Data Standards Body

Technical Working Group

Decision Proposal 158 – Participant capability discovery

Contact: Mark Verstege

Publish Date: 5th February 2021

Feedback Conclusion Date: 5th March 2021

Context

In the development of recent versions of the standards the ongoing need for the management of change across a multitude of participants at different velocities has become clear.

Having mechanisms that allow ADRs and DHs to advertise the features that they support at a granular level enables different participants to implement support for changes to the standards at different rates.

Furthermore, it facilitates greater certainty in a many-to-many ecosystem where relying on hard compliance dates is not viable. Instead, it empowers serving participants (producers) to release new capability at a pace that aligns to their development cycles and business strategy; and empowers relying participants to integrate with solutions in a reliable, targeted manner with greater confidence that their solutions will be resilient and gracefully adapt to changes arising from standards obligations of many participants.
By decoupling the technical discovery mechanism of the ecosystem participants compliance obligation dates this reduces the need for hard inference of a compliance date being met by all participants at the same time. If a single participant misses one or more compliance dates this should not break the ecosystem or create hard failure points for relying participants.

A number of these mechanisms already exist through the use of end point versioning and the adoption of the OpenID Connect Discovery end point, however, there are potential gaps for specific types of change that needs to be addressed.

Whilst the OpenID Connect Discovery end point (refer to OpenID.Discovery and RFC8414) supports a well-defined way to advertise the metadata for an authorisation server, it is not designed to support metadata discovery for the resource server. As such, a method which enables CDR participants to publish metadata that describes their implementation beyond the authorisation server is extremely useful. This may be the same solution for both ADRs and DHs or, given their different security models and roles in the CDR, be solved through different mechanisms.

This decision proposal addresses the DSB's roadmap item published in the DSB Future Plan: https://github.com/ConsumerDataStandardsAustralia/future-plan/issues/5

This decision proposal contains changes to the API standards. Based on feedback to this decision proposal, a follow up decision proposal will be presented incorporating the feedback and target solution including any implementation considerations and proposed obligation dates.

Feedback received

Aspects of participant discoverability have previously been consulted on here: https://github.com/ConsumerDataStandardsAustralia/standards/issues/19

Feedback received in consultation of November 2020 changes to the data standards received feedback through implementation calls, maintenance iterations calls and responses to Decision Proposal 135 that a technical mechanism of feature discoverability was important for both Data Recipients and Data Holders where this allowed decoupling from compliance dates and inferred methods of determining availability of features in the ecosystem.

Decision To Be Made

Determine the types of metadata to be supported by participants for more robust integration, the method of advertising implementation metadata and which parties can access this metadata.

Identified Options

Mechanism of discovery

In conjunction with the existing discovery capability defined in the ecosystem, participants need a reliable way to discover the relevant metadata that describes the implementation they are interacting with.

It is expected that regardless of the mechanism, consideration of how often relying participants refresh the metadata will be important so their software correctly works correctly with the latest version of the implementation they are connecting to.

Option A: Well-known Resource Server Discovery Document

Participants publish their metadata which describes the characteristics and functionality of their implementation in relation to their CDR obligations and commercial extensions via a well-known end point. This is similar to the OpenID.Discovery end point but specific to CDR implementation concerns.

For example:

<participant_base_uri>/discovery/.well-known/cdr-configuration

Implications and considerations

Pros:

- Decouples security and CDR application logic concerns
- Allows CDR metadata to be hosted on an endpoint independent of their authorisation server

- Participants can host metadata on their infrastructure
- Allows participants to deploy within their existing DevOps processes
- Cacheable for serving participants

Cons:

- Requires ADRs as well as DHs to implement a new endpoint

Considerations:

- Needs to be accessible under a base URI at a predictable location
- Cache refreshing must be considered (frequency of clients accessing the metadata, and how participants discover when the document has changed)

Option A2: ADRs publish extended metadata using their Software Statement Assertion and Dynamic Client Registration

This option is an alternative option only relevant to ADRs and another of the options presented would need to be considered for DHs.

ADRs already register metadata with DHs using Dynamic Client Registration. Extending on this, the SSA is expanded to include additional implementation metadata so the ADR client can transfer all CDR business logic using the same mechanism. When the ADR makes changes to their implementation, they do so via the CDR Register and re-generate the SSA which they then update each DH with using Dynamic Client Registration.

Implications and considerations

Pros:

- This has the advantage that the ADR can leverage existing CDR Register capability and workflows
- Reduces caching concerns for DHs as they are notified of changes when they are made
- ADR changes are anticipated to be less frequent that DHs, so leveraging an existing implementation pattern can lower the work for ADRs
- Leverages existing metadata caching strategies

Cons:

- ADRs must update their SSA using Dynamic Client Registration with all connected DHs whenever they make a change: this places a burden on ADRs
- Dependent on the CDR Register to generate the SSA: likely to be restrictive with limited flexibility for ADRs to support commercial extensions within the CDR
- Requires CDR Register development and requires all ADRs and DHs to update their consumption of the CDR Register APIs
- Likely to be slower to propagate change, especially where an ADR is connected to many DHs they will have to recursively call each DH to update their client registration

Considerations:

- May create a higher rate of change and load on ADRs, DHs and the CDR Register as any change to an ADR implementation needs to flow through all three actors
- Limits access to trusted (accredited) participants only

Option B: CDR Register Participant Metadata APIs

Leveraging the existing APIs hosted by the CDR Register, the CDR Register is extended to support additional ADR and DH metadata that is available through the ADR and DH register APIs. Participants

must upload their metadata whenever any changes are published into production so that their metadata accurately describes their implementation.

Ideally the CDR Register provides an API interface for trusted participants to programmatically update their metadata with the CDR Register as part of their deployment processes.

Implications and considerations

Pros:

- Decouples security and CDR business logic concerns
- Leverage existing APIs hosted by the CDR Register
- Leverages existing metadata caching strategies

Cons:

- Changes depend on a third party (CDR Register development) and impact all participants
- Limited control for participants where metadata is updated through DevOps release processes
- Requires the CDR Register to provide an interface to capture the new metadata and continue to invest in updates

Considerations:

- Limits access to trusted participants only
- Requires CDR Register development and requires all ADRs and DHs to update their consumption of the CDR Register APIs
- This would create ongoing build considerations to the CDR Register to support changes
- May limit the ability of participants to develop commercial extensions and voluntary metadata

Option C: Extend the use of the OpenID Discovery end point to advertise CDR implementation capability

Data Holders publish CDR features and capability using their existing OpenID metadata discovery endpoint. The schema is extended to include the CDR specific implementation concerns. ADRs would be required to support an OpenID metadata discovery endpoint.

Implications and considerations

Pros:

- This has the advantage that DHs can extend an existing end point without the need to host another API

Cons:

- Couples security concerns with CDR business logic
- Requires DHs to customise their authorisation servers
- ADRs must implement a non-standard OpenID metadata discovery solution, or must use another method to advertise their implementation capability

Considerations:

- ADRs do not operate as authorisation servers within the CDR and don't currently have any obligations to publish authorisation server metadata

Option D: No change

No change is proposed under this option. Rather than provide a method of discoverability, the approach is considered on a case-by-case basis for each change and must adopt existing mechanisms to determine which participants are live with what functionality.

In this situation, ADRs and DHs would continue to infer the relevant implementation information of each participant through data standards obligation dates and existing mechanisms like endpoint versioning headers.

Implications and considerations

Pros:

- No changes required for ADRs and DHs

Cons:

- As the complexity of the ecosystem grows this will create a more brittle solution with greater reliance on hard obligation dates and inference
- Participants cannot advertise future changes or commercial extensions
- Not scalable across hundreds of ADRs and hundreds of DHs across many sectors
- ADRs cannot easily determine what obligations a given DH has implemented when a DH has been granted an extension or dispensation

Considerations:

- Feature discoverability is inferred through other implementation parameters or compliance / obligation dates

Categories of discoverable metadata

Primarily, the information to be shared as metadata should describe the implementation of each participant to facilitate robust and resilient transition across functional changes and obligations. This information would not include information security metadata where normative standards apply because it is assumed the OpenID.Discovery document will be the appropriate place for this sort of metadata. The categories below are not exhaustive – feedback from the community may identify other categories of information that might also be relevant. The DSB is interested in feedback on other information where participants would see value in having it exposed.

Table 1: Discoverable metadata

Metadata Category	Description	Participants Applied To
Version and build information	Describe the version of the ADR or DH implementation including metadata such as the server timezone and build version and build date.	ADRs DHs

Metadata Category	Description	Participants Applied To
Available endpoints	Metadata that describes which endpoints are supported by the participant including which versions and methods. Metadata includes commencement dates (the date the endpoint is scheduled to be available) and deprecation dates (the date the endpoint version is planned to be retired). In part it codifies future dated obligations at an implementation level. This allows clients to discover which versions are supported prior to API endpoint version negotiation as well as plan in advance for upcoming support of future versions. It also provides a mechanism to discover commercial extensions beyond the required set of CDR APIs.	ADRs DHs
Supported consent models	DHs describe what version of the CDR consent model they support and which aspects are planned for release or in production including scopes, normative standards and consent capability. This would include advertising features such a fine-grained consent, concurrent consent support, CDR Arrangement ID support, and any specific versions of the CDR consent model defined over time. It would also allow DHs to advertise whether action initiation is available (possibly by end point). Other data such as the time period of data and characteristics of the data may also be advertised.	DHs
Product categories and phases	Allows DHs to publish a Rules-aligned set of product support such as which product types and product phases the participant offers. This would reduce the need for phasing tables especially where individual DHs can apply for exemptions or	DHs
Functional capability	Participants can advertise the specific implementation capabilities of their service. For example, a DH may advertise which NPP service overlays they support.	ADRs and DHs
Customer types	Which designated customer types and models are supported (e.g. single account holders, joint account holders, delegated authorities). This allows ADRs to determine whether certain customer segments can be serviced and tailor the CX accordingly.	DHs
Authentication methods	Advertise any authentication methods that can be requested by ADRs for consumer authentication	DHs via OpenID.Discovery
Commercial / voluntary extensions	Advertise available APIs, consent capability, scopes, API schemas and the like to fully describe any commercial extensions beyond the core set of CDR APIs and required data.	ADRs and DHs
Developer Support	Publish locations of developer support including technical contacts, developer portal URIs, registration URIs, terms and conditions.	ADRs and DHs

Metadata Category	Description	Participants Applied To
Brand vs DH entity	Should the metadata be describable at the DH high level entity only or at the individual brand-level within a DH. If the latter, the metadata document would need to cater for multiple brand-level documents within the one document, or accessible as separate documents	DHs

Timeliness of metadata availability

Part of resilient transition as changes are made in software systems — especially a many-to-many software ecosystem — is advertising changes ahead of time to ensure relying participants can update their software ahead of changes to the systems they interface with. ADRs especially will benefit from knowing ahead of time, when individual DHs will release new capability, product obligations and technical interfaces so they can plan and develop their client software accordingly to manage consumer expectation and improve consumer experience.
With each of these options, a period of transition is still important so clients can continue to rely on their existing integrations and phase in adoption of new functionality. This enables trusted participants to discover the addition of new features now available, plan uplift of their integration and transition in an accepted period of time to utilise new features where existing features are to be deprecated.

Option A: Support advertising future capability

The discovery mechanism allows participants to describe current state and future state functionality so changes that are planned for production release can be advertised ahead of time. This might include phasing considerations where APIs are migrated over time to different information security models, scopes or payload responses.

Future changes would be required to be made available with an agreed lead time in advance (such as 24 hours).

Implications and considerations

Pros:

- Increases certainty and planning for all participants
- Relying participants can build transition logic ahead of time and accelerate build of key functionality when required
- Promotes competitive tension as participants who go to market earlier with new changes will offer more functionality and may be more attractive to integration
- Works for commercial extensions as well as core CDR capability
- Reduces the number of times ADRs need to poll DHs to check if the metadata is updated / changing

Cons:

- Requires participants to plan changes ahead of time and know their target release dates
- Presents challenges where rollbacks or hotfixes are unscheduled for release into production

Considerations:

- Discoverability would need to support start date (may change prior to go live but gives participants more certainty) and end date/deprecation dates of different versions or use version negotiation of the metadata responses similar to API versioning.
- Feedback on mandatory obligations to publish changes ahead of time with agreed lead time is being sought. Should participants be required to publish upcoming future changes with strict guidelines, or should discoverability only support future dating capability but leave it to the discretion of the participant implementing the changes?
- If a participant delays or changes their release of software, are there any compliance considerations?

Option B: Publish new capability only when it is available

Participants only publish their current capability including any interfaces/capability being deprecated. Relying participants build their software to a known set of discovery mechanisms that are described in the data standards but do not know ahead of the release of new software when a given participant will go live with that capability other than any hard compliance obligations (latest dates for participants to be compliant).

Implications and considerations

Pros:

- Less work for DHs and less impact where changes to release schedules (e.g. delays) create higher change management overheads

Cons:

- Participants do not know what changes will be released ahead of time which may increase the likelihood of integration issues
- ADRs need to regularly poll the DH end point to check for changes

Considerations:

- Another way to look at this option is a non-mandatory Option A. The metadata discovery document is specified such that it allows for current, deprecated and future changes to be advertised but does not enforce mandatory obligations for participants to advertise future changes prior to them going live. Where participants want to advertise planned changes, they can do so.
- Caching considerations need to be factored in because ADRs will need to regularly check whether changes have occurred – this may also increase load on DH systems.
- API endpoints may benefit by supporting the metadata document version as a request/response header.
- Caching overheads could be reduced by setting clear expectations that existing capability must be supported over a transition period that does not result in hard break points.

Current Recommendation

It is recommended that a well-known resource server discovery document be specified in JSON schema (Option A for "mechanisms of discovery") that allows both ADRs and DHs to describe their implementations. This schema should allow participants to describe planned (future) changes but not create any binding obligations for participants to publish these future changes (Option B for “timeliness of metadata availability”) ahead of production release. In other words, participants would be required to advertise their current implementations including deprecated capability but,

although encouraged, would not be required to publish implementation metadata ahead of time. The reasoning is that the standards would accommodate reasonable transition periods guaranteeing time for adoption of new features before older capability is deprecated.

It is also recommended that this be accessible to any trusted participant (e.g. ADRs and DHs can call any other DHs discovery endpoint) but excludes public third party clients from discovering detailed implementation information.

Feedback is specifically sought on the following questions:

1. Should the method of publishing metadata be the same for ADRs and DHs?
2. Is it useful that ADRs can discover the metadata for another ADR?
   Specifically, to support ADR to ADR interactions including disclosure of consent and data between two accredited persons?
3. Similarly, is it useful that DHs can discover the metadata of other DHs?
   For example, in white labelling arrangements allowing brand owner DH and white label DH to discover the implementation details of each?
4. What metadata is most useful for participants to know about to improve interoperability?
5. What expectations do participants have with respect to how early they should be notified of another participant's upcoming support of new capability?
6. What criteria determine what is available in a well-known discovery document versus participant APIs published on the CDR Register?
7. Should metadata discovery be restricted to accredited persons within the regime, or publicly available?
8. What considerations, if any, are required with respect to versioning of metadata publication?

Implementation Considerations

Implementation considerations will be fully considered based on feedback to the identified options or other options proposed in the feedback provided. Noting this, it is expected that transition of any standards arising from this consultation could be staged in phases as other targeted consultations arise. Largely, discoverability is a concern where larger new capability is introduced into the ecosystem requiring a period of technical transition for participants (both serving participants and relying participants).

It is expected that introducing enhanced discoverability as soon as possible will better enable all participants to thereby depend on a reliable mechanism to improve transition considerations across other changes introduced to the ecosystem.