C-Scale

Copernicus – eoSC AnaLytics Engine

The Metadata Query Service
Discovering EO data across the federation

Zdeněk Šustr, CESNET
sustr4@cesnet.cz
Service Introduction

MQS (Metadata Query Service)

• Evolved from the original plan to federate Copernicus Data providers within C-SCALE
  • Federation in terms of **access** has been covered in the previous talk
  • **Discovery** across the federation a major goal
  • **Avoid** creating yet another metadata catalogue!

• Main premise: partners already know where their data are
  • Bring their discovery interfaces under a common one
    ▶ single point
    ▶ shared protocol
  • describe their datasets and data retention policies
  • use that to pre-select candidates and redistribute user queries
Datasets and Retention Policies

• Grasp the fact that different partners have different data
  • National archives (full history, limited area)
  • Discipline archives (limited selection of product types, varying retention time)
  • Redistribution services (global coverage, short retention)
  • Big players (ambition to build global archive)

⇒ Not every query needs to be redistributed to every partner
  • Understand the query, select matching providers
  • Currently taking into account only product type
    ▶ Area and time filters not yet applied
Providers’ Catalogue

- Called the *EO Resource Catalogue* initially → confusion
- Has **only** provider information, not data (product) information
  - Partners
  - Contacts
  - Services
  - Endpoints
- Adopting the well known GOC-DB ([https://goc.egi.eu/](https://goc.egi.eu/))
  - The “Grid Configuration Database”, put to new uses
  - Keep track of members and relevant service endpoints
    - [https://goc.egi.eu/gocdbpi/public/?method=get_service_endpoint&scope=C-SCALE](https://goc.egi.eu/gocdbpi/public/?method=get_service_endpoint&scope=C-SCALE)
- Originally also intended for datasets and retention policies, but the choice of protocol changed that
Choosing the Common Protocol

• Chose from those already used in the federation?
  • OpenSearch, OData, CWS, STAC
  • Then implement translation

• **STAC** selected
  • modern protocol
  • lots of products supporting it
  • active community
  • [https://mqs.eodc.eu/stac/v1/](https://mqs.eodc.eu/stac/v1/)

• Side effect!
  • Greater granularity wrt. OpenSearch or CWS
  • Cannot stay completely true to original intention not to build YAMC
    ▶ Required detail simply not available in existing DBs
Choosing the Common Protocol

• Chose from those already used in the federation?
  - OpenSearch, OData, CWS, STAC
• Then implement translation

STAC selected

• modern protocol
• lots of products supporting it
• active community

https://mqs.eodc.eu/stac/v1/

• Side effect!

• Greater granularity wrt. OpenSearch or CWS
• Canot stay completely true to original intention not to build YAMC
  ▶ Required detail simply not available in existing DBs
Choosing the Common Protocol

• Chose from those already used in the federation?
  • OpenSearch, OData, CWS, STAC
  • Then implement translation
• **STAC** selected
  • modern protocol
  • lots of products supporting it
  • active community
  • [https://mqs.eodc.eu/stac/v1/](https://mqs.eodc.eu/stac/v1/)
• **Side effect!**
  • Greater granularity wrt. OpenSearch or CWS
  • Can not stay completely true to original intention not to build YAMC
    ➤ Required detail simply not available in existing DBs
Challenges

• Standardized STAC Collections structure?
  • At least for members who are building new STAC databases, it might make sense to use a common collection structure
  • Developing now

• Paging
  • How to handle item paging when multiple backends respond?
  • Cache and collate own pages? Send more than the query asked for?
Thank you

Questions any time

Zdeněk Šustr, CESNET
sustr4@cesnet.cz