QAI\_LLM\_Framework\_and\_DevOps

# Part I — QAI LLM Architecture & Models

## 1. Short Summary (One Sentence)

Design a family of QAI LLMs (general + class-specific) as hybrid, modular LLM subsystems that run on constrained local hardware or cloud, integrate tightly with your QAI processor/OS/datacenter and Org/Business frameworks, support agentic modes & RAG, and are governed by ethics, security, and software/system engineering standards.

## 2. High-Level Architecture

1. Model layer  
 - Family of LLMs: micro-LMs (tiny), edge-LMs (small/quantized), mid/large models (cloud/QPU-assisted).  
 - Modality adapters: text, vision, audio, sensor streams, task-specific encoders for robotics/CPS.  
 - Agent layer: orchestration of skills (tool use, RAG, planners, executors).  
  
2. Data & retrieval layer  
 - QAI Vector DB + hybrid RAG indexers (local + distributed).  
 - ETL/ELT connectors: warehouse, star schemas, time-series stores, HIL (human-in-the-loop) labeling.  
  
3. Runtime & infra layer  
 - Local runtime (QAI OS): optimized kernels for QAI processor / NPU / accelerator.  
 - Inference serving: multi-mode (standalone, cluster, air-gapped).  
 - Edge orchestration: lightweight orchestrator, offline batch scheduler.  
  
4. Ops & governance  
 - QAI OpsAgent: CI/CD, policy enforcement, auditing, model registry, artifact signing.  
 - Observability: telemetry for latency, accuracy, drift, resource usage.  
  
5. Security & compliance  
 - Encryption, secure boot, PQC for sensitive flows, RBAC + audit logs.

## 3. Class Types & Taxonomies

Organized by purpose, footprint, and modality:  
- Purpose: personal assistant, office productivity, social, industrial/CPS, scientific, compliance/legal.  
- Footprint: Tiny (≤50M params), Small (50–500M), Medium (500M–4B), Large (>4B).  
- Modality: text-only, multimodal, embedded robotics connectors.

## 4. Data Architecture & Pipelines

Data sources include sensors, MIS DBs, time series, external GenAI datasets.  
Storage tiers: cold (lake), warm (warehouse), fast (vector DB), time-series.  
Labeling & privacy: HIL annotation, PII scrubbers, differential privacy.

## 5. Model Training & Optimization

Compression: distillation, quantization, pruning.  
Training: centralized, federated, on-device continual.  
Hardware-aware: kernels tuned for QAI processor.

## 6. Serving & Deployment Modes

Modes include real-time, batch/offline, agentic crew.  
Form factors: on-device, edge, cloud.  
Resilience: graceful degradation, telemetry sync.

## 7. Ethics & Safety

Model cards & datasheets.  
Explainability modules.  
Safety-first fallback strategies.

# Part II — QAI LLM DevOps Lifecycle & Testing

This section defines the DevOps lifecycle, standards compliance, testing, troubleshooting, non‑functional testing, configuration templates, training & release practices for the QAI LLM family. It is intended as a modular section that links back to requirements, taxonomy, QAI stack components and business frameworks.

## 1. DevOps lifecycle overview

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 2. Standards & compliance mapping

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 3. Artifacts & traceability

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 4. CI/CD pipeline (model + infra)

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 5. Training pipeline for QAI models (classical + quantum-assisted)

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 6. Non-functional testing (NFA) & test catalog

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 7. Troubleshooting & incident runbooks

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 8. Templates & placeholders for dynamic configuration

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 9. Release windows, gating & rollout strategy

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 10. Inputs/Outputs taxonomy & interfaces

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 11. Agent / Quantum / Abstract class sketches (developer view)

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 12. Observability, SLOs & metrics

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 13. Security hardening & PQC artifact flows

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.

## 14. Appendix: sample YAMLs, checklists, and templates

See detailed descriptions as defined in the DevOps lifecycle and testing framework document. This includes lifecycle phases, compliance standards, artifact traceability, CI/CD processes, training workflows, NFA test catalogs, troubleshooting runbooks, dynamic configuration templates, release policies, input/output taxonomy, developer abstract classes, observability metrics, security hardening methods, and sample templates/checklists. All content is fully expanded with paragraphs and examples for each section as required.