HTX
Unified AI Platform

Connecting intelligence across agencies for a safer Singapore

August 2025

/thoughtworks

Table of contents

Summary

This report presents Thoughtworks' perspective on key design considerations, referred to as "big rocks", for the AI platform that HTX is envisioning. Rather than being exhaustive, it focuses on selected areas that will have significant impact on the success of a large-scale, vision-based AI system supporting multiple Home Team agencies. Each section highlights technical challenges, strategic trade-offs, and enabling technologies relevant to simulation, training, and deployment.

Using the "Three-Computer" solution as a guide, the report explores how HTX can:

- Design scalable, multi-tiered inference pipelines for real-time large scale video analytics.
- Adapt foundation models to Singapore's local context through fine-tuning on real and synthetic data.
- Leverage simulation environments for model validation and synthetic dataset generation.
- Use multimodal embeddings to enable fast, intuitive search via natural language queries.
- Generate photorealistic synthetic footage to expand and balance training datasets.
- Deploy AI agents that detect, reason, and trigger actions autonomously.
- Fuse video, sensor, and audio data to improve accuracy and situational awareness.

The goal is to spark interesting conversations between HTX and Thoughtworks to help shape a shared roadmap for a unified AI platform.

Introduction

The HTxAI movement is aligned with Singapore's National AI Strategy (NAIS 2.0) to position Singapore as a global leader in AI by 2030. As part of its AI strategy, the HTxAI movement is exploring the potential of applying cutting-edge developments in Artificial Intelligence (AI) for public safety and complex environments across agencies. Building AI powered solutions that can serve the needs of civil defence, border control, law enforcement, and other Home Team departments requires careful consideration of how AI infrastructure is designed, deployed, and managed at scale.

In contrast to siloed purpose-specific solutions, we propose that HTX adopt a **unified AI platform** approach that enables cross-agency coordination and multi-purpose applications. By integrating real-time perception (e.g. object detection) with Vision-Language Models (VLMs), simulation environments, and AI agents, HTX agencies will leverage a shared architecture, where models and AI agents respond to natural language queries, sensor data, and city-wide events. The outcome is a more capable, connected platform, empowering each department to respond faster, act smarter, and collaborate more effectively for a safer Singapore.

The need for holistic situational intelligence to enable **timely detection** and **response** to events across the homeland

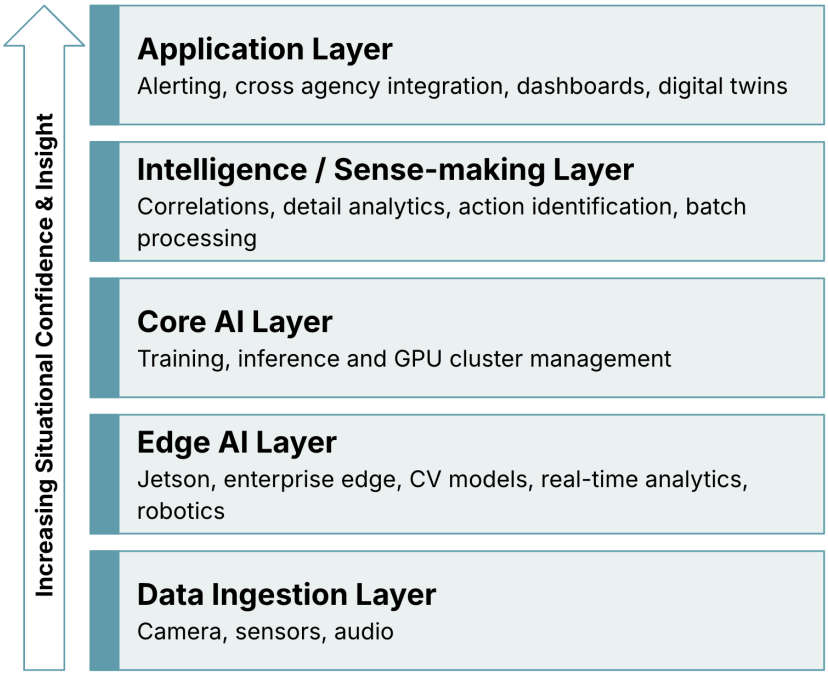
Police Force	Prison Service	Civil Defence	Immigration & Checkpoints	Internal Security Department	Central Narcotics Bureau
Crime prevention, public order, traffic enforcement	Inmate rehabilitation, prison safety	Emergency response, fire and rescue	Border security and customs control	Counter-terrorism, national security	Drug enforcement and surveillance
Crowd detection	Behavior monitoring (self harm, fights, exchanges)	Accidents (vehicle, people)	Behavioral anomaly	Threat recognition (unidentified objects)	Facial recognition for known drug offenders
Face find or recognition	Perimeter surveillance	Smoke and Fire	Licence plate + face recognition	Cross camera tracking (person of interest)	congestion on drugs hotspot
License plate recognition	Inmate reactions (gang related)		Crowd detection	Vehicle tracking across city	Behavior tracking (cross camera reid)
Behavioral patterns	Visitors (objects, behavior analysis etc...)			Crowd dynamics (protest/riot risk)	Hidden object detection

Authorities that HTX supports and their potential use cases

Platform Architecture

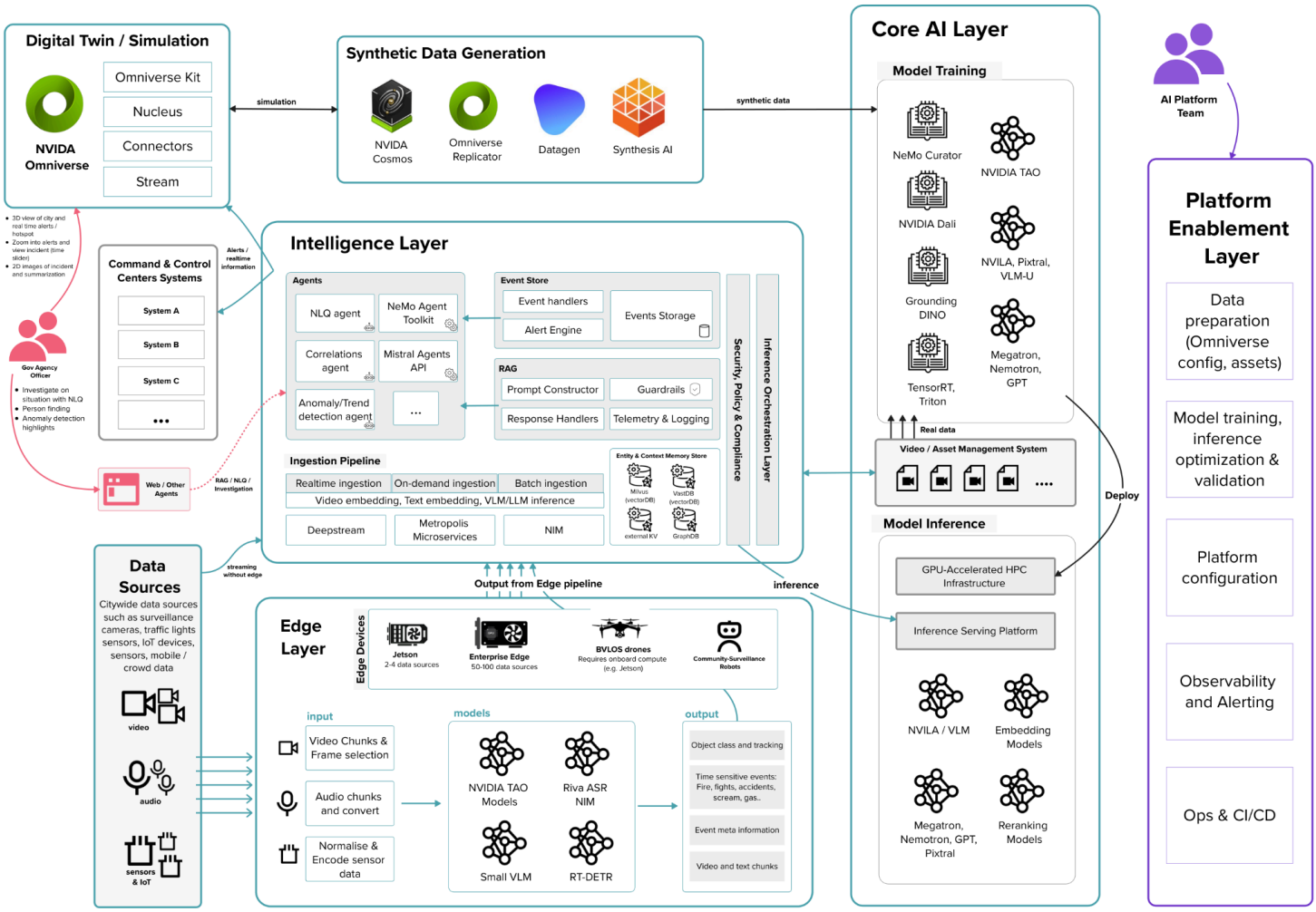
To realize this vision we propose a layered AI platform approach that focuses on reuse, scale and adaptability. At a high level, this platform architecture is composed of:

- **Application Layer** enables rapid development and deployment of AI-powered use cases by integrating with operational systems in existing command centres (e.g. POCC, FCCS, ICCC)
- **Sense-making Layer** generates insights through alerting, event streaming and batch analytics.
- **Core AI Layer** centralizes training, fine tuning and inference of pre-trained models on high-performance infrastructure.
- **Edge AI Layer** performs real-time analytics close to the source using platforms like Jetson and Enterprise AI servers.
- **Data Ingestion Layer** captures multimodal inputs, such as video, audio, and sensor signals, from across Singapore.



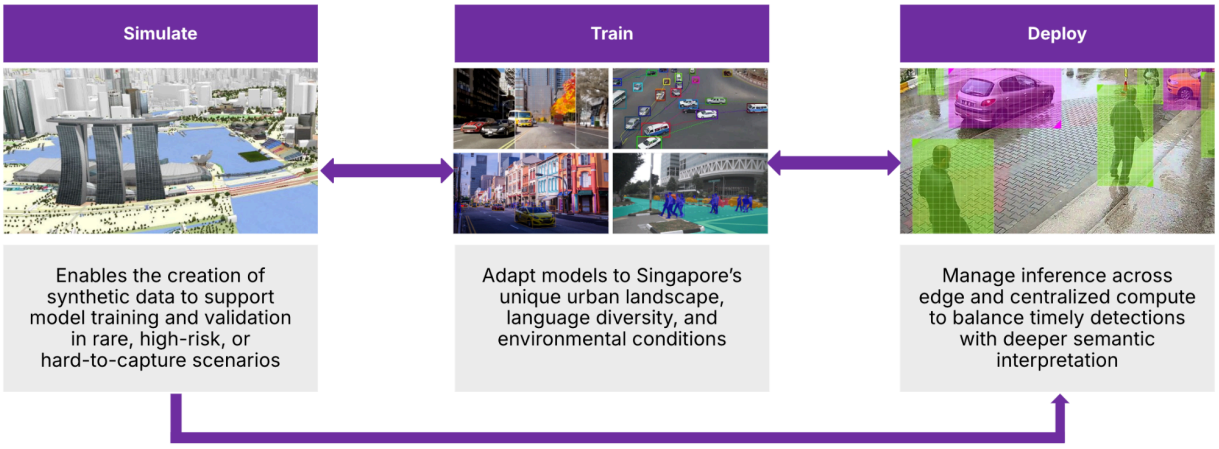
Conceptual framework for HTX's Unified AI Platform

With this architecture, individual data sources feeds (e.g. video, sensors, IoT, crowd mobile devices etc) can be interpreted differently across agencies: fire detection for civil defence, road obstruction for traffic authorities, or crowd disturbances for law enforcement.

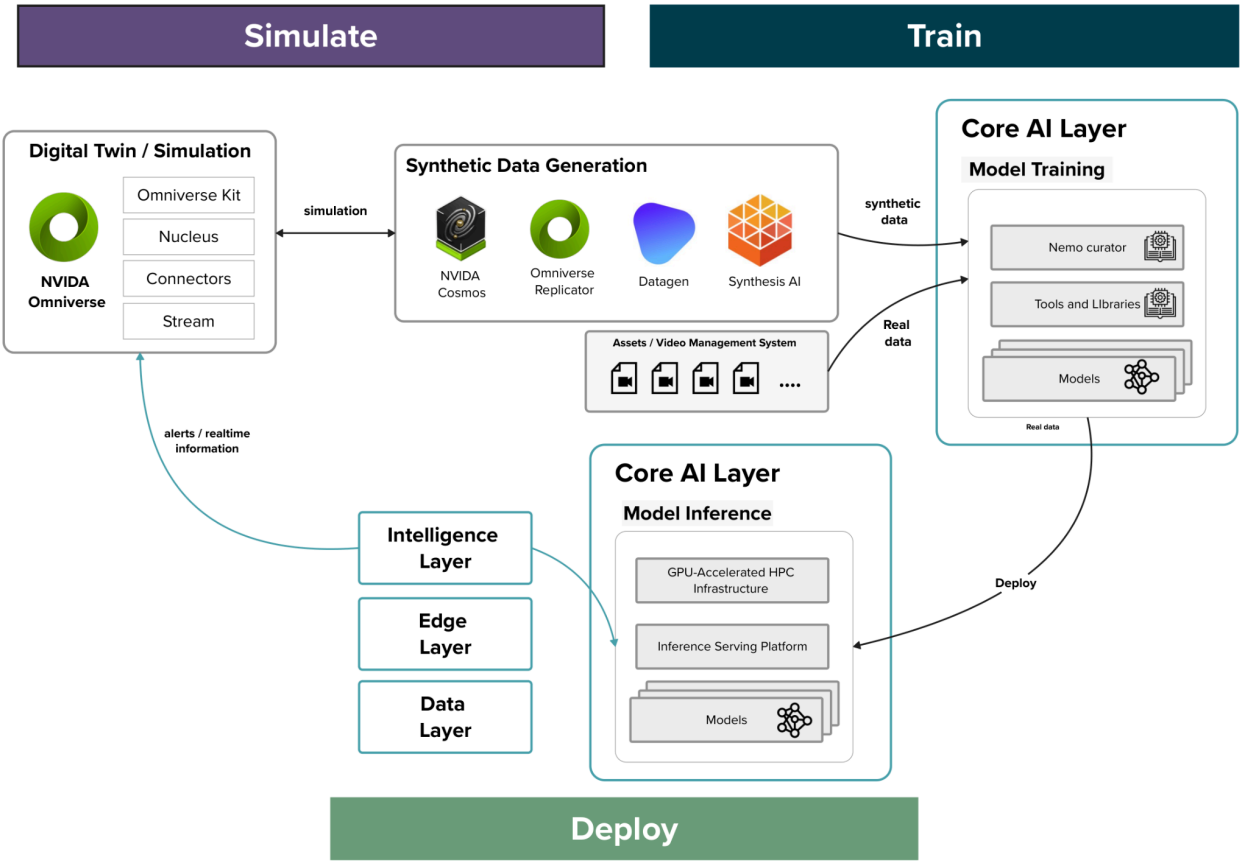


Architecture for Unified AI Platform

To realise the Unified AI Platform vision, we must address a set of foundational “big rocks”, high-impact challenges critical to success. The “**Three-Computer**” solution provides a good mental model for understanding the three distinct but interconnected stages of AI platform development: **simulation**, **training**, and **deployment**. Together, these stages form a reinforcing flywheel that improves model quality, scalability, and real-world performance. In this report, we’ll use this framing to explore a selection of key focus areas, “big rocks” that will help inform and shape the overall vision.



Three-Computer solution to developing AI systems



Mapping the three-computer mental model the the architecture

Let's break down each stages i.e. **simulate** - **train** - **deploy** to better understand the architecture and how it addresses the "big-rocks" challenges

Simulate

The architecture talks about 2 distinct stages under simulation i.e. Synthetic Data Generation and Digital Twin.

Synthetic Data Generation

Simulations can be used to generate large volumes of high-quality synthetic video data to augment real-world footage for model training. This is especially valuable in domains where labeled data is scarce, sensitive, or difficult to collect (e.g., emergencies, crowd events). By controlling environments and variables, HTX can create diverse and balanced datasets that improve model robustness and generalization.

Using synthetic data generation engines for creating photorealistic labeled datasets.

Digital Twin: validate solutions in digital worlds

In addition to training, simulation environments play a critical role in preparing AI systems for deployment. They enable the models and use case logic to be validated against edge cases and rare events without relying on real-world occurrences. By leveraging digital twins of city streets, infrastructure and public spaces, we can test how solutions perform under challenging conditions like occlusion, poor lighting and unusual behavior.

Tech Stack:

- **Omniverse**: 3D simulation and digital twin platform that enables realistic virtual environments for training, validation and prototyping.
- **Cosmos:** Scenario orchestration layer for Omniverse that allows large-scale multi-agent simulations.
- **Replicator:** Synthetic data generation engine built into Omniverse for creating photorealistic, labeled datasets.
- **Cosmos Nemotron:** Foundation model built for simulation, enabling reasoning, interaction, and task planning within synthetic environments.

Train

Adapting Models

Pre-trained models are a great way to get started, but they may not perform reliably in specific environments. Pre-trained models are trained on datasets that may not reflect the cultural nuances, ethnic distribution or urban landscape (MRT, HDB, etc.) of Singapore. These models (visual or language) must be adapted using transfer learning, refining them with local context drawn from real and simulated data.

NVIDIA TAO Toolkit for transfer learning

This adaptation is achieved through fine-tuning, which exposes the models to region-specific characteristics. By fine-tuning on this domain-specific data, models become significantly more accurate and robust in downstream tasks.

For vision based models this may include:

- Architectural features (e.g. MRT), cultural cues, signage, and traffic patterns
- Existing camera angles, sensor resolutions, and frame rates
- Local lighting and weather variations (e.g. glare, fog, night scenes)

For language base models that may require adding:

- Singapore location and landmark names
- Common government acronyms and local terms
- Government department names and definitions

Tech Stack:

- **TAO Toolkit:** Transfer learning and fine-tuning of pre-trained vision models using domain-specific datasets.
- **NeMo Video Curator:** Data curation tool that helps identify, extract, and label video segments for targeted fine-tuning.
- **NeMo Image Curator:** Similar to the Video Curator but focused on still imagery.
- **Grounding DINO:** A tool for generating annotations (e.g., bounding boxes) for objects in images or frames.

Deploy

Multi-Tiered Inference for Scalability

In large scale deployments that rely on visual sensors (e.g. 100k+ cameras), treating all video streams equally is both inefficient and economically unsustainable. As these networks grow, it's essential to manage compute, bandwidth, and storage.

Without these optimization, deployments risk:

- Wasting resources by running high-cost inference on low-value or irrelevant footage.
- Failing to scale during events such as emergencies, protests, or natural disasters.
- Creating fragile systems that are too costly and complex to maintain.

To address these challenges, HTX must adopt architectural strategies that balance the needs of each Home Team agency's use case with compute, networking and storage constraints.

For example, large scale camera-based systems typically use edge (or centralized) object detection models (e.g., YOLO) as a triage mechanism. These models filter out uninteresting frames before they reach more compute-intensive pipelines, reducing network bandwidth and GPU utilization. However, even state-of-the-art object detectors are prone to false negatives, particularly in scenarios like motion blur, night-time conditions or occlusions. As a result, important footage can be discarded too early in the pipeline, resulting in missed detections.

Object detection also produces sparse metadata (e.g., bounding boxes, class labels), which may be insufficient to describe complex or unexpected scenes (e.g. collapsed bridge, a derailed train). In these situations, downstream systems need richer semantic context.

An emerging alternative to object detection models, is to use vision-language models (VLMs) to generate textual representations of video frames or short clips. When VLMs return structured outputs (e.g. JSON scene descriptions), they provide more expressive representations that can be used to trigger alerts or workflow automation based on

natural-language-like visual understanding, rather than a list of rigid pre-defined object detection classes.

However, VLM inference is significantly more computationally demanding than running object detection models, making it cost-prohibitive to run across high-frame-rate streams at scale.

So in order to effectively scale camera inputs, while maintaining adequate visual representation fidelity, it's worth exploring a multi-tiered inference strategy:

- **Tier 1: High-Frequency, Low-Cost Object Detection.** Real-time triage, coarse filtering, and metadata tagging of video streams (e.g. 30 fps). Ideally object detection models are running as close to the source as possible to minimise network and GPU utilization.
- **Tier 2: Low-Frequency, High-Value Semantic Understanding.** Get a deeper semantic interpretation of visual scenes using VLMs on selected frames (e.g. 1–2 fps) or clips.

When combined with edge and centralised inference, this multi-tiered approach allows HTX to scale to large networks of cameras (e.g. 100K+ camera feeds) without overwhelming compute resources, while still preserving the ability to extract rich semantic information.

Tech stack:

- **DeepStream SDK**: High-throughput video analytics pipelines at the edge or data center.
- **Jetson and Enterprise AI servers:** Deployed close to the camera to process video locally, reducing network load and enabling real-time detection.
- **Triton Inference Server:** Multi-model inference workflows across GPUs with support for dynamic batch sizes and prioritization.
- **Metropolis Microservices:** Manage and deploy multiple inference pipelines across distributed edge and cloud systems, with monitoring, alerting, and lifecycle management.
- **Event Streaming (e.g. Kafka):** High-throughput event bus for transporting video metadata, alerts, and inference outputs.

Multimodal Inputs (Data sources and Edge Layer)

Real time video feeds are great for detecting objects under ideal conditions, but often lack additional contextual information that can be helpful to understand a scenario. Videos provide a 2D representation of the scene, they lack 3D information, suffer from occlusions and are susceptible to performance degradation in low-light conditions.

To enhance accuracy and reduce false positives or missed detections, solutions must augment video with other data sources (e.g. fire alarms, traffic signals, sound). This multimodal fusion allows solutions to better understand events, infer intent, or event severity.

For example:

- Fire detection becomes significantly more reliable when visual cues (e.g. smoke) are reinforced by data from smoke alarms or temperature sensors.
- Crowd disturbance detection improves when visual patterns are paired with audio to detect elevated sound levels.

This layered understanding is essential for solutions to operate effectively across complex environments.

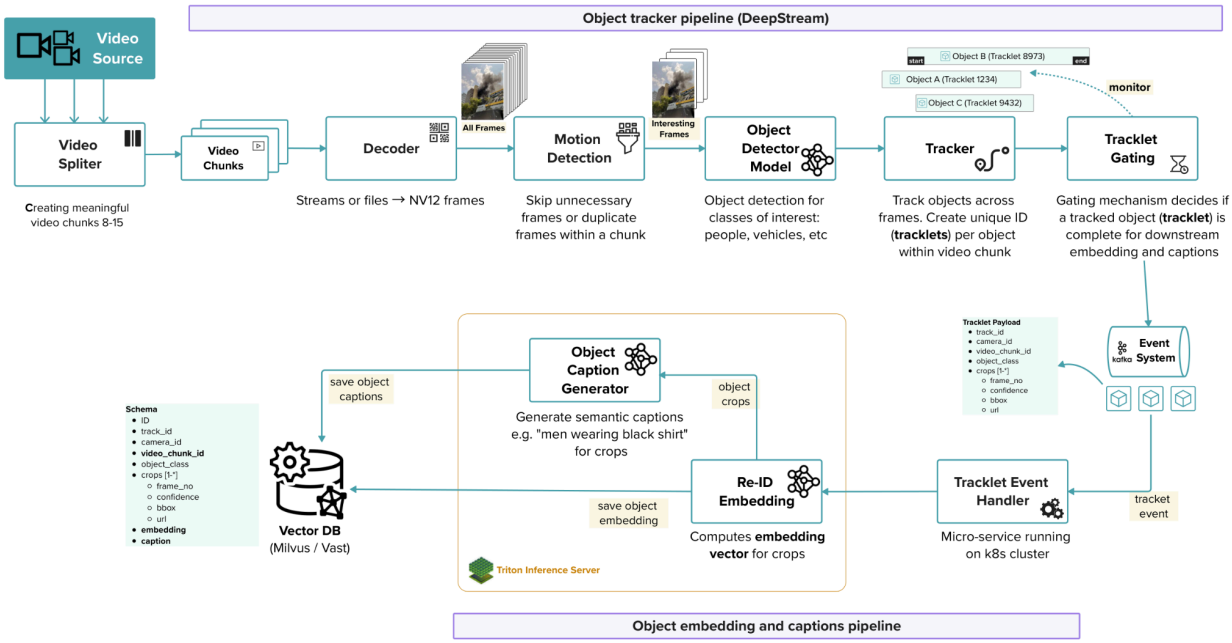
Tech stack:

- **DeepStream SDK:** Processes multi-sensor inputs (video, audio) and enables real-time fusion through customizable GStreamer pipelines.
- **Riva:** Provides real-time audio event classification, Automatic Speech Recognition (ASR) to extract insights from sound.
- **Triton Inference Server:** Executes multimodal models using ensemble pipelines that combine video, audio, and sensor inputs.

Ingestion Pipeline (Intelligence Layer)

Effective data ingestion is key to supporting a wide range of use cases. Complex scenarios—such as large-scale object re-identification (ReID) for persons, vehicles, and more—must efficiently balance thousands of concurrent camera streams, high detection accuracy, GPU resource constraints, and fast searchability of embeddings. At this scale, the GPU infrastructure is typically the single largest cost driver, so the input pipeline needs to be designed to minimise unnecessary processing while preserving critical events. Our approach combines motion-based frame filtering, multi-stream batching, and track-aware filtering to ensure that only meaningful person crops are passed to the ReID

embedding and object caption stages. This not only reduces the compute footprint but also allows the ReID embedding service to scale independently from the detector, handling bursty loads during peak activity.



ReID input exemplar ingestion pipeline to support Re-ID use-case

The table below outlines the pipeline stages in more detail that can work with real-time camera streams, or with video files.

Stage	Purpose
Camera Ingest	Handles network ingestion from IP cameras or files.
Decode	Hardware-accelerated video decode. Decodes H.264/H.265 into raw NV12 frames.
Motion Detection	Detects activity to trigger inference. Can be configured for ROI-based motion detection. If motion isn't detected the frame isn't sent to the Detector.
Stream batcher	Batch frames across cameras or files. Mandatory for multi-stream GPU efficiency. *Not shown in the diagram*

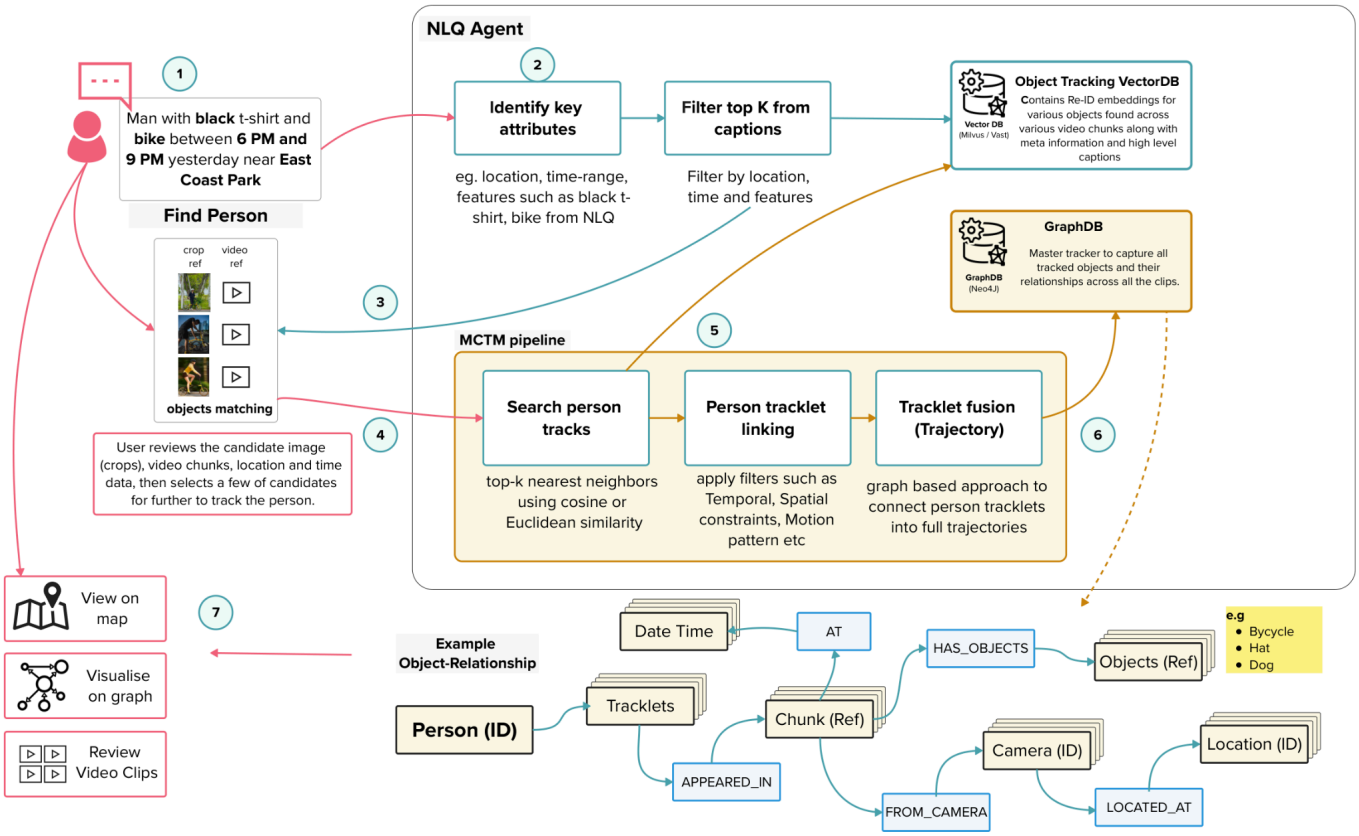
Detector	Full-frame object detection. Runs a detector model (PeopleNet, YOLO, PeopleNet-ViT) with TensorRT. **Outputs**: bounding boxes, class labels (Person), confidence, cropped image.
Tracker	Track objects on a single camera across frames. Uses a DeepStream Tracker (NvDCF, DeepSORT, IOU or ByteTrack) algorithm to maintain track IDs over time. **Outputs**: tracking status (new, ongoing, lost), track confidence.
Track-aware filtering	Decide which person crops to embed based on track status. Checks track time since last embed, IoU drift, occlusion, confidence then decides whether to send the image crop to ReID embedding.
ReID Embedding	Computes embedding vectors for selected crops in a track.
Object Caption Generator	Generate semantic captions e.g. "men wearing black shirt" for crops

AI Agents (Intelligence Layer)

As video, audio, and sensor networks scale to 100K+ streams across Singapore's Home Team agencies, detecting events after they occur is no longer enough. HTX must design proactive, adaptive solutions that enable timely decision-making and coordinated responses. At this scale, relying on humans to monitor and react in real time becomes economically unsustainable. Meeting this challenge requires a shift from static rules based pipelines to intelligent, agent-based systems that can reason over data, integrate with existing platforms, and collaborate across agency boundaries.

Hardcoded inference logic, such as "if person enters zone, then alert", leads to rigid systems that struggle in complex or evolving situations. These rule-based approaches often fail in real-world scenarios, such as incidents across multiple locations, or crowds during protests. In these situations, AI systems must respond to ambiguous inputs, integrate contextual information from sensors and external systems, and provide timely support to inform decision-making.

AI agents form the core of the intelligence or sense-making layer and are purpose-built to support various multi-turn investigations, such as natural language query (NLQ)-based person identification or video clustering. Consider the scenario where the office is initiating a person finding investigation using a Natural Language query to find a person's track in a city.



Officer using an Agent to find a person's track in a city using natural language query (NLQ) interface.

1. The query may specify time, location, and descriptive attributes to target relevant cameras and tracklets — for example "Man with black t-shirt and bike between 6 PM and 9 PM yesterday near East Coast Park"
2. The NLQ agent would Identify key attributes from the query such as **location**, **time-range** and **visual features** such as black t-shirt, bike.
3. Using these key attributes, the agent would query the Vector DB for track metadata, object crop images and reference video chunks and present it to the user.

4. The user reviews the candidate image (crops), video chunks, location and time data, then selects a few candidates to indicate further interest in tracking the respective person(s) tracks for a given time range and location.
5. This triggers the Multi-Camera Track Matching (MCTM) pipeline, which searches for likely matches across the camera network by combining visual similarity with spatio-temporal constraints
6. The identified tracklets are then fused into complete trajectories using a graph-based approach, where each tracklet is a node and connections are formed based on matching confidence. These fused trajectories are stored in a graph database (GraphDB), capturing complex relationships between observations across time and cameras
7. The results can then be presented to the user as a graph visualization, map-based person tracks, or raw video chunks for review. The user continues to interact with the NLQ Agent, providing feedback, applying filters, and refining the matches until the target is identified.

A critical architectural decision is where agents should live:

- **Edge Agents:** Operate on-site (e.g. at border checkpoints or correctional facilities) to respond quickly to local events using lightweight reasoning and cached embeddings.
- **Central Agents:** Aggregate data from multiple edge sites, run deeper multi-modal reasoning (e.g. combining video with contextual data from external systems), and coordinate system-wide behavior.

To operate effectively across Home Team agencies and legacy infrastructure, AI agents must be able to access context from existing systems such as incident management platforms or traffic control systems. Standards like the Model Context Protocol (MCP) offer a structured way to add knowledge into the agent ecosystem, enabling integration with existing Home Team systems.

By treating the platform as a coordinated fleet of reasoning entities, rather than fixed pipelines of models and rules, HTX can build flexible, adaptive solutions that scale with complexity.

Tech stack:

- **AI Agent Framework** (NeMo Agent Toolkit, CrewAI, Haystack Agents): Build and orchestrate LLM-based agents with tool use, memory, and multi-agent collaboration.
- **Visual Search & Summarisation (VSS)**: Embed, retrieve, and semantically query massive video datasets.
- **Kafka:** Event-driven messaging between distributed agent nodes.
- **Model Context Protocol (MCP):** Integrate with legacy systems and operational data sources.
- **Synthetic Data Tools** (Omniverse SimReady): Simulate agent interactions in synthetic environments for testing and training.

