CLOSE THE VISIBILITY GAP

Data Integration Is the Hidden Risk in Healthcare Rollups

Executive Summary

The success of health care rollups depends on rapid operational unification across every new acquisition. However, most platforms cannot see across their own portfolios. Each acquired practice brings its own EHR, billing system, and data conventions, and the integration required to produce consolidated reporting routinely takes twelve to thirty-six months. The underlying problem is structural: both traditional engineering and new AI-generated scripting produce maintenance costs that grow quadratically with portfolio size. Every new acquisition makes every existing integration harder to maintain. At scale, the integration team spends its entire capacity on upkeep, and the rollup stalls.

The cost of this visibility gap is concrete. Based on industry benchmarks for denial-related revenue loss, duplicative staffing, and deferred operational improvements, we estimate that mid-to-large platforms lose $200,000 to $500,000 per month while integration remains incomplete. More than 40% of healthcare organizations report losing 10% or more of annual revenue to leakage (Sage Growth Partners and Fibroblast, 2024), up to 65% of denied claims are never resubmitted (Healthcare Financial Management Association, 2024), and firms without modern data architecture capabilities forgo a 15 to 20 percent advantage in portfolio company value creation (Ganti, Aditya, 2025).

We introduce a technology that compresses typical integration timelines from months to weeks, with a team of 3-5 instead of 15-25 to twenty-five, and keeps maintenance roughly constant regardless the number of active sources. In benchmark testing across 60 structural format variants spanning the data types found in healthcare rollups, the system achieves 99.0% mapping accuracy at $3.30 per field mapping. By accelerating integration by four to five months per acquisition, a ten-clinic portfolio can unlock over $2 million in year-one EBITDA improvement, translating to approximately $24 million in enterprise value at a 12x exit multiple. This means faster post-acquisition visibility, earlier intervention on denial and leakage patterns, and a scalable data foundation that supports both operational improvement and portfolio value creation.

Intended Audience (Technical Edition)

Target Audience: CDOs, CTOs, and Data Architects
Technical Focus: Architectural scalability by solving the quadratic maintenance problem where upkeep grows with portfolio size.
Key Framework: Concept-based mapping from $n \times m$ pairwise connections to an $n+m$ semantic layer.
Benchmarks: 99.0% mapping accuracy across 60 structural variants (HL7, FHIR, EAV, and flat files).
Maintenance Target: Flat-load upkeep with review burden under 1 hour/week for 200+ integrations.

The Visibility Gap

A PE-backed dermatology platform closes on its fifth acquisition in eighteen months. Each practice arrived with its own EHR, its own billing system, and its own way of classifying everything from denial reasons to payer identifiers. The integration consultants are four months into a data-mapping project at the third site and have not yet started on the fourth. The CFO cannot produce a consolidated denial report for the board. Meanwhile, the sixth deal is already in letter of intent.

This is the default experience of healthcare consolidation in the United States today. And the gap it exposes between the pace of acquisition and the pace of data integration is the single greatest underappreciated risk in PE-backed healthcare rollups.

The Consolidation Wave

The healthcare rollup thesis is now the dominant PE playbook in the sector. Add-on acquisitions now account for approximately seventy-three percent of all PE buyout deals (Choi, Jinny and Walters, Kyle and Tang, Kenny and Corridore, Jim and Wright, Brian, 2025; CohnReznick, 2024), each bringing its own EHR, billing system, and years of accumulated local data conventions onto the platform. The target specialties, dermatology, ophthalmology, gastroenterology, dental, orthopedics, and behavioral health, share the profile that makes rollups attractive: fragmented markets, favorable payer mixes, and high-margin procedural revenue. They also share the profile that makes data integration hardest, because no two practices encode their operations the same way.

The economic logic depends entirely on integration speed. Top-performing PE firms target operational unification within six months of close, because every month of delayed integration is a month of unrealized synergies: duplicate billing teams, fragmented reporting, and missed opportunities to optimize across the portfolio. The operating partner who assembled a twenty-clinic platform is not managing twenty clinics. They are managing the gap between twenty clinics and one unified operation.

What the Data Landscape Actually Looks Like

The challenge is not abstract. It is concrete and immediate.

When a new clinic is brought onto a platform, the data problems begin at the field level. Something as fundamental as date of birth may be stored as `Patient_DOB` in one system and `BirthDate` in another, in different formats, with different data types. Multiply this by provider identifiers, insurance plan codes, procedure codes, diagnostic codes and dozens of other fields, the scope of the mapping problem becomes clear. Each discrepancy is a point of friction that can break downstream processes, from claims submission to financial reporting.

The problem intensifies with clinical and billing codes. One acquired practice may still use legacy ICD-9 codes for certain diagnoses. Another has developed proprietary internal codes for procedures that have no standard equivalent. Payer identifiers differ across every system: United Healthcare might be `UHC001` in one billing platform and `UnitedHealth-99` in another. Until someone maps every payer code to a single master list, consolidated financial reporting by payer is impossible.

Denial management illustrates the problem at its sharpest. Consider five clinics, each handling denials differently:

- **Clinic A** attaches standardized ANSI CARC codes (e.g., CARC 45, CARC 27) to each denial.
- **Clinic B** stores free-text reasons typed by billers ("Insurance expired," "Need medical notes").
- **Clinic C** uses internal numeric codes on a proprietary scale of 1 to 10.
- **Clinic D** records denial information in a general notes field mixed with other data.
- **Clinic E** tracks only whether a claim was paid or not, with no denial reason captured at all.

This is not a technology problem. It is a **visibility** problem. The data needed to make decisions exists, but it cannot be seen in a unified way.

The human cost is significant. Without integrated systems, staff become the integration layer. Firms rely on the "swivel-chair" pattern where billing specialists manually transfer information between screens, run separate reports from multiple systems, and reconcile data in spreadsheets. Finance and operations teams at portfolio companies spend roughly half their time gathering and reconciling data rather than analyzing it (Ganti, Aditya, 2025). This time produces no analytical value and is available for neither improvement nor patient care.

The financial cost is equally concrete, and it scales superlinearly with portfolio size. A small rollup integrating three to five practices typically spends $100,000 to $300,000 over six to nine months. A mid-size platform with ten to twenty practices faces $500,000 to $1.5 million and twelve to eighteen months. Large platforms with thirty or more sites routinely exceed $2 million over two to three years. Adding each new source system multiplies complexity rather than merely adding to it. Integrating fifty systems can cost roughly ten times what five systems cost, not ten times more per system.

Every month of incomplete integration extends the visibility gap: the period when consolidated analytics and revenue cycle improvements simply do not exist. Manual data processes alone extend deal timelines by an estimated 30% (Ganti, Aditya, 2025). For a mid-to-large platform, we estimate the combined monthly cost of this blackout at $200,000 to $500,000, spanning three categories: revenue leakage from unresolved denials, duplicative labor across parallel workflows, and the carrying cost of legacy systems that cannot yet be decommissioned. The components are individually well-documented. Peer-reviewed research cites Optum data showing the average facility loses 2 to 3 percent of yearly revenue to preventable leakage (Algarni, Ayed and others, 2023), and a survey of healthcare executives found that over 40% of organizations lose 10% or more of annual revenue to leakage (Sage Growth Partners and Fibroblast, 2024). For a fifteen-site platform generating $75 million in aggregate revenue, even conservative assumptions—2.5% revenue cycle leakage ($156,000/month), $50,000 to $150,000/month in duplicative back-office FTEs, and $50,000 to $100,000/month in deferred billing and denial management improvements—produce a combined gap of $256,000 to $406,000 per month, consistent with the estimated range. Given that up to 65% of denied claims are never resubmitted (Healthcare Financial Management Association, 2024), the cost of fragmented denial visibility alone can reach into the millions annually.

Why Current Approaches Fail at Scale

Healthcare data is uniquely resistant to standardized integration. Every EHR implementation is customized. Coding schemes vary not just between vendors but between installations of the same vendor. Local conventions, such as abbreviations, workarounds, custom fields, accumulate over years of use. Two hospitals running identical Epic instances can have radically different data architectures underneath.

This heterogeneity means that any integration approach must contend with structural diversity at a scale that most enterprise tools were not designed for.

Traditional human engineering has been the default approach for decades. Skilled engineers analyze source schemas, write mapping logic, build transformation pipelines, and test them against production data. The per-integration cost runs $2,800 to $5,800, and a team of fifteen to twenty-five can handle a batch of fifty sources in eight to ten weeks.

The problem compounds with maintenance. Each source drifts independently as EHR vendors release updates, coding standards change, and local conventions evolve. At fifty active integrations, a platform can expect two to five variation events per week, each requiring an engineer to investigate, diagnose, fix, test, and deploy. By two hundred integrations, the maintenance load alone demands five to ten dedicated engineers, and the cost compounds quadratically. This is not because each fix is harder, but because the number of things that can drift grows with the square of the number of sources.

AI-generated integration scripts (using large language models to write transformation code) appears on the surface to solve the cost problem. Initial per-integration cost drops to $135 to $540. But each integration produces a bespoke script with no shared architecture. At five hundred sources, the “codebase” is five hundred unrelated scripts. Any payer code update, new compliance requirement or other cross-cutting change requires touching all of them. The maintenance cost at scale actually exceeds traditional engineering, because the code is unfamiliar, inconsistent, and difficult to modify reliably. Plausible errors, such as the wrong field mapped, or type coercion, incorrectly pass along as silent errors and may not surface until downstream analysis fails.

The core issue is not the cost of any single integration. It is the **shape of the maintenance curve**. Both traditional and AI-scripted approaches produce maintenance costs that grow quadratically with the number of active integrations. At small scale, this is manageable. At the scale of a growing rollup, where each quarter brings new acquisitions, and the portfolio may reach fifty to two hundred active data sources, the integration team becomes entirely consumed by maintaining what already exists. New acquisitions stall. The rollup thesis breaks.

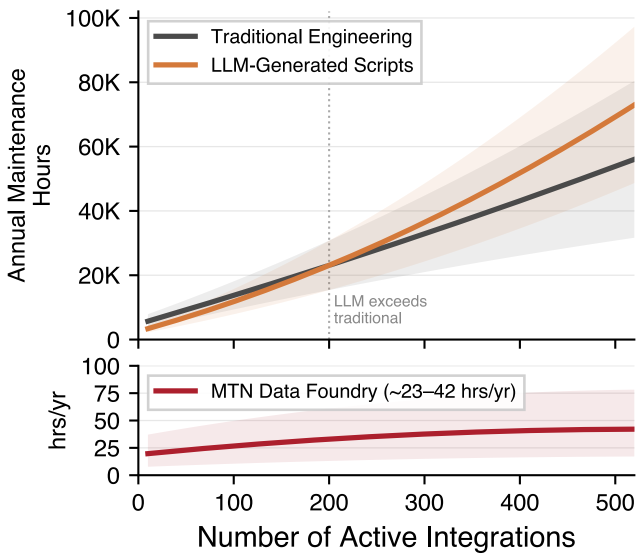

Figure 1. Annual maintenance hours by number of active integrations. Traditional and LLM hours are derived from labor costs at a $65/hr fully-loaded rate; Data Foundry hours are derived from documented variation-event rates (events/week $\times$ review minutes). LLM-generated scripts start cheaper but exceed traditional engineering beyond $\sim$200 integrations because each bespoke script must be touched individually for any cross-cutting change. Shaded regions indicate ranges.

The track record of these approaches reinforces the structural concern. Large-scale IT projects run an average of 45% over budget while delivering 56% less value than predicted (McKinsey & Company, 2012). In healthcare, the pattern is at least as severe: industry estimates consistently place the failure or significant shortfall rate for enterprise healthcare IT projects at seventy to eighty percent. When an integration project does fail at a mid-size hospital, the sunk cost typically runs into the millions, with substantial additional recovery costs required just to start over. Meanwhile, manual data processes extend deal timelines by an estimated 30%, and firms that invest in modern data architecture outperform peers by 15 to 20 percent in portfolio company value creation (Ganti, Aditya, 2025).

Maintenance compounds the problem further. Annual upkeep typically runs fifteen to twenty percent of the initial integration investment, and more than half of total ownership cost accrues after go-live. Schema drift, vendor updates, and evolving local conventions create a perpetual re-mapping burden that most initial project budgets fail to account for. Hidden costs, from data quality remediation to staff turnover mid-project to compliance rework, routinely push actual spend well beyond initial projections.

> **The question for any platform operator is straightforward: at your current rate of acquisition, when does the maintenance curve cross the line where integration overhead exceeds the operational capacity of your team?** For most growing rollups, the answer is sooner than expected.

An Architectural Approach

What a Solution Must Do

The requirements for a solution are structural, not incremental.

First, the solution must adapt to each source system's unique data model without requiring months of manual field-by-field mapping. Healthcare data is too heterogeneous and too frequently customized for any approach that depends on predefined schemas or static mapping templates.

Second, it must learn the relationships between disparate data structures rather than relying on rules written by engineers who may or may not understand the clinical context. The mapping logic needs to be transparent, auditable, and correctable by domain experts who are not software engineers.

Third, it must deliver usable integrated data in weeks, not quarters. The economics of a rollup do not accommodate eighteen-month integration timelines. Every month of delay carries a concrete cost in unrealized synergies and extended visibility gap.

Fourth, the maintenance cost must remain roughly constant as the number of active integrations grows. This is the critical requirement. Any solution whose maintenance burden scales with the square of the portfolio size will eventually break for the same reasons the current approaches break. The architecture must be designed so that adding the fiftieth source costs no more to maintain than the fifth.

Finally, the solution must make its accuracy trade-offs explicit. In healthcare, silent data errors carry clinical and financial consequences. A system that passes unmapped or incorrectly mapped fields through without flagging them is worse than one that rejects them outright. The standard should be: data is either correct and validated, or explicitly pending human review.

These requirements sound like a wish list, but they are architecturally achievable once you reframe the problem. The combinatorial explosion in current approaches comes from mapping every source to every other source. If instead every source maps to a single shared concept layer, the number of required mappings grows linearly, not quadratically, and maintaining source fifty does not require touching sources one through forty-nine.

Introducing Data Foundry

The MTN Data Foundry is built on a concept-based architecture designed to meet these requirements.

The core principle is straightforward. Rather than building pairwise mappings between every source system and every target system (which produces the combinatorial explosion described in the previous section), Data Foundry maps every source to a shared concept layer. Each concept represents a standardized data element: "Patient.DateOfBirth," "Claim.DenialReasonCode," "Provider.NPI," and so on. When a new source system arrives, its fields are mapped to this existing concept vocabulary, not to every other system in the portfolio. The result is that adding a new integration does not require touching any existing integration. The concept layer acts as a universal translator.

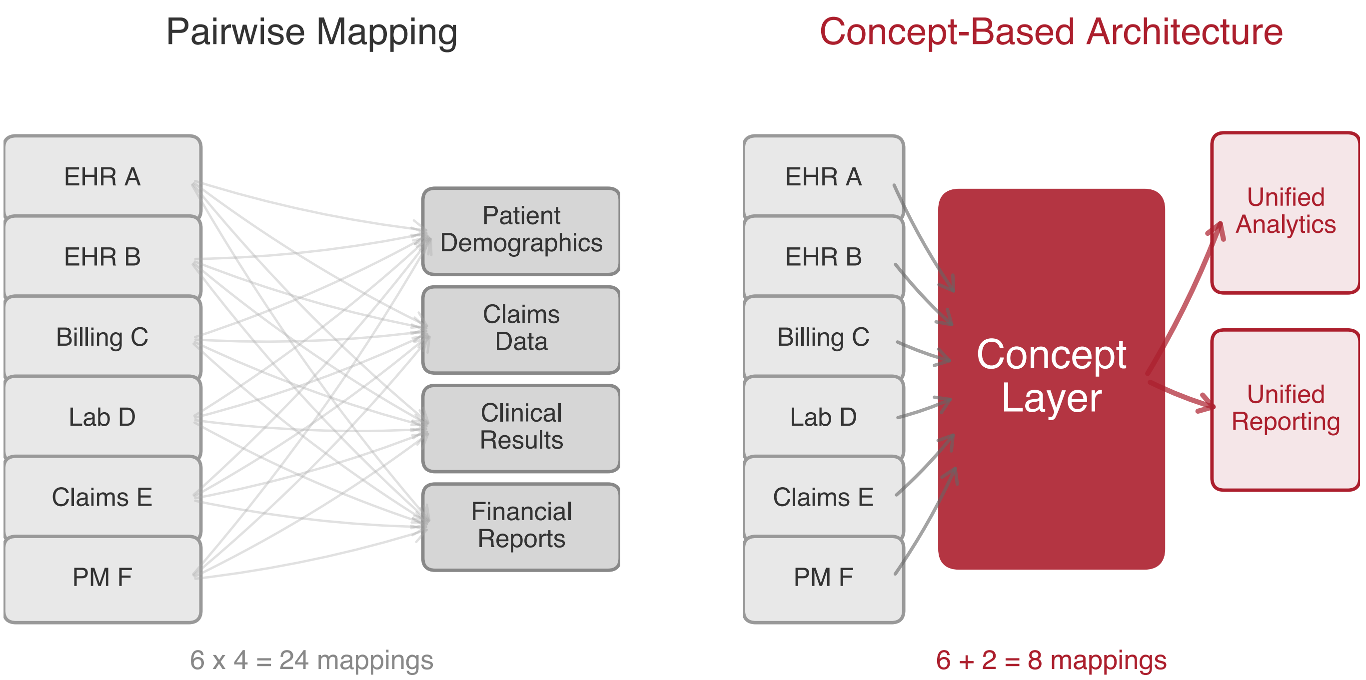

Figure 2. Pairwise mapping (left) produces $n \times m$ connections that grow combinatorially. Concept-based architecture (right) reduces this to $n + m$ mappings through a shared semantic layer.

The platform operates through five integrated capabilities:

Schema Visualization. When a new data source is loaded, Data Foundry presents its structure visually, allowing analysts to see field names, data types, and sample values alongside the existing concept vocabulary. This replaces the traditional process of digging through vendor documentation or compiling field inventories in spreadsheets. For denial management across five EHRs, the analyst can immediately see where each system stores denial codes, how they are formatted, and where the gaps are.

Self-Organizing Concept Maps. Using the schema metadata and sample data, the system automatically proposes mappings between source fields and target concepts. It identifies obvious matches (such as "DOB" to "Patient.DateOfBirth"), detects format differences (such as "M/F" versus "Male/Female"), and flags ambiguous cases for human review. In benchmark testing, the AI correctly handles the straightforward majority of mappings, leaving a focused review surface for the analyst.

Approval Dashboard. Every AI-proposed mapping is presented for human review before activation. The dashboard shows confidence scores, sample data values, and suggested transformations. An analyst (not a senior engineer) can review and approve a typical integration in five to fifteen minutes. Nothing goes live without explicit human sign-off. This is not a fully automated black box. It is a human-in-the-loop system that uses AI to compress the work, not to eliminate oversight.

Rehydration. Once mappings are approved, the system materializes the unified dataset. Source data is transformed according to the approved mappings and loaded into a common output format. This is the step where five clinics' denial data, each stored in its own format, becomes a single consolidated denial table with standardized fields and codes. The output can feed any downstream analytics tool, BI dashboard, or operational system.

Self-Healing Maintenance. When a source system changes (a vendor update, a new field, an unfamiliar code), the platform detects the deviation automatically and queues it for review. An analyst receives a notification, reviews the proposed re-mapping, and approves the update. At two hundred active sources generating two to five variation events per week, the total review burden is less than one hour per week. This is what keeps the maintenance curve flat.

Benchmarking Methodology

Data Foundry has been evaluated across 11 structurally diverse data sources and 60 format variants, spanning flat files, nested hierarchies, entity-attribute-value structures, columnar batches, pipe-delimited feeds, and wide sparse-column layouts.

Benchmarking a healthcare data integration platform against actual EHR and billing data is constrained by PHI regulations and the proprietary nature of each installation's configuration. Instead, the evaluation uses structurally equivalent data sources that reproduce the same format diversity, schema complexity, and variation patterns found in healthcare rollups. The mapping table below shows why this is methodologically rigorous rather than a limitation: every benchmarked structural variant maps directly to a real healthcare data type.

How Benchmarked Data Maps to Healthcare

Benchmarked Source Type	Healthcare Rollup Equivalent
Wearable devices	Remote patient monitoring feeds (CGMs, pulse oximeters, connected BP cuffs)
Industrial monitors	Clinical device streams (bedside vitals, infusion pumps, spirometry, imaging equipment)
IoT gateways	Clinic network hubs and middleware aggregating from multiple devices or rooms
Standardized interchange formats	HL7 v2, FHIR R4 bundles, C-CDA documents, X12 EDI (837/835/270/271)
Multi-domain platforms	EHR/PM suites (Epic, Athenahealth, NextGen, eClinicalWorks, ModMed, Nextech)

Benchmarked Structural Variant	Healthcare Data That Looks Like This
Flat (CSV/delimited)	Payer fee schedules, eligibility rosters, credentialing spreadsheets, A/R aging exports
Nested (hierarchical/JSON)	FHIR resource bundles, C-CDA summaries, multi-level charge/payment hierarchies
Entity-attribute-value	Lab results keyed by LOINC, vitals stored as observation rows, custom flowsheet data
Columnar batch	Nightly 835 remittance files, batch 837 claims, clearinghouse rejection reports
Pipe-delimited	HL7 v2 message segments (PID, OBX, DG1), legacy lab interfaces, pharmacy feeds
Wide/sparse-column	Multi-specialty encounter forms, survey instruments, research datasets

Every acquisition adds three to five of these structural variants simultaneously. A single clinic joining the platform may contribute an EHR export, HL7 feeds, batch billing files, and device data, each with its own format. Sixty source-variant combinations is a realistic count for a ten-to-twenty clinic rollup, not an inflated estimate.

Performance Results

Metric	Result
Mapping Accuracy	99.0%
Format Consistency	100.0%
Record Capture Rate	96.2%
Average Mapping Time	442 seconds
Average Cost per Mapping	$3.30
Total Field Mappings Evaluated	480

Mapping accuracy measures the percentage of data fields correctly mapped to the target concept. Format consistency measures whether the same field in the same source is mapped identically across all structural variants. Record capture rate measures the percentage of expected normalized records successfully produced after transformation.

The Economics

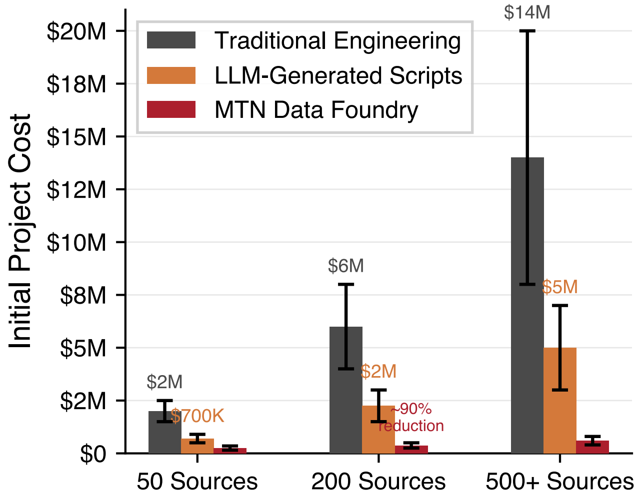

Figure 3. Initial project cost by portfolio scale and integration approach. Data Foundry costs remain sub-linear due to concept reuse across sources.

Where the Cost Savings Come From

A typical $2 million enterprise healthcare data integration project breaks down roughly as follows:

- Project management and coordination: 10-15% ($200-300K)
- Data discovery and source analysis: 10-15% ($200-300K)
- Architecture and infrastructure: 15-20% ($300-400K)
- **Mapping and transformation development: 25-35% ($500-700K)**
- Testing and QA: 15-20% ($300-400K)
- Deployment, monitoring, and training: 10-15% ($200-300K)

Data Foundry's primary displacement target is the mapping and transformation layer, where it achieves a 90 to 95 percent cost reduction. But the effects cascade: testing effort drops 60 to 75 percent (because structural coverage checking replaces most manual test cases), data discovery drops 30 to 50 percent (because automated fingerprinting handles structural profiling), and project management overhead drops 20 to 30 percent simply because the project is shorter.

Industry cost benchmarks validate the baseline. Mid-size rollups with ten to twenty practices typically spend $500,000 to $1.5 million over twelve to eighteen months on integration through traditional means. Large platforms with thirty or more sites routinely exceed $2 million over two to three years. These figures are consistent across multiple industry sources.

EBITDA Impact

The financial benefit of faster integration is direct. Consider a newly acquired practice generating $10 million in annual revenue at a 15% EBITDA margin. The PE thesis projects that centralized billing, improved denial management, and operational economies will raise that margin to 20%, producing an additional $500,000 in annual EBITDA. If traditional integration takes six months, those improvements are delayed by half a year. By the Data Foundry compressing integration just one to two months, roughly four to five additional months of enhanced margin are realized in year one: approximately $208,000 in additional EBITDA per clinic.

Across a portfolio of ten acquisitions, the acceleration effect alone can exceed $2 million in year-one value. At a representative exit multiple of 12x EBITDA, that $2 million in additional annual earnings translates to $24 million in enterprise value, a return that dwarfs the cost of the integration platform itself.

Industry data reinforces the stakes. Each month of visibility gap costs a mid-to-large platform an estimated $200,000 to $500,000 in combined revenue leakage, duplicative labor, and legacy system carrying costs (see derivation in *The Visibility Gap*). Manual data processes alone extend deal timelines by an estimated 30% (Ganti, Aditya, 2025), and large-scale IT projects run an average of 45% over budget while delivering 56% less value than predicted (McKinsey & Company, 2012). The risk of doing nothing, or doing it the conventional way, is not hypothetical.

Conclusion

The Visibility Imperative

The pattern is clear. Private equity investment in healthcare continues to accelerate, with approximately seventy-three percent of PE buyout deals structured as add-on acquisitions (Choi, Jinny and Walters, Kyle and Tang, Kenny and Corridore, Jim and Wright, Brian, 2025; CohnReznick, 2024). Each acquisition adds new systems, new data structures, and new integration burden. The question facing every platform operator is not whether to integrate, but how to integrate at a pace that matches the pace of acquisition.

The core risk is not technical complexity. It is visibility. When a platform cannot produce consolidated reporting across its portfolio, when denial patterns remain hidden in five different coding systems, when the CFO's board presentation depends on spreadsheets assembled by hand, the rollup thesis is operating on faith rather than data. Every month of that visibility gap carries a concrete cost: an estimated $200,000 to $500,000 per month for a mid-to-large platform in combined revenue leakage, duplicative labor, and deferred synergies (see *The Visibility Gap* for derivation and supporting sources).

Current integration approaches share a common structural limitation: their maintenance costs grow quadratically with portfolio size. At small scale, this is manageable. At the scale most PE-backed platforms are targeting, it becomes the binding constraint on growth.

A concept-based architecture offers a fundamentally different cost curve. By mapping every source to a shared concept layer rather than building pairwise integrations, the marginal cost of adding new sources remains roughly constant. Self-healing maintenance keeps the ongoing burden flat. The result is not a marginal improvement in integration economics. It is a structural change in what is operationally feasible.

Next Steps

We offer an integration complexity assessment: a two-week analysis that maps your current data landscape, identifies the specific visibility gaps across your portfolio, and estimates the cost of delayed integration at your current acquisition pace.

Whether you are evaluating a platform acquisition, midway through a multi-site integration, or planning for the next phase of growth, the core question is the same: how many active data sources will you have in eighteen months, and does your current integration approach scale to that number?

MTN is a research and technology company with deep roots in clinical operations, computational neuroscience, and machine intelligence. Our work has been published in published in Nature journals, PNAS, JMIR, Chest, PlosCompBio, The Royal Society and other leading venues. We bring these conversations the perspective of researchers and advisors with clinical and technical backgrounds.

Technical Leadership

Warren Pettine, MD — Co-Founder and CEO. Assistant Professor at the University of Utah where he leads the Medical Machine Intelligence (M^{2}Int) Lab. Trained in machine learning research at Harvard, Stanford, NYU Yale. Prior health policy experience in the U.S. Congress and service on the University of Utah Institutional Review Board ground MTN's approach in policy and regulatory expertise.

Matthias Christenson, PhD — AI Architect. Investigator with the M^{2}Int Lab. PhD and postdoctoral research at Columbia University in computational ML, with prior industry experience as a Deep Learning Research Engineer at DeepLife training foundational models on genomic and biometric data. Leads MTN's technical architecture design and data model development.

Brian Locke, MD, MSCI — Clinical AI Lead. Investigator with the M^{2}Int Lab. Active ICU physician and Assistant Professor at Intermountain Healthcare, bringing firsthand understanding of clinical workflows across academic medical centers and integrated delivery networks. Provides the methodological rigor for the clinical and operational implications of MTN's technology.

Samuel Wecker, Lead Systems Engineer. Over twelve years building and scaling production software, including as a founding engineer at a startup that grew to a billion-dollar platform. Specializes in unifying disparate systems and data sources at scale. Leads Data Foundry's core platform development.

Contact: Warren Pettine, MD, CEO of MTNwarren@themtn.aihttps://www.themtn.ai