Over the Air (OTA) Updates: Requirements for a Full System Solution

John H. Floros

Field Application Engineer

October 2018 | AMF-AUT-T3180

NXP | CONNECTS

Agenda

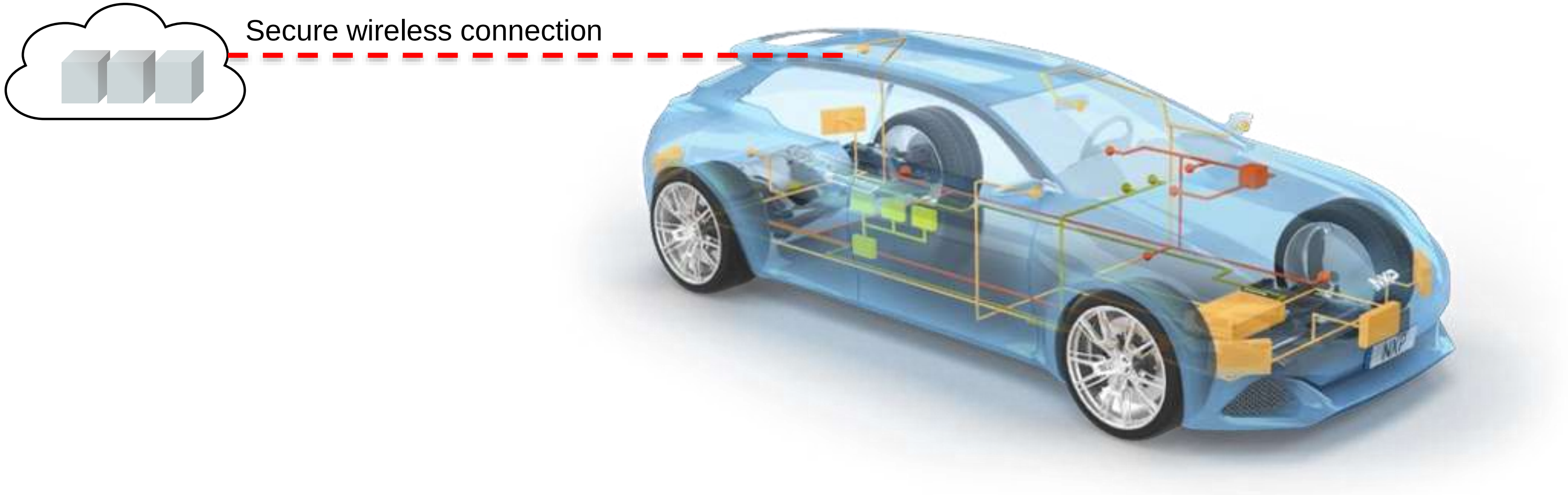
- Introduction
- OTA Architecture
- Update Methods
- S32K1xx Features and Use Cases
- S32K Next Gen Features and Use Cases
- Summary
- For More Information…

Objective

- Overview of OTA and its challenges
- Understand how NXP handle over the air updated in their portfolio
- Understand how to handle over the air updates in edge nodes MCUs such as S32K devices

What are Over-the-Air Updates?

Vehicle firmware updates received wirelessly (Wi-Fi, cellular…) from the cloud instead of through wired connection in a repair garage (sometimes referred to as FOTA).



Car Manufacturer (OEM) Demands

- Minimal impact on driver (no down time)
- No risk of leaving vehicle unusable
- Security to prevent rogue updates or theft

OEM Benefits

- Save money – no recall required
- Ability to patch critical bugs/security vulnerabilities
- Revenue generation with new features

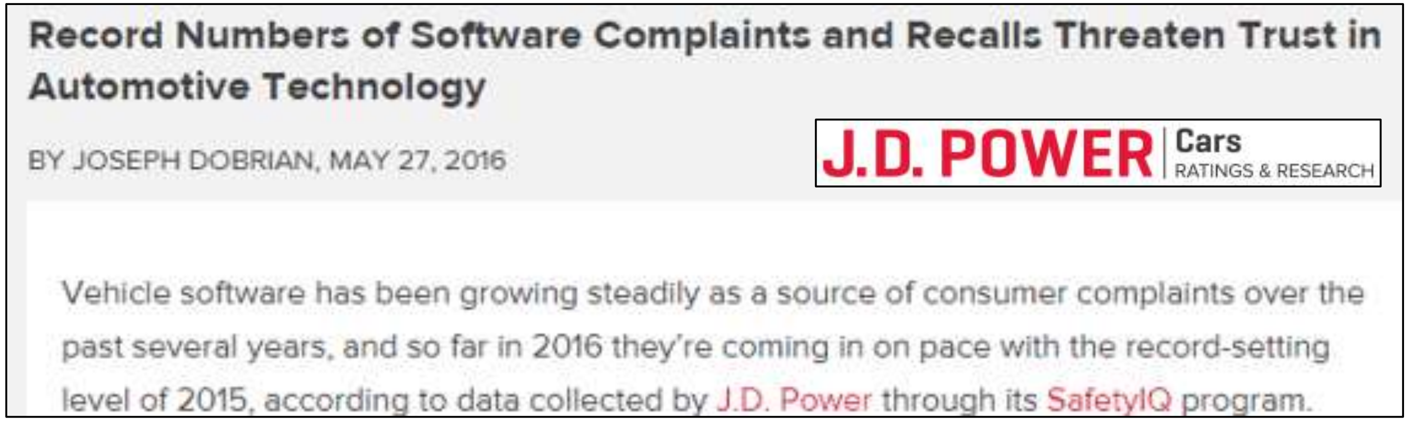
Key Drivers for OTA Updates

- Premium vehicles have over 100M lines of code! (Windows 10 has 50M)
- 15% of vehicle recalls and 60% of warranty costs are firmware related

- Firmware updates require vehicle to be returned to the garage
  - Time-consuming and costly
- No guarantee customer will return it for recall

- Difficult to deliver new features to vehicle owners
- OEMs are missing post-purchase, revenue-generation opportunities

Automotive Software Recalls Are Growing!

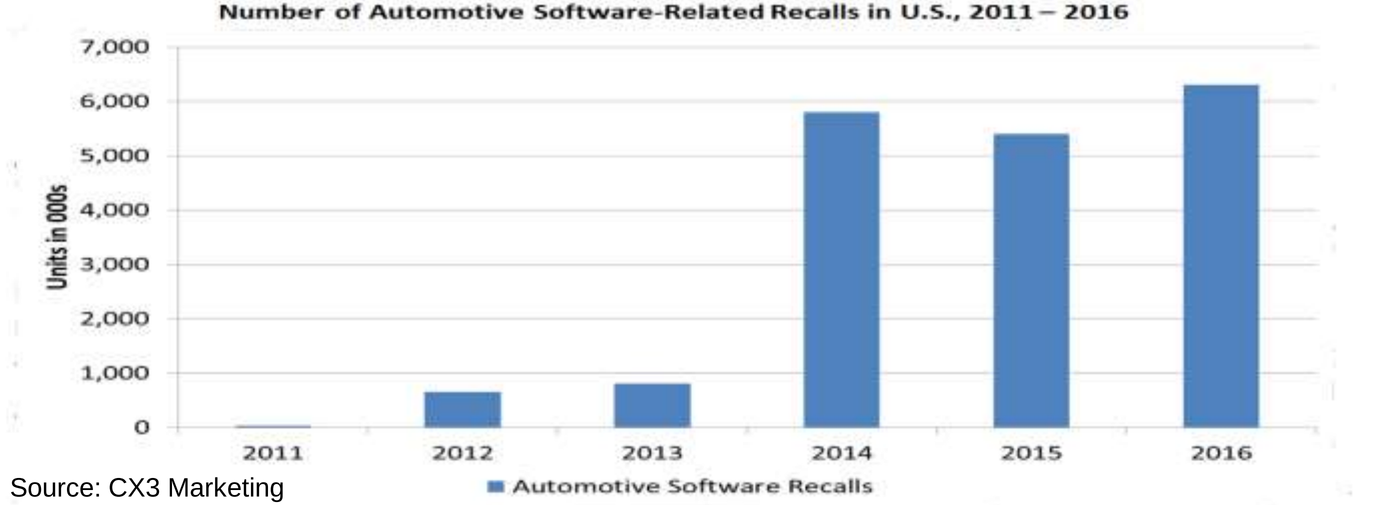


Record Numbers of Software Complaints and Recalls Threaten Trust in Automotive Technology

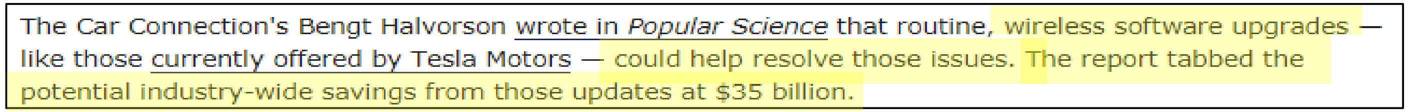
BY JOSEPH DOBRIAN, MAY 27, 2016

J.D. POWER | Cars RATINGS & RESEARCH

Vehicle software has been growing steadily as a source of consumer complaints over the past several years, and so far in 2016 they're coming in on pace with the record-setting level of 2015, according to data collected by J.D. Power through its SafetyIQ program.



Source: CX3 Marketing



The Car Connection's Bengt Halvorson wrote in *Popular Science* that routine, wireless software upgrades — like those currently offered by Tesla Motors — could help resolve those issues. The report tabbed the potential industry-wide savings from those updates at $35 billion.

Source: C3X Marketing

Software Flaw in Airbag Triggers GM's Recall

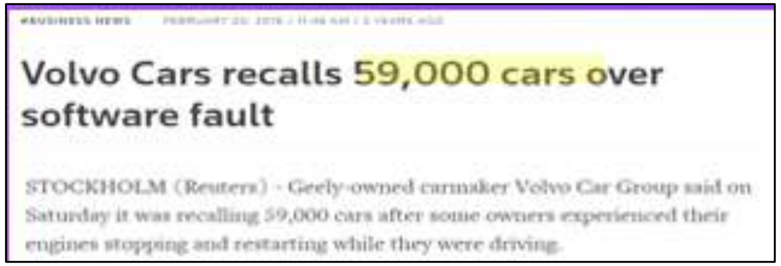
Exposes a larger problem: Managing growing software in cars

Junko Yoshida
9/9/2016 08:30 PM EDT

LOGIN TO RATE

GM Recalls 4 Million Vehicles to Fix Air Bag Software Defect

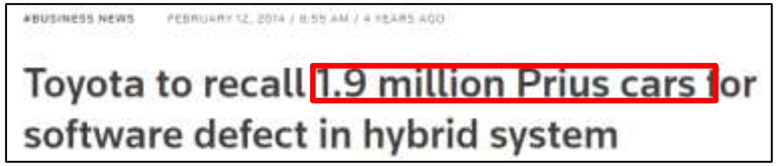
POSTED BY REGION NEWS TEAM ON SEPTEMBER 13, 2016 | 8:39 AM IN LOCAL NEWS, REGION NEWS



#BUSINESS NEWS FEBRUARY 20, 2016 / 11:46 AM / 2 YEARS AGO

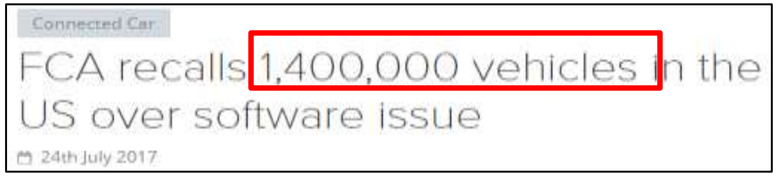
Volvo Cars recalls 59,000 cars over software fault

STOCKHOLM (Reuters) - Geely-owned carmaker Volvo Car Group said on Saturday it was recalling 59,000 cars after some owners experienced their engines stopping and restarting while they were driving.



#BUSINESS NEWS FEBRUARY 12, 2014 / 8:55 AM / 4 YEARS AGO

Toyota to recall 1.9 million Prius cars for software defect in hybrid system



Connected Car

FCA recalls 1,400,000 vehicles in the US over software issue

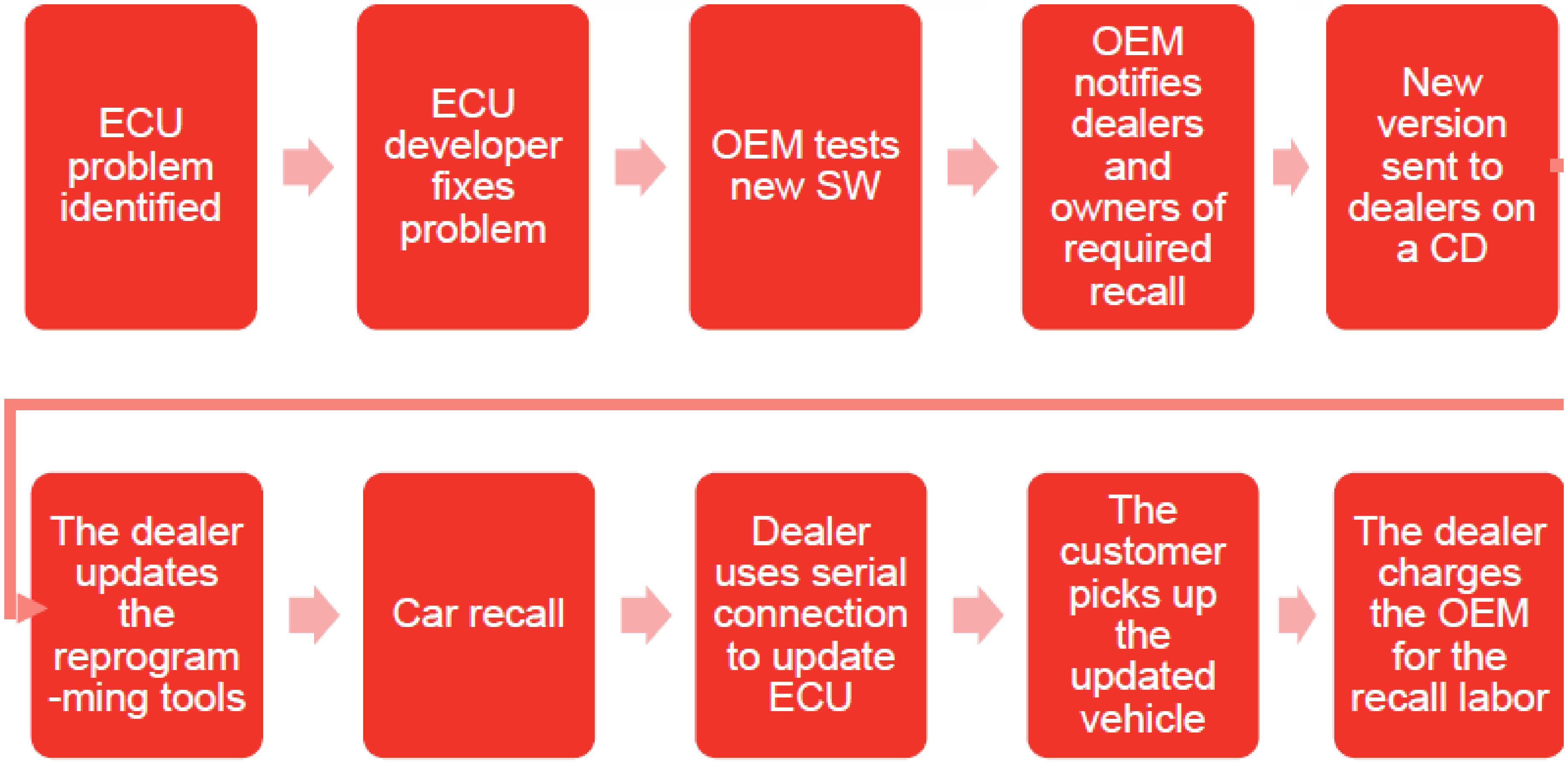
24th July 2017

08 SEP

Audi recalls vehicles with defective ECU software

Recall Masters Audi Recalls, Volkswagen Recalls

Common Recall Process



ECU problem identified

ECU developer fixes problem

OEM tests new SW

OEM notifies dealers and owners of required recall

New version sent to dealers on a CD

The dealer updates the reprogram-ming tools

Car recall

Dealer uses serial connection to update ECU

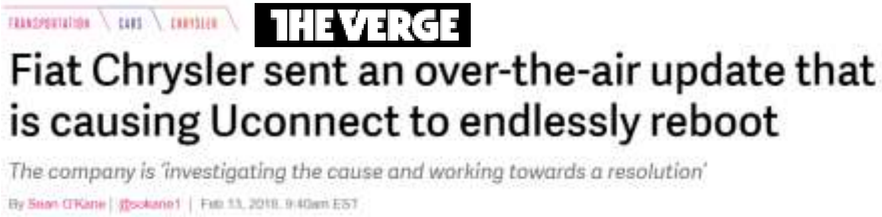
The customer picks up the updated vehicle

The dealer charges the OEM for the recall labor

Vector and Redbend (2014).*Update ECUs using Delta- and Over-the-Air-Technology* [PDF Slides]. Retrieved April 22, 2016 from https://vector.com/portal/medien/cmc/events/Webinars/2014/Vector_RedBend_Webinar_Flashing_over_the_air_and_delta_technology_20140121_EN.pdf

OTA Update Concerns

- Cybersecurity / hijack attacks
- Problem causing inoperable vehicle



TRANSPORTATION \ CARS \ CHRYSLER

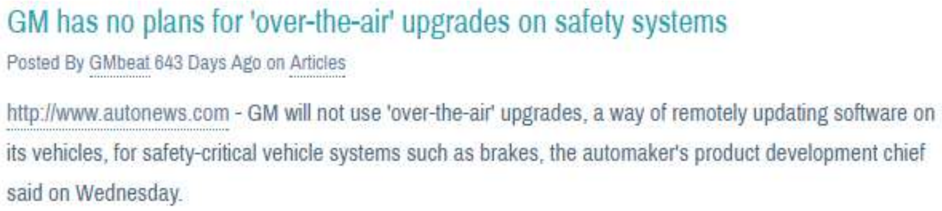
THE VERGE

Fiat Chrysler sent an over-the-air update that is causing Uconnect to endlessly reboot

The company is 'investigating the cause and working towards a resolution'

By Sean O'Kane | @sokane1 | Feb 13, 2018, 9:40am EST

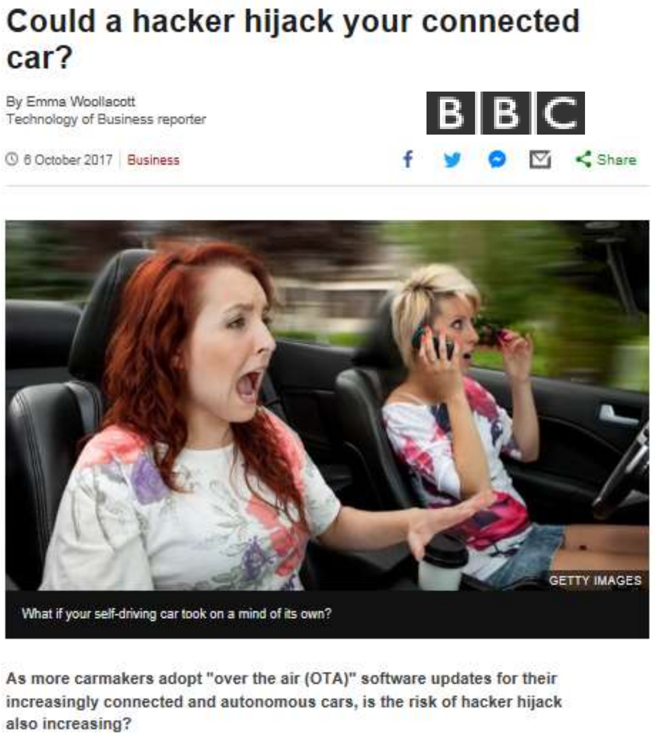
- Applicability across vehicle



GM has no plans for 'over-the-air' upgrades on safety systems

Posted By GMbeat 643 Days Ago on Articles

http://www.autonews.com - GM will not use 'over-the-air' upgrades, a way of remotely updating software on its vehicles, for safety-critical vehicle systems such as brakes, the automaker's product development chief said on Wednesday.



Could a hacker hijack your connected car?

By Emma Woollacott
Technology of Business reporter

BBC

6 October 2017 | Business

What if your self-driving car took on a mind of its own?

As more carmakers adopt "over the air (OTA)" software updates for their increasingly connected and autonomous cars, is the risk of hacker hijack also increasing?

Implementation

- Is implementation vulnerable to cyber attacks?
- Can firmware be rolled back if problem?
- Can OTA update be done w/o down time?

Consumer acceptance

- Ask customer for every update?
- Will customer continually postpone updates?
- Will customer refuse certain updates?

It is an Attractive Target for Hackers!

Valuable Data

- Collection of data/info
- Storage of data
- Diagnostic functions

Protect Privacy

High Vulnerability

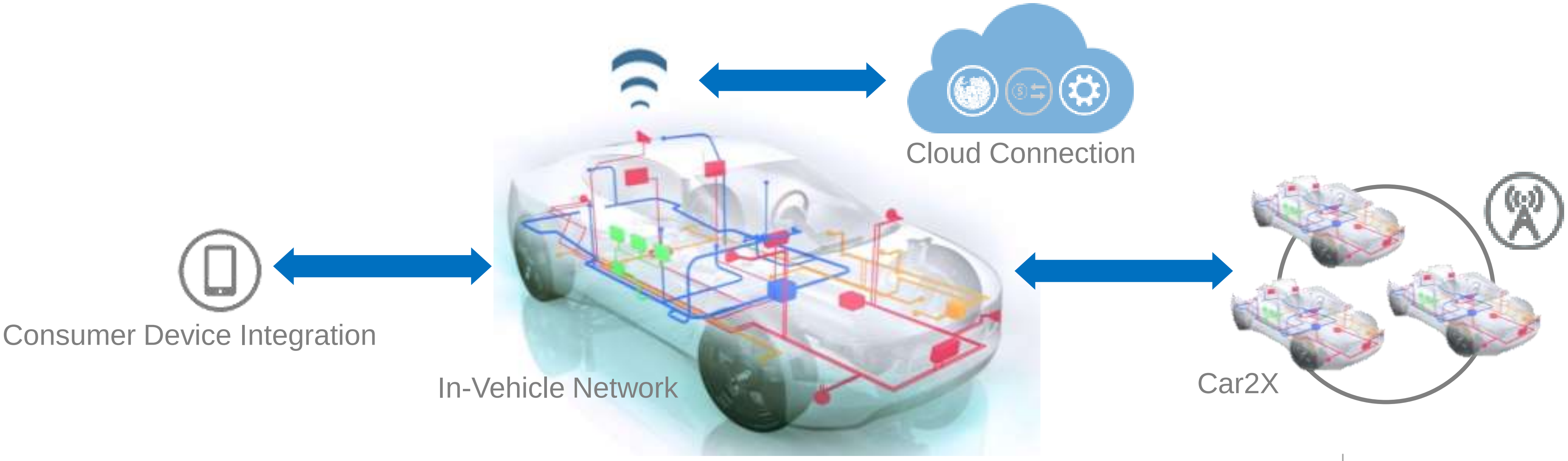
- Increasing number of nodes
- More advanced features
- X-by-Wire

Increase Safety

Easy (Remote) Access

- Fully Connected Car
- External & internal interfaces
- Wired & wireless interfaces

Prevent Unauthorized Access



Cloud Connection

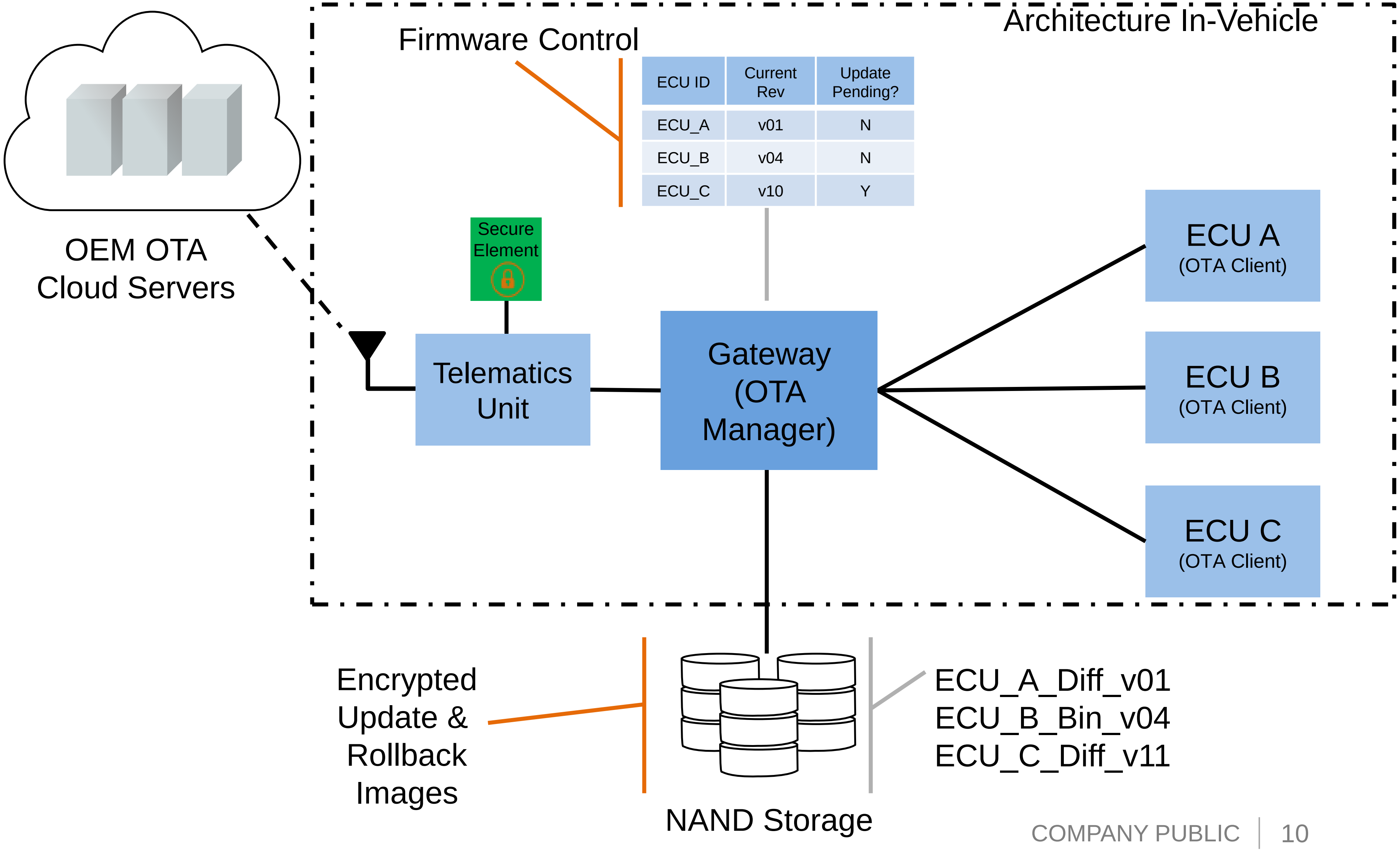
Consumer Device Integration

In-Vehicle Network

Car2X

OTA Architecture Update Flow

Full-Vehicle OTA Update Flow



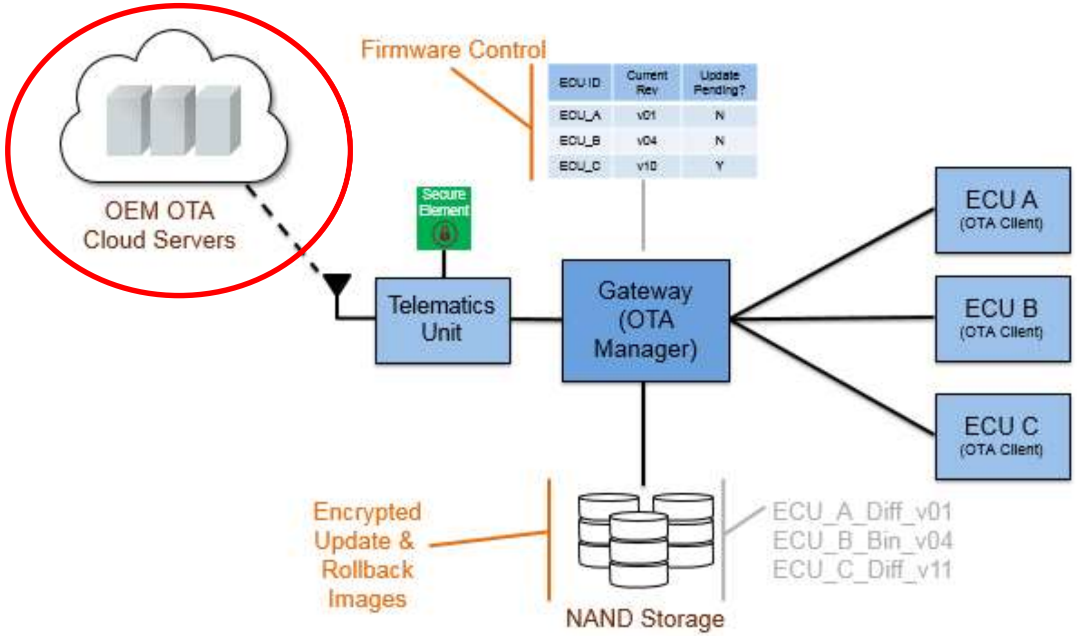
ECU ID	Current Rev	Update Pending?
ECU_A	v01	N
ECU_B	v04	N
ECU_C	v10	Y

OEM Cloud Server

- Vehicle database
  - Contains details of all vehicles by VIN / serial number
  - Lists software currently on each vehicle
  - Manages dependency between firmware versions on multiple nodes

- Software database
  - Contains all software, firmware, maps etc.
  - Generates diff files is required

- Real-time monitoring and reporting
  - Receives usage information and error codes from active vehicles
  - Able to poll vehicles to update local database



ECU ID	Current Rev	Update Pending?
ECU_A	v01	N
ECU_B	v04	N
ECU_C	v10	Y

Backend Connection (TCU)

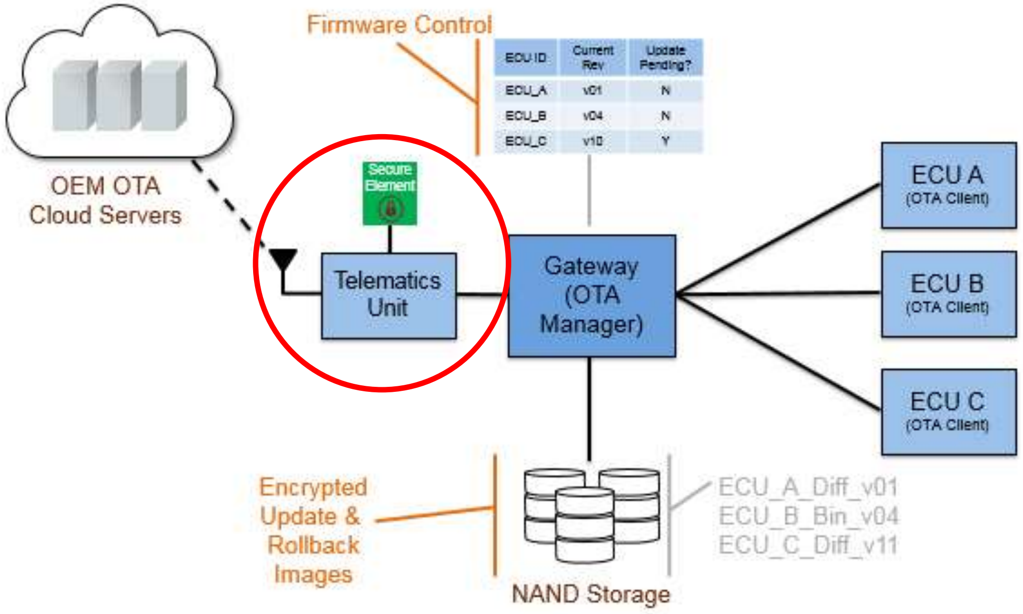
IP Connectivity

- Establishes physical communication link and IP connections
- Handles multiple comms protocols (Wi-Fi, cellular, V2X, etc.)

Logical Connectivity to Cloud Servers (Services)

→ depends on vehicle architecture. Logical connections can also be managed by gateway.

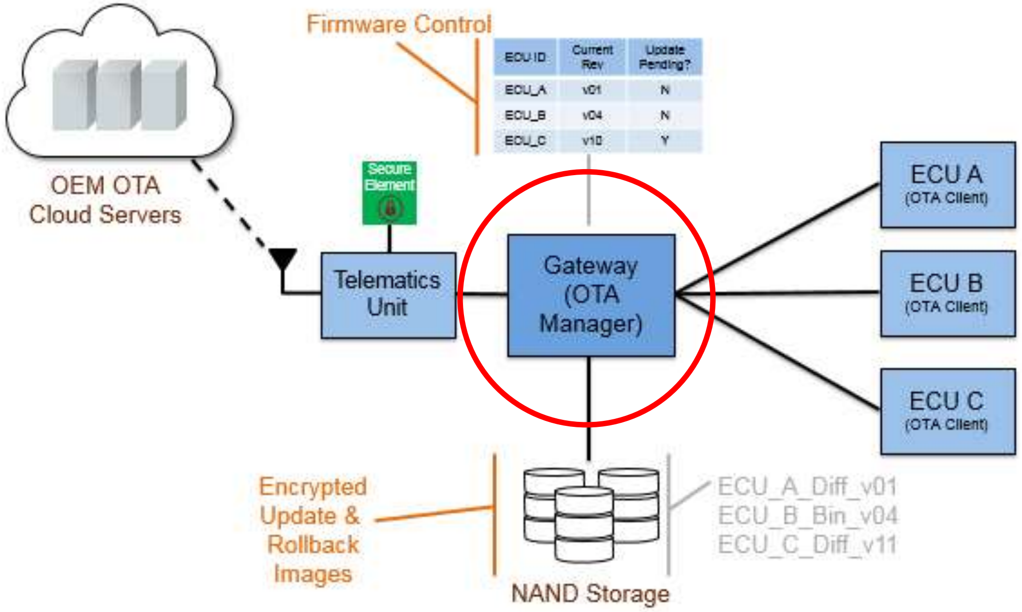
- Authenticates the vehicle and server
  - Uses Secure Element as root of trust
- Establish secure connection between vehicle and OEMs server
- Handle loss of connection issues
- Hand off package to the OTA Manager



ECU ID	Current Rev	Update Pending?
ECU_A	v01	N
ECU_B	v04	N
ECU_C	v10	Y

Gateway (OTA Manager)

- Contains database of all ECUs
- Perform hashing and authentication on received image
- Unpack the received file and split for individual ECUs
- Stores updates until ready to install
- Can be used to create diffs
- Synchronize updates across multiple nodes
- Establish secure channel with ECU (end node)
- Prompt IVI to display update details to driver (if required)
- Commence UDS diagnostic session with end node
- Success message is reported back to the cloud



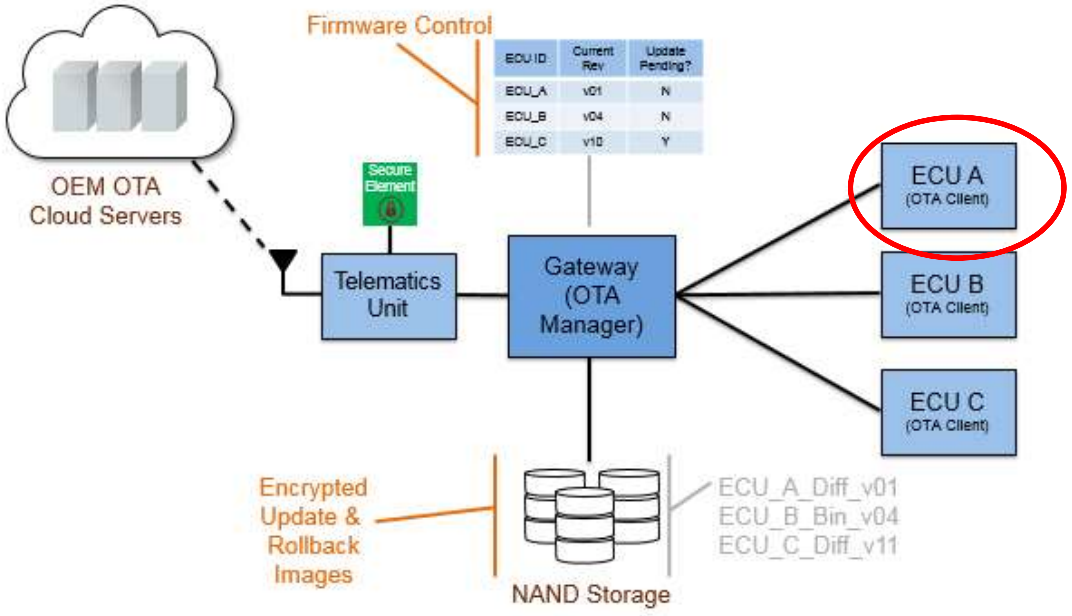
ECU ID	Current Rev	Update Pending?
ECU_A	v01	N
ECU_B	v04	N
ECU_C	v10	Y

End Node (OTA Client)

- Verify/Authenticate image
- Decrypt Image (optional)
- Use version number to confirm firmware is new
- Perform diff calculation (diff updates only)
- Erase flash block
- Program update
- Switch to new version and notify OTA Manager

Software:

- Runs on the end node to be updated
- Performs actual the flashing operation
- Typically works along with bootloader
- Can be very small (<2K)



ECU ID	Current Rev	Update Pending?
ECU_A	v01	N
ECU_B	v04	N
ECU_C	v10	Y

OTA Update Methods
End Node (OTA Client)

OTA Assumptions

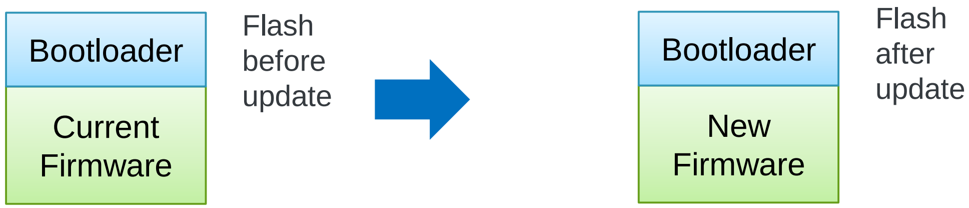
- End node:
  - gets partial or full image for flashing
  - will have at least enough spare erased flash for a full image
  - receives updated software over serial link
  - has boot block which never changes with OTA updates
- Best case: update is performed while running existing software
- Before new software becomes active, application/boot software can perform:
  - Security validation
  - Functional validation
- New software starts on reset following the update completion

Over the Air (OTA) Update Methods 1/2

In general, there are 2 methods for performing updates to an end node

In Place Update:

Update is performed on top of existing version



Advantages:

- No need for additional flash

Cost:

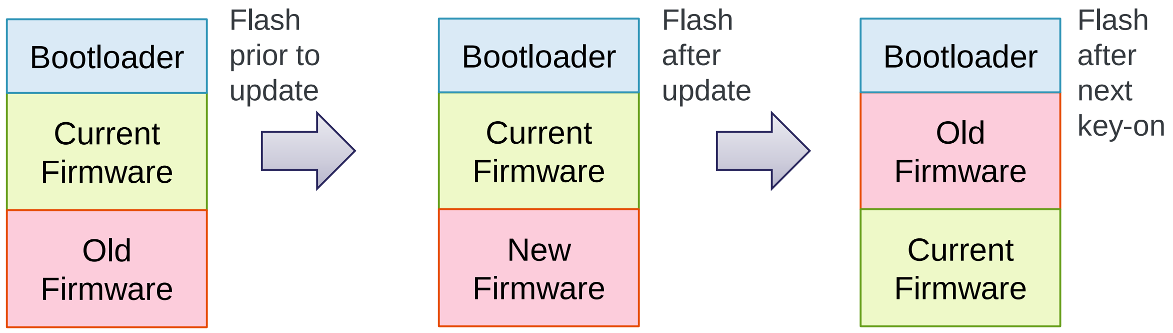
- Requires vehicle downtime during update process
- Not possible to instantly "roll-back" if an issue occurs
- Higher risk to have an ECU inoperable

Note: Bootloader is typically not updated via OTA.

Over the Air (OTA) Update Methods 2/2

A/B Swap Update:

2 versions of firmware exist in internal flash at the same time.



Advantages:

- Update can be carried out whilst application is actively running from flash
- Always have original firmware to roll back to in case of issue
- Vehicle always available – guaranteed no vehicle downtime regardless of update errors

Cost:

- Requires 2x flash application storage
- Higher max current (run current in block A + erase/program current in block B)

Note: Bootloader is typically not updated via OTA

MCU Features to Facilitate OTA Updates - 1/2

RWW Flash

Allows read accesses from a flash block whilst another block is being erased or programmed.
Allows application to continue executing during update process.

Lockable Flash Regions

Provides additional protection to critical code such as the bootloader during the update process to prevent accidental deletion.

Multi Core

Can allow one core to perform the update whilst the other is dedicated to running the existing application, minimising performance impact.

Brownout Detection and Recovery

A reset during an update can leave the flash in an undefined state. The MCU should be able to detect an occurrence of this, and aid the bootloader in performing a recovery upon exit from reset.

MCU Features to Facilitate OTA Updates - 2/2

Flash Remapping / MMU

Allows instant switching between new and old firmware images stored in different physical locations in flash.

Lifecycle Management

Securely log the current firmware version and prevent illegal attempts to rollback to a previous version or to install unauthorised software or hardware.

Cryptographic Security

Enables storage of private keys and fast decryption and authentication of incoming update. Also authenticates firmware each boot to prevent tampering

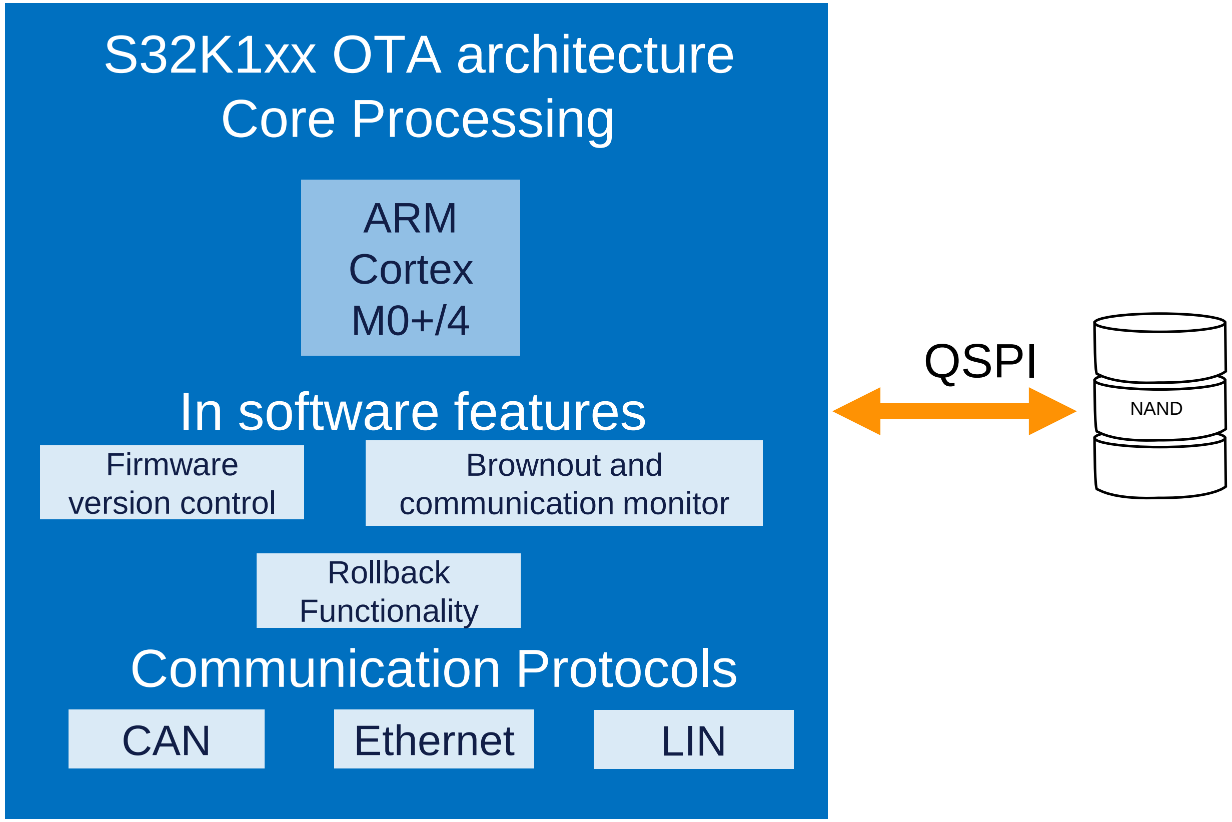
Small code flash block sizes

Smaller flash blocks will improve update speed and efficiency. An in-place differential update will only modify a small number of locations, which need to be backed up, erased and reprogrammed.

S32K1xx Capabilities

S32K1xx: OTA Client Features

- OTA Client
- Software Features (Bootloader)
  - Version Control
  - Brownout and communication monitor.
- Roll back functionality
- External memory support.
  - QSPI
- Communication protocols



S32K1xx OTA Scenario: A-B Swap

Pros/Limitations

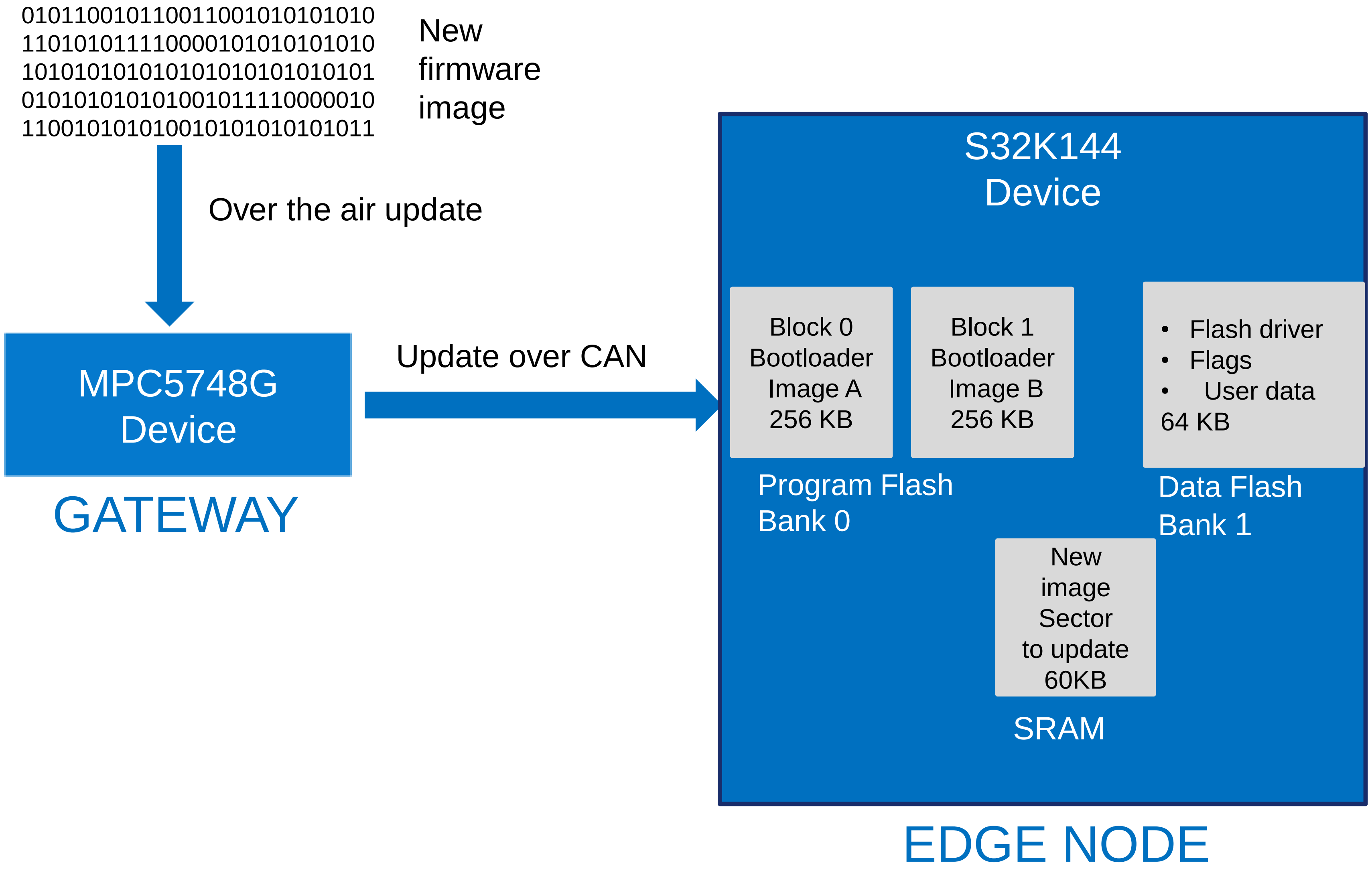
- Pro: A-B swap allows backup immediately available

- Limitations: compared to large MCUs with multiple code partitions, updating the image cannot be done live.

S32K148 A/B Swap Use Cases

- Use of external QuadSPI
- 512kB update while running application
- Only a specific section to be updated

S32K144 A/B Swap



010110010110011001010101010
1101010111110000101010101010
101010101010101010101010101
010101010101001011110000010
110010101010010101010101011

New
firmware
image

Over the air update

MPC5748G
Device

GATEWAY

Update over CAN

S32K144
Device

Block 0
Bootloader
Image A
256 KB

Block 1
Bootloader
Image B
256 KB

Program Flash
Bank 0

- Flash driver
- Flags
- User data

64 KB

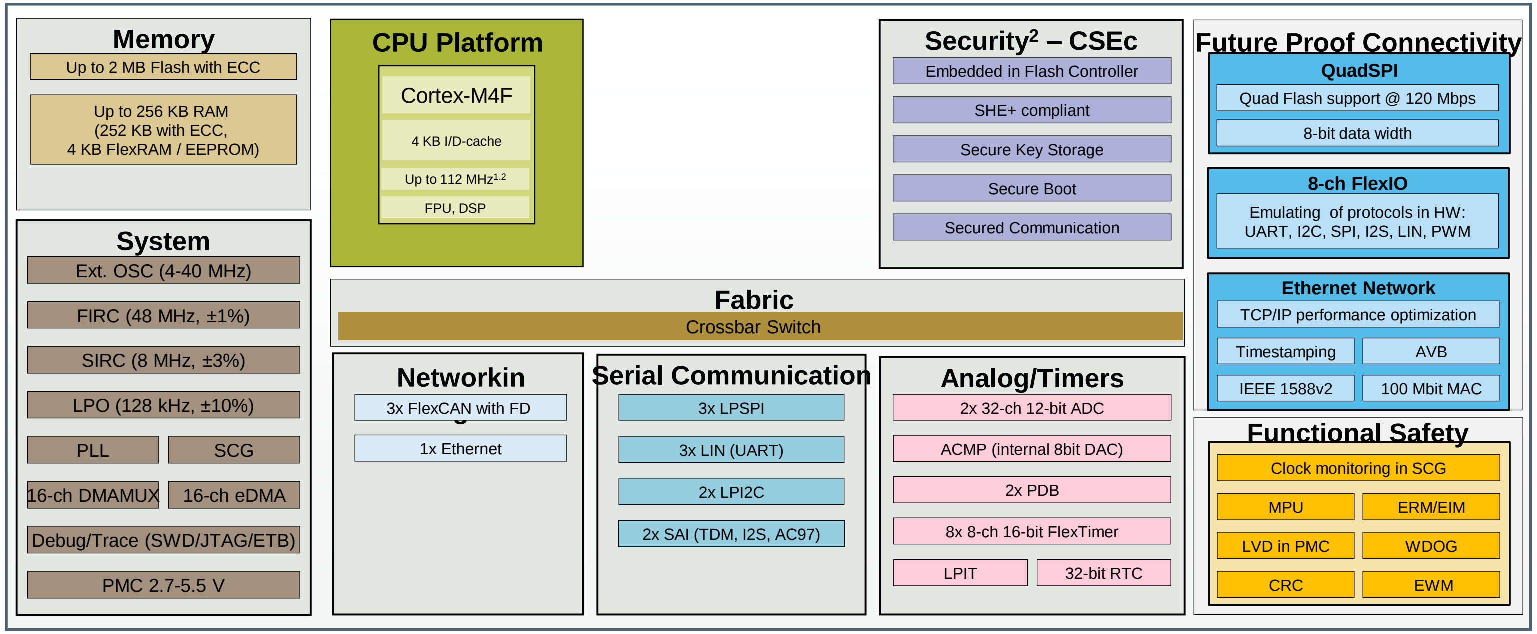
Data Flash
Bank 1

New
image
Sector
to update
60KB

SRAM

EDGE NODE

S32K148: ASIL B 2M General Purpose MCU With HW Security



Memory

Up to 2 MB Flash with ECC

Up to 256 KB RAM (252 KB with ECC, 4 KB FlexRAM / EEPROM)

System

Ext. OSC (4-40 MHz)

FIRC (48 MHz, ±1%)

SIRC (8 MHz, ±3%)

LPO (128 kHz, ±10%)

PLL | SCG

16-ch DMAMUX | 16-ch eDMA

Debug/Trace (SWD/JTAG/ETB)

PMC 2.7-5.5 V

CPU Platform

Cortex-M4F

4 KB I/D-cache

Up to 112 MHz^{1,2}

FPU, DSP

Security^2 – CSEc

Embedded in Flash Controller

SHE+ compliant

Secure Key Storage

Secure Boot

Secured Communication

Fabric

Crossbar Switch

Networkin

3x FlexCAN with FD

1x Ethernet

Serial Communication

3x LPSPI

3x LIN (UART)

2x LPI2C

2x SAI (TDM, I2S, AC97)

Analog/Timers

2x 32-ch 12-bit ADC

ACMP (internal 8bit DAC)

2x PDB

8x 8-ch 16-bit FlexTimer

LPIT | 32-bit RTC

Future Proof Connectivity

QuadSPI

Quad Flash support @ 120 Mbps

8-bit data width

8-ch FlexIO

Emulating of protocols in HW: UART, I2C, SPI, I2S, LIN, PWM

Ethernet Network

TCP/IP performance optimization

Timestamping | AVB

IEEE 1588v2 | 100 Mbit MAC

Functional Safety

Clock monitoring in SCG

MPU | ERM/EIM

LVD in PMC | WDOG

CRC | EWM

Specifications:

- **Cores:** ARM Cortex-M4F @112 MHz max
- **Memory:** 2 MB Flash, 256 KB RAM (252 KB with ECC, 4 KB FlexRAM/EEPROM)
- **Temp Range:** Ta -40 to 125°C (Tj=135°C)
- **Power Supplies:** 2.7-5.5 V
- **Packaging:** 11 x 11 mm, 1 mm pitch 100 MapBGA (up to 89 usable pins). 20 x 20 mm, 0.5 mm pitch 144 LQFP (up to 128 usable pins). 24 x 24 mm, 0.5 mm pitch 176 LQFP (up to 156 usable pins).

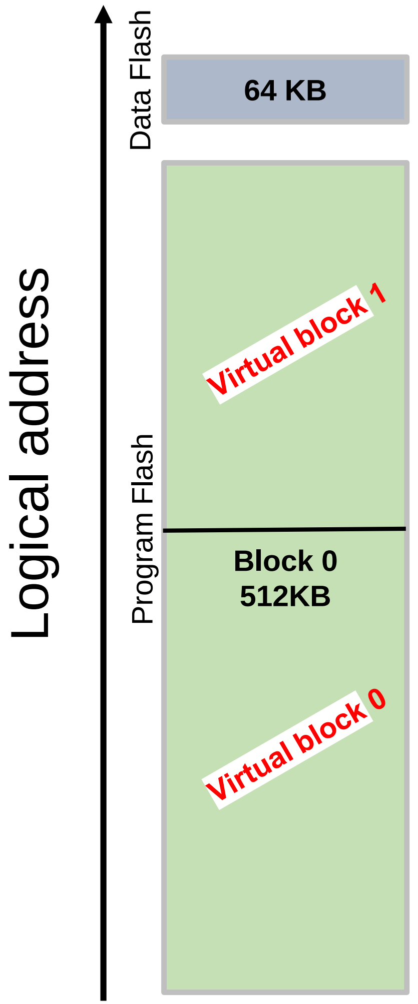
Key Features:

- **High Performance:** Powerful ARM Cortex-M4F core
- **Advanced Automotive Communication:** CAN FD + Ethernet + Audio interface
- **Functional Safety:** Developed as per ISO 26262 with target ASIL B
- **Security:** HW security engine (SHE+ compliant)
- **Low Power**: Low leakage tech. Best in class STOP current: 25-40 uA (device dependent)
- **Full solution offering:** AUTOSAR, SDK, Design Studio IDE

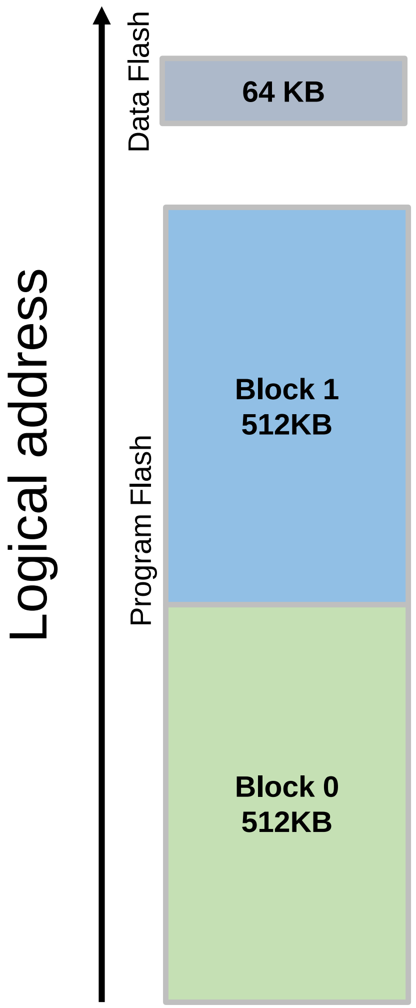
Footnote:

1. 112MHz not valid with M temp (125C).
2. Write or erase access to security (CSEc) or EEPROM is allowed only when device operating in RUN mode (up to 80MHz). No write or erase access to security and EEPROM allowed when device running at HSRUN mode (112MHz).

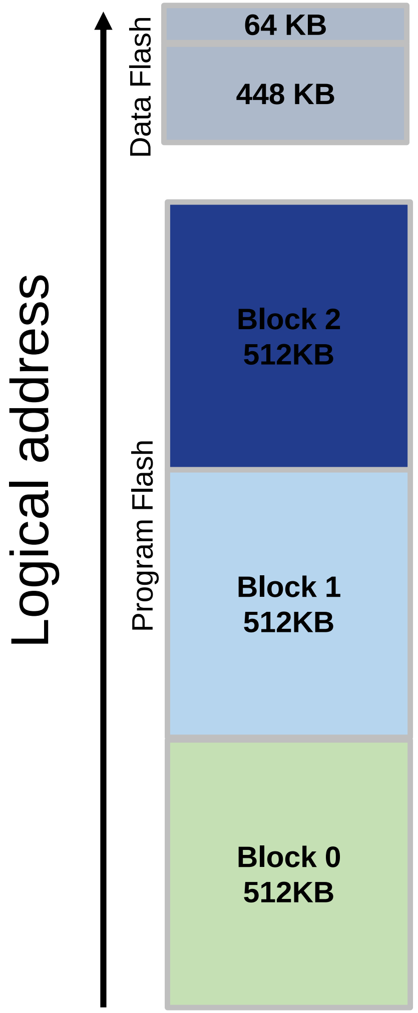
S32K14x: Flash Architecture



S32K144



S32K146



S32K148

S32K14x: Flash Architecture

FOTA Relevant Features:

Device	Program Flash	Program Flash sector size	Program Flash Read partitions	Flex memory	Flex memory sector size
S32K142	256kB	2kB	1	64kB	2kB
S32K144	512kB	4kB	1	64kB	2kB
S32K146	1MB	4kB	2	64kB	2kB
S32K148	1.5MB	4kB	3	512kB	4kB

- RWW between Dflash and Program Flash
- RWW between Program Flash read partitions

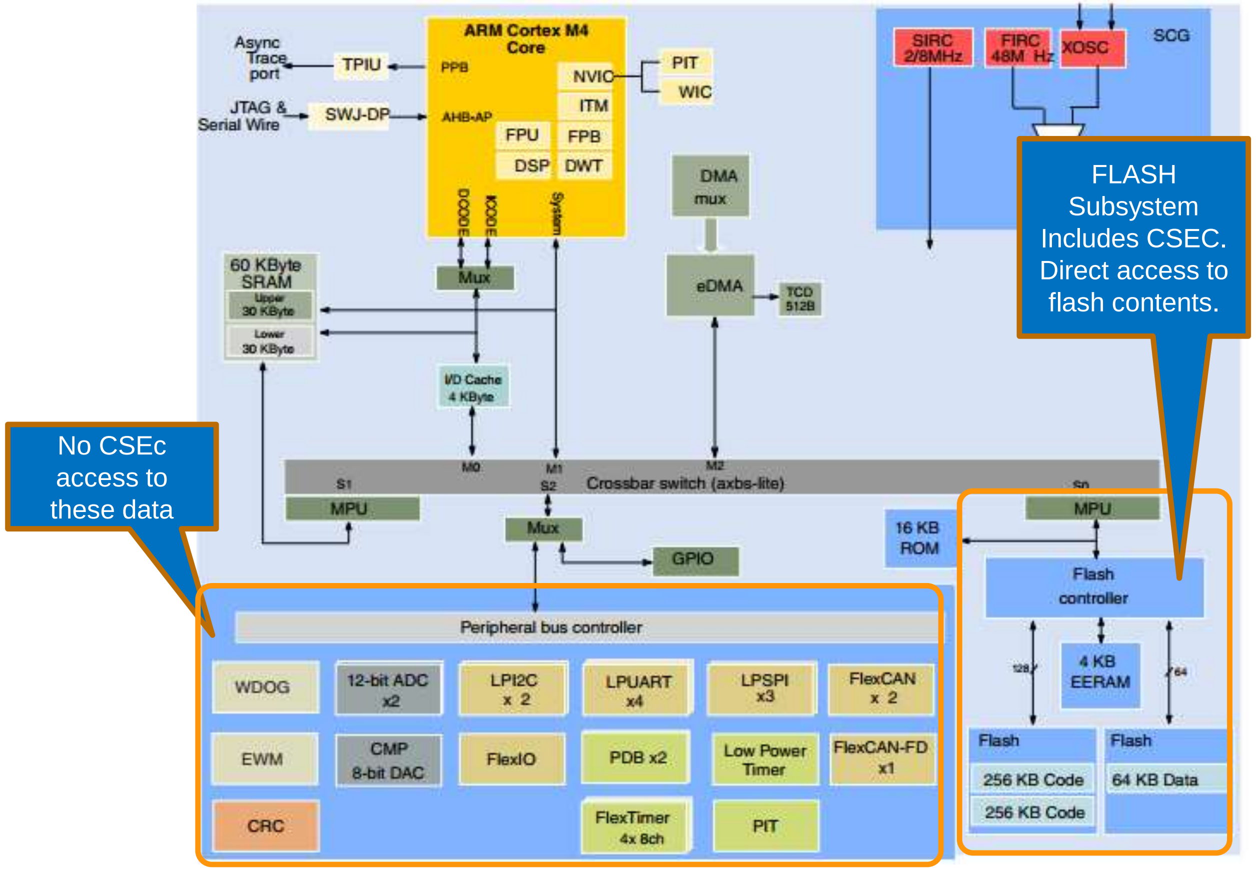
Key Additional Flash Features:

- **C90TFS** (Thin-Film-Storage) technology
- ECC support: **Single Bit Error Correction and Double Bit Error Detection**
  - 32bit ECC word in data flash
  - 64bit ECC word in program flash
- Access time: **Flash clock is about #1/4 of the core clock**

S32K Security Module (CSEc) – Overview

- SHE functionality moves from dedicated master module into the flash system
- **SHE Specification compliant**
- **Secure key storage** only accessible by CSEc
- **True Random Number** System
- **Sequential boot / parallel** boot supported
- CSEc supports **AES-128** with ECB, CBC and CMAC mode
- **Crypto Keys**
  - Several General-Purpose keys
  - Special Purpose keys (e.g. Secret, Master and Secure-Boot Key & CMAC)
  - Support of additional encrypted keys in public flash memory.

- KEY-Properies
  - Write-protection
  - Secure-Boot-Failure
  - Debug-Connect
  - Wildcard-UID
  - Key-Usage (key or CMAC)
  - Verify-Only
  - 28bit-Update-Counter

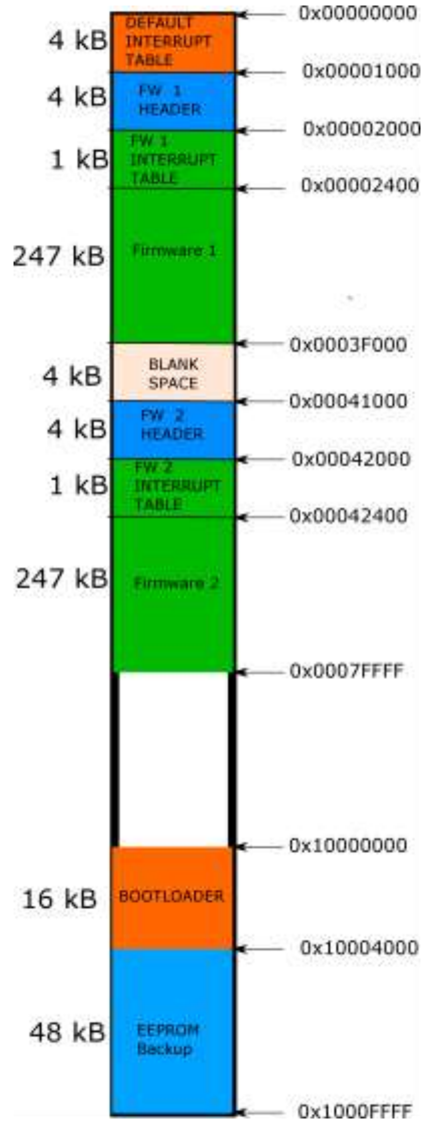


No CSEc access to these data

FLASH Subsystem Includes CSEC. Direct access to flash contents.

S32K144 Use Case

S32K144 Use Case: Memory Map for A/B Swap



- Default Interrupt table and bootloader not erased.
- 0x000000004 -> stores bootloader Reset Handler
- Reset Handler located at Bootloader space
- FW HEADER:
  - Fw version .
  - Developers information.
  - Validation.
  - Erased/Updated after each firmware update
  - Size: 4kB (sector size)
- FW size 248kB (62 sectors)
- RWW between bootloader and firmware application.
- EEPROM: Store secure keys, application usage.

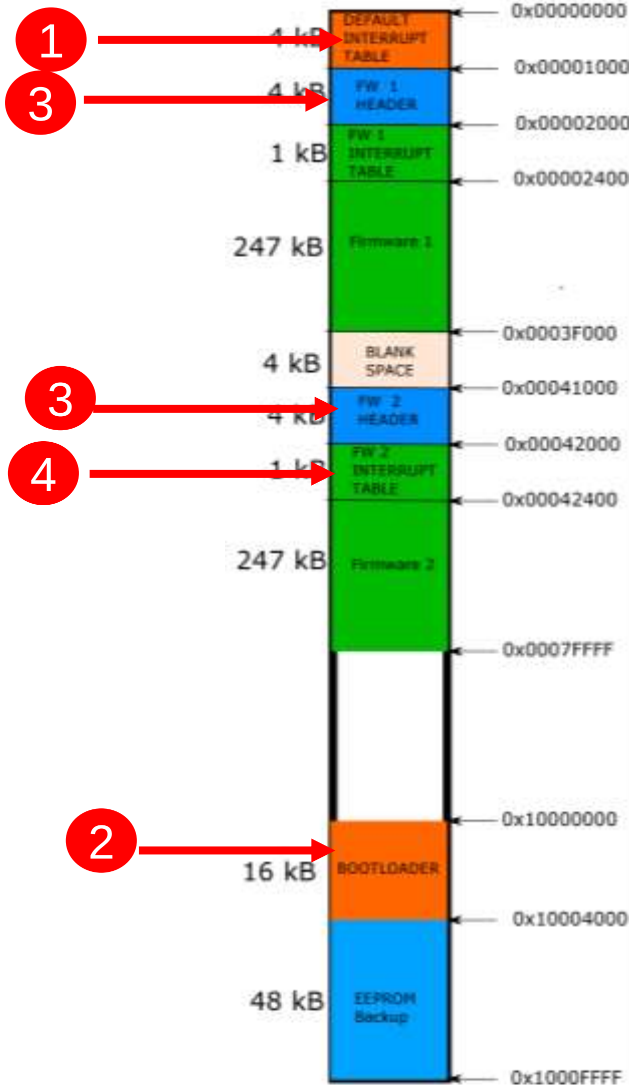
S32K144 Use Case: Memory Map for A/B Swap

- 1. After Reset: fetch PC value @ 0x00000004

- 2. Bootloader init peripherals

- 3. Bootloader search for oldest and newest image.
  - Check FW Header information
  - Value 0x55AA55AA, at end of fw header
  - Assign FW to be updates (Oldest)

- 4. Jump to newest application
  - Relocate VTOR table
  - PC fetch value from new firmware interrupt table



S32K144 Use Case: Memory Map for A/B Swap

- 1. After Reset: fetch PC value @ 0x00000004

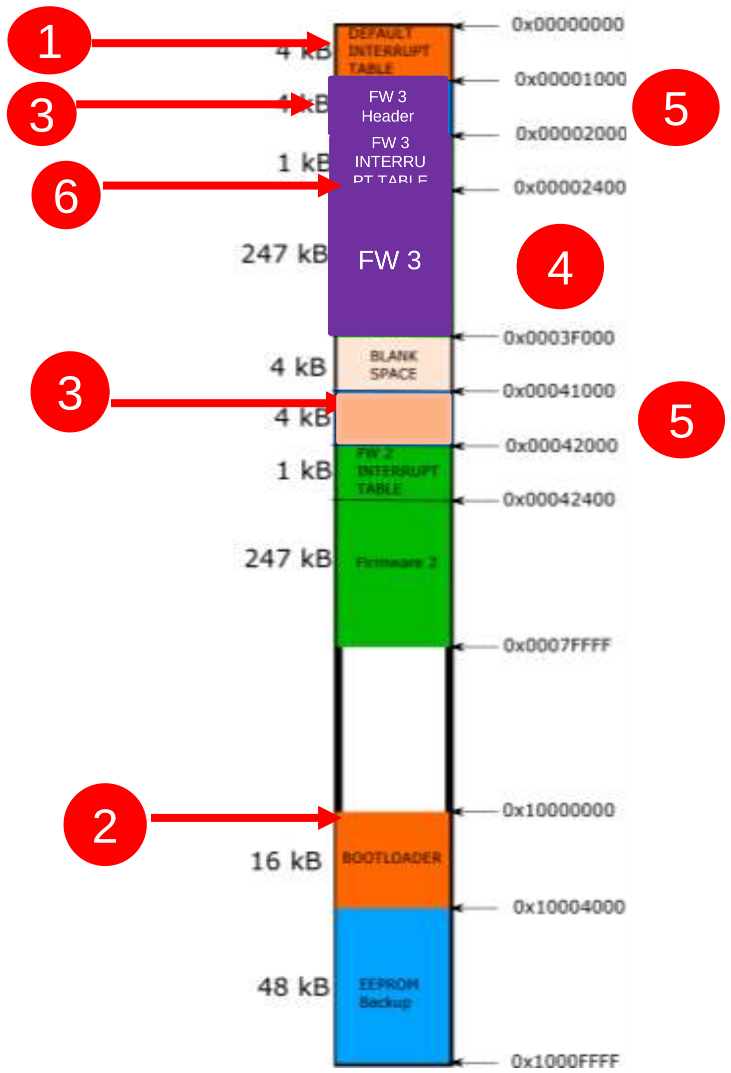
- 2. Bootloader init peripherals

- 3. Bootloader search for oldest and newest image.
  - Check FW Header information
  - Value 0x55AA55AA, at end of fw header
  - Assign FW to be updates (Oldest)

- 4. Update trigger received.
  - Receive header first
    - Validate is a new version
  - Start updating new firmware in oldest location

- 5. Update Completed
  - Deinit bootloader peripherals
  - Update new firmware header
  - Erase/Update older firmware header

- 6. Jump to new application
  - Relocate VTOR table
  - PC fetch value from new firmware interrupt table

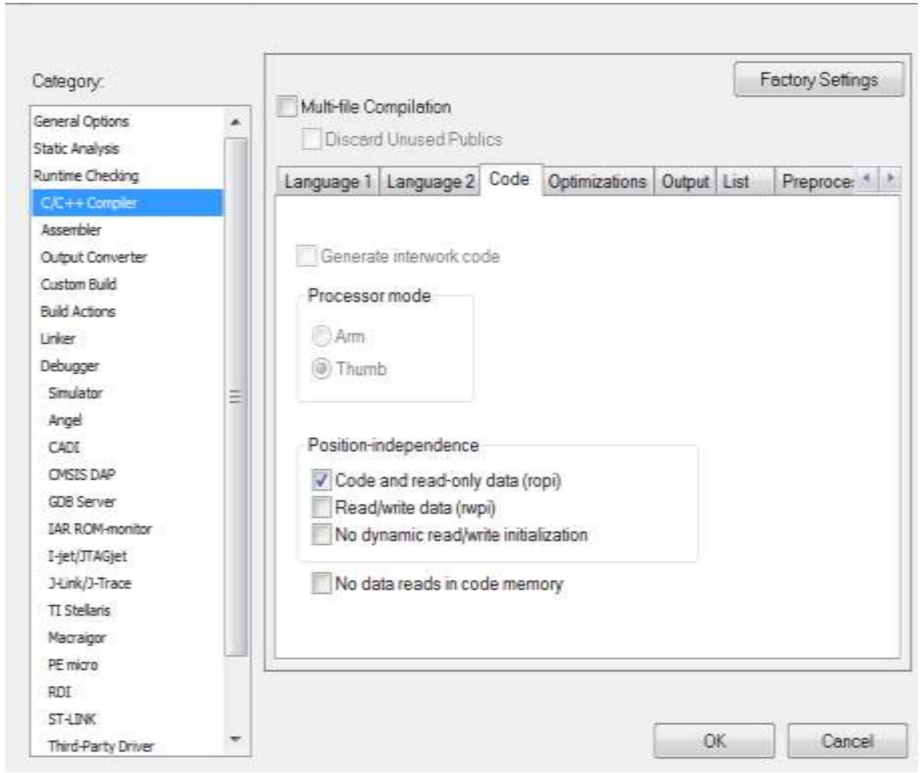


S32K144 Use Case: A/B Swap Options Without Flash Remapping

- Problem:
  - 2 images in different physical address.
  - No flash swap, flash remapping feature

- Solutions:
  - Separate object file for each firmware.
    - Requires more overhead in file management!
  - Position independent code
    - Same linker file for all firmware updates
    - No file management
    - No absolute branches
    - Offset to each interrupt table entry needs to be added. Done automatically by bootloader!
    - Addresses of the interrupt table, should be modified.

IAR ropi feature



□ Multi-file Compilation

□ Discard Unused Publics

□ Generate interwork code

Processor mode

○ Arm

● Thumb

Position-independence

☑ Code and read-only data (ropi)

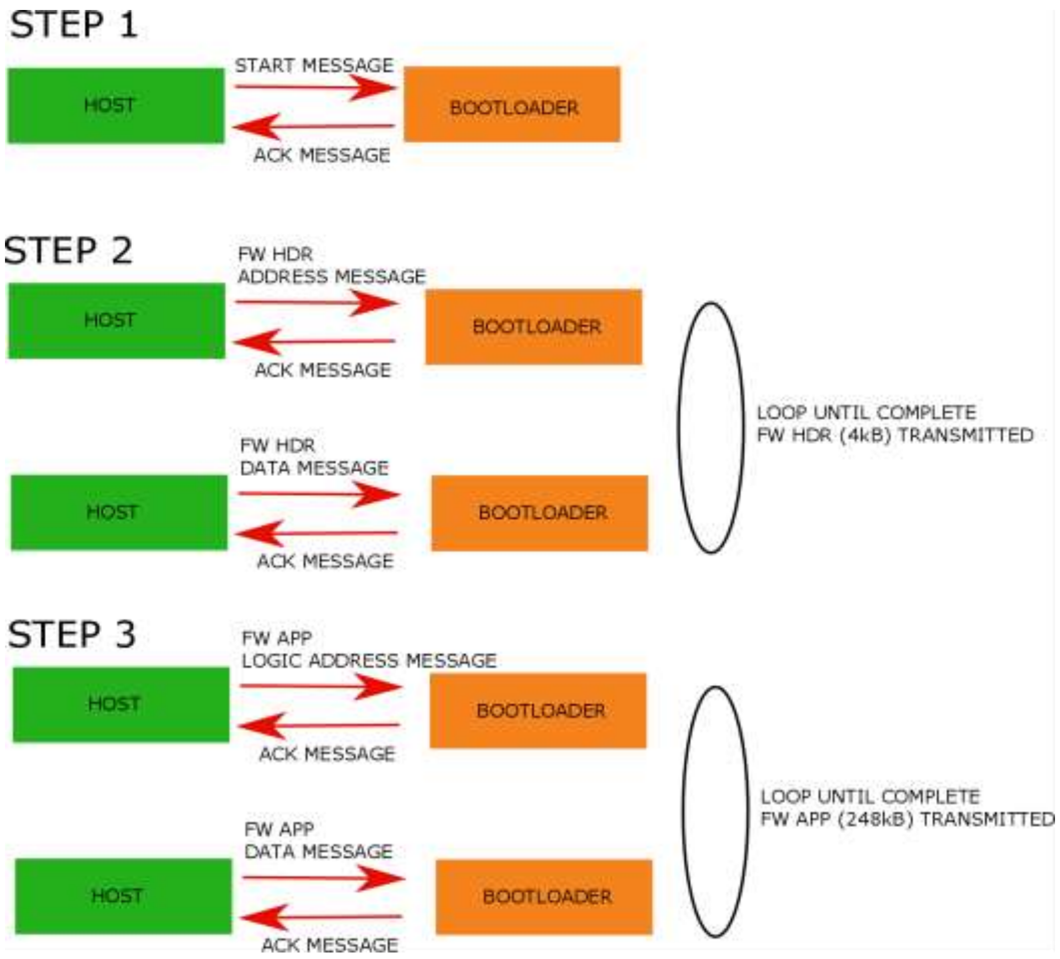
□ Read/write data (rwpi)

□ No dynamic read/write initialization

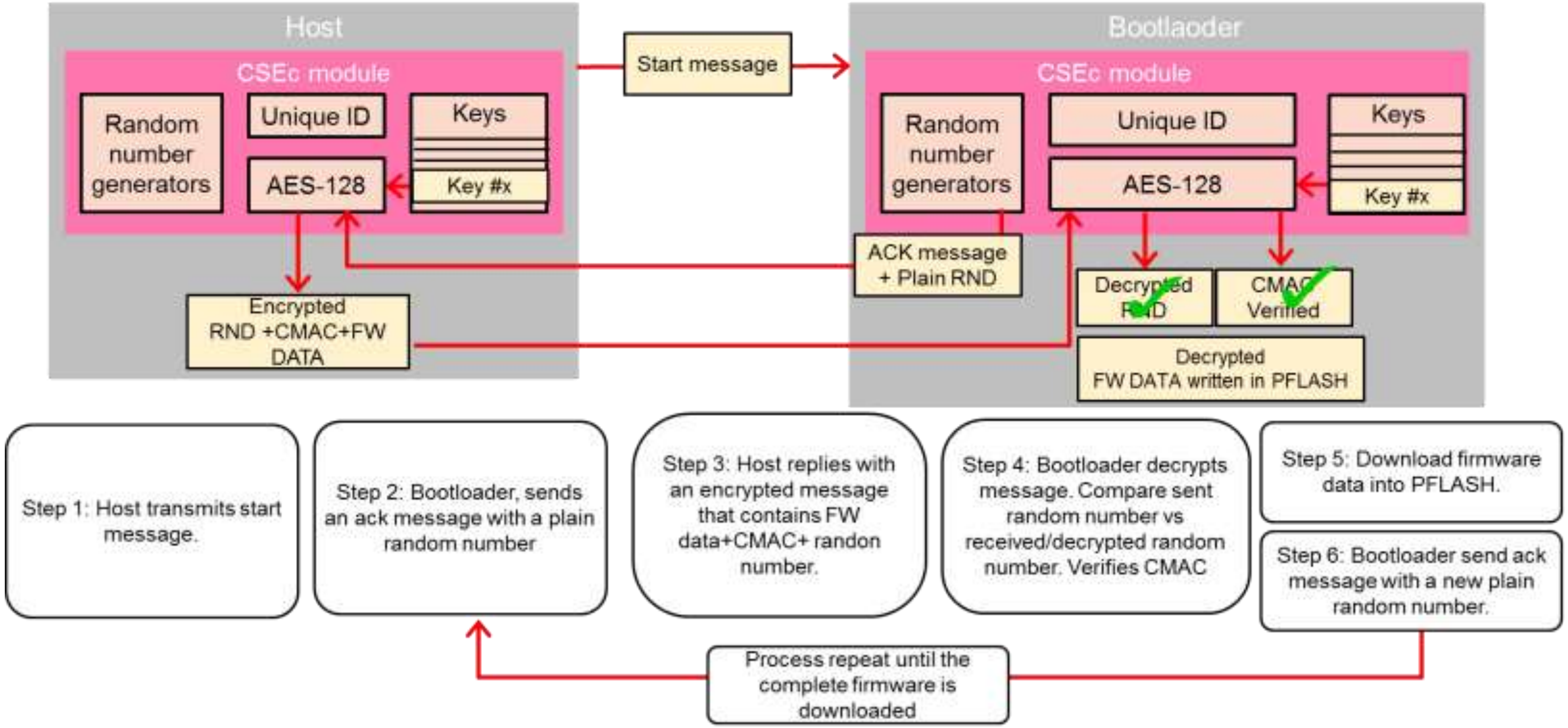
□ No data reads in code memory

S32K144 Use Case: Communication Process

- Step 1: Trigger update
  - Communication Message from Host to edge node (bootloader fw)
  - Response of ack form host to edge node.
- Step 2: Transmit Header
  - Host sends address
  - Edge node responds with Ack
  - Host sends header data
  - Edge node validate data
  - Edge node responds with Ack
- Step 3: Transmit Application
  - Host sends app logic address
  - Edge node responds with Ack
  - Host sends app data
  - Edge node receives and write data into flash
  - Edge node responds with Ack



S32K144 Use Case: Secure Communication Process



Step 1: Host transmits start message.

Step 2: Bootloader, sends an ack message with a plain random number

Step 3: Host replies with an encrypted message that contains FW data+CMAC+ randon number.

Step 4: Bootloader decrypts message. Compare sent random number vs received/decrypted random number. Verifies CMAC

Step 5: Download firmware data into PFLASH.

Step 6: Bootloader send ack message with a new plain random number.

Process repeat until the complete firmware is downloaded

- Random number: protects against replay attacks → Authenticity and freshness of message
- Encryption: protects against eavesdropping → Confidentiality
- CMAC → Data integrity

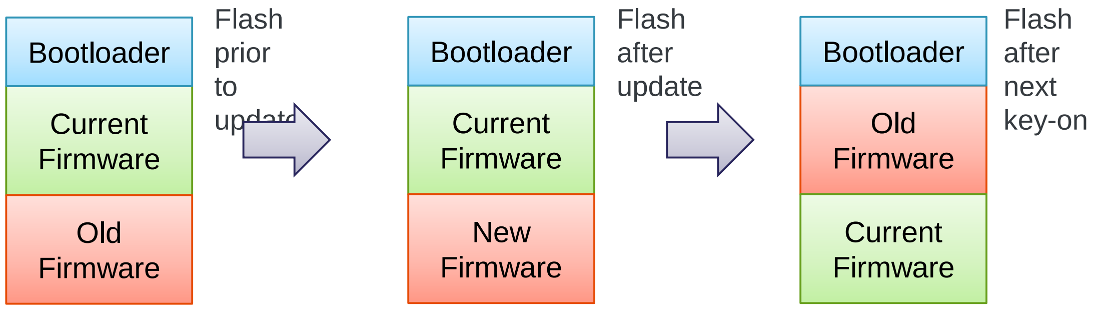
S32K Next Gen End Node (OTA Client)

Over The Air (OTA) Update Methods

S32K next generation will fully support both update methods:

A/B:

2 versions of firmware exist in internal flash.



Advantages:

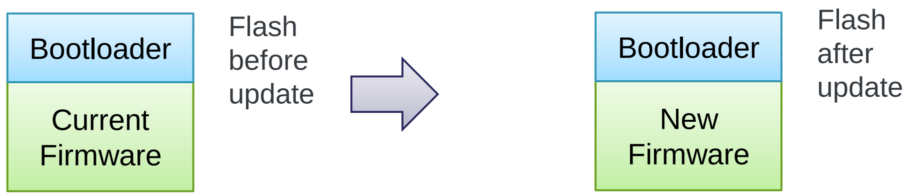
- Update can be carried out whilst application is actively running from flash
- Always have original firmware to roll back to in case of issue
- Vehicle always available – guaranteed no vehicle downtime regardless of update errors

Cost:

- Requires 2x flash application storage

In Place:

Update is performed on top of existing version



Advantages:

- No need for additional flash (although 1 additional empty flash block typically required during update process)

Cost:

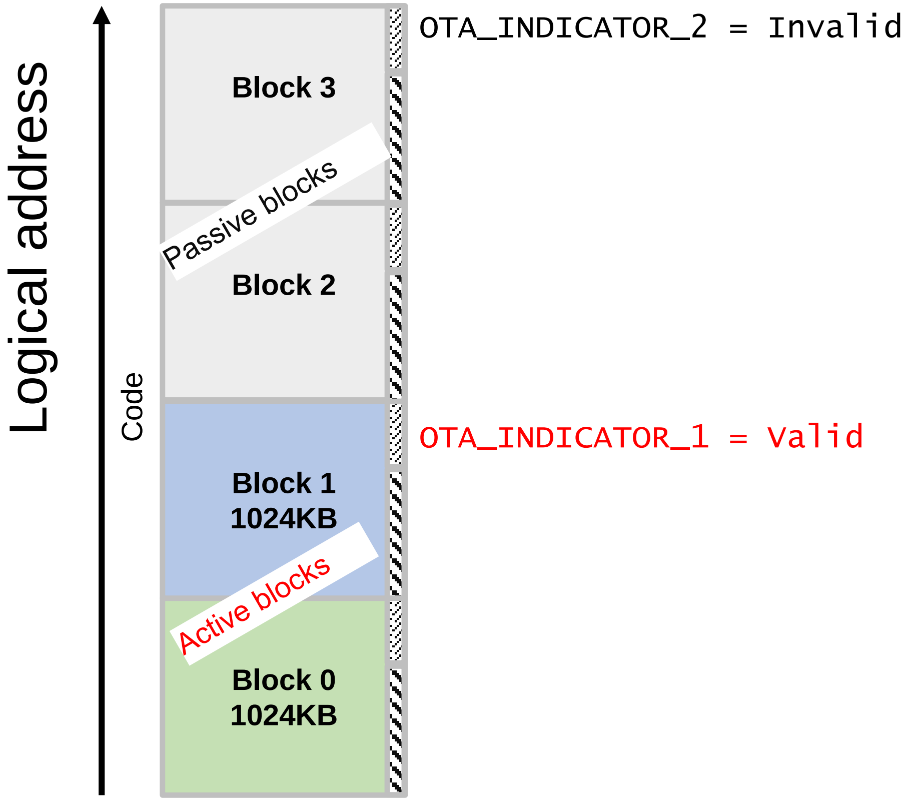
- Requires vehicle downtime during update process
- Not possible to instantly “roll-back” if an issue occurs
- Higher risk to have an ECU inoperable

S32K Next Gen: OTA Flash Features

Flash Read-While-Write Functionality

This feature allows for the firmware to be updated whilst the vehicle is in motion

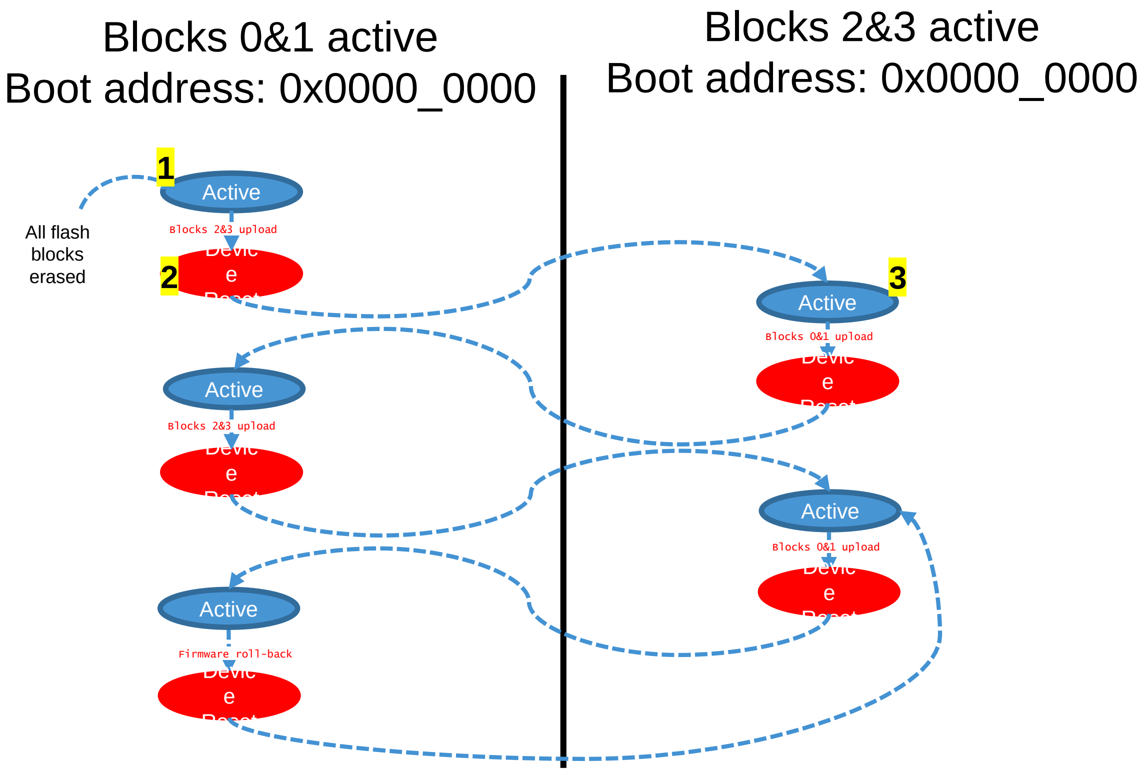
- When OTA is enable in the part, device flash divides in 2 types of blocks.
- Allows for the flash to be updated whilst simultaneously executing code from it
- Active blocks is the where the application code is located.
- Passive blocks is where the rollback image is located.
- RWW available between active and passive blocks. Allows for the flash to be updated whilst simultaneously executing code from it



S32K2xx: OTA Remapping Features

Flash Swap

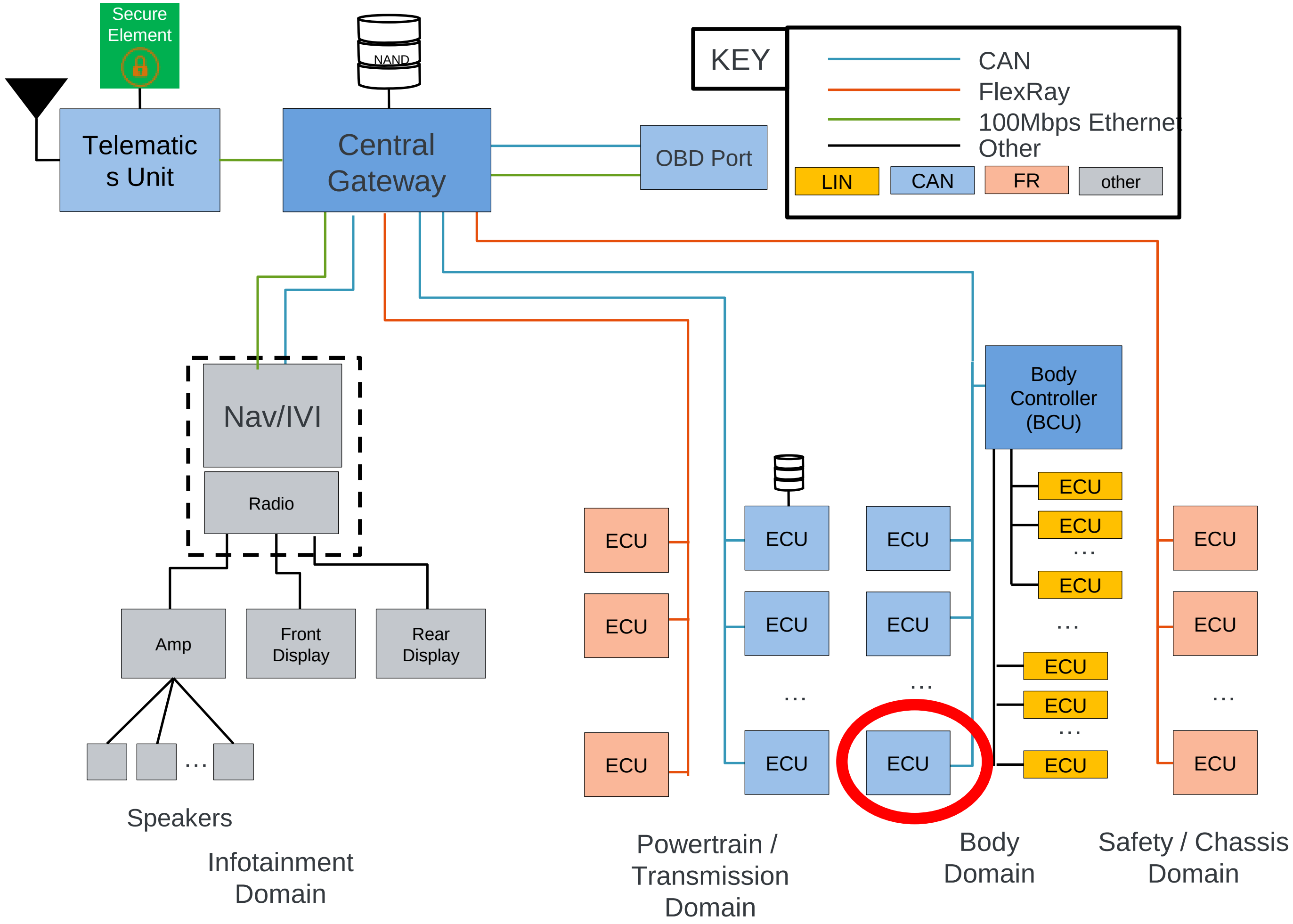
- Allows for instant switching between firmware versions
- Automatic firmware translation
- Instant version swap after device reset.
- Rollback capability.



1. Cores executes firmware from flash blocks 0&1 (active) after all flash blocks are erased.
2. After new image is uploaded to passive flash blocks 2&3 (OTA indicator updated) a device reset can be triggered.
3. After device reset, passive flash blocks 2&3 will become active, mapped at low address space and new firmware image will execute.

S32K Next Gen OTA Client Use Cases

OTA Use Case: 2 FW Versions in Internal Memory



Example ECU A
Flash: 2x internal flash available
Security: Supports CMAC authentication and AES-128 decryption
Connection to Gateway: FlexRay

Vehicle Downtime: **none**
Security: **high**

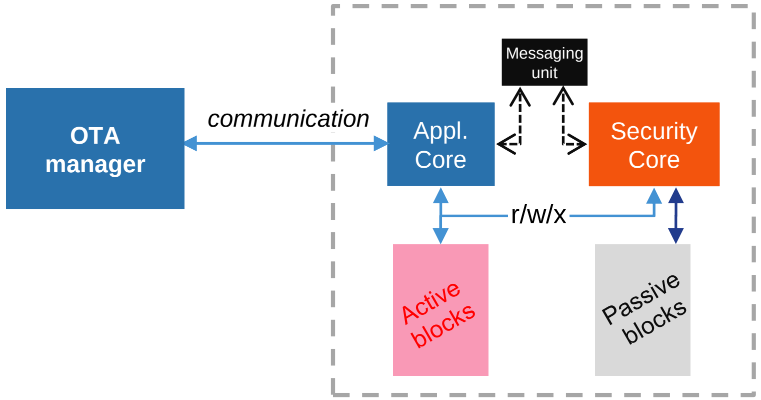
Steps:

- Encrypted binary trickle downloaded and stored onto empty “B” flash on ECU.
- Firmware is decrypted and integrity checked as it is downloaded. Allows end-to-end security
- Once download complete, GW switches ECU to use new firmware from next boot

S32K Next Gen Over-the-Air Update – Use Cases

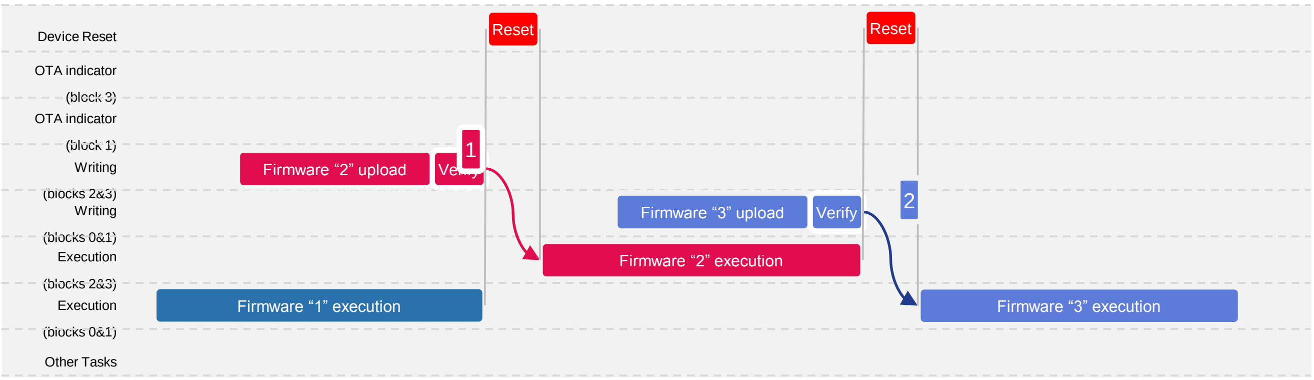
Use case: Both active and passive images stored in the internal code flash

FOTA Hardware Architecture

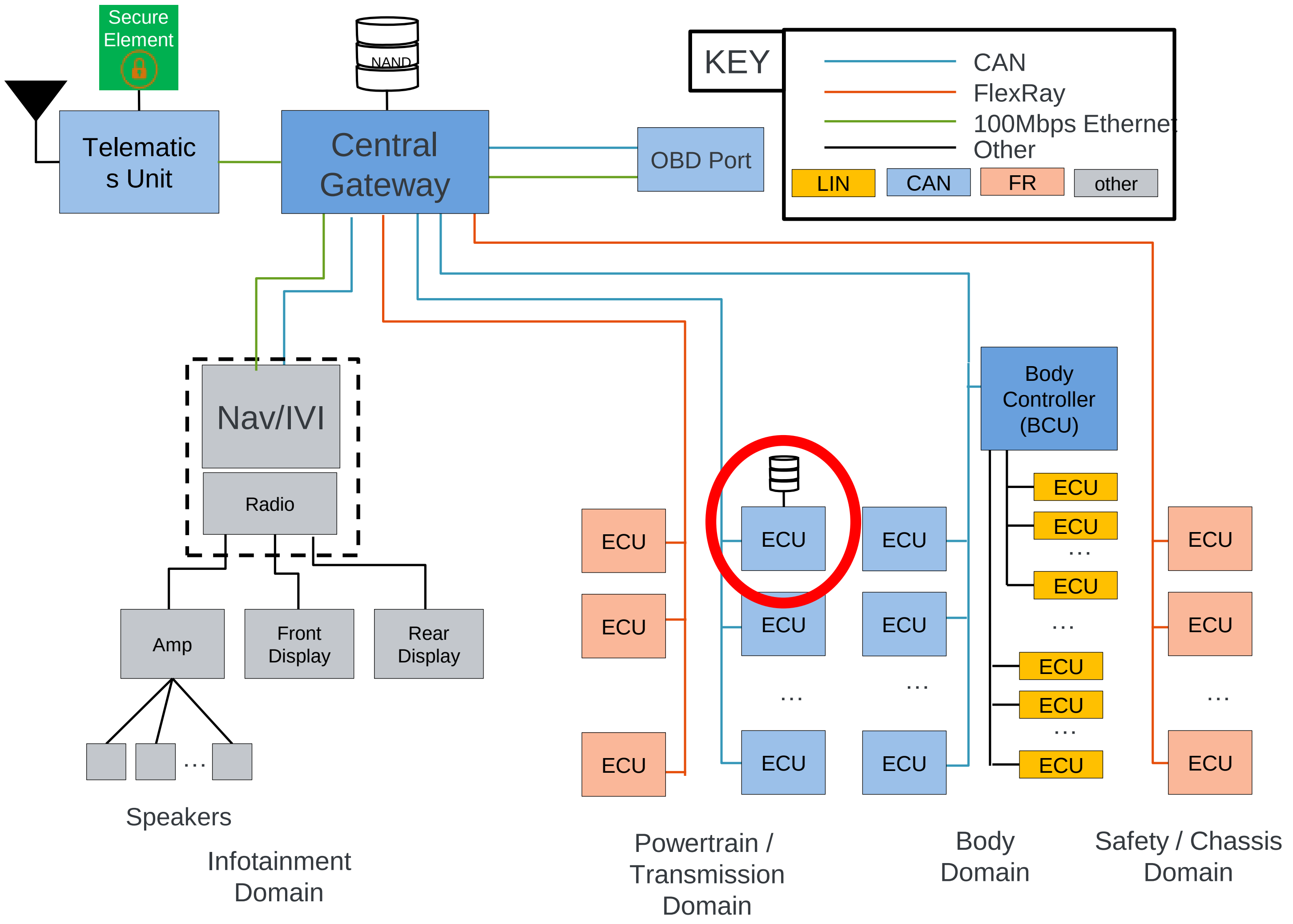


- Current firmware executes and simultaneously uploads new firmware image into passive flash blocks
- After new image is uploaded into passive flash blocks, verified and OTA indicator in passive flash block updated device can initiate reset
- After device reset new image will execute

Firmware Upload



OTA Use Case: 2 FW in Internal Memory + Local Repository



Example ECU B
Flash: Internal flash with external NAND flash for local storage of a local firmware repository.
Security: Supports CMAC authentication and AES-128 decryption
Connection to Gateway: CAN

Vehicle Downtime: **none**
Security: **high**

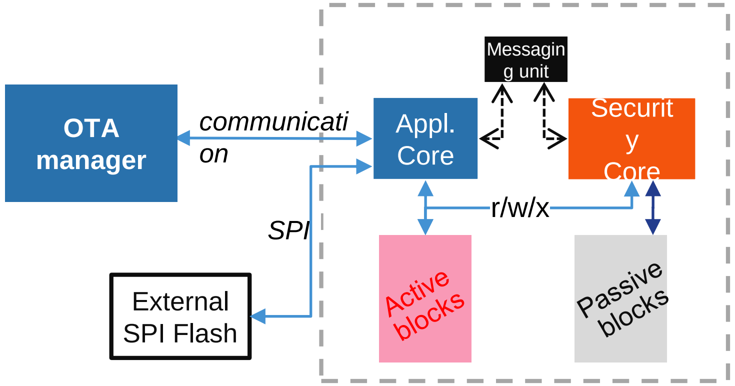
Steps:

- Encrypted binary downloaded and stored onto GW. Checks authentication and integrity.
- GW sends to ECU as a background task – stored in external NAND.
- Update triggered by GW. Binary decrypted by ECU.

S32K Next Gen Over-the-Air Update – Use Cases

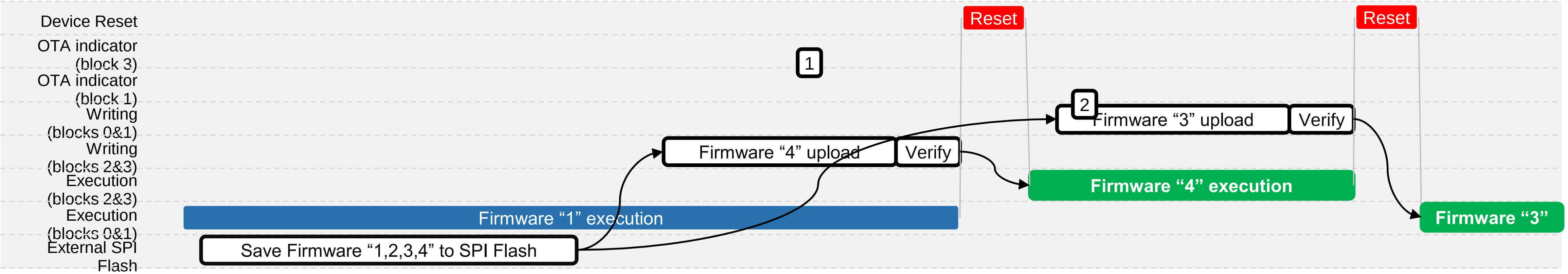
Use case: Keep several application images in external SPI flash

FOTA Hardware Architecture

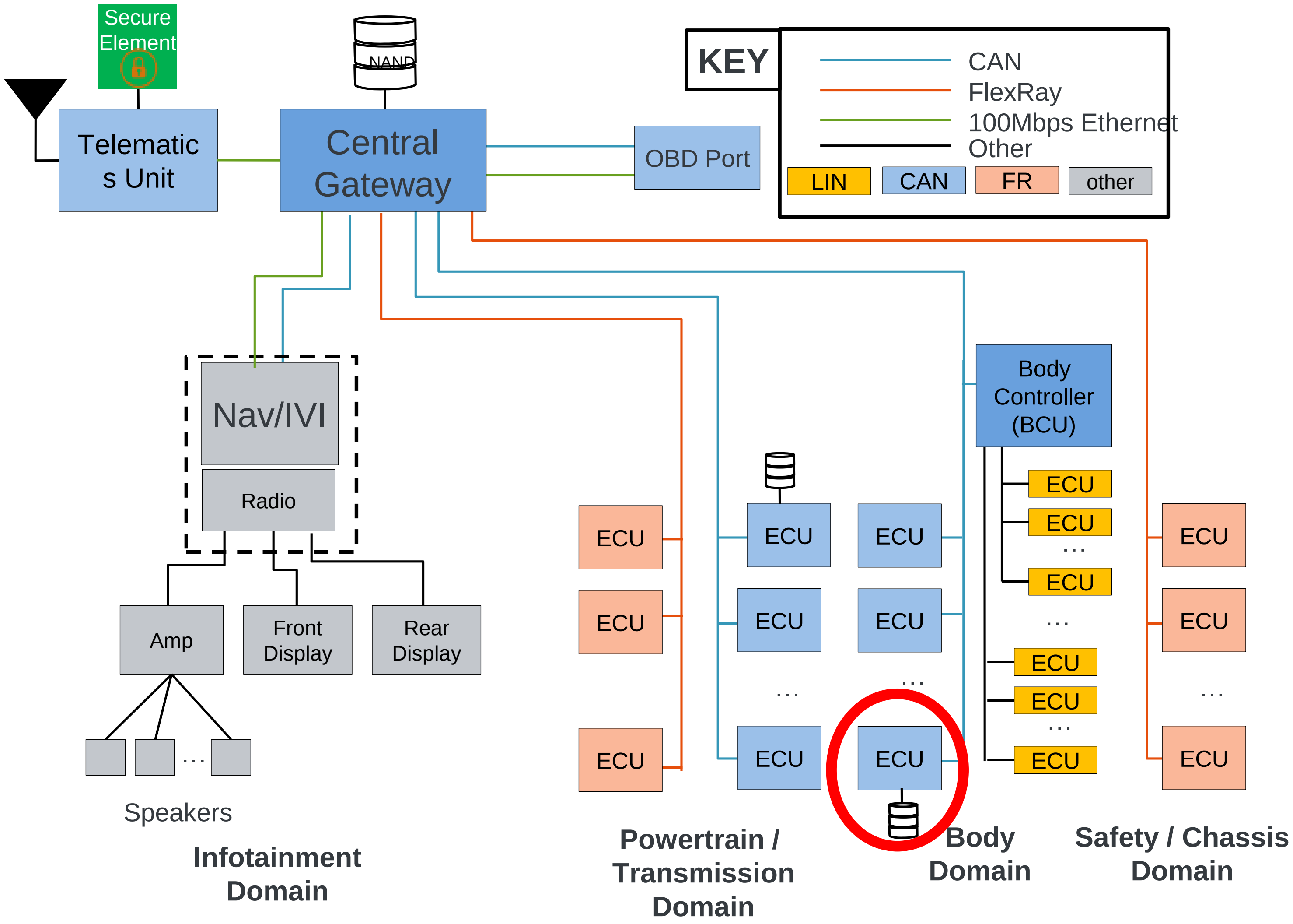


- Current firmware executes in parallel with storing firmware images within an external SPI flash
- Selected firmware will uploaded to passive flash blocks
- After selected image is uploaded to passive flash blocks, verified and OTA indicator in passive flash block updated device can initiate reset
- After device reset selected new image will execute

Firmware Upload



OTA Use Case: 1 FW in Memory + External Memory



Example ECU C

Flash: Internal flash with external NAND flash for local storage of new binary

Security: Supports CMAC authentication and AES-128 decryption

Connection to Gateway: CAN

Vehicle Downtime: **long**

Security: **high**

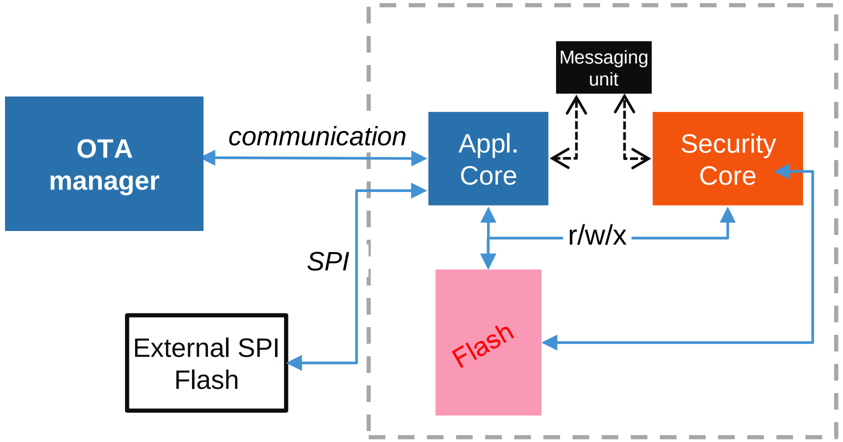
Steps:

- Encrypted binary downloaded and stored onto GW. Checks authentication and integrity
- GW sends to ECU as a background task – stored in external NAND
- Update triggered by GW carried out during vehicle downtime. Binary decrypted by ECU

S32K Next Gen Over-the-Air Update – Use Cases

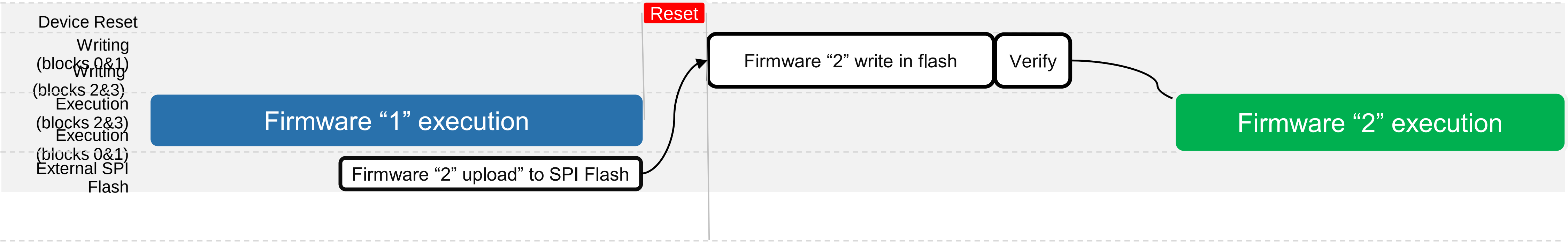
Use case: In place update using external flash.

FOTA Hardware Architecture



- In case the whole flash memory is required for firmware.
- Current firmware executes in parallel, while storing firmware images within an external SPI flash.
- After device reset selected new image will be uploaded to device flash.
- After verification, new firmware image is executed from flash.

Firmware Upload



Summary

Market Problem

- ECU reprogramming outside garage. Seamless update for driver (zero down time).
- Always guarantee a working firmware in ECU as backup.
- Attractive target for hackers. Opens a door for security vulnerability.

Solutions

In vehicle OTA architecture

- OTA manager.
- OTA clients.

Reliable and robust update

- Power and communication loss detection.
- Multiple version of firmware available.

Attack protection

- Against firmware stealing.
- Against malicious firmware installation.

S32K Features

Memory features

- Read while write between flash banks.
- Automatic firmware address translation.
- OTA agent firmware.
- Backup firmware.

OTA client features

- Rollback functionality.
- In hw firmware version control.
- In hw brownout and communication monitor.

Security hardware

- Encryption/ decryption of data.
- Firmware authentication check.

Summary

- OTA: In field device reprogramming.
- Vehicle in field reprogramming its a new trend.
- Different reprogramming methods are applied to each vehicle ECU.
- NXP devices are prepared across different use cases.
- New use cases are always welcome.

Additional Resources From NXP

OTA Insights

- NXP Automotive Software Over-the-Air Updates Video
- "Making Full Vehicle OTA Updates a Reality" white paper by Daniel Mckenna
- Body Electronics: An OTA Solution for Edge Nodes Using S32K by Osvaldo Romero

Gateways and Security

- NXP Central Gateway Site
- NXP Security Layers for Connected Cars

NXP Products to Support OTA

- MPC574xB/C/G Automotive MCUs (body control and gateways)
- S32K Automotive General Purpose Microcontrollers (end nodes)

Q&A and Fill Out Surveys

SECURE CONNECTIONS
FOR A SMARTER WORLD

www.nxp.com