<교통단속카메라 시스템>

Software Architecture Description

작성자: 한범진, 허윤아, 위기화, 진항우

(소속: 2조)

Revision History

Version	Date	Summary
0.1	09.20	#1 Project Overview 작성
0.2	09.27	#2.3 System Feature List까지 작업 및 #1 Project Overview 수정
0.3	10.04	#3.1 Primary Functionality까지 작업 (System Sequence Diagram 제외) 및 #2 System Overview 수정
0.4	10.11	#3.1 Primary Functionality까지 작업 (System Sequence Diagram 포함) 및 #2와 #3 수정
0.5	10.18	#3.2 Quality Attribute Scenario 작업 및 #2.3 System Feature List 수정
0.6	11.01	#3.2를 포함하여 전반적인 수정
0.7	11.08	#4.1 Candidate Designs per QA 작업
0.8	11.22	#4.2, #4.3, #5.1 작업

Index

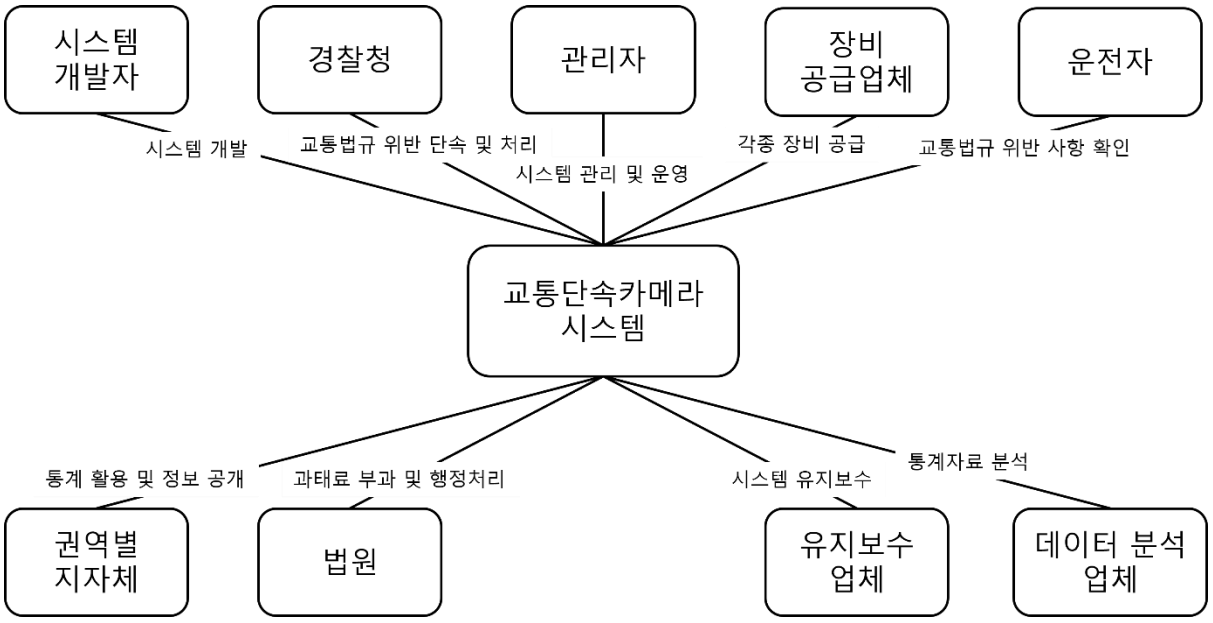
1. Project Overview

1.1. Project Background

본 프로젝트는 과속, 신호 위반, 불법 주정차 등 **도로교통법 위반을 단속**할 수 있는 카메라 시스템들로부터 교통법규 위반 정보를 입력 받아, 처리하고, 외부로 전송하기 위한 **교통단속카메라 시스템의 아키텍처 디자인**을 위한 프로젝트이다. 교통단속카메라 시스템의 서버는 각 권역 (도, 시 등의 지자체 단위)마다 존재하고, 각 권역의 서버 데이터를 모아 처리하는 전국 단위의 중앙 서버가 별도로 존재한다.

프로젝트의 대상이 되는 교통단속카메라 시스템 외부의 각종 단속 카메라 시스템들은 카메라로 촬영한 정보를 교통단속카메라 시스템으로 실시간으로 전송한다. 각 권역별 서버에서는 해당하는 권역 내의 각종 단속 카메라 시스템들로부터 입력 받은 정보를 토대로 촬영된 **차량의 번호가 유효한지 확인**하고, 차량 번호를 토대로 **소유주 정보를 조회**하여 이 정보를 중앙 서버로 전송한다. 중앙 서버에서는 이 정보들을 종합하여 **AI로 교통법규 위반 정보를 판단**하여 경찰청 등 외부 서버로 전송한다. 이 외에도, 개별 단속 카메라 시스템은 경찰청 등에서 제공한 도난 차량 정보, 범죄에 연루된 차량 정보 등을 입력 받으면, 이를 감지하여 **즉시 정보를 제공**할 수 있다. 이 정보는 실시간으로 각 권역별 서버와 중앙 서버에 전송되어 외부에서 즉각적으로 대응할 수 있도록 한다.

1.2. Business Context Diagram



1.3. Stakeholder List

Stakeholder	Description
시스템 개발자	설명: 시스템을 개발하는 사람/단체. 관심사: 시스템의 확장성 및 향후 업그레이드 가능성을 높이고자 하며, 다른 이해관계자와의 효과적인 의사소통 및 협력을 원함.
경찰청	설명: 교통법규 위반을 단속하고 처리하기 위한 단체. 관심사: 교통법규 위한 사항을 확인하고자 함. 위반한 사람의 인적사항과 어떤 교통법규를 위반하였는지 확인해야 함. 도난차량 등 범죄에 연루된 차량의 신속한 추적을 원함.
관리자	설명: 시스템 관리 및 운영을 담당하는 사람/단체. 관심사: 시스템을 사용할 때 편리하게 사용하고자 하며, 안정적인 운영과 신속한 기술 문제 해결을 원함.
장비 공급업체	설명: 각종 하드웨어 장비를 공급하는 업체. 관심사: 하드웨어 장비 공급을 통해 금전적 이익을 얻고자 함. 장비의 성능, 신뢰성 및 시장 경쟁력을 높이고자 함.
운전자	설명: 교통법규를 준수하며 도로 위의 각종 차량을 운전하는 사람. 관심사: 본인이 어떤 교통법규 항목에 위반되어 어떤 처벌을 받게 되었는지 확인하고자 함.
권역별 지자체	설명: 각 권역마다 존재하는 지방자치단체. 관심사: 데이터 분석 업체로부터 제공받은 통계자료를 활용하고 정보를 공개함으로써 공공의 이익을 추구함.
법원	설명: 교통법규를 위반한 사람에게 과태료 등 행정적 처리를 하는 기관 관심사: 위반한 사람의 인적사항과 어떤 교통법규를 위반하였는지 알고 싶어함. 교통법규를 어느 정도로 위반하여, 어떤 수준의 행정적처리가 이뤄지는 것이 맞는지 알고 싶어함.
유지보수 업체	설명: 시스템의 유지보수를 수행하는 업체. 관심사: 단속 시스템이 문제없이 24시간 동작하기를 원함. 유지보수성이 높기를 원함.
데이터 분석 업체	설명: 제공받은 데이터를 토대로 분석하는 업체. 관심사: 되도록 많은, 양질의 데이터를 가지고 통계자료를 만들고 분석하고자 함.

1.4. Business Goal List

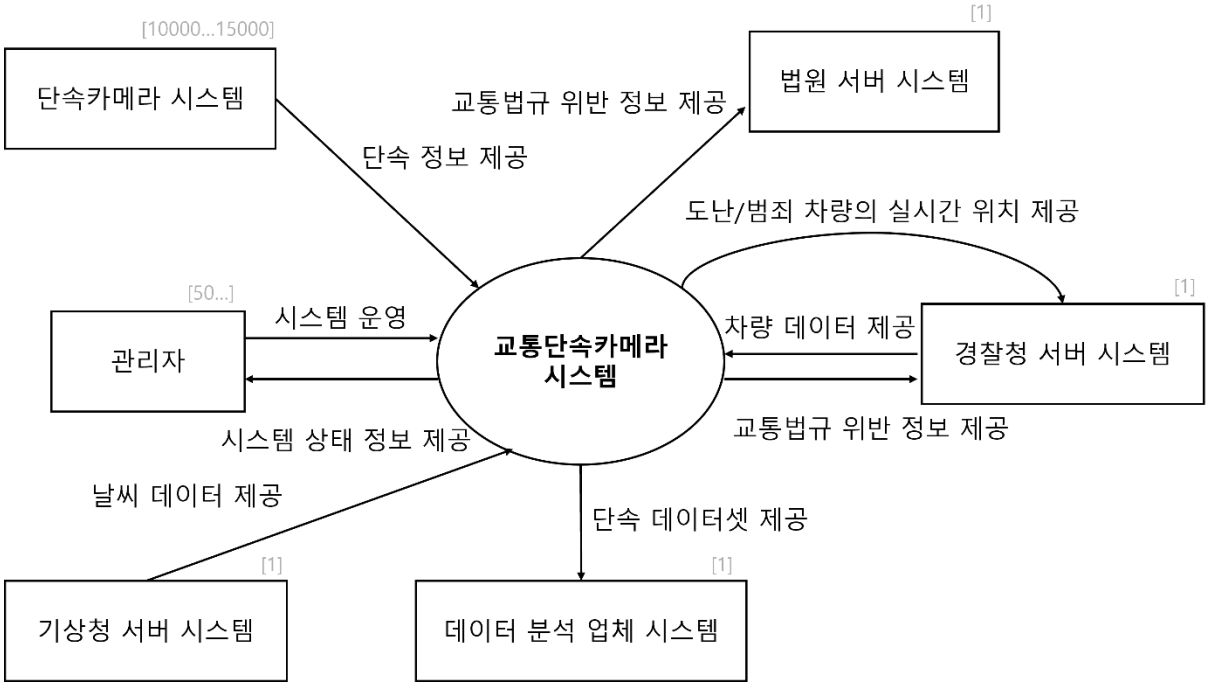
Stakeholder	Business Goal		
	ID	Statement	I
시스템 개발자	BG-01	시스템의 개발을 통해 더 많은 고객을 유치하고자 함. - 양질의 시스템을 개발하여, 지난 해 대비 30% 더 많은 잠재적인 고객을 얻고자 함.	하
	BG-02	시스템의 개발을 통해 금전적 이익을 얻고자 함. - 양질의 시스템을 개발하여, 타 업체 대비 120% 이상의 수익을 얻고자 함.	하
경찰청	BG-03	교통 안전 향상 및 사고 예방을 위한 효과적인 시스템 구축을 원함. - 전년도 대비 130% 이상의 시민이 교통법규를 준수하길 원함.	상
	BG-04	범죄에 연루된 차량을 빠르게 추적하기를 원함. - 목표한 차량이 카메라 시스템에 인식되었을 경우 1분 이내에 해당 정보를 전달받기를 원함	상
관리자	BG-05	시스템을 효율적으로, 편리하게 관리하길 원함. - 시스템의 특정 기능을 실행 시 1분 이상 지연되지 않길 원함.	중
장비 공급업체	BG-06	장비 공급을 통해 금전적 이익을 얻고자 함. - 타 업체 대비 130% 이상의 수익을 얻고자 함.	하
운전자	BG-07	운전자는 본인의 도로법규 위반 사항을 빠르고 간편하게 확인하기를 원함 - 본인의 6개월 이내의 도로법규 위반 사항을 1분 이내에 확인하기를 원함.	중
권역별 지자체	BG-08	데이터 분석 업체로부터 얻은 통계 자료를 활용하고 시민들에게 이를 제공함으로써 공공의 이익을 취하고자 함. - 3개월 간의 통계 데이터를 제공하여, 권역 내의 모든 시민이 접속하고 확인할 수 있도록 하고자 함.	상
법원	BG-09	법원에서는 행정적 절차를 처리하기 위해 필요한 정보를 누락 없이 전달받기를 바람.	상
	BG-10	제공받은 증거가 법적 소송에서 활용하기에 충분하기를 원함. - 95% 이상의 정확도로 분석된 자료를 제공받고자 함.	상
유지보수 업체	BG-11	유지보수 업체는 장비에 발생한 문제를 1시간 이내에 확인할 수 있기를 원함.	중

데이터 분석 업체	BG-12	데이터 분석 업체는 잘 정리된 유의미한 데이터를 활용할 수 있기를 원함.	중

* I : Importance << 상 중 하로 구분 >>

2. System Overview

2.1. System Context Diagram



Name	Description
단속카메라 시스템	유형: 시스템 역할: 한 컴퓨터 장비에 연결된 여러 카메라가 신호 위반, 과속, 불법 주정차 등에 대한 단속을 수행하여 이미지, 시간, 속도, 카메라 정보 등의 단속 정보를 저장하고 제공하는 시스템 관련 Stakeholder: 유지보수 업체, 장비 공급업체 시스템의 사양: 야간 촬영이 가능한 고해상도 카메라, 속도 측정 센서, 자체 DB, 네트워크 통신 장비 시스템의 품질 수준: - 가용성 (availability): 99.9% (24시간동안 상시 가동. 단, 차량이 적은 시간대에는 잠시 작동을 멈추고 DB를 초기화) - 사이버보안 (cybersecurity): 99.99% (데이터 전송 도중 정보가 탈취되거나 변경되지 않도록 함) - 정확성 (correctness): 95% (정확한 정보를 전송)
경찰청 서버 시스템	유형: 시스템 역할: 교통단속카메라 시스템으로부터 도난/범죄 차량의 실시간 위치 정보와 교통법규

	위반 정보를 제공받고, 차량 데이터를 제공 관련 Stakeholder: 경찰청, 운전자 시스템의 사양: 대용량 DB, 원활한 네트워크 통신 시스템의 품질 수준: - 신뢰성: 99.999% - 사이버보안 (cybersecurity): 99.999% (데이터 전송 도중 정보가 탈취되거나 변경되지 않도록 함) - 데이터 무결성: 99.999% (올바른 차량 데이터가 보관되어야 함)
법원 서버 시스템	유형: 시스템 역할: 교통법규 위반 정보를 제공받음 관련 Stakeholder: 법원, 운전자 시스템의 사양: 대용량 DB, 원활한 네트워크 통신 시스템의 품질 수준: - 데이터 무결성: 99.999% (올바른 위반 정보가 보관되어야 함)
데이터 분석 업체 시스템	유형: 시스템 역할: 교통법규 위반 정보를 제공받아 교통 데이터 분석, 통계 보고서 생성. 관련 Stakeholder: 데이터 분석 업체 시스템의 사양: 대용량 데이터 처리 능력, 대용량 DB, 원활한 네트워크 통신 시스템의 품질 수준: - 정확한 분석 결과: 99%의 확률로 편향되지 않은 데이터 분석 결과를 도출함 (되도록 정확한 통계 결과 제공)
관리자	역할: 사용자 숙련도: 서버 시스템 운영에 대한 전문성을 보유하고 있음 핵심 기대 사항: 시스템을 통하여 단속카메라 시스템 현황을 파악하고자 함. 시스템에 문제가 발견되었을 경우 빠르게 파악하고 조치할 수 있기를 원함.
기상청 서버 시스템	유형: 시스템 역할: 날씨 데이터 제공 관련 Stakeholder: 데이터 분석 업체, 경찰청 시스템의 사양: 대용량 DB, 원활한 네트워크 통신 시스템의 품질 수준: - 정확성 (correctness): 90% (정확한 날씨 데이터를 제공)

2.2. External Interface List

Name	Description
단속 정보 제공	역할: 단속카메라 시스템이 실시간 단속 정보 제공 System interface: HTTPS 특성: - 단속 데이터가 너무 많은 경우의 시스템 부하 고려 필요 - 단속카메라 시스템이 단속 정보를 실시간으로 제공하므로 원활한 네트워크 통신이 필요
시스템 운영	역할: 관리자가 시스템을 관리하고 운영 User interface: Web UI 특성: - 관리자는 서버 데이터의 주기적인 초기화를 관리 - 문제 발생 시 필요한 조치를 취할 수 있어야 함
시스템 상태 정보 제공	역할: 시스템 관리자에게 시스템의 현황 및 문제점을 제공 User Interface: Web UI 특성: - 지속적인 시스템 모니터링 데이터 제공 - 신속하게(문제가 파악된지 1초 이내로) 전달되어야 함.
교통법규 위반 정보 제공	역할: 경찰청 서버 시스템과 법원 서버 시스템에 교통법규 위반 정보를 제공 System interface: HTTPS 특성: - 하루에 한 번, 주기적인 제공 필요
도난/범죄 차량의 실시간 위치 제공	역할: 경찰청 서버 시스템에 도난 및 범죄 차량의 데이터를 제공 System interface: HTTPS 특성: - 신속하게(데이터가 생성된지 1초 이내로) 전달되어야 함. - 요청이 종료될 때까지 안정적으로 지속 전달되어야 함.
차량 데이터 제공	역할: 경찰청 서버 시스템에 차량 데이터를 요청 System interface: HTTPS 특성: - 요청빈도는 일 1회 정도로 높지 않을 것으로 예상 - 요청당 소량의 신규 및 삭제 데이터 정보만 전달되므로 시스템 부하 고려가 필요 없을 것으로 예상
단속 데이터셋 제공	역할: 데이터 분석 업체에 일정 기간 동안의 단속 데이터셋을 제공 System interface: API 특성: - 전달 빈도는 주당 1회로 높지 않을 것으로 예상 - 전달 데이터셋이 기간내의 많은 정보를 포함하고 있으므로 시스템 부하 고

	려 필요
날씨 데이터 제공	역할: 시스템에 날씨 데이터 제공 System interface: API 특성: - 날씨 데이터가 정확하지 않거나 중간에 비는 경우에 대한 고려 필요

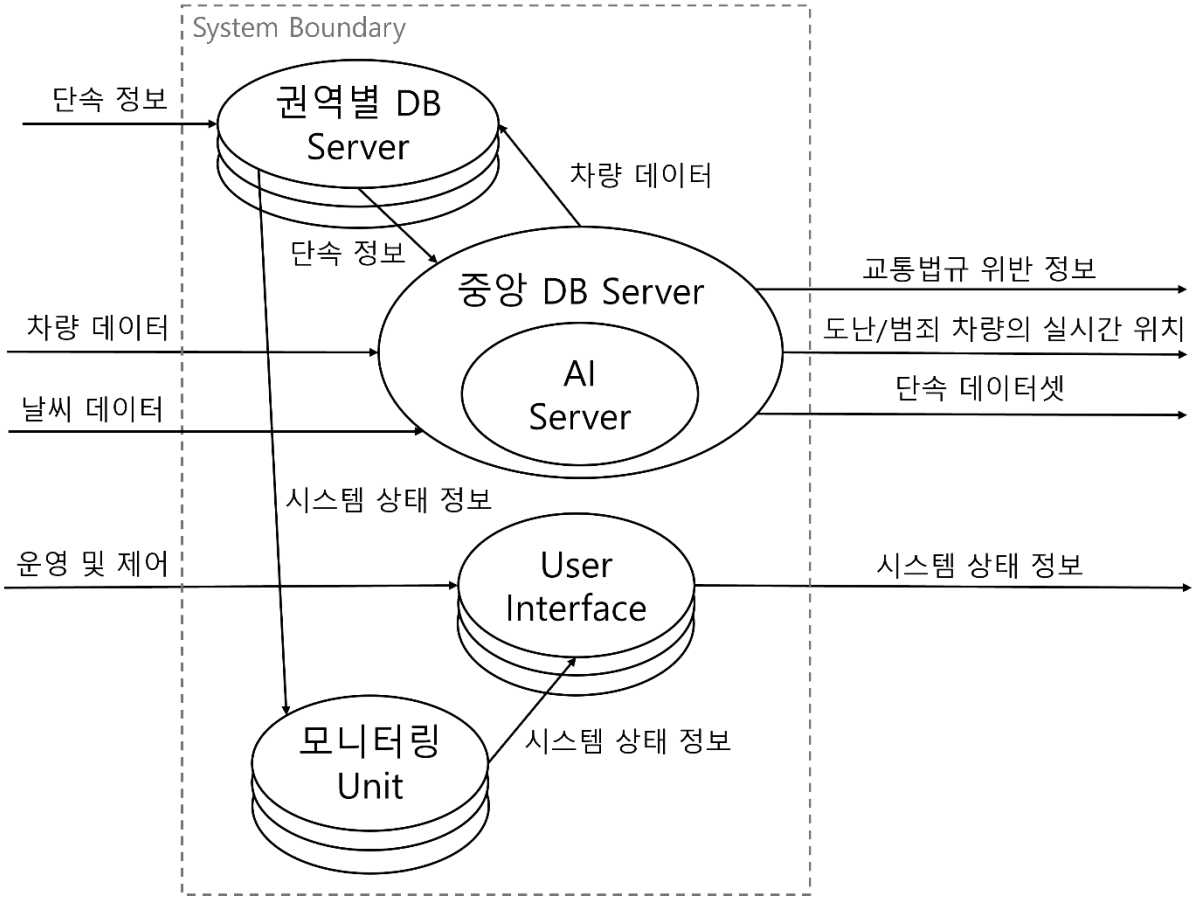
2.3. System Feature List

ID	Title	Description	I	Related Business Goal ID
SF-01	실시간 단속 정보 제공	각 단속카메라 시스템으로부터 실시간 단속 정보를 수집하고 이를 권역별 서버에서 처리하여 중앙 서버에 전송하는 기능	상	BG-01, BG-04
SF-02	과속 차량 자동 식별	규정된 속도를 초과하는 차량을 자동으로 식별하고 기록하는 기능	상	BG-03
SF-03	시스템 확장 대비	추후 시스템이 확장될 것을 대비하여 문서로 잘 기록해두고, 체계적으로 시스템을 구축해야 함.	상	BG-01, BG-02
SF-04	실시간 장비 모니터링	관리자는 모든 장비의 상태를 실시간으로 모니터링할 수 있음.	중	BG-05, BG-06, BG-12
SF-05	시스템 관리	관리자가 시스템 설정 (단속카메라 시스템의 추가, 변경, 삭제 등)을 변경하고 소프트웨어 업데이트를 수행할 수 있음.	중	BG-05
SF-06	자동 백업 및 복구	시스템 데이터를 자동으로 백업하고 필요 시 복구할 수 있는 기능	중	BG-01, BG-05
SF-07	교통 법규 위반 자동 분류	단속된 위반 사항을 자동으로 분류하고 관련 법규에 따라 처리함.	중	BG-03
SF-08	데이터 익명화 및 공유	수집된 데이터를 익명화하여 연구 및 분석 목적으로 안전하게 공유할 수 있음.	상	BG-12
SF-09	머신러닝 기반 예측 분석	과거 데이터를 기반으로 미래의 교통 패턴과 위반 가능성을 예측하는 기능	중	BG-12
SF-10	다중 언어 지원	시스템 인터페이스와 보고서를 다양한 언어로 제공함.	하	BG-01, BG-02, BG-05
SF-11	국제 표준 준수	시스템이 국제 교통 안전 및 데이터 보안 표준을 준수함.	중	BG-01, BG-05
SF-12	실시간 번호판 인식	AI를 활용하여 실시간으로 차량 번호판을 인식하고 기록하는 기능	상	BG-04
SF-13	실시간 교통 모니터링	교통 흐름과 사고를 실시간으로 감지하고 모니터링하는 기능	상	BG-03
SF-14	실시간 데이터베이스 동기화	중앙 데이터베이스와 지역 시스템 간의 실시간 데이터 동기화 기능	중	BG-04, BG-08
SF-15	AI 기반 이미지 분석	딥러닝 알고리즘을 사용하여 차량 이미지를 자동으로 분석하는 기능	상	BG-03
SF-16	신속한 차량 추적	범죄 차량을 인식하고 실시간으로 추적하여 차량이 인식될 경우 1분 내로 정보를 제	상	BG-04

		공함.		
SF-17	실시간 알림 시스템	문제 발생 시 관리자에게 즉시 알림을 제공하는 기능	상	BG-05
SF-18	**실시간 기상 정보 통합**	외부 기상 API와 연동하여 실시간 기상 조건을 기록하고, 단속 데이터와 연계하는 기능	중	BG-01, BG-12
SF-19	데이터 암호화 및 보안	모든 중요 데이터는 암호화되어 안전하게 전송되고 보관됨.	상	BG-10
SF-20	**차량 데이터 업데이트**	차량 및 차량주 정보를 주기적으로 경찰청을 통해 업데이트 하는 기능	상	BG-03, BG-04, BG-09, BG-10
SF-21	법원 연계 시스템	단속 데이터를 법원 시스템과 자동으로 연계하여 법적 처리 과정을 간소화함.	상	BG-09, BG-10
SF-22	장비 문제 식별	시스템 장비에서 발생한 문제를 1시간 이내에 자동으로 인식하고 분류함.	상	BG-11
SF-23	유지보수 요청 자동화	자동으로 유지보수 요청을 생성하고 업체에 통지함.	상	BG-11
SF-24	장비 공급 계약 관리	장비 공급업체와의 계약 조건, 가격, 거래 기록 등을 관리함.	하	BG-06
SF-25	장비 공급망 최적화	공급망을 분석하여 장비 공급의 효율성을 높임.	하	BG-06
SF-26	데이터셋 자동 생성	데이터 분석 업체에게 제공할 주기적인 데이터셋을 자동 생성함.	중	BG-12
SF-27	충분한 서버 용량	서버 용량이 충분하여 데이터를 저장함에 있어서 문제가 발생하지 않도록 함.	중	BG-01, BG-02, BG-06
SF-28	관리자 피드백 시스템	관리자가 개발자에게 피드백을 전달할 수 있는 기능	하	BG-01, BG-02, BG-05
SF-29	교통법규 위반 정보 처리	인공지능을 활용하여 수집한 교통법규 위반 정보 (위반 여부)를 처리함.	상	BG-03

* I: Importance << 상 중 하로 구분 >>

2.4. Domain Model



2.5. Assumptions about the System

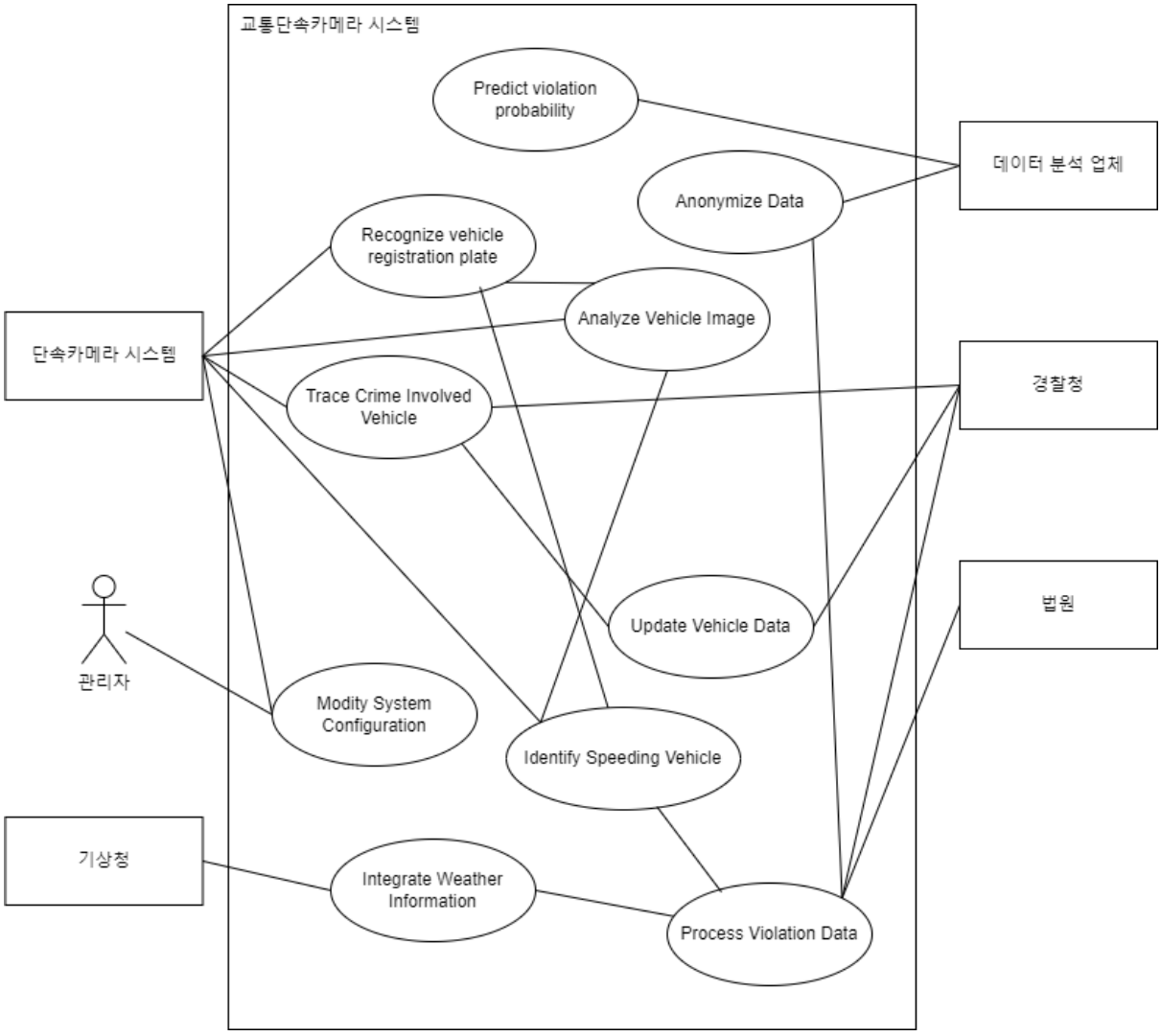
- 본 시스템은 민간 업체에서 개발한 시스템으로, 경찰청 서버에 데이터를 제공한다고 가정한다.

3. Architectural Drivers

3.1. Primary Functionality

3.1.1. Use Case Diagram

ss



3.1.2. Actor List

Name	Description
관리자	시스템을 업데이트하거나, 연결된 단속카메라 시스템의 구성을 변경하는 등의 관리 업무를 함.

단속카메라 시스템	교통법규 위반을 감지하여 시스템에 단속 대상 차량의 이미지 데이터를 제공함.
데이터 분석 업체	수집된 데이터를 분석하여 빈번하게 사고 발생하는 지역이나 시간대를 예측함.
경찰청	시스템이 처리한 데이터를 제공받아 벌금을 부과하고 운전자에게 알림.
법원	시스템이 처리한 데이터를 제공받아 법적 공방에서 증거로 활용함.

3.1.3. Use Case List

ID	Title	Summary of Description	Priority		System Feature ID	ASR?
			I	D		
UC-01	Predict Violation Probability	교통법규 위반 발생 가능성 예측	하	상	SF-09	X
UC-02	Recognize Vehicle Registration Plate	차량 번호판 식별 및 판독	상	중	SF-12	O
UC-03	Anonymize Data	개인 식별 정보 제거 또는 가림	하	하	SF-08	X
UC-04	Analyze Vehicle Image	교통법규 위반 차량의 이미지 처리 및 분석	상	상	SF-15	O
UC-05	Trace Crime Involved Vehicle	범죄와 연관된 차량 추적	중	중	SF-16	X
UC-06	Update Vehicle Data	차량 정보 수정 및 갱신	상	하	SF-20	X
UC-07	Modify System Configuration	단속카메라 시스템의 구성 변경	중	중	SF-05	X
UC-08	Identify Speeding Vehicle	구간단속 시 규정 속도를 위반하는 차량 식별	상	하	SF-02	O
UC-09	Integrate Weather Information	현재 날씨 데이터를 시스템에 통합	하	상	SF-18	O
UC-10	Process Violation Data	차량 이미지로부터 위반 정보 분석 및 처리	상	상	SF-29	O

* I: Importance (Business 관점) D: Difficulty (Techniques 관점) << 상 중 하로 구분 >>

3.1.4. UC-01: Predict Violation Probability

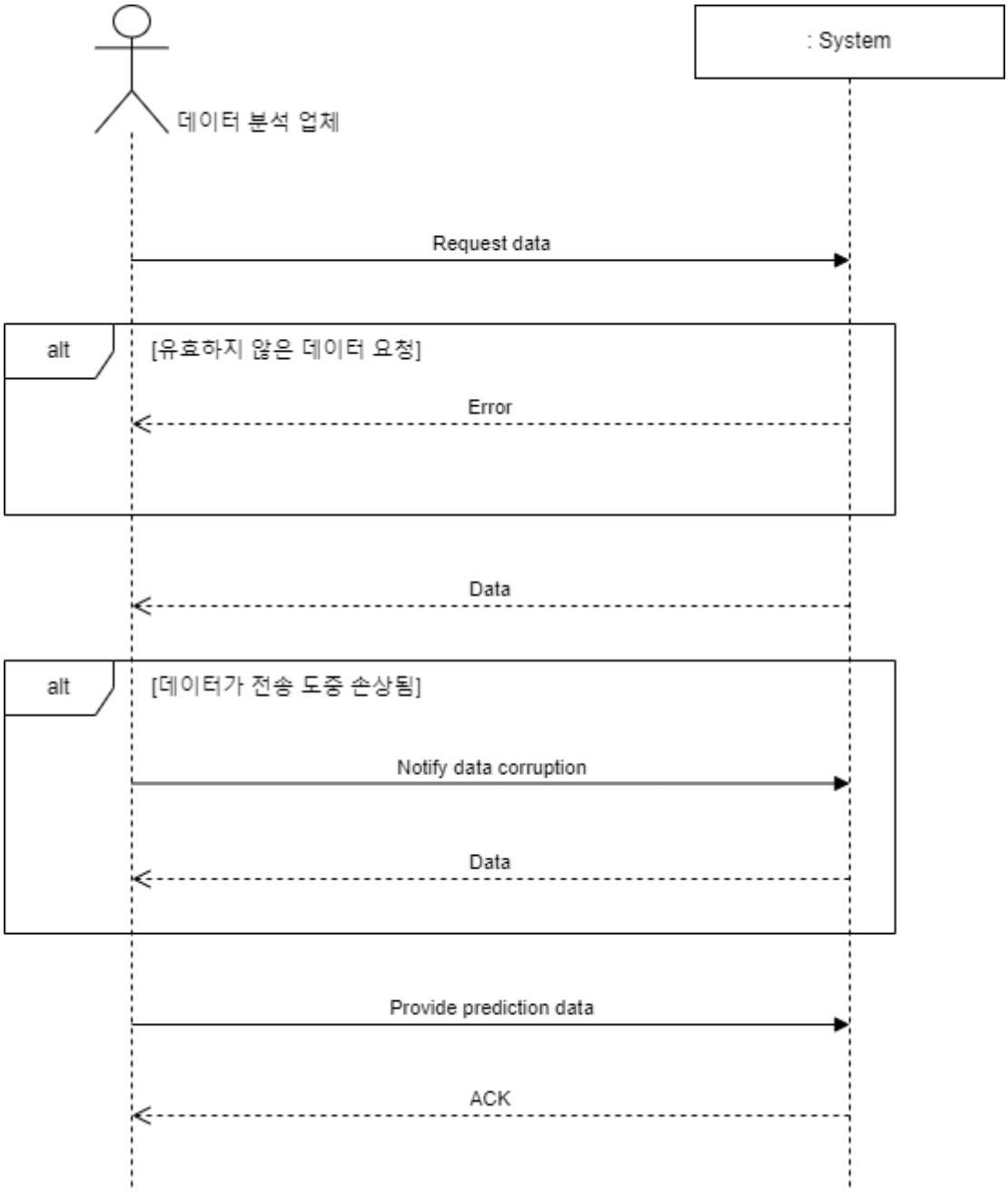
3.1.4.1. Scenario List

Scenario Title	Scenario Description
위반 가능성 예측	데이터의 분석을 통해 교통법규 위반이 자주 발생하는 구간을 식별하여 추후 해당 구간에서의 위반 가능성을 예측한다.
유효하지 않은 데이터 요청	유효하지 않은 데이터 요청 시 에러를 반환한다.
데이터가 전송 도중 손상됨	데이터가 전송 도중 손상되면 데이터를 재전송한다.

3.1.4.2. Use Case Description

Use Case	Predict violation probability
Actor	데이터 분석 업체
Description	데이터를 분석하여 교통법규 위반 발생 가능성을 예측
Stakeholders	데이터 분석 업체
Preconditions	예측을 할 수 있을 정도로 충분한 데이터가 쌓인 상태여야 한다.
Main Scenario	(A) : 데이터 분석 업체, (S) : System 1. (A) 데이터 분석 업체는 API를 통해 시스템에 데이터를 요청한다. 2. (S) 시스템은 데이터 분석 업체가 요청한 데이터를 제공한다. 3. (A) 데이터 분석 업체는 제공받은 데이터를 기반으로 특정 구간에서의 위반 발생 가능성을 예측한다. 4. (A) 예측한 데이터를 시스템에 제공한다.
Alternative Scenario	[유효하지 않은 데이터 요청] 1. (A) 데이터 분석 업체에서 API를 통해 유효하지 않은 데이터를 요청한다. 2. (S) 시스템은 에러를 반환하여 유효하지 않은 데이터를 요청했음을 알린다. [데이터가 전송 도중 손상됨] 3. (A) 손상된 데이터를 제공받았음을 시스템에게 알린다. 4. (S) 시스템은 해당 데이터를 재전송한다.

3.1.4.3. System Sequence Diagram



System Interface / Operation	**Description**
Request data	데이터 분석 업체가 시스템에 데이터를 요청한다.
Notify data corruption	전송받은 데이터가 전송 도중에 손상되었음을 알린다.
Provide prediction data	예측한 데이터를 시스템에 제공한다.

3.1.5. UC-02: Recognize Vehicle Registration Plate

3.1.5.1. Scenario List

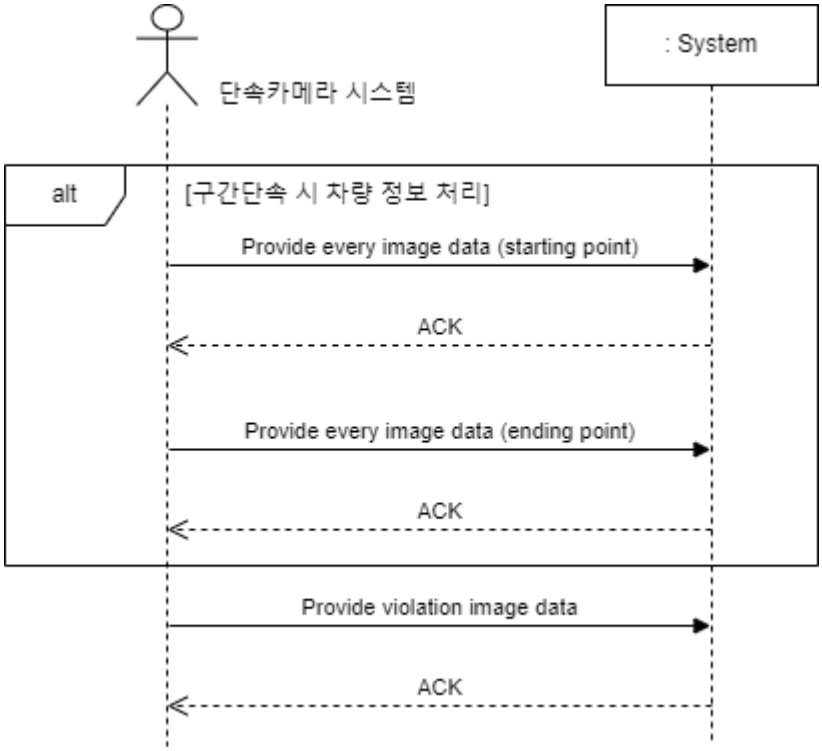
Scenario Title	Scenario Description
차량 번호판 인식 및 기록	단속카메라 시스템이 제공하는 이미지 데이터로부터 차량 번호를 추출하여 DB에 저장된 정보와 비교하고, 차량 번호가 유효하다면 이를 기록한다.
구간단속 시 차량 정보 처리	구간단속을 수행하는 단속카메라 시스템에서 제공하는 이미지 데이터를 처리하여 차량 번호를 추출하고, DB에 저장된 정보와 비교하여 차량 번호가 유효하다면 이를 기록한다.
유효하지 않은 이미지 데이터	제공받은 이미지 데이터가 유효하지 않다면 해당 데이터는 폐기한다.

3.1.5.2. Use Case Description

Use Case	Recognize Vehicle Registration Plate
Actor	단속카메라 시스템
Description	차량 번호판을 식별 및 판독하여 기록
Stakeholders	경찰청
Preconditions	제공받은 차량 데이터를 가지고 있어야 한다.
Main Scenario	(A) : 단속카메라 시스템, (S) : System 1. (A) 단속카메라 시스템에서 교통법규를 위반했다고 판단한 차량의 이미지 데이터를 권역별 DB에 전송한다. 2. (S) 시스템은 전송받은 이미지 데이터로부터 차량번호를 추출한다. 3. (S) 권역별 DB 서버는 중앙 DB 서버로부터 입력 받은 차량 번호 데이터를 바탕으로 단속카메라 시스템으로부터 입력 받은 이미지 데이터의 차량 번호와 비교하여 이것이 유효한 데이터인지 확인한다. 4. (S) 유효한 차량번호인 경우, 이를 권역별 DB 서버와 중앙 DB서버에 기록한다.
Alternative Scenario	[구간단속 시 차량 정보 처리] 1. (A) 구간단속을 수행하는 단속카메라 시스템이 시작 지점에서의 모든 이미지 데이터를 전송한다. 2. (A) 구간단속을 수행하는 단속카메라 시스템이 종료 지점에서의 모든 이미지 데이터를 전송한다.

	3. (S) 권역별 DB 서버는 전송받은 이미지 데이터로부터 차량 번호를 추출한다. 4. (S) 권역별 DB 서버는 중앙 DB 서버로부터 입력 받은 차량 번호 데이터를 바탕으로 단속카메라 시스템으로부터 입력 받은 이미지 데이터의 차량 번호와 비교하여 이것이 유효한 데이터인지 확인한다. 4. (S) 유효한 데이터인 경우, 이를 권역별 DB에 기록한다. [유효하지 않은 이미지 데이터] 4. (S) 제공받은 이미지 데이터로부터 추출한 차량 번호가 유효하지 않은 번호라고 판별한다. 5. (S) 해당 데이터를 폐기한다.

3.1.5.3. System Sequence Diagram



System Interface / Operation	Description
Provide every image data	시작 지점과 끝 지점에서의 모든 이미지를 제공한다.
Provide violation image data…	교통법규를 위반했다고 판단한 차량의 이미지를 제공한다.

3.1.6. UC-03: Anonymize Data

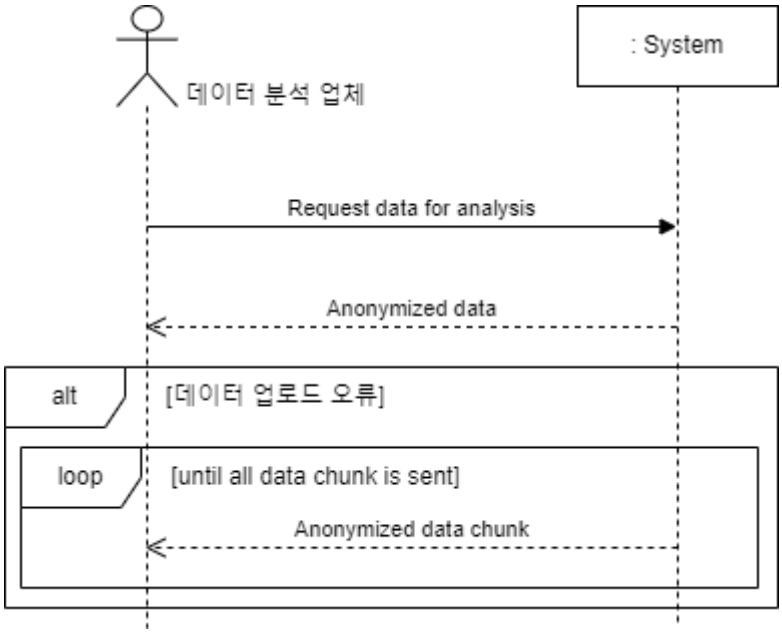
3.1.6.1. Scenario List

Scenario Title	Scenario Description
데이터 익명화	개인정보 데이터를 제거하여 익명화한다.
데이터 업로드 오류	요청된 데이터 크기가 커서 정상적으로 제공되지 않았을 경우 데이터를 나누어 처리한다.

3.1.6.2. Use Case Description

Use Case	Anonymize Data
Actor	데이터 분석 업체
Description	수집된 개인 정보를 제거하거나 가려 데이터를 익명화하는 기능
Stakeholders	데이터 분석 업체
Preconditions	시스템에 데이터 분석 업체에 제공할 수 있을 만큼 충분한 데이터가 존재해야 한다.
Main Scenario	(A) : 데이터 분석 업체, (S) : System 1. (A) 데이터 분석 업체는 분석을 위한 데이터를 요청한다. 2. (S) 요청받은 데이터를 확인하여 익명화 여부를 판단한다. 3. (S) 익명화한 데이터를 데이터 분석 업체에 제공한다.
Alternative Scenario	[데이터 업로드 오류] 3. (S) 데이터 제공 시 오류가 발생하면, 데이터를 여러 청크로 나누는 과정을 거친 후 제공한다.

3.1.6.3. System Sequence Diagram



System Interface / Operation	Description
Request data for analysis	데이터 분석 업체에서 분석을 위한 익명화된 데이터를 요청한다.

3.1.7. UC-04: Analyze Vehicle Image

3.1.7.1. Scenario List

Scenario Title	Scenario Description
차량 이미지 분석	입력 받은 이미지로부터 차량 이미지를 분석한다.
이미지 품질 불량	저화질 또는 손상된 이미지를 개선하여 분석을 시도한다.
다중 차량 감지	한 이미지에서 여러 대의 차량을 감지하고 개별적으로 분석한다.

3.1.7.2. Use Case Description

Use Case	Analyze Vehicle Image
Actor	단속카메라 시스템
Description	교통법규 위반 차량의 이미지 처리 및 분석
Stakeholders	경찰청
Preconditions	단속카메라 시스템과의 연결이 정상적으로 이루어지고 있어야 한다.
Main Scenario	(A) : 단속카메라 시스템, (S) : System 1. (A) 단속카메라 시스템에서 교통법규를 위반했다고 판단한 차량의 이미지 데이터를 권역별 DB에 전송한다. 2. (S) 권역 DB에서는 입력된 이미지 데이터를 중앙 DB로 전송한다. 3. (S) 중앙 DB의 AI 서버에서는 입력된 이미지 데이터를 분석하여 충분한 품질의 이미지인지 판단한다. 4. (S) 이미지 품질이 분석하기에 충분한 경우, 이미지 분석을 진행한다. 5. (S) 분석한 이미지 데이터를 DB에 저장한다.
Alternative Scenario	[이미지 품질 불량] 3. (S) 시스템이 이미지 품질이 분석하기에 부적합함을 감지한다. 4. (S) 이미지 개선 작업을 진행한다. 5. (S) 개선된 이미지를 분석하여 충분한 품질의 이미지인지 판단한다. 6. (S) 개선 후에도 분석이 불가능한 경우, 분석 불가한 이미지로 분석을 마무리한다. [다중 차량 감지] 4. (S) 시스템이 이미지에서 여러 대의 차량을 감지한다. 5. (S) 각 차량을 개별적으로 분리하여 분석한다. 6. (S) 각 차량에 대한 분석한 이미지 데이터를 별도로 저장한다.

3.1.7.3. System Sequence Diagram

System Interface / Operation	Description
Provide violation image data	단속카메라 시스템에서 교통법규를 위반했다고 판단한 차량의 이미지 데이터를 제공한다.

3.1.8. UC-05: Trace Crime Involved Vehicle

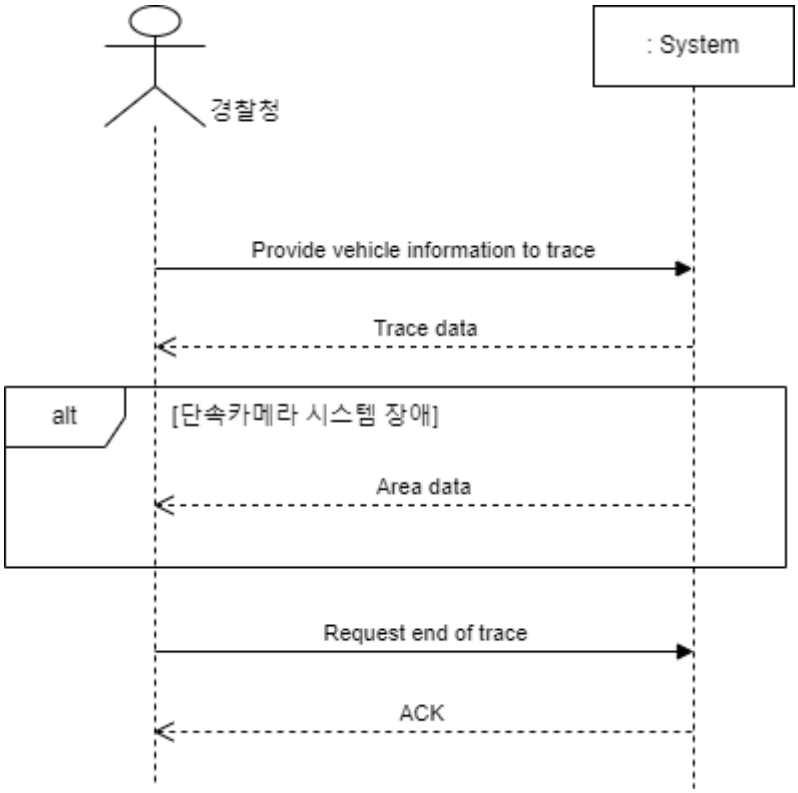
3.1.8.1. Scenario List

Scenario Title	Scenario Description
범죄 연루 차량 추적	경찰청으로부터 제공받은, 범죄와 연관된 차량을 추적한다.
단속카메라 시스템 장애	감시카메라 시스템에 장애가 발생하여 추적이 불가능한 경우 구역 정보를 제공한다.

3.1.8.2. Use Case Description

Use Case	Trace Crime Involved Vehicle
Actor	경찰청
Description	범죄와 연관된 차량 추적
Stakeholders	경찰청
Preconditions	제공받은 차량 데이터를 가지고 있어야 한다. 경찰청에서 요청하는 차량 정보는 유효한 차량 정보이다.
Main Scenario	(A) : 경찰청, (S) : System 1. (A) 경찰청에서 추적을 원하는 차량 정보를 시스템에 제공한다. 2. (S) 입력된 차량 정보를 기반으로 중앙 DB에서 요청된 기간 동안의 차량의 위치와 시간을 추출한다. 3. (S) 추출된 정보를 경찰청에 전송한다. 4. (S) 지속적으로 차량의 이동 경로를 추적하여 실시간으로 경찰청에서 전송한다. 5. (A) 필요한 조치가 취해진 이후 추적 종료 요청을 보낸다.
Alternative Scenario	[단속카메라 시스템 장애] 4. (S) 감시카메라 시스템 장애가 있는 곳이 있어 실시간으로 추적이 불가능한 구역이 있는 경우 해당 구역 정보를 같이 전달한다.

3.1.8.3. System Sequence Diagram



System Interface / Operation	**Description**
Provide vehicle information to trace	경찰청에서 추적할 차량에 대한 정보를 제공한다.
Request end of trace	추적의 종료를 요청한다.

3.1.9. UC-06: Update Vehicle Data

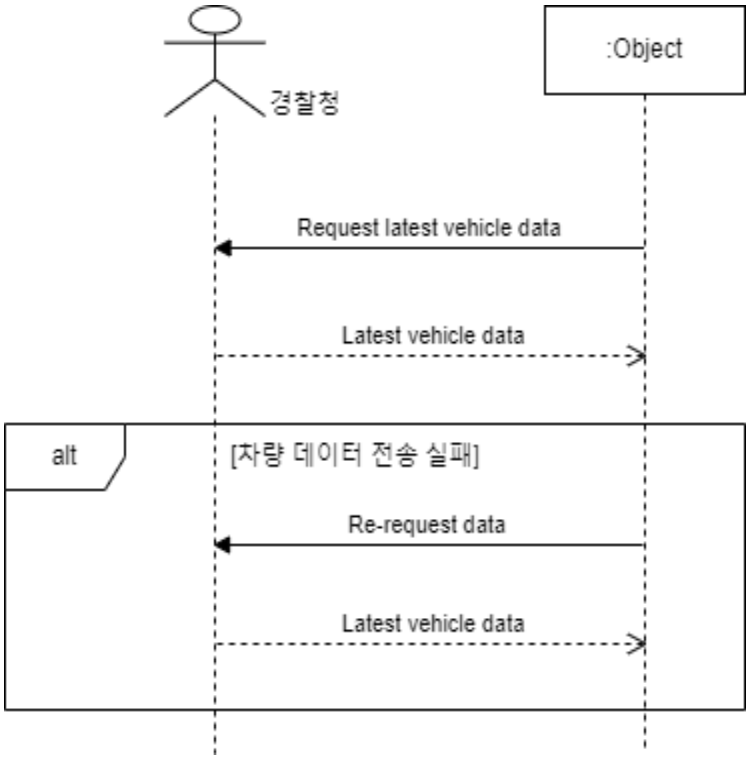
3.1.9.1. Scenario List

Scenario Title	Scenario Description
차량정보 갱신	경찰청으로부터 최신 차량 정보를 입력 받아 수정 및 갱신한다.
차량 데이터 전송 실패	경찰청에서 차량 데이터 전송에 실패하면 데이터를 재요청한다.
수신 데이터 오류	수신 데이터에 오류가 발생한 경우 해당 데이터를 재요청한다.

3.1.9.2. Use Case Description

Use Case	Update Vehicle Data
Actor	경찰청
Description	차량 정보 수정 및 갱신
Stakeholders	경찰청, 법원
Preconditions	경찰청은 최신 차량 정보를 가지고 있어야 한다.
Main Scenario	(A) : 경찰청, (S) : System 1. (S) 시스템은 일정 주기에 따라 경찰청에 최신 차량 정보 제공을 요청한다. 2. (A) 경찰청은 요청을 받아 시스템에 최신 차량 데이터를 전달한다. 3. (S) 시스템은 수신된 데이터를 검증한다. (누락 여부, 중복 데이터 확인 등) 4. (S) 시스템은 갱신된 데이터를 DB에 반영한다. 5. (S) 시스템은 주기적으로 이 과정을 반복하여 최신 차량 데이터를 유지한다.
Alternative Scenario	[차량 데이터 전송 실패] 2. (A) 경찰청이 요청을 받았지만, 장애로 인해 데이터를 시스템에 전송하지 못한다. 3. (S) 시스템은 일정 시간 응답이 없음을 감지하고, 오류 로깅을 한 후 경찰청에 재요청을 한다. 4. (S) 시스템은 이후 응답 결과에 따라 자동으로 재시도한다. [수신 데이터 오류] 4. (S) 시스템에 일부 데이터에 오류가 있음을 감지한다. 5. (S) 시스템은 오류 데이터를 기록하고, 그 외의 데이터를 업데이트한다. 6. (S) 시스템은 오류 데이터를 위해 다시 경찰청에서 요청한다.

3.1.9.3. System Sequence Diagram



System Interface / Operation	Description
…	

3.1.10. UC-07: Modify System Configuration

3.1.10.1. Scenario List

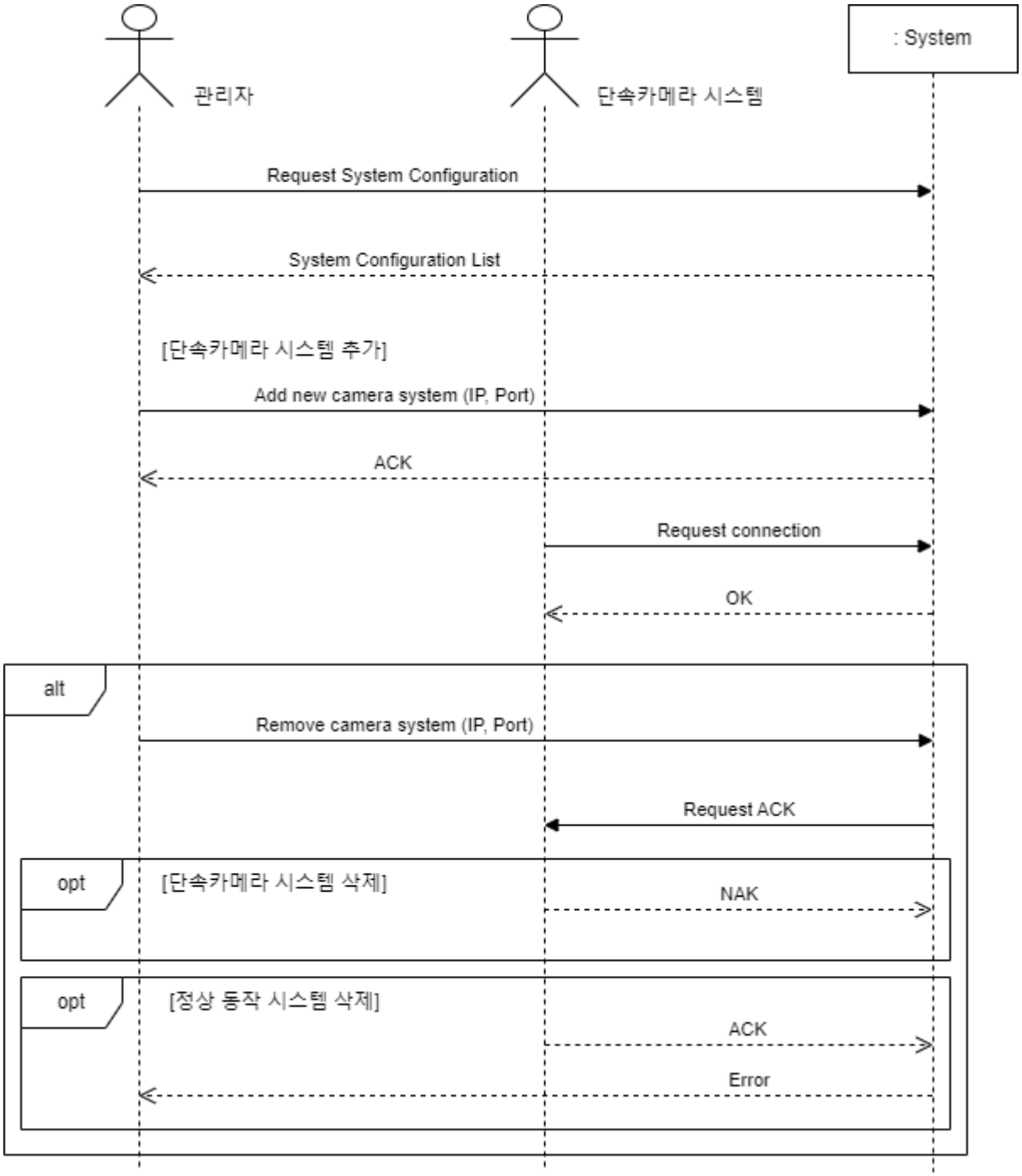
Scenario Title	Scenario Description
단속카메라 시스템 추가	관리자가 권역별 DB에 저장되어 있는 단속카메라 시스템의 구성에 새로운 단속카메라 시스템을 추가한다.
단속카메라 시스템 삭제	관리자가 권역별 DB에 저장되어 있는 단속카메라 시스템의 구성에서 더 이상 유효하지 않은 단속카메라 시스템을 삭제한다.
정상 동작 시스템 삭제	관리자가 정상적으로 동작하고 있는 단속카메라 시스템을 삭제하려고 하면, 경고 메시지를 띄운다.

3.1.10.2. Use Case Description

Use Case	Modify System Configuration
Actor	관리자, 단속카메라 시스템
Description	권역별 DB에 저장되어 있는 단속카메라 시스템의 구성 변경
Stakeholders	관리자
Preconditions	권역별 DB에는 정상적으로 동작하고 있는 단속카메라 시스템의 구성이 저장되어 있다.
Main Scenario	(A_1) : 관리자, (A_2) : 단속카메라 시스템, (S) : System 1. (A_1) 관리자는 자신이 관리하는 권역의 DB에 단속카메라 시스템의 구성을 요청한다. 2. (S) 권역 DB는 요청에 따라 단속카메라 시스템의 구성을 제공한다. 3. (A_1) 관리자는 새롭게 추가된 단속카메라 시스템의 추가를 요청한다. 4. (S) 시스템은 추가 요청이 들어온 단속카메라 시스템과의 연결 정보를 저장한다. 5. (A_2) 새로운 단속카메라 시스템이 연결을 요청한다. 6. (S) 시스템은 연결 요청을 수락한다.
Alternative Scenario	[단속카메라 시스템 삭제] 3. (A_1) 관리자는 더 이상 유효하지 않은 단속카메라 시스템의 삭제를 요청한다. 4. (S) 시스템은 삭제 요청이 들어온 단속카메라 시스템이 정말 유효하지 않은지 확인하기 위해 단속카메라 시스템에 ACK를 요청한다. 5. (S) 일정 시간동안 ACK가 도착하지 않는 경우, 유효하지 않은 단속카메라 시스템이라고 판단하여 이를 DB에서 삭제한다.

	[정상 동작 시스템 삭제] 3. (A_1) 관리자가 정상적으로 동작하는 단속카메라 시스템의 삭제를 요청한다. 4. (S) 시스템은 삭제 요청이 들어온 단속카메라 시스템이 유효한지 확인하기 위해 ACK를 요청한다. 5. (A_2) 시스템으로 ACK를 보낸다. 6. (S) ACK를 받으면, 관리자에게 경고 메시지를 띄운다.

3.1.10.3. System Sequence Diagram



System Interface / Operation	Description
Request System Configuration	관리자가 시스템에 단속카메라 시스템의 구성을 보여줄 것을 요청한다.
Add new camera system	관리자가 새로운 단속카메라 시스템을 추가할 것을 요청한다.
Remove camera system	관리자가 기존의 단속카메라 시스템의 삭제를 요청한다.
Request connection	단속카메라 시스템이 연결을 요청한다.

3.1.11. UC-08: Identify Speeding Vehicle

3.1.11.1. Scenario List

Scenario Title	Scenario Description
과속 차량 식별	구간단속을 수행하는 단속카메라 시스템으로부터 모든 차량의 이미지 데이터를 받아와 과속 여부를 판단하여 과속했다면 차량 정보를 식별하고 기록한다.
규정속도 준수	구간단속을 수행하는 단속카메라 시스템으로부터 모든 차량의 이미지 데이터를 받아와 과속 여부를 판단하여 규정 속도를 준수한 경우 해당 정보를 폐기한다.

3.1.11.2. Use Case Description

Use Case	Identify Speeding Vehicle
Actor	단속카메라 시스템
Description	규정 속도를 위반하는 차량 식별
Stakeholders	경찰청
Preconditions	단속카메라 시스템이 규정 속도를 위반하는 차량을 감지하여 촬영할 수 있어야 한다.
Main Scenario	(A) : 단속카메라 시스템, (S) : System 1. (A) 단속카메라 시스템은 규정 속도를 위반하는 차량을 감지하여 촬영하고, 시스템에 이미지 데이터를 제공한다. 2. (S) 시스템은 제공받은 이미지 데이터를 바탕으로 차량 정보를 추출하고 (UC-02 Main Scenario 2부터) 나면 과속 여부를 판별한다. 3. (S) 과속 차량이라고 판단되면, 이를 중앙 DB에 기록한다.
Alternative Scenario	[규정속도 준수] 3. (S) 규정속도를 준수한 차량이라고 판단되면, 해당 데이터는 폐기한다

3.1.11.3. System Sequence Diagram

System Interface / Operation	Description
Provide image data	시스템에 속도 위반 차량의 이미지를 제공한다.

3.1.12. UC-09: Integrate Weather Information

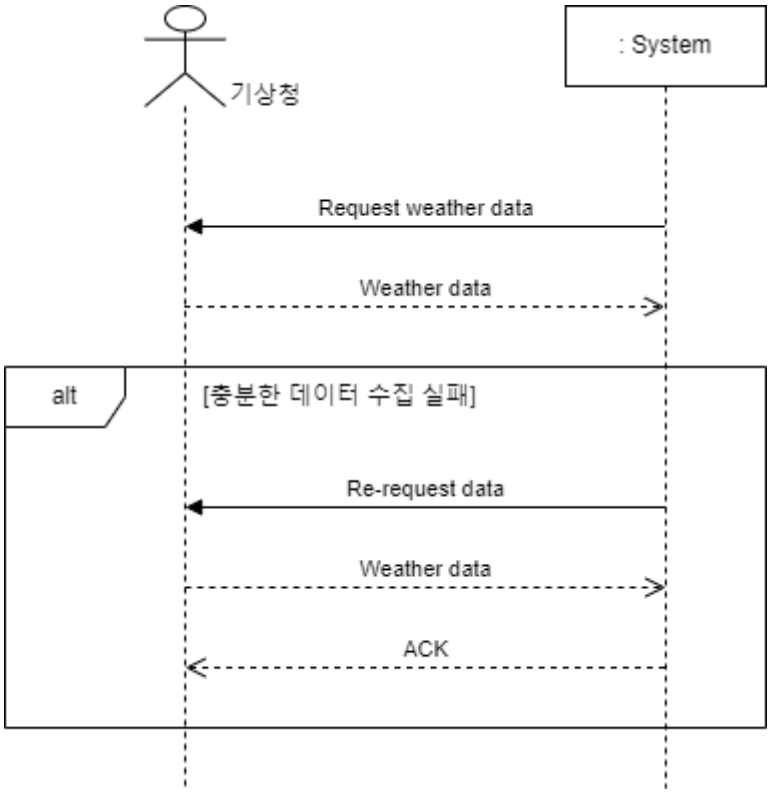
3.1.12.1. Scenario List

Scenario Title	Scenario Description
날씨 데이터 통합	날씨 데이터를 DB에 통합하여 AI가 수행하는 단속 데이터 분석에 활용한다.
충분한 데이터 수집 실패	날씨 데이터 수집에 실패하는 경우 AI를 통해 날씨 변화 추이를 분석하여 데이터 분석에 활용한다.

3.1.12.2. Use Case Description

Use Case	Integrate Weather Information
Actor	기상청
Description	날씨 데이터를 DB에 통합하여 AI가 수행하는 단속 데이터 분석에 활용
Stakeholders	데이터 분석업체, 개발자
Preconditions	1. 기상청으로부터 실시간 날씨 데이터를 받을 수 있는 API 연동이 설정되어 있어야 한다. 2. 교통 데이터베이스가 구축되어 있어야 한다.
Main Scenario	(A) : 기상청 (S) : System 1. (S) 시스템은 기상청에 주기적으로 날씨 데이터를 요청한다. 2. (A) 기상청은 요청받은 날씨 데이터를 제공한다. 3. (S) 시스템은 받은 날씨 데이터를 교통 데이터와 통합하여 저장한다. 4. (S) 시스템은 통합된 데이터를 AI를 활용하여 분석한다.
Alternative Scenario	[충분한 데이터 수집 실패] 3. (S) 시스템이 날씨 데이터 수신에 실패한 경우, 일정 시간 후 재시도한다. 4. (S) 여러 번의 시도 후에도 실패할 경우, AI를 통해 날씨 변화 추이를 분석한다.

3.1.12.3. System Sequence Diagram



System Interface / Operation	Description
…	

3.1.13. UC-10: Process Violation Data

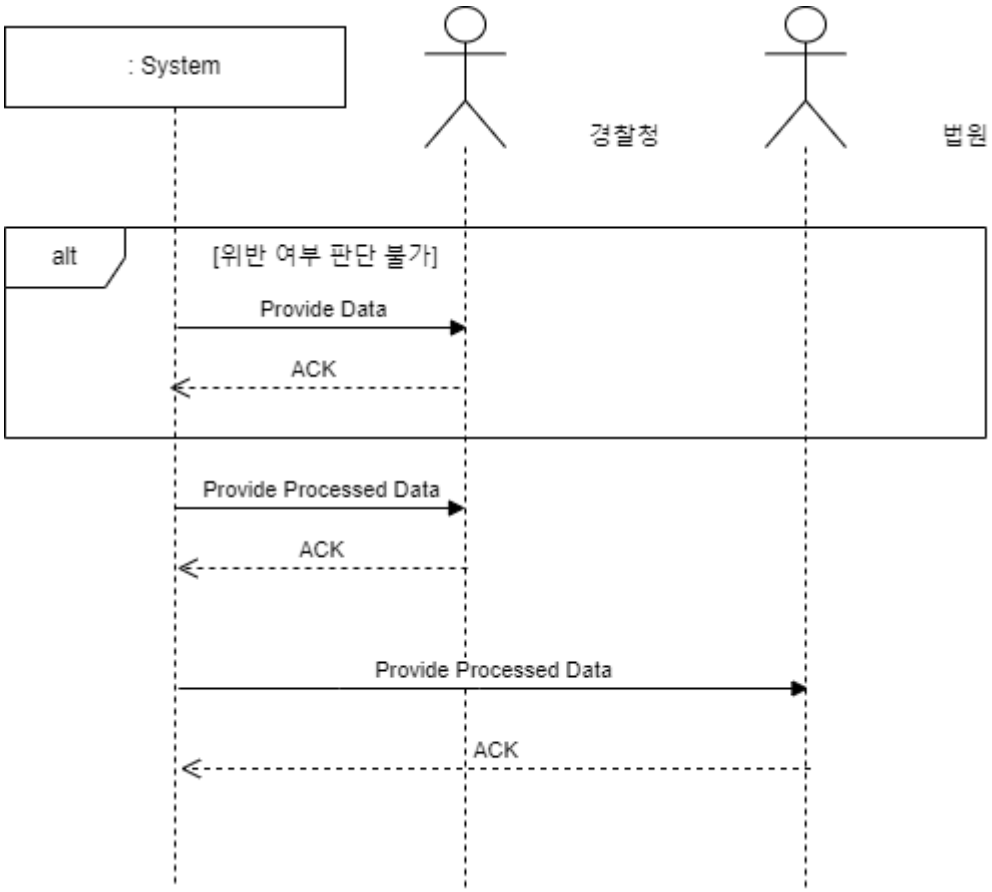
3.1.13.1. Scenario List

Scenario Title	Scenario Description
위반 여부 분석 및 처리	분석된 이미지로부터 교통법규 위반 정보를 판단하고 처리한다.
위반 여부 판단 불가	교통법규 위반 정보를 분석하지 못한다.

3.1.13.2. Use Case Description

Use Case	Process Violation Data
Actor	경찰청, 법원
Description	분석된 이미지로부터 교통법규 위반 정보를 판단하고 처리한다.
Stakeholders	경찰청
Preconditions	분석된 이미지가 시스템에 입력되어야 한다.
Main Scenario	(A_1) : 경찰청, (A_2) : 법원, (S) : System 1. (S) 시스템은 분석된 이미지를 가져와서 위반 정보를 판단한다. 2. (S) 위반했다고 판단되는 데이터를 가공한다. 3. (S) 경찰청과 법원에 가공한 데이터를 제공한다. 4. (A_1) 제공받은 데이터를 DB에 저장한다. 5. (A_2) 제공받은 데이터를 DB에 저장한다.
Alternative Scenario	[위반 여부 판단 불가] 2. (S) 위반 여부가 판단이 불가하다고 처리된 경우, 판단에 필요한 데이터를 함께 포함하여 경찰청에 전달한다. 3. (A_1) 제공받은 데이터를 DB에 저장한다.

3.1.13.3. System Sequence Diagram



System Interface / Operation	Description
…	

3.2. Quality Attribute Scenario

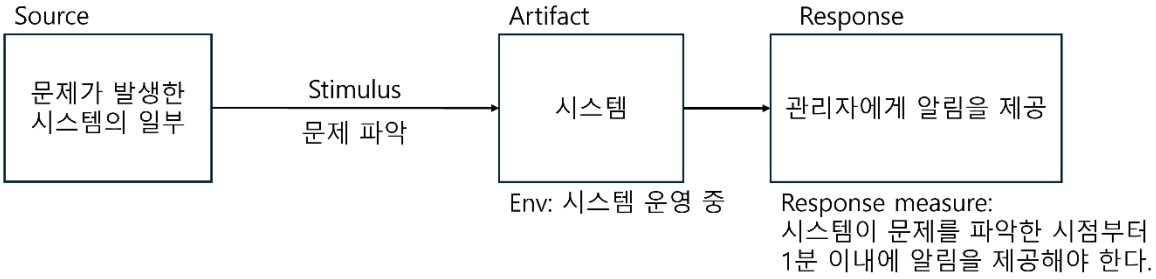
3.2.1. The QAS List

ID	Title	QA Type	Priority		Related Use Case	System Feature ID
			I	D		
QAS-01	문제 발생시 즉시 관리자에게 알림	Performance Efficiency (Time Behavior)	상	중	UC-07	SF-17
QAS-02	시스템에 새로운 기능 추가	Maintainability (Modifiability)	중	중	UC-07	SF-03
QAS-03	시스템 데이터의 복구	Reliability (Recoverability)	상	중	All UC	SF-06
QAS-04	다양한 언어 제공	Usability (Operability)	하	하	UC-07	SF-10
QAS-05	데이터 접근에 대한 로깅	Security (Accountability)	상	중	UC-06, 08, 10	SF-19

3.2.2. QAS-01: 문제 발생시 즉시 관리자에게 알림

Scenario Refinement: Performance Efficiency (Time behavior)

Raw Scenario : 문제 발생 시 관리자에게 즉시 알림을 제공한다.



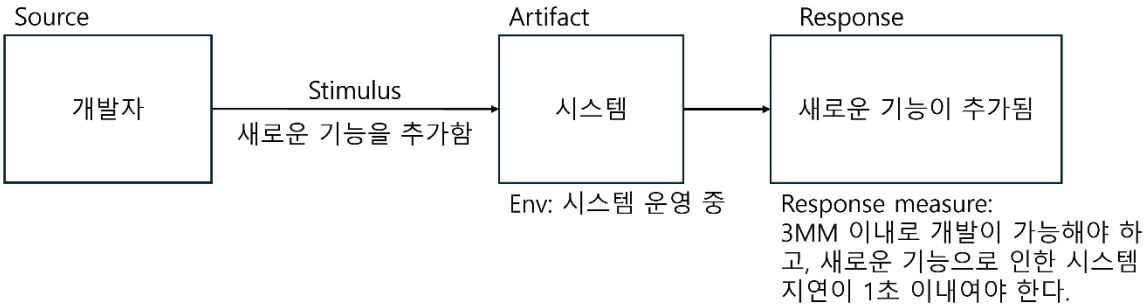
Refined Scenario : 시스템의 일부에 문제가 발생했을 때, 시스템이 문제를 파악하고, 파악한 시점으로부터 1분 이내에 관리자에게 알림을 제공해야 한다.

QA Type	Performance Efficiency (Time behavior)
Description	문제가 발생한 경우 이를 파악하여 관리자에게 즉시 알림을 제공해야 한다.
Source of Stimulus	문제가 발생한 시스템의 일부
Stimulus	문제 발생
Artifact	시스템
Environment	시스템 운영 중
Response	문제를 파악하여 관리자에게 알림을 제공
Response Measure	문제를 파악한 시점부터 1분 이내에 알림을 제공해야 한다.
Summary of QAS	시스템의 일부에 문제가 발생했을 때 파악한 시점으로부터 1분 이내에 관리자에게 알림을 제공해야 한다.

3.2.3. QAS-02: 시스템에 새로운 기능 추가

Scenario Refinement: Maintainability (Modifiability)

Raw Scenario : 시스템에 새로운 기능이 추가될 때 이로 인한 기능의 지연이 거의 없어야 하고 단시간 내에 개발할 수 있어야 한다.



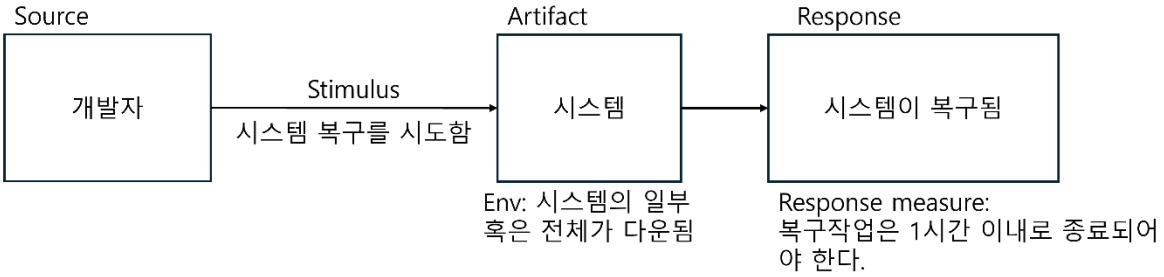
Refined Scenario : 개발자가 새로운 기능을 추가할 때, 3MM 이내로 개발이 가능해야 하고, 새로운 기능으로 인한 시스템 지연이 1초 이내여야 한다.

QA Type	Maintainability (Modifiability)
Description	시스템에 새로운 기능이 추가될 때 이로 인한 기능의 지연이 거의 없어야 하고 단시간 내에 개발할 수 있어야 한다.
Source of Stimulus	개발자
Stimulus	새로운 기능의 추가
Artifact	시스템
Environment	시스템 운영 중
Response	새로운 기능이 추가됨
Response Measure	3MM 이내로 개발이 가능해야 하고, 새로운 기능으로 인한 시스템 지연이 1초 이내여야 한다.
Summary of QAS	개발자가 새로운 기능을 추가할 때, 3MM 이내로 개발이 가능해야 하고, 새로운 기능으로 인한 시스템 지연이 1초 이내여야 한다.

3.2.4. QAS-03: 시스템 데이터의 복구

Scenario Refinement: Reliability (Recoverability)

Raw Scenario : 시스템이 다운되는 경우 복구해야 한다.



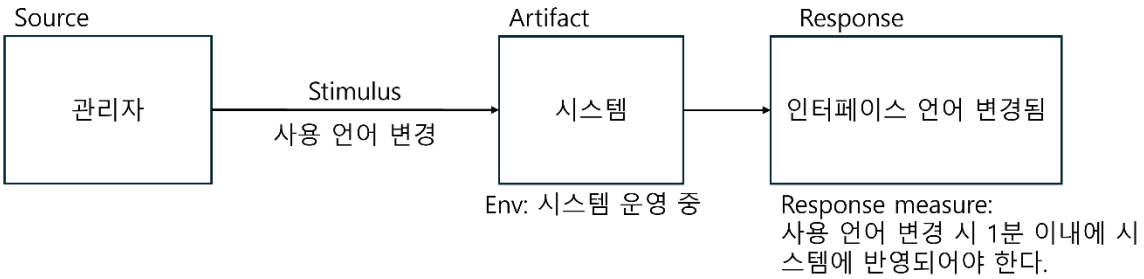
Refined Scenario : 개발자가 시스템의 복구를 시도할 때, 복구 작업은 1시간 이내로 종료되어야 한다.

QA Type	Reliability (Recoverability)
Description	시스템이 다운되는 경우 복구해야 한다.
Source of Stimulus	개발자
Stimulus	시스템 복구 시도
Artifact	시스템
Environment	시스템의 일부 혹은 전체가 다운됨
Response	시스템이 복구됨
Response Measure	복구작업은 1시간 이내로 종료되어야 한다.
Summary of QAS	개발자가 시스템의 복구를 시도할 때, 시스템의 복구 작업은 1시간 이내로 종료되어야 한다.

3.2.5. QAS-04: 다양한 언어 제공

Scenario Refinement: Usability (Operability)

Raw Scenario : 시스템 인터페이스와 보고서를 다양한 언어로 제공한다.



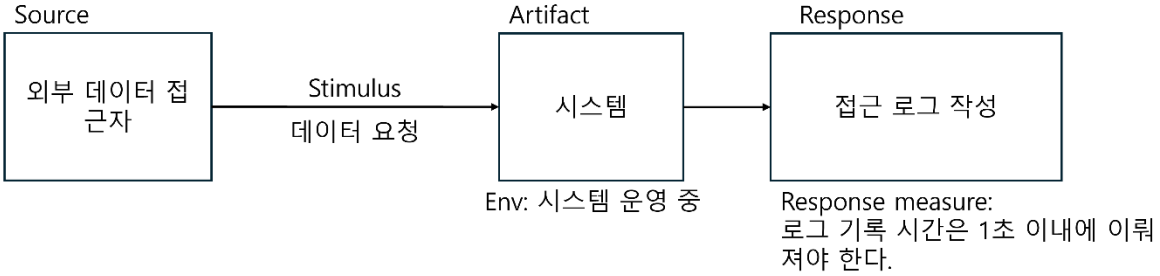
Refined Scenario : 관리자가 사용 언어의 변경을 시도하면, 시스템의 인터페이스 언어는 1분 이내에 변경되어야 한다.

QA Type	Usability (Operability)
Description	시스템 인터페이스는 다양한 언어를 지원해야 한다.
Source of Stimulus	관리자
Stimulus	사용 언어 변경
Artifact	시스템
Environment	시스템 운영 중
Response	인터페이스 언어 변경됨
Response Measure	사용 언어 변경 시 1분 이내에 시스템에 반영되어야 한다.
Summary of QAS	관리자가 사용 언어의 변경을 시도하면, 시스템의 인터페이스 언어는 1분 이내에 변경되어야 한다.

3.2.6. QAS-05: 데이터 접근에 대한 로깅

Scenario Refinement: Security (Accountability)

Raw Scenario : 외부에서 데이터에 접근한 모든 경우에 대해 로그를 남긴다.



Refined Scenario : 외부에서 데이터를 요청하는 경우, 시스템은 접근 로그를 1초 이내에 작성해야 한다.

QA Type	Security (Accountability)
Description	외부에서 데이터에 접근한 모든 경우에 대해 로그를 남긴다.
Source of Stimulus	외부 데이터 접근자
;/Stimulus	데이터 요청
Artifact	시스템
Environment	시스템 운영 중
Response	접근 로그 작성
Response Measure	로그 기록 시간은 1초 이내에 이뤄져야 한다.
Summary of QAS	외부에서 데이터를 요청하는 경우, 시스템은 접근 로그를 1초 이내에 작성해야 한다.

3.3. Constraint

3.3.1. Business Constraint List

ID	Title	Description
BC-01		
BC-02		

3.3.2. Technical Constraint List

ID	Title	Description
TC-01		
TC-02		

4. Architecture Design & Evaluation

4.1. Candidate Designs per QA

4.1.1. Candidate Design List

QA	QAS	Candidate Design	Candidate Design Approach (CDA)
QA1: Performance Efficiency (Time Behavior)	QAS-01	QA1_CD-01	QA1_CD-01_CDA-01: Centralized Processing Pattern **QA1_CD-01_CDA-02: Pipe-filter pattern** QA1_CD-01_CDA-03: Edge Aggregation Pattern
QA2: Maintainability (Modifiability)	QAS-02	QA2_CD-01	**QA2_CD-01-CDA-01: Microservices Architecture + Database per Service** QA2_CD-01-CDA-02: Layered Architecture + Encapsulation QA2_CD-01-CDA-03: Ports and Adapters + Dependency Injection
QA3: Reliability (Recoverability)	QAS-03	QA3_CD-01	QA3_CD-01_CDA-01: Event-Driven Architecture + Saga Pattern **QA3_CD-01_CDA-02: Active-Passive Architecture + State Replication** QA3_CD-01_CDA-03: Microservices Architecture + Circuit Breaker Pattern
QA4: Usability (Operability)	QAS-04	QA4_CD-01	**QA4_CD-01_CDA-01: UI Pattern + Command Pattern** QA4_CD-01_CDA-02: Model-View-Controller (MVC) Pattern QA4_CD-01_CDA-03: Observer Pattern + Strategy Pattern
QA5: Security (Accountability)	QAS-05	QA5_CD-01	QA5_CD-01_CDA-01: Role-Based Access Control + Immutable Log Pattern + Logging QA5_CD-01_CDA-02: Attribute-Based Access Control + Immutable Log Pattern + Logging **QA5_CD-01_CDA-03: Rule-Based Access Control + Immutable Log Pattern + Logging**

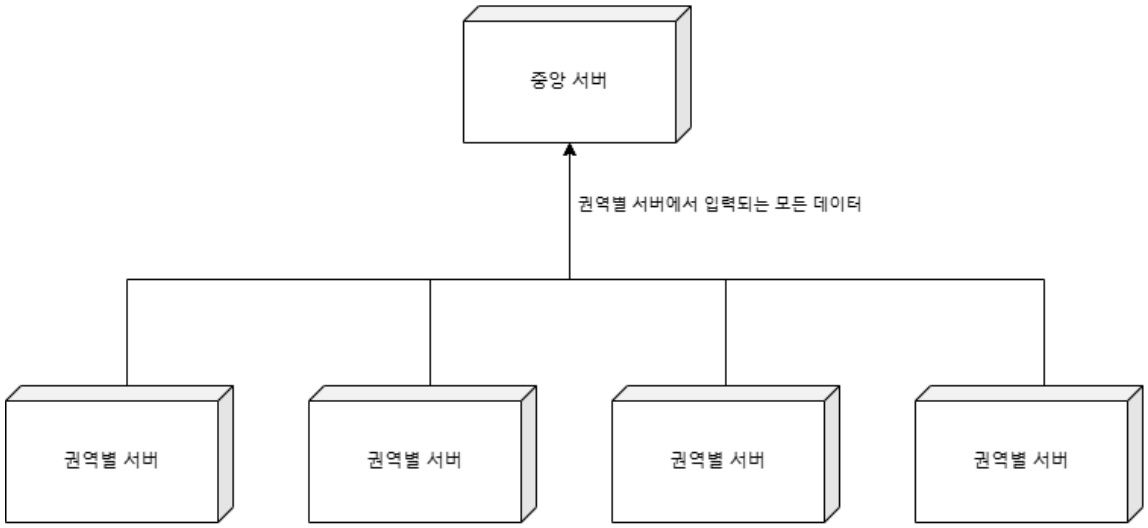
4.1.2. QA1: Performance Efficiency – Time Behavior

4.1.2.1. Design Goal

시스템은 시스템의 일부에 문제가 발생한 경우 이를 파악할 수 있어야 하며, 파악한 시점으로부터 1분 이내에 관리자에게 알림을 제공해야 한다. 이 외에도 권역별 DB 서버에서는 개별 단속카메라 시스템들로부터 데이터를 받으면 이를 실시간으로 처리할 수 있어야 하고, 이를 중앙 DB 서버로 전송하여 각종 정보를 만들어낸 후 외부 시스템에 제공할 수 있도록 해야 한다. 또한, 범죄에 연루된 차량의 경우 실시간으로 감시를 하고 있다가, 차량의 위치 정보와 같은 각종 정보가 얻어지는 경우 실시간으로 처리를 해야 한다.

4.1.2.2. Candidate Design Approach List

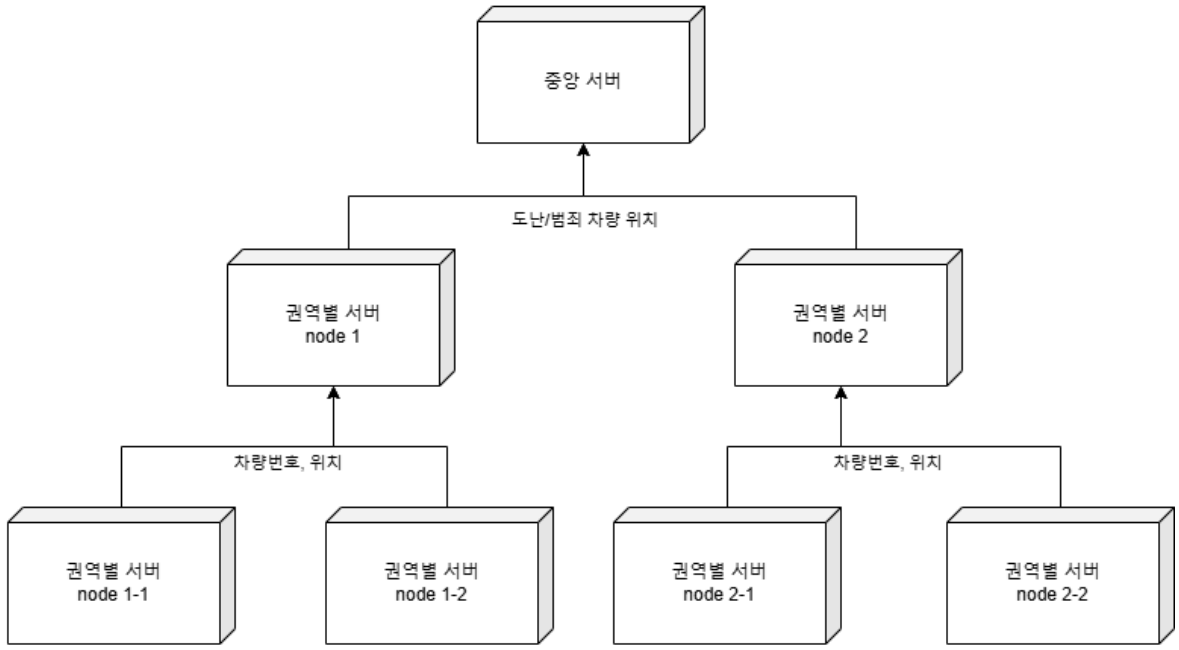
4.1.2.2.1. CDA #1 Description: Centralized Processing Pattern



CDA ID	*QA1_CD-01_CDA-01*
Description	각 권역별 서버는 단속카메라 시스템에서 받아온 모든 정상 데이터를 중앙DB에 실시간 전달한다. 중앙 DB는 각 권역별 서버로부터 받은 모든 데이터를 처리하여, 도난 및 범죄 차량을 특정하고 위치를 파악하여 실시간으로 경찰청 서버에 전달한다.
Pros	모든 작업이 중앙에서 통합적으로 처리되므로, 성능이 일관되고 예측 가능함. 또한 중앙 서버의 성능을 높게 구성하면 강력한 성능으로 처리속도를 올릴 수 있음.
Cons	모든 데이터가 중앙으로 몰리면서 중앙 서버의 과부하나 네트워크 문제가 발생할 수 있어, 실시간성이 충족되지 않을 수 있음.

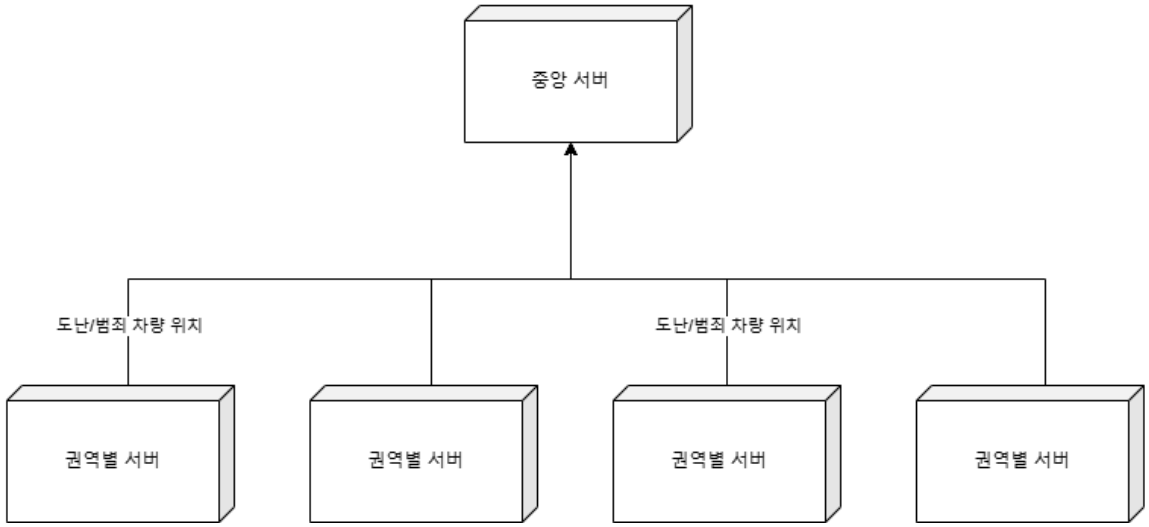
	데이터가 증가하면 중앙 서버가 병목되기 쉬워 처리 성능이 떨어질 수 있음.

4.1.2.2.2. CDA #2 Description: Pipe-filter pattern



CDA ID	*QA1_CD-01_CDA-02*
Description	권역별 서버 n-n 에서는 단속카메라 시스템에서 받아온 데이터를 1차 가공하여 차량 번호 및 위치정보로 필터링한다. 필터링 된 정보를 권역별 서버 n 에 전달하고 권역별 서버 n은 가지고 있는 도난 및 범죄 차량 데이터를 사용하여, 타겟 차량인지 여부를 판단 후, 타겟 차량의 데이터만 중앙 서버에 전달함.
Pros	중요한 데이터만 중앙으로 전송되기 때문에, 중앙 서버의 부담이 줄어서 응답 속도가 빠름. 불필요한 데이터가 중간에서 필터링 되어 사라지므로, 중앙 서버의 자원이 더욱 효율적으로 사용됨.
Cons	중간 과정의 필터링 단계에서 지연이 발생할 수 있고, 필터링이 비정상 동작 시 성능이 저하될 수 있음.

4.1.2.2.3. CDA #3 Description: Edge Aggregation Pattern



CDA ID	*QA1_CD-01_CDA-03*
Description	각 권역별 서버에서 도난 및 범죄 차량의 데이터를 가지고 있고, 각 권역별로 수집된 단속카메라 데이터를 가공하여, 도난 및 범죄 차량 여부를 판단함. 판단 이후 타겟 차량 데이터만 중앙 서버로 전달함.
Pros	엣지에서 대부분의 작업이 완료되고 중앙으로는 중요한 데이터만 전달되므로, 실시간 성능을 내기 좋음 엣지에서 처리된 데이터를 전달받는 중앙서버는 최소한의 작업만 수행하여, 응답이 빠름.
Cons	엣지 장치에서 충분한 성능을 제공해야, 속도가 나올 수 있음.

4.1.2.3. Decision and Rationale

Performance		**Analysis**	**CDA #1** Centralized Processing Pattern	**CDA #2 (selected)** Pipe-filter pattern	**CDA #3** Edge Aggregation Pattern
ID	**Title**				
QAS-01	문제 발생시 즉시 관리자에게 알림	Pros	(+) 중앙 서버의 성능을 높게 구성하여 높은 처리속도	(+) 중앙 서버의 응답 속도가 비교적 빠름. (+) 중앙 서버 자원 효율적 사용	(++) 중앙서버 응답이 빠름.
		Cons	(-) 중앙 서버 과부하 발생가능 (-) 네트워크 문제가 발생가능	(-) 필터링 단계에서 지연 발생 (-) 필터링 비정상 동작 시 성능이 저하	(-) 엣지 각각에 충분한 성능을 제공해야 함 (-) 엣지에서 지연 발생함

Candidate Design:

QA	QAS	CD	Description
QA1: Performance Efficiency (Time Behavior)	QAS-01	QA1_CD-01 (Pipe-filter pattern)	이미지 처리 같은 고성능의 컴퓨팅 자원을 모든 엣지에 제공하는 것은 현실적으로 어렵고, 또한 시간으로 수집되는 단속 데이터의 수가 매우 많기 때문에 전국의 모든 데이터를 안정적으로 한 곳에 전부 모아 처리하는 것은 어렵다. 각 1단계 서버에서 기본적인 이미지 처리를 통해 차량번호 등의 기본적인 데이터와 위치 데이터만을 남겨, 2단계 서버에 전달하고, 2단계 서버는 중앙서버로부터 받아온 도난 및 범죄 차량의 데이터와 1단계 서버에서 제공받은 데이터를 비교하여, 타겟인 차량 데이터만을 필터링한다. 필터링 된 정보만을 중앙 서버에 전달하고 중앙 서버는 전달받은 데이터를 경찰청 서버에 전달한다. 이렇게 구성하면 도난 및 범죄 차량의 데이터를 모든 권역별 서버에 동기화해야 하는 문제도 어느정도 해결하면서, 네트워크로 전달되는 데이터의 양도 적은 편이기 때문에 안정적이고 빠른 데이터 전달이 가능하다.

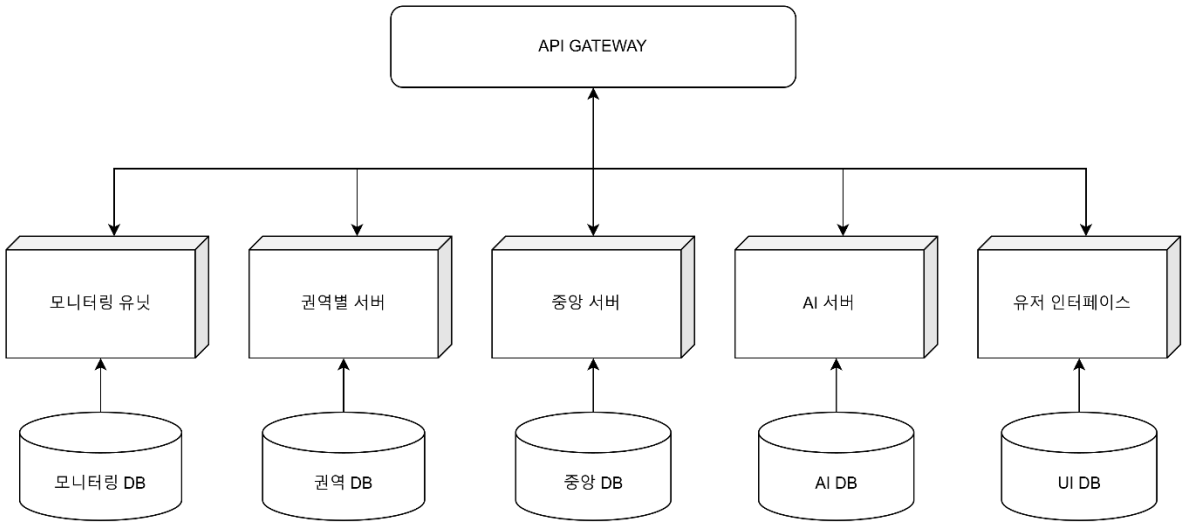
4.1.3. QA2: Maintainability – Modifiability

4.1.3.1. Design Goal

시스템에 새로운 기능을 추가하는 것이 용이해야 하고, 이것이 기존의 기능에 영향을 주지 않아야 한다. 또한, 새로운 기능이 추가될 때 이로 인한 기능의 지연이 거의 없어야 하고 단시간 내에 개발할 수 있어야 한다.

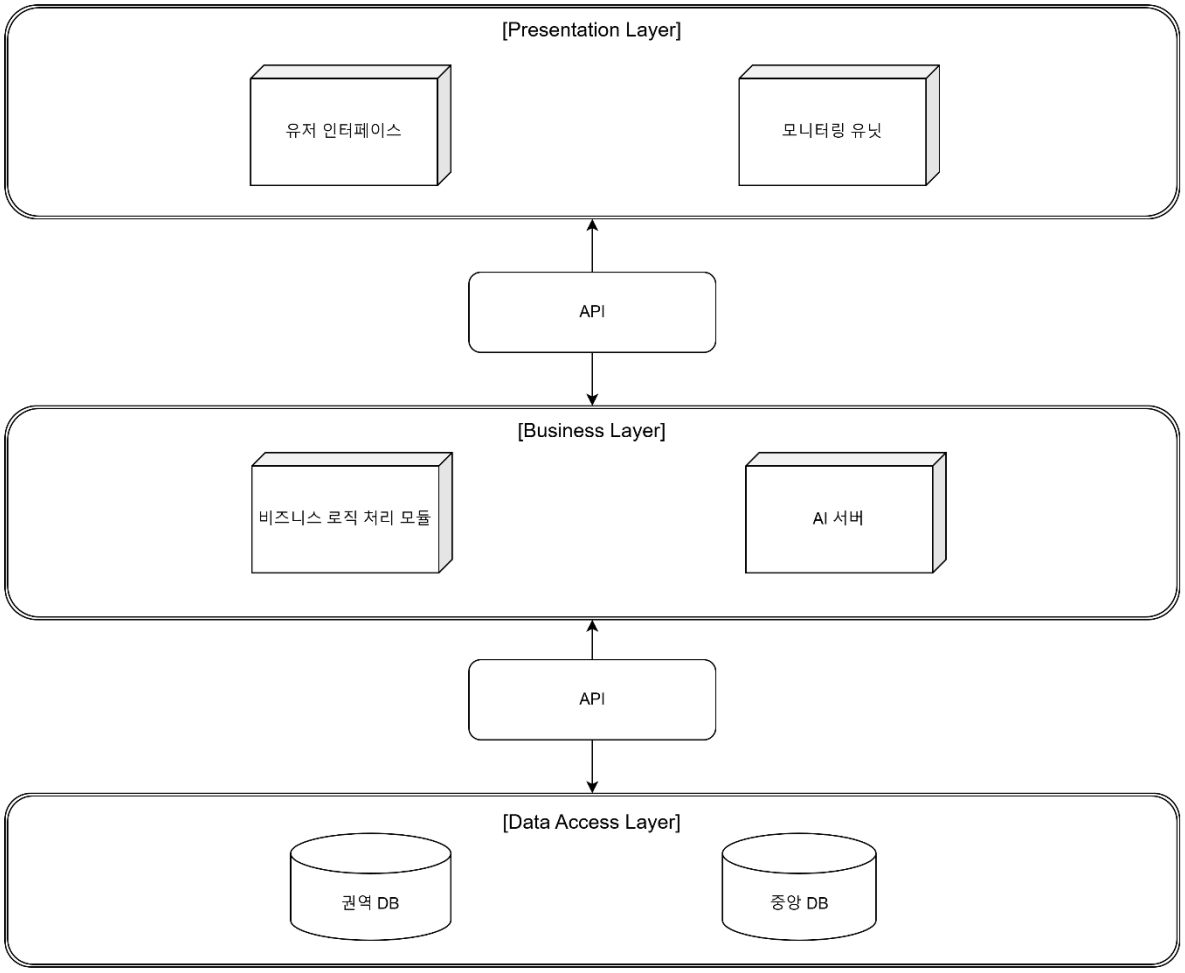
4.1.3.2. Candidate Design Approach List

4.1.3.3. CDA #1 Description: Microservices Architecture + Database per Service



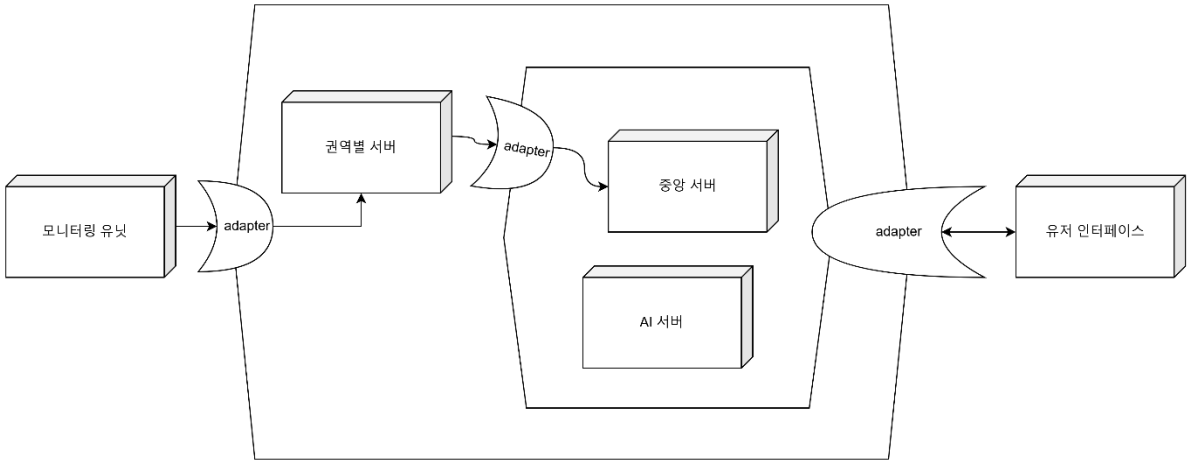
CDA ID	*QA2_CD-01-CDA-01*
Description	각 주요 기능을 모두 독립적인 서비스로 분리하여 독립적인 마이크로 서비스로 설계한다. 또한 각 서비스가 자체 데이터베이스를 가지게 하여, 데이터베이스가 해당 서비스 내에서만 사용되도록 만든다.
Pros	각 서비스가 데이터베이스를 자체적으로 가지고 있기 때문에, 각 서비스 별로 데이터를 독립적으로 관리할 수 있다. 한쪽의 데이터베이스 구조가 바뀌어도 다른 쪽은 아무런 영향이 없다. 또한, 추가적인 서비스가 필요한 경우 해당 서비스만 확장하고 필요한 부분만 쉽게 변경할 수 있다.
Cons	데이터 일관성 유지가 어렵기 때문에 시스템 전체 일관성 보장에 어려움이 있다.

4.1.3.3.1. CDA #2 Description: Layered Architecture + Encapsulation



CDA ID	*QA2_CD-01-CDA-02*
Description	시스템을 기능별로 계층화하여, 데이터 계층과 비즈니스 로직 계층, 표현 계층으로 나눈다. 또한 각 계층의 내부 구현을 캡슐화하여, 서로 다른 계층이 서로의 구현에 의존하지 않게 한다.
Pros	각 계층이 독립적으로 변경될 수 있고 유지보수 가능하기 때문에 변경 용이성이 높다. 또한 각 계층의 내부 구현을 숨기고 있기 때문에 서로 다른 계층에 미치는 영향을 줄일 수 있다. 계층간 결합도를 줄여서 기능 추가 및 수정이 용이하다.
Cons	큰 구조적 변경이 필요할 경우 개발 계층의 수정이 복잡해질 수 있다.

4.1.3.3.2. CDA #3 Description: Ports and Adapters + Dependency Injection



CDA ID	*QA2_CD-01-CDA-03*
Description	시스템의 주요기능들이 서로 직접 연결되지 않도록 각 기능이 포트를 통해 상호작용하게 하고, 구체적인 구현은 어댑터로 분리한다. 또한 각 포트가 어댑터를 외부에서 받아와서, 필요에 따라 어댑터를 쉽게 교체할 수 있게 한다.
Pros	외부에서 들어오는 어댑터만 교체하여 기능 및 시스템 간의 연결방식을 쉽게 변경할 수 있다. 각 기능 간의 결합도가 낮아져 새로운 외부 시스템의 추가나 기존 시스템 변경에 대응하기 쉽다. Dependency Injection을 통해 모의 객체를 사용하여 테스트할 수 있어 테스트가 용이하다.
Cons	각 기능에 대한 포트와 어댑터를 구성하고 관리하기 때문에 의존성 관리가 복잡하다.

4.1.3.4. Decision and Rationale

Performance		**Analysis**	**CDA #1 (selected)** Microservices Architecture + Database per Service	**CDA #2** Layered Architecture + Encapsulation	**CDA #3** Ports and Adapters + Dependency Injection
ID	**Title**				
QAS-02	시스템에 새로운 기능 추가	Pros	(+) *데이터베이스 독립관리 가능* *(++) 서비스 확장성*	(+) 계층별 변경 용이함 (+) 기능 추가 수정 쉬움	(+) 의존성 관리가 유연함 (+) 테스트 용이함
		Cons	(-) *데이터 일관성 어려움*	(-) 계층간 영향을 미치는 큰 변경은 복잡함	(-) 의존성 관리가 복잡해질 수 있음

Candidate Design:

QA	**QAS**	**CD**	**Description**

QA2: Maintainability (Modifiability)	QAS-02	QA2_CD-01 *(*Microservices Architecture + Database per Service*)*	하위 계층의 변경이 상위 계층에 영향을 줄 수도 있고, 기능별로 독립된 서비스를 만들기 어렵기 때문에 CD2는 적합하지 않다. 또한 CD3의 경우 각 시스템의 결합되지 않게 해주지만, 포트와 어댑터를 구성하고 관리하는 복잡성이 증가한다. 또한 독립적인 데이터베이스를 전제하지 않기 때문에 추가적인 설계가 필요하다. 외부 및 내부에서 각 서비스에 필요한 요청사항은 모두 API GATEWAY를 통하게 함으로써, 각 서비스를 완전히 독립적으로 관리할 수 있다. 또한 서비스별로 DB를 가지고 있기 때문에 해당 서비스 전용의 DB를 구축하기 좋다. 이런 독립성을 기반으로 이후에 추가될 다양한 기능을 마이크로 서비스로 구현하여 손 쉽게 확장할 수 있다.

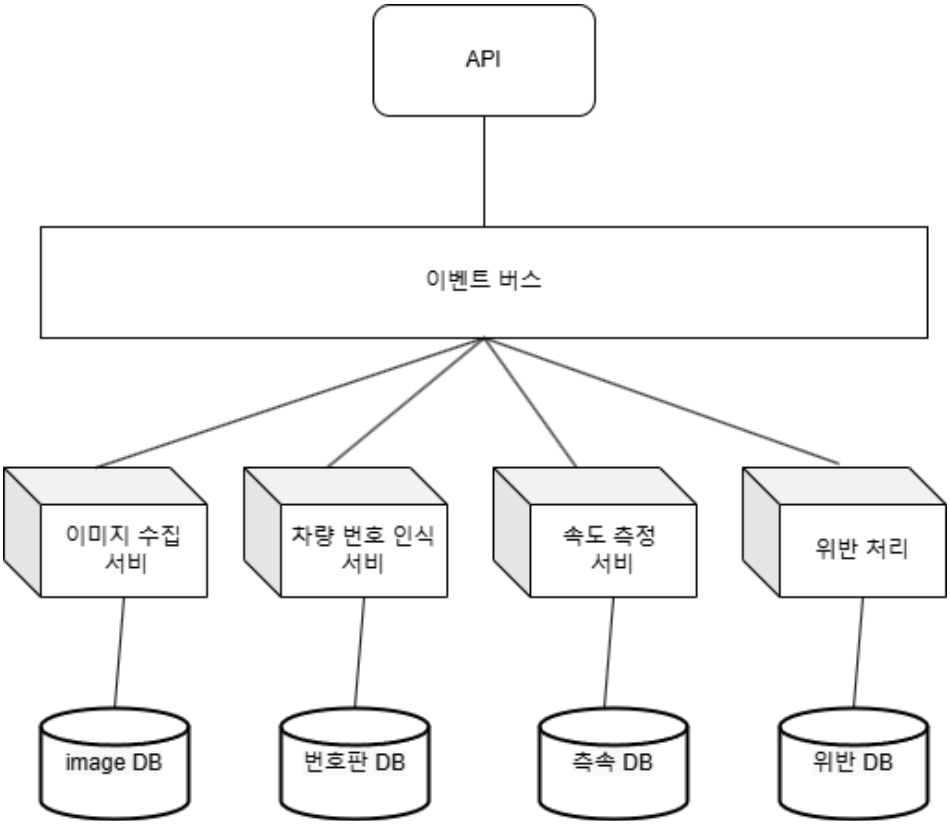
4.1.4. QA3: Reliability – Recoverability

4.1.4.1. Design Goal

시스템의 신뢰성과 복구 가능성을 향상시켜 서비스 중단, 노드 장애 또는 트랜잭션 실패와 같은 일부 장애가 발생하더라도 시스템이 신속하게 복구되고, 사용자 경험과 비즈니스 연속성에 미치는 영향을 최소화하도록 한다.

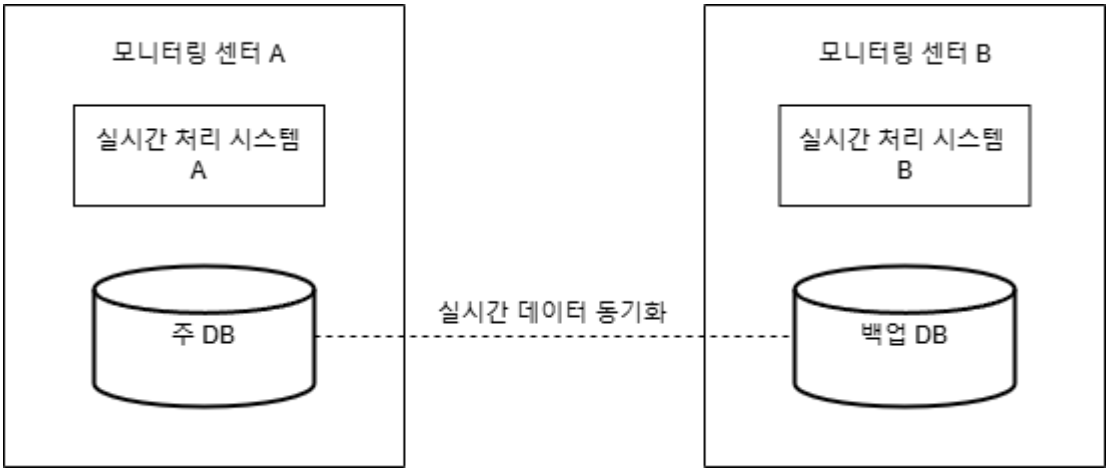
4.1.4.2. Candidate Design Approach List

4.1.4.2.1. CDA #1 Description: Event-Driven Architecture + Saga Pattern



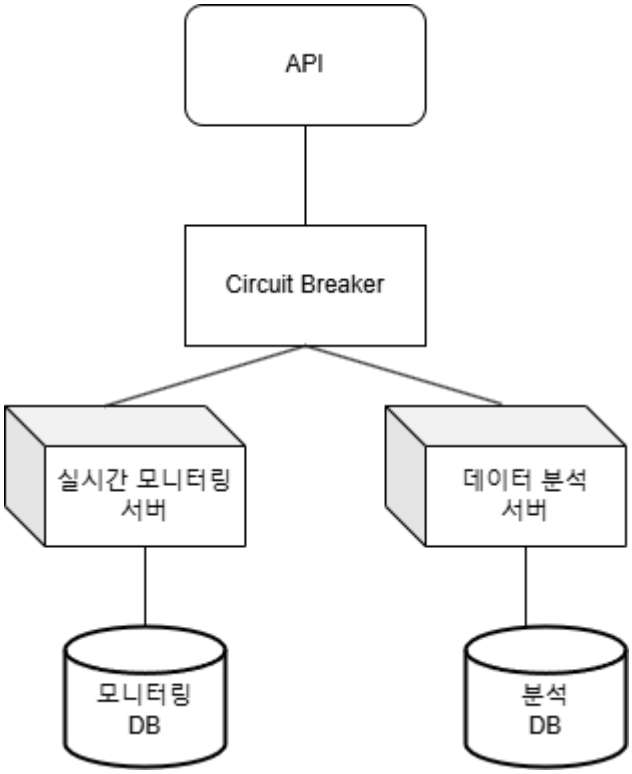
CDA ID	QA3_CD-01_CDA-01
Description	시스템에서 주요 서비스들이 서로 직접 통신하지 않고, 이벤트 버스를 통해 비동기로 상호작용한다. 각 서비스는 독립적으로 이벤트를 처리하며, Saga 패턴을 통해 분산된 트랜잭션을 관리한다. 트랜잭션 단계마다 보상 작업을 정의하여, 일부 단계에서 실패하더라도 시스템 일관성을 유지할 수 있다.
Pros	이벤트를 저장하고 재생하여 실패한 작업을 복구할 수 있다. 또한, Saga 패턴을 통해 복잡한 분산 트랜잭션을 간단한 단계로 나누고, 보상 작업으로 실패를 처리할 수 있다.
Cons	이벤트 관리 및 Saga 조율이 추가되면서 시스템 설계와 구현이 복잡해진다. 또한, 비동기 이벤트 흐름으로 인해 문제 원인을 파악하고 추적하는 데 시간이 걸릴 수 있다. 그리고 비동기 처리로 인해 모든 서비스에서 데이터의 일관성이 즉시 보장되지 않을 수 있다. 추가로, 장시간 실행되는 트랜잭션에 필요한 이벤트 데이터를 저장하고 관리하는 데 비용이 증가한다.

4.1.4.2.2. CDA #2 Description: Active-Passive Architecture + State Replication



CDA ID	QA3_CD-01_CDA-02
Description	Active-Passive 아키텍처와 상태 복제는 복구 능력에 중점을 둔 설계 패턴입니다. Active 노드는 모든 작업을 처리하며, Passive 노드는 실시간 상태 복제를 통해 Active 노드와 동기화를 유지합니다. Active 노드에 장애가 발생하면 Passive 노드는 복제된 최신 상태를 활용하여 매끄럽게 작업을 인수해 서비스 연속성과 데이터 무결성을 보장하며, 다운타임을 최소화합니다. 이 패턴은 빠른 장애 전환, 실시간 데이터 일관성, 간소화된 복구 프로세스를 통해 중요한 작업 시스템에 효율적인 복구 능력을 제공한다.
Pros	Active-Passive 아키텍처는 기본적으로 백업 노드를 준비하는 구조이기 때문에, 재해 발생 시 Passive 노드를 활용하여 빠르게 복구할 수 있다.
Cons	상태를 복제하는 과정에서 실시간 동기화가 어려울 수 있다. 또한, 리소스가 비효율적으로 사용될 수 있다. 그리고 Active 노드에 장애가 발생하면 Passive 노드가 즉시 전환되어야 하는데, 전환 과정에서 추가적인 지연이 발생하거나, 중간 데이터 손실 가능성이 존재할 수 있다. 추가로, 상태 복제를 위해 추가적인 네트워크 리소스와 처리 시간이 요구되기 때문에 전체 성능이 약간 저하될 수 있다.

4.1.4.2.3. CDA #3 Description: Microservices Architecture + Circuit Breaker Pattern



CDA ID	QA3_CD-01_CDA-03
Description	Microservices Architecture는 애플리케이션을 독립적인 작은 서비스로 분할하여 개발, 배포, 확장이 용이하도록 하는 아키텍처이다. 각 마이크로서비스는 고유한 비즈니스 기능을 수행하며, 서로 독립적으로 동작한다. Circuit Breaker 패턴은 마이크로서비스 간의 통신에서 발생할 수 있는 오류를 관리하고, 시스템이 특정 서비스의 장애로 인해 전체 시스템이 영향을 받지 않도록 방지하는 패턴이다. 이 패턴은 서비스 호출 중에 실패가 발생하면 일정 시간 동안 호출을 차단하고, 서비스가 복구되면 다시 연결을 시도한다. 이를 통해 시스템의 안정성을 높이고, 장애의 전파를 방지할 수 있다.
Pros	장애가 발생한 서비스는 자동으로 차단되고, 복구된 후에 다시 연결을 시도하므로 시스템 전체의 다운타임을 최소화할 수 있다.
Cons	**1. 복잡성(Complexity)** 여러 서비스가 서로 통신하고 상호작용하는 구조로, 관리와 모니터링이 복잡해질 수 있다. **2. 데이터 관리(Data Management)** 여러 마이크로서비스가 각각 독립적인 데이터 저장소를 관리해야 하므로 데이터 일관성 유지가 어려울 수 있다. **3. 통합 테스트 어려움(Integration Testing Challenges)** 개별적으로 서비스가 동작하지만, 전체 시스템의 동작을 테스트하는 것은 시간이 많이 걸리고 복잡할 수 있다.

	4. 오버헤드(Overhead) 각 서비스 간의 통신을 위한 네트워크 호출이 추가되므로, 시스템의 성능 저하를 초래할 수 있다.

4.1.4.3. Decision and Rationale

Performance		**Analysis**	**CDA #1** Event-Driven Architecture + Saga Pattern	**CDA #2 (selected)** Active-Passive Architecture + State Replication	**CDA #3** Microservices Architecture + Circuit Breaker Pattern
ID	**Title**				
QAS-03	시스템 데이터의 복구	Pros	(+) 실패한 작업을 복구 (+) 분산 트랜잭션 처리	(++) 빠른 복구 (+) 데이터 자동 복구	(+) 장애 격리 (+) 개별 서비스 복구 지원
		Cons	(-) 복잡성이 증가됨 (-) 디버깅이 어려움 (-) 데이터 일관성이 떨어질 수 있음 (-) 이벤트 스토리지의 비용이 증가할 수 있음	(-) 실시간 동기화가 어려울 수 있음 (-) 리소스가 비효율적으로 사용될 수 있음 (-) 데이터 손실 가능성 (-) 성능 저하 가능성	(-) 데이터 일관성 유지가 어려움. (-) 관리와 모니터링이 어려워짐 (-) 통합 테스팅이 어려움 (-) 서비스간 통신이 오버헤드로 작용

Candidate Design:

QA	**QAS**	**CD**	**Description**
QA3: Reliability (Recoverability)	QAS-03	QA3_CD-01 (Active- Passive Architecture + State Replication)	Active-Passive 아키텍처와 상태 복제는 복구 능력에 중점을 둔 설계 패턴입니다. Active 노드는 모든 작업을 처리하며, Passive 노드는 실시간 상태 복제를 통해 Active 노드와 동기화를 유지합니다. Active 노드에 장애가 발생하면 Passive 노드는 복제된 최신 상태를 활용하여 매끄럽게 작업을 인수해 서비스 연속성과 데이터 무결성을 보장하며, 다운타임을 최소화합니다. 이 패턴은 빠른 장애 전환, 실시간 데이터 일관성, 간소화된 복구 프로세스를 통해 중요한 작업 시스템에 효율적인 복구 능력을 제공한다.

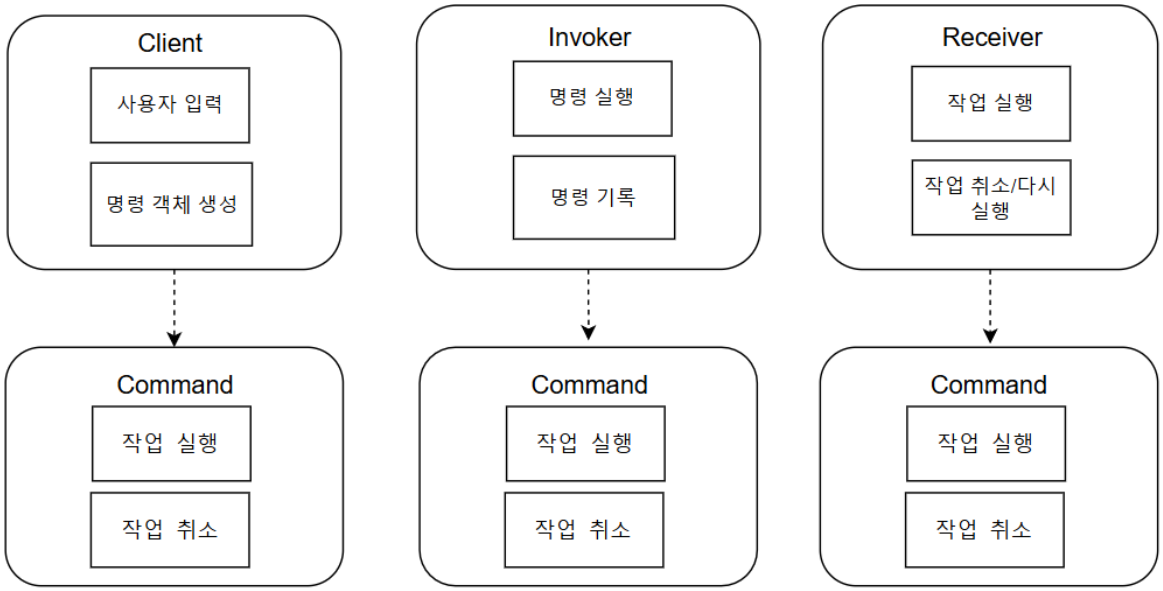
4.1.5. QA4: Usability – Operability

4.1.5.1. Design Goal

사용자 인터페이스는 직관적이고 사용하기 쉬워야 하며, 일반 사용자가 1분 이내에 원하는 작업을 수행할 수 있어야 한다. 모든 사용자 작업은 명확한 피드백을 제공해야 하며, 오류 발생 시 적절한 안내 메시지를 표시해야 한다. 또한 사용자의 숙련도에 관계없이 기본적인 시스템 조작이 가능해야 한다.

4.1.5.2. Candidate Design Approach List

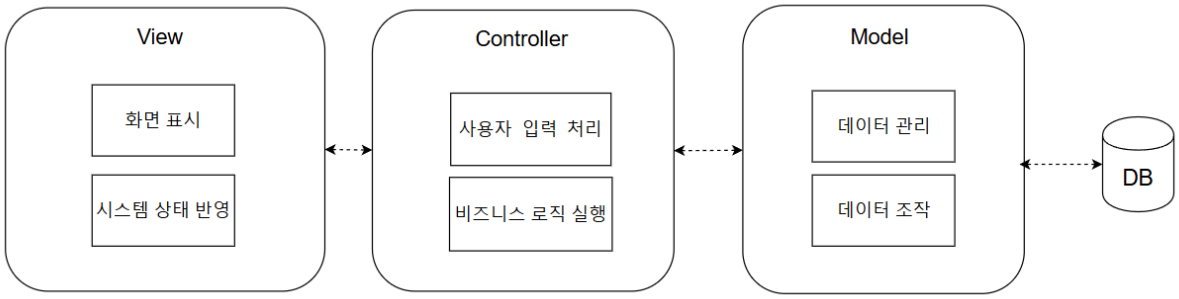
4.1.5.2.1. CDA #1 UI Command Pattern



CDA ID	*QA4_CD-01-CDA-01*
Description	사용자 인터페이스 작업을 위한 통일된 명령 패턴을 적용한다. 모든 사용자 작업은 명령 객체로 캡슐화되며, 통일된 프로세서를 통해 관리되고 실행된다. 시스템은 각 작업에 대한 즉각적인 피드백을 제공하며, 작업의 실행 취소/다시 실행 기능을 지원한다.
Pros	1. 통일된 작업 처리 방식으로 사용자 경험의 일관성 향상 2. 작업의 실행 취소/다시 실행 지원 3. 새로운 작업 기능 확장 용이

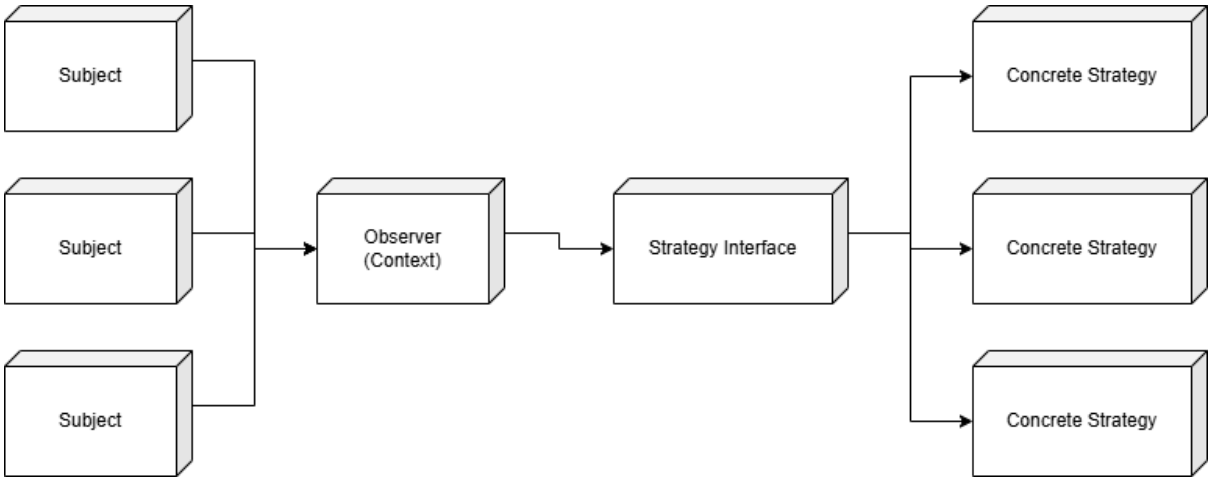
	4. 작업 행위의 추적 및 기록 가능 5. 언어 전환 이력 추적 및 관리가 용이
Cons	1. 시스템 복잡도 증가 2. 명령 기록 유지를 위한 추가 메모리 필요 3. 단순 작업의 과도한 캡슐화 가능성

4.1.5.2.2. CDA #2 MVC Pattern



CDA ID	*QA4_CD-01-CDA-02*
Description	MVC 아키텍처 패턴을 사용하여 사용자 인터페이스를 구성하고, 데이터, 표시, 제어 로직을 분리한다. 인터페이스의 신속한 응답을 보장하고 시스템 상태 변화를 즉시 반영한다. 컨트롤러를 통해 사용자 입력을 통합 관리하여 일관된 작업 경험을 제공한다.
Pros	1. 인터페이스 일관성 구현에 유리 2. 다양한 화면 뷰 구현 용이
Cons	1. 단순 기능에 대한 과도한 복잡성 2. 시스템 응답 속도 영향 가능성 3. 추가 개발 작업량 필요

4.1.5.2.3. CDA #3 Observer + Strategy Pattern



CDA ID	*QA4_CD-01-CDA-03*
Description	옵저버와 전략 패턴을 조합하여 인터페이스 요소의 동적 업데이트와 작업 행위의 유연한 구성을 구현한다. 시스템 상태 변화 시 관련 인터페이스 요소가 자동으로 업데이트되며, 다양한 상황에 따라 작업 전략을 동적으로 조정할 수 있다.
Pros	1. 인터페이스 요소의 시스템 상태 변화 자동 반영 2. 작업 행위의 손쉬운 구성 및 수정 3. 개인화된 작업 설정 지원
Cons	1. 성능 부하 발생 가능성 2. 과도한 상태 변화 시 사용자 경험 저하 3. 복잡한 구성 시 사용 난이도 증가

4.1.5.3 Decision and Rationale

Performance		**Analysis**	**CDA #1 (selected)** UI Command Pattern	**CDA #2** MVC Pattern	**CDA #3** Observer + Strategy Pattern
ID	**Title**				
QAS-04	다양한 언어 제공	Pros	(+) 언어 전환 이력 추적 및 관리가 용이 (+) 사용자 경험의 일관성 향상 (+) 작업의 실행 취소/다시 실행 지원 (+) 새로운 작업 기능 확장 용이 (+) 작업 행위의 추적 및 기록 가능	(+) 인터페이스 일관성 구현에 유리 (+) 다양한 화면 뷰 구현 용이	(+) 인터페이스 요소의 시스템 상태 변화 자동 반영 (+) 작업 행위의 손쉬운 구성 및 수정 (+) 개인화된 작업 설정 지원
		Cons	(-) 개발 및 유지보	(-) 단순 기능에 대	(-) 성능 부하 발생

			수 부담이 있음 (-) 시스템 복잡도 증가 (-) 명령 기록 유지를 위한 추가 메모리 필요 (-) 단순 작업의 과도한 캡슐화 가능성	한 과도한 복잡성 (-) 시스템 응답 속도 영향 가능성 (-) 추가 개발 작업량	가능성 (-) 과도한 상태 변화 시 사용자 경험 저하 (-) 복잡한 구성 시 사용 난이도 증가

Candidate Design:

QA	QAS	CD	Description
QA4: Usability (Operability)	QAS-04	QA4_CD-01	시스템 인터페이스는 다양한 언어를 지원해야 한다. 관리자가 사용 언어의 변경을 시도하면, 시스템은 1분 이내에 해당 언어로 인터페이스를 변경해야 한다. 또한, 시스템이 지원하지 않는 언어를 리소스로 취급하여 효율적으로 관리하고, 빠른 전환을 지원한다. 이때 언어 리소스는 모듈화 되어 관리되며, 새로운 언어 추가가 용이하도록 설계한다. 또한 사용자별 언어 설정을 유지하고, 시스템 전반에 걸쳐 일관된 언어 표시를 보장한다.

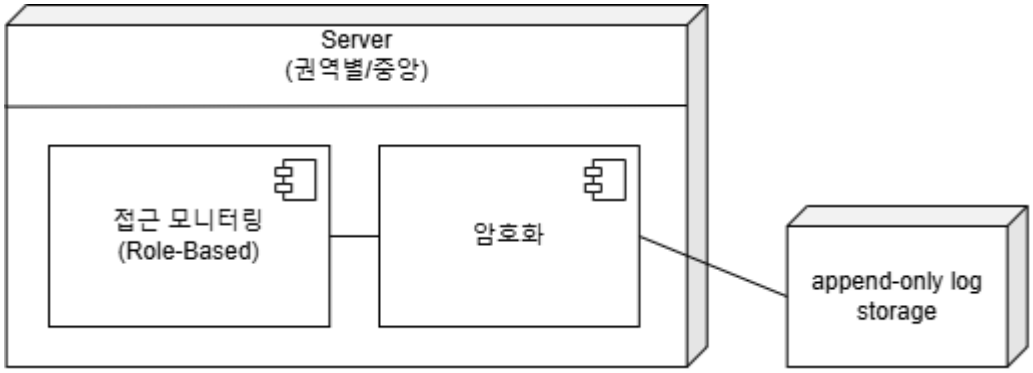
4.1.6. QA5: Security – Accountability

4.1.6.1. Design Goal

시스템은 처리한 단속 데이터를 외부 시스템 (경찰청 DB 서버, 법원 DB 서버)으로 보낸다. 이는 외부에서 시스템 데이터에 접근하여 데이터를 받아가는 것인데, 이때 시스템 데이터에 접근하는 모든 대상에 대하여 로그를 남긴다. 또한, 권역별 DB 서버 간에도 접근 시에 로그를 남겨, 접근이 허용된 서버의 관리자만이 해당 서버에 접근하였음을 확인할 수 있도록 한다. 시스템은 이 로그를 외부에서 건드릴 수 없도록 해야 한다.

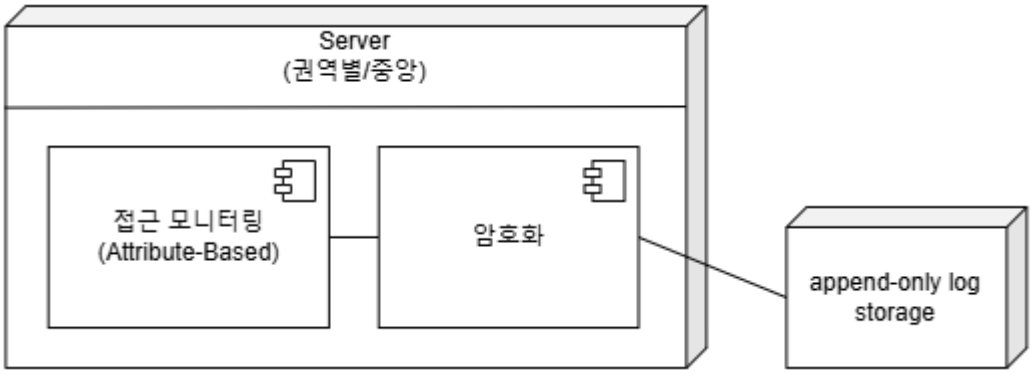
4.1.6.2. Candidate Design Approach List

4.1.6.2.1. CDA #1 Description: Role-Based Access Control + Immutable Log Pattern + Logging



CDA ID	*QA5_CD-01_CDA-01*
Description	Role-Based Access Control은 사용자가 수행할 수 있는 작업을 그들의 역할에 따라 할당하는 방식으로, 사용자의 역할에 따라 시스템 내에서 수행 가능한 작업이 달라진다. 이때 접근한 사용자에 대한 로깅을 수행하면, 어느 사용자가 접근했는지에 대한 확인이 가능하므로 accountability를 확보할 수 있다. 로그에 대해서는 immutable log pattern을 적용하여 append-only log storage에 암호화된 로그를 생성하도록 한다.
Pros	관리가 용이하고, 새로운 사용자를 추가하거나 기존 사용자를 관리할 때 효율적이다. 특히 대규모 시스템에서 조직적인 권한 관리를 위해 유용하다.
Cons	역할이 세분화되지 않으면 유연성이 떨어질 수 있고, 큰 시스템에서 역할이 너무 많아지면 관리가 복잡해질 수 있다.

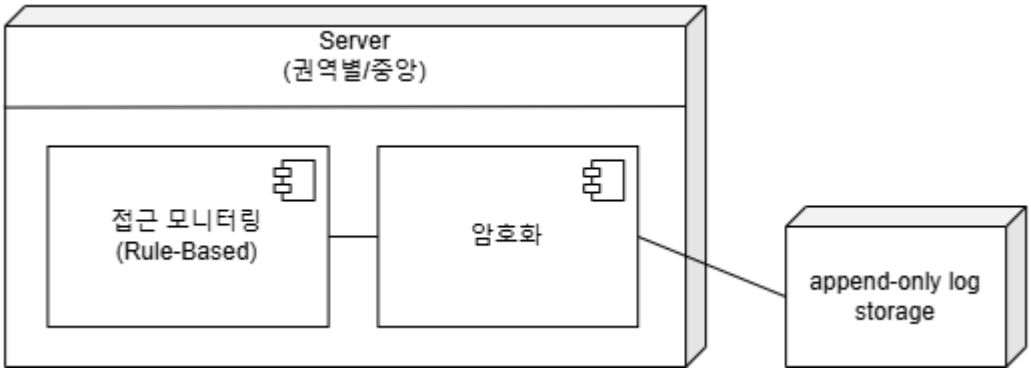
4.1.6.2.2. CDA #2 Description: Attribute-Based Access Control + Immutable Log Pattern + Logging



CDA ID	*QA5_CD-01_CDA-02*
Description	Attribute-Based Access Control은 사용자의 속성과 리소스의 속성 등 다양한 속성을 기반으로 접근을 제어하는 방식으로, 속성의 조합을 사용하여 매우 세밀한 권한 설정이 가능하다. 이때 접근한 사용자에 대한 로깅을 수행하면, 어느 속성을 가진 사용자가 어느 속성을 가진 리소스에 접근했는지에 대한 확인이 가능하므로

	accountability를 확보할 수 있다. 로그에 대해서는 immutable log pattern을 적용하여 append-only log storage에 암호화된 로그를 생성하도록 한다.
Pros	상황에 맞춰 동적인 접근 제어가 가능하며, 세부적이고 유연한 정책 설정이 가능하다.
Cons	정책 설정이 복잡해질 수 있고, 설정 및 유지 관리가 Role-Based Access Control에 비해 어려울 수 있습니다.

4.1.6.2.3. CDA #3 Description: Rule-Based Access Control + Immutable Log Pattern + Logging



CDA ID	*QA5_CD-01_CDA-03*
Description	Rule-Based Access Control은 미리 설정된 규칙에 따라 접근 권한을 부여하는 방식으로, 특정 시간, IP 주소, 위치와 같은 조건에 따라 권한이 주어진다. 이때 접근한 대상에 대한 로깅을 수행하면, 어느 속성을 가진 사용자가 어느 속성을 가진 리소스에 접근했는지에 대한 확인이 가능하므로 accountability를 확보할 수 있다. 로그에 대해서는 immutable log pattern을 적용하여 append-only log storage에 암호화된 로그를 생성하도록 한다.
Pros	조건에 맞춘 접근 제어로 보안을 강화할 수 있다.
Cons	설정된 규칙에 의존하므로 규칙이 너무 복잡해지면 관리가 어려울 수 있다.

4.1.6.3. Decision and Rationale

Performance		**Analysis**	**CDA #1** Role-Based Access Control + Immutable Log Pattern + Logging	**CDA #2** Attribute-Based Access Control + Immutable Log Pattern + Logging	**CDA #3 (selected)** Rule-Based Access Control + Immutable Log Pattern + Logging
ID	**Title**				
QAS-05	데이터 접	Pros	(+) 새로운 사용자를	(+) 동적인 접근 제어	(+) 조건에 맞춘 접근

	근에 대한 로깅		추가하거나 기존 사용자를 관리할 때 효율적 (+) 사용자에 대한 관리가 용이 (+) 접근 대상에 대한 로깅 (+) append-only log storage를 사용하여 암호화된 로그 생성	가능 (+) 세부적이고 유연한 정책 설정 가능 (+) 접근 대상에 대한 로깅 (+) append-only log storage를 사용하여 암호화된 로그 생성	제어로 보안을 강화할 수 있음 (+) 접근 대상에 대한 로깅 (+) append-only log storage를 사용하여 암호화된 로그 생성
		Cons	(-) 역할이 세분화되지 않으면 유연성이 떨어질 수 있음 (-) 역할이 너무 많아지면 복잡해질 수 있음	(-) 정책 설정이 복잡해질 수 있음 (-) 설정 및 유지 관리가 어려울 수 있음	(-) 규칙이 너무 복잡해지면 관리가 어려울 수 있음

Candidate Design:

QA	**QAS**	**CD**	**Description**
QA5: Security (Accountability)	QAS-05	QA5_CD-01 (Rule-Based Access Control + Immutable Log Pattern + Logging)	이 시스템은 차량 정보 및 운전자 정보와 같이 민감한 정보를 다루는 시스템으로, 데이터의 보호가 중요하게 여겨진다. 이때, 시스템 데이터의 암호화 뿐만 아니라 접근을 제어하고 접근한 대상을 로그로 남겨 관리할 필요가 있다. Rule-Based Access Control을 사용하면, 정해진 규칙을 만족하는 대상만이 시스템에 접근할 수 있다. 접근 대상을 사용자로만 한정 짓기 어렵다는 시스템의 특성을 고려하여 이 패턴을 선택하였다. 이 디자인은 또한 외부에서 수정할 수 없는 로그를 남길 수 있도록 immutable log pattern과 logging을 적용하여, append-only log storage에 암호화된 로그를 남길 수 있도록 한다. 시스템은 주기적으로 이 로그를 확인함으로써 규칙을 위반하는 대상이나, 규칙을 위반하지는 않더라도 의심스러운 대상이 접근하지는 않았는지를 확인할 수 있다. CDA #1의 경우, 사용자의 역할에 대해서만 구분하므

			로, 역할이 불분명한 사용자에 대해서는 관리가 용이하지 않을 것으로 판단하였다. CDA #2의 경우, 사용자의 속성과 리소스의 속성에 대한 구분 및 정책 설정이 복잡해지면 관리가 어려울 것으로 판단하였다.

4.2. Candidate Designs Evaluation for all QAs

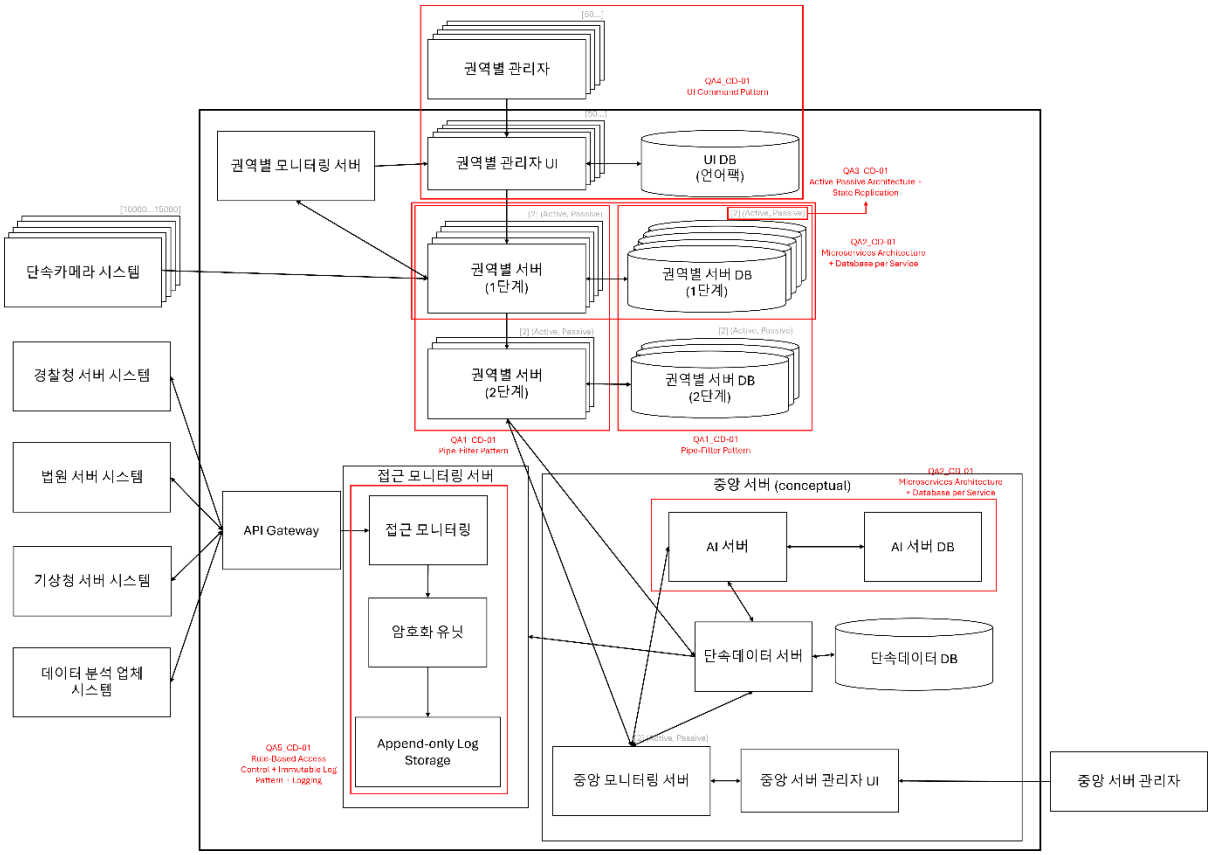
QA	QAS	Analysis	QA1_CD-01 Pipe-filter pattern	QA2_CD-01 Microservices Architecture + Database per Service	QA3_CD-01 Active-Passive Architecture + State Replication	QA4_CD-01 UI Command Pattern	QA5_CD-01 Rule-Based Access Control + Immutable Log Pattern + Logging
QA1 Performance	**QAS-01**	Pros	(+) 중앙 서버의 응답 속도가 비교적 빠름. (+) 중앙 서버 자원 효율적 사용	(+) 서비스 간의 독립성 확보 (+) 최적화 용이 (+) 수평 확장 용이	(+)Active 노드만 트래픽을 처리하므로 부하 분산 설계 없이 단일 노드의 성능에 집중할 수 있음	(+) 캡슐화 된 명령 객체는 저장 또는 재사용이 용이하여 반복 계산을 피하고 응답 속도를 높일 수 있다.	(+) RBAC는 미리 정의된 규칙에 따라 접근 권한을 부여하므로, 권한 확인이 빠르게 수행됨 (+) Immutable Log Pattern은 변경 없이 데이터 추가만 이루어지므로 효율적
		Cons	(-) 필터링 단계에서 지연 발생 필터링을 병렬 처리함 (-) 필터링 비정상 동작 시 성능이 저하 필터링 비정상 동작 시 문제가 해결될 때까지 필터링 없이 데이터를 전송	(-) 서비스 간의 통신 지연 서비스 분산의 세분화 정도(granularity)를 조정하여 통신 지연을 줄임 (-) 데이터 일관성 관리 오버헤드 증가 즉각적 동기화가 필요한 데이터를 구분해 관리 목표를 설정하고 오버헤드를 줄임	(-)Passive 노드는 Active 노드의 대체를 위해 대기 중이므로, 리소스 활용이 비효율적 Passive 노드를 read-only 상태로 활용하여, 쓰기 작업이 Active 노드에 집중되게 하여 리소스를 활용	(-)Command 객체를 생성함으로써 메모리 사용량과 CPU 사용량 증가 객체 풀링을 사용해서 객체 재사용율을 올려 사용량을 줄임	(-) 복잡한 규칙의 처리에 있어 비용 발생 규칙관리 툴을 사용 (-) 로그 처리 및 저장 비용 증가 로그의 압축 및 인덱싱을 통해 비용 절감 (-) 실시간 로그 분석으로 인한 추가 리소스 필요 로그 기록 이전에 외부 로그 분석 서비스를 활용
QA2 Maintainability	**QAS-02**	Pros	(+) 각 필터의 독립성 확보 (+) 재사용성 (+) 필터의 확장 용이	(+) 데이터베이스 독립관리 가능 (++) 서비스 확장성	(+)Passive 노드를 유지보수 중에도 Active 노드에서 시스템을 계속 운영할 수 있음	(+) 각 command에 대한 독립적 수정 및 테스트 가능 (+) 새로운 command 추가 시	(+) 규칙 수정과 추가 용이 (+) 일관된 형식으로 로그를 저장하므로 유

						기존 코드에 영향 없음 (+) UI와 command 간의 결합도가 낮아 유지보수성 향상	지보수성 향상 (+) 다른 사용자 그룹이나 리소스에 규칙 재사용 가능
		Cons	(-) 필터 간 연결 및 데이터 흐름이 복잡해질 수 있음 모든 필터가 표준화된 입력 및 출력 인터페이스를 사용하도록 설계	(-) 데이터 일관성 어려움 즉각적 동기화가 필요한 데이터를 구분해 관리 목표를 설정하고 오버헤드를 줄임	(-)Active-Passive 상태 복제를 위한 추가적인 설정 및 복제 시스템 관리가 필요 데이터베이스 복제 자동화 도구 사용	(-) command의 개수가 증가하면 복잡성 증가 Command를 카테고리화하여 묶음 처리	(-) 규칙이 복잡해지면 유지보수 어려움 규칙관리 툴을 사용 (-) 저장소 용량 관리 및 로그의 장기 보존 정책 필요 로그 보존 정책을 정의 (-) 규칙 변경 시 철저한 테스트 필요 테스트 프로세스를 문서화하고 준수하도록 하여 테스트 효율성 증대
QA3 Reliability	**QAS-03**	Pros	(+) 특정 필터의 장애가 다른 필터로 전이되지 않음 (+) 부분 복구로 복구 시간 단축 가능	(+)특정 서비스나 데이터베이스 문제가 다른 서비스로 전이되지 않음 (+)각 데이터베이스가 분리되어 있어 장애가 발생한 서비스만 복구하면 됨.	(++) 빠른 복구 (+) 데이터 자동 복구 (+) 고장 기간동안 성능이 완전히 상실되지 않음	(+) command의 undo/redo 쉽게 구현 가능 (+) command를 저장하면 장애 발생 시 복구 용이	(+) 신뢰 가능한 로그 데이터 (+) 규칙 기반의 안정성
		Cons	(-) 파이프의 의존성으로 인해 복구가 어려워질 수 있음 (-) 오류나 장애가 시스템 전체로 전파될 수 있음 Circuit Breaker 패턴	(-) 여러 데이터베이스가 관련된 트랜잭션 실패 시 롤백 처리가 어려움 트랜잭션 로그를 사용하여 변경 추적 및 복구	(-) 데이터 손실 가능성 다중 패시브 노드로 구성하여 손실 가능성을 낮춤	(-) command 실행 중 오류 발생 시 상태 복구의 부담 오류가 발생한 경우, 사용자에게 명확한 안내 메시지와 복구 옵션을 제공 (-) command의 상태 관리에 대한 부	(-)높은 트래픽에서 모든 작업을 기록할 경우 로그 생성이 시스템의 안정성에 영향을 미칠 수 있음 계획되지 않은 접근에 대해서

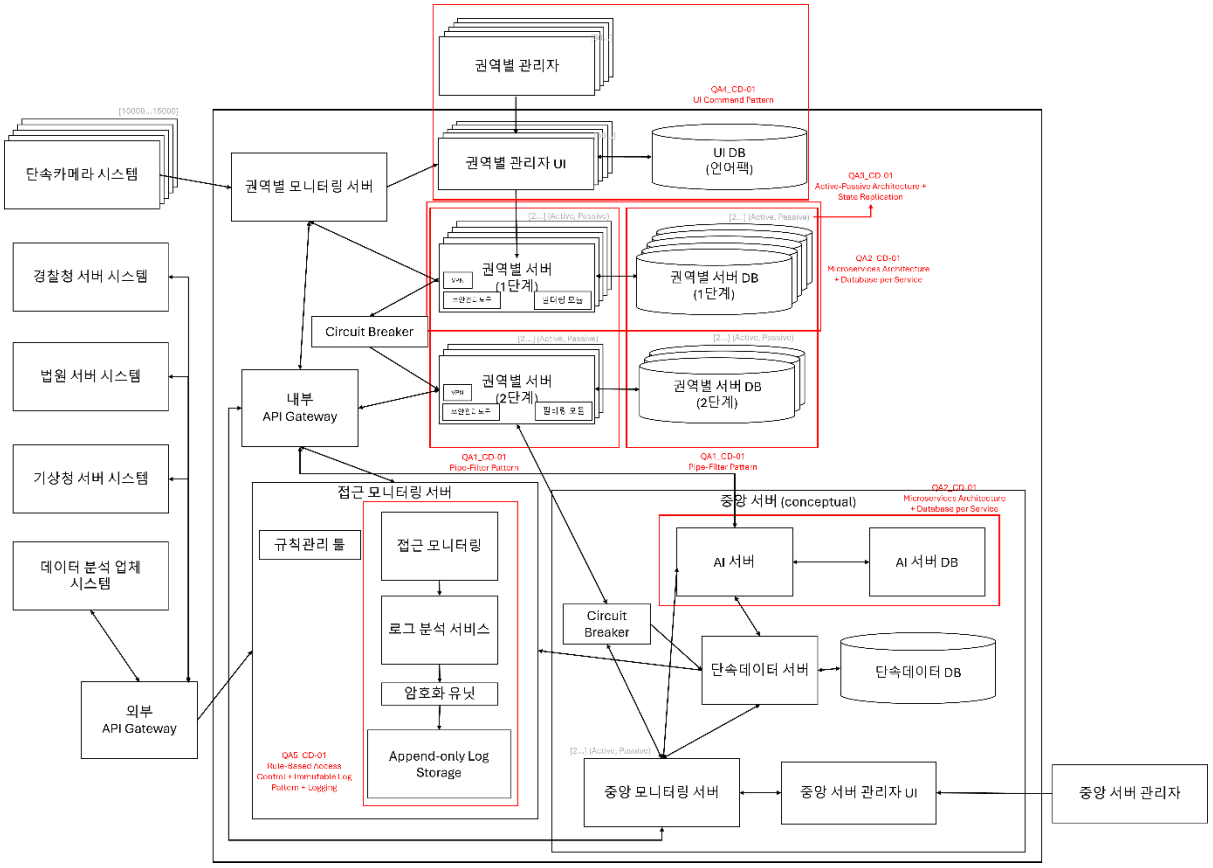
			을 적용하여 실패가 발생하는 경우 우회할 수 있도록 하고, 장애의 전파를 막는다.			담 Command의 상태 관리 정책을 정의하고, 시스템적으로 구분하여 일관성 유지	만 로깅
QA4 Usability	**QAS-04**	Pros	(+) 전체 데이터 흐름을 쉽게 이해하고 관리할 수 있음 (+) 각 필터의 상태와 성능에 대한 모니터링이 가능하므로 운영 효율성 증가	(+)서비스마다 다른 데이터 구조를 사용 가능하므로 사용성이 향상됨. (+)각 팀이 자신들의 데이터베이스를 관리할 수 있어 효율적 작업 가능	(+)단일 노드가 응답하므로 사용 경험이 단순하고 직관적	(+) 언어 전환 이력 추적 및 관리가용 이 (+) 사용자 경험의 일관성 향상 (+) 작업의 실행 취소/다시 실행 지원 (+) 새로운 작업 기능 확장 용이 (+) 작업 행위의 추적 및 기록 가능s	(+)RBAC는 사용자 역할에 따라 권한을 부여하므로, 관리자와 사용자가 쉽게 이해하고 사용할 수 있음
		Cons	(-) 필터와 파이프가 많아지면 운영 복잡성 증가 (-) 파이프와 필터가 증가하면 데이터 흐름 간 종속성 관리에 추가적 노력 필요 필터를 모듈화하여 독립적으로 배포 및 관리할 수 있게 함	(-)여러 데이터베이스에 걸쳐 데이터를 검색해야 할 경우 사용자 경험이 저하될 가능성이 있음 API 게이트웨이에 통합 검색이 가능한 API를 제공	(-)Failover 시 복제 지연으로 인해 사용자 데이터가 완전히 동기화되지 않은 상태에서 혼란을 초래할 가능성이 있음 동기식 복제를 통해 완전한 동기화를 보장	(-) 단순 작업의 과도한 캡슐화 가능성 복잡한 작업만 Command로 만들어서 처리	(-) 역할 정의가 너무 세분화되면 사용자가 자신의 권한 범위를 이해하기 어려울 수 있음 자신의 역할이 가진 권한 범위를 시각적으로 파악할 수 있도록 대시보드 제공
QA5 Security	**QAS-05**	Pros	(+) 필터 별 접근 제어 가능 (+) 특정 필터에서의 보안 집중적으로 수행 가능	(+) 데이터 접근 제어 및 데이터 격리 강화 (+) 서비스 별 로깅 및 추적 가능	(+)Passive 노드는 외부 접근이 없으므로 공격 표면이 작음	(+) 각 명령의 실행을 기록할 수 있어 사용자의 행동을 추적하고 비정상적인 동작을 감지하는 데 편리함	(+) 조건에 맞춘 접근 제어로 보안을 강화할 수 있음 (+) 접근 대상에 대한 로깅 (+) append-only log storage를 사용하여 암호화된 로그 생성

		Cons	(-) 필터별 별도의 보안 관리가 필요하므로 오버헤드 증가 보안 관리 도구를 활용 (-) 파이프를 통한 데이터 전달 중 보안 위협 데이터 전송 시 암호화를 적용해서 공격 방지	(-) 다중 보안 설정 필요 일관된 보안 정책을 적용하여 관리 (-) 서비스 간 데이터 공유에 있어 보안 문제 발생 가능 VPN을 통해 내부 네트워크 접근을 하도록 함. VPN 암호화방식으로 외부 접근 차단	(-)Active와 Passive 간 데이터 복제 시 네트워크 전송에서 보안 취약점이 발생할 가능성이 있음 데이터 전송 시 암호화를 적용해서 공격 방지 (-)장애 발생 후 Passive 노드 활성화 시, 적절한 인증 및 상태 동기화가 이루어지지 않으면 보안 문제가 생길 수 있음 동기식 복제를 통해 완전한 동기화를 보장	(-) command의 입력이 유효한 입력인지 확인하지 않으면 injection 공격으로 이어질 수 있음 Injection 공격 방지 logic 적용	(-) 규칙이 너무 복잡해지면 관리가 어려울 수 있음 규칙관리 툴을 사용

4.3. Design Decision



(Version 1, 수정 전)

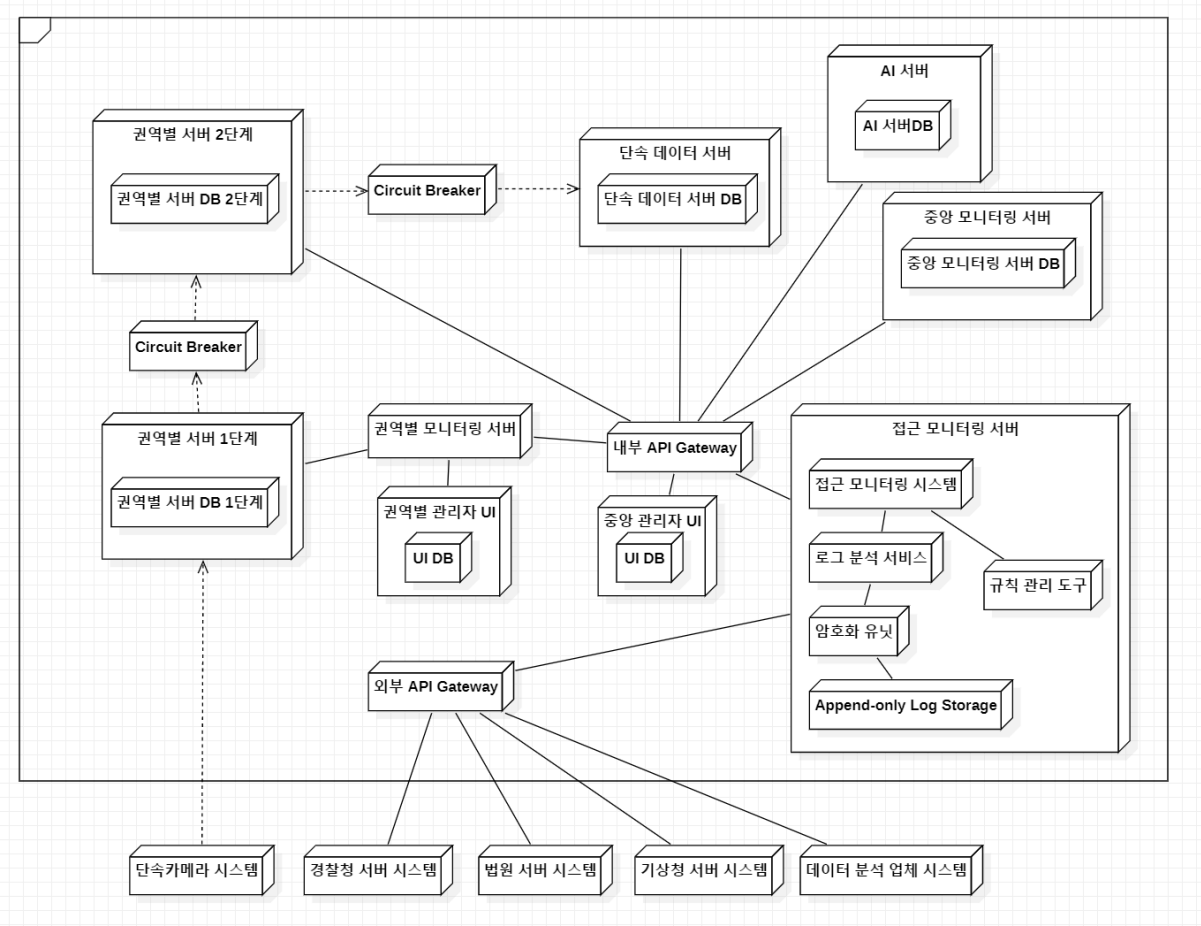


(Version 2, 수정 후)

5. Architecture Design Description

5.1. Architecture Overview

5.1.1. Architecture Overview Diagram



5.1.2. Node Specification

Name	Description
단속카메라 시스템	단속을 위한 외부 시스템으로, 교통법규 위반 정보를 감지하고 관련 데이터를 수집하여 전달함.
경찰청 서버 시스템	경찰청에서 사용하는 외부의 서버 시스템으로, 교통 단속 데이터를 저장하고 관리함. 또한, 차량 데이터 (운전자 정보, 차량 번호 등)를 관리하고 저장하며, 본 시스템에 제공함.
법원 서버 시스템	법원에서 사용하는 외부의 서버 시스템으로, 교통 단속 데이터를 저장하고 관리함.
기상청 서버 시스템	기상청에서 사용하는 서버 시스템으로, 기상 데이터를 제공함.
데이터 분석 업체 시스템	데이터 분석 업체에서 사용하는 시스템으로, 교통 단속 데이터와 이를 활용하여 분석한 정보를 저장하고 관리함.
권역별 모니터링 서버	각 권역별로 설치된 모니터링 서버로, 서버의 상태를 모니터링함.
권역별 관리자 UI	각 권역별로 사용하는 관리자 인터페이스로, 서버와 데이터를 관리하고 모니터링할 수 있도록 함.
권역별 서버 (1단계)	권역별로 사용하는 1단계 서버로, 이는 pipe-filter architecture에서 비롯된 node임. 직접적으로 관리자 UI와 연결되어 있고, 보안 관리 도구와 VPN, 필터링 모듈을 포함함. 필터링 모듈을 거쳐 2단계 권역별 서버로 필터링된 단속 데이터를 전달함.
권역별 서버 (2단계)	권역별로 사용하는 2단계 서버로, 마찬가지로 pipe-filter architecture에서 비롯된 node이고, 보안 관리 도구와 VPN, 필터링 모듈을 포함함. 권역별 서버 (1단계)에 연결되어 있으며, 필터링 된 단속 데이터를 받아, 이를 한 번 더 필터링하여 전달함.
권역별 서버 DB (1단계)	권역별로 사용하는 1단계 서버의 DB로, 권역별 서버 (1단계)에서 처리한 데이터를 저장함.
권역별 서버 DB (2단계)	권역별로 사용하는 2단계 서버의 DB로, 권역별 서버 (2단계)에서 처리한 데이터를 저장함.
내부 API Gateway	내부 시스템 간의 API 호출을 관리하는 게이트웨이.
외부 API Gateway	외부 시스템에서 본 시스템의 API를 호출하는 경우 이를 관리하는 게이트웨이.
규칙 관리 도구	Rule-Based Access Control에서 비롯된 노드로, 시스템에 접근하는 대상을 제한하는 규칙을 가지고 있으며, 이를 관리할 수 있는 도구임.

접근 모니터링 시스템	Access Control 및 Logging을 위하여, 시스템에 접근하고자 하는 외부 엔티티 (entity)의 접근을 모니터링함.
로그 분석 서비스	접근 모니터링 시스템으로부터 모니터링 데이터를 제공받으면, 해당 로그를 분석함. 주기적으로 시스템에 접근하는 대상 (경찰청 서버 시스템, 법원 서버 시스템 등)이 해당 주기에 맞춰 시스템에 접근하는 것은 로그를 별도로 남기지 않도록 함.
암호화 유닛	분석된 로그를 암호화하는 노드로, 데이터의 보안을 위하여 존재함.
Append-only Log Storage	로그를 추가만 할 수 있는 (append-only) storage로, 암호화된 접근 로그를 추가함. 정의된 로그 보관 정책에 맞게 데이터를 가지고 있다가 일정 주기마다 일괄적으로 데이터를 삭제하는 등의 작업을 할 수 있음.
AI 서버	데이터 분석, 예측 또는 기타 지능형 처리 작업에 사용되는 인공지능 모델 또는 알고리즘을 실행하는 서버.
AI 서버 DB	AI 서버에서 사용하는 데이터를 저장하는 DB로, 그간 쌓인 교통법규 위반 데이터 및 모델 파라미터를 포함함.
단속 데이터 서버	교통 단속과 관련된 데이터를 처리하는 서버로, 교통법규 위반 정보를 관리함.
단속 데이터 서버 DB	단속 데이터 서버에서 처리하는 데이터를 저장하는 DB.
중앙 모니터링 서버	다양한 구성 요소에서 데이터를 집계하여 교통위반 정보 모니터링이 가능하도록 하는 중앙 허브.
중앙 모니터링 서버 DB	모니터링 서버에서 수집한 데이터를 저장하는 DB.
중앙 관리자 UI	내부 API Gateway에 연결되어 시스템 전체의 관리가 가능하도록 하는 UI로, 각종 규칙을 관리하거나, 상태를 확인하거나, 모니터링 할 수 있음.
Circuit Breaker	장애가 연쇄적으로 전파되는 것을 방지하여 시스템 탄력성과 연속성을 보장하는 보호 조치를 위한 노드. 권역별 1단계와 2단계 서버 사이에 존재하며, 중앙 모니터링 서버와 단속 데이터 서버 사이에 존재하여, 장애의 전파를 방지함.

5.1.3. Execution Environment Specification

Node	Name	Description
권역별 관리자 UI	HTML5 Browser	HTML5 브라우저에서 실행되는 웹 기반 사용자 인터페이스로, React 또는 Angular와 같은 프레임워크를 활용하며, WebLogic Server 12c를 기반으로 하는 API 서버를 통해 지원함.
규칙 관리 도구	WebLogic Server12c	ava 기반 서버 (예: Tomcat 또는 WebSphere)에서 실행되는 웹 애플리케이션으로, 규칙 관리에 MySQL 데이터베이스를 사용함.
로그 분석 서비스	Log Analysis Platform	Linux 서버에서 실행되는 Elastic Stack (ELK)을 사용하여 로그 분석을 수행하는 서비스.
암호화 유닛	Encryption Service	Kubernetes에서 실행되는 마이크로 서비스로, OpenSSL을 사용하여 데이터 암호화를 수행함.
중앙 관리자 UI	HTML5 Browser	HTML5 브라우저에서 실행되는 웹 기반 사용자 인터페이스로, React 또는 Angular와 같은 프레임워크를 활용하며, WebLogic Server 12c를 기반으로 하는 API 서버를 통해 지원함.
Circuit Breaker	Middleware Component	Java 기반 서버에서 실행되는 Spring Cloud 또는 Hystrix와 같은 미들웨어 컴포넌트로, 장애 허용성을 보장함.

5.1.4. Communication Path Specification

Path	Description
외부 API Gateway - 경찰청 서버 시스템	REST API (실시간 API 요청이 아닌 경우) 및 WebSocket (범죄 차량 단속과 같은 실시간 API 요청이 필요한 경우)를 사용하여 데이터를 주고 받음.
외부 API Gateway - 법원 서버 시스템	REST API를 통해 단속 데이터를 받아 감.
외부 API Gateway - 기상청 서버 시스템	REST API를 통해 기상 데이터를 받아 옴.
외부 API Gateway - 데이터 분석 업체 시스템	REST API를 통해 단속 데이터를 받아가고 분석된 데이터를 받아 옴.
외부 API Gateway - 접근 모니터링 서버	API Gateway로부터 접근 정보 (외부 API Gateway의 통신 기록)를 받아와 저장함.
내부 API Gateway - 접근 모니터링 서버	API Gateway로 접근 모니터링 데이터를 제공하여 중앙 관리자가 중앙 관리자 UI를 통해 확인할 수 있도록 함.
내부 API Gateway - 중앙 모니터링 서버	API Gateway를 통하여 다른 서비스 노드들의 상태를 확인할 수 있도록 함.
내부 API Gateway - AI 서버	API Gateway를 통하여 단속 데이터 서버로부터 단속 데이터를 받아오고, 중앙 관리자 UI 및 중앙 모니터링 서버에 데이터 및 상태를 전송함.
내부 API Gateway - 단속 데이터 서버	API Gateway를 통하여 AI서버로부터 요청 받은 데이터를 건네주고, AI처리된 데이터를 받아올 수 있다. 중앙 관리자 및 중앙 모니터링 서버에 데이터 및 상태를 전송한다.
내부 API Gateway - 권역별 서버 2단계	API Gateway로 교통법규 위반 데이터를 전송하여, 단속 데이터 서버에서 이를 처리할 수 있도록 함.
내부 API Gateway - 권역별 모니터링 서버	API Gateway를 통해 권역별 모니터링 결과를 중앙 관리자 UI에 전달할 수 있도록 함.
내부 API Gateway - 중앙 관리자 UI	API Gateway를 통해 서비스 노드들이 제공하는 정보 (모니터링 정보, 데이터 처리 정보, 시스템 상태 정보 등)를 받아 옴.
권역별 모니터링 서버 - 권역별 관리자 UI	권역별 서버의 모니터링 상태 및 조작을 권역별 관리자 UI를 통해 할 수 있도록 함.
권역별 모니터링 서버 - 권역별 서버 1단계	권역별 모니터링 서버는 권역별 서버 1단계의 상태를 모니터링함.
단속카메라 시스템 -	각 단속카메라 시스템이 얻은 단속 데이터를 가장 가까운 권역별 서버 1단

권역별 서버 1단계	계에 직접 전달함.
권역별 서버 1단계 - Circuit Breaker - 권역별 서버 2단계	권역별 서버 1단계는 Circuit Breaker를 사이에 두고 권역별 서버 2단계와 연결되며, 이는 장애의 파를 막기 위함임. 권역별 서버 1단계에서 처리되고 필터링 된 데이터는 권역별 서버 2단계로 전송됨. 실시간 추적 대상 데이터에 대해서 권역별 서버 1단계에서는 차량 번호 및 위치정보로 필터링 함.
권역별 서버 2단계 - Circuit Breaker - 단속 데이터 서버	권역별 서버 2단계는 Circuit Breaker를 통과하여 필터링 된 데이터를 최종 단속 데이터 저장 지점인 단속 데이터 서버에 전달함. 권역별 서버 2단계에서는 실시간 추적 대상 데이터에 대해서 내부 API Gateway 를 통해 받아온, 도난 및 범죄 차량 데이터를 사용하여 타켓 차량 여부를 판단하고 필터링 함.
접근 모니터링 시스템 - 로그 분석 서비스	로그 분석 서비스는 접근 모니터링 시스템이 처리한 접근 로그 데이터를 받아 옴.
접근 모니터링 시스템 - 규칙 관리 도구	규칙 관리 도구는 접근 모니터링 시스템의 규칙을 관리하며, 규칙이 적용될 수 있도록 저장하고 제공함.
로그 분석 서비스- 암호화 유닛	로그 분석 서비스에서 분석된 로그는 암호화 유닛으로 전달됨.
암호화 유닛 - Append-only Log Storage	암호화 유닛에서 암호화된 로그는 append-only log storage로 전달됨.

5.2. Structure View

5.2.1. Static Structure Model

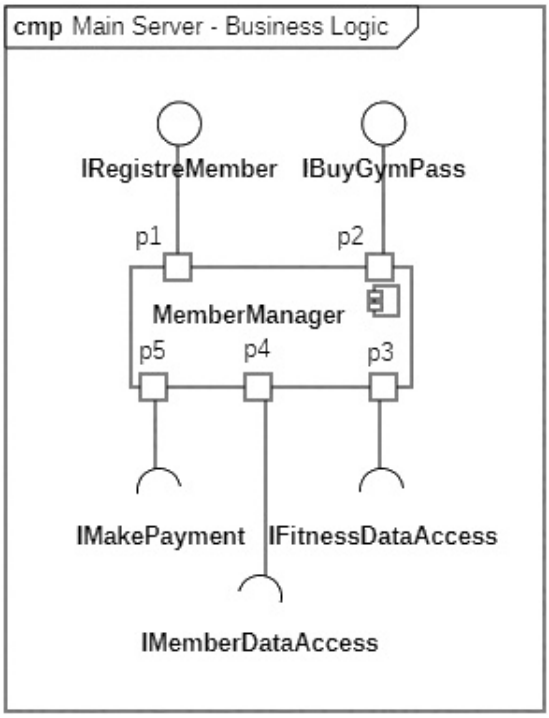
5.2.1.1. Static Structure Diagram

5.2.1.2. Element List

Name	Responsibility	Relevant ADs
Layer1		-
…		-
Layer3		-
자동차번호 판인식	이 컴포넌트는 차량 번호를 인식한다. 번호판 이미지를 받아서 차량 번호를 추출한다. 번호판 이미지는 형태1, 형태2, 형태3 등을 지원한다. QA-01, QA-02를 달성하기 위하여 번호 추출 성능은 초당 100임	UC-01, UC-02 QA-01, QA-02, QA-03
Component2		UC-02 QA-01 QA-03
…		
Component5		UC-03 QA-02

5.2.2. Component 1 Name

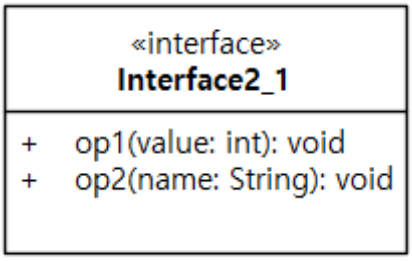
5.2.2.1. Component 1 Specification



5.2.2.1.1. Interface List

Name	Kind	Description
IRegisterMember	Provided	회원 가입을 요청하는 interface
IBuyGymPass	Provided	이용권 구매를 요청하는 interface
IMakePayment	Required	결제를 요청하는 interface
IMemberDataAccess	Required	DB에 저장된 회원 정보에 접근하는 interface
IFitnessDataAccess	Required	DB에 저장된 피트니스 정보에 접근하는 interface

5.2.2.1.2. *Interface2_1* Interface Specification



Operation	Responsibility
op1()	
op2()	

5.2.3. Component 2 Name

5.2.4. Component 3 Name

5.3. Behavior View

5.3.1. *UC-01 Title* Use Case Behavior Model

5.3.1.1. Behavior Diagram

5.3.1.2. Behavior Description

5.3.2. *UC-02 Title* Use Case Behavior Model

5.3.3. *UC-03 Title* Use Case Behavior Model

5.3.4. *UC-04 Title* Use Case Behavior Model

5.3.5. *UC-05 Title* Use Case Behavior Model

5.4. Deployment View

5.4.1. Artifact Deployment Model

5.4.1.1. Artifact Deployment Diagram

5.4.1.2. Artifact Deployment Specification

Name	Deployment Target		Description and Rationale
	Node	Execution Environment	
Artifact1	Application Server	WebLogic Server 12c	
Artifact2	Application Server	Ubuntu	
Artifact3	Database Server	MariaDB 10.4	

5.4.2. Artifact Definition Model

5.4.2.1. Artifact Definition Diagram

5.4.2.2. Artifact Definition Specification

<

Name	Manifested Component	Depends on	Description and Rationale
Artifact1	*Component1*		
Artifact2	*Component2* *Component3* *Component4*		
Artifact3	*Component5*		

6. Component Design Description

6.1. *Component 1* Design Description

6.1.1. Overview

6.1.2. Component Structure Diagram

6.1.3. Element List

Name	Responsibility

6.1.4. Design Rationale

QA/QAS	Relevant Elements	Description
QA1		
QAS-04	-	-
…		

6.1.5. Component Behavior Diagram

6.2. *Component 2* Design Description

6.3. *Component 3* Design Description

7. Architecture Traceability Summary

7.1. Architecture Traceability Graph

7.2. Summary of Traceability Items

Traceability Item		Description
ID	Title	

7.3. Safety Case