

# Brown University Campus Security Monitoring Platform - Design & Architecture

This design incorporates insights from the recent December 13, 2025, mass shooting at Brown University, where a gunman killed two people and injured others during finals week in the Barus & Holley engineering building. Based on web research, the incident highlighted critical gaps in traditional surveillance: despite ~1,200 cameras, blind spots in older structures led to reliance on community-submitted footage, delayed response times (over 3 hours for full searches), and manual investigations. Experts recommend real-time AI-driven threat detection, data fusion from multiple sensors, predictive alerts, and ethical governance to prevent escalation—lessons echoed in similar incidents like the 2022 Uvalde school shooting (delayed response due to poor integration) and 2018 Parkland (surveillance failures in coverage). An effective system could have provided sub-second alerts, tracked the shooter via fused Wi-Fi/motion data, and automated lockdowns, potentially reducing response time by 30-40% as seen in post-incident analyses at peer institutions.

Drawing from the ELI Peru project (a unified surveillance dashboard with real-time mapping, graph analytics, and AI integrations) and Visium's TruContext Platform (real-time event correlation, graph-based fusion of cyber/physical data, and predictive AI), this demo system reuses patterns like webhook ingestion, topology graphs, and AI-driven incident management. It emphasizes ethical AI under the IREX/ELI framework, integrating ~800 CCTV cameras with access controls, motion sensors, and Wi-Fi tracking for proactive campus security.

## 1. System Architecture Diagram

(Text-based representation; in a real implementation, use tools like Draw.io or Lucidchart for visuals.)

text

[Frontend: React App (Vite + Tailwind + Framer Motion)]

- Landing Page (Public: Videos, PPTs, Docs)
- Auth Layer (Login: admin/admin)
- Dashboards: Real-Time Location Tracking, Camera Monitoring, Alerts & Actions (Lazy-loaded components, Context Menus for actions)

|

[Backend: Express + tRPC API]

- Authentication (JWT via Jose)
- Routers: Events, Incidents, Users, Config
- Webhook Endpoint (for IREX cameras, Cisco Wi-Fi)
  - | (Real-time processing: Zod validation, Drizzle ORM persistence)

|

[Data Layer]

- PostgreSQL (Primary DB: Events, Alerts, Logs, Mock Data)
  - Relations: Users <-> Sessions; Events <-> Incidents
- Future: Neo4j (Graph: Topology/Relationships); InfluxDB (Time-series metrics)

|

[Integrations]

- IREX Smart Cameras (RTSP/ONVIF streams, Webhooks for events)
- Access Control (RFID/Fobs: API pulls for entry logs)
- Motion Sensors (Webhook/API for triggers)
- Cisco Catalyst Center (Wi-Fi Tracking: REST API for device locations via TruContext)
- AI: Google Gemini (Image metadata, Analytics, Predictions)

|

[External Services]

- Cloudinary (Image/Video Storage & CDN)
- Kafka (Future: High-throughput event streaming from TruContext patterns)

High-level flow: Sensors/cameras send data via webhooks/APIs → Backend processes/fuses (using TruContext-inspired correlation) → Persists to DB → Frontend polls/subscribes for real-time updates. Scalable for demo (mock data) to production (billions of events).

## 2. Database Schema Design

Using PostgreSQL with Drizzle ORM (reused from ELI). Schema focuses on events, incidents, and tracking for demo; mock data seeded via scripts.

SQL

```
-- Users Table (for auth)
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    username VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL, -- Hashed 'admin'
    role ENUM('admin', 'viewer') DEFAULT 'viewer',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Sessions Table
CREATE TABLE sessions (
    id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(id),
    token VARCHAR(512) NOT NULL, -- JWT
    expires_at TIMESTAMP NOT NULL
);

-- Events Table (Core: Fused sensor/camera data)
CREATE TABLE events (
    id SERIAL PRIMARY KEY,
    type ENUM('camera_alert', 'access_entry', 'motion_detect', 'wifi_probe', 'f
    timestamp TIMESTAMP NOT NULL,
    location JSONB, -- {building: 'Barus & Holley', floor: 2, coords: [x,y]}
    details JSONB, -- e.g., {device_mac: 'xx:xx', person_id: 'anon-123', aler
    source VARCHAR(100), -- 'IREX Camera', 'Cisco Wi-Fi', etc.
    incident_id INTEGER REFERENCES incidents(id), -- Link to incident
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Incidents Table (AI-generated or manual)
CREATE TABLE incidents (
    id SERIAL PRIMARY KEY,
    status ENUM('open', 'in_progress', 'resolved') DEFAULT 'open',
    description TEXT,
    priority ENUM('low', 'medium', 'high') NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    assigned_to INTEGER REFERENCES users(id)
);
```

```
-- Config Table (App settings)
CREATE TABLE config (
    id SERIAL PRIMARY KEY,
    key VARCHAR(255) UNIQUE NOT NULL, -- e.g., 'history_trail_duration'
    value JSONB NOT NULL -- e.g., {default: 30, unit: 'minutes'}
);

-- Indexes for performance
CREATE INDEX idx_events_timestamp ON events(timestamp);
CREATE INDEX idx_events_type ON events(type);
```

Mock data: Seed 1,000+ events simulating real-time activity (e.g., via Python script using Faker). Future: Add Neo4j for graph queries (e.g., MATCH (person)-[:MOVED\_TO]->(location)).

### 3. Component/Page Structure

Reused from ELI: React app with lazy loading, Tailwind for dark-themed UI (campus-inspired: brown/red accents), Framer Motion for animations.

- **Landing Page** (Public route): Hero with B-roll video (embed YouTube/Vimeo), sections for presentation videos, PowerPoint embeds (via SlideShare or local iframes), sales docs (PDF links from attachments like "Brown\_ELI\_Presentation\_Final.pptx"), product overview. Login button routes to /login.
- **Login Page**: Simple form (username/password: admin/admin), JWT auth via tRPC.
- **Main Layout** (Protected): Sidebar nav (Dashboards: Location Tracking, Camera Monitoring, Alerts), header with live indicator.
  - **Real-Time Location Tracking Dashboard** (Priority): Interactive 2D floor plans (Leaflet or SVG-based, generated via Google AI tool). Avatars/icons for people (fused from Wi-Fi/RFID/face/phone data). Animations: Movement trails (lines with fade-out, toggle on/off, configurable 30min default via config). History slider (minutes-days). Context menu: Zoom, flag incident.
  - **Camera Monitoring Dashboard**: Grid view (~800 feeds as thumbnails, mock RTSP embeds via HLS). Single-view modal. Alert overlays (red pulsing from events).
  - **Alerts & Activity Dashboard**: Live ticker (3s refresh, pause-on-hover), searchable logs, filters (type, location). Correlation view (graph from events, reused ELI topology).

Components: Reusable from ELI - ContextMenu, EventCard (animated), LiveBadge, FilterPanel (AI-powered like Gemini filters).

#### 4. API Endpoint Specifications

tRPC routers (reused ELI structure). Base: /api/trpc.

- **Auth Router:**
  - `login(input: {username: string, password: string})` : Returns JWT token.
- **Events Router:**
  - `getRecent(query: {limit: number=100, type?: string})` : Returns array of events (sorted by timestamp).
  - `create(input: {type: enum, timestamp: date, location: json, details: json})` : Inserts event (for webhooks).
- **Incidents Router:**
  - `list(query: {status?: enum})` : Returns incidents.
  - `createFromEvent(input: {eventId: number, priority: enum})` : AI-assisted creation (call Google Gemini for description).
  - `update(input: {id: number, status: enum})` : Updates status.
- **Config Router:**
  - `get(key: string)` : Returns value.
  - `set(input: {key: string, value: json})` : Updates (admin only).
- **Webhook Endpoint** (Express POST /webhook): Validates payload (Zod), persists to events, triggers AI analysis (Gemini for metadata/predictions).

Protected via JWT middleware. Error handling: Zod for validation, Sonner toasts on frontend.

## 5. Integration Approach for Each Data Source

- **IREX Smart Cameras** (~800 CCTVs): Webhook for events (e.g., motion/weapon detection), RTSP/ONVIF for streams (mock HLS in demo). Fuse with TruContext patterns: Correlate events in backend for context scoring.
- **Access Control (Door Key Fobs/RFID)**: Poll API (or webhook) for entry logs. Track only if RTLS-enabled (per Cisco notes); map to locations. Demo: Mock events every 10s.
- **Motion Sensors**: Webhook/API pushes for triggers. Fuse: Link to nearest camera/event for incident creation.
- **Wi-Fi Location Services (Cisco Catalyst Center)**: REST API integration via CMX/Spaces for device probes (phones/laptops). Pull locations (x,y on floor maps), update events table. Note: Only Wi-Fi devices; fobs require RFID tags. Demo: Simulate via script generating random probes.
- **Facial Recognition/Phone Signatures**: Embedded in IREX cameras; webhook sends detections. AI (Google Gemini): Extract metadata, predict anomalies.
- General: Use Kafka-inspired queuing (future) for high-throughput; ethical filters (anonymize data per ELI/IREX).

Demo mode: All via mock generators (set in .env: MOCK\_MODE=true).

## 6. UI/UX Mockup Recommendations Based on ELI Project

- **Aesthetic**: Dark "Command Center" theme (glassmorphism from ELI), with Brown University branding (red/brown accents). Mobile-responsive (Tailwind).
- **Location Dashboard**: Pokémon-style 2D view - SVG floor plans with animated dots (Framer Motion paths for trails). Toggle: Checkbox for history (fade lines). Slider for time range.
- **Camera Grid**: Thumbnail carousel with click-to-enlarge (reused ELI image previews).
- **Alerts**: Ticker with hover-pause, searchable table (shadcn/ui DataTable).
- Enhancements: AI filters (e.g., "Show high-risk movements") from ELI Gemini integration. Empty states: "No activity - Campus secure" with icons.
- Accessibility: ARIA labels, keyboard nav (reused ELI patterns).

## 7. Implementation Roadmap

- **Week 1 (Setup):** Clone ELI repo, adapt structure. Set up PostgreSQL/.env, seed mock data. Implement auth/landing page.
- **Week 2 (Core Features):** Build Location Dashboard (integrate mock floor plans, animations). Add events/incidents routers.
- **Week 3 (Integrations & Dashboards):** Mock webhooks for sources. Develop Camera/Alerts dashboards, fuse data.
- **Week 4 (AI & Polish):** Integrate Google Gemini for analytics/predictions. Add config for trails. Test real-time (3s polls).
- **Week 5 (Demo Prep):** Embed attachments (PPTs, PDFs). Performance optimize (lazy load). Deploy to Vercel.
- Post-Demo: Add Neo4j for graphs, Kafka for scaling, full Cisco/IREX APIs.

This design, built under Visium Technologies with IREX/ELI integration, creates an impressive POC demonstrating proactive security—fusing data for real-time insights while upholding ethical standards.