API-Design and Standardization

COVESA AMM, Novi Michigan, Sept 2024

Achim Henkel

BOSCH

Agenda

- How to bring VSS to an international standard label
- API requirements and API stacks
- Open sourcing OpenAPI creation
- Safety capable API for Motion Controllers
- Next steps

Vehicle API Standardization

One-Pager

Scope

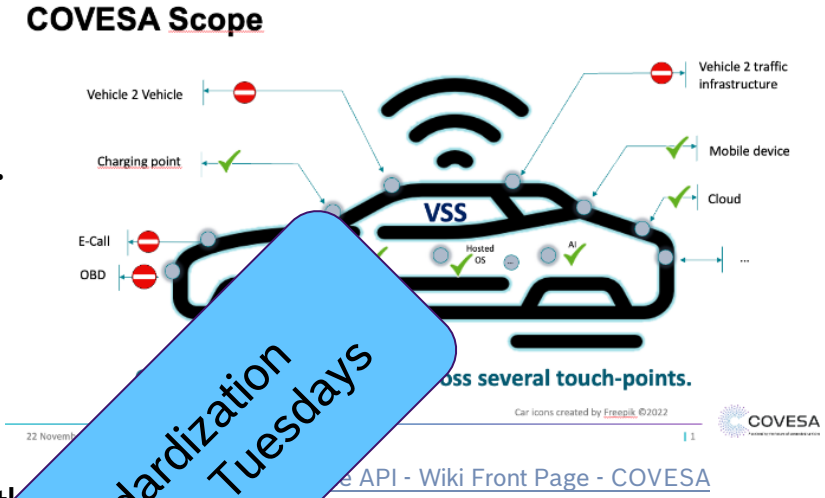
- We aim for a software-standard for a SDV (“Software Defined Vehicle”).
- Providing unified interface to the vehicle sensors/actuators and functions.
- Independent and community-based API definition (e.g., COVESA).

Purpose

- Cross industry usability (OEMs, TIERx, non-automotive partners[1])
- Reducing software-adaptation effort and accelerating field-updates
- Providing regularly new API-extensions (e.g., API-releases all 6 months, t
- Control API access (e.g., token concept)
- Supporting the different local API-initiatives (all over the world).

Solution

- First version of the standard: Data model[2] (semantic for data, q nits) as released by COVESA
- Future versions: API-tooling/-certification, API-syntax, ASIL-exten API-reference-implementation[3].



Joint the standardization
In the CVI group on Tuesdays

1) Charging service provider, cloud service provider, insurances, consumer electronics ...) 2) COVESA VSS Version 4.2 has approx. 1000 data points
3) The standard targets the higher OSI-layers (s. OSI-layer 5 and higher of the OSI Service Conventions (ISO/IEC 7498))

COVESA: Workshop Sept 24th 2024, Novi

Latest News : From Workshop yesterday

What to achieve

One API for all domains (Body, Infotainment, Powertrain..)

Interoperability between different vehicle platforms

Harmonize the interface language in-& outside the vehicle

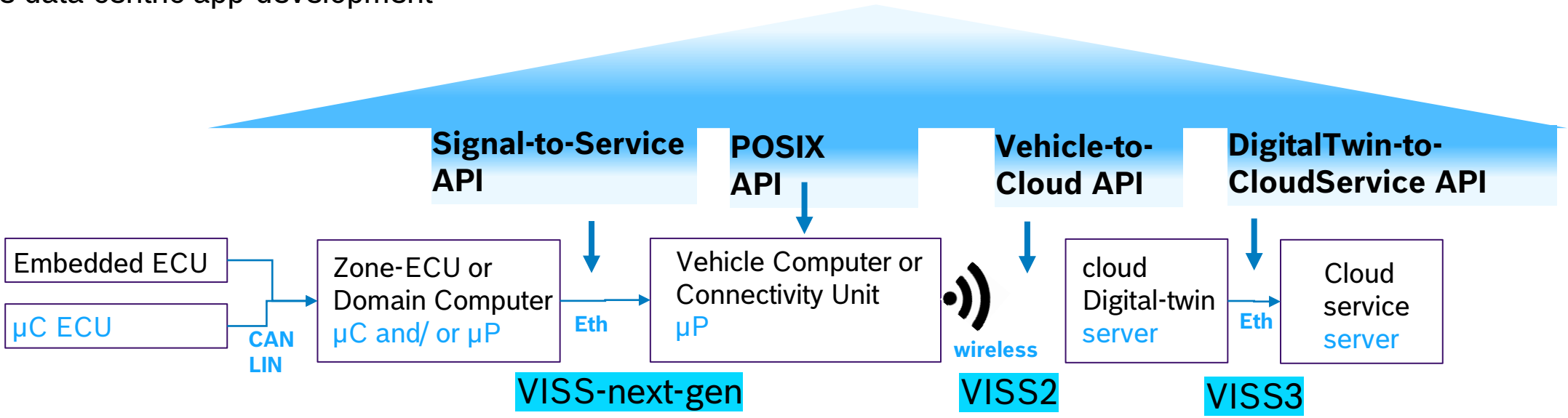
Enable data centric app-development

How to achieve

Start with a Data-based API standard
(function/capabilities-based-API might follow later)

Step 1: One data model for all APIs (VSS+units+Quantities)

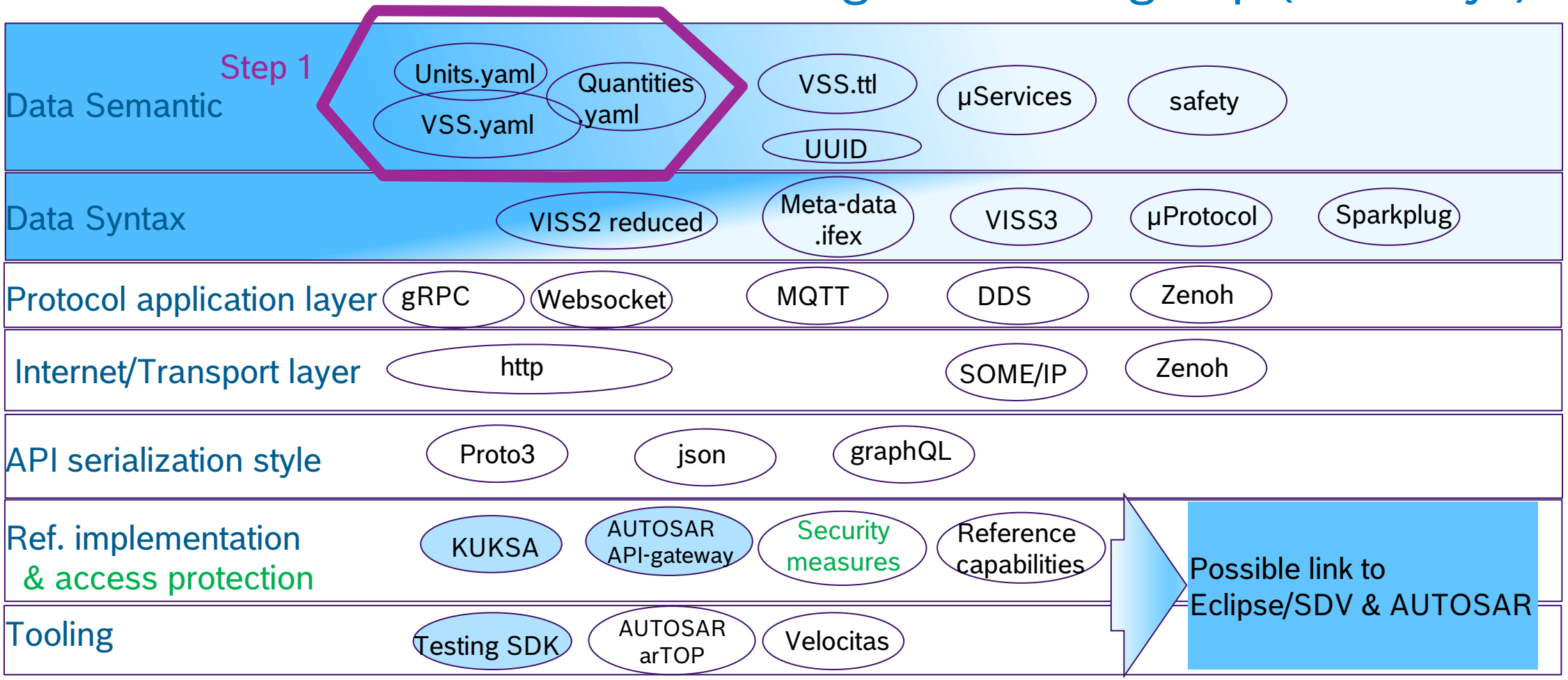
Step 2: Enable different implementation (e.g. VISS-versions)



BOSCH

COVESA: Workshop Sept 24th 2024, Novi

API Standardisation Elements are aligned in CVI group (Tuesdays)



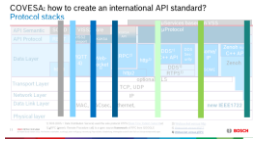
CVI : Common Vehicle Interface group as part of the data expert group

BOSCH

Agenda

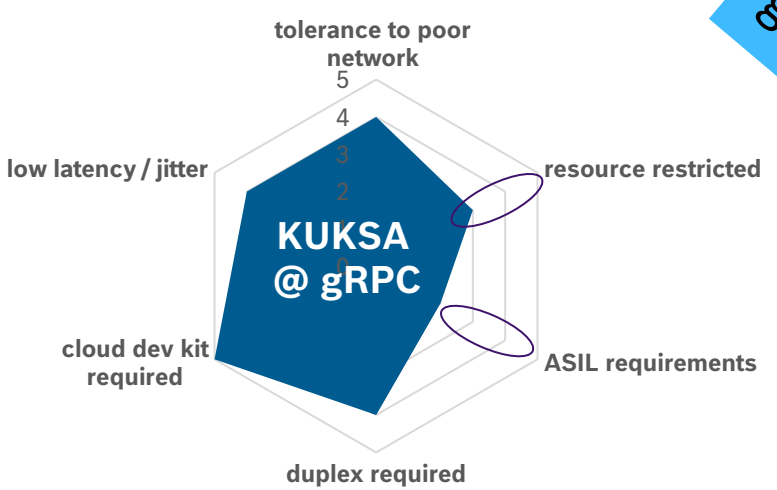
- How to bring VSS to an international standard label
- API requirements and API stacks
- Open sourcing OpenAPI creation
- Safety capable API for Motion Controllers
- Next steps

API stack performance indicators



Reading		Reading & Writing
SOVD	VISS2/3 incl KUKSA	µServices

- We need a gRPC-integration to access easily the cloud-community (best with SOME/IP & DDS-interfaces)
- **gRPC: ASIL performance and resource efficient implementation** is crucial for market success
  - The gRPC implementation combined with a PROTO-interface description is best positioned to create an alternative / **bridge to existing solutions** of the smartphone APIs.
  - **Protobuf** might be the best compromise for an API syntax description for all different use-cases



gRPC (proto) is Open Source

BOSCH

Agenda

- How to bring VSS to an international standard label
- API requirements and API stacks
- Open sourcing OpenAPI creation
- Safety capable API for Motion Controllers
- Next steps

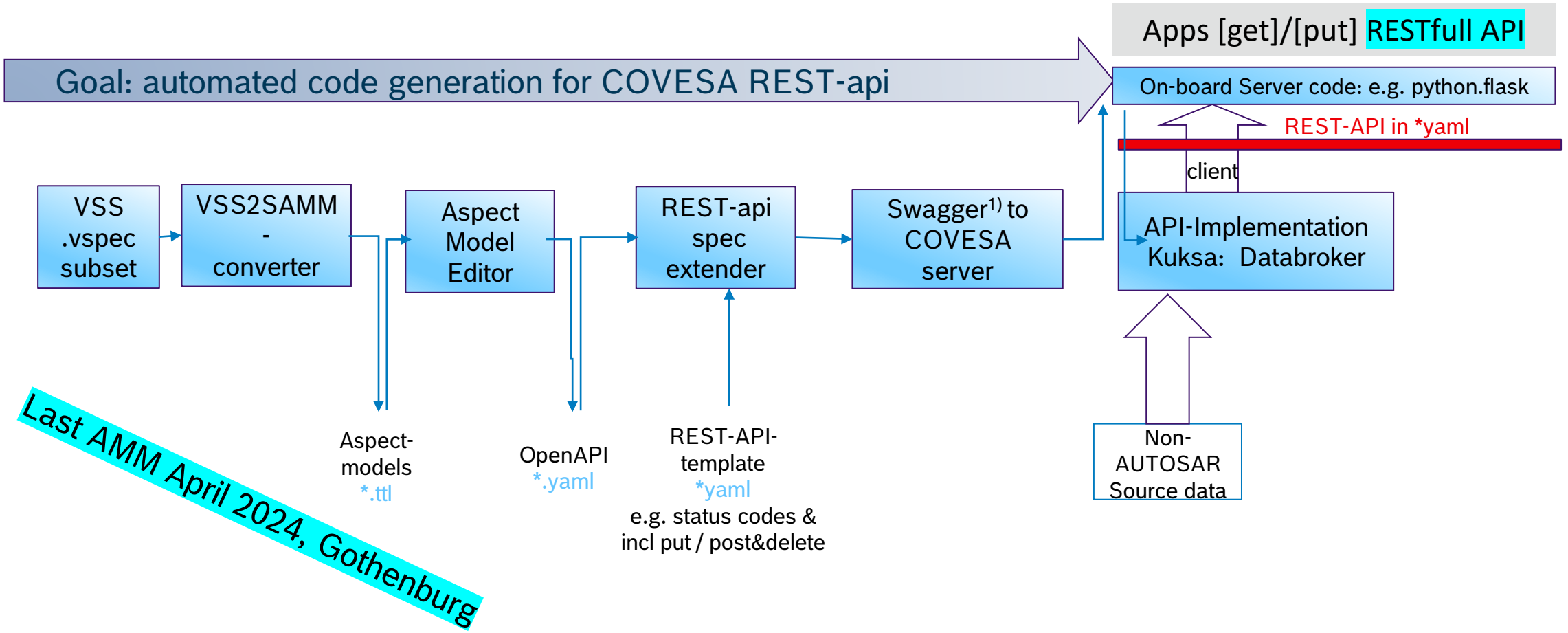
BOSCH

AUTOSAR-COVESA Worksplit (Bosch-proposal)
Extension COVESA on-board Workflow

Non AUTOSAR

AUTOSAR-IP

Open Source: Mozilla Public License 2.0



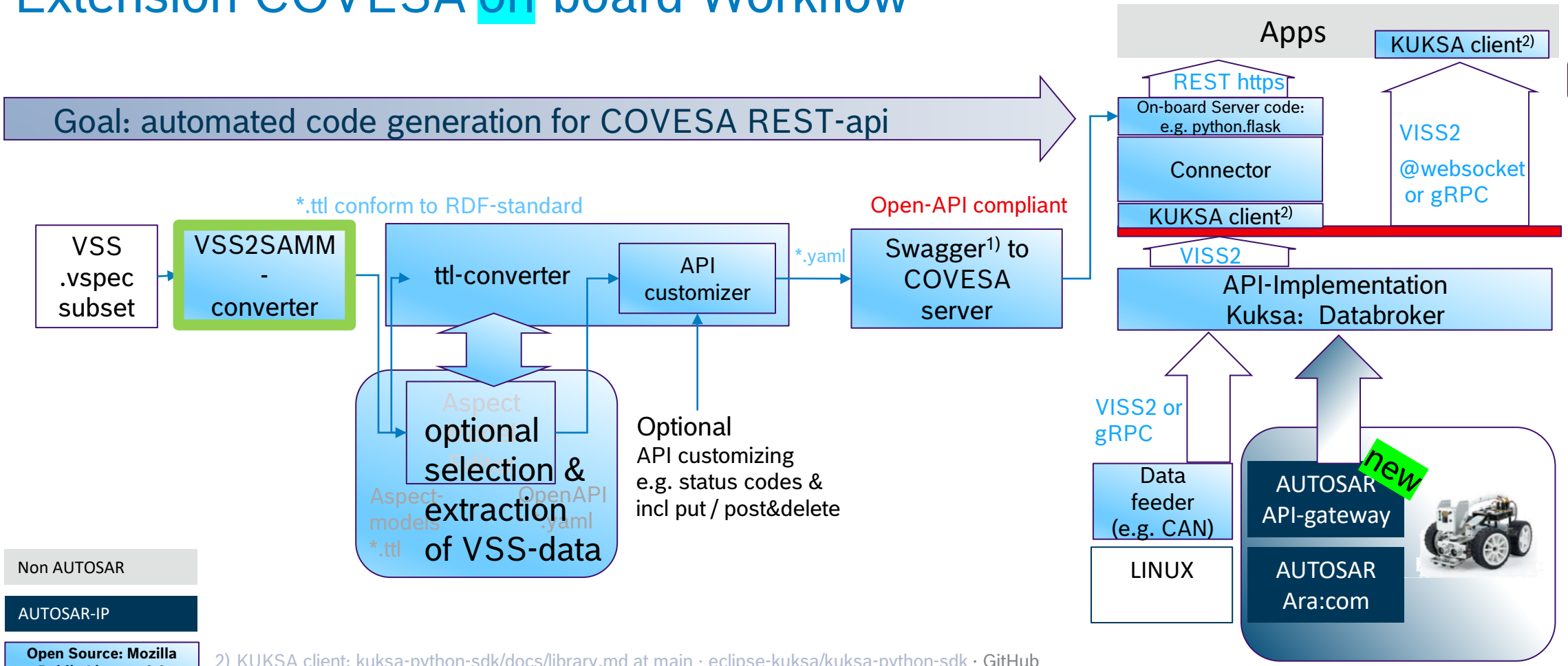
yaml is supporting comments and type-referencing e.g. multi-use type “door”

1) Swagger Editor (bosch.com)

BOSCH

AUTOSAR-COVESA Worksplit (Bosch-proposal)

Extension COVESA on-board Workflow

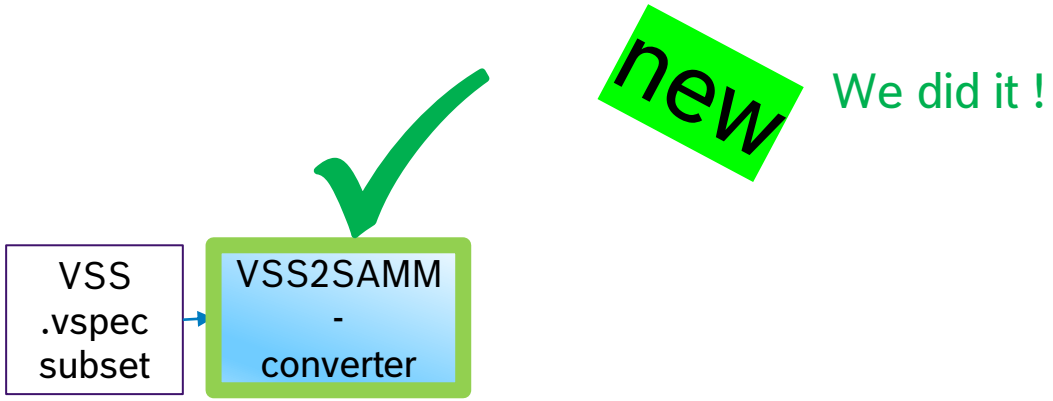


2) KUKSA client: kuksa-python-sdk/docs/library.md at main · eclipse-kuksa/kuksa-python-sdk · GitHub

1) Swagger Editor (bosch.com)

BOSCH

AUTOSAR-COVESA Worksplit (Bosch-proposal)
Extension COVESA on-board Workflow



Contribution to the VSS-tool repository available:
Converter from VSS.vspec to
a Semantic Aspect Meta Model (SAMM)
Name of Converter: VSS2SAMM (Phyton-script)

Converter docu

vss-tools/docs/samm.md at master · COVESA/vss-tools · GitHub

The resulting SAMM is RDF-standard[1] conform

It can easly imported in the AME
(Aspect Model Editor in Eclipse)

Download esmf-AME editor

Releases · eclipse-esmf/esmf-aspect-model-editor (github.com)

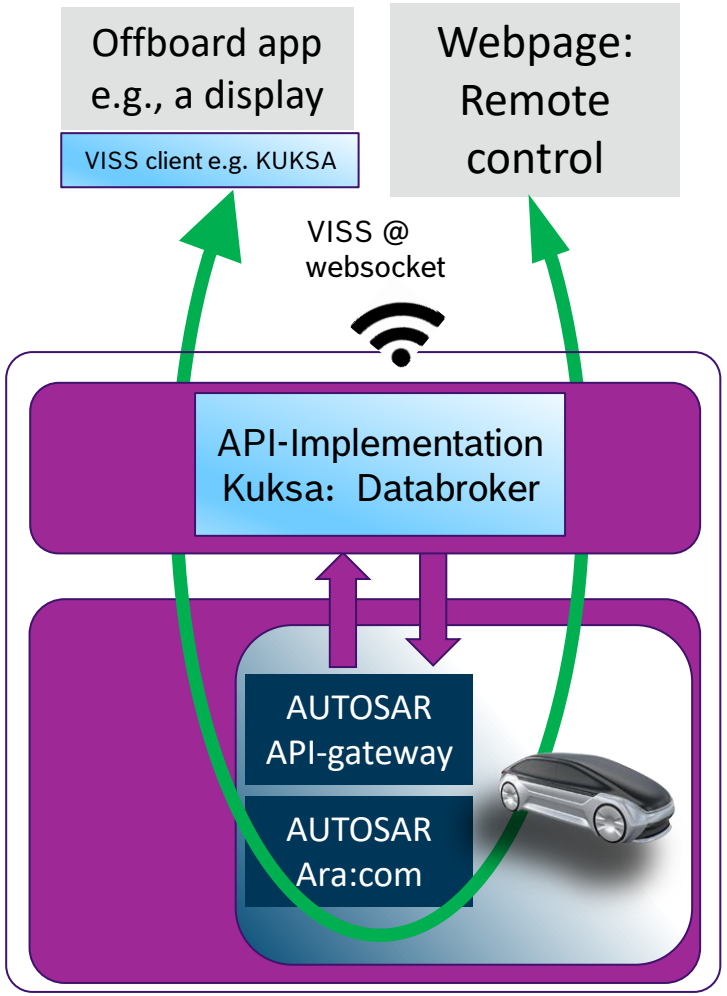
1) https://www.w3.org/RDF/

BOSCH

AUTOSAR-COVESA Worksplit (Bosch-proposal)

Workflow ready for V&V and virtual testing

1. Complete virtual testing tool-chain possible
2. Different blocks can be virtualized (e.g. docker)
3. Compatible with AUTOSAR
4. Incl. Token access control (e.g. JWT)
5. Bidirectional (e.g. control command “stop the car”)



virtualized

AUTOSAR-IP

Open Source: Mozilla Public License 2.0

yaml is supporting comments and type-referencing e.g. multi-use type “door”

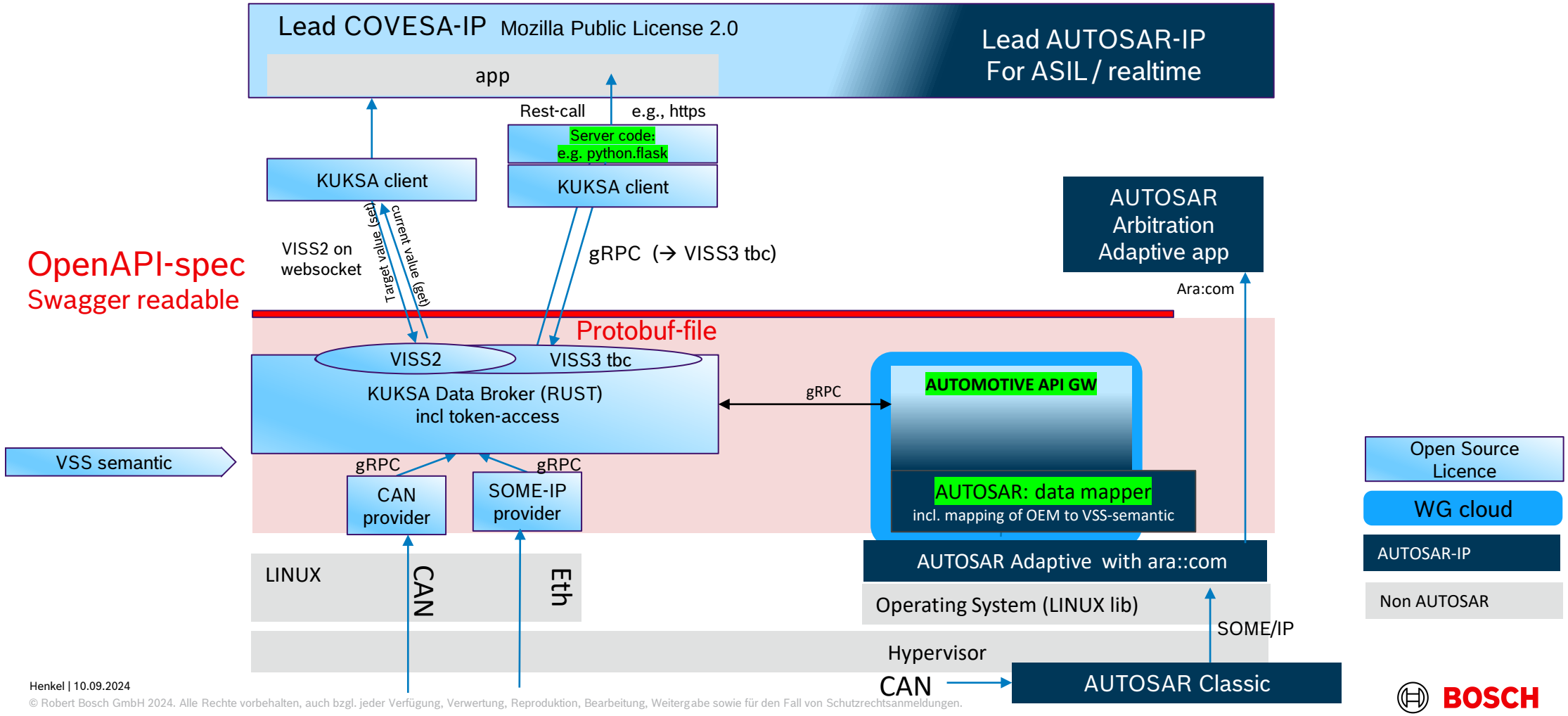
BOSCH

Agenda

- How to bring VSS to an international standard label
- API requirements and API stacks
- Open sourcing OpenAPI creation
- Safety capable API for Motion Controllers
- Next steps

AUTOSAR-COVESA Worksplit (Bosch-proposal)

Concept for the AUTOSAR / COVESA API

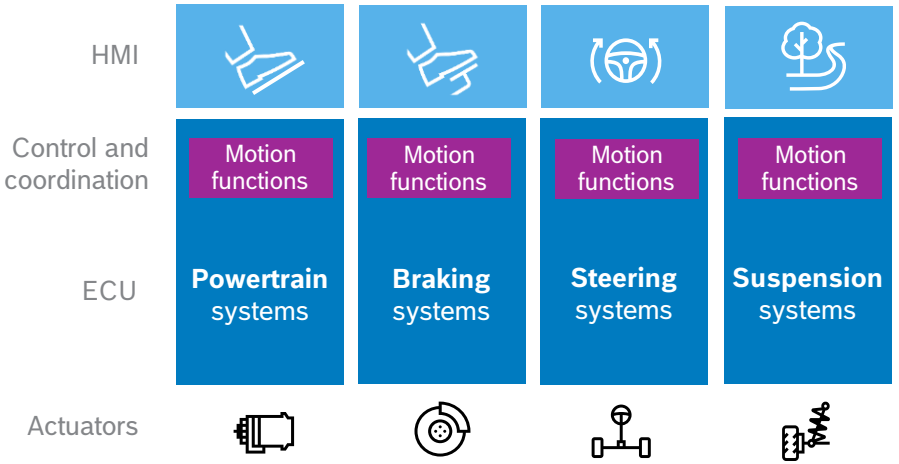


BOSCH

Trends towards centralized and integrated vehicle motion functions

Distributed

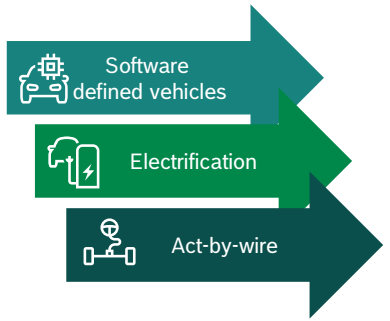
Distributed vehicle motion functions



- Motion features spread to control units of specialized actuators
- Interaction between domain specific implementations through multiple interfaces

Trends

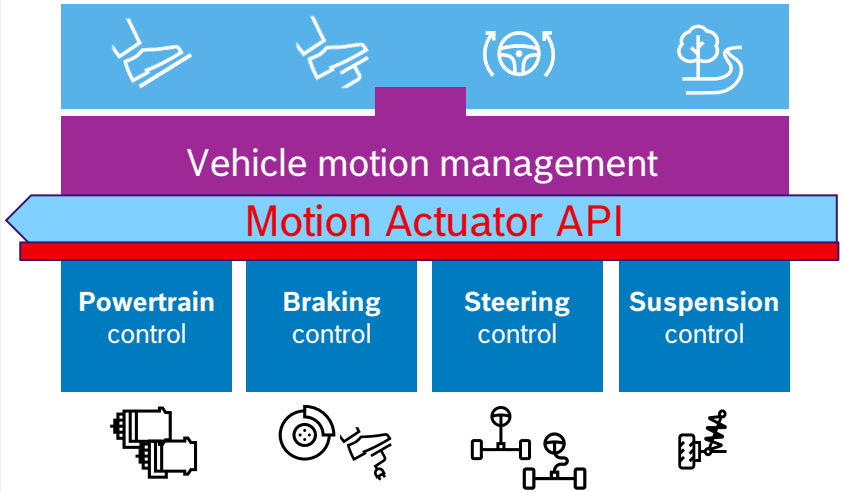
New market requirements



- Flexible deployment in new E/E architectures
- Use potential of new vehicle motion actuators
- Manage complexity and software reuse
- Scalability: Actuators and vehicle segments

Centralized

Centralized and integrated vehicle motion functions



- Functional integration and centralization
- Common control and coordination enables new functions, avoids complexity and leverages full actuator potential

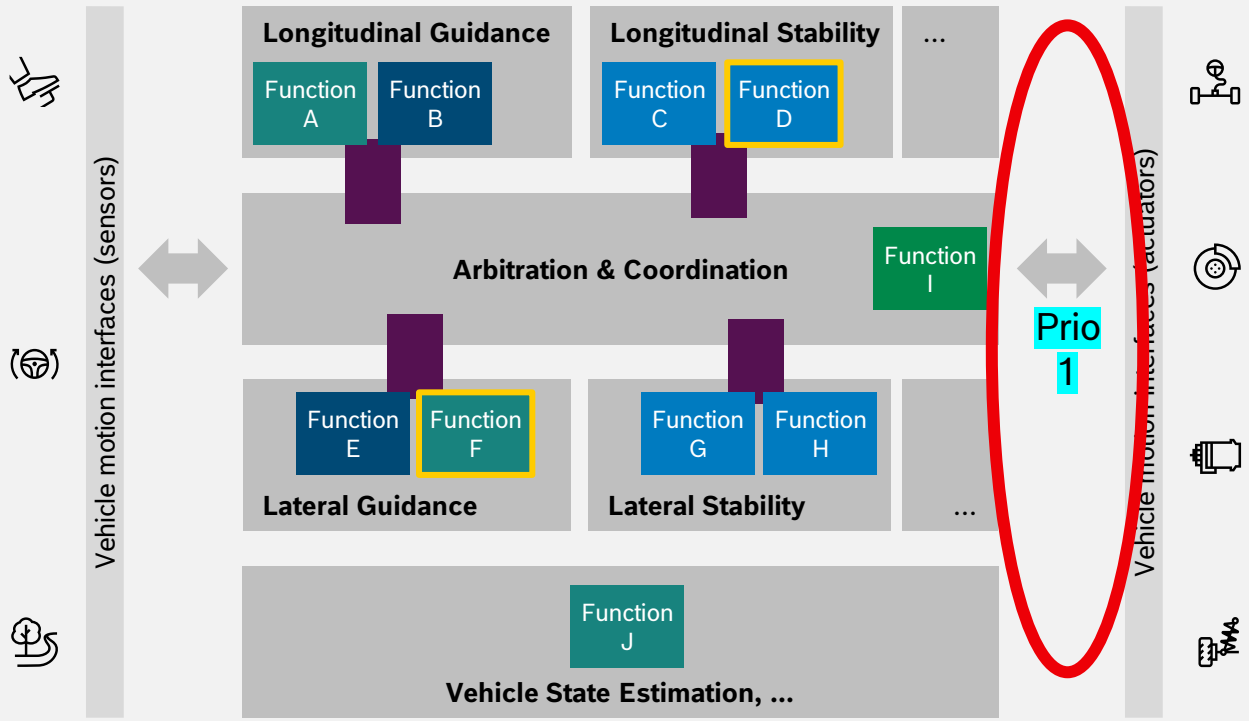
VMM architecture principle and demand for standardization

Why

Change from deeply embedded (HW/SW co-design) towards centralized motion SW requires:

- Modular & scalable motion architecture
- xDomain cooperation ability to cope w/ functions, control units and actuators coming from different partners
- Configurable modules
- Re-use of SWCs
- Flexible SW deployment in various E/E architectures
- Multi-actuator control for all features
- SW driven product design

Bosch VMM architecture



Concept

- Modular architecture set-up divided by motion coordination tasks
- Standardized interfaces (API) to actuator level and in between modules
- Central arbitration to reduce complexity and to enable xDomain functions and cooperation

Challenge: Fit into each OEM architecture incl. their legacy

Let us shape the vehicle motion management together by:

- Aligned architecture
- Standardized interfaces

API: Application programming interface

Third party function

Agenda

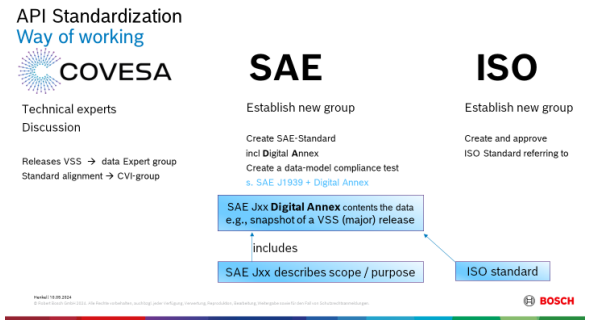
- How to bring VSS to an international standard label
- API requirements and API stacks
- Open sourcing OpenAPI creation
- Safety capable API for Motion Controllers
- Next steps

BOSCH

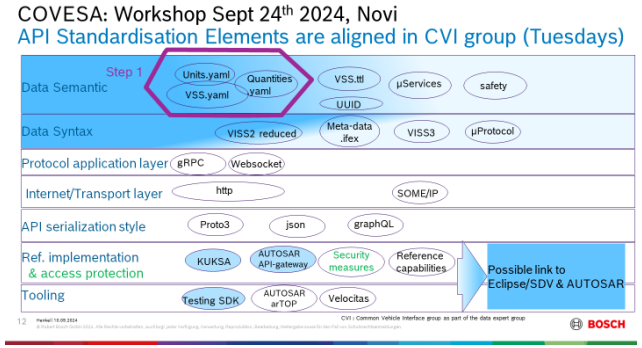
COVESA: how to create an international API standard?

Next steps: Let's come together

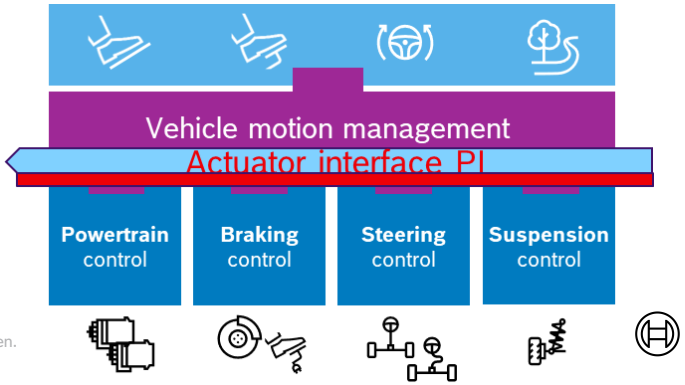
Join CVI group for aligning the **standardization** content
Decide on Standard body (SAE, ISO, IEEE,...)



Join **API** discussion and **reference implementation**
(e.g., Proto3, OpenAPI, GraphQL, ...)
in **CVI group (Tuesdays)**



Join **safety extension** for Motion API in data expert group



BOSCH

Thank you

to Martin Lunt, Kostadin Ivanov, Felix Loesch, Markus Heger, Andreas Lock

BOSCH