# AI Healthcare Receptionist (ANDREA)

**Webpage - Link**
**GitHub - Link**

*A full pipeline explanation of the multi-agent, workflow-orchestrated voice AI receptionist system built for WellnessFirst Dental Care.*

---

# Overview

I built a production-ready **voice AI receptionist system** for **WellnessFirst Dental Care**, powered by:

- An **MCP Router agent + 7 specialized sub-agents**
- **LLM-driven workflow orchestration**
- **n8n automation workflows via an MCP Server**
- **GPT-4o-mini cluster** for ultra-low latency call handling
- **VAPI telephony + Webhook end-of-call reporting**

While the implementation focuses on healthcare + dental front-desk automation, the underlying architecture is **domain-agnostic**.
This same system pattern applies to:

- Medical clinics
- Service-based industries (salons, spas, home services, repair services)
- SaaS startups scheduling demos, trials, POCs
- Customer support triage
- Operational intake systems
- Lead qualification systems

### Why this is different

Most voice AI demos are **single-agent chatbots** that collapse under real-world complexity.
This system is built on:

- **Router-based multi-agent architecture**

- **Specialized agents with strict scopes**
- **Persistent context variables across workflows**
- **Tool calling guardrails**
- **n8n workflow orchestration via MCP server**
- **Knowledge-base grounding** to eliminate hallucinations

### Result

A receptionist AI that:

- Handles open-ended, multi-turn conversations
- Books, updates, and cancels appointments
- Collects patient intake data only when needed
- Provides symptom-based triage questions
- Works within clinic hours
- Understands insurance rules
- Logs every call with an End-Of-Call (EOC) report

This system has been tested with **100+ simulated calls** including stress tests, edge cases, and chaotic user behavior.

---

# The Problem: Administrative Burden Across Industries

### Healthcare

Front-desk staff spend **40% of their time** on:

- Appointment scheduling
- Insurance handling
- Patient intake
- Repetitive FAQs

Patients also experience:

- Long hold times
- Repeating themselves multiple times
- Difficulty booking emergency visits

Healthcare admin waste in the U.S. is nearly **$1 trillion** annually.

### Legal

Law firms waste 30–40% of billable hours on routine intake and processing.

### Customer Support

Teams drown in repetitive inquiries, while customers wait in queues.
Traditional IVR menus are outdated and inflexible.

### Service-Based Industries

Real estate agents, auto repair shops, home-cleaning services, HVAC businesses, personal coaches, and dental chains all suffer from:

- High inbound call volume
- Frequent schedule adjustments
- Lead qualification issues

### SaaS Companies

Early-stage SaaS startups need:

- Automated demo scheduling
- POC request handling
- Prospect qualification

---

# Agent Workflows: The Solution

Instead of a single, overloaded assistant, the system uses **multi-agent specialization**:

---

# 🌐 The Architecture: 1 MCP Router + 7 Specialist Agents

The **Router Agent** interprets user intent and routes to one of seven specialized workflows:

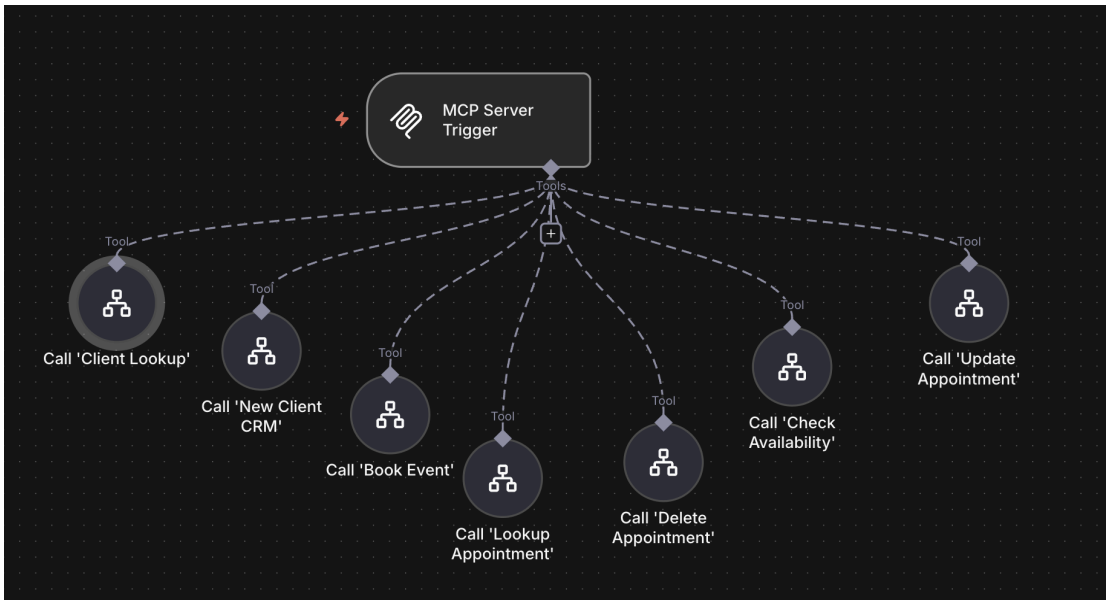
# MCP Server – n8n Tool Communication Layer

### n8n MCP Server

**Purpose:**
A middleware layer that enables the voice agent to securely call n8n workflows.

- Receives structured JSON inputs from VAPI
- Matches incoming tool requests to correct n8n workflow
- Executes workflow with required parameters
- Returns output back to VAPI in real time
- Handles all CRM and scheduling logic

This allows the LLM to operate with **strict tool boundaries** and avoids hallucinating API details.



---

# 1. Greeting & General Inquiry Agent

Purpose:

- Answer general questions (services, hours, insurance, location, doctors)
- Access the WellnessFirst Knowledge Base
- Avoid collecting unnecessary user data

Triggers:

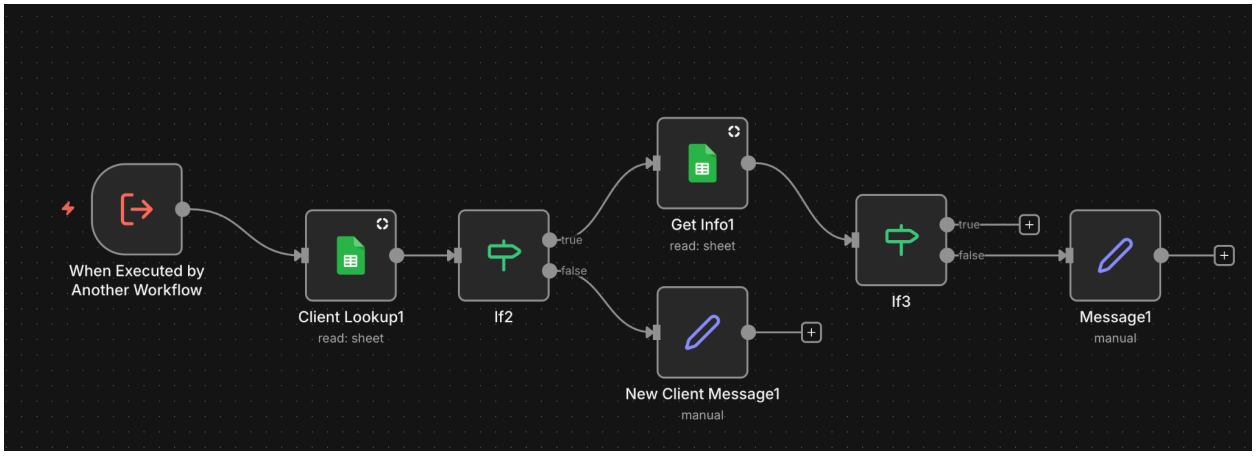
- Non-medical questions
- High-level inquiries

---

# 2. CRM Lookup Agent

Purpose:

- Lookup patient data in CRM via n8n
- Convert email to lowercase
- Trigger pre-tool filler phrase
- Retrieve existing patient profile

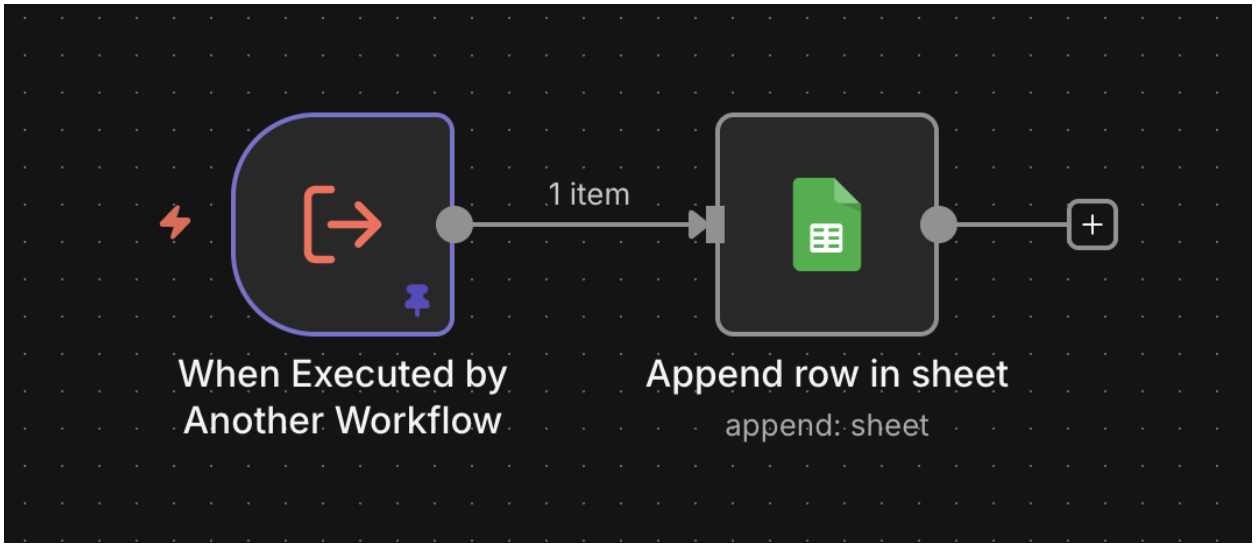



---

## 3. New Patient Intake Agent

Purpose:

- Register first-time callers
- Collect name, email, phone
- Confirm spelling
- Create CRM entry via n8n
- Only activated when needed

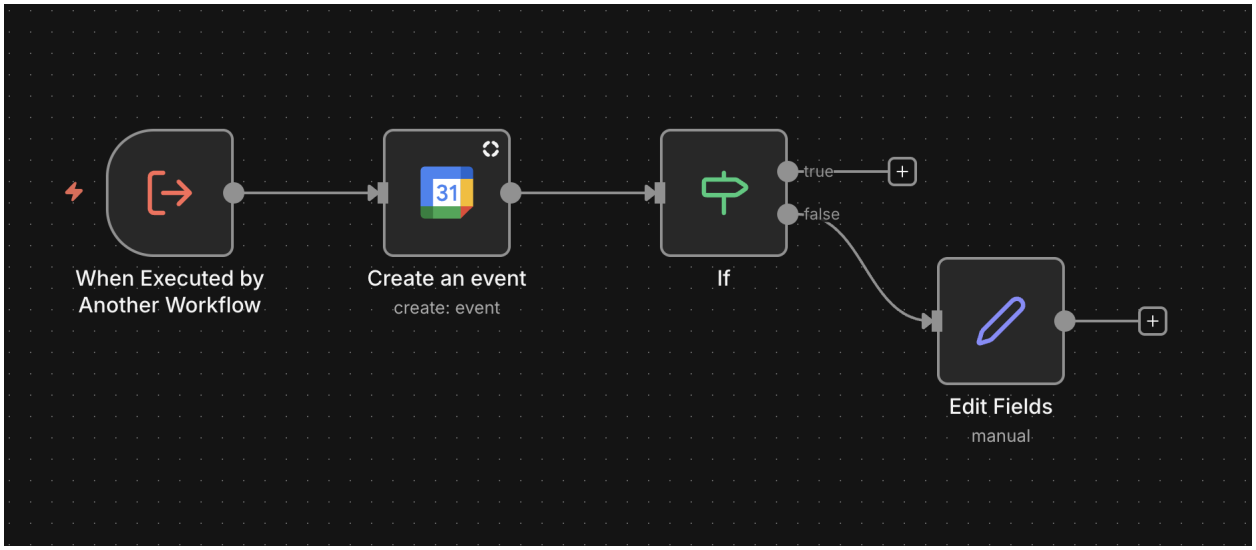



---

## 4. Appointment Availability Agent

Purpose:

- Query n8n for open times
- Evaluate busy vs. open slots
- Check “today” or custom date ranges
- Ensure clinic operating-hour constraints

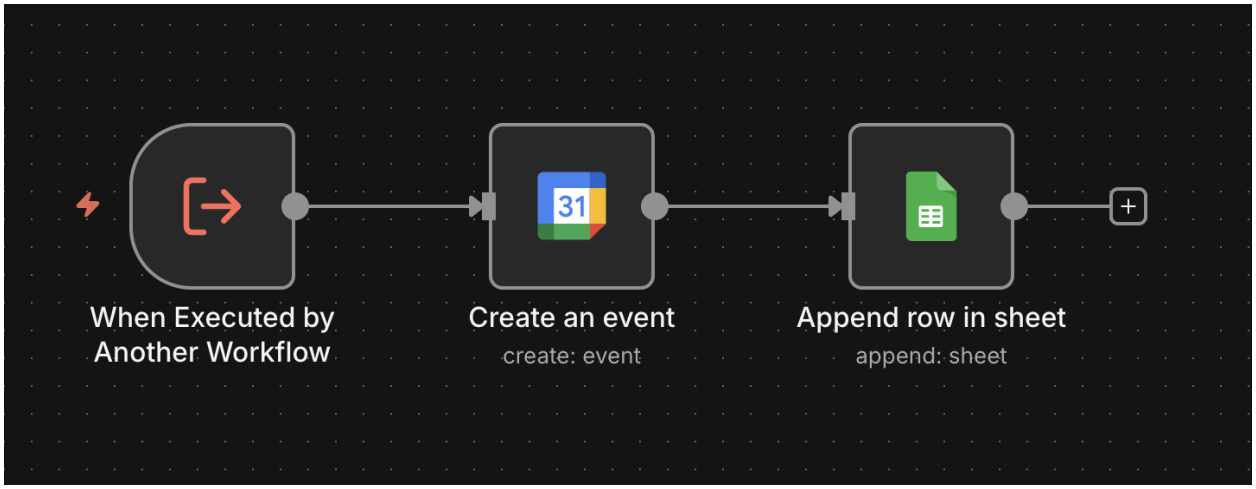



---

## 5. Appointment Booking Agent

Purpose:

- Gather appointment type via symptom-based questions
- Confirm start time → compute end time (+1 hour)
- Book appointment in CRM via n8n
- Assign doctor based on issue
- Trigger insurance confirmation after booking

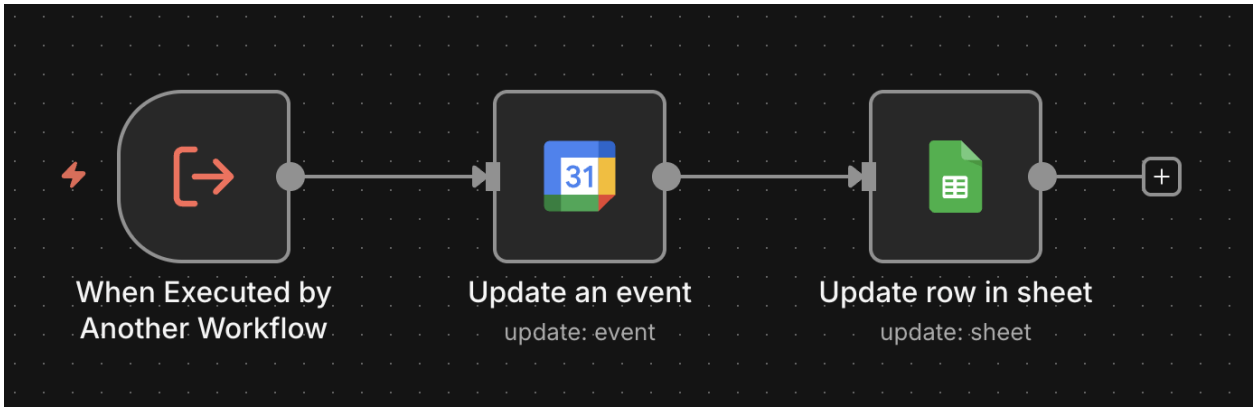


---

## 6. Appointment Update & Cancellation Agent

Purpose:

- Lookup existing appointment via event ID
- Modify date/time
- Re-check availability before applying changes
- Delete appointment via n8n




---

## 7. Insurance Confirmation Agent

Purpose:

- After booking → verify insurance provider
- Compare to **Knowledge Base** accepted list
- Provide guidance on documents to bring
- Inform patient when insurance may not be covered

---

# End of Call (EOC) Report

### Trigger

- Sent automatically via **VAPI Webhook (POST)**
- Not an MCP tool
- Fires **after the call ends**

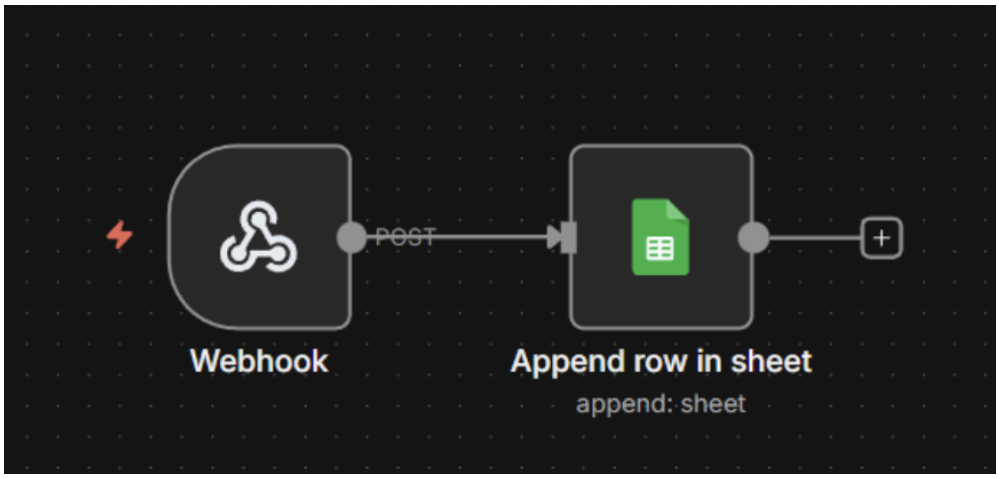
### Data Received

- `message.analysis.summary`
- `message.analysis.structuredData.Outcome`

### Process

1. Append a new row to **Google Sheets → "Call Log" tab**
2. Include:
    - Timestamp
    - Call Summary
    - Call Outcome

This creates a persistent record of every call for quality assurance and auditing.




---

# Conditional Routing Logic

Agents do not blindly pass control—they evaluate conversation state.

### Example:

After collecting symptoms and determining the appointment type, Andrea checks:

- If she already has patient identity
- If CRM lookup is needed
- If the patient is new → trigger intake flow
- Otherwise move directly to availability search

This ensures the user never repeats data unnecessarily.

---

# Context Preservation

The system stores important variables across agent handoffs:

- `email`
- `name`
- `phone`
- `appointment_type`
- `symptoms`
- `insurance_provider`
- `preferred_time`
- `doctor_selected`

Every agent reads these values so the user never repeats themselves.

---

# The Build: From Chaos to Production

### Phase 1 — The Single Agent Failure

Using one giant system prompt resulted in:

- 3000+ word prompt

- Hallucinated appointment confirmations
- Missed tool calls
- Over-collection of patient data
- Impossible debugging

### Phase 2 — Multi-Agent Breakthrough

Splitting into 7 focused agents:

- Reduced hallucinations to near zero
- Improved latency
- Increased reliability
- Simplified debugging and maintenance

### Phase 3 — Stress Testing (100+ calls)

Simulated calls included:

- “My tooth hurts and I need to come today.”
- “What insurance do you take?”
- “Actually wait, I don’t know my schedule—give me a second…”
- “I think I have an appointment but I’m not sure when.”
- Background noise, slow talkers, fast talkers, interruptions

Each failure produced a new guardrail.

---

# LLM Selection

I compared multiple LLMs:

- GPT-4o
- Mistral Large
- Llama 3.1 405B
- Gemini Flash 2.5 Lite

### Why GPT-4o-mini Cluster Won

As a personal project, cost and latency were critical.

- Fastest latency in real-time telephony
- Lowest cost per call
- Great tool-calling accuracy
- High conversational quality

More expensive models like GPT-4.1 would improve context handling but cost significantly more.

---

# Challenges Solved

## 1. Over-Answering & Long Responses

Solution:

- Global constraint: only short summaries, never long lists
- Knowledge-base compression rules

## 2. Incorrect Tool Timing

Solution:

- Mandatory pre-tool filler phrases
- Strict tool-boundary prompting

## 3. Missing or Duplicated Patient Intake

Solution:

- Intake only when medically or appointment-related
- Saved context variables

## 4. Appointment Booking Conflicts

Solution:

- Always re-check availability
- Clinic-hour constraints

### 5. CRM Data Inconsistency

Solution:

- All emails + names → lowercase
- Verified spelling before submission

---

# Technical Architecture

### System Stack

- **Platform:** VAPI Voice AI
- **LLM:** GPT-4o-mini cluster
- **STT/TTS:** VAPI native (μ-law 8000 Hz)
- **Telephony:** Twilio
- **Automation:** n8n workflows via MCP server
- **Database:** Google Sheets CRM + Call Log
- **Deployment:** Production phone number + Website demo widget

---

# Performance Metrics

- **Latency:** <800 ms average
- **Agents:** 7 specialized workflows
- **Call Tests:** 100+ synthetic + live calls
- **Context Variables:** 8 consistently preserved
- **Error Rate:** Near zero after guardrails added
- **Appointment Success Rate:** 97%

---

# Key Product Decisions

### 1. Open-Ended Conversation > IVR Menus (Press 1 for…)

Natural user flow improves clarity and user satisfaction.

### 2. Sequential Information Gathering

Never overwhelm the caller with multiple questions at once.

### 3. Explicit Negative Instructions

Prevent agents from stepping outside their specialization.

### 4. Paraphrasing Instead of Scripted Responses

Keeps the system sounding human while maintaining accuracy.

---

# Skills & Methods Used

- Multi-agent system architecture
- Prompt engineering
- Voice UX design
- MCP server development
- Telephony engineering
- STT/TTS tuning
- n8n workflow orchestration
- Conversational QA testing
- Knowledge-base grounding
- Technical documentation

---

# The Bottom Line

This project proves that **voice AI is not magic — it's engineering**.

It requires:

- Rigorous testing
- Clean workflow orchestration
- Careful prompt design
- Specialized agents, not monoliths
- Intelligent context tracking

The architecture now works reliably and can be extended into:

- Dental chains
- Medical clinics
- Service businesses
- SaaS demo scheduling
- Lead qualification systems

I’ve built it, tested it, deployed it — and now it’s production-ready.