D3.3.- Mid UP2DATE Architecture definition

V1.0

Document information

Contract number	871465
Project website	https://h2020up2date.eu/
Contractual deadline	M21
Dissemination Level	PU
Nature	Report
Author	IKL (Irune Agirre, Irune Yarza, Imanol Mugarza)
Contributors	IAV (Jan Loewe, Thorsten Moehring), OFF (Gregor Nitsche), MM (Freda Yuri)
Reviewer	MM (Stefania Botta), QM (Peio Onaindia)
Keywords	Safety, security, update management, architecture

Notices:

This project and the research leading to these results has received funding from the European Community's H2020 program [H2020-ICT-2019-2] under grant agreement 871465

Change log

VERSION	DESCRIPTION OF CHANGE
V0.0	First draft by IKL
V0.1	Initial description of secure communications
V0.2	IKL inputs completed
V0.3	IAV inputs integrated
V0.4	MM inputs integrated
V0.5	Corrections based on MM and QM review
V1.0	First version ready for submission

Table of contents

List of figures

List of tables

Acronyms and Abbreviations

AES	Advanced Encryption Standard
CA	Certificate Authority
CAN	Controller Area Network
CAN-FD	Controller Area Network Flexible Data-Rate
CIA	Confidentiality, Integrity and Authenticity
CMAC	Cipher-based message authentication code
ECU	Electronic Control Unit
FMECA	Failure Modes, Effects and Criticality Analysis
FSM-CSM	Functional Safety and Cyber-security Management
HSM	Hardware Security Module
ICT	Information and communications technologies
MCCPS	Mixed-Criticality Cyber-Physical Systems
OBD	On Board Diagnosis
OEM	Original Equipment Manufacturer
OTASU	Over-the-air Software Updates
PKI	Public Key Infrastructure
RoT	Root-of-Trust
SASE	Safety and Security
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
UDS	Unified Diagnostic Services
VIN	Vehicle Identification Number
VPN	Virtual Private Network

1 EXECUTIVE SUMMARY

This report is the intermediate version of the UP2DATE architecture definition and takes the initial architecture definition of deliverable D3.2 as baseline. It presents the technical advances done in WP3 tasks T3.2 (“Definition of UP2DATE Safety and Security (SASE) Architecture”), T3.3 (“Continuous safety and security assessment”) and T3.4 (“Definition and implementation of secure communication mechanisms for dynamic software evolution in critical heterogenous platforms”) at month 21 of the project.

During this period, the initial architecture definition has been complemented with further state machine definitions, system reaction to errors and a description of the update testing process based on the SASE properties previously defined in D3.1 and the contract-based virtual integration and compatibility tools defined in D4.1. In addition, the deliverable presents a combined safety-security risk assessment approach, which has been applied to the initial architecture and the definition of the corresponding risk treatment through safety and security countermeasures. As the main objective of task T3.3 is evaluating these outcomes with a certification authority, this information (together with the initial architecture definition of D3.2) has been reported in a self-contained safety-security concept that was delivered to the external certification authority (included in Annex A of this report). In Annex B, we also provide the outcomes of the review meetings held with TÜV Rheinland in the form of slides used in the meeting and the resulting list of open points. Finally, the concept definition and design of secure communications is also included in this report.

2 INTRODUCTION

One of the objectives of UP2DATE rests on carrying out an assessment of safety and security certifiability of the concepts for Over-the-air Software Updates (OTASU) in Mixed-Criticality Cyber-Physical Systems (MCCPS) with an external certification authority. This assessment seeks to define a clear route for the future certification of UP2DATE contributions and for identifying possible certification implausibilities of the approach soon enough to address them in the course of the project.

The certification activities of UP2DATE will be accomplished in a number of incremental stages, starting by the definition and review of the safety and security (SASE) concept included in this report. This initial SASE concept includes a description of the strategies and measures adopted to guarantee the system functional safety and cyber-security based on the cross-domain architecture presented in previous UP2DATE deliverable D3.2. The definition and early review of a safety concept is a common step on the safety lifecycle of critical systems, to evaluate the suitability of the defined techniques before starting with the detailed design and implementation phase. Since emerging critical systems have external connectivity, as in the case of the UP2DATE architecture, the concept also addresses security features, as the system is susceptible to security attacks that could also have an influence on safety. Therefore, the SASE concept considers both safety and security threats and countermeasures and their mutual influence following the recommendations of IEC TR 63069 technical report [1]. To this end, the architecture definition of D3.2 has been updated with further refinements and adaptations and a risk assessment of it has been conducted. Based on the results of the risk assessment, a number of countermeasures were defined, and all this has been documented on a self-contained document delivered to the external certification authority (TÜV Rheinland) for review.

This report describes the safety and security assessment strategy and summarizes the adaptations done to the architecture since M15 in Section 3. The details of the new version of the architecture, together with the risk assessment and countermeasure definition, is then included in the SASE concept, a self-contained document delivered to TÜV Rheinland and attached to this report in Annex A. Note that both Annexes of this document are separate materials used with the certification body and therefore are included here as separate documents, with their own frontpage and page numbering. Apart from the SASE concept, Section 4 includes a description of the early design of secure communications for update package and monitoring data transmission as a result of Task T3.4 of the project. Finally, Section 5 draws the main conclusions.

2.1 Relation to other UP2DATE deliverables

This report is the second in a series of three incremental deliverables defined in the Description of Action:

- **D3.2 Initial UP2DATE Architecture definition,** due on M15. This report describes the first version of the architecture which is built upon a joint safety and security update management procedure. Both the procedure and the architecture are based on a combination of project requirements with safety and security requirements coming from standards.

- **D3.3 Mid UP2DATE Architecture definition,** due on M21. This deliverable will report the mid version of the architecture. This version extends D3.2 with further details on the safety and security design considerations, mechanisms and arguments on the different update approaches and architecture supported by a risk assessment (i.e., the SASE concept).
- **D3.4 Final UP2DATE Architecture safety-security concept,** due on M27. This will be the final version of the UP2DATE architecture and its *SASE concept*. As in the previous case, it will include an incremental redesign of the architecture based on implementation improvements and on the feedback of the certification authorities. In addition, this final version will also consider and report the implications of adapting the proposed solution to domain specific standards (e.g., EN 5012x, ISO 26262, ISO 21434, TS 50701).

As in previous version, the architecture of this report is used as reference for WP4 and WP5 update and monitoring middleware respectively. In particular, the high-level state machines incorporated in this version of architecture definition are then detailed in WP4 update workflow definition of D4.2.

3 SAFETY AND SECURITY ASSESSMENT

The goal of the safety-security concept assessment is to get an independent review report on the suitability of the research concepts of UP2DATE (theoretical approach). To this end, following the certified Functional Safety and Cyber-security Management (FSM-CSM) methodology of Ikerlan [2], a safety-security concept is defined and delivered to the certification authority for review.

The SASE concept (included in Annex A of this report) is comprised of the system specification (i.e., the requirements), the high-level architecture definition and a risk assessment that aims to identify potential system hazards and their impact on the architecture. The first two aspects were already defined in deliverable D3.2 and have been included in the SASE concept with some complementary information described in next Subsection 3.2.

3.1 Assessment Strategy

The incremental assessment strategy includes two main activities, first the cross-domain architecture is evaluated on an early SASE concept and then on the second stage, a consolidated SASE concept will be grounded to at least one case study in task T6.3 of WP6. To this end, the following activities and procedure have been agreed with TÜV Rheinland, a subcontracted party that will act as external certification authority:

1. Activity 1 [2021] Safety and security concept (IEC 61508 and IEC 62443) of the safe and secure software update procedure and architecture
2. Activity 2 [2022] Safety and Security concept (IEC 61508 / EN 5012x and IEC 62443) of the safe and secure software update procedure and architecture on a specific Railway Use case
3. Contrast on further evolutions of the UP2DATE architecture

This report focuses on Activity 1, for which the plan of Figure 1 has been defined.

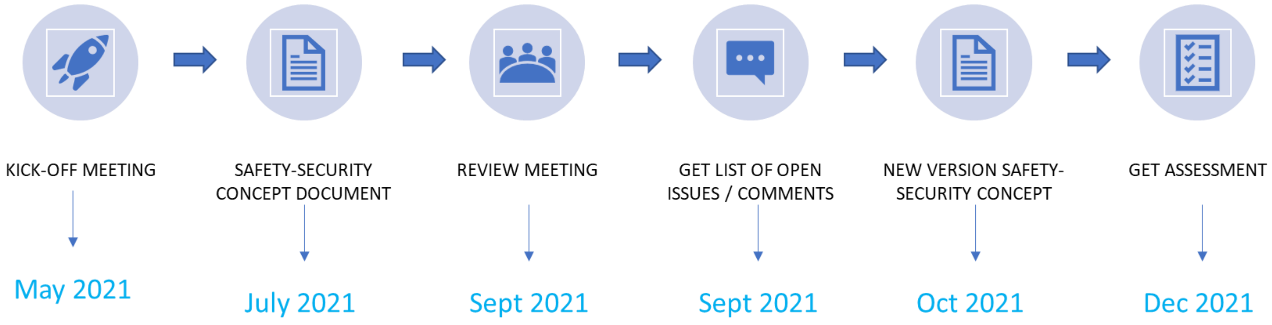
KICK-OFF MEETING
May 2021
SAFETY-SECURITY CONCEPT DOCUMENT
July 2021
REVIEW MEETING
Sept 2021
GET LIST OF OPEN ISSUES / COMMENTS
Sept 2021
NEW VERSION SAFETY-SECURITY CONCEPT
Oct 2021
GET ASSESSMENT
Dec 2021

Figure 1: SASE concept review steps and plan

As shown in this plan, this deliverable includes part of this process (SASE concept (Annex A), review meeting slides and list of open issues (Annex B)). The resolution of such open issues and the final assessment will be delivered in the final version of the UP2DATE Architecture safety-security concept, due on M27 (D3.4).

3.2 Architecture refinements

Annex A of this deliverable includes the SASE concept that describes the mid version of the architecture. This architecture is based on that defined in D3.2 and includes improvements and refinements based on the project progress. All these changes are directly described in the Annex, which is a self-contained document created for the review by certification authorities. In this section we identify which are the main differences with respect to D3.2 and we point out to the Annex section where they are described.

3.2.1 Safe and secure software development process

The SASE concept introduces the safe and secure software development process for updates (refer to Section 2.2 in Annex A of this deliverable). In line with the recommendations of functional safety standards, modularity is a key aspect for software modifications. Therefore, Section 2.2.1 of Annex A described the component-based software architecture that allows the system to be composed of qualified mixed-criticality software components (i.e., compliant-items according to IEC 61508 [3]). Then, Section 2.2.2 of Annex A presents the update testing process, a composable testing solution that combines a worst-case performance and resource usage analysis with a novel virtual compatibility and integration testing technique.

3.2.2 Operation modes

Deliverable D3.2 presents the reference architecture, defining the services provided by each element: server, gateway, and end-device. The architecture is refined by detailing through state machines the operation modes of the system elements at multiple granularities. Section 3.2 in Annex A is related to the gateway:

- Section 3.2.1 presents the gateway operation modes.
- Section 3.2.2.3 presents the operation modes of a partition running in the mixed-criticality gateway.
- Sections 3.2.3 and 3.2.4 provide a detailed view of the operation of the user, monitoring middleware and update middleware partitions.
- Section 3.2.4.1 presents a complete view of the gateway during an update showing the active states on each partition.

The operation modes of the end-device are provided in Section 3.3.3 of Annex A.

3.3 Risk assessment

The risk assessment, which is covered in Part III of Annex A, plays an important role in the safety and security management process, since it enables the identification and qualification of the safety failures, security threats and risks. This study can be then used as input for the specification of safety and security requirements. Section 4 presents the risk assessment itself, whereas Section 5 covers risk treatment.

Section 4.1 of Annex A starts by describing the combined safety and security methodology adopted for the risk assessment, which is based on safety and security standards such as IEC 61508, IEC 62443, ISO 27005 and aligned with the automotive cybersecurity standard ISO 21434. The first step in the assessment is the context establishment provided in Section 4.2 (in this case the automotive use case). Then, Section 4.3 of Annex A proceeds with the risk identification by determining the assets, the dependencies among them and the safety and security threats associated to them. The dimensions, in other words, the properties such as integrity or confidentiality, of the asset are also considered. Based on the built threat model, Section 4.4 of Annex A evaluates the potential impact on each of the affected dimensions and provides an estimation of the probability of attack or failure. From a safety perspective, these values are obtained by means of a Failure Modes, Effects and Criticality Analysis (FMECA) included as an Appendix to the SASE concept (Appendix B). Finally, Section 4.5 of Annex A evaluates the risk associated to each asset.

Then, the risk treatment phase addresses previously identified risks. Sections 5.1 and 5.2 of Annex A identify the multiple safety and security techniques and measures, which are latter described in Section 5.3. Safety countermeasures intent to reduce the residual risk to acceptable levels for the target integrity level and security countermeasures aim to reduce attack surface and the security risks. Among the different countermeasures, the virtual compatibility and integration checks defined within WP4 are also included for TÜV review. To this end, the contents of D3.1 (safety and security criteria for contracts) and D4.1 (UP2DATE Middleware Foundations) are summarized in Appendix C of Annex A. Finally, Section 5.4 of Annex A presents a list of possible systems reaction to the main errors in the gateway and end-device, including potential failures in the update process.

4 SECURE COMMUNICATIONS

This section describes an early design proposal for a communication procedure, data exchange and payload transportation scheme for OTASU in the UP2DATE architecture. The three main elements of the UP2DATE architecture (i.e., server, gateway and end-devices) belong to different security zones and are connected through communication channels that provide the security functions that enable secure communication. In each zone, only the corresponding communication protocols (depicted in Figure 2) should be authorized.

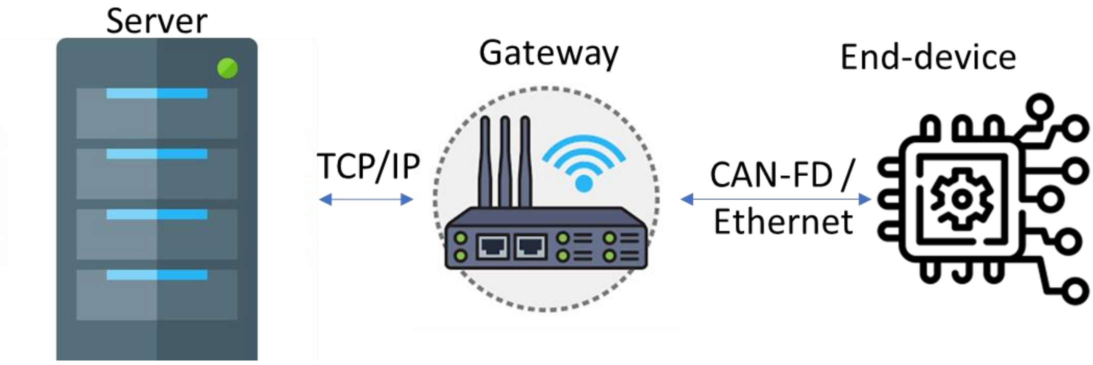

Figure 2: Communication protocols

On the one hand, there is the communication between the server and the gateway which is based on TCP/IP (Section 4.2) and on the other hand, the communication between the gateway and the end-devices which can be either CAN-FD based (automotive use-case) or Ethernet-based (railway use-case) (Section 4.3). Before describing the detailed solution, we specify the requirements for this secure communication in Section 4.1.

4.1 Secure Communication Requirements

Table 1 presents the secure communication requirements detailed from the high-level UP2DATE requirements of D3.2 (the source column specifies the identifier of related system requirement). These requirements are based on the following assumptions:

- An attacker (especially on the automotive use case where an OBD-II connector, providing access to the internal bus is usually installed in the device) could compromise the gateway.
- The end-device is a resource constraint platform, with limited computation capabilities. Therefore, resource intensive operations, including asymmetric cryptographic, are challenging. Symmetric cryptography is then used.
- Fieldbus communications provide low bandwidth and message payload.

Table 1: Secure communications Requirements[1]

Req. ID	Requirement	Source
Fnc-Req-0001	The server shall ensure authentication via an external authentication server	RACH_14
Fnc-Req-0002	The end device shall authenticate itself before any communication	RACH_14
Fnc-Req-0003	The gateway shall authenticate itself before any communication	RACH_14
Fnc-Req-0004	The server shall ensure that a valid authentication is only valid for a defined period of time	RACH_14
Fnc-Req-0005	The server shall provide an endpoint for authentication	RACH_14
Fnc-Req-0007	The server shall be able to communicate with the end device through the gateway	RUCY_14
Fnc-Req-0008	The communication between server and end device shall be encrypted and authenticated (i.e. symmetric KDF algorithm)	RUCY_14
Fnc-Req-0009	The communication between server and gateway shall be encrypted and authenticated	RUCY_14
Fnc-Req-0011	The gateway must not allow information or files been exchanged between the server and the end-device to be decoded or stored	RUCY_14
Fnc-Req-0014	The server shall provide endpoints for communication	RACH_14
Fnc-Req-0049	The server shall transmit a digital signature together with the update package.	RUCY_15
Fnc-Req-0050	The gateway/end-devices shall check update file integrity based on the digital signature to provide protection from corruption, malicious code installation and execution.	RUCY_15
Fnc-Req-0051	The gateway shall securely transmit monitoring information to the server, guaranteeing confidentiality.	RUCY_24
Fnc-Req-0052	The end-device shall securely transmit monitoring information to the gateway, guaranteeing confidentiality.	RUCY_23
Usa-Req-0001	The server shall inform the user of the gateway/end device about an available update	RUCY_07

4.2 Server-Gateway communication

In order to provide a basic security for the communication over a public network, every communication between gateway and server will be operated through a TLS-based VPN tunnel with device/gateway individual credentials. This will fulfil the requirements, that both client and server need to be authenticated and the communication needs to be encrypted.

Potential attack vector for computer systems are always given by externally exposed ports. To fulfil the requirement that the server shall be able to inform the user (and thus the gateway) about available updates, either a polling mechanism need to be implemented or the gateway needs to expose such a port. VPNs allow the gateway to provide the capability of accepting incoming connections only from within a trusted environment. Furthermore, as each VPN

[1] Please note, that only requirements relevant to this chapter are listed here.

endpoint can be containerized, it can be assured that compromised devices cannot be used to attack other devices, as they will be imprisoned in the corresponding container.

4.2.1 Security Certificates

One main problem in electronics computing communication is the impossibility to verify that the sender of the communication is really who it claims to be. To draw an analogy with person identification, documents such as the passport are used accompanied by a signature or a photograph that identifies to the person. Certificates are used to imitate this process using digital methods. Therefore, a digital certificate is a digital document that certifies that the public key that it contains belongs to the entity (person, device, computer…) who identifies. This is issued by a Certification Authority (CA) and guarantees that this CA has verified and trusts the identity of the entity to whom the certificate belongs.

To allow the use of public keys on a network such as the Internet, a valid and reliable key distribution infrastructure is need. A Public Key Infrastructure (PKI) allows a company to have electronic authentication systems (who is it and what is it?), with confidentiality, data integrity and non-repudiation for their network applications, using advanced technology, such as digital signatures, cryptography and digital certificates.

A public key infrastructure relies on digital signature technology, which uses public key cryptography. The basic idea is that the secret key of each entity is only known by that entity and is used for signing. This key is called the private key. There is another key derived from it, called the public key, which is used for verifying signatures but cannot be used to sign. This public key is made available to anyone and is typically included in the certificate document. PKI provides "trust services" in plain terms trusting the actions or outputs of entities, be they people or computers. Trust service objectives respect one or more of the following capabilities: Confidentiality, Integrity and Authenticity (CIA).

- Confidentiality: The privacy of user transactions is protected by encrypting data streams and messages. The confidentiality function may be intended to prevent the unauthorized disclosure of information locally or across a network. By using PKI, users are able to ensure that only an intended recipient can “unlock” (decrypt) an encrypted message.
- Integrity: Guaranteeing message integrity is another important function of PKI. PKI has built-in ways to validate that all the outputs are equivalent to the inputs. Any alter of the data can be immediately detected and prevented. Often it is not of high importance to prevent the integrity being compromised (tamper proof), however, it is very important that, if integrity is compromised, there is clear evidence of it having done so (tamper evident).
- Authenticity: Authentication is the process of verifying that the users are who they say they are. PKI provides a means for senders and recipients to validate each other's identities.
- Non-Repudiation: PKI ensures that an author cannot refute that they signed or encrypted a particular message once it has been sent, assuming the private key is secured. Here digital signatures link senders to their messages. Only the sender of

the message could sign messages with their private key and therefore, all messages signed with the sender's private key originated from that specific individual.

The operations carried out by a PKI can be very diverse, but the most important are:

- Issue, renew and revoke a certificate.
- Create, review, and revoke a key.
- Check the status of a certificate.
- Carry out management and administration operations.
- Store certificates in a reminder list.
- Publication of keys: once the keys are created, the PKI allows the dissemination of the public keys, as well as locating the public keys of other users, along with their status (key revoked).

One of the advantages of a PKI is that it does not depend on any specific technology but provides a framework of action on which to work, according to the needs in each case. To create the certificates, a PKI-based trust chain should be created. Figure 3 depicts a PKI and certificate structure proposal. The root_CA is the root Certificate Authority (CA), which due to its criticality, is usually handled by an external cyber-security services hosting enterprise.

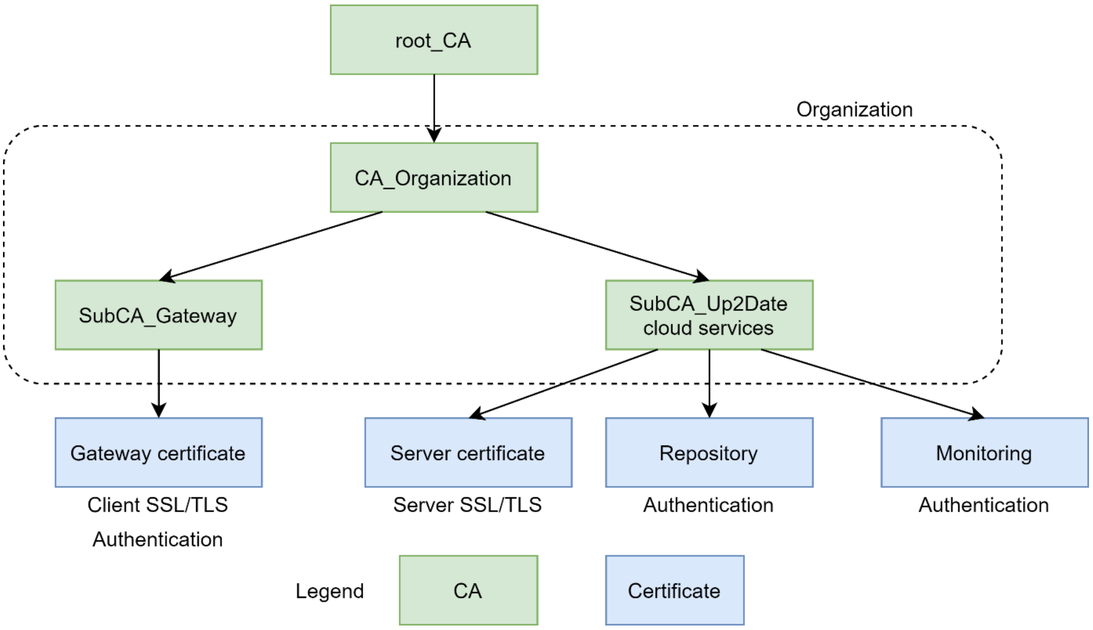

Figure 3: Public Key Infrastructure and certificate structure proposal

Although it might change depending on the needs of each organization, an organization-level CA (CA_Organization) will be created. In this way, each company will be independent from the others to control and maintain its PKI and related certificate structure. Afterwards, different Sub_CAs are commonly created, focused on the functionality and the type of the certificates required. The proposed structure is flexible enough to grow as needed to fulfil actual and future needs as well as it is adaptable to the requirements.

There are numerous options available for ready-made certificate management systems to work. Many of them are open source like “Lamassu IoT”, others only have a paid version such as “Windows Server”, “Digicert” or “Entrust”, and some have a basic free version like “EJBCA” or “Vault” that can be upgraded with a payment version (which usually includes support). Therefore, as there are already stablished solutions to create certificates, in the proof of concept of UP2DATE we create certificates by means of the current state of the art tools, for example, through an external PKI services, as there is no added value on creating our own PKI for the project.

4.3 End-device communication

Based on the performed safety and security risk assessment of Annex A (Part III), security risks associated to the integrity (I) and authentication of service users (A_S) in the end-device communications (assets ED.COM.01) shall be addressed. Thus, a security scheme addressing those security issues should be implemented. Both the update package (transmitted from the server to the end-device) and the monitoring data (transmitted from the end-device to the gateway and server) need to be protected.

4.3.1 Update package communication protocol

The Update protocol for Automotive end-device is based on standard UDS (ISO 14229) with the addition of a set of diagnostic services for the scope of achieving the required countermeasures identified by the risk analysis. It is necessary to implement a good compromise between the complexity of the countermeasures and the limited computational resources of an automotive microcontroller with the objective of reuse on simpler architectures (e.g., smart sensor/actuator). For the railway use case, the same security solution will be adopted, although adapted to the railway use case communication protocols and technology, instead of the automotive UDS specifications.

The update package sent during OTASU is authenticated (NIST CMAC SP 800-38B) and then encrypted by usage of symmetric cryptography (NIST AES-128 FIPS-197) taking benefit of the hardware acceleration provided by AURIX internal Hardware Security Module (HSM). On every end-device session, the keys used for CMAC generation/verification and AES encryption/decryption are different, this is a countermeasure for key discovering attacks. For this countermeasure it is important that the derived keys are the same between the server and the end-device so it is necessary a synchronization before the switch to the new keys for the next OTASU session. In the next description the gateway acts as a router between server and end-device and, if omitted, the sequence of information is always as that depicted in Figure 2.

The process is as follows:

- The server and the end-device generate temporal symmetric keys by means of a symmetric key derivation function (*K_authentication* and *K_encryption*). The Label of the key is used to distinguish the scope of the derived key.

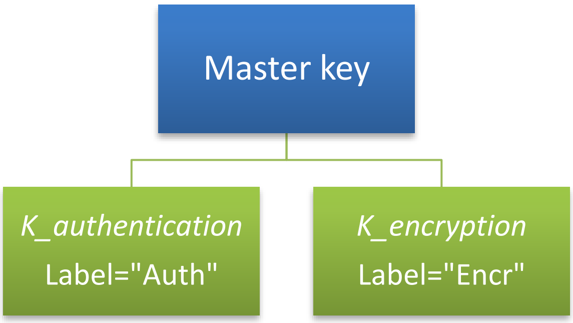

Figure 4: OTASU Key hierarchy

- The counter (KDF_CNT) used for key derivation is sent by the server at the beginning of OTASU session and the end-device provides the acknowledge if the server KDF_CNT matches the internal one.
- If the end-device acknowledges the KDF_CNT, then the server:
  - derives and creates *K_authentication* and *K_encryption* using the new KDF_CNT value
  - creates an update authentication code (a CMAC) (using *K_authentication)*
  - encrypts the update package (using *K_encryption*)
- The update package is sent to the gateway that performs UDS/railway download sequence to the end-device.
- The end-device then:
  - derives and creates *K_authentication* and K_encryption using the new KDF_CNT value sent previously by the server
  - decrypts the update package (using *K_encryption*)
  - verifies the authenticity of the update (using *K_authentication*).
  - Allows the switch to the new updated software update only if authentication is successful
- Once the software update is accomplished, the gateway performs the acknowledge routine request to end-device that:
  - creates the acknowledge message and append the CMAC signature (using *K_authentication*).
  - encrypts the ACK+CMAC message (using *K_encryption*)
- The server sends the ACK to the end-device (using the same session keys) but with new incremented counter so
  - creates the acknowledge message for incremented counter and append the CMAC signature (using *K_authentication*).
  - encrypts the ACK+CMAC message (using *K_encryption*)

- Both the server and the end-device increment their counters ready for the next OTASU

The diagram in Figure 5 shows the OTASU for Automotive end-device process. Note that the same mechanisms will be used for railway except the UDS download sequence.

Since the update package is encrypted and authenticated, an intruder that has compromised the gateway, cannot tamper the update package.

It is assumed that at the initial production of the system (normal during factory production) the production process ICT infrastructure guarantees that the server and the end-device will share the same AES master secret key (master secret key injection and registration). It must also be mentioned that the master secret key should be stored on a secure storage. An attacker should be not able to access such secret. To this end, additional security measures should be implemented in the end-device. Most likely, temporal key generation tasks should or might be accomplished by a secure Root-of-Trust (RoT) element.

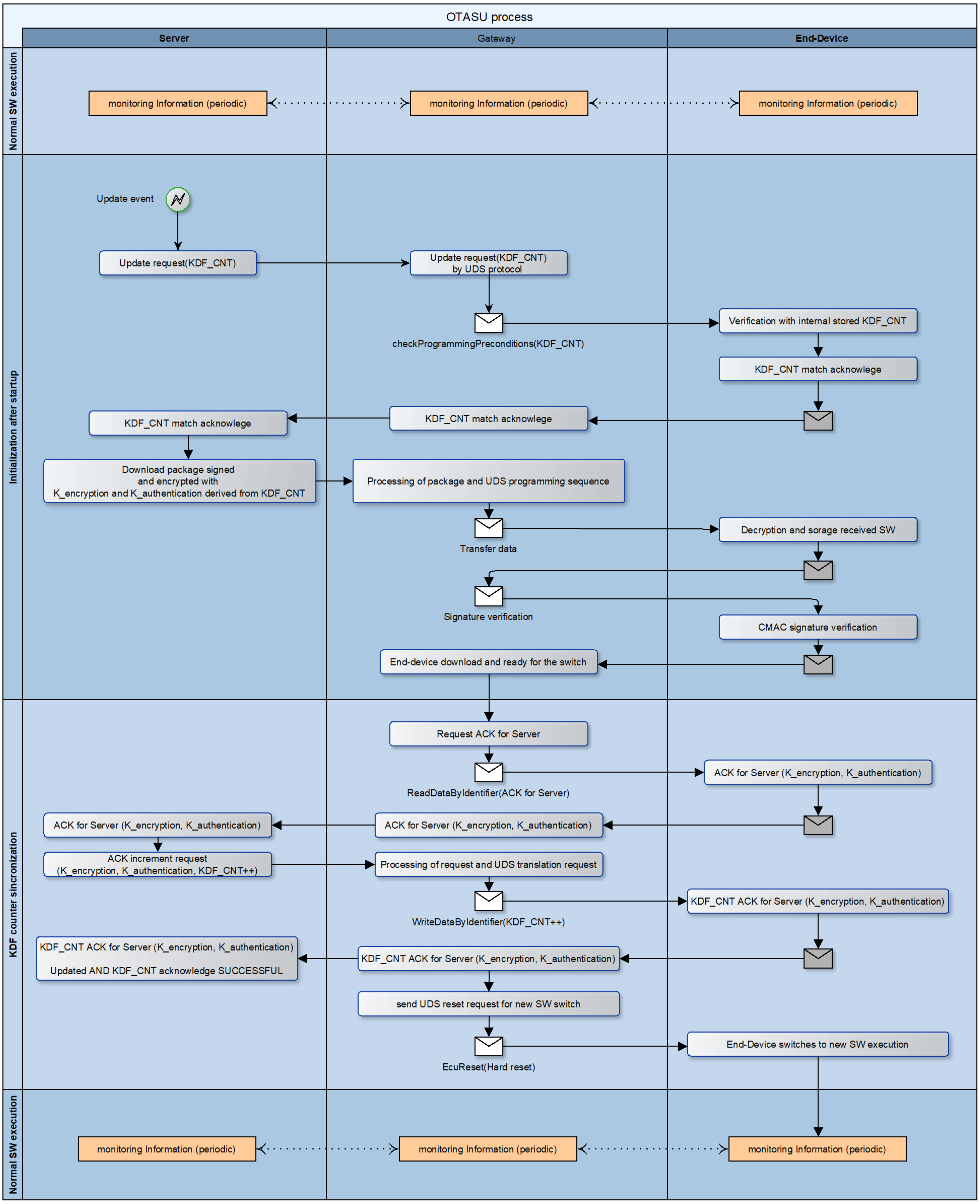
OTASU process
Server
Gateway
End-Device
Normal SW execution
monitoring Information (periodic)
monitoring Information (periodic)
monitoring Information (periodic)
Initialization after startup
Update event
Update request(KDF_CNT)
Update request(KDF_CNT) by UDS protocol
checkProgrammingPreconditions(KDF_CNT)
Verification with internal stored KDF_CNT
KDF_CNT match acknowlege
KDF_CNT match acknowlege
KDF_CNT match acknowlege
Download package signed and encrypted with K_encryption and K_authentication derived from KDF_CNT
Processing of package and UDS programming sequence
Transfer data
Decryption and sorage received SW
Signature verification
CMAC signature verification
End-device download and ready for the switch
KDF counter sincronization
Request ACK for Server
ReadDataByIdentifier(ACK for Server)
ACK for Server (K_encryption, K_authentication)
ACK for Server (K_encryption, K_authentication)
ACK for Server (K_encryption, K_authentication)
ACK increment request (K_encryption, K_authentication, KDF_CNT++)
Processing of request and UDS translation request
WriteDataByIdentifier(KDF_CNT++)
KDF_CNT ACK for Server (K_encryption, K_authentication)
KDF_CNT ACK for Server (K_encryption, K_authentication)
KDF_CNT ACK for Server (K_encryption, K_authentication)
Updated AND KDF_CNT acknowledge SUCCESSFUL
send UDS reset request for new SW switch
EcuReset(Hard reset)
End-Device switches to new SW execution
Normal SW execution
monitoring Information (periodic)
monitoring Information (periodic)
monitoring Information (periodic)

Figure 5: OTASU for Automotive end-device process

4.3.2 Monitoring data communication protocol

An OTASU similar approach is used for the authentication of periodic monitoring messages. In this case the encryption is not used in order to not overload the microcontroller with encryption/decryption tasks. As the automotive CAN-FD or railway signalling periodic messages communication requires prompt reaction of the system for the real-time management the same scheme and algorithms from update protocol are used instead of more complex solution like AUTOSAR Secure Onboard Communication.

It is assumed that at the initial production of the system (normally during factory production) the industrialization process including ICT infrastructure guarantees that the server, the gateway and the end-device will share the same AES master secret key (master secret key injection and registration). Several alternatives can be possible depending on the OEM ICT technology and proper infrastructure.

As this part is out of scope of UP2DATE project it is assumed the following use case:

- The AES master secret key is created for each vehicle class part number and it is required that all the ECUs in the vehicle are equipped with this key during the ECU supplier production. During the vehicle manufacturing process, after the vehicle personalization (immobilizer configuration, VIN registration, vehicle optional configuration for the customer) the production process performs a request to the gateway that will initiate the internal master key update change specific for that vehicle. The new AES master key could be derived from the information of the vehicle personalization plus initial master key. This part is out of UP2DATE project scope, so it is assumed that the gateway and end-device have the same master key unique for each vehicle. In the railway use case, a AES master secret key will be created for each controller.

After this phase (out of UP2DATE scope) the effective AES keys for authentication and encryption of the first power on CAN-FD or Ethernet message is derived from AES master secret keys with KDF_CNT equal to zero (*K0_authentication, K0_encryption).*

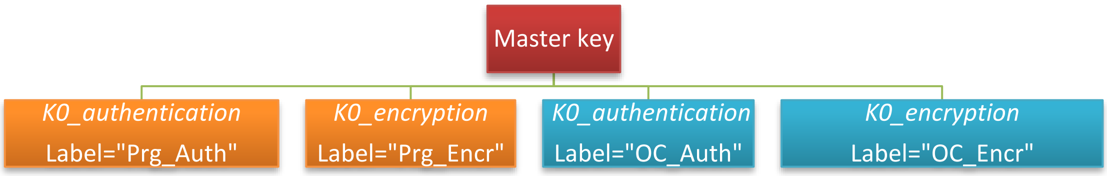

Figure 6: Key hierarchy at power on

At each power on, the gateway issues a first event CAN-FD or ethernet message with a true random number as KDF_CNT encrypted with *K_encryption* with KDF_CNT equal to zero

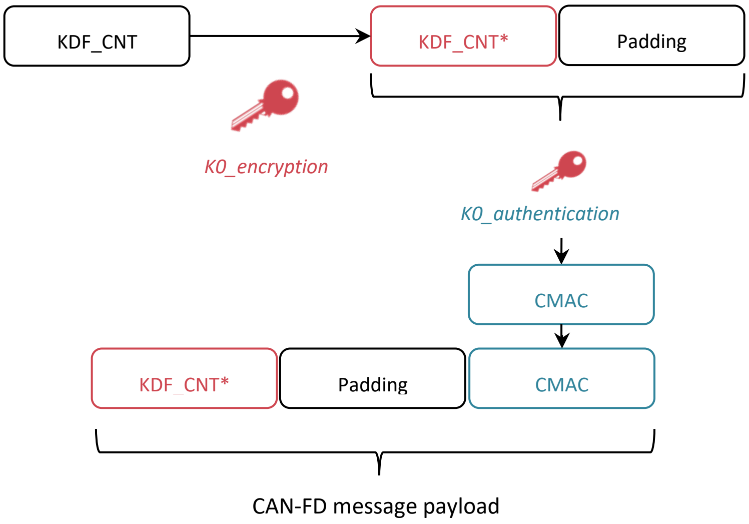

Figure 7: CAN-FD message payload with encrypted KDF_CNT

(*K0_encryption)* plus a CMAC calculated with *K_authentication* with KDF_CNT equal to zero (*K0_authentication).*

The end-device checks the CMAC of the received message with the same *K0_authentication* and if the encrypted KDF_CNT including padding is authentic then it performs the decryption of KDF_CNT by *K0_encryption* generated key as shown in Figure 7 for the Automotive use case.

After this phase the end-device derives:

- the K_authentication for the subsequent communication using the KDF_CNT (K_{TRN}_authentication)
- the K_encryption for acknowledge message encryption (K_{TRN}_encryption)

and sends the acknowledge event message padded, encrypted (*K_{TRN}_encryption)* and then signed by CMAC using the new derived keys (*K_{TRN}_authentication).*

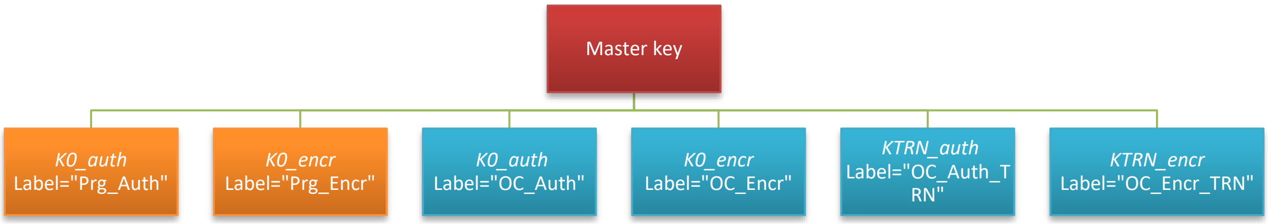

Figure 8: Key hierarchy during normal monitoring communication

The gateway performs the CMAC verification (*K_{TRN}_authentication)* followed by decryption (*K_{TRN}_encryption)* and from this moment the gateway and end-device establish the *K_{TRN}_authentication* as CMAC verification key for all subsequent monitoring messages output from the end-device. The end-device sets the new key only at the confirmation of event message. A summary of the process is reported in the following picture of Figure 9 for the Automotive use case. The same process would apply for Railway.

Since the monitoring information is authenticated, the gateway can distinguish if the received message is sent by the end-device (legitimate) or is sent by an attacker. It is assumed that the monitoring data contains a rolling counter as anti-replay attack countermeasure.

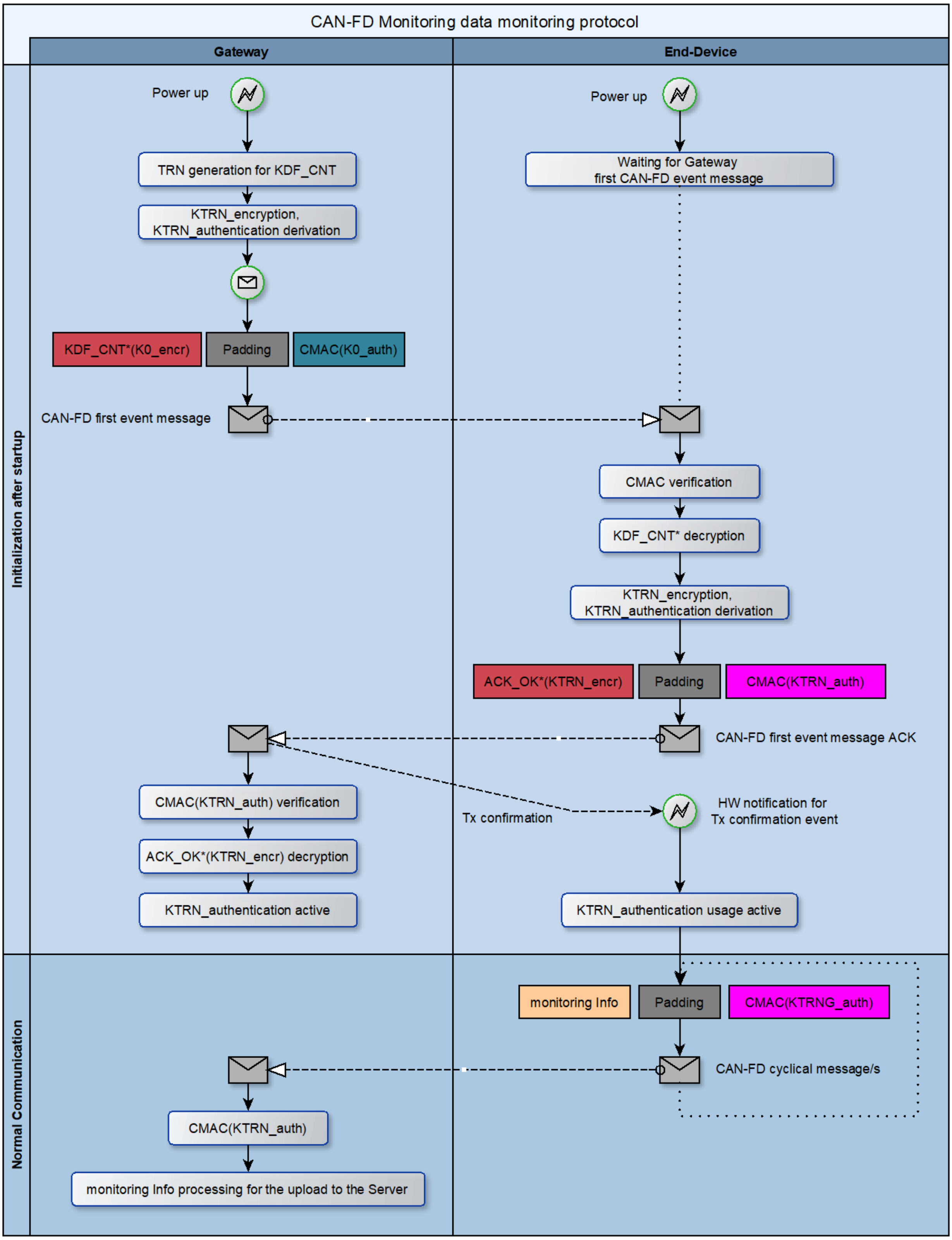
CAN-FD Monitoring data monitoring protocol
Gateway
End-Device
Initialization after startup
Normal Communication
Power up
TRN generation for KDF_CNT
KTRN_encryption,
KTRN_authentication derivation
KDF_CNT*(K0_encr)
Padding
CMAC(K0_auth)
CAN-FD first event message
Power up
Waiting for Gateway
first CAN-FD event message
CMAC verification
KDF_CNT* decryption
KTRN_encryption,
KTRN_authentication derivation
ACK_OK*(KTRN_encr)
Padding
CMAC(KTRN_auth)
CAN-FD first event message ACK
CMAC(KTRN_auth) verification
Tx confirmation
HW notification for
Tx confirmation event
ACK_OK*(KTRN_encr) decryption
KTRN_authentication active
KTRN_authentication usage active
monitoring Info
Padding
CMAC(KTRNG_auth)
CAN-FD cyclical message/s
CMAC(KTRN_auth)
monitoring Info processing for the upload to the Server

Figure 9: Onboard Communication protocol

5 CONCLUSIONS

This deliverable summarizes the progress of WP3 from M18 to M21, giving an update over the work reported in previous deliverable D3.2. The requirement specification and architecture definition of the project have been complemented with a risk assessment and countermeasure definition to build a safety-security concept. This safety-security concept has been reviewed by TÜV Rheinland, a certification authority external to the project. The early evaluation of the SASE concept allows detecting implausibilities and weak point in the design of the architecture, to take them into account in future redesigns. In addition, even though the complete certification of the UP2DATE architecture is not one of the project objectives due to its TRL, the assessment of the SASE concept provides evidence and a clear route for the future certification of the concepts defined in the project. An important outcome of the review meeting was that, according to TÜV Rheindland, from the logical point of view the safety-security concept covers the fundamentals of safety and security requirements and it shows a clear path to continue working on, with very solid foundations for an European project. On the next phase, the WP will work on addressing the review comments of TÜV Rheinland and the resulting final SASE concept will be included in next version of the deliverable (D3.4).

The report has also exposed the initial design for the secure communications, showing two approaches for the server/gateway communication and for the server/end-device communication. This early design will be refined in the next months when its implementation will start.

6 REFERENCES

[1] International Electrotechnical Commission, *IEC TR 63069: Industrial-process measurement, control and automation – Framework for functional safety and security, technical report*, 1.0 ed. 2019.

[2] T. Rheinland, "IKERLAN Functional Safety Management Certificate (No. 968/FSM 138.02/19). IEC 61508 Parts 1-7:2010 - E/E/PE System Realisation (Phase 10)FSM 138, Maturity Level 2 - Managed," ed, 2019.

[3] International Electrotechnical Commission, "IEC 61508 Functional safety of Electrical/Electronic/Programmable Electronic safety-related systems (Second edition)," *International Electrotechnical Commission, Geneva, Switzerland,* 2010.

ANNEX A SAFETY AND SECURITY CONCEPT FOR SOFTWARE UPDATES ON MIXED-CRITICALITY SYSTEMS

[For the moment this is included in a separate file. Refer to D3.3_Annex_InitialSASEConcept.docx file in /WPs/WP3/Deliverables/D3.3/D3.3_Annexes folder.]

ANNEX B REVIEW MEETING PRESENTATIONS AND LIST OF OPEN POINTS

- Review meeting slides [TBD] [will be included after the meeting]