BIZA·IO

Response to Decision Proposal 182: InfoSec Uplift for Write

Introduction

The Data Standards Body (DSB) has provided Decision Proposal 182 (DP182) without recommendations to assess the uplift of the Information Security Profile that governs the Data Standards and therefore more broadly the CDR.

Within this proposal there is initial discussion around a transition from the underlying FAPI ID2 (Draft 6) profiles to the FAPI 1.0 (Final) profiles and then further questions with regard to the adoption of FAPI 2.0 family of specifications. As a provider of a software-as-a-service solution for Data Holders ("Biza.io Platform" or "Biza.io HaaS") we have provided initial feedback with respect to the analysis conducted by the DSB as well as regarding protocol support related to our recommendations.

Standards Evolution

Biza.io feels it is important to point out that the FAPI profiles are the result of evolution occurring within the open data and open banking spaces and, therefore, the naming and approach to their definition has evolved in accordance.

Specifically, within the *Context* part of DP182, various references to *Read* vs. *Write* have been made. The reality is that, as demonstrated by the final naming of FAPI 1.0, the definition of a security profile based on whether the operation was of read or write type is unsuitable for the use cases that jurisdictions (including Australia) are now looking to solve.

A simplified example of this is that accessing (*"read"*) an individual's identity information is potentially worth far more to a bad actor than potentially any payment that could be initiated (*"write"*) by that individual. Further, broadly speaking the FAPI 2.0 Baseline profile effectively *starts* with FAPI 1.0 Advanced as a baseline.

Because of these observations Biza.io is of the opinion that Standards should follow the sequential nature of their publication. In many respects the CDR InfoSec Profile is an *aberration* that is neither aligned to the historical international standard (FAPI1 ID2), the currently final standard (FAPI1 Final) or the future standard(s) (FAPI2).

Statements of Support

Biza.io supports the statements made within the GitHub thread for DP182 by:

- The Australian Banking Authority
- The OpenID Foundation
- Commonwealth Bank
- Westpac Bank
- Ping Identity

References

The following documents were used and are referenced during preparation of this analysis:

Document Name	Date (Version)	URL
Consumer Data Standards Information Security Profile (CDR InfoSec Profile)	June 30 2021 (1.11.0)	https://consumerdatastandardsaustralia.github.io/standards/#security-profile
Decision Proposal 182 (DP182)	7 July 2021 (Version 4)	https://github.com/ConsumerDataStandardsAustralia/standards/issues/182
FAPI 1.0 Part 1 Analysis	5 July 2021 (f6a473f)	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md
FAPI 1.0 Part 2 Analysis	5 July 2021 (f6a473f)	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part2-20210621.md
FAPI 2.0: Baseline (FAPI 2.0)	July 21 2021 (52842d1)	https://bitbucket.org/openid/fapi/src/master/FAPI_2_0_Baseline_Profile.md
FAPI Working Group	27 July 2021 (Unknown)	https://openid.net/wg/fapi/
Financial-grade API Security Profile 1.0 - Part 1: Baseline Financial-grade API Security Profile 1.0 - Part 2: Advanced (FAPI 1.0)	March 12 2021 (1.0 FINAL)	https://openid.net/specs/openid-financial-api-part-1-1_0.html https://openid.net/specs/openid-financial-api-part-2-1_0.html
Financial-grade API: Client Initiated Backchannel Authentication Profile (FAPI CIBA)	15 August 2019 (Draft-02)	https://openid.net/specs/openid-financial-api-ciba-ID1.html
Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)	October 17 2018 (Draft-02)	https://openid.net/specs/openid-financial-api-jarm.html
Grant Management for OAuth 2.0 (GM-API)	15 July 2021 (Draft-02)	https://openid.net/specs/fapi-grant-management-02.html

OAuth 2.0 Authorization Server Metadata (RFC8414)	June 2018 (RFC8414)	https://datatracker.ietf.org/doc/html/rfc8414
OAuth 2.0 Pushed Authorization Requests (PAR)	12 July 2021 (draft-ietf-oauth-par-09)	https://datatracker.ietf.org/doc/html/draft-ietf-oauth-par
OAuth 2.0 Rich Authorization Requests (RAR)	16 Nov 2021 (Draft 5)	https://datatracker.ietf.org/doc/html/draft-ietf-oauth-rar-05
OpenID Connect Discovery 1.0 (OIDD)	November 8 2014 (1.0 incorporating errata set 1)	https://openid.net/specs/openid-connect-discovery-1_0.html
OpenID Connect for Identity Assurance 1.0 (OIDC-IDA)	6 July 2021 (Draft 11)	https://openid.net/specs/openid-connect-4-identity-assurance-1_0.html
OpenID Continuous Access Evaluation Profile (OID-CAEP)	8 June 2021 (Draft 01)	https://openid.net/specs/openid-caep-specification-1_0.html
OpenID Foundation	Unknown (Unknown)	https://openid.net/foundation/
OpenID Shared Signals and Events Framework Specification 1.0 (OID-SSE)	8 June 2021 (Draft 01)	https://openid.net/specs/openid-sse-framework-1_0.html
Proof Key for Code Exchange by OAuth Public Clients (PKCE)	September 2015 (RFC7636)	https://datatracker.ietf.org/doc/html/rfc7636
RFC Style Guide (RFC7322)	September 2014 (RFC7322)	https://datatracker.ietf.org/doc/html/rfc7322
The OAuth 2.0 Authorization Framework (OAuth 2.0)	October 2012 (RFC6749)	https://datatracker.ietf.org/doc/html/rfc6749

Towards the Consent of Everything

Biza.io believes that the role of Information Security for Data is becoming increasingly intertwined with the associated Consent of that Data. While traditionally it was possible to look at securing data separately from permission to access the data the world, particularly with respect to consumer data, is quickly merging these concepts together.

Historically the DSB has treated many aspects of the consumer consent in isolation of the information security standards underlying it. This has resulted in only limited success with brittle business rules being implemented at presentation layers (collapsing of data clusters, deterministic behaviour between OpenID scopes and CDR scopes etc) and arbitrary parameters (ie. `sharing_duration` claims) triggering yet more brittle business rules for claim propagation (ie. `sharing_expires_at`). Added to this technical isolationism has been the introduction of complex rules frameworks, often driven more by political timelines than business-like common-sense combined with a general apathy to technological realisation.

These factors have resulted in an implementation which, while leading in a legislative context, belies the reality of its inflexibility in implementation. From Biza.io's perspective the window to correct course on this journey is closing, particularly in the context of international organisations developing solutions to solve the same problems.

In essence, Australia had “the jump” technically some two years ago but is, at best only a “nose ahead” now of other jurisdictions.

Defining a Consent Taxonomy

Biza.io believes that the key to achieving suitably defined consent is the definition of a consent taxonomy that can be uniformly defined, expanded upon, and adopted across industries using existing (RAR) technical standards coupled with an appropriate legal framework.

Biza.io’s preference for the format of such a taxonomy would be in the form of a JSON Schema with sufficient documentation to allow for a decision engine to reliably parse both the request and response content. Once such a taxonomy is defined it would become increasingly easier to appropriately define common use cases such as those hypothesised within DP183.

Biza.io has elected to provide examples of such a taxonomy in its response to DP183 to demonstrate how this could be applied. Biza.io provides examples of potential `authorization_details` (RAR) with a type of `cdr_sharing_arrangement_v2`.

It is worth noting that Biza.io *already* manages CDR Arrangements using a RAR type of `cdr_sharing_arrangement_v1`. That is to say, Biza.io has *already* defined a consent taxonomy within its own product and effectively bridged the brittle Data Standards approach into it.

Suffice to say that Biza.io's HaaS environment *already* supports Rich Authorisation Requests making the potential transition impacts negligible for our customer base.

Tracking Consent State

In addition to the need to be able to uniformly define consent in a machine processable way, Biza.io also believes that the consumer experience will not be complete until it is possible for counterparties (ie. Data Recipients etc) to discover details about existing arrangements.

To solve for this problem Biza.io personnel have developed in collaboration with other FAPI Working Group members the emerging Grant Management for OAuth 2.0 (GM-API) specification. GM-API describes a way, via an OAuth2 extension, of inspecting arrangements ("grants"), both current and expired using their unique identifier, rather than cycling through bound refresh tokens. This allows for a Data Recipient to dynamically receive state changes including extensions to sharing, additions/removals of permissions or basically any other suitably described parameter within the initial RAR response.

As part of this authorship participation and in part due to a parallel build of our HaaS platform, Biza.io *already* manages arrangements internally using this mechanism and as a consequence can commit to support for this specification immediately.

Ensuring Multi-Device and Holder Trusted App Support

Biza.io believes a critical component of continuing enhancement of trust within the CDR ecosystem is the integration of secondary factors into existing, trusted, digital experiences Holders already provide. A simplified example of this is the use of a push notification for approval of sharing, payment or otherwise via a customer's existing internet banking experience. Such a pattern is quite familiar to users, particularly with the rapid adoption of authenticator applications such as those within the Microsoft Office 365 environment.

Because of this believe, Biza.io supports the adoption of the OpenID Connect Client-Initiated Backchannel Authentication Flow (CIBA) into the Data Standards. Biza.io is actively working on the implementation of CIBA and expects to have full support for this specification within the recommended adoption timelines outlined in our question responses.

Question Responses

What are the existing gaps or concerns with the information security profile?

As an actively contributing member of the FAPI Working Group and co-author of the emerging Grant Management API specification, Biza.io and its personnel strongly believe in the value of international standards alignment. Through such alignment an ecosystem can benefit from the lessons learned from other jurisdictions and combine this with a pool of talented specialists who are able to bring together ***centuries*** of accumulated experience to consider interoperability problems in the context of security ones.

Consequently, Biza.io's key concern with the existing information security profile is that it does not currently align with the internationally accepted specification and, due to ambiguity introduced during its authorship, is now difficult to maintain with respect to the evolution of security best practice.

Key gaps that exist within the existing information security profile include:

1. Lack of PKCE support
2. Lack of enforcement of PKCE with PAR
3. Lack of explicit constraints of request object lifespans

Nonetheless, Biza.io wishes to repeat once again, *categorically*, that we ***do not*** support cherry-picking of components of the FAPI profiles and instead strongly recommend complete alignment with the FAPI 1.0 Final specifications as a high priority task.

Finally, Biza.io is uncomfortable with the level of oversight the CDR InfoSec Profile currently has. It is not aware of a formal information security assessment having been conducted on the profile for some years nor is it aware of any published and credentialed involvement of information security experts in the continued development of the profile beyond the *volunteer* work conducted by parties such as specialists within institutions or solutions providers such as Biza.io.

What gaps or concerns with the information security profile would prevent voluntary extension to write operations by a data holder?

Biza.io does not consider information security in the context of read vs. write as read operations can have just as much, if not higher, impact as write operations. Indeed, such archaic separation is one which is deprecated thinking within working groups and an oversimplification of the problem space.

What we would instead highlight is that voluntary use cases are more likely in an environment with international alignment (along with easier implementation) which the realignment of the CDS to the FAPI 1.0 profile would deliver (particularly the adoption of PKCE).

What aspects of version 1.0 of the FAPI Advanced Security profile, if any, should be prioritised for adoption by the CDR?

Biza.io provides the following high-level summary of prioritisation while incorporating further enhancements within the timeline.

Immediately:

- We do not believe these changes will have any significant impact on the ecosystem at this stage:
  - Mandate Request Object lifespan constraints immediately
  - Mandate PAR `request_uri` reuse restrictions
  - Mandate multiple brands as separate issuers
- Introduce PKCE support and therefore `response_type` of code only (without ID Token)

Within 3 months:

- Mandate PAR only Request Object submission

Within 6 months:

- Mandate PKCE+PAR support
- Align PAR adoption to Draft-09
- Introduce optional FAPI CIBA support

Within 9 months:

- Mandate complete alignment to FAPI 1.0 Part 1: Baseline (Final) and FAPI 1.0 Part 2: Advanced (Final) profiles;
- Formally adopt as Optionally supported the FAPI 2.0 specifications :
  - FAPI 2.0: Baseline Security Profile
  - Grant Management for OAuth 2.0
- Deprecate FAPI 1.0 profiles

Within 15 months:

- Mandate complete adoption of FAPI 2.0 profiles;
- Retire FAPI 1.0: Final

What priority should be given to transitioning to FAPI 2.0?

Biza.io believes that the CDR is currently flirting with being “left behind” as other ecosystems are very close to achieving CDR feature equality already. In addition, we believe that the DSB has an opportunity to embrace the FAPI 2.0 family to solve challenges it is already aware of (fine-grained consent, “purpose based” consent, consent status and discovery).

While the DSB *could* develop solutions to these challenges outside of the FAPI 2.0 process, doing so would alienate the international community further and likely lead to already limited vendor adoption being reduced to negligible as vendors instead seek to find larger markets (such as the U.S) which are already moving towards the international standards being developed.

What additional patterns or normative standards should be considered for adoption to reduce the risk of write operations?

Once again, Biza.io believes risks are equal regardless of operation but that the infosec profiles for consent operations are more than security controls and increasingly pathways to complex consumer consent behaviours and aspirations.

Consequently, Biza.io recommends the adoption of the following Standards:

1. Proof Key for Code Exchange (PKCE):
   To ease implementation complexity and enhance security

2. Rich Authorisation Requests (RAR):
   To deliver a platform for fine grained consent

3. Grant Management for OAuth 2.0 (GM-API), of which Biza.io personnel are co-authors:
   To deliver arrangement discovery capability

4. Financial-grade API: Client Initiated Backchannel Authentication Profile:
   To allow for additional authentication mechanisms to achieve higher levels of authority

In addition Biza.io recommends the evaluation of the following Standards:

- OpenID Shared Signals and Events Framework (OID-SSE):
  As a potential framework for signal processing
- OpenID Continuous Access Evaluation Profile (OID-CAEP):
  As a potential profile for state change notifications between parties
- OpenID Connect for Identity Assurance 1.0 (OIDC-IDA):
  As a potential framework for Identity Assurance and/or KYC requirements

What additional changes, if any, that should be considered for maximising international operability?

Despite various assurances by the DSB, there has been a minimal amount of observed engagement between key data standards bodies (notably OpenID Foundation) and the DSB itself. There has been tacit interaction by the ACCC and a number of minor workshops but limited continuing engagement.

While the government may be engaging other government agencies in its endeavours, there continues to be a technical void in engagement with leading international standards bodies.

Biza.io remains disappointed by the DSB's apparent lack of engagement with these bodies and ultimately sees a missed opportunity for Australia to lead the world in the definition and adoption of complex consent patterns for consumer data sharing.

What steps could be taken by the DSB to assure the efficacy of the information security profile?

Biza.io recommends that the DSB re-baseline the Standards in a format which acts as a derivation of the underlying Standards rather than its current form which ambiguously redefines it. The Data Standards with relation to authorisation and consent should, in Biza.io's opinion, be a profile of those underlying standards and seek to therefore copy the structure and approach of those standards.

Put more bluntly, the DSB should ditch Slate and adopt markdown2rfc or similar tools to produce IANA aligned documentation (RFC7322). Roughly paraphrasing a starting point for such a document:

The authorisation server shall support the provisions specified in clause 5.2.2 of Financial-grade API Security Profile 1.0 - Part 2: Advanced. In addition, the authorisation server
1. shall……
2. shall…..
3. may….

In addition, Biza.io recommends that the DSB focus on conformance tools to ensure both Data Holders and Data Recipients implement solutions which deliver and enforce robust security controls.

Appendix 1: Upstream Standards Analysis

Biza.io thanks the DSB for the analysis conducted between FAPI 1.0 ID2 and FAPI 1.0 Final and PAR Draft-01 and Draft-08. As this analysis appears to make a number of statements regarding what is considered breaking changes, it seems important to provide commentary on these documents, particularly the parts highlighted as a change as they represent potential build impacts to implementers.

FAPI Part 1 ("Baseline")

Authorisation Code Reuse

Analysis Commentary	§ 5.2.2. (13): Previous authorisation codes MUST be rejected (previously this was a should). NOTE: This will not impact the CDS as it is already required
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#522-authorization-server
Biza.io Commentary	The commentary **incorrectly** states that the CDS **already** requires code reuse to be rejected. This is incorrect. The current Standards make no such statement and Biza.io is aware of implementations which do not implement this protection.
Impact to Biza.io implementations	None. Biza.io HaaS *already* enforces authorisation code reuse protections.

Support for OAuth2 Metadata (RFC8414)

Analysis Statement	§ 5.2.2. (22): Qualifies the requirements to follow [OIDD] for discovery metadata. NOTE: This will not impact the CDS, already required
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#522-authorization-server
Biza.io Commentary	We agree with this statement however wish to highlight that FAPI 2.0 currently mandates both OIDD and RFC8414 metadata.
Impact to Biza.io implementations	Minor. Biza.io HaaS currently supports OIDD but does not yet support RFC8414 metadata.

Scopes in Token Response

Analysis Commentary	§ 5.2.2 (15): changes the requirement to such that scopes must be returned with the access token if "the request was passed in the front channel and was not integrity protected". This will likely have breaking impacts to ADR clients that rely on the scopes being present when the access token is requested via the back

	channel. It will mean that clients need to obtain the authorised list of scopes by calling the token endpoint or token introspection endpoint. The point of "integrity protected" also warrants discussion. There are significant benefits in the AS returning the authorised list of scopes to the client to ascertain the final consumer's directives for consent. Where a DH does not support a scope, the list will be a subset of what the client originally requested.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#522-authorization-server
Biza.io Response	There is some commentary about the scope parameter being returned in the token response. Biza.io makes the following observations: 1. The FAPI profile mandates the list of scopes be returned if it is exposed to manipulation in the front channel (ie. not sent via PAR) 2. The FAPI profile still mandates in the 4th note of 5.2.2 that the server must return the list of scopes if it is different from those request 3. The modification to not mandate scopes if unchanged was an alignment with the standard OAuth2 behaviour Also within the commentary is a comment that such an adoption may cause breaking changes for Data Recipients. Biza.io is not aware of *any* Data Recipient who are not already comparing requested scopes vs. granted scopes. Finally, a number of notes are made regarding RAR and discovery documents. We don't feel these are directly relevant to the stated intent of the comparison.
Impact to Biza.io implementations	None. Biza.io HaaS *already* includes scopes granted at all times.

Clients missing openid scope

Analysis Commentary	(*New section*) Not applicable.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#5223--clients-not-requesting-openid-scope
Biza.io Commentary	This section was added because with PKCE mandated on PAR endpoints it is no longer necessary to use the `code id_token` hybrid flow and therefore an ID Token is not required for completion of

	authorisation. On this basis the `openid` scope is not required unless an ID Token is explicitly desired. In an environment without the use of the hybrid flow Biza.io sees minimal value in an ID Token being requested (and therefore the `openid` or `profile` scopes being required) as the data sets provided by other Data Standards endpoints provide the same data and more.
Impact to Biza.io implementations	Minor. Biza.io HaaS *already* supports the use of PKCE but does not advertise it directly within CDR deployments.

BCP212 Withdrawal

Analysis Commentary	§ 5.2.3. (5): Requirement #5 has been withdrawn:
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#523--public-client
Biza.io Commentary	BCP212 wasn't withdrawn but considered an ambiguous duplicate of Section 7.5
Impact to Biza.io implementations	None.

Content-Type Header Requirement

Analysis Commentary	§ 6.2.1. (9): Content-type header requirement has changed from Content-Type: application/json; charset=UTF-8 to Content-Type: application/json. The CDS requires conformance to [RFC7231] which means the content type's media type must be application/json but Content-Type may include wildcard, and charsets. The interpretation is currently clear in the CDS that a DH can't reject a request where the Content-Type value is well-formed according to [RFC7231]. CDS requirement which requires consultation.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#621--protected-resources-provisions
Biza.io Commentary	This relates to a number of threads: • https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/issues/256 • https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/issues/27

	- https://bitbucket.org/openid/fapi/issues/236/charset-not-needed-for-application-json In essence the resultant change permits the charset to be defined or omitted.
Impact to Biza.io implementations	None. Biza.io HaaS will accept and if necessary, transform charsets on demand.

Resource Server x-fapi-customer-ip-address Behaviour

Analysis Commentary	§ 6.2.1. (13) Adds additional requirement: shall not reject requests with a `x-fapi-customer-ip-address` header containing a valid IPv4 or IPv6 address. This means that DHs cannot reject requests based on the contents of the `x-fapi-customer-ip-address` is a valid IPv4 or IPv6. That said, this may be a value inspected by the DHs WAF and rejections made based on its contents or the IP address of the requesting client for one or more security reasons.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#621--protected-resources-provisions
Biza.io Commentary	Biza.io disputes the value of *any* x-fapi header supplied by a Data Recipient beyond a correlation identifier for debugging on the basis that inspecting such headers would involve potentially trusting a bad actor. This same comment applies to Data Holders who implement WAF rules to make *any* type of decisioning based on a Data Recipients input headers because doing so would be easily forged and at best result in undefined behaviour - for instance blocking a Data Recipient because their customer originates from an IP which the Data Holder does not wish to accept. Biza.io notes that this is an active discussion in FAPI 2.0 (https://bitbucket.org/openid/fapi/issues/282/fapi-20-x-fapi-headers) but currently FAPI 2.0 does not include *any* x-fapi- headers.
Impact to Biza.io implementations	None. Biza.io HaaS will accept these headers but only actively utilises `x-fapi-correlation-id`.

TLS & DNSSEC Considerations

Analysis Commentary	• FAPI 1.0 Final includes statements regarding prevention of TLS stripping attacks. <snip quote> • Expected that the CDS will defer to FAPI 1.0 specs in this regard. DH feedback is warranted to understand any impacts DHs foresee to existing implementations.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#71--tls-and-dnssec-considerations
Biza.io Commentary	We note that the cdr.gov.au domain does not currently contain DNSSEC glue despite the risk being raised 18 months ago (https://github.com/cdr-register/register/issues/149#issuecomment-786961066) Fundamentally Biza.io's point of view is that if the Register itself is susceptible to DNS attacks then the ecosystem as a whole is exposed.
Impact to Biza.io implementations	Moderate. Biza.io HaaS *already* implements HTTP STS protections but does not currently mandate DNSSEC on customer endpoints.

Multiple Brands as Separate Tenants

Analysis Commentary	(*New section*) Multiple brands as separate tenants under one Authorization Server must use separate issuers. This may be done at the domain or path level.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part1-20210614.md#77--discovery--multiple-brands
Biza.io Commentary	It is still early days regarding legal entities with many different brands however this protection is important for blast radius containment and was also observed within the UK Open Banking ecosystem.
Impact to Biza.io implementations	None. Biza.io HaaS *already* uses seperate issuers for different brands.

FAPI Part 2 (“Advanced”)

ID Token and JARM

Analysis Commentary	5.1. Introduction Moves statements related to ID tokens as detached signatures to section 5.1.1 Still requires s_hash Permits JARM, which is detailed in 5.1.2 5.1.1. ID Token as Detached Signature No differences. 5.1.2. JWT Secured Authorization Response Mode for OAuth 2.0 (JARM) Moves majority of the statements from section 5.2.5 of Draft 06 into this section Defers some statements to the JARM spec - notably recommending the AuthZ Server should advertise supported response modes using the response_modes_supported metadata parameter
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part2-20210621.md#51--introduction
Biza.io Commentary	The primary reason for the migrations of these clauses are: JARM is only recommended where non-repudiation of requests is required. As a consequence it only exists in FAPI 2.0 Advanced profile which has not yet been accepted as an Implementers Draft. The FAPI WG broadly recommends PAR for the purposes of integrity protected request initialisation. ID Token’s may not *need* to be issued to be FAPI 1.0 compliant if an OP chooses to use PKCE exclusively and not supply the openid scope.
Impact to Biza.io implementations	N/A: Clarification only.

Request Object Expirations, Audience and Not Before Claims

Analysis Commentary	§ 5.2.2. (13): shall require the request object to contain an exp claim that has a lifetime of no longer than 60 minutes after the nbf claim (emphasis added) Breaking Change - most likely a config change to implementations but some IAM vendors don't currently cater for OOTB configuration of the exp validation lifetime.

	§ 5.2.2. (14): No change § 5.2.2. (15): NOTE: new clause. shall require the aud claim in the request object to be, or to be an array containing, the OP's Issuer Identifier URL Breaking Change - what was a SHOULD in [OIDC] is now a SHALL (must). The audience must be the OP's issuer identifier URL or an array that contains the OP's issuer identifier URL § 5.2.2. (17): NOTE: new clause. - Sets validation time requirement on the nbf claim: shall require the request object to contain an nbf claim that is no longer than 60 minutes in the past - Breaking Change ○ ADRs must provide the nbf claim with a value no longer than 60 minutes prior to the authorisation request ○ DHs must validate that the nbf claim's value is no longer than 60 minutes from receipt of the authorisation request
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part2-20210621.md#522--authorization-server
Biza.io Commentary	The mandating of a maximum request object lifespan using exp, a not before claim of nbf and an explicit aud that contains the OP Issuer Identifier URL was a *direct* result of observed attacks within active ecosystems using Request Object reuse. Biza.io believes most (possibly all) holders already enforce these controls and is not aware of any Data Recipients who experience problems due to these clauses because Biza.io HaaS already enforces all of them.
Impact to Biza.io implementations	None. Biza.io HaaS *already* enforces exp validation and aud enforcement.

PAR with PKCE

Analysis Commentary	§ 5.2.2. (18): NOTE: new change.

	- Requires ADRs to use PKCE where PAR is used for authorisation requests shall require PAR requests, if supported, to use PKCE (RFC7636) with S256 as the code challenge method. Breaking Change Currently PAR is used by the CDS without the need for PKCE. We either need to transition clients towards PKCE and, at least, as a minimum support a period of transition where the ID Token is used as a detached signature rather than PKCE as an alternative.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part2-20210621.md#522--authorization-server
Biza.io Commentary	Biza.io **strongly** supports the adoption of PKCE with PAR for a number of reasons: - PKCE is currently accepted as the security best practice for OAuth2 installations - There exists request object attacks for PAR without PKCE - Adoption of PKCE+PAR will, by definition, significantly reduce the implementation requirements for prospective ADR's as they will no longer be required to support hybrid flow with ID Token encryption
Impact to Biza.io implementations	This particular change is the largest single blocker to the Biza.io HaaS Platform being FAPI 1.0 Certified. Essentially, because the CDS permits PAR without PKCE, Biza.io cannot be both FAPI 1.0 compliant and CDS compliant. As a result the current CDS is having an explicit impact on Biza.io's ability to export its technology to countries fully aligned with FAPI 1.0. Currently Biza.io permits PAR requests without PKCE.

JARM Support

Analysis Commentary	§ 5.2.2.2. (1): "if the response_type value code is used in conjunction with the response_mode value jwt" then JWT secured authorisation responses are to be used in accordance with [JARM] **Impacts to the CDS.** Currently the CDS does not support JARM. This was originally a finding of the Fortian review and has also been recommended by the OIDF. Should the CDS transition to supporting JARM and PKCE exclusively and not the Hybrid Flow with response_type "code id_token"? Suggest that this change should be adopted. However, in doing so, there would be breaking change and a transition period required.

Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-fapi-part2-20210621.md#5222--jarm
Biza.io Commentary	As stated previously JARM is no longer recommended unless required for non-repudiation. There is significant complexity with implementing JARM which is entirely avoided by the much simpler to implement combination of PAR+PKCE. As it stands the FAPI 2.0: Baseline profile does not attempt to solve for non-repudiation and so therefore JARM is not introduced until the Advanced profile. While both the Fortian review and OIDF at one stage recommended it as a front-channel integrity protection, this was nearly **2 years ago**, a long time within this space. Biza.io would note that rather than JARM being adopted, the DSB should instead consider this a lesson of how quickly deprecation can occur. In addition, Biza.io does not see value in continuing to support the hybrid flow into the future, nor an authorisation flow outside of PKCE and, in fact, sees such support (of Hybrid) or adoption (of JARM) as a likely constraint to the adoption of the CDR itself.
Impact to Biza.io implementations	Undefined. The Biza.io HaaS platform does not currently intend to support JARM requests until such time as non-repudiation of requests is required. Fundamentally Biza.io is focused on FAPI 2.0 compliance for which JARM only exists in the draft, non-ratified Advanced Profile.

PAR Draft-8

Endpoint Auth Method

Analysis Commentary	§ 2. &para; 4: Introduces client authentication can be negotiated via "token_endpoint_auth_methods_supported" (currently supported in the CDS) or "token_endpoint_auth_method" (this also existed in PAR Draft 01) NOTE: For completeness, it may be beneficial for the CTS to provide test cases that verify correct implementation for client and server of the "token_endpoint_auth_methods_supported" and "token_endpoint_auth_method" parameters including where both are provided. CDS uses "private_key_jwt". Currently the behaviour is ambiguous if the AS provides different values in these two metadata parameters and one is not "private_key_jwt".
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-par-20210704.md#2--pushed-authorization-request-endpoint
Biza.io Commentary	We note that the `token_endpoint_auth_methods_supported` metadata is within the discovery document while the `token_endpoint_auth_method` is the client metadata post registration. All these clauses do is restate OAuth2 norms and negotiation for Data Recipients is mandated to result in `private_key_jwt`.
Impact to Biza.io implementations	None. Biza.io HaaS *already* uses private_key_jwt exclusively.

Request Reuse

Analysis Commentary	§ 2.2.: Drops mention that the `request_uri` is intended for one time use. There is currently no clause in FAPI 1.0 that restricts this meaning that Authorisation Servers MAY implement the `request_uri` in such a way that it can be re-used within its lifetime or be recycled. This is still cryptographically bound to the oAuth client however there may be issues where it could be replayed within a short period of time. **Old clause** Since the request URI can be replayed, its lifetime SHOULD be short and preferably limited to one-time use. **NOTE:** Should the CDS prevent this, or is this replay not seen as an issue - it may be useful where the client attempts to go through the authorisation flow but encounters a technical issue and can replay the `request_uri` without re-staging it though this seems to be of little benefit. It is pre-authentication so again, there is limited opportunity for malicious use/replay attack. The only question is

	whether a simplified authentication flow may be impacted if the consumer is not required to authenticate.
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-par-20210704.md#22--successful-response
Biza.io Commentary	We note that this SHOULD parameter is actively tested by the FAPI Conformance Suite. As a consequence Biza.io currently blocks re-use of consumed `request_uri` parameters. Further, as we already limit request object lifespans the `request_uri` returned is already time-bound. As Biza.io HaaS already enforces these constraints we have not observed any impacts experienced by active Data Recipients.
Impact to Biza.io implementations	None. Biza.io HaaS *already* restricts `request_uri` re-use and request object lifespans.

Client Redirect URIs

Analysis Commentary	2.4. Management of Client Redirect URIs **New section** Allows for the provision of per-request redirect_uris that have not been previously registered with the Authorisation Server. The authorization server MAY allow such clients to specify "redirect_uri" values that were not previously registered with the authorization server. This is not currently permitted in the CDR which requires valid redirect_uris to be registered. It is worth reviewing this in light of other provisions such as sector_identifier_uri and may also have implications for PPID generation if this allowance is adopted or considered in a future iteration of the standards. This may provide (with consideration) a way to deal with ADR SaaS / Outsourced Service Provider arrangements where the client is managed by a trusted third-part of the ADR
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-par-20210704.md#24--management-of-client-redirect-uris
Biza.io Commentary	FAPI 1.0 already constrains redirect_uri's to those which are pre-authorised.
Impact to Biza.io implementations	None. Biza.io HaaS already restricts redirect_uri's to those provided in the combinatorial subset of Register SSA, Registration Request and, if applicable, the Sector Identifier URI.

Mandatory PAR Support

Analysis Commentary	10.1. OAuth Authorization Server Metadata Adds the following registered property" require_pushed_authorization_requests - this will need to be considered if the CDS mandated authorisation requests always use PAR or DHs can choose not to support request objects sent by value delivered to the authorisation endpoint. There may be advantaged for DHs to secure & simplify their implementations to only support PAR and perform request object validation at the PAR endpoint after client authentication as opposed to receiving the request object at the authorisation endpoint. Feedback welcome from DHs and ADRs
Reference	https://github.com/ConsumerDataStandardsAustralia/standards-maintenance/blob/master/reviews/2021-05/analysis/analysis-par-20210704.md#101--oauth-authorization-server-metadata
Biza.io Commentary	Biza.io supports the adoption of PAR only request object submission. We are not aware of *any* Data Recipient implementation preferring Request Objects in the front-channel and indeed see such approaches as simultaneously exposing the ecosystem to historical attack methods *and* damaging the user experience in the event of failures (because requests aren't evaluated until the browser lands on the OP).
Impact to Biza.io implementations	Minimal. Biza.io HaaS *already* accepts PAR requests and would happily move to a PAR only environment.

About Biza.io

Biza.io are the market leaders in Data Holder solutions to the Consumer Data Right and are the only *pure-play* CDR vendor in Australia. Founded by the former Engineering Lead of the Data Standards Body (DSB), Biza.io has been involved in the Data Standards setting process since the very beginning and its personnel remain the largest non-government contributors to the consultations. In addition to its participation within the CDR, Biza.io is also a contributing member of the Financial-grade API (FAPI) Working Group, contributors to the FAPI 1.0 information security profile and co-authors of the Grant Management for OAuth 2.0 specification.

About Our Customers

As of July 2021, Biza.io is directly responsible for delivering, or heavily involved in the verification of, **one in three** of all active Data Holders. Beyond just a contractual engagement Biza.io considers all its customers partners in the journey toward open data. Our customers choose us to not only achieve compliance but to compete then command the consumer data ecosystem.