

AI-Assisted River Discharge Measurement Through Citizen Science and Mobile Technology

Kayathri Vigneswaran, Hugo Retief, Jai Clifford-Holmes, Hansaka Tennakoon and Mariangel Garcia Andarcia

December 2025



Authors

Kayathri Vigneswaran¹, Hugo Retief², Jai Clifford-Holmes², Mariangel Garcia Andarcia¹, Hansaka Tennakoon¹

¹International Water Management Institute (IWMI), Colombo, Sri Lanka

²Association for Water and Rural Development (AWARD), Hoedspruit, South Africa

Acknowledgments

This work was conducted as part of the CGIAR Accelerator for Digital Transformation. We would like to thank all funders who supported this research through their contributions to the CGIAR Trust Fund (www.cgiar.org/funders). Forming part of the Citizen Science for Water Management in Limpopo River Basin initiative, implemented within the Digital Innovations for Water Secure Africa (DIWASA) project and the CGIAR Accelerator for Digital Transformation, this work was made possible through the financial support of the Belgian development agency (Enabel), the Leona M. and Harry B. Helmsley Charitable Trust, the United Nations Development Programme (UNDP) through the Global Environment Facility (GEF), and the Microsoft Corporation. We extend our appreciation to the Limpopo Watercourse Commission (LIMCOM) for their ongoing partnership in advancing transboundary water resource management across the basin. We also thank our implementing partners, GroundTruth and the Association for Water and Rural Development (AWARD), for their contributions to citizen science capacity building and technical development.



CGIAR Accelerator for Digital Transformation

Digital Transformation “co-creates inclusive solutions leveraging advancements in AI, machine learning, modeling and big data analytics” to improve decision-making across food, land and water systems. It supports responsible, AI-enabled research and digital services that help partners design evidence-based policies, investments and innovations for climate-resilient development.

Citation

Vigneswaran, K.; Retief, H.; Clifford-Holmes, J.; Garcia Andarcia, M.; Tennakoon, H. 2025. *AI-assisted river discharge measurement through citizen science and mobile technology*. Colombo, Sri Lanka: International Water Management Institute (IWMI). CGIAR Accelerator for Digital Transformation. 53p.

© 2025 International Water Management Institute. Some rights reserved. This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International License (CC by 4.0).

Front cover photo: AI citizen scientists using a WhatsApp-based Smart Gauge to photograph a river gauge plate on a river (composite of IWMI app/AI generated imagery). (graphic: IWMI)

Back cover photo: GroundTruth staff Nkosingithandile Sithole, Ayanda Lephane and Nick Pattinson (from left to right) reviewing the MiniSASS application during field training in South Africa. (photo: GroundTruth)

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. Boundaries used in the maps do not imply the expression of any opinion whatsoever on the part of CGIAR concerning the legal status of any country, territory, city, or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. Borders are approximate and cover some areas for which there may not yet be full agreement.

Contents

| | |
|-----------------------------------|----|
| Contents | 2 |
| Executive Summary | 3 |
| Introduction | 4 |
| Protocol Design Framework | 9 |
| System Architecture | 12 |
| API Endpoint Reference | |
| AI Processing Pipeline | 19 |
| Data Model | 27 |
| WhatsApp Bot Integration Workflow | 29 |
| Security Considerations | 39 |
| Discussion and Conclusions | 40 |
| References | 42 |

LIST OF TABLES

LIST OF FIGURES

Summary

This technical report documents the development of a novel protocol for measuring river discharge through the integration of artificial intelligence and citizen science participation. The work was undertaken within the Enabel/Wehubit 'Citizen Science for Water Management in Limpopo River Basin' project, addressing the need for accessible discharge measurement protocols in data-scarce basins. Implemented as part of the broader LIMCOM-UNDP/GEF programme for Integrated Transboundary River Basin Management. The protocol seeks to address a fundamental challenge in hydrological monitoring: basins with the weakest observational infrastructure stand to benefit most from digital decision support systems, yet lack the data required to drive them.

The Vision API serves as the backend infrastructure for a WhatsApp-based citizen science platform, enabling community members to contribute water level readings by photographing gauge plates at monitoring stations across the Limpopo Basin's four riparian countries. A custom WhatsApp Bot Service deployed on AWS ECS manages the conversational interface, guiding users through image submission, station selection, location sharing, and result validation without requiring dedicated mobile app installation.

The system uses a two-step AI approach to read gauge-plate photos: it first identifies the waterline and scale on the image, and then converts this into a water-level reading. This design improves reliability under real field conditions and achieved strong accuracy in testing ($R^2 = 0.84$; average error 5.43 cm on high-quality images).. Validated water level readings are converted to discharge values using station-specific rating curves accessed through the FlowTracker API.

Key achievements documented in this report include the successful implementation of a three-stage AI processing pipeline achieving reliable gauge reading extraction across variable field conditions; a WhatsApp-first design prioritising digital inclusiveness for users with limited connectivity, storage, or technical familiarity; a two-stage validation workflow that ensures data quality while continuously generating ground-truth data for model improvement; The platform includes secure user access with permissions linked to specific monitoring stations in line with LIMCOM governance, and it is hosted on scalable cloud infrastructure to support the project's expected user uptake..

The protocol demonstrates that citizen science, combined with hybrid artificial intelligence, can meaningfully augment traditional hydrological monitoring in data-scarce transboundary basins while maintaining scientific rigour through embedded quality assurance.

Introduction

Project Context

This technical report documents the development of a novel protocol for measuring river discharge through the integration of mobile applications, artificial intelligence, and citizen science participation. The work forms a core component of the "Citizen Science for Water Management in Limpopo River Basin" project, an 18-month initiative funded by Enabel through the Wehubit programme's call for "Data-driven Digital Social Innovations in Africa" (IWMI 2024). Led by the International Water Management Institute (IWMI) in partnership with GroundTruth and the Association for Water and Rural Development (AWARD), the project aims to develop a citizen science water monitoring prototype that has direct impact on water resource management across the transboundary Limpopo Basin.

The initiative represents part of the broader "Integrated Transboundary River Basin Management for the Sustainable Development of the Limpopo River Basin" programme, implemented by the Limpopo Watercourse Commission (LIMCOM) in partnership with the Global Water Partnership Southern Africa (GWPSA), with support from the United Nations Development Programme (UNDP) through funding from the Global Environment Facility (GEF).

Background and Rationale

Accurate and timely river discharge data are fundamental to effective water resource management, flood forecasting, and drought monitoring (Beven 2012). However, many river basins—particularly in developing regions—suffer from declining hydrometric networks due to infrastructure degradation, limited financial resources, and institutional capacity constraints (Hannah et al 2011). The Limpopo River Basin, Southern Africa's fourth largest international basin shared by Botswana, Mozambique, South Africa, and Zimbabwe, exemplifies these challenges with a gradual decline in the number of active stations and data records (Figure 1). Sparse hydrological data, uneven monitoring infrastructure, and limited institutional capacity make collecting data across national borders a persistent challenge, with operational gauging stations providing insufficient spatial and temporal coverage for robust decision-support systems (Figure 2)(LIMCOM 2019).

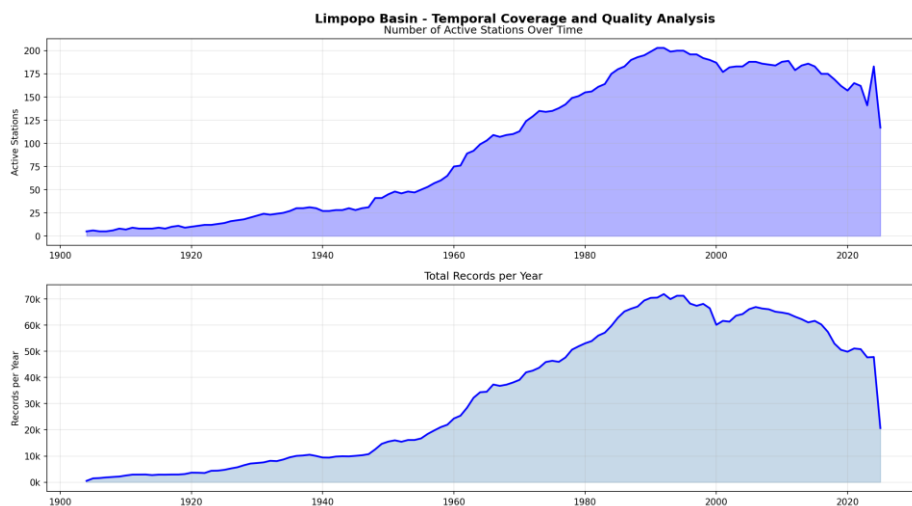


Figure 1 Temporal coverage of the Department of Water Affairs and Sanitation (DWS) of South Africa hydrological monitoring network in the South African part of the Limpopo River Basin.

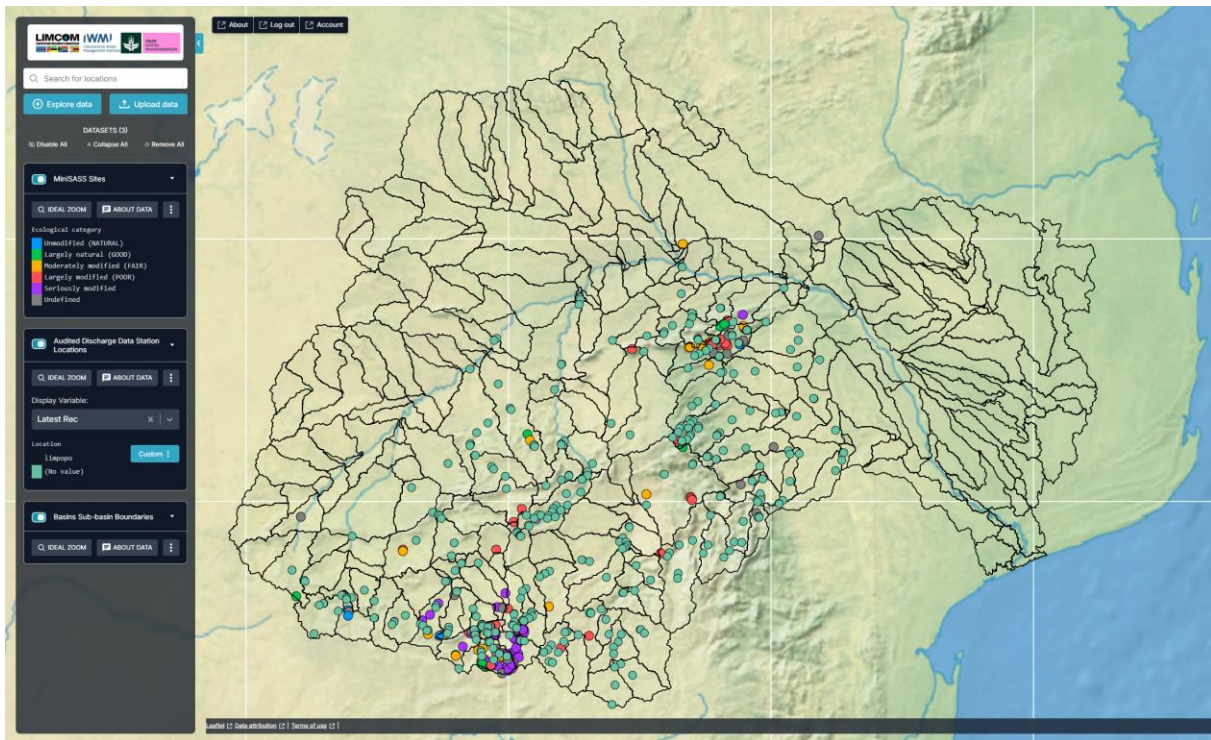


Figure 2 Screenshot of the Limpopo Digital Twin showing the locations of river hydrological monitoring stations and citizen science monitoring stations [Source: IWMI]

Traditional river monitoring methods relying on pressure probes and expensive cabling often incur high installation and maintenance costs, typically requiring calibration at intervals of up to two weeks. These constraints are particularly acute in transboundary settings where coordinated infrastructure investment remains challenging. Imagery-based solutions can significantly reduce these costs while providing continuous, verifiable field data and enabling more frequent validation.

The emergence of Digital Twin technology offers transformative potential for water resource management by creating virtual representations of physical systems that integrate real-time data streams with simulation models (Rasheed et al 2020). IWMI has developed a Digital Twin for the Limpopo Basin (Garcia *et al.* 2024 and Afham 2024) that provides an advanced virtual representation enabling water managers to visualize real-time data, model watershed processes, and generate forecasts on water availability and quality. However, the effectiveness of Digital Twins depends critically on data availability and quality. This creates a paradox: the basins most in need of advanced decision-support tools are often those with the weakest monitoring infrastructure.

Citizen science presents a promising approach to address this data gap by engaging community members in systematic observation and data collection (Buytaert et al 2014). Within the LIMCOM Digital Twin framework, citizen scientists play a crucial role by contributing local data on water quality, flow levels, and ecosystem indicators that complement official datasets and fill critical gaps. Such community-generated data are stored in shared databases and accessed by the Digital Twin through Application Programming Interfaces (APIs) that enable smooth integration with other monitoring systems. Critically, citizen observations capture localized data on subtle shifts in streams, tributaries, and water use patterns that might otherwise go unnoticed (Langa and Kiala 2025).

When combined with artificial intelligence for quality control and data extraction, citizen-contributed observations can augment traditional monitoring networks cost-effectively while building local capacity and awareness (See et al 2016). The proliferation of smartphones with high-quality cameras and the near-universal adoption of messaging platforms like WhatsApp create unprecedented opportunities for mobile-based citizen science initiatives in water resource monitoring.

The Citizen Science Water Monitoring Framework

The broader Enabel-funded project deploys three primary citizen science tool groups across the Limpopo Basin:

1. MiniSASS (Mini Stream Assessment Scoring System): Enables communities to monitor river health by sampling aquatic macroinvertebrates, providing biological indicators of water quality. The MiniSASS

smartphone application digitizes and streamlines ground surveys, using AI vision recognition to confirm measurements taken by citizen scientists.

2. **Clarity Tube and Water Quality Tools:** The clarity tube measures water turbidity, providing rapid insights into water quality that complement laboratory analyses. Additional apparatus expand the range of water quality parameters collected.
3. **AI-Driven Stream Flow Monitoring:** Utilizing AI and photo recognition, this tool monitors stream flows through gauge plate readings, enabling automated extraction of water level data from citizen-submitted photographs. This protocol is the focus of this report.

Photographs provide an auditable record that can be reviewed, moderated, and reprocessed as the models improve, whereas manual entry alone is difficult to verify after the fact. AI assistance also materially reduces interpretation error in practice: gauge plates are not intuitive for occasional users, and even after a brief explanation, workshop exercises have shown wide variation in readings among participants compared with trained technicians. By providing an on-screen estimate and highlighting the detected waterline, the system guides users toward a correct reading, while still allowing human confirmation or correction as part of quality assurance.

Importantly, the same AI approach can be applied beyond WhatsApp submissions to low-cost, fixed camera deployments at stations, offering a scalable alternative to more expensive in-river instrumentation (e.g., pressure probes) where budgets and maintenance capacity are constrained.

These tools are integrated within the EnviroChamps citizen science model, where participants collect water resource data and receive recognition through UNICEF's YOMA platform—a blockchain-based system that records skills and contributions in a digital CV while providing tangible incentives such as training vouchers (IWMI, 2025). The project aims to establish a transboundary network of at least 80 active citizen scientists contributing 10,000 data points annually across the basin.

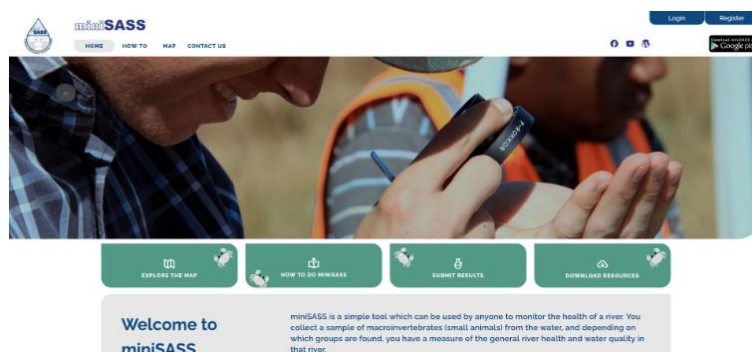


Figure 3 MiniSASS Landing Page the gateway to various citizen science based tool and methods (<https://minisass.org/>).

Objectives and Scope

This report documents the protocol and system developed for river discharge measurement through citizen science and AI integration. Specific objectives include:

- Design a protocol/application for measurement of discharge using gauge plate recognition
- Develop Mobile/WhatsApp integration enabling citizen science participation
- Support the development and training of AI models for automated gauge reading extraction
- Integrate citizen-contributed data with the broader Digital Twin framework

The protocol includes a functional Mobile/WhatsApp integration with citizen science capabilities which has been achieved through the development of the Vision API documented in this report.

Approach and Innovation

The protocol developed represents a novel integration of several technological components that responds directly to the project's emphasis on digital inclusiveness and accessibility. Rather than requiring users to install a dedicated mobile application, the system leverages WhatsApp—a platform already familiar to most potential citizen scientists in the region—as the primary user interface. This design decision significantly lowers barriers to participation while ensuring the system functions on basic smartphones with limited storage capacity. As Roman (2025) notes, the approach seeks to "democratise data collection" while "overcoming the digital divide by teaching citizen scientists how to incorporate digital technologies."

The AI processing pipeline implements a hybrid architecture combining computer vision and large language model capabilities through three sequential stages:

- **Stage 1 — Waterline Detection (YOLOv8):** YOLOv8 is a computer-vision model that detects objects in images; in this system it is used to find the gauge plate in a photo and locate the waterline on the scale. A YOLOv8 object detection model localises the gauge plate within the submitted photograph and identifies the precise point where the water surface intersects the scale. Angular correction using Sobel gradient analysis compensates for camera misalignment, while Canny edge detection refines the waterline position to pixel-level accuracy.
- **Stage 2 — Scale Gap Estimation (YOLOv8-Pose):** A YOLOv8-Pose model performs keypoint localisation to detect individual scale markers along the gauge face. The median pixel distance between consecutive markers (D_m) and the distance from the nearest marker to the waterline (D_n) are calculated, yielding a scale gap ratio ($R = D_n / D_m$) that enables sub-interval precision.
- **Stage 3 — Reading Extraction (Gemini 2.0 Flash):** Google Gemini's multimodal large language model analyses the cropped gauge image to extract visible numbers and their sequence. The final water level reading (W) is calculated algorithmically ($W = M - R \times 10$), combining the LLM-extracted major scale value (M) with the geometric scale gap ratio (R) from Stage 2.


This hybrid approach leverages the geometric precision of computer vision models for spatial measurement while exploiting the contextual reasoning capabilities of large language models for interpreting potentially ambiguous scale markings. Critically, the architecture does not rely solely on end-to-end AI interpretation—the algorithmic combination of outputs from both model types achieves substantially higher accuracy than either approach in isolation. Subsequent conversion of validated gauge readings into volumetric discharge leverages established rating curves maintained through the FlowTracker system, enabling end-to-end automation of river flow estimation.

Quality assurance is embedded throughout the workflow through a two-stage user validation process. Validation is primarily a quality-check step, not a requirement that users can reliably read gauge plates unaided. The AI provides an initial estimate and highlights the detected waterline to guide the user; the confirmation (or correction) prevents misreadings from entering the database, creates labelled examples for ongoing model improvement, and preserves an auditable record that can also support future fixed-camera deployments. Citizens are asked to confirm both the detected waterline position and the extracted numerical reading, providing ground-truth data for continuous model improvement while ensuring erroneous readings are flagged before entering the operational database. This approach acknowledges both the capabilities and limitations of current AI systems while ensuring data quality sufficient for operational use.

Initial model evaluation demonstrates robust predictive capability: Gemini Stage 2 processing (image with scale gap metadata) achieved a coefficient of determination (R^2) of 0.84 with MAE of 5.43 cm for optimal quality imagery, representing approximately 91% of the evaluation dataset (Vigneswaran et al 2025). Performance on all images combined yielded $R^2 = 0.58$, with accuracy degradation observed under challenging conditions including blur, corrosion, and scale occlusion. Notably, processing with image alone (Stage 1, without geometric metadata) achieved only $R^2 = 0.16$, confirming that the hybrid architecture—combining YOLOv8-Pose geometric measurements with LLM visual reasoning—is essential for reliable gauge reading extraction. These results indicate the system performs reliably under typical field conditions while highlighting the importance of the user validation workflow for challenging image quality scenarios.

Contributions

This report presents several contributions to the field of citizen science-based hydrological monitoring. First, we introduce the first WhatsApp-native platform for river discharge measurement, eliminating the need for dedicated mobile application installation and significantly lowering barriers to participation in regions with limited digital infrastructure. Second, we develop and evaluate a hybrid AI architecture that combines YOLOv8 computer vision models for geometric measurement with Google Gemini large language model capabilities for numerical interpretation—an approach that substantially outperforms either technique in isolation. Third, we implement a



two-stage validation workflow that simultaneously ensures data quality through citizen confirmation while generating continuous ground-truth training data, enabling ongoing model improvement without dedicated annotation campaigns. Fourth, we demonstrate integration with transboundary governance structures through station-level permissions aligned with LIMCOM institutional arrangements, providing a template for coordinated citizen science deployment across international river basins. Together, these contributions address a fundamental challenge in water resource management: enabling advanced digital decision support in precisely those basins where traditional monitoring infrastructure is weakest.

Related Work

Automated river gauge reading has attracted growing research attention as an alternative to costly sensor-based monitoring infrastructure. Liu and Huang (2024) evaluated deep learning approaches for water level measurement, achieving mean absolute errors of 4–6 cm using convolutional neural networks trained on fixed-camera imagery. However, their approach required consistent camera positioning and lighting conditions rarely achievable in citizen science contexts where image quality varies substantially.

Citizen science platforms for water resource monitoring have proliferated in recent years, though few integrate artificial intelligence for data extraction. The CrowdWater project (Seibert *et al* 2019) enables community members to contribute stream level observations through a dedicated mobile application, using virtual staff gauges that require users to estimate water levels visually rather than photographing physical gauge plates. FreshWater Watch (Thornhill *et al* 2016) engages citizens in water quality monitoring across multiple countries, demonstrating the viability of transboundary citizen science networks, though without the AI-assisted measurement capabilities developed here. The MiniSASS platform, which forms part of the broader project context for this work, successfully deploys AI vision recognition for macroinvertebrate identification but addresses water quality rather than discharge measurement.

Large language models with multimodal capabilities represent an emerging approach to visual measurement tasks. Recent work has demonstrated LLM effectiveness for interpreting complex visual scenes (Yang *et al.*, 2023), though application to hydrological instrumentation remains largely unexplored. The hybrid architecture presented in this report—combining geometric precision from purpose-trained computer vision models with contextual reasoning from general-purpose LLMs—addresses limitations inherent to either approach in isolation and, to our knowledge, represents the first application of such an architecture to citizen science-based river monitoring.

Protocol Design Framework

Conceptual Framework

The discharge measurement protocol is grounded in participatory monitoring approaches that recognise community members as capable and valued contributors to scientific data collection (Conrad and Hilchey, 2011). Unlike traditional hydrological monitoring that positions communities as passive beneficiaries of technical outputs, this framework actively engages citizens in the observation-validation-action cycle, creating what Buytaert *et al.* (2014) describe as "democratised hydrology."

The protocol architecture reflects a deliberate separation of concerns: citizens contribute what they do best—regular, localised observations with contextual awareness—while AI systems handle the technical complexity of image interpretation and numerical extraction. This division acknowledges that effective citizen science requires minimising cognitive load on participants while maximising the scientific value of their contributions (Bonney *et al.* 2009).

The workflow is structured around five sequential stages connecting citizen observers with the Digital Twin database (Figure 4). Each stage addresses specific technical requirements while maintaining simplicity from the user perspective:

- Stage 1 — Image Capture:

The citizen photographs the gauge plate via WhatsApp, with the platform automatically capturing essential metadata including GPS coordinates, timestamp, and device information. This approach reduces the need for manual data entry, reducing both user effort and transcription errors. If the image is unclear or metadata cannot be captured (e.g., location services disabled), the bot prompts the user to retake the photo for validation and future reprocessing. The use of WhatsApp as the submission channel ensures compatibility with low-end smartphones and leverages existing user familiarity with the platform's camera interface.

- Stage 2 — Object Detection:

A YOLOv8 model identifies the gauge plate boundaries within the submitted image and locates the precise point where the water surface intersects the gauge scale. The model draws a visual annotation (red line) at the detected water level, creating an interpretable output that users can verify. This stage transforms an unconstrained field photograph into a spatially-referenced measurement zone.

- Stage 3 — Reading Extraction:

The cropped gauge region is processed through a dual-AI pipeline. First, a secondary YOLOv8 model detects individual scale markers to establish pixel-to-centimetre calibration. Second, Google Gemini's multimodal large language model analyses the image to identify visible numbers and their sequence. A calculation algorithm combines geometric measurements with LLM-extracted values to compute the precise water level reading in centimetres.

- Stage 4 — User Validation:

A two-stage confirmation workflow ensures data quality before operational ingestion. In the first validation, users confirm whether the detected water line position is correct; if incorrect, they provide a manual reading. In the second validation, users confirm the AI-extracted numerical reading. This approach generates ground-truth data for continuous model improvement while flagging erroneous readings before they enter the database. The validation design reflects the principle that AI should augment rather than replace human judgement in scientific data collection.

- Stage 5 — Discharge Calculation:

The validated water level reading is converted to volumetric discharge (m^3/s) using station-specific rating curves accessed through the FlowTracker API. Rating curves represent the empirically-derived relationship between water stage and discharge for each monitoring station, accounting for local channel geometry and hydraulic characteristics. The final discharge value is stored in the Digital Twin database and made available for basin-wide analysis and visualisation. A key constraint is that discharge accuracy is ultimately limited by how recently each station's rating curve has been validated, particularly after floods or channel changes.

2.2 Design Principles

The protocol design adheres to several key principles informed by citizen science best practices (Wiggins and Crowston 2011), human-computer interaction research for development contexts (Dell and Kumar 2016), and the project's commitment to digital social innovation:

- Accessibility

No dedicated application installation is required; WhatsApp integration leverages existing digital literacy and device capabilities. This decision directly addresses the "digital divide" challenge identified in the project's design phase, recognising that potential citizen scientists may have limited smartphone storage, unreliable data connections, or unfamiliarity with app installation procedures. The protocol functions on basic smartphones with cameras, requiring only an active WhatsApp account and intermittent network connectivity for data submission.

- Simplicity

From the user perspective, participation requires a single action: photographing the gauge plate. All technical processing—object detection, reading extraction, unit conversion, and database integration—occurs transparently in the backend. This design follows the principle of progressive disclosure: users see only the information they need at each step, with complexity hidden unless explicitly requested. The cognitive load on participants is minimised through a guided, photo-first workflow that does not require specialized hydrological training. Participants are asked to perform a simple confirmation step when possible, but submissions are retained as draft records and are only accepted into the validated dataset after review by designated moderators. This ensures data quality while keeping the participation barrier low for occasional or non-specialist users..

- Transparency

Users receive visual feedback at each processing stage, including annotated images showing the detected water line and extracted readings. This transparency serves multiple purposes: it builds trust by demonstrating how AI systems interpret their photographs; it enables informed validation by showing exactly what is being confirmed; and it supports learning by helping users understand what constitutes a good gauge photograph. When AI interpretations are incorrect, users can identify and correct errors, maintaining agency over their contributions.

- Data Quality

Multiple validation checkpoints ensure data quality before observations enter operational systems. The two-stage confirmation workflow captures both geometric accuracy (water line position) and numerical accuracy (extracted reading), addressing the distinct failure modes of each AI component. User corrections are stored alongside AI predictions, creating paired datasets for model retraining. Additionally, metadata validation (timestamp format, coordinate bounds, file type) occurs at submission to reject malformed inputs early in the pipeline.

- Scalability

The architecture supports expansion to additional monitoring stations, river basins, and countries without fundamental redesign. Station-specific parameters (rating curves, gauge configurations) are retrieved dynamically from the FlowTracker API rather than hardcoded, enabling new stations to be onboarded through configuration rather than code changes. The WhatsApp-based interface requires no localised application deployment, and the containerised backend infrastructure can scale horizontally to accommodate increased submission volumes.

- Interoperability

Data collected through the protocol integrates with existing monitoring infrastructure through standardised APIs. The Digital Twin ingests citizen-contributed observations alongside official gauging station data, enabling comparison and gap-filling. All data follows FAIR principles (Findability, Accessibility, Interoperability, Reusability), supporting the project's commitment to open data and cross-border information sharing across LIMCOM member states.

- Digital Inclusiveness Considerations

The protocol design incorporates digital inclusiveness principles aligned with the CGIAR's Multidimensional Digital Inclusiveness Index (MDII) framework. Key considerations include:

- Language Accessibility:

While the current implementation operates in English, the architecture supports multilingual deployment. Text prompts, validation messages, and feedback can be localised without modifying the core processing pipeline. Future development will prioritise translation into languages prevalent in the Limpopo Basin, including Portuguese, Zulu, and Venda.

- **Connectivity Resilience:**

The WhatsApp-based submission pathway tolerates intermittent connectivity, as messages queue locally until network access is available. Image compression occurs client-side, reducing data transfer requirements. The validation workflow is designed to complete in a single conversational session, minimising the risk of abandoned submissions due to connection drops.

- **Device Inclusivity:**

The protocol functions on entry-level smartphones with basic camera capabilities. No specialised sensors, external hardware, or high-resolution imaging is required. Processing occurs server-side, placing computational demands on cloud infrastructure rather than user devices.

- **Skill Accessibility:**

The single-action submission model (photograph and send) requires no specialised technical skills beyond basic WhatsApp usage. Visual feedback and conversational prompts guide users through the validation workflow using familiar interaction patterns. Training materials developed for the broader citizen science programme support onboarding, but the protocol is designed to be usable without formal instruction.

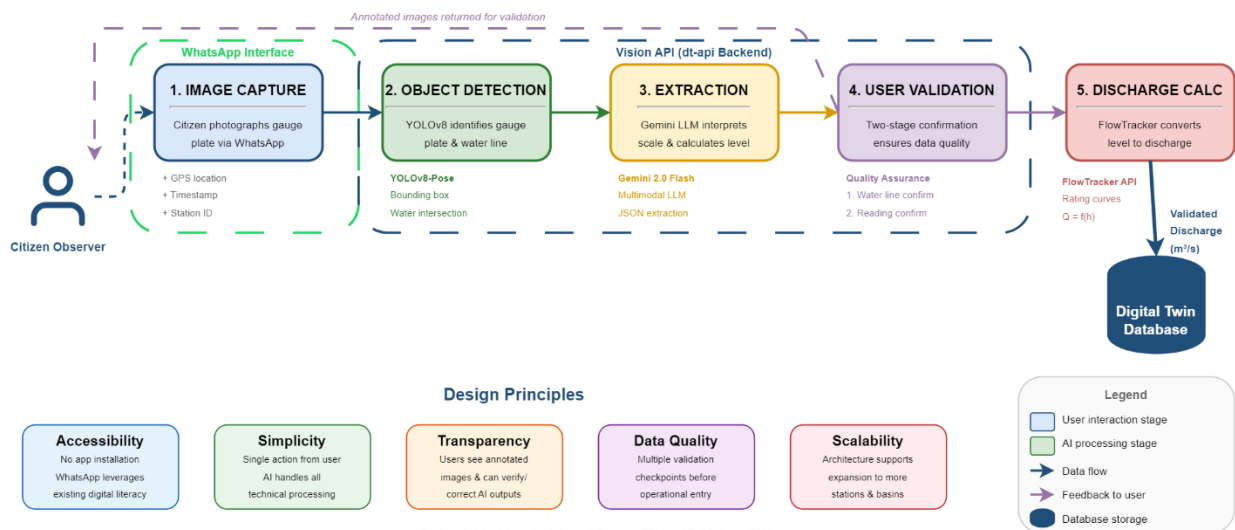


Figure 4 Discharge Measurement Protocol - Conceptual Framework [Source: IWMI]

System Architecture

Architectural Design Philosophy

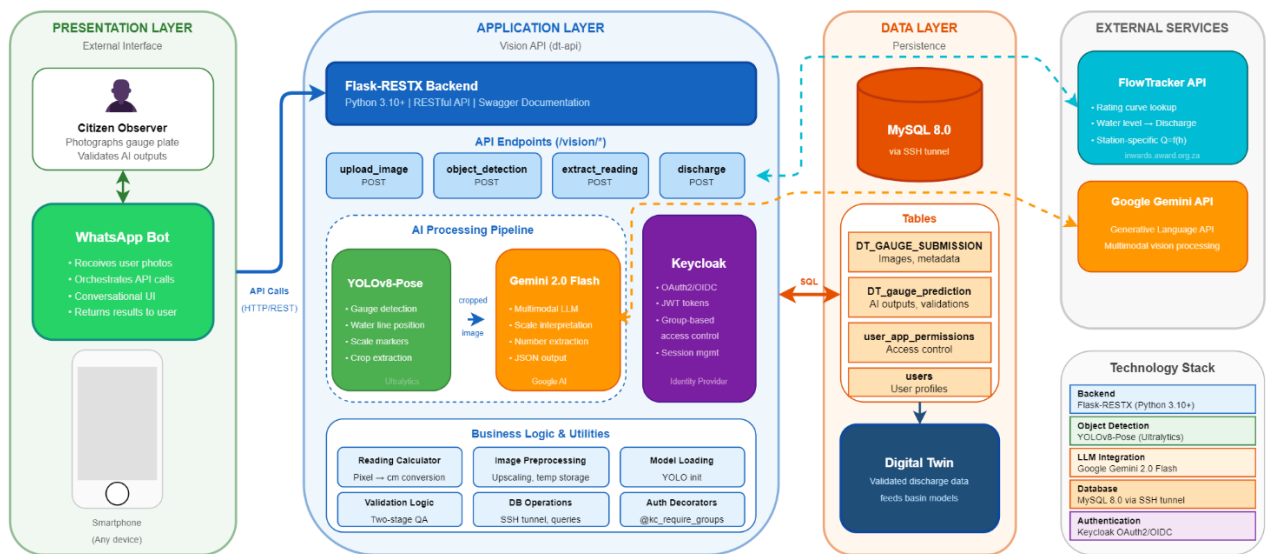
The system architecture follows a layered design pattern that separates concerns across presentation, application, and data tiers. This approach provides several advantages for citizen science deployments in resource-constrained environments: independent scaling of components based on demand, isolation of failures to prevent cascade effects, and flexibility to substitute technologies as requirements evolve (Fowler 2002).

The architecture prioritizes three key qualities:

- 1. Accessibility over Complexity:** The system interfaces with users through WhatsApp rather than a custom mobile application, eliminating installation barriers and leveraging existing platform capabilities for camera access, GPS capture, and message queuing.
- 2. Resilience over Performance:** Components are designed to tolerate intermittent connectivity, partial failures, and variable response times typical of deployments across multiple countries with heterogeneous network infrastructure.
- 3. Interoperability over Independence:** The system integrates with existing Digital Twin infrastructure, authentication services, and external APIs rather than reimplementing functionality, reducing development effort and ensuring consistency across the broader platform.

High-Level Architecture

The system architecture comprises three interconnected layers that collectively process citizen-submitted gauge photographs from image capture through to discharge calculation (Figure 5):



LIMCOM Digital Twin – Activity 2.4: Vision API System Architecture

Figure 5 System Architecture - High Level Overview [Source: IWMI]

Presentation Layer — WhatsApp Bot Interface

The WhatsApp bot serves as the primary user interface, providing a conversational interaction model familiar to target users across the Limpopo Basin. Built on the WhatsApp Business API, the bot orchestrates the complete submission workflow through a sequence of messages, prompts, and responses.

Key responsibilities include:

- Receiving gauge plate photographs from citizen scientists

- Extracting and forwarding image metadata (timestamp, GPS coordinates, device information)
- Presenting AI-annotated images for user validation
- Collecting confirmation responses and manual corrections
- Delivering final discharge results to users
- Managing conversation state across multi-turn interactions

The bot implements a stateful conversation model where each submission progresses through defined stages (upload → detection → extraction → validation → discharge). Session state is maintained externally, enabling the bot to resume interrupted conversations and handle users who submit multiple images concurrently.

Application Layer — Vision API (dt-api)

The Vision API provides the backend intelligence for the discharge measurement protocol. Implemented as a Flask-RESTX application, the API exposes RESTful endpoints that encapsulate AI processing, business logic, and external service integration.

The API is organised into functional modules:

Image Processing Module: Handles image upload, validation, storage, and retrieval. Validates file types (JPEG/PNG), extracts EXIF metadata where available, and manages temporary file storage during processing.

Object Detection Module: Executes YOLOv8 models for gauge plate localization and water line detection. Manages model loading, inference execution, result post-processing, and annotated image generation.

Reading Extraction Module: Orchestrates the dual-AI pipeline combining YOLOv8 scale detection with Gemini LLM visual analysis. Implements the reading calculation algorithm and handles edge cases (missing scale gaps, negative readings, scale variations).

Validation Module: Records user confirmations and corrections, updating prediction records with ground-truth data. Supports the two-stage validation workflow with separate endpoints for water line and reading confirmation.

Discharge Calculation Module: Interfaces with the FlowTracker API to convert validated water level readings to discharge values using station-specific rating curves.

Authentication Module: Integrates with Keycloak for OAuth2/OIDC-based identity management, providing user registration, login, token refresh, and session management capabilities.

The API follows RESTful design conventions with consistent endpoint naming, HTTP method semantics, and JSON response formats. Error handling provides meaningful feedback for debugging while avoiding exposure of sensitive implementation details.

Data Layer — MySQL Database

The MySQL database provides persistent storage for all submission data, AI predictions, user validations, and system metadata. The schema is designed to support both operational workflows and analytical queries for model performance assessment.

Primary tables include:

DT_GAUGE_SUBMISSION: Stores initial image submissions including binary image data (BLOB), user identifiers, timestamps, GPS coordinates, station assignments, and processing status flags. Each submission receives a unique identifier used for tracking through subsequent processing stages.

DT_gauge_prediction: Stores AI predictions and user validations linked to submissions via foreign key. Captures AI-generated readings, user corrections (both stages), scale gap measurements, confidence indicators, and prediction timestamps. The dual-column structure for validations (user_validation/user_corrected_reading and user_validation_2/user_corrected_reading_2) supports the two-stage confirmation workflow.

Database connectivity is secured through SSH tunnelling, providing encrypted transport without requiring direct exposure of the database server to public networks. Connection pooling manages concurrent access from multiple API instances.

Data Flow Architecture

The complete data flow for a gauge reading submission proceeds through six phases:

Phase 1 — Submission (User → WhatsApp → API → Database)

1. User captures gauge photograph via WhatsApp camera
2. WhatsApp bot receives image and extracts available metadata
3. Bot calls POST /vision/upload_image with image file and metadata
4. API validates inputs and stores submission in DT_GAUGE_SUBMISSION
5. API returns submission_id to bot for subsequent operations

Phase 2 — Object Detection (API → AI Models → Database)

1. Bot calls POST /vision/object_detection with submission_id
2. API retrieves image from database
3. YOLOv8 processes image to detect gauge plate and water line
4. API generates annotated image with water line visualisation
5. Results stored; annotated image returned to bot

Phase 3 — First Validation (User → WhatsApp → API → Database)

1. Bot displays annotated image to user
2. User confirms water line position or provides correction
3. Bot calls POST /vision/confirm_reading with validation response
4. API records validation in DT_gauge_prediction

Phase 4 — Reading Extraction (API → AI Models → Database)

1. Bot calls POST /vision/extract_reading with submission_id
2. API retrieves cropped gauge image from detection stage
3. YOLOv8 "base" model detects scale markers for calibration
4. Gemini LLM analyses image to extract visible numbers
5. Algorithm calculates water level reading from combined inputs
6. Prediction stored in DT_gauge_prediction; reading returned to bot

Phase 5 — Second Validation (User → WhatsApp → API → Database)

1. Bot displays AI-extracted reading to user
2. User confirms reading accuracy or provides correction
3. Bot calls POST /vision/confirm_ai_reading with validation response
4. API records second-stage validation

Phase 6 — Discharge Calculation (API → External API → User)

1. Bot calls POST /vision/discharge with submission_id

2. API determines final reading (user correction takes priority over AI)
3. API calls FlowTracker API with station ID and water level (converted to metres)
4. FlowTracker returns discharge value from rating curve lookup
5. API returns discharge to bot; bot displays result to user

Technology Stack

The technology stack was selected to balance performance, maintainability, and compatibility with the broader Digital Twin infrastructure. Selection criteria included: maturity and community support, suitability for the specific technical requirements, alignment with team expertise, and licensing compatibility with open-source project goals.

Table 1 Core system components, selected technologies, and rationale for the Vision API and WhatsApp-based discharge measurement workflow.

| Component | Technology | Selection Rationale |
|-------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend Framework | Flask-RESTX (Python 3.10+) | Lightweight framework with built-in API documentation (Swagger/OpenAPI), strong ecosystem for scientific computing and ML integration, consistent with broader Digital Twin codebase |
| Object Detection | YOLOv8-Pose (Ultralytics) | State-of-the-art real-time object detection with pose estimation capabilities, permissive licensing (AGPL-3.0), active maintenance, and extensive documentation |
| LLM Integration | Google Gemini 2.0 Flash | Multimodal capabilities for combined image-text analysis, competitive performance on visual reasoning tasks, cost-effective API pricing for high-volume inference |
| Database | MySQL 8.0 | Mature relational database with strong BLOB handling for image storage, compatibility with existing Digital Twin infrastructure, robust replication and backup capabilities |
| Authentication | Keycloak OAuth2/OIDC | Open-source identity management with standards-based protocols, group-based access control, federation capabilities for future multi-organisation deployment |
| External APIs | FlowTracker | AWARD-maintained rating curve database covering Limpopo Basin gauging stations, RESTful interface, established operational track record |
| Containerisation | Docker | Consistent deployment across development, staging, and production environments, isolation of dependencies, compatibility with cloud orchestration platforms |
| Image Processing | OpenCV, Pillow | Industry-standard libraries for image manipulation, format conversion, and preprocessing |

Integration Architecture

The Vision API integrates with several external systems to deliver end-to-end functionality:

FlowTracker API Integration

The FlowTracker system, maintained by AWARD, provides access to rating curves for gauging stations across the Limpopo Basin and broader Southern African region. The integration operates as follows:

- Endpoint: https://inwards.award.org.za/api/flowtracker/fetch_rating

- Parameters: Station identifier, water level (meters)
- Response: Discharge value (m³/s) interpolated from rating curve
- Error Handling: Returns a clear error message and continues the workflow without crashing when station not found or rating curve unavailable

Rating curves represent empirically-derived stage-discharge relationships specific to each monitoring station. The FlowTracker database is maintained through periodic field campaigns that update curves as channel geometry evolves.

Keycloak Authentication Integration

User authentication leverages Keycloak, an open-source identity and access management solution. The integration supports:

- User Registration: New users created in Keycloak realm with automatic group assignment
- Authentication: OAuth2 password grant flow returning JWT access and refresh tokens
- Authorisation: Group-based access control via `@kc_require_groups()` decorator
- Token Management: Refresh token rotation for session continuity

Two user groups govern API access:

- dt-vision-users: Standard access to vision processing endpoints
- vision-app-admin: Administrative access for system management

Meta WhatsApp Cloud API Integration

The WhatsApp bot leverages the Meta WhatsApp Cloud API for message handling and media management, enabling the conversational interface that connects citizen scientists with the Vision API backend.

API Configuration

The integration requires the following setup steps:

1. Register a developer account on Meta for Developers (<https://developers.facebook.com>)
2. Create a new application and add the WhatsApp product
3. Configure a webhook URL to receive message and media events
4. Verify the webhook endpoint using a verification token
5. Obtain a permanent access token for production deployment

Access Token Management

- Permanent access tokens are required for production deployment, replacing the temporary tokens issued during development
- Tokens are securely stored in AWS Secrets Manager rather than environment variables
- Token rotation is performed periodically to prevent unauthorized access

Webhook Configuration

The Flask server provides the API service and exposes a webhook endpoint that Meta's WhatsApp servers call when users send messages. A webhook is a URL that WhatsApp calls automatically to deliver incoming messages to our service, allowing it to process the request and return a response. The endpoint handles:

- Webhook verification challenges (GET requests with `hub.verify_token`)

- Incoming message events (POST requests with message payloads)
- Media download authorization using the permanent access token

Message and Media Flow

- Endpoint: https://graph.facebook.com/v18.0/{phone_number_id}/messages
- Media retrieval: https://graph.facebook.com/v18.0/{media_id}
- Authentication: Bearer token in Authorization header
- Rate limits: Subject to Meta's standard API rate limiting policies

Digital Twin Integration

Validated discharge measurements are made available to the broader LIMCOM Digital Twin platform through shared database access and API federation. Citizen-contributed observations complement official gauging station data, enabling:

- Gap-filling for stations with intermittent official monitoring
- Cross-validation of automated sensor readings
- Enhanced spatial coverage in under-monitored tributaries
- Near-real-time situational awareness during flood events

Deployment Architecture

The Vision API is deployed as a containerised application within the Digital Twin infrastructure:

Container Configuration:

- Base image: Python 3.10 slim
- Dependencies managed via requirements.txt
- Environment variables for configuration (database credentials, API keys, Keycloak settings)
- Health check endpoint for orchestration monitoring

Resource Management:

- CPU-bound inference (YOLOv8) with limited thread count for stability
- Memory allocation sized for concurrent image processing
- Temporary file cleanup after processing completion

Network Configuration:

- HTTPS termination at load balancer
- SSH tunnel for database connectivity
- Outbound access to Gemini API and FlowTracker endpoints

Operational Considerations:

- Logging to centralised aggregation service
- Metrics collection for performance monitoring

- Automated restart on failure detection

AI Processing Pipeline

Overview

The AI processing pipeline represents the core technical innovation of the discharge measurement protocol, combining computer vision and large language model capabilities to automate gauge reading extraction from citizen-submitted photographs. The pipeline addresses a fundamental challenge in imagery-based hydrological monitoring: translating variable-quality field photographs into precise numerical water level readings suitable for operational use.

The framework integrates vision-based waterline detection, YOLOv8-Pose scale extraction, and multimodal large language models (Gemini 2.0 Flash) for automated river gauge plate reading. This hybrid architecture leverages the complementary strengths of geometric precision from object detection models for spatial measurements, combined with contextual reasoning from multimodal LLMs for numerical interpretation (Figure 7).

The pipeline processes each submission through three sequential stages: waterline detection, scale gap ratio estimation, and reading extraction. The complete workflow executes in approximately 3–4 seconds under typical conditions.

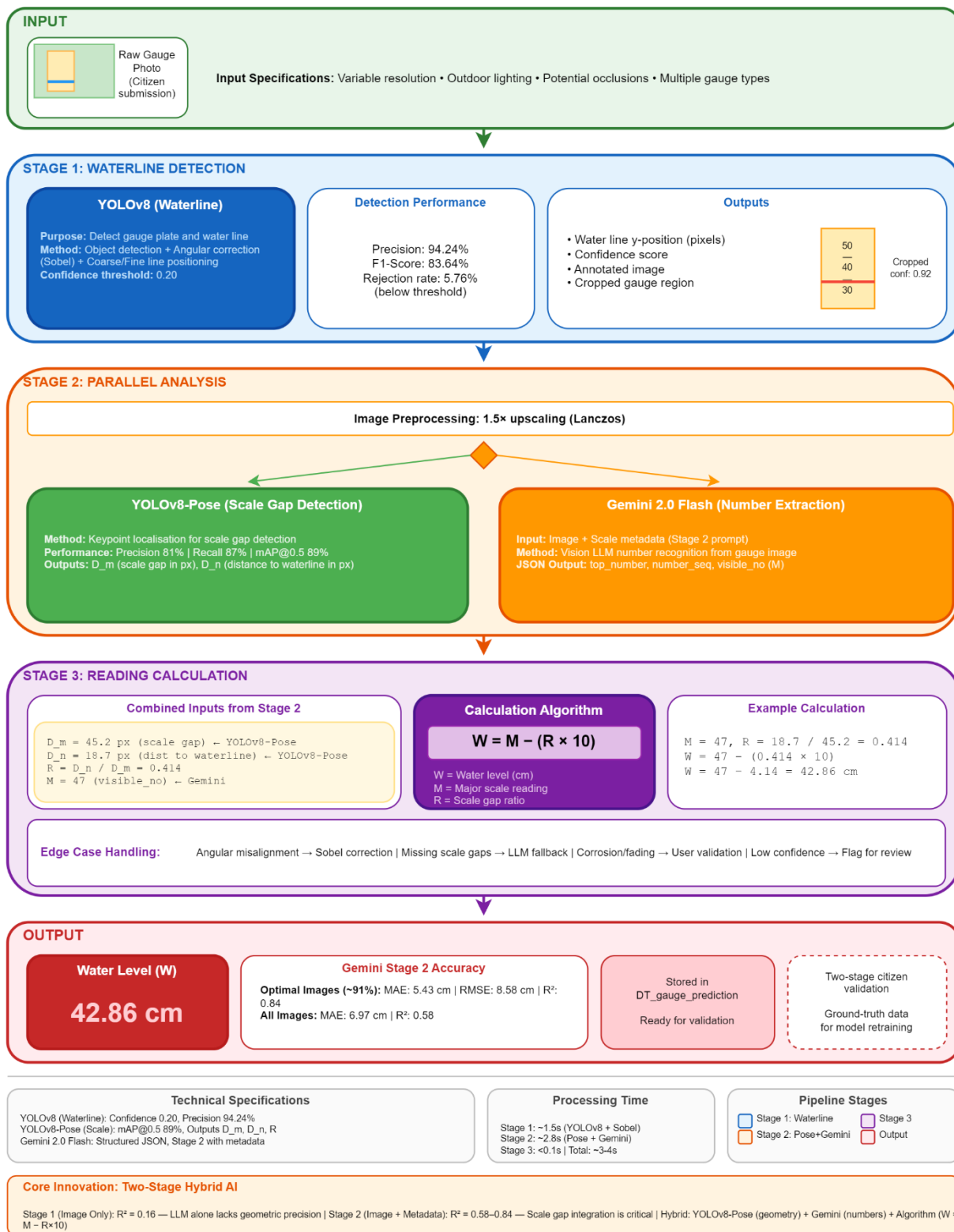


Figure 6 AI Processing Pipeline – Technical Architecture [Source: IWMI]

Stage 1: Waterline Detection (YOLOv8)

The first stage localises the gauge plate within the raw citizen photograph and identifies the water line position. This stage transforms an unconstrained field image—potentially containing vegetation, infrastructure, reflections, and other visual noise—into a focused measurement with precise water line positioning.

Detection Methodology

Waterline detection is performed through a multi-step process combining object detection with image processing techniques:

- **Object Detection:** YOLOv8 identifies the gauge plate region within the submitted image
- **Angular Correction:** Gradient information is extracted using the Sobel operator to correct for camera angle misalignment
- **Coarse Line Positioning:** Edge detection using the Canny operator identifies the approximate waterline through horizontal edge intensity profiling
- **Fine Line Positioning:** Vertical gradient (Sobel Y) computed within a narrow region (± 5 rows) around the coarse line identifies the precise water line position

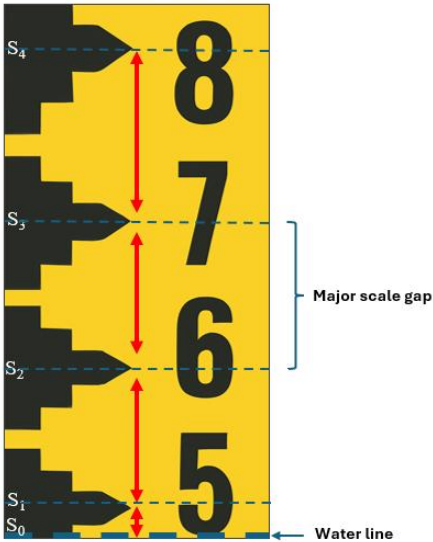


Figure 7 Diagram of template gauge plate used in the field [Source: IWMI]

Model Specifications:

| Parameter | Value | Rationale |
|----------------------|--------------------|--------------------------------------------------------------------------------------------------------|
| Architecture | YOLOv8 | One-stage detection with CSPDarknet53 backbone |
| Confidence Threshold | 0.20 | Balances detection sensitivity with false positive rejection; 5.76% of images rejected below threshold |
| Validation Zone | ± 5 pixel rows | True positive defined as detection within this zone of annotated waterline |
| Input Resolution | 640 × 640 px | Standardised resolution for consistent inference |

Detection Performance

Quantitative assessment on the evaluation dataset demonstrated robust waterline detection:

| Metric | Value |
|---------------------|--------------------------|
| Precision | 94.24% |
| F1-Score | 83.64% |
| False Positive Rate | 0% (at confidence >0.20) |

The confidence distribution showed the majority of predictions concentrated in the high-confidence range (0.85–1.0). Lower confidence values (0.30–0.60) were observed in challenging cases caused by scale invisibility due to corrosion, interference from surrounding objects (grass, debris), and image blurring.

Detection Outputs

The waterline detection stage produces outputs that feed downstream processing:

1. **Water Line Y-Position:** Vertical pixel coordinate where water intersects the gauge scale
2. **Confidence Score:** Model confidence (0.0–1.0) displayed to users on annotated images
3. **Annotated Image:** Original photograph with red horizontal line at detected water level and confidence percentage overlay
4. **Cropped Gauge Region:** Extracted sub-image containing the gauge plate for Stage 2 processing

Stage 2: Scale Gap Ratio Estimation (YOLOv8-Pose)

The second stage employs YOLOv8-Pose to detect scale markers on the gauge plate through keypoint localisation, enabling pixel-to-centimetre calibration essential for accurate reading calculation.

Model Architecture

The YOLOv8-Pose architecture integrates object detection and pose estimation within a single framework, enabling simultaneous prediction of bounding boxes, object classes, confidence scores, and keypoint coordinates. The architecture comprises three major components:

- **Backbone:** Feature extractor with convolutional and C2f units, concluding with Spatial Pyramid Pooling Fast (SPPF) module
- **Neck:** Fuses feature representations from different scales using concatenation and upsampling layers
- **Head:** Pose units that localise both objects and structural features, predicting bounding box coordinates, class probabilities, and keypoint positions

Model Specifications:

| Parameter | Value | Rationale |
|----------------------|----------------------------|------------------------------------------------------------------|
| Architecture | YOLOv8-Pose | Enables combined detection + keypoint localisation |
| Confidence Threshold | 0.20 | Balances detection sensitivity with false positive rejection |
| NMS Method | Custom IoU-based filtering | Handles overlapping detections from multi-scale gauge markings |
| Execution Device | CPU | Ensures deployment stability; GPU optional for higher throughput |
| Thread Limiting | Enabled | Prevents resource contention in containerised environment |

Scale Gap Detection Performance

| Metric | Value |
|--------------|-------|
| Precision | 81% |
| Recall | 87% |
| mAP@0.5 | 89% |
| mAP@0.5-0.95 | 80% |

Geometric Calculations

The scale gap detection outputs enable precise geometric calibration:

Major Scale Gap (D_m): Computed as the median of differences between consecutive major scale keypoints:

$$D_m = \text{Median}(s_2 - s_1, s_3 - s_2, s_4 - s_3, \dots, s_n - s_{n-1})$$

Distance to Waterline (D_n): The pixel distance from the lowest detected major scale marker to the waterline intersection:

$$D_n = s_1 - s_0$$

Scale Gap Ratio (R): The ratio enabling sub-interval precision:

$$R = D_n / D_m$$

These outputs— D_m , D_n , and total pixel height (H_y)—are passed to the LLM for reading extraction.

Stage 3: Reading Extraction (Gemini 2.0 Flash LLM)

The third stage employs the Gemini 2.0 Flash multimodal large language model to extract numerical values from the cropped gauge image, combining visual perception with contextual reasoning.

Two-Stage LLM Approach

The reading extraction was developed and evaluated using a two-stage approach:

- **LLM Stage 1 (Image Only):** The model receives only the pre-processed, waterline-detected image, requiring it to infer both the gauge reading and scale spacing directly from visual information.
- **LLM Stage 2 (Image + Scale Metadata):** The model receives the image plus scale gap ratio metadata (D_m , D_n , R), enabling accurate conversion of pixel distances to real-world measurements.

The operational deployment uses Stage 2, as incorporating scale gap information substantially improves predictive accuracy.

Prompt Engineering

A structured prompt guides the AI model in extracting water-level readings:

- Identify the topmost visible digit on the gauge scale

- Determine the full number sequence from top to bottom
- Detect any partially visible digit at the lowest edge (water line position)

LLM Outputs (Structured JSON):

```

json
{
  "top_number": 50,
  "number_seq": [50, 40, 30, 20],
  "visible_no": 47
}

```

- **top_number:** Highest numerical value visible on the gauge scale
- **number_seq:** Sequence of visible numbers confirming scale direction and interval
- **visible_no:** Number corresponding to the water line position (major scale reading M)

Reading Calculation Algorithm

The final water level reading combines the LLM-extracted major scale value with the geometric scale gap ratio:

$$W = M - (R \times 10)$$

Where:

- W = Water level reading (cm)
- M = Major scale reading at waterline (from LLM: visible_no, representing the scale mark just above water)
- R = Scale gap ratio (D_n / D_m from YOLOv8-Pose)

Calculation Example:

Given:

- M = 47 (major scale mark visible just above water line)
- D_m = 45.2 pixels (average major scale gap)
- D_n = 18.7 pixels (distance from "47" marker to water line)
- R = 18.7 / 45.2 = 0.414

$$\begin{aligned}
 W &= 47 - (0.414 \times 10) \\
 W &= 47 - 4.14 \\
 W &= 42.86 \text{ cm}
 \end{aligned}$$

Note: Actual scale interpretation depends on gauge plate design; this example assumes 10 cm intervals between major markers.

Edge Case Handling

| Edge Case | Detection Method | Handling Strategy |
|----------------------|-------------------------------|-----------------------------------------------------------------------|
| Missing scale gaps | Fewer than 2 markers detected | Flag for user validation; LLM-only estimation with reduced confidence |
| Angular misalignment | Sobel gradient analysis | Automated rotation correction before processing |
| Corrosion/fading | Low confidence detection | User prompted to confirm or retake photograph |
| Partial occlusion | Incomplete number sequence | LLM interpolation based on visible pattern |
| Poor lighting | High variance in detections | Flag submission for manual review |

Performance Summary

Model Comparison (All Image Categories, After Outlier Removal)

| Model Configuration | Bias (cm) | MAE (cm) | RMSE (cm) | R ² |
|---------------------|-----------|----------|-----------|----------------|
| GPT-4o Stage 1 | 13.97 | 17.35 | 22.58 | 0.45 |
| GPT-4o Stage 2 | 4.34 | 9.99 | 17.49 | 0.49 |
| Gemini Stage 1 | 7.06 | 10.27 | 14.28 | 0.63 |
| Gemini Stage 2 | 1.98 | 6.97 | 13.68 | 0.58 |

Performance on Optimal Quality Images

When evaluated on images with clear scales, daylight conditions, no blur, no corrosion, and no visual obstacles:

| Model Configuration | Bias (cm) | MAE (cm) | RMSE (cm) | R ² |
|---------------------|-----------|----------|-----------|----------------|
| GPT-4o Stage 1 | 14.71 | 16.56 | 21.68 | 0.54 |
| GPT-4o Stage 2 | 4.89 | 9.33 | 15.99 | 0.56 |
| Gemini Stage 1 | 7.90 | 9.38 | 11.71 | 0.80 |
| Gemini Stage 2 | 3.04 | 5.43 | 8.58 | 0.84 |

Gemini Stage 2 achieved the best performance across all metrics, with the lowest errors and highest correlation with ground-truth measurements (R² = 0.84 for optimal images).

Image Quality Sensitivity

The results demonstrate strong sensitivity to image quality:

| Image Category | Proportion | Gemini Stage 2 Performance |
|---------------------|------------|---------------------------------|
| Optimal quality | ~91% | MAE = 5.43 cm, RMSE = 8.58 cm |
| Sub-optimal quality | ~9% | MAE = 11.70 cm, RMSE = 13.53 cm |
| All images | 100% | MAE = 6.97 cm, RMSE = 13.68 cm |

These findings confirm that the two-stage user validation workflow is essential: for challenging images, user corrections ensure data quality despite reduced AI accuracy.

Core Innovation: Hybrid AI Architecture

The pipeline's hybrid architecture addresses limitations of single-model approaches:

Why Not Pure Computer Vision?

Traditional object detection models excel at geometric tasks but struggle with the contextual interpretation required to read gauge scales—particularly when numbers are partially visible, non-standard fonts are used, or scale orientation varies.

Why Not Pure LLM Analysis?

Stage 1 results demonstrate this limitation: without scale gap metadata, LLMs achieved weak correlations (R² = 0.13–0.16), as they lacked the geometric precision required for accurate interpolation between scale markers.

The Hybrid Advantage:

By separating concerns—geometric measurement via YOLOv8-Pose, numerical interpretation via Gemini, integration via algorithm—the pipeline achieves substantially higher accuracy than either approach in isolation. Stage 2 results show the benefit of combining scale gap ratio metadata with LLM visual reasoning (R² improvement from 0.16 to 0.58 for Gemini).

This modular architecture also enables independent improvement of each component as models evolve, and the structured output format (JSON) ensures reliable downstream processing.

API Endpoint Reference

The Vision API exposes six core endpoints implementing the discharge measurement workflow (`upload_image`, `object_detection`, `extract_reading`, `confirm_reading`, `confirm_ai_reading`, `discharge`) plus authentication endpoints for user management. Full API documentation including request parameters, response formats, and error codes is provided in Appendix A.

Data Model

The database schema supports the complete citizen science workflow from user registration through submission, AI processing, validation, and discharge calculation. The schema (Figure 8) comprises ten interconnected tables organised into four functional groups: user management, submission handling, AI prediction, and hydrological reference data.

User Management Tables

app_users: Stores user profile information synchronised from Keycloak authentication, including kc_sub (Keycloak subject identifier), email, display_name, organization, and session timestamps.

kc_users: Maps Keycloak subjects to internal user identifiers for cross-referencing.

user_app_permissions: Controls application-level access including can_submit, can_moderate, can_app_admin flags, and plate_quota_month for submission rate limiting.

user_station_permissions: Manages station-specific permissions, allowing fine-grained control over which users can submit to or moderate specific monitoring stations.

Submission Tables (Two-Stage Workflow)

DT_GAUGE_SUBMISSION: Stores raw submissions as received from the WhatsApp bot, preserving the original image and metadata before any processing or moderation. Key fields include image_data (LONGBLOB), geographic coordinates, timestamps, mobile_number, user_id, and station reference.

plate_submissions: Stores processed and moderated submission records, linking raw submissions to corrected outputs. This table supports the quality assurance workflow with fields for stage_height_corrected, discharge_computed, status (ENUM for workflow state), review_notes, reviewer_kc_sub, and moderation timestamps. Foreign keys link to raw_submission_id and station_id.

AI Prediction Table

DT_gauge_prediction: Stores AI processing outputs from the Vision API pipeline and captures two-stage user validation results:

| Field Group | Fields | Description |
|---------------------|------------------------------------------------------------------------|---------------------------------|
| Image Reference | image_path, submission_id | Links to source submission |
| YOLOv8-Pose Outputs | scale_gap_pixels (D_m), distance_to_bottom (D_n), total_pixel_height_y | Geometric measurements |
| Gemini Outputs | top_reading, visible_no (M), number_seq | Extracted numerical values |
| Calculated Reading | expected_reading, corrected_reading | $W = M - (R \times 10)$ result |
| First Validation | user_validation, user_corrected_reading | Citizen confirmation/correction |
| Second Validation | user_validation_2, user_corrected_reading_2 | Follow-up confirmation |

Hydrological Reference Tables

stations: Monitoring station registry containing code, name, geographic coordinates, and is_active status flag.

rating_curves: Stage-discharge relationships for each station, storing curve_params (rating curve coefficients) used by the FlowTracker API to convert validated water level readings to discharge values.

Data Flow

The submission workflow progresses through the schema as follows:

1. **Submission Receipt:** WhatsApp bot stores raw image and metadata in DT_GAUGE_SUBMISSION
2. **AI Processing:** Vision API processes image; results stored in DT_gauge_prediction linked via submission_id
3. **User Validation:** Two-stage citizen confirmation updates user_validation and user_corrected_reading fields
4. **Moderation:** Approved submissions promoted to plate_submissions with corrected values and moderator notes
5. **Discharge Calculation:** Validated stage heights passed to FlowTracker API using rating_curves parameters

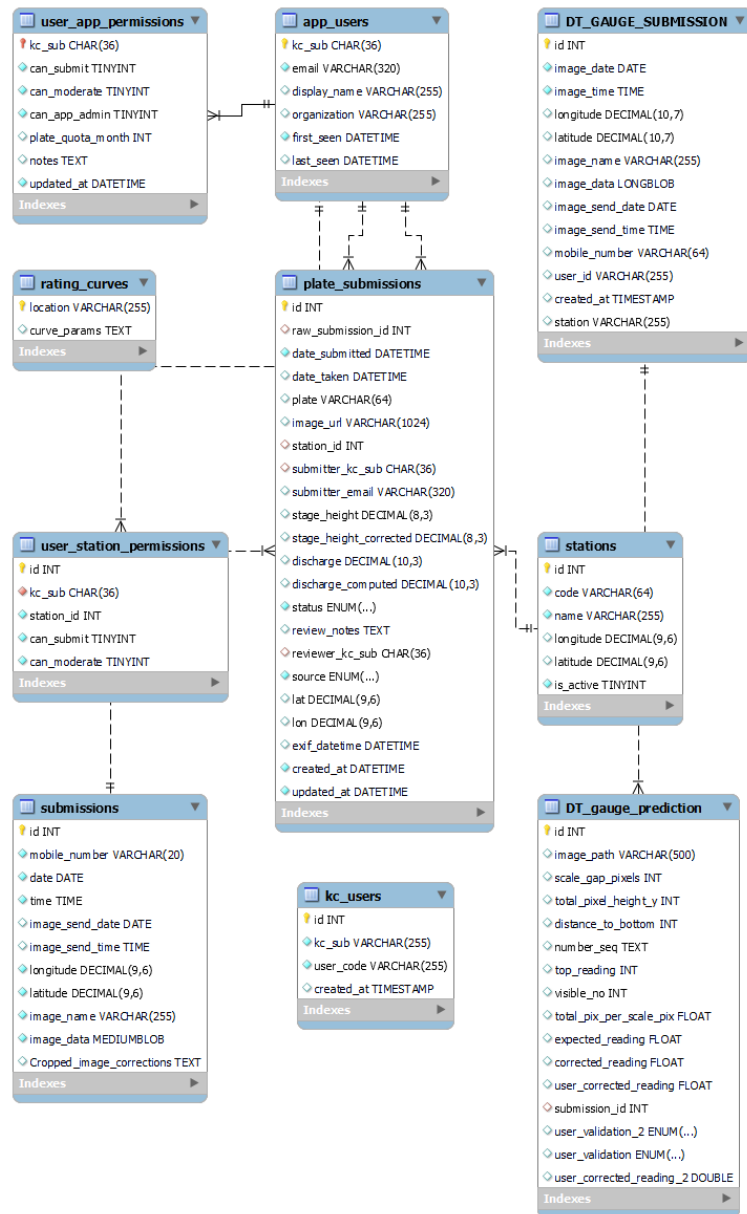


Figure 8 DT-Vision schema [Source: IWMI]

WhatsApp Bot Integration Workflow

Platform Architecture

The citizen science interface is implemented as a custom conversational bot service deployed on AWS ECS, integrated with the WhatsApp Business API (Cloud API) for message transport. This architecture separates the messaging channel from the application logic, enabling full control over conversation design, state management, and integration with the Vision API backend.

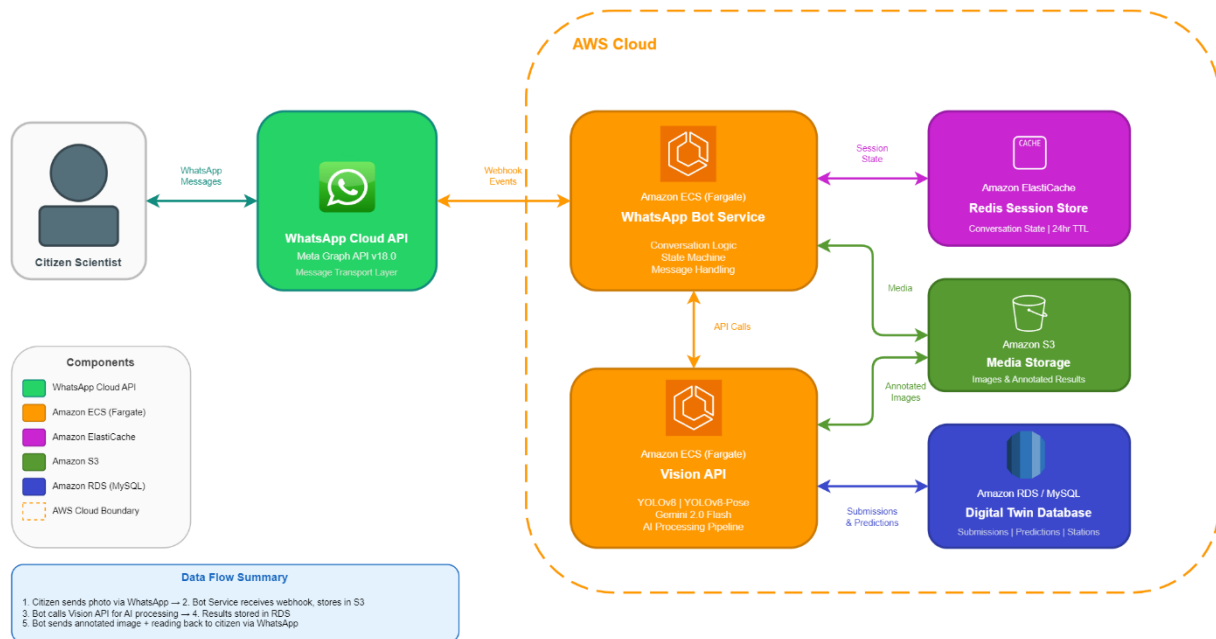


Figure 9 Custom Bot Deployment on AWS ECS with WhatsApp Business API Integration [Source: IWMI]

System Components

| Component | Deployment | Technology | Purpose |
|----------------------|-------------------|-----------------|---------------------------------------------------------|
| WhatsApp Bot Service | AWS ECS (Fargate) | Python/Flask | Conversation logic, state management, API orchestration |
| Session Store | AWS ElastiCache | Redis | Conversation state persistence across messages |
| Vision API | AWS ECS (Fargate) | Flask | AI processing pipeline |
| WhatsApp Cloud API | Meta Platform | Graph API v18.0 | Message transport (send/receive) |
| Media Storage | AWS S3 | — | Temporary image storage during processing |

Architectural Flow

The WhatsApp Cloud API serves purely as a message transport layer—all conversation intelligence, interactive flows, and integration logic resides in the custom bot service.

Bot Service Implementation

Webhook Handler

The bot service exposes a webhook endpoint that receives all incoming WhatsApp events. Each message triggers the conversation engine which determines the appropriate response based on current session state:

```
python
@app.route('/webhook', methods=['POST'])
def webhook():
    payload = request.get_json()
    for entry in payload.get('entry', []):
        for change in entry.get('changes', []):
            message = change['value'].get('messages', [{}])[0]
            sender = message.get('from')
            # Load or create session
            session = session_manager.get_session(sender)
            # Route to conversation handler
            response = conversation_engine.handle(message, session)
            # Send response via WhatsApp Cloud API
            whatsapp_client.send(sender, response)
    return 'OK', 200
```

Session State Management

Each user's conversation progress is tracked through a session object persisted in Redis. Sessions maintain state across the asynchronous message-based interaction pattern:

```
python
class UserSession:
    user_id: str
    phone_number: str
    state: ConversationState
    submission_id: Optional[int]
    station_id: Optional[int]
    location: Optional[Tuple[float, float]]
    image_date: Optional[date]
    ai_reading: Optional[float]
    created_at: datetime
    updated_at: datetime
    expires_at: datetime # 24-hour TTL
```

Conversation State Machine

The bot implements a finite state machine controlling the conversation flow. Each state defines valid user inputs and corresponding transitions:

| State | Trigger | Bot Action | Next State |
|-------|-------------|--------------------------------------------|----------------|
| IDLE | Any message | Send welcome + "Send a photo of the gauge" | AWAITING_IMAGE |

| | | | |
|----------------------------|------------------------|----------------------------------------|----------------------------|
| AWAITING_IMAGE | Image received | Store image, prompt for station | AWAITING_STATION |
| AWAITING_STATION | Station selected | Store station, prompt for location | AWAITING_LOCATION |
| AWAITING_LOCATION | Location shared | Store coordinates, prompt for date | AWAITING_DATE |
| AWAITING_DATE | Date confirmed/entered | Call Vision API, show annotated image | AWAITING_WATERLINE_CONFIRM |
| AWAITING_WATERLINE_CONFIRM | "Yes" button | Extract AI reading, display result | AWAITING_READING_CONFIRM |
| AWAITING_WATERLINE_CONFIRM | "No" button | Prompt to retake photo | AWAITING_IMAGE |
| AWAITING_READING_CONFIRM | "Yes" button | Calculate discharge, show final result | COMPLETE |
| AWAITING_READING_CONFIRM | "No" button | Prompt for manual reading | AWAITING_MANUAL_READING |
| AWAITING_MANUAL_READING | Number entered | Store correction, calculate discharge | COMPLETE |
| COMPLETE | Any message | Thank user, reset session | IDLE |

Interactive Message Design

The bot constructs rich interactive messages using WhatsApp's supported formats, creating an intuitive user experience that minimises typing and reduces input errors.

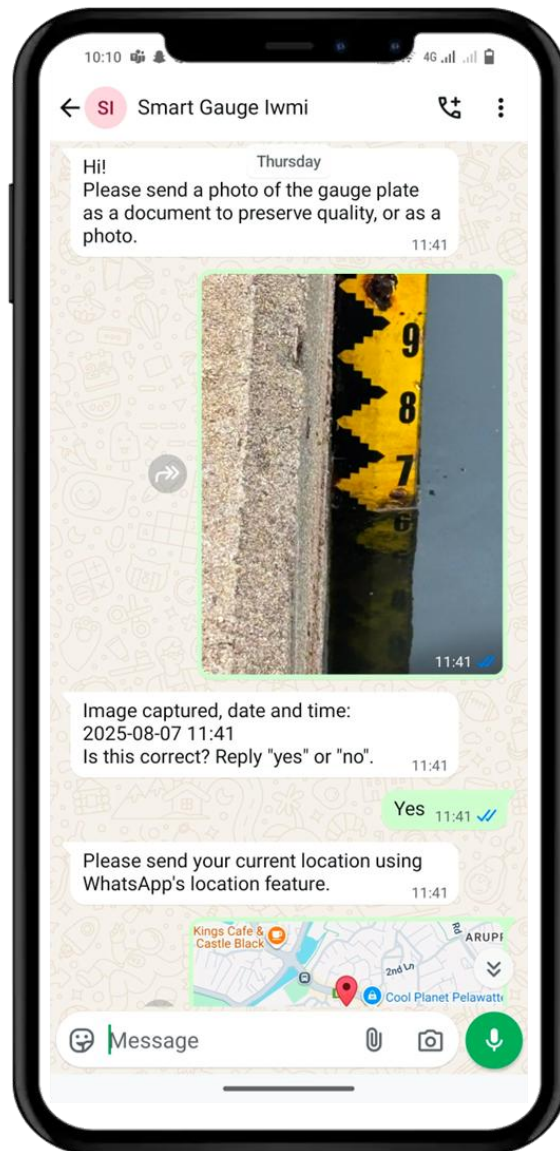


Figure 10 Screenshot of the Vision Bot in action initiated the process after a triggering "hi" [Source: IWMI]

Station Selection (Interactive List)

When prompting for monitoring station, the bot presents an interactive list populated from the stations table:

```

json
{
  "type": "interactive",
  "interactive": {
    "type": "list",
    "header": { "type": "text", "text": "Select Station" },
    "body": { "text": "Which monitoring station is this gauge located at?" },
    "action": {
      "button": "View Stations",
      "sections": [
        {
          "title": "Limpopo Basin Stations",
          "rows": [
            { "id": "stn_001", "title": "Beitbridge", "description": "Limpopo Main - Zimbabwe border" },
            { "id": "stn_002", "title": "Chokwe", "description": "Limpopo Main - Mozambique" },
            { "id": "stn_003", "title": "Pafuri", "description": "Limpopo-Luvuvhu confluence" }
          ]
        }
      ]
    }
  }
}

```

Location Request

The bot requests the user's location using WhatsApp's native location sharing, which provides accurate GPS coordinates:

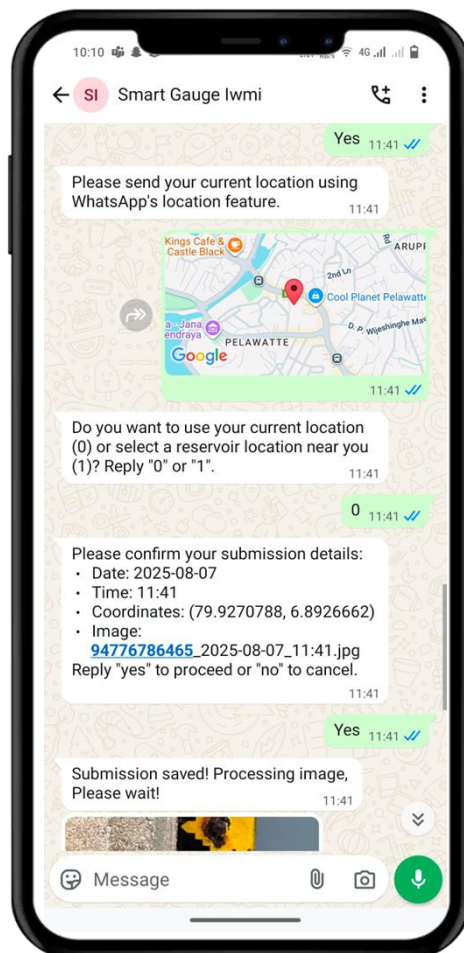


Figure 11 The vision API endpoints require a location to be submitted before any processing begins [Source: IWMI]

```
json
{
  "type": "interactive",
  "interactive": {
    "type": "location_request_message",
    "body": {
      "text": "Please share your current location so we can verify the gauge position."
    },
    "action": {"name": "send_location"}
  }
}
```

Date Confirmation (Quick Reply Buttons)

For date entry, the bot offers the current date as a default with an option to specify a different date:

```
json
{
  "type": "interactive",
  "interactive": {
    "type": "button",
    "body": {
      "text": "When was this photo taken?"
    },
    "action": {
      "buttons": [
        {"type": "reply", "reply": {"id": "date_today", "title": "📅 Today"}},
        {"type": "reply", "reply": {"id": "date_yesterday", "title": "📅 Yesterday"}},
        {"type": "reply", "reply": {"id": "date_other", "title": "Enter date..."}}
      ]
    }
  }
}
```

Waterline Confirmation (Image + Buttons)

After processing, the bot displays the annotated image with the detected waterline and requests confirmation:

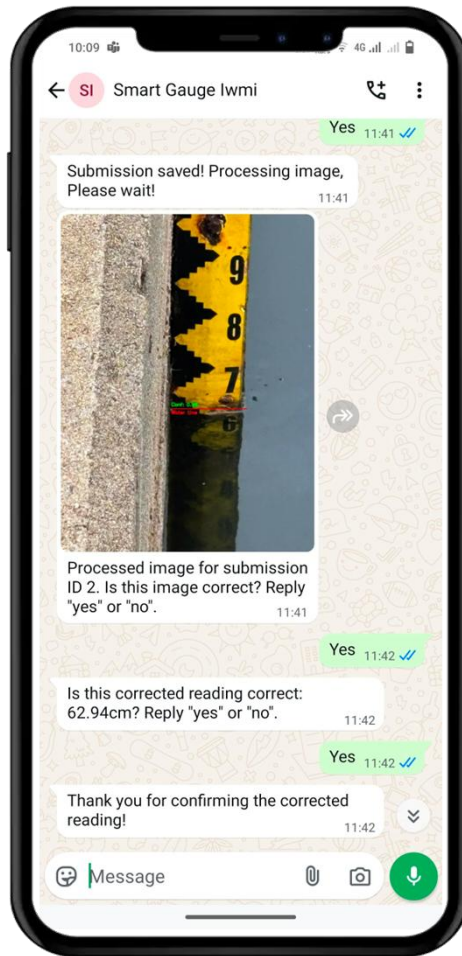


Figure 12 Once complete the Vision bot has identified the water line super imposed it on the image and returned it to the user for confirmation [Source: IWMI]

```
json
{
  "messaging_product": "whatsapp",
  "to": "{{user_phone}}",
  "type": "image",
  "image": {
    "link": "https://api.digitaltwins.iwmi.org/images/annotated_12345.jpg",
    "caption": "Red line shows detected water level. Confidence: 92%"
  }
}
```

Followed immediately by:

```
json
{
  "type": "interactive",
  "interactive": {
    "type": "button",
    "body": {"text": "Is the red line correctly positioned at the water level?"},
    "action": {
      "buttons": [
        {"type": "reply", "reply": {"id": "waterline_yes", "title": "✓ Yes, correct"}},
        {"type": "reply", "reply": {"id": "waterline_no", "title": "X No, retake"}}
      ]
    }
  }
}
```

Reading Confirmation

The AI-extracted reading is presented with confirmation buttons:

```
json
{
  "type": "interactive",
  "interactive": {
    "type": "button",
    "body": {
      "text": "📏 *Water Level Reading*\n\n*42.86 cm*\n\nIs this reading correct?"
    },
    "action": {
      "buttons": [
        {"type": "reply", "reply": {"id": "reading_yes", "title": "✓ Confirm"}},
        {"type": "reply", "reply": {"id": "reading_no", "title": "X Correct it"}}
      ]
    }
  }
}
```

Final Results

The complete submission result is presented in a formatted text message:

```
json
{
  "type": "text",
  "text": {
    "body": "✅ *Submission Complete*\n\n📍 *Station:* Beitbridge\n📅 *Date:* 10 December 2025\n\n*Water Level:* 42.86 cm\n💧 *Discharge:* 15.23 m³/s\n\nThank you for your contribution to water monitoring in the Limpopo Basin! 🌱"
  }
}
```

Complete User Journey

The following table summarises the end-to-end interaction flow:

| Step | User Action | Bot Response | Backend Process |
|------|-----------------------|--------------------------------------------------------|-------------------------------------------------------------|
| 1 | Sends "Hi" | Welcome message + instructions | Create session (IDLE → AWAITING_IMAGE) |
| 2 | Sends gauge photo | "Photo received. Which station?" + station list | Download media, store in S3 |
| 3 | Selects station | "Share your location" + location request | Store station_id in session |
| 4 | Share's location | "When was this taken?" + date buttons | Store coordinates in session |
| 5 | Taps "Today" | "Processing..." then annotated image + confirm buttons | POST /vision/upload_image → POST /vision/object_detection |
| 6 | Taps "✓ Yes, correct" | "Extracting reading..." then reading + confirm buttons | POST /vision/confirm_reading → POST /vision/extract_reading |
| 7 | Taps "✓ Confirm" | Final results message | POST /vision/confirm_ai_reading → POST /vision/discharge |
| 8 | — | Session reset, ready for next submission | Store to DT_gauge_prediction, clear session |

Media Handling

User-submitted images are processed through a secure pipeline:

- Webhook Receipt:** WhatsApp delivers message with media_id reference
- Media URL Retrieval:** Bot calls GET https://graph.facebook.com/v18.0/{media_id} to obtain temporary download URL
- Image Download:** Bot downloads binary from URL (valid ~5 minutes) with OAuth bearer token
- S3 Storage:** Image stored in S3 bucket with submission ID as key
- Vision API Processing:** S3 URL or binary passed to POST /vision/upload_image
- Annotated Image Return:** Vision API returns annotated image, bot uploads to S3 and sends WhatsApp image message with S3 URL

Error Handling

The bot service implements comprehensive error handling to maintain conversation continuity under adverse conditions:

| Error Condition | Detection | User Message | Recovery |
|----------------------|-----------------------------------|---------------------------------------------------------------------|--------------------------|
| Invalid image format | MIME type ≠ image/jpeg, image/png | "Please send a JPEG or PNG photo of the gauge." | Remain in AWAITING_IMAGE |
| Image too small | Resolution < 640px | "Image too small. Please send a clearer photo." | Remain in AWAITING_IMAGE |
| Gauge not detected | Detection confidence < 0.20 | "Couldn't detect the gauge. Please retake with full gauge visible." | Remain in AWAITING_IMAGE |

| | | | |
|------------------------|------------------------------|------------------------------------------------------------------------------|-----------------------------------|
| Low reading confidence | AI confidence < threshold | "Reading uncertain: 42.8 cm. Please confirm or enter correct value." | Proceed with manual option |
| Invalid manual entry | Non-numeric or out of range | "Please enter a number between 0 and 200 (e.g., 45.2)" | Remain in AWAITING_MANUAL_READING |
| Station not recognised | station_id not found | "Station not found. Please select from the list." | Show station list |
| Location too far | Distance > 500m from station | "Location doesn't match station. Please verify you're at the correct gauge." | Prompt re-selection |
| Vision API timeout | Response > 30s | "Processing taking longer than expected. Please wait..." | Retry with backoff |
| Vision API error | 5xx response | "Technical issue. Please try again in a few minutes." | Log error, notify admin |
| Session expired | Redis key missing | "Session expired. Please send a new photo to start again." | Reset to IDLE |
| Rate limit exceeded | plate_quota_month reached | "Monthly limit reached (30 submissions). Contact your supervisor." | Block until reset |

Security Considerations

Authentication Architecture

The system implements a dual authentication strategy: Keycloak OAuth2/OIDC for administrative and API access, and WhatsApp's native phone verification for citizen scientists.

Keycloak Integration:

Administrative users authenticate via Keycloak, receiving JWT Bearer tokens with 5-minute expiry. Refresh token rotation ensures long-lived sessions remain secure, with tokens invalidated after single use. API endpoints are protected using decorator-based access control that validates JWT claims against required Keycloak groups.

WhatsApp User Verification:

Citizen scientists are authenticated through WhatsApp's inherent SIM-based phone verification, providing strong identity assurance without requiring separate credentials. Conversations are bound to verified phone numbers, and first-time users are automatically registered in the system.

Authorisation Model

Role-based access control operates at two levels. Application-level permissions (stored in `user_app_permissions`) control global capabilities: submission creation, moderation rights, administrative access, and monthly submission quotas. Station-level permissions (stored in `user_station_permissions`) provide granular control over which users can submit to or moderate specific monitoring stations.

Data Protection

Encryption in Transit

All connections use TLS 1.2+, including client-to-API (HTTPS), WhatsApp webhooks, database connections (TLS over SSH tunnel), and inter-service communication.

Encryption at Rest

Database storage uses AES-256 encryption via AWS KMS. S3 objects are encrypted using server-side encryption. Session data in ElastiCache is encrypted at rest.

Infrastructure Security

The system deploys within an AWS VPC with public subnets (Application Load Balancer only) and private subnets (ECS tasks, RDS, ElastiCache). Security groups restrict database and cache access to application containers only. Containers run as non-privileged users with resource limits enforced. Credentials are injected via AWS Secrets Manager rather than environment variables.

API Security

All inputs are validated before processing, including MIME type and size limits for images, coordinate range validation, and numeric bounds checking for readings. Rate limiting prevents abuse: 60 image uploads per hour, 10 authentication attempts per minute per IP. WhatsApp webhook requests are validated using HMAC signature verification to prevent spoofing.

Audit and Compliance

Security events (authentication, authorisation denials, submissions, admin actions) are logged to AWS CloudWatch with PII redacted. Data resides in AWS Africa (Cape Town) region where available. Citizen scientists acknowledge data usage terms on first interaction, and location sharing is explicitly requested rather than automatically collected. Third-party data sharing (Meta for WhatsApp, Google for Gemini AI) operates under standard data processing agreements.

Discussion and Conclusions

Achievement of Objectives

The Vision API demonstrates that hybrid AI architectures combining computer vision with large language models can achieve operationally useful accuracy for river gauge reading extraction. Evaluation on 548 images from the Limpopo Basin yielded waterline detection precision of 94.24%, scale gap detection mAP@0.5 of 89%, and reading extraction R^2 of 0.84 with MAE of 5.43 cm under optimal imaging conditions (Vigneswaran et al 2025). These results indicate performance comparable to existing automated gauge reading systems (Liu and Huang 2024) while offering greater adaptability across gauge plate designs.

The three-stage AI processing pipeline—combining YOLOv8 for waterline detection, YOLOv8-Pose for scale gap estimation, and Gemini 2.0 Flash for reading extraction—achieves robust performance under field conditions. Evaluation on the Limpopo Basin dataset demonstrated waterline detection precision of 94.24%, scale gap detection mAP@0.5 of 89%, and reading extraction accuracy of $R^2 = 0.84$ with MAE = 5.43 cm on optimal quality images (Vigneswaran et al 2025). These metrics indicate the system is suitable for operational hydrological monitoring where traditional infrastructure is limited.

The custom WhatsApp Bot Service enables citizen science participation without requiring dedicated app installation, addressing a significant barrier to adoption in resource-constrained contexts. By separating conversation logic from the messaging transport layer, the architecture provides full control over user experience while leveraging WhatsApp's ubiquitous presence across the Limpopo Basin's four riparian countries. The two-stage validation workflow serves a dual purpose: ensuring data quality through citizen confirmation while continuously generating ground-truth training data that enables ongoing model improvement without dedicated annotation campaigns.

Innovation and Contribution

The protocol represents a novel contribution to citizen science approaches for hydrological monitoring, addressing the "data paradox" inherent to Digital Twin development—basins with the weakest monitoring infrastructure stand to benefit most from digital decision support, yet lack the observational data required to drive such systems (Roman, 2025).

The hybrid AI architecture combines computer vision for geometric measurement with large language models for numerical extraction, addressing limitations inherent to single-model approaches. The manuscript findings confirm this design rationale: processing with image alone achieved $R^2 = 0.16$, while incorporating scale gap metadata improved performance to $R^2 = 0.58-0.84$, demonstrating that geometric context from YOLOv8-Pose is essential for accurate reading extraction (Vigneswaran et al 2025). The algorithmic combination ($W = M - R \times 10$) leverages each model's strengths rather than relying on end-to-end AI interpretation.

The WhatsApp-based interface significantly lowers barriers to participation compared to dedicated mobile applications. This design prioritises digital inclusiveness, recognising that potential citizen scientists may face constraints including limited smartphone storage, unreliable data connections, or unfamiliarity with app installation procedures. The conversational interface with interactive buttons and multi-language support further reduces friction for users with varying digital literacy levels. The protocol's transboundary design, with station-level permissions and integration with the FlowTracker rating curve database, enables coordinated data collection across the basin's four countries under LIMCOM governance structures.

Limitations and Future Work

Several limitations warrant acknowledgement. The current training dataset comprises 548 images from Limpopo Basin gauge plates; performance on gauge designs from other regions requires validation before broader deployment. Model accuracy degrades under challenging imaging conditions—optimal quality images achieved MAE = 5.43 cm while sub-optimal images (blur, corrosion, poor lighting) showed MAE = 11.70 cm, though the validation workflow mitigates this by enabling user correction. The system's dependence on third-party APIs (Google Gemini, Meta WhatsApp) introduces potential points of failure outside operator control, and the image processing workflow requires sustained connectivity that may challenge users in areas with intermittent coverage.

Future development will focus on continuous model improvement through retraining on accumulated validation data, offline capability for image capture during connectivity gaps, expanded language support (Setswana, Shona, Xitsonga), image quality scoring to provide immediate user feedback, and integration with the UNICEF YOMA blockchain platform for citizen scientist recognition and credentialing.

Conclusion

The Vision API and associated protocol demonstrate that citizen science, combined with hybrid artificial intelligence, can meaningfully augment traditional hydrological monitoring in data-scarce transboundary basins. The system achieves reading accuracy comparable to previous automated gauge reading methods (Liu and Huang 2024) while offering superior adaptability across gauge plate designs through LLM-based interpretation.

By lowering technical barriers through WhatsApp-based interaction, embedding quality assurance through two-stage validation, and integrating with existing hydrological infrastructure through the FlowTracker API, the system enables community members to contribute scientifically valuable discharge observations while building local engagement with water resource management. This work provides a foundation for scaling the approach for scaling the approach across the Limpopo Basin's planned network of 80 citizen scientists generating 10,000 annual observations, and potentially to other river systems facing similar monitoring challenges.

References

- Afham, Abdul; Silva, Paulo; Ghosh, Surajit; Kiala, Zolo; Retief, H.; Dickens, Chris; Garcia Andarcia, Mariangel. 2024. Limpopo River Basin Digital Twin Open Data Cube Catalog. Colombo, Sri Lanka: International Water Management Institute (IWMI). CGIAR Initiative on Digital Innovation. 22p.
- Beven, K.J. 2012. Rainfall-runoff modelling: the primer. 2nd ed. Chichester, UK: Wiley-Blackwell.
- Bonney, R.; Cooper, C.B.; Dickinson, J.; Kelling, S.; Phillips, T.; Rosenberg, K.V.; Shirk, J. 2009. Citizen science: a developing tool for expanding science knowledge and scientific literacy. *BioScience*, 59(11): 977–984. doi:10.1525/bio.2009.59.11.9.
- Buytaert, W.; et al. 2014. Citizen science in hydrology and water resources: opportunities for knowledge generation, ecosystem service management, and sustainable development. *Frontiers in Earth Science*, 2: 26. doi:10.3389/feart.2014.00026.
- Conrad, C.C.; Hilchey, K.G. 2011. A review of citizen science and community-based environmental monitoring: issues and opportunities. *Environmental Monitoring and Assessment*, 176(1–4): 273–291. doi:10.1007/s10661-010-1582-5.
- 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. 4p.
- Dell, N.; Kumar, N. 2016. The ins and outs of HCI for development. In: Proceedings of the 2016 CHI conference on human factors in computing systems. San Jose, CA, USA: ACM. pp. 2220–2232. doi:10.1145/2858036.2858081.
- Fowler, M. 2002. Patterns of enterprise application architecture. Boston, USA: Addison-Wesley.
- Hannah, D.M.; et al. 2011. Large-scale river flow archives: importance, current status and future needs. *Hydrological Processes*, 25(7): 1191–1200. doi:10.1002/hyp.7794.
- International Water Management Institute (IWMI). 2024. Citizen science for water management in Limpopo river basin: project proposal. Colombo, Sri Lanka: International Water Management Institute (IWMI).
- Langa, N.; Kiala, Z. 2025. Citizen scientists take the lead in tracking southern Africa's transboundary river basin. IWMI Blog. 11 November 2025. Available at: <https://www.iwmi.org/blogs/citizen-scientists-take-the-lead-in-tracking-southern-africas-transboundary-river-basin/>.
- Limpopo Watercourse Commission (LIMCOM). 2019. Limpopo river basin monograph. Maputo, Mozambique: Limpopo Watercourse Commission (LIMCOM).
- Liu, W.-C.; Huang, W.-C. 2024. Evaluation of deep learning computer vision for water level measurements in rivers. *Heliyon*, 10: e25989. doi:10.1016/j.heliyon.2024.e25989.
- Rasheed, A.; San, O.; Kvamsdal, T. 2020. Digital twin: values, challenges and enablers from a modeling perspective. *IEEE Access*, 8: 21980–22012. doi:10.1109/ACCESS.2020.2970143.
- Roman, H. 2025. Quoted in: Storr, S. 2025. A network of citizen scientists to protect freshwater resources in southern Africa. IWMI Blog. 18 September 2025. Available at: <https://www.iwmi.org/blogs/a-network-of-citizen-scientists-to-protect-freshwater-resources-in-southern-africa/>.
- See, L.; et al. 2016. Crowdsourcing, citizen science or volunteered geographic information? the current state of crowdsourced geographic information. *ISPRS International Journal of Geo-Information*, 5(5): 55. doi:10.3390/ijgi5050055.
- Seibert, J.; Strobl, B.; Etter, S.; Hummer, P.; van Meerveld, H.J. 2019. Virtual staff gauges for crowd-based stream level observations. *Frontiers in Earth Science*, 7: 70. doi:10.3389/feart.2019.00070.
- Storr, S. 2025. A network of citizen scientists to protect freshwater resources in southern Africa. IWMI Blog. 18 September 2025. Available at: <https://www.iwmi.org/blogs/a-network-of-citizen-scientists-to-protect-freshwater-resources-in-southern-africa/>.
- Thornhill, I.; Loiselle, S.; Lind, K.; Ophof, D. 2016. The citizen science opportunity for researchers and agencies. *BioScience*, 66(9): 720–721. doi:10.1093/biosci/biw089.
- Vigneswaran, K.; Retief, H.; Clifford-Holmes, J.; Garcia Andarcia, M.; Tennakoon, H. 2025. Hybrid framework for automated river gauge reading: integrating YOLOv8 waterline detection and Gemini 2.0 Flash LLM. Unpublished manuscript. Colombo, Sri Lanka: International Water Management Institute (IWMI).

Yang, Z.; Li, L.; Wang, J.; Lin, K.; Azarnasab, E.; Ahmed, F.; Liu, Z.; Liu, C.; Zeng, M.; Wang, L. 2023. MM-REACT: Prompting ChatGPT for multimodal reasoning and action. arXiv:2303.11381. Available at: <https://arxiv.org/abs/2303.11381>.

Wiggins, A.; Crowston, K. 2011. From conservation to crowdsourcing: a typology of citizen science. In: Proceedings of the 44th Hawaii international conference on system sciences. Kauai, HI, USA: IEEE. pp. 1–10. doi:10.1109/HICSS.2011.207.



Appendix A: API Endpoint Reference

Overview

The Vision API exposes a RESTful interface for the discharge measurement protocol, organised into two endpoint groups: core vision processing endpoints that implement the five-stage workflow, and authentication endpoints that manage user identity and access control. All endpoints follow consistent conventions for request/response formatting, error handling, and authentication.

Base URL: <https://api.digitaltwins.iwmi.org>

Content Types:

- Request: multipart/form-data for image uploads; application/x-www-form-urlencoded for other endpoints
- Response: application/json for data responses; image/jpeg for annotated images

Authentication: Unless otherwise specified, endpoints require a valid JWT Bearer token in the Authorization header, obtained through the authentication endpoints.

Core Vision Processing Endpoints

The six core endpoints implement the discharge measurement workflow sequentially. Each endpoint corresponds to a specific processing stage and must be called in order for a given submission.

POST /vision/upload_image

Purpose: Initial image submission from WhatsApp bot with associated metadata. This endpoint is public (no authentication required) to facilitate seamless WhatsApp bot integration.

Authentication: None required (public endpoint to facilitate WhatsApp bot integration)

| Request Parameters:Parameter | Type | Required | Description |
|------------------------------|--------|----------|-----------------------------------------------------------|
| image | File | Yes | Gauge plate photograph (JPEG or PNG format, max 10MB) |
| user_id | String | Yes | WhatsApp user identifier (phone number hash or unique ID) |
| timestamp | String | Yes | Capture timestamp in format YYYY-MM-DD HH:MM:SS |
| longitude | Float | No | GPS longitude coordinate (decimal degrees, WGS84) |
| latitude | Float | No | GPS latitude coordinate (decimal degrees, WGS84) |
| station | String | No | Gauging station identifier (e.g., B7H026) |
| imageSendDate | String | No | Date image was sent via WhatsApp (YYYY-MM-DD) |
| imageSendTime | String | No | Time image was sent via WhatsApp (HH:MM:SS) |

Processing Logic:

1. Validates presence of required fields (image, user_id, timestamp)
2. Validates timestamp format compliance
3. Validates file type (JPEG/PNG only) and size constraints
4. Generates unique submission_id
5. Inserts metadata record into DT_GAUGE_SUBMISSION table

6. Stores binary image data in database BLOB field
7. Saves temporary file copy for subsequent processing

Response (Success - 201 Created):

```

json
{
  "status": "success",
  "submission_id": 12345,
  "message": "Image uploaded successfully"
}

```

Response (Error - 400 Bad Request):

```

json
{
  "status": "error",
  "message": "Invalid timestamp format. Expected YYYY-MM-DD HH:MM:SS"
}

```

Error Codes:

| Code | Condition |
|------|-------------------------------------------------------------------------|
| 400 | Missing required field, invalid timestamp format, unsupported file type |
| 413 | Image file exceeds size limit |
| 500 | Database insertion failure |

POST /vision/object_detection

Purpose: Execute YOLOv8 model to detect gauge plate boundaries and water line intersection point.

Authentication: Bearer Token (required groups: dt-vision-users or vision-app-admin)

Request Parameters:

| Parameter | Type | Required | Description |
|---------------|---------|----------|-----------------------------------------------------------|
| submission_id | Integer | Yes | Submission identifier from upload_image response |
| user_id | String | Yes | WhatsApp user identifier (must match original submission) |

Processing Logic:

1. Retrieves original image from DT_GAUGE_SUBMISSION table
2. Loads YOLOv8 (gauge plate detection variant)
3. Executes inference via user_image_val() function
4. Detects gauge plate bounding box coordinates
5. Identifies water line intersection at bottom edge of detected region
6. Draws red annotation line at water intersection point
7. Overlays confidence score on annotated image
8. Crops detected gauge region for subsequent reading extraction
9. Saves annotated image to temporary storage

Response (Success - 200 OK): Returns annotated JPEG image with Content-Type: image/jpeg

The annotated image displays:

- Original photograph with gauge plate region highlighted
- Red horizontal line indicating detected water level
- Confidence percentage overlaid on image

Response (Error - 404 Not Found):

```
json
{
  "status": "error",
  "message": "Submission not found or image not available"
}
```

Response (Error - 422 Unprocessable Entity):

```
json
{
  "status": "error",
  "message": "Gauge plate not detected in image. Please retake photograph."
}
```

Error Codes:

| Code | Condition |
|------|-----------------------------------------------------------------|
| 401 | Missing or invalid authentication token |
| 403 | User not in authorised group |
| 404 | Submission ID not found |
| 422 | Model failed to detect gauge plate (confidence below threshold) |
| 500 | Model inference failure |

POST /vision/extract_reading

Purpose: Extract numerical water level reading from cropped gauge image using dual-AI pipeline (YOLOv8 + Gemini LLM).

Authentication: Bearer Token (required groups: dt-vision-users or vision-app-admin)

Request Parameters:

| Parameter | Type | Required | Description |
|---------------|---------|----------|--------------------------|
| submission_id | Integer | Yes | Submission identifier |
| user_id | String | Yes | WhatsApp user identifier |

Processing Pipeline:

1. **Prerequisite Check:** Verifies cropped gauge image exists from prior object_detection call. Returns error if detection stage was skipped or failed.
2. **Scale Detection:** Executes YOLOv8-Pose model on cropped image to detect individual scale marker positions through keypoint localisation.. Calculates scale_gap_pixels as average pixel distance between consecutive detected markers.
3. **LLM Vision Analysis:** Sends cropped gauge image to Gemini 2.0 Flash with structured prompt requesting:

- a. top_number: Highest visible number on gauge scale
- b. number_seq: Sequence direction (ascending/descending)
- c. visible_no: Number closest to (but above) the water line

4. **Reading Calculation:** Applies geometric formula:

$$\text{corrected_reading} = (\text{visible_no} \times 10) - (10 / \text{scale_gap_pixels} \times \text{gap_to_water})$$

Where gap_to_water is the pixel distance from the nearest scale marker to the detected water line.

5. **Database Storage:** Inserts prediction record into DT_gauge_prediction table with AI-extracted reading, scale gap measurement, and processing timestamp.

Response (Success - 200 OK):

```
json
{
  "status": "success",
  "submission_id": 12345,
  "ai_reading": 47.5,
  "scale_gap_pixels": 28.3,
  "confidence": "high",
  "message": "Reading extracted successfully"
}
```

Response (Error - 400 Bad Request):

```
json
{
  "status": "error",
  "message": "Object detection must be completed before reading extraction"
}
```

Error Codes:

| Code | Condition |
|------|-----------------------------------------------------------|
| 400 | Prerequisites not met (detection not completed) |
| 401 | Missing or invalid authentication token |
| 422 | Scale markers not detected; LLM failed to extract numbers |
| 502 | Gemini API unavailable or returned error |
| 500 | Reading calculation failure |

Notes:

- The reading value is returned in centimetres
- Confidence level is derived from scale detection quality and LLM response consistency
- If scale gap detection fails, the endpoint returns an error rather than an unreliable reading

POST /vision/confirm_reading

Purpose: First-stage user confirmation of object detection results (water line position accuracy).

Authentication: Bearer Token (required groups: dt-vision-users or vision-app-admin)

Request Parameters:

| Parameter | Type | Required | Description |
|-------------------|---------|-------------|-------------------------------------------------------------------|
| submission_id | Integer | Yes | Submission identifier |
| user_id | String | Yes | WhatsApp user identifier |
| is_correct | String | Yes | User response: "yes", "no", or "need help" |
| corrected_reading | Float | Conditional | Required when is_correct is "no"; user-provided water level in cm |

Processing Logic:

1. Retrieves prediction record for submission
2. Records user validation response in user_validation column
3. If is_correct is "no", stores user-provided value in user_corrected_reading column
4. Updates prediction timestamp

Response (Success - 200 OK):

```
json
{
  "status": "success",
  "submission_id": 12345,
  "validation_recorded": true,
  "user_response": "yes"
}
```

Validation Outcomes:

| is_correct Value | System Behaviour |
|------------------|---------------------------------------------------------------------------|
| "yes" | Water line position accepted; proceed to reading extraction |
| "no" | User correction recorded; manual reading used for subsequent calculations |
| "need help" | Flags submission for review; may trigger support workflow |

Notes:

- This validation step generates ground-truth data for YOLOv8 model retraining
- User corrections take priority over AI detections in downstream calculations
- The "need help" response enables graceful handling of edge cases

POST /vision/confirm_ai_reading

Purpose: Second-stage confirmation of AI-extracted numerical reading.

Authentication: Bearer Token (required groups: dt-vision-users or vision-app-admin)

Request Parameters:

| Parameter | Type | Required | Description |
|---------------|---------|----------|--------------------------------------------|
| submission_id | Integer | Yes | Submission identifier |
| user_id | String | Yes | WhatsApp user identifier |
| is_correct | String | Yes | User response: "yes", "no", or "need help" |

corrected_reading Float Conditional Required when is_correct is "no"; user-provided reading in cm

Processing Logic:

1. Retrieves prediction record for submission
2. Records user validation response in user_validation_2 column
3. If is_correct is "no", stores user-provided value in user_corrected_reading_2 column
4. Updates prediction timestamp

Response (Success - 200 OK):

```
json
{
  "status": "success",
  "submission_id": 12345,
  "validation_recorded": true,
  "user_response": "yes",
  "final_reading": 47.5
}
```

Notes:

- This two-stage validation separates geometric accuracy (water line position) from numerical accuracy (reading extraction)
- Second-stage corrections generate training data for Gemini prompt refinement
- The final_reading in the response reflects any user corrections applied

POST /vision/discharge

Purpose: Calculate river discharge from confirmed water level reading using station-specific rating curves.

Authentication: Bearer Token (required groups: dt-vision-users or vision-app-admin)

Request Parameters:

| Parameter | Type | Required | Description |
|---------------|---------|----------|--------------------------------------------------------------|
| submission_id | Integer | Yes | Submission identifier |
| height | Float | No | Explicit water level override (cm); bypasses stored readings |

Processing Logic:

1. Station Retrieval: Fetches station identifier from DT_GAUGE_SUBMISSION record
2. Reading Priority Resolution: Determines final water level reading using priority order:
 - a. Explicit height parameter (if provided)
 - b. user_corrected_reading (first-stage user correction)
 - c. AI reading (if user_validation is "yes")
 - d. user_corrected_reading_2 (second-stage user correction)
3. Unit Conversion: Converts reading from centimetres to metres for FlowTracker API compatibility
4. Rating Curve Lookup: Calls FlowTracker API endpoint: GET https://inwards.award.org.za/api/flowtracker/fetch_rating?station={station_id}&level={height_metres}
5. Response Processing: Extracts discharge value from FlowTracker response; handles interpolation between rating curve points

Response (Success - 200 OK):

```
json
{
  "status": "success",
  "submission_id": 12345,
  "station": "B7H026",
  "water_level_cm": 47.5,
  "water_level_m": 0.475,
  "discharge_m3s": 15.23,
  "rating_curve_date": "2024-06-15"
}
```

Response (Error - 404 Not Found):

```
json
{
  "status": "error",
  "message": "Station B7H026 not found in FlowTracker database"
}
```

Response (Error - 422 Unprocessable Entity):

```
json
{
  "status": "error",
  "message": "Water level 250cm exceeds rating curve range for station B7H026"
}
```

Error Codes:

| Code | Condition |
|------|-----------------------------------------------------------|
| 400 | No valid reading available (all sources null/unvalidated) |
| 404 | Station not found in FlowTracker; submission not found |
| 422 | Water level outside rating curve valid range |
| 502 | FlowTracker API unavailable |
| 500 | Discharge calculation failure |

Notes:

- Discharge is returned in cubic metres per second (m³/s)
- The rating_curve_date indicates when the station's rating curve was last updated
- Rating curves are maintained by AWARD through periodic field calibration campaigns

Authentication Endpoints

The Vision API implements OAuth2/OIDC authentication through Keycloak, providing secure identity management and group-based access control. These endpoints manage user registration, session lifecycle, and token management.

Token Details:

- access_token: Short-lived JWT (5 minutes) for API authorisation
- refresh_token: Long-lived token (30 days) for obtaining new access tokens

- `id_token`: Contains user identity claims (OpenID Connect)

| Endpoint | Description |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------|
| POST <code>/vision/auth/register</code> | Creates new user in Keycloak; adds to <code>dt-vision-users</code> group; issues JWT tokens |
| POST <code>/vision/auth/login</code> | Authenticates user; returns <code>access_token</code> , <code>refresh_token</code> , <code>id_token</code> |
| POST <code>/vision/auth/refresh</code> | Refreshes expired access token using refresh token |
| POST <code>/vision/auth/logout</code> | Invalidates user session |
| GET <code>/vision/me</code> | Returns current user's profile information |



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-portfolio-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)

