Automated IIoT Service Deployment Using MEC in Drone-Assisted Smart Agriculture

Team No. 16

Problem Statement

- Modern farming faces challenges in rapidly deploying and managing digital services like crop monitoring, pest detection, and irrigation control

- Traditional cloud-based or manual IT setups cause high delay (latency), network congestion, and integration obstacles—especially over vast or remote fields where reliable connectivity is critical

- Manual or slow deployment makes it impossible to react to sudden changes (like weather or pests) and to scale up for large harvest operations

Is This an Edge Problem ?

- In agriculture, many critical tasks like monitoring crops with drones, analyzing sensor data, or detecting pests need to be performed close to where data is produced (in the field).

- Cloud-Based Approaches Are Too Slow Sending farm and drone data to faraway cloud servers causes delays (high latency) and network congestion, making fast reactions impossible and overloading rural connections.

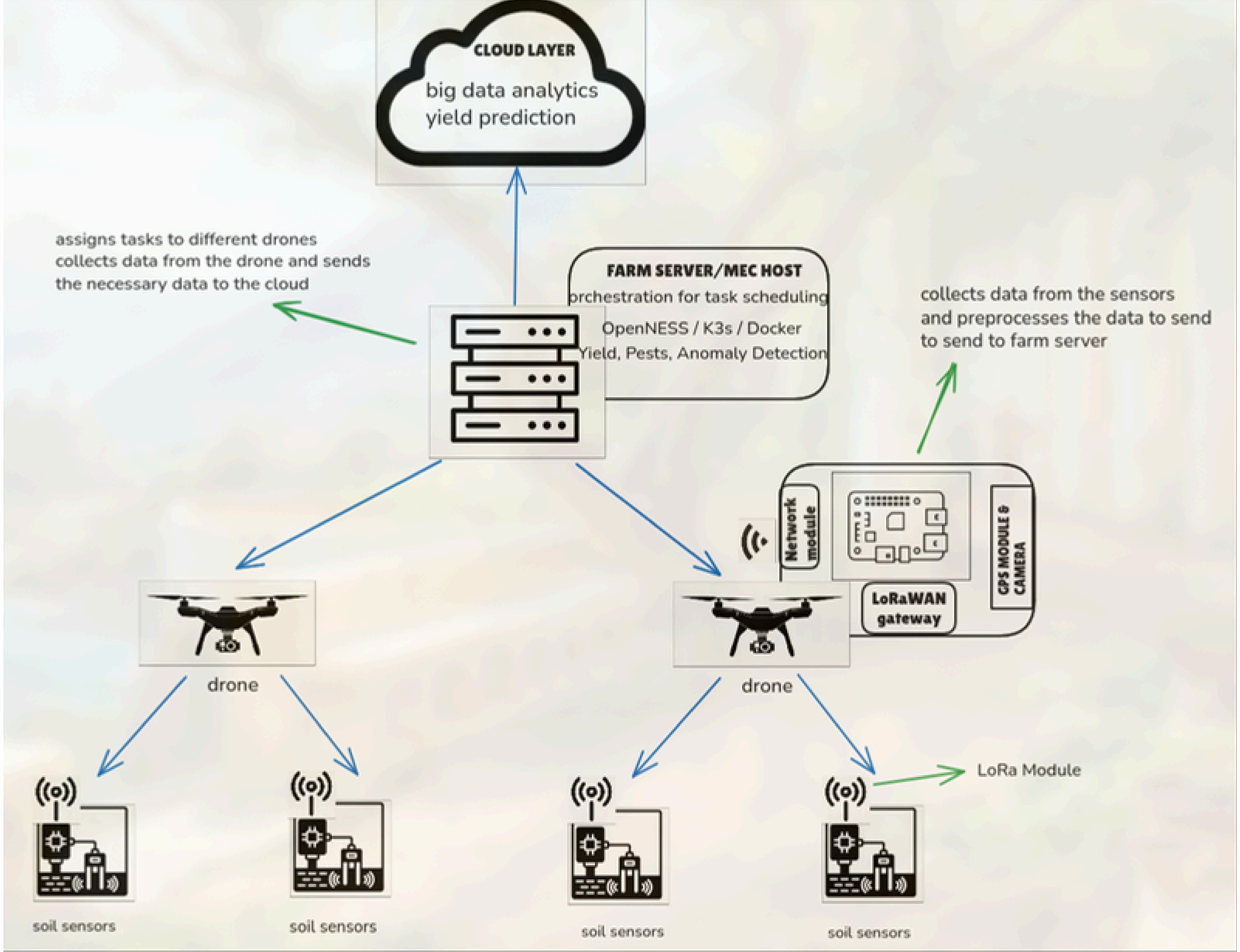
- Edge computing solves this by bringing processing and service deployment right to the farm's edge, enabling fast, local analysis and actions (via drones, sensors, and edge servers).

- **MEC (Multi-access Edge Computing) :**

  MEC is a telecom-driven form of edge computing that places compute resources at mobile network edges, unlike general edge computing, offering low latency and network-aware services through 4G/5G integration.

- **Research Focus:**

  The proposed MEC-based architecture automates IIoT service deployment at the edge and fog, ensuring reliable, fast, and scalable operation right where farming decisions are needed most—not in distant data centers.


CLOUD LAYER
big data analytics
yield prediction
assigns tasks to different drones
collects data from the drone and sends
the necessary data to the cloud
FARM SERVER/MEC HOST
orchestration for task scheduling
OpenNESS / K3s / Docker
Yield, Pests, Anomaly Detection
collects data from the sensors
and preprocesses the data to send
to send to farm server
Network module
GPS MODULE & CAMERA
LoRaWAN gateway
drone
drone
LoRa Module
soil sensors
soil sensors
soil sensors
soil sensors

Simulation Setup

We implement our smart farming architecture using YAFS (Yet Another Fog Simulator), a Python-based event-driven simulator designed for fog, edge, and cloud environments. YAFS allows dynamic application deployment, custom mobility, and detailed event control.

Simulation Workflow

Modules and Tasks

- SensorModule simulates periodic data emission from soil sensors
- DroneProcessor handles basic filtering and sends images
- AnomalyDetector runs on MEC to check for irrigation issues, diseases
- CloudUploader sends processed data to cloud storage for ML tasks

Realism Features

- Drone movement modeled using mobility events
- LoRaWAN simulated as high-latency, low-bandwidth links
- Feedback loops simulate real-time task redeployment

This simulation setup enables detailed analysis of edge-fog-cloud orchestration under IIoT constraints.

Simulator	Strengths	Limitations
iFogSim2	Strong fog and cloud support; allows VM modeling; mature Java-based framework	Poor mobility support; limited to fixed placement; less flexibility for IIoT
LEAF	Built for federated learning; Python-based; scalable architecture	Not suitable for IIoT hardware modeling; no mobility; focuses only on ML
SimPy	General-purpose, highly customizable; full control over logic via Python	No native fog/edge support; must build all abstractions manually
YAFS	Event-driven, supports edge-fog-cloud hierarchy, mobility events, dynamic scheduling, Python-based	Smaller community, fewer visual tools, steeper learning curve

Why YAFS is the Right Fit

- Mobility and Dynamic Placement: YAFS supports custom mobility models (essential for drone simulation) and dynamic placement strategies that match our edge-first scheduling.
- Python-Based and Modular: Easy to integrate with our workflow and customize using Python scripts.

State of the Art Literature Review

1. Reinforcement Learning for ABS Scheduling (2022)

Proposed reinforcement learning-based task scheduling for aerial base stations in agriculture. Lacks multi-layer MEC orchestration and fog–cloud integration.

2. Q-Learning Offloading Model for UAVs (2022)

Developed a Q-learning-based task offloading model considering deadlines and battery levels. No MEC architecture or service deployment using APIs.

3. UAV Trajectory Optimization in MEC (2019)

Focused on UAV trajectory optimization and offloading in MEC systems.Lacks real-time task scheduling logic and MEC API support.

4.DAG-Based IIoT Scheduling with LAPS (2023)

Applied DAG-based scheduling for IIoT tasks using low-altitude platforms. Static setup, no drone mobility or dynamic orchestration.

Comparison of Reviewed Works

Paper	Task Scheduling	Drone/ABS Mobility	MEC API Use
Reinforcement Learning for ABS Scheduling (2022)	✅ Energy/Deadline-aware	✅ Aerial Station	❌ No
Q-Learning Offloading Model for UAVs (2022)	✅ Q-learning	✅ UAV-based	❌ No
UAV Trajectory Optimization in MEC (2019)	✅ Offloading Logic	✅ UAV Planning	❌ No
DAG-Based IIoT Scheduling with LAPS (2023)	✅ DAG Scheduling	❌ No	❌ No
IIoT-as-a-Service with MEC Framework (Davoli et al., 2021)	✅ Latency + CPU + Power	✅ Drone-supported	✅ MEC011, MQTT

Identified Gaps + Our Improvements

Gaps Found in Existing Works	How Our Work Overcomes These Gaps
No use of MEC APIs	MEC integration (for dynamic service deployment)
No real-time metrics like latency, CPU usage, and power considered	Real-time task scheduling using latency, CPU, and power
Limited or no drone mobility support in scheduling frameworks	Full drone mobility support with adaptive placement
No orchestration across edge, fog, and cloud layers	Multi-layer orchestration with intelligent service placement
Mostly static or semi-mobile deployments (limited real-world use)	Built for fully mobile, dynamic IIoT environments

TASK SPLIT

NIRVESH
Topology and node modelling

PRITHIV.A
Module & Event Logic

DANVANTH SC
Scheduling implementation

VIVIN RAKUL
Result Analysis and evaluation

THANK YOU!

Q&A 🤝