

Smart Swiftlet Farming

An End-to-End IoT Monitoring
& Control Platform

Muhammad Fuad Fakhruzzaki

Project Lead · System Architect · IoT & Backend Engineer

2025-2026



Project at a Glance

Objective:

Monitor and control microclimate conditions inside swiftlet houses.

My Role:

Project leadership, system architecture, IoT and backend development.

Output:

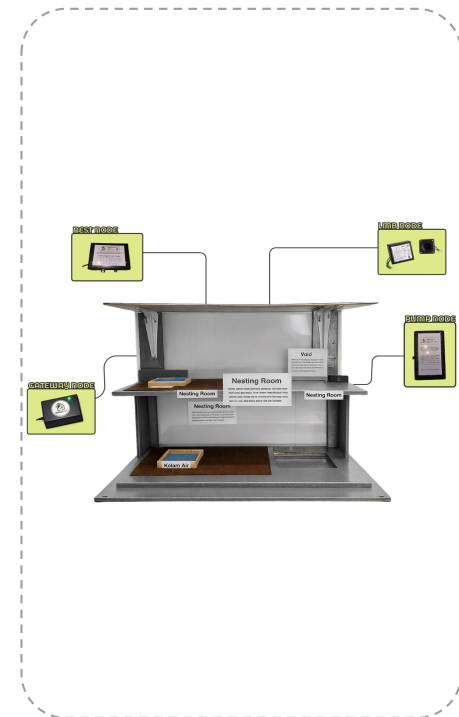
An integrated platform connecting field devices, backend services, automated control, and web monitoring.

Programming Languages I Used

- **C/C++:** ESP32 firmware and embedded control.
- **Go:** backend APIs and MQTT telemetry ingestion.

Core Stack

ESP32 / FreeRTOS / ESP-NOW
MQTT over TLS / Go + Chi
PostgreSQL + TimescaleDB /
Docker



Why This Project Needed to Exist

01



Uneven
Microclimate

Target: 26–29°C · 80–
95% RH

**Observed (upper
floor):** 31–33°C · below
65% RH

02



Existing Humidity &
Audio Setup

Water pools on each
floor supported room
humidity.
Audio was operated
manually.
Humidity control lacked
sensor-based feedback.

03



Limited Monitoring
Visibility

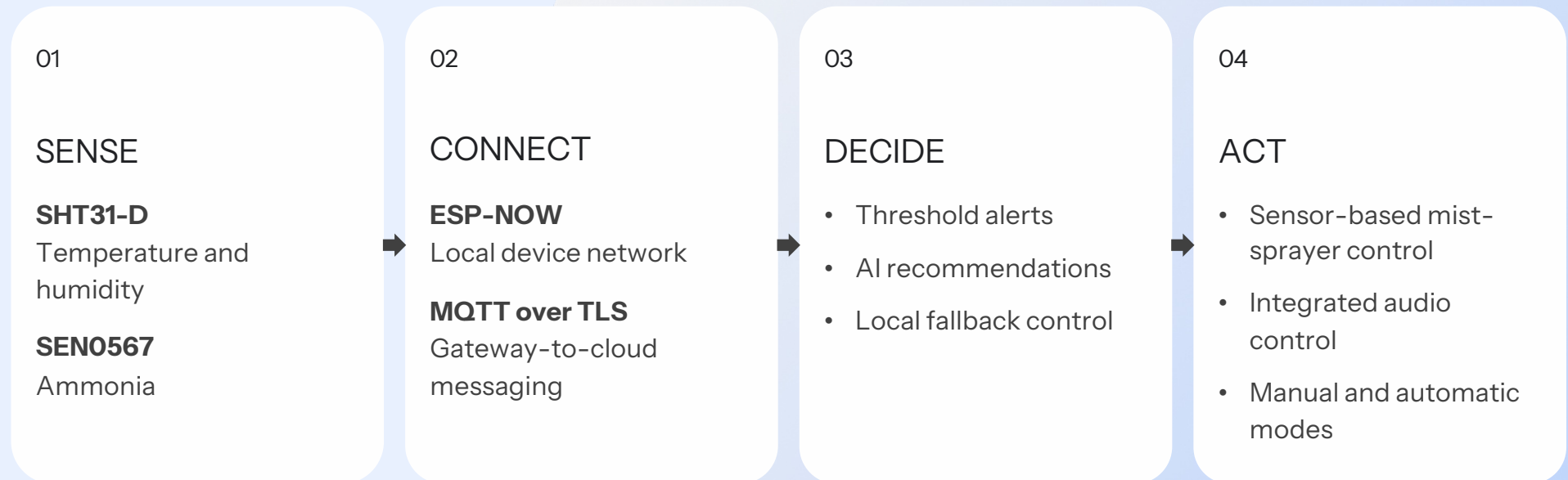
No integrated,
continuous view of
temperature, humidity,
ammonia, and historical
environmental
conditions.

>70%

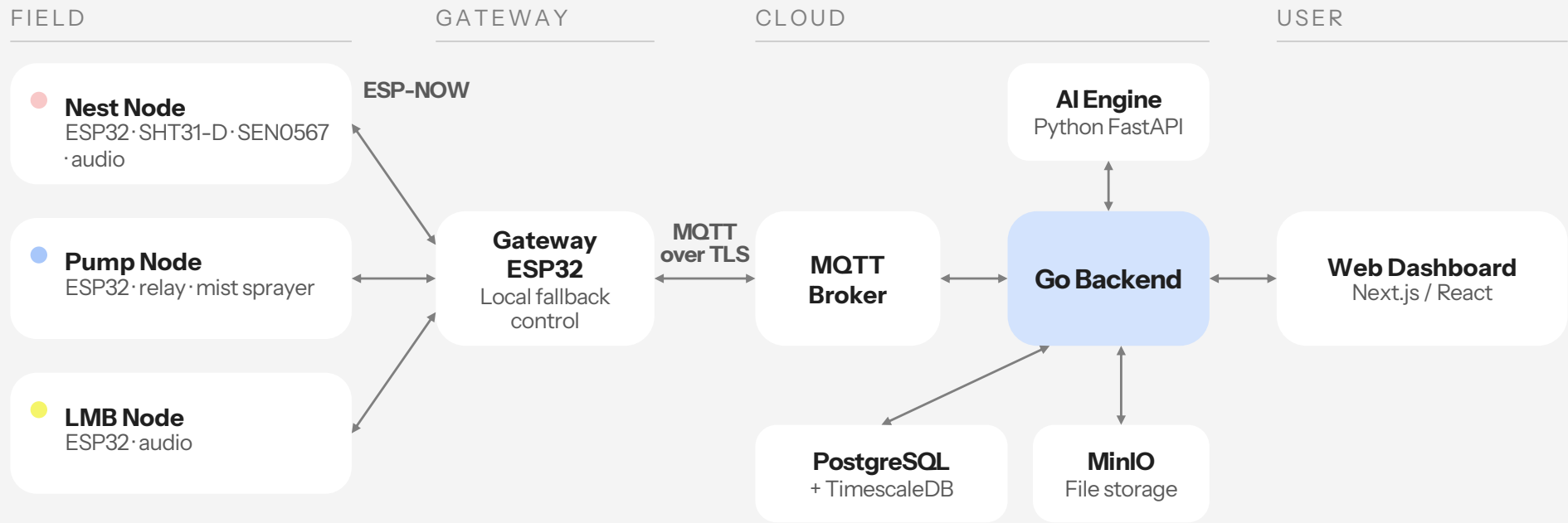
of nests were
concentrated on
the lower floor.

Observation at the project
site

Solution Concept



End-to-End System Architecture



→ telemetry to backend ← commands to gateway & actuators

Local fallback control maintains essential actuator operation during cloud connectivity interruptions.

My Ownership

01

Architecture & Leadership

- Led technical direction and stack selection.
- Designed the end-to-end architecture.
- Defined interfaces across IoT, backend, AI, and web.

02

Embedded / IoT

- Built ESP32 firmware using C/C++ and FreeRTOS.
- Integrated sensors, actuators, and ESP-NOW.
- Implemented gateway MQTT/TLS connectivity.

03

Backend

- Built Go + Chi APIs and telemetry ingestion.
- Integrated PostgreSQL and TimescaleDB.
- Implemented authentication and RBAC.

04

Integration & Validation

- Integrated hardware with backend services.
- Coordinated AI and web integration.
- Conducted deployment and reliability testing.

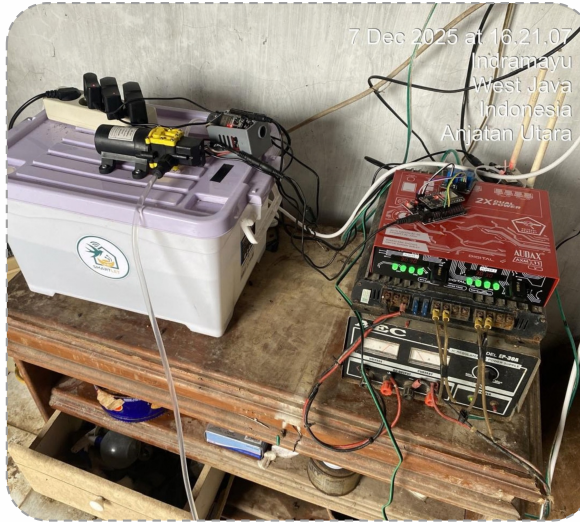
Engineering Decisions

CONSTRAINT	DECISION	RATIONALE
Local device networking	ESP-NOW	Connects field nodes locally through the gateway using the ESP32 radio.
Device-to-cloud messaging	MQTT over TLS	Supports telemetry and control messaging through pub/sub with encrypted transport.
Concurrent backend workloads	Go + Chi	Supports APIs, MQTT subscribers, and concurrent telemetry ingestion.
Telemetry and business data	PostgreSQL + TimescaleDB	Stores time-series readings alongside relational application data.
Cloud interruptions	Local fallback with hysteresis	Maintains essential actuator control when cloud services are unavailable.

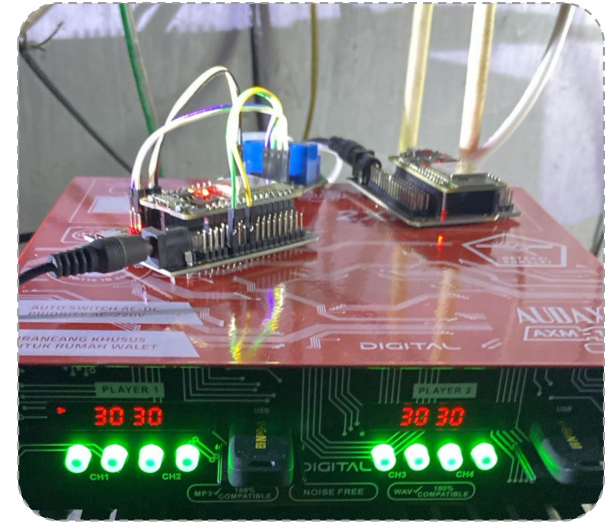
Implementation in the Field



● Hardware assembly



● Field installation



● Actuator integration

Sense → Validate → Transport → Process → Respond

Field Validation & Results

Laboratory testing and a 7-day field deployment at RBW Swiftlead, Indramayu.

7 DAYS

Continuous field
operation

0

Gateway ESP32
crashes/restarts during
field testing

$\geq 99\%$

Packet delivery
success*

1.04 s

Average dashboard-
to-relay response**

Maximum observed deviation against reference: 0.2°C temperature / 0.9 percentage points RH.

*Separate ESP-NOW and MQTT TLS tests. **Mean of five activation/deactivation tests.

Thank You

Questions & Discussion

An integrated platform for monitoring, control, and field validation.

Key lessons

- Technology choices follow system constraints.
- Reliability includes failure handling.
- Ownership requires understanding the whole system.

Muhammad Fuad Fakhruzzaki

Project Lead · System Architect · IoT &
Backend Engineer

Backend Source Code

[Github Repository](#)

