Resilient Adaptive Edge-Cloud Framework (RAECA) – Summary

Problem: Rural and remote regions face unreliable power and intermittent connectivity, which disrupts cloud/edge services for agriculture, telemedicine, and emergency response. We need a lightweight, energy■efficient, fault■tolerant framework that continues operating under power/network constraints.

Objectives:

• Design a resilient, energy■efficient edge■cloud framework for rural environments.
• Ensure reliability and fault tolerance via adaptive orchestration and federated learning.

Architecture:

Layers: (1) Device: IoT sensors/actuators. (2) Micro■Edge: ESP32 preprocessing, thresholds, TinyML (optional), MQTT QoS, ring buffer. (3) Edge Gateway: Mosquitto, Node■RED, TFLite, SQLite cache, energy/link monitors, adaptive scheduler. (4) Cloud: FL aggregator, storage, dashboards, model registry/policy.

End■to■End Flow:

Sensors sample → ESP32 filters/compresses → publishes via MQTT (QoS1/2, LWT). Gateway authenticates, caches, enriches, runs inference → streams to cloud when backhaul is available. Compute placement adapts to SoC/CPU/link. Federated learning aggregates updates in the cloud and rolls out signed models.

Resilience & Efficiency:

• Fault tolerance: store■and■forward, persistent sessions, ordered replay, watchdogs, brownout handling.
• Energy efficiency: duty cycling, adaptive sampling, event■driven reporting, quantized models.
• Adaptive orchestration: health■ and policy■driven placement across ESP32/gateway/cloud.
• Trust■aware: mTLS, device identity/ACLs, signed model/policy updates.

Outcomes:

Precision agriculture: timely irrigation, fewer missed alerts despite outages. Telemedicine: local triage with resilient uploads. Emergency services: on■device event detection and immediate actuation with guaranteed delivery semantics.