WP-2026-006 — The Migration Suite: Migration Engine as a Vendor-Neutral Accelerator, and a Complete Path for Legacy-PACS Retirement (Whitepaper)

Document number: DOC-2026-451

Whitepaper: WP-2026-006

Version: 3

Category: Marketing Material

Product family: The Migration Suite — XyDromatics Migration Engine + XyDromatics Migration + SynthInSight

Owner: Synthology Healthcare Solutions Group, LLC (SHSG)

Audience: Imaging / PACS directors, migration-project leads, enterprise-imaging architects, IT infrastructure & security, procurement

Regulatory posture: XyDromatics Migration Engine and XyDromatics Migration are **non-device software under FD&C Act §520(o)(1)(D)** — they transfer, store, convert, and display medical device data without clinical interpretation. **SynthInSight** is **general-purpose business software** (operational analytics); it is not a medical device and has no clinical function. Nothing in this whitepaper describes a medical device.

Companion: Renders on the marketing site at `/whitepaper/migration-suite` (after approval)

Cross-references: RI-2026-366 (Migration Suite); RI-2026-337 (federated Q/R proxy); RI-2026-338 / DCR-2026-1856 (instance-level forward delta)

A note on how this paper is written

Every capability described below is grounded in the shipped product. Where a capability is **licensed** or **off by default**, this paper says so. Where a claim would require a **measured benchmark**, this paper does not make it: Synthology does not publish throughput, "N× faster," or "reduces load by N%" figures for the Migration Suite because those numbers depend entirely on the customer's source PACS, network, and study mix. What we describe is **architecture and behavior** — what the software does and does not do — not a performance promise. The section "What we do not claim" states the boundaries explicitly.

Executive summary

Retiring a legacy PACS is two problems at once: **moving the data**, and **not disrupting the clinical reads** that are still happening on the source archive while you move it. The Migration Suite addresses both, and it is

designed so a customer can adopt it **incrementally** — starting with a single product that complements a migration tool they already own, and expanding to a complete, single-vendor workflow if and when they want it.

- **Migration Engine is vendor-neutral.** It can front the legacy source(s) as a standards-based DICOM Query/Retrieve provider: a downstream migration server — Synthology's own, or a third party's — issues one query and Migration Engine answers for the whole legacy estate. This lets a customer keep the migration tool they already have and put Migration Engine in front of it to take query load off the live production PACS. *(Licensed capability; see §2 for the exact protocol scope and configuration it requires.)*
- **XyDromatics Migration is the migration server.** When a customer wants the migration itself from Synthology, XyDromatics Migration **normalizes non-conformant DICOM to PS3.x**, de-duplicates studies, remediates DICOM tags by pre-authored rules, applies configurable quality gates, and reconciles source against destination when a batch completes.
- **SynthInSight is one PHI-free view of the whole migration.** Both engines report aggregate counts — how many jobs completed, failed, or were skipped — to a single analytics home. No patient, study, or accession data is in that feed.
- **PHI stays on the on-premise image path.** Only de-identified aggregate counts reach SynthInSight.

Migration Engine and XyDromatics Migration are non-device software under FD&C Act §520(o)(1)(D); SynthInSight is general-purpose analytics.

---

§1 — The problem

1.1 A migration competes with clinical work

A legacy-PACS retirement is not a maintenance-window job; it runs for weeks or months, and for most of that time the legacy archive is still the system of record for clinical reads. A migration tool that queries and pulls directly from that live archive is competing with radiologists for the same connections and the same I/O. The usual mitigations — throttling the migration, running it only overnight — trade migration duration against clinical impact, and neither removes the coupling.

1.2 "Just copy it" produces a worse archive

Bit-for-bit copying moves the legacy archive's accumulated problems into the new one: duplicate studies from prior partial migrations, tag inconsistencies from years of different modalities and HL7 feeds, and studies that never should have been retained. The new PACS inherits the old one's data-quality debt.

1.3 Nobody can see the whole migration

Migration progress typically lives in whatever tool is doing the moving. When more than one tool is involved — a query proxy here, a migration server there — there is no single, consistent view of how many studies have moved, failed, or been skipped, and building one usually means exporting logs that carry PHI.

The Migration Suite is organized around these three problems: **offload the live source, clean the data in transit, and see the whole thing without moving PHI.**

---

§2 — Migration Engine: the vendor-neutral accelerator

2.1 What it is

Migration Engine is a **standalone, non-device data-mover**. Its defining capability for a migration is a **federating DICOM Query/Retrieve provider**: with the federation feature licensed and enabled, Migration Engine presents a single DICOM Q/R endpoint that fronts *N* configured legacy source PACS. A downstream DICOM client — a third-party migration server, or Synthology's own XyDromatics Migration — connects to Migration Engine as if it were one archive:

- On **C-FIND** (query), Migration Engine fans the query across every configured member source **concurrently**, then returns a **single, de-duplicated, merged** study list (matched by Study Instance UID, with Accession Number as a fallback), tagging each result with which source holds it. The client sees one unified catalog instead of having to query each legacy system separately.
- On **C-MOVE** (retrieve), Migration Engine retrieves the study from the holding source(s) and delivers it to the requested destination.

This is what "vendor-neutral" means concretely: **the migration server you already own can point at Migration Engine as its source**, and Migration Engine handles talking to the real legacy systems. The federation behavior is verified end-to-end by an automated real-host wire test (multiple source servers → Migration Engine → a receiving sink), asserting that the exact instances arrive.

2.2 What it requires (stated plainly)

This is a **licensed capability** (`federated_qr`) and is **off by default** — an operator enables it and configures it. In particular:

- The federating provider speaks **C-FIND and C-MOVE** (plus C-ECHO and Modality Worklist). It does **not** offer **C-GET** to downstream clients; a migration server that only supports C-GET retrieval is not a fit for this path.
- Migration Engine retrieves in a **buffered (store-and-forward)** mode (see §3.1). For a study to be delivered onward, the **C-MOVE destination the client names must be known to Migration Engine** — either Migration Engine's own receiving endpoint or a **destination PACS the operator has registered as a target**. A third-party destination that is not registered is staged but not forwarded. This is a one-time configuration step, not a per-study action.
- Members of the federation are **DICOM (DIMSE) query sources**. A DICOMweb source participates in query federation through a separate mechanism (see §2.3), not as a member of this DICOM Q/R proxy.

We state these boundaries because a migration architect needs them to design the cutover — not as fine print after the sale.

2.3 DICOMweb sources

Migration Engine works with DICOMweb archives in two distinct, licensed ways:

- **As a migration source (`dicom_web_source`):** Migration Engine queries a source over **QIDO-RS** and retrieves its studies over **WADO-RS**, then forwards them to the target — a complete DICOMweb-to-target migration path.

- **In federated query (`federated_dicomweb`):** a **QIDO-RS** archive can join the federated query fan-out so its studies merge into the same unified result alongside DICOM (DIMSE) sources (aligned with IHE XCA-I). This path is **query-only** (QIDO); it does not itself retrieve via WADO.

2.4 Scale controls

- **Parallel retrieval (`parallel_cmove`):** Migration Engine can migrate multiple studies concurrently via a configurable worker pool; without the feature it runs one study at a time. This is a concurrency control, not a benchmarked speedup.
- **Rate limiting (`rate_limiting`):** per-source and global throttles — a requests-per-second cap and a megabytes-per-second throughput cap — protect an older source PACS and the downstream target during a large migration. Enforcement is verified on the running product by an in-process self-test that confirms throttling actually occurs.

---

§3 — What Migration Engine does in transit (independent of the downstream tool)

3.1 Query the source once, then forward from a buffer

Migration Engine uses a **store-and-forward** model: each study is retrieved from the source **once per successful retrieval** into local staging, and then forwarded to the destination **from that buffer**. The write leg to the destination never reaches back to the legacy source. (A retrieval that fails or returns partially is retried against the source; the per-study instance check in §3.3 then avoids re-sending instances already delivered.) In the federated path, Migration Engine pulls each holder to its own staging endpoint, assembles the study, then forwards the assembled study to the caller.

The practical effect is that the legacy source answers the migration once per study and is otherwise left alone — the offload described in §1.1.

3.2 Optional lossless transcoding

On the federated buffered path, Migration Engine can **losslessly transcode** each study's images to an operator-chosen transfer syntax before delivery (`transcode_target_ts`, opt-in and licensed). Permitted targets are **lossless only** — JPEG Lossless, JPEG 2000 Lossless, and the uncompressed syntaxes. The step is **fail-closed**: if a required codec is missing or a non-lossless target is requested, that instance is aborted rather than forwarded incorrectly. Non-image objects (structured reports, key-object selections, presentation states, encapsulated PDFs) pass through unchanged. This is a data-format conditioning step; it does not alter image content and is never lossy.

3.3 Skip re-sending what the target already has (and report overlap)

Migration Engine has two distinct, real mechanisms related to studies already present at the target. It is important to describe them precisely:

- **Instance-level forward delta (on by default):** before forwarding a study, Migration Engine asks the target which SOP Instances it already holds and forwards **only the missing instances**. This avoids re-

sending images the destination already has. If the target cannot be queried, the full study is forwarded (fail-safe).

- **Target-overlap detection report (optional analysis, `dedup_against_target`, off by default):** for planning, Migration Engine can inventory the source and query the target to classify each study's overlap (absent / partially present / present-by-count / unknown), keyed on Study Instance UID by instance count. This is a **report** — it does not skip, drop, or suppress anything.

What Migration Engine does not do: it does not currently *skip pulling* a study from the source because the target appears to already have it. Pre-retrieval suppression is deliberately not shipped — a study is still fully retrieved from the source; only re-*forwarding* of already-present instances is avoided. We call this out because it is an easy capability to overstate.

3.4 Optional data transforms before forwarding

Migration Engine can apply **operator-authored data transforms** to each instance **just before forwarding** (`data_transforms`): **tag transformations** — add, modify, copy, regex-replace, or delete tags by pre-authored, priority-ordered rules — and **normalization**, the PS3.x-conformance policy sweeps described in §4.2. Rules are per-project and use the same shared rule model and algorithm as XyDromatics Router, so a rule authored once behaves identically across products. As on XyDromatics Migration, an edit that changes a clinical-meaning tag is **not blocked**; it is recorded with an "advisory" flag on the audit trail. With no rules configured, the step is a no-op.

---

§4 — XyDromatics Migration: the migration server

When a customer wants the migration itself from Synthology, **XyDromatics Migration** is the migration server. It is a standalone application that moves imaging over standard DICOM networking (C-FIND / C-MOVE to source and destination peers) and forwards HL7 reports over MLLP; its jobs and pipeline configuration persist in local files. It co-hosts its own vendor-neutral archive, so migrated studies land in an archive it controls (it is a migration server *with* an archive, not an archive-less relay).

4.1 De-duplication (on by default)

Before each retrieval, XyDromatics Migration runs an **idempotent pre-flight**: if a study is already present in the migration target, it is **skipped** (or narrowed to an image-level top-up of only the missing instances). Identity is the **Study Instance UID**, and "already migrated" is judged by comparing the source's reported instance count against the local archive's count. It is **on by default** and can be bypassed per-batch (force a full re-pull) or per-source. It is fail-safe: if the source does not report an instance count, or the local archive is unavailable, it falls through to a full retrieval, and it never deletes anything. This is a **study-level skip** optimization by UID and counts — not a byte-level or hash-based de-duplication.

4.2 Normalization — PS3.x conformance, first in the ingest chain

XyDromatics Migration runs a dedicated **DICOM Normalization Engine** as the **first** step of its ingest pipeline, so every later step — validation, tag remediation, quality gates — operates on **PS3.x-compliant** input. It is a set of canned, toggleable conformance policies (not free-hand editing), including: trim / collapse whitespace, canonicalize dates and times, regenerate malformed UIDs and general UID hygiene, strip private tags, force the

character set to UTF-8, normalize Patient Name caret (^) delimiters, file-meta sanity, SOP-class consistency, and VR-length compliance (report or truncate). Policies are grouped into presets (off / baseline / strict / custom) and are **configurable and off unless enabled**. On a policy failure the operator chooses the per-instance behavior — log-and-continue (the default) or divert the instance to the rejection queue. This is what keeps years of accumulated legacy non-conformance from being copied verbatim into the net-new archive (the data-quality-debt problem in §1.2).

4.3 Tag remediation by pre-authored rules

XyDromatics Migration remediates DICOM tags through a rules engine: operator-authored, priority-ordered rules with nine condition operators (is-blank, is-not-blank, equals, not-equals, contains, not-contains, starts-with, ends-with, wildcard-match) and eight actions — **set** (which can add an absent tag), **copy**, **prepend**, **append**, **regex** replace, **clear** (null a value, keeping the element), **lookup** (table-driven), and **delete** (remove the element). Rules are **pre-authored and persisted** — a preset specification applied deterministically on ingest, not hand-editing at apply time. With no rules configured, remediation is a no-op.

The engine can write any tag, including clinical-meaning tags; consistent with Synthology's advisory posture, such an edit is **not blocked** — instead the transit-audit record is flagged "advisory" when a clinical-meaning tag changed, purely for the audit trail. This is a **non-blocking audit flag**, not an approval or hold step.

4.4 Configurable quality gates and a rejection queue

XyDromatics Migration's ingest pipeline includes DICOM validation, missing-data derivation, and configurable quality gates (required / not-empty / min-length / regex / value-in / conditional-not-blank), backed by a rejection queue. When enabled, an instance that fails a reject-policy rule or gate is **not written** to the migration archive — it is diverted to the rejection queue (bytes and metadata retained) for operator triage and one-click re-feed. These are **metadata/conformance** checks applied **per instance on ingest**, and they are **rule-driven**: with nothing configured, nothing is gated. They are configurable gates that, when enabled, gate each ingested instance — not an automatic per-study pass/fail.

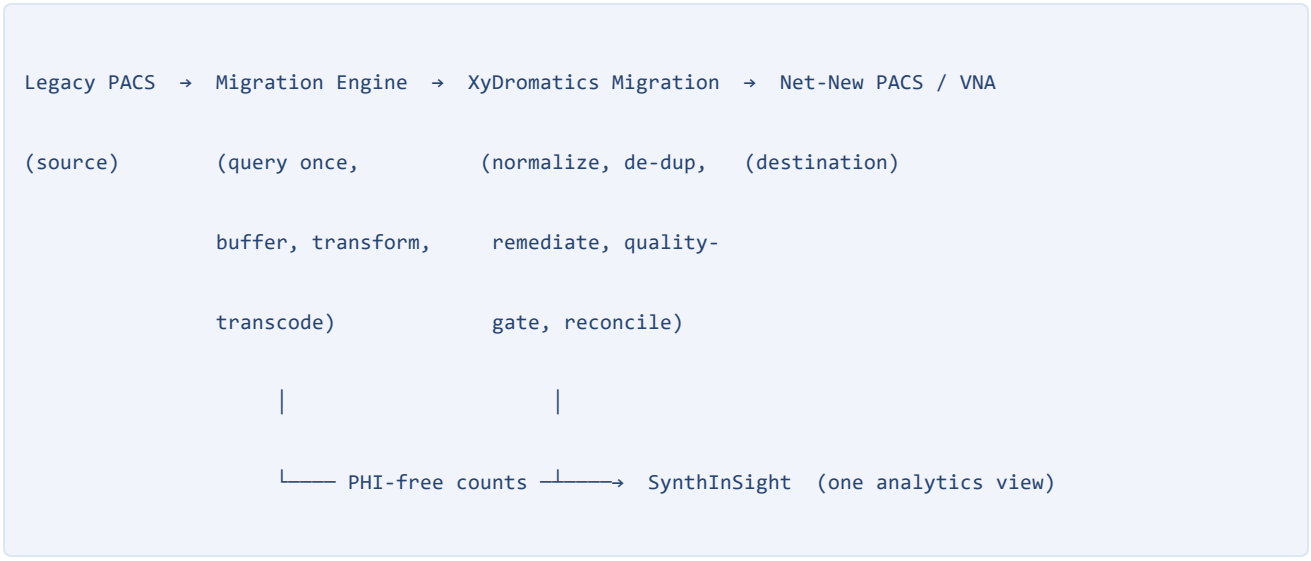
4.5 Automatic post-migration reconciliation

When every job in a batch reaches a terminal state, a background service **reconciles automatically**: for each completed study it queries the **source** PACS (C-FIND) and compares the source's series/instance counts against the migration archive, classifying each study as matched, partial (missing series/instances), missing, source-extras, or source-unreachable, and persisting per-study results plus a per-batch summary. Operators can re-run a batch, and an **opt-in** setting can auto-requeue partial/missing studies. The comparison is **count-based** against XyDromatics Migration's own archive; the source is the side queried over the wire. Reconciliation fidelity depends on the source populating the count return keys.

---

§5 — The complete Migration Suite

The three products compose into one workflow, and each is licensed independently — the Suite is a **commercial bundle, not a fused binary**:



- **Migration Engine** takes the query load off the live source and conditions studies in transit.
- **XyDromatics Migration** cleans and gates each study and reconciles the result.
- **SynthInSight** provides one operational view across both (§6).

Two reference architectures:

1. **Migration Engine + a third-party migration server.** The customer keeps their existing migration tool and points it at Migration Engine as its source, to offload the live PACS and (optionally) condition studies in transit. This is the incremental, single-product entry point.
2. **The full Suite.** Migration Engine fronts the source; XyDromatics Migration is the migration server; SynthInSight is the analytics view. One vendor, one workflow.

---

§6 — One PHI-free view of the whole migration

Both Migration Engine and XyDromatics Migration write **one aggregate row per completed migration job** to a shared feed that **SynthInSight** reads as a single, cross-product "Migration Suite" analytics view. Because product is a breakdown dimension, the two roll up together and split apart on demand, alongside instance, source, modality, status, and channel.

The feed is PHI-free by construction. Each row records only these fields:

`product`, `instance`, `timestamp_utc`, `source_id`, `source_name`, `modality`, `status`, `channel`, `images`, `bytes`, `duration_ms`, `attempt_count`, `first_pass`, `skipped`.

There is **no field capable of holding** a patient name, patient ID, Study Instance UID, accession number, or free-text note. The per-patient / which-study detail stays in each product's own on-premise store (XyDromatics Migration's dashboard and study list; Migration Engine's per-study trace), behind that product's PHI-reveal control. Only the de-identified aggregate rows reach SynthInSight. A short settle window (default 12 hours) means a job that is requeued shortly after finishing does not publish a stale outcome.

Two honest notes on reading these numbers: the metric is **per migration job/operation**, not per distinct study (a reconciliation second pass is a separate job), and `bytes` is populated by Migration Engine but reported as 0

by XyDromatics Migration (whose job model counts images, not bytes). SynthInSight itself is **general-purpose business software** — it handles counts, not images, and is not a medical device.

---

§7 — What we do not claim

In the interest of an accurate evaluation:

- **No performance figures.** This paper makes no throughput, studies-per-hour, "N× faster," or "reduces load by N%" claim. Those depend on the customer's environment; Synthology does not publish them for the Suite.
- **Migration Engine does not skip *pulling* studies already at the target.** It reports overlap and avoids re-*sending* instances the destination already has (§3.3); it does not suppress retrieval from the source.
- **De-duplication is study-level, by UID and counts** (§4.1) — not byte-level or hash-verified.
- **Reconciliation is count-based** against XyDromatics Migration's own archive (§4.4) — not a per-image hash comparison, and the destination side is XyDromatics Migration's archive, not an independent third-party destination queried over the wire.
- **Transcoding is lossless only** (§3.2) — never lossy, never "compression" in a sense that implies data loss.
- **The federated Q/R proxy and several conditioning features are licensed and off by default** (§2) — real, shipped capabilities, but explicitly enabled and configured, not automatic.

---

§8 — Regulatory & privacy posture

- **XyDromatics Migration Engine** and **XyDromatics Migration** are **non-device software under FD&C Act §520(o)(1)(D)** — they transfer, store, convert, and display medical device data without clinical interpretation. Neither is a medical device, and neither has any clinical function or clinical lane. XyDromatics Migration surfaces this posture as machine-readable fields on its status API; Migration Engine carries it in its product metadata and DICOM conformance documentation.
- **SynthInSight** is **general-purpose business software** — fleet operational analytics. It is not a medical device, has no clinical function, and handles no PHI (aggregate counts only).
- **PHI stays on the on-premise image path.** Only de-identified aggregate counts reach SynthInSight; the shared feed has no field capable of carrying a patient, study, or accession identifier.

---

§9 — Where to go next

- Product pages: `/products/migration-engine`, `/products/vna-migration`, `/products/synthinsight`, and the Suite overview at `/products/migration-suite`.
- The one-page Migration Suite dataflow poster is on `/posters`.

- To scope a legacy-PACS retirement — with the full Suite or with Migration Engine in front of your existing migration server — contact Synthology for an architecture review.

> **Regulatory classification.** All Synthology and XyDromatics software described here is non-device software under the FD&C Act §520(o)(1)(D) software exclusion — not a medical device, and no clinical diagnostic function. Published by Synthology Healthcare Solutions Group, LLC as a controlled document; the authoritative version is the approved record in the Synthology quality-management system.