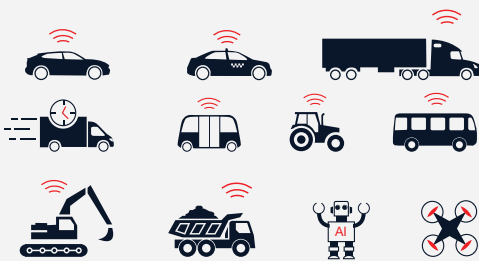


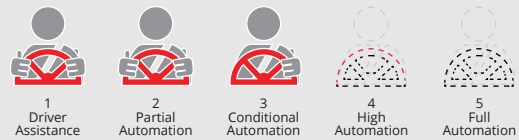
# Apex.OS

The vehicle OS.

## All Applications



## All Levels of Automation



## All Phases

Prototyping  
Product Development  
Deployment

## All Components

Sensors                      Perception  
ECUs                            Localization  
Driver Assistance          Planning  
Safety Systems              Teleop

**Apex.OS** → It can be used as a functionally safe plug-in-replacement for ROS.

→ OEMs and Tier 1s can integrate white labeled Apex.OS into their operating systems.

# Why Apex.OS



## Situation

- Software frameworks enable modern modular software development.
- **ROS** (Robot Operating System) is the pre-dominantly used software framework for robotics and autonomous driving.
- **ROS is open source** with open APIs.
- **ROS is great for prototyping.**



## Problem

- ROS software **does not run in realtime.**
- ROS is **not sufficient for use in safety-critical products.**
- ROS-based software is **not certifiable.**



## Solution

We take **ROS** from an open-source framework to a commercial, supported, and certified product based on open APIs: **Apex.OS**



## Benefits Apex.OS

Prototyping in ROS can directly transition to product development

- + **Simplified workflows**
- + **Much faster and cheaper to market**

ROS APIs and toolchain can be reused

- + **Employees knowing ROS do not need to be retrained**
- + **Faster and cheaper to market**
- + **Easier to find qualified employees**
- + **Makes you more attractive to candidates**

ROS-based code can be certified

- + **Much faster and cheaper to safety-certified products**



## Execution

Abstraction of underlying hardware, middleware, RTOS, and dependencies into simple-to-use APIs

- + **Real-time and embedded applications can be developed much easier**
- + **Addresses shortage of experts; easier to find suitable employees**
- + **Removes hardware, middleware, and RTOS vendor lock-in**