Simple Cone Search 2 and Transitioning To It

Markus Demleitner

Astro-CC Technology Forum 1, 2025-11-08

Cone Search 2

- The Use Case
- What stays
- What's new
- How do we get rid of ConeSearch-1?

SCS2 Use Case

Why have an S-protocol for catalogue searches at all when we have TAP?

> 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.

. . . and the consumer wants to easily retrieve an astronomical catalogue or reasonable subsets of it.

So: An operator should be able to write and register an SCS2 service from scratch within, say, a week assuming a pre-existing VOTable library.

Side note: What would be useful for that is a piece XSLT turning a VOTable response into a VODataService tableset.

What's Old

Input parameters: RA, DEC, SR, VERB

Output "data model": `meta.id;meta.main`, `pos.eq.(ra|dec);meta.main`

Alternatives:

- SIAP2-style POS on input? This would be kind-of nice for symmetry, but it would mean requiring polygon support, and that would no longer be simple at all.
- "DM" markup using utypes? This would be simple, but I have been unable to find a convincing benefit.
- Drop VERB? Have you met someone who actually liked it? Don't we have enough bandwidth to always return all columns these days?

What's New: DALI Compliance

- DALI error messages
- Mandatory VOSI capabilities and tables
- Mandatory `scs2` endpoint name
- Mandatory `RESPONSEFORMAT` support (but that is trivially implementable as only VOTable output is required)
- Mandatory `MAXREC` support

What's New: Input Parameters

- Encourage custom constraints using DALI-interval (and their declaration in the VOResource interface)
- Optional SIAP2 POS – this barely makes my cost-benefit threshold, but I suggest we should think about it
- Optional UPLOAD support: A table with RA, DEC, SRC columns for multicone – that sounds more complicated than it is, and multicone seems really useful to me.

What's New: The TABLE parameter

I'd like to have *the option* to have site-wide SCS2 services that work as auxiliary capability for data-like resource records.

Solution: An accept-mandatory TABLE parameter, pass-optional for single-table services, accepting names from the tableset.

The data-like records would come with a tableset; problem: how do we say which tables are eligible for SCS2?

The current draft has nothing on that. I think the solution should be a capability type for auxiliary capalities with pertainingTable element(s). That might be useful for TAP and Obscore, too.

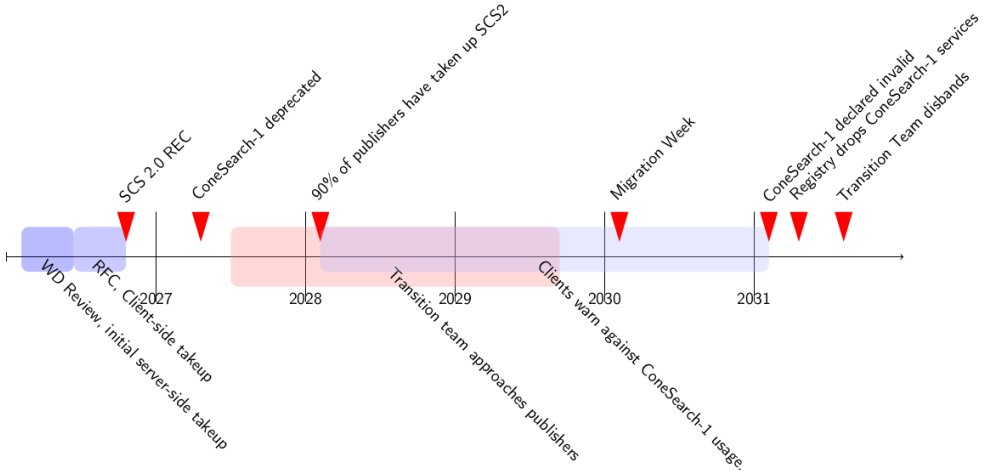
Much as I would like this, I do wonder if it is worth the effort; this mainly depends on how valuable we feel the custom parameters are. Once we want those, a global cone search service will not cut it *for them*.

Transition Plan

From where I stand, the main purpose of SCS2 is as an exercise for major version transitions.

Goal: At the end of the transition, no ConeSearch-1 services are left in the VO. Here's a sketch of how this might work "if we all pull together as a team"[1]:

[1] Pink Floyd reference intended



Implementation Status

I've not written any code yet.

I'll do that if there's not too much strife about this at the Interop.

It really shouldn't take longer than a week within DaCHS, UPLOAD and POS included.

What about you?