

Automotive Secure OTA Update

2025.08

자동차융합대학



GENERAL MOTORS
GM TECHNICAL CENTER KOREA



2023년 무상수리/리콜 비용 270억



자동차 제조사가
OTA를 서둘러야 하는 이유

Q. 제네시스 GV60에 적용된 무선 소프트웨어 업데이트 기술이란 무엇인가?

OTA총괄팀 | 무선 소프트웨어 업데이트 기술은 무선 통신을 활용하여 차량의 소프트웨어를 최신 버전으로 업데이트 할 수 있는 기술이다. 이를 통해 GV60에 적용된 다양한 **소프트웨어와 제어기는 언제나 최신 상태로 유지**되며, 고객은 더욱 안전하고 편리한 차량 상태를 경험하고 새로운 기능의 혜택도 누릴 수 있다.

Q. GV60의 무선 소프트웨어 업데이트 기술의 차별화된 특징은 무엇인가?

OTA총괄팀 | 무선 소프트웨어 업데이트는 크게 **소프트웨어 업데이트(Software Over the Air, SOTA)**와 **펌웨어 업데이트(Firmware Over the Air, FOTA)**로 나뉜다. **소프트웨어 업데이트(SOTA)**는 내비게이션, 인포테인먼트 시스템의 업데이트로 기존 제네시스 차량에도 적용된 기술이다. 하지만 **펌웨어 업데이트(FOTA)**는 차량에 적용된 다양한 제어기를 개선하는 기술로, GV60에 제네시스 최초로 적용됐다. GV60는 차량의 동력 성능(VCU), 브레이크(IEB), 스티어링(MDPS), 전자제어 서스펜션(ECS), 페이스 커넥트 등 차량 전반의 성능 및 기능 개선이 가능하며, 이를 통해 고객은 서비스센터 방문 없이도 최신으로 개선된 차량 시스템을 경험할 수 있다.

[Ref] <https://www.hyundai.co.kr/story/CONT0000000000002550>

Q. 무선 소프트웨어 업데이트를 통한 차량 업데이트는 어떻게 진행되나?

OTA총괄팀 | 무선 소프트웨어 업데이트는 크게 2단계로 진행된다. 첫 번째 단계는 다운로드인데, 차량은 주행 중에 신규 소프트웨어를 무선 통신으로 다운로드하고 업데이트 준비를 진행한다. 다운로드는 주행 여부와 상관없이 진행되지만, 무선 통신 연결이 해제되면 다운로드 과정이 중단될 수 있다. 설치 단계에서는 운전자가 정차 후 전원을 끄면, 차량이 업데이트를 하기에 안전한 상태인지 확인하는 과정을 거치고, 인포테인먼트 디스플레이를 통해 업데이트 승인 여부를 알리는 정보창을 띠운다. 운전자가 시작을 선택하면 새로운 소프트웨어의 설치가 진행되고, 디스플레이를 통해 업데이트 진행률을 확인할 수 있다. 유의할 점은 소프트웨어 업데이트 중에는 안전 조치로써 차량 운행이 제한된다. 따라서 충분한 시간 여유가 있을 때 업데이트 진행이 필요하다.

Q. PE 시스템의 업데이트로 출력 향상이나 드라이브 모드 추가 등 주행 성능의 개선도 가능한가?

전자네트워크개발팀 | 무선 소프트웨어 업데이트에 해당하는 제어기 중 PE 시스템을 관제하는 VCU(EV Control Unit)가 포함돼 있으므로 기술적으로 가능하다. 단, 소프트웨어의 개선 또는 해당 기능의 개발이 선행되어야 하고, 차량의 하드웨어도 그에 맞도록 개선이 필요하다.

아울러 차량의 성능을 효율적으로 구현하는 데 필요한 다양한 설정도 변경할 수 있다. 예를 들면, 가속 페달의 반응 설정을 개선해 가속 특성의 변화가 가능하며, 각 드라이브 모드의 특성을 세밀하게 조율할 수도 있다. 또한 PE 시스템뿐만 아니라, 스티어링의 감도나 전자식 서스펜션의 설정을 달리해 주행 질감의 개선도 가능하며, 에어백 전개 설정이나 페이스 커넥트의 개선을 통해 차량의 안전성 및 보안 성능도 개선할 수 있다.

[Ref] <https://www.hyundai.co.kr/story/CONT0000000000002550>

Q. 무선 소프트웨어 업데이트로 ADAS(Advanced Driver Assistance System) 기능의 개선도 기대할 수 있다?

전자네트워크개발팀 | GV60는 무선 소프트웨어 업데이트 기술이 제네시스 브랜드 최초로 적용된 차량으로써 차량과 ADAS를 제어하는 제어기에 무선 소프트웨어 업데이트 기술이 적용돼 있으며, 향후 ADAS 기능의 확장이나 개선도 가능하다. 단, ADAS는 크게 자율주행과 자율주차 부문으로 구분돼 있는데, GV60의 경우 우선 자율주차 부문에만 무선 소프트웨어 업데이트 기술이 적용돼 있다. 따라서 차후 자율주차 부문에서 새로운 기능이 추가되거나 편의성이 개선될 수 있다.

Q. 무선 소프트웨어 업데이트의 적용 범위가 확장된 만큼 보안도 중요할 것 같다. 해킹을 대비한 보안 대책도 마련돼 있나?

전자네트워크개발팀 | 소프트웨어 업데이트가 무선 통신으로 이뤄짐에 따라 사이버 보안의 중요성에 대해서도 깊이 공감하고 있다. 특히 무선 소프트웨어 업데이트 기능이 다수의 제어기로 확장된 만큼, 차량의 안전성을 확보하기 위해 빈틈없는 대책을 마련했다. 우선 고객이 스스로 무선 통신으로 소프트웨어 업데이트를 진행해야 하는 특성을 고려해 서버와 차량 간의 인증을 통한 무선 통신 보안을 적용했다. 또한 제어기 업데이트 시 루파일 변조나 외부 해킹을 막기 위해 업데이트 대상 제어기별로 하드웨어 보안 모듈을 탑재했다.

OTA총괄팀 | 최근 자동차에 대한 사이버 공격 위협이 등장하면서 자동차 사이버 보안에 대한 규제 역시 등장하고 있다. 대표적인 것이 UN 유럽경제위원회 세계 포럼이 발표한 ‘UNECE WP.29’ 규정인데, 제네시스 GV60 역시 해당 사이버 보안 규정에 따라 대비를 하고 있으며, 전담팀(TF) 및 연구소 각 부문에서도 대응하고 있다. 참고로 향후에는 사이버 보안 법규를 충족하지 못하면 차량 판매 또한 불가능하다. 법규를 충족한 차량이라면 고객이 충분히 안심하고 차량을 이용할 수 있다는 이야기다.

[Ref] <https://www.hyundai.co.kr/story/CONT0000000000002550>

Q. 스마트폰의 OTA 업데이트 시 앱 충돌, 로딩 속도 지연, 오류 발생 등 이상 증세를 보이는 경우도 있다. 이와 같은 오류 증상을 줄이기 위한 대책도 마련돼 있나?

OTA총괄팀 | 제네시스 GV60는 **소프트웨어 업데이트 시 발생하는 오류 증상을 줄이기 위해 다양한 검증 절차와 복구 방법**을 마련했다. 우선 오류가 발생하지 않도록 신규 제어기 소프트웨어 개발 단계에서 철저한 검증 절차를 마련했으며, OTA 관제 시스템은 실시간으로 업데이트 상황을 모니터링하면서 이상 징후를 감시하고 대응한다. 만약 이상 징후가 발견되면 자동으로 해당 소프트웨어의 배포를 중단하고, 긴급출동 서비스와 연동해 빠르게 문제를 해결할 수 있도록 대비하고 있다.

전자네트워크개발팀 | 설령 오류가 발생해도 GV60는 운행에 문제가 없도록 대응한다. 만약 무선 소프트웨어 업데이트 진행 시 오류가 발생하면 다시 한번 업데이트를 수행하고, 그래도 실패한다면 기존 버전으로 복구하는 과정을 거친다. 또한 각 제어기는 기능적으로 서로 상호작용을 하면서 작동되므로, 하나의 제어기가 복구된다면 모든 제어기가 동시에 복구돼 차량 주행에 이상이 없도록 조치한다.

Q. 무선 소프트웨어 업데이트 과정 중 차량 운행이나 충전이 가능한가?

전자네트워크시험팀 | **새로운 소프트웨어의 설치 단계에서는 기본적으로 차량 운행이 제한된다.** 이는 업데이트 영역이 차량 전반으로 확장되었으며, 각종 제어기의 역할이 고객의 안전과 직결돼 있기 때문이다. 설치 단계에서는 차량 운행과 충전이 불가능하며, V2L(Vehicle to Load) 기능도 사용할 수 없다. 단, 안전과 관련된 최소한의 기능인 도어 개폐, 비상등 점멸, 전동 트렁크 개폐 등을 가능하다. 따라서 GV60는 차량의 전원을 껏을 때, 업데이트 진행 여부를 묻는 절차와 예상 소요 시간 정보를 제공하는 기능을 마련했다. 운전자는 디스플레이를 통해 업데이트 내역을 미리 확인할 수 있으며 안심하고 업데이트를 진행할 수 있다.

[Ref] <https://www.hyundai.co.kr/story/CONT0000000000002550>

Q. 무선 소프트웨어 업데이트는 고객에게 어떤 가치를 제공할 수 있나? 또, 무선 소프트웨어 업데이트는 어떤 방향으로 발전이 예상되나?

전자네트워크개발팀 | 우리는 스마트폰과 같은 전자제품을 경험하면서 신제품이 출시하면 금세 구형이 되어버린 느낌을 받곤 한다. 이는 스마트폰보다도 훨씬 비싼 자동차도 마찬가지다. 신차는 언제나 새롭고 더욱 편리한 기능을 탑재하기 때문이다. 하지만 무선 소프트웨어 업데이트 기술이 적용됐다면 고객의 차량은 끊임없이 개선될 것이다. 결국 이 기술의 장점은 지속적인 품질 개선과 상품성 개선이며, 이는 고객 차량의 수명을 늘리는 현실적인 방법이 될 것으로 예상한다.

또한 차량을 개선하기 위한 고객의 수고도 확연히 줄어들 것이다. 기존에는 서비스센터를 방문하기 위해 고객이 시간과 비용을 들여야 했지만, GV60는 무선 통신으로 소프트웨어를 다운로드하고, 고객이 직접 업데이트를 진행해 곧바로 개선된 차량의 기능과 품질을 경험할 수 있다. 아울러 고객이 원하는 기능을 무선 소프트웨어 업데이트를 통해 빠르게 반영하여 향후에는 고객 맞춤형 차량 기능도 제공할 수 있을 것으로 기대한다.

[Ref] <https://www.hyundai.co.kr/story/CONT0000000000002550>

CONTENTS

01

OTA 개요

02

SDV와 OTA

03

OTA 관련 법규와 표준

04

OTA 프로세스 개요

05

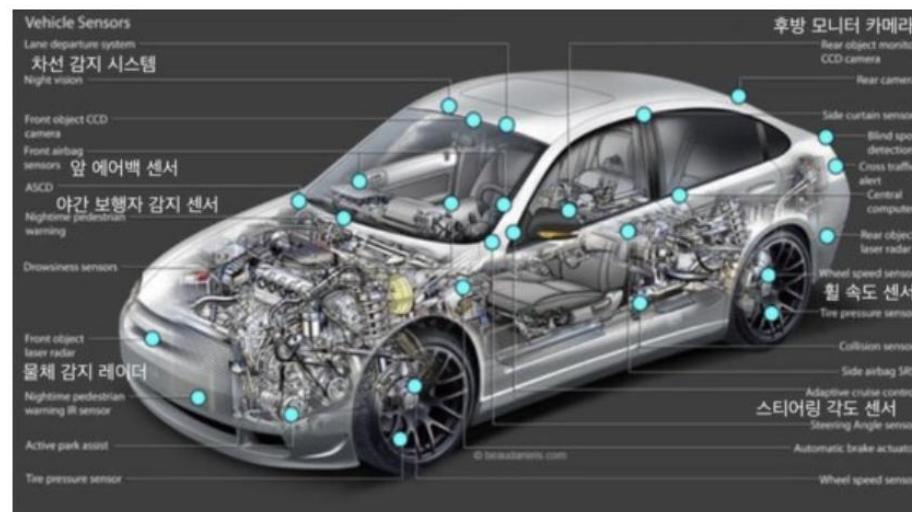
OTA 요소 기술의 이해

01

OTA 개요

■ 정의

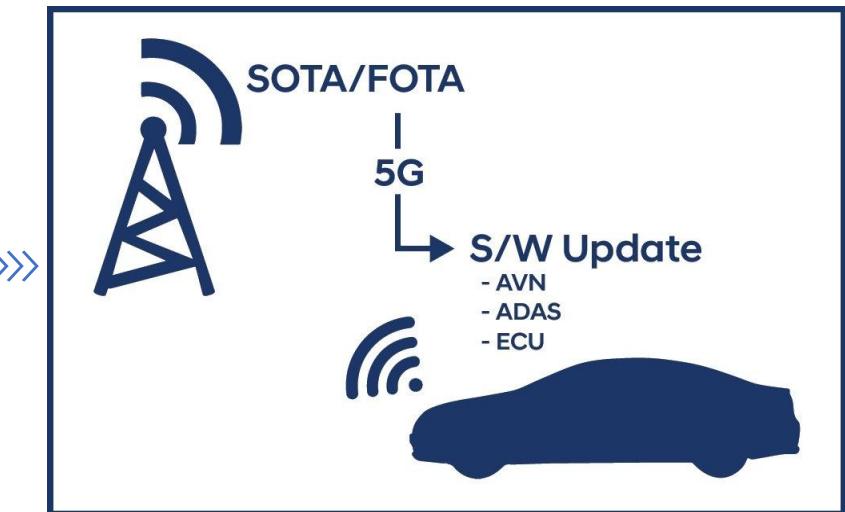
- 최신의 차량에는 제동, 조향 및 트랙션(구동력) 제어와 같은 다양한 안전 및 편의 관련 기능을 향상시키기 위해 약 150여개 이상의 ECU와 Sensor들이 내장되어 있음
- 특히, ECU는 특정 제어 기능을 수행하기 위해 하드웨어 뿐 아니라 소프트웨어가 포함되어 구성됨
- 제조사는 차량의 수명 주기 안전 및 보안과 관련하여 악영향을 줄 수 있는 소프트웨어의 결함을 수정하고 성능을 개선하기 위해 소프트웨어를 업데이트하는 기술을 의미



'무선 소프트웨어 업데이트'로 제어할 수 있는 주요 차량 센서 자료:지멘스



유선 업데이트



무선 업데이트

*OBD: On-Board Diagnostics

■ 로컬 업데이트

- 제조업체가 소유자에게 직접 서신, 이메일, 차량 내 알림 시스템을 통해 차량을 서비스 센터 또는 공인 정비소로 가져가도록 안내하는 [전통적인 소프트웨어 업데이트 방식](#)을 의미
- 정비사가 일반적으로 *OBD-II 포트를 통해 전용 도구를 사용하여 소프트웨어를 업데이트
- 시간과 자원이 효율적이지 않아 인건비가 많이 들고 고객의 불만이 높음

■ 무선 원격 업데이트 (OTA)

- OTA는 "Over-the-air"의 약자로서, [무선통신으로 소프트웨어를 업데이트하는 기술](#)
- 차량에 적용하면 정비소를 방문하지 않아도 자동차에 새 기능 추가, 오류 개선, 보안 강화 등이 가능
- 특히, 리콜(Recall)이 요구되는 일부 결함의 경우에도 차량을 엔지니어에게 전달하지 않고 OTA만으로 해결가능

구분	기존	OTA
업데이트 통보	영업사원·서비스센터	불필요
소프트웨어 배포	서비스 센터 배포	차량으로 바로 배포
차량 이용	센터 입고 후 가능	업데이트 종료 후 가능

■ SOTA (Software OTA)

- 구동계를 제외한 인포테인먼트 부분의 SOTA를 널리 사용하고 있음
- FOTA와 달리 업데이트를 진행하면서 기능을 이용할 수 있어야 하기 때문에 아래의 기능 등이 필요함
 - 메모리 이중화 기술
 - 백그라운드 다운로드 및 설치 기술
- 주로, 네비게이션 업데이트나 무선 통신을 활용한 서비스 기능들이 SOTA에 해당

■ FOTA (Firmware OTA)

- Firmware는 장치에 적용되어 구동에 영향을 주는 소프트웨어를 의미함
- 즉, 하드웨어를 다루고 작동시킬 수 있는 low level control을 할 수 있어, 반드시 필요한 체계임
- 대표적인 사례로 컴퓨터의 CPU, GPU의 펌웨어 업데이트와 ECU 펌웨어 업데이트 등이 있음

OTA

S-OTA
(Software OTA)



F-OTA
(Firmware OTA)

자동차 SW 증가 및 SDV 전환의 효과적 대응에 필수



SDV 전환

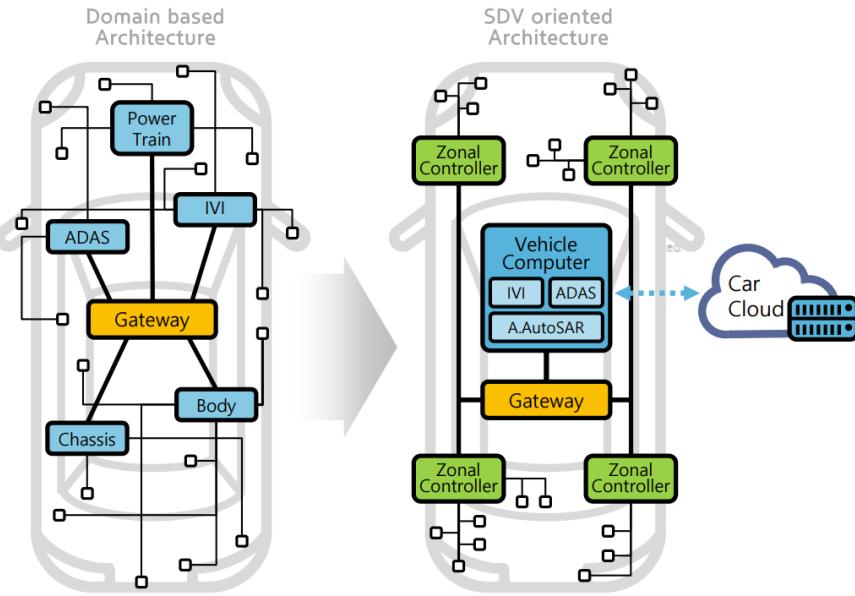
- SDV로 인해 빈번한 SW 업데이트가 필요해짐 → 업데이트 비용 증가
- OTA 적용을 위해 전반적인 플랫폼 변화 필요 → Zonal Architecture

ECU 제어를 위한 SW 증가

- 최신 차량 평균 150여개 이상의 ECU 사용
- SW 복잡성 증가에 따른 업데이트 필요
- SW 오류로 인한 Recall 대응

보안 및 업데이트 Regulation

- UNR.155, UNR.156
- ISO 21434, ISO 24089
- 유럽 수출 필수



'도메인 중심의 Architecture'의 한계

- 차량 내 기능 증가에 따라 컨트롤 장치 (DCU) 증가
- 차량 내 연결 배선의 증가로 복잡성/중량 증가
- DCU-부품 간 연동을 위한 HW/SW 복잡성 증가
- 기능 개선 향상을 위해 많은 비용과 시간 필요
- DCU 기반의 Architecture 유연성에 한계

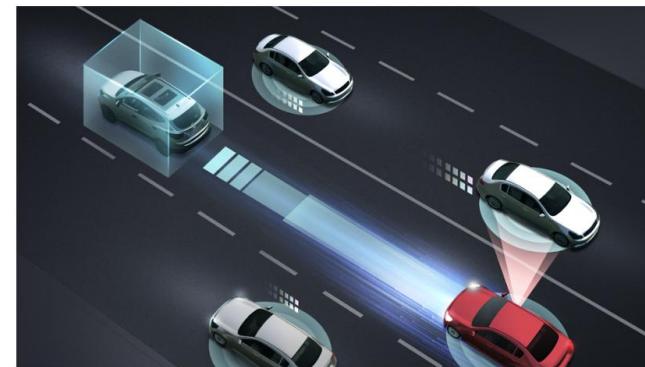
'SDV 지향 Architecture'로의 진화

- 여러 개의 도메인 컨트롤러를 하나의 고성능 제어기로 대체
- 배선의 단순화, 개발 비용과 차량 무게의 축소
- VC-Cloud 간 통신을 통한 용이한 OTA 업데이트
- HW의 추가보다 SW 정의에 의해 차량의 특성 결정

*Ref: 2023년 자동차공학회 전기전자시스템부문 워크샵, 현대모비스

■ 자율 주행에 필수이기 때문에

- OTA는 내비게이션 메뉴 UI (User Interface) 변경 같은 기본적인 소프트웨어 업데이트만을 위한 기능이 아니며, 주행 거리, 가속 능력, 차량 제어 기능, 주행 보조 기능, 디지털 Cockpit(콕핏) 기능까지 업그레이드할 수 있음
- 특히 자율 주행처럼 하루가 다르게 새로운 기능이 출시되고, 기존 단점이 보완되는 분야는 OTA가 필수
- 즉, 오늘 출고된 자동차도, 석 달 전 출고된 자동차라도 OTA를 통해 모두 최신 소프트웨어로 업데이트가 될 수 있음
- 자동차들끼리 서로 자동차 소프트웨어 기능과 버전이 같은 것으로 맞춰지는 것이 중요
 - 자율 주행을 할 때 도로의 자동차들끼리 서로 신호를 주고받으면서 주행 속도도 조절하고 방향도 바꾸고 긴급 상황을 관리하는데, 이때 어떤 자동차는 과거 소프트웨어 버전이라 제대로 신호를 주고받지 못한다면 사고로 이어질 수 있음
- 또한 자율 주행은 특정 자동차 한 대가 그 기능을 가지고 있다고 해서 가능한 것이 아니며, 도로 위 전체 자율 주행 자동차들과 교통 시스템이 모두 최신 기능으로 업데이트되어 서로 커뮤니케이션 가능한 상태여야 가능함



[Ref] <https://live.lge.co.kr/ota/>

■ 주행 빅데이터 수집에 필수이기 때문에

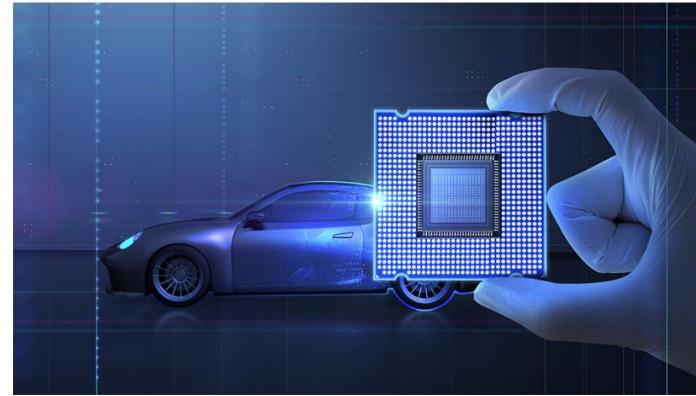
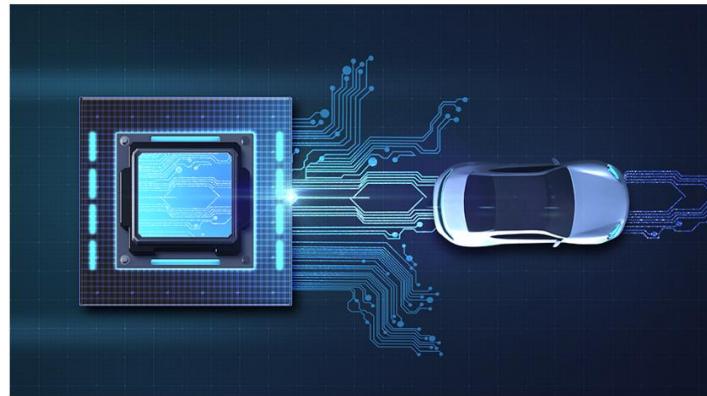
- OTA가 중요한 또 다른 이유 중 하나는 차에게 '빅데이터'라는 소중한 정보를 주기 때문
- OTA가 있기 때문에 자율 주행차가 주행 중 어떤 문제가 많이 발생하는지 알 수 있는 빅데이터 생성
 - 자율 주행차와 교통 시스템 간의 통신에 필요한 소프트웨어 실시간 업데이트나 자동차와 자동차 회사 간 기능 유지보수에 필요한 소프트웨어 실시간 업데이트 등을 통해 다양한 정보를 수집할 수 있음
- 수집된 데이터를 통해 '차선 변경 시 옆 차 인식 속도가 문제가 많이 생기는구나', '자동차 회사에서 보완한 이런 소프트웨어가 적용된 뒤로는 문제 발생이 없어졌네. 이런 기능이 핵심이구나'와 같은 데이터들을 분석할 수 있음
- OTA 덕분에 주행 전, 주행 중, 주행 후 시간 대 별로 민감한 기술 이슈를 실시간으로 파악 할 수 있음
- 이러한 과정이 궁극적으로, 자율 주행차를 더 완벽하게, 안전하게, 편리하게 완성시킬 수 있게 함



[Ref] <https://live.lge.co.kr/ota/>

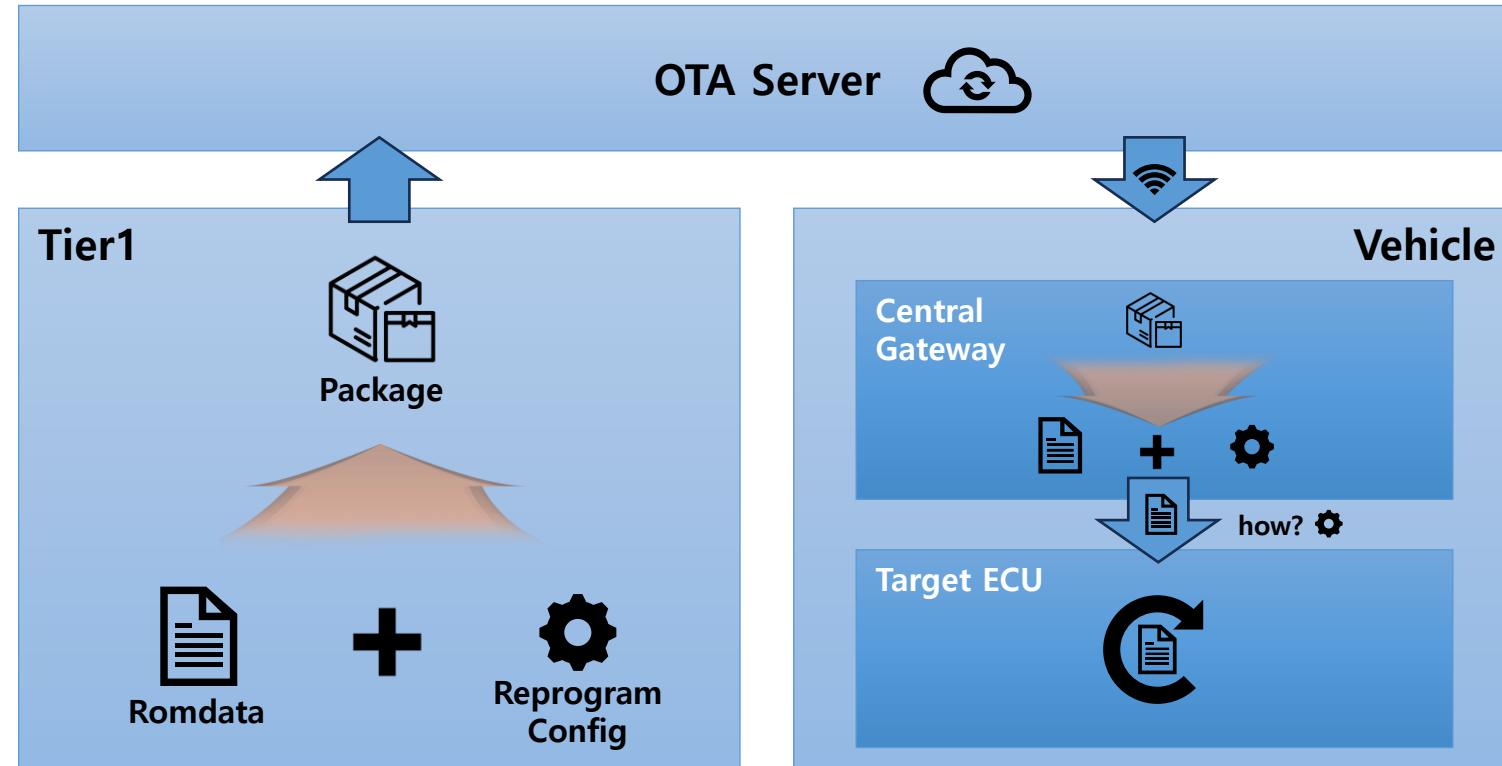
■ 자동차 고객과 회사의 시간, 비용, 노력을 아껴주기 때문에

- OTA 기능은 자동차 소프트웨어를 최신 상태를 유지해주므로, 운전자는 서비스 센터나 판매점을 찾는 횟수를 줄일 수 있음.
- 특히 소프트웨어, IT 비중이 획기적으로 늘어난 전기차는 물리적인 접촉 없이도 OTA로 업데이트 하기에 적합
- 따라서 OTA로 자동차 소프트웨어를 업데이트하는 것은 고객 뿐 아니라 자동차 회사의 시간, 비용, 노력도 아껴주는 효과가 있음



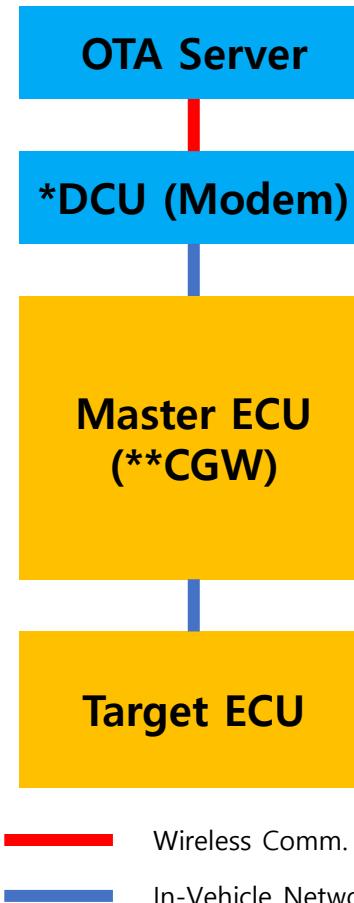
[Ref] <https://live.lge.co.kr/ota/>

OTA 개요 - OTA 절차



OTA 개요 - OTA 절차 (cont'd)

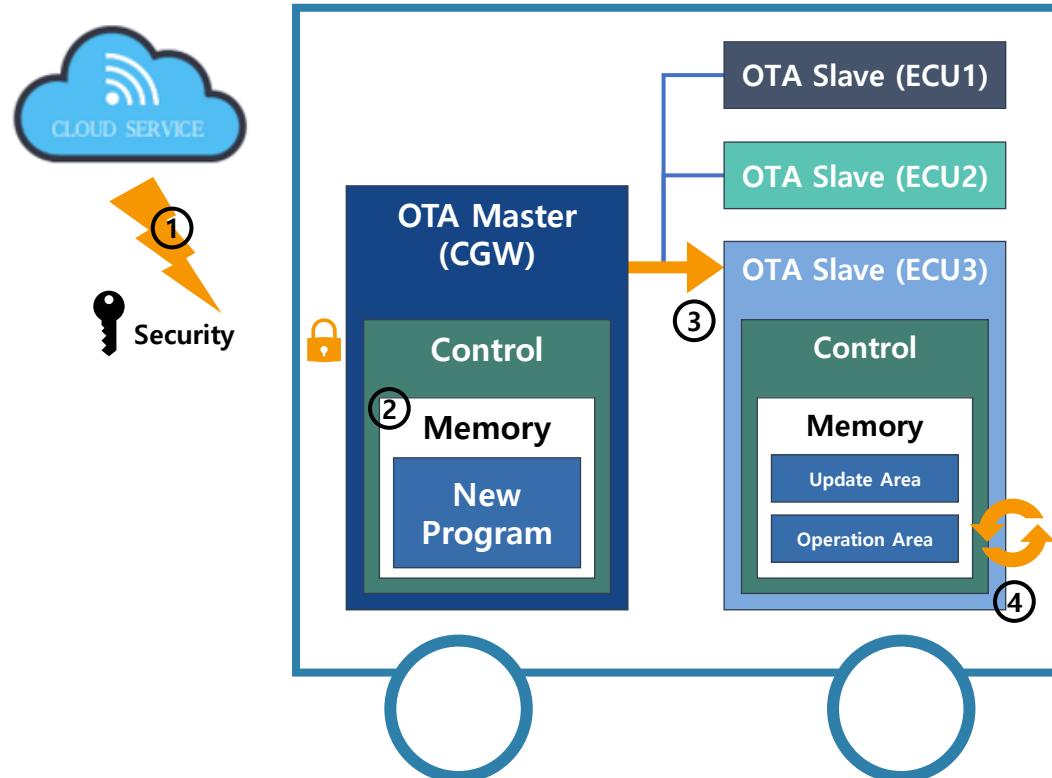
*DCU: Data Connectivity Unit
**CCU: Central Gateway
***VIN: Vehicle Identification Number



구성	설명
OTA Server	- ***VIN 및 제어기 버전 기반 OTA 실행 판단 - 제어기 ROM Data 및 진단법규 데이터 저장 - DM (Download Manager) Master
DCU (Modem)	- 서버와 무선통신 수행 (e.g., LTE, WiFi, WAVE, etc.)
Master ECU (CCU)	- DM Client 및 OTA Master를 포함 - 버전 체크, 무선 다운로드, 리프로그래밍 수행 - ROM file 관리, OTA 우선순위 판단 등을 수행
Target ECU	- 수행제어기 내 Reprogramming을 수행하는 로직으로 리프로그래밍 명령에 따라 수행제어기의 코드/데이터 영역을 삭제하고 관리 로직으로부터 수신한 ROM file을 쓰는 역할

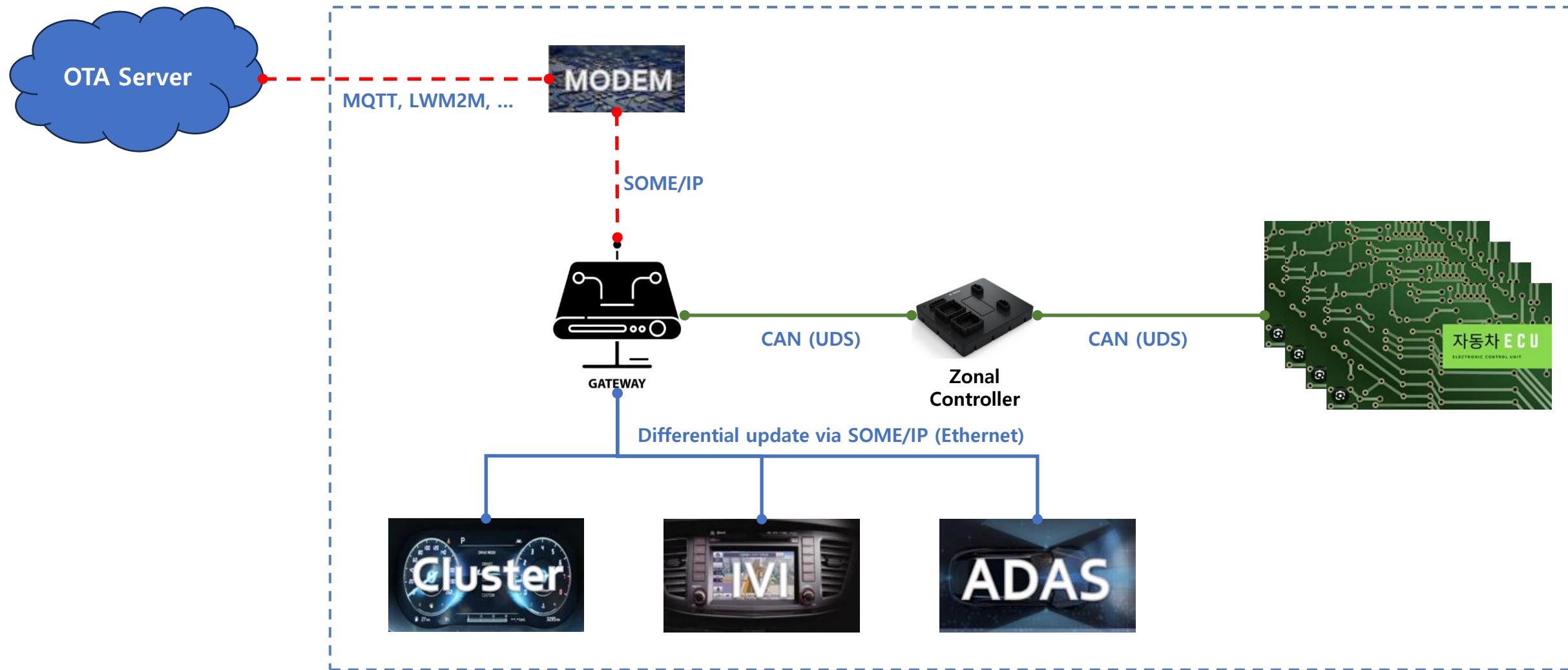
— Wireless Comm. (LTE)

— In-Vehicle Network



1. Cloud Service에서 업데이트할 ECU의 소프트웨어를 자동차의 OTA Master(CGW)로 보낸다.
2. Security Key 확인 후, OTA Master의 Memory에 업데이트 소프트웨어를 저장한다.
※Security Key: EVITA에서 정의하는 HSM (Hardware Security Module) Full Level에 의해 해킹에 대한 보안을 강화
3. 정지한 상태에서 New Program을 해당 ECU(OTA Slave)로 전송한다.
4. Update Data를 Flash Memory에 저장한다. 저장완료 후 Reset이 되면, 업데이트 소프트웨어로 동작을 한다.

OTA 개요 - OTA Architecture



■ 법적인 문제

- OTA는 **현행법상 국내에선 불법임**
 - 자동차 관리법에 따라 점검, 정비에 해당하는 OTA는 반드시 **등록된 사업자와 장소에서만 진행돼야 하기 때문**
 - 이에, OTA가 가능한 카 메이커들이 OIA 서비스를 실시할 수 있었던 이유는, 20년 6월 산업통상자원부 규제 샌드박스에 OTA가 추가되면서 **특례 승인을 거친 업체들에 한해 2년간 한시적으로 서비스 운영이 가능**하게 되었기 때문
- 23년 7월에 차량 사이버 보안과 소프트웨어 업데이트에 관련한 국내 법규가 별도로 통과될 예정이었으나, 지연이 되고 있는 실정이며, 해당 **법규가 통과되면 임시 승인이 아닌 정식 승인을 통해서 OTA 서비스 가능**함

현대차·테슬라 등, 무선 SW업데이트(OTA) '임시허가' 종료... "오는 7월부터는 '정식허가'로 서비스"

□ 박시하기자 | ○ 승인 2023.05.16 17:55 | ○ 댓글 0

5	<p>□ 자동차 전자제어장치 무선 업데이트 정비업소 외 허용</p> <ul style="list-style-type: none">○ (현황) 자동차 전자제어장치를 무선으로 업데이트 (OTA)하는 정비행위는 자동차 정비업소를 방문해야 가능○ (개선) 차량소유자가 정비업소를 방문하지 않고도 자동차 전자제어장치를 무선 업데이트를 할 수 있도록 허용	<p>자동차관리법 시행규칙 제 132조 개정('23.5.)</p>
---	---	--------------------------------------

자동차관리법 시행규칙 제132조 개정[사진=국토교통부 홈페이지 캡처]

테슬라 자동차 OTA 임시허가 연장

By 김아지트 ● 2023년 07월 20일

자동차 전자제어장치 무선 업데이트 서비스 OTA(Over-the-air) : 임시허가 연장 확인서

'자동차 전자제어장치 무선 업데이트 서비스(테슬라코리아)' 임시허가 연장 확인서	
기관명	산업통상자원부
부서명	규제샌드박스팀
담당자명	문정흔
생산일자	2023.07.19
문서번호	1450538-846
보존기간	3년
단위업무	규제샌드박스 사후관리업무
공개여부	비공개
분류체계	산업·통상·중소기업>산업진흥·고도화>산업기술 진흥>산업기술정책>산업기술정책·시책의 수립 시행>규제샌드박스 사후관리업무
본문파일	본 문서는 비공개 문서이므로 열람이 불가능합니다. 필요 시 청구신청 하시기 바랍니다.
* 원문파일은 스토어에서 PDF파일, hwp파일 등의 링크를 설치해서야 보실 수 있습니다.	

■ 보안 이슈

- 협력사들과 함께 모든 모듈에 대해 보안 검증을 거쳐야 함
- 테슬라는 애초에 OTA를 고려하고 자동차 설계를 한 것이라 2012년에 OTA 서비스가 가능 했었고, 테슬라 이외의 카 메이커는 혼자 모든 Control Unit을 개발하는 것이 아니고 사업상 경제적인 이유로 다른 모듈에 대한 협력사와 함께 진행하고 있음
- 이에, 다른 협력사들과 함께 OTA 관련 보안 검증을 다 거친 다음에 진행할 수 있기 때문에 협력사별 Control Unit에 한 번에 시스템을 맞추는 것은 시간이 많이 걸릴 수 밖에 없음



■ 현실적인 이유

- OTA라는 기술은 예전부터 있던 기술이고, 예전부터 있던 기술임
- 다만, 테슬라가 차에 적용을 적극적으로 한 것
- 기술의 난이도만 보면 그렇게 어려운 기술은 아니지만, OTA를 차에 적용하게 되면 난이도가 급격하게 높아짐
- 일반적인 차량들의 대략적인 구조는...
 - 메인프로세서(AP): 인포테인먼트 + 네비게이션
 - 카메라 모듈: ADAS 인식 등에 사용되는 프로세서 + 카메라 등의 조합
 - 각종 센서류: 서브프로세서(MCU) (차량제어, 초음파, 레이다, 등)
- 일반적인 차량 회사들이 차를 만들 때 위의 3가지 중 직접 만드는 건 거의 없고, 대부분 supplier들이 생산에서 공급함 → 여기서부터 문제가 발생함!

■ 현실적인 이유 (cont'd)

- 만약 자동차 제조사에서 소비자들이 원하니까 OTA 기능을 포함한다고 하면,
 - OTA를 어디에 포함할 것인지?, 통신 모듈은 어디에 넣을지?, 누가 주관해서 전체 시스템 인터페이스를 맞출 것인지?, 누가 업데이트를 할 것인지?
- 일반적으로 통신 모듈은 메인 프로세서(AP)에 자동차 제조사에서 포함해서 출시됨
- 따라서 OTA는 메인프로세서에서 진행이 되고, 이 프로세서만 업데이트하는 OTA는 가능함
 - 그래서, 이미 네비게이션 등의 OTA 업데이트는 가능
- 만약, 테슬라와 같이 전체 시스템을 업데이트 하려고 하면,
 - ADAS 성능이나 버그도 수정해야 하고,
 - 센서류들의 MCU들에도 펌웨어가 수정되어야 함

■ 현실적인 이유 (cont'd)

- 이러한 환경에서 OTA를 지원하려면,
 - 기존 차량 업체들이 supplier들에게 확실한 interface를 제공하고, 유지보수 되어야 함 → 현실적인 어려움
- 한 부품이 여러 시리즈에 걸쳐서 모두 동일하게 혹은 비슷하게 유지해야 하는데, supplier를 장기간 계약하고, 유지하는 것이 어려움
- 즉, OTA는 소비자 입장에서 원하는 기능이지만, 제조사 입장에서는 제대로 구현해서 넣으려고 하니, 지금 있는 시스템(e.g., E/E Architecture)을 갈아 엎어야 하고, 초기 비용과 유지 비용이 많이 소비될 것임

02

SDV와 OTA

■ Software Defined Vehicle (SDV)

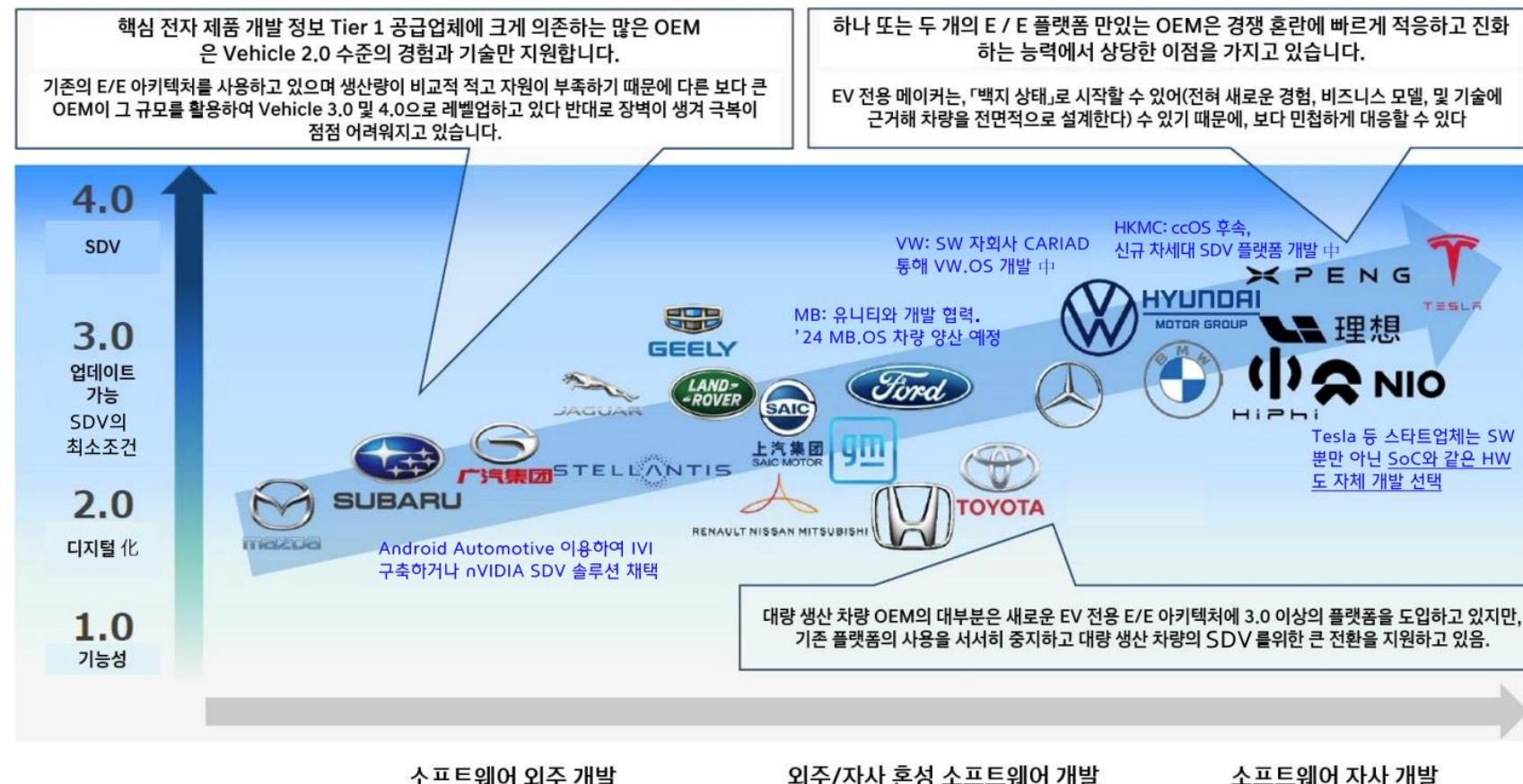
- 소프트웨어 중심으로 구동되는 이동수단을 의미하며, 기존의 하드웨어 기반의 자동차가 소프트웨어 중심의 전자장치로 지속적으로 변화해 온 결과이다.
 - SDV는 필요한 소프트웨어를 최적의 상태로 구동 시킬 수 있도록 하드웨어 스펙을 디자인한다.
 - 모델 연식에 초점을 맞춘 개발 사이클 대신 Agile 방법론이 지속적인 소프트웨어 개발을 주도하며, OEM은 출고 후에도 OTA 기능을 통하여 차량에 소프트웨어를 업그레이드 할 수 있어야 한다.
 - 차량 내 보안 소프트웨어는 사이버 공격을 탐지 및 방어하기 위해 훨씬 중요해지고, 시스템 내의 개별 구성 요소만을 보호하는 것이 아니라, 차량 전체 시스템을 보호해야 한다.

The Era of the Software Defined Car



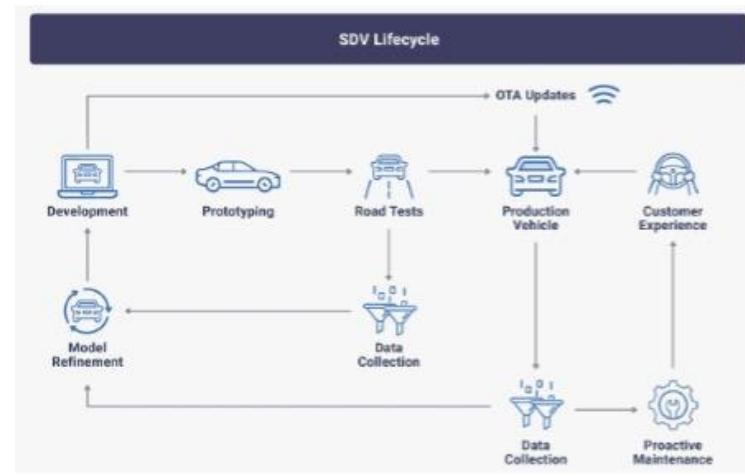
출처: <https://image.slidesharecdn.com/5gautonomouscar-160702144203/95/5g-and-autonomous-vehicle-3-638.jpg?cb=1467471040>

- 역사가 비교적 짧은 신규 전기차 브랜드는 White Label 상태로 SDV를 개발할 수 있어서 보다 민첩히 대응 중
- 각 OEM들은 비즈니스 규모, 소프트웨어 역량 등을 고려하여 여러가지 협업 모델로 추진 중



SDV 기술의 정의 및 의미

- ❖ SDV란 제품 수명주기 동안 OTA를 통해 성능/기능을 개선 및 추가, 다양한 서비스와 컨텐츠를 제공하여,
 - “고객”에게 지속적으로 **새로운 경험의 가치**를 줄 수 있는 차량,
 - “OEM”에게 개발 자원 절약 및 데이터 활용으로 제품 수명주기동안 **지속적인 수익을 창출** 하는 차량
 - “부품사”에게 HW규모의 경제와 **SW의 부가가치를 창출**하는 차량,
 - “IT/SP*”에게 APP 또는 플랫폼 **생태계를 확장할** 수 있는 차량
- ❖ 자동차의 수명 주기가 연장되고 SOP** 이후에도 성능 향상 가능

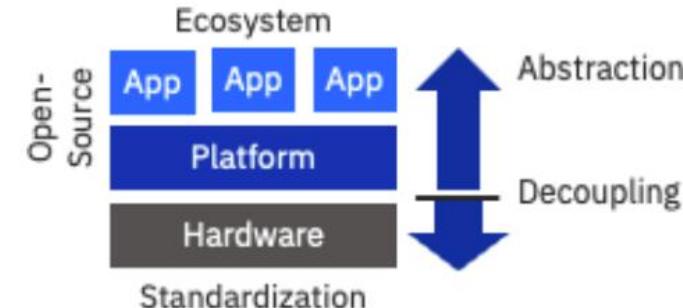


SDV 기술의 BASE

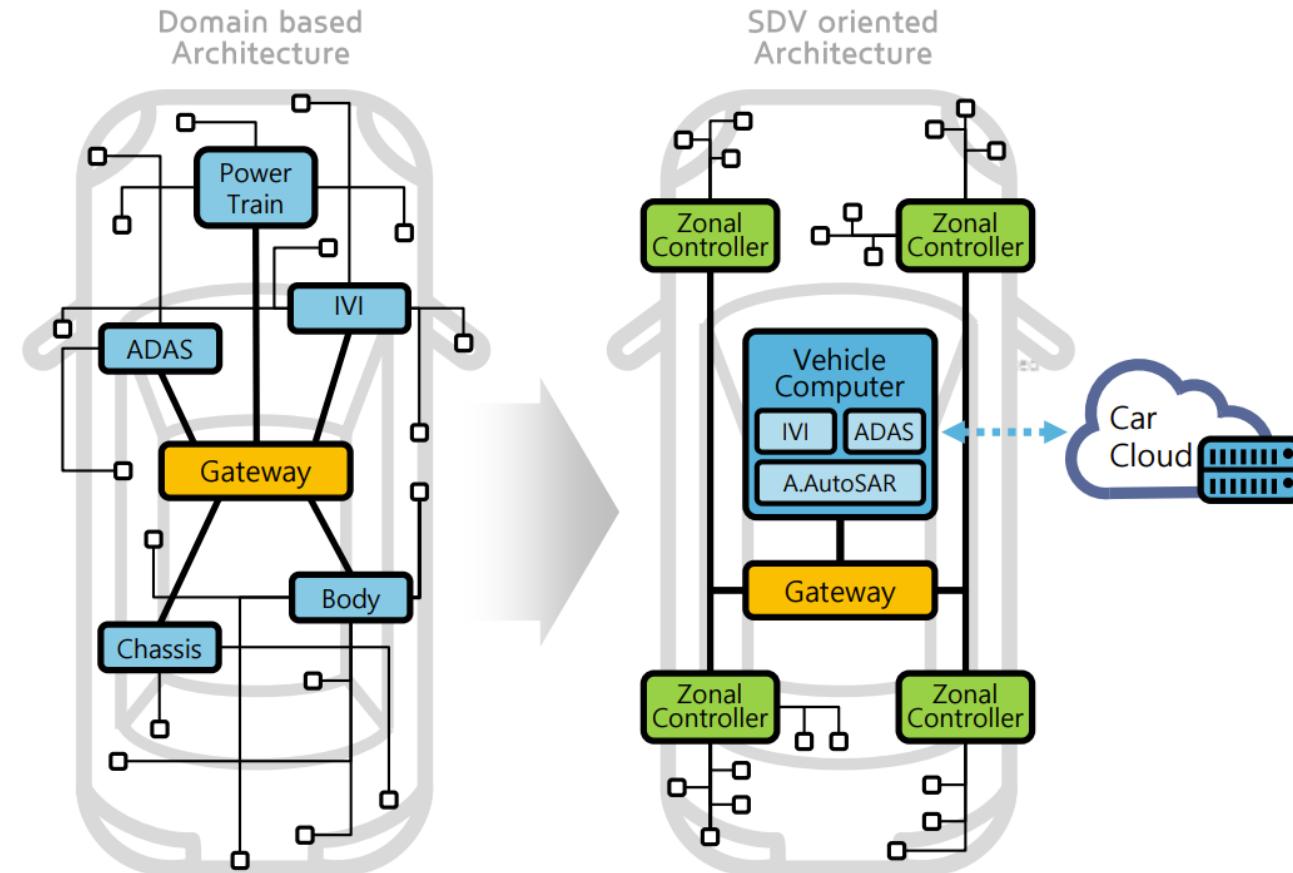
- ❖ SDV 기술의 기본(Base) 정의: 구성 요소
 - **ECU 통합화 및 중앙화 E/E Architecture** 기반으로 차량 복잡도와 비용 감소
 - **하드웨어 – 소프트웨어 디커플링**을 통한 새로운 서비스/기능 제공
 - 소프트웨어 플랫폼 / 클라우드 서비스로 차량 기능 확장, 다양화
 - 차량 소프트웨어 **OTA 업데이트**로 차량 수명 주기와 성능 증대



- ❖ 디커플링, 추상화, 표준화 및 오픈소스를 통해 앱생태계 생성



SW 정의에 따라 차량의 특성이 정의 가능한 아키텍처 확보 필요



* DCU (Domain Control Unit)

'도메인 중심의 Architecture'의 한계

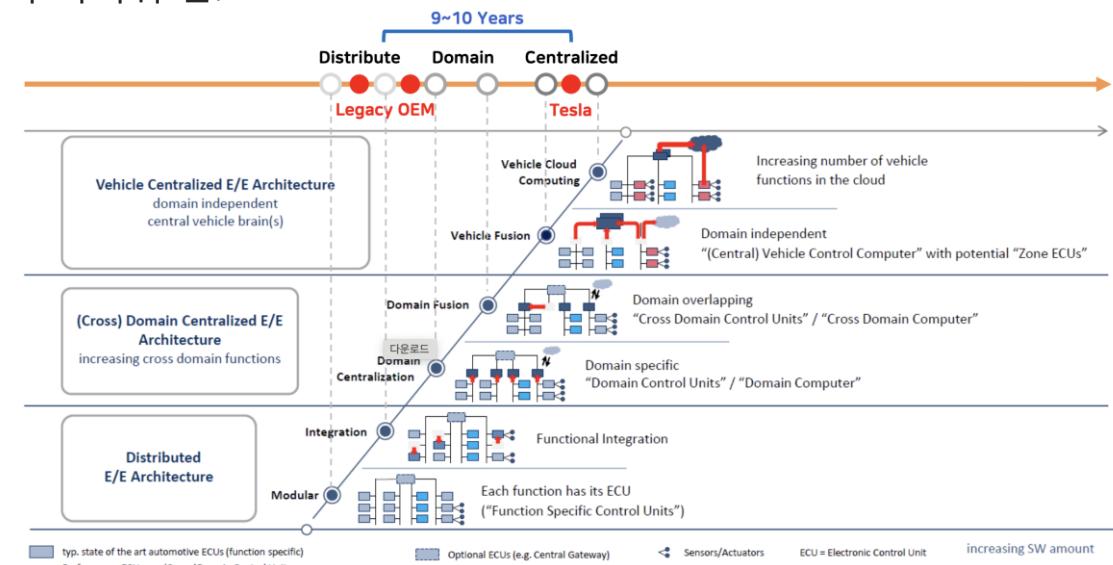
- 차량 내 기능 증가에 따라 컨트롤 장치 (DCU) 증가
- 차량 내 연결 배선의 증가로 복잡성/중량 증가
- DCU-부품 간 연동을 위한 HW/SW 복잡성 증가
- 기능 개선 향상을 위해 많은 비용과 시간 필요
- DCU 기반의 Architecture 유연성에 한계

'SDV 지향 Architecture'로의 진화

- 여러 개의 도메인 컨트롤러를 하나의 고성능 제어기로 대체
- 배선의 단순화, 개발 비용과 차량 무게의 축소
- VC-Cloud 간 통신을 통한 용이한 OTA 업데이트
- HW의 추가보다 SW정의에 의해 차량의 특성 결정

■ Distributed E/E Architecture → Centralized E/E Architecture

- 차량 내 전자 장치(ECU)의 증가
 - ECU가 증가하면서 배선과 통신이 복잡해지고, 비용이 증가함 → ECU의 수를 줄이고 네트워크의 단순화가 필요해짐
- ADAS의 발전
 - 기존에는 BSD(후측방 경보), AVM(어라운드 뷰) 등 센서 별로 회사가 나누어 따로 제작하여 납품하였음.
 - 그러나 SCC(스마트 크루즈), AEB(자동긴급제동장치) 등 ADAS가 등장하면서 카메라와 RADAR 센서를 융합하게 되었고, 더 이상 센서 별로 나누어 개발하기 어려워짐.



Automotive E/E Architecture (Trends of Future E/E Architecture, BOSCH)

(OTA 관점에서...)

SDV란 자동차 내부 다수의 ECU에 OTA를 적용할 수 있는 환경(??)

OTA의 필요성

- SDV는 SW 중심의 자동차를 개발하는 것으로, 빈번한 SW 업데이트가 필요 → 업데이트 비용 증가
- 구독 서비스 등 SW 일부 기능 업데이트에 대한 요구사항 증가
- SW 개발 다변화 요구사항 증가 → Tier1 도움 없이 SW 개발 배포 필요

OTA의 장점

- 필요 기능 최소 업데이트 가능 → 업데이트 비용 최적화
- SW 개발 다변화 가능 → Tier1 의존성 감소

자동차에서의 OTA

- 다수의 ECU에 적용
- 하드웨어 및 운영체제의 다양성
- OTA 적용을 위한 전반적인 플랫폼 변화 필요 (Zonal Architecture 등)
- 안전을 고려하여, 스마트폰 대비 엄격한 SW 관리 정책 필요

03

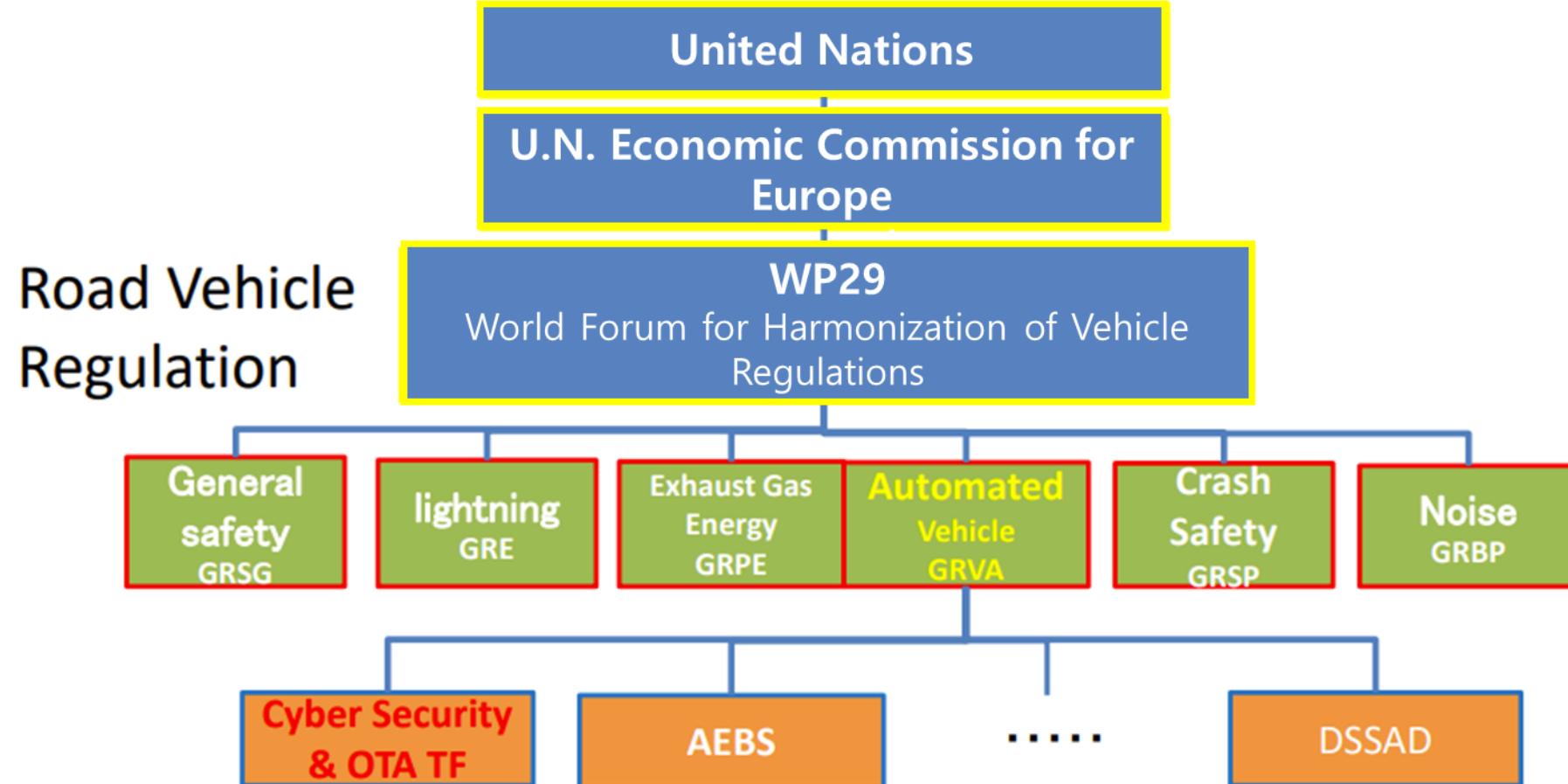
OTA 관련
법규와 표준

■ 자동차의 변화

- 차량의 자동화, 커넥티비티(Connectivity), 공유 모빌리티 제공 및 자율주행 차량 개발 등 차내 시스템의 디지털화/소프트웨어화로 필드 문제에 대한 개선 및 기존 기능 업그레이드 등 모바일 제품과 같이 신속한 SW업데이트 필요 (OTA)
- 차량 전자시스템에 대한 불법 해킹 시도 및 Volkswagen(VW) 배기ガ스 관련 불법 개조 사례 발생 등 차량 SW에 대한 보안 안정성 및 미승인 위/변조에 대한 대책 및 업데이트에 대한 법규 규제의 필요성 대두

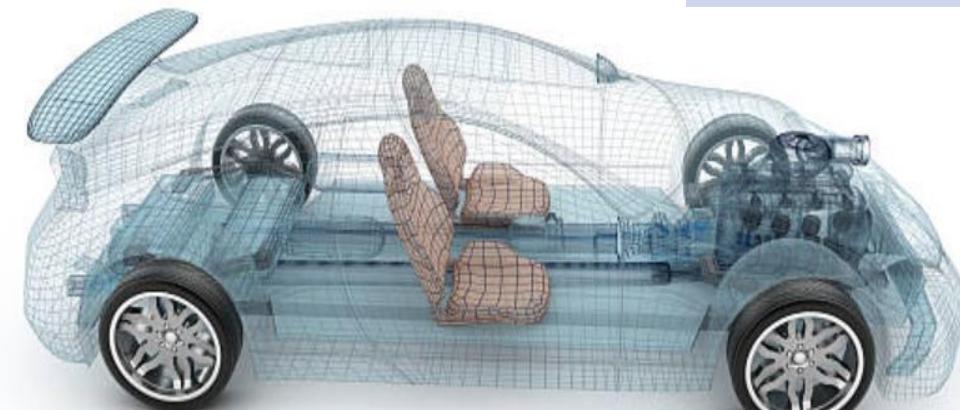
■ 법규 발효

- 양산된 차량에 대해 SW 변경 시, 차량 형식 승인과 관련성을 확인하고 안전하게 업데이트 이루어 질 수 있도록 규제하기 위해, UNECE에서 WP가 구성되어 "소프트웨어 업데이트 법류"를 작성하고 최종 초안이 '20.3.4에 배포되었고 WP29 181차 총회('20.6.4)에서 채택되어 본 내용으로 법규 발효 됨 ('21.1.22)
- 법규의 주요 요구사항은 다음과 같음
 - 제조사는 소프트웨어 업데이트 전반에 걸쳐 프로세스를 정립하고 산출물들을 관리하고 이에 대한 제조사별 소프트웨어 업데이트 관리 체계에 대한 R156.00 법규 준수 인증 획득 필요
 - 차량형식에 법규 별 소프트웨어 식별자, 업데이트 보안 및 OTA 기능 요건을 반영하고 차량 별 형식승인(VTA) 획득 필요



OTA 관련 표준 – 자동차 관련 법규 (UN Regulation 위주)

UN Reg. No. 155	UN Reg. No. 156	UN Reg. No. 79	UN Reg. No. 10	UN Reg. No. 94
Cybersecurity for vehicle	Software update	Steering effort/ADAS	EMC	Frontal impact
UN Reg. No. 28	UN Reg. No. 142		UN Reg. No. 13H	UN Reg. No. 137
Audible warning device	Type installation		Brake (회생제동)	Full width frontal impact
UN Reg. No. 141	UN Reg. No. 16			UN Reg. No. 95
TPMS	Seatbelt			Side impact
UN Reg. No. 46	UN Reg. No. 51			UN Reg. No. 135
Rearview mirrors	Sound Level			Rear impact for HEV
UN Reg. No. 127				UN Reg. No. 138
Pedestrian protection				Motor: Electric consumption
EU 2019/2144		EU 2022/545	UN Reg. No. 100	
GSR		EDR	Electric safety for REESS	

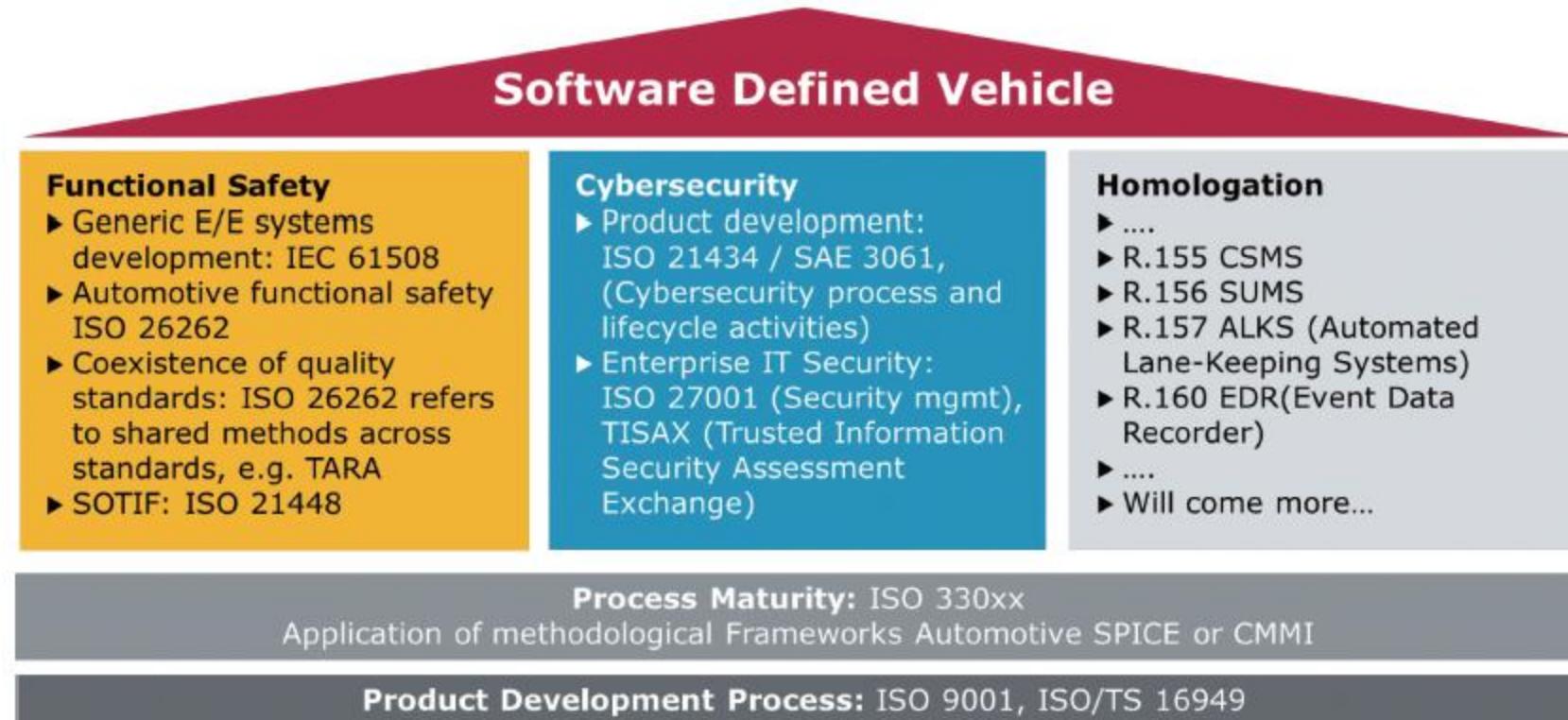


차량공통인증법규

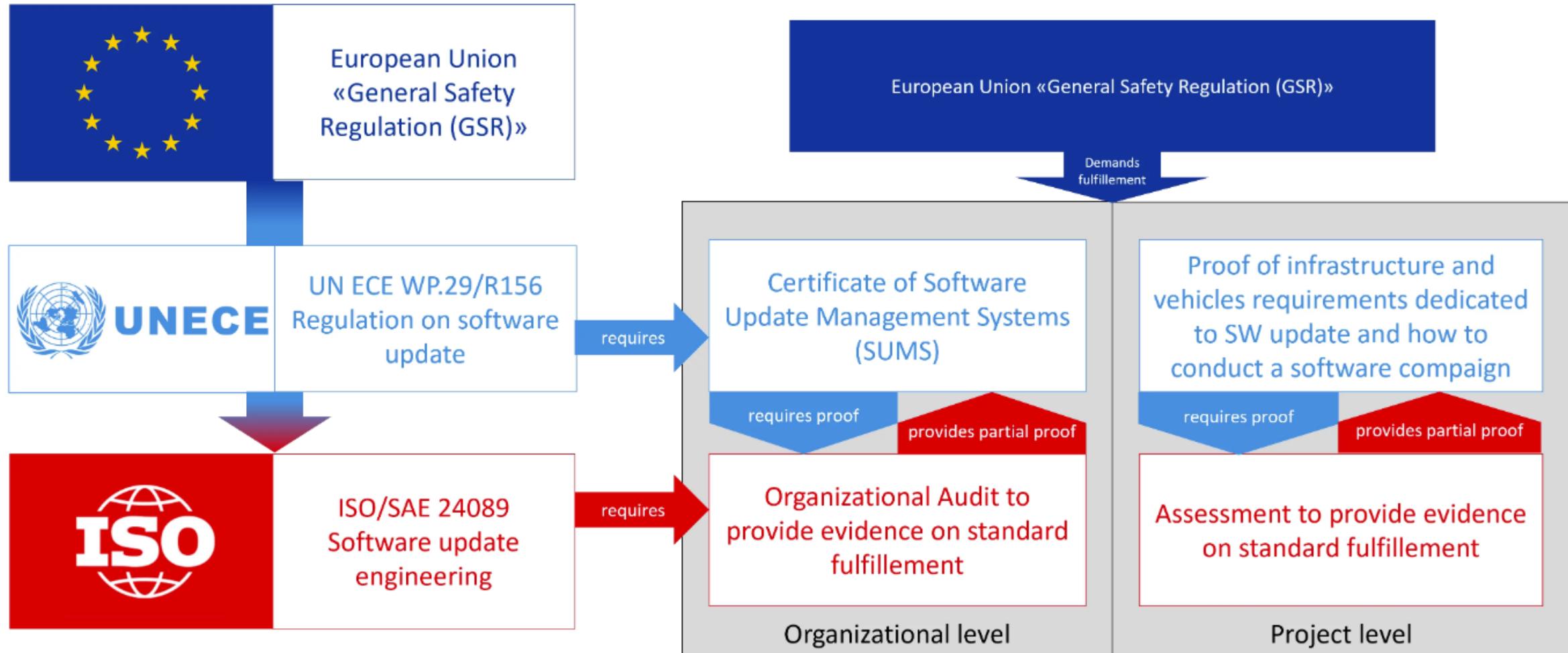
GSR (유럽안전규정)

ADAS관련인증법규

전기차관련인증법규



OTA 관련 표준 - OTA를 위한 법규 및 표준

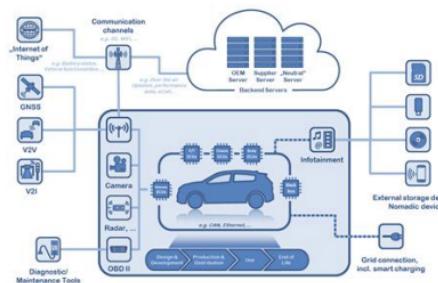


[Ref] <https://certx.com/automotive/software-update-for-road-vehicles-ep-1-overview-of-un-r156-and-iso-24089/>

Need both CSMS/SUMS Certification (management system) and Type approval

CSMS/SUMS certification

※CSMS: Cyber Security Management System
SUMS: Software Update Management System



- Acquired as an organization
- Valid for 3 years
- Related servers, factories, services, etc. are also subject to risk analysis
- Business management system implemented and practiced?

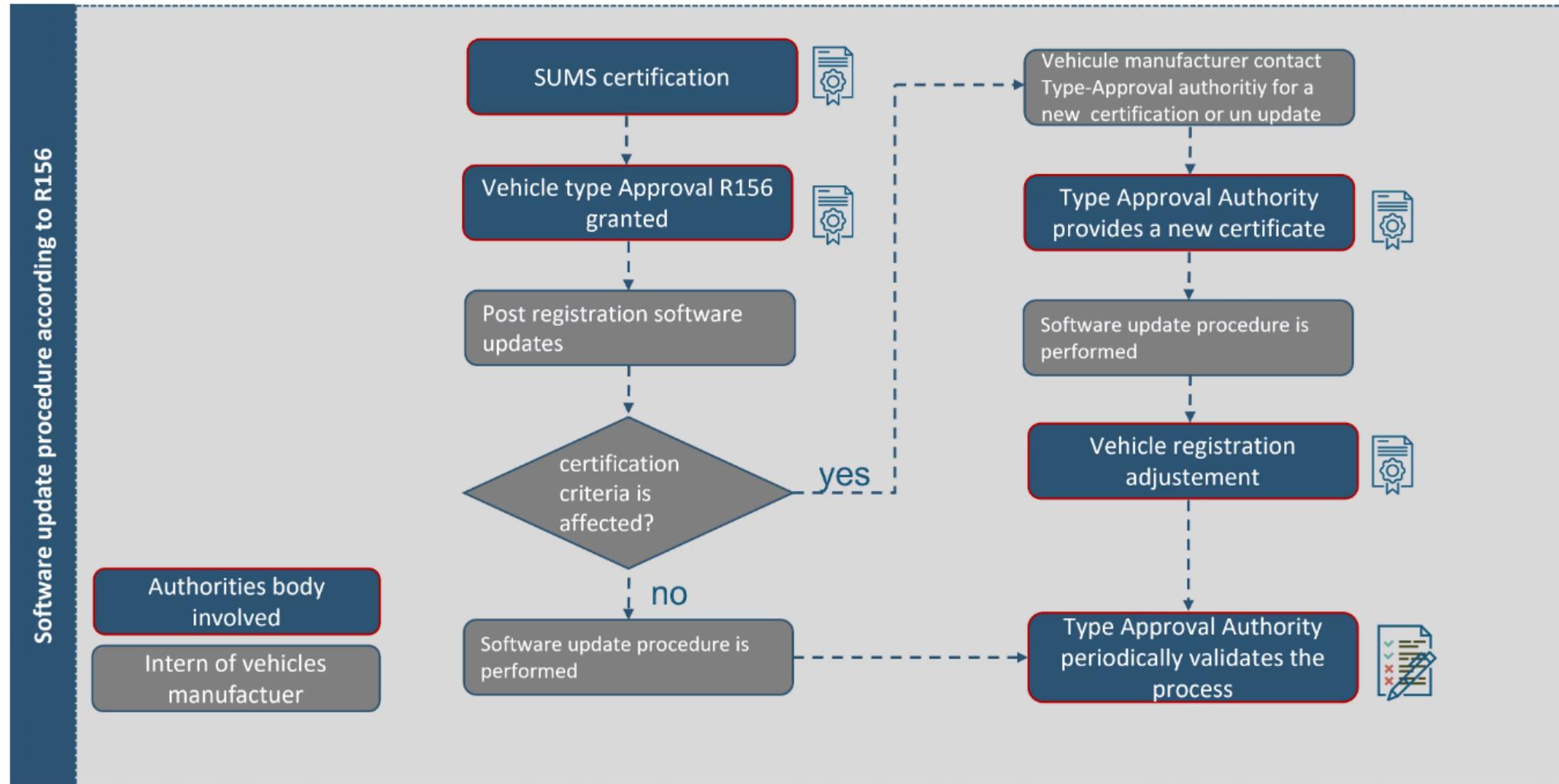
Type approval



Acquired for each vehicle type

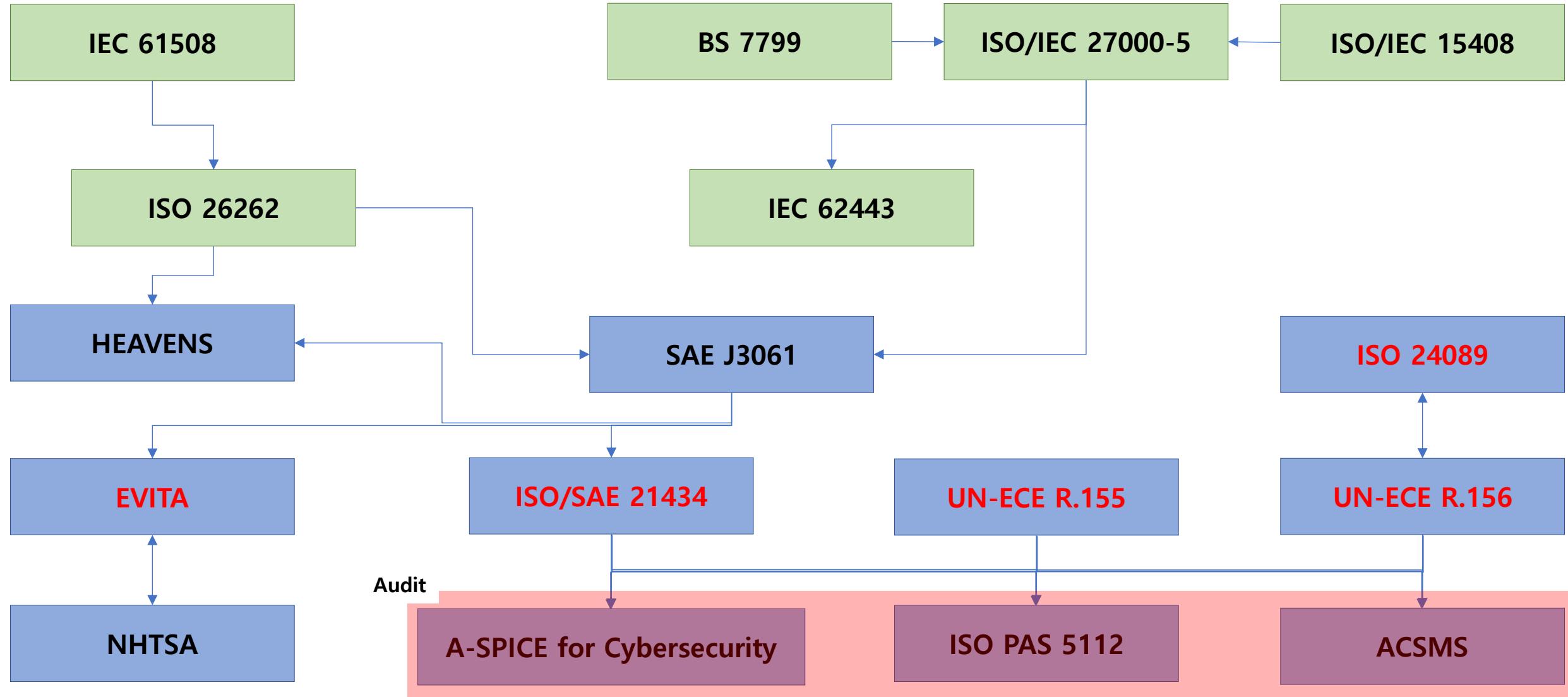
- Are there any development results according to the process?
- Results of implementation of type requirements for target vehicles, countermeasures, and evaluation
- Verification of effectiveness of measures (vehicle test)

OTA 관련 표준 - 법류 관점에서 SW 업데이트 절차



[Ref] <https://certx.com/automotive/software-update-for-road-vehicles-ep-1-overview-of-un-r156-and-iso-24089/>

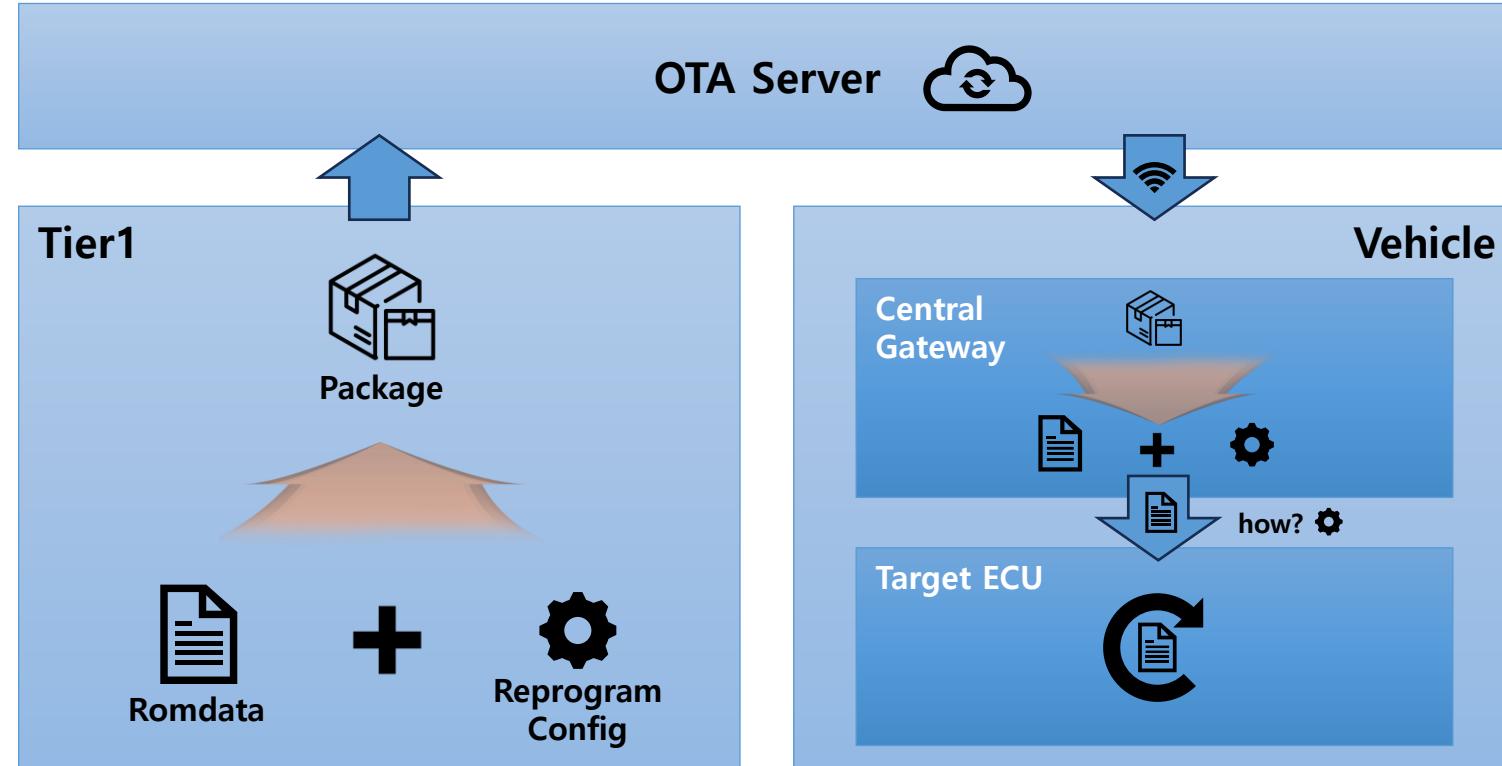
OTA 관련 표준 - 자동차 사이버보안 관련 표준 및 법규 관계도



04

OTA 프로세스
개요

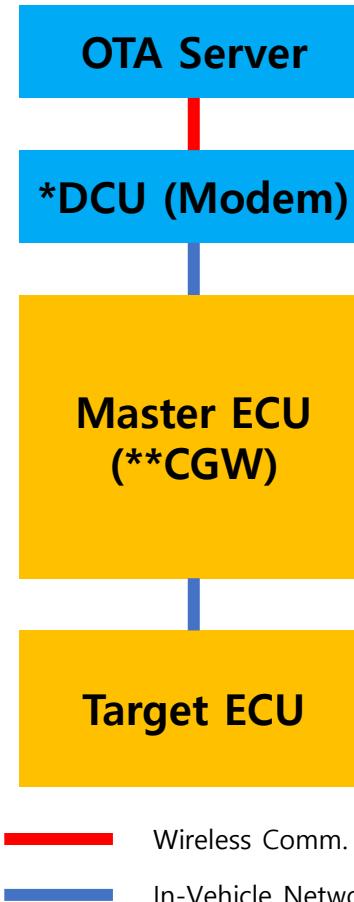
OTA 프로세스 개요 - Overview



*Ref: OTA기술세미나, 현대오토에버

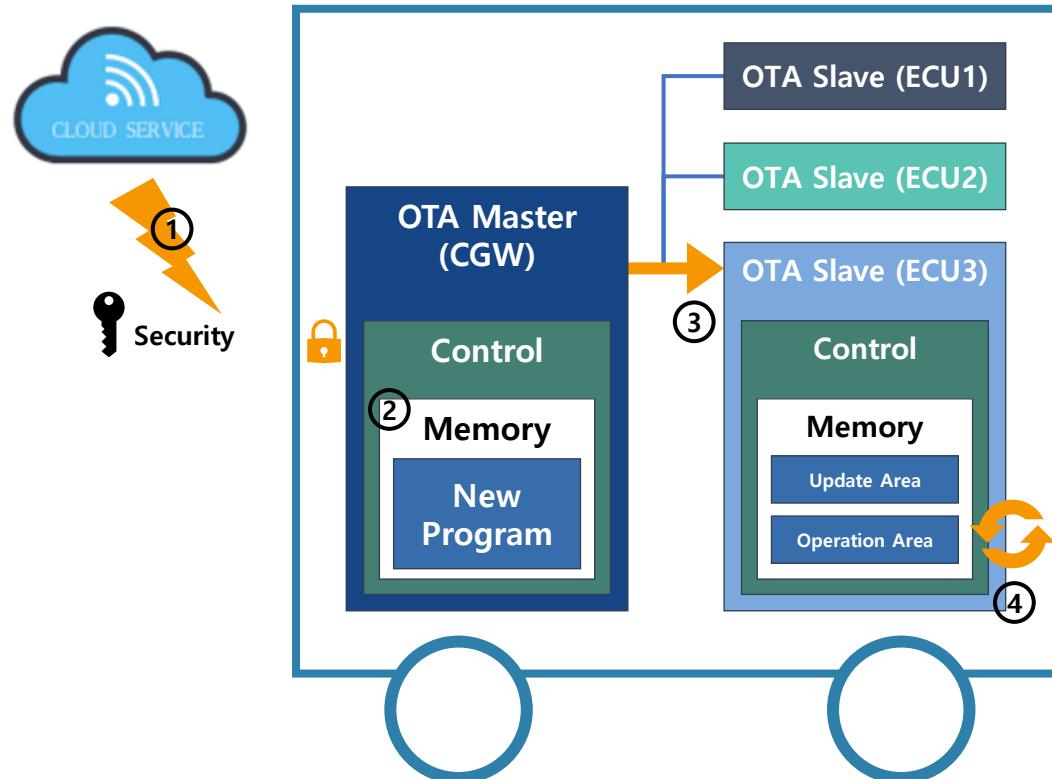
OTA 프로세스 개요 - In vehicle

*DCU: Data Connectivity Unit
**CCU: Central Gateway
***VIN: Vehicle Identification Number



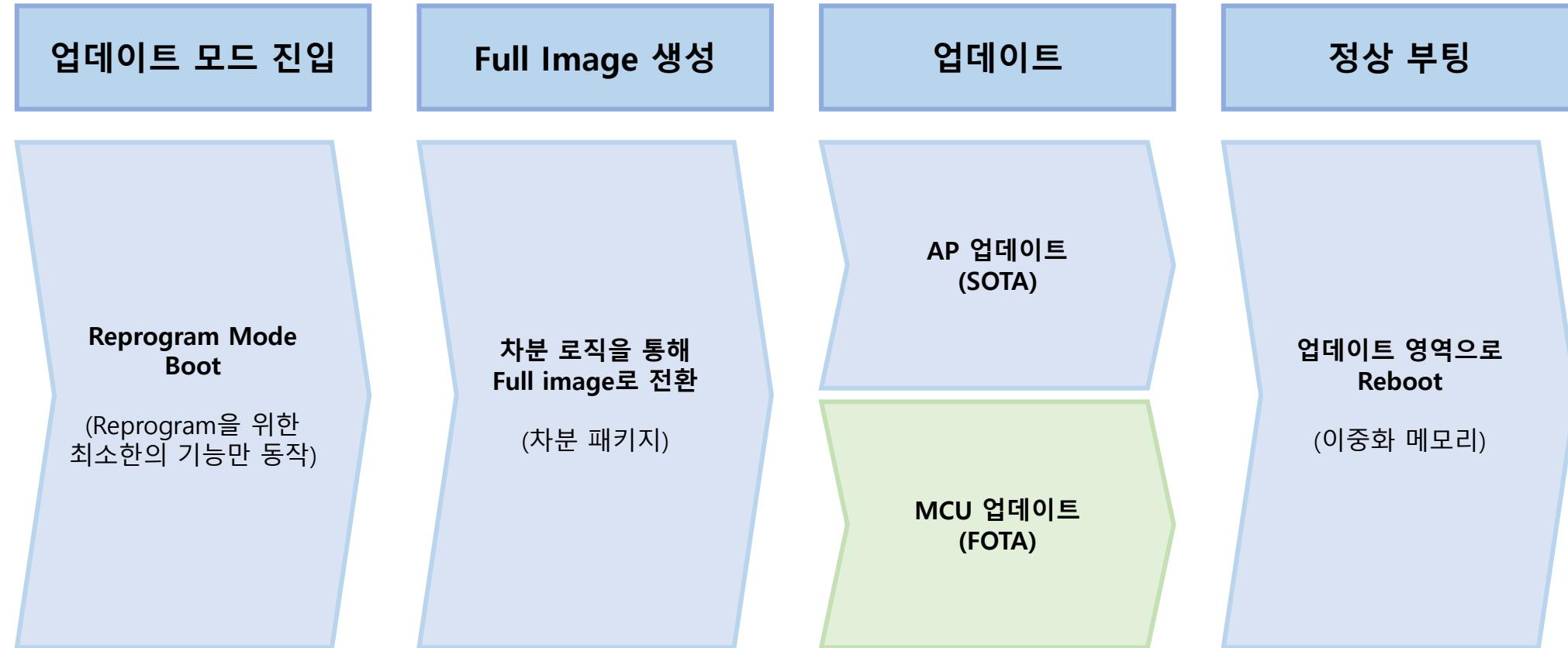
구성	설명
OTA Server	- ***VIN 및 제어기 버전 기반 OTA 실행 판단 - 제어기 ROM Data 및 진단법규 데이터 저장 - DM (Download Manager) Master
DCU (Modem)	- 서버와 무선통신 수행 (e.g., LTE, WiFi, WAVE, etc.)
Master ECU (CCU)	- DM Client 및 OTA Master를 포함 - 버전 체크, 무선 다운로드, 리프로그래밍 수행 - ROM file 관리, OTA 우선순위 판단 등을 수행
Target ECU	- 수행제어기 내 Reprogramming을 수행하는 로직으로 리프로그래밍 명령에 따라 수행제어기의 코드/데이터 영역을 삭제하고 관리 로직으로부터 수신한 ROM file을 쓰는 역할

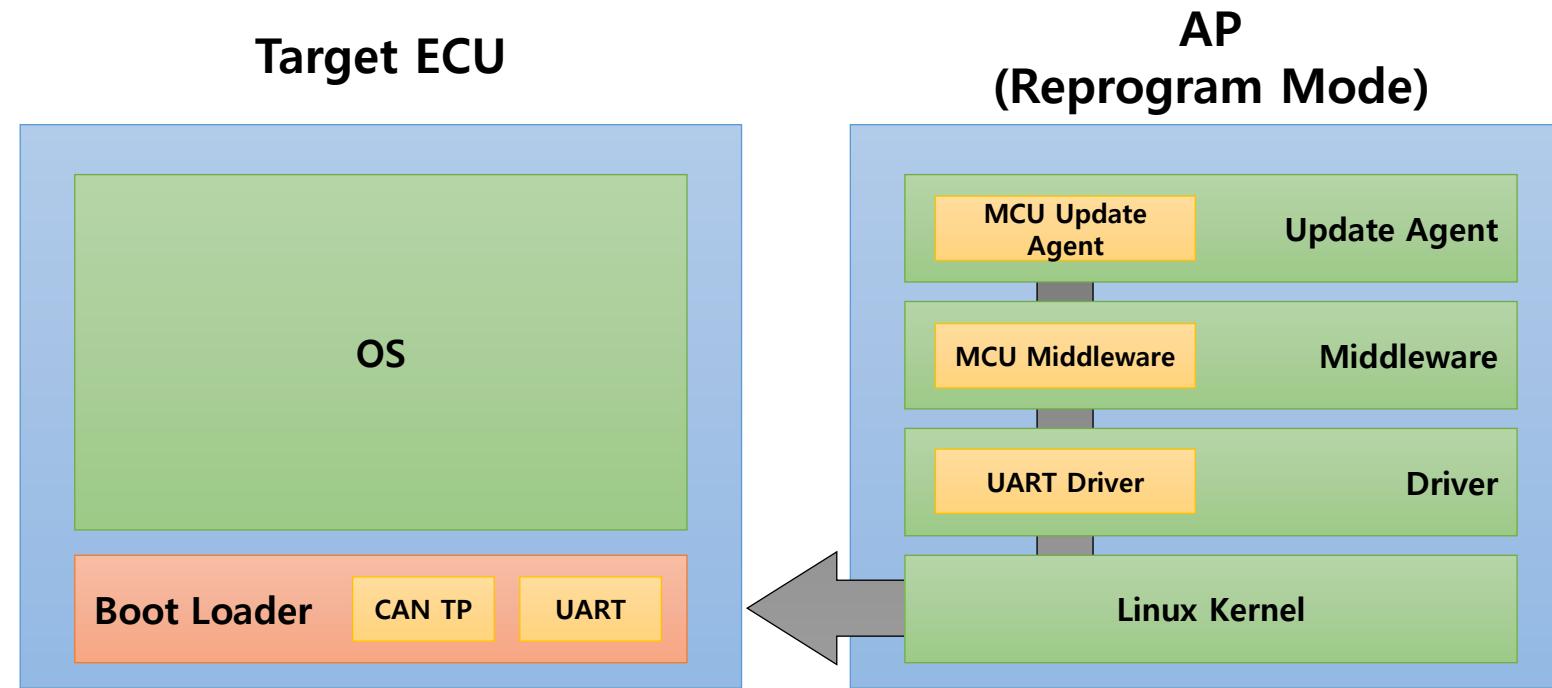
OTA 프로세스 개요 - ECU



1. Cloud Service에서 업데이트할 ECU의 소프트웨어를 자동차의 OTA Master(CGW)로 보낸다.
2. Security Key 확인 후, OTA Master의 Memory에 업데이트 소프트웨어를 저장한다.
※Security Key: EVITA에서 정의하는 HSM (Hardware Security Module) Full Level에 의해 해킹에 대한 보안을 강화
3. 정지한 상태에서 New Program을 해당 ECU(OTA Slave)로 전송한다.
4. Update Data를 Flash Memory에 저장한다. 저장완료 후 Reset이 되면, 업데이트 소프트웨어로 동작을 한다.

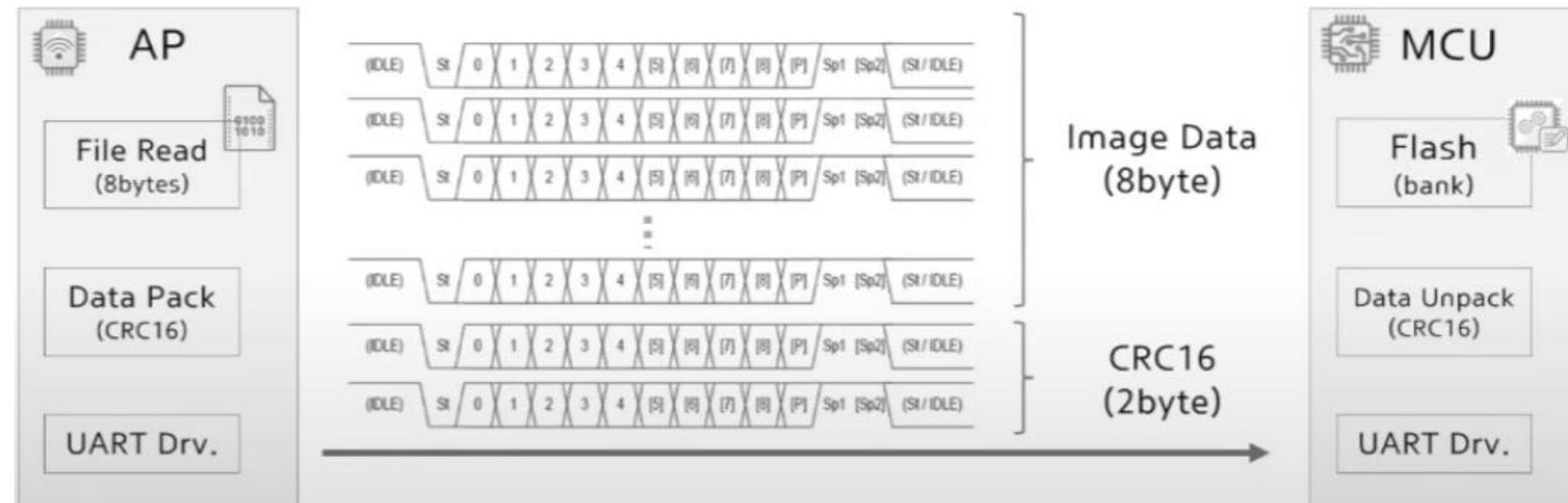
OTA 프로세스 개요 - Target Update





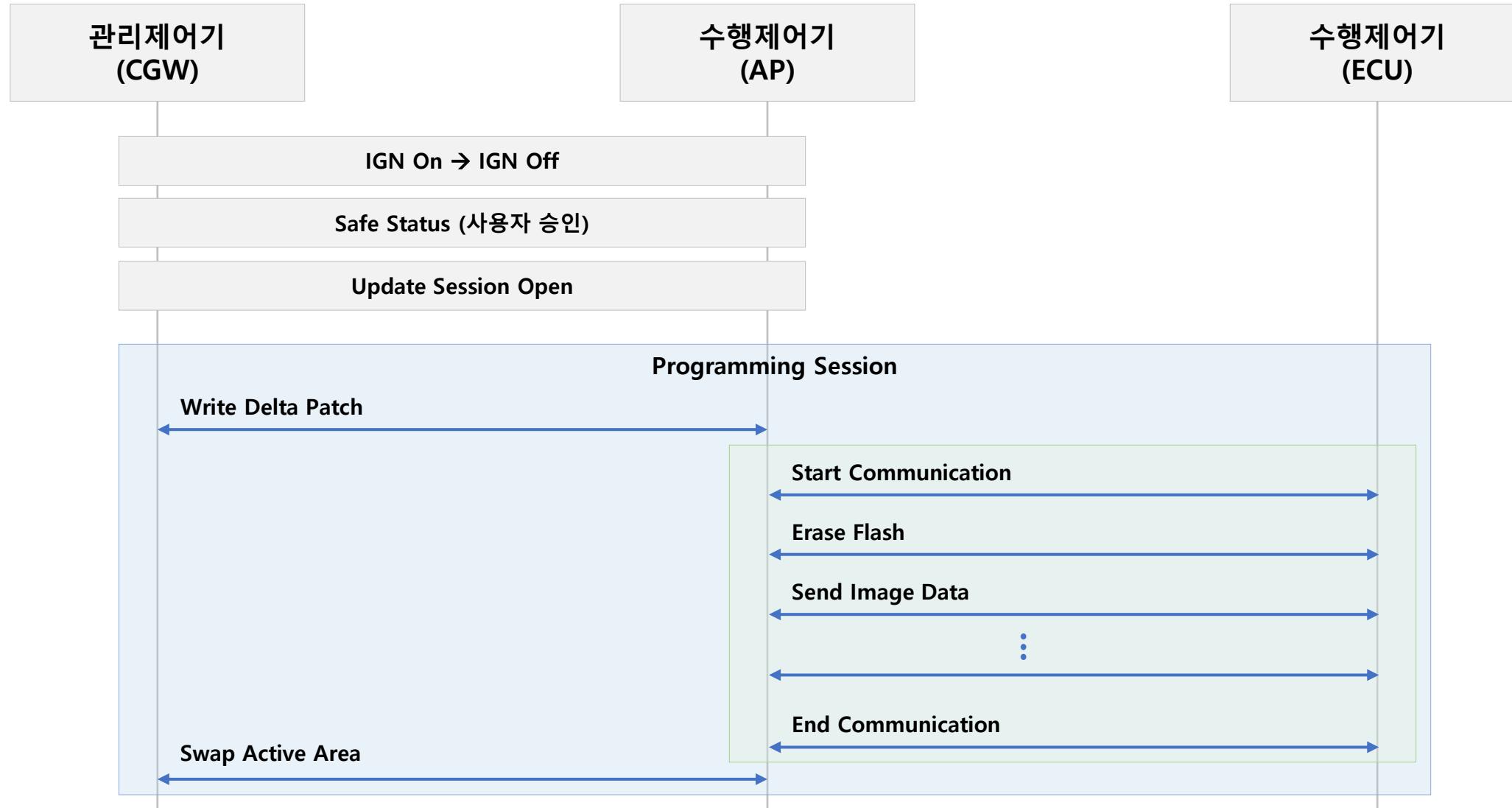
■ 업데이트 시 UART 통신 에러로 인하여 BIN 파일 오전달 될 수 있음

- UART 패킷에 CRC16 적용 (매 패킷 송수신시 CRC 검출 로직 적용)
- Flash Memory Write 시 Data Validation Check (mobilgence에서 제공하는 기능)

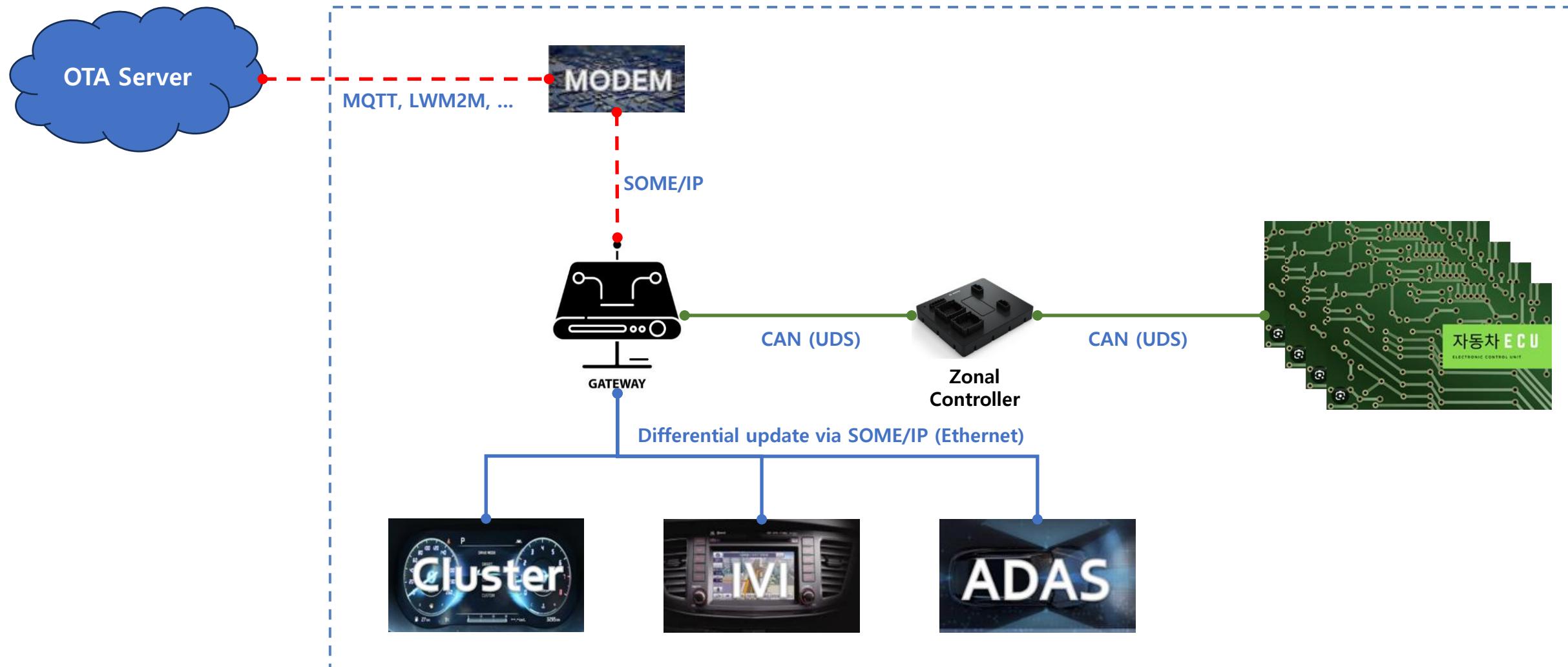


CAN TP를 그대로 사용하였기 때문에 한 프레임을 8byte씩 묶어서 송신함

OTA 프로세스 개요 - Target Update (cont'd)

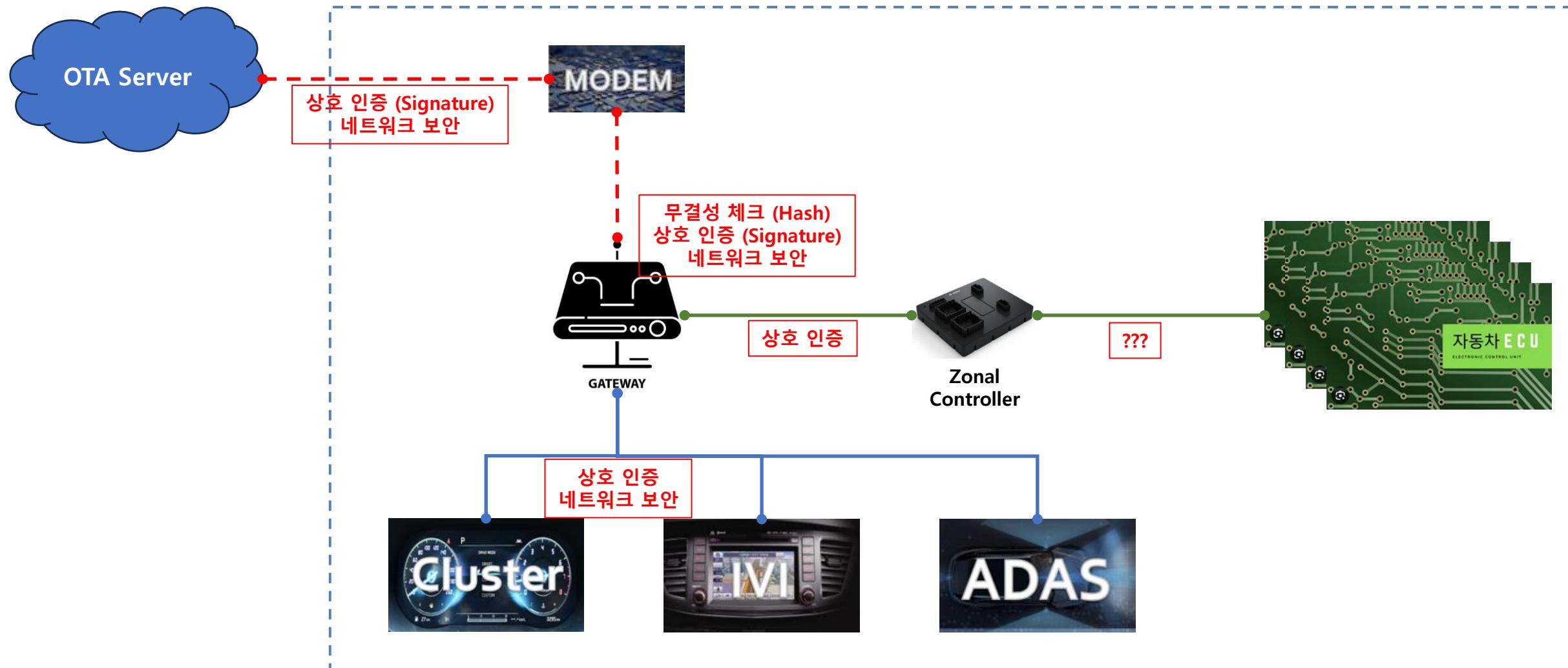


OTA 프로세스 개요 - Detailed Architecture



*Ref: 자동차공학회 전기전자시스템부문 워크샵, 현대모비스

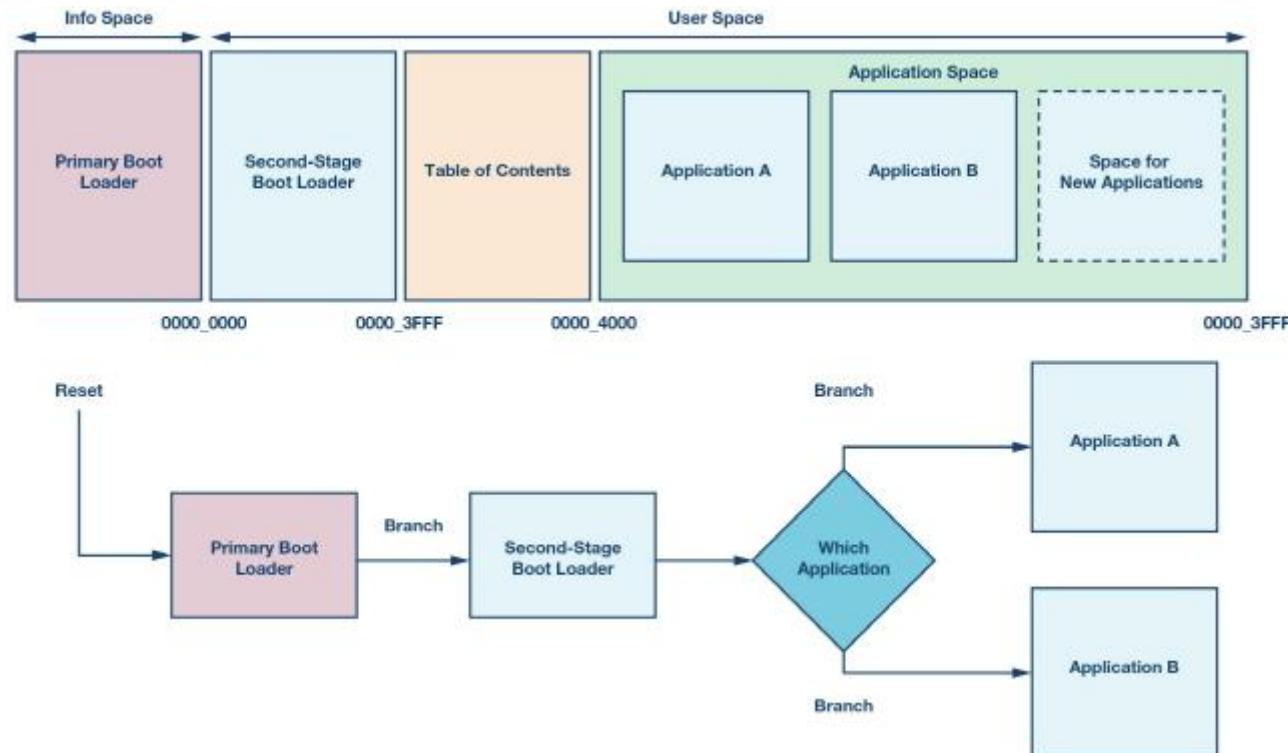
OTA 프로세스 개요 - Detailed Architecture



■ 부트 시퀀스에 대한 이해

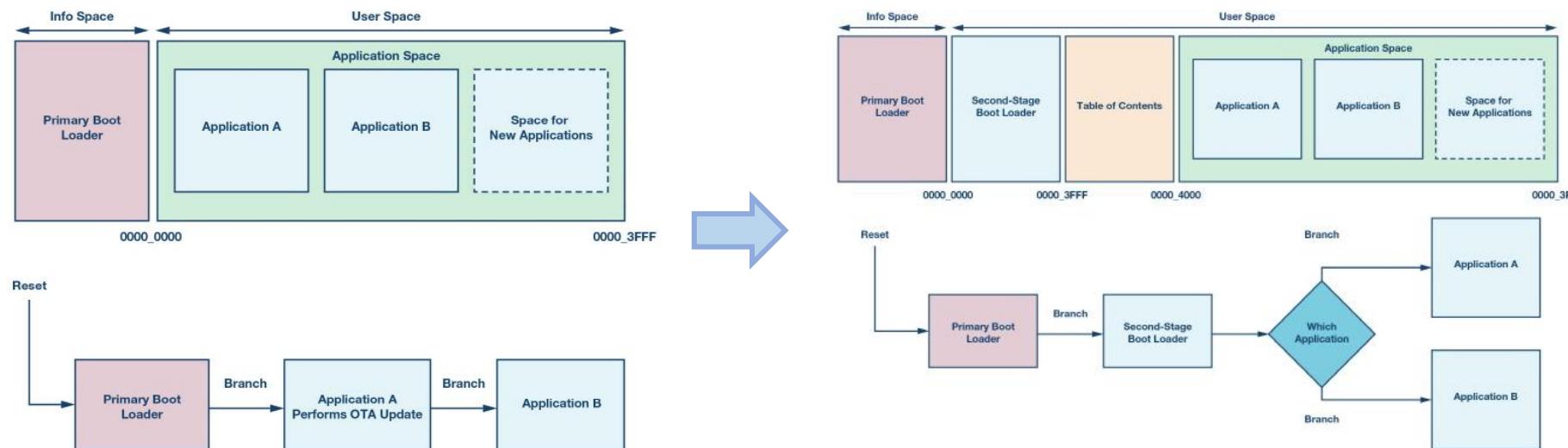
- 1단계 부트로더 (Primary bootloader)
 - MCU의 ROM(Read-Only Memory)에 영구 상주하는 소프트웨어 application
 - 1단계 부트로더가 상주하는 메모리 영역은 정보 공간으로서, 종종 사용자의 접근을 차단함
 - 이 application은 리셋 할 때마다 실행되는데, 필수적인 하드웨어 초기화를 수행하고 사용자 소프트웨어를 메모리에 로드
 - 만약 MCU가 플래시 메모리 같이 비휘발성 메모리를 내장하고 있다면 부트로더가 로딩 할 필요 없이 플래시 메모리에 있는 프로그램으로 제어를 넘길 수 있음
- 2단계 부트로더 (SSBL, Second Stage Boot Loader)
 - 만약 1단계 부트로더가 OTA 업데이트를 지원하지 않으면 SSBL(Second Stage Boot Loader)이 필요
 - 1단계 부트로더와 마찬가지로 SSBL 역시 리셋 할 때마다 실행 → 다만 OTA 업데이트 프로세스 부분만 실행

SSBL을 사용할 때의 메모리 맵과 Boot Flow



■ SSBL이 반드시 필요한 이유 (1/2)

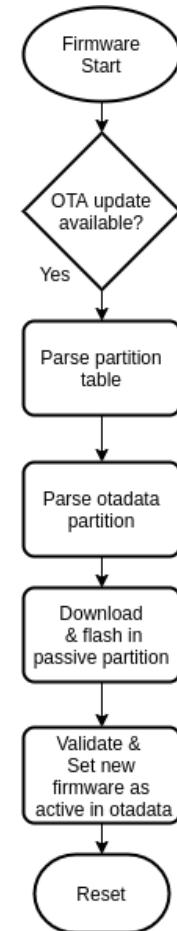
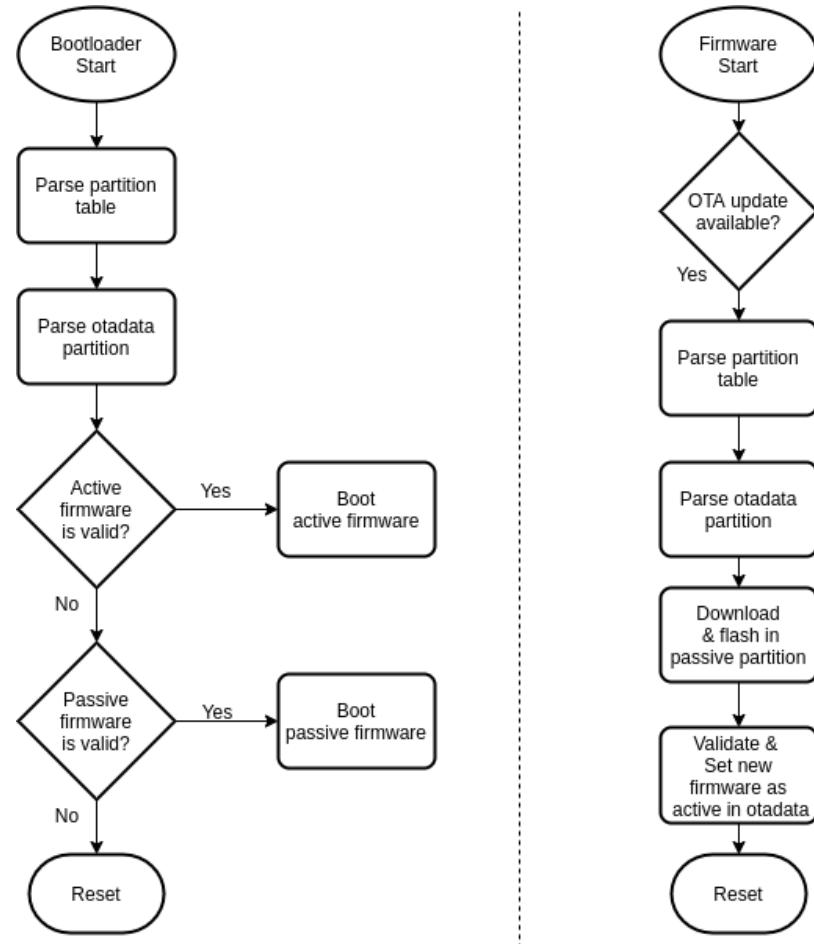
- SSBL을 생략하고 모든 OTA 업데이트 기능을 사용자 application에 바로 적용하는 경우에...
 - MCU에 기본으로 설치되어 있는 application A는 OTA 업데이트 관련 소프트웨어를 포함하며, 이 소프트웨어를 사용해 서버의 요청이 있을 때 application B를 다운로드할 수 있음
 - 다운로드가 완료되고 application B를 검사한 후에는 application A가 application B의 리셋 핸들러로 분기 명령을 실행해서 application B로 제어를 넘김
 - 리셋 핸들러는 소프트웨어 application의 진입 지점이 되는 작은 코드 조각으로서, 리셋 할 때 실행됨
 - 이 경우, 함수 호출을 의미하는 분기를 실행해서 리셋이 일어난 것처럼 흉내 냈 → 이 방법은 다음 두 가지 측면에서 문제!!



■ SSBL의 역할

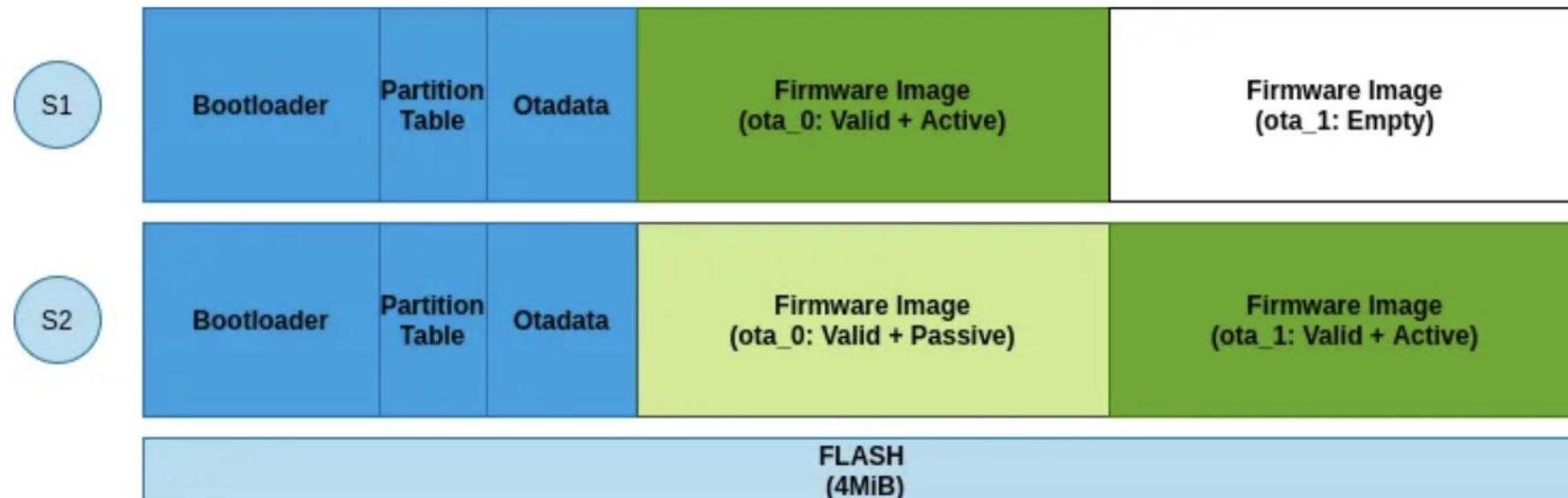
- 현재 application이 어디서 시작되는지 판단하고 그 주소로 분기하는 역할
 - MCU 메모리 내에서 다양한 application의 위치는 ToC(Table of Contents)에 저장
 - 이것은 SSBL과 소프트웨어 application이 소통하기 위해서 사용하는 영구 메모리의 공유 영역임
 - 즉, OTA 업데이트 프로세스가 완료되면 ToC가 새로운 application 정보로 업데이트됨
 - 전체적인 OTA 업데이트 프로세스를 SSBL로 집어 넣을 수도 있음
 - 이 경우에는 application이 간단히 ToC로 업데이트를 요청하고 리셋을 실행하기 위한 플래그를 설정함
 - 그러면 SSBL이 다운로드 시퀀스와 검증 프로세스를 수행함
 - 이 방식은 코드 중복성을 최소화하고 소프트웨어 application을 단순화할 수 있음 → 하지만 SSBL 자체를 업데이트해야 하는 새로운 문제를 야기함 (업데이트 코드를 업데이트해야 함)
- 결국, SSBL에 어떤 기능들을 넣을지 결정하는 것은 클라이언트 디바이스의 메모리 제약, 다운로드 된 application의 유사성, OTA 업데이트 소프트웨어의 이식성에 따라서 결정!

■ Boot sequence w/o rollback



- bootloader와 application은 모두 파티션 테이블을 사용하여 플래시의 다양한 파티션과 해당 오프셋에 대한 정보를 추출
- 파티션은 otadata 내부에 저장된 시퀀스 번호를 기반으로 활성 펌웨어(최신 업데이트)를 선택하는 역할을 담당
- otadata 파티션에는 2개의 플래시 섹터(액티브/패시브 펌웨어와 유사, 8KiB 크기)가 할당되므로 OTA 업데이트 중에 시퀀스 번호 및 기타 매개변수를 업데이트하는 동안 전원 안전이 가능함

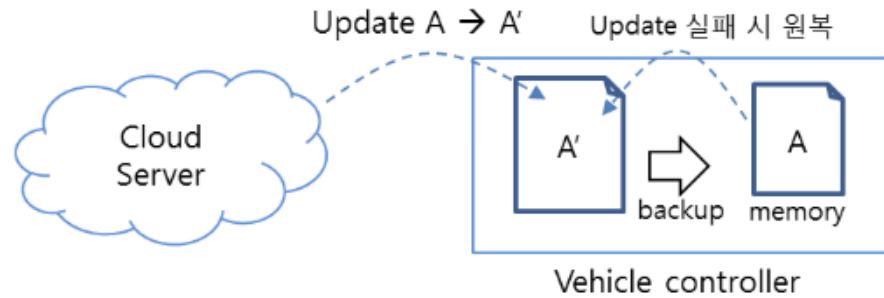
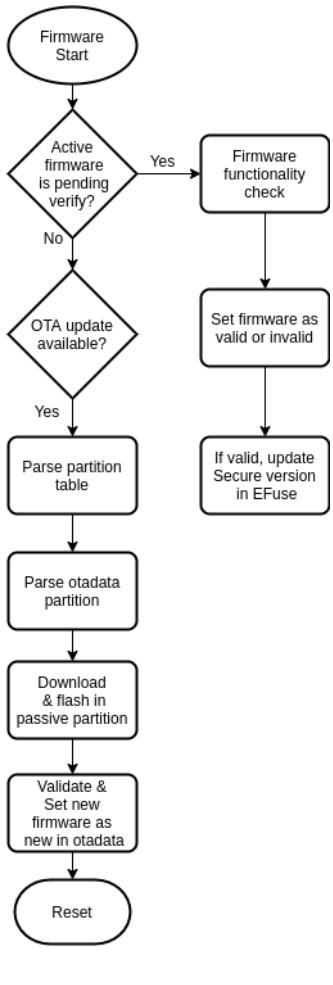
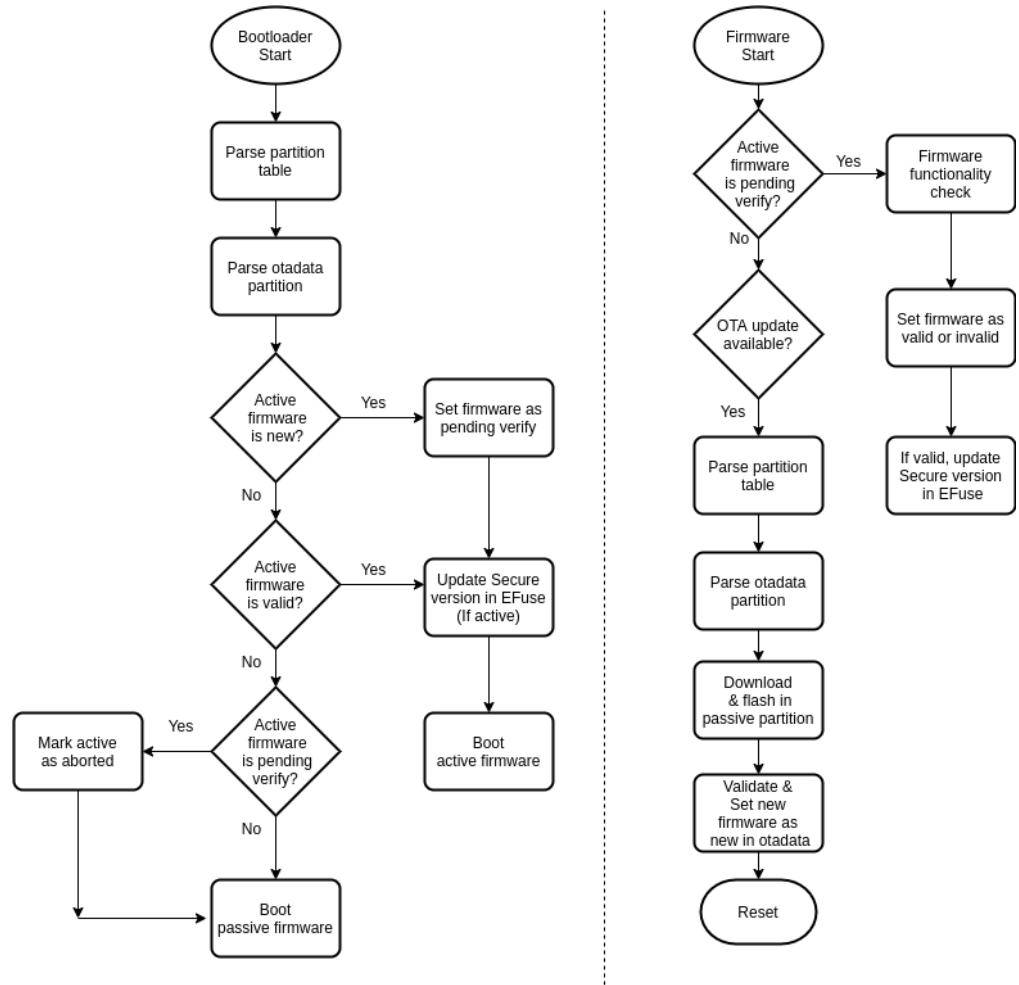
OTA 프로세스 개요 - Rollback (cont'd)



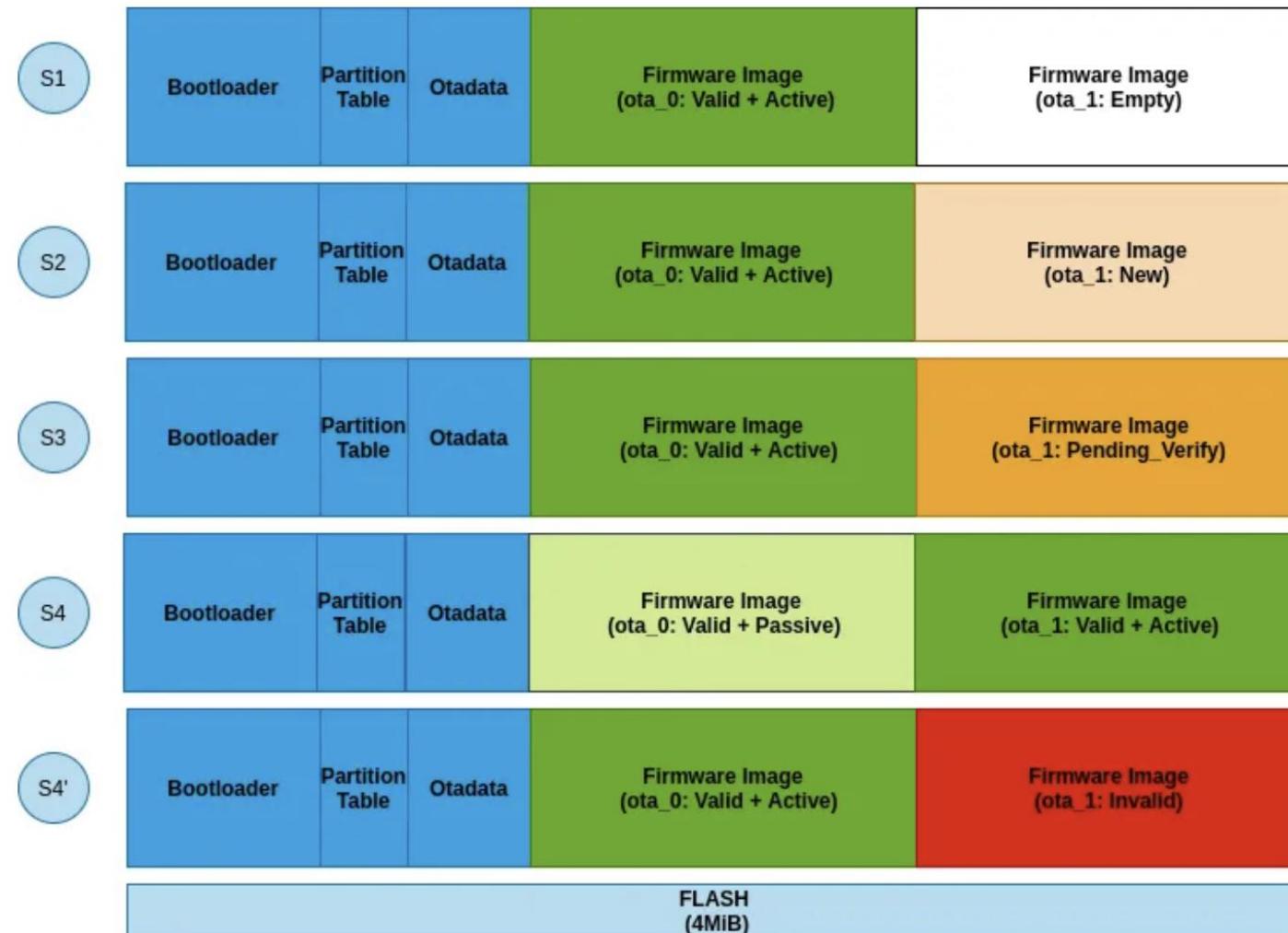
일반 OTA 업데이트의 플래시 레이아웃 전환

OTA 프로세스 개요 - Rollback (cont'd)

■ Boot sequence with rollback



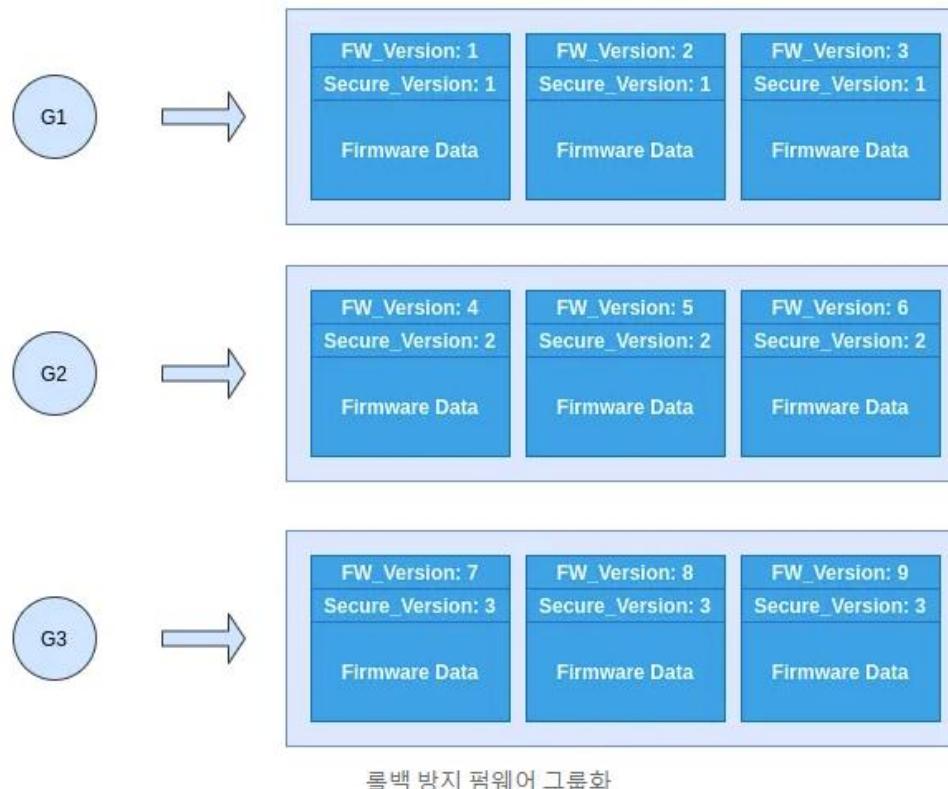
OTA 프로세스 개요 - Rollback (cont'd)



롤백 OTA 업데이트를 통한 플래시 레이아웃 전환

■ Anti-Rollback

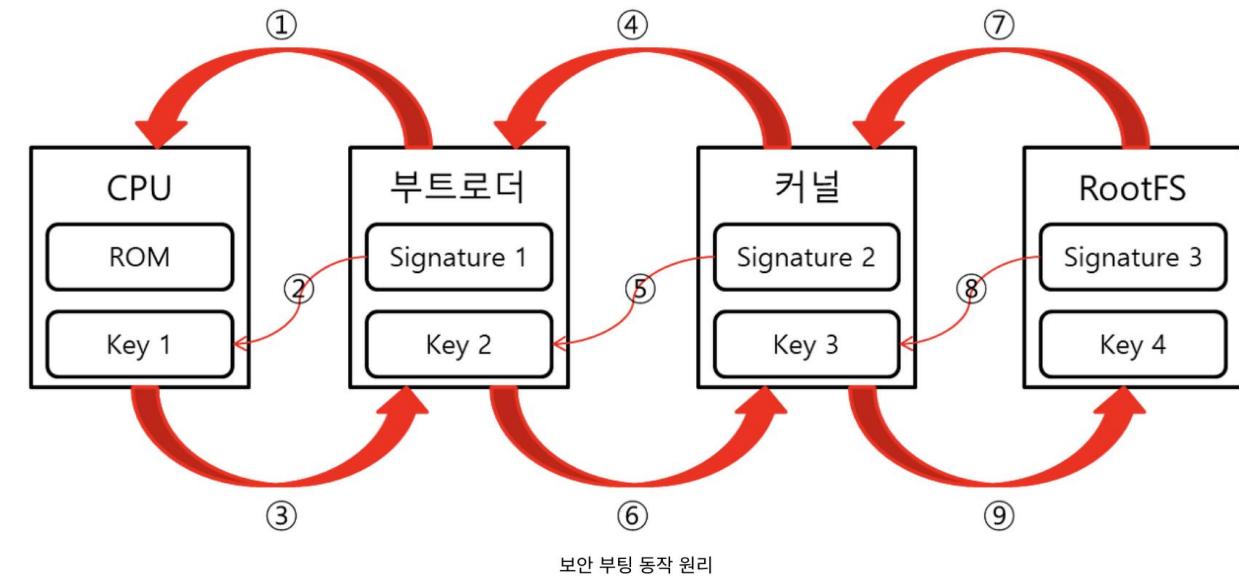
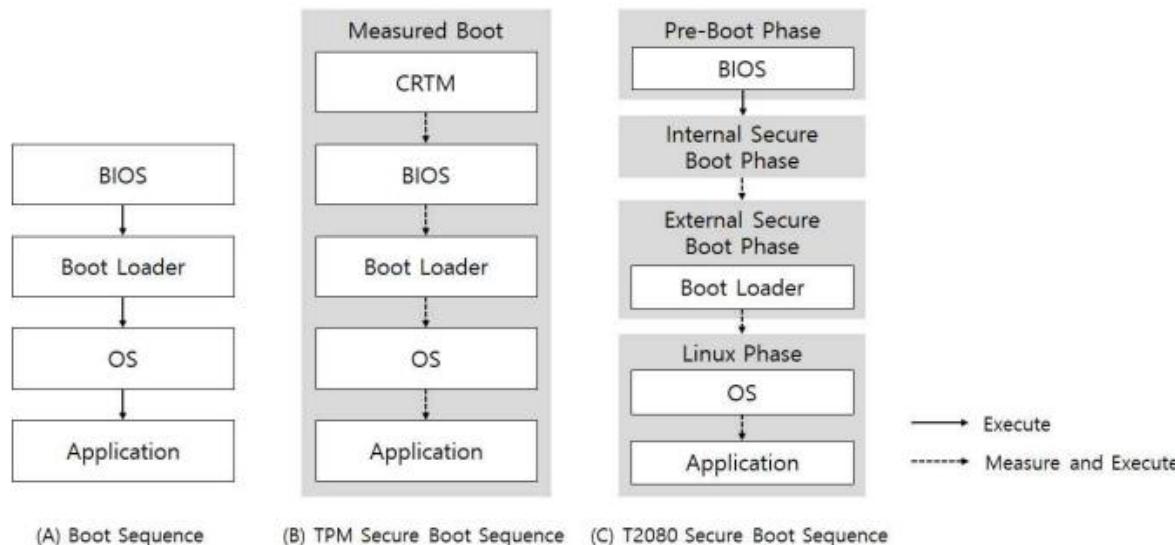
- 롤백 방지는 내장된 보안 버전이 장치의 EFuse(일회성 프로그래밍 가능 메모리)에 프로그래밍된 버전보다 높은 경우에만 펌웨어를 실행할 수 있는 기능



- 각 그룹(G1/G2/G3)에는 펌웨어 버전은 다르지만 보안 버전은 동일한 펌웨어 이미지 3개가 있음
- 장치 롤백 방지 예약 EFuse에 2비트가 설정된 경우(보안 버전 2를 의미) G2 및 G3 그룹의 펌웨어만 장치에서 실행할 수 있고, 보안 버전이 낮은 그룹 G1의 펌웨어 이미지는 부팅이 허용되지 않음
- EFuse의 버전은 새 펌웨어 이미지의 기능이 확인된 후에만 업데이트되므로 롤백 방지는 롤백과 긴밀하게 결합됨

■ Booting Process

- 전원이 들어오면 ROM(BIOS)에서 사전에 정해진 위치에서 Boot Loader를 읽고 Boot Loader는 초기화를 진행하고, OS를 로딩한다.
- 만약, Booting 과정에서 공격자가 원하는 프로그램을 실행한다면??



05

OTA 요소 기술의 이해

OTA 요소 기술의 이해

요소 기술	상세	비고
FOTA	<ul style="list-style-type: none">OTA 서버와 통신을 위한 프로토콜 (서버는 자동차 회사에서 운영)새로운 업데이트 존재 여부 확인패키지 다운로드 및 업데이트 상태 전달Gateway ECU 혹은 AVN에서 DM Client 동작OMA DM 1.x, OMA DM 2.0, LWM2M, MQTT 등 표준	<ul style="list-style-type: none">복잡한 차량 환경에 일관된 업데이트 정책 적용 가능Linux, Android, QNX, Classic Autosar 등
Differential Update	<ul style="list-style-type: none">차량 내부의 다양한 제어기에 업데이트 정보를 전달하고, 상태를 수집하여, DM에 전달 (Gateway ECU에서 동작)Adaptive Autosar에서는 UCM Master	<ul style="list-style-type: none">Binary 파일의 변경부분 업데이트업데이트 최소화를 위한 차분 솔루션 적용 필수 (무선 데이터 최적화)
A/B 업데이트	<ul style="list-style-type: none">OTA 정보를 제어기에 전달하고, 업데이트 결과를 전달 받기 위한 통신CAN 기반 – 표준 UDS 프로토콜을 사용Ethernet 기반<ul style="list-style-type: none">- SOME/IP- Adaptive Autosar ARA COM (SOME/IP or DDS)- gRPC	<ul style="list-style-type: none">Active & Inactive 두개의 저장 공간 확보
UDS	<ul style="list-style-type: none">Full Update와 Differential Update 두 종류Differential Update는 다시 A/B와 In-place Update로 구분업데이트 완료 후 Hash 알고리즘 (MD5 or SHA256)으로 검증File 혹은 Streaming 기반 업데이트	<ul style="list-style-type: none">Classic Autosar 업데이트 방식MCU 업데이트
Security	<ul style="list-style-type: none">현대 자동차 사용 – Secure Flash ½The Update Framework(TUF) & Uptane	<ul style="list-style-type: none">업데이트 위변조 방지 (해킹 등의 위협으로부터 운전자 보호)

■ A/B Differential Update

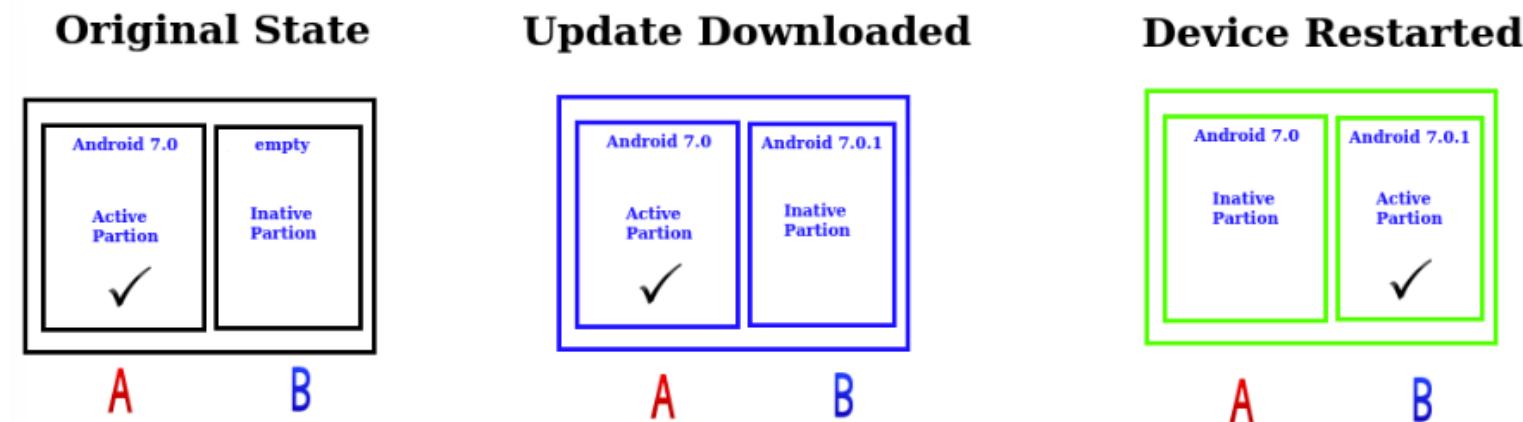
- 제어기 저장 공간을 충분하게 확보 가능한 경우 사용
- 저장 공간을 이중화(A/B)하여, 하나의 저장공간만을 동작 시 사용하고, 다른 하나의 저장 공간은 업데이트를 위해 사용 → In-place Update 대비 안전한 업데이트 가능
- 현대자동차의 모든 차분 솔루션은 A/B 업데이트에 기반함



장점	단점
<ul style="list-style-type: none"> Runtime 중 업데이트 가능 업데이트 오류 발생 시 복구 시나리오가 단순함 활성화된 파티션에 일반적인 오류 발생 시, 비활성화된 파티션을 응급으로 활용할 수 있음 가장 안전한 업데이트 방식 	<ul style="list-style-type: none"> 비활성화된 파티션을 유지해야 하므로, 제품 원가 상승

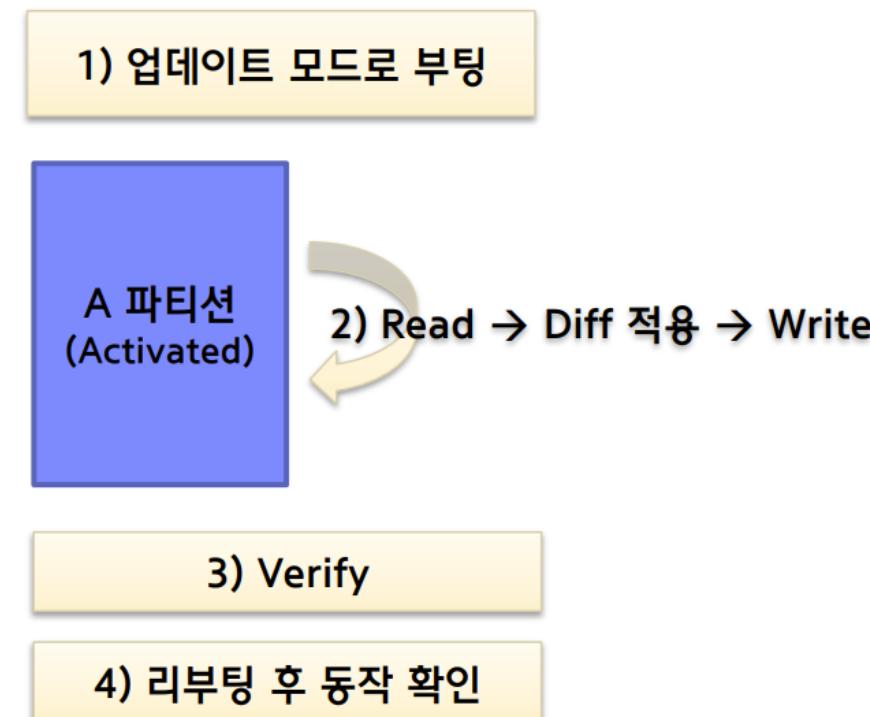
■ A/B Update (a.k.a. Seamless Update)

- OTA 업데이트를 수행할 때 부팅 가능한 시스템을 보장하고 가동 중단 시간 최소화
- 개별 파티션으로 나뉘는 방식은 데이터 부분이 실제 시스템 파일이 있는 장소가 아닌 별도의 장소에 저장된다는 것을 의미함 → 이렇게 하면 시스템 보안에 도움이 되며(데이터 파티션은 자체 파일 및 폴더 권한 세트를 가질 수 있음) 원활한 업데이트를 위해 매우 편리함
- 이 모든 작업이 완료되면 재부팅하여 새 시스템 파티션을 사용할 수 있고, 파티션에 번호가 매겨져 있기 때문에 파티션을 이동하거나 이름을 바꿀 필요 없이 이 작업을 수행할 수 있음



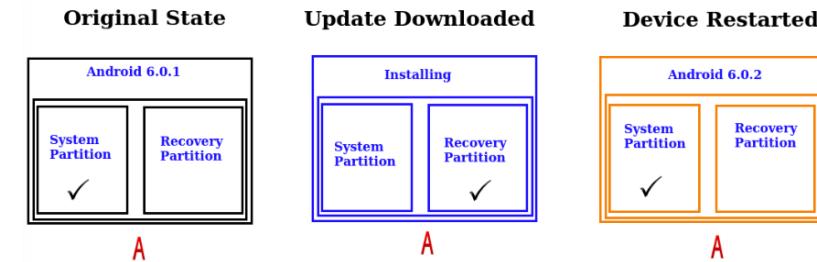
■ In-place Differential Update (a.k.a. Non-A/B update)

- 여분의 저장공간 확보가 불가능한 제어기 차분 업데이트를 위해 사용
- 별도의 업데이트만을 위한 시스템 부팅 환경을 구축해야 함
- 오류 발생 시 복구 시나리오가 복잡함



■ Procedure of in-place update

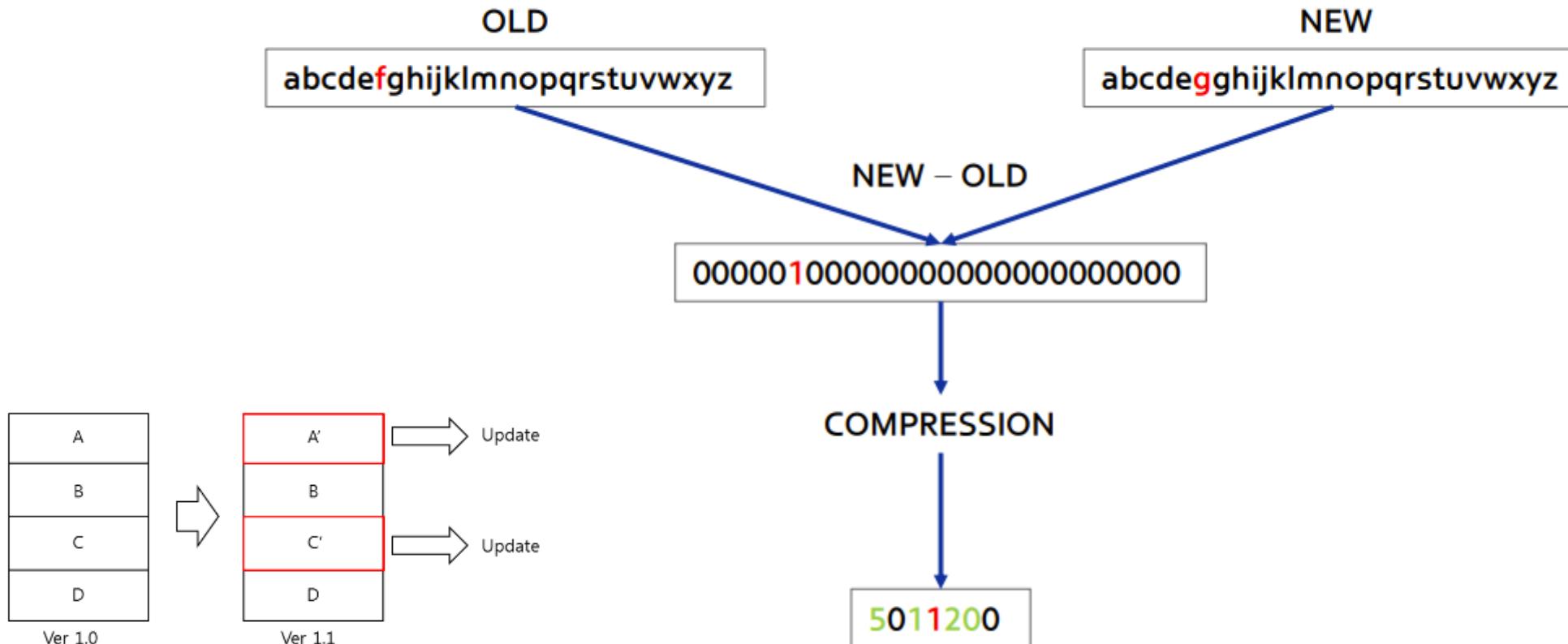
- 장치가 OTA 서버에서 검사를 수행하고 업데이트 가능성에 대한 알림을 받음
- Cache 혹은 Data partition에 OTA 패키지를 다운로드
- 인증서에 대해 암호화 서명을 확인 → 서명 확인 후 정상이면 업데이트 설치 메시지 표시
- 장치가 Recovery mode로 재부팅
- Recovery mode에서 공개키를 이용해 패키지의 암호화 서명을 확인 → 서명 확인 후 정상이면 필요에 따라 패키지 를 추출하여 부팅, 시스템 등을 업데이트
- 이때, 시스템 파티션에 New Recovery Partition의 컨텐츠가 포함될 수 있음 (optional)
- 장치가 정상적으로 재부팅
 - 새로 업데이트된 부팅 파티션이 로드 → 새로 업데이트된 시스템 파티션에서 바이너리를 마운트하고 실행 시작
 - 정상적으로 시작될 때 시스템은 복구 파티션의 내용을 원하는 내용과 비교하여 확인 → 서로 다르므로, 복구 파티션은 시스템 파티션에서 원하는 내용으로 re-flashing (이후 부팅 시, 복구 파티션에서는 이미 새 컨텐츠가 포함되어 있으므로 re-flashing 필요 없음



■ Differential Update 제약사항

- 차분 솔루션은 old & new 데이터의 유사성에 기반하기에 이에 따른 제약이 존재함

요소 기술	상세	비고
파일 종류에 따른 제약사항	<ul style="list-style-type: none"> 압축파일 유사한 데이터라도 압축되면, 데이터 유사성이 훼손되어 차분 데이터 적용 어려움 별도의 전용 알고리즘 개발이 필요함 <ul style="list-style-type: none"> - 멀티미디어 파일 (동영상, 음악) - 전용 압축 알고리즘 사용되어 차분 데이터 추출 불가능 암호화 파일 <ul style="list-style-type: none"> - 데이터 유사성이 암호화 이후 소실되어 차분 데이터 추출 불가능 	
원본 데이터 보존 필요	<ul style="list-style-type: none"> 차분 생성 알고리즘: $New - Old = Diff$ 차분 적용 알고리즘: $OLD + Diff = New$ Old가 변조되는 경우 정상적인 New 데이터 생성이 불가능 제어기에서 차분 적용 대상 데이터는 절대 변조되면 안됨 업데이트 시작 전 Old에 대해 Hash 검사 진행 업데이트 완료 후 New에 대해 Hash 검사 진행 	



소프트웨어 업데이트 시 전체를 업데이트하는 것이 아닌 수정된 부분만 업데이트하여 시간을 단축할 수 있다.

■ Hash 알고리즘 사용

- Old 데이터 변조 확인
 - 차분 솔루션 적용하기 이전, Old 데이터 변조 확인을 위해 Hash 알고리즘으로 검토 (SHA1 사용 가능)
- 업데이트 성공 여부 확인
 - 차분 업데이트 완료 후, 업데이트 성공 여부를 확인하기 위해 Hash 알고리즘 적용하여 확인 (MD5 사용 가능)
- OTA 패키지 변조 여부 확인
 - 배포된 OTA 패키지 변조 여부 확인을 위해 사용 (SHA256 이상 필수)

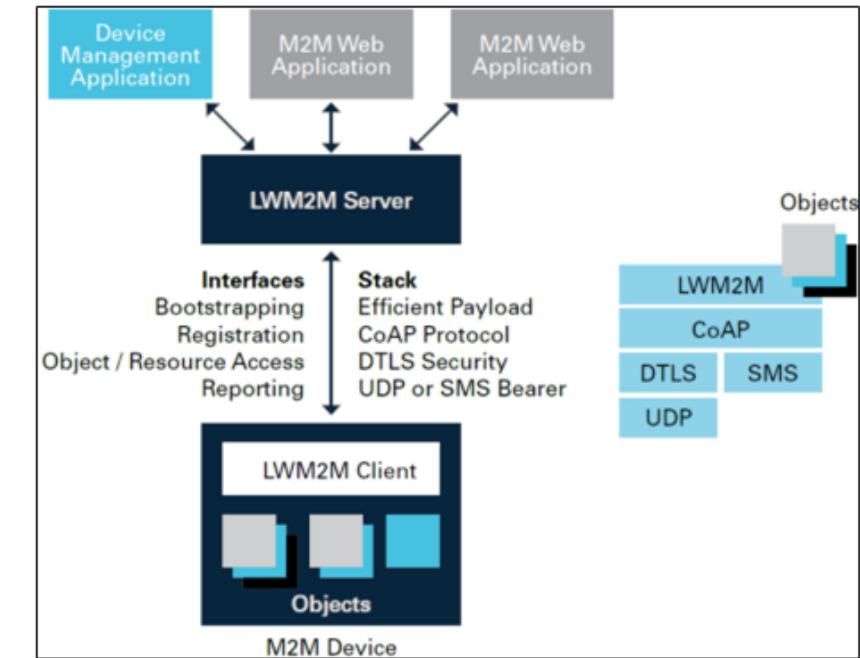
알고리즘 종류	성능	보안
MD5	매우 빠름	나쁨
SHA1	빠름	나쁨
SHA256/512	느림	좋음

OTA 요소 기술의 이해 - OTA Protocol

Protocol	특징	비고
OMA DM 1.x	<ul style="list-style-type: none">2007년 1.2 releaseSync ML 기반 (XML)다양한 Transport Layer 사용 (HTTP, OBEX 등)현대자동차 OTA 기반 프로토콜소수의 오픈소스 존재하지만, 현재 관리되고 있지 않음	
OMA DM 2.x	<ul style="list-style-type: none">2016년 2.0 releaseJson 기반RESTful Interface보다 많은 Device Management Command 추가됨강화된 보안	
LwM2M	<ul style="list-style-type: none">2017년에 1.0 release → 2020년 11월에 1.2 releaseIoT 장치 관리자를 위한 통신 프로토콜로 리프로그래밍 관련 사양 포함되어 있음다양한 오픈소스 구현체가 존재함	
MQTT	<ul style="list-style-type: none">2019년 MQTT 5.0 releaseMQTT는 가볍고 효율적이고, 확장성과 신뢰성 그리고 보안성을 지원하는 특징으로 인해 IoT에서 사실상의 데이터 전송 표준으로 인식됨	

■ Lightweight Machine to Machine (LwM2M)

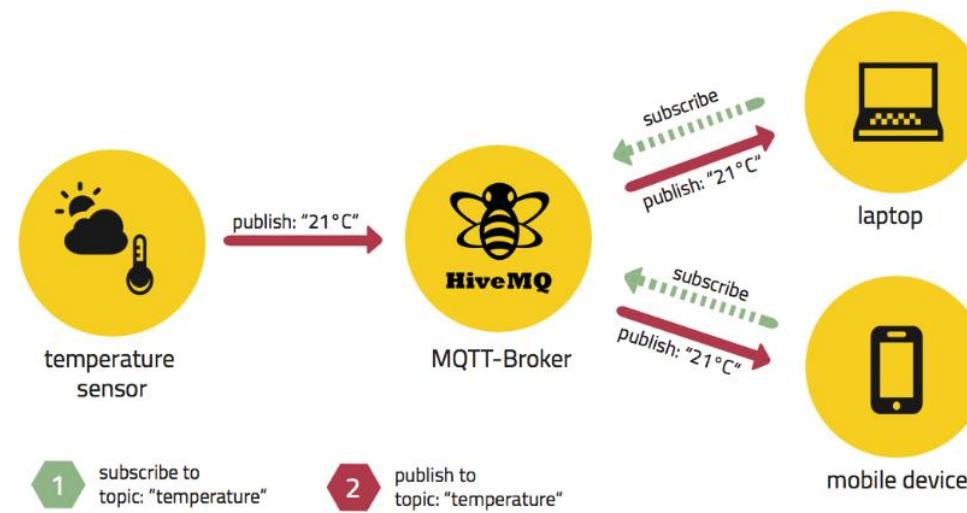
구분	설명
*CoAP 사용	<ul style="list-style-type: none"> IoT 전송 프로토콜 CoAP 사용 (CoAP 기반으로 하여 메시지가 작고, 빠르고 다양한 IoT 기기 지원 가능함)
DTLS 기반 보안	<ul style="list-style-type: none"> DTLS 기반 보안 기술 적용 PSK (Pre-Shared Key) / Public Key를 활용하여 Provisioning 및 Secure Booting 지원
oneM2M 표준	<ul style="list-style-type: none"> 다수 장치를 효율적으로 관리 oneM2M 표준 기술
기타	<ul style="list-style-type: none"> 응용개발 용이 재사용성 증가 저사양부터 고사양 디바이스 지원



*CoAP: M2M 노드들 사이 통신을 지원하기 위한 REST 기반 프로토콜

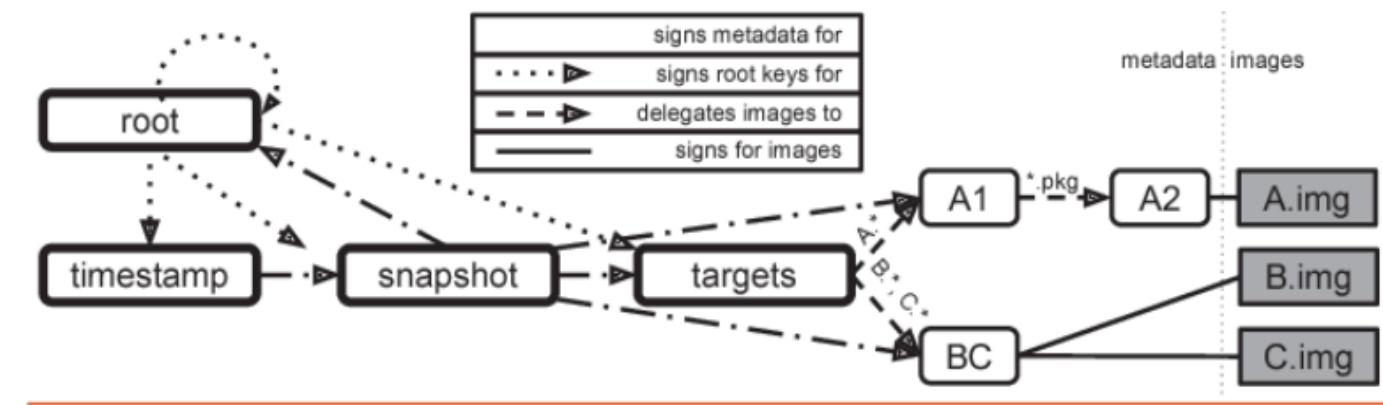
■ Message Queueing Telemetry Transport (MQTT)

- Publish – Subscribe 기반의 메시지 송수신 프로토콜
 - 작은 코드 공간이 필요하거나 네트워크 대역폭이 제한되는 원격 통신을 위해, 즉 IoT와 같은 제한된, 혹은 대규모 트래픽 전송을 위해 만들어진 프로토콜
 - TCP/IP 프로토콜 위에서 동작하지만 동시에 굉장히 가벼우며, 많은 통신 제약들을 해결
 - ✓ MQTT는 Bluetooth나 Zigbee처럼 별도의 모듈로 별도의 대역폭을 갖는 통신 규약이 아닌, WiFi나 기타 방법을 통해 인터넷을 통해 TCP/IP 기반의 메시지 송수신을 한다는 것을 의미
 - ✓ 이러한 장점들 때문에 Facebook Messenger가 MQTT를 채택했고, 우아한형제들(배달의 민족 서비스 기업)에서도 중계 시스템 개선을 위해 MQTT를 도입하려 시도한 적이 있음



■ TUF 개요

- A framework for securing software update systems
- 다양한 OTA 해킹에 대한 대응 시나리오와 Private Key 유출 등의 사고 대응 시나리오를 포함하는 사양
- 차량용으로 확장된 사양은 Uptane (<https://uptane.github.io/>)



< Separation of duties between roles on a compromise-resilient repository >

■ TUF 설계 원칙

1) Trust

- 정상적으로 신뢰 된 파일만 다운로드 받아야 함
- 파일에 대한 신뢰성은 영원하면 안되고, 갱신되지 않으면 소멸되어야 함 → 파일에 대한 신뢰성은 오직 root만이 부여할 수 있음

2) Mitigating Key Risk (Compromise-Resilience)

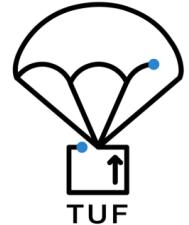
- 암호화에 사용되는 키에 대한 보안
 - ✓ 빠르고 안전하게 키 교체 및 폐기와 같은 보안 기능을 제공해서 키 노출을 최소화하고 키가 노출되더라도 안전성이 유지되는 기능을 제공

3) Integrity

- Server에 저장되어 있는 파일에 대한 무결성과 Server의 저장소 자체에 대한 무결성을 보장

4) Freshness

- 소프트웨어가 Update되는 경우
 - ✓ 대부분 Bug 혹은 취약점을 보완하는 Update가 많이 이루어지기 때문에 소프트웨어는 항상 최신 상태를 유지해야 함
- Client가 Update 요청 시
 - ✓ 공격자가 최신 version 소프트웨어를 취약점이 있는 소프트웨어로 바꿔 치기 해서 client에 설치하게 한 후 취약점을 이용해 공격할 수 있기 때문에 client가 update하기 위해 최신 version 소프트웨어를 받아오는 과정이 중요함
 - ✓ 따라서 TUF에서는 Update 시 현재 version 보다 이전 version 파일은 차단하는 등 update process가 정상적인지 확인해주는 역할을 수행



Thank You



- Lab: <https://mose.kookmin.ac.kr>
- Email: sh.jeon@kookmin.ac.kr