



Diagnostic Orders Decision Guide

Use this guide to design laboratory and imaging workflows, from placing an order through reviewing results and completing follow-up. Decide which steps your application owns, how it connects to performing facilities, and who handles exceptions. Leave with a defined ordering approach, result model, and owner for each stage.

Companion to [Labs and Imaging](#), [Order Labs and Imaging](#), and [Results and Review](#).

This guide covers laboratory and imaging workflows. Use the integration-specific docs for vendor setup and supported operations. The review queues and follow-up rules below are application design recommendations to implement with Medplum resources.

Section 1: Use Case and Participants

1.1 Which diagnostic services are in scope?

- Laboratory tests sent to external labs
- Tests performed in your own facility or at the point of care
- Imaging studies performed externally or internally
- A combination

Why it matters: laboratory workflows coordinate specimens; imaging workflows usually coordinate appointments and access to reports or images. An internal laboratory also needs catalog and laboratory operations design beyond sending orders.

1.2 Which parts of the workflow will your application own?

- Ordering and receiving results
- Receiving and reviewing results ordered elsewhere
- Coordinating fulfillment of orders created elsewhere
- Performing tests and producing results

Why it matters: a results-only application needs patient matching and review routing even when it has no original order. An ordering application also needs transmission, cancellation, and outstanding-order tracking.

1.3 Who participates, and who is accountable?

Identify who can prepare and authorize orders, collect specimens or arrange appointments, resolve integration exceptions, review results, cover absent clinicians, and communicate with patients. Decide whether those responsibilities vary by organization, location, or service line.

Why it matters: the ordering clinician, performing organization, result reviewer, and follow-up owner may be different people. Unassigned work needs an explicit fallback.

1.4 Which systems are already involved?

List the performing labs and imaging facilities, existing interfaces, vendor accounts, scheduling systems, and image archives. For each, identify what it can send and receive: orders, acknowledgments, cancellations, status updates, structured results, documents, or image access.

Why it matters: the application's workflow must reflect the capabilities of each connected facility. Connectivity to one laboratory does not establish imaging support or identical behavior across every facility.

Section 2: Approach-Defining Decisions

Resolve these choices before selecting components or building interfaces. If a capability is out of scope, skip its detailed section.

Decision	What changes based on the answer	Details
External integration, internal fulfillment, or results only?	Connectivity, write authority, and whether local originating orders exist	3.1
Vendor menu or internally maintained catalog?	Catalog ownership, order codes, grouping, and maintenance	3.2
Clinician orders directly or staff prepare for authorization?	Draft/release workflow, required information, and permissions	3.3
On-site collection, external collection, home kit, or imaging?	Specimen tracking, logistics, and appointment coordination	3.4
Can the destination amend, cancel, and support recurring orders?	Replacement orders, acknowledgment tracking, and occurrence tracking	3.5
Matched results only or unsolicited results too?	Identity resolution, exception ownership, and backfill	3.6
Structured values, original documents, or diagnostic images?	Result modeling, trends, extraction, and viewer integration	3.7
Who reviews, communicates, and follows up?	Review queues, coverage, escalation, and closure rules	3.8
Which teams share access and operational responsibility?	Memberships, policies, and exception queues	3.9

Section 3: Key Decisions

3.1 How will orders and results cross system boundaries?

Questions: Which facilities must be supported at launch? Is electronic ordering required, or is a requisition sufficient? Can the interface return acknowledgments and status updates? Who handles setup and exceptions?

Situation	Recommended approach
Using the Medplum Health Gorilla integration	Follow its onboarding, order profiles, send/receive Bots, and account requirements. Confirm the needed facilities and capabilities before committing the workflow.
Connecting directly to a lab or imaging facility	Scope the HL7v2 or API interface in both directions, including identifiers, acknowledgments, corrections, cancellations, and recovery.
Performing tests internally	Create the operational FHIR resources through your application; separately scope the laboratory or imaging system that performs and reports the work.
Requisition or manual handoff	Track who handed off the order and how completion will be reconciled. Producing a document does not establish that the facility accepted the order.
Receiving results only	Start with patient matching, source identifiers, and review ownership. Allow results without a local originating order.

Do you need an on-site protocol bridge?

- **Devices or systems use HL7/MLLP, ASTM, or DICOM inside a facility network:** Use the [Medplum Agent](#) to bridge supported local protocols to Medplum. Assign an owner for the host, channel configuration, connectivity, and recovery. Bots handle workflow-specific mapping and reconciliation; transporting a message does not establish that its patient, order, or result was matched correctly.
- **Imaging equipment sends DICOM C-STORE:** Use the Agent's [DICOM channel](#). Select storage=dicomweb to populate Medplum's DICOM resources for viewing, and check the documented server/Agent version requirements. The default binary mode stores files for Bot processing and does not populate that imaging model.
- **The source already supports HTTPS APIs or DICOMweb:** Integrate directly; an Agent is not required solely because the data is diagnostic. DICOMweb sources can send studies through STOW-RS. The [Medplum CLI](#) also supports file/directory uploads for testing or controlled imports.

Decide before building: Name the authoritative system for the order, fulfillment status, and report. Define which updates may flow back into each system.

References: [Health Gorilla](#), [HL7 interfacing](#).

3.2 What can clinicians order, and who maintains the menu?

Questions: Are tests selected from a performing lab's menu or an internally curated catalog? Are panels ordered as a unit? Do visit templates offer common order sets? Who maintains codes and required questions when a vendor changes its menu?

Situation	Recommended approach
External laboratory menu	Preserve the performing lab's orderable codes and requirements. A standardized observation code alone may not identify an orderable test in that lab's compendium.

Situation	Recommended approach
Internal diagnostic catalog	Use the documented catalog model: ObservationDefinition for measurements, SpecimenDefinition for specimen requirements, PlanDefinition for orderable services or panels, and ActivityDefinition for the procedures that fulfill them.
One orderable panel	A panel can be one requested service with multiple result observations. Follow the destination's panel definition rather than creating an order for every analyte.
Several independent services ordered together	Use separate ServiceRequest resources. A shared requisition can group requests; basedOn describes a request being fulfilled, not an arbitrary shopping-cart grouping. Follow any integration-specific profile.
Health Gorilla orders	Follow its parent order and individual test ServiceRequest structure. Each test references the parent through basedOn; some labs require splitting into separate requisitions.
Reusable clinical order sets	Keep reusable definitions separate from the patient-specific ServiceRequest instances they generate. Assign an owner for changes.

Use verified terminology mappings and preserve source codes. Leave an unresolved mapping explicitly unresolved rather than guessing a LOINC, SNOMED, or diagnosis code.

Record: Catalog owner, code source, order boundaries, and requisition grouping. The [FHIR R4 ServiceRequest specification](#) distinguishes a single requested service from multiple requests.

References: [Diagnostic Catalog](#), [Health Gorilla order structure](#).

3.3 What must happen before an order can be released?

Questions: Can staff prepare orders for a clinician? Are orders placed within an encounter or independently? Which demographics, indications, Ask on Order Entry (AOE) answers, billing details, and supporting documents does the destination require?

Recommended baseline: Represent the request with ServiceRequest, including the patient, requesting clinician, intended service, and performing organization when known. Link the encounter when applicable. Use status = draft while composing and intent = order for an actual clinical order; define a controlled authorization and transmission step.

Requirement	Design consequence
Staff prepare; clinician authorizes	Give those roles distinct permissions and actions. Define the authorization record and who can trigger transmission.
Test-specific AOE	Capture the destination's required answers. The Health Gorilla pattern stores QuestionnaireResponse resources in the individual test's supportingInfo.
Insurance, patient, or account billing	Capture the billing route and required supporting information before sending. For Health Gorilla, check practitioner and practice lab account requirements.

Requirement	Design consequence
Imaging requires authorization or scheduling coordination	Assign that work explicitly and decide whether it must finish before transmission, scheduling, or performance.

Saving a FHIR order and successfully transmitting it are separate events. Follow the chosen integration's lifecycle: Health Gorilla's send Bot changes the order to active after transmission and uses on-hold for processing errors.

Record: Who prepares, authorizes, and sends the order, and what must be present at each step.

References: [Order Labs and Imaging](#), [Health Gorilla Sending Orders](#).

3.4 How will the patient or specimen reach the performing facility?

Questions: Who collects the specimen? Who schedules imaging? Are home kits involved? How will staff identify missing collections, missed appointments, rejected specimens, or incomplete kits?

Situation	Recommended approach
Specimen collected on site	Record the actual Specimen, collection details, and order linkage. Include requisition and label handling in the workflow.
Collection at an external draw station	Give the patient the destination and preparation instructions; define what evidence confirms collection and who follows up when it is missing.
Home collection	Track kit fulfillment, return, receipt, and rejection as operational work. A shipped kit does not establish specimen collection or completed testing.

Situation	Recommended approach
Imaging at another facility	Track the appointment handoff and expected completion. Use Appointment if your application manages scheduling; otherwise reconcile the external appointment status.
Multiple specimens or collection times	Preserve the individual specimen relationships and independent progress. Use the integration's order-splitting behavior when required.

If your application only needs to display the clinical order state, use `ServiceRequest` directly. Add Task when staff need assigned work such as arranging collection or chasing an appointment; set focus to the relevant order and for to the patient. Define operational progress separately from the clinical order status. See the [FHIR R4 Task relationship to requests](#).

Record: Fulfillment owner, required tracking steps, and evidence that each step finished.

3.5 What happens when an order changes or never completes?

Questions: Can orders be edited after transmission? How is cancellation acknowledged? What happens after a rejected specimen, missed appointment, or overdue result? Are recurring orders in scope?

Situation	Recommended approach
Draft changes	Edit before release and revalidate destination requirements.

Situation	Recommended approach
Change after transmission	Follow the destination's amendment rules. If replacement is required, coordinate cancellation and link the replacement with <code>ServiceRequest.replaces</code> .
Cancellation	Use revoked for the cancelled clinical order, while separately tracking whether the external destination acknowledged the cancellation.
Order entered by mistake	Use entered-in-error; preserve the history needed to understand what happened.
Overdue or failed fulfillment	Create an assigned exception with a next action. Define when staff retry, contact the patient or facility, cancel, or request a new order.
Destination supports a standing order	Capture the authorized timing and validity, and track each performance and report independently. Agree how results identify the standing order and occurrence.
Each occurrence requires a new order or requisition	Create separately authorized requests for the individual occurrences. Do not reuse a completed one-time order as a recurring work queue.

Integration constraint: Updating an active Health Gorilla order in Medplum does not modify it in the performing lab's system. Do not offer an edit action that implies an external change the integration cannot perform.

Record: Amendment/replacement rules, cancellation evidence, recurring-order model, and overdue-work owner. A timeout after sending is an

uncertain outcome: reconcile the destination's order identifier before resending. Conditional FHIR writes alone do not prevent duplicate external orders.

References: [Order lifecycle](#), [Cancelling and Amending Orders](#).

3.6 How will incoming results find the correct patient and order?

Questions: Which identifiers survive the round trip? Can reports arrive without an order? What happens on repeated messages, ambiguous matches, or missed webhooks?

Recommended baseline: Preserve source identifiers with their system namespaces. Link matched reports through `DiagnosticReport.basedOn` and measurements through `DiagnosticReport.result`. Do not assume one order always produces exactly one report or one delivery. In FHIR R4, `DiagnosticReport.basedOn` is optional and repeatable; populate it for known order relationships and validate each linkage. See [DiagnosticReport definitions](#).

Situation	Recommended approach
Known patient and order	Match using the agreed source and order identifiers; retain the identifiers needed to reconcile with the performing system.
Known patient, no local order	Retain the report without inventing an order relationship. Route it to an accountable review owner.
Unknown or ambiguous patient	Route for identity resolution. Inspect the connector's behavior before deciding whether new patient creation is acceptable.

Situation	Recommended approach
Duplicate delivery or replay	Use conditional create/update keyed on stable source identifiers. Distinguish a repeated event from a genuine report revision, and prevent older deliveries from overwriting newer results.
Missed delivery	Provide a reconciliation and backfill process, then review unresolved exceptions.

The Health Gorilla receiver flags unsolicited reports with DetectedIssue. If it cannot match the patient, it creates a new patient from the supplied demographics and flags an unknown-patient issue. Plan a queue to resolve these cases. Other interfaces need their own explicit matching policy.

Record: Matching identifiers, duplicate/revision rules, and who resolves unmatched reports.

References: [Receiving Results](#), [Working with FHIR](#).

3.7 What result detail must the application display and use?

Questions: Do users need trends and rules on discrete values, the original report, diagnostic images, or all three? Must results from several performing organizations be compared?

Need	Recommended approach
Discrete lab values and trends	Store Observation values with their units, source reference ranges, interpretation, and timing. Validate comparability before combining results across sources.

Need	Recommended approach
Report summary and lifecycle	Use DiagnosticReport with links to the observations. Preserve partial, preliminary, final, amended, and corrected states as supplied.
Original PDF or narrative report is sufficient	Preserve it through presentedForm or the integration's documented DocumentReference relationship. You do not need to manufacture discrete observations.
PDF-only source, but trends or rules need discrete results	Add an extraction and verification workflow with source provenance. Treat extracted values as candidates until validated; a PDF attachment alone cannot support reliable numeric trends.
Imaging interpretation only	Use the report's narrative and/or attached radiology report.
Diagnostic images and viewer access	Choose Medplum's DICOM storage and hosted OHIF viewer, or connect an existing archive/viewer. Resolve the imaging choices below.

For uploaded documents, use Medplum's Binary/Attachment helpers, reference Binary/{id}, and set the appropriate securityContext. Do not put base64 content in Attachment.data.

Where will images live, and how will clinicians view them?

Questions: Does a PACS already own the images? Must your application store and display studies? Should the chart and viewer switch patient or study together?

- **Medplum DICOM storage and hosted OHIF viewer**

- **Choose when:** Your application needs image storage and viewing without deploying a separate image server and viewer. Medplum’s hosted cloud includes a configured **OHIF viewer**.
- **Advantages:** DICOMweb provides storage (STOW-RS), study/series queries (QIDO-RS), and metadata/frame retrieval (WADO-RS), using Medplum authentication and access policies.
- **Tradeoffs:** DICOM support is beta and implements a subset of DICOMweb. Validate the needed modalities, transfer syntaxes, and viewer operations. Current QIDO study queries ignore filtering and pagination parameters; use FHIR search on `DicomStudy` when those are required. Self-hosted OHIF deployment is an enterprise feature.
- **Existing PACS or external viewer**
 - **Choose when:** An imaging facility already owns image storage and interpretation, and your application mainly coordinates orders and reviews reports.
 - **Advantages:** Retains the facility’s established archive and viewing workflow; avoids requiring a second image copy just to display the report.
 - **Tradeoffs:** Define viewer launch, authentication, study identifiers, and image availability with the facility.
`DiagnosticReport.imagingStudy` can reference an `R4 ImagingStudy`, but that reference alone does not retrieve images or launch a viewer.
- **FHIRcast context synchronization**
 - **Choose when:** Clinicians work across a chart, viewer, or reporting application and need their current patient or study context to stay aligned.
 - **Advantages:** Medplum supports FHIRcast STU3, and its OHIF integration supports context synchronization. This can accompany either storage choice when the participating applications support it.
 - **Tradeoffs:** Define the shared session/topic, participating applications, and supported events. FHIRcast synchronizes context; use DICOMweb for image retrieval and persisted

resources for order/result state. Each application still needs appropriate authentication and access.

Plan reconciliation explicitly: Medplum stores images using its custom DicomStudy, DicomSeries, and DicomInstance resources and Binary files. These are distinct from FHIR R4 ImagingStudy; ingestion does not automatically create an ImagingStudy or link a FHIR Patient. The DICOM patientId is a string, not a patient reference. Use source-aware patient and accession matching, for example in a Bot, and route ambiguous matches for review. If your workflow requires it, create the R4 ImagingStudy and report links after reconciliation.

Image upload and image readiness are also separate: pixel extraction runs asynchronously, so a study can appear before its frames are ready to display. Include a loading/retry path. Do not assume replaying an upload is harmless: study and series creation is conditional on DICOM UIDs, but instance creation is not. Define duplicate-instance handling for retries and backfills.

Record: Required result formats, archive/viewer ownership, supported imaging operations, patient/order reconciliation, and whether context synchronization is needed.

References: [Results and Review](#), [Binary Data](#), [DICOM](#), [DICOM Data Model](#), [DICOMweb API](#), [OHIF Viewer](#), [FHIRcast](#).

3.8 What closes the loop after results arrive?

Questions: Who reviews routine and critical results? Who covers absences? When are patients notified? How are follow-up actions tracked? Does a correction require another review?

Recommended baseline: Keep report finality, clinician review, patient communication, and follow-up completion distinct. `DiagnosticReport.status = final` describes the report; it does not mean a clinician has reviewed it or contacted the patient.

Requirement	Suggested application pattern
One reviewer, no separate queue required	Record acknowledgment and the report version reviewed in your clinical workflow, for example with Provenance. Do not change the report's clinical status to mean reviewed.
Assigned work, coverage, or overdue review must be tracked	Create a Task focused on the report, with the patient in Task . for. Define how to record the reviewer, review time, and version reviewed.
Covering clinician or unassigned result	Route through a defined coverage rule and fallback queue. Avoid leaving ownership implicit in the ordering clinician field.
Critical-result escalation	Implement the clinical team's escalation and acknowledgment policy, including coverage and unresolved alerts. An abnormal flag is input to that workflow.
Patient communication	Define release and communication rules with the responsible team. Track communication separately from reviewing the result.
Follow-up testing, visit, or referral	Create the appropriate follow-up resource and assigned work; define whether it must complete before the original review work closes.

Requirement	Suggested application pattern
Amended or corrected report	Reopen or create review work according to the correction policy, retaining which earlier version was reviewed and communicated.

Use narrowly scoped Subscriptions and Bots for routing. Account for both creation and updates, including meaningful changes that keep the same report status. Make review-work creation idempotent so repeated deliveries do not create duplicate tasks.

Record: Review owner, escalation policy, patient communication rules, correction handling, and what counts as closed.

References: [Task](#), [Subscription Extensions](#).

3.9 Who can access the data and resolve operational failures?

Questions: Which organizations share patients? Can ordering and covering clinicians see the same results? What can collection staff, integration services, and patients read or change? Who owns failed sends and unresolved matches?

Enforce access with AccessPolicy and ProjectMembership. Give each patient-specific resource its own applicable patient reference; access does not propagate merely because a resource links to an order or report. Include documents and their Binary security context in that design.

For Medplum DICOM storage, explicitly design and test access to the DICOM resources and image binaries. Matching a DICOM patient ID or creating an ImagingStudy link does not automatically establish patient-compartment access for the stored images.

Define separate operational ownership for transmission failures, identity exceptions, overdue fulfillment, and unreviewed results. Each queue needs an owner and a resolution action.

Record: Access boundaries and owners for each exception queue.

Reference: [Access Control Decision Guide](#).

Before You Implement

Record these decisions in the implementation plan:

- Services, facilities, integration paths, and supported message types.
- Catalog owner, ordering permissions, required information, and release rules.
- Collection or scheduling responsibilities and completion evidence.
- Identifier mapping, result matching, correction handling, and recovery rules.
- Image storage, viewer, context synchronization, and image-access boundaries when in scope.
- Review, coverage, patient communication, and follow-up ownership.
- Access boundaries and the definition of a closed diagnostic workflow.

Walk through a routine lab order, an imaging order, a rejected or cancelled order, an unmatched result, a corrected report after review, and an absent reviewer. The design should make the next action and its owner clear in each case.