Data Migration Decision Guide

Companion to the Migrating to Medplum docs.

Section 1 describes your current state and goals. Section 2 identifies the factors that change your migration approach. Section 3 helps you make and document the resulting decisions.

Section 1: Migration Context & Participants

These questions establish what is moving, why it is moving, and who owns each decision before implementation begins.

1.1 What outcome defines success?

- Replace a legacy clinical or operational system
- Consolidate several systems into one Medplum project
- Backfill history while the source remains active
- Seed selected data for a new workflow
- Preserve a read-only archive
- Build a repeatable onboarding pipeline for future sources

Why: determines whether this is a finite conversion, ongoing synchronization, or a reusable product capability.

1.2 Which sources and interfaces are in scope?

For each source, identify:

- Business and technical owners
- Extraction method and available formats
- Data volume and rate of change

- Stable identifiers
- Time zone, locale, and encoding conventions
- Known quality issues and source variants
- Deletion, merge, and correction history
- Whether the source remains active after cutover

Why: a database snapshot, FHIR endpoint, HL7 feed, document archive, and DICOM repository need different extraction and reconciliation plans.

1.3 What data and history are required at go-live?

Consider:

- People, organizations, locations, and coverage
- Encounters, appointments, current clinical state, and clinical history
- Forms, communications, tasks, consents, and other workflow state
- Documents, files, and imaging
- Billing, terminology, profiles, and configuration

For each domain, choose full history, current records only, a date range, summary data with legacy archive access, or metadata with files retained elsewhere.

Why: "all data" is not an actionable scope. Each domain needs an explicit boundary, owner, target, and acceptance criteria.

1.4 Where is each domain authoritative during and after migration?

- Source remains authoritative
- Medplum becomes authoritative at cutover
- Authority changes in phases
- An external system remains authoritative for a domain
- Both systems coexist under an explicit conflict policy

Why: coexistence without one write authority per domain creates conflicts and makes reconciliation inconclusive.

1.5 Who approves the migration?

Assign accountable owners for:

- Source meaning and extraction
- Patient identity
- FHIR mapping and profiles
- Terminology and clinical interpretation
- Security and compliance
- Pipeline operations and reconciliation
- Go-live and rollback

Why: many migration decisions require clinical, operational, and compliance judgment rather than engineering judgment.

1.6 What operational change accompanies the migration?

- Which user workflows, reports, and integrations change at cutover?
- How much planned downtime can each workflow tolerate?
- Which users need training, and who owns readiness?
- Who communicates the freeze, cutover, rollback, and return-to-service status?
- What support coverage and post-go-live support period are required?
- How are user-reported data discrepancies triaged after cutover?

Why: technically correct data is not a successful migration if users cannot safely perform their work after go-live.

Section 2: Approach-Defining Factors

Every migration must define scope, identifiers, mapping ownership, validation, security, testing, cutover authority, and acceptance. The matrix below identifies additional complexity, not optional safeguards.

Mark each factor Yes, No, or Unknown. For every Yes or Unknown answer, follow the linked decision section.

#	Factor	What it changes	Decision section	Answer
1	Multiple source systems, versions, or local variations	Inventory, identifier systems, and mapping layers	3.1, 3.3	
2	Cross-source or existing-record patient matching	Identity workflow and reference sequencing	3.2	
3	Automated merge execution during migration	Review thresholds, audit, and reversal risk	3.2	
4	Custom profiles, extensions, or implementation-guide conformance	Mapping approval and validation configuration	3.3, 3.4	
5	One source record producing several linked FHIR resources	Stable IDs, all-or-nothing writes, and reconciliation	3.3, 3.5	
6	Expected load time may exceed the cutover window or service limits	Import path, throughput testing, and recovery	3.5, 3.7	
7	The source remains active while data moves	Incremental changes, write authority, and conflicts	3.6	
8	Little or no downtime is acceptable	Change capture, rehearsal depth, and rollback	3.6, 3.7	

#	Factor	What it changes	Decision section	Answer
9	Documents, files, images, or DICOM	Storage, access, integrity, and retrieval testing	3.9	
10	Historical workflows must remain operational	Domain-specific modeling and validation	3.3, 3.4	
11	External systems remaining authoritative after cutover	Long-term ownership and synchronization	3.6, 3.10	
12	Reusable onboarding or ongoing synchronization pipeline	Operational ownership, versioning, and reconciliation	3.1, 3.10	

Section 3: Decision Deep Dives

Complete 3.1 through 3.8 and 3.10 for every migration, adjusting the depth to your situation. Complete 3.9 when files, imaging, or another complex data type is in scope. For each section, document the decision, accountable owner, and acceptance criteria.

Define What Will Move

3.1 Scope, Inventory & Ownership

Decide what migrates, how far back each domain goes, and who can accept the result. See Planning your Migration.

Recommendation: Profile the qualifying source population before committing dates, volumes, or validation samples. Give every domain explicit inclusion rules, exclusions, and an acceptance owner.

Questions:

- Which domains and history are required on day one, and what is excluded?
- Which sources contribute to each domain, and can they produce a consistent snapshot or a reliable list of later changes?
- What are the qualifying record counts, file volumes, and change rates?
- Who owns source meaning, mapping, acceptance, and retained legacy access?

Situation	Approach
Small, inactive source	Use a finite extract, transform, load, and reconcile process with a defined freeze window.
Active source	Take a baseline snapshot, then process later changes from a recorded source position until cutover.
Several sources	Keep separate inventories, identifier systems, mappings, and reconciliation results for each source.
Only current state is required	Define “current” separately for each domain; do not apply one generic filter to all resources.
Full history stays in a legacy archive	Define what moves, how older data is retrieved, how long the archive remains, and who owns access.
Scope is uncertain	Profile representative source data before committing dates or sample sizes.

Output: Complete the Source and Domain Inventory.

3.2 Identity, Identifiers & Deduplication

Decide whether a source record remains distinct, maps to an existing resource, or enters a master-record workflow. See Patient Deduplication Architectures and Patient `$match`.

⚠️ **One-way door:** Loading dependent clinical data before resolving identity can spread references across duplicate patients. Later merges may require extensive repair and clinical review.

Recommendation: Preserve source identifiers using a different identifier system URI for each source. Keep ambiguous matches separate and require human review unless an approved matching policy permits automatic action.

Questions:

- Which source keys are stable and unique, and which identifier system URI represents each source?
- Where can duplicates occur: within a source, across sources, or against Medplum?
- What evidence and risk threshold permit automatic matching versus human review?
- Will migration merge records, or only report candidates for later remediation?

Situation	Approach
Stable source key	Preserve it in `identifier` under a source-specific system URI; use it for writes that are safe to repeat.
Different systems reuse the same value	Use distinct identifier system URIs; never treat the bare value as globally unique.
Source patient may match an existing Patient	Run the approved identity workflow before loading dependent records.
Match is ambiguous	Keep records separate and route to human review.

Situation	Approach
Existing target ID is not known	Use a conditional reference only when the identifier finds exactly one resource and the migration identity can search it.
Related resources are created together	Use a transaction with `urn:uuid` references.
Related resource was loaded earlier	Use a source-to-target ID lookup or a conditional reference that resolves uniquely.
Keys are missing or unstable	Define a governed replacement key before loading; do not use mutable demographics as the key for repeated runs.

A source identifier answers “which source record is this?” It does not by itself answer “which real-world person is this?”

3.3 Mapping & Terminology Governance

Decide how source meaning maps to FHIR R4 and how decisions are versioned and approved. See Governing Data Mappings and Converting Data to FHIR.

⚠️ **One-way door:** Changing identifier systems, profile or extension URLs, resource boundaries, or code systems after applications depend on them can require both data and application migrations.

Recommendation: Version mappings independently from pipeline code, preserve source meaning when no verified standard mapping exists, and require domain review before production.

Questions:

- Which FHIR resources, fields, profiles, and terminology represent each source concept?

- Which mappings are direct, conditional, one-to-many, unsupported, or excluded?
- How are missing, ambiguous, free-text, and source-specific values handled?
- Are profiles declared per resource in `meta.profile` or applied through `Project.defaultProfile`?
- Who approves changes, and which records must be reprocessed when a mapping changes?

Situation	Approach
Direct mapping	Record source meaning, target path, transformation, null behavior, and validation.
One source entity spans several concepts	Create distinct resources and preserve relationships.
Trustworthy standard code	Preserve its system, code, and display under the approved mapping.
Local terminology only	Preserve a governed local code and text; standard-code enrichment is a separate review.
No verified code mapping	Retain source meaning and route for review; do not invent a code.
No standard field fits	Check established extensions and profiles before defining a custom extension.
One profile applies by resource type	Configure `Project.defaultProfile`; it applies when the resource has no `meta.profile`.
Profile varies by source record	Set the applicable profile explicitly in `meta.profile`.

Situation	Approach
Mapping changes after rehearsal	Version the mapping and code together; define reprocessing or cleanup.

Use the mapping register example to define the information your project must track. Mapping approval is distinct from code review.

Prove It Can Move Safely

3.4 Validation, Reconciliation & Acceptance

Decide how the team will prove structural conformance, referential integrity, semantic correctness, completeness, and workflow readiness. See Validating and Reconciling Migrated Data and Testing and Accepting Migrated Data.

Recommendation: Validate every record structurally and account for every eligible source record. Supplement automated checks with repeatable, random, and risk-based chart review; require named sign-off.

Questions:

- What must reconcile exactly, and which differences are expected and explainable?
- Which relationships, values, and high-risk cohorts require deeper review?
- How is every success, rejection, skip, retry, record held for review, and exception reported?
- Which evidence and thresholds must each stakeholder approve?

Situation	Approach
One source row maps to one resource	Reconcile eligible, successful, rejected, skipped, and held-for-review counts.
One source row maps to several resources	Reconcile each target type and expected relationship separately.

Situation	Approach
Deduplication changes counts	Report source records, master records, match decisions, and unresolved candidates separately.
Profile conformance is required	Confirm the intended profiles are installed and selected and terminology validation is configured, then run `$validate`; structural validity is not clinical proof.
Clinical meaning cannot be proven by counts	Use repeatable, reproducible random, and risk-based whole-chart samples.
Exceptions remain	Track owner, severity, remediation, acceptance impact, and approval.

Do not prescribe a universal verification percentage. Choose samples after exact extraction counts are known and adjust for variability, risk, complexity, and defect history.

3.5 Load Architecture & Production Operations

Decide which writes must succeed or fail together, how the pipeline resumes, and how much throughput the cutover requires. Implementation details and project prerequisites are covered in Building Migration Pipelines.

Recommendation: Use asynchronous batches for high-volume independent writes and synchronous transactions only for small groups that must succeed or fail together. Make every path safe to rerun, observable, and resumable.

Questions:

- Which writes are independent, and which small groups must succeed or fail together?
- Which import path and measured throughput fit the data shape and cutover window?

- How are results, checkpoints, retries, and rejected resources recorded?
- Which downstream automations run, and what thresholds pause the load?

Output: Choose a load path, define groups that require all-or-nothing writes and source-of-truth boundaries, and document recovery and side-effect handling in the pipeline design.

Plan the Transition

3.6 Cutover, Incremental Changes & Coexistence

Decide when Medplum becomes authoritative and prevent two systems from independently accepting authoritative writes. See Phased Adoption and Cutover.

⚠️ **One-way door:** A cutover is unsafe when two systems accept authoritative writes for the same domain without deterministic synchronization and conflict ownership.

Recommendation: Choose the smallest cutover unit that preserves one clear write authority and can be rehearsed end to end. Use a phased transition only when domains, sites, or cohorts can be separated without ambiguous ownership.

Questions:

- Is transition big bang, phased, parallel, or backfill-only?
- Which system owns reads and writes for each domain and phase?
- How are changes since the previous extract captured, and can any path bypass the authoritative system?
- How are dual-write failures and conflicts handled?
- What evidence triggers go-live, pause, rollback, and legacy retirement?

Situation	Approach
Small, low-change source	Freeze, extract the final changes, load, verify, and switch.

Situation	Approach
Higher operational risk	Prefer phased transition by domain, site, tenant, or user cohort.
Baseline plus incremental changes	Record the last source change processed and use overlapping windows that are safe to run again.
Temporary dual write	Keep one authoritative path, capture failures durably, and reconcile continuously.
Medplum reads start first	Use feature flags and compare behavior while writes remain in the source.
Legacy becomes an archive	Make it read-only, test access, and disable remaining write integrations.

Output: Complete the Authority and Coexistence Matrix.

3.7 Testing & Dress Rehearsals

Decide how closely testing must match production volume, infrastructure, access, and workflow behavior. See Testing and Accepting Migrated Data.

Recommendation: Run repeatable mapping tests and rehearse interruption, recovery, reconciliation, access, workflows, and rollback. Use production-scale data and infrastructure when volume, performance, or the cutover window creates material risk.

Questions:

- Which representative data and known edge cases can testing safely use?
- Does the test environment match production features, access, integrations, and scale?
- Have safe reruns, interruption, recovery, throttling, and partial failure been tested?

- Have users rehearsed workflows, documents, cutover, and rollback?

Situation	Approach
Deterministic mapping	Unit test normal, null, malformed, boundary, repeated, and unsupported cases.
Unknown source variation	Profile production-like data and add each discovered defect to regression tests.
Strict cutover window	Rehearse full volume on production-like infrastructure and measure every stage.
Pipeline can be interrupted	Stop and resume during rehearsal; verify checkpoints and counts.
Clinical workflows depend on migrated data	Run user acceptance tests with representative patient journeys.
Access varies by role or tenant	Test allowed and denied access using actual migrated references and compartments.

A small pilot does not prove production throughput. A full-volume load does not prove clinical meaning. Test both when those risks apply.

Protect the Data and Hand Off Operations

3.8 Security, Compliance & Audit

Decide how protected data is extracted, staged, transmitted, logged, retained, and destroyed. See Access Policies and HIPAA Compliance.

Recommendation: Use dedicated least-privilege credentials, approved encrypted storage, PHI-minimized logs, and explicit retention and disposal rules for every temporary artifact.

Questions:

- Which people, systems, and environments may access migration PHI?
- How are extracts, credentials, logs, rejected records, and reports protected?
- What audit, retention, revocation, and secure-disposal rules apply?
- What incident process handles a wrong-patient, wrong-tenant, or wrong-project load?

Situation	Approach
Production automation	Use a dedicated `ClientApplication` and narrowly scoped `AccessPolicy`, not a personal token.
Extracts contain PHI	Use approved encrypted storage, restrict access, and destroy temporary copies after acceptance.
Logs contain source data	Minimize fields and apply regulated-data access and retention controls.
Production-like test data is needed	Prefer synthetic or de-identified data unless approved controls authorize PHI.
Multi-tenant target	Define the tenant `AccessPolicy`; populate each resource's own patient references and/or governed `meta.account`; test allowed and denied access.
Wrong-scope load	Stop the pipeline, preserve evidence, assess exposure, and use a reviewed correction plan.

Security review covers the pipeline and temporary artifacts, not only final FHIR resources.

3.9 Documents, Binaries & Complex Data

Decide whether complex data becomes structured resources, binary content, metadata, external references, or a separate workstream.

Recommendation: Treat each complex domain as its own scoped workstream. Do not load it until its FHIR model, content storage, access rules, integrity checks, and retrieval workflow are approved.

Questions:

- Which complex domains are in scope, and must they remain operational or only viewable?
- Are original bytes, structured data, searchable metadata, or all three required?
- How are identity, clinical context, access, integrity, and retrieval verified?
- Which unsupported or ambiguous data needs human review or legacy retention?

Use Binary Data, the Messaging Data Model, and the DICOM Data Model rather than defining those models inside the migration plan.

3.10 Monitoring, Exceptions & Handoff

Decide how production runs are controlled and when migration ownership transfers to normal operations.

Recommendation: Assign service ownership before production. Keep exceptions visible through resolution, define automatic pause thresholds, and close the migration only after operational handoff and temporary-access cleanup.

Questions:

- Which metrics, alerts, and thresholds pause a run?
- Who owns technical, identity, clinical, and terminology exceptions?
- Does the pipeline end or become an ongoing production integration?
- What ends the post-go-live support period and permits credential, staging, and source-access retirement?

Situation	Approach
Finite migration	Define completion, archive evidence, revoke credentials, and dispose of temporary data.

Situation	Approach
Ongoing synchronization	Treat it as a production integration with service ownership, alerts, runbooks, and reconciliation cadence.
Transient failure	Retry with bounded backoff from a durable checkpoint.
Repeatable mapping or data failure	Hold the record for review, fix the mapping or data, add a regression test, and safely rerun affected records.
Identity ambiguity	Route to the identity-review owner, not a generic retry queue.
Clinical ambiguity	Route to the clinical or terminology owner.
Error threshold exceeded	Pause the affected stream, preserve state, and resume only after approval.
Users need new workflows	Assign training, communications, support, and readiness owners before cutover.

Before You Implement

Record:

- Data scope, history boundaries, and exclusions
- Identifier, matching, and mapping decisions
- Load path, recovery approach, and operational owner
- Write authority, cutover approach, and rollback criteria
- Validation evidence, acceptance thresholds, and approvers
- Security controls and unresolved risks

Do not begin production migration work while any decision that changes identity, clinical meaning, write authority, or acceptance remains unowned or unresolved.