Challenges for the Automotive Platform of the future

Guido Ghisio

ADA – Europe 2016

RELIABLE SOFTWARE TECHNOLOGIES

Scuola Superiore Sant'Anna – PISA

15 June 2016

Electronics

Index

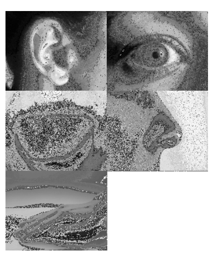
- What tomorrow Vehicle will be
- ADAS growth & evolution
- From Adas to Automated Driving
- Automotive SW development at its best

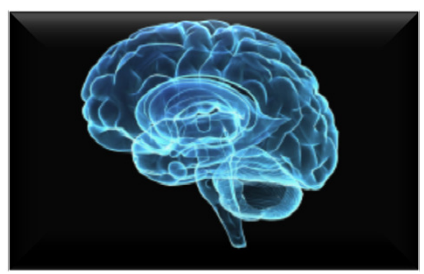
What tomorrow Vehicle will be

What tomorrow Vehicle will be

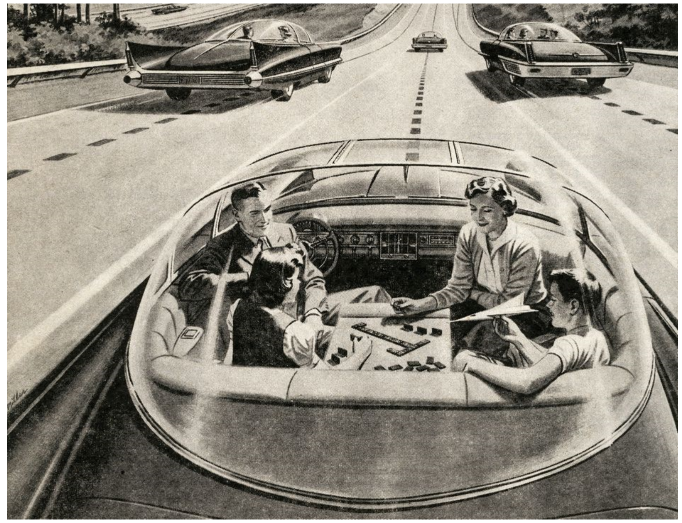
The vehicle once was a **passive platform**,
completely controlled by the **human driver**.

Sense → Think & Decision → Act



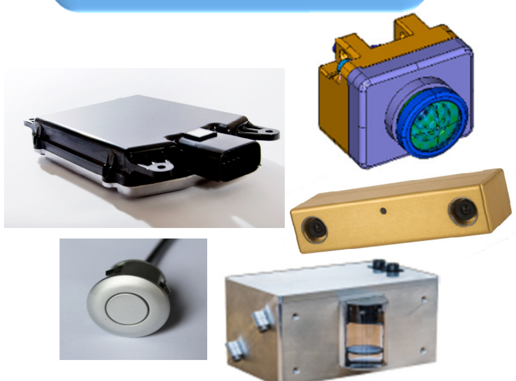


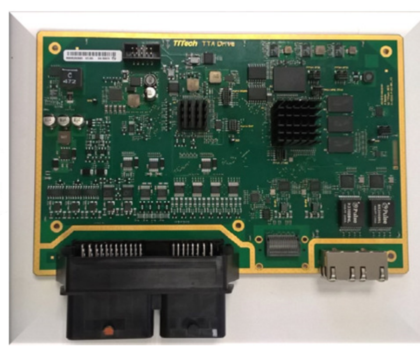
What tomorrow Vehicle will be



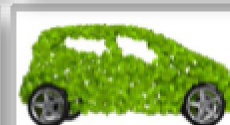
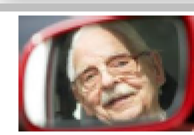
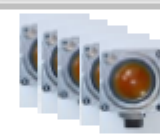
In the next future, more and more functions of car driving will be **automated**.

Sense → Think & Decision → Act





Automated Driving Motivations

1	*Road Safety: Vision Zero*	Road safety improvements by reducing human driving errors	
2	*Traffic management*	- Optimization of traffic flow management - Convenient, time efficient driving via automation	
3	*Reducing Emissions*	Reduction of fuel consumption & CO_2 emission (through optimization of traffic flow management)	
4	*Demographic Change*	- Support unconfident drivers - Enhance mobility for elderly people	
5	*Innovation High technology*	- New economic paradigm – supporting innovation policies of regions, nations - Competitiveness / high skill employment	

Automated Driving Motivations

Road Safety: 94% of all accidents are caused **by human error** (source: http://www-nrd.nhtsa.dot.gov/pubs/812115.pdf)

Table 1. Driver-, Vehicle-, and Environment-Related Critical Reasons

Critical Reason Attributed to	Estimated Number	Estimated Percentage* ± 95% conf. limits
Drivers	2,046,000	94% ±2.2%
Vehicles	44,000	2% ±0.7%
Environment	52,000	2% ±1.3%
Unknown Critical Reasons	47,000	2% ±1.4%
Total	2,189,000	100%

*Percentages are based on unrounded estimated frequencies (Data Source: NMVCCS 2005–2007)

Table 2. Driver-Related Critical Reasons

Critical Reason	Estimated (Based on 94% of the NMVCCS crashes) Number	Estimated (Based on 94% of the NMVCCS crashes) Percentage* ± 95% conf. limits
Recognition Error	845,000	41% ±2.2%
Decision Error	684,000	33% ±3.7%
Performance Error	210,000	11% ±2.7%
Non-Performance Error (sleep, etc.)	145,000	7% ±1.0%
Other	162,000	8% ±1.9%
Total	2,046,000	100%

*Percentages are based on unrounded estimated frequencies (Data Source: NMVCCS 2005–2007)

A great effort is applied by the whole automotive industry (OEMs, Tier1, Tier2) to increase driving safety by avoiding the two major causes of human error:

Distracted driving

Reckless driving

ADAS growth and evolution

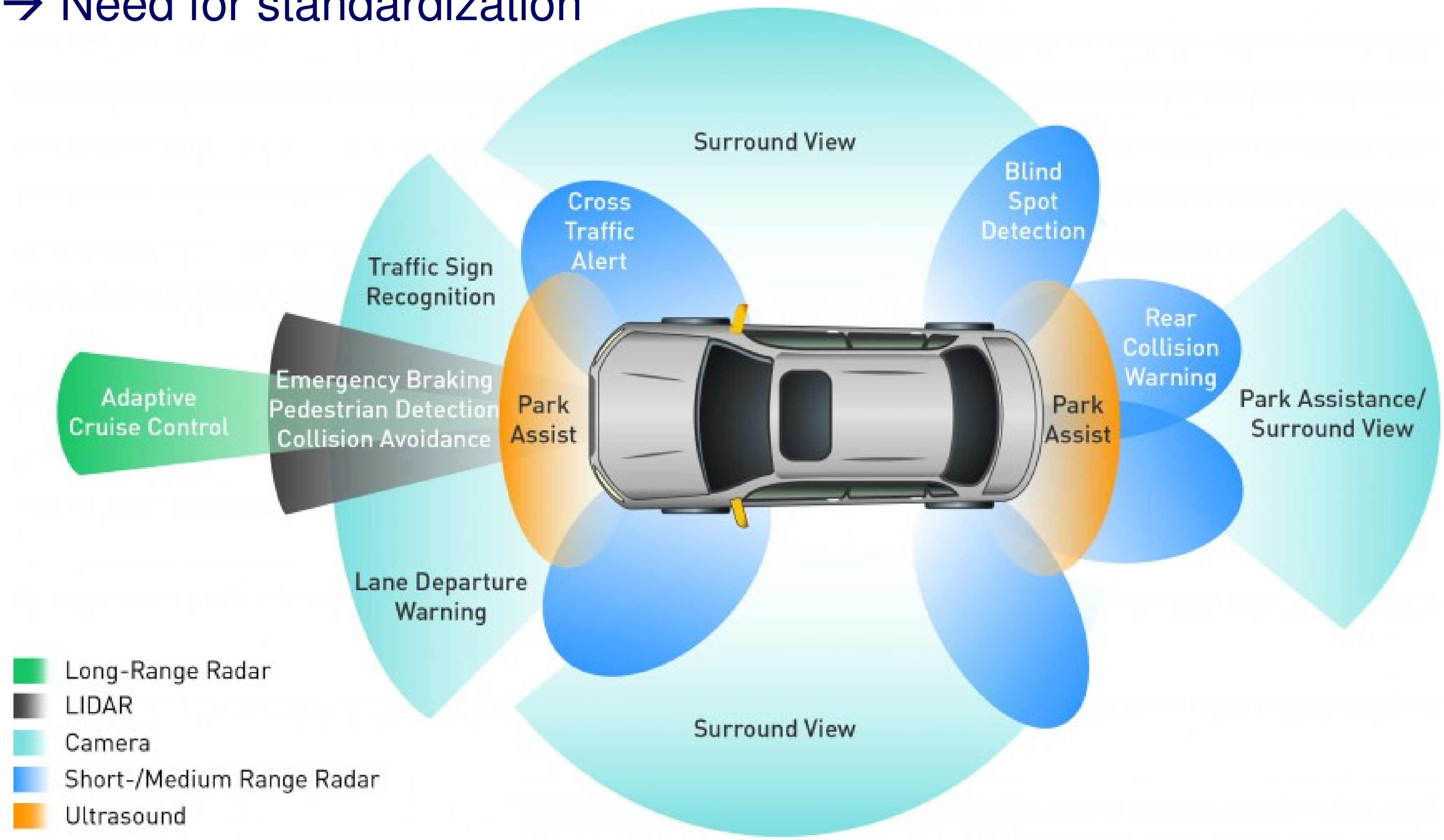
ADAS growth and evolution

- **Different subsystems** (maybe from several suppliers) within the same car

This can be a problem not only in terms of communication, but also control:

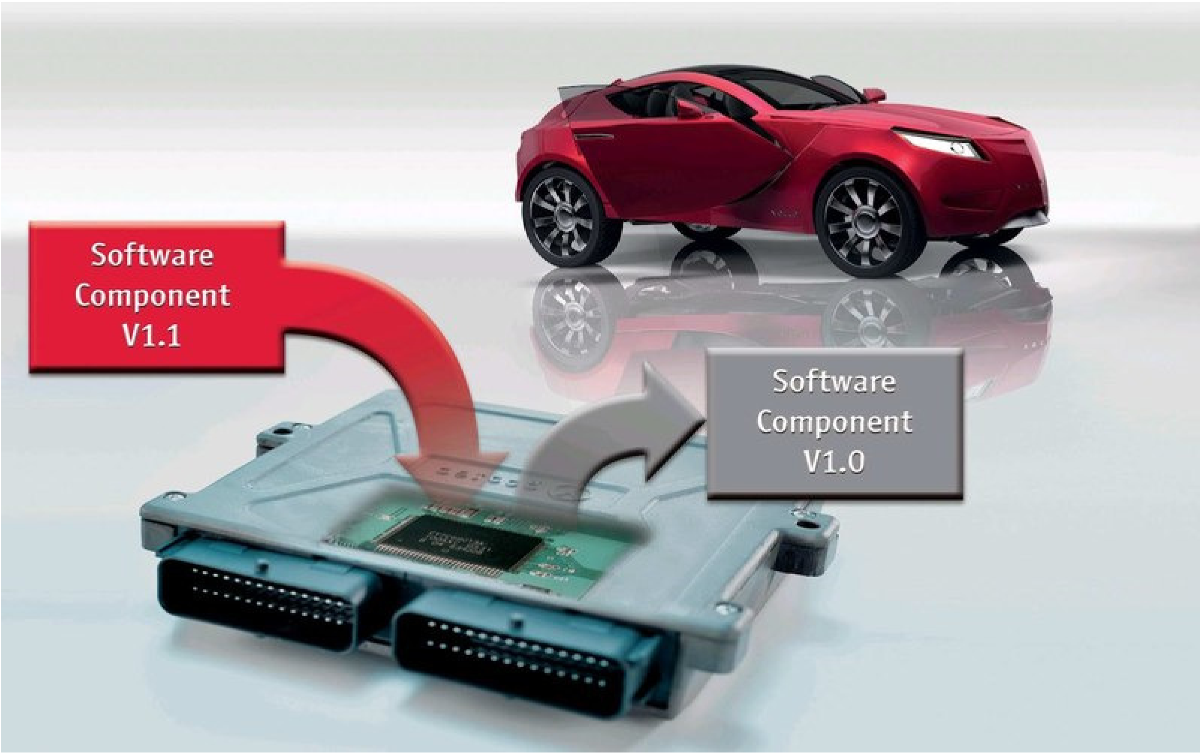
- Conflict between different systems' decisions/reactions to the same events

  → Need for standardization



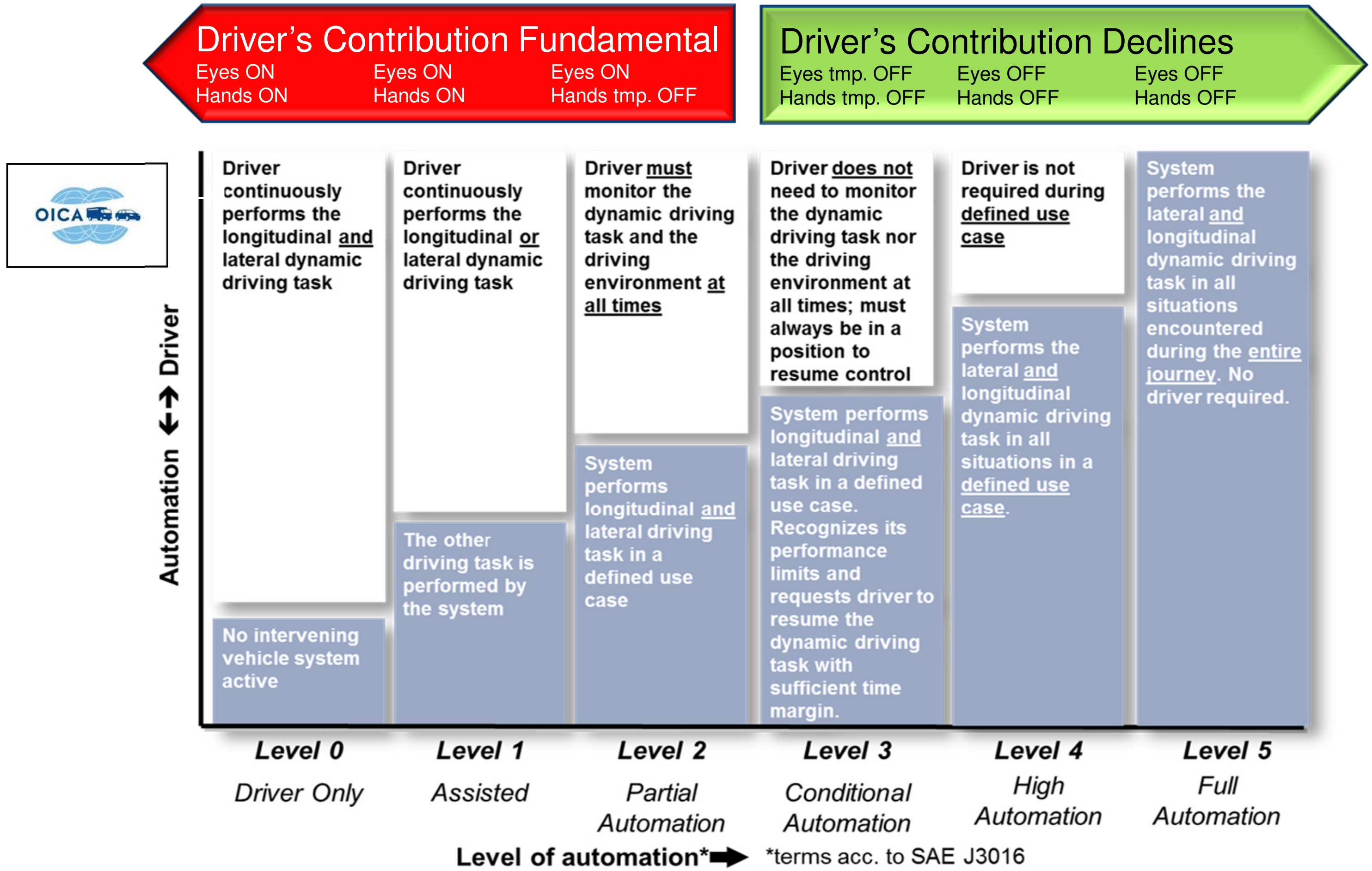
ADAS growth and evolution

The more useful innovation to help the future implementation of ever growing and heterogeneous ADAS, including subsystems coming from different suppliers, will be the use of a **shared platform**.



From Adas to Automated Driving

OICA Levels of Automated Driving



SAE (Society of Automotive Engineers)
NHTSA (National Highway Traffic Safety Administration)
OICA (Organisation Internationale des Constructeurs d'Automobiles)
MLIT (Ministry of Land, Infrastructure, Transport and Tourism)
CLEPA (Comité de Liaison de la construction d'Equipements et de Pièces d'Automobiles)
(European Association of Automotive Suppliers)
ERTRAC (European Road Transport Research Advisory Council)

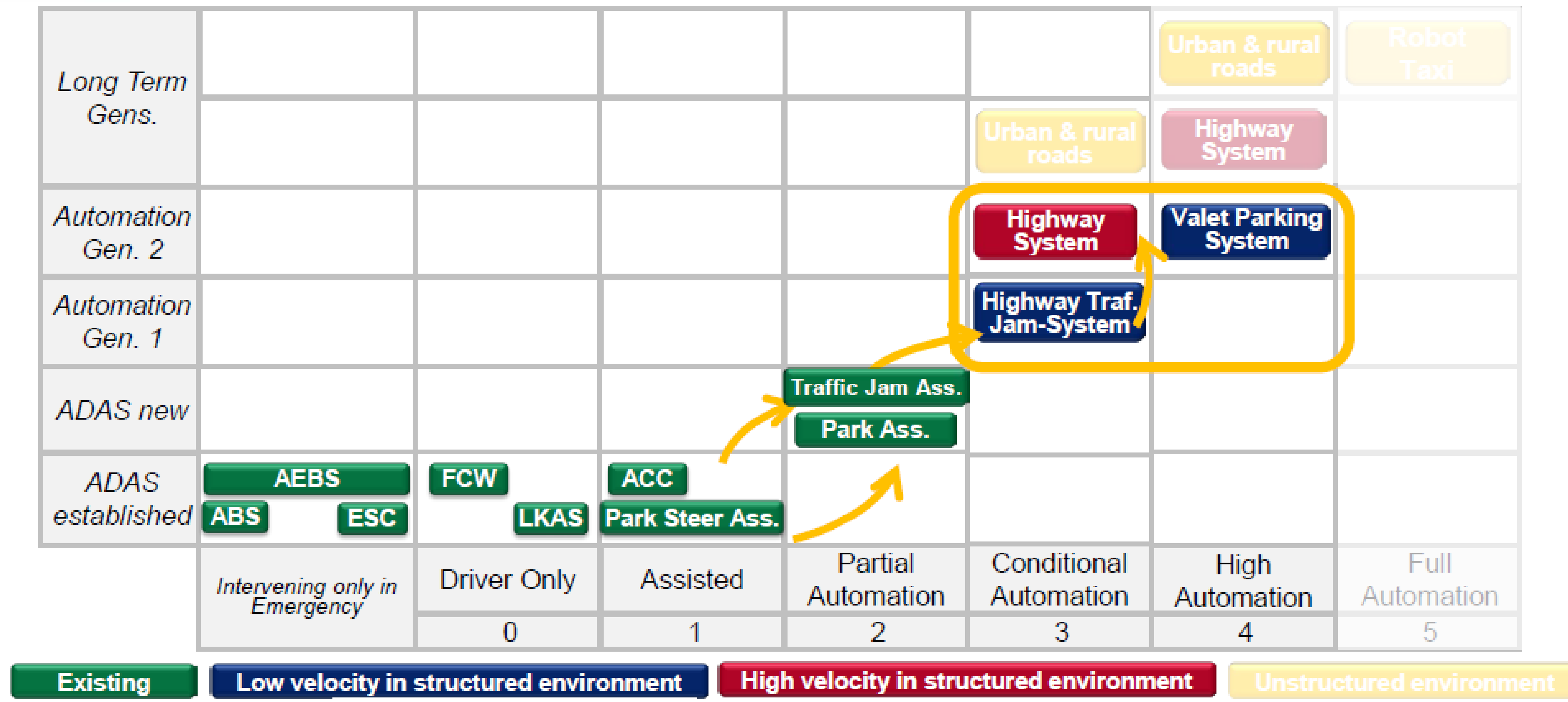
OICA Roadmap Automated Driving

	Intervening only in Emergency	Driver Only 0	Assisted 1	Partial Automation 2	Conditional Automation 3	High Automation 4	Full Automation 5
Longterm Gens.						Urban & rural roads	Robot Taxi
					Urban & rural roads	Highway System	
Automation Gen. 2					Highway System	Valet Parking System	
Automation Gen. 1					Highway Traf. Jam-System		
ADAS new				Traffic Jam Ass. Park Ass.			
ADAS established	AEBS ABS ESC	FCW LKAS	ACC Park Steer Ass.				

Existing | Low velocity in structured environment | High velocity in structured environment | Unstructured environment

Automated Driving , OICA 07.2015

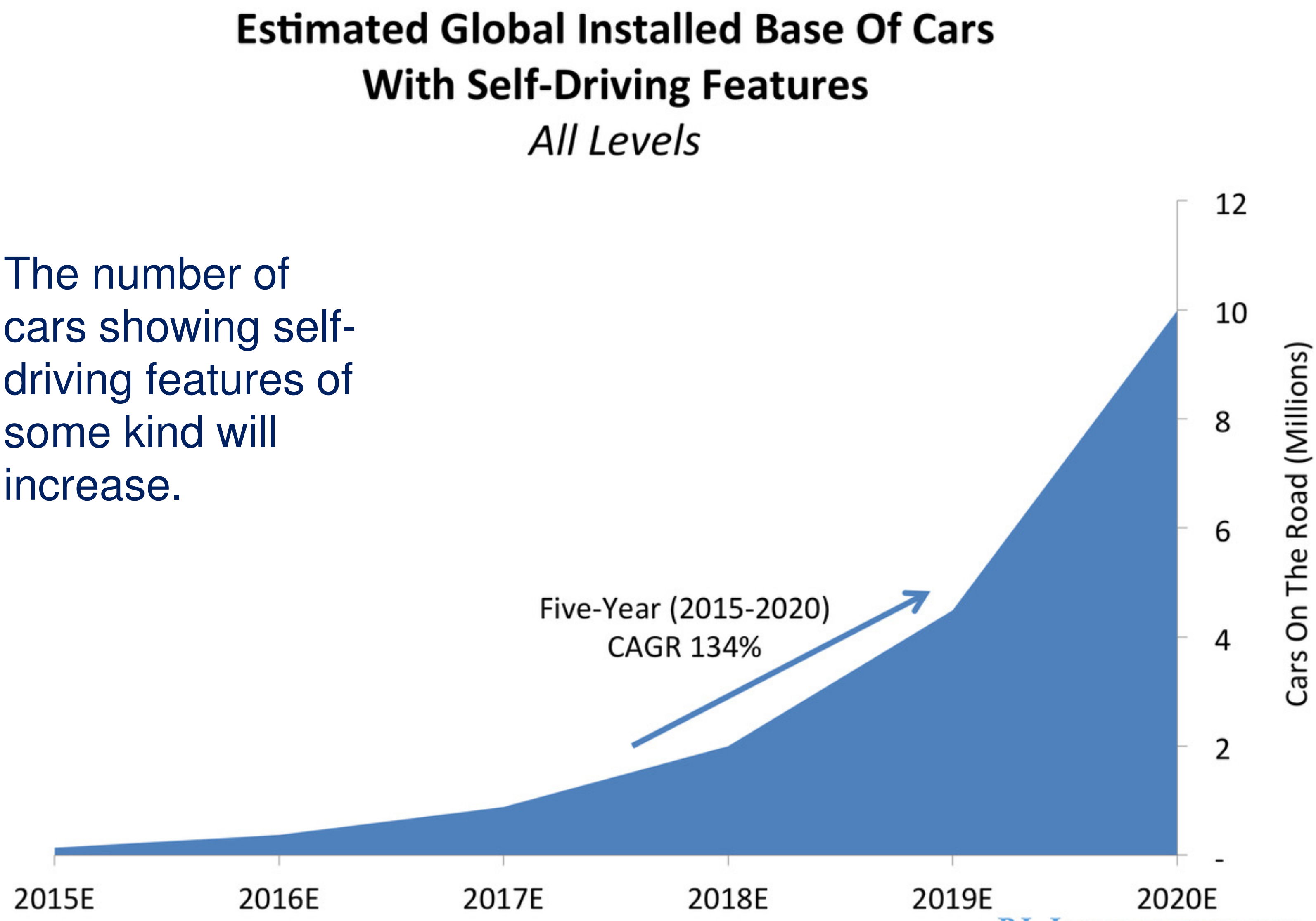
OICA Roadmap Automated Driving



Automated Driving , OICA 07.2015

Automated Driving : steps

Automated Driving : numbers



The number of cars showing self-driving features of some kind will increase.

Source: BI Intelligence Estimates, 2015 *CAGR = Compounded Annual Growth Rate

Automated Driving : HW Platform

A new kind of automotive hardware platform is required.

Traditional solutions are not suitable:

- They are too power-demanding
- They waste too much space

Allowing
Isolation
Predictability
Programmability
Functional Safety
Openness

Reducing
Cost
Power
Size

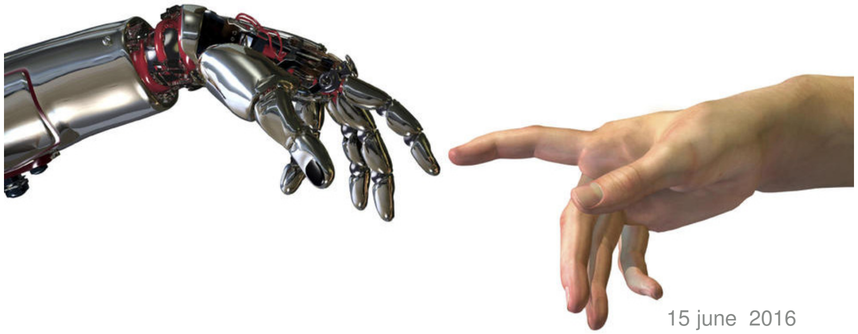
Automated Driving : Functional Safety

With the increase of electronic intervention on driving, the Functional Safety of **Electric/Electronic Systems** becomes more and more important.

Functional Safety is the "absence of **unreasonable risk** due to **hazards** caused by **malfunctioning behaviour** of **E/E systems**" (*from ISO 26262*).

ISO 26262 covers the **whole development cycle** of safety-relevant systems.

The next release of ISO 26262 (due in 2018) will also include new technologies raising in the automotive domain, like multi-core systems and MEMS devices.



Automated Driving

Low-power High-performance Computing

At the same time, automotive microcontrollers are facing a real **evolution**:

- Computing performance is increasing
- Power consumption is decreasing

Entire research areas finally have the supporting technologies to make their way into the automotive domain.

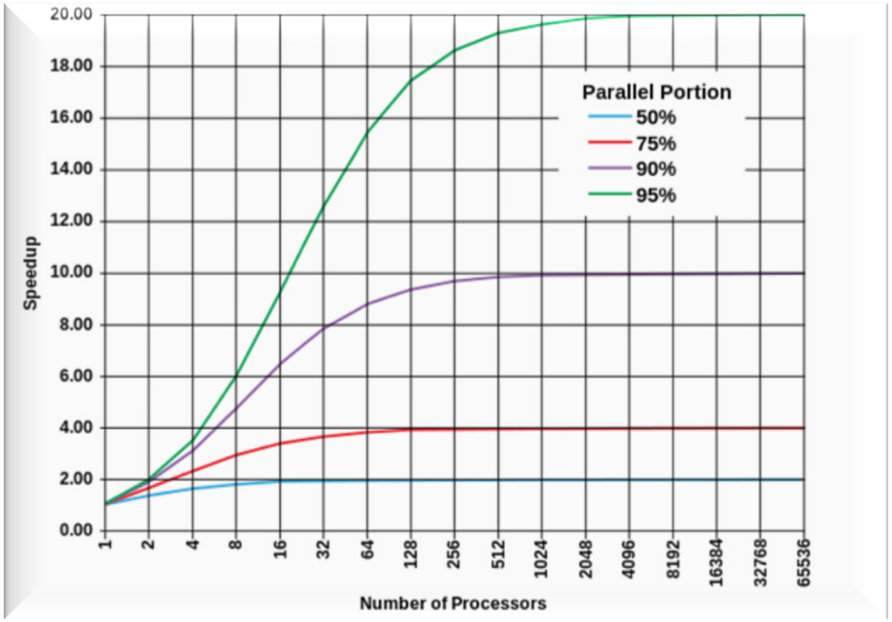
Automated Driving

Many-core architectures and centralized ECUs

The main reason to use many-core systems is

to achieve **higher computing power**, exploiting the benefits of software parallelization;

Amdahl's Law theorizes the speed up of a system when increasing the number of cores, with respect to possibility to *parallel* a given SW.



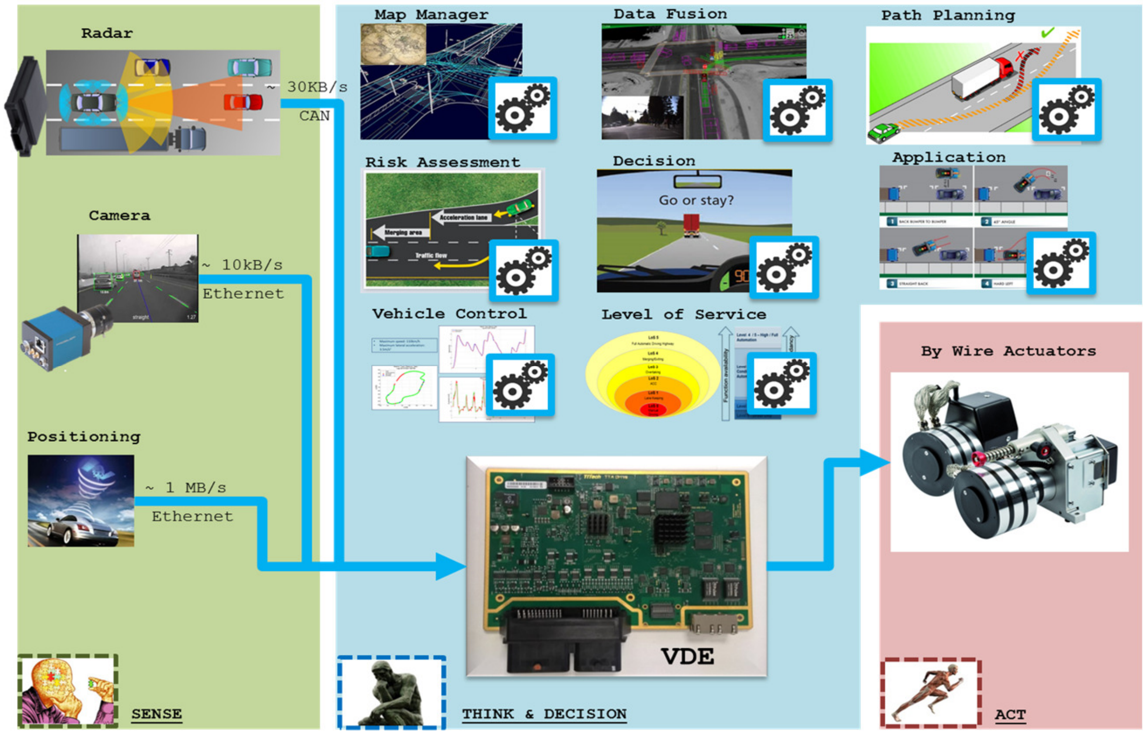
Instead of speeding up the same functionality, it is possible to use the gained computing performance to embed more functionalities in the same ECU;

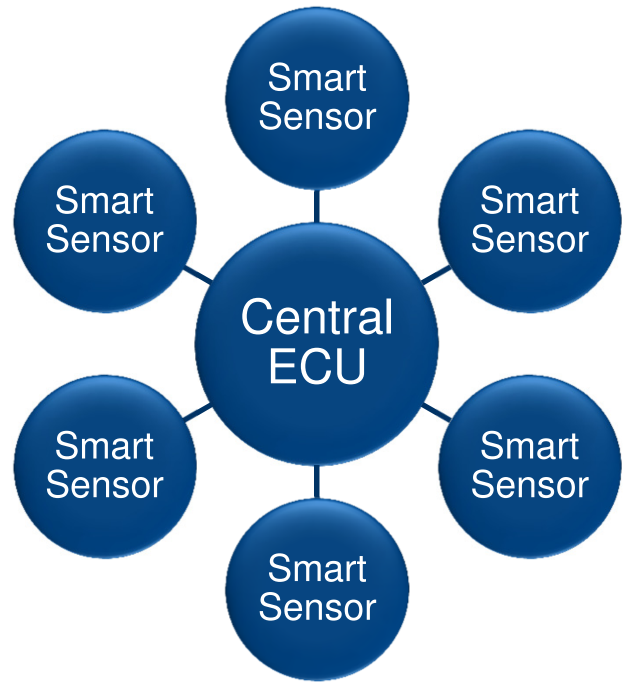
This enables the choice for bigger **centralized ECUs**, to decrease the number of different systems needed by the vehicle;

With multi-core systems, it is possible to host in the same ECU many different functions, possibly having different responsibilities on vehicle dynamics (and thus **different ASIL**) and running completely in parallel on different cores, to ensure **Freedom from Interference**.

Automate Driving current architecture (distributed)

❑ Smart sensors send objects tracked in space and time

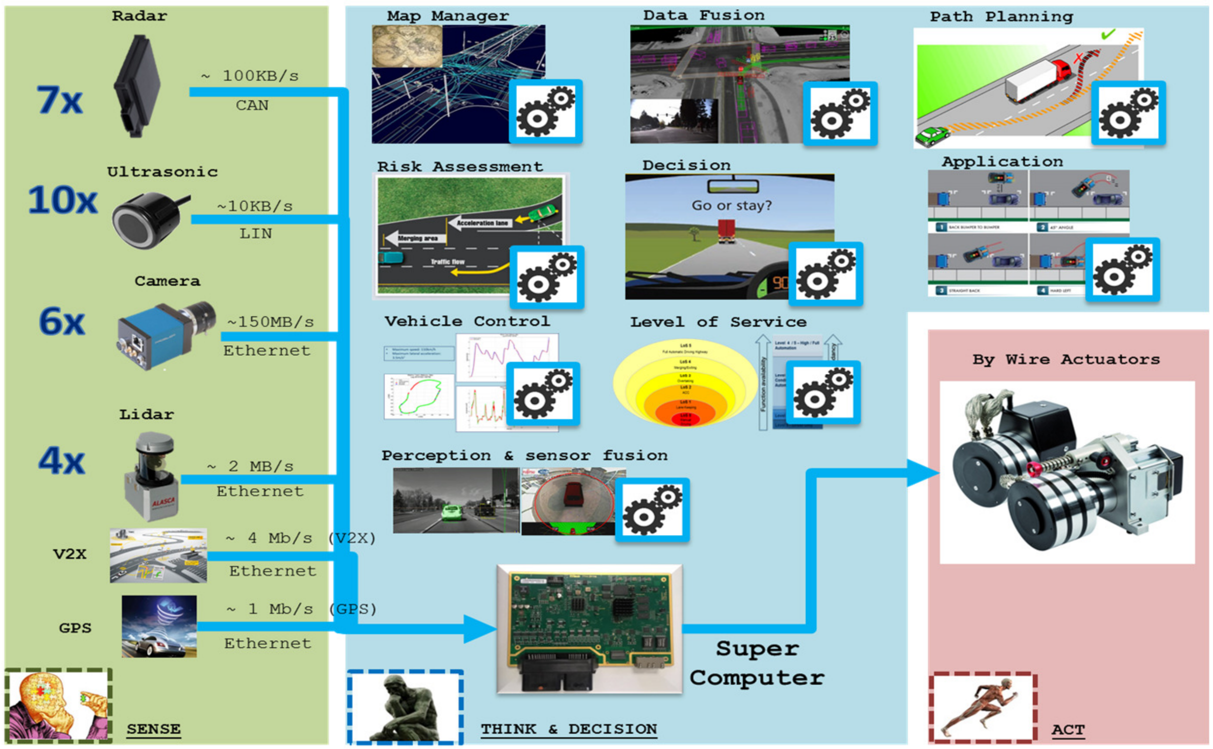


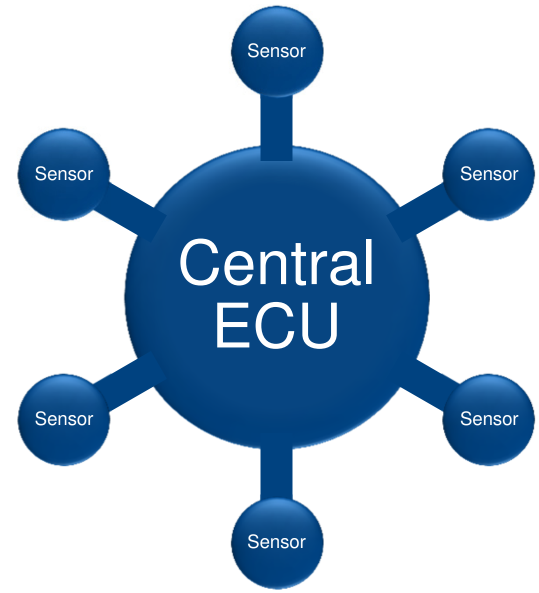


- ✓ More flexible and expandable
- ✓ Limited bandwidth requested for pre-processed data
- × Increased cost
- × Requires more complex sensors

Automated Driving My 2020 architecture (centralised)

❑ Dumb sensors send raw data





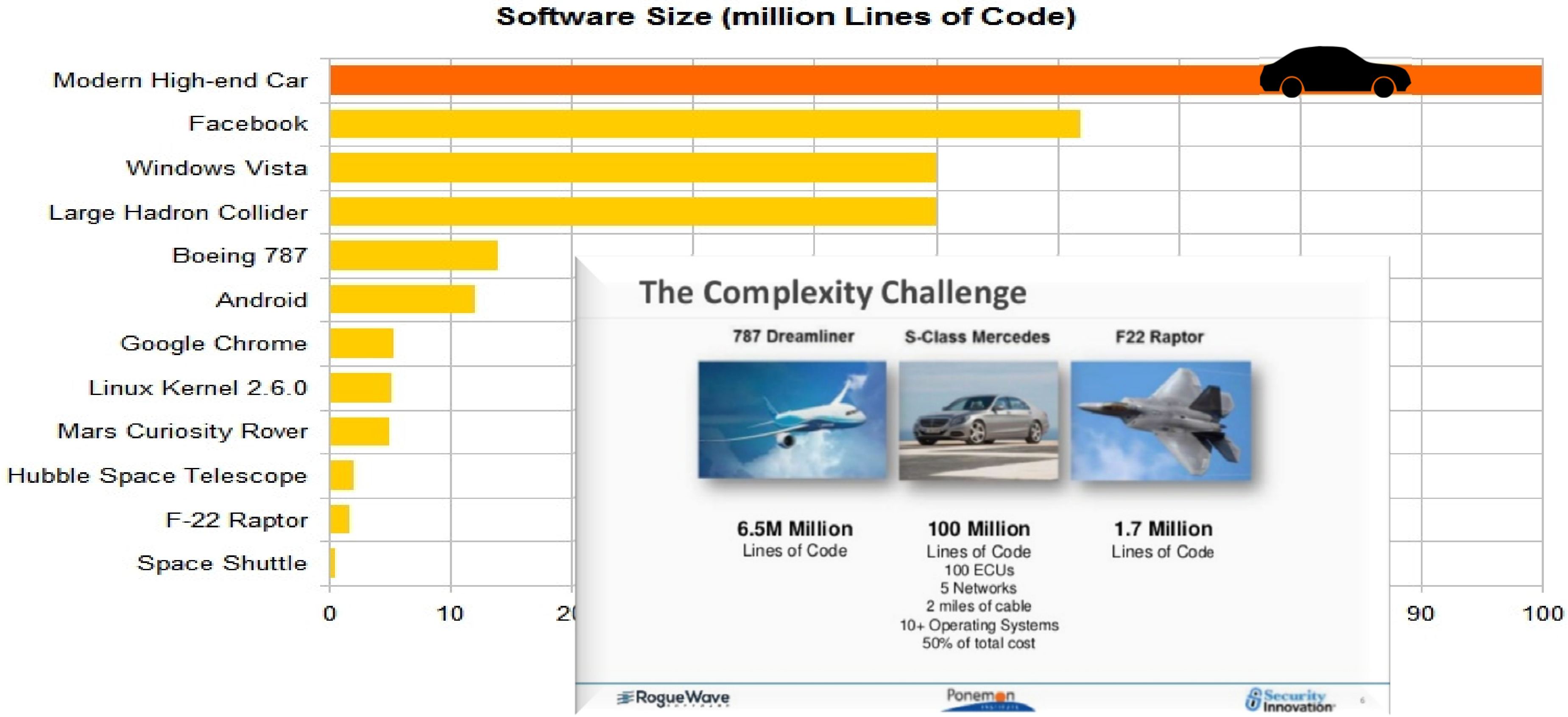
- ✓ Simpler sensors with no logic can be used
- ✓ Reduced cost
- × Less scalable
- × Raw data transmission demands high bandwidth

Automotive SW development at its best

Automotive SW development at its best

Cars have become **the most demanding application** for SW development.

(Source: http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code)



Automotive SW development at its best

Distributed Multi development

Distributed development allows the different SW modules used in a single ECU to be developed by **different SW departments or companies**:

- It helps each team **differentiate its expertise** in the different topics;
- It also enables **third-party SW development**;
- It can also drive an **increase in SW quality**.

Distributed development requires **standardization** to minimize issues during integration of SW modules.

Design for Validation

Design for Validation is the practice of conceiving systems taking into account that they are going **to be tested**.

ECU programming is done in a way that **eases validation testing** and limits the need for further **code instrumentation**, thus reducing **test invasiveness**.

Another advantage is the **reduction of costs** during test phase, because the code instrumentation is lighter.

SW reusability and portability

SW reusability and portability

In order to **reduce costs** and **increase SW reliability**, it is necessary to develop modules that are **reusable** and **portable**, exploiting **standard architectures**.

Conclusions

- **Standard platforms** are fundamental to allow next generations cars.

- There are **many different challenges**, that the **vertical-horizontal automotive industry** is already aware of, and many others are yet to be discovered.

- **Unified Securised** platforms should be the best solution for all challenges

Thank you

June 2016