# Cloud IoT Energy Monitoring and Automation Platform



14-minute assessed presentation with five end-to-end simulation



Member 1 - Intro | Member 2 - Logic | Member 3 - Demo

**Telemetry -> decision -> control -> verification**

# Run of Show

Clear 14-minute pacing for 3 presenters

- 00:00-05:00 Presentation by Member 1 and Member 2
- 05:00-14:00 Live demo by Member 3
- Target: prove full loop, not only visualization
- Final 40s reserved for crisp conclusion

**Timeline**

**Member 1**

Problem, architecture, firmware strategy

**Member 2**

Automation rule, evidence, demo handoff

**Member 3**

Live simulation and verification

# Why This Project Matters

From passive dashboards to active operations

## Problem

- Many IoT systems only visualize data
- Operational decisions remain manual and delayed
- Tool fragmentation increases integration overhead
- Low traceability makes audit difficult

## Our Objective

- Ingest live telemetry reliably
- Evaluate a deterministic 15-minute rule
- Dispatch control commands automatically
- Verify every action through feedback and logs

# End-to-End Platform Architecture

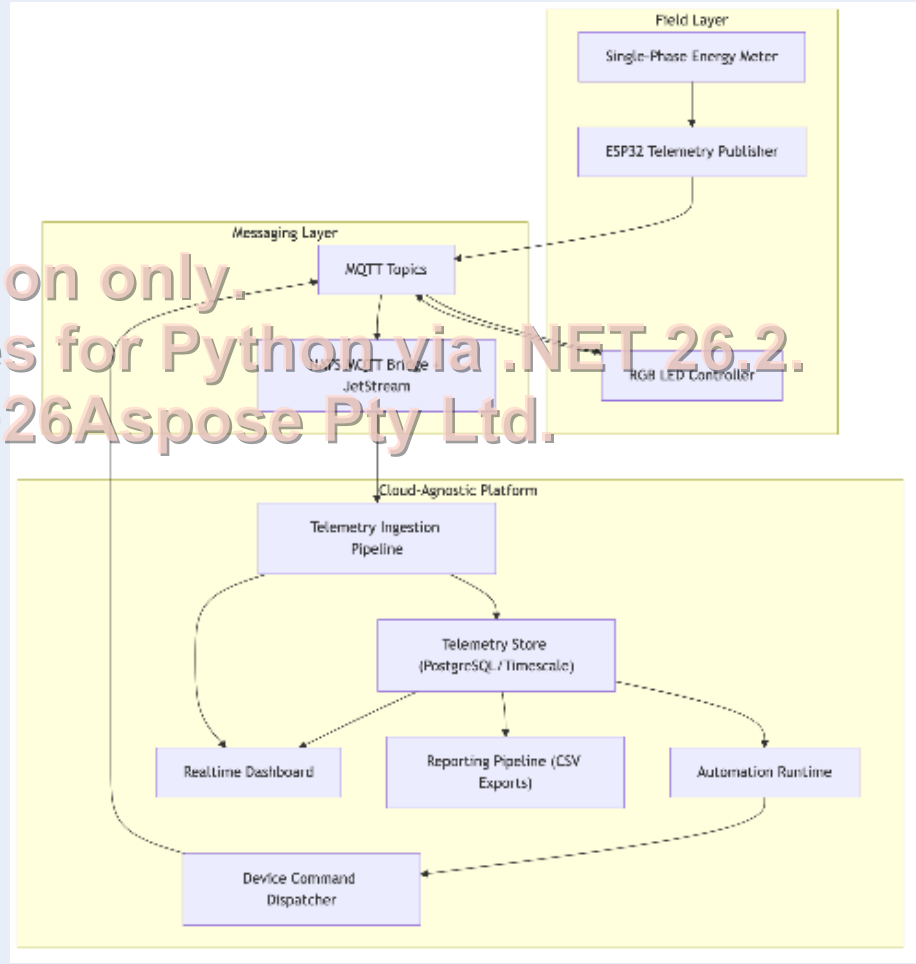
Single operational boundary for ingest, automation, control, and reporting

- Unified telemetry contract
- Near real-time condition evaluation
- Deterministic command lifecycle
- Shared source for dashboard + exports

## What this architecture gives us

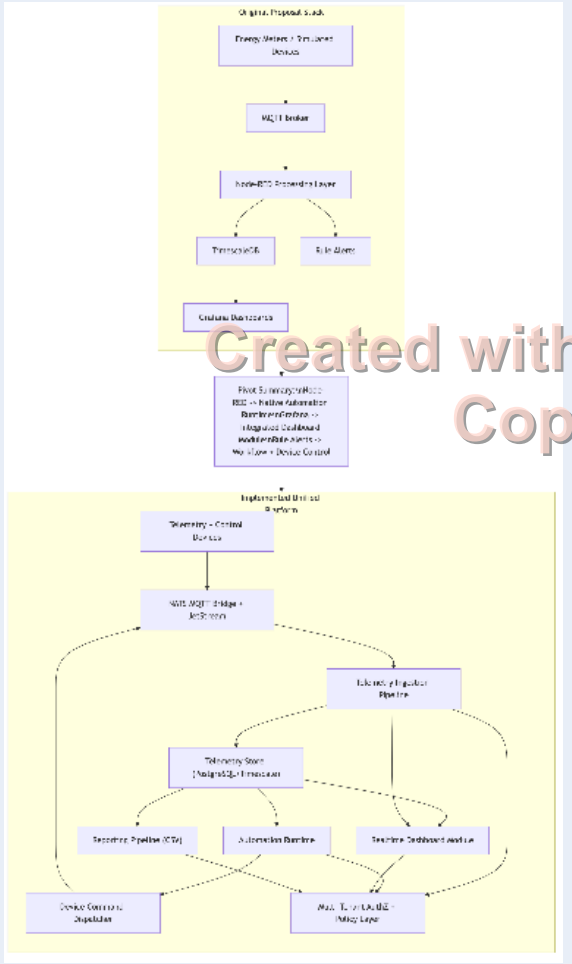
- Repeatable governance at team scale





# Design Decision

Why we moved from fragmented tools to an integrated platform



## Practical benefits in this project

- One governance boundary for users, workflows, and evidence
- Fewer integration seams to maintain under time pressure
- Lower risk of mismatched schemas and policies
- Faster onboarding of new devices using shared contracts
- Cleaner story for assessment and real operations

# Firmware Strategy

Reusable ESP32 runtime + device-specific modules

## Common Runtime Layer

- WiFi/MQTT lifecycle
- LWT + presence publishing
- Reconnect and retry behavior
- Shared telemetry envelope

## Device Modules

- RGB actuator: control + state acknowledgment
- PZEM meter: RS485 Modbus polling
- Read-only safety for meter.v1
- Same contract, easier future expansion

# Automation Logic

15-minute energy rule drives deterministic control

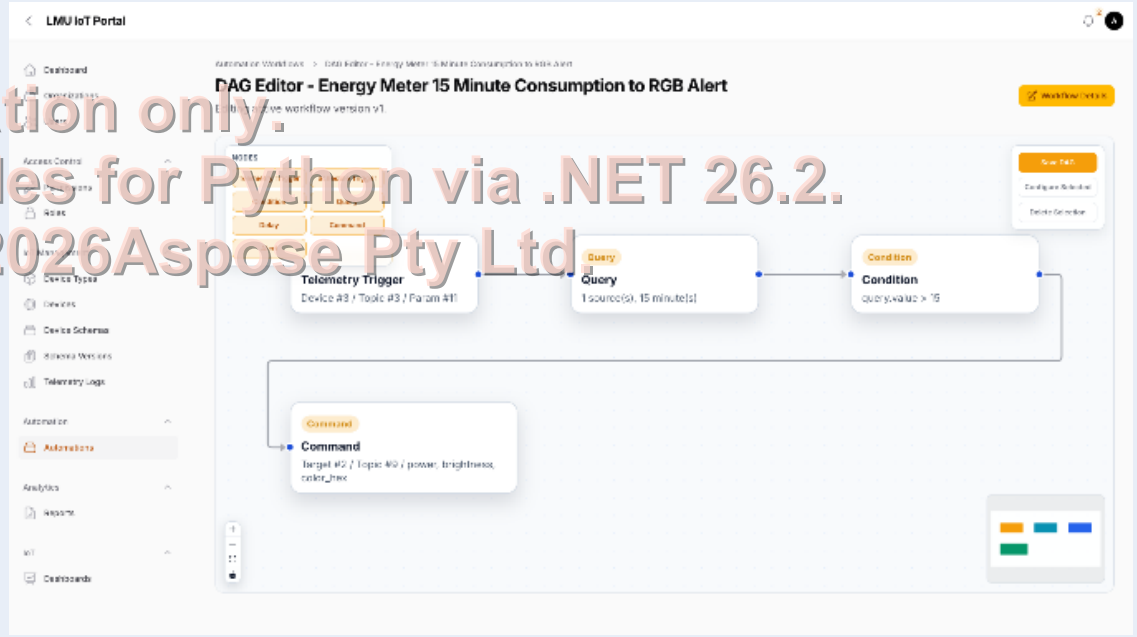
**Consumption (15 min) = MAX(total_energy_kwh) - MIN(total_energy_kwh)**



- If threshold passes -> dispatch RGB alert command
- If threshold fails -> no control side effects

## Branching behavior

- Every step stored in run-step records
- Command lifecycle reconciled with device feedback



# Live Demo Plan (Member 3)

9-minute simulation with measurable checkpoints

## Minute-by-minute flow

- 05:00-06:00 Baseline state and expected behavior
- 06:00-08:00 Simulated meter publishes rising telemetry
- 08:00-09:30 Rule threshold crosses and workflow fires
- 09:30-11:00 RGB command dispatch + acknowledgment
- 11:00-12:30 Lifecycle evidence + CSV report
- 12:30-13:30 Negative path: no command when rule fails
- 13:30-14:00 Conclusion

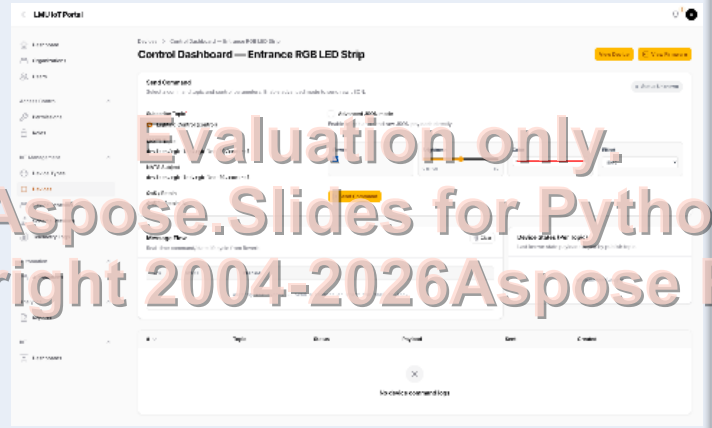


## 4 Success Checkpoints

- Telemetry spike visible
- Rule passes
- Command dispatched
- Feedback verified

# Operational Evidence

What assessors should see during and after the demo





**Realtime Dashboard** | **Control Lifecycle** | **Reporting Pipeline**

**Evidence chain: ingest -> evaluate -> command -> acknowledge -> export**

# Demo Risk Control

Keep delivery stable even if live environment is noisy

## Live risks + mitigation

- Service not ready -> verify tabs and baseline data 10 min before
- Simulation lag -> keep fallback snapshot tabs pre-opened
- Rule not triggering -> pre-check threshold and latest telemetry
- Live delay -> narrate using cached records as source of truth

## If live demo fails

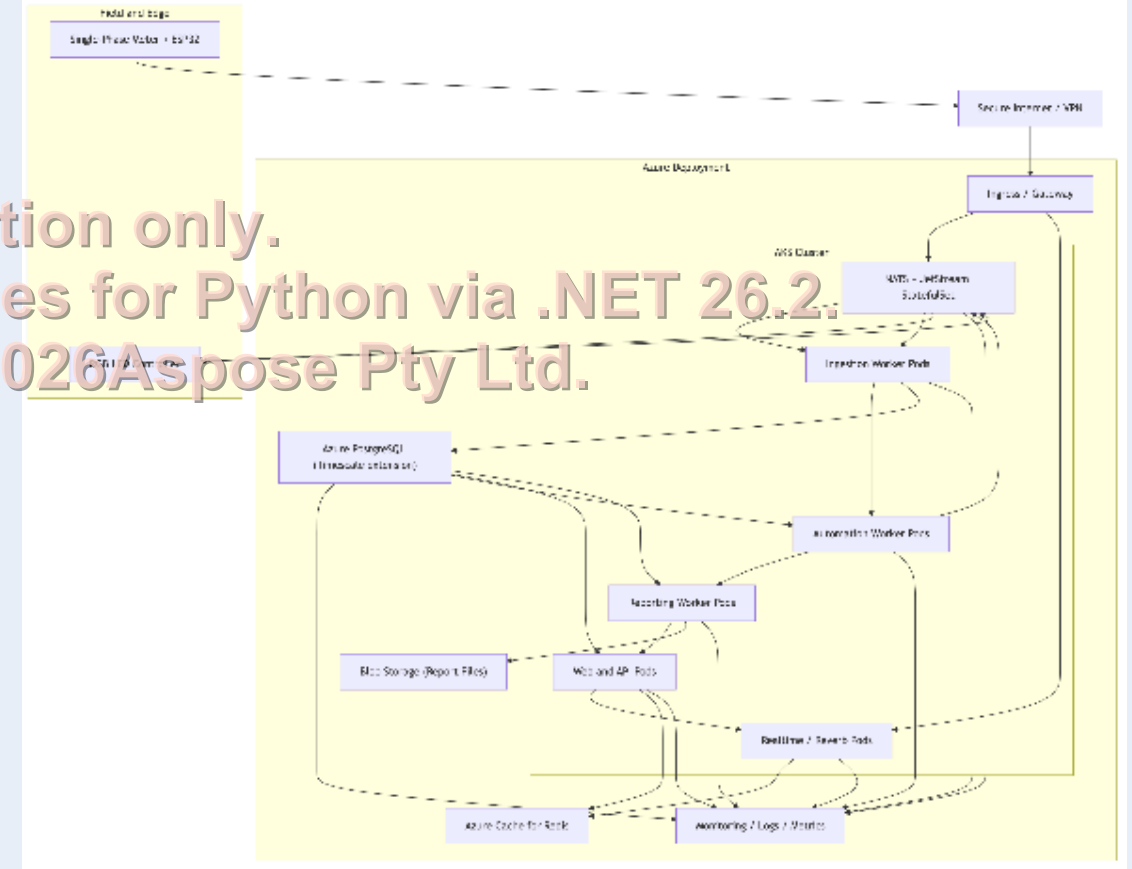
- Use prepared screenshots in fixed order
- Keep the same 4 checkpoint narrative
- Show that each stage has stored evidence
- Finish with the same closing statement

# Scalability and Portability

Designed for growth beyond a classroom prototype

## what scales well

- Cloud-agnostic deployment model
- Kubernetes-ready service decomposition
- Schema-based device onboarding
- Queue-backed ingestion and reporting
- Audit-friendly command and run records



# Full-Loop IoT Proven

## Ingest. Decide. Act. Verify.



Q&A

Presentation team: Member 1, Member 2, Member 3