OpenID Connect

Presentation at the 19th TF-EMC2 meeting by Roland Hedberg <roland.hedberg@adm.umu.se>

Datum

The ancestry

- OpenID 1.0
  - Identity framework
- OAuth
  - Managing resources

OpenID Connect

OpenID Connect is an identity framework that provides authentication, authorization, and attribute transmission capability.

- Following 4 slides *borrowed* from presentation by Mike Jones -Microsoft

Recent Timeline

- Weekly spec calls began, January 2011
- Open issued closed at IIW, May 2011
- Result branded “OpenID Connect”, May 2011
- Developer feedback, May 2011 to present
- Functionally complete specs, July 2011
- Formal issue tracking began, July 2011
- Interop testing, September 2011
- Simpler specs published incorporating developer feedback, September 2011

Design Criteria

Easy Things Easy

Harder Things Possible

Modular Design

How We Make It Easy

- Build on OAuth 2.0
- Use JavaScript Object Notation (JSON) data structures
- Can only build functionality that you need
- Goal: Easy implementation on all modern web platforms

OpenID

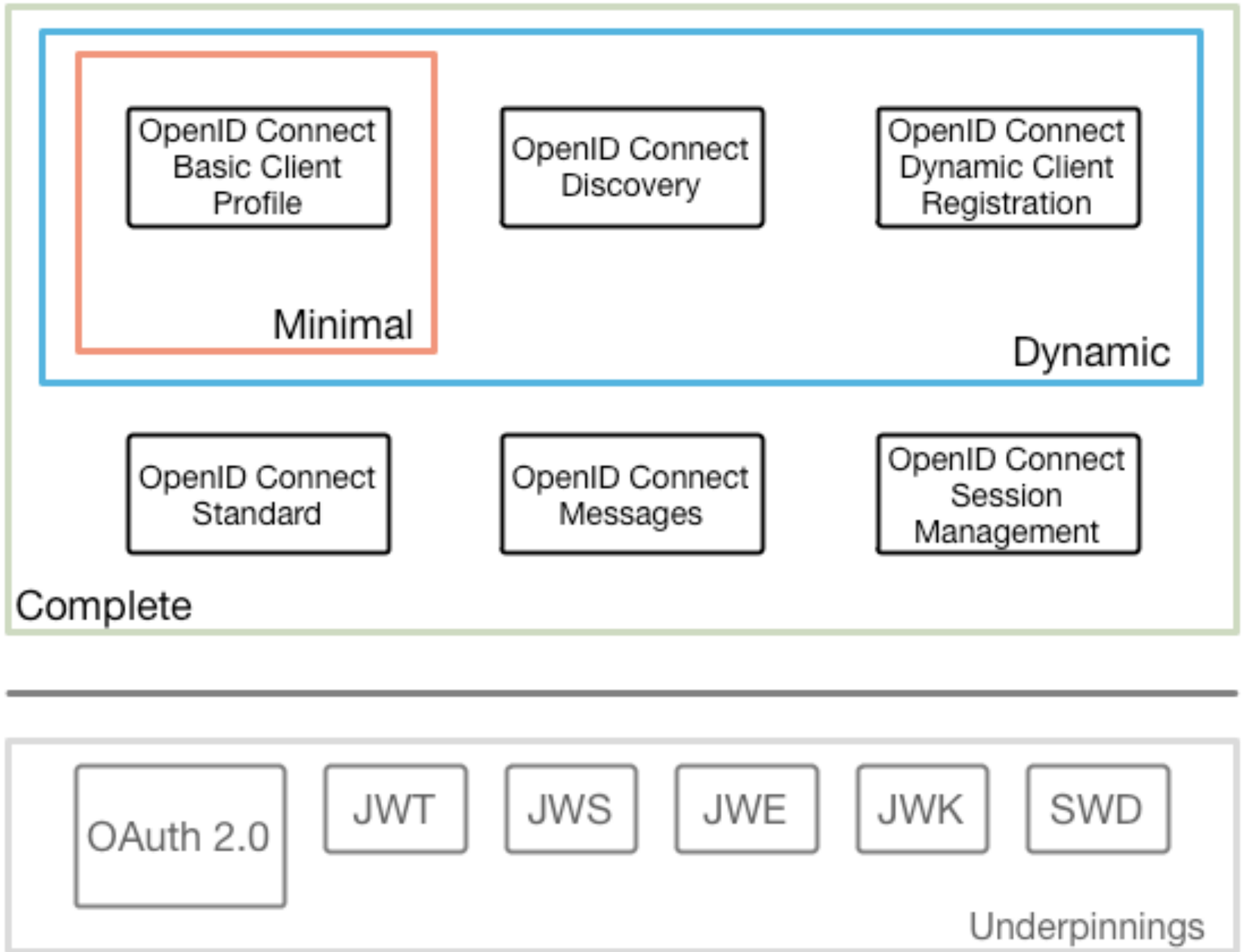
Harder Things Possible

Claims Aggregation

Distributed Claims

Encrypted Claims

OpenID Connect Overview



OpenID Connect Protocol Suite

6 September 2011

http://openid.net/connect

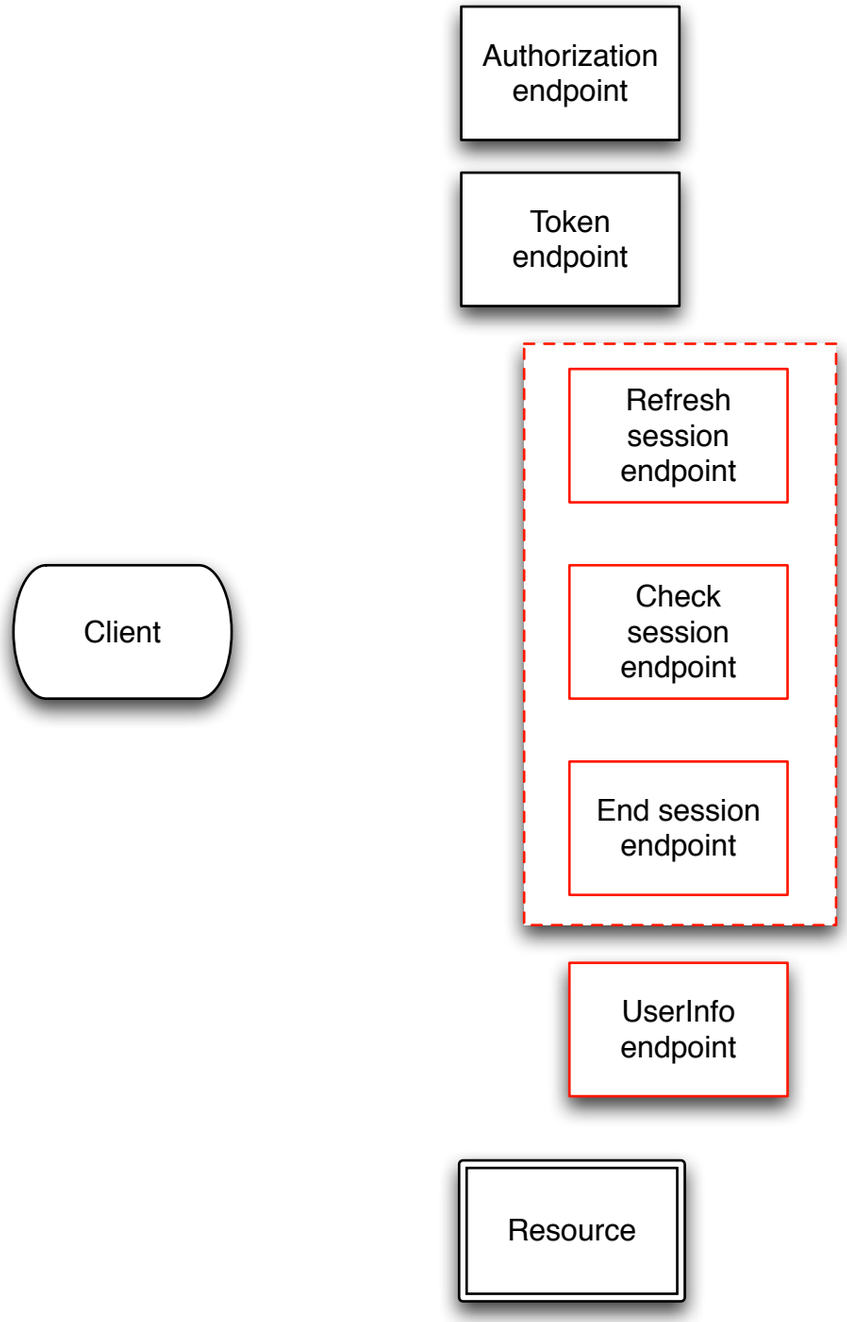
Home of standards to be

- OAuth2.0 in IETF OAUTH wg
- JWS, JWE and JWK in IETF JOSE wg
- JWT and SWD (Simple Web Discovery) homeless !

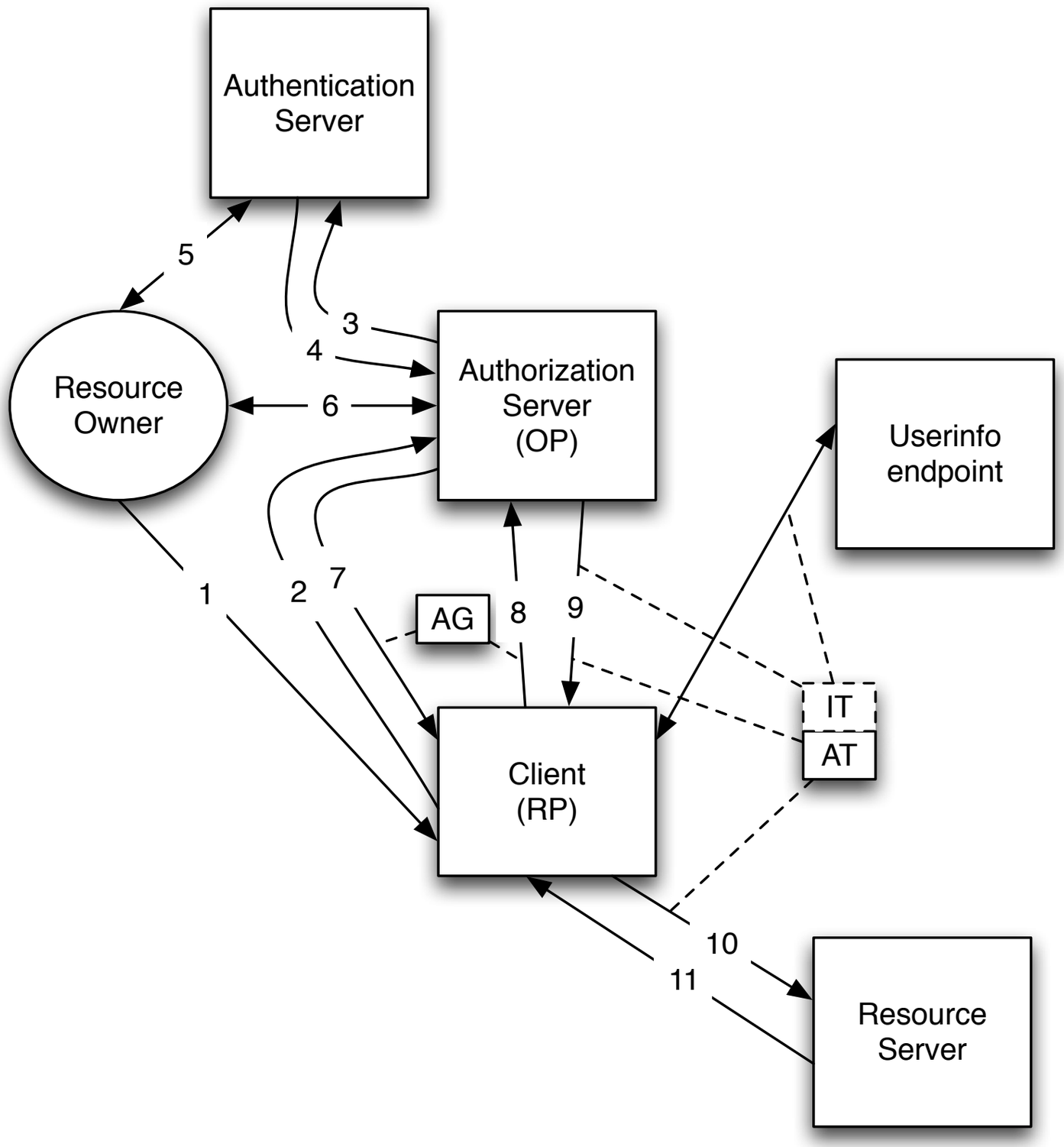
OAuth 2.0 grant types

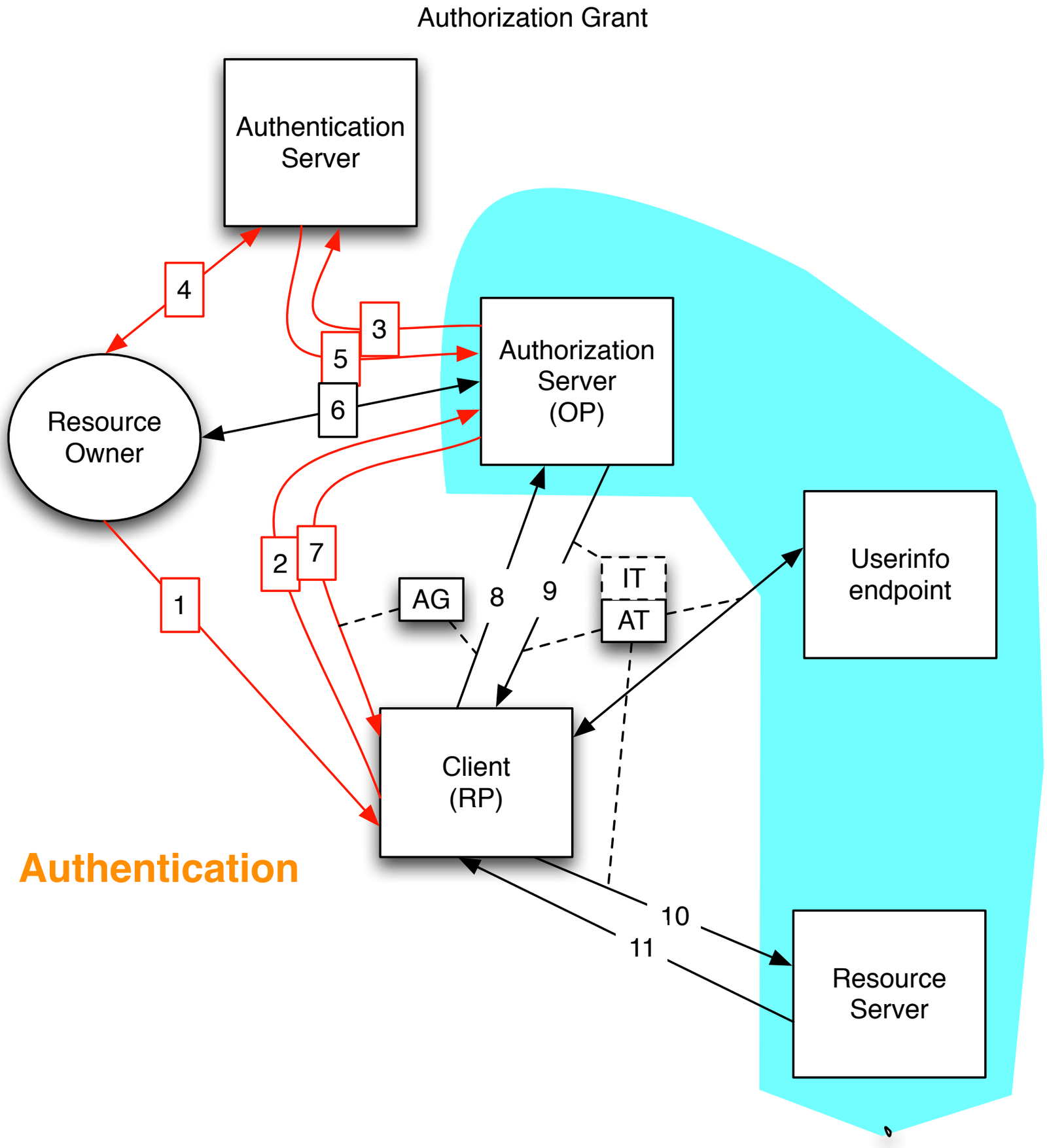
- Authorization code
- Implicit
- Resource owner password credentials (not in OpenID Connect)
- Client credentials (not in OpenID Connect)

OIDC additions

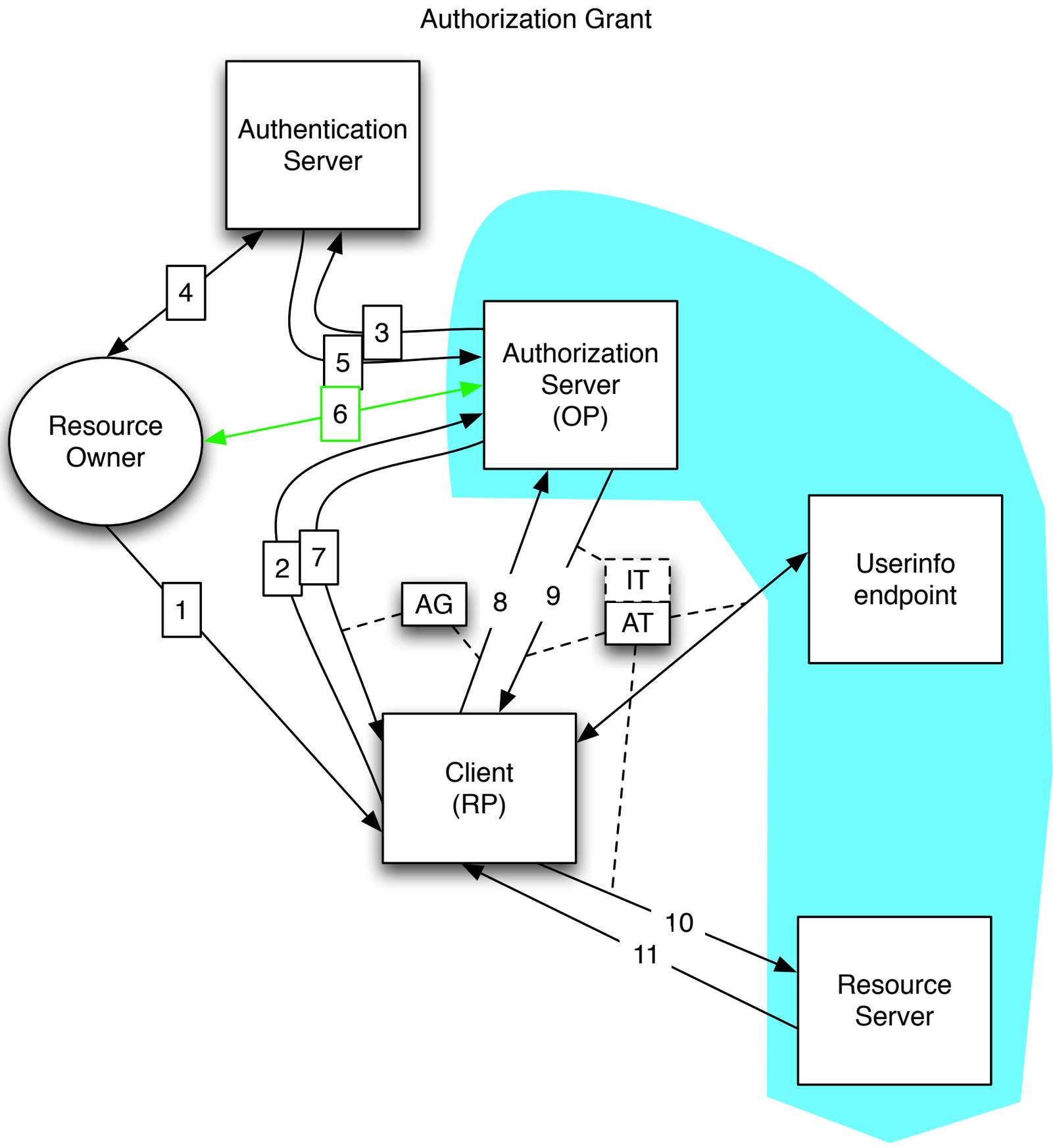


Authorization Grant

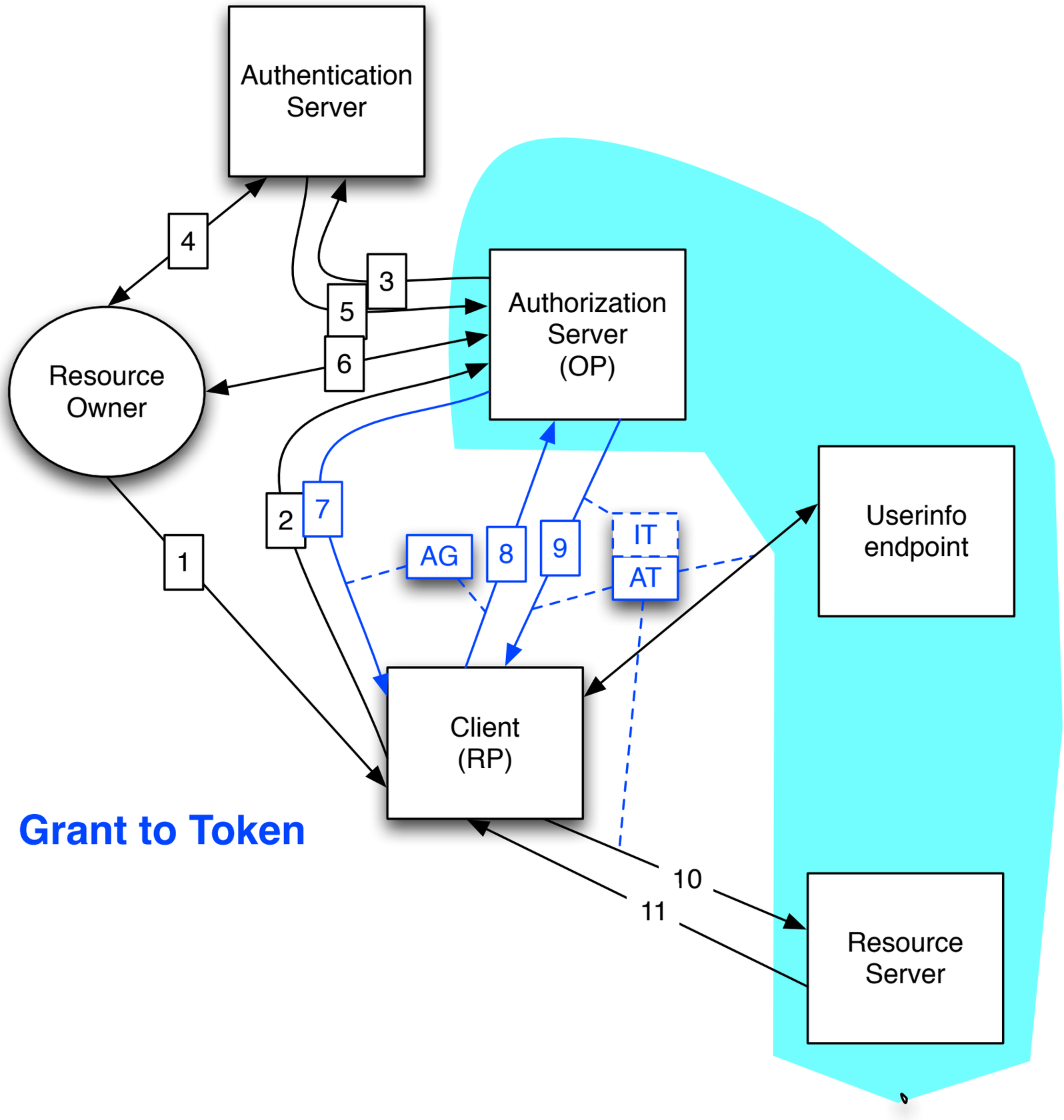



Authorization Grant
Authentication Server
4
3
5
6
Resource Owner
Authorization Server (OP)
2
7
1
AG
8
9
IT
AT
Userinfo endpoint
Client (RP)
Authentication
10
11
Resource Server

Authorization



Authorization Grant



Grant to Token

Authorization -request

https://server.example.com/op/authorize?
response_type=code%20id_token
&**client_id**=s6BhdRkqt3
&**redirect_uri**=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&**scope**=openid
&**nonce**=n-0S6_WzA2Mj
&**state**=af0ifjsldkj
&**request**=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyZXNwb25zZV90eXBlIjoiY2
9kZSBpZF90b2tlbiIsImNsaWVudF9pZCI6InM2QmhkUmtxdDMiLCJyZWRpcmVjdF91cmkiOi
JodHRwczpcL1wvY2xpZW50LmV4YW1wbGUuY29tXC9jYiIsInNjb3BlIjoib3BlbmlkIHByb2
ZpbGUiLCJzdGF0ZSI6ImFmMGlmanNsZGtqIiwidXNlcmluZm8iOnsiY2xhaW1zIjp7Im5hbW
UiOm51bGwsIm5pY2tuYW1lIjp7Im9wdGlvbmFsIjp0cnVlfSwiZW1haWwiOm51bGwsInZlcm
lmaWVkIjpudWxsLCJwaWN0dXJlIjp7Im9wdGlvbmFsIjp0cnVlfX0sImZvcm1hdCI6InNpZ2
5lZCJ9LCJpZF90b2tlbiI6eyJtYXhfYWdlIjo4NjQwMCwiaXNvMjkxMTUiOiIyIn19.2OiqR
grbrHkA1FZ5p_7bc_RSdTbH-wo_Agk-ZRpD3wY

Authorization
-request object unpacked

```
{
      "response_type": "code id_token",
      "client_id": "s6BhdRkqt3",
      "redirect_uri": "https://client.example.com/cb",
      "scope": "openid profile",
      "state": "n-0S6_WzA2Mj",
      "nonce": "af0ifjsldkj",
      "userinfo"{
         "claims":{
            "name": null,
            "given_name": null,
            "family_name": null,
            "nickname": {"optional": true},
            "email": null,
            "verified": null,
            "picture": {"optional": true},
         },
         "format": "signed
      }
      "id_token":{
         "max_age": 86400,
         "iso29115": "2"
      }
}
```

Authorization -response

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?
code=SplxlOBeZQQYbYS6WxSbIA
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
&id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ
9.eyJpc3MiOiJodHRwOlwvXC9zZXJ2ZXIuZXhhbXBsZS5
jb20iLCJ1c2VyX2lkIjoiMjQ4Mjg5NzYxMDAxIiwiYXVk
IjoiaHR0cDpcL1wvY2xpZW50LmV4YW1wbGUuY29tIiwiZ
XhwIjoxMzExMjgxOTcwfQ.eDesUD0vzDH3T1G3liaTNOr
faeWYjuRCEPNXVtaazNQ
```

AccessTokenRequest

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded;charset=UTF-8

grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
```

AccesTokenResponse

```
HTTP/1.1 200 OK

Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
    "access_token": "SlAV32hkKG",
    "token_type": "Bearer",
    "refresh_token": "8xLOxBtZp8",
    "expires_in": 3600,
    "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOl
wvXC9zZXJ2ZXIuZXhhbXBsZS5jb20iLCJ1c2VyX2lkIjoiMjQ4Mjg5NzYxMDAxIiwiYXVkIj
oiaHR0cDpcL1wvY2xpZW50LmV4YW1wbGUuY29tIiwiZXhwIjoxMzExMjgxOTcwfQ.eDesUD0
vzDH3T1G3liaTNOrfaeWYjuRCEPNXVtaazNQ"
}
```

ID Token revealed

```
{
  "iss": "https://server.example.com/op",
  "user_id":"24400320",
  "aud":"s6BhdRkqt3",
  "exp":1320502962,
  "iso29115": 2,
  "nonce":"af0ifjsldkj"
}
```

UserInfoRequest

```
https://server.example.com/op/userinfo?
access_code=SlAV32hkKG
&schema=openid
```

UserInfoResponse

```
{
  "name": "Jane Doe"
  "given_name": "Jane",
  "family_name": "Doe",
  "email": "janedoe@example.com",
  ”verified”: true,
  "picture": "http://example.com/janedoe/me.jpg"
}
```

Discovery (SWD)

Principal: joe@example.com

GET /.well-known/simple-web-discovery?principal=joe%40example%2Ecom&service=http%3A%2F%2Fopenid%2Enet%2Fspecs%2Fconnect%2F1%2E0%2Fissuer
Host: example.com

HTTP/1.1 200 OK
Content-Type: application/json

{
 "locations":["https://example.com/auth"]
}

Dynamically getting endpoint info

```
GET /.well-known/openid-configuration HTTP/1.1
Host: example.com

{
 "authorization_endpoint": "https://example.com/connect/authorize",
 "issuer" : "https://example.com",
 "token_endpoint": "https://example.com/connect/token"
 "user_info_endpoint": "https://example.com/connect/user",
 "check_id_endpoint": "https://example.com/connect/check_id",
 "refresh_session_endpoint": "https://example.com/connect/refresh_session",
 "end_session_endpoint": "https://example.com/connect/end_session",
 "jwk_document": "https://example.com/jwk.json",
 "registration_endpoint": "https://example.com/connect/register",
 "scopes_supported": ["openid"],
 "flows_supported": ["code", "token"],
 "iso29115_supported": ["http://www.idmanagement.gov/schema/2009/05/icam/openid-trust-level1.pdf"],
 "identifiers_supported": ["public", "ppid"]
}
```

Session Management

- A session starts with the AuthorizationRequest
- Refresh session
  - refresh an ID Token that has expired
- Check session
  - Validate and decode JWS encoded ID Tokens
- End session
  - ends a session

Aggregated claims


```
{"name": "Jane Doe",
   "given_name": "Jane",
   "family_name": "Doe",
   "birthday": "01/01/2001",
   "email": "janedoe@example.com",
   "_claim_names": {
       "address": "src1",
       "phone_number": "src1",
       },
   "_claim_sources": {
       "src1": {
           "JWT":"jwt_header.jwt_part2.jwt_part3"
       },
   }
}
```

Distributed claims

```
{
   "name": "Jane Doe",
    "given_name": "Jane",
    "family_name": "Doe",
    "email": "janedoe@example.com",
    "birthday": "01/01/2001",
    "_claim_names": {
              "payment_info": "src1",
              "shipping_address": "src1",
              "credit_score": "src2"
               },
    "_claim_sources": {
         "src1": {
               "endpoint": "https://bank.example.com/claimsource"},
         "src2": {
               "endpoint": "https://creditagency.example.com/claimshere",
               "access_token": "ksj3n283dke"}
    }
}
```

So why should you care?

Working Together

Google

facebook

Microsoft

PingIdentity

OpenID Connect

YAHOO!

NRI

salesforce.com
Success On Demand.™

AOL

YAHOO! JAPAN