**ISO #####-#:####(X)**

ISO TC ###/SC ##/WG #

Secretariat: XXXX

**Information technology — Financial-grade API: Client Initiated Backchannel Authentication Profile**

PAS Submission

© ISO 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

|  |
| --- |
| [**TOC**](#toc) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Draft-02 | D. Tonge | |  | Moneyhub | |  | J. Heenan | |  | Authlete | |  | T. Lodderstedt | |  | Yes | |  | B. Campbell | |  | Ping Identity | |  | August 15, 2019 | |

# Financial-grade API: Client Initiated Backchannel Authentication Profile

### Warning

This document is not an OIDF International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

### Copyright notice

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.

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.

### Foreword

The OpenID Foundation (OIDF) promotes, protects and nurtures the OpenID community and technologies. As a non-profit international standardizing body, it is comprised by over 160 participating entities (workgroup participants). The work of preparing implementer drafts and final international standards is carried out through OIDF workgroups in accordance with the OpenID Process. Participants interested in a subject for which a workgroup has been established has the right to be represented in that workgroup. International organizations, governmental and non-governmental, in liaison with OIDF, also take part in the work. OIDF collaborates closely with other standardizing bodies in the related fields.

Final drafts adopted by the Workgroup through consensus are circulated publicly for the public review for 60 days and for the OIDF members for voting. Publication as an OIDF Standard requires approval by at least 50 % of the members casting a vote. There is a possibility that some of the elements of this document may be the subject to patent rights. OIDF shall not be held responsible for identifying any or all such patent rights.

Financial-grade API consists of the following parts:

* Part 1: Read-Only API Security Profile
* Part 2: Read and Write API Security Profile
* Financial-grade API: Client Initiated Backchannel Authentication Profile
* Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)
* Financial-grade API: Pushed Request Object

Future parts may follow.

This parts is intended to be used with [RFC6749], [RFC6750], [RFC7636], [OIDC], and [CIBA].

### Introduction

The Financial-grade API Standard provides a profile for OAuth 2.0 suitable for use in financial services. The standard OAuth method for the client to send the resource owner to the authorization server is to use an HTTP redirect. Parts 1 and 2 of this specification support this interaction model and are suitable for use cases where the resource owner is interacting with the client on a device they control that has a web browser. There are however many use-cases for initiating payments where the resource owner is not interacting with the client in such a manner. For example, the resource owner may want to authorize a payment at a "point of sale" terminal at a shop or fuel station.

This document is a profile of the OpenID Connect Client Initiated Backchannel Authentication Flow [CIBA] that supports this decoupled interaction method. The CIBA spec allows a client that gains knowledge of an identifier for the user to obtain tokens from the authorization server. The user consent is given at the user's Authentication Device mediated by the authorization server. This document profiles the CIBA specification to bring it in line with the other FAPI parts and provides security recommendations for its use with APIs that require financial-grade security.

Although it is possible to code an OpenID Provider and Relying Party from first principles using this specification, the main audience for this specification is parties who already have a certified implementation of OpenID Connect and want to achieve a higher level of security. Implementors are encouraged to understand the security considerations contained in section 7.5 before embarking on a 'from scratch' implementation.

### Notational conventions

The keywords "shall", "shall not", "should", "should not", "may", and "can" in this document are to be interpreted as described in ISO Directive Part 2 [ISODIR2]. These keywords are not used as dictionary terms such that any occurrence of them shall be interpreted as keywords and are not to be interpreted with their natural language meanings.

### Table of Contents

[1.](#_1.__Scope)  Scope  
[2.](#_2.__Normative)  Normative references  
[3.](#_3.__Terms)  Terms and definitions  
[4.](#_4.__Symbols)  Symbols and Abbreviated terms  
[5.](#_5.__Read)  Read and Write API Security Profile  
    [5.1.](#_5.1.__Introduction)  Introduction  
    [5.2.](#_5.2.__Read)  Read and Write API Security Provisions  
        [5.2.1.](#_5.2.1.__Introduction)  Introduction  
        [5.2.2.](#_5.2.2.__Authorization)  Authorization Server  
        [5.2.3.](#_5.2.3.__Confidential)  Confidential Client  
            [5.2.3.1.](#_5.2.3.1.__General)  General Provisions  
    [5.3.](#_5.3.__Extensions)  Extensions to CIBA authentication request  
[6.](#_6.__Accessing)  Accessing Protected Resources  
    [6.1.](#_6.1.__Introduction)  Introduction  
    [6.2.](#_6.2.__Client)  Client Provisions  
[7.](#security-considerations)  Security Considerations  
    [7.1.](#_7.1.__Introduction)  Introduction  
    [7.2.](#_7.2.__Authentication)  Authentication sessions started without a users knowledge or consent  
    [7.3.](#_7.3.__Reliance)  Reliance on user to confirm binding messages  
    [7.4.](#_7.4.__Loss)  Loss of fraud markers to OpenID provider  
    [7.5.](#_7.5.__Incomplete)  Incomplete or incorrect implementations of the specifications  
    [7.6.](#_7.6.__JWS/JWE)  JWS/JWE Algorithm considerations  
    [7.7.](#_7.7.__Authentication)  Authentication Device security  
    [7.8.](#_7.8.__CIBA)  CIBA token delivery modes  
    [7.9.](#_7.9.__TLS)  TLS considerations  
    [7.10.](#_7.10.__Algorithm)  Algorithm considerations  
    [7.11.](#_7.11.__Encryption)  Encryption algorithm considerations  
[8.](#_8.__Privacy)  Privacy Considerations  
[9.](#_9.__Acknowledgement)  Acknowledgement  
[10.](#_10.__Bibliography)  Bibliography  
    [10.1.](#_10.1.__Appendix)  Appendix A - Examples  
        [10.1.1.](#_10.1.1.__A.1)  A.1 Signed Authentication Request with private\_key\_jwt client authentication  
[§](#_Authors'_Addresses)  Authors' Addresses

|  |
| --- |
| [**TOC**](#toc) |

### 1.  Scope

This document specifies the method for an application to:

* obtain OAuth tokens via a backchannel authentication flow in an appropriately secure manner for financial data access and other similar situations where the risk is higher;
* use tokens to interact with protected data via REST endpoints.

|  |
| --- |
| [**TOC**](#toc) |

### 2.  Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applied. For undated references, the latest edition of the referenced document (including any amendments) applies.

[ISODIR2] - ISO/IEC Directives Part 2 [ISODIR2]: http://www.iso.org/sites/directives/2016/part2/index.xhtml

[CIBA] - OpenID Connect Client Initiated Backchannel Authentication Core [CIBA]: http://openid.net/specs/openid-client-initiated-backchannel-authentication-core-1\_0.html

[FAPI1] - FAPI Read Only API Security Profile [FAPI1]: https://openid.net/specs/openid-financial-api-part-1.html

[FAPI2] - FAPI Read Write API Security Profile [FAPI2]: https://openid.net/specs/openid-financial-api-part-2.html

[FAPILI] - FAPI Lodging Intent [FAPILI]: https://bitbucket.org/openid/fapi/src/master/Financial\_API\_Pushed\_Request\_Object.md

|  |
| --- |
| [**TOC**](#toc) |

### 3.  Terms and definitions

For the purpose of this standard, the terms defined in RFC6749, RFC6750, RFC7636, OpenID Connect Core and OpenID Connect Client Initiated Backchannel Authentication Core apply.

|  |
| --- |
| [**TOC**](#toc) |

### 4.  Symbols and Abbreviated terms

**API** – Application Programming Interface

**FAPI** - Financial-grade API

**HTTP** – Hyper Text Transfer Protocol

**OIDF** - OpenID Foundation

**REST** – Representational State Transfer

**TLS** – Transport Layer Security

|  |
| --- |
| [**TOC**](#toc) |

### 5.  Read and Write API Security Profile

|  |
| --- |
| [**TOC**](#toc) |

### 5.1.  Introduction

The OIDF Financial-grade API (FAPI) is a REST API that provides JSON data representing higher risk data. These APIs are protected by the OAuth 2.0 Authorization Framework that consists of [RFC6749], [RFC6750], [RFC7636], and other specifications.

The Client Initiated Backchannel Authentication Flow [CIBA] specifies an alternate method of users granting access to their resources whereby the flow is started from a consumption device, but authorized on an authentication device.

The following sections specify a profile of CIBA that is suited for financial-grade APIs.

|  |
| --- |
| [**TOC**](#toc) |

### 5.2.  Read and Write API Security Provisions

|  |
| --- |
| [**TOC**](#toc) |

### 5.2.1.  Introduction

This profile applies to both Read-Only APIs and Read-and-Write APIs.

This spec should be read in conjunction with OpenID Connect Client Initiated Backchannel Authentication Core [CIBA] and with parts 1 [FAPI1] and 2 [FAPI2] of the Financial-grade API specification.

|  |
| --- |
| [**TOC**](#toc) |

### 5.2.2.  Authorization Server

The Authorization Server shall support the provisions specified in clause 5.2.2 of Financial-grade API - Part 1 and clause 5.2.2 of Financial-grade API - Part 2.

In addition the Authorization server, for all operations,

1. shall only support Confidential Clients for Client Initiated Backchannel Authentication flows;
2. shall ensure unique authorization context exists in the authorization request or require a binding\_message in the authentication request;
3. shall not support CIBA push mode;
4. shall support CIBA poll mode;
5. may support CIBA ping mode;
6. shall require Backchannel Authentication Endpoint requests to be signed as described in [CIBA] 7.1.1;
7. shall require user authentication to an appropriate level for the operations the client will be authorized to perform on behalf of the user;
8. shall, if it supports the acr claim and the client has requested acr, return an 'acr' claim in the resulting ID token;
9. shall require the Signed Authentication Request to contain nbf and exp claims that limit the lifetime of the request to no more than 60 minutes;
10. may require clients to provide a request\_context claim as defined in section 5.3 of this profile; and
11. should not use the login\_hint or login\_hint\_token to convey "intent ids" or any other authorization metadata

**NOTE:** As per [CIBA], login\_hint, login\_hint\_token and id\_token\_hint are used only to determine who the user is. In scenarios where complex authorization parameters need to be conveyed from the Client to the AS, implementers should consider the "lodging intent" pattern described in [FAPILI]. The use of parameterized scope values or the use of an additional request parameter are both supported by this specification. Examples of both patterns are shown in [FAPILI].

**NOTE:** The binding message is required to protect the user by binding the session on the consumption device with the session on the authentication device. An example use case is when a user is paying at POS terminal. The user will enter their user identifier to start the [CIBA] flow, the terminal will then display a code, the user will receive a notification on their phone (the authentication device) to ask them to authenticate and authorize the transaction, as part of the authorization process the user will be shown a code and will be asked to check that it is the same as the one shown on the terminal.

**NOTE:** The FAPI CIBA profile only supports CIBA ping and poll modes, therefore it is only possible to retrieve access tokens and optionally refresh tokens from the token endpoint. The same security requirements for the token endpoint as detailed in [FAPI1] and [FAPI2] apply.

**NOTE:** Given that the CIBA flow places an added level of trust on the Client, the FAPI CIBA profile requires the use of Signed Authentication Requests. This will enable the Authorization Server to store such requests, in an easily verifiable form, for future auditing purposes.

**NOTE:** While the format of the login\_hint and login\_hint\_token parameters are not defined by [CIBA] or this profile, implementers may wish to consider https://tools.ietf.org/html/draft-ietf-secevent-subject-identifiers for a standards based method of communicating user identifiers.

|  |
| --- |
| [**TOC**](#toc) |

### 5.2.3.  Confidential Client

|  |
| --- |
| [**TOC**](#toc) |

### 5.2.3.1.  General Provisions

A Confidential Client shall support the provisions specified in clause 5.2.4 of Financial-grade API - Part 1 [FAPI1] and clause 5.2.4 of Financial-grade API - Part 2 [FAPI2].

In addition, the Confidential Client

1. shall only send Signed Authentication Requests as defined in [CIBA] 7.1.1 to the Backchannel Authentication Endpoint;
2. shall ensure sufficient authorization context exists in authorization request or shall include a binding\_message in the authentication request; and
3. shall ensure the Authorization Server has authenticated the user to an appropriate level for the client's intended purpose.

|  |
| --- |
| [**TOC**](#toc) |

### 5.3.  Extensions to CIBA authentication request

This profile defines the following extensions to the authentication request defined in [CIBA] section 7.1.

1. request\_context: OPTIONAL. a JSON object (the contents of which are not defined by this specification) containing information to inform fraud and threat decisions. For example, an ecosystem may require relying parties to provide geolocation for the consumption device.

|  |
| --- |
| [**TOC**](#toc) |

### 6.  Accessing Protected Resources

|  |
| --- |
| [**TOC**](#toc) |

### 6.1.  Introduction

The provisions detailed in Parts 1 and 2 of the Financial-grade API specification apply fully. The benefit of the CIBA specification is that once tokens are issued they can be used in the same manner as tokens issued via authorization code flows.

|  |
| --- |
| [**TOC**](#toc) |

### 6.2.  Client Provisions

In situations where the client does not control the consumption device, the client

1. shall not send x-fapi-customer-ip-address or x-fapi-auth-date headers; and
2. should send metadata about the consumption device, for example geolocation and device type.

|  |
| --- |
| [**TOC**](#toc) |

### 7.  Security Considerations

|  |
| --- |
| [**TOC**](#toc) |

### 7.1.  Introduction

The [CIBA] specification introduces some new attack vectors not present in OAuth 2 redirect based flows. This profile aims to help implementers of [CIBA] for financial-grade APIs to reduce or eliminate these attack vectors. There are however further security considerations that should be taken into account when implementing this specification.

|  |
| --- |
| [**TOC**](#toc) |

### 7.2.  Authentication sessions started without a users knowledge or consent

As this specification allows the client to initiate an authentication request it is important for the authorization server to know whether the user is aware and has consented to the authentication process. If widely known user identifiers (e.g. phone numbers) are used as the login\_hint in the authentication request then this risk is worsened. An attacker could start unsolicited authentication sessions on large numbers of authentication devices, causing distress and potentially enabling fraud. For this reason this profile highly recommends login\_hint to have the properties of a nonce with the expectation being that it will be generated from an authorization server owned client authentication device. Given the high levels of friction that this may impose it's anticipated that Authorization Servers may have to accept an id\_token\_hint as an alternative mechanism for Client Subject identification.

If a client wishes to store the id\_token returned from an authorization server for later use as an id\_token\_hint, care must be taken to ensure that the customer identification mechanism used to retrieve the id\_token is appropriate for the channel being used. For illustration a QR code on a 'club card' may be an appropriate identifier when using a POS terminal under CCTV but it might not be an appropriate identifier when used in online ecommerce.

In addition, [CIBA] provides an optional user\_code mechanism to specifically mitigate this issue, it may be appropriate to require the use of user\_code in certain deployments.

|  |
| --- |
| [**TOC**](#toc) |

### 7.3.  Reliance on user to confirm binding messages

Depending on the hint used to identify the user and the Client's user authentication processes, it may be possible for a fraudster to start a malicious [CIBA] flow at the same time as a genuine flow, with both flows using the genuine user’s identifier. If the scope of access requested is similar then the only way to ensure that a user is authorizing the correct transaction is for the user to compare the binding messages on the Authentication and Consumption devices.

If this risk is deemed unacceptable then implementers should either consider alternative mechanisms of verifying the binding message (e.g. conveying it to the Authentication device via a QR code), or use ephemeral user identifiers generated on the Authentication device.

|  |
| --- |
| [**TOC**](#toc) |

### 7.4.  Loss of fraud markers to OpenID provider

In a redirect-based flow, the authorization server can collect useful fraud markers from the user-agent. In a [CIBA] flow the separation of consumption and authentication devices reduces the data that can be collected. This could reduce the effectiveness of any fraud detection system.

|  |
| --- |
| [**TOC**](#toc) |

### 7.5.  Incomplete or incorrect implementations of the specifications

To achieve the full security benefits, it is important the implementation of this specification, and the underlying OpenID Connect and OAuth specifications, are both complete and correct.

The OpenID Foundation provides tools that can be used to confirm that an implementation is correct:

https://openid.net/certification/

The OpenID Foundation maintains a list of certified implementations:

https://openid.net/developers/certified/

Deployments that use this specification should use a certified implementation.

|  |
| --- |
| [**TOC**](#toc) |

### 7.6.  JWS/JWE Algorithm considerations

CIBA Authorization Servers and Clients shall follow the guidance around JWT signing and encryption Algorithms in [FAPI2] 8.6 and 8.6.1.

|  |
| --- |
| [**TOC**](#toc) |

### 7.7.  Authentication Device security

This profile and the underlying specifications do not specify how the Authorization Server should initiate and perform user authentication and authorization of consent on the authentication device.

Implementors must use appropriately strong methods to communicate with the authentication device and to authenticate the end user.

|  |
| --- |
| [**TOC**](#toc) |

### 7.8.  CIBA token delivery modes

[CIBA] defines 3 ways that tokens can be delivered to the client.

The push mode is not permitted by this specification as it delivers tokens to the client by calling an endpoint owned by the client. This substantially differs from the established pattern of retrieving tokens by presenting client authentication to the token endpoint, and it may have security concerns that are currently unknown.

The poll and ping modes both follow the established convention of retrieving tokens from the token endpoint and hence do not have this concern.

The ping mode delivers a notification to an endpoint owned by the client. The information contained in this notification is limited to the auth\_req\_id for the request, as described in [CIBA] 10.2. The bearer token used by the authorization server to access this resource is not sender constrained. If the backchannel\_client\_notification\_endpoint, the auth\_req\_id and the client\_notification\_token are known to an attacker, they may be able to force the client to call the token endpoint repeatedly or before the authentication has completed. For most deployments this is not a significant issue.

|  |
| --- |
| [**TOC**](#toc) |

### 7.9.  TLS considerations

As confidential information is being exchanged, all interactions shall be encrypted with TLS (HTTPS).

The recommendations for Secure Use of Transport Layer Security in [BCP195] shall be followed, with the following additional requirements:

1. TLS version 1.2 or later shall be used for all communications.
2. A TLS server certificate check shall be performed, as per [RFC6125].
3. For TLS versions below 1.3, only the following 4 cipher suites shall be permitted:
   * TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
   * TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
   * TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
   * TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
4. When using the TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 or TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 cipher suites, key lengths of at least 2048 bits are required.

|  |
| --- |
| [**TOC**](#toc) |

### 7.10.  Algorithm considerations

For JWS, both clients and authorization servers:

1. shall use PS256 or ES256 algorithms;
2. should not use algorithms that use RSASSA-PKCS1-v1\_5 (e.g. RS256);
3. shall not use none;

|  |
| --- |
| [**TOC**](#toc) |

### 7.11.  Encryption algorithm considerations

For JWE, both clients and authorization servers

1. shall not use the RSA1\_5 algorithm.

|  |
| --- |
| [**TOC**](#toc) |

### 8.  Privacy Considerations

There are no additional privacy considerations beyond those in [CIBA] 15.

|  |
| --- |
| [**TOC**](#toc) |

### 9.  Acknowledgement

The following people contributed heavily towards this document:

* Nat Sakimura (Nomura Research Institute) -- Chair, Editor
* Anoop Saxana (Intuit) -- Co-chair, FS-ISAC Liaison
* Anthony Nadalin (Microsoft) -- Co-chair, SC 27 Liaison
* Edmund Jay (Illumila) -- Co-editor
* Dave Tonge (Moneyhub) -- Co-chair, UK Implementation Entity Liaison
* Brian Campbell (Ping Identity)
* John Bradley (Yubico)
* Henrik Biering (Peercraft)
* Axel Nennker (Deutsche Telekom)
* Ralph Bragg (RAiDiAM)
* Joseph Heenan (Authlete)
* Torsten Lodderstedt (yes.com)
* Takahiko Kawasaki (Authlete)

|  |
| --- |
| [**TOC**](#toc) |

### 10.  Bibliography

[RFC6749] - The OAuth 2.0 Authorization Framework [RFC6749]: https://tools.ietf.org/html/rfc6749

[RFC6750] - The OAuth 2.0 Authorization Framework: Bearer Token Usage [RFC6750]: https://tools.ietf.org/html/rfc6750

[OIDC] - OpenID Connect Core 1.0 incorporating errata set 1 [OIDC]: http://openid.net/specs/openid-connect-core-1\_0.html

|  |
| --- |
| [**TOC**](#toc) |

### 10.1.  Appendix A - Examples

The following are non-normative examples of the FAPI-CIBA requests and responses.

All examples use private\_key\_jwt client authentication with the following key:

{

"kty": "EC",

"d": "gM\_\_X2faDsb4s6QLer9h-y4KzLIgwt5Jz2dJi5r64Pc",

"use": "sig",

"kid": "thrwqnuer",

"crv": "P-256",

"x": "YPczq3aBrd8PjtFsXX\_HPZNwnzp89vAGjgQXm4cOgdQ",

"y": "eqE4OZu0V07qXi9ojhQAeqKndWp0QwUfB3aNp4dYYPQ",

"alg": "ES256"

}

|  |
| --- |
| [**TOC**](#toc) |

### 10.1.1.  A.1 Signed Authentication Request with private\_key\_jwt client authentication

This example includes various optional fields, some of which may not be applicable to some deployments. Line wraps within values are for display purposes only.

POST /backchannel-authorization-endpoint HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded

request=eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InRocndxbnVl

ciJ9.eyJpc3MiOiIzMDExODMzNzM4MTQ5NzkiLCJhdWQiOiJodHRwczovL3NlcnZ

lci5leGFtcGxlLmNvbS8iLCJpYXQiOjE1NjQ5MDI3MzgsIm5iZiI6MTU2NDkwMjc

zOCwiZXhwIjoxNTY0OTAzMDM4LCJqdGkiOiJBSnhaUnBOcWxnNjJVVGR5MzdndSI

sInNjb3BlIjoib3BlbmlkIHBheW1lbnRzIiwiYWNyX3ZhbHVlcyI6InVybjptYWN

lOmluY29tbW9uOmlhcDpzaWx2ZXIgdXJuOm1hY2U6aW5jb21tb246aWFwOmJyb25

6ZSIsImNsaWVudF9ub3RpZmljYXRpb25fdG9rZW4iOiJfTWlVT1kwN0VPQ3ZXUjV

CVnVPTD0iLCJsb2dpbl9oaW50Ijoiam9obkBleGFtcGxlLmNvbSIsImJpbmRpbmd

fbWVzc2FnZSI6IlMyNFIiLCJ1c2VyX2NvZGUiOiI2MzY1IiwicmVxdWVzdGVkX2V

4cGlyeSI6IjEyMCIsInJlcXVlc3RfY29udGV4dCI6eyJsb2NhdGlvbiI6eyJsYXQ

iOjUxLjE3Mzk3LCJsbmciOi0xLjgyMjM4fX0sInBheW1lbnRfaW50ZW50Ijp7ImF

tb3VudCI6IjE2NS44OCIsImN1cnJlbmN5IjoiR0JQIiwiY3JlZGl0b3JfYWNjb3V

udCI6eyJzY2hlbWVfbmFtZSI6IlVLLk9CSUUuU29ydENvZGVBY2NvdW50TnVtYmV

yIiwiaWRlbnRpZmljYXRpb24iOiIwODA4MDAyMTMyNTY5OCIsIm5hbWUiOiJBQ01

FIEluYyJ9fX0.6YQ2j27lXlsfw5QFUoDDbkXJnu8ldi6Tw8LwUEg\_C1w2ru\_tksY

yIN81jv4Q0NXwRBtWsojahPFynZJa39Q3Yg&

client\_assertion=eyJraWQiOiJ0aHJ3cW51ZXIiLCJhbGciOiJFUzI1NiJ9.ey

JzdWIiOiIzMDExODMzNzM4MTQ5NzkiLCJhdWQiOiJodHRwczovL3NlcnZlci5leG

FtcGxlLmNvbS8iLCJpc3MiOiIzMDExODMzNzM4MTQ5NzkiLCJleHAiOjE1NjQ5MD

I3OTgsImlhdCI6MTU2NDkwMjczOCwianRpIjoiNnNSVndWdVpseDFERUJjSEVIaH

gifQ.b9fpM3hUv5Nex9DZOYS8AGUiBMIFnlvf5YgRmUqzBhljIGr4M5f-mkt2VOM

ImaKe-LaUMeD5y\_PZGaBiDTo50A&

client\_assertion\_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-asser

tion-type%3Ajwt-bearer

which contains the JWT payload:

{

"iss": "301183373814979",

"aud": "https://server.example.com/",

"iat": 1564902738,

"nbf": 1564902738,

"exp": 1564903038,

"jti": "AJxZRpNqlg62UTdy37gu",

"scope": "openid payments",

"acr\_values": "urn:mace:incommon:iap:silver urn:mace:incommon:iap:bronze",

"client\_notification\_token": "\_MiUOY07EOCvWR5BVuOL=",

"login\_hint": "john@example.com",

"binding\_message": "S24R",

"user\_code": "6365",

"requested\_expiry": "120",

"request\_context": {

"location": {

"lat": 51.17397,

"lng": -1.82238

}

},

"payment\_intent": {

"amount": "165.88",

"currency": "GBP",

"creditor\_account": {

"scheme\_name": "UK.OBIE.SortCodeAccountNumber",

"identification": "08080021325698",

"name": "ACME Inc"

}

}

}

|  |
| --- |
| [**TOC**](#toc) |

### Authors' Addresses

|  |  |
| --- | --- |
|  | Dave Tonge |
|  | Moneyhub |
| **Email:** | [dave.tonge@moneyhub.com](mailto:dave.tonge@moneyhub.com) |
| **URI:** | <http://moneyhubenterprise.com> |
|  |  |
|  | Joseph Heenan |
|  | Moneyhub |
| **Email:** | [joseph@authlete.com](mailto:joseph@authlete.com) |
| **URI:** | <http://authlete.com> |
|  |  |
|  | Torsten Lodderstedt |
|  | YES |
| **Email:** | [torsten@lodderstedt.net](mailto:torsten@lodderstedt.net) |
| **URI:** | <http://yes.com> |
|  |  |
|  | Brian Campbell |
|  | Ping Identity |
| **Email:** | [bcampbell@pingidentity.com](mailto:bcampbell@pingidentity.com) |
| **URI:** | <http://www.pingidentity.com/> |