Go, change the world

RV College of Engineering, Bengaluru - 560059.

Title: Scalable Cloud Native Architecture for Real-Time Wildlife Anomaly Detection

Amol Vyas | Akshat Gupta | Ayush Aman
1RV23CS032 | 1RV23CS027 | 1RV23CS059 **SDG:15**

Introduction

Biodiversity monitoring is critical for achieving SDG 15 (Life on Land), yet conservationists struggle with the manual processing of terabytes of camera trap imagery. Traditional web platforms often fail under the heavy load of field data uploads due to synchronous blocking. This project introduces a **Cloud-Native Wildlife Detection System** that automates species identification using AI while ensuring system stability through a decoupled microservices architecture.

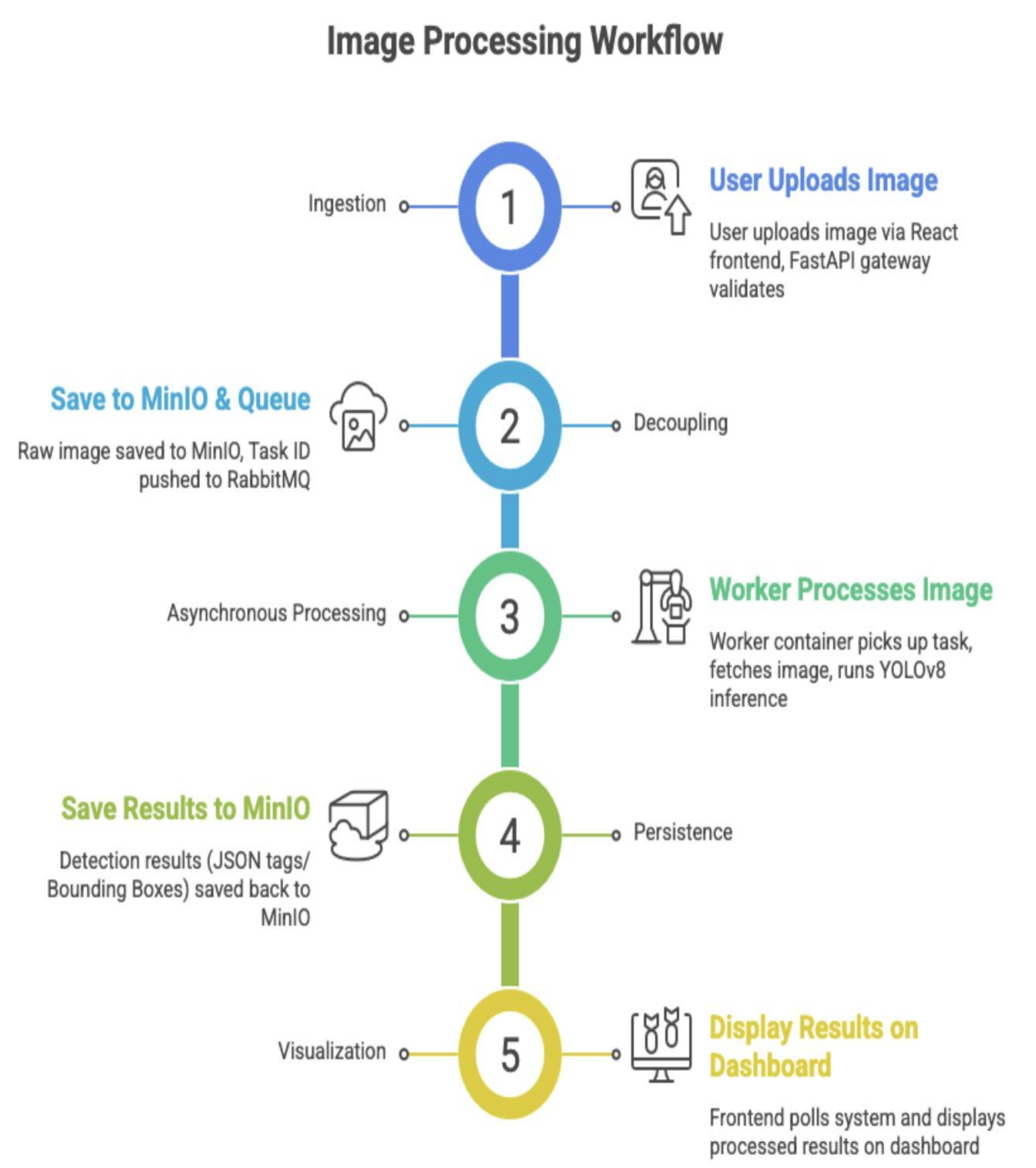
Problem Definition

The Bottleneck: Manual analysis of wildlife images is labor-intensive and delays conservation insights by weeks or months.
The Technical Failure: Existing monolithic applications crash when researchers attempt to upload large batches of images simultaneously because the server tries to process them instantly (blocking I/O).

Objectives

Develop a Resilient Architecture: separate the User Interface from heavy AI computation using asynchronous message queuing.
Ensure Zero-Latency Ingestion: allow users to upload massive datasets with immediate acknowledgement
Scalable AI integration: deploy models in isolated containers

Methodology



Tools used

- **Frontend:** React (Vite), Tailwind CSS (User Interface).
- **Backend API:** Python FastAPI (Asynchronous Request Handling).
- **Message Broker:** RabbitMQ (Task Queuing & Decoupling).
- **Storage:** MinIO (S3-Compatible Object Storage).
  AI Model: YOLOv8 Nano (Real-time Object Detection).
- **Infrastructure:** Docker & Docker Compose.

Results and Discussions

- **Performance:** In load tests simulating **500 concurrent users**, the system maintained a **100% success rate** with zero dropped connections.
- **Latency:** The median response time for uploads was **127ms** (milliseconds), proving that the asynchronous queue successfully prevented server blocking.
- **Scalability:** The system demonstrated that AI processing speed is independent of upload speed; the queue acts as a buffer, allowing the system to absorb traffic spikes without crashing.

Conclusions

The project successfully demonstrates that transitioning from a monolithic to an **event-driven microservices architecture** solves the critical scalability challenges in wildlife monitoring[19]. By utilizing **RabbitMQ** and **MinIO**, we achieved a "crash-proof" ingestion pipeline that aligns with modern cloud standards. This architecture provides a robust, reproducible blueprint for deploying AI tools in resource-constrained field environments.

Outcome of the work

- **Product:** A fully functional, containerized **Wildlife Anomaly Detection Prototype** capable of real-time ingestion and asynchronous processing.
- **Publication:** A research paper.
- **SDG Impact:** Direct contribution to **SDG 15 (Life on Land)** by accelerating data analysis for biodiversity conservation.

References

1. M. S. Norouzzadeh et al., "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," Proc. Natl. Acad. Sci. U.S.A., 2018.
2. C. Lin et al., "An AI-based wild animal detection system for smart city applications," Biodivers. Data J., 2023.
3. G. Jocher et al., "YOLOv8 by Ultralytics," 2023.

Mentor

Dr. Ramakanth Kumar P