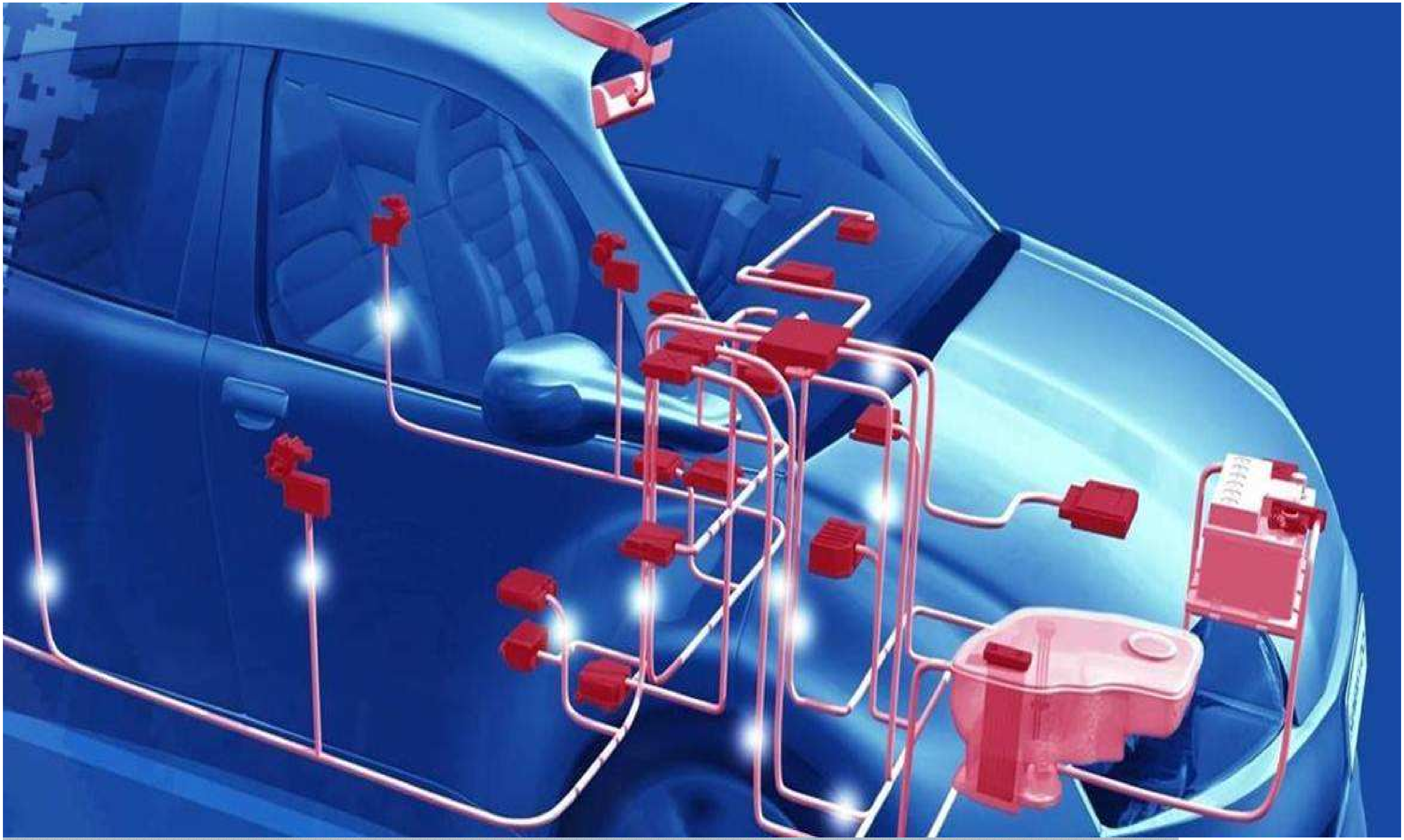
# Real-Time in the Prime-Time
ECRTS 2012






DRIVING | EMBEDDED EXCELLENCE

# ETAS
## Who are we?

ETAS

### ETAS GmbH

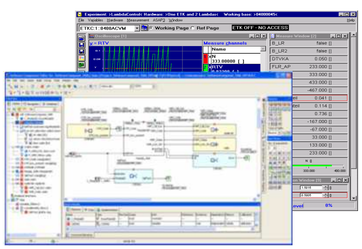
- **Shareholder: 100 % Robert Bosch GmbH**
- **Headquarters: Stuttgart, Germany**
  **15 additional offices worldwide**

### Portfolio – Tools, Services, Consulting

Supporting our customers through **every step** of the embedded software **development process**

### Customers – Vehicle OEMs, ECU suppliers, ...

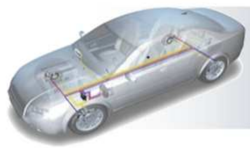
Audi BOSCH Ford FIAT GROUP PSA PEUGEOT CITROËN
GM RENAULT TOYOTA MAN VOLVO NISSAN HONDA
CHRYSLER DENSO 上海汽车工业(集团)总公司 Mahindra MITSUBISHI FUSO
HYUNDAI DELPHI DAIMLER 第一汽车 Valeo Continental








DRIVING | EMBEDDED EXCELLENCE



**An overview of the automotive industry**

**Real-time networking**

**Real-time issues in the ECU**

**What’s next? (Where you can help!)**

# 75,000,000

**Annual worldwide car production**

# 1,100,000,000

**Annual worldwide mobile phone production**

# €42.8 billion

**Annual vehicle export sales**

# 17.1 million


**Vehicles manufactured annually in Europe**

# 12.1 million

**People employed in Europe making them**

# Electronics Dominates Manufacturing Costs

**75%**
**Materials**

**15%**
**Labour**

**15%**
**Other Expenses**

**ELECTRONICS**

**15%**
**Mid-range**

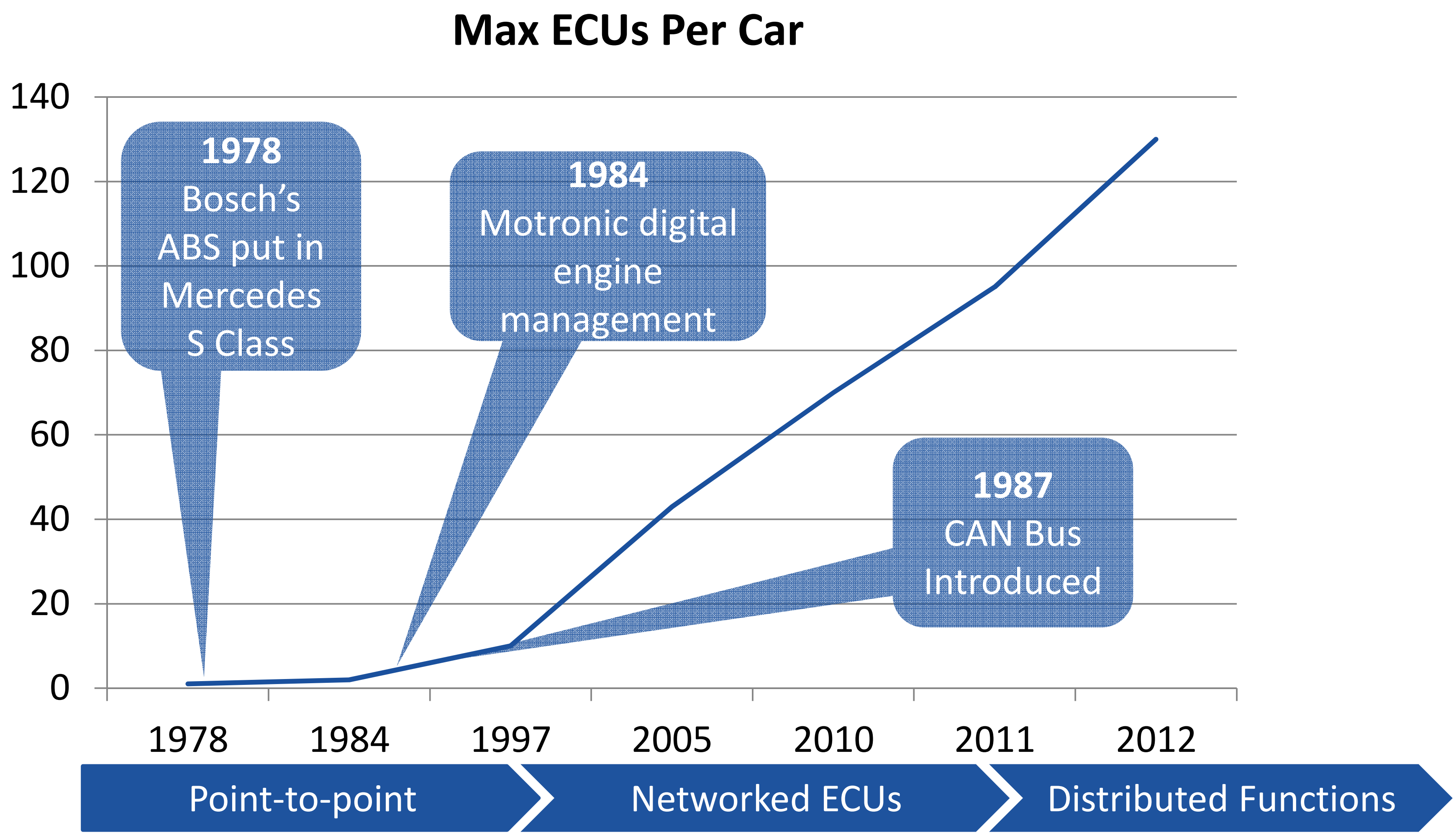
**30%**
**High-End**

**40%**
**2015 Estimate**

**Sources**
McKinley, 2010

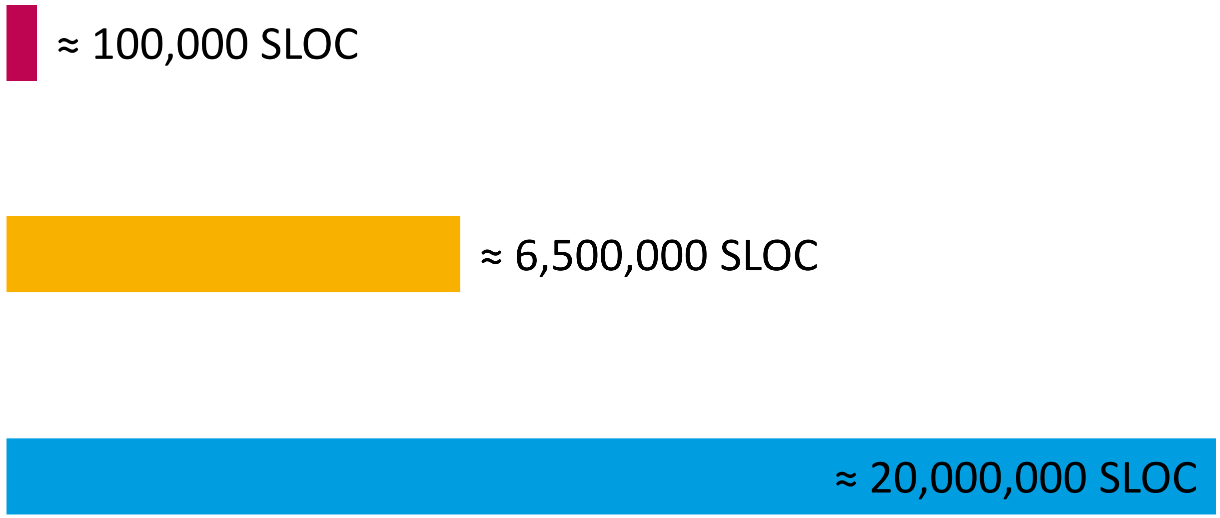
# The Modern Car

## How much electronics?



# The Modern Car

## How much software?





**Sources**
Pavey & Winsborrow, “Demonstrating Equivalence of Source Code and PROM Contents”, Computer Journal Vol 36, No 7, 1993
Charette, “This car runs on code”, IEEE Spectrum, Feb 2009

# Vehicle Domains: Powertrain
## (Or what does all that stuff do?)

ETAS

- Engine Management
  - Injection/Spark timing
  - Emissions control
- Transmission Control
  - Gear selection
  - Terrain Adjustment
- Real-time issues
  - Pressure wave control on diesel engines
  - Deadlines a function of angular rotation
  - Lots of data communication

# Vehicle Domains: Chassis
(Or what does all that stuff do?)

- Braking
    - Anti-Lock Braking (ABS) since 1978
- Traction Control
    - Electronic Stability (ESP) since 1995
- Steering assist
- Adaptive cruise control

- Real-time issues
    - Wheel rotation driving
    - Slip detect
    - Brake force distribution

# Vehicle Domains: Body
(Or what does all that stuff do?)

- Wiper control / rain sensing
- Wing mirrors
- Vehicle access
- Window lift/anti-trap/pinch
- Electronic seats
- Heating/ventilation
- Park pilot
- Lane departure warning
- Airbags
- Blind spot warning

- Real-time issues
  - End-to-end latency guarantees
    - e.g for brake lights
  - Distributed functionality

**Image:** ATZ Online

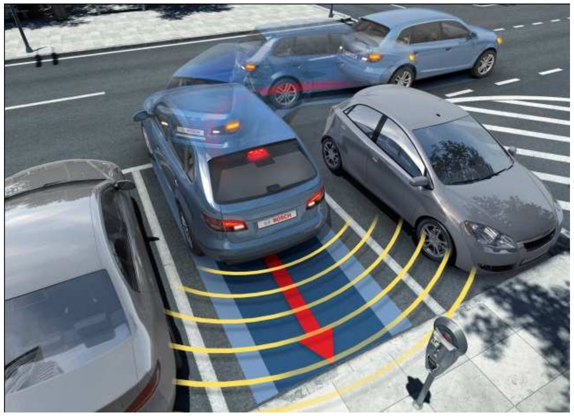


**Image:** Robert Bosch GmbH

# Vehicle Domains: In-Vehicle Infotainment (IVI)
## (Or what does all that stuff do?)

- Radio/CD/MP3 integration
- Navigation/Mapping
- TV
- Internet
- Telephony

- This area accounts for an increasing part of the "user experience"

- Real-time issues
  - Quality of service similar to those for similar "PC" applications

# Automotive Software Development

## Who does what?

ETAS

- Car Makers (the OEM)
  - System integrator
  - Responsible for network scheduling
  - Maximize re-use of systems Between vehicles
- ECU suppliers (Tier1s)
  - ECU integrator
  - Responsible for ECU scheduling
    - Application constraints
    - Network constraints
  - Minimize bespoke development
- And all of this under MASSIVE cost pressure

**OEMs**

**Tier1s**

**…and many more**

- Long standing desire to harmonize automotive SW architectures
  - ~1995 OSEK/VDX
  - ~2003 AUTOSAR
- AUTOSAR is the "new big thing"
  - Distributed communication framework for automotive systems
- AUTOSAR means
  - More collaboration
  - More integration
  - More interest in timing issues

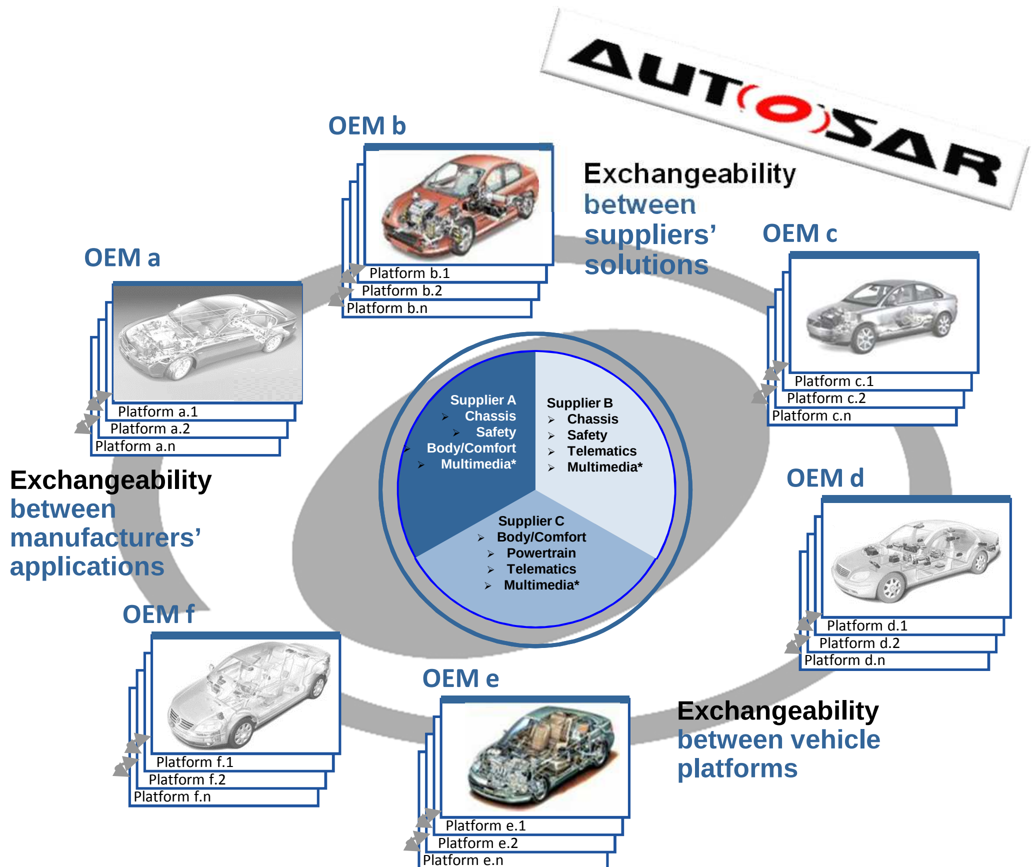
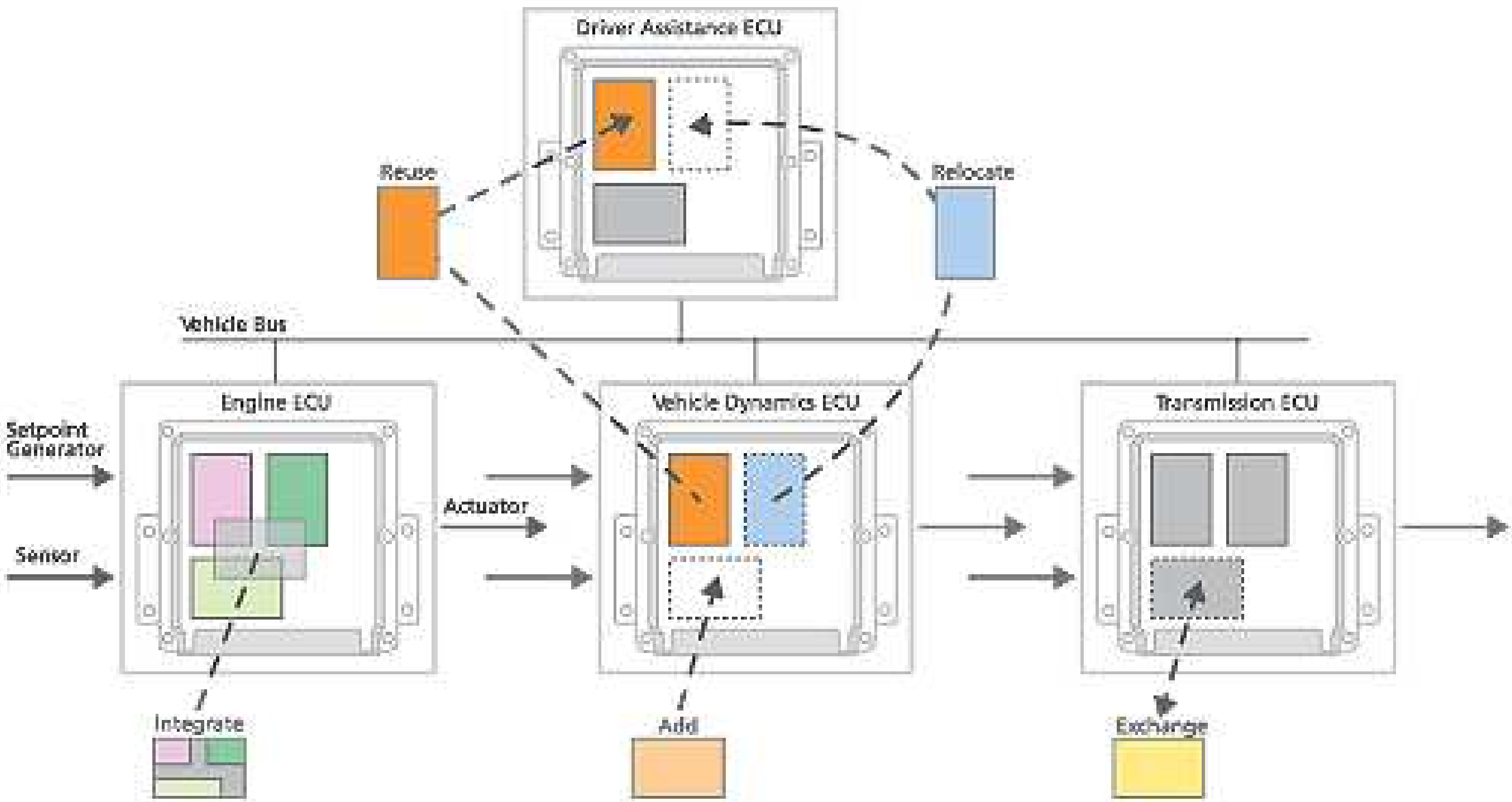


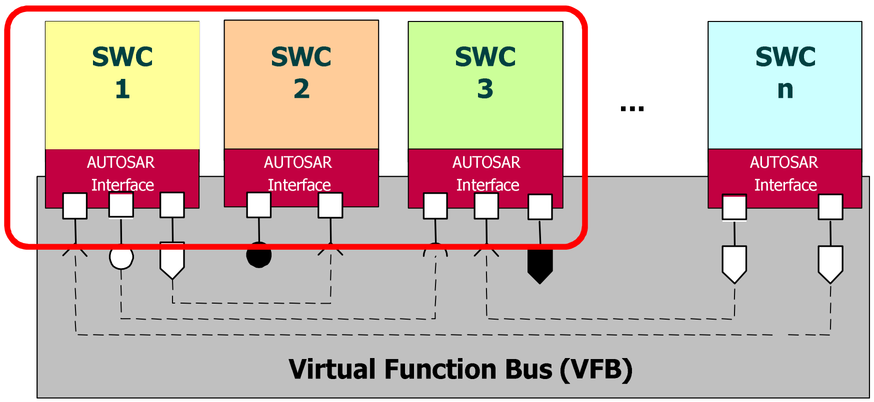

**Image:** AUTOSAR GdR

# AUTOSAR
## Freedom to be flexible



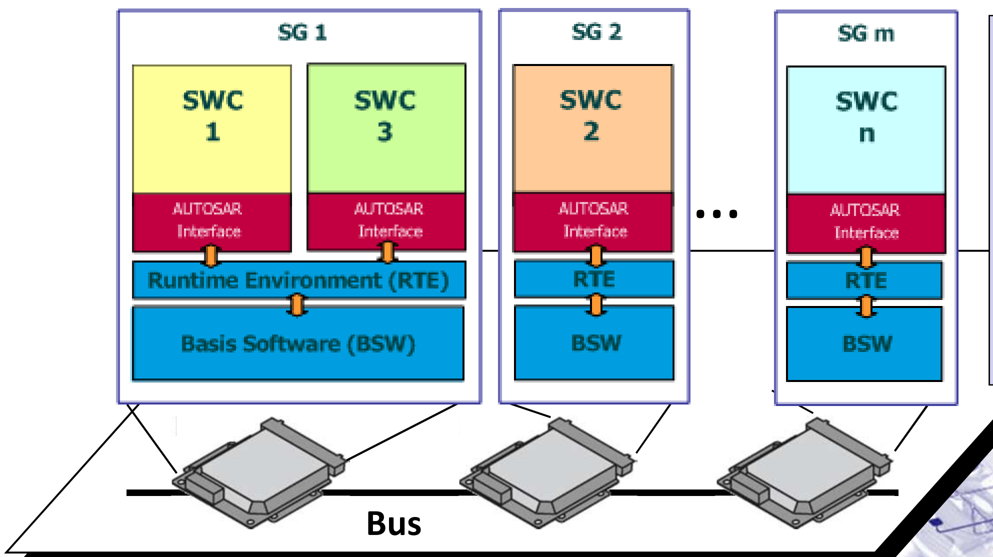

 

ETAS




### Standardized Software Architecture

- Software Components (SWCs) as building blocks
- Virtual Function Bus (VFB) defines communication model
  - Abstraction that makes software components portable
- Users define “what” but not “how”




### Standardized Basic Software

- Run-Time Environment (RTE) encapsulates VFB
- Standardized OS, communication model, memory model, device drivers, etc.
- All APIs are defined
- All functionality is specified

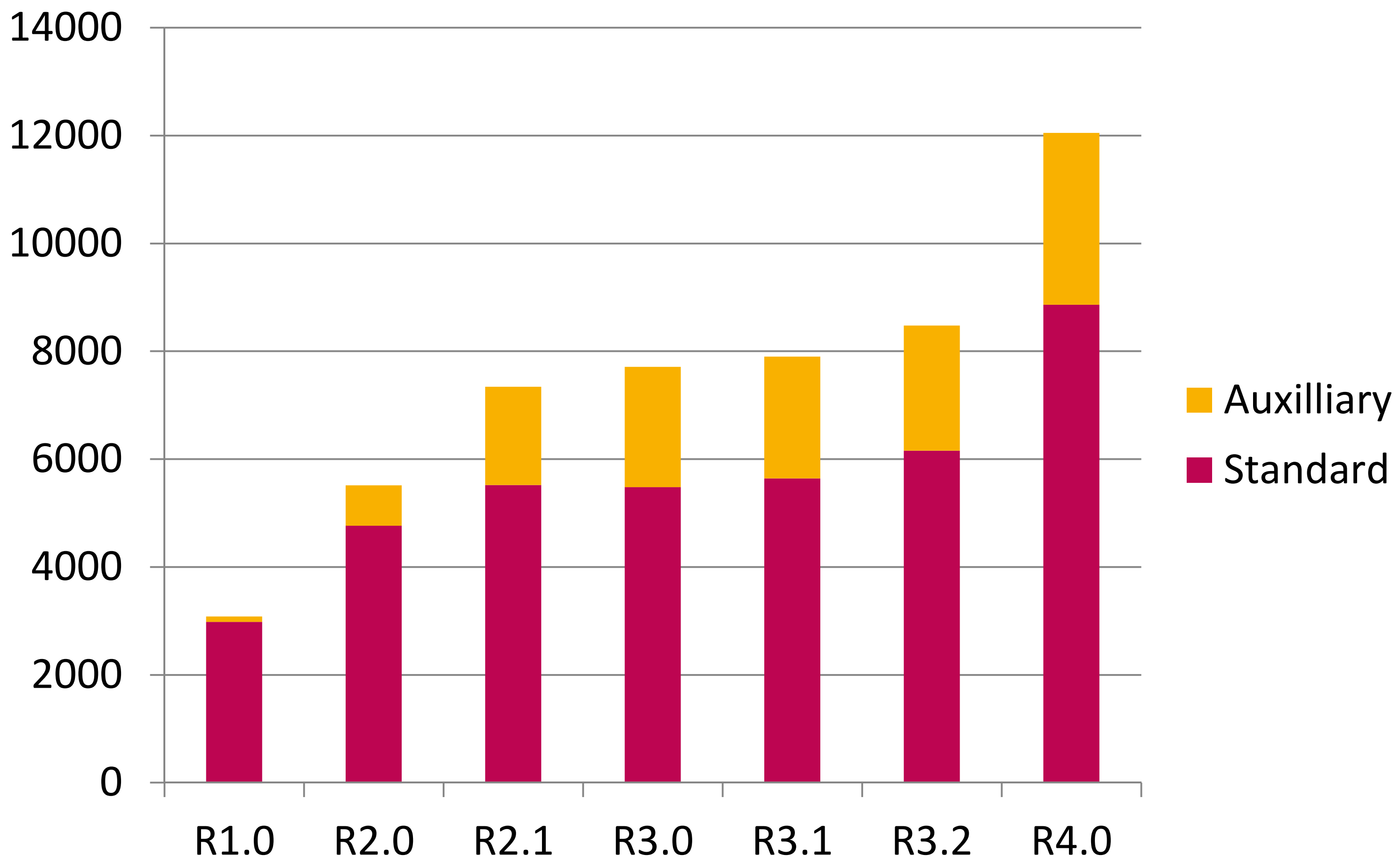
### Timing Aspects

- Capture of timing parameters in configuration
  - Timing chains to cover end-to-end
- AUTOSAR OS provides temporal protection

DRIVING | EMBEDDED EXCELLENCE

# AUTOSAR
## Size of releases (By pages of specification)



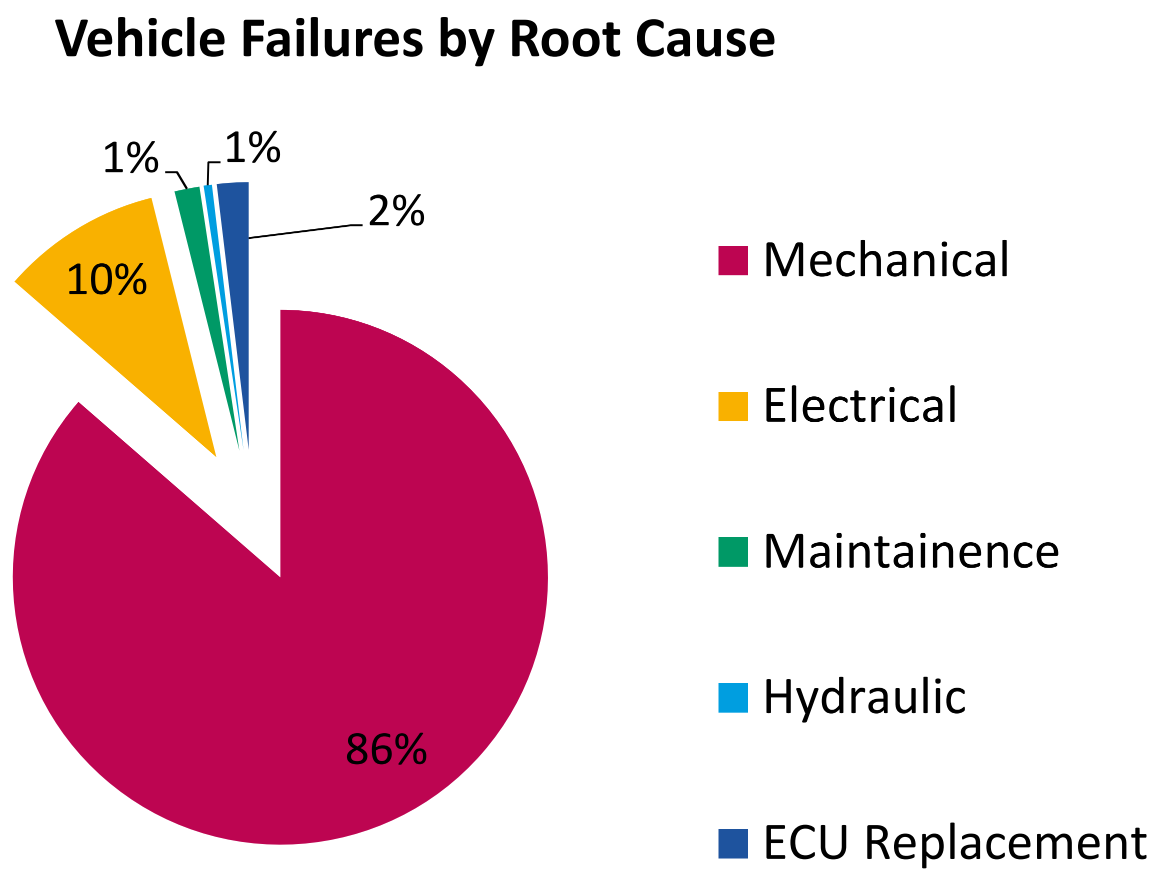
# Why are we interested in scheduling?

## Exceptional Reliability Demands

| | |
|---|---|
| **Cars** | • 400,000 vehicles per year |
| **Hours** | • 7 hours driving per week |
| **Years** | • On road for 20 years |
| **Use** | • 2,912,000,000 hours |

**1000x more "flying hours" than the entire Boeing 737 fleet has made since 1968**

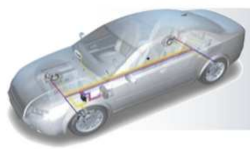
# How often does the software break?




Source: Mike Ellims, “On wheels, nuts and software”

An overview of the automotive industry



Real-time networking

Real-time issues in the ECU

What's next? (Where you can help!)

## Point-to-point wiring until the 1990s

- Expensive to fit
- Heavy
- "Cheap" connectors are major fault source

## Networked ECUs ease the problem

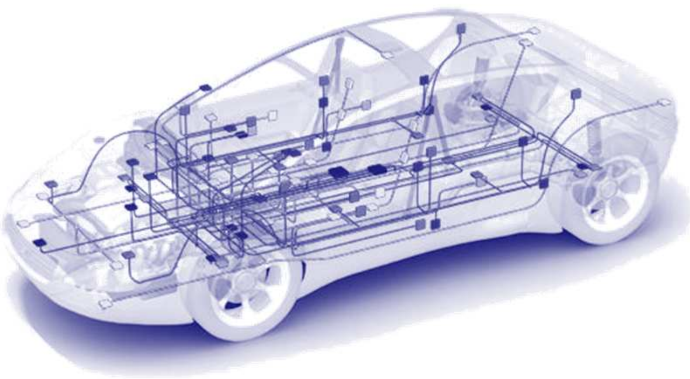
- Less wire = lighter
- Fewer connections = reduce fault sources

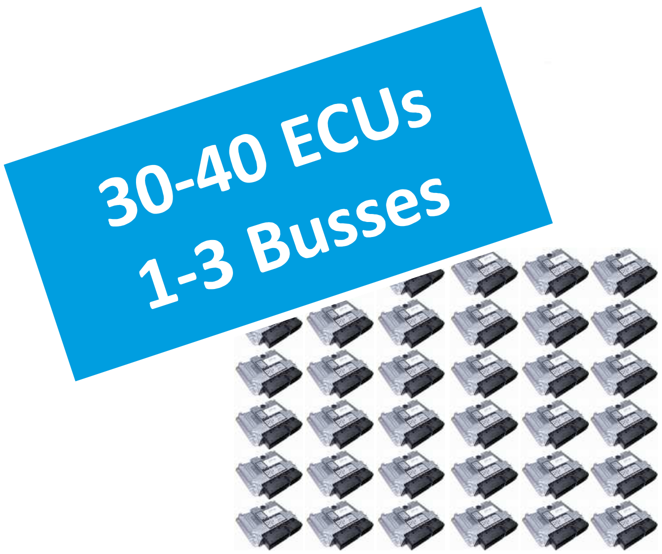
## But wiring remains a challenge

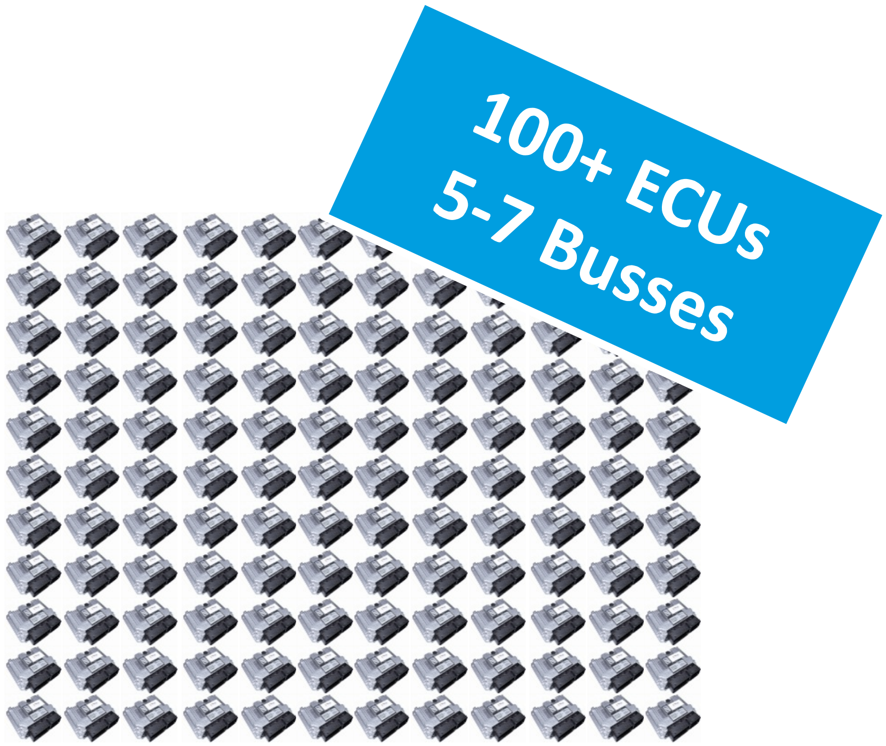
- 3rd most expensive part of a car
- 3rd heaviest part

# Electronics Architectures





ETAS

# "Classic Domains"

LOCAL INTERCONNECT NETWORK

19.2 kbit/s

50x Faster

Up to 1Mbit/s

10x Faster

Up to 10Mbit/s

# "Multimedia/Infotainment"

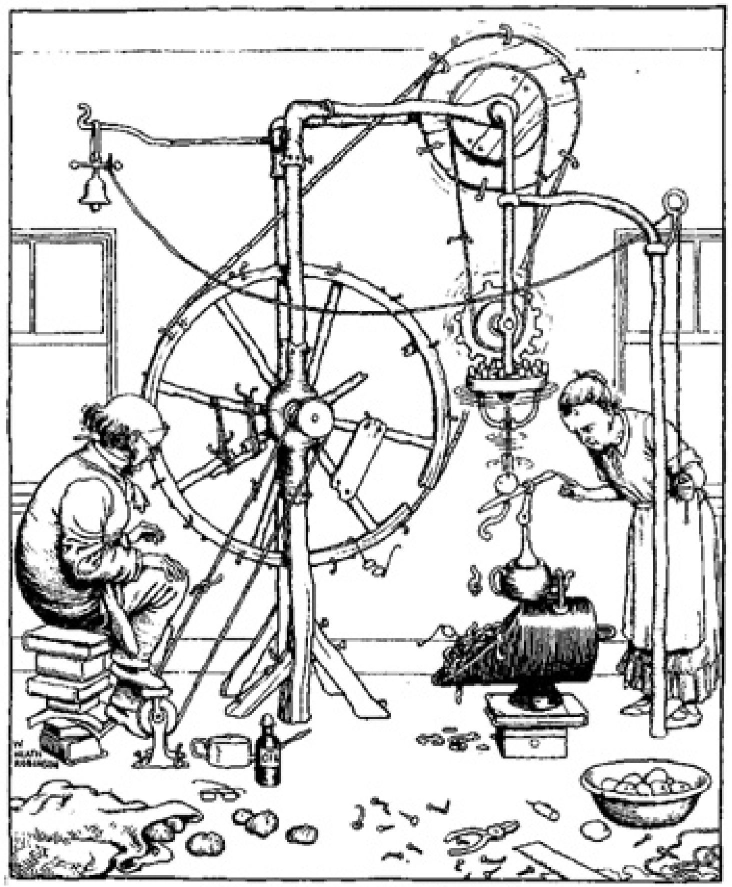
# And gateways between them all

# "You cannot run CAN bus reliably at more than 35%* bus utilization"

† Meme: something believed, but not true
* Figure may vary, but not significantly

# …and impacts for the use of response time analysis



The Professor's invention for peeling potatoes.

**Legacy of ad-hoc "solutions" to make CAN scheduling "work"**

**+**

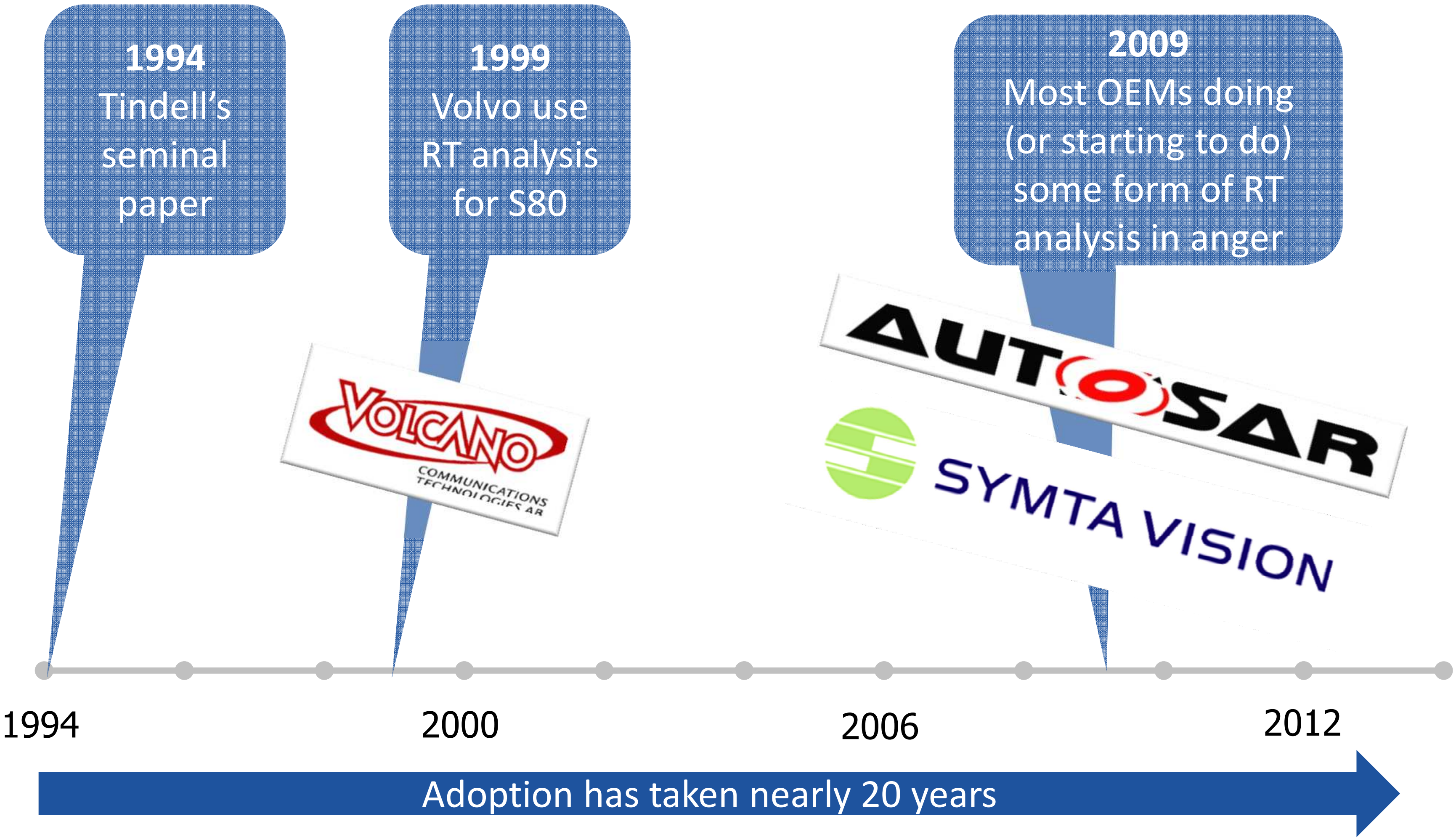
**Seldom a "clean sheet" for the network design**

**=**

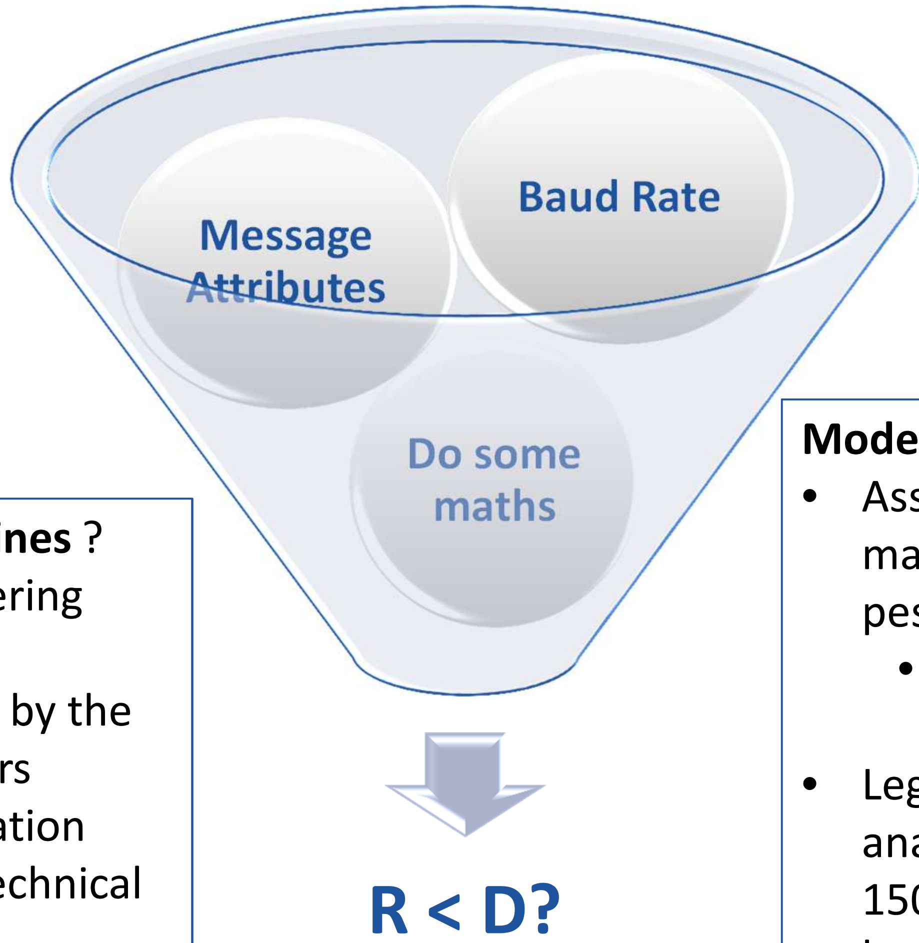
***Existing* timing behavior can be hard to model with acceptable pessimism**

# Adoption Timeline



# Challenges




**What are the deadlines** ?

- A system engineering issue
- Often not known by the network engineers
- Business organization problem: not a technical one

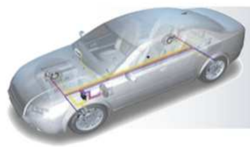
**Modeling**

- Assuming minimum period makes analysis too pessimistic
  - Offsets, event models are essential
- Legacy network models analyses are often for 150% of the vehicle (e.g . both petrol and diesel engine versions)

An overview of the automotive industry

Real-time networking



Real-time issues in the ECU

What's next? (Where you can help!)

# Memory

- 4MB ROM/256kB RAM is "huge"
- 256kB ROM/32kB RAM is "typical"


# Speed

- 280MHz is "fast"
- 40MHz is "typical"

ANALOG DEVICES
freescale semiconductor
ST
FUJITSU THE POSSIBILITIES ARE INFINITE

# Harsh environment

Infineon Never stop thinking
RENESAS Everywhere you imagine.
MICROCHIP
NEC
TEXAS INSTRUMENTS

# Significant number of sporadics

**Many applications are inherently event driven**
**Short (<100us) explicit deadlines**

# Many functions in few tasks

**50-300 functions per task is common**
**Varying WCET: from <1us to 100+us**

# Lots of shared, global, data

**Thousands of messages typical in powertrain**
**Asynchronous communication**

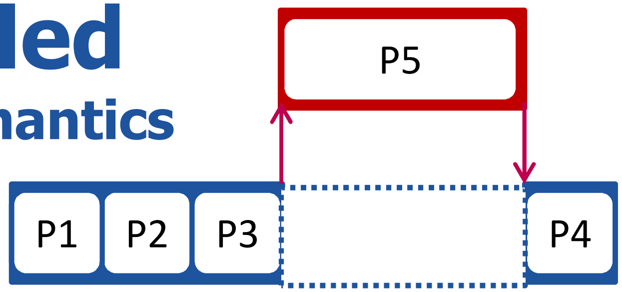
# Wide use of OSEK/AUTOSAR OS

**Small footprint , low overhead**

**ETAS puts 1,000,000 OSs *per week* on the road**


# Data intense applications often co-operatively scheduled

**With state-based communication semantics**

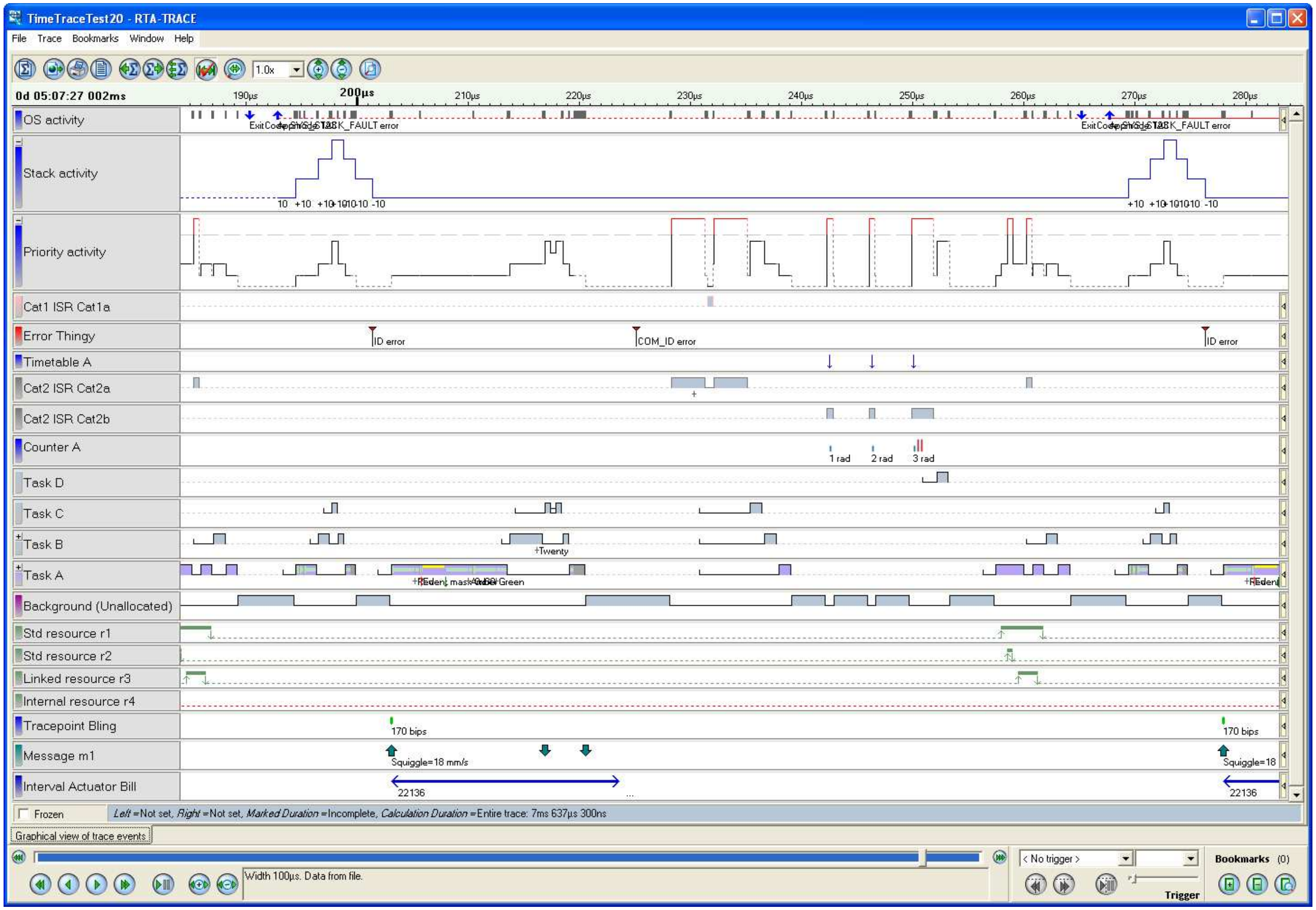



# Cyclic legacy in design

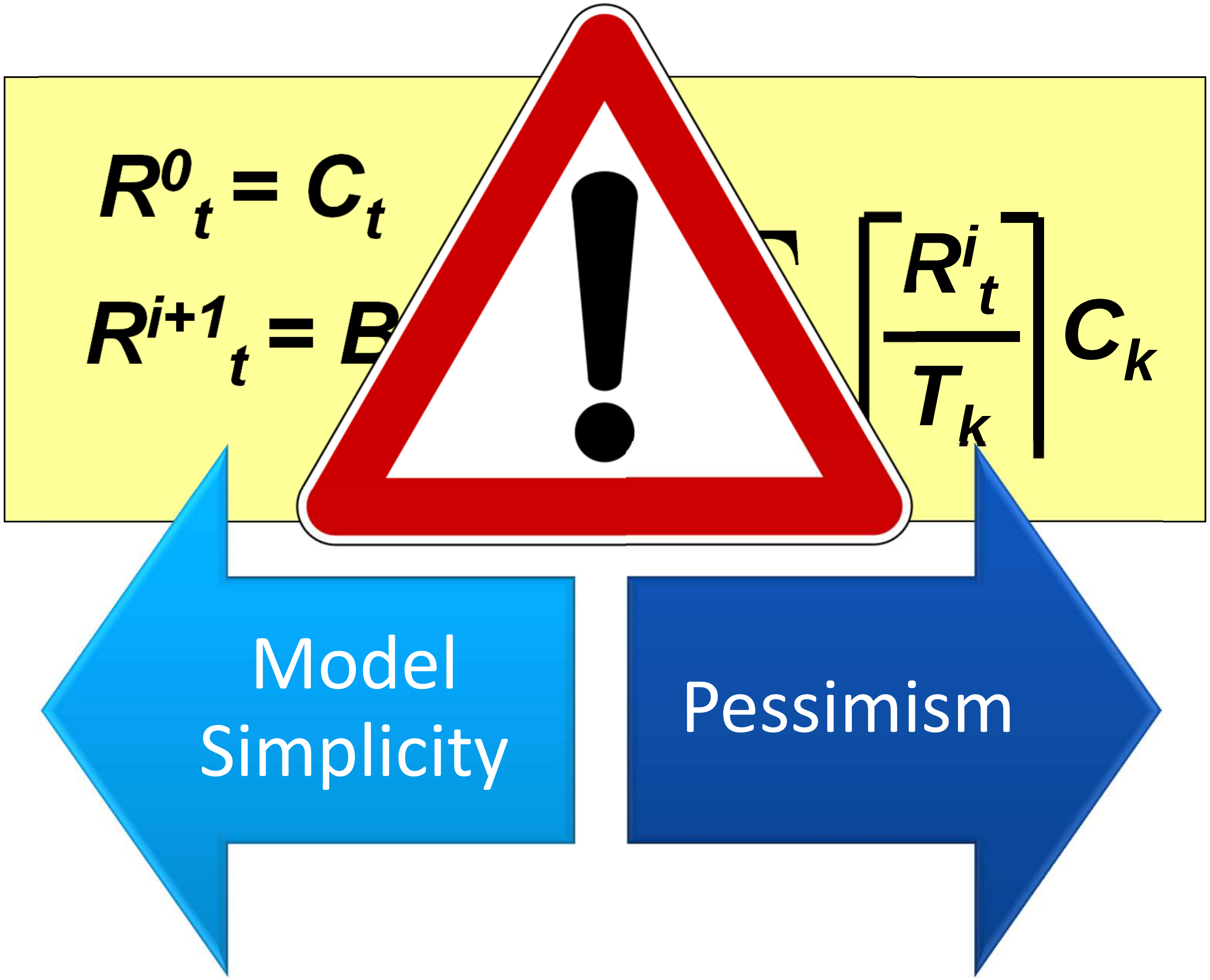
**1,2,5,10,20,50,100,1000ms periods**

# OSEK OS in Action





R0t = Ct
Ri+1t = B
Rit
Tk
Ck
Model Simplicity
Pessimism

# DMA + Offsets + Event Models

**This is “~~state-of-the-art~~” “state-of-the-practice”**

# In daily use at some Tier1’s

**Goal: Avoid costly “surprises” during test**

# Transparent to working engineers

- Typically lots of mode dependent behavior
  - Both ET
  - And Periods
  - With uncorrelated WC peaks

- Randomization of periods

- Many, many things in AUTOSAR are implemented on top of task self suspension

```
TASK(Variant_Execution){
  f1();
  f2();
  if (rpm < 4000) {
    f3();
    f4();
  }
  f5();
}
```

# Summary

Benefits of schedulability analysis are clearly seen

Putting it into industrial practice can be hard

There is seldom a "clean sheet"

People want to know about what they've already built first

Important to show the result & how it relates to reality

- Schedulable: Yes/No isn't enough
- "What sequence of activations happens before it breaks?

An overview of the automotive industry

Real-time networking

Real-time issues in the ECU

What's next? (Where you can help!)

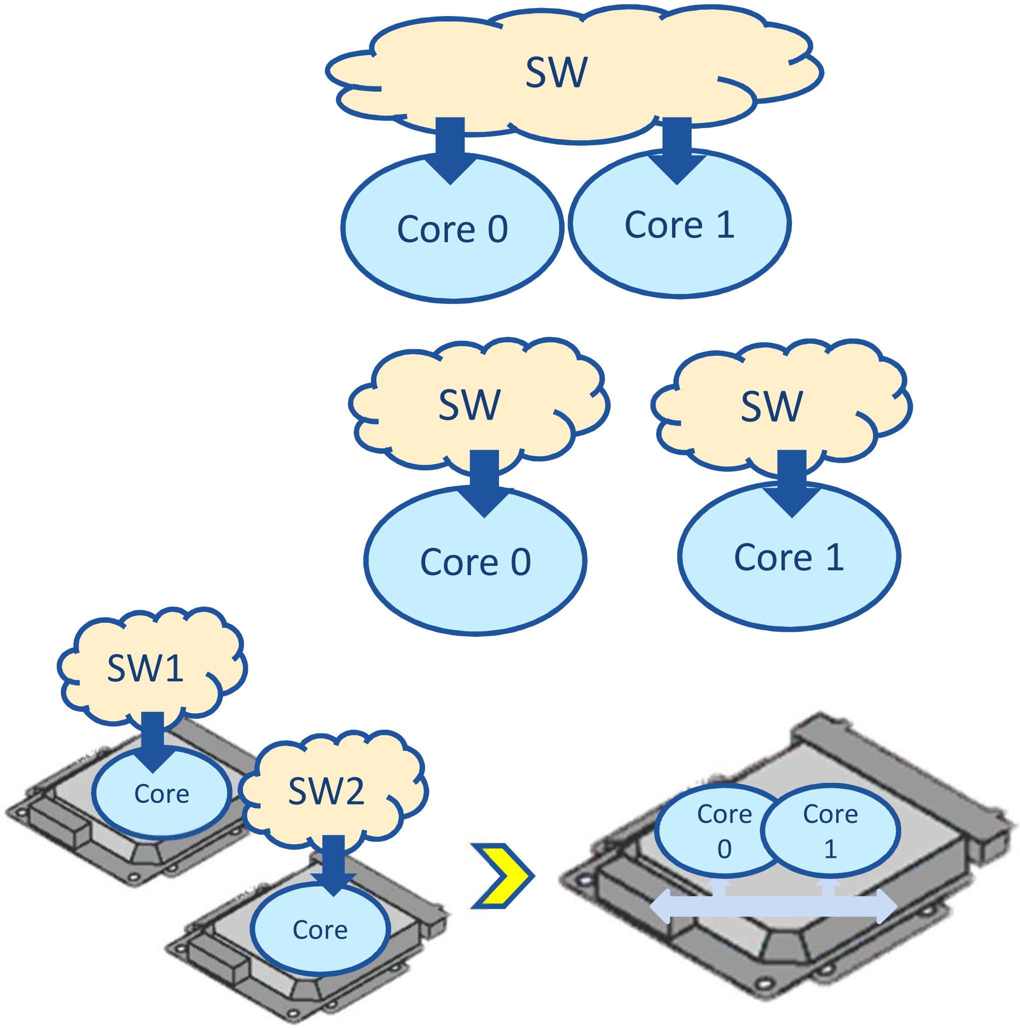
 

# Challenge#1: Multicore Systems
## What are people using this for?

- **Case 1: More computing power**
  - **Same thermal requirements, but more can be done**

- **Case 2: Redundancy in case of failure**
  - **Dual cores in lock-step**
  - **Or monitor on 2nd core**

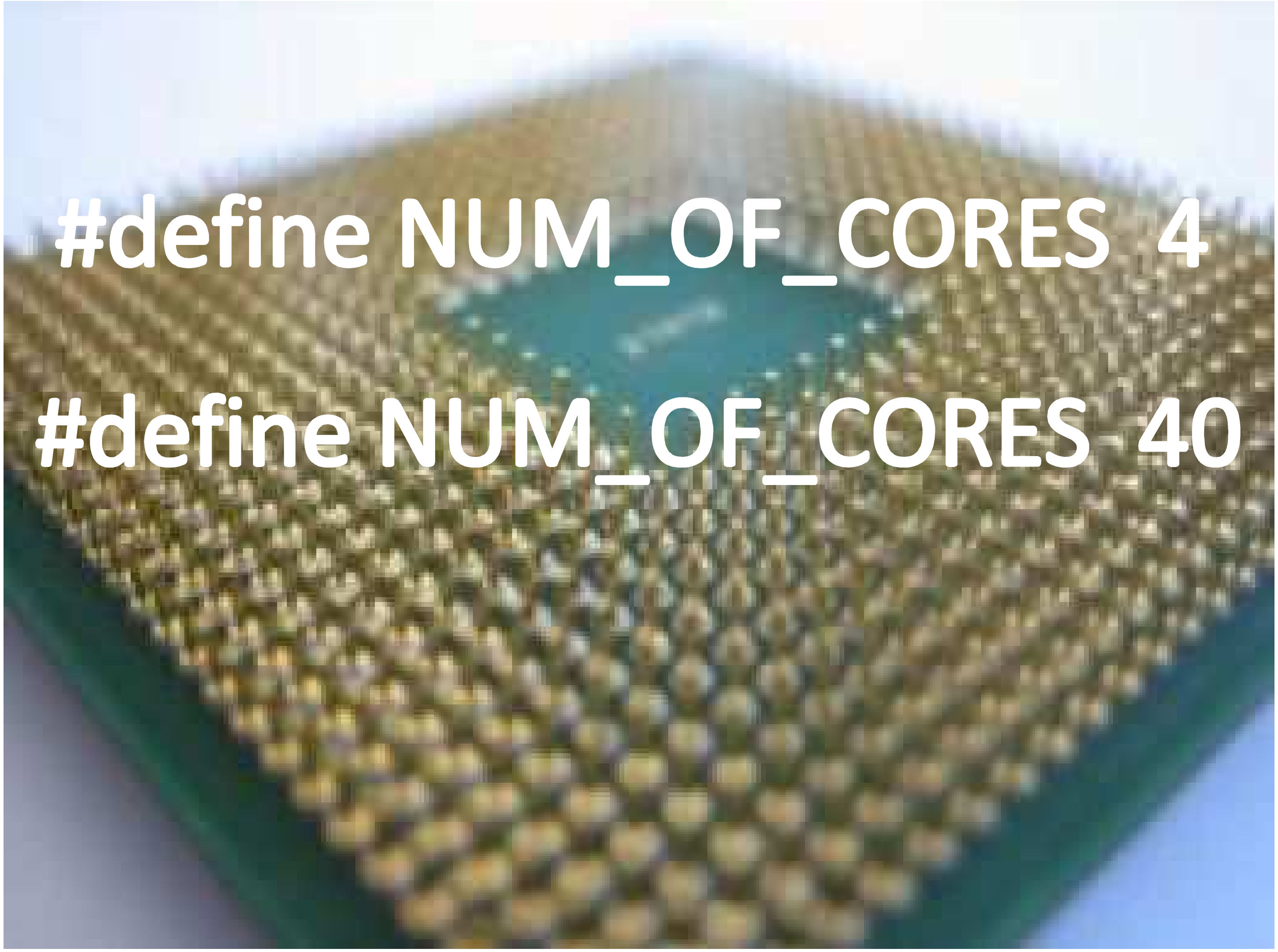
- **Case 3: Aggregation**
  - **Combine smaller ECUs into one multicore ECU**





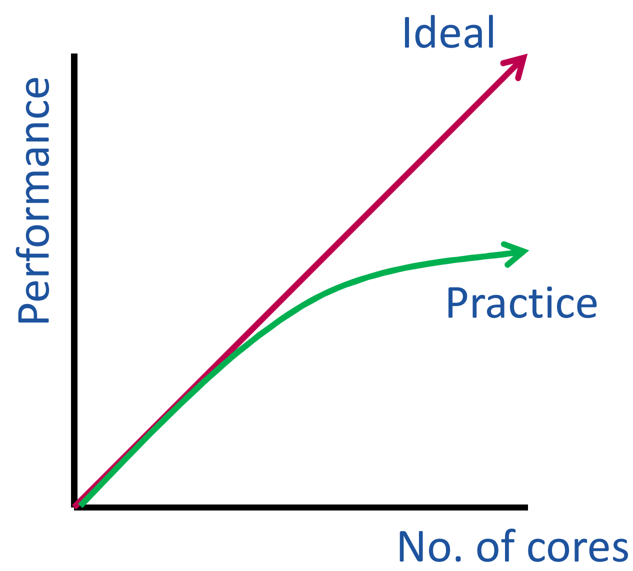


DRIVING | EMBEDDED EXCELLENCE


#define NUM_OF_CORES 4
#define NUM_OF_CORES 40

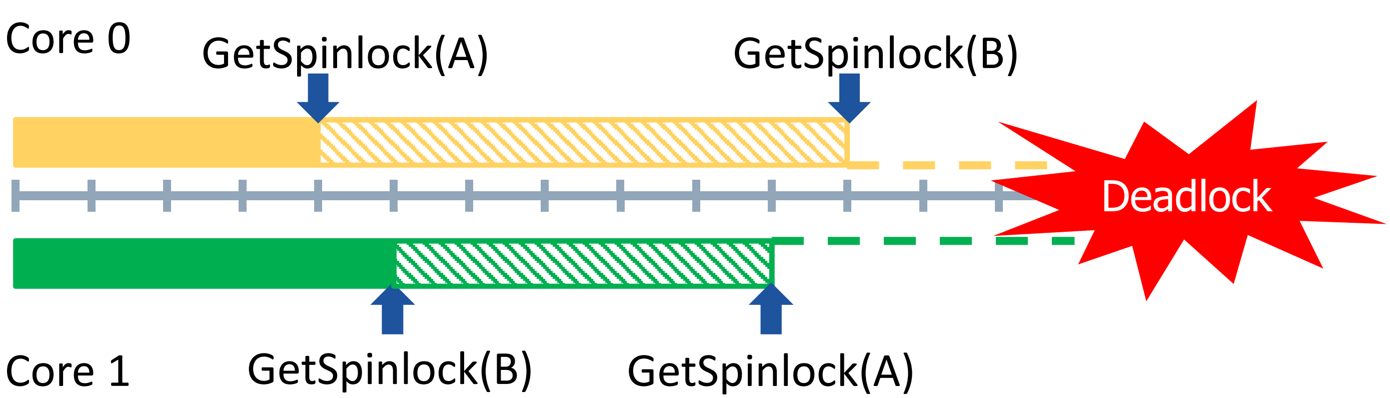
# Challenge#1: Multicore Systems
## Performance Assumptions

- **Mention multicore and users assume dynamic task migration**
  - **Probably too expensive for this systems with automotive characteristics**
- **They also assumes linear performance gains**
  - ***N* cores => *N* * single core performance**
- **The real world has proven to be a disappointment**
  - **Communication delays**
    - Due to HW bus contention, OS spinlock contention
  - **Timing issues**
    - Algorithms are not perfectly parallel
  - **OS overheads**
    - API calls take longer



# Challenge#1: Multicore Systems
## Old Problems, Re-introduced

- AUTOSAR OS R4.0 provides a standardized multicore OS
  - OS objects are statically allocated to cores
  - Each core runs an independent scheduler
- Some dubious design choices for critical sections between cores
  - Spinlocks and (optional) deadlock avoidance mechanism means potential for deadlock



# Why not just have lots of computing power and use virtualization?

# Already happens in infotainment

# But how do we make that work for hard real-time & mixed criticality system?

# Challenge #3: Distribution beyond the car

- Next generation of innovation means cars interacting
- With ”the internet”
  - Navigation influencing power train/chassis
  - Road conditions from telemetry
- With each other
  - Local traffic flow control
  - Emergency braking

# Challenge #3: Distribution beyond the car

Coming Soon to a vehicle near you…


**Image:** Google Inc.

# Thanks for listening!