Mphasis
The Next Applied

ACCELERATING ENTERPRISE AGENCY

Governed Front2Back™

FINANCIAL ANALYST MEET 2026

AI Applied at Scale

Ptn By: Managing Partner & CTO Mphasis.ai

May 27, 2026

Unpacking Mphasis Tria™

The Mphasis Tria™ Architecture: Insight, Foresight, Execute

The power of Tria™ is not any one layer in isolation.
The power is the closed loop: understand, decide, act, measure, and improve.

It Starts with How Enterprises Think

Layer 3 Execute - Orchestration + Guardrails

A controlled execution that converts decisions into tasks, approvals, system actions, and monitoring-with risk-based routing and bounded autonomy

Layer 2 Foresight - Decisioning Engine (Causal + Optimization)

A predictive decision engine that reasons about interventions (not correlations), quantifies tradeoffs, and produces an executable decision graph

Layer 1 Insight - Enterprise Memory (Ontology + Knowledge Graph)

A living model of the enterprise: entities, relationships, constraints, workflows, and telemetry-continuously updated from core systems and process artifacts

AI That Transforms

A system that takes objectives, translates them into intervention questions, estimates credible effects, optimizes under constraints, and outputs a structured plan that can be executed and monitored.

Unpacking Mphasis Tria™

Mphasis Tria™

Layer 3 – Execute
Orchestration + Guardrails

Layer 2 – Foresight
Decisioning Engine
Causal+ Optimization

Layer 1 – Insight
Enterprise Memory
Ontology + Knowledge Graph

Agentic Execution & Orchestration Models

- Desktop Agents
  Coding Agents
  (Modernization, Security, etc.)
  Agentic Skills
  (enhancing capabilities)
- Mphasis AI Superhighway™
- Mphasis NeoIP™
  (Engg. Agents - Orchestration, Agentic Lifecycle, NeoRigel™, NeoZeta™, NeoSaba™, NeoRaina™, NeoCrux™, AIOps
  Business Agents – SOPia, DAISY, RuleIQ)

Decisioning Intelligence Models

- Agentic AI Multistep Reasoning
- Decision AI
  Behavioral Economics, Predictive Optimization, Scenario Planning/Simulation and Real time recommendations

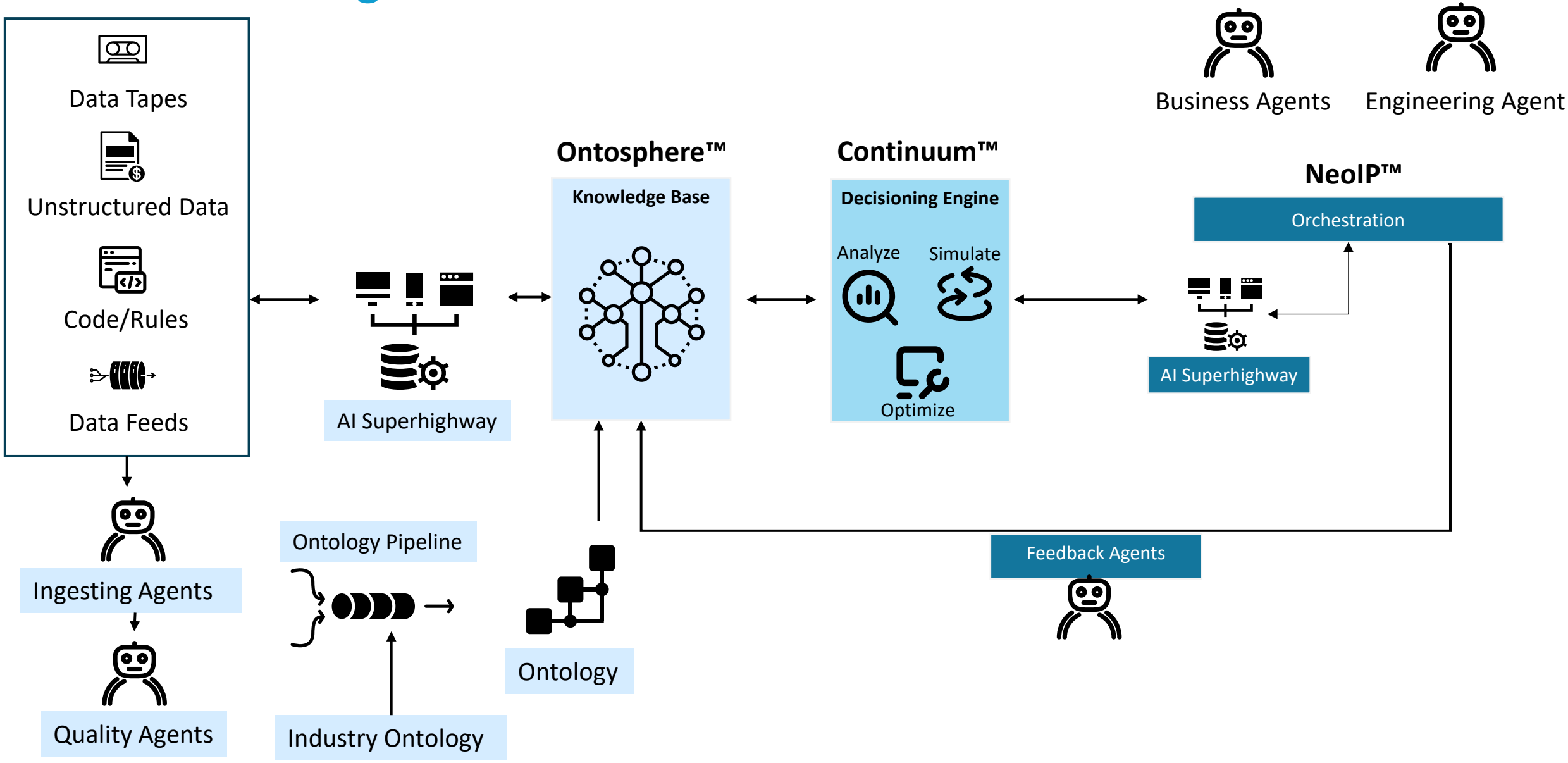
Building a Living Memory of the Enterprise

- Mphasis Ontosphere™
  (Ontology powered Dynamic Knowledge Graph – Living Enterprise Memory)
- Mphasis AI Superhighway™ , Integration Services (I2I), Data & Control Pane

Foundation Models
(Gemini/Sonnet/Opus & API)

Mphasis

From Knowledge to Autonomous Action



Mphasis Modernize™ Takes This to the Real World

Mphasis Modernize™ is a product line powered by Mphasis Tria™

- Goes beyond converting legacy systems into modern equivalents
- The end goal is not a modern copy of the old — it is enterprise Agency
- Extracts institutional knowledge embedded in legacy operations
- Builds governed decisioning frameworks for enterprise use
  Orchestrates agentic execution — enabling systems to sense, decide, and act autonomously
- Transforms enterprises from static, human-dependent engineering and business operations into autonomous, intelligent systems

One Agency Pattern, Three areas of Application

Built on Mphasis Tria™, Mphasis Modernize™ addresses three modernization surfaces to deliver enterprise Agency.

Technology Stack Modernization

Mainframe migration, legacy apps, data platforms — decades of business logic no one fully understands

The target: Agency-capable architecture, not a modern copy

Business Operations Modernization

Mortgage origination, trade processing, claims handling etc. - transforming decision-intensive operations with governed intelligence

Layer 1 builds domain memory. Layer 2 replaces human judgment with governed decisions. Layer 3 orchestrates outcomes

Process Modernization

Workflow automation, BPM transformation, and end-to-end process reengineering across the enterprise

Redesign workflows so processes self-optimize — shifting from manual handoffs to agentic orchestration

The pattern is always the same: extract knowledge, separate intent from implementation, build toward Agency.

Every Knowledge-Intensive Operation Is a Candidate

Verticals

Banking
Loan origination, credit decisioning, AML/fraud investigation

Insurance
Claims adjudication, underwriting triage, policy servicing

Capital Markets
Trade settlement, custody reconciliation, corporate actions

Healthcare
Prior authorization, medical coding, benefits verification

Domains *(independent of vertical)*

- Payment operations
- Customer onboarding
- Finance and accounting
- Procurement and vendor management
- Contract management
- Document processing
- HR and talent operations
- Customer service workflows
- Compliance and regulatory reporting
- Reconciliation
- IT service management
- Supply chain and logistics

Take Payment Operations — Here's the Problem

Case Study | Business Operations Modernization | Payment Operations

01

Fragmented across streams

Wire, ACH, SWIFT, CHIPS, RTP, cross-border — **7 payment streams, each with its own rules**

- Different regulatory frameworks per stream
- Distinct resolution logic and exception paths
- No single operator can master all of them

02

Exception-driven

64 distinct processes across 7 streams — **every exception requires human judgment**

- Operators must identify stream and process first
- Misidentification delays resolution
- 93 SOPs — no single operator knows them all

03

Knowledge walks out the door

Experienced operators hold **decades of resolution logic in their heads**

- When they leave, the knowledge leaves
- 6–12 months to train a new operator
- Regulation changes require retraining everyone

7 streams. 64 processes. 93 SOPs. Decades of tribal knowledge. This is what we walked into.

One Bank. 7 Streams. 64 Processes. 93 SOPs.

Case Study | Business Operations Modernization | Payment Operations

7

Payment Streams

- Wire, ACH, SWIFT, CHIPS, RTP, cross-border
- Each with different regulatory frameworks
- Distinct resolution logic per stream

64

Distinct Processes

- Multiple processes per stream
- Operators must identify stream and process first
- Misidentification delays resolution

93

Standard Operating Procedures

- No single operator knows all 93
- Regulation changes require retraining
- Updates must reach operators before next case

Measurable Impact — Faster Decisions, Faster Results
Wire and ACH streams. Others following.

Case Study | Business Operations Modernization | Payment Operations

Hours → Minutes

Targeted exception resolution time

Weeks → Days

New operator time to effectiveness

Consistent

Resolution quality — regardless of operator tenure or time of day

4+ Domains

Commercial Lending · KYC · Fraud · Trade all extend from the same platform

Built for payments. The same architecture now extends to every operations domain — at a fraction of the original build cost.

Card Core Modernization — Killing Tech Debt

FINTECH

Card Core Modernization

50M

Lines of Code

- Decades of COBOL carrying critical configuration logic
- Undocumented policy logic spanning thousands of client institutions
- Zero documentation across legacy payment workflows

$3T	Annual financial services transactions running on mainframe-dependent systems
220B	> 220B LOC legacy code exist in the financial services industry that has undocumented business logic encoded in it.
Now?	Speed is no longer aspirational — real-time payments, cloud mandates, and regulatory shifts demand transformation now

Enterprises typically use AI-assisted tools learn the legacy behavior and produce a modern version of the same thing.

Mphasis
The Next Applied

Case Study	Legacy Modernization	Legacy Modernization

01

AI code analysis

Tools learn what the code does. Output: a spec for a system that behaves identically

02

Big-bang rewrite

Same rules, same logic, same limitations — in a modern language. Debt preserved

03

Same decisions forever

Changing a business rule still requires an IT program. The structural problem is intact

Result: hundreds of millions spent on functional equivalence.

When Legacy Code Breaks Every AI Tool

Case Study | Payment Operations | Cobol Codebase Samples

PAYMENT RECORD STRUCTURE

```
*-----------------------------------------------------------------
* PAYMENT INPUT RECORD - 300 bytes
*-----------------------------------------------------------------
 FD  PAYMENT-FILE
     RECORDING MODE IS F
     BLOCK CONTAINS 0 RECORDS
     RECORD CONTAINS 300 CHARACTERS.
 01  PAYMENT-RECORD.
     05  PAY-ACCT-ID              PIC X(12).
     05  PAY-CUST-NAME            PIC X(60).
     05  PAY-CUST-ADDRESS         PIC X(80).
     05  PAY-AMOUNT               PIC 9(11)V99.
     05  PAY-CURRENCY-CODE        PIC X(3).
     05  PAY-TXN-TYPE             PIC X(4).
     05  PAY-REMARKS              PIC X(100).
     05  FILLER                   PIC X(28).

*-----------------------------------------------------------------
```

MULTI-LANGUAGE CUSTOMER HANDLING

```
 01  WS-SAMPLE-CUSTOMER-NAMES.
*    French name - UTF-8: é = C3A9, è = C3A8, ç = C3A7
     05  WS-CUST-NAME-FR          PIC X(40)
         VALUE 'Fran' & X'C3A7' & 'ois L' & X'C3A9'
             & 'ger' & X'C3A8' & 're'.
*    German name - UTF-8: ü = C3BC, ö = C3B6, ä = C3A4
     05  WS-CUST-NAME-DE          PIC X(40)
         VALUE 'J' & X'C3BC' & 'rgen M' & X'C3BC'
             & 'ller-K' & X'C3B6' & 'nig'.
*    Spanish name - UTF-8: ñ = C3B1, á = C3A1, é = C3A9
     05  WS-CUST-NAME-ES          PIC X(40)
         VALUE 'Jos' & X'C3A9' & ' Mart' & X'C3AD'
             & 'nez-Pe' & X'C3B1' & 'a'.
*    Nordic name - UTF-8: å = C3A5, ø = C3B8, æ = C3A6
     05  WS-CUST-NAME-NO          PIC X(40)
         VALUE 'Bj' & X'C3B8' & 'rn ' & X'C3A5'
             & 'kerstr' & X'C3B8' & 'm'.
*    Polish name - UTF-8: ł = C5BC, ó = C3B3, ż = C5BA
     05  WS-CUST-NAME-PL          PIC X(40)
         VALUE 'Rafa' & X'C5BC' & ' W' & X'C3B3'
             & 'jciechowski'.
*    Legacy LATIN-1 encoded name (single-byte high chars)
*    ä=E4  ö=F6  ü=FC  ß=DF
     05  WS-CUST-NAME-LATIN1      PIC X(40)
         VALUE X'C4' & 'nders' & X'F6' & 'n '
```

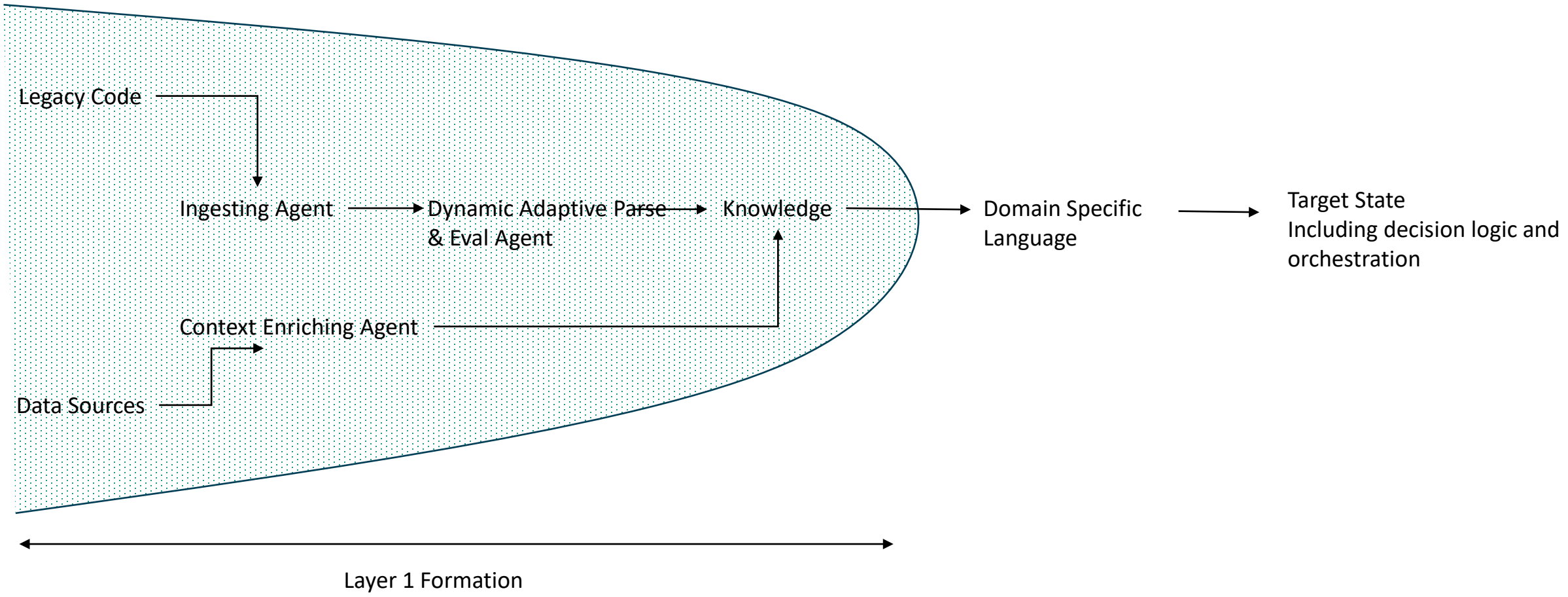
These special characters and multi-byte encoding workarounds are unique to this legacy codebase — and they consistently throw off AI-based code analysis tools. This is exactly what was happening for this card customer: automated parsing couldn't reliably interpret the COBOL structures, leading to misclassification and failed modernization attempts.

```
      *  VALIDATION RULES  (§ = rule ref, ¶ = sub-clause)           *
      *  § R1  Amount    :  MIN ≤ Amt ≤ MAX  (range: $0.01–$999999)  *
      *  § R2  Account   :  Cust ∈ MasterFile  AND  Status ≠ 'S'     *
      *  § R3  Currency  :  CurrCode ∈ {$,€,£,¥,₹,CHF}              *
      *  § R4  Balance   :  Balance ≥ OverdraftFloor  (warn if ≤ 0)  *
      *                                                              *
      *  MATH LEGEND:  ± tolerance  ≈ approx  × rate  ÷ ratio        *
      *                ∑ total  ∏ product  √ root  ∞ unbounded       *
      *  ARROWS:       → pass  ← reject  ↑ escalate  ↔ bilateral     *
      *                ⇒ implies  ⊂ derived from  ↓ next step        *
      *  MISC:         ° precision  § section  ¶ clause              *
      *==============================================================*
       2100-VALIDATE-PAYMENT.
      *    § R1 → Amount range check: MIN ≤ amount ≤ MAX
      *          Out-of-range ⇒ RC=8, flag ← INVALID, exit ↓
           IF PAY-AMOUNT < WS-MIN-PAYMENT
              OR PAY-AMOUNT > WS-MAX-PAYMENT
               MOVE +8             TO WS-VALIDATION-RC
               SET AMOUNT-INVALID  TO TRUE
               GO TO 2100-VALIDATE-EXIT
           END-IF
           SET AMOUNT-VALID        TO TRUE          *⇒ R1 passed

      *    § R2 → Customer ∈ MasterFile  AND  Status ≠ 'S' (Suspended)
      *          If ∉ file or status = Suspended ⇒ RC=8 ← reject ↓
           READ CUSTOMER-FILE INTO CUSTOMER-RECORD
              KEY IS CUST-ID = PAY-CUST-ID
              INVALID KEY
                 MOVE +8             TO WS-VALIDATION-RC
                 SET ACCOUNT-INVALID TO TRUE
                 GO TO 2100-VALIDATE-EXIT
           END-READ

           IF CUST-SUSPENDED                      *≠ Active ⇒ decline
              MOVE +8             TO WS-VALIDATION-RC
              GO TO 2100-VALIDATE-EXIT
           END-IF
           SET ACCOUNT-VALID       TO TRUE          *⇒ R2 passed

      *    § R3 → Currency ∈ {USD=$, EUR=€, GBP=£, JPY=¥, INR=₹, CHF}
      *          If ∉ approved table ⇒ RC=8, flag ← INVALID ↓
           PERFORM 2110-VALIDATE-CURRENCY          *⇒ R3 evaluated

      *    § R4 → Balance ≥ OverdraftFloor (° 2 decimal precision)
      *          Balance ≤ 0 ⇒ RC=4 warn (±tolerance applies) ↓
           IF WS-VALIDATION-RC = ZERO
              IF CUST-BALANCE <= ZERO
                 MOVE +4          TO WS-VALIDATION-RC
              END-IF
           END-IF

       2100-VALIDATE-EXIT.
           EXIT.
```


```
      *  VALIDATION RULES  (§ = rule ref, ¶ = sub-clause)           *
      *  § R1  Amount    :  MIN ≤ Amt ≤ MAX  (range: $0.01–$999999)  *
      *  § R2  Account   :  Cust ∈ MasterFile  AND  Status ≠ 'S'     *
      *  § R3  Currency  :  CurrCode ∈ {$,€,£,¥,₹,CHF}              *
      *  § R4  Balance   :  Balance ≥ OverdraftFloor  (warn if ≤ 0)  *
      *                                                              *
      *  MATH LEGEND:  ± tolerance  ≈ approx  × rate  ÷ ratio        *
      *                ∑ total  ∏ product  √ root  ∞ unbounded       *
      *  ARROWS:       → pass  ← reject  ↑ escalate  ↔ bilateral     *
      *                ⇒ implies  ⊂ derived from  ↓ next step        *
      *  MISC:         ° precision  § section  ¶ clause              *
      *==============================================================*
       2100-VALIDATE-PAYMENT.
      *    § R1 → Amount range check: MIN ≤ amount ≤ MAX
      *          Out-of-range ⇒ RC=8, flag ← INVALID, exit ↓
           IF PAY-AMOUNT < WS-MIN-PAYMENT
              OR PAY-AMOUNT > WS-MAX-PAYMENT
               MOVE +8             TO WS-VALIDATION-RC
               SET AMOUNT-INVALID  TO TRUE
               GO TO 2100-VALIDATE-EXIT
           END-IF
           SET AMOUNT-VALID        TO TRUE          *⇒ R1 passed

      *    § R2 → Customer ∈ MasterFile  AND  Status ≠ 'S' (Suspended)
      *          If ∉ file or status = Suspended ⇒ RC=8 ← reject ↓
           READ CUSTOMER-FILE INTO CUSTOMER-RECORD
              KEY IS CUST-ID = PAY-CUST-ID
              INVALID KEY
                 MOVE +8             TO WS-VALIDATION-RC
                 SET ACCOUNT-INVALID TO TRUE
                 GO TO 2100-VALIDATE-EXIT
           END-READ

           IF CUST-SUSPENDED                      *≠ Active ⇒ decline
              MOVE +8             TO WS-VALIDATION-RC
              GO TO 2100-VALIDATE-EXIT
           END-IF
           SET ACCOUNT-VALID       TO TRUE          *⇒ R2 passed

      *    § R3 → Currency ∈ {USD=$, EUR=€, GBP=£, JPY=¥, INR=₹, CHF}
      *          If ∉ approved table ⇒ RC=8, flag ← INVALID ↓
           PERFORM 2110-VALIDATE-CURRENCY          *⇒ R3 evaluated

      *    § R4 → Balance ≥ OverdraftFloor (° 2 decimal precision)
      *          Balance ≤ 0 ⇒ RC=4 warn (±tolerance applies) ↓
           IF WS-VALIDATION-RC = ZERO
              IF CUST-BALANCE <= ZERO
                 MOVE +4          TO WS-VALIDATION-RC
              END-IF
           END-IF

       2100-VALIDATE-EXIT.
           EXIT.
```


Mphasis
The Next Applied

How the solution Decodes Legacy at Scale



Mphasis Tria™ will help supercharges and accelerate towards the goals.

Target Outcomes: Speed to Market & Revenue Impact

Speed to Market

- **60–70% reduction** in deployment cycles
- **50% faster** new card product launches
- **80% fewer** manual touchpoints

Customer Revenue Impact

- **Uplift opportunity** in revenue
- **25–30% reduction** in compliance-related operational costs
- **10–15% improvement** in issuer client retention through faster, more responsive servicing

Mphasis Modernize™ Takes This to the Real World

Mphasis Modernize™ is a product line powered by Mphasis Tria™

- Goes beyond converting legacy systems into modern equivalents
- The end goal is not a modern copy of the old — it is enterprise Agency
- Extracts institutional knowledge embedded in legacy operations
- Builds governed decisioning frameworks for enterprise use
  Orchestrates agentic execution — enabling systems to sense, decide, and act autonomously
- Transforms enterprises from static, human-dependent engineering and business operations into autonomous, intelligent systems

Thank You!

For more information, please contact:

investor.relations@mphasis.com

AI WITHOUT INTELLIGENCE IS ARTIFICIAL™

At Mphasis, engineering has been in our DNA since inception.

Mphasis is an AI-led, platform-driven company with human-in-the-loop intelligence, helping global enterprises modernize, infuse AI, and scale with agility. The Mphasis.ai unit and Mphasis AI-powered 'Tribes' are focused on client outcomes and embed artificial intelligence and autonomy into every layer of the enterprise technology and process stack. Mphasis built NeoIP™, a breakthrough AI platform that orchestrates a powerful pack of AI solutions and platforms to deliver impactful outcomes across the enterprise IT value chain, as we believe ***'AI Without Intelligence Is Artificial™'***. Mphasis NeoIP™ is powered by the Mphasis Ontosphere™, a dynamic and ever-evolving knowledge base, delivering continuous and constant innovation through perpetual intelligent engineering—driving end-to-end enterprise transformation.

At the heart of our approach is customer-centricity—reflected in our proprietary Front2Back™ transformation framework, which uses the exponential power of cloud and cognitive to deliver hyper-personalized digital experiences ($C=X2C^2_{TM}=1$) and build strong relationships with marquee clients. Our Service Transformation solutions enable enterprises to pivot from legacy systems and operations to secure, adaptive, cloud-first operating models with minimal disruption. Continuous investments in platforms, such as the Neo series, enable enterprises to stay efficient, relevant, and ahead in a dynamic AI-first world. Mphasis is a Hi-Tech, Hi-Touch, Hi-Trust company, rooted in a learning and growth culture. Click here to know more. (BSE: 526299; NSE: MPHASIS)

Important Confidentiality Notice