Deploying Agentic AI Applications with OPEA on Intel®

Ezequiel Lanza

Open Source AI Evangelist,

LF AI & Data Chair/Board

@ Intel

Kelli Belcher

AI Software Solutions Engineer

@ Intel

Agenda

- Introduction to AI Agents
- Overview of OPEA
- OPEA AgentQnA Blueprint
- Workshop
  - Build and Deploy AgentQnA on Kubernetes
  - Modify your RAG database to change agent behaviour
  - Add a web search agent to the architecture

What is an agent?

Agents not only generate text – they use tools, delegate tasks to workers, and execute complex workflows based on rules or learning models.

- **Tools:** Interfaces for querying APIs, triggering functions, or transforming data (e.g., search, calculator, database access).
- **Workers:** Specialized sub-agents or microservices that retrieve, analyze, or transform data for specific tasks.
- **Agents:** Decision-makers that orchestrate tools and workers to achieve a goal. They follow predefined rules or learned policies.

🔑 Key Features of AI Agents

- **Autonomous:** Operate with minimal human intervention.
- **Perceptual:** Analyze information from text, voice, images, or structured data.
- **Goal-oriented:** Work to accomplish tasks such as answering questions or summarizing content.
- **Interactive:** Communicate with humans or with other agents.
- **Adaptive:** Improve over time through learning or feedback.

Types of Agents

Agents can be classified based on their reasoning and decision-making capabilities:

- **Reactive Agents** Respond only to current inputs, without memory.
- **Goal-Based Agents** Use goals to guide their decisions.
- **Utility-Based Agents** Optimize for the best possible decision.
- **Learning Agents** Improve with data and experience.

Agents in action:

🚗 Reactive Agent: A self-driving car stops when the traffic light is red, but doesn't plan ahead.

🛣️ Goal-Based Agent: The car not only stops but also adjusts its speed to optimize fuel efficiency.

The Power of AI Agents

- Large language models are excellent at generating natural language — but real-world applications demand more than just fluent responses.
- To truly assist users, AI systems must:
  - Use tools and APIs to take action
  - Retain and reason across multiple steps
  - Query databases or retrieve relevant documents
  - Operate reliably in production environments
- *The Challenge:* Building these capabilities from scratch is complex. Most frameworks either oversimplify agent logic or lock you into rigid architectures that are hard to scale or customize.
- OPEA gives you the tools to go beyond chat and build systems that can reason, act, and specialize.

Open Platform for Enterprise AI (OPEA)

Open source platform that organizes Gen AI chaos

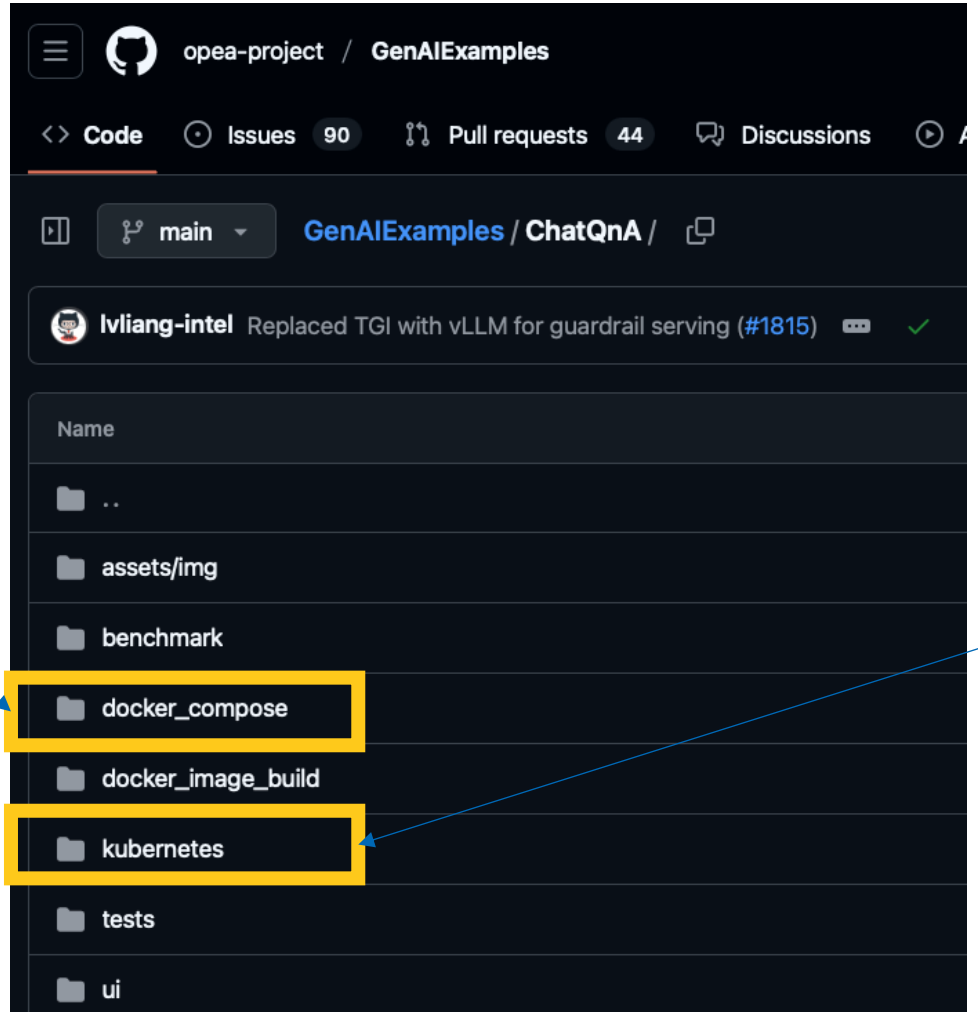
THE LINUX FOUNDATION

Open Platform for Enterprise AI

- Composable building blocks for generative AI systems
- Integrates LLMs, data stores, and prompt engines
  RAG AI blueprint with component stack structure and end-to-end workflow
- Additional stacks for translation, code generation, images
- Generative AI assessment tool for performance, features, trustworthiness and enterprise-grade readiness

Deployment Options

Docker compose: VM deployment



Helm Charts: Kubernetes deployment

🔍 Built for Real Applications

- OPEA bridges the gap between research prototypes and production-ready deployments:
  1. Vector Database Integration
  2. Stateful Workflows
  3. Flexible Tool Integration
  4. Production Observability
  5. Scalable Architecture

Features	What It Enables
Agentic Blueprints	Skip boilerplate — use prebuilt, customizable agent workflows (RAG, tool use, multi-agent)
Modular by Design	Swap in any LLM, memory backend, or external tool
Deployment-Ready	Includes Docker, APIs, tracing, and monitoring
Enterprise-Friendly	Secure, observable, and compatible with on-prem or cloud deployments
Community-Driven	Actively maintained by contributors from Intel and the open-source ecosystem

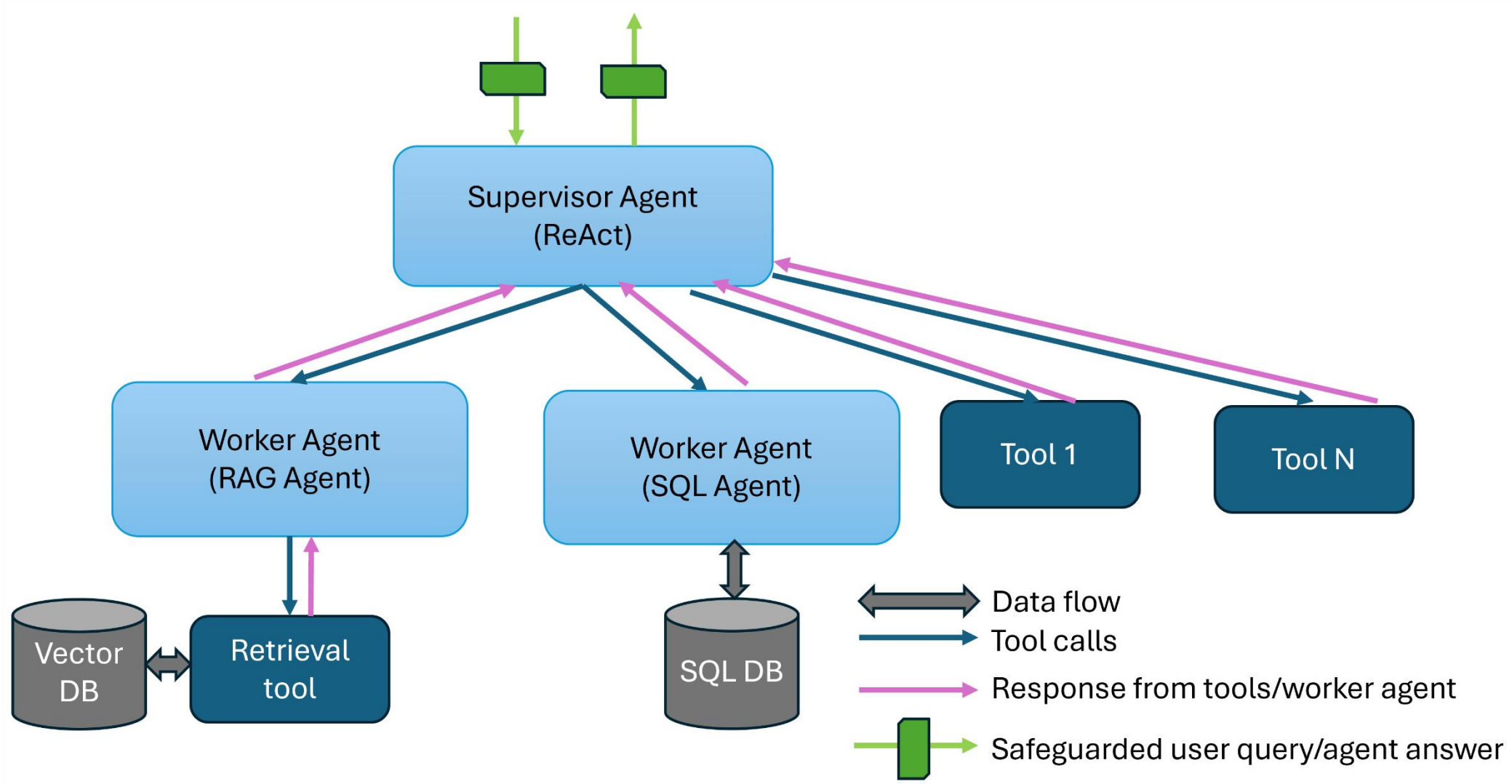
What is AgentQnA

- An OPEA blueprint that demonstrates an agentic question-and-answering system.
- It follows a hierarchical multi-agent architecture with specialized roles.
- Key Components:
  - **Supervisor Agent:** Analyzes incoming questions and routes them to appropriate specialized agents.
  - **RAG Agent:** Handles knowledge-based questions by retrieving relevant documents from vector databases.
  - **SQL Agent:** Converts natural language to SQL queries for structured data retrieval.
  - **Tool integration:** Agents can use external tools like web search, calculators, APIs, and more.

Why should AI Agents be used for question-answering

1. **Improve relevancy of retrieved context:** Compared to conventional RAG, RAG agents significantly improve the correctness and relevancy of the answer because of the iterations it goes through.
2. **Expand scope of skills:** The supervisor agent interacts with multiple worker agents that specialize in different skills (e.g., retrieve documents, write SQL queries, etc.). Thus, it can answer questions with different methods.
3. **Hierarchical multi-agents improve performance:** Expert worker agents, such as RAG agents and SQL agents, can provide high-quality output for different aspects of a complex query, and the supervisor agent can aggregate the information to provide a comprehensive answer.

OPEA AgentQnA Architecture



What are you building?

Notebook 1: Learn AgentQnA and Deploy the blueprint on a Kubernetes cluster on Intel® Tiber AI Cloud.

Notebook 2: Modify your RAG database to change agent behaviour.

Notebook 3: Add a web search agent to the architecture.

By the end of this tutorial, you'll be able to deploy production-ready AI agents that can:

1. **Think:** Use advanced reasoning capabilities with LLMs.
2. **Search:** Query knowledge bases intelligently using Retrieval-Augmented Generation (RAG).
3. **Act:** Execute real-world tasks using external tools and APIs.
4. **Scale:** Run reliably and efficiently on Kubernetes infrastructure.

Prerequisites

1. Jupyter Notebook
2. Kubernetes Cluster Access
3. Command-Line Tools
   - helm
   - kubectl
   - Docker
   - curl and jq
4. API Key based on LLM source
5. Network

DENVR dataworks

intel GAUDI

AI Applications and Services

Samples | AI Apps and Functions

OpenAI API's

Models and Monitoring

LLM Gateway, Key Management

Models

Grafana | Prometheus

Inferencing Engine and Authorization

vLLM Inferencing Engine

API Gateway APISIX | Security Keycloak

Orchestration Layer

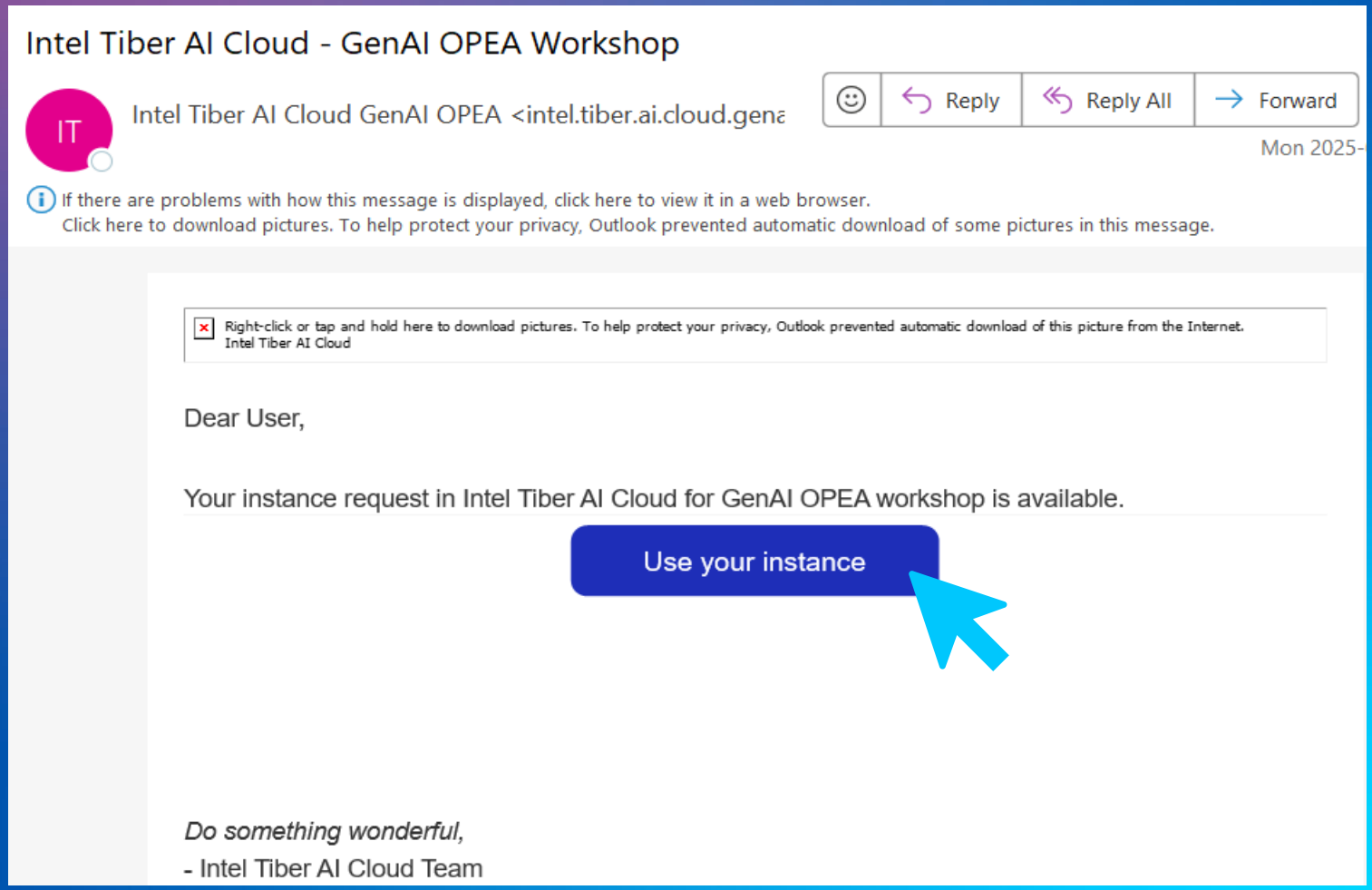
Kubernetes Orchestrator | Helm Charts

Infrastructure Core Components

Operating System Ubuntu 22.04/ RHEL / Fedora | Gaudi SW (Operators,Firmware and Drivers) | XEON Drivers and Firmware

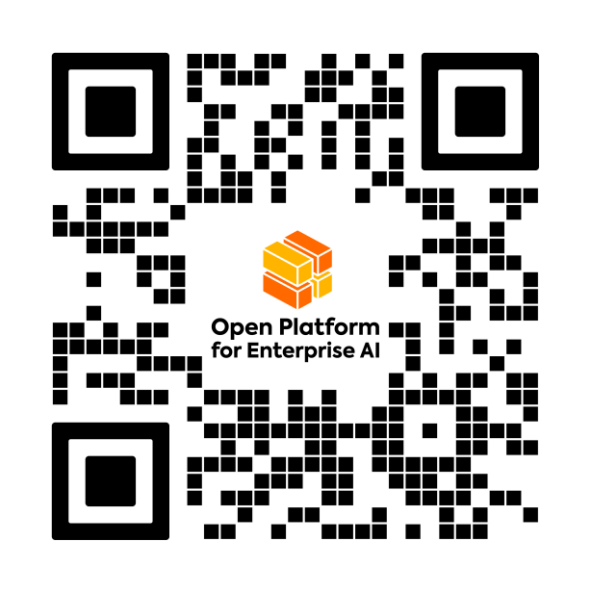
Intel® AI for Enterprise Inference

Look for GenAI OPEA Workshop email from Intel



Workshop!

Participate in OPEA



Visit OPEA.dev

- Join a Working Group
- Bring Enterprise AI use cases
- Contribute code, docs, projects, blueprints, and more
- Provide feedback
- Evangelism and Promotion of OPEA in YOUR communities, events

Open Platform for Enterprise AI

Thank you!

Notices and disclaimers

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Intel® technologies may require enabled hardware, software, or service activation. No product or component can be absolutely secure. Your costs and results may vary.