Data Standards Body
Technical Working Group
Decision Proposal 001 – API Principles

Contact: James Bligh

Publish Date: 25th July 2018

Feedback Conclusion Date: 10th August 2018

Context

The development of API definitions requires that any small decisions be made over a wide range of concerns. These decisions are sometimes trivial but in some cases there is no obvious option and some form of trade off must be made. In these situations a set of principles that articulate what is considered critical to the overall success of the standard can be very important to help inform the decision and ensure that the right trade offs are made.

These principles should be high level and they should change infrequently. The principles should also be applicable across the standard as a whole as it expands and evolves.

Decision To Be Made

The list of principles that will govern the API standard definition process will be determined.

Identified Options

Option 1 – Use UK Standards Principles

The principles used for the UK Open Banking standards are outlined in Appendix A.

As the Australian standards will be based on the UK standards the UK principles are an obvious starting point for consideration.

The UK principles are a combination of high level guidance (such as the use of RESTful APIs) and lower level guidance (such as the handling of idempotency and field optionality). The principles are also defined separately for each API group and change between versions. As a result the principles are not firm and therefore may not be useful in the Australian context without modification.

In addition, the UK principles do not address issues that have already been identified as important in the Australian context such as customer experience.

Option 2 – Submit To Principles From Another Organisation

Principles from another organisation could be adopted. Candidate organisations would include json:api and Google.

The benefit of leveraging principles from an external organisation is that there has already been extensive testing and consultation of these principles. The downside is that the context for the principles is usually tied to the needs of the specific organisation and these may not completely align with the needs of the Australian API standards.

Option 3 – Tailor Principles To Australian Context

Define a list of principles tailored to the Australian context drawing upon multiple sources including the UK standards.

Current Recommendation

The current recommendation is to follow option 3 and define a set of principles tailored to the specific needs of the Australian Consumer Data Standards.

The following principles, classified as Outcome Principles and Technical Principles, are proposed:

Outcome Principles

These principles articulate qualitative outcomes that the API definitions should seek to deliver.

Principle 1: APIs are secure

The API definitions will consider and incorporate the need for a high degree of security to protect customer data. This includes the risk of technical breach but also additional concerns of inadvertent data leakage through overly broad data payloads and scopes. The security of customer data is a first order outcome that the API standards must seek to deliver.

Principle 2: APIs use open standards

In order to promote widespread adoption, open standards that are robust and widely used in the industry will be used wherever possible.

Principle 3: APIs provide a good customer experience

The API definitions will consider and incorporate the customer experience implications. The APIs should support the creation of customer experiences that are simple and enticing to use.

Principle 4: APIs provide a good developer experience

To ensure that the entry hurdle for new developers is low the experience of the developers that are building clients using the APIs will be considered. The ability for a developer to easily understand

and write code using the APIs in modern development environments should be facilitated by the API standards.

Technical Principles

These principles articulate specific technical outcomes that the API definitions should seek to deliver.

Principle 5: APIs are RESTful

The API standards will adhere to RESTful API concepts where possible and sensible to do so. In particular the concepts of statelessness and resource orientation will be followed.

Principle 6: APIs are implementation agnostic

The underlying implementation of the APIs should not be constrained or driven by the API definitions and standards. Conversely, the underlying implementation choices should not be visible or derivable to the client applications using the APIs.

Principle 7: APIs are simple

As complexity will increase implementation costs for both providers and clients as well as reduce the utility of the APIs, API definitions should seek to be as simple as possible but no simpler.

Principle 8: APIs are performant

The API definitions should consider and incorporate performance implications during design ensuring that repeated calls are not necessary for simple use cases and that payload sizes do not introduce performance issues.

Principle 9: APIs are consistent

The API definitions across the full suite of APIs should be consistent with each other as much as possible. Where possible common data structures and patterns should be defined and reused.

Principle 10: APIs are extensible

The API definitions and standards should be built for extensibility. This extensibility should accommodate future APIs categories and industry sectors but it should also allow for extension by data providers to create unique, value add offerings to the ecosystem.

Appendices

Appendix A: UK Open Banking Principles

The principles from the Accounts/Transactions specification v1.1.0 can be found at:
https://openbanking.atlassian.net/wiki/spaces/DZ/pages/5785171/Account+and+Transaction+API+Specification+-+v1.1.0#AccountandTransactionAPISpecification-v1.1.0-DesignPrinciples

The principles from the Read/Write specification v2.0.0 can be found at:
https://openbanking.atlassian.net/wiki/spaces/DZ/pages/127009221/Read+Write+Data+API+Specification+-+v2.0.0#Read/WriteDataAPISpecification-v2.0.0-DesignPrinciples

An excerpt from the v2.0.0 specification is below:

RESTful APIs

The API adheres to RESTful API concepts where possible and sensible to do so.
However, the priority is to have an API that is simple to understand and easy to use. In instances where following RESTful principles would be convoluted and complex, the principles have not been followed.
References:

- The highest level Data Description Language used is the JSON Schema : http://json-schema.org/
- Best Practice has also been taken from the Data Description Language for APIs; JSON API : http://jsonapi.org/
- The Interface Description Language used is the Swagger Specification version 2.0 (also known as Open API) : http://swagger.io/

Standards

The OBIE principles for developing API standards:

- OBIE will adopt existing standards where relevant/appropriate to minimise re-inventing the wheel.
- The Standards currently being reviewed include ISO20022, and FAPI.
- OBIE will favour developer/user experience over and above adoption of existing Standards, in order to create a more future proof Standard.
- OBIE will work with other relevant bodies to align with, contribute to and/or adopt other Standards work, especially relating to creation of Standards around APIs and JSON payloads.

ISO 20022

The CMA Order requires the CMA9 Banks to be aligned with the Regulatory and Technical Standards (RTS) under PSD2.
A previous draft of the EBA RTS required that the interface "shall use ISO 20022 elements, components or approved message definitions". In keeping with that requirement, the API payloads are designed using the ISO 20022 message elements and components where available.
The principles we have applied to re-use of ISO message elements and components are:

- Where relevant - the API payloads have been **flattened** so that they are more developer friendly. This has been a request from the developer community, and the stakeholders involved in the design workshop.
- Only elements that are required for the functioning of the API endpoint will be included in the API payload. API endpoints are defined for specific use-cases (not to be generically extensible for all use-cases).
- We will modify ISO 20022 elements where the existing standard does not cater for an API context (such as filtering, pagination etc.). An example is having latitude and longitude in decimal format - as this is how developers will work with latitude and longitude; or using simple types (e.g., a single date-time field) instead of a complex type (e.g., a choice field with a nesting of date and time).

Extensibility

It is intended that the API flows will be extended to cater for more complex use-cases in subsequent releases - and we have kept this in mind during the design.

Idempotency

Idempotency is difficult to implement consistently and leverage consistently.
As a result, idempotency is used sparingly in the Open Banking API specifications; with a preference to allow TPPs to simply re-submit a request under failure conditions.
APIs have been defined to be idempotent, where not doing so would cause a poor PSU user-experience or increase false positive risk indicators.

Message Signing

Digital signatures will facilitate non-repudiation for Open Banking APIs.
However, the solution for digital signatures (if required in a future release) has been agreed and the approach required to achieve this is described in Basics / Message Signing.

Agnostic to Payment Schemes

The API will be designed so that it is agnostic to the underlying payment scheme that is responsible for carrying out the payment.
As a result, we will not design field lengths and payloads to only match the Faster Payments message, and will instead rely on the field lengths and definitions in ISO 20022. Due diligence has been carried out to ensure that the API has the necessary fields to function with Bacs payments - as per agreed scope.
We will provide further mapping guidance to ensure that differences are understood between the Open Banking Payment API standard, and FPS and Bacs schemes where applicable.

Status Codes

The API uses two status codes that serve two different purposes:

- The HTTP Status Code reflects the outcome of the API call (the HTTP operation on the resource). The Security Working Group has stated that granular error codes may expose threat vectors - so these are limited to the HTTP Status Codes.
- A Status field in some of the resource payloads reflects the status of resources.

Unique Identifiers (Id Fields)

A REST resource should have a unique identifier (e.g. a primary key) that may be used to identify the resource. These unique identifiers are used to construct URLs to identify and address specific resources.

However, considering that some of the resources described in these specifications do not have a primary key in the system of record, the Id field will be optional for some resources. An ASPSP that chooses to populate optional ID fields must ensure that the values are unique and immutable.

Definition of Optionality

For endpoints and fields within each resource, the following definitions apply:

- 'Mandatory' endpoints or fields marked **must** be implemented by the ASPSP.
- 'Conditional' endpoints or fields **must** be implemented by the ASPSP if these are made available to the PSU in the ASPSP's existing Online Channel (subject to Note 1 below).
- 'Optional' endpoints **may** be implemented by the ASPSP.

'Online Channel' refers to the superset of the ASPSPs website interface or mobile application (i.e. any information provided to the PSU in either channel).

Notes

1 It is up to each ASPSP to make their own regulatory interpretation, based on eg PSD2, as to which of the 'Conditional' endpoints and fields must be implemented.

2 ASPSPs are free to decide whether to implement any of the 'Optional' endpoints and fields.

ASPSPs **must** make documentation available to TPPs as to which conditional and optional endpoints and fields are implemented for this specification. The method for providing this documentation will be covered in the implementation guidelines.