AXON Digital Evidence Management System (DEMS) - Engineering Design

System Scale & Requirements

Scale Parameters

- **Data Volume**: Hundreds of petabytes of video evidence
- **Users**: 1M+ law enforcement officers globally
- **Ingestion Rate**: Assume ~500k concurrent uploads during peak shifts
- **Geographic Distribution**: Global deployment across multiple regions
- **Retention Period**: 7-25+ years (varies by jurisdiction)

1. Evidence Upload Flow

Upload Pipeline Design

- Upload request => Upload Service => S3 Presigned URL => Done => Event Bus => Evidence Processor => Done => Event Bus => Audting and Indexing

Components

Upload Service

Technology: AWS S3

Key Features:

- **Resumable Uploads**: Chunked multi-part uploads (10MB chunks)
- **Network Resilience**: Automatic retry with exponential backoff
- **Client-side Encryption**: Device encrypts before upload (E2EE)
- **Pre-signed URLs**: Time-limited, scoped upload permissions

Flow:

1. Client requests upload token from API Gateway
2. Service validates user, generates pre-signed URL with metadata
3. Client uploads directly to object storage with client-side encryption
4. Upload completion triggers event to Evidence Processor Worker

Evidence Processor Worker

Technology: AWS Lambda or AWS SQS + Daemon

Stages:

1. **Validation**: Format validation, corruption check, hash verification
2. **Virus Scanning**: COTS
3. **Metadata Extraction**: Video metadata, GPS, timestamp, device ID
4. **Transcoding**: Multiple quality levels for video (original, high, medium, low)
5. **Thumbnail Generation**: Key frame extraction for quick preview
6. **ML Analysis**: Object detection, face detection (optional), scene analysis...etc

Event Publishing: Each stage publishes to event bus for audit trail

Storage Strategy

Primary Storage: Object storage (S3)

- **Encryption**: In transit and at rest
- **Storage Classes**:
  - Standard S3: Recent evidence (0-90 days)
  - S3 Glacier: Archive (>90 days)
- **Durabilty**: AWS targets 11 9s
- **Versioning**: Enabled to prevent accidental deletion

Metadata Storage:

- Relational DB (e.g. PostgressSQL) for Transactional data (S3 object names, case numbers..etc)
- Document store (e.g. MongoDB) for additional metadata

2. Search Flow

Search Requirements

- **Metadata Search**: e.g. Officer, date, location, case number, device ID
- **Full-text Search**: Transcription search (if videos are transcribed)
- **Geospatial Search**: Location-based queries
- **Temporal Search**: Time-range queries
- **Complex Queries**: Boolean logic, filtering, faceting
- **Sub-second Latency**: <500ms p95 for most queries

Elasticsearch or similar should be able to support all of these use cases

Architecture

- Search Request => API Gateway => Search Service => Elasticsearch
- *Discussion Point*: I'm questioning whether aggressive caching provides significant ROI here. Unlike social media or news sites where popular content gets viewed millions of times, evidence access follows a highly localized, long-tail pattern. Each video might only be viewed by 3-10 people total over its lifetime - the officer, their supervisor, and case-related legal staff. Geographic distribution further fragments the cache - there's no 'trending video' that everyone needs. This suggests we should focus our optimization efforts elsewhere, like upload performance and search speed.

Indexing Strategy

Technology: Elasticsearch or similar

Indexing Pipeline:

1. Upload completion event triggers indexer
2. Extract metadata from storage and processing results
3. Index to appropriate regional cluster
4. Publish index completion event

Search Service Notes

- **Consistency Model**: Accept eventual consistency (seconds to minutes lag)
- **Auditing**: Log queries for traceability (published on the event bus)

3. Audit & Immutability Architecture

Requirements

- **Permanent Record**: Every action recorded forever
- **Complete Chain of Custody**: Who accessed what, when, why
- **Discussion point**: what are the regulatory frameworks we need to comply with?

Architecture

- System Action => Event Bus => Audit Log Service => Immutable Storage

Immutable Storage

Technology: Append-only log (Kafka with 10 day retention) + S3 Object Lock

Kafka acts as our central audit event bus. Multiple services consume these events in real-time - for example, an alerting service watching for suspicious activity, a query service indexing for fast searches, and a dashboard service for metrics. Simultaneously, Kafka Connect archives everything to S3 for permanent, immutable storage. This gives us both real-time responsiveness and long-term durability.

Storage Strategy:

1. **Primary**: Kafka topics with infinite retention (compressed)
2. **Archive**: S3 with Object Lock (WORM - Write Once Read Many)
3. **Index**: Time-series database (InfluxDB/TimescaleDB) for queries

Object Lock Configuration:

- Compliance mode (cannot be deleted by anyone, including root)
- Retention period matches legal requirements
- Legal hold capability for active investigations