

# Software Updates Over-The-Air II

## A process and risk analysis by mean of simulation

project number	9810009
authors	Jonas Römer (AQI) Dr. Jürgen Großmann (Fraunhofer FOKUS)
contact us	<a href="mailto:kontakt@aqigmbh.de">kontakt@aqigmbh.de</a>
date	07.09.2019

This publication is based on the expert knowledge from the AQI and scientific partners. It represents a consolidated position on the relevant issues.

The work including all its parts is protected by copyright. Any use that is not expressly permitted by copyright law requires prior consent.

# Table of contents

<b>1</b>	<b>MOTIVATION AND OBJECTIVES .....</b>	<b>4</b>
<b>2</b>	<b>EVALUATION AND REVISION OF THE CLASSIFICATION SCHEMES .....</b>	<b>5</b>
2.1	Original classification scheme .....	5
2.2	Problem.....	6
2.3	Evaluation method .....	7
2.4	Result of the evaluation .....	7
2.5	Alternative classification along attributes .....	8
2.6	Conclusion.....	10
<b>3</b>	<b>IDEALIZED OTA-PROCESS .....</b>	<b>11</b>
3.1	Process boundaries.....	11
3.2	Descriptive means for process representation.....	12
3.2.1	Modeling with BPMN 2.0 .....	12
3.2.2	Text description .....	14
3.2.3	Roles in the process.....	16
3.3	UN ECE Task Force on OTA Updates .....	17
3.4	Process contents .....	20
3.5	Update generation.....	21
	OTA-G-051 [new].....	21
	OTA-G-020.....	23
	OTA-G-030 [reg. auth. only] .....	24
	OTA-G-031 [reg. auth. only] .....	24
	OTA-G-090.....	25
	OTA-G-070 [optional] .....	26
	OTA-G-060.....	27
	OTA-G-071 [optional] [new].....	28
	OTA-G-110.....	28
	OTA-G-120 [reg. auth. only] .....	29
	OTA-G-130.....	29
	OTA-G-141 [new].....	30
3.6	Update distribution.....	32
	OTA-V-020.....	33
	OTA-V-030.....	34
	OTA-V-040 [optional] [reg. auth. only] .....	34
	OTA-V-050.....	35
	OTA-V-060.....	36
	OTA-V-081 [new].....	38
	OTA-V-090.....	39
	OTA-V-100.....	40
	OTA-V-120.....	41
	OTA-V-140.....	41
	OTA-V-150 [new].....	43
3.7	Update installation .....	45
	OTA-I-020.....	46
	OTA-I-010.....	46
	OTA-I-011 [new].....	48

---

OTA-I-030.....	48
OTA-I-040.....	49
OTA-I-050.....	50
OTA-I-060.....	50
OTA-I-070.....	51
OTA-I-080.....	51
OTA-I-090.....	52
OTA-I-100 [reg. auth. only].....	52
<b>4 SIMULATION .....</b>	<b>54</b>
4.1 MATLAB tool.....	54
4.2 Use Case comparison.....	55
4.2.1 Car-Security-Incident-Response (Car-SIR) .....	56
4.2.2 Updating the software in infotainment .....	58
4.2.3 Recall of software in the field on KBA instruction .....	60
4.2.4 Conclusion.....	62
4.3 Process optimization.....	63
4.3.1 Static influencing factors .....	63
4.3.2 Dynamic influencing factors .....	63
4.3.3 Conclusion.....	70
<b>5 PROJECT CLOSURE .....</b>	<b>71</b>
Discontinued activities in relation to project OTA-1 .....	72
OTA-G-010 [not applicable].....	72
OTA-G-040 [not applicable].....	72
OTA-G-050 [not applicable].....	73
OTA-G-080 [not applicable].....	73
OTA-G-100 [not applicable].....	73
OTA-G-140 [not applicable].....	74
OTA-V-070 [not applicable].....	74
OTA-V-010 [not applicable].....	74
OTA-V-110 [not applicable].....	74
OTA-V-130 [not applicable].....	75

## Table of figures

Figure 1: Classification scheme.....	5
Figure 2: Classification using attributes.....	9
Figure 3: Legend of the used BPMN 2.0 .....	13
Figure 4: Definition of the Process-ID.....	14
Figure 5: Example of the representation of process variations .....	14
Figure 6: Scope of Works of the UN ECE TF on CS/OTA .....	17
Figure 7: Contents of the Recommendation on Software Updates .....	18
Figure 8: Schema of the update process .....	20
Figure 9: Process modeling "generation".....	21
Figure 10: Process modeling "distribution".....	32
Figure 11: Process modeling "installation" .....	45
Figure 12: Logical structure of the simulation tool.....	54
Figure 13: Use Case Car-SIR - lead time per vehicle .....	57
Figure 14: Use Case Car-SIR - lead time per activity .....	57
Figure 15: Use Case Infotainment - lead time per vehicle .....	59
Figure 16: Use Case Infotainment – lead time per activity.....	60
Figure 17: Use Case Airbag - lead time per vehicle.....	61
Figure 18: Use Case Infotainment - lead time per activity.....	62
Figure 19: Evaluation of the number of server slots / number of vehicles .....	64
Figure 20: Evaluation of the distribution of the transmission technology .....	65
Figure 21: Comparison of both communication methods.....	67
Figure 22: Influence of the request after confirmation from the customer .....	68
Figure 23: Influence of the probability that an update will be accepted .....	69

## table directory

Table 1: Example of a text description of process activities.....	16
Table 2: Summary of process optimization .....	70

# 1 MOTIVATION AND OBJECTIVES

The aim of this follow-up project is to critically question the results of the project "Software Updates Over-the-Air 1" and to extend them by current developments. The first project consisted mainly of researching, linking and structuring partial aspects in order to design an idealized, holistic OTA update process. The follow-up project is methodically based primarily on the programming of an own simulation tool. To develop this tool, each individual activity had to be analyzed in detail and questioned in order to develop a suitable model. In the conclusion, the current status of international standardization procedures and committee work was examined.

The result is a complete update of the process description from the first report. The three phases generation, distribution and installation describe 12, 11 and 11 activities respectively. Of these 34 activities, 6 are new, with 10 activities from the previous project being dropped because they were replaced by other activities, merged into others, or did not correspond to the chosen level of abstraction. Each of these activities is described and supplemented with various attributes to provide a structured collection of knowledge about the OTA process chain.

The simulation was then used to compare different use cases with each other, perform a critical path analysis and iteratively optimize one of these use cases in order to investigate the influence of different process parameters on the throughput time of the process. Before the process description is presented, the discussion of the classification approaches proposed in the previous project follows.

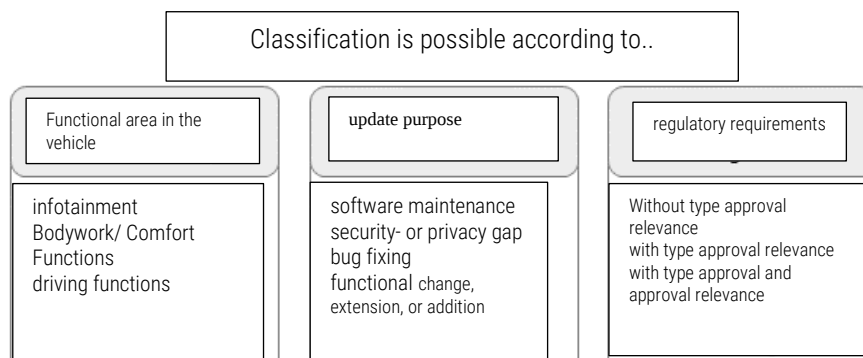
The aim of the project series is to create over-the-air transparency through structuring and to establish process capability for the topic of software updates. Because OTA updates are not purely technical, but complex business processes.

## 2 EVALUATION AND REVISION OF THE CLASSIFICATION SCHEMES

A systematic classification of software updates along the functional area in the vehicle, the purpose of the update or the regulatory requirements enables the quality, robustness and efficiency of the OTA update process to be increased. Depending on the respective update class, the requirements for the process modules can be adapted or varied. In addition, (minimum) standards and subtasks and processes can be stored for the various classes. This creates a degree of freedom for the company, through which the effort and process duration can be adapted to the specific needs of the classes. In addition, a classification can also be used to prioritize the distribution of updates or to qualify providers of OTA-relevant service providers.

### 2.1 Original classification scheme

In the first OTA project, three dimensions were proposed for a possible classification of OTA updates and their influence on an OTA update was explained. The dimensions shown in the Figure 1 represent the original classification scheme.



*Figure 1: Classification scheme*

Updates can first be differentiated according to the functional area they address. On the one hand, this reflects the organizational structures in the automotive industry and, on the other hand,

considers the mechanical and software differences between infotainment, body/comfort and driving functions. The technical delimitation between updates also reflects the possible risk to occupants and/or the environment that could arise as a result of a failure of the function.

A further classification can be made according to the purpose of the update. Analogous to the updates in the IT sector, a distinction is made here between preventive, corrective and additive software updates. In addition to the varying information from customers and regulatory authorities, the need for an update to be urgent can also be derived from such a classification.

The last dimension only considers the interface to the regulatory authority and thus the type approval and/or approval relevance. In the case of such a dependency, additional requirements must be addressed to the update process.

When each of these three classifications is applied, process variations can already arise that offer added value when an update is carried out. Within a classification an urgency of the update can be derived, the need for information can be described to the customer or also a gradation of the extent of the validation of the functionality after the installation.

However, the latter also applies to two classifications: the update target and the functional area. For example, for software maintenance or infotainment updates, a check of the complete installation routine is sufficient, while for troubleshooting and driving functions, a check of the function itself should take place after the installation.

Even a classification of software updates according to one of these three classifications creates added value for the processing of the OTA update process.

## 2.2 Problem

However, the classification of updates according to only one of these classifications does not take into account all the requirements of the individual process activities. While a classification offers a gradation for one requirement, this requirement does not have to be considered for another classification. The classification along one dimension is therefore not sufficient to consider all necessary process requirements. Due to the independence of the classification dimensions, potentials for process optimization are lost.

For example, an update can be an infotainment troubleshooting or a driving function troubleshooting. The latter troubleshooting may be relevant to product safety, which is why (a) it should be carried out more quickly, i.e. more time-critically, and (b) it may be required to be reported. If these requirements were also applied to infotainment troubleshooting (e.g. non-functioning GUI elements) based on a classification according to the update purpose, potentials in terms of throughput time and costs would not be exploited here, because the highest



requirements would always have to be exploited through possible safety relevance. Only by linking the two classifications could the potentials be exploited. In addition, there may be infotainment error corrections (e.g. missing information display on the vehicle status) that still have time-critical safety relevance. This would be unambiguously depictable neither with the independent nor with linked dimensions. Rather, at least the third dimension would have to be added, since the update might be subject to notification and the authorities' view would thus become relevant.

If all possible classifications are considered completely and logically, twelve or more process variations could follow. Classification would no longer be meaningful, as there are too many variations. Therefore, this section examines whether there are similarities between the many possible classification combinations, so that the number can be broken down to a few process variations. The aim is a harmonized classification with a few classes.

## 2.3 Evaluation method

In order to identify superfluous classes and to harmonize the entire classification scheme, the classification scheme was analyzed and validated by applying it to the entire process chain as developed in the first OTA project. The individual activities were systematically discussed in relation to the class. It was analyzed whether new requirements arise for the individual activities by the classes. If there were the same requirements for several classes for a defined set of activities, this would be an indication to combine these classes.

## 2.4 Result of the evaluation

At first it can be stated that especially in the first process phase "generation" the process activities show no differences from the point of view of the entire classification scheme. The requirements, which would change depending on a class, are aimed at established processes, such as the development of updates, but not at the process of distributing an over-the-air update itself.

It can also be seen that the interface with the regulatory authority (e.g. in the case of recalls) only means supplementary activities. Other activities of the OTA process chain are not varied by this interface.

For the other activities, there are no combinations that would harmonize the classifications and reduce them to a small number. The combinations are multiple and not unique. An assignment to

a new classification would ultimately only represent a compromise. As a result, the desired increase in efficiency could not be achieved and the necessary quality assurance is not sufficiently considered because, for example, the required key figures could not be defined precisely enough. The desired reduction of complexity (understanding, execution, etc.) is not achievable.

## 2.5 Alternative classification along attributes

It makes much more sense to assign properties to the update, which are used to define further requirements for activities. In addition to "simple" updates, the following attributes can be assigned to software updates:

- **Time critical**
- **Safety critical**
  - **safety-relevant**
  - **Security-relevant**
- **Authority-relevant**
- **Mandatory or Optional**

**None, one or more attributes can be assigned to an update.** Their effects and examples are described in more detail below.

The attribute authority-relevant allows you to differentiate when the regulatory authority must be integrated. This would be the case for a defective product posing a hazard. This probably affects many updates that also apply to the safety-relevant attribute. However, this attribute can also be used for additive updates (i.e. function extensions) that affect the driving function and where errors in the installation routine could possibly have safety-critical effects that must be adequately safeguarded in advance. The attributes always also refer to the update target - for example, a safety-relevant ECU. Otherwise every update would be security relevant, because the transmission path has to be protected. In this report, however, such an update is to be understood as an update of the security mechanisms in the vehicle (see below).

This type of 'classification' of updates by attributes creates the necessary freedom in the evaluation of updates, so that, for example, two different updates of body functions can be evaluated differently if it is one time a bug fix and the other time a function improvement. The update activities described below can thus be supplemented with conditions that are assigned to the attributes and that are to be executed during corresponding updates. This allows the process to be optimized for the different update types with their specific boundary conditions.

Depending on the company policy, for example, the time-critical update can be downloaded to the customer's vehicle as soon as it is activated so that it can be installed directly. In the case of updates that are not time-critical, the download only takes place with the customer's consent if the vehicle is for example in a WiFi. This is ultimately a cost factor, but creates added value for the customer, since the time for the update consists only of the installation time and no longer of download time and installation time. This can increase the acceptance for the prompt implementation of such an update.

	time-sensitive	Security relevant	safety relevant	authorities	obligatory
update of an infotainment app or software maintenance					
hardening of the security system		X			
close a security gap e.g. with acute threat of personal data stored in the	X	X			
Activation (on demand) of a driving function or a motor program			X		
correcting a fault in the airbag actuation	X		X	X	X
deposit new legal framework conditions regarding automated driving assistance systems				X	
improve control of exhaust gas post-processing	X			X	

**Figure 2: Classification using attributes**

In order to illustrate how the respective attributes are intended towards the update goal, the update of an infotainment app or software maintenance should be considered as an example of such a "simple" update. For the use of navigation maps, it is not necessary to update them immediately after the publication of an update. For example, it cannot be classified as security, safety or authority-relevant due to a lack of security risks when updating navigation data. Although an update can influence the driving experience and thus be relevant for the customer and manufacturer, it can still not be assigned an attribute.

The hardening of the security system as a preventive measure is not time-critical or imminently relevant for the safety of road users and therefore does not have the attribute of safety or authority-relevant. As protection against external attacks, however, it is very much security relevant.

The closing of a security gap, on the other hand, can occur promptly after an attack on systems that contain personal data or compliance-relevant data in order to prevent further data collection. As such an attack is security-critical but does not represent a concern for the regulatory authority, an update to close such a security gap only gets the attribute security relevant. The mandatory attribute results from various contexts, but in that case, there are no legal obligations or a shift in liability claim that would force this attribute.

The commissioning of a drive function or a new motor program can lead to a hazard if the specific update or the software contained is faulty. Therefore, such an update is safety-relevant. Since it is not about a reactive closing of security gaps on the software side or something similar, the attribute security relevant does not result. Manufacturers and end users can decide in different forms whether such an update should be installed at all (On Demand), accordingly the attributes are time-critical, authority-critical, and obligatory not appropriate.

The update to correct an error in the airbag control, on the other hand, requires marking with all attributes. Incorrect ignition of airbags poses a hazard and is therefore relevant to safety and the authorities. Correspondingly, the elimination of this error is also time-critical and mandatory.

Updates of this kind are particularly relevant to the authorities, as they, for example, deposit new legal framework conditions regarding automated driver assistance on the vehicle. In addition to authority-relevant, the attribute then applies time-critically in cases of updates to improve the control of exhaust gas post-processing.

## 2.6 Conclusion

This type of classification / description of updates along attributes is considered in the following chapters and the process notation is supplemented accordingly.

The original classification scheme, on the other hand, was not suitable for making the OTA process chain more efficient in the long term by grading the requirements for the activities of the process according to different classes. The three dimensions presented were too independent of each other to select one dimension. A revision of the classification scheme was also unsuccessful as it was not possible to harmonize the three dimensions on a common basis.

## 3 IDEALIZED OTA-PROCESS

In this section, an idealized OTA update process is described in detail. The description is based on a state analysis using scientific research, individual interviews with representatives of the automotive industry and a workshop at the AQI with nominated experts of the VDA members as part of the first project. The presentation of the idealized process should serve as a uniform basis for discussion, joint communication and further analysis. First, the limits of the process to be specified are defined before the selected process representation is described and the process along its activities is described in detail.

A high degree of abstraction was chosen to describe this process in order to avoid the description of technical and organizational details. The aim was to build a common understanding between all participants. Each activity described can be deepened and specified by further activities. With increasing precision, general validity can no longer be given, since the differences between the various processes of the member companies involved are no longer taken into account. These details and differences are also due to the competition in the industry of the still 'young' technology Over-The-Air Updates. (Technical specifications can be referred to within this document. From the point of view of competition, however, there is no need to describe or define requirements in this respect.)

### 3.1 Process boundaries

The process described here primarily involves the distribution of software as an OTA update to the vehicles of a vehicle manufacturer. The description does not include the identification of the need for an update or the production and provision of the software update.

The beginning of the process described here is a recognized software update requirement. The detailed process of detecting errors or a need for software updates is not discussed in this report. For the triggering (*start*) of the OTA process described here, it is only required that such a software update requirement has been determined. The identification of such a need can be done fundamentally through three different channels:

- (1) from the troubleshooting process

- (2) through strategic product development
- (3) by customer feedback or customer request, e.g. for a function in the vehicle<sup>1</sup>

The necessary engineering of the update package itself is also an established, highly competition-relevant process, to which this report only refers in order to be able to supplement it with OTA-specific requirements.

The process description *ends* after the feedback about the status of the installation of the update by the vehicle to the OEM. This is either registered as successful (and possibly reported to the regulatory authority as a successfully completed recall) or, if the installation fails several times, further processing takes place as part of the conventional workshop process. (The workshop process is an undecided part of this analysis).

The focus of this report is the OTA update in the sense of writing access to the vehicle. Reading activities are described in a few sections, with reference to the further requirements for accessing possibly personal data, which are also not discussed in this report.

Furthermore, over-the-air updates can be distinguished between configuration updates (COTA updates), software or firmware updates (so-called SOTA or FOTA updates) and updating navigation maps or writing 3rd party content. In the following, all update types are referenced together as OTA updates and possible distinctions are controlled via attributes (see Chapter 2).

The OTA process is designed or formulated in such a way that only one update is handled at a time, but which can be rolled out for different target systems. This means that for multiple updates - even if they address the same target system - a corresponding number of such process chains would have to be instantiated in parallel (at least mentally).

## 3.2 Descriptive means for process representation

The idealized OTA process was both graphically modeled and described in text form. A detailed explanation and aspects to be considered in the chosen presentation are explained in this chapter.

### 3.2.1 Modeling with BPMN 2.0

BPMN 2.0 was selected as the notation for the process representation. The following legend describes the elements used in the modeling language:

---

<sup>1</sup> The networking of the vehicles enables direct communication with the customer. Among the users there are groups who like to give feedback back to manufacturers and actively participate in product developments.

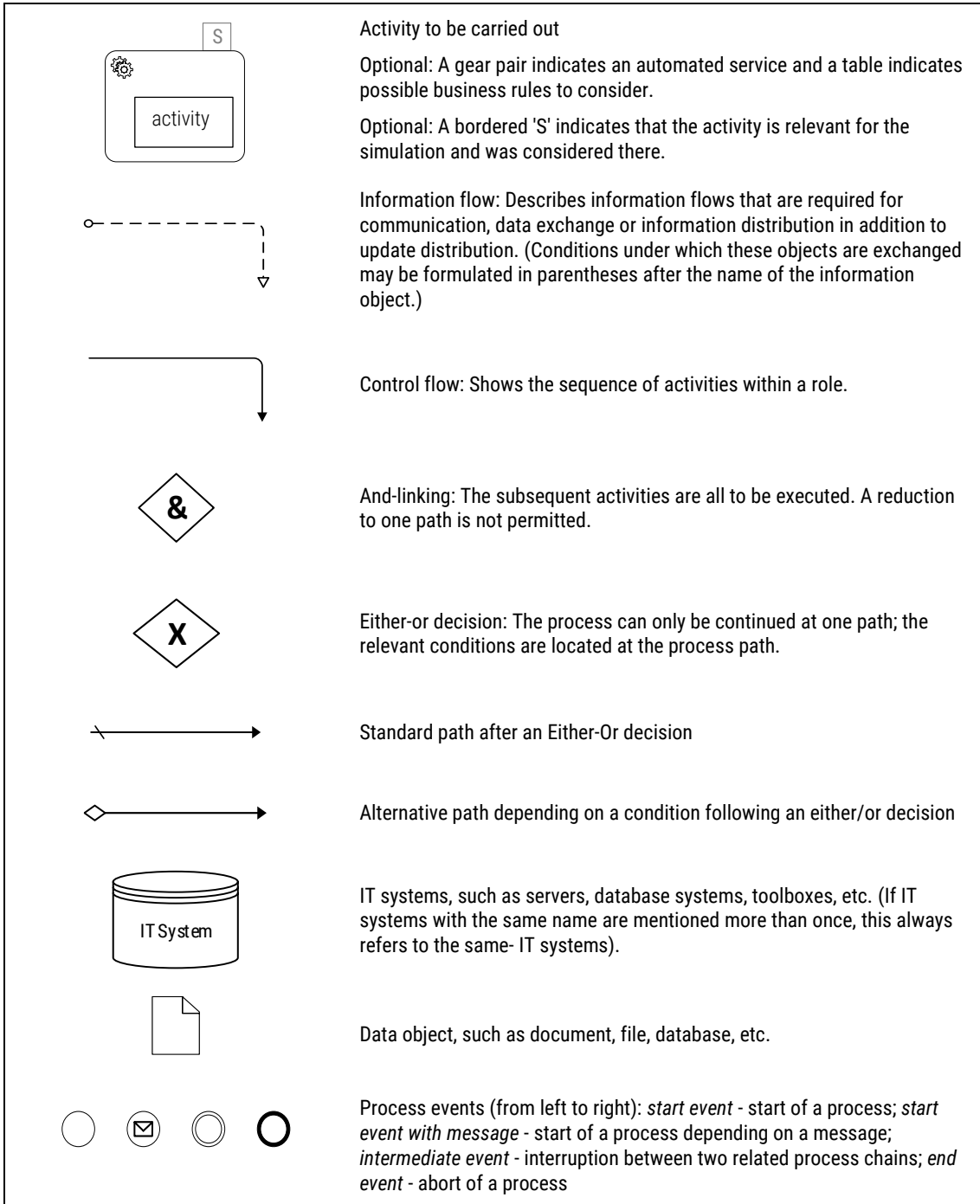


Figure 3: Legend of the used BPMN 2.0

All activities of the process can be identified by a unique process ID, which is then cross-referenced in this documentation. This is made up of three components:

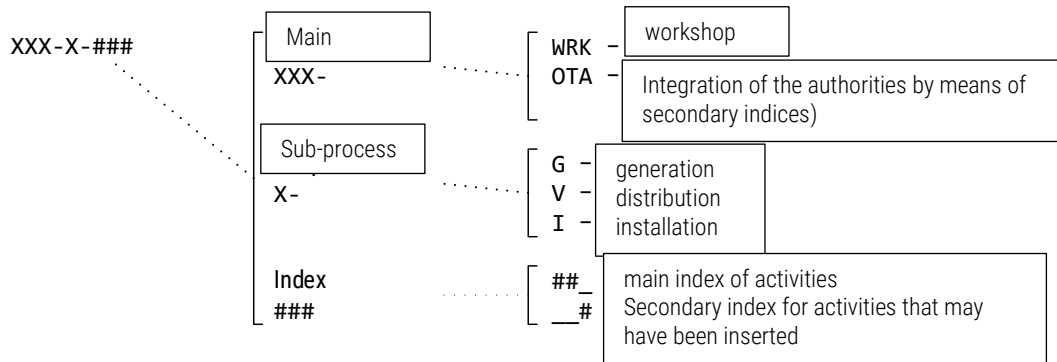


Figure 4: Definition of the Process-ID

During the state assessment, two main processes were initiated: the conventional update process (WRK) for software updates, where the updates are performed by testers in workshops, and the process to perform such Over-The-Air (OTA) updates.

Activities that could also be part of the conventional process are shown in **black** if they can also be found in the OTA process. All new or changed activities due to OTA updates are displayed in **blue**.

Both processes vary when a regulatory authority needs to be involved. These alternations of the process are highlighted in **red**. The activities and information flows are accordingly omitted if the regulatory authority is not involved in the update process. In the text variant, the note "only regul. authority" indicates the extension of the process chain.

Figure 5 shows a corresponding example.

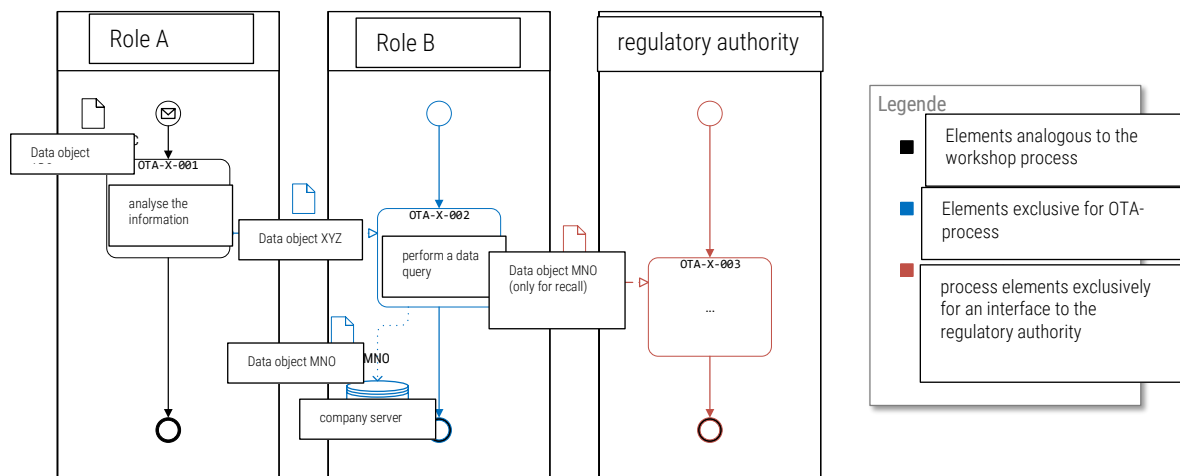


Figure 5: Example of the representation of process variations

### 3.2.2 Text description

Each activity is also displayed in a table. Each cell in the table describes one of the following properties of an activity. However, these are only listed if they are relevant for this activity:

- **Input:** Information object required for the activity.



- **Brief description:** in the description of the activity the person responsible for the activity is highlighted in *italics and bold*, and only in *italics* are all other roles mentioned.
- **RACI model:** The RACI model describes the organizational relationships between the roles involved in an activity.
  - Responsible:** Responsible role for implementation.
  - Accountable:** Role that may have overall accountable responsibility.
  - Consulted:** Role that supports the implementation in an advisory capacity.
  - Informed:** Role that must be informed during the execution of the activity.<sup>2</sup>
- **Output:** Information object that results from the activity.
- **Conditions:** Possible different characteristics that are dependent on other factors, such as the attributes of the update.
- **Opportunities:** Opportunities have a positive influence on one of the stakeholders of the OTA process.
- **Risks:** Risks have a negative impact on one of the stakeholders of the OTA process.
- **Security risks:** Security risks are risks that can be traced back to a cyber security vulnerability. These can be exploited by third parties to the detriment of a stakeholder in the OTA process.
- **Parameters:** describes the parameters relevant for the simulation to influence the activity and its effect on the process chain.
- **UN ECE requirement:** if the regulations of the UN ECE impose a requirement on one of the activities, the original wording of this requirement is deposited here with reference to the chapter of origin.

The entire table is structured as follows:

<u>Input:</u> Information object ABC	
<b>Role A</b> analyzes data object ABC and informs <i>role B</i> using data object XYZ.	
R	Role A
A	Update Coordinator
C	(vehicle owner)
I	Role B
<u>Output:</u> Information object XYZ	
<b>Conditions:</b> If time-critical:	

<sup>2</sup> The RACI model has been exemplarily inserted in order to emphasize the complexity and different interfaces within the respective organizations. The companies themselves should revise the RACI model according to their own structures.

a) then... b) then... If safety-relevant: a) then...
<b>Chance-E###: ...</b> <b>Risk E###: ...</b> <b>Security risk E###: ...</b>
Parameter1: Description of the parameter Parameter2: ...
UNE ECE REQUIREMENTS <sup>3</sup> : #.#.# ...

Table 1: Example of a text description of process activities

Compared to the process designed in the first project, all activities were revised. If activities were significantly changed in their content, the original activity was removed, and the new activity was given a "1" higher index. All remote activities can be found in the appendix. An appropriate justification as to why they have been omitted has been added. All completely new activities are highlighted with [New] in the heading, all optional activities with [Optional].

### 3.2.3 Roles in the process

In the process, different roles are used to represent the agent, department, or organizational unit in the respective specific supply chain. The roles described can vary in their names but also in their areas of responsibility in the various companies or can even be performed in personal union. They are used here to structure the contents:

- The *update coordinator* assumes the coordination of the software update on the OEM side and represents the overall decision maker responsible for the success of the update.
- The *development manager* is responsible for the technical solution.
- The *developer* implements the solution.
- The *Digital Service Provider* represents the internal or external department/company that takes on the additional tasks of distributing the update at OTA. That is, it closes the 'gap' in the transmission that exists between the OEM's servers and the local memory in the vehicle.

The following roles also exist:

<sup>3</sup> The references always refer to Annex A of the 'Final Draft' of the UN ECE Recommendation of 21 September 18.

- *vehicle owner*
- *vehicle*
- *regulatory authority*

### 3.3 UN ECE Task Force on OTA Updates

One of the current driving legal projects is based on the results of the UN ECE Task Force on Cyber Security and Over-the-Air Issues, which can be translated into new legal requirements for OTA updates in the future. The Task Force is a subgroup of the Informal Working Group on Intelligent Transport Systems / Automated Driving (IWG IST/AD) of the UNECE World Forum for Harmonization of Vehicle Regulations (WP.29). Due to this relevance, the requirements of the Task Force are incorporated into the following idealized OTA process and located in the relevant OTA activities (see Annex A of the Recommendation for Software Updates).

Within the UN ECE, two recommendations for the two areas of cybersecurity and software updates were finally drawn up. It should be emphasized that in the area of updates, the scope of tasks no longer only included OTA updates in the course of processing but was generally extended to include software updates.

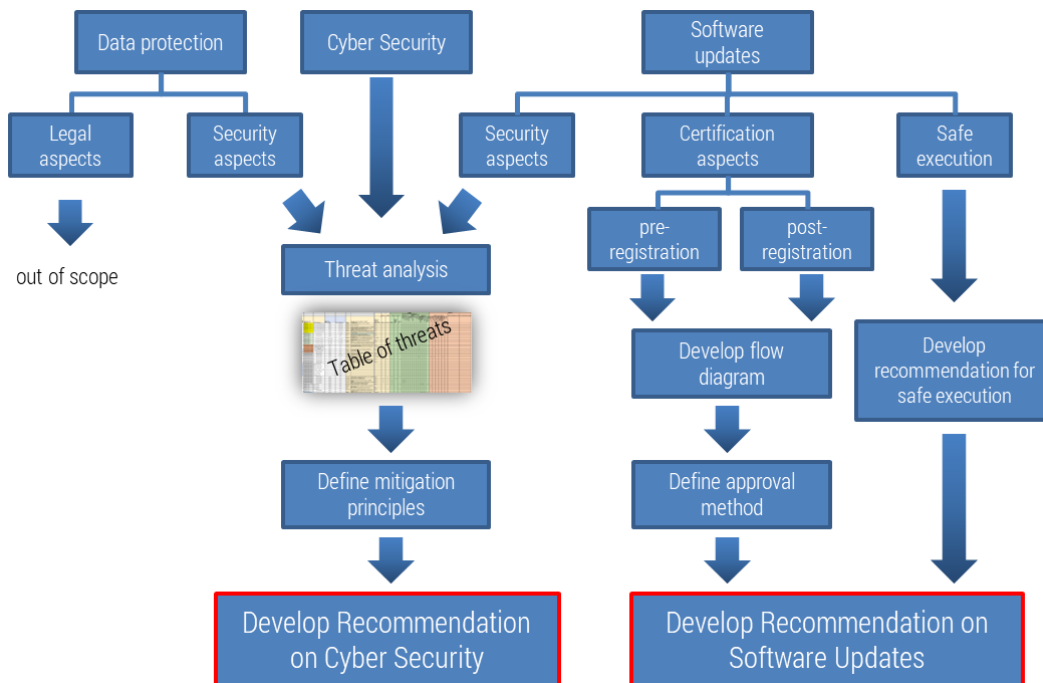


Figure6: Scope of Works of the UN ECE TF on CS/OTA

Figure 7 shows the contents that were developed in the recommendation for the software updates.

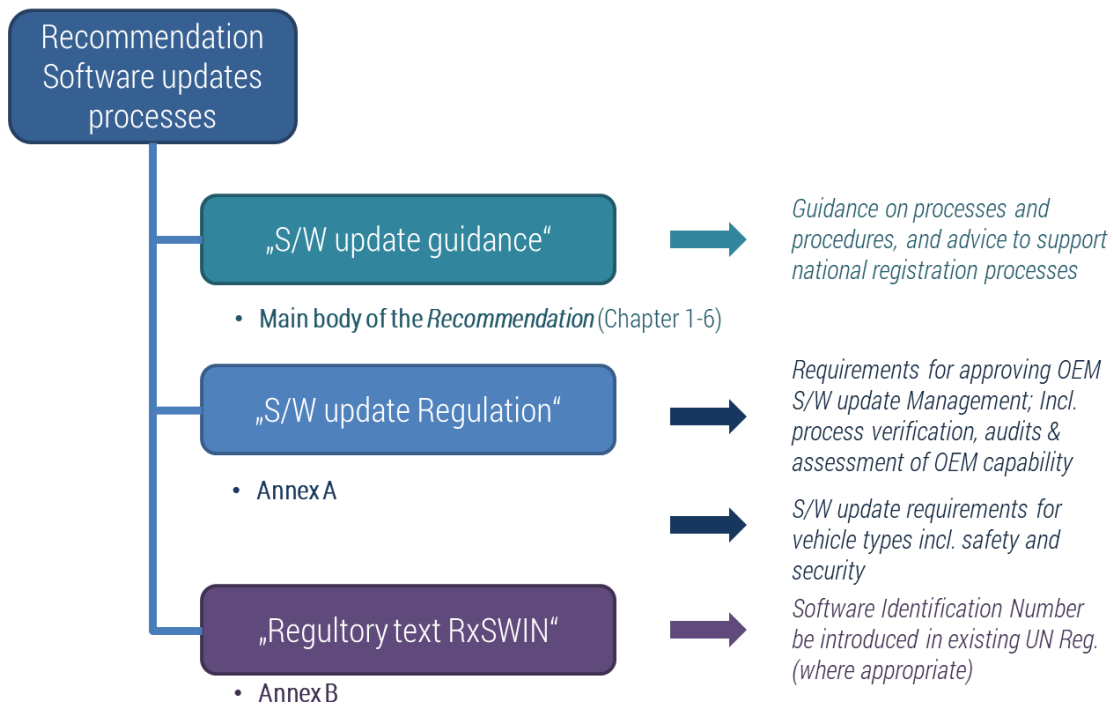


Figure 7: Contents of the Recommendation on Software Updates

The main part of the "S/W update guidance" describes adjustments to the certification process of vehicles in order to consider the official requirements of the UN ECE for software updates. In addition to the certification aspects, safety and security aspects must also be considered in order to be able to carry out software updates in a legally compliant manner.

In particular, the relationship between OEM and regulatory authority is described, while Annex A "S/W update Regulation" contains concrete requirements for the OTA process and the organizational requirements themselves. These requirements from Annex A are subsequently integrated into the idealized process and assigned to the corresponding activities. The original English text is always used.<sup>4</sup>

In addition, however, general process requirements or process prerequisites are described in Annex A.

For example, certification of a Software Update Management System (SUMS) is required to confirm that the requirements of the UN ECE are met. With such a confirmation, the regulatory authority first issues a type approval to update software for corresponding vehicle types. Such a certificate is valid for three years and requires a new certification afterwards. If necessary, it can come even occasionally to a renewed examination of the SUMS certificate. However, if the

<sup>4</sup> The document refers to the status of 21 September of the UN ECE documentation - "Final Draft": <https://wiki.unece.org/pages/viewpage.action?pageId=60362218>

SUMS certificate expires due to a missing extension, non-disclosure of changes or a negative intermediate test, this has no effect on vehicle type approvals already granted.<sup>5</sup>

In addition, there are several UN ECE requirements that could not be assigned to any specific activity in the subsequent idealized process. For reasons of transparency, these are documented below.

- 7.1.1.1 A process whereby information relevant to this regulation is documented and securely held at the vehicle manufacturer and can be made available to an Approval Authority or Technical Service upon request without any burden;
- 7.1.1.2 A process whereby information regarding all initial and updated software versions, including integrity validation data, and relevant hardware components of a type approved system can be uniquely identified;
- 7.1.1.3 A process whereby, for a vehicle type that has an RXSWIN, information regarding the RXSWIN of the vehicle type before and after an update can be accessed and updated. This shall include the ability to update information regarding the software versions and their integrity validation data of all relevant software for each RXSWIN.
- 7.1.1.4 A process whereby, for a vehicle type that has an RXSWIN, the vehicle manufacturer can verify that the software version(s) present on a component of a type approved system are consistent with those defined by the relevant RXSWIN;
- 7.1.2.1 Documentation describing the processes used by the vehicle manufacturer for providing software updates and any relevant standards used to demonstrate their compliance;
- 7.2.1.2.2 The RXSWIN shall be easily readable in a standardized way via the use of an electronic communication interface, at least by the standard interface (OBD port).
- 7.2.1.2.3 The vehicle manufacturer shall protect the RXSWINs on a vehicle against unauthorised modification. At the time of Type Approval, the means implemented to protect against unauthorized modification of the RXSWIN chosen by the vehicle manufacturer shall be confidentially outlined.

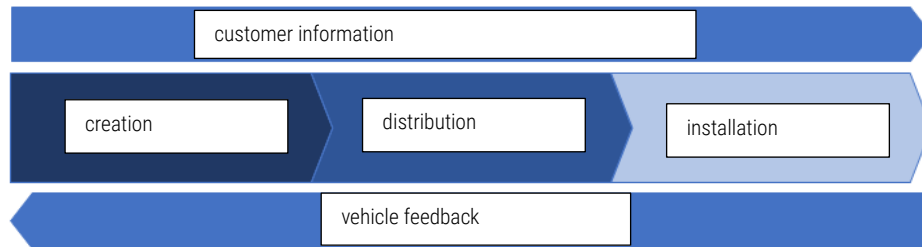
In the third document, Annex B "Regulatory text RXSWIN", the introduction of a "Regulation X Software Identification Number" is described, the subject matter of which, however, lies outside the scope of this report.

---

<sup>5</sup> This paragraph summarizes Chapters 3 to 6 for an overview. Reference should also be made here to Chapters 9 'Conformity of Production' and 10 'Penalties for non-conformity production'.

## 3.4 Process contents

The process of an update is basically divided into three sub-processes, which are accompanied by two central tasks.



**Figure 8: Schema of the update process**

The **generation** includes tasks such as the classification and communication of the update requirement, the involvement of the regulatory authority if necessary, the development of the update as well as the quality control of this update and a subsequent publication of the update under appropriate protection by cyber security measures.

The **distribution of the** update begins with the provision of the update in a distribution network, continues with the notification of the vehicle and the customer, who has to approve the update, and ends with the transfer of the update to the vehicle.

The **installation** then begins with a verification of the update and a check of the vehicle configuration before the installation is initiated. After the installation has been completed, it must be tested, and the result reported back.

**Customer information** must always be considered along the entire process chain. Customers must be continuously informed about update availability, content and progress. An accompanying **vehicle feedback** supports the identification and avoidance of problems and their subsequent elimination for all affected vehicles.

In the following, the OTA update process is described, illustrated and examined. In each case for the process steps generation, distribution and installation.

### 3.5 Update generation

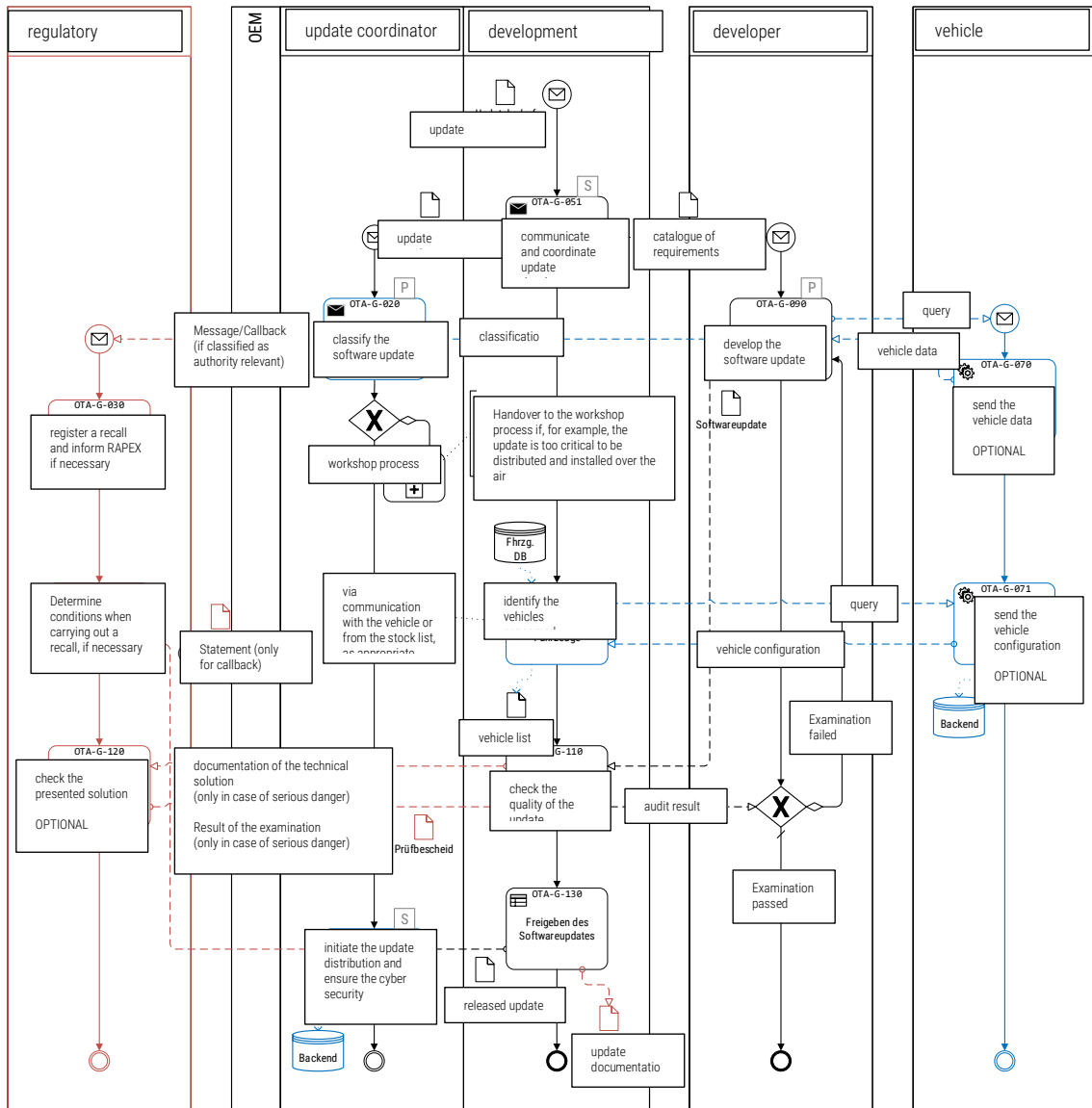


Figure 9: Process modeling "generation"

#### OTA-G-051 [new]

**Input:** Update requirement, requirements catalog / requirement specification

The **development manager** coordinates the development of the update with the **developers** within and/or outside the OEM. For this purpose, he formulates and coordinates the requirements for the update<sup>6</sup>. It also communicates the start of the update development to the **update coordinator**.

<sup>6</sup> Depending on the development method, this may be a continuous process along the update development or an upstream activity that concludes with a result, e.g. a requirement specification.

R	Update coordinator
A	Update coordinator
C	development manager
I	developer, regulatory authority
<u>Output:</u> Update requirement, catalogue of requirements / specifications, update documentation analogous to UN ECE	
<p><code>duraGenerateUpdate</code>: Duration [in hours] required to generate the update. Refers to the duration of the activities from the Generation subprocess, including activity OTA-G-130.</p> <p><code>nVehicles</code>: Number of vehicles [-] that require the update. A refinement and verification of the index follows in OTA-G-060.</p>	
<p>UN ECE REQUIREMENT:</p> <p>7.1.2.5. Documentation for all software updates for that vehicle type describing:</p> <ol style="list-style-type: none"> <li>1. The purpose of the update;</li> <li>2. What systems or functions of the vehicle the update may impact;</li> <li>3. Which of these are type approved (if any);</li> <li>4. If applicable, whether the software update affects any of the relevant requirements of those type approved system;</li> <li>5. Whether the software update affects any system type approval parameter;</li> <li>6. Whether an approval for the update was sought from an approval body;</li> <li>7. How the update may be executed and under what conditions;</li> <li>8. Verification that the software update will be conducted safely and securely.</li> <li>9. Verification that the software update has undergone adequate verification and validation procedures.</li> </ol>	

## Explanation

The distribution of software updates over-the-air also results in new requirements that must be added to the specifications. These result on the one hand from the state of the art, such as requirements for the transmission or the installation routine depending on the performance of the ECU, and on the other hand from the requirements of UN ECE and the own goal of establishing a robust OTA update process. These include in particular safety requirements to be defined according to the objective of the update, e.g. that the vehicle shall not be parked on a slope when an update to the braking system is installed, or that a vehicle shall not be updated while in motion.

The cited paragraph of the UN ECE regulations serves as an initial assessment of what is required on the part of the UN ECE. These requirements also imply other requirements that could be added to the specifications. Further implications arise from the other activities described here.

The documentation required by the UN ECE can be added as a draft to the update requirement in order to be able to carry out a better classification in OTA-G-020 and thus, if necessary, integrate the regulatory authority. The final documentation could be forwarded to the regulatory authority during OTA-G-130 "Approving the software update" as an attachment to the update package.



**OTA-G-020**

<u>Input</u> : update requirement notification, requirement requirement / requirement specification, documentation analogous to UN ECE	
The <b>update coordinator</b> classifies the software update. <sup>7</sup>	
R	Update coordinator
A	Update coordinator
C	development manager
I	developer, regulatory authority
<u>Output</u> : Addition of update class to the update requirement notification; if necessary, notification to the <i>regulatory authority</i>	
<p><b>Conditions:</b></p> <p><b>(A)</b> If the update is relevant to product safety, the <i>regulatory authority must be</i> involved (→ OTA-G-030).</p> <p><b>(B)</b> If the update is too critical to be distributed over-the-air, here is the abort and transfer to the workshop process (→ WRK-G-040)</p> <p><b>(C)</b> If the update is time-critical, the regulatory authority may possibly be involved at a lower level and only after Anschluss has been informed of the measure.</p>	
<p><b>Chance-E001:</b> The update process can be made more efficient by classifying OTA updates and the associated variation of the process execution depending on the classification.</p> <p><b>Risk-E002:</b> A wrong classification can lead to a violation of a reporting obligation to the <i>regulatory authority</i>.</p> <p><b>Risk-E003:</b> A wrong classification can lead to the update not being sufficiently tested and errors in the field resulting.</p> <p><b>Risk-E027:</b> Some legal regulations exist regarding the responsibility for the maintenance of vehicles for the seller and therefore this responsibility may lie with the dealer's workshop and not with the OEM, so that the OEM itself may not install any updates.</p>	
Class: this class stores the update properties according to the classification made.	
<p>UN ECE REQUIREMENT:</p> <p>7.1.1.8 A process to assess, identify and record whether a software update will affect any type approved systems. This shall consider whether the update will impact or alter any of the parameters used to define the systems the update may affect or whether it may change any of the parameters used to type approve those system (as defined in the relevant legislation);</p> <p>7.1.1.9 A process to assess, identify and record whether a software update will add, alter or enable any functions that were not present, or enabled, when the vehicle was type approved or alter or disable any other parameters or functions that are defined within legislation. The assessment shall include consideration of whether:</p> <ol style="list-style-type: none"> <li>1. Entries in the information package will need to be modified</li> <li>2. Test results no longer cover the vehicle after modification</li> </ol> <p>7.1.1.10 A process to assess, identify and record if a software update will affect any other system required for the safe and continued operation of the vehicle or if the update will add or alter functionality of the vehicle compared to when it was registered</p>	

<sup>7</sup> Classification schemes see chapter 2.4

7.1.4.2 The vehicle manufacturer shall demonstrate the processes and procedures they will use to ensure that, when an over the air update requires a skilled person, such as a mechanic, in order to complete the update process, the update can only proceed when such a person is present.

### Explanation

If the OEM recognizes that a software update results in a retroactive effect on the type approval by the software update, the process must be initiated with the regulatory authority. It shall be examined whether an extension of the existing type-approval is required or whether a new type-approval is required.

### OTA-G-030 [reg. auth. only]

<u>Input</u> : Message / Callback Login	
The <b>regulatory authority</b> registers the recall and informs RAPEX about the recall if necessary.	
R	Regulatory authority
A	Update coordinator
C	Update coordinator
I	RAPEX if necessary
<u>Output</u> : Callback registration	

### OTA-G-031 [reg. auth. only]

<u>Input</u> : Callback registration	
If necessary, the <b>Regulatory Authority will</b> set a deadline for the fulfilment of the recall and/or further conditions.	
R	Regulatory authority
A	
C	
I	Update Coordinator
<u>Output</u> : Instruction to OEM if necessary	
UN ECE REQUIREMENT:	
8.1. Every modification of the vehicle type shall be notified to the approval authority which granted the approval. The approval authority may then either:	
8.1.1. Consider that the modifications made are unlikely to have an appreciable adverse effect and that in any case the vehicle still complies with the requirements; or	
8.1.2. Require a further test report from the technical service responsible for conducting the tests.	

**OTA-G-090**

<u>Input</u> : Specifications, vehicle data
The <b>developer</b> develops the new software update. This also includes defining technical boundary conditions for the update. If necessary, existing vehicle data can be used to improve the diagnosis and the update itself (→ OTA-G-070). For this purpose, the <b>developer</b> can optionally send a corresponding request to the <b>vehicle</b> .
R      Developer A      Development manager C      (if applicable, vehicle), (if applicable, department that reported the update requirement) I      (update coordinator if necessary, if there are relevant changes)
<u>Output</u> : Software update, if necessary request for vehicle data
<b>Conditions:</b> <b>(A)</b> If time-critical, the update should be distributed separately from other packages.
<b>Chance-E004:</b> Remote access to data to support troubleshooting can lead to improved error interpretation and resolution. <b>Risk-E005:</b> The basic conditions for the data exchange of personal data between OEM and possible external <b>developer</b> have to be clarified. <b>Risk-E006:</b> The developed software update does not meet the requirements for secure software and makes the updated vehicle vulnerable to attacks (e.g. data theft, IP theft, malware, etc.). <b>Security Risk E007:</b> Poor key management during software development (but also during the entire lifecycle) gives attackers the ability to destroy, exchange, or limit the availability of authentication keys. <b>Security risk E007b:</b> The release of the update for distribution via USB sticks / smartphone opens another interface whose protection must be considered in the entire design. <b>Chance-E026:</b> For security-critical updates, a separate distribution of updates that change behavior, interface, or other customer-remarkable features may be helpful in obtaining rapid approval of the update.
sizeUpdate: Size of the Update [in MB] speed#G: Speed of various transfer techniques (WiFi, 3G, 4G and 5G or via USB stick) p#G: Distribution of which transmission technology is expected to be used in which percentage of 100% for transmission into the vehicle.
UN ECE REQUIREMENT: 7.1.1.5. A process whereby any interdependencies of the updated system with other systems can be identified; 7.1.2.2. Documentation describing the configuration of any relevant type approved systems before and after an update, this shall include unique identifiers for the type approved system's hardware and software and any relevant vehicle or system parameters; 7.1.2.3. For every RXSWIN, there shall be documentation describing the software relevant to the RXSWIN of the vehicle type before and after an update. This shall include information of the software versions and their integrity validation data for all relevant software for each RXSWIN. 7.1.3.3. The processes used to verify and validate software functionality and code for the software used in the vehicle are appropriate.

## Explanation

When developing the update, basic cyber security relevant measures (such as secure coding) must be taken into account on the one hand, and measures to optimize the update package for over-the-air transmission (such as consideration of relevant frameworks) on the other. This also includes concepts such as the transmission of only individual differential information (e.g. diff files), i.e. data packets containing only the modified code (and corresponding scripts for protection and installation), or the streaming of update content. This must be considered in the design.

The optimization for different transmission techniques depends both on the criticality and on the user group. Critical updates could preferably be distributed via mobile radio. This significantly increases network coverage and reduces the probability of an update abort (→see OTA-G-130). The update can thus reach the target vehicle faster - even at slower speeds, as the accessibility is higher. However, the OEM may have to bear the costs of the transfer itself.

With regard to the customer group, on the one hand their affinity to technology must be considered, as must their sensitivity to necessary updates, and on the other hand their purchasing power and possible differences in the installed connectivity modules or completed mobile phone packages in the target vehicles.

As an option, the software update can either be downloaded to a mobile data storage device (e.g. a smartphone or USB stick) and imported directly into the vehicle from this data carrier or downloaded via a smartphone as a radio module using tethering without direct vehicle connectivity. However, these transmission variants represent a further cyber security risk because another interface is opened, and third-party devices are used that cannot be secured by the OEM.

When an update is developed within the supply chain, there must be security throughout the entire chain and the update package must be protected against external manipulation.

Regarding Chance-E026: the separate, i.e. atomic distribution of updates makes it possible, for example, for security-critical updates to be quickly distributed to the target vehicles and also approved, since an atomic security update has fewer installation requirements than other updates. A short time that the vehicle is not usable motivates the driver to install the critical vehicle rather than when the update is part of a larger package for which the vehicle cannot be used for a longer period of time. However, the disadvantage of atomic distribution could be that customers receive too many requests for updates. The latter might feel disturbed by this frequency and develop an automatism to postpone or reject these updates.

## OTA-G-070 [optional]

Input: Request for vehicle data

The **vehicle** sends - if a request was initiated in OTA-G-090 - the vehicle data to the *developer*. The vehicle data includes, for example, diagnostic data or, depending on the owner's approval, load or driving profiles. With the help of this data the update development can be improved.

R	Vehicle
A	Development Manager
C	(if applicable, vehicle owner / user for consent to data use)
I	Developer
<u>Output:</u> Vehicle configuration / vehicle data (to OTA-G-090);	
<b>Risk-E008:</b> In the event of unclear framework conditions under which data may be retrieved from the vehicles, regulatory regulations may be violated.	
<b>Security Risk E009:</b> Unauthorized access to corporate or personal data during data transfer.	
<b>Security-Risk-E010:</b> The data interface and/or other vehicle electronics are manipulated or misused.	

### Note on simulation

This activity was not considered in the simulation (chapter 4), because the trigger activity OTA-G-090 (ideally) is not finished before the desired feedback from the vehicle can be considered. Thus, OTA-G-090 is decisive in terms of time for the simulation.

### OTA-G-060

<u>Input:</u> vehicle configuration (from OTA-G-071); data from own database	
The <i>development manager</i> begins to identify the affected vehicles. This can be done by directly requesting the OTA-capable vehicles ('Shoulder-Tap') or indirectly by accessing a database in which the vehicles could write their configuration / other properties regularly (at a specified time interval) ('Vehicle-DB'). If necessary, this database will also be supplemented with further data from internal sources (e.g. customer portal).	
R	development managers
A	development managers
C	
I	vehicles
<u>Output:</u> List with the vehicles to be updated; if necessary request (to OTA-G-071)	
<b>Chance-E011:</b> Remote access to the vehicles allows the affected vehicles to be identified more reliably and their current software status to be analyzed. This avoids initiating an update process that could fail due to lack of compatibility.	
UN ECE REQUIREMENT: 7.1.1.5. A process whereby any interdependencies of the updated system with other systems can be identified; 7.1.1.6. A process whereby the vehicle manufacturer can identify target vehicles for a software update; 7.1.2.4. Documentation listing target vehicles for the update and verification of the compatibility of the registered configuration or last known configuration of those vehicles with the update.	

**OTA-G-071 [optional] [new]**

<u>Input:</u> Request
The <b>vehicle</b> transmits the vehicle configuration (software versions, installed hardware, etc.) according to OTA-G-060.
R      Vehicle A      Development Manager C I
<u>Output:</u> Vehicle configuration (on OTA-G-060)
<b>Risk-E-012:</b> In case of unclear framework conditions under which data may be retrieved from the vehicles, regulatory regulations may be violated. <b>Security Risk E013:</b> Unauthorized access to corporate or personal data during data transmission. <b>Security Risk E014:</b> The data interface and/or other vehicle electronics are manipulated or misused.

**OTA-G-110**

<u>Input:</u> software update; documentation about the update
The <b>development manager</b> checks the quality of the new software version. If a serious hazard is <sup>8</sup> suspected, the chosen solution is documented and forwarded to the <i>regulatory authority</i> for further examination (→ OTA-G-120).
R      Development manager A      Development manager C      Quality, developer I      (regulatory authority / technical service if necessary)
<u>Output:</u> Test result; if necessary, quality-checked update; if necessary, documentation of the technical solution (only in case of serious hazard)
<b>Conditions:</b> <b>(A)</b> the software update meets the requirements of the specifications and passes the <i>regulatory authority's</i> examination. The release process can be initiated (→ OTA-G-130). <b>(B)</b> it meets the requirements of the specifications but does not pass the <i>regulatory authority</i> test. The <i>development manager</i> must revise the specifications (→ OTA-G-050). <b>(C)</b> it does not meet the requirements of the specifications. The development manager calls on the developer to make improvements (→ OTA-G-090).

<sup>8</sup> In accordance with the applicable opinion of the relevant regulatory authority:  
([https://www.kba.de/SharedDocs/FAQ/DE/Marktueberwachung/Produktsicherheit/ernste\\_Gefaehrdung.html](https://www.kba.de/SharedDocs/FAQ/DE/Marktueberwachung/Produktsicherheit/ernste_Gefaehrdung.html))

**Risk E015:** The update changes the vehicle functionality beyond the permitted level so that the approval can expire.

UN ECE REQUIREMENT:

7.1.1.7. A process to verify, before a software update is issued, the compatibility of possible software/hardware configurations for the registered configuration or last known configuration of the target vehicles with the software update;

7.1.1.12.A process whereby the vehicle manufacturer shall be able to make the information according to paragraph 7.1.2.3. and 7.1.2.4. available to relevant Authorities or Technical Services.

### OTA-G-120 [reg. auth. only]

Input: Documentation of the technical solution

Only in case of serious danger: The **regulatory authority** checks the presented solution for conformity.

R Regulatory Authority

A

C update coordinator

I

Output: test report

UN ECE REQUIREMENT:

8.1.3. Confirmation or extension or refusal of approval, specifying the alterations, shall be communicated by means of a communication form conforming to the model in Annex 2 to this Regulation. [...]

### OTA-G-130

Input: quality-tested update, update documentation analogous to UN ECE

The **development manager** coordinates the internal release process.

R Development manager

A

C Departments according to internal guidelines

I

Output: released update; update documentation analogous to UN ECE

UN ECE REQUIREMENT:

7.1.2.5. Documentation for all software updates for that vehicle type describing:

1. The purpose of the update;

- |    |  |
|----|--|
| 2. | What systems or functions of the vehicle the update may impact;  |
| 3. | Which of these are type approved (if any);   |
| 4. | If applicable, whether the software update affects any of the relevant requirements of those type approved system; |
| 5. | Whether the software update affects any system type approval parameter;  |
| 6. | Whether an approval for the update was sought from an approval body;   |
| 7. | How the update may be executed and under what conditions;  |
| 8. | Verification that the software update will be conducted safely and securely.                                       |
| 9. | Verification that the software update has undergone adequate verification and validation procedures.               |

### Explanation

After successful quality inspection, the release process can be initiated. Depending on the organizational form, this can take longer and be an obstacle to the necessary reactivity in an update process for critical updates. An adjustment of the hierarchy for different update types can optimize this and reduce a possible time delay. For example, in the case of updates that are only relevant to security, the release can only be located with the security experts. A release by the person responsible for the entire vehicle and others could be omitted if necessary. It would ultimately also be the responsibility of the shortened release hierarchy to escalate uncertainties further.

### Note on simulation

The temporal effect of this activity is to be considered in the simulation in the duration of the entire generation process.

### OTA-G-141 [new]

<u>Input</u> : released update, requirements catalog	
The <b>update coordinator</b> initiates the update distribution by uploading the update to an update server / backend and notifies the <i>digital service provider</i> . In the course of the upload to the <i>digital service provider</i> , the <i>update coordinator</i> checks the initiated security-relevant mechanisms against the catalogue of requirements for compliance. If necessary, he will contribute further, non-technical requirements of the <i>regulatory authority</i> .	
R	Update coordinator
A	Cyber-Security Department
C	
I	Digital service provider
<u>Output</u> : released and signed update	
duraForward2CDN: describes the time required to perform this activity depending on the internal processes.	



duraUpdateMax: describes the time that the update is distributed over-the-air at the longest. This may be required by a specification of the KBA and ultimately terminates the transmission via OTA in the simulation automatically.

UN ECE REQUIREMENT:

7.1.3.1. The process they will use to ensure that software updates will be protected to reasonably prevent manipulation before the update process is initiated;

7.1.3.2. The update processes used is protected to reasonably prevent it being compromised, including development of the system update;

7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.

### Explanation

The aim of this activity is to secure the secure channel between the originator (OEM) and the destination (customer) of the update, so that it can be assumed that as long as the secure channel exists, the content of the packet is irrelevant during transmission.

### 3.6 Update distribution

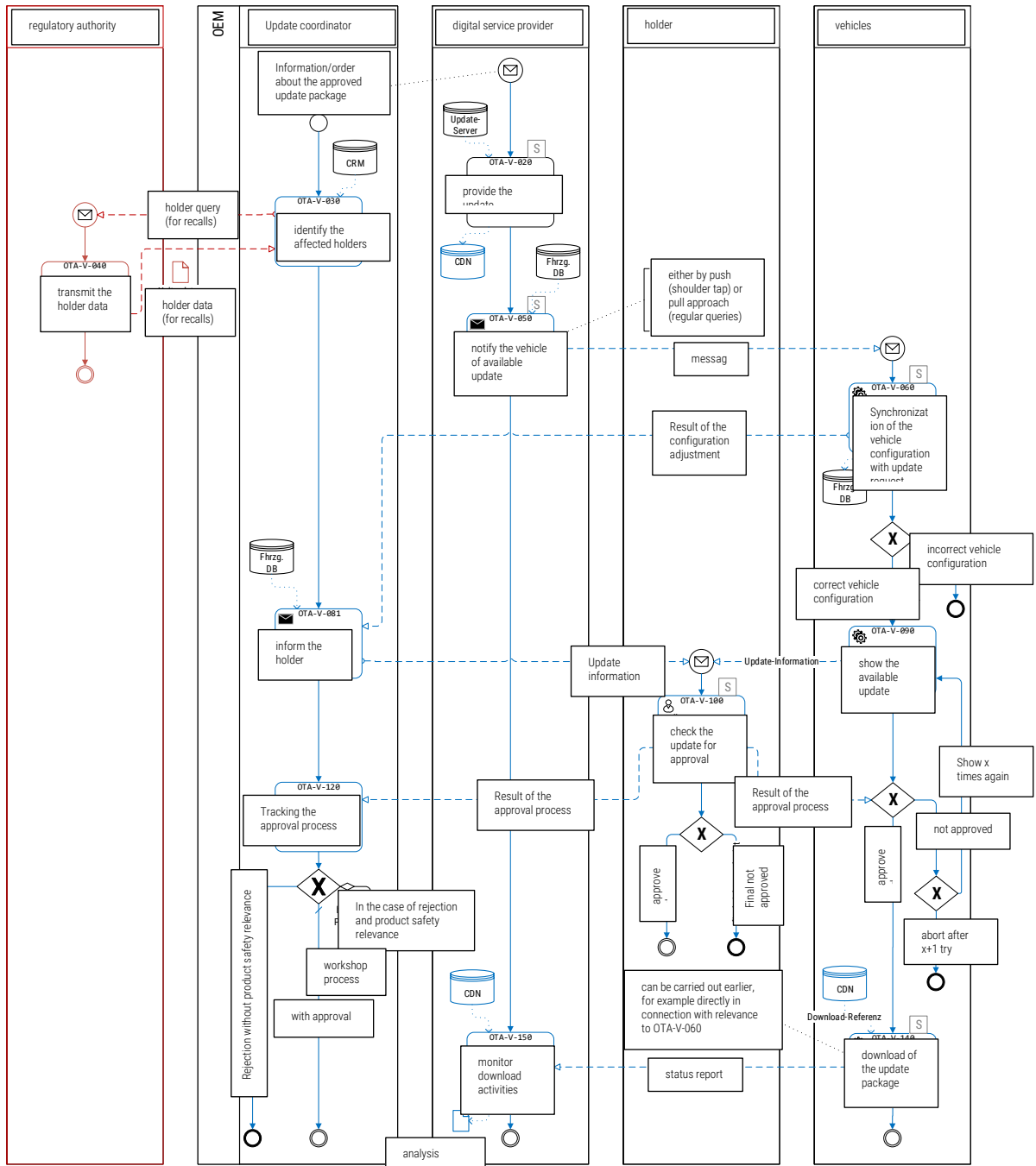


Figure 10: Process modeling "distribution"

**OTA-V-020**

<u>Input</u> : released and signed update	
The <i>digital service provider</i> provides the update on a corresponding server structure / content delivery network (CDN). <sup>9</sup>	
R	Digital service provider
A	
C	
I	update coordinator
<u>Output</u> : Update provided in CDN	
<b>Risk-E016</b> : By using a digital service provider as an intermediary, the update is exposed to security risks on the third-party infrastructure (e.g. theft of technical know-how or personal data, lack of availability, breach of integrity, etc.).	
<b>Security risk E017</b> : Insufficient security at the digital service provider leads to a corresponding security risk: update data can be read, manipulated updates can be installed, malware can be installed.	
duraUpload2CDN: describes the duration required by the digital service provider to mobilize and provide the update for retrieval by the vehicles.	

**Explanation**

The infrastructure of the *digital service provider* is a possible target for malicious manipulation of the update process. Here, the update itself can be attacked or the infrastructure and thus the communication between the backend and the vehicle fleet can be attacked.

This security risk can be addressed by clear regulations on the security measures to be taken by the *digital service provider*. This can take place within the scope of the order and be supported by the safety certification of service providers.

Other technical measures, such as partitioning the backend, can also be supported. Such and other technical measures mean on the one hand a higher maintenance effort for the digital service provider, but on the other hand they increase the security level by limiting, for example by partitioning, the access of an attacker to a part of the vehicle fleet and the corresponding updates.

**Note on simulation**

Internal and external factors must be considered to parameterize the duration of this activity. The decisive factor is whether the *digital service provider* is in-house or an external contractual partner has been commissioned with this. The performance of this organization is accordingly decisive for the execution of the activity, but also for the communication between the client and the

---

<sup>9</sup> To distribute the update to the vehicles, a new server structure is required, which must serve a significantly higher number of accesses than when distributing the update to the workshops. If necessary, a content delivery network must be set up or integrated via a third-party provider.

contractor organizational unit. This can be improved through an efficient process and the provision of the necessary information. Extensive automation can further shorten the duration.

### OTA-V-030

<u>Input:</u> CRM tool, vehicle owner data (via the <i>regulatory authority</i> )	
The <b>update coordinator</b> links the identified vehicles with the corresponding keepers for correct addressing. On the one hand the data comes from the internal CRM tool, on the other hand a comparison with the <i>regulatory authority</i> takes place.	
R	Update coordinator
A	
C	Customer Relations
I	Regulatory authority
<u>Output:</u> Vehicle owner request / FIN list of identified vehicles	
<p><b>Conditions:</b></p> <p><b>(A)</b> If authority-relevant, then the keeper data are to be queried in order to reach the keepers also on channels outside the vehicle if necessary and to inform about the update.</p> <p><b>(B)</b> This activity can also take place earlier, parallel to the update development (e.g. analog with OTA-G-060).</p>	

### Explanation

It must be clarified which channel is sufficient to inform the customer - also in the case of recalls. The communication in the vehicle does not always reach the owner and thus possibly also no legally valid approval of the update. In addition, it is necessary to link the owner data in order to check under which regulatory supervision the vehicle is currently located (e.g. if the vehicle is resold abroad).

### Note on simulation

This activity was not simulated because it is a parallel activity to critical activities. Their influence on the overall throughput time of the process is secondary. Even with safety-critical updates, this is not an obstacle, as the vehicles can be addressed without this activity and can receive the update.

### OTA-V-040 [optional] [reg. auth. only]

<u>Input:</u> Vehicle owner request	
If necessary, the <b>Regulatory Authority shall</b> transmit the keeper data on the vehicles concerned.	
R	Regulatory Authority

A	
C	
I	Update Coordinator
<u>Output:</u> vehicle owner data	

### Explanation

This activity is also listed because it could ultimately be a temporal component in the overall process - depending on the regulatory framework.

### OTA-V-050

<u>Input:</u> Vehicle database	
The <b>digital service provider</b> notifies the vehicles of the available software update. This is done either directly via a 'shoulder tap' or indirectly by the <i>vehicle</i> checking at defined time intervals whether updates are available ('pull method').	
R	Digital service provider
A	Update coordinator
C	Update coordinator
I	Vehicle
<u>Output:</u> message	
<p><b>Security Risk E018:</b> The message can be manipulated to instruct the vehicle to roll-back to a previous, potentially incorrect version.</p> <p><b>Risk-E019:</b> The message does not reach the vehicle, so the vehicle remains in an unsafe condition. It may not be possible to establish a connection to the vehicle (technical defect, privacy mode of the vehicle owner, etc.).</p> <p><b>Chance-E020:</b> Due to high available capacities, more vehicles can be addressed and updated simultaneously than via the existing workshop and sales networks.</p>	
<p>methodCallVehicle: differentiated between shoulder tap and pull variant</p> <p>duraToCallVehicle: the duration to initiate a message both on the server (Shoulder-Tap method) and in the vehicle (Pull method).</p> <p>pSuccessCallVehicle: the probability that a vehicle will be reached / that the connection can be established.</p> <p>maxCallsAtATime: the maximum number of messages sent to a vehicle at one time in succession should not result in a direct response.</p> <p>duraNextCalls: the duration between two notification times.</p> <p>maxCallAttempts: the maximum number of messages to a vehicle before the OTA process is aborted and transferred to the workshop process.</p> <p>nDaysToAskForUpdate: Number of days at which the vehicle requests updates on the server (Pull method only)</p>	

## Explanation

This process step can be carried out in two variants. Either the vehicles are notified directly by means of a so-called 'Shoulder-Tap' by sending a message to each vehicle informing it that an update is available. The other variant corresponds to a 'pull' configuration, in which the vehicles independently request the backend for available updates at regular intervals with an existing data connection. The method must be defined before the process is carried out (`methodCallVehicle`). An advantage of the Shoulder-Tap method is the speed at which all vehicles can be reached, but the disadvantage is that a higher server capacity must be available. In addition, the Shoulder-Tap method is also more difficult to control, since the access rates are much more volatile than with the Pull method.

With the pull method, the vehicles access the server over a longer period, resulting in a smoothed access curve. Accordingly, demand can be optimized more cost-effectively and even access times can be shortened, as server overloads or bottlenecks can be more easily avoided.

Due to the delayed distribution via pull method, even if errors occur, only a small number of vehicles may be affected before the update is recalled. Such a beta phase that only a part of the fleet is updated at first, can also be planned and is then generally independent of both methods, but more effective by means of Shoulder-Tap, since the time delay until the feedback of the last vehicle could be smaller.

The probability that a vehicle will be reached / the connection can be established (`pSuccessCallVehicle`) depends, among other things, on the vehicle type and user type. A Smart EV primarily has a different profile than a city car than a commuter vehicle, which travels over land more frequently. Long overland journeys, for example, influence the accessibility of the vehicle through varying network coverage in the mobile network. In contrast, coverage by high-speed mobile communications is significantly higher due to the higher demand in the city. On the other hand, private charging infrastructure, which may offer WiFi-connections in addition to electricity and thus increase the accessibility of vehicles, is available much less frequently.

## OTA-V-060

<u>Input</u> : message about available update	
The <b>vehicle compares</b> the vehicle configuration with the update requirements (installed hardware, installed software, etc.) and checks whether the vehicle configuration has changed in the meantime due to a workshop visit (e.g. after an accident) or whether the software status has changed due to previous updates. The process is automated.	
R	Vehicle
A	Update coordinator
C	Digitaler Dienstleister
I	Update coordinator

Output: Result of the configuration adjustment

**Risk E021**: The conditions under which data may be retrieved from the vehicle shall be clarified.

**Security risk E022**: The data interface and / or other vehicle electronics can be manipulated or misused via the data interface. This can be done by a Denial of Service (DoS) attack, for example, so that the message cannot be received or processed because the Connectivity Unit is overloaded with processing too many requests.

**Chance E023**: Frequent communication with the vehicles ensures that the database is always up-to-date.

`duraCheckConfig`: the time it takes to check the vehicle configuration on the vehicle side and respond to the server.

`pSuccessConfig`: the probability that vehicle configuration is correct.

### Explanation

This activity takes place automatically and its robustness thus depends on the quality of the technical implementation of the function in the vehicle. This also influences the simulation and the quality of the process result.

**OTA-V-081 [new]**

<u>Input</u> : Result of the configuration adjustment, updated vehicle database	
The <b>update coordinator</b> informs the <i>vehicle owner</i> about the update.	
R	Update coordinator
A	
C	
I	Halter
<u>Output</u> : update information	
<p><b>Opportunities-E024:</b> The possibility of simpler and quicker software changes enables new business models (e.g. Function on Demand).</p> <p><b>Chance-E025:</b> By performing the software update without the participation of a workshop, cost and time savings are possible.</p> <p><b>Risk E028:</b> Insufficient information from the vehicle owner about the software update could constitute an unauthorized interference with the owner's property rights.</p> <p><b>Security risk E030:</b> The communication of the update to the current owner of the vehicle must be carried out with special regard to privacy requirements. For example, payment data could be compromised in the case of subsequent function activations.</p> <p><b>Risk-E031:</b> Updates that are more critical in terms of time or security are not atomically packaged and may not be considered.</p>	
<p>UN ECE REQUIREMENT:</p> <p>7.1.1.11 A process whereby the vehicle user is able to be informed about updates.</p> <p>7.2.2.2 The vehicle manufacturer shall demonstrate that the vehicle user is able to be informed about an update before the update is executed. The information provided may contain:</p> <ul style="list-style-type: none"> <li>• The purpose of the update. This could include the criticality of the update and if the update is for recall, safety and/or security purposes;</li> <li>• Any changes implemented by the update on vehicle functions;</li> <li>• The expected time to complete execution of the update;</li> <li>• Any vehicle functionalities which may not be available during the execution of the update;</li> <li>• Any instructions that may help the vehicle user safely execute the update;</li> <li>• In case of groups of updates with a similar content one information may cover a group.</li> </ul>	

**Explanation**

The information about the update can take place in different ways: by mail, e-mail, on-screen in the vehicle, smartphone application, etc. The update contents should be presented to the customer in a clear and understandable manner to inform him about the changes to the vehicle and the duration during which the vehicle will not be available to the customer due to the update.

When informing the customer about a recall update, it must be clearly emphasized that this is a recall measure which must be carried out up to a certain point in time. The update may only be



withheld by the customer as long as the corresponding period is defined, after which the vehicle would have to be demonstrated in the workshop or the vehicle might lose its registration.

### OTA-V-090

<u>Input</u> : update information, correct vehicle configuration	
The <b>vehicle displays</b> the available update via the in-vehicle infotainment and requests approval for the update from the <i>owner</i> , if necessary.	
R	vahrzeug
A	update coordinator
C	
I	vehicle owner
<u>Output</u> : Update information (e.g. information screen with option to confirm / delay / reject update)	
<b>Risk-E032</b> : Incorrect information about the update (e.g. duration of the installation) leads to a negative experience for the customer.	
<b>Security risk E033</b> : The update function can be blocked, e.g. by denial of service attacks (overload of the data network or overload of the receiver unit), so that the vehicle remains in an unsafe state.	

### Explanation

In the case of an update, the customer can be offered the possibility to carry out the update, to be reminded again later or to reject the update. It is important to provide the customer with detailed and, above all, enough information. The goal is - among others - that the customer perceives security-critical updates as such and reacts accordingly by approving the installation in a timely manner. This can be achieved by informing the customer directly about the decisive parameters: Duration required for the update; possible costs incurred for the transfer; activities to be performed by the driver; etc. A simple and direct language as well as corresponding menu guidance support this additionally.

In principle, it must be taken into account that the update information in the vehicle can also reach persons who are not mandated for the update approval.

**OTA-V-100**

<u>Input</u> : Update information	
The <b>owner</b> checks the information provided to him and decides whether the update is approved for installation.	
R	vehicle owner
A	vehicle owner
C	
I	Digital service provider, update coordinator
<u>Output</u> : Result of the approval process (approval, deferral or final rejection)	
<b>Either</b> he approves the update <b>or</b> he does not approve the update. In the negative case, the update can be repeated up to x times. If the update is still not approved, the Over-The-Air update process will be aborted and (if necessary / relevant) further tracked in the conventional workshop process.	
<b>Chance-E034</b> : OTA updates reduce the effort for the vehicle customer with software updates.	
<b>Risk-E035</b> : Unauthorized persons approve an update so that unwanted software / updates are installed in the vehicle by the owner.	
<b>Risk-E036</b> : <b>If the</b> keeper rejects the update, the vehicle may remain in a possibly safety-critical condition.	
<b>Risk-E037</b> : Installation without prior consent may result in unauthorized interference with the owner's property rights.	
duraRfC <sup>10</sup> : Duration taken up by the approval process. This is influenced by the technical implementation and communication as well as by the information processing by the vehicle owner.	
duraNextRfC: Duration until the availability of the update is displayed again in the vehicle after the update has been reset.	
maxDuraRfC: maximum duration that an update is displayed before, for example, the workshop process is initiated.	
maxRfC: maximum number of times an update can be postponed. (Can also be calculated from duraNextRfC and maxDuraRfC or specified by these.)	
pSuccessRfC: Probability of a customer approving the update.	

**Explanation**

An implementation of the approval of the update from outside the vehicle is recommended. The owner can be informed about updates via the vehicle key, a smartphone app or similar and can initiate these while he is not at the vehicle. In this way the vehicle owner has more possibilities to integrate the update process into his daily routine and not only to be reminded during the movement. This increases the probability that an update will be approved promptly.

In addition, an analysis of the driving profile can be used to suggest an appropriate time to the owner when the vehicle will not be used, and an update can be carried out without restriction.

<sup>10</sup> RfC: Request for Conformation (project-specific name)

Clean documentation and information preparation for the customer increases the likelihood that the customer will install the update and shortens the customer's processing time until approval. Rarer resetting of the update becomes so more likely. It is also important to strike a balance between the frequency with which the customer must be notified in order to have an update approved or installed as quickly as possible, and the frequency with which the customer may not feel disturbed by the update information.

Another factor to consider in this activity is the respective user groups of the vehicles addressed by the update. For groups with less technical affinity, the information must be communicated differently in language and format. Appropriately trained service staff could be explicitly available as contact persons to support the approval in case of queries. In addition, technology-savvy customers may be open to new software versions and their new functions (analogous to so-called early adapters).

From a legal point of view, it should be noted that the person addressed also has a mandate to approve the update to the vehicle. For example, with fleet vehicles this is usually not the driver.

### OTA-V-120

<u>Input</u> : Result of the approval process
The <b>update coordinator tracks the owner's</b> approval process.
R (update coordinator) A update coordinator C I
<u>Output</u> : if necessary, initiation of the workshop process in the event of rejection
Note: This can be an automated activity. The <i>update coordinator</i> is responsible for monitoring and ensuring that all requirements - e.g. those of the <i>regulatory authority</i> - are met.
<b>Conditions:</b> (A) If permission is granted, the Digital Service Provider will be notified accordingly. (B) If approval has not taken place, either the OTA process is terminated (for an update without product safety relevance) or (C) is terminated and transferred to the shop floor process (for an update with product safety relevance).

### OTA-V-140

<u>Input</u> : Result of the approval process
The <b>vehicle</b> loads the update package into the system via an air interface. After receiving a positive response from the approval process, a script is executed, and a corresponding download reference is

queried to download the update. Alternatively, the update package can also be loaded (and buffered) via a third-party device, which is connected to the vehicle via cable and thus installs the update ("tethering").	
R	Vehicle, digital service provider
A	update coordinator
C	(vehicle owner)
I	Vehicle owner, update coordinator, (regulatory authority)
<u>Output</u> : update stored locally in the <i>vehicle</i>	
<p>If time-critical:</p> <p>a) then the update should be brought forward and be approved in advance of the driver's approval can be downloaded.</p> <p>b) then the update should be distributed via mobile radio and not exclusively via WiFi.</p> <p>If safety-relevant:</p> <p>c) then additional security measures should be taken. (e.g. the transfer to the vehicle is realized using the network provider's own radio channel.)</p>	
<p><b>Risk-E037</b>: The vehicle cannot initiate the download and the update cannot be loaded, e.g. due to an overload of the server infrastructure or a faulty connectivity unit.</p> <p><b>Risk-E038</b>: The update is transmitted incompletely or incorrectly.</p> <p><b>Security risk E039</b>: The data interface and/or other vehicle electronics are manipulated or misused.</p> <p><b>Security risk E039b</b>: With tethering, the corresponding devices are not within the manufacturer's sphere of influence / protection, so that damage could be caused by compromised devices.</p>	
<p>nDownloadSlots: the number of slots available for download. Corresponds to the simulation of the bandwidth. A slot can operate a vehicle.</p> <p>pSuccessDownload: The probability for the success of an update can be improved (A) by better communication or (B) by using mobile networks (possibly at the expense of own costs). Or by activating the download with a smartphone application.</p> <p>duraDownloadIdle: This time can be shortened by installing better hardware.</p>	
<p>UN ECE REQUIREMENT:</p> <p>7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.</p>	

## Explanation

To increase customer satisfaction, the update can be downloaded directly after activation, i.e. before the actual authorization of the update by the vehicle owner. An installation can therefore take place without delay directly after authorization. However, this may be associated with higher costs, e.g. if the update is not then authorized or if the OEM bears the costs of the mobile radio transmission itself. Instead of the customer who has purchased a corresponding option from the OEM or provides the mobile connection through his own mobile hotspot.

An update can also be distributed in multiple 'waves' to the vehicles to distribute access to the server infrastructure over time. In the case of updates that are not time-critical, the functionality in the field can also be checked first.

### Note on simulation

Several slots are available on the server side for the vehicles to be updated, from which the update can be retrieved in parallel (`nDownloadSlots`). The decision to successfully download an update after occupying a free slot depends on a probability factor (`pSuccessDownload`). This in turn depends on the customer group and the technology (WiFi has a significantly lower coverage and a slower connection speed, so that, for example, the connection is interrupted or no network at all is available as a result of the customer continuing his journey). If the update is successful, this will be noted in the database. The more slots are available for download, the faster the update can be downloaded - but at the same time costs are rising. The option to limit the number of slots is also used in the simulation to divide the bandwidth for several parallel updates and to optimize it according to the prioritization of the updates.

If the download fails, the download will be tried several times at this time (`maxDownloadAttempts`). If this number is exceeded, a further download is not attempted at first. The vehicle reference is buffered on a stack according to the First-In-First-Out logic. After a specified period of time, the stack accesses a slot for the next download attempt (`duraNextDownload`). After a maximum number of download attempts, the vehicle reference is removed from the OTA process (`maxDownloadAttempts`). The update must be carried out either by a workshop or manually.

Regardless of success, the duration of a download attempt consists of the transmission time depending on the transmission technology (3G, 4G, 5G or WiFi) and the on-board and server-side performance time (`duraDownloadIdle`). This time is needed on both sides of the transmission channel to process the update and also validate it in the vehicle.

The simulation also implements the option for customers to transfer the update to the vehicle via tethering: either by using their mobile phone as a mobile hotspot for data transfer or by downloading the update to a USB stick via a PC or laptop and then connecting it to the car. Opening such an interface creates additional risks, as the customers' tethering devices may not have been adequately secured and may contain malicious software that could also threaten the vehicle after the device has been connected. The protection of such untrustworthy devices is to be particularly considered with such an offer.

### OTA-V-150 [new]

<u>Input</u> : Status message	
The <b>digital service provider</b> monitors the download activities with regard to possible bottlenecks, error messages or other feedback.	
R	Digital service provider
A	Digital service provider
C	Vehicle owner, Vehicle

---

I	Update coordinator, (regulatory authority)
<u>Output:</u> analytical report	

**Explanation**

A sustainably implemented monitoring activity is important in order to be able to react in time to irregularities in the update process. In the best-case scenario, a serious error will affect significantly fewer customers.

### 3.7 Update installation

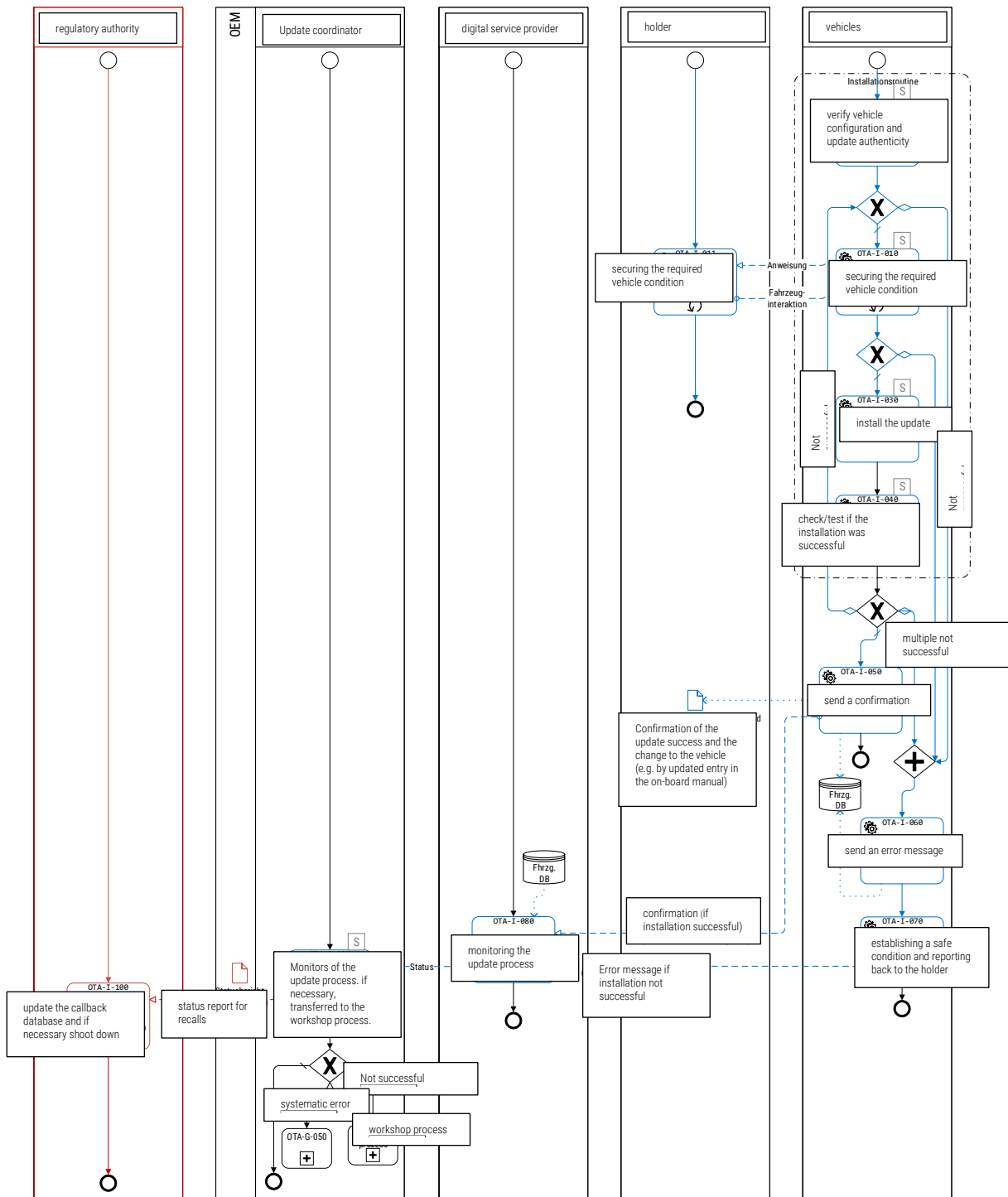


Figure 11: Process modeling "installation"

**OTA-I-020**

<u>Input</u> : update stored locally in the <i>vehicle</i>
The <i>vehicle</i> checks whether it meets the requirements and vehicle configuration for the update (memory space, necessary software libraries, etc.) and verifies the authenticity of the update.
R      vehicle A      Development manager C I
<u>Output</u> : Confirmation of the correct vehicle configuration; request for further software packages if necessary
<b>Either</b> the requirements are met (→ OTA-I-030) <b>or</b> the verification fails. If the output is negative, an error message is sent to the backend (→ OTA-I-060) <b>or</b> prerequisites are missing that can be fulfilled by reloading software. This means only a delay and no termination of the process.
<b>Risk-E040</b> : A software update is installed that is not compatible with the vehicle, resulting in a loss of functionality. <b>Risk-E041</b> : The update cannot be installed due to lack of operational resources or it aborts during installation so that the vehicle remains in an undesired state. <b>Security risk E042</b> : Unauthorized, manipulated or counterfeit software is released for installation.
duraCheckConfig: Duration required by the vehicle to check the vehicle configuration and respond to the server. (analog OTA-V-060) pSuccessConfig: Probability that vehicle configuration is correct. (analog OTA-V-060)
UN ECE REQUIREMENT: 7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.

**Explanation**

If necessary, the vehicle may lack the standard software libraries required for the update, which must first be loaded and installed before the update can be carried out. This in turn would correspond to a separate update process and is therefore not considered further.

If the vehicle has an unsuitable configuration<sup>11</sup>, e.g. ECUs other than those intended for the update, the update is marked as failed and reported to the backend accordingly.

**OTA-I-010**

<u>Input</u> : update stored locally in the <i>vehicle</i> and confirmation of correct vehicle configuration
--

<sup>11</sup> The difference between the current vehicle condition and a previous configuration request may be due to an accident, for example.



<p>The <b>vehicle performs</b> a routine to ensure the required vehicle condition (e.g. battery SoC, ignition, window position, gear, etc.). If a criterion needs to be operated by a user or if a criterion is not fulfilled, this is communicated to the vehicle occupant and checked again after confirmation. If the customer cannot be reached, the update must be postponed.</p>	
R	vehicle
A	development manager
C	
I	vehicle owner
<p><u>Output</u>: Confirmation of the correct condition, instruction to the customer or error message</p>	
<p><b>Either</b> the required vehicle condition can be successfully established (→ OTA-I-020) <b>or</b> it requires the support of a user (OTA-I-011) <b>or</b> the routine is not successful. If the output is negative, an error message is sent to the backend (→ OTA-I-060).</p>	
<p>If safety critical:</p> <p style="padding-left: 40px;">a) then the highest requirements for the condition of the vehicle. If applicable, this is for security-relevant Updates not required if no safety-relevant areas are addressed.</p> <p>If time-critical:</p> <p style="padding-left: 40px;">b) then the driver must often be reminded of the necessary condition of the vehicle at a high interval.</p>	
<p>maxChecksPerSession: maximum number of attempts of a status check per time point</p> <p>maxSessionsPerDay: maximum number of examination times per day</p> <p>maxChecksPerVehicle: maximum number of condition checks until the process is terminated</p> <p>pSuccessVehicleState: probability that the condition of the vehicle is correct</p> <p>duraCheckState: Duration for checking the condition of the vehicle</p>	
<p>UN ECE REQUIREMENT:</p> <p>7.1.4.1. The vehicle manufacturer shall demonstrate the processes and procedures they will use to assess that over the air updates will not impact safety if conducted during driving.</p> <p>7.2.2.1.2 The vehicle manufacturer shall ensure that software updates can only be executed when the vehicle has enough power to complete the update process (including that needed for a possible recovery to the previous version or for the vehicle to be placed into a safe state).</p> <p>7.2.2.3. In the situation where the execution of an update whilst driving may not be safe, the vehicle manufacturer shall demonstrate how they will:</p> <ul style="list-style-type: none"> <li>▪ Ensure the vehicle cannot be driven during the execution of the update;</li> <li>▪ Ensure that the driver is not able to use any functionality of the vehicle that would affect the safety of the vehicle or the successful execution of the update</li> </ul>	

### Note on simulation

The number of attempts to test the vehicle condition depends strongly on the user profile. In the simulation, this can be addressed via probability. In addition, the probability is determined by the robustness of the activity itself and the requirements of the update. In the case of an installation routine that was triggered, for example, by an app, the user is not necessarily on the vehicle at the selected installation time, so that a faulty status parameter

**OTA-I-011 [new]**

<u>Input</u> : statement	
The <b>vehicle owner</b> shall take the measures required by the vehicle to safeguard the condition of the vehicle.	
R	vehicle owner
A	update coordinator
C	
I	vehicle
<u>Output</u> : vehicle interaction	

**Explanation**

This activity by the customer is listed separately as it is an uncertainty factor that can be reduced by appropriate communication with the customer. This includes the simplicity of the language and the statements in in-vehicle infotainment.

**OTA-I-030**

<u>Input</u> : update stored locally in the vehicle, confirmation of correct vehicle configuration and correct vehicle condition	
The <b>vehicle</b> decrypts, unpacks and automatically installs the update.	
R	Vehicle
A	Development manager
C	
I	Vehicle owner
<u>Output</u> : Signal that the installation has been completed.	
<p><b>Chance-E043:</b> The installation takes place independently without any necessary personnel and without corresponding costs.</p> <p><b>Risk-E044:</b> The installation is aborted (e.g. by a change in the vehicle condition) and the vehicle remains in an undesirable system condition.</p> <p><b>Security risk E045:</b> After decryption of the update on the gateway, the update and know-how can be read on the CAN-BUS, for example.</p>	
<p>powerECU: Performance of the used computer unit</p> <p>pSuccessInstall: probability that the update will be installed successfully</p> <p>duraNextInstallation: Duration until the next installation (after a failed one) is tried.</p> <p>maxInstallsPerVehicle: maximum number of installation attempts</p>	

## Explanation

The performance of the target ECU and the package size have a significant influence on the duration of the activity. Especially the flashing of less powerful ECUs can take longer. One measure, for example, would be to have decryption performed by a more powerful CPU, such as the infotainment CPU. A disadvantage, however, would be that the update package would then be sent unencrypted to the target ECU and can thus be read in the on-board network.

## OTA-I-040

<u>Input</u> : Signal that the installation has been performed.	
The <b>vehicle</b> checks / tests whether the installation was successful. Corresponding test mechanisms are to be implemented, which check the installation routine and if necessary, afterwards the changed functionality.	
R	Vehicle
A	Development manager
C	
I	
<u>Output</u> : Test result of the installation	
<b>Either</b> the installation was successful <b>or</b> the installation failed. If an error occurs, you can first try to repeat the installation routine (→ OTA-I-010). If the installation fails several times, an error message is generated (→ OTA-I-060).	
If safety-relevant: <p style="margin-left: 40px;">a) a test procedure for safety-relevant functions shall be recommended to check whether the functions described by the functions affected by the installation also work as intended.</p>	
<b>Risk E046</b> : After the installation or an interruption of the installation, a fault is detected so that the vehicle is no longer ready to drive. The vehicle must be returned to a roadworthy condition.	
<b>Risk-E046b</b> : After installation, the installation will <u>not be</u> stopped or an error will not be detected, which may result in damage due to a malfunction.	
duraCheckInstall: Duration required to perform the check.	
UN ECE REQUIREMENT: 7.2.2.1.3 When the execution of an update may affect the safety of the vehicle, the vehicle manufacturer shall demonstrate how the update will be executed safely. This may be achieved through technical means and/or through a process that will require the vehicle user to provide verification that the vehicle is in a state where the update can be executed safely.	

## Explanation

The test procedure can be graded according to the systems involved in order to make testing more efficient. For example, infotainment systems can be sufficiently tested during development so that only the successful installation has to be verified. While safety-relevant systems should also be

tested in the vehicle in order to confirm their functionality after installation. In such a case, it can be tested, for example, whether the actuators can be addressed or whether a beta test is first carried out in the field, i.e. that the new software runs parallel to the previous software version and the results of the new software version are verified.

### OTA-I-050

<u>Input</u> : Test result of the installation	
The <b>vehicle</b> sends a confirmation of successful installation to the vehicle database / backend and to the user.	
R	Vehicle
A	Development manager
C	
I	Vehicle owner
<u>Output</u> : Confirmation (e.g. updated database entry or message in IVI)	
<p><b>Risk-E047</b>: The change to the vehicle is not documented, so that a negative customer experience occurs when the vehicle is used or resold, e.g. because a function behaves differently than expected or before the update.</p> <p><b>Security risk E048</b>: The successfully performed update is denied (e.g. this can lead to a failure of payments with additional features).</p> <p><b>Risk-E049: Due</b> to a failure to provide feedback on the success of the update, the sale of a service or a recall cannot be documented as completed.</p>	
<p>UN ECE REQUIREMENT:</p> <p>7.2.1.2.1 Each RXSWIN shall be uniquely identifiable. When type approval relevant software is modified by the vehicle manufacturer, the RXSWIN shall be updated if it leads to a type approval extension or to a new type approval.</p> <p>7.2.2.4. After the execution of an update the vehicle manufacturer shall demonstrate how the following will be implemented:</p> <ul style="list-style-type: none"> <li>▪ The vehicle user is able to be informed of the success (or failure) of the update;</li> <li>▪ The vehicle user is able to be informed about the changes implemented and any related updates to the user manual (if applicable).</li> </ul>	

### OTA-I-060

<u>Input</u> : Test result of the installation	
The <b>vehicle</b> sends an error message about the failed update to the vehicle database / backend and possibly to the user.	
R	Vehicle

A	Development manager
C	
I	Vehicle owner
<u>Output:</u> Error message (e.g. updated database entry)	
<b>Risk-E050:</b> Error message is not sent or sent with wrong content, so that the OEM is not informed about occurred errors.	

**OTA-I-070**

<u>Input:</u> Error message (e.g. updated database entry)	
The <b>vehicle</b> tries to restore a safe condition and communicates this and the result of the attempt to the driver / keeper.	
R	Vehicle
A	Update coordinator
C	
I	Vehicle owner, development manager
<b>Risk-E051:</b> The vehicle cannot be switched to Fail/Safe mode, so that the vehicle may no longer be usable by the driver.	
<b>Risk-E051b:</b> The vehicle loses connection to the backend and the vehicle remains in Privacy Mode - a mode in which the communication to the OEM is deliberately turned off by the customer. A fail/safe mode with sufficient functionality must be ensured.	
<u>Output:</u> (supplementary) error message	
UN ECE REQUIREMENT: 7.2.2.1.1 The vehicle manufacturer shall ensure that the vehicle is able to restore systems to their previous version in case of a failed or interrupted update or that the vehicle can be placed into a safe state after a failed or interrupted update.	

**Explanation**

The goal is to keep the restrictions for the customer as low as possible and to make all functionalities available again after an error. It should be considered whether the corresponding function should be deactivated for security reasons in the case of security-relevant errors that have failed to be corrected. The minimum objective is to bring the vehicle into a running condition. Then the normal condition in the workshop must be restored.

**OTA-I-080**

<u>Input:</u> Confirmation or error message
---

The <b>digital service provider</b> monitors the progress of the update process. This includes the monitoring of the process flow during the entire update distribution as well as the monitoring of the installation routine and the handling / escalation with possible error messages of individual vehicles.	
R	Digital service provider
A	
C	Update coordinator
I	Development manager
<u>Output:</u> Status of the entire process / bug report	
<b>Risk-E052:</b> (Systematic) errors in the update or due to an installation routine can only occur at the customer's site in the field without direct support from a workshop employee being possible. This can result in high restrictions on the performance of the vehicle vis-à-vis the customer.	

### OTA-I-090

<u>Input:</u> Status of the entire process / bug report	
The <b>update coordinator</b> monitors the update process and coordinates the escalation of possible errors in the process.	
R	Update coordinator
A	
C	Development managers
I	Regulatory authority
<u>Output:</u> Status report (only for callbacks)	
<p><b>Conditions:</b></p> <p>(A) If necessary, the update must be followed up in the workshop process (→ WRK-V-010).</p> <p>(B) The update may contain a systematic error. In this case, the distribution must be stopped and an updated update generated (→ OTA-G-050).</p> <p>(C) <b>In the</b> case of an ongoing recall, the <i>update coordinator shall</i> forward a status report to the <i>Regulatory Authority</i> at agreed time intervals.</p>	

### OTA-I-100 [reg. auth. only]

<u>Input:</u> Status Report	
The <b>Regulatory Authority</b> shall update the recall database and, where appropriate, complete the recall in accordance with its respective procedures.	
R	Regulatory authority
A	

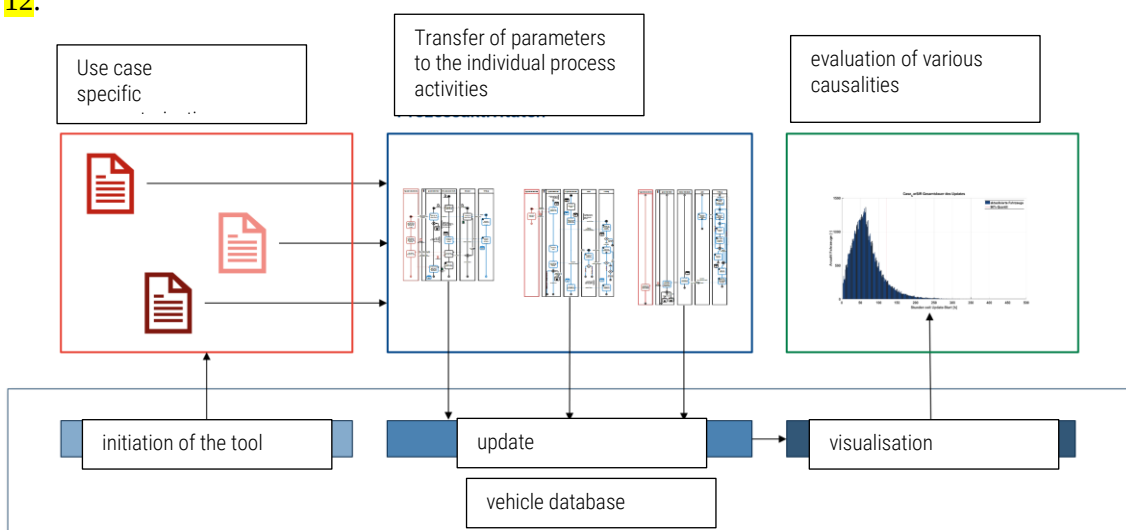
C
I      Update coordinator
<u>Output</u> : status report

## 4 SIMULATION

For a detailed analysis of the process and deepening of the contents, a simulation was programmed for OTA updates within the scope of this project. This simulation is mainly a "means to an end" to generate knowledge, e.g. by modelling individual activities, and to validate the process.

### 4.1 MATLAB tool

The simulation was implemented using MATLAB and has the logical sequence shown in [Figure 12](#).



**Figure 12: Logical structure of the simulation tool**

In the following, the individual steps illustrated in [Figure 12](#) are explained individually.

(1) First, the simulation tool is initiated. This also includes reading in the vehicle data. These data identify the target vehicles via the license plate and describe the technical equipment of the vehicles regarding the performance of the connectivity unit and the target ECU. The corresponding database is the basis, which is also supplemented by the results of the simulation (represented by the blue horizontal box below). The most important dimension in the simulation is the duration that the update requires over all process steps ( $dura_{Total}$ ).



(2) In the second step, the simulation user must parameterize the process and adapt it to the framework conditions. There are 42 parameters available for this purpose, the

- the technology (built-in mobile radio technology, computing power of the ECU, etc.),
- the relationship between customer and OEM (at what intervals the customer is informed, how often the update may be rejected, etc.) and
- the customer himself (how often does the customer reject the update, when is the customer at the vehicle, etc.)

describe.

Distribution functions were integrated into the simulation, above all in order to map customer behavior more realistically. These can also be parameterized in order to better map the respective customer group via the center of gravity, the aspect ratio and the density of the distribution.

(3) The tool now gradually transfers these parameters to the programmed activities of the idealized OTA process described in Chapter 3, calculates for each vehicle a specific lead time for each of these activities depending on the technology, the driver and the infrastructure and adds these values to a total lead time.

A total of 13 of the 26 mandatory activities and two optional activities were programmed. No critical influence on the lead time was assumed for the other activities, as they either take place in parallel or can be considered in parameters of other activities.

(4) After each activity, the vehicle database is updated accordingly to the duration of the update per activity and vehicle and to the success / failure of the activity.

(5) In preparation for the final analysis, the data shall be processed and visualized.

(6) Based on the simulation results, various causalities along the process chain can now be analyzed manually. It should be noted that the influence of some factors is degressive (contrary to the expectation of a linear influence) or there are dependencies between parameters, as shown below.

## 4.2 Use Case comparison

For the execution of this simulation, three different application cases were distinguished, each of which entails a significantly different parameterization. In the following, the application cases are first described in tabular form. This information corresponds to the information that could form the basis of an initial update requirement message that initiates the OTA update process.

It should be noted that the results here are purely exemplary and do not make any binding statements. Rather, they are used to examine the effect of process and parameter variations on process performance with the same basic assumptions. There was a lack of quantifiable comparative data for comparison with reality. These could have been used in the first step to map a completed update process in order to subsequently optimize it. Because this image of reality was missing, some well-founded assumptions had to be made.

#### 4.2.1 Car-Security-Incident-Response (Car-SIR)

<b>name</b>	Update to fix a security problem in the Bluetooth stack.
<b>trigger</b>	Security researchers discover a vulnerability in the Head Unit, allowing malware to be installed on the Head Unit.
<b>Technical description</b>	Due to the presence of an error in the Bluetooth stack, malware can be installed on the head unit of a manufacturer. The Head Unit is connected to the system CAN bus. At the moment, it does not use any dedicated safety mechanisms, so that other ECUs can be put into diagnostic mode via the Head Unit and can be moved to download a modified firmware update. The modified firmware updates can be used to generate safety-critical CAN signals, thus influencing critical vehicle functionality.
<b>Affected Software/ECUs</b>	Head Unit with faulty Bluetooth stack (Blueborn gap) Head Unit with malware Possible control units with modified firmware
<b>actions</b>	Software update on the Head Unit so that a bug-fixed Bluetooth stack is used. If necessary, remove malicious code from the head unit (if necessary, reinstall the entire software). If necessary, update the compromised firmware on other ECUs.
<b>Legal framework</b>	Safety gaps that can lead to critical functionality being impaired in the vehicle must be reported and recalled.

Initially, the Use Case Car-SIR was parameterized. This was based on a time-critical, mandatory security update of 10 MB in size. This took around 16 hours to arrive at the CDN and was distributed to 100,000 vehicles via WiFi and mobile radio using the shoulder-tap method. The target system, which addresses the update, is the head unit, whereby small restrictions for the end user are assumed, since a high computer performance is available and no primarily safety-relevant system is addressed. This was considered in the simulation by a higher probability for the approval of the update by the customer.

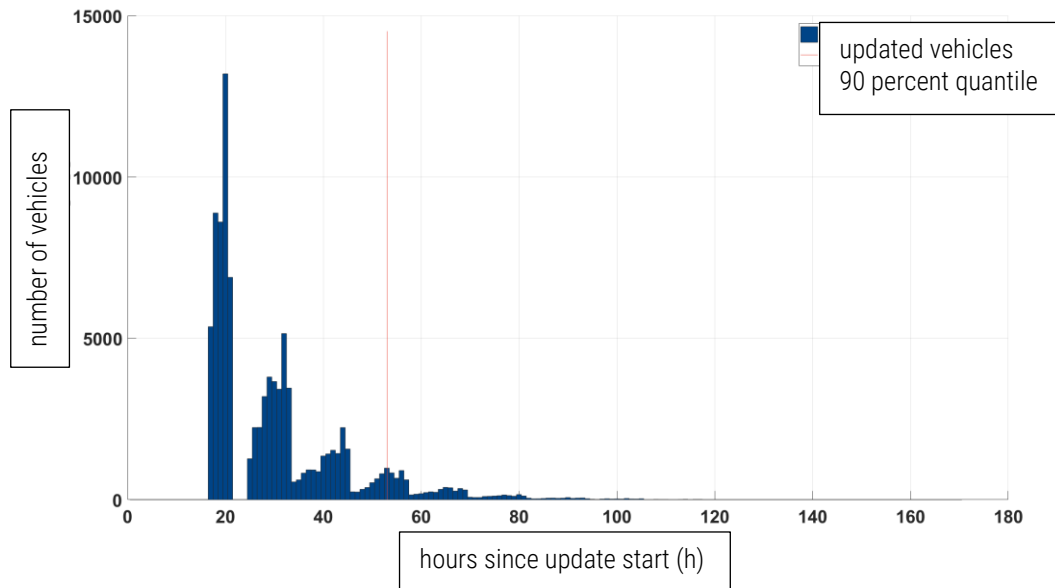


Figure 13: Use Case Car-SIR - lead time per vehicle

Figure 13 shows that 90% of the vehicles were updated after about 55 hours. Nevertheless, some vehicles have only been updated after 170 hours and thus after more than three times that time. Only successful vehicles are included in the graph. In the end, about 3.3% of the updates failed with this update, as shown in the Figure 14

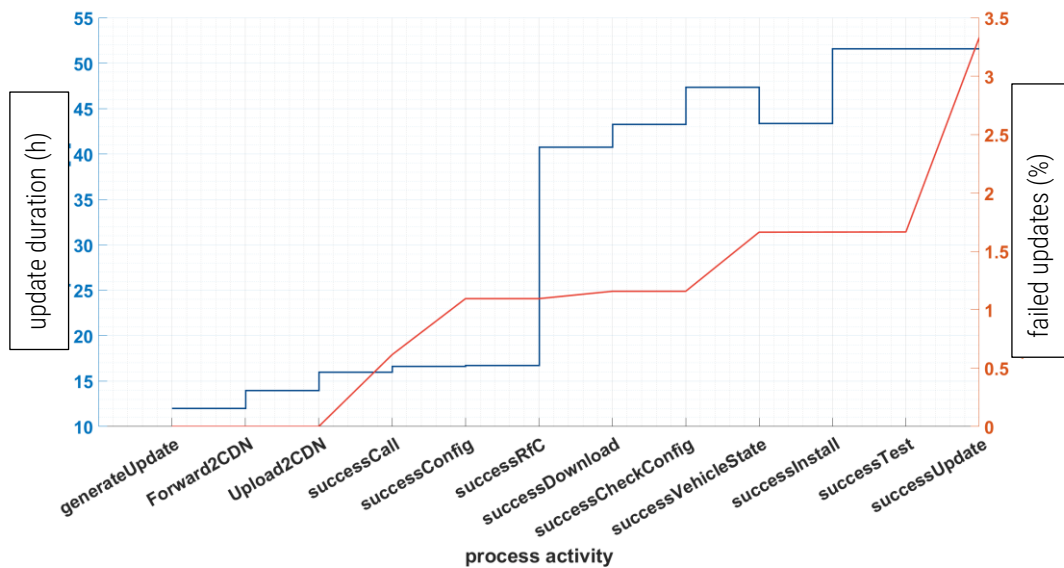


Figure 14: Use Case Car-SIR - lead time per activity

This graph also shows the duration of the individual activities in relation to each other. The higher the level between two activities (process activity), the longer the process step takes. Due to the small update size and the transmission via mobile radio, the activities are no bottlenecks that describe the transmission or the automatic, technical processing of the update. Rather, it is the confirmation of the installation by the customer that delays the throughput time (*successRfC*).

This can have several reasons: on the one hand it can be simply that due to the short total throughput time the users simply were not at the vehicle or on the other hand that the update was suggested at an unfavorable time.

For the milestone *successVehicleState* there is a logical error in the display, because the update time decreases here. This is due to the increase in failed updates to this activity. These are omitted in the following step in the calculation, since the database of the illustration is based only on the values of successful vehicles. These failed vehicles probably include vehicles with a noticeably higher update time than the average. If these are omitted now, the average update time decreases due to the missing outliers with very high update duration.

#### 4.2.2 Updating the software in infotainment

<b>name</b>	Updating the software in infotainment
<b>trigger</b>	An automobile manufacturer updates its infotainment system to a new major version. The starting point was customer feedback for easier operation, improved efficiency and compatibility with third parties.
<b>Technical description</b>	Compared to the competition, customers perceive the system as "sluggish" and lack important functions. In addition, smartphones are no longer compatible with the latest operating system version (e.g. due to a new transmission standard).
<b>Affected Software/ECUs</b>	infotainment system
<b>Vehicles concerned /Vehicle classes</b>	Several mid-range and luxury vehicles from one manufacturer
<b>actions</b>	On the one hand, the menu navigation is changed, new functionalities are added, the software is "cleaned up" and adapted to the latest versions from third-party providers such as Apple, Android or Spotify.
<b>Legal framework</b>	

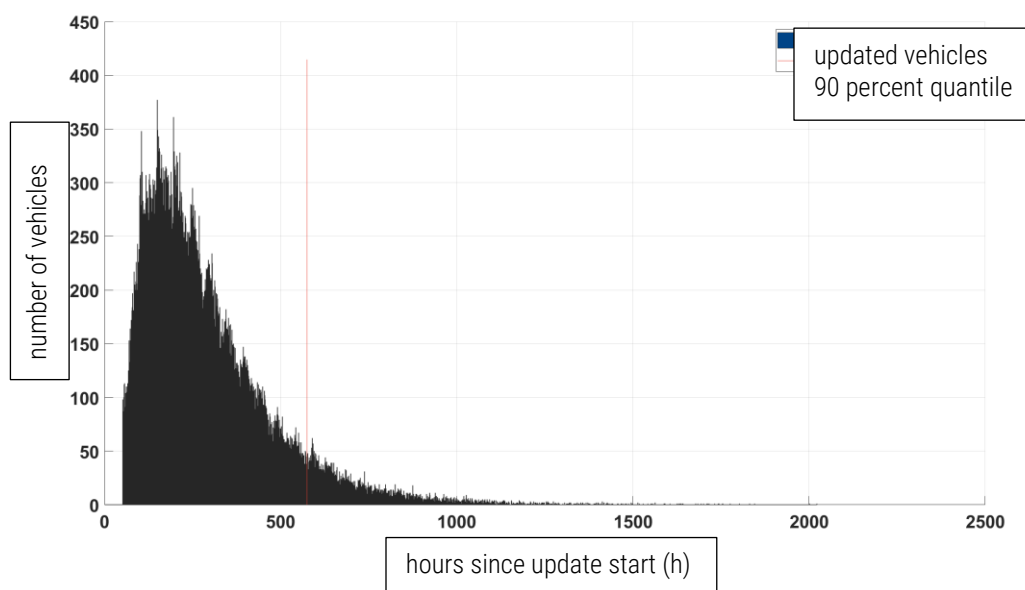
Due to the system architecture and the scope of the new functions, an update package with a size of 1000 MB was simulated. As the update provides added value for the customer and is not a troubleshooting, there is no transfer at the expense of the OEM and a high percentage of transfers via WiFi or USB tethering is assumed (70%). The other 30 % are charged via mobile radio by the vehicles whose owners have concluded a corresponding mobile radio option, which is available as optional equipment for the described mid-range vehicles.

However, due to the update size and the high percentage of WiFi transmissions, the likelihood that the vehicle will be reached, and the update will be downloaded successfully and completely in one session decreases. In addition, reaching customers is delayed by the lack of a mobile phone

connection, as additional channels must be selected to communicate the update to the customer (e.g. newsletters, apps, etc.).

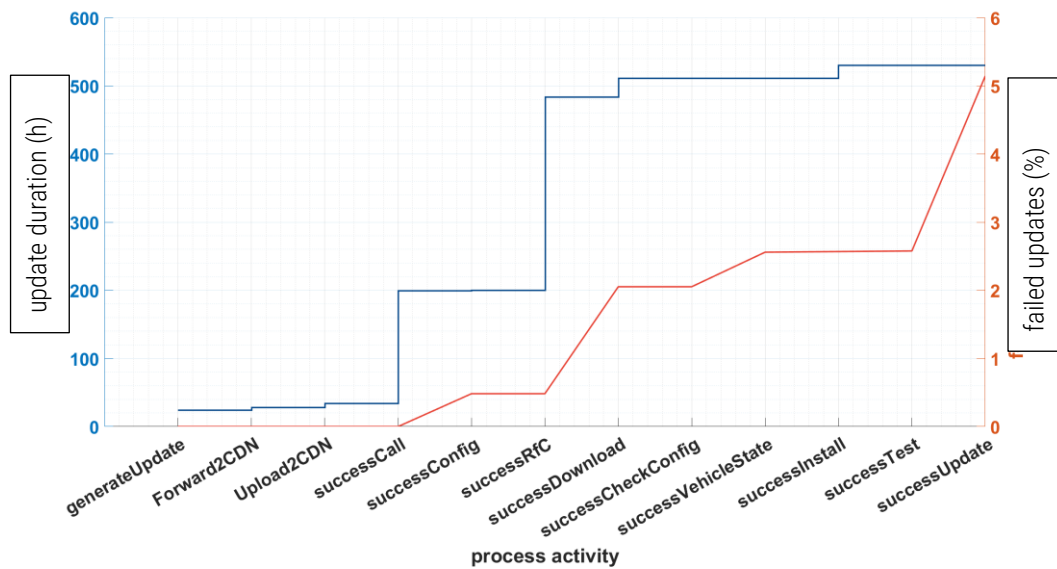
Since such a delay can be assumed from the outset and because of the lower priority of the update, the maximum efficiency for the servers was selected in such a way that an appropriate, small number of server slots was provided. In addition, a high probability for the existence of a correct vehicle configuration was assumed ( $p_{SuccessConfig}$ ), since there are no interdependencies of the infotainment system with other ECUs regarding the installation. As a result - especially since the update is not safety-relevant - there are also fewer requirements for the condition of the vehicle that has to be manufactured. The lower requirements are considered in the simulation as a higher probability that the user / vehicle will successfully establish this state ( $p_{SuccessVehicleState}$ ).

The probability that the update will be accepted ( $p_{SuccessRfC}$ ) is rather low, since the update is not critical (it does not fix any bugs) but at the same time represents a higher effort for the customer. The installation and download time are high due to the package size. This effect compensates the disadvantage of the low download success somewhat, since the download can be started / continued several times up to the authorization of the update. When the update is finally authorized, it can be assumed that the download has already taken place. However, if the vehicle is moved in the meantime ( $p_{SuccessInstall}$ ), the installation process may then be interrupted due to the length of the installation process.



**Figure 15: Use Case Infotainment - lead time per vehicle**

To achieve the 90 percent quantile, it takes about 550 hours with the same number of vehicles in this use case. While in the Car-SIR use case around 10,000 vehicles were updated per hour in the initial phase, this figure is only a maximum of 375 vehicles.



**Figure 16: Use Case Infotainment – lead time per activity**

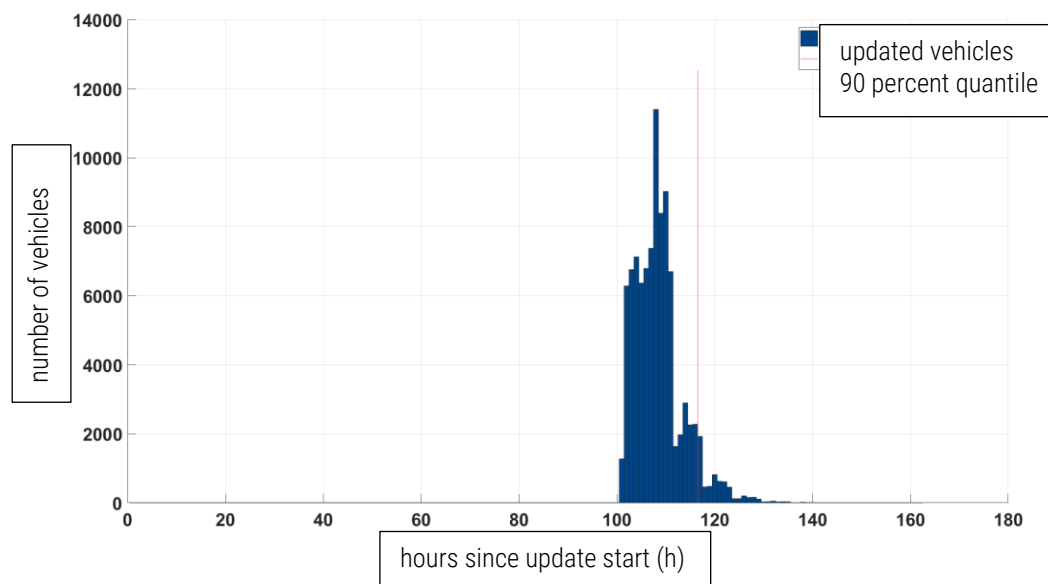
Most of the time during the update process is spent notifying the vehicle or customer because of the high percentage of WiFi transmissions. Afterwards a lot of time passes until the update is authorized by the customer. The actual download itself contributes relatively little to the overall runtime.

### 4.2.3 Recall of software in the field on KBA instruction

<b>name</b>	Recall for troubleshooting in the airbag
<b>trigger</b>	An automobile manufacturer finds that a software module in a supplier's sensor is faulty and that under certain conditions there may be delays in triggering the side airbag.
<b>Technical description</b>	The software of the side airbag sensor (acceleration) is faulty, so that under certain boundary conditions data is only provided late, so that in the event of an accident there is a delay in the deployment of the side airbag.
<b>Affected Software/ECUs</b>	Side airbag sensor (acceleration)
<b>Vehicles concerned /Vehicle classes</b>	Several mid-range vehicles from several manufacturers
<b>actions</b>	Error correction in the software module by the supplier and application of the error-corrected software by the OEM.
<b>Legal framework</b>	Airbags are part of the safety system (occupant protection) in the vehicle. ECE R114 regulates the replacement of airbag systems

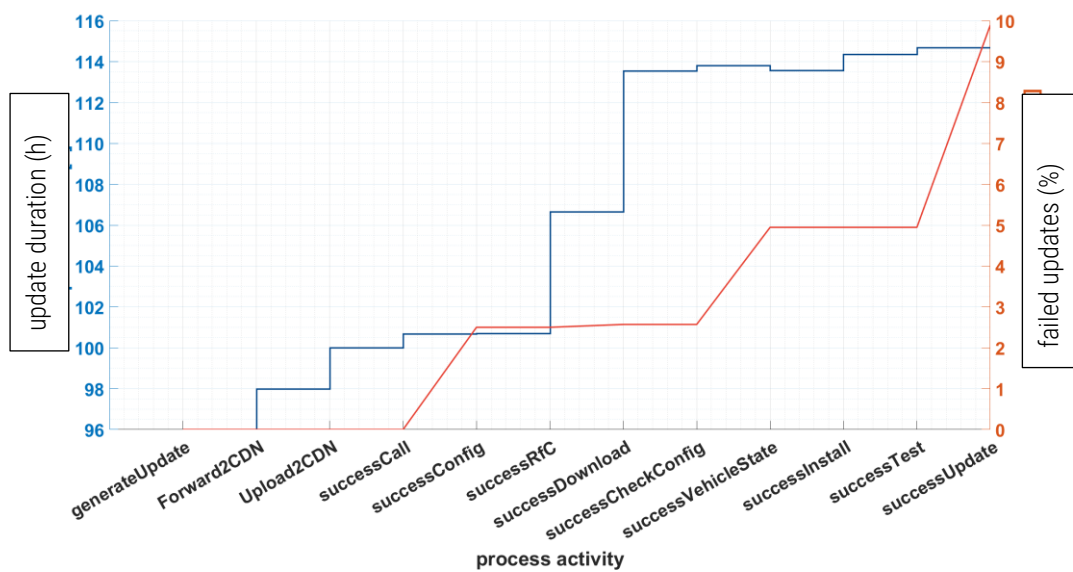
The third use case describes a recall in the field due to a faulty side airbag sensor. This is a safety-relevant, authority-relevant, time-critical and mandatory update, for which a fictitious deadline of seven days has been set until the end of the OTA update.

The update package is 10 MB in size, 80% of it is distributed via mobile phone and shoulder tap. Due to the urgency of the situation, an attempt is made to reach the vehicle every 6 hours - with correspondingly high server capacities. It is assumed that this reduces the probability that the vehicle will be reached for each of these requests. In addition, it is assumed that the configuration is more often faulty, as it can be assumed that the airbag or control unit was replaced after an accident and this was not properly communicated / digitized. On the other hand, high probabilities were assumed for accepting the update, establishing the correct condition for installation and a successful installation. On the one hand because the user himself could be at risk if he does not install the update and on the other hand because only a single ECU is affected, so that the installation itself has no high requirements.



**Figure 17: Use Case Airbag - lead time per vehicle**

This example shows a long lead time that has arisen due to the complexity of supplier relationships. Software and sensor are not produced by the OEM or Tier1- supplier, but by a supplier company that does not have a direct contractual relationship with the OEM. After the demand message, the problem is escalated in the supply chain. The assumptions made here nevertheless allow a distribution to a large number of vehicles within the fictitious seven-day period.



**Figure 18: Use Case Infotainment - lead time per activity**

However, the disadvantage is a high number of failed updates, as shown in the **Figure 18**, due to the incorrect configuration, the incorrect vehicle condition or inadequate verification of the installation. Furthermore, this use case shows that this time the success of the download has a significant influence on the processing time.

#### 4.2.4 Conclusion

All three use cases presented are exemplary and not simulated based on real data. Nevertheless, logical assumptions have been made to compare the use cases with the simulation.

It should be noted that in all three cases the confirmation of the update by the customer has a significant influence. The design of the notification about the update is therefore an important factor in optimizing the processing time of the OTA update. Other important activities include sending the message to the vehicle itself, the download process and the configuration check. The influence of the individual parameters is examined in the following section.



## 4.3 Process optimization

Using the simulation tool, the OTA process was iteratively optimized for the use case "Car-Security-Incident-Response (Car-SIR)". The influence of each parameter variation on the overall result was considered step by step. The ideal goal for a security gap must be to react within 24 hours with a security relevant patch and close the gap. In Section 4.2.1, this use case was initially parametrized, with a lead time of approximately **216 hours or 9 days**.

Starting from this initial use case, the individual parameters were now varied step by step in order to examine their influence on the total time. Only one parameter is varied at a time. The most efficient value is then selected for this parameter. With this optimum, the process is recalculated, and the next parameter is varied on the basis of this intermediate result.

The target parameter is the 90% quantile of the total update duration, i.e. the time at which 90% of the vehicles have been successfully updated. For the baseline scenario, this value is around **216 hours or 9 days**.

### 4.3.1 Static influencing factors

Some parameters contribute as constant factors to the duration of the update process. They were nevertheless added to the simulation to draw attention to optimization potentials within these activities. This can be achieved by increasing internal efficiency or by improving communication between stakeholders. These influencing factors include the following parameters:

- Duration for generating the update,
- Duration for communicating the update to the digital service provider,
- Duration for uploading the update to a CDN, etc.

In addition, further technical, constant influencing factors have been implemented, some of which have an influence on the duration that the vehicle is not available to the customer, e.g. during installation:

- Duration for processing a server request,
- Duration to check / validate the update before installation,
- The time it takes to restore the vehicle to the correct condition,
- Duration for checking the success of the installation routine, etc.

### 4.3.2 Dynamic influencing factors

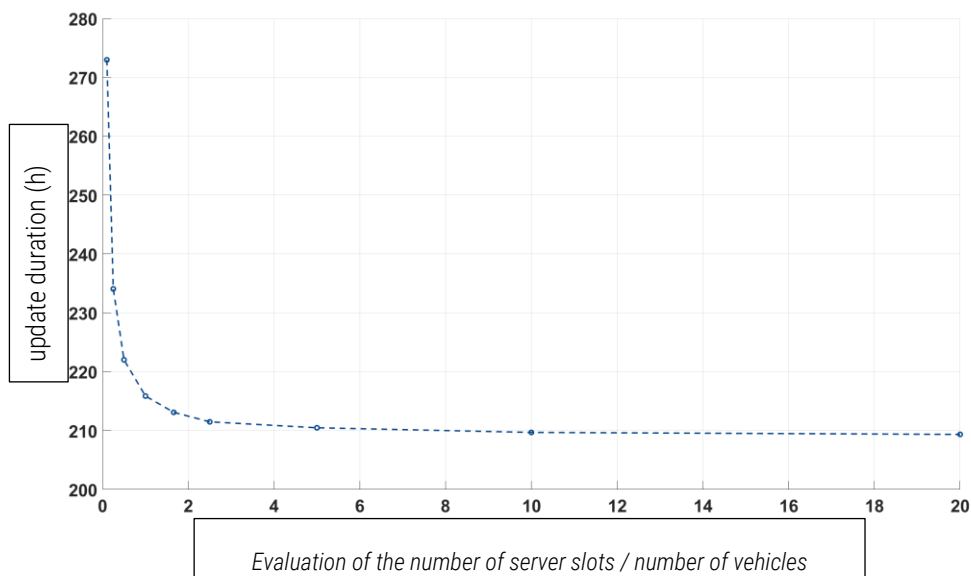
In addition, there are influencing factors that can be varied for each update by the update coordinator. These dynamic factors and the influence of their variation are examined below.

### Number of server slots / number of vehicles

First, the number of vehicles was varied, and the number of available server slots considered.

If this number of server slots remains absolutely constant, the duration of the update increases with the number of vehicles.

If one keeps the number of vehicles constant and varies the number of server slots, a degressively decreasing trend can be observed. Which is independent of the absolute magnitudes but vary according to the ratio of the number of server slots to the number of vehicles - see **Figure 19**.



**Figure 19: Evaluation of the number of server slots / number of vehicles**

This means that an ever-increasing increase in server capacity ultimately only has a minimal impact on the throughput time of the update. Assuming that rising capacities also cause rising costs, this is no longer economical as soon as the curve is almost horizontal. From an efficiency point of view, the optimum is to be found in the bending. Here the curve begins to sink more and more slowly and every investment in additional capacities has an ever decreasing effect on the update duration.

In order to optimize this use case in relation to the update size and urgency, an ultimately efficient but also as effective as possible distribution of the update is envisaged and therefore a ratio of 2.5 percentage points (previously 1 percentage point) at the end of the bend is selected for the further analyses, as the horizontal course begins here. In order to optimize the computing time, 100,000 vehicles are calculated, since - as shown - only the relative ratio of both characteristic values and not the absolute values are decisive for the duration.

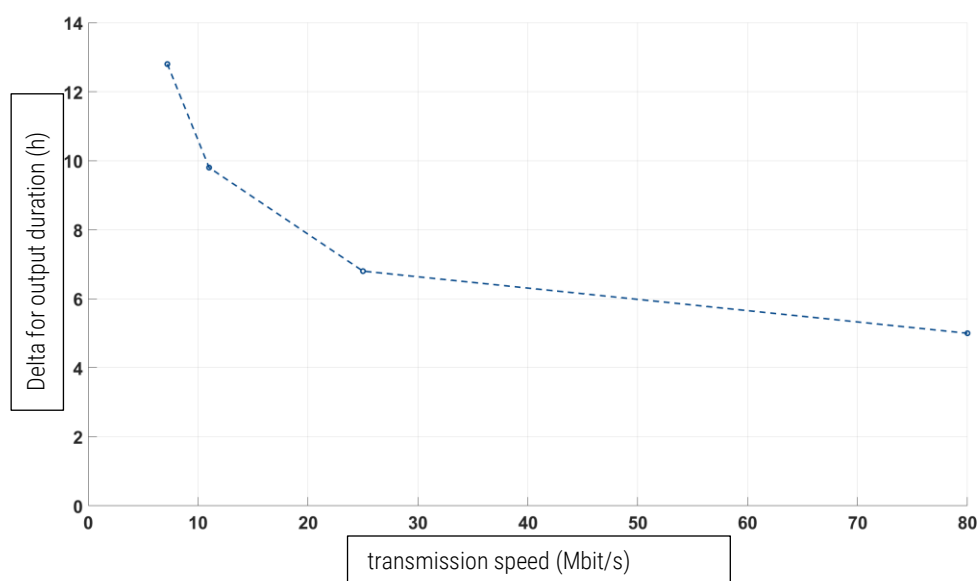
This adjustment reduces the execution time for the entire update from 216 hours to 211.5 hours.

The allocation of capacities to updates and thus their prioritization can have a further influence if several updates are distributed over an infrastructure. However, this was not considered within this project.

### Distribution of transmission technology

In the simulation, it is determined for each vehicle on the basis of a previously defined distribution which transmission technology is used to install the update on the car. This includes transmission via WiFi, 3G or 4G. Alternatively, this can also be done by tethering, i.e. the customer saves the update on a data carrier and then connects it to the car or makes his smartphone available as a mobile hotspot via which the vehicle can download the update.

If this distribution is varied, it can be seen - related to the respective delta between the total processing times - that the transmission speed has a degressive effect on the total processing time. If the transmission speed is halved, the delta does not double, but only increases damped - see **Figure 20**.



**Figure 20: Evaluation of the distribution of the transmission technology**

A similar effect can be observed with variation of the update size. The delta to the output duration is increased accordingly for each transmission speed by a factor. A parallel displacement of the curve shown in the **Figure 20**

Here dependencies to other parameters predominate. An isolated view is not sufficient here. It can be assumed that transmissions of large packets via WiFi are more likely to stop or be interrupted, as the customer does not necessarily ensure that the connection to WiFi is maintained until the end of the download. While a moving vehicle can also be addressed via mobile radio and the success of the download is ensured by the much higher network coverage.

For the further course of the analysis, the share of WiFi and tethering is set at 10% each, since due to criticality the 10 MB update is distributed via mobile radio (also at the manufacturer's own expense). The assumption, however, is that some customers are still in WiFi or have activated a private mode and thus do not allow a radio connection between the vehicle and the backend. Due to the regional distribution of target vehicles, it is assumed that a further 30% will receive the

update via 3G and the remaining 50% will receive the update via 4G. The total lead time remains at 211.5 hours.

### **Size of the update package**

In the following, the size of the update package is now increased from 10 MB over 100, 250, 500, 1000 to 2500 MB.

As expected, the delta increases linearly with the size of the update packet to the initial case of 10 MB. This means that while the total duration increases by 2.75 hours at 250 MB, it increases by 25.65 hours at 2500 MB. This effect can also be seen in relation to the number of server slots / number of vehicles. Here the absolute delta varies with respect to the performance of the selected ratio.

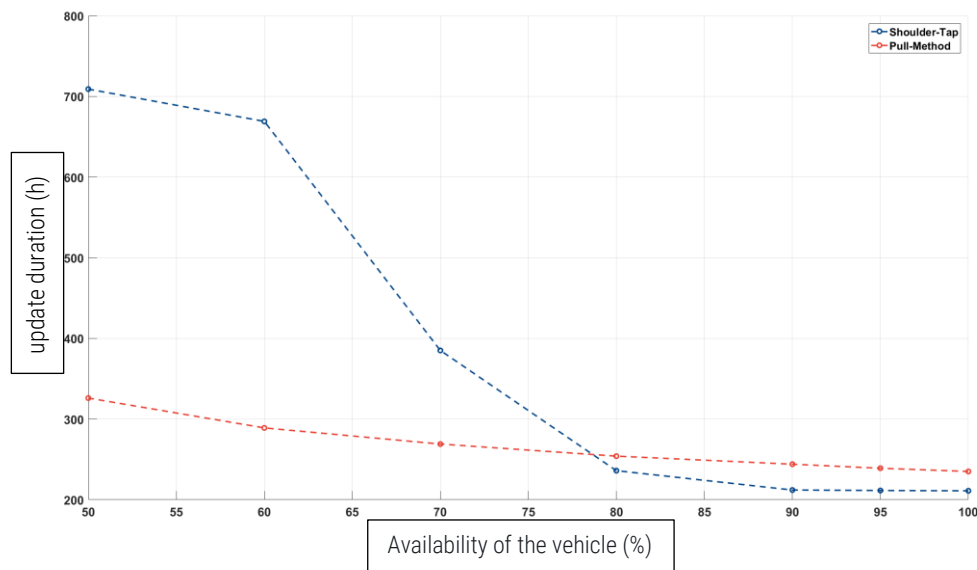
It should be noted that the installation time of the individual user can vary greatly depending on the size of the package, if major updates leave the vehicle out of service for a longer period to perform the installation. For example, a larger update can have a negative effect on the probability that a customer will accept an update. If the time window for the installation becomes larger, it becomes more difficult for the customer to carry out an update spontaneously, more planning is invested so that the vehicle is not required during the update process.

### **Communication method (Shoulder-Tap or Pull-Method)**

The Shoulder-Tap method enables a real-time communication of the update to the vehicle fleet. Nevertheless, effects such as the lack of connection of vehicles can reduce this advantage. The pull method, on the other hand, has the advantage of better planning. This not only saves costs because less server capacity is needed over a period of time, but it can also be better countered if the update contains a systematic error.

Due to the delayed distribution, more reaction time is available to stop the update before too many vehicles are affected. In addition, with the pull method, the vehicle itself selects the time when it has a connection to the server. This can also be supported by learning algorithms to increase the likelihood of a successful download and installation.

The shoulder tap is recommended for time-critical updates, but otherwise the pull method is more cost-effective. Therefore, the Shoulder-Tap method is chosen here for the example update. For this purpose, it is assumed that 90% of the vehicles can be reached.



**Figure 21: Comparison of both communication methods**

The comparison of the two methods depending on the accessibility of the vehicles (Figure 21) shows that the advantage of the shoulder tap is equalized in the range between 75% and 80% accessibility and even develops into a disadvantage for the update duration. In the simulation, this is due to the fact that the request is postponed for several hours after the vehicle has not been reached several times. With the pull method, however, the vehicle "knows" when it can be reached and so the success of the individual requests increases.

### Check the correct configuration

The verification of the correct configuration takes place at two points in time in this scenario. First during the identification of the target vehicles and then as part of the installation routine. Ultimately, however, this is only a correction factor in the overall view. The throughput time for the vehicles does not change, as the update is aborted in case of unsuitable configurations. These failed vehicles, however, distort the 90% quantile, since due to the distribution of the failure mainly affects vehicles with shorter throughput times, the 90% quantile shifts to the rear. Individual runaways with very long running times are more important.

The probability stored for this test can be based on empirical values, for example. It increases with better integrated processes up to the workshop or the improved communication with the vehicles.

Even if this factor only has a corrective effect, it still serves to forecast the update duration, especially with regard to possible default risks. If more vehicles fail during the update process, these vehicles may have to be recalled to the workshop. The process costs increase and can thus be planned in.

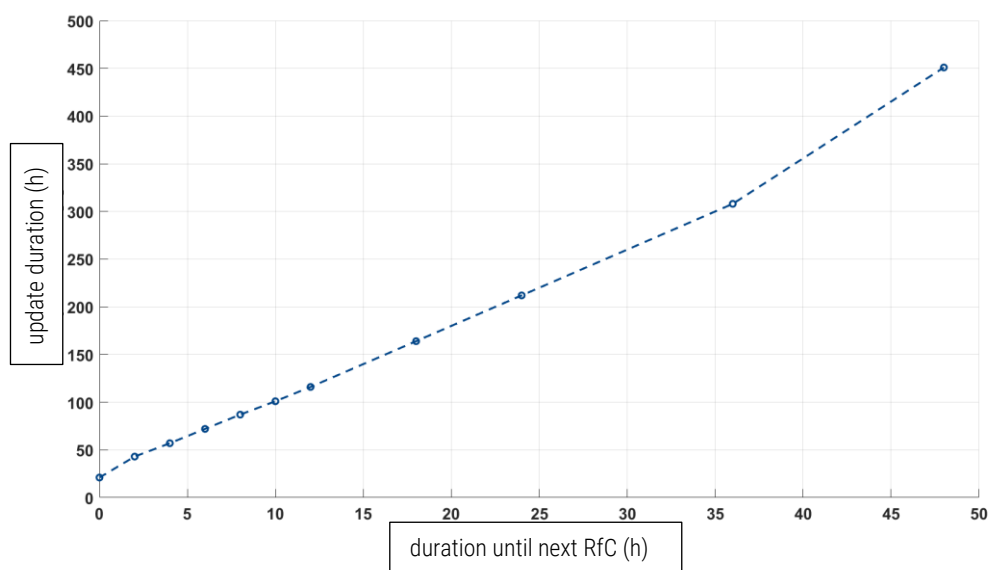
### Confirmation of the update

The activity of the owner approving the update and e.g. executing the update in the vehicle via the IVI has a significant influence on the process chain.

Two parameters in particular must be taken into account. The regularity of the request (e.g. in the vehicle by the IVI) after confirmation and the likelihood that the driver will accept the update.

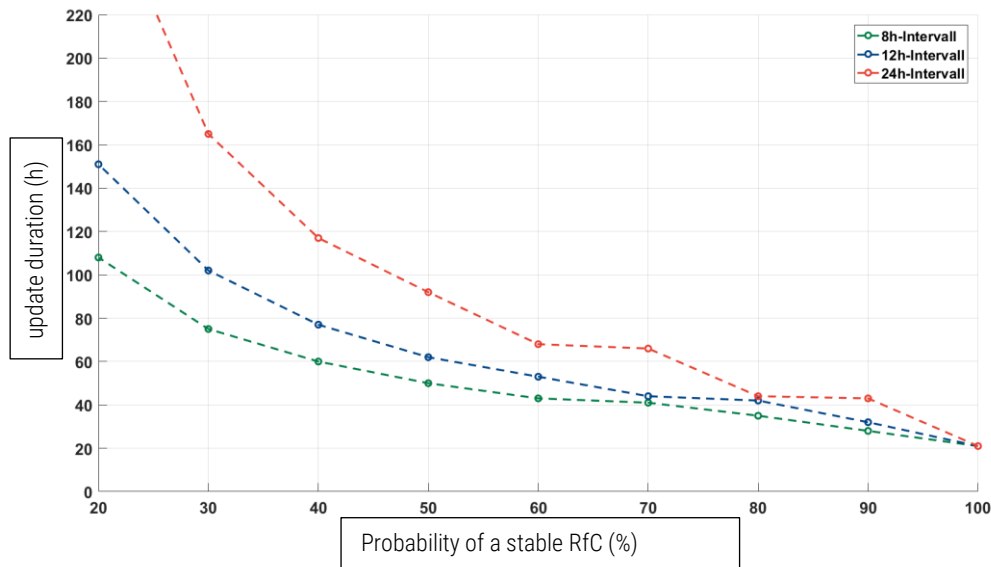
The influence of the request on the update duration is linear. From about 40 hours update time with one notification every 4 hours, the curve develops almost linearly to the last tested value and one notification every 48 hours with an update time of about 450 hours. If one assumes that this could be a mandatory update, the update duration would drop to around 21 hours in this scenario.

The influence of the request on the update duration is linear. From about 40 hours update time with one notification every 4 hours, the curve develops almost linearly to the last tested value and one notification every 48 hours with an update time of about 450 hours. If one assumes that this could be a mandatory update, the update duration would decrease to about 21 hours in this scenario (duration until the next request equals 0). The values were recorded with a 25% probability of accepting the update.



**Figure 22: Influence of the request after confirmation from the customer**

It must be examined to what extent the frequency of the notification may have an influence on the willingness of the customer to actually confirm the update. In the following it is assumed that frequent notification of the customer leads to the driver being confronted too often in inappropriate situations and therefore routinely ignoring the message. This should therefore reduce the probability that the customer will confirm the update with the reminder. This will be examined in the next step.



**Figure 23: Influence of the probability that an update will be accepted**

**Figure 23** shows the effect if the probability of the update being accepted is varied. This is almost linear in the range above 50 % and then progressive. In addition, the update duration scales up by a factor if the interval of demand for confirmation of the update is varied.

In addition to the already discussed negative influence of a too frequent demand on the probability of acceptance, this probability can also be positively influenced by the design of the message to the customer. The better customers are informed about the content, possible consequences, installation time and other factors, the better they will be able to assess the actual relevance of the update for their vehicle and prepare to update accordingly.

An update interval of 12 hours is planned for the further course and urgency of the update. Compared to the 8-hour interval, this assumes a higher probability of 60 % for the assumption. The update time drops to around 52.9 hours.

### Probability of a download success

The probability of a download success has no significant influence on the 90% quantile of the update duration. This remains constant as far as possible. Only if too many vehicles fail in this process step due to the frequent failure of the download attempt is the 90% quantile negatively affected. However, this would have to affect more than 10% of the vehicle fleet in order to have an impact. It is assumed that this value - for this process step alone - is unlikely. The update duration thus remains at around 52.9 hours.

### Number of installation attempts per day

After the successful download of the update package, the installation is triggered. A parameter has been set here to describe the number of times that the update will be attempted to install. This was initially set to 1, which means that every 24 hours an attempt is made to start the installation. However, due to the urgency of the update, this value is increased to 3. This would mean that the

driver would be suggested an installation every 8 hours. This can reduce the update time to 48.4 hours. It should be noted, however, that the more frequently the update is proposed, the less likely it is that the update will be accepted, as the user is "disturbed" too often and develops an automatic rejection process.

### 4.3.3 Conclusion

The analysis shows the different influence of the different parameters. When selecting the parameter for various parameters, it becomes clear that a compromise must be made between the speed of the update process and other important factors such as (economic) efficiency, but also customer acceptance etc. These were discussed as examples in the optimization process. Table 2 summarizes the optimizations made, logs the changed update duration and shows how the variation of individual parameters affects the update duration.<sup>12</sup>

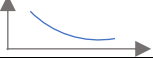
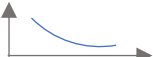


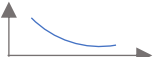
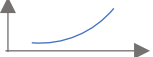
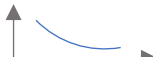
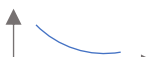
process parameters	optimisation	update duration	influence type
initial duration		215.9 hours	
Number of server slots / number of vehicles	Increase of the relative ratio from 1% to 2.5%.	211.5 hours	
Distribution of transmission technology	The following scenario follows: WiFi 20%, 3G 30%, 4G 50%.	211.5 hours	
Size of the update package	Follows from the scenario: 10 MB	211.5 hours	
Accessibility of the vehicle	Is set at 90% for Shoulder-Tap method.	212.2 hours	
Probability of correct configuration	Correction factor, set to 99.5%.	212.2 hours	
Period of repeated reminder of the update	The user is now reminded every 12 hours of the authorization update instead of every 24 hours.	86.8 hours	
Probability of accepting the update	Due to the security relevance of the update, the probability is increased from 25% to 60%.	52.9 hours	
Number of installation attempts per day	Due to urgency, the number is increased from 1 to 3.	48,4 hours	

Table 2: Summary of process optimization

<sup>12</sup> On the ordinate of the symbolic diagram in Table 2 is the update duration and on the abscissa the respective varied parameters are plotted.



## 5 PROJECT CLOSURE

In this project the results of the first project were further deepened and cross-checked. First, the proposed classifications for software updates were compared. It was found that combining and supplementing these combinations would only result in a new classification with too many classes due to the different viewing angles. The advantages of a classification would be leveled by a too cumbersome applicability. Rather, it is recommended to classify updates using attributes - see Section 2.5.

Chapter 3 then contains a significantly revised version of the idealized process designed in the first software update over-the-air project. This was revised in all activities in terms of content depth - e.g. by the consequences of the various and possible attributes - but also corrected in many places so that activities were omitted, and others were added. The requirements of the UN ECE 'Taskforce on Cyber Security and OTA Issues', which are now linked to the individual activities of the idealized process, were a particular aspect of the study.

A third work package of the project was the simulation of this process to compare different use cases and to investigate the influence of different process parameters. The simulation was programmed for this in the AQI in MATLAB. The process chain has already been validated through modeling and subsequent implementation alone, which has made a decisive contribution to detailing the process description.

The core of this project is this process explication, which describes a holistic, idealized OTA process from the generation of the update through distribution to installation. This can be used at association level as a basis for discussion, internally for comparison with UN ECE requirements or for structuring one's own OTA process.

## Appendix

### Discontinued activities in relation to project OTA-1

#### **OTA-G-010 [not applicable]**

<del>Input: Information about a malfunction in the vehicle. The source of the information can be different: it can be internal through own observations or external through communication with the regulatory authority, the customers, the press or similar.</del>
<del>The <b>update coordinator</b> analyses the information and identifies the need for a software update for an affected component / system.</del>
<del>Output: Update requirement message including error image and possible cause for further tracking</del>

Corresponding processes have already been established for the identification of update requirements. This activity is therefore omitted from the OTA process, since there are no OTA-specific requirements for it. The OTA process now starts directly with an already identified update requirement or the defined requirements / specifications of the update.

Nevertheless, it should be noted that the need for a software update can be fundamentally met through three different channels:

- (1) troubleshooting process
- (2) Strategic product development
- (3) Customer feedback or customer request, e.g. for activating a function in the vehicle<sup>13</sup>

#### **OTA-G-040 [not applicable]**

<del>The <b>update coordinator</b> communicates the update requirement to the <i>development manager</i>.</del>
<del>Output: Documented update requirement message</del>

<sup>13</sup> The networking of the vehicles enables direct communication with the customer. Among the users there are groups who like to give feedback back to manufacturers and actively participate in product developments.

This activity only fitted if the process only considers updates for fixes. However, since other use cases are now taken into account, this activity is not performed. It is merged into the new activity OTA-G-051, which now starts the process from the relevant department.

### **OTA-G-050 [not applicable]**

<del>The <b>update coordinator</b> defines the specifications for the software update.</del>
<del>Output: Specifications for software update</del>

Justification see OTA-G-040.

### **OTA-G-080 [not applicable]**

<del>Input: vehicle configuration / vehicle data</del>
<del>The <b>development manager</b> analyzes the transmitted vehicle configuration / the extracted vehicle data and supports the <b>developer</b> with the results of the analysis of the transmitted system / diagnostic data on the vehicle condition.</del>
<del>Output: diagnostic results<sup>14</sup></del>

Not necessary because internal processes are mapped for which no specifications are required. The contents of the activity are covered by the activity OTA-G-090.

### **OTA-G-100 [not applicable]**

<del>The <b>developer</b> transmits the software update to the update server of the OEM and informs the <b>development manager</b> about the new software status.</del>
<del>Output: Software update available to the OEM; notification of availability of the update.</del>

Not necessary because internal processes are mapped for which no specifications are required. The contents of the activity are covered by the communication flow between activity OTA-G-090 and OTA-G-110.

<sup>14</sup> The diagnostic results can be different. The basis are market data of affected vehicles, possible are also the vehicle configurations under which the errors occur, up to user activities where the errors occurred.

**OTA-G-140 [not applicable]**

The <del><i>update coordinator</i></del> releases the software update for the implementation of the recall on the customer vehicles.
--

<u>Output:</u> released update
--------------------------------

The activity is omitted and is represented by the activity OTA-G-130, which reflects the internal release process of the respective company. A differentiation into two separate activities does not achieve any added value, especially since the activity of releasing is only conditionally OTA-specific. Nevertheless, it can have a considerable influence on the performance of the process chain.

**OTA-V-070 [not applicable]**

The <del><i>digital service provider</i></del> updates the vehicle database according to the feedback of the vehicle.
---

<u>Output:</u> News about updated vehicle database
--

Note: If the vehicle configuration is not compatible, i.e. does not meet the requirements of the update, the vehicle is omitted from the further update process. This circumstance and its cause will be documented by the digital service provider and transmitted to the client in an update report.
--

These activities can be omitted as they can be automated by OTA-V-060 and do not require any additional administrative effort.

**OTA-V-010 [not applicable]**

<u>Input:</u> unlocked update package
---------------------------------------

The <del><i>update coordinator</i></del> forwards the approved update to the <del><i>digital service provider</i></del> .
---

<u>Output:</u> uninstalled update package
---

Not applicable because the activity is merged into OTA-G-141 (the update packet is sent to the *Digital Service Provider*) and OTA-V-020 (the *Digital Service Provider* receives the update packet).

**OTA-V-110 [not applicable]**

The <del><i>vehicle</i></del> manages the approval process and forwards the decision to install the update.
---

Output: Result of the approval process

**Conditions:** ~~(A) If approval is granted, the update coordinator will be notified accordingly. (B) If approval has not been received, the update can be displayed again. If the update is finally rejected, the update coordinator must also be informed.~~

This activity runs automatically and does not require any special explication, since no special requirements result from this. Further requirements are included in activity OTA-V-120.

### OTA-V-130 [not applicable]

~~The **digital service provider** initiates the start of the download (e.g. by sending a download reference).~~

Output: Download reference

~~Note: A) This activity can also take place directly after the update has been activated, i.e. before the update has actually been authorised by the vehicle owner. To increase customer satisfaction, the update can be loaded while the confirmation is still pending. An installation can therefore take place without delay directly after the confirmation. However, this is associated with higher costs, e.g. if the update is not then authorized. B) An update can also be distributed to the vehicles in several 'waves' to distribute access to the server structure over time.~~

The activity can be omitted as it is already implied in the vehicle notification (OTA-V-050).