

# Eingebettete Systeme

Prinzipien und Komponenten  
Eingebetteter Computer Systeme

Wintersemester 2005/2006

Prof. Dr. Jörg Kaiser  
Embedded Systems and Operating Systems (EOS)

# Allgemeine Information

**Dozent:**

Prof. Dr. Jörg Kaiser  
Institut für Verteilte Systeme (IVS)  
Arbeitsgruppe Eingebettete Systeme und Betriebssysteme  
Geb. 29 Zimmer 323  
kaiser@ivs.cs.uni-magdeburg.de

**Sekretariat:**

Petra Duckstein  
29 Zimmer 405  
duckstein@ivs.cs.uni-magdeburg.de  
67 18345

**Übungsgruppenleiter:**

Michael Schulze,  
Institut für Verteilte Systeme (IVS)  
Arbeitsgruppe Eingebettete Systeme und Betriebssysteme  
{kiebel, mschulze}@ivs.cs.uni-magdeburg.de

# Inhalt

- ➔ Einführung: Was ist ein eingebettetes System?  
Was macht den Unterschied aus?  
Was muss man können?
- ➔ Sensoren, Aktoren und ihre  
Unterstützung durch Funktionseinheiten  
in Micro-Controllern  
Analoge Schnittstellen  
Zeitgeber und Zähler  
Unterbrechungsbearbeitung
- ➔ Zuverlässigkeit und Fehlertoleranz
- ➔ Zeitgerechte Ausführung
- ➔ Betriebssysteme für eingebettete Systeme

# web Ressourcen

[http://ivs.cs.uni-magdeburg.de/eos/de/lehre/ws0506/vl\\_pkes](http://ivs.cs.uni-magdeburg.de/eos/de/lehre/ws0506/vl_pkes)

Empfehlung für Vorlesung und die Übungen:

Erzähle mir und ich vergesse.  
Zeige mir und ich erinnere.  
Laß mich tun und ich verstehe.

Konfuzius, 551 - 479 v.C.

# was ist ..? ein erster Definitionsversuch

## Eingebettetes System

Ein Artefakt mit einer informationsverarbeitenden Komponente, welche die Funktionen des Artefakt über eine gewohnte oder angepasste Benutzerschnittstelle unterstützt und erweitert.

oder

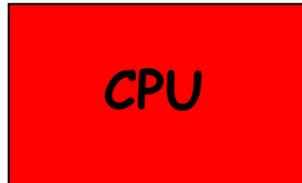
Ein Computer, der als (unsichtbarer) Bestandteil eines Geräts gekauft wird.

ein eingebettetes System:

- erfüllt eine spezielle Aufgabe.
- hat spezielle, problemangepasste Hardware und Software.
- hat spezielle, problemangepasste Benutzerschnittstelle.

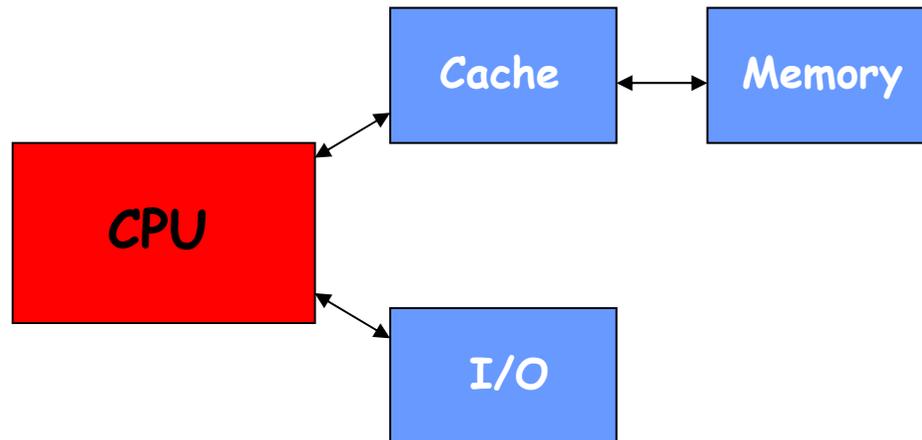
# Systemansicht einer Computerarchitektur:

Leistungs-Eigenschaften gemessen in: **Performance**



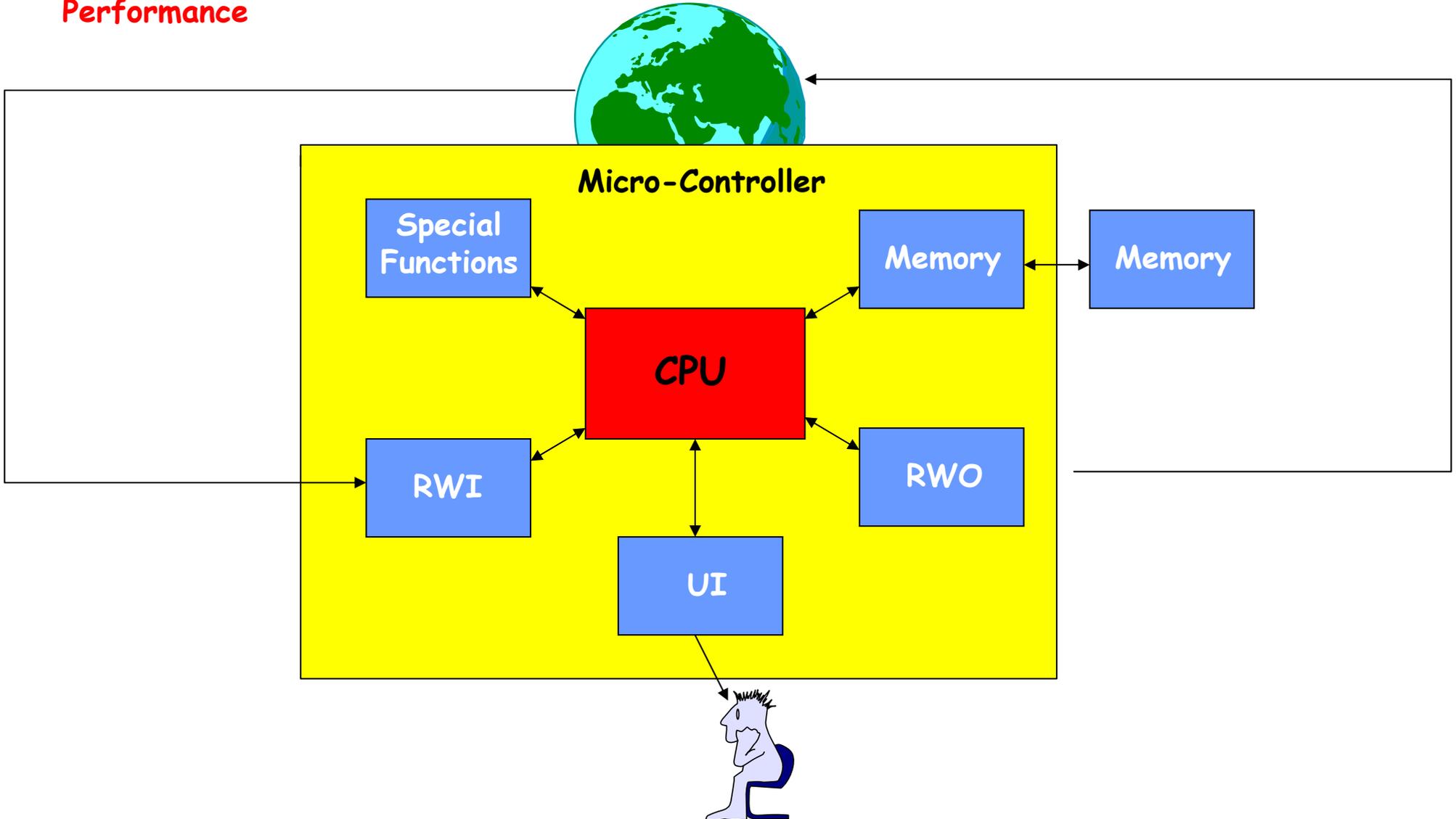
# Systemsicht eines weitdenkenden Computerarchitekten:

Leistungs-Eigenschaften gemessen in: **Performance und Kosten**



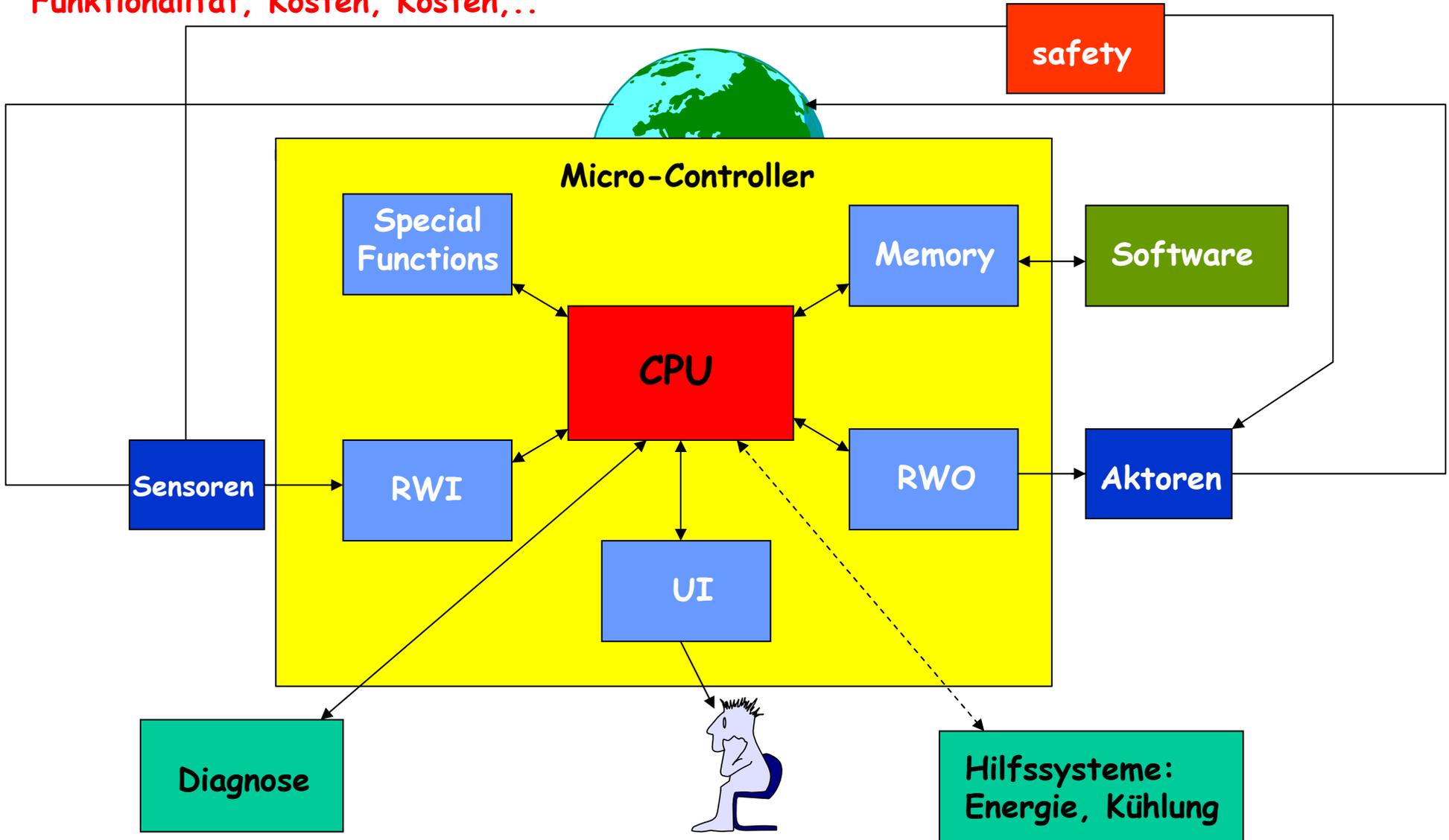
# Systemansicht eines Architekten für eingebettete Systeme:

Leistungs-Eigenschaften gemessen in: **Kosten**, **I/O-Möglichkeiten**, **Speichergröße**, **Performance**

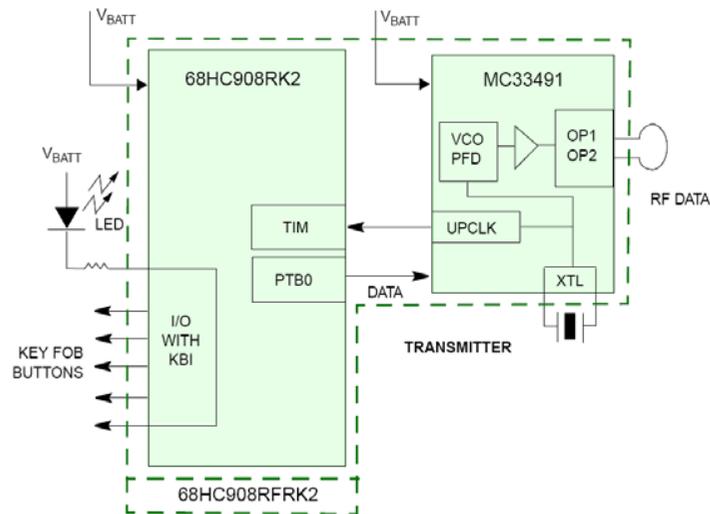


# Systemansicht eines Architekten für Kontrollsysteme:

Leistungs-Eigenschaften gemessen in: **Kosten, Time-to-Market, Kosten, Funktionalität, Kosten, Kosten,...**



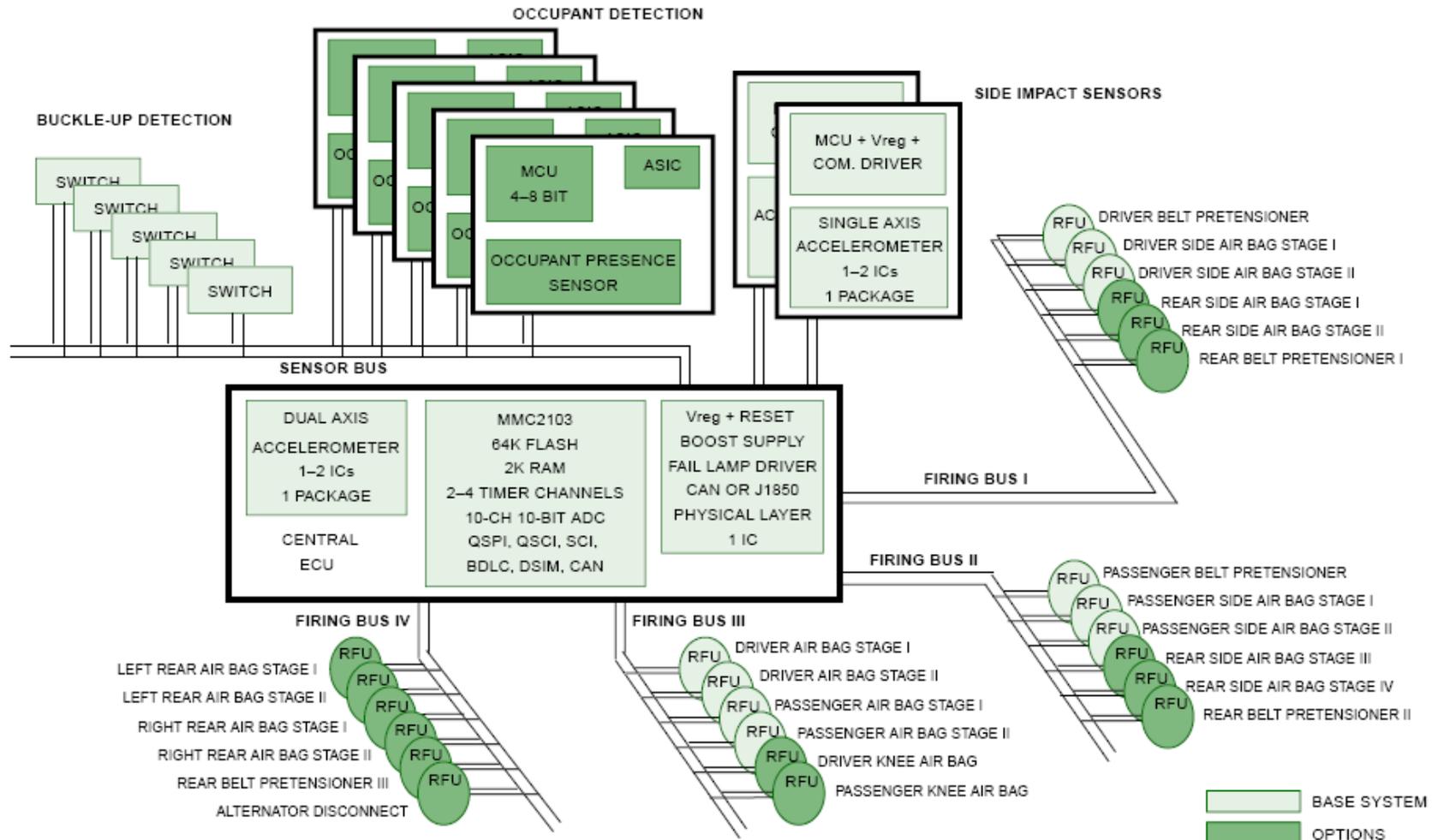
## Beispiele eingebetteter Systeme: drahtloser Türöffner



Performance ~ 100KIPS,  
Speicher <1K,  
"handgestrickte" Software,  
5 Jahre Batterielevensdauer einer Knopfzelle

Mehr Beispiele für Steuerungsaufgaben in:  
 Aufzügen, Turbinen, Flugzeugen, Autos, industrielle Automatisierung:  
 Sicherheitskritische Systeme, Echtzeitverarbeitung, verteilte Prozessoren, hohe Komplexität.

### Beispiel: verteilte Airbag Auslösung

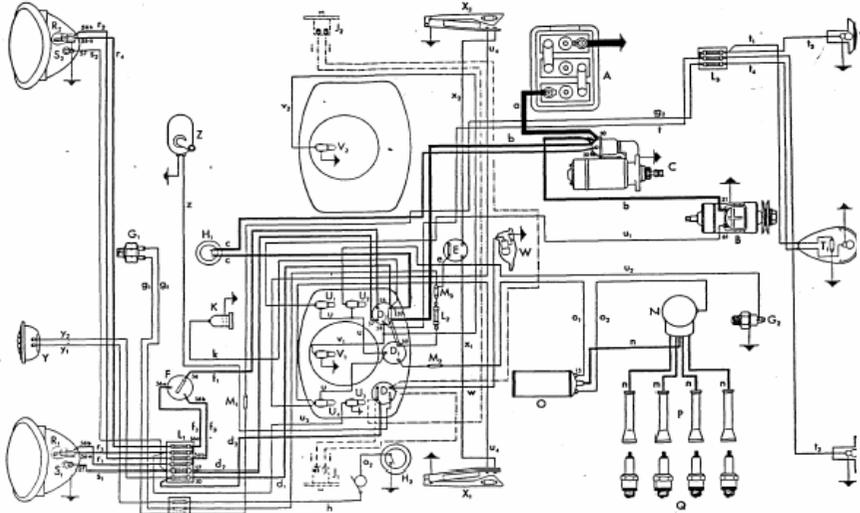


# vom notwendigen Randbereich:

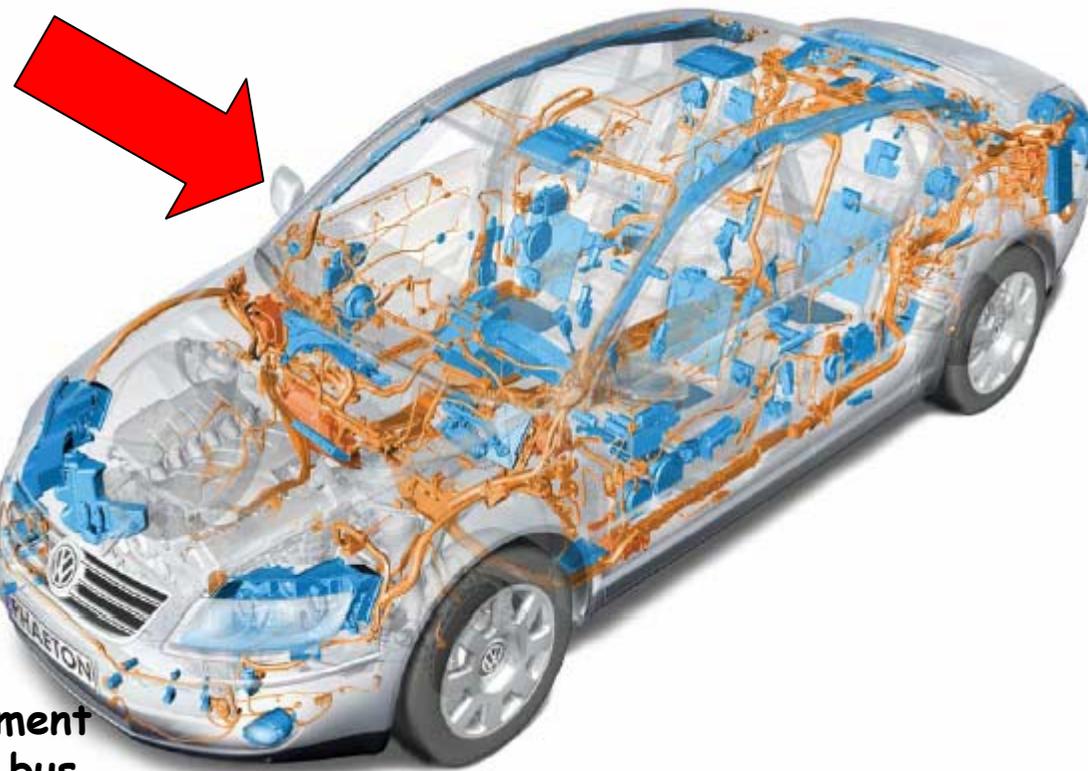


zur zentralen Problemstellung.

Elektrischer Schaltplan (Volkswagen)



KABELSCHÜSSEL														
r	schwarz-weiß-grün	1,0 mm <sup>2</sup>	h	blau	0,75 mm <sup>2</sup>	l	grün	1,0 mm <sup>2</sup>	u	blau	0,5 mm <sup>2</sup>	v	grün-grün	0,5
l <sub>1</sub>	weiß-schwarz	2,5 mm <sup>2</sup>	i	grün-grün	0,75 mm <sup>2</sup>	l <sub>2</sub>	gelb	1,5 mm <sup>2</sup>	h	grün-rot	0,5 mm <sup>2</sup>	v <sub>1</sub>	schwarz-weiß	1,0
f <sub>1</sub>	weiß	3,5 mm <sup>2</sup>	k	rot	0,5 mm <sup>2</sup>	f <sub>2</sub>	weiß-schwarz	1,5 mm <sup>2</sup>	h <sub>1</sub>	grün-schwarz	0,5 mm <sup>2</sup>	v <sub>2</sub>	schwarz-grün	1,0
f <sub>2</sub>	gelb	2,5 mm <sup>2</sup>	n	schwarz	0,83 mm <sup>2</sup>	f <sub>3</sub>	weiß	1,5 mm <sup>2</sup>	h <sub>2</sub>	grün	0,5 mm <sup>2</sup>	v <sub>3</sub>	blau	1,0
h	schwarz-rot	0,75 mm <sup>2</sup>	w	schwarz	0,75 mm <sup>2</sup>	f <sub>4</sub>	grün-schwarz	0,5 mm <sup>2</sup>	h <sub>3</sub>	schwarz-rot	0,75 mm <sup>2</sup>	v <sub>4</sub>	schwarz-gelb	1,0



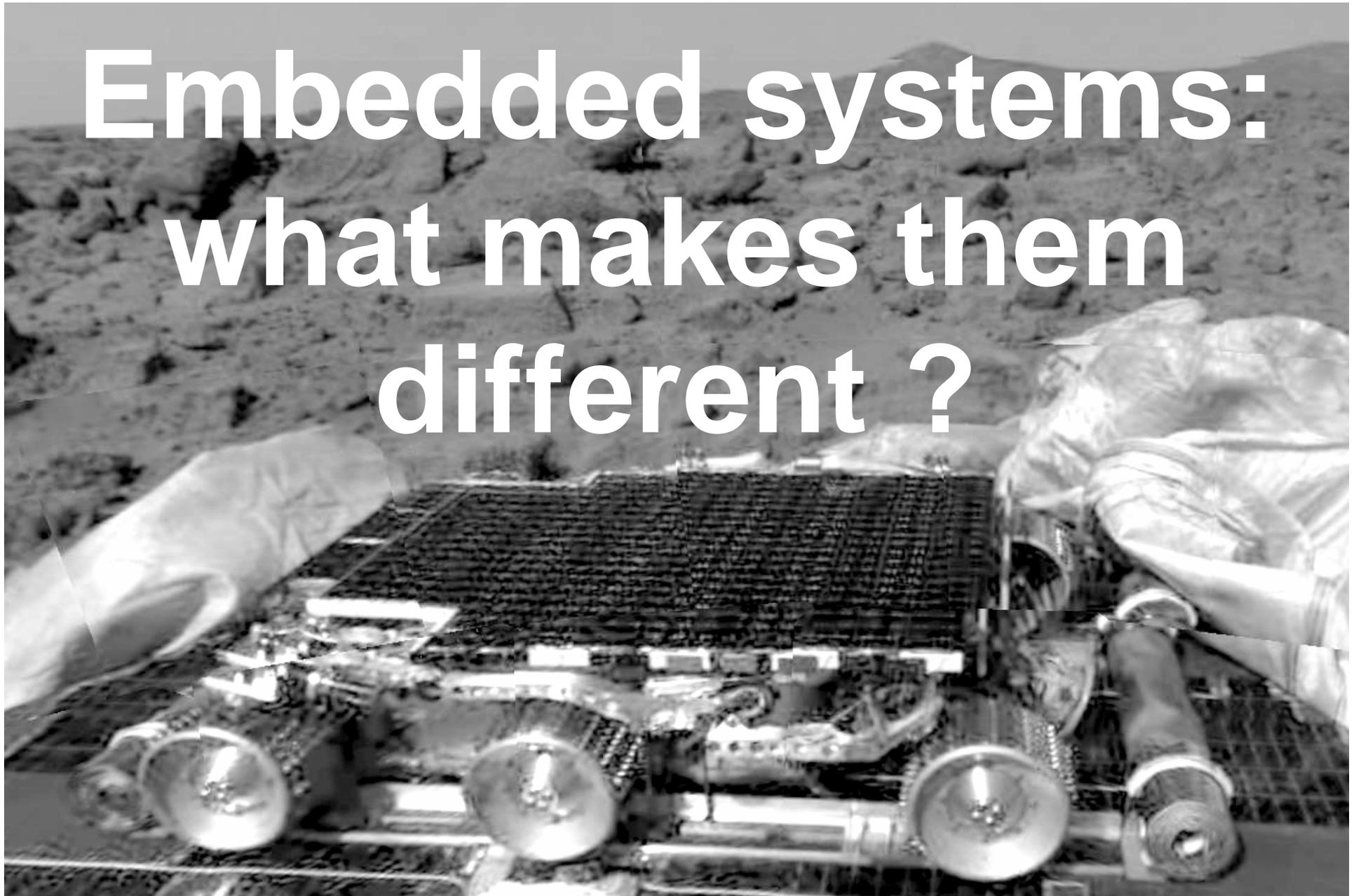
- 11.136 electrical parts
- 61 ECUs
- Optical bus for information and entertainment
- Sub networks based on proprietary serial bus
- 35 ECUs connected to 3 CAN-Busses
- 2500 signals in 250 CAN messages

Der Wert zukünftiger Automobile wird bis zu 40% durch eingeb. HW/SW Komponenten bestimmt.

# Applications

- **Control of industrial production processes**
- **Plant management**
- **Power plant control (conventional, nuclear)**
- **Railway control systems**
- **Flight control systems**
- **Environmental acquisition and monitoring**
- **Space missions**
- **Military systems**
- **Telecommunication systems**
- **Robotics**
- **Team Robotics and autonomous vehicles**
- **Virtual and augmented reality**

**Embedded systems:  
what makes them  
different ?**



# Typen eingebetteter Systeme

## Allgemeine eingebettete Systeme:

Funktionen (und Probleme) ähnlich wie general purpose Rechner aber in einer speziellen "Verpackung".

Beispiele: Videospiele, Set-top-Boxen, PDAs, Navigationssysteme, Geldautomaten,...

## Kontrollsysteme:

Kontrolle physischer Prozesse, Rückkopplungsschleifen unter Echtzeitbedingungen

Beispiele: Motoren, chemische Prozesse, Kraftwerke, Fly-by-Wire-Systeme, ABS, EPS,....

## Signalverarbeitungssysteme:

Verarbeitung von Datenströmen, Störungsbeseitigung, Filterung, Signalerkennung

Beispiele: Radar, Sonar, Funktechnik, Video(de)kompression, Verschlüsselung....

## Kommunikation und Netze:

Übertragung, Verbindung, Routing

Beispiele: Telefonverbindungsanlagen, Basisstationen, Router, Internetinfrastruktur....

## Welche Funktionen werden in eingebetteten Systemen vorzugsweise benötigt?

### Kontrollfunktionen:

PID Kontrolle, Fuzzy Logic, ...

### Anwendungsbezogene Schnittstellenrealisierung:

Knöpfe, Anzeigen, LEDs, Beeper,

### Fehlerbehandlung:

Erkennung, Rekonfiguration und Recovery

### Signalverarbeitung:

Digitale Filter,

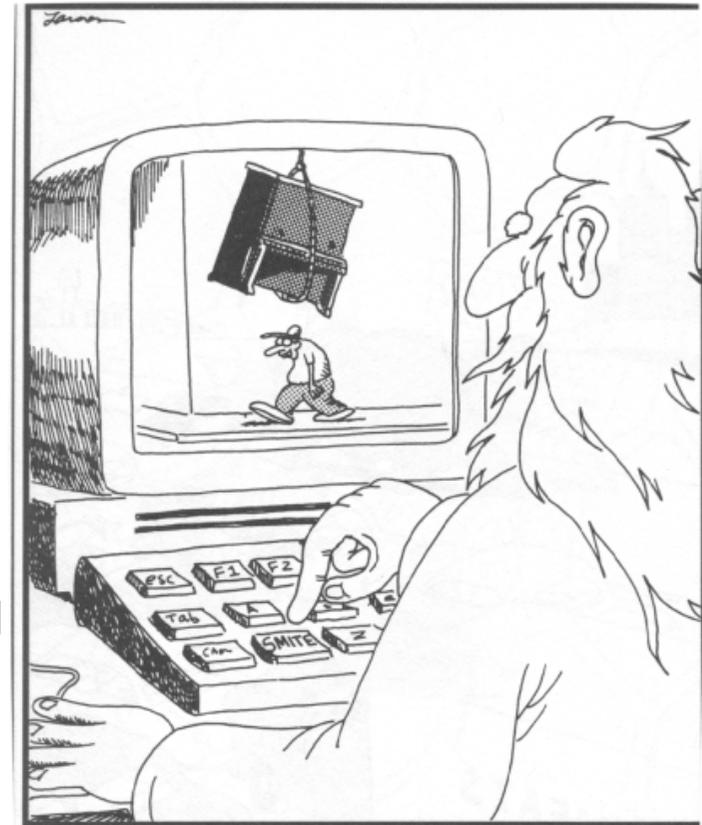
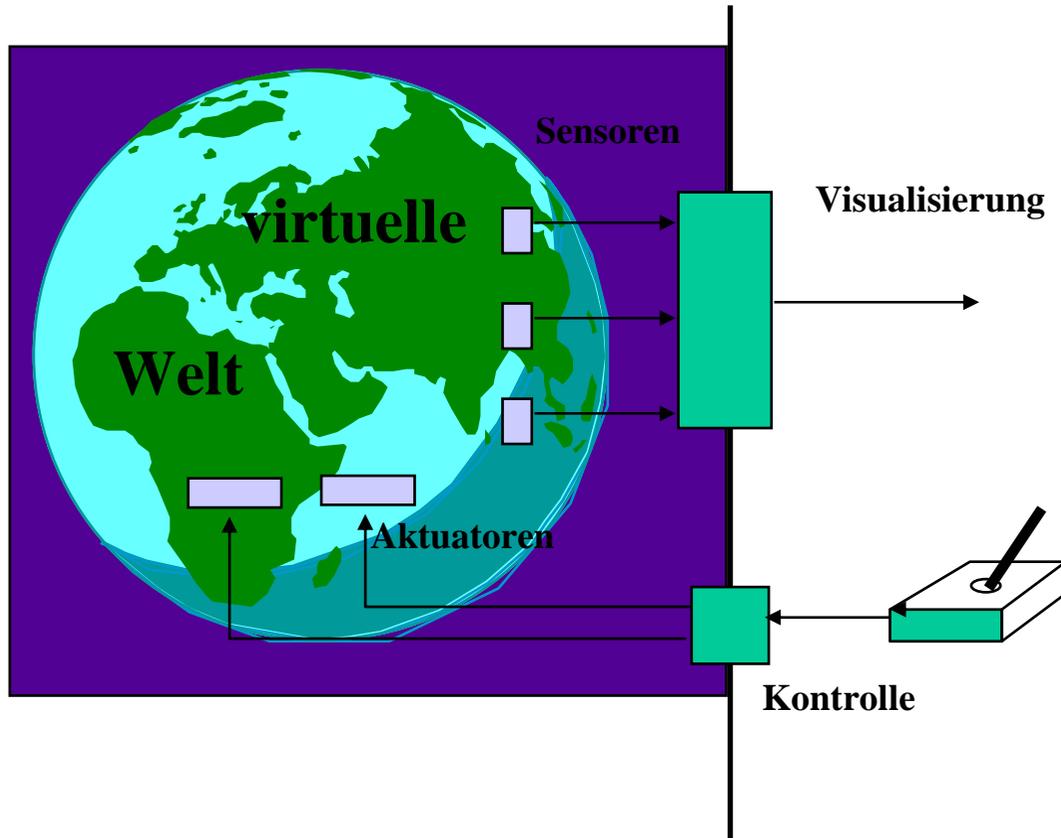
## Eigenschaften:

**Reaktiv:** Berechnungen und Aktionen werden abhängig von externen Ereignissen durchgeführt.

**Echtzeitfähig:** Korrektheit von Berechnungen und Aktionen ist eine Funktion der Zeit.

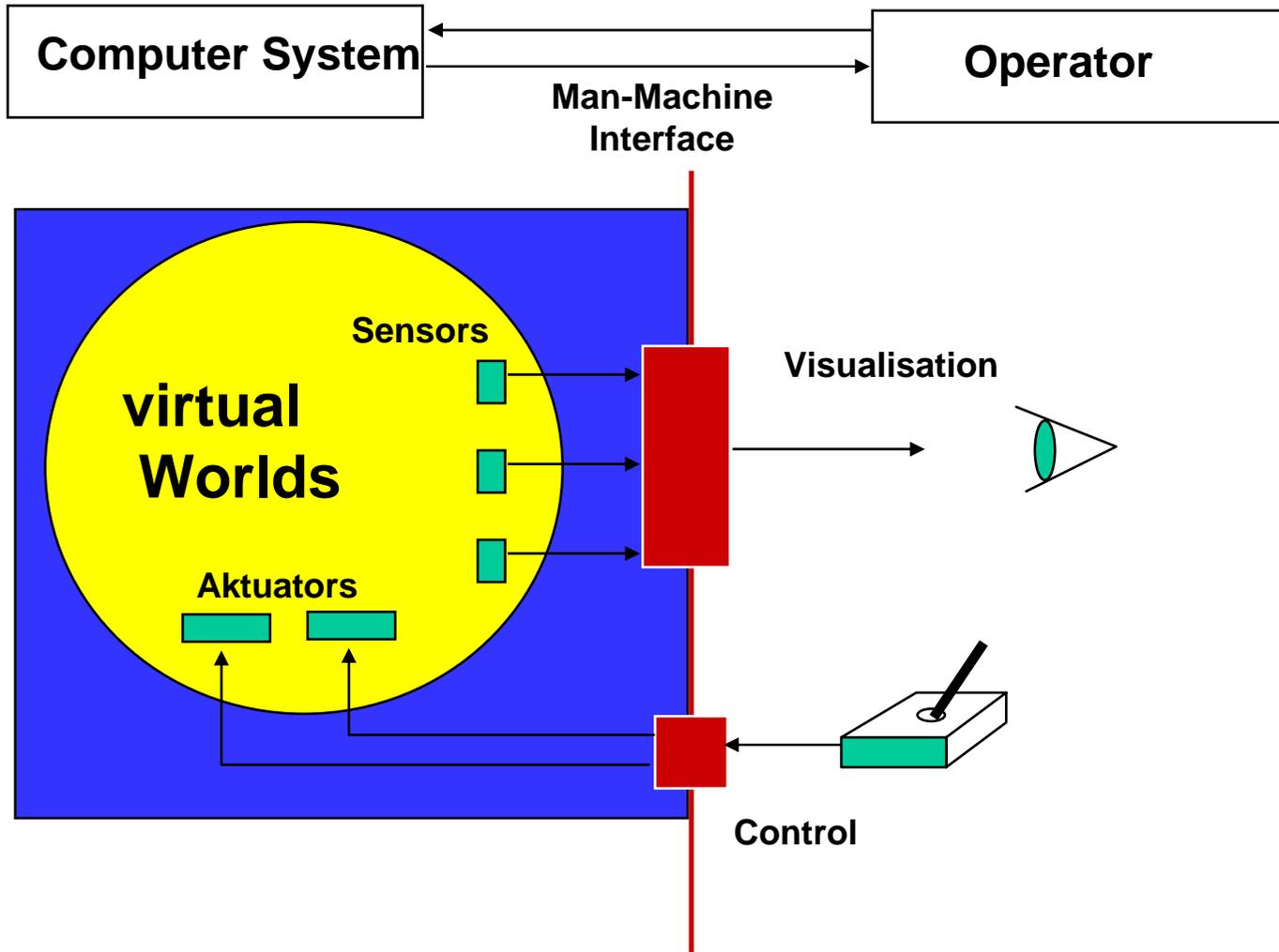
# Simulated Worlds

## The Masters of the (Time) Universe

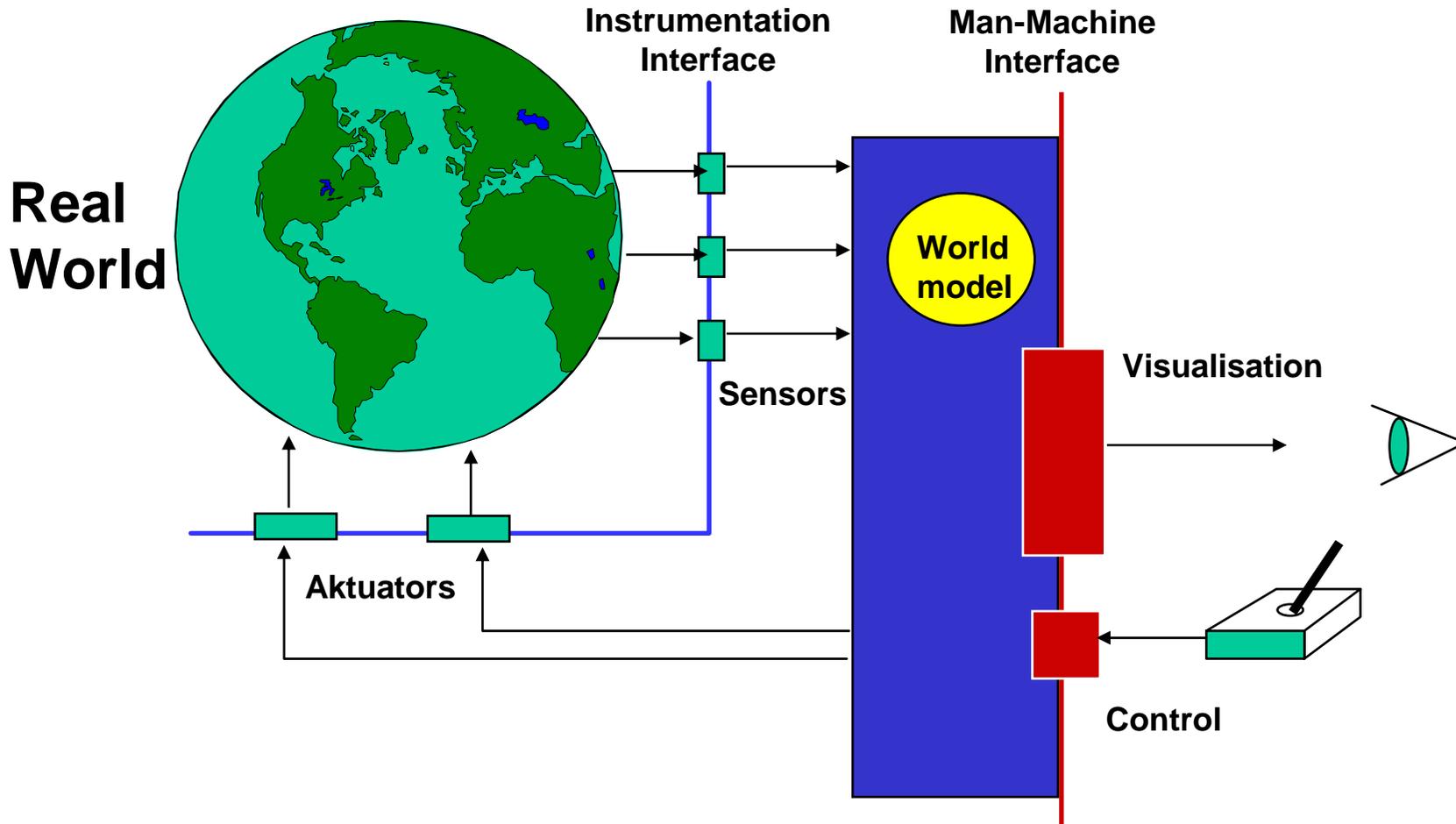
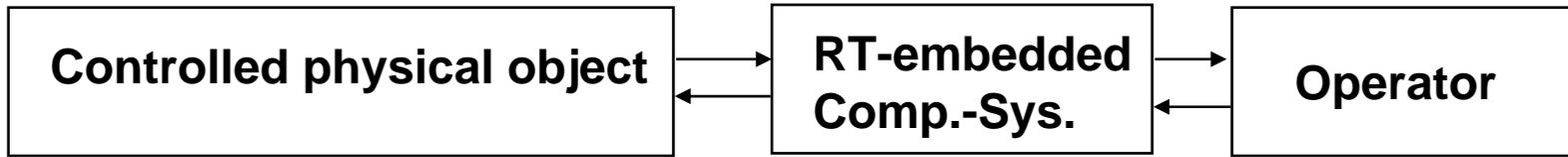


God at His computer

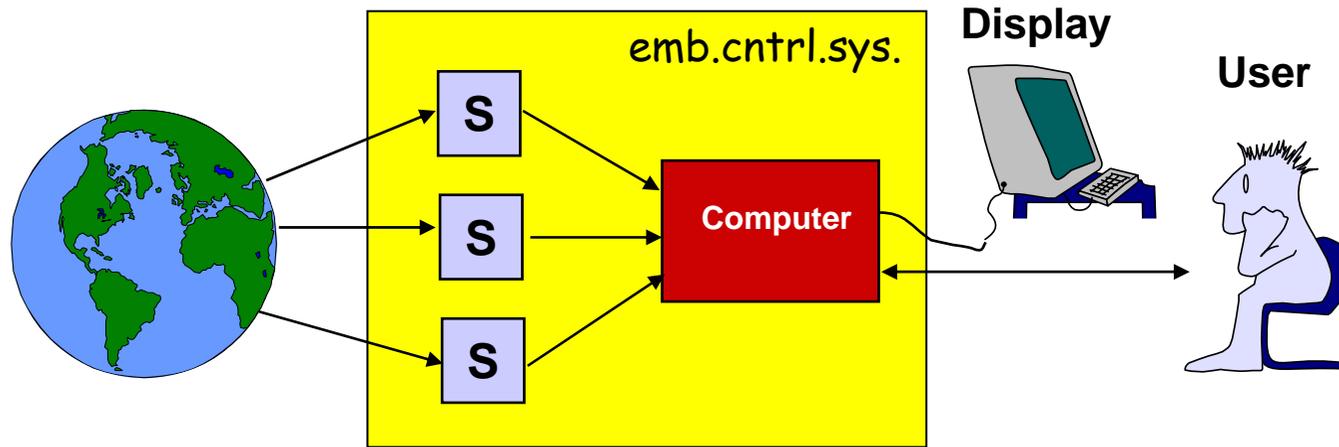
# A Computer System



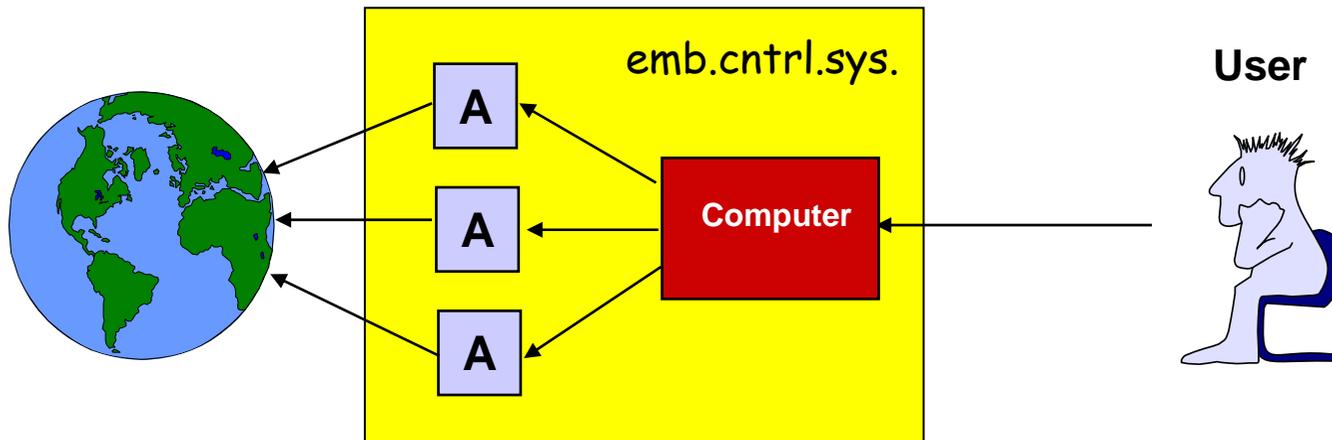
# A Computer System to Control Physical Processes



## a.) Monitoring System

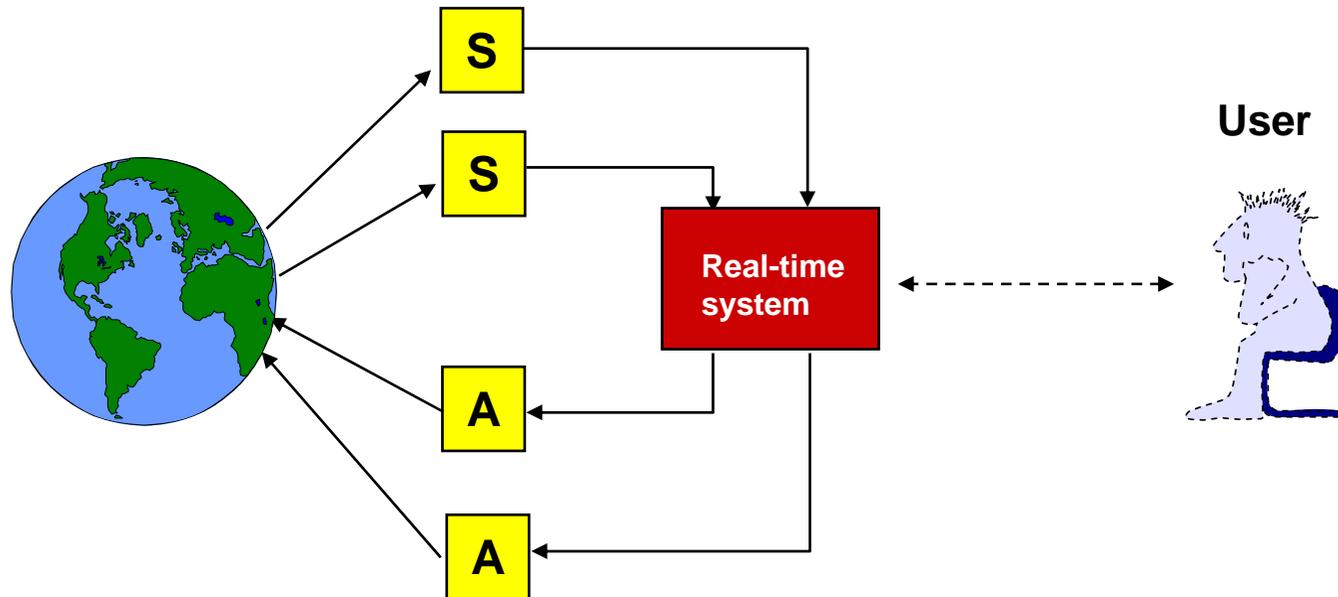


## b.) "Open Loop" System



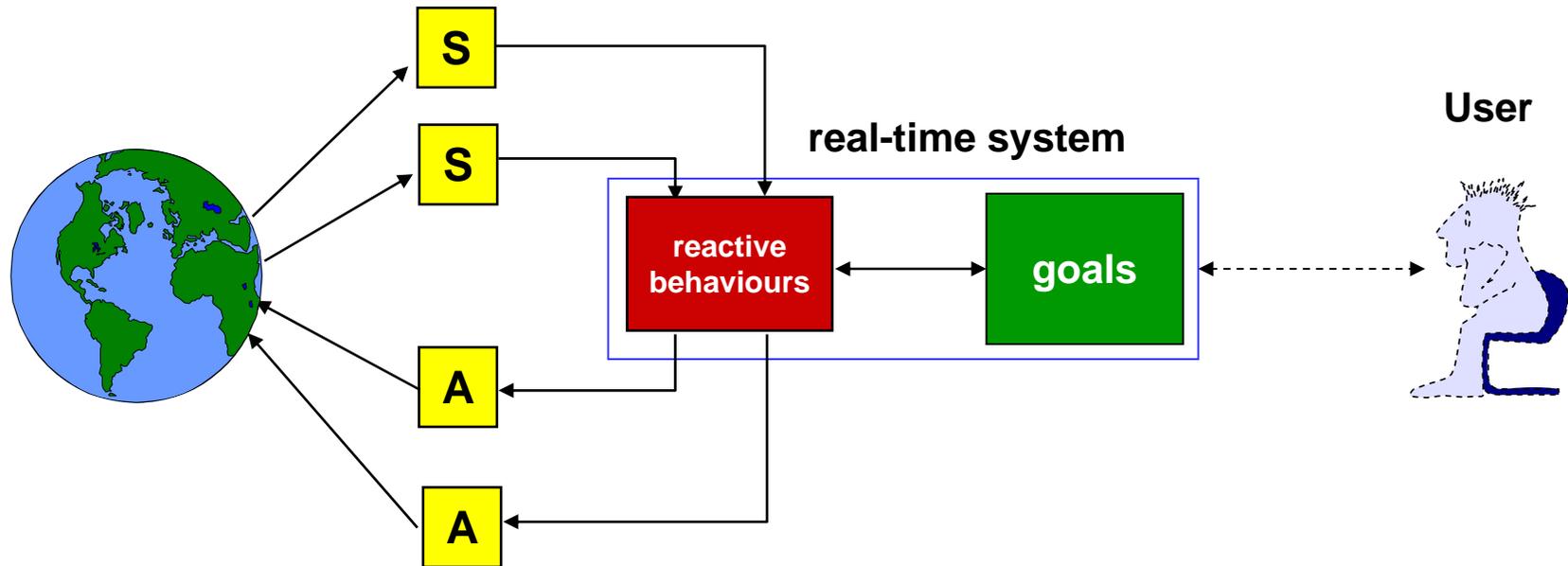
**a + b = Human-in-the-loop**

## c.) “Feedback Control” System



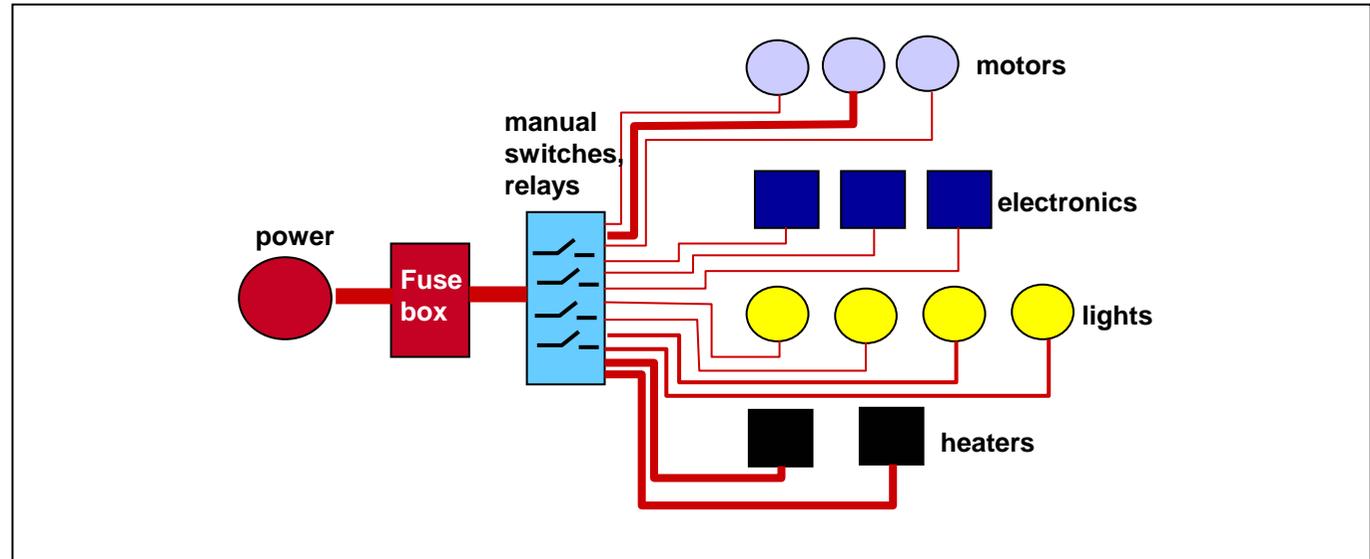
- Actuation changes the environment
- ➔** The environment is a communication channel!

## d.) autonomous control

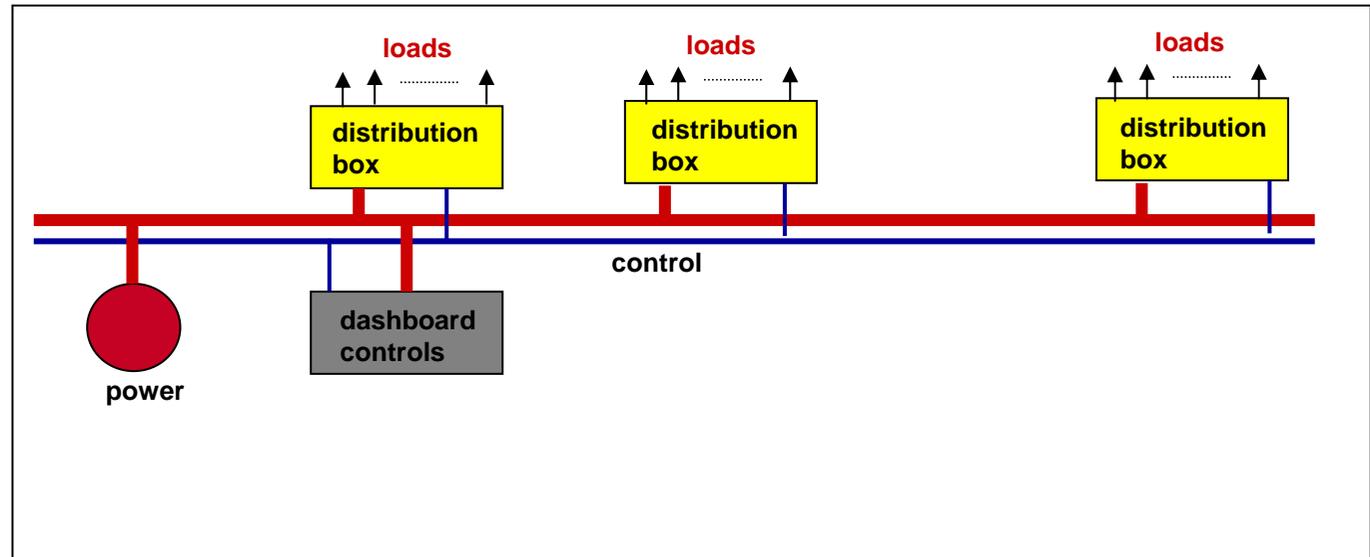


# Scenario: Car Control

yesterday

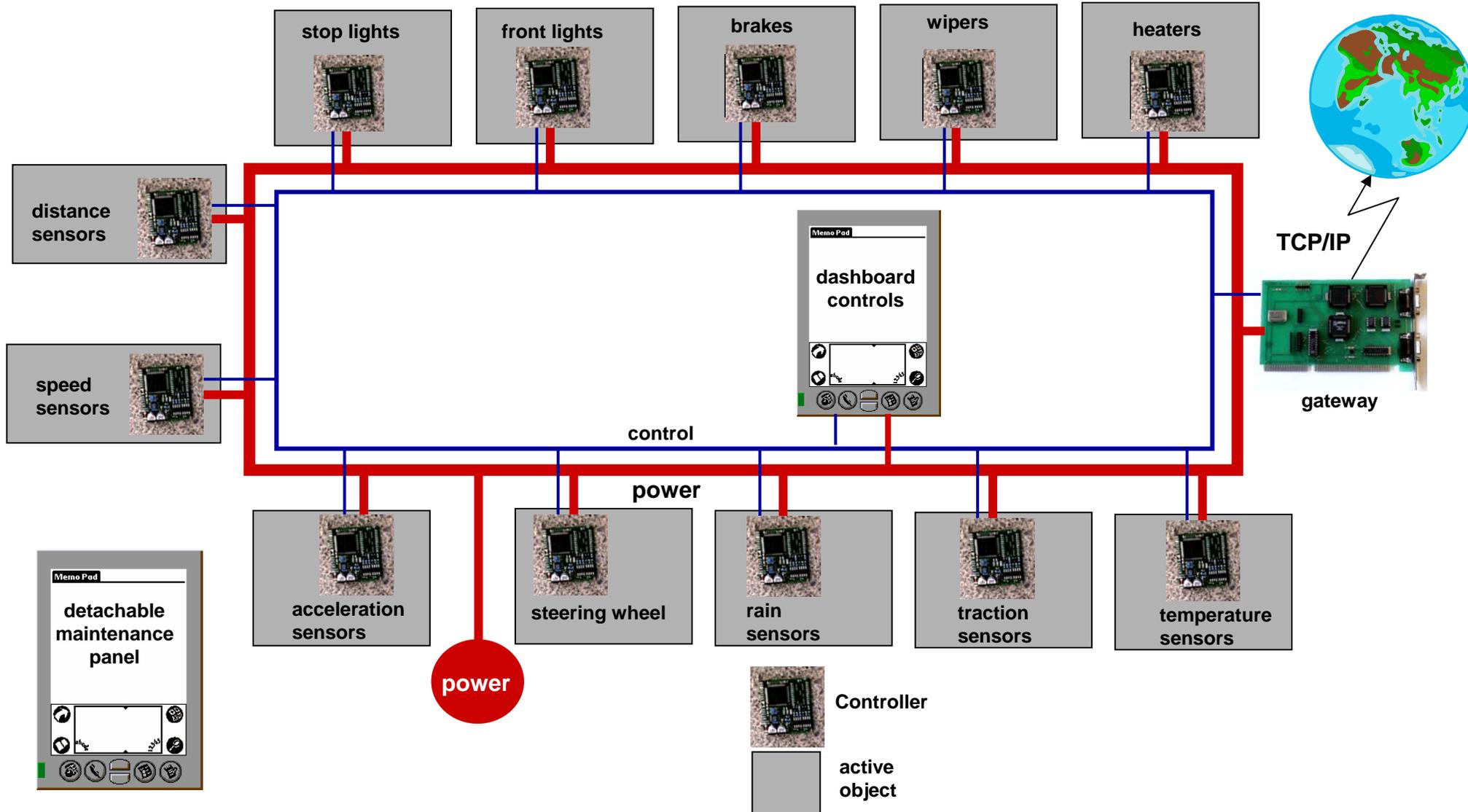


today

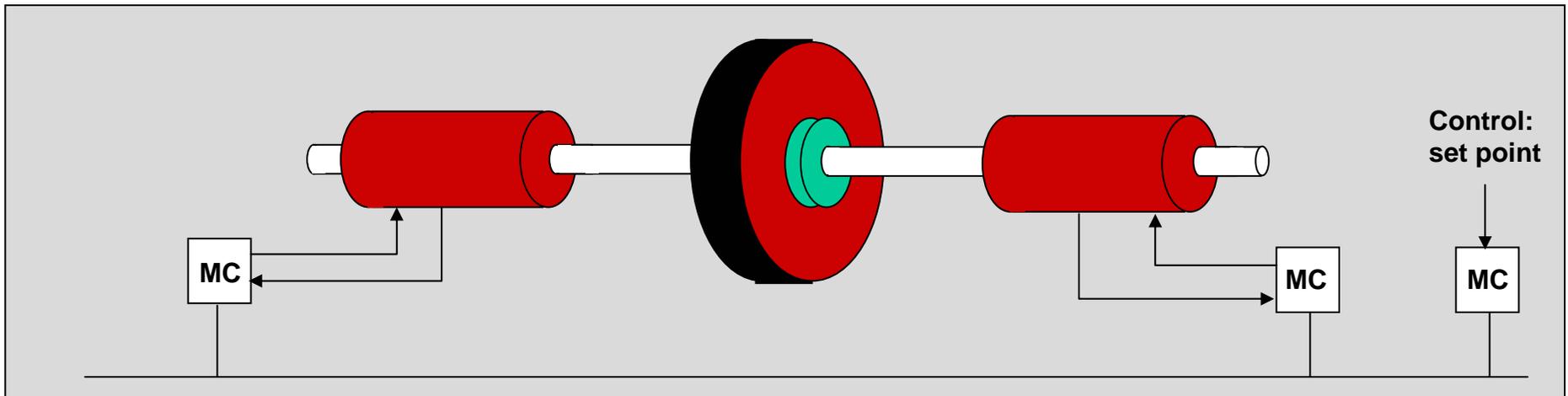
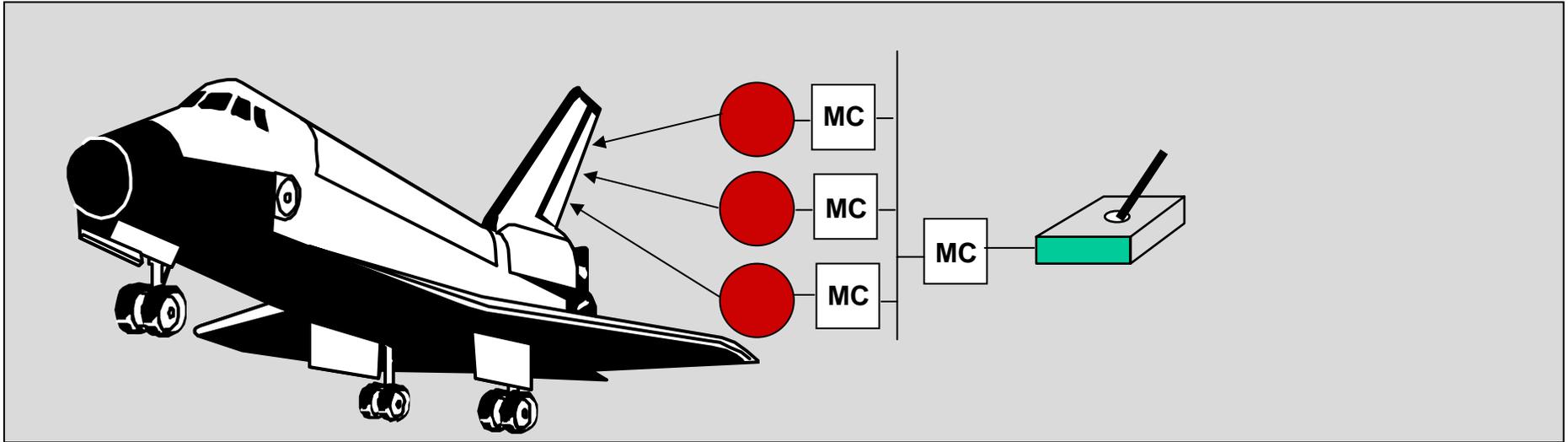


Die Ursache für jedes zweite  
liegende gebliebene Auto ist ein  
Defekt der Elektronik !  
Die Welt, Sept. 2002

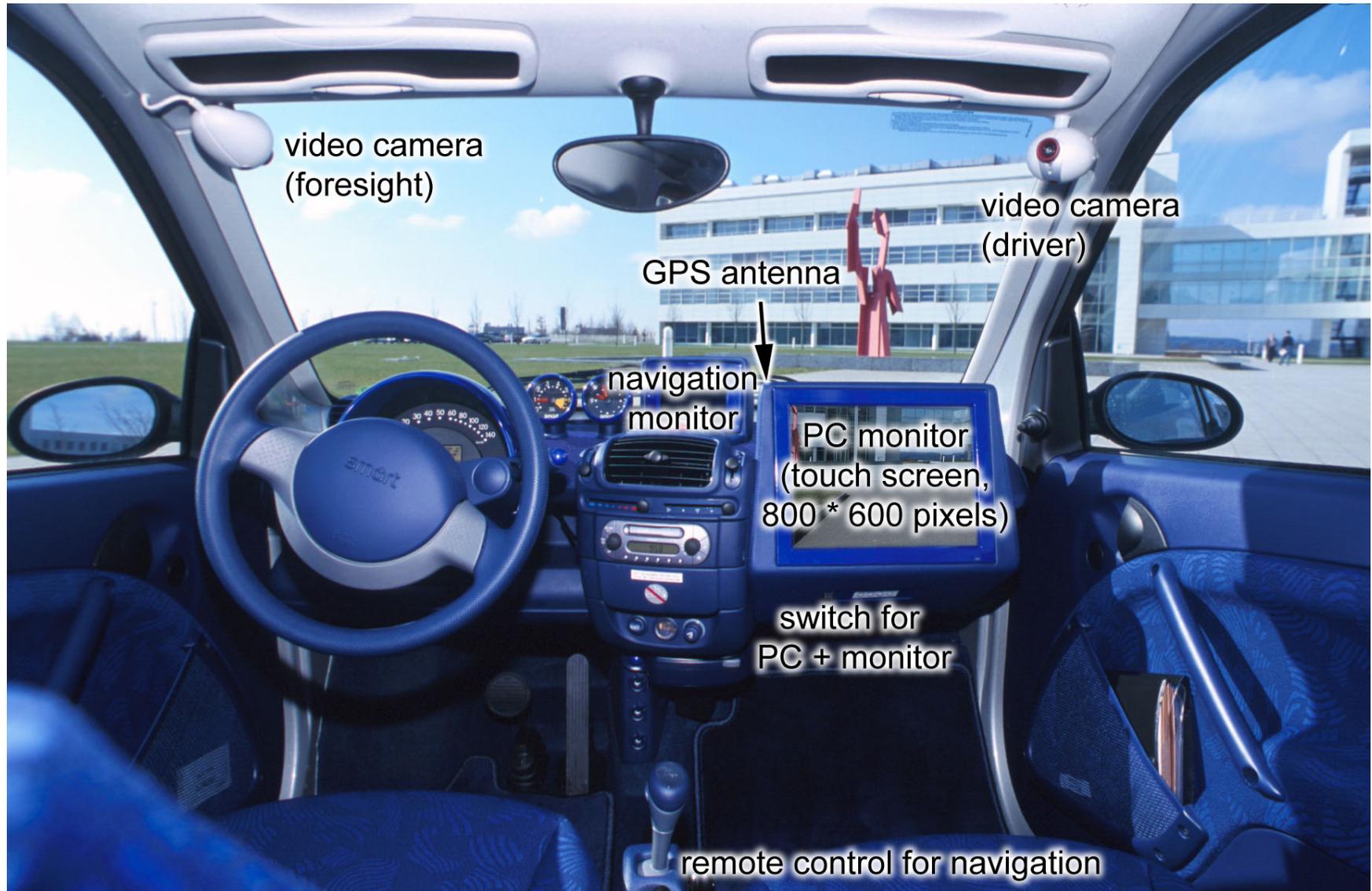
# future: distributed co-operative control



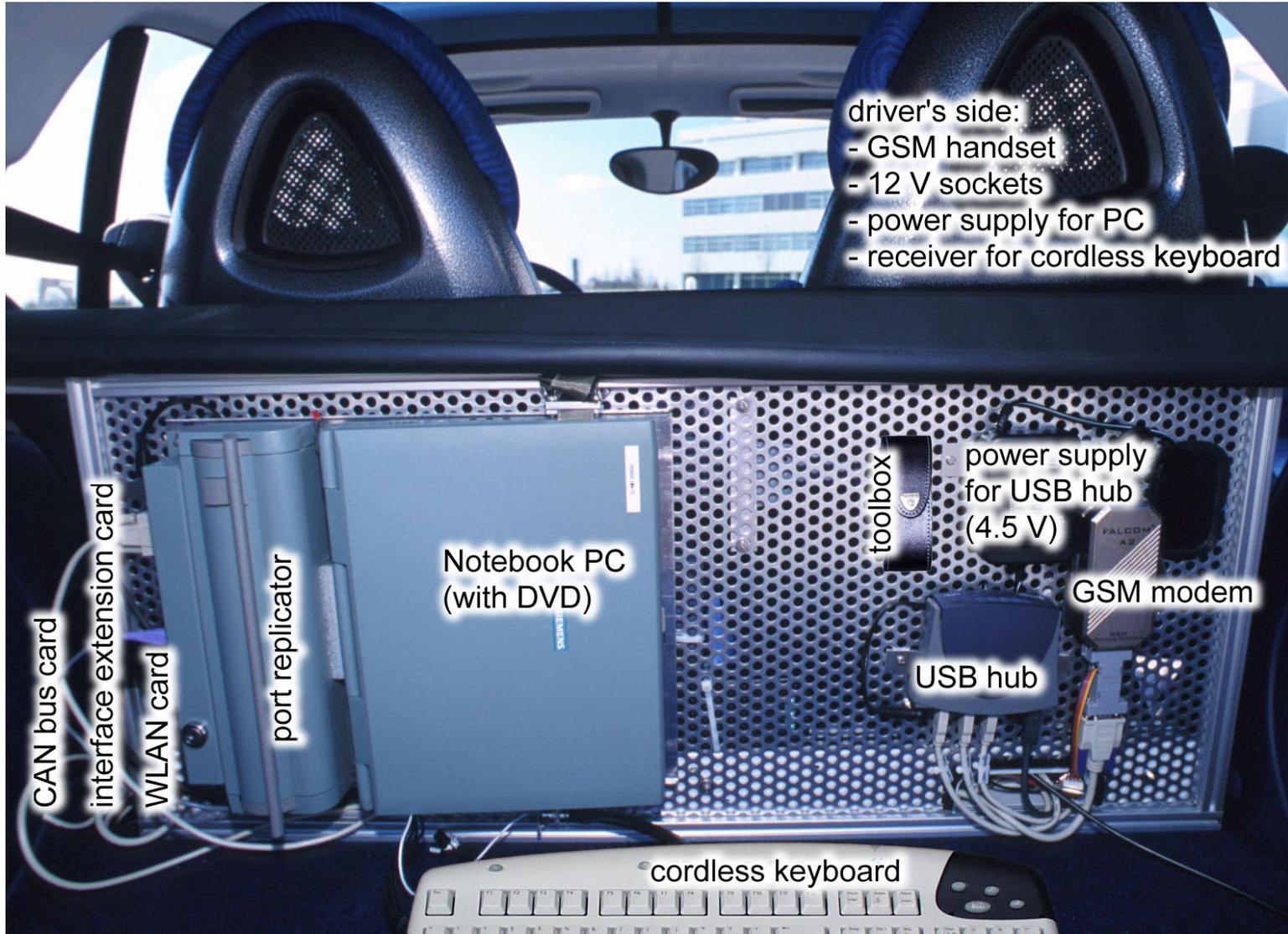
# Szenarien die Konsens benötigen über das WAS und WANN !



# A multimode Interface



# Smart Equipment

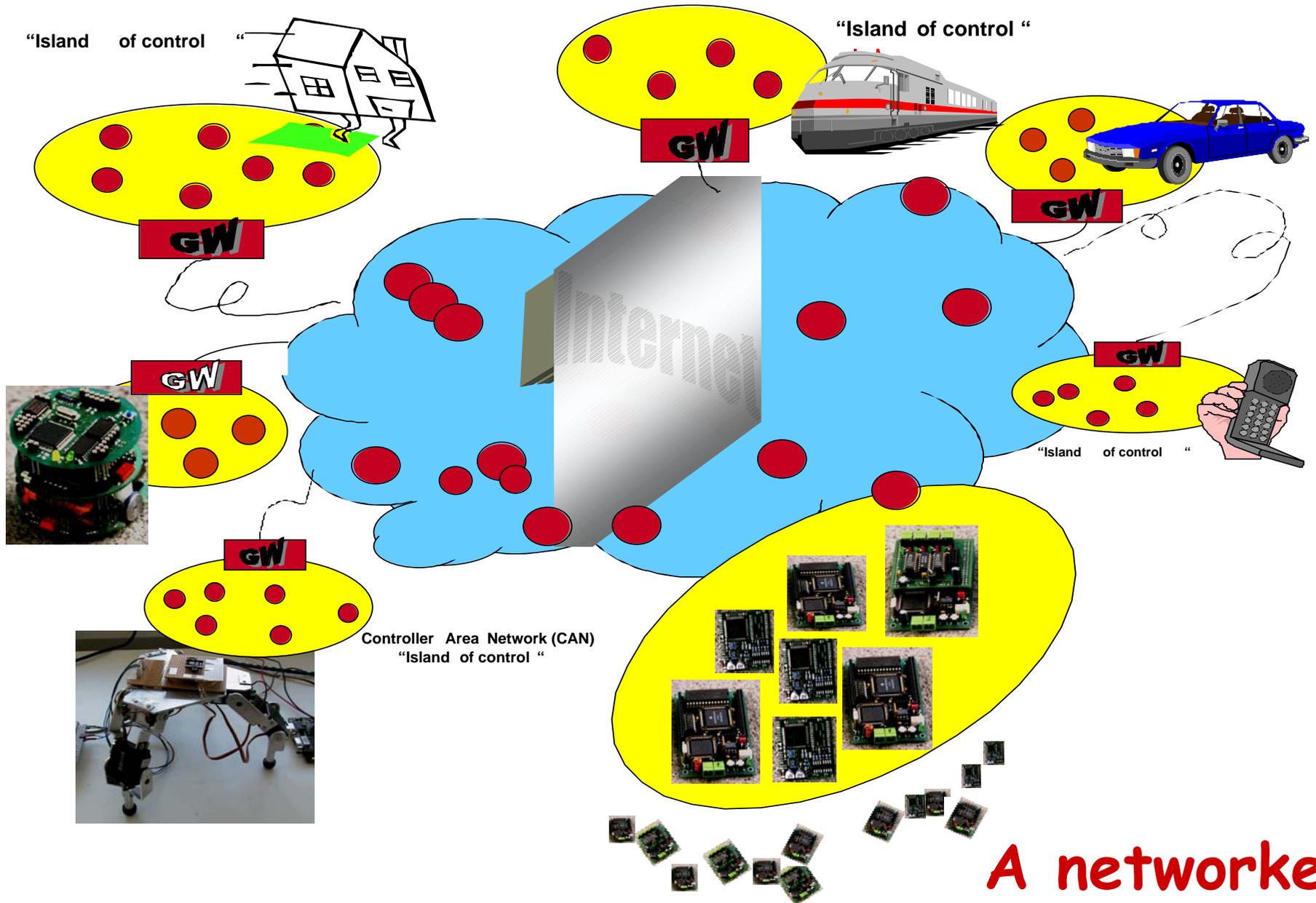


# Services in a Car?

- Restricted interaction with the driver.
- Used by a broad community without specific IT knowledge.
- Services must be designed to work without direct driver interaction!
- User preferences and on-board equipment have to be considered when selecting and using services !
- Context information is necessary to provide an adequate service !

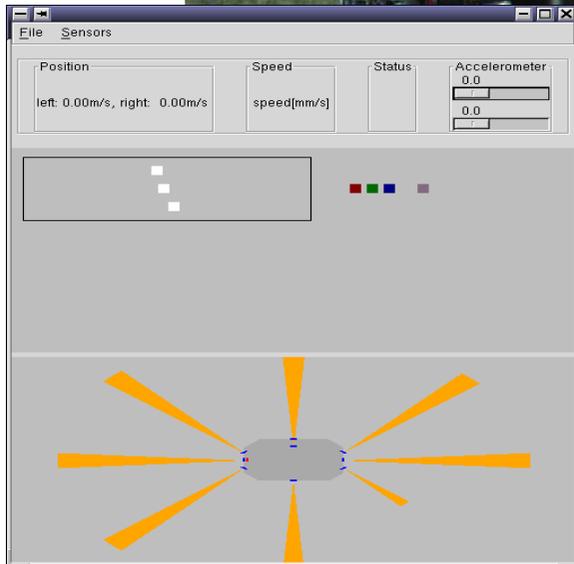
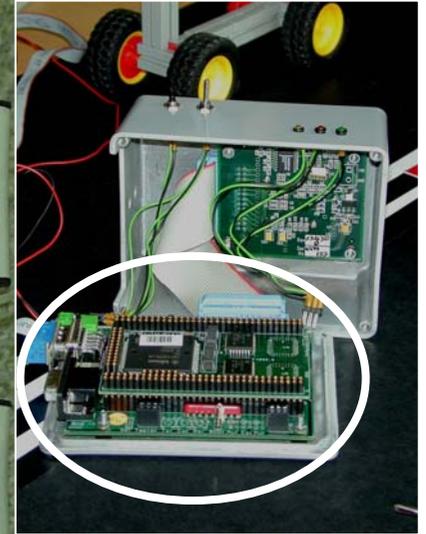
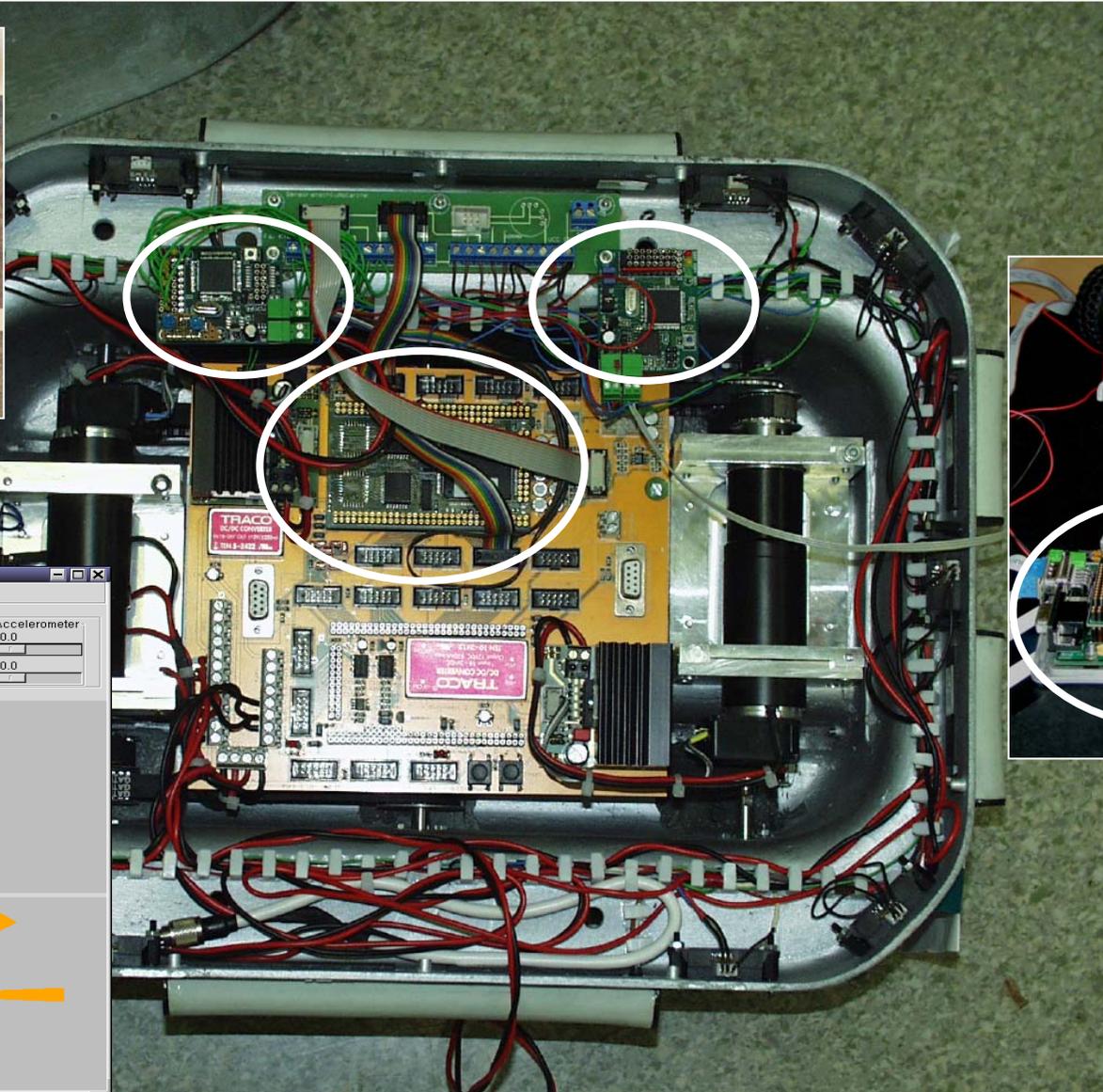


EINE UNFALLURSACHE, DIE AN HÄUFIGKEIT ZUNIMMT:  
UNAUFMERKSAMKEIT BEIM FAXEN

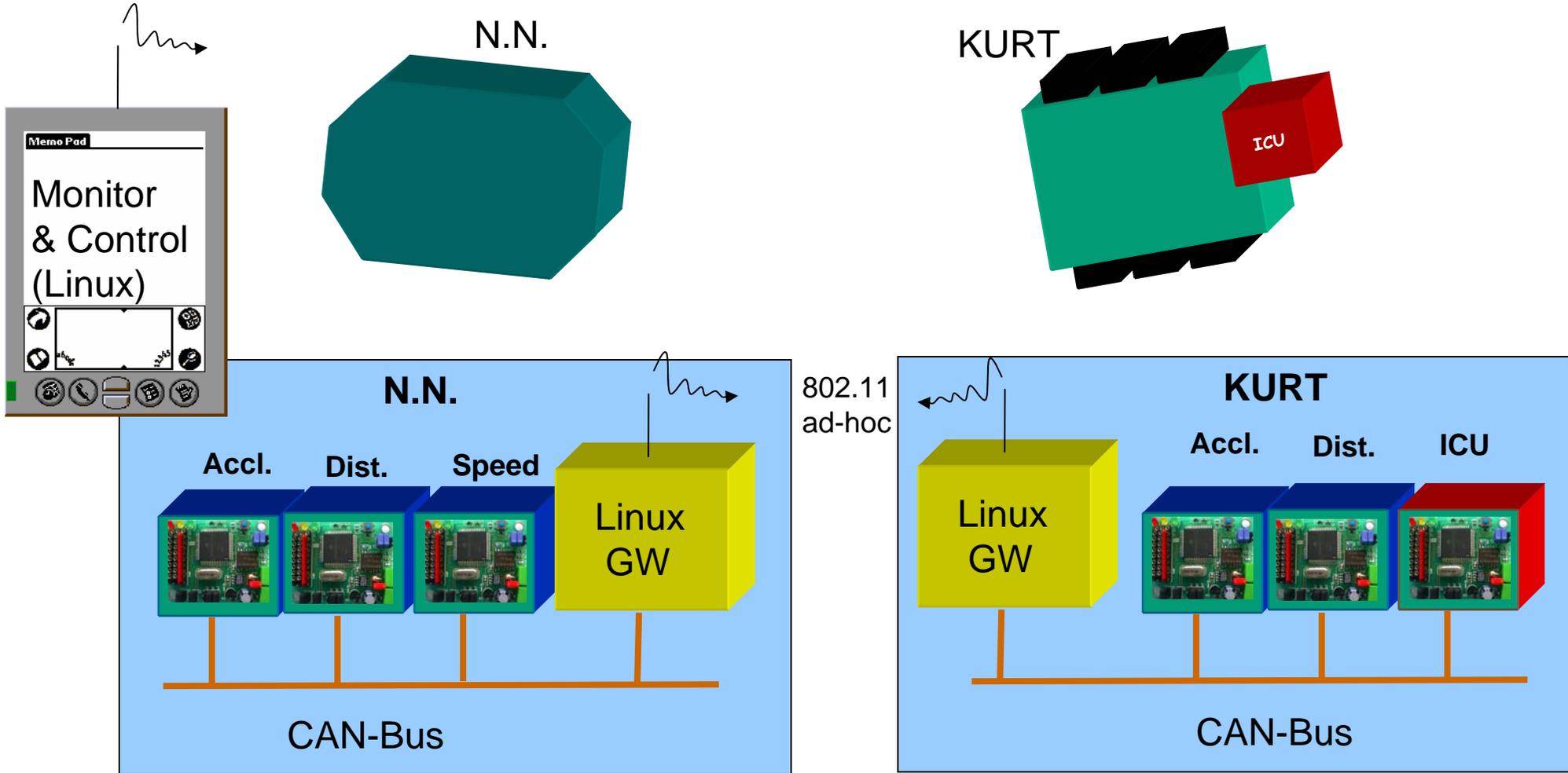


**A networked  
physical world**

# Distributed Robot Control

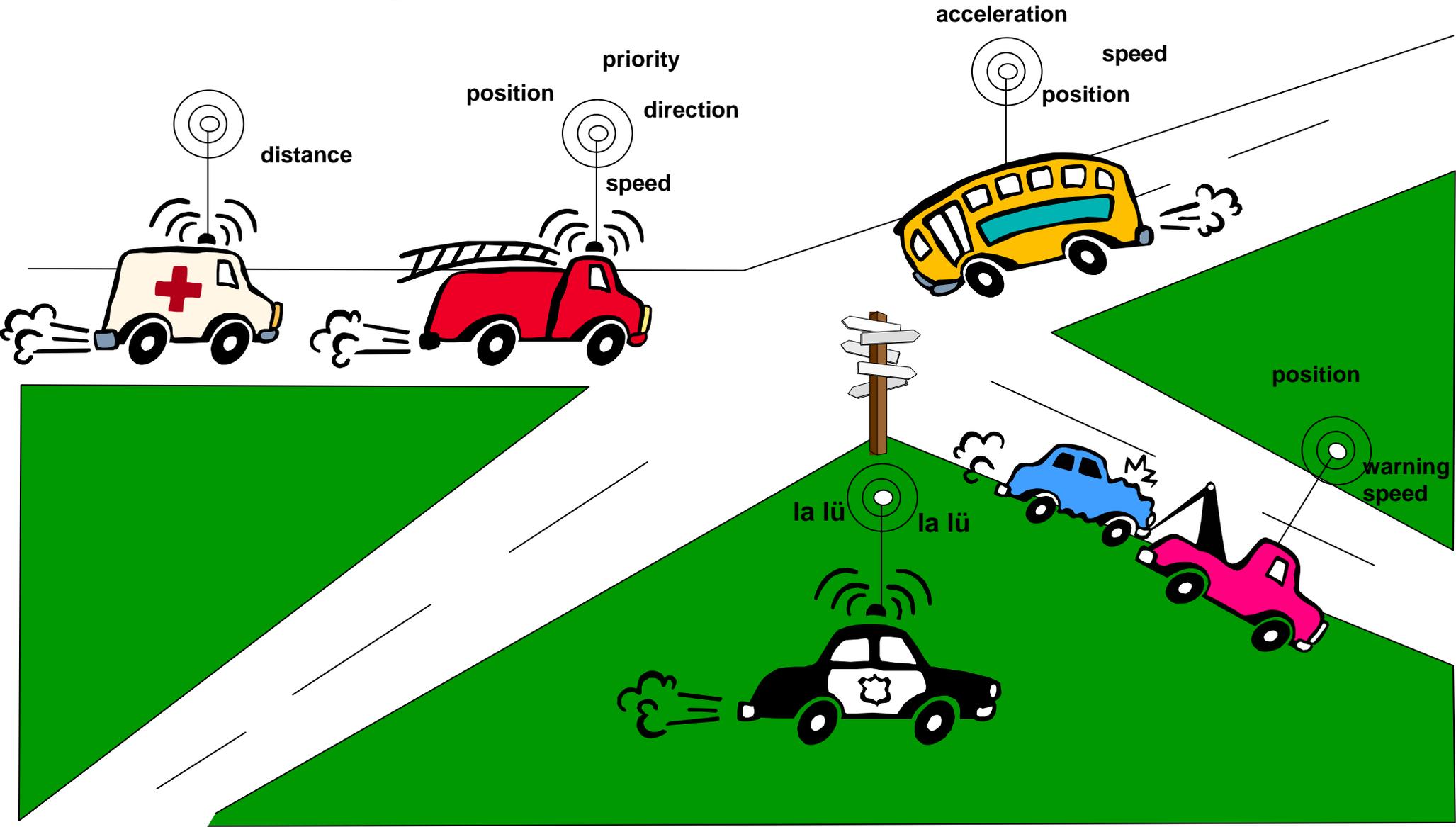


# Demo Scenario





# Co-operating vehicles



# Murphy's laws on real-time systems

- Murphy's general law:** If something can go wrong, it will go wrong.
- Murphy's constant:** Damage to an object is proportional to its value.
- Johnson's first law:** sooner or later, the worst possible combination of circumstances will happen.
- Corollary:** A system must always be designed to resist the worst possible combination of circumstances

**There is no technical solution for a 100 % safe  
System !**

**Predictability of operation is always based on  
explicit and implicit assumptions about:**

- the environment**
- the system components**

**There are good scientific and engineering principles  
to build such systems and predict their operation.**

# Literatur:

## Embedded Systems und Micro-Controller:

Wayne Wolf:  
Computers as Components  
Principles of Embedded Computing System Design  
Academic Press, 2001

Uwe Brinkschulte, Theo Ungerer:  
Mikrocontroller und Mikroprozessoren  
Springer Lehrbuch, 2002

## Architektur von RT-Emb.-Systemen und Übersicht:

Paulo Verissimo, Luis Rodrigues:  
„Distributed Systems for System Architects“  
Kluwer Academic Publishers, 2001

Hermann Kopetz:  
„Real-Time Systems, Design Principles for Distributed  
Embedded Applications“  
Kluwer Academic Publishers, 1997

## RT-Scheduling

Giorgio C. Buttazzo:  
Hard Real-Time Computing Systems  
„Predictable Scheduling Algorithms and  
Applications“  
Kluwer Academic Publishers, Boston, Dordrecht,  
London, 2000

Dieter Zöbel, Wolfgang Albrecht  
Echtzeitsysteme - Grundlagen und Anwendungen  
Informatik-Lehrbuch-Reihe  
(Hrsg. B. Mahr, A. Schill, G. Vossen)  
International Thomson Publishing Comp., 1995

*weitere Literatur in der Vorlesung.*