



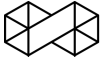
# Home Automation

Context and History of Home Assistant and ESPHome

---

**RIAT Bootcamp**

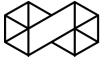
A case study in open-source governance, sustainability, and local-first design.



# What is Home Assistant?

A **free, open-source** smart home controller and integration platform. **One point of control** for devices from **any manufacturer**, running **locally on your hardware** without cloud services.

Web browser, Android, or iOS. Voice commands via Google Assistant, Amazon Alexa, Apple Siri, or Home Assistant's own local "**Assist**" assistant.



# What is ESPHome?

A **free, open-source** system to control your **ESP32, ESP8266, BK72xx, RP2040, and other microcontrollers** with simple, powerful configuration files — and to control them remotely through home automation systems.

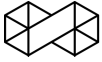
Firmware is generated, compiled, and flashed from a single **YAML file**. **Over-the-air updates** are built in. **No IDE, no build tools, no opaque binary.**



# Act I

A SCRIPT THAT TURNED ON SOME LIGHTS

---



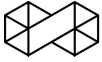
# Act I — The founding

On **September 17, 2013**, **Paulus Schoutsen**, a Dutch software engineer, started a personal project to interface with Philips Hue lights. He published it on GitHub in November 2013 as **Home Assistant**. The Apache 2.0 license was chosen from the start <sup>1</sup>.

2013 is the moment the Internet of Things (IoT) became a marketing category. Every sensor and thermostat was acquiring a proprietary cloud account.

Schoutsen's founding principle — ***local control and privacy first*** — was not a business decision. It was an **architectural response to a threat model**:

| *What happens to your home automation when the vendor shuts down, gets acquired, or changes its API?*



# Act I — Empirical answers

That question has been answered multiple times:

- **Revolv** — acquired by Nest (Google) in October 2014 for an undisclosed sum. Sold for \$299 with a "lifetime subscription." Nest shut down the cloud on **May 15, 2016**. All hardware permanently inoperable. The U.S. Federal Trade Commission (FTC) investigated but chose not to pursue enforcement <sup>2 3</sup>.
- **Insteon** — shut down servers abruptly on **April 15, 2022** without notice, while the status page still read "All Services Online." The HA integration page now carries a permanent warning <sup>4</sup>.
- **Wink** — with one week's notice in May 2020, imposed a **\$4.99/month subscription** on hardware sold with "no monthly fees" printed on the box <sup>5</sup>.



# Act II

HASS.IO AND THE USABILITY PROBLEM

---



# Act II — Solving installation

Home Assistant grew rapidly as a do-it-yourself (DIY) project, but had a real barrier: running it required Linux administration skills.

In **July 2017**, the team released **Hass.io** — a managed operating system (OS) image for single-board computers (Raspberry Pi) — reducing setup time to minutes<sup>1</sup>.

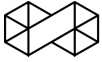
*This is a recurring pattern in open source: software gets built by those who can compile from source, and the community scales when someone solves the installation problem. Hass.io was that solution.*



# Act III

A STUDENT BUILDS THE MISSING PIECE

---



# Act III — Otto Winter

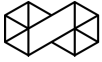
**Otto Winter** is an Austrian computer science student at the **Technische Universität Wien**.

- Competed in the Austrian Olympiad in Informatics from 2015 onwards
- Qualified for the **International Olympiad in Informatics in Tsukuba, Japan (2018)** <sup>6</sup>
- Around that time, created what would become **ESPHome**

The project started as two repositories:

- `esphomelib` — a C++ framework for ESP microcontrollers
- `esphomeyaml` — a Python tool to generate C++ from YAML

With **version 1.10.0**, both were merged and rebranded as **ESPHome** <sup>7</sup>. The YAML interface had become the primary way to use the project; the original names were "too technical."



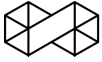
# Act III — The problem ESPHome solved

ESP32, ESP8266, BK72xx, RP2040, and others are inexpensive (**under €5 per chip**), Wi-Fi capable, and programmable. But using them required:

- Writing C++ code from scratch
- Setting up the build tools yourself
- Wiring up over-the-air (OTA) update logic yourself

ESPHome replaced that. You describe the sensor, the pin, the name in a text file — ESPHome turns that into a working program and installs it on the device.

*The result: no IDE, no build tools, no closed firmware from the manufacturer.*



# The declarative paradigm

```
sensor:  
  - platform: dht  
    pin: D2  
    temperature:  
      name: "Living room temperature"  
    humidity:  
      name: "Living room humidity"
```

ESPHome reads this file, converts it into the program that runs on the chip, and installs it on the device. Over-the-air updates (pushing new code over Wi-Fi) are built in.

*Mental model: describe what you want, not how the machine should do it. The declarative layer is the contribution.*



# The reproducibility dimension

The YAML file is the **complete, human-readable specification** of everything running on the device.

Given the same YAML and the same ESPHome version, you can:

- **Reproduce** the firmware
- **Audit** it
- **Diff** it
- **Version-control** it

This is the structural opposite of consumer IoT firmware, distributed as signed binaries with no published source and silently updated.

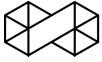
*For work on verifiable and reproducible systems, this is not a minor point — it is the core value proposition stated differently.*



# Act IV

THE COMMERCIAL QUESTION

---



# Act IV — Nabu Casa, Inc.

**Nabu Casa, Inc.** was founded in **2018** in Irvine, California, by **Paulus Schoutsen** and **Pascal Vizeli** — two of the principal founders of Home Assistant <sup>8</sup>.

Stated motivation: fund ongoing development through a cloud service that put privacy first — not through data brokerage or advertising.

The model:

- Home Assistant core remains **Apache 2.0** — free, open, permissive
- Nabu Casa sells one service: **Home Assistant Cloud** — remote access, encrypted voice relay, off-site backups <sup>9</sup>
- No investors, no advertising, no data monetisation — funded entirely by user subscriptions <sup>8</sup>
- Nabu Casa cannot retroactively relicense the existing open source code

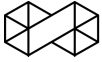
*This is the "open core plus services" model — the project stays free and open-source, a single paid service funds further development.*



# Act V

AN ACQUISITION TO PRESERVE, NOT TO MONETIZE

---



# Act V — The acquisition

In **March 2021**, Nabu Casa acquired ESPHome from Otto Winter<sup>10</sup>.

Stated reason: **continuity**. ESPHome had become critical infrastructure for the Home Assistant ecosystem, and a solo student maintainer is a single point of failure.

Winter had started the project roughly three years earlier. The acquisition ensured it would outlast any single person's available time.

Conditions of the sale:

- ESPHome remained **free and open source**
- Winter **stayed on** as a contributor
- **No revenue stream was gained**, no user base was monetised
- Only motivation stated: **preventing abandonment**



# Act VI

GOVERNANCE AS A LEGAL CONSTRAINT

---



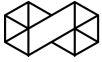
# Act VI — The Open Home Foundation

In **April 2024**, Home Assistant, ESPHome, and 250+ related projects were transferred to the **Open Home Foundation (OHF)**, a non-profit <sup>11</sup>.

Apache 2.0 alone could not prevent a new owner from closing APIs, adding telemetry, or deprecating local control. The Foundation is the structural firewall against that.

Projects under OHF include Music Assistant, Zigbee2MQTT, Z-Wave JS, Rhasspy, WLED, and Piper.

*A non-profit structure is the only governance model that makes "it can never be bought, it can never be sold" a legal reality rather than a promise <sup>12</sup>.*

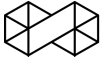


# Act VI — The cloud badge sunset

The OHF also discontinued the "**Works with Home Assistant**" cloud certification badge in August 2024, stating publicly:

| *"All cloud-based products are always doomed to stop working, some sooner than others"* <sup>13</sup>

**This is the threat model, made operational:** the OHF retired a revenue-adjacent program because certifying cloud products would dilute the local-first commitment. The decision is the principle.



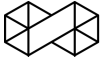
# The reverse engineering dimension

A significant portion of HA's integration library was **not built from published documentation**. It was built by reverse engineering proprietary protocols — analysing network traffic, decoding Zigbee and Bluetooth Low Energy (BLE) frames, reconstructing undisclosed device APIs.

Most manufacturers — Tuya is the canonical example — do not publish local APIs. Cloud routing is the business model. Interoperability has to be reconstructed by the community.

**Concrete example: LocalTuya** — a community integration enabling local control of hundreds of cheap Tuya-based smart home products. Built by community members who reverse engineered Tuya's local network protocol. The project's own documentation traces its code back to contributors who decoded the protocol through traffic analysis <sup>14</sup>.

*The Home Assistant community has functioned, in part, as a distributed reverse engineering apparatus, using the method to create interoperability that manufacturers declined to provide.*



# The open standard question — Matter

Home Assistant has been a vocal advocate for **Matter** — a smart home connectivity standard.

## Timeline:

- December 2019: announced as "Project Connected Home over IP" (CHIP) by **Amazon, Apple, Google, Samsung SmartThings, Zigbee Alliance** <sup>15</sup>
- October 4, 2022: Matter 1.0 published <sup>15</sup>
- Royalty-free, with the software development kit (SDK) open source under Apache License

*The governance tension: Matter's development was led by companies with strong interests in keeping smart home data flowing through their ecosystems. The spec is open; its principal backers each operate competing platforms.*

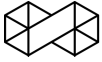


# Scale — Home Assistant by the numbers

- **2 million** active installations (2025) <sup>16</sup>
- **21,000** GitHub contributors in 2024 alone <sup>12</sup>
- **#1** on GitHub Octoverse in 2023 and 2024 <sup>17 12</sup>; **#6** in 2025 as AI projects dominate <sup>18</sup>
- **3,000+** brands supported <sup>19</sup>
- **250+** projects under Open Home Foundation governance <sup>11</sup>
- **Apache 2.0** license <sup>1</sup>

In 2024, HA out-contributed VS Code (Microsoft) and Flutter (Google) — projects with large dedicated engineering teams <sup>19</sup>.

*The scale is the distribution: 2 million independent installations, each running locally, each operator controlling their own data. There is no central fleet management, and the architecture intentionally makes wide-scale centralised deployment difficult.*

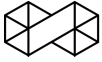


# The stack

## Three languages, three layers:

- **Python** — core runtime, ~3,000 integrations
- **TypeScript** — frontend, runs in the browser
- **C++** — firmware, via ESPHome (YAML generates C++)

One runs on a server. One in a browser. One on a €5 chip.



# Local-first by design

## **Protocols** (no mandatory cloud routing)

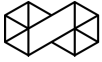
- Zigbee, Z-Wave, Matter, Thread, Wi-Fi, Bluetooth LE — all local-first

## **Voice** (fully offline)

- Whisper for speech recognition, Piper for speech synthesis — both local

## **Governance** (license and structure)

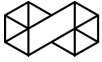
- Apache 2.0
- Open Home Foundation (non-profit steward)



# Discussion hooks

Four questions for this audience.

- 1. Reproducibility and verifiability** — Is the ESPHome build itself reproducible in the strict technical sense, bit-for-bit across build environments? What would full supply-chain verifiability look like for a consumer IoT device?
- 2. Reverse engineering as method** — How should an open source project relate to RE contributions? Is RIAT's framing of RE as a cultural method a useful lens here?
- 3. The sustainability trap** — Otto Winter built critical infrastructure alone, then had to sell it. What institutional models could have prevented that? Does the OHF model solve this, or just delay it?
- 4. Apache 2.0 and the commons** — The permissive license enables Nabu Casa's model but also enables proprietary forks. The GPL copyleft license would prevent the latter but might have prevented the former. Is there a license that better matches OHF's stated values?



# Sources

1. [Home Assistant — Wikipedia](#)
2. [Revolv shutdown \(April 2016\) — The Next Web](#)
3. [Insteon shutdown \(April 2022\) — The Register](#)
4. [Otto Winter — GitHub](#)
5. [Nabu Casa acquires ESPHome \(March 2021\) — HA blog](#)
6. [Open Home Foundation projects](#)
7. [GitHub Universe 2024 wrap-up — HA blog](#)
8. [LocalTuya — community RE integration](#)
9. [State of the Open Home 2025 — HA blog](#)
10. [GitHub — "The local-first rebellion"](#)



# Thank you

---

RIAT Bootcamp