Intruder Detection - Distributed Systems

Alan Gallo, Cedric Sillaber, Frantisek Sova

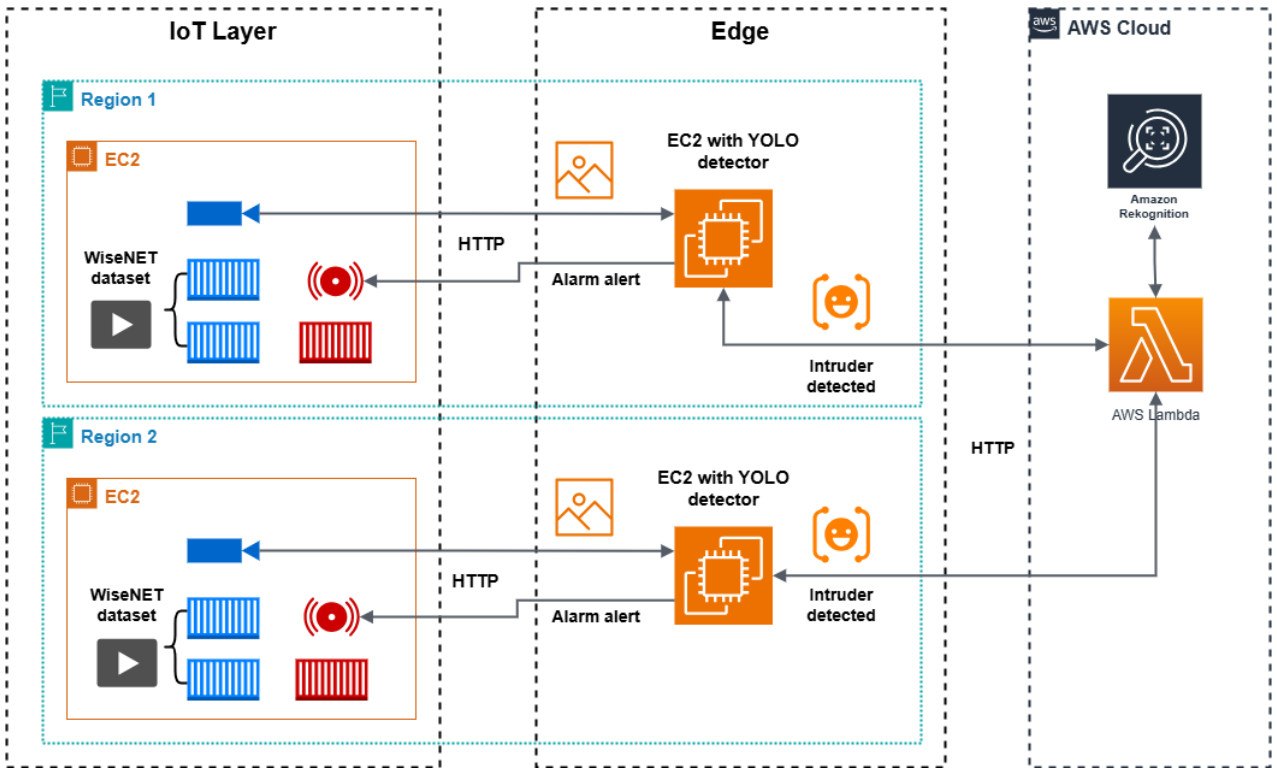
January 2025

Content

- System architecture
- Implementation details
- Evaluation

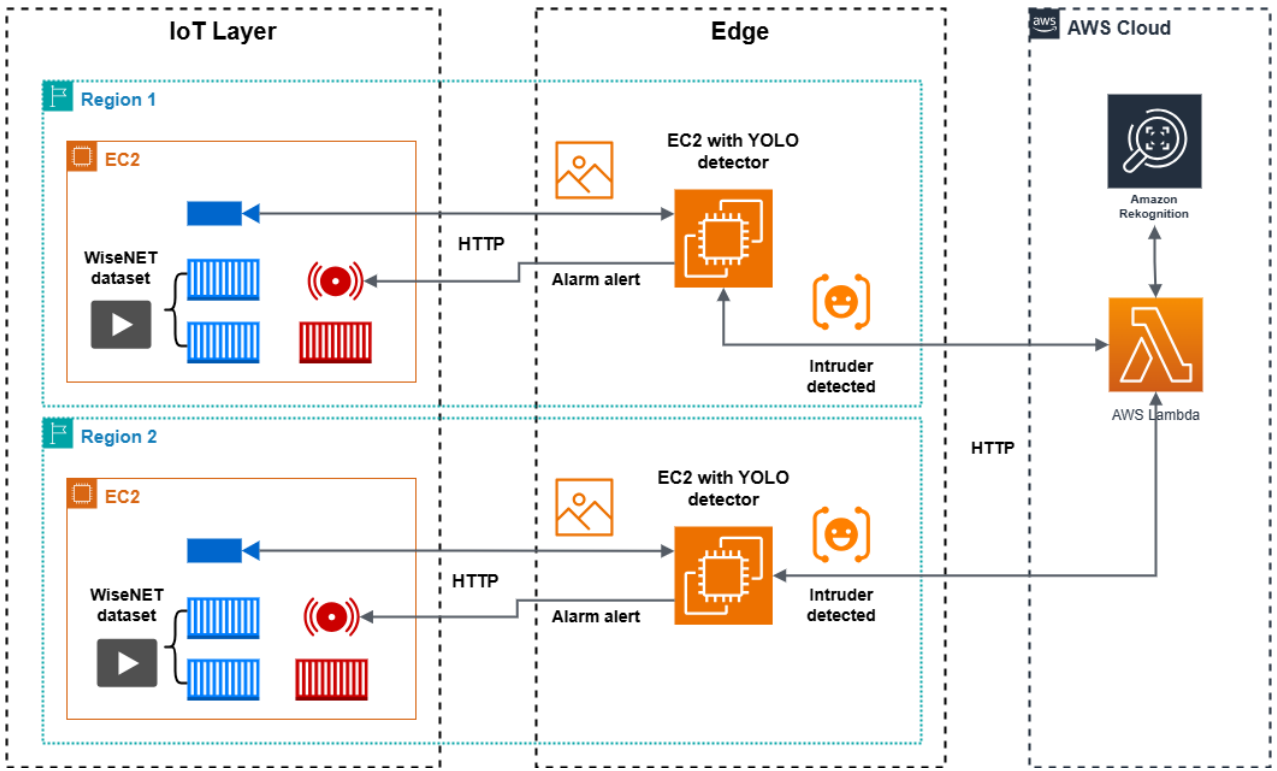
System architecture 1

System architecture in milestone 1:



System architecture 2

Final system architecture:



IOT - Edge

- socketIO for communication
- bla bla

socket.io

Edge - IOT

- Flask Rest Api
- bla bla

IoT Implementation Highlights

Frame Processing

- Key Features:
  - Configurable frame skip
  - Real-time frame rate simulation
  - Robust error handling

Communication

- socket.io for communication between IOTs and Cloud.
- good for continuous dataflow
- setup once, then just send data

socket.io

Edge Server Architecture

Async Processing Pipeline

- two workers working asynchronously:
  - worker 1: Frame buffer queue
  - worker 2: YOLO-based person detection
- Cloud service integration

Communication: Flask REST

- REST API for communication between EDGE and Cloud
- Edge sends HTTP request to Cloud $\rightarrow$ Cloud sends HTTP response

Cloud Service Features

AWS Rekognition Integration

- Face collection management
- Known face indexing
- High-confidence matching

Security Features

- Configurable confidence thresholds
- Automatic collection management
- Error handling and logging

Example Workflow/ Controlgraph

example of intruder detection. what happens at:

- IOT? what are steps there
- EWDGE? what are the possible steps
- Cloud
- back to edge and alarm, why and how

System Performance

current blueprint generated by GenAI

- **Detection Metrics**
  - YOLO detection ratio
  - Face recognition accuracy
  - Processing latency
- **System Reliability**
  - Connection resilience
  - Error handling effectiveness
  - Resource utilization
- **Scalability**
  - Multiple camera support
  - Queue performance
  - Cloud service responsiveness

Evaluation