FINANCIAL™ DATA EXCHANGE

Control Considerations

V4.0
December 2023

Legal Notice

Financial Data Exchange, LLC (FDX) is a standards body and adopts this Control Considerations document 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
4.0	Updated content to align with FAPI 1.0/2.0, FDX Security Model. Global revisions and redesign for a more concise document. Changed document status to publicly available.	**December 2023**
3.3	Updates to images	**October 2021**
3.2	Updates to Security Model references	**September 2020**
3.1	Content edits	**August 2020**
3.0	Content edits	**October 2019**
2.1	Initial Document Release	**September 2019**

Contents

Introduction and Goals

Audience

This FDX control considerations document has been written for identity and security practitioners and stakeholders in the North American financial services sector. This would include developers, security, and incident response teams, as well as advocates for consumer privacy and security.

Opportunities

The North American financial services industry has advanced significantly over the last few years. Not too long-ago user financial data was locked up behind propriety APIs, accessible only via one-off business agreements, or sometimes only available via screen scraping by borrowing the user's login credentials.

We now live in a world where FI's, aggregators, and other related parties freely share financial information via FDX standard APIs that follow best practices for security and interoperability while providing users control over the sharing of their information via fully integrated consent mechanisms.

To best implement, or fully utilize these APIs requires using best current practices for the fin-tech ecosystem. For the purposes of FDX control considerations, this ecosystem is broken down into three main areas:

- User lifecycle management
- Registration between financial ecosystem participants
- Secure usage of the FDX APIs

Challenges

The nature of these services, if not properly implemented, can pose significant security risks to the end users, the data providers, and the data recipients. Guidance from several sources address control practices that should be considered. The security guidelines in this document follow best practice recommendations based on NIST, OIDF, and ISO standards.

Goals

The goal of this document is to provide a consistent set of control considerations and show how they can be applied to provide a zero-trust framework for several specific flows such as user registration, third party client registration, and user authentication.

FDX strives to be thorough, concise, flexible, and task-oriented in its artifacts, outlining framework and general approaches, Control Considerations included. Our goal is that the framework and general approaches we are advocating are flexible enough to accommodate any additional considerations that readers may want to include.

This is a living document. If there are key aspects that you would like to see added, please reach out to FDX support at fdxsupport@financialdataexchange.org.

This initial version of the document is TLP White and may be freely distributed. A follow on TLP Amber version of this document may be produced to include references to relevant FDX RFC's and call out any normative requirements for FDX members who wish to pass the FDX certification program.

Financial Data Exchange (FDX)

The Financial Data Exchange (FDX) is a non-profit financial industry organization dedicated to promoting and enhancing a common interoperable standard and operating framework for sharing consumer financial data. FDX puts consumers in control of their personal financial data. Open to all financial institutions and fintech companies, FDX facilitates collaboration in the development, growth, and industry acceptance of the standard and security requirements. For more information and to join, visit www.financialdataexchange.org.

Guiding Principles and Standards

FDX provides a common taxonomy for the Open Banking ecosystem in our publicly available *Taxonomy of Permissioned Data Sharing* document. Please refer to this document for a definition of terms.

Our recommendations and examples of best practices lean heavily on industry and government documents. In particular, we have referenced the following documents heavily:

NIST 800-63-4

800-63 https://pages.nist.gov/800-63-4/sp800-63.html

800-63a https://pages.nist.gov/800-63-4/sp800-63a.html

800-63b https://pages.nist.gov/800-63-4/sp800-63b.html

800-63c https://pages.nist.gov/800-63-4/sp800-63c.html

OAuth 2.0 Authorization Framework https://www.rfc-editor.org/rfc/rfc6749

OpenID Foundation Financial-grade API Security Profile 1.0 - Part 2: Advanced https://openid.net/specs/openid-financial-api-part-2-1_0.html

Additional standards that valuable when considering best practices and security controls can be found in the following publications –

- NIST SP 800-44 Guidelines on Securing Public Web Servers
- NIST SP 800-95 Guide to Secure Web Services
- NIST SP 800-53 SC-5 Denial of Service Protection

- NIST SP 800-41 Guidelines on Firewalls and Firewall Policy
- OWASP Top 10 Web App
- OWASP Top 10 API Sec
- OWASP Automated Threats
- NIST SP 800-53 SC-28 Protection of Information at Rest
- ISO29100 and ISO29134
- End user device malware detection

Use Case Best Practice and Control Considerations

Below are specific examples of what the authors consider to be industry best practice for securely implementing key use cases. These examples are standards-based wherever possible and adhere to FDX and other security requirements and frameworks. These examples also follow the tenets of zero-trust, ensuring that there is no inherent or implicit trust between components – all connections between any two entities are both authenticated and authorized.

Enrollment and Identity Proofing

Enrollment and identity proofing are two closely related, but technically separate steps that take place when a customer/user wants to participate in the banking or other products/services offered by an FI/Non-FI.

FDX considers the criteria and best practices for enrollment and identity proofing to come from the following two NIST documents:

- SP 800-63-3-4 | Digital Identity Guidelines
- SP 800-63A-4 | Enrollment & Identity Proofing

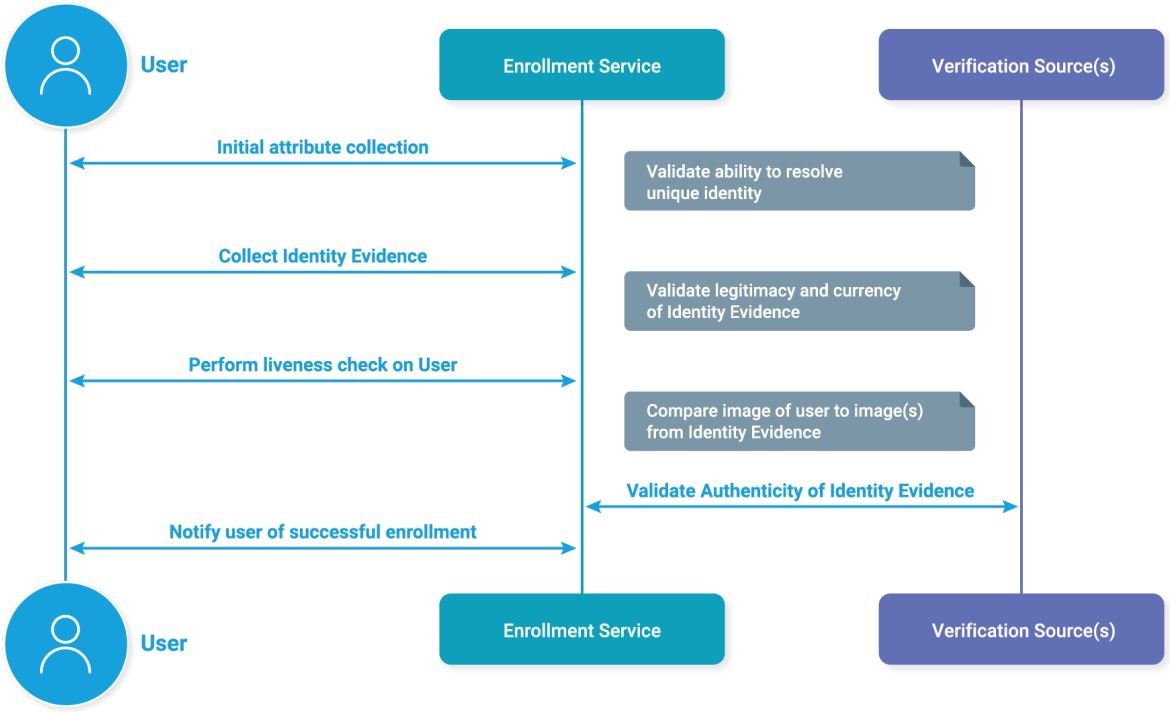
The overall flow and steps for enrollment and identity proofing are described below.

Overall Flow

Enrollment and identity proofing looks to accomplish several items:

- Resolution and Attribute Collection
- Identity Proofing
- Authentication Factor Binding

Enrollment and Identity Proofing Flow



Resolution and Attribute collection

The first is resolution – the ability to uniquely distinguish an individual's identity among a given population of subscribers or users. The scope of uniqueness may be as small as a social group or business department, or as large as needing to be globally unique.

This is accomplished by the collection of attributes: Name, email address, physical address, etc. Some of these attributes will be used to uniquely identity the user, an activity known as disambiguation. Other attributes are collected when needed by the service or application to which the user is subscribing or using.

It is important that the principles of privacy by design and data minimization be followed.

- Only information truly needed for serving the subscriber should be collected.

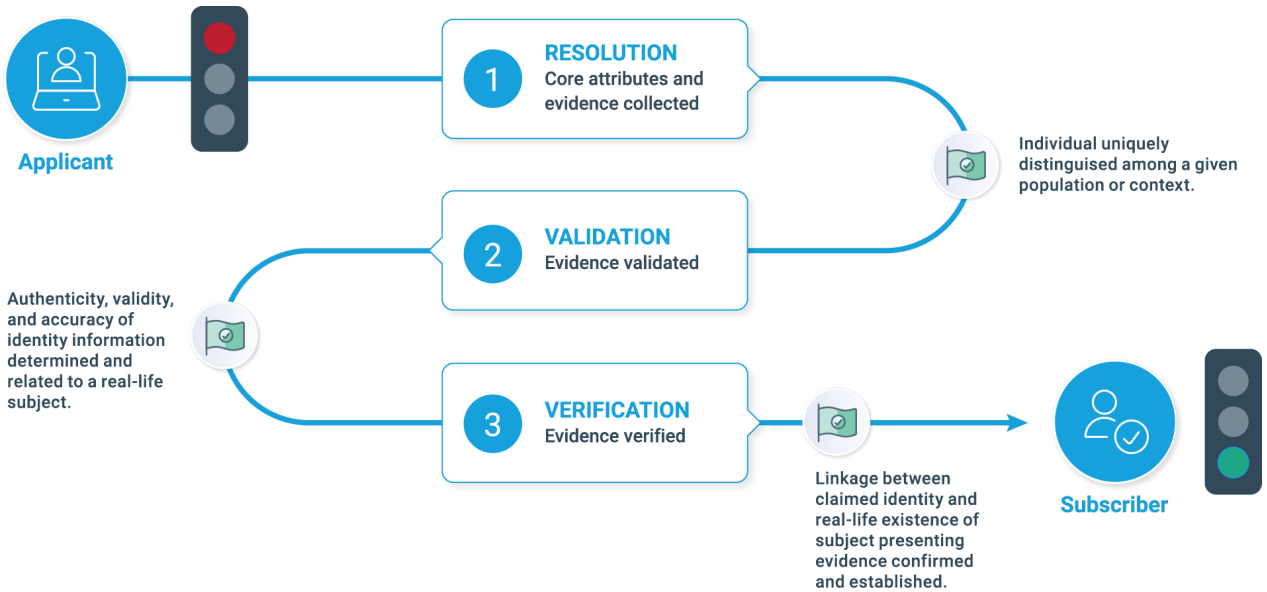
- The data should be stored and encrypted using best current practices, and managed in observance of any consumer data rights or other applicable regulations.
- Access to the data should be tightly controlled to only persons and processes having a legitimate need for data access.
- The user should be informed of and consent to any sharing of data with third parties.
- Data should only be kept for as long as necessary.
- It must be possible for the user to request removal of the data.

Identity Proofing

While we have collected attributes that should uniquely identity the user within the required population scope, we do not necessarily have any linkage between those attributes and a "real" identity. For many scenarios that is fine, but in the financial services ecosystem we are typically going to want to validate that the claimed identity corresponds with the applicant's real-world identity.

The level of rigor, and the processes needed to do this, are defined as Identity Assurance Levels (IAL) in NIST 800-63-A.

This diagram supplied by NIST 800-63-4A provides a good view of the identity proofing process and its steps, which we will further define within our larger context of enrollment.



Resolution

The registration service/CSP collecting one or more pieces of identity evidence, such as a driver's license or passport.

Resolution and disambiguation were accomplished during the attribute collection phase described above.

Validation

The CSP validates the authenticity, accuracy, and currency of the presented evidence.

The CSP first validates the evidence to make sure that it is legitimate and unexpired. The CSP then makes sure that the evidence is authentic and corresponds to an actual identity by checking them against authoritative and credible sources.

Verification

The CSP verifies that the verifies that the person submitting the identity evidence is in fact the person referred to by the. Identity evidence.

The most common method of achieving this is to have the applicant take a photo of themselves with liveness check to ensure they are not doing something like taking a photo of a photo. The photograph is then compared by the CSP to the picture(s) on one or more of the supplied pieces of evidence to ensure this is the same person.

Authentication Factor Binding

Now that we have established and validated a unique identity, we need a way for the user to authenticate themselves when they want to use the system. This is done by collecting and binding authentication factors to the identity.

The number and type of authentication factors collected is determined by the value of the resources. NIST-88063-B lists a series of Authentication Assurance Levels (AAL) with increasing number and security of authentication factors, including phishing resistant factors.

An exhaustive description of the strengths and weaknesses of authentication factors is outside the scope of this document but broadly there are three different kinds of authentication factors:

- Knowledge factors – "something you know". These are shared secrets between the CSP and the user. Password is the most common, but knowledge-based authentication is still in use in some use cases such as a user logging in from a new device. The Answers to several questions are collected and bound to the identity. A typical question might be "What was the name of your first pet?".

- Possession factors. – “something you have”. This might be a phone for OTP, or an email account. With possession factors it is important to have a backup factor or factor available in case the user loses or loses access to, the possession factor.
- Biometric factors – “something you are”. The most convenient and often most secure factors.

Enrollment and Identity Proofing Controls

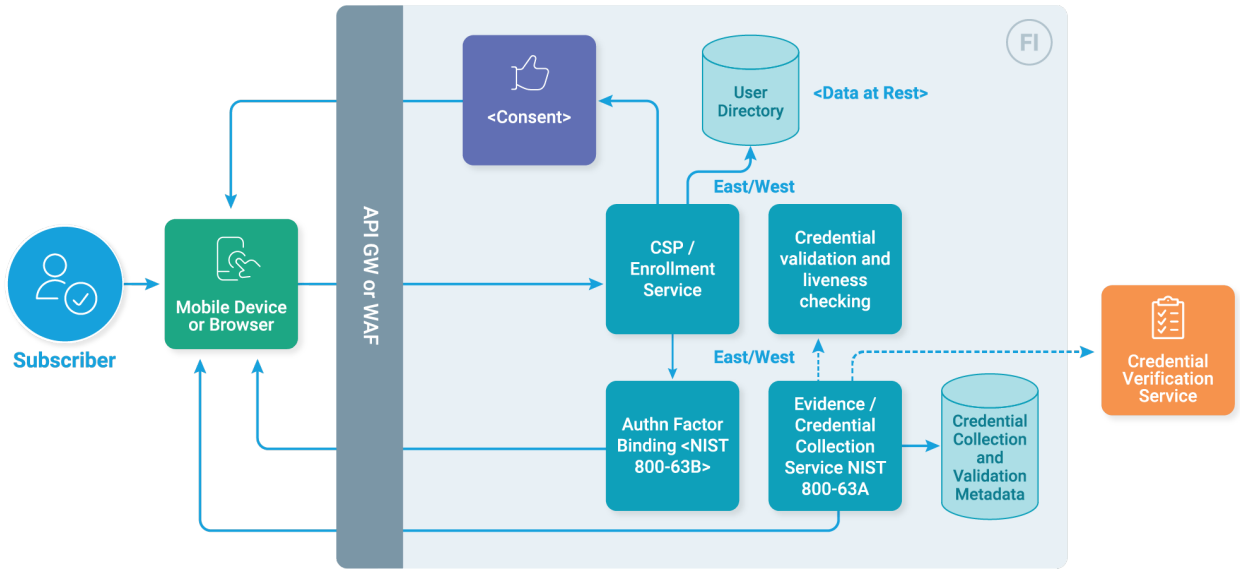
For enrollment and identity proofing there are two key technical controls

– peripheral network security controls such as an API Gateway or WAF hosted by the FI/non-FI or IDS/IPS etc.

– Internal security controls and zero trust best practices within the enrollment service deployment.

For a full listing of generally applicable security control considerations, please see Appendix A

Enrollment and Identity Proofing Controls



User Consent

The following section of the Controls Considerations document outlines the normative criteria to be present for management of Consent; Consent being defined per the FDX Taxonomy of Permissioned Data Sharing.

This section covers the interactions between Data Recipient, Data Provider and the End User with the objective of conveying the user’s intent to grant their permission for data sharing.

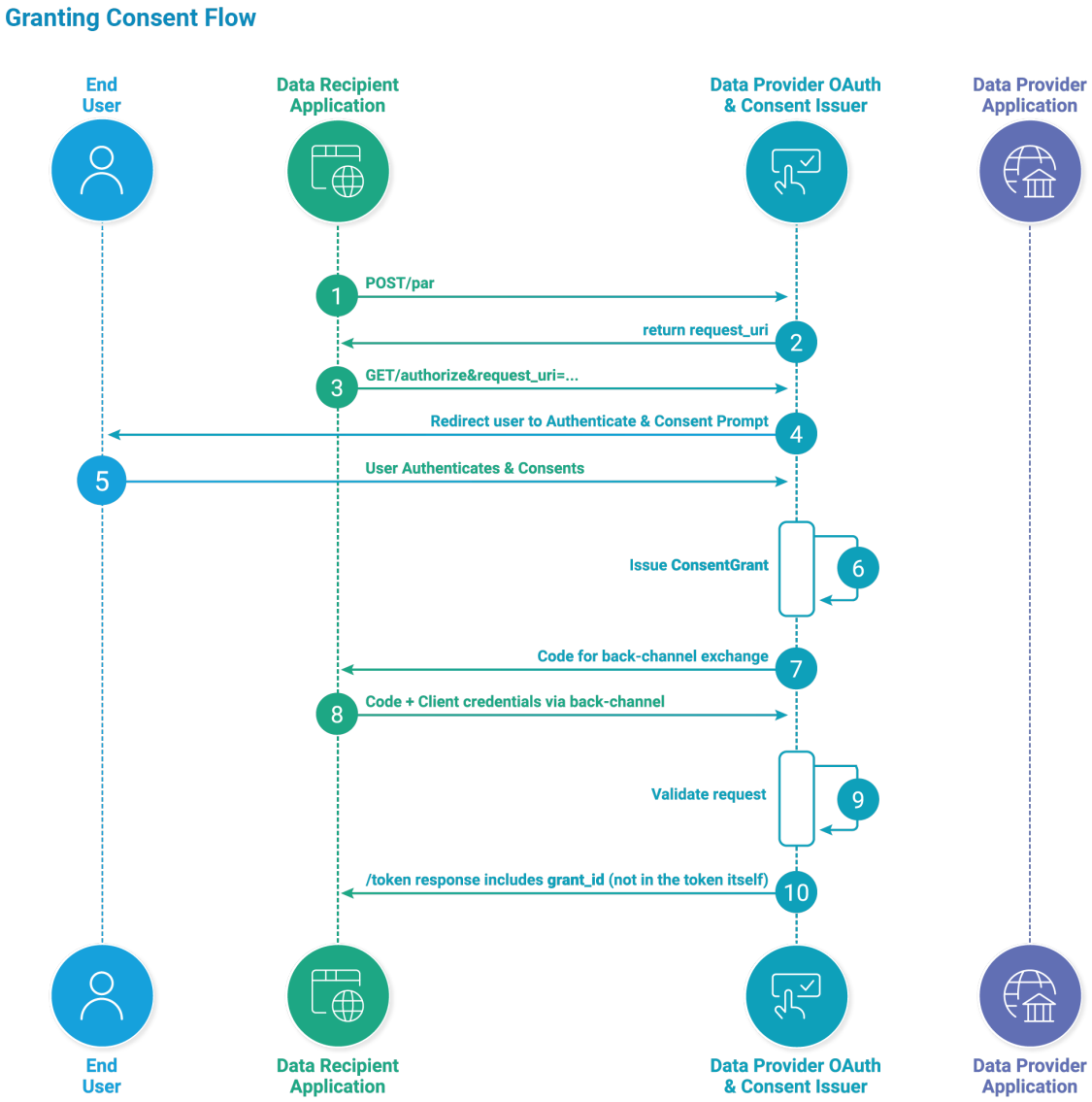
There are two main interactions in the scope of this document: "Granting consent" and "Querying and/or retrieving the current state of consent". Note that "Modification" and "Revocation of consent" are not covered in this document.

Granting consent

Data Recipient initiates this interaction with a request for the creation of consent to Data Provider, such as by OAuth 2.0 Pushed Authorization Requests (the standard for FDX implementers).

The End User session is redirected to authenticate and grant consent with the Data Provider, which will generate the consent grant and return an authorization code to the Data Recipient.

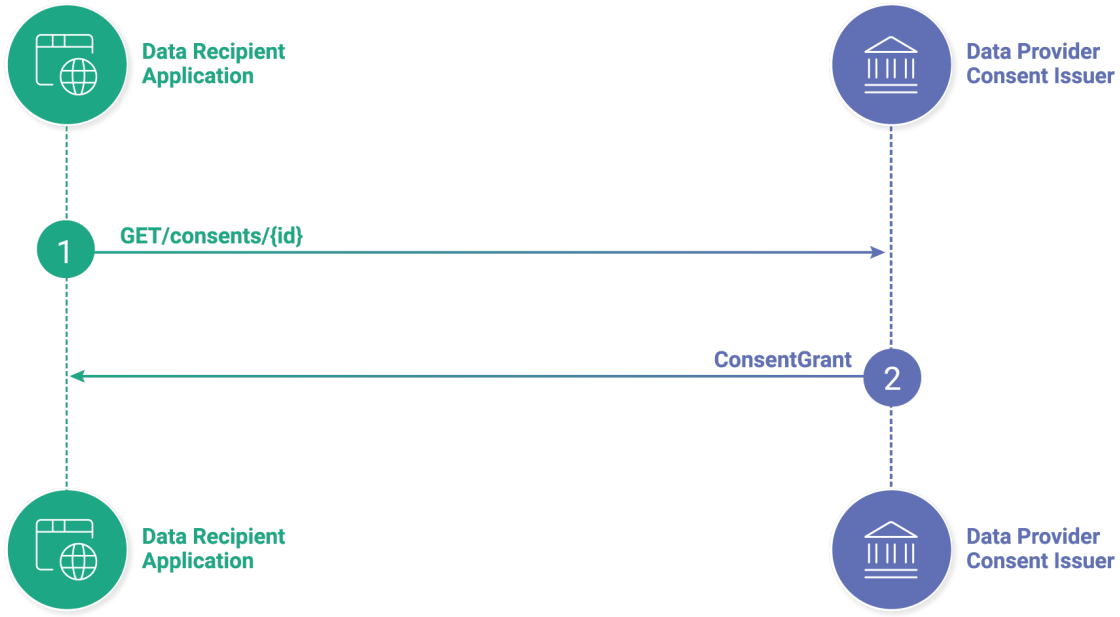
The last step of the "Granting consent" shows Data Recipient exchanging the authorization code for an access token and the consent id.



Retrieving the state of consent

The Data Recipient initiates this interaction by requesting the status of a particular consent (identified by the consent id obtained in the last step of the "Granting consent" interaction) and the Data Provider responds, if the request is valid and authorized, with the ConsentGrant.

Retrieving Consent Flow



User Consent Controls

All APIs accessing the Consent data record are expected to be using current security protocols for authentication, securing data in transit, securing the payload and authorization of capabilities upon authentication. Additionally, the Consent data record is expected to be encoded / encrypted / tokenized by current data protection techniques while being stored at rest.

Financial Institutions (FI's) and non-FI's are expected to ensure deploying controls and processes that would ultimately provide assurance on the three cornerstones of security - Confidentiality, Integrity and Availability while managing Consent.

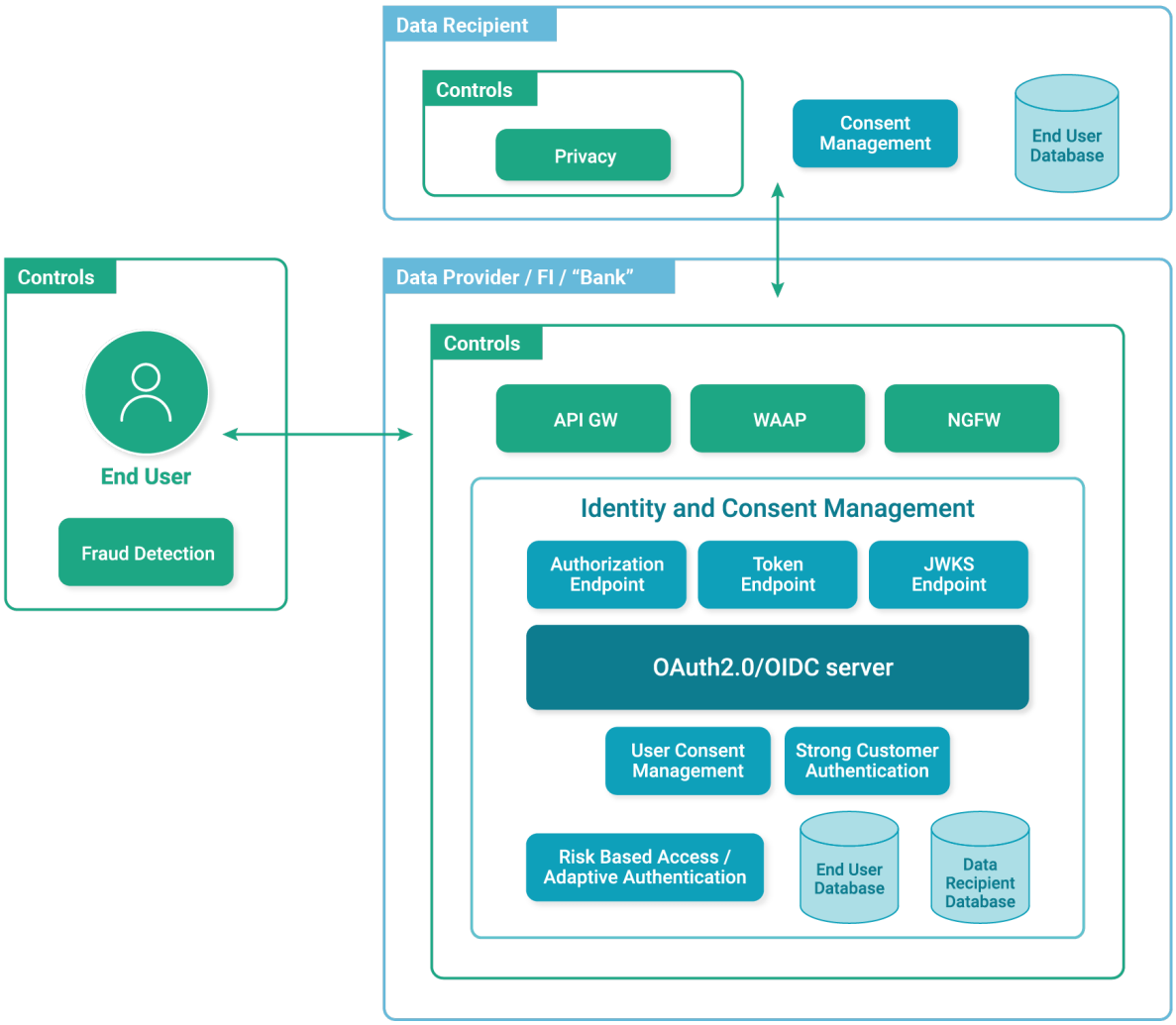
At all times, the technical security controls supporting a Consent should –

- ensure the following from a Consent authentication perspective –
    - that the Consent is granted by an authenticated user and/or,
    - that the Consent is used by an authenticated client on behalf of the user,
- ensure that only authenticated and authorized users and systems are accessing, reading, using and/or modifying a Consent data record,

- ensure that the integrity of the Consent data record is maintained,
- ensure that all copies of the Consent data record – both active and archived, are stored securely,
- ensure that the Consent data record is deleted per requirements.

For a full listing of generally applicable security control considerations, please see Appendix A

Consent Controls



User Authentication

NIST 800-63B-4 defines digital authentication as follows:

"Digital authentication is the process of determining the validity of one or more authenticators used to claim a digital identity."

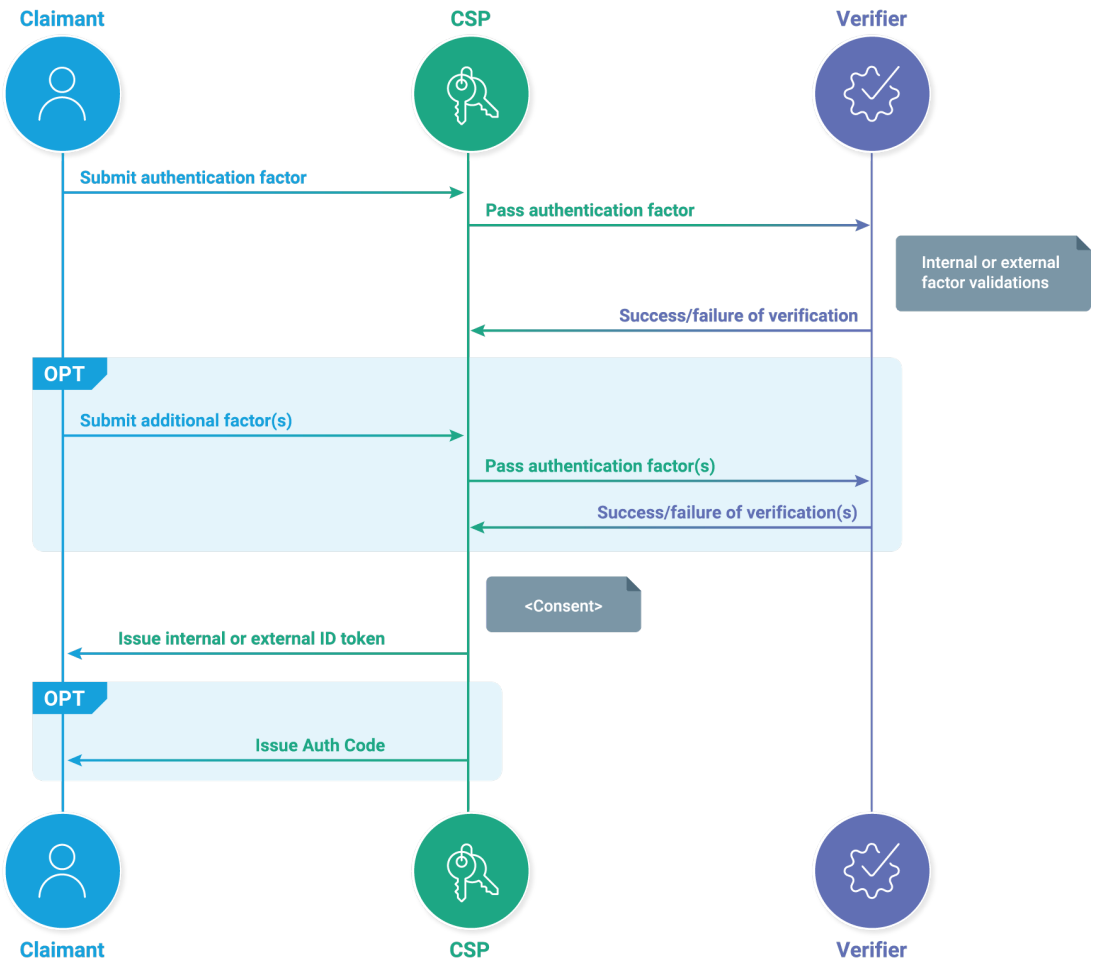
This is not the same as proving that the authenticated user corresponds to any specific real-world identity. There may be no correspondence at all, or the identity may have been verified at a specific IAL level in accordance with NIST 800-63-A

The PSD2 Regulatory Technical specifications on Strong Customer Authentication (SCA-RTS) defines the following high-level requirement for authentication: “Where payment service providers apply strong customer authentication in accordance with Article 97(1) of Directive (EU) 2015/2366, the authentication based on two or more elements categorized as knowledge, possession and inherence shall result in the generation of an authentication code”.

Authentication is typically initiated because the user has not yet established their identity with the CSP/Authorization server. Users can also be asked to re-authenticate after they have already established their identity in situations where either the resource server requires a stronger level of authentication due to the sensitivity/value of a resource the user is attempting to access, or at any time due to elevated risk as determined by either the resource server or the CSP.

With these standards in mind, we look at the authentication ceremony between a claimant and the CSP/Authorization Server.

User Authentication Flow



Authentication Flow

The CSP collects authentication factors(authenticators) from the claimant until the claimant is associated with a subscriber identity to a level that meets a specific AAL or to a level that the CSP or resource server determines matches the sensitivity of the requested access.

An important consideration is that the CSP may look at contextual factors during the authentication and dynamically vary the type and number of authentication factors collected.

Specifying the number and type of authentication factors is outside the scope of this document, but we strongly recommend the use of phishing-resistant authentication factors, such as FIDO and Passkey. They provide a strong combination of security and usability.

These authenticators are then validated by verifiers. These verifiers are typically within the CSP organization, but external verifiers may also be used.

Step-Up Authentication

In the preceding section we have shown authentication mainly within the context of the CSP/Authorization server. The number and type of authentication factors used by the authorization server may vary based on corporate policy, desired AAL, requested access token scope, or dynamic contextual factors such as IP reputation, device posture, etc.

More recently there has been a desire that for certain sensitive operations involving customer data, Data Providers may need to require elevated authentication strengths or recentness.

This makes sense because it is the data provider/resource server that has the most knowledge regarding the resources being requested, and the dynamic context surrounding the actual request for the resource.

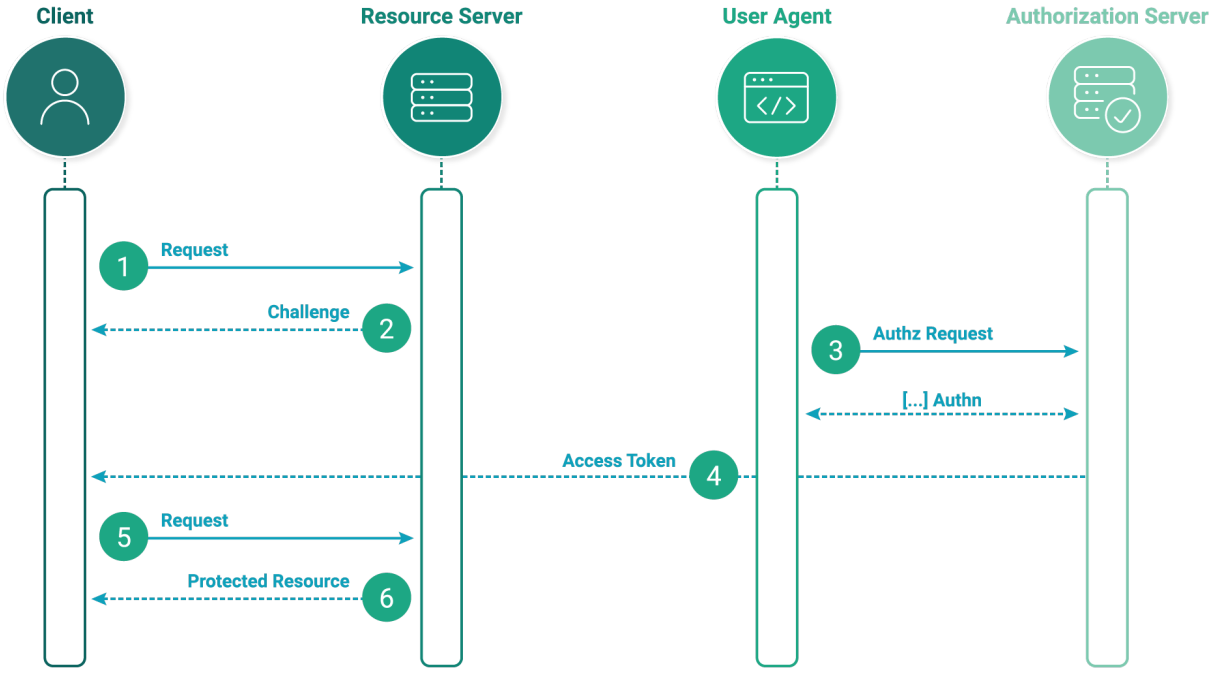
This need has been met in OAuth by introducing a protocol for resource server initiated step-up authentication. (see https://www.rfc-editor.org/rfc/rfc9470.html)

This document extends the error codes collection defined by [RFC6750] with a new value, insufficient_user_authentication, which can be used by data providers/resource servers to signal to the client that the authentication associated with the access token presented with the request does not meet the current authentication requirements.

This document also introduces acr_values and max_age parameters for the Bearer authentication scheme challenge defined by [RFC6750], which the data provider/resource server can use to explicitly communicate to the client the required authentication strength or recentness.

The client can use that information to reach back to the CSP/authorization server with an authorization request specifying the authentication requirements indicated by including the acr_values or max_age authorization request parameters.

Step Up Authentication Flow

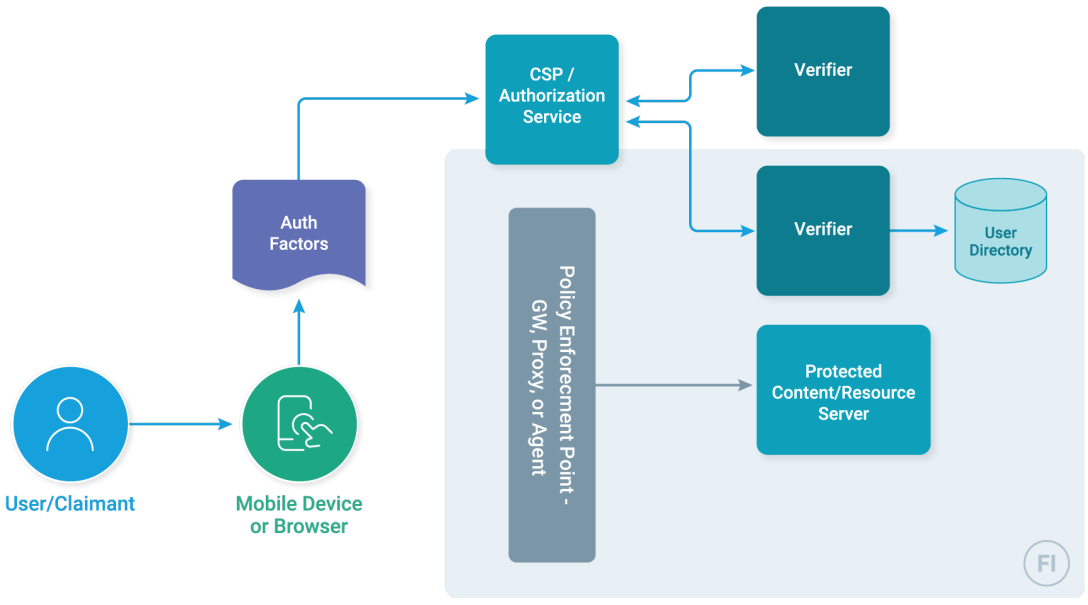


Authentication Controls

Although by definition Authentication is a publicly accessible function, controls and contextual considerations can improve the usability, security, and efficiency of the authentication process.

For a full listing of generally applicable security control considerations, please see Appendix A.

Authentication Controls



User Authorization

User authorization is the process of determining whether a user or a client acting as a delegate has access to a specific instance of a resource. As the FDX security standards mandate FAPI advanced for protecting sensitive resource endpoints, we view significant parts of the authorization ecosystem through the lens of OAuth and OpenID Connect.

These authorization decisions may involve RBAC, ABAC, or full dynamic authorization. Each company will want to determine the authorization framework and number and kind of authorization factors that best meets their needs. The following resources may prove helpful as companies review their authorization framework and needs:

- Gartner CARTA
- Forrester Zero Trust

Authorization flow

There are two time when authorization decisions come into play:

- By the authorization server during access token issuance
- By the resource server when access to the resource is being requested

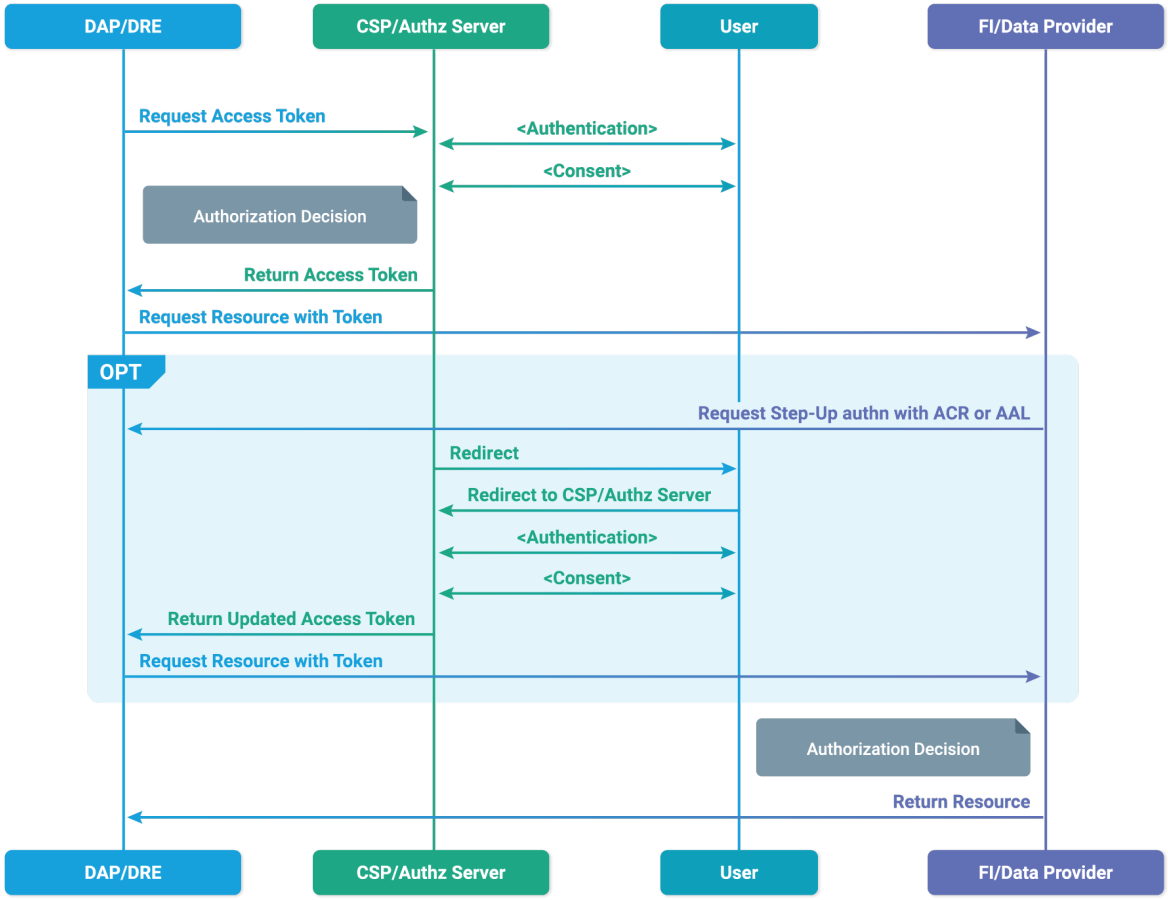
In the sequence diagram below there are the following roles:

The client is an OAuth/OIDC client from either a data recipient or a data access platform.

The resource server is an FI or data provider.

The CSP/authorization server can be part of the data providers infrastructure, or a 3rd party offering trusted by the FI/data provider.

Authorization Flow



Authorization Controls

As mentioned, FDX subscribes to the requirements laid out per FAPI protocols for the purposes of authorization controls. As such the authorization controls in an open banking solution should include, but not be limited to, the following best practice framework:

- Only support confidential clients.
- Authenticate confidential client using TLS_CLIENT_AUTH or PRIVATE_KEY_JWT
- Use PKCE with confidential clients.
- Ensure the requested access token scopes are allowed for this client.
- Only issue sender constrained access tokens using MTLS or DPoP.

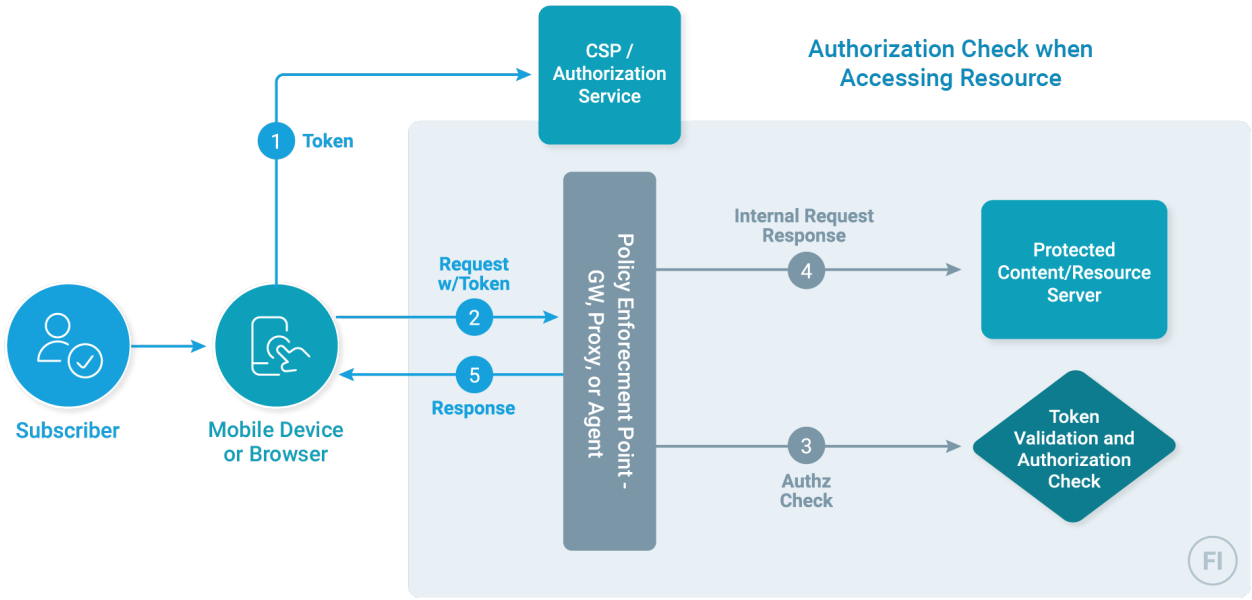
When a client requests a resource from a resource server, the following sequence and considerations apply:

- The client has an access token.

- The token passes the following validity checks - correct audience, known issuer, not expired, and valid signature.
- The token contains the appropriate scopes for accessing the resource.
- Contextual risk checks. Best practice access control continuously monitors and assesses the overall risk around any given transaction. Even though all the preceding checks may have passed, there are still items that should cause a resource server to block the request or take additional steps to validate the request or gain additional trust in the requestor.

For a full listing of generally applicable security control considerations, please see Appendix A.

Authorization Controls



Dynamic Client Registration

Dynamic client registration is the process by which data recipients and/or aggregators register as OAuth clients with data providers/FIs. Best practices as well as FDX requirements are to follow RCS 7591, OAuth Dynamic client registration, accepting only confidential clients as per FAPI advanced.

We also recommend RFC 8414 as a best practice. We also believe that parties should consider implementing Aaron Parecki's Oauth client intermediary metadata draft when the data will be passed through multiple parties, as in the case where a data aggregator is retrieving data from multiple sources for a data recipient, or for example when a data recipient is using a sub-processor as part of identity proofing during an enrollment flow.

Metadata request

Registered clients have a set of metadata values associated with their client identifier at an authorization server, such as the list of valid redirection URIs or a display name.

These client metadata values are used in two ways:

- as input values to registration requests, and
- as output values in registration responses

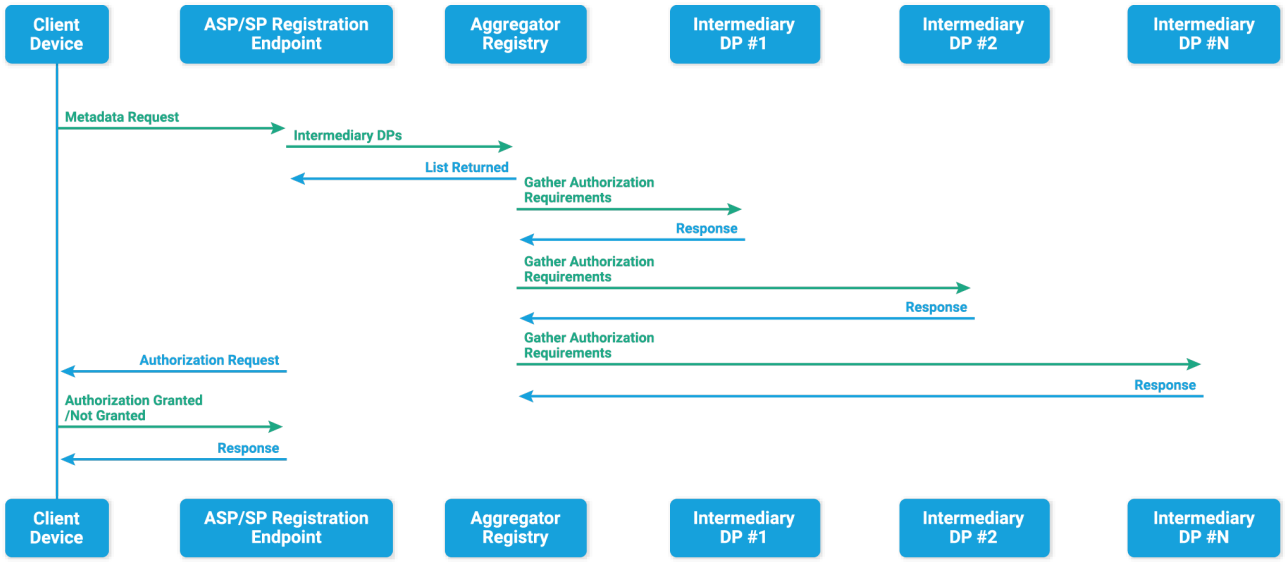
The first thing the client must do is to find the location of the authorization server's client registration endpoint.

RFC 8414 defines a metadata format that an OAuth 2.0 client can use to obtain the information needed to interact with an OAuth 2.0 authorization server, including its endpoint locations and authorization server capabilities. This is an important step as the location of the registration endpoint may vary, and in fact some authorization servers do not support dynamic client registration

Endpoint Registration Flow

Now that the client has the URL of the authorization server dynamic client registration endpoint, it can submit its request. As mentioned, the request and response format are dictated by RFC 7591, OAuth dynamic client registration. The entire flow, including the metadata request, are shown below.

Endpoint Registration Flow



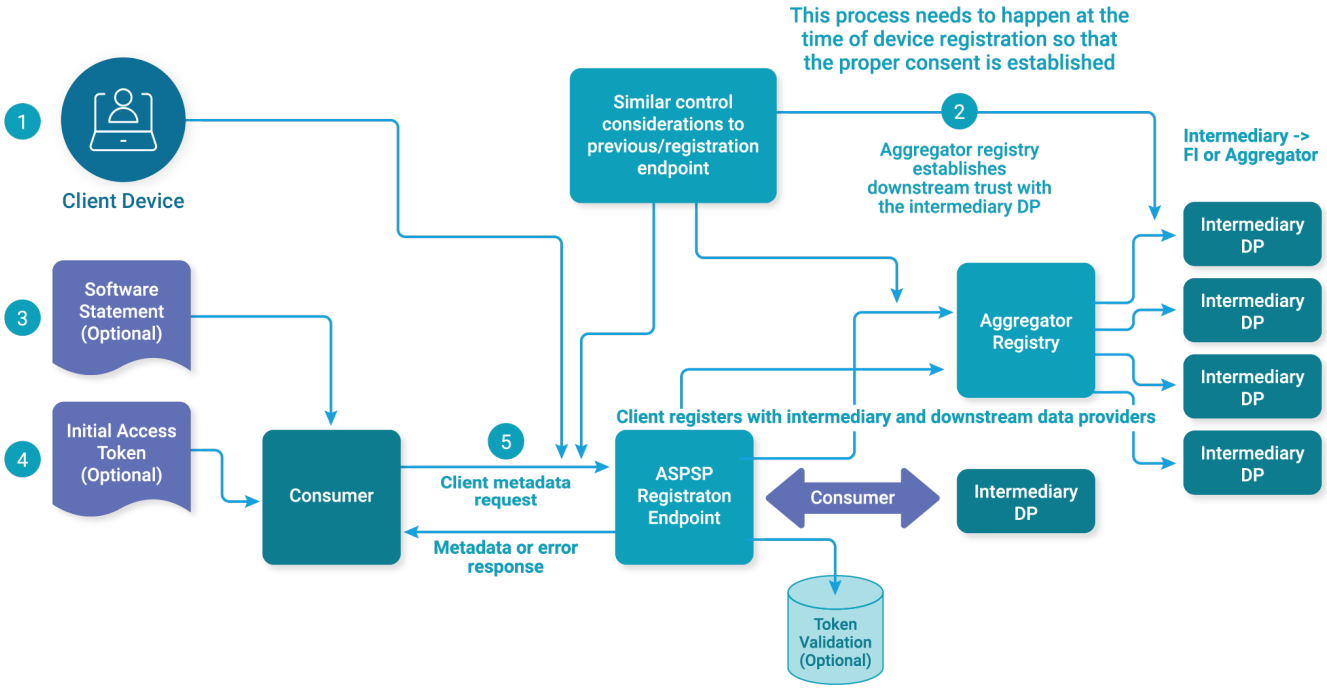
Endpoint Registration Controls

For the registration sequence the following control considerations hold:

- In cases where a third party will collect/augment or present information from a downstream party (an intermediary), it is recommended to provide the list of intermediaries to the authorization server
- A software statement is a JSON Web Token (JWT) [RFC7519] that asserts metadata values about the client software as a bundle. When presented to the authorization server as part of a client registration request, the software statement MUST be digitally signed or MACed using JSON Web Signature (JWS) [RFC7515] and MUST contain an "iss" (issuer) claim denoting the party attesting to the claims in the software statement.
- It is RECOMMENDED that software statements be digitally signed.
- It is RECOMMENDED that software statements contain the "software_id" claim to allow authorization servers to correlate different instances of software using the same software statement.
- Initial Access Tokens are vendor-specific tokens provided to the client to authenticate the client to the application being registered. They are optional unless intermediary identities are being presented.

For a full listing of generally applicable security control considerations, please see Appendix A.

Endpoint Registration Controls



Conclusion

It is our hope that this overview of several common open banking use cases, along with the control points and control considerations gives implementers and security professionals some guidance and validation of best practices.

If you or your organization has been considering joining the Financial Data exchange, or are already a member and would like to contribute more directly, please reach out to us at fdxsupport@financialdataexchange.org.

Appendix A – Control Considerations and Control Points

FDX has given thought to the following controls points that are core to the open banking / open finance / open data ecosystem:

- Perimeter security controls/measures such as web application firewalls (WAF) or API gateway or next-gen firewall (NGFW): – these controls are responsible for bot detection, ensuring IP reputation, device posture and authentication of device, and micro segmentation. In addition to these preventive characteristics, these perimeter security measures are responsible for logging, monitoring, and alerting of the network traffic to and from the ecosystem.
- Internal security controls: an important aspect of zero trust is that there is a control plane between internal components and services. Service meshes such as Istio or Linkered are a popular way to implement service to service (East-West) authorization.
- Databases are a critical part of the ecosystem as they store data at rest. Data must always be stored in a secure manner such that only authenticated and authorized users can access and/or edit it.

Universal Controls

Control points	Technical controls	Implementation standards / methods
WAAP (Web Application and API Protection)	WAF: • Negative security model / • Positive security model • Source reputation scoring • Security event logging • L7 Denial of Service attack prevention • Brute-force and leaked-credential attack protection. • Logging and SIEM/SOAR integration	• NIST SP 800-44 Guidelines on Securing Public Web Servers • OWASP Top 10 Web App
	Advanced API Security: • Adherence to the FDX API OpenAPI spec • Discovery of shadow APIs	• NIST SP 800-95, Guide to Secure Web Services • OWASP Top 10 API Sec
	Bot identification and management	• OWASP Automated Threats
	Denial of Service	• NIST SP 800-53 SC-5: Denial Of Service Protection
API GW	• Authentication and authorization • Quota management	• NIST SP 800-204, Security Strategies for Microservices-based Application Systems

(NG)FW	• Layer 3-4 Denial of Service attack prevention • Prevention of port scanning	• NIST SP 800-41 Guidelines on Firewalls and Firewall Policy
Service Mesh	• IDS/IPS • Service Identities • Service to Service Authz • Inside flow analysis / response analytics / Anomaly detection	• ISO / IEC 27039
Databases	• Privacy protection for data at-rest	• NIST SP 800-53 SC-28 Protection Of Information At Rest • ISO29100 and ISO29134
Client-side protection	• Fraud detection	• End user device malware detection