28 INTEGRATION CHAPTERS / ONE IMPLEMENTATION REFERENCE

The Healthcare Integration Bible

What to connect. How to build it. How to prove it works.

From EHRs and pharmacy to payer APIs, cloud FHIR stores and AI workflows. A practical guide to the decisions, failure modes and evidence behind dependable connected healthcare.

Original AI-generated editorial illustration. No real patients or customer deployments depicted.

START HERE

Read for the decision you need to make.

This is a field reference, not a requirement to read seventy pages before starting a conversation.

Buying an integration	Read the decision map (p. 5), your integration chapter and the scoping worksheet (p. 67). Ask for the named deliverables before comparing proposals.
Designing the system	Read the shared architecture (p. 6), the relevant protocol and domain chapters, then the mapping and security controls (pp. 63–64).
Preparing for go-live	Use each chapter's acceptance scenarios with the test strategy and operating model (pp. 65–66). Replace every illustrative threshold with an agreed target.

How to read each chapter

The first page explains the connection and implementation sequence. The second identifies caveats, three acceptance scenarios, a synthetic worked example and the evidence to request. These are starting test packs, not exhaustive certification suites.

Scope and limits

US-focused, with a separate India / ABDM chapter. The book covers 28 major integration topics, not every vendor endpoint or licensed transaction specification. Engineering patterns are Nirmitee recommendations. Primary references are linked in chapters and in the source index. Check current versions, partner capabilities and legal applicability before implementation.

A BOUNDARY WORTH KEEPING

This guide is not medical or legal advice, a compliance certification, a vendor endorsement or a guarantee of performance. Clinical decisions belong to qualified clinicians; legal applicability belongs to your counsel. All examples are synthetic. Do not put patient data, credentials or confidential records into the worksheet.

CHAPTER INDEX / 1 OF 2

Find your integration.

Select a chapter title in the PDF to jump directly to its implementation spread.

CHAPTER INDEX / 2 OF 2

Find your integration.

Select a chapter title in the PDF to jump directly to its implementation spread.

ARCHITECTURE BEFORE TECHNOLOGY

Choose the workflow. Then the interface.

Several standards may participate in one care journey. Select the transport and representation by the business outcome, not by the most fashionable acronym.

Live clinical events	Start with supported event interfaces: often HL7 v2 or approved event feeds. Define ordering, acknowledgment and downstream completion.
User-facing chart access	Assess FHIR and SMART, supported resources and launch context. Confirm permissions, tenant availability and actual data completeness.
Population-level data	Assess bulk exports, batch files or approved extraction. Define snapshot boundaries, incremental changes and reconciliation.
Orders, images, prescriptions	Use the domain contract: LIS workflows, DICOM/PACS routes or prescribing-network standards. A generic FHIR endpoint may not complete the workflow.
Claims and authorizations	Separate eligibility, claim submission, adjudication, remittance and prior authorization. Choose supported trading-partner contracts for each.
Longitudinal data products	Evaluate a governed FHIR store, openEHR repository or analytical platform against actual consumers. Preserve source meaning and lifecycle.
AI-assisted action	Start read-only or draft-first where appropriate. Establish evidence, narrow tool permissions and approved human review before write-back.

A USEFUL FIRST QUESTION

“Which action must happen in which system, for whom, and what evidence will prove it happened correctly?” If that sentence is unclear, an API list will not resolve the scope.

THE COMMON FOUNDATION

Connections need an operating system.

Keep partner variation at the edges. Share reliability and governance where the contracts genuinely align; do not flatten clinically meaningful differences.

01 **Partner-specific adapters** → 02 **Shared policy + processing** → 03 **Product and care workflows**

At the edge	Versioned transports, profiles, partner IDs, local code maps, endpoint capabilities and source provenance.
At the boundary	Authentication, authorization, consent/purpose checks, tenant isolation, schema and business validation.
In durable processing	Correlation, deduplication, ordering, bounded retries, quarantine and controlled replay. Document which guarantees each destination supports.
Across the lifecycle	Versioned configuration, automated fixtures, release approvals, observability, access audits and reconciliation to business outcomes.
Outside automation	Clinical judgment, ambiguous identity review, legal interpretation and financial exceptions remain with accountable people.

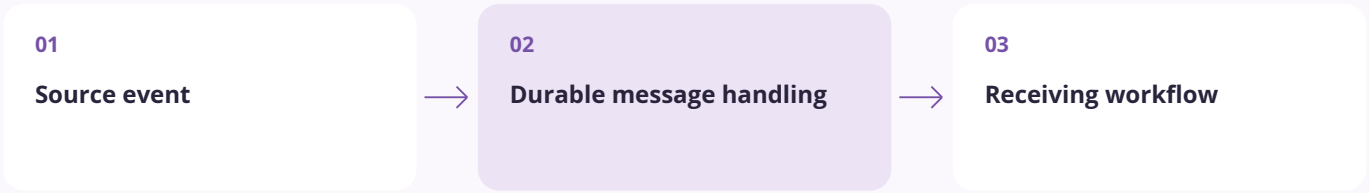
Do not promise universal adapters

A shared model reduces repetition only where semantics are compatible. New EHRs still need capability checks, mappings, onboarding and tests. Estimate those tasks explicitly; there is no defensible universal “two-week integration” guarantee.

01 / CLINICAL MESSAGING

HL7 v2 event interfaces

HL7 v2 commonly carries admissions, transfers, discharges, orders and results between clinical systems. A message is an event within a local workflow, not just a delimited record. The partner interface specification determines versions, trigger events, optional fields, local segments and acknowledgment behavior. Start with the business event and the receiving workflow rather than assuming every sender uses the same profile.



01 Agree the event contract

Inventory event types, sending facilities, identifier authorities, time zones, character encodings and field usage. Obtain representative synthetic messages for normal, corrected and cancelled events. Ask what creates each event in the source user interface and what a receiver is expected to change.

02 Build the transport boundary

Agree MLLP, a protected network route or another supported transport. Define message framing, connection recovery, timeout handling and the point at which receipt is acknowledged. Persist accepted work before a crash can lose it. Separate transport connectivity from application processing status.

03 Map and reconcile

Parse with a version-aware parser. Maintain a mapping specification for repetitions, components and local fields. Record message control IDs with sender context; do not assume global uniqueness. Reconcile source events with destination outcomes and provide a controlled replay path.

Primary reference: HL7 v2

01 / CAVEATS AND ACCEPTANCE

HL7 v2 event interfaces

What can go wrong—and what your team should be able to demonstrate.

An ACK is not a clinical outcome

Document the acknowledgment mode and what the receiver has actually accepted. A message can be durably received yet fail later business validation. Surface downstream failures separately.

Sequence changes meaning

A correction, cancellation or patient merge arriving late can invalidate an earlier event. Define ordering boundaries by patient, encounter or order rather than treating all messages as independent.

Local variation is normal

Empty, absent and explicit null values may carry different meanings. Z-segments and local code systems require agreement; silently discarding them can remove essential workflow context.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Disconnect after receipt but before acknowledgment; resend and verify one intended business action.

02. Deliver a corrected result and an out-of-order cancellation; verify the documented final state.

03. Send an unknown assigning authority and an invalid local code; confirm quarantine and actionable error reporting.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic laboratory result arrives twice after a network interruption. The transport IDs and order/result identifiers are correlated; the receiver preserves the accepted result once. A later correction updates the proper result version and remains distinguishable from the original. The operator can show both event history and current state.

Ask for: Interface specification, mapping register, ACK/retry contract, synthetic regression pack and a replay runbook.

02 / RESOURCE EXCHANGE

FHIR REST APIs and profiles

FHIR organizes exchange around typed resources and defined interactions. A server may expose only a subset of resources, operations and searches. Its CapabilityStatement and implementation guides are the starting contract, not evidence that every theoretically valid FHIR request is supported. Separate API availability, profile conformance, data completeness and permission to use the data.

01 Pin the supported contract

Record base URL, FHIR release, profiles, terminology, supported search parameters, operations and write permissions. Obtain the server capability statement and counterparty examples. Treat production, sandbox and each tenant as independently validated environments.

02 Implement safe interactions

Follow returned pagination links and constrain destinations to approved hosts before sending credentials. Use version-aware updates where supported. Decide whether conditional creates, transaction bundles or server-specific idempotency facilities meet the workflow. A batch is not an atomic transaction.

03 Validate meaning and completeness

Validate profiles but also check references, coding systems, units, dates, provenance and the fields your workflow requires. Capture OperationOutcome details without logging patient payloads. Reconcile partial responses and define when an empty search means no data versus insufficient access.

Primary reference: FHIR R4 REST API

02 / CAVEATS AND ACCEPTANCE

FHIR REST APIs and profiles

What can go wrong—and what your team should be able to demonstrate.

FHIR is not a universal data model for your product

Profiles and extensions differ. Preserve source meaning and document information loss when normalizing into a shared model.

Retries can duplicate writes

After an ambiguous timeout, establish whether the request committed before retrying. A new POST may create another resource or business action.

Search is a contract too

Server support, default limits and search behavior vary. Do not construct unsupported filters or assume a full patient chart from one resource search.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Update against an outdated version and verify conflict handling does not overwrite a concurrent change.

02. Traverse multiple pages with duplicate or updated records and verify deterministic reconciliation.

03. Return a valid resource missing a business-critical field; confirm the workflow surfaces incompleteness.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

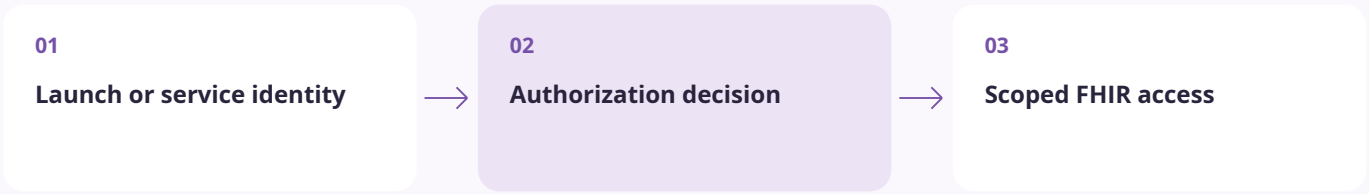
A synthetic care application needs current medication orders. Its adapter records source identifiers, status and authored dates, rather than treating every returned medication-related resource as an active prescription. A revoked permission produces an access outcome, not an empty medication list presented as clinical certainty.

Ask for: Versioned API contract, profile package, authorization matrix, mapping specification and interaction tests.

03 / AUTHORIZATION AND LAUNCH

SMART on FHIR applications

SMART on FHIR defines authorization and launch patterns for applications using FHIR. User-launched applications and backend services have different identities and trust assumptions. A successful token exchange does not mean a user may access every patient, nor that the application has the launch context its workflow expects. Design authorization as part of the clinical experience.



01 Choose the right identity model

Determine whether the application is launched from an EHR, initiated independently by a user, or operates as an approved backend service. Register the correct redirect URIs, client type and supported scopes. Validate discovery metadata instead of embedding assumptions about endpoints.

02 Protect the authorization flow

Use supported authorization-code and PKCE protections where applicable. Bind the authorization response to the originating session, validate issuer and audience expectations, and keep tokens out of URLs and logs. Store refresh credentials only where the security model permits.

03 Design expired and changed context

Provide a clear path when consent is denied, the user changes patients, context is absent or a token expires. Re-check authorization for the actual operation. Prevent a previous patient context from leaking into the next launch or browser tab.

Primary reference: SMART App Launch

03 / CAVEATS AND ACCEPTANCE

SMART on FHIR applications

What can go wrong—and what your team should be able to demonstrate.

Scope syntax and support vary by version

Pin the SMART version and granted scopes. Do not treat requested scopes as granted permissions.

Backend does not mean unrestricted

Backend access still needs a counterparty agreement, permitted population and technical authorization. Separate tenant credentials and lifecycle ownership.

Embedded browsers are a deployment concern

Cookies, pop-ups, frame restrictions and session behavior can differ from a standalone browser. Test the actual supported launch environment.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Deny consent, omit patient context and expire the token; verify safe, understandable user states.

02. Launch two synthetic patients in separate tabs and prove data and session context remain separated.

03. Rotate a service credential and revoke a client; verify recovery and access denial without stale authorization.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

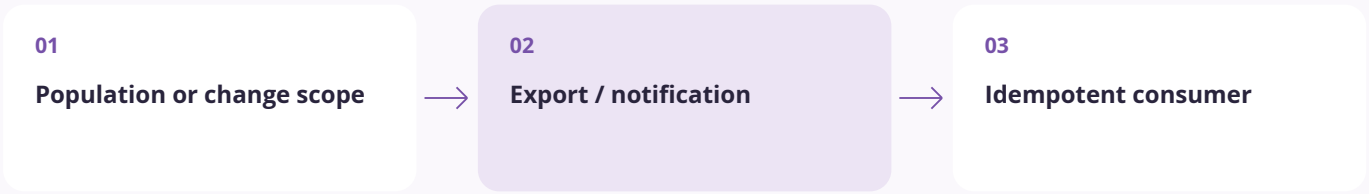
An EHR-launched application opens a patient task. It displays the current launch context before performing a write, requests only the needed access and prompts the user when the session can no longer authorize the action. A background ingestion worker uses a separately approved service identity, not the clinician’s saved token.

Ask for: Launch sequence, identity and scope matrix, client-registration checklist and negative authorization test pack.

04 / POPULATION DATA

Bulk FHIR and event-driven feeds

Population export, change polling and event notifications solve different freshness and scale needs. A bulk export is typically an asynchronous job, while an event may only tell a consumer that something changed. Neither automatically supplies a complete, ordered and exactly-once history. Agree population membership, snapshot boundaries and reconciliation before selecting the mechanism.



01 Define the population and time boundary

Specify which patients and resource types are included, how membership changes are represented and whether the extract is a snapshot or incremental view. Record the timestamp semantics and policy for late-arriving updates. Identify required deletion or restriction signals.

02 Operate jobs and file retrieval

Track job status, partial errors, file manifests and expiry. Validate content types, resource types and approved download hosts. Stream large files into controlled storage, keep credentials out of object URLs and checkpoint processing so one failure does not restart the entire population.

03 Close the gap between runs

Combine notifications or polling with a reconciliation strategy. Deduplicate by stable source identifiers and version context. Model a durable cursor, a safe overlap window and a controlled replay boundary. Confirm whether a notification carries data or requires another authorized fetch.

Primary reference: FHIR Bulk Data Access

04 / CAVEATS AND ACCEPTANCE

Bulk FHIR and event-driven feeds

What can go wrong—and what your team should be able to demonstrate.

A watermark can miss late data

Source commit time, clinical time and export time are not interchangeable. Overlap and reconcile rather than assuming perfectly ordered arrival.

Membership can change during processing

Document how additions, removals and consent changes affect an export already in progress. Do not retain unrestricted copies indefinitely.

Bulk is not a real-time clinical feed

Choose freshness objectives from the workflow. A periodic export should not be represented as a live alerting channel.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Interrupt processing midway through a file and verify restart without duplicate downstream effects.

02. Add a late update and a population removal between runs; verify the agreed reconciliation policy.

03. Expire a file URL and return a partial job error; verify operator visibility and safe retry.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

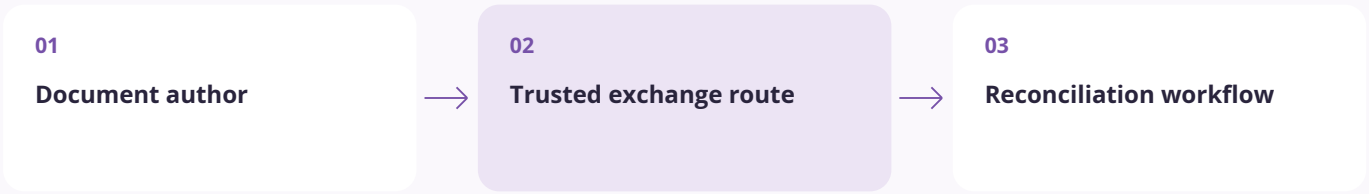
A synthetic risk-analysis cohort is exported nightly. The pipeline records job IDs and file checksums, lands immutable source files and updates the curated model idempotently. A member leaving the permitted cohort triggers the agreed retention and access process, not merely disappearance from the next file.

Ask for: Population definition, export contract, checkpoint design, change semantics and reconciliation evidence.

05 / CLINICAL DOCUMENT EXCHANGE

C-CDA, documents and Direct

Clinical documents carry a composed record with narrative, structured entries and context such as authorship and encounter. A document transport, a document registry and a clinical-data API are different layers. Direct messaging can be a route for secure document exchange, but receiving a message does not establish the quality, completeness or clinical acceptance of its attachments.



01 Agree the document package

Specify document type, template versions, required sections, narrative expectations, attachments and replacement relationships. Record patient and encounter identifiers, author information, confidentiality and creation time. Obtain representative synthetic documents with missing and amended sections.

02 Secure and validate the envelope

Approve the exchange route, trusted participants and attachment types. Apply size limits and safe XML parsing; do not resolve untrusted external entities. Verify document identifiers and structural rules before extracting content. Keep the original artifact under appropriate access and retention controls.

03 Reconcile into the destination

Preserve narrative and provenance alongside normalized data. Define how duplicates, amended documents and conflicting statements are reviewed. Avoid silently converting a historical problem or medication statement into a current active record.

Primary reference: HL7 C-CDA

05 / CAVEATS AND ACCEPTANCE

C-CDA, documents and Direct

What can go wrong—and what your team should be able to demonstrate.

Narrative and entries can disagree

A structured parser may not capture all clinically relevant context. Route discrepancies for review rather than asserting that extraction reproduces the author's meaning.

Delivery is not incorporation

Transport delivery, application receipt and clinician review are separate states. Define evidence for each state that matters.

Document replacement is not simple deletion

Preserve supersession and historical context according to policy. A revised document may amend only part of the information.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Submit the same document through two routes and verify duplicate handling preserves source provenance.

02. Process an amended document with conflicting medication status; verify review and update behavior.

03. Reject an unsafe XML construct and an oversized attachment without exposing contents in logs.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

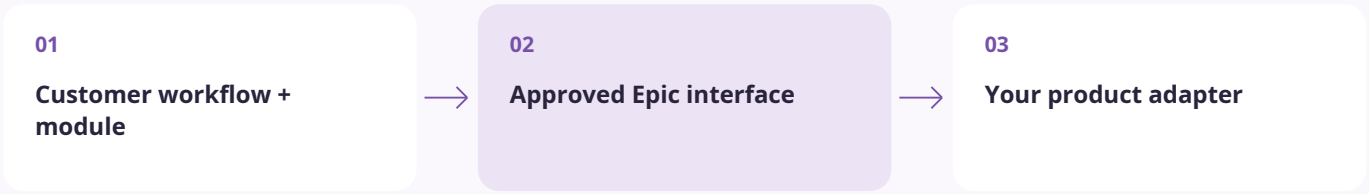
A synthetic referral includes a C-CDA document and an image report. The receiving team can see the authored narrative, parsed problems and medication statements, each tied to its source. A follow-up document is linked as a revision and does not silently create a second referral.

Ask for: Document and transport contract, safe parser configuration, reconciliation rules and revision test cases.

06 / EHR INTEGRATION

Epic: match the workflow to the interface

Epic integration begins with a specific customer workflow and its supported interface, not with “connect to Epic” as one task. Clinical APIs, event interfaces, scheduling, documents and specialty workflows have different access and implementation requirements. Module context matters: for example, cardiology workflows associated with Cupid should be scoped through the applicable interface and local operational owner, not inferred from a generic FHIR endpoint.



01 Identify the local workflow owner

Describe the user action, module context, data needed and write-back destination. Ask the customer to confirm the supported interface, installed capability and responsible application and interface teams. Separate vendor documentation from the customer’s enabled configuration.

02 Complete access and environment planning

Register the application where required and agree customer approvals, test identities, connectivity and launch context. Maintain a per-customer configuration record. Sandbox success is a technical milestone; it does not replace production access approval or workflow validation.

03 Validate the actual destination

Demonstrate how data appears to the user and what business state changes. For writes, test the allowed status transitions and audit trail. For reads, confirm population, filters and field completeness. Package local variations as explicit adapter configuration.

Primary reference: Epic interface catalog

06 / CAVEATS AND ACCEPTANCE

Epic: match the workflow to the interface

What can go wrong—and what your team should be able to demonstrate.

A module is not an API entitlement

Cupid, Radiant or another specialty context does not imply that every desired operation is exposed. Confirm the specific interface and licensing/access path with the customer.

The same brand is not the same deployment

Sites can differ in configuration, identifiers, workflows and supported versions. Do not promise a single untested connector for all hospitals.

Read access is not write access

Finding a resource in an API catalog does not establish support for creating or updating that resource in the intended workflow.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Read and, if permitted, write a synthetic scenario through the actual target environment and user workflow.

02. Repeat an ambiguous write after a timeout and prove the product avoids duplicate clinical actions.

03. Launch with different user permissions and patient contexts; verify appropriate denial and local audit visibility.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

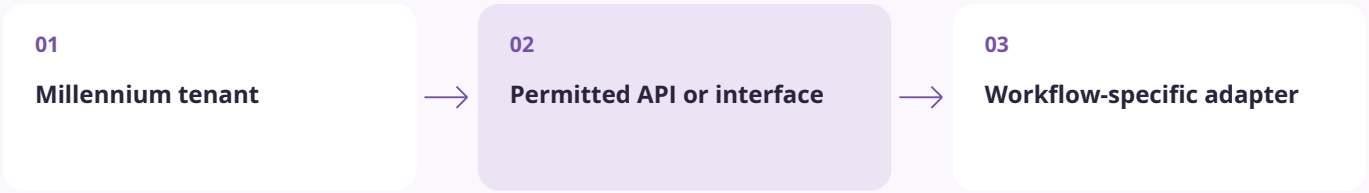
A synthetic cardiology product requires an appointment context, a procedure result and a document returned to the chart. The team scopes those as separate capabilities with distinct owners. It validates where the document appears and how it is linked, rather than equating a successful upload with completion of the cardiology workflow.

Ask for: Customer-specific interface matrix, approval dependencies, test evidence and configuration/runbook package.

07 / EHR INTEGRATION

Oracle Health Millennium

Oracle Health Millennium integration requires selection of the correct published API family and customer environment. FHIR R4, legacy endpoints and other EHR interfaces may expose different capabilities and semantics. Treat the tenant, application registration, permitted resources and workflow as a single validated integration boundary. A product name or enterprise database technology does not authorize direct database access.



01 Select and pin the API family

Document API version, supported interactions, resource profiles and endpoint environment. Identify the local source of patient, encounter and practitioner identifiers. Verify each needed field against representative responses rather than assuming parity with another vendor's FHIR implementation.

02 Establish the trust boundary

Agree application registration, credentials, customer approvals, network paths and permitted population. Test the authorization flow in the intended launch or backend mode. Keep tenant routing isolated so a configuration mistake cannot cross organizational boundaries.

03 Prove workflow behavior

Exercise the actual read or write operation and inspect the receiving workflow. Capture error behavior, paging, limits and update semantics. Build regression fixtures around the differences your product must absorb, with an explicit owner for vendor and customer changes.

Primary reference: Oracle Millennium APIs

07 / CAVEATS AND ACCEPTANCE

Oracle Health Millennium

What can go wrong—and what your team should be able to demonstrate.

Do not conflate product families

Millennium APIs, a healthcare data repository and other Oracle offerings are not interchangeable. Match documentation to the actual product and release.

Local identity is not portable

Patient and encounter references need an assigning context. Preserve source identifiers rather than treating a demographic match as a proven identity.

Data availability varies

A valid API response can still omit fields required by your workflow. Record missing-data behavior and obtain agreement on acceptable alternatives.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Configure two synthetic tenants and verify credentials and identifiers cannot cross boundaries.

02. Exercise paging, an unsupported search and an authorization failure; confirm distinct product outcomes.

03. Verify permitted writes in the receiving application, including status and encounter linkage.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

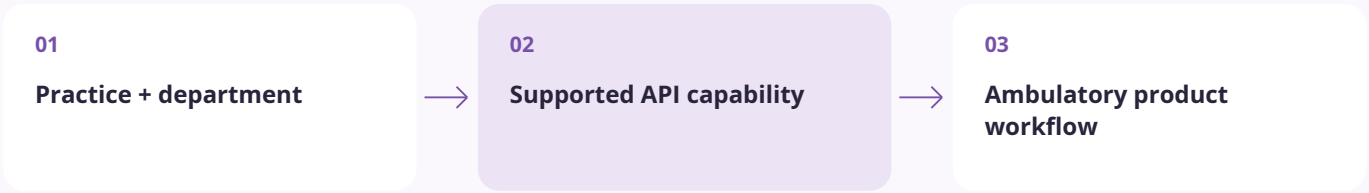
A synthetic discharge-follow-up service reads encounters and contact details from a customer environment. It checks encounter status and source timestamps before creating a follow-up task. Missing permission is surfaced as an integration issue, not treated as evidence that no discharge occurred.

Ask for: Pinned API/environment matrix, identity mapping, authorization tests and customer-specific workflow evidence.

08 / EHR INTEGRATION

athenahealth and ambulatory workflows

Ambulatory integration often spans scheduling, registration, clinical documentation and billing. athenahealth offers developer resources for different API capabilities; select the interface appropriate to the product and workflow. Practice and department context can matter as much as the patient identifier. Scope operational actions separately from read-only clinical-data access.



01 Map the business workflow

Identify whether the product needs appointment availability, patient registration, encounter context, documents or billing information. Record practice, department and provider identifiers and the owner of each local configuration. Define who corrects a record when a source value is wrong.

02 Confirm access and resource behavior

Review the current developer documentation and partner/customer requirements for each capability. Validate authentication, pagination, limits and supported mutation behavior. Do not infer that certified FHIR access exposes all operational functions available through other APIs.

03 Design state synchronization

Choose how appointment changes, document additions and encounter updates reach the product. Establish conflict handling when staff edit the same record. Maintain reconciliation and do not rely only on a successful synchronous response.

Primary reference: athenahealth developer portal

08 / CAVEATS AND ACCEPTANCE

athenahealth and ambulatory workflows

What can go wrong—and what your team should be able to demonstrate.

Patient matching needs context

Practice-level identifiers and organization configuration can affect lookup behavior. Avoid accidental duplicate registration when a search does not return a match.

Scheduling is a state machine

Availability, booking, cancellation, rescheduling and no-show behavior are distinct actions. Race conditions can invalidate a previously displayed slot.

Document upload is not completed charting

Confirm document category, association, visibility and user review. A stored attachment may not satisfy the intended documentation workflow.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Book a synthetic slot while another actor changes availability; verify a clear conflict path.

02. Cancel and reschedule an appointment and reconcile the final state across systems.

03. Upload a permitted document and verify practice, patient, encounter and document-type linkage.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic patient-engagement service reminds a patient about an appointment. A reschedule arrives after the reminder was queued. The integration re-checks the current appointment state and prevents the obsolete notification, while retaining enough audit context to explain why it was suppressed.

Ask for: Practice/department configuration, workflow transition map, permitted-API inventory and synchronization tests.

09 / MULTI-EHR DELIVERY

MEDITECH, eClinicalWorks and NextGen

A multi-EHR product needs a capability matrix per vendor, product family, release and customer. MEDITECH Greenfield provides an API integration entry point for Expanse. eClinicalWorks and NextGen integrations also require their own current developer and customer-supported routes. Do not claim functional parity from the fact that several systems expose FHIR. Design the product to represent unavailable capabilities honestly.

01 **Per-vendor contract** → 02 **Versioned adapter** → 03 **Shared product capability**

01 Inventory each target separately

For MEDITECH, establish the relevant Expanse/Greenfield path and customer scope. For eClinicalWorks, confirm the specific product, published API program and local authorization. For NextGen, distinguish the ambulatory EHR/interface route from Mirth integration-engine work. Record evidence rather than assuming a shared commercial arrangement.

02 Define a product capability model

Describe needed capabilities such as patient lookup, results, appointments and document return. Mark each as supported, conditionally supported, unsupported or unverified for each target. Keep vendor-specific semantics in the adapter without erasing data provenance.

03 Qualify every deployment

Use one common scenario set plus vendor-specific edge cases. Record environment, version, supported fields and local configuration. Introduce adapters incrementally and maintain a change-review process when a customer or vendor upgrades.

Primary reference: MEDITECH Greenfield

09 / CAVEATS AND ACCEPTANCE

MEDITECH, eClinicalWorks and NextGen

What can go wrong—and what your team should be able to demonstrate.

A generic adapter can hide important loss

If one vendor supplies encounter context and another does not, the product must not invent equivalent context. Make limitations visible to product and operations.

Do not reuse another customer's approval

Technical access and commercial rights are deployment-specific. Treat sandbox credentials and production onboarding as separate assets.

Brand names hide product differences

An EHR product, interface engine and data platform from the same company are not a single integration surface.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Run the same synthetic scenario across each adapter and compare meaning, not only JSON shape.

02. Disable a capability and verify the product communicates the limitation without a misleading empty state.

03. Change a vendor configuration and verify regression tests identify the affected workflows.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

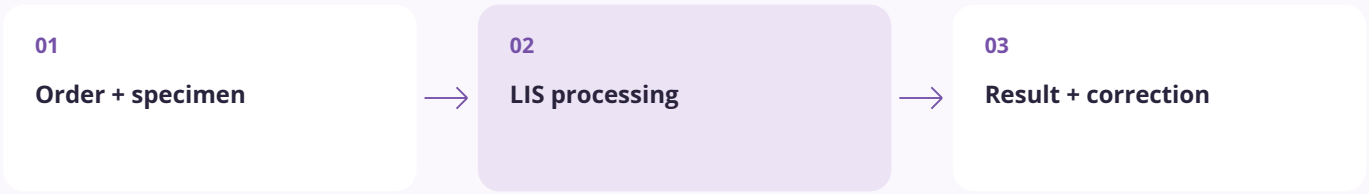
A synthetic clinical product supports result display across three EHRs but can return documents through only two approved routes. Its capability record controls the UI and contract scope. Sales and implementation teams can see the difference, preventing a read-only connection from being sold as bidirectional chart integration.

Ask for: Per-product capability matrix, adapter contracts, local prerequisites and cross-vendor regression evidence.

10 / DIAGNOSTIC WORKFLOW

Laboratory and LIS integration

Laboratory integration is a chain from order intent through specimen handling to preliminary, final and corrected results. The laboratory information system (LIS), analyzer middleware and ordering EHR may each own different identifiers. A result value without specimen, method, unit, status or reference context can be clinically misleading. Agree the entire order/result lifecycle.



01 Build a catalog and identity crosswalk

Map local orderable tests, panels and result components. Preserve specimen and accession identifiers alongside placer/filler order references. Use terminology mappings reviewed by laboratory experts; a similar display name is not sufficient evidence that two tests are equivalent.

02 Model the lifecycle

Specify order acceptance, specimen collection, rejection, cancellation, preliminary result, final result and correction handling. Establish how panel completion is represented and how result components link to the order. Include units, reference ranges, abnormal flags and relevant dates.

03 Validate downstream presentation

Inspect the actual clinical display and alerting boundary. Keep source status and corrected history visible. Reconcile missing components and unmatched results. Agree who resolves code-map changes and who handles critical-result communication outside the interface.

Primary reference: LOINC laboratory orders

10 / CAVEATS AND ACCEPTANCE

Laboratory and LIS integration

What can go wrong—and what your team should be able to demonstrate.

Unit conversion needs clinical review

Do not convert values because units look compatible. Confirm analyte, method, scale and conversion validity; retain original values and units where appropriate.

Critical-result delivery is a workflow

A message receipt does not prove that the responsible clinician was notified. Scope acknowledgment and escalation separately.

Panels are not fixed forever

Catalog changes can add, remove or rename components. Version mappings and test partial panels and corrected components.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Receive an unmatched accession and verify quarantine rather than attaching it to a guessed patient.

02. Send preliminary, final and corrected values for one synthetic component; verify status and history.

03. Change a local test code or unit and confirm that unreviewed mappings cannot silently enter clinical use.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic metabolic panel arrives in parts. The application retains each component's status and collection context, then recognizes completion according to the agreed contract. A later correction affects one component and does not create an unrelated second panel or erase the original provenance.

Ask for: Catalog crosswalk, lifecycle map, specimen/order identity rules and clinical display acceptance evidence.

11 / DIAGNOSTIC WORKFLOW

Imaging, DICOM and radiology

Imaging integration spans scheduling and orders, modality worklists, image storage, reports and viewer access. DICOM objects and DICOMweb services address imaging exchange; a report document is not the image study itself. A radiology information system, PACS, archive and EHR may each hold part of the workflow. Define which system owns study identity, report status and access decisions.

01 Map the imaging journey

Identify the order, accession, patient, study, series and instance identifiers and the authority behind each. Document modality, transfer syntax, storage and viewer requirements. Confirm worklist and report interfaces separately from image retrieval.

02 Implement the approved exchange

Choose supported DICOM services or DICOMweb query, retrieve and store operations as appropriate. Validate object and metadata consistency, transport security and archive authorization. Limit retrieval to intended studies and handle large payloads without loading entire examinations into application memory.

03 Prove clinical access and amendments

Test viewer context, report availability, amended reports and patient-identity corrections. Reconcile received instances against expected study content. Establish whether the product stores images, stores references or opens a controlled viewer, and who owns each retention obligation.

Primary reference: DICOMweb standard

11 / CAVEATS AND ACCEPTANCE

Imaging, DICOM and radiology

What can go wrong—and what your team should be able to demonstrate.

One study can be incomplete

A successful object transfer does not prove a complete examination. Track expected and observed content and the workflow's completion signal.

Metadata can contain identifying information

De-identification must address metadata, private attributes and possible burned-in pixel information. Do not assume deleting a patient-name field makes an image anonymous.

Viewer links can bypass intended controls

Avoid long-lived unrestricted URLs. Confirm session, patient context, authorization and expiry at the viewer boundary.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Interrupt a multi-instance transfer and verify reconciliation identifies missing content.

02. Open a viewer with an unauthorized user and an expired link; confirm access is denied.

03. Process an amended report and corrected patient association while preserving audit history.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic radiology product receives a report before all study objects are available. It displays report and image availability as separate states, rather than claiming the examination is complete. A later authorized viewer launch retrieves the intended study without exposing an archive-wide credential.

Ask for: Imaging identity model, service matrix, viewer-security design and study-completeness test pack.

12 / MEDICATION WORKFLOW

Pharmacy and electronic prescribing

Medication integration includes distinct workflows: prescribing, renewals, cancellations, dispensing, history, benefit checks and pharmacy claims. NCPDP SCRIPT, pharmacy transaction standards and clinical FHIR resources are not interchangeable. Select the transaction, adopted version and network route appropriate to the actual use case. Licensed standards and approved partner specifications are required for production implementations.

01 Separate the medication events

Identify who may prescribe, amend, renew or cancel and which system owns the medication order. Distinguish an order from a dispense and an administration. Record drug identifiers, strength, dose form, route, quantity and pharmacy identity without assuming a display-name match is safe.

02 Confirm the exchange program

Review the required standard/version, network onboarding, credentialing and partner testing. Scope controlled-substance requirements separately with the appropriate compliance owner. Do not substitute a generic FHIR write for an approved prescription transmission workflow.

03 Reconcile responses and changes

Track business acknowledgments, rejection, renewal requests, cancellation outcomes and dispensing information. Preserve correlation across amended transactions. Present unresolved changes to authorized users and prevent retries from creating unintended duplicate prescriptions.

Primary reference: NCPDP ePrescribing resources

12 / CAVEATS AND ACCEPTANCE

Pharmacy and electronic prescribing

What can go wrong—and what your team should be able to demonstrate.

Cancellation is not guaranteed reversal

A cancellation request can arrive after dispensing or fail to match. Surface the actual response and the need for follow-up.

Medication history is not an active medication list

Historical claims or fills do not establish current use, adherence or clinical intent. Preserve source and time context.

Licensing and access are real dependencies

Do not copy proprietary transaction guides into project documentation or assume access to a pharmacy network is included in software development.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Transmit a synthetic cancellation with an unmatched identifier and verify visible follow-up.

02. Simulate a timeout after submission; verify reconciliation before any resend.

03. Present historical and discontinued medication information without incorrectly marking it as currently prescribed.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

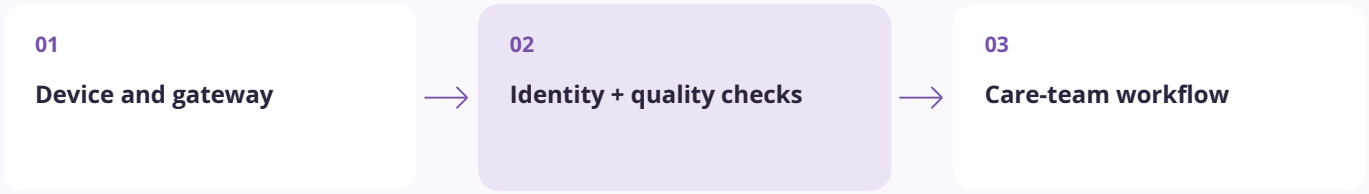
A synthetic refill workflow receives a renewal request. It routes to an authorized clinician, records the decision and correlates the response to the original request. A transport success never bypasses clinical authorization, and a rejected response remains visible to the responsible team.

Ask for: Medication-event model, standards/network prerequisites, correlation rules and safety-oriented workflow tests.

13 / MEASUREMENTS IN CONTEXT

Devices and remote patient monitoring

A device integration connects measurements to the correct person, device, time and care workflow. A value without its unit, acquisition context and quality status can be misleading. Consumer wearables, home monitoring gateways and bedside devices have different latency, reliability and clinical-use assumptions. Separate data collection from any clinical alerting or treatment workflow; each needs an accountable owner.



01 Specify measurement context

Record device identifiers, patient assignment history, measurement type, units, firmware and clock behavior. Distinguish acquisition time from upload time. Confirm whether readings are raw, averaged, manually entered or algorithmically derived.

02 Handle disconnected operation

Use durable gateway queues, bounded retries and deduplication keys appropriate to the device. Preserve late readings without presenting them as current. Apply backpressure and define the maximum acceptable data age for each consuming workflow.

03 Close the care loop

Route accepted measurements to the agreed record or queue. Define alert ownership, coverage hours, escalation and acknowledgment separately from ingestion. Obtain clinical approval for thresholds and presentation; the integration team should not invent medical rules.

Primary reference: FHIR Observation

13 / CAVEATS AND ACCEPTANCE

Devices and remote patient monitoring

What can go wrong—and what your team should be able to demonstrate.

Assignment can change

A shared device may be reassigned or used by the wrong household member. Never infer a patient solely from the most recent device owner without an approved assignment rule.

Units and time are safety concerns

Distinguish unit conversion from clinical interpretation. Device clock drift and daylight-saving changes can corrupt trends even when individual values look plausible.

No reading is not a normal reading

Distinguish offline, poor signal, unsupported measurement and no patient activity. Missing observations cannot be silently displayed as reassuring clinical status.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Replay a buffered day of readings and verify acquisition times and duplicates.

02. Reassign a device between two synthetic patients and verify historical attribution.

03. Simulate a failed alert destination and confirm escalation is visible, not marked complete.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic blood-pressure device uploads after an overnight outage. Yesterday's readings enter the longitudinal record with their original times. The dashboard indicates delayed arrival and applies the clinically approved late-data policy instead of treating every observation as a live emergency.

Ask for: Device assignment model, measurement dictionary, freshness contract and clinically approved escalation test pack.

14 / PATIENT ACCESS WORKFLOWS

Scheduling, telehealth and engagement

Patient-facing integrations need to coordinate capacity, bookings, visit preparation, communication and follow-up. An available slot is not a confirmed appointment. A video session is not proof that a visit was clinically completed. Design the state changes across the scheduling system, patient application and care team before adding reminders or self-service rescheduling.

01 Choose the booking authority

Identify which system owns provider calendars, visit types, location and resource constraints. Record eligibility rules and whether slots can be held. Define a booking state model, including requested, confirmed, cancelled and failed.

02 Coordinate changes safely

Correlate source and destination appointment IDs. Handle concurrent booking, cancellations and rescheduling as business operations with compensating actions. Do not assume a multi-system transaction can be rolled back atomically.

03 Minimize communication exposure

Use approved templates and channels, record communication preferences and keep sensitive clinical details out of unsecured reminders. Make expiring telehealth links patient-specific and define proxy, interpreter and caregiver access.

Primary reference: FHIR Appointment

14 / CAVEATS AND ACCEPTANCE

Scheduling, telehealth and engagement

What can go wrong—and what your team should be able to demonstrate.

Availability expires

A slot returned by search may be taken before confirmation. Present a recoverable choice to the user rather than a false confirmation.

Status names differ

Booked, arrived, checked in, in progress and completed may have different local meanings. Map the workflow, not only labels.

A proxy is not the patient

Caregiver contact details and portal permissions need explicit handling. An appointment reminder should not expose another person's health information.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Race two requests for the final slot and verify only the authoritative booking is confirmed.

02. Cancel after a reminder was queued and verify suppression or correction.

03. Change patient time zone and verify appointment time remains unambiguous.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

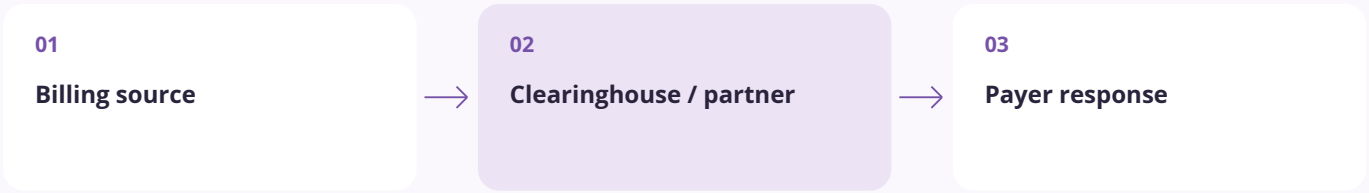
A synthetic patient requests a telehealth visit. The app confirms only after the scheduling authority returns a booking ID. If video-room creation subsequently fails, the booking remains traceable and staff receive a remediation task; the patient is not told the entire visit disappeared.

Ask for: Appointment state machine, ownership matrix, communication policy and concurrency/compensation tests.

15 / REVENUE CYCLE / SUBMISSION

Eligibility, claims and clearinghouses

Revenue-cycle integration connects registration and billing data to payer-facing transactions. Common US transaction families include 270/271 eligibility, 837 claims and 276/277 claim status. Implementation requires licensed standards where applicable, trading-partner companion guides, enrollment and a clear interpretation of responses. The clearinghouse is an intermediary; its acceptance is not a payer adjudication decision.



01 Map the trading relationship

Inventory payer IDs, submitter and receiver identifiers, service lines and enrollment status. Confirm test/production endpoints, certificates, supported transactions and companion-guide versions. Separate professional, institutional and dental claim requirements.

02 Validate before submission

Check required identifiers, code sets, dates, coverage and balancing rules. Assign traceable internal submission IDs and preserve the exact submitted version. Avoid exposing entire claims in application logs or support tickets.

03 Model response layers

Track file receipt, syntax acknowledgment, claim acceptance, adjudication and payment independently. Correlate batch and claim-level responses. Provide a work queue for unresolved submissions and agree when a timeout warrants investigation rather than automatic resubmission.

Primary reference: CMS transaction overview

15 / CAVEATS AND ACCEPTANCE

Eligibility, claims and clearinghouses

What can go wrong—and what your team should be able to demonstrate.

Eligibility is not a payment guarantee

A returned benefit response does not guarantee that a future claim will be paid. Preserve response date, requested service context and payer limitations.

Rejection differs from denial

Front-end rejection and adjudication denial have different remediation paths. Avoid a single failed status that hides responsibility.

Partner rules change

A syntactically valid transaction can violate a payer-specific rule. Maintain controlled companion-guide updates and regression fixtures.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Accept a batch while rejecting one claim; verify independent claim states.

02. Receive a delayed acknowledgment after a timeout; verify no duplicate submission.

03. Route an unknown payer identifier to review rather than a guessed destination.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

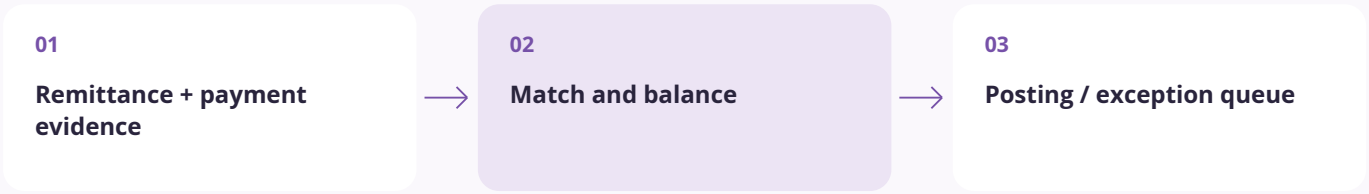
A synthetic claim receives a successful transport receipt but a claim-level rejection for an invalid provider identifier. The billing queue shows the rejection reason and submitted version. A corrected submission is linked to the original attempt without being counted as a paid claim.

Ask for: Trading-partner matrix, transaction mapping, layered status model and reconciliation dashboard.

16 / REVENUE CYCLE / RECONCILIATION

Remittance, payments and denial workflows

Remittance integration translates payer adjudication information into an auditable financial workflow. An electronic remittance advice and a funds movement are related but separate records. Claim-level and service-line adjustments need precise mapping, including reversals and reprocessing. Design posting controls with the finance team; an integration must never invent a balancing amount merely to make totals match.



01 Define matching and accounting rules

Agree claim, service-line and payment identifiers, adjustment code handling and posting boundaries. Separate patient responsibility, contractual adjustments and unresolved amounts. Obtain approved examples of reversals, partial payments and split remittances.

02 Build controlled posting

Parse the supported remittance format and preserve the source artifact. Reconcile expected totals and create proposed postings. Require authorized review for ambiguous matches or high-risk corrections. Make repeat processing idempotent at the appropriate payment and line level.

03 Route actionable exceptions

Group unmatched payments, denials and underpayment questions by accountable team. Retain payer reasons and evidence. Track correction and appeal workflows without conflating operational follow-up with a guarantee of recovery.

Primary reference: CMS electronic billing

16 / CAVEATS AND ACCEPTANCE

Remittance, payments and denial workflows

What can go wrong—and what your team should be able to demonstrate.

Funds and advice arrive separately

Do not mark a bank deposit reconciled merely because an advice file arrived. Support timing differences and missing counterparts.

Reversals are not ordinary duplicates

A reversal can intentionally negate earlier adjudication. Correlate it to the original and preserve the accounting trail.

Code meaning needs context

A reason code alone may not determine the correct financial action. Use current licensed references and finance-approved interpretation.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Receive the same remittance twice and verify no duplicate ledger posting.

02. Apply a reversal followed by corrected adjudication and verify net balances.

03. Leave an ambiguous claim match unresolved with visible ownership and evidence.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

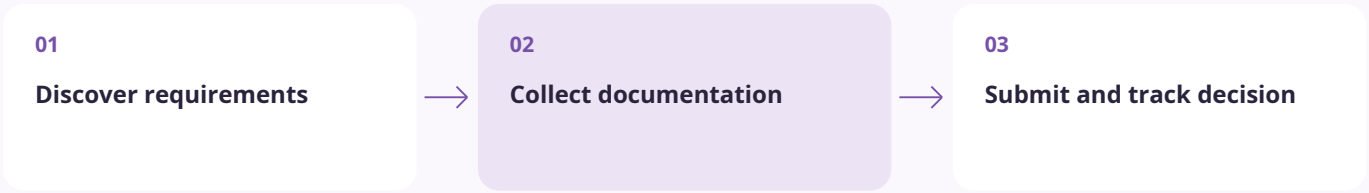
A synthetic remittance spans several claims but its bank deposit has not arrived. The system records adjudication and proposed postings separately from cash reconciliation. When the deposit appears, the matching process links the evidence; an unexplained difference remains an exception.

Ask for: Posting specification, financial reconciliation controls, reversal scenarios and an exception ownership model.

17 / PRIOR AUTHORIZATION WORKFLOW

Da Vinci CRD, DTR and PAS

Prior authorization automation spans discovery of coverage requirements, gathering supporting documentation and exchanging an authorization request and response. CRD, DTR and PAS address different parts of that journey. Treat them as connected workflow capabilities, not interchangeable endpoints. Payer rules, EHR launch support, clinical review and the authorization system of record remain explicit dependencies.



01 Specify the end-to-end scenario

Choose a concrete service, ordering context and payer product. Pin compatible implementation-guide versions and test artifacts. Identify the system that owns authorization decisions, requests for information and changes to an existing request.

02 Connect rules to evidence

Represent documentation requirements in a versioned form. Preserve prefilled data provenance and let authorized users review missing or uncertain answers. Separate executable rule evaluation from untrusted content and record which version supported each submission.

03 Persist the authorization lifecycle

Correlate requests, attachments and responses. Represent pending, approved, denied and additional-information states explicitly. Handle asynchronous responses, updates and resubmission with an audit trail. Reconcile API state with the payer’s operational workflow.

Primary reference: HL7 Da Vinci PAS

17 / CAVEATS AND ACCEPTANCE

Da Vinci CRD, DTR and PAS

What can go wrong—and what your team should be able to demonstrate.

Reference code is not readiness evidence

A working demo does not establish durability, tenant isolation, security or production conformance. Test those concerns independently.

Documentation is not always computable

Some questions need human judgment or unavailable evidence. Provide a review path rather than fabricating an answer.

A response can change later

Model amendments, expiration and changed service details. Do not freeze the application at its first successful response.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Request more information after an initial submission and verify a correlated response.

02. Restart the service with a pending authorization and verify durable recovery.

03. Attempt to access another tenant's request and verify rejection and an audit record.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic imaging authorization starts with a coverage check. The documentation workflow prefills available chart data and leaves a missing clinical rationale for review. The submitted request remains pending until the decision system responds; a successful HTTP status is not shown as approval.

Ask for: Scenario trace across CRD/DTR/PAS, version matrix, persistence design and asynchronous lifecycle tests.

18 / US PAYER INTEROPERABILITY

CMS-0057-F: a coordinated API program

CMS-0057-F covers specified impacted payers, with operational provisions generally starting in 2026 and API requirements generally in 2027; exact dates depend on payer type. Its API scope includes Patient Access enhancements, Provider Access, Payer-to-Payer and Prior Authorization. The 2024 rule excludes drug prior authorizations. Obtain payer-specific legal applicability advice and check subsequent final rules before fixing a compliance plan.

01 **Applicability + source inventory** → 02 **API and permission workstreams** → 03 **Evidence + operations**

01 Create an obligation-to-evidence register

For each applicable obligation, name an owner, authoritative source, implementation component and acceptance evidence. Separate required standards from recommended implementation guides. Do not treat CRD/DTR/PAS alone as the entire program.

02 Build source and permission foundations

Inventory claims, encounters, clinical information and authorization records. Define how member identity, provider attribution, permissions and provenance are established. Test both authorized access and exclusion cases before optimizing API throughput.

03 Plan operational readiness alongside APIs

Connect decision workflows, reporting, support and incident response to accountable teams. Maintain an evidence pack with test results, data lineage, access controls, known gaps and sign-offs. Rehearse a degraded source system rather than testing only a complete sandbox dataset.

Primary reference: CMS-0057-F final-rule fact sheet

18 / CAVEATS AND ACCEPTANCE

CMS-0057-F: a coordinated API program

What can go wrong—and what your team should be able to demonstrate.

Different APIs have different access models

Do not reuse one undifferentiated permission check across patient, provider and payer exchange. Map the applicable conditions explicitly.

A deadline is not a delivery estimate

Work backward from external dependencies, source-data readiness and partner onboarding. Avoid guaranteeing a production date before these are understood.

Proposals are not final requirements

Maintain a dated regulatory register. Clearly label proposed changes and have counsel determine which final obligations apply.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Withdraw or change an applicable permission and verify downstream access behavior.

02. Compare source records against published API content for completeness and exclusions.

03. Trace one requirement from rule reference through implementation to signed acceptance evidence.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

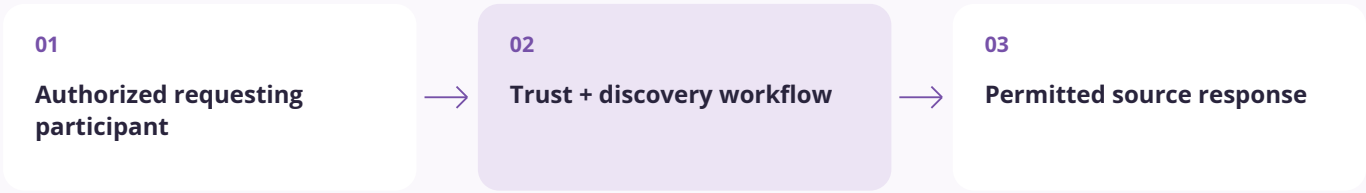
A synthetic payer can return FHIR resources but cannot reliably associate providers with members. The program records attribution as a blocking dependency for the relevant workflow, rather than declaring the API complete because sample requests pass. A separate owner resolves and tests the attribution process.

Ask for: Applicability matrix, API scope, dependency plan and an obligation-linked evidence register; not a compliance certification.

19 / CROSS-ORGANIZATION EXCHANGE

HIEs and trusted exchange networks

Health information exchanges and national exchange frameworks combine technology with participation agreements, identity, permitted purposes and operational trust. Network connectivity is not universal permission to retrieve a person's records. Identify the actual network, participant role and supported exchange workflow. TEFCA is a US trust-framework context; it is not a substitute for a local interface contract or legal review.



01 Resolve participation prerequisites

Identify the contracted route, onboarding sponsor, permitted purposes, identity requirements and production approvals. Record the actual technical endpoints and supported query or document workflow, rather than assuming one common interface across networks.

02 Constrain discovery and retrieval

Define patient matching inputs, uncertainty thresholds and human review. Verify requester identity and purpose before retrieval. Preserve source organization, document identifiers and timestamps so aggregated information can be traced back.

03 Handle cross-network differences

Normalize metadata conservatively and retain originals where permitted. Reconcile duplicate documents and conflicting information without overwriting provenance. Define who handles misdirected records, incorrect matches and requests that cannot be fulfilled.

Primary reference: ASTP/ONC TEFCA

19 / CAVEATS AND ACCEPTANCE

HIEs and trusted exchange networks

What can go wrong—and what your team should be able to demonstrate.

Finding a record does not authorize every use

Technical reachability and legal permission are separate. Secondary use needs its own assessment.

No response is ambiguous

A negative result can reflect matching, availability, permissions or missing source participation. Do not portray it as proof of no medical history.

Duplicates can contain meaningful revisions

Distinguish repeat transport from amended documents and new encounters. Blind content deduplication can erase clinically relevant updates.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Submit insufficient demographics and verify no unsafe automatic match.

02. Retrieve two versions of a synthetic document and preserve provenance.

03. Deny a disallowed purpose of use even when the endpoint is technically reachable.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

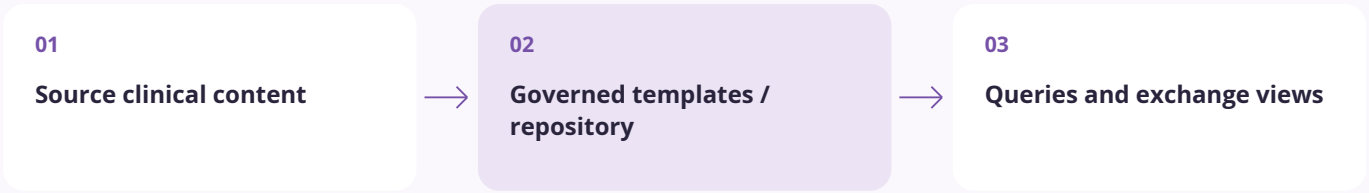
A synthetic referral application retrieves a document through an approved exchange route. It presents the originating organization and document date. A second source supplies a conflicting medication history; the application flags the difference instead of choosing a clinical truth on its own.

Ask for: Participation dependency register, purpose-of-use controls, matching policy and provenance-preserving retrieval tests.

20 / CLINICAL DATA ARCHITECTURE

openEHR and longitudinal clinical models

openEHR separates a stable reference model from clinical models expressed through archetypes and templates. It can support a longitudinal clinical repository and query layer. FHIR and openEHR address related but different concerns; an API mapping between them needs explicit semantic decisions. Neither adopting a repository nor exporting FHIR automatically preserves every detail of the source clinical model.



01 Govern the clinical model

Choose approved archetypes and templates with clinical stakeholders. Version them, document terminology bindings and define ownership of changes. Model units, context, timing and optionality before importing production information.

02 Implement repository boundaries

Agree EHR identifiers, composition lifecycle, version handling and supported REST/query operations. Enforce authentication, authorization and tenant isolation outside any assumption that the data model provides them automatically.

03 Map exchange views deliberately

Create reviewed mappings from source data into compositions and from the repository into required FHIR profiles or other outputs. Document unmapped information and preserve provenance. Regression-test queries when templates evolve.

Primary reference: openEHR REST API overview

20 / CAVEATS AND ACCEPTANCE

openEHR and longitudinal clinical models

What can go wrong—and what your team should be able to demonstrate.

Model changes affect consumers

A template update can change query paths or required fields. Plan compatibility and migration, not just deployment.

A mapping may be lossy

Not every source concept has a direct target representation. Expose loss or ambiguity rather than flattening it silently.

Repository and workflow are separate

A well-modeled composition does not by itself create an order, notification or clinical task in another system.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Load two supported template versions and verify expected query behavior.

02. Round-trip a synthetic record through an exchange mapping and document losses.

03. Attempt a cross-patient or cross-tenant query without permission and verify denial.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

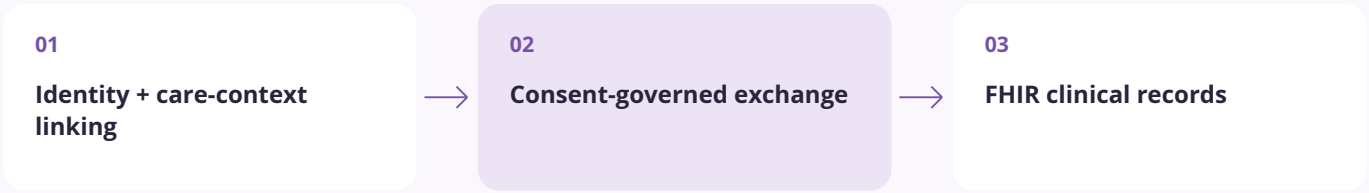
A synthetic assessment is stored using a governed template. A downstream FHIR view exposes the agreed observations while retaining links to source composition and version. When the template changes, the old assessment remains interpretable and consumer tests run before release.

Ask for: Clinical model governance, template/version register, mapping decisions and repository/query regression tests.

21 / INDIA / CONSENTED HEALTH EXCHANGE

ABDM and India-specific exchange

An ABDM project connects local hospital or product workflows to India's digital health ecosystem. Scope ABHA-related identity journeys, record linking and consented health-information exchange separately. Map the applicable milestone and HIP/HIU responsibilities to the current sandbox contract and FHIR implementation guide. US payer and EHR assumptions should not be imported into this workflow without review.



01 Scope actual user journeys

Document registration, linking, consent, record generation and retrieval scenarios with the hospital or HIMS team. Identify where staff or patients act, where callbacks arrive and which organization owns each credential and endpoint.

02 Prepare clinical data, not just APIs

For M2/M3-related exchange work, assess whether required patient records can be generated and consumed in the applicable FHIR document profiles. Inventory local forms, codes and custom workflows. Budget explicit mapping and remediation for data that is not already structured.

03 Treat onboarding as evidence gates

Agree internal reviews, functional/WASA-related checks where applicable, security testing and external demonstrations against the current official process. Track feedback, corrections and production access as dependencies. Confirm publication or observation windows with the onboarding authority; do not promise a universal fixed duration.

Primary reference: ABDM FHIR implementation guide

21 / CAVEATS AND ACCEPTANCE

ABDM and India-specific exchange

What can go wrong—and what your team should be able to demonstrate.

Milestone labels do not describe all work

A nominal milestone can involve substantial local data and workflow customization. Show buyers the actual records, screens and callbacks in scope.

Consent is part of the runtime

Expiry, revocation, requested scope and failed callbacks need tests. A previous successful consent cannot authorize every future exchange.

Sandbox success is not production approval

Maintain environment separation, credential ownership and deployment evidence. External review timing is not fully controlled by the delivery team.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Expire a synthetic consent before retrieval and verify the request is not fulfilled improperly.

02. Retry an exchange callback and verify one correlated workflow state.

03. Validate a locally generated record against the pinned ABDM profile and review clinical meaning.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

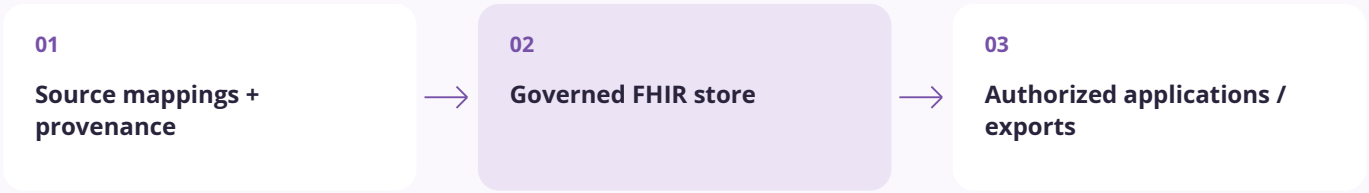
A synthetic hospital has working identity screens but stores discharge summaries as inconsistent text. The delivery plan creates a structured extraction and review workflow, a FHIR mapping and profile tests before attempting the exchange demonstration. Custom data work is visible in scope and acceptance.

Ask for: Milestone-to-workflow scope, FHIR record mappings, consent/callback tests and a dated onboarding evidence plan.

22 / HEALTHCARE DATA PLATFORMS

Managed and self-hosted FHIR stores

A FHIR store can become a governed exchange repository, an application backend or a normalized access layer over source systems. Managed options include Azure Health Data Services, Google Cloud Healthcare API and AWS HealthLake; self-hosting is another operating model. Compare the specific supported features, regions and contracts. A managed endpoint does not automatically normalize source data or make your complete application compliant.



01 Define the store’s responsibility

Decide whether it is authoritative, a replica or a derived projection. Record supported resources, profiles, search, history, transactions and bulk operations. Test required features on shortlisted services rather than treating the FHIR label as feature parity.

02 Design security and lifecycle

Choose tenant isolation, network paths, key management and permission enforcement. Define retention, backup/restore, deletion, source correction and access audit policies. Verify contractual and regional requirements with security and legal owners.

03 Prove workload economics

Benchmark realistic resource sizes, queries, imports and concurrency. Include indexing, egress, logs, storage growth and operations in cost estimates. Exercise a restore and a controlled export so exit strategy is demonstrated, not merely stated.

Primary reference: Azure Health Data Services FHIR overview

22 / CAVEATS AND ACCEPTANCE

Managed and self-hosted FHIR stores

What can go wrong—and what your team should be able to demonstrate.

Cloud responsibility is shared

Your application still owns correct authorization, mappings, secrets and operational procedures. Provider attestations are not your end-to-end certification.

Imports can be partially unsuccessful

Reconcile accepted and rejected resources, references and counts. An import job finishing does not prove clinical completeness.

Self-hosting buys control and work

Plan upgrades, capacity, database care, backups, security patches and on-call ownership. Compare lifecycle cost, not only infrastructure price.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Import a mixed-validity dataset and reconcile every accepted or rejected record.

02. Restore an isolated environment and verify references and access controls.

03. Benchmark a representative search workload and measure cost and latency under load.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

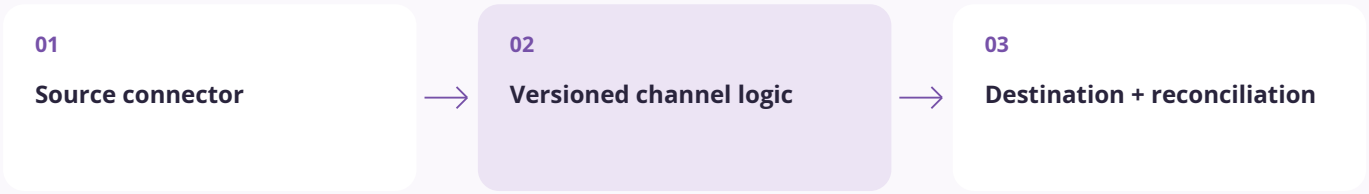
A synthetic team wants a cloud FHIR store for a patient application. Discovery reveals that one required search and the proposed tenancy model need validation. A proof-of-fit exercise tests these first; the team avoids committing to a platform based only on a sample create/read demo.

Ask for: Platform scorecard, responsibility matrix, benchmark results, ingestion reconciliation and restore/exit evidence.

23 / ROUTING AND TRANSFORMATION

Mirth and integration-engine operations

An integration engine routes, transforms and coordinates messages between systems. It can centralize transport concerns, but uncontrolled channels become another form of technical debt. Treat engine configuration, scripts, dependencies and deployment as software. Confirm the exact product version, licensing and support terms; do not assume that historical community-edition arrangements apply to every current release.



01 Separate reusable and partner logic

Keep shared validation, logging and transport utilities distinct from partner-specific mappings. Document each channel's inputs, outputs, dependencies, retry policy and stateful behavior. Avoid hidden coupling through shared global variables.

02 Promote controlled releases

Store exportable configuration and code in version control without credentials. Use environment-specific secret references, automated regression fixtures and peer review. Define deployment order and rollback behavior when multiple channels depend on one another.

03 Operate for failure

Set queue and disk thresholds, retention and secure payload-access rules. Track oldest unprocessed work, destination failures and reconciliation gaps. Provide a controlled replay procedure that prevents duplicate downstream clinical or financial actions.

Primary reference: NextGen Mirth Connect documentation

23 / CAVEATS AND ACCEPTANCE

Mirth and integration-engine operations

What can go wrong—and what your team should be able to demonstrate.

A green channel can hide bad outcomes

Transport success does not establish that the receiving workflow completed. Monitor business outcomes alongside engine status.

Retries can amplify outages

Use bounded backoff and destination protection. An unbounded queue draining at once can overwhelm a recovering partner.

Scripts accumulate risk

Review custom dependencies, runtime versions and security patches. One opaque script should not contain every mapping and business rule.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Stop a destination, build a queue and verify recovery within the agreed rate limit.

02. Deploy a mapping change and roll back without losing queued work.

03. Replay a previously delivered synthetic message and verify duplicate protection.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

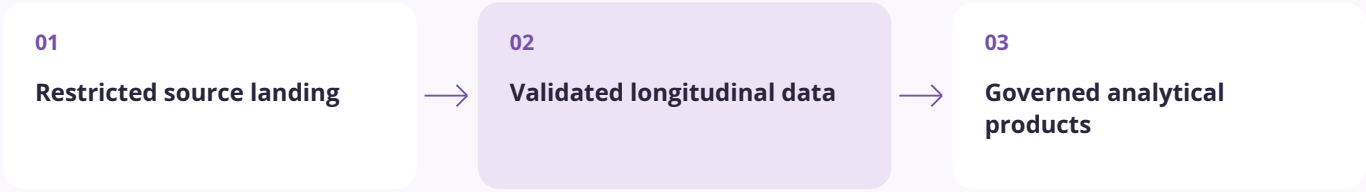
A synthetic laboratory destination goes offline. Messages remain durable, queue-age alerts identify the affected interface and operations contacts the owner. After recovery, controlled draining and reconciliation establish which results reached the destination; operators do not simply press replay on every message.

Ask for: Channel inventory, versioned deployment package, regression fixtures and queue/replay operating procedures.

24 / DATA ENGINEERING

Databricks, Snowflake and healthcare analytics

Healthcare data engineering turns source events, records and files into trustworthy analytical products. A layered raw, validated and curated design can separate ingestion from business interpretation. Databricks and Snowflake are platform choices, not substitutes for clinical semantics or governance. Define the decision each dataset supports, its allowed users and the freshness it needs before choosing a processing engine.



01 Land traceable source data

Capture source, extraction window, schema version and record identifiers. Use approved immutable retention where appropriate, with access and deletion policies. Separate processing timestamps from clinical event dates and keep an ingestion manifest for reconciliation.

02 Build tested transformations

Define identity resolution, code and unit handling, deduplication and late-arriving corrections. Version transformation logic and data contracts. Test incremental processing against a full rebuild on representative synthetic data before trusting a watermark.

03 Publish governed products

Assign dataset owners, definitions and quality thresholds. Enforce row/column access and validate exported datasets independently. Track lineage to source and expose freshness and known gaps to analysts. Measure workload cost and manage runaway jobs.

Primary reference: Databricks medallion architecture

24 / CAVEATS AND ACCEPTANCE

Databricks, Snowflake and healthcare analytics

What can go wrong—and what your team should be able to demonstrate.

A table can be current but wrong

Freshness does not prove population completeness or valid denominators. Reconcile source coverage and business meaning.

De-identification needs expert review

Removing names alone does not establish that data is safe for unrestricted use. Have qualified privacy specialists approve the method and intended use.

Analytical truth differs from operational state

A historical cohort and a live care queue need different correction and latency policies. Do not reuse a dataset without checking its contract.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Apply a late correction and verify historical and current views follow the documented policy.

02. Rebuild a partition and compare counts, keys and aggregates against incremental output.

03. Query as two roles and verify sensitive columns and populations remain restricted.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

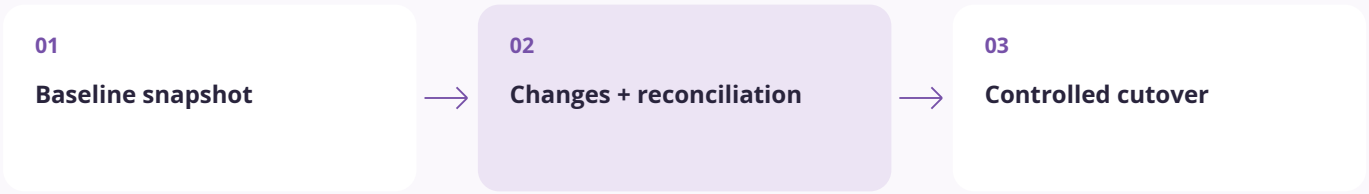
A synthetic utilization dashboard counts encounters from several sources. A replay previously doubled one facility's encounters. The revised pipeline uses source-scoped identifiers, versioned deduplication and reconciliation by facility/date; the dashboard shows coverage limitations rather than a misleading complete total.

Ask for: Source-to-product lineage, data contracts, quality tests, access policies and freshness/cost operating metrics.

25 / STATE TRANSFER AND CUTOVER

Migration, CDC and historical backfill

A migration moves historical state while current workflows continue changing it. Change data capture, API extraction, bulk export and file imports are different mechanisms with different guarantees. Agree the source of truth, cutover boundary and reconciliation method. A successful row count is necessary but insufficient: identifiers, references, meaning and operational continuity must survive the move.



01 Profile before committing dates

Inventory volumes, schema variation, missing identifiers, attachments and retention constraints. Sample difficult records, not just typical ones. Define transformation rules and quantify records requiring manual review or source remediation.

02 Bridge baseline and live changes

Establish a consistent extraction boundary and capture changes during backfill. Define overlap handling, delete semantics and idempotent writes. Track watermarks durably and reconcile gaps before advancing them. Never assume a timestamp alone establishes total event order.

03 Rehearse cutover and recovery

Run a production-like rehearsal with synthetic or approved protected data. Agree freeze windows, acceptance thresholds, rollback triggers and who authorizes the switch. Preserve a controlled path to investigate records after cutover.

Primary reference: FHIR Bulk Data Access

25 / CAVEATS AND ACCEPTANCE

Migration, CDC and historical backfill

What can go wrong—and what your team should be able to demonstrate.

Counts can match while references break

Check referential integrity, statuses, units and representative workflows in addition to totals.

Deletes and corrections need a policy

Historical snapshots can resurrect deleted or superseded information. Align migration behavior with retention and correction obligations.

Rollback may not be symmetric

Once users create new work in the target, restoring the source is not just reversing a copy. Plan reconciliation of post-cutover changes.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Modify a record during backfill and verify the final version is correct.

02. Resume after a failed batch without duplicates or skipped ranges.

03. Rehearse a rollback trigger and account for all target-created changes.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

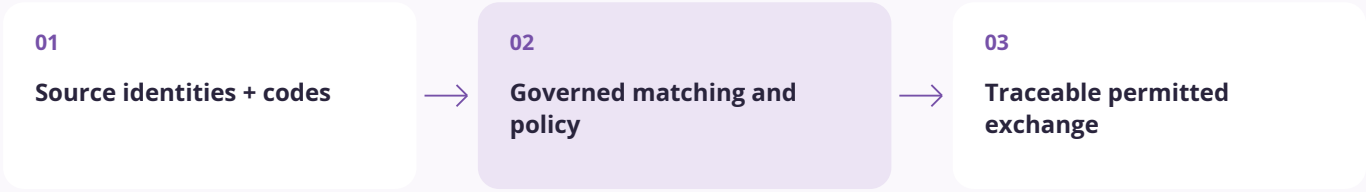
A synthetic migration loads a baseline while new results continue arriving. The team records an overlap window and reconciles source versions before cutover. A small unresolved exception set remains visible with named owners; it is not hidden behind a headline completion percentage.

Ask for: Profiling report, baseline/change boundary, reconciliation evidence and signed cutover/rollback runbook.

26 / SHARED SEMANTIC FOUNDATIONS

Identity, consent and terminology services

Shared identity, permission and terminology services support many integrations but solve different problems. Identity establishes which entity a record refers to. Consent and authorization determine whether an operation is permitted. Terminology preserves the meaning of coded information. A correct patient match does not confer permission, and a valid code does not prove that it is clinically appropriate in context.



01 Model identity with provenance

Scope identifiers by assigning authority and entity type. Define merge, unmerge, alias and uncertain-match workflows. Set automated-match thresholds with accountable stakeholders and route ambiguity to review rather than guessing.

02 Make policy executable and explainable

Record policy sources, purposes, actors, scope and effective periods. Evaluate permissions at the operation boundary and cache only with explicit invalidation rules. Log the decision and policy version without unnecessarily duplicating sensitive data.

03 Govern terminology mappings

Record source and target code systems, versions, mapping direction and review status. Preserve original codes and units. Treat approximate mappings as approximate, and involve qualified domain reviewers for clinically consequential transformations.

Primary reference: FHIR security and privacy

26 / CAVEATS AND ACCEPTANCE

Identity, consent and terminology services

What can go wrong—and what your team should be able to demonstrate.

Merge mistakes propagate

A wrong identity association can contaminate every downstream system. Plan correction propagation and access restriction while investigating.

Consent is not the only lawful basis

Applicable legal authority and contractual rules vary. Have privacy/legal owners define the policy rather than treating one checkbox as universal permission.

Mappings are not always one-to-one

A broad source concept cannot always be safely converted into a more specific target concept. Retain uncertainty and avoid invented precision.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Merge and unmerge synthetic identities and trace downstream correction.

02. Change a permission while a cached session is active and verify enforcement.

03. Process an unmapped or deprecated code and surface an actionable exception.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

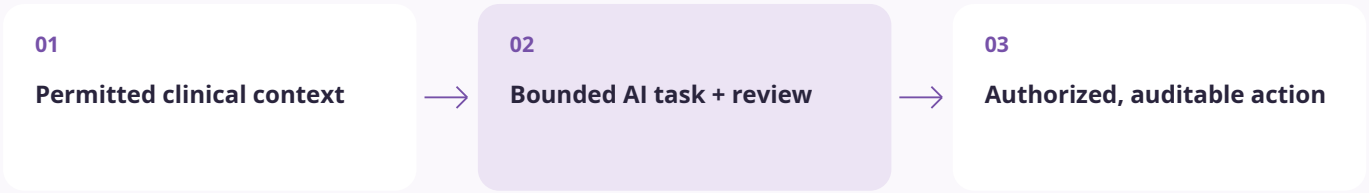
Two synthetic facilities use the same local patient number for different people. The shared service includes assigning authority in its key and prevents an accidental merge. A later reviewed association retains both source identifiers and an audit trail rather than overwriting one.

Ask for: Identity lifecycle policy, permission decision matrix, terminology register and correction propagation tests.

27 / AI-ENABLED WORKFLOWS

AI agents and clinical-system write-back

AI integrations connect models or agents to clinical information and operational tools. Read-only summarization, draft creation and autonomous action have different risk profiles. Define the intended use and required human review before selecting tools. Treat retrieved notes and documents as untrusted data, not instructions that can expand an agent's permissions or redirect patient information.



01 Constrain the task and tools

Name the user, decision supported and prohibited actions. Grant narrow tool permissions and enforce them server-side. Separate read, draft and commit operations. Prevent model-generated URLs or instructions from bypassing approved destinations and access checks.

02 Preserve evidence and uncertainty

Attach source references, timestamps and relevant limitations to outputs. Validate structured fields before use and avoid interpreting absent data as negative findings. Route clinically consequential content through qualified review under an approved workflow.

03 Control release and monitoring

Version prompts, models, retrieval and tool schemas. Evaluate against representative synthetic or properly governed cases, including adversarial inputs. Monitor overrides, unsupported claims and action failures. Maintain a kill switch and a non-AI fallback.

Primary reference: NIST AI Risk Management Framework

27 / CAVEATS AND ACCEPTANCE

AI agents and clinical-system write-back

What can go wrong—and what your team should be able to demonstrate.

Fluent text can be unsupported

Require evidence checks; confidence-sounding language is not a reliability measure.

Prompt injection crosses data boundaries

A note may contain instructions to send data elsewhere. Tool authorization must remain independent of model compliance.

Write-back changes the risk

A draft and a committed order or billing action are not equivalent. Use explicit review, confirmation and concurrency checks appropriate to the workflow.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Embed malicious instructions in a synthetic note and verify no unauthorized tool action.

02. Change source data after draft creation and verify review detects stale context.

03. Disable the model service and verify users retain a safe manual workflow.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

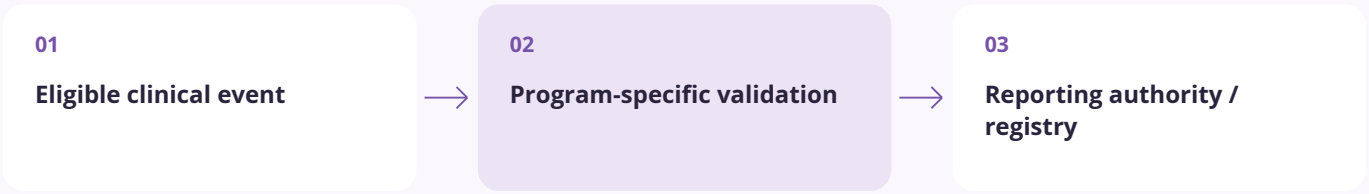
A synthetic documentation assistant proposes a draft note with source links. The clinician reviews and edits it before an authorized commit. A concurrent chart update triggers a refresh warning. The audit trail distinguishes generated text, human edits and the final committed record.

Ask for: Intended-use statement, tool permission model, evaluation set, human-review design and rollback/kill-switch evidence.

28 / REPORTING BEYOND THE ENCOUNTER

Public health, registries and specialist feeds

Public-health and registry integrations connect clinical events to specific reporting programs. Electronic case reporting, immunization exchange, quality registries and specialist reporting differ in triggers, data requirements and acknowledgment workflows. Genomics and pathology feeds also require specialist interpretation of versions and context. Start with the recipient's implementation guide and responsible program team, not a generic export.



01 Define the reporting obligation and route

Identify jurisdiction, program, permitted use, enrollment and supported format. Record who determines reportability and which clinical events trigger evaluation. Keep program-specific logic versioned and reviewed by the responsible clinical or public-health team.

02 Map complete evidence

Include required demographics, encounter context, codes and source provenance. For specialist results, preserve report versions and clinically relevant interpretation context. Do not reduce a complex result to a single code merely to fit a convenient schema.

03 Reconcile acknowledgments and corrections

Track transport receipt separately from program acceptance and follow-up. Correlate amendments, duplicate events and rejected submissions. Provide an exception workflow with the recipient and retain evidence under the approved policy.

Primary reference: CDC electronic case reporting

28 / CAVEATS AND ACCEPTANCE

Public health, registries and specialist feeds

What can go wrong—and what your team should be able to demonstrate.

Jurisdiction changes the contract

A successful connection to one program does not establish compatibility with another. Maintain separate capability and onboarding records.

Clinical interpretation belongs to specialists

Genomic variants, pathology amendments and reportability rules require qualified domain review. Integration transformations should not invent diagnoses.

Corrections must reach the recipient

A later amended report can change the meaning of a submission. Track correction obligations instead of assuming the first transmission is final.

ACCEPTANCE SCENARIOS / SYNTHETIC TEST DATA

01. Trigger a synthetic reportable event and verify the approved routing and payload.

02. Receive a program rejection and confirm accountable follow-up.

03. Send an amended specialist result and verify correlation with the original submission.

WORKED EXAMPLE / NOT A CLINICAL INSTRUCTION

A synthetic report is technically delivered but rejected for missing jurisdiction-specific information. The reporting queue retains the source event and rejection, assigns remediation and sends a correlated correction. The dashboard counts accepted reports separately from transmitted messages.

Ask for: Program-specific trigger and mapping contract, enrollment dependencies and acceptance/correction evidence.

REUSABLE DELIVERY TOOL

A mapping contract you can review.

A field list is not enough. Make meaning, failure behavior and ownership inspectable. This synthetic example is an illustration, not a deployable mapping.

Business outcome	A corrected laboratory result updates the intended result in the receiving workflow without losing the original audit trail.
Source identity	Facility + assigning authority + patient identifier; order and result identifiers scoped to the sender. Document merge behavior.
Clinical meaning	Source test code and version; approved target mapping; original unit and approved conversion; specimen and collection time; result status.
Missing or invalid input	Quarantine an unknown patient authority or unmapped critical code. Do not silently invent a target or convert missing data to normal.
Event semantics	Distinguish preliminary, final, corrected and cancelled results according to the agreed source contract. Define out-of-order behavior.
Delivery contract	Record acknowledgment meaning, timeout behavior, correlation key and deduplication window. Reconcile receiving application state.
Change control	Name clinical and technical reviewers. Version the mapping and fixtures; record compatibility and rollback effects.
Acceptance evidence	Show original and corrected synthetic messages, mapping output, downstream state, audit history and a controlled replay result.

Use the same structure for appointments, claims, device readings and FHIR transformations. Replace the example with your actual counterparty contract.

CROSS-CUTTING CONTROLS

Security is a workflow property.

The following is an engineering review aid—not a complete HIPAA checklist or legal determination. Assign a control owner and retain test evidence for your actual deployment.

Identity and least privilege	Separate human, application and operator identities. Test denied actions, expired credentials and tenant boundaries. Rotate secrets without embedding them in channel exports.
Network and encryption	Protect transport and stored information using approved mechanisms. Restrict endpoints, validate certificates and prevent credentials following untrusted redirects.
Sensitive-data minimization	Collect only needed data. Redact logs, traces, analytics and support attachments. Restrict payload inspection and record privileged access.
Permission and purpose	Evaluate access at the operation boundary. Model revocation, scope and effective time. Have privacy owners approve intended uses and disclosures.
Supplier responsibility	Verify contracts, applicable BAAs, subprocessors, regions and incident obligations. Document your responsibilities even when a cloud provider operates infrastructure.
Retention and recovery	Define deletion and retention across source, queue, store, backup and export. Restore in an isolated environment and verify permissions before reopening access.
Incident response	Identify decision makers, containment steps and communication obligations. Exercise a suspected misrouted record scenario with legal/security teams.

Reference: HHS HIPAA Security Rule resources

EVIDENCE BEFORE GO-LIVE

Test the failure you hope never happens.

Passing schema validation is only the first layer. Agree pass/fail criteria with the people who own the clinical, financial and operational consequences.

Contract / conformance	Pin versions, profiles and companion guides. Validate positive and negative fixtures; record intentional deviations and counterparty acceptance.
Semantic correctness	Review units, codes, identifiers, references, statuses and dates. Include missing, corrected, cancelled and conflicting information.
Workflow acceptance	Show the intended outcome in the receiving system. Include human review, exception handling and the effect of a partial multi-system failure.
Resilience	Interrupt transport, restart processing and expire credentials. Test ambiguous timeouts, duplicates, out-of-order events and controlled replay.
Security / privacy	Test wrong tenant, wrong patient, insufficient permission and sensitive-data leakage. Include malicious retrieved content for AI workflows.
Performance / capacity	Use a documented workload: volume, payload size, concurrency and arrival pattern. Measure queue age, end-to-end latency and recovery under backlog.
Reconciliation / recovery	Account for sent, accepted, rejected and unresolved records. Exercise backup restoration and verify downstream state—not only service availability.

A RELEASE GATE WITH SUBSTANCE

Go live only when agreed critical scenarios pass, unresolved risks have an accountable acceptance decision, operational owners are trained, and rollback/recovery has been rehearsed. A screenshot of a successful API response is not this evidence.

AFTER THE LAUNCH

Make ownership visible at 2 a.m.

The operating model should distinguish an unavailable interface, delayed data and a wrong business outcome. Each can require a different response.

Service indicators	Track oldest outstanding work, source-to-destination latency, error classes, reconciliation gaps and data freshness by partner and workflow.
Targets	Agree service hours, escalation, recovery-time and recovery-point objectives. Set thresholds from workflow needs and measured baselines, not generic marketing numbers.
Triage	Identify affected partner, tenant and time range without exposing payloads broadly. Separate authentication, validation, mapping, destination and infrastructure faults.
Containment	Pause unsafe writes when necessary. Preserve evidence, quarantine problematic work and communicate impact to the designated clinical or business owner.
Recovery	Fix the cause, test representative work, drain at a safe rate and reconcile outcomes. Replay only with documented duplicate and ordering protections.
Handover	Provide system diagrams, access procedures, mapping/version registers, dashboards, alert routing, runbooks, known limitations and named support owners.
Continuous change	Review partner notices and expiring credentials. Run fixtures before profile, code-system or vendor upgrades. Update the dependency and risk register.

Example incident exercise: a partner accepts messages but stops applying results. Your health check stays green. Can reconciliation detect the gap, identify affected records and prove a safe recovery?

YOUR IMPLEMENTATION BRIEF / FILLABLE WORKSHEET

Turn the unknowns into a scope.

Use sanitized project context only. Save a local copy after editing; PDF viewers differ in form support. Share this brief only through an approved channel.

01 / INTENDED OUTCOME

Which action must happen in which system, and who relies on it?

02 / SYSTEMS AND ACCESS

List vendors, versions, partners and known access/onboarding constraints.

03 / HIGHEST-RISK UNKNOWNS

What is uncertain about source data, permissions, workflow or dependencies?

04 / ACCEPTANCE AND OWNERSHIP

What evidence proves success, and who signs off and operates the integration?

Do not include names of patients, medical records, credentials or confidential payloads. A systems-level description is enough for an initial scoping conversation.

FROM REFERENCE TO IMPLEMENTATION

Bring the difficult connection.

Nirmitee.io helps healthcare teams scope, build and operate connected products and integration workflows. Start with one concrete use case and the uncertainty holding it back.

What we will discuss

Your intended workflow, source and destination systems, access constraints, available data and the evidence you need for acceptance. We can identify what needs discovery before an implementation estimate is credible.

Scoped integration project	A defined workflow, counterparties, mappings, test evidence and handover. Confirm exclusions and external dependencies in the statement of work.
Platform or data foundation	Shared adapters, a FHIR store, governed data pipelines or a reusable integration layer. Validate a thin end-to-end slice before expanding.
Specialist delivery support	Architecture review or engineering support alongside your team. Agree responsibility, decision rights and operational ownership explicitly.

Inspect our open-source starting points

Nirmitee publishes Da Vinci CRD, DTR and PAS reference implementations. Review the code, documentation and limitations. Reference implementations are not production certification or a substitute for security, durability and conformance work.

CRD / coverage requirements

DTR / documentation workflows

PAS / authorization exchange

A USEFUL NEXT CONVERSATION

Bring the worksheet, a systems diagram and one sanitized example of the workflow. We will use them to discuss scope and the next discovery or delivery step. Do not email patient data.

Book an implementation conversation

PRIMARY REFERENCE INDEX / 1 OF 2

Keep the contract current.

Links are starting references, not proof that a specific vendor tenant supports every capability. Pin the actual release and obtain the counterparty's implementation guide.

01 / HL7 v2

02 / FHIR R4 REST API

03 / SMART App Launch

04 / FHIR Bulk Data Access

05 / HL7 C-CDA

06 / Epic interface catalog

07 / Oracle Millennium APIs

08 / athenahealth developer portal

09 / MEDITECH Greenfield

10 / LOINC laboratory orders

11 / DICOMweb standard

12 / NCPDP ePrescribing resources

13 / FHIR Observation

14 / FHIR Appointment

Release discipline

This edition was prepared on 13 September 2026. Some references use a moving latest URL; record the version and retrieval date in your project. The handbook's implementation and test recommendations are original engineering guidance, not quoted requirements from these sources.

Licensed and local specifications

Obtain applicable licensed X12/NCPDP material and partner companion guides. Do not build production transactions from this handbook alone. Vendor names identify integration contexts; no affiliation or endorsement is implied.

PRIMARY REFERENCE INDEX / 2 OF 2

Keep the contract current.

Links are starting references, not proof that a specific vendor tenant supports every capability. Pin the actual release and obtain the counterparty's implementation guide.

15 / CMS transaction overview

16 / CMS electronic billing

17 / HL7 Da Vinci PAS

18 / CMS-0057-F final-rule fact sheet

19 / ASTP/ONC TEFCA

20 / openEHR REST API overview

21 / ABDM FHIR implementation guide

22 / Azure Health Data Services FHIR overview

23 / NextGen Mirth Connect documentation

24 / Databricks medallion architecture

25 / FHIR Bulk Data Access

26 / FHIR security and privacy

27 / NIST AI Risk Management Framework

28 / CDC electronic case reporting

Google Cloud / FHIR concepts

AWS HealthLake / service overview

Snowflake / row access policies

HHS / security resources

Quick terms: ACK = acknowledgment · CDC = change data capture · EMPI = enterprise master patient index · HIP/HIU = health information provider/user · IG = implementation guide · LIS = laboratory information system · PACS = picture archiving and communication system · RCM = revenue cycle management.