

# Evernect Portfolio Documentation

## Project: Local Booster Voice AI Assistant

---

### Overview

Client: Local Booster — an AI-powered advertising platform based in Poland

Deliverable: A fully automated inbound call handling system using a Voice AI agent integrated with CRM and calendar booking infrastructure.

---

### Problem Statement

Local Booster was running an active inbound call channel but had no automation in place to handle it. Every call required a human to pick up, qualify the caller, answer product questions, or schedule demo meetings. This created three core problems:

- Calls outside working hours went unanswered entirely, losing warm leads at the moment of peak interest
- Sales reps were spending time on repetitive FAQ-style questions from existing clients rather than focusing on high-value conversations
- There was no consistent, reliable process for collecting lead data and booking demos — it depended entirely on individual rep performance

The client had three distinct products (Local Booster, Strona, Asystemt) and needed the assistant to handle inquiries across all three intelligently, not just one.

---

### Solution

We built a two-layer system: a VAPI voice AI agent as the caller-facing interface, and an n8n automation backend handling all CRM operations, calendar logic, and data flow.

#### Layer 1 — The Voice AI Agent (VAPI)

The agent is named Alex and is built to sound human, not robotic. The system prompt was engineered with a strict speaking style guide — short sentences, no stacked questions, natural contractions, no mention of internal systems to the caller. This was a deliberate design choice: callers should feel like they're speaking to a knowledgeable team member, not an IVR system.

Alex operates across two main flows:

Flow A (Existing Clients / Support): Alex greets the caller, detects support intent, pulls from three separate knowledge bases (Local\_Booster\_KB\_1, Asystemt\_Local\_Booster\_kb,

Strona\_Local\_Booster\_kb) called once at startup, answers FAQ questions from stored context, and escalates to a human ticket only when genuinely needed. This prevents unnecessary tool calls mid-conversation and keeps responses fast and natural.

Flow B (New Leads / Sales): Alex gives a brief product overview across all three Local Booster products, then qualifies the lead with product-specific question sets — different questions depending on whether the caller is interested in Local Booster (ad platform), Strona (website product), or Asystent (AI receptionist). This personalization was intentional; generic qualification doesn't convert. After qualification, Alex transitions to demo booking directly inside the call.

The booking flow is carefully sequenced: Alex first checks real-time availability, presents up to three slots verbally (in plain spoken language — "two thirty PM" not "14:30"), collects name and company name, then fires the booking tool only once all details are confirmed. Slot filtering logic (for "something later" or "after 2pm" requests) runs entirely in memory — no additional API calls.

## Layer 2 — n8n Automation Backend

The backend consists of five distinct workflow segments:

Call Initiation & CRM Lookup: When a call starts, a webhook (`get-local-booster1`) receives the caller's phone number from VAPI. An HTTP request fires to GHSL's contacts search API. If the contact exists, their ID is passed to the VAPI assistant as a variable override (`contact_id`). If not, a new GHSL contact profile is automatically created and that new ID is passed instead. This means every caller is in CRM from the first second of the call, regardless of whether they've ever called before.

Availability Check — Preferred Date (`availability-check-ai`): When a caller names a specific date, the AI calls this webhook. A JavaScript code node parses the tool arguments, maps the project name to the correct GHSL calendar ID (three separate calendars for the three products), computes a 24-hour timestamp window, and queries GHSL's free-slots API. The response is validated — if slots exist they're returned to VAPI; if the array is empty, a "no slots" message is returned; if the API call itself fails, a technical difficulty message is returned. Every path is handled.

Availability Check — Now (`availability-check-now-ai`): Identical logic but defaults to today's date if no preferred date is provided by the AI. Falls back to `$now` at runtime so the date is always accurate.

Appointment Booking (`book-local-booster`): Triggered once the caller confirms a slot. A JS code node extracts the confirmed slotDateTime and maps to the correct calendar. A POST fires to GHSL's appointments API with calendarId, locationId, contactId, and startTime. Success returns a confirmation message to VAPI; failure returns an error message so Alex can handle it gracefully without silence or confusion.

All three calendar systems (local\_booster, strona, asystem) are managed through a single unified n8n workflow using a calendar mapping object — clean, maintainable, and easy to extend.

---

## Tech Stack

| Component                  | Tool                           |
|----------------------------|--------------------------------|
| Voice AI Agent             | VAPI                           |
| Automation / Orchestration | n8n                            |
| CRM & Calendar             | GoHighLevel (GHL)              |
| Knowledge Base             | VAPI Knowledge Base Tools (x3) |
| Contact Search & Creation  | GHL Contacts API               |
| Appointment Booking        | GHL Calendars API              |
| Scripting                  | JavaScript (n8n Code nodes)    |

---

## Key Design Decisions Worth Highlighting

**Single KB call at startup:** The knowledge base is called once at the very beginning of the call and stored in the agent's context. The system prompt explicitly instructs the agent not to call it again. This prevents latency spikes mid-conversation and makes the agent feel faster and more natural.

**Product-specific qualification:** Rather than a generic lead form, the qualification questions branch depending on which product the caller is interested in. A caller asking about website design (Strona) gets different questions than one asking about ad management (Local Booster) or call automation (Asystem).

**Memory-based slot filtering:** When a caller says "something later" or "after 3pm," the agent filters the already-stored slot list in memory instead of making another API call. This keeps the conversation flowing without unnecessary latency.

**Automatic contact creation:** No caller falls through the cracks. If they're not in CRM, they get created automatically at call start — before any conversation even happens. This means even if the call drops, the phone number is already logged.

---

## Results

- Inbound calls handled 24/7 with zero human required for standard flows

- New callers automatically profiled in GHL from call initiation
- Demo bookings completed live during the call with real-time calendar sync across three separate product calendars
- Existing client support resolved via FAQ with clean human escalation for edge cases
- Consistent lead data collection (name, company) on every new lead call
- Full error handling on every API path — no silent failures exposed to the caller

Documentation prepared by Evernect. For the web presence projects ([www.localbooster.pl](http://www.localbooster.pl), [strona.localbooster.pl](http://strona.localbooster.pl), [asystent.localbooster.pl](http://asystent.localbooster.pl)), the respective project lead should complete that section before Monday EOD.

