G. Fernandez

Telefonica

F. Walter

Deutsche Telekom AG

Aug 18, 2016

OpenID Connect MODRNA Client initiated Backchannel Authentication Flow

1.0

draft-mobile-client-initiated-backchannel-authentication-01

Abstract

OpenID Connect allows RP's to authenticate their users for clients of

all types, including browser-based JavaScript and native mobile apps,

to launch sign-in flows and receive verifiable assertions about the

identity of signed-in users.

In all of these flows initiated by the RP, the end-user interaction

from the consumption device is required and, they are based on HTTP

redirection mechanisms. However, some use cases not covered by these

flows have been raised, where the RP needs to be the initiator of the

user authentication flow and the end-user interaction from the

consumption device is not needed.

The MODRNA Client initiated Backchannel Authentication Flow specifies

a new authentication flow, by means of which the RP's that know the

user identifier they want to authenticate (e-mail, telefon number),

will be able to initiate an interaction flow to authenticate their

users without having end-user interaction from a consumption device.

\*Remark:\* As there is no any consumption device through which the

end-user is interacting with the Relying Party, this flow will not

cause any user credentials to go through the RP. So it should be

highlighted that traditional username/password authentication could

not be used because and only out-band mechanisms will work in

conjunction with this flow.

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 2

1.1. Requirements Notation and Conventions . . . . . . . . . . 2

2. Terminology . . . . . . . . . . . . . . . . . . . . . . . . . 2

3. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . 3

4. Backchannel Authentication Endpoint . . . . . . . . . . . . . 4

4.1. Authentication Request . . . . . . . . . . . . . . . . . 4

4.2. Authentication Request Validation . . . . . . . . . . . . 6

4.3. Successful Authentication Request Acknowledgement . . . . 7

Fernandez & Walter Expires February 19, 2017 [Page 1]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

4.4. Authentication Request Acknowdlegment Validation . . . . 8

5. Authorization Server Obtains End-user Consent/Authorization . 8

6. Getting The Transaction Result . . . . . . . . . . . . . . . 9

6.1. Token Request Using Polling Mechanism . . . . . . . . . . 9

6.2. Successful Token Polling . . . . . . . . . . . . . . . . 10

6.3. Token Notification . . . . . . . . . . . . . . . . . . . 10

6.3.1. Successful Token Notification . . . . . . . . . . . . 10

6.3.2. Client Notification Endpoint . . . . . . . . . . . . 11

6.4. Token Error Response . . . . . . . . . . . . . . . . . . 12

6.5. Authentication Error Response . . . . . . . . . . . . . . 12

7. Security Considerations . . . . . . . . . . . . . . . . . . . 12

8. Privacy Considerations . . . . . . . . . . . . . . . . . . . 12

9. References . . . . . . . . . . . . . . . . . . . . . . . . . 12

9.1. Normative References . . . . . . . . . . . . . . . . . . 12

9.2. Informative References . . . . . . . . . . . . . . . . . 13

Appendix A. Acknowledgements . . . . . . . . . . . . . . . . . . 13

Appendix B. Notices . . . . . . . . . . . . . . . . . . . . . . 13

Appendix C. Document History . . . . . . . . . . . . . . . . . . 14

Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . . 14

1. Introduction

OpenID Connect MODRNA Client initiated Backchannel Authentication

Flow 1.0 is an authentication flow of the OpenID Connect Core 1.0

[OpenID.Core] specification intended to allow RP's who already know

the identifier of an end-user to initiate an authentication flow to

request user authentication. Moreover it defines Mandatory to

Implement features to ensure interoperability of clients.

1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in [RFC2119].

Throughout this document, values are quoted to indicate that they are

to be taken literally. When using these values in protocol messages,

the quotes MUST NOT be used as part of the value.

2. Terminology

This specification uses the term "OpenID Provider (OP)" and "Relying

Party (RP)" as defined by OpenID Connect Core [OpenID.Core]. This

specification also uses the following terms:

Consumption Device (CD) A user agent, most probably a browser, on

which the user consumes the actual service provided by the Relying

Party.

Fernandez & Walter Expires February 19, 2017 [Page 2]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

Authentication Device (AD) A mobile device on which the user will

authenticate the actual login.

3. Overview

This specification defines a new authentication flow based on

[OpenID.Core].

It introduces a new endpoint used to initiate user authentication

using a backchannel request. This new endpoint utilizes existing

Authentication Requests and defines new parameters as appropriate.

For example, it re-uses the scope parameter but it omits nonce, state

and redirect\_uri, which are need to perform and secure authentication

transactions on the front channel.

CIBA polling is illustrated in the following diagram:

+--------+ +--------+

| | | |

| |<---(1) CIBA Request-------------------------->| |

| | | |

| | +--------+ | |

| | | | | |

| Client | | End- |<--(2) User interactions --------->| OP |

| | | User | | |

| | | | | |

| | +--------+ | |

| | | |

| |----(3a) CIBA Polling Request----------------->| |

| |<---(3b) CIBA Polling Response-----------------| |

| | | |

+--------+ +--------+

If the Client expects to be notified about the authentication result

asynchronously then it has to specify a Client Notification Endpoint

during its registration at the Authorization Server.

Fernandez & Walter Expires February 19, 2017 [Page 3]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

CIBA notification is illustrated in the following diagram:

+--------+ +--------+

| | | |

| |<---(1) CIBA Request-------------------------->| |

| | | |

| | +--------+ | |

| | | | | |

| Client | | End- |<--(2) User interactions --------->| OP |

| | | User | | |

| | | | | |

| | +--------+ | |

| | | |

| |<---(3) CIBA Notification Callback ------------| |

| | | |

+--------+ +--------+

4. Backchannel Authentication Endpoint

The Backchannel Authentication Endpoint performs Authentication of

the End-User. This is done by sending an HTTP POST message directly

from the Relying Party to the Authorization Server's Backchannel

Authentication Endpoint, using request parameters defined by OAuth

2.0 and additional parameters and parameter values defined by OpenID

Connect.

Communication with the Backchannel Authentication Endpoint MUST

utilize TLS. See Section 16.17 [OpenID.Core] for more information on

using TLS.

4.1. Authentication Request

An Authentication Request is an OAuth 2.0 [RFC6749] Authorization

Request that requests that the End-User be authenticated by the

Authorization Server.

MODRNA Client initiated Backchannel Authentication defines an

authentication request that is requested directly from the client to

the Authorization Server without going through the user's browser.

The client MUST send an authentication request to the Authorization

Server by building an "HTTP POST" request that will take to the

Authorization Server all the information needed to authenticate the

user without asking them for their identifier.

The client MUST authenticate to the Backchannel Authentication

Endpoint using the authentication method registered for its

client\_id, as described in Section 9 of [OpenID.Core].

Fernandez & Walter Expires February 19, 2017 [Page 4]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

Authentication Requests are made using the MODRNA profile. Only the

following parameters are taking in consideration in the Client

initiated Backchannel Authentication flow. The rest of the request

parameters defined in OAuth 2.0 [RFC6749] MUST be ignored by the

Authorization Server.

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\_req\_id REQUIRED. It is a unique id provided by the RP that

will be used by the Identity Provider as a beared token to

authenticate the callback request to send the tokens to the RP.

client\_notification\_endpoint OPTIONAL. Callback URI to which the

response will be sent. This URI MUST exactly match one of the

"Client initiated Backchannel Authentication" callback URI values

for the Client pre-registered at the OpenID Provider, with the

matching performed as described in Section 6.2.1 of [RFC3986]

(Simple String Comparison). When using this flow, the Callback

URI SHOULD use the https scheme. This paramter will be MANDATORY

if the RP has been registered to receive callbacks.

At the time of registering in Mobile Connect, the RP must provide

the information about whether receiving the token response through

a callback to the "client\_notification\_endpoint" or through a

polling mechanism where the RP MUST poll the authorization server

repeatedly as defined in

Successful Authentication Request Acknowledgement until the end-

user completes the approval process.

acr\_values REQUIRED. As defined in

OpenID Connect MODRNA Authentication Profile 1.0.

login\_hint\_token OPTIONAL. As defined in

OpenID Connect MODRNA Authentication Profile 1.0.

id\_token\_hint OPTIONAL. As defined in Section 3.1.2.1 of

[OpenID.Core].

login\_hint OPTIONAL. As defined in Section 3.1.2.1 of

[OpenID.Core].

context REQUIRED. This parameter provides a context in the Mobile

Connect request that clearly identifies the action/transaction

that the user is being asked to authorise.

Fernandez & Walter Expires February 19, 2017 [Page 5]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

As the Client initiated Backchannel Authentication flow does not have

an interaction with the end-user through a consumption device, it is

REQUIRED that the RP provides one (and only one) of the hints

specified above in the authentication request, that is

"login\_hint\_token", "id\_token\_hint" or "login\_hint".

The following is a non-normative example from an authentication

request (with line wraps within values for display purposes only):

POST /bc-authorize HTTP/1.1

Host: server.example.com

Content-Type: application/json

{

"scope": "openid",

"client\_req\_id": "8d67dc78-7faa-4d41-aabd-67707b374255",

"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"

}

4.2. Authentication Request Validation

The Authorization Server MUST validate the request received as

follows:

1. Authenticate the Client.

e.g.: by validating the Client Credentials as described in

Section 9 on [OpenID.Core].

2. The Authorization Server MUST validate all the OAuth 2.0

parameters according to the MODRNA specification.

3. In the event of a request contains more than one of this hints

specified in Authentication Request (Section 4.1), the Identity

provider MUST return an "invalid\_request" error response as per

Section 3.1.2.6 on [OpenID.Core].

Fernandez & Walter Expires February 19, 2017 [Page 6]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

4. The Authorization server MUST validate that the hint provided

(login\_hint, login\_token\_hint or id\_token\_hint) exist and are not

expired, otherwise it should return an error to notify that the

hint provided has not been found.

5. If the authentication request contains a

"client\_notification\_endpoint" when it has not been provided at

the registering time or viceversa, it is contained when it has

not been provided, the Identity Provider MUST return an

"invalid\_request" error response, per Section 3.1.2.6 on

[OpenID.Core].

6. If the "client\_notification\_endpoint" parameter is present, it

MUST exactly match one of the callback URI values for the Client

pre-registered at the OpenID Provider, with the matching

performed as described in Section 6.2.1 of [RFC3986] (Simple

String Comparison).

NOTE: the callback URI values registered for the

"client\_notification\_endpoint" are different than those

registered for the redirect\_uri in the Authorization Code flow.

7. The Authorization Server MUST verify that all the REQUIRED

parameters are present and their usage conforms to this

specification.

As specified in OAuth 2.0 [RFC6749], Authorization Servers SHOULD

ignore unrecognized request parameters.

If the Authorization Server encounters any error, it MUST return an

error response, per Section 3.1.2.6 on [OpenID.Core].

4.3. Successful Authentication Request Acknowledgement

If the Authentication Validation Request is OK, the Authorization

Server will return an HTTP 200 OK response to the RP to indicate that

the authentication request has been accepted and it is going to be

processed. The body of this response will contain:

auth\_req\_id REQUIRED. It is a unique id to identify the

authentication request (transaction) made by the RP. The

"auth\_req\_id" will be sent too in the token through the

"client\_notification\_endpoint" to allow the RP to correlate the

authentication request and the received tokens.

expires in REQUIRED. Expiration time of the Authentication in

seconds since the auth\_request was received.

Fernandez & Walter Expires February 19, 2017 [Page 7]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

interval OPTIONAL. The minimum amount of time in seconds that the

client SHOULD wait between polling requests to the token endpoint.

This parameter will only be present in case of the authentication

request doesn't take the "client\_notification\_endpoint" parameter

or there are not any callback URI registered as

"client\_notification\_endpoint" for the RP.

If the Authentication Validation Request is NOK, the Authorization

Server MUST return an error response, per Section 3.1.2.6.

The following is a non-normative example from an authentication

response

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

}

4.4. Authentication Request Acknowdlegment Validation

If the client (RP) receives an HTTP 200 OK, it MUST validate that all

the required parameters are received. The RP should keep the

authentication request identifier "auth\_req\_id" and the expiration

time in order to match it with token response.

The client will have to keep the expiration time as well to be able

to discard the authentication request acknowdlegment.

5. Authorization Server Obtains End-user Consent/Authorization

Once the End-User is authenticated, the Authorization Server MUST

obtain an authorization decision before releasing information to the

Relying Party. When using the Client initiated Backchannel

Authentication flow, there is not any interactive dialogue between

the Authorization Server and the end-user through the consumption

device, so the consent establishing only can take place via the

authentication device or by establishing consent via conditions for

processing the request or other means (for example, via previous

administrative consent). Sections 2 and 5.3 of [OpenID.Core]

describe information release mechanisms.

Fernandez & Walter Expires February 19, 2017 [Page 8]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

NOTE: the current version of the Client initiated Backchannel

Authentication Flow spec. does not specify any mechanism to use the

authentication device to obtain the end-user consent.

6. Getting The Transaction Result

6.1. Token Request Using Polling Mechanism

When the authentication request doesn't include the

"client\_notification\_endpoint" parameter or there are not any

callback URI registered for this notification endpoint for this RP in

the Identity Provider, the client must poll the token endpoint until

the end-user grants or denies the request.

The client polls at reasonable interval which MUST NOT exceed the

minimum interval provided by the authorization server via the

"interval" parameter (if provided).

The client makes a "POST" request to the token endpoint by sending

the following parameters using the "application/x-www-form-

urlencoded" format with a character encoding of UTF-8 in the HTTP

request entity-body:

grant\_type REQUIRED. Value MUST be set to

"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. The

"auth\_req\_id" will be sent too in the Successful Token Polling to

allow the RP to correlate the authentication request and the

received tokens (id\_token and access\_token).

The following is a non-normative example of a token request (with

line wraps within values for display purposes only).

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=1c266114-a1be-4252-8ad1-04986c5b9ac1

}

Fernandez & Walter Expires February 19, 2017 [Page 9]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

6.2. Successful Token Polling

If the "client\_notification\_endpoint" is not present the

Authorization Server will return the token as an HTTP response to the

RP's poll request. If the user is well authenticated, the

Authorization Server will returns a succesful response that includes

an ID Token and an Access Token. Otherwise it responses with an

error as defined in Token Error Response.

The following is a non-normative example of a token response sending

as an HTTP 200 OK Response to the RP's poll request (with line wraps

within values for display purposes only).

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"

}

6.3. Token Notification

6.3.1. Successful Token Notification

When the client is registered for client notifications (through

"client\_notification\_endpoint"), the Authorization Server will send

the token response making a "POST HTTP Request" to the RP's

client\_notification\_endpoint. If the user is well authenticated, the

Authorization Server returns a successful response that includes an

ID Token and an Access Token and a Refresh Token. If the request

Fernandez & Walter Expires February 19, 2017 [Page 10]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

failed client authentication or is invalid, the authorization server

sends an error\_message as described in Token Error Response.

All the parameters in the successful response are defined in

Section 4.1.4 of OAuth 2.0 [RFC6749] except the new one "auth\_req\_id"

that will be the same received in the request as defined in

Successful Authentication Response (Section 4.3). The response uses

the application/json media type.

The following is a non-normative example of a token response sending

as a HTTP POST request to the callback uri specified in the

"client\_notification\_endpoint" parameter by the RP (with line wraps

within values for display purposes only). The request is

authenticated through a bearer token that is the value of the

"client\_req\_id" provided by the RP in the Authentication Request.

POST /cb HTTP/1.1

Host: client.example.com

Authorization: Bearer 8d67dc78-7faa-4d41-aabd-67707b374255

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"

}

6.3.2. Client Notification Endpoint

The Client Notification Endpoint is set by the RP during the

registration phase. This endpoint is intended to receive the result

of the authentication (id\_token, access\_token and refresh token) and

it requires the request to be authenticated using a "bearer token"

created by the RP and sent to the OP in the Authentication request as

the value of the parameter "client\_req\_id".

Fernandez & Walter Expires February 19, 2017 [Page 11]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

Communication with the Client Notification Authentication Endpoint

MUST utilize TLS. See Section 16.17 [OpenID.Core] for more

information on using TLS.

6.4. Token Error Response

If the Token Request is invalid or unauthorized, the Authorization

Server constructs the error response according to the section 3.1.3.4

Token Error Response of [OpenID.Core]. In addition to the error

codes defined in Section 5.2 of [RFC6749], the following error codes

are specific for the Client Initiated Backchannel 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.

6.5. Authentication Error Response

TBD

7. Security Considerations

The login hint token SHOULD be digitally signed by the issuer. This

ensures authenticity of the data and reduces the threat of an

injection attack. The signature allows the OP to authenicate and

authorize the sender of the hint and prevent collecting of phone

numbers by rogue clients.

8. Privacy Considerations

The login hint token is encrypted in order to protect the user's

MSISDN from being revealed to the client unintentionally.

9. References

9.1. Normative References

[I-D.jones-oauth-amr-values]

Jones, M., Hunt, P., and A. Nadalin, "Authentication

Method Reference Values", draft-jones-oauth-amr-values-05

(work in progress), February 2016.

Fernandez & Walter Expires February 19, 2017 [Page 12]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

[MODRNA.Authentication]

Connotte, J. and J. Bradley, "OpenID Connect MODRNA

Authentication Profile 1.0", Sep 2016,

<https://bitbucket.org/openid/mobile/raw/default/draft-

mobile-authentication-01.txt>.

[OpenID.Core]

Sakimura, N., Bradley, J., Jones, M., de Medeiros, B.,

Mortimore, C., and E. Jay, "OpenID Connect Core 1.0",

December 2013, <https://openid.net/specs/openid-connect-

core-1\_0.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119,

DOI 10.17487/RFC2119, March 1997,

<http://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform

Resource Identifier (URI): Generic Syntax", STD 66,

RFC 3986, DOI 10.17487/RFC3986, January 2005,

<http://www.rfc-editor.org/info/rfc3986>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework",

RFC 6749, DOI 10.17487/RFC6749, October 2012,

<http://www.rfc-editor.org/info/rfc6749>.

Appendix A. Acknowledgements

The following have contributed to the development of this

specification.

Appendix B. Notices

Copyright (c) 2015 The OpenID Foundation.

The OpenID Foundation (OIDF) grants to any Contributor, developer,

implementer, or other interested party a non-exclusive, royalty free,

worldwide copyright license to reproduce, prepare derivative works

from, distribute, perform and display, this Implementers Draft or

Final Specification solely for the purposes of (i) developing

specifications, and (ii) implementing Implementers Drafts and Final

Specifications based on such documents, provided that attribution be

made to the OIDF as the source of the material, but that such

attribution does not indicate an endorsement by the OIDF.

Fernandez & Walter Expires February 19, 2017 [Page 13]

OIDC MODRNA Client initiated Backchannel Authentication FlowA1.02016

The technology described in this specification was made available

from contributions from various sources, including members of the

OpenID Foundation and others. Although the OpenID Foundation has

taken steps to help ensure that the technology is available for

distribution, it takes no position regarding the validity or scope of

any intellectual property or other rights that might be claimed to

pertain to the implementation or use of the technology described in

this specification or the extent to which any license under such

rights might or might not be available; neither does it represent

that it has made any independent effort to identify any such rights.

The OpenID Foundation and the contributors to this specification make

no (and hereby expressly disclaim any) warranties (express, implied,

or otherwise), including implied warranties of merchantability, non-

infringement, fitness for a particular purpose, or title, related to

this specification, and the entire risk as to implementing this

specification is assumed by the implementer. The OpenID Intellectual

Property Rights policy requires contributors to offer a patent

promise not to assert certain patent claims against other

contributors and against implementers. The OpenID Foundation invites

any interested party to bring to its attention any copyrights,

patents, patent applications, or other proprietary rights that may

cover technology that may be required to practice this specification.

Appendix C. Document History

[[ To be removed from the final specification ]]

-01

o Initial draft

o Added OIDF Standard Notice

Authors' Addresses

Gonzalo Fernandez Rodriguez

Telefonica I+D

Email: gonzalo.fernandezrodriguez@telefonica.com

Florian Walter

Deutsche Telekom AG

Email: F.Walter@telekom.de

Fernandez & Walter Expires February 19, 2017 [Page 14]