Cloud IoT Energy Monitoring and Automation Pl

Presentation and live demonstration

End-to-end system: telemetry ingestion, 15-minute rule evaluation, automated control, and auditable evidence.

- Single integrated platform boundary
- Simulated energy device for live proof of behavior
- Focus on operational correctness and traceability

Problem and Objective

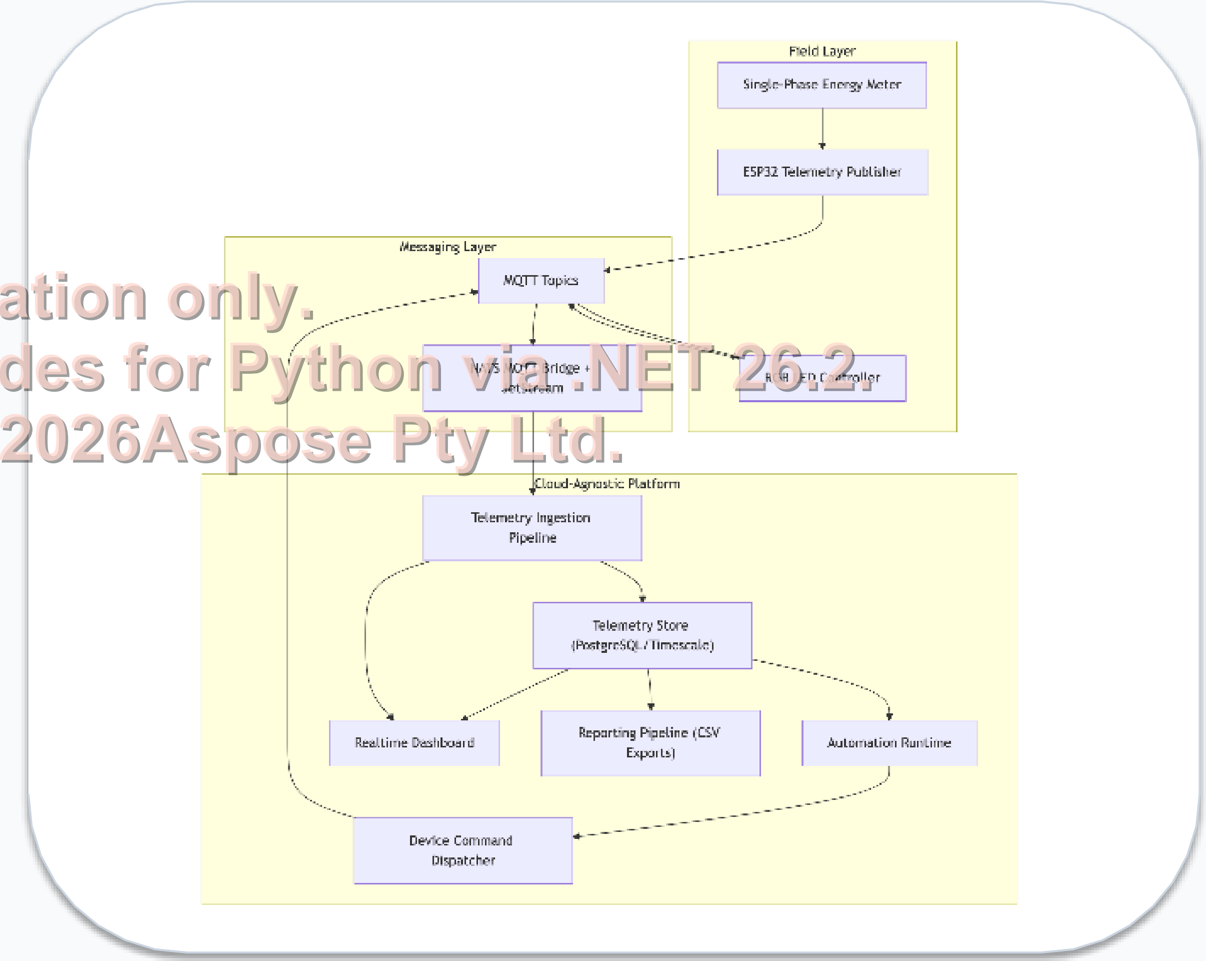
From passive monitoring to automated action

- Many IoT deployments stop at dashboards and manual intervention
- Operational reaction is delayed when humans must inspect every change
- This project implements a deterministic telemetry-to-action loop
- Goal: trigger control behavior based on validated 15-minute energy conditions
- All outcomes must be traceable through run and command records

End-to-End Architecture

A single telemetry substrate for ingestion, automation, control, and reporting

- Devices publish telemetry over MQTT-style topic contracts
- Ingestion validates and persists canonical telemetry records
- Automation evaluates conditions and dispatches control commands
- Dashboard and reporting consume the same consistent data source

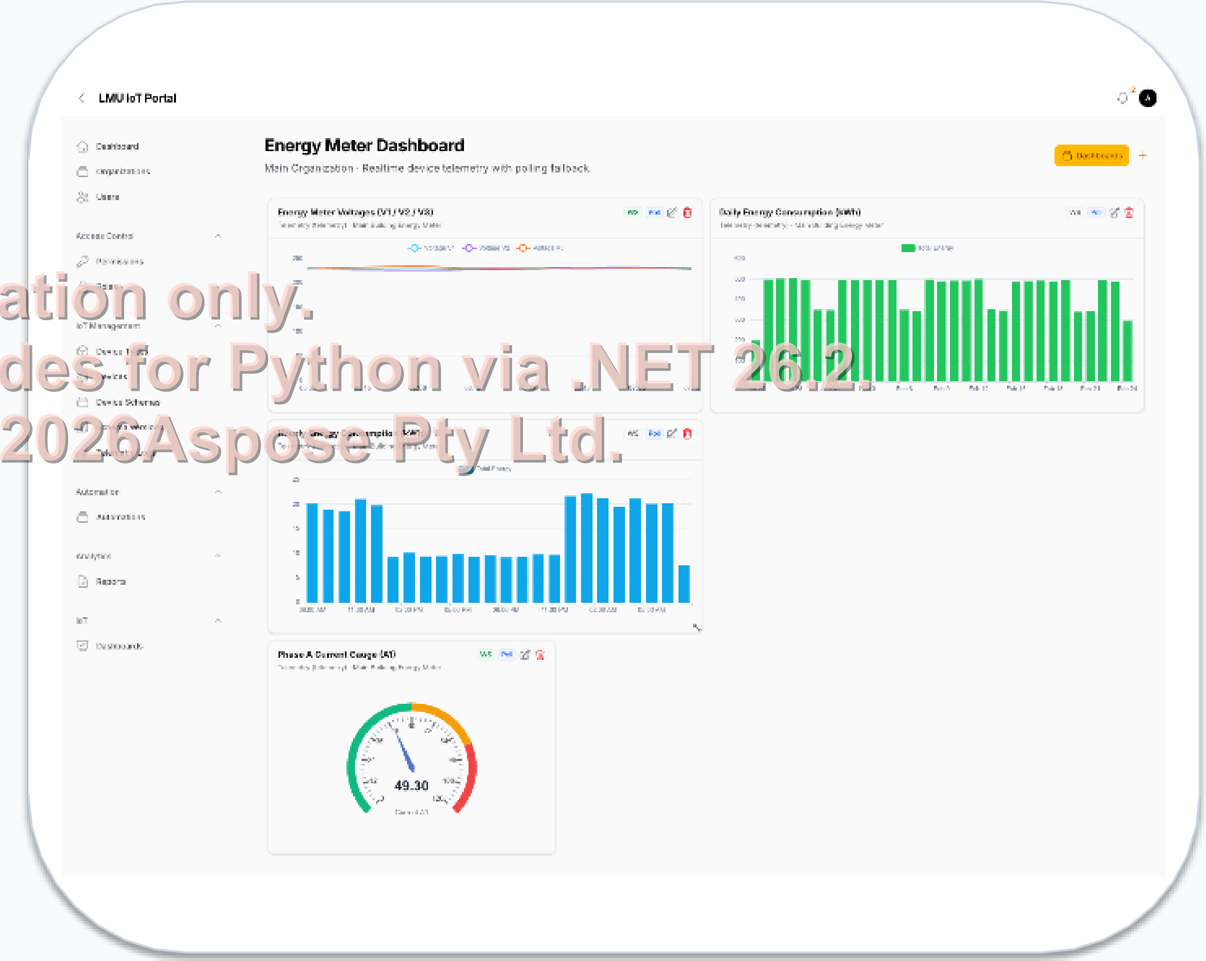

Field Layer
Single-Phase Energy Meter
ESP32 Telemetry Publisher
RGB LED Controller
Messaging Layer
MQTT Topics
NATS MQTT Bridge + JetStream
Cloud-Agnostic Platform
Telemetry Ingestion Pipeline
Telemetry Store (PostgreSQL/Timescale)
Realtime Dashboard
Reporting Pipeline (CSV Exports)
Automation Runtime
Device Command Dispatcher

Architecture overview

Firmware Strategy

Reusable ESP32 runtime with modular device logic

- Common runtime handles WiFi, MQTT, presence, and reconnect behavior
- RGB actuator module implements control + state acknowledgment
- PZEM meter module implements RS485 Modbus telemetry polling
- Read-only meter path in v1 avoids unsafe write-side effects

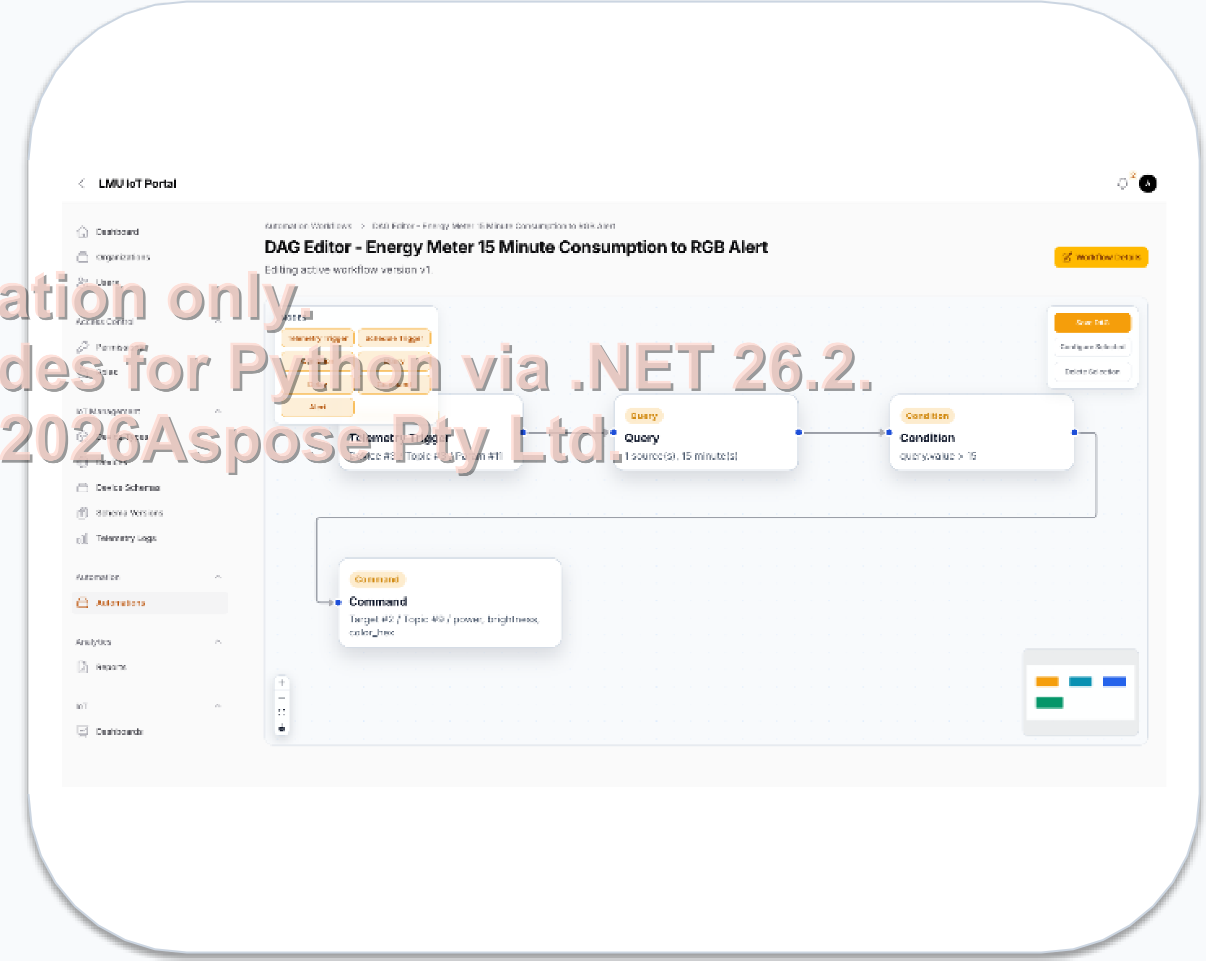

LMU IoT Portal
Dashboard
Organizations
Users
Access Control
Permissions
IoT Management
Automation
Automations
Analytics
Reports
IoT
Dashboards
Energy Meter Dashboard
Main Organization · Realtime device telemetry with polling fallback
Dashboards
Energy Meter Voltages (V1 / V2 / V3)
Daily Energy Consumption (kWh)
Hourly Energy Consumption (kWh)
Total Energy
Phase A Current Gauge (A1)
WS
Poll
49.30

Telemetry dashboard view

Automation Logic

15-minute window rule with deterministic branching

- Rule: MAX(total_energy_kwh) - MIN(total_energy_kwh) over 15 minutes
- Threshold pass branch dispatches an RGB alert command
- Threshold fail branch correctly avoids control side effects
- Command lifecycle is reconciled with device feedback for auditability



DAG Editor - Energy Meter 15 Minute Consumption to RGB Alert

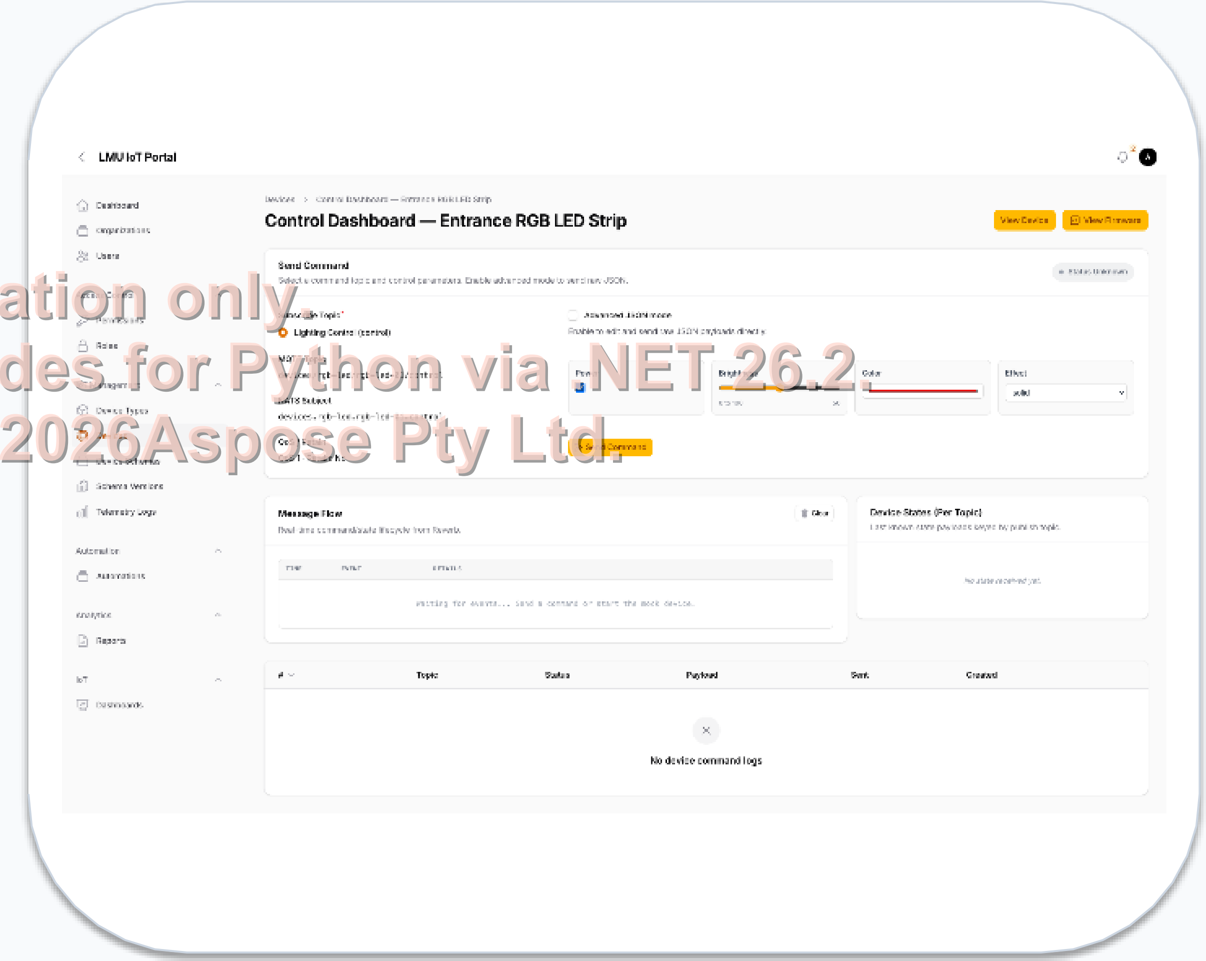
Editing active workflow version v1

Automation DAG and condition flow

Demo Walkthrough

Simulation-first proof of end-to-end behavior

- Start from normal baseline with no active alert
- Publish simulated meter telemetry with rising energy values
- Observe real-time dashboard updates and rule evaluation
- Verify command dispatch and device acknowledgment
- Show reporting evidence and a safe negative-path case



Devices > Control Dashboard — Entrance RGB LED Strip

Control Dashboard — Entrance RGB LED Strip

Send Command

Select a command topic and control parameters. Enable advanced mode to send raw JSON.

Subscribe Topic*

● Lighting Control (control)

□ Advanced JSON mode

Enable to edit and send raw JSON payloads directly.

MQTT Topic

devices/rgb-led/rgb-led-.../control

NATS Subject

devices.rgb-led.rgb-led-...control

QoS / Retain

QoS 1 · Retain No

Power ☑

Brightness

Color

Effect: solid

Message Flow

Real-time command/state lifecycle from Reverb.

TIME	TYPE	DETAILS

Waiting for events... Send a command or start the mock device.

Device States (Per Topic)

Last known state payloads keyed by parent topic.

No state received yet.

#	Topic	Status	Payload	Sent	Created

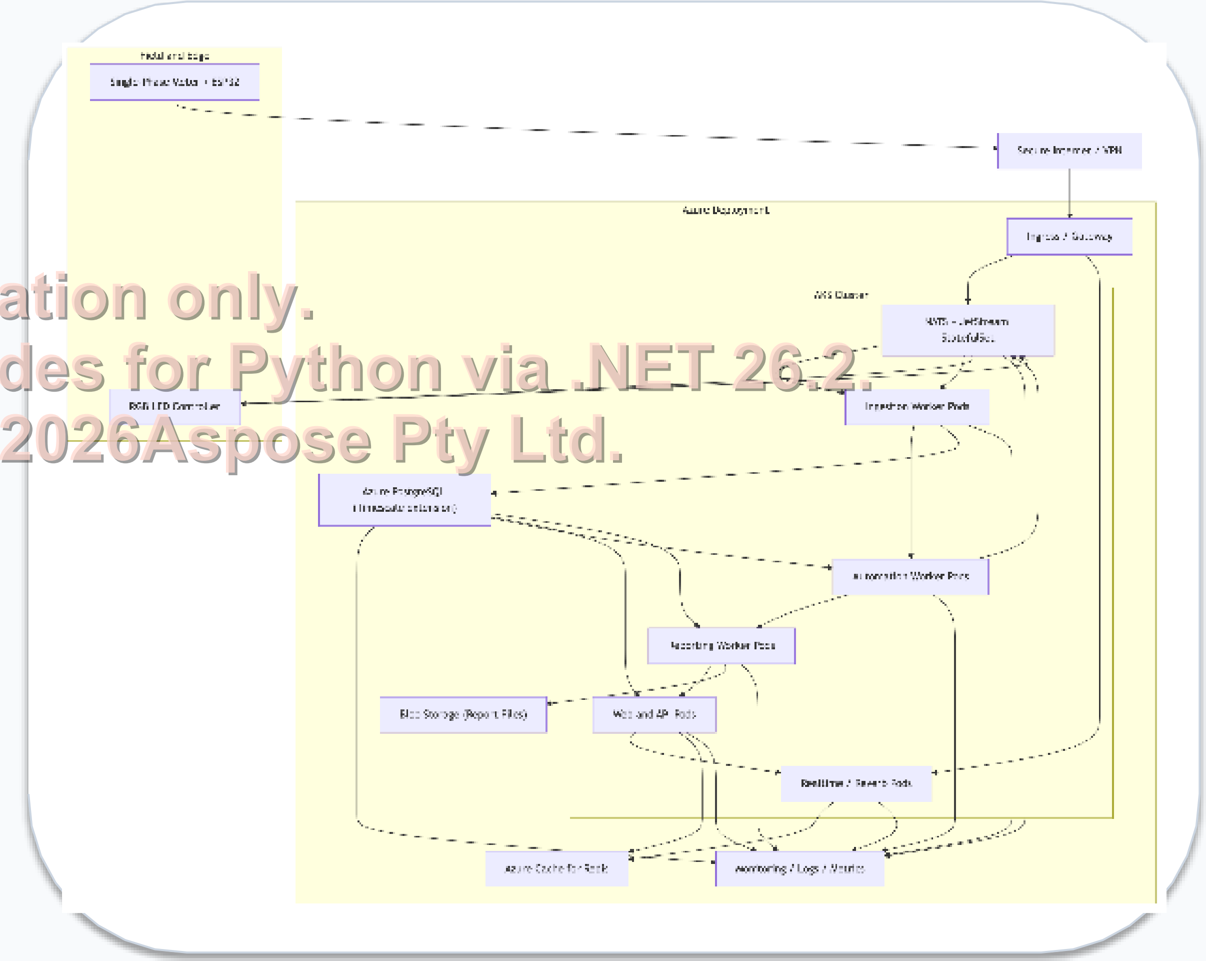
No device command logs

Control lifecycle and feedback reconciliation

Deployment and Scalability

Cloud-agnostic architecture with Kubernetes-ready separation

- Workloads can scale independently across ingestion, automation, and reporting
- Containerized service boundaries support managed Kubernetes platforms
- Queue-backed processing smooths spikes in telemetry and export requests
- Contract-driven onboarding supports future device expansion



Portable deployment model

Conclusion

Full-loop IoT operations demonstrated

- Telemetry is converted into deterministic automated control
- The platform provides clear operational and audit evidence
- Architecture and firmware patterns are reusable for future growth
- Result: ingest, decide, act, and verify within one coherent system

Q&A