COMPLIANCE TO CAPABILITY

CMS Interoperability Mandates Require More Than APIs

Executive Summary

The CMS Interoperability and Prior Authorization Final Rule (CMS-0057-F) represents the most ambitious interoperability mandate in U.S. healthcare history. It requires Medicare Advantage organizations, Medicaid managed care plans, and qualified health plan issuers to build FHIR APIs for provider access, payer-to-payer exchange, and electronic prior authorization. This has an estimated industry cost of $1.55 billion over ten years and 6.9 million burden hours in year one alone (Centers for Medicare & Medicaid Services, 2024). Compliance deadlines begin January 2026 for operational provisions and January 2027 for API requirements.

But compliance is not capability. The mandates solve data transport. They do not solve data semantics. Data arriving through Provider Access APIs, payer-to-payer exchange, and PA submissions will flow in nominally standard formats, but will not retain consistent meaning across systems. Only 59% of SNOMED CT concepts map directly to ICD-11 equivalents (Fung, Kin Wah and Xu, Julia and Bodenreider, Olivier, 2024). Patient match rates between organizations can be as low as 50% (The Pew Charitable Trusts, 2018). The result: plans will build the compliance infrastructure but still be unable to compute quality measures for Star Ratings, capture risk adjustment revenue accurately, or auto-adjudicate prior authorizations at the CAQH benchmark of $0.05 per electronic transaction versus $3.52 per manual transaction (CAQH, 2024).

The structural problem is familiar. Both manual mapping and integration-engine approaches produce maintenance costs that grow quadratically with the number of provider connections. A concept-based architecture that maps every source to a shared semantic layer keeps maintenance roughly constant as the provider network grows. In benchmark testing across 60 structural format variants spanning the data types payers ingest, this approach achieves 99.0% mapping accuracy at $3.30 per field mapping, compressing typical integration timelines from months to minutes per source.

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 Compliance-to-Capability Gap

A large Medicare Advantage plan has spent eighteen months and significant budget building its Provider Access API to comply with CMS-0057-F. The API is live. Data is flowing from hundreds of network providers. But the quality team still cannot compute Star measures without manual chart chase. The risk adjustment team is finding that diagnoses arriving through the new API do not survive normalization. Conditions coded in provider-local conventions do not map cleanly to HCC categories. The prior authorization team built an electronic intake channel, but submissions arrive in formats so inconsistent that auto-adjudication rates remain in single digits. The CIO is reporting to the board that the compliance project is on track. The CDO knows the data is connected but not computable.

The gap this scenario exposes between the presence of data and the usability of data is the single greatest underappreciated risk facing payers under these mandates.

The Mandate Wave

CMS is using interoperability rules to industrialize payer workflows around member access, provider access, payer-to-payer exchange, and electronic prior authorization, largely via HL7 FHIR APIs with required operational timeframes and reporting (Centers for Medicare & Medicaid Services, 2024).

Two rules define the current mandate. The 2020 Interoperability and Patient Access Final Rule (CMS-9115-F) established the baseline, requiring impacted payers to implement FHIR Patient Access APIs and related data access policies (Centers for Medicare & Medicaid Services, 2020). The 2024 Interoperability and Prior Authorization Final Rule (CMS-0057-F) extends this significantly, adding a Provider Access API, Payer-to-Payer API, Prior Authorization API, PA status in the Patient Access API, and operational PA requirements with mandatory reporting (Centers for Medicare & Medicaid Services, 2024).

The compliance deadlines create a compressed execution window. Operational provisions, including PA process requirements and reporting, begin January 1, 2026. Most API build requirements are due around January 1, 2027 (Centers for Medicare & Medicaid Services, 2024). The impacted payers include Medicare Advantage organizations, Medicaid and CHIP fee-for-service agencies, Medicaid and CHIP managed care plans, and certain qualified health plan issuers on the federal exchange.

The industry-wide cost is substantial. CMS's regulatory impact analysis estimates approximately $1.55 billion in total compliance cost over ten years across impacted payers, with first-year burden of about 6.9 million hours (Centers for Medicare & Medicaid Services, 2024). Medicare Advantage remaining costs alone total roughly $311.8 million under the 2020 rule's primary estimate (Centers for Medicare & Medicaid Services, 2020). TEFCA, the parallel "network-of-networks" framework, creates additional connectivity pathways under common governance (Assistant Secretary for Technology Policy, 2024). Globally, 78% of countries now have data exchange regulations, and 73% of those include FHIR in their mandate (Weistra, Ward, 2025).

CMS's quantified "savings" in the 2024 rule are primarily provider paperwork savings, approximately $15.8 billion over ten years for physician practices, not payer savings. CMS explicitly treats these as illustrative and excludes them from the monetized accounting tables (Centers for Medicare & Medicaid Services, 2024). The payer side of the ledger is dominated by build and compliance costs. **Payers recapture value only if data quality, identity matching, and workflow penetration are solved.**

Connected but Not Computable

The problem is not building APIs. The real problem is semantic interoperability: data that flows in standard formats but does not retain consistent meaning or utility across systems.

Syntactic interoperability, getting data to move in standard message formats, is largely achievable with current technology. However, semantic interoperability, ensuring that shared data means the same thing on both sides, remains elusive. Two systems can be 100% HL7 or FHIR compliant and still fail to interoperate due to optionality, local extensions, and implementation variability.

The evidence is specific:

- A JAMIA pilot study mapping between SNOMED CT and ICD-11 found only **59% direct equivalents**. The remainder required post-coordination or had no match, with challenges including non-synonymous synonyms, granularity mismatches, and composite concepts (Fung, Kin Wah and Xu, Julia and Bodenreider, Olivier, 2024).

- Patient match rates between organizations can be as low as **50%**, even when both run the same EHR system (The Pew Charitable Trusts, 2018). The average duplicate record rate within organizations is **18%** (The Pew Charitable Trusts, 2018).
- In one national survey, **38% of U.S. providers reported an adverse event within two years** due to patient record mismatches (eHealth Initiative and NextGate, 2020).

For a Medicare Advantage plan, these numbers have direct revenue consequences.

Risk adjustment. A provider sends a FHIR Condition resource with a SNOMED code for diabetes. The plan's risk adjustment system needs an ICD-10 HCC category. If the mapping does not exist or is imprecise, the condition silently drops from the RAF calculation. Multiply this by thousands of conditions across hundreds of providers, and poor schema destroys risk-adjusted revenue without anyone knowing what was lost. Small improvements in RAF accuracy translate to millions in per-member-per-month revenue. The distinction matters: this is not about aggressive coding. It is about preventing information loss at the schema level.

Quality measures. Star Ratings determine quality bonus payments and rebate percentages. Moving from 3.5 to 4 stars can increase a plan's rebate percentage by approximately five percentage points, worth hundreds of millions for large plans. But most Star measure failures are not clinical. They are data availability failures: value set misalignment, timestamp ambiguity, provenance confusion, and encounter normalization errors. A lab result arriving with an unmapped LOINC code or ambiguous encounter context cannot be used for measure computation. The quality team falls back to manual chart chase.

Prior authorization. CAQH reports plan cost of approximately $3.52 per manual prior authorization transaction versus $0.05 per electronic transaction (CAQH, 2024). CMS-0057-F mandates the API plumbing that can convert portal and fax workflows into structured transactions. But auto-adjudication requires consistent schema, predictable documentation fields, and normalized clinical elements. Without semantic reconciliation, the "electronic" PA is still a document dump, and the transaction enters the manual queue.

The pattern is consistent: organizations overinvest in interface engines and API infrastructure while underinvesting in terminology services and mapping maintenance, leading to connected systems that do not speak the same language.

Why Current Approaches Break at Scale

Integration engines, Rhapsody, MuleSoft, Mirth, and their equivalents, solve transport and connectivity. They get data from point A to point B in standard formats. They are necessary infrastructure. But they were designed for a problem that is now largely solved: moving messages between systems. They were not designed for the harder problem that CMS mandates now expose: ensuring that the data retains consistent meaning after it arrives.

Traditional human engineering has been the default approach for the semantic layer. Skilled engineers analyze source schemas, write mapping logic, build transformation pipelines, and test them against production data. Each provider interface carries substantial upfront cost, and large health systems routinely maintain hundreds of custom interfaces.

The problem compounds with provider network scale. Two installations of the same Epic instance can produce structurally different FHIR bundles because of local customization, extensions, and coding conventions. Code sets change annually. Vendor versions update. Benefit designs evolve. At fifty active integrations, a plan can expect multiple 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) appear on the surface to solve the cost problem. Initial per-integration cost drops significantly. But each integration produces a bespoke script with no shared architecture.

At five hundred provider connections, the "codebase" is five hundred unrelated scripts. Any code set update, new CMS compliance requirement, or cross-cutting change to terminology mappings 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. In a payer context, the consequences of silent errors are severe: a plausible but incorrect SNOMED-to-HCC mapping passes validation but erodes RAF revenue; a subtly wrong LOINC mapping makes a lab result invisible to quality measure computation. These errors may not surface until downstream analysis fails or an audit reveals the gap.

The core issue is not the cost of any single provider 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 large MA plan onboarding hundreds of network providers under CMS mandate deadlines, the integration team becomes entirely consumed by maintaining what already exists. New provider connections stall. The compliance timeline breaks.

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). Annual maintenance typically consumes a significant fraction of the initial integration investment, and more than half of total ownership cost accrues after go-live.

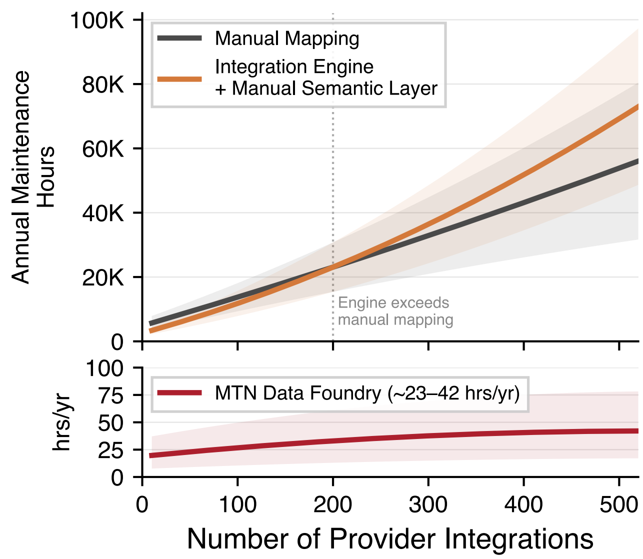

Figure 1. Annual maintenance hours by number of active provider integrations. Shaded regions indicate ranges.

The question for any plan operator is straightforward: as your provider network grows with membership, when does the maintenance curve cross the line where integration overhead exceeds the capacity of your data team? For most large MA plans building out CMS-mandated APIs, the answer is sooner than expected.

An Architectural Approach

What a Solution Must Do

1. The solution must adapt to each provider's unique data model without requiring months of manual field-by-field mapping. Every provider's FHIR implementation is customized. Two installations of the same EHR can produce structurally different bundles. Any approach that depends on predefined schemas or static mapping templates will break under the diversity of a large provider network.
2. 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.

3. It must deliver computable data in weeks, not quarters. The 2026 to 2027 compliance window does not accommodate eighteen-month integration timelines. Every month of delay is a month of data that cannot be used for Star Ratings, risk adjustment, or prior authorization automation.
4. The maintenance cost must remain roughly constant as the number of provider connections grows. This is the critical requirement. Any solution whose maintenance burden scales with the square of network size will eventually break for the same reasons the current approaches break. The architecture must be designed so that adding provider connection two hundred costs no more to maintain than connection five.
5. The solution must make its accuracy trade-offs explicit. In payer operations, silent data errors carry direct revenue consequences. An unmapped diagnosis code is lost RAF revenue. An uncomputable quality measure is a missed Star opportunity. 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.

Introducing the MTN 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 provider's data format and every downstream payer system (risk adjustment, quality measurement, PA rules engines, analytics), Data Foundry maps every source to a shared concept layer. Each concept represents a standardized data element: "Condition.ICD10Code," "Observation.LOINCCode," "Encounter.Type," "Medication.RxNormCode," and so on. When a new provider's data arrives, its fields are mapped to this existing concept vocabulary, not to every downstream consumer. The result is that adding a new provider integration does not require touching any existing integration. The concept layer acts as a universal translator.

The platform operates through five integrated capabilities:

Schema Visualization. When data arrives from a new network provider's FHIR API, 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. An analyst can immediately see where this provider stores diagnosis codes, how lab results are formatted, and where the gaps are.

Self-Organizing Concept Maps. Using the schema metadata and sample data, the system automatically proposes mappings between the provider's fields and target concepts. It identifies obvious matches (SNOMED conditions to ICD-10 HCC categories), detects format differences (date formats, coding system variants), and flags ambiguous cases for human review. In benchmark testing, the system 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 provider 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 clinical data from hundreds of providers, each structured differently, becomes a single computable dataset for risk adjustment calculation, quality measure computation, and PA auto-adjudication. The output can feed any downstream analytics tool, rules engine, or operational system.

Self-Healing Maintenance. When a provider's EHR updates, a code set changes, or an unfamiliar value appears, 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 or more active

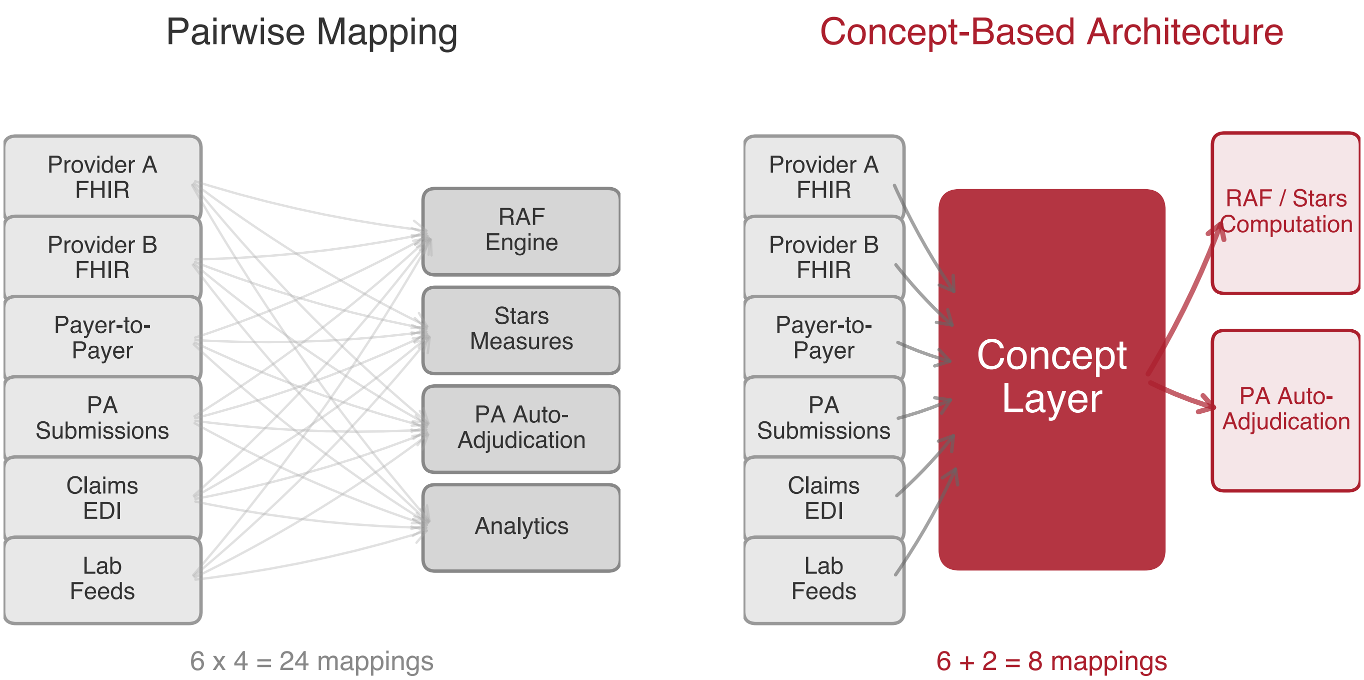

Figure 2. Pairwise mapping (left) produces $n \times m$ connections that grow combinatorially. Concept-based architecture (right) reduces this to $n + m$ mappings through a shared semantic layer.

provider connections generating variation events weekly, the total review burden remains less than one hour per week. This is what keeps the maintenance curve flat.

How Benchmarked Data Maps to Payer Operations

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 payer data integration platform against actual provider FHIR feeds and claims data is constrained by PHI regulations and the proprietary nature of each provider's EHR configuration. Instead, the evaluation uses structurally equivalent data sources that reproduce the same format diversity, schema complexity, and variation patterns found in payer integration environments. The mapping tables show why this is methodologically rigorous rather than a limitation: every benchmarked structural variant maps directly to a real payer data type.

Every provider connection adds three to five of these structural variants simultaneously. A single provider sending data through the Provider Access API may contribute FHIR bundles, HL7 v2 legacy feeds, batch claims files, and lab results, each in its own format. A large MA plan with five hundred or more network providers faces well over a thousand source-variant combinations. Sixty format variants in the benchmark represents a conservative slice of real-world payer complexity.

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

Every benchmarked source type maps directly to a real payer data integration scenario.

Benchmarked Source Type	Payer Integration Equivalent
Wearable devices	RPM feeds: CGMs, BP cuffs, pulse oximeters via Provider Access API or direct integrations
Industrial monitors	Clinical device streams: bedside vitals, lab instruments, imaging equipment (proprietary formats per vendor)
IoT gateways	Health system middleware aggregating clinical data before FHIR API transmission
Standardized interchange formats	FHIR R4 bundles (Provider Access, P2P, PA APIs), X12 EDI (837/835/270/271), C-CDA, HL7 v2
Multi-domain platforms	EHR/PM suites (Epic, Cerner, Athenahealth, eCW, NextGen) – each with proprietary data models

Benchmarked structural variants map to the format diversity found in payer data environments.

Benchmarked Structural Variant	Payer Data That Looks Like This
Flat (CSV/delimited)	Eligibility rosters, capitation files, provider directories, HEDIS flat files, pharmacy crosswalks
Nested (hierarchical/JSON)	FHIR bundles (Patient→Conditions→Meds→Obs), P2P exchange, PA structures
Entity-attribute-value	Quality measures (one row per measure per member), lab results by LOINC, HCC capture tables
Columnar batch	835 remittance, batch 837 claims, encounter submissions, PA decision batches, capitation reconciliation
Pipe-delimited	HL7 v2 feeds (ADT, labs, observations), pharmacy dispense feeds, immunization registry submissions
Wide/sparse-column	Multi-specialty encounter forms, CAHPS surveys, SDOH screenings, risk assessment questionnaires

of expected normalized records successfully produced after transformation.

For a payer audience, these results answer a specific question: what percentage of inbound provider data becomes computable for risk adjustment, quality measures, and PA automation, and how fast?

The Economics

Where the Revenue Impact Sits

The economic case for concept-based payer data integration rests on four value pools, each tied directly to plan economics.

Star Ratings. Star Ratings determine quality bonus payments and rebate percentages. Moving from 3.5 to 4 stars increases rebate percentage by approximately five percentage points. For large MA plans, this translates to hundreds of millions in annual bonus revenue. Most Star measure gaps are data availability failures, not clinical failures. Concept-based mapping makes clinical data computable for quality measures by resolving the value set misalignments, timestamp ambiguities, and encounter normalization errors that force manual chart chase.

Risk adjustment. Accurate risk adjustment means premiums align with actual member acuity. Small improvements in RAF capture accuracy translate to millions in PMPM revenue across MA populations. Data Foundry prevents the information loss at the schema level that silently erodes RAF revenue: diagnoses that fail to ingest, map incorrectly, do not survive deduplication, or lose attributability. This is not about aggressive coding. It is about ensuring clinical history is computable on day one after a member enrolls or a provider connects.

Prior authorization. The CAQH benchmark is clear: $3.52 per manual PA transaction versus $0.05 per electronic transaction (CAQH, 2024). But electronic does not mean automated. Auto-adjudication requires consistent schema, predictable documentation fields, and normalized clinical elements. Without semantic reconciliation at the intake layer, the “electronic” PA remains a document dump that enters the manual queue. Data Foundry enables the structured intake that makes auto-adjudication possible at scale.

Membership retention. Better payer-to-payer data exchange reduces the blind spot period for new members, where pricing assumptions are uncertain, risk scores lag, and care gaps are invisible. A one percentage point reduction in churn for a mid-size MA plan can preserve tens to hundreds of millions in annual premium revenue. CMS frames payer-to-payer exchange as preventing loss of critical information when patients change plans (Centers for Medicare & Medicaid Services, 2025). Faster time to a computable longitudinal record stabilizes premium economics.

The Maintenance Economics

The structural argument is the same as in PE-backed healthcare rollups: maintenance cost determines long-term feasibility.

At fifty provider integrations under the traditional approach, the plan needs a dedicated engineering team just for maintenance. Each interface carries substantial build and ongoing cost, and the majority of total ownership cost accrues after go-live as source schemas drift and code sets change.

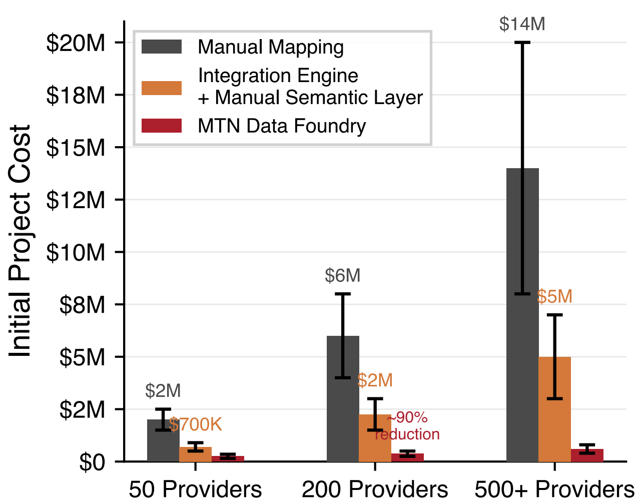

Figure 3. Initial project cost by provider network scale and integration approach. Data Foundry costs remain sub-linear due to concept reuse across sources.

With Data Foundry's concept-based architecture, the same fifty integrations require one part-time reviewer. At two hundred provider connections, where the traditional approach demands five to ten dedicated engineers and the maintenance burden has grown quadratically, Data Foundry's review burden remains under one hour per week. The per-field mapping cost of $3.30 compares to interface-level costs that run into the tens of thousands.

The $1.55 billion industry compliance cost is already being spent. The question is whether it produces compliance only, or compliance plus capability. Plans that treat interoperability as an IT compliance project will see costs. Plans that treat it as revenue infrastructure will see Star bonuses, RAF integrity, and PA efficiency.

Conclusion

The Capability Imperative

CMS-0057-F will connect payer systems to provider data at unprecedented scale. The compliance infrastructure is being built across the industry at a projected cost of $1.55 billion over ten years. The question that remains is whether the data flowing through those APIs will be usable for the decisions that determine plan economics: risk adjustment, quality measurement, and prior authorization automation.

The structural challenge is clear. Semantic interoperability does not come free with syntactic compliance. APIs solve transport. Without automated semantic reconciliation, payers will have connected systems that do not speak the same language. The maintenance burden of manual mapping grows quadratically with provider network size, creating the same scaling constraint documented in PE-backed healthcare rollups. At the scale of a large MA plan's provider network, the integration team eventually spends its entire capacity on maintaining what already exists, with no bandwidth left for new capability.

A concept-based architecture offers a fundamentally different cost curve. By mapping every provider's data to a shared concept layer rather than building pairwise integrations, the marginal cost of adding new provider connections 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 at network scale.

We propose a single metric to evaluate the effectiveness of any payer data integration approach: the **Com-**

putable Clinical Data Rate, defined as the percent of inbound clinical and claims data immediately usable for risk, quality, or authorization decisions without manual intervention. This metric applies across RAF, Stars, and PA. It is executive-friendly, architecture-agnostic, and hard to game. It captures the gap between having data and being able to use it.

Next Steps

We offer a Computable Clinical Data Rate Assessment: a focused analysis that measures the current CCDR across your provider network, identifies the specific semantic gaps reducing Stars, RAF, and PA performance, and quantifies the revenue impact of closing those gaps.

Whether you are mid-compliance build, post-go-live and finding the data is not usable, or planning for the January 2027 API deadlines, the core question is the same: what percentage of the data flowing through your new APIs is actually computable?

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