ONC TEFCA RECOGNIZED COORDINATING ENTITY

TEFCA Facilitated FHIR Implementation Guide

DRAFT 2

Dec 2022

DRAFT

© 2022 The Sequoia Project

Contents

DRAFT

© 2022 The Sequoia Project

1 Introduction

This implementation guide (IG) outlines policy, technical, and process requirements for Qualified Health Information Networks (QHINs), Participants, and Subparticipants to exchange TEFCA Information (TI) using the HL7® FHIR® standard, under the terms of the Common Agreement for Nationwide Health Information Interoperability (Common Agreement) and associated standard operating procedures (SOPs). The requirements outlined in this guide enable QHINs, Participants, and Subparticipants to exchange TI without relying on a QHIN to broker the retrieval of data; QHINs, Participants, and Subparticipants connect directly to query for and provide TI in the FHIR format. The guide supports both business-to-business (B2B) and consumer-facing individual access workflows, and relies on the HL7 Security for Scalable Registration, Authentication, and Authorization FHIR Implementation Guide v1.0 (hereafter referred to as the HL7 Security IG).

Capitalized terms are used throughout this implementation guide. These terms have the same definition as terms defined in the QHIN Technical Framework (QTF) and/or Common Agreement. Additional terms specific to the TEFCA Facilitated FHIR IG are defined here:

- Operator – Any Actor that controls a client or server used to exchange TI in FHIR format. The Operator is identified in the x.509 certificate associated with the client or server.

2 Role Requirements

2.1 FHIR Query Initiator

An Actor with the declared role of a FHIR Query Initiator initiates queries to retrieve information held by an Actor in the Responding Actor role. An Actor with the declared role of a FHIR Query Initiator shall support the technical requirements throughout this IG that are specifically described as applying to the FHIR Query Initiator role. The FHIR Query Initiator operates one or more client applications. This Actor generally aligns with Query Source in the QTF.

2.2 Responding Actor

An Actor with the declared role of a Responding Actor provides information in response to queries by an Actor in the FHIR Query Initiator role. An Actor with the declared role of a Responding Actor shall support the technical requirements throughout this IG that are specifically described as applying to the Responding Actor role. The Responding Actor is associated with one or more FHIR servers. Each FHIR server is associated with an

ONC TEFCA RECOGNIZED COORDINATING ENTITY
© 2022 The Sequoia Project

Authorization Server. The Responding Actor may be the data holder or may be another TEFCA Actor acting as a FHIR Gateway.

3 General Requirements

3.1 PROVENANCE

Responding Actors SHALL use the Provenance Resource[1] to define the source of the data and as a record of any transformations to convert the data to or from FHIR Resources as per Use of Provenance to record Import and Transform. The US Core v4.0 Provenance profile SHALL be included in any response where data has been transformed.
The following bullets list the constraints for the FHIR Provenance resource used with a Facilitated FHIR exchange:

- Provenance.target SHALL be references to all FHIR resources extracted from the document referenced by the Provenance.entity element.
- Provenance.policy SHALL contain the static URI "`urn:oid:2.16.840.1.113883.3.7204.1.5.1.2`".
- Provenance.agent SHALL contain at least one entry [1..*] describing the system that extracted the elements from the document.
    - Provenance.agent.type SHALL contain code "`assembler`" from the code system `http://hl7.org/fhir/ValueSet/provenance-agent-type`
    - Provenance.agent.who SHOULD be a Device resource identifying the system that extracted the data. If no Device resource exists, the Organization resource who conducted the extraction SHALL be used.
- Provenance.entity SHALL contain one element [1..1] describing the source document the information was extracted.
    - Provenance.entity.role SHALL be the code "`source`" from the `http://hl7.org/fhir/provenance-entity-role` codesystem
    - Provenance.entity.what SHALL be a reference to the DocumentReference resource pointing to the original document.

3.2 PATIENT MATCHING

All Actors MUST support the FHIR $match operation to allow for full responses to Patient Discovery queries from FHIR Query Initiators. Responding Actors SHOULD have the capability to return more than one potential patient match when a patient search yields more than one match. Responding Actors SHALL NOT return more than one potential match when such action could be a violation of HIPAA or other Applicable Law, as with the Individual Access Services (IAS) Exchange Purpose, for example. When FHIR Query Initiators request

[1] For information on the Provenance resource, see: https://www.hl7.org/fhir/provenance.html

© 2022 The Sequoia Project

only "certain" matches of operation $match (i.e., require “onlyCertainMatches”=true), Responding Actors SHALL honor that request by returning only a unique match. Responding Actors SHALL NOT return more than 100 potential matches when onlyCertainMatches is set to false.
All FHIR Query Initiators MUST include all known demographics supported in its $match query for patient discovery with the exception of a Social Security Number, which MAY be included.
FHIR Query Initiators SHOULD normalize all patient demographic data elements according to USCDI standards before attempting to query. The lone exception to this rule is for address, which MUST conform to the Project US@ Technical Specification[2]. Queries for information MUST include all demographic parameters that are available and can be sent and are not constrained by applicable law. Demographics SHOULD follow, at a minimum, USCDI v1 defined demographics.[3] Responding Actors SHALL NOT require more than all USCDI v1 demographics plus administrative gender before returning a patient list response.

3.3 Error Responses

Errors resulting from FHIR transactions SHOULD use the OperationOutcome resource to return both human readable and machine processable information with sufficient detail to allow the client to determine if the error can be corrected at the client side, such as via a retry operation due to the resource being busy, or a fatal error. Actors MAY choose to obscure some of these details for security reasons. Any such choices SHOULD be linked to identified security concerns.

3.4 Version Compatibility

Actors SHALL continue to support any capabilities previously supported for TEFCA purposes under a particular FHIR Release, until support for that FHIR Release has been officially sunsetted by the RCE. The RCE will provide advance notice of such sunsetting and will collaborate with the TEFCA community to develop reasonable timelines for such sunsetting.

3.5 Access Token Lifetime

Authorization Servers SHALL issue access tokens with a lifetime up to 60 minutes depending on organizational policy. An Authorization Server MAY also issue a refresh token to an application using the Authorization Code Grant type. If the Authorization Server issues a refresh token to an application that has requested and has been authorized to use the

[2] Project US@ Technical Specification for Patient Addresses, Domestic and Military, available at: https://oncprojectracking.healthit.gov/wiki/download/attachments/180486153/Project%20US@%20FINAL%20Technical%20Specification%20Version%201.0.pdf

[3] *United States Core Data for Interoperability* (USCDI v1 Summary of Data Classes and Data Elements) - available at: https://www.healthit.gov/isa/united-states-core-data-interoperability-uscdi

© 2022 The Sequoia Project

"offline_access" scope, the refresh token lifetime SHALL be no less than three months unless a shorter lifetime aligns with applicable institutional policies.

3.6 Endpoint Lifetime

If an Actor updates its endpoints listed in the QHIN/RCE Directory for any reason other than FHIR Release support, the Actor MUST continue to support transactions received at its previously listed endpoint(s) for a minimum of 48 hours after the QHIN has submitted the new endpoints in the RCE Directory.

4 Use Cases/Workflows

4.1 Patient Discovery, Authentication, & Authorization

4.1.1 Assumptions:

- The FHIR Query Initiator knows sufficient patient demographics for a successful match as determined by the Responding Actor.
- Each Actor has the appropriate service endpoint(s) and other connectivity information for any other Actors above or below it in the hierarchy with which it connects directly.
- The RCE Directory includes the organization facility name(s) FHIR endpoints and HomeCommunityID(s) for all current QHINs, Participants, and Subparticipants. Each Participant and Subparticipant is matched to the appropriate QHIN.
- Each QHIN maintains an up-to-date copy of the RCE Directory.
- Responding QHINs know the current FHIR endpoints and HomeCommunityIDs for any Responding Actors.
- Each QHIN has either a Record Locator Service (RLS) OR Enterprise Master Patient Index (eMPI) OR uses other techniques to perform patient lookup within the Service Level Requirements timeout limitation as specified in the QHIN Service Level Requirements Policy.

4.1.2 Nominal Flow:

1. The FHIR Query Initiator initiates the Patient Discovery workflow as described in the QTF and receives a list of Responding Actors that returned possible patient matches.
    a. Each response returns either a IHE Document Repository or a FHIR endpoint. The FHIR endpoint is not specific to the patient context for the queried patient.
    b. Note: This step is not required in all cases, for example if the FHIR Query Initiator already knows that a Responding Actor holds data about the patient from a previous query to the QHIN Directory.
2. The FHIR Query Initiator discovers the endpoints associated with each Responding Actor it wants to transact with using FHIR.

© 2022 The Sequoia Project

   a. This IG does not define the mechanism a FHIR Query Initiator uses to discover the endpoints. A QHIN, for example, might provide direct access to its QHIN Directory. However, queries for Patient Discovery SHALL be done via the QHIN Directory.
   b. Note: This step is not required in all cases, for example if the FHIR Query Initiator already knows the endpoints associated with a Responding Actor from a previous query to the QHIN Directory.
3. If the FHIR Query Initiator does not have a valid client_id for use with a Responding Actor, the FHIR Query Initiator registers a client with the Responding Actor's Authorization Server:
   a. The FHIR Query Initiator constructs a software statement as specified in the HL7 Security IG and section 5.2.3 of this IG. The software statement includes the FHIR Query Initiator's TEFCA certificate and other metadata.
   b. The FHIR Query Initiator submits a registration request to the registration endpoint of the Responding Actor's Authorization Server. The registration request includes the TEFCA Basic App Certification object. The TEFCA Basic App Certification object includes the B2B Authorization Extension Object. The value in the purpose_of_use field of the B2B Authorization Extension Object matches one of the Exchange Purposes supported by the Authorization Server.
   c. The Responding Actor's Authorization Server processes the FHIR Query Initiator's registration request and returns a client_id.
4. The FHIR Query Initiator requests an access token following the Oauth 2.0 Authorization Code Grant flow (per Section 5.2.4) or Client Credentials flow (per Section 5.2.5). Business-to-business (B2B) applications can use either the Authorization Code Grant flow or Client Credentials flow. Consumer-facing applications must use the Authorization Code Grant flow.
   a. In the Authorization Code Grant flow, the FHIR Query Initiator first obtains an authorization code
      i. The FHIR Query Initiator uses the authorization code to request an access token. When requesting an access token, the FHIR Query Initiator includes an authentication token, signed with their private key.
         1. If the FHIR Query Initiator is operating on behalf of a patient or a patient's representative for the purpose of Individual Access Services (IAS), the FHIR Query Initiator includes the TEFCA User Authorization Extension Object.
            a. The Responding Actor attempts to match the demographic information provided in the TEFCA User Authorization Extension Object with demographic information as known by the Responding Actor. One of the following occurs:
               i. The Responding Actor is confident a match was found and accepts evidence of identity proofing. The Responding Actor issues an access token.

© 2022 The Sequoia Project

ii. The Responding Actor is not confident a match was found or does not accept evidence of identity proofing. The Responding Actor denies the access request. The workflow ends.

b. In the Client Credentials flow, the FHIR Query Initiator requests an access token. The FHIR Query Initiator includes an authentication token in the request, signed with their private key.
   i. The authentication token includes the B2B Authorization Extension Object.
   ii. The Responding Actor validates the request and responds as per Section 5.2.2.2 of the HL7 Security IG.

5. The FHIR Query Initiator has obtained an access token and can begin the information query flow described in Section 4.2.
6. An audit entry of the request is made by both the Responding Actor and FHIR Query Initiator.

4.1.3 Alternate flow 1: The FHIR Query Initiator includes Access Consent

1. The FHIR Query Initiator includes one or more Access Consent Policy (ACP) identifiers in the Authorization Extension Object included in the authentication token as part of their access token request.
   a. The consent_policy element of the Authorization Extension object includes the appropriate OID from QTF-096 of the QTF.
   b. The consent_reference element of the Authorization Extensions object includes one or more URLs of consent objects (FHIR Consent resource or FHIR DocumentReference resource) associated with the OID in the consent_policy element.
   c. One of the two following results occurs:
      i. The (i)ACP is accepted, and the access token returned.
      ii. The (i)ACP is not accepted and, as part of the response, a URL to a DocumentReference of the policy or form required is returned, indicating further or different requirements.
2. Nominal Flow resumes at step 5.

4.1.4 Post Conditions:

- The FHIR Query Initiator has a valid access token for an information query.

4.2 INFORMATION QUERY

4.2.1 Assumptions:

- The FHIR Query Initiator has fulfilled the requirements of Section 4.1 and has a valid access token.

© 2022 The Sequoia Project

4.2.2 Nominal Flow:

1. The FHIR Query Initiator obtains the CapabilityStatement for the FHIR server associated with the Responding Actor. The CapabilityStatement enables the FHIR Query Initiator to evaluate whether the FHIR server supports the FHIR resources, profiles, IGs, operations, etc. sought by the FHIR Query Initiator.
    a. Note: this IG does not define a mechanism for the FHIR Query Initiator to negotiate with a Responding Actor if the FHIR server does not support particular FHIR resources, profiles, IGs, operations, etc.
2. The FHIR Query Initiator executes a patient search using the $match operation as per Section 5.2.6. The FHIR Query Initiator includes all known demographics in the match request.
3. The FHIR Query Initiator queries for information about any patients that were sufficiently matched, limited by the scopes granted by the Responding Actor's Authorization Server.
4. The Responding Actor returns any FHIR resources matching the FHIR Query Initiator's request.
5. The FHIR Query Initiator and Responding Actor create audit log entries of the request(s)FHIR Query Initiator.

4.2.3 Post Conditions:

- The FHIR Query Initiator has obtained requested FHIR resources from the Responding Actor.

5 Infrastructure

5.1 FHIR Endpoints & Endpoint Discovery

Responding Actors SHALL use the FHIR CapabilityStatement resource to define FHIR server capabilities. Implementers SHALL provide at least one publicly discoverable CapabilityStatement where CapabilityStatement.kind="instance". Implementers SHALL provide a CapabilityStatement for each endpoint associated with a FHIR server, defining the capabilities available at that endpoint. Each endpoint SHALL provide access to at least one FHIR resource. Discovery of Endpoints shall be executed by a query to the QHIN Directory service which will have the FHIR endpoint(s) for the Participant and Subparticipant.
Capabilities listed SHALL include all FHIR Implementation Guides listed within this document that are supported by the Responding Actor, in addition to any other capabilities specific to that system.
This CapabilityStatement will only be made available by the server and will not be copied into the RCE/QHIN Directory. This approach will minimize redundant data and associated maintenance, thus reducing out-of-date/sync capability statements while reducing the number of centralized points to establish a connection that could fail. Further

© 2022 The Sequoia Project

implementation experience MAY yield adding other data to the RCE Directory, but that will be addressed later based on implementation feedback.

Link to Directory IG: TBD

5.2 AUTHENTICATION/TRUST

X.509 certificates establish the authenticity of Actors implementing this IG. The RCE will issue one "Intermediate" certificate to each QHIN to seed the QHIN's Certificate Authority. A QHIN's Certificate Authority issues certificates to downstream Actors. Certificates used by any Actor SHALL be chained to the RCE "root" certificate.
A "client secret" SHALL NOT be utilized for authentication. An x.509 certificate chained to the RCE certificate sufficiently enables an Authorization Server to assess whether a client application is trusted.

5.2.1 TEFCA certificates

5.2.1.1 Introduction

Actors SHALL conform to the certificate policy and profile requirements described in the Technical Trust Requirements, with the additions and exceptions noted below.

5.2.1.2 Issuance

QHINs SHALL issue certificates to each Operator of a FHIR client or server. If multiple instances of a client or server are maintained by different Operators, then each Operator SHALL be issued a separate certificate. An Operator MAY group various client and/or server functions together as a single application with a single certificate, or divide them into separate applications with separate certificates, subject to the restrictions below. Depending on organization policy, certificates issued to a single Operator MAY be issued on a per-organization basis (e.g., when one Operator secures the same application on behalf of multiple organizations), or MAY be issued more granularly on a per-application basis (e.g., when one Operator wishes to use separate certificates for software components that run on different servers or perform different functions).
Grouped software components that share a single certificate will be treated as one application by Authorization Servers and MUST be able to use a single client_id assigned by an Authorization Server. If using a single client_id is not practicable, Operators SHOULD use separate certificates for each component.
X.509 certificates used for transactions within a QHIN's network environment SHALL NOT be used for authentication for FHIR transactions between Participants and Subparticipants across QHIN networks.

5.2.1.3 Structure

The value of the Common Name attribute of the Subject Distinguished Name (SDN) SHALL be a human readable name for the Application as provided by the Operator. The Operator's legal business name, city, and state SHALL be included in the SDN of the certificate as the values of the Organization, Locality, and State attributes.

© 2022 The Sequoia Project

The Operator and the FHIR Application SHALL be jointly identified by a unique URI listed in a uniformResourceIdentifier entry in a certificate's Subject Alternative Name (SAN) extension, i.e., each certificate is issued in the context of an Operator and an Application. If a certificate is used to secure an https service endpoint, then the host's DNS name SHALL also be included in the SAN extension as a dnsName entry.
The subject key SHALL be an RSA key (min 2048 bit, either RSA256 or RSA384) or an elliptic curve key (ES256 or ES384).
Client authentication for the workflows described in this guide is achieved using tokens that are digitally signed by the client as described in Section 5.2.3. Thus, a server SHALL NOT additionally require client authentication at the time of the TLS handshake to access the OAuth 2.0 and FHIR endpoints identified for these workflows.

5.2.2 JSON Web Token (JWT) Requirements

The workflows described by this guide use the JWT and JSON Web Signature (JWS) specifications to create digitally signed JWTs that establish the authenticity of Actors. All JWTs MUST be constructed in accordance with the requirements of Section 1.2 of the HL7 Security IG.

5.2.3 Client Registration

Dynamic client registration provides a scalable and automated process for establishing details about client applications and their Operators. Implementers SHALL support dynamic client registration as described in this section. A FHIR Query Initiator SHALL register its client application(s) with the Responding Actor's Authorization Server before querying for information.
This version of the TEFCA Facilitated FHIR IG defines workflows for two types of clients:

1. Confidential clients: Conventional server-based web applications that can maintain a secret.
2. Backend services: for business-to-business connections, backend services can access data directly without user intervention.

Public clients, which do not have a private key, are not currently permitted. Future versions of the IG may allow for this use case.

5.2.3.1 Registration API

5.2.3.1.1 Discovery

Implementers SHALL provide discoverable metadata for each Authorization Server at the {baseURL}/.well-known/udap URL, including the server's authorization, token, and registration endpoints as per Section 2 of HL7 Security IG. This URL MUST be accessible to client applications without requiring client authentication.

5.2.3.1.2 Required Authorization Server Metadata

The metadata returned from the {baseURL}/.well-known/udap metadata endpoint defined above SHALL conform to the requirements listed in the table below and SHALL represent the server's capabilities for the workflows described in this guide. For elements that are

© 2022 The Sequoia Project

represented by arrays, returning an empty array SHALL be interpreted by clients to mean that the corresponding capability is NOT supported by the server.

Table 1: Required Authorization Server Metadata

Element	Optionality	Requirement
udap_versions_supported	required	A fixed array with one string element: ["1"]
udap_profiles_supported	required	An array of two or more strings identifying the core UDAP profiles supported by the Authorization Server. The array SHALL include: "udap_dcr" for UDAP Dynamic Client Registration, and "udap_authn" for UDAP JWT-Based Client Authentication. If the grant_types_supported parameter includes the string "client_credentials", then the array SHALL also include: "udap_authz" for UDAP Client Authorization Grants using JSON Web Tokens to indicate support for Authorization Extension Objects.
udap_authorization_extensions_supported	required	An array of zero or more recognized key names for Authorization Extension Objects supported by the Authorization Server. If the Authorization Server supports the B2B Authorization Extension Object defined in Section 5.2.1.1, then the following key name **SHALL** be included: ["hl7-b2b"]

© 2022 The Sequoia Project

Element	Optionality	Requirement
udap_authorization_extensions_required	conditional	An array of zero or more recognized key names for Authorization Extension Objects required by the Authorization Server in every token request. This metadata parameter **SHALL** be present if the value of the udap_authorization_extensions_supported parameter is not an empty array. If the Authorization Server requires the B2B Authorization Extension Object defined in Section 5.2.1.1 in every token request, then the following key name **SHALL** be included: ["hl7-b2b"]
udap_certifications_supported	required	An array of one or more certification URIs supported by the Authorization Server, with a minimum of the Basic App Certification ["https://rce.sequoiaproject.org/udap/profiles/basic-app-certification"]
udap_certifications_required	conditional	An array of one or more certification URIs required by the Authorization Server. This metadata parameter **SHALL** be present and list, at a minimum the Basic App Certification: ["https://rce.sequoiaproject.org/udap/profiles/basic-app-certification"]
grant_types_supported	required	An array of one or more grant types supported by the Authorization Server, e.g.: ["authorization_code", "refresh_token", "client_credentials"] The "refresh_token" grant type **SHALL** only be included if the "authorization_code" grant type is also included.

© 2022 The Sequoia Project

Element	Optionality	Requirement
scopes_supported	recommended	An array of one or more strings containing scopes supported by the Authorization Server. The server **MAY** support different subsets of these scopes for different client types or entities. Example for a server that also supports SMART App Launch v1 scopes: ["openid", "launch/patient", "system/Patient.read", "system/AllergyIntolerance.read", "system/Procedures.read"]
authorization_endpoint	conditional	A string containing the absolute URL of the Authorization Server's authorization endpoint. This parameter **SHALL** be present if the value of the grant_types_supported parameter includes the string "authorization_code"
token_endpoint	required	A string containing the absolute URL of the Authorization Server's token endpoint for UDAP JWT-Based Client Authentication.
token_endpoint_auth_methods_supported	required	Fixed array with one value: ["private_key_jwt"]
token_endpoint_auth_signing_alg_values_supported	required	Array of strings identifying one or more signature algorithms supported by the Authorization Server for validation of signed JWTs submitted to the token endpoint for client authentication. All implementations SHALL support RS256, SHOULD support ES256, and MAY support ES384 and RS384. For example: ["RS256", "ES384"]

© 2022 The Sequoia Project

Element	Optionality	Requirement
registration_endpoint	required	A string containing the absolute URL of the Authorization Server's registration endpoint.
registration_endpoint_jwt_signing_alg_values_supported	recommended	Array of strings listing one or more JWA algorithm identifiers supported by the Authorization Server for validation of signed software statements, certification, and endorsements submitted to the registration endpoint. All implementations SHALL support RS256, SHOULD support ES256, ES384 and RS384. For example: ["RS256", "ES384"]
signed_metadata	required	A string containing a JWT listing the server's endpoints, as defined in [Section 2.3] below.

5.2.3.1.3 Software Statement

To register dynamically, a client application first constructs a software statement. The software statement is a JWT signed by the client using the private key that corresponds to the public key listed in its X.509 certificate. The JOSE header and payload of the JWT are constructed as per section 3.1 of the HL7 Security IG FHIR IG. The JOSE Header SHALL contain the required elements specified in Section 5.2.2of this guide. The client determines which of the signature algorithms are supported in common by the client and the server, and then signs the software statement using one of the RS256, ES256, RS384, or ES384 signature algorithms as defined in RFC 7518; the client SHOULD chose the strongest mutually supported signature algorithm. All implementations SHALL support RS256, and SHOULD support ES256, ES384 and RS384. All X.509 certificates SHALL have RS256 keys and SHOULD have ES256, ES384 and RS384 keys.

The unique client URI used for the 'iss' claim SHALL match the uriName entry in the Subject Alternative Name extension of the client application's X.509 certificate. The software statement is intended for one-time use with a single Authorization Server. As such, the 'aud' claim SHALL list the URL of the Authorization Server's registration endpoint, and the lifetime of the software statement ('exp' minus 'iat') SHALL be 5 minutes.

© 2022 The Sequoia Project

5.2.3.1.4 Inclusion Of Certifications and Endorsements

This IG supports the certifications framework outlined in Section 3.3 of the HL7 Security IG. Authorization Servers SHALL support the inclusion of certifications and TEFCA-Defined Extensions. Authorization Servers SHALL ignore unsupported or unrecognized certifications, i.e., the inclusion of an unsupported or unrecognized certification SHALL NOT be a reason for an Authorization Server to return an error response.

This guide defines the TEFCA Basic App Certification Profile in Section 5.2.3.2. Client applications SHALL include the TEFCA Basic App Certification object in their registration request.

The following is a non-normative example software statement, prior to Base64URL encoding and signature.

```
{
 "alg": "RS256",
 "x5c": ["MIEE8DCCA … remainder omitted for brevity"]
}
{
 "iss": "http://example.com/my-application",
 "sub": "http://example.com/my-application",
 "aud": "https://as.example.net/register",
 "exp": 1525209377,
 "iat": 1525209077,
 "jti": "random-jti-generated-by-client"
 "client_name": "My Application",
 "redirect_uris": ["https://example.com/redirect"],
 "contacts": ["mailto:tefcacontact@example.com"],
 "logo_uri": "https://www.example.com/HealthApp.png",
 "grant_types": ["authorization_code"],
 "response_types": ["code"],
 "token_endpoint_auth_method": "private_key_jwt",
 "scope": "user/Patient.read", "user/Procedure.read"
}
```

5.2.3.1.5 Request Body

The registration request is submitted by the client to the Authorization Server's registration endpoint.

```
POST /register HTTP/1.1
Host: as.example.net
Content-Type: application/json
{
 "software_statement": "...the signed software statement
JWT...",
 "certifications": ["...a signed certification JWT…"]
 "udap": "1"
}
```


© 2022 The Sequoia Project

The Authorization Server validates and processes the registration request as per Section 3.2.3 of the HL7 Security IG. This includes validation of the JWT payload and signature, X.509 certificate chain, and registration parameters. If the registration is successful, the Authorization Server SHALL return an HTTP 201 response which includes the unique client_id that the client app will use to interact with the Authorization Server's authorization and token endpoints and the registration metadata that was accepted. The Authorization Server SHALL grant registration requests received from clients submitting software statements signed with valid TEFCA certificates assuming all authorization logic and purpose of use requirements have been reviewed and accepted.

5.2.3.2 TEFCA Basic App Certification Profile

In most cases, the Application Operator will provide additional authorization metadata to the data holder's Authorization Server at the time of the access token request using the TEFCA Authorization Extension Object, as discussed in Sections 5.2.4 and 5.2.5. When some or all this metadata will not vary across subsequent token requests, e.g., all requests will be for a single purpose of use, then the Application Operator MAY also provide some or all this authorization metadata to the data holder's Authorization Server at the time of registration. This is accomplished by including a TEFCA Basic App Certification JWT in the certifications array of the registration request. This self-asserted certification re-uses the TEFCA Authorization Extension Object keys defined in Section 5.2.5, with the additional requirements described throughout this section. The certification JWT included by the client app MUST conform to the header requirements in of Section 1.2 of the HL7 Security IG and the claims requirements in the following table:

Table 2 TEFCA Basic App Certification JWT Claims

Element	Optionality	Requirement
iss	required	Issuer of the JWT -- unique identifying client URI. This must match the value of a uniformResourceIdentifier entry in the Subject Alternative Name extension of the client's certificate included in the 'x5c' JWT header and MUST match the value of the 'iss' claim of the software statement with which this certification is submitted See https://tools.ietf.org/html/rfc5280#section-4.2.1.6
sub	required	Same as 'iss'

© 2022 The Sequoia Project

Element	Optionality	Requirement
exp	required	Expiration time integer for this self-assertion, expressed in seconds since the "Epoch" (1970-01-01T00:00:00Z UTC). Since this certification may be reused for multiple registrations, the expiration time SHALL be no more than three years after the time the JWT is issued and SHALL predate the certificate expiration date
iat	required	Issued time integer for this authentication JWT, expressed in seconds since the “Epoch”
jti	required	A nonce string value that uniquely identifies this certification JWT. This value SHALL NOT be reused by the client app in another JWT with the same ‘iss’ value before the time specified in the ‘exp’ claim has passed
certification_name	required	String with fixed value: "TEFCA Basic App Certification"
certification_uris	required	Fixed array with single string element: ["https://rce.sequoiaproject.org/udap/profiles/basic-app-certification"]
extensions	required	A JSON Object containing the key “hl7-b2b” with a value equal to a B2B Authorization Extension Object, as defined in Section 5.2.5, with the additional requirements discussed below.

For the purposes of this certification profile, inclusion of the ‘version’ element of the B2B Authorization Extension Object defined in Section 5.2.5 SHALL be required. However, inclusion of all other extension object elements is OPTIONAL for this certification profile. Specifically, the Operator MAY include only those extension object keys whose values will not vary across subsequent access token requests. For example, if all subsequent access token requests by the Application will be made by one organization and only for the purpose of treatment, then the ‘organization_id’, ‘organization’, and ‘purpose_of_use’ keys and the corresponding values could be included in the B2B Authorization Extension Object within the TEFCA Basic App Certification. Elements whose value may vary in subsequent access token requests SHALL NOT be included in this certification.

© 2022 The Sequoia Project

An Operator SHOULD NOT submit a TEFCA Basic App Certification with the Authorization Extension Object containing only the 'version' element, as such a certification provides no additional information to the data holder's Authorization Server. When an Operator includes a TEFCA Basic App Certification in its registration request, an Authorization Server MAY reject subsequent access token requests by this app that contain authorization metadata that does not match the corresponding values declared in the certification. In addition, an Authorization Server MAY require that one or more elements of the TEFCA Basic App Certification, such as purpose_of_use, be supplied at registration time. Authorization servers that reject a registration request due to a missing element SHOULD respond with an informative error identifying that element.

5.2.3.3 Modifying Registrations

The client URI in the Subject Alternative Name of an X.509 certificate uniquely identifies a single application and its operator over time. A previously registered client application MAY request a modification of its previous registration with an Authorization Server by submitting another registration request to the same Authorization Server's registration endpoint using a certificate with a Subject Alternative Name client URI entry matching the original registration request. Note that this may be a different certificate than the one used in the previous registration, such as in the case of certificate renewals or re-keyings. The registration request SHALL include all parameters required for a new registration and all parameters for which modification is requested. Such requests SHALL be processed as described in Section 3.4 of the HL7/UDAP Security for Scalable Registration, Authentication, and Authorization FHIR IG.

If the Authorization Server returns a different client_id in the registration response, the client application SHALL use only the new client_id in subsequent transactions with the Authorization Server. The responding Authorization Server MUST disable the old client_id so that it cannot be used for subsequent requests. However, the retired client_id SHOULD be preserved by the Authorization Server so that it can be associated with log entries and the requester.

5.2.4 Authorization Code Grant Type (3-legged OAuth 2.0)

Applications that wish to exchange data with a FHIR server first authenticate and authorize themselves and their users with an Authorization Server to obtain an access token. This section outlines the flows for user-facing applications. Only confidential apps are supported in this version of this guide. Clients and servers MAY optionally support UDAP Tiered OAuth for User Authentication.

5.2.4.1 Obtaining an Authorization Code

The client application SHALL obtain an authorization code as per section 2.0.9 of the HL7 SMART App Launch IG v2.

5.2.4.2 Obtaining an Access Token

Confidential clients will follow the procedure defined in Section 4 of the HL7 Security IG.

The responder typically authenticates the end user during the authorization step when using the authorization_code grant type. If the organization operating the requesting application

© 2022 The Sequoia Project

has additionally identity-proofed the end user of its application, then the requesting application MUST provide metadata about the user to the data holder as additional authorization information at the time of the token request by adding this information to the authentication JWT in the form of the TEFCA-specific authorization extension as per Section 4.2.1 of the FHIR UDAP Security IG, and as detailed further below.
The user metadata submitted by the requesting application in the extension object SHALL correspond to the verified identity attributes of the permitted user (verified as per Section 3.2) who is making the request. Note that for patient requests (i.e., where the purpose of use code is "REQUEST") where this user is not necessarily the patient who is the transaction subject. In this case, the verified user MAY instead be a patient's authorized representative, or the verified user MAY be a person or a system at an organization acting on the specific written authorization of the patient. A responder MUST indicate support or lack of support for this extension object by including or omitting the "tefca_user" key from its list of supported authorization extension objects in its UDAP metadata. If a responder does not support this extension object, it MAY ignore the associated metadata. Alternatively, if the responder has explicitly indicated in its UDAP metadata that this extension object is not supported, it MAY instead return an "invalid_request" error response for a token request containing this extension object.
If the responder supports the use of the TEFCA User Authorization Extension object, the responder is responsible for validating that verified user identity metadata submitted by the application reasonably matches the responder's own records for the app user that was authenticated by the responder before issuing an access token. If the data holder cannot validate the user information, it SHALL return an invalid_grant error in response to the token request.
The authentication JWT submitted by the client app MUST conform to the requirements in Section 4.2.1 of the HL7 Security IG. If the "TEFCA_user" authorization extension object is used by the client, the client SHALL additionally include an "extensions" claim in the authentication JWT containing this authorization extension object, as per the UDAP JWT-Based Client Authentication profile.

Table 3 TEFCA User Authorization Extension object

Extension Name: "tefca_user"		
Element	**Optionality**	**Requirement**
version	required	Fixed string value: "1"
purpose_of_use	required	Fixed string value: "REQUEST"
user_information	required	FHIR Patient resource with all known demographics.

© 2022 The Sequoia Project

Extension Name: "tefca_user"		
Element	**Optionality**	**Requirement**
Ial_vetted	required	A comma separated list of demographic attributes provided within the Patient resource as specified in the IAS Exchange Purpose Implementation SOP available here:
consent_policy	required	The Access Consent Policy Identifier corresponding to the asserted Access Policy that represents the identity proofing level of assurance of the user, array of string values from the subset of valid policy OIDs in Appendix B that represent identity proofing levels of assurance, each expressed as a URI, e.g. ["urn:oid:2.16.840.1.113883.3.7204.1.1.1.1.12"]
consent_reference	optional	An array of FHIR DocumentReference or Consent resources where the supporting access consent documentation can be retrieved, each expressed as an absolute URL, e.g. ["https://tefca.example.com/fhir/R4/DocumentReference/consent-70796b65"]

The parameters for the POST request to the Authorization Server's token endpoint MUST conform to the requirements in section 4.2.2 of the HL7 Security IG.
Servers SHALL process and respond to token requests as per Section 4.2.2 of the HL7 Security IG. If the Authorization Server issues a refresh token to an application that has requested and has been authorized to use the "offline_access" scope, the refresh token lifetime SHALL be no less than three months unless a shorter lifetime aligns with applicable institutional policies. If an application that has requested and has been authorized to use the "offline_access" scope presents a valid refresh token to an Authorization Server to obtain a new access token, the Authorization Server SHOULD also issue a new refresh token valid for a new period of no less than three months unless a shorter lifetime aligns with applicable institutional policies.
Non-normative example (white space and line breaks added for clarity, not URL-encoded):
Request:

```
POST /token HTTP/1.1
Host: as.example.com
Content-type: application/x-www-form-urlencoded
grant_type=authorization_code&
 code=authz_code_from_resource_holder&
```


© 2022 The Sequoia Project

```
 client_assertion_type=urn:ietf:params:oauth:client-assertion-
type:jwt-bearer&
 client_assertion=eyJh[…remainder of AnT omitted for brevity…]&
 udap=1
```

Response (success):

```
HTTP/1.1 200 OK
Content-Type: application/json
{
 "access_token": "example_access_token_issued_by_AS",
 "token_type": "Bearer",
 "expires_in": 3600
}
```

5.2.5 Client Credentials Grant Type (2-legged OAuth 2.0)

Privileged applications acting on their own behalf, or without direct user action required via the data holder's authorization endpoint, use the client_credentials grant type. Authorization Servers SHALL NOT issue refresh tokens to clients using this grant type. For this workflow, clients and servers SHALL conform to the requirements for use of client credentials grants in Section 5 of the HL7 Security IG.

Since users do not interact directly with the data holder's authorization endpoint when using the client_credentials grant type, clients SHALL provide additional authorization information to the data holder at the time of the token request by adding this information to the authentication JWT in the form the B2B Authorization Extension Object. The table below defines TEFCA-specific values for the B2B Authorization Extension Object that SHALL be followed in addition to the requirements defined in the HL7 Security IG.

Table 4 TEFCA specific Values for the B2B Authorization Extension object

Extension Key Name: "hl7-b2b"		
Element	**Optionality**	**Requirement**
version	required	Fixed string value: "1"
organization_id	required	String containing the URL of requestor's Organization resource at the RCE Directory server: TBA "https://.../Organization/2.16.840.1.113883.301.560.6999"
Organization_name	required	String containing the requestor's human readable organization name, e.g., "ABC Hospital"

© 2022 The Sequoia Project

Extension Key Name: "hl7-b2b"		
Element	**Optionality**	**Requirement**
subject_id	conditional	String containing the human readable name of the person responsible for originating the request. SHALL be present when applicable, e.g. "Dr. Mary Johnson"
purpose_of_use	required	An array of strings containing the purpose for which the data is requested, from the code set of permitted purposes found in *Table 7: Exchange Purpose Accepted Codes* in QTF-023 of the QHIN Technical Framework
consent_policy	optional	The Access Consent Policy Identifier corresponding to the asserted Access Policy, array of string values from the list of valid OIDs in QTF-096, each expressed as a URI, e.g. ["urn:oid: 2.16.840.1.113883.3.7204.88.1.1.1.2.3"]
consent_reference	optional	An array of FHIR DocumentReference or Consent resources where the supporting access consent documentation can be retrieved, each expressed as an absolute URL, e.g. ["https://tefca.example.com/fhir/R4/DocumentReference/consent-70796b65"]

When the B2B Authorization Extension Object is included in a token request and the data holder determines that the authorization metadata submitted is insufficient for the data holder to grant access because the data holder requires one or more (Instance) Access Consent Policies to be asserted but the requestor has omitted the ACP parameter or has asserted a policy that is not acceptable to the data holder, then the Authorization Server SHALL return an invalid_grant error response to the token request, and this error response SHOULD include the TEFCA Authorization Extension Error object in the 'extensions' object of the error response.

© 2022 The Sequoia Project

Table 5 TEFCA Authorization Extension Error object

Extension Name: "hl7-b2b"		
Element	**Optionality**	**Requirement**
consent_required	required	The list of acceptable Access Consent Policy Identifier(s) corresponding to the asserted Access Policy required for authorization, an array of string values from the list of valid policy OIDs in Appendix A of this IG, each expressed as a URI, e.g. ["urn:oid:2.16.840.1.113883.3.7204.88.1.1.1"]
consent_form	optional	A URL as a string where the required consent form may be downloaded, if applicable, e.g. "https://implementer1.example.com/consentForms/sample1.pdf"

Responders supporting use cases that require transmission of consent information SHALL support the consent_policy and consent_reference claims and SHALL be able to resolve a DocumentReference or Consent resource included in the consent_reference array.
If the requested purpose of use is not supported by the responder, the responder SHALL return an invalid_grant error response to the requesting application.

5.2.6 Individual Access Services (IAS) Requests

Responders MUST support the Authorization Code Grant type for IAS Requests (i.e., where the purpose_of_use code is REQUEST). This corresponds to authorization workflow 2. The responder SHALL support this workflow and the responder SHALL support the TEFCA User Authorization Extension object, defined in Section 5.2.4.2 and identified by the extension key "tefca_user".
A client application requesting a token for Patient Requests using the client credentials grant type SHALL include the TEFCA User Authorization Extension Object in its token request instead of the hl7-b2b Authorization Extension Object. The user metadata submitted by the requesting application in the TEFCA User extension object SHALL correspond to the verified identity attributes of the permitted user (verified to IAL2 as per the Common Agreement and IAS Exchange Purpose Implementation SOP) who is making the request. Note that this user is not necessarily the patient who is the transaction subject, i.e., the verified user MAY instead be a patient's authorized representative. Before issuing an access token, the responder SHALL validate that the verified user identity metadata submitted by the application matches the responder's own records for a person that is authorized to make patient requests in accordance with the Common Agreement and SHALL limit the patient data accessible using the access token accordingly. If the submitted user information does not sufficiently match a person known to the responder, or if the responder does not support

© 2022 The Sequoia Project

this workflow for Patient Requests, it SHALL return an invalid_grant error in response to the token request.
Example:
Below is a non-normative example of complete authentication JWT header and claims with authorization information prior to Base64URL-encoding and signing:

```
{
 "alg": "RS256",
 "x5c": ["MIIEczCCA1ugA…remainder of Base64 encoded certificate
omitted for brevity…"]
}
{
 "iss": "http://implementer1.example.com/tefca-fhir-app",
 "sub": "myClientID",
 "aud": "https://implementer2.example.net/token",
 "exp": 1557843252,
 "iat": 1557843852,
 "jti": "Q1E6g2PY91nmj5bSJJ-CZQ",
 "extensions": {
 "tefca": {
 "version": "1",
 "organization_id":
"https://directory.rce.sequoiaproject.org/Organization/2.16.840.
1.113883.19.347473",
 "organization": "ABC Hospital",
 "subject_id": "Dr. Mary Johnson",
 "purpose_of_use": "TREATMENT",
 "acp": ["urn:oid:2.16.840.1.113883.3.7204.88.1.1.1.5"],
 "acp_reference":
["https://tefca.example.com/fhir/R4/DocumentReference/consent-
70796b65"]
 }
 }
}
```

After generating an authentication JWT, the client requests a new access token via HTTP POST to the FHIR Authorization Server's token endpoint URL as per section 5.2.2.2 of the HL7/UDAP Security for Scalable Registration, Authentication, and Authorization FHIR IG.
Examples (non-normative, white space, and line breaks added for clarity, not URL-encoded):

Request:

```
POST /token HTTP/1.1
Host: as.example.com
Content-type: application/x-www-form-urlencoded
grant_type=client_credentials&
scope=system/Patient.read&
```


© 2022 The Sequoia Project

```
client_assertion_type=urn:ietf:params: :client-assertion-
type:jwt-bearer&
client_assertion=eyJh[…remainder omitted for brevity…]&
udap=1
```

Response (success):

```
HTTP/1.1 200 OK
Content-Type: application/json
{
 "access_token": "example_access_token_issued_by_AS",
 "token_type": "Bearer",
 "expires_in": 3600
}
```

Response (failure, Authorization Server requires a specific Access Consent policy to be asserted):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
 "error": "invalid_grant",
 "error_description": "An Access Consent policy must be asserted
for this purpose of use.", "extensions": {
 "tefca" : { "acp_required":
["urn:oid:2.16.840.1.113883.3.7204.88.1.1.1.1"] }
 }
}
```

5.3 Clinical Data Exchange

5.3.1 FHIR Version and FHIR Implementation Guide Support Requirements

All Actors SHALL support FHIR version 4.0.1 with FHIR US Core Implementation Guide V5.0.1where data is available (e.g., US Core Pediatric BMI for Age Observation Profile need not be supported if the information is not collected). In addition, the following FHIR Implementation Guides SHOULD be supported:

- Bulk Data Access IG v2.0.0
- Mobile access to Health Documents (MHD) v4.1.0
- Da Vinci Payer Data Exchange v2.0.0, when released

Other FHIR Implementation Guides MAY be supported as appropriate. Actors MUST list all IGs above supported in their FHIR server CapabilityStatement and SHOULD list any other supported FHIR IG.

All actors SHOULD support the extensions for version conversion as specified in the FHIR Core section 2.7.0.7 Extensions for converting between versions.

© 2022 The Sequoia Project

FHIR version 4.0.1 and the versions of the above listed FHIR IGs SHALL be supported until officially sunsetted by the RCE in favor of a later version. This sunsetting will allow for a 3-month phase-out of the sunsetted version.

5.3.2 Patient Discovery

Except for the SMART on FHIR auth code flow in which the `launch/patient` SMART scope is requested, granted and the Patient ID is subsequently provided, Patient Discovery SHALL be performed using the FHIR Patient Resource $match operation. Each query SHALL include, but is not limited to, all available USCDI patient demographics with a minimum of (where known): first name, last name, date of birth, birth sex, current address (normalized as per the Project US@ Technical Specification), phone number(s), and email address(es) plus administrative gender. All implementers SHALL support these demographics. A responder MAY ignore any additional demographic elements included that are not supported.
The $match request operation SHALL have the following parameters:

Table 6 Patient $match Required Parameters

Parameter	Required Value
onlyCertainMatches	SHALL be set to true for Individual Access Services Requests; when set to true, the server SHALL return only certain matches; when absent or set to false, the server MAY return probable matches, but is not required to do so if its organizational policy allows only certain matches to be returned
Count	Optional, server MAY send fewer results than specified Note that clients should be careful when using this, as it may prevent probable—and valid—matches from being returned
Resource	Patient resource with included demographic parameters formatted as per Project US@ Technical Specification

© 2022 The Sequoia Project