e-inf-astro

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

PUBLIC DOMAIN

SCS2 Towards Proposed Recommendation

Markus Demleitner

IVOA Perth Interop, 2026-11-07

What happened so far?

At the Strasbourg Interop, I have presented a Working Draft for SCS2 (below: “WD1”)

There was substantial feedback.

A second WD is out now, and it’s a different beast.

Well: That’s what we do breaking changes for.

Here is a rundown of the changes. Protest now if you feel this shouldn’t be in the PR.

Dropping UPLOAD and POS parameters

WD1 had a SIAv2-like POS parameter, and it did multicone via an UPLOAD of query parameters.

POS might have allowed polygons, which would definitely not have been simple any more.

UPLOAD was a pain in implementation, and we have a new plan for multicone.

So, both are gone.

CIRCLE instead of RA, DEC, SR

Instead of POS, we now use a SIAv2-style CIRCLE.

And it's repeatable, so we get multicone for (spec-wise) free.

Implementors can set a limit for the number of CIRCLEs; they can and should declare that in a new maxCone item in the registry extension [prefer maxCones?].

With CIRCLE, the venerable RA, DEC, SR parameters have no function any more. So, they're gone.

Error messages are now plain text rather than VOTable

This is *machine-readable code:human-readable message* similar to SIAv2 and Datalink (but SIAv2 still has a VOTable/query status armor).

I'd have liked to see that in DALI, but there's no harm prototyping it in SCS2 first.

Main id can be of any type now

SCS1 and WD1 wanted the main id to be strings.

It's nice to have static types, but the killer argument was that it sucks if you pull Gaia ids (native bigints) and do a TAP upload of the result: The comparisons would fail.

More liberal language on SCS1 backwards compatiblity

WD1 wanted SCS1 capabilities along SCS2 even for new SCS2 services coming online during the transition period.

We still *want* that but toned down the language.

At least nothing breaks that has been working before while transition lasts.

Deprecating cs:TestQuery in the registry extension

SimpleDALRegExt has halfway abstract ways of specifying test queries for the benefit of validators.

That was a lot of effort for constructing query strings.

Let's not do that any more and instead use vs:ParamHTTP's testQuery item directly.

Next steps

- Update migration plan to start a year later
- Get a few more co-authors on board (what about you?)
- Push out a PR
- Go for RFC
- Form a transition team (Join the fun! Talk to me!)



(obligatory AI pun: This is stablediffusion's idea of "Join the Fun! Talk to me!")