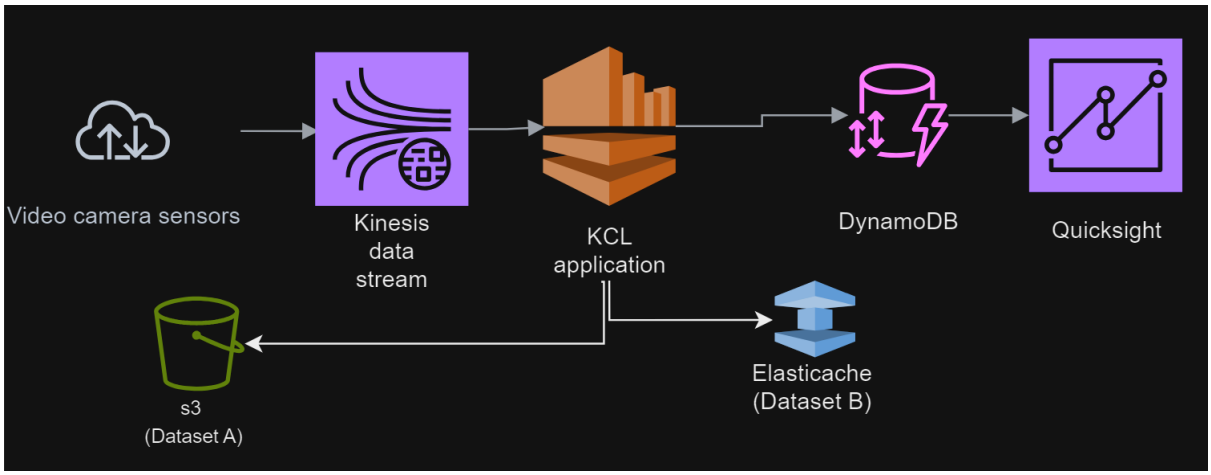
Proposed Architecture for Real-Time Data Processing

Solution Overview

We'll use AWS services to handle real-time ingestion, processing, storage, and visualization.

1. **Ingestion:** Use Kinesis Data Streams to collect data from cameras.
2. **Processing:** Use Kinesis Client Library (KCL) to handle the data processing.
3. **Storage:** Store processed data in DynamoDB for low-latency queries and archive raw data in S3.
4. **Analytics & Dashboard:** Use QuickSight to visualize the real-time data from DynamoDB

Architecture Diagram



Component Breakdown

1. Ingestion (Kinesis Data Streams)

- Captures high-volume streaming data from cameras.
- Retains data for a few hours for real-time processing.
- Scales automatically based on throughput

2. Processing (KCL Application)

- Performs real-time joins to prepare it for downstream analytics
- Handles deduplication via DynamoDB conditional writes.
- Archives raw data to S3

3. Storage (Elasticache + DynamoDB + S3)

- **ElastiCache:** Stores Dataset B in memory for sub-millisecond lookups.
- **DynamoDB:** Stores joined and enriched data for low-latency queries.

- **S3:** Archives raw event data for historical analysis.

4. Analytics & Dashboard (QuickSight)

- Connects directly to DynamoDB for real-time visualization.

Considerations & Trade-offs

1. **Deduplication Strategy:**
    - Use conditional writes to DynamoDB based on detection_oid.
    - Implement TTL for tracking processed events to prevent duplicates
2. **Join Strategy:**
    - **Pre-load Dataset B into ElastiCache for faster possible lookups**
    - Join performed in-memory within KCL application
3. **Latency vs. Cost Trade-offs:**
    - KCL on EC2/ECS has High Maintenance and troubleshooting shooting challenges
    - ElastiCache reduces lookup latency but adds slight operational complexity.

Questions for the PM

1. How quickly must the data be available in the dashboard? (Milliseconds vs. seconds)
2. How complex are the dashboard queries? (Will DynamoDB be sufficient?)
3. Expected data retention period? (Do we need tiered storage in S3?)
4. Does the system need to scale beyond 10K events/sec ?
5. How frequently does Dataset B (geographical data) change? (Impacts cache refresh strategy)

Why This Approach?

- **Handles High Volume:** Processes 10K events/second with room to scale.
- **No Timeout Constraints:** KCL applications run continuously.
- **Immediate Join Results:** ElastiCache ensures geographical location data is joined instantly.
- **Horizontally Scalable:** KCL consumers can be added to handle increasing load.
- **Cost-Effective:** Optimized for high throughput