# Pushed Authorization Requests

draft-ietf-oauth-par

IETF-107, 20.04.2020, virtual

Brian Campbell, Nat Sakimura, Dave Tonge, Filip Skokan, Torsten Lodderstedt

# What is it?

- New pushed authorization request endpoint, which
  - allows clients to push the payload of an OAuth 2.0 authorization request to the authorization server via a direct request and
  - provides them with a request URI (as defined in draft-ietf-oauth-jwsreq) that is used as reference to the data in a subsequent authorization request
- Two modes
  - Authorization request parameters in the PAR body
  - Authorization request parameter in (signed/encrypted) request object

# Traditional OAuth Authorization Request

GET /authorize?response_type=code
&client_id=s6BhdRkqt3
&state=af0ifjsldkj
&redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1
Host: as.example.com

# PAR: same payload but sent directly to AS (incl. client authn)

**POST /as/par** HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded
**Authorization: Basic czZCaGRSa3F0Mzo3RmpmcDBaQnlxS3REUmJuZlZkbUl3**

response_type=code&
client_id=s6BhdRkqt3&
state=af0ifjsldkj&
redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb

# PAR: AS answers with reference to uploaded data

HTTP/1.1 201 Created
Cache-Control: no-cache, no-store
Content-Type: application/json


{
    **"request_uri": "urn:example:bwc4JK-ESC0w8acc191e-Y1LTC2"**,
    "expires_in": 90
}

# PAR: Authorization Request using JAR request_uri

```
GET /authorize?request_uri=
    urn%3Aexample%3Abwc4JK-ESC0w8acc191e-Y1LTC2 HTTP/1.1
```

# Benefits

- Support for large authorization requests (e.g. in authorization_details, claims parameters)
- TLS provides Integrity & Confidentiality protection (confidential & public clients)
- Client authentication and authorization prior to the start of user interaction (confidential clients)
- Signed request object additionally provides non-repudiation

# Status

- Adoption as WG draft
- Part of FAPI 2 baseline profile
- Several implementations exist
- Implementation in ID-Porten (Norwegian eID system), yes® qualified electronic signature service
- Implementation in Norwegian eHealth system planned this autumn
- Proposed for adoption in Australian CDR initiative
- Some open topics

# Open: request_uri must refer to JWT

- According to draft-ietf-oauth-jwsreq request_uri must refer to JWT
- PAR deviates from this requirement since request_uri is produced and consumed by the AS
- Annabelle proposed to add the following text to PAR:
  - As defined in [JAR], the request_uri parameter is required to reference a Request Object JWT. An AS MAY violate this requirement when it is generating request URIs intended for its own consumption (e.g., URIs for pushed requests). This requirement exists to ensure interoperability in cases where the provider of the request_uri is a separate entity from the consumer, such as when a client provides a URI referencing an object stored on the client's backend service. When the AS is both provider and consumer, this interoperability concern does not apply.

# Can AS require PAR?

- Client specific and/or AS wide policy
- Meaning: client is no longer allowed to use traditional (RFC6749) authorization requests

# Can AS require request object?

- Client specific and/or AS wide policy
- Similar discussion in OpenID Connect WG resolved by using request_object_signing_alg client metadata parameter to signal client uses signed request objects only

# Guidance on the request URI structure needed?

- Brian: should there be some more guidance provided on or requirements around the structure of the URI value? For example it could use the RFC6755 subnamespace and registry and be of the form **urn:ietf:params:oauth:request_uri:<>**, which gives a clear indication of what it is and would keep people from inventing their own URIs.
- Filip: I think implementers should be free to either use their own URIs (gives them flexibility), but also have the option to use a registered urn: sub namespace as Brian suggested.
- Brian: using a string representation of a UUID as a URN per https://tools.ietf.org/html/rfc4122#section-3 might also be an option to mention or use in the examples