Data Standards Body

Technical Working Group

Decision Proposal 240 – ADR Metrics

Contact: James Bligh

Publish Date: 1st March 2022

Feedback Conclusion Date: <TBA>

Context

During 2021 a strategy was developed for consulting on enhancements to the metrics collection mechanisms for the Consumer Data Right (CDR), including the Get Metrics API. Consultation on this strategy can be found at:
https://github.com/ConsumerDataStandardsAustralia/standards/issues/145

Consequently, consultations on endpoint metrics and brand aware metrics were conducted. Additional consultations, including a consultation on Consumer Experience metrics, were deferred due to the amount of consultation being undertaken in the second half of 2021.

The consultations undertaken can be found at:

- Endpoint metrics
  https://github.com/ConsumerDataStandardsAustralia/standards/issues/164
- Brand aware metrics
  https://github.com/ConsumerDataStandardsAustralia/standards/issues/165

The result of both consultations was:

- to take no immediate action, although it was flagged that endpoint metrics would be of value to the ACCC; and
- consider an alternative to obtain the metrics from Accredited Data Recipients (ADRs) rather than from Data Holders (DHs).

In the February meeting of the Data Standards Advisory Committee the issue of the quality of data being provided by DHs was raised by several members. This discussion highlighted that ADRs were having issues with the data provided by some DHs including non-compliant response payloads, poor quality data and data latency not commensurate with other digital channels.

It has been difficult to quantify and address these issues as there is no current mechanism for detailed and timely data relating to these types of issues to be reported. In addition, manual reporting is hindered by the fact that the data being transferred is private customer data and cannot be shared, even with the regulator, without explicit consent.

This decision proposal seeks to obtain feedback on a specific solution to address these concerns whereby a Get Metrics API that reports on DH data quality, to be hosted by ADRs, is added to the CDR standards.

Decision To Be Made

Determine whether an ADR hosted Get Metrics API that reports on DH performance, from the perspective of an invoking client system, be added to the CDR standards and, if so, what form it should take.

Identified Options

As this decision proposal explicitly seeks advice on a Get Metrics API to be hosted by ADRs solution options are not defined in this section. Instead, various options for the form and extent of the proposed API are outlined.

Hereafter the proposed API will be referred to as the ADR Metrics API.

Metrics to be included

To address the issues identified regarding data quality it is assumed that the data included in the ADR Metrics API response would be categorised by DH. Aggregated data across all DHs would not be considered helpful as it would not facilitate targeted remediation.

At a minimum the metrics included in the ADR Metrics API would need to include:

- Average response time for the period
- Number of invocations for the period
- Number of rejections arising from exceeded traffic thresholds for the period
- Number of error responses for the period
- Number of non-conformant response payloads for the period

In addition, the API response could also include the following:

- Metrics broken down by endpoint (aligned to the consultation on Endpoint Metrics)
- Specific proxy metrics to measure data latency such as the minimum age of the most recent bank transaction or energy billing event received
- Subjective assessment by the ADR of the quality of the data provided
- Drop off rates returning to the ADR during the consent flow
- Metrics related to security endpoints as well as resource endpoints
- The highest version of the endpoint requested and returned
- ADR counts related to customers such as number of consents created, expired or withdrawn

Feedback on the validity of these additional metrics, or other metrics that could be considered, would be welcome.

Optionality

The normal approach for the inclusion of APIs in the CDR standards is that implementation by the relevant participant is mandatory.

Considering that the metrics to be targeted by the ADR Metrics API is providing information on the performance of DHs it would be reasonable to consider making the implementation of this API optional.

This would also accommodate different types of ADRs now able to participate in the CDR regime. It may, for instance, be reasonable to make the ADR Metrics API mandatory for unrestricted ADRs but optional for sponsored ADRs.

Feedback on whether this API should be optional, mandatory, or conditional on ADR accreditation type would be welcome.

Manual reporting only

As an alternative to the definition and implementation of an API a JSON schema is defined that is then requested by the ACCC from specific ADRs on an ad hoc or periodic basis.

This approach would reduce the implementation costs for participants but would still allow for a known data set to be instrumented and delivered by ADRs when requested.

The downside of this approach is that the processes for requested, obtaining and processing metrics would be manual in nature and would not be able to be fully automated.

Phased implementation

The concerns related to data quality are current whereas the implementation of a new component of the CDR standards would take time and would not become useful until implemented.

As a result of this may be of value to consider a phasing approach, perhaps leveraging a variety of the options previously outlined, for the development of an ADR Metrics API. For example, implementation could be phased as follows:

Phase 1: ADRs manually provide data on an ad hoc basis to the ACCC on a voluntary basis. This data would be as JSON using the same schema as would be used if the data were delivered via API.

Phase 2: ADRs voluntarily implement the ADR Metrics API and the ACCC commences using these APIs to obtain metrics.

Phase 3: Mandatory implementation of ADR Metrics API occurs (possibly for specific types of ADR yet to be defined).

Current Recommendation

As a starting point the recommendation of the DSB is as follows:

1. An ADR Metrics API will be defined (see next section) and incorporated into the CDR standards but implementation of this API will be initially voluntary.
2. In the short term, before this API is implemented by any ADRs, ADRs may voluntarily submit metrics in the defined form to the ACCC to allow for issues with specific DHs to be reported to the ACCC for their follow up.
3. Data returned by the ADR Metrics API will be categorised by DH brand using the DH id as defined by the CDR Register (ie. *dataHolderBrandId*).
4. The ADR Metrics API will return the following data:
    - Average response time for the period per endpoint
    - Number of invocations for the period per endpoint
    - Number of rejections arising from exceeded traffic thresholds for the period per endpoint
    - Number of error responses for the period per endpoint
    - Highest endpoint version requested per endpoint
    - Highest endpoint version returned per endpoint
    - Number of non-conformant response payloads for the period per endpoint
    - Specific proxy metrics to measure data latency
    - Drop off rates returning to the ADR during the consent flow
5. Metrics for security endpoints will not be included in the ADR Metrics API
6. Data returned by the ADR Metrics API would be for the previous full calendar month
7. Until implementation is complete ADRs that wish to voluntarily report data would be encouraged to submit the data that would be requested via the ADR Metrics API directly to the ACCC

ADR Metrics API

High Level Information

Title	Obtain ADR metrics
HTTP Method	GET
URI	/admin/metrics/data-recipient
Security Scope	Private to ACCC Only
Pagination	Not supported
Specific Errors	No specific error payloads expected to be returned
Path Parameters	None
Query Parameters	None

Request Payload

Not applicable

Response Payloads

HTTP Response Code: 200 OK

Field	Type	Mandatory	Description
data	Object	Mandatory	
{			
dataHolders	Array of objects	Mandatory	An array with each element representing the metrics for a single data holder brand that the data recipient has interacted with
[{			
dataHolderBrandId	String	Mandatory	The ID of the data holder brand that these metrics apply to
consentMetrics	Object	Optional	Object containing metrics related to the consent authorisation flow
{			
initiationCount	NaturalNumber	Mandatory	Number of consent authorisations initiated with the data holder
responseCount	NaturalNumber	Mandatory	Of the authorisations initiated how many resulted in redirect back to the data recipient

Field	Type	Mandatory	Description
acceptanceCount	NaturalNumber	Mandatory	Of the authorisations redirected back to the data recipient how many indicated acceptance
tokenObtainedCount	NaturalNumber	Mandatory	Of the authorisations accepted how many resulted in the data recipient successfully obtaining an access and/or refresh token
}			
endpoints	Array of objects	Optional	Array of objects with each element representing the metrics for a specific API that was called by the
[{			
endpoint	Enum	Mandatory	The ID of the specific endpoint that the metrics apply to. Valid values yet to be defined
versionRequest	Positive Integer	Mandatory	The highest version of the endpoint requested (ie. the value of the x-v request header)
versionResponse	Positive Integer	Mandatory	The highest version of the endpoint returned (ie. the value of the x-v response header)
invocations	Positive Integer	Mandatory	The number of invocations of the endpoint
averageResponse	Number	Mandatory	The average response time of the endpoint
minorSchemaErrors	Number	Mandatory	The number of responses with minor schema errors that did not result in the payload being unusable or unreadable
fatalSchemaErrors	Number	Mandatory	The number of responses with fatal schema errors that made the payload unusable
errors	Array of objects	Mandatory	Error counts per HTTP status code (excludes successful statuses)
[{			
status	Positive Integer	Mandatory	The HTTP status for the error
count	Positive Integer	Mandatory	The number of invocations that resulted in this response status
}]			
}]			

Field	Type	Mandatory	Description
latencyMetrics	Array of objects	Optional	Array of objects containing metrics related to data latency
[{			
type	Enum	Mandatory	The type of data that latency is being measured for. Valid values are: • BANK_TRANSACTION Represents the latency of bank transactions obtained from any of the bank transaction APIs • ENERGY_BILLING Represents the latency of energy billing records obtained from any of the energy billing APIs
minimumAge	Positive Integer	Mandatory	The lowest record age of the applicable record in an API response from the specific data holder. The age of a record is measured in number of seconds between the record age and the time of the invocation. For instance, for bank transactions the age of the record is the difference in seconds between the time of the transaction API invocation and the most recent of *postingDateTime*, *valueDateTime* or *executionDateTime*. Note that the latency of records obtained from invocations that included filters that may prevent the most recent transaction from being returned should not be included.
}]			
}			
links	Object	Mandatory	
{			
self	URI	Mandatory	Fully qualified link to this API call
}			
meta			
{			
}			

Implementation Considerations

The implementation considerations for this proposal have been mainly addressed in the Identified Options section in the content related to optionality and phasing.

Any additional feedback related to implementation concerns and timing covering issues not raised in this decision proposal would be helpful in determining any final decision.