

# Turning Contract Library into Executable Contracts



**Ashish Jaiman**  
Founder & CEO

## Executive Summary

Most payers have already invested in *contract centralization*, a “library” that improves **access** to provider agreements, amendments, exhibits, and fee schedules. That investment typically succeeds at **storage and transparency** (find it, read it, benchmark it) but fails at **execution and integrity** (price the claim correctly, consistently, and explainably).

Payer’s environment, deep amendment history, entity complexity, carve-outs, “lesser-of” clauses, and negotiated exceptions create a predictable failure mode. **Contracts are visible but not operational**. Claims then drift to base terms, stale fee schedules, or generic pricing logic.

The result is **leakage**, avoidable provider abrasion, weak audit defensibility, and renewal negotiations that lack clause-level proof.

**Nēdl Labs is an activation layer for contract execution.** We convert “contracts” into **executable, versioned, clause-traceable logic**. This governed “contract-as-code” system can drive pre-pay repricing, post-pay defensible audit, contract-abuse detection, and negotiation intelligence.

<https://www.linkedin.com/pulse/contracts-executable-code-ashish-jaiman-18qne/>

## The Core Problem: “Contracts Library” ≠ “Contracts Execute”

Traditional Contract Lifecycle Management (CLM) and compliance tools are designed primarily for lawyers and strategists, not for payment engines. While they excel at document organization, workflow compliance, and market-rate benchmarking, they struggle with **deterministic execution**

## Contract Drift is the Default Outcome

Even with a clean repository, payment outcomes drift because the adjudication stack typically:

- Prices against a base agreement while missing subsequent amendments.
- Applies generic global edits that ignore contract-specific intent.
- Uses stale fee schedules or misapplied effective dates.
- Cannot reconcile exceptions and “stacking logic” across multiple layers.

This drift isn’t “human error” alone; it’s structural. Contracts are written for humans, yet pricing must be deterministic, real-time, and auditable.

## Amendment Sprawl Breaks “Read-Only” Systems

Over the years, a single provider relationship accumulates massive complexity:

- **The Stack:** Base agreement + multiple amendments + exhibits + fee schedules.
- **The Conditions:** Carve-outs by place of service, specialty, code ranges, modifiers, and network tiers.
- **The Math:** “Lesser-of” logic, caps/floors, stop-loss/outlier methodologies, and bundling rules.
- **The Exceptions:** Negotiated terms that override global edits.

A repository can store and display this complexity. Still, it cannot reliably answer the only question that matters at claim time: *“What is the correct payment method for this claim, today, for this entity/network, under the current term hierarchy?”*

## The Audit/Appeal Problem is Really a "Provenance" Problem

When providers dispute reimbursement or the payer challenges billing patterns, the hard part is proving the logic chain. Without clause-level provenance, both sides end up in expensive, abrasive loops trying to prove:

- Which contract version was active?
- Which clause controlled?
- Which inputs were used?
- How was the math applied?
- Why did an exception trigger (or fail to trigger)?

These platforms potentially can answer “what does the contract say,” but they cannot answer “how do I mathematically execute this conflict-resolved logic on Claim #123?” Ultimately, they serve as excellent digital filing cabinets, whereas Payer requires a digital calculator to solve the execution gap.

# The Nēdl Labs Core: Neuro-Symbolic Architecture

## The "AI Dilemma" in Payment Integrity

To solve the "Contract Execution" problem, you need two contradictory capabilities: the flexibility to read human language and the rigidity to execute audits. Existing market solutions fail because they pick only one side:

- **Pure Generative AI (The "Hallucination" Risk):** AI is a probabilistic predictor. They are excellent at summarizing a 50-page PDF, but dangerous for math. They cannot guarantee that  $\$100 + 10\%$  will equal  $\$110$  every time, nor can they reliably execute multi-step logic without "drifting."
- **Pure Symbolic/Rule Engines (The "Brittleness" Risk):** Traditional rule engines (e.g., Drools) are deterministic but blind. They cannot read scanned PDFs, understand nuances of the "lesser-of" rule, or interpret a messy layout where a footer modifies a table. They break the moment the input format changes.

## The Solution: System 1 (Perception) + System 2 (Logic)

Nēdl Labs has developed a Neuro-Symbolic architecture that decouples "Understanding" from "Execution." We use Compound AI to read the contract and Symbolic Logic to enforce it.

### The "Neuro" Layer (Perception & Normalization)

- **Role:** The "Reader & Extractor" that handles unstructured chaos.
- **Tech Stack:** Compound AI (Clinical Models, Vision Encoders, Meta extractors, ...)
- **Function:**
  - **Semantic Segmentation:** Ingests messy PDFs, identifying boundaries between Base Agreements, Amendments, and Exhibits.
  - **Entity Extraction:** Normalizes varied terms ("Facility," "ASC," "Site of Care") into a canonical ontology.
  - **Auto-formalization:** This is our key innovation. The AI does not calculate the rate; it translates the natural-language clause into an Intermediate Representation, a structured pseudo-code representation of the clause's intent that does not execute the calculation.

### The "Symbolic" Layer (Reasoning & Execution)

- **Role:** The "Calculator" that guarantees 100% auditability.
- **Tech Stack:** Knowledge Graphs + Constraint Solvers (Python/Prolog).
- **Function:**
  - **Temporal Resolution:** We model contracts not as documents, but as a **Time-Valid Graph**. Nodes (Terms) and Edges (Relationships) have specific validity

windows. The system traverses the graph to find the exact active path for a specific Date of Service and resolves stacking amendments deterministically.

- **Constraint Satisfaction:** We execute the Intermediate Representation as rigid logic.
  - *Input:*  $\min(\text{Billed Charges}, (\text{Medicare Rate} * 1.10))$
  - *Execution:* The solver fetches the variables and computes the result. It cannot hallucinate because it is bound by mathematical constraints.
- **Traceability:** Every output is linked to a "Proof Tree", a digital thread connecting the final dollar amount back to the specific logic step and the original PDF source text.

## The Result: A "Transparent Glass Box" System

By combining these approaches, we achieve what neither can do alone:

- **Flexibility of Ingestion:** We accept dirty data (scans, emails, PDFs, filesystems,..).
- **Certainty of Execution:** We deliver audit-grade, reproducible financial outcomes.
- **Transparency of Logic:** We provide a "Transparent Glass Box" view where every decision is fully explainable, unlike the "Black Box" of pure Neural Networks.

## The Executable Contract System

Think of the Nēdl Labs solution as three tightly coupled layers that transform static documents into active financial controls.

### Contract Intelligence Layer (Understanding + Structuring)

*The "Perception Engine" that converts unstructured chaos into structured assets.*

- **Clause Inventory:** Every clause becomes an addressable object with a unique ID.
- **Normalized Terms:** Standardizing rates, formulas, code sets, conditions, and exceptions into a common data model.
- **Entity Model:** resolving the complex "Who" and "Where" (Provider, Network Tier, Plan, Geography, Site-of-Service, and Effective Dates).
- **Amendment/Addenda Graph:** A temporal map of precedence rules, explicitly tracking what overrides what to resolve conflicts between documents.

### Contract-as-Code Layer (Compilation + Governance)

*The "Compiler" that turns structured data into governed, executable logic.*

- **Formal Rule Representation:** Converts terms into a Domain Specific Language (DSL) or structured rules engine.

- **Versioned "Contract Packages":** Creates a specific, versioned logic package for every Provider/Entity/Network combination.
- **Automated Test Suite:**
  - *Unit Tests:* Validates individual clause logic.
  - *Regression Tests:* Re-runs historical claims to ensure accuracy and catch unintended variance.
- **Change Control:** Full governance including diffs across versions, approval workflows, and immutable audit logs.

## Execution & Evidence Layer (Runtime + Explainability)

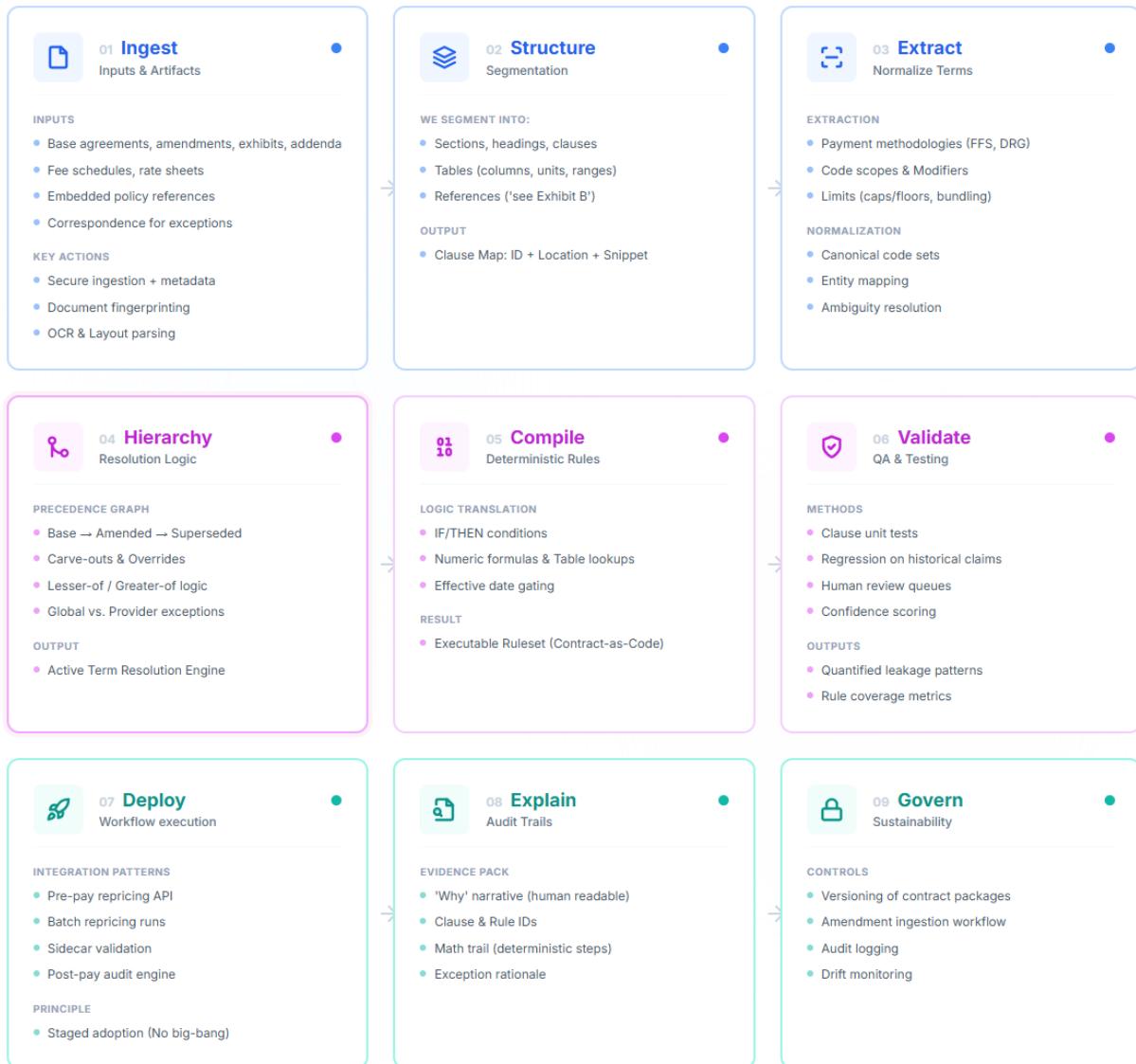
*The "Engine" that powers adjudication and defends the result.*

- **Pricing/Repricing Runtime:** A high-performance API (or batch process) for real-time adjudication.
- **Per-Claim Trace:** Generates a complete lineage for every decision: *Inputs → Rules Fired → Clause Citations → Math → Payment.*
- **Evidence Packs:** Automated documentation generation for audits, appeals, and negotiations.
- **Active Monitoring:** Continuous detection of drift, anomaly signals, and rule coverage gaps.

## Contract-to-Execution Pipeline

End-to-End Architecture: From Ingestion to Governance

● INGEST   ● COMPUTE   ● EXECUTE



## Practical Outcomes: What Payer Gets

We deliver three specific business results that shift the focus from "managing documents" to "managing spend."

### Elimination of "Contract Drift" (Pre-Pay)

**The Outcome:** Claims are paid against the *exact* active amendment hierarchy for that specific date of service, not a stale base agreement.

**The Value:** We stop the 3-5% of frequent leakage caused by missed fee schedules or addenda during adjudication.

## "Hard Denial" FWA Detection

**The Outcome:** We move from generic "anomaly detection" (which is easily appealed) to Intent-Based Denials. We flag billing patterns that technically pass code edits but mathematically violate specific contract clauses (e.g., bundling rules or exclusions).

**The Value:** Converts "soft" recoveries into defensible, hard savings that hold up to provider challenge.

## Negotiation Leverage (\$)

**The Outcome:** Negotiators enter renewals with a "Clause-Level P&L." We quantify precisely how much specific terms (like Stop-Loss thresholds or Carve-Outs) cost the plan last year.

**The Value:** Enables data-driven counter-offers, potentially saving millions in future contract cycles.

## Success Metrics (KPIs)

We measure success by Financial Integrity, not "AI Accuracy."

Metric Category	Key Performance Indicator (KPI)	Target / Goal
<b>Financial Impact</b>	<b>Identified Variance (\$)</b>	Total dollar value of leakage found vs. baseline historical pricing.
	<b>Drift Reduction (%)</b>	% decrease in claims paid on stale/expired contract terms.
<b>Operational Speed</b>	<b>Amendment Cycle Time</b>	Reduction in time to operationalize a new amendment (from <b>months to days</b> ).
<b>Defensibility</b>	<b>Trace Completeness (%)</b>	% of high-value claims with a complete <i>Payment Math Clause</i> audit trail.
<b>Coverage</b>	<b>Spend Governed (%)</b>	% of total provider spend is now governed by "Executable" rules rather than manual review.

## Use Cases

Use Case Category	The Current "Filing Cabinet" Gap (Status Quo)	The Nedl "Executable" Fix (Proposed Solution)	Business Impact (ROI)
<b>Payment Integrity (Pre-Pay)</b>	<p><b>"Contract Drift":</b>            Claims are adjudicated against outdated Base Agreements because recent addenda are trapped in PDF/image formats.</p>	<p><b>Temporal Knowledge Graph:</b> The engine automatically selects the currently active rate path (traversing all amendments) based on the specified Date of Service.</p>	Eliminates <b>3-5%</b> of pre-pay leakage caused by using expired rates.
<b>Complex Logic Execution</b>	<p><b>Static Lookup:</b>            Cannot handle "Lesser-Of" or "Stop-Loss" clauses that require real-time math (e.g., <i>"Pay 110% of Medicare OR Billed Charges"</i>).</p>	<p><b>Symbolic Solver:</b>            Dynamically calculates multiple pricing scenarios in real-time and enforces the "Lesser-Of" logic deterministically.</p>	Prevents overpayments on high-dollar claims by enforcing complex caps.
<b>FWA &amp; Anomaly Detection</b>	<p><b>Code-Based Edits:</b>            Only catches standard coding errors (e.g., duplicates). Misses "Contract Abuse" where providers technically pass edits but violate intent.</p>	<p><b>Intent-Based Detection:</b> Extracts the <i>intent</i> of a clause (e.g., <i>"Add-on code valid only with complex surgery"</i>) to flag abusive billing patterns.</p>	Identifies <b>~10-15%</b> more recoverable than standard rules engines.
<b>Contract Negotiations</b>	<p><b>Aggregate Guesswork:</b>            Negotiators rely on high-level averages, unaware of specific</p>	<p><b>"What-If" Simulations:</b> Re-run historical claims against hypothetical terms (e.g., <i>"What if we raise the Stop-Loss</i></p>	Empowers data-driven renewals, potentially saving <b>millions</b> in future spend.

	clauses driving unprofitability.	<i>threshold?"</i> ) to predict exact variance.	
<b>Provider Onboarding</b>	<b>Manual Entry:</b> Takes 30-90 days to interpret and manually load complex contracts into the adjudication system.	<b>Auto-Ingestion:</b> "Neuro" layer extracts fee schedules and rules from PDFs instantly, auto-populating the logic engine.	Reduces operational backlog and provider abrasion; accelerates "Time-to-Value."
<b>Contextual Benchmarking</b>	<b>Disconnected Data:</b> Negotiators lack a precise "apples-to-apples" view of market rates because external transparency files (MRFs) are too messy to map complex internal terms.	<b>Market Normalization:</b> Ingests public transparency data and maps it through the same logic engine to align external market rates against Payer's internal "Executable Rates."	Identifies immediate <b>rate compression opportunities</b> (e.g., <i>"We pay 9% above market for this code"</i> ) for actionable leverage.

## Key Differentiation:

Feature / Capability	Nedl Labs	Doczy (Arete)	Turquoise Health	Icertis (ICI)
<b>Primary Value Prop</b>	<b>Payment Execution</b> (Paying claims accurately)	<b>Productivity</b> (Helping humans read faster)	<b>Transparency</b> (Benchmarking market rates)	<b>Compliance</b> (Legal workflow & storage)
<b>Core Technology</b>	<b>Neuro-Symbolic AI</b> (LLM + Logic Graph)	<b>Generative AI</b> (RAG + Chatbot)	<b>Data Aggregation</b> (MRF Parsing)	<b>Traditional CLM</b> (Workflow + Rules)
<b>Handling "Math"</b>	<b>Deterministic</b> (Calculates "Lesser-Of" precisely)	<b>Probabilistic</b> (Guesses/Summarizes text)	<b>Benchmark-based</b> (Compares vs. Market)	<b>Metadata-based</b> (Tags fields, doesn't calc)
<b>Handling "Time"</b>	<b>Temporal Graph</b> (Resolves addenda conflicts)	<b>Static Search</b> (Finds the doc, misses the timeline)	<b>Snapshot</b> (Current Rate focus)	<b>Version Control</b> (File-level, not Logic-level)

Output Format	JSON / Python Code (Machine Executable)	Text Summary (Human Readable)	Rate Table / Dashboard (Strategy Readable)	Signed PDF (Legally Binding)
<b>The "Kill" Argument</b>	<i>"We stop pre-pay leakage."</i>	<i>"They summarize the leakage."</i>	<i>"They show you the market price, not your price."</i>	<i>"They store the contract; they don't execute it."</i>

## Conclusion: Turning a Sunk Cost into a Strategic Asset

Payers do not need another software tool to *store* contracts; it requires an engine to *execute* them. The previous investment in a "Contract Library" likely failed not because the digitization technology was ineffective, but because it was never designed for adjudication. It built a library when you needed a calculator.

Nēdl Labs offers a clear path to fix this. We do not ask you to rip and replace your existing infrastructure. Instead, we propose to sit on top of it, ingesting the static artifacts you already have and converting them into the dynamic, executable logic your claims system is missing.

By validating this with a targeted pilot on your most complex, high-variance contracts, we will prove that the solution to "Contract Drift" isn't more manual review, it's better math.

We are ready to start immediately.

## About the Author

### Nēdl Labs

Nēdl Labs is pioneering AI-native payment integrity solutions for healthcare payers. Our neuro-symbolic AI platform combines neural networks' pattern recognition with symbolic reasoning's explainability, enabling payment integrity systems that simultaneously reduce leakage and build provider trust.

Nēdl brings deep expertise in responsible AI, healthcare policy, and enterprise product development to the payment integrity challenge.

## Contact:

### **Ashish Jaiman**

Founder & CEO, Nēdl Labs,  
*Neuro-Symbolic AI for Payment and Revenue Integrity*

**Web:** [www.nedllabs.com](http://www.nedllabs.com)

**Email:** [ashish@nedllabs.com](mailto:ashish@nedllabs.com)

**LinkedIn:** [linkedin.com/in/ashishjaiman](https://linkedin.com/in/ashishjaiman)

**Locations:** Washington, D.C. | Silicon Valley | Dallas, TX

**© 2026 Nēdl Labs. All rights reserved.**

This white paper is provided for informational purposes only. No part of this publication may be reproduced without written permission.

The information contained herein is subject to change without notice. This document is not a commitment to provide any particular product, feature, or service.