Based on the research into the **December 13, 2025, Barus & Holley shooting** and the requirements for the **ELI Campus Guardian** platform, here is the comprehensive design and architecture specification.

1. Research & Incident Context

- **The Incident:** On December 13, 2025, an active shooter entered the **Barus & Holley** engineering building. The suspect was "casing" the building hours prior, visible on neighborhood cameras but undetected by campus security due to lack of correlation.
- **Security Failure Points:**
  - **Passive vs. Active:** 1,200 cameras were recording to disk (forensic) rather than being analyzed in real-time.
  - **Data Silos:** The suspect entered an unlocked side door. Wi-Fi probes likely detected his device, but this data was not linked to the camera feeds or door access logs.
  - **Blind Spots:** Police struggled to locate the shooter in the complex layout of the engineering wing.
- **ELI Solution:** A "Single Pane of Glass" that correlates **Cisco Wi-Fi location**, **IREX Weapon Detection**, and **Access Control** to track threats in real-time on a 2D gamified map.

---

2. System Architecture

We will use an **Event-Driven Microservices Architecture** to handle high-frequency location updates from 11,000+ students.

High-Level Diagram

Code snippet

```
graph TD
    subgraph "Edge / Sensors"
        Cam[IREX Smart Cameras] -->|RTSP + Webhooks| Ingest
        Wifi[Cisco Catalyst Center] -->|Firehose API (JSON)| Ingest
        Door[Access Control (Lenel)] -->|Syslog| Ingest
    end

    subgraph "ELI Core Platform"
        Ingest[Ingestion Service (FastAPI)] -->|Normalized Events| Correlation[TruContext Correlation Engine]
        Correlation -->|Write| DB[(PostgreSQL + PostGIS)]
```

```
    Correlation -->|Publish| Socket[WebSocket Server]

    Ethic[Ethical Governor] -.->|Privacy Masking| Socket
  end

  subgraph "Frontend (React/Next.js)"
    Socket -->|Real-time Stream| Dashboard
    Dashboard -->|Render| Map[2D 'Pokemon' Live Map]
    Dashboard -->|Stream| Video[Smart Camera Grid]
  end
```

---

3. Database Schema (PostgreSQL + PostGIS)

We will use **PostGIS** for spatial queries (essential for "Find cameras near this person") and **TimescaleDB** (if available, or partitioned tables) for the high-volume location history.

SQL

```sql
-- 1. Configuration & Auth
CREATE TABLE config (
  key VARCHAR(50) PRIMARY KEY,
  value TEXT -- e.g., 'DEMO_MODE' = 'TRUE'
);

-- 2. Infrastructure (Maps)
CREATE TABLE floor_plans (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100), -- 'Barus & Holley - Floor 1'
  image_url TEXT,    -- Output from Google Nano Banana
  bounds GEOMETRY(POLYGON), -- GPS coordinates of building corners
  pixels_per_meter FLOAT
);

-- 3. Device Inventory
CREATE TABLE devices (
  id UUID PRIMARY KEY,
  type VARCHAR(20), -- 'CAMERA', 'WIFI_AP', 'DOOR_READER'
  external_id VARCHAR(100),
  location GEOMETRY(POINT), -- Location on the map
```

```
    stream_url TEXT
);

-- 4. Active Tracking (The "Dots")
CREATE TABLE tracked_entities (
    id UUID PRIMARY KEY,
    type VARCHAR(20), -- 'STUDENT', 'FACULTY', 'THREAT'
    display_name VARCHAR(100),
    cisco_mac_hash VARCHAR(64),
    irex_face_id VARCHAR(64),
    current_location GEOMETRY(POINT),
    risk_score INT DEFAULT 0,
    last_updated TIMESTAMP DEFAULT NOW()
);

-- 5. History Trails (For the 30-min Rewind)
CREATE TABLE location_history (
    time TIMESTAMP NOT NULL,
    entity_id UUID REFERENCES tracked_entities(id),
    location GEOMETRY(POINT),
    source VARCHAR(20) -- 'WIFI', 'CAMERA_ESTIMATE'
);
-- Create hypertable for performance (if TimescaleDB installed)
-- SELECT create_hypertable('location_history', 'time');

-- 6. Alerts
CREATE TABLE alerts (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMP DEFAULT NOW(),
    type VARCHAR(50), -- 'WEAPON_DETECTED', 'TAILGATING'
    severity INT, -- 1 (Low) to 5 (Critical)
    metadata JSONB -- { "snapshot": "url", "confidence": 0.98 }
);
```

---

4. Key Features & Component Structure

A. Real-Time Location Tracking ("The Pokémon Map")

- **Component:** LiveTacticalMap.tsx
- **Library: Leaflet.js** (lightweight) or **Pixi.js** (if handling >1000 moving dots for performance).
- **Visuals:**

    - **Floor Plan:** Generated via **Google Nano Banana** as requested, overlaid as an L.imageOverlay.
    - **Avatars:**
        - 🔵 **Blue Dot:** Anonymous Student (Wi-Fi).
        - 🟢 **Green Dot:** Faculty/Staff (Badge Match).
        - 🔴 **Red Pulse:** High Risk/Threat.
- **History Slider:** A React range slider (rc-slider) at the bottom.
    - **Logic:** When dragged to "-10m", the WebSocket creates a request to GET /api/history?lookback=600. The frontend renders the path as a fading polyline.

B. Camera Monitoring Dashboard

- **Component:** SmartCameraGrid.tsx
- **Logic:** "Zero-Click" interface. The grid is empty or shows main entrances by default. When an alert triggers, the relevant camera **automatically** populates the main view.

C. Integration with ELI (Peru Project)

- **Reuse:** Adopt the **"Dark Mode" UI Kit** (Navy Blue backgrounds, Neon Cyan accents) and the **RTSP Video Player** component (hls.js wrapper) from the Peru codebase.

---

5. API Endpoint Specifications

Base URL: /api/v1

Endpoint	Method	Description	Payload Example
/ingest/irex	POST	Webhook for IREX Object Detection	{ "class": "weapon", "confidence": 0.95, "cam_id": "BH-104" }
/ingest/cisco	POST	Firehose for Wi-Fi Location (X,Y)	{ "mac": "a1:b2...", "x": 105.4, "y": 33.2, "floor": "BH_L1" }
/map/entities	WS	**WebSocket** for live dot movement	[{ "id": "123", "lat": ..., "lng": ..., "type": "student" }]
/map/history	GET	Retrieve trail coordinates	?entity_id=123&start_time=...

/alerts	GET	Fetch active alerts list	?active=true
/demo/trigger	POST	**Simulation Only:** Starts the "Active Shooter" script	{ "scenario": "barus_holley_incident" }

---

6. Integration Approach

1. **Cisco Catalyst (Wi-Fi Tracking):**
   - **Constraint:** Standard keycards are invisible to Wi-Fi.
   - **Solution:** We rely on **Device Fingerprinting**. We assume most students carry a phone. The system correlates "Door Badge Swipe at Room 101" with "New Wi-Fi Connection in Room 101" to link a specific phone MAC address to a student Identity.
   - **Demo Mode:** Since we don't have live Cisco access, we will write a Python script (mock_cisco.py) that generates 50 "boids" (moving dots) that wander the hallways of the Barus & Holley floor plan.
2. **IREX Cameras:**
   - **Demo Mode:** Use pre-recorded MP4 clips of the Brown campus. Overlay bounding boxes using HTML5 Canvas (<canvas>) drawn over the video element to simulate live AI detection.
3. **Access Control:**
   - **Simulation:** A background script will generate random "Door Open" events. If a "Blue Dot" (Phone) is not near the door when it opens, the system triggers a **"Forced Entry / Tailgating"** alert.

---

7. Implementation Roadmap (90-Day Plan)

- **Phase 1: Barus & Holley Pilot (Days 1-30)**
  - **Goal:** Recreate the December 13th incident to show "What could have been done."
  - **Deliverables:**
    - Google Nano Banana floor plans for Barus & Holley.
    - "Pokemon" Map View with simulated students.
    - Scripted Scenario: Suspect enters, Weapon Detected by IREX, System Auto-Locks doors (simulated).
- **Phase 2: Residential Expansion (Days 31-60)**
  - **Goal:** Keeney Quad & Dorms.
  - **Deliverables:**
    - Integrate "Crowd Surge" alerts (e.g., too many people in a hallway at 2 AM).
    - Deploy the "History Trail" feature for analyzing foot traffic.
- **Phase 3: Campus-Wide & Handover (Days 61-90)**

    - **Goal:** Full deployment.
    - **Deliverables:**
        - Load test with 800 mock camera streams.
        - Final UI Polish (Visium Branding, Dark Mode).
        - Documentation & Sales Video generation.

8. Demo Login Credentials

- **URL:** localhost:3000 (or demo deployment URL)
- **Username:** admin
- **Password:** admin