# Full-Cycle Automation: Drug-to-Reimbursement

Linking Drug Inventory, Patient Records, and Insurance Billing through an advanced N8N Workflow Pipeline.

**GOAL**

100% Reimbursement for High-Cost Injectables

**SYSTEM**

N8N Orchestration Engine

# Executive Summary

## The Challenge

In ophthalmology, high-cost injectable drugs represent significant financial risk. A recurring failure to scan/record vial usage creates a broken link between:

📄 Supplier Invoice (Cost)

🔗 **Broken Link**

🧾 Insurance Claim (Revenue)

## The Impact

This omission leads directly to unreimbursed costs.

The practice is forced to absorb expenses that can be hundreds or thousands of dollars per single vial.

**Solution: Eliminate the "Human Action" dependency.**

# Problem Statement & Strategic Goals

## The Problem

We are losing revenue due to a manual documentation gap at the point of drug administration.

The lack of an integrated, mandatory checkpoint allows vials to be dispensed without a corresponding digital billing record, resulting in non-payment from insurance carriers.

## Primary Goal

**Ensure every purchased vial is reimbursed.**

By automatically validating and linking supplier, inventory, and billing data.

## Secondary Goal

Determine true medication profitability after full reimbursement reconciliation.

# Systems Integration Landscape

## Besse (Supplier)

Order #, Type, Qty, Cost

Source of Truth: Cost & Quantity

## Podis (Inventory)



Zoledronic Acid Injection
4 mg/5 mL
(0.8 mg/mL)
Concentrate for intravenous infusion
Sterile Concentrate
Not for direct injection.
Dose must be diluted.

Dispense Log, Patient ID

Source of Truth: Usage

## MoDMed (EHR)

Insurance, Claims, Reimb.

Source of Truth: Financial Outcome

## Fridge Log

Physical Daily Count

Audit Validation Point

# The Proposed N8N Orchestration

N8N acts as the central engine to implement a closed-loop data integration plan.

| Phase | N8N Functionality | Objective |
|---|---|---|
| **Phase 1:** **Ingestion** | Trigger on invoice receipt. Extract (OCR), parse, and standardize data from Supplier (Besse). | Establish the total digital count and cost baseline of purchased vials. |
| **Phase 2:** **Reconciliation** | Monitor Inventory (Podis) for usage, link to patient record, and cross-check against stock. | Mandate a digital link for every dispensed vial before billing proceeds. |
| **Phase 3:** **Billing & Audit** | Automate claim submission (MoDMed) and flag discrepancies. | Ensure every used vial is billed and reimbursed. |

# Gap Analysis

**Current Workflow Gap**

- 1. Invoice Received

- 2. Vials entered into Inventory

- 3. Vial used on patient

- **4. HUMAN ACTION REQUIRED (Scan/Record)**
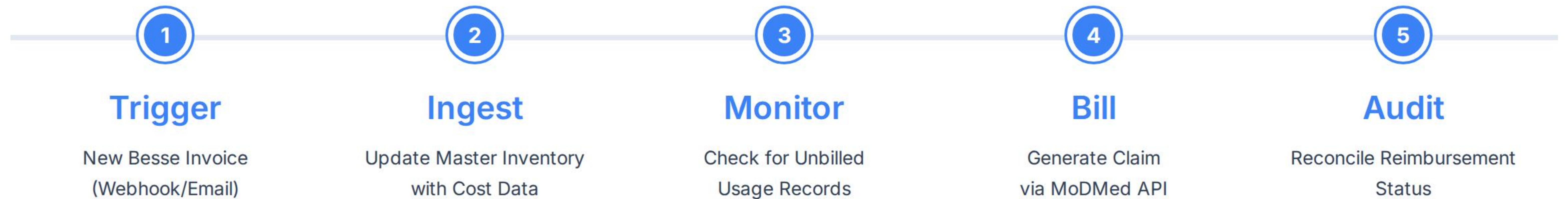
- 5. Billing team generates claim

**The Failure Point:** Step 4 is a single point of failure. If staff forget to scan, the downstream claim submission lacks evidence, leading to lost revenue.

# Full-Fledged Pipeline Overview

**1**

## Trigger

New Besse Invoice
(Webhook/Email)

**2**

## Ingest

Update Master Inventory
with Cost Data

**3**

## Monitor

Check for Unbilled
Usage Records

**4**

## Bill

Generate Claim
via MoDMed API

**5**

## Audit

Reconcile Reimbursement
Status

# Architecture & Integration Points

| Integration Point | N8N Node Type | Connection Protocol |
|---|---|---|
| 📄 Besse Invoice | Email Trigger + AI/OCR Node | IMAP / Optical Character Recognition |
| 📦 Podis Log | Database Node (Postgres/MySQL) | JDBC/ODBC Direct DB Connection |
| 💻 MoDMed Portal | HTTP Request Node | REST/SOAP API (Billing & Status) |
| ⚠ Error Log | Slack / Google Sheets Node | Webhook / API Integration |

# N8N Workflow Logic

## Key Node Sequence:

1. Webhook (New Invoice)

2. AI/OCR Node (Extract Data)

3. Database (Inventory Update)

4. Cron Trigger (Usage Check)

5. IF Node (Is Billed?)

6. HTTP Request (Submit Claim)

7. Wait Node (Processing)

8. IF Node (Is Reimbursed?)
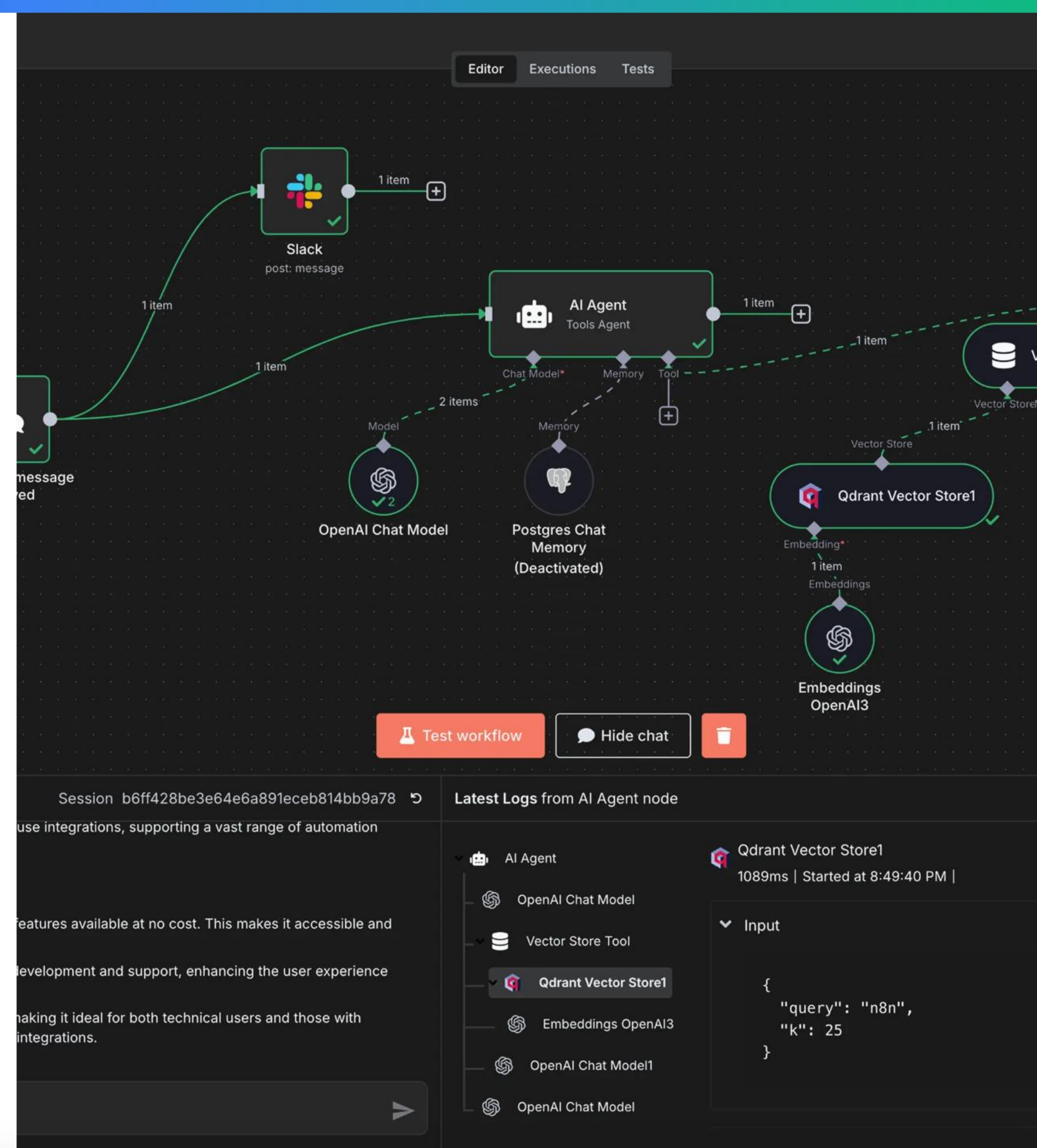
# Phase 1: Invoice Data Ingestion

## Trigger & Extraction

**Node:** Besse Invoice Trigger (Email + OCR)

The workflow activates upon receipt of a new PDF/CSV invoice.

> **OCR/AI Node:** Reliably extracts:
>
> - Order Number
> - Medication Type
> - Quantity (N vials)
> - Full & Discount Price


Data Extraction Concept

# Phase 1: Inventory Master Update



## Digital Twin Creation

**Node:** Database Node

For every Quantity (N) vials purchased, N individual inventory records are created.

> **Record Initialization:**
>
> - STATUS = 'IN_STOCK'
> - BILLED_TO_PATIENT = 'FALSE'

This creates the mandatory digital paper trail required for future reconciliation.

# Phase 1: Physical Count Validation

## Closing the Loop

**Node:** Webhook/Manual Trigger + IF Node
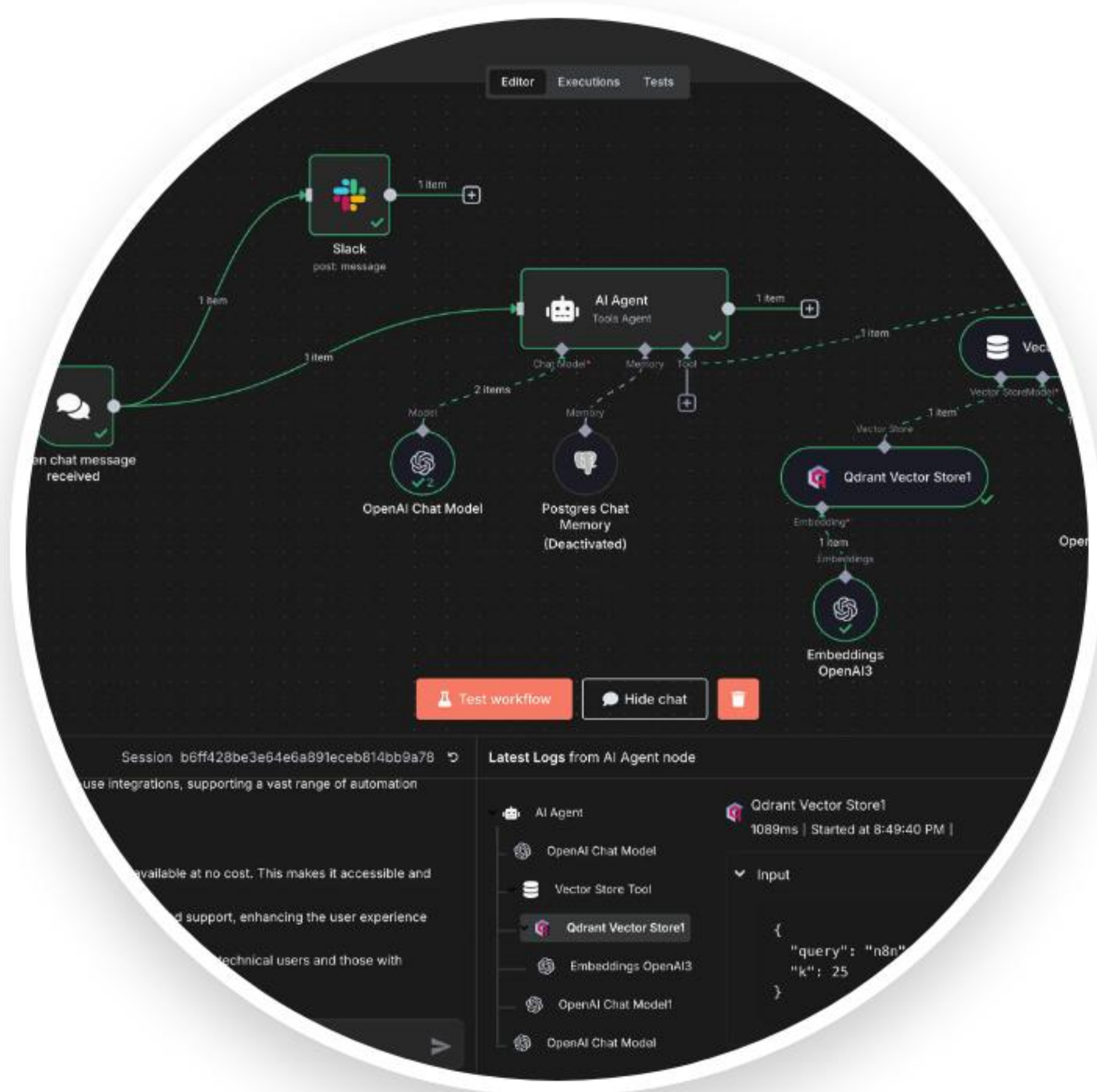
Integrates the "Fridge Log" physical count.

- **Logic:** IF (Digital Count == Physical Count) THEN Proceed.
- **Audit:** Mismatches trigger an immediate alert to the Inventory Manager via Slack.

Prevents stock issues before drug usage begins.

# Phase 2: Usage Monitoring Trigger



## Continuous Monitoring

**Node:** Cron Schedule (Every 15 Mins)

Queries the Dispense Log for records meeting criteria:

- `ADMINISTRATION_DATE = Today`
- `IS_RECONCILED = 'FALSE'`

**Batching:** Processes each usage record individually (Atomic) to ensure fail-safe billing for every single vial.

# Phase 2: Data Enrichment & Linking

## Building the Claim Object

**Node:** Database + Merge Node

Combines data sources into a single object:

- **Input:** Vial ID, Patient ID (from Usage)

- **Enrichment 1 (MoDMed):** Fetch Insurance & Consent Status.

- **Enrichment 2 (Inventory):** Fetch original Cost/Discount Price.

Result: A complete data packet ready for billing.

# Phase 2: Billing Submission


Digital Claim

## Automated Submission

**Node:** Code Node → HTTP Request

1. **Safety Check:** IF Consent == APPROVED. (Else: Error Branch)

2. **Payload:** Construct FHIR/JSON claim body.

3. **Submit:** POST to MoDMed API.

4. **Update:** Set Podis record to 'PENDING'.

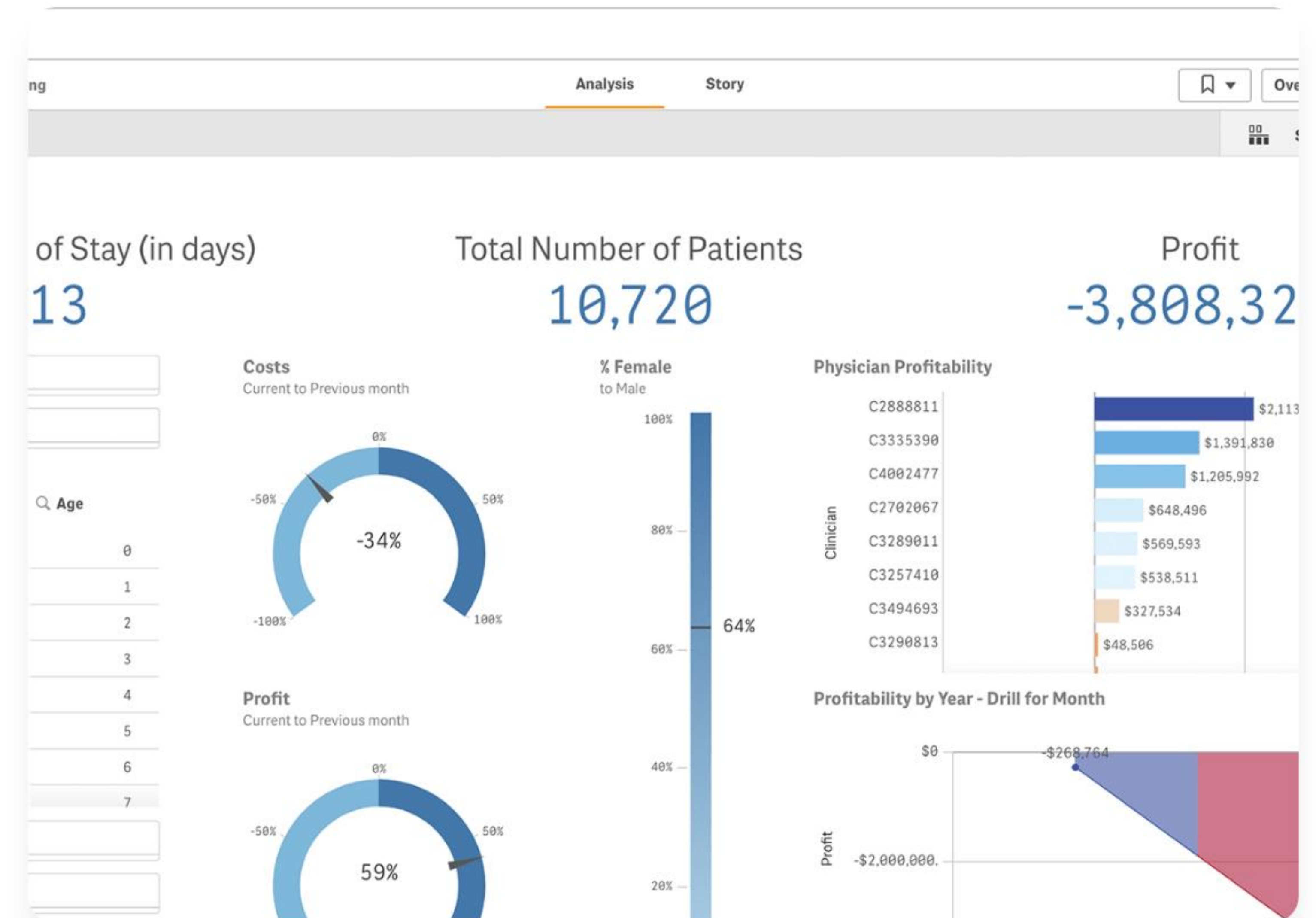**Result:** Immediate claim submission without human intervention.

# Phase 3: Claim Status Monitoring

## Closing the Financial Loop

**Node:** Webhook (MoDMed) or Cron Check

Asynchronous monitoring of the submitted claim.

- Queries MoDMed using Claim ID.
- Retrieves Status: **PAID**, **DENIED**, or **PENDING**.
- Extracts Amount Reimbursed and Denial Codes (if applicable).

# Phase 3: Financial Reconciliation

## Profitability Calculation

**Node:** Code Node

The core business logic calculation is performed for every vial:

$$Profit = Reimbursed - Cost + Copay$$

The full record (Cost, Usage, Reimbursement, Profit) is logged into the BI Database.

## Final Status Update

**Success:** Flag record as 'PAID'.

**Analytic Value:** Allows precise tracking of net revenue per drug type, identifying which medications are actually profitable after insurance adjustments.

# Phase 3: Discrepancy & Alerts

## Denial Handling

IF Status == 'DENIED':

1. Trigger immediate Slack/Teams notification to Billing Team.

2. Include Vial ID, Patient ID, and Denial Code.

3. Add to "Resubmission Queue" Google Sheet.

## Profitability Alerts

**Low Margin Warning:**

If Profit < Threshold, alert Management.

This prompts a review of supplier contracts or insurance fee schedules.

✔ **Outcome:** Proactive revenue cycle management.

# System Robustness (Level 3)

### Decoupled Workflows

Separation of Ingestion, Usage, and Reconciliation prevents a single failure from crashing the entire system.

### Error Triggers

Dedicated Error Trigger nodes catch API/DB failures, ensuring the workflow never fails silently.

### Immutable Audit

Every major data event is logged to a secure file for full compliance and audit trails.

# Image Sources



https://www.henryscheindental.com/Products/1436983_US_Group_01_600×600.jpg

Source: www.henryscheindental.com



https://media.gettyimages.com/id/1251428223/sv/foto/pharmacist-scanning-barcode-of-medicine-drug.jpg?s=612×612&w=gi&k=20&c=3b5FfXDVdHIaXSgjH3SQKAwyCSnnBY-P6xbf3bq-icQ=

Source: www.gettyimages.se



https://n8niostorageaccount.blob.core.windows.net/n8nio-strapi-blobs-prod/assets/Agent_chat_818315ae64.webp

Source: n8n.io



https://www.definitivehc.com/sites/default/files/2023-03/Medical%20billing%20process_28MAR23.png

Source: www.definitivehc.com



https://static.vecteezy.com/system/resources/previews/050/442/116/large_2x/modern-data-center-with-server-racks-and-blue-led-lights-for-cloud-computing-and-data-storage-photo.jpg

Source: www.vecteezy.com



https://assets.qlik.com/image/upload/w_1720/q_auto/qlik/glossary/dashboard-examples/seo-financial-dashboards-cfo-dashboard_f9frfk.png

Source: www.qlik.com