

OAuth 2.0 Authority Metadata

Describing the authority behind scopes and RAR types

`draft-zehavi-oauth-authority-metadata-01`

Yaron Zehavi · Raiffeisen Bank International · IETF 127 OAuth WG · San Francisco

Photo: Golden Gate Bridge from Battery East, Frank Schulenburg, CC BY-SA 4.0, via Wikimedia Commons

The problem: identifiers are not semantics

- scopes_supported and authorization_details_types_supported advertise **names**, not authority
  - What operations, resources, and effects does a scope or RAR type actually permit?
- Example: payments.manage - read? create? execute? delete? The name alone does not say
- This matters when an AS or PDP must decide whether requested authority:
  - is permitted by an existing grant or local policy
  - aligns with an approved task, purpose, or Mission
  - is broader than necessary, or has a lower-authority alternative
  - requires particular token presentation or authentication properties

> Agents, token exchange and assertion grants make this acute: the issuer often is not the party that defined the scope.

Classification: GENERAL

What the draft defines - and what it does not

Defines

- authority_metadata parameter for Protected Resource Metadata (RFC 9728) and AS Metadata (RFC 8414)
- One authority description model for scopes and RAR types
- Reusable access requirement profiles (PRM only)
- Attenuation relationships: can_be_attenuated_to
- Per-entry version; family as grouping only

Semantic input to decisions - not the decision. Composes with AuthZEN token issuance / exchange profiles.

Classification: GENERAL

What the draft defines - and what it does not

Defines

- authority_metadata parameter for Protected Resource Metadata (RFC 9728) and AS Metadata (RFC 8414)
- One authority description model for scopes and RAR types
- Reusable access requirement profiles (PRM only)
- Attenuation relationships: can_be_attenuated_to
- Per-entry version; family as grouping only

Does not define

- A universal policy language or action vocabulary
- A replacement for RAR type definitions / schemas
- An AS policy decision algorithm
- New OAuth errors or remediation protocol
- Action approval or transaction evidence protocols

Semantic input to decisions - not the decision. Composes with AuthZEN token issuance / exchange profiles.

Classification: GENERAL

Two publication roles

Protected Resource (RFC 9728)

- Authority semantics the resource applies when enforcing
- Resource-bound via the resource attribute
- Entries carry id, access_requirement_profile, relationships
- May publish access_requirement_profiles

Authorization Server (RFC 8414)

- AS's local understanding of its authority vocabulary
- Identified by kind + value; no id, profiles or relationships
- Useful to clients, policy services, admin tools
- Does not publish issuance policy or predict decisions

- Partial publication is allowed: omission means neither unsupported nor unrestricted
- Matching AS and PR descriptions do not establish trust or guarantee acceptance

Classification: GENERAL

Authority entry model

Attribute	Req.	Meaning
kind	REQ	scope | authorization_details_type
value	REQ	one scope value or RAR type (exact match)
description	OPT	human-readable; untrusted text
family	OPT	grouping
version	OPT	opaque version, latest only
authority	REQ	array of structured descriptions (combined)
id	PR	unique within the document
access_requirement_profile	PR	name of a profile
relationships	PR	e.g. can_be_attenuated_to

Structured authority description

```
{
  "resource_type": "payment",
  "actions": [
    "payment:read",
    "payment:execute"
  ],
  "effects": ["external-transfer"],
  "risk": "high"
}
```

- Structure is defined, vocabularies are not
- URI identifiers RECOMMENDED; e.g. AAuth R3 vocabularies
- RAR: describes what the type can express, not a concrete object

Classification: GENERAL

Example: protected resource metadata (1/2)


```
{
  "resource": "https://documents.example.com",
  "authorization_servers": ["https://as.example.com"],
  "scopes_supported": ["documents.read", "documents.manage"],
  "authorization_details_types_supported": ["document-access"],
  "authority_metadata": {
    "access_requirement_profiles": {
      "standard_access": {
        "token": {
          "max_lifetime": 3600
        }
      },
      "sensitive_access": {
        "token": {
          "max_age": 300,
          "max_lifetime": 600,
          "sender_constraining_methods": ["dpop"]
        },
        "authentication": {
          "acr_values": ["phishing-resistant"],
          "max_age": 300
        }
      }
    },
    "authorities": "..."
  }
}
```

`standard_access`
Token lifetime of at most 1 hour.

`sensitive_access`
Token issued within 5 min, lifetime of at most 10 min, DPoP-bound (RFC 9449; must be supported per RFC 9728 metadata).

`authentication`
Phishing-resistant authentication within the last 5 min. acr_values lists alternatives.

`Named & reusable`
Profile names are local to the document and referenced by authority entries (next slide).

`Requirement, not instruction`
States what the resource requires to accept a token - not what the AS must issue.

Classification: GENERAL

Example: protected resource metadata (2/2)


```
"authorities": [
  {
    "id": "documents.read",
    "kind": "scope",
    "value": "documents.read",
    "family": "document_access",
    "version": "1",
    "authority": [{
      "resource_type": "document",
      "actions": ["document:read"]
    }],
    "access_requirement_profile": "standard_access"
  },
  {
    "id": "documents.manage",
    "kind": "scope",
    "value": "documents.manage",
    "family": "document_access",
    "version": "2",
    "authority": [{
      "resource_type": "document",
      "actions": ["document:read", "document:update"]
    }],
    "access_requirement_profile": "sensitive_access",
    "relationships": [{
      "relation": "can_be_attenuated_to",
      "target": "documents.read"
    }]
  },
```

```
  {
    "id": "document_access.rar",
    "kind": "authorization_details_type",
    "value": "document-access",
    "family": "document_access",
    "authority": [{
      "resource_type": "document",
      "actions": ["document:read", "document:update"]
    }],
    "access_requirement_profile": "sensitive_access"
  }
]
```

documents.read -> standard_access
Read only, version 1.

documents.manage -> sensitive_access
Read + update, version 2 (semantics changed); can be attenuated to documents.read.

document-access (RAR)
Type-level: concrete authority depends on the object's actions and document restrictions.

Classification: GENERAL

Access requirement profiles

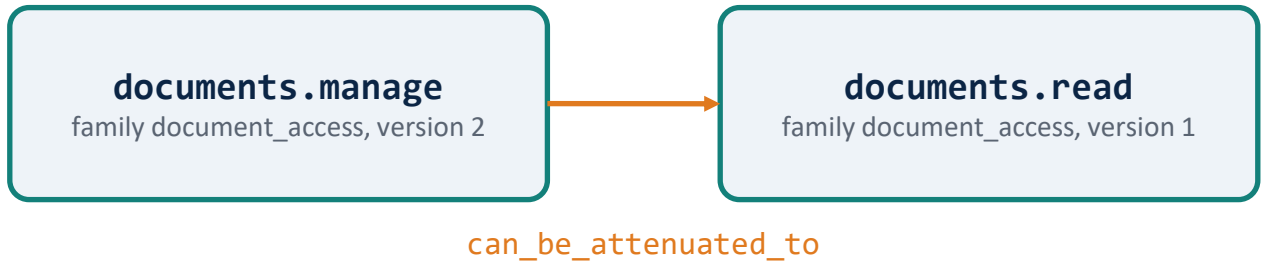
Protected resource only

token	authentication	description
• max_age (s since issued) • max_lifetime (s, iat->exp) • sender_constraining_methods (dpop | mtls; alternatives)	• acr_values (alternatives) • max_age (s since auth event, per RFC 9470) • Both present => both apply	• Human-readable • Profile names are local • No implied standard levels

- A profile states what the resource requires to accept a token - it does not instruct the AS to issue
- No profile => no additional requirements; RFC 9728 resource-wide requirements still apply; profiles list only supported mechanisms
- Multiple authorities in a token: all applicable requirements hold; a weaker profile never cancels a stronger one
- Requirements are not enforcement; a short-lived token does not prove recent authentication

Classification: GENERAL

Attenuation relationships




```
client: resource=https://documents.example.com
        scope=documents.manage

AS policy (read-only task) may:
  - issue documents.read
  - decline + suggest documents.read
  - require consent / revised request
```

- Target represents no greater authority than the source - same document
- RAR / cross-kind: identifies a candidate only - concrete containment needs type semantics; never drop or widen parameters
- No mandatory search, target selection, or new error code - AS policy decides

Classification: GENERAL

How an authorization server uses it

1. Resolve PR via resource indicator (RFC 8707 / 9728)
2. Resolve scope / RAR authority semantics
3. Evaluate vs grant, local policy, Mission
4. Inform PDP (AuthZEN issuance / exchange)
5. Check access requirement profile
6. Identify lower-authority alternatives

No processing sequence is mandated

Publication is not an issuance decision

- Published authority does not mean every client may request it, every subject may grant it, the AS will issue it, or resource will accept it

Applies equally to token requests, token exchange (RFC 8693) and assertion grants such as ID-JAG.

Classification: GENERAL

Layering with related work

Mission-Bound Authorization `draft-mcguinness-oauth-mission`	Task / purpose approval; tokens bound to a Mission
Grant catalog (proposed) `protected, relationship-specific`	Ceilings + combinations per client / issuer / tenant
Authority Metadata `draft-zehavi-oauth-authority-metadata`	What scopes / RAR types mean; requirements; attenuation
RAR Metadata & Remediation `draft-ietf-oauth-rar-metadata-remediation`	RAR type schemas; insufficient-authorization remediation
Discovery foundations `RFC 8414 · RFC 9728 · RFC 9396 · RFC 8707`	AS / PR metadata, RAR, resource indicators

AuthZEN token issuance / exchange profiles consume these semantics for policy evaluation

Classification: GENERAL

Asks of the working group

- Is describing the authority of scopes / RAR types a problem the WG wants to solve?
- Feedback on the model: PR vs AS roles, profiles, attenuation, versioning
- Catalog: extend this draft or a separate companion document layered on it?
- Align with Mission-Bound Authorization and RAR Metadata - joint design work welcome
- Reviewers, implementers and use cases (agents, ID-JAG, token exchange, payments)

```
Draft:   https://datatracker.ietf.org/doc/draft-zehavi-oauth-authority-metadata/
Source:  https://github.com/yaron-zehavi/oauth-authority-metadata
Mission: https://datatracker.ietf.org/doc/draft-mcguinness-oauth-mission/
RAR md:  https://datatracker.ietf.org/doc/draft-ietf-oauth-rar-metadata-remediation/
```

Thank you · oauth@ietf.org

Classification: GENERAL