

A Lightweight Cloud-to-Edge Orchestration Architecture for Smart Agriculture Across Terrestrial and Non-Terrestrial Networks

Joaquín Cruces¹, Alba Carmona¹, Diogo Oliveira¹, Sergio Aguilar¹

¹ Sateliot, Barcelona, 08038, Spain, {joaquin.cruces, alba.carmona, diogo.oliveira, sergio.aguilar}@sateliot.com

Abstract

The increasing demand for smart agriculture applications relies on the efficient collection and processing of data from spatially distributed sensors. A major bottleneck for large-scale adoption is the lack of connectivity in remote agricultural areas, where terrestrial communication infrastructures are often unavailable. To address this limitation, recent advancements have focused on integrating non-terrestrial infrastructure, such as satellite systems, to extend coverage to underserved regions. However, the integration of terrestrial (TN) and non-terrestrial networks (NTN) to support reliable data collection remains challenging, particularly under resource constraints and intermittent connectivity conditions. This paper presents a lightweight orchestration architecture developed within the AGRARIAN project, enabling application deployment across a cloud-to-edge continuum spanning edge devices, satellite nodes, and cloud platforms. The proposed framework supports TN-NTN convergence, aligning with emerging 5G and future 6G architectures to extend connectivity and digital services to remote agricultural areas. Based on containerization and a lightweight Kubernetes framework, the orchestrator manages application deployment, reducing operational complexity and enabling flexible service provisioning across infrastructure layers. By supporting distributed data collection and processing, the architecture provides a scalable foundation for resilient agricultural monitoring systems.

Keywords: Cloud-to-Edge Computing, Lightweight Orchestration, Terrestrial–Non-Terrestrial Network Convergence, Smart Agriculture, 5G/6G Networks

1. Introduction

The rapid evolution of modern communication networks has fostered the development of hybrid infrastructures that seamlessly integrate TN and NTN [Liu et al., 2022]. This convergence is vital for enabling connectivity in remote agricultural regions, establishing the primary motivation behind the AGRARIAN architecture where traditional terrestrial networks exhibit severe coverage and capacity limitations. Concurrently, the proliferation of Internet of Things (IoT) sensors deployed for distributed smart agriculture necessitates robust computing capabilities to ensure low latency, real-time data processing, and localized decision-making [Bonomi et al., 2012; Shi et al., 2016].

This operational environment creates a device-to-edge-to-cloud continuum, radically extending computation capabilities across diverse network boundaries [Milojicic, 2020]. Despite the operational potential of this continuum, orchestrating distributed applications across hybrid TN/NTN environments presents significant technical challenges. Traditional cloud-centric orchestration frameworks are inherently unsuitable for edge devices and non-terrestrial infrastructure due to resource scarcity, latency and bandwidth constraints, and dynamic network topologies. Specifically, satellite computing environments endure stringent computational and energy constraints alongside intermittent connectivity that cause heavy, conventional orchestration platforms to fail [Milojicic, 2020; Costa et al., 2023].

To overcome these barriers, this paper proposes a novel, lightweight orchestration framework specifically engineered for hybrid TN/NTN infrastructures to enable the device-to-edge-to-cloud continuum in smart agriculture. The primary contributions of this work include a resource-aware architecture leveraging lightweight cloud-native technologies optimized for

constrained satellite hardware, automated Continuous Integration/Continuous Deployment (CI/CD) pipelines adapted for distributed systems, and the design of a representative agricultural use case scenario implemented within a testbed incorporating NTN infrastructure to evaluate the framework's performance across the cloud-to-edge continuum.

2. Orchestration Framework

Deploying and managing microservices in hybrid terrestrial and space-based segments requires a specialized orchestration framework capable of functioning under severe hardware limits and unreliable network conditions [Taherizadeh et al., 2018; Vaño et al., 2023]. Consequently, the architecture utilizes K3s, a lightweight, fully compliant Kubernetes distribution, and Docker for application containerization [Merkel, 2014; Rancher Labs, 2019].

Docker was chosen because it packages applications along with their dependencies into isolated, lightweight Linux containers, ensuring absolute consistency and portability across the highly heterogeneous hardware of the cloud, fog, and edge boundaries. K3s was selected as the orchestrator because it significantly condenses the traditional Kubernetes control plane; it replaces the heavy data store with a lightweight SQLite database and packages all requisite components into a single minimal binary. This reduces the orchestration memory footprint from gigabytes to megabytes, uniquely enabling it to run on embedded Raspberry Pi field gateways and highly constrained satellite payload processors without overwhelming available compute resources [Rancher Labs, 2019]. Furthermore, the K3s server-agent topology inherently handles intermittent connectivity; local edge agents can continue executing their localized Docker containers and delay-tolerant tasks even when connection to the central cloud control plane drops during satellite orbital passes.

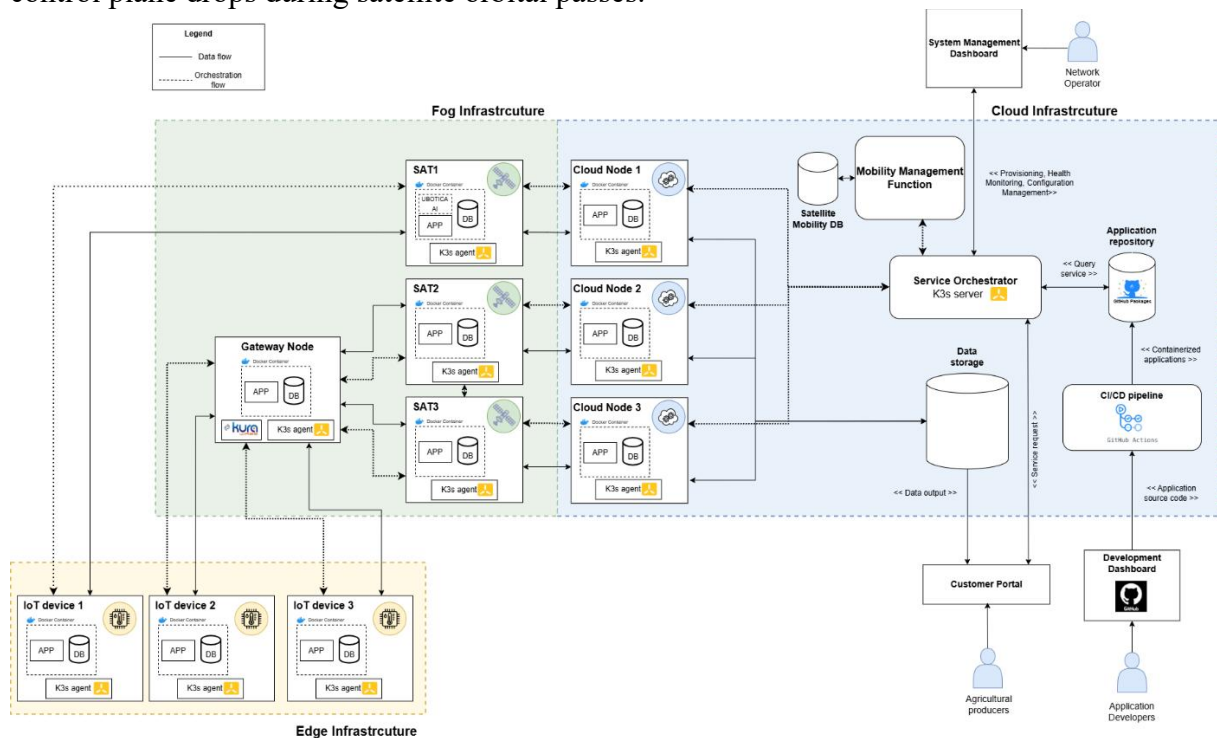


Fig 1. Architecture of AGRARIAN's distributed ecosystem illustrating the interactions between the orchestrator and distributed nodes across cloud, fog, and edge infrastructures.

Fig 1 illustrates the architecture of the distributed ecosystem designed in the AGRARIAN project. This ecosystem allows workloads to migrate across differing infrastructure domains based on computational and latency constraints. The infrastructure spans high-performance Cloud Nodes for long-term storage and central orchestration, intermediate Fog Infrastructure

utilizing satellite nodes for localized space-based preprocessing, and Edge Infrastructure comprised of constrained IoT devices interfacing directly with agricultural sensors. Applications deployed within the system are contributed by developers through a GitHub interface to an open-source application repository [AGRARIAN Project, 2026].

The orchestration framework relies on a K3s cluster to provide lightweight container orchestration across the cloud-to-edge continuum, abstracting heterogeneity of the infrastructure into a unified cluster. The architecture utilizes a topology with a cloud-based server node hosting the control plane and multiple worker nodes running applications into different layers across the infrastructure. The central server node executes the K3s server component, which manages scheduling decisions, resource allocation, and overall orchestration logic, while each worker node runs a K3s agent enabling it to receive workloads from the control plane.

The deployment model integrates these differing computing nodes seamlessly into the same operational cluster. Edge nodes consist of resource-constrained IoT devices that run microservices locally for sensing, preprocessing, and actuation. The non-terrestrial satellite segment consists of satellite-based nodes acting as intermediate fog workers, which provide resilience against connectivity disruptions and enable on-board applications like data filtering. Finally, the cloud nodes represent high-performance servers tasked with complex data aggregation, analytics, and hosting system dashboards.

Microservices are deployed across this cluster utilizing a strategic approach based on resource constraints and application characteristics. Applications with high CPU or memory demands are naturally offloaded to cloud nodes, while lightweight services reside at the edge or satellite boundaries. Furthermore, latency-sensitive tasks (such as actuator control) are prioritized for edge nodes, whereas delay-tolerant services execute on satellites or in the cloud. Network availability also dictates scheduling, permitting tasks to be explicitly placed on edge or satellite nodes during periods of intermittent connectivity to ensure localized functionality until cloud connection is reestablished.

The integration of non-terrestrial nodes into the orchestration framework introduces a fundamental challenge not present in terrestrial environments: the intrinsic mobility of LEO satellites. Unlike fixed edge or cloud nodes, satellite nodes are only intermittently available and have short and predictable connectivity windows. These windows can be accurately estimated using Two-Line Element (TLE) data. To address this constraint, the proposed architecture extends the baseline Kubernetes orchestration with a mobility-aware layer capable of forecasting satellite visibility and aligning application execution with these time-bounded communication opportunities. Introducing the Mobility Management Function ensures that workload placement decisions account not only for resource availability and latency constraints, but also for the availability of satellite nodes.

Building on this capability, the framework adopts an event-driven orchestration strategy that enables deferred and conditional deployment of services targeting satellite infrastructure. When users need to deploy a service, the specific satellite on which the workload will be executed remains transparent to them. Through the Customer Portal, a user can request service to be deployed in a satellite node, and the next visible satellite is automatically selected by the Mobility Management Function (see Fig 1). Applications are defined using HELM charts and initially instantiated in an inactive state, with execution postponed until a predicted satellite pass occurs. The activation mechanism leverages Kubernetes Event-Driven Autoscaling (KEDA), which triggers the scaling of workloads to the desired state upon reaching the scheduled visibility window [Avila et al., 2025]. As a result, the system enables efficient

utilization of intermittent non-terrestrial resources, demonstrating the feasibility of incorporating mobility-aware scheduling into unified TN/NTN orchestration workflows.

3. Application Deployment and Data Collection Workflows

3.1. Use case and testbed

To validate the orchestration framework, a representative agricultural use case scenario was designed focusing on field crop monitoring. This use case is highly dependent on remote soil moisture and temperature tracking to enable automated irrigation.

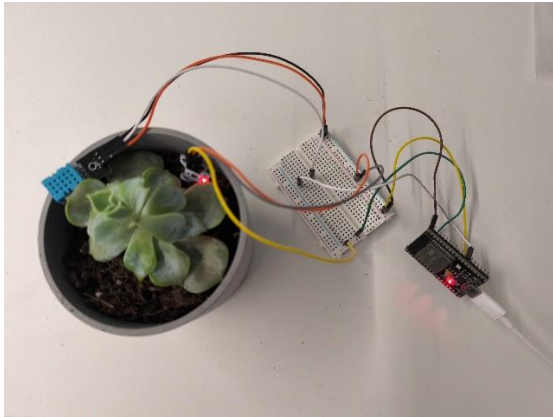


Fig 2. Experimental testbed setup demonstrating the IoT sensor deployment

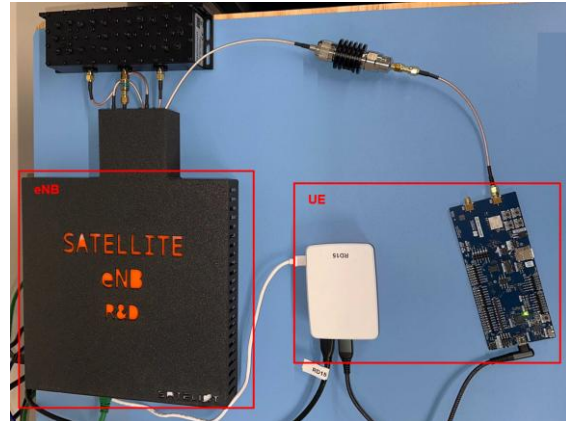


Fig 3. Sateliot's Testbed

For validation of the proposed framework, a testbed was developed in Sateliot's laboratory to emulate a 5G NTN Narrowband IoT (NB-IoT) environment, leveraging space-ready hardware and IoT devices capable of direct satellite communication (see **Error! Reference source not found.**). The physical configuration featured:

- a) **IoT device (UE):** comprising soil moisture and temperature sensors, a Nordic nRF9151 SMA DK enabling NB-IoT NTN connectivity [Nordic Semiconductor, 2026], a Raspberry Pi 4 operating as the User Equipment (UE) edge node responsible for data collection and agregation.
- b) **Emulated satellite (eNB):** a Software Defined Radio (SDR) platform emulating the satellite eNodeB payload.

To accurately reflect non-terrestrial operational constraints, the testbed is capable of emulating signal propagation delays, attenuation, and the intermittent connectivity windows typical of Low Earth Orbit (LEO) satellite communications.

3.2. Application lifecycle

The full application lifecycle achieves an end-to-end cloud-to-edge continuum, transitioning seamlessly from code generation to field-level data processing, as show in Fig. 4.

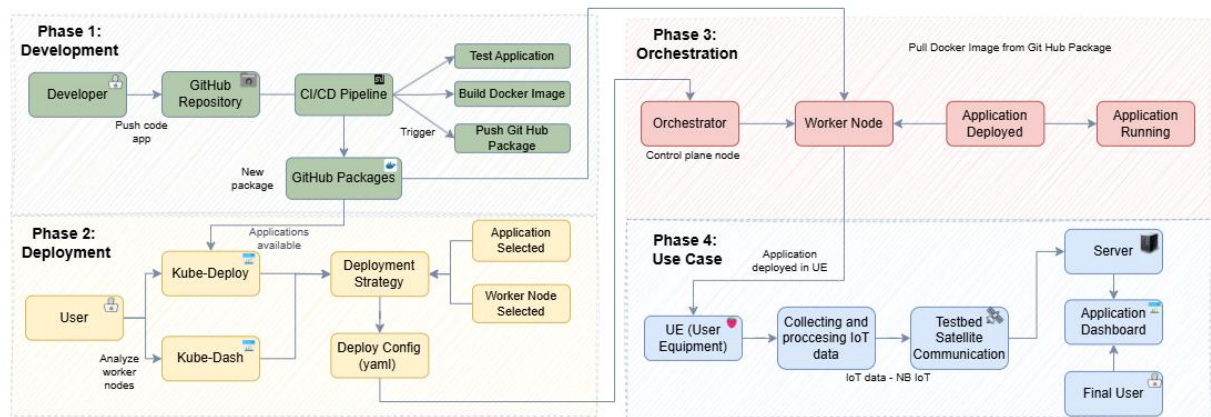


Fig. 4. End-to-end application lifecycle from development to application deployment.

The deployment phase initiates when developers push the source code of the application to the GitHub repository, triggering an automated CI/CD pipeline via GitHub Actions. This pipeline automatically builds and tests the Docker container, pushing the validated image to the GitHub Packages registry.

Central to this deployment phase is the dashboard integration workflow, which creates a comprehensive management environment allowing users to monitor system health and deploy applications seamlessly. The workflow begins with a system monitoring interface (Fig. 5), where users assess real-time cluster health, resource availability, and node status across all computing domains. Based on this, users proceed to the deployment dashboard (Fig. 6) to select the specific application from the catalog and configure deployment parameters. Users choose the target node utilizing the real-time workload information provided by the monitoring dashboard. Once the deployment is initiated, the edge-based UE node automatically performs a pull operation from the application repository through the K3s orchestrator and initializes the microservice locally.

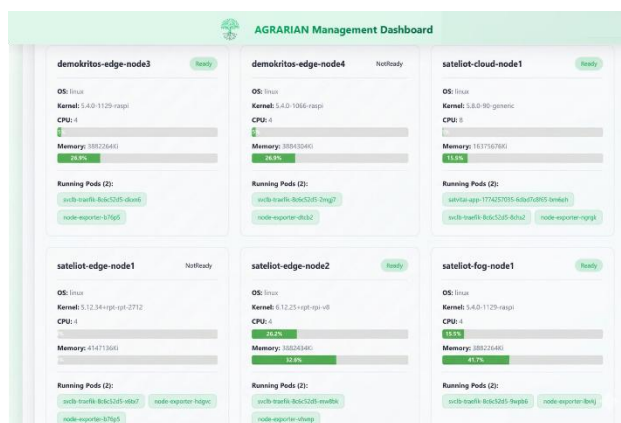


Fig. 5. Management dashboard.

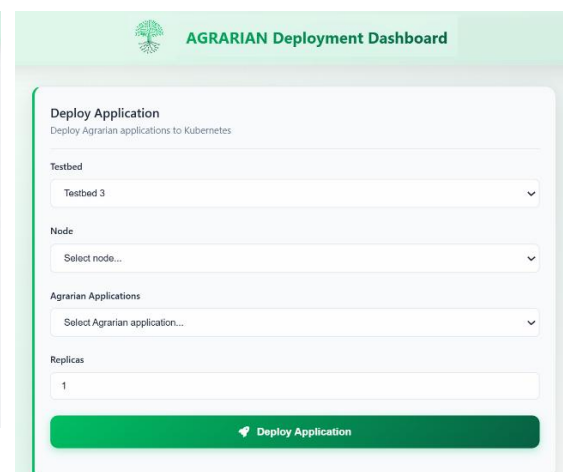


Fig. 6. Deployment dashboard.

Once active, data generation occurs via an ESP32 microcontroller, which queries a DHT11 environmental sensor and an analog soil moisture sensor. The ESP32 parses these metrics into a structured digital string and forwards them over a UART connection to the UE node.

For NTN transmission, the application uses the User Datagram Protocol (UDP) to avoid connection-establishment overhead and support stateless operations. This choice is particularly

suited to LEO satellite environments, where intermittent connectivity and short visibility windows limit the effectiveness of connection-oriented protocols and favor lightweight, delay-tolerant data transfer mechanisms. The telemetry data traverses the service link from the IoT NB-IoT modem to the emulated satellite eNodeB, forwards through the feeder link to the terrestrial 5G Core Network, and routes via a VPN into the cloud infrastructure where an application server with a graphical interface for the user to see the crop monitoring information is available. Each time a satellite pass becomes available, and a feeder link connection is established, the information is updated and an irrigation recommendation is displayed.



Fig. 7. Main dashboard interface with real-time crop monitoring.

Conclusion

The seamless integration of terrestrial and non-terrestrial infrastructure is a key enabler for scaling smart agriculture applications globally, particularly in remote areas lacking reliable connectivity. This paper concludes that traditional cloud-centric orchestration solutions are not well suited to operate under the dynamic conditions and resource constraints of edge and satellite environments. In response, the proposed lightweight cloud-to-edge orchestration architecture, based on containerization and a K3s control plane, demonstrates that distributed application deployment can be efficiently managed across heterogeneous infrastructure layers while addressing limitations in computation, energy, latency, and mobility. Furthermore, the test scenario with the 5G NTN testbed shows that end-to-end application workflows, from IoT-based data acquisition to satellite-enabled transmission and cloud-based visualization, can be executed across the integrated cloud-to-edge continuum. The proposed architecture is applicable to a wide range of use cases and provides a foundation for future resilient and distributed agricultural monitoring systems.

References

- Liu, X., Liu, J., Chen, W., and Zhang, Z. (2022) Hybrid satellite–terrestrial networks toward 6G: Key technologies and challenges. *Sensors*.
- Bonomi, F., Mito, R., Zhu, J., and Addepalli, S. (2012) Fog computing and its role in the Internet of Things. In: *Proceedings of the MCC Workshop on Mobile Cloud Computing*. ACM.
- Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016) Edge computing: Vision and challenges. *IEEE Internet Things J.*
- Milojicic, D. (2020) The edge-to-cloud continuum. *Computer*.

Costa, B., Bachiega, J., de Carvalho, L. R., and Araujo, A. P. F. (2023) Orchestration in fog computing: A comprehensive survey. *ACM Comput. Surv.*

Taherizadeh, S., Stankovski, V., and Grobelsnik, M. (2018) A capillary computing architecture for dynamic Internet of Things: Orchestration of microservices from edge devices to fog and cloud providers. *Sensors*.

Vaño, R., Lacalle, I., Sowiński, P., S-Julián, R., and Palau, C. E. (2023) Cloud-native workload orchestration at the edge: A deployment review and future directions. *Sensors*.

Merkel, D. (2014) Docker: Lightweight Linux containers for consistent development and deployment. *Linux J*.

Rancher Labs. (2019) K3s: Lightweight Kubernetes. Cloud Native Computing Foundation.

AGRARIAN Project. AGRARIAN Application Repository. <https://github.com/agrarian-application-repository>, accessed March 27, 2026.

Avila, J. L., Rouzegar, H., Calveras Augé, A. and Ruiz-de-Azua, J. A. (2025) Spatial-Aware MANO Systems for Efficient Mobility Management of Satellite-Based Non-Terrestrial Networks. *IEEE Open J. Commun. Soc.*, 6, doi:10.1109/OJCOMS.2025.3568892.

Nordic Semiconductor. nRF9151 SMA DK Development Kit. <https://www.nordicsemi.com/Products/Development-hardware/nRF9151-SMA-DK>, accessed March 27, 2026.