

WG関係者外秘

API Standardization in JASPAR for SDV



***Japan
Automotive
Software
Platform
and
Architecture***

2025 Dec 11th
Yasushi Ando

1. JASPAR Overview
2. API Standardization in JASPAR
3. Future Perspectives

What is JASPAR?

- Japan Automotive Software Platform and Architecture
- Established in 2004
- Focuses on the standardization of automotive software and E/E architecture

Mission

Through standardization activities in Japanese automobile industry, we will realize the following.

- Development of common foundation for automotive industry
- Increased development productivity
- Promoting technological development

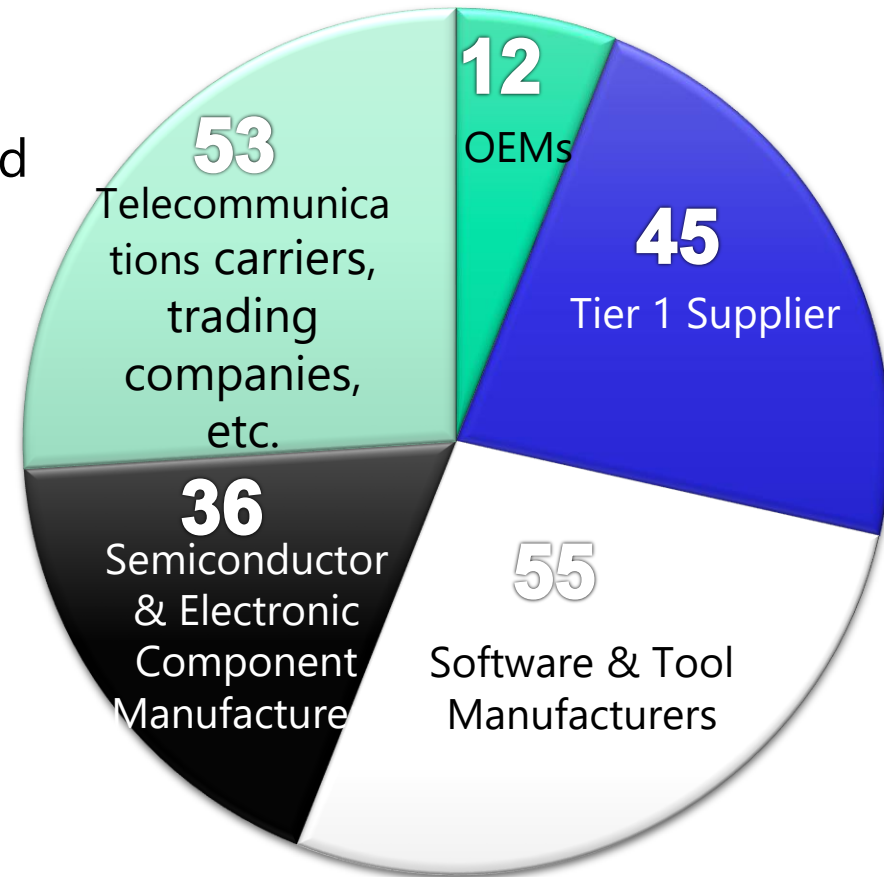
JASPAR promotes the standardization of vehicle systems in Japanese automotive industry

JASPAR Member

- Total corporate members: 201 (as of May 2025)
- In addition to the above, 42 group companies and universities and research institutes (13 people) participated

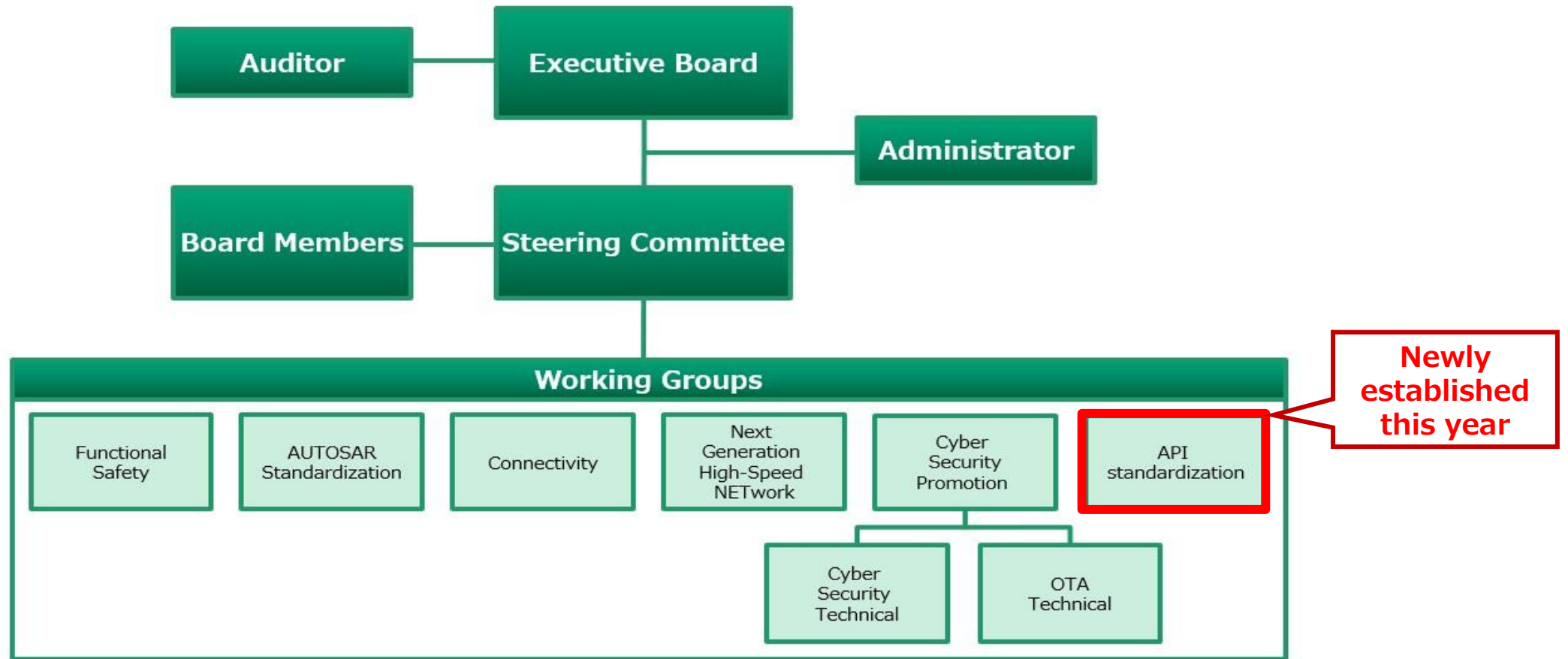
Board Members

- 5 companies lead JASPAR's activities as board companies.
 - Toyota Motor Corporation
 - Nissan Motor Co., Ltd.
 - Honda Motor Co., Ltd.
 - DENSO CORPORATION
 - Toyota Tsusho Corporation



Industry breakdown of corporate members

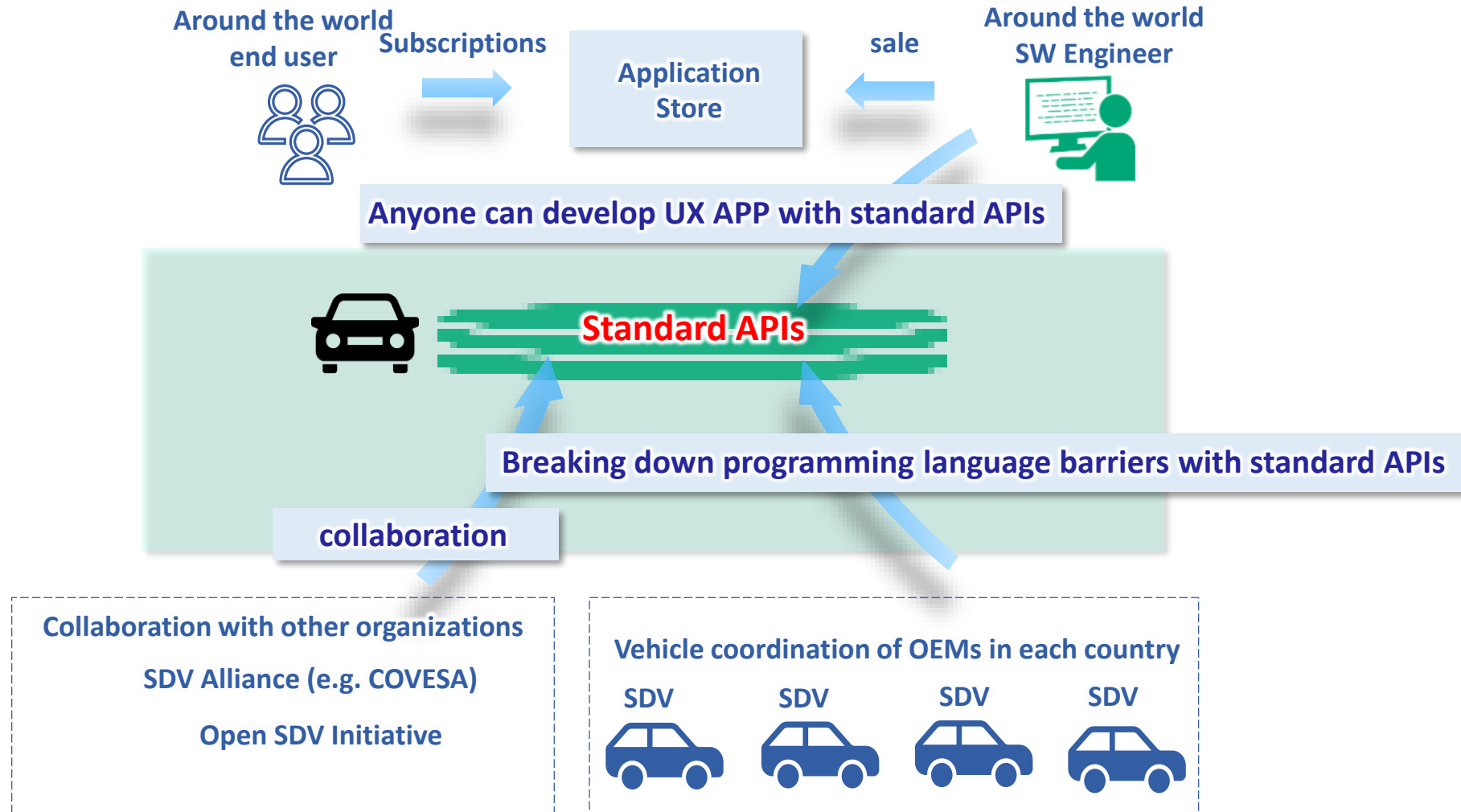
Consist of OEMs, suppliers, SW development companies, and various other companies and organizations related to the automotive industry



7 WGs have been formed to focus on specific technical fields and issues.

1. JASPAR Overview
2. API Standardization in JASPAR
3. Future Perspectives

Standardization of APIs for SDV



To realize SDV, it is essential to standardize the APIs, which is the common language for SDV.

API WG Overview: What will the JASPAR API WG provide?

Comprehensive and User-Friendly APIs

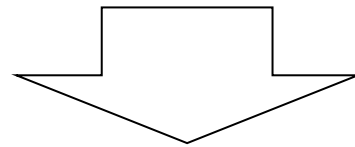
- APIs which abstract vehicle functions
 - I/F Spec Document
 - Source Code
 - API Testing Programs
 - Sample Code (Tutorials)

+

Development Infrastructure/Tools

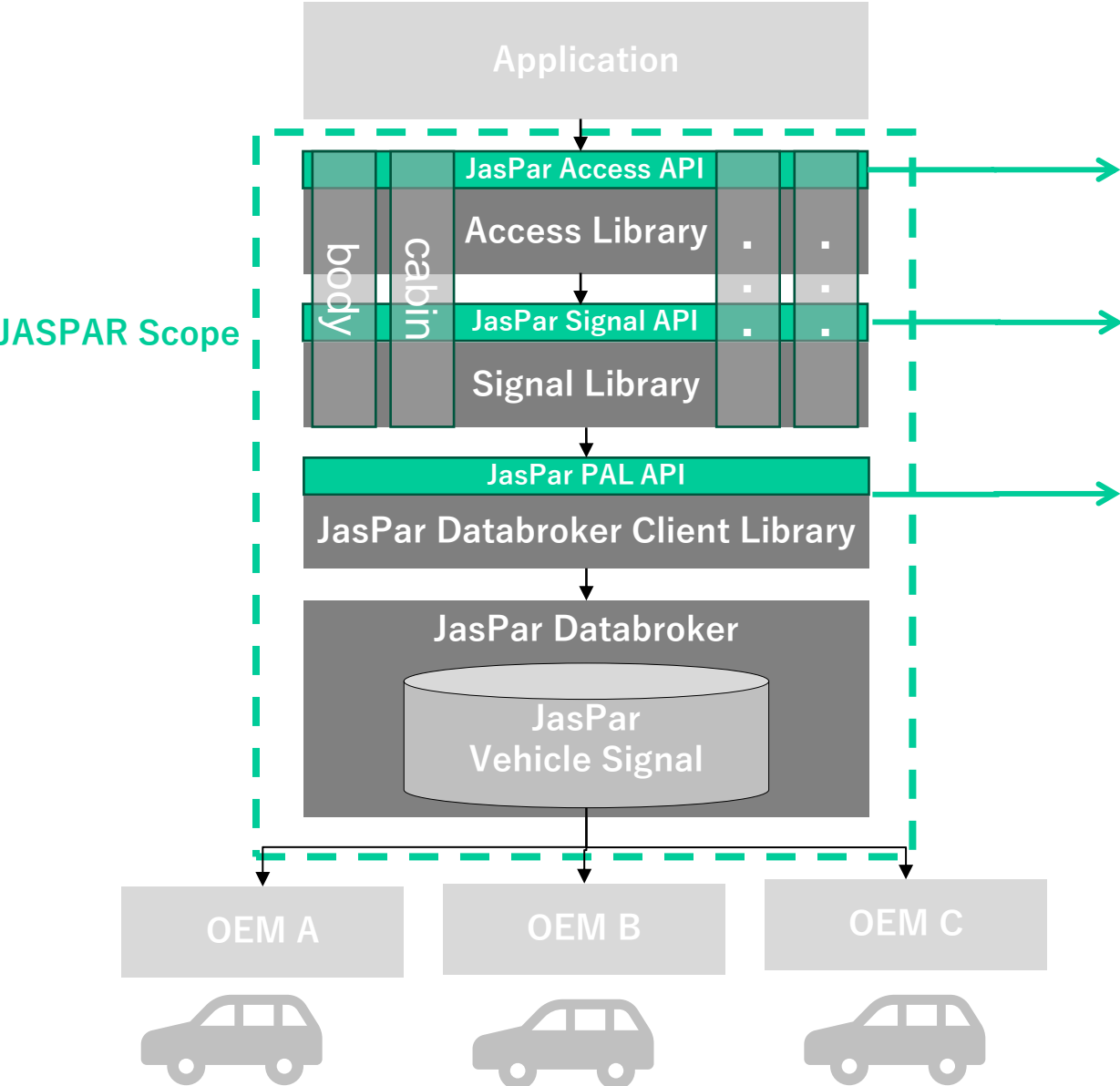
- SDK/Runtime
- API Management Infrastructure, etc.

Release the deliverables as **OSS**



- Improve software development productivity
- Build ecosystems

Architecture Overview



Access API (Under discussion)

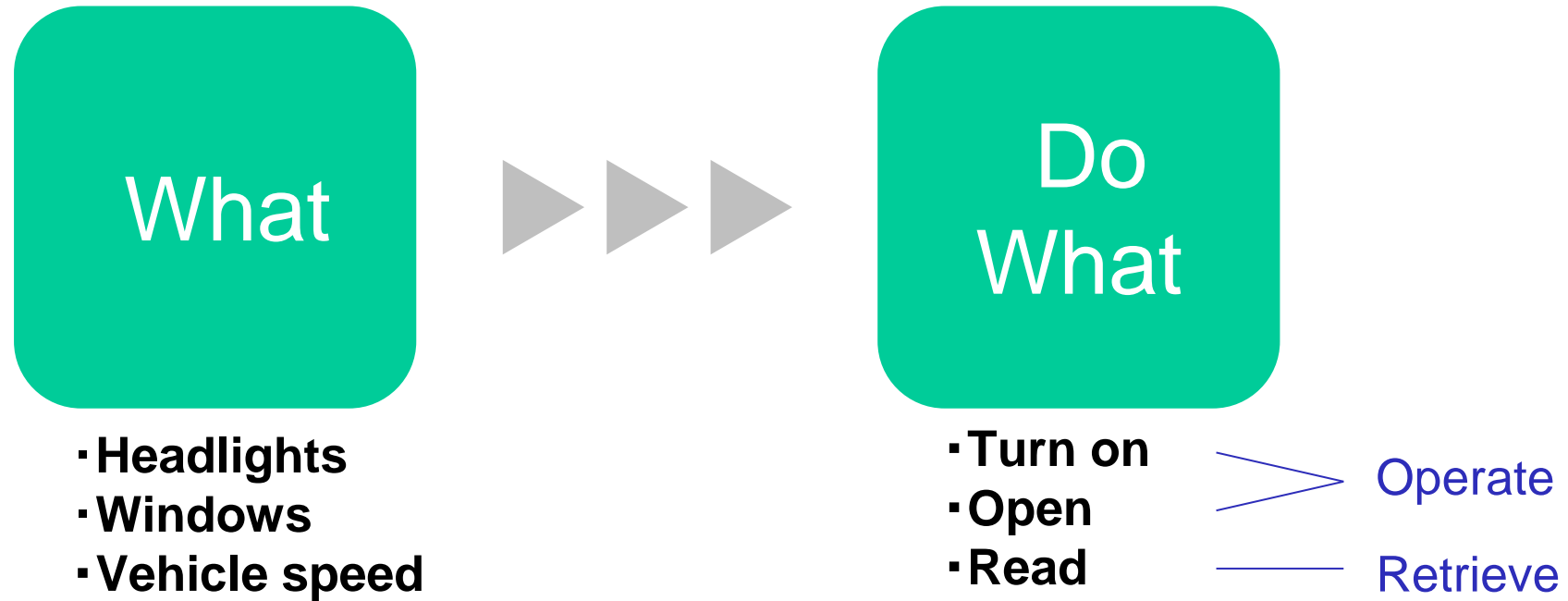
- Abstracted APIs (Abstract data structure (vehicle signals) and provide more user-friendly APIs)

Signal API

- Access Vehicle Signals
- Vehicle Signals are created based on COVESA VSS (standardize signals for vehicle sensor/actuator)

PAL API

- PAL: Platform Abstraction Layer
- Abstract OEMs Middleware/OS

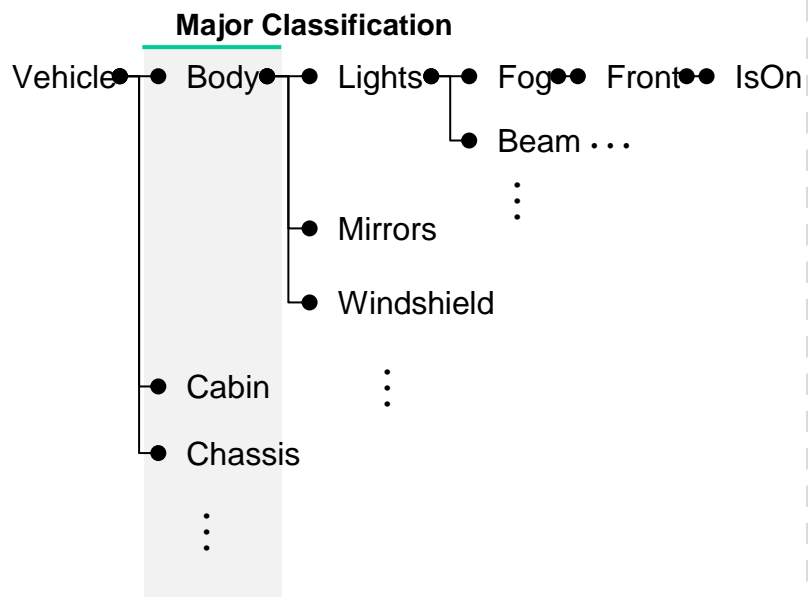


Vehicle equipment and functions vary widely
⇒ Clean up similar/duplicated items based on COVESA VSS
(e.g. defroster, de-icer, defogger...)

"Operate" and **"Retrieve"**

- VSS^[1] : Classifies vehicle equipment and functions with **tree structure** (e.g., Body::Light::Fog Light::Front::ON state)
 - ⇒ Based on this VSS data structure, define APIs that can operate/retrieve the VSS data

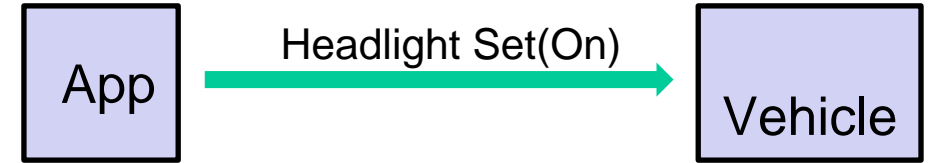
VSS Classification (Tree Structure)



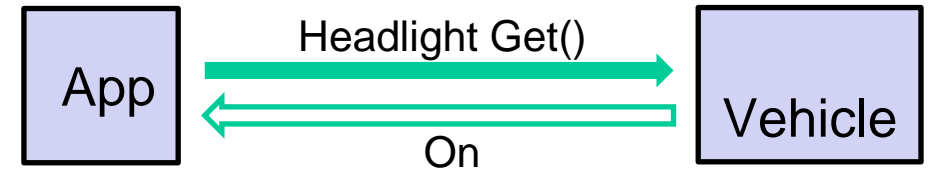
Vehicle Signal Specification:
https://covesa.github.io/vehicle_signal_specification/

Major Category	Overview	Number of Subcategories
Body	Body shape, hood/trunk, horn, rain sensor, windows, lighting system, side mirrors, etc.	14
Cabin	Shades, Air Conditioning, Roof/Doors, Rearview Mirror, Interior Lights, Seats, etc.	30
Occupant	Occupant identification for each seat, head and eye position, etc.	3
LowVoltageBattery	(Auxiliary Battery) Nominal (fully charged) voltage/capacity, current voltage, current balance (current)	1
ADAS	Level, CC, LK, Obstacle Detection, ABS, TCS, ESC, EBD, DMS, etc.	11
Exterior	Outside temperature, humidity, light intensity, etc.	1
Driver	User identification information, distracted driving detection, hands-off detection, information receptivity, fatigue level, heart rate, etc.	2
Diagnostics	Retrieving active diagnostic trouble codes	1
CurrentLocation	Retrieve location information, timestamp, etc.	3
Trailer	Retrieve whether a trailer is connected	1
AngularVelocity	Acquire acceleration in the roll, pitch, and yaw directions of the vehicle	1
Acceleration	Acquire vehicle acceleration in forward/backward, left/right, and up/down directions	1
Connectivity	Determines network connection status with cloud / Availability of offboard functions	1
VehicleIdentification	VIN, WMI, brand, model, year, body type, number, manufacturing date, color, etc.	1
Service	Maintenance service requirement, distance/time to service (dealer?)	1
Chassis	Accelerator, brake, steering, parking brake, etc.	8
Powertrain	Engine/motor signals, fuel/battery signals, transmission signals, etc.	22

- Operate Method
 - Set
(Application performs the operation at any desired time)



- Retrieval Method
 - Get
(App retrieves data at any time)



- Subscribe
(Retrieve when the value changes)



1. JASPAR Overview
2. API Standardization in JASPAR
3. Future Perspectives

(1) Derivalables

- **API**
 - I/F Spec Documentation (incl. behavior definitions)
 - Source Code
 - API Evaluation Program
 - Sample Code
- **SW Development Process and Support Tools**
 - API Design & Development Process
 - SDK, Runtime, and API Management Infrastructure

(2) Scope of Standardization

- **From Interface to API Implementation**
 - Ensure the consistent behavior of APIs by standardizing the Middleware/HAL implementation.
 - Accelerate the development process by utilizing the software assets of participating companies.

(3) OSS

- **Release the Deliverables as OSS**
 - By making them OSS, we aim to ensure that our standardized APIs will be widely used.
 - Promote collaborations with other OSS initiatives (such as AGL ,COVESA, and Eclipse) and ecosystems.

Through these 3 perspectives, we will promote the standardization of APIs and address related topics

- In the MVP phase, establish minimal APIs and infrastructure
- For the Adoption phase, extend and promote the APIs so that many developer can use

	2025/04	2025/12	2026/01	2026/12
	MVP Phase		Adoption Phase	
Activities	<p>Develop minimal APIs and basis for the development</p> <ul style="list-style-type: none"> • Develop primitive APIs for accessing vehicle data based on COVESA VSS. • Make design rules and development process. Set up development infrastructure. 		<p>Develop SDK to promote API adoption</p>	
	<p>API evaluation (PoC)</p> <ul style="list-style-type: none"> • Evaluate the APIs through application development 		<p>Make process to support API adoption (e.g. maintenance process, release process)</p>	
	<p>OSS preparation</p> <ul style="list-style-type: none"> • Make processes for OSS development 		<p>API improvement/extension (e.g. support non-functional requirements, more abstracted APIs)</p>	
			<p>OSS execution</p> <ul style="list-style-type: none"> • Release deliverables as OSS. • Collaborate with other OSS community 	
Release Plan			<p>● R1.0 ('26/xx)</p>	

- JASPAR has started API WG activities involving Japan's leading automobile-related companies since 2025 Feb
- Started Signal APIs development (incl. databroker and development tools) based on COVESA VSS
- JASPAR plans to deliver the first set of APIs in 2026 as OSS
- To promote broader adoption of the APIs, JASPAR is willing to collaborate with various open-source communities and standardization bodies (e.g. COVESA, Eclipse, AGL, OSDVI, etc.)