

# AuthVO Status

Mark Taylor (Bristol)

IVOA Interop  
Görlitz

Distributed Services and Protocols WG

15 November 2025

`$Id: authvo.tex,v 1.10 2025/11/05 14:46:08 mbt Exp $`

# Outline

- AuthVO Document
  - Content/scope recap
  - Current status
- OAuth 2.0 in AuthVO
  - Brief introduction

# AuthVO Document Content

## Aim:

*“This document explains how VO services can manage the authentication process for interoperability with clients, especially non-browser clients<sup>\*</sup>. Particularly, this document describes [how services advertise their support of specific authentication schemes](#) and [how clients can discover and use this information](#) to access protected resources.”*

<sup>\*</sup> “Non-browser clients”: e.g. Python, TOPCAT, Aladin, ...

*“Authenticated services in the VO are not expected to change their authentication frameworks to any “VO-sanctioned” technology, but by implementing the proposals here they can become usable in a broader range of scenarios.”*

## Content:

- Explains use of **WWW-Authenticate** challenges to advertise supported authentication
- Lists/defines recommended authentication schemes:
  - ▷ **Basic** (RFC 7617)
  - ▷ **ivoa\_cookie** (defined here)
  - ▷ **ivoa\_x509** (defined here)
- Describes behaviour for VOSI/non-VOSI services with mandatory/optional/no authentication
  - These parts are implemented in production services/clients (ESDC, CADDC, DaCHS; TOPCAT)

For more details, see [presentation from last Interop](#)

# AuthVO Document Status

## Status:

- Still in early draft
- Under development on github:

<https://github.com/ivoa-std/AuthVO>

# AuthVO Current Activity

## Changes since College Park:

- Document renamed ~~IAP (Interoperable Authentication Protocol)~~  
→ **AuthVO** (Authentication in the VO)
- Adjustments to **ivoa\_x509** scheme:
  - ▷ Clarify that an existing certificate (from another CA) may be used
  - ▷ Add “unparameterised” variant: certificates OK, no issuer endpoint offered

## Outstanding issue: **How to deal with OAuth 2.0?**

- Many services use OAuth 2.0 for authentication
- Extensive discussion: [issue #6](#), [PR #10](#), [PR #18](#), private emails etc
- Consensus on rough approach:
  - ▷ Align with existing OAuth 2.0 standards where possible
- ... but not yet on all details.

# OAuth 2.0 Requirements

Requirements for non-browser clients to use OAuth 2.0 in the VO:

- Client needs to acquire an **Access Token** (Bearer Token), and maybe a Refresh Token etc too
- Client needs to know which resources this token can be used to access
  - ▷ This information must be from a trusted source (Authorization Server not Resource Server) to avoid token disclosure  
— See [RFC 9700 Section 4.9.1](#) “Access Token Phishing by Counterfeit Resource Server”

Why is OAuth 2.0 hard for VO clients?

- Most of OAuth 2.0 standards landscape assumes non-VO-like scenarios:
  - ▷ typical OAuth 2.0 clients are web-based, have out-of-band knowledge about services
- Non-browser client code can't contain secrets (user has access in principle to all client code)
- Non-browser clients are generally not trusted with user credentials
- OAuth 2.0 is large and complicated
- Services have many options within OAuth 2.0 and may not support all
- Security pitfalls abound

# OAuth 2.0 Progress

## Agreed?

- Use [RFC 6749](#) and [RFC 6750](#) for basic OAuth 2.0 operations
- Use [RFC 8414](#) Authorization Server Metadata
- Use [RFC 8628](#) Device Authorization Grant for non-browser clients

## Not agreed?

- Custom WWW-Authenticate `ivwa_bearer` scheme vs. [RFC 9728](#) `resource_metadata` parameter to locate Authorization Server Metadata
- Dynamic ([RFC 7591](#)) vs. static client registration
- How to determine which resources token can be used for: [RFC 9207](#)? IVOA Registry records?
- How much detail to include in text: just list RFCs? detailed examples?

## Experimental implementation

- Parts of flow involving `ivwa_bearer`, [RFC 6749](#), [RFC 6750](#), [RFC 8414](#), [RFC 7591](#), [RFC 8628](#) implemented in prototype between TOPCAT and SKAO

*“Good heavens there are a lot of OAuth RFCs.”*  
— Russ Allbery [AuthVO PR#18](#)

## OpenID Connect

- OIDC is an authentication layer on top of OAuth 2.0
    - ⇒ OIDC use in the VO can be layered on top of AuthVO??
    - ⇒ OIDC doesn't need much discussion in the AuthVO document???
- (but I don't understand this topic very well)*

## Next Steps

My preference: more experimentation with OAuth 2.0

- Come up with a proposal (PR or informal text)
- Implement in a service
- Try it out with a client (I volunteer TOPCAT)
- See if other services/clients can/will do the same thing
  - ... If successful update normative/example text in AuthVO Doc
  - ... Otherwise iterate