Implementing a 21 CFR 11-compliant FHIR-based PRO Collection and Storage System with Digital Signing and Journaling of FHIR Resources

Table of Contents

Document History

Contributor	Revision/Date	Summary
Chris Gentle	0.1/February 18, 2020	Initial Draft
Chris Gentle	0.2/March 13, 2020	Add ACME Protocol, Use Cases
Chris Gentle, Marsh Marques, Marc Natter, Helen Waite	0.3/June 24, 2020	Revised acme protocol. Review comments. Added safety section draft.
Chris Gentle	0.4/July 20, 2020	Updating HIPAA information

Introduction

21 CFR Part 11 describes a set of requirements for closed systems intended to ensure that clinical data is collected from authentic sources and stored in a system that protects the integrity of the data in a verifiable way. To collect Patient Reported Outcomes for examination by the FDA, 21 CFR Part 11 requirements are applicable to those systems that collect and analyze those PROs.

The FHIR standard will be increasingly important for FDA studies reliant on PROs, and without guidance implementers may resort to a variety of bespoke or proprietary approaches to meeting 21 CFR Part 11 requirements.

21 CFR Part 11 requires that a system implement measures that protect it from external and internal attacks[1] and provide "the ability to discern invalid or altered records[2]"

If a FHIR-based warehouse is to protect against tampering and make tampering evident to system operations and auditors then it must maintain a log of transactions with digital signatures, and a way of proving that the log itself has not been altered or fabricated.

A process that derives benefit from a 21 CFR Part 11-compliant system is safety and adverse event reporting and analyses. A 21 CFR Part 11 system provides assurances about the integrity of data provided to report and act on safety events.

This document describes two complementary approaches to meeting 21 CFR Part 11 requirements for authenticity, integrity and non-repudiability of captured PRO data:

- FHIR client-generated signatures of verifiably stored client-originated resources
- Server-side implementation of a digitally verified change ledger

These requirements not strictly met with current FHIR guidance and documents a recommended guidance for meeting 21 CFR 11 requirements when using a FHIR server as the repository for PROs.

Glossary

Term	Definition or Reference
PKI	Public Key Infrastructure _

Requirements

These requirements summarize the 21 CFR Part 11 controls for closed systems that are beyond the current scope of the FHIR PRO Implementation Guide[3]. 21CFR Part 11 requires a set of system controls and procedures that are intended to provide protections against both outsider and insider threats and methods of assessing the integrity of the system during operation and audit.

Any system that is collecting PROs under the 21 CFR Part 11 constraints should adhere to these requirements. FHIR-based systems can meet 21 CFR Part 11 requirements by following the recommendations and requirements in a proposed 21 CFR 11 FHIR PRO Implementation Guide which would refer to the existing FHIR PRO Implementation Guide.

Epic 01: Any FHIR client interacting with the FHIR server shall provide a digitally signed Provenance resource for any resource data they create, update or delete.

So that the resource data in the FHIR server can be tested for validity and so that "the signer cannot readily repudiate the signed record as not genuine"[4]

So that record deletion can be traced to an appropriately authorized user.

So that unsigned resources can be easily detected and processed according to their provenance-lacking status.

Epic 02: Creation, modification and deletion of records shall be stored in a tamper-evident or tamper-proof log

So that any resource's provenance can be examined to validate the identity of the client that changed the resource, the time that the change occurred, and that the signature matches the content of that resource version.

So that the log can be tested for tampering.

So that the FDA can be provided FHIR resources and proof of provenance for review and copying of records[5].

The PRO Implementation Guide for FHIR

The existing PRO Implementation Guide[6] (IG) recommends use of the FHIR R4 SDC Questionnaire, QuestionnaireResponse, Adaptive Questionnaire and Adaptive QuestionnaireResponse resources and defines profile for each of these.

The PRO IG includes capability statements and guidance for use of these resources.

The 21CFR11 IG supplements the PRO IG and expands it with implementation guidance around the provision of Provenance resources and system-level requirements that are outside the scope of FHIR standards changes.

Client-Provided Provenance

A FHIR client that provides PRO data to a server must have an identity that can be verified by the server it interacts with and by a 3rd party like the FDA to establish the authenticity of that client's interactions with the server. To allow this we propose that clients participate in a PKI system where they maintain a client secret and establish their authentic identity with a revocable and rotatable public certificate signed by a certificate authority with a method of revocation.

A server can assess the authenticity of a client's provided resources by asking a FHIR client to sign those resources using its private key. The server can then verify and store both the signature and the resource knowing that server authenticated the client and that client verified the integrity of the resource data sent.

An auditor that wishes to assess the veracity of the stored PRO data would be able to take each version of a resource and test whether there is a corresponding client signature for that resource and assess whether it is indeed a resource that was signed by the client.

The figure below describes a FHIR protocol client-server interaction sequence for client-signing of individual resources.

The client POSTs, PATCHes or PUTs a resource, and when that REST operation is complete a copy of the new resource with version and identifier information is returned to the client in the HTTP response. The client converts the resource that is returned in the HTTP response to a canonical form and signs a hash of that "canonicalized" resource for inclusion in the corresponding provenance resource with its private key.

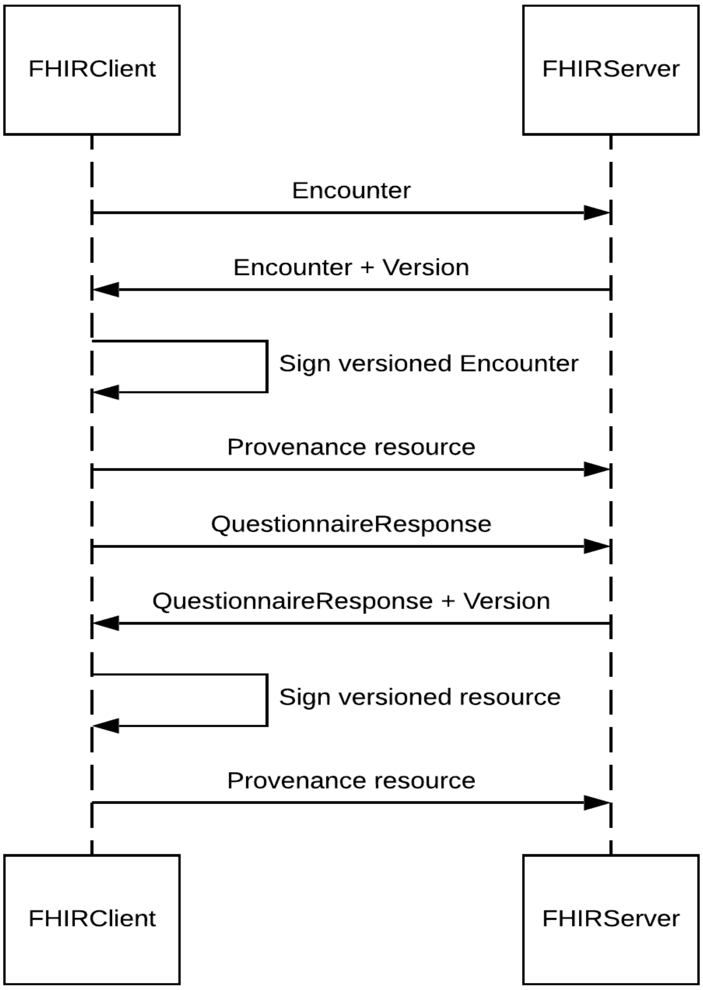


Figure 1 - FHIR Protocol Signing Sequence

This will work to provide verifiable provenance for individual resources, but a system-wide journal is required to detect a range of attacks including deletions.

Server Capabilities and Constraints

A 21CFR11 PRO FHIR server should:

- Support JSON resource formats for all PRO interactions. The digital signature process relies on use of a format that can be canonicalized, and which can isolate server-modified resource information so that signatures can be verified.
- Respond to requests with the Prefer: return=representation header set with the full JSON of the server resource. Without this, the client is unable to verify that the resource data persisted by the server is substantially the same as the resource that was sent by the client.

Populating the Provenance Resource

The main features of a Provenance resource are to establish, for a resource, the following characteristics:

- The time that provenance was established
- The resource instance (and version) that provenance is established for
- Who created or modified the resource instance?
- The signature of the object provenance is captured for

This is not a typical use of the Provenance resource. The usual use case for a signature in Provenance is to capture a reference to a scan of a signature or a hash of a document referenced in another resource. Often the FHIR server is left to generate a Provenance resource immediately after the successful POST or PUT of a non-Provenance resource without a signature. The FHIR standard provides mechanism to include a JSON partially complete Provenance resource in an HTTP header that the server can add when the non-Provenance resource has been successfully persisted[1].

Below is an example of a JSON Provenance resource for a QuestionnaireResponse with an external digital signature for that QuestionnaireResponse. The other keys reference the source Patient's identity and identity certificate to aid later verification.

```
{
  "resourceType": "Provenance",
  "id": "42",
  "text": {
    "status": "generated",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Example provenance resource 42.</div>"
  },
  "target": [
    {
      "reference": "QuestionnaireResponse/42/_history/1"
    }
  ],
  "occurredPeriod": {
    "start": "2019-10-25",
    "end": "2019-10-25"
  },
  "recorded": "2015-06-27T08:39:24+10:00",
  "policy": [
    "http://acme.com/fhir/Consent/25"
  ],
  "location": {
    "reference": "Location/1"
  },
  "reason": [
```

[1] https://www.hl7.org/fhir/provenance.html#header

```
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/v3-ActReason",
          "code": "HCOMPL",
          "display": "Compliance"
        }
      ]
    }
  ],
  "agent": [
    {
      "type": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/v3-ParticipationType",
            "code": "AUT"
          }
        ]
      },
      "who": {
        "identifier": {
          "system": "urn:ietf:rfc:3986",
          "value": "mailto:odm-fhir-gw@chip.org"
        }
      }
    },
    {
      "id": "cert-04121321-4af5-424c-a0e1-ed3aab1c349d",
      "type": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/v3-ParticipationType",
            "code": "DEV"
          }
        ]
      },
      "who": {
        "reference": "Device/04121321-4af5-424c-a0e1-ed3aab1c349d"
      }
    }
  ],
  "signature": {
    "type": [
        {
          "system": "urn:iso-astm:E1762-95:2013",
          "code": "1.2.840.113549.1.1.1",
          "display": "Verification Signature"
        }
      ],
    "when": "2019-10-25T01:39:24+10:00",
    "who": {
        "reference": "Patient/1"
      },
    "targetFormat": "application/fhir+json",
    "sigFormat": "application/signature+json",
    "data": "9139be98f16cf53d22da63cb559bb06a93338da6a344e28a4285c2da33facb7080d26e7a09483779a016eeb
c207602fc3f90492c2f2fb8143f0fe30fd855593d"
  }
}
```

Figure 2 – Example Provenance Resource

The Role of a Ledger

A threat that is not prevented with signing individual resources and individual versions of resources within a system is that a bad actor can remove both those resources and the signatures of those resources. Without a trustworthy external reference to those resources there is no way to assess the veracity of those resources or even know that they existed. A bad actor might remove observations or subjects from a study who do not support a hypothesis, biasing a study in a way that is undetectable by readers or regulators.

One way to solve this is to maintain two copies of data, one read-only in an isolated system, and compare the two systems against system state during audit. Additionally, the read-only version of the system could be a tamper-evident ledger of signed transactions that can be used to assess the integrity of the original data. This works for most threat models, but a compromised server is able to misrepresent the system state and the external copies if it filtered transactions in real time rather than during post-hoc tampering.

Most FHIR security guides and resources are concerned with outsider threats: attackers trying to penetrate the client and the server to access or change data. 21 CFR Part 11 asks systems to defend themselves -standard method of allowing a client access to verify FHIR transactions and the store itself. Assertions of provenance represented by the Provenance resource are assertions made by the server about its view of the provenance of the stored resources. Provenance resources have fairly unique transaction semantics that ask the server to sign resources on the client's behalf.

Another item for discussion is whether clients can verify the resource and the signature themselves. The ability to compare the resource as acknowledged by the server against the client's copy of that resource, the ability to verify that it is appropriately signed, and to test that a transaction is present that persists it are necessary to provide an ability for a client to verify that provenance has been indelibly captured.

A Proposed Ledger Architecture

Using a blockchain or similar ledger to audit the integrity of a system requires all clients of the blockchain to participate in the blockchain's transaction protocol to provide that capability. The Blockchain allows information to be stored, but if the information being stored is one part of a system validating the operation of another part of a system then both the operation and the confirmation that the operation was intended needs to be stored.

For this reason, it may not be sufficient, in a system that is intended to detect insider tampering, to only store server-provided assertions of integrity on a ledger.

The design below shows a single FHIR client that interacts with a single FHIR server. When the FHIR client provides data like a PRO QuestionnaireResponse, then the FHIR server writes the data to its persistence layer, usually transforming the FHIR data structure into a convenient persistence format.

In the architecture below a "tee" function either takes a copy of the canonical resource and signs a hash to record in the ledger or encrypts the entire resource and stores that.

The ledger can then be used to validate individual transactions and resource versions, and the server could even add transactions that validate partial or total system state to make it convenient to assess total system integrity quickly.

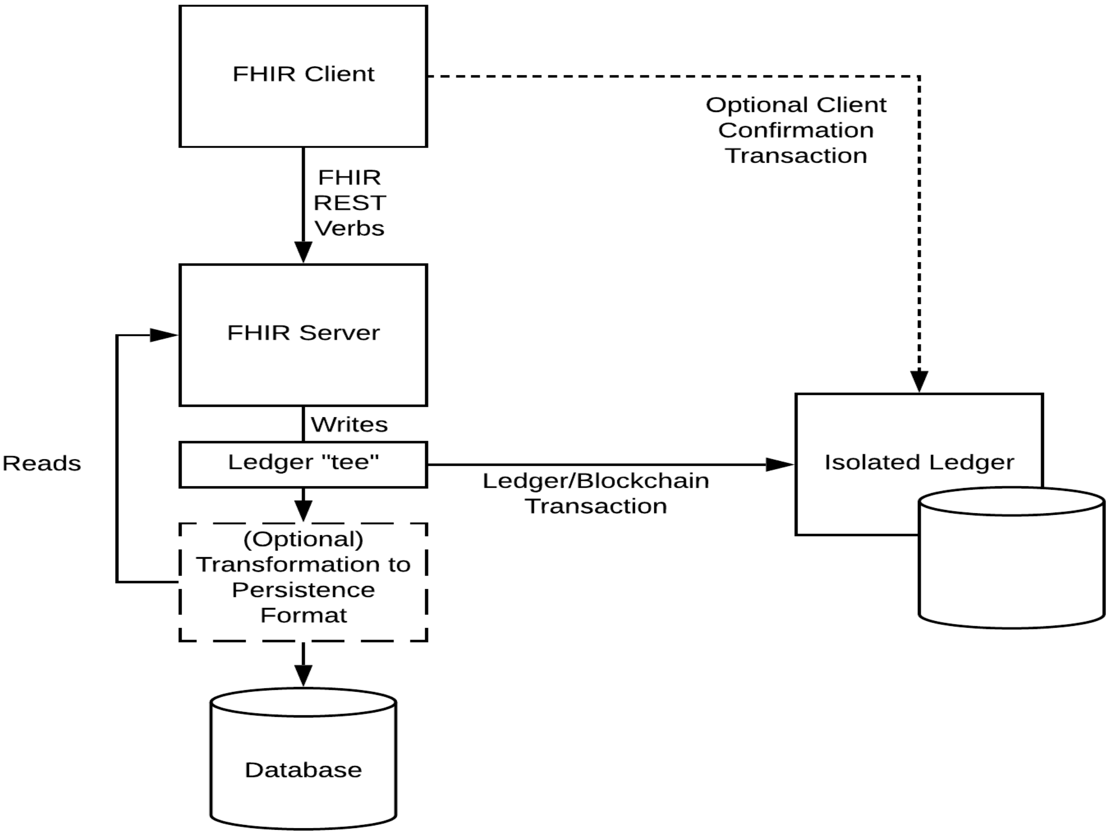

FHIR Client
Optional Client Confirmation Transaction
FHIR REST Verbs
FHIR Server
Reads
Writes
Ledger "tee"
Ledger/Blockchain Transaction
Isolated Ledger
(Optional) Transformation to Persistence Format
Database

Figure 3 - Simplified Context for Signing and Journaling

Client Participation

Because a compromised server might never persist ledger entries that would help indict it, the clients should participate in the transaction as well. Using either a ledger system with smart contracts(?) a client could enter a transaction on the ledger that confirms the integrity of the server's version or disavow it and raise a red flag. In a system like AWS QLDB (described below) the client might require a side-channel to the QLDB ledger to provide an independently sourced version of the expected resource state that can be compared under audit.

Ledger Implementations and FHIR

In this section we discuss the other options for implementation of the ledger and existing blockchain-related FHIR implementations and papers.

HyperLedger, Blockchain & Etherium-based Approaches

There are several approaches to providing a secure blockchain-based FHIR server that are useful to compare with our requirements.

"FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data" by Zhang, White, Schmidt, Lenz & Rosenbloom [7] is an approach that discusses the Bitcoin and Etherium blockchains for the purposes of clinical data sharing. The paper documents an Etherium-based approach with smart contracts and clinical data FHIR resources that are to be shared are stored directly (encrypted) on the blockchain with participating and authorized sites exchanging tokens to facilitate sharing. Inherent in this architecture are tamper-proofing and digitally signed resources that do not require provenance, as per-user wallet addresses and key pairs are maintained.

A flier for an event at the January 2019 IHE North American Connectathon by Umberto Cappellini outlines "Blockchain and the IHE QEDm/mXDE Profiles for the Provenance of Health Documents"[8]

> An MHD Document Recipient grouped with a mXDE Data Element Extractor and a QEDm Clinical Data Source will receive an MHD document bundle containing a CCDA document, convert it to FHIR resources, calculate provenance for each of them, and store it in the Hyperledger Blockchain.
>
> Later on, an mXDE Data Element Provenance Consumer grouped with a QEDm Clinical Data Consumer will query for Data Elements including the corresponding provenance information and use the grouped MHD Document Consumer to obtain the original document in which the data elements were recorded.

This seems like a similar, but more IHE-focused approach to data interchange using Linux Foundation's HyperLedger as a more private blockchain instead of Etherium.

1upHealth provides an example implementation[9] of a JavaScript FHIR provenance plugin that stores cryptographic hashes of FHIR resources stored in 1upHealth in the Etherium blockchain.

AWS QLDB-based Approach

Amazon's QLDB[10] provides a hybrid ION-JSON/relational database with a blockchain-like journal to record transactions and establish authenticity of the entire store and individual transactions. Given the combination of a signed blockchain and a fairly traditional relational model it should be possible to build a FHIR server that uses this technology to maintain a trusted journal of FHIR datastore transactions.

The ODM Play With FHIR (OPWF) service that Boston Children’s Hospital has already constructed could prototype use of the service. While OPWF uses the FHIR JSON resource format, it does not have client-server protocol semantics for persisted FHIR, so versioning is left to the underlying storage technology (S3, or in this case the QLDB journal) and this versioning information is not available through the protocol. There are some other drawbacks of a serverless FHIR implementation including the lack of FHIR-based search semantics, capability and other semantic queries.

QLDB is not a standard implementation, so it is not useful as a FHIR standards requirements, but the principle of implementing a persistence layer that maintains a ubiquitous digital ledger that is isolated and protected, or replicated, makes the tamper-evidence and tamper-proofing qualities available for all resources.

An approach could be to provide a paper specification and example implementation of a minimal non-distributed QLDB-like store that transparently maintains a digital ledger of all entity changes.

An open-source self-hosted alternative to QLDB is ImmuDB, a general immutable cryptographic journal that provides high performance storage, retrieval and integrity verification[2].

Other Approaches

One David Hay's FHIR Blog's "Tamper resistant auditing in FHIR"[11] post he discusses using both SecurityEvent (now AuditEvent[12] in DSTU3 & R4) and Provenance to ensure transaction integrity, with a PKI system to identify participants.

The system he proposes is:

- The application creates and saves the SecurityEvent resource.
- Next, it creates a Provenance resource. The provenance resource will refer back to the SecurityEvent resource (Provenance.target).
- Next generate a Hash of the SecurityEvent resource.
- Create a signature by encrypting the Hash using the systems private key and include the public key (certificate) in the signature, which is then saved in the Provenance resource. (Provenance.integritySignature).
- Save the Provenance resource.

All of the above steps would need to be executed on the server in a single transaction.

AuditEvent resources are designed to be created by any actor in a FHIR system, and not just about REST interactions[13]:

> All actors - such as applications, processes, and services - involved in an auditable event should record an AuditEvent. This will likely result in multiple AuditEvent entries that show whether privacy and security safeguards, such as access control, are properly functioning across an enterprise's system-of-systems. Thus, it is typical to get an auditable event recorded by both the application in a workflow process and the servers that support them. For this reason, duplicate entries are expected, which is helpful because it may aid in the detection of tampering. For example, fewer than expected actors being recorded in a multi-actor process or attributes related to those records being in conflict, which is an indication of a security problem. There may be non-participating actors, such as trusted intermediary, that also detect a security relevant event and thus would record an AuditEvent, such as a trusted intermediary.

As in 21 CRF 11 almost all data-affecting events are auditable, it seems reasonable that AuditEvents should be generated for each action.

At the end of the post there is a mention of the problem of being unable to use the suggested AuditEvent approach to detect when a system has had data deleted, and that a blockchain-like ledger could be appropriate. The author refers to John Moehrke, one of the original FHIR architects, as an expert in Blockchain and related items.

[2] https://immudb.io

Threat Models

This section catalogs a range of anticipated attack vectors and discusses how a 21 CFR 11 compliant FHIR system can mitigate or prevent these attacks. These are attack scenarios that a 21 CRF 11 FHIR implementation can be assessed against.

Fabrication of Client Data

There are a number of approaches that an attacker could use to attempt to fabricate data from a client.

Attacker Compromises a Client.

This is out of scope of this paper as it is dealt with through security technologies and processes already required for a FHIR PRO system.

Client security can be assured through the use of OAuth2 and OIDC authentication and authorization as recommended by the security standards referred to by the PRO Implementation Guide.

If a client's credentials are compromised, or if there is a security process failure resulting in a client compromise then the mitigation would be to invalidate the credentials of that client or user and flag the data provided as tainted. The processes around identifying cohort members who have withdrawn consent would be appropriate to use for clients that have provided PRO data while compromised.

Attacker Creates a Fake Client

This would require a compromise of the study enrolment process and creation of credentials based on impersonation or misrepresentation. As such it is out of scope of this paper, but a major concern of any study collecting PROs.

Tampering with Server Data

In order to tamper with server data an attacker would have to penetrate the server's security or be an insider with privileges necessary to change the data persisted on disk. Any resource, or version of a resource, that is changed or corrupted can be checked against the digital signature provided by the originating client in the corresponding Provenance resource.

If an attacker corrupts a FHIR server to a non-working condition, then application level integrity checks and application failures would indicate that a compromise has occurred, and IT recovery procedures would have to be started. See reference 21 CFR Part 11 section 11.10 (c), which states "(c) Protection of records to enable their accurate and ready retrieval throughout the records retention period."

Modify a Resource

If an attacker has access to the server then modifying a resource will be detectable from verifying the resource against the client signature provided in the Provenance resource.

An additional integrity violation cue for an auditor is a large number of changes at once for clients that would typically provide PROs over a longer timeframe.

Add a resource

When an attacker adds a resource then they would also have to secure the secret key of a valid client to fabricate a Provenance resource for a new resource or the invalid signature would indicate that tampering has occurred.

Remove a Resource

When an attacker removes a resource then the existence of a Provenance resource for a missing resource would indicate that tampering has occurred.

Fail to persist a resource and a corresponding provenance resource

This information would never be added to the server and would not appear in the ledger. If the client is not able to directly verify the presence of its signature in the ledger then this may go undetected.

Remove a Provenance Resource

When an attacker removes a resource and its corresponding Provenance resource then this is not detectable unless there is also a journal that can be cross-referenced against the system state.

If the attacker tries to delete or modify the journal, then any new change to the server would detect that the journal is in an inconsistent state and raise the alarm.

Replay Attacks

It is possible that an advanced attacker could use the journal to remove resources by completely replaying all of the transactions recorded in the journal except those that the attacker wished to remove. This would maintain the entry-to-entry cryptographic integrity of the journal and remove targeted resources.

To defend against this attack then some other approaches could be employed:

- Maintaining the journal externally to the FHIR server, maintaining a read-only replica, or maintaining a sanitized signature-only journal in a public or isolated system like Etherium, Bitcoin Blockchain, HyperLedger or the Amazon Web Services Quantum Ledger Database (QLDB).

Reference Architectures

Three main architectures are discussed below to illustrate the implementation concerns and the unique requirements that impact these systems. A purely FHIR-based system with FHIR clients and servers that interact entirely within the scope of the PRO Implementation Guide is the easiest to discuss. In generating the 21 CFR 11 extensions to the PRO Implementation Guide some real-world systems where a FHIR server is the ultimate store for the PRO data, but the electronic data capture is implemented by an industry standard EDC with options for standards-based yet non-FHIR or non-QuestionnaireResponse resource export of the PRO data.

Non-FHIR to FHIR Gateway

One of the most common architectures for a PRO collection system is when a non-FHIR EDC like Medidata RAVE or EPIC is used to collect the data. Downstream systems for warehousing, analysis and distribution of the PRO data can be implemented as FHIR if a gateway is employed to transform the collected EDC format into FHIR for storage and analysis access.

In this architecture the EDC delivers PRO questionnaires to study subjects and study protocols handle the identification of the subject. The boundary of the 21 CFR 11-compliant system is larger than the FHIR ecosystem, so the FHIR Implementation Guide has to provide guidance about how to handle the upstream data.

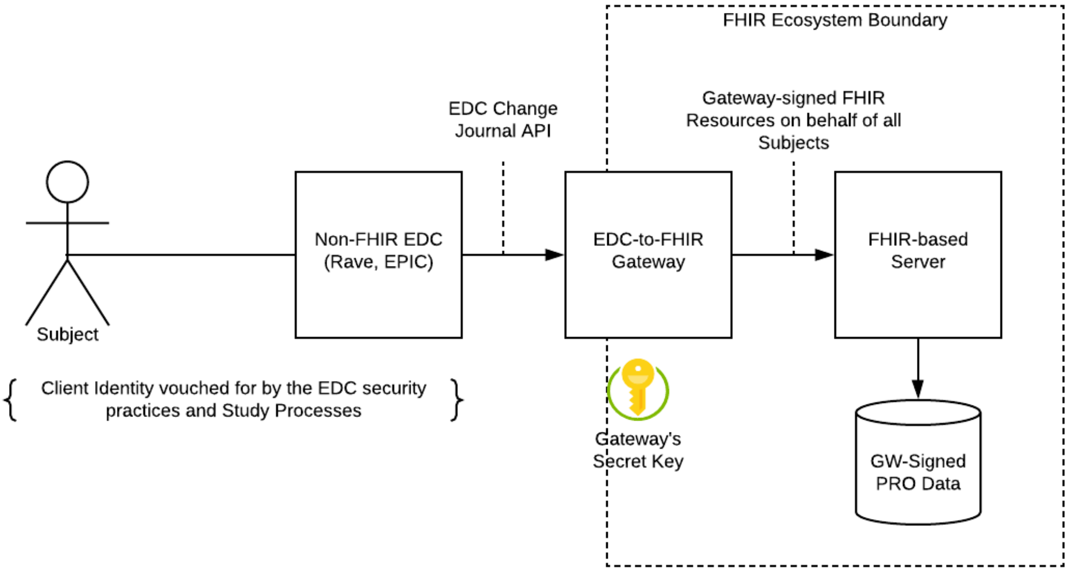

FHIR Ecosystem Boundary
EDC Change Journal API
Gateway-signed FHIR Resources on behalf of all Subjects
Subject
Non-FHIR EDC (Rave, EPIC)
EDC-to-FHIR Gateway
FHIR-based Server
{ Client Identity vouched for by the EDC security practices and Study Processes }
Gateway's Secret Key
GW-Signed PRO Data

Figure 4 - System Context and Signing Location (Gateway Archetype)

Any upstream EDC much have the ability to present a journal of changes to the collected PRO and related state data that allows an external system like a FHIR-based warehouse to replicate the state of all PROs in the EDC externally and repeatably.

To establish the provenance of the data the upstream EDC has to be trusted by the 21 CFR 11-compliant FHIR Ecosystem to associate an accurate non-repudiable identity with each transaction. The Gateway may need to provide an ability to protect the identity of the stored subject data and protect against disclosure of personally identifying information, but this is beyond the scope of the Implementation Guide.

The FHIR system's approach for asserting provenance at the gateway application is to use its own identity to sign resources imported from the EDC. By doing so the FHIR system is asserting that the FHIR resource persisted from the upstream EDC's data representation is an accurate reflection of the resource it constructed from the EDC's representation, and that it trusts that the identity and metadata associated with the EDC's data accurately represents the provenance of the PRO. The resources are added, updated and deleted and Provenance with external resource signatures generated at the gateway.

In a purely FHIR environment the subject, or the client that is the proxy for the client's identity would assert the provenance of the PRO data. In the gateway architecture the Provenance resource with the external resource signature asserts that the resource received from the EDC was persisted accurately by the gateway.

Entirely FHIR Implementation

In an entirely FHIR implementation of a PRO collection and persistence system, full provenance can be captured. After enrolment controls, an application or app used by a subject can establish a secret key and request an identity certificate using either the OAuth sequence of a SMART on FHIR server, or an Open ID Connect server and a manager public key infrastructure (MPKI) that provides consumers of the certificate and generated signatures with the ability to determine the veracity of data, and the status of the identity certificates used to sign the contributed PRO data.

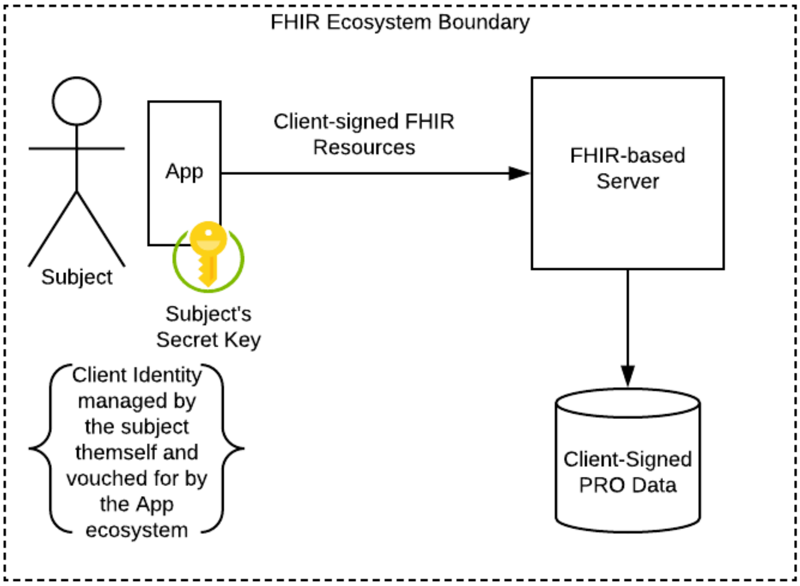

FHIR Ecosystem Boundary
Subject
App
Client-signed FHIR Resources
FHIR-based Server
Subject's Secret Key
Client Identity managed by the subject themself and vouched for by the App ecosystem
Client-Signed PRO Data

Figure 5 - System Context and Signing Location (Entirely FHIR System)

If a subject uses apps on multiple devices or multiple PRO reporting applications then the identity provider or identity certificate generation process should ensure that the subject identifier is accurate, even though the secret key may be different for each source app or application. The identity certificate thumbprint and subject identifier will be referenced in the Provenance resource so that the correct certificate status can be established, and signature verified.

A Provenance design pattern for systems without QuestionnaireResponse resources

The EPIC Electronic Data Capture system[14] provides an optional FHIR-based interface to access some data in its internal proprietary stores. For the collection of PRO data EPIC can make use of LOINC codes as metadata for PRO Questionnaire answers. Unlike the FHIR PRO Implementation Guide recommendations, a system that takes data from the EPIC FHIR server API provides PRO response as Observation Resources instead of elements in a QuestionnaireResponse resource.

The figure below shows the flow of data in three model architectures:

1. The architecture described in the FHIR PRO Implementation Guide

2. PRO questionnaires delivered by and responses stored by RAVE before being warehoused in a FHIR server store as QuestionnaireResponse resources, and
3. PRO questionnaires delivered by and responses stored by EPIC before being warehoused in a FHIR server store as Observation resources

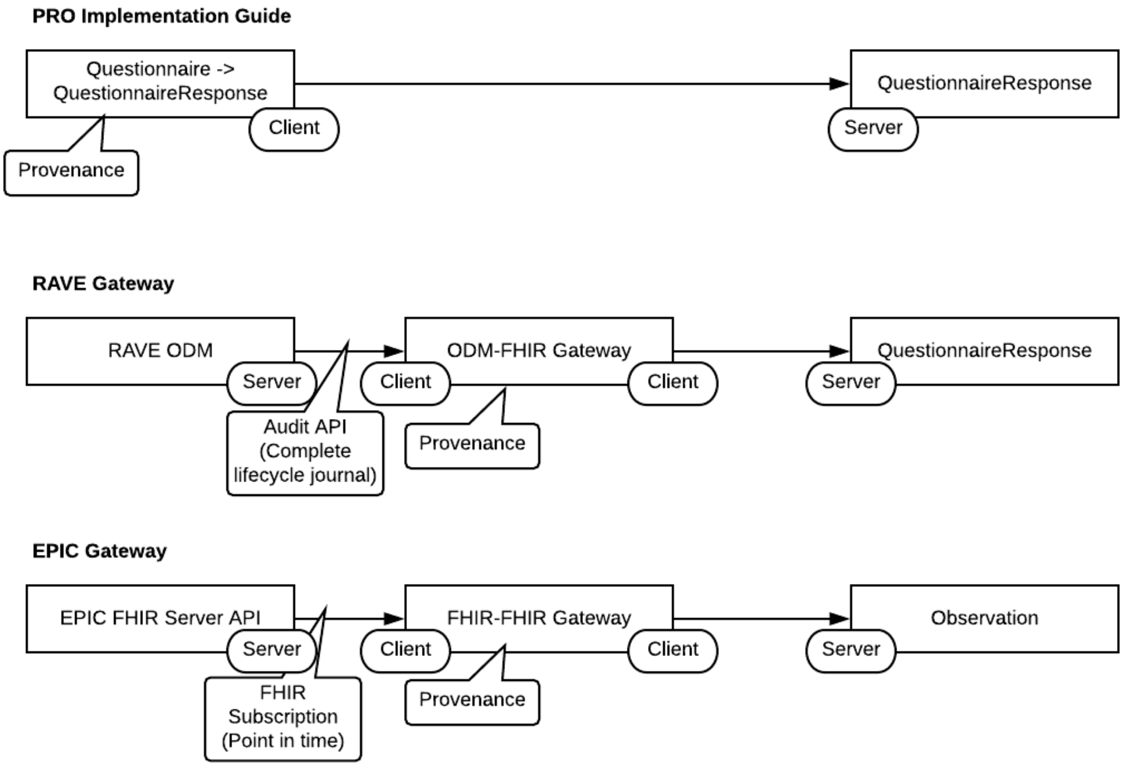


Figure 6 – Data Flows for FHIR QuestionnaireResponse and Observation Resource Persistence

This architecture presents a similar Provenance and auditing problem to other non-FHIR to FHIR gateway architectures. The 21 CFR 11 PRO Implementation Guide describes a design template for a FHIR server that maintains a tamper-evident journal of state changes of a subset of resources and those resources can be of any type. This design template will also work to assure the integrity and auditability of Observation resources.

The “upstream” EPIC system needs to be separately assessed for 21 CFR 11 compliance, and the downstream FHIR implementation can use the Provenance resource to store digital signatures of the stored Observation resources in the same manner as has previously been described for QuestionnaireResponse resources.

The FHIR server storing the Observation resources should also implement a journal to provide comprehensive tamper-evidence.

An implementation issue that is not clear from the public information about EPIC’s FHIR and other API support is whether there is a complete journal of data state changes that can be used to verify that the state of the source data is fully synchronized with the FHIR server, or allow creation or recreation of the FHIR warehouse data at any time.

Public Key Infrastructure Design Considerations

A system that uses public key infrastructure (PKI) requires careful design and planning. Clients and servers must agree to trust a certificate authority (CA) to issue identity certificates and maintain certificate status and many public certificate authorities exist that can provide this service.

Identity and CA certificates need to be provided to system participants like auditors to verify the integrity of signatures or decrypt information. Certificates expire, are revoked, and each certificate's content and status have to be available to users of the system for the lifetime of the stored data to allow future verification of system and resource integrity.

Storing identity certificates so they can be accessed when required for auditing requires a protocol that sits beside the PRO QuestionnaireResponse delivery that provides the FHIR Server with copies of the client's certificate whenever the client's certificate is changed or used to sign a resource. If the appropriate identity certificate to verify a resource can't be found by an auditor, then that resource's provenance can't be established, and it should be treated with suspicion or excluded from use.

Additionally, an identity certificate's status needs to be considered. If a certificate's status is that it has been invalidated, then resources signed with that certificate after revocation do not have established provenance and should be excluded from use.

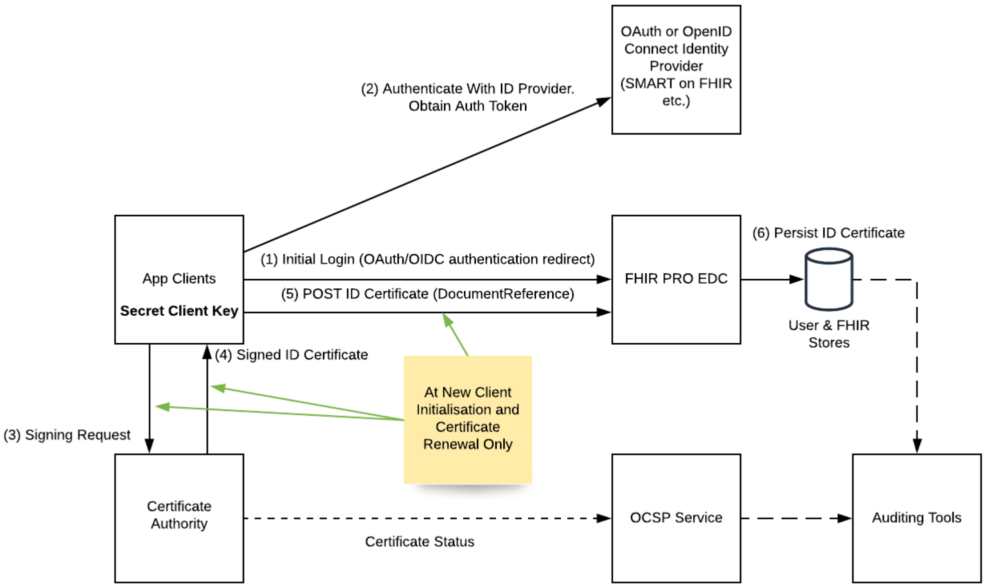

OAuth or OpenID Connect Identity Provider (SMART on FHIR etc.)
(2) Authenticate With ID Provider. Obtain Auth Token
App Clients
Secret Client Key
(1) Initial Login (OAuth/OIDC authentication redirect)
(5) POST ID Certificate (DocumentReference)
FHIR PRO EDC
(6) Persist ID Certificate
User & FHIR Stores
(4) Signed ID Certificate
(3) Signing Request
At New Client Initialisation and Certificate Renewal Only
Certificate Authority
Certificate Status
OCSP Service
Auditing Tools

Figure 7 - Certificate Management Model

An app-style client in Figure 7 would:

- Log in to the FHIR Server working through the OAuth2 or OIDC protocol to establish a secure session
- Once logged in, check to see if the client's existing certificate is present and valid (not expired or revoked)
- If there is no client certificate, then use a local secret to create an identity certificate signing request. Send the CSR to the CA's CSR API and receive the CA-signed identity certificate.
- POST the identity certificate inline in a DocumentReference to the FHIR PRO server.

As with PRO data, the client has to send a subsequent Provenance resource with a digital signature for the DocumentReference resource to notarise that the certificate was properly persisted by the server. If this was not done then an insider attacker could substitute another identity certificate.

For "gateway model" PRO systems, the process is the same, but may not require an automated CA. Acquiring a single identity certificate for the gateway can be a planned and manual process, though the same requirement to store the certificate in the FHIR server exists.

Identity Certificates, OAuth2 and HIPAA

Because identity certificates are issued against an OAuth2-authenticated identifier, the certificate that is stored in a 21CFR11 PRO store is identifying information in most circumstances. Care should be taken to ensure that auditors and others with access to the DocumentResource public certificate information are duly authorized.

Public CAs and Managed PKI

Most public CAs have a service for manually providing public certificates for certifying domain names and hosts and some offer fast or automated methods of issuing personal identity certificates for email signing and encryption or applications like ours.

Some public CAs offer an Automatic Certificate Management Environment (ACME) (RFC-8555) service for automating SSL certificate generation, but the "challenge" used to verify the identity of the certificates issued require Domain Name Service (DNS) or HTTP to the host requesting the certificate. This protocol would be suitable for our needs if it supported a challenge that verified a user identity (email address or logged-in status).

There is a class of service called Managed PKI services where a local certificate authority is delegated authority to issue specific types of certificates within a chain of trust. These managed PKI services are rigorously audited by certificate authorities, and recently have become easier to access because cloud vendors offer services like AWS KMS, Azure Key Vault and Google Key Management Service all provide APIs to support.

Another option is to run a self-signed certificate authority, independent of any broader trust arrangement. This would be fine for totally internal use, and would provide audit capabilities, but would require additional verification by 21CFR11 auditors that might otherwise come from the governance requirements of a public CA or the CA of a managed PKI service.

The upside of a private certificate authority is that many programming libraries are available to support a modified ACME protocol or API-based CA service.

What capabilities does the CA need?

In the diagram above the process of generating an identity certificate and having it signed by a mutually agreed CA is the responsibility of the client. The CA has to be able to correctly identify the client to meet 21CFR11 requirements. Given the system's client authentication also needs to meet requirements for accurately identifying, consenting and authorizing patients this can be leveraged as proof of identity for the system to provision CA-signed identity certificates to those patients' FHIR clients. These identity certificates are stored on the FHIR server and the patient's private key can be used to sign FHIR resources and provide that signature in a cryptographically-verifiable Provenance resource.

Manually processing certificate signing requests for patient participants would be complex and unwieldy for most systems, so automation is highly desirable.

A client adhering to the PRO and related implementation guides might use OAuth2 or OpenID Connect to authenticate with a SMART on FHIR service with an email address or other form of user ID. Once authenticated, the subject has proven they have ownership of the account. The authenticated session and OAuth2/OIDC claims are sufficient proof that a certificate can be signed and issued by a system's managed certificate authority.

In a closed gateway system, the client might manually interact with a CA to generate an identity certificate and REST POST or otherwise provide the certificate only on the rare occasions when it is changed or regenerated.

How much to specify in the IG?

For the implementation guide there is a question of how prescriptive the specification should be. An example is a system with a built-in certificate authority that relies on the FHIR server and client being aware of OAuth2 and providing an ACME-like interface that the client can call when it requires a new identity certificate for signing resources. The CA provides features for logging the revocation of certificates.

This model can be extended to an external CA, or a managed PKI, but there is no method of automating certificate signing without implementing a custom API that I know of. Prescribing a modified ACME[15]-like protocol for the implementation guide makes the solution more complex, but perhaps an example implementation would help facilitate adoption and lower implementation complexity.

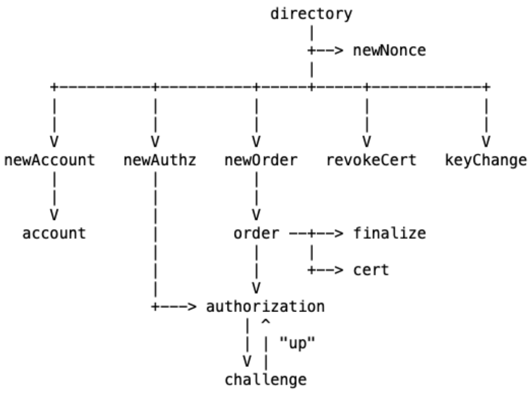
A modified ACME API or a Simple REST API

ACME is a protocol largely for automating the signing of certificates for use to secure Internet domains and web sites. There is no direct means *in the standard* to automate the signing request and proof provision required to issue an identity certificate, but it is mentioned:

> *Note that while ACME is defined with enough flexibility to handle different types of identifiers in principle, the primary use case addressed by this document is the case where domain names are used as identifiers.*

The ACME API is a useful framework for understanding the requirements of a 21 CFR 11 system where FHIR clients need to perform certificate orders. However, ACME makes some assumptions that can cause complexity:

1. ACME assumes it will operate as a standalone server with its own registration and authentication mechanism that is OAuth-like in that each client request must provide a nonce. This parallels and duplicates functionality an OIDC or OAuth 2.0 "resource server" implementation could provide.
2. The standard flow for ordering a certificate includes a requirement for a client to poll until a set of validation criteria is met, but these indeterminate duration tests can be omitted from the protocol.



ACME Resources and Relationships

Figure 8 - RFC-8555 ACME Protocol APIs

As a 21 CFR 11-compliant FHIR server already has an OIDC or OAuth 2.0 authentication and authorization interface, the account and authorization processes should be ignored and the OIDC or OAuth 2.0 equivalents used.

The API endpoints required for a service that is providing clients with a certificate issuance service are:

- /sign – request that the server signs the client's provided certificate. The Server verifies the client is authenticated and that the identity in the Canonical Name (CN) of the certificate matched the identity of the user.
- /keyChange – Inform the server that the client has regenerated its private key.
- /revokeCert – notify the server that the certificate has been revoked as of a specified date.

A client that has successfully authenticated can order an identity certificate from the 21CFR11-compliant system. Mechanically, the client uses its authorization token as prrof the client ID is already known and can be issued a system-signed identity certificate with the authenticated ID. In this regard no challenge need be issued (or the challenge portion of the protocol can be ignored) and the order object can transition directly to the valid state. The REST methods newOrder, revokeCert and keyChange are all available to an authorized client.

When a server issues a certificate, the client stores a copy of that certificate with the FHIR server by POSTing it embedded in a DocumentReference[16] with a reference to the client's associated Patient resource as a context/sourcePatientInfo reference or Device resource context/related reference (depending on whether the client is a Patient client or a gateway application for an EDC) and an accompanying signature in a Provenance resource.

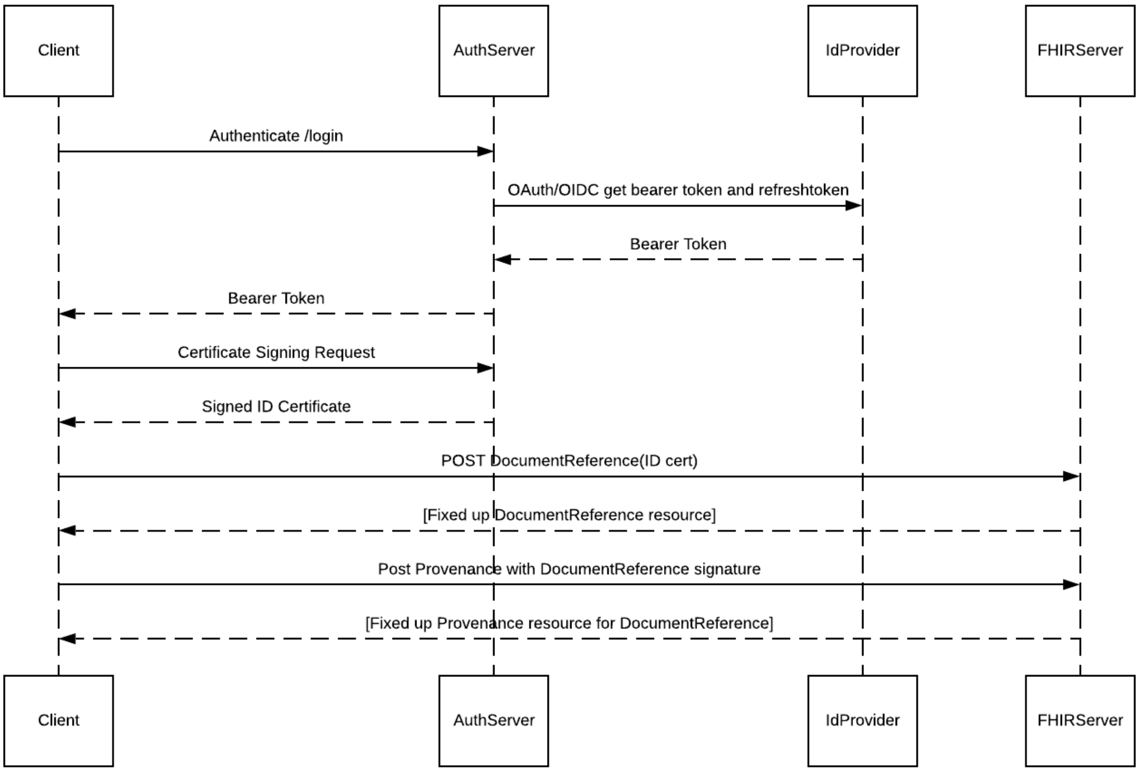

Client
AuthServer
IdProvider
FHIRServer
Authenticate /login
OAuth/OIDC get bearer token and refreshtoken
Bearer Token
Bearer Token
Certificate Signing Request
Signed ID Certificate
POST DocumentReference(ID cert)
[Fixed up DocumentReference resource]
Post Provenance with DocumentReference signature
[Fixed up Provenance resource for DocumentReference]
Client
AuthServer
IdProvider
FHIRServer

Figure 9 - OAuth 2.0 and FHIR flow for ID certificate orders

Illustrative Use Cases

Protection against external breach

Protection against external breaches is mostly dealt with in the FHIR Security Specification[17] Requirements referenced from the PRO Implementation Guide[18].

This specification describes mechanisms for Authentication, Authorization and Access Control and how to communicate access denial. To summarize, OpenID Connect (OIDC) [19] is the primary recommendation for authentication, otherwise OAuth2[20] with the SMART-on-FHIR[21] OAuth implementation profile as a model implementation.

Use case 1: An attacker attempts to authorize with an unregistered user ID or bad credentials

Primary Actor: Attacker (An unregistered user)

Scope: 21CFR11 PRO FHIR System Authentication Subsystem

Brief:

The attacker attempts to authenticate as a client with the FHIR PRO system though they are not registered with the system or uses incorrect credentials for a registered user.

Stakeholders:

- Auditors & System Administrators – can determine that an unsuccessful authentication attempt was made and the user identifier attempting the authentication from the system logs.

Postconditions

Minimal Guarantees:

- A log of successful and unsuccessful authentications is available for Auditors & System Administrators

Success Guarantees:

- The attacker is prevented from accessing any part of the system other than authentication.
- The attacker will not proceed to authorization or be able to access any system state.

Preconditions:

- The attacker has access to the authentication system.

Basic flow:

The attacker uses a client to attempt to access the FHIR based PRO system.

The FHIR-based system presents an authentication interface. This could be in an app, or a web page that is using OpenID Connect or OAuth 2.0 as the underlying technology to interface with the Identity Provider System.

The attacker enters their incorrect credentials (credentials for a user that does not exist, or for a user that does exist, but without the password and other identity verification factors).

The system informs the attacker that the credentials are incorrect and may navigate away from the authentication interface to mitigate the performance impact of simple brute force attacks.

Extensions:

An OpenID Connect or OAuth 2.0 authentication can require many forms of identification. The steps that are taken during authentication could simply be entering a user ID and password, using a hardware token, or being challenged to enter a one-time password code sent through SMS or generated by an app.

Use case 2: (Insider Tampering) An authenticated attacker submits data on behalf of another user

Primary Actor: A malicious registered patient (A registered user with intent to fabricate data for another user)

Scope: 21CFR11 PRO FHIR System Authentication and Authorization Subsystems

Brief:

The attacker authenticates as a client with the FHIR PRO system and then submits data for another user. This assumes that a client has been altered such that the attacker can attempt to enter data for a different user. The system will accept the resources, but the resources will not have a signature that matches the client they are signed on behalf of.

Because we do not change the FHIR protocol the emphasis is on detecting malfeasance and keeping all data in a journal of state changes. We rely on analysis and auditing to detect attempted counterfeiting of PRO data by insider threats. When this kind of attempted counterfeiting by an enrolled client is detected then it is the responsibility of the auditor or analyst to ensure this data is excluded from analysis and action is taken to prevent subsequent fraudulent behavior.

Any analysis should validate the data before use, any ETL should filter data without valid provenance.

Stakeholders:

- Auditors & System Administrators – can determine that the data submitted for the user is invalid.

Postconditions

Minimal Guarantees:

- A log of successful and unsuccessful authentications is available for Auditors & System Administrators
- Each client-submitted resource can be verified by whether the signature used to sign the resource belongs to the patient in the Provenance resource and whether the signature is correct for the resource referred to in that signing Provenance resource.

Success Guarantees:

- The submitted resource has an accompanying Provenance resource that allows an auditor or analyst to detect if the resource was signed with a secret key that belongs to the patient's client.

Preconditions:

- The attacker has access to the authentication system.
- The attacker can authenticate as a valid client of the system.

Basic flow:

The attacker (an authorized user) uses their client to authenticate themselves the FHIR based PRO system.

The FHIR-based system presents an authentication interface. This could be in an app, or a web page that is using OpenID Connect or OAuth 2.0 as the underlying technology to interface with the Identity Provider System.

The attacker enters their *correct* credentials.

The attacker submits a QuestionnaireResponse for another patient.

The attacker generates a signature for the QuestionnaireResponse with a secret key that is not associated with the client or Patient the QuestionnaireResponse data corresponds to.

Extensions:

The attacker could attempt to provide a certificate to the FHIR store that claims to be for the client that they are providing masqueraded data for ahead of sending the QuestionnaireResource data.

The server could place a server-side intermediary filter between the client and FHIR server to ensure that the client only sends resources that reference their own Patient resource and return a 401 status for violations.

Protection against internal corruption

Primary Actor: Willful or accidental system corruption (Abstract actor)

Scope: 21CFR11 PRO FHIR System Audit Capabilities

Brief:

When an insider alters data or a computer or storage system corrupts data it should be evident that data has been corrupted. A 21 CFR 11 PRO FHIR server has perimeter security to prevent unauthorized access, and temper-evidence to detect changes to data without verifiable Provenance. A 21 CFR 11 PRO FHIR server is unconcerned with the distinction between system corruption and tampering by a system intruder or insider attacker.

When the data stored in the FHIR server is altered any user of that data should be able to make use of both the cryptographic journal and Provenance resource signatures to verify the integrity of collected data.

Any analysis should validate the data before use. Any ETL should filter data without valid provenance. Any evidence of tampering or corruption should initiate an incident response to analyze the threat and scope of the system compromise.

Stakeholders:

- Auditors & System Administrators – can determine that the data submitted for the user is invalid.

Postconditions

Minimal Guarantees:

- A log of successful and unsuccessful authentications is available for Auditors & System Administrators
- Each client-submitted resource can be verified by whether the signature used to sign the resource belongs to the patient in the Provenance resource and whether the signature is correct for the resource referred to in that signing Provenance resource.
- The cryptographic journal provides a method or verifying journal integrity and the cumulative and historical state of system resources.

Success Guarantees:

- Submitted resources have an accompanying Provenance resource that allows an auditor or analyst to detect if the resource was signed with a secret key that belongs to the patient's client.

Preconditions:

- The attacker has read and write access to system internals.

Basic flow:

The attacker makes a change to the system, changing a response element in a QuestionnaireResponse.

An auditor checks the data by:

1. Verifying the cryptographic journal has not been tampered with (usually with verification of cumulative and block hashes)
2. Verifying each resource's signature in its corresponding Provenance resource

When the signature is invalid then the resource is examined and, optionally, the journal checked to determine which of the Provenance or Resource might have been altered.

Extensions:

Server data corruption can be assessed from the same flow.

Safety Requirements

The role of 21 CFR Part 11 compliance in Safety Processes

During a clinical trial, post marketing surveillance or clinical study, Serious Adverse Events (SAE) and Emerging Safety Issues (ESI) are handled with processes that initiate a process of reporting, analysis and remediation. The additional provenance and integrity measures provided by a 21CFR11-compliant system can assist with immediately providing provenance for reported outcomes. This aids in the rapid understanding of the impact and background of an adverse event or a safety issue that goes beyond the usual detail and implicit integrity of a traditional audit trail.

A typical adverse event reporting process involves these steps:

- A site or individual in the study discovers a reportable event.

- The record of the event is reported to a safety reporting system.
- Clinical and other information is gathered about the subject and the event. Investigators ensure a complete record is gathered.
- A report is created and communicated to the study's principal stakeholders and study sites.
- Study sites notify local IRBs per their guidelines.
- Sites and stakeholders notify affected participants according to their reporting obligations.

These steps are often automated. For instance, a RAVE EDC-based system could be configured to automatically or semi-automatically enter a safety notification into a safety response workflow like Argus for management and normalization of the submission to local standard formats[3].

Technical advantages from adopting the 21 CFR 11 PRO Implementation Guide

When a patient or other authority reports an adverse outcome or some evidence that indicates a safety issue then a study or trial can make all of the PRO data related to that event along with a cryptographically verifiable history of that event for analysis by authorized staff or stakeholders. When a condition is detected by the system or via manual intervention, the FHIR-base PRO repository has the information necessary to generate data for the relevant subject and their complete PRO reporting history including changes and deletions, that is a reproducible record of the can be used for reporting and contact procedures.

Without the 21 CFR 11-compliant data then there might be a need to manually notarise gathered evidence, slowing down the time to notify affected stakeholders or take corrective action. Because the system can report its own integrity and the hisotyr of all PRO and PRO-related resources, threats to safety like tampering and system corruption are able to be tested and quickly discounted from, or included in, analysis.

Pitfalls of likely implementations

Inevitably, some 21 CFR 11 systems will be a hybrid of the system described in the 21 CFR 11 PRO Implementation Guide and FHIR and non-FHIR systems that have to take a traditional approach to 21 CFR 11 compliance.

To meet 21 CFR 11 compliance, the combination of these systems will need to be designed with the requirements of 21 CFR 11 and audited for compliance. Simply using the 21 CFR 11 PRO Implementation Guide on a portion of an entire system is not going to grant 21 CFR 11 compliance to other system components.

Issues and Shortcomings

1. Should the journal always be isolated from the FHIR server to prevent replay attacks?

[3] Examples are Medwatch or CIOMS format

2. FHIR does not have semantics for a one-shot resource-signing transaction. During network outages or protocol failures resources might be persisted in the FHIR server that have no accompanying provenance resource.

3. FHIR systems often have to interface with non-FHIR systems. When, for instance, the FHIR client is an ETL or gateway process, the client can provide provenance for itself, but not "upstream" systems. Should the implementation guide include a method of asserting or describing the approach to provenance in upstream systems?

4. Information that has been entered cannot be deleted. When a patient asks for their data to be removed, how can it be removed from a blockchain which has a requirement to retain every piece of history? The FHIR specification makes support for deletion an optional capability, so it may be that the Implementation Guide should recommend an approach to deletion based on the jurisdiction or consent contract of the study the system supports.

For example - where a patient's relationship with a resource is 1..N and the resource has no other "ownership" relationship, then even if individual transactions can't be removed, the entire chain of events pertaining to that patient can be removed in isolation.

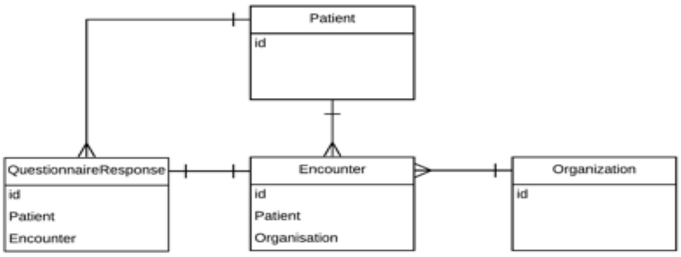


Figure 10 – Typical FHIR PRO Resources E-R Diagram

Acknowledgements

This project was supported by the Food and Drug Administration (FDA) of the U.S. Department of Health and Human Services (HHS) as part of a financial assistance award [U24FD006543] totaling $273,666 with 100 percent funded by FDA/HHS during the project period 01-September-2018 through 31-August-2022. The contents are those of the author(s) and do not necessarily represent the official views of, nor an endorsement, by FDA/HHS, or the U.S. Government. Subsequent funding support for incorporation and real-world usage of this project into the Childhood Arthritis Research and Rheumatology Alliance (CARRA Registry) data warehouse at Boston Children's Hospital is being provided by CARRA, Inc.

References

"National HIE Governance Forum – Identity and Access Management for Health Information Exchange " December 2013
https://www.healthit.gov/sites/default/files/identitymanagementfinal.pdf

1. 21 CFR Part 11 - Sec. 11.10 Controls for closed systems "Persons who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records, and to ensure that the signer cannot readily repudiate the signed record as not genuine."

2. 21 CFR Part 11 - Sec. 11.10 (a) "Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records."

3. http://hl7.org/fhir/us/patient-reported-outcomes/2018Sep/index.html

4. 21CFR11 Sec 11.10

5. 21 CFR Part 11 Sec. 11.10(b) "(b) The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records."

6 http://hl7.org/fhir/us/patient-reported-outcomes/2019May/

7. https://www.dre.vanderbilt.edu/~schmidt/PDF/FHIRChain-jnca.pdf

8. https://grapevineworldtoken.io/media/filer_public/98/ae/98aeae30-b3c6-45de-bc66-17a7b94f7d91/grapevine_brings_blockchain_to_ihe.pdf

9. https://github.com/1uphealth/fhirprovenance

10. AWS Quantum Ledger Database: https://aws.amazon.com/qldb/

11. https://fhirblog.com/2014/02/27/tamper-resistant-auditing-in-fhir/

12. https://www.hl7.org/fhir/auditevent.html

13. https://www.hl7.org/fhir/auditevent.html#bnc

14 https://www.epic.com

[15] Automatic Certificate Management Environment https://tools.ietf.org/html/rfc8555

[16] DocumentReference resource reference: https://www.hl7.org/fhir/documentreference.html

[17] FHIR Security Specification: http://hl7.org/fhir/security.html#http

[18] PRO Capability Statement – Security Requirements: https://build.fhir.org/ig/HL7/patient-reported-outcomes/capstatements.html#security-requirements

[19] OpenID Connect Specifications: https://openid.net/developers/specs/

[20] RFC-6749 – The OAuth 2.0 Authorization Framework: https://tools.ietf.org/html/rfc6749

[21] SMART on FHIR Resources: http://docs.smarthealthit.org