SIBROS

WP.29 Requirements and Effects on OEMs and Third-Party Suppliers

www.sibros.com

WP.29 Requirements and Effects on OEMs and Third-Party Suppliers

First published November 2021

www.sibros.com

CONTENTS

Introduction

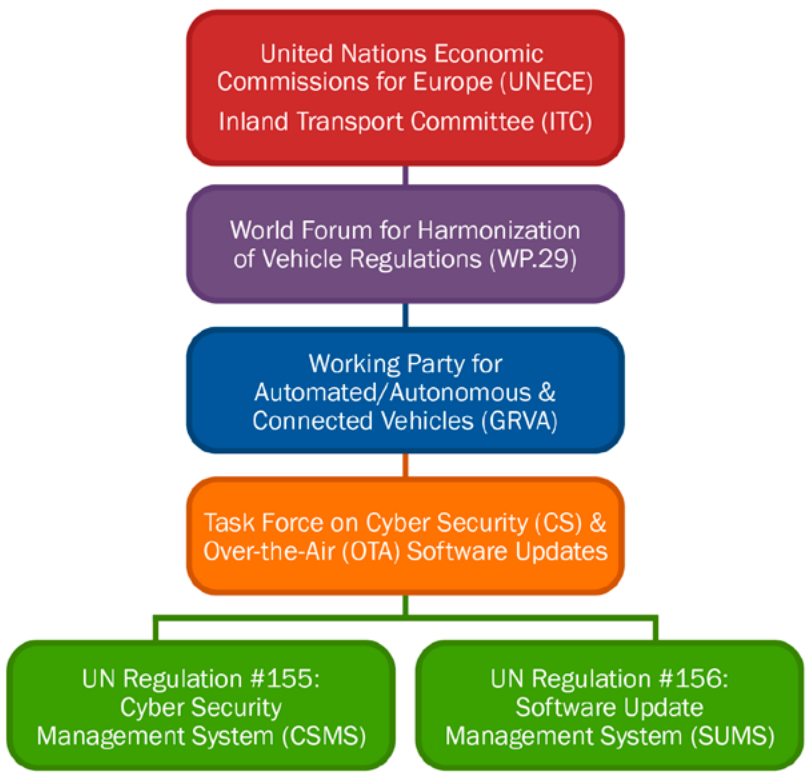
Security, convenience, and innovation are the three pillars upon which consumers base their opinions regarding a new product. When a manufacturer fails to keep these pillars in mind, they risk not only the success of their new product but also their entire company. The automotive and connected car industry is not exempt from these driving principles. In fact, it depends upon them.

A vehicle without effective security measures places the lives of the driver and everyone else on the road at stake. If a car fails to offer convenience, it is quickly overlooked in favor of those that do. Gone are the days when people bought a new car when, and only when, the old one failed. Today consumers eagerly await advances in innovation, and when a manufacturer fails to provide, the motivation to purchase a new model ceases to exist.

These three pillars are inextricably intertwined. The desire for convenience drives the need for innovation, which creates a necessity for novel security

measures. If one pillar falls, the rest fall with it, often taking the manufacturer along for the ride. To prevent the collapse of an entire industry and ensure the continued safety of consumers, **Original Equipment Manufacturers** (OEMs) must adhere to specific standards and regulations pertaining to security, innovation, and convenience. But who is responsible for assessing the need for and mandating such standards?

Table 1
UNECE Structure



WP.29, otherwise known as, the Working Party for World Forum for Harmonization of Vehicle Regulations. WP.29 is a regulatory forum within the **United Nations Economic Commission for Europe** (UNECE), (Table 1).

It oversees regulations related to multiple realms in the automotive sector, including everything from vehicle safety to environmental concerns. Not only do WP.29 regulations affect OEMs, but also the third-party suppliers that work alongside them.

In June 2020, WP.29 adopted two new regulations. UN Regulation #155 addresses the increasing demand for **heightened cybersecurity** in the automotive sector, while UN Regulation #156 outlines safety requirements for **software management update systems in connected vehicles**.

As we delve into the inner workings of these new WP.29 regulations, we will explain their importance, assess their impact on the automotive sector, and outline how OEMs can achieve compliance.

01

The Importance of R155 and R156

The Importance of R155 and R156

Software updates for connected vehicles are becoming increasingly advanced and successful execution of these is pivotal to an OEM's success in the competitive and ever-changing landscape of technological innovations. Software updates serve as a vital way to continuously deliver not only new features that customers expect—as they would from a smartphone—but also to deploy critical updates that ensure the safety of drivers, passengers, and even pedestrians.

The ability to successfully manage and send updates is only half the battle. As software advances, the threat of cyberattacks on vehicles continues to grow. This includes both passive attacks, such as eavesdropping via telecom systems, and active attacks, such as uploading malware to prevent brake functionality.

ON THE IMPORTANCE OF AUTOMOTIVE CYBERSECURITY

> *"We've seen countries go after each other and cause all sorts of damage, and when you have something like automotive hacking, it's such a huge risk in this area.*

JUSTIN CAPPOS, FOUNDER OF UPTANE

"We've seen countries go after each other and cause all sorts of damage, and when you have something like automotive hacking, it's such a huge risk in this area," says Justin Cappos, NYU Professor, and founder of Uptane. Vulnerabilities to these attacks place drivers' lives in jeopardy and are costly in terms of the OEM's reputation, recall expenditures, warranty claims, and the level of trust they instill in their customers.

In response to the rapid evolution of the connected vehicle industry, WP.29 adopted R155 and R156. These regulations establish clear standards for software cybersecurity and update systems as a means of ensuring the continued safety of drivers, passengers, and pedestrians alike.

The first regulation, R155, requires the implementation of processes that assess and address the risks associated with cyber threats, as well as mechanisms to protect vehicles against cyberattacks. This systematic risk-based approach, or **Cyber Security Management System** (CSMS), is required in the following vehicles:

- Categories M and N
- Passenger cars, vans, trucks, and buses
- Vehicles with level 3 and above automated driving functions
  - Automated pods
  - Shuttles
  - Trailers with one or more **Electronic Control Unit** (ECU)

The second regulation, R156, dictates standards and requirements for the implementation of a **Software Update Management System** (SUMS). This is defined as a systematic approach that includes organizational processes and procedures to ensure the safe and successful delivery of software updates.

Compliance Deadlines

These regulations impact 54 countries around the globe, including both EU and Japanese markets. In addition, they are expected to have lasting indirect effects on more than 20 million vehicles located in over 60 countries, not including commercial vehicles.

WP.29 has mandated two deadlines by which OEMs must establish both a CSMS and a SUMS that comply with the requirements outlined in R155 and R156.

Deadlines for R155 and R156 Compliance

JULY 2022	Compliance deadline for all **new** vehicle models
JULY 2024	Compliance deadline for **existing models** undergoing first time registration

Failure to adhere to these requirements will severely impact the OEM's future success. Although manufacturers may continue to sell in markets not under the jurisdiction of WP.29, they will be prohibited from the production and distribution of vehicles in regions where the regulations apply.

The first step to achieving compliance is understanding the requirements laid out by UN Regulation #155 and #156.

02

Cybersecurity Management System Requirements

Cybersecurity Management System Requirements

In terms of R155, manufacturers must implement and maintain a system capable of managing cybersecurity risks to both the software update system and the vehicle design. The two possible avenues for compliance are the research and development of in-house software or the purchase of an existing solution.

Regardless of which route the manufacturer takes, the OEM's software solution must include control processes across two specific areas: cyber risk management protocols and vehicle type.

Management Protocols

The OEM must have a cybersecurity management system that will last for the lifetime of the vehicle, including all steps of the manufacturing process: development, production, and post-production. The CSMS must address the 7 key vulnerabilities and 69 attack methods outlined in WP.29 R155, as well as those listed in Annex 5 (Table 2).

Table 2
Cyber Threat Vulnerabilities and Attack Methods

High-level and sub-level descriptions of vulnerability/threat		Example of vulnerability or attack method
Threats regarding back-end servers related to vehicles in the field	1. Back-end servers used as a means to attack vehicle 2. Services form back-end server being distrupted 3. Vehicle related data on back-end being lost or compromised	• Insider attack (abuse of privileges by staff) • Unauthorized internet access to server • Unauthorized physical access • Attack on back-end servers that stops functioning • Loss of information from the cloud • Information breach by unintended sharing of data
Threats to vehicles regarding communication channels	4. Spoofing of messages or data received by vehicle 5. Communication channels used to conduct unauthorized manipulation, deletion or other amendments to vehicle held code/data 6. Communication channels permit untrusted/unreliable messages to be accepted or are vulnerable to session hijacking/replay attacks 7. Information can be readily disclosed (eg. through eavesdropping) 8. Denial of service attacks to disrupt vehicle function 9. Unprivileged user able to gain access to vehicle systems 10. Viruses embedded in communication media are able to infect vehicle systems 11. Messages received by vehicle (eg. through V2X or diagnostic messages) contain malicious content	• Spoofing of messages • Sybil attack • Communication channels permit code injection (eg. tampered software binary) • Communication channels permit manipulate, overwrite, and/or erasure of vehicle held data/code • Communication channels permit introduction, manipulation, erasure, or overwriting of data/code to the vehicle • Accepting information from an unreliable or untrusted source • Man in the middle attack • Replay attack • Interception of information • Sending a large amount of data to the vehicle so it is unable to provide normal services • Black hole attack • Unprivileged user gaining privileged access • Virus embedded in communication infecting vehicle systems • Malicious internal (eg. CAN), V2X, diagnostics, or proprietary messages

High-level and sub-level descriptions of vulnerability/threat		Example of vulnerability or attack method
Threats to vehicles regarding their update procedures	12. Misuse or compromise of update procedures 13. Denying legitimate updates	• Compromise of over-the-air software update procedures • Compromise of local/physical software update procedures • Software manipulation before the update process • Compromise of cryptographic keys allowing invalid updates • Preventing the rollout of critical software updates
Threats to vehicles regarding unintended human actions facilitating a cyber attack	14. Legitimate actors taking actions that unwittingly facilitate a cyberattack	• Spoofing of messages • Innocent victim tricked into taking action to unintentionally load malware or enable an attack • Defined security procedures are not followed
Threats to vehicles regarding their external connectivity and connections	15. Manipulation of the connectivity of vehicle functions enables a cyberattack (including telematics, systems permitting remote operations and systems using short range wireless communications) 16. Hosted 3rd party software (eg. entertainment applications) used as a means to attack vehicle systems 17. Devices connected to external interfaces used to attack vehicle systems (eg. USB ports, OBD port)	• Manipulation of functions designed to remotely operate systems (eg. remote key, immobilizer, and charging pile) • Manipulation of vehicle telematics • Interference with shortrange wireless systems or sensors • Corrupted applications or those with poor software security used to attack vehicle systems • External interfaces such as USB or other ports used as a point of attack through code injection • Media infected with a virus connected to a vehicle system • Diagnostic access (eg. dongles in OBD port) used to facilitate an attack (eg. manipulate vehicle parameters)

High-level and sub-level descriptions of vulnerability/threat		Example of vulnerability or attack method
Threats to vehicle data/code	18. Extraction of vehicle data/ code 19. Manipulation of vehicle data/ code 20. Erasure of data/code 21. Introduction of malware 22. Introduction of new software or overwrite existing software 23. Disruption of systems or operations 24. Manipulation of vehicle parameters	• Product piracy • Unauthorized access to owner's privacy information • Extraction of cryptographic keys • Illegal/unauthorized changes to the vehicle's electronic ID • Identity fraud • Action to circumvent monitoring system • Data manipulation to falsify vehicle's driving data • Unauthorized changes to diagnostic data • Unauthorized deletion/manipulation of system event logs • Introduction of malicious software or malicious software activity • Fabrication of software of the vehicle control system or information system • Denial of Service • Unauthorized access to falsify the configuration parameters of key functions (eg. brake data, airbag deployed threshold, etc.) • Unauthorized access to falsify the charging parameters (eg. charging voltage, charging power, battery temperature)

High-level and sub-level descriptions of vulnerability/threat		Example of vulnerability or attack method
Potential vulnerabilities that could be exploited if not sufficiently protected or hardened	25. Cryptographic technologies can be compromised or are insufficiently applied 26. Parts or supplies could be compromised to permit vehicles to be attacked 27. Software or hardware development permits vulnerabilities 28. Network design introduces vulnerabilities 29. Unintended transfer of data can occur 30. Physical manipulation of systems can enable an attack	• Combination of short encryption keys and long period of validity enables attacker to break encryption • Insufficient use of cryptographic algorithms to protect sensitive systems • Using already or soon to be deprecated cryptographic algorithms • Hardware or software engineered to enable an attack or fails to meet design criteria to stop an attack • Software bugs • Using remainders from development can permit access to ECUs or permit attackers to gain higher privileges • Superfluous internet ports left open, providing access to network systems • Circumvent network separation to gain control • Information breach (eg. personal data may be leaked when the car changes users/is sold) • Manipulation of electronic hardware (eg. man in the middle attack) • Replacement of authorized electronic hardware with unauthorized electronic hardware • Manipulation of sensor data

The OEM must have protocols and security processes in place to identify, categorize, and effectively address these vulnerabilities. In addition to performing risk assessment and management, the CSMS must be able to respond to and mitigate novel threats and adapt to new risks by continually testing itself for areas of weakness. The OEM must also ensure that any outsourced materials or supply chain partners are compliant with WP.29 regulations.

To achieve compliance for these management protocols the OEM needs to demonstrate to an approval authority, such as a technical service auditor, that their CSMS includes all the processes listed above for the entire vehicle lifecycle.

Vehicle Type Approval

Before vehicles are placed on the market, they must undergo vehicle type approval. Manufacturers must perform an exhaustive risk assessment of all critical elements of in-vehicle software and hardware and have implemented security measures to successfully protect these components against potential cyberattacks. Any vulnerable elements must have appropriate and proportionate backup measures to deal with both attempted and successful cyberattacks.

In other words, the vehicle type must have monitoring capabilities that can detect and prevent cyberattacks, and relay relevant data to the manufacturer. The CSMS should then utilize this data to perform an in-depth **Threat Analysis and Risk Assessment** (TARA) and implement additional fleetwide security measures to mitigate future risks. Finally, OEMs must demonstrate that these systems are in place and report any relevant attacks at least once a year to their technical service auditor or appropriate regulatory body.

Achieving CSMS Compliance

As previously mentioned, OEMs must manufacture or outsource the cyber-security framework, or CSMS, for their OTA update system. Creating this type of software in-house requires a massive allotment of resources as well as personnel with relevant technological expertise. On paper this route may appear more cost-effective, however in the long run it often leads to expensive delays, dead ends, and potential recalls. The other route is to purchase a ready to implement OTA solution that utilizes a time-tested CSMS, such as the Deep Connected Platform (DCP) offered by Sibros.

DCP utilizes an open-source CSMS framework called Uptane. This novel software update system is the first of its kind and was developed specifically for the automotive industry. It utilizes a multiple access level structure, that works to safeguard OTA updates from manipulation and hacking that leads to system-wide failures.

“The image repository inside of Uptane does not contain a [single] key that is trusted to sign updates,” says Uptane’s

founder Cappos. Instead, it is designed with a ‘separation of trust’ organizational structure, which means full access to stored updates can only be achieved through a series of designated approval signatures at multiple sign-in and entry points, both on and off-line.

Once an update is accessed it requires additional signatures before it can be delivered. This mitigates the risk of sending unauthorized or malicious updates, as a hacker would need to simultaneously break into multiple access points, each with a different security key, to prove successful. If a security key ever does become compromised, Uptane’s implicit and explicit mechanisms allow for smooth and easy invalidation and replacement of the specific key.

What’s more, the multi-layered cryptographic algorithms utilized by Uptane’s CSMS framework maintain the security of update authenticity and integrity, by guarding against unauthorized changes to software version numbers. Once again, any version number modification requires authorized signatures from multiple parties and can only be performed in the event of a relevant software change. To ensure compliance and consistency, additional mechanisms allow OEMs and third-party suppliers to retain records of all original signatures throughout the update modification and delivery process.

Adopting an existing OTA update solution like Sibros’ DCP easily satisfies R155 requirements. By establishing a compromise-resistant system to block malicious attempts to manipulate or deliver unapproved or damaged updates, DCP’s cybersecurity management system identifies and mitigates cyberattacks and protects the authenticity and integrity of updates before, during, and after rollout.

03

Software Update Management System Requirements

Software Update Management System Requirements

The second piece of the WP.29 compliance puzzle is R156. This pertains to Software Update Management Systems (SUMS) and includes four main requirements:

- Information security best practices
- Version checks
- Target identification processes
- Safety assurance mechanisms

This regulation mandates verification and documentation of every step of the SUMS processes to achieve compliance. In addition, OEMs must assure and prove that any outsourced software—such as that provided by an OTA software firm—adheres to R156 requirements as well as any regional regulations for data storage. R156 is not the first standard of its time. Procedures like ISO 26262 and SOC Type 2 outline similar requirements.

Information Security Best Practices

Any information related to R156 must be documented and safely stored. OEMs must use best practices for information security systems, this includes providing evidence that all relevant information is stored with the appropriate security controls in place, and that employees are trained on documentation practices, security procedures, and quality assurance.

All servers must be secure and remain secure, regulated, and monitored for data protection. Documented information must include processes:

- For OTA updates
- To safeguard against cyberattacks
- To prevent unauthorized access
- To verify and validate software functionality

In addition, manufacturers are expected to keep a history of software version numbers for each vehicle type, including a detailed description of update purpose, potential affected systems, and conditions under which an update can be safely performed.

Servers that house all relevant documentation must be available for review by approval authorities upon request. If a manufacturer fails to comply with these best practices or to show sufficient evidence of compliance, they will be subject to delays in production and/or distribution.

Version Checks

Version checks ensure that updates are protected against manipulation and mitigate cybersecurity risks throughout the software development lifecycle. They also allow OEMs to track software changes and verify the software version in each vehicle component matches their records.

To accomplish this, all software and relevant components must have **RX Software Identification Numbers** (RXSWIN). These numbers must be easily readable via either electronic communication or a standard diagnostics interface. RXSWIN modifications must only be executed by authorized personnel and only when a change is made to the software. To protect against unauthorized alterations, they must be safeguarded by enhanced security measures. Finally, before a software update rollout, the SUMS must verify that the RXSWIN in the component matches the information securely housed in the OEMs database.

Here is an example of an effective version check process: A **Body Control Module** (BCM) has multiple firmware images, each with a unique RXSWIN making them readily identifiable. Before anything in the BCM is updated or changed, the software version undergoes a validation cycle that checks the requirements at the system, software, and hardware levels. All information regarding version checks is then documented, securely housed, and made available to approval authorities.

Target Identification Processes

R156 specifies the need for target identification processes. This means OEMs must have a process for identifying target vehicles for software updates, as fleetwide updates are rare. Targeted vehicles must be identifiable and groupable via their **Vehicle Identification Number** (VIN) or any other number of characteristics.

Mayank Sikaria, CTO and Co-Founder of Sibros, emphasizes the importance of the ability to create vehicle groups, “This feature is something that is required by WP.29 but is also in my opinion, one of the most important features for an OEM.”

While many existing OTA solutions can group based on static features, few have the capacity to target vehicles using dynamic features. This capability differentiates Sibros’ from its competitors. “Sibros’ OTA solution can dynamically look at the information that is coming from the car to our Deep Logger product, and it can identify faults or different sort patterns based on that log data and can request, trigger, or alert an OEM that these vehicles need a software update,” says Sikaria.

ON CREATING VEHICLE GROUPS

> *This feature is something that is required by WP.29 but is also in my opinion, one of the most important features for an OEM.*
>
> MAYANK SIKARIA, CTO OF SIBROS

As previously discussed, once a target group is created, the SUMS goes through each vehicle and its components to verify all software RXSWIN before it proceeds with the update rollout. In addition to keeping records accurate and accessible for review, this provides OEMs with the data necessary to identify potential incompatibilities between new software and the current configuration in each vehicle.

Safety Assurance Mechanisms

SUMS must have safety assurance mechanisms in place to prevent updates from being installed when they compromise the integrity of related components or impact driver safety. These updates shall only occur when the vehicle is in a safe state. As part of ISO 26262 functional safety effort, the OEMs are required to perform **Hazard and Risk Assessment** (HARA) to derive the safety goals. In addition, OEMs should also conduct a **Failure Mode and Effect Analysis** (FMEA). For each potential failure that can occur in the vehicle, the OEM should have a recommended action or functional requirement in case an update fails.

Once the OTA system establishes that the vehicle is in a safe state and has sufficient power to complete the update, it should initiate an update. The software update will occur, if and only if, all safety preconditions are met. Finally, there must be a mechanism that notifies the owner before an update is executed, as well as whether the update was successful or unsuccessful. OEMs must document these processes and make them available to approval authorities upon request. When an update is in progress, the vehicle should ensure that it continues to be in the safe state (ie. vehicle in park).

Achieving SUMS Compliance for OTA Processes

Although it is the combined responsibility of the OEM, other contributors,

and third-party suppliers to adhere to R156, the OEM serves as the sole point of contact for approval authorities. The manufacturer must therefore ensure that all parties involved are complying with R156, including the maintenance of records and related documentation.

To better understand what is expected of OEMs, chapter 7.1.1 of R156 outlines the processes verified in compliance assessments and breaks them into twelve subsections:

1. Process at the Manufacturer	• The manufacturer has a process in place to undergo compliance certification
2. Unique Identifiers	• All software versions and updates are differentiable by unique identifiers to ensure validity • Includes failsafe to prevent unapproved updates from being sent or received
3. Vehcile Software Database	• Thorough records of each vehicle's software history, including failed and pending updates
4. Vehicle Manifest	• Full vehicle manifest allows traceability of all software and hardware changes for every vehicle in the fleet
5. Vehicle Integration	• Software update processes include integration verification mechanisms to identify interdependencies between systems in the vehicle
6. Vehicle Grouping	• Software allows the grouping of vehicles based on both static and dynamic features
7. Compatibility Validation	• Process to validate specific hardware/software combinations in a vehicle grouping for update rollouts
8. Change Impact Analysis	• Analyses to identify the potential impact of any update on the entire system
9. Feature Update Analysis	• Analyses that identify whether an update will push customer-facing features and vehicle performance outside of accepted parameters
10. Safety Impact Analysis	• Measures to evaluate whether an update will integrate safely or result in novel hazards
11. User Information	• Process to notify the owner of new software installations and changes • Must include the option to accept or postpone the change • Must provide notification of failed updates and installation issues
12. Update documentation	• Every aspect listed above is thoroughly documented and available upon request

All twelve of these requirements must be satisfied to demonstrate R156 compliance. According to Florian Rohde, an original founding member of vehicle software validation at Tesla, former director of integration validation at Nio, and cofounder of iProcess, “You have a creation process, you have a transfer process, and you have a receiving process.”

These three components build upon each other to create an efficient, effective, and WP.29 regulation compliant OTA process.

ON EFFECTIVE WP.29 COMPLIANCE PROCESSES

> *You have a creation process, you have a transfer process, and you have a receiving process.*

FLORIAN ROHDE, CO-FOUNDER OF IPROCESS

Creation Process

The creation process begins with a change request for new features, updates, or fixes. Changes first go to a control board to determine the effort required to deploy the change and the possible effects on other modules. Next, the update’s safety impact on the existing system is reviewed and assessed, after which the update is either approved or denied.

If denied, it is sent back to the drawing board. If approved, it goes to the development department where the update package is created. Next, the

integration department determines the best way to integrate the update and sends it off for validation. During this step, a team runs the updates through the entire system to look for any issues. If nothing arises, they validate the update and upload it to the cloud.

Transfer Process

The transfer process is when the update package is sent from cloud storage to the specified vehicles. It begins when the cloud initializes a health check on the vehicles slotted to receive the update. The cloud starts by confirming the RXSWIN in the vehicle's manifest match those in the database. If the check reveals anything unexpected, the system must determine whether the update is still possible. Once everything meets the specified requirements, the cloud waits to establish a strong and stable connection with the receiver (vehicle) and transmits the update package either in its entirety or in subsets, depending on package size.

The cloud waits for the receiver to send installation feedback. A 'failure' response causes the entire transfer process to reinitialize from the beginning. A 'success' response triggers the documentation of all changes and ends the transfer process.

Receiving Process

Receiving happens in conjunction with the transfer process and occurs within the vehicle itself. For the receiving process to begin the vehicle must have a stable connection with the cloud via a 4G, 5G, or Wi-Fi. The vehicle then receives a request from the cloud for information regarding its health, location, and existing software components. In response, the vehicle creates and uploads its full manifest and waits for the transfer process to confirm that the data matches the history stored in the cloud.

Upon receiving positive verification, the vehicle downloads the specified package from the transfer server. The package undergoes a validation check to ensure both the update and its source are correct. Once all security checks are passed, the download is stored, and the updates are unpacked and assigned to their designated components.

This is where owner communication comes in. The vehicle's user interface allows the driver to communicate with the system. Via a secure vehicle gateway, or hub, the owner can acquire information regarding update status and vehicle health.

Prior to installation, the system notifies the user of an impending update along with the option to commence the installation process or postpone. This notification includes a reminder that the vehicle will be inaccessible during installation and provides a time estimate, if available. Once customer approval is granted, update installation is initiated along with a full system check.

If installation in every component is successful, the vehicle undergoes a full system reboot, and the information is relayed to the cloud for documentation. In the event of a failure, the entire process must be reinitialized either via the receiver or via a request from the transfer server. If multiple failures occur, the system will notify the owner of the need to take the vehicle to a dealership to remedy the issue.

When appropriately implemented, the creation, transfer, and receiving processes satisfy all twelve requirements outlined in R156 Chapter 7.1.1 (Table 3).

Table 3
Satisfying R156 Requirements with Processes

CREATION PROCESS

7.1.1.1	**Processes at the Manufacturer**
7.1.1.5	**Vehicle Integration**
7.1.1.7	**Compatibility Validation**
7.1.1.8	**Change Impact Analysis**
7.1.1.9	**Feature Update Analysis**
7.1.1.10	**Safety Impact Analysis**
7.1.1.12	**Update Documentation**

TRANSFER PROCESS

7.1.1.2	**Unique Identifiers**
7.1.1.3	**Vehicle Software Database**
7.1.1.4	**Vehicle Manifest**
7.1.1.6	**Vehicle Grouping**
7.1.1.1	Processes at the Manufacturer
7.1.1.12	Update Documentation

RECEIVING PROCESS

7.1.1.11	**User Information**
7.1.1.1	Processes at the Manufacturer
7.1.1.4	Vehicle Manifest
7.1.1.5	Vehicle Compatibility
7.1.1.7	Compatibility Validation
7.1.1.12	Update Documentation

The bolded items represent the twelve requirements outlined in R156 Chapter 7.1.1 satisfied through the combination of the creation, transfer, and receiving processes.

Conclusion

WP.29 regulations #155 and #156 are designed to ensure driver safety and data protection are maintained as new conveniences and innovations continue to revolutionize the connected car industry. They establish clear standards for cybersecurity and software update management systems while providing guidelines on how to achieve compliance.

Even with a clear understanding of the mandated methods and procedures, the task of developing CSMS and SUMS that adhere to WP.29 regulations is a daunting one. While some OEMs prefer to carve their own way, this path is often fraught with setbacks, software bugs, and additional expenditures.

Sibros offers OEMs a clear, effective, and simple solution. The Deep Connected Platform is specifically designed for rapid integration; fleetwide data logging, management, commands, and diagnostics; comprehensive vehicle-wide updates for the vehicle's lifecycle; and user ease. Not only is it hardware agnostic and cloud-ready, but it is fully compliant with WP.29 standards and other regulations of its kind.

Prepare for the R155 and R156 deadline. Remain at the leading edge of security, convenience, and innovation. Connect with Sibros today.

About The Contributors

Mayank Sikaria, CTO & Co-Founder of Sibros

Mayank Sikaria is CTO at Sibros where he oversees firmware, engineering and technology development for the company's Deep Connected Vehicle platform used by leading OEMs and mobility brands. Prior to co-founding Sibros, Mayank worked at Faraday Future where he incorporated state-of-the-art practices in the vehicle development and validation phases. Mayank holds three patents, as well as Functional Safety (ISO 26262) certification and a bachelor's degree in Computer and Electrical Engineering from the University of California, Davis.

Justin Cappos, Founder of Uptane

Justin Cappos is a professor in the Computer Science and Engineering department at New York University and founder of Uptane. Justin's research philosophy focuses on improving real world systems, often by addressing issues that arise in practical deployments. His dissertation work was on Stork, the first package manager designed for environments that use operating system virtualization, such as cloud computing. Improvements in Stork, particularly relating to security, have been widely adopted and are used on the majority of Linux systems via integrations into Apt, YUM, YaST, and Pacman. His later research advances have been adopted into production use including by Microsoft, IBM, VMware, Cloudflare, Docker, RedHat, ControlPlane, Datadog, and git, as well as a substantial percentage of automobiles.

Florian Rohde, Co-Founder of iProcess

Florian Rohde worked for several years in the "classic" automotive world at Siemens and Continental managing the validation of the first generation electric power steering systems. During this time he was responsible for the system validation of the first generation electric power steering systems, some of the first ASIL-D projects worldwide. From 2012 to 2018 he served as Senior Manager of Vehicle Firmware Validation where he implemented the continuous validation concept, enabling the launch of vehicle software packages within 24 hours from code change to OTA deployment. After Tesla, Florian served as Director of System Integration and Validation at NIO where he collaborated closely with teams in Shanghai to develop technology providing continuous and fully automated integration and validation for new products and features. Florian joined iProcess in 2019 as a consultant where he helps leading OEMs integrate complex and safety-relevant mechatronic systems, and evolve software over time into cutting-edge products.

SIBROS

Sibros powers the connected vehicle ecosystem with its Deep Connected Platform (DCP) for safe and secure deep software updates, data collection, and diagnostics in one integrated system. DCP supports any vehicle architecture - from ICE, Hybrid, EV to Fuel Cell - while also supporting rigorous safety, security, and compliance standards such as ISO 26262, GDPR, and WP.29 (among others) required to operate in most countries.

By combining powerful automotive software and data management tools in one platform, Sibros empowers OEMs to reduce recalls and warranty claims and address hundreds of connected vehicle use cases spanning fleet management, predictive maintenance, data monetization, owner personalization, and beyond.

www.sibros.com