# Trends in Telematics

# 2022

**Knobloch & Gröhn GbR**

**Burgwall 15**

**D-443135 Dortmund**

**Germany**

**Date: 01 August 2022**

## Contents

# 1 Introduction

The following study describes the status quo and the predominant trends in the field of Telematics. Its aim is to give a short, but precise overview for public and private decision makers what technical options and trends will either support or hinder intended use cases and business cases.

## 1.1 Update on former reports

This study is to a certain extend an update of a previous study, " OEM 3rd Party Telematics - General Analysis" conducted in 2018 by Knobloch & Gröhn GbR (KG) on behalf of FIGIEFA (Fédération Internationale des Grossistes, Importateurs & Exportateurs en Fournitures Automobiles).

## 1.2 Scope

The study aims to identify key characteristics, abilities and limitations of technical concepts for access to and integration in the Software defined and connected vehicle. It will summarize the status and trends for each solution type. As an example, the status and trends Neutral Servers, Marketplaces and ExVe Solutions will be shown in one chapter.

### 1.2.1 Estimated impact of lessons learned from "Digitalization" in other sectors

"Digitalization", roughly characterized as an ever increasing degree of digital information flow in the development, sales and logistics for either digital services (like apps) or "Digitalized services" (like coded hardware) has affected other sectors earlier than the automotive sector. Key characteristics of "Digitalization" within markets are et. al. the creation of few, but big digital ecosystems dominated by few creators (see. e.g. phone market with Apple and Google). Because the connected and software defined vehicle is a computer network on wheels with a high demand for safety & security it will be interesting to determine to what degree the "Basic laws of Digitalization" like a rather brutal economy of scale have already influenced and shaped the Automotive market.

## 2 Evolutions around the Software Defined Vehicle

Software in the car is far away from being a new trend. The first Engine control units that used embedded software and computers in cars were introduced at the end of the 1970s, start of the 1980s.[1]

### 2.1 Technical evolution - EE-Architecture

As a rather rough summary since then, more and more small computers, Electronic control units, were introduced in the car over time with some modern vehicles ending up with around 150 small computers installed[2]. This "computer network on wheels" is connected via different topologies and different Bus systems (CAN, LIN, MOST) to name just a few.

The typical Electric/Electronic (EE)-Architecture of a recent car was despite first standardized elements like the CAN Bus or first implementations of AUTOSAR a rather complex and highly OEM company-specific solution.

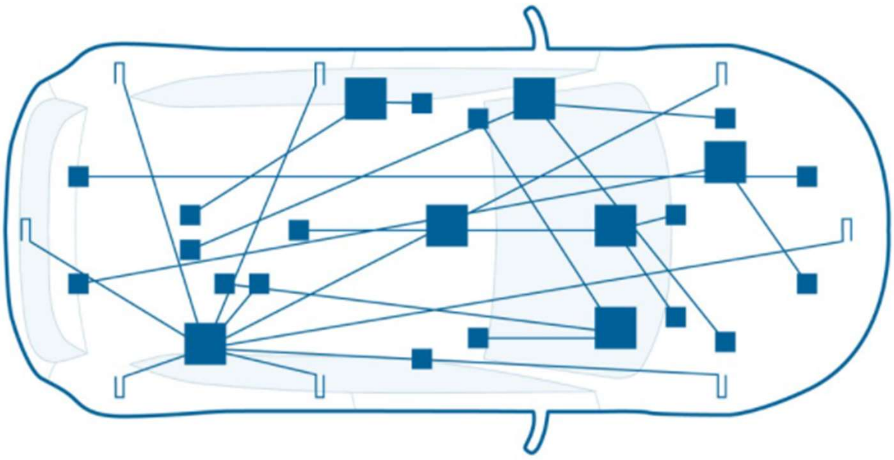


**Figure 1: Highly distributed EE-architecture.** [3]

In these vehicles, despite the introduction of some more powerful computers (Cross domain computers), there is still a great number of dedicated embedded Control units (ECUs) in the "normal" domain oriented architecture.

1 https://en.wikipedia.org/wiki/Engine_control_unit
2 https://en.wikipedia.org/wiki/Electronic_control_unit
3 https://www.bosch-mobility-solutions.com/en/mobility-topics/ee-architecture/

## 2.2 Organisational and Business Model impact

The usage and the evolution of technology has a strong relation with the way an enterprise that uses this technology is set up from an organization viewpoint (internal structure) and it impacts the depth of manufacturing and the resulting supplier network structure and typical deliverables (external structure).

Just to illustrate the trend it can be assumed that within the typical hierarchical set up of an OEM there was always a department assigned to develop the door/doors of the vehicle. After engineering the door with just mechanical elements in the early days of the automobile, the department has experienced the introduction of electrics and electronics and is now responsible for a door that feature electronically adjustable, retractable and heatable mirrors. A typical OEM is likely to order a door ECU from one or two of his Tier1 suppliers.

As a short summary of the situation the size of the "Digital ecosystem Door" was rather small. At the car, only the sensors and actors around the door where affected. Within the OEM a dedicated department was responsible for functionality of the door and the budget for all elements of the door, mechanical as well as electronic ones (ECU). In terms of know-how, the hardware and software of an ECU was built and developed by a Tier1 upon a "Black Box specification" of the OEM.

## 2.3 Trends towards stronger centralization

Bringing more and more dedicated ECUs in the car raised more and more issues over time. The more computers had to be bought, the higher the overall costs for the car, the higher the weight of the car. Furthermore, in terms of software on these computers each and every computer had to be equipped with the nearly same basic software components, e.g. for proper communication via the bus systems before the actual application software (to control e.g. the mirrors of the door) could be developed on top of these basic components. Last but not least the operating software (e.g. OSEK-OS / AUTOSAR) and the programming languages (e.g. C, C++) on these dedicated computers where rather automotive or IOT-device specific so the developers required could not be drawn from the large pool of software developers that were familiar with more popular operating systems like Linux, Android and programming languages like Java, C# or Python.

So in general OEMs have tried and are trying to reduce the number of ECUs inside their vehicles.

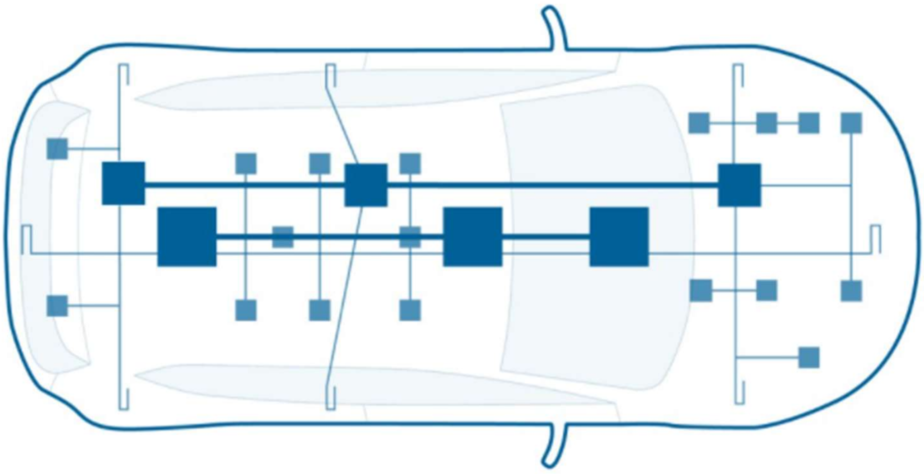


**Figure 2: Shift towards a zone oriented architecture with fewer, more powerful computers.**

In the architecture shown above from Bosch, fewer, but more powerful hardware nodes replace an increasingly larger number of dedicated ECUs.
In the fictive example, the actual application to control the door that resided on a dedicated ECU would be moved to a more powerful computer that hosts amongst the door software also software for climate control or infotainment.

This trend in digitalization, whereby more powerful hardware supports the operation of different apps providing different services on one hardware, can also be found in other sectors. The modern smartphone replaced not only the phone, but also the radio, the music player, the movie player as separate devices. The modern smart TV with a more powerful hardware inside rendered the former set top boxes obsolete. Subscribing to different channel sets or Movie platforms just requires the installation of a different app.

This technical trend towards a more centralized setup with fewer, albeit more powerful computers is also recognized by other consulting companies in the automotive sector, e.g. by McKinsey & Company.

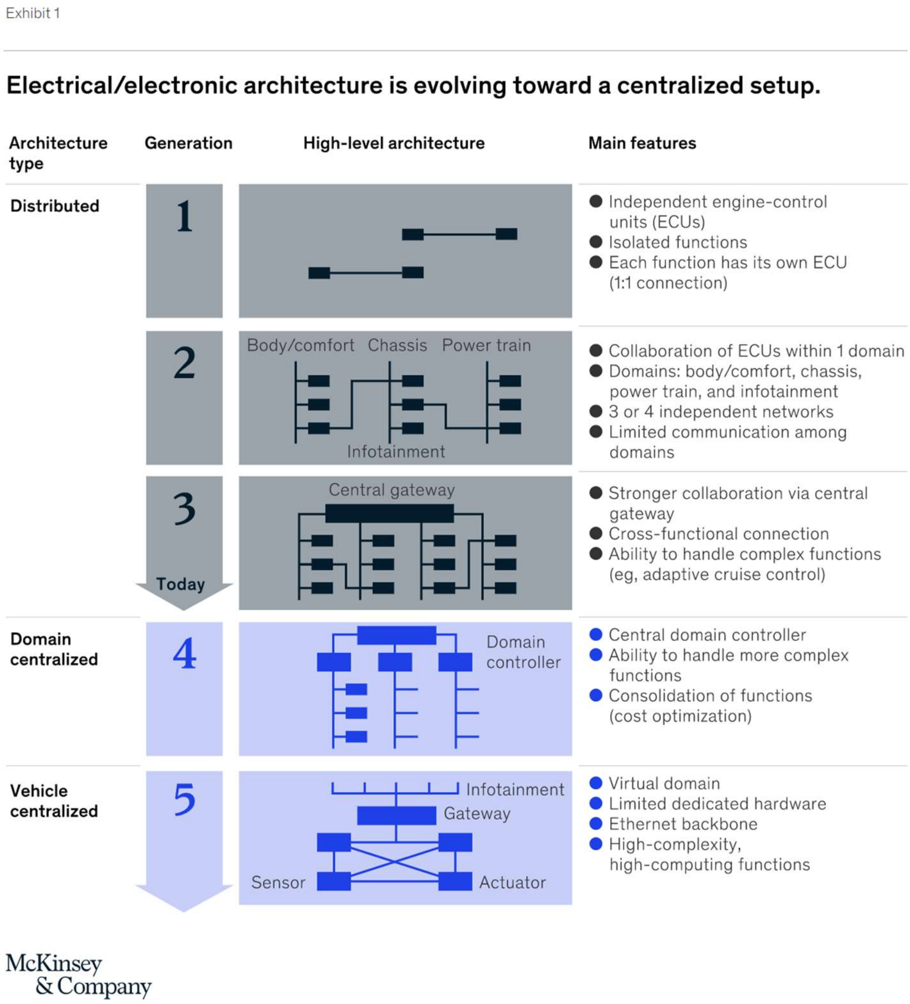



**Figure 3: Evolution towards a centralized setup**[4]

Focusing on the end-points of this evolution it becomes obvious why and where the influence of Silicon Valley companies with their excellent know how in hardware and software is increasing in the modern, software defined vehicle.

### 2.3.1 Generation 1/2 Architectures

In the Generation 1 architecture dedicated embedded ECUs are controlling dedicated functions as described.

---

[4] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/automotive-software-and-electrical-electronic-architecture-implications-for-oems

Very often, the functions controlled required a Real Time ability, so within the ECUs Real time operating systems prevailed. If ECUs communicated (Generation 2), they did it via signals to other ECUs.

The know-how needed to develop these systems was very "Automotive centric", hardware was expensive and software therefore and for security and safety reasons kept rather simple and static in its behavior.

Summary: OEMs (customers) and especially Tier1s (Providers) who not only developed these systems according to the specification of an OEM but very often designed them from the ground up were familiar with this "Hardware Centric"-Automotive IT.

### 2.3.2 Generation 5 Architecture

The more powerful computers that are replacing more and more smaller embedded ECUs offer new possibilities. However, to effectively exploit these possibilities, a different know-how set is required.

They can be programmed in state of the Art and widespread programming languages like Java, C# or Python, they communicate with each other via ethernet, secure communication via webservices and encryption is possible as is the hosting of different Virtual machines on one hardware node.

Summing up these powerful computers are more from the world of "normal computing" for laptops, smartphones and even servers or clouds when concepts like virtualization and hypervisors are introduced into the car.

Thus, it comes as no surprise that the companies that dominate the domains of "normal computing" with their technologies, applications and operating systems are having a distinctive advantage when these technologies find their way into the car.

## 2.4 "Digital" Effects on Organizational Setup and Business model (OEMs, Tiers)

With the increasing power of the digital device, more and more functions/applications can be operated using the same device with the same hardware limits and the same layers of basic software functionality.

While this fosters the achievement of lessened costs, complexity and weight, it has several other consequences on OEM as well as on supplier side.

### 2.4.1 OEM Impacts

On the OEM side (internal structure), more and more departments have now to agree on the common hardware, it's limits and costs. The Digital Ecosystem of the more powerful computer increased in size in comparison to the old Door-ECU computer and now encompasses e.g. the door department, the climate control department and the infotainment department.

Three OEM head of departments who formerly had their own budget to select the supplier and the performance of the dedicated ECU have now to agree on sizing, costs and financial split on their budgets to select the appropriate Super ECU. It is not unlikely that this will result ultimately in the creation of a

"Super Department" (e.g. "Comfort & Body") with a new Head that is responsible for all three functionalities and the selection of the supporting hardware.

This would be characteristic for ecosystems created by increased digitalization. These systems tend to increase in size, leading to a reduced number of different systems overall and a reduction in the number of system owners.

### 2.4.2 Tier 1 Impacts

For the Tier1 that formerly developed the software on the dedicated ECU and sold it together with the box, the effects can be also quite high: The new supercomputers are currently not manufactured by the typical Tier1 but by US companies like e.g. NVidia.[5]

So the Tier1 that formerly sold the door ECU loses the revenue in selling the hardware boxes (e.g. 20.000 per year for a popular model) together with the basic software components and the actual control application can in the future only charge the OEM for a license of the Tier1's door app that runs now on 20.000 NVidia computers in 20.000 cars.. To make things worse for the Tier1, customers are usually more willingly to pay for hardware in combination with software (because the hardware that can be touched symbolizes more worth to the human eye than the "invisible" software), and the replaceability of the Tier1 is raised significantly.

In the past the Tier 1 negotiated with the OEM once per year and got a contract to supply 20.000 boxes for an amount of X Euro for the next production year and volume of a given model. If the hardware had to be replaced over the lifetime of a vehicle, the original Tier1 was usually the only spare part supplier.

With the advent of Over The Air (OTA) Updates, this hardware business model can now be drastically evolved in a typical software business model. It would be a valid option to pay the Tier1 that developed the door application on a per vehicle and per day usage base until either the OEM and/or the customer decide that they want to download and use from now on the software of a competitor.

The typical bigger digital ecosystems has more competitors and the Tier1s find themselves in a far more volatile position than before.

### 2.4.3 Know how distribution

Many OEMs have in the past outsourced large portions of car know-how to their supplier network that Tesla's Elon Musk described their way of engineering as "Catalogue engineering" from the supplier base while Tesla prides itself of a much higher degree of vertical integration[6].

Tesla's strategy has focused on the in-house product development with the vertical integration and direct control of the full software and hardware development. This has helped them cope better than other

[5] https://nvidianews.nvidia.com/news/nvidia-introduces-drive-agx-orin-advanced-software-defined-platform-for-autonomous-machines
[6] https://cleantechnica.com/2020/10/23/elon-musk-explains-teslas-vertical-integration-vs-catalog-engineering/

OEMs with the chip supply crisis and has inspired others to try replicate the success by setting up their own software development companies (e.g. Cariad for Volkswagen)

Because this rather radical move can cause severe problems[7], it is not yet determined whether the future know how on software and hardware (albeit for larger systems) will be still with the Tier1s (old scenario) or if every OEM will try to follow Tesla's route, building up their own software and system's know how and rendering Tier1s down to pure suppliers of sensor and actor hardware with fairly limited IOT-intelligence.

### 2.4.4 The new Know-How-Frontline

The following Level-Model depicts an order of platform solutions inside (Levels 1 to n) and outside (Level 0) of the vehicle.

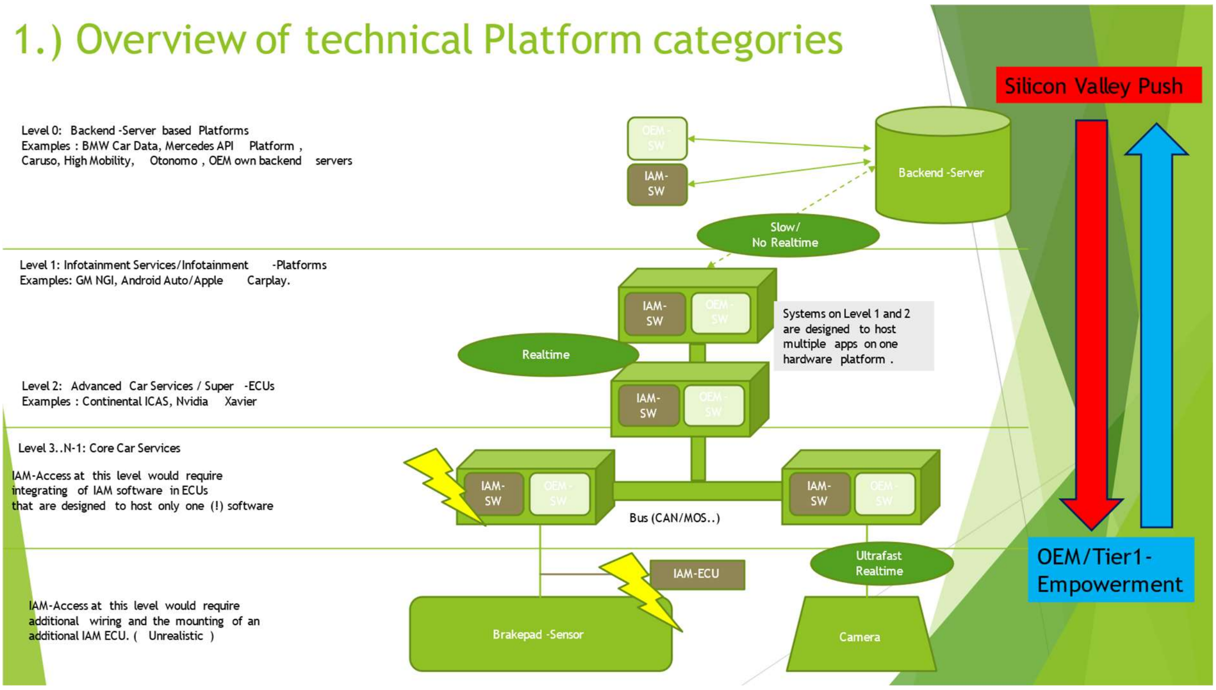



**Figure 4: Levels of Platform categories**

The red arrow indicates the Push from the silicon valley companies from the cloud (ExVe, Level 0) to use their technology know how in higher Levels nearer to the actual Actors and sensors. The blue Arrow depicts the attempt from the big automotive players (OEMs/Tier1s)

[7] https://www.golem.de/news/cariad-vw-soll-software-entwicklungsprobleme-bei-e-autos-haben-2204-164953.html

to evolve from real time control know-how about sensors and actors  (Level N) to more user/cloud centric levels of application.

## 2.5 OEM & Silicon Valley: Cooperation vs Competition

The majority of OEMs find themselves under high pressure due to the success of new OEMs from the US (Tesla, GM) and China (SAIC, BYD). That's why many OEMs have decided to team up with one of two solution providers from California: Nvidia and Qualcomm.

**Chip-Kooperationen bei Automotive-Zentralrechnern**

| Auto-Betriebssystem/Infotainment | |
|---|---|
| **Nvidia** | **Qualcomm** |
| Audi | BYD |
| Hyundai/Kia | GM |
| Mercedes | Honda |
| | Nio |
| | Opel, Peugeot |
| | Renault |
| | Smart, Geely |
| | Stellantis |
| | Volvo, Polestar |
| | Xpeng |

| Automatisiertes Fahren | |
|---|---|
| **Nvidia** | **Qualcomm** |
| Audi | BMW |
| Aurora | Ferrari |
| BYD | General Motors |
| Cruise | Great Wall Motor |
| JLR | Renault |
| Lucid | Volkswagen |
| Mercedes | |
| Nio | |
| Pony.AI gehört zu Toyota | |
| Toyota | |
| Vinfast | |
| Volvo, Polestar | |
| Xpeng | |
| Zoox | |

**Figure 5: Chip-Cooperations from OEMs for Infotainment and Automated Driving.** [8]

The above picture shows which OEM has partnered with either Nvidia or Qualcomm to supply the hardware-chips for either Infotainment/Car-OS or automated driving.

### 2.5.1 Domain Automated Driving

In the article from the German newspaper "Handelsblatt"[9] it is stated that while the hardware might be bought from the supplier, the actual software for the automated driving for Volkswagen would still be developed by the OEM (Volkswagen/Cariad) and a Tier1 (Bosch in the case of Volkswagen). If that would be the case, the key software competence for

[8] https://www.handelsblatt.com/unternehmen/autobranche-volkswagen-setzt-beim-automatisierten-fahren-auf-chips-von-qualcomm/28286598.html
[9] https://www.handelsblatt.com/unternehmen/autobranche-volkswagen-setzt-beim-automatisierten-fahren-auf-chips-von-qualcomm/28286598.html

Automated driving would be with the OEMs and/or Tier1s while the Nvidias and Qualcomms would be "just" hardware suppliers.
A background check on the cooperation of Mercedes and Nvidia raises doubts about the suggested split in deliverables and competences in this cooperation.
"The NVIDIA DRIVE platform includes a full system software stack designed for automated driving AI applications. NVIDIA and Mercedes-Benz will jointly develop the AI and automated vehicle applications that include SAE level 2 and 3, as well as automated parking functions (up to level 4)."[10] .
Apparently Nvidia offers far more, a "full system software stack for automated driving Ai applications", than "just" the hardware.
According to Nvidias Ceo, Nvidia can provide OEM customers with everything needed for automated driving, from "just the computer´" up to an end2end-solution.[11]
The respective video is tellingly called "Nvidia Drive vs. Tesla Full Self Driving" and the OEM (Mercedes) on which cars the NVidia system is running, is not even mentioned in the title.

#### 2.5.1.1 Google & Apple aiming to leapfrog

Two of the biggest IT-companies from the Silicon valley seem to have decided that when it comes to automated up to autonomous driving, leapfrogging the OEMs will be the preferred option.

The Google company for automated driving, waymo, develops a full autonomous "electronical driver".

[10] https://nvidianews.nvidia.com/news/mercedes-benz-and-nvidia-to-build-software-defined-computing-architecture-for-automated-driving-across-future-fleet
[11] https://www.youtube.com/watch?v=UnPaTb0__JU

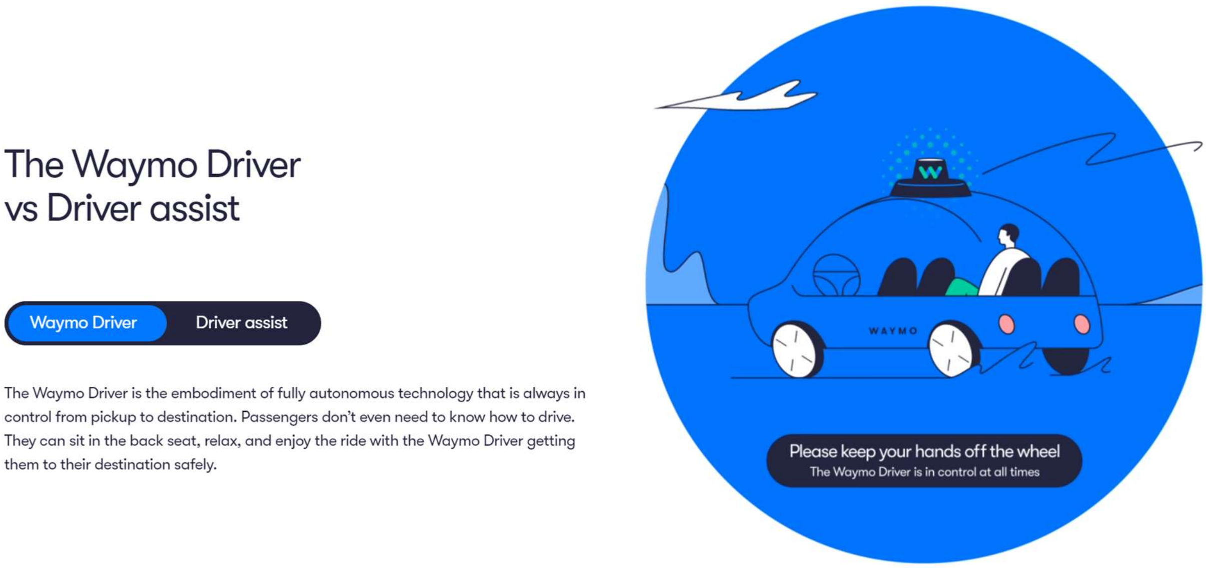



**Figure 6: The Waymo driver**[12]

This rather clear cut approach limits by design the need to integrate too closely with OEM components and other assistance systems.


**Figure 7: Waymo one cars**[13]

Without an in-depth knowledge of the technical details the depicted cars seem to convey the notion that everything the autonomous driver from Waymo needs in terms of sensors, actors, cloud connectivity and software was mounted and retrofitted to rather "normal cars". This

[12] https://waymo.com/waymo-driver/
[13] https://waymo.com/waymo-one/

approach would enable Waymo to install their electronic driver basically in every OEM car that features electronic control of the car's basic functionalities like electronic braking, acceleration, steering etc.

After several twists and turns with their own project for autonomous driving, Apple seem to have decided to follow the same route of developing a fully autonomous vehicle.[14]

**Assumed business models**

It is likely that Google and Apple would follow the business models from their smartphone business. Google could then basically sell or license the complete solution for automated driving to OEMs that lack the power to develop their own solution in the same way they license the Android operating system to smartphone developers like Samsung. This would free them from the need to invest in the expensive car development itself. Maybe they will really develop just one car under their name or buy and rebrand an existing OEM-model as a platform to showcase the abilities of their autonomous driving system.

Apple is likely to follow the saying from Alan Kay, quoted by Steve Jobs "People who are really serious about software should make their own hardware"[15]. In the same way Apple develops and sells own phones (IPhones), own computers (Macbooks) they will try to sell their autonomous vehicle with their software on board under their own brand.

Summing up: When it comes to automated driving, the biggest players from the Silicon Valley have opted for competition with the OEMs to preserve their know-how in this strategical area.

### 2.5.2 Domain Infotainment/In Car OS

The models for cooperation in this domain will be presented in a separate chapter, "OEM solutions offered in combination with Hyperscalers".

## 2.6 Technical Evolution of the car's lifecycle: Merger of markets

The introduction of more and more powerful computer hardware and software components in the software defined vehicle already had significant impacts on needed know how, business models, and

[14] https://www.bloomberg.com/news/articles/2021-11-18/apple-accelerates-work-on-car-aims-for-fully-autonomous-vehicle
[15] https://www.youtube.com/watch?v=XAfTXYa36f4

supplier networks for all stakeholders involved in the process of designing and developing a car in the first place.

Although not so visible at first glance is the drastic change that the technical evolution of the Software defined vehicle brings to the car's lifecycle and the related markets and stakeholders.

### 2.6.1 Different life cycle times

There are three different types of components in the software defined vehicle when it comes down to life cycle times and/or development cycle times:

- Software components (e.g. an App in the infotainment system)
- Hardware components (e.g. the computer on which the infotainment app is running)
- Mechanical/Electrical components (e.g. the cockpit itself, in which the infotainment computer is integrated)

The cycle times increase drastically along this list. As a rule of thumb, a new software version with new or updated functionality can be developed within weeks or months (critical security patches might take only days of development). For hardware components, Moore's law applies in the sense that the computing capacity doubles roughly every two years[16].

That's why most consumer or office hardware (Laptops, Tablets, Smartphones) is replaced roughly every 4 to 6 years. Cars however with all their Mechanical and electrical components have a lifetime of around 12 years[17].

For most cars that lacked the possibility to upgrade either the hardware or the software on the hardware this led to a rather poor experience for the customer when buying either used cars or "new" cars that have been produced for quite some time and for which the hardware and software components were designed, bought and developed years ago. The infotainment experience even in brand new luxury cars was very soon outclassed by the consumer's smartphone experience when he bought himself a brand new iPhone or a premium Android phone.

Technical evolution is going to change this.

### 2.6.2 Over the Air updates

Modern cars (like the smartphones of their drivers) offer the possibility of Over-the-air-Updates (OTA). These software updates are not limited to infotainment software per se, instead, the ability which software on which level can be updated depends highly on the respective OEM's chosen EE-

[16] https://en.wikipedia.org/wiki/Moore%27s_law
[17] https://www.aarp.org/auto/trends-lifestyle/info-2018/how-long-do-cars-last.html

architecture and IT-competence. Tesla e.g. is able to update its battery handling [18]or its Automated Driving software[19].

Commercially, these new functionalities offered via software updates throughout the lifetime (!) of the car can either be sold to the customers for a direct revenue or they are a means to enhance brand loyalty.

### 2.6.3 Cyber Security as a process

The software defined connected vehicle is likely to become a prominent target for cyber-attacks. All the new or updated software components have to follow a rigid testing process to ensure a high level of cybersecurity. However, there is not and likely will never be a perfect process, so there is always a slight possibility that experienced attackers find and exploit security loopholes.

That's why as a first measure, every developer of a software component, might it be the OEM, a Tier1, an IAM stakeholder, must be integrated in the car's Cyber Security Management system to be informed about the issue and to have the possibility to fix the issue with a security patch installed in the car via OTA. But not all security issues can be fixed by software updates alone.

As a second measure, there has to be the possibility to upgrade the hardware and the computing power inside the vehicle to successfully withstand hacker attacks over the lifetime of the vehicle. A slightly simplified example: Professional Hackers have the benefit to be able to use the latest cutting edge server computing power to try to hack a car, e.g. with a brute force attack on the encryption. Let's assume the original hardware on which the software for Cooperative Intelligent Transport was installed is able to securely exchange 40 Messages per second with the car's environment using a 64 bit-key. This might be "unbreakable" via brute force by a server hardware of the development time. But with Moore's law in mind it is not unlikely that after 6 years a server computer with 2*2*2 = 8 times the power can successfully crack the encryption of the car's computer.

To prevent this, there has to be the possibility to exchange at least the encryption computer hardware once, maybe several times (e.g. every 4 years) over the lifetime of the car to keep the car cyber secure.

Cyber security is not a one-off solution in software and hardware that is once installed in a vehicle and prevents the car forever against cyber-attacks. Instead, it's a process with a need for regular updates in software and hardware.

### 2.6.4 The car as a process

The "old car" was developed and sold with a fixed set of capabilities and functionalities. Once sold on the primary market, functionalities where only maintained and repaired by the Aftermarket. There was no substantial enhancement or change in functionalities, big recalls like on the emission scandal were rather few in numbers.

[18] https://driveteslacanada.ca/software-updates/tesla-deploys-emergency-software-update-2022-4-5-12-to-calibrate-lfp-batteries/
[19] https://insideevs.com/news/586924/tesla-fsd-beta-update-many-potential-improvements/

Now the first OEM companies have taken up the idea to constantly upgrade and update the functionalities of the car not only by software or hardware updates and not only limited to security related aspects.

Tesla offers customers an upgrade of their automated driving computer to be able to run more sophisticated and resource demanding versions of their FSD-software[20].

Toyota has started a hardware and software update service "Kinto" that on top of software and hardware updates offers also new security features or interior updates[21].

### 2.6.5 Merger of markets

Ultimately, the technical evolution and the different life cycles and development of the different element groups (Software, Hardware, Mechanical/Electrical components) will lead to a perception of the car as a process and the formerly rather strict borders between primary and aftermarket will start to vanish.

While there will be more business opportunities with all the updates and upgrades for competent stakeholders there is the need that all stakeholders work together very tightly along the same security guidelines to ensure the operational safety and security of the vehicle as the integrated overall system/platform.

[20] https://www.teslarati.com/tesla-fsd-hw3-upgrade-price-update/
[21] https://europe.autonews.com/automakers/toyota-start-hardware-software-update-service

# 3 OEM solutions offered in combination with Hyperscalers

The previous chapter about the "Software Defined Vehicle" offered insight whys OEMs started to partner with Silicon Valley companies for highly software driven use cases like In-Car operating System/Infotainment or Automated driving. While the aforementioned companies like Qualcomm and Nvidia are developing software for autonomous vehicles together with OEMs, the two big players Apple and Google apparently decided to not cooperate with OEMs there.

The picture looks different for the domains of infotainment and on-board Operating systems.

## 3.1 Solutions on Level 1: Infotainment

Here, the push for access to the customer in the car started with the advent of Apple Carplay and Android Auto in the car Infotainment domain (Level 1) already in 2014.

These platforms use screen mirroring/projection technology to display the user interface from a smartphone based application on the car's HMI. The driver can also interact with the application via the car dashboard using the touch controls and knobs of the vehicle .

Today, almost every new model from a larger OEM offers support for Apple's Carplay and Google's Android Auto at least on the car's option list.

(See Apple[22] or Google[23])

Technically, both solutions (CarPlay as well as Android Auto) in their standard version for all developers offer only the use of a rather small subset of the car's actuators and sensors (Screens, phones, Speakers, Knobs).

In addition, to ease the testing for driver distraction and a compliance with design guidelines, the graphical representation of an APP has to follow one of few predesigned app templates, called "entitlements" by apple.

[22] https://www.pocket-lint.com/cars/news/apple/127644-apple-carplay-which-cars-support-it-what-can-it-do-and-how-does-it-work
[23] https://www.android.com/intl/de_de/auto/compatibility/#compatibility-vehicles

| Entitlement | Category |
|---|---|
| com.apple.developer.carplay-audio | Audio |
| com.apple.developer.carplay-communication | Communication |
| com.apple.developer.carplay-charging | EV Charging |
| com.apple.developer.carplay-maps | Navigation |
| com.apple.developer.carplay-parking | Parking |
| com.apple.developer.carplay-quick-ordering | Quick Food Ordering |

**Figure 8: Apple CarPlay App categories & Entitlements[24]**

For Android Auto, the list is almost identical, although the naming is slightly different[25]:

1. Media Apps
2. Messaging Apps
3. Navigation Apps
4. Points of Interest Apps (Formerly subdivided in charging and parking app)

### 3.1.1 Comparison with the 2018 study

In comparison with the 2018 study, the number of available templates has increased, but only to a limited extent. In 2018 Apple CarPlay was limited to Media and Messaging Apps only. Since then Waze as a very popular application has found its way in both solutions probably due to consumer pressure.

Apps using the other categories like EV-charging or Quick Food Ordering up to now are rather limited, one example would be the App from Dunkin ' Donuts

[24] https://developer.apple.com/documentation/carplay/requesting_carplay_entitlements
[25] https://developer.android.com/training/cars

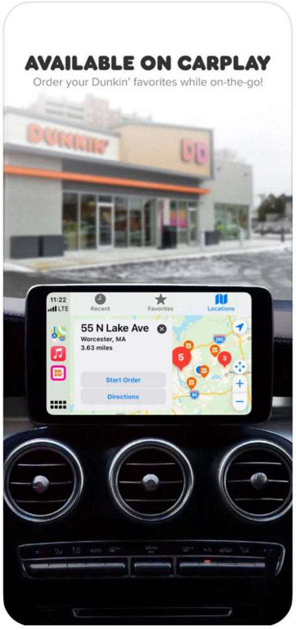



**Figure 9: One example for a quick food ordering template.[26]**

### 3.1.2 Missing templates for Aftermarket applications

Assessing the current set of templates for the two platforms Android Auto and Apple CarPlay and the set of apps that currently support these platforms the absence of aftermarket applications and templates is worth noting. There are no workshop-, Insurance- or Roadside assistance- apps because the templates are missing.

Stakeholders from all sectors discussed the possibilities, pros and cons to encourage Apple and Google to come up with additional templates for aftermarket use cases internally or bilaterally. But up to now there is no publicly available information that either Apple or Google is working on an enhancement of the template set.

A second option to overcome the access barriers for IAM providers would be to use the "design free", open versions of Apple CarPlay and Android Auto. These versions are intended for apps for automakers, who can expose more signals and actors to the system and can render their own User Interface.

[26] https://apps.apple.com/us/app/dunkin/id1056813463


**Figure 10: Seat App from 2016, using the OEM version of Apple CarPlay (taken from 2018 study)**

While this is still possible today, this would require OEMs and IAMs to set up B2B agreements and it would be the OEMs that have to test the final app for e.g. compliance with Driver Distraction Guidelines before they submit the app to Apple/Google.

At the time of this study, no examples of this approach apart from the Seat example are known to the author.

### 3.1.3 A harsh lesson in Digital Ecosystems

It is worth noting that in the early years of the solutions some OEMs like BMW had concerns about letting Google in particular into their vehicles and offered only Apple Car Play support in the early years. From 2020 on, upon requests from customers, BMW conceded to integrate both solutions in future models[27].

For BMW and other OEMs with similar concerns this was a harsh lesson in the principles of Digital Ecosystems.

[27] https://www.autobild.de/artikel/bmw-startet-android-auto-2020-smartphone-infotainment-google-16170705.html

When the average European spends roughly around 45 Minutes per day in the car[28] but around 230 Minutes with his smartphone[29], then the customer is around 5 times longer in the Digital Ecosystem of Apple or Google than in that of the car. Obviously a significant number of Smartphone users would rather select a different car model with their favorite Smartphone family integrated than switch their phone family (e.g. from Google To Apple) just to buy a car model that features only one type of smartphone integration.

In these years (2014-2019), many OEM representatives pointed out frequently that the OEMs have the technical means to control Google and to shield customer data as well as OEM business models from the influence of Google.

Technically, this was correct. Google did not manage to "hack it's way" into the car's infotainment system. Google did not need to do so. Google enveloped the customer of the car so neatly in its Ecosystem with all the Apps and Digital Assistants that the customer decided his future vehicle had to support Android Auto. The OEMs had to just build it in to not lose the customer.

## 3.2 Solutions on Level 2

Apple is currently not offering software platforms below Level 1 (Infotainment, CarPlay), although they announced during the time of the study the new CarPlay Version for IOS 16 that seems to aim “deeper” in the car and will be presented later in the study).Instead, for "deeper car access" they seem to focus on building ultimately their own autonomous vehicle (described in chapter "Software Defined Vehicle").
Out of the other big tech companies:

- Microsoft,
- Amazon,
- Google,

Google is by far the most relevant player today when it comes down to Level 2 solutions and On-Board operating systems on that level.
Microsoft is an important supplier for most OEMs when it comes down to using their cloud technology Microsoft Azure. Onboard, OEMs like GM partner with Microsoft to benefit also from their know-how in Edge technology[30]. Given the fact that GM partners with Microsoft and Cruise, their dedicated Autonomous Driving company with 9 years of experience it is however likely that Microsoft is in this partnership not comparable in terms of Automated

[28] https://ec.europa.eu/eurostat/documents/3433488/5298273/KS-SF-07-087-DE.PDF/0d50ff3c-a042-4c49-85e8-5333c92a7186
[29] https://de.statista.com/statistik/daten/studie/1186676/umfrage/durchschnittliche-taegliche-smartphone-nutzung-nach-apps/
[30] https://news.microsoft.com/2021/01/19/cruise-and-?gm-team-up-with-microsoft-to-commercialize-self-driving-vehicles/

Driving Knowhow with Nvidia in the partnership with Mercedes. For the purposes of Level 2 operating systems, also GM is in a partnership with Google[31].
Amazon AWS offers a range of solutions for OEMs, from cloud to edge computing. They offer integrated solutions to support the software defined vehicle, including their IoT FleetWise product, which can collect, transform and transfer vehicle data to the cloud in near real time. This solution required the use of the VSS specification (see section 4.2.2.2).

Amazon teamed up with Stellantis at the start of 2022 to deliver the STLA Smart Cockpit solution[32] and roll it out from 2024 on.
The move came as a bit of a surprise because originally Stellantis wanted to cooperate with Google and use Android Automotive[33].
Thus, while for a time it seemed that Google would be the only one interested in a deeper access in the vehicle with Apple just interested in the projection mode it now looks as if the "battle for the in-car operating system" would include at least 3 Silicon Valley companies with Apple (CarPlay 16), Google (Android Automotive) and Amazon (e.g. with the Stellantis Smart Cockpit).

### 3.2.1 Google Android Automotive

Android Automotive as a level 1+ or Level 2 operating system is currently the operating system that is likely to see the most widespread usage in future car models.
First showcase in the Polestar 2 in 2020, many and big OEMs have announced to feature it in future Car Models.

#### 3.2.1.1 Functional abilities

Android Automotive is the car's operating system on level 2, it does not depend on a connection to an Android Phone of the driver like it would be the case for Android Auto, its level 1 cousin depicted previously.
For technical in depth-details, please see the official documentation[34].

---

[31] https://www.t3.com/news/android-automotive-and-over-the-air-updates-coming-to-gm-cars
[32] https://www.stellantis.com/en/news/press-releases/2022/january/amazon-stellantis-collaborate-on-software-solutions
[33] https://www.auto-motor-und-sport.de/tech-zukunft/stellantis-opel-peugeot-fiat-amazon-infortainment-2024/)
[34] https://source.android.com/devices/automotive

We partnered with Google to develop the world's first car with a built-in Android-powered operating system, making life on the road easier, safer, and more intuitive.


**Figure 11: Android operating system in a Polestar 2.**[35]

As there is no alternative or additional OEM operating system on-board, Android Automotive offers "By design" access to all data points and functions that a driver interacts with using the Car's Cockpit and it's controls, e.g.:

- Seating,
- Air Condition,
- Charging,
- Navigation,
- Music,
- Integration of Google Assistant
- Reaction on car's problems (DTCs etc.).
- ......

As it is the case for the  Android running on an phone, the architecture features a similar interplay of the three stakeholders:

1. Android (The provider of the operating system and some core apps)
2. OEM (the developer of the hardware on which the Operating System runs and a service provider of multiple OEM apps)
3. 3rd party (Developers of services/apps)

[35] https://www.polestar.com/global/polestar-2/interior/software-apps/google-built-in/

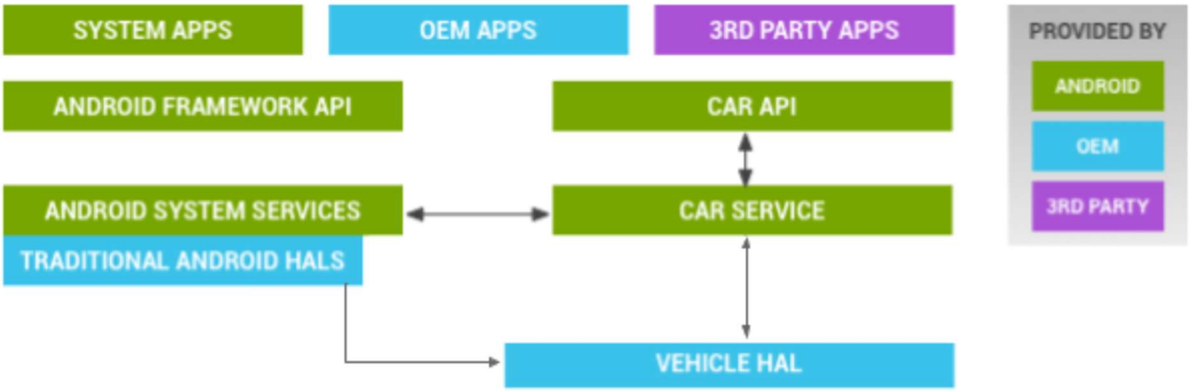



**Figure 12: Basic Architecture of Android Automotive**

### 3.2.1.2 Enforced Standardization via HAL

The first task of any OEM who wants to use Android Automotive in his vehicles is to implement the HAL (Hardware abstraction layer)[36].

The basic idea behind this abstraction is a software standardization for the Apps that run on Android Automotive. No matter what temperature sensors actors or climate control ECUs an OEM decides to use: Google mandates that the access from the app to the temperature has to be exactly the same (!) on every (!) vehicle that runs Android Automotive.

[36] https://source.android.com/devices/automotive/vhal/properties

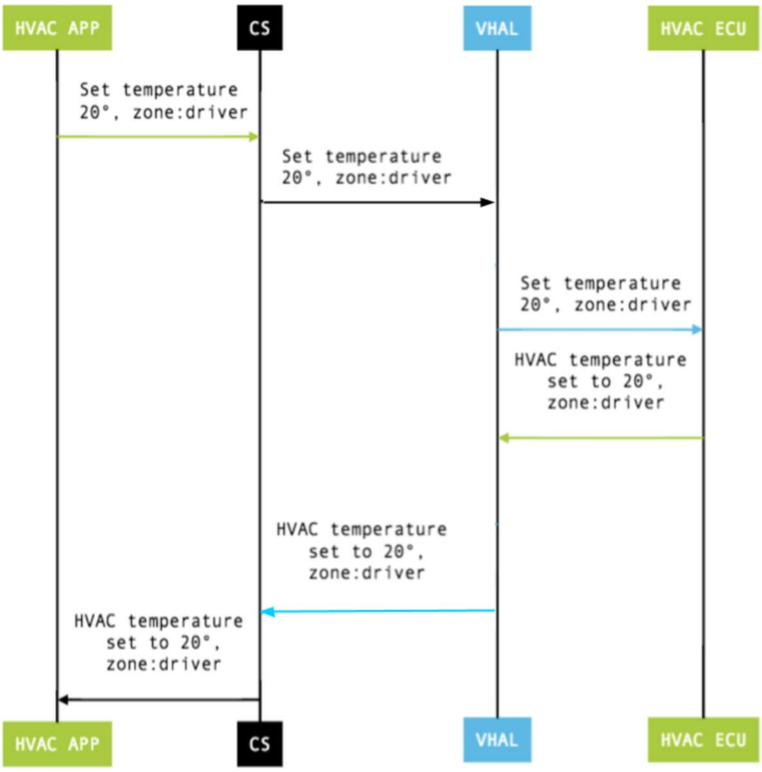



**Figure 13: Setting the temperature in a standardized way via Android Automotive[37]**

A list of the standardized properties of the car that can be set, read or observed can be found on https://developer.android.com/reference/android/car/VehiclePropertyIds.

These around 130 standardized properties (controllable data points) are just one domain of standardization. Google added APIs to control the camera systems, the displays, Audios, Bluetooth, car controls like knobs etc.

#### 3.2.1.3 Google Standardization summary: It's all about and around the driver

In a nutshell the whole interface towards the driver is standardized by Google as well as everything the driver needs to communicate with and control his car during operation.

It is a typical Google Approach: Care about the consumer first and then later for technical details of all the services and products that the consumer selects or interacts with on the Google platform. Keeping in mind the levels presented in a previous chapter this is the typical "Top Down approach" (From Customer to Technology) of Silicon Valley.

Note: typical standardizations of the automotive sector follow the opposite direction. They are typically focussed on standardizing "things" (e.g. OBD-Connector) or rather low level protocols like Unified

[37] https://source.android.com/devices/automotive/vhal/properties

Diagnostics Services and only recently standardizations like Autosar Adaptive or Common Vehicle Interface Initiative aim to standardize on Levels up to the consumer interface.

#### 3.2.1.4 Access for third party apps on Android Automotive (Polestar)

The success of the large scale digital ecosystems of e.g. Google and Apple is based on a strong community of 3rd party app developers and a large set of 3rd party apps the user can choose from.

For Polestar, the third party app "easy park" was the first to be available for polestar drivers to download and use:

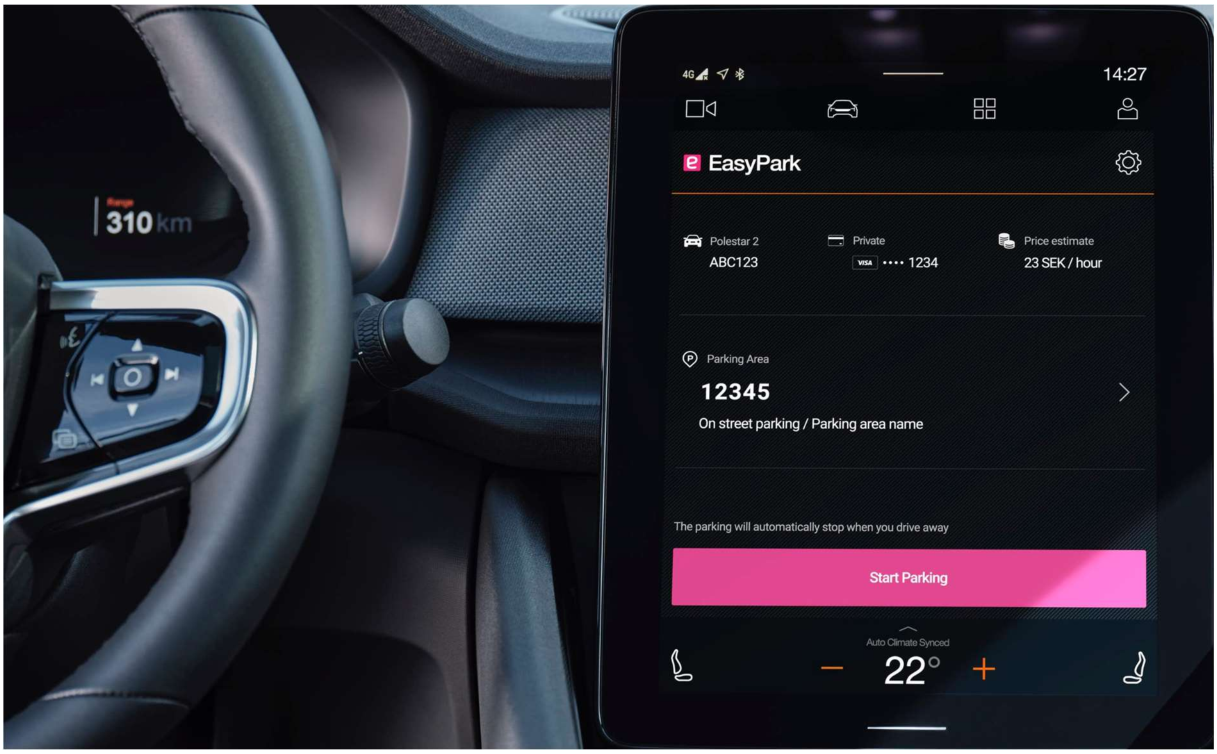



**Figure 14: EasyPark App inside a Polesatr 2[38].**

#### 3.2.1.5 OEM still the B2B partner for App developers

The development process on the polestar website[39] describes the interactions and roles needed for the OEM and the 3rd party App developer. All legal and commercial issues are first between the OEM as the responsible owner of the platform "car" and the 3rd party App developer. They sign the development contracts, they agree on the business model. It is the OEM who does the final testing for safety and security and access to the App store is only granted via the OEM. So while Google is responsible for all

[38] https://www.polestar.com/de/news/a-smarter-way-to-park/
[39] https://www.polestar.com/us/developer/get-started/emulator/

the technical details, legally everything is between the OEM and the App developer, Google remains here "in the background" like any other Tier1 supplier that offered services or products to an OEM.

### 3.2.2 Android Automotive – Technology & Business Models

Android is recognized by most people as the Operating System on their smartphone. The family of phones produced by brands like Samsung or Huawei is often called "Android-Phones" in competition with Apple's IPhones although Android is a software, an operating system and should thus be compared to IOS, the operating system of the IPhones.

When the operating system "Android Automotive" and the influence of its developing company Google on the partners from the automotive sector are investigated, not only technology details, but also licsensing rights, ownerships, ecosystem size have to be taken into account to explain the impact on the different business models.

#### 3.2.2.1 The Technology behind Android - Linux

The operating system Android is based on the open-source operating system Linux[40], in the sense that it uses the Linux Kernel and some other "typical Linux" components. Linux as an operating system has become a dominant force as a server operating system as well as an operating system for "intelligent hardware", from Phones to Smart-TVs, set top boxes or other IOT-devices. Linux is developed by a community, so there is no company that "owns" Linux and it can be used without license fees.

#### 3.2.2.2 Android Automotive as an Open Source Technology

Google as the developing company of Android and the "extended" version Android Automotive have decided to put the operating system Android Automotive itself also under an Open Source License[41] like the "normal" Android[42].

So basically an OEM that has reached a certain level of maturity in software design and integration can opt to install Android Automotive in his cars without involving Google at all and without entering into any business relationship with Google contractually.

[40] https://en.wikipedia.org/wiki/Linux
[41] https://source.android.com/devices/automotive/start/what_automotive
[42] https://source.android.com/

#### 3.2.2.3 The business elements of Android Automotive - The Google Automotive Services

Despite the fact that the large portion of Android and Android Automotive is open-source, Google has developed a set of software components that interact with the operating system and that require a license agreement from the implementing OEM with Google, if the OEM wishes to use these services.

The set of component is called "Google Automotive services" (GAS) and includes e.g.:

1. Google Maps & Navigation
2. Google Assistant
3. Google Playstore

A brief architectural overview is presented below:

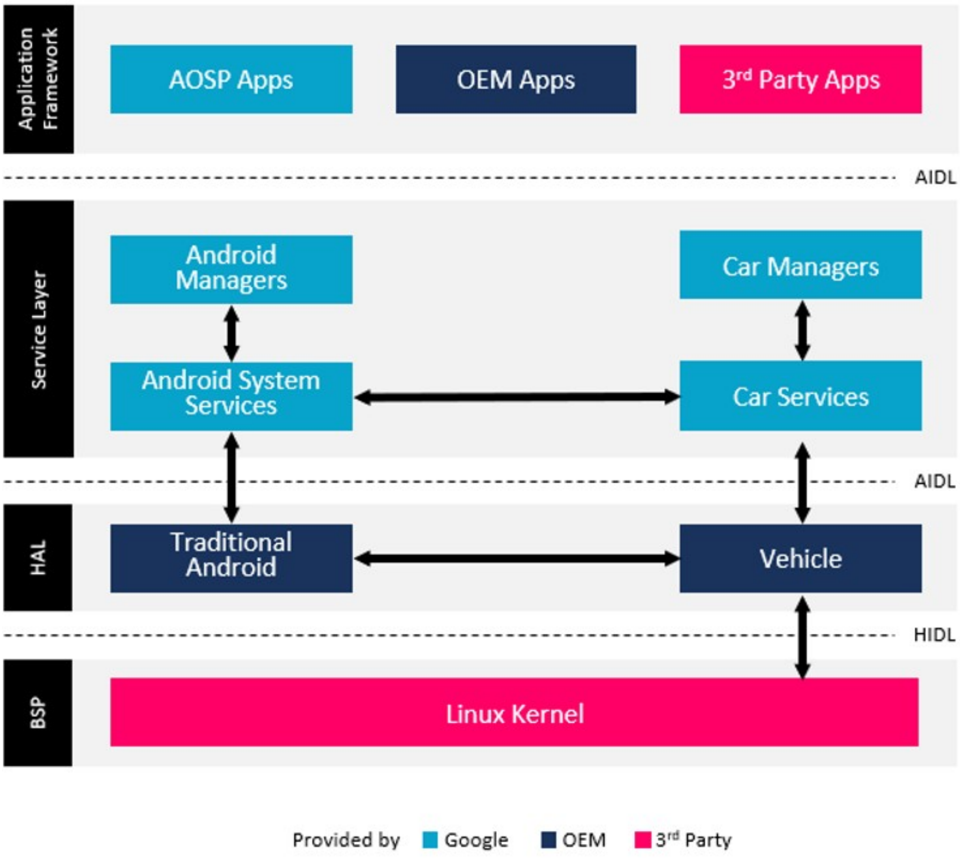



**Figure 15: Google Android Automotive In Vehicle components and layers[43]**

As a rough classification these elements are the user and business focused components where Google is most interested in. By having users navigate with their solution Google not only can improve the navigation service by using this real time data. They improve their know how about user behavior further and as a result can claim that their user focused advertising in Google Search and Google Maps

[43] https://www.androidautomotivebook.com/android-automotive-embedded-os-whitepaper/

improves and thus increases in worth. The same ideas are behind the exclusive usage of the Google Assistant. The more users train the speech detection neural network inside the Assistant, the better the quality of the solution itself, the higher the chance that more and more users will prefer it over competing solutions from either Silicon Valley ("Hey, Alexa.." from Amazon") or from OEMS ("Hey Mercedes.." from Mercedes). Should Google also manage to establish the Google Assistant as the predominant solution for Digital Assistants like they did for the Google Search, then they position themselves in a real gatekeeper position between consumers and customers and can charge high fees for advertising.

With the control over the Play store, Google offers the customer the access to the largest collection of apps in the Android world developed by the majority of Android app developers. Here, Google can track every sale and can collect their platform provider fees. Only apps adapted for automotive use are offered to OEMs, so the full Play Store catalogue is not available.

Annotation: The exact same approach is taken by Google in the Smartphone world. The usage of the Android operating System is free of charge as an open source offer, however, the most popular applications have to be licensed as the Google Mobile Services, containing e.g.[44]:

Google Search

Maps, Navigation

Google Play

..

When using Android Automotive for the car sector, an OEM has now two options to choose from: with the Google Automotive Services and thus with Google as a company or without them.

#### 3.2.2.4 Business Model 1: Integration in Google's commercial ecosystem

Polestar was the first company to introduce Android Automotive with the Google Automotive Services in a car in 2020 with the Polestar 2. Since then, a number of OEMs have decided that they want to integrate Android Automotive with GAS in their future cars[45].

What this move comes down to for an OEM is basically an integration in the consumer and commercial Google Digital ecosystem. On the pro side, the vast majority of users is looking for apps in the Google Play Store, in return the majority of app developers will develop their apps and try to integrate them at this location. So for the motoring customer in the car, a visit at the Google Play Store to search for attractive apps to spend his money on is a rather natural move. In addition, Google will take care of the app store, manage security and notifications for app developers etc. A special and privileged treatment can be expected for the first adopters from the OEM side (like Polestar, Volvo) because Google wants to promote its solution.

[44] https://en.wikipedia.org/wiki/Google_mobile_services
[45] https://en.wikipedia.org/wiki/Google_mobile_services

On the downside, inside the Google ecosystem the OEM has limited influence on the overall design of the ecosystem guidelines and is in a rather weak position when it comes down to negotiating shares for app- or in-app-sales.

#### 3.2.2.5 Business Model 2: Proprietary Commercial ecosystem on top of Google's technical ecosystem Android

An OEM or a technology partner can use the Open Source Versions of Android or Android Automotive without the Google Automotive Services and without a contractual agreement including payment obligations with the company Google.

Amazon e.g. did exactly that when they developed their own version of Android, called Fire.OS.[46]

In this move Amazon used the evolving versions of the open source Android and replaced the Google Mobile Services with their own solutions. Instead of an integration with the Google Cloud they integrated the Amazon Web Service Cloud, the Google Play Store was replaced by the Amazon App Store, the Kindle Store etc.

This approach basically creates a Google independent proprietary commercial ecosystem on top of the Android Automotive software itself, the technical digital ecosystem.

So when a press article stated that Stellantis announced a move away from Google (Android) to Amazon[47] this was only partially correct.

The Amazon Fire OS inside is still based on Google's Android software, albeit in the Open Source Version and only vital elements from the Google Automotive Service software list are replaced by Amazon's own developments, so the Google Digital Assistant is replaced by the Amazon Pendant, Alexa and the App stores are own developments not operated by Google.

Apparently, the idea of setting up an own commercial Digital ecosystem sounds appealing to other OEMs. There is e.g. already a tech player Faurecia that together with Aptoide, another provider of an Google independent app store for Android Apps, offers OEMs like BMW the App store operation without Google[48]. Also Mercedes will use Aptoide[49].

At least as an interim solution, Volkswagen will also follow the same approach with a non-Google operated playstore[50].

[46] https://en.wikipedia.org/wiki/Fire_OS
[47] https://www.auto-motor-und-sport.de/tech-zukunft/stellantis-opel-peugeot-fiat-amazon-infortainment-2024/
[48] https://www.faurecia.com/en/newsroom/faurecia-delivers-app-store-solution-future-bmw-group-vehicles
[49] https://www.yahoo.com/video/faurecia-aptoide-signs-mercedes-infotainment-060000777.html
[50] https://www.auto-motor-und-sport.de/neuheiten/vw-betriebssystem-os-2025-android-automotive/

They will use the Open Source Software of Android and will try to set up the commercial ecosystem themselves or via chosen suppliers like Faurecia/Aptoide.

This decision eliminates the need to set up contractual agreements with Google and pay Google license fees and/or platform sales fees on the App/Play Store. Instead Mercedes, Volkswagen and BMW are replacing Google in the role of the App store provider by Faurecia/Aptoide. They still need alternatives for all the other elements of the GAS, e.g. a Digital Assistant, Navigation etc. to compete "functionally" with an Android Automotive with all the GAS from Google.

The main issue however might again be not a technical one, instead a commercial consideration.

#### 3.2.2.6 Business Models: Summary and risks

Looking at the involved OEMs Stellantis, Volkswagen, BMW, Mercedes, Gm, Ford, Renault, Mitsubishi, Volvo etc., that use Android Automotive either with the commercial ecosystem Google or without it, it remains clear that Android Automotive will play an important role, maybe even a sort of de facto technical standard in the foreseeable future in the Automotive sector.

Examples of OEM business models for Android Automotive

| OEM | Open Source/Licensed Android Automotive | App Store Provider |
|---|---|---|
| BMW[51] | Open Source | Faurecia |
| Mercedes[52] | Open Source | Faurecia |
| Volkswagen[53] | Open Source | T.B.A. |
| Volvo, Polestar[54] | Licensed | Google |
| Stellantis[55] | Open Source based Fire OS | Amazon |
| Ford[56] | Licensed | Google |
| GM[57] | Licensed | Google |
| Renault, Nissan, Mitsubishi[58] | Licensed | Google |

Since now multiple OEMs have decided to adopt Android Automotive only in the open source version, it will be interesting to see, how this market evolves. It seems however undisputed that Android Automotive will be the dominant solution for the in-vehicle Operating System. Harman e.g. estimates tat in five years 60% of the infotainment systems will feature Android Automotive.[59]

**Next Harsh lessons on ecosystems: Size does matter and content is king**

In the ideal world of the OEMs, they would enjoy the benefit that someone else (Google) develops with its own resources the operating systems of their cars, keeps it functionally up to date and constantly cyber secure with a sound cybersecurity management system and all this for free.

51 https://www.faurecia.com/en/newsroom/faurecia-delivers-app-store-solution-future-bmw-group-vehicles

52 https://www.faurecia.com/en/newsroom/forvia-partners-app-integration-with-major-german-car-manufacturer

53 https://9to5google.com/2021/07/16/volkswagen-android-automotive-adoption-plan/

54 https://arstechnica.com/cars/2022/01/with-qualcomm-and-android-the-polestar-3-is-more-smartphone-like-than-ever/

55 https://www.stellantis.com/en/news/press-releases/2022/january/amazon-stellantis-collaborate-on-software-solutions

56 https://arstechnica.com/cars/2022/04/ford-delays-switch-to-android-automotive-until-2023/

57 https://www.theverge.com/2019/9/5/20851021/general-motors-android-auto-google-infotainment

58 https://techcrunch.com/2018/09/17/google-partners-with-renault-nissan-mitsubishi-to-put-android-into-millions-of-vehicles/

59 https://ignitedevelopers.harman.com/emergence-of-android-in-automotive

They would only have to pay a set of rather small suppliers to come up with Alternatives for the GAS, operate their app stores and the OEMs can thrive on the revenues to come from app-sales or in-app sales.

This is rather unlikely for four reasons:

a.) Google has as the developer of the technical ecosystem Android Automotive always an advantage in time. They know what features will come next, they steer the development. All the independent App store providers will just have to follow.

b.) If a typical Google App store contains more than a Million apps, why would customers and developers visit the "alternative app" store with just around 250 apps (e.g. Faurecia[60]) in the first place? It remains to be seen, if the up to now rather small amount of car specific apps will increase faster in the Google Play Store or in the OEM operated app stores. Despite the same technology of the app store behind, consumers visit these stores because of the content inside. “Content is king.”

c.) Possible Quality advantage of GAS services. To compete with Google's GAS, the alternatives of the OEMs must have an equal or better performance. However, developing Navigations that are better than Google Maps or Digital Assistants with a better speech recognition as the Google Assistant is not an easy task. Should OEMs fail to deliver here, Customers might turn away from their cars because of a poor user experience.

d.) High risk of Google using its control over the technological ecosystem Android Automotive in the future. The indirect control of Google even without the use of GAS should not be underestimated. Five years from now a majority of new cars is likely to feature Android Automotive inside, either with or without GAS. This would give Google as the developers for Android Automotive an immense power. They could decide from one day to the next to just stop publishing future versions of Android Automotive as Open Source and instead sell and support new versions only under b2b license agreements enforcing the use of GAS. It would be then up to the OEMs and suppliers like Faurecia if they feel powerful enough to conduct the evolution of a new version of Android on their own. Otherwise, they would be forced to enter into this agreement.

## 3.3 Summary OEM Solutions offered in combination with Hyperscalers

In an attempt to keep up with the technological frontrunners like Tesla, most OEMS have decided to team up with Hyperscalers like Amazon or Google when it comes down to On-Board operating systems for the levels 1 and 2.

The most prominent system here is Google's Android Automotive that is either installed publicly like on the Polestar2 or "under the hood", may this be the Amazon-Stellantis cooperation where Amazon uses

[60] https://www.yahoo.com/video/faurecia-aptoide-signs-mercedes-infotainment-060000777.html

Android or several OEM-operating Systems like VW OS or MB OS that also are Android based when looking at the supporting App Store from Faurecia for "Android Automotive Apps".

In terms of technical standardization, Google is thus establishing a rigid software standard for the industry. Already 10 out of 15 Top OEM already have selected Android Automotive as the future operating system[61].

Technically, the platform and the process design around it is designed from the ground up to operate and integrate 3rd Party-Apps, so there is no technical argument that would prevent 3rd party apps from the car at least in that levels (see. e.g. Polestar EasyPark App).

On the commercial side, OEMs try to establish an economical ecosystem on top of the Google technical ecosystem by establishing their own app stores with their own platform fees.

However, due to market laws and principles of digital ecosystems there are severe doubts that this approach will be successful in the long run.

[61] https://www.all-electronics.de/veranstaltungen/automobil-elektronik-kongress/die-highlights-des-25-automobil-elektronik-kongress-tag-2-641.html

# 4 Standardization for Vehicle Data & Functions

Standardization plays a vital role in the creation of successful digital ecosystems. Only if digital services like apps can rely on and/or be built on top of a standards, a service developer can reach a high amount of customers for his service. To give some examples: Today more than a billion websites are online[62] and accessible via a variety of web browsers thanks to the standardization of et. al. the "Website languages" HTML and Java Script.

Out of the more than 6 Billion Smartphones in use today[63], roughly 70% are Android Phones, roughly 30% are iPhones[64].

Thanks to the very strict standards a creator of a current Android App can basically offer his service unchanged in just one version to around 3 billion customers worldwide, a developer of an iPhone app to more than 1 billion customers, even if users of outdated versions are taken into account.

If the European car sector wants to replicate this success and really create a similar digital ecosystem, then the creation of a comparable level of strict (1) standardization of data, functions, development guidelines, security requirements etc. across (2) brands, make and models is a mandatory prerequisite from a technical and even more from a commercial perspective. A "weak" standard (1), where implementing OEMs can decide, which elements in terms of e.g. data and functions are supported on a given vehicle, brand and model will cause developers to either focus on the rather small common subset of the standard supported on all models or will lead to a complete rejection by developers. The same holds true for a strict standard that is only supported by either a specific model within a brand or only within a small brand in terms of total car sales (2). Such a standard will never reach a customer base broad enough to attract enough developers.

When standards are discussed in the automotive industry it is worth to distinguish two "types" of standards:

1. Access- Standards: The "weaker" form of standardization
2. Standardizations by name: The "stricter" form of standardization

The aforementioned standard of HTML, Java Script, IOS or Android are examples of the stricter form while the automotive sector has been accustomed more to the "weaker" Access standards and is now shifting more towards "stricter" types.

## 4.1 Access Standards

To explain an access standard, assume that two different vehicles of potentially different OEMs both have a supported data point "fuel level".

[62] https://www.statista.com/chart/19058/number-of-websites-online/
[63] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/
[64] https://gs.statcounter.com/os-market-share/mobile/worldwide

However, OEM 1 has decided to implement this datapoint in different applications from different Tier1s and in different ECUs than OEM 2. Even the naming inside these applications might be different (e.g. "Fuel Level" vs. "Tank Percentage"). To make this data point at least "accessible" to outside diagnostics software, the two OEMs agree on an access standard. A very minimal access standard could contain two functions:

1. getListofSupportedDataPoints()
2. getDataPointById(int ID)

OEM1 might then elect to number "Fuel Level" as a data point with the Id 5 and specify it in liters while OEM 2 might decide to number "Tank Percentage" as data point with the ID 18 and specify it I percentage from 0..100.

As a consequence, the developer of the diagnostics software has then to:

a.) retrieve (and potentially pay for) the information on the implementation of the standard for every vehicle his service should run on.

b.) Build and test specific adoptions for each vehicle.

For "small and deep" diagnostics functionalities (e.g. triggering EGR valves) that are "by design" to a large extent OEM specific, the approach of "just" an access standard is understandable and acceptable.

For a rather "broad and shallow" use case like a gas station recommendation app that is only interested in retrieving the fuel level and advising the next gas station to the driver it is just not commercially viable (while of course technically possible) to retrieve something as simple as the fuel level in such a burdensome and OEM specific way.

Nonetheless, as mentioned before, access standards have their merits when it comes to deep access and can be found "offboard" as well as "onboard" in current and planned vehicles.

### 4.1.1 ExVe Access Standards

The ExVe standards are a suite of ISO standards covering 20077, 20078 and 20080. Basically the 20077 describes general concepts of the Extended Vehicle, 20078 details how to retrieve data elements ("resources" in the ISO standard) in OEM specific numbering, ISO 20080 describes remote diagnostics based on an OEM specific numbering of diagnostics relevant elements like ecus or functions inside ECUs.

#### 4.1.1.1 ISO 20078

In the ISO-ExVe suite and in the ISO 20078[65] data elements are called and exposed as "resources". ISO 20078 describes how these data elements could be queried and retrieved via webservices. Typical for an access standard, resources/data points are identified by OEM specific numbers (called Resource identifiers) as in the fuel level example.

[65] https://www.iso.org/standard/80183.html

Up to now there is no mandatory set of predefined data points/resources required as it is the case today e.g. for the legally mandated Emission-OBD-identifiers[66].

#### 4.1.1.2 ISO 20080

The ISO Standard ISO 20080[67] is a remote diagnostics standard. It’s basic functionalities are a subset of functionalities covered in on-board diagnostics standards like UDS[68].

In terms of “strictness” and “completeness”, ISO 20080 in it’s current 2019 release still has some issues.

Advanced diagnostics functionalities like “Upload/Download of data” from the UDS standard are left out completely in the scope of the Standard and for the advanced elements within the standard that go beyond a pure read access (The Use Cases 9-11) no consensus could be reached between the OEMs as offering parties, so not even an access standard was possible here.

**Table A.2 — Mapping of use cases to REST APIs**

| UC | Use case name | REST API | Resource(s) | Comment |
|---|---|---|---|---|
| 01 | Use case discovery | resourceReadouts | Not applicable | |
| 02 | Identify ECUs installed in the vehicle | ecuReadouts | ECU Readout | |
| 03 | Read Diagnostic Trouble Codes (DTCs) | dtcReadouts | DTC Readout | |
| 04 | Read readiness codes | readinessCodeReadouts | Readiness Code Readout | |
| 05 | Read DTC snapshot data | dtcSnapshotReadouts | DTC Snapshot Readout | |
| 06 | Read selected diagnostic parametric dynamic data | parameterReadouts | Parameter Readout | |
| 07 | Read malfunction indicator status | malfunctionIndicatorReadouts | Malfunction Indicator Readout | |
| 08 | Clear DTCs | clearDtcJobs | Clear DTC Job | |
| 09 | Adjust the settings of a selected system | Not applicable / No standardized API due to differences between offering parties. | Not applicable | System setting input and result are offering party specific. |
| 10 | Activation of actuators | Not applicable / No standardized API due to differences between offering parties. | Not applicable | Actuator input and result are offering party specific. |
| 11 | Activate a self-test routine | Not applicable / No standardized API due to differences between offering parties. | Not applicable | Self-test input and result are offering party specific. |

**Figure 16: From Annex 1, ISO 20080:2019, Use Cases and agreed/disagreed mapping to a standard RestAPI**

[66] https://en.wikipedia.org/wiki/OBD-II_PIDs
[67] https://www.iso.org/standard/66979.html
[68] https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

Up to now, the number of implementations by OEMs seem to be rather small. Mercedes offers a subset of the ISO 20080 use cases on it's developer portal[69].

An advanced version including activation of routines like "honk the horn" was tested during the ExVe Proof of Concept in 2018 on an experimental car from a different OEM but up to now hasn't made it to the production state. As of May 2022, also three popular data marketplaces (high-Mobility, Caruso, Otonomo) don't support it.

Summing up, ISO 20080 was a starting point for remote diagnostics but has yet to reach stricter standardization and a broader base of implementation. Another issue of 20080 is the missing level of detail how to involve the driver (in his role as "remote diagnostics facilitator" in the standard) in carrying out potentially safety/security critical operations. Without an Onboard-HMI described in the standard there is just no way to ensure that it is really the driver in the vehicle that confirms that it is safe for the passengers in the car to e.g. "close the windows" via a remote command.

### 4.1.2 Onboard Access Standards

Inside the vehicle on-board access standards have been popular especially for the "deep" diagnostics access. Currently widely used is UDS, due to technological evolution SOVD is currently under development as a potential successor.

#### 4.1.2.1 UDS

The Unified Diagnostic Services (UDS[70]) are an access standard that is based on a view of the car as essentially being a set of small computers (ECUs) that contain functions and data points identified by OEM specific numbering. For diagnostic purposes, data can be loaded up and down to different Ecus, ECUs can be queried about their status and problems and functions can be executed to e.g. clear DTCs, conduct a self-test routine etc.

Today, it is widely used for diagnostics in OEM as well as IAM workshop in a scenario where one diagnostics tester software is connected to the car.

#### 4.1.2.2 SOVD

Technical evolution has made additional functionalities and updates on UDS functionalities necessary.

The Service-Oriented Vehicle Diagnostics[71] is a new standard development by ASAM that is as of June 2022 under public review and addresses many of the new challenges for diagnostics.

a.) Diagnostics scenarios: While UDS focusses on the Workshop scenario, where one tester is attached to the car and the mechanic conducts the diagnosis ("Proximity diagnostics"), SOVD covers also the "Remote Diagnostics" where a roadside assistance or diagnostics call center

[69] https://developer.mercedes-benz.com/products/remote_diagnostic_support/docs#_remote_diagnostic_support
[70] https://en.wikipedia.org/wiki/Unified_Diagnostic_Services
[71] https://www.asam.net/project-detail/sovd-service-oriented-vehicle-diagnostics/

interacts with the car and the "On-Board Diagnostics", where one or more diagnostics application monitor and control the car's state permanently.

b.) Accounting for High Performance Computers – UDS was focused on dedicated Ecus, while SOVD is designed to work also with High Performance Computers with multiple applications running simultaneously.

c.) Support for OTA – SOVD contains functionalities to deal with Over the Air updates of car software.

d.) Enhanced coverage of security and logging requirements: SOVD supports authentication of diagnostics client software as well as logging to allow assigning of responsibilities ("Who did what when to the car?")

e.) Multi-Client-support: In the UDS model, only one tester was attached to the car via the one and only OBD-port. In the modern world, several diagnostics client might seek access to a car's element at the same time. An independent tester in the workshop might want to check an ECU status at the same time that an onboard-diagnostics app from the OEM or another IAM tries to access the same resource. SOVD takes this account by incorporating "locking and releasing" of elements.

Summary: SOVD can be viewed as the "Access standard of the future" for diagnostics, covering on-board as well as offboard-access. Nevertheless, it is still – by design – an access standard. A developer of a diagnostics software has to get the specific documentation for each vehicle from the respective OEM and has to test his solution also on all different vehicles.

## 4.2 Standardizations by Name

In Access Standards, the execution of a dedicated function or the setting/reading of a specific data point requires the knowledge about the OEM specific implementations and OEM specific numberings of the functions and data points. Standardizations by name are stricter. There is exactly one name (or in general : one identifier) that identifies a function or a data point across all vehicles of all OEMs supporting the standard. A developer of a tank station app would thus have to implement just one way to retrieve the fuel level and just one way to set the navigation target to the selected gas station after the display of the warning "fuel level low" and could then offer his services to all customers of vehicles that support this Standard by Name.

### 4.2.1 Annotation: Legally mandated Standards "By Name"

The cross-OEM-support for standards require that by definition any authority use case that affects all OEMs and all vehicles will be a digital "By Name- Standard".

#### 4.2.1.1 Past example: E-OBD

To be able to check the emission behavior on all vehicles with just one device and one software on this device, all OEMs must support the list of mandated Identifiers for emission related data points[72].

See below an example for the engine speed, measured in rpm.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0C | 12 | 2 | Engine speed | 0 | 16,383.75 | rpm | $\frac{256A+B}{4}$ |

**Figure 17: Example for "by Name" or "by cross OEM unique identifier" referenced data point**

As an explanation: the Engine speed on every vehicle can be retrieved by asking the car for the data point with the PID 12 (decimal), or 0C (hex coded). It will be coded in two bytes, A (High Byte) and B (Low Byte). Two Bytes can represent positive integer values form 0 to 65535. The formula (256*A+B)/4 will thus yield the given min value for rpm 0 (in this case both bytes have the value 0) or the given max-value of 16383,75 (this is 65535/4).

The process of querying this parameter is also strictly standardized. This standard definition is thus not “technologically neutral”. It is exactly this “strictness” that is vital for it’s authority use case. As a commercial “side effect”, there are numerous commercial OBD-Dongles available with related apps that query and display exactly these legally mandated identifiers.

The “name” in this standardization is here the cross-OEM-standardized (!) PID, the value 12.

#### 4.2.1.2 Current example: e-Call

The e-Call-Legislation is another example of a strict digital standard by name. Because every PSAP (Puplic Safety Answering Point)  has to be able to communicate digitally with any new type approved vehicle, all the communication standards and all information communicated had to be standardized down to the last detail[73].

The referenced standard for the details, e.g. for the information encoded in the transmitted Minimum Set of Data (MSD) is EN 15722:2011.

#### 4.2.1.3 Planned example: CITS

Within the CITS-Initiative, it is planned that future vehicles exchange messages about their positions, headings and traffic situations amongst themselves and stationary ITS-stations to enhance traffic safety[74].

[72] https://en.wikipedia.org/wiki/OBD-II_PIDs

[73] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015R0758&from=EN
[74] https://www.car-2-car.org/fileadmin/documents/Basic_System_Profile/Release_1.5.2/C2CCC_RS_2037_Profile.pdf

To ensure the cross-OEM and cross-OEM interoperability, also CITS is a strict digital standard by name, defining everything from communication protocols over security requirements down to message and data structure and quality.

Annotation: All the aforementioned standards supporting authority use cases are On-Board standards, they force the OEMs to implement them in the new vehicles.

### 4.2.2 Onboard Standards by Name

Aside from the standards by name supporting authority use cases, there is currently also a great trend towards strict standardization on-board driven by commercial interests. Standardization that leads to a common automotive software platform would create large benefits due to the laws of economy of scale and would also increase security. For details, see e.g. the McKinsey report[75].

The next paragraphs will briefly present prominent examples.

#### 4.2.2.1 De-facto Standard Android Automotive

One example of a standard by name introduced already and with increased implementation figures for future car is Android Automotive. It's background and details were already discussed in a previous chapter.

That's why in this paragraph the focus is on the benefit of a standardization by name achieved by such a standard in contrast to access standards.

[75] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform

### Show toasts

Your app can display a toast using `CarToast` as shown in this snippet:

Kotlin Java

```
CarToast.makeText(carContext, "Hello!", CarToast.LENGTH_SHORT).show()
```

**Figure 18: How to display a Text to a driver on any car running Android Automotive[76]**

The above code snippet is an example of code that allows the developer of an Application to display a short text (in this case „Hello!“) to the driver on any car of every OEM that supports this standard. There is no OEM specific way to offer a similar functionality, the snippet above is the one and only option to carry out the task of displaying a brief message to the driver.

### 4.2.2.2 Common Vehicle Interface Initiative (Covesa/Genivi & W3C)

Covesa – short for Connected Vehicle Systems Alliance – is an open technology alliance, founded by Bosch and BMW/Mini to support collaboration and standardization for connected and software defined vehicles. A prominent example for a proposed standardization is VSS, the Vehicle Signal Specification[77].

In general, VSS is an attempt to standardize data names/data identifiers across implementing partners to support the creation of a digital ecosystem via this strict standardization.

[76] **https://developer.android.com/training/cars/apps#display-notifications**

[77] https://github.com/COVESA/vehicle_signal_specification

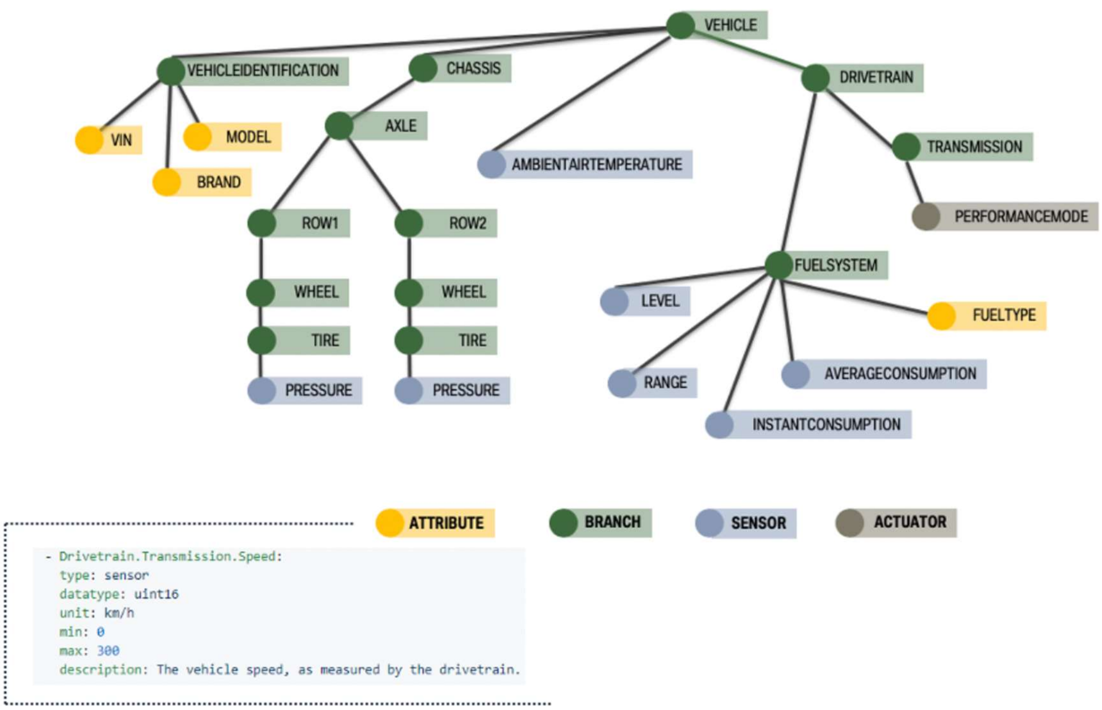



**Figure 19: Example for VSS[78]**

The idea is to arrange the "digital elements" in the car (Sensors reading data points from the car (retrieving e.g. a seat's position) , Actuators executing functions to alter data in the car, e.g. adjusting a seat's position) in a tree-like format and structure.

Like in the example of emission OBD-PIDs, also here the datatype, unit, Minimum and Maximum value are defined as another example of a strict standardization.

### 4.2.3 Offboard Standards by Name

To clarify the wording "Offboard" in this section, it should be noted that e.g. the authority standards like Ecall or CITS mentioned above are also Onboard as well as Offboard Standards. They require the OEM to implement something "in the car" (that's why they can be called on-board standards) but they are also used to communicate with other cars or backends like the PSAPs, so in this sense they are also offboard standards. A "pure" offboard standard would be a standardization by name only on the backend systems of an OEM, e.g. a stricter version of the ISO 20078 standard with a minimum set of data and functions to be supported across all OEMs.

[78] https://covesa.github.io/vehicle_signal_specification/introduction/overview/

### 4.2.3.1 Sensoris

Sensoris – short for Sensor Interface Specification[79] is another initiative to standardize data and data exchange information for sensors inside the vehicle as well as between vehicles and backends/clouds, according to its vision:

> SENSORIS –SENSOR Interface Specification – is an open group of actors from the global vehicle industry, map and data providers, sensors manufacturers and telecom operators who joined forces under the form of this innovation platform. Driven by the common vision, the actors believe that defining and appropriate interface for exchanging information between the in-vehicle sensors and a dedicated cloud as well as between clouds will inevitably:

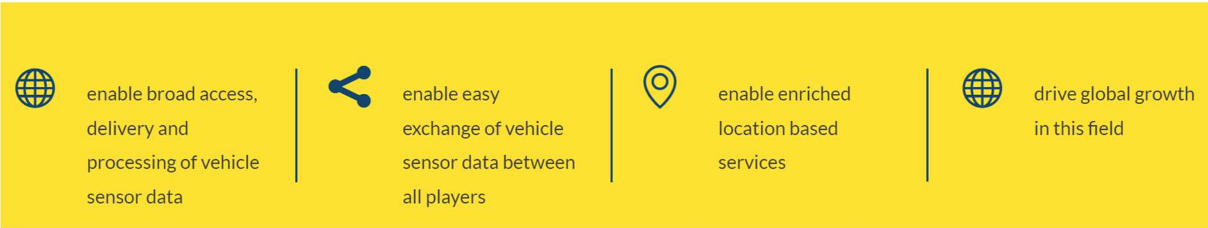



**Figure 20: The vision of the Sensoris-Group[80].**

The current public specification is v1.3.1 and can be found on[81].

The proposed data points and formats cover mainly aspects needed for safe and secure assisted and automated driving, so a certain overlap with data points needed for CITS is natural.

Like in CITS, Sensoris proposes a standard for vehicle orientation and speed:

Vehicle speed.

| Field | Type | Description |
|---|---|---|
| envelope | sensoris.protobuf.types.base.EventEnvelope | Event envelope. |
| value_and_accuracy | sensoris.protobuf.types.spatial.XyzVectorAndAccuracy<br>Unit Meter per Second Resolution 0.1 Range [0, ) | Value and accuracy. |

| Supported event relations |
|---|
| 1 sensoris.protobuf.categories.trafficmaneuver.Maneuver TRIGGERED_BY 1..* sensoris.protobuf.categories.localization.VehicleSpeed |

**Figure 21: Example for the definition of vehicle speed in Sensoris. (V1.3.1)**

[79] https://sensoris.org/
[80] https://sensoris.org/vision/
[81] https://sensoris.org/presentations/

However, in terms of scope, Sensoris goes way beyond CITS. Cars can describe what their current status is like in CITS with position , heading, acceleration and rotation, but also describe what their sensors perceive from the environment, e.g. Traffic light detected with signal Yellow at position x,y,z.

Sharing the perceptions of the environment with the clouds of e.g. Map providers will enable the Map providers to update their electronic maps more frequently for more accuracy and other cars can then benefit from the insights made by previous cars (Hazard detected at position X,y,z.).

A potential (not mentioned on the website) usage scenario would also be vehicle-2-vehicle exchange of perceived environment information in a sort of CITS 2.0 approach.

Cars could assure each other about their mutual detection and a car in front of another car could inform the following car about possible objects ahead already before the front car starts breaking.

## 4.3 Summary Standardization

Standardization of data, functions and processes around the development and testing of software based on these standards is the key to create successful digital ecosystems.

While access standards like SOVD or UDS serve their purpose in deep diagnostics of rather small and specific groups, large scale ecosystems can only be created via strict standardizations by name.

In terms of standardization power inside the automotive industry, there is a natural ranking:

- The highest power to standardize has the legislator (e.g. the EU), which can prescribe digital standards like eCall or in the past eOBD for all vehicles in Europe.
- Below this, commercial players can try to create their own ecosystem. Here the silicon valley players are in a privileged position currently, especially Google with their Android system.
- Individual OEMs seem to lack the power to establish a strict standardization “just for their own fleet” because the size is just not big enough to attract enough developers in comparison with the ecosystems that Google Android Auto/Automotive or the advertised new Apple CarPlay present[82].

So if the European Commission wants to stick to the vision of a Digital Single Market, it will be very likely up to them to create this market via a standardization mandatory at least for all cars in Europe. Should that not happen, the laws of the economy of scale will predict a win for the silicon valley giant’s standardizations especially in the lower levels of car integration, so levels 0..2.

[82] https://www.bloomberg.com/news/newsletters/2022-06-12/apple-s-aapl-ios-16-carplay-is-precursor-to-apple-car-wwdc-2022-recap-l4bczhc6


**Figure 22: preview of new Apple CarPlay[83]**

As a final and recent example for the push of Silicon valley companies into Level 1 and 2 see the above preview of the new Apple CarPlay that will require more integration into the car than the classic version and thus be more like Android Automotive (The OS) than Android Auto (The mirrored solution).

[83] https://www.cnet.com/roadshow/pictures/next-generation-apple-carplay-preview-wwdc-2022/

## 5 Neutral Servers & Marketplaces

For years, various consultants, experts and writers have declared that "Data is the new Oil", see e.g. forbes.[84]

That's why up to now terms like Data Lake, Data Space or Data Marketplace have attracted many investors due to the belief that the more data is available within a given concept/group/company, the higher the payoff for a potential invest should be. Very often this was supported by the wrong impression that Google would be a company that collects and sells data. This is completely wrong: Google offers data based (!) advertising services integrated in seemingly free data based (!) services like search or navigation. Google is not (!) a Data Marketplace, it doesn't sell data to private or business customers.

This chapter will present some of the most prominent Neutral servers and Marketplaces, describe their current and potential future relevance and highlight some vital obstacles that prevent greater influence as of today.

### 5.1 Commercial impact/relevance, Negotiation power

Prominent examples for Neutral servers/Data Marketplaces are e.g. in random order:

- Caruso Dataplace (https://www.caruso-dataplace.com/)
- Otonomo (https://otonomo.io/)
- High Mobility (https://de.high-mobility.com/)
- Smart Car (https://smartcar.com/)

The company size in terms of employees is rather small compared to other players like OEMs, Tier1s or large Parts trading companies.

According to the website, Smartcar has around 30 employees[85], Otonomo as a bigger player announces 135 employees as of end 2021[86].

All companies listed above are quite young (SmartCar was founded in 2014, Otonomo was founded in 2015, Caruso 2017, Start of Data Platform for High Mobility in 2019- though the company itself was founded in 2013).

The current financial situation is only reported publicly by Otonomo.

[84] https://www.forbes.com/sites/forbestechcouncil/2019/11/15/data-is-the-new-oil-and-thats-a-good-thing/?sh=b6b7fc473045
[85] https://smartcar.com/about/
[86] https://investors.otonomo.io/static-files/68ca8707-61ad-468a-a23f-6b5954576b20


**Figure 23: Investor information from Otonomo for 2021[87].**

According to these numbers, with a 2021 revenue of 1.7 Million USD against an operating loss of 16.2 Million USD Otonomo is still an investment. The other companies are not publishing financial results publicly, but are also looking for and finding investors. This year Smartcar raised 24 Million USD[88], High Mobility sold share to the strategic investor DAT in 2020[89], activities that would be unnecessary if the revenues made with these Marketplaces would be higher.

### Negotiation power

The figures presented above should not be seen as indicators fur future commercial success of single companies. Instead, they are shown to give a rough indication why the group of players from this sector has up to now failed to reach in negotiations with the far bigger OEMS:

- common, albeit not standardized set of data points and functions
- a minimum quality of data in terms of latency or sampling frequency
- a common set of pricing models (e.g. flat rate or per function call) to choose from, although with OEM specific pricing.

This rather weak position makes it up to now extremely difficult for Neutral Servers and their customers to develop multi brand solutions with a reliable business case. The details are explained in the following sections.

87 https://investors.otonomo.io/static-files/68ca8707-61ad-468a-a23f-6b5954576b20
88 https://investors.otonomo.io/static-files/68ca8707-61ad-468a-a23f-6b5954576b20
89 https://www.dat.de/news/dat-beteiligt-sich-am-telematik-start-up-high-mobility/

## 5.2 Technical abilities

The technical abilities and suggested applications domains are very similar, the same holds true for the overall technical approach.

These platforms try to standardize the usage of the data and functions offered by the different OEM backends/ExVes or obtained via the use of OBD dongles so that developers of applications can rely on a common standard.


**Figure 24: The vision of Caruso to create one API/standard for all OEMs**[90]

The general idea of the creation of a cross-OEM standard as the foundation of a thriving Digital ecosystem is fully in line with the insights and remarks about the role of standardization described in the previous chapter. However, several limits hinder this idea as of now.

## 5.3 Technical limits

By design, as a server based solution that is based on the server based solution ExVe of the OEMs each Data Marketplace-solution shares some systematic issues.

### 5.3.1 Lack of Real time ability

Neutral Servers „talk“ to the car via the backend/ExVe of the respective OEM. This results in high latencies and a far lower level of availability in comparison to on-board standards like e.g. Android Automotive.

[90] https://www.caruso-dataplace.com/developer-zone/

### 5.3.2 Lack of User Interface

The Neutral Server/Marketplace-solutions deal only with the data points (e.g. mileage) or sometimes actions (flash lights) that the OEM decide to offer via their backend. For the access to the driver, only smartphone access is available, which is potentially distractive whilst driving.

### 5.3.3 Lack of common set of data/functions amongst underlying OEM backends that can be standardized

While the main business idea of all of these companies is the creation of a unified standard/API, this has nevertheless the prerequisite that every OEM essentially offers the same data points/functions, albeit with potentially different namings, maybe different data types etc.

Something, that is missing completely, can't be standardized by any company.

Unfortunately, the common data set amongst even the big OEMs is strikingly small.

**Coverage of IAM data points**

| Data point name | Category | Supported brands* |
|---|---|---|
| Odometer Value | Vehicle Position, Movement & Surroundings | 100% |
| Fuel Tank Level in Liters OR Percent | Vehicle Powertrain Resources | 76% |
| GPS Position & Time at Current Value | Vehicle Position, Movement & Surroundings | 71% |
| Remaining Electric Range/Distance | Vehicle Powertrain Hardware | 71% |
| High voltage Battery Charging Status | Vehicle Powertrain Hardware | 71% |
| Speed of Vehicle at Current Value | Vehicle Position, Movement & Surroundings | 53% |
| Buckles Status | Vehicle Non-Powertrain Hardware | 53% |
| Power Steering Warning Lamp | Vehicle Non-Powertrain Hardware | 53% |
| Coolant Temperature High Warning | Vehicle Powertrain Resources | 53% |
| Tyre pressure Warning Indicator Status | Vehicle Non-Powertrain Hardware | 53% |
| Windshield Wiper Fluid Level | Vehicle Non-Powertrain Hardware | 53% |
| Acceleration of Vehicle | Vehicle Position, Movement & Surroundings | 47% |
| Brake Pad Low Warning | Vehicle Non-Powertrain Hardware | 47% |
| Airbag Warning Lamp | Vehicle Non-Powertrain Hardware | 47% |
| Brake Fluid Level Low Warning | Vehicle Non-Powertrain Hardware | 47% |
| Oil Pressure Warning Lamp | Vehicle Powertrain Resources | 47% |
| Window Status | Vehicle Non-Powertrain Hardware | 35% |
| Door Status | Vehicle Non-Powertrain Hardware | 35% |
| Trunk Status | Vehicle Non-Powertrain Hardware | 35% |
| Convertible Roof Status | Vehicle Non-Powertrain Hardware | 29% |
| Diagnostic Trouble Codes | Vehicle Health & Maintenance | 17% |

| Data point name | Category | Supported brands* |
|---|---|---|
| Distance to next Service | Vehicle Health & Maintenance | 17% |
| Time to next Service | Vehicle Health & Maintenance | 17% |
| Door Lock Status | Vehicle Non-Powertrain Hardware | 17% |
| Quiescent Current/ value | Vehicle Powertrain Hardware | 17% |
| Full Lock Status | Vehicle Non-Powertrain Hardware | 17% |
| Time until next Required Oil Change | Vehicle Powertrain Resources | 17% |
| Sunroof Status | Vehicle Non-Powertrain Hardware | 17% |
| Service Indicator | Vehicle Health & Maintenance | 12% |
| Low Oil Indicator | Vehicle Powertrain Resources | 12% |
| Malfunction Indicator All Systems & ECUs Status | Vehicle Health & Maintenance | 12% |
| Remaining Brake Fluid Life | Vehicle Non-Powertrain Hardware | 12% |
| Distance until Brakes Service | Vehicle Non-Powertrain Hardware | 12% |
| Due Date for Brake Fluid Change | Vehicle Non-Powertrain Hardware | 12% |
| Tilting Status of Sunroof | Vehicle Non-Powertrain Hardware | 12% |
| Vehicle High Voltage Battery State of Charge | Vehicle Powertrain Hardware | 6% |
| Total Remaining Range | Vehicle Powertrain Resources | 6% |
| Diesel Additive Range | Vehicle Powertrain Resources | 6% |
| Diesel Additive Status (Low Indicator) | Vehicle Powertrain Resources | 6% |
| Distance until Engine Oil Service | Vehicle Powertrain Resources | 6% |
| Oil Temperature High Warning | Vehicle Powertrain Resources | 0% |
| Battery Lamp Status | Vehicle Powertrain Hardware | 0% |

**Based on CLEPA / ACEA 42 data items list*
*Maximum 100% for 17 brands: Audi, BMW group (2), Ford, Hyundai, KIA, Mercedes-Benz, Porsche, Stellantis group (8), VW - representing 74% of the German passenger car park on 1st January 2021 (KBA).*

CARUSO dataplace

4 CARUSO-DATAPLACE.COM
#carusodataplace

**Figure 25: Coverage of data points amongst 17 brands. (Source: Caruso status report. 19 May 2022)**

Following the principles of digital ecosystems and the vital role of standardization this is undoubtedly one of the biggest obstacles in the way of commercial success.

Neutral Servers can' standardize data points or functions across OEMs if the respective data points or functions are simply missing for some OEMs or some car models from OEMs.

The same holds true for data quality attributes like latency or frequency. Neutral servers can neither increase the speed of data transmission and lower the latency nor can they "invent" additional data points for an OEM that samples data at a lower rate (e.g. once per Minute) in comparison with another OEM that samples the same data point every second.

Technically, the “slowest” OEM-ExVe with the highest latency and the lowest sampling rate determines the data quality that a multibrand use case can safely assume.

## 5.4 Commercial Limits

The commercial limits apply to to the relation to the OEMs as well as to the relations with potential customers.

### 5.4.1 Lack of standardization/Market power

With the size and market power of each individual Marketplace comparably small in relation to the size of the OEMs, the “Balance of negotiation power” is tilted heavily in favour of the OEMs. The Marketplaces have to take, what’s offered as data points and functions by the OEM, leading to the highly fragmented coverage in data points just described. This rather frustrating experience is unlikely to change in the near future due to the gross differences in size and market power.
When Google, Apple or Amazon negotiate standards with OEMs, the relation reverses. These companies tell the OEMs exactly, what devices onboard have to implement what features (Display Size, Microphone quality), and what data points and functions have to be made available to e.g. Google’s Android Automotive via the Vehicle Hardware Abstraction Layer.

### 5.4.2 Lack of pricing standardization

Aside from the technical fragmentation in the offerings of different OEMs in terms of common data points and functions, also the pricing and pricing model varies between OEMs.

Price overview

The following prices apply to the use of Connected RMI Services:

| | Price |
|---|---|
| **One-off setup fee for every service setup** | |
| Service partner setup | |
| Accident assistance setup | 50,00€* |
| Roadside assistance setup | |
| **Service partner services** | |
| Per service ticket – Automatic Service Call [1)] | |
| Per service ticket – Battery Guard Call [1)] | |
| Per service ticket – Accident Action Call [1)] | 3,00€* |
| Per remote key read in KaSIO [2)] | |
| Per VIN query in the Dealer Cockpit [2)] | |
| **Accident and roadside assistance services** | |
| Per accident assistance event [3)] | 4,00€* |
| Per roadside assistance event [3)] | |

**Figure 26: Pricing model for BMW services: Setup fee plus event based costing[91].**

The above example includes a setup fee for these kind of services. In return, providers of services can enter a location and a phone number. In case an event is triggered (e.g. service need), the

BMW e.g. offers individual retrieval of data points/events against fixed prices (0,29 Euros, 0,09 Euros) , caps this with a 5 Euro fee per Month and has an elaborated volume based discount on top it[92].

Mercedes had during a POC conducted by KG on behalf of the EC in 2018/2019 a per Call based model for remote diagnostics:

Small: 1 Call for 5.99 Euro

Medium: 10 Calls for 54,99 Euros.

Large: 100 Calls for 544,99 Euros.

However, this pricing was also at that time subject to B2B negotations around a specific use case.

To sum it up: as can be expected in a free economy, prices and pricing models vary across OEMs today.

This makes it inherently difficult for Data Marketplace companies and their potential customers, the multibrand developers alike to come up with a reliable business case.

91 https://bmw-cardata.bmwgroup.com/thirdparty/public/repair-and-maintenance/pricing

92 https://bmw-cardata.bmwgroup.com/thirdparty/public/car-data/pricing

High Mobility e.g. tries to at least come up with a standard pricing model (data per Month per use case) and a "starting price" in the range of 0.50 Euros to 5.35 Euros per Car and per Month[93].

However, also here discounts of OEMs can apply (see e.g. BMW) and make business predictions complex. An car insurance company e.g. can not easily predict, how many of the interested customers of their future App will be BMW owners and thus eligible for refunds.

## 5.5 Trust Limits/Competition concerns

Understandably OEMs have an inherent conflict of interest when offering or selling data to "unknown" customers. That's why some initial and public pricings are higher than the ones negotiated between two known parties (OEM and Provider) in a B2B agreement covering also other legal elements.

Not only are the prices lower, also the extent of data and functionality offered can be significantly higher when the OEM gets to know his partner in a B2B negotiation and can determine, if the onboarding of the partner is a benefit to or a detriment for his own service offerings around the vehicle.

Due to NDAs, the sensitive information about extent of data/functions and prices unfortunately can't be referenced in a public report.

## 5.6 Lack of technical Standardizations in the source ExVe offerings

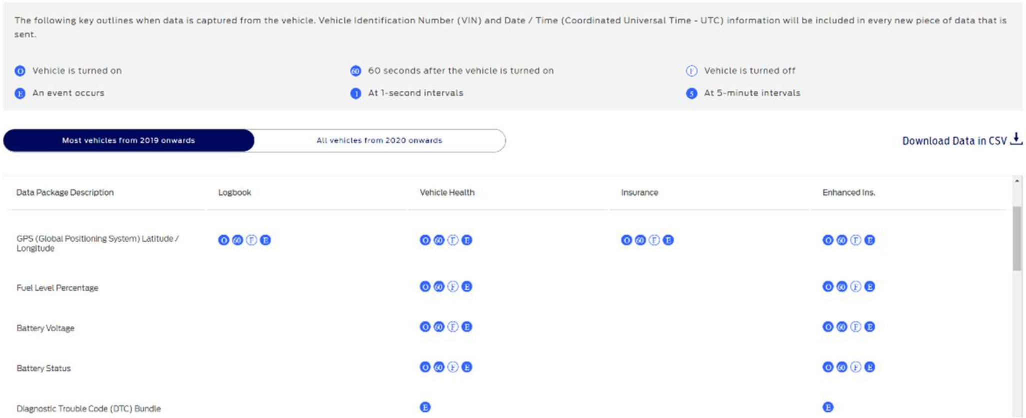
Standardisation across OEM and sometimes even within an OEM is in a very early stage. Since the suite of standards 20088,20078 and 20080 is rather vague, basically each and every chosen technology and each and every set of exposed data & functions can be declared „ExVe-Standard compliant".

So some OEMs decided to establish a request/response-scheme for data access („Request mileage from vehicle x"…Response is Mileage from Vehicle X), some added event based delivery (e.g. send data when trip is finished, when theft detected), some timed based delivery (read data every 3 days, every minute, every 15 seconds etc.). Few can trigger rather simple actions most users are familiar from their OEM remote app like charging (Electric vehicles) , sending navigation targets from remote, honking horns or flashing lights to detect a parked car.

This makes it burdensome for every marketplace provider to come up with one unified set of data and functions in one quality (frequency, latency) and renders multibrand development for predictable prices close to impossible.
To give examples:

93 https://www.high-mobility.com/pricing

Ford just pushes data from the car in different packages:



The following key outlines when data is captured from the vehicle. Vehicle Identification Number (VIN) and Date / Time (Coordinated Universal Time - UTC) information will be included in every new piece of data that is sent.

O Vehicle is turned on
60 60 seconds after the vehicle is turned on
F Vehicle is turned off
E An event occurs
1 At 1-second intervals
5 At 5-minute intervals

Most vehicles from 2019 onwards | All vehicles from 2020 onwards

Download Data in CSV

| Data Package Description | Logbook | Vehicle Health | Insurance | Enhanced Ins. |
|---|---|---|---|---|
| GPS (Global Positioning System) Latitude / Longitude | O 60 F E | O 60 F E | O 60 F E | O 60 F E |
| Fuel Level Percentage | | O 60 F E | | O 60 F E |
| Battery Voltage | | O 60 F E | | O 60 F E |
| Battery Status | | O 60 F E | | O 60 F E |
| Diagnostic Trouble Code (DTC) Bundle | | E | | E |

**Figure 27: Pushed data by Ford, timings vary between data points.**

Audi however established a typical request/response scheme, albeit for just three use cases[94].

Data packages are clustered in Use cases.

1.) First Notification of loss:

Type: Push in case of an event.

2.) Pay as you drive: Mileage, parking time of the vehicle.
Frequency: Can be pulled every three days

3.) Vehicle status, surprisingly named as an Insurance product, delivers the status of doors, boot, bonnet and sliding roof. This might be useful in case of theft – did the driver left a door open?

Frequency: also to be pulled max every 3 days. It is hard to see how this helps theft analytics in between these three day sample intervals.

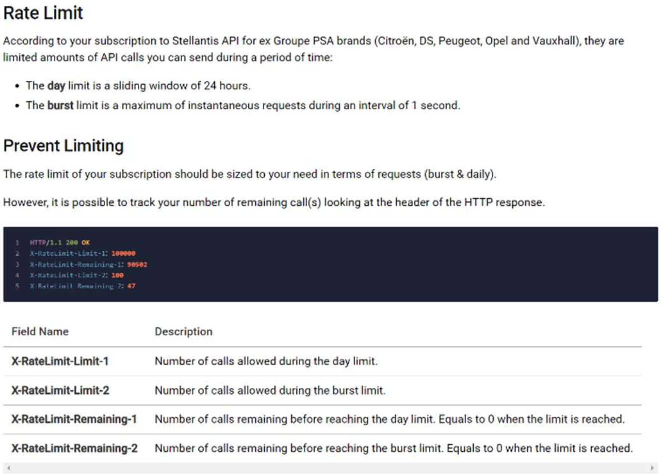
Stellantis uses even different technologies for different vehicles[95].

For small electric vehicles completely different technologies and APIs are used than for the rest of the fleet.

When looking at the „normal API“, different rate limits in terms of latency and frequency can be negotiated with Stellantis[96].

---

[94] https://www.audi.de/de/brand/de/extended-vehicle-data-en/data-products/vehicle-status.html
[95] https://developer.groupe-psa.io/mobile-sdk/sdk-features/ami&rocks-e/#article
[96] https://developer.groupe-psa.io/webapi/b2c/overview/errors/#article



### Rate Limit

According to your subscription to Stellantis API for ex Groupe PSA brands (Citroën, DS, Peugeot, Opel and Vauxhall), they are limited amounts of API calls you can send during a period of time:

- The **day** limit is a sliding window of 24 hours.
- The **burst** limit is a maximum of instantaneous requests during an interval of 1 second.

### Prevent Limiting

The rate limit of your subscription should be sized to your need in terms of requests (burst & daily).

However, it is possible to track your number of remaining call(s) looking at the header of the HTTP response.

```
HTTP/1.1 200 OK
X-RateLimit-Limit-1: 100000
X-RateLimit-Remaining-1: 90502
X-RateLimit-Limit-2: 100
X-RateLimit-Remaining-2: 47
```

| Field Name | Description |
| --- | --- |
| **X-RateLimit-Limit-1** | Number of calls allowed during the day limit. |
| **X-RateLimit-Limit-2** | Number of calls allowed during the burst limit. |
| **X-RateLimit-Remaining-1** | Number of calls remaining before reaching the day limit. Equals to 0 when the limit is reached. |
| **X-RateLimit-Remaining-2** | Number of calls remaining before reaching the burst limit. Equals to 0 when the limit is reached. |

Figure 28: Rate limits per day and per 1 second "bursts" as part of a subscription at PSA.

As a next issue, some APIs are for fleet operators B2B, others are for B2C-customers.


Abbildung 1: Snapshot from[97], showing different APIs for different user groups.

Mercedes clusters data also in use cases but sometimes even within a use case, data can be send in a given quality[98].

[97] https://developer.groupe-psa.io/

[98] https://developer.mercedes-benz.com/products/connect_your_business/details

| | | | | |
|---|---|---|---|---|
| 1018 Position data 15 | Position.LastKnown.Heading | degrees | Position-heading of vehicle in approx. 15s intervals: heading of vehicle | onInterval |
| | Position.LastKnown.Latitude | degrees | Position-lat of vehicle in approx. 15s intervals: latitude in WGS84 coordinates | onInterval |
| | Position.LastKnown.Longitude | degrees | Position-long of vehicle in approx. 15s intervals: longitude in WGS84 coordinates | onInterval |
| 1019 Position data 30 | Position.LastKnown.Heading | degrees | Position-heading of vehicle in approx. 30s intervals: heading of vehicle | onInterval |
| | Position.LastKnown.Latitude | degrees | Position-lat of vehicle in approx. 30s intervals: latitude in WGS84 coordinates | onInterval |
| | Position.LastKnown.Longitude | degrees | Position-long of vehicle in approx. 30s intervals: longitude in WGS84 coordinates | onInterval |
| 1005 Position data 120 | Position.LastKnown.Heading | degrees | Position-heading of vehicle in approx. 120s intervals: heading of vehicle | onInterval |
| | Position.LastKnown.Latitude | degrees | Position-lat of vehicle in approx. 120s intervals: latitude in WGS84 coordinates | onInterval |
| | Position.LastKnown.Longitude | degrees | Position-long of vehicle in approx. 120s intervals: longitude in WGS84 coordinates | onInterval |

Figure 29: Within a fleet management use cases, positions from cars can be sampled every 15 secs, 30 or 120 secs.

### 5.6.1 Summary standardization on ExVes

In the future due to either market pressure or legal requirements a further standardization is likely. Up to now, marketplace operators report it in interviews as a real challenge to at least to a certain degree try to unify the disparate technologies.
By sheer logic, they can only agree on the least common denominator. For data frequency, one possible action taken by at least one supplier is data filling or interpolation. If one OEM sends data every 15 seconds and the other only every 60 secondds, then the 60 seconds value can be send three times to give the illusion of a higher frequency. However, if a use case like navigation relies on positions reported at least every 15 seconds, then there is no way how a 60-second value could be sufficient in dense traffic, filled in or not.

## 5.7 Summary Neutral Servers and Marketplaces

The commercial idea to just “sell access” to a variety of data points and functions that is common in the ExVe as well as in the subsequent Data Marketplace is in our assessment oversimplified and thus runs into a list of problems, starting from legal considerations (GDPR, competition issues) over pricing considerations (“What might this data point be worth for a potential competitor?”) down to the technical issues (Lack of coverage in data points and functions).

**The far superior idea for the author is the familiar concept of app development for platforms as it is used in the smartphone sector.**

First the partners clarify legal issues like NDAs, IP, GDPR etc., then a payment model is agreed (typically a fixed percentage of every revenue made via the App on the platform goes to the platform provider) while the consumption of data points and the access to functions by the app is always free within the technical limits (e.g. bandwidth). This eliminates any issue with predicting data costs for the Service provider.

On a positive side, the technological experience of Marketplace operators can render them ideal technical enablers after (!) especially bigger customers have sorted out the commercial and legal details with various OEMs and it is now "just" about enabling the data or function access and keeping it available within predefined SLAs.

# 6 Mobility Data Spaces & Clouds

In recent years, throughout Europe, several Initiatives have started aimed at creating systems called "Data Space". This chapter will discuss briefly their history and present their current status and challenges.

## 6.1 History of Data Spaces, related concepts

With more and more data becoming electronically available in Computer systems, more and more attempts have been developed and practically applied to integrate these data for better analytics and/or better service provision throughout at least the last 30 years (Of course scientific research on the topic started already years ahead.)

### 6.1.1 Data Warehouse, Data Marts – structured within an Enterprise

In the 90s of the last century, Bill Inmon (1992-Building the Data Warehouse) and Ralph Kimball (1996 – The Data Warehouse Toolkit) published two of the most relevant books to integrate the data within (!) an enterprise.

For the level of application, roughly speaking the term "Data Mart" referred to an integration of all data within a department, while "Data Warehouse" comprised an integration of all Data within an enterprise.

It should be kept in mind that according to Inmon a "Data Warehouse" is a separate data storage for reporting and analytics that just reads data from the source systems (called transaction systems). So no data is fed back from the warehouse in the various source systems to enhance their interoperability.

### 6.1.2 Data Lake – unstructured for "full information preservation"

Though intuitively a "Lake" is bigger than a "Warehouse", a Data Lake need not necessarily be bigger than a Data Warehouse. The difference is more in the way, the data is structured within the Data Warehouse and the Data Lake.

The Data Warehouse is in comparison with the Data Lake the far more structured approach. A harmonized enterprise data model is developed, selected data from different sources is transferred into the new system. However, some Data Warehouse projects suffered from a list of issues:

- The effort to create a common data model and to transfer the selected data was perceived too high.
- In the end, many users felt that the Warehouse only answered questions that they already knew the answer to within their old system environment.
- Some discovered that in then end still vital information was missing from the source systems, so that the enterprise data model had to be enhanced again and the transfer programs had to be adopted accordingly.

To overcome these issues with lost information or a high integration effort, the Data Lake concept, put forward by James Dixon starting in 2010, proposed to store data from various sources in its raw format in a system called Data Lake, so to stay away from first ensuring interoperability and standardizing data types and formats between different sources. This ensures that obviously "No Information is lost", but makes analytics across these sources far more challenging than in a Data Warehouse. Concepts like Hadoop[99] have been developed for these "Big Data"-purposes.

Authors opinion: When the highly manual effort of creating an enterprise standard for data was found too high "just for analytics", too much hope was placed on the promised features of AI and Big Data Analytics to find the Gold Nuggets in "Data Mining". That's not to say AI, Big Data Analytics or Data Mining don't have their merits, but they can't fully replace the effort to manually ensure standardization.

### 6.1.3 Varying interpretations of "Data Lake" and "Data Warehouse"

In some enterprise projects and in the perception of the wider public the definitions of "Data Lake" and "Data Warehouse" was watered down and applied too generously. While Inmon requested one enterprise data model and thus one Data Warehouse per enterprise, especially large companies created several big data bases for specific purposes as multiple "Data Warehouses", though normally these would qualify "just as Data Marts. Other projects created Warehouses that fed back data into transaction systems, thus enhancing interoperability but violating their read only data from the original definition.

Last not least, maybe due to the appealing new name, big projects for enhancing interoperability also by means of standardization where called "Data Lakes", though especially this upfront effort was one of the reasons Data Lakes were built initially out of dumping raw data without any standardization in one data storage.

This rather sloppy use of terms within several enterprises across industries makes it difficult to exactly determine the nature of a project.

But all in all, either one of the terms denote efforts within (!) an enterprise to leverage its digital potential in terms of either enhanced interoperability or enhanced insights via improve analytics.

### 6.1.4 Non-technical reasons for failure of related concepts

Research and industry have made great progress in developing technologies, algorithms, concepts and processes to increase insights into and interoperability between different IT-systems.

The reasons so many Data Warehouse, Data Lakes and Big Data Projects have not been as commercially successful as intended can thus be found predominantly on other dimensions:

a.) Legal

b.) Commercial/Organizational

[99] https://en.wikipedia.org/wiki/Apache_Hadoop

The latter one (Commercial), while often publicly overshadowed by the first dimension (Legal) is in the author's perception the more frequent reason for failure.

#### 6.1.4.1 Legal issues for failure

There are indeed for good reason laws in place that prevent unauthorized duplication or transfer of data, e.g. the GDPR or special laws applicable to the handling of health data. Other restrictions on data sharing can be imposed by NDAs or contractual agreements that limit the scope of authorized viewers on certain data. Especially across Enterprises, sometimes within, these Legal issues prevent enhanced interoperability or enhanced analytics.

#### 6.1.4.2 Commercial/Organizational reasons for failure

Very often the aforementioned legal concerns are grossly exaggerated by either internal managers of departments or external suppliers to prevent enhanced insights or enhanced interoperability because their commercial interests might be negatively affected.

To name just a few examples from past projects:

- The creation of a really large Big Data project can have team sizes of 100+ employees and easily budgets in the magnitude of tenth of millions. The newly assigned manager will therefore have a big relevance within the company and Managers on the same hierarchy level might not be interested in another competitor for the next promotion, so they try to cut team size and budget of the new Big Data project even before the start.
- "Knowledge is power"[100] applies also within the company and despite public claims, some managers and employees are not willing to share their data and their knowledge even within the same company without any real personal compensation because they feel that either they empower potential competitors for a new position or they feel obsolete, once their formerly secret know how that rendered them indispensable was transferred.
- Suppliers of IT-systems are also very aware of "Knowledge is power". That's why very often vendors offer a variety of options to integrate data from other sources and systems into their own database, but very reluctant to offer data exports to other systems especially in standardized formats. If only within their system certain types of data can be processed and thus certain business processes can be carried out, they reach an exclusive position within an enterprise's IT-Landscape. Should the data and the interfaces to the data be standardized, then they would face a far stiffer competition and could excel only via better algorithms, better user interfaces or simply lesser costs than their competitors. This applies also to internally developed systems of internal suppliers.

The listed reasons forced Amazon Founder Jeff Bezos to come up with an API mandate for all systems and teams within his company.

[100] https://en.wikipedia.org/wiki/Scientia_potentia_est

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
4. It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols – doesn't matter.
5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
6. Anyone who doesn't do this will be fired.
7. Thank you; have a nice day!

Figure 27: Jeff Bezos' API Mandate[101]

This request for interoperable data points and functions in an API for each system enhanced the competitiveness of the enterprise amazon as a whole because each and every system and team had to face a tough competition from within or outside the company and could just not sit back relaxed on the data. This „legislation" for the digital ecosystem Amazon was „enforceable", because the author of the mandate owned the system and could fine non-compliants (See e.g. Mandate point 6).

## 6.2 Data Space Definition

According to Wikipedia, Data Spaces are:

„**Dataspaces** are an abstraction in data management that aim to overcome some of the problems encountered in data integration system. The aim is to reduce the effort required to set up a data integration system by relying on existing matching and mapping generation techniques, and to improve the system in "pay-as-you-go" fashion as it is used. Labor-intensive aspects of data integration are postponed until they are absolutely needed."[102]

With this definition, they pick up one idea from the Data Lake that basically states (in its original definition), that they data from different data sources can stay in it's formats and the analytics is basically left to Big Data algorithms and AI. In contrast, a Data Warehouse Approach according to Inmon would require that one target data model with exactly one set of data attributes and format is developed upfront.

[101] https://nordicapis.com/the-bezos-api-mandate-amazons-manifesto-for-externalization/
[102] https://en.wikipedia.org/wiki/Dataspaces

### 6.2.1 Scope of Data Space

An important difference between Data Lakes or Data Warehouses and the Concept of a Data Space is the scope. Lakes and Warehouses are usually set up within an enterprise to consolidate data and ease interoperability within this legal entity. The scope of a Data Space is bigger and can span across enterprises and countries.

### 6.2.2 Decentralized approach

Because data is such a sensible good, one central concept of a Data Space is that the data stays within the company that owns or produced it. The data is not replicated and transformed into one big database as it is the case for a Data Warehouse.

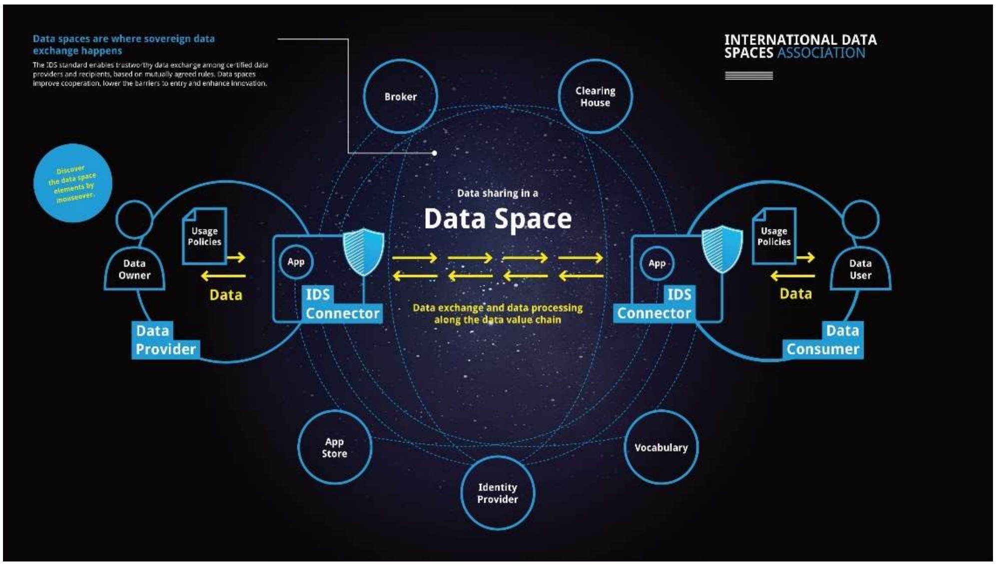



Figure 28: Overview International Data Space[103]

Instead a Data Space is essentially a big virtual database, as depicted in the above figure from the International Data Spaces Association. Because the data remains in local storage, the Data Space is essentially a technology and process framework that helps Data Consumers to find and consume useful data from Data Providers.

Typical steps, tools and toles involved in the data exchange are:

1.) The members of the data space agree on a sector specific „Vocabulary“ to ease the finding of useful data. Let’s assume, terms like „VIN“ and „Mileage“ would be part of that vocabulary.

2.) The Data Owners, identified and authorized by the Identity provider expose via a Connector tool the catalogue of data they would be willing to offer for consumption.

[103] https://github.com/International-Data-Spaces-Association/idsa/blob/main/images/IDSA-Infographic-Data-Sharing-in-a-Data-Space.jpg

3.) The Broker „talks“ to the different IDS connectors and thus builds up an inventory which data is available from what Data provider (E.g. OEMS A and B both offer VIN and Mileage)

4.) Authorized by an identity provider, Data Consumers (e.g. workshops) visit the broker, find relevant data at offering providers and consume them according to the usage policies attached to the data by the offering Data providers.

5.) The app store contains apps for data transformation from one format into another or for data analytics and can be used for the data exchange between Providers and consumers.

Similar elements can be found in the new project (first tenders for concepts are closed at the start of 2022 ) for a future Common European Mobility Data Space:

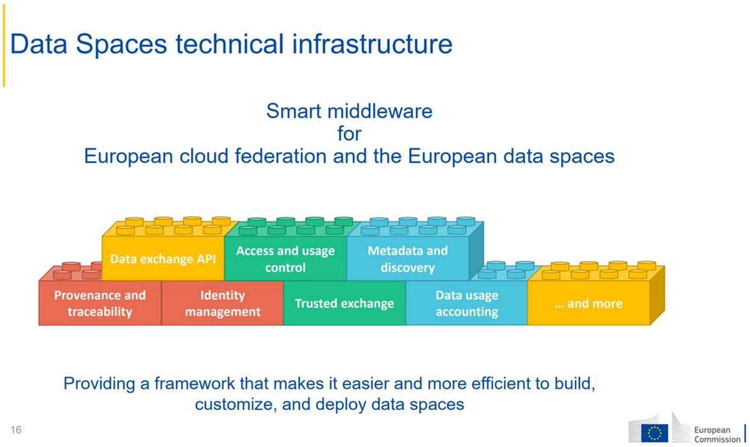



**Figure 29: Technical Infrastructure, Focus on Smart Middleware**[104]

It should be noted that Data Spaces, while being designed to operate across countries, are sector specific.

[104] https://digital-strategy.ec.europa.eu/en/events/workshop-common-european-mobility-data-space

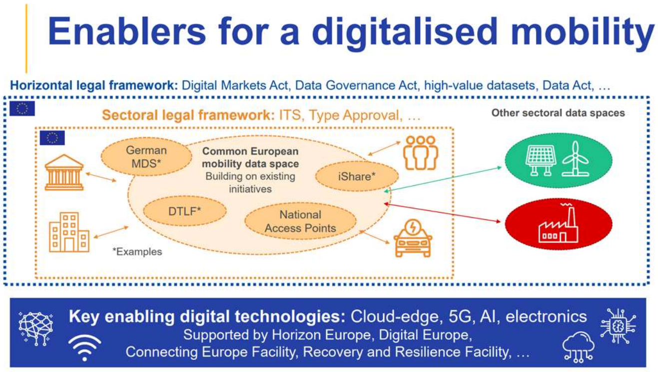



**Figure 30: The vision of the future Common European Mobility Data Space[105]**

This accounts e.g. for a sector specific vocabulary (A Health Data Space will use a different vocabulary than a Mobility Data Space or potentially sector specific rules on the usage and data protection. (Health data is likely to be on average more sensitive than Mobility Data).

## 6.3 Limits of decentralization: Who „owns“ the Data Space?

While the concept of the Data Space can be viewed as a distributed (Data stays at the Providers) and ad-hoc Data Warehouse (Data is only retrieved and used on demand) involving multiple Providers and consumers, the question remains who develops and operates against what compensation or with what business model the single middleware and takes responsibility in case of issues between Data Space participants. The role of the clearing house e.g. is limited to pure logging of transactions and computing the billable usage of data[106].

## 6.4 Status of Mobility Data Space Project (s)

The German Mobility Data Space[107] started in 2019 and is funded by the German Federal Ministry for Digital and Transport. Note that the hint on the website „Participating in the Mobility Data Space is free of charge until 2024.“ puts emphasis on the issue which entity operates against what compensation the Data Space once initial public funding is exceeded.

---

[105] https://digital-strategy.ec.europa.eu/en/events/workshop-common-european-mobility-data-space

[106] https://www.dataspaces.fraunhofer.de/en/software/clearing_house.html#:~:text=The%20Clearing%20House%20is%20a,transactions%20between%20participants%20are%20offset

[107] https://mobility-dataspace.eu/

From a technology viewpoint, the middleware is developed from its predecessors, the Industrial Data Space (starting in 2014, founded by the Federal Ministry of Education[108] and later relabelled as International Data Space in 2016[109].

Up to now, several Proof of Concepts had been conducted for the German mobility data space and it's predecessors, several use case have been identified[110].

Unfortunately, the detailed status of the use cases and to what extent and depth the use case use what concept or tool from the MDS is not revealed, so the current maturity of the tools and processes can't be assessed reliably.

The European project is to start in 2022, can therefore only be in a very early stage but shall take into account existing projects like the German MDS which should give the European project a head start.

## 6.5 Issues of the Data Space Concept

As a successor and an evolution of several data integration and system interoperability concepts like Data Warehouses and Data Lakes the Data Space concept surely features advanced technology and tackles various issues that e.g. Data Warehouse projects with their high upfront effort for developing an enterprise wide data model.

Notwithstanding these advantages, technology has its limits and legal and commercial questions remain.

### 6.5.1 Technological limits

Two technological limits can be found in the areas of trust/security and application domain.

#### 6.5.1.1 Trust/Security related limits

The idea, that full control over the usage of data by the Data Provider can be ensured by technology alone, seems a bit ambitious. Data has the advantage/disadvantage, that it can be cloned and copied at virtually no costs. Even with the latest security technology of certificates, watermarks etc the whole process chain of data usage has one weak link: The Human being.

Once displayed to a human operator on a website by a browser, ultimately just a photo with Optical character recognition can be used to extract all the needed data and to copy and distribute it. Other methods like automated browsing via successive HTML-calls by a tool and extracting data from the HMTL language might even prove easier and more effective.

[108] https://www.fraunhofer.de/content/dam/zv/de/Forschungsfelder/industrial-data-space/Industrial-Data-Space_whitepaper.pdf
[109] https://de.wikipedia.org/wiki/International_Data_Spaces
[110] https://mobility-dataspace.eu/use-cases

That's not to say that certain technologies don't have their merits and should not be applied, but they should be accompanied by legal measures (e.g. enforceable contracts) and high commercial fines for violations.

#### 6.5.1.2 Application domain limits

With data normally stored at their origins and exchanged only ad hoc between parties and servers it's very likely that real time applications will be out of scope for this concept.

### 6.5.2 Legal Issues

Since data is a highly sensitive good and technology has its limits in ensuring data protection, it is vitally important to establish a clear legal framework for any future Data Space. Is there e.g. one operator of an app store, multiple operators, who has the final say on updates to the vocabulary?

This "strict approach" would basically establish one "Platform or Data Space Owner" like Google for the Android Operating System or Amazon for the Amazon-Online-Shop who would have the power and mandate to fine or even dismiss Providers or Consumers from the Data Space.

A "Middle approach" would define a set of legal templates that must be used whenever two parties agree on data exchange. Violations must then be sorted out between the parties.

A "soft approach" would focus just on the technology and leave all commercial and legal details to the players that agree to terms.

### 6.5.3 Commercial issues

A very imported aspect that affects not only the Data Space, but also every other idea of buying/using data is the question of the "fair" value of a data point.

In short, the value depends not only on the data point and the data quality (e.g. data frequency), but commercially it depends on the intended use case and on the use case/service portfolio of the offering party.

A data point location with a data quality attribute frequency of once per Minute might be sufficient for the use case of roadside assistance but is insufficient for inner city live navigation.

A new OEM that wants to enter the market in Europe and has no service network might sell the data points far cheaper to workshops than an established OEM who wants to protect revenues from his own service network by raising the costs for potential competitors.

So the question of assigning "Fair" prices to data points is difficult for all stakeholders in a Data Space.

## 6.6 Summary Data Spaces

Data Spaces are an important evolution in the domain of Data Integration and System interoperability.

Their technical means and security technologies can ensure a best in class exchange of data between Providers and Consumers. Especially all the standardization efforts for protocols and vocabulary will prove useful for the setup of Digital business models within a Domain like e.g. Mobility.

Nevertheless, as a Digital Business model is needed, a likely successful sequence of steps would be:

a.) Use the discovery technology to find potential partners for a digital service

b.) Set up B2B contracts between the parties, agree on commercials.

c.) Use technology to exchange data along the defined legal guidelines for the service in question.

d.) Use – if applicable for e.g. a pay per usage model – technology like Clearing of the Data Space for reliable controlling of exchanged amounts.

## 7 Summary & Outlook

We are entering the era of the Software Defined Vehicle. This has introduced new players into the automotive market. This includes technology providers such as Google, Apple or Nvidia, but also new Vehicle Manufacturers like Tesla, Rivian or an increasing number of Chinese manufacturers, with a ground up, ‘digital first’ approach.

New highly standardised in-vehicle platforms, such as Android Automotive from Google, are being increasingly integrated in vehicles and will rapidly reach massive deployment. In some cases OEMs choose to license a full Google solution, including its AppStore and services (Maps, Voice Assistant etc.), while some OEMs opt for an open-source version of Android Automotive and choose 3rd party technology providers to provide the supporting services. In both scenarios these “on-board ecosystems” are highly standardized, reliable, real time enabled and strictly governed by Google. That is also the case for solutions provided by Apple (CarPlay).

By contrast, If we look at what OEMs in Europe are offering today for access to data and functions, the degree of standardisation is extremely low, in terms of the extent/ scope of data and functions being made available, the quality of that data and also the unharmonized formatting of data & functions. This is also true of the pricing and business models proposed. This makes it very difficult for service providers who wish to develop multi-brand services. It also hampers the ability of any company operating as a Neutral Server or Marketplace to build a highly standardized digital ecosystem with a reliable price model and business case on top of such a fragmented offering.

However, there are industry initiatives, such as Covesa or Sensoris, which strive to enable stricter standardisation. Widespread adoption of such standards would facilitate the development of data driven services and innovation and benefit European industry. Unless European legislation mandates the use of such standards, under the normal rules of digital markets, the scale of the biggest technology players will result in them creating de facto standards, driven by their wide scale deployments. This would give these technology players an inherent advantage, to the detriment of European industry and result in European standards being defined in Silicon Valley or China rather than in Europe and based on European regulatory or market needs.

In conclusion, the technical elements required for enabling data driven innovation in the automotive market are coming into place. However, current access methods proposed by OEMs to service providers do not meet the needs of industry. A lack of standardisation, incompatible pricing and business models combined with severe limitations on the extent of quality of data being made available are significant impediments to the development of digital ecosystems. In-vehicle application platforms, offering secure and standardised access to vehicle data, functions and resources can transform the data access paradigm.

To simplify it, in digital markets, with their brutal economy of scale, the biggest players that are able to promote the strictest standards will win. Thus, based on current trends, without regulatory intervention, the rules of the economy of scale will predict a likely win for the silicon valley giants and their Asian counterparts, over the fragmented efforts of individual European OEMs.

The outlook for the European automotive industry depends therefore, to a large extent, on the measures taken by the European regulator. Should the EC intervene and promote stricter standardization for authority as well for commercial use cases, the cumulative size of the European Automotive players from OEM as well as IAM side will at least give European industry a fair chance to compete against Silicon Valley and competitors from Asia and to derive full benefit from the digital transition and the opportunities of connected and automated mobility.

## 8 List of Figures