Pushed Authorization Requests

https://tools.ietf.org/html/draft-lodderstedt-oauth-par

IETF-106, 21.11.2019, Singapore

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

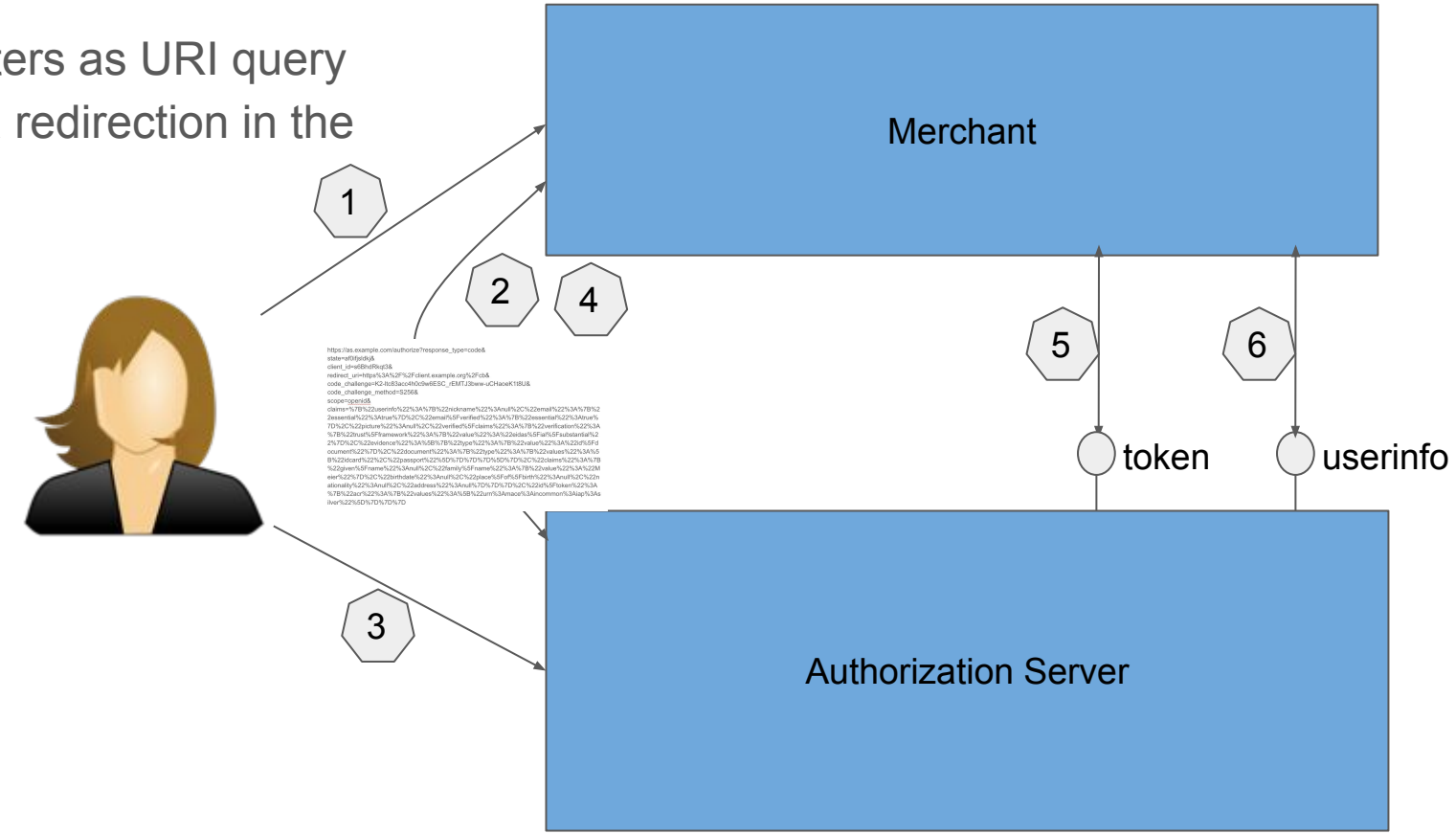
Example Authorization Request
(OpenID for ID Assurance)

https://as.example.com/authorize?response_type=code&
state=af0ifjsldkj&
client_id=s6BhdRkqt3&
redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb&
code_challenge=K2-ltc83acc4h0c9w6ESC_rEMTJ3bww-uCHaoeK1t8U&
code_challenge_method=S256&
scope=openid&
claims=%7B%22userinfo%22%3A%7B%22nickname%22%3Anull%2C%22email%22%3A%7B%22essential%22%3Atrue%7D%2C%22email%5Fverified%22%3A%7B%22essential%22%3Atrue%7D%2C%22picture%22%3Anull%2C%22verified%5Fclaims%22%3A%7B%22verification%22%3A%7B%22trust%5Fframework%22%3A%7B%22value%22%3A%22eidas%5Fial%5Fsubstantial%22%7D%2C%22evidence%22%3A%5B%7B%22type%22%3A%7B%22value%22%3A%22id%5Fdocument%22%7D%2C%22document%22%3A%7B%22type%22%3A%7B%22values%22%3A%5B%22idcard%22%2C%22passport%22%5D%7D%7D%7D%5D%7D%2C%22claims%22%3A%7B%22given%5Fname%22%3Anull%2C%22family%5Fname%22%3A%7B%22value%22%3A%22Meier%22%7D%2C%22birthdate%22%3Anull%2C%22place%5Fof%5Fbirth%22%3Anull%2C%22nationality%22%3Anull%2C%22address%22%3Anull%7D%7D%7D%2C%22id%5Ftoken%22%3A%7B%22acr%22%3A%7B%22values%22%3A%5B%22urn%3Amace%3Aincommon%3Aiap%3Asilver%22%5D%7D%7D%7D


```
{
  "userinfo":{
    "nickname":null,
    "email":{
      "essential":true
    },
    "email_verified":{
      "essential":true
    },
    "picture":null,
    "verified_claims":{
      "verification":{
        "trust_framework":{
          "value":"eidas_ial_substantial"
        },
        "evidence":[
          {
            "type":{
              "value":"id_document"
            },
            "document":{
              "type":{
                "values":[
                  "idcard",
                  "passport"
                ]
              }
            }
          }
        ]
      },
      "claims":{
        "given_name":null,
        "family_name":{
          "value":"Meier"
        },
        "birthdate":null,
        "place_of_birth":null,
        "nationality":null,
        "address":null
      }
    }
  },
  "id_token":{
    "acr":{
      "values":[
        "urn:mace:incommon:iap:silver"
      ]
    }
  }
}
```

OAuth authorization request

sends parameters as URI query parameters via redirection in the user-agent



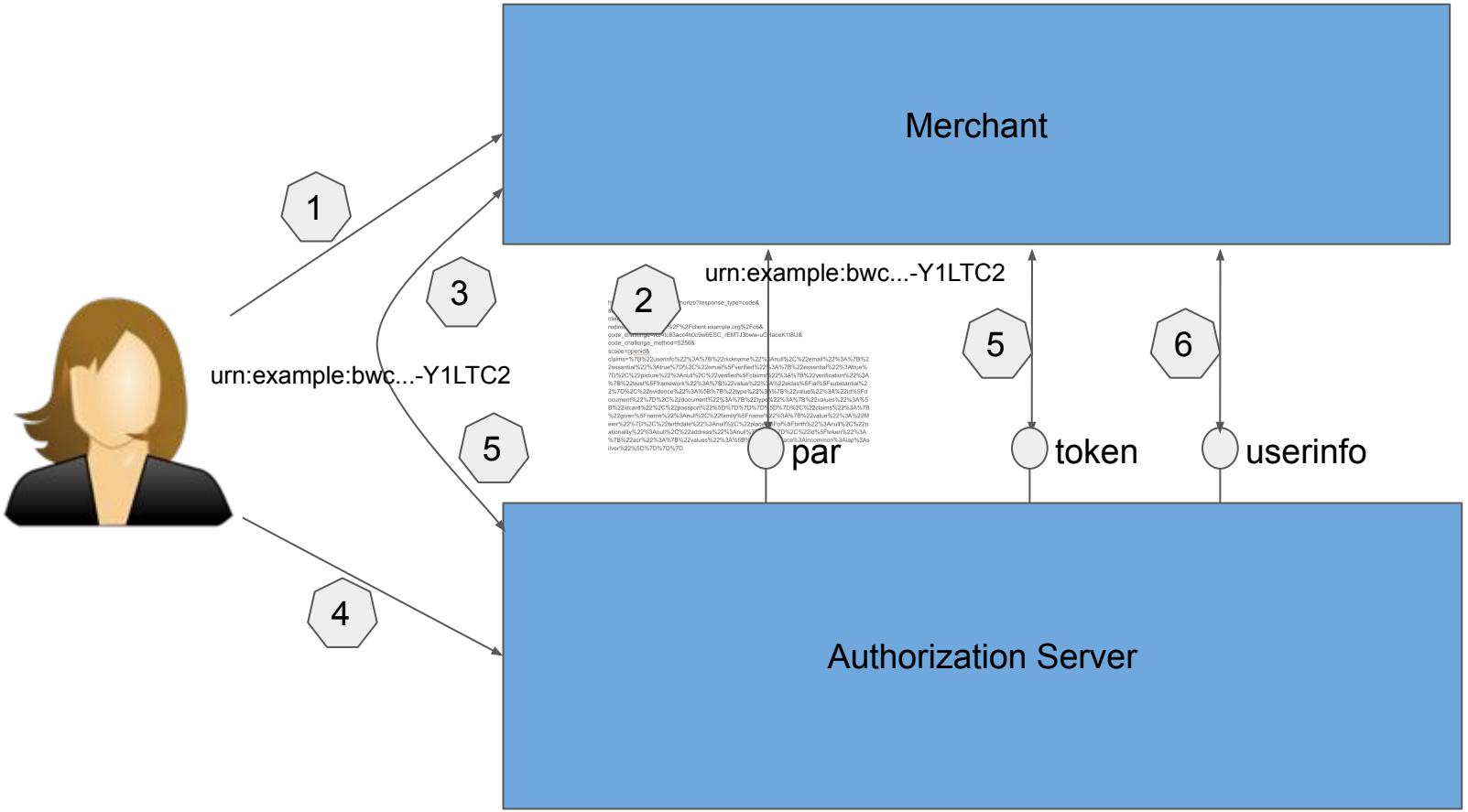
Challenges

- There is no cryptographical integrity and authenticity protection
- There is no mechanism to ensure confidentiality of the request parameters
- Authorization request URLs can become quite large in some use cases

JWT Secured Authorization Request (JAR)

- Security: Allows sending authorization requests in signed and encrypted request objects in JWT format
- Size: *request_uri* allows sending just a URI referring to the request object

Pushed Authorization Requests (Overview)



Pushed Authorization Requests

- **draft-lodderstedt-oauth-par** defines the pushed authorization request endpoint, which allows a client to push the payload of an authorization request to the AS via a direct (POST) request
- The AS provides the client with a request URI (JAR) that is used as reference to the data in a subsequent authorization request

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
```

PAR: signed request object as alternative payload

POST /as/par HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0Mzo3RmpmcDBaQnIxS3REUmJuZlZkbUl3

request=eyJraWQiOiJrMmJkYyIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJzNkJoZFJrcXQzIiwiYXVkIjoiaHR0cHM6Ly9zZXJ2ZXIuZXhhbXBsZS5jb20iLCJyZXNwb25zZV90eXBlIjoiY29kZSIsImNsaWVudF9pZCI6InM2QmhkUmtxdDMiLCJyZWRpcmVjdF91cmkiOiJodHRwczovL2NsaWVudC5leGFtcGxlLm9yZy9jYiIsInNjb3BlIjoiYWlzIiwic3RhdGUiOiJhZjBpZmpzbGRraiIsImNvZGVfY2hhbGxlbmdlIjoiSzItbHRjODNhY2M0aDBjOXc2RVNDX3JFTVRKM2J3dy11Q0hhb2VLMXQ4VSIsImNvZGVfY2hhbGxlbmdlX21ldGhvZCI6IlMyNTYifQ.O49ffUxRPdNkN3TRYDvbEYVr1CeAL64uW4FenV3n9WlaFIRHeFblzv-wlEtMm8-tusGxeE9z3ek6FxkhvvLEqEpjthXnyXqqyJfq3k9GSf5ay74ml_0D6lHE1hy-kVWg7SgoPQ-GB1xQ9NRhF3EKS7UZIrUHbFUCF0MsRLbmtIvaLYbQH_Ef3UkDLOGiU7exhVFTPeyQUTM9FF-u3K-zX-FO05_brYxNGLhVkO1G8MjqQnn2HpAzlBd5179WTzTYhKmhTiwzH-qlBBI_9GLJmE3KOipko9TfSpa26H4JOlMyfZFl0PCJwkByS0xZFJ2sTo3Gkk488RQohhgt1I0onw

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-Y2LTC3",
    "expires_in": 90
}
```

PAR: Authorization Request using JAR request_uri

```
GET /authorize?request_uri=
      urn%3Aexample%3Abwc4JK-ESC0w8acc191e-Y2LTC3 HTTP/1.1
```

Advantages

- Robust solution even for large authorization request payloads
- Significantly improved security
  - Integrity
  - Confidentiality
  - Authenticity
  - Client authentication and authorization ahead of authorization process
  - Seems to be resistant against mix-up (analysis ongoing)
- Easy to use for client developers with simple migration path
- Easy to implement for AS developers (combines authz & token endpoint logic)
- Even higher security level by passing signed/encrypted request objects
- transaction-specific registration of redirect_uri for confidential clients eases client management in AS/OP federations

Status

- -01 revision (based on previous work at the FAPI WG)
- several implementations/prototypes exist (connect2id, oidc-provider, authlete, ID-Porten, yes.com, …)

Would the WG consider to adopt this draft?