I need your help to create a new ELI application: # Brown University Campus Security Monitoring Platform - Design & Architecture

Project Overview

Design a comprehensive campus security web application for Brown University with approximately 800 CCTV cameras. This is a demo/proof-of-concept system to be presented following recent campus security incidents. The application will integrate multiple security data sources and provide real-time situational awareness.

Research Requirements

1. **Search the web** for information about recent security incidents at Brown University (December 2024) to understand:

- Timeline and details of the incident(s)
- Security system gaps or failures that were identified
- Recommendations from security experts
- Similar incidents at other universities and lessons learned
- What an effective camera/security system could have done differently

2. **Retrieve context from our existing ELI project** (Peru surveillance platform):

- GitHub: IanNoble-Visium/ELI (ELI Dashboard web - ELI Unified Dashboard - Peru Surveillance Platform)
- Identify reusable code, architecture patterns, UI components, and functionality

3. **Retrieve context from Visium Technologies**:

- Review the TruContext Platform on the Visium Technologies website for design patterns and integration approaches

Core Integrations

The platform will correlate data from:

- **Smart Cameras**: IREX smart camera partnership (same technology as Peru ELI project)
- **Access Control**: Door key fobs and RFID card readers
- **Motion Sensors**: Building-wide motion detection
- **Wi-Fi Location Services**: Cisco Catalyst Center integration (or equivalent custom solution)

- Track Wi-Fi-enabled devices (phones, laptops) via probe/association data
- Note: Standard door-access keycards cannot be tracked unless they are RTLS-capable Wi-Fi/RFID tags
- Integration with Cisco Spaces/CMX for real-time location visualization
- **Facial Recognition**: Camera-based identification
- **Phone Signatures**: Device fingerprinting for tracking

Key Features & Dashboards

1. Real-Time Location Tracking Dashboard (Priority Feature)

- Display interactive floor plans of classrooms, hallways, conference rooms, and common areas
- Render a 2D top-down view (similar to Pokémon-style game maps) showing:
- Real-time positions of detected individuals as icons/avatars
- Movement animations updated in near real-time
- Data fusion from: Wi-Fi location, RFID/fob access events, facial recognition, phone signatures
- **History Trail Feature**:
- Toggle on/off historical movement paths
- Default trail duration: 30 minutes
- Configurable range: minutes to days
- Floor plans will be generated using [specify actual tool - e.g., Google AI/custom generator]

2. Camera Monitoring Dashboard

- Live feeds from ~800 CCTV cameras
- Grid and single-camera views
- Alert overlays and event markers

3. Alerts & Activity Dashboard

- Real-time alerts from all integrated systems
- Activity logs with filtering and search
- Correlation of events across data sources

4. Landing/Demo Page

- B-roll and presentation videos
- PowerPoint presentations and sales materials
- Product documentation
- Login portal

Technical Architecture

Authentication

- Demo login page with default credentials: `admin` / `admin`

Database Layer

- **PostgreSQL** (primary, configured via `.env`):
- Mock real-time data for demo
- Alerts and activity logs
- User sessions and configuration
- **Future Integrations** (design for extensibility):
- Neo4j: Network/relationship topology
- Cloudinary (or similar): Image storage and CDN
- Google AI Vision: Image metadata extraction and analysis
- Google AI: Analytics, correlation engine, and predictive alerts

Configuration

- All database connections and API keys via `.env` file
- Environment-based configuration for demo vs. production modes

Deliverables Requested

1. System architecture diagram
2. Database schema design
3. Component/page structure
4. API endpoint specifications
5. Integration approach for each data source
6. UI/UX mockup recommendations based on ELI project
7. Implementation roadmap

Constraints

- This is a demonstration system with mock data
- Must be impressive for a live demo presentation
- Should clearly show differentiated value in campus security scenarios RAW NOTES: I need your help and need your suggestions on designing a college security application. I would like to create a camera monitoring web application for Brown university, with ~800 CCTV cameras, in the wake of the recent shooting in the past week and they are welcoming us to demo our product.

  This application would also work with other security systems, door key fobs, motion detectors, WiFi AP/WAP (using Cisco Catalyst Center integration or homebrew equilivant), etc. all correlated with the smart cameras from IREX.

  Use code, ideas, functionality, etc. from our existing ELI web page that is monitoring smart cameras in Peru, we plan on using the same technology in partnership with IREX for this application for Brown University.

  Search the internet/social media/news/etc. to get more detail on the improvements, full situation, technical mishaps, what a good camera system would have done, etc. on what a good camera system would have done to better help the situation that recently happened at Brown University and other similar situations at other universities that this system we are going to develop will prevent.

  On of the pages/dashboards will have floor plans of class rooms, hallways, conference rooms, etc. (I will use Google Nano Banana to generate these floor plans) in a live simulation showing people moving around (almost like a Pokiman video game 2d view, etc.) detected by the Wi-Fi, RFID cards, face recognitions, phone signatures, etc. all in combination to determine where people are near real-time and track a history of where they have been with a toggle to turn off/on the history trail that is configured to be initially 30 minutes, but can go back for hours or days.

  This is a demo and will have a landing page with b-roll videos, presentation videos, PowerPoints, document, sales info, etc. and a login page with the default login/password of admin/admin.

  The backend database will be PostgreSQL configured in the .env to store the initial mock real-time data, alerts, activities, etc. In the future Neo4J (topology), Cloudify (images) will be used in the future, Google AI to scan images to get meta data from the images, and Google AI to give analytics/correlation/predictions/etc.

  Please get more context/ideas/functionality/etc. on how to design this application from the previous ELI project for Peru: Github - IanNoble-Visium/ELI: ELI Dashboard web - ELI Unified Dashboard - Peru Surveillance Platform

  Please get more context/ideas/functionality/etc. on how to design this application from the

Visium Technologies website - Visium Technologies - TruContext Platform

Cisco Catalyst tracking - but only for Wi-Fi devices (including phones) and other RF tags—not directly from a physical keycard/fob unless that badge itself is a Wi-Fi/RFID tag integrated with Cisco's location services. Cisco Catalyst Center (formerly DNA Center) can show device locations on imported floor maps when combined with the right wireless/location components.cisco+3

What Catalyst Center Can Show

Catalyst Center can import or create building and floor maps, place APs on them, and generate RF heatmaps for coverage visualization.cisco

When integrated with a location engine (CMX, Cisco Spaces, or RTLS platform), wireless client locations can be visualized on those same floor maps, typically updated in near real time.cisco+2

Client 360 and similar views let you search by user, IP, or MAC and see where that client is connected, including historical location information.study-ccnp+2

Tracking Phones / Wi-Fi Devices

Any device with Wi-Fi enabled (such as smartphones) can be tracked via probe/association data from surrounding APs, which feed into CMX/Spaces to provide real-time or recent locations.cisco+1

Cisco Spaces' Detect & Locate and related apps provide "device tracker" functionality, showing current and last-known location of Wi-Fi devices on uploaded digital maps.cisco+2

This works even if the device is not actively in use, as long as it is periodically sending Wi-Fi probes or is associated to the network.study-ccnp

Keycards / Fobs (Badges)

Traditional access-control keycards/fobs that only work on door readers and do not emit Wi-Fi or compatible RF beacons cannot be located on Catalyst Center floor maps. study-ccnp

To track people by "badge," the badge must be an RTLS-capable tag (for example, Wi-Fi or active RFID tags used in healthcare or asset tracking) that integrates with Cisco location services such as Cisco Spaces or a partner RTLS platform. Those tags then appear as trackable devices on maps.blogs.cisco+2

Some third-party healthcare/location systems integrate with Cisco Spaces to deliver staff/patient badge tracking over the same Wi-Fi infrastructure, but that relies on specialized tags, not standard door-access cards.securitashealthcare+1

Putting It Together for Your Use Case

Real-time map tracking in a room is feasible for:

Wi-Fi phones and laptops connected to or probing on your WLAN.cisco+1

RTLS-enabled badges or tags integrated with Cisco Spaces/CMX and mapped into your building/floor hierarchy.blogs.cisco+2

Not feasible directly for:

Plain "dumb" keycards/fobs used only for doors, with no Wi-Fi/RTLS function and no integration to a location platform.study-ccnp

So, Catalyst Center with the appropriate Cisco location stack can show people's approximate positions on an imported floor plan via their wireless devices or RTLS badges,

but not purely from standard keycard/fob events unless those badges are themselves trackable RF devices integrated into the system.cisco+3
https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/dna-center/2-3-5/user_guide/b_cisco_dna_center_ug_2_3_5/m_work-with-wireless-2d-and-3d-maps.html
https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/dna-center/guide-c07-744044.html
https://study-ccnp.com/wi-fi-device-location-tracking-in-wireless-networks/
https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/m-product-overview.html
https://www.reddit.com/r/Cisco/comments/ue1dcf/dna_center_find_where_user_is_connected_to_the/
https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/m_map.html
https://spaces.cisco.com/detect-locate/
https://blogs.cisco.com/tag/rtls
https://www.securitashealthcare.com/news/securitas-healthcare-and-cisco-partner-bring-next-generation-location-services-healthcare
https://www.youtube.com/watch?v=ntfON_zcpjg
https://video.cisco.com/detail/video/5830481085001
https://support.ekahau.com/hc/en-us/articles/20294484365723-Ekahau-Cisco-DNA-C-Dashboard-Integration-How-To-Guide
https://www.youtube.com/watch?v=qsfafNrR0Cc
https://oomnitza.zendesk.com/hc/en-us/articles/27323073658775-Creating-an-extended-integration-for-Cisco-DNA-Center-network-assets
https://www.cisco.com/c/en/us/td/docs/wireless/spaces/config-guide/ciscospaces-configuration-guide/m-locations-and-maps.html
https://docs.paloaltonetworks.com/iot/iot-security-integration/network-management/integrate-iot-security-with-cisco-dna-center
https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/device-tracking.html
https://developer.cisco.com/docs/dna-center/2-3-5/overview/
https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/dna-center-assurance/2-3-5/b_cisco_dna_assurance_2_3_5_ug/b_cisco_dna_assurance_2_3_3_ug_chapter_01111.html
https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/catalyst-center/221585-utilize-the-traffic-telemetry-appliance.html

From <https://www.perplexity.ai/search/is-it-possible-for-cisco-catal-S9T6rCOySXqxRhBkGhO2sw>

Brown's December 2025 shooting exposed major gaps in camera coverage, real-time monitoring, and data fusion, which this Brown Campus Security Monitoring Platform can directly address by unifying cameras, sensors, and Wi-Fi location into an AI-driven, TruContext/ELI-style architecture. Below is a concrete design you can use as the basis for your new ELI application and presentation.[1] [2] [3] [4]

1. Incident context & design goals

- Brown has ~1,200 cameras today, yet investigators reported critical blind spots, especially in older buildings like Barus Holley, and no clear video of the masked shooter leaving campus.[4] [1]
- Traditional VMS recorded footage but did not provide real-time detection, correlation with access control, or automated alerts, forcing manual video review and reliance on community-submitted clips.[4]
- Expert commentary after the incident emphasizes that camera *count* is meaningless without comprehensive coverage, active monitoring, and AI-driven threat detection and correlation across systems.[3] [1] [4]

Design goals for this demo system:

- **Close coverage gaps** virtually by fusing cameras, access control, Wi-Fi location, and motion sensors into a single situational awareness layer.[5] [4]
- **Provide real-time detection** (weapons, unusual motion, crowding) using IREX AI modules and future Google AI, instead of purely forensic recording.[6] [7] [4]
- **Deliver an ethical, explainable platform** aligned with ELI/TruContext principles (layered analytics, role-based access, full audit trails).[8] [4]

2. High-level system architecture

Textual diagram (you can convert this into a visual diagram for slides):

- **Presentation Layer (Web App)**
  - React/Next.js single-page app modeled on the existing ELI dashboard: global map, camera list, event stream, incident detail drawer.[5] [4]
  - Core pages: Landing/Demo, Login, Real-Time Location Tracking, Camera Monitoring, Alerts & Activity, Admin/Config.[5]
- **API Gateway / BFF (Backend-for-Frontend)**
  - Node.js/Express or NestJS REST/GraphQL layer with JWT session management for `admin/admin` demo login.[4]
  - Aggregates data from:
    - ELI/IREX video analytics service (mocked in demo).
    - Cisco Spaces/Catalyst Center location API (mocked, with a realistic JSON schema).[9] [10] [11]
    - Access control/motion sensor simulator.
    - PostgreSQL for events, alerts, sessions, configuration.[4]
- **Integration & Analytics Layer (inspired by TruContext + ELI)**
  - **Ingestion services**:

        - Video events from IREX (weapons detected, person-of-interest via facial recognition, loitering, crowding).[7] [6] [4]
        - Access control events (door open/close, fob tap, alarm) via webhook/queue (mock producer).
        - Wi-Fi/RTLS events from Cisco Spaces/CMX (device seen at zone X, with accuracy metadata).[10] [11] [12] [9]
        - Motion sensor events from building systems (simulated).
    - **Correlation engine (TruContext pattern)**:
        - Normalizes all events to a common schema: `{time, source, entity_id, location_id, confidence, attributes}`.[4]
        - Builds relationships between entities: device ↔ person ↔ location ↔ camera; this is where future Neo4j will be used.[4]
        - Computes risk scores based on rules (e.g., "weapon detection + crowd running + door forced open in same zone within 1 minute").[4]
    - **Ethical guardrails**:
        - Policy engine governing who can see identity-level data (faces, names, device IDs) vs anonymized silhouettes.[8] [4]
        - Full audit logs of every search, playback, and export (TruContext / ELI governance model).[8] [4]
- **Data Stores**
    - **PostgreSQL**:
        - Event log, alerts, camera metadata, floor plans, user accounts, sessions, dashboard layouts.[4]
    - **Future**:
        - **Neo4j**: graphs of people/devices/locations/incidents, powering "who/what was near here in last 10 minutes?" queries.[4]
        - **Cloudinary** (or similar): thumbnails, snapshots, and uploaded evidence.[4]
        - **Google AI Vision / Google AI**: object/scene tagging, anomaly detection, predictive alerts.[4]
- **External Platforms (Mocked for Demo)**
    - IREX Smart City/ELI stack: AI video analytics, mapping/floor plan camera placement, alerts, facial recognition, weapons detection.[13] [14] [6] [7]
    - Cisco Spaces/Catalyst Center: location of Wi-Fi clients and RTLS tags on imported floor maps, via REST APIs or webhooks.[15] [11] [12] [9] [10]

3. Database schema (PostgreSQL – key tables)

Focus on what you need to drive the dashboards and correlation.

Core entities

- **users**
  - `id`, `username`, `password_hash`, `role`, `created_at`, `last_login_at`.[4]
- **cameras**
  - `id`, `name`, `building_id`, `floor_id`, `latitude`, `longitude`, `floor_x`, `floor_y`, `rtsp_url`, `status`, `tags` (JSONB: e.g., "weapon_detection=true").[13] [4]
- **buildings / floors**
  - `buildings`: `id`, `name`, `campus_zone`, `address`, `risk_level`.
  - `floors`: `id`, `building_id`, `name`, `level_number`, `floor_plan_id`.[4]
- **floor_plans**
  - `id`, `building_id`, `floor_id`, `image_url`, `scale`, `origin_x`, `origin_y`, `metadata` (JSONB, for zones like classrooms and exits).[4]
- **devices (Wi-Fi / RTLS / phones)**
  - `id`, `mac_hash`, `device_type`, `owner_person_id` (nullable), `last_seen_at`, `last_location_id`, `privacy_mode`.[9] [10] [4]
- **persons**
  - `id`, `external_id` (student/staff ID), `name`, `role`, `photo_url`, `risk_flags` (JSONB), `watchlist` (boolean).[4]

Event & correlation layer

- **events**
  - Generalized ingestion table.
  - `id`, `source_type` (`camera`, `access_control`, `wifi`, `motion`, `manual`), `source_id`, `event_type`, `entity_type` (`person`, `device`, `unknown`), `entity_id` (nullable), `location_id`, `timestamp`, `payload` (JSONB), `confidence`.[4]
- **alerts**
  - `id`, `severity`, `category` (`weapon`, `loitering`, `tailgating`, `door_forced`, `active_shooter`), `description`, `status`, `trigger_event_ids` (array), `created_at`, `acknowledged_by`, `resolved_at`.[4]
- **location_history**
  - For real-time map + history trail.
  - `id`, `entity_type` (`device`, `person`), `entity_id`, `floor_id`, `x`, `y`, `accuracy_m`, `source` (`wifi`, `rtls`, `face_reid`, `access_infer`), `timestamp`.[11] [10] [9] [4]
- **audit_logs**

- `id`, `user_id`, `action`, `target_type`, `target_id`, `timestamp`, `details` (JSONB).[8] [4]

This schema lets you power all requested dashboards and future expansion to Neo4j and Google AI.

4. Component & page structure

Modeled on existing ELI dashboard patterns (global map, multi-source event rail, drill-down incident views).[5] [4]

Top-level pages

1. **Landing / Demo Hub**
   - Hero section: "Transforming Brown Campus Security with Ethical Layered Intelligence."[5] [8]
   - Sections:
     - Background on Brown incident and security gaps.
     - Explainer video (B-roll) and Peru ELI demo reels.
     - Links to whitepaper ("Reimagining Campus Security"), PR, and sales deck.[8] [5] [4]
     - "Enter Live Demo" login button.
2. **Login Page**
   - Simple form with username/password, pre-filled tooltip: `admin / admin` for demo.
   - Story: note banner about audit logging and privacy to underscore ethical positioning.[8] [4]
3. **Real-Time Location Tracking Dashboard**
   - Layout:
     - Left: building/floor selector (tree: Campus → Building → Floor, similar to Cisco Spaces location hierarchy).[16] [15] [10]
     - Center: 2D floor plan canvas (Pokémon-style top-down) rendering:
       - Static assets: walls, doors, rooms (from generated floor plans).
       - Dynamic overlays: avatars for people/devices, color-coded by role/risk.[4]
     - Right: entity detail panel + timeline.
   - Components:
     - **FloorPlanCanvas** (WebGL/Canvas/SVG) with zoom/pan and layering.
     - **LivePresenceLayer**: subscribes to `location_history` stream (WebSocket) and animates movement.
     - **TrailToggle**: switch to enable/disable history trails, with time slider (30 minutes default, up to days).[4]
     - **FilterBar**: filter by role, risk score, event type (e.g., show only high-risk individuals or specific devices).

4. **Camera Monitoring Dashboard**
    - Layout:
        - Top: map/floor selector and camera filters (building, tag, analytics capability).[13] [4]
        - Main: switchable modes
            - Grid view: 2×2, 3×3, 4×4.
            - Single camera view with PTZ controls (mock) and overlayed AI detections.
        - Side rail: current AI events per camera (weapon detection, intrusion, etc.).[6] [7] [4]
    - Components:
        - **CameraGrid**: thumbnail tiles (using HLS or JPEG snapshot mocks).
        - **CameraDetail**: video player + bounding-box overlays for detected objects.
        - **CameraMapOverlay**: cameras on building/floor map, similar to IREX's map/floor plan camera placement.[14] [13] [4]
5. **Alerts & Activity Dashboard**
    - Layout:
        - Top: severity and source filters.
        - Left: real-time alert list, prioritized by risk score.
        - Center: timeline and correlation graph preview (camera → device → door → person).
        - Right: incident detail / playbook (checklist actions, radio dispatch, notes), aligned with TruContext incident workflows.[4]
    - Components:
        - **AlertList**, **AlertDetail**, **EventTimeline**, **CorrelationGraphPreview** (future Neo4j-driven).
6. **Admin & Config**
    - Manage buildings/floors, import floor plan images, position cameras and APs manually (demo UI similar to Cisco map editors).[15] [16] [4]
    - Configure integration endpoints (IREX API mock, Cisco Spaces credentials, simulated access control feed) via `.env`.[4]

5. API endpoint specifications (examples)

Base: `/api/v1`

Authentication

- `POST /auth/login`
    - Body: `{ "username": "admin", "password": "admin" }`.
    - Response: `{ "token": "JWT...", "user": { ... } }`.

Floor plans & locations

- `GET /buildings` → list buildings and stats.
- `GET /buildings/:id/floors` → floors with floor plan metadata.
- `GET /floors/:id/plan` → `{ imageUrl, scale, origin, zones[] }`.
- `GET /floors/:id/presence?windowMinutes=30`
  - Returns current & recent positions:
  - `[ { "entityType": "device", "entityId": "...", "x": 10.2, "y": 14.1, "accuracy": 2.0, "lastSeen": "...", "trail": [ ... ] } ]`.
- **WebSocket** `/ws/location`
  - Stream of `{ entityType, entityId, floorId, x, y, accuracy, source, timestamp }`.

Cameras & video analytics

- `GET /cameras?buildingId=&floorId=&tag=weapon_detection`
- `GET /cameras/:id`
- `GET /cameras/:id/stream` → HLS URL or snapshot URL (for demo, return placeholder).
- `GET /cameras/:id/events?since=` → IREX AI events (weapons, loitering, crowding).[7] [6] [13]

Alerts & events

- `GET /alerts?status=open&severity=high`
- `GET /alerts/:id`
- `POST /alerts/:id/acknowledge`
- `GET /events?entityId=&sourceType=&since=`

Cisco / Wi-Fi integration (behind the scenes)

In the demo, these endpoints call simulated services but are shaped like real Cisco Spaces/CMX data:[12] [10] [11] [9]

- **Internal service** polls or receives webhooks from Cisco:
  - `/integrations/cisco/spaces/webhook`
    - Accepts: `{ mac, x, y, floorId, accuracy, lastSeen }`.
    - Transforms to `location_history` insert + `events` record.

Administration

- `POST /admin/floor-plans` – upload new floor plan image and metadata.
- `POST /admin/cameras` – create camera with coordinates.

6. Integration approaches (per data source)

Smart cameras / IREX

- Connect Brown's (mock) cameras via RTSP/ONVIF to IREX Smart City / AI platform, which:
  - Runs object, weapon, people, and vehicle detection, plus "Searchveillance" for rapid forensic search.[14] [6] [7] [13]
  - Supports mapping cameras on a scalable map and building floor plans, mirroring Peru ELI capabilities.[14] [13] [4]
- The Brown ELI web app ingests IREX events over:
  - REST: `GET /irex/events?since=`.
  - Or WebSocket / MQTT channel for low-latency alerts.

Access control

- Ingest door events (badge in/out, forced open, door held) from existing access control system via webhook:
  - Payload: `{ doorId, personId?, cardId, eventType, timestamp }`.
- Correlate with cameras covering that doorway and nearby Wi-Fi/RTLS positions to infer movement direction.[4]
- Important constraint: **standard keycards are not trackable as location beacons**; they only provide door event markers unless paired with RTLS tags.[17] [18] [10]

Motion sensors

- Treat as simple binary sensors tied to floor zones (`zone_id`).
- In location dashboard, motion anomalies (motion in restricted area after hours) appear as pulsing overlays on the floor plan.

Wi-Fi location (Cisco Catalyst Center + Cisco Spaces/CMX)

- Use Cisco Spaces/Catalyst Center location APIs:
  - Import campus/floor maps and AP placements using location hierarchy and CMX tethering features.[16] [10] [15]
  - Retrieve device locations via:
    - Pull APIs or webhooks from Spaces Detect & Locate, which exposes current and historic location of Wi-Fi clients and tags.[10] [11] [9]
- Data model:
  - Device ID as hashed MAC; positions with accuracy and last-seen timestamps.
- Capabilities:
  - Track phones/laptops that probe or associate with the WLAN, updating near real time.[19] [12] [9]

    - Track RTLS-enabled tags/badges integrated with Cisco Spaces or partner RTLS platforms (like healthcare staff/patient badges).[18] [10]
- Limitations (explicitly call out in UI and docs):
    - Cannot track "dumb" keycards/fobs that lack Wi-Fi/RTLS signaling; you only know they used a door at a given time.[18] [10]

Facial recognition

- Provided by IREX's ethical facial recognition modules:
    - **Use cases in demo**:
        - Watchlist match alerts for known banned individuals.
        - Post-incident search for a suspect's last known trajectory.
    - Privacy controls:
        - Default to anonymized blobs, reveal identity only under elevated conditions (e.g., active incident role) and only to authorized roles.[14] [8] [4]

Phone signatures / device fingerprinting

- Combine Wi-Fi MAC hashes, device type, OS fingerprint, and browser user agent from captive portal/auth logs to create a device signature.[12] [9] [10]
- Associate signatures with individuals post hoc when they authenticate with campus SSO or register devices.

7. UI/UX patterns to reuse from Peru ELI & TruContext

From ELI and TruContext content, several reusable patterns stand out:[5] [4]

- **Single pane of glass**:
    - Multi-layer dashboards showing cameras, sensors, and events over geospatial context (map/floor plan).[4]
- **Event-driven UI**:
    - Right-hand activity stream of alerts with quick filters and click-through to video and context, as used in ELI incident workflows.[4]
- **Graph & timeline views**:
    - TruContext's graph-based analytics for relationships between entities and events; represent as simplified "incident graph" in the UI.[4]
- **Privacy-first UX**:
    - Toggles to mask/unmask identities, strong warning banners when viewing PII, and visible audit log link on incident view to support ethical AI narrative.[8] [4]

For the Pokémon-style floor plan:

- Use a tile-based or vector grid aesthetic; show simplified avatars color-coded by:

    - Green: normal.
    - Amber: policy anomaly (e.g., tailgating, after-hours presence).
    - Red: correlated threat (weapon detection nearby, forced door, rapid crowd movement).
- Animate movement by interpolating positions between location updates, smoothing Wi-Fi granularity.[11] [9]

8. Implementation roadmap (demo-focused, ~60–90 days)

Leverage the 90-day roadmap pattern already outlined in your Brown proposal, tailored to a demo.[5]

Phase 1 (Weeks 1–3): Foundations & mock integrations

- Stand up PostgreSQL, backend API skeleton, and React/Next.js front end.
- Implement core schema (users, cameras, floor plans, events, alerts, location_history).
- Build mock simulators for:
    - IREX video events (weapon detection, person-of-interest).
    - Cisco Spaces device locations for a handful of floors.
    - Door and motion events.

Phase 2 (Weeks 4–6): Dashboards & live demo flows

- Implement:
    - Real-Time Location Tracking dashboard with floor selection, avatars, and 30-minute default history toggle.
    - Camera Monitoring dashboard with grid/single view and event overlays.
    - Alerts & Activity dashboard with correlation preview.
- Connect simulators via WebSockets to demonstrate “live” incidents.

Phase 3 (Weeks 7–9): TruContext/ELI polish & narrative

- Add correlation rules and risk scoring, including an “active shooter scenario” scripted demo showing:
    - Weapon detection in Barus Holley corridor.
    - Crowd running behavior.
    - Doors locking and access events.
    - Wi-Fi devices clustering and moving.
- Add privacy controls, audit logging views, and an “Ethical AI” info drawer referencing ELI and TruContext principles.[8] [4]
- Integrate landing page content (PR, whitepaper, Peru case study, videos) for a compelling end-to-end story.[5] [8] [4]

9. What a better system would have changed (talking points)

You can use these in the presentation to connect architecture to Brown's incident:

- **From blind spots to coverage intelligence**: show a heatmap of camera coverage vs. high-risk buildings, highlighting how the system flags coverage gaps proactively rather than after a tragedy. [1] [3] [4]
- **From passive recording to real-time detection**: demonstrate simulated weapon detection in a corridor and immediate notifications with associated cameras and floor-plan positions. [6] [7] [4]
- **From siloed systems to fused context**: replay a scripted incident where door access logs, Wi-Fi location, and video analytics converge to track the suspect's path in near real time, rather than manually trawling through hundreds of feeds. [1] [4]
- **From opaque surveillance to ethical AI**: emphasize anonymization, role-based access, and complete audit trails as differentiators versus typical "surveillance-only" platforms. [8] [4]

If you want, the next step can be: a compact JSON/YAML "spec doc" for the APIs, or a component-by-component React layout sketch you can hand to your dev team.

⁂

1. https://www.cnn.com/2025/12/18/us/brown-university-surveillance-cameras-shooting-suspect
2. https://k12ssdb.substack.com/p/brown-university-has-1200-security
3. https://volt.ai/blog/seven-questions-university-leaders-must-ask-their-police-departments-after-brown-university-tragedy
4. Reimagining-Campus-Security.pdf
5. Brown_ELI_Presentation_Email.pdf
6. https://iotrevolution.co.za/irex/
7. https://www.aitechsuite.com/tools/irex.ai
8. Campus-Safety-Initiative-PR-v1.docx
9. https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/m-product-overview.html
10. https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/m_datasource.html
11. https://www.cisco.com/c/en/us/td/docs/wireless/spaces/detect-and-locate/b-cisco-cle/m_hyperlocation.html
12. https://blog.iptel.com.au/cisco-spaces-introduction
13. https://irex.ai/resources/datasheet
14. https://www.youtube.com/watch?v=dLIfkzJBYKs
15. https://www.cisco.com/c/en/us/td/docs/wireless/spaces/config-guide/ciscospaces-configuration-guide/m-locations-and-maps.html
16. https://www.cisco.com/c/en/us/td/docs/wireless/spaces/config-guide/ciscospaces-configuration-guide/m-defining-the-location-hierarchy.html

17. https://theconversation.com/theres-little-evidence-tech-is-much-help-stopping-school-shootings-272233
18. https://runbooks.ciscospaces.io/docs/cisco-spaces-os-runbook-cisco-validated
19. https://www.campussafetymagazine.com/resources/using-campus-security-cameras-for-effective-emergency-response/172302/
20. https://www.pbs.org/newshour/nation/students-community-frustrated-with-official-response-after-brown-university-shooting
21. https://www.wfdd.org/education/2025-12-16/how-college-campus-security-has-changed-to-prepare-for-violent-attacks
22. https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/catalyst-center-assurance/2-3-7/b_cisco_catalyst_assurance_2_3_7_ug/b_cisco_catalyst_assurance_2_3_6_ug_chapter_010000.html
23. https://halosbodycams.com/blog/how-body-cameras-support-active-shooter-response-on-campus