**ResQNet+ — Deployment Submission**

**Project: ResQNet+ — AI Emergency Response & Satellite-Based Disaster Monitoring (Liberia)**

**1. Overview**

This deployment describes how ResQNet+ moves from trained models to a reliable, secure, and observable production system supporting rapid emergency alerts and satellite-driven hazard detection (floods, bushfires) in Liberia. The deployment pipeline includes: (a) packaging/serializing models, (b) containerizing the serving API, (c) provisioning cloud resources (API, job workers, database, object storage), (d) standing up observability, and (e) a staged rollout (dev → staging/UAT → prod) with canary releases. The mobile app (Android/PWA) and an operations dashboard for NDMA/first responders consume the same backend APIs.

**2. Model Serialization**

Primary models: (i) satellite hazard segmentation/detection, (ii) report triage/anomaly scoring. Models are exported to ONNX for portable, high‑performance CPU inference with ONNX Runtime. We also retain framework‑native artifacts for reproducibility:  
• TensorFlow/Keras: SavedModel (.pb) for segmentation baselines.  
• Scikit‑learn/XGBoost: joblib/JSON.  
Versioning: Each model is assigned a semantic version (e.g., hazard-seg: 1.2.0) and stored in object storage under a model registry path (s3://resqnet-models/ {model name}/{version}/). Metadata (training data hash, metrics, hyperparameters) is logged alongside artifacts. For storage efficiency we enable optional float16 weights and ZIP compression; mobile‑side models (if any) use quantization‑aware variants.

**3. Model Serving**

Real‑time serving: FastAPI microservice (Python) hosting ONNX Runtime sessions for CPU‑first, autoscaled containers behind a load balancer. TensorRT/GPU workers can be enabled for surge events. Batch inference: Celery workers pull satellite tiles (Sentinel‑1/2, MODIS) from object storage, run scheduled detection, and write hotspots to PostGIS for map visualizations. All services run in containers with distinct deployments for dev/staging/prod; environment configs are managed via IaC (Terraform) and secrets in a managed vault.

**4. API Integration**

Model APIs are exposed behind an authenticated gateway. Key endpoints (JSON over HTTPS):  
• POST /v1/predict/hazard — body: { "tile\_uri": "s3://.../tile.tif", "bbox": [minx,miny,maxx,maxy], "ts": "ISO8601" }  
 ↳ response: { "hotspots": [{"lon": ..., "lat": ..., "score": 0.93}], "mask\_uri": "s3://.../mask.tif" }  
• POST /v1/triage/report — body: citizen/first‑responder incident payload (text, media URIs, GPS)  
 ↳ response: { "priority": "HIGH|MEDIUM|LOW", "confidence": 0.88 }  
• GET /v1/models/{name}/versions/latest — returns active model version and checksum.  
Input formats follow GeoTIFF/COG for imagery and GeoJSON for geometry; responses include confidence scores and URIs to derived artifacts.

**5. Security Considerations**

• Transport & Identity: TLS 1.2+, OAuth2/JWT; short‑lived tokens issued via the gateway; per‑role scopes (Citizen, Responder, Dispatcher, Admin). API keys only for system integrations.  
• Authorization & Data Minimization: RBAC enforced at the gateway and service layers; exact citizen GPS coordinates are visible only to authorized responders—public maps show reduced‑precision geohashes. PII is encrypted at rest (KMS) and redacted in logs. Media stored with object‑level access policies.  
• App & Infra Hardening: Least‑privilege IAM, network policies, WAF/rate‑limiting, signed container images, dependency scanning (Trivy), and audit logs. Incident data retention follows NDMA agreements (e.g., 24 months) with periodic purges.  
• Secrets: Managed secret store; no secrets in images; rotation policies in place.

**6. Monitoring and Logging**

Observability stack: Prometheus/Grafana for metrics, OpenTelemetry traces, and centralized logs (ELK/OpenSearch). Key metrics and alerts:  
• Service SLOs: p95 latency < 300 ms (real‑time predict), 5xx error rate < 2%.  
• Pipeline Health: queue backlog age, batch job duration, satellite tile throughput.  
• Model Quality: precision/recall on labeled pilot areas; drift via PSI/KS on inputs; threshold stability. Trigger retraining when drift persists > 7 days or precision < 0.85.  
• Delivery: push/SMS success > 98%; incident creation to dispatch < 60s (p95).  
Dashboards: Ops (SRE), Incident Command (NDMA), and Model Quality. Pager alerts on sustained SLO breaches; weekly reports summarize uptime, incidents handled, and model health.