

Prior Authorization Support (PAS) API User Guide



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



evoBrix X v2026-08
August 31, 2026

Table of Contents

Purpose and Scope 1

Document Conventions 1

Audience Requirements 1

Chapter 1 – Overview 2

 1.1 Data Flow 3

Chapter 2 – Prior Authorization Support API 4

 2.1 PA Submission flow Overview 4

 2.1.1 PA Submission Request 4

 2.1.2 Pre-Signed URL Mechanism.....5

 2.1.3 Process PA Request.....8

 2.1.4 Process PA Response to EHR Systems 10

 2.1.5 Prior Authorization Support API Documentation (Submission flow) 10

 2.2 PA Inquiry flow Overview.....11

 2.2.1 Processing of PA Inquiry Request.....11

 2.2.2 Process PA Inquiry Response to EHR Systems.....12

 2.2.3 Prior Authorization Support API Documentation (Inquiry flow) 12

 2.3 FHIR Server Sandbox Environment.....13

 2.4 Accessing the Sandbox Environment 14

 2.5 Capability Statement 15

 2.6 Supported Scopes.....15

 2.7 Resources 16

 2.8 Request Headers 16

Chapter 3 –Connecting to the Prior Authorization Support.....17

Chapter 4 – Third-Party App Registration 18

 4.1 Acentra Sandbox Environment.....18

 4.2 Environments 18

Chapter 5 – Glossary 19

Chapter 6 – Acronyms 20

Tables

Table 1. Prior Authorization Support API Solution Components..... 2

Table 2. Validation Level..... 8

Table 3. Attachment Processing 9

Table 4. Online Documentation – Prior Authorization Support API (Submission flow)..... 10

Table 5. Online Documentation – Prior Authorization Support API (Inquiry flow) 13

Table 6. Base URLs for Prior Authorization Support API..... 14

Table 7. Resources and URLs for Prior Authorization Support API 16

Table 8. Request Headers 16

Table 9. Glossary..... 19

Table 10. Acronyms..... 20

Figures

Figure 1. Acentra Interoperability Solution 2

Figure 2 Pre-Signed URL Flow 6

Figure 3. Acentra Sandbox Environment – Online Developer Documentation..... 14

Figure 4. Third-Party App Registration Process 18

Preface

Purpose and Scope

The CMS Interoperability and Prior Authorization Final Rule (CMS-0057F) require CMS-regulated payers to implement and maintain a Prior Authorization Support API that enables direct submission and management of Prior Authorization requests from EHR/EMR Systems, including checking the status of previously submitted requests.

This rule emphasizes that Prior Authorization Support APIs must be able to communicate whether the Source System approves the prior authorization request (and the date or circumstance under which the authorization ends), denies the prior authorization request (and a specific reason for the denial), or requests more information.

This document describes Acentra Health 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 Health Prior Authorization Support API, and for Clinical Systems (Providers/EHRs/Practice Management Systems etc.) implementing the API to retrieve data from Payors via the API.

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 Prior Authorization Support (PAS) FHIR IG (v2.2.0: STU 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).

The diagram provides the overall Acentra Interoperability Solution.

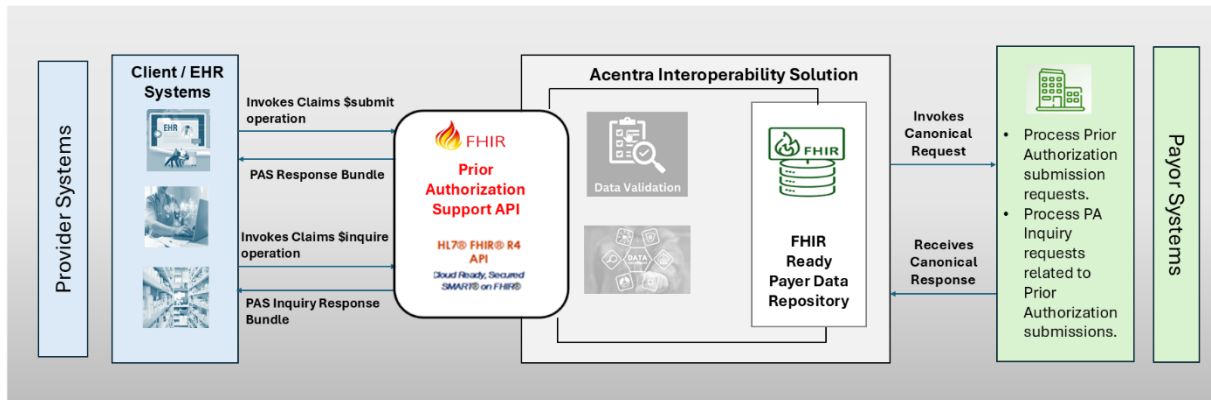


Figure 1. Acentra Interoperability Solution

The table below describes each component of the Acentra Interoperability Solution.

Table 1. Prior Authorization Support API Solution Components

Component	Features
FHIR Server	Implements a RESTful FHIR API service that supports applicable PAS profiles in PAS Request Bundle and PAS Response Bundle for the PAS Submission flow as well as PAS Inquiry Request Bundle and PAS Inquiry Response Bundle for the PAS Inquiry flow. It is built in accordance with the Da Vinci PAS Implementation Guide (v2.2) and aligned with CMS-0057-F Final Rule. The FHIR Server performs five levels of validation on the request payload in accordance with the Implementation Guide (IG). Upon successful validation, the validated data is stored in the FHIR-Ready Payer Data Repository (FPDR), facilitating the transformation of canonical requests and responses to and from FHIR Server. Subsequently, the FHIR Server constructs a FHIR-compliant response, ensuring standardized data exchange and seamless interoperability across integrated systems.
Data Repository	FHIR Ready Payer Data Repository (FPDR) enables inter-related models for storing and retrieving data via logical views for Prior Authorization Request Submission and Tracking information. For PAS submission flows, the combination of claim type and payer ID, and for PAS inquiry flows, the combination of PA tracking number and member ID stored in the FPDR, drive

Component	Features
	the routing logic to determine and invoke the corresponding source System–specific web service endpoints.
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 Prior Authorization Service API by implementing the CMS-recommended standalone launch authorization sequence based on the OAuth 2.0 standard. EHR system connecting to the FHIR server must authenticate through a manual process over secure TLS channels. The PAS API integrates with SMART® on FHIR to obtain an access token via OAuth 2.0 authorization. The registered and authorized EHR system is permitted to invoke the Claim/\$submit operation for Prior Authorization submission and the Claims/\$inquire operation for PA Inquiry requests.

1.1 Data Flow

Submission Flow

As part of the PAS request submission data flow, patient demographic details, provider information, claim details, and supporting documentation are received from EHR/EMR Systems and packaged into a PAS Request Bundle. These requests are securely transmitted to Acentra Health Interoperability Solution and imported into the FHIR-ready Payer Data Repository (FPDR) where five levels of validation are performed in accordance with the PAS API Implementation Guide (IG). The FHIR server, based on the validated request data, generates a canonical request and routes the request to the appropriate source-specific web service based on the routing configuration. The FHIR Server, then receives the canonical Response from the Source System, converts into PAS Response bundle in a FHIR Compliant way and communicates back to the EHR/EMR Systems.

Inquiry Flow

As part of the PAS inquiry data flow, the EHR/EMR Systems initiates a request to check the status of an existing prior authorization. The inquiry includes key identifiers such as patient details, provider information, and the original claim reference, packaged into a PAS Inquiry Bundle. This request is securely transmitted to Acentra Health Interoperability Solution, where four levels of validation are performed in accordance with the PAS API Implementation Guide (IG).

The FHIR server, based on the validated request data, generates a canonical request and routes the request to the appropriate source-specific web service based on the routing configuration. The FHIR Server, then receives the canonical Response from the Source System which holds the status of the prior authorization —such as Approved, Denied, Partially Approved, or Pending. The response is converted into a PAS Inquiry Response Bundle in FHIR Compliant way and communicates back to the EHR/EMR Systems.

Chapter 2 – Prior Authorization Support API

The CMS Interoperability and Prior Authorization Final Rule mandates that impacted payers implement and maintain a Prior Authorization API to streamline and automate the prior authorization process facilitating the electronic exchange of prior authorization requests and decisions directly from Client Systems.

2.1 PA Submission flow Overview

Prior Authorization Support (PAS) API enables the EHR/EMR Systems to submit and payer systems to receive prior authorization requests using HL7® FHIR® standards. To initiate a prior authorization request, the Client System invokes the Claims \$submit operation which includes the applicable input profiles packaged within PAS Request bundle, which is then transmitted to the FHIR server for validation, processing, and subsequent response generation.



For PAS submissions, it is recommended to use Base64-encoded attachments with a combined content size of less than 10 MB.

2.1.1 PA Submission Request

Client Systems can initiate a Prior Authorization Support (PAS) submission by invoking the “Claims \$submit operation”. Any supporting attachments included in the submission must be provided as Base64-encoded content using the PAS DocumentReference Profile.



Acentra supports both the versions of PAS IG, i.e., 2.1.0 and 2.2.0 ballot. Though 2.1.0 has the provision to send externally referenced attachments in attachment.url, Acentra supports the attachments only in attachment.data.

Attachment Submission Methods

The EHR/EMR System can submit attachments through the DocumentReference.attachment.data element using either of the following methods:

1. **Inline Submission** – The document content is Base64-encoded and included directly in DocumentReference.attachment.data.
2. **Pre-Signed URL-Based Submission** – The document is uploaded to a secure external storage location, and the corresponding pre-signed URL is Base64-encoded and submitted in DocumentReference.attachment.data. This mechanism is available based on system configuration and is intended for attachments that cannot be submitted inline due to payload-size limitations.

Sending Attachments Using the Pre-Signed URL Mechanism

The PAS Implementation Guide (IG) STU 2.2.0 Ballot requires attachment information to be included within the PAS DocumentReference resource. The applicable PAS profile does not directly support substituting the required inline document content with an external URL.

At the same time, the AWS API Gateway imposes a maximum request payload of 10 MB, which limits the size of documents that can be included as Base64-encoded inline content within a PAS request.

To support attachments that exceed the inline payload limit, Acentra provides a configurable Pre-Signed URL Mechanism for secure large-document submission.



In both methods, the EHR/EMR System populates DocumentReference.attachment.data. The contentType identifies whether the Base64-encoded value represents the actual document content or a URI referencing the document.

2.1.2 Pre-Signed URL Mechanism

The Pre-Signed URL Mechanism enables document externalization by allowing EHR/EMR Systems to upload attachments directly to FHIR platform-managed Amazon S3 storage using secure, time-limited AWS S3 pre-signed URLs.

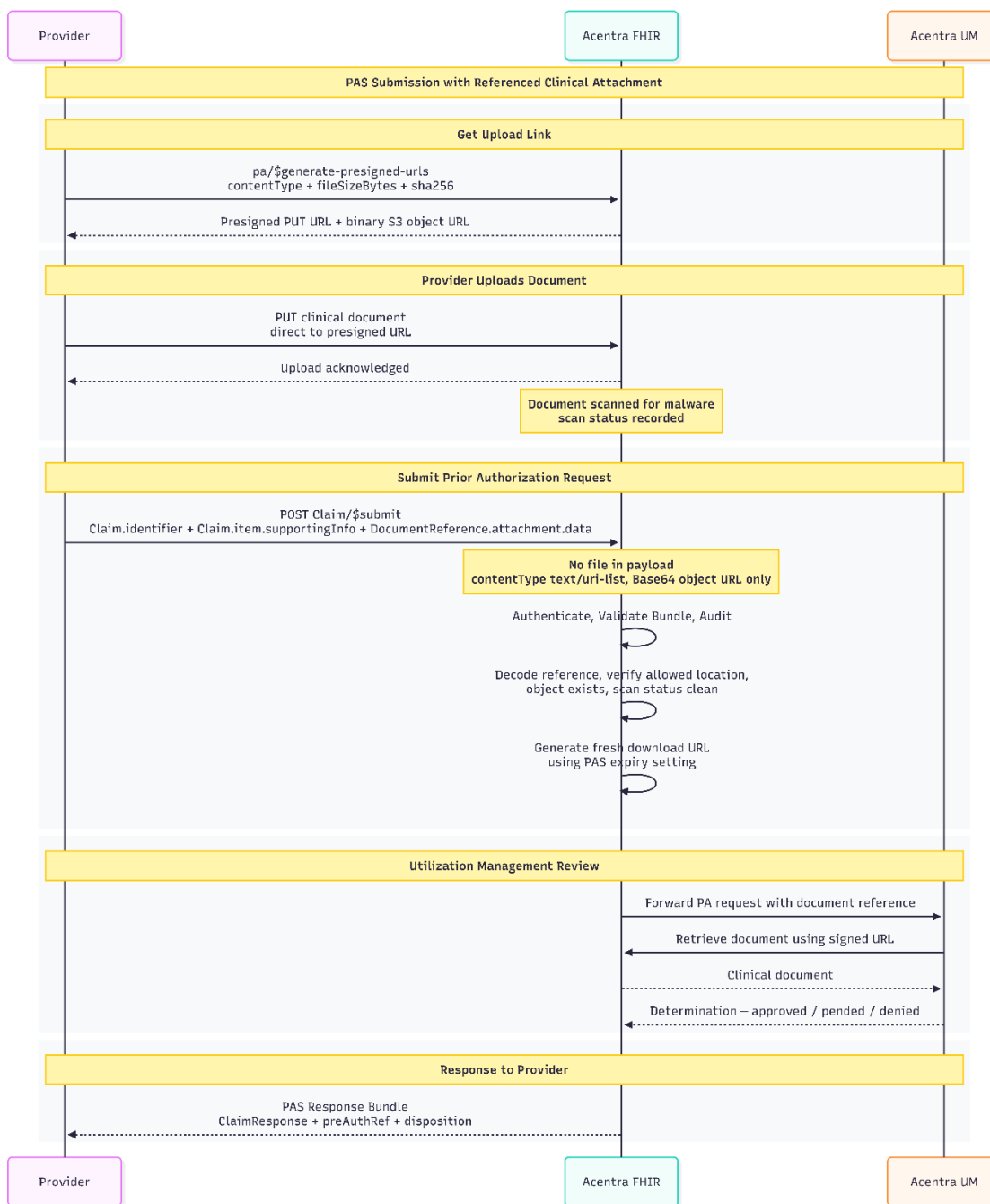


Figure 2 Pre-Signed URL Flow

Instead of embedding the document content within the FHIR request, the EHR/EMR system uploads the document to the designated S3 location. The FHIR platform provides a stable, platform-issued FHIR Binary handle that is referenced in the *DocumentReference.attachment.data* to identify and access the uploaded document.

The functional flow is as follows:

1. Authenticate with the FHIR Platform

The EHR/EMR System authenticates using the existing SMART on FHIR (SoF) authentication process and obtains a bearer token.

2. Request Pre-Signed Upload URLs

For documents requiring externalized upload, the EHR/EMR System requests a pre-signed URL by invoking the Binary/\$generate-presigned-urls operation and providing the required document information, including content type, file size, and document reference identifier.

The FHIR platform validates the request and returns a time-limited pre-signed S3 upload URL and a stable FHIR Binary handle for each document.

3. Upload Documents to S3

The EHR/EMR System uploads the document directly to the designated FHIR platform-managed S3 storage using the pre-signed URL, avoiding the need to include the complete document content in the PAS request. The document is transferred securely over TLS and does not pass through the FHIR API Gateway. After uploading, the FHIR platform initiates an asynchronous virus scan of the document.

In both methods, the EHR/EMR System populates `DocumentReference.attachment.data`. The `contentType` identifies whether the Base64-encoded value represents the actual document content or a URI referencing the document.



Virus scanning is performed asynchronously outside the FHIR request and transformation flow to ensure that attachment scanning does not impact FHIR processing performance.

4. Submit the FHIR Bundle

The EHR/EMR System submits the PAS FHIR Bundle using the standard submission operation.

- The EHR/EMR System Base64-encodes the applicable pre-signed URL and populates the encoded value in `DocumentReference.attachment.data`.
- For documents uploaded through the Pre-Signed URL Mechanism, `DocumentReference.attachment.data` contains the stable FHIR Binary handle, not the temporary pre-signed URL.



When populating the pre-signed URL as Base64-encoded value, the EHR system must set `DocumentReference.attachment.contentType` to `text/uri-list`. The `text/uri-list` content type indicates that the Base64-encoded value in `DocumentReference.attachment.data` represents a URI to the document rather than the actual document content.

5. Validate FHIR Submission

The FHIR platform validates the submitted Bundle for FHIR structure, applicable profile requirements, and attachment-specific rules. For externally uploaded documents, the FHIR platform also validates that the Binary reference belongs to an approved FHIR platform location.

6. Validate and Process the Uploaded Document

The FHIR platform resolves the FHIR Binary handle to the corresponding document in S3 and verifies that the document has successfully passed the virus scan. It also validates the actual document type and size against the information provided when the pre-signed URL was generated.

After successful validation, the FHIR platform retrieves the document and prepares it in the format required by the downstream Payer System.

2.1.3 Process PA Request

When the EHR/EMR System invokes the \$submit operation, the FHIR System validates the request and attachments before sending them to the configured Source System.

Request Validation

The FHIR System performs the following levels of validation:

Table 2. Validation Level

Level	Validation	Description
Level 1	Syntax Integrity	Verifies that the request is valid JSON and can be processed as a FHIR request.
Level 2	FHIR Structure	Verifies that the request conforms to FHIR R4 and applicable profile requirements, including required elements, cardinality, and data types.
Level 3	Relational Validation	Verifies relationships and dependencies between elements and referenced resources.
Level 4	Code Set Validation	Verifies that coded elements use the allowed values defined by the applicable Implementation Guide.
Level 5	Attachment Validation	Verifies attachment file type, file size, number of files, and applicable attachment configuration.

The request proceeds only when all required validations are successful.



For PAS submissions, use Base64-encoded attachments with a total content size of less than 10 MB to ensure efficient processing and support the 15-second target response time.

Pre-Signed URL Validation

When *DocumentReference.attachment.contentType* is set to: **text/uri-list**,

the FHIR Server identifies the attachment as a URL-based document reference.

The system:

1. Decodes the Base64-encoded value provided in *DocumentReference.attachment.data*.
2. Validates that the decoded value represents a permitted Amazon S3 object reference.
3. Verifies that the referenced document is available at the expected storage location.
4. Processes the document according to the configured attachment delivery method.



Uploaded documents are subject to the platform's virus-scanning process. Only documents that successfully pass the virus scan are eligible for downstream processing.

Attachment Processing

After receiving the attachment, the FHIR System processes it according to the configured delivery method for the Source System.

Table 3. Attachment Processing

Attachment Received From EHR	Configured Delivery Method	Processing
Inline Base64	Base64	The document is validated and delivered to the Payer System as a Base64-encoded attachment.
Inline Base64	URL	The document is stored securely and delivered to the Payer System using a pre-signed download URL.
Pre-Signed URL (contentType = text/uri-list)	URL	The document is retrieved and delivered to the Payer System as a pre-signed S3 download URL.

2.1.4 Process PA Response to EHR Systems

Once the Source system has completed the PA Processing, it returns a Canonical Response to the FHIR server. The FHIR server then transforms this Canonical response into a FHIR compliant Response represented as a single Bundle containing a PAS Claim Response (and or referenced resources) conformant to the “PAS Response Bundle” profile that includes the details relevant to the prior authorization decision. If the prior authorization Bundle cannot be processed due to validation errors or other non-business-errors, it generates an Operation Outcome with Status. For synchronous operations, PAS response is returned, within the defined SLA time of 15 seconds.

Audit Logging is done for traceability purposes that includes the details of successfully validated request/invalid request/ Routing configuration failure/ response constructed and returned/ error responses from source System.

2.1.5 Prior Authorization Support API Documentation (Submission flow)

The table below provides links to the Prior Authorization Support API (Submission flow) online documentation.

Table 4. Online Documentation – Prior Authorization Support API (Submission flow)

Function	Online Documentation
IG References	Prior Authorization Support (PAS) IG Prior Authorization Implementation Guide Home Page - Da Vinci Prior Authorization Support (PAS) FHIR IG v2.2.0-ballot PAS Request Bundle PAS Request Bundle - Da Vinci Prior Authorization Support (PAS) FHIR IG v2.2.0-ballot PAS Response Bundle PAS Response Bundle - Da Vinci Prior Authorization Support (PAS) FHIR IG v2.2.0-ballot ESMD FHIR Implementation Guide 1.0.0 - esmd Generate Presigned URL API - ESMD FHIR Implementation Guide v1.0.0
PAS Request profiles	PAS Beneficiary Patient PAS Claim PAS Coverage

Function	Online Documentation
	PAS Device Request PAS Document Reference PAS Encounter PAS Insurer Organization PAS Medication Request PAS Nutrition Order PAS Practitioner PAS PractitionerRole PAS Requestor Organization PAS Service Request PAS Subscriber Patient
PAS Response profiles	PAS Claim Response PAS CommunicationRequest
Error Codes	 FHIR-Error%20Codes _PAS%20API.xlsx

2.2 PA Inquiry flow Overview

The prior authorization inquiry operation enables the EHR/EMR Systems to inquire about prior authorization submissions including checking the status of requests and performing general inquiries. To initiate a prior authorization inquiry request, the Client System invokes the Claims \$inquire operation which includes the applicable input profiles packaged within a PAS Inquiry Request bundle. It is then transmitted to the FHIR server for validation, processing, and subsequent response generation.

2.2.1 Processing of PA Inquiry Request

Once the Client Systems invoke the claims \$inquire operation, the request payload is transmitted to the FHIR Server. Upon receiving the request, the FHIR Server performs four levels of validation on the request payload by performing the following sequence of validations before further processing.

Level 1 – Syntax Integrity

Level 2 – FHIR Structural Requirements

Level 3 – Inter-Segment (Relational) Validation

Level 4 – Code Sets validation

- JSON structure validation to ensure the payload is syntactically correct and conforms to FHIR R4 standards.
- Conformance and Relational validations to ensure the presence of mandatory data elements based on cardinality, conditionally required elements based on parent elements and relational validation between parameters within request profiles and referenced request profiles.
- Validation of the Coded elements with required conformance that adhere to the allowed code sets defined in the Implementation Guide
- The FHIR Server validates that the combination of the PA Tracking Number (from the PAS Claim Inquiry Profile) and the Member ID (from the referenced Patient profile) exist in ETL tables. This validation is performed against the records obtained from the PAS Response Bundle during the Claim \$submit operation and corresponding Payer ID will be identified. If a matching source system record is found, the Routing Configuration logic determines the appropriate source system–specific web service endpoint to which the request payload for inquiry can be routed for further processing.

If a request fails to meet the validations or no matching source system record is found, the system returns an Operation Outcome response with relevant diagnostic messages and halts further processing. Upon successful validation, the FHIR Server transforms the request into a canonical request structure by mapping the FHIR parameters to their corresponding canonical parameters. It then invokes the appropriate source system–specific web service, based on configured routing logic, to initiate Prior Authorization (PA) Inquiry Process.

2.2.2 Process PA Inquiry Response to EHR Systems

Once the Source system has completed the PA Inquiry Processing, it returns a Canonical Response to the FHIR server. The FHIR server then transforms this Canonical response into a FHIR compliant Response represented as a single Bundle containing a “PAS Claim Inquiry Response” (and other referenced resources) conformant to the “PAS Inquiry Response Bundle” profile that includes the details relevant to the prior authorization inquiry response. If the prior authorization Bundle cannot be processed due to validation errors or other non-business-errors, it generates an Operation Outcome with Status.

Audit Logging is done for traceability purposes that includes the details of successfully validated request/invalid request/ Routing configuration failure/ response constructed and returned/ error responses from source System.

2.2.3 Prior Authorization Support API Documentation (Inquiry flow)

The table below provides links to Prior Authorization Support API (Inquiry flow) online documentation.

Table 5. Online Documentation – Prior Authorization Support API (Inquiry flow)

Function	Online Documentation
IG References	Prior Authorization Support (PAS) IG Prior Authorization Implementation Guide Home Page - Da Vinci Prior Authorization Support (PAS) FHIR IG v2.2.0-ballot PAS Inquiry Request Bundle PAS Inquiry Request Bundle PAS Inquiry Response Bundle PAS Inquiry Response Bundle
PAS Inquiry Request profiles	PAS Claim Inquiry PAS Beneficiary Patient PAS Coverage PAS Insurer Organization PAS Practitioner PAS PractitionerRole PAS Requestor Organization PAS Subscriber Patient
PAS Inquiry Response profiles	PAS Claim Inquiry Response PAS CommunicationRequest
Error Codes	 FHIR-Error%20Codes _PAS%20API.xlsx

2.3 FHIR Server Sandbox Environment

Acentra's Interoperability Solution's Sandbox Environment provides a synthetic but reliable dataset of applicable profiles in PAS API to help facilitate testing of the Prior Authorization Support 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 2, the Acentra Sandbox FHIR server documentation portal provides all the required information for the third-party app developers to register their app and connect and access the Sandbox FHIR server for Prior Authorization Support API.

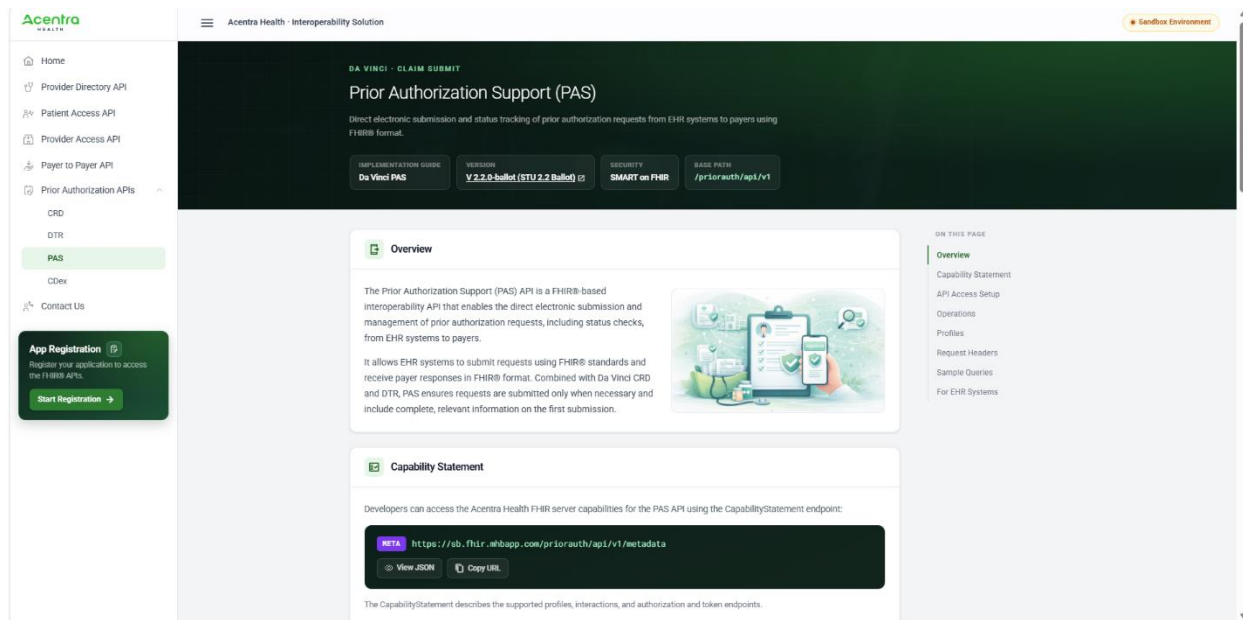


Figure 3. Acentra Sandbox Environment – Online Developer Documentation

2.4 Accessing the Sandbox Environment

Before you can access Prior Authorization Support API, you will need to register your application with Acentra's Sandbox Environment.



Refer to the Sandbox FHIR server documentation information at <https://sandbox.mhbapp.com/index.html>.

Base URLs

Three types of FHIR server URLs are provided below for Prior Authorization Support.

- Acentra Sandbox
- UAT
- Production

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

Table 6. Base URLs for Prior Authorization Support API

Environment	Base URL Prior Authorization Support 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/

Environment	Base URL Prior Authorization Support API
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 CDex API can be accessed using the Capability Statement endpoint: https://<<base_url_priorauth>>/metadata.

Please refer to Table 4 for the base URLs for Prior Authorization Support 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 PAS API.

- system/Patient.read
- system/Claim.read
- system/Claim.create
- system/ClaimResponse.read
- system/CommunicationRequest.read
- system/Coverage.read
- system/DeviceRequest.read
- system/DocumentReference.read
- system/DocumentReference.create
- system/Organization.read
- system/MedicationRequest.read
- system/NutritionOrder.read
- system/Practitioner.read
- system/PractitionerRole.read
- system/ServiceRequest.read

- system/Encounter.read
- system/OperationOutcome.read

2.7 Resources

The Acentra Interoperability Solution's Prior Authorization Support API supports the resources listed in the table below.

Table 7. Resources and URLs for Prior Authorization Support API

Resources	URL
PA-Submission	https://{base_url}/Claim/\$submit
PA-Inquiry	https://{base_url}/Claim/\$inquire
Generate Presigned URL	https://{base_url}/\$generate-presigned-urls
Regenerate Presigned URL	https://{base_url}/\$regenerate-presigned-urls

2.8 Request Headers

All requests for Prior Authorization Support Resources require the headers outlined in the table below.

Table 8. Request Headers

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

Chapter 3 –Connecting to the Prior Authorization Support

- EHR/EMR Systems must be registered and approved via the App Registration Portal and FHIR Admin Portal.
- Access is secured using the SMART Backend Services Authorization Framework for system-to-system integration.
- All communications must be established over TLS to ensure encrypted data transmission.
- Authentication is performed using OAuth 2.0 and JWT-based mechanisms.
- Only authorized applications can access FHIR APIs based on defined scopes.
- For detailed onboarding, configuration, and authentication steps, refer to the respective user guides:
 - App Registration User Guide
 - Admin Portal User Guide
- After successful registration and authentication, the EHR/EMR Systems will know the relevant Payer's endpoint and can invoke
 - Claim/\$submit operation for PA submission
 - Claims/\$inquire operation for PA Inquiry request

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 perform third-party app verification.

If approved, the payer organization notifies Acentra, which then issues credentials (client-id) to the 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 the Figure 3.

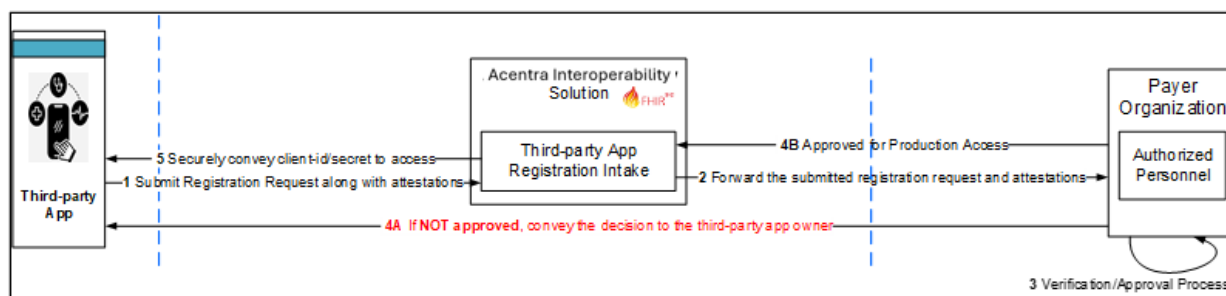


Figure 4. Third-Party App Registration Process

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 9. Glossary

Term	Definition
Application Programming Interface	An interface that defines interactions between multiple software applications.
CARIN	A non-partisan, multi-sector alliance focused on providing digital health information to consumers. Also known as CARIN Alliance.
Acentra Interoperability Solution	A solution created to support the business and technical requirements defined in the CMS mandate, CMS Final Rule 0057-F.
Source 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 Source Organization in relation to the environment.
Sandbox	An environment containing hardware, instrumentation, simulators, software tools, and other support. Also known as a "test bed".
Tenant	See "Source 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 (EHR/EMR Systems).

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 10. Acronyms

Acronym	Definition
API	Application Programming Interface
CDex	Clinical Data Exchange
CMS	Centers for Medicare & Medicaid Services
EDI	Electronic Data Interchange
EHR	Electronic Health Record
EOB	Explanation of Benefits
FHIR	Fast Healthcare Interoperability Resources
FPDR	FHIR Ready Payer Data Repository
ID	Identification
IG	Implementation Guide
HIPAA	Health Insurance Portability and Accountability Act
OIDC	Open ID Connect
PAS	Prior Authorization Support
REST	Representational State Transfer
SAML	Security Assertion Markup Language
SMART	Substitutable Medical Applications and Reusable Technologies
STU	Standard for Trial Use
URL	Uniform Resource Locator
USCDI	United States Core Data and Interoperability
PDex	Payer Data Exchange