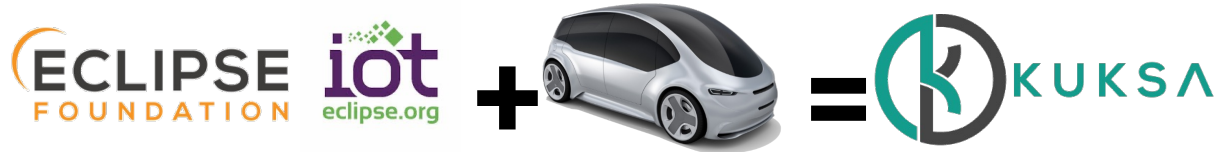


Eclipse Kuksa

Eclipse SDV Contribution Day,
Sven Erik Jeroschewski, Bosch.IO

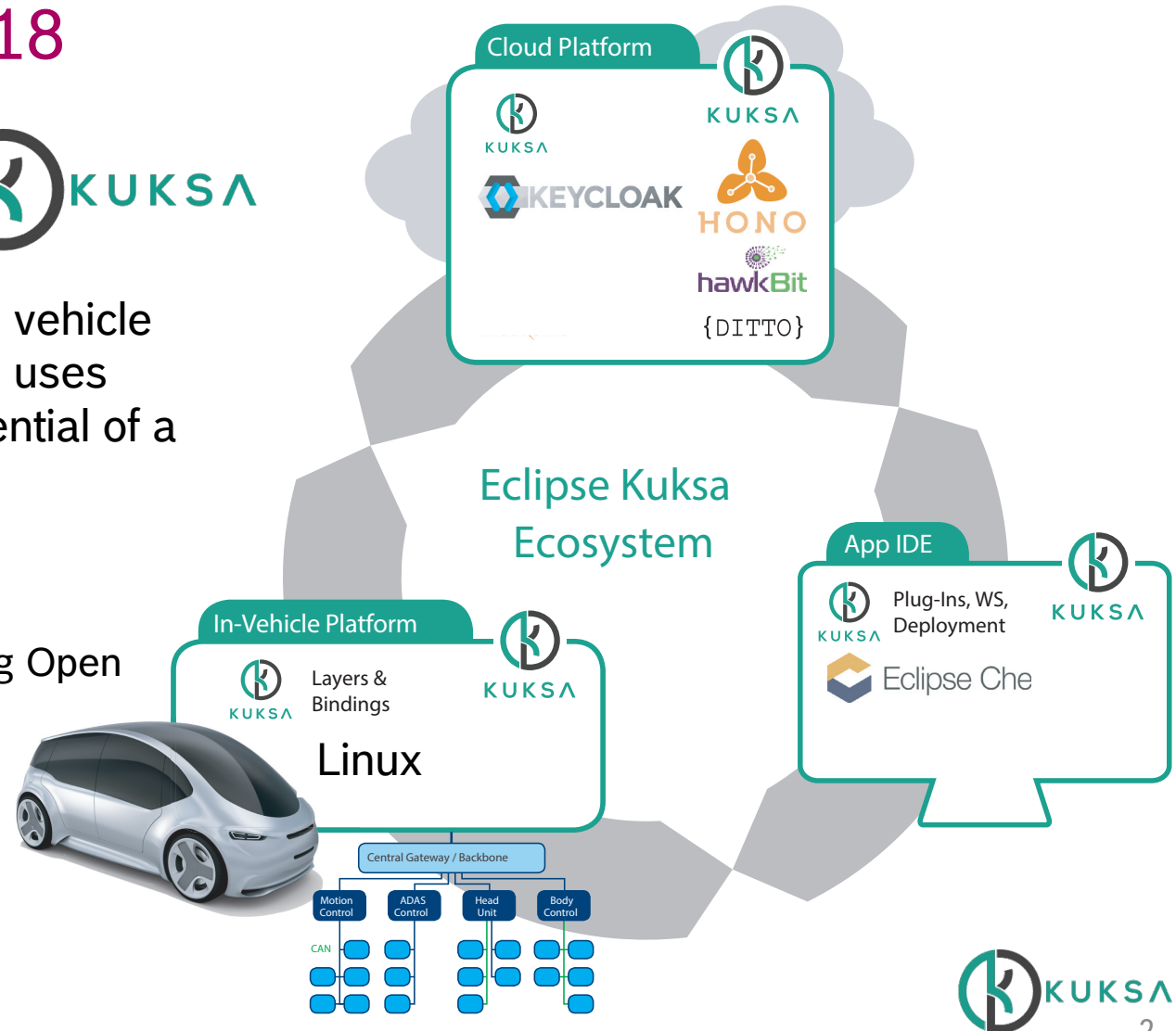
Eclipse Kuksa

The Kuksa Ecosystem in 2018



Mission: Create a **cross-vendor** connected vehicle platform that relies on **open standards** and uses **open source software** to leverage the potential of a **large developer community!**

- use and foster Open Source solutions
- create a harmonized composition of existing Open Source projects
- enriched with specific Kuksa components



Software Infrastructure for Vehicles

What if ..

...we can access standardized car data in our cloud backend

...we can use standard IT technologies to run software around vehicles

...we can use these applications to interact with the car

...everybody has the option to use these features

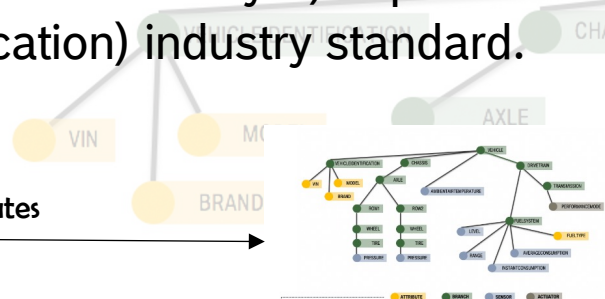
...we can focus on developing customer value instead of reinventing the whole stack

An Implementation of Open Standards

Kuksa.val (Vehicle Abstraction Layer) implements an in-vehicle server for the Covesa VSS (Vehicle Signal Specification) industry standard.



creates



Cooperation



Industry association
<https://www.covesa.global>

VSS: Open data model, taxonomy of sensors & actors in a car
https://covesa.github.io/vehicle_signal_specification/

W3C Automotive
<https://www.w3.org/auto/wg>

creates

Technical specification(s) and protocol(s) to access VSS data (VISS – Vehicle Information Service Specification)

<https://www.w3.org/TR/vehicle-information-service/>



uses

implements

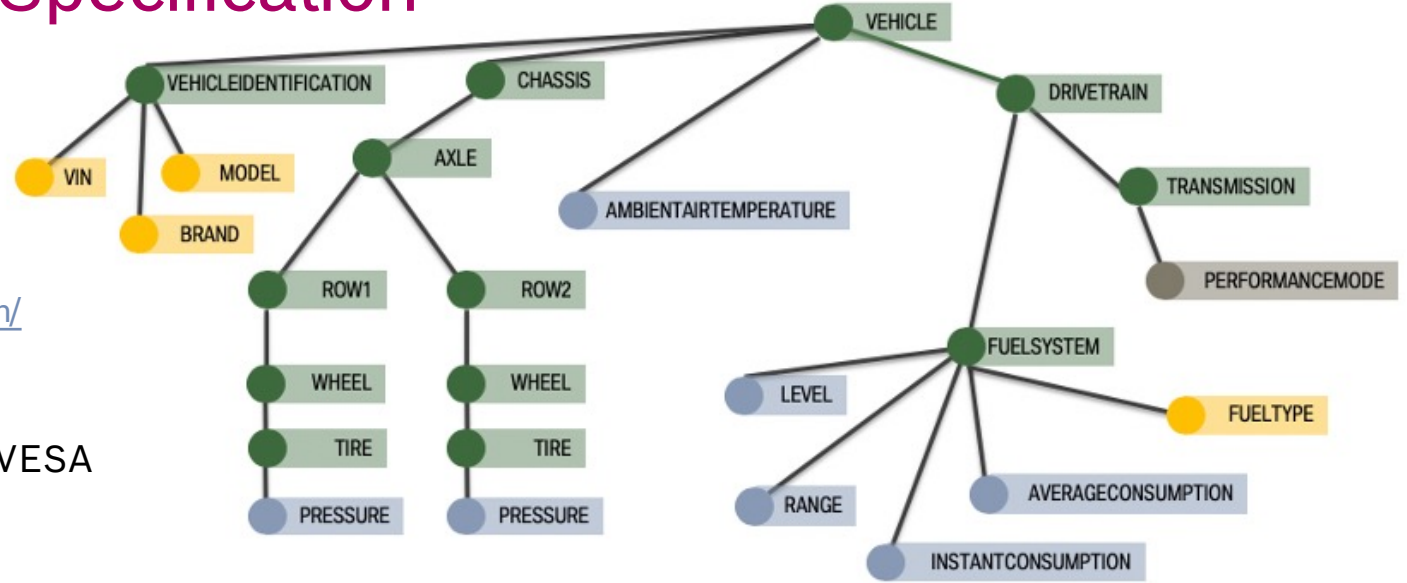
VSS – Vehicle Signal Specification

A generic, extensible data model currently specified within COVESA

https://covesa.github.io/vehicle_signal_specification/

A protocol to access data based on the COVESA VSS model specified within W3C

<https://www.w3.org/TR/vehicle-information-service/>



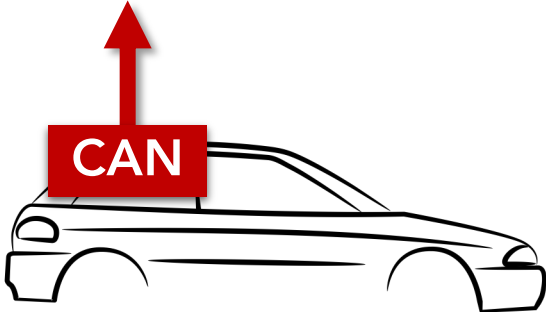
LEGEND:

- ATTRIBUTE (Yellow circle)
- BRANCH (Green circle)
- SENSOR (Blue circle)
- ACTUATOR (Grey circle)

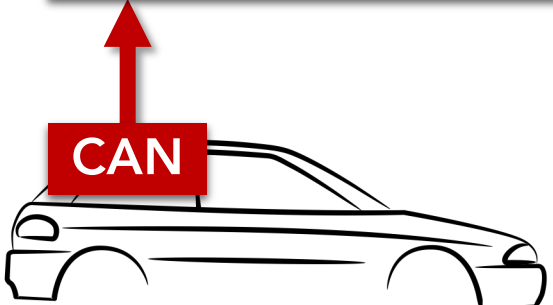
```
- Drivetrain.Transmission.Speed:  
  type: sensor  
  datatype: uint16  
  unit: km/h  
  min: 0  
  max: 300  
  description: The vehicle speed, as measured by the drivetrain.
```

Kuksa.val

System Overview



Kuksa.val System Overview



```
vcan0 442 [8] AF EF 9C 09 52 15 30 43
vcan0 6D5 [8] 08 F5 D8 5F 19 14 3B 57
vcan0 5FB [0]
vcan0 2BA [8] 48 F6 A1 74 92 28 97 38
vcan0 26C [8] AA 0A A0 7E E9 32 AA 67
vcan0 72A [3] CB 1B C7
vcan0 4DD [8] 8C 37 6D 7E 39 AB 2E 3C
vcan0 4C0 [8] 1E D5 D7 13 02 91 A6 47
vcan0 135 [8] 54 A6 D6 0A 0A A0 15 49
vcan0 2F6 [8] 24 B4 50 20 F2 70 3F 67
vcan0 26D [8] 5F 57 84 24 27 8D 9D 20
vcan0 714 [7] BA B5 34 59 80 19 CA
vcan0 123 [4] 69 4C 74 2F
vcan0 588 [4] 40 71 EB 74
vcan0 141 [8] 9E 5D 7D 40 01 1E C7 46
vcan0 1F4 [8] 03 AF 6D 0E 2A BB AF 3F
vcan0 789 [8] 34 5B C5 08 80 DA C4 5A
vcan0 0B5 [8] 72 4B 04 42 22 EB E2 5B
vcan0 0A4 [1] 49
vcan0 079 [3] 38 71 14
vcan0 668 [2] EA F1
vcan0 5D6 [8] 55 76 34 0D 18 77 E9 6D
vcan0 3F6 [4] 19 95 80 34
vcan0 680 [8] 15 41 8F 1E 3A FE D0 13
vcan0 175 [8] BA D8 95 6E 2A 8A 56 60
```



Kuksa.val

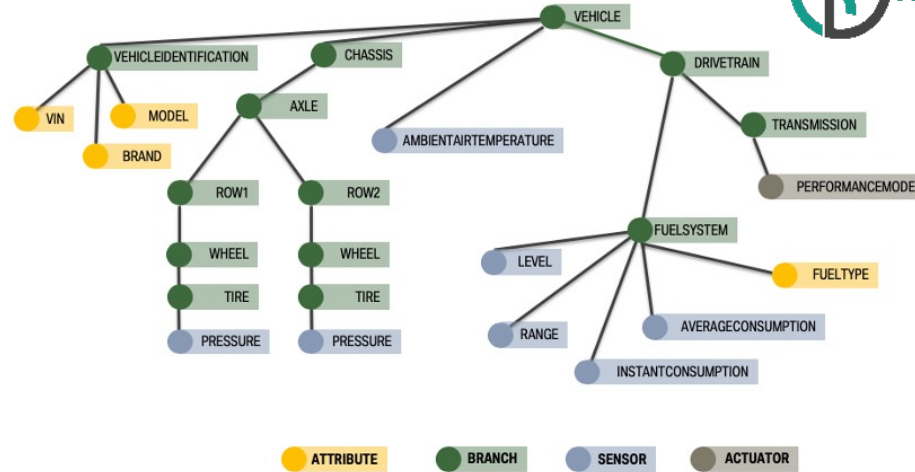
System Overview

```

3000 B0_599 ID257UISpeed: 8 VehicleBus
3001 SG_UIspeedUnits257 : 32|12@1+ (1,0) [0|0] "" Receiver
3002 SG_UIspeed_signed257 : 12|12@1+ (0.08,-40) [-40|287.6] "KPH" Receiver
3003 SG_UIspeed_abs257 : 24|8@1+ (1,0) [0|255] "MPHKPH" Receiver
3004 SG_UIspeedChecksum257 : 0|8@1+ (1,0) [0|0] "" Receiver
3005 SG_UIspeedCounter257 : 8|4@1+ (1,0) [0|0] "" Receiver
3006 SG_UIspeedHighSpeed257 : 33|9@1+ (1,0) [0|510] "MPHKPH" Receiver
    
```

```

3 UIspeed_sig
4 minupdate
5 targets:
6 - Vehic
    
```



```

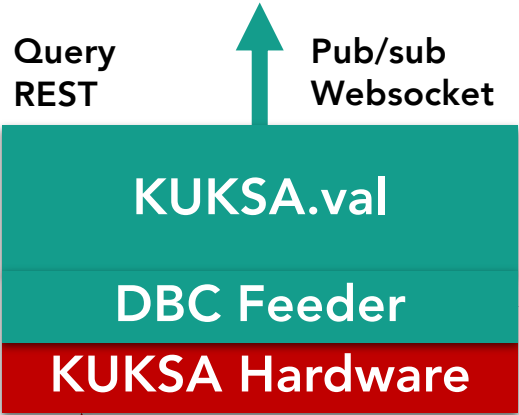
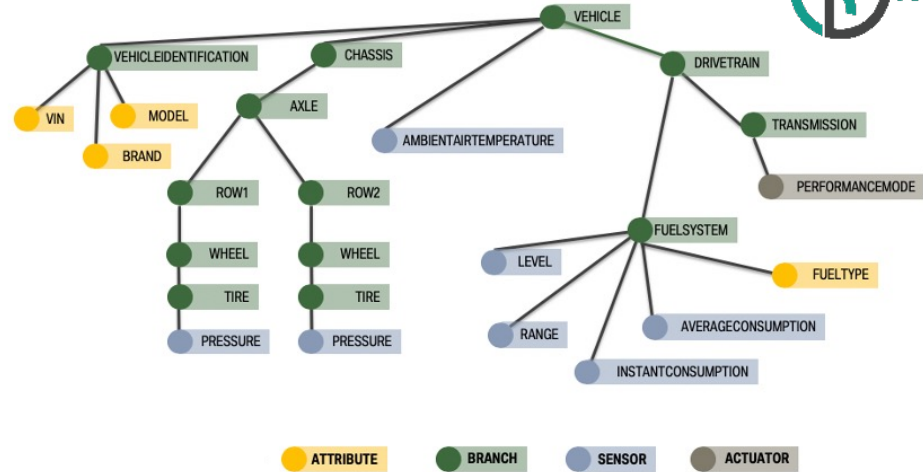
vcan0 442 [8] AF EF 9C 09 52 15 30 43
vcan0 6D5 [8] 08 F5 D8 5F 19 14 3B 57
vcan0 5FB [0]
vcan0 2BA [8] 48 F6 A1 74 92 28 97 38
vcan0 26C [8] AA 0A A0 7E E9 32 AA 67
vcan0 72A [3] CB 1B C7
vcan0 4DD [8] 8C 37 6D 7E 39 AB 2E 3C
vcan0 4C0 [8] 1E D5 D7 13 02 91 A6 47
vcan0 135 [8] 54 A6 D6 0A 0A A0 15 49
vcan0 2F6 [8] 24 B4 50 20 F2 70 3F 67
vcan0 26D [8] 5F 57 84 24 27 8D 9D 20
vcan0 714 [7] BA B5 34 59 80 19 CA
vcan0 123 [4] 69 4C 74 2F
vcan0 588 [4] 40 71 EB 74
vcan0 141 [8] 9E 5D 7D 40 01 1E C7 46
vcan0 1F4 [8] 03 AF 6D 0E 2A BB AF 3F
vcan0 789 [8] 34 5B C5 08 80 DA C4 5A
vcan0 0B5 [8] 72 4B 04 42 22 EB E2 5B
vcan0 0A4 [1] 49
vcan0 079 [3] 38 71 14
vcan0 668 [2] EA F1
vcan0 5D6 [8] 55 76 34 0D 18 77 E9 6D
vcan0 3F6 [4] 19 95 B0 34
vcan0 6B0 [8] 15 41 8F 1E 3A FE D0 13
vcan0 175 [8] BA D8 95 6E 2A 8A 56 60
    
```



Kuksa.val

System Overview

```
{
  "action": "subscription",
  "subscriptionId": "42",
  "path": "Vehicle.Speed",
  "value": "100"
}
```

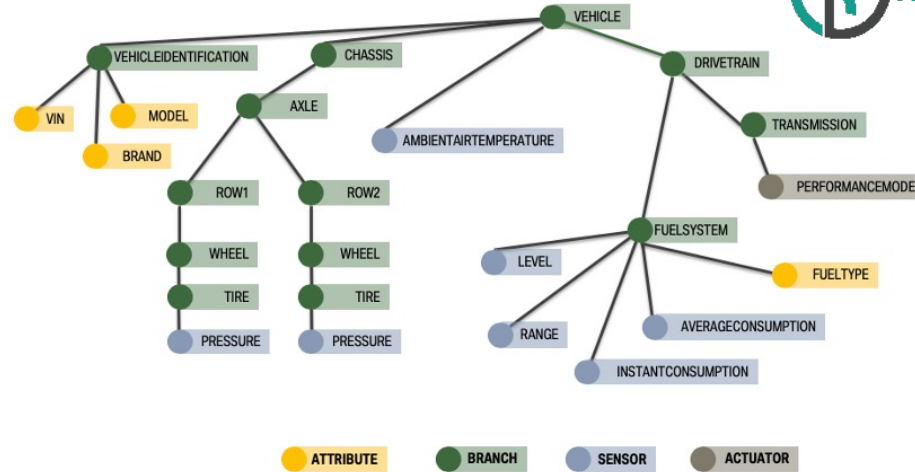


vcn0	442	[8]	AF	EF	9C	09	52	15	30	43
vcn0	6D5	[8]	08	F5	D8	5F	19	14	3B	57
vcn0	5FB	[0]								
vcn0	2BA	[8]	48	F6	A1	74	92	28	38	
vcn0	26C	[8]	AA	0A	A0	7E	E9	32	AA	67
vcn0	72A	[3]	CB	1B	C7					
vcn0	4DD	[8]	8C	37	6D	7E	39	AB	2E	3C
vcn0	4C0	[8]	1E	D5	D7	13	02	91	A6	47
vcn0	135	[8]	54	A6	D6	0A	0A	15	49	
vcn0	2F6	[8]	24	B4	50	20	F2	70	3F	67
vcn0	26D	[8]	5F	57	84	24	27	8D	9D	20
vcn0	714	[7]	BA	B5	34	59	80	19	CA	
vcn0	123	[4]	69	4C	74	2F				
vcn0	588	[4]	40	71	EB	74				
vcn0	141	[8]	9E	5D	7D	40	01	1E	C7	46
vcn0	1F4	[8]	03	AF	6D	0E	2A	BB	AF	3F
vcn0	789	[8]	34	5B	C5	08	80	DA	C4	5A
vcn0	0B5	[8]	72	4B	04	42	22	EB	E2	5B
vcn0	0A4	[1]	49							
vcn0	079	[3]	38	71	14					
vcn0	668	[2]	EA	F1						
vcn0	5D6	[8]	55	76	34	0D	18	77	E9	6D
vcn0	3F6	[4]	19	95	B0	34				
vcn0	6B0	[8]	15	41	8F	1E	3A	FE	D0	13
vcn0	175	[8]	BA	D8	95	6E	2A	8A	56	60



Kuksa.val

System Overview



Data Consumer

Query
REST

Pub/sub
Websocket

KUKSA.val

DBC Feeder

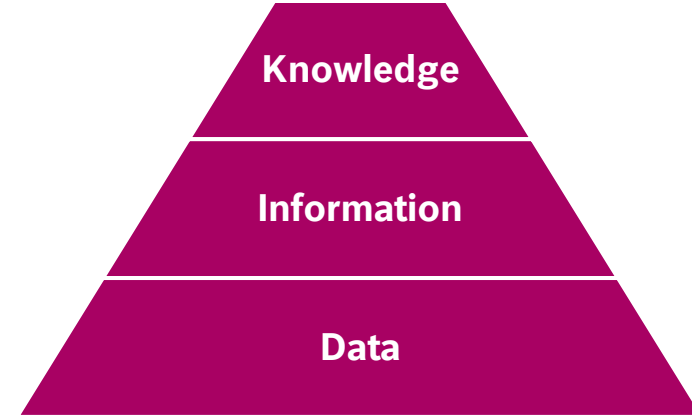
KUKSA Hardware

CAN



```

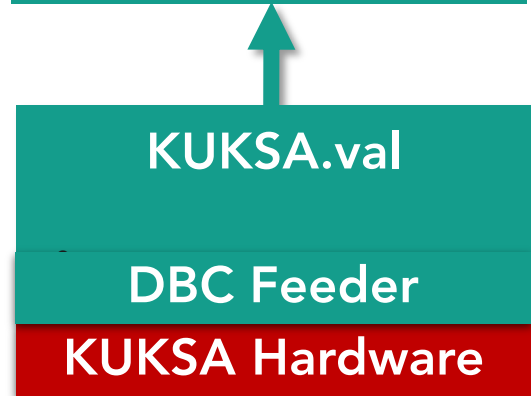
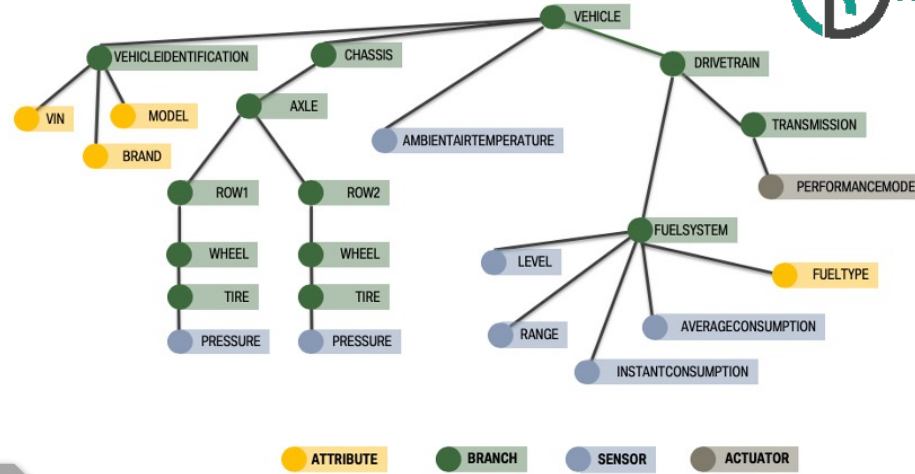
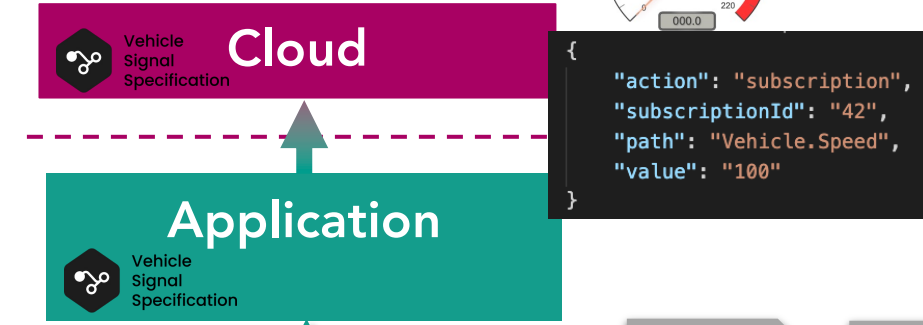
vcan0 442 [8] AF EF 9C 09 52 15 30 43
vcan0 6D5 [8] 08 F5 D8 5F 19 14 3B 57
vcan0 5FB [0]
vcan0 2BA [8] 48 F6 A1 74 92 28 97 38
vcan0 26C [8] AA 0A A0 7E E9 32 AA 67
vcan0 72A [3] CB 1B C7
vcan0 40D [8] 8C 37 6D 7E 39 AB 2E 3C
vcan0 4C0 [8] 1E D5 D7 13 02 91 A6 47
vcan0 135 [8] 54 A6 D6 0A 0A A0 15 49
vcan0 2F6 [8] 24 B4 50 20 F2 70 3F 67
vcan0 26D [8] 5F 57 84 24 27 8D 9D 20
vcan0 714 [7] BA B5 34 59 80 19 CA
vcan0 123 [4] 69 4C 74 2F
vcan0 588 [4] 40 71 EB 74
vcan0 141 [8] 9E 5D 7D 40 01 1E C7 46
vcan0 1F4 [8] 03 AF 6D 0E 2A BB AF 3F
vcan0 789 [8] 34 5B C5 08 80 DA C4 5A
vcan0 0B5 [8] 72 4B 04 42 22 EB E2 5B
vcan0 0A4 [1] 49
vcan0 079 [3] 38 71 14
vcan0 668 [2] EA F1
vcan0 5D6 [8] 55 76 34 0D 18 77 E9 6D
vcan0 3F6 [4] 19 95 80 34
vcan0 680 [8] 15 41 8F 1E 3A FE D0 13
vcan0 175 [8] BA D8 95 6E 2A 8A 56 60
  
```



Kuksa.val System Overview

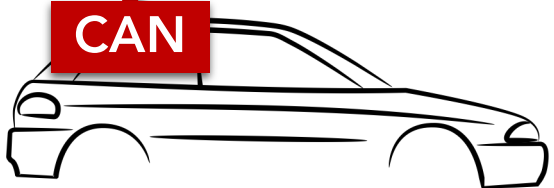
Offboard

Onboard

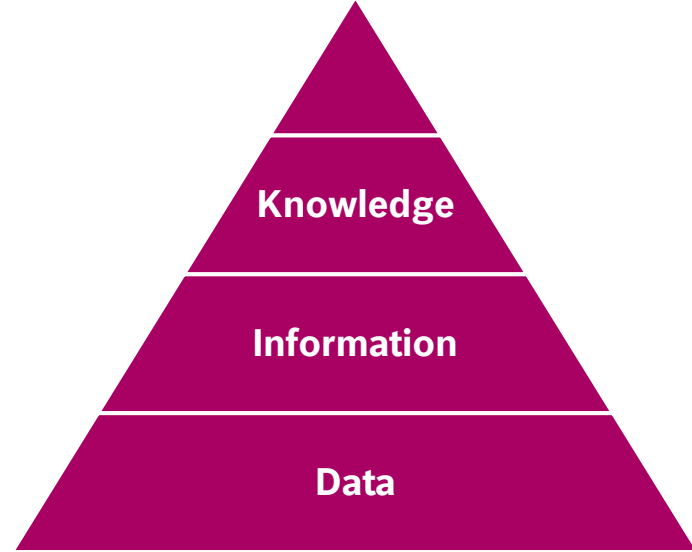


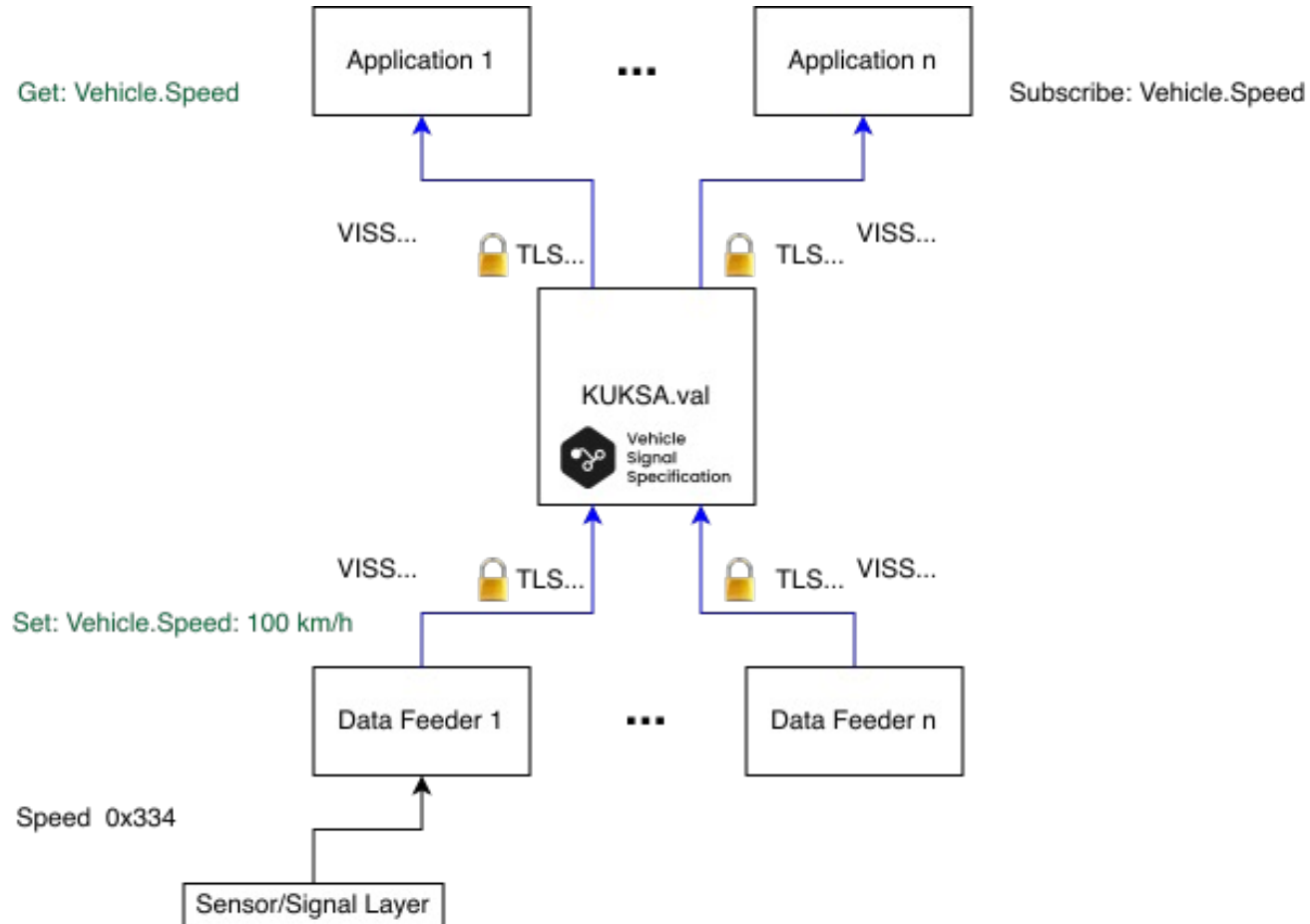
```

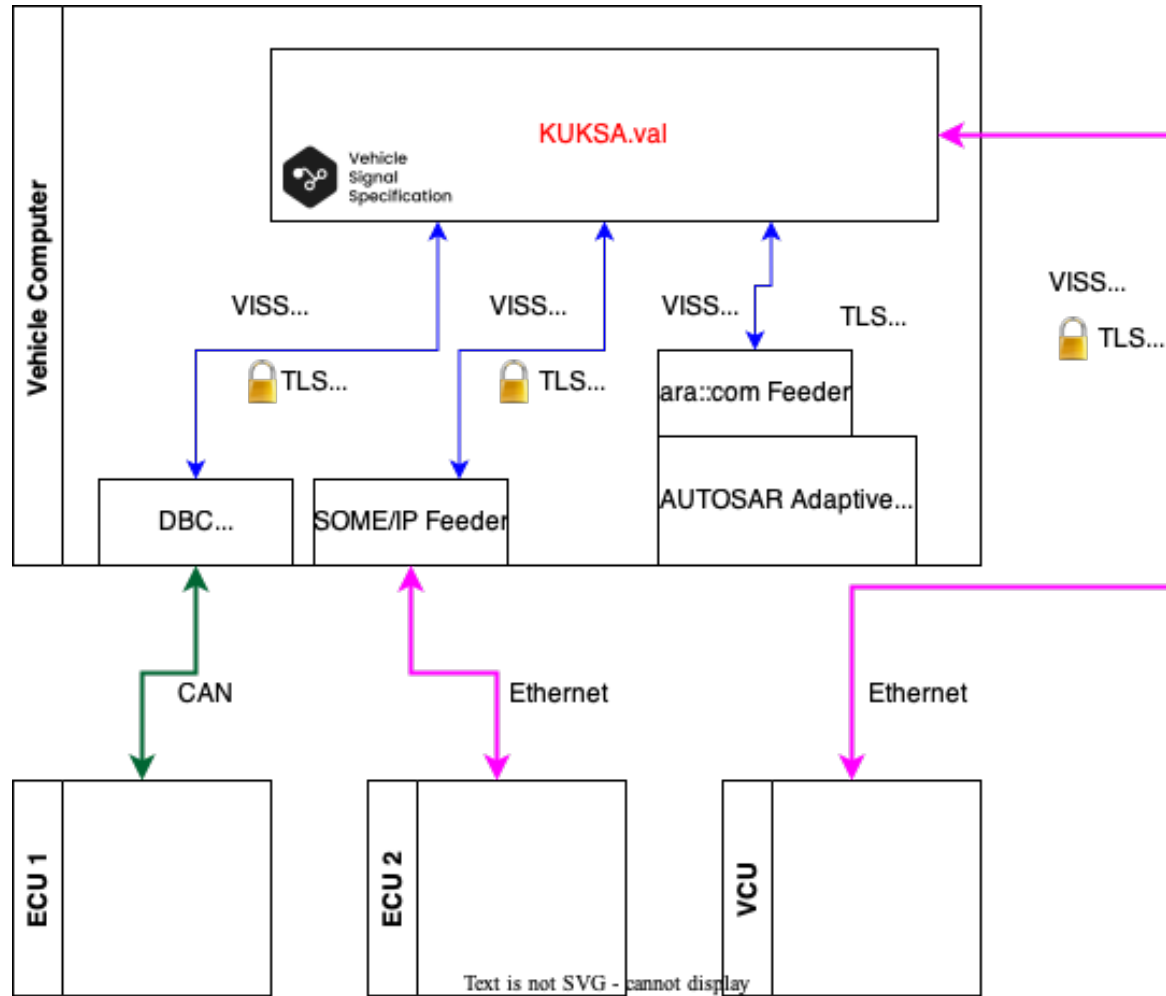
3 UIspeed_signed257:
4   minupdatedelay: 100
5   targets:
6     - Vehicle.Speed
    
```



vcand	442	[8]	AF EF 9C 09 52 15 30 42
vcand	405	[8]	08 F5 D8 5F 19 14 3B 57
vcand	5FB	[8]	
vcand	2BA	[8]	48 F6 A1 74 92 2B 97 3B
vcand	26C	[8]	AA BA A0 7E E9 32 AA 67
vcand	72A	[3]	03 3B C7
vcand	400	[8]	8C 37 60 7E 39 AB 2E 3C
vcand	4C0	[8]	1E 05 07 13 02 91 A6 47
vcand	133	[8]	54 A6 D6 BA BA A0 15 49
vcand	2F6	[8]	24 BA 50 28 F2 70 3F 67
vcand	26D	[8]	5F 57 84 24 27 80 90 20
vcand	714	[7]	BA B5 34 59 80 19 CA
vcand	123	[4]	69 4C 74 2F
vcand	588	[4]	4B 71 EB 74
vcand	141	[8]	9E 5D 70 40 01 1E C7 46
vcand	314	[8]	03 AF 6D 0E 2A BB 3F 3F
vcand	789	[8]	34 5B C5 08 89 DA C4 5A
vcand	085	[8]	72 4B 04 42 22 ED E2 5B
vcand	0A4	[1]	49
vcand	079	[3]	38 71 14
vcand	668	[2]	6A F2
vcand	5D6	[8]	55 76 34 0D 18 77 E9 6D
vcand	3F6	[4]	19 95 80 3A
vcand	680	[8]	15 41 8F 1E 3A FE D0 13
vcand	175	[8]	8A D8 95 0E 2A 6A 56 60



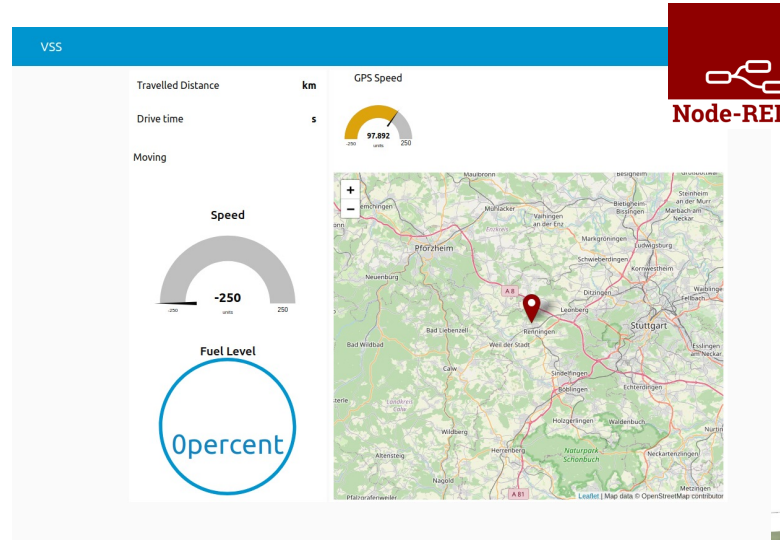




Text is not SVG - cannot display

Kuksa.val Features

- Kuksa.Val Server: providing access to VSS data using W3C VISS websockets or gRPC (written in C++)
- Kuksa.Val Databroker (in Rust) for accessing VSS data using gRPC
- Python & Go SDK
- Interactive test client (CLI)
- Feeders (DBC, GPS, Replay)
- Examples
- Docker images for arm64 and x86_64



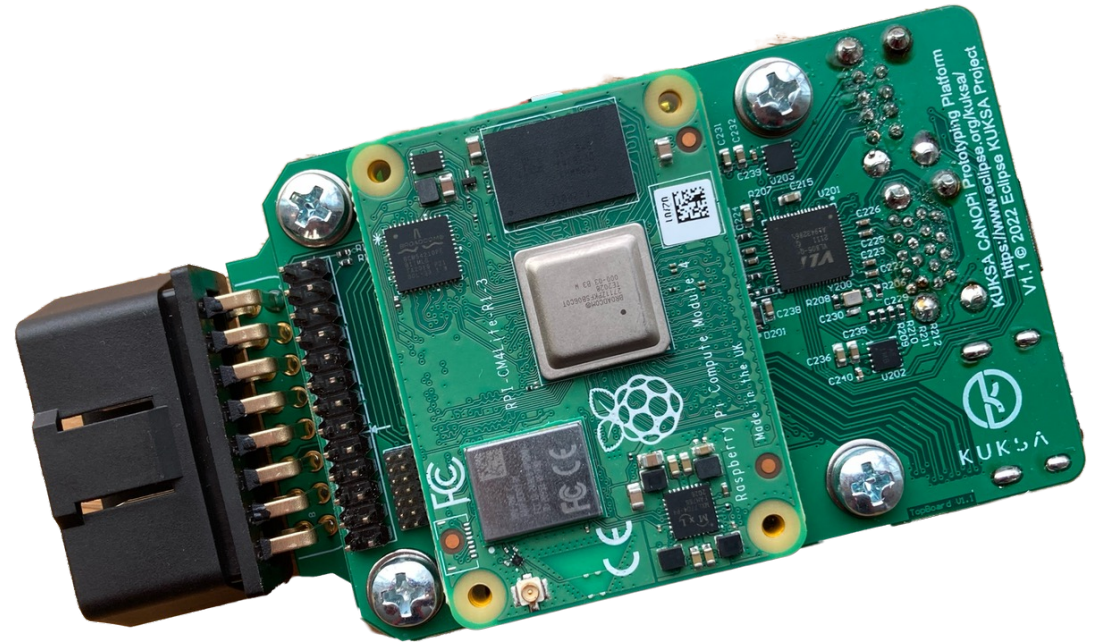
```
Welcome to fish, the friendly interactive shell
```

```
> █
```


Kuksa.hardware

Open Hardware for Kuksa

- Baseboard for Raspberry Pi Compute Module 4
- Schematics are Open Source
- Access vehicle data on OBD (only CAN supported)



Updates

Upcoming topics

- Cloud:
 - Kuksa.Cloud @ Home
- Val:
 - Unified gRPC API for relevant components like Val-Server, Databroker, Python SDK, Go SDK
 - All feeders work with val-server and databroker
 - Databroker differentiates between between current and target values for actuators
 - Python SDK automatically released in PyPi

Thank You!

Stay in contact:

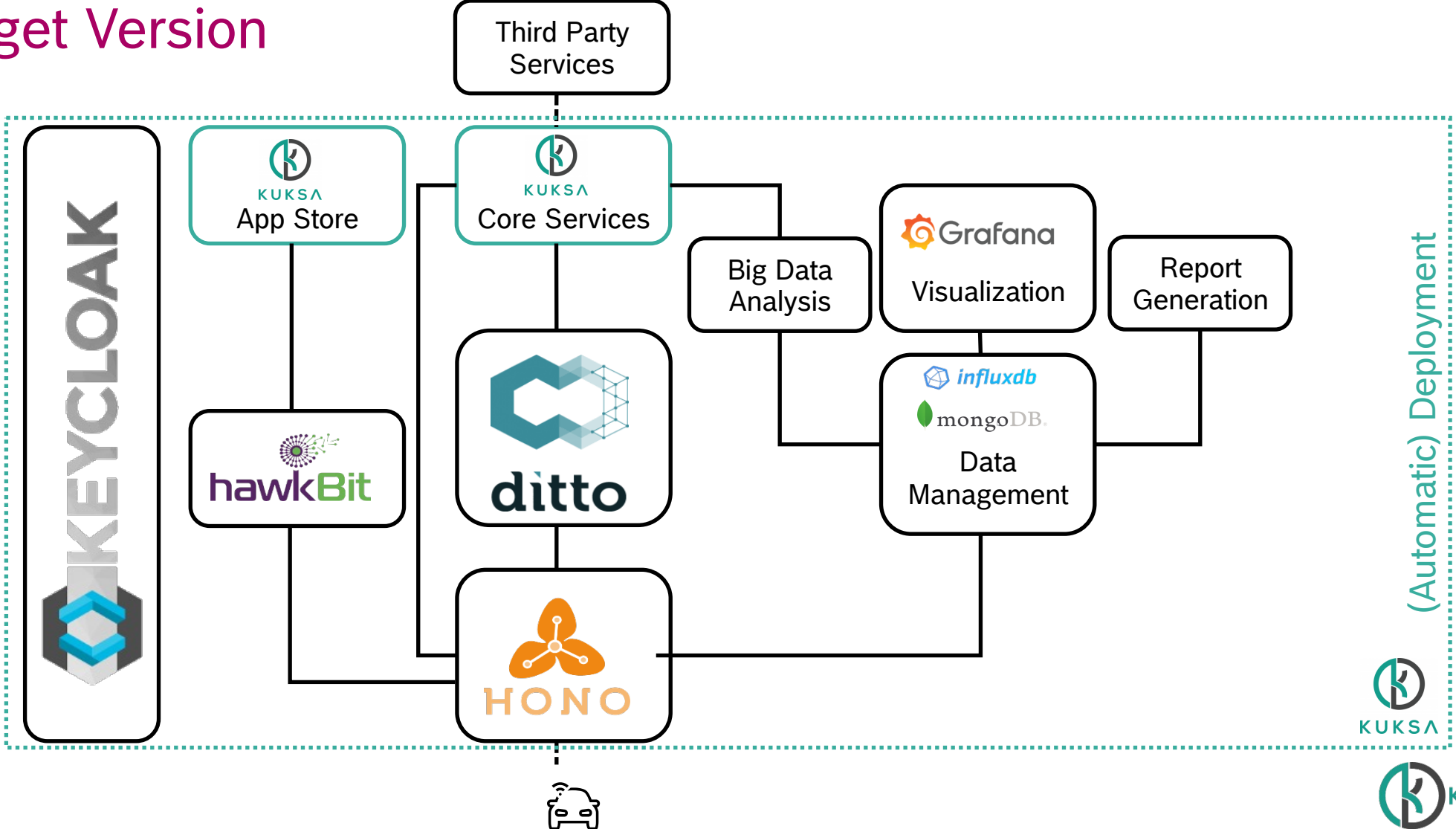
github.com/eclipse/kuksa.val
eclipse.org/kuksa

svenerik.jeroschewski@bosch.io

Back Up

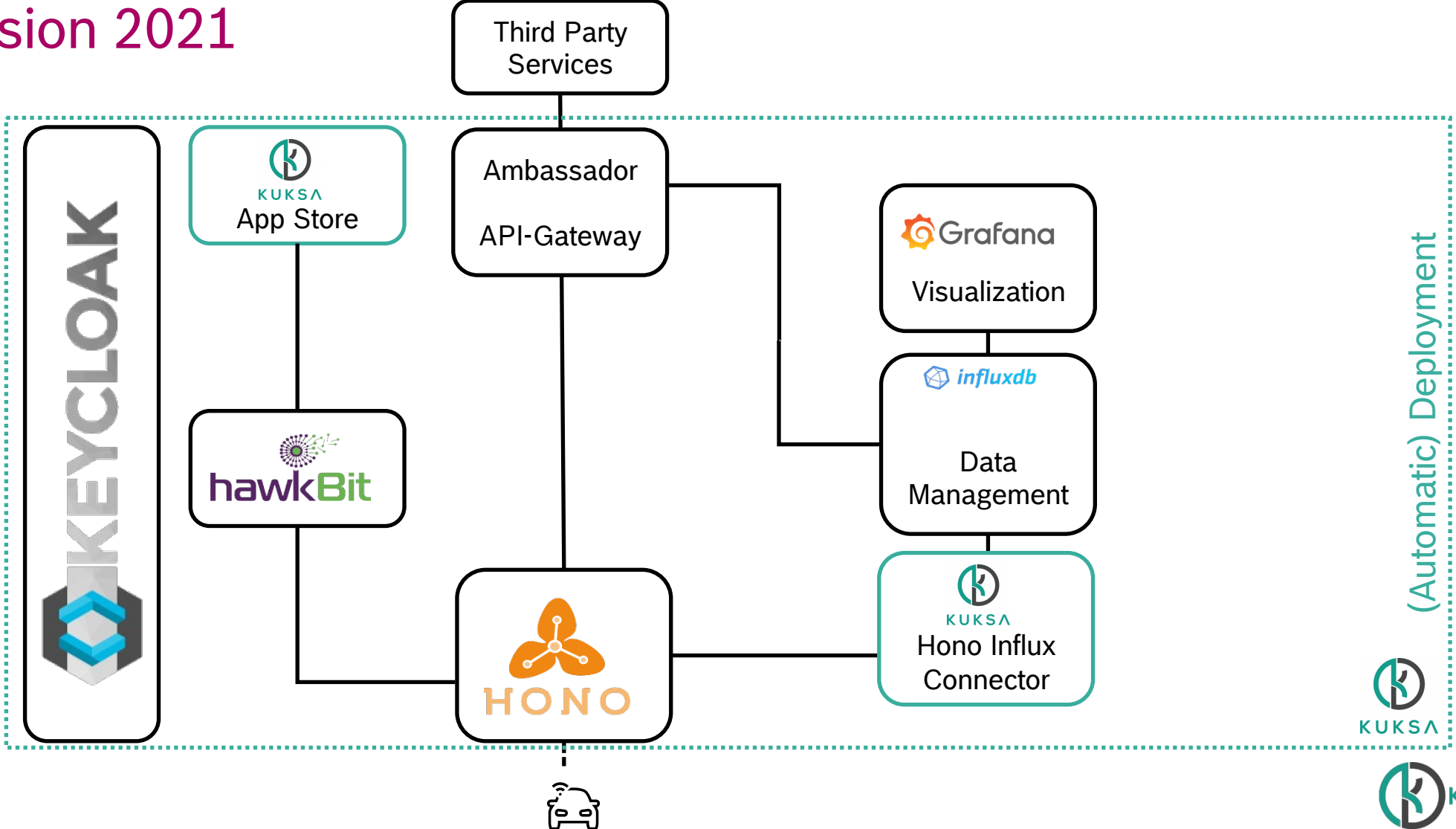
Eclipse Kuksa Cloud Backend

Target Version



Eclipse Kuksa Cloud Backend

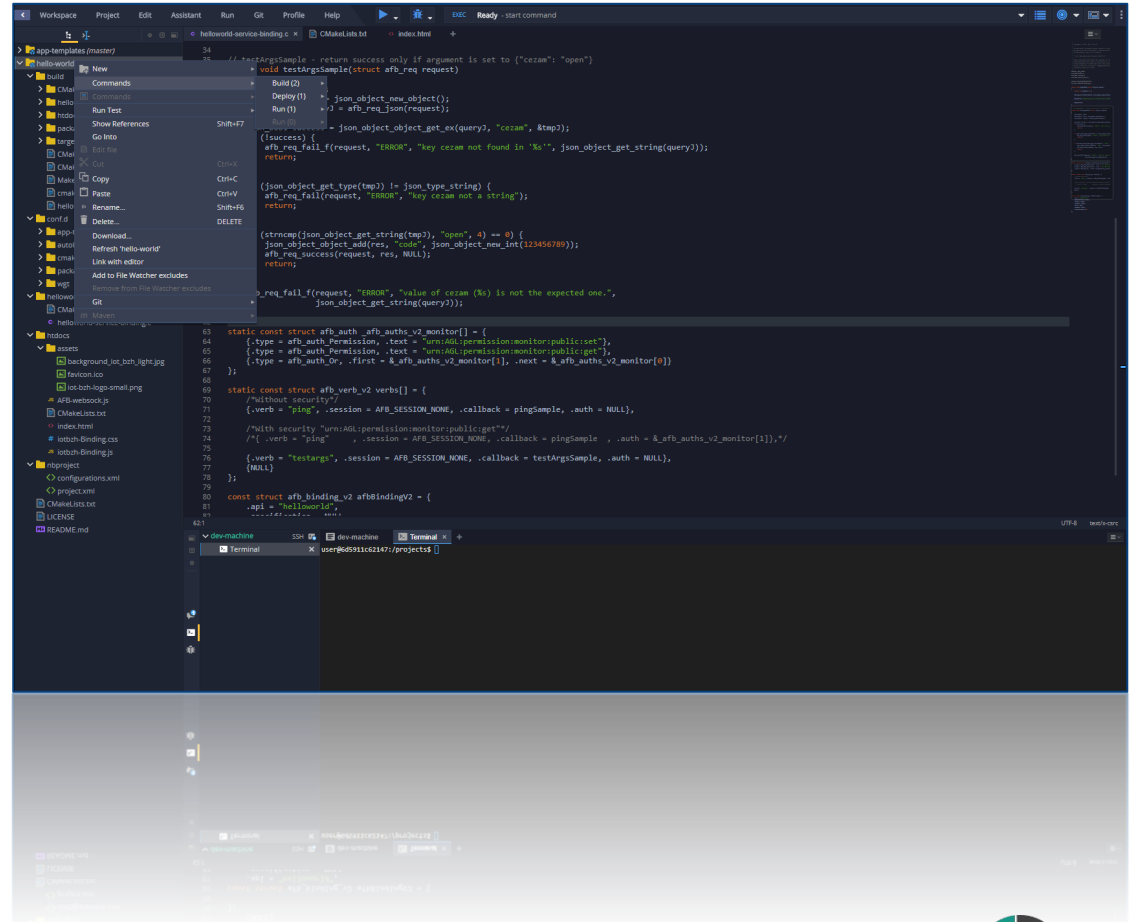
Version 2021



Eclipse Kuksa

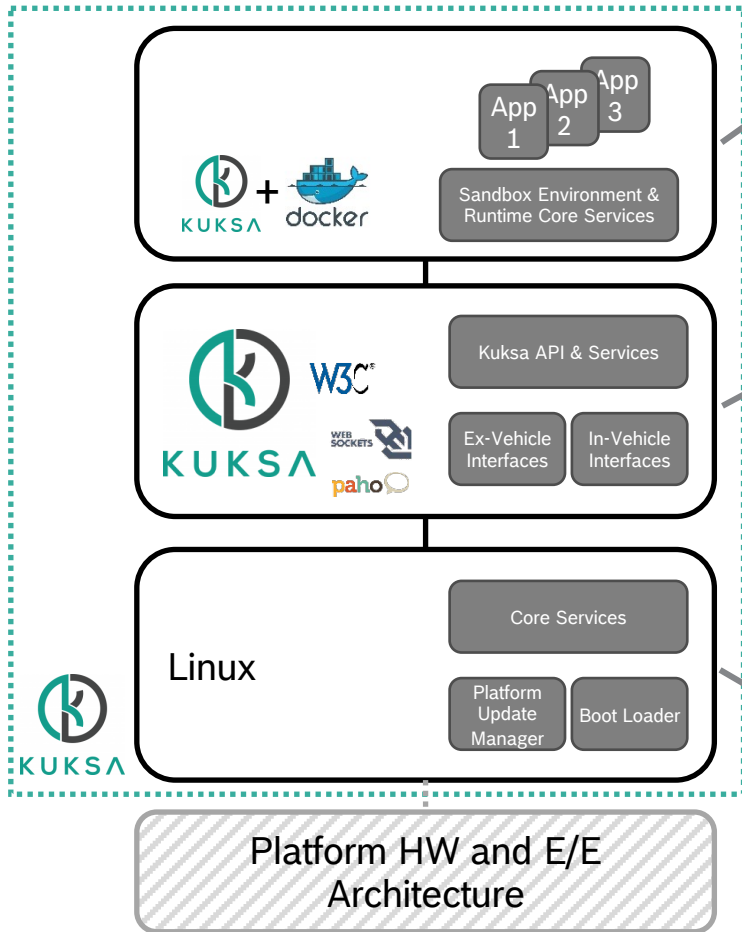
Kuksa IDE

- ▶ Based on Eclipse Che
- ▶ Allows Cloud and In-Vehicle Application development
- ▶ Platform independent
- ▶ Shared workspaces
- ▶ Almost configuration free
- ▶ Docker-based: VPN planned to allow remote / network independent cross compilation



Eclipse Kuksa

Kuksa In-Vehicle Platform



Application layer:

- Runs 3rd party apps on the platform
- Contains a Sandbox Environment & Additional Services

Middleware layer (Yocto layer):

- APIs to abstract the vehicles' E/E architecture (W3C VISS, Sensoris...)
- Communication Services to manage network access and provide data from the vehicle
- Includes communication libs, protocols, security layers,...

OS layer:

- Reuse of OE's existing services, layers, HW abstractions, AGL services, etc.