Software Defined Vehicle
Our **Golden Rules**

- We are **code** first (but not code only)
- Projects are **self-governing** to enable community driven success.
- Aim for success by selecting **robust, sustainable, well engineered projects**.
- **No kingmakers.** Similar or competitive projects are not excluded for reasons of overlap or competition.
- **Don’t reinvent the wheel.** Adopt and promote useful work happening in other industry groups.
- **No single stack.** Encourage interoperability for the emergence of a variety of stacks and distributions to serve the community and adopters.
- Define specifications derived from **open source projects** which have demonstrated broad adoption and real world use.
- **We want to build** automotive grade software (from QM to ASIL-D) which will be used in series production.
- **We are here to help our projects.**

Source: SDV [Charter](#)
## SDV - A short history (15+ busy months)

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 22nd</td>
<td>Announced at EclipseCon 2021</td>
</tr>
<tr>
<td>Feb 1st</td>
<td>Charter approved</td>
</tr>
<tr>
<td>March 3rd</td>
<td>First Steering Committee Meeting</td>
</tr>
<tr>
<td>June 30th</td>
<td>1st SDV Contribution Day</td>
</tr>
<tr>
<td>Sept. 22nd</td>
<td>2nd SDV Contribution Day</td>
</tr>
<tr>
<td>Oct 27th</td>
<td>1st SDV Hackathon</td>
</tr>
<tr>
<td>Nov 7th - 9th</td>
<td>SDV Hackathon at BCX</td>
</tr>
<tr>
<td>March 29th - 30th</td>
<td>1st Community Day</td>
</tr>
<tr>
<td>June 6th</td>
<td>Automotive Open Source Summit</td>
</tr>
</tbody>
</table>

**31.3.2022**  
**16 members**
SDV Membership Commitments

38 members today (and growing)
The SDV WG Scope

**SDV.Edge**
- AD/Domain Functions
- (Containerized) Apps
- Middleware
- OS

**SDV.Ops**
- Business Domain Services
- Mobility Solutions: Eg. Software Management, Data Services
- Cloud Infrastructure

**SDV.Dev**
Toolchains and workflows development, applications integration and management
The SDV project landscape (from 0 one year ago)
SDV Projects with Contributions & Releases

**SDV.Edge: Vehicle**
- **Chariott**: Metadata-driven middleware/abstraction HW layer, gRPC based framework
- **eCAL**: Interprocess communication using a publish/subscribe pattern.
- **Ibeji**: Digital twin of vehicle state representation
- **p3com**: Transport agnostic, layered communication protocol from mechatronic layer to Cloud.

**SDV.Ops: Cloud**
- **KUKSA**: Vehicle Interfaces adaptation into a basic interfaces using simple APIs (eg VSS).
- **ROS**: adaptive framework and runtime platform

**SDV.Dev**
- **muto**: Quickstart linux based image for "SDV distribution"
- **Leda**: Development toolchain for in-vehicle applications.
- **Velocitas**: Application: Showcase new lighting concepts

Source: SDV Projects
What use cases can you solve today?

**Driving Score**
- Measure the quality of driving by calculating a score of the driving style
- Implement a vehicle app to calculate the score and visualize it

**Passenger Welcome**
- The goal is for a passenger to feel comfortable entering and using a vehicle
- Ideate on ideas how the vehicle should behave

**Hack the Truck**
- Do you dream big? Yes?
- Have you ever hacked an actual truck? No?
- Well, here you can! We bring the truck - you bring your creative ideas

**Virtual Vehicle Application**
- build in-vehicle applications using telemetry data generated with simulation tools
- the two simulation tools used are Carla and AirSim

**Control Vehicle Lights**
- Control vehicle lights using Eclipse SommR and KUKSA.val
- Integrate lights with other hack challenges

**Let’s Play OSM and CARLA**
- Bring real world city and street data into the simulator’s virtual world
- Implement best way of driving in this world from A to B and visualize it

Source: https://eclipse-sdv-hackathon-bcx.github.io/github/
Don’t reinvent the wheel. The ecosystem around...
Why “SDV” may be the wrong name for the Working Group
The state of Open Source

- **VALUE APPROPRIATED**
  - 0 DENY
  - 1 USE
  - 2 CONTRIBUTE
  - 3 CHAMPION

- **TIME**
  - SINGLE PROJECT
  - SCOPE
  - MULTIPLE PROJECT

- **ENGINEERING DRIVEN**
- **BUSINESS DRIVEN**

- **VALUE CO-CREATION**
- **VALUE APPROPRIATION**
- **COUPLING MANAGEMENT**
Why **Open Source** may not be enough

**Open Collaboration**

- **Contribution**
  - Transfer of IP rights
  - Future influence

- **Collaboration**
  - Rules for collaboration (Governance)
  - Predictability
  - Sustainability

**Open Source**

- **Adoption**
  - Open Source License
  - Four Freedoms
Our Impact: Open Innovation at Industrial Scale

**Governance Layer**
The Eclipse Foundation provides an open, vendor-neutral development environment to enable collaboration.

**Collaboration Layer**
Technology Producers jointly define roadmap and build core capabilities.

**Competition Layer**
Commercial Adopters focus resources on rapidly building differentiating features.

€13+ billion of shared investment to date
Let’s talk about the Collaboration Layer (and the Value Line)

- Least common denominator
- Jointly solving today's challenges
- Think big for the future
Let’s talk about the Collaboration Layer (and the Value Line)

Building blocks
A project is like a piece of the cake
That’s what SDV looks like today
Let’s talk about the Collaboration Layer (and the Value Line)

Middleware reloaded
SDV as middleware reloaded?
An Open Source middleware stack

- Leveraging existing technologies
- Easy and fast extensible
- Smart concepts (e.g. containers) for SW updates
- In a limited number of variants (SDV distributions)
- Includes needed tooling for configuration, testing and validation (and more)

This would need more of your cakes and developers
Let’s talk about the Collaboration Layer (and the Value Line)

Platform
Enabling the **platform economy** with open source

Source: Dr. Holger Schmidt
SDV as a platform?

Google Play

- **Play In-app Billing Library**
  - Provides APIs to help you implement Google Play's in-app billing and subscription features.
  - Reference: User Guide

- **Play Core Library**
  - Provides APIs to help you request, monitor, and manage on-demand downloads for Play Feature Delivery, Play Asset Delivery, and offers additional APIs such as in-app updates and in-app reviews. This library is available in Java, Native code, and Unity.

- **Play Install Referrer library**
  - Provides APIs to securely retrieve referral content from Google Play.
  - Reference
We understand!

We, the community, need to **build trust first** (before you may want to go “all-in”)

- By showing that we **can deliver** value
- By demonstrating **thought leadership**
- By building the network with other initiatives

But keep in mind, you may be seen as part of “the community”