

TRATON ONE OS

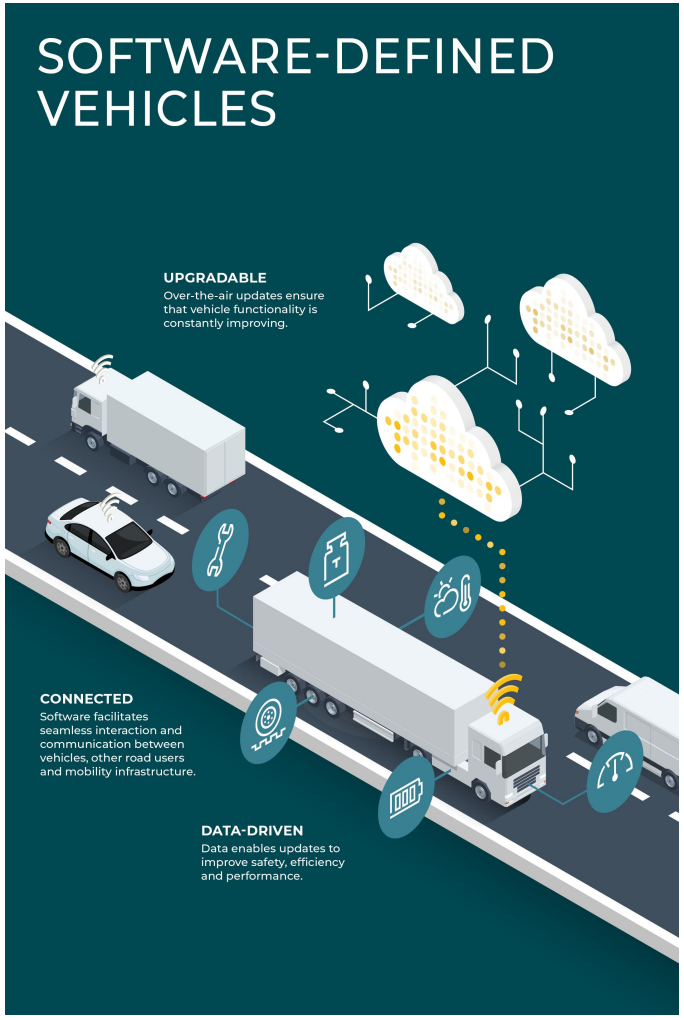
and its relation to Eclipse projects

Eclipse SDV Community Meetup, Sunnyvale, CA, 5 November 2026

Janakiram Sistla, TRATON Group R&D

WHY TRUCKS CHANGE THE SDV EQUATION

Fleet economics set the software



Downtime has an operating cost

Staged updates and safe rollback must fit fleet schedules and protect vehicle availability.

Vehicles outlast software cycles

Security support and hardware compatibility must continue through years of service.

Every mission adds configuration

Powertrains and bodybuilder equipment demand modular software with stable interfaces.

Value extends beyond the vehicle

Vehicle data must connect to dispatch, maintenance and charging workflows.

The SDV measure of success: more productive hours and lower cost per kilometer

TRATON ONE OS AT A GLANCE

1

A shared software foundation

TRATON ONE OS brings in-house, partner and open-source components together on central compute to deliver vehicle functions.

2

A common API across workloads

Real-time, service-oriented and cloud workloads use a common TRATON ONE OS API, with execution suited to each function.

3

Software support throughout vehicle life

Full-system, partial and individual application updates, together with configuration changes, support vehicles beyond the start of production.

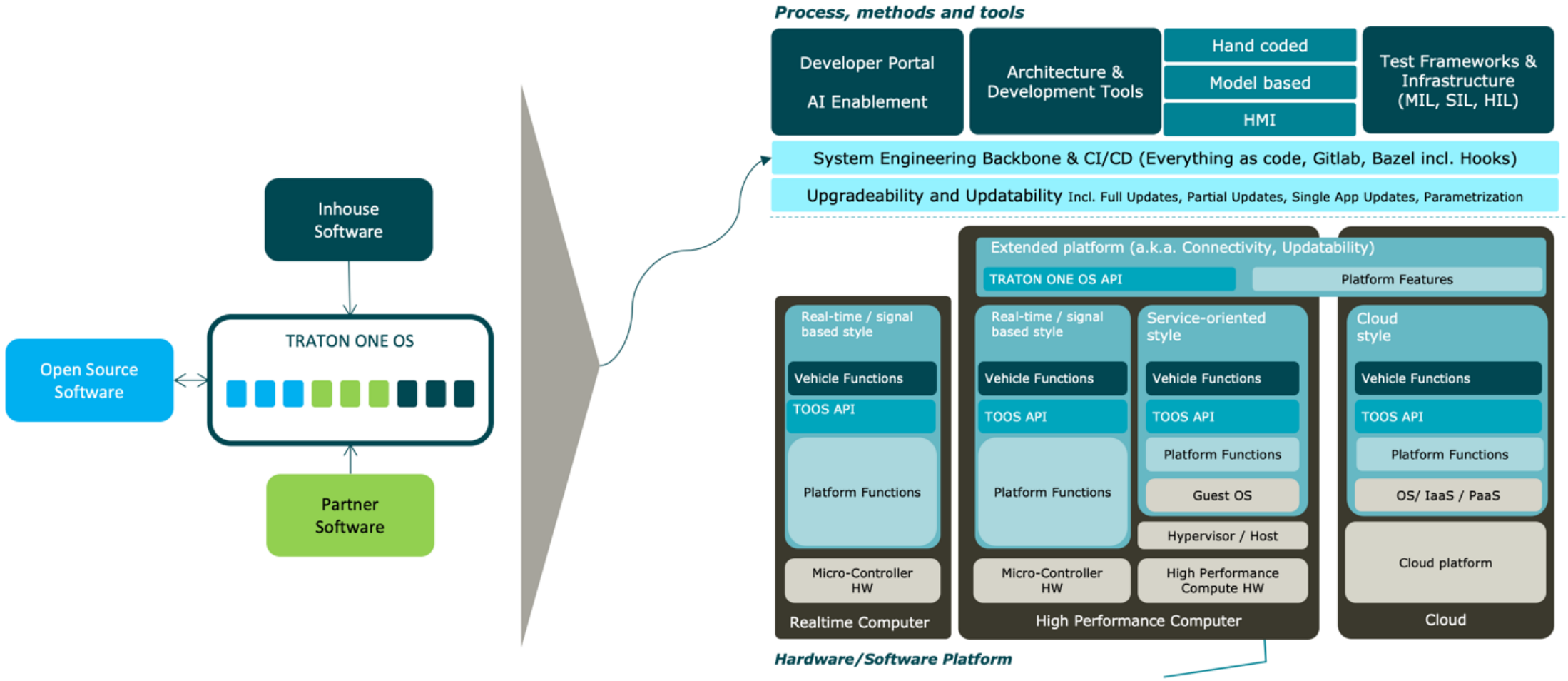
TRATON ONE OS integrates in-house, partner and open-source software on central compute

Development and validation toolchain

Everything as code | GitLab + Bazel | MIL / SIL / HIL

Developer portal | AI enablement

TRATON ONE OS: MODULAR SOFTWARE ARCHITECTURE



Software composition

In-house, partner and open-source components form TRATON ONE OS.

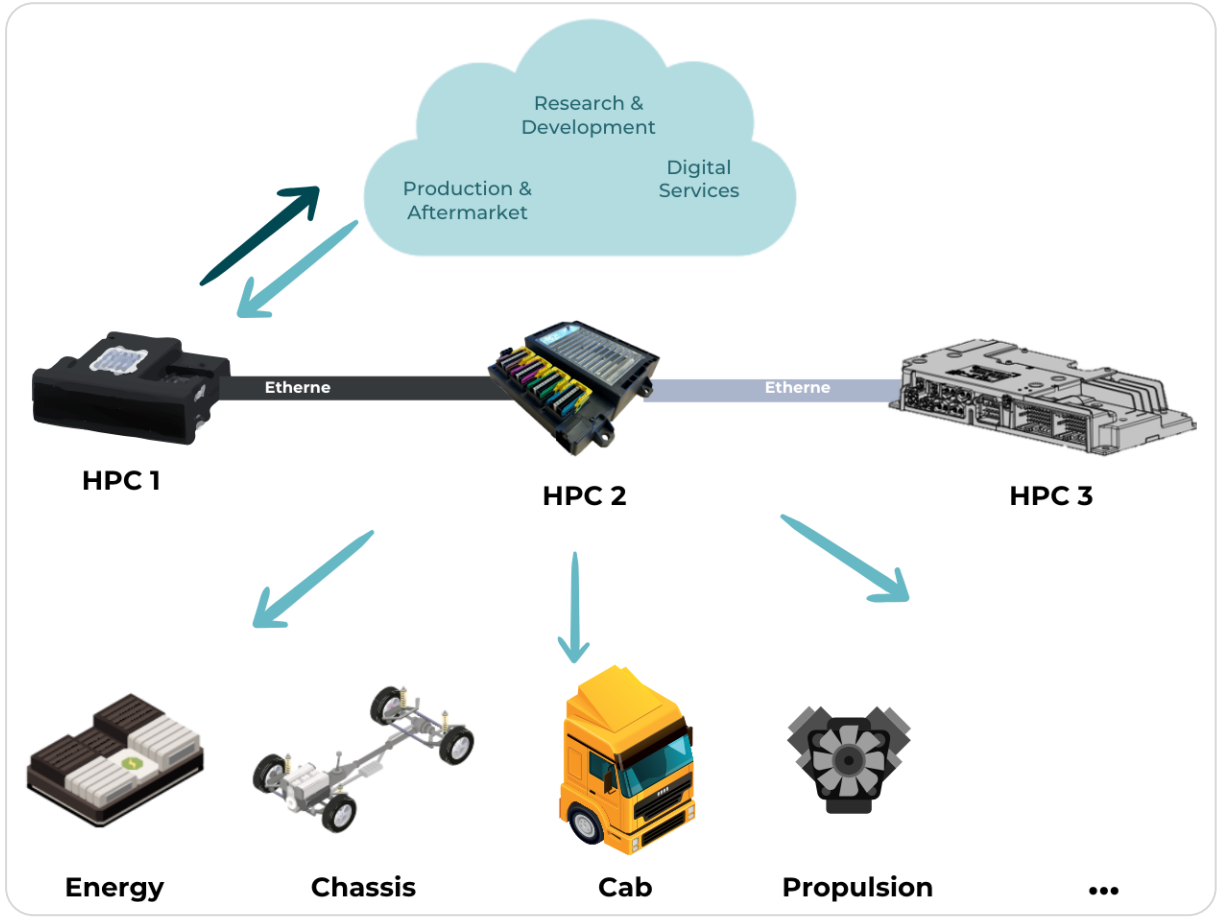
Execution environments

The TOOS API spans real-time, service-oriented and cloud software.

Lifecycle support

Shared engineering tools and CI/CD support full, partial and app updates.

CENTRAL COMPUTE WITH FOUR SOFTWARE FLAVOURS



Zonal architecture: high-performance computers (HPC) connected to the energy, chassis, cab and propulsion zones

Four software stacks address different execution needs

Adaptive AUTOSAR

Runs Adaptive AUTOSAR applications on QNX.

ASIL B

Android

Provides the software environment for infotainment.

QM

Eclipse SDV on Linux

Runs containerized applications on Linux. Series introduction is in preparation.

QM

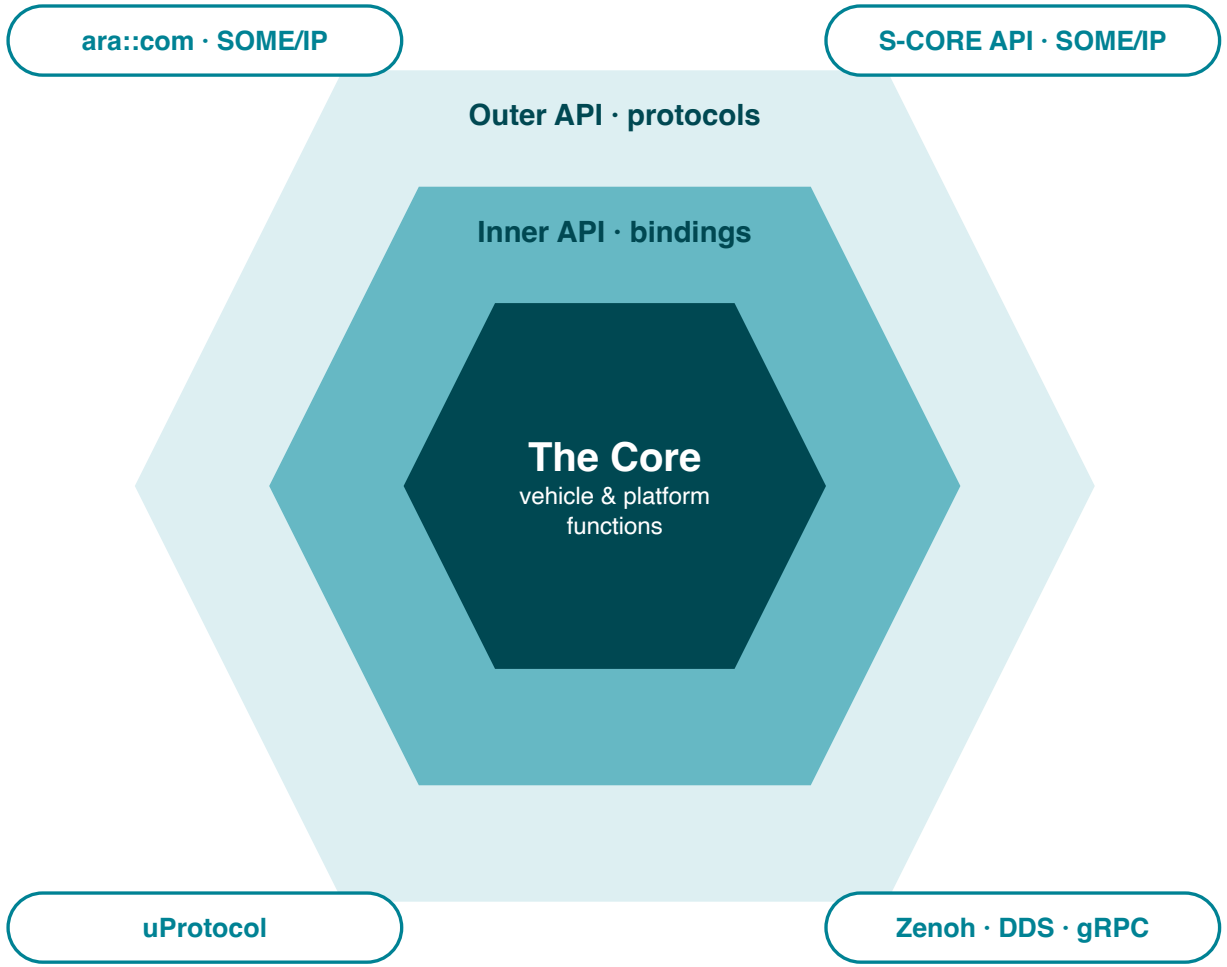
Eclipse SDV safety stack

Explores a safety-grade stack. The base operating system remains undecided.

ASIL B target: proof of concept

Teal stacks use Eclipse SDV, with distinct maturity levels

THE TRATON ONE OS API: SWAP THE PLUG, KEEP THE FUNCTION



The Core – what the truck does

Vehicle user functions serve drivers and fleets; platform functions keep the vehicle itself running.

Inner API – how functions are written

Language-level bindings such as ara::com or the Eclipse S-CORE API.

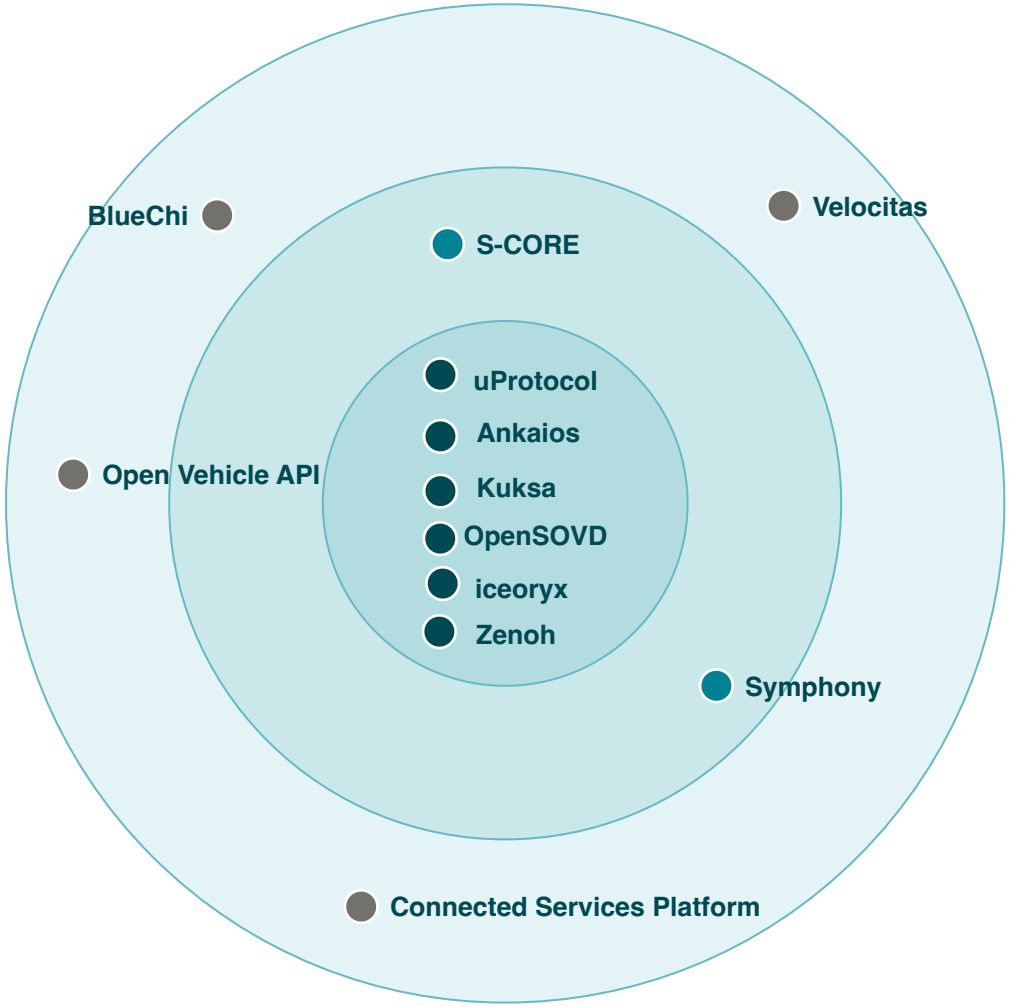
Outer API – how data travels

SOME/IP, DDS, gRPC or Zenoh on the wire; COVESA VSS or Protobuf as data formats.

Our rule of thumb

Open APIs at the edges, freedom to swap the implementation underneath – and never at the cost of a start of production.

OUR OPEN-SOURCE TECHNOLOGY RADAR



Twelve Eclipse projects we work with hands-on

Very high likelihood of use

Communication, orchestration and vehicle data – the backbone for our QM containers on Linux.

High likelihood of use

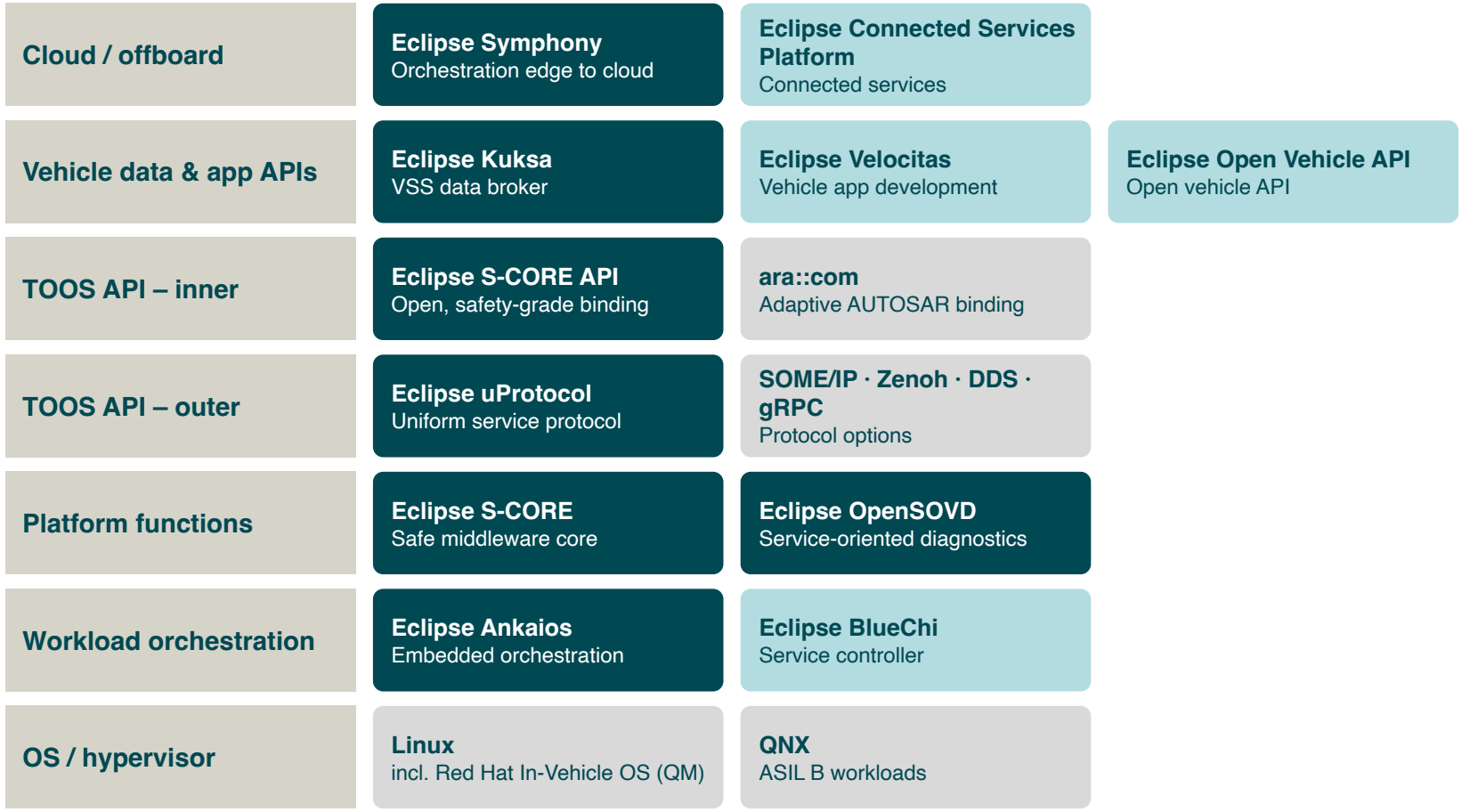
Safety-grade middleware, diagnostics and edge-to-cloud orchestration – maturing alongside our ASIL B proof of concept.

Exploring

App development, service management and connected services – we are watching fit and momentum.

Snapshot of our current evaluation – not a commitment to series use.

WHERE ECLIPSE SDV PROJECTS FIT IN TRATON ONE OS



Illustrative mapping of the projects on our radar onto the TRATON ONE OS layers. Evaluation status – no commitment to series use.

1 Open where functions meet the platform

We want an open onboard API build on Eclipse SDV projects – the implementations underneath stay swappable.

2 A backbone for the QM stack

uProtocol, Kuksa and Ankaios cover communication, vehicle data and orchestration for containers on Linux.

3 A safety-grade path

S-CORE brings the middleware for our ASIL B track; OpenSOVD brings modern, service-oriented diagnostics.

4 Mix, don't monopolise

Open source sits next to inhouse and partner software – each layer can pick what fits best.

ECLIPSE S-CORE: HOW WE EVALUATE AND WHAT TRUCKS NEED

How we evaluate – hands-on, PoC / pilot

QM and ASIL B
safety and security properties for both integrity levels

Linux and QNX
running on our high-performance computers

Performance & scalability
from a single function to the full variant range

Fit for SOP timelines
release maturity and a maintenance story we can rely on

What commercial vehicles need on the road to S-CORE 1.0

1. **Stable, versioned APIs** – a dependable leading open API across releases
2. **Reusable safety & security work products** – to support our ASIL B argumentation
3. **Long-term support** – trucks and buses stay in service for many years – maintenance and hyper care beyond SOP
4. **Modularity for many variants** – four brands, one modular system – compose and dispose, no monolith
5. **Commercial-vehicle use cases** – e.g. uptime, diagnostics (with OpenSOVD) and fleet operation

How we engage

Evaluate hands-on
Twelve projects, PoCs and pilots – today

Work in the open
setting up Inner Source and an OSPO

Give back
contribute fixes, features and truck know-how

SIX LESSONS FROM OUR JOURNEY SO FAR

Open source is not a side project – it is an integral principle of what we build.

1. **Shared engineering principles**
   Agree on modularity and everything as code before choosing tools or software stacks.

2. **Traceable design decisions**
   Record design choices and their rationale so teams can review changes as the architecture evolves.

3. **A roadmap tied to production**
   Define the target architecture, then plan achievable steps for each start of production.

4. **Sourcing by component**
   Choose in-house, supplier or open-source software for each component. Retain the expertise to integrate it.

5. **Safety needs evidence**
   Assess open-source maturity and the evidence needed to support the vehicle's ASIL B safety case.

6. **Ownership beyond launch**
   Plan support and maintenance from the outset, with clear responsibility for updates throughout vehicle life.

SHARE THE FOUNDATION, COMPETE ON WHAT'S ON TOP

1

Share what doesn't differentiate

Middleware, orchestration, diagnostics and vehicle data access are better built once – together – than many times alone.

2

Turn specifications into running code

Open implementations make interfaces precise, tested and maintained – faster than standards on paper.

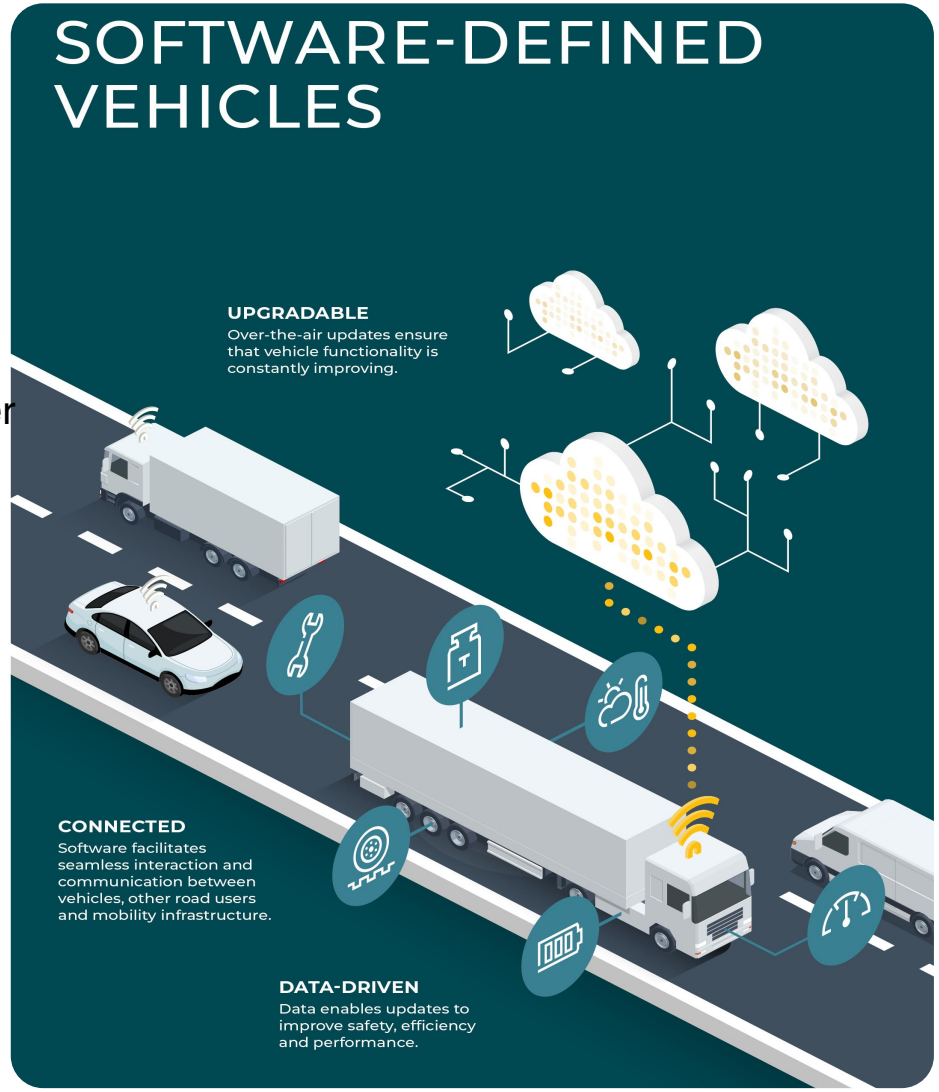
3

Bring commercial vehicles to the table

We bring truck and bus requirements – uptime, long lifecycles, variants – into the Eclipse SDV projects.

Let's talk

S-CORE on the road to 1.0 · uProtocol, Kuksa and Ankaios in practice · OpenSOVD for truck diagnostics





#thankYou