MQTT + Ollama =

Building Home Automation That Actually Works (And Doesn't Spy on You)

About me

- From Columbus, OH (Victorian Village)
- Professional Developer since 04’
- Enjoy the perks of a private airline
- Find me at Comfest or DooDah Parade
- Support local dev community
  - COJUG, CBus AI Week, StirTrek, etc
- Let’s empower people through IoT and LLMs

What we will do

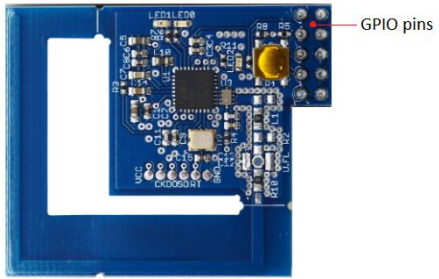
- Build a functional, Alexa-like private voice assistant
- Make it so that it can be used offline
- Connect tools to the LLM to let it interact with the world.
- Control smart devices (over mqtt, or directly via ZWave) using your voice assistant

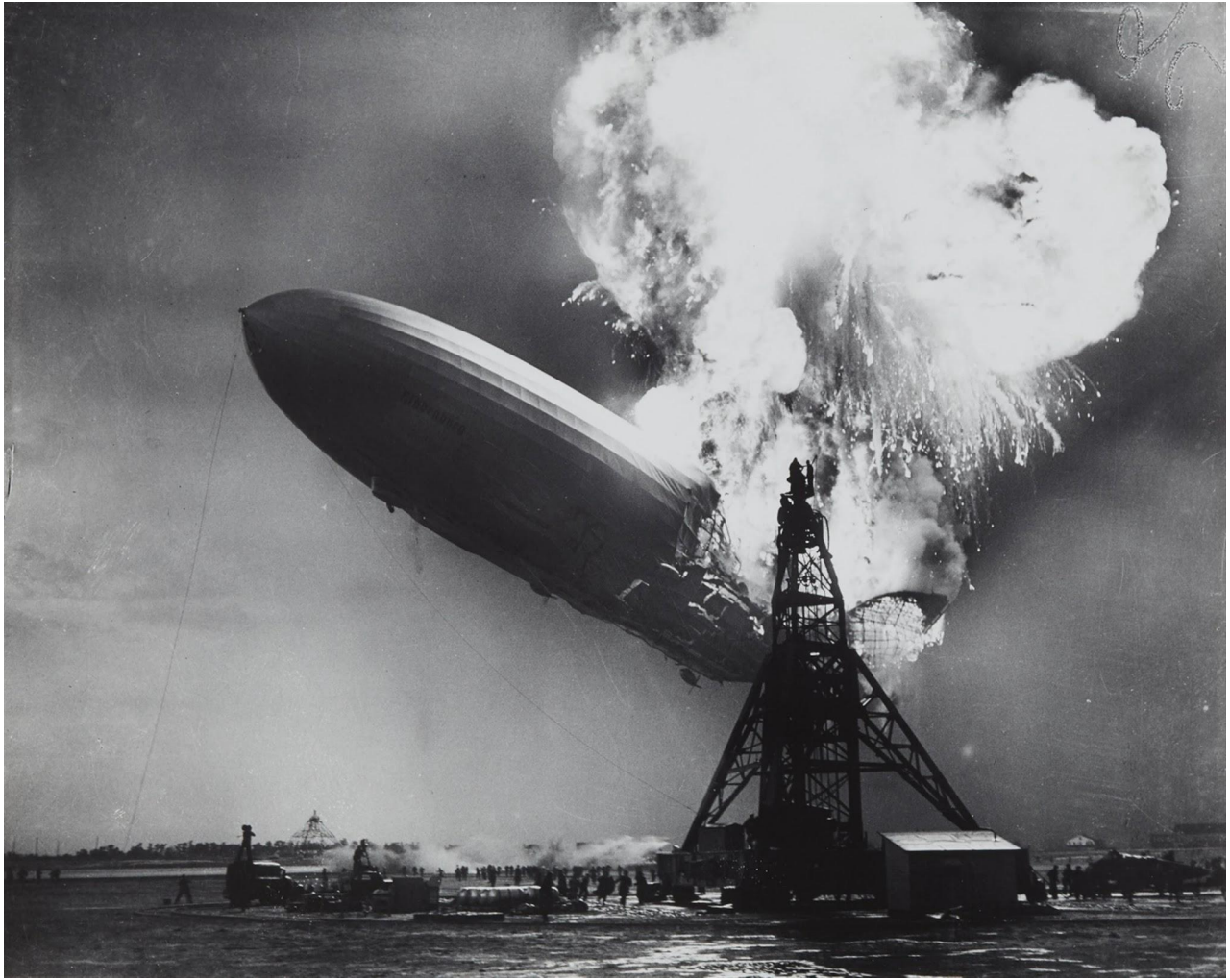
What do we need?

Raspberry Pi 5
16gb ($99.00)

USB Speakers And Microphone
$14.99 $5

ZPi-7
$54.99



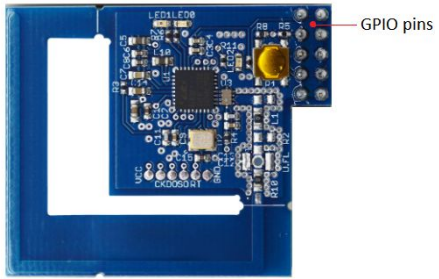


Uh Oh!!!!!

Raspberry Pi 5
16gb ($99.00)

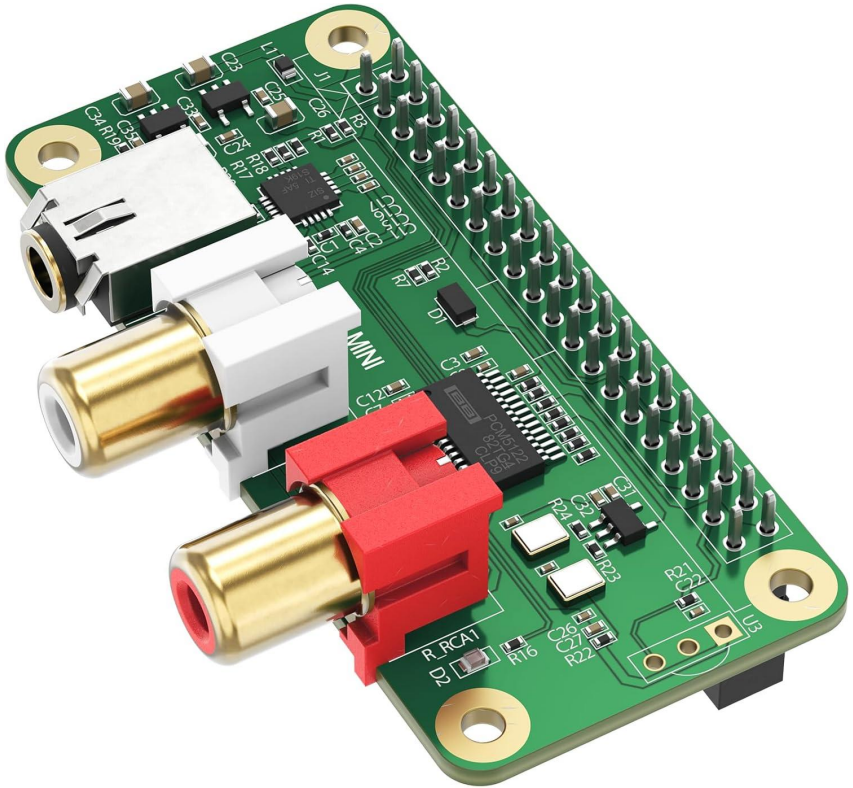
~~USB Speakers~~ And ~~Microphone~~
~~$14.99~~ $5

ZPi-7
$54.99



So I added this

DAC Mini Hat



Home Assistant

THE ERA OF OPEN **VOICE**

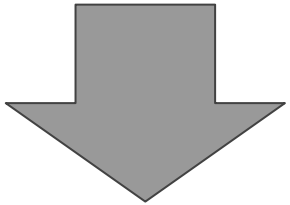
We don’t need no stinkin easy way

Compared to

Amazon Alexa
$50
(No ZWave)

The “Easy” Way

Home Assistant



AWS Lambda

Limitations

- The HA integration supports limited functionality
- New tooling requires using 3rd party skills causing more data leakage
- Harder to integrate additional context like RAG
- Requires internet connection

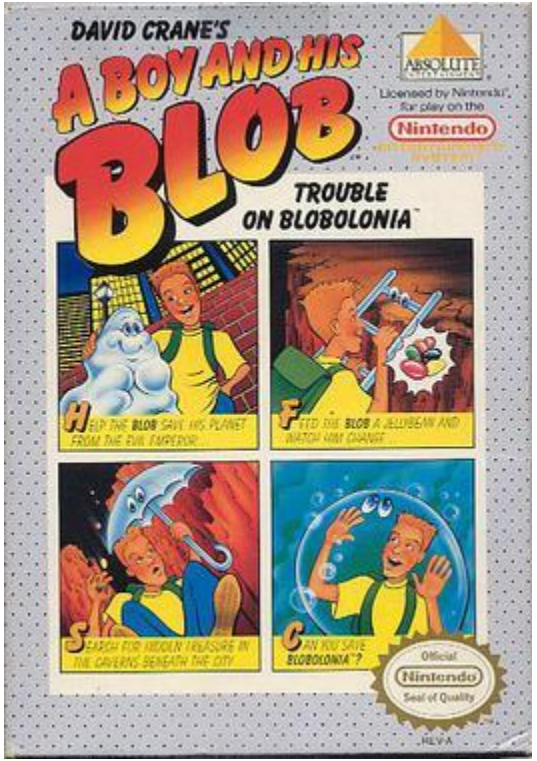
Amazon Alexa Recording Allegations & Lawsuits

- Using children’s data illegally
  - 2:23-cv-00811
  - Amazon agreed to pay a $25 million civil penalty and was required to implement stronger data deletion and privacy safeguards.
- Unintended Recordings/Targeted Ads (In Progress)
  - 2:21-cv-00750
  - A federal judge certified a nationwide class action.
- Unlawful collection of users' biometric voice data (voiceprints) without informed written consent,
  - 1:19-cv-05061 / 1:21-cv-06010
  - A federal judge certified a nationwide class action.

Requirements

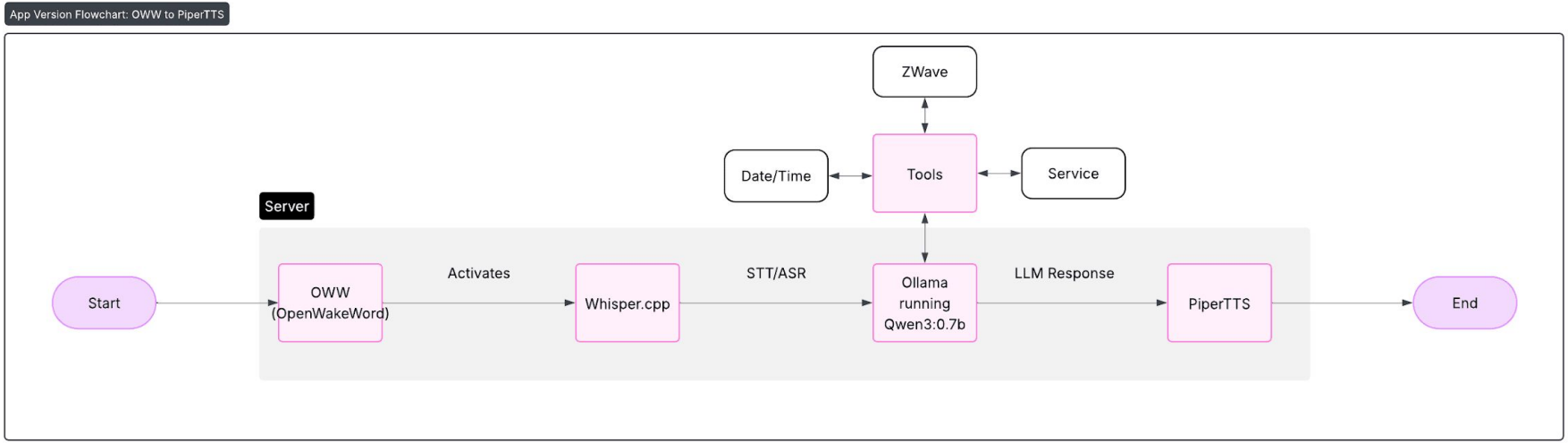
- Work offline
- Extensible
- Easy to acquire parts
- Open Source Licenses

What AI Feels Like

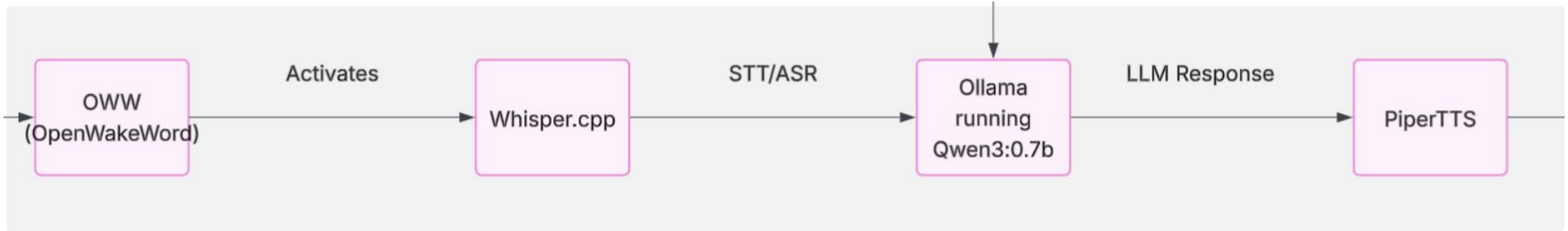



7000
TR=19
VANILLA = UMBRELLA

Design Diagram



Workflow


OWW
(OpenWakeWord)
Activates
Whisper.cpp
STT/ASR
Ollama
running
Qwen3:0.7b
LLM Response
PiperTTS

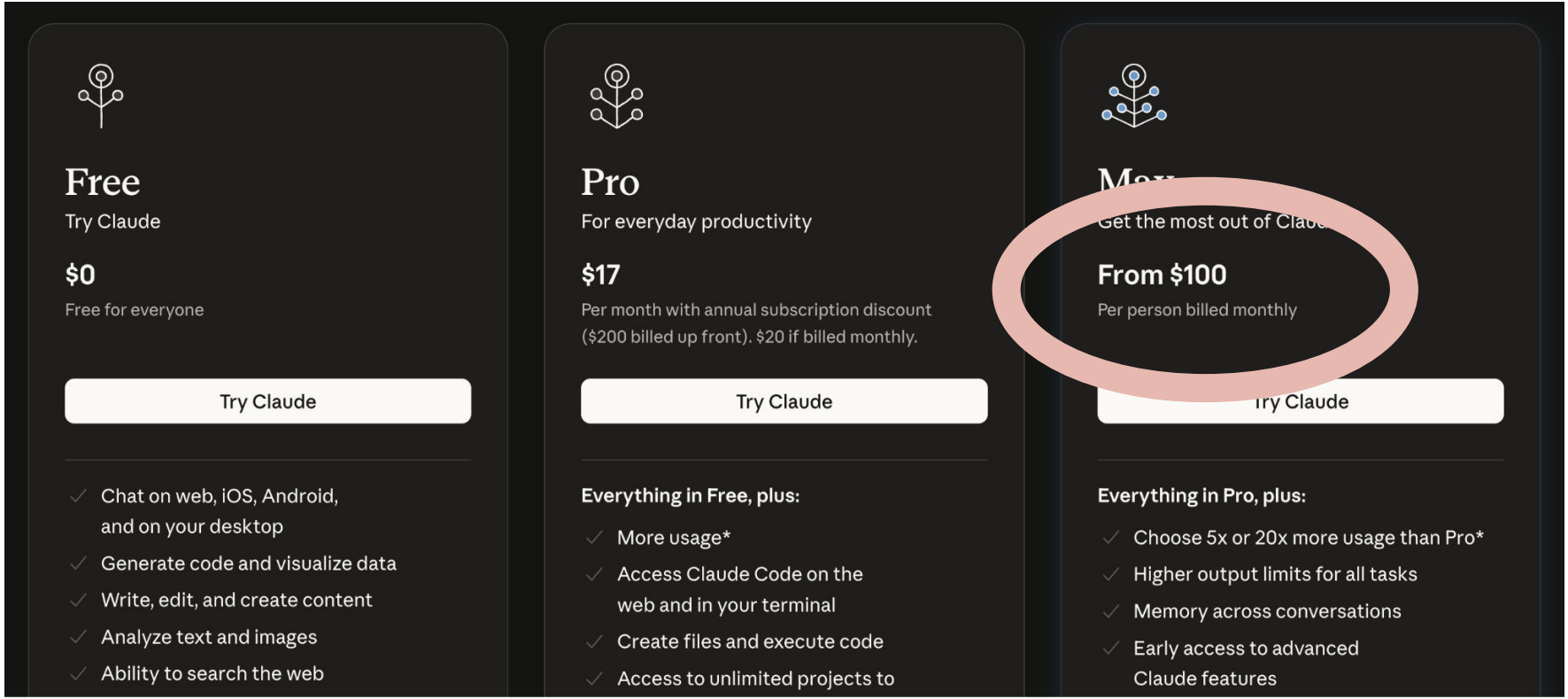
Foundational Models

- **A large AI model trained on broad, diverse datasets** — often using self-supervised learning.
- **Adaptable to many downstream tasks** through fine-tuning or prompting (e.g., translation, classification, summarization).
- It serves as a **general base for building specialized applications**, rather than being trained for one narrow task.

A foundation model is an AI model trained on broad data at scale such that it can be adapted to a wide range of downstream tasks.

Wikipedia

Good but expensive



Secret Sauce

Why Qwen?

- ✅ **Very small parameter options** (e.g., 0.5B–1.8B) → practical on **Raspberry Pi 5**
- ✅ Lower **RAM + VRAM footprint** than most open LLMs
- ✅ Runs reasonably with **CPU-only inference**
- ✅ Performs well with **quantization (INT8 / INT4)**
- ✅ Strong **reasoning + instruction following** for its size
- ✅ Competitive performance vs larger models (LLaMA, Mistral) at similar parameter counts
- ✅ Works well with lightweight runtimes (llama.cpp, vLLM variants)
- ✅ Actively maintained by Alibaba
- ✅ Clear positioning as a **foundation model family**
- ✅ Friendly for **RAG, agent, and IoT workflows**
- ✅ Multiple sizes → easy to scale up/down
- ✅ Open weights → supports **fine-tuning, LoRA, edge customization**
- ✅ Good **multilingual support** (English + Chinese especially)

Limitations

- ❌ Smaller models **cannot match GPT-4 / Claude / large LLaMA** on complex reasoning
- ❌ Limited long-context depth compared to large proprietary models
- ❌ Heavily quantized models may lose nuance
- ❌ Smaller community than LLaMA
- ❌ Fewer prebuilt domain-specific fine-tunes
- ❌ Best suited for **task-focused or agent-based usage**, not free-form chat at scale

How to listen

- “Wake word”
- TTS
- AI/LLM
- STT

Pros

- Simple and quick to get working
- Easy to install locally
- Free (as in beer) to use and modify
- Robust community support
- No chance to spy
- Works completely offline

Cons

- Slow/Laggy
- Must use less robust models on most devices
- No support
- Voices either require training or are lame

OpenSpec

- Creates specification, design and task files
- Allows for multiple agents to work on tasks at the same time
- Ensures that test requirements and other important tasks are completed
- Simple markdown design
- Works across CLI tool (Q/Kiro, OpenCode, Codex)


There be AI generated graphics ahead

OpenWakeWord Explained

- OpenWakeWord is free and open-source software
- It detects custom wake words entirely offline
- Uses deep learning to recognize specific audio cues
- High-level functionality involves continuous audio processing

Training New Wake Words

- OpenWakeWord allows training custom wake words
- New wake words are trained using audio data
- The training process utilizes deep learning models
- This enables offline detection of unique phrases

Whisper.cpp Explained

- Whisper.cpp is a high-performance STT (Speach2Text) engine.
- It is optimized for faster processing on local hardware.
- Provides accurate and reliable speech-to-text conversion.
- It runs entirely offline for maximum privacy.
- Whisper.cpp is a C/C++ port of OpenAI's Whisper model and is available under the MIT License.

LLM Tools Explained

- Tools allow the Large Language Model to interact with the external world.
- LLM first determines the intent and necessary parameters for a tool.
- The model then executes the selected tool with the determined arguments.
- The tool's output is returned to the model for the final response generation.

Example

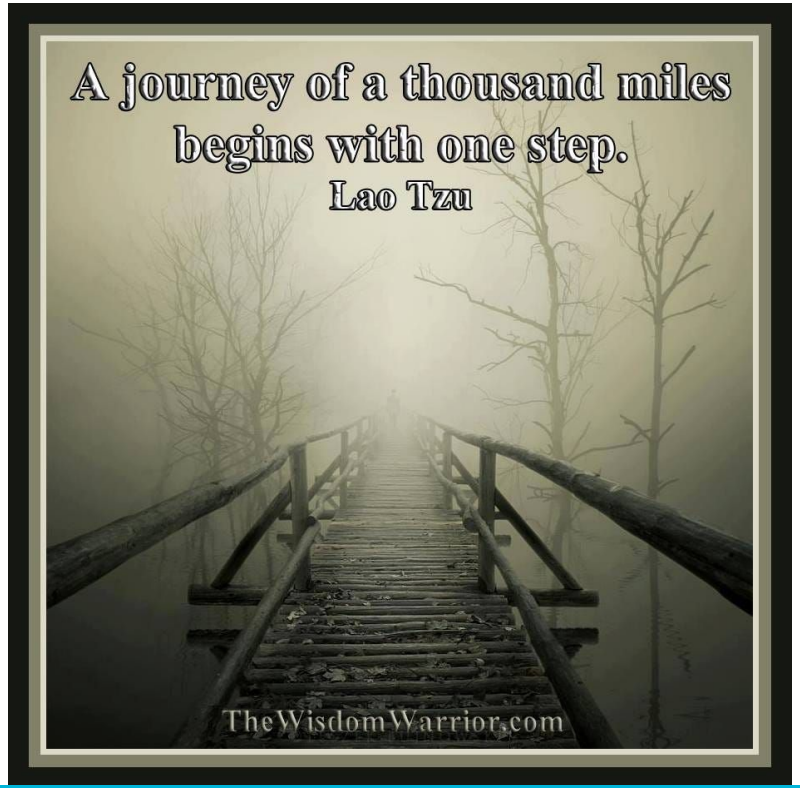
```
import * as z from "zod"
import { ChatOpenAI } from "@langchain/openai"
import { createAgent } from "langchain"

const getUserName = tool(
  (_, config) => {
    return config.context.user_name
  },
  {
    name: "get_user_name",
    description: "Get the user's name.",
    schema: z.object({}),
  }
);
```

```
const getWeather = tool(
  ({ city }, config) => {
    const writer = config.streamWriter;

    // Stream custom updates as the tool executes
    writer(`Looking up data for city: ${city}`);
    writer(`Acquired data for city: ${city}`);

    return `It's always sunny in ${city}!`;
  },
  {
    name: "get_weather",
    description: "Get weather for a given city.",
    schema: z.object({
      city: z.string(),
    }),
  }
);
```

```
const agent = createAgent({
  model: new ChatOpenAI({ model: "gpt-4o" }),
  tools: [getUserName],
  contextSchema,
});
```

Demo

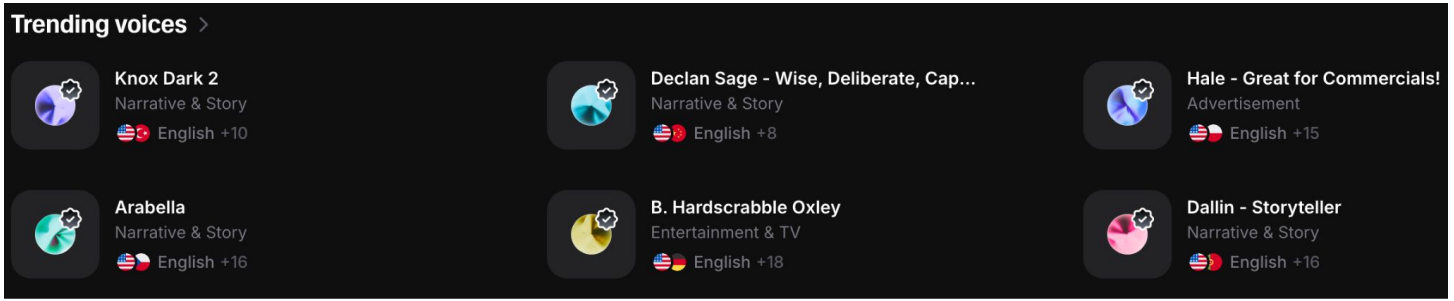
Good First Step

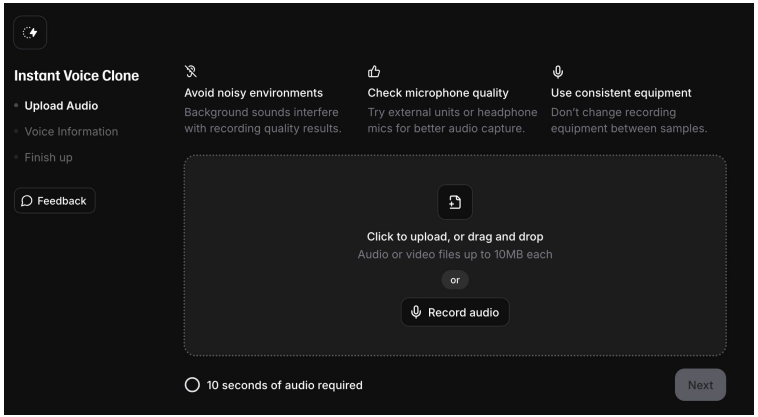


11ElevenLabs

- Requires internet
- Allows you to select multiple voices including multiple languages
- Faster voice training if you want to use your own voice.
- Also handles STT but Whisper seems to work fine.
- Relatively low cost

Selecting A Voice





Using ElevenLabs

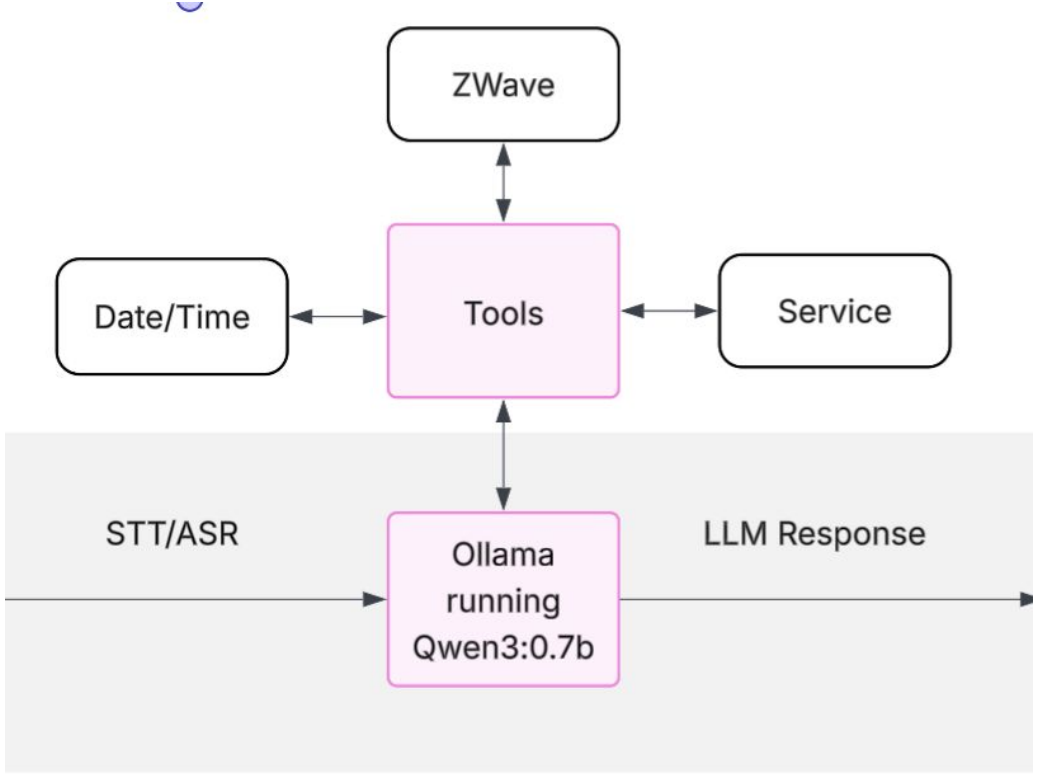
```
elevenlabs: {
    apiKey: process.env.ELEVENLABS_API_KEY,
    voiceId: process.env.ELEVENLABS_VOICE_ID || '2i0Vtk39FYVTw6Tx1mC9', // Default: George (deep, authoritative male)
    modelId: process.env.ELEVENLABS_MODEL_ID || 'eleven_v3',
    stability: process.env.ELEVENLABS_STABILITY ? Number(process.env.ELEVENLABS_STABILITY) : 0.5,
    similarityBoost: process.env.ELEVENLABS_SIMILARITY_BOOST ? Number(process.env.ELEVENLABS_SIMILARITY_BOOST) : 0.75,
    style: process.env.ELEVENLABS_STYLE ? Number(process.env.ELEVENLABS_STYLE) : 0.0,
    useSpeakerBoost: process.env.ELEVENLABS_USE_SPEAKER_BOOST !== 'false',
},
```

```
// Use streaming API for lower latency
const audioStream : ReadableStream<Uint8Array> = await client.textToSpeech.stream(config.elevenlabs.voiceId, request: {
    text: speechText,
    model_id: config.elevenlabs.modelId,
    output_format: 'mp3_44100_128', // High-quality MP3
    voice_settings: {
        stability: config.elevenlabs.stability || 0.5,
        similarity_boost: config.elevenlabs.similarityBoost || 0.75,
        style: config.elevenlabs.style || 0.0,
        use_speaker_boost: config.elevenlabs.useSpeakerBoost || true,
    },
});
```

LLM Tooling Overview

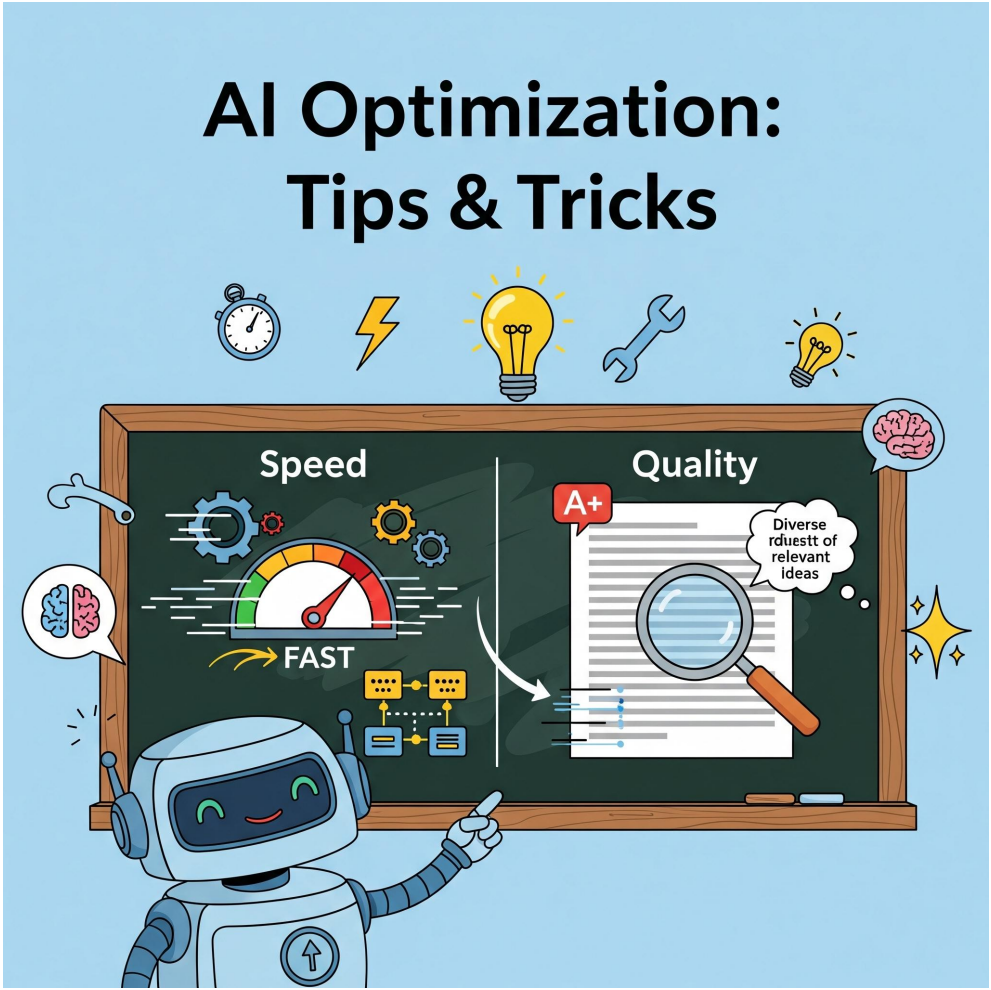
- Tools extend the LLM's capability to interact externally.
- They allow the AI to perform actions in the physical world.
- Tools translate natural language intent into executable code.
- This functionality enables integration with smart home devices.

Tools


ZWave
Date/Time
Tools
Service
STT/ASR
Ollama
running
Qwen3:0.7b
LLM Response

Quality vs Speed

- Use /nothink parameter to speed up responses
- Increase parameters context to increase answer quality
- Important Balance



Options for Device Control

ZWave

- Point 2 Point
- S2 allows for encrypt
- 908 Mhz
- Great for low power

ZigBee

- Daisy Chain
- Great for large areas
- 2.4 Ghz
- Great for dense networks

WiFi

- Works with most devices
- High Bandwidth
- 2.4Ghz
- May use proprietary protocol

ESP32

- Best when device doesn’t have others.
- Provides Bluetooth, WiFi, etc

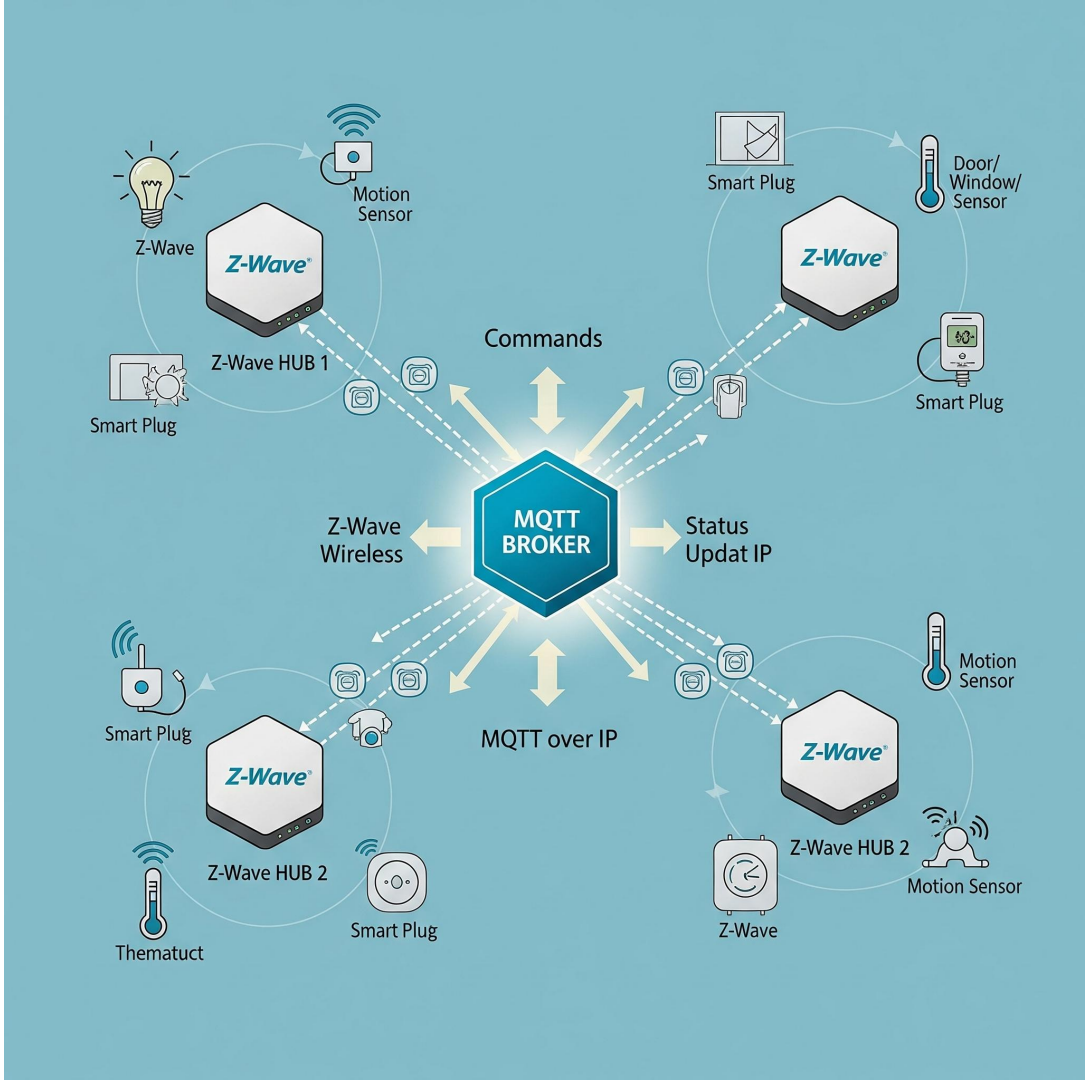
ZWave Device we are using

[ZWave S2 Outdoor Plug]

$31

ZWave-JS-UI for Control

- ZWave-JS-UI connects Z-Wave devices via MQTT.
- It translates Z-Wave commands into MQTT messages.
- Use ZWave-JS-UI for device control through an MQTT server.
- The interface manages devices by reading and writing MQTT data.



Why MQTT?

- MQTT is a lightweight messaging protocol perfect for IoT.
- It offers reliable, bi-directional communication between systems.
- The publish/subscribe model decouples devices and services effectively.
- Low bandwidth and minimal overhead are ideal for constrained devices.

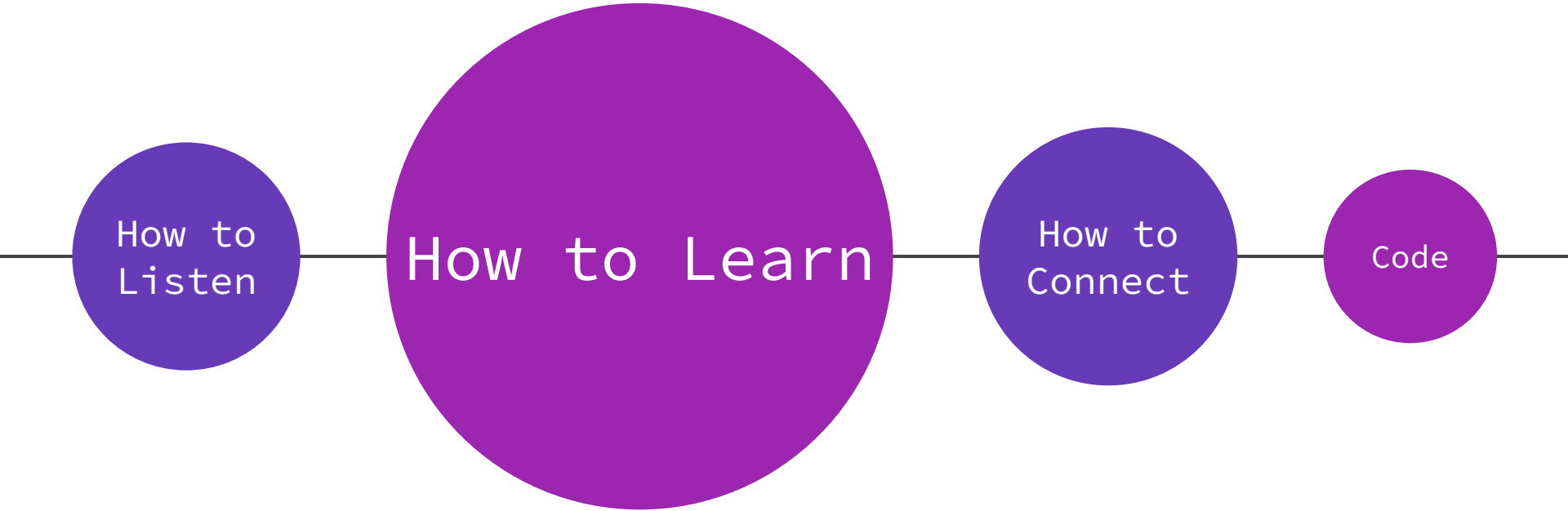
LLM to MQTT Control

- The LLM tool is configured to connect to the MQTT server.
- The model translates your commands into MQTT messages.
- This allows the AI to control smart home devices directly.
- The tool sends control messages to manage devices.

AI and Meshtastic

- MQTT allows AI interaction with Meshtastic networks.
- Meshtastic communications use MQTT as a transport layer.
- AI can send and receive messages over Meshtastic via MQTT.
- This creates a powerful bridge for automation and control.

What we will learn



Questions?

Hope you enjoyed it!

Feedback

MQTT + Ollama = Building Home Automation That Actually Works
(And Doesn't Spy on You)