Alternative Workflow and OAuth Parameters for the Authentication and Authorization for Constrained Environments (ACE) Framework

draft-ietf-ace-workflow-and-params-03

Marco Tiloca, RISE
Göran Selander, Ericsson

IETF 121 Meeting – Dublin – November 8th, 2024

Recap

Set of updates to RFC 9200

1. **Define an alternative workflow for uploading the access token**
   - The AS uploads the access token to the RS, on behalf of C
   - Preferable if the C-RS communication leg is constrained, while the AS-RS leg is not
2. **Define additional OAuth parameters to use in ACE**
   - Enabling the alternative workflow (1), also with dynamic update of access rights
   - Effectively enabling the issue of an access token for a group-audience
   - Enabling bidirectional access control using a single access token (3)
3. **Define a method to enforce bidirectional access control using a single access token**
4. **Amend two requirements on transport profiles of ACE**
5. **Deprecate the original payload format of error responses**
   - Instead, use the problem-details format from RFC 9290

Updates in v -03

- **Editorial fixes and improvements, including:**
  - Lowercase use of “client”, “resource server”, and “authentication server”
  - Split elision and comments in the examples in CBOR diagnostic notation
  - Amended definition of new entries in IANA registries
  - Heavily trimmed Appendix B, as some planned parameters are defined now
- **Clarification – Section 1.1 “Terminology”**
  - A “token series” ends when the latest access token in the series becomes invalid
- **Fixed naming of audience-related parameters**
  - ACE uses “audience” for message parameters, but “aud” for claims of access tokens
  - Fixed parameter names: s/aud/audience , s/aud2/audience2 , s/rev_aud/rev_audience
  - The name of claims were fine as-is: aud, rev_aud

Updates in v -03

- **Parameter “access_token”**
  - For the Access Token Response, relevant when using the alternative workflow
  - Made the hash function sha-256 mandatory to implement. Consistent with:
    - The used method specified in RFC 6920 and its compliance requirements
    - The requirement of the optional service from *draft-ietf-ace-revoked-token-notification*

- **New parameter “token_series_id” (byte string) – Section 3.6**
  - Identifier of a token series; used for updating access rights, irrespective of the ACE workflow
    - In Access Token Requests: only when requesting an access token for updating access rights
    - In Access Token Response: only when issuing the first access token of a new series
  - All access tokens of a token series include the claim “token_series_id” with the same value
  - Profiles already able to identify token series by other means must not use this parameter
    - Other profiles need this parameter, e.g., *draft-ietf-ace-group-oscore-profile*

Updates in v -03

- **New parameter “to_rs” (byte string) – Section 3.3**
  - Optional for the Access Token Request, when using the alternative workflow
  - Must be accompanied by the parameter “token_upload”
  - Value: information that the AS has to upload to RS on behalf of C, when uploading the access token

- **New parameter “from_rs” (byte string) – Section 3.3**
  - Optional for the Access Token Response, when using the alternative workflow
  - Must be accompanied by the parameter “token_upload” with value 0 (successful token upload)
  - Value: information that the AS provides to C on behalf of RS, after a successful token upload

- **C and RS would exchange the same information directly, if using the original workflow**

- **The semantics of the parameter values depends on the profile of ACE used**

Updates in v -03 (continued)

- **Use of “to_rs” and “from_rs” in the OSCORE profile – Section 3.3.1**
  - Effectively enabling the alternative workflow in this profile
  - C and RS can exchange (N1, ID1) and (N2, ID2) through the AS
- **“to_rs” encodes a CBOR map with two fields:**
  - The nonce N1 generated by C, as a CBOR byte string
  - The Recipient ID ID1 generated by C, as a CBOR byte string
- **“from_rs” encodes a CBOR map with two fields:**
  - The nonce N2 generated by RS, as a CBOR byte string
  - The Recipient ID ID2 generated by RS, as a CBOR byte string
- **The AS builds:**
  - The POST request to /authz-info at RS, using N1 and ID1 from “to_rs”
  - The information in “from_rs”, using N2 and ID2 from the response from RS

```
Access Token Request

Header: POST (Code=0.02)
Uri-Host: "as.example.com"
Uri-Path: "token"
Content-Format: 19 (application/ace+cbor)
Payload:
{
  / audience /  5 : "tempSensor4711",
  / scope /     9 : "read",
  e'token_upload' : 0,
        e'to_rs' : h'a2182848018a278f7faab55a182b421645'
}


Access Token Response

Header: Created (Code=2.01)
Content-Format: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
     e'token_upload' : 0,
          e'from_rs' : h'a2182a4825a8991cd700ac01182c420000',
  / ace_profile / 38 : / coap_oscore / 2,
    / expires_in / 2 : 3600,
    / cnf /        8 : {
      / osc / 4 : {
        / id / 0 : h'01',
        / ms / 2 : h'f9af838368e353e78888e1426bd94e6f'
      }
    }
}
```

Next steps

- **OAuth parameter “anchor_cnf”**
  - Provide guidelines for its use with group-audiences (see also the Github issue #2)

- **Alternative ACE workflow**
  - Detailed handling of an uploaded access token on the RS side
  - Parameter “updated_rights” to prevent ambiguities when dynamically updating access rights (*)

- **Bidirectional access control with a single access token (see also the Github issue #1)**
  - Setup with two different Authorization Servers
  - More practical considerations on applicability, limitations, and group-audiences

- **Comments and reviews are welcome!**

(*) Appendix B provides a high-level plan for addressing this point

Thank you!

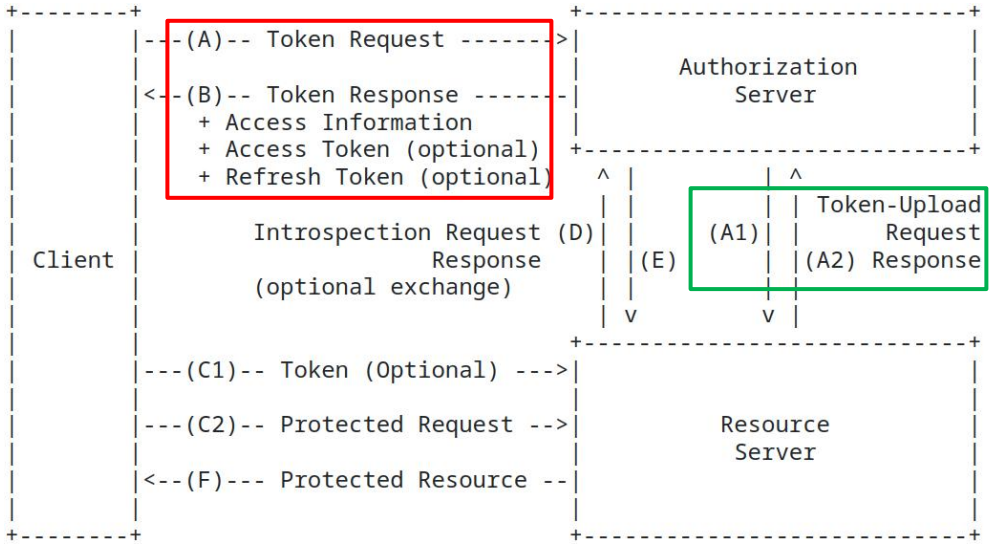
Comments/questions?

https://github.com/ace-wg/ace-workflow-and-params

Backup

Alternative workflow

- **(A) C-to-AS Token Request as usual**
  - C explicitly opts-in for the new workflow, by including the new parameter “token_upload”
  - The final choice about using it is on the AS
- **(A1) The AS uploads the access token to RS, on behalf of C**
  - No intention to replace the original workflow
  - The AS can dynamically choose the workflow to use, e.g., based on the specific RS
- **(A2) The AS receives a response from RS**
- **(B) AS-to-C Token Response**
  - New parameter “token_upload”, with value 0 (successful upload) or 1 (failed upload)
  - **0** → The Response includes: the access token; or a token hash; or neither. Then, C skips step C1.
  - **1** → The Response includes the access token. Then, C performs step C1.

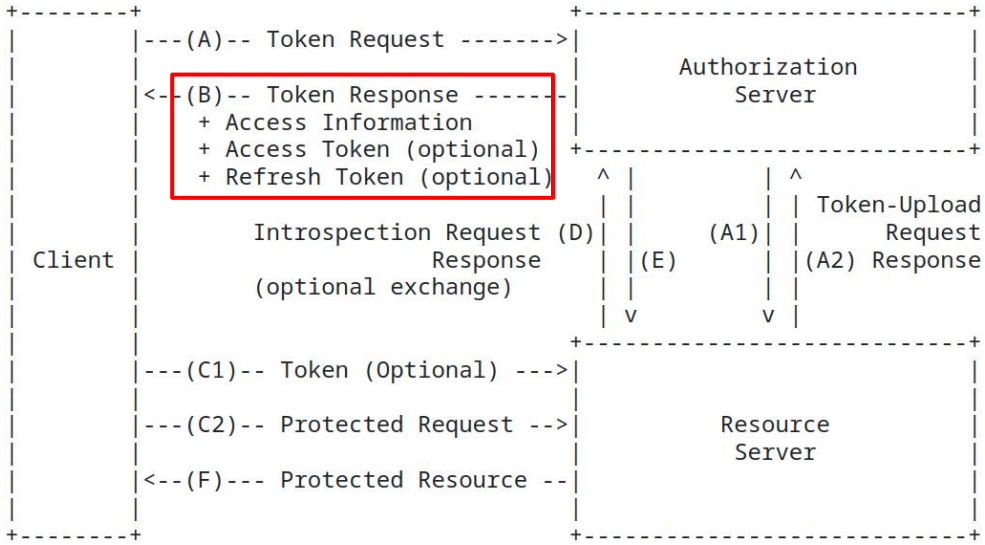


Examples with alternative workflow

```
Access Token Response

Header: Created (Code=2.01)
Content-Format: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
  e'token_upload' : 0,
  / expires_in / 2 : 3600,
  / cnf /        8 : {
    / COSE_Key / 1 : {
      / kty / 1 : 4 / Symmetric /,
      / kid / 2 : h'3d027833fc6267ce',
      / k /  -1 : h'73657373696f6e6b6579'
    }
  }
}
```

Example 1: the AS successfully uploaded the access token




```
Access Token Response

Header: Created (Code=2.01)
Content-Format: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
    e'token_upload' : 1,
  / access_token / 1 : h'd08343a1...4819',
    / (full CWT elided for brevity;
       CWT contains the symmetric PoP key in the "cnf" claim) /
  / expires_in /   2 : 3600,
  / cnf /          8 : {
    / COSE_Key / 1 : {
      / kty / 1 : 4 / Symmetric /,
      / kid / 2 : h'3d027833fc6267ce',
      / k /  -1 : h'73657373696f6e6b6579'
    }
  }
}
```

Example 2: the AS attempted to upload the access token but failed