

Qdrant Problem Statement - Convoke

4.0 - A Pan-IIT AI/ML Hackathon

Search, Memory, and Recommendations for Societal Impact

01. Motivation

Modern AI systems increasingly interact with multimodal data - text, code, images, audio, video, and structured signals. Societal challenges such as healthcare access, climate resilience, education equity, and public safety require systems that can reliably search, remember, and reason over long-term, evolving Knowledge.

Societal challenges - such as healthcare access, climate resilience, public safety, education equity, misinformation detection, and accessibility - require systems that can:

- **Search** across massive, heterogeneous datasets
- **Maintain long-term memory** across time and interactions
- **Recommend** context-aware, personalized, and responsible actions

This challenge invites participants to design **AI agents or applications powered by Qdrant**, a high-performance vector search engine, to enable **reliable retrieval, memory, and reasoning over multimodal data** - not as a demo, but as a system addressing a **real-world societal problem**.

02. About Qdrant

Qdrant is an open-source, production-grade **vector search engine** designed for high-performance similarity search, filtering, and scalable AI workloads.

Qdrant enables:

- Dense & sparse vector search
- Hybrid search (semantic + keyword + metadata filtering)
- Multimodal embeddings (text, image, audio, video, code)
- Real-time updates and low-latency retrieval
- Strong consistency and horizontal scalability

Qdrant is used globally to power:

- AI agents and copilots
- Retrieval-Augmented Generation (RAG)
- Long-term memory systems
- Recommendation engines
- Multimodal search platforms

Learn more: <https://qdrant.tech>

03. The Challenge

Build an **AI agent or application** that uses **Qdrant as a core component** to power one or more of the following capabilities:

- **Search** – semantic or hybrid retrieval over multimodal data
- **Memory** – persistent, long-term, or evolving knowledge storage
- **Recommendations** – context-aware suggestions or decision support

The system **must address a societal challenge**, such as (but not limited to):

- Healthcare & mental health support
- Climate & environmental monitoring
- Disaster response & public safety
- Education & accessibility
- Civic tech & governance
- Misinformation & digital trust
- Inclusion, assistive technologies, or equity

04. System Expectations

Your solution should clearly demonstrate:

1. Effective Multimodal Retrieval

- Storing and querying **non-text data** (images, audio, video, code, sensor data, etc.)
- Correct use of vector embeddings and similarity search
- Meaningful metadata filtering and payload design

2. Memory Beyond a Single Prompt

- Long-term or session-based memory
- Evolving representations (updates, deletions, decay, reinforcement)
- Clear distinction between **knowledge**, **context**, and **interaction history**

3. Societal Relevance & Responsibility

- A clearly defined real-world problem
- Thoughtful handling of bias, safety, privacy, or explainability
- Realistic deployment assumptions

4. Evidence-Based Outputs

- Outputs grounded in retrieved data

- Traceable reasoning paths (what was retrieved, why, and how it influenced decisions)
- Avoiding hallucinated or ungrounded responses

05. Example Directions

Participants may explore ideas such as:

- A **multimodal disaster-response agent** retrieving satellite images, emergency audio logs, and text reports
- A **healthcare memory assistant** tracking patient history, reports, and medical images over time
- An **education accessibility system** recommending personalized learning resources using text, video, and interaction data
- A **misinformation detection engine** correlating claims across articles, images, and social media snippets

These are **examples, not constraints**.

06. Technical Requirements

Mandatory

- **Qdrant must be used as the primary vector search engine**
- Vectors must represent **meaningful semantic or multimodal signals**
- Demonstrate **search, memory, or recommendation** as a system capability (not just storage)

Allowed & Encouraged

- Any embedding models (open-source or hosted)
- Any LLMs or agent frameworks
- Hybrid architectures (symbolic + neural)
- Streaming or batch ingestion
- Re-ranking, filtering, or feedback loops

Not Required

- Training large models from scratch
- Proprietary datasets
- UI-heavy implementations (CLI or API is acceptable)

08. Documentation Requirements

Your submission **must include documentation** that explains:

1. Problem Statement

- What societal issue are you addressing?
- Why does it matter?

2. System Design

- Architecture overview
- Why Qdrant is critical to your solution

3. Multimodal Strategy

- What data types are used
- How embeddings are created and queried

4. Search / Memory / Recommendation Logic

- How retrieval works
- How memory is stored, updated, and reused

5. Limitations & Ethics

- Known failure modes
- Bias, privacy, or safety considerations

09. Deliverables

1. Code (Reproducible)

- End-to-end runnable
- Clear setup instructions

2. Documentation / Report

- Maximum 10 pages (excluding appendix)
- Architecture diagrams encouraged

3. Demo or Examples

- Sample queries, outputs, or interaction logs
- Screenshots or short recordings optional

10. Evaluation Criteria

Submissions will be evaluated on:

- Correct and meaningful use of Qdrant
- Quality of retrieval and memory design
- Societal relevance and impact
- System clarity and robustness
- Thoughtful documentation and reasoning
- Creativity without sacrificing correctness

10. Resources

- [Qdrant Overview](#)
- [Qdrant GitHub](#)
- [Qdrant Articles](#)
- [Qdrant Demos and Tutorials](#)