



VISUAL INCIDENT INTELLIGENCE

SKETCH & SEARCH HACKATHON

PRESENTED BY: GOUTHAMI NADUPURI
*GITHUB: [HTTPS://GITHUB.COM/GOUTHAMIN25/VISUAL-INCIDENT-
INTELLIGENCE](https://github.com/gouthamin25/visual-incident-intelligence)*

VISUAL INCIDENT INTELLIGENCE

- During real-world incidents, engineers and security teams rely heavily on visual evidence — monitoring dashboards, network sketches, and architecture diagrams. However, these visuals are usually treated as static images and cannot be searched, compared, or reused effectively.
- Visual Incident Intelligence changes that. It uses multimodal AI to understand what an image represents and vector search to find similar past incidents, enabling faster investigation, better decision-making, and reduced resolution time.

PROBLEM

The Problem

- During incidents, teams share:
 - Dashboard screenshots
 - Network sketches
 - Architecture diagrams
- These visuals are:
 - Not searchable
 - Not reusable
 - Hard to compare with past incidents

Result

- Slower triage
- Repeated investigations
- Higher MTTR

WHY EXISTING TOOLS FALL SHORT

Current Systems-

- Log-based search
- Keyword-driven tickets
- Text-only AI tools

Missing Capability-

- No understanding of visual meaning
- No semantic search over images
- No reuse of visual incident knowledge

SOLUTION OVERVIEW

- Visual Incident Intelligence
- Upload an incident-related image
- Multimodal AI understands what the image represents
- Vector search finds similar past incidents
- System explains:
 - Why it matched
 - What actions to take next

HOW IT WORKS (ARCHITECTURE)

- End-to-End Flow
- Image upload (sketch / screenshot / diagram)
- Gemini Vision extracts incident signals
- Signals → embeddings
- Stored in Qdrant
- Semantic search retrieves similar incidents
- Explainable results + actions

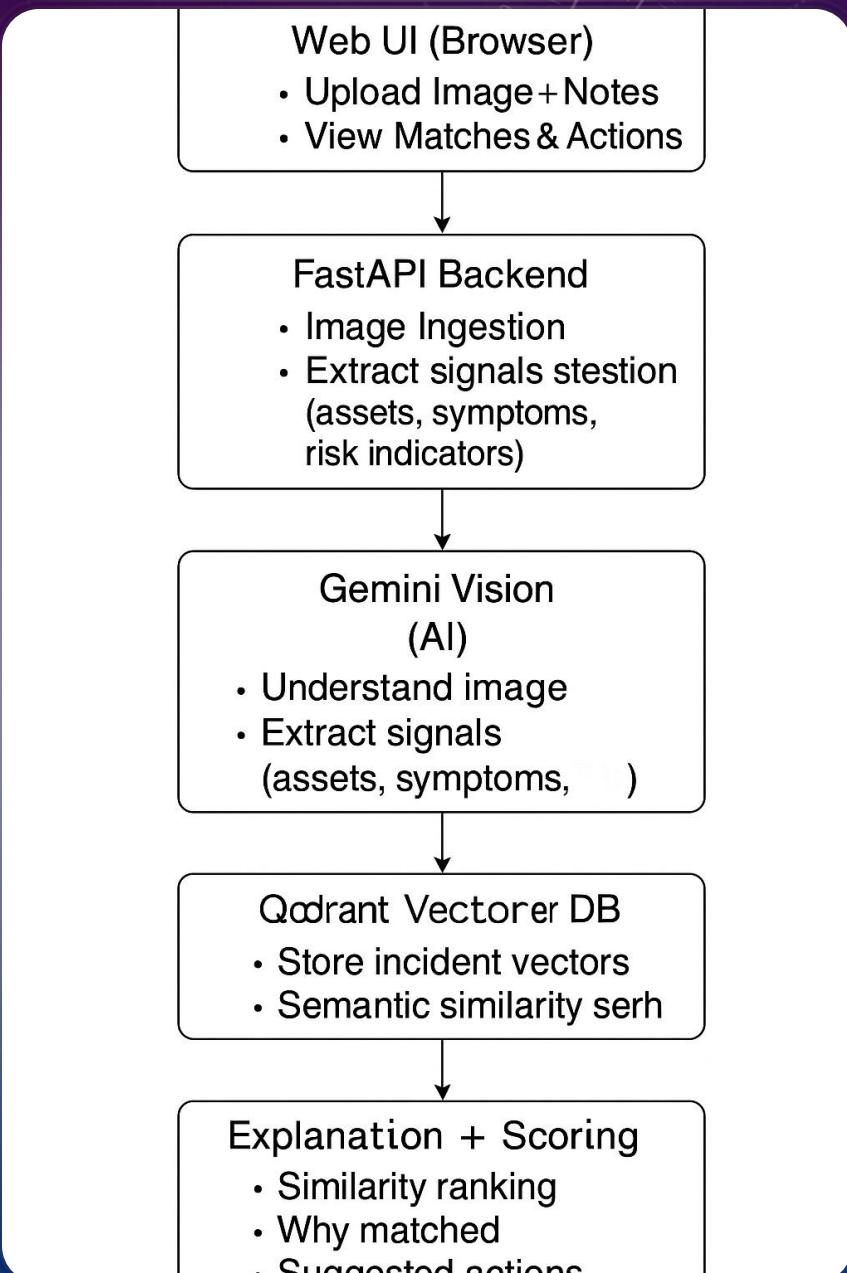
VISUAL INCIDENT INTELLIGENCE – END-TO-END ARCHITECTURE

A user uploads an incident-related image through the web UI.

The backend sends it to Gemini Vision, which understands what the image represents — dashboards, network paths, or architecture components.

That meaning is converted into vector embeddings and stored in Qdrant.

When a new incident arrives, we perform semantic vector search to retrieve similar past incidents and return explainable matches and suggested actions.



SKETCH & SEARCH ALIGNMENT

Alignment with Hackathon Theme-

- Sketch
- Visual inputs: screenshots, sketches, diagrams

Search-

- Semantic vector search using Qdrant
- Search by meaning, not keywords

Key Insights-

- We search incidents based on what the image means.

TECH STACK

Backend-

- FastAPI
- Python 3.11
- Pydantic

Multimodal AI-

- Google Gemini Vision (via Google AI Studio)
- Image + text understanding

Vector Search-

- Qdrant
- Semantic similarity search
- Cosine distance embeddings

Embeddings-

- Sentence Transformers (all-MiniLM-L6-v2)

Frontend-

- HTML
- CSS
- JavaScript

Development & Debugging-

- Antigravity (IDE + agent-assisted workflows)

ENGINEERING & PLATFORM IMPLEMENTATION

- Developed backend services using FastAPI
- Implemented Qdrant vector collections for semantic incident search
- Integrated Gemini Vision (Google AI) for multimodal understanding
- Built embedding pipeline using Sentence Transformers
- Debugged and validated API flows using Antigravity (IDE + agent tools)
- Developed and validated the system using both local terminal workflows and Google Antigravity IDE

The screenshot shows the Antigravity IDE interface. The code editor window displays Python code for interacting with a Qdrant database:

```
import os
from qdrant_client import QdrantClient
from qdrant_client.http import models as qm

def get_qdrant() -> QdrantClient:
    url = os.getenv("QDRANT_URL", "http://localhost:6333")
    api_key = os.getenv("QDRANT_API_KEY") or None
    return QdrantClient(url=url, api_key=api_key)

def ensure_collection(client: QdrantClient, name: str, dim: int):
    if not client.collection_exists(name):
        client.create_collection(
            collection_name=name,
            vectors_config=qm.VectorParams(size=dim, distance=qm.Distance.COSINE),
        )
```

The terminal window shows a command-line interface with the text "H1 for Command, H2L for Agent". The status bar at the bottom indicates "Ln 16, Col 1" and "Antigravity - Settings".

visual-incident-intel — qdrant_store.py · incident-intel — requirements.txt



BUILT & DEBUGGED USING GOOGLE ANTIGRAVITY

Used Antigravity IDE for:

- Code navigation and debugging
- API integration validation
- Frontend–backend alignment
- Accelerated development with agent-assisted workflows
- Ensured production-ready structure

LIVE DEMO WALKTHROUGH

```
Terminal Shell Edit View Window Help

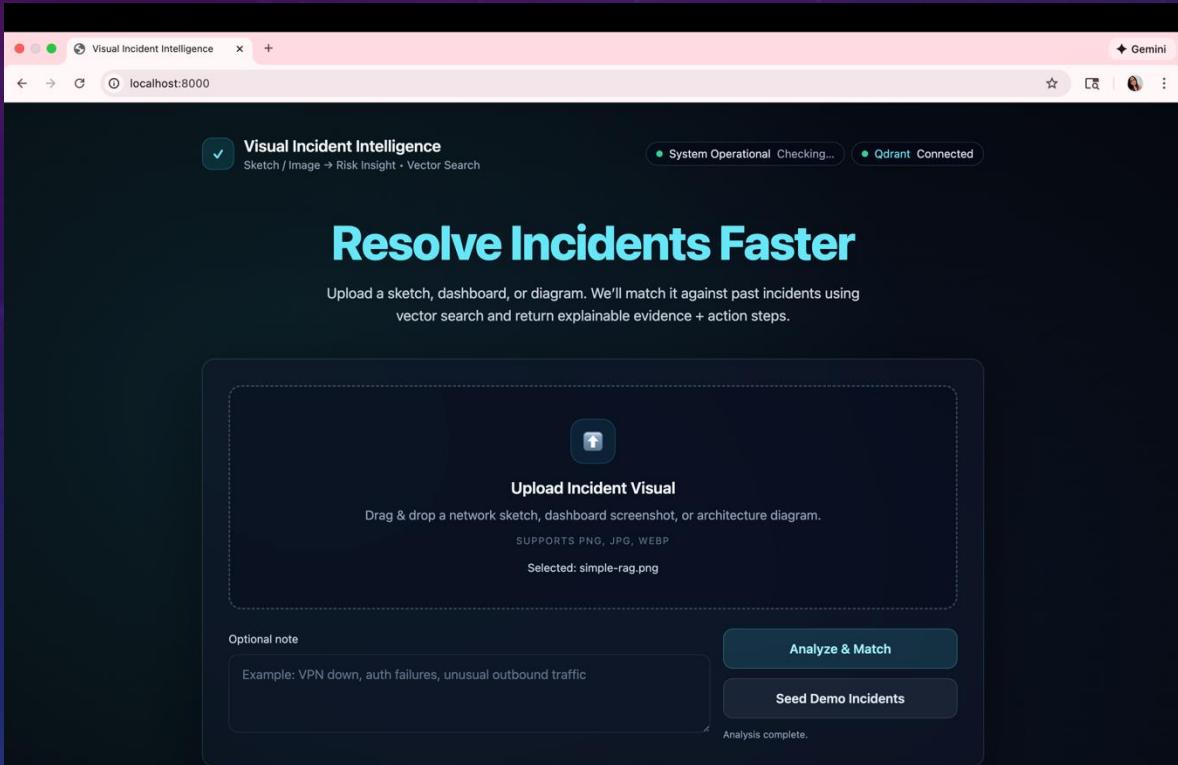
visual-incident-intel — qdrant - 103x43
Last login: Fri Dec 12 17:52:34 on ttys016
(base) gouthami@Gouthamis-MacBook-Air ~ % cd visual-incident-intel
(base) gouthami@Gouthamis-MacBook-Air visual-incident-intel % /Users/gouthami/qdrant
<jemalloc>: option background_thread currently supports pthread only
[Output truncated]

Version: 1.16.2, build: d2834de0
Access web UI at http://localhost:6333/dashboard

2025-12-13T22:43:06.958597Z  WARN qdrant::settings: Config file not found: config/config
2025-12-13T22:43:06.962248Z  WARN qdrant::settings: Config file not found: config/development
2025-12-13T22:43:06.969863Z  INFO storage::content_manager::consensus::persistent: Loading raft state f
rom ./storage/raft_state.json
2025-12-13T22:43:06.998987Z  INFO qdrant: Distributed mode disabled
2025-12-13T22:43:07.008176Z  INFO qdrant: Telemetry reporting enabled, id: 80681b5a-5fde-486e-b8d2-013e
038db0ff
2025-12-13T22:43:07.282586Z  WARN qdrant::actix::web_ui: Static content folder for Web UI './static' do
es not exist
2025-12-13T22:43:07.282792Z  INFO qdrant::actix: TLS disabled for REST API
2025-12-13T22:43:07.286539Z  INFO qdrant::actix: Qdrant HTTP listening on 6333
2025-12-13T22:43:07.288483Z  INFO actix_server::builder: starting 7 workers
2025-12-13T22:43:07.291827Z  INFO actix_server::server: Actix runtime found; starting in Actix runtime
33", workers: 7, listening on 0.0.0.0:6333
2025-12-13T22:43:07.333969Z  INFO qdrant::tonic: Qdrant gRPC listening on 6334
2025-12-13T22:43:07.333984Z  INFO qdrant::tonic: TLS disabled for gRPC API
2025-12-13T22:43:24.356584Z  INFO actix_web::middleware::logger: 127.0.0.1 "GET /dashboard HTTP/1.1" 40
4 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
143.0.0.0 Safari/537.36" 0.018592
2025-12-13T22:43:33.430943Z  INFO actix_web::middleware::logger: 127.0.0.1 "GET /collections HTTP/1.1"
200 65 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chro
me/143.0.0.0 Safari/537.36" 0.024814
2025-12-13T22:47:36.357103Z  INFO actix_web::middleware::logger: 127.0.0.1 "GET /collections/incidents_
semantic/exists HTTP/1.1" 200 83 "-" "python-client/1.16.2 python/3.11.4" 0.028181
2025-12-13T22:47:36.421126Z  INFO storage::content_manager::toc::collection_meta_ops: Creating collecti
on incidents_semantic
2025-12-13T22:47:38.437589Z  INFO actix_web::middleware::logger: 127.0.0.1 "PUT /collections/incidents_
semantic HTTP/1.1" 200 71 "-" "python-client/1.16.2 python/3.11.4" 2.051358
[Output truncated]

visual-incident-intel — zsh - 121x50
>Last login: Sun Dec 14 12:09 AM
(base) gouthami@Gouthamis-MacBook-Air visual-incident-intel % /Users/gouthami/anaconda3/lib/python3.11/multiproces
sing/resource_tracker.py:224: UserWarning: resource_tracker: There appear to be 1 leaked semaphore objects to clean up at
shutdown
  warnings.warn('resource_tracker: There appear to be %d'
[47627]
[Output truncated]
```

PROJECT IMAGE



USE CASES & IMPACT

Who Is This For?

- Security Operations (SOC)
- SRE / DevOps teams
- Cloud infrastructure teams
- Enterprise IT operations

Impact

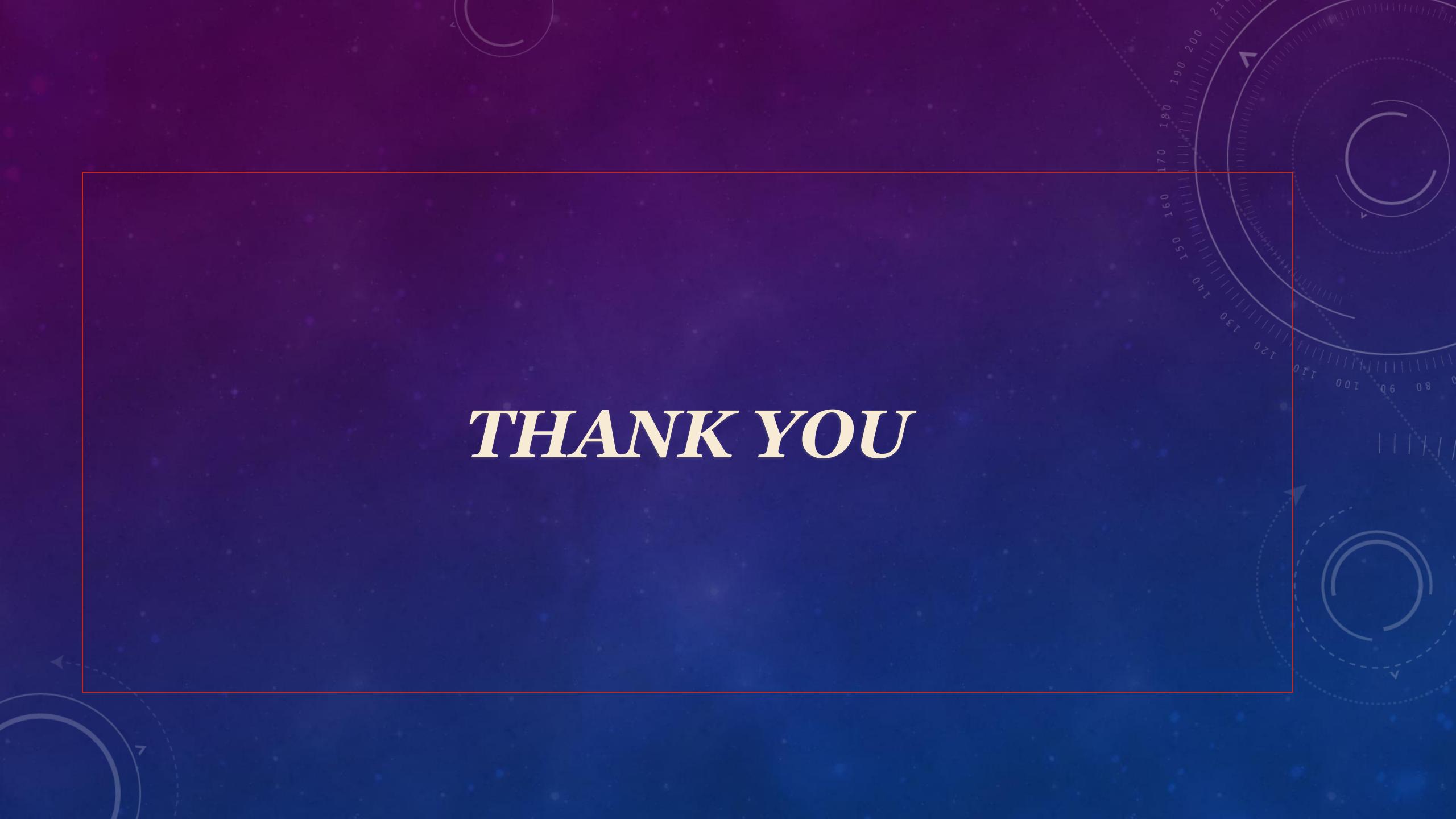
- Faster incident triage
- Knowledge reuse
- Reduced MTTR
- Better decisions under pressure

WHY THIS PROJECT STANDS

- Real-world incident response problem
- Multimodal AI (image + text understanding)
- Vector search as the core retrieval mechanism
- Explainable similarity & confidence scoring
- Strong Sketch & Search alignment
- Transforms static visuals into searchable knowledge
- Reduces mean time to resolution (MTTR)

FINAL TAKEAWAY

- Visual Incident Intelligence turns screenshots and sketches into reusable, searchable incident knowledge
- Enables faster incident triage through semantic similarity search
- Demonstrates the real power of Sketch & Search beyond text



THANK YOU