

CMS-0057-F

From requirements *to a working care journey.*

Understand your starting point. Scope the right build.
Know what to expect from your implementation partner.



AI-generated illustrative scene; not a customer or clinical record.

START HERE

A clearer buying decision. A more useful first conversation.

This kit helps you identify the work you need, compare implementation options and explain your priorities to a delivery partner.

YOUR QUESTION	GO TO
Which starting point sounds like our organization?	03 - Buyer decision map
Which requirements and API workstreams belong in our scope?	04-05 - Boundaries and scope worksheet
How do our systems and Da Vinci workflows fit together?	06-10 - Architecture, mapping and reference code
What does a testable, operable implementation require?	11-15 - Controls, evidence and delivery
What should we bring to a useful implementation call?	16-19 - Risks, estimate inputs and project brief
Where can we verify the claims and explore the code?	20 - Linked source library

Use it with three people in the room.

Bring a product or program owner, an integration lead and an operations or security representative. Assign a regulatory owner to confirm applicability. No one person needs to know every answer.

HOW THE WORKSHEETS WORK

Type into the shaded fields, save a local copy, or print the pages. Use **verified**, **partial**, **unknown**, or **not applicable** for evidence status. An unknown is a planning input, not a failing score.

Keep patient data, credentials and confidential attachments out of this workbook. It is an engineering planning aid, not legal advice, an audit or certification. Sources and version boundaries are on page 20.

01 / FIND YOUR STARTING POINT

Which situation sounds like you?

You do not need to replace everything or commission every workstream. Start with the gap that prevents your next decision.

01 We need clarity on scope.

→ READINESS ASSESSMENT

A gap matrix, architecture options, dependency owners and a phased recommendation.

02 We have a platform. Connections are missing.

→ SCOPED INTEGRATION

Source mappings, agreed interfaces, workflow tests and an operating handover.

03 We know the scope. We need delivery capacity.

→ IMPLEMENTATION WORKSTREAM

An engineering team working against named deliverables and acceptance scenarios.

04 We started. The build is not working.

→ RECOVERY REVIEW

A reproducible baseline, prioritized fixes and evidence of what can be retained.

Nirmitee's offer is implementation engineering: assessment, scoped delivery and recovery. Third-party platforms, network participation, regulatory approval and operating support must be explicitly agreed in scope.

02 / RULE BOUNDARIES

Separate the rule from the implementation plan.

Use the final rule as the authority. Use this kit to organize the engineering work. [1]

Confirm who is in scope.

CMS-0057-F covers specified Medicare Advantage, Medicaid, CHIP and federally facilitated exchange QHP payers. Applicability and exact dates differ by payer type. Provider-facing products may have integration work without carrying the same payer obligations.

WORKSTREAM	REGULATORY ORIENTATION [1]
Patient Access	Add specified prior-authorization information to patient-facing access.
Provider Access	Support data access for treatment relationships, attribution and patient opt-out.
Payer-to-Payer	Support patient opt-in and qualifying historical data exchange.
Prior Authorization	Discover documentation requirements and exchange requests and responses.

Dates are not interchangeable.

Operational provisions generally begin in 2026; API dates generally begin in 2027. The 72-hour expedited and seven-calendar-day standard decision timeframes exclude FFE QHP issuers under this rule. The first public prior-authorization metrics were due March 31, 2026.

DO NOT COLLAPSE THESE DISTINCTIONS

The 2024 rule excludes drug prior authorizations. CRD, DTR and PAS are recommended implementation guides, not the entire rule. Record applicable standards and versions separately from the reference software you evaluate. Confirm later rulemaking with your regulatory team.

Planning action: have your regulatory owner approve the applicability record before the delivery team treats it as scope. [1] CMS final-rule fact sheet; source library on page 20.

WORKSHEET 1 / SCOPE

Make the scope explicit.

Record a first pass. Mark unknowns rather than inventing answers.

ORGANIZATION AND POPULATION

Payer program or product; member population; applicable date and who confirmed it.

API WORKSTREAMS AND BOUNDARIES

Patient Access / Provider Access / Payer-to-Payer / Prior Authorization. For each: build, extend, consume or out of scope.

EXISTING CAPABILITY AND EVIDENCE

What already runs? Record the endpoint, document or test reference - not credentials.

ACCEPTANCE OWNER AND FIRST SCENARIO

Who signs off? Describe one end-to-end workflow the implementation must support.

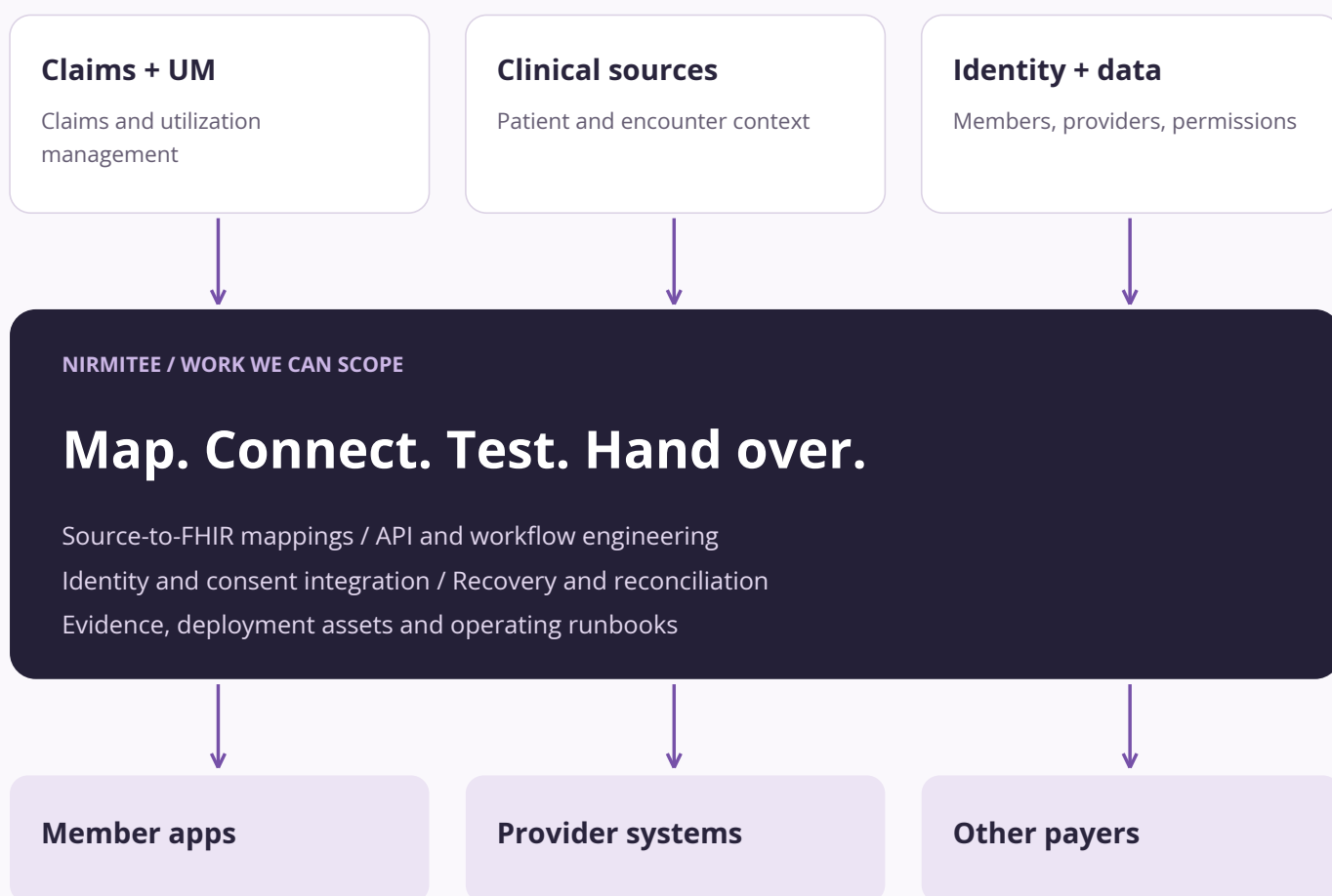
Output: a scope statement your product, engineering and regulatory owners can review together. Not-applicable decisions need a rationale and owner.

03 / ARCHITECTURE

Keep what works. Build the missing connections.

Illustrative scope map: the boundary is agreed after reviewing your systems, access rights and available data.

YOUR EXISTING ENVIRONMENT



A platform can be part of the answer.

If you already use a FHIR platform, we can scope the surrounding integration and workflow work. If the underlying data or interfaces are missing, those dependencies belong in the plan. Keeping the current estate is a design option, not an unconditional promise.

What the buyer receives: an architecture and responsibility map identifying retained systems, new components, supported interfaces, evidence and owners. This is not a claim that Nirmitee supplies an off-the-shelf payer network.

04 / DATA ENGINEERING

Prove the data path before expanding the API surface.

Begin with one representative, synthetic scenario. Build the mapping contract before scaling ingestion.

MAPPING DIMENSION	WHAT TO RECORD
Origin and access	System owner, supported interface, permitted environment, extraction cadence and access dependency.
Identity and linkage	Member, coverage, provider and encounter identifiers; crosswalk owner; unresolved-match handling.
Meaning	Target resource/profile, terminology, units, dates, null semantics and assumptions approved by the domain owner.
Change behavior	Creation, update, correction, deletion or restriction. Decide how consumers see each change.
Quality and recovery	Validation rule, quarantine behavior, retry policy, reconciliation query and accountable operator.

Example: a changed authorization status

A synthetic request moves from pending to approved in the UM system. Trace the event through ingestion, mapping, persistence and the consuming API. Record which timestamp drives freshness, how a duplicate event is handled and how a missed event is recovered.

EVIDENCE TO REQUEST

A mapping specification, representative synthetic fixtures, rejected-record examples and a reconciliation report. “We support FHIR” does not establish that your source data is complete or correctly mapped.

Policy provenance: record the rule and questionnaire version used for each case. Decide how in-progress documentation behaves when a policy changes; retain enough context to explain the original evaluation.

05 / A CARE JOURNEY, EXPLAINED

The request is not finished until the right person knows.

Synthetic imaging-order example. Illustrative workflow only, not coverage or clinical advice.
Reference-server links: page 9. [2-4]



CRD	→ DTR	→ PAS
Is authorization needed?	What evidence is missing?	What is the outcome?
Discover payer requirements.	Collect and review documentation.	Submit, inquire and receive updates.

If more information is needed: the payer reviewer requests evidence; the provider team supplies it; the request resumes; the final outcome returns to the workflow. Define owners, correlation and recovery at each handoff. An acknowledgment is not an approval.

06 / ENGINEERING YOU CAN INSPECT

Read the code. Then ask harder questions.

Nirmitee publishes three payer-side reference servers in Go and .NET. Use them to explore behavior and integration assumptions, not as a turnkey compliance product. [2-4]

CRD / COVERAGE REQUIREMENTS DISCOVERY

Inspect CDS Hooks discovery, coverage responses and sample payer rules. Compare the service behavior with the context your provider workflow can actually supply.

[GitHub / davinci-crd-server](#)

[Published Inferno results and caveats](#)

DTR / DOCUMENTATION TEMPLATES AND RULES

Inspect questionnaire packages, CQL libraries, value sets and adaptive questionnaire behavior. This is the payer side; the client and EHR workflow are separate integration work.

[GitHub / davinci-dtr-server](#)

[Published Inferno results and caveats](#)

PAS / PRIOR AUTHORIZATION SUPPORT

Inspect submission, inquiry and pended-request notifications. Evaluate persistence, decision-system connectivity and subscriber recovery before planning production use.

[GitHub / davinci-pas-server](#)

[Published Inferno results and caveats](#)

INTERPRET TEST EVIDENCE ACCURATELY

Read the test-kit version, configuration, skipped cases and known issues with each report. The DTR repository documents known upstream test-kit issues. Published conformance results are not an independent certification or proof of your end-to-end production readiness.

Repository descriptions reviewed September 13, 2026. This guide does not claim that Nirmitee reference test suites were independently rerun for the PDF.

07 / REFERENCE TO PRODUCTION

The code is a starting point. These are the remaining decisions.

Treat this as a production-gap review, not a criticism of reference software. A reference implementation and a managed service solve different problems.

REFERENCE CAPABILITY	PRODUCTION DECISION / EVIDENCE
Synthetic coverage rules	Who owns real benefit and medical-policy rules? Require reviewed rule fixtures and a change-approval process.
Example request/response	Which source populates every required field? Require source-to-target mappings and representative boundary cases.
Sample state and notifications	Where is durable state? Require restart recovery, duplicate protection, retry limits and reconciliation tests.
Demonstration authentication	Which identities, scopes and trust relationships apply? Require negative authorization tests and key-rotation evidence.
Conformance test reports	Which deployed version and counterparty are covered? Add workflow, load, recovery and operational acceptance evidence.
Local run instructions	Who deploys, monitors and supports the service? Require infrastructure configuration, runbooks and accountable owners.

USEFUL EVALUATION EXERCISE

Run one synthetic scenario through your proposed boundaries. Log every place where a sample rule, in-memory assumption or manual action must become an owned production capability. That list becomes a scoped backlog.

08 / SECURITY AND IDENTITY

Access is a decision. Make it explainable.

Use these engineering review questions with your security and privacy teams. They are not a substitute for a formal risk assessment.

CONTROL AREA	REVIEW QUESTION	EVIDENCE
Identity	Can we distinguish app, user, provider organization and member context?	Identity-flow diagram; token and scope tests.
Permission	What authorizes this specific disclosure? How are restrictions and consent changes applied?	Policy decision examples; revocation tests.
Isolation	Can one tenant or member obtain another's data through search, export or logs?	Negative-access test results.
Secrets	How are credentials issued, rotated and revoked without an outage?	Key lifecycle procedure; rotation exercise.
Audit	Can support explain who accessed what, why and when without copying PHI into logs?	Redacted audit examples; retention policy.
Incident response	Who handles suspected exposure or unauthorized access?	Escalation owner; tested response runbook.

Do not test only valid tokens.

Exercise expired credentials, incorrect audience, insufficient scope, a revoked relationship and an unauthorized export request. Confirm the service fails safely and leaves actionable evidence.

Use synthetic fixtures during evaluation. Agree approved environments and data-handling terms before exchanging any sensitive information.

09 / ACCEPTANCE TESTING

A passing endpoint is not the finish line.

Define acceptance across five layers. Attach evidence to the deployed version, not just a developer workstation.

LAYER	EXAMPLE ACCEPTANCE CHECK	EVIDENCE
Contract	Expected profiles, operations, response codes and terminology validate.	Versioned test report.
Workflow	A request needing more information is completed and reflected in the consuming workflow.	End-to-end trace with synthetic data.
Data	A corrected source record produces the expected API representation without silent duplication.	Reconciliation and mapping results.
Resilience	A dependency interruption recovers without losing work or creating duplicate business actions.	Failure-injection and replay results.
Operations	An on-call operator can identify, triage and resolve the simulated failure using the runbook.	Exercise record and owner sign-off.

Choose explicit failure criteria.

HTTP success or a submission acknowledgment is not an authorization decision. A timeout must not become an approval. A replay must not create an extra business request. Include denial, missing-information and expired-authorization scenarios. Define the response, retry boundary and operator action before implementation.

DEFINITION OF DONE

Scenario + expected result + environment/version + evidence location + named approver. Record open exceptions with risk acceptance and a resolution owner; do not hide them inside a green dashboard.

WORKSHEET 2 / BUILD OR BUY

Compare the boundary. Not the feature list.

Complete this for each platform or delivery partner. Require evidence for each answer.

COVERAGE AND EXCLUSIONS

Which workflows are demonstrated? Which are roadmap items, custom work or unsupported?

INTEGRATION OWNERSHIP

Who connects sources, maintains mappings and resolves data-quality gaps? What access is assumed?

OPERATING MODEL AND COSTS

Licensing, usage, infrastructure, implementation, support and change costs. Who owns incidents?

EVIDENCE AND EXIT PATH

What can we test before committing? How do we export data, retain code access and transition support?

Decision rule: compare the same workflow and responsibilities across candidates. A lower license price can hide a larger integration scope. This worksheet does not recommend a vendor or invent a cost estimate.

10 / WHAT YOU RECEIVE

Not just working software. A build your team can own.

Illustrative deliverable previews. Final documents, code access and acceptance criteria are agreed in the statement of work.

SCOPE PACK	EVIDENCE PACK	HANDOVER PACK
BOUNDARY APIs, workflows, exclusions	SCENARIO Missing evidence / resumed PA	OPERATE Deploy / monitor / recover
DEPENDENCY Source access / named owner	RESULT Expected vs. actual behavior	ESCALATE Source, platform, integration
DECISION Build, extend or integrate	RECORD Version / trace / approver	OWNERSHIP Runbook / owner / review

HOW DELIVERY PROGRESSES

01	Agree scope	Named boundaries, assumptions and owners.
02	Prove a slice	One synthetic workflow validates the key dependencies.
03	Build and validate	Selected scenarios pass; exceptions are recorded.
04	Transfer operations	Your receiving team demonstrates support and recovery.

What is agreed before work begins: repository access, IP and component licenses, deployment accounts, responsibilities, evidence, fees and change control. Timing follows source readiness and approvals; these gates are not a fixed-duration promise.

11 / OPERATIONAL HANDOVER

Design for the day after go-live.

The API belongs to an operating system of people, alerts, rules and recovery procedures.

OPERATING SIGNAL	QUESTION TO RESOLVE
Freshness and reconciliation	How do we detect stale or missing source changes? Who reconciles the backlog and verifies recovery?
Pending workflow age	Which requests need attention, by which owner? Separate transport health from business processing status.
Error and retry behavior	Which failures retry automatically? Where do exhausted retries go? Who can safely replay them?
Access and usage	Which anomalous access patterns trigger investigation? Who reviews audit evidence and retention?
Change compatibility	How are profile updates, source-schema changes and payer-rule changes tested before release?
Support accountability	Who owns source outages, platform defects and integration failures? What escalation and coverage are contracted?

Run the handover exercise.

Give the receiving team a simulated delayed update and a failed notification. Ask them to locate the cause, identify the affected synthetic records, recover safely and verify the final state using only the supplied runbooks.

Choose alert thresholds and service objectives from the agreed workflow and operating model. This kit intentionally does not invent uptime or response-time commitments.

WORKSHEET 3 / RISKS

Name the blocker. Name the owner.

Use a short risk record that leads to a decision, not a generic red-amber-green score.

HIGHEST-IMPACT DEPENDENCY

What must be true for delivery to proceed? Example: permitted access to a source-system test environment.

EVIDENCE STATUS

Verified / partial / unknown / not applicable. What evidence supports that status?

OWNER AND NEXT ACTION

Who can resolve it? What action will they take, and by what agreed review date?

FALLBACK AND DECISION NEEDED

If this stays unresolved, what changes in scope, architecture or sequence? Who approves that choice?

Reuse this page for each critical risk. Prioritize dependencies that affect multiple workstreams, not only the most visible development task.

WORKSHEET 4 / ESTIMATE INPUTS

A useful estimate starts with a visible scope.

Supply these inputs before asking for a committed implementation price or delivery date.

WORKFLOW AND INTEGRATION COUNT

Selected APIs; consuming applications; source systems; counterparties; environments and required variations.

DATA AND ACCESS READINESS

Available interfaces, synthetic samples, mapping coverage, historical load and update behavior.

NON-FUNCTIONAL AND REVIEW NEEDS

Expected volumes, recovery objectives, security review, procurement and acceptance participants.

CONSTRAINTS AND COMMERCIAL BOUNDARIES

Target milestone and rationale; internal capacity; platform choices; what should be included or excluded.

Ask for assumptions, exclusions, dependencies and change-control terms alongside the estimate. A precise number without those boundaries can be less useful than an honest range.

WORKSHEET 5 / BRING THIS TO THE CALL

Your one-page implementation brief.

A high-level summary is enough to start. Keep this free of patient data and confidential attachments.

WHAT NEEDS TO WORK?

Describe the buyer, user and outcome in one or two sentences.

WHAT EXISTS TODAY?

Name the system types, available APIs and work already completed.

WHAT IS GETTING IN THE WAY?

Choose the biggest unknown: scope, data, integration, testing, capacity or operations. Add context.

WHAT DECISION SHOULD THE CONVERSATION HELP YOU MAKE?

Assessment, selected workstream, platform integration or recovery review? Note who should join.

[Book an implementation call with Nirmitee](#)

12 / YOUR NEXT STEP

Have the checklist. Need the implementation team?

Bring your biggest blocker. Let's decide what a useful first engagement would look like.

STARTING POINT	WHAT WE CAN SCOPE TOGETHER
Readiness assessment	Gap matrix, system inventory, mapping priorities, architecture options, risks and a sequenced implementation recommendation.
Implementation workstream	Selected API and integration scope, versioned code, test evidence, deployment assets and operational handover.
Existing-build review	Reproducible baseline, conformance and workflow gaps, salvageable components and a prioritized recovery plan.

The first conversation

We discuss your system landscape, target workflows and current blockers. Together we identify the information needed to scope the next step. Detailed assessment, project deliverables, fees and timelines are agreed separately before work begins.

**Your systems. Your blockers.
A practical next step.**

[Book an implementation call >](#)



cal.com/yogeshdaga/webcalls

Prefer to share context first? Visit nirmitee.io/get-in-touch/. No presentation is required. Do not send patient records or credentials. Agree confidentiality arrangements before sharing sensitive project material.

SOURCE LIBRARY / KEEP EVALUATING

Standards change. Keep the evidence close.

Edition 02. Sources reviewed September 13, 2026. Recheck applicability, published versions and repository status when you begin a project.

1 / CMS final-rule fact sheet

Regulatory orientation, API workstreams, dates, exclusions and required versus recommended standards.

[Open source reference >](#)

2 / Nirmitee CRD reference server

Coverage-discovery implementation, setup and published Inferno results.

[Open source reference >](#)

3 / Nirmitee DTR reference server

Documentation templates and rules; published test results include known test-kit caveats.

[Open source reference >](#)

4 / Nirmitee PAS reference server

Prior-authorization submission, inquiry and notification examples; review production assumptions.

[Open source reference >](#)

What this kit does - and does not - establish

The architectures, questions, worksheets and acceptance examples are Nirmitee planning recommendations. They are not regulatory text. The kit does not establish legal applicability, certification, complete organizational compliance or a completed payer-wide production reference.

Use and attribution

Prepared by Nirmitee.io. You may share this kit with your internal evaluation team. Repository licenses apply separately. HL7 and FHIR are HL7 International trademarks; no CMS or HL7 endorsement is implied. Healthcare scenes are AI-generated illustrations, not customer photographs. Document previews are illustrative, not client deliverables.

Fillable fields may behave differently across PDF viewers. For reliable editing, use a desktop PDF reader and save a local copy. Blank fields are intentional worksheet spaces, not missing content.