CGIAR
SCIENCE FOR A FOOD-SECURE FUTURE

REPORT

Limpopo Digital Twin Architecture Report

Emmanuel Ayodele Jolaiya, Hugo Retief, Paulo Silva, Abdul Afham, Keerththanan Vickneswaran, and Mariangel Garcia Andarcia

December 2025

Authors

Emmanuel Ayodele Jolaiya, Consultant - Geospatial Software & Data Engineer, International Water Management Institute (IWMI), Colombo, Sri Lanka
Hugo Retief, Senior Researcher and Development Manager, Association for Water and Rural Development (AWARD), Hoedspruit, South Africa
Paulo Silva, Consultant - Solutions Architect, IWMI, Colombo, Sri Lanka.
Abdul Afham, Consultant - Junior Data Engineer IWMI, Colombo, Sri Lanka
Keerththanan Vickneswaran, Assistant Research Officer - Generative Artificial Intelligence, IWMI, Colombo, Sri Lanka.
Mariangel Garcia Andarcia, Research Group Leader - Water Futures Data & Analytics (WFDA), IWMI, Colombo, Sri Lanka.

Acknowledgments

This work was conducted under the CGIAR Initiative on Digital innovation and finalized under the CGIAR Accelerator for Digital Transformation, which advances sustainable agrifood systems through digital solutions and the International Water Management Institute's Digital Innovations for Water Secure Africa (DIWASA) project. Wewould like to thank all funders who supported this research through their contributions to the CGIAR Trust Fund (www.cgiar.org/funders).

We also wish to thank the Leona M. and Harry B. Helmsley Charitable Trust for their financial support for the DIWASA project and the Limpopo Watercourse Commission (LIMCOM) for their support to this project.

CGIAR Accelerator for Digital Transformation

The CGIAR Digital Innovation Initiative accelerates the transformation towards sustainable and inclusive agrifood systems by generating research-based evidence and innovative digital solutions. It is one of 32 initiatives of CGIAR, a global research partnership for a food-secure future, dedicated to transforming food, land, and water systems in a climate crisis.

Citation

Jolaiya, E. A.; Retief, H.; Silva, P.; Afham, A.; Vickneswaran, K.; Garcia Andarcia, M. 2025. *Limpopo Digital Twin architecture report.* Colombo, Sri Lanka: International Water Management Institute (IWMI). CGIAR Accelerator for Digital Transformation. 18p.

Front cover photo: 3D Point Cloud of the Crocodile River (Upper) (*photo*: Author's creation)
Back cover photo: (*photo*: Joshua Sortino/Unsplash)

Disclaimer

This publication has been prepared as an output of the CGIAR Accelerator for Digital Transformation and has not been independently peer reviewed. Responsibility for editing, proofreading, and layout, opinions expressed, and any possible errors lies with the authors and not the institutions involved.

Contents

List of Figures

Acronyms and Abbreviations

Acronym	Description
AI	Artificial Intelligence
ALB	Application Load Balancer
API	Application Programming Interface
AWS	Amazon Web Services
CD	Continuous Deployment
CDN	Content Delivery Network
CI	Continuous Integration
CPU	Central Processing Unit
DNS	Domain Name System
DT	Digital Twin
ECR	Elastic Container Registry
ECS	Elastic Container Service
EO	Earth Observation
EO3	Earth Observation version 3 metadata specification
GIS	Geographic Information System
HTTPS	Hypertext Transfer Protocol Secure
IaC	Infrastructure as Code
IoT	Internet of Things
JS	JavaScript
JSON	JavaScript Object Notation
OAuth2	Open Authorization 2.0
ODC	Open Data Cube
OGC	Open Geospatial Consortium
OIDC	OpenID Connect
OWS	Open Web Services
REST	Representational State Transfer
S3	Simple Storage Service
SQL	Structured Query Language
SSO	Single Sign On
SWAT+	Soil and Water Assessment Tool Plus
TLS	Transport Layer Security
VPC	Virtual Private Cloud
WAF	Web Application Firewall
WMS	Web Map Service
WMTS	Web Map Tile Service
YAML	YAML Ain’t Markup Language

Executive Summary

The Limpopo Digital Twin (DT) supports basin wide data sharing, scenario analysis, and decision support for the four member states of the Limpopo River Basin (Garcia et al. 2024). The platform brings together earth observation data, hydrological models, geospatial services, and prediction tools within a cloud-based environment designed for reliability, security, and long-term maintainability.

This report provides a full description of the system architecture, including application design, microservices, cloud infrastructure, data pipelines, and supporting platform services. It documents how the components interact, how they are deployed, and the operational processes that keep the system stable. The content reflects the current production implementation as deployed on Amazon Web Services (AWS).

The deployment uses a container-based model on Amazon Elastic Container Service (ECS) Fargate, supported by a multi zone Virtual Private Cloud (VPC), Application Load Balancer (ALB), Amazon Aurora MySQL database, and Amazon Simple Storage Service (S3) for earth observation and modelled datasets. Identity and access management is provided through Keycloak, which delivers user authentication and controlled access across platform applications. Amazon Secrets Manager, CloudWatch, Route 53, and Elastic Container Registry (ECR) serve as supporting services for configuration, monitoring, routing, and container images.

Development and operations follow an automated workflow using GitHub Actions, which builds container images, publishes them to Elastic Container Registry (ECR), and updates the running ECS services. This approach ensures consistent deployments and reduces the operational load for the development team. Monitoring is supported through CloudWatch logs and external uptime checks, allowing continuous tracking of service health.

The architecture is designed to scale with demand, support partner integrations, and maintain separation between internal services through network level isolation. The structure allows new components, models, and applications to be integrated without redesigning existing services.

This report serves as a reference for technical teams, administrators, and partner organizations involved in the continued development and extension of the Limpopo DT.

Introduction

The Limpopo DT supports decision making for water management activities within the Limpopo River Basin. It integrates sensor data, geospatial datasets (e.g., landuse and land cover), hydrological models (e.g., precipitation), and operational workflows into a unified digital platform that supports real time monitoring, prediction, and scenario testing (Afham et al., 2024).

This report focuses on the software and deployment architecture of the Limpopo DT. Architecture diagrams are used as the main tool for describing how system components are structured, how data flows between them, how users and external systems interact with the platform, and how services are deployed within the cloud environment (Kruchten, 1995; Henk et al., 2002). These diagrams provide a shared technical view for development, operations, and long term platform maintenance.

For this DT, architecture diagrams are essential because the platform brings together many technologies, data sources, and processing workflows. The system is designed using modular architectural principles so that components can be extended or updated without disrupting existing services. Through clearly defined architectural views, the report explains how modelling services, Internet of Things (IoT) data streams, geospatial tools, security services, and cloud infrastructure cooperate to deliver reliable and scalable decision-support capabilities.

System Overview

The Limpopo DT consists of a set of coordinated components that together support data acquisition, processing, analysis, and visualization. These components include data ingestion pipelines, geospatial processing services, hydrological and analytical modeling engines, cloud-based storage layers, application programming interfaces, and user facing applications. Each component performs a focused function while contributing to a shared system objective. The overall architecture is designed to support continuous updates, high availability, and the integration of new data sources as the platform evolves.

The system ingests data from multiple sources, including meteorological measurements, satellite imagery, hydrological observations, and administrative datasets from partner institutions. These inputs are processed through backend services and analytics pipelines that support simulation, monitoring, reporting, and planning. Processed outputs are made available to decision makers through web-based dashboards, geospatial visualization tools, and Artificial Intelligence (AI) assisted interfaces.

Architectural Design Principals

The Limpopo DT is designed around a set of core architectural principles derived from service oriented architecture practices described by Papazoglou and van den Heuvel (2007). These principles define how system components are structured, deployed, and integrated to ensure robustness, scalability, and interoperability.

As shown in Figure 1, the architecture centres on a modular microservices approach, with each capability implemented as an independent service and exposed through an API first integration model. Cloud native deployment enables elastic scalability, allowing services to scale independently based on demand. Security by design is applied across all services through standardised authentication, authorisation, and secure communication mechanisms. Interoperability with partner platforms is treated as a first class requirement to support data exchange and integration across organisational boundaries. Together, these principles guide all infrastructure and software design choices within the Digital Twin.

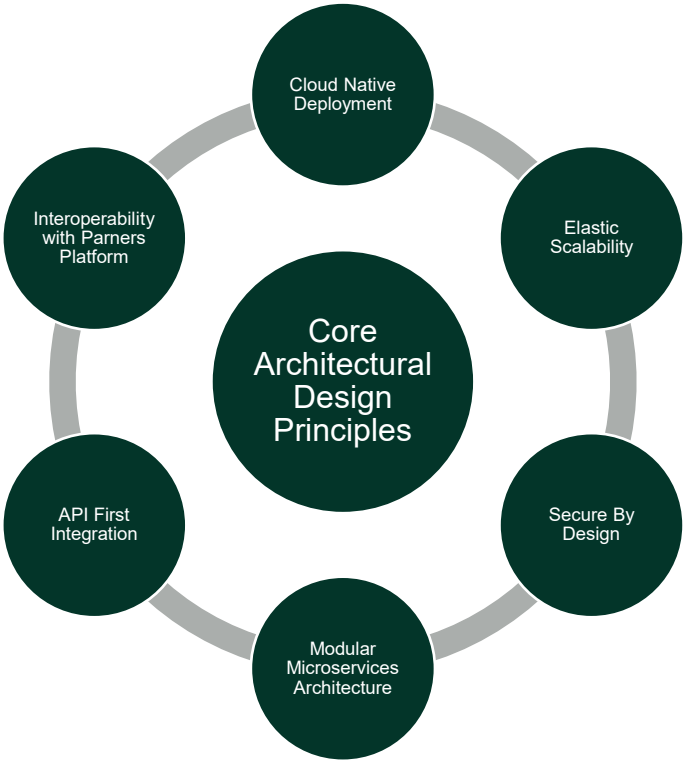


Figure 1. Core Architectural Design Principles of the Limpopo Digital Twin (*Source:* Author's creation)

High Level Platform Architecture

The Digital Twin ecosystem is composed of four logical layers:

- Data Layer
- Processing and Services Layer
- Application Layer
- DevOps and Operations Layer

Data flows from earth observation satellites, sensors, and external services into cloud processing systems, analytics engines, and visualization platforms. User interaction occurs primarily through TerriaMap (Terria, 2021) and the Water Copilot interface. A high-level view of these interactions across the platform is shown in Figure 2.

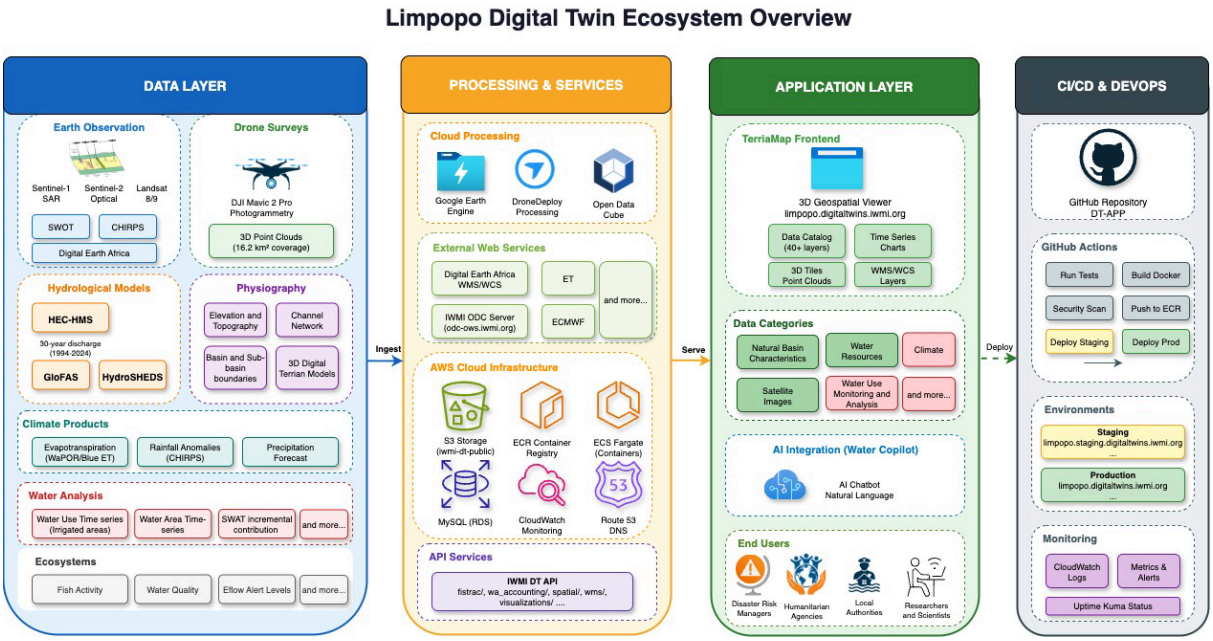


Figure 2. Limpopo Digital Twin ecosystem overview showing the data layer, processing and services, application layer, and CI CD and DevOps environment (*Source:* Author's creation)

Microservices Architecture

The Limpopo Digital Twin is implemented using a microservices architecture, in which the overall system is deconstructed into a collection of small, independent services. Each microservice is responsible for a single, well defined business function and can be developed, deployed, scaled, and maintained independently. This architectural style improves scalability, fault isolation, maintainability, and the speed of system evolution compared to monolithic designs (Alshuqayran, 2016; Newman, 2021). Communication between services is achieved through well-defined Application Programming Interfaces (APIs), which ensures loose coupling while enabling coordinated system behaviour.

The microservices are logically grouped into two main functional categories, as illustrated in Figure 3.

Data Generating Services

These services are responsible for acquiring, generating, indexing, and storing data used throughout the Digital Twin platform.

- Hydrological Modelling Service

  This service executes the Soil and Water Assessment Tool Plus (SWAT+) simulations to generate river flow forecasts and scenario based water resource assessments. The outputs support planning, risk analysis, and long term hydrological studies (Chambel et al., 2024).

- Water Quality Monitoring Service

  This service integrates Internet of Things sensor data to provide near real time assessment of water quality conditions. Observations are validated, structured, and persisted in the MySQL database for downstream analytics and visualization.

- Open Data Cube (ODC) Service

  The ODC provides large scale indexing, storage, and retrieval of multi temporal geospatial datasets. It manages earth observation products stored in cloud object storage and exposes them for analytics and visualization (Afham et al., 2024).

- External Data Integration

  This service ingests datasets from external sources such as evapotranspiration products, climate datasets, and environmental monitoring platforms. These datasets are harmonized and prepared for use by analytics and modelling services.

Analytics and Visualization Services

These services transform raw and modelled data into actionable information and deliver it to end users.

- Water Copilot Service

  This service provides an Artificial Intelligence based conversational interface that allows users to query hydrological, climate, and operational data using natural language. It supports decision making for water availability and risk assessment in the Limpopo River Basin (Vickneswaran et al., 2024).

- Analytics Service

  This service processes large scale hydrological, climate, and environmental datasets by combining outputs from modelling services, IoT streams, and external data sources. It supports reporting, trend analysis, and scenario evaluation.

- TerriaMap Dashboard Service

  This service provides the primary client interface for the DT. It integrates geospatial layers, real time sensor feeds, analytics outputs, and AI generated insights into an interactive web based mapping and visualization environment. It supports scenario analysis, temporal exploration, and decision oriented data access.

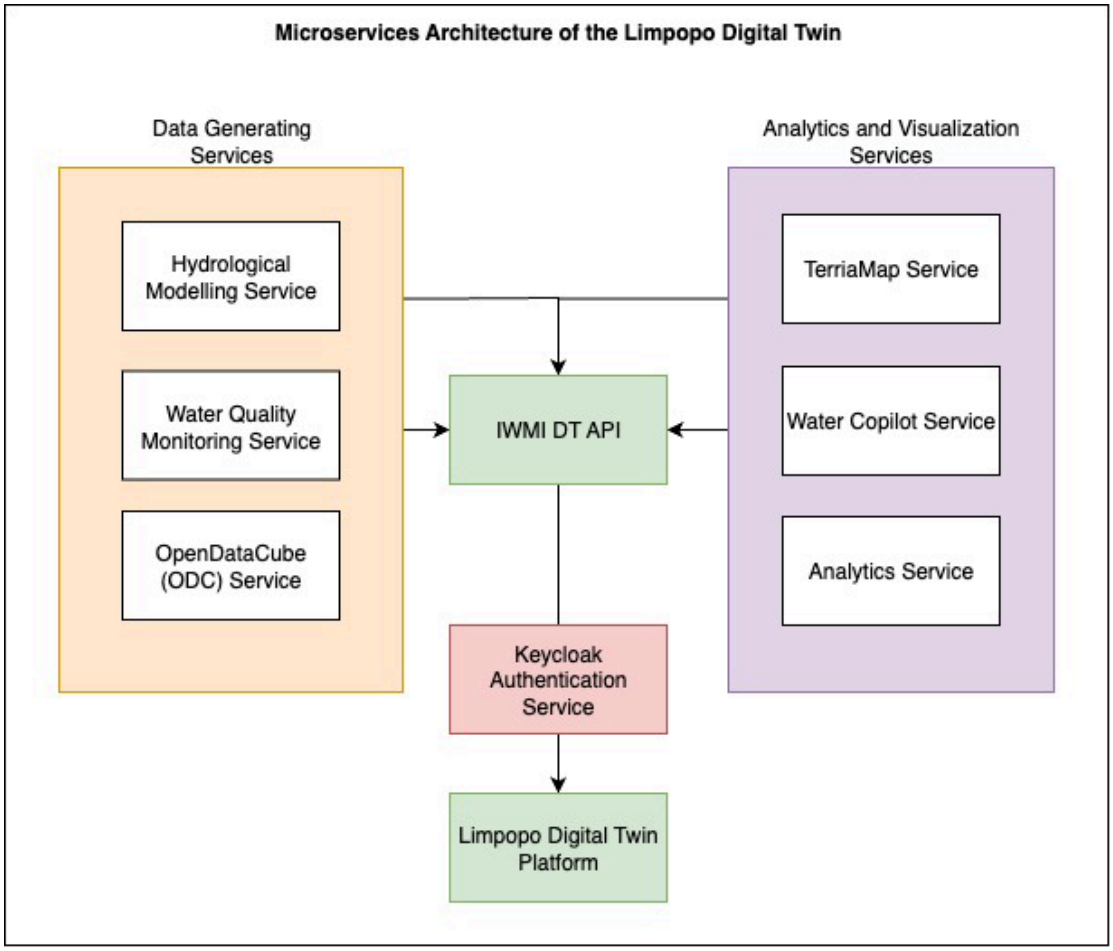


Figure 3. Microservices architecture of the Limpopo Digital Twin (*Source*: Author's creation)

API and Inter Service Communication

All microservices within the Limpopo DT communicate using Representational State Transfer (REST) based interfaces that follow the architectural principles introduced by Fielding (2000) over secure Hypertext Transfer Protocol Secure (HTTPS) connections. The architecture follows an API first design approach, where services expose well defined endpoints that can be consumed by frontend applications, partner systems, and internal processing components.

Public facing APIs are accessed through the Application Load Balancer (ALB) and secured using Open Authorization 2.0 (OAuth2) bearer tokens issued by Keycloak. Internal service to service communication occurs within the Virtual Private Cloud and relies on private networking and security group enforcement.

API payloads are exchanged using JavaScript Object Notation (JSON) format. Read intensive services such as ODC Open Web Services (OWS) operate independently from write intensive ingestion services, enabling fault isolation and horizontal scaling. This approach aligns with service oriented design principles and improves maintainability and extensibility of the overall platform (Papazoglou and van den Heuvel, 2007; Fielding, 2000).

Through this separation of concerns, each microservice remains focused on a clearly defined responsibility while contributing to the overall objectives of the DT. The use of APIs for inter service communication ensures flexibility in scaling individual components and simplifies future system extensions without disrupting existing services.

Cloud Platform Architecture

The cloud environment follows guidance from the AWS Well Architected Framework, which supports architectural decisions related to reliability, operational excellence, performance efficiency, cost optimization, and security. The Limpopo DT is deployed on a cloud-based infrastructure hosted on AWS. This environment forms the foundation for scalable computing, secure data storage, high availability, and automated operations across all platform services. The deployment is designed to support continuous service delivery, elastic scaling, and reliable access for users throughout the Limpopo River Basin region.

Compute and Container Management

Each application is packaged as a container image and stored in Amazon ECR. These images are deployed as ECS tasks that run independently for each service. This approach enables:

- Independent deployment and scaling of each microservice
- Isolation between services for improved reliability
- Rapid rollback and controlled version management

Storage and Data Management

The platform uses a layered cloud storage architecture to support different data types:

- Amazon Simple Storage Service

Used as the primary storage layer for earth observation datasets, Cloud Optimized GeoTIFFs, modelling outputs, and other unstructured data products. S3 provides durable, scalable, and cost efficient storage for large raster and time series datasets.

- Amazon Aurora MySQL

Used for structured platform data, including water quality observations, system metadata, user related records, and analytics products that require relational querying.

- Open Data Cube PostgreSQL Backend

Used for indexing metadata of multi temporal geospatial products managed by the ODC framework, supporting fast spatial and temporal discovery.

Networking and Traffic Management

- An Application Load Balancer serves as the main entry point into the DT platform. All incoming traffic from users and external systems is routed through the load balancer to the active ECS tasks.
- Domain Name Service routing maps platform subdomains, such as limpopo.digitaltwins.iwmi.org, to the load balancer, ensuring stable and predictable access for web clients and programmatic services.
- The platform operates within a VPC that separates public facing services from internal resources. Security groups and subnet isolation enforce controlled network access between microservices, databases, and storage systems.
- All inbound traffic to the ALB is inspected by AWS Web Application Firewall (WAF) before routing. This ensures that only clean, validated traffic reaches backend services.

Authentication and Identity Infrastructure

User authentication and access control across the Limpopo DT are provided by Keycloak, an open source Identity and Access Management system that supports OAuth2 and OpenID Connect protocols (Chatterjee and Prinz, 2022). Keycloak is deployed as a containerized service within the ECS Fargate environment.

The authentication flow follows a token based access model, shown in Figure 4:

1. Users authenticate through the Keycloak login interface.
2. Upon successful authentication, Keycloak issues an OAuth2 access token.
3. The frontend application attaches this token to all subsequent API requests.
4. Backend services validate tokens before granting access to protected resources.

This mechanism ensures secure service access, supports single sign on across applications, and enables role-based authorization enforcement across all platform components.

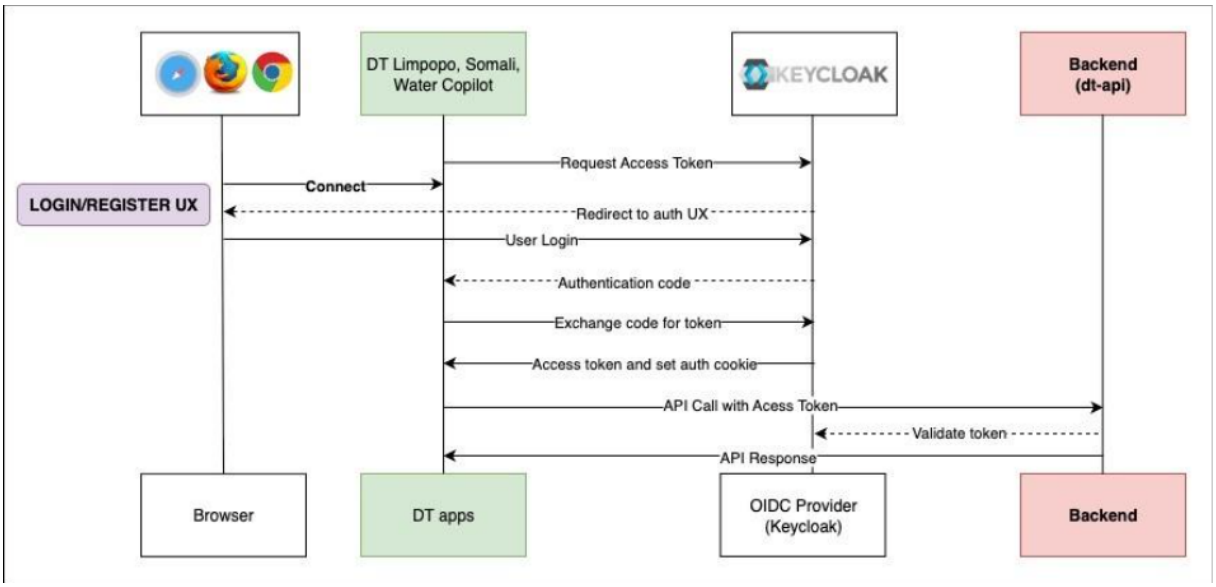


Figure 4. Single Sign On authentication flow for the Limpopo Digital Twin, showing user login, token issuance, and secure API access *(Source:* Author's creation)

Virtual Private Cloud and Subnet Segregation

All Digital Twin services operate within a dedicated Virtual Private Cloud. Public subnets are used to host the ALB, which serves as the controlled entry point for all external traffic. Private subnets host ECS services, analytics engines, Keycloak authentication services, and supporting platform components.

The ODC PostgreSQL database and the Amazon Aurora MySQL database are deployed in isolated private subnets with no direct internet exposure. Network access to these resources is restricted through tightly scoped security groups that only permit traffic from authorized ECS services.

Outbound internet access for ECS services is controlled through managed network gateways and explicit security rules. This design minimizes attack surfaces while maintaining required external connectivity for data ingestion and service operations.

Secrets, Monitoring, and Observability

AWS Secrets Manager stores sensitive configuration values such as database credentials, API keys, private keys, and authentication secrets. These secrets are injected dynamically into ECS tasks at runtime.

Amazon CloudWatch collects system logs, service level metrics, and operational health indicators across all microservices. This enables centralized monitoring, diagnostics, and performance tracking.

External Availability Monitoring public endpoints are monitored using Uptime Kuma to provide independent verification of service availability.

Automated Deployment and DevOps Integration

All services are deployed using a fully automated Continuous Integration and Continuous Deployment pipeline powered by GitHub Actions. The deployment pipeline performs the following:

- Builds and tags service container images
- Pushes images to Amazon ECR
- Updates ECS task definitions
- Triggers zero downtime service updates

This approach ensures that the production environment remains tightly synchronized with the source code while minimizing manual operational overhead. Figure 5 illustrates the complete Continuous Integration and Continuous Deployment (CI/CD) workflow used by the Digital Twin platform. The process starts with developer commits to the DT-APP repository, triggering automated GitHub Actions that run tests, perform security scans, build and tag Docker images, and push them to Amazon ECR. These images are then deployed to ECS Fargate in both staging and production environments, with CloudWatch providing continuous monitoring.

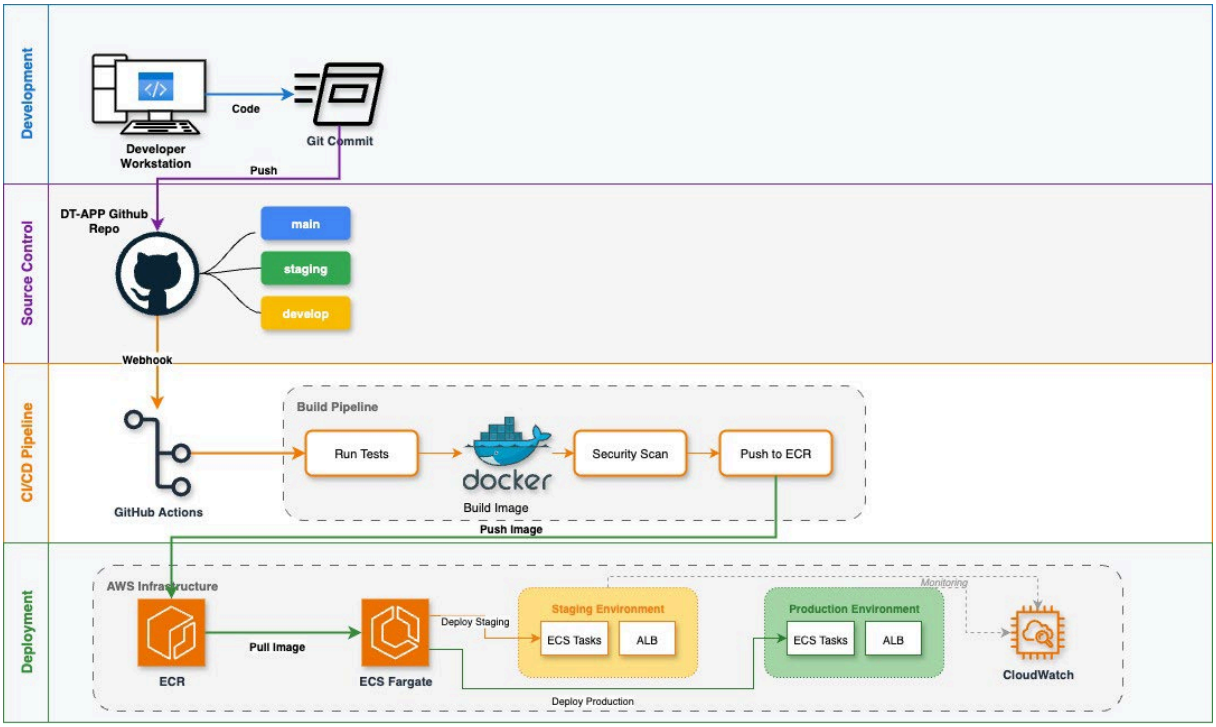


Figure 5. CI/CD workflow for the Limpopo Digital Twin showing development, source control, automated build steps, container image publishing, and deployment to staging and production environments. (*Source:* Author's creation)

Regional Deployment Strategy

All infrastructure is deployed in the AWS af-south-1 (Cape Town) region, which provides low latency connectivity for users within Southern Africa. The deployment spans multiple availability zones to increase fault tolerance, maintain service continuity, and support resilient operation of all DT services as illustrated in Figure 6.

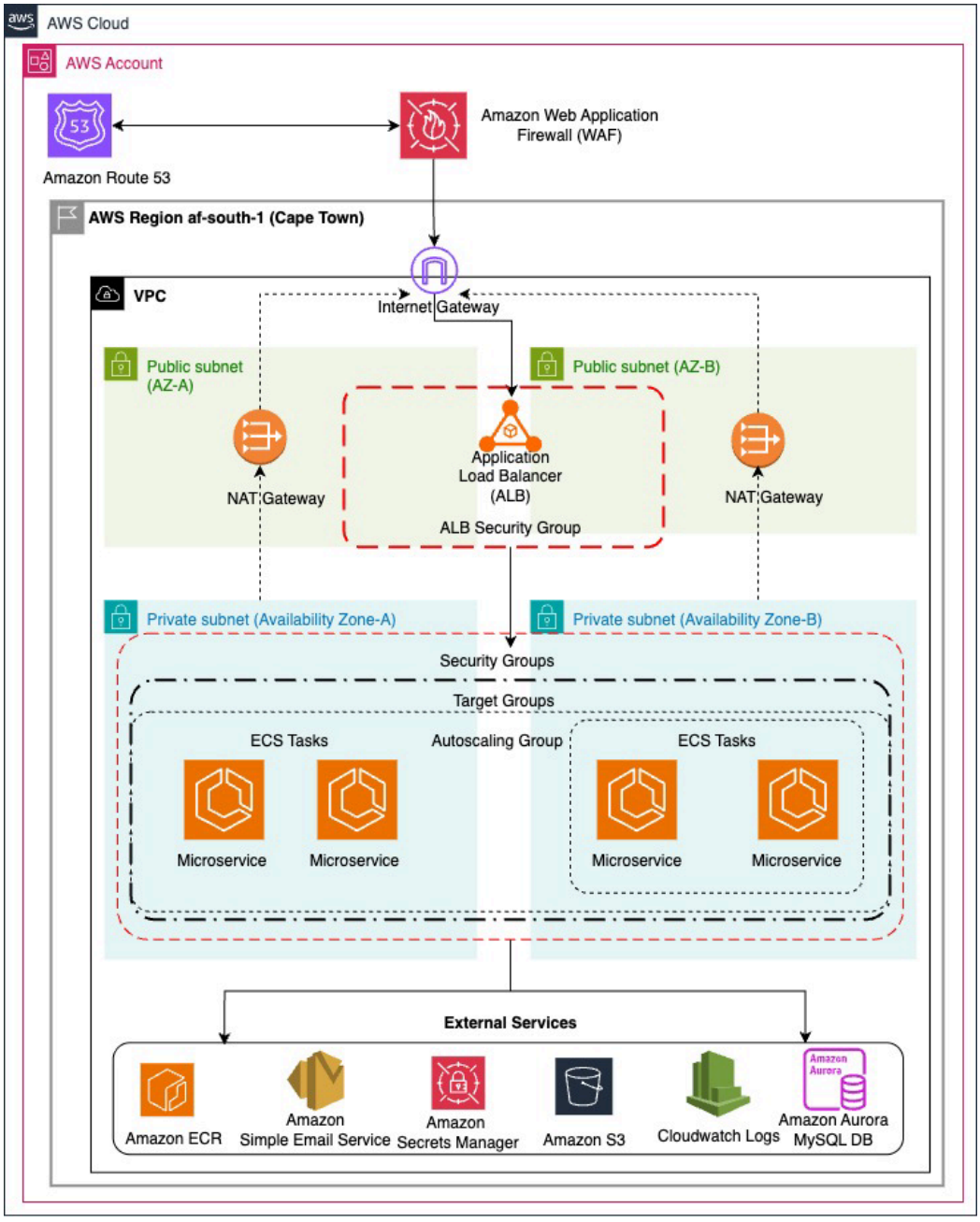


Figure 6. AWS Deployment Architecture of the Limpopo DT, illustrating the multi-zone VPC, compute, and data services (S3 and MySQL) deployed across two Availability Zones in the af-south-1 region. *(Source*: Author's creation)

Application and Frontend Architecture

The Application Architecture defines how users interact with the Limpopo DT through web-based interfaces and how the frontend communicates with backend services. As illustrated in Figure 7, the architecture follows a client server model in which the frontend applications consume secured APIs exposed by the DT backend services.

At the core of the frontend layer is TerriaMap, which serves as the primary geospatial visualization interface. TerriaMap provides interactive maps, temporal controls, charting, and scenario exploration capabilities for hydrological, environmental, and climate data. The application consumes Open Geospatial Consortium (OGC) compliant services, including Web Map Service (WMS) and Web Map Tile Service (WMTS) endpoints exposed through ODC OWS, as well as REST based APIs provided by the IWMI DT API and analytics services.

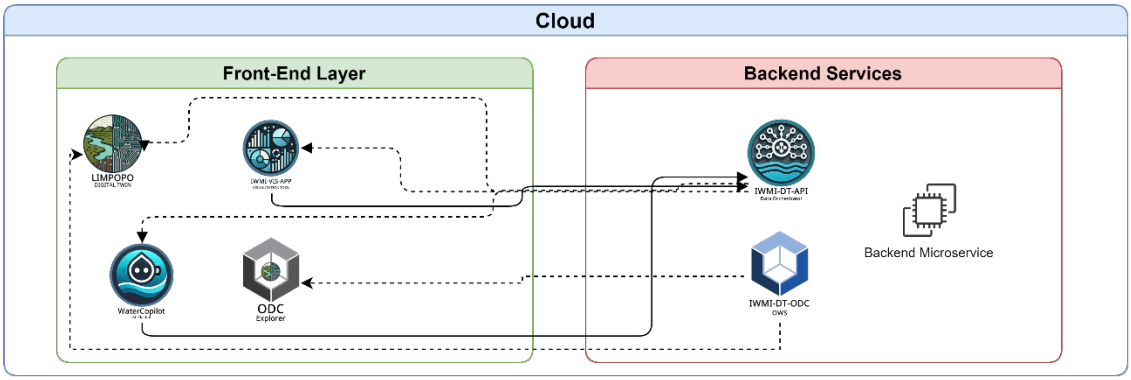


Figure 7. High-Level Application Architecture illustrating the separation between the Front-End Layer and the Backend Services. *(Source:* Author's creation)

TerriaMap is configured using JSON based catalogue and environment configuration files. These configuration files define:

- Data source endpoints
- Dataset metadata and grouping
- Default geographic extents and visualization parameters
- Feature activation and user interface behavior

The configuration files are stored on Amazon S3 and are loaded dynamically by the application at runtime. This approach allows system administrators to update datasets, services, and visualization behavior without requiring code level changes or application redeployment.

Staging and Production Environments

The frontend follows a dual environment deployment strategy, consisting of:

- **Staging Environment**, used for testing new datasets, API changes, configuration updates, and feature enhancements.
- **Production Environment**, used for live operational decision support.

Each environment loads a distinct set of JSON configuration files from separate S3 locations. This ensures that staging experiments and validation activities do not interfere with live production operations. Backend API endpoints, authentication realms, and dataset catalogs are also environment specific.

Authentication and Secure Access

All frontend applications are protected by Keycloak based authentication. Users are required to authenticate before accessing secured APIs or datasets. After authentication, Keycloak issues OAuth2 access tokens that are attached to all API requests made by the frontend. This mechanism ensures:

- Secure access to protected services
- Role based access control
- Unified single sign on across Digital Twin applications

Frontend Technology Stack

Custom extensions to TerriaMap are developed using React JS and external libraries, enabling:

- Interactive charts and dashboards
- Advanced dataset filtering
- Integration with AI based services such as Water Copilot
- Enhanced scenario exploration tools

Through this architecture, the frontend layer remains fully decoupled from backend services, relies entirely on secure APIs for data access, and supports controlled deployment across staging and production environments.

Data Architecture

The data architecture of the Limpopo DT is designed to support the ingestion, processing, storage, and discovery of large volumes of geospatial, hydrological, and time series data. The platform manages heterogeneous data types, including raster imagery, vector datasets, sensor observations, tabular records, and metadata.

Primary data sources include Earth observation satellites, IoT sensor networks, hydrological models, and administrative datasets from partner institutions. Raw raster products are stored in Amazon S3 as Cloud Optimized GeoTIFFs, enabling efficient partial reads and scalable access by analytics and visualization services. Each dataset is accompanied by EO3 compliant metadata in YAML format, which defines spatial extent, temporal coverage, and band structure.

The ODC PostgreSQL backend serves as the metadata indexing layer, enabling fast spatial and temporal search across multi temporal raster products (Kopp et al., 2019). Structured relational data, including sensor registrations, water quality measurements, authentication mappings, and operational records, are stored in Amazon Aurora MySQL.

This layered approach separates physical storage from logical indexing and service access, allowing high throughput analytics, efficient visualization, and long-term data preservation within a unified architecture (Figure 8).

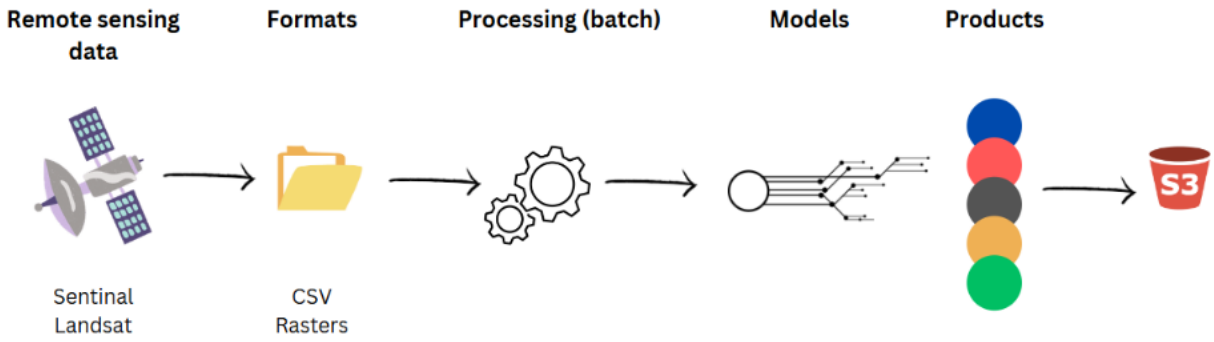


Figure 8. Data flow from the spatial products into the storage environment, showing how each component feeds the final repository in S3. *(Source*: Author's creation)

Observability, Logging and Reliability

Operational visibility and system reliability are supported through a combination of centralized logging, metrics collection, service health monitoring, and external availability verification.

Amazon CloudWatch collects logs from all ECS services, including application logs, authentication events, and system level diagnostics. CloudWatch metrics provide real time insight into CPU utilization, memory usage, request throughput, and service health across all running tasks.

The ALB performs continuous health checks on each service endpoint and automatically removes unhealthy tasks from active routing. Public service availability is independently monitored using Uptime Kuma (Louis Lam, 2025), which provides external endpoint verification and uptime reporting.

Together, these mechanisms enable fault detection, rapid diagnostics, and automated service recovery, ensuring continuous operational reliability across the Digital Twin platform.

Performance and Scalability

The Limpopo Digital Twin architecture is designed to scale horizontally in response to changing computational demands. All core services operate as stateless ECS tasks, allowing independent scaling without service coupling. As processing loads increase, additional task replicas can be provisioned automatically to maintain system responsiveness.

Large scale raster data access is optimized using Cloud Optimized GeoTIFFs and Open Data Cube indexing, significantly reducing spatial query latency for visualization and analytics. Read heavy access patterns are separated from ingestion operations to avoid performance contention.

The platform is designed to support near real time data updates for hydrological monitoring and scenario analysis. While the current deployment operates within a single AWS region, future enhancements may introduce multi region redundancy for extended disaster recovery capabilities.

Security and Access Control

Network Security

All Digital Twin services operate within secured Virtual Private Clouds with strict subnet isolation and security group enforcement. Public access is limited to the ALB, while all backend services and databases operate in private subnets. Encrypted communication is enforced on all external interfaces using transport layer security (TLS).

AWS Web Application Firewall
AWS WAF protects the platform's public-facing endpoints by filtering malicious traffic before it reaches the ALB. It enforces managed rule groups to block common attack patterns such as SQL injection and cross-site scripting. Custom rules are configured to allow necessary Keycloak authentication flows and TerriaMap redirects while preventing unauthorized access. WAF operates as the first layer of inbound protection for all DT web applications.

Application Security

Application level access control is enforced through Keycloak based authentication and role based authorization. Only authorized users are permitted to modify critical datasets and system configurations. API level security enforces token validation on every request.

Data Protection

Sensitive datasets remain in private Amazon S3 buckets until explicit clearance is granted for publication. Encryption at rest and encryption in transit are enforced across all databases and storage layers.

Limitations and Future Enhancements

The current deployment does not yet implement Infrastructure as Code (IaC) for automated provisioning of cloud resources, environments, and configuration policies. Introducing IaC would improve repeatability and strengthen governance across staging and production systems.

Multi region redundancy is not yet enabled, and the present disaster recovery strategy relies on backups stored within a single AWS region (Cape Town). Expanding the architecture to support cross region replication and failover would improve resilience, particularly for mission critical services supporting basin wide operations.

Additional enhancements planned for future releases include stronger automation for security enforcement, extended monitoring coverage, and improved scalability for analytics services as data volumes grow.

Conclusion

The Limpopo Digital Twin integrates modular services, stable data pipelines, advanced modelling tools, and a flexible geospatial framework. The system is designed to grow through the addition of new datasets, models, and applications, allowing it to adapt as partner needs and basin management priorities evolve. Its architecture supports operational decision-making, long-term planning, and real time monitoring across the region.

The platform provides a dependable foundation for integrated water resource management and strengthens collaboration among member states and technical partners. It also establishes a scalable base on which future digital innovations can be developed, ensuring continued improvement in data driven water management for the Limpopo River Basin.

References

Afham, A.; Silva, P.; Ghosh, S.; Kiala, Z.; Retief, H.; Dickens, C.; Garcia Andarcia, M. 2024. *Limpopo River Basin Digital Twin Open Data Cube Catalog*.

Amazon Web Services. AWS Well Architected Framework. Whitepaper. Retrieved December 1, 2025 from https://docs.aws.amazon.com/pdfs/wellarchitected/latest/framework/wellarchitected-framework.pdf

Chambel-Leitão, P.; Santos, F.; Barreiros, D.; Santos, H.; Silva, Paulo; Madushanka, Thilina; Matheswaran, Karthikeyan; Muthuwatta, Lal; Vickneswaran, Keerththanan; Retief, H.; Dickens, Chris; Garcia Andarcia, Mariangel. 2024. Operational SWAT+ model: advancing seasonal forecasting in the Limpopo River Basin.

Chatterjee, A., & Prinz, A. (2022). Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study. Sensors, 22(5), 1703. https://doi.org/10.3390/s22051703

Garcia Andarcia, M.; Dickens, C.; Silva, P.; Matheswaran, K.; Koo, J. 2024. Digital twin for management of water resources in the Limpopo River Basin: a concept. Colombo, Sri Lanka: International Water Management Institute (IWMI). CGIAR Initiative on Digital Innovation Working Paper 5. 4p. doi:10.5337/2024.218

Henk Koning, Claire Dormann, and Hans van Vliet. 2002. Practical guidelines for the readability of IT-architecture diagrams. In Proceedings of the 20th annual international conference on Computer documentation (SIGDOC '02). Association for Computing Machinery, New York, NY, USA, 90–99. https://doi.org/10.1145/584955.584969

IWMI 2025. IWMI DT API Documentation. Retrieved December 1, 2025, from https://api.digitaltwins.iwmi.org

Kopp, S.; Becker, P.; Doshi, A.; Wright, D. J.; Zhang, K.; Xu, H. 2019. Achieving the full vision of earth observation data cubes. *Data* 4(3): 94.

Louis Lam 2025. Uptime Kuma, Github repository. Retrieved September 6, 2025, from https://github.com/louislam/uptime-kuma.

N. Alshuqayran, N. Ali and R. Evans, "A Systematic Mapping Study in Microservice Architecture," 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), Macau, China, 2016, pp. 44-51, doi: 10.1109/SOCA.2016.15.

Newman, S. 2021. Building microservices: designing fine-grained systems. Sebastopol, USA: O'Reilly Media, Inc.

Oluwatosin, H. S. (2014). Client-server model. IOSR Journal of Computer Engineering, 16(1), 67-71.

Papazoglou, M.P., van den Heuvel, WJ. Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 16, 389–415 (2007). https://doi.org/10.1007/s00778-007-0044-3

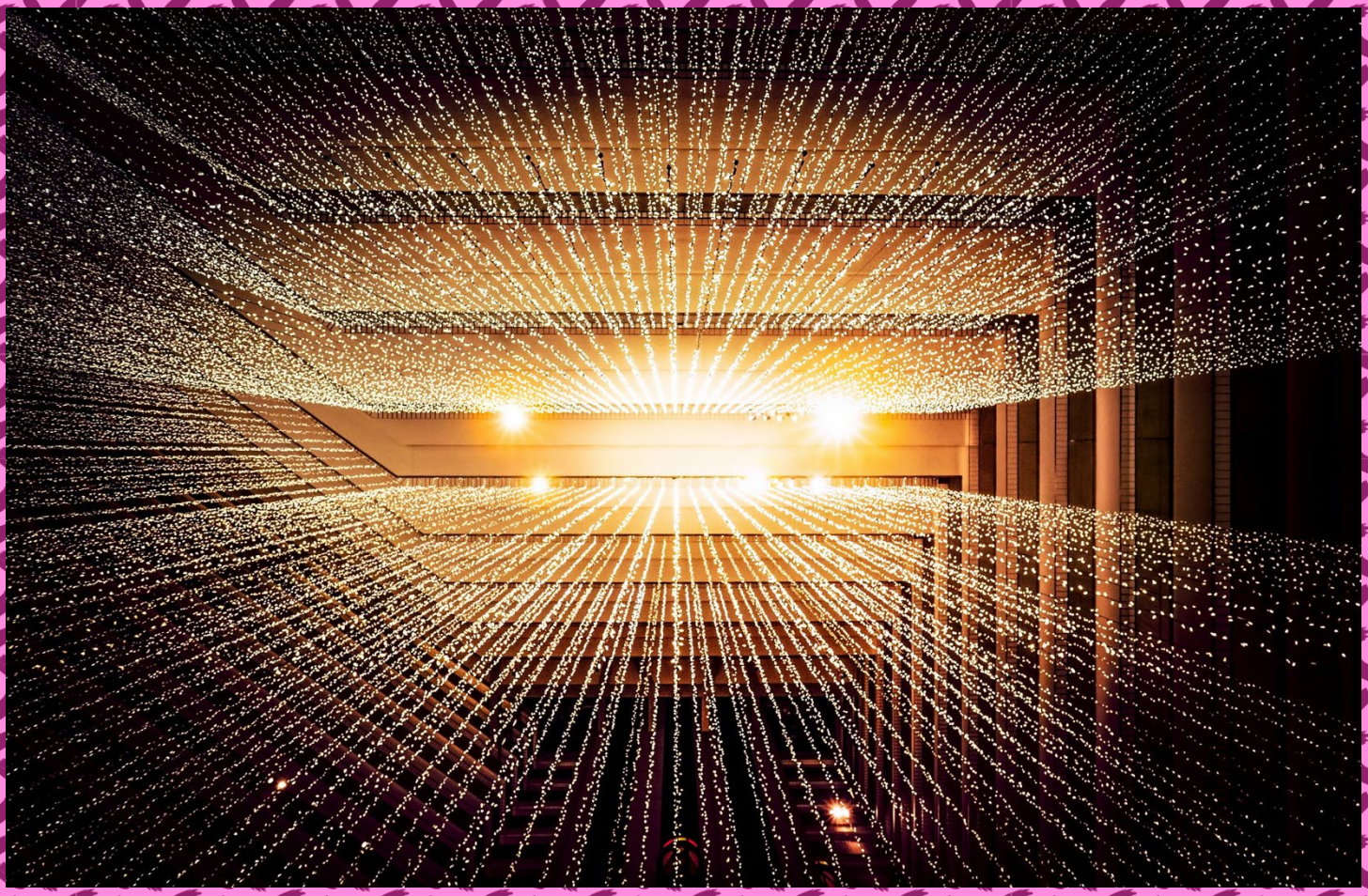
Kruchten, P.B., "The 4+1 View Model of architecture," in IEEE Software, vol. 12, no. 6, pp. 42-50, Nov. 1995, doi: 10.1109/52.469759.

Tao, F., and Qi. 2019. "Make More Digital Twins." *Nature* 573 (7775): 490–91.

Terria 2021. TerriaMap, Github repository. Retrieved September 1, 2025, from https://github.com/TerriaJS/TerriaMap.

Vickneswaran, K.; Retief, H.; Padilha, R.; Dickens, C.; Silva, P.; Ghosh, S.; Garcia Andarcia, M. 2024. WaterCopilot: a water management AI virtual assistant for the Limpopo River Basin Digital Twin - user guide V0 202410.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E. and Bouwman, J., 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific* (1):1-9.



CGIAR is a global research partnership for a food-secure future. CGIAR science is dedicated to transforming food, land, and water systems in a climate crisis. Its research is carried out by 13 CGIAR Centers/Alliances in close collaboration with hundreds of partners, including national and regional research institutes, civil society organizations, academia, development organizations and the private sector. www.cgiar.org

To learn more about this and other Science Programs and Accelerators in the CGIAR Research Portfolio 2025–2030, please visit www.cgiar.org/cgiar-research-porfolio-2025-2030/

Contact
Mariangel Garcia Andarcia, Research Group Leader - Water Futures Data & Analytics (WFDA), International Water Management Institute (IWMI), Colombo, Sri Lanka (M.GarciaAndarcia@cgiar.org)

CGIAR DIGITAL TRANSFORMATION

IWMI International Water Management Institute