

Coverage Requirement Discovery (CRD) API User Guide



Prepared By:
Acentra Health
1600 Tysons Boulevard, Suite 1000
McLean, VA 22102



evoBrix X v2026-06
July 08, 2026

Table of Contents

Preface..... 1

Purpose and Scope..... 1

Document Conventions 1

Audience Requirements 1

Chapter 1 – Chapter 1 – Overview..... 2

 1.1 Data Flow 3

Chapter 2 – Chapter 2 – Coverage Requirement Discovery (CRD) API 4

 2.1 Overview 4

 2.1.1 Components of Coverage Requirement Discovery (CRD) API..... 5

 2.1.2 Processing of CDS Hooks 5

 2.1.3 CDS Hooks Response and CDS Service Feedback 7

 2.1.4 CRD API Documentation 7

 2.2 FHIR Server Sandbox Environment..... 8

 2.3 Accessing the Sandbox Environment 9

 2.4 Base URLs 9

 2.5 Capability Statement 10

 2.6 Supported Scopes..... 10

 2.7 Resources..... 10

 2.8 Request Headers 11

Chapter 3 – Chapter 3 –Connecting to Coverage Requirement Discovery API 12

Chapter 4 – Chapter 4 – Third-Party App Registration 13

 4.1 Acentra Sandbox Environment..... 13

 4.2 Environments 13

Chapter 5 – Chapter 5 – Glossary..... 14

Chapter 6 – Chapter 6 – Acronyms..... 15

Tables

Table 1. Coverage Requirements Discovery (CRD) API Solution Components..... 2

Table 2: CDS Hooks 5

Table 3: Online Documentation - CRD API 8

Table 4: Base URLs for Base URLs for CRD API 9

Table 5: Resources and URLs for and URLs for CRD API..... 10

Table 6: Feedback URLs for CRD API..... 11

Table 7: Request Headers 11

Table 8: Glossary..... 14

Table 9: Acronyms..... 15

Figures

Figure 1. Acentra Interoperability Solution- CRD API..... 2

Figure 2: Acentra Sandbox Environment – Online Developer Documentation..... 9

Figure 3: Third-Party App Registration Process 13

Preface

Purpose and Scope

The CRD (Coverage Requirements Discovery) API allows EHR/EMR systems to initiate discovery requests to the registered CRD Server through supported CDS Hooks to determine the coverage requirement discovery information such as list of covered items and services, Prior authorization requirements and documentation requirements. The CDS Server then responds back to the CRD Clients with appropriate CDS Cards or System actions providing Clinical Decision Support (CDS) to EHR/EMR systems.

This document describes Acentra's Interoperability Solution that supports the new CMS mandate.

Document Conventions

This document uses the following conventions:

- The term "you" refers to the application owners who use this document.
- Hyperlink names appear in bold fonts.

Audience Requirements

This document is intended for third-party application owners or developers who need to know how to connect their applications to Acentra's Coverage Requirements Discovery (CRD) API, and for Payers implementing the CRD API to provide data to third-party applications via the API.

Chapter 1 – Chapter 1 – Overview

The Acentra Interoperability Solution is engineered to support the business and technical requirements defined in the CMS mandate (CMS Final Rule 0057 F). The solution leverages Da Vinci - Coverage Requirements Discovery 2.2.0 - STU 2.2 as anchors in conjunction with data exchange standards, such as HL7® Fast Healthcare Interoperability Resources (FHIR®), United States Core Data and Interoperability (USCDI), and CMS recommended Implementation Guides (IGs).

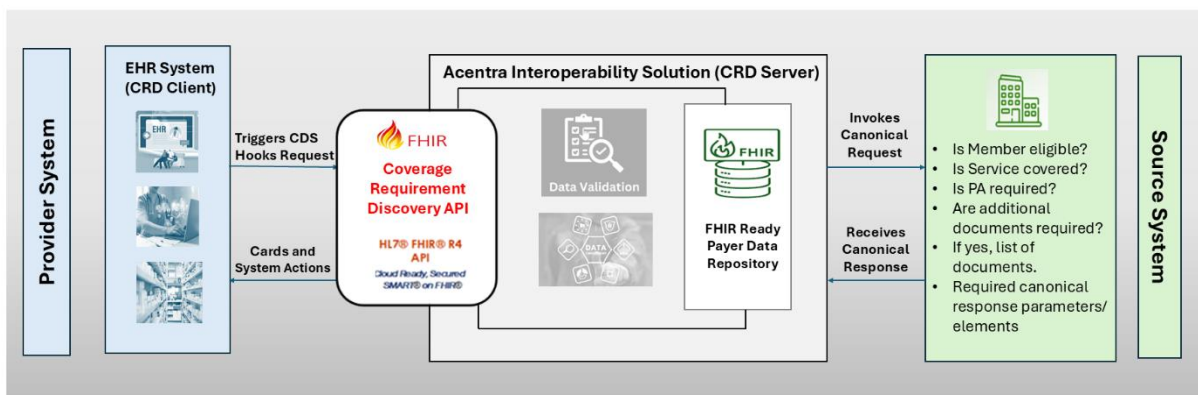


Figure 1. Acentra Interoperability Solution- CRD API

The table below describes the features of the components of the Acentra Interoperability Solution.

Table 1. Coverage Requirements Discovery (CRD) API Solution Components

Component	Features
FHIR Server	Implements a RESTful FHIR API service, which is a CDS Hooks-compliant System that supports applicable profiles in Primary Hooks and Secondary Hooks, and related frameworks based on Da Vinci - Coverage Requirements Discovery Implementation Guide and aligned with CMS-0057-F Final Rule. Performs four levels of validation on the request payload received from the CRD Client. This includes JSON structure validation, mandatory elements and conditionally required elements validation, validating that the coded elements adhere to the allowed code sets defined in the Implementation Guide (IG) and prefetched data validation. Once these validations are complete, the FHIR Server generates a canonical request structure and invokes internal source System. The FHIR Server, then receives the canonical response from Source System, prepares a FHIR Compliant Response enabling seamless interoperability across Systems.

Component	Features
	Enables transformation of request payload from FHIR Server to Canonical request structure and Canonical response to FHIR compliant Response and responds back to CRD Client.
Data Repository	All request data elements and response data elements are stored in JSON format on the Audit table in FPDR for tracking and auditing purposes. FHIR Ready Payer Data Repository (FPDR) enables inter-related models for storing and retrieving data via logical views for generating Responses on coverage discovery and Prior authorization (PA) determination.
FHIR Adapter	Facilitates the retrieval of data from FPDR and act as integration layer between the FHIR Server and FPDR.
SMART on FHIR Module	Secures the Coverage Requirements Discovery (CRD) API by implementing the CMS-recommended standalone authorization sequence based on the OAuth 2.0 standard. CRD Client (EHR System) connecting to the FHIR server must authenticate through a manual process over secure TLS channels. The CRD API integrates with SMART® on FHIR to obtain an access token via OAuth 2.0 authorization. The CRD Client must include a signed JWT in every hook request sent to the CRD Server. The registered and authorized CRD Client (EHR System) is permitted to invoke the CDS Hooks request to perform coverage discovery and Prior authorization (PA) determination.

1.1 Data Flow

The CRD Client initiates discovery requests to the registered CRD Server, triggered by real-time clinical workflow events through CDS Hooks. Upon receiving a successful discovery response from the CRD Server, the CRD Client initiates a CDS Hooks request with the request payload. The CRD Server performs four levels of validation on the request payload received from the CRD Client. The FHIR server, based on the validated request data, generates a canonical request structure and invokes internal source System web services to perform coverage discovery and prior authorization determination. The FHIR Server, then receives the canonical response from Source System, prepares a FHIR Compliant Response and returns appropriate CDS Cards or System actions to CRD Clients, providing Clinical Decision Support (CDS) to EHR/EMR systems.

Chapter 2 – Chapter 2 – Coverage Requirement Discovery (CRD) API

2.1 Overview

Coverage Requirement Discovery (CRD) API facilitates the secure movement of coverage requirement information data from Payer Systems (Producers) to authorized Providers (Consumers) accessible via authorized third-party consumer applications. It leverages the CMS-recommended Da Vinci - Coverage Requirements Discovery Implementation Guide (IG) in conjunction with data exchange standards, such as HL7 FHIR, United States Core Data and Interoperability (USCDI), and SMART on FHIR Application Launch Framework (Standalone Sequence).

Acentra's Interoperability Solution ensures that the data in transit and at rest is encrypted. To ensure authorized access, EHR System connecting with the FHIR server must authenticate through a manual process over secure TLS channels. The CRD API integrates with SMART® on FHIR to obtain an access token via OAuth 2.0 authorization. The registered and authorized CRD Client must include a signed JWT to invoke the CDS Hooks request to perform coverage discovery and Prior authorization (PA) determination.

2.1.1 Components of Coverage Requirement Discovery (CRD) API

CRD Client:

A CRD Client is an EHR System that initiates discovery requests to the registered CRD Server, triggered by real-time clinical workflow events through CDS Hooks.

CRD Server:

CRD Server is a CDS Hooks-compliant System that generates canonical structure and invokes Source Systems to retrieve coverage requirement discovery information. Upon receiving canonical responses from the source Systems, the CRD Server maps and transforms the data into CDS Hooks-compliant FHIR responses and returns them to CRD clients. CDS Hooks 2.0 APIs supported by the CRD Server are Discovery API, Service API and Feedback API.

CDS Hooks:

The CDS Hooks enables CDS Clients to call CDS Services (FHIR) at specific points in the clinical workflow (called hooks) and receive response from CDS service.

2.1.2 Processing of CDS Hooks

Types of CDS Hooks:

There are 2 types of CDS Hooks- Primary Hooks and Secondary Hooks.

Primary Hooks:

The 'Primary' hooks are “Appointment Book”, “Orders Sign”, and “Order Dispatch”. CRD Servers must, at minimum, return a “Card” and “Coverage Information” System action for these hooks as a response,

Secondary Hooks:

The 'Secondary' hooks are “**Orders Select**”, “**Encounter Start**”, and “**Encounter Discharge**”. CRD Servers can **return “Cards”** to these hooks but are not mandated to do so.

CRD Client can trigger the Primary or Secondary hooks by using any of the resources in the supported profiles.

Table 2: CDS Hooks

S. No	CDS Hooks	Description
1	Primary Hooks	Appointment Book Order Sign Order Dispatch

2	Secondary Hooks	Orders Select Encounter Start Encounter Discharge
---	-----------------	---

Note: In our implementation, The CRD server supports both the Primary and Secondary hooks, when a CDS request is received from a CRD Client.

Processing of Hooks:

Once the CRD Server receives a CDS Hook request from the CRD Client, four levels of validation are performed in CDS Hook request such as,

Level 1 – Syntax Integrity

Level 2 – FHIR Structural Requirements

Level 3 – Inter-Segment (Relational) Validation

Level 4 – Code Sets validation

The CRD Server validates the CDS Hook request initiated by the CDS Client by performing the following sequence of validations before further processing.

- **JSON structure validation** to ensure compliance with CDS Hooks 2.0 specification.
- **Element validations** to ensure the presence of mandatory data elements (required elements in each FHIR resource), conditionally required elements based on parent element and inter segment relational elements.
- Validation of the **Coded elements** with required conformance that adhere to the allowed code sets defined in the Implementation Guide.
- Validating the presence of **Prefetched data** and canonical business identifiers for each resource type.
- If a request fails to meet the validations, the System must return an Operation Outcome response with relevant diagnostic messages and halt further processing.
- Upon successful validation, The FHIR System generates a canonical request structure by mapping the FHIR parameters into their corresponding Canonical parameters and then invokes the **Internal Source System** with the Canonical request payload to initiate the Coverage Discovery and Prior Authorization (PA) determination process.

2.1.3 CDS Hooks Response and CDS Service Feedback

Once a canonical response is received from the internal Source System (e.g., evoBrix X), the CRD Server extracts the available data elements from the Canonical Response and maps them to a CDS Hooks compliant Response. The CRD Server constructs a CDS Hooks response containing one or more Cards, which are returned to the CDS Client to support real-time clinical decision-making. The FHIR System must implement the Coverage Information response type, since supporting coverage discovery is the primary objective of the CRD workflow.

Cards:

- Cards are the primary response structure in CDS Hooks. Each card is a structured JSON object that communicates a message, guidance, or interactive action to the end-user.

System Action:

- The System Actions should be returned along with the CDS Cards in **Primary hooks** when the CDS Service determines that the service has coverage-related information, requirements for Prior Authorization, any additional information required for prior authorization processing, thereby instructing the EHR to apply a coverage-information extension to the submitted resource (ServiceRequest, MedicationRequest, etc.).
- **“Coverage-assertion-ID”** generated within FHIR will be included in that extension that serves as a tracking ID. The FHIR System will populate the cards.source.label through configuration based on Source Systems. The CRD services must return responses for all supported hooks within the defined SLA time of **5 seconds**.

CDS Service Feedback Endpoint:

The CDS Server must expose a feedback endpoint as required by the CDS Hooks specification, allowing the CDS Client to optionally send feedback. The CRD Client can accept one or more suggestions, if the response was received within the defined SLA time of 5 seconds or the entire card can either be ignored or overridden (with “overrideReasons” or without reason).

Following actions can be performed on the card:

- Suggestion Accepted
- Card Overridden (Without Reason)
- Card Overridden (With Reason)
- Card Ignored

Audit Logging must be done for traceability purposes that include the details of validated request/invalid request/ response constructed and returned/ feedback on the card/action.

2.1.4 CRD API Documentation

The table below provides links to the online documentation for the CRD API.

Table 3: Online Documentation - CRD API

Function	Online Documentation
IG References	CRD API IG CRD IG Home Page - Da Vinci - Coverage Requirements Discovery v2.2.0-ballot CDS Hooks https://cds-hooks.hl7.org/2.0/ Coverage information extension IG Coverage Information - Da Vinci - Coverage Requirements Discovery v2.2.0-ballot
CDS Hooks- Primary Hooks and Secondary Hooks	CRD Appointment with Order CRD Appointment without Order CRD Encounter CRD Communication Request CRD Device Request CRD Medication Request CRD Nutrition Order CRD Service Request CRD Vision Prescription
Error Codes	 FHIR-Error%20Codes _CRD%20API.xlsx

2.2 FHIR Server Sandbox Environment

Acentra's Interoperability Solution's Sandbox Environment provides a synthetic but reliable dataset of applicable profiles in CRD API to help facilitate testing of the CRD API. The Sandbox dataset represents real data and ensures consistency across the dataset.

Detailed information about Acentra's Interoperability Solution Sandbox Environment can be found at <https://sandbox.mhbapp.com/index.html>. The site includes information for contacting Acentra to register your application.

As shown in Figure 3, the Acentra Sandbox FHIR server documentation portal provides all the required information for the third-party app developers to register their application and connect and access the Sandbox FHIR server for CRD API.

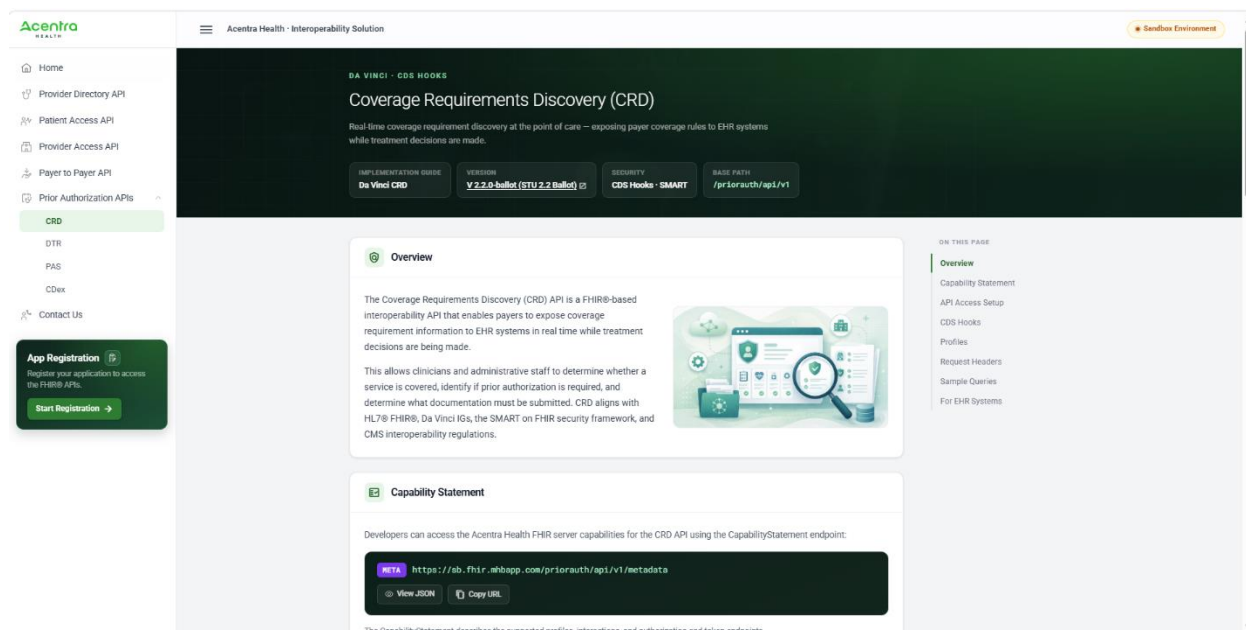


Figure 2: Acentra Sandbox Environment – Online Developer Documentation

2.3 Accessing the Sandbox Environment

Before you can access the CRD API, you will need to register your application with Acentra's Sandbox Environment. Please refer to the Sandbox FHIR server documentation information at <https://sandbox.mhbapp.com/index.html>.

2.4 Base URLs

Three types of FHIR server URLs are provided below for CRD API.

- Acentra Sandbox
- UAT
- Production

The table below provides the base URLs for the FHIR server for the CRD API. Please note that the base URLs are not accessible via browser and are meant to be accessed programmatically.

Table 4: Base URLs for Base URLs for CRD API

Environment	Base URL CRD API
Acentra Sandbox	https://sb.fhir.mhbapp.com/priorauth/api/v1/
UAT	<a href="https://<<tenant-id>>.np.fhir.mhbapp.com/priorauth/api/v1/">https://<<tenant-id>>.np.fhir.mhbapp.com/priorauth/api/v1/
Production	<a href="https://<<tenant-id>>.fhir.mhbapp.com/priorauth/api/v1/">https://<<tenant-id>>.fhir.mhbapp.com/priorauth/api/v1/

2.5 Capability Statement

The capability statement describes the set of capabilities (behaviors) implemented by the FHIR Server; for example, it describes the enabled profiles, interactions, and provides information about the authorization and token endpoints.

The FHIR server capabilities for CRD API can be accessed using the Capability Statement endpoint: https://<<base_url_priorauth>>/metadata.

Please refer to Table 4 for the base URLs for CRD API. For example - the sandbox environment Capability URL will be: <https://sb.fhir.mhbapp.com/priorauth/api/v1/metadata>.

2.6 Supported Scopes

The FHIR Authorization Server supports the following scopes to enable secure, backend system-to-system access for CRD API.

- system/Appointment.read
- system/CommunicationRequest.read
- system/DeviceRequest.read
- system/Encounter.read
- system/MedicationRequest.read
- system/NutritionOrder.read
- system/ServiceRequest.read
- system/VisionPrescription.read
- system/OperationOutcome.read

2.7 Resources

The Acentra Interoperability Solution's Coverage Requirement Discovery (CRD) API supports the resources listed in Table 5.

Table 5: Resources and URLs for and URLs for CRD API

Resources	URL	Method Type
Discovery URL	https://{base_url}/cds-services	GET
Appointment-Hook	https://{base_url}/cds-services/appointment-book-crd	POST
Order Select	https://{base_url}/cds-services/ order-select-crd	POST
Order-Sign	https://{base_url}/cds-services/ order-sign-crd	POST

order-Dispatch	https://{base_url}/cds-services/ order-dispatch-crd	POST
Encounter-Start	https://{base_url}/cds-services/ encounter-start-crd	POST
Encounter-Discharge	https://{base_url}/cds-services/ encounter-discharge-crd	POST

Table 6: Feedback URLs for CRD API

Resources	URL	Method Type
Appointment-Hook	https://{base_url}/cds-services/appointment-book-crd/feedback	POST
Order Select	https://{base_url}/cds-services/ order-select-crd/feedback	POST
Order-Sign	https://{base_url}/cds-services/ order-sign-crd/feedback	POST
Order-Dispatch	https://{base_url}/cds-services/ order-dispatch-crd/feedback	POST
Encounter-Start	https://{base_url}/cds-services/ encounter-start-crd/feedback	POST
Encounter-Discharge	https://{base_url}/cds-services/ encounter-discharge-crd/feedback	POST

2.8 Request Headers

All requests for Prior Authorization (CRD) API resources require the headers outlined in Table 7.

Table 7: Request Headers

Headers	Values
content-type	application/json
accept	application/json
authorization	Bearer <<auth_code>>

Chapter 3 – Chapter 3 –Connecting to Coverage Requirement Discovery API

- The CDS Client must register with the CRD Server through a manual process over **TLS** to enable a secure connection.
- This ensures only authorized CDS Clients can connect to the CDS Service.
- Once the secure channel is established via TLS, the CDS Client sends a GET request to Discovery Endpoint to identify which CDS Hooks are supported by the CRD Server.
- A formal Service Agreement must be set up between CRD Client and CRD Server. This agreement includes:
 - Which hooks are supported (Agreement on which hooks will be triggered and responded to).
 - **JWT** public key exchange (for token verification).
- After the manual setup is complete, the CDS Client includes a signed JWT in every hook request to the CRD Server.
- The CRD Server must validate the JWT signature using the CRD Client's public key.

Note: Authentication is assumed as done in our implementation. Provider's EHR/EMR System onboarding is done through **SMART on FHIR** and background services

CDS Service Discovery Endpoint:

Once the Secure channel is established via TLS, A CDS Service should be discoverable via a stable **Discovery Endpoint** by CDS Clients that includes information such as a description of the CDS Service, when it should be invoked, and any data that is requested to be prefetched.

To generate the CDS recommendations, CDS Services require advanced access to specific FHIR resources which can be achieved through the following methods - **Prefetch** using predefined prefetch templates. As a result, the CDS Service receives the required contextual FHIR resources as part of the CDS request, eliminating the need for additional callbacks to the EHR Systems.

Chapter 4 – Chapter 4 – Third-Party App Registration

4.1 Acentra Sandbox Environment

The Acentra Sandbox Environment is available for third-party app owners to perform connectivity and testing. Third-party app owners can approach Acentra directly or via payer organization to register their application. If the payer organization approves the request, then Acentra will provide access. In cases where the third-party app owner approaches Acentra directly, it will be Acentra's discretion to approve or deny the request.

4.2 Environments

In the Acentra Interoperability Solution, the intake process collects the registration and attestation information from the third-party app owner and sends it to the authorized personnel within the payer organization (or Tenant). The payer organization reviews the information provided by Acentra to conduct third-party app verification.

If approved, the payer organization notifies Acentra, which then issues credentials (client-id/secret) to third-party application owners for the appropriate environment: Production. If the application registration is not approved, then the payer organization notifies the application owner directly and conveys the decision to Acentra of the outcome. A graphical representation of this process is illustrated in Figure 2.

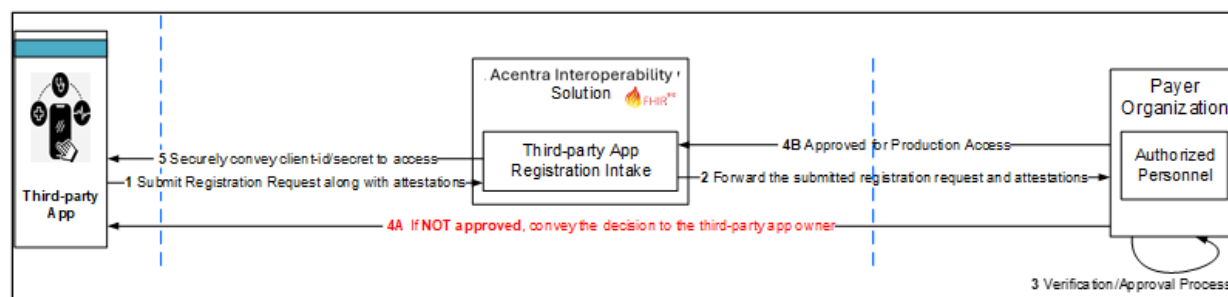


Figure 3: Third-Party App Registration Process

Chapter 5 – Chapter 5 – Glossary

This section provides definitions for document-specific terms in the following format: Entries are in alphabetical order, and items beginning with numbers will come first.

Table 8: Glossary

Term	Definition
Application Programming Interface	An interface that defines interactions between multiple software applications.
Acentra Interoperability Solution	A solution created to support the business and technical requirements defined in the CMS mandate, CMS Final Rule 0057-F.
Payer Organization	A business entity that adjudicates and renders payment for health care claims, including pharmacy claims. Examples include Medicare, state Medicaid programs, and commercial insurance carriers. In this document, "Tenant" is also used to indicate the Payer Organization in relation to the environment.
Sandbox	An environment containing hardware, instrumentation, simulators, software tools, and other support. Also known as the "test bed".
Tenant	See "Payer Organization."
Third-Party App	An application or program that is neither Acentra nor the Washington State Health Care Authority, that has authorized access via the API for the purpose of delivering data to authorized users (Providers).

Chapter 6 – Chapter 6 – Acronyms

This section provides definitions for document-specific terms in the following format: Entries are listed in alphabetical order, and items beginning with numbers will come first.

Table 9: Acronyms

Acronym	Definition
API	Application Programming Interface
CMS	Centers for Medicare & Medicaid Services
CDS	Clinical Decision Support
CRD	Coverage Requirement Discovery
EHR	Electronic Health record
FHIR	Fast Healthcare Interoperability Resources
FPDR	FHIR Ready Payer Data Repository
ID	Identification
IG	Implementation Guide
JSON	JavaScript Object Notation
JWT	JSON Web Token
HIPAA	Health Insurance Portability and Accountability Act
HTTP	HyperText Transfer Protocol
TLS	Transport Layer Security
REST	Representational State Transfer
SLA	Service Level Agreement
SMART	Substitutable Medical Applications and Reusable Technologies
STU	Standard for Trial Use
URL	Uniform Resource Locator
USCDI	United States Core Data and Interoperability