

Pipeline 1 — Jarvis (stem-AI-assistent)

Een herbruikbaar recept om voor een bedrijf een **pratende AI-assistent** te bouwen die je via een website (stem + camera) of via een telefoongesprek bedient — en die ook echt dingen kan aansturen (verlichting, rolluiken, apparaten). Dit is hoe we Jarvis voor Haarvisie hebben gebouwd, inclusief de problemen waar we tegenaan liepen en hoe we ze oplosten. Bedoeld om opnieuw te bouwen en aan te bieden.

1. Doel

- **Wat:** een AI waar je gewoon tegen práát, die antwoordt met een natuurlijke stem, meekijkt via de camera, en op commando je smart home bedient.
 - **Voor wie:** een ondernemer die handsfree wil werken (in de auto, in de salon) en geen losse apps wil.
 - **Klaar als:** je zegt "doe de rolluiken dicht" en het gebeurt + Jarvis bevestigt kort.
-

2. Bouwstenen (de stack)

- **Brein + stem:** Google **Gemini Live** (native-audio model) — realtime, onderbreekbaar, NL-stem.
- **Twee ingangen:**
 1. **Website** (`jarvis.haartips.nl`): de browser praat via een dunne relay met Gemini Live.
 2. **Bellen** (`voice.haartips.nl`): WhatsApp-belt het nummer → een brug verbindt het gesprek met Gemini.
- **Server:** een kleine VPS (huur-server, altijd aan, vast IP — nodig om van buitenaf bereikbaar te zijn).

- **Aansturing:** function-calling (Gemini roept een "functie" aan) → die functie bedient het dashboard/IFTTT.
-

3. Stap-voor-stap

A. De website-assistent (de makkelijkste ingang)

1. Eén statische HTML-pagina die de microfoon opneemt + de camera kan tonen.
2. Een **transparante relay** (klein scriptje) tussen de browser en Gemini Live: de browser stuurt de instellingen + audio, de relay voegt de geheime sleutel toe en pijpt alles door. De browser bepaalt dus zelf de "persoonlijkheid" (systeemtekst) en de stem.
3. Een aparte **tekst/foto-backend** voor losse vragen + camerabeeld-analyse.
4. **Bediening toevoegen:** in de instellingen die de browser verstuurt, declareer je een functie (bv. `bedien_rollyuk`). Komt die functie binnen → de pagina roept een klein server-endpoint aan dat het dashboard bedient, en stuurt het resultaat terug naar Gemini.

B. De bel-assistent (voor handsfree/CarPlay)

1. Gebruik de **WhatsApp Business Calling API** (gratis, werkt via CarPlay) — geen Twilio nodig.
2. Een **brug-server** vangt het belletje op, zet een directe audioverbinding op (WebRTC) en koppelt de audio aan Gemini Live (in beide richtingen).
3. Zelfde function-calling als de website → dezelfde acties via stem.

C. Veilig opleveren

- Altijd een **backup** van de werkende versie vóór elke wijziging.
- **Nooit herstarten tijdens een actief gesprek** (check eerst dat er 0 gesprekken lopen).
- Testen met een testnummer voordat je het echte nummer aanzet.

4. Valkuilen + oplossingen (hier zit de echte waarde)

- **Verkeerde modelnaam.** Het juiste Gemini-Live-model heeft een specifieke naam; de "Vertex"-naam bestaat niet op de gewone API → leeg antwoord. Oplossing: de exacte `native-audio-preview` -naam gebruiken.
- **Audio kwam niet door (bellen).** Drie fixes nodig: (1) de verbindings-handshake had te veel "vingerafdrukken" → terugbrengen naar één; (2) de ontvang-lus stopte na de begroeting → in een buitenlus zetten; (3) de binnenkomende audio bevatte opvul-bytes → alleen de echte geluids-bytes doorsturen. Pas daarna was het gesprek tweerichting verstaanbaar.
- **Microfoon in de controlekamer geblokkeerd.** Als je de assistent in een ingebed venster (iframe) toont, blokkeert de browser standaard de microfoon → de gebruiker kan nergens "toestaan" geven. Oplossing: het venster expliciet toestemming geven (`allow="microphone; camera"`).
- **Stem-keuze.** De vloeiendste NL-stem (Sulafat) bleek dezelfde als onze verslag-stem; instelbaar per ingang. Tip: laat de klant kiezen uit een paar samples.
- **Transcriptie nog niet perfect.** De spraak-naar-tekst hapert soms; te verbeteren met een gespecialiseerde NL-spraakmodule in de pijplijn (oren → brein → stem).

5. Hergebruik / opschalen

Voor een nieuwe klant/salon:

1. Eigen subdomein + VPS-plekje.
2. Persoonlijkheid (systeemtekst) aanpassen op het bedrijf — 1 alinea.
3. Twee standen: **persoonlijk** (eigenaar, mag alles bedienen) en **klant** (alleen bedrijfsvragen, GEEN apparaatbediening) — belangrijk voor veiligheid.
4. Welke acties mag de assistent doen? Per klant de functies kiezen (licht, rolluik, afspraak maken...).
5. Optioneel het belnummer aanzetten voor handsfree.

Verkoopbaar als: "Een AI-assistent met je eigen stem en naam, die je bedrijf kent en je apparaten bedient — via je website of gewoon door te bellen."

Eerste van een reeks pipelines. Volgende: het Licht- & Rollluik-dashboard (smart home in 1 controlekamer).