REDCap™

Clinical Data Interoperability Services

Technical Information

(How it all works)

This document contains the technical information regarding the details of how the “Clinical Data Interoperability Services” functionality works in REDCap. It describes how REDCap can be launched inside the EHR user interface, how the EHR and REDCap communicate securely with each other, and how data flows into REDCap from the EHR using the SMART on FHIR web services.

Note: This document is not required for setting up the “Clinical Data Interoperability Services” functionality in REDCap. It is provided here merely to offer a technical background and security context for those interested in how these features will function in REDCap.

A. Launching REDCap from inside the EHR

After the “Clinical Data Interoperability Services” functionality has been fully set up and enabled, there exist two methods for establishing a connection between their EHR account and their REDCap account. And once the connection with the EHR has been established, the user may begin pulling clinical data into REDCap from the EHR. The first method is using an embedded REDCap window inside the EHR interface, and the second is for a user to log in to their EHR directly while inside REDCap after being prompted. Both methods require a login both to the EHR and to REDCap at the same time. Prior to REDCap 9.5.2, launching REDCap from inside the EHR was the only way to establish connection with the EHR, but now this can be done directly via a login to the EHR while inside REDCap.

When REDCap is launched inside the EHR, the EHR will pass the user’s EHR username to REDCap, and the very first time the user does this, they will have to log in to REDCap using REDCap’s normal login page. Once they have logged in, REDCap will have their EHR username stored in its database, and it will know the REDCap account to which the EHR username belongs. Thus when a user launches REDCap from the EHR at any time afterward, they will not need to log in to REDCap because REDCap will automatically know who they are once they have passed the EHR/FHIR authorization process.

Depending on the EHR vendor and how the vendor has configured the launch point in the EHR, REDCap may be launched by a button, link, or menu item from any number of places in the EHR application. Regardless, REDCap will be launched as an embedded window inside the EHR. This is called an EHR launch context that initiates the SMART on FHIR authorization process, which utilizes OAuth 2, an industry-standard protocol for authorization.

The SMART on FHIR authorization process begins with the EHR calling the REDCap “Redirect URL”, which is provided on the “Clinical Data Interoperability Services” setup page in the REDCap Control Center. The EHR redirects the EHR user to the REDCap Redirect URL, passing a session-specific launch identifier to REDCap. This request to REDCap is secure and occurs over an SSL/TLS connection. REDCap then receives this request from the EHR, after which REDCap calls the EHR’s FHIR/authorization server. (Note: Although the typical SMART on FHIR process will make a request to the FHIR Conformance Statement, which itself contains the URL to the FHIR authorize end-point and token end-point as seen in Diagram 1, since REDCap will already have these URLs stored, that step is skipped in the process in order to save time.) REDCap then redirects the user to the authorize end-point URL on the FHIR/authorization server, and it sends the launch identifier that was initially provided by the

EHR. The authorize end-point will validate the launch identifier, and if valid, will redirect the user back to REDCap's Redirect URL, returning an authorization code as a parameter in the URL's query string. REDCap then exchanges that authorization code for an access token by making a POST request to the FHIR/authorization server's token end-point. If the authorization code is valid, a FHIR access token is returned. This access token can be thought of as a temporary API key and is what will be used when making FHIR requests to the EHR later to export patient data.

REDCap has a FHIR client ID and client secret, which is stored in REDCap and was provided by the EHR technical team during the setup process. Both the client ID and client secret are sent in the HTTP Basic Authorization header to the token end-point to request a new access token for a user. The FHIR/authorization server uses the client ID and secret to confirm the identity of the sender for security reasons.

After REDCap has received a FHIR access token from the token end-point, the access token will have an expiration time (typically one hour or less), and after that expiration time has passed, the token will no longer be viable for exporting clinical data. However, the setup of the EHR/FHIR functionality in REDCap requires that Refresh Tokens be enabled, which means that any expired token can easily be swapped for a new token by making another request to the FHIR/authorization server's token end-point (seen in Diagram 2).

After a user completes the authorization process and receives an access token, the token is stored in REDCap's database to be used at any point afterward for exporting clinical data from the EHR via the FHIR web services. When that token expires, REDCap can automatically refresh the token at any time so that REDCap will always be able to pull data via Clinical Data Pull from the EHR. This maintaining of a valid access token via the refresh method allows for continuity of service for the CDP functionality. The workflow for refreshing an access token is displayed in Diagram 2.

B. Using Clinical Data Pull (CDP)

Launching REDCap from inside the EHR is the first initial step that a user must complete before utilizing other parts of the "Clinical Data Interoperability Services" functionality. Part of that functionality is to use the embedded view of REDCap in the EHR to add a patient to a REDCap project and access their data in the project. Another part of that functionality is to utilize Clinical Data Pull specifically. "Clinical Data Pull" refers to when REDCap makes a web request to the EHR/FHIR server to export a patient's data to import it into a REDCap project. CDP can be used in the EHR-embedded view of REDCap or it can be used when accessing REDCap normally inside a web browser (i.e., outside the context of the EHR). When making FHIR requests to the EHR, REDCap will use the FHIR access token stored for the user in the REDCap database. And if the token has expired, it will refresh the token via the FHIR token refresh method.

The Clinical Data Pull module is a project-level module that a REDCap administrator must enable for a given REDCap project. Once enabled, a user with specific user privileges in the project will be able to go to the CDP Field Mapping page, and can map their fields in their REDCap project to fields in the EHR. Once the mapping has been completed, clinical data can then be pulled from the EHR by entering the medical record number of a patient into the mapped "MRN" field on a REDCap data entry form in the project. Once the MRN is entered, clinical data will be fetched in real time from the EHR via the FHIR web service. The patient data retrieved will be stored in a temporary holding area in REDCap's database and will be encrypted (i.e., encrypted at rest). Then in the REDCap user interface, it will display to the user that data has been retrieved from the EHR, and will prompt the user to open an adjudication screen where they can view the clinical data and verify the data before clicking a Save button, which will then officially import the data into their REDCap project.

Diagram 1: EHR launch sequence + SMART on FHIR authorization process using OAuth 2

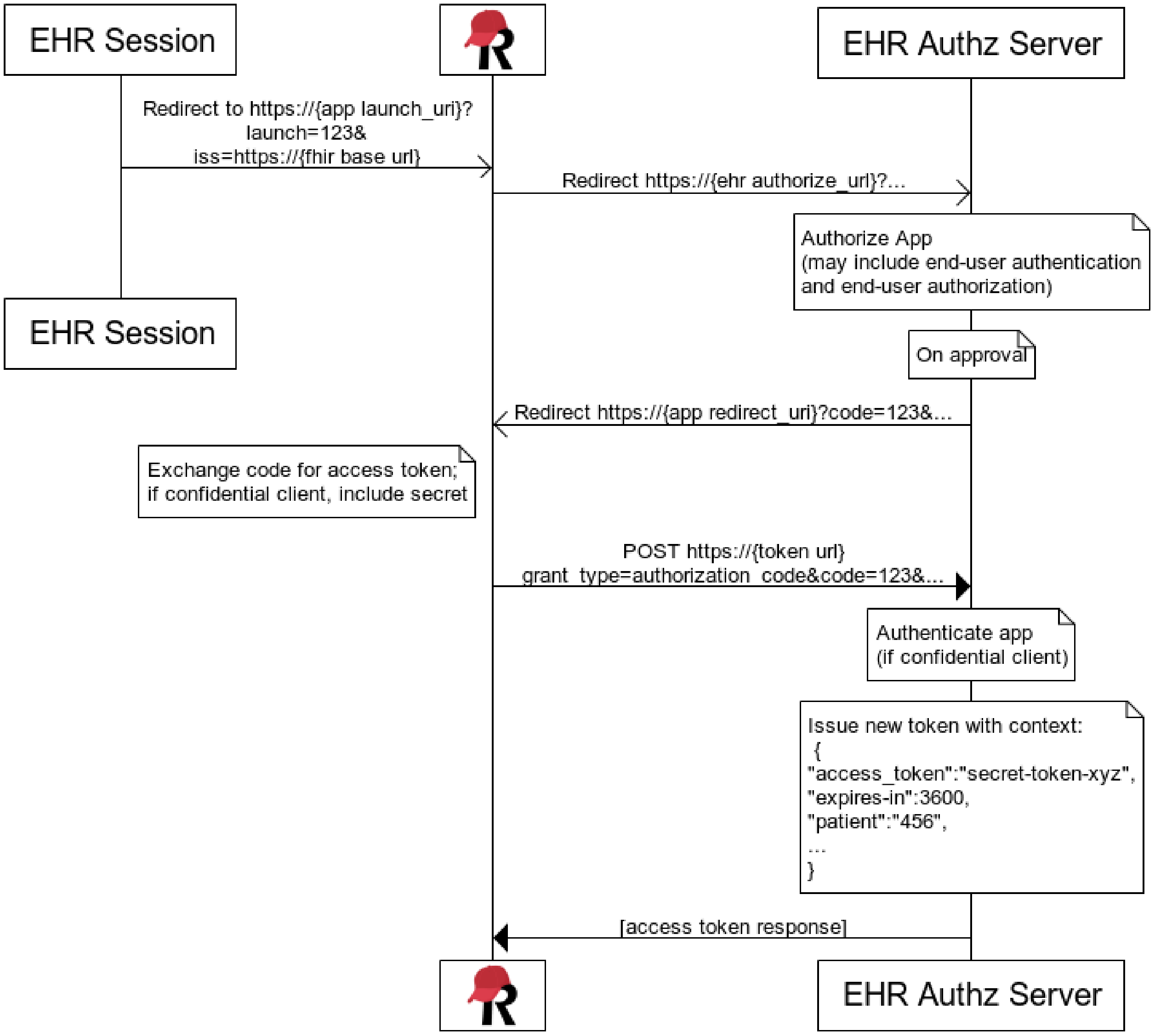
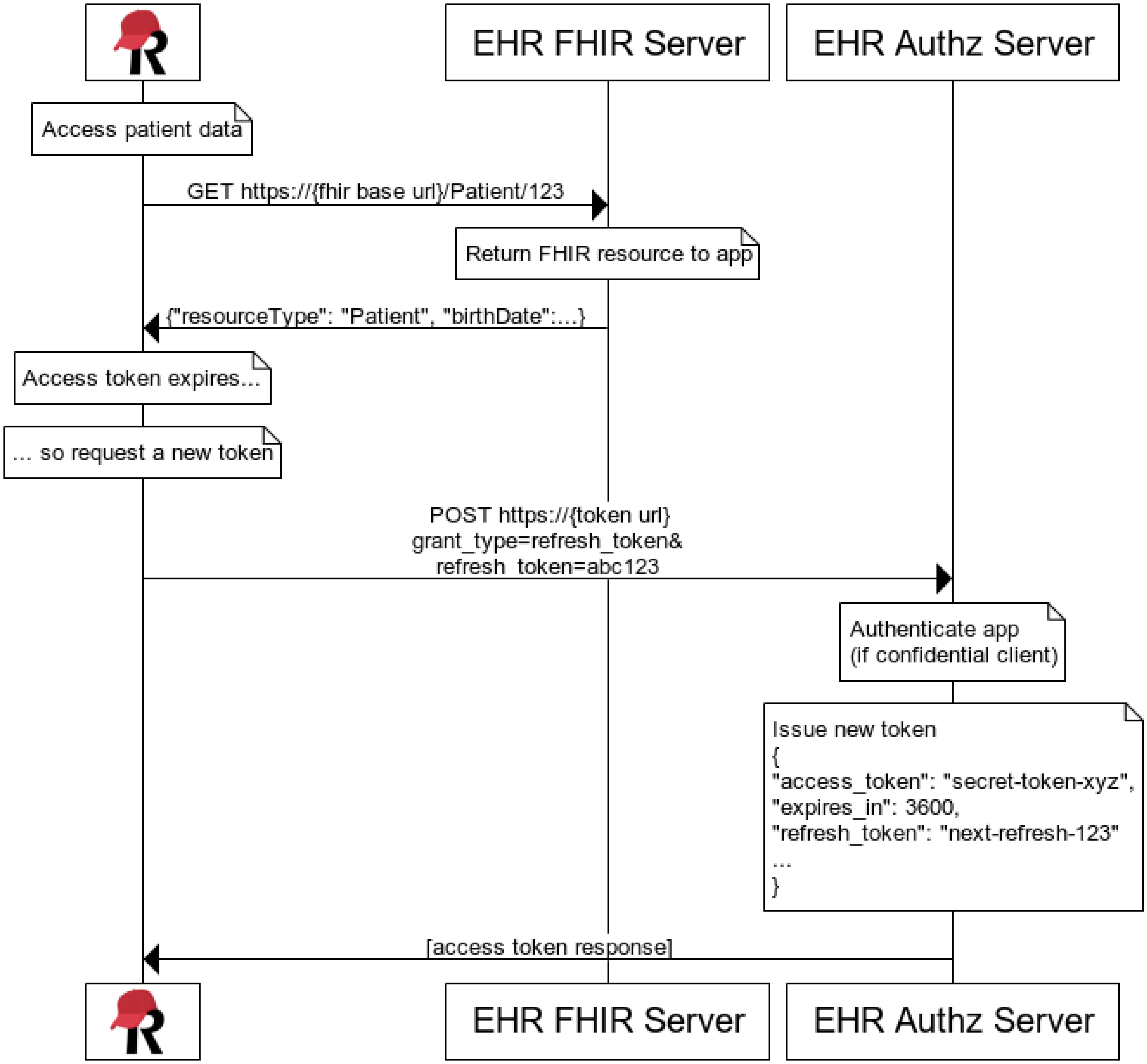

EHR Session
EHR Authz Server
Redirect to https://{app launch_uri}?
launch=123&
iss=https://{fhir base url}
Redirect https://{ehr authorize_url}?...
Authorize App
(may include end-user authentication
and end-user authorization)
EHR Session
On approval
Redirect https://{app redirect_uri}?code=123&...
Exchange code for access token;
if confidential client, include secret
POST https://{token url}
grant_type=authorization_code&code=123&...
Authenticate app
(if confidential client)
Issue new token with context:
{
"access_token":"secret-token-xyz",
"expires-in":3600,
"patient":"456",
...
}
[access token response]
EHR Authz Server

Diagram 2: Data retrieval from EHR via SMART on FHIR + Refresh access token


EHR FHIR Server
EHR Authz Server
Access patient data
GET https://{fhir base url}/Patient/123
Return FHIR resource to app
{"resourceType": "Patient", "birthDate":...}
Access token expires...
... so request a new token
POST https://{token url}
grant_type=refresh_token&
refresh_token=abc123
Authenticate app
(if confidential client)
Issue new token
{
"access_token": "secret-token-xyz",
"expires_in": 3600,
"refresh_token": "next-refresh-123"
...
}
[access token response]
EHR FHIR Server
EHR Authz Server