This deliverable is currently awaiting approval from the European Commission.

Project title	**DIGITAfrica** – Towards a Comprehensive Pan-African Research Infrastructure in Digital Sciences
Grant agreement #	101187966 (CSA - HORIZON-INFRA-2024-DEV-01-02)
Project duration	36 months (01/01/2025 – 31/12/2027)
Project URL	www.digitafrica.eu
Due date	30 / 06 / 2026
Submission date	29 / 06 / 2026
Dissemination level	Public
Version	1.0

D2.1 DIGITAfrica Blueprint v1

Responsible author(s): Joyce Mwangama (UCT), Damien Saucez (Inria)

Reviewers: Adam Belloum (UvA), Thomas Magedanz (TUB)

Funded by the European Union

DIGITAfrica project has received funding from the EU Horizon Europe research and innovation Programme and Switzerland under Grant Agreement No. 101187966. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Document version history

Version	Date	Inputs	Responsible
0.1	30/03/2026	Initial ToC of the document	Joyce Mwangama (UCT) and Damien Saucez (INRIA)
0.2	10/05/2026	Education	Cheikh Ahmadou Bamba Gueye (UN)
0.3	20/05/2026	Edge-AI blueprint	Nikos Makris (UTH)
0.4	20/05/2026	Use cases	Joyce Mwangama (UCT)
0.5	30/05/2026	Heterogeneous Networking Blueprint	Damien Saucez (INRIA)
0.6	05/06/2026	Pre-review version	Damien Saucez (INRIA)
0.7	08/06/2026	Executive summary	Damien Saucez (INRIA)
0.8	22/06/2026	Expanded Networking Blueprint and added diagrams	Tariro Mukute (UCT)
0.9	23/06/2026	Address reviewer comments	Damien Saucez (INRIA)
1.0	29/06/2026	Final editing	Serge Fdida (SU)

Executive summary

This document presents the first version of the DIGITAfrica Blueprint, developed through Work Package 2. The Blueprint concept is of utmost importance as it provides a proof-of-concept of our methodology and already at this early stage, a common playground both for research and education.

It combines the results of baseline service requirements analysis across five partner countries (Cameroon, Kenya, Senegal, South Africa, and Tunisia), sustainability-by-design principles developed with GreenDIGIT, and two complementary reference architectures: the Edge-Interoperable AI & ML Blueprint and the Heterogeneous Networking Blueprint.

The DIGITAfrica Blueprints are modular, open-source reference architectures that African research institutions can adopt and adapt according to their needs and resources. Built from reusable baseline services, they provide a flexible foundation for developing sovereign and sustainable digital infrastructure. Their development follows an iterative, community-driven process that incorporates stakeholder feedback and continuous validation.

The deployment strategy follows a five-tier model, where each tier represents increasing levels of scale, capability, coordination, and investment. Tier 0 targets highly resource-constrained rural environments, while Tier 1 supports campus laboratories and educational institutions. Tier 2 introduces regional hubs that aggregate services and resources across multiple sites, Tier 3 enables national-scale coordination and governance, and Tier 4 establishes a pan-African federation supporting cross-border collaboration and continent-wide services. Technologies remain compatible across all tiers, ensuring a clear and sustainable upgrade path. Partners can adopt the tier most appropriate to their context and may deploy multiple blueprint instances across different tiers to address diverse requirements.

Although each blueprint addresses distinct objectives, all are built upon a common architectural foundation. DIGITAfrica defines five categories of shared services that provide the core capabilities required across all deployments: resource pools, identity management, connectivity, user interfaces, and data services. The resource pool layer relies on containerised workloads based on Docker and Kubernetes-compatible orchestration platforms, providing a consistent mechanism for deploying and managing applications from lightweight edge devices to regional and national infrastructures. Identity management is built around Keycloak and OpenID Connect (OIDC), enabling federated authentication, authorization, and secure access control across institutions and deployment tiers. Connectivity services provide the networking foundation required to interconnect distributed sites and services. User access is primarily delivered through JupyterHub, which offers a common, browser-based environment for research, experimentation, education, and service interaction. Data services ensure effective data management, governance, quality assurance, and lifecycle control. Together, these common services promote interoperability, operational consistency, and technology reuse across all blueprints.

The Edge-AI Blueprint provides a framework for deploying artificial intelligence and machine learning services across distributed edge and cloud environments. It supports edge-to-cloud orchestration, federated learning, data management, and federated identity services through a five-tier deployment model ranging from low-cost offline edge nodes to pan-African AI federations. The architecture enables AI applications in digital health, precision agriculture, and education while promoting privacy, reproducibility, and local data processing.

The Heterogeneous Networking Blueprint delivers a low-cost, progressively deployable networking infrastructure that integrates technologies such as 5G, WiFi 6/6E, LoRaWAN, and satellite connectivity. Built on open-source network functions, it combines local edge processing with centralised services through policy-driven traffic management. The blueprint supports resilient connectivity for research and education use cases, including telemedicine, smart agriculture, and digital learning environments.

Both blueprints address key African infrastructure challenges, including limited resources, unreliable power, variable connectivity, and diverse levels of operational expertise. Sustainability, energy efficiency, data sovereignty, and privacy are embedded throughout the architecture, providing a practical framework for building resilient, scalable, and locally governed digital research infrastructure across Africa.

A first implementation of both blueprints has already been completed and is publicly available through the DIGITAfrica Git repository. Beyond their architectural definition, the blueprints have been validated through practical deployment and hands-on training activities. They were successfully used during the DIGITAfrica Winter School in Nairobi and the DIGITAfrica project meeting in Tunis, where more than 60 participants gained direct experience deploying, operating, and experimenting with the infrastructure components. These early training and validation activities demonstrate the practicality, accessibility, and maturity of the proposed architectures while providing valuable feedback for future iterations of the DIGITAfrica infrastructure to happen in Work Package 3.

Table of contents

List of tables

List of figures

Abbreviations

Abbreviation	Definition
CAPEX / OPEX	Capital Expenditure / Operational Expenditure
CNF	Container Network Function
CSI	Container Storage Interface
CUE	Carbon Usage Effectiveness
GPU	Graphics Processing Unit
FAIR	Findable, Accessible, Interoperable, Reusable
JWT	JSON Web Token
k8s / K3s	Kubernetes / lightweight Kubernetes distribution
LDAP	Lightweight Directory Access Protocol
LoRaWAN	Long Range Wide Area Network
MEC	Multi-access Edge Computing
ML	Machine Learning
NFV	Network Function Virtualisation
NPU	Neural Processing Unit
NR	New Radio (5G air interface)
NREN	National Research and Education Network
OAI	OpenAirInterface
OIDC	OpenID Connect
O-RAN	Open Radio Access Network
PUE	Power Usage Effectiveness
PVC	Persistent Volume Claim
RBAC	Role-Based Access Control
RI	Research Infrastructure
SDN	Software-Defined Networking
SDR	Software-Defined Radio
TVET	Technical Vocational and Education and Training
UPF	User Plane Function
VNF	Virtual Network Function

1 Introduction

1.1 Context

The African Union's Digital Transformation Strategy for Africa (DTSA) 2020–2039 sets out an ambitious agenda to build a Digital Single Market, expand universal connectivity, and digitize the agriculture, health, and education sectors across the continent. Despite significant progress in national research and education network (NREN) development and growing ICT ecosystems in several partner countries, major infrastructure gaps persist — particularly in rural connectivity, access to high-performance computing, edge infrastructure, and AI tooling. The DIGITAfrica project (Horizon Europe Grant Agreement No. 101187966) is a Coordination and Support Action tasked with laying the foundations for a pan-African comprehensive Research Infrastructure (RI) in Digital Sciences, addressing these gaps through collaborative blueprint design, pilot deployments, and capacity building.

This deliverable, D2.1, presents the first version of the DIGITAfrica Blueprint. It synthesises the outputs of three tasks carried out within Work Package 2: Task 2.1 (Baseline Service Requirements), Task 2.2 (Synergies with the GreenDIGIT sustainability framework), and Task 2.3 (Blueprint Design). Together, these tasks produced a consolidated picture of the service requirements across five partner countries — South Africa, Kenya, Senegal, Cameroon, and Tunisia — a set of sustainability-by-design principles informed by GreenDIGIT, and two complementary technical blueprints: the Edge-Interoperable AI & ML Blueprint and the Heterogeneous Networking Blueprint.

1.2 What are blueprints?

Blueprints are modular, open-source reference architecture that any African research institution can adopt, adapt, and deploy — at their own pace, within their own resource constraints. Blueprints are not products to buy. They are community-owned specifications for building sovereign, sustainable digital infrastructure.

Blueprints are **composable**, i.e., built from reusable baseline services where sites mix and match what they need.

Blueprint are based on the concept of "thought experiments", a type of experiment that Schrödinger considered impossible to realize (The Photon box). What matters in DIGITAfrica is not the infrastructure but the thought experiments and the support of the full research life cycle, including FAIR data and reproducibility.

It is impossible for DIGITAfrica to be exhaustive, neither we pretend to be able to identify these thought experiments. We call for the research community to suggest how an infrastructure and its data sets can be deployed in DIGITAfrica for that purpose.

The key for the success of an ambitious projects such as DIGITAfrica is to take enough time, effort, and rigour to set the vision, requirements, and culture.

In DIGITAfrica, we leverage the concept of blueprint to ease collaboration between engineers and non-engineers, researchers and practitioners. As such, we define a common terminology that doesn't require in-depth, technical knowledge to produce consistent vision on the application with a focus on the future objective. But as the devil is in the details, we accompany the blueprints with a set of reference implementations.

Blueprints are documented and discussed with the community and built iteratively. In each cycle, we pick up the appropriate actors to review and validate the advances made so far and check their adequacy with the baseline. Each iteration is used to refine the definition of the DIGITAfrica infrastructure.

Finding the right actors in the cycles is essential and this is where the blueprint and its multi-level of readings is an essential tool. Indeed, in every cycle we involved both the management and engineers of the platform community but also researchers from communities that are supposed to be supported by DIGITAfrica. As a consequence, the blueprint is always aligned with the actual needs of researchers and can put priority on what is really essential to the targeted researchers.

Each iteration being a refinement of the previous one, we progress in parallel with blueprint components. At the current stage, we are iterating on the *Edge-AI blueprint* and the *Heterogeneous Networking blueprint* and their components.

Concretely, a DIGITAfrica blueprint is not a single piece of software but a versioned, open-source specification plus a matching reference implementation. The specification fixes which common services are composed, how they are chained, and how the composition is specialised per tier; the reference implementation realises that specification as a git repository containing deployment automation (Ansible playbooks, docker-compose files, and kubernetes/k3s manifests) that an institution can clone and run. In other words, where a common service is a single reusable building block (for example, Keycloak for identity or MinIO[1] for object storage), a blueprint is a documented assembly of several such building blocks, configured and orchestrated to deliver an end-to-end capability for a given application domain.

As a concrete example, the Edge-AI Blueprint is, at Tier-1, the composition of the cluster service (a multi-node k3s cluster), the notebook service (JupyterHub), the user-portal service (Keycloak/OIDC), and the storage service (NFS and MinIO), chained together so that an authenticated user can open a browser, land in an isolated notebook environment, and run a reproducible AI workflow against shared datasets. This composition is delivered as the blueprint repository at the Edge-AI blueprint repository URL[2]. The scope of a blueprint is therefore bounded: it covers the services it composes, the tiers it targets, and the use cases it is validated against — and explicitly excludes the application-specific "secret sauce" (the actual AI models or network experiments) that researchers build on top of it.

[1] https://www.min.io/
[2] https://gitlab.inria.fr/digitafrica/blueprints/services/heterogeneous-networking-blueprint

It helps to situate blueprints within the four layers that DIGITAfrica deployments are built from. At the base sits the infrastructure (servers, radio units, and other hardware). On top of this run the common services (Section 3.2) — kubernetes/K3s, identity management, JupyterHub, storage — delivered as Ansible playbooks. A blueprint is the next layer up: a composition of these services configured to deliver an end-to-end capability, such as a deployment that brings up a generic 5G core network or a set of AI/ML containers. Finally, a use case selects and combines blueprints to meet a concrete need — for example, an e-health deployment that brings up a network, adds authentication, and then runs a domain-specific service on top.

A blueprint is therefore more than a set of notebooks. Where notebooks provide the user-facing entry point, a blueprint also includes the supporting infrastructure that those notebooks rely on — for instance the docker-in-docker mechanism that lets a notebook spawn its own containers, or an MLflow service backing an AI workflow. The notebook is how a user drives the blueprint; the blueprint is the whole assembly of services and supporting infrastructure behind it.

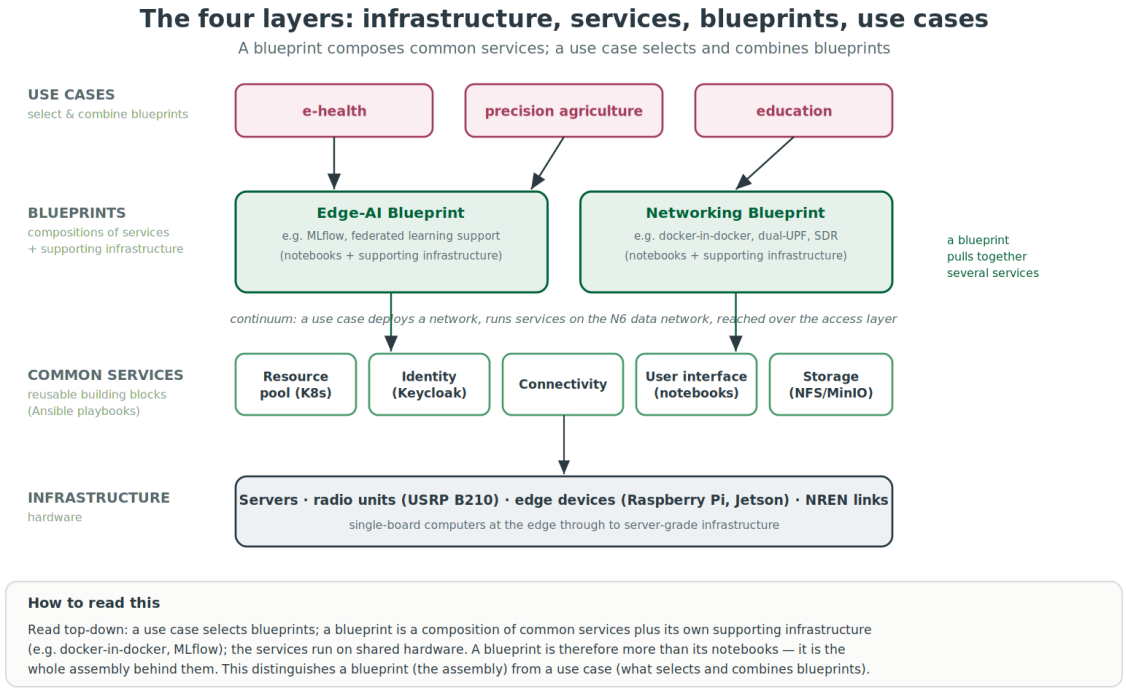

Figure 1 - The four-layer model — infrastructure, common services, blueprints, and use cases. A blueprint composes common services plus its own supporting infrastructure; a use case selects and combines blueprints.

1.3 Specificities of DIGITAfrica

DIGITAfrica blueprints are shaped by a specific set of constraints and opportunities that distinguish them from comparable European research infrastructure blueprints. These specificities must be explicitly acknowledged in both the architecture and the implementation choices.

First, resource constraints are real and heterogeneous. Power infrastructure is often unreliable, requiring solar and battery backup at lower-tier deployments. Network connectivity ranges from high-speed NREN links at national institutions to intermittent satellite or LTE in rural areas. Hardware budgets are tightly limited, necessitating ultra-low-cost computing platforms such as Raspberry Pi and Nvidia Jetson devices at Tiers 0 and 1, while also ensuring a clear upgrade path to standard server-grade infrastructure at higher tiers.

Second, operational expertise varies widely across partner institutions. The blueprints must therefore rely exclusively on open-source, well-documented technologies and must be deployable without requiring highly specialised infrastructure skills. Automation through Ansible playbooks and GitOps-style configuration management is a core requirement, not an optional feature.

Third, the dual mandate of research and education is central. The blueprints must serve both advanced research experiments — such as federated learning, Open RAN, and IoT-based precision agriculture — and teaching environments accessible to students on smartphones with limited bandwidth. JupyterHub-based notebook environments serve as the common interface across both use cases.

Fourth, sustainability is not optional. In contexts where power costs are high and energy access is uncertain, infrastructure must be designed to minimise energy consumption, support renewable power sources, and quantify its environmental footprint. The GreenDIGIT synergy work from T2.2 provides a concrete methodology for embedding these principles into the blueprint architecture, including Power Usage Effectiveness (PUE) and Carbon Usage Effectiveness (CUE) tracking, lifecycle assessment, and energy-aware workload orchestration.

Fifth, data sovereignty and privacy protection are paramount, particularly for use cases in digital health and agriculture. The blueprint designs ensure that sensitive data can be processed locally at the edge without requiring centralisation, and that federated learning and FAIR data principles are supported from the lowest tiers of deployment.

DIGITAfrica blueprints are shaped by a specific set of constraints and opportunities that distinguish them from comparable European research infrastructure blueprints. These specificities must be explicitly acknowledged in both the architecture and the implementation choices.

1.4 Education

The African Union's visionary 2020-2030 Digital Transformation Strategy for Africa (DTSA)[3] includes plans to create a Digital Single Market, expand universal access to basic connectivity, improve the enabling environment, and digitize the agriculture, health, and education sectors. The overarching goal of DIGITAfrica is to lay the foundations for a pan-African comprehensive Research Infrastructure in Digital Sciences. DIGITAfrica needs to effectively identify priority digital skill needs for developing micro-credits particularly for young graduates and women. Therefore, we conducted parallel surveys of Professors/Lecturers and Students in Cameroon, Kenya, Senegal, South Africa, and Tunisia to figure out needed digital skills for university programs.

This survey was carried out from 15th September 2025 to 20th November 2025. The survey was conducted using Google forms; participants were contacted by emails sent to mailing lists of universities. More than 1500 students and more than 200 professors were contacted; they were invited to access a link to the Google form. Data from African universities shows that while foundational digital literacy is strong, there are critical shortages in the advanced skills needed for the digital economy. Both groups highlighted several consistent insights such as skills gaps are concentrated in advanced digital domains, curricula are not fully aligned with industry needs, structural barriers limit both teaching and learning, and strong demand for micro-Credentials. As results, students and staff agree these are the highest-priority areas for training and micro-credentials in AI/ML, data, cybersecurity, cloud, and robust programming. Additionally, a lack of practical, hands-on exposure limits employability.

To address the identified gaps in advanced digital domains, it is recommended to prioritize the systematic integration of these advanced digital skills into university curricula through targeted, practice-oriented interventions. Specifically, African higher education institutions should:

- Strengthen modular training and micro-credentials in AI/ML, Data Science, Cybersecurity, Cloud Computing, and modern programming frameworks, aligned with industry standards.
- Leverage SLICES-RI and SoBigData outcomes, to provide hands-on, experimental, and research-driven learning experiences.
- Enhance practical exposure through laboratories, testbeds, project-based learning, and industry co-designed assignments.
- Support faculty upskilling to ensure sustainable delivery of advanced digital skills.

Implementing these measures will contribute to improving curriculum relevance, strengthening employability, and aligning higher education outcomes with the needs of the digital economy. Therefore, both provided blueprints promote education and skills

[3] Website: https://au.int/en/documents/20200518/digital-transformation-strategy-africa-2020-2030 (Accessed: 02/05/2026)

development. For instance, by leveraging the Heterogeneous Networking Blueprint, a lightweight cloud/edge educational laboratory with CI/CD automation is currently being deployed. The platform, designed for educational service delivery, is intended to be efficiently accessible from students' smartphones while minimizing bandwidth consumption and ensuring usability in resource-constrained environments. Tightly articulating the Blueprints and the education components is vital in the methodology developed in DIGITAfrica.

2 Blueprints

2.1 Edge-AI Blueprint

The Edge-Interoperable AI & ML Blueprint defines an experimental framework for deploying and reproducing artificial intelligence and machine learning workflows across distributed edge and cloud infrastructures. It enables edge-to-cloud orchestration, allowing AI services to run close to data sources — such as sensors, clinics, or local innovation hubs — while supporting privacy protection and technical reproducibility. The blueprint is designed around four main pillars: edge orchestration and compute management, edge AI capabilities (training, inference, and federated learning), storage and data integration, and user access and authentication via federated identity.

The blueprint implements a five-tier deployment model (Tiers 0–4) ranging from fully offline solar-powered micro-edge pods (US$1,000–1,500) to a pan-African research federation. At Tier 0, lightweight single-board computers support local inference and small-scale training with AI accelerators. At Tier 1, multi-node K3s clusters[4] provide multi-user JupyterHub environments for campus-level experimentation. At Tier 2, regional hubs run full end-to-end ML pipelines with MLflow model registries and S3-compatible storage. Tiers 3 and 4 enable national GPU clusters and cross-border federated AI. The target is to reach at least Tier 2 within the project lifetime, with further development beyond 2028. Representative use cases span digital health (AI-assisted telemedicine and federated diagnostics in South Africa), precision agriculture (IoT-based crop monitoring in Kenya), and education labs (CI/CD-automated Jupyter workflows in Senegal).

2.2 Heterogeneous Networking Blueprint

The DIGITAfrica Heterogeneous Networking Blueprint defines a powerful yet low-cost, progressively deployable networking and computing infrastructure tailored for African research and education. It explicitly addresses the ten functional service requirements identified by DIGITAfrica partners and is inspired by the SLICES 5G/Post-5G blueprints and the GreenDIGIT sustainability framework. The blueprint combines heterogeneous access technologies — 5G NR where available, WiFi 6/6E, LoRaWAN for wide-area IoT, and satellite

[4] A K3s cluster is a lightweight Kubernetes cluster built using K3s, designed to run containerized applications with reduced overhead compared to standard Kubernetes.

or microwave backhaul for resilience — into a unified, standards-based architecture built on open-source containerised network functions.

A key architectural innovation is dual User Plane Function (UPF) anchoring: a local Edge UPF provides low-latency breakout for latency-sensitive services, while a Central UPF handles cloud services and inter-site traffic. Policy-based dynamic traffic steering selects the appropriate path per service. As with the Edge-AI Blueprint, the Heterogeneous Networking Blueprint follows the same five-tier model, from offline micro-edge pods with WiFi and LoRaWAN (Tier 0, US$500–1,500) to a pan-African federated research infrastructure (Tier 4). Representative use cases include resilient telemedicine at district hospitals and rural clinics (South Africa), IoT-enabled precision agriculture with LoRaWAN sensor networks (Kenya), and offline-first digital learning environments (Senegal). The goal is to reach at least Tier 2 by project end.

2.3 Support for education

Most of African universities face significant challenges in conducting large-scale research in telecommunications networks and artificial intelligence. There's a shortage of specialists in AI and advanced telecom systems, and many qualified professionals emigrate. Furthermore, we observe insufficient collaboration between academia, industry, and government, as well as limited access to high-quality datasets and contemporary curricula. One of the key objectives of both blueprints is to establish a high-quality technical vocational and education and training (TVET) system that is accessible, inclusive, and aligned with labor market needs. Among the expected outcomes are: *(i)* the reinforcement of apprenticeship training and distance learning; *(ii)* curriculum reform in higher education; and *(iii)* enhanced cooperation, mobility, and research collaboration among universities, public institutions, the private sector, and research centres.

The proposed lightweight blueprints backend, in conjunction with JupyterHub, facilitates multi-user access to notebooks. This enables students to engage in interactive computing environments for hands-on experimentation in networking and artificial intelligence, eliminating the need for local installation. Furthermore, by integrating JupyterHub with k3s, we establish a flexible and sustainable architecture that can accommodate distributed workloads, such as networking experiments and AI pipelines. This architecture ensures accessibility for users with limited bandwidth and heterogeneous devices like smartphone.

For instance, k3s and JupyterHub offer a comprehensive suite of capabilities, enabling users to:

- Conduct networking lab exercises, including simulations of Software-Defined Networking (SDN)/ Network Function Virtualisation (NFV) and cloud environments.
- Execute AI models and develop distributed architectures.
- Test federated learning scenarios.

These capabilities are accessible without the need for an expensive laboratory or a stable high-speed internet connection. Both designed and deployed blueprints highlight the

potential of lightweight, cloud-native infrastructures to democratize access to advanced digital skills and support scalable, context-aware education in the Global South. They are also valued and exploited in summer classes and hands-on delivered initially within partners' premises but open to be duplicated elsewhere (outside the DIGITAfrica consortium).

3 Architecture

The architecture is designed with modularity at its core, allowing each deployment to determine the specific features it needs to support. This approach provides flexibility and adaptability, ensuring that different operational environments can implement only the components that are relevant to their requirements, without introducing unnecessary complexity.

A common interface for interacting with the blueprints is provided through notebooks. This choice is driven by their accessibility and ease of adoption, as they are widely used and can be quickly learned by users with varying levels of expertise. Notebooks require only a web browser to operate, placing minimal demands on client-side resources while still offering an interactive and flexible environment for development, experimentation, and deployment. In addition, this approach ensures that users benefit from a consistent toolset environment across all notebooks, eliminating the need to learn or adapt to different tools for each use case. The decision to use notebooks stems from the observation that this tool is already widely adopted by virtually all partners for teaching and for their classes.

The blueprints are tightly coupled and conceived as compositions of **common services** [*see Sec.* 3.2], which act as foundational building blocks across the system. These shared services ensure consistency and reusability, while also enabling the architecture to address the specific requirements of both *Edge AI* and *heterogeneous networking* blueprints. As a result, the system achieves a balance between standardization and specialization.

In addition, the architecture follows a **tiered design** approach (see Section 3.1), where each of the 5 tiers introduces progressively higher levels of complexity and capability. This enables incremental development and deployment, allowing systems to evolve over time while maintaining clarity in the separation of concerns across different layers.

This combination of modularity and tiered structuring makes the architecture particularly well-suited for the integration of vertical use cases. Different partners, industries or application domains can adopt and extend the system according to their specific needs, leveraging only the relevant tiers and services.

Sustainability is also embedded into the design from the outset. By enabling selective feature adoption and incremental scaling, the architecture minimizes unnecessary resource usage and supports efficient operation. This built-in sustainability contributes to long-term maintainability and adaptability of the system.

3.1 Multi-tier deployments

The deployment strategy follows a **five-tier model**, where each tier represents an increasing level of scale, capability, coordination, and CAPEX/OPEX. Regardless of the level of sophistication of the tier, each tier supports a, more or less advanced, version of every blueprint.

Tier-0 - Rural - corresponds to highly resource-constrained environments such as rural schools or clinics. At this level, deployments are lightweight and focused on essential services, often operating with limited connectivity and computing resources. The emphasis is on accessibility, low power consumption, and the ability to function in isolated or intermittently connected settings. Its lightweight design allows it to be deployed as strictly virtual entities and makes no assumption on the hardware.

Tier-1 - Campus Lab - represents campus-level environments, such as laboratories or educational institutions. Here, more computational resources and stable connectivity are available, enabling experimentation, development, and local validation of services. This tier often acts as a testing ground for solutions before they are scaled further. At this stage, specific hardware may be considered though the focus remains on the ability to be deployed with very limited budget and operational skills.

Tier-2 - Regional hub - expands to a regional hub, where multiple Tier-1 sites can be interconnected. At this level, the infrastructure supports aggregation, coordination, and more advanced data processing capabilities. Regional hubs can provide shared services, improve resource utilization, and enable collaboration across several local deployments. As tier-2 covers a larger community, more specific hardware can be considered and one may consider that the operators have a higher skill level than for the lower tiers.

Tier-3 - National hub - operates at the national level, introducing centralized governance, broader data integration, and large-scale service orchestration. This tier ensures consistency, policy enforcement, and interoperability across regions, while supporting nationwide applications and infrastructure management. At tier-3, it is important to have strong governance and highly skilled operators given the very large span. This requires continuous training of persons involved in operating the national hub.

Tier-4 Pan-African - represents a pan-African federation, where multiple national systems are interconnected. This highest tier enables cross-border collaboration, international knowledge sharing, and large-scale data exchange, fostering a unified ecosystem that can support continent-wide initiatives, innovation, and resilience. This tier implies sustainable financial and operational resources with long term engagement.

To facilitate seamless scaling from one tier to the next, the technologies used at each level are designed to remain fully compatible with those of the higher tiers. This ensures that the efforts invested in implementation and operation at a given tier can be directly leveraged as the system evolves upward.

Each partner has the flexibility to choose the tier level that best matches its needs, constraints, and objectives. This allows deployments to be tailored to specific contexts, whether focusing on lightweight local setups or more advanced, large-scale infrastructures. Furthermore, a partner may deploy multiple instances of the blueprints, either at the same tier level or across different tiers, depending on the diversity of use cases and environments it aims to support.

3.2 Common services

Each blueprint targets its own specific and independent objectives, but the DIGITAfrica infrastructure architecture is designed to ensure a shared foundation across all of them. Although the usage, goals, and target of the blueprints may differ, the underlying infrastructure—covering hardware, software, and knowledge components—remains mostly common. Each blueprint can therefore be seen as a specialized adaptation of the same core architecture, tailored to meet its particular requirements.

This is done thanks to the definition of the common services. Within DIGITAfrica, we have identified the following 5 categories of services required to implement any blueprint. This decomposition is based on studies conducted within DIGITAfrica and on expertise gained through GreenDIGIT[5] and SLICES-RI[6].

- **Resource pool:** is designed to enable DIGITAfrica to provide users with access to both virtual and physical resources, packaged in a way that is meaningful, accessible, and aligned with the objectives of each blueprint. To achieve this, the available resources must be effectively managed and orchestrated across multiple users, ensuring fair and efficient allocation. This creates the need for a dedicated service responsible for managing the resource pool. In practice, this is implemented through clusters that absorb and execute the workloads associated with the blueprints.
- **Identity management:** is a critical component of the architecture, ensuring that users, services, and systems can be securely and consistently identified across all tiers of deployment. It provides a unified mechanism for authentication, authorization, and access control, enabling trusted interactions between distributed resources and blueprints. By maintaining a coherent identity framework, the system ensures that access to data and services is properly governed, regardless of where a user or workload is located within the infrastructure. This fundamental capability ensures security, accountability, and seamless interoperability across the entire DIGITAfrica ecosystem.
- **Connectivity:** is a fundamental requirement of the system, as it enables communication between distributed components, resources, and users across all tiers of deployment. It ensures that data, services, and workloads can be exchanged reliably between local environments, regional hubs, national infrastructures, and the broader federation. Without robust and adaptable connectivity, the coordination of

[5] https://greendigit-project.eu/
[6] https://slices-ri.eu/

blueprints, resource pooling, and orchestration of workloads would not be possible at scale. Therefore, connectivity acts as the essential backbone that links all parts of the architecture into a coherent and functional whole.

- **User interface:** is an essential component of the system, as it provides the primary point of interaction between users and the underlying blueprints and services. It must be designed to be intuitive, easy to learn, and accessible to users with varying levels of technical expertise. By reducing complexity and presenting functionalities in a clear and consistent manner, the user interface enables faster adoption and more effective use of the platform. This ease of use is particularly important in diverse deployment contexts, where users may have limited digital training or experience.
- **Data:** is a key pillar of DIGITAfrica, strong data management solutions guarantee data quality, governance, and lifecycle control, ensuring that information remains accurate and relevant over time.

The concrete implementation of the common services is described in Section 4 and illustrated in Figure 2. Given the specific context of DIGITAfrica (see Section 1.1), these services must be designed and implemented to rely exclusively on open-source and lightweight solutions, leveraging widely adopted and documented technologies – remember that we are talking about the common services, not the actual blueprint specific secret sauce. It is also essential that these services can be deployed, operated, and modified without requiring highly specialized expertise, ensuring ease of adoption and long-term sustainability across all deployment environments.

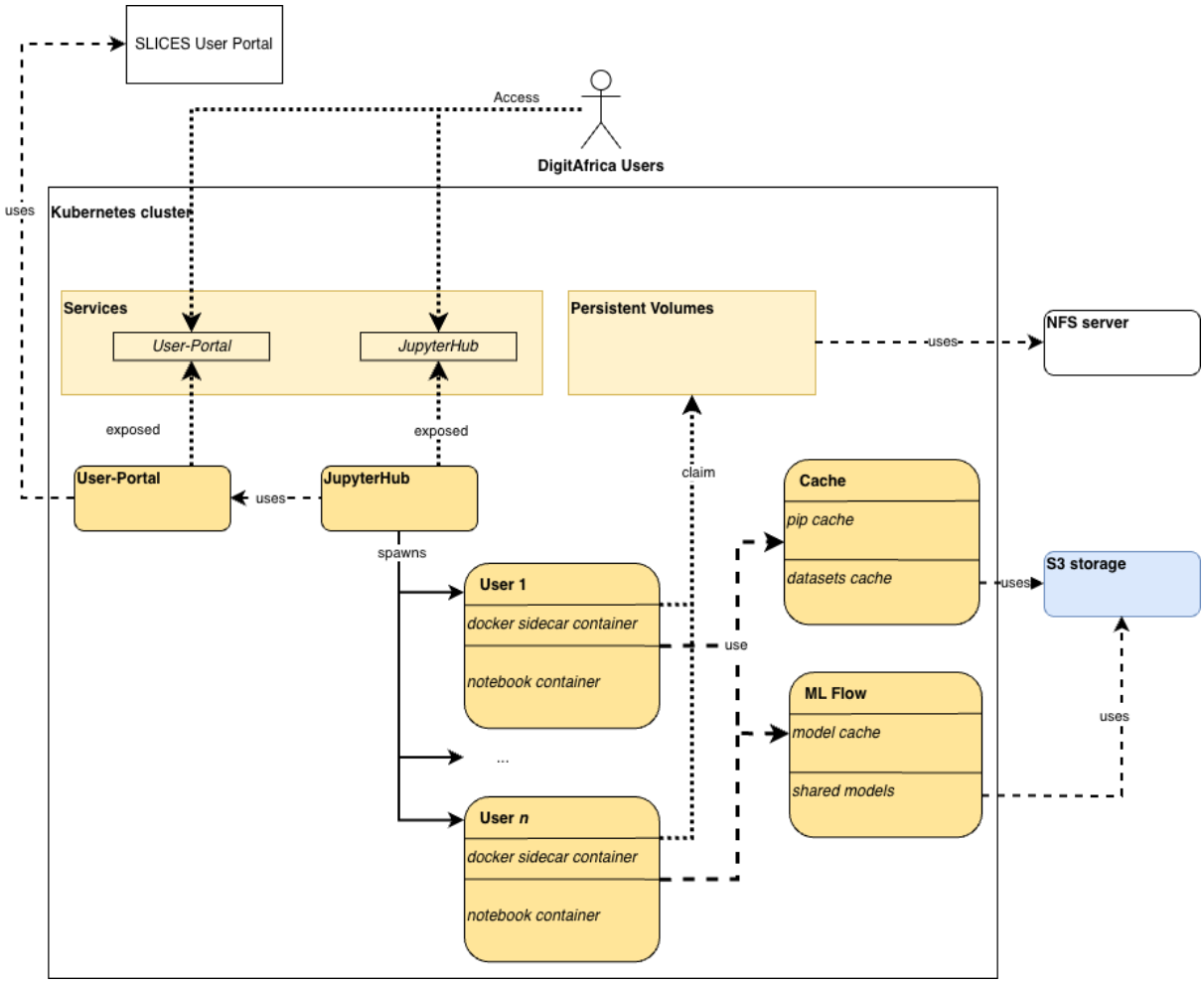

Figure 2 - Overall blueprints integration architecture

3.2.1 Cluster

The basic substrate to support the two blueprints is the establishment of a cluster service. The cluster should support different deployment scales, in line with the tiered model introduced in Section 3.1. As nodes evolve, lower tiers should be able to integrate with the higher ones, while also services deployed at each cluster can be reused across the different blueprints (e.g., AI/ML workloads from the Edge-AI blueprint can be reused for different purposes in the heterogeneous networking blueprint). At the lowest levels, the cluster may correspond to a single node or a lightweight local deployment, suitable for constrained environments such as rural schools, laboratories, or small institutional settings. The cluster should be able to progressively evolve, by adding more infrastructure elements or more compute infrastructure to higher tiers. At higher tiers, the cluster should be able to evolve into a multi-node environment capable of hosting several users (in the order of thousands) and services simultaneously. As such, this makes the cluster service a key enabler of the incremental deployments in DIGITAfrica.

To support this functionality and incremental updates, the cluster service is designed around Kubernetes-compatible deployment models. This provides the flexibility to host all the services using Container Network Functions (CNFs) or Virtual Network Functions (VNFs), e.g., VMs deployed with extensions like KubeVirt[7] and handled seamlessly within the cluster manager. Depending on the computational capabilities of each site and the tier that it implements, the infrastructure may rely either on a full Kubernetes or on a lightweight k3s distribution. K3s is suitable for computationally constrained infrastructure, usually residing in edge environments, with devices that can be even battery powered. Both deployments support the same API for deploying/managing services/workloads on the infrastructure. Therefore, minimal handling for the cluster type needs to be implemented for the services that implement each blueprint. Full Kubernetes deployments may be considered for larger sites requiring more advanced operational capabilities, integration with existing infrastructure (e.g., NRENs), or higher levels of scalability/high availability. In all cases, the adoption of such a solution allows the use of a common orchestration model across tiers.

The current implementation direction follows this principle. For example, in the case of the Edge-AI blueprint, Tier-0 can rely on a single-node deployment with Jupyter notebooks and basic monitoring components, while Tier-1 can rely on a multi-node k3s cluster hosting JupyterHub and supporting authenticated users. In the case of Heterogeneous Networking blueprint, the current implementation focuses on a Tier-1 environment, where the cluster service provides the Kubernetes-compatible execution layer on which JupyterHub, and the networking notebooks are deployed. For both blueprints, JupyterHub is implemented as the same service, using the same manifests.

3.2.2 User-portal

The user-portal service is responsible for managing user access and accountability through authentication, identity management, and access control. It serves as the primary entry point for accessing any service within DIGITAfrica.

Given the constraints of DIGITAfrica, the user portal must operate reliably in highly constrained environments and remain resilient to interruptions and connectivity losses. It must also avoid becoming a single point of failure.

In addition, partners may enforce different user management policies and accountability requirements. Therefore, the user portal must support local user management capabilities. At the same time, access to the research infrastructure should remain seamless for users. This implies that the user portal must be composed of local instances that collectively operate as a single virtual service.

The user portal follows the same tiered approach as the rest of the infrastructure. Local user portals can be deployed independently and interconnected with portals at the same or higher tiers to form a federated identity and authentication service. This federation is

[7] https://kubevirt.io/

implemented using OpenID Connect (OIDC), where each local or regional user portal is linked to the others through OIDC-based delegation.

The OpenID Connect protocol enables the various common services to delegate user authentication to an identity provider. Services receive the information required to establish user sessions and enforce access policies from the OIDC server through authentication tokens. These tokens are subsequently used when invoking APIs across the different services, eliminating the need for users to authenticate repeatedly. In the blueprint architecture, this functionality is provided consistently across all blueprints through the user-portal service.

While OIDC is mandatory for interconnecting user portals, local deployments remain free to use the identity management technologies best suited to their environment. For example, at Tier-0, the identity provider may simply consist of a local user database, whereas university campuses (Tier-1) typically rely on LDAP services for user identity management and authentication.

The user-portal service is implemented using Keycloak, a highly scalable, cloud-native solution that offers extensive flexibility while providing an administration interface that does not require highly specialized personnel.

Furthermore, to providing user-friendly graphical interfaces for accessing resources, the user portal must also support fully automated machine-to-machine workflows, enabling reproducible research and large-scale experimentation.

3.2.3 Storage

The storage service provides the data layer required by the services and blueprints implemented for DIGITAfrica. The storage service needs to support persistence (in some cases), sharing, and exchange of data across users, services, and workloads, and integrate seamlessly with the cluster orchestrator used. Since both blueprints require access to datasets and models, and provide as outputs experiment results, or training material, storage is treated as a common basic architectural service rather than as a blueprint-specific component.

The storage service covers two complementary needs. The first is a shared filesystem storage, which enables multiple nodes or services within a cluster to access common data. This is necessary, for example, when workloads, such as notebook environments, need persistent user directories or when services deployed across nodes require access to shared volumes (e.g., a common dataset for a training process). As such, a solution based on an NFS server is implemented, in order to addresses this need by providing centralized shared storage and persistent volumes for services running on the infrastructure. NFS was selected as a solution as it integrates seamlessly with the cluster service.

The second need is a (long term) object storage, which provides well defined interfaces (e.g., S3-based) for pushing and pulling data objects. This is particularly useful for datasets,

ML models, experimental results, and other artefacts. For such an implementation, the well-established MinIO solution is selected, as it also provides the capability to integrate with OIDC based solutions for authentication.

In addition to persistent storage, caching services may be used to improve efficiency and reduce repeated downloads or transfers, something typical in rural areas where several of the DIGITAfrica use cases will be executed. An S3 cache component is implemented, intended to support caching for examples such as installation packages and dataset storage.

The storage service is used by both blueprints. Edge-AI services use it to store datasets (in the S3 storage), trained models (in ML-Flow that integrates with the S3 storage), or inference results (NFS storage across workloads). The Heterogeneous Networking blueprint uses storage services for pushing experiment traces (S3-storage), configurations, or measurements and share them across CNFs within a cluster (NFS based storage).

3.2.4 Notebooks

In order to provide an intuitive user interface to the end users, notebooks are used, and bundled as a service deployed within the blueprints. The notebook service offers a web-based interface through which users can access tools, execute code, run experiments, process data, and interact with the services deployed on the underlying infrastructure. The notebook service integrates with the prior services, as follows: 1) It can be deployed within any cluster, by using manifests such as Jupyterhub, 2) It is integrated with the user portal for authenticating/isolating user workspaces, 3) Can integrate with the NFS/S3-based storage components. The users can therefore use a single browser to interface with the infrastructure and the deployed blueprint. This is also very convenient for education and training, as notebooks provide predefined workflows, and allow users to modify them, and progressively adapt them to their own use cases.

At the architectural level, the notebook service resides within the cluster, and interfaces the identity, and storage services. The cluster provides the execution environment, and allows exposing of infrastructure parameters that are relative to each blueprint (e.g. Software Defined Radios for the Heterogeneous Networking blueprint, GPUs/NPUs for the Edge-AI blueprint). The user-portal provides authentication using OIDC, and the storage service provides sharing of the datasets used for training or results produced over the blueprint. This makes the notebook service the primary interface through which users consume the common services and execute blueprint-specific workflows. Extensions to the service are done through the implementation of other supporting services, tailored to the needs of each blueprint. For example, the integration of notebook access with docker-in-docker implementation is needed for the case of the Heterogeneous Networking blueprint, to allow spawning new containers through the notebook.

3.3 Edge-AI Blueprint

The Edge-AI Blueprint defines the chaining of different DIGITAfrica services that enables deploying and reproducing artificial intelligence and machine learning workflows across distributed edge and cloud infrastructures. Its target is to enable AI services to run close to the places where data is produced and used according to the defined use cases, such as sensors, clinics, farms, schools, laboratories, while still allowing integration and exchanges with higher-tier infrastructures when connectivity and governance allow it.

This blueprint is particularly important as in the DIGITAfrica context many AI infrastructures depend on centralized or externally hosted cloud resources. The blueprint therefore enables local processing, autonomous operation, technical reproducibility, and interoperability between institutions and regions, and privacy protection for the data that are used within the hosted workloads. A local installation should be able to operate independently in disconnected or low-connectivity conditions, while also being able to participate in a wider federation through lightweight APIs and federated identity mechanisms.

3.3.1 Blueprint tiering

The blueprint follows the five-tier deployment model introduced in Section 3.1 and specializes each tier for AI/ML workloads. The tiered approach allows each site to gradually develop its services/deployment, depending on the local infrastructure availability. A deployment can start from a lightweight edge installation using resource constrained devices (e.g., Raspberry PIs) with very basic local training and later evolve towards regional, national, or Pan-African infrastructures supporting larger workloads without major changes to the overall architecture. The following table summarizes the role of each tier.

Table 1 -Tier breakdown for the Edge-AI Blueprint

Tier	Example Location	Capabilities	Budget
0	Rural school	Offline services + caching + solar, local training (limited) and inference, AI accelerator for local training	$1–1.5k
1	Campus lab	Edge AI & inference, AI accelerator for local workloads, multi-user access, can locally integrate tier-0 nodes	$2–5k
2	Regional hub	Distributed AI workloads, model lifecycle management, Local storage, redundant backhaul	$15–30k

3	National	Large scale AI/ML research, GPU cluster, CI/CD, model lifecycle, high-speed links through NRENs	$100k+
4	Pan-African	Federation across NRENs	Shared

This tiered model allows the blueprint to address both constrained and advanced pan-African deployment environments. At Tier-0, the emphasis is on local autonomy, low power consumption, and the ability to operate with intermittent or no connectivity. At Tier-1, the blueprint supports teaching, experimentation, and local AI development through multi-user notebook environments, with the implementation of Role Based Access Control (RBAC). At Tier-2, regional hubs can aggregate models, datasets, and results from several lower-tier deployments and implement full end-to-end ML pipelines. At Tier-3 and Tier-4, the focus shifts towards national and Pan-African coordination, including shared GPU resources, model lifecycle services, federated access, model sharing across participating infrastructure, and interconnection through research and education networks.

A central design principle of the Edge-AI Blueprint is that it is built by composing the common services defined in Section 3.2 rather than by implementing a monolithic platform. Cluster creation, notebook deployment, storage, authentication, and monitoring are reusable services that can be combined to instantiate the blueprint at different tiers. This means that the same blueprint can be deployed in a lightweight form at the edge or in a more complete form at a regional or national site.

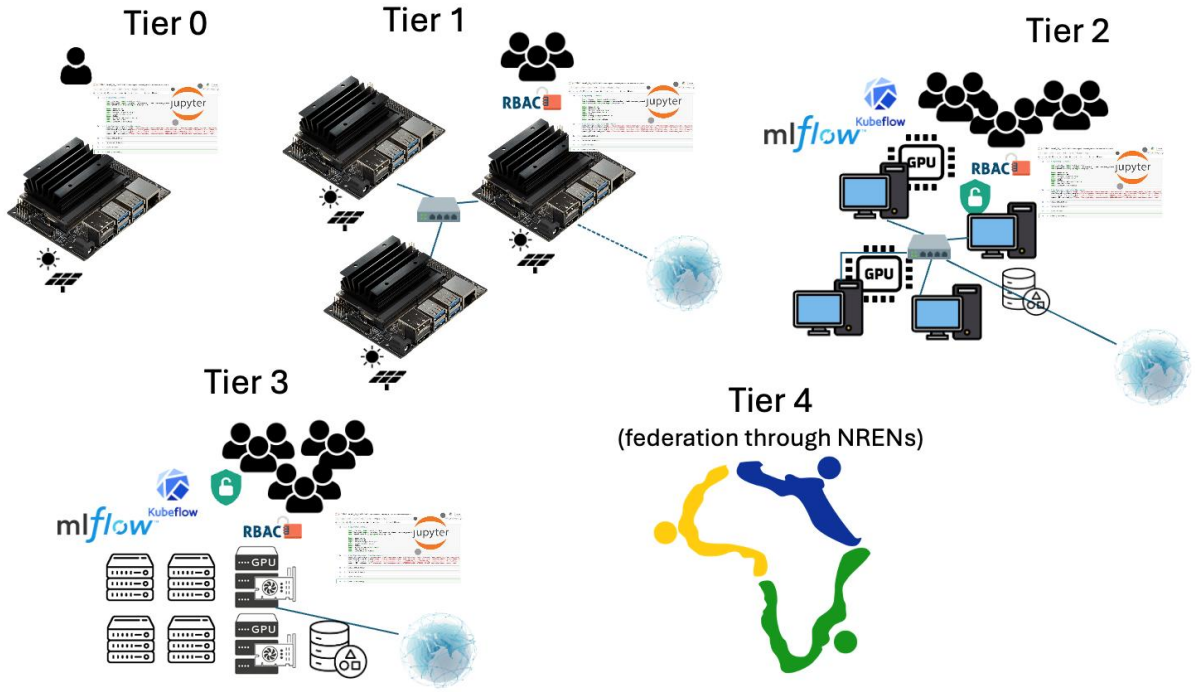

Figure 3 – Tier-based incremental deployments for the edge-AI blueprint

3.3.2 Edge-AI Blueprint Architecture

The Edge-AI Blueprint should allow AI computations to be done locally with the data source, whether it is in a rural healthcare facility, a farm, a school, or a research lab. Any cloud services can be remotely synchronized in a controlled manner when a secure connection is available. Services should be capable of functioning in a disconnected or low connectivity environment, be easily installed on regular machines or lightweight virtual instances, and should not require frequent long distance data transfers.

Simultaneously, the blueprint caters for the shared priorities that have been identified by the partner countries, that is to say, digital and remote education, telemedicine, sustainable agriculture, and AI driven innovation platforms. Each of these areas requires the development of tools that can support locally deployed edge models, collect data from local IoT devices or health sources, and allow both experts and learners access to the development environments.

An individual installation should be capable of operating independently, i.e., without external assistance (which is an indispensable condition for rural and low infrastructure environments), and can also, via lightweight APIs and federated identity systems, be part of a larger network of similar installations. This should obviate the problem of fragmented national solutions and at the same time, should meet the requirement of synchronization

and cooperation between the countries, thus, it is in line with the DIGITAfrica's mission of shared and sustainable digital development.

The Edge-AI Blueprint was designed around four main pillars, based on the requirements that each one provides: 1) edge orchestration and compute management, 2) Edge-AI capabilities, 3) storage (both short-term and long-term) and data integration, and 4) user access and authentication.

Edge Orchestration

This part of the architecture is handling the compute part and the orchestration of workloads within the context of the joint edge/cloud continuum. The component provides all the necessary control logic for directing how the AI and data processing workloads are orchestrated, placed, executed and exposed across heterogeneous edge sites. It abstracts the complexity of resources, and provides the needed APIs for enabling all the above functions. As such, it provides a seamless experience to orchestrate workloads, regardless of where they are placed/executed, or the type of infrastructure that they manage. The component should be able to allow fine grained control of the workload scheduling process, low-latency scheduling, as well as resource and workload isolation across different users that could use the resources at the same time.

For this purpose, a component that presents a mature stack with standardized APIs and easily extensible is required. An example of such an approach is the Kubernetes stack, that also provides lightweight approaches for managing resource constrained devices (e.g., the Rancher k3s). K8s can manage workloads deployed as microservices, taking advantage of different container runtimes like CRI-O and containerd. Further integration to a multicluster environment across different countries is also supported, if deemed appropriate to fulfil the needs of DIGITAfrica.

The underlying hardware that can be managed can range from simple compute devices located at the edge or the core cloud, to clusters of GPU servers, or resource constrained devices with different processor architectures (e.g. ARM64 based vs x86). The respective functionality is implemented by the basic service on cluster creation.

Edge-AI capabilities

The edge AI component represents the computational intelligence of the blueprint, and the main functionality delivered to the end users. It enables the training, inference and federated learning scenarios at the network edge. Its purpose is to use computationally capable edges, in order to minimize the distance between the actual generation of data and their consumption, with reduced latency while also preserving data sovereignty and data privacy. In case that accelerated processing is enabled at the edge (e.g., GPU enabled edge), the components should take advantage of this functionality. The component needs to also support reproducible pipelines, model versioning, and experiment tracking, in close collaboration with the computational component for the edge.

Tools such as Kubeflow that can orchestrate ML pipelines can play a key role in this. Frameworks such as pytorch and tensorflow can be used for the training and inference of workloads directly at the edge. Such tools already take advantage of accelerators that are available on the infrastructure (e.g., GPUs) so they are deemed as a good choice for deployment.

A key component that should be available among the DIGITAfrica partners is the support for hosting workloads for federated learning processes. Given the limited availability for inter-country and even inter-site highspeed links, federated learning should be supported. The respective service should be deployed through the notebook service, and accompanying tools that are automatically installed with the blueprint.

Storage and integration with the architecture

As the integration of AI components usually requires the processing of large volumes of datasets, storage components become key for the success of the blueprint. The storage regards either ephemeral or user-based storage, that can be attached to different workloads running on the infrastructure and ensures that data can be served consistently with high-speeds. Versioning of the data within this component can ensure reproducibility of experiments.

Therefore, the system should provide low-latency access to local storage for the different worker nodes, used as ephemeral storage during different experiments/use cases, and scalable storage for larger datasets and long-term storing/sharing of data. Storage components should be able to integrate with the computation components for the edge, e.g., through a Container Storage Interface (CSI) driver. Key solutions for local ephemeral storage include the provisioning of storage pools through NFS/CEPH, and attaching them to the workloads through Persistent Volumes (PVCs). Long term storage can be similar, or use object-based storage, with frameworks such as Min.io. As a good practice, any dataset containing information related to human behaviour should be accompanied by clear and complete documentation on how personal data is handled. This includes indicating whether the dataset contains personally identifiable information, describing the anonymization or pseudonymization methods used, and explaining how compliance with data protection regulations (e.g., GDPR) is ensured. Documentation should also specify data usage policies, consent procedures, retention periods, and the security and ethical measures applied to protect sensitive information. Providing this level of transparency helps in supporting responsible data sharing practices and fosters user trust. The functionality for this service is deployed through the different implementations for storage-based services (e.g., S3, S3-cache, NFS based).

User Access and Authentication

User access and authentication are the entry point to the blueprint. The user access layer should provide a unified authentication and authorization mechanism across the services composing the blueprint, and potentially across the different blueprints. Users should not need to manage separate credentials for each service. Consider for example the case where a user needs to authenticate twice to use the infrastructure, initially for running the

workload, and subsequently for pulling data that are stored and needed for the training job. Therefore, authentication should be delegated to a common identity service, allowing notebook environments, storage services, and other blueprint components to rely on the same identity provider. For this purpose, the architecture relies on federated identity mechanisms with OIDC.

As the end users interface the notebook service the authentication layer should therefore be integrated with the notebook environment and all the common services that are needed (e.g., storage). After logging in, users use their own isolated workspace. This architecture choice enables supporting multi-user execution while preserving separation between users and experiments.

3.3.3 Target Use Cases

The Edge-AI Blueprint targets several representative DIGITAfrica use cases. In **South Africa**, edge telemedicine and federated learning are proposed to support AI-based diagnostic assistance in rural and peri-urban clinics while keeping patient data local. In **Kenya**, edge AI for precision agriculture will be used, deploying IoT sensors and local models to support irrigation and yield prediction. In **Senegal**, education labs and CI/CD automation can provide Jupyter environments for students and reproducible training sessions. In **Cameroon**, localized AI research can support applications such as natural-language processing for local dialects or mobility data analysis. In **Tunisia**, edge-to-cloud integration can support health and IoT testbeds connecting hospitals and universities through governed and traceable data exchange.

Sustainability is also embedded into the Edge-AI Blueprint. At the infrastructure level, lower-tier deployments may rely on solar power and batteries, depending on local site conditions. At the operations level, the blueprint can support energy-aware workload placement through the cluster creation, and improved accelerator utilization, using labelling/tags of nodes. At the service level, relevant metrics include energy per inference, per training job, or per user session. At the experiment level, the blueprint can support the estimation of CO_2 emissions per experiment. These parameters can be used for research purposes in the respective domains and are particularly important for deployments in constrained environments, where energy availability and operational cost directly affect the long-term viability of the infrastructure.

Overall, the Edge-AI Blueprint targets in creating a reusable architectural pattern for deploying AI capabilities close to users and data sources. By combining services for cluster creation, paired with a common OIDC based authentication system via the user-portal, coupled with externally attached storage, and providing notebook-based access, the blueprint enables DIGITAfrica to build local and scalable AI infrastructures suitable for both research and education.

3.4 Heterogeneous Networking Blueprint

The Heterogeneous Networking Blueprint defines the chaining of different DIGITAfrica services for deploying and managing networks that could span multiple technologies — 5G, WiFi, fibre, satellite, mesh — within a single, open, software-defined framework. That enables deploying and reproducing wireless and cellular networking workflows. Its target is to enable wireless and cellular networking services and experimentations and used according to the defined use cases, such as sensors, clinics, farms, schools, laboratories.

This blueprint is particularly important as in the DIGITAfrica context connectivity gaps are prominent with most rural and peri-urban areas still depending on outdated and fragmented infrastructure prone to frequent outages and difficulties to find local expertise.

The blueprint therefore enables DIGITAfrica partners and users to deploy, experiment, and operate their own wireless and cellular networks.

3.4.1 Blueprint tiering

This tiered model allows the blueprint to address both constrained and advanced pan-African deployment environments. Similarly to the Edge-AI blueprint, at Tier-0, the emphasis is on local autonomy, low power consumption, and the ability to operate with intermittent or no connectivity with the rest of the ecosystem. At Tier-1, the blueprint supports teaching, experimentation, and local networking development through multi-user notebook environments, with the implementation of Role Based Access Control (RBAC). At Tier-2, regional hubs can aggregate network traffic from several lower-tier deployments and implement full end-to-end networking pipelines. At Tier-3 and Tier-4, the focus shifts towards national and Pan-African coordination, including shared high-end resources, complex high availability deployments, and interconnection through research and education networks.

Table 2 - Tier breakdown for the Heterogeneous Networking Blueprint

Tier	Example Location	Capabilities	Budget
0	Rural school	Offline services, solar-powered, local cache. Local single instance JupyterHub.	$0.5k–1.5k
1	Campus lab	Basic infra, Local services, multi-instances JupyterHub access	$2–5k
2	Regional hub	Edge UPF + MEC, Multi-access, Model serving.	$15–30k

3	National	5G core, S3 + HPC, NREN-connected	$100k+
4	Pan-African	Federation across NRENs, federated identity, shared resources	Shared

A central design principle of the Heterogeneous Networking Blueprint is that it is built by composing the services defined in Section 3.2 and common with the Edge-AI blueprint rather than by implementing a monolithic platform. Cluster creation, notebook deployment, storage, authentication, and monitoring are reusable services that can be combined to instantiate the blueprint at different tiers. This means that the same blueprint can be deployed in a lightweight form at the edge or in a more complete form at a regional or national site or even combined with Edge-AI blueprint deployments to leverage resources and expertise.

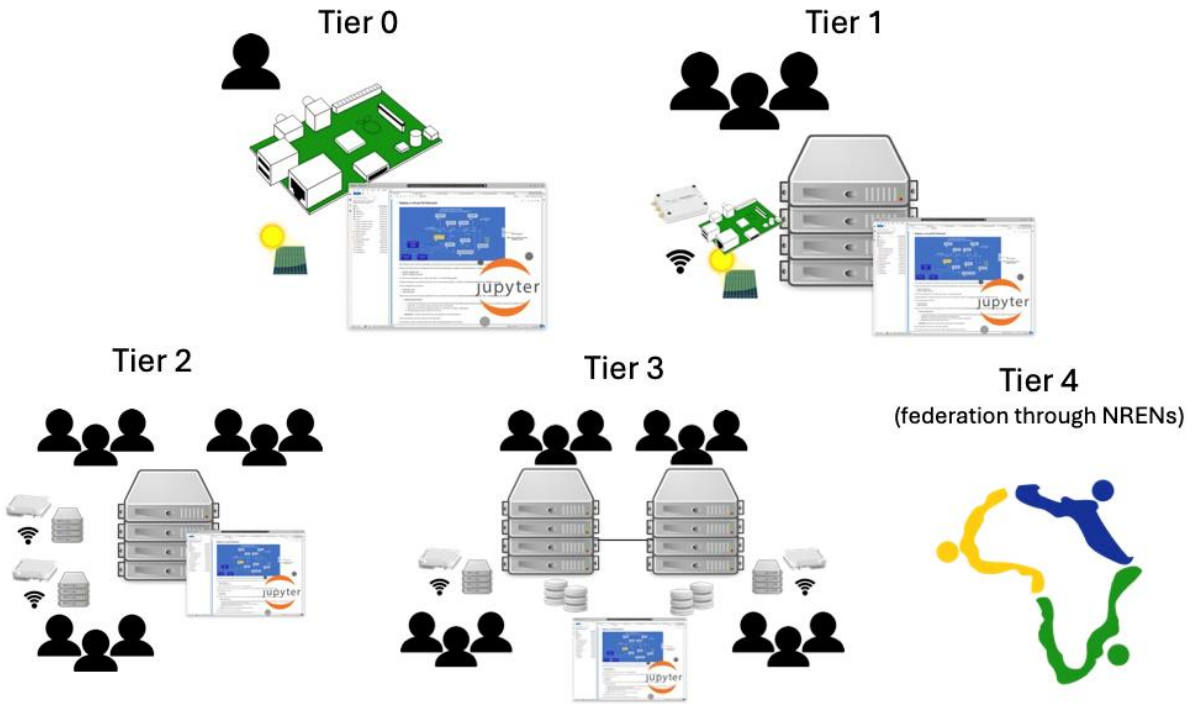

Figure 4 – Tier-based incremental deployments for the Heterogeneous Networking blueprint

3.4.2 Heterogeneous Networking Blueprint Architecture

The Heterogeneous Networking Blueprint provides a unified, software-defined framework that lets a site deploy and operate connectivity across several access technologies at once — cellular (5G NR), WiFi, low-power wide-area (LoRaWAN), and satellite backhaul — while keeping latency-sensitive processing local. As with the rest of DIGITAfrica, the blueprint is not a monolithic platform but a composition of the common services of Section 3.2, specialised for networking. A local deployment must be able to operate autonomously under

intermittent connectivity, run on modest hardware, and progressively federate with higher tiers when links and governance allow.

The blueprint is designed around four main pillars, mirroring the structure of the Edge-AI Blueprint so that the two can be deployed independently or side by side: 1) heterogeneous access and radio integration, 2) the software-defined core and dual-UPF traffic management, 3) edge compute, orchestration, and hardware-in-the-loop, and 4) user access, authentication, and shared storage. The following paragraphs describe each pillar; the concrete technology choices are summarised in the table at the end of this subsection.

Heterogeneous Access and Radio Integration

This pillar integrates the multiple radio access technologies into a single, coherently managed access layer. Cellular access is built on open-source 5G stacks rather than proprietary equipment, so that a site can stand up its own network using commodity software-defined radios. The reference stack targets at least 3GPP Release 16 and use OpenAirInterface (OAI) for the gNB/RAN, paired with the 5G core described in the next pillar. Software-defined radios are connected to the host via hardware passthrough (see the implementation in Section 4.3), with USRP B210 as the reference devices. Beyond cellular, the access layer can integrates WiFi, LoRaWAN for wide-area, low-bandwidth IoT sensing, and a resilient backhaul option for sites without fibre. The reference backhaul uses satellite target, e.g., Starlink LEO terminal which is assumed at tier 3. The pillar's role is to present these heterogeneous bearers to the upper layers through common, standards-based interfaces so that services need not be aware of which medium carries their traffic.

Software-Defined Core and Dual-UPF Traffic Management

The core network is deployed as cloud-native, containerised 5G network functions running on the cluster service, using the open-source 5G core from OpenAirInterface (OAI). A central architectural feature is dual User Plane Function (UPF) anchoring: a local Edge UPF provides low-latency local breakout for latency-sensitive services (for example, on-site telemedicine or local inference), while a central UPF handles cloud services and inter-site traffic. Policy-based dynamic traffic steering selects the appropriate data path per service, so that a single deployment can keep sensitive or delay-critical traffic local while still reaching federated resources when connectivity permits. Because the core functions run as standard workloads on the same Kubernetes/K3s substrate as every other DIGITAfrica service, they inherit the blueprint's tiering, monitoring, and upgrade path without bespoke infrastructure. In a distributed deployment, this same dual-UPF mechanism is what lets a zone route traffic to its local data network or to a remote one.

Edge Compute, Orchestration, and Hardware-in-the-Loop

Networking experiments differ from pure software workloads in that they require direct, low-level access to radio and compute hardware. This pillar reuses the cluster service (K3s at lower tiers, full Kubernetes at higher tiers) to orchestrate the network functions, and extends the notebook service with two networking-specific capabilities described in Section 4.3: safe docker-in-docker, which lets a 5G experiment instantiate its own cluster inside a notebook without granting the notebook privileged host access; and hardware passthrough,

which exposes a host USB or PCIe device (such as an SDR or radio unit) directly to the container running the experiment. Together these allow a complete radio-access-plus-core experiment to be deployed, run, and torn down reproducibly from a browser, while preserving isolation between users and protecting the host. Edge compute co-located with the access layer also supports Multi-access Edge Computing (MEC) at Tier-2 and above, enabling services to run adjacent to the radio.

User Access, Authentication, and Shared Storage

As with the Edge-AI Blueprint, users enter the blueprint through the notebook service and authenticate once via the user-portal (Keycloak/OIDC), with sessions and isolated workspaces carried across the blueprint's services. The storage service provides the data layer for networking work: shared NFS volumes let network functions within a cluster exchange configurations and measurements, while S3-compatible object storage (MinIO) holds experiment traces, packet captures, and datasets for longer-term sharing. Reusing exactly the same identity and storage services as the Edge-AI Blueprint is what allows the two blueprints to be combined on a single site — for example, feeding network measurements into an AI pipeline — without duplicating infrastructure, while still allowing the Networking Blueprint to be deployed entirely on its own.

3.4.3 Distributed and multi-tenant deployment

The pillars above describe a deployment on a single machine. In practice, the value of the Heterogeneous Networking Blueprint comes from operating across multiple radios, multiple tenants, and multiple sites — this is what makes it heterogeneous and distributed rather than a single isolated 5G network. Three capabilities provide this. Multi-tenancy lets several independent radio deployments share one server. The connectivity service interconnects deployments and emulates the realistic links between them. Distributed 5G then combines these to run coordinated deployments across separate locations, with a local data network for the services users actually consume. Together they form a natural progression: scaling up on one machine, connecting machines and sites, and finally running a distributed deployment across them. Of the three, multi-tenancy has been validated on hardware; the other two are the design direction that builds on it.

Multi-tenancy on a single server

A shared site that mirrors the SLICES-RI model must be able to host more than one independent radio deployment on a single server. Multi-tenant operation using USRP B210 software-defined radios has been validated on a shared server at UCT as a proof-of-concept for the blueprint.

The main constraint encountered is that each USRP B210 requires its own independent USB controller. When two radios share a single controller the USB bus becomes saturated, causing unstable operation; when each radio is placed on a separate controller, operation is stable.

Stable multi-tenant operation is secured through three configuration practices, which the blueprint incorporates as standard. Each radio is addressed explicitly by its serial number in

the gNB configuration, so that a deployment always binds to the intended radio rather than an arbitrary one. Each tenant's radio is isolated by mapping its specific USB bus into the corresponding container, keeping tenants independent. And device discovery is performed before each deployment, because a radio's bus identifier can change after its firmware is loaded; resolving the identifier first ensures the configuration remains correct.

Interconnection and link emulation: the connectivity service

Once more than one deployment exists — whether several tenants on one server or several sites — they must be interconnected, and the links between them must be made realistic for experimentation in African conditions. This is the role of the connectivity service, one of the five common services (Section 3.2), described here in the networking context where it is most needed.

The connectivity service is envisaged as a lightweight virtual machine running Linux and providing tunnelling and proxying — for example VXLAN, WireGuard, OpenVPN, or a reverse proxy — together with the ability to shape link behaviour using Linux traffic control (tc). A site that wishes to interconnect DIGITAfrica instances, or to emulate a particular link, deploys this service and diverts the relevant traffic through it.

The service is also intended to provide network emulation, so that high-latency or low-bandwidth links can be reproduced for experimentation and teaching. This reuses the established approach of emulating latency and bandwidth with Linux queueing disciplines and tc — the same technique which emulators such as Mininet rely. A heavier inter-cluster solution (Submariner) was considered but judged too complex for the resource- and skill-constrained DIGITAfrica context. The connectivity service is not implemented yet, and is targeted for tier 3.

3.4.1.1 Distributed 5G and the generic data network

Combining multi-tenancy with the connectivity service enables the blueprint's distributed (edge) configuration, in which two or more 5G deployments at different locations are interconnected. Within such a deployment, heterogeneous access is exposed at the user side: a device can attach to the same 5G core through either 5G NR or WiFi, integrating the two access technologies behind a common core.

On the data-network side of the core — the N6 interface — each location hosts a local, Kubernetes-based data network: in implementation terms an additional k8s node or namespace, with a DNS service, where partners deploy general internet services that are not themselves 5G-specific. This reflects an observation from the partner use cases: most do not depend directly on 5G, but rather need ordinary internet services reachable over a 5G (or WiFi) bearer.

This is where the distributed configuration connects directly to the dual-UPF anchoring introduced earlier in this blueprint. Each distributed zone runs a local Edge UPF alongside a Central UPF, and policy-based traffic steering decides, per service, whether traffic breaks out to that zone's own local data network or is routed across the interconnect to a remote

data network at another site (or to central cloud services). Latency-sensitive or data-sovereign services are kept on the local data network through the Edge UPF, while traffic that must reach services hosted elsewhere is steered through the Central UPF and over the links provided and emulated by the connectivity service. The dual-UPF model is therefore not only a single-site low-latency mechanism but also the means by which a distributed deployment chooses between local and remote data networks.

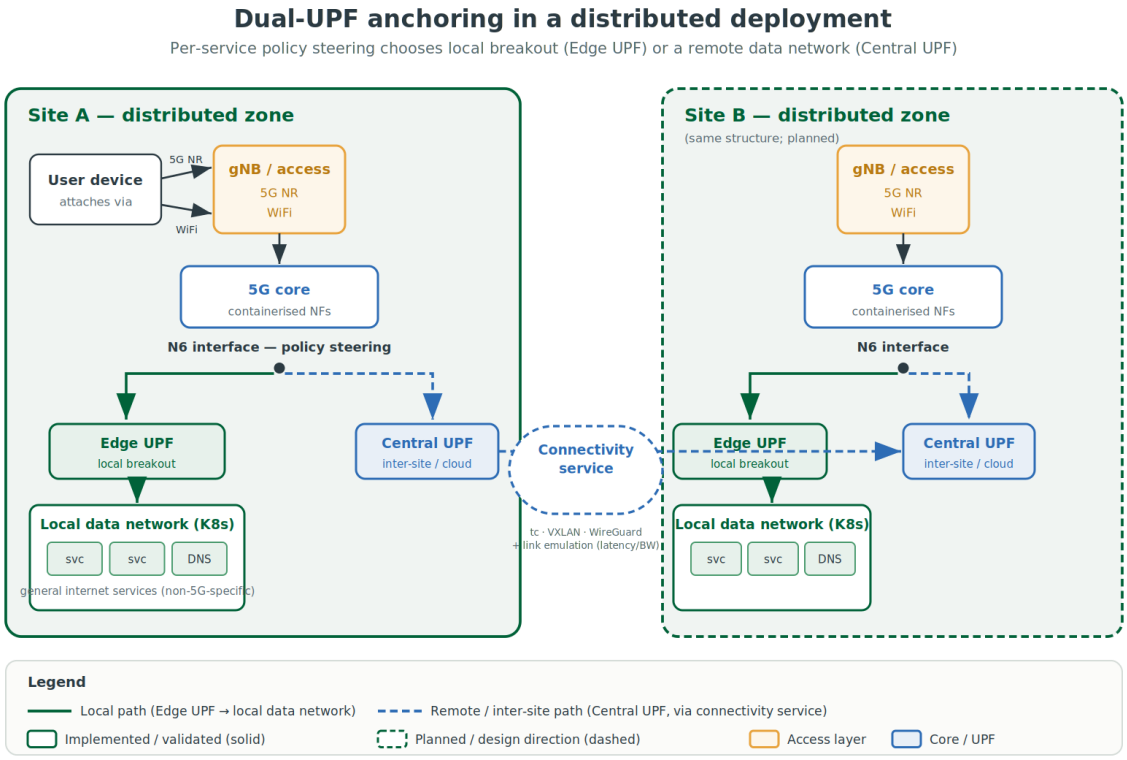

Figure 5 - Dual-UPF anchoring in a distributed deployment — per-service policy steering routes traffic to the local data network (Edge UPF) or to a remote data network at another site (Central UPF, via the connectivity service).

Providing the local “generic cloud” itself requires little additional machinery, since it is essentially another instance of the common services without the edge-AI or networking blueprint installed; the missing piece is the connectivity service of the previous part. This motivates framing the two blueprints as a continuum rather than as separate silos: a use case is realised by deploying a (possibly minimal) network with the Networking Blueprint, deploying its services on the N6 data network, and reaching them over the access layer — with the Edge-AI Blueprint providing the services and workflows that run on that data network.

3.4.4 Target Use Cases

The use cases targeted by the Heterogeneous Networking Blueprint are the same representative DIGITAfrica blueprints considered for the Edge-AI Blueprint. Indeed, while these use cases have significant AI requirements addressed by the Edge-AI Blueprint, they also impose substantial networking requirements that are supported by the Heterogeneous Networking Blueprint.

In South Africa, edge telemedicine and federated learning are proposed to support AI-based diagnostic assistance in rural and peri-urban clinics while keeping patient data local. The resulting federation of services and the collection of data from remote locations raise fundamental networking challenges. In Kenya, precision agriculture requires the deployment of IoT sensors that depend on an underlying communication infrastructure. In Senegal, education labs and CI/CD automation provide Jupyter environments for students and support reproducible training sessions. In Cameroon, localized AI research relies on data collection from heterogeneous entities distributed across different locations. In Tunisia, edge-to-cloud integration raises the challenge of connecting hospitals and universities in a manner that is lightweight, privacy-preserving, and resilient.

Sustainability is also a core consideration of the Heterogeneous Networking Blueprint. At the infrastructure level, lower-tier deployments may rely on solar energy and battery systems, depending on local site conditions, which requires communication devices and protocols to be energy-efficient. At the operational level, the blueprint can support energy-aware communication configuration and path selection. At the service level, relevant metrics include energy consumption per transmission and the energy efficiency of achieved goodput. At the experiment level, the blueprint can support the estimation of CO_2 emissions associated with individual experiments. These parameters can be used as research metrics across the different application domains and are particularly important in constrained environments, where energy availability and operational costs directly affect the long-term sustainability of the infrastructure.

Overall, the Heterogeneous Networking Blueprint aims to provide a reusable architectural pattern for deploying networking capabilities close to users and their needs. By combining services for cluster provisioning, a common OIDC-based authentication system through the user portal, access to external compute and storage resources, and notebook-based user interfaces, the blueprint enables DIGITAfrica to build scalable and locally adapted communication infrastructures that support both research and education.

4 Implementation

This section delves into the implementation details and what has been implemented as a proof-of-concept during the first reporting period. The implementation follows the exact same modular approach as the architecture, with common services being implemented and re-used for the different blueprint. Two approaches have been followed for the implementation, depending on the complexity of each service. Each service implementation is available as a docker container instance deployed with docker-compose files, or for the more complex services, ansible scripts are available, able to run and deploy each playbook on a target machine.

The blueprints were used during the 2026 Nairobi DIGITAfrica Winter School and the 2026 Tunis DIGITAfrica Workshop to train participants in 5G networking and AI data processing. In total, more than 60 students were trained using the blueprints.

To gather comprehensive feedback on the blueprints and assess their usability, the training sessions were delivered directly by the blueprint architects and developers. In addition, prospective developers from the host countries were invited to participate and observe how researchers and students can use the blueprints to support their work. This hands-on approach provided valuable insights into both the user experience and the development requirements of the platform.

4.1 Common services

4.1.1 Cluster

The cluster implementation provides the compute substrate for the DIGITAfrica blueprints. The current implementation is based on K3s, a lightweight Kubernetes distribution suitable for constrained environments. As several of the devices used in the context of DIGITAfrica need to be powered through renewable energy sources (e.g., solar-power), devices such as Raspberry-Pis and Nvidia Jetson are expected to be used for Tier-0 and Tier-1 deployments for target use cases. K3s is used as it has small footprint, can run on such constrained devices, while preserving the Kubernetes API and operational model. The compatibility with the Kubernetes API allows each lower Tier to migrate to larger Tier deployments if resources at each partner site are available.

The cluster repository[8] provides an Ansible-based deployment of k3s clusters on a set of nodes. Each node described in the Ansible inventory has a role: Server-nodes run the control-plane and API part of the cluster, while the agent-nodes are workers who join the cluster. This breakdown allows a node to be reconfigured if the operating context changes at a location, e.g., a single-node Tier-0 device, can be used to join a Tier-1 deployment as a worker.

[8] https://gitlab.inria.fr/digitafrica/blueprints/services/k3s-cluster

```
### TIER 1 NODES ###
[tier1_server]
digitafrica-edge-node1 ansible_host=10.64.45.176 ansible_ssh_pass=REDACTED ansible_become_pass=REDACTED

[tier1_agents]
digitafrica-edge-node2 ansible_host=10.64.45.179 ansible_ssh_pass=REDACTED ansible_become_pass=REDACTED
digitafrica-edge-node3 ansible_host=10.64.45.175 ansible_ssh_pass=REDACTED ansible_become_pass=REDACTED

[tier1:children]
tier1_server
tier1_agents

[all:vars]
ansible_user=ubuntu
ansible_become=true
#ansible_ssh_common_args='-o ProxyJump=proxy@bastion1.theblueprintfactory.org'
```

Figure 6 – Configuration hooks for defining roles for each machine (e.g. prior Tier-0 node joining Tier-1 as a worker)

Cluster networking is handled using the Traefik controller[9]. This approach has several benefits as it allows the user to select the way that services are exposed outside of the cluster in the following cases:

1. Deployment with no Traefik controller, allows the exposure of the services with approaches such as NodePort (i.e., port forwarding a specific ingress port to a service port).
2. Deployment with Traefik, allowing services to be reached through paths specified in the URL of the service. Such an approach redirects the traffic to the service port, if the specific path has been specified in the path of the request.
3. Deployment with TLS termination, allowing all the services deployed within the cluster to be exposed with a secure endpoint. This approach allows the administrator of the cluster to specify 1) self-signed certificates, 2) certificates issued by the Let's Encrypt provider, or 3) their own existing certificates.

[9] https://traefik.io/

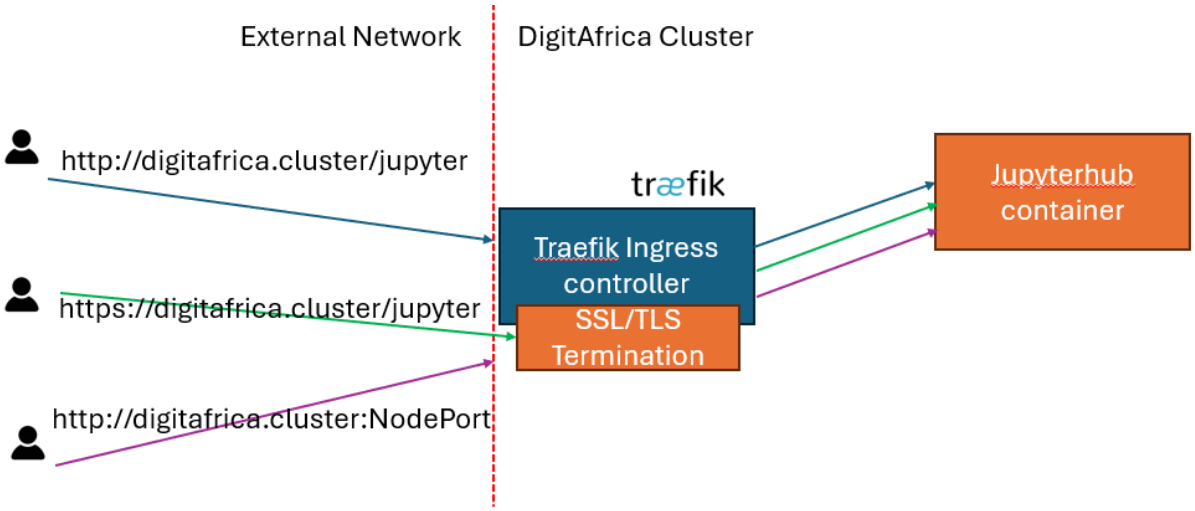

Figure 7 – Modes of exposing services from the DIGITAfrica clusters

Although the current implementation focuses on k3s, higher Tiers are expected to use Kubernetes variants. Such approaches will enable higher scalability, high-availability, while preserving the same core orchestration concepts and APIs, while being compatible with the services already available.

4.1.2 User-portal

The user portal is implemented using Keycloak[10]. Each site deploys its own Keycloak instance and configures it according to local requirements and constraints. Keycloak supports the registration of multiple identity providers, each of which may use a different authentication protocol. The most common supported protocols are LDAP, OAuth 2.0, and OIDC. In addition, Keycloak provides a local user database for managing accounts directly.

Each site can therefore configure its Keycloak instance to integrate with its existing identity management system (e.g., a university LDAP directory) or create local accounts directly within Keycloak (e.g., for Tier-0 deployments). The site also connects its Keycloak instance to DIGITAfrica's OIDC identity provider. This approach enables users to seamlessly access DIGITAfrica services using their preferred identity provider.

Services are connected to the user portal through the use of JSON Web Tokens (JWTs). JWTs are signed identity tokens that can be validated by any service that trusts the token issuer. Using JWTs makes it possible to build complex service ecosystems without requiring every service to be explicitly registered with the user portal. Instead, each service only needs to trust the user portal and be able to validate the tokens it issues. A service may trust multiple

[10] https://www.keycloak.org/

identity providers if necessary. This approach has proven highly scalable and represents the current industry standard for authentication and authorization in distributed systems.

For Tier-0 deployments, Keycloak is typically deployed as a standalone Docker container. For higher tiers, Keycloak is deployed as a Kubernetes pod with an associated service within the cluster. The user-portal implementation is provided through the user-portal repository at DIGITAfrica and includes all the Ansible playbooks and roles required to deploy and configure the service. For Tier-1 and higher deployments, the implementation assumes the presence of a Kubernetes cluster, either based on K3s or a full Kubernetes distribution.

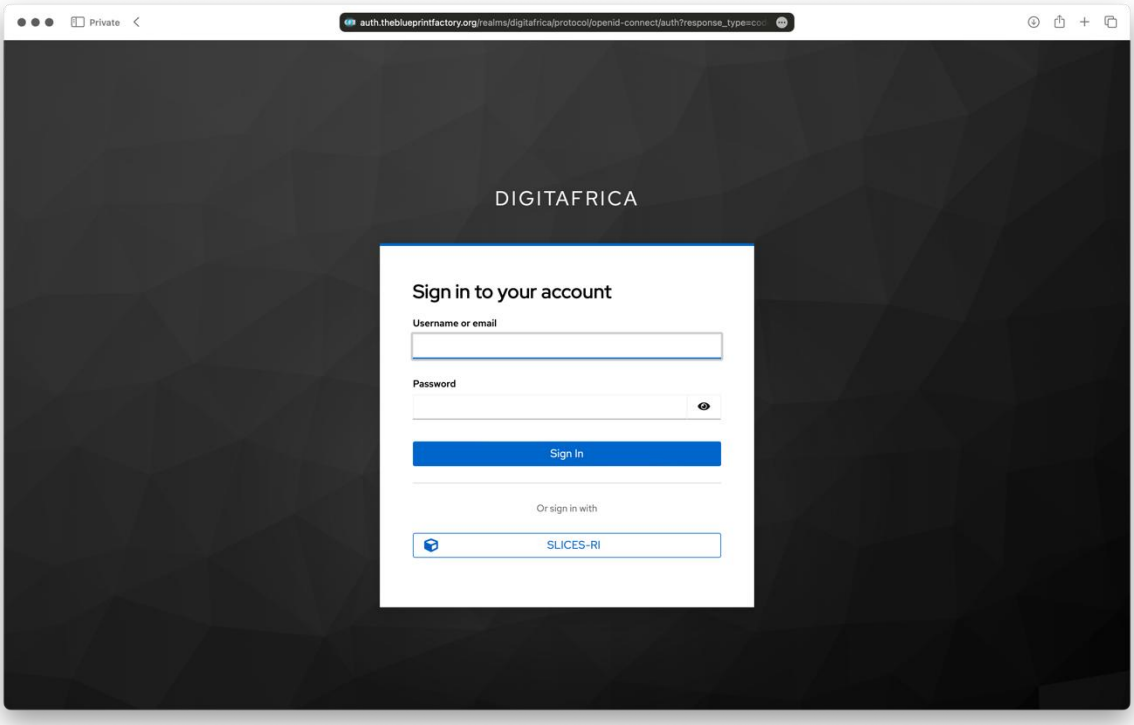

Figure 8 - Example of a DIGITAfrica user-portal connected to the SLICES-RI identity provider

4.1.3 Storage

The storage implementation covers both shared filesystem storage and object-based storage. These two forms of storage address different needs in the DIGITAfrica infrastructure. Shared filesystem storage is required for persistent volumes and user workspaces, while object storage is better suited for datasets, trained models, experiment outputs, and long-term sharing across services or sites.

The shared storage implementation is provided through the NFS server repository[11]. This repository contains Ansible playbooks to deploy and configure an NFS server and its clients

[11] https://gitlab.inria.fr/digitafrica/blueprints/services/nfs-server

within the cluster. The purpose of this service is to provide centralized shared storage across nodes and persistent volumes for services running on the infrastructure, such as JupyterHub.

Object storage is implemented through the MinIO-based S3 storage repository[12]. This repository installs and configures MinIO as an S3-compatible object store on a bare-metal or virtual-machine Ubuntu server. The deployment supports raw block devices, which are formatted and mounted for MinIO storage (XFS format), as well as a directory access when dedicated disks are not available at the target site. The administrator deploying the implementation can specify the MinIO credentials, S3 API port, web console port, and credentials for the Keycloak/OIDC integration. The latter allows each user who has an account in the portal of each site, to also have a specific storage bucket within the server.

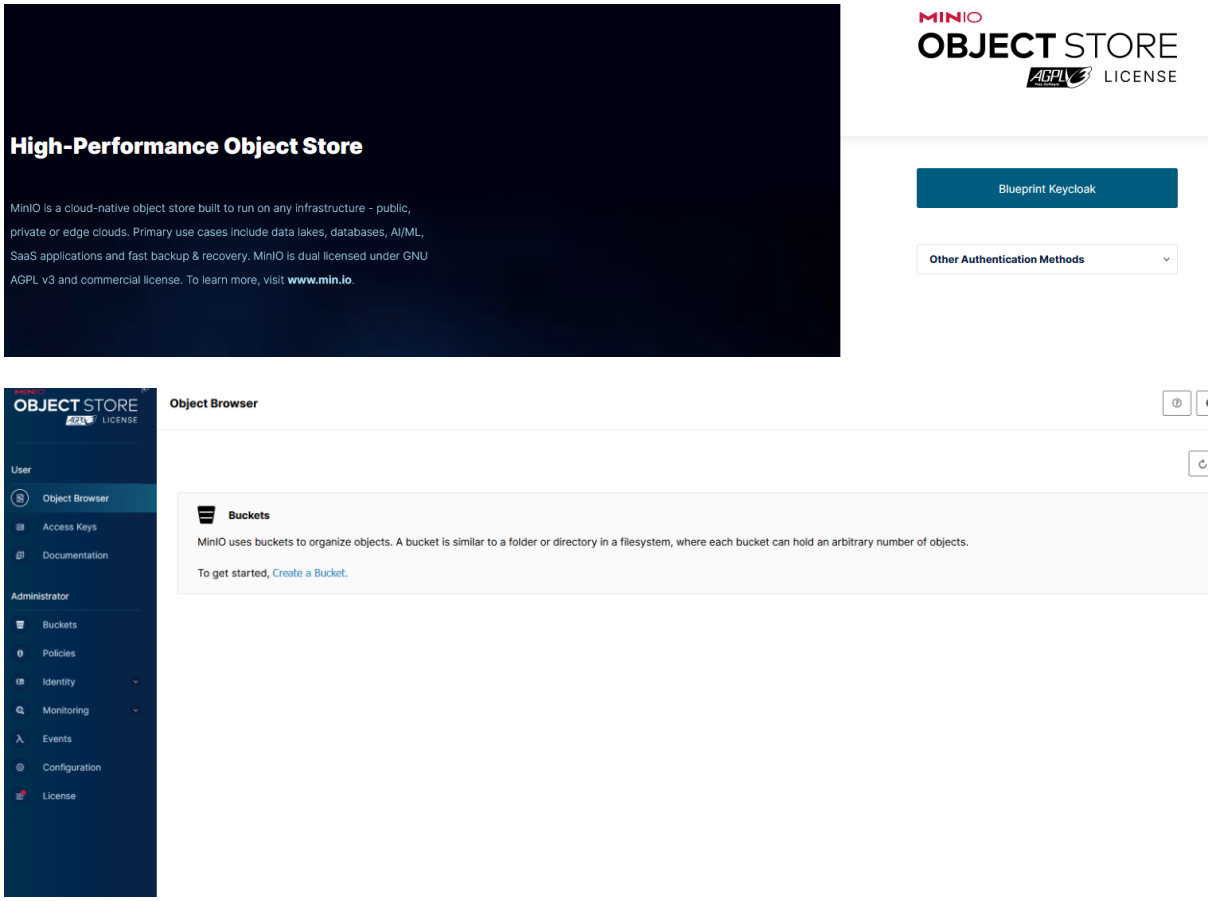

Figure 9 – S3-storage instantiation and link with portal OIDC

4.1.4 Notebooks

Notebooks are executed in controlled environments managed by JupyterHub. JupyterHub is essentially a frontend to notebooks that automatically spawns notebook instances when users want to run notebooks. In tier 0, JupyterHub is deployed as a single docker container and it deploys notebooks as docker containers too. For tier 1 and above, JupyterHub is

[12] https://gitlab.inria.fr/digitafrica/blueprints/services/s3-storage-minio

deployed in the Kubernetes cluster of the site and spawns notebooks as pods in the Kubernetes cluster, enabling full isolation of users and scalability. As JupyterHub is linked with the user-portal, notebooks are linked with their users. That way, whenever users login, they find back their environment as they left it last time they used the service.

4.2 Edge-AI Blueprint

The Edge-AI Blueprint implementation builds on the common services described above. Its current repository chains together these services, and enables deployment automation for Tier-0 and Tier-1 Edge-AI environments. The implementation is currently a proof-of-concept for these Tiers, but establishes the core deployment patterns needed to enable an environment that hosts AI/ML workflows on lightweight edge and cluster-based infrastructures.

The current implementation supports three main deployment modes. The first is a Tier-0 bare-metal deployment, where Jupyter notebooks are deployed on a single node using Docker. This deployment also includes cAdvisor and NodeExporter, allowing basic monitoring of container-level and node-level statistics. The second is a Tier-0 single-node K3s deployment, where a single-node K3s cluster is installed and Jupyter is deployed on top. The third is a Tier-1 K3s deployment, where a multi-node K3s cluster is deployed and JupyterHub is installed to support authenticated multi-user access.

The Edge-AI implementation assumes that use cases will deploy their functionality through scripts and notebooks running on the infrastructure. This is consistent with the architecture of the blueprint, where notebooks are the primary user interface and where AI workloads are expected to be delivered as reproducible workflows. The current code has been tested using a three-node cluster based on Raspberry Pi 5 devices, Nvidia Jetson Nano devices, and Ubuntu-based VMs, showing that the implementation can target both lightweight and edge-oriented hardware.

Networking for the blueprint services is handled through Traefik Ingress when K3s is used. The Ansible code provides a common expose_mode variable for configuring the manner of exposure. In ingress mode, services are exposed through the Traefik ingress controller provided by K3s. In nodeport mode, services are exposed through port-forwarding mappings, useful for constrained environments, testing, or debugging. The Tier-1 implementation also supports several TLS modes, including self-signed certificates, Let’s Encrypt certificates, externally provided certificates, or HTTP-only operation. This allows the same deployment logic to support both local testbeds and more production-like environments.

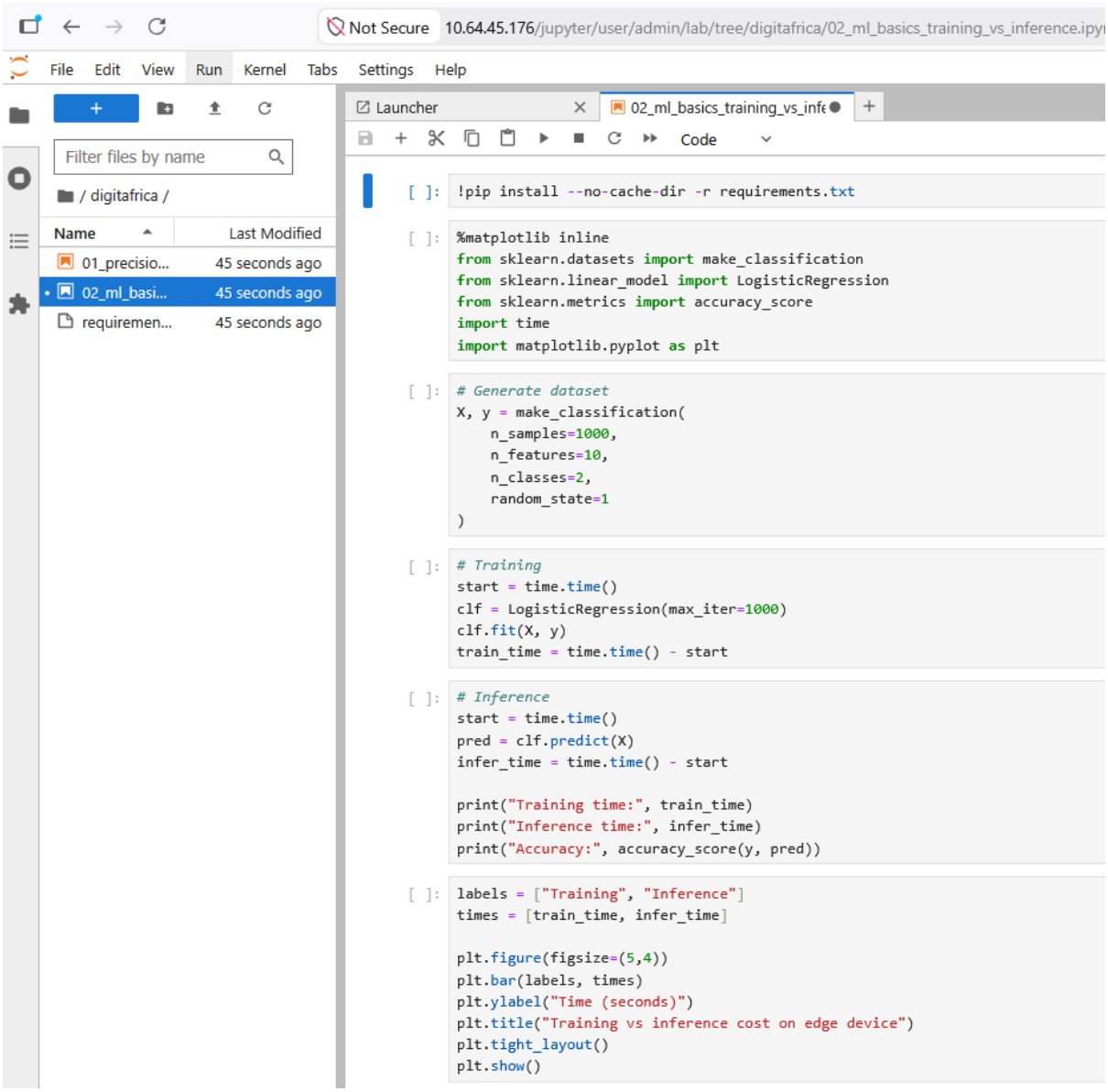

Figure 10 - Basic notebooks with the initial deployment of the Edge-AI blueprint

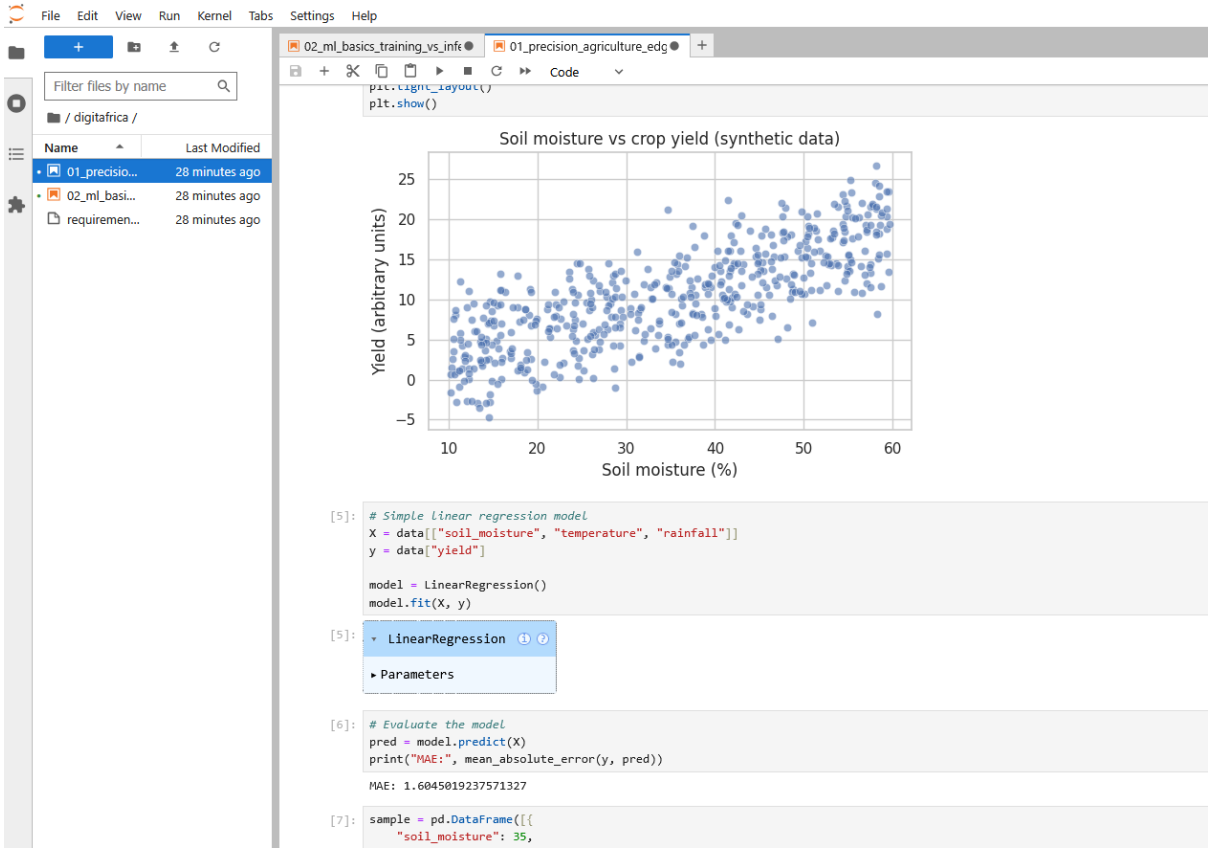

Figure 11 – Basic notebooks with the initial deployment of the Edge-AI blueprint

Authentication can be enabled through the OpenID Connect bindings available in the playbook. When OIDC is enabled, JupyterHub delegates login to Keycloak or another compatible identity provider. When OIDC is disabled, the deployment falls back to a dummy authenticator, which is useful for local testing and early experimentation. This makes the same implementation usable both in lightweight development settings and in deployments requiring proper user authentication.

The Edge-AI implementation therefore currently provides the basic operational layer for AI/ML experimentation: cluster deployment, notebook access, monitoring, ingress exposure, and authentication. Future implementation work can extend this foundation with more advanced AI services, such as ML pipeline orchestration, federated learning frameworks, model registries, accelerator-aware scheduling, and long-term model sharing through the storage services described above.

4.3 Heterogeneous Networking Blueprint

The Heterogeneous Networking blueprint implementation builds on the common services described above. Its current repository[13] chains together these services, and enables deployment automation for Tier-1 heterogeneous networking environments. The implementation is currently a proof-of-concept for this Tier, but establishes the core deployment patterns needed to enable an environment that hosts networking in cluster-based infrastructures.

The current implementation only supports one deployment mode: a Tier-1 k3s deployment, where a multi-node k3s cluster is deployed and JupyterHub is installed to support authenticated multi-user access.

The heterogeneous networking blueprint implementation assumes that use cases will deploy their functionality through notebooks running on the infrastructure. This is consistent with the architecture of the blueprint, where notebooks are the primary user interface. The current code has been tested using a multi-nodes clusters based on AMD64 servers and Ubuntu-based VMs, showing that the implementation can target both flexible and specific hardware.

The heterogeneous networking blueprint extends the standard DIGITAfrica JupyterHub service to support docker-in-docker scenarios in a safe and isolated manner. Most of modern 5G deployments are designed in a cloud-native way on top of Kubernetes and using multiple nodes. With enable docker-in-docker inside notebooks directly, 5G experiments can deploy their own cluster in the notebook instead of having to rely on an external cluster. Doing that way significantly reduces the OPEX as it is not required to deploy and maintain a multi-tenant cluster. However, docker-in-docker requires the notebook to have privileged access to the system, which would endanger the whole infrastructure. To prevent security threats, notebooks do not directly run docker daemons. Instead, whenever a notebook is spawned in the heterogeneous networking blueprint, the system deploys a dedicated docker daemon and the notebook has only access to the API of the daemon, not the daemon itself, ensuring full isolation of notebooks.

Heterogeneous networking also requires direct access to the hardware running on the hosting machines. Access to the hardware is provided via hardware passthrough from the host to the guest container. For example, when a deployment requires direct access to a USB port to connect to a radio unit, the system automatically create a hardware passthrough between the USB device on the host and the container running the notebook. That way, the container accesses the device directly instead of a virtualized USB device.

Concretely, the current proof-of-concept deploys an open-source 5G stack on a K3s cluster. The radio access and core network are both provided by OpenAirInterface (OAI), implementing 3GPP Release 16. Radio connectivity in the testbed is achieved using USRP

[13] https://gitlab.inria.fr/digitafrica/blueprints/services/heterogeneous-networking-blueprint

B210 over USB 3.0, attached to the notebook deployed container through the hardware-passthrough mechanism described above. These functions are packaged as the containerised, cloud-native deployment the architecture (Section 3.4.2) calls for, and are deployed and configured through the Ansible automation in the blueprint repository.

The heterogeneous networking implementation therefore currently provides the basic operational layer to deploy networking experimentation with hardware in the loop: cluster deployment, notebook access, monitoring, ingress exposure, and authentication. Future implementation work can extend this foundation with more advanced networking services. For example, as of now the heterogeneous networking blueprint networking has been focused on 5G, in the near future it will add the support of WiFi, LoRaWAN, and satellite backhaul.

All the details on configuration parameters for tier 1 deployment can be found in https://gitlab.inria.fr/digitafrica/blueprints/services/heterogeneous-networking-blueprint.

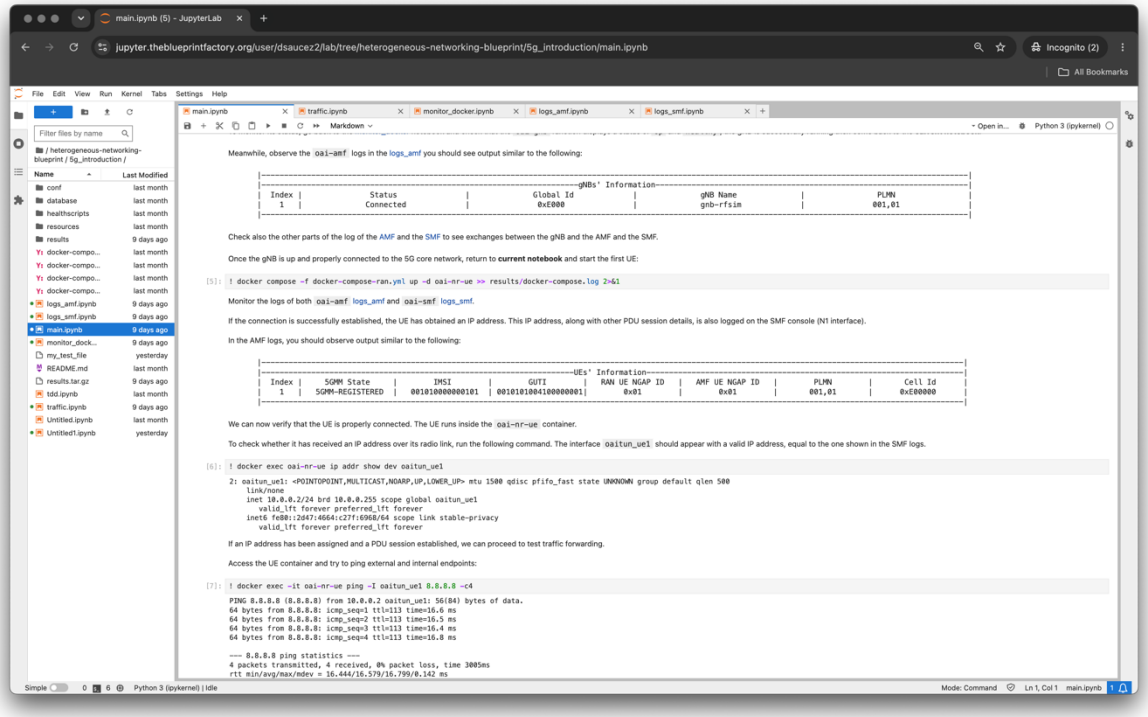

Figure 12 - Example of execution of a 5G network in the heterogeneous networking blueprint

4.4 Current limitations and open issues

Two limitations of the current proof-of-concept should be stated explicitly. First, while notebooks are the intended common interface, this is a goal rather than a guarantee: notebook-based delivery works well for AI/ML workloads, but networking workloads that drive real radio hardware are harder to fit into the notebook model, since they require USB passthrough and device allocation (see Section 3.4.3). Networking blueprints may therefore

retain non-notebook deployment paths. Where passthrough into the cluster proves impractical, a documented fallback is to deploy the gNBs outside the cluster and drive them through an API.

Second, the composition of services — central to the blueprint concept — is not yet solved elegantly. At present, composing a blueprint from its constituent parts is done by running several notebooks in a prescribed order, which is functional but not satisfactory. Providing a clean composition mechanism is identified as future work for WP3.

5 Conclusions

This deliverable presents the first version of the DIGITAfrica Blueprint, synthesising the outputs of Tasks 2.1, 2.2, and 2.3 into a coherent architectural framework for a pan-African research infrastructure in digital sciences. Two complementary blueprints — the Edge-AI Blueprint and the Heterogeneous Networking Blueprint — have been designed, documented, and partially implemented as proof-of-concept deployments. Both share a common modular architecture built on open-source, Kubernetes-compatible infrastructure, a five-tier progressive deployment model, a unified notebook-based user interface via JupyterHub, and common services for cluster management, identity management, storage, and connectivity.

The blueprints are directly grounded in the baseline service requirements identified across the five partner countries, addressing priority use cases in digital health, precision agriculture, and education. Sustainability principles from the GreenDIGIT project have been embedded from the outset — through energy-aware architecture choices, solar-compatible low-power hardware at lower tiers, and provisions for tracking Power Usage Effectiveness and carbon footprint across the infrastructure lifecycle.

The current implementation provides working proof-of-concept deployments for Tier-0 and Tier-1 of the Edge-AI Blueprint and Tier-1 of the Heterogeneous Networking Blueprint, validated on Raspberry Pi 5 clusters, Nvidia Jetson devices, virtual machines, and AMD64 servers. The target for the project lifetime is to reach at least Tier 2 for both blueprints, with pilot deployments at partner institutions enabling hands-on experimentation, research, and teaching. The blueprints have been used during the 2026 Nairobi DIGITAfrica Winter School and the 2026 Tunis DIGITAfrica Workshop, reaching more than 60 students, trained directly by the blueprint architects and developer. Planned next steps include Tier-2 hardware deployments at regional hub sites, integration of federated learning and MLflow model lifecycle management, deployment of 5G core and O-RAN hardware, operationalisation of GreenDIGIT sustainability monitoring, and finalisation of educational notebook content for partner university curricula. This version of the blueprint will be updated iteratively as deployments mature, leading towards the final design study in WP3.

Blueprints are an important component of our methodology demonstrating that we can transform the demand into a common playground and the associated learning components to be onboarded by the African partners and extensively used for education, training and research. More blueprints can be initiated if time and resources allows, following the same

methodology. We have showed that, at mid-project, we have been able to identify common needs best suited to the African context, design and deploy the first instance of these blueprints; exploit them in various events such as summer schools and hands-on. This effort will be further validated and consolidated during the second part of the project.

6 Annex I – Basic configurations

Tier-0

Table 3 - Configuration parameters tier-0

Variable	Default	Description
tier0.k3s_mode	none	none = Docker-only | single = k3s cluster | agent = join Tier-1
tier0.expose_mode	ingress	ingress = Traefik (only valid when k3s_mode=single) | nodeport = raw ports
tier0.notebook_port	8888	Port the Jupyter container listens on
tier0.jupyter_nodeport	30888	NodePort used when expose_mode=nodeport
tier0.modelcache_nodeport	30080	NodePort for model cache when expose_mode=nodeport
tier0.enable_modelcache	false	Deploy the static model-cache service
tier0.notebook_user	digitafrica	Local Linux user that owns notebook files
tier0.notebook_dir	/opt/digitafrica/notebooks	Directory mounted into the Jupyter container
tier0.notebook_password	digitafrica	Plain-text password (also set the hash below)
tier0.k3s_version	v1.30.4+k3s1	k3s version to install (only when k3s_mode=single)
tier0.k3s_agent_server_host	—	Tier-1 server hostname to join (only when k3s_mode=agent)

tier0.monitoring.node_exporter_port	9100	Node Exporter port
tier0.monitoring.cadvisor_port	8080	cAdvisor port

Tier-1

Table 4- Configuration parameters tier-1

Variable	Default	Description
tier1.expose_mode	ingress	ingress = Traefik on port 80/443 | nodeport = no Traefik, plain ports
tier1.tls_mode	selfsigned	selfsigned | letsencrypt | provided | none — see TLS
tier1.tls_cert_dir	—	Local path to tls.crt + tls.key (only when tls_mode=provided)
tier1.tls_acme_email	—	Email for Let's Encrypt registration (only when tls_mode=letsencrypt)
tier1.k3s_version	v1.30.4+k3s1	k3s version
tier1.k3s_server_host	—	Hostname of the k3s server node
tier1.k3s_cluster_cidr	10.42.0.0/16	Pod CIDR
tier1.k3s_service_cidr	10.43.0.0/16	Service CIDR
tier1.digitafrica_namespace	digitafrica	Kubernetes namespace for all app resources
tier1.jupyterhub.jupyterhub_public_url	https://<host>/jupyter	**Must match node's IP or DNS name.** Used for the final access URL.
tier1.jupyterhub.jupyter_nodeport	30888	NodePort for JupyterHub when expose_mode=nodeport

tier1.jupyterhub.storage_class	local-path	StorageClass for user PVCs. Tested with the nfs-client and the NFS-Server implementation
tier1.jupyterhub.jupyterhub_admin_users	["admin"]	List of JupyterHub admin usernames
tier1.mlflow.enabled	false	Deploy MLflow experiment tracking
tier1.mlflow.mlflow_port	5000	MLflow listening port
tier1.grafana_enabled	false	Deploy Grafana dashboards