FINANCIAL™ DATA EXCHANGE

Foundational Requirements for Data Recipients

Legal Notice

Financial Data Exchange, LLC (FDX) is a standards body and adopts this [document title] for general use among industry stakeholders. Many of the terms, however, are subject to additional interpretations under prevailing laws, industry norms, and/or governmental regulations. While referencing certain laws that may be applicable, readers, users, members, or any other parties should seek legal advice of counsel relating to their particular practices and applicable laws in the jurisdictions where they do business. See FDX's complete Legal Disclaimer located at http://www.financialdataexchange.org for other applicable disclaimers.

Revision History

Document Version	Notes	Date
1.0	Initial Document Release This document was created as a result of FDX RFC 0203 and incorporates the full contents of the RFC for public release.	**December 2022**

Contents

1 Introduction

Foundational Requirements have been established to ensure that FDX Certification applicants meet standards that protect Data Recipients and ensure proper interoperability as expected by the End Users. While these requirements vary based on the provider's level of use, they exist for all in the areas of **Functional** and **Non-Functional** requirements. Functional requirements refer to the successful interaction between *end user authorization*, the *FDX API*, *minimum call compliance*, and *data*. Non Functional requirements refer to *availability*, *performance*, *scalability*, and *security*.

This document is part of a series of certification documents and explains foundational requirements. Note that meeting these requirements alone is not sufficient for certification. Please refer to the Certification Model for an understanding of methodology and testing.

1.1 Terminology

Please refer to Taxonomy for a complete definition of terms used in this document. The relationship between data provider, recipient, agent, and end user are key concepts throughout FDX implementation.

Specific to this process, the term **Certification Case** refers to a use case scenario on a pre-production environment that is only applicable to certification.

1.2 Scope

The scope includes the following areas which are described in detail in the Requirements section:

Functional Requirements

- *User Experience (UX)/End User Authorization:*
    - Authorization End User URL
    - Responsive pages
    - Authorization Code
    - State parameter
    - FDX User Experience - Providers should comply with recommendations specified by the FDX User Experience and Consent document
- *API Endpoints*
    - Authorization endpoint
    - FDX Data API endpoints
- *Minimum Call Compliance*
    - FDX Maintenance endpoints
    - Data API
- *Test Data*

Non-Functional Requirements (Formerly Operational)

- *Availability*
- *Performance (Response time and SLAs)*
- *Scalability*
    - Scalability will not be certified in pre-production
    - Providers are required to provision production capacity to handle requester requirements, provided that the requester follows usage best practices (see section 2.2.3)
- *Security*
    - End user authentication
    - End user authorization
    - API connection security for authentication, authorization, and data transfers

1.3 Prerequisites

An FDX certification applicant must meet these prerequisites to be considered for certification:

- A pre-production environment must be available with the following functionality that is equivalent to production:
    - Test profiles for all the supported certification cases and account types, with the supported data set (see “Representative Test Data” in Section 2.1 below)
    - Support for the OAuth 2.0 User Experience for relevant screens and substantially similar content
    - Support for all relevant error codes
    - Support for all relevant end-points
- The certification applicant must employ the latest published FDX API (currently v4.0) or current supported version within the last 12 months (v3.0). All base URI should include the version that is implemented.
    - *Note: OFX is not in scope for this document*
- Documentation for FiAttributes when utilized

As a general practice, the pre-production environment should maintain configuration and test data prescribed within the scope for requirements. Availability and other Non-Functional requirements for the pre-production environment are listed in the “Non-Functional Requirements” section below.

2 Requirements

2.1 Assumption

- Any participant in the FDX ecosystem is good data partner, and steward of resources and security
- The compliance to, and enforcement of this certification is outside the scope of this RFC. All requirements are to be written based on an ability to be tested, but the methodology will not be defined here.
- All participants are expected to follow the laws and regulations of their local jurisdictions.
- A Data Receiving Entity may be either a Data Access Platform or a Data Recipient, this certification is aimed at the technical certification standards and best practices of data receiving entities who directly interact with Data Providers in the consumer permissioned data sharing ecosystem.
- All data shared in the FDX ecosystem is permissioned by the end user customer who owns the data.
- Governance, onboarding and certification of a Data Recipient who operates as a Data Access Platform is out of scope and will be handled in a follow-up RFC that has been approved by the TRC and Strategic Planning Taskforce.
- This certification is silent on any bilateral agreement between the parties mentioned.
- This certification is focused on Data Receiving Entities who are consuming data directly from an FDX API, and any other data access methods are out of scope for this RFC.

2.2 Definition of Certification

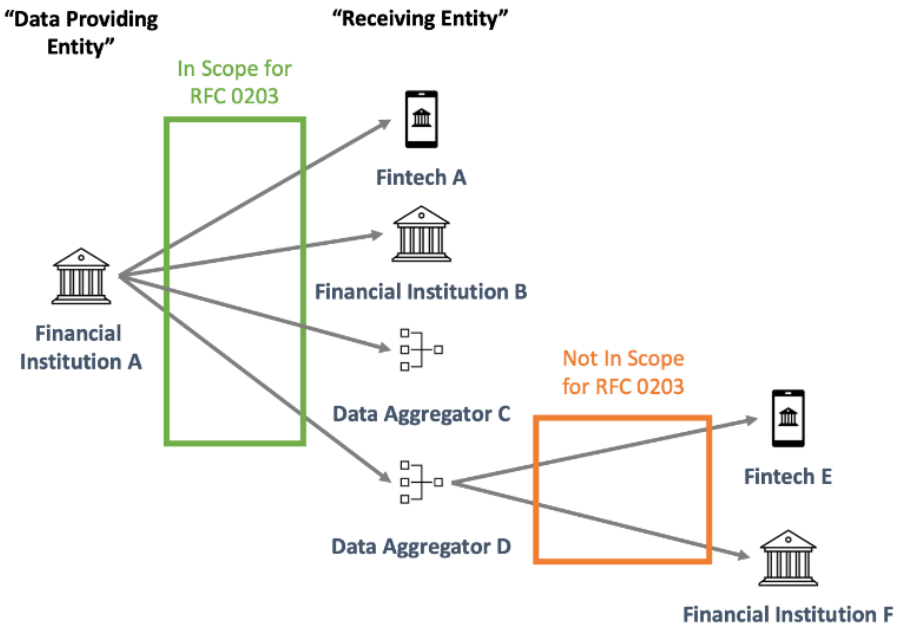
- Behavioral, technology, adoption, and interoperability API standards of Data Receiving Entity that are required to certify as an FDX data recipient, which will inform standing in the broader FDX community.
- Data Receiving Entity will need to continue to demonstrate that they are able to consume the latest version of the FDX standard as a Recipient.
- Dual role certification will be a gap, brought up to TRC.

2.2.1 Scope

This document will apply to any Data Recipients and Data Access Platforms who are acting as a Data Receiving Entity (as illustrated in the Scope diagram below) for consumer permissioned data access to a Data Provider. This document will describe the nature of the certification that will be required of each partner institution. These

institutions will need to confirm certification by conducting required testing as directed by FDX.

2.2.1.1 Scope Diagram



Definition of Participants – Refer to the Taxonomy of Permissioned Data Sharing v1.3 document for complete definitions. Some key relevant terms are as follows:

- Data Provider (DP) - Financial Institutions where end user goes to transact their financial transactions. These could included Banking transactions, Brokerage transactions, etc.
- Data Access Platform (DAP) - Firms that engage in the practice of retrieving data on behalf of another firm. Often the individual sharing their data does not know that there sharing via the DAP. An example of this would be firms like Yodlee who aggregate on behalf of other financial institutions.
- Data Recipient (DR) - Accredited data recipients can request, on behalf of a customer with the customer's consent, to collect and use data held by a data holder to provide a specific product or service. Data recipient may elect to become a Data Access Provider, if / when they are given permission to share data with a financial institution.
- End User (EU) - The individual who owns the account that is being shared.
- Data Receiving Entity - Any Entity (DAP, DP, DR, Intermediary, Sub-Intermediary) that is in the role of directly receiving data from an FDX API.
- Related FDX RFCs:
    - RFC 0012 - App ID with OAuth 2.0

- RFC 0153 - Recipient Registration Automation
- RFC 0206 - Recipient Registration With Delegation to an Ecosystem Registry
- RFC 0224 - Taxonomy additions from the Data Recipient TF

2.2.2 Authorization

All FDX API data is end user permissioned using the FDX approved authorization methods. Data access tokens will not be issued for data aggregation unless consent has been granted to the data provider by the customer.

In order for the customer to authorize data sharing with an FDX Data Receiving Entity, the Data Receiving Entity MUST redirect the user to an FDX approved authorization method.

2.2.2.1 Revocation

- Data Receiving Entities MUST offer end users a means of revoking access. The customer/application user MUST have a path to ensuring that the flow of any additional data can be halted. Not all Data Providers offer this functionality, but any application can serve this function. Please refer to the Data Recipient Revoke section of the User Experience Guidelines version 2.0 on pages 51 & 52.
- If a Data Provider offers the ability to revoke data access, then the Data Receiving Entity MUST enable the customer utilization of this functionality.

2.2.2.2 Notifications

- If a Data Provider has implemented the guidelines of RFC 0188 for event notifications, then the Data Receiving entity MUST demonstrate ability to both receive and consume the notification events.
- Using the revoke notification object, If a token has been revoked, then the application should be notified before the next attempted refresh is rejected.

2.2.2.3 Disclosures

In order to become certified, data recipients SHOULD clearly indicate to the user the details of the data being shared and duration of the consent. The intent of this guideline is to:

- Explain the business purpose
- The details of the data that is being shared (accounts, transactions, statements, etc.)
- State how long this consent will be active
- Tell the end user what happens next

- Provide an option to cancel the flow

2.2.2.4 Disclosure Process

- Data Receiver **MUST** communicate the details of the data being shared which is in scope for consent prior to redirect to the End User Authorization Journey
- Data Receiver **MUST** clearly communicate to the End User that they will be redirected to the data Provider to complete the Consent process
- Data Receiver **MUST** provide an option to cancel the process at any time
- Data Receiver **SHOULD** have the ability to communicate the duration of consent (persistent, time-based, one-time) in the disclosures process
- Data Receiver **SHOULD** explain the business purpose, the parties involved, and **MAY** provide a link to intermediary/DAP and Data Recipient privacy notices during the Disclosures process

2.2.2.5 Select a Provider

- Data Receiver **MUST** provide a means to search for the Data Provider
- Data Receiver **MUST** clearly represent the Data Provider by the known brand name and/or Logo.
- Data Receiver **MUST** provide a means to cancel the pre-consent authorization flow (RFC 0166)

2.2.2.6 Required Request Data Headers

All recommended FDX Data request headers are specified with the latest major FDX version (refer to section *6.1 Headers* in the API Specification v5.2 document).

Data Receivers **MUST** provide the following data request headers at a minimum:

- Cache-Control: no-cache, no-store
- Authorization: Bearer <bearer_token>
- Accept: application/json
- x-fapi-interaction-id: <unique_number/string>

2.2.2.7 Data Recipient Registration

- Data Recipient (client application) Registration SHOULD follow FDX RFC 0153 & 0206.
- Dynamic Data Recipient Registration support (where applicable to a Data Receiving Entity in the case of multiple application support): a Data Receiver MUST demonstrate the ability to participate in a dynamic registration model that follows FDX registration specifications. DREs with a single application SHOULD have this ability, but this is not directly applicable.

2.2.2.8 FDX Registry Participation

- In order to be certified (receive a badge), Data Receiving Entity MUST be part of the FDX directory listing (conformance directory that will post Data Provider information once they have been certified)

2.2.3 Security

Full Security and Control standards are the ultimate source of truth for FDX requirements. Familiarizing yourself with the *FDX API Security Model* document and the standards within is highly recommended.

2.2.3.1 Authorization

The foundational Security and Authorization building blocks for Consumer Permissioned Data Sharing are based off of OAuth 2.0, Open ID Connect, and FAPI.

- OAuth 2.0 - The underlying security foundation (RFC 6749)
- OIDC (OpenID Connect) - Identity layer add-on to OAuth 2.0 protocol (OpenID Connect)
- FAPI (Financial-grade API) - Advanced security Profile for OAuth and OIDC (FAPI Part 2)

The *FDX Foundational Requirements for Data Providers* document defines the core security requirements on the basis of OAuth 2.0 and OIDC (https://fdx.atlassian.net/wiki/spaces/FDX/pages/29655070/Foundational+Requirements+for+Data+Providers#2.2.4-Security-Requirements). FDX has since adopted the FAPI security standards on top of the baseline security requirements for Data Providers (RFC 0122 - FDX Adoption of FAPI). The latest FDX Security standards are kept up-to-date in the FDX API Security Model, which is a continually revised document in the FDX API release library.

The totality of the OAuth 2.0, OIDC and FAPI security standards give Data Providers a broad range of implementation options that Data Receivers must be capable of supporting. The following are the base Data Receiver Security Authorization requirements necessary for Certification.

2.2.4 Requirements originating in OAuth 2.0

- Data Receivers **MUST** support the OAuth 2.0 Authorization Code Grant (/authorize) flow, facilitating a user redirect an FDX approved authorization method.

- Data Receivers **MUST** support inclusion of Client_id and response_type=code in the Authorization end point (Provider requirement AS2)
- Data Receivers **MUST** support access token requests (/token) for both the authorization_code and refresh_token grant_types
- Data Receivers **MUST** support passing the grant_type, code or refresh_token, redirect_uri and client_id in the access token request (Provider requirement ID7)
- Data Receivers **SHOULD** immediately exchange the returned authorization code for a token set
- Data Receivers **SHOULD** immediately utilize an access token to call the FDX API at the Data Provider to retrieve the data for the consumer
- Data Receivers **SHOULD** encrypt and store a refresh_token for later use if the Data Recipient use case requires future data updates

2.2.5 Requirements originating in OpenID Connect

- Data Receivers **MUST** include an https redirect_uri in the Authorization redirect. Data Receivers **MUST** have the capability to pre-register the redirect_uri associated with the Data Recipient client_id (Requirements strengthened in FAPI). Note: All redirect URIs need to be base64 URL encoded to guarantee safe redirects.
- Data Receivers **MUST** be capable of of passing scope in the Authorization Redirect and **MUST** be capable of passing the Data Recipient required scope parameters available from the Data Provider if applicable
- Data Receivers **SHOULD** include the state parameter in the Authorization Redirect to maintain state between the authorization request and the response. The state should be in a cryptographically bound format to avoid cross-site forgery.

```
GET /authorize?
    response_type=code
    &scope=openid%20profile%20email
    &client_id=s6BhdRkqt3
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1
  Host: server.example.com
```

- Data Receivers **MUST** support the ability to receive a JWT id_token along with an access_token and refresh_token in the /token response from the Data Provider (Provider requirement ID1)
- In the event that a token is encrypted, Data Receivers **MUST** be capable of decrypting the id_token utilizing the agreed upon method with the Data Provider

2.2.6 Requirements Originating in FAPI

- Data Receiving Entities **MUST** support MTLS for communications with the Data Provider Authorization end point as a mechanism to support legitimate receivers of access tokens **OR** Data Receivers **MUST** support private_key_jwt as another potential means of authenticating as a confidential client to a Data Provider Authorization service. This is subject to changes in the Control Consideration Standards (DPoP is coming).
- Data Receiving Entities **MUST** support generating and passing a state parameter in the Authorization redirect which will be returned by the Data Provider in the Authorization response
- Data Receiving Entities **MUST** include the redirect_uri in the Authorization redirect. Data Receivers **MUST** have the capability to pre-register the redirect_uri with the Data Provider

2.2.6.1 Encryption

- In order to be certified Data Receiving Entity MUST prove that they are able to implement the mandatory guidelines of the FDX Sensitive Data Taskforce RFC
    - Acceptable encryption/decryption modalities are defined by the FDX Sensitive Data Taskforce.
    - RFC-0063 guides the implementation and will be helpful for entities who are looking for certification.
    - RFC-0011 lays out the specific encryption requirements of the Sensitive Data Taskforce
- Data Receiving Entities MUST have an encryption methodology in place to be 'certified' by FDX.
- Data Receiving Entities MUST have a decryption methodology in place to be 'certified' by FDX
- o Minimum encryption capabilities as defined in the current FDX API specs MUST be in place (FDX API v5.0 - Fall Release 2021)

2.2.6.2 Operational Requirements

Abiding by these operational requirements will contribute to the scalability availability & performance of the consumer permissioned data sharing ecosystem. If you want good data, be a good data partner.

From the FDX Foundational Requirements for Data Providers:

- Data receiving entities MUST capture the timeframe of the data that is received so that any use of the data includes understanding of the age of the data on a per account basis. For example, data as of for a particular set of attributes MUST be captured if available.
- Data Receiving Entities MUST limit invoking the range of successful lookback transaction history to a single query or session per registered application. DREs

MUST NOT request the full transaction history repeatedly. On subsequent calls, the range MUST be limited to the extent required to successfully capture all new transactions.

- Data Receiver **SHOULD** be able to distribute Batch refreshes over an agreed upon window with the Data Provider
- Data Receiving Entities MUST have the capability to observe and regiment defined rate limits that are set on a per provider basis. When a rate limit is breached, the FDX specified error code is 429.
- Excessive number of requests by a Data Receiving Entity. As an example: A Data Provider MAY place a limit requirement and/or preferred refresh timing on batch requests if there are infrastructure issues related to API access.
- If temporary limits are necessary, they SHOULD be communicated to Data Receiving Entities and Aggregators, both via the API itself and direct communication channels.
- Data Receiving Entities SHOULD demonstrate the capability to handle notifications from a Data Provider which inform the DRE that information has been updated. This capability allows data receiving entities to only request data when new information is available, conserving resources for everyone in the chain. Relates to RFC-0188 from the Notifications Taskforce
- Customer driven refreshes should not be limited, but Data Receiving Entities SHOULD demonstrate the ability to set up a configurable refresh rate that the customer may trigger during the same session.
- Data Receiver **SHOULD** have the capability to control and monitor batch traffic separately from End User initiated traffic

2.2.7 Data Recipient Responsibilities

2.2.7.1 Call Structure

- When a data recipient creates a new use case, the recipient SHOULD consider the minimum data calls required to fulfil the objective.
- Calling for token refresh MUST always be the first step of any aggregation. Active bearer/access tokens are required to aggregate. The refresh token will be used to request a new bearer/access token.
    - It is important to work with the individual institution to understand the correct lifespans and orders of token usage.
- Completion of data calls - all data calls MUST be scheduled to complete prior to the expiration of a newly issued bearer/access token.
- Call sequencing - FDX documentation always mentions which endpoints that you must call, but not the order that these calls must be made. It is expected that a Data Provider has the required sequencing of their API in the Data Provider documentation. Until this is defined within FDX, a Data Receiving Entity SHOULD follow the call sequencing laid out by the Data Provider documentation.
  Data recipients SHOULD place the calls in one of three sequences:

    - Call Sequence #1: /accounts?resultType=lightweight, iterate getting /account/{accountId}?resultType=details, and iterate calling each /account/{accountId}/transactions
    - Call Sequence #2: /accounts?resultType=details, then iterate calling each /account/{accountId}/transactions
    - Call Sequence #3: /accounts?resultType=details and transactionsIncluded=TRUE - returns a list of accounts details and embedded transactions for the range requested (single call)
- Data Recipients MUST demonstrate the ability to handle and respond to errors that indicate maximum date range has been exceeded. The response to this kind of error is to make calls that fit within the API date range window.

2.2.7.2 Accounts Retrieval

- Data receiving entities SHOULD provide only the relevant account or list of relevant accounts (accountId) when requesting details.
- Data Receivers SHOULD limit FDX account details API calls to the account types required for the Data Recipient use case.

2.2.7.3 Transactions Retrieval

- When required by the Data Provider API, Data Receiving Entities SHOULD send a start and end time for a transactions request. Start and end time parameters should be governed by the API specification and bilateral agreement in place. The specifics of these details are critical as they impact the resources of a Data Provider.
- Data Receiving Entities MUST, where available, check the transactionsIncluded flag to prevent calling the transactions endpoint for accounts which do not support transactions. When applicable, Data Receiving Entities MUST only request transactions on accounts where transactionsIncluded=TRUE
- If the lastActivityDate is present in the Data Provider response, Data Receiving Entities MUST ingest this information, and use it to determine whether a transactions call is required. This is intended to prevent unnecessary transactions requests.

2.2.7.3.1 Pagination

- Correct use of offset and limit for pagination is necessary. This should be governed by the API spec of the Data Provider. Data Providers will not always have consistent use of pagination and thus Data Receiving Entities MUST paginate anytime that an offset and/or page indicator is present.

2.2.7.4 Error Handling - FDX Error Codes

Refer to the Error Codes for Data Recieving Entities spreadsheet for a full list of error codes.

- A Data Receiving Entity MUST demonstrate the ability to ingest and handle error responses on API calls.
- A Data Receiving Entity MUST follow the FDX error handling guidelines.

2.2.7.4.1 FDX Error Handling Guidelines

Refer to the Certification Cases Errors section of the Data Provider Requirements document for error handling guidelines.

2.2.7.4.2 Partial Error Response

- Not all the data that was requested was returned by the data provider. When this takes place, a Data Receiving Entity MUST ingest and handle the error accordingly. HTTP Status Code 206.
    - 501, 401 errors are an example of this CERTIFICATION WG END 8/23/22

2.2.7.4.3 Maintenance Windows

- When receiving an FDX Maintenance Window error code, a Data Receiving Entity MUST demonstrate the ability to halt requests to the API.
- Data Receiving Entities SHOULD consult the FDX registry to be familiar with the standard maintenance windows of an FDX registered Data Provider API.

2.2.7.4.4 Retry After

- If the Data Provider provides retry-after instructions, then a Data Receiving Entity SHOULD honor the instructions. Retry-After instructions may be provided along with a 503 maintenance HTTP status code.

2.2.7.5 File Downloads

Image hosting can be both resource and cost intensive for Data Providers. As responsible data partners and stewards, Data Receiving entities should be cognizant of how this will impact the ecosystem.

2.2.7.6 Image Retrieval

2.2.7.6.1 Transaction Images

- Transaction images may be a check, receipt, invoice, etc related to a particular transaction or set of transactions.

- Within transactions, merchants can store an invoice. Some institutions use images here, and this would be a separate data call under the transactions endpoint. Guidance should come from individual Data Provider image retrieval specifications. (eg. Retrieval within observed windows and quantities may be the certification test on this requirement)

2.2.7.6.2 Document Downloads

- User requested statements SHOULD NOT be limited by the Data Receiving Entity
- Data Receiving Entities MUST limit requesting a document to a single query or session per registered application. DREs MUST NOT request an individual document repeatedly without cause (for example, deleted for security purposes).