CLIENT INITIATED BACKCHANNEL AUTHENTICATION

Authentication Request

- NEW Authorization ENDPOINT: bc-authorize
- Authentication Requests are made using the MODRNA profile
- ONLY the follow parameters are considered:
    - **scope**: REQUIRED. OpenID Connect implements authentication as an extension to the OAuth 2.0 by including the openid scope value in the Authorization Request.
    - **client_notification_endpoint**: OPTIONAL. Callback URI to which the response will be sent.
        - If the "client_notification_endpoint" is not present the RP MUST polls the authorization server repeatedly
    - **acr_values**: REQUIRED. As defined in MODRNA Authentication profile.

Authentication Request II

```
POST /bc-authorize HTTP/1.1
Host: server.example.com
Content-Type: application/json
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
{
   "scope": "openid",
   "client_notification_endpoint": "https://client.example.com/cb",
   "acr_values": "mod-mf",
   "login_hint_token": "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00ifQ.
   OKOawDo13gRp2ojaHV7LFpZcgV7T6DVZKTyKOMTYUmKoTCVJRgckCL9kiMT03JGe
   ipsEdY3mx_etLbbWSrFr05kLzcSr4qKAq7YN7e9jwQRb23nfa6c9d-StnImGyFDb
   Sv04uVuxIp5Zms1gNxKKK2Da14B8S4rzVRltdYwam_lDp5XnZAYpQdb76FdIKLaV
   mqgfwX7XWRxv2322i-vDxRfqNzo_tETKzpVLzfiwQyeyPGLBIO56YJ7eObdv0je8
   1860ppamavo35UgoRdbYaBcoh9QcfylQr66oc6vFWXRcZ_ZT2LawVCWTIy3brGPi
   6UklfCpIMfIjf7iGdXKHzg.
   48V1_ALb6US04U3b.
   5eym8TW_c8SuK0ltJ3rpYIzOeDQz7TALvtu6UG9oMo4vpzs9tX_EFShS8iB7j6ji
   SdiwkIr3ajwQzaBtQD_A.
   XFBoMYUZodetZdvTiFvSkQ"
}
```

Successful Authentication ACK

- **auth_req_id**: REQUIRED. Unique id to identify the authentication request (transaction) made by the RP.
- **expires_in**: REQUIRED. Expiration time of the Authentication in seconds since the auth_request was received.
- **interval**: OPTIONAL. The minimum amount of time in seconds that the client SHOULD wait between polling requests to the token endpoint. ONLY be present when "client_notification_endpoint" parameter no present in auth_req or not any callback URI registered.

Successful Authentication ACK II

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
    "auth_req_id": "1c266114-a1be-4252-8ad1-04986c5b9ac1",
    "expires_in": 3600,
    "interval": 2
}
```

Token Request Using Polling Mechanism

- When NO "client_notification_endpoint" client must polls the token endpoint until the end-user grants or denies the request.
- The polling interval MUST NOT exceed the minimum interval provided by the authorization server via the "interval" parameter (if provided).

The client makes a request to the token endpoint

- **grant_type**: REQUIRED. Value MUST be set "urn:openid:params:modrna:grant-type:backchannel_request".
- **auth_req_id**: REQUIRED. It is a unique id to identify the authentication request (transaction) made by the RP.

Token Request Using Polling Mechanism II

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
{
   grant_type="urn:openid:params:modrna:grant-type:backchannel_request"
   &auth_req_id=40582304823
}
```

Token Response

- If user is well authenticated => succesful response that includes an ID Token and an Access Token. When token is returned in a request using the polling mechanism it also will contain the auth_req_id.
- If the Token Request is invalid or unauthorized => error token response as in OpenID Conect spec. some error codes are specific for this flow:
- **authorization_pending**: The authorization request is still pending as the end-user hasn't yet been authenticated.
- **slow_down**: The client is polling too quickly and should back off at a reasonable rate.

Token Response Example II

SUCCESSFUL RESPONSE

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": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
         yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5
         NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzZ
         fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz
         AKfQ.ggW8hZ1EuVLuxNuuIJKX_V8a_OMXzR0EHR9R6jgdqrOOF4daGU96Sr_P6q
         Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ
         NqeGpe-gccMg4vfKjkM8FcGvnzZUN4_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd
         QyHE5lcMiKPXfEIQILVq0pc_E2DzL7emopWoaoZTF_m0_N0YzFC6g6EJbOEoRoS
         K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3_osjzw2PAithfubEEBLuVVk4
         XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"
}
```

ISSUING TOKEN

When the authentication request includes a "client_notification_endpoint", the Authorization Server will send the token response making a "POST HTTP Request" to the RP's client_notification_endpoint.

ISSUING TOKEN II

```
POST /cb HTTP/1.1
Host: client.example.com
Content-Type: application/json
{
                "auth_req_id": "1c266114-a1be-4252-8ad1-04986c5b9ac1"
                "access_token": "SlAV32hkKG",
                "token_type": "Bearer",
                "refresh_token": "8xLOxBtZp8",
                "expires_in": 3600,
                "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
                yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5
                NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzZ
                fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz
                AKfQ.ggW8hZ1EuVLuxNuuIJKX_V8a_OMXzR0EHR9R6jgdqrOOF4daGU96Sr_P6q
                Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ
                NqeGpe-gccMg4vfKjkM8FcGvnzZUN4_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd
                QyHE5lcMiKPXfEIQILVq0pc_E2DzL7emopWoaoZTF_m0_N0YzFC6g6EJbOEoRoS
                K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3_osjzw2PAithfubEEBLuVVk4
                XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"
}
```

Callback Flow

Polling Flow