RAR Metadata & Error Remediation

https://datatracker.ietf.org/doc/draft-ietf-oauth-rar-metadata-remediation/



IETF 127 San Francisco

Yaron Zehavi

Status since IETF 126 Vienna

Adopted by the OAuth WG

Seeing interest and adoption in:

- MCP (SEP-2643), Raiffeisen Bank International, HelseID (Norwegian healthcare)
- AuthZEN: COAZ-RAR binding builds on the RAR types metadata endpoint

Since Vienna:

- Resolved #17: Retrying tokens whose RAR authorizes a range (raised by Nick Watson)
- New proposals: AuthZEN mapping in types metadata, non-OAuth (URL) remediation

Classification: PUBLIC

Draft overview

- New endpoint for RAR types metadata (schema, documentation & examples)
- When token's RAR authority is insufficient ➔ WWW-Authenticate response:
  - Error code: *insufficient_authorization* (aligned draft-rosomakho-oauth-txn-challenge)
  - Actionable authorization details objects (derived from failing request)
  - String reference helping client to locate an **existing** eligible token

Classification: PUBLIC

Authorization Server RAR types metadata endpoint

authorization_details_types_metadata_endpoint

```
{
  "authorization_details_types_metadata": {
    "type": {
      "version": "...",
      "description": "...",
      "documentation_uri": "...",
      "schema": { },
      "schema_uri": "...",
      "examples": [ ]
    }
  }
}
```

Classification: PUBLIC

Resource Server: insufficient token RAR authority

- Normative error code: *insufficient_authorization*
- Optional human readable *error_description*
- *authorization_remediation* (base64url encoded JSON)
  - Actionable RAR objects
  - When resulting token is multi use: *authorization_reference*, guiding client on token selection

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="insufficient_authorization",
error_description="Additional authorization is required",
authorization_remediation=eyJhdXRob3JpemF0aW9uX2RldGFpbHMiOlt7InR5cGUiOiJkaX...
```

Classification: PUBLIC

authorization_remediation single-use example

The decoded `authorization_remediation` contents in this example are:

```
{
    "authorization_details": [{
            "type": "payment_initiation",
            "instructed_amount": {
                "currency": "EUR",
                "amount": "100.00"
            },
            "creditor_account": {
                "iban": "DE02120300000000202051"
            }
        }
    ]
}
```

Classification: PUBLIC

authorization_remediation multiple-use example

The decoded `authorization_remediation` contents in this example are:

```
{
    "authorization_details": [{
            "type": "direct_debit_mandate",
            "DebtorAccount": {
                "SchemeName": "UK.OBIE.SortCodeAccountNumber",
                "Identification": "08080021325698",
                "Name": "JohnDoe"
            },
            "CreditorAgent": {
                "SchemeName": "UK.OBIE.BICFI",
                "Identification": "NWBKGB22"
            },
            "CreditorAccount": {
                "SchemeName": "UK.OBIE.SortCodeAccountNumber",
                "Identification": "08080021325698",
                "Name": "ACMECorp"
            },
            "MandateStatus": "Active",
            "CreationDateTime": "2026-06-01T09:00:00+00:00"
        }
    ],
    "authorization_reference": "Yb7q3AC5d"
}
```

Locate eligible tokens without understanding RAR semantics

Classification: PUBLIC

Topics for discussion

1. AuthZEN mapping in RAR types metadata

- COAZ-RAR (Omri Gazitt): add *`authzen_mapping`* to type's metadata object

2. Non-OAuth remediation (issue #18)

- Remediate via user interaction at a URL, analogous to MCP URL elicitation?

3. Update on a resolved issue (#17)

- Establishing client has satisfying token, when a token's RAR authorizes a range (e.g. up to 100$, new request for 80$)

Classification: PUBLIC

1. AuthZEN mapping in RAR types metadata

COAZ-RAR (Omri Gazitt, OIDF AuthZEN WG)

- How a RAR entry maps to an AuthZEN Authorization API request
- Types metadata endpoint consumed by RS

The ask for this draft

- Consider adding `authzen_mapping` to type's metadata object

Discussion with Omri:

- There's exactly one COAZ mapping per RAR type
- AS is the definer and distribution channel

```
GET /authorization_details_types_metadata

{
  "payment_initiation": {
    "description": "Initiate a payment",
    "schema_uri": "https://as.example/pi.json",
    "authzen_mapping": {
      "evaluation": {
        "action": {
          "name": "$cell.action",
          "properties": {
            "amount":
              "$detail.instructedAmount.amount"
          }
        },
        "resource": {
          "type": "creditor_account",
          "id": "$detail.creditorAccount.iban"
        }
      }
    }
  }
}
```

Classification: PUBLIC

2. Non-OAuth remediation & non-remediable failures (issue #18)

Some failures are remediated by an RS state change, not a new grant

- Analogous to MCP URL-mode elicitation

Current user must act: `remediation_url` (+ optional `expires_in`)

- HTTPS only; same origin / eTLD+1 or sub-domain
- Opaque, short-lived, single-purpose; no tokens, PII or direct object identifiers
- Target MUST require user authentication

Another user (e.g. resource owner) must act:

- `status=owner-action-required`: terminal, client does not poll
- `status=owner-approval-pending` + `interval`: client may poll
- Or no remediation parameters, if RS prefers not to disclose

Questions for the WG

- In scope for this WG?
- In scope for this draft?

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer
 error="insufficient_authorization",
 authorization_remediation=eyJ...

{
 "remediation_url":
   "https://rs.example/approve/8f3k2",
 "expires_in": 300
}
```

Classification: PUBLIC

3. Resolved #17: RAR authorizing a range

Issue (raised at IETF 126 by Nick Watson)

- Authority may be a range: debit ≤ 100$, path `/var/*`
- 80$ and 100$ challenges yield different `authorization_details` and `authorization_reference`
- Reference lookup misses an eligible token
- Can client identify existing valid token without mastering RAR semantics?

```
1. Debit 100$ -> 401
   ref="A", RAR: amount 100$
   new grant -> token T_A

2. Debit 80$ (T_A not tried)
   -> 401, ref="B"
   no token for "B"

3. Retry with T_A
   (same RS origin, multi-use)
   -> 200 OK
```

Resolution (Client Processing Rules, §7.1)

> *"Client MAY attempt calling resource server, either on first attempt or as a remediation step, using any valid tokens which were obtained following a remediation challenge from same resource server origin, which included an authorization_reference,* ***as such tokens are not limited for single-use****."*

- Assumes few in-session tokens per user & RS: negligible latency
- Step 4 loop protection still applies

Classification: PUBLIC

Next steps

Seeking WG input on:

- Metadata extensibility for `authzen_mapping`
- Scope of non-OAuth remediation (#18) and signaling non-remediable failures (#16)

Implementation feedback welcome

- Discussion on mailing list and the GitHub issue tracker

Classification: PUBLIC