The future is open

ARC CORE

Real Time Systems
TDDD07

Dec 8, 2017

The future is open
ARC CORE

Introduction

Daniels Umanovskis

- MSc, Computer Science, LiU
- Software developer at ARCCORE Linköping since 2013

Agenda

- ARCCORE
- Automotive real-time software
- Reliability and robustness
- Latest challenges
- Testing in practice
- Customer examples
- The future and lessons learned

ARCCORE

- Automotive software company
- ~100 engineers
  - Sweden: Gothenburg, Linköping
  - Global: Munich, Bangalore, Shanghai, Palo Alto
  - Customers all around the world

Photo: Björn Henrichsén

The future is open
ARC CORE

Automotive software

- Modern cars are software driven
- Up to 100 ECUs in a regular car
- Software is often life-critical

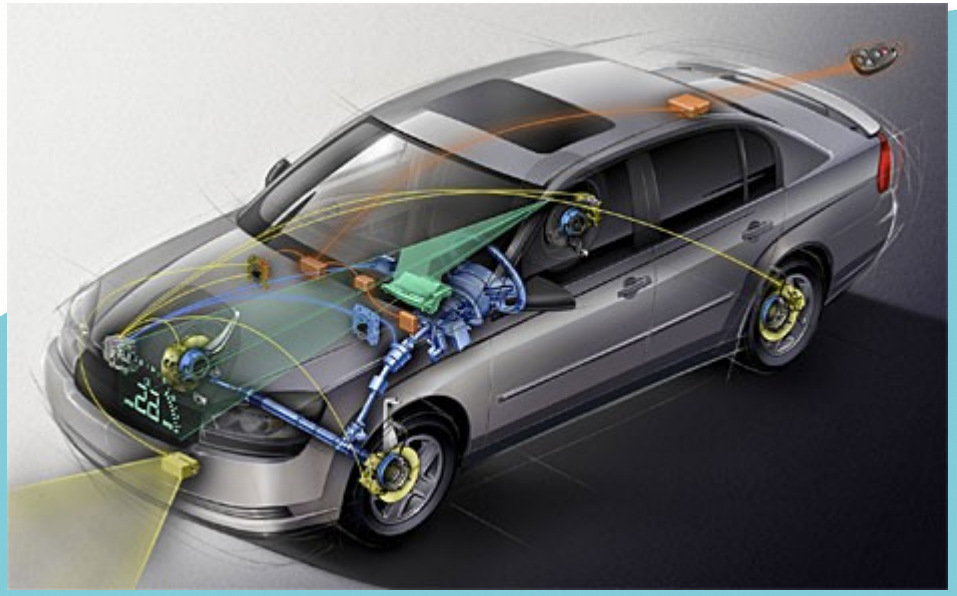


Image: General Motors

Automotive software

Photo: Volvo Cars

Automotive software

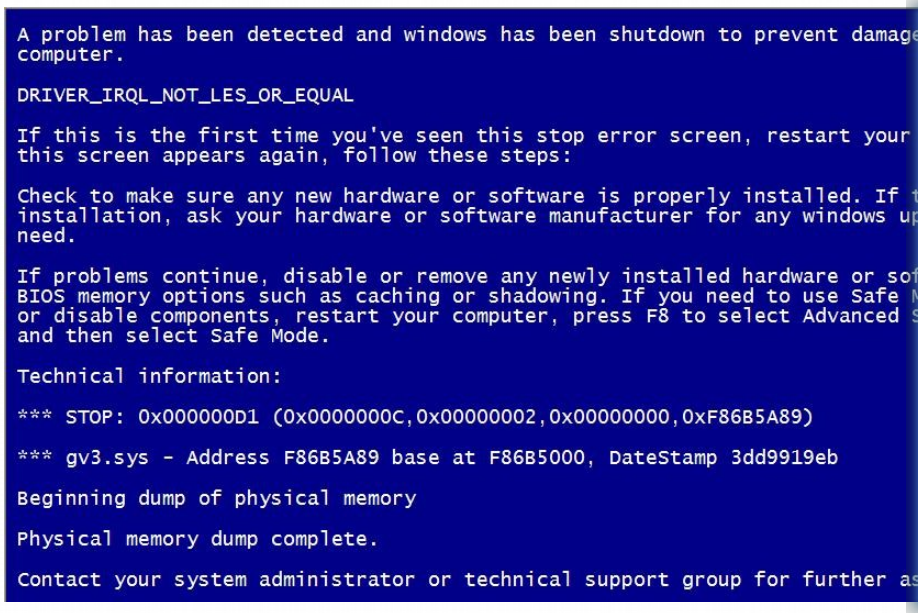
Photo: Beck Diefenbach / Reuters

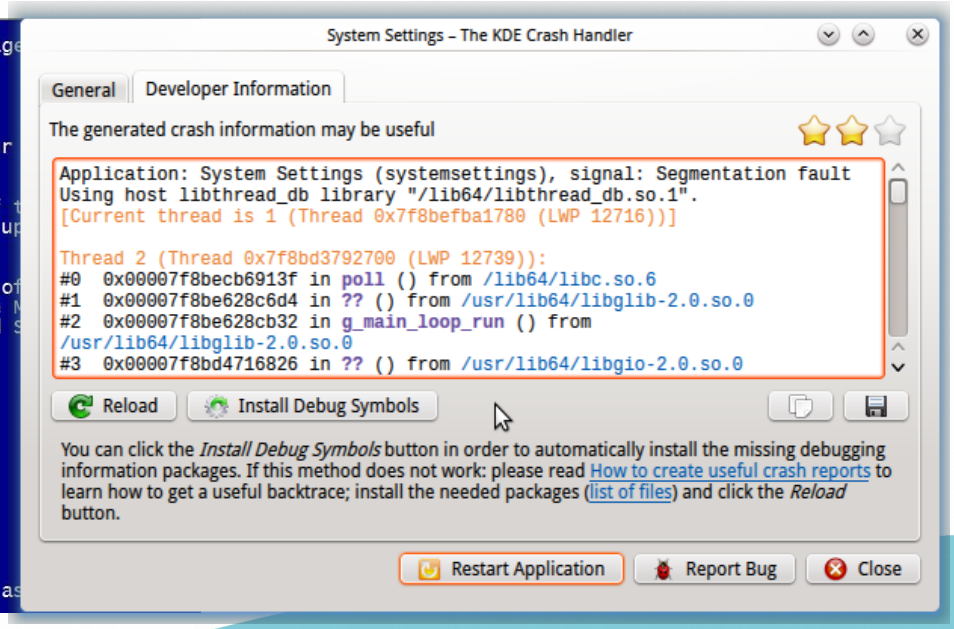
Automotive software

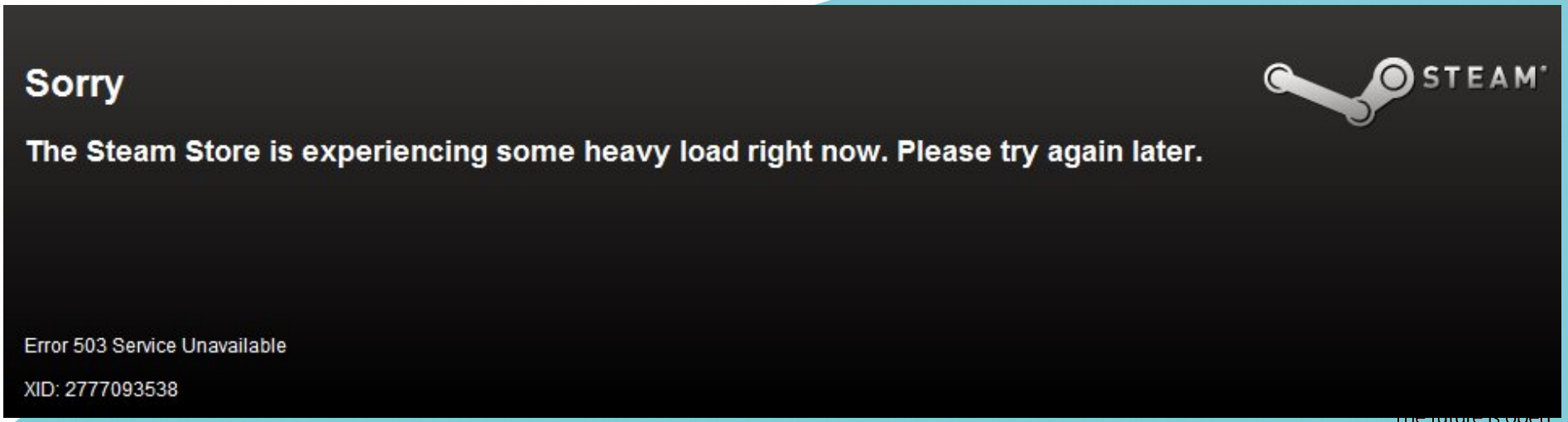
https://www.youtube.com/watch?v=APnN2mClkmk

Footage: Hans Noordsij

Oops, a bug







Oops, a bug

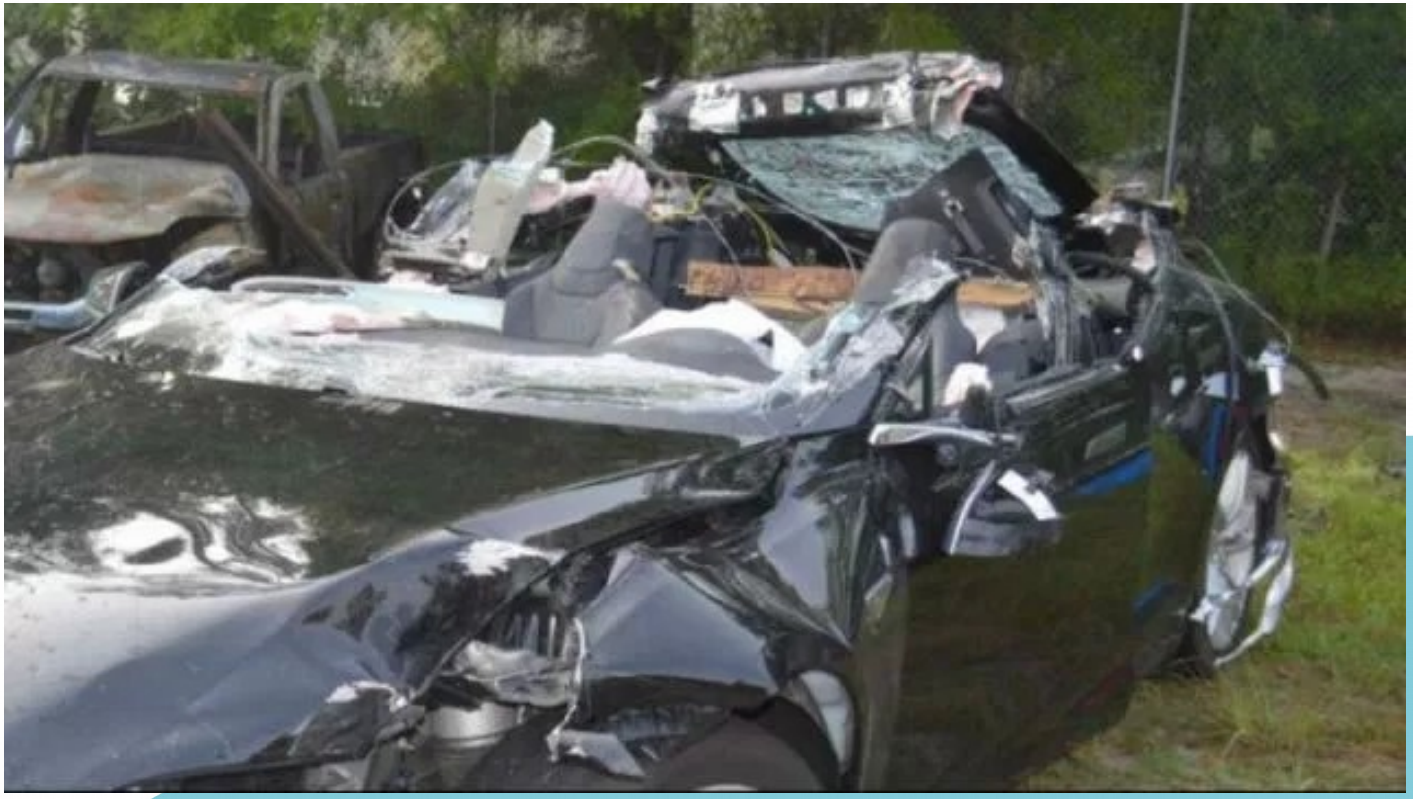


Photo: Florida Highway Patrol

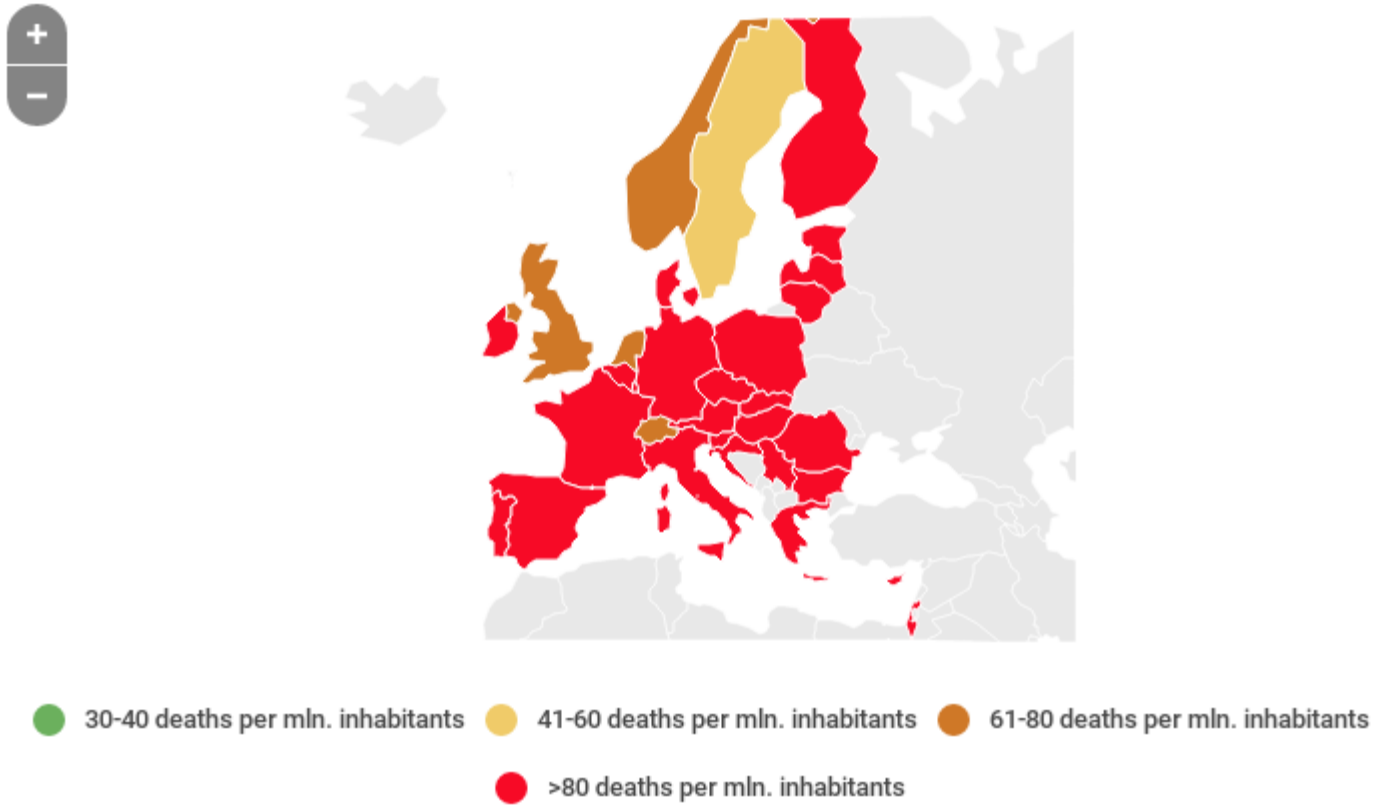
Oops, a bug

Witness photograph / Twitter

And other bugs...

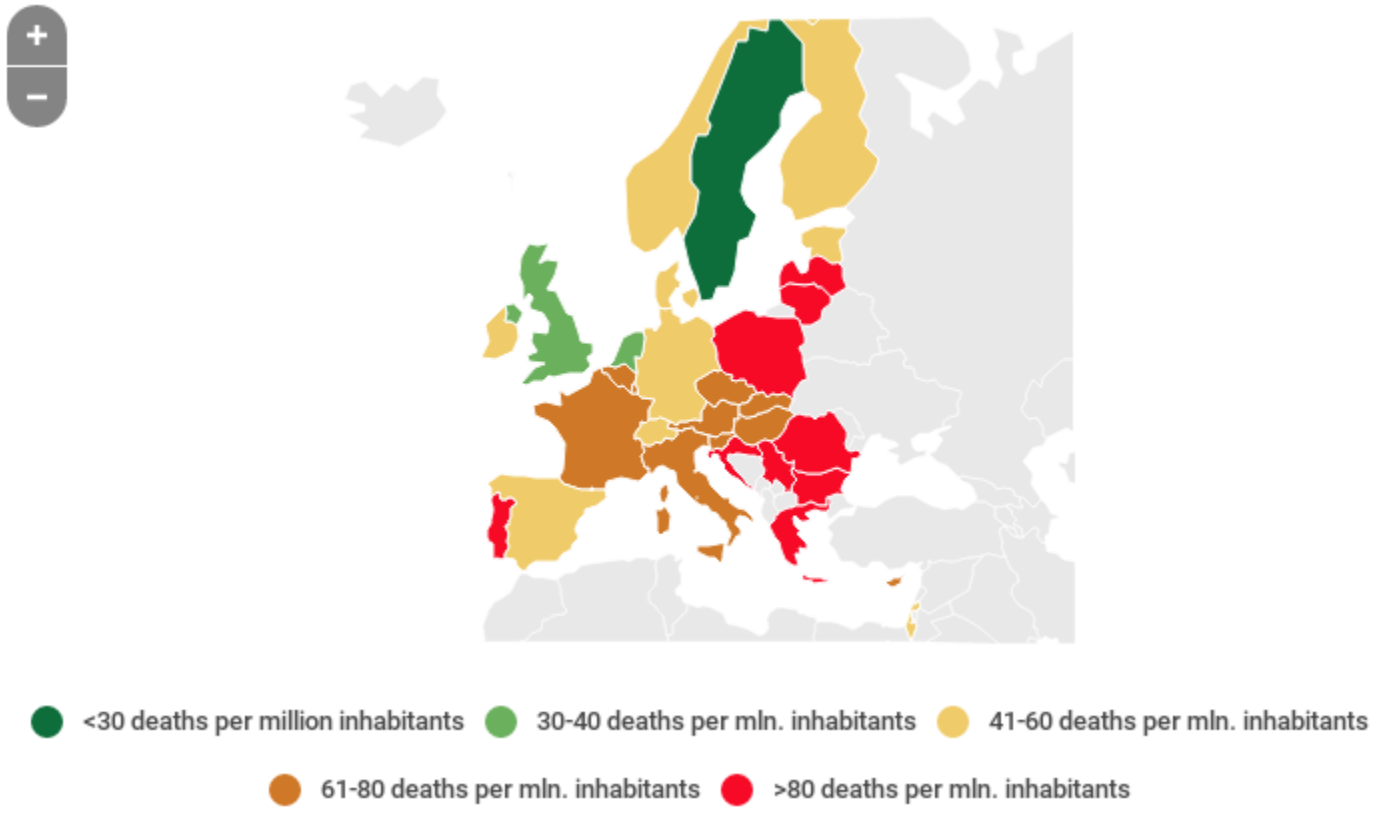
Image: Volvo Cars

Traffic safety (2001)



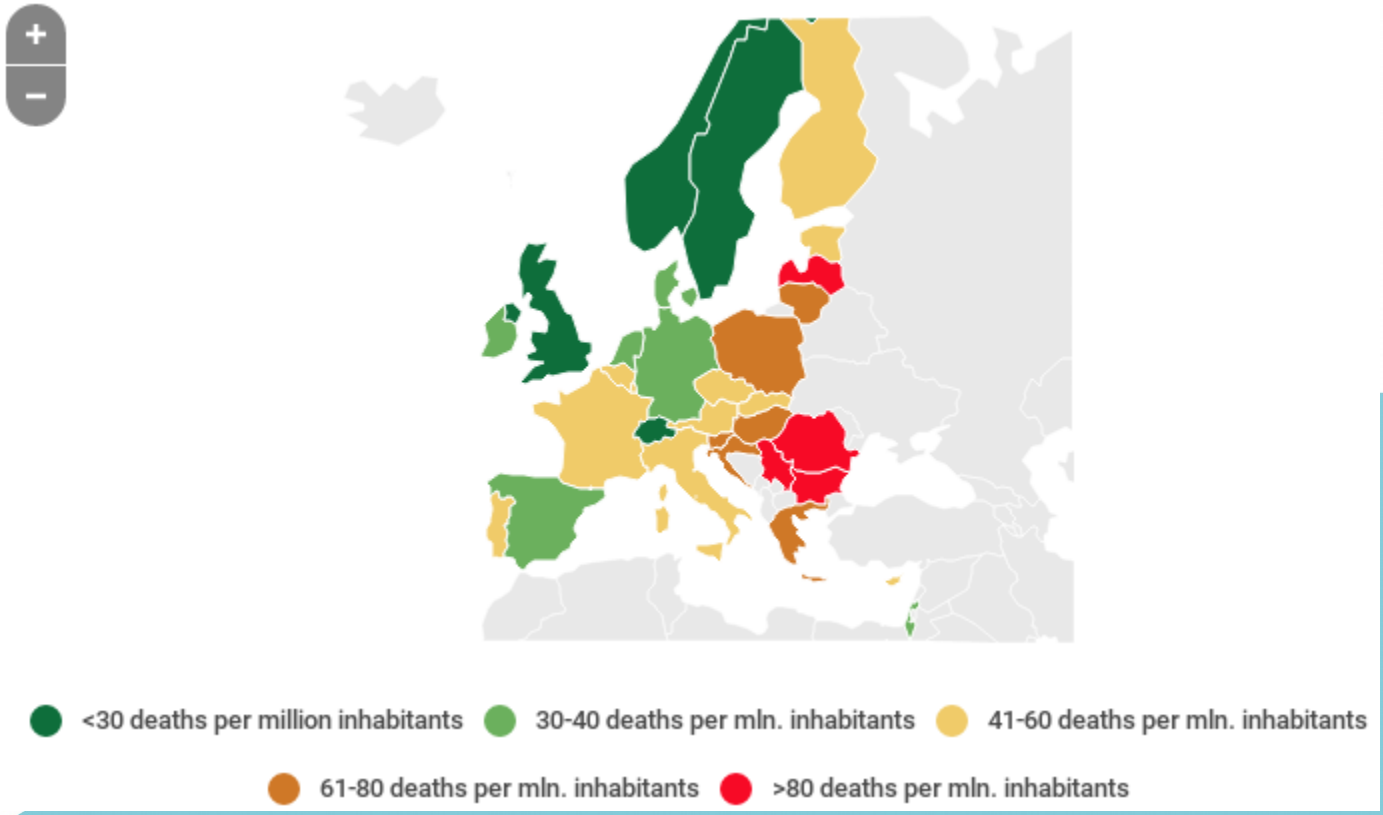
Source: European Transport Safety Council

Traffic safety (2010)



Source: European Transport Safety Council

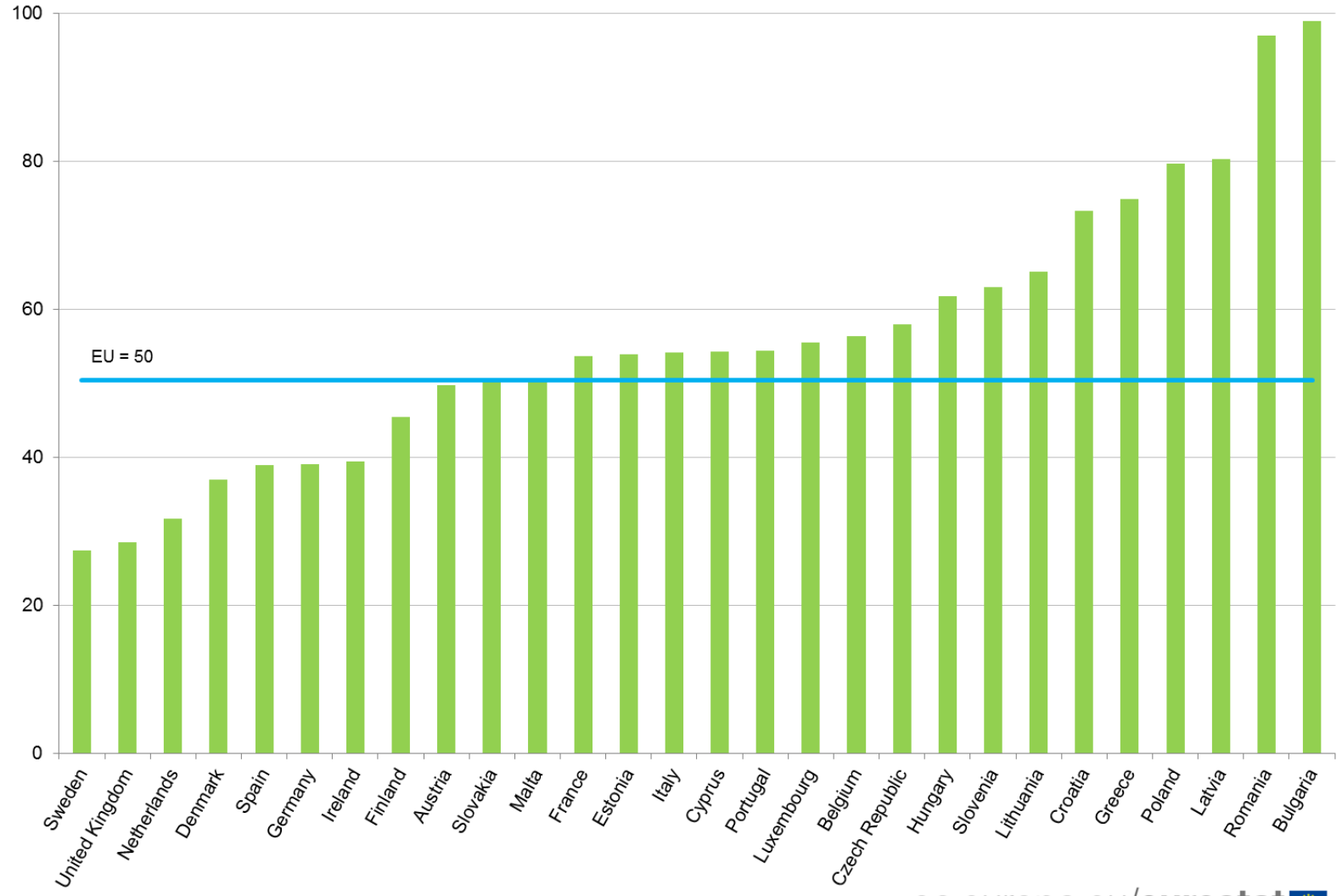
Traffic safety (2016)



Source: European Transport Safety Council

Traffic safety

Road traffic victims per million inhabitants in the EU Member States, 2016



ec.europa.eu/eurostat

Automotive standards

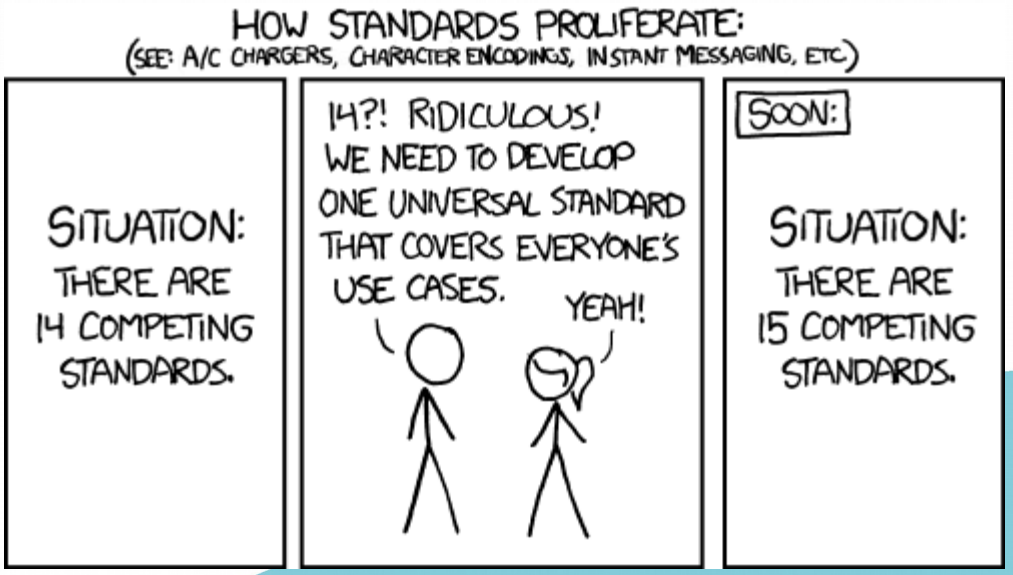


Image: xkcd.com

Automotive standards

- Operating system: OSEK
- Communication buses: CAN, FlexRay
- Ultimate standard: AUTOSAR

Automotive standards

Software should consist of components that can be reused or replaced

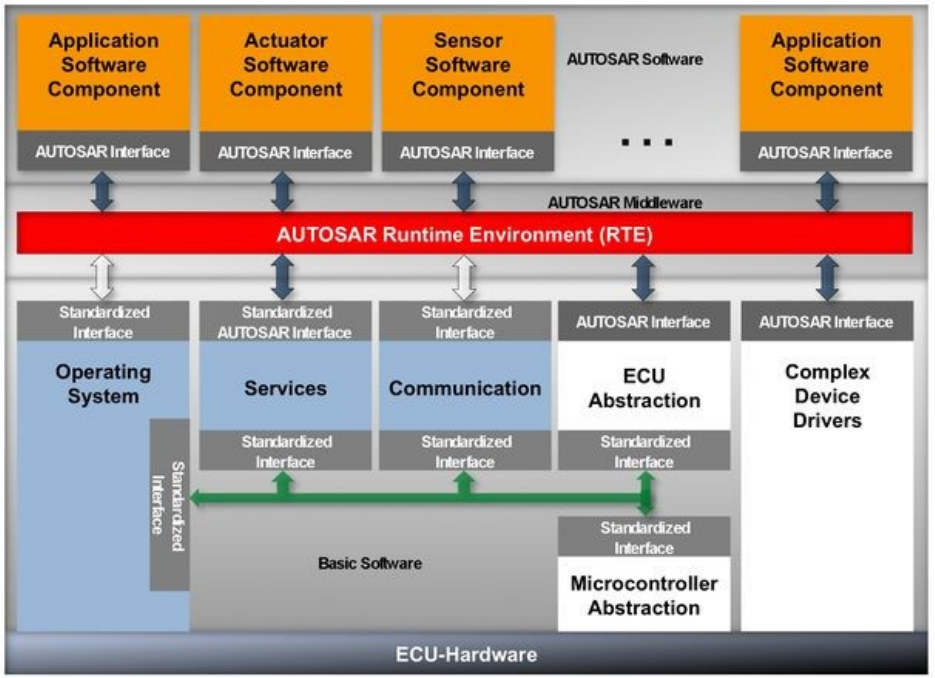


Image: AUTOSAR Consortium

Automotive standards

Automotive software is static!

Automotive standards

Automotive software is static!

- OSEK is a ‘time-triggered’ OS
- In communication, the routes and frames are static
- No memory allocation. AUTOSAR forbids that

Reliablility and robustness

99% is not good enough!

Importance of robustness

Things do go wrong

- Autonomous driving / driver assistance
  - Physical damage to sensors or buses
  - Weather conditions
  - CPU load
- Sensor fusion problems
- Hacking

Importance of robustness

We must avoid false positive actions

- Emergency braking at highway speeds
- Airbag deployment
- Software update initiation
- Many more!

Importance of robustness

We must react to timing errors

- We may have missed sensor data
- We may have failed to react to the driver’s input
- We may have failed to run consistency checks
- We may have broken a modulated signal

ISO 26262

Defines Automotive Safety Integrity Levels (ASIL)

ASIL = Severity * Exposure * Controllability

- Note: controllability means ability to withdraw from the hazard. It does not mean warnings or similar prevention
- ASIL can be achieved with a mixture of hardware and software methods

ISO 26262

Source: ISO-26262

Severity	F = Exposure x Controllability					
	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
S0-No Injuries	QM	QM	QM	QM	QM	QM
S1-Slight and Moderate Injuries	ASIL B	ASIL A	QM	QM	QM	QM
S2-Serious, Including Lifethreatening, Injuries, Survival Probable	ASIL C	ASIL B	ASIL A	QM	QM	QM
S3-Life-Threatening Injuries (Survival Uncertain) or Fatal Injuries	ASIL D	ASIL C	ASIL B	ASIL A	QM	QM

Reproduced from [2]

Hardware methods

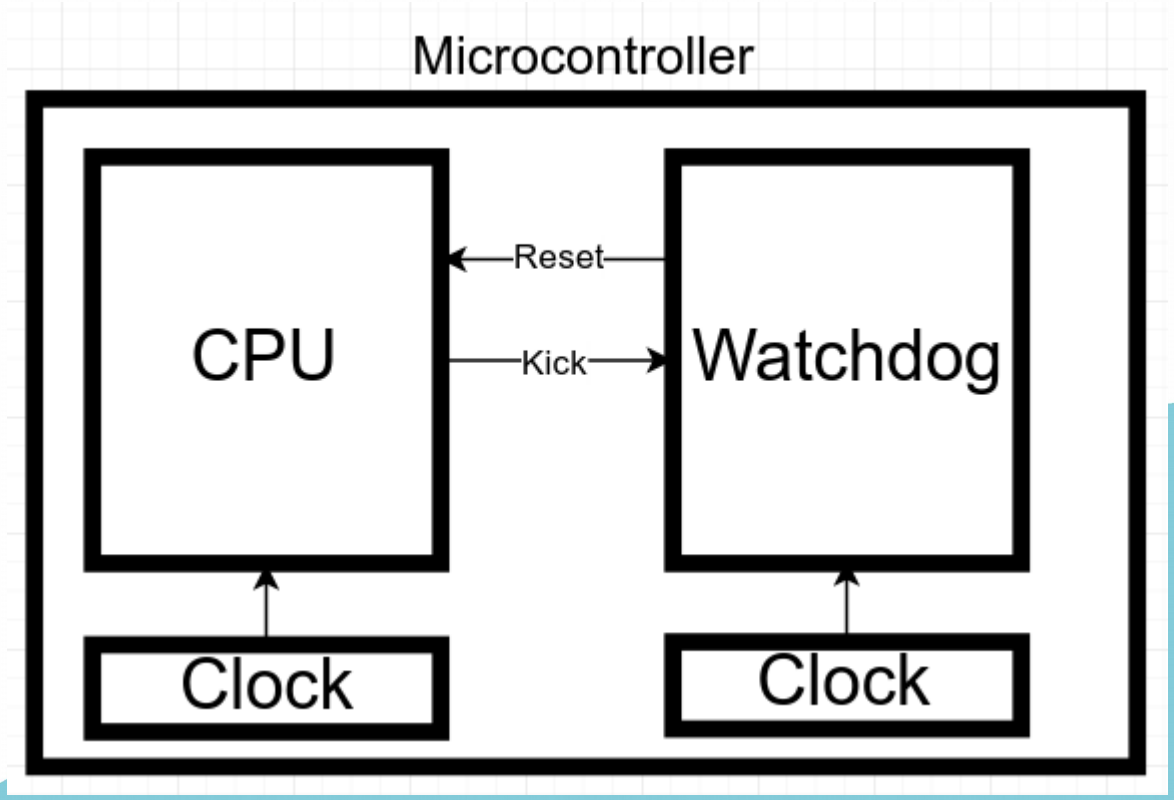
- Physical redundancy.
  - FlexRay has two channels. Used for redunance in a safety context.
  - Multipath I/O with homo- or heterogeneous buses
- Specially designed hardware
    - ECC RAM
    - Lockstep CPUs
    - In extreme cases: triple modular redundancy

Software methods

- Software development follows stringent practices
- Process partially defined by ISO26262
- Unit tests with very high coverage requirements
- Static analysis tools
- Compliance checks with e.g. the MISRA guidelines

Software methods

- Internal watchdog supervision:



Or external: same idea, different microcontrollers

Software methods

Is-alive supervision with a watchdog

- Can additionally be implemented via network messages
- Can also be used for supervision of non-critical components

Execution flow supervision

- Define “checkpoints” in the code
- Checkpoints must be hit in a certain order
- Can also be used for supervision of non-critical components
- Different failure tolerance could be specified

Software methods

Something failed… now what?

Software methods

Something failed… now what?

Software methods

Beyond the usual: triple modular redundancy

- Implement three software components calculating some value, use voting
- Interesting in a multi-ECU case

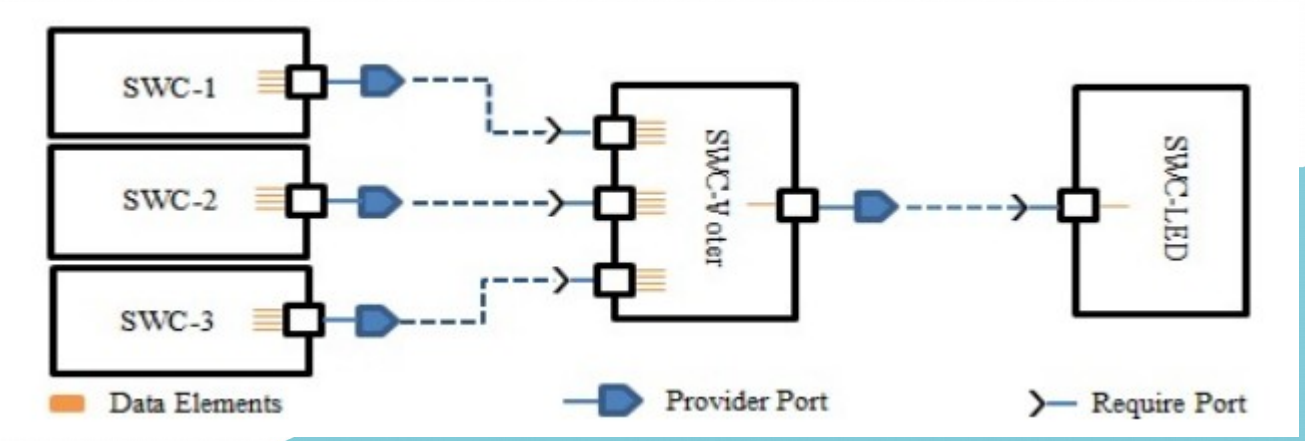


Image reproduced from [3]

Latest challenges

- Increased use of multicore systems
- Including heterogeneous processors – very challenging
- Higher risk of deadlocks
- Preemptive scheduling makes analysis very difficult

Testing in practice

- Run the software
- Run it again
- Run it for longer
- Go back to step 1

Testing in practice

- Long tests with varying data
- Fault injection (mandatory for safety-critical systems)
- Stress testing
- Hardware-in-the-loop tests
- Playback of real captured data for a test system

Testing in practice

- CPU load tends to be very empirical
  - Hooks before/after switching tasks
  - Hooks before/after certain interrupt routines
  - Counters in the idle loop
  - Non-academic, not 100% accurate, but very suitable
- Use the oscilloscope!

Testing in practice

- Lots of time spent on test tools
- Test specifications and test reports are “high-trust” documents

In conclusion - challenges

- Major challenges:
  - Utilizing dual-core systems
  - Working with heterogeneous multi-CPU systems
  - Dealing with the multitude of communications buses
- Tooling challenges:
  - Tools are behind theoretical state-of-the-art
  - All tools have a high cognitive load
- Functional safety is hard, measuring safety is hard

The bleeding edge

- Car-to-car communication, car-to-X communication

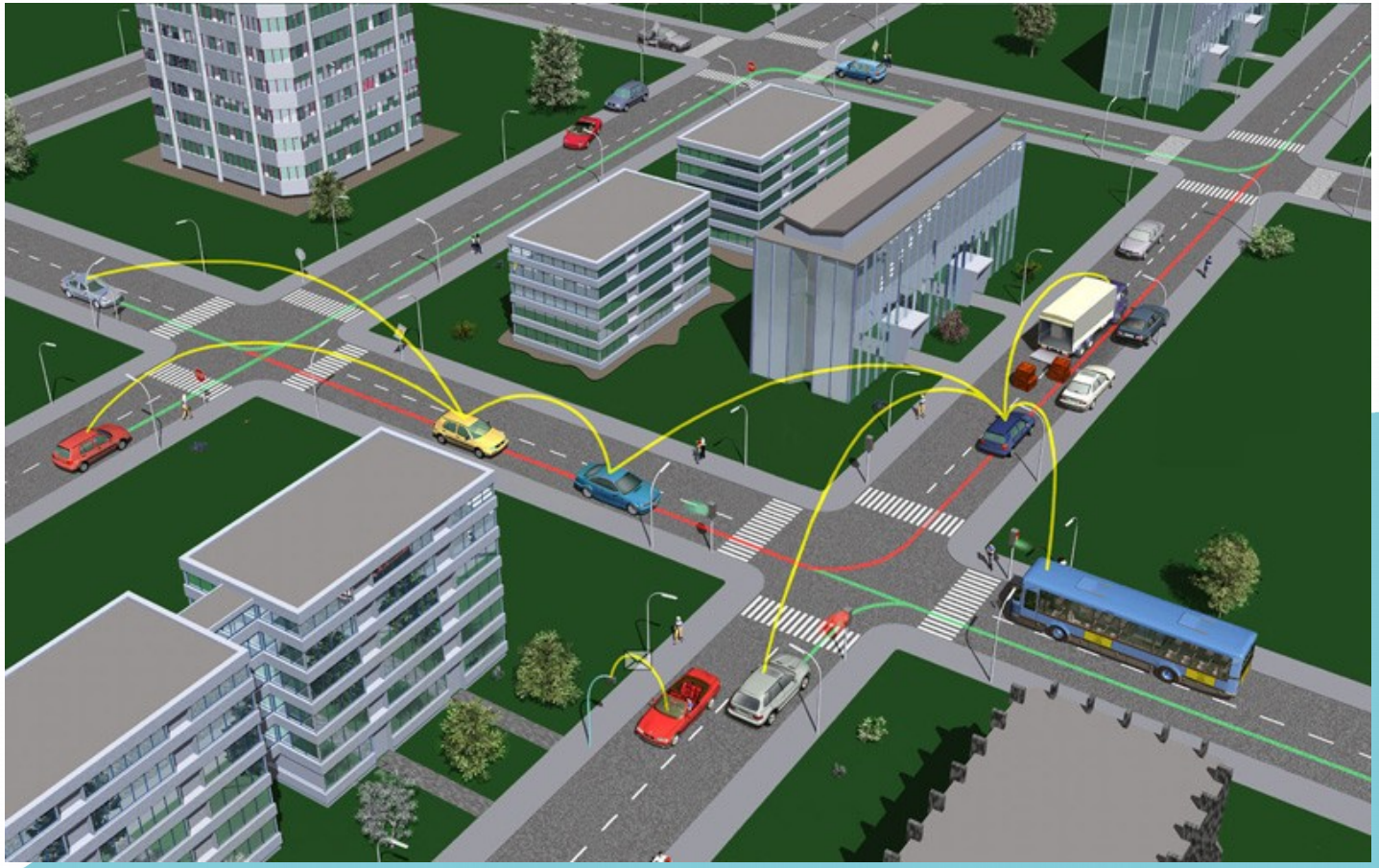


Image: Car2Car Consortium

The bleeding edge

- Car-to-car communication, car-to-X communication

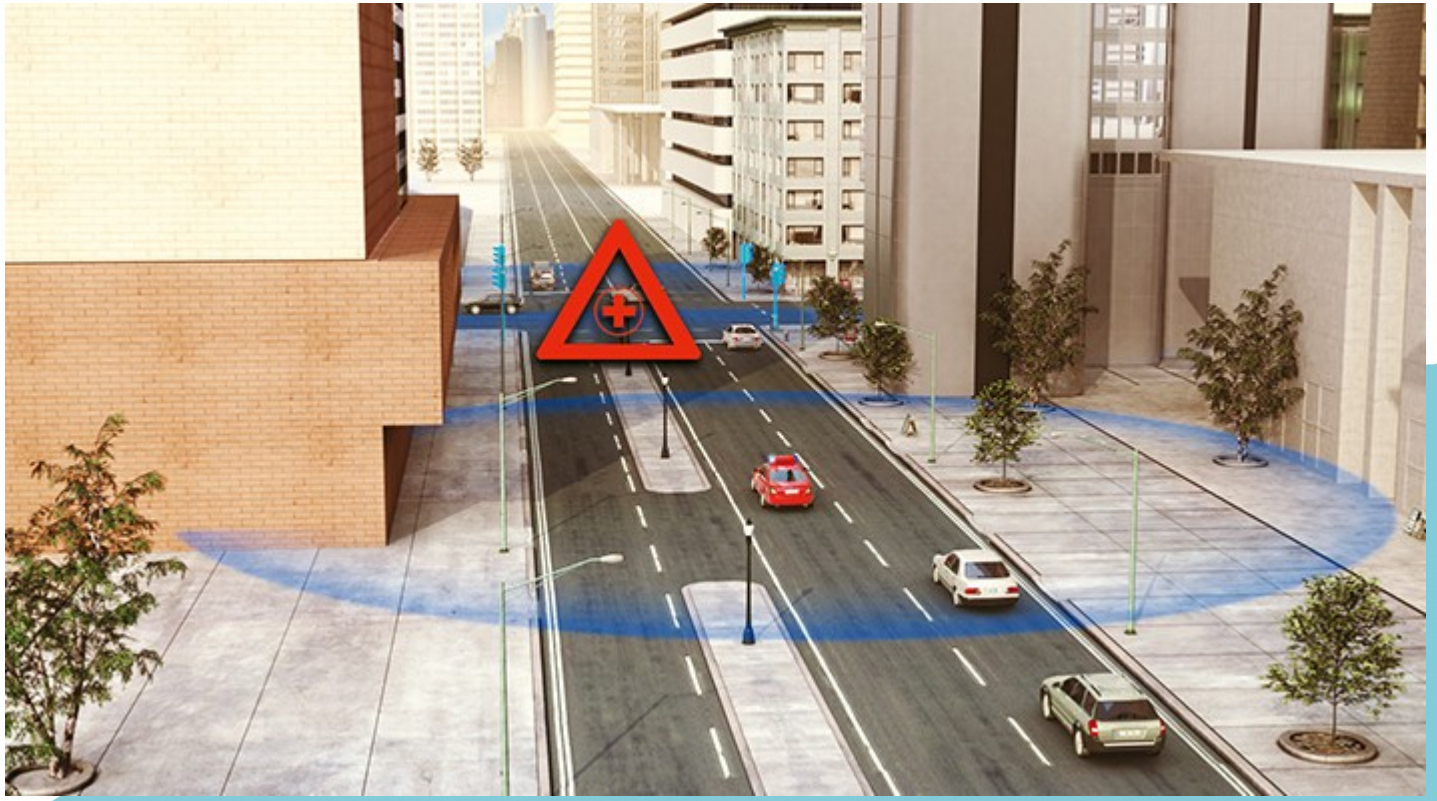


Image: Mercedes Benz

The bleeding edge

- Cadillac CTS 2017 includes car-to-car as a standard feature

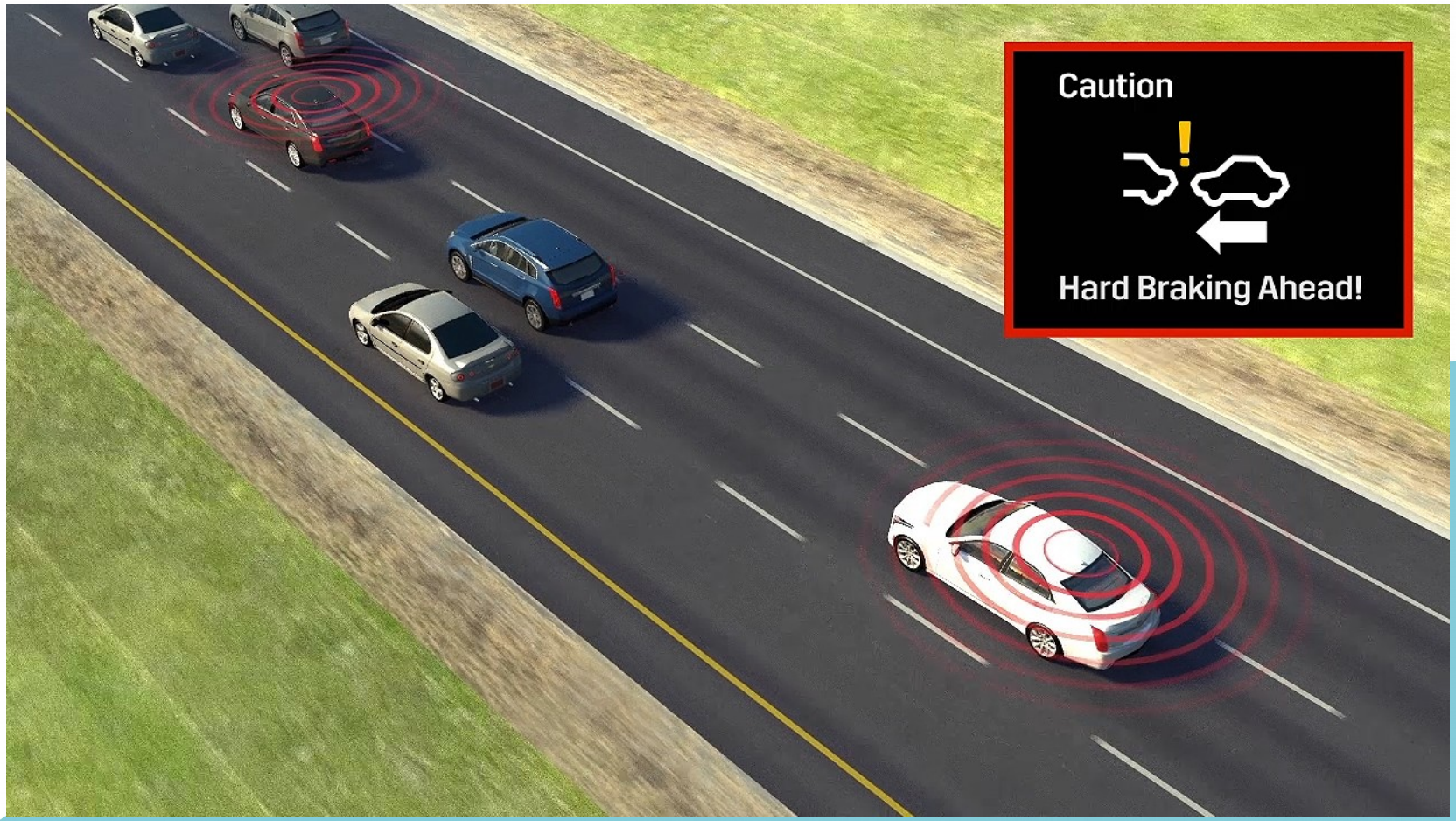


Image: Cadillac

The future

- Improved traffic through intersections
  - See [7] for algorithm based on car-to-car
- Or maybe traffic lights could be eliminated
  - MIT-led study, [8]

The future

- Static solutions will no longer be enough.
- Next-generation Adaptive Autosar in active development
- Virtual machines and hypervisors to be important
- POSIX-compliant automotive OS to be important

The future

Software security is rapidly growing in importance!

Citations

[1] Fausten, M. (2010). Accident avoidance by evasive manoevres. Proceedings of the 4th Tagung Sicherheit durch Fahrerassistenz (TVSD, Munich, April 15–16).

[2] FEV Electronics (2011, March). Functional Safety. FEV Spectrum, 46, pp. 2-3.

[3] S. K. Paul, D. M. Sarwar (2013). A study of Software Implemented Fault Tolerance in AUTOSAR Based Systems. Chalmers University of Technology.

[4] Bertout, A., Forget, J., & Olejnik, R. (2013). Automated runnable to task mapping.

[5] Khenfri, F., Chaaban, K., & Chetto, M. (2015, February). A novel heuristic algorithm for mapping AUTOSAR runnables to tasks. In Pervasive and Embedded Computing and Communication Systems (PECCS), 2015 International Conference on (pp. 1-8). SCITEPRESS.

[6] Sailer, A., Schmidhuber, S., Deubzer, M., Alfranseder, M., Mucha, M., & Mottok, J. (2013, September). Optimizing the task allocation step for multi-core processors within AUTOSAR. In Applied Electronics (AE), 2013 International Conference on (pp. 1-6). IEEE.

Citations

[7] Maslekar, N., Mouzna, J., Boussedjra, M., & Labiod, H. (2013). CATS: An adaptive traffic signal system based on car-to-car communication. Journal of network and computer applications, 36(5), 1308-1315.

[8] Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L. I., Frazzoli, E., Helbing, D., & Ratti, C. (2016). Revisiting street intersections using slot-based systems. PloS one, 11(3), e0149607.

Thank you!

Questions?

Email: daniels.umanovskis@arccore.com

THANK YOU FOR YOUR INTEREST

For more information, please visit www.arccore.com

The future is open
ARC CORE