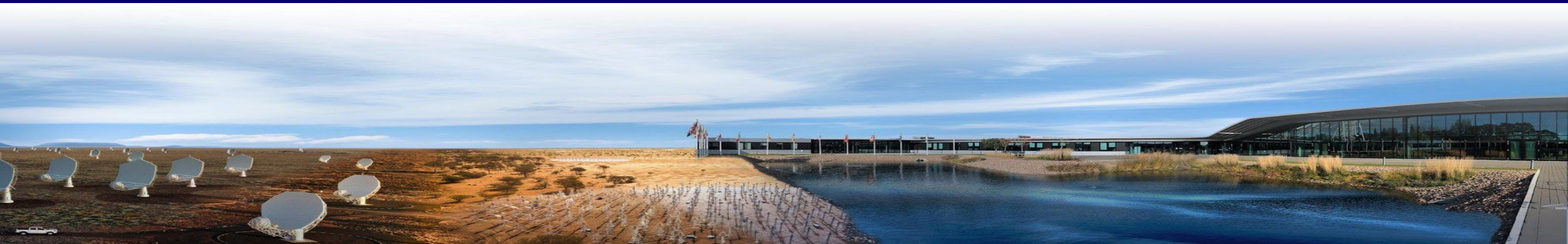
IVOA INTEROP · DSP SESSION · AUTHVO

AuthVO bearer tokens: aligning with RFC 9728

What the new working draft says, what the reference clients measured, and how a patched INDIGO IAM would make the whole thing simpler

Jesús Salgado · SRCNet Software Architect and Development Lead · SKA Observatory

No OAuth background needed: the first slides explain the problem and the vocabulary in plain words; the technical names are given alongside.



AGENDA

What this update covers

1. **The problem, in plain words**
   Why a client must check a service before logging in, and the six terms used in the rest of the talk

2. **What the draft now says**
   PR #23: before and after, and what the review decided

3. **The reference clients**
   authvo-clients: what it tests, and the results

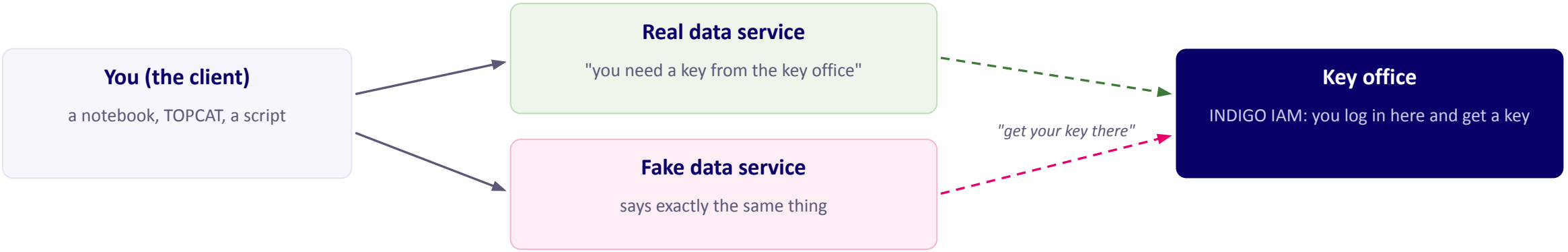
4. **INDIGO IAM: released versus patched**
   What the IAM does today, what a patched IAM changes, and what we learned running both

5. **Next steps**
   What we ask the group, and what SKAO will do

THE PROBLEM · IN PLAIN WORDS

A fake service can trick you into fetching a real key for it

Think of a token as a key, and of the identity provider (INDIGO IAM) as the key office that issues keys after you log in.



The risk

Both services point to the same, real key office. If you simply obey, you log in, get a genuine key and hand it to the fake service, which can then use it on the real services as if it were you.

The fix in AuthVO

Before logging in, ask the key office itself: "do you issue keys for this service?" The key office publishes a list of the services it serves. Not on the list: stop, no login, no key.

The rest of the talk is about how the client does that check, who publishes the list, and what is printed on the key.

VOCABULARY

Six terms, in plain words and in OAuth words

The rest of the slides use the plain term first and the technical name in grey.

Key

access token (bearer token)

What the client sends with each request. Whoever holds it can use it, so it must only go to the right services.

Key office

authorization server: INDIGO IAM

Where the user logs in and the client gets keys. It also publishes an information page about itself.

Door sign

Protected Resource Metadata (PRM), RFC 9728

A small document each service publishes: "I am this exact address, and my keys come from that key office".

The key office's list

protected_resources, in its RFC 8414 information page

The services the key office vouches for. The client checks it before login. No list means no login.

Written on the key

audience (aud), requested with resource (RFC 8707)

Which services the key opens. A key for one service is useless at another, so a leaked key does less harm.

The add-on

metadata proxy ("augmenter")

A small program in front of today's INDIGO IAM that adds the list, because the released IAM does not publish it yet.

THE FLOW · AS IN THE MERGED DRAFT

Five steps; the user only logs in after step 3

A client that starts from nothing but a URL. Each check blocks one specific trick.

1 Knock

Ask the service for data. It answers "no key: read my door sign".

401 + resource_metadata

2 Read the sign

The sign must name exactly the address you asked for. Redirects are not followed.

PRM resource = URL

3 Ask the key office

Is the service on the key office's list? If not, stop here, before any login.

protected_resources

4 Get the key

Log in, and ask for a key written for this service, or for every listed service.

device grant + resource

5 Use the key

The service checks the key is genuine, current, has the right permission, and is meant for it.

iss, scope, exp, aud

WHICH STEP STOPS WHICH TRICK

The trick	Stopped at	How
A fake service names the real key office	Step 3	It is not on the key office's list
A service shows a sign written for another address	Step 2	The address on the sign must match exactly
A sign that redirects somewhere else	Step 2	Redirects are never followed
A key stolen from one service and used at another	Step 5	The key says which services it is for

STATUS · NOVEMBER 2026

The text is agreed; the key office is now the open item

Three strands of work. The specification and the clients are done; what remains is getting the list into INDIGO IAM itself.

The AuthVO draft

Done

PR #23 approved and merged after two review rounds. Working draft dated 2026-10-29 ready for the IVOA document repository.

The reference clients

Done

Python and Java clients plus a test lab. They now run against two set-ups: today's IAM with the add-on (22 of 22 checks pass) and a patched IAM on its own (20 of 20).

INDIGO IAM

Open

The released IAM (1.15) still does not publish the list, so we need the add-on. A patched IAM in my fork publishes it itself and passes the same tests. Not yet proposed upstream.

Two asks for this session: publish the working draft for wider review, and agree that INDIGO IAM should publish the list natively, starting from the patched version.

Before and after, in plain words

The previous draft used RFC 9728 names but not its rules, so standard OAuth libraries would have needed VO-specific changes. The new text follows the RFC, plus one VO rule: the list.

Question	Before (PR #10)	Now (PR #23, merged)	What it means
Where is the door sign?	On the key office's server	On the service itself, as RFC 9728 says	Standard libraries work unchanged
How must the sign match?	It only had to "cover" the address	Exactly the address asked for, query included	Services with changing addresses (TAP jobs, DataLink) write the sign per request
Who vouches for the service?	Being on the same host as the key office	The key office's list (protected_resources)	Services can live anywhere; the key office decides
If there is no list?	Not covered	The client stops: no list, no login	Safe by default
What is written on the key?	One service, mandatory	At least the service asked about; optionally every listed service	The client chooses: fewer logins, or less harm if a key leaks

Also in PR #23: corrections against the RFCs it cites (error codes, token-validation wording, refresh tokens, references). None changes the design. Timeline: issue 14 Oct, PR 23 Oct, merged end of October.

PR #23 · REVIEW OUTCOME

Two review rounds, one compromise, approved and merged

The reviewer asked us to stay closer to RFC 9728. Every point was adopted; the question of what to write on the key was settled by agreement.

1	**Exact match on the sign**	Adopted	The sign must name exactly the address asked for (RFC 9728 §3.3). Our extra check is no longer needed.
2	**"Covering" only inside the list**	Adopted	A list entry also covers the addresses below it; together they form the service's domain.
3	**What to write on the key**	Compromise	SHOULD: at least the service asked about. MAY: every listed service. Fewer logins versus less harm if leaked.
4	**Wording and examples**	Adopted	Plural wording; the examples list two services, only one of which covers the address asked for.
5	**Second round: four edits**	Adopted	One repeated sentence and one overstatement removed; any key office that passes both checks may be used.

Open for Perth: the reviewer finds the text longer than it needs to be, repeating RFC content instead of pointing to it. Worth a group decision before the next draft.

THE REFERENCE CLIENTS · AUTHVO-CLIENTS

A test lab that tries both honest and dishonest services

Open source: github.com/jesusjuansalgado/authvo-clients. Everything runs on one laptop with Docker.

1 **Two clients**

Python and Java, behaving identically. They do the five steps and stop when a check fails.

2 **A real key office**

INDIGO IAM, either the released 1.15 with the add-on, or the patched version on its own.

3 **An honest and a fake service**

The honest one writes its door sign for each request; the fake one names the real key office.

4 **Nine scenarios**

The normal login, plus each trick: fake service, missing list, wrong sign, redirect, changing addresses.

22 / 22

checks pass with today's IAM and the add-on

20 / 20

checks pass with the patched IAM alone

0

changes to the clients to switch between them

1

command to switch: ./setup.sh --native

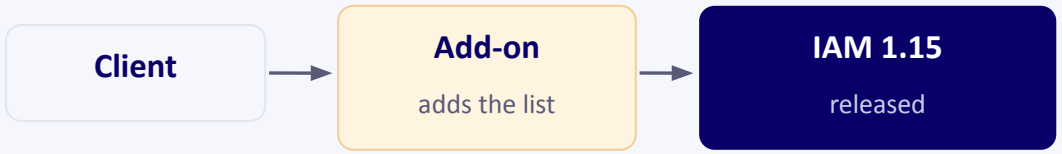
Why 20 and not 22: one scenario removes the list from the add-on's file. The patched IAM always lists at least itself, so that case cannot be staged there; a list that does not include the service is still tested.

NEW THIS WEEK · AUTHVO-CLIENTS PR #8 TO #10

The same tests, run against two key offices

Same clients, same honest and fake services, same scenarios. Only the key office changes.

Today: released IAM + add-on



- The list lives in the add-on's file
- Two components to run and keep in step
- Command: ./setup.sh

22 of 22 checks pass

Proposed: patched IAM



- The list lives in the IAM's own configuration
- One component; the add-on disappears
- Command: ./setup.sh --native

20 of 20 checks pass

Both set-ups are kept on purpose: the add-on is what SRCNet can deploy now; the patched IAM is what we want to propose to INDIGO, and the tests prove the clients need no change when it arrives.

INDIGO IAM · MEASURED WITH THE TEST LAB

What each key office does when asked for a key

The client asks for a key and says which services it is for. Green is what AuthVO needs.

The client asks for a key for…	IAM 1.10	IAM 1.15 (released)	IAM patched (fork)
one listed service	Key, nothing written on it	**Key for that service**	**Key for that service**
two listed services at once	Key, nothing written on it	Key for the first one only	**Key for both**
a service that is not on the list	Key, nothing written on it	Key for it anyway	**Refused (invalid_target)***
an address with a query (?ID=…)	Accepted, ignored	**Refused (invalid_target)**	**Refused, before login**
…and does the key office publish the list?	No	No: the add-on is needed	**Yes, in both information pages**

** With the setting iam.resource-indicators.restrict-to-protected-resources=true. It is off by default in the patch; the test lab turns it on.*

In short: the released IAM is half-way there. The patched IAM does everything the draft asks of a key office, so the add-on could be retired.

RELEASED VERSUS PATCHED · WHO NOTICES WHAT

What changes for users, clients and operators

Measured by fetching and comparing the key office's information pages in both set-ups, and by running all scenarios.

	Today (IAM 1.15 + add-on)	Patched IAM	Who notices
Information pages	Same content	Same content, plus the IAM's own address in the list	Nobody: clients behave the same
One key for the whole domain	Not possible: only the first service is written	**Works: every listed service is written**	Users: fewer logins
Asking for an unlisted service	A key is issued	**Refused by the key office (if switched on)**	Nobody, unless something is wrong
Editing the list	Takes effect at once (file re-read each time)	Needs an IAM restart (read at start-up)	Operators
Components to run	IAM + add-on + routing rules	IAM only	Operators

Bottom line: clients see no difference, users get the domain-wide key, operators run one component less but must restart the IAM to change the list.

What our fork adds to INDIGO IAM

Each row is something the released IAM (1.15) does not do, what the fork adds, why AuthVO needs it, and the standard it implements. The first three replace the add-on.

Missing in IAM 1.15	Added in the fork	Why it matters	Standard	Switch
The RFC 8414 information page	Also served at /.well-known/oauth-authorization-server	Step 3 reads the key office's page from there	RFC 8414 §3	Always on
The key office's list	protected_resources in both information pages: the configured services plus the IAM itself	Step 3: the check before login. No add-on needed	RFC 9728 §4	Config list
Several services on one key	Every requested service is checked and written on the key (aud), not only the first	The domain-wide key the draft allows (MAY)	RFC 8707 §2	Always on
Refusing unlisted services	invalid_target for any value not exactly in the list	A wrong or malicious request stops at the key office	RFC 8707 §2	Opt-in
A door sign for the IAM's own APIs	/.well-known/oauth-protected-resource, and 401 answers point to it	The IAM's own APIs follow RFC 9728 like any service	RFC 9728 §3, §5.1	Always on
Door signs for other services	Hosted at /protected-resource-metadata/{id}, checked at start-up	Helps fixed-address services; not used by AuthVO's flow	RFC 9728 §2, §5	Optional
An audience check on the IAM's APIs	Keys not written for the IAM are rejected; keys with no audience still work	Stops a domain-wide key from opening the IAM's own APIs	RFC 9068 §4, RFC 7519 §4.1.3	Opt-in
Correct error handling	invalid_target returned to the client with its state; device login checks resources at the start	Clients get a clear error instead of an error page	RFC 6749 §4.1.2.1, RFC 8628 §3.1	Bug fix

Size: four commits on INDIGO's develop branch, 31 files, about 1,700 lines, half of them tests. Opt-in settings, off by default: restrict-to-protected-resources and require-iam-audience. The RFC 9728 §4 list is optional in the RFC; AuthVO makes it mandatory.

WHAT WE LEARNED RUNNING BOTH

Four findings worth raising with INDIGO

None blocks the proposal; two of them should be part of it.

The key office lists itself

Raise upstream

The patched IAM always puts its own address in the list. So a whole-domain key also opens the IAM's own management interfaces, unless the IAM also checks what is written on keys (require-iam-audience).

Editing the list needs a restart

Raise upstream

The add-on re-reads its file on every request; the IAM reads its configuration once. Fine for a list that rarely changes, but worth stating.

Newer login pages need a CSRF token

Handled

The fork is based on INDIGO's development branch, whose login forms require the token a browser sends anyway. The test scripts now send it too. No effect on users.

Other differences are not ours

Noted

Old options (implicit grant, 'none' signatures) are gone in the newer code. One permission (vo.read) is missing from the advertised list, cause unknown, but keys with it work.

Still to settle: the IAM can also host door signs for services, but only one fixed sign per service. Services whose addresses change (TAP jobs, DataLink) must keep writing their own sign per request.

The patch: what it does and what is left

Branch rfc9728-protected-resource-metadata on my fork of indigo-iam/iam, based on its develop branch: four commits, with tests. Not yet proposed upstream.

DONE

Publishes the list	From one list in its configuration, in both information pages (RFC 8414 and OpenID)
Writes every requested service on the key	So one key can cover the whole domain
Refuses unlisted services (opt-in)	invalid_target, returned to the client
Door signs for its own APIs	And, optionally, hosted signs for other services
Checks keys used on its own APIs (opt-in)	A key not written for the IAM is rejected
Tested with authvo-clients	Both clients, all scenarios: 20 of 20

LEFT TO DO

1. **Choose the defaults**
   Both new checks are off by default; agree what INDIGO should switch on
2. **Add the two findings**
   The IAM in its own list, and list changes needing a restart
3. **Open the upstream PR**
   Linked to discussion #1077 and to the AuthVO draft

The idea: one list in the IAM's configuration feeds the information pages and the refusal of unlisted services, so they can never disagree, and the add-on can be retired.

PLANS

Next steps

Items 1 and 2 need the group; the others are under way at SKAO.

1 Publish the working draft

Upload WD 2026-10-29 to the IVOA document repository for wider review.

2 Decide the document's length

Agree in Perth how much RFC text the draft should repeat, before the next version.

3 Settle the patch

Choose its defaults and address the two findings: the IAM in its own list, and restarts.

4 Propose it to INDIGO IAM

Open an upstream pull request, linked to discussion #1077 and to the AuthVO draft.

5 SRCNet: deploy now

INDIGO IAM 1.15 with the add-on, the set-up that passes 22 of 22 today.

6 SRCNet: simplify later

Drop the add-on once a released INDIGO IAM publishes the list itself.

Feedback welcome: on the draft in ivoa-std/AuthVO, on github.com/jesusjuansalgado/authvo-clients, and on the INDIGO IAM patch before it goes upstream.

Thank you

Questions and discussion: publishing the draft, its length, and the list in INDIGO IAM itself

Jesús Salgado · SRCNet Software Architect and Development Lead · SKA Observatory

