

AI-04005 Medical Patient Workflow

Owner	Ⓜ Rigel Arcayan
Tags	MD
Created time	July 8, 2026 1:29 AM

Generate a typical medical patient workflow from beginning to end but map as swim lanes as front desk, medical assistant, MD billing, labs, etc and use the n8n interface methodology.

A typical medical patient workflow can be modeled like an **n8n automation workflow** using **swim lanes** where each lane represents a healthcare role/system. The idea is that every patient interaction becomes a sequence of **events** → **triggers** → **actions** → **decisions** → **integrations** → **outcomes**.

Below is a **primary care clinic workflow (new patient + visit + follow-up)** mapped in an n8n-style architecture.

Medical Patient Journey Workflow (n8n Swim Lane Model)

PATIENT

|

v

FRONT DESK

|

v

MEDICAL ASSISTANT

|

v

MD / PROVIDER

|

v

LAB / IMAGING

|

v

BILLING / INSURANCE

|

v

PHARMACY

|

v

PATIENT FOLLOW-UP

Swim Lane 1 — PATIENT

n8n Node Type	Event
Trigger	Patient requests appointment
Form Node	Online registration
Database Node	Demographics captured
Upload Node	Insurance card, ID, records
Wait Node	Appointment date arrives
Notification Node	SMS/email reminder

Workflow Start Trigger

Patient submits appointment request

|

↓

Create Patient Record

|

↓

Send Confirmation

Swim Lane 2 — FRONT DESK

n8n Workflow

Webhook Trigger

|

↓

Check Patient Exists?

|

YES ————— NO

|

|

Update Chart Create MRN

|

Verify Insurance

|

Schedule Visit

|

Send Appointment Confirmation

Systems Connected

- EHR
- Insurance Eligibility API
- Calendar System
- SMS Gateway

Example n8n nodes:

[Webhook]

|

[Patient Lookup]

|

[IF Existing Patient]

|

[Create Appointment]

|

[Send SMS]

Swim Lane 3 — MEDICAL ASSISTANT

Pre-Visit Workflow

Appointment Arrival

|

↓

Patient Check-In

|

↓

Vitals Collection

|

↓

Medication Reconciliation

|

↓

Chief Complaint Documentation

|

↓

Room Patient

|

↓

Notify MD

n8n representation:

Trigger:

Patient Status = ARRIVED

|

v

Collect:

- BP
- HR
- Temperature
- Weight
- Allergies

|

v

Update EHR

|

v

Task Assigned → MD

Swim Lane 4 — MD / PROVIDER

Clinical Decision Workflow

Patient Roomed

|

↓

Review History

|

↓

Interview Patient

|

↓

Physical Exam

|

↓

Generate Assessment

|

↓

Create Differential Diagnosis

|

↓

Order Tests

|

↓

Create Treatment Plan

AI-Assisted MD Workflow

Example:

Patient Symptoms Entered

|

↓

AI Clinical Assistant

|

↓

Generate:

- Possible Diagnosis
- Red Flags
- Recommended Tests
- Medication Review

|

↓

MD Validates

|

↓

Final Decision

Important:

AI provides decision support; the licensed clinician remains responsible for diagnosis and treatment decisions.

Swim Lane 5 — LAB / IMAGING

Order Processing

MD Orders Lab

|

↓

Order Sent Electronically

|

↓

Lab Performs Test

|

↓

Result Generated

|

↓

Result Interface Returns

|

↓

EHR Updated

n8n:

[EHR Trigger]

|

↓

[HL7/FHIR Connector]

|

↓

[Lab System]

|

↓

[Result Parser]

|

↓

[Update Patient Chart]

Swim Lane 6 — AI Medical Record Intelligence

(Optional advanced workflow)

New Lab Result Arrives

|

↓

OCR / FHIR Parser

|

↓

AI Clinical Summary

|

↓

Risk Detection

|

↓

MD Notification

Example:

Lab Result:

LDL = 220

AI Agent:

"Critical hyperlipidemia detected.

Compare previous values.

Consider cardiovascular risk review."

|

↓

MD Reviews

Swim Lane 7 — BILLING / INSURANCE

Revenue Cycle Workflow

Visit Completed

|

↓

Generate CPT Codes

|

↓

Generate ICD-10 Diagnosis

|

↓

Submit Claim

|

↓

Insurance Processing

|

↓

Payment Received

n8n:

[EHR Visit Complete]

|

↓

[Coding AI Assistant]

|

↓

[Claim Generator]

|

↓

[Insurance API]

|

↓

[Payment Reconciliation]

Swim Lane 8 — PHARMACY

Prescription Created

|

↓

Electronic Prescription

|

↓

Pharmacy Receives

|

↓

Medication Filled

|

↓

Patient Notification

Swim Lane 9 — FOLLOW-UP CARE

Chronic Disease Example

Diagnosis:

Diabetes

|

↓

Create Care Plan

|

↓

Schedule Follow-up

|

↓

Monitor Labs

|

↓

AI Population Health Alert

|

↓

MD Intervention

Complete n8n-Style Master Workflow

START

|

|

Webhook:

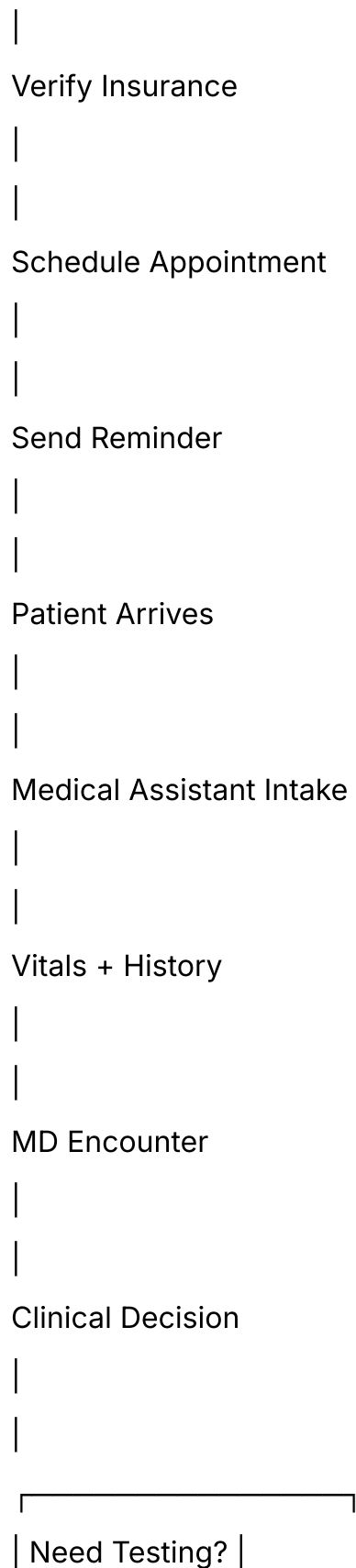
Patient Appointment Request

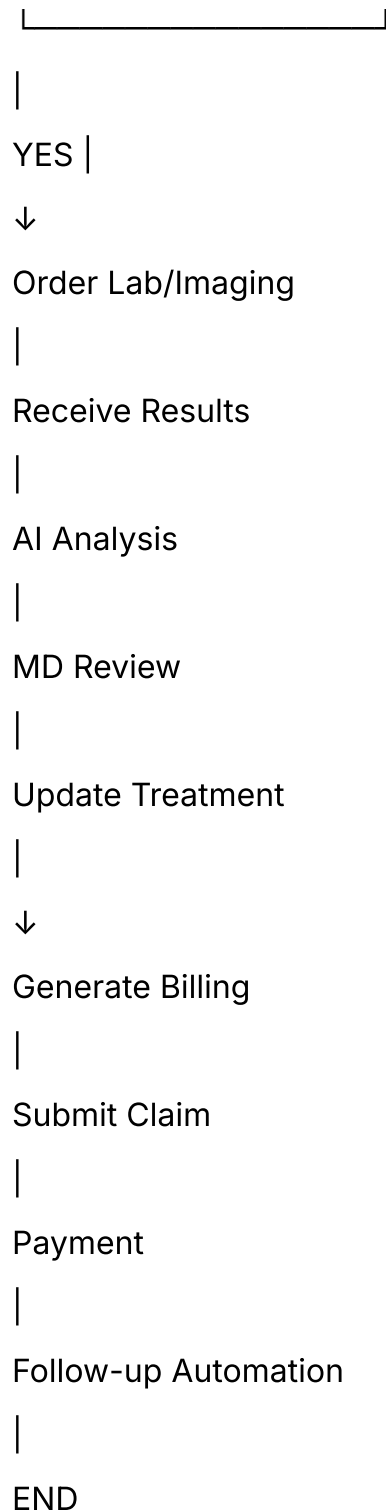
|

|

Create/Update Patient Record

|





Example n8n Node Architecture

TRIGGERS

- |
- |— Appointment Created
- |— Patient Arrived
- |— Lab Result Received
- |— Insurance Response
- |— Payment Received

PROCESSING NODES

- |
- |— Patient Database
- |— EHR Connector
- |— AI Medical Agent
- |— Rules Engine
- |— Document OCR
- |— Notification Engine

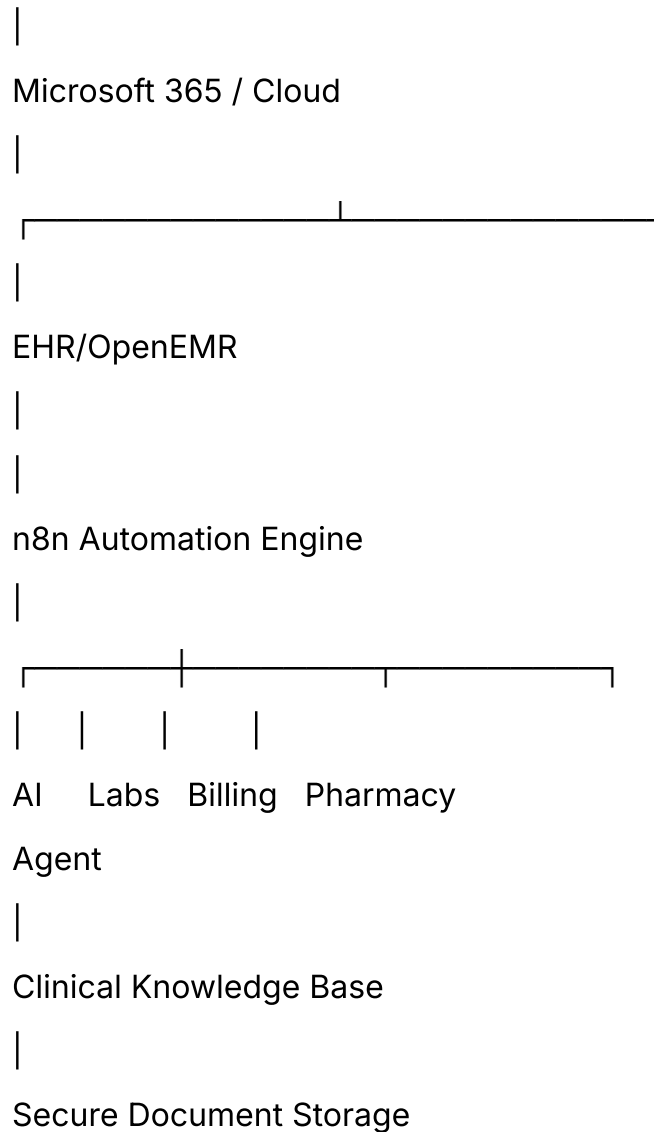
OUTPUT NODES

- |
- |— SMS
- |— Email
- |— MD Task
- |— Pharmacy
- |— Billing System
- |— Patient Portal

Small Medical Office Cloud Architecture (modern implementation)

PATIENT PORTAL

|



Convert the workflow mentioned into visual sketch and use the n8n workflow template

For a small medical clinic, this workflow pattern can be implemented with:

- n8n as the automation engine
- OpenEMR or another EHR as the clinical record system
- Microsoft 365 / SharePoint for secure documents

- FHIR/HL7 interfaces for labs and imaging
- AI assistants for summarization, chart review, and workflow automation

This is essentially the blueprint for a **“digital medical office operating system”** where humans make clinical decisions and automation handles coordination, data movement, and administrative tasks.

To model a medical patient workflow using the **n8n interface methodology**, we treat every human action as an event trigger, data payload, or webhook, and map the traditional "swimlanes" into functional node paths, conditional routing (IF/Switch nodes), and data transformations.

Here is how a typical patient encounter maps out from check-in to billing inside an n8n workflow canvas.

🔗 Trigger: Patient Event

- **Node Type:** Webhook / App Trigger (e.g., EHR API or Phreesia Check-in)
- **Payload:** { "patientId": "12345", "appointmentId": "9876", "status": "arrived" }

🔄 Lane 1: Front Desk (Intake & Verification)

[Webhook: Arrived] → [HTTP Request: Fetch Patient Data] → [IF: Copay Due?] → (Yes) → [Stripe/POS Action]

→ (No) → [Update EHR Status: Ready for MA]

- **Node 1: HTTP Request (Fetch Demographics & Insurance)**
 - *Action:* Queries the Electronic Health Record (EHR) to pull current insurance eligibility and demographic data.
- **Node 2: IF Node (Copay / Balance Check)**
 - *Condition:* {{ \$json.copayAmount > 0 }}
 - *True Path:* Route to **Payment Node** (collect copay via POS integration).

- *False Path*: Pass directly to next node.
- **Node 3: Set Node (Update Status)**
 - *Action*: Updates patient status in the EHR queue to "Ready for Triage". This action acts as the webhook trigger that alerts the next lane.

ðŸ™° Lane 2: Medical Assistant (Triage & Vitals)

[Trigger: Status Change] → [EHR Form Input: Vitals] → [Switch: Vitals Out of Range?] → (Flag to MD)

→ (Normal) → [Update EHR: Ready for MD]

- **Node 1: EHR Trigger (Patient Status == "Ready for Triage")**
 - *Action*: Pushes a notification to the MA's dashboard.
- **Node 2: EHR Form Input Node (Vitals Capture)**
 - *Data Collection*: Captures Blood Pressure, Heart Rate, Temp, and Weight.
- **Node 3: Switch Node (Clinical Red Flags)**
 - *Logic*: Evaluates if vitals cross critical thresholds (e.g., BP > 180/120).
 - *Path A (Critical)*: Immediately injects an alert into the Provider's active queue.
 - *Path B (Normal)*: Automatically updates the room status to "Ready for Provider".

ðŸŒŒ ðŸ™Š Lane 3: MD / Provider (Exam & Documentation)

[Trigger: Room Ready] → [EHR Action: Clinical Note] → [Sub-Workflow Split]

→ Path A: [Labs Order]

→ Path B: [E-Prescription]

→ Path C: [Checkout/Coding]

- **Node 1: EHR Trigger (Patient Status == "Ready for Provider")**
 - *Action:* Assigns the active chart to the provider's screen.
- **Node 2: Wait Node (Clinical Encounter)**
 - *Action:* Pauses the automated workflow while the physician conducts the physical exam, reviews history, and dictates the SOAP note.
- **Node 3: Fork / Multi-Branch Node (Orders)**
 - Based on the MD's assessment, this forks the payload into parallel streams:
 - *Branch 1 (Labs):* Triggers the Laboratory lane if `{{ $json.orders.labs == true }}`.
 - *Branch 2 (Rx):* Triggers the e-Prescription router via Surescripts API.
 - *Branch 3 (Discharge):* Routes the chart to Coding & Billing.

§ Lane 4: Labs & Diagnostics (Execution)

[Fork: Labs Order] → [HTTP Request: Send to LIS] → [Wait for Webhook: Results] → [IF: Critical?] → (Yes) → [Alert MD] → (No) → [Send to Patient Portal]

- **Node 1: HTTP Request (Send to LIS)**
 - *Action:* Automatically transmits order details to the Laboratory Information System (LIS).
- **Node 2: Wait for Webhook Node (Lab Results)**
 - *Action:* Holds the sub-workflow until the lab processing analyzer returns an HTTP POST payload containing the panel values.
- **Node 3: IF Node (Abnormal Flag Check)**
 - *True (Critical Value):* Flags the MD lane for urgent review.
 - *False (Normal Value):* Auto-publishes the results to the Patient Portal.

ðŸ’° Lane 5: Billing & Coding (Administration)

[Trigger: Note Finalized] âž” [Execute Workflow: Auto-Code Assessment] âž” [Clearinghouse API] âž” [IF: Clean Claim?] âž” (Yes) âž” [Post Payment] âž” (No) âž” [Slack Alert: Billing Team]

- **Node 1: EHR Trigger (Encounter Closed / Signed by MD)**
 - *Payload:* { "icd10": ["I10"], "cpt": ["99213"], "modifier": "25" }
- **Node 2: Data Transformation Node (Format Claim File)**
 - *Action:* Converts the clinical encounter payload into a standardized EDI 837 transaction file format required by insurance payers.
- **Node 3: HTTP Request (Submit to Clearinghouse)**
 - *Action:* Transmits the EDI file to the billing clearinghouse API.
- **Node 4: IF Node (Claim Status Response)**
 - *Condition:* {{ \$json.status == "Accepted/Clean" }}
 - *True Path:* Route to **Accounts Receivable ledger** to await remittance advice (EDI 835).
 - *False Path (Rejection):* Generates a high-priority task ticket in the billing team's work queue for human intervention.

ðŸ÷^ Workflow Completion

- **Final Node:** Set Node (Archive Data & Log KPIs)
- **Action:** Marks the appointmentId as Completed, logs the total cycle time (Check-in to Discharge) to an analytics database, and terminates the active execution branch.