

PROBLEM STATEMENT — MODEL 4

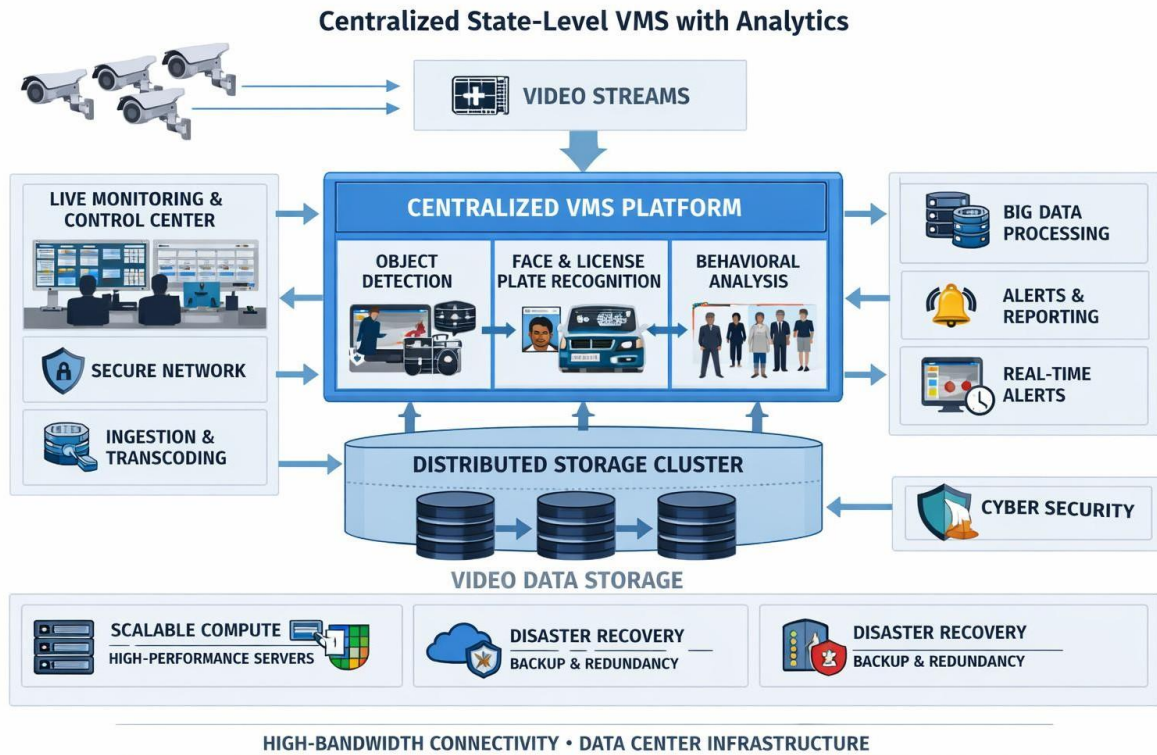


Fig 4

Tech Stack is illustrative , any open source may be used.

✦ Problem ID

- GOG-CCTV-04

✦ Problem Title

- Centralized State-Level Video Management System (VMS) with Analytics

✦ Theme

- Smart Governance / Advanced Monitoring

✦ Category

- Software + System Design

✦ Organization / Ministry

- Government of Gujarat – Home Department

✦ Industry Type

- Government Department

✦ Domain Tags

- Centralized VMS, Video Analytics, AI Monitoring, Big Data, Smart City

✦ Problem Statement (Description)

As outlined in Model 4 – Central VMS, all CCTV cameras are integrated into a single centralized platform that enables unified monitoring, management, and advanced analytics. This approach provides a consolidated view of monitoring across departments, improving coordination and decision-making.

However, such centralization requires robust infrastructure, including high-capacity storage, reliable network connectivity, and scalable system architecture. Ensuring performance, availability, and security at scale becomes critical to support large volumes of video data and real-time processing needs.

✦ Problem Statement (Short Abstract)

- Develop a centralized VMS platform integrating CCTV feeds with storage, monitoring, and AI-based analytics.

✦ Objectives

- Centralize monitoring
- Enable analytics
- Ensure scalability

✦ Expected Solution / Deliverables

- Central VMS platform
- Distributed storage system
- AI analytics module

- High availability design

✦ Dataset / Inputs

- Video streams
- Open AI datasets
- Video processing throughput benchmark (e.g., 100+ streams simultaneously)

✦ Constraints

- High infrastructure requirements
- Must ensure security

✦ Evaluation Parameters

- Performance
- Analytics capability
- Reliability

✦ Success Criteria

- Stable centralized system
- Accurate analytics

✦ Expected Outcome

- Unified monitoring ecosystem
- Enhanced decision-making

✦ Suggested Technology Stack (Open Source Only)

- TensorFlow / PyTorch
- FFmpeg / GStreamer
- Kubernetes
- Any other open-source technologies may also be used

✦ Use Case Scenarios

- Smart cities
- State Monitoring

✦ Adoption / Deployment Readiness

- State data center deployment

✦ Difficulty Level

- Advanced

✦ Impact

- Centralized governance
- AI-driven insights

SCORING RUBRIC — MODEL 4
(Out of 100 Marks)







EVALUATION CRITERIA	MARKS	DESCRIPTION / JUSTIFICATION
 System Architecture & Integration Depth	25	Measures how well the solution integrates Cameras, Registry (M1) and Control Room (M2) into a unified, scalable, and maintainable ecosystem.
 Core Functional Modules & Workflow Automation	25	End-to-end automation—discovery, connectivity, status, health, uptime, and alerting—are critical for real operational utility.
 Reliability, Performance & Scalability	20	The platform must handle large-scale, real-time operations with high accuracy and stability, ensuring use of pre-tested data instead of live production data.
 Security & Access Control	20	The platform must handle large-scale, real-time operations, with high accuracy and stability, ensuring use of pre-tested data instead of live production data.
 Innovation & Intelligence	5	Secure device onboarding, encrypted communication, robust authentication, and audit trails are essential.
 API Ecosystem & Extensibility	5	Well-designed, documented APIs enable easy integration with future systems (Analytics, VMS).
TOTAL	100	

Fig 4.1