Task 02: Cloud Architecture

Edge Devices (NVIDIA Jetson):

- Purpose: These embedded computers are deployed at the edge to capture and preprocess video streams from IP cameras.
- Functionality: NVIDIA Jetson devices are equipped with GPUs and designed for edge computing tasks, enabling efficient processing of video data in real-time.

RabbitMQ (Message Broker):

- Purpose: RabbitMQ acts as a message broker facilitating communication and coordination between edge devices and cloud-based components.
- Functionality: It enables asynchronous messaging between distributed systems, ensuring reliable data transmission and decoupling of components.

YOLO (You Only Look Once) for Object Detection:

- Purpose: YOLO is used for real-time object detection in video streams captured by IP cameras.
- Functionality: It utilizes deep learning techniques to detect and classify objects within each frame of the video stream efficiently.

DeepSort for Object Tracking:

- Purpose: DeepSort is employed for object tracking, allowing the system to track detected objects across consecutive frames.
- Functionality: It utilizes deep learning-based techniques to track objects by associating detections over time and maintaining consistent identities for each object.

Elasticsearch:

- Purpose: Elasticsearch is utilized for indexing and searching metadata extracted from video data streams.
- Functionality: It provides fast and efficient full-text search capabilities, enabling quick retrieval of relevant information from large volumes of data.

Google Cloud Storage:

- Purpose: Google Cloud Storage serves as a scalable and durable storage solution for storing raw and processed video data.
- Functionality: It provides reliable storage infrastructure with high availability and low latency, suitable for storing large volumes of data in the cloud.

Firebase:

- Purpose: Firebase is integrated into the user interface to provide real-time updates, user authentication, and cloud messaging features.
- Functionality: It offers a suite of tools and services for building mobile and web applications, including real-time database, authentication, and hosting.

Compute Cluster (NVIDIA GPU-based Kubernetes):

- Purpose: The compute cluster hosts containerized workloads, including data processing, analysis, and machine learning tasks.
- Functionality: Kubernetes manages the deployment, scaling, and orchestration of containerized applications, while NVIDIA GPUs accelerate computations for tasks like deep learning and video processing.

API Gateway:

- Purpose: The API Gateway serves as a central entry point for incoming requests from users and edge devices.
- Functionality: It handles authentication, routing, rate limiting, and other API management tasks, ensuring secure and efficient communication with the system's backend services.

User Interface (Web Dashboard):

- Purpose: The user interface provides a graphical interface for users to interact with the system, view live streams, configure settings, and receive alerts.
- Functionality: It is developed using technologies like Node.js, webRTC, and React.js or Angular for front-end development, offering a seamless user experience across devices.

GitLab Runner:

- Purpose: GitLab Runner is used for continuous integration and continuous deployment (CI/CD) pipelines for building, testing, and deploying components of the system.

NodeJS:

- Purpose: Node.js is used for developing server-side applications or services, including the backend logic of the user interface.
- Functionality: It provides a runtime environment for executing JavaScript code on the server, facilitating the development of scalable and performant web applications.

webRTC:

- Purpose: webRTC is integrated into the user interface for real-time communication and streaming between clients and the cloud-based video management system.
- Functionality: It enables peer-to-peer communication between web browsers, allowing for real-time audio and video streaming without the need for plugins or additional software.

Docker:

- Purpose: Docker is used for containerization of application components, ensuring consistency and portability across different environments.
- Functionality: It enables the packaging of applications and their dependencies into lightweight, portable containers, facilitating deployment and scalability in cloud environments.