FINANCIAL™ DATA EXCHANGE

Recipient Registration Guidelines

Version 1.2
November 2024

Legal Notice

Financial Data Exchange, LLC (FDX) is a standards body and adopts this Recipient Registration Guidelines 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
1.2	Edits to the scope table in section 2.1.1. Request Metadata, deprecated Data Clusters in place of Oauth scopes.	**Jan 2023**
1.1	Updated links, minor edits made.	**Jan 2023**
1.0	Initial document release created as a result of RFC 0153 Recipient Registration Automation.	**May 2022**

Contents

1.0 Introduction

1.1 Purpose

This document provides the FDX guidelines and best practices for Data Recipient Registration. In conjunction with other FDX publications such as the FDX API, Control considerations, and UX Guidelines, FDX recommends usage of these Recipient Registration guidelines to foster Automation, Interoperability, Optimization, Security, Compliance, Flexibility and Adoption across the data sharing ecosystem.

Recipient Application registration describes the process of providing Data Recipient information to Data Providers for Consent and/or Authorization and/or Consent Management purposes. Data Recipient registration solves three problems:

1. Data minimization: Data Recipients must receive only the data necessary to power the consumer-authorized use case.
2. User transparency: End users must be made aware of which entities they are engaging with, typically by displaying name and/or logo on relevant screens including authorization.
3. Ecosystem security: Data Providers and Data Access Platforms need visibility into downstream Data Recipients so as to mitigate security risks.

The Data Recipient Registration process is an essential yet often overlooked component of a Permissioned Data Sharing implementation. Current estimates indicate over 5000 Fintech and Financial Institution Data Recipients in the US data sharing landscape. Data Providers who are just getting started with Permissioned Data Sharing integrations, face the challenge of mass registering thousands of Data Recipients through dozens of Data Access Platform and Direct Recipient OAuth Clients. Additionally, there are dozens of new Data Recipients launching new Financial Application capabilities every month, each requiring registration through each available Provider.

Data Access Platforms face the challenge of mass registering hundreds or even thousands of Recipient application for each new Data Provider who adopts Permissioned Data Sharing. As FDX API and OAuth adoption grows, Data Access Platforms face a significant challenge registering new Data Recipients with hundreds, and eventually thousands of Financial Data Providers.

1.2 Scope

Data Recipient Registration Guidelines Version 1.0 will focus on Dynamic Client Registration (DCR) as foundational Data Provider capability for optimizing the Permissioned Data Sharing ecosystem. Dynamic Client Registration IETF RFC 7591 is an IETF OAuth 2.0 specification for registering Data Recipients through a Data Provider hosted API.

Dynamic Client Registration removes much of the human effort required to manage and maintain a rapidly growing recipient application ecosystem, while driving rapid adoption and process scalability. Providers who implement DCR according to the FDX Guidelines save on administrative time and costs involved with first time mass registrations, complex application approval processes, manual recipient scope updates, and simple Data Recipient metadata changes. DCR gives Data Access Platform the ability to automate the workflow for new Recipient registrations and mass migrations. In turn this saves on administrative time spent manually triggering registrations, fixing manual errors, following-up on approvals, managing Provider availability communications with Recipients, and managing scope changes.

Recipient Registration technical and integration methods are rapidly evolving. Additional Data Recipient Registration Guidelines and options will be published in future versions of this document. Delegated Recipient Registration and the FDX Registry are among the Recipient Registration options in the roadmap.

1.3 Definitions

See the FDX Taxonomy document for a full list of Permissioned Data Sharing terms. These are several of the common terms used in this document to define the Dynamic Client Registration Process.

End Users: include consumers, individuals acting in a business capacity, and entities, such as a business or other legal entity, who are giving permission to share their data.

Data Providers: the entities who hold End Users' Financial Account Information, including and without limitation to banks, credit unions, brokerages, bank and investment service providers, and direct lenders.

Data Recipients: service companies, applications (financial apps), financial institutions, products and services where End Users (on their own or through their End User Delegates) manage or act on their finances, whether actively managing their finances (such as moving money or applying for credit) or passively doing so (such as garnering recommendations or insights).

Application: A financial capability under the ownership of a Data Recipient that services a Consumer Permissioned Data Sharing Use Case.

Data Access Platforms: intermediaries that facilitate financial data access, transit, storage and/or permissioning on behalf of Data Recipients or End Users, also commonly referred to as "Data Aggregators". In some cases, Data Access Platforms do not have a direct relationship with the End User

Intermediaries: Data Access Platforms, Service Providers, or any other entity in the data sharing chain between a Data Provider to a Data Recipient (**Note**: Term Not defined in the Taxonomy document)

OAuth Client: the entity which performs the direct permissioned data sharing interaction between a Data Recipient and Data Provider. The OAuth Client is either a Data Access Platform or a Direct Data Recipient.

Dynamic Client Registration: the OAuth 2.0 technical standard for registering Data Recipient Applications with Data Providers.

1.4 Key Principles

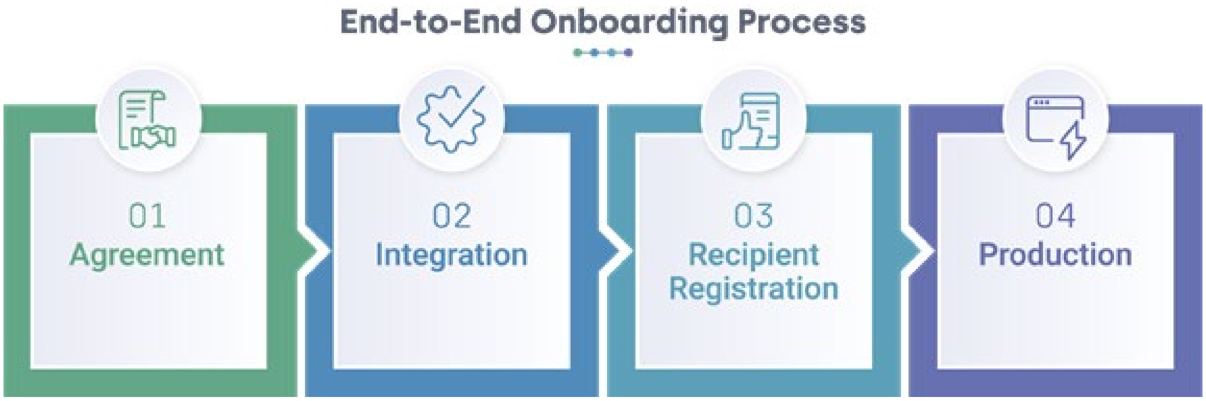
- **Automation -** Providers **MUST** implement a Dynamic Client Registration API to allow for Automation in the Recipient Registration process.
- **Interoperability & Consistency** - Provider Client Registration API end point implementations **MUST** adhere to the OAuth 2.0 Dynamic Client Registration IETF standards in IETF RFC 7591 and IETF RFC 7592. Provider DCR implementations **SHOULD** also follow FDX Defined OAuth Extensions to add intermediaries and status as proposed in FDX RFC 0117 and as revised in the guidelines defined in this document.
- **Flexibility -** For Providers who have already built a Custom Client Registration API, the Provider **SHOULD** plan for a future upgrade to the Dynamic Client Registration standard.
- **Security** (Info Security and Recipients) - Data Recipient registration **MUST** be secured with a method that ensures the Recipient Registration request is from a known Partner (Data Access Platform or Direct Recipient). Data Providers **MUST** secure the Dynamic Client Registration end point with OAuth 2.0 authentication as defined in IETF RFC 6749 (Access Token). Data Providers **MUST** restrict the Client Authentication to mTLS via registered client certificates. FDX Recommends that Data Providers **SHOULD** use the practices in IETF RFC 8705 to bind the Credential (Bearer Token) to the secure channel.

Data Providers **MAY** implement an immediate Approval and transmission, or **MAY** Deploy an offline approval process for Recipient application registrations.

- **Compliance -** Data Access Platforms and Data Recipients **MUST** accurately provide all required elements for Recipient Application registration

1.5 End-to-End Recipient Onboarding process

Recipient Application Registration is a significant activity in the Financial Data sharing journey. It is important to understand where this process fits into the overall Onboarding process.



1.5.1 Agreement

As the first step in the process, Data Access Platform and Direct Recipients organizations establish partnerships with Financial Data Providers where they must agree to work together to transition from legacy credential-based aggregation methods, to a Consumer Permissioned Data Sharing integration. In many cases, these partnerships involve bi-lateral data sharing agreements. These agreements often include provisions for liability, recipient ecosystem terms, security, data governance, consent/token management and API usage. Bi-lateral agreements can vary in scope and complexity by Provider. Some Data Providers only require very simple agreements. In nearly all cases, Data Providers require that some form of agreement is in place, as pre-cursor to production Data Recipient registration and Data Recipient application launches.

1.5.2 Integration

Though common for Integration to start following a Bi-Lateral agreement, arrangements are often made to start integration in parallel. Integration is defined by all pre-production development and testing activities between a Data Provider and an OAuth Client (Data Access Platform or Direct Recipient), to achieve production readiness. This process step involves connecting to a Data Provider pre-production environment, technical integration with OAuth 2.0 compliant Authorization services, FDX API Integration, and Use case data certification. The FDX Certification Model document describes the end to end process in detail.

1.5.3 Application Registration

After an agreement is in place and the Integration is complete, the iterative and continual process of Data Recipient Registration is ready to start. This includes:

- Mass Migration of all existing Data Recipients for Data Access Platforms*
- Ongoing and continual new Recipient Registrations
- Multi-Site Providers (Digital Providers and FIs with more than one Credentialed end point)
    - Mass Migration for each newly rolled-out Site
    - Multi-site registrations for new Recipient Registrations

*Note the Direct Recipients often need to only register one time for Providers that only operate a single site

The document will focus on Dynamic Client Registration as a means for servicing Data Recipient Application Registration.

1.5.4 Production

Once a Recipient Application is registered, end consumers are able to participate in the secure Data Sharing journey between the Provider and Recipient as described in the FDX UX Guidelines.

1.6 Recipient Life Cycle

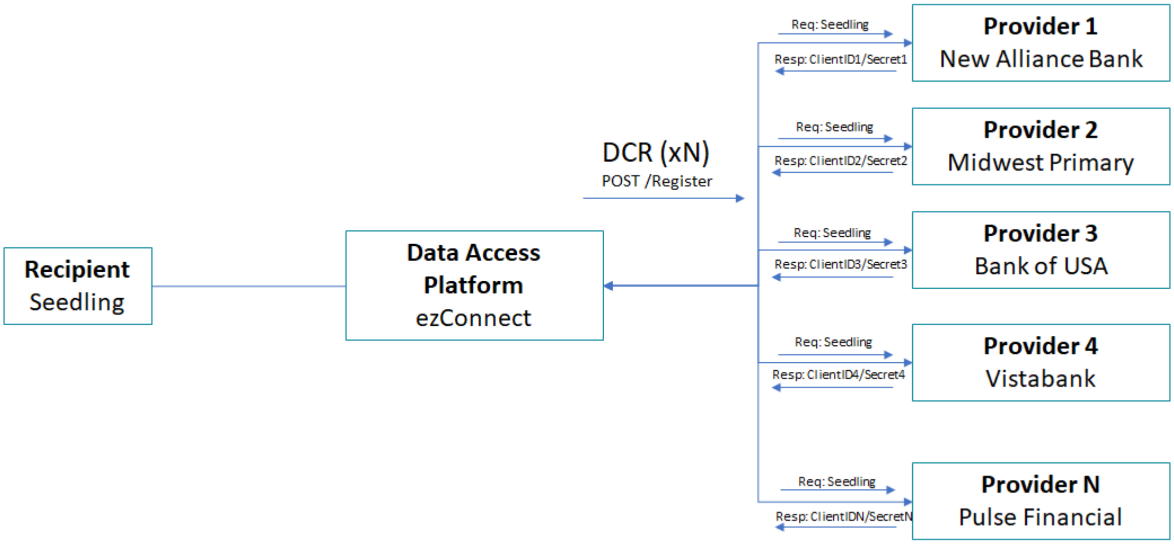
The bi-lateral agreements between Data Providers and Data Access Platforms establish the ground rules for downstream Recipient eligibility to participate in the Data Sharing ecosystem. Guidelines for Recipient eligibility are out of the scope of this document. Once a Data Recipient is integrated with a Data Access Platform and has passed through security and governance vetting procedures, the Data Recipient is ready to start the Recipient Life Cycle.

- Recipient Registration
- Recipient Change Management
- Recipient Revocation

Direct Recipients serve as both the OAuth Client as well as the Recipient Application within the same business organization. Direct Recipients follow the same Recipient Life Cycle as Recipients who utilize a Data Access Platform as an Intermediary. For illustrative purposes, examples in this section will describe the Recipient life cycle from the perspective of a Data Access Platform.

1.6.1 Recipient Registration

Eligible Recipients are individually Registered with each integrated Data Provider.

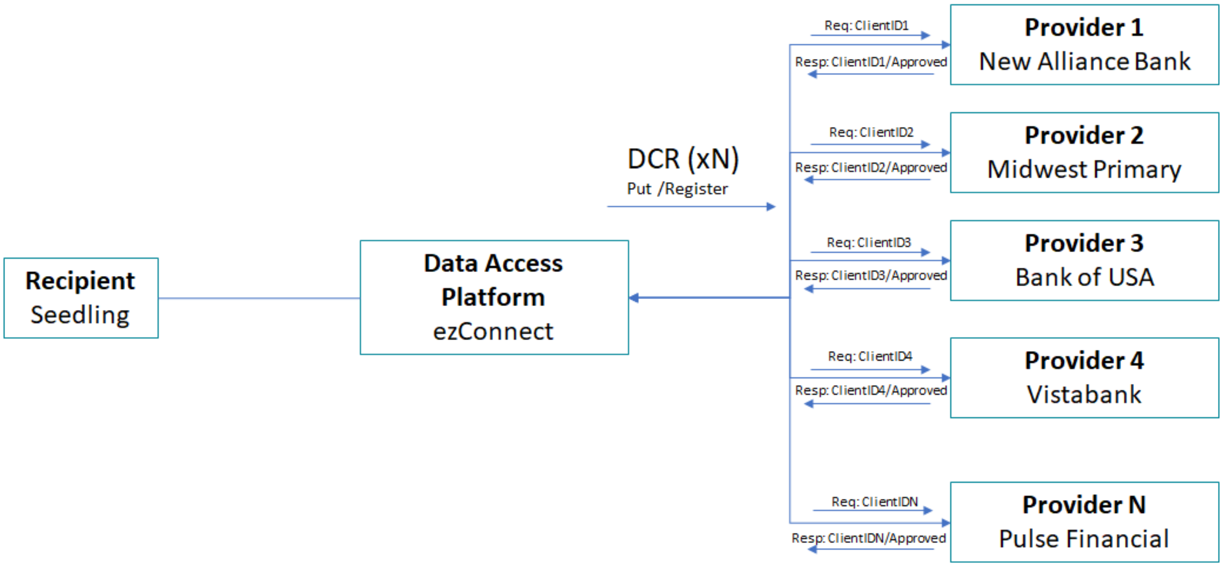


When a Recipient onboards with a Data Access Platform, the Data Access Platform collects all of the required metadata that is needed for Data Provider registrations. This data is then used in the Recipient Registration process. Required metadata includes Recipient Name, description, eligible Redirect URLs, Recipient URL and Recipient Contact Information. Additionally, the Data Provider may require all Intermediary information to be sent in the registration request. This includes the Data Access Platform name, description, URL, contacts and sub-intermediaries that are part of the data sharing chain. Additional fields may include the Recipient logo, Data Cluster Scope and Consent Duration. Most Recipient metadata fields are documented in the FDX OAuth

Extensions section (originally FDX RFC 0117). FDX RFC 153, this Recipient Registration Guidelines Document, extends the Recipient Metadata.

1.6.2 Recipient Change Management

Once a Data Recipient is registered with a Data Provider, end users are able to participate in the full Consent Life cycle for permissioned Data sharing. This life cycle includes Grant Consent, Manage Consent and Revoke Consent. Registered Data Recipients can evolve over time as functionality is enhanced, use cases are added, and business relationships are expanded. These changes often result in metadata changes, which trigger updates to the original registration.

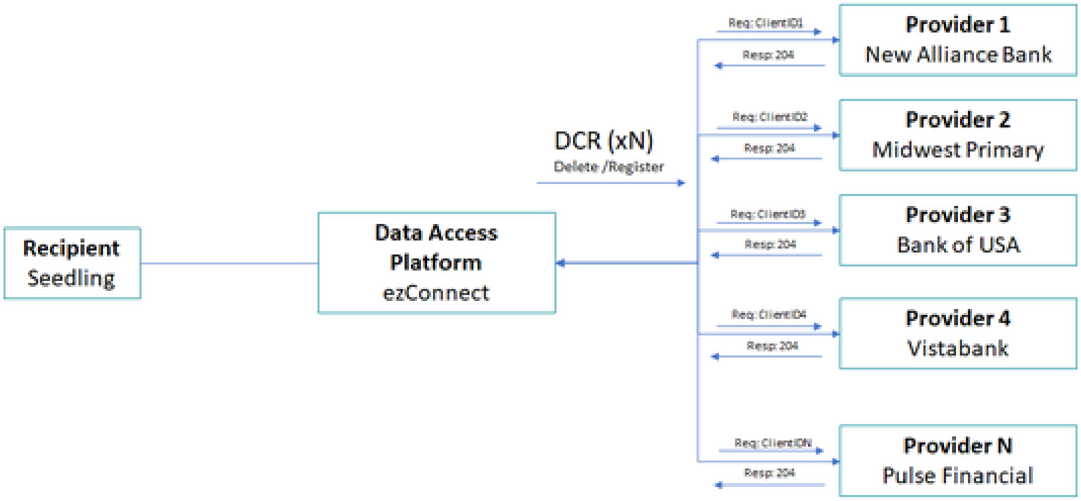


Common Recipient changes include:

- Recipient Application Name change due to rebranding or acquisition
- Recipient Logo change due to rebranding or acquisition
- Data Cluster Scope changes due to new Use Cases added by the Data Recipient

1.6.3 Recipient Revocation

In the event that a Data Recipient discontinues offering Permissioned Data sharing services, the Data Access Platform must notify all registered Data Providers.



Dynamic Client Registration gives the Data Access Platform the ability to remove a Data Recipient.

2.0 Dynamic Client Registration Standard

There are 3 IETF RFCs and 2 FDX RFCs that describe the holistic call standards and metadata set which describes the FDX permissioned Data Sharing experience.

Reference	Origin	Content
OAuth 2.0 Dynamic Client Registration Protocol	IETF RFC 7591	Original DCR RFC with create (POST) function, metadata definitions and Security
OAuth 2.0 Dynamic Client Registration Management	IETF RFC 7592	Follow on IETF RFC with Read (GET), Update (PUT) and remove (DELETE) functions
OAuth 2.0 Client Intermediary Metadata	OAuth Intermediary Definition	Intermediary definition
FDX OAuth Extensions	FDX RFC 0117 - FDX OAuth Extensions FDX API Security Model v3.5 Document	FDX Metadata extension with Intermediary identity and status definitions, and guidance for usage of DCR scope attribute for Data Clusters
Recipient Registration Automation	FDX RFC 0153 - Recipient Registration Automation (as encapsulated in this document)	FDX Dynamic Client Registration Data Provider implementation and Data Recipient usage guidelines

2.1 Recipient Registration Metadata

The OAuth Client organization will serve as the Registering Party for the Data Recipient. Direct Data Recipients will register themselves as a Data Recipient. Data Access Platforms will register the Data Recipients who rely on their Services.

- Registering party **MUST** collect the Recipient and intermediary metadata
- Registering party **MUST** ensure accuracy for all recipient and intermediary fields. If there is a question about how a recipient is displayed (Name, Logo), the Registering party **MUST** work with the Data Recipient to obtain the correct information.
- Recipient and Intermediaries **SHOULD** be easy to identify based on clear names, logos and URIs that can be looked up on the the internet.

2.1.1 Request Metadata

The following fields are available in the DCR POST "/Register" API Call. Any fields not displayed below are described in IETF RFC7591 and IETF RFC 7592.

Name	Required/ Optional	Description	Available in UX?	Definition reference
Recipient Level Data				
client_name	Required	The Data Recipient or Data Recipient Application name displayed by Data Provider during the consent Flow as well as in the Consent Dashboard. The client_name **MUST** be a user recognizable name.	Yes	IETF RFC7591
description	Optional	A short description of the Data Recipient application, that **MAY** be presented by the Data Provider to the End-User during the consent flow or in the consent dashboard. The description **MAY** also be included in Data Recipient/Data Access Platform owned Initiate and Disclose steps in the Consent journey. Data Providers **SHOULD** require the description field. Data Recipient registering parties **SHOULD** include the recipient description during registration. Inclusion of description will facilitate transparency to help end consumers recognize Recipient Applications for past consumer permissioned data sharing events.	Yes	IETF RFC7591
redirect_uris[]	Required	An array of eligible Redirect URI targets that **MUST** be pre-registered for the consumer redirect transmission following Data Provider Authorization and Consent completion. OAuth 2.0 defines multiple Redirect URIs in an array for operational purposes including load balancing, redundancy, multi-data center support and multiple environment support. The redirect URI **SHOULD** be owned by the registering party.	No	IETF RFC7591
logo_uri	Optional	Data Recipient Logo URL location. The Recipient Logo **MAY** be included along with the client_name in the Consent Flow and Consent Dashboard.	Yes	IETF RFC7591
client_uri	Optional	The URI which provides additional information about the Data Recipient, and possibly the specific application that is being registered.	Yes	IETF RFC7591

Name	Required/ Optional	Description	Available in UX?	Definition reference
contacts[]	Optional	Array of strings representing ways to contact individuals responsible for the Data Recipient application. The contacts array **SHOULD** accurately represent the email aliases of the support groups for the Data Provider in the event of an issue. "contacts":["customer_support@company.com", "abuse@company.com", "site_reliability@company.com"], Contacts **SHOULD** be included if offered as a Data Provider option. For Data Access Platform managed Data Recipients, the support contacts **MAY** include the Data Access Platform email aliases and/or the Data Recipient aliases.	No	IETF RFC7591

Name	Required/ Optional	Description	Available in UX?	Definition reference
scope	Optional	String field with a list of scopes (see FDX User Experience Guidelines) for Data Recipient access that are defined in FdxOauthScope, and OauthScope enumerations. The string should list the scopes with a space separating each entry. Prior use of Data Cluster enumerations as scopes are deprecated as of FDX API v6.3. "scope": "fdx:accountdetailed:read fdx:transactions:read" Scopes include: • fdx:accountbasic:read • fdx:accountdetailed:read • fdx:bills:read • fdx:customercontact:read • fdx:customerpersonal:read • fdx:images:read • fdx:investments:read • fdx:notifications:publish • fdx:notifications:subscribe • fdx:paymentsupport:read • fdx:rewards:read • fdx:statements:read • fdx:tax:read • fdx:transactions:read Deprecated Data Clusters include: • CUSTOMER_CONTACT • CUSTOMER_PERSONAL • ACCOUNT_BASIC • ACCOUNT_DETAILED • ACCOUNT_PAYMENTS • INVESTMENTS • TRANSACTIONS • STATEMENTS	Yes	IETF RFC7591
duration_type[]	Optional	The duration of consent for the Data Recipient consumers. Options include: • persistent • time_bound • one_time This is an array for Providers that offer duration optionality during consumer authorization and consent.	Yes	FDX RFC 0153 (this document)

Name	Required/ Optional	Description	Available in UX?	Definition reference
duration_period	Optional	Duration period is required when duration_type=time_bound. The duration_period is the maximum consent duration that would be requested for a Recipient consumer. The Data Provider may reject the Recipient DCR request if duration_period requested exceeds the maximum allowed duration offered by the Provider.	Yes	FDX RFC 0153 (this document)
lookback_period	Optional	The maximum number of days allowed for Data Recipient consumers to obtain in transaction history, effective from the current date.	Yes	FDX RFC 0153 (this document)
registry_references[]	Optional	An array of external registries containing registered_entity_name, registered_entity_id and registry fields for the registries where the data recipient is registered. All three fields must be populated if included in the registration.	No	FDX RFC 0206 (as in the Recipient Registration with Delegation document)
registered_entity_name	Optional	The legal company name for the recipient. This may be the same as the client_name used in the consent display. In many cases this could be a parent company name or the recipient name with Corporation or LLC.	No	FDX RFC 0153 (this document)
registered_entity_id	Optional	An ID representing the company that can be looked up from a legal identity registry source.	No	FDX RFC 0153 (this document)
registry	Optional	The Registry source. Values include PRIVATE, FDX, GLEIF, ICANN.	No	FDX RFC 0153 (this document)
Intermediary Level Data				
intermediaries[]	Optional	An array of all intermediaries in the chain. The first intermediary listed **MUST** be the Data Access Platform if applicable. The remaining intermediaries should include all service providers in the chain between the Data Access Platform and the Recipient. The last intermediary listed **MUST** be the intermediary directly linked to the Data Recipient.	Yes	FDX RFC 0117

Name	Required/ Optional	Description	Available in UX?	Definition reference
name	Optional	A Provider recognizable name for the intermediary. The Data Access Provider Intermediary name **MAY** may be presented during the Initiate, Disclose and Select Data Provider portions of the Consumer Permissioned Access User Experience.	Yes	FDX RFC 0117
description	Optional	A short description of the intermediary that **MAY** be also presented to the End-User along with the intermediary name in the consent flow.	Yes	FDX RFC 0117
uri	Optional	The URI provides additional information about this Intermediary in the chain. The Data Access Platform URI **MAY** be included along with the name and description during the consent flow.	Yes	FDX RFC 0117
logo_uri	Optional	Intermediary logo URI location. Similar to the other Intermediary data elements, the Data Access Platform Logo **MAY** be included in the consent journey.	Yes	FDX RFC 0117
contacts[]	Optional	Array of strings representing ways to contact individuals responsible for the Intermediary services. The contacts array **SHOULD** accurately represent the email aliases for the intermediary support groups. "contacts":["customer_support@company.com", "abuse@company.com", "site_reliability@company.com"],	No	FDX RFC 0117
registry_references[]	Optional	An array of external intermediary registries containing registered_entity_name, registered_entity_id and registry fields. All three fields must be populated if included in the registration.	No	FDX RFC 0206
registered_entity_name	Optional	The legal company name for the intermediary.	No	FDX RFC 0153 (this document)
registered_entity_id	Optional	An ID representing the intermediary that can be looked up from a legal identity registry source	No	FDX RFC 0153 (this document)
registry	Optional	The Registry source. Values include PRIVATE, FDX, GLEIF, ICANN.	No	FDX RFC 0153 (this document)

2.1.2 DCR Request Example

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "My Example Client",
  "description": "Recipient application for specified financial use case",
  "redirect_uris": ["https://partner.example/callback"],
  "logo_uri": "https://client.example.org/logo.png",
  "client_uri": "https://example.net/",
  "contacts": ["support@example.net"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS INVESTMENTS",
  "duration_type": ["time_bound"],
  "duration_period": 365,
  "lookback_period": 365,
  "registry_references": [
  {
    "registered_entity_name": "Data recipient company legal name",
    "registered_entity_id": "4HCHXIURY78NNH6JH",
    "registry": "GLIEF"
  }],
  "intermediaries": [
  {
    "name": "Data Access Platform Name",
    "description": "Company servicing permissioned financial data sharing",
    "uri": "https://partner.example/",
    "logo_uri": "https://partner.example/logo.png",
    "contacts": ["support@partner.com"],
    "registry_references": [
    {
      "registered_entity_name": "Data Access Platform company legal Name",
      "registered_entity_id": "JJH7776512TGMEJSG",
      "registry": "FDX"
    }]
  },
  {
    "name":“Digital Service Provider Name",
    "description": "Digital Service Provider to the Recipient",
    "uri":"https://sub-partner-one.example/",
    "logo_uri":"https://sub-partner-one.example/logo.png",
    "contacts": ["support@sub-partner-one.com"],
    "registry_references": [
    {
      "registered_entity_name": "Service Provider legal company Name",
      "registered_entity_id": "9LUQNDG778LI9D1",
      "registry": "GLIEF"
    }]
    }]
}
```

2.1.3 Response Metadata

Name	Required / Optional	Description	Definition Reference
client_id	Required	OAuth 2.0 client identifier. Unique ID representing Data Recipient and Identity Chain combination	IETF RFC7591
client_secret	Optional / Recommended	OAuth 2.0 client secret string. **MUST** be unique for each "client_id". Used by the OAuth Client (Data Access Platform or Direct Recipient) to authenticate to the Authorization Server token endpoint	IETF RFC7591
grant_types	Required	Array of OAuth 2.0 grants made available to the Data Recipient. “authorization_code” and “refresh_token” **SHOULD** be the expected Grant types. Grant types of “implicit”, “password” and “client_credentials” **MUST** not be provided as options.	IETF RFC7591
token_endpoint_auth_method	Required	Requested Authentication method for Authorization Server.	IETF RFC7591

Name	Required / Optional	Description	Definition Reference
registration_client_uri	Optional	Fully qualified URI for subsequent DCR calls (GET, PUT, DELETE) for managing the Data Recipient registration (for this Client ID). This endpoint URL **SHOULD** be formed through the use of a server-constructed URL string that combines the client registration endpoint's URL and the issued "client_id" for this client, with the latter as either a path parameter or a query parameter.	IETF RFC 7592
registration_access_token	Optional	String containing a unique DCR access token to be used in subsequent operations to manage the Data Recipient registration (for this Client ID). Previously recommended, but made Optional for FAPI compliant Providers.	IETF RFC 7592

Name	Required / Optional	Description	Definition Reference
status	Optional / Recommended	Status defined in FDX RFC 0117 - FDX OAuth Extensions . Valid values include: • Approved - Fully Active • Tentative - Active but pending an initial or change approval • Pending - Inactive and pending an initial approval • Rejected - Registration rejected • Inactive - Registered but inactive Client ID Providers who have an offline approval process **MUST** include status in the DCR response	FDX RFC 0117
Recipient and Intermediary Metadata fields	Required	All metadata fields sent in the DCR request **MUST** be included in the response to confirm receipt	See section 2.1.1 Request Metadata

2.1.4 DCR Response Example

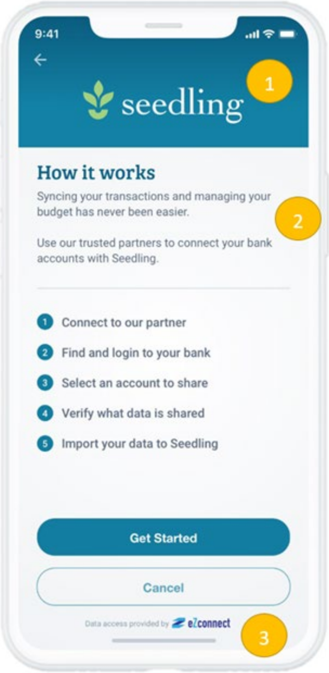
```
HTTP/1.1 201 Created
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
  "client_id": "V8tvEkZWh",
  "client_secret": "SpsuwZIxnp8bBEhp5sk1EKiIKTZ4X4DKU",
  "grant_types": ["authorization_code", "refresh_token"],
  "token_endpoint_auth_method": "private_key_jwt",
  "registration_client_uri": "https://server.example.com/register/V8tvEkZWh",
  "status": "Approved",
  "client_name": "My Example Client",
  "description": " Recipient application for specified financial use case",
  "redirect_uris": ["https://client.example.org/callback"],
  "logo_uri": "https://client.example.org/logo.png",
  "client_uri": "https://example.net/",
  "contacts": ["support@example.net"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS INVESTMENTS“,
  "duration_type": ["time_bound"],
  "duration_period": 365,
  "lookback_period": 365,
  "registry_references": [
  {
    "registered_entity_name": "Data recipient company legal name",
    "registered_entity_id": "4HCHXIURY78NNH6JH",
    "registry": "GLIEF"
  }],
  "intermediaries": [
  {
    "name": "Data Access Platform Name",
    "description": "Company servicing permissioned financial data sharing",
    "uri": "https://partner.example/",
    "logo_uri": "https://partner.example/logo.png",
    "contacts": ["support@partner.com"],
    "registry_references": [
    {
      "registered_entity_name": "Data Access Platform company legal Name",
      "registered_entity_id": "JJH7776512TGMEJSG",
      "registry": "FDX"
    }]
  },
  {
    "name":“Digital Service Provider Name",
    "description": "Digital Service Provider to the Recipient",
    "uri":"https://sub-partner-one.example/",
    "logo_uri":"https://sub-partner-one.example/logo.png",
    "contacts": ["support@sub-partner-one.com"],
    "registry_references": [
    {
      "registered_entity_name": "Service Provider legal company Name",
      "registered_entity_id": "9LUQNDG778LI9D1",
      "registry": "GLIEF"
    }]
  }]
}
```

2.2 Registration and UX Guidelines Alignment

This section provides visuals to illustrate how Recipient Registration metadata maps to the secure Data Sharing Customer experience.

2.2.1 Initiate

#	DCR Metadata	Comment
1	client_name	Recipient name
1	logo_uri	Recipient logo
2	description	Recipient short description
3	intermediaries.logo_uri	Data Access Platform Intermediary logo



How it works

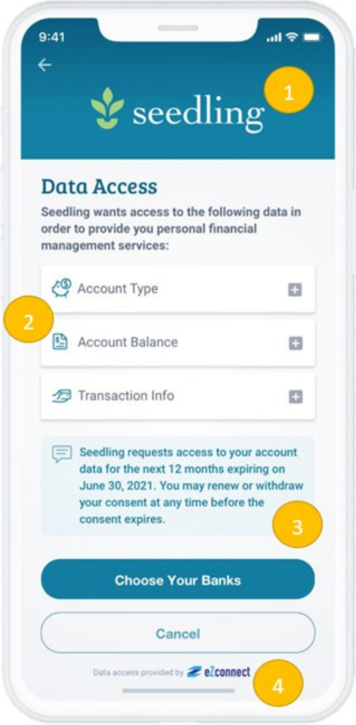
Syncing your transactions and managing your budget has never been easier.

Use our trusted partners to connect your bank accounts with Seedling.

1 Connect to our partner
2 Find and login to your bank
3 Select an account to share
4 Verify what data is shared
5 Import your data to Seedling

2.2.2 Disclose

#	DCR Metadata	Comment
1	client_name	Recipient name
1	logo_uri	Recipient logo
2	scope[]	Data Clusters in scope for access grant
3	duration_type	Duration of requested consent. Time based consent in this example
3	duration_period	The duration period for a time based duration type
4	intermediaries.logo_uri	Data Access Platform Intermediary logo



Data Access

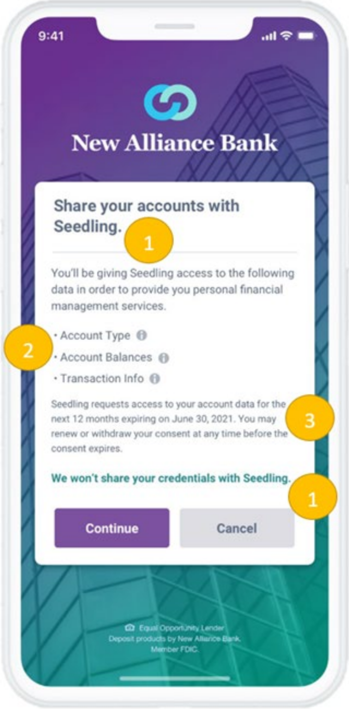
Seedling wants access to the following data in order to provide you personal financial management services:

Account Type
Account Balance
Transaction Info

Seedling requests access to your account data for the next 12 months expiring on June 30, 2021. You may renew or withdraw your consent at any time before the consent expires.

2.2.3 Consent

#	DCR Metadata	Comment
1	client_name	Recipient name
2	scope[]	Data Clusters in scope for access grant
3	duration_type	Duration of requested consent. Time based consent in this example
3	duration_period	The duration period for a time based duration type



Share your accounts with Seedling.

You'll be giving Seedling access to the following data in order to provide you personal financial management services.

- Account Type
- Account Balances
- Transaction Info

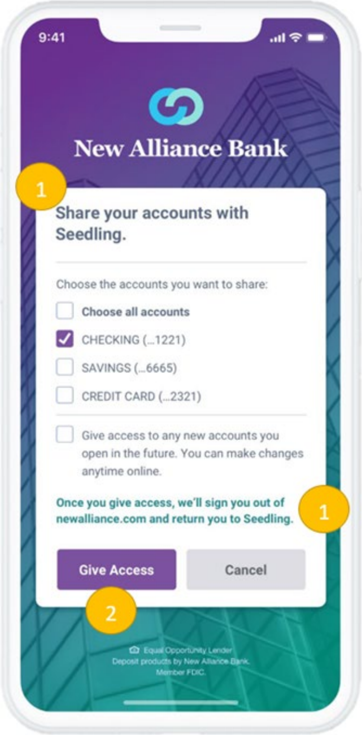
Seedling requests access to your account data for the next 12 months expiring on June 30, 2021. You may renew or withdraw your consent at any time before the consent expires.

We won't share your credentials with Seedling.

Equal Opportunity Lender
Deposit products by New Alliance Bank.
Member FDIC.

2.2.4 Authorize

#	DCR Metadata	Comment
1	client_name	Recipient name
2	redirect_uri[]	The redirect URI to which the Authorization code is sent at the completion of the consent process



Share your accounts with Seedling.

Choose the accounts you want to share:

□ Choose all accounts
☑ CHECKING (...1221)
□ SAVINGS (...6665)
□ CREDIT CARD (...2321)

□ Give access to any new accounts you open in the future. You can make changes anytime online.

Once you give access, we'll sign you out of newalliance.com and return you to Seedling.

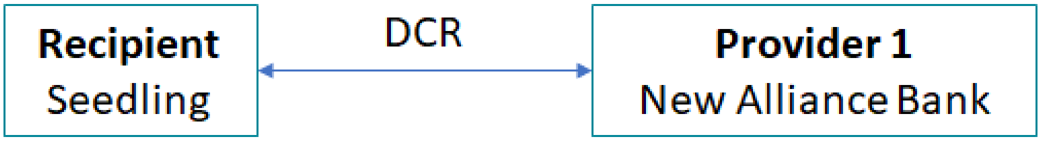
Equal Opportunity Lender
Deposit products by New Alliance Bank.
Member FDIC.

2.3 Financial Data Flows

Dynamic Client Registration supports a multitude of Data Recipient Registration possibilities.

2.3.1 Direct Data Recipient

Direct Recipient application (Seedling) is also the OAuth Client. Direct recipient secures Client ID/Secret and directly services Application End Consumers redirect and token Authorization with Provider (New Alliance Bank).

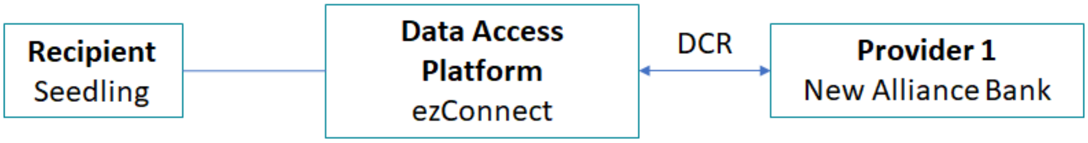


Direct Recipient DCR Request Example

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://seedling.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seedling.com/",
  "contacts": ["support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS",
  "duration_type": "persistent"
}
```

2.3.2 Data Access Platform

Data Access Platform (ezConnect) registers Data Recipient (Seedling) to Data Provider (New Alliance Bank) via DCR. Data Access Platform secures Client ID/Secret and directly services Recipient End Consumers redirect and token Authorization with Provider.

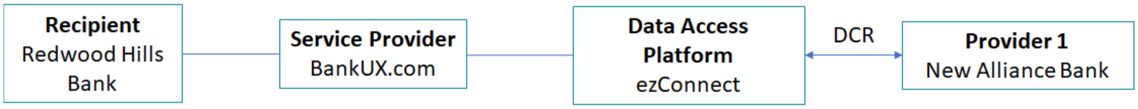


Data Access Platform DCR Request Sample

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@ezconnect.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS",
  "duration_type": "persistent",
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezconnect.com/",
    "logo_uri": "https://ezconnect.com/logo.png",
    "contacts": ["support@ezconnect.com"]
  }]
}
```

2.3.3 Data Access Platform with Recipient Chain

ClientID/Secret is issued through DCR from Provider (New Alliance Bank) to DAP (ezConnect) for end Recipient (Redwood Hills Bank) whose application is operated by an Online Banking Service Provider (BankUX.com). Subsequently, ezConnect secures Client ID/Secret and services redirect and token authorization for Redwood Hills Bank through a service provider integration with BankUX.com.



Data Access Platform DCR Request Sample

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Redwood Hills Bank",
  "description": "Personalized Banking integrating all your accounts",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://olb.redwoodhillsbank.com/logo.png",
  "client_uri": "https://olb.redwoodhillsbank.com.com/",
  "contacts": ["support@ezconnect.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS",
  "duration_type": "persistent",
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezconnect.com/",
    "logo_uri": "https://ezconnect.com/logo.png",
    "contacts": ["support@ezconnect.com"]
  },
  {
    "name":"BankUX.com",
    "description": "Full Service Online and Mobile Banking made easy",
    "uri":"https://bankux.com/",
    "logo_uri":"https://bankux.com/logo.png",
    "contacts": ["support@bankux.com"]
  }]
}
```

3.0 Dynamic Client Registration Automation Use Cases

Dynamic Client Registration gives Data Access Platforms the ability to Automate Recipient onboarding with a Data Provider. A few of the ways Data Access Platforms can wire in DCR Automation include:

- Integration with Administration tools, enabling a Customer Support Engineer to trigger DCR for a Recipient to one or many Data Providers
- Automatic DCR trigger to all available Data Providers when a new Data Recipient is onboarded to a Data Access Platform

Recipient metadata data **SHOULD** be common for all Provider Registrations as described in section 2.1.1.

Data Provider DCR metadata requirements **MAY** vary, and the Data Access Platform **MUST** manage Metadata pertaining to each implementation. Provider implementation variables will include:

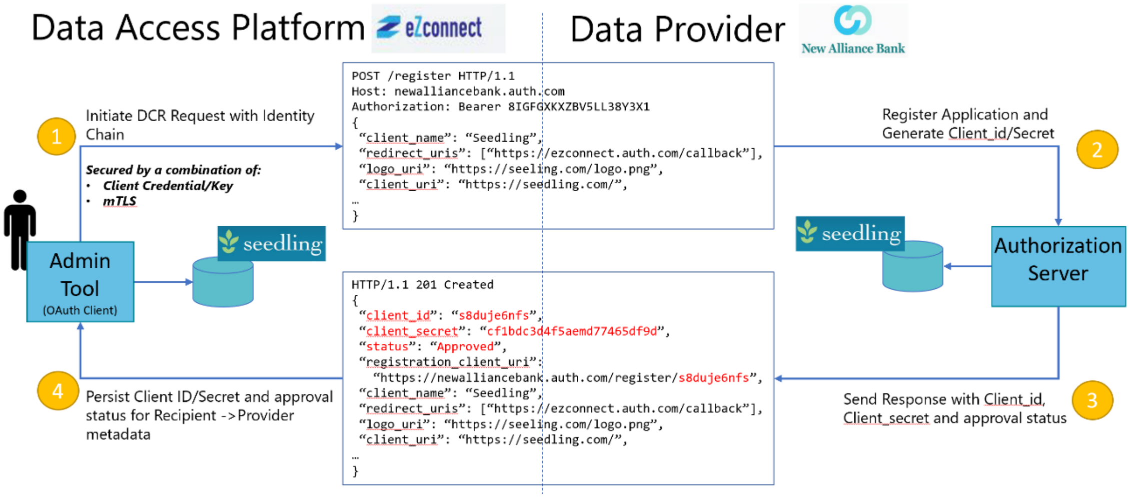
1. DCR Registration URL
2. Required Data Recipient attributes
3. Required Intermediary attributes
4. Scope Options
5. Approval handling

The following sections describe each of the Dynamic Client Registration Automation Use cases. Automation trigger points are at the discretion of the OAuth Client organization. For simplification purposes, the examples are using a Data Access Platform as the OAuth client while representing a single DCR Transaction. The processes shown also apply for Direct Recipients.

3.1 Recipient Registration with Immediate Approval

Dynamic Client Registration with an immediate approval is the easiest use case to automate. This is a single transaction, resulting in a Client ID/Client secret that can be immediately utilized by Recipient application consumers to permission data sharing.

DCR Registration (Post) - Immediate Approval


Data Access Platform
eZconnect
Data Provider
New Alliance Bank
1
Initiate DCR Request with Identity Chain
Secured by a combination of:
- *Client Credential/Key*
- *mTLS*
Admin Tool (OAuth Client)
seedling
2
Register Application and Generate Client_id/Secret
seedling
Authorization Server
3
Send Response with Client_id, Client_secret and approval status
4
Persist Client ID/Secret and approval status for Recipient ->Provider metadata

```
POST /register HTTP/1.1
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

```
HTTP/1.1 201 Created
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Approved”,
 “registration_client_uri”:
   “https://newalliancebank.auth.com/register/s8duje6nfs”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

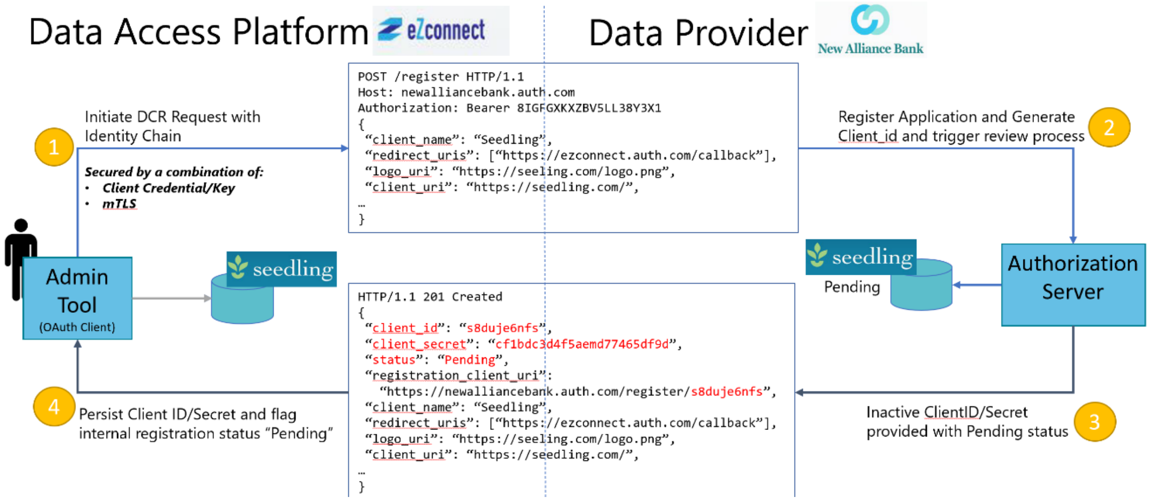
1. Data Access Platform submits DCR request to the Data Provider for a new Data Recipient
2. Data Provider receives the DCR request, performs validation checks (uniqueness of client_name/intermediaries combo), creates new registry entry for new Data Recipient, and generates unique Client ID and Client Secret
3. Data Provider sends response to Data Access Platform with new Client ID and Secret with an approval status. Status is an optional field for Data Providers with an immediate approval process
4. Data Access Platform persists the Client ID and Secret for Data Recipient to Provider relationship in its metadata store. Data Access Platform also stores registration client URI if required for subsequent DCR management calls. **Client ID/Secret are ready for use for Permissioned Data Sharing!**

3.2 Recipient Registration with Offline Approval

For Providers who require an offline approval process, DCR can still be automated. For Automation to be possible, Providers who implement DCR **MUST**:

1. Provide a Status in the response
2. Implement READ (GET)

Step 1 - DCR Registration Request (POST) - Offline Approval



1 Initiate DCR Request with Identity Chain

Secured by a combination of:
- ***Client Credential/Key***
- ***mTLS***

```
POST /register HTTP/1.1
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

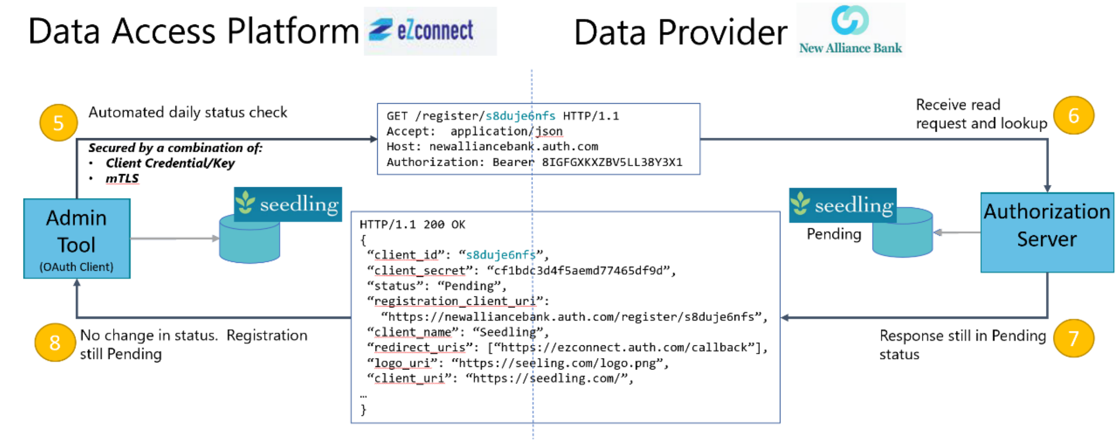
2 Register Application and Generate Client_id and trigger review process

3 Inactive ClientID/Secret provided with Pending status

```
HTTP/1.1 201 Created
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Pending”,
 “registration_client_uri”:
   “https://newalliancebank.auth.com/register/s8duje6nfs”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

4 Persist Client ID/Secret and flag internal registration status “Pending”

Step 2 - DCR Status Check (GET) - No Change



5 Automated daily status check

Secured by a combination of:
- ***Client Credential/Key***
- ***mTLS***

```
GET /register/s8duje6nfs HTTP/1.1
Accept:  application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
```

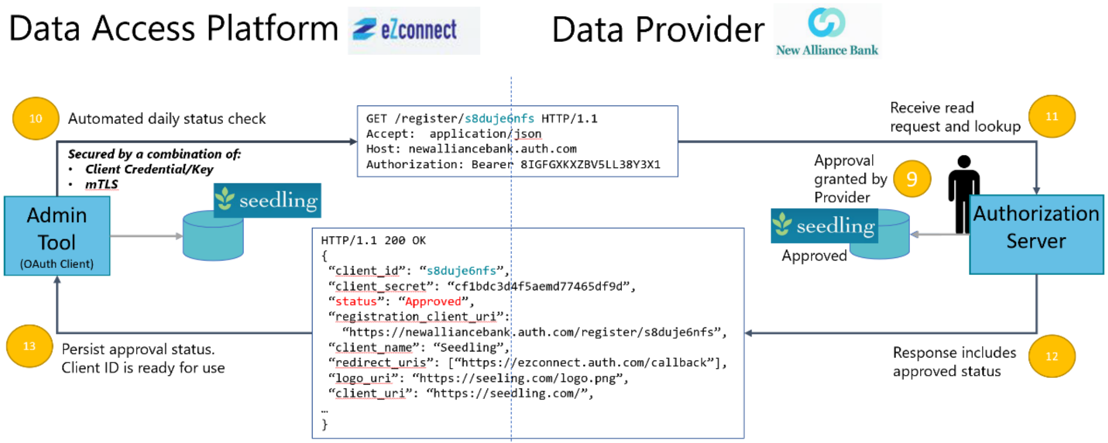
6 Receive read request and lookup

7 Response still in Pending status

```
HTTP/1.1 200 OK
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Pending”,
 “registration_client_uri”:
   “https://newalliancebank.auth.com/register/s8duje6nfs”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

8 No change in status. Registration still Pending

Step 3 - DCR Status Check (GET) - Approval and Completion



```
GET /register/s8duje6nfs HTTP/1.1
Accept:  application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
```

```
HTTP/1.1 200 OK
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Approved”,
 “registration_client_uri”:
   “https://newalliancebank.auth.com/register/s8duje6nfs”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “client_uri”: “https://seedling.com/”,
…
}
```

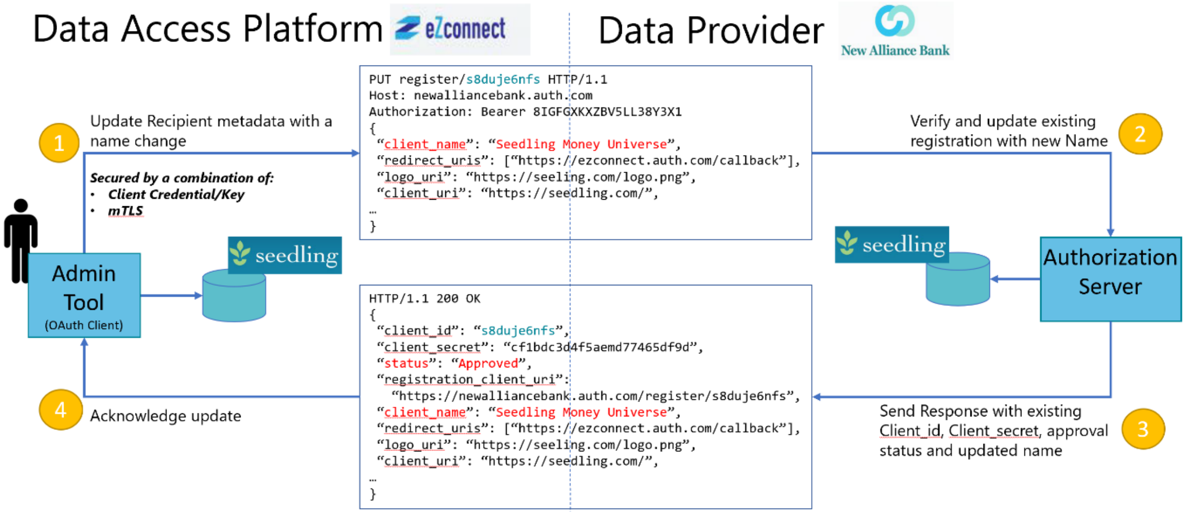
1. Data Access Platform submits DCR request to the Data Provider for a new Data Recipient
2. Data Provider receives the DCR request, performs validation checks (uniqueness of client_name/intermediaries combo), creates new local registry entry for new Data Recipient, and generates unique Client ID and Secret. Request is marked for Internal Provider review.
3. Data Provider sends response to Data Access Platform with new Client ID and Secret with Pending status.
4. Data Access Platform persists the Client ID and Secret for Recipient to Provider relationship. Data Access Platform also stores registration client URI (if required) for subsequent DCR management calls. **Pending status is persisted and Recipient Access to Data Provider is inactive**
5. Data Access Platform submits DCR GET request to check status. Data Access Platform DCR status check polling processes **SHOULD** be rated limited hourly increments or greater
6. Provider receives request and queries Recipient details
7. Provider responds with latest status which is still pending
8. **Data Access Platform receives response with Pending status and Recipient Access to Data Provider is inactive**
9. Provider Approves Recipient
10. Data Access Platform submits DCR GET request to check status
11. Provider receives request and queries Recipient details
12. Provider responds with latest status, which is updated to Approved

13. **Data Access Platform persists Approved status. Client ID/Secret are ready for use for Permissioned Data Sharing!**

3.3 Update Registration with Immediate Approval

Data Providers **SHOULD** allow simple Recipient and Intermediary metadata changes to be made immediately with no approvals. Here the simple registration update process.

DCR Update (PUT) - Immediate Approval


Data Access Platform
eZconnect
Data Provider
New Alliance Bank
1
Update Recipient metadata with a name change
Secured by a combination of:
• Client Credential/Key
• mTLS
Admin Tool
(OAuth Client)
seedling
2
Verify and update existing registration with new Name
seedling
Authorization Server
3
Send Response with existing Client_id, Client_secret, approval status and updated name
4
Acknowledge update

```
PUT register/s8duje6nfs HTTP/1.1
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
 "client_name": "Seedling Money Universe",
 "redirect_uris": ["https://ezconnect.auth.com/callback"],
 "logo_uri": "https://seeling.com/logo.png",
 "client_uri": "https://seedling.com/",
…
}
```

```
HTTP/1.1 200 OK
{
 "client_id": "s8duje6nfs",
 "client_secret": "cf1bdc3d4f5aemd77465df9d",
 "status": "Approved",
 "registration_client_uri":
   "https://newalliancebank.auth.com/register/s8duje6nfs",
 "client_name": "Seedling Money Universe",
 "redirect_uris": ["https://ezconnect.auth.com/callback"],
 "logo_uri": "https://seeling.com/logo.png",
 "client_uri": "https://seedling.com/",
…
}
```

1. Data Access Platform submits DCR request to the Data Provider to make a simple change for an existing active Data Recipient. A simple change could include client_name, logo_uri, client_uri, and contact at the Data Recipient or Intermediary level.

2. Data Provider receives the DCR request, looks up existing client ID, automatically compares the metadata fields, and updates the local registry with change.

3. Data Provider sends response to Data Access Platform existing client_id, client_secret and an approval status. Status is an optional field for Data Providers with an immediate approval process for all changes. The client metadata is included in the response, including the changed fields.

4. Data Access Platform validates the response from the Data access platform, confirms that the updated fields have been approved, and finalizes the update to the local Recipient registry. **The updated Metadata is live and visible in production where applicable!**

3.4 Update Registration with Offline Approval

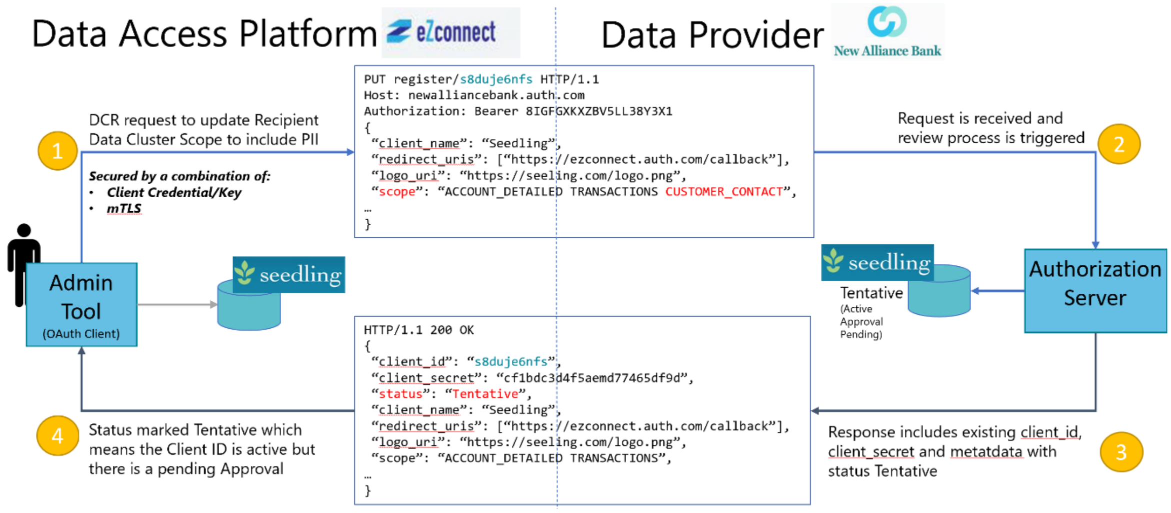
Providers who require an offline approval process for any types of updates, **MUST** include the following capabilities in their DCR implementation:

1. Provide a Status in the response

2. Implement READ (GET)

Here is the standard process for DCR Update Automation with an offline approval process:

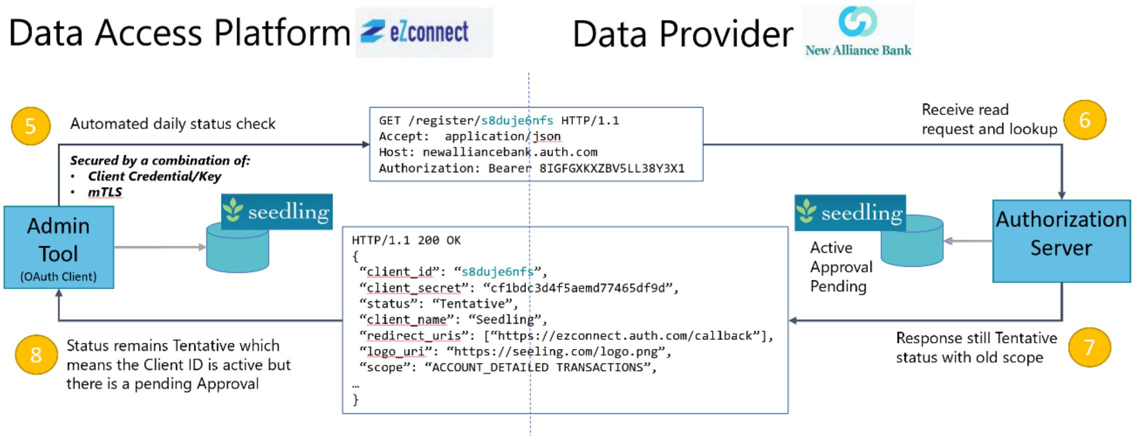
Step 1 - DCR Update Request (PUT) - Offline Approval


Data Access Platform
eZconnect
Data Provider
New Alliance Bank
1
DCR request to update Recipient Data Cluster Scope to include PII
Secured by a combination of:
• Client Credential/Key
• mTLS
Admin Tool
(OAuth Client)
seedling
2
Request is received and review process is triggered
Authorization Server
seedling
Tentative
(Active Approval Pending)
3
Response includes existing client_id, client_secret and metatdata with status Tentative
4
Status marked Tentative which means the Client ID is active but there is a pending Approval

```
PUT register/s8duje6nfs HTTP/1.1
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “scope”: “ACCOUNT_DETAILED TRANSACTIONS CUSTOMER_CONTACT”,
…
}
```

```
HTTP/1.1 200 OK
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Tentative”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “scope”: “ACCOUNT_DETAILED TRANSACTIONS”,
…
}
```

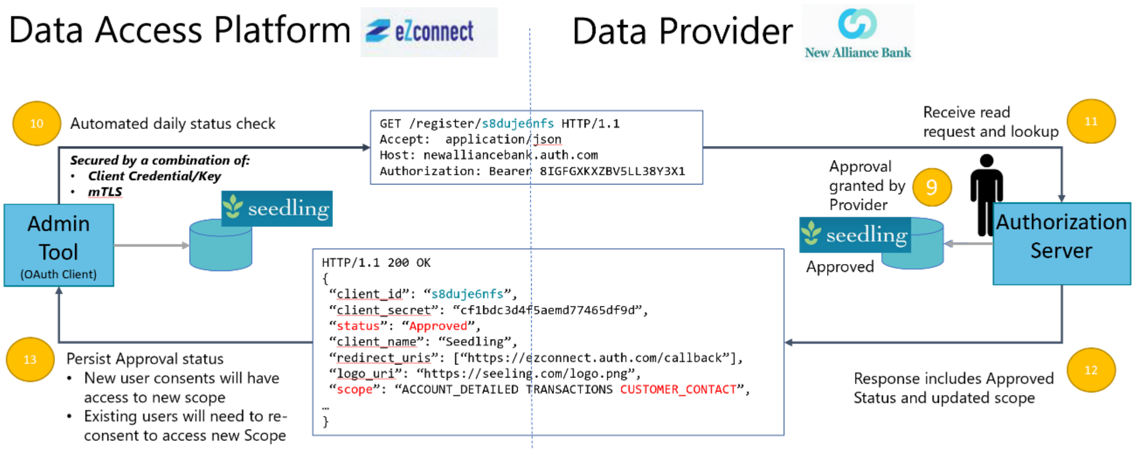
Step 2 - DCR Status Check (GET) - No Change


Data Access Platform
eZconnect
Data Provider
New Alliance Bank
5
Automated daily status check
Secured by a combination of:
• Client Credential/Key
• mTLS
Admin Tool
(OAuth Client)
seedling
6
Receive read request and lookup
Authorization Server
seedling
Active Approval Pending
7
Response still Tentative status with old scope
8
Status remains Tentative which means the Client ID is active but there is a pending Approval

```
GET /register/s8duje6nfs HTTP/1.1
Accept:  application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
```

```
HTTP/1.1 200 OK
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Tentative”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “scope”: “ACCOUNT_DETAILED TRANSACTIONS”,
…
}
```

Step 3 - DCR Status Check (GET) - Approval and Completion


Data Access Platform
eZconnect
Data Provider
New Alliance Bank
10
Automated daily status check
Secured by a combination of:
• Client Credential/Key
• mTLS
seedling
Admin Tool
(OAuth Client)
13
Persist Approval status
• New user consents will have access to new scope
• Existing users will need to re-consent to access new Scope
Receive read request and lookup
11
Approval granted by Provider
9
seedling
Approved
Authorization Server
12
Response includes Approved Status and updated scope

```
GET /register/s8duje6nfs HTTP/1.1
Accept:  application/json
Host: newalliancebank.auth.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
```

```
HTTP/1.1 200 OK
{
 “client_id”: “s8duje6nfs”,
 “client_secret”: “cf1bdc3d4f5aemd77465df9d”,
 “status”: “Approved”,
 “client_name”: “Seedling”,
 “redirect_uris”: [“https://ezconnect.auth.com/callback”],
 “logo_uri”: “https://seeling.com/logo.png”,
 “scope”: “ACCOUNT_DETAILED TRANSACTIONS CUSTOMER_CONTACT”,
…
}
```

1. Data Access Platform submits DCR request to Update the data cluster scope for an existing Active data recipient
2. Data Provider receives the DCR request, looks up existing client ID, automatically compares the metadata fields, and determines that an approval is required. Change Request is marked for Internal Provider review.
3. Data Provider sends response to Data Access Platform with existing Client ID, Client Secret and active metadata including scope. Status is Tentative, which means that the Client ID is still active for existing permissioned consumers, but there is an approval pending.
4. Data Access Platform persists the “Tentative” status for the Recipient. This triggers a process to check status with the Provider on a daily basis. **The current user experience is as follows:**
   a. **Existing consumers of Recipient Application are active with original scope**
   b. **New consumers only have access to the original scope in the consent process**
5. Data Access Platform submits DCR GET request to check status. Data Access Platform DCR status checks **SHOULD** should be performed a in a maximum of hourly increments
6. Provider receives request and queries Recipient details
7. Provider responds with latest status which is still “Tentative”
8. Data Access Platform maintains the “Tentative” status for the Recipient as well as the daily status check process. **Current user experience is as follows:**
   a. **Existing consumers of Recipient Application are active with original scope**
   b. **New consumers only have access to the original scope in the consent process**
9. Provider Approves Recipient Scope change

10. Data Access Platform submits DCR GET request to check status
11. Provider receives request and queries Recipient details
12. Provider responds with latest status, which is updated to Approved
13. Data Access Platform persists Approved status. **The User Experience is as follows:**
    a. **Existing consumers of Recipient Application are active with original scope**
    b. **New consumers have access to the new scope in the consent process**
    c. **Existing consumers will need to reconsent to gain access to the additional Scope**

3.4.1 Update Handling Guidelines

Data Providers **SHOULD** allow simple Recipient and Intermediary metadata changes to be made immediately with no approvals. The following are the high level guidelines for Data Providers to consider when establishing rules around Data Recipient chain metadata approvals and forced consumer re-consents.

Name	Approval Required	Re-Consent Required	Notes
Recipient Level Data			
client_name	No	No	
description	No	No	
redirect_uris[]	Yes	No	
logo_uri	No	No	
client_uri	No	No	
contacts[]	No	No	
scope[]	Variable	Yes	If the scope is expanded, existing consents remain active with the old scope. Re-consent required to obtain a new scope.
		No	If the scope is reduced, existing consents remain active with the new reduced scope.

Name	Approval Required	Re-Consent Required	Notes
duration_type[]	No	Yes	If the duration type is expanded, existing consents remain active with the previous duration type. Re-consent is required to obtain the new duration type.
		No	If the duration type is reduced, the existing consents remain active with the new reduced duration type.
duration_period	Variable	Yes	If the duration period is expanded, existing consents remain active with previous duration period. Re-consent required to obtain new duration period.
		No	If the duration period is reduced, existing consents remain active with the new reduced duration period.
lookback_period	Variable	Yes	If the lookback period is expanded, existing consents remain active with the previous lookback period. Re-consent is required to obtain the new lookback period.
		No	If the lookback period is reduced, existing consents remain active with the reduced lookback period.
registered_entity_name	Variable	Variable	
registered_entity_id	Variable	Variable	
registry	Variable	Variable	
Intermediary Level Data			
intermediaries[]	Variable	Variable	
name	Variable	Variable	

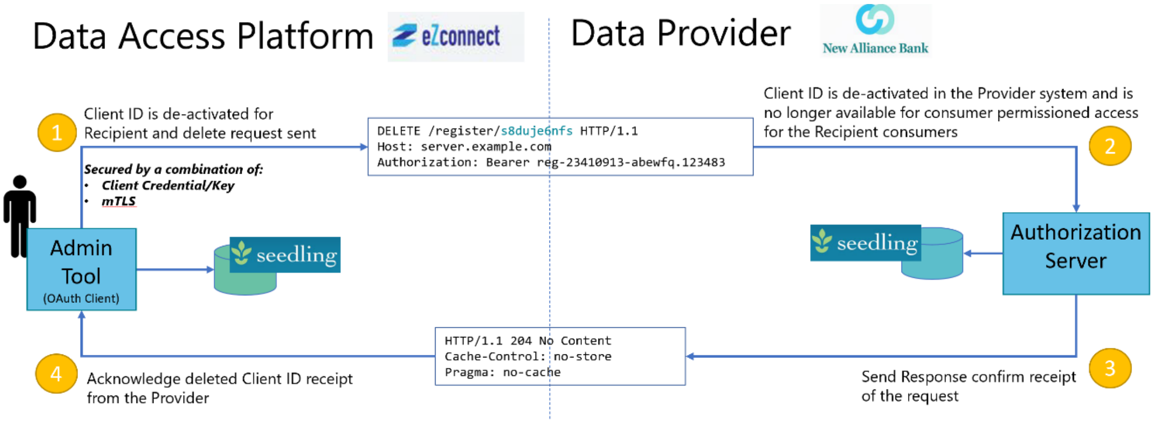
Name	Approval Required	Re-Consent Required	Notes
description	No	No	
uri	No	No	
logo_uri	No	No	
contacts[]	No	No	
registered_entity_name	Variable	Variable	
registered_entity_id	Variable	Variable	
registry	Variable	Variable	

3.5 Revoke Registration

The Recipient registration life cycle ends when a Data Recipient is deleted. Data Providers **SHOULD** implement a DELETE capability as part of their DCR implementation. Upon deletion of a Data Recipient a Provider **SHOULD** immediately revoke all consumer tokens associated with the deleted Client ID.

Here is the standard process flow for Revoking Recipient Registration.

Revoke Registration (DELETE)



Client ID is de-activated for Recipient and delete request sent

Secured by a combination of:

- *Client Credential/Key*
- *mTLS*

```
DELETE /register/s8duje6nfs HTTP/1.1
Host: server.example.com
Authorization: Bearer reg-23410913-abewfq.123483
```

Client ID is de-activated in the Provider system and is no longer available for consumer permissioned access for the Recipient consumers

Send Response confirm receipt of the request

```
HTTP/1.1 204 No Content
Cache-Control: no-store
Pragma: no-cache
```

Acknowledge deleted Client ID receipt from the Provider

1. Data Access Platform sends request to delete a Data Recipient Registration with the Provider
2. Data Provider receives the request and disables the Client ID
    a. Data Providers **SHOULD** revoke all active tokens for the data Recipient
3. Data Provider sends the response acknowledging that the Client ID is deleted
4. Data Access Platform confirms Provider disablement of Client ID Expected User Experience:
    a. Data Access Platform **SHOULD** no longer list the Provider during the Select Data Provider process (Step 3 in UX Guidelines):
        I. If a consumer is allowed to select the Provider, the consent redirect **SHOULD** fail due to an invalid Client ID
    b. The Data Access Platform **SHOULD** stop all "/token" and FDX API calls to the Provider. If the Data Access Platform makes additional refresh attempts using existing refresh tokens:
        I. The next "/token" refresh call for each permissioned consumer **SHOULD** fail with an invalid Client ID error from the Data Recipient
        II. Following the failure, the Data Access Platform should stop subsequent attempts from the Data Recipient

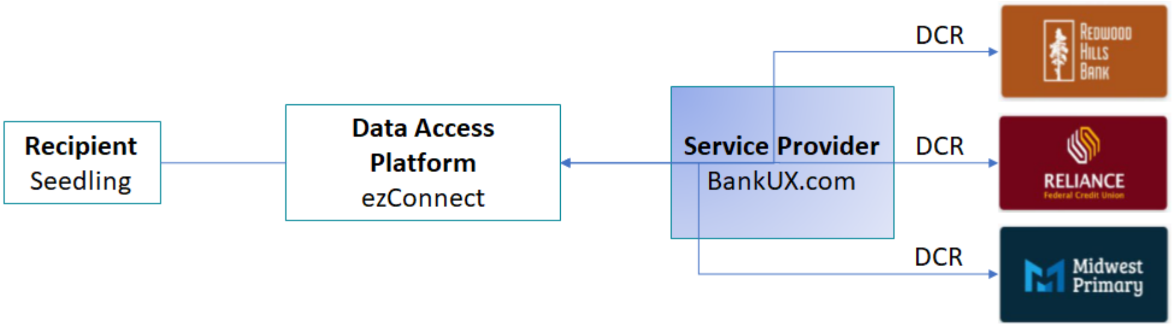
3.6 Financial Service Providers/Multiple Site Scenarios

Many Financial Data Providers have more than one consumer permissioned end point (Site). Subsequently, a consumer could have a different login for each of those end points. A few common examples include:

1. A Bank that offers multiple banking products under separate consumer URLs with entirely separate applications such as Retail, Private, Small Business or Corporate Banking
2. A Bank or Service Provider that offers merchant branded credit cards, each with a separate consumer URL and possible a separate Mobile App
3. A Digital Service Provider who hosts online and mobile banking services for many banks
4. A Digital Service Provider who hosts branded Wealth management or Employee Benefit products including 401ks, HSAs, or 529 plans

Providers with multiple Sites **MUST** provide a separate DCR end point for each Site and generate a separate Client ID for each Recipient registration to each Site.

The Online and Mobile Banking Service Provider example will be used to illustrate how a multi-site Provider should handle recipient Registrations.



In this example, service Provider BankUX.com hosts services for Redwood Hills Bank, Reliance Federal Credit Union, and Midwest Primary. Data Access Platform ezConnect has an agreement and integration with BankUX.com and wants to register Seedling as a new Recipient to each of the hosted banks. ezConnect will need to make a separate DCR request for Seedling for each of the bank registrations.

Redwood Hills Bank DCR Request

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: bankux.auth.com/redwoodhillsbank
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezconnect.com/",
    "logo_uri": "https://ezconnect.com/logo.png"
  }]
}
```

Reliance Federal Credit Union DCR Request

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: bankux.auth.com/reliancefcu
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezconnect.com/",
    "logo_uri": "https://ezconnect.com/logo.png"
  }]
}
```

Midwest Primary DCR Request

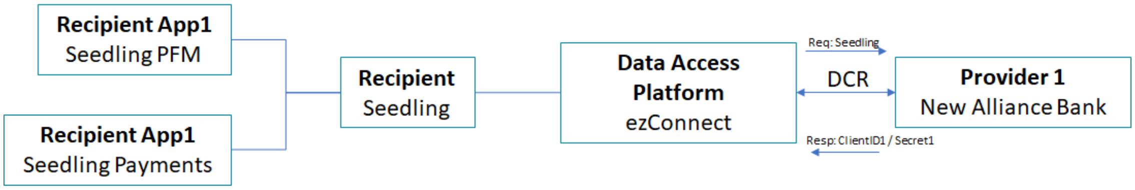
```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: bankux.auth.com/midwestprimary
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": "Manage your finances all in one place from any device"
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezconnect.com/",
    "logo_uri": "https://ezconnect.com/logo.png"
  }]
}
```

3.7 Recipients with multiple Applications or Use Cases

- A Data Recipient **MAY** choose to service all Permissioned Data sharing use cases through a single customer consent to a Provider, which subsequently requires a single Client ID Registration through DCR.
- If a Data Recipient has a single consent to cover all Data Recipient Application use cases, and the Data Provider requires scope during registration, then the Client ID Registration (serviced by a Data Access Platform or Direct Data Recipient) **MUST** include Scope which covers all use cases.
- A Data Recipient **MAY** choose to require a separate consumer consent action for two or more separate applications, which subsequently requires a separate Client ID Registration for each Recipient Application.
- If a Data Recipient has multiple applications that each require a separate consent, and the Data Provider requires scope during Registration, then the Client ID Registration (serviced by a Data Access Platform or Direct Data Recipient) **MUST** register for the minimal Scope required for each separate Client ID registration.
- If a Data Recipient has multiple applications that each require a separate Client ID Registration, then the client_name and description **MUST** be different for each registration and **MUST** be human-readable and clear about the Data Recipient application use case to which the consumer is consenting to share their Provider data.
- If a Data Recipient utilizes multiple Data Access Platforms for consumer permissioned data sharing services, the Data Access Platforms **MAY** register the Data Recipient with the same client_name.
- Data Providers **MUST** be able to support multiple Data Recipient registrations with the same client_name. If the same client_name is registered by multiple Data Access Platforms for the same Data Recipient, then the Data Provider **MUST** respond with a separate Client ID/Secret for each registration.

3.7.1 Recipient with a single DCR covering Multiple Applications/Use Cases

A Data Recipient organization may choose to service all permissioned data sharing use cases through a single customer consent. In this case, the Data Access Platform makes a single DCR request with a superset of the entire scope required to support all of the Data Recipient Use cases.

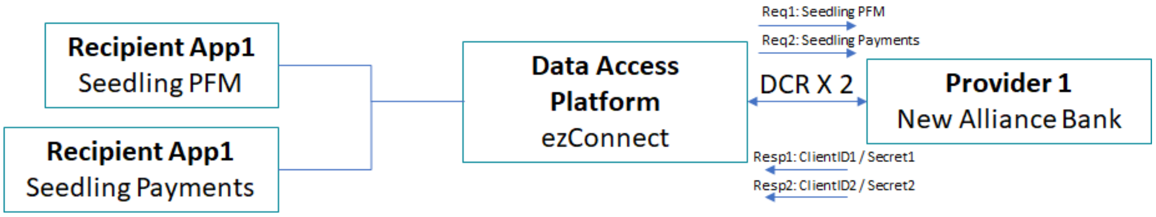


DCR Request for Seedling covering multiple Applications/Use Cases

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: NewAllianceBank.App-Register.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": "Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@ezconnect.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS CUSTOMER_CONTACT",
  "duration_type": ["persistent"],
  "lookback_period": 365,
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezConnect.com/",
    "logo_uri": "https://ezConnect.com/logo.png",
    "contacts": ["support@ezconnect.com"]
  }]
}
```

3.7.2 Recipient with separate DCR for each Application/Use Case (Single DAP)

A Data Recipient organization may operate multiple financial applications and to choose to have a separate consumer permissioned event for each application. In this case, the Data Access Platform registers for a separate Client ID for each of the Recipient application. The registration would include a separate client_name that describes the Recipient Use Case or Application name. If applicable, the registration will include a separate scope to minimize consumer access to each separate application use case.



DCR Call for Recipient App1 - PFM

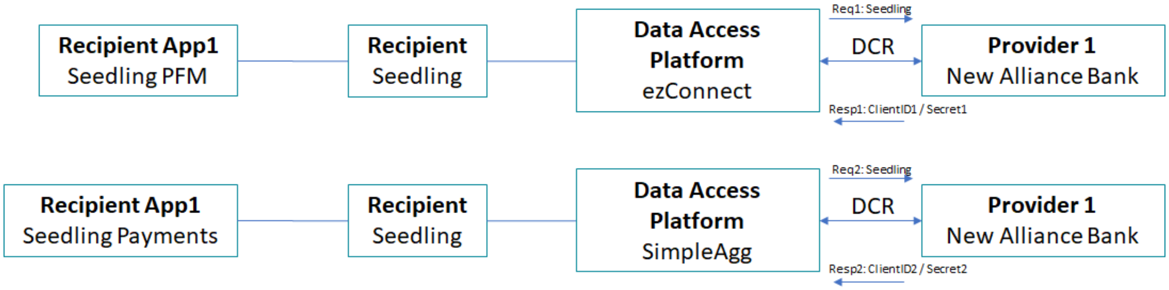
```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: NewAllianceBank.App-Register.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling PFM",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@ezconnect.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS",
  "duration_type": ["persistent"],
  "lookback_period": 365,
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezConnect.com/",
    "logo_uri": "https://ezConnect.com/logo.png",
  }]
}
```

DCR Call for Recipient App2 - Payments

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: NewAllianceBank.App-Register.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling Payments",
  "description": “Link your Bank account to fund your Seedling Wallet",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@ezconnect.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED CUSTOMER_CONTACT",
  "duration_type": ["time_bound"],
  "duration_period": 90,
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezConnect.com/",
    "logo_uri": "https://ezConnect.com/logo.png"
  }]
}
```

3.7.3 Recipient using multiple Data Access Platforms

A Data Recipient organization may operate multiple financial applications through multiple different data access platforms. Each Data Access Platform will need to separately register the Data Recipient for the use case(s) that they power.



DCR Call for Recipient App1 from ezConnect

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: NewAllianceBank.App-Register.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": “Manage your finances all in one place from any device",
  "redirect_uris": ["https://ezconnect.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@ezconnect.com","support@seedling.com"],
  "scope": "ACCOUNT_DETAILED TRANSACTIONS",
  "duration_type": ["persistent"],
  "lookback_period": 365,
  "intermediaries": [
  {
    "name": "ezConnect",
    "description": "Best in class financial experiences for your customers",
    "uri": "https://ezConnect.com/",
    "logo_uri": "https://ezConnect.com/logo.png",
    "contacts": ["support@ezconnect.com"]
  }]
}
```

DCR Call for Recipient App2 from SimpleAgg

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: NewAllianceBank.App-Register.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1
{
  "client_name": "Seedling",
  "description": "Link your Bank account to fund your Seedling Wallet",
  "redirect_uris": ["https://simpleagg.auth.com/callback"],
  "logo_uri": "https://seeling.com/logo.png",
  "client_uri": "https://seeling.com/",
  "contacts": ["support@SimpleAgg.com", "support@seedling.com"],
  "scope": "ACCOUNT_DETAILED CUSTOMER_CONTACT",
  "duration_type": ["time_bound"],
  "duration_period": 90,
  "intermediaries": [
  {
    "name": "SimpleAgg",
    "description": “Access over 1 million Financial Data Sources",
    "uri": "https://SimpleAgg.com/",
    "logo_uri": "https://SimpleAgg.com/logo.png",
    "contacts": ["support@SimpleAgg.com"]
  }]
}
```

4.0 Securing Application Registration

The following security requirements apply to Provider DCR implementations. These requirements are in line with general FDX Security Guidelines including FDX adoption of the FAPI Standard.

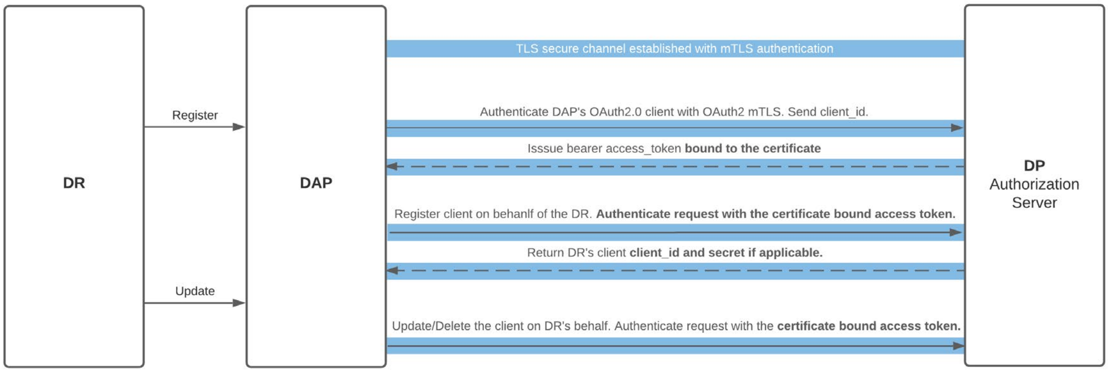
1. Data Recipient registration **MUST** be secured with a method that ensures the Recipient Registration request is from a known Partner (Data Access Platform or Direct Recipient).
2. Data Providers **MUST** secure the Dynamic Client Registration end point with OAuth 2.0 authentication as defined in IETF RFC 6749 (Access Token).
3. Data Providers **MUST** adhere to FAPI Authorization server requirements for Partner Client Authentication and sender-constrained access tokens as stated in section 5.2.2 #5, #6, and #14 (FAPI Part 2 Authorization Server).
4. Partners **MUST** use and Data Providers **MUST** enforce the practices in IETF RFC 8705 to bind the Credential (Bearer Token) to the secure channel during DCR call execution.

4.1 Recipient Registered by Data Access Platform

This diagram illustrates one of the recommended setup and process flows for securing Dynamic Client Registration implementations between a Data Access Platform and Data Provider.

Data Access Platform managed Data Recipient Client Registration via DCR

Recommended - Strong DAP client authentication; use of **sender constrained access tokens.**


DR
Register
Update
DAP
TLS secure channel established with mTLS authentication
Authenticate DAP's OAuth2.0 client with OAuth2 mTLS. Send client_id.
Isssue bearer access_token bound to the certificate
Register client on behanlf of the DR. Authenticate request with the certificate bound access token.
Return DR's client client_id and secret if applicable.
Update/Delete the client on DR's behalf. Authenticate request with the certificate bound access token.
DP
Authorization
Server

4.1.1 mTLS Setup Process

The first step in the process involves a partnership integration between the Data Access Platform and Data Provider.

- Certificates shared and registered between DAP and Provider

- Unique Partner Client ID generated and shared by Provider to DAP

4.1.2 Obtain Access Token for DCR

Prior to a DCR call, a short lived Access token is obtained from the Authorization server.

CURL EXAMPLE

```
curl --request POST \
--data "grant_type=client_credentials" \
--data "scope=client_register" \
--data "client_id=PARTNER_CLIENT_ID" \
--cert "myClientCertificate.pem" \
--key "myClientCertificate.key.pem" \
https://server.example.com/register/token
```

HTTP EXAMPLE

```
POST /register/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Host: server.example.com

grant_type=client_credentials
&scope=client_register
&client_id=PARTNER_CLIENT_ID
```

RESPONSE

```
{
  "access_token": "MzRDdRb21aYWdraDA1OV94UnlPTl",
  "token_type": "Bearer",
  "expires_in": 600,
  "scope": "client_register"
}
```

4.1.3 DCR Call with Access Token

Once an Access Token is obtained, the DCR call can be made to register the Data Recipient. Note that the same process applies for GET, PUT and DELETE calls as well. Also, in most DCR implementations, the Authorization server acts as the Resource Server.

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com
Authorization: MzRDdRb21aYWdraDA1OV94UnlPTl
{
  "client_name": "Data Recipient Name",
   ...
}
```

4.2 Direct Recipient Registration

Direct Data Recipient to Data Provider DCR integrations follow the exact same FDX security standards, integration steps and execution steps as a Data Access Platform.