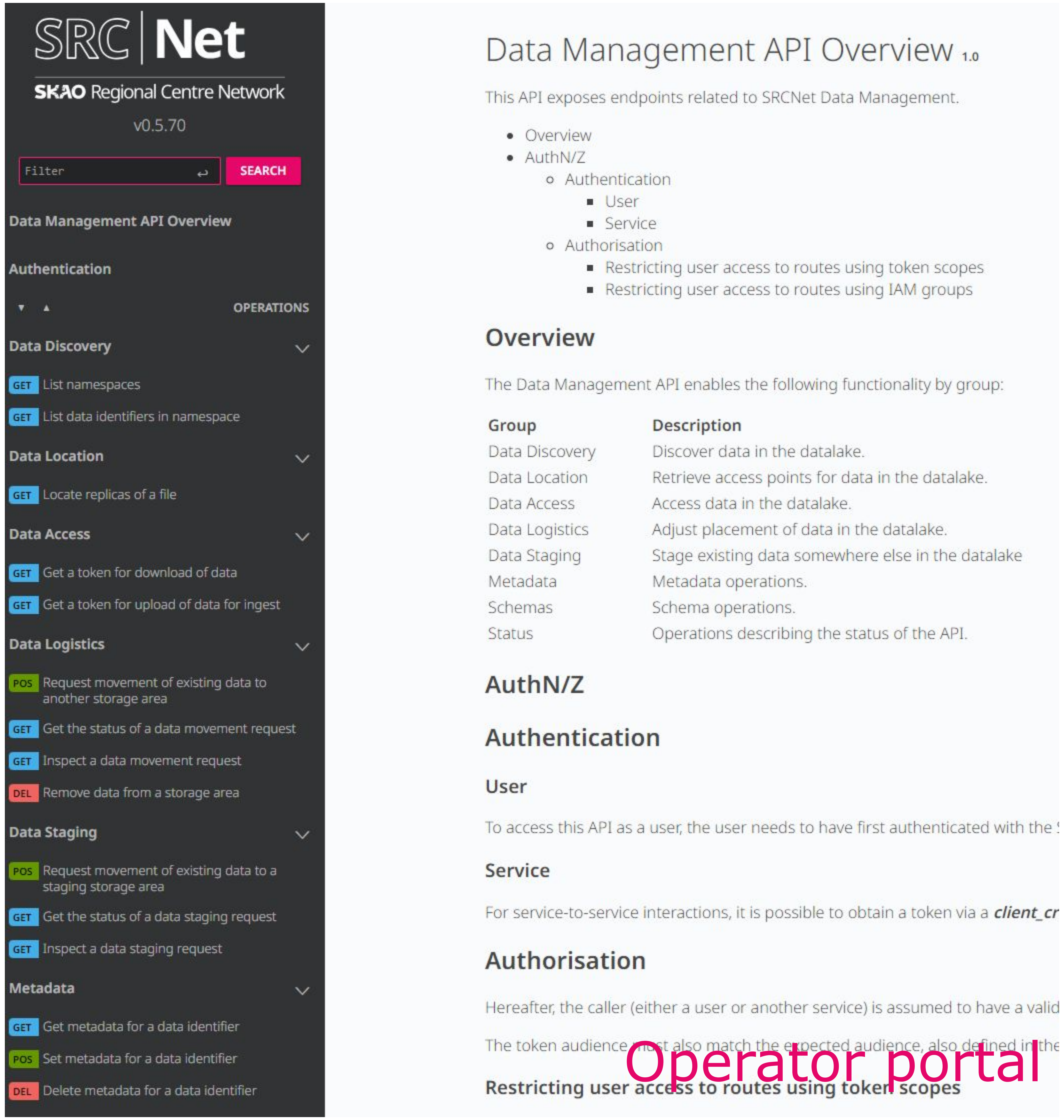
- Authorising access to an API's route
- Authorising a token exchange for a particular service
- Authorising access to a service

Authentication

- Requesting tokens
- Exchanging tokens for access to different services

Data Management API (DM API)

+ FastAPI + RapiDoc



SRC | Net

SKAO Regional Centre Network

v0.5.70

Data Management API Overview

Authentication

Data Discovery

- GET List namespaces
- GET List data identifiers in namespace

Data Location

- GET Locate replicas of a file

Data Access

- GET Get a token for download of data
- GET Get a token for upload of data for ingest

Data Logistics

- POS Request movement of existing data to another storage area
- GET Get the status of a data movement request
- GET Inspect a data movement request
- DEL Remove data from a storage area

Data Staging

- POS Request movement of existing data to a staging storage area
- GET Get the status of a data staging request
- GET Inspect a data staging request

Metadata

- GET Get metadata for a data identifier
- POS Set metadata for a data identifier
- DEL Delete metadata for a data identifier

Data Management API Overview 1.0

This API exposes endpoints related to SRCNet Data Management.

- Overview
- AuthN/Z
  - Authentication
    - User
    - Service
  - Authorisation
    - Restricting user access to routes using token scopes
    - Restricting user access to routes using IAM groups

Overview

The Data Management API enables the following functionality by group:

Group	Description
Data Discovery	Discover data in the datalake.
Data Location	Retrieve access points for data in the datalake.
Data Access	Access data in the datalake.
Data Logistics	Adjust placement of data in the datalake.
Data Staging	Stage existing data somewhere else in the datalake
Metadata	Metadata operations.
Schemas	Schema operations.
Status	Operations describing the status of the API.

AuthN/Z

Authentication

User

To access this API as a user, the user needs to have first authenticated with the

Service

For service-to-service interactions, it is possible to obtain a token via a ***client_cr***

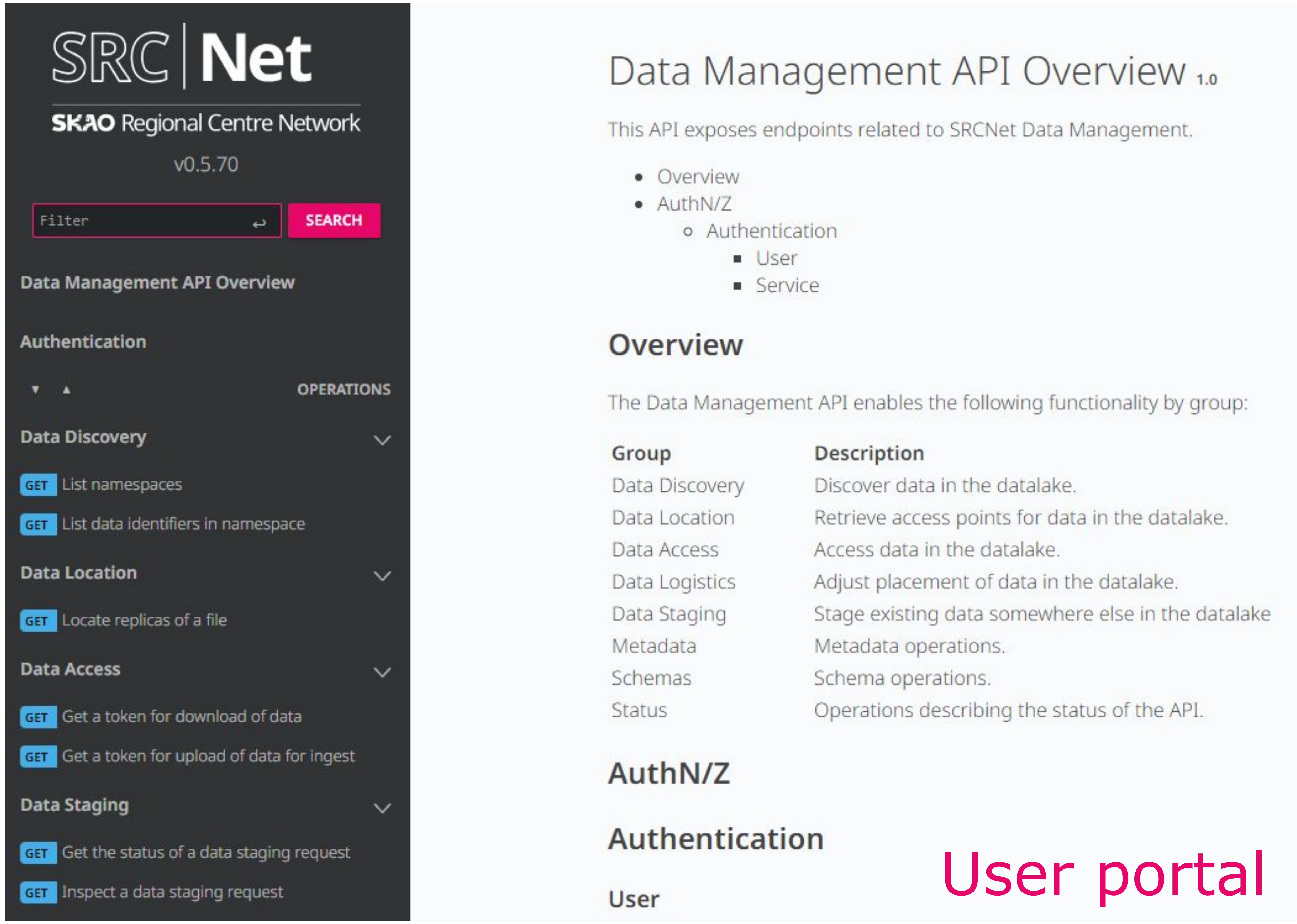
Authorisation

Hereafter, the caller (either a user or another service) is assumed to have a valid

The token audience must also match the expected audience, also defined in the

Restricting user access to routes using token scopes

Operator portal



SRC | Net

SKAO Regional Centre Network

v0.5.70

Data Management API Overview

Authentication

Data Discovery

- GET List namespaces
- GET List data identifiers in namespace

Data Location

- GET Locate replicas of a file

Data Access

- GET Get a token for download of data
- GET Get a token for upload of data for ingest

Data Staging

- GET Get the status of a data staging request
- GET Inspect a data staging request

Data Management API Overview 1.0

This API exposes endpoints related to SRCNet Data Management.

- Overview
- AuthN/Z
  - Authentication
    - User
    - Service

Overview

The Data Management API enables the following functionality by group:

Group	Description
Data Discovery	Discover data in the datalake.
Data Location	Retrieve access points for data in the datalake.
Data Access	Access data in the datalake.
Data Logistics	Adjust placement of data in the datalake.
Data Staging	Stage existing data somewhere else in the datalake
Metadata	Metadata operations.
Schemas	Schema operations.
Status	Operations describing the status of the API.

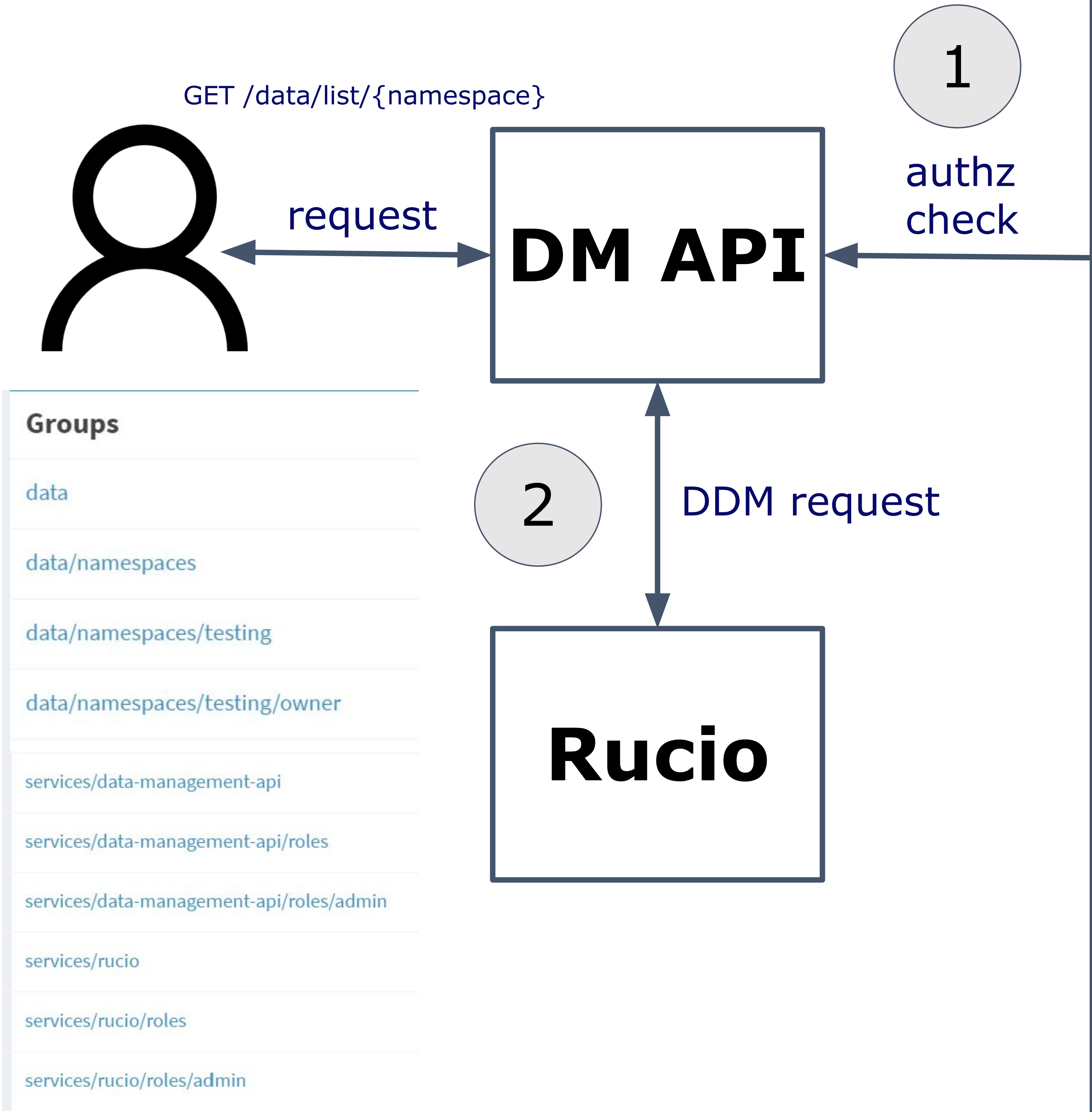
AuthN/Z

Authentication

User

User portal

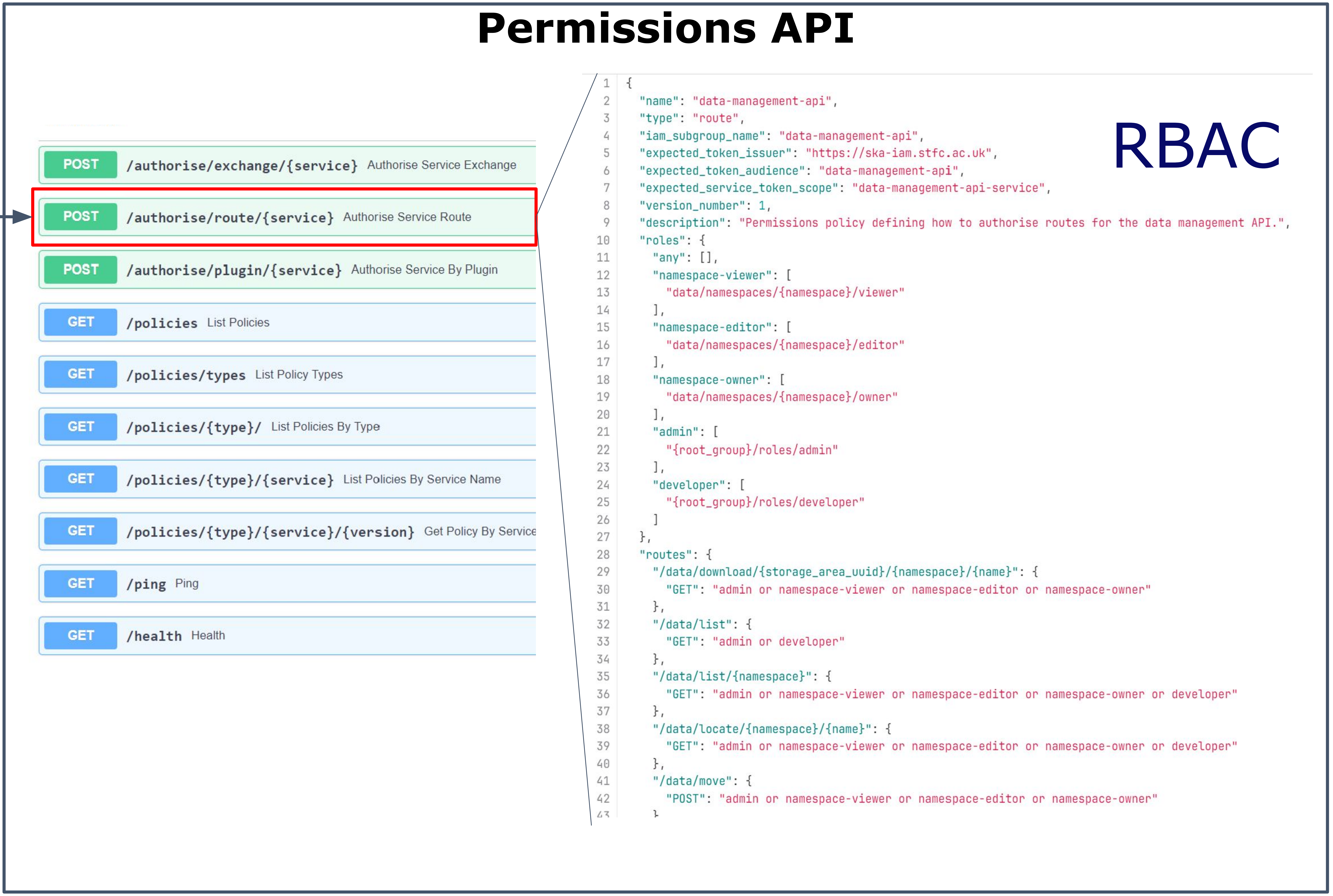
Permissions API (& interaction with DM API)



Groups

- data
- data/namespaces
- data/namespaces/testing
- data/namespaces/testing/owner
- services/data-management-api
- services/data-management-api/roles
- services/data-management-api/roles/admin
- services/rucio
- services/rucio/roles
- services/rucio/roles/admin

Permissions API



- POST `/authorise/exchange/{service}` Authorise Service Exchange
- POST `/authorise/route/{service}` Authorise Service Route
- POST `/authorise/plugin/{service}` Authorise Service By Plugin
- GET `/policies` List Policies
- GET `/policies/types` List Policy Types
- GET `/policies/{type}/` List Policies By Type
- GET `/policies/{type}/{service}` List Policies By Service Name
- GET `/policies/{type}/{service}/{version}` Get Policy By Service
- GET `/ping` Ping
- GET `/health` Health

RBAC

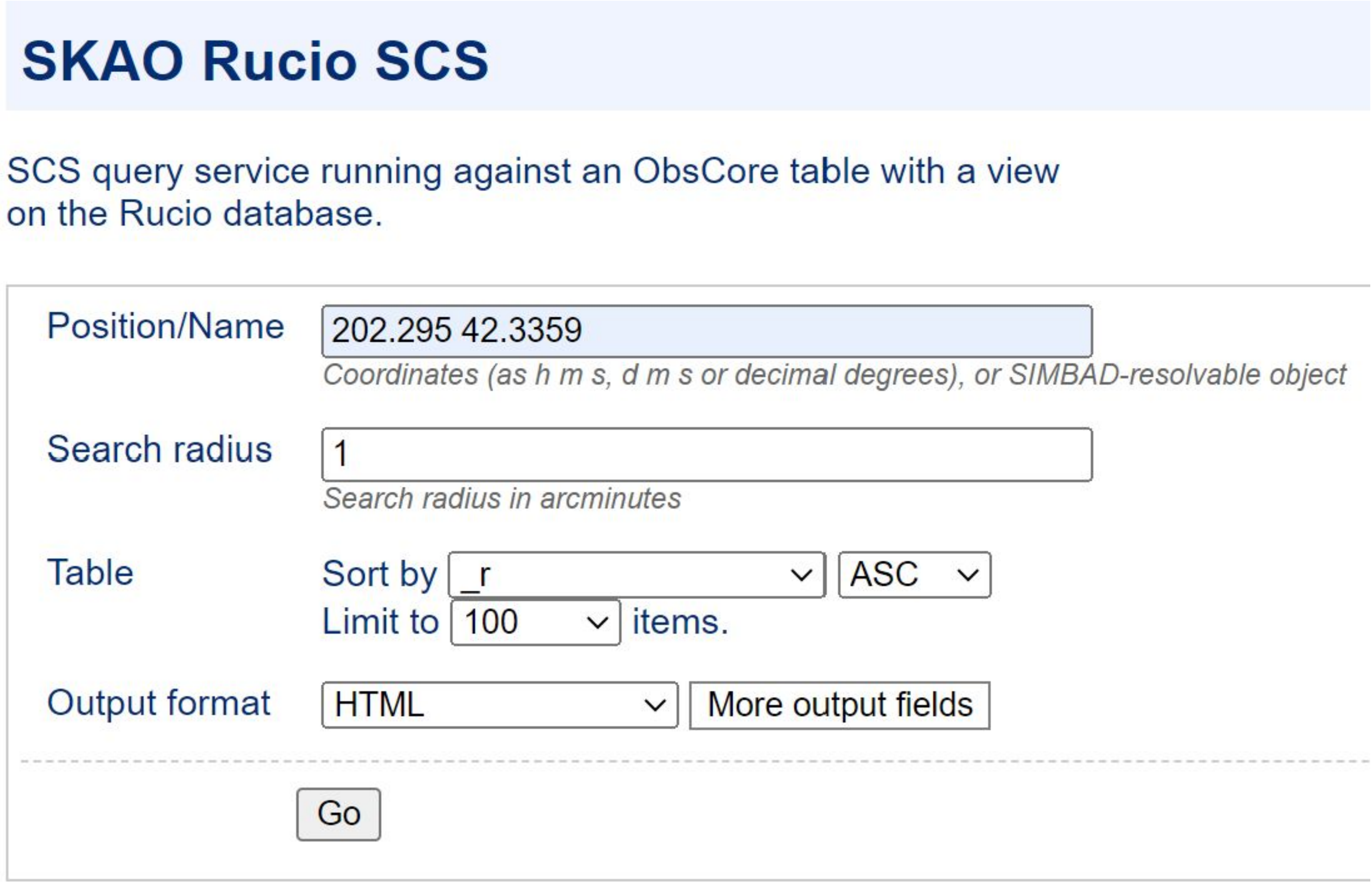
```
{
  "name": "data-management-api",
  "type": "route",
  "iam_subgroup_name": "data-management-api",
  "expected_token_issuer": "https://ska-iam.stfc.ac.uk",
  "expected_token_audience": "data-management-api",
  "expected_service_token_scope": "data-management-api-service",
  "version_number": 1,
  "description": "Permissions policy defining how to authorise routes for the data management API.",
  "roles": {
    "any": [],
    "namespace-viewer": [
      "data/namespaces/{namespace}/viewer"
    ],
    "namespace-editor": [
      "data/namespaces/{namespace}/editor"
    ],
    "namespace-owner": [
      "data/namespaces/{namespace}/owner"
    ],
    "admin": [
      "{root_group}/roles/admin"
    ],
    "developer": [
      "{root_group}/roles/developer"
    ]
  },
  "routes": {
    "/data/download/{storage_area_uuid}/{namespace}/{name}": {
      "GET": "admin or namespace-viewer or namespace-editor or namespace-owner"
    },
    "/data/list": {
      "GET": "admin or developer"
    },
    "/data/list/{namespace}": {
      "GET": "admin or namespace-viewer or namespace-editor or namespace-owner or developer"
    },
    "/data/locate/{namespace}/{name}": {
      "GET": "admin or namespace-viewer or namespace-editor or namespace-owner or developer"
    },
    "/data/move": {
      "POST": "admin or namespace-viewer or namespace-editor or namespace-owner"
    }
```

A (very) brief overview of (some of the) services that build off of this

Data Discovery

IVOA DaCHS

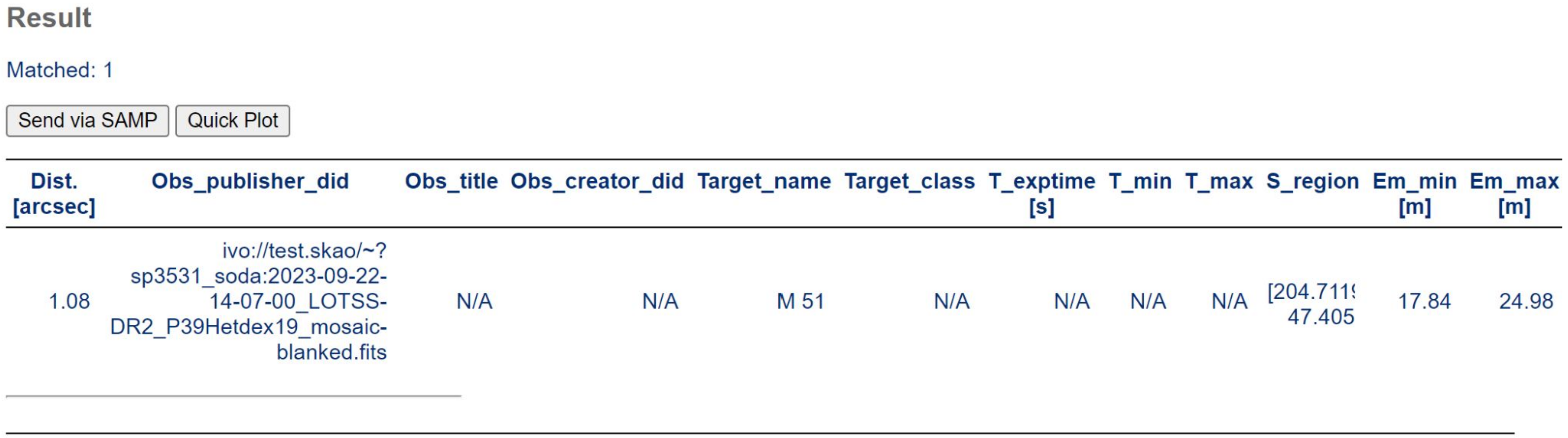
A software suite implementing various International Virtual Observatory Alliance (IVOA) protocols, e.g. Simple Cone Search (SCS)



SKAO Rucio SCS

SCS query service running against an ObsCore table with a view on the Rucio database.

Position/Name	202.295 42.3359 *Coordinates (as h m s, d m s or decimal degrees), or SIMBAD-resolvable object*
Search radius	1 *Search radius in arcminutes*
Table	Sort by _r ASC Limit to 100 items.
Output format	HTML



Result

Matched: 1

Dist. [arcsec]	Obs_publisher_did	Obs_title	Obs_creator_did	Target_name	Target_class	T_exptime [s]	T_min	T_max	S_region	Em_min [m]	Em_max [m]
1.08	ivo://test.skao/~?sp3531_soda:2023-09-22-14-07-00_LOTSS-DR2_P39Hetdex19_mosaic-blanked.fits	N/A	N/A	M 51	N/A	N/A	N/A	N/A	[204.711 47.405	17.84	24.98

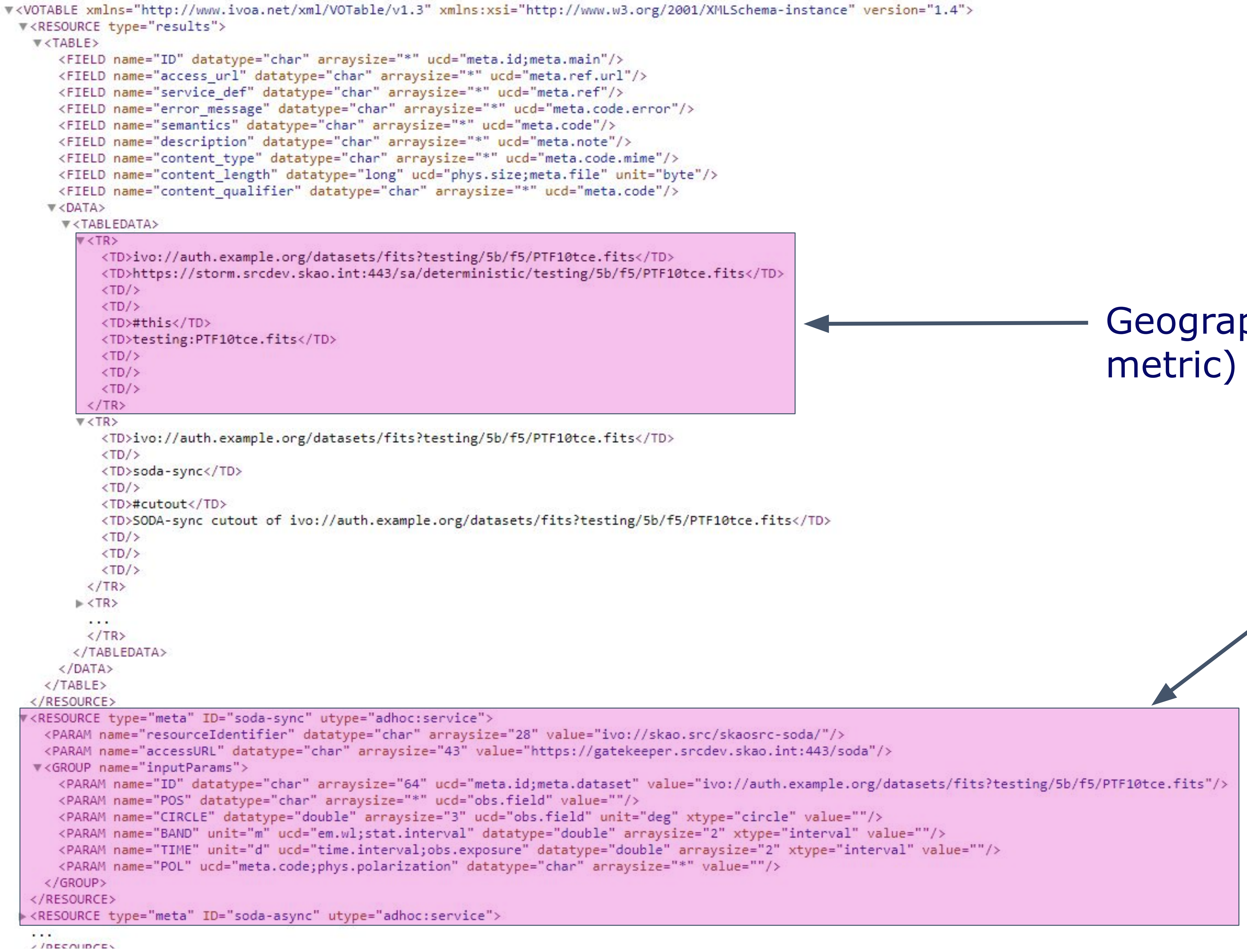
Metadata from external postgres instance managed by Rucio plugin system

Data Location

IVOA Datalink

A service acting as the glue between IVOA services and Rucio (via the DM API)

e.g. GET https://datalink.ivoa.srcdev.skao.int/rucio/links?id=testing:PTF10tce.fits



```
▼<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.4">
  ▼<RESOURCE type="results">
    ▼<TABLE>
       <FIELD name="ID" datatype="char" arraysize="*" ucd="meta.id;meta.main"/>
       <FIELD name="access_url" datatype="char" arraysize="*" ucd="meta.ref.url"/>
       <FIELD name="service_def" datatype="char" arraysize="*" ucd="meta.ref"/>
       <FIELD name="error_message" datatype="char" arraysize="*" ucd="meta.code.error"/>
       <FIELD name="semantics" datatype="char" arraysize="*" ucd="meta.code"/>
       <FIELD name="description" datatype="char" arraysize="*" ucd="meta.note"/>
       <FIELD name="content_type" datatype="char" arraysize="*" ucd="meta.code.mime"/>
       <FIELD name="content_length" datatype="long" ucd="phys.size;meta.file" unit="byte"/>
       <FIELD name="content_qualifier" datatype="char" arraysize="*" ucd="meta.code"/>
      ▼<DATA>
        ▼<TABLEDATA>
          ▼<TR>
             <TD>ivo://auth.example.org/datasets/fits?testing/5b/f5/PTF10tce.fits</TD>
             <TD>https://storm.srcdev.skao.int:443/sa/deterministic/testing/5b/f5/PTF10tce.fits</TD>
             <TD/>
             <TD/>
             <TD>#this</TD>
             <TD>testing:PTF10tce.fits</TD>
             <TD/>
             <TD/>
             <TD/>
           </TR>
          ▼<TR>
             <TD>ivo://auth.example.org/datasets/fits?testing/5b/f5/PTF10tce.fits</TD>
             <TD/>
             <TD>soda-sync</TD>
             <TD/>
             <TD>#cutout</TD>
             <TD>SODA-sync cutout of ivo://auth.example.org/datasets/fits?testing/5b/f5/PTF10tce.fits</TD>
             <TD/>
             <TD/>
             <TD/>
           </TR>
          ▶<TR>
           ...
           </TR>
         </TABLEDATA>
       </DATA>
     </TABLE>
   </RESOURCE>
  ▼<RESOURCE type="meta" ID="soda-sync" utype="adhoc:service">
     <PARAM name="resourceIdentifier" datatype="char" arraysize="28" value="ivo://skao.src/skaosrc-soda/"/>
     <PARAM name="accessURL" datatype="char" arraysize="43" value="https://gatekeeper.srcdev.skao.int:443/soda"/>
    ▼<GROUP name="inputParams">
       <PARAM name="ID" datatype="char" arraysize="64" ucd="meta.id;meta.dataset" value="ivo://auth.example.org/datasets/fits?testing/5b/f5/PTF10tce.fits"/>
       <PARAM name="POS" datatype="char" arraysize="*" ucd="obs.field" value=""/>
       <PARAM name="CIRCLE" datatype="double" arraysize="3" ucd="obs.field" unit="deg" xtype="circle" value=""/>
       <PARAM name="BAND" unit="m" ucd="em.wl;stat.interval" datatype="double" arraysize="2" xtype="interval" value=""/>
       <PARAM name="TIME" unit="d" ucd="time.interval;obs.exposure" datatype="double" arraysize="2" xtype="interval" value=""/>
       <PARAM name="POL" ucd="meta.code;phys.polarization" datatype="char" arraysize="*" value=""/>
     </GROUP>
   </RESOURCE>
  ▶<RESOURCE type="meta" ID="soda-async" utype="adhoc:service">
   ...
   </RESOURCE>
```

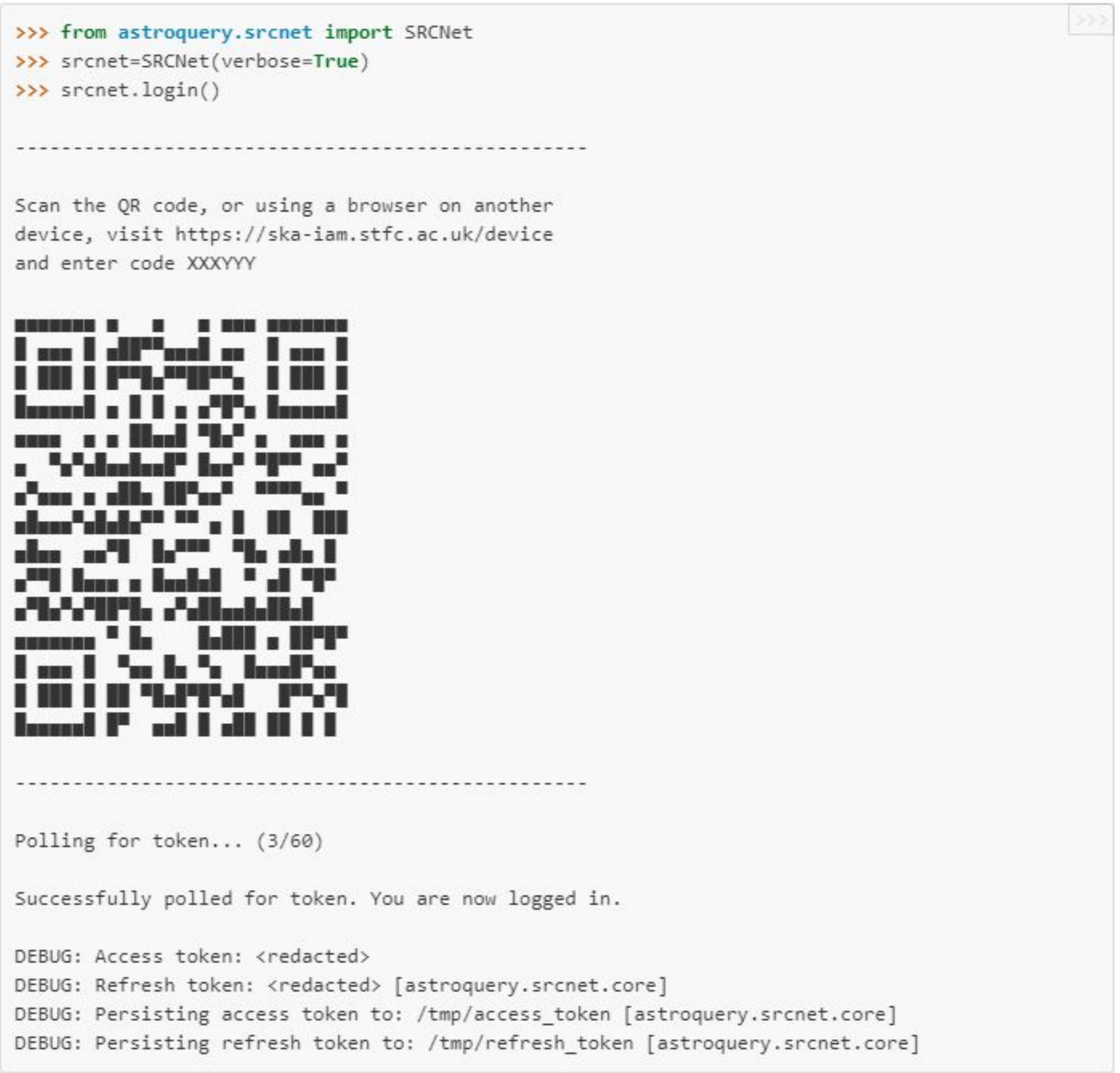
Geographically nearest (or any other metric) replica

Services available at node

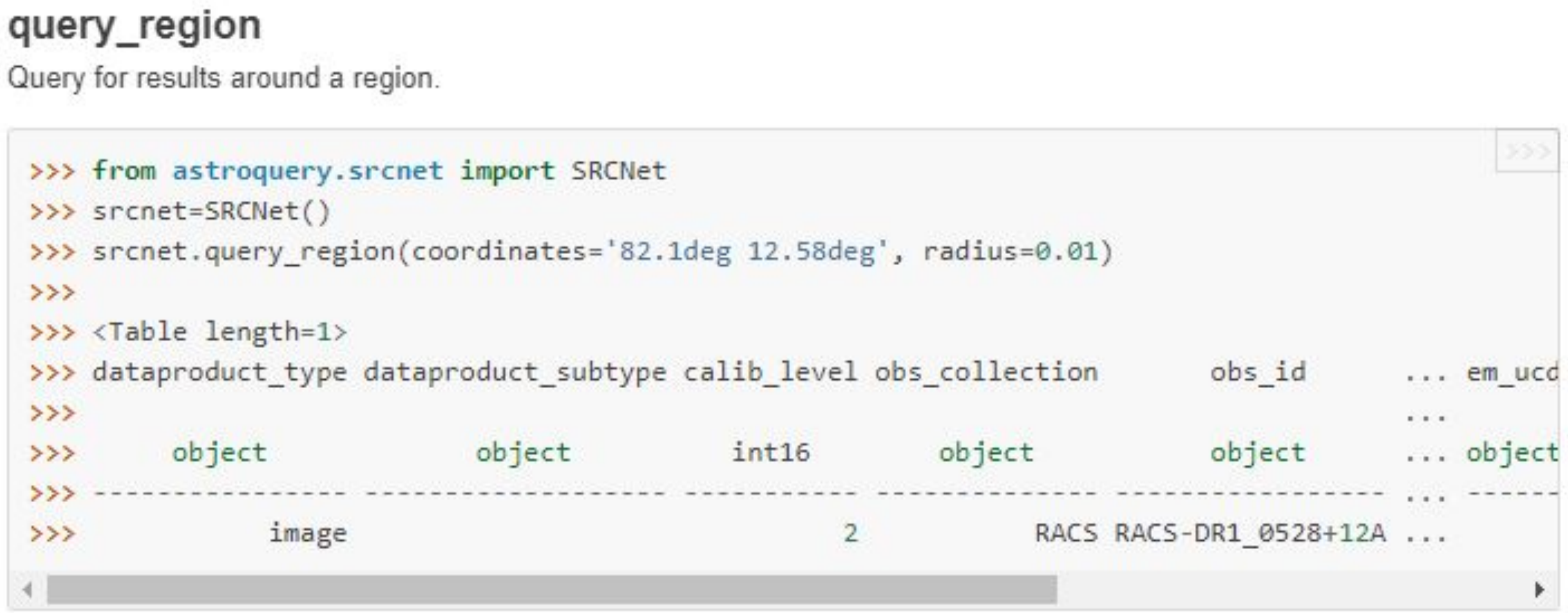
Data Access (also discovery)

astroquery extension

An extension to a widely used Python package offering astronomers a unified interface to query diverse astronomical databases using IVOA standards and protocols



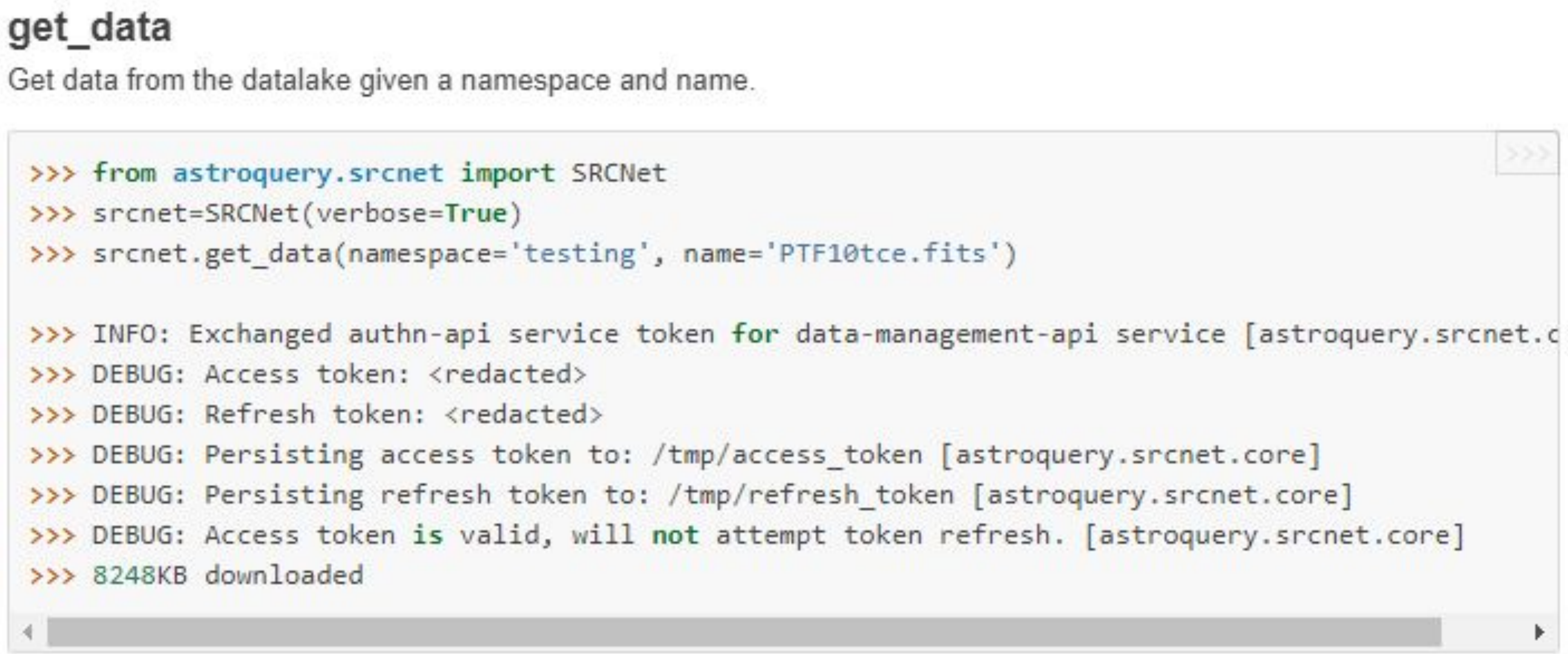
```
>>> from astroquery.srcnet import SRCNet
>>> srcnet=SRCNet(verbose=True)
>>> srcnet.login()

--------------------------------------------------

Scan the QR code, or using a browser on another
device, visit https://ska-iam.stfc.ac.uk/device
and enter code XXXYYY

--------------------------------------------------

Polling for token... (3/60)

Successfully polled for token. You are now logged in.

DEBUG: Access token: <redacted>
DEBUG: Refresh token: <redacted> [astroquery.srcnet.core]
DEBUG: Persisting access token to: /tmp/access_token [astroquery.srcnet.core]
DEBUG: Persisting refresh token to: /tmp/refresh_token [astroquery.srcnet.core]
```



query_region

Query for results around a region.

```
>>> from astroquery.srcnet import SRCNet
>>> srcnet=SRCNet()
>>> srcnet.query_region(coordinates='82.1deg 12.58deg', radius=0.01)
>>>
>>> <Table length=1>
>>> dataproduct_type dataproduct_subtype calib_level obs_collection      obs_id      ... em_ucd
>>>                                                                                  ...
>>>      object             object          int16        object          object      ... object
>>> ---------------- ------------------- ----------- -------------- ---------------- ... ------
>>>            image                               2           RACS RACS-DR1_0528+12A ...
```



get_data

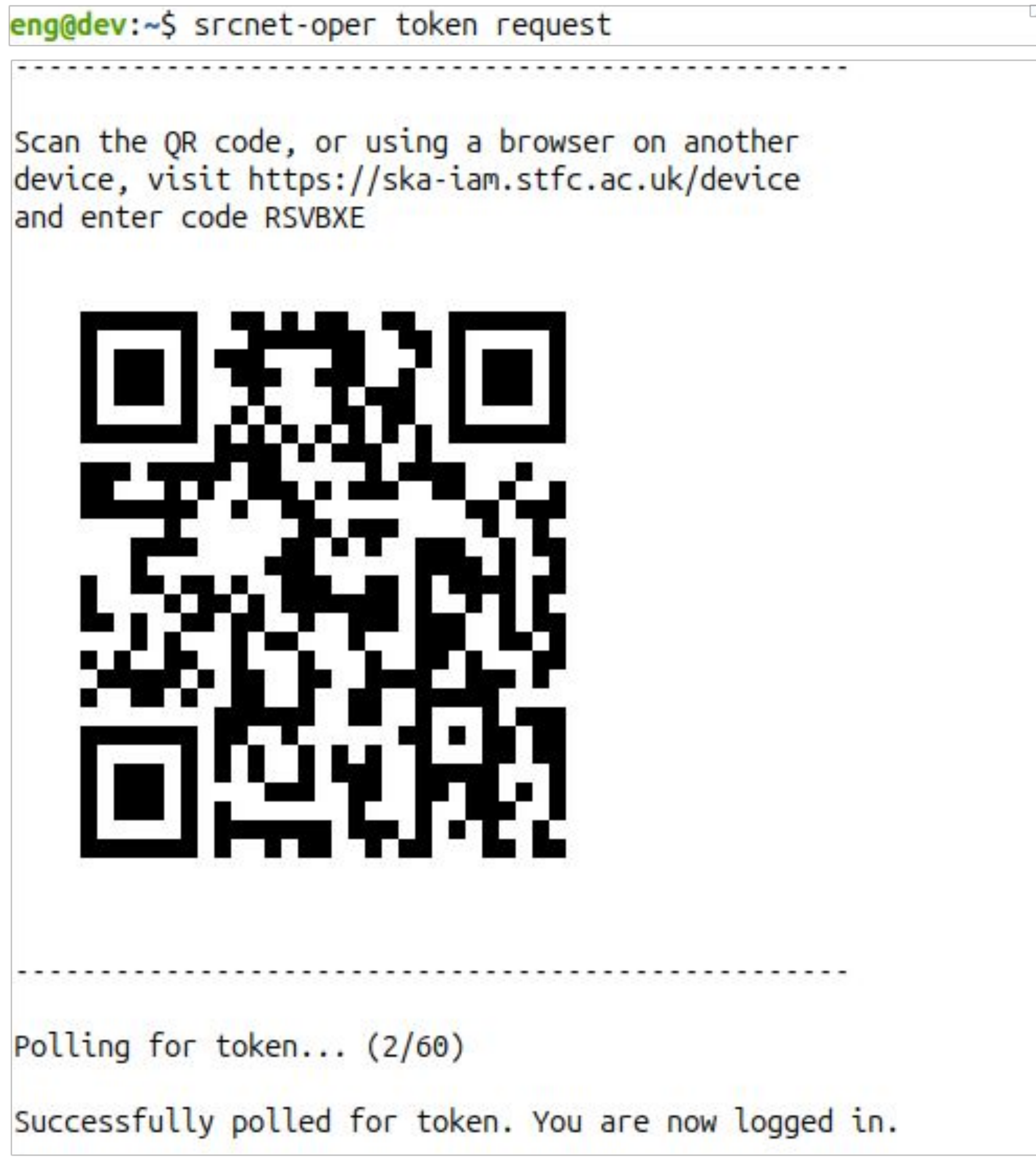
Get data from the datalake given a namespace and name.

```
>>> from astroquery.srcnet import SRCNet
>>> srcnet=SRCNet(verbose=True)
>>> srcnet.get_data(namespace='testing', name='PTF10tce.fits')

>>> INFO: Exchanged authn-api service token for data-management-api service [astroquery.srcnet.c
>>> DEBUG: Access token: <redacted>
>>> DEBUG: Refresh token: <redacted>
>>> DEBUG: Persisting access token to: /tmp/access_token [astroquery.srcnet.core]
>>> DEBUG: Persisting refresh token to: /tmp/refresh_token [astroquery.srcnet.core]
>>> DEBUG: Access token is valid, will not attempt token refresh. [astroquery.srcnet.core]
>>> 8248KB downloaded
```

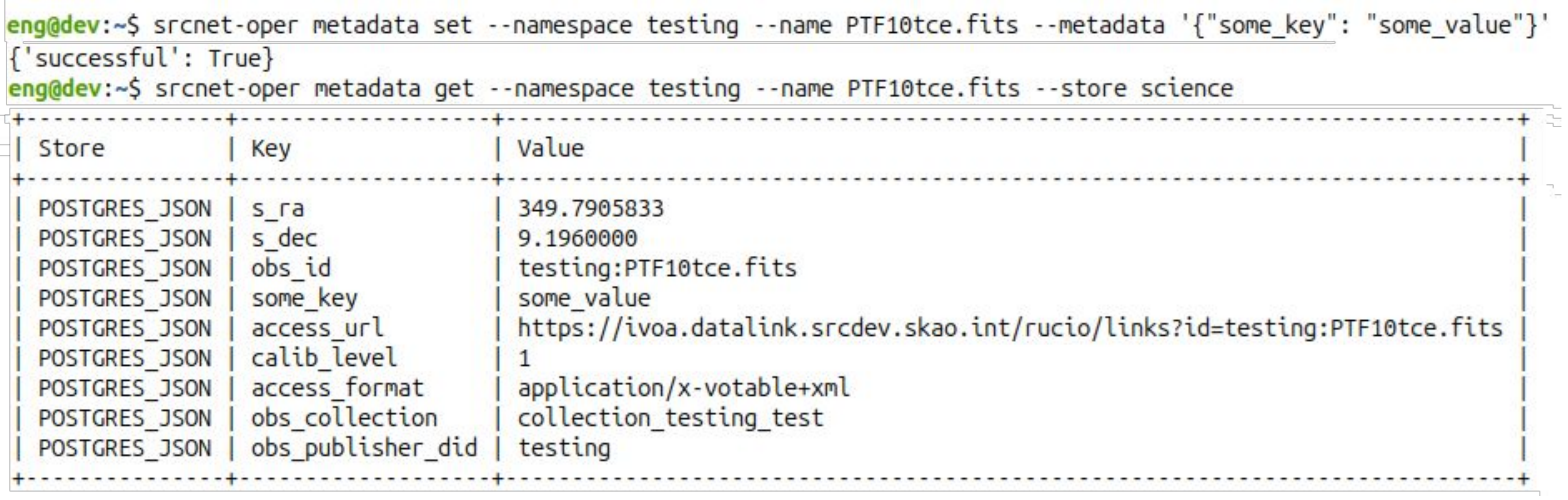
Data Curation and Logistics

srcnet-oper

A command line tool with a focus on **high level admin/operator** flows, e.g.:

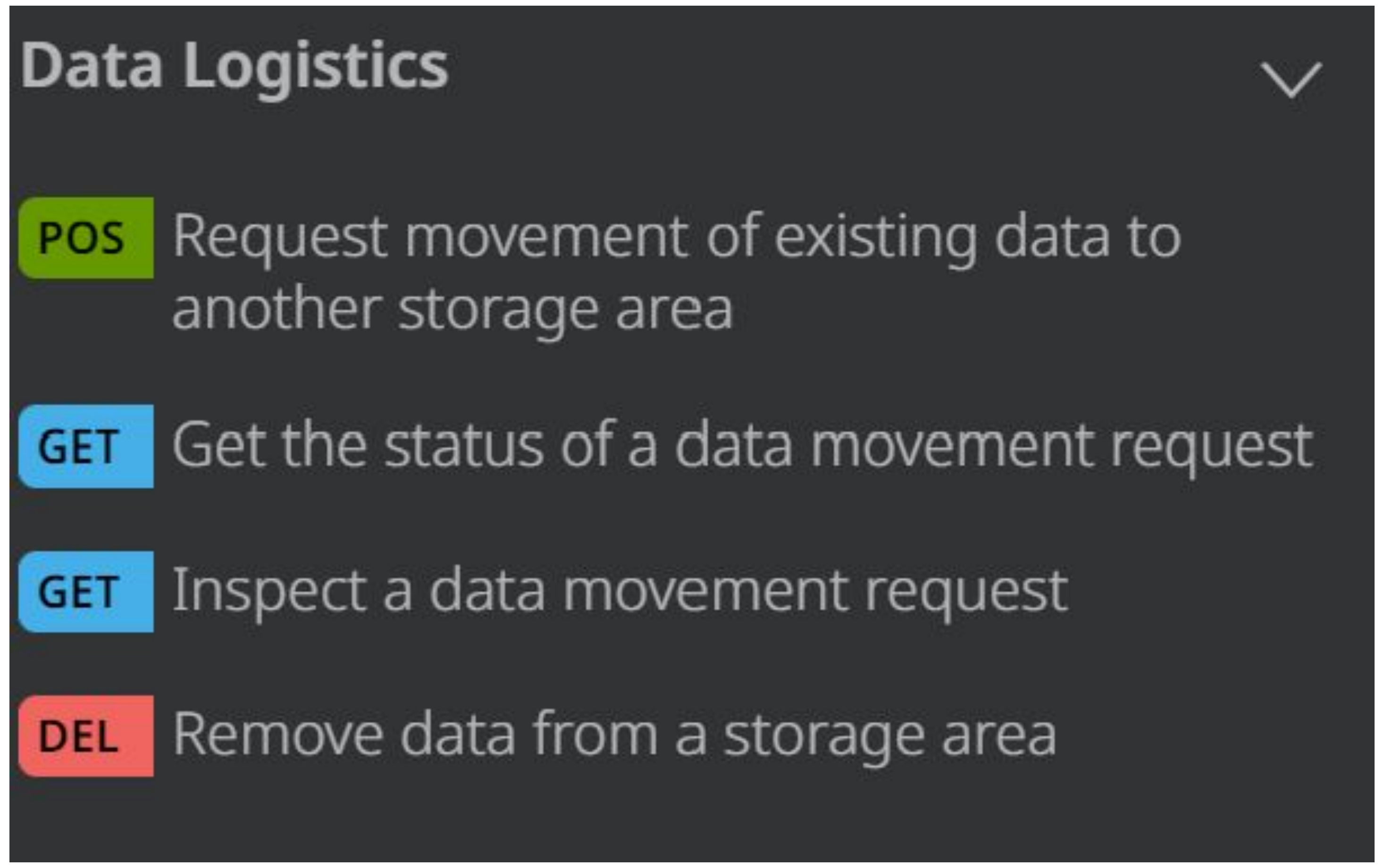


```
eng@dev:~$ srcnet-oper token request
----------------------------------------------------

Scan the QR code, or using a browser on another
device, visit https://ska-iam.stfc.ac.uk/device
and enter code RSVBXE

----------------------------------------------------

Polling for token... (2/60)

Successfully polled for token. You are now logged in.
```



```
eng@dev:~$ srcnet-oper metadata set --namespace testing --name PTF10tce.fits --metadata '{"some_key": "some_value"}'
{'successful': True}
eng@dev:~$ srcnet-oper metadata get --namespace testing --name PTF10tce.fits --store science
+---------------+------------------+-----------------------------------------------------------------------------+
| Store         | Key              | Value                                                                       |
+---------------+------------------+-----------------------------------------------------------------------------+
| POSTGRES_JSON | s_ra             | 349.7905833                                                                 |
| POSTGRES_JSON | s_dec            | 9.1960000                                                                   |
| POSTGRES_JSON | obs_id           | testing:PTF10tce.fits                                                       |
| POSTGRES_JSON | some_key         | some_value                                                                  |
| POSTGRES_JSON | access_url       | https://ivoa.datalink.srcdev.skao.int/rucio/links?id=testing:PTF10tce.fits |
| POSTGRES_JSON | calib_level      | 1                                                                           |
| POSTGRES_JSON | access_format    | application/x-votable+xml                                                   |
| POSTGRES_JSON | obs_collection   | collection_testing_test                                                     |
| POSTGRES_JSON | obs_publisher_did | testing                                                                    |
+---------------+------------------+-----------------------------------------------------------------------------+
```

TBD:



Data Logistics

POS Request movement of existing data to another storage area

GET Get the status of a data movement request

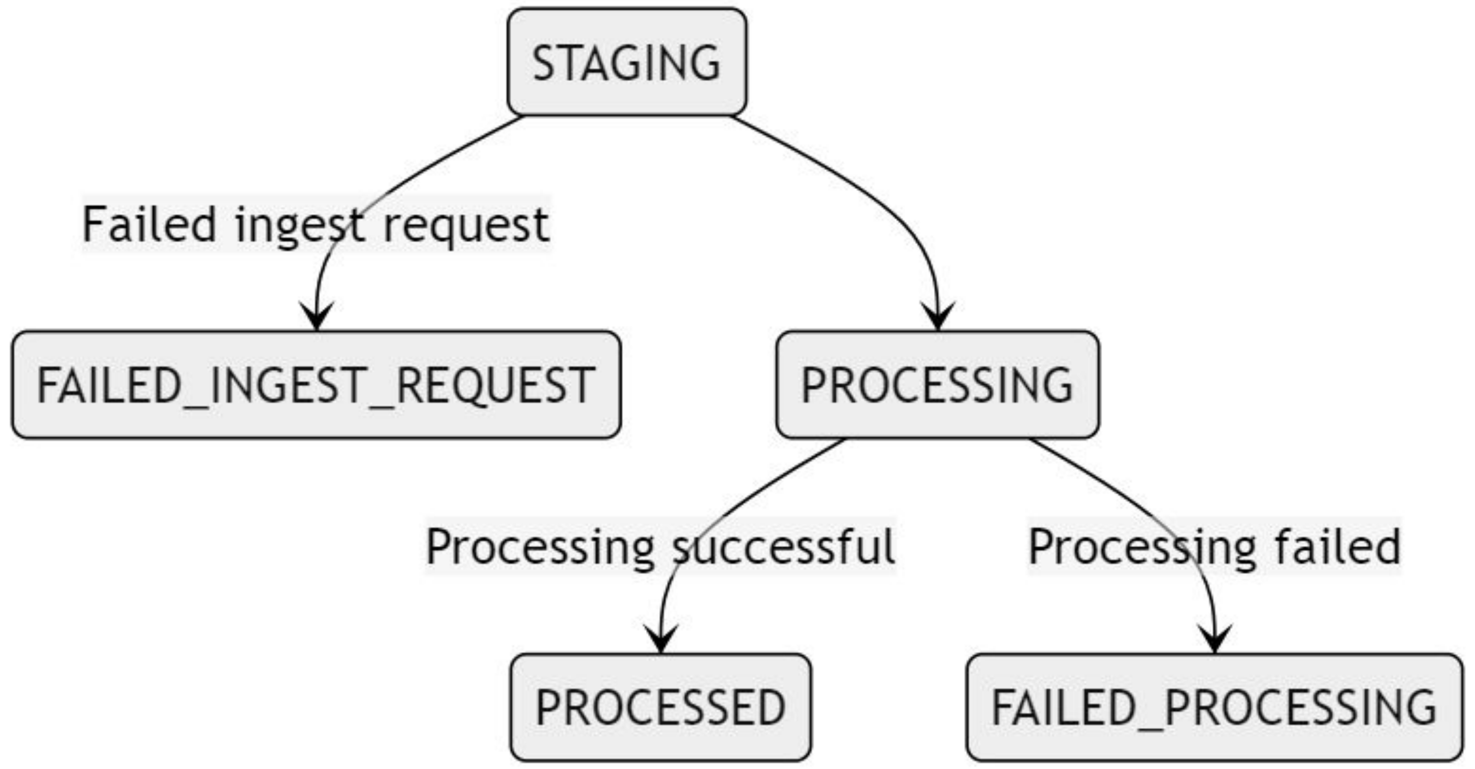
GET Inspect a data movement request

DEL Remove data from a storage area

Data Ingestion

Data Ingestor Service

A service to ingest data products into the datalake



STAGING

Failed ingest request

FAILED_INGEST_REQUEST

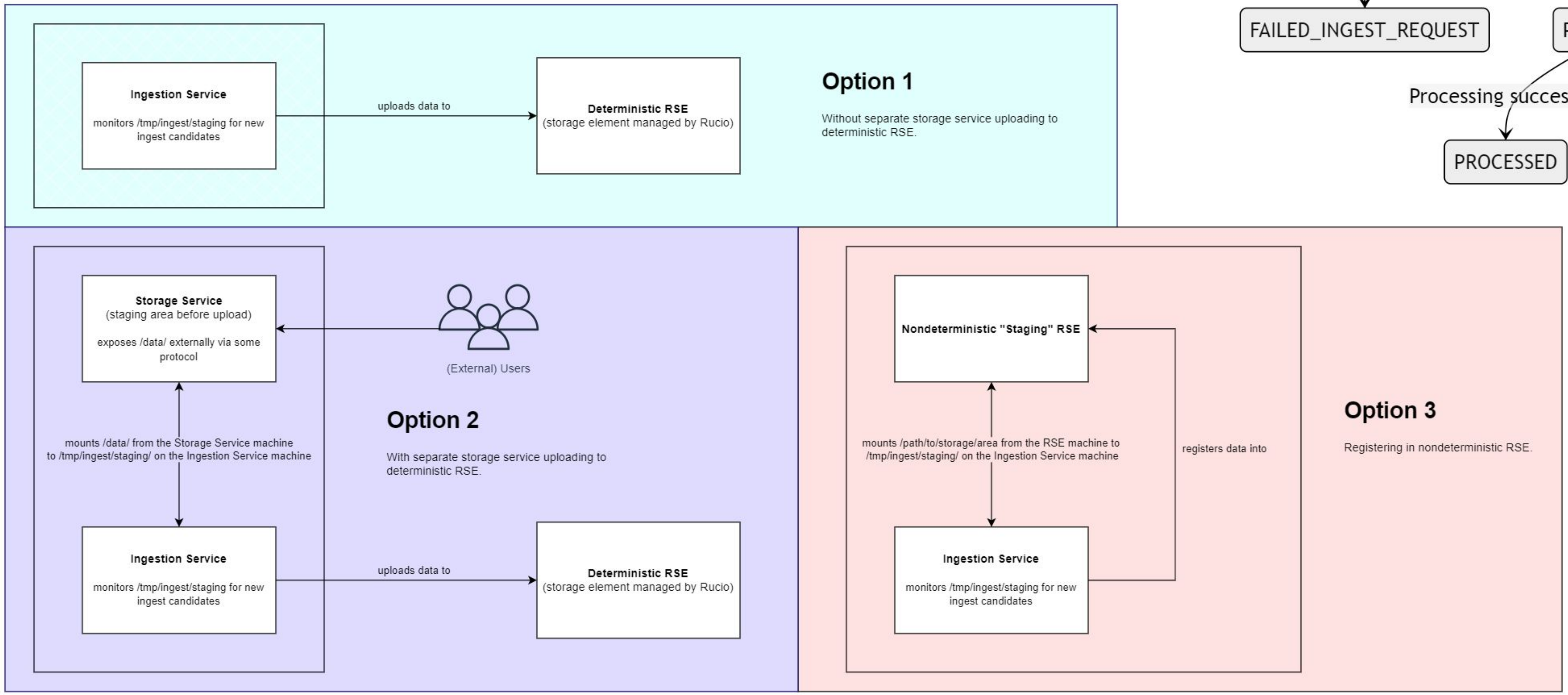
PROCESSING

Processing successful

Processing failed

PROCESSED

FAILED_PROCESSING



Ingestion Service

monitors /tmp/ingest/staging for new ingest candidates

uploads data to

Deterministic RSE
(storage element managed by Rucio)

Option 1

Without separate storage service uploading to deterministic RSE.

Storage Service
(staging area before upload)

exposes /data/ externally via some protocol

(External) Users

mounts /data/ from the Storage Service machine to /tmp/ingest/staging/ on the Ingestion Service machine

Ingestion Service

monitors /tmp/ingest/staging for new ingest candidates

uploads data to

Deterministic RSE
(storage element managed by Rucio)

Option 2

With separate storage service uploading to deterministic RSE.

Nondeterministic "Staging" RSE

mounts /path/to/storage/area from the RSE machine to /tmp/ingest/staging/ on the Ingestion Service machine

registers data into

Ingestion Service

monitors /tmp/ingest/staging for new ingest candidates

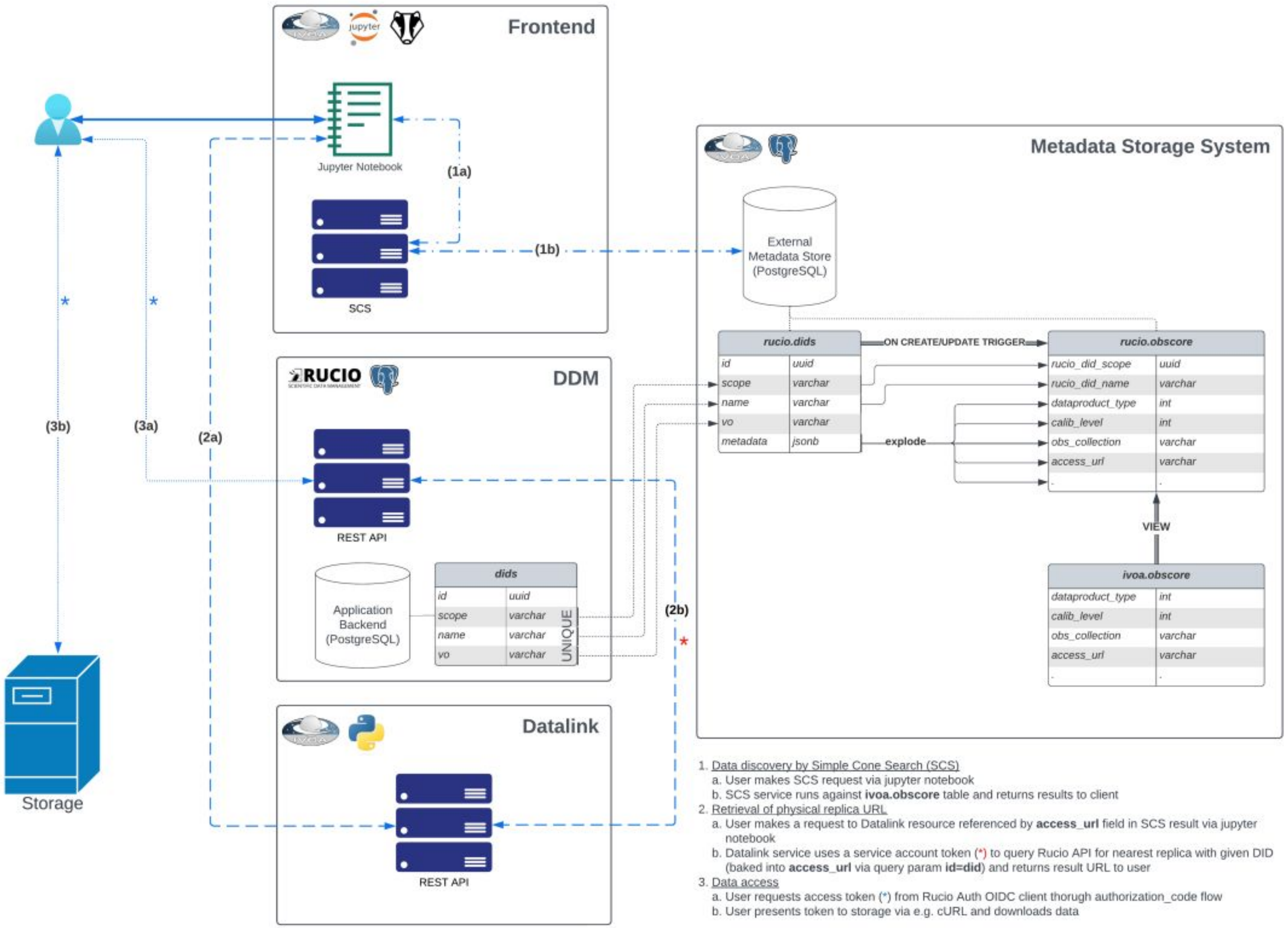
Option 3

Registering in nondeterministic RSE.

Conclusions

- Rucio will form the backbone of the SRCNet v0.1 DDM component
- Functionality will be hidden behind a “Data Management” API hooked into a RBAC Permissions system
- Slowly moving towards meeting the required v0.1 DM functionality
- Rapidly losing more hair (possibly related)
- Thanks!

v0.1 DM Architecture



rucio.dids	
id	uuid
scope	varchar
name	varchar
vo	varchar
metadata	jsonb

rucio.obscore	
rucio_did_scope	uuid
rucio_did_name	varchar
dataproduct_type	int
calib_level	int
obs_collection	varchar
access_url	varchar
.	.

ivoa.obscore	
dataproduct_type	int
calib_level	int
obs_collection	varchar
access_url	varchar
.	.

dids	
id	uuid
scope	varchar
name	varchar
vo	varchar

1. Data discovery by Simple Cone Search (SCS)
   a. User makes SCS request via jupyter notebook
   b. SCS service runs against **ivoa.obscore** table and returns results to client
2. Retrieval of physical replica URL
   a. User makes a request to Datalink resource referenced by **access_url** field in SCS result via jupyter notebook
   b. Datalink service uses a service account token (*) to query Rucio API for nearest replica with given DID (baked into **access_url** via query param **id=did**) and returns result URL to user
3. Data access
   a. User requests access token (*) from Rucio Auth OIDC client thorugh authorization_code flow
   b. User presents token to storage via e.g. cURL and downloads data