GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

PUBLIC
DOMAIN

SCS2 Prototype Implementation

Markus Demleitner

IVOA 2026 Northern Spring Interop, Strasbourg

SCS2: Implementation Experience

- Recap: What's SCS2 and why?
- What was easy and what was not
- What you can try right now
- Transition Team
- Participate!

Why SCS2?

- We need to get experience managing breaking changes
- ConeSeach-1 has several features that are odd in today's VO
- As a compact and manageable protocol, cone search is a good example
- As the most widely deployed protocol in the IVOA (33615 services as of 2026-06-07), it's an actual exercise: If we can make it there, we'll make it anywhere.

Central Use Case

From the current WD
https://ivoa.net/documents/ConeSearch/20260529/:

> *A data publisher wants to make one or more catalogues accessible to the scientific public* **with minimal effort** *while still satisfying the consumer use case.*

[minimal effort $\approx$ implementation within 1 week on top of something like pyVO]

What's SCS2?

- Remaining: RA, DEC, SR, VERB arguments, VOTable+UCD output
- New: DALI RESPONSEFORMAT, MAXREC
- New: DALI tables and capabilities endpoints
- New: DALI error reporting
- New: In-document registry rules (but almost constant from SimpleDALRegExt; however: mandatory tableset)
- New: Multicone using DALI UPLOAD
- New: SIAPv2-compatible POS
- New: Multi-Table services using TABLE
- New: Case-insensitive parameters, reject unsupported parameters

Implementation

I have done a full implementation of SCS2 with all optional features (but only single-table) in DaCHS $\sim$ 1 week of net work time.

It's available in 2.13.1 or later.

(but: simple co-existence with ConeSearch-1 still TBD)

Hard, But Unavoidable

- Case-sensitive parameters: Surprisingly, this was painful in DaCHS because of common DALI code with protocols that are case-insensitive. *Let's never do case insensitivity again.*
- Rejecting non-supported parameters: Clashed with previous UPLOAD internals.

Both of these items rectify serious legacy flaws. If we break things, let's fix these even if they're painful on top of legacy code.

Surprisingly Hard: UPLOAD

Some of our protocols support multiple queries in one request (e.g., repeatable ID in Datalink).

This is a pain in HTTP parameters. I think *if* we want this (and I generally doubt it), we should UPLOAD parameter sets. SCS2 proposes that.

The upload itself I had, the pain was that Postgres doesn't plan

```
WHERE EXISTS (
  SELECT 1 FROM uploaded_table as u
  WHERE
    in_cone(u.ra, u.dec, queried.ra, queried_dec, u.sr))
```

efficiently. I never quite understood why, but it's hard and I resorted to doing proper joins, which was painful.

Probably Not Worth It: POS

DaCHS lets you write

```
POS=POLYGON 10 15 10 20 15 20
```

This was fairly cheap in DaCHS, as it sits on top of pgshere.

It would probably be much harder to do it on top of just astropy.

Polygons would certainly be cool, but perhaps not cool enough to warrant the complication. Things too large to pull fully in a pinch should be in TAP anyway.

Try It

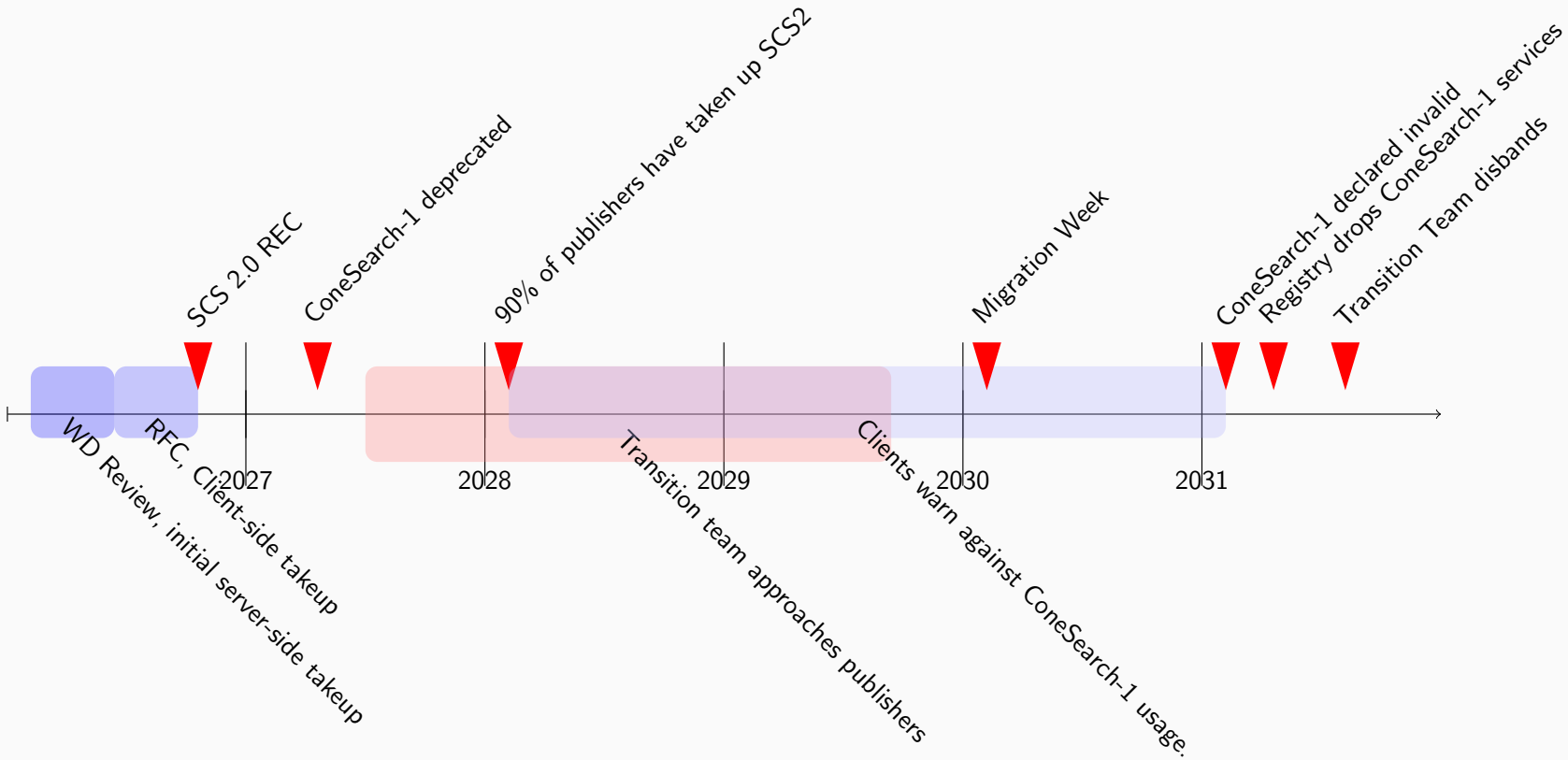
There's SCS2 on GAVO's Gaia DR3 lite service:
http://dc.g-vo.org/gaia/q3/cone2/scs2.xml?.

Discover this and perhaps more:

```
select access_url, res_description
from rr.resource
natural join rr.capability
natural join rr.interface
where standard_id like 'ivo://ivoa.net/std/scs2#query-2.0'
```

Next Steps

My ambitious (and already obsolete) transition plan from late 2025:



Transition Team

This whole thing will only work if there are people on the transition team that actually want to make that happen.

Me, I'll just plough on with this if I believe such a transition team will form.

So: Pledges?

Stelios' Issues I

Stelios has put in a list of great questions into a github bug. I'll post some of them on the DAL list in the coming weeks:

- Non-VOTable responses for errors?
- Drop VERB?
- Be explicit about POST?
- Actually require ConeSearch-1 for the transition period?
- Exactly one of Cone, POS, and Upload is painful for static validation

Stelios' Issues II

- Relax requirement to call the query endpoint /scs2?
- Relax requirement that the identifier is a string?
- Explicitly split search details and wire details in the spec?