



# SDV: Eclipse Kuksa

What is it? Compliance? What is up next?

Sebastian Schildt

Openchain Automotive Workshop, Sept. 2024, Stuttgart

# whoami



Name	Sebastian Schildt
Employer	<b>ETAS</b>
Job by day	Something with SDV
.. and beyond	SDV Vehicle Abstraction, OSS evangelist, COVESA and Eclipse contributor



COPYRIGHT (C) 2024, ECLIPSE FOUNDATION. | THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)

# SDV - Software Defined Vehicle

... is many things to different people

CI/CD  
Dev processes, agile  
SDN

Cloud native

Containers



Software Updates  
HW&SW Decoupling  
Shift-to-QM  
Interfaces

*more efficient*

Becoming 10 times *Faster* developing automotive software  
*cheaper*

## Many variants in Automotive platforms

**Virtually no common interface semantics**

**10 times faster does not help you if you  
need to do 100 times more!**



### SDV warning

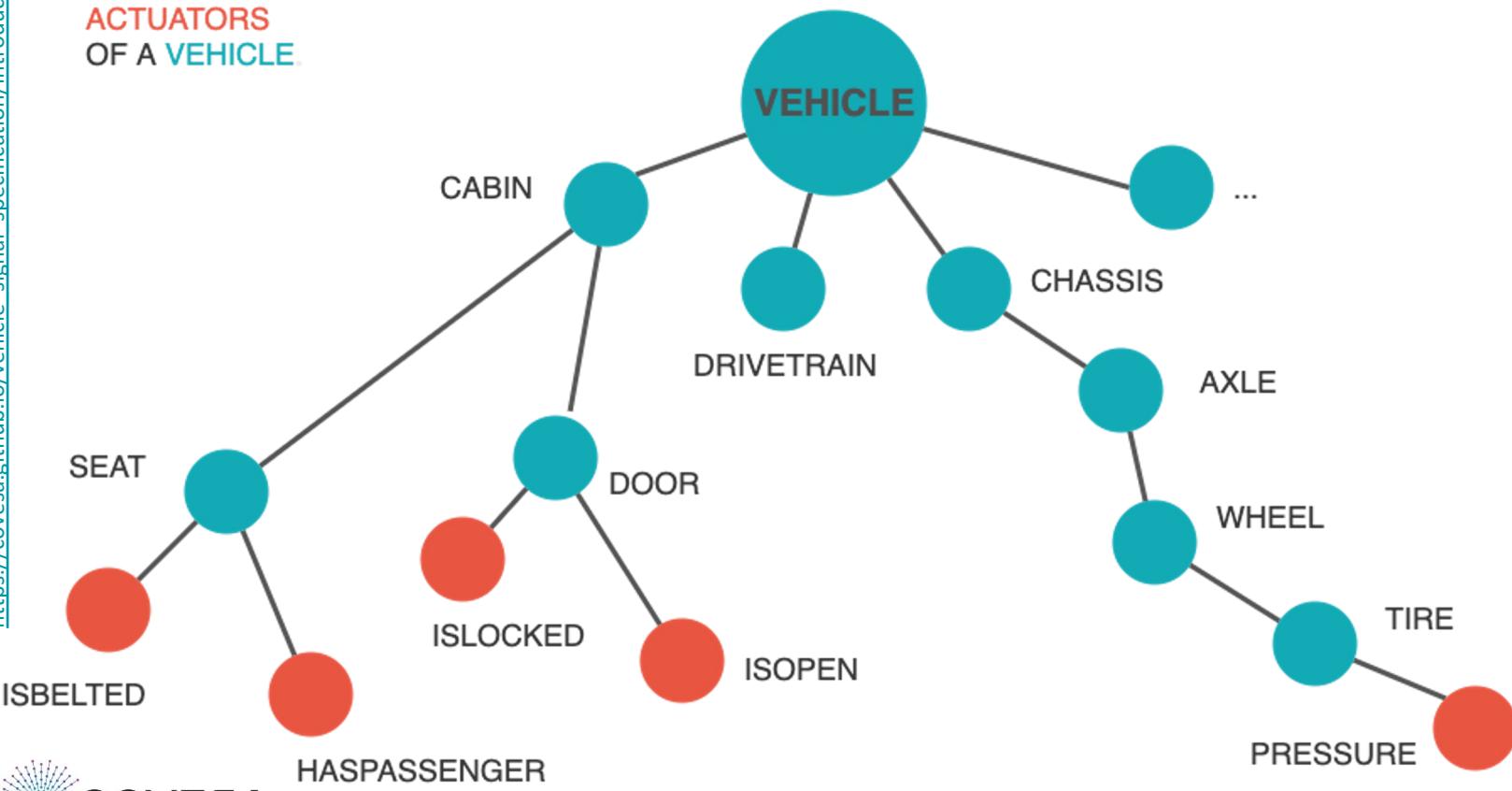
Without common interfaces to vehicle functions and data,  
software reuse and interoperability are impossible

# COVESA Vehicle Signal Specification (VSS)

TAXONOMY  
FOR  
ATTRIBUTES,  
SENSORS AND  
ACTUATORS  
OF A VEHICLE



[https://covesa.github.io/vehicle\\_signal\\_specification/introduction/overview/](https://covesa.github.io/vehicle_signal_specification/introduction/overview/)



Vehicle.Drivetrain.Transmission.Speed  
type: sensor  
datatype: float  
unit: km/h  
description: The vehicle speed as measured by the drivetrain

## YAML SPECIFICATION

- Simple, flexible and protocol agnostic common approach for describing vehicle data
- Extensible data model & catalog with industry supported tooling.

# digital.auto VSS browser

playground.digital.auto Search ACME Car (EV) v0.1 Vehicle APIs Prototypes Sign in

COVESA VSS ▾

Search Filter

Recently viewed APIs

Vehicle	BRANCH
Vehicle.ADAS	BRANCH
Vehicle.ADAS.ABS	BRANCH
Vehicle.ADAS.ABS.IsEnabled	ACTUATOR
Vehicle.ADAS.ABS.IsEngaged	SENSOR
Vehicle.ADAS.ABS.IsError	SENSOR
Vehicle.ADAS.ActiveAutonomyLevel	SENSOR
Vehicle.ADAS.CruiseControl	BRANCH
Vehicle.ADAS.CruiseControl.IsActive	ACTUATOR
Vehicle.ADAS.CruiseControl.IsEnabled	ACTUATOR
Vehicle.ADAS.CruiseControl.IsError	SENSOR
Vehicle.ADAS.CruiseControl.SpeedSet	ACTUATOR
Vehicle.ADAS.EBA	BRANCH
Vehicle.ADAS.EBA.IsEnabled	ACTUATOR
Vehicle.ADAS.EBA.IsEngaged	SENSOR
Vehicle.ADAS.EBA.IsError	SENSOR

List View Tree View



Vehicle.Cabin.Door Discussion (0) BRANCH

Tags

VSS Specification

Description All doors, including windows and switches.

Type branch

UUID fd7f4d16f8965419a9a69fd66b40c1d7

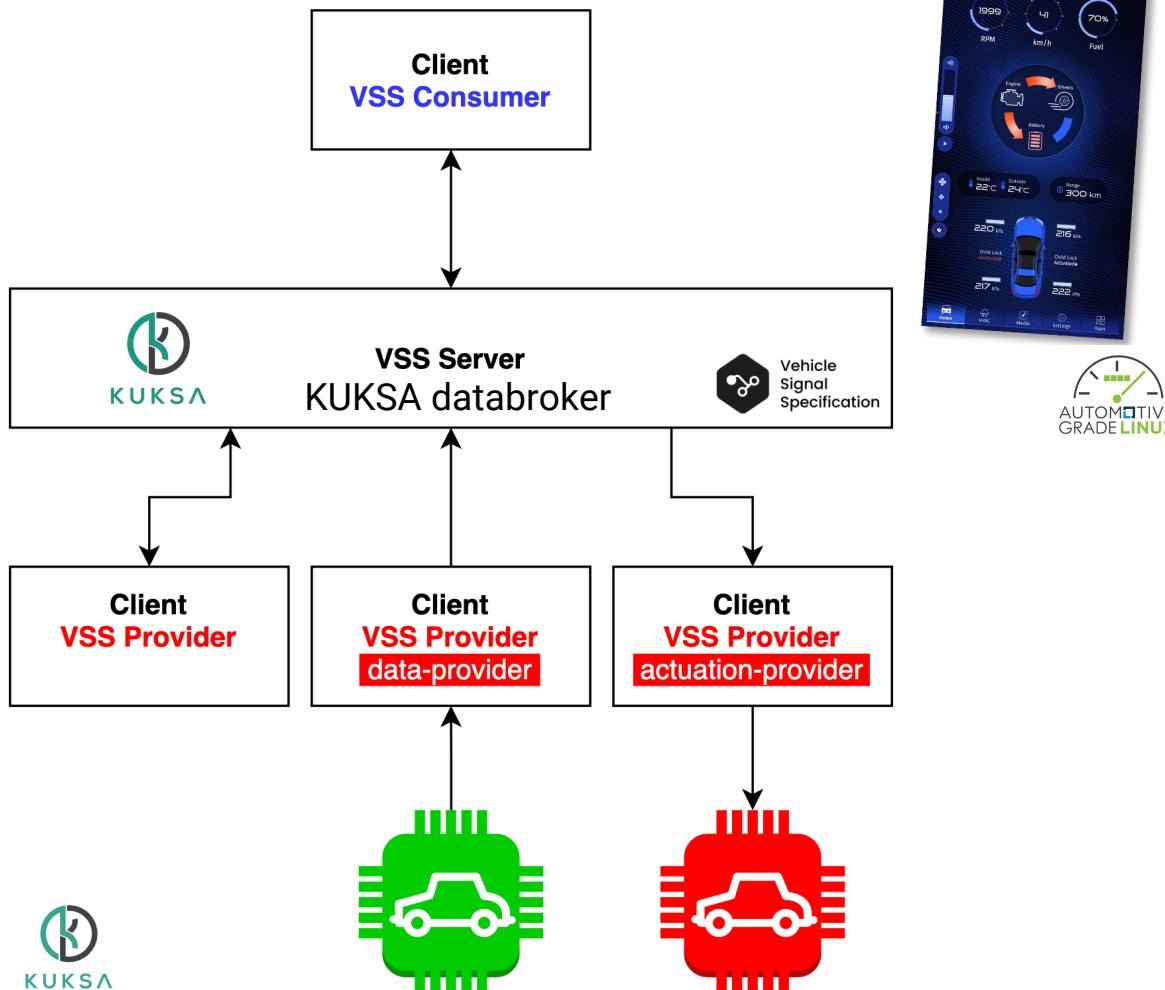
Source [https://github.com/COVESA/vehicle\\_signal\\_specification](https://github.com/COVESA/vehicle_signal_specification)

Report

For a first "feel" of the VSS model you can try the VSS browser in the digital.auto exploration platform for vehicle applications

<https://digitalauto.netlify.app/>

# Taxonomy of in-vehicle VSS components



- Interacts with Vehicle represented by the VSS model
  - Vehicle Computer function
  - IVI App
  - External consumer device
- Holds current vehicle state in VSS format
- Provides an API to interact with VSS signals
- VSS provider syncs of the vehicle with VSS model of the server
  - **data-provider** makes sure that the actual state of a vehicle is represented in VSS (historically known as “feeder”)
  - **actuation-provider** ensure that the target value of a VSS actuator is reflected by the actual state of a vehicle

## Modern platform

- KUKSA databroker (the VSS server) is written in 100% Rust

## Efficient enough for today's vehicle computers

- The KUKSA databroker container is <10 MiB in size, and has been run on platforms with less than 64 MiB RAM

## Language Agnostic

- Main Interface based on GRPC: Can be used with virtually any language/platform

## Options & Batteries included

- Contains various example providers, e.g. CAN, SOMEIP, GPS,..
- Fine grained authorization concept on signal level
- Basic support for VISS V2 websocket protocol
- ...

# KUKSA & Compliance

## Can we even OSS??

2018

KUKSA was one of the earlier automotive OSS approvals at Bosch



Became Eclipse foundation project, due to experience from our SW subsidiary doing IoT OSS there

Bosch view\* at the time

Common OSS



\*only slightly simplified

# KUKSA & Compliance

## Relevant Events



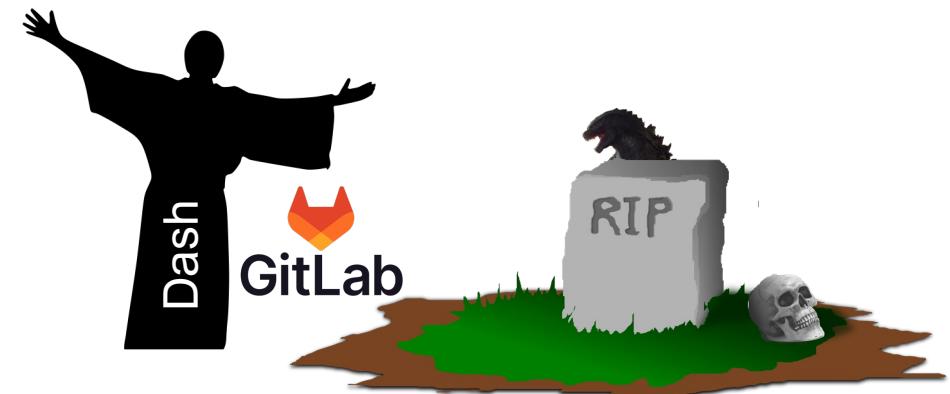
Migrate to Rust



Eclipse CQ process



Eclipse Tooling & Process Improvements



License compliance “for our own sake”

2023

### Our goals here as OSS project

- Compliance, Supply Chain challenges manifest themselves on an integration/product level
- As OSS “component” we can not solve them
- Not in cope
- Topic is gaining more importance (e.g. CRA discussion)
- **Be a good citizen, make it PAINLESS for our integrators users**



CycloneDX

SPDX

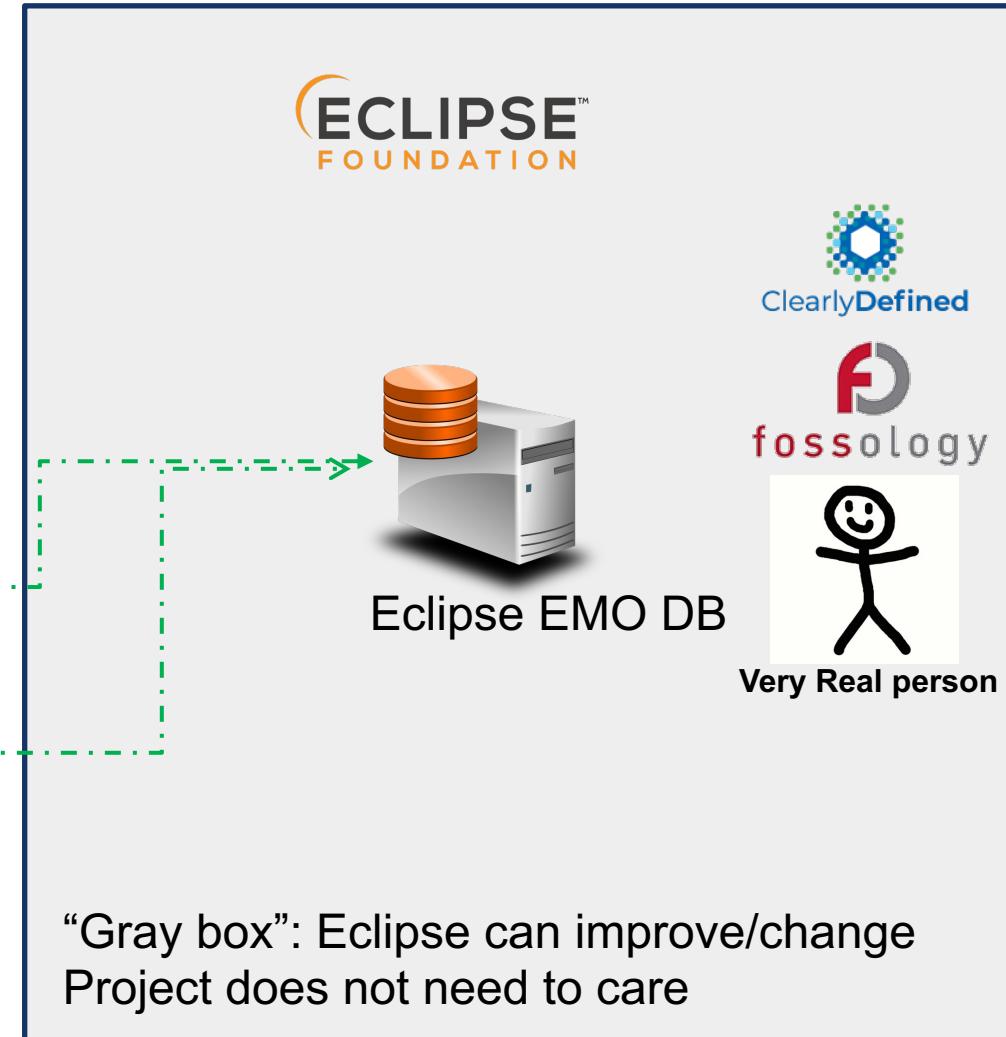
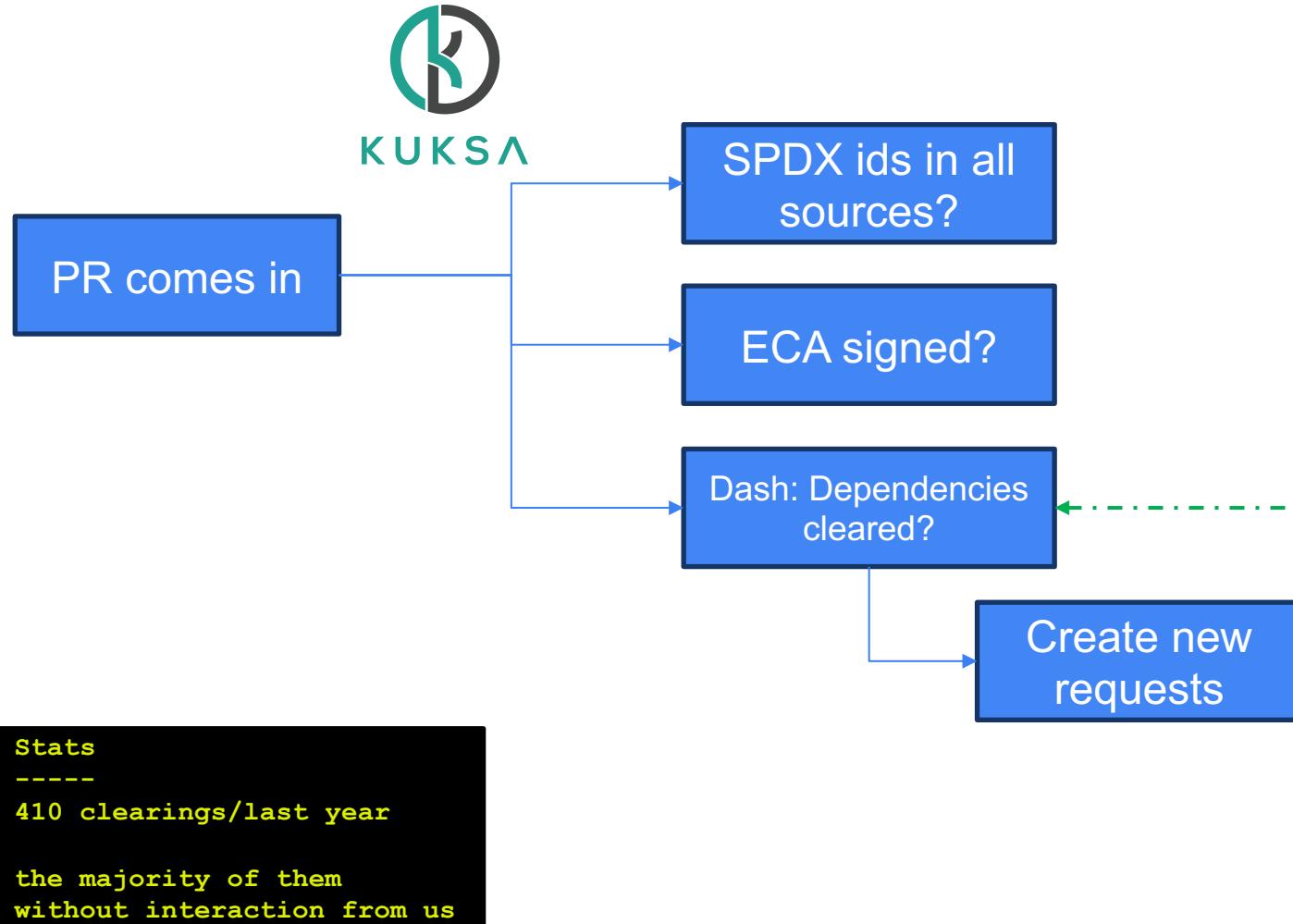
Protect  
our asses

License compliance “for  
our own sake”

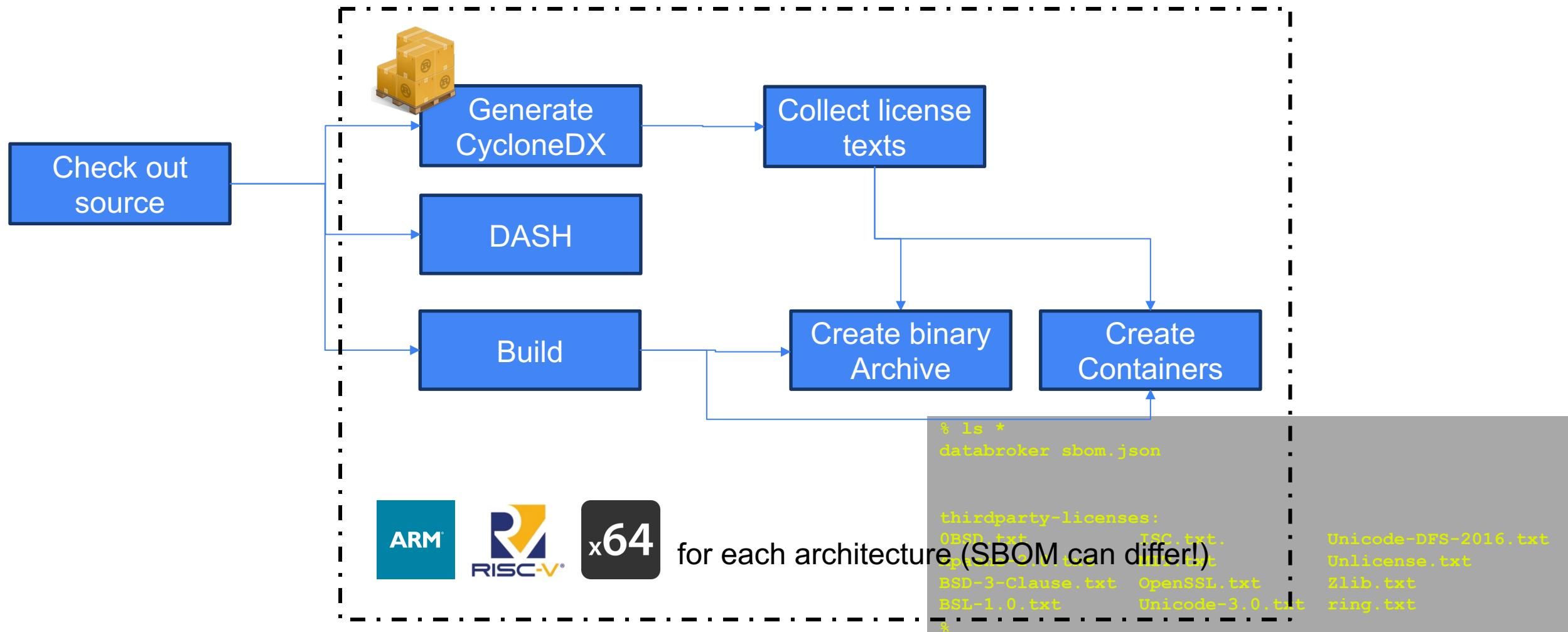
It's a  
feature

License compliance/BOM  
transparency “for  
downstreams sake”

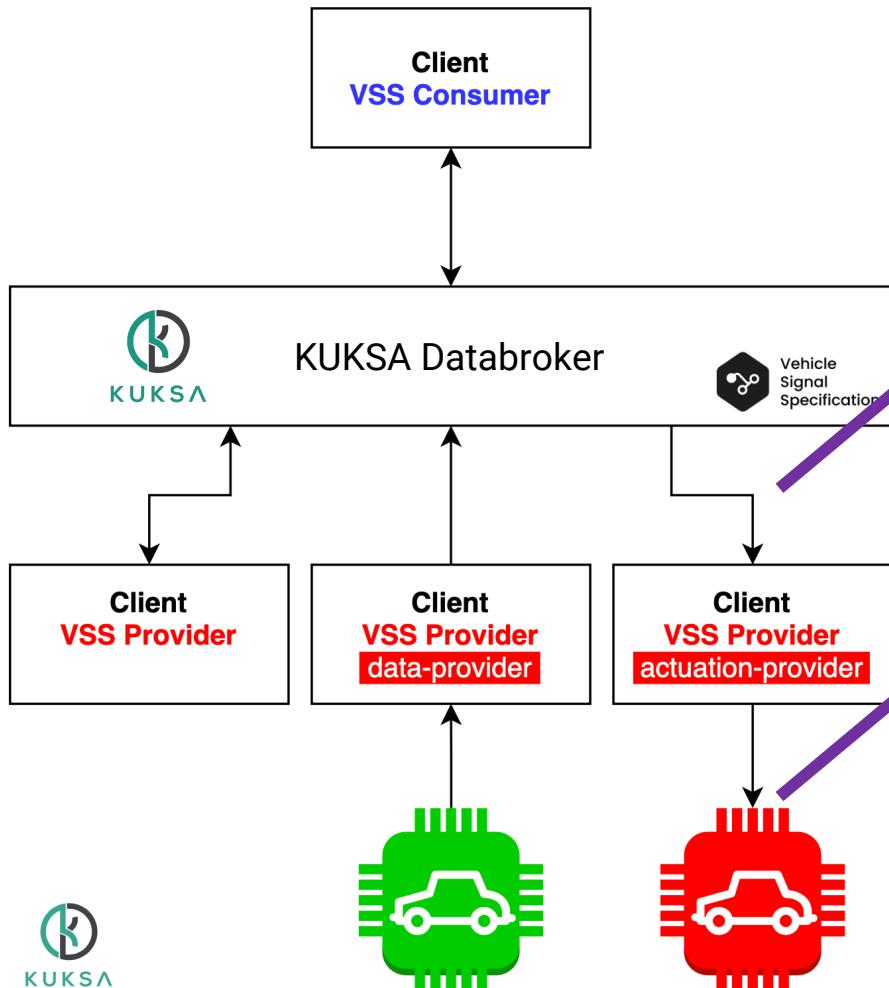
# Current process: Adding new code



# Current process: Building (release) artifacts



# What Is Next?



## API:

Performance improvements, and awareness which signals are currently served by providers (operational status of providers)

## IEEE 1722

Cooperating with Open1722 project, bringing this high performance Ethernet transport into the KUKSA / VSS world



# THANK YOU!

KUKSA



<https://eclipse.github.io/kuksa.website/>

/me



<http://sdv.expert>

COVEZA VSS



[https://covesa.github.io/vehicle\\_signal\\_specification/](https://covesa.github.io/vehicle_signal_specification/)

Eclipse SDV



<https://sdv.eclipse.org>

ETAS OSS



<https://www.etas.com/en/open-source-software.php>