Data Standards Body
Technical Working Group

Decision 061 to 066 & 068 – Information Security Profile

Contact: James Bligh

Publish Date: 24th May 2019

Decision Approved By Chairman: 28th May 2019

Context

The Information Security profile has been undergoing consultation since mid to late 2018. During this time a number of decisions have been made regarding the approach the CDR regime will take to ensure information security practices are consistently applied to protect participants as data is shared.

This decision defines an iteration of the full information security profile and encompasses the consultation that has occurred since the last full release of the profile in December 2018. This consultation has included:

- Regular, in person, workshops with security experts representing the impacted Banks that were facilitated by the Australian Banking Association
- Open consultation via the Information Security GitHub repository
- A series of seven decision proposals number 061 – 066 and 068 on various specific security related topics

It should be noted that the profile contained in this decision has some areas where further consultation is being sought. These are described in the *Issues under Continuing Consideration* section below and also called out throughout the decision in the sections of the profile that may be impacted.

In addition, an independent review of the profile will be conducted in parallel with community consultation to ensure that the profile, as a whole, provides a firm basis of security for the regime.

These, along with some other drivers for change to the profile, are described below in the *Drivers For Future Variation* section.

Decision To Be Made

Define the next release of the information security profile including specific areas where consensus is still being sought.

Feedback Provided

This content of this decision is based on the proposals and the associated feedback can be found at the following locations:

- **Information Security Issues Tracker**
  https://github.com/ConsumerDataStandardsAustralia/infosec/issues
- **Decision Proposal 061 – Client & Customer Authentication**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/61
- **Decision Proposal 062 – Authorisation Flow**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/62
- **Decision Proposal 063 – Normative References**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/63
- **Decision Proposal 064 – Scopes, Claims & Tokens**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/64
- **Decision Proposal 065 – Transaction Security**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/65
- **Decision Proposal 066 – Authorisation End Points**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/66
- **Decision Proposal 068 – Reauthorisation**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/68
- **Open Feedback Cycle 6**
  https://github.com/ConsumerDataStandardsAustralia/open-banking/issues/67

This feedback has been applied to the baseline Information Security Profile agreed to and published in December 2018 that can be found via the release notice at:
https://consumerdatastandards.org.au/standards/christmas-2018-working-draft

Issues under Continuing Consideration

The aggregated feedback arising from the above consultations has resulted in many additions to the information security profile for which there is broad consensus. In a number of cases, however, consensus has still to be reached.

Where possible, a suggested position is taken to provide certainty for in-flight implementations. For some issues, rather than present a single position, the options that are being considered are presented to help clarify and focus for further consultation and input being sought.

These areas for further consideration, along with a summary of how they are addressed in this decision, are as follows:

- **Authorisation/Authentication flow**
  While a detailed proposal for the authentication flow has been published for consultation the variety of feedback has indicated that more consultation is required. This decision does not include additional specificity around the authentication flow that was not present in the December 2018 draft. The decision does articulate the options that will be the focus of further consultation in a non-normative callout. Additional CX testing is also being conducted which will be taken into account as well.

- **Inclusion of a Consent API**
  A decision on the inclusion or exclusion of a Consent API is not articulated in this decision. Options for further consultation on this topic are presented and discussed in a non-normative callout.
- **Dynamic client registration**
  The mechanism for client registration has been excluded from this decision as this choice is derived from the design of the CDR Registry, which has not yet been published. The areas that may be impacted by the design of the CDR Registry have been identified via non-normative callouts.
- **Reauthorisation**
  Further consultation on the mechanism for reauthorisation is deemed to be required. For this reason only options for reauthorisation have been presented via a non-normative callout. In the interim the profile has presented a default position that reauthorisation will be achieved using the normal authorisation process.

Drivers For Future Variation

The resolution of the areas of continuing consideration described above will result in changes to the Information Security Profile described in this decision. Similarly, on-going international standards development will provide future input to subsequent versions of the profile. There are also additional drivers external to the Data Standards Body that may result in changes occurring to the Information Security Profile. These are:

- **Consumer Data Right Rules**
  As for all CDR Standards, changes to the CDR Rules published by the ACCC may result in changes to the Information Security Profile. It is the obligation of the CDR Standards to implement the CDR Rules.
- **Consumer Data Right Registry**
  The design of the CDR Registry is being developed through collaboration of the various bodies tasked with implementing the CDR Regime. When this design is ready for open consultation the Information Security Profile may change to align with the published design.
- **Consumer Experience Testing**
  The CDR CX Stream is currently performing CX testing on a range of journeys including authentication and reauthorisation. The findings of this testing will result in changes to the Information Security Profile to ensure the best, most secure, experience for customers.
- **Legislation**
  The CDR Regime is subordinate to the establishing legislation. Any legislation passed by the Federal Government that impacts the CDR Regime will need to be accommodated. This may result in changes to the Information Security Profile.
- **Independent Security Review**
  As occurred for the previous draft and again suggested by community feedback, an independent security review of the Information Security Profile will be conducted. This is good practice and a similar process was undertaken after the December 2018 draft of the CDR Standards was published. The findings of this review may result in changes being made to the Information Security Profile.

In all cases consultation will be conducted with key community players to ensure the standards are reasonable and implementable within an appropriate timeframe.

Decision For Approval

NOTE: In the description of the decision below any clarifications, areas of continuing consideration and areas where change arising from external drivers is particularly likely are highlighted using a callout box such as this. These callouts should not be considered normative.

Information Security Profile Overview

This information security profile builds upon the foundations of the Financial-grade API Read Write Profile [FAPI-RW] and other standards relating to Open ID Connect 1.0 [OIDC].

For information on the specific normative references that underpin this profile refer to the Normative References section.

Symbols and Abbreviated terms

- API: Application Programming Interface
- CA: Certificate Authority
- CDR: Consumer Data Right
- CDR-SP: Consumer Data Right Security Profile
- CIBA: Client Initiated Backchannel Authentication
- CL: Credential Level
- DH: Data Holder
- DR: Data Recipient
- DTA: Digital Transformation Agency
- FAPI: Financial API
- HoK: Holder of Key
- JSON: The JavaScript Object Notation
- JWA: JSON Web Algorithms
- JWE: JSON Web Encryption
- JWK: JSON Web Key
- JWKS: JSON Web Key Set
- JWS: JSON Web Signing
- JWT: JSON Web Token
- IP: Identity Proofing
- LoA: Level of Assurance
- LoAs: Levels of Assurance
- MTLS: Mutually Authenticated Transport Layer Security
- OIDC: Open ID Connect
- PI: Personal Information
- PKI: Public Key Infrastructure
- PPID: Pairwise Pseudonymous Identifier
- REST: Representational State Transfer
- TDIF: Trusted Digital Identity Framework
- TLS: Transport Layer Security

- VoT: Vector of Trust

CDR Federation

The CDR Federation will facilitate the secure exchange of consumer data and federation metadata between multiple system entities that will assume one or more of the following roles:

- **Data Holder**: Multiple Data Holders will be supported.
- **Data Recipient**: Multiple Data Recipients will be supported.
- **Registry**: Only one registry will be supported and will be maintained by the Australian Competition and Consumer Commission (ACCC).

Data Holder

The Data Holder (DH) is a system entity that authenticates a consumer (resource owner or user), as part of an authorisation process initiated by a Data Recipient, and issues an authorisation for that Data Recipient to access the consumer's data via published APIs.

A Data Holder assumes the role of an [OIDC] OpenID Provider.

Data Recipient

A Data Recipient (DR) is system entity that is authorised by a Data Holder to access consumer resources (APIs). A Data Recipient MUST capture consumer consent prior to commencing an authorisation process with a Data Holder.

A Data Recipient MUST be accredited in order to participate in the CDR Federation. Accreditation rules for Data Recipients are beyond the scope of this artifact.

A Data Recipient assumes the role of an [OIDC] Relying Party (Client).

Registry

NOTE: This section is reflective of the positioning of the Registry in the CDR Rules Exposure Draft

The Registry is a central point of discovery for both Data Holders and Data Recipients. Data Holders and Data Recipients must be created as entities in the Registry in order for them to participate as members of the CDR Federation. The functionality of the Registry will include but will not be limited to:

- **Management of Identities and Access**: The Registry will allow registered persons, on behalf of Data Holders and Data Recipients, to manage the metadata of their associated organisations and systems.
- **Management of Certificates**: The Registry will facilitate the issuing, management and revocation of digital certificates.
- **Discoverability and Search**: The Registry will expose APIs and GUIs (Web applications) in order to support metadata queries across Registry entities.

Authentication Flows

This profile supports a subset of the authentication flows specified [OIDC] as constrained further by FAPI [FAPI].

Specifically the Hybrid Flow outlined at section 3.3 of [OIDC] is supported. This flow MUST be supported by Data Holders.

No other flows are currently supported.

OIDC Hybrid Flow

The [OIDC] Hybrid Flow is a type of redirection flow where the CDR Consumer's user agent is redirected from a Data Recipient's (Relying Party) web site to a Data Holder's Authorisation endpoint in the context of an [OIDC] authentication request. The Hybrid flow incorporates aspects of both the implicit flow and authorisation code flow detailed under [OIDC].

Only a response_type (see section 3 of [OIDC]) of code id_token SHALL be allowed.

The request_uri parameter SHALL NOT be supported.

NOTE: This process of customer authentication and the customer experience of the subsequent process to obtain customer consent has been the subject of significant consultation and feedback. To date, a consensus position has not been achieved to the satisfaction of the Data Standards Body.

As a result of this lack of consensus this decision does not contain additional specificity around the requirements for the authentication flow. Additional specificity will be added to the profile once a position on the authentication flow has been reached.

The following specific options for the authentication flow have been identified:

1. **Standard Redirect Flow**
   Username and password are captured in a redirected web page and consent is then obtained in the redirected web page.
2. **Redirect With OTP Flow**
   Username is captured in a redirected page. The Data Holder then provides a one-time password via another channel, which is then captured in the redirected page to authenticate the customer. Consent is then obtained in that same web page
3. **Redirect With Known Channel**
   Username is captured in a redirected page. Customer then proceeds to a known digital channel, authenticates and provides consent.
4. **Client Initiated Backchannel Authentication**
   A decoupled and asynchronous authentication flow that is defined by FAPI.
5. **CDR Specific Decoupled**
   A decoupled flow proposed by a community member where a one-time identifier is obtained from a known digital channel and then provided to the data recipient with consent being completed afterwards in an experience provided by the Data Holder.

Previously, a decision to pursue Redirect With Known Channel had been made by the Data Standards Body and a proposal for how this was to be implemented had been provided for consultation. This proposal is not included in this decision. CX testing of the flows described above is ongoing.

Client Authentication

Data Holders MUST support the authentication of Data Recipients using the private_key_jwt Client Authentication method specified at section 9 of [OIDC].

Data Holders MUST support the authentication of the CDR Register using the private_key_jwt Client Authentication method specified at section 9 of [OIDC].

Data Recipients MUST support the authentication of Data Holders using the private_key_jwt Client Authentication method specified at section 9 of [OIDC].

While MTLS is utilised for transaction security and as a Holder of Key mechanism the PKI Mutual TLS OAuth Client Authentication Method SHALL NOT be supported as the mechanism for client authentication.

private_key_jwt

The private_key_jwt authentication method is enabled through the delivery of an encoded [JWT] signed using the Client's private key and thus facilitates non-repudiation.

Client public keys MUST only be obtained from the CDR Register.

The [JWT] represents an assertion that MUST include the following claims:

- iss: The client ID of the bearer.
- sub: The client ID of the bearer.
- aud: The URL of the endpoint being invoked.
- exp: A JSON number representing the number of seconds from 1970-01-01T00:00:00Z to the UTC expiry time.
- jti: A unique identifier generated by the client for this authentication.

The following claims MAY be included:

- iat: A JSON number representing the number of seconds from 1970-01-01T00:00:00Z to the UTC issued at time.

When invoking a protected endpoint, the aforementioned assertion MUST be sent with the POST method and MUST include the following parameters:

- grant_type: This parameter MUST only be included when invoking the Token Endpoint and MUST be set to authorisation_code or client_credentials.
- code: This parameter MUST only be included when invoking the Token Endpoint after utilising the Hybrid Authentication flow. This is the value of the code parameter returned in the authorisation response.
- client_id: The ID of the calling Client.
- client_assertion_type: This MUST be set to urn:ietf:params:oauth:client-assertion-type:jwt-bearer.
- client_assertion: The encoded assertion JWT.

OIDC Client Types

In reference to the client types referenced in section 2.1 of [OAUTH2]:

- Confidential Clients MUST be supported under this profile.
- Public clients MUST NOT be supported.

Consent

Consent requirements will be communicated between the Data Recipient and Data Holder via the authorisation request object. The primary mechanism for capturing consent will be scopes and claims under [OIDC].

NOTE: Previously the CDR Information Security profile indicated that a separate API, a Consent API, for establishing complex consent prior to authorisation being initiated would be included.

Additional clarification and insight obtained from various sources that steer the standards has indicated the reasons for requiring a Consent API, such as the need to support fine-grained permissions or consent mutability, are no longer needed for the initial implementation of the CDR regime. As a result, support for a Consent API is not included in this decision.

It is possible that such a mechanism will be needed in the future, or maybe needed to support Data Holder specific extensions. If this is the case then a Consent API would need to be considered for inclusion in this profile.

It should be noted that, in considering this issue, it is a salient point that there is no current international standard for a Consent API. A number of candidate draft standards are being proposed to various organisations but these are still in progress. The draft candidate to the OpenID Foundation's Financial API (FAPI) Working Group can be found at:
https://bitbucket.org/openid/fapi/src/master/Financial_API_Lodging_Intent.md
It should be noted that this is a candidate for a pattern and does not include specifics for payloads or interfaces. These would have to be defined specifically for the CDR regime.

During consultation various community members indicated the preference for inclusion of the Consent ID for the initial implementation. This being considered by the Data Standards Body and will be the subject of further consultation. To help focus consultation the following options for a Consent API have been identified.

Option 1: Defer inclusion of a Consent API until a requirement exists
In this option the inclusion of a Consent API will be deferred until a later date when a specific requirement is introduced that requires such a pattern to be adopted. Until this time the profile would be managed to ensure that the inclusion of the Consent API would be a non-breaking change and therefore able to be easily accommodated.

Option 2: Include a Consent API as an optional mechanism
In this option the specifics of a CDR Consent API would be defined but would be defined as optional. No mandated CDR requirements would be dependent on the Consent API to function. The operation of the Consent API in the context of the CDR would, however, be clearly defined and if a Data Holder wished to include the pattern to support extension functionality (such as payment initiation) this could be accommodated.

Option 3: Include a Consent API as a mandatory mechanism
Equivalent to option 2 except that the implementation of the Consent API would be mandatory. The initial definition of the Consent API payload would, however, be minimalistic until a specific need is identified that would result in fields being added to the payload.

The previously stated preference of the Data Standards Body is to refrain from mandating the Consent API until a standard has been formally adopted or until a specific requirement exists.

Scopes and Claims

CDR Data Scopes

The CDR specific scopes are as follows:

Scope Name	Scope ID	Description
Basic Bank Account Data	bank_basic_accounts	This scope would allow for the third party to access basic information of the customer's accounts. Includes simple account information including balance. Does not include account identifiers, product information or transaction data.
Detailed Bank Account Data	bank_detailed_accounts	This scope would allow for the third party to access detailed information of the customer's accounts. This scope is effectively additional authorisation to the Basic Bank Account Data scope. Granting this authorisation only makes sense if the Bank Account Data scope is also authorised. Includes basic account information plus account identifiers and product information. Does not include transaction data.
Bank Transaction Data	bank_transactions	This scope would allow the third party to access transaction data for accounts. This scope is effectively additional authorisation to the Basic Bank Account Data scope. Granting this authorisation only makes sense if the Basic Bank Account Data scope is also authorised. Includes all account transaction data.
Bank Payee Data	bank_payees	This scope allows access to payee information stored by the customer. Includes payee information such as billers, international beneficiaries and domestic payees.
Bank Regular Payments	bank_regular_payments	The scope would allow the third party to access regular payments and associated data. Includes Direct Debits, Scheduled Payments.

Scope Name	Scope ID	Description
Basic Customer Data	common_basic_customer	The scope would allow the third party to access personally identifiable information about the customer. For retail customers this would be information about the customer themselves. For business customers it would imply the name of specific user but also information about the business. Includes name and occupation for individuals or name, business numbers and industry code for organisations
Detailed Customer Data	common_detailed_customer	The scope would allow the third party to access more detailed information about the customer. Includes the data available with the Basic Customer Data scope plus contact details. Includes basic data plus phone, email and address information.
Public	NA	Openly accessible information. A customer would never need to grant this scope. This scope is included so that end points that can be called without requiring authorisation can be identified. Includes access to openly available information such as generic product information.

Scopes

In addition to CDR data scopes the following scopes MUST be supported:

- ***openid***: As described as section 3.1.2.1 of [OIDC], this scope MUST be present on each authentication request.
- ***profile***: Data Holders MUST support the profile scope as described in section 5.4 of [OIDC]. This scope MAY be present on an authentication request.

Claims

The following normal [OIDC] claims MUST be supported. This list includes, but is not limited to, [OIDC] standard claims:

- ***sub***: Pairwise Pseudonymous Identifier (PPID) for the End-User at the Data Holder.
- ***acr***: Authentication Context Class Reference. MUST contain a valid ordinal LoA value.
- ***auth_time***: Time when the End-User authentication occurred. Its value is a JSON number representing the number of seconds from 1970-01-01T00:00:00Z to the UTC auth_time.
- ***name***: End-User's full name in displayable form including all name parts.
- ***given_name***: Given name(s) or first name(s) of the End-User.
- ***family_name***: Surname(s) or last name(s) of the End-User.
- ***updated_at***: Time the End-User's information was last updated. Its value is a JSON number representing the number of seconds from 1970-01-01T00:00:00Z to the UTC updated_at time.

- The following [VOT] claims MAY be supported:
- ***vot***: MUST contain a valid VoT value.
- ***vtm***: The [VOT] trustmark URI.

The following additional claims MUST be supported:

- ***refresh_token_expires_at***: indicates the date-time at which the most recently provided refresh token will expire. Its value MUST be a number containing a NumericDate value, as specified in section 2 of [JWT]. If no refresh token has been provided then a zero value should be returned.
- ***sharing_expires_at*:** indicates the date-time at which the current sharing arrangement will expire. Its value MUST be a number containing a NumericDate value, as specified in section 2 of [JWT]. If consent is not complete or a *sharing_duration* was not requested in the authorisation request object then a zero value should be returned.

NOTE: During consultation it was proposed that sharing and refresh token expiry would be obtained via additional fields on the token and introspection end points via a Sharing ID to be provided by the Data Holder. In response to feedback this position has been modified. The Sharing ID concept has now been eliminated and the expiration of sharing is now being accessed via an additional claim as described above.

This is approach is aligned to the use of the *http://openbanking.org.uk/refresh_token_expires_at* claim specified in the Read/Write Data Specification for the UK Open Banking standards.

Tokens

ID Token

ID Tokens are specified in section 2 of the [OIDC] standard. In accordance with [FAPI-RW], ID Tokens must be signed and encrypted when returned to a Data Recipient from both the Authorisation Endpoint and Token Endpoint.

As described under section 5.2.2 of the [FAPI-RW] profile, ID Tokens MUST include the following claims (in addition to the mandatory claims specified in section 2 of the [OIDC] standard) as part of Hybrid Flow authentication:

- nonce: String value used to associate a Client session with an ID Token.
- s_hash: Hash of the state value.
- c_hash: Hash of the authorisation_code value.

The c_hash value MUST be generated according to section 3.3.2.11 of [OIDC].
The s_hash value MUST be generated according to section 5.1 of [FAPI-RW].

ID Tokens MUST be signed by Data Holders as specified in section 8.6 of [FAPI-RW].

The ID Token returned from the Authorisation Endpoint MUST NOT contain any Personal Information (PI) claims.

An ID Token MUST not contain both a vot claim (see Vectors of Trust) and an acr claim.

If the ID Token contains a vot claim, it MUST also contain a vtm claim:

- vtm: The trustmark URI as specified in section 5 of [VOT] .

Access Token

Access Tokens MUST be used as specified in section 10.3 of [OAUTH2].

An Access Token MUST expire 10 minutes after the Data Holder issues it.

The process for refreshing an Access Token is described in section 12.1 of [OIDC].

Refresh Token

Refresh Tokens MUST be supported by Data Holders.

The usage of Refresh Tokens is specified in section 12 of [OIDC].

The expiration time for a Refresh Token MUST be set by the Data Holder.

Refresh Token expiration MAY be any length of time greater than 28 days but MUST NOT exceed the end of the duration of sharing consented to by the Consumer.

Data Holders MAY cycle Refresh Tokens when an Access Token is issued. If Refresh Token cycling is not performed then the Refresh Token MUST NOT expire before the expiration of the sharing consented by the Customer.

The revocation or expiration of the currently active refresh token should be understood to effectively revoke or expire the sharing arrangement as a whole.

Token Expiry

The expiry time for issued access tokens and refresh tokens must be deterministic for the Data Recipient.

In order to achieve this:

- The Data Holder MUST indicate the lifetime in seconds of the access token in the *expires_in* field of the JSON object returned by the token end-point (see section 4.2.2 of [OAUTH2]).
- The Data Holder MUST indicate the expiration time of the refresh token using the *refresh_token_expires_at* claim.

NOTE: During consultation it was proposed that sharing and refresh token expiry would be obtained via additional fields on the token and introspection end points via a Sharing ID to be provided by the Data Holder. In response to feedback this position has been modified. The Sharing ID concept has now been eliminated and the expiration of a refresh token is now being accessed via an additional claim as described above.

This is approach is aligned to the use of the *http://openbanking.org.uk/refresh_token_expires_at* claim specified in the Read/Write Data Specification for the UK Open Banking standards.

Identifiers and Subject Types

The identifier for an authenticated end-user (subject) MUST be passed in the sub claim of an ID Token and UserInfo response as defined by [OIDC].

The Data Holder MUST generate the sub value as a Pairwise Pseudonymous Identifier (PPID) as described in section 8 of [OIDC]. Furthermore, the identifier SHOULD also be unique relative to the scenario in which the end-user has authenticated. For example, the identifier generated for the same person when they are using a business account SHOULD be different to the identifier that is generated when that same individual is authorising as an individual.

It is RECOMMENDED that the sub value is generated as a universally unique Identifier (UUID) [RFC4122].

Levels of Assurance (LoAs)

Levels Of Assurance (LoAs), returned after a successful authentication MUST be represented in Single Ordinal form where a single LoA value is represented.

Data Holder's MUST support this mechanism.

Single Ordinal

A Single LoA value is carried in the acr claim which is described in section 2 of [OIDC].

- An LoA of 2 is represented by the URI: urn:cds.au:cdr:2
    - The authenticator used to attain this level MUST conform with at least the Credential Level CL1 rules specified under the Trusted Digital Identity Framework [TDIF] Authentication Credential Requirements specification.
- An LoA of 3 is represented by the URI: urn:cds.au:cdr:3
    - The authenticators used to attain this level MUST conform with the Credential Level CL2 rules specified under the Trusted Digital Identity Framework [TDIF] Authentication Credential Requirements specification.

READ operations SHALL only be allowed where at least an LoA of 2 has been achieved during the establishment of consent.

WRITE operations SHALL only be allowed where:

- At least an LoA of 3 has been achieved during the establishment of consent, or
- At least an LoA of 2 has been achieved during the establishment of consent and a subsequent challenge/response has resulted in an LoA of 3 being achieved within the lifespan of the current Access Token.

Transaction Security

Use of TLS

All HTTP calls MUST be made using HTTPS incorporating TLS >= 1.2. This MUST include calls to public, unauthenticated end points.

Use of MTLS

All backchannel communication between Data Recipient systems and Data Holder systems MUST incorporate, unless stated otherwise, MTLS as part of the TLS handshake:

- The presented Client transport certificate MUST be issued by the CDR Certificate Authority (CA). The Server MUST NOT trust Client transport certificates issued by other authorities.
- The presented Server transport certificate MUST be issued by the CDR Certificate Authority (CA). The Client MUST NOT trust Server transport certificates issued by other authorities.

End points for transferring CDR Data that are classified as not requiring authentication do not require the use of MTLS.

Holder of Key Mechanism

NOTE: As a clarification to questions raised during consultation this section asserts that resource requests must be validated to ensure the client certificate and access token match. This is an equivalent position to section 6.2.1 of the UK Open Banking Information Security Profile.

MTLS MUST be supported as a Holder of Key Mechanism.

OAUTB SHALL NOT be supported due to a lack of industry adoption.

MTLS Holder of Key allows issued tokens to be bound to a client certificate as specified in section 3 of [MTLS].

Ciphers

Only the following cipher suites SHALL be permitted in accordance with section 8.5 of [FAPI-RW]:

- 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

Request Object

The Request Object is a signed and encoded JWT specified in section 6.1 of [OIDC]. As per [FAPI-RW] section 5.2.2, the request parameter MUST be present on requests to the [OIDC] Hybrid Authorisation Endpoint. The Request Object enables [OIDC] requests to be passed in a single and self-contained parameter.

Request Objects MUST be signed by Data Recipients as specified in section 8.6 of [FAPI-RW].

Request Object references SHALL NOT be supported.

The iss claim SHALL NOT be supported as it duplicates the role of the client_id claim.

Requesting Sharing Duration

To facilitate the specification of the duration for consent to share CDR data that is approved by the consumer, a mechanism for the Data Recipient to specify a sharing duration to the Data Holder is required.

To accomplish this, the Data Holder MUST support an additional field in the authorisation request object named ***sharing_duration***. The ***sharing_duration*** field MUST be handled as follows:

- The value of the ***sharing_duration*** parameter will contain the requested duration for sharing, in seconds.
- If the ***sharing_duration*** value exceeds one year then a duration of one year will be assumed.
- If the ***sharing_duration*** value is zero or absent then once off access will be assumed and only an Access Token (without a Refresh Token) will be provided on successful authorisation.
- If the ***sharing_duration*** value is negative then the authorisation should fail.

The Data Recipient is able to obtain the expiration of sharing via the *sharing_expires_at* claim.

NOTE: During consultation it was proposed that sharing and refresh token expiry would be obtained via additional fields on the token and introspection end points via a Sharing ID to be provided by the Data Holder. In response to feedback this position has been modified. The Sharing ID concept has now been eliminated and the expiration of sharing is now being accessed via an additional claim as described above.

This is approach is aligned to the use of the *http://openbanking.org.uk/refresh_token_expires_at* claim specified in the Read/Write Data Specification for the UK Open Banking standards.

End Points

NOTE: This section does not include any end points that will be defined by the design of the CDR Registry. This includes end points related to Data Recipient registration, certificate revocation, Data Recipient discovery or Data Holder discovery. These specifics are beyond the scope of this decision.

OpenID Provider Configuration Endpoint

Hosted By	Data Holder
Transport Security	TLS
Client Authentication Required	No
Bearer Token Required	No

Data Holders MUST make their OpenID Provider Metadata available via a configuration endpoint as outlined in Section 3 and 4 of the OpenID Connect Discovery standards [OIDD].

Where a Data Holder is supporting Vectors of Trust [VOT] the published OpenID Provider metadata SHALL reflect that support.

At a minimum, the Data Provider metadata MUST include:

- issuer: URL that the Data Holder asserts as its Issuer Identifier.
- authorization_endpoint: URL of the Authorization Endpoint.
- token_endpoint: URL of the Token Endpoint.
- introspection_endpoint: URL of the Introspection Endpoint.
- revocation_endpoint: URL of the Revocation Endpoint.
- userinfo_endpoint: URL of the UserInfo Endpoint.
- jwks_uri: URL of the JWKS Endpoint.
- scopes_supported: This list of supported scopes.
- claims_supported: The list of supported claims.
- acr_values_supported: The supported ACR values.

Data Holders that support Vectors of Trust [VOT] MUST include:

- vot_values_supported: The list of supported component values.

Authorisation Endpoint

Hosted By	Data Holder
Transport Security	TLS
Client Authentication Required	No
Bearer Token Required	No

The requirements for the Authorisation Endpoint are specified in section 3.3.2 of [OIDC] and further specified under section 5.2.2 of [FAPI-RW]. This endpoint is invoked as part of the Hybrid Authentication flow.

Token Endpoint

Hosted By	Data Holder
Transport Security	MTLS
Client Authentication Required	Yes
Bearer Token Required	No

The requirements for the Token Endpoint are specified in section 3.3.3 of [OIDC].

To obtain an Access Token, an ID Token, and a Refresh Token, the Data Recipient sends a Token Request to the Token Endpoint.

Data Holders MUST support a Token Endpoint.

UserInfo Endpoint

Hosted By	Data Holder
Transport Security	MTLS
Client Authentication Required	No
Bearer Token Required	Yes

The requirements for the UserInfo Endpoint are specified in section 5.3 of [OIDC].

Data Holders MUST support a UserInfo Endpoint.

Introspection Endpoint

Hosted By	Data Holder
Transport Security	MTLS
Client Authentication Required	Yes
Bearer Token Required	No

Data Holders MUST implement an Introspection Endpoint to allow Data Recipients to determine the status and expiry date of Refresh Tokens. The requirements for an Introspection Endpoint are described in section 2 of [RFC7662].

Introspection of Refresh Tokens MUST be supported.

Introspection of Access Tokens and ID Tokens MUST NOT be supported.

An Introspection Endpoint Response SHALL only include the following fields:

- active: Boolean indicator of whether or not the presented token is currently active.
- exp: A JSON number representing the number of seconds from 1970-01-01T00:00:00Z to the UTC expiry time.

Token Revocation Endpoint

Hosted By	Data Holder & Data Recipient
Transport Security	MTLS
Client Authentication Required	Yes
Bearer Token Required	No

Data Holders and Data Recipients MUST implement a Token Revocation Endpoint as described in section 2 of [RFC7009].

Requirements for Data Holder implementations
The Revocation Endpoint serves as a revocation mechanism that allows a Data Recipient to invalidate its tokens as required to allow for token clean up. It also provides a mechanism for a Data Recipient to notify the Data Holder of the revocation of a sharing arrangement by the Customer in totality as required by the ACCC CDR Rules. This revocation will have been actioned by the Customer via the Data Recipient's consent dashboard as described in the ACCC CDR Rules.

Revocation of Refresh Tokens and Access Tokens MUST be supported.

If consent is withdrawn by a Customer in writing or by using the Data Holder's dashboard the Data Holder MUST use the Data Recipient's implementation of the revocation endpoint with the current Refresh Token to notify the Data Recipient.

Requirements for Data Recipient implementations
The Revocation Endpoint, when implemented by the Data Recipient allows a Data Holder to notify the Data Recipient of the revocation of a sharing arrangement by the Customer in totality as required by the ACCC CDR Rules. This revocation will have been actioned by the Customer via the Data Holder's consent dashboard as described in the ACCC CDR Rules.

Revocation of Access Tokens MUST be not be supported.

Revocation of Refresh Tokens MUST be supported and will be used to notify the Data Recipient of sharing revocation

If consent is withdrawn by a Customer in writing or by using the Data Recipient's dashboard the Data Recipient MUST use the Data Holder's implementation of the revocation endpoint with the current Refresh Token to notify the Data Holder.

Reauthorisation Mechanism

When an authorisation has expired due to the expiration of the time requested using the *sharing_duration* claim in the request object then authorisation needs to be re-established using the full authorisation process.

NOTE: The specification of a simplified flow for reauthorisation is intended and consultation has begin on this topic. Currently the Data Standards Body believes that more consultation on this topic is required to ensure the standards represent the best position for the regime. Once a single approach is decided upon the text for this section will be replaced with the new position.

To simplify further consultation the following two options have been identified as the focus for further feedback. The Data Standards Body does not have a specific recommendation or preference regarding these two options.

Option 1: Client Initiated Backchannel Authentication
FAPI defines a protocol for an asynchronous and de-coupled mechanism for a Data Recipient to request authentication from a Data Holder known as Client Initiated Backchannel Authentication (CIBA). The specification for CIBA can be found at:
https://bitbucket.org/openid/fapi/src/master/Financial_API_WD_CIBA.md?fileviewer=file-view-default

This mechanism could be used to initiate a reauthorisation of an existing consent that has been established for an extended duration.

Previous consultation elicited feedback that this mechanism is quite complex and the implementation costs for Data Recipients and Data Holders are unwarranted for the relatively simple use case of reauthorisation. Alternate feedback supported this option due to the fact that it is an established standard.

Option 2: CDR Specific Mechanism
An alternative option is to define a CDR specific mechanism for reauthorisation. The specifics of such a mechanism are to be determined but, as an example, it could consist of the following:

- an end point to be implemented by Data Holders that would be called by Data Recipients to initiate reauthorisation
- determination that reauthorisation has occurred would be obtained by a Data Recipient by requesting an *id_token* with the *sharing_expires_at* claim
- specifics of how the Data Recipient and Data Holder interact with the Customer to obtain consent for reauthorisation would be defined by the CDR CX Guidelines

This mechanism would be specific to the CDR regime and would not be supported by an external standard. Implementation of this mechanism would, however, be relatively simple.

Normative References

Reference (used to identify the reference in other proposals)	Type
[FAPI-R]	Financial-grade API - Part 1: Read Only API Security Profile: https://openid.net/specs/openid-financial-api-part-1.html
[FAPI-RW]	Financial-grade API - Part 2: Read and Write API Security Profile: https://openid.net/specs/openid-financial-api-part-2.html
[JSON]	The JavaScript Object Notation (JSON) Data Interchange Format: https://tools.ietf.org/html/rfc7159
[JWA]	JSON Web Algorithms (JWA): https://tools.ietf.org/html/draft-ietf-jose-json-web-algorithms-40
[JWK]	JSON Web Key (JWK): https://tools.ietf.org/html/rfc7517
[JWT]	JSON Web Token (JWT): https://tools.ietf.org/html/rfc7519
[JWS]	JSON Web Signature (JWS): https://tools.ietf.org/html/rfc7515
[JWE]	JSON Web Encryption (JWE): https://tools.ietf.org/html/rfc7516
[MTLS]	OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens: https://tools.ietf.org/id/draft-ietf-oauth-mtls-07.html

Reference (used to identify the reference in other proposals)	Type
[OAUTH2]	The OAuth 2.0 Authorization Framework: https://tools.ietf.org/html/rfc6749
[OIDC]	OpenID Connect Core 1.0 incorporating errata set 1: http://openid.net/specs/openid-connect-core-1_0.html
[OIDD]	OpenID Connect Discovery 1.0 incorporating errata set 1: http://openid.net/specs/openid-connect-discovery-1_0.html
[TDIF]	Digital Transformation Agency - Trusted Digital Identity Framework https://www.dta.gov.au/our-projects/digital-identity/join-identity-federation/accreditation-and-onboarding/trusted-digital-identity-framework
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels https://tools.ietf.org/html/rfc2119
[RFC7009]	OAuth 2.0 Token Revocation: https://tools.ietf.org/html/rfc7009
[RFC7523]	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants: https://tools.ietf.org/html/rfc7523
[RFC7662]	OAuth 2.0 Token Introspection: https://tools.ietf.org/html/rfc7662
[VOT]	Vectors of Trust, draft-richer-vectors-of-trust-15 https://tools.ietf.org/html/draft-richer-vectors-of-trust-15

Informative References

Reference (used to identify the reference in other proposals)	Type
[BCP195]	Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS): https://tools.ietf.org/html/bcp195
[CDR]	Consumer Data Right: https://www.accc.gov.au/focus-areas/consumer-data-right
[FAPI]	Financial-Grade API - Home Page https://openid.net/wg/fapi/
[RFC4122]	A Universally Unique Identifier (UUID) URN Namespace – version 4: https://tools.ietf.org/html/rfc4122
[X.1254]	X.1254 - Entity authentication assurance framework: https://www.itu.int/rec/T-REC-X1254-201209-I/en