

# OSEK/VDX

---

## Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug

Vehicle Distributed Executive

Thomas Batt  
t.batt@microconsult.com

## 1. Anforderungen an ein Echtzeitbetriebssystem im KfZ

## 2. RTOS Abstraktion

## 3. OSEK in der Übersicht

- Vorstand und Informationsquelle im Internet
- Dokumente und Spezifikationen

## 4. OSEK Betriebssystem-Mechanismen in der Übersicht

- Task-Management
- Interrupt-Handling
- Ereignisse
- Ressourcen-Management
- Alarme

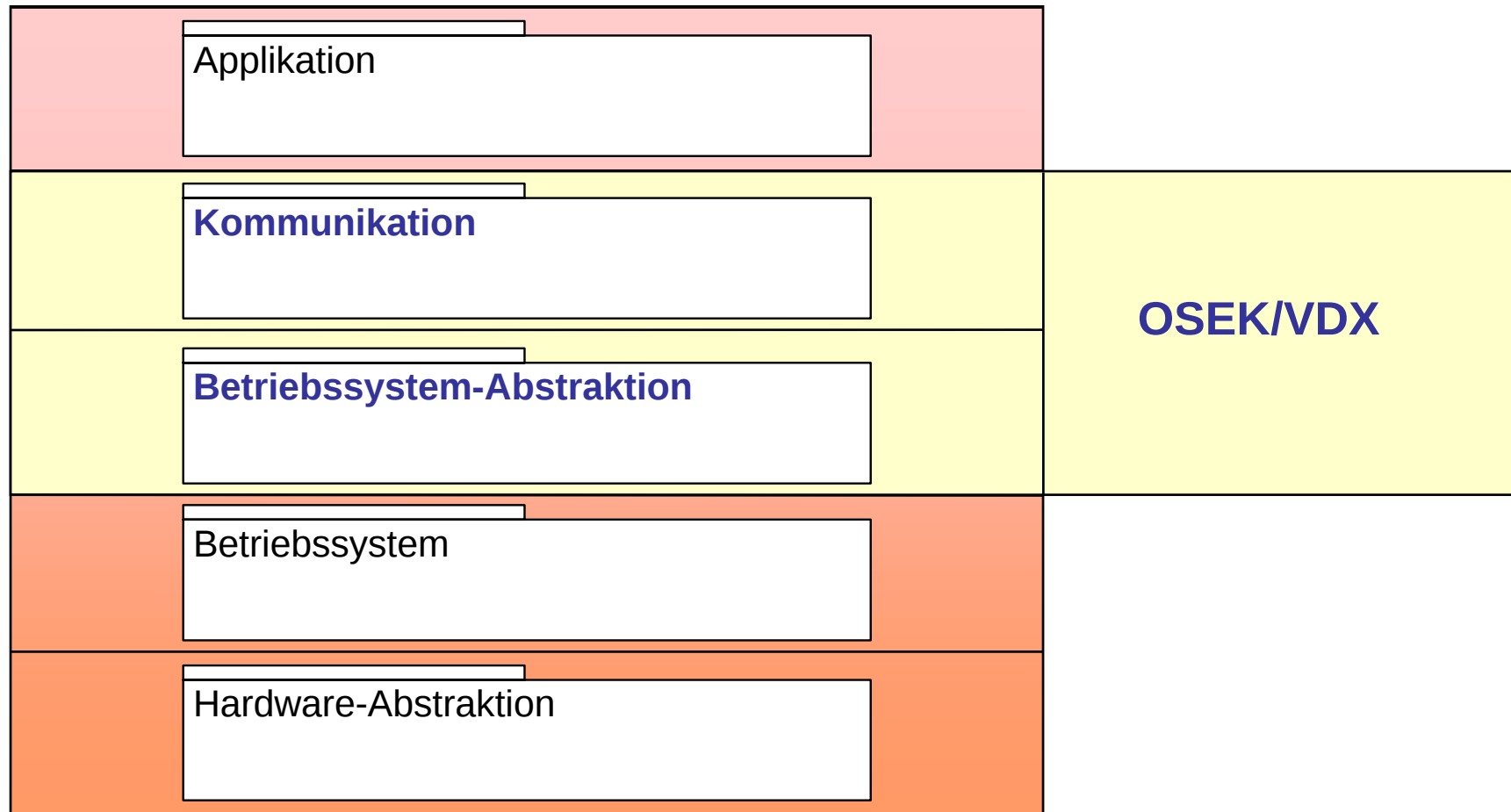
## 5. OSEK-Implementierungen

## 6. Zusammenfassung

## OSEK/VDX

- Hohe Geschwindigkeit
- Hohe Zuverlässigkeit
- Hohe Sicherheit
- Minimaler Ressourcenverbrauch
- Berücksichtigung von Kommunikation
- Unabhängigkeit zur Hardware
- Unabhängigkeit zum Betriebssystem
- Niedrige Kosten (Preis- und Lizenzpolitik der Betriebssystem-Hersteller)



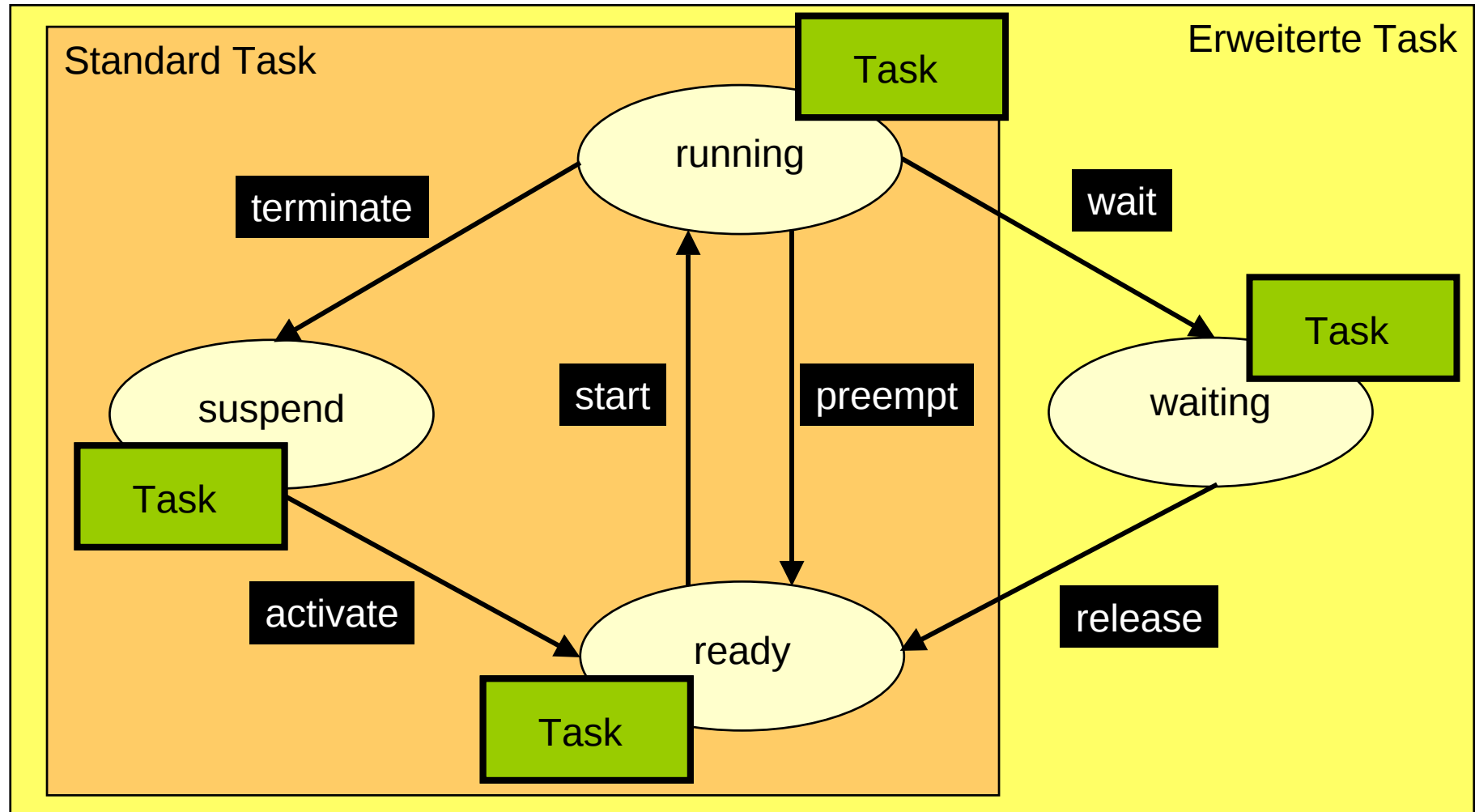


### **Vorstand der Organisation OSEK/VDX (Automobilhersteller und Zulieferer):**

- Adam Opel AG
- BMW AG
- Daimler Chrysler AG
- University of Karlsruhe – IIT
- PSA
- Renault SA
- Robert Bosch GmbH
- Siemens AG
- Volkswagen AG

Informationsquelle im Internet: <http://www.osek-vdx.org>

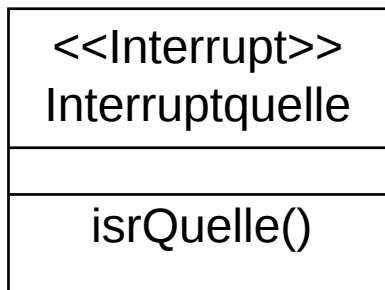
| Dokument                                        | Inhalt                                                                                                                                                                                                |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OS</b>                                       | ( <b>O</b> perating <b>S</b> ystem) Spezifikation der Mechanismen und des APIs für eine OSEK-konforme Betriebssystem-Implementierung                                                                  |
| <b>COM</b>                                      | ( <b>C</b> ommunication) Spezifikation eines Software-Kommunikationsinterfaces zur Vereinfachung der Verbindung verschiedener Systeme                                                                 |
| <b>NM</b>                                       | ( <b>N</b> etwork <b>M</b> anagement) Spezifikation des Konzepts und des APIs für das Netzwerk-Management                                                                                             |
| <b>OIL</b>                                      | ( <b>O</b> SEK <b>I</b> mplementation <b>L</b> anguage) Spezifikation der Konfiguration einer OSEK Applikation und des Betriebssystems                                                                |
| <b>ORTI Part A</b><br><b>ORTI Part B</b>        | ( <b>O</b> SEK <b>R</b> un <b>T</b> ime <b>I</b> nterface) Teil A: Spezifikation der „Kernel Objekt Interface Language“ (KOIL) – Syntax; Teil B: Beschreibung der einzelnen Kernel Objekte - Semantik |
| <b>Binding Document</b>                         | Spezifikation des Zusammenspiels der Spezifikationen                                                                                                                                                  |
| <b>OSEK/VDX Time-Triggered Operating System</b> | Spezifikation der Mechanismen und des APIs für eine OSEKtime-konforme Betriebssystem-Implementierung                                                                                                  |
| <b>OSEK/VDX Fault Tolerant Communication</b>    | Spezifikation einer fehlertoleranten Kommunikationsschicht                                                                                                                                            |



Prioritätsbasierendes Scheduling ( preemptive / non-preemptive )



## Bearbeitung mit Interrupt-Serviceroutine (Kategorie 1)



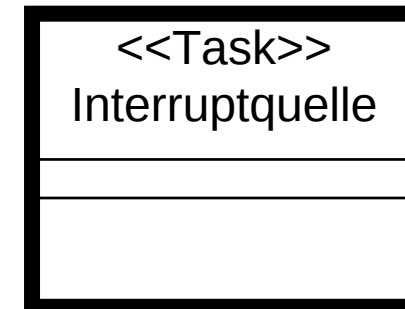
### Vorteil:

- Schnellste Bearbeitung

### Nachteil:

- Keine Betriebssystem-Unterstützung
- Je nach HW-Architektur feste ISR-Zuweisung

## Bearbeitung mit Task in der Laufzeitumgebung (Kategorie 2)



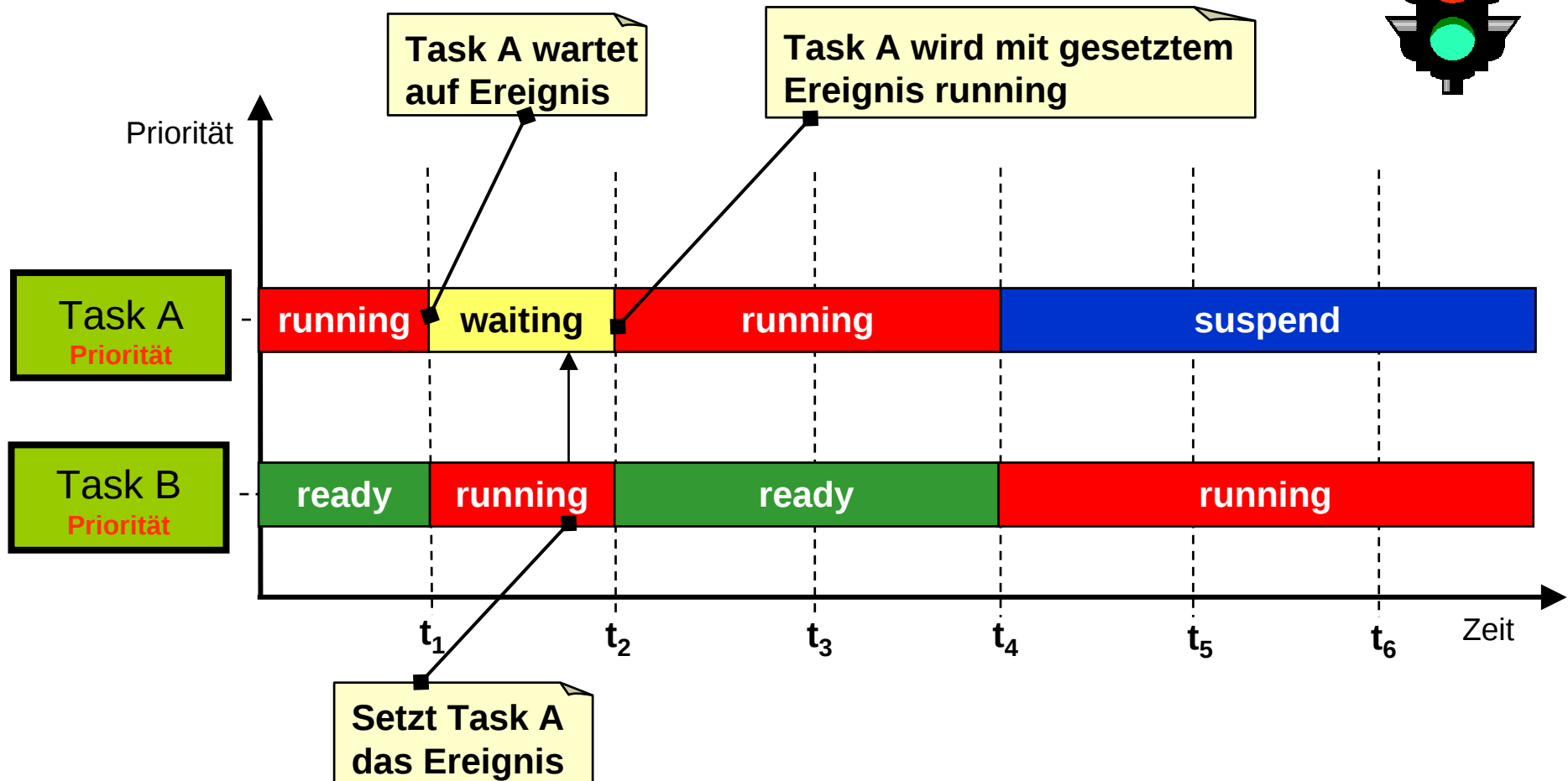
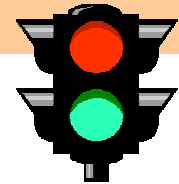
### Vorteil:

- Betriebssystem-Unterstützung
- Interruptkanal ist nach Taskaktivierung wieder frei
- Interruptbearbeitung zur Laufzeit austauschbar

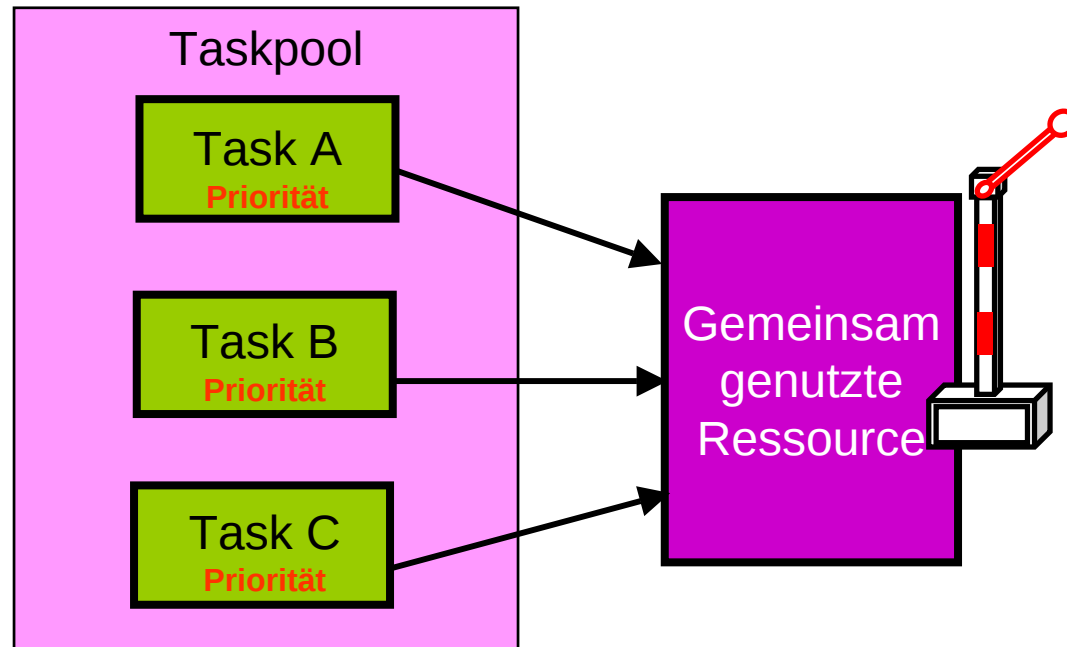
### Nachteil:

- Langsame Bearbeitung, bedingt durch Scheduling

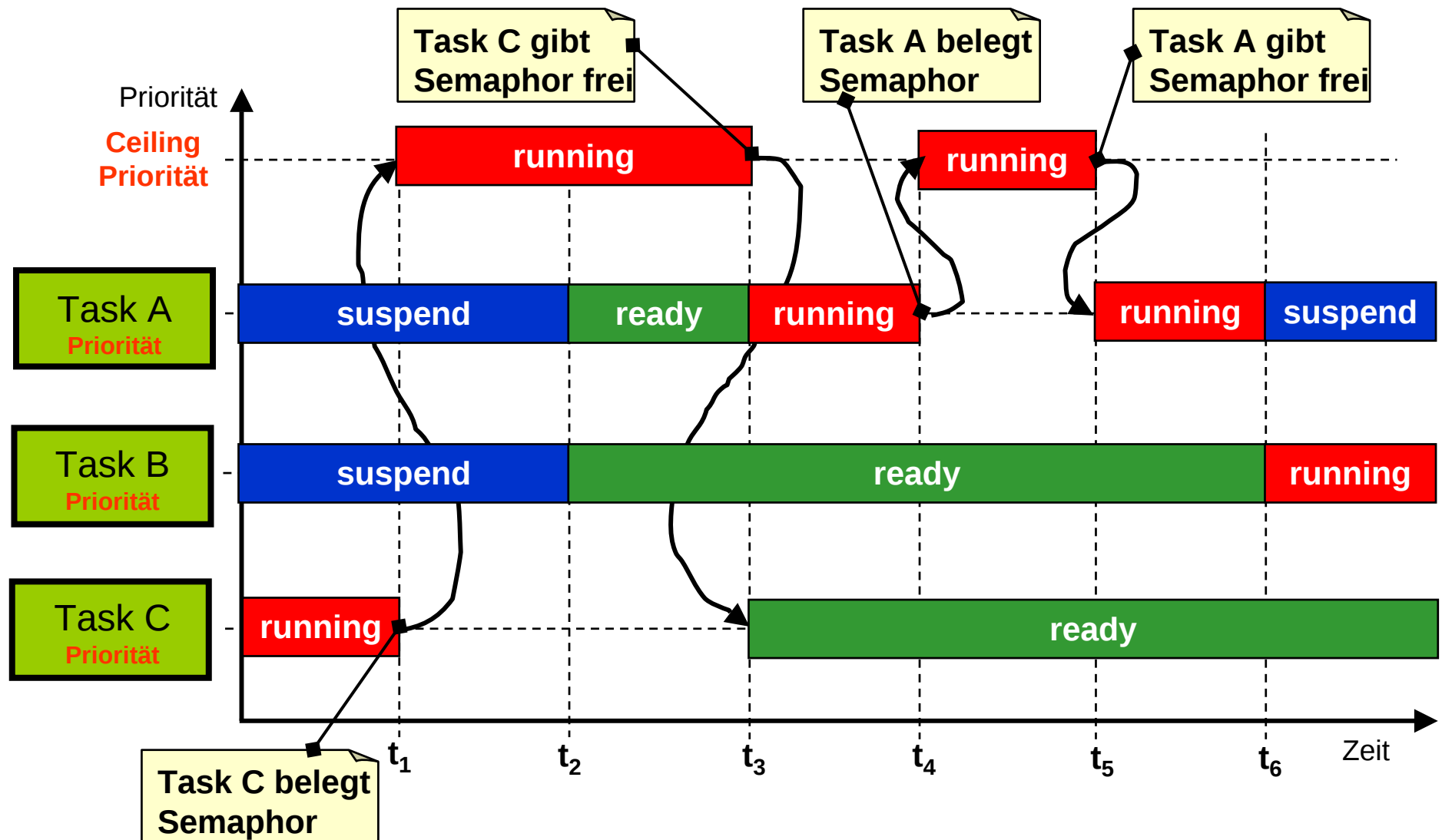
- Synchronisationsmechanismus
- Nur für erweiterte Task



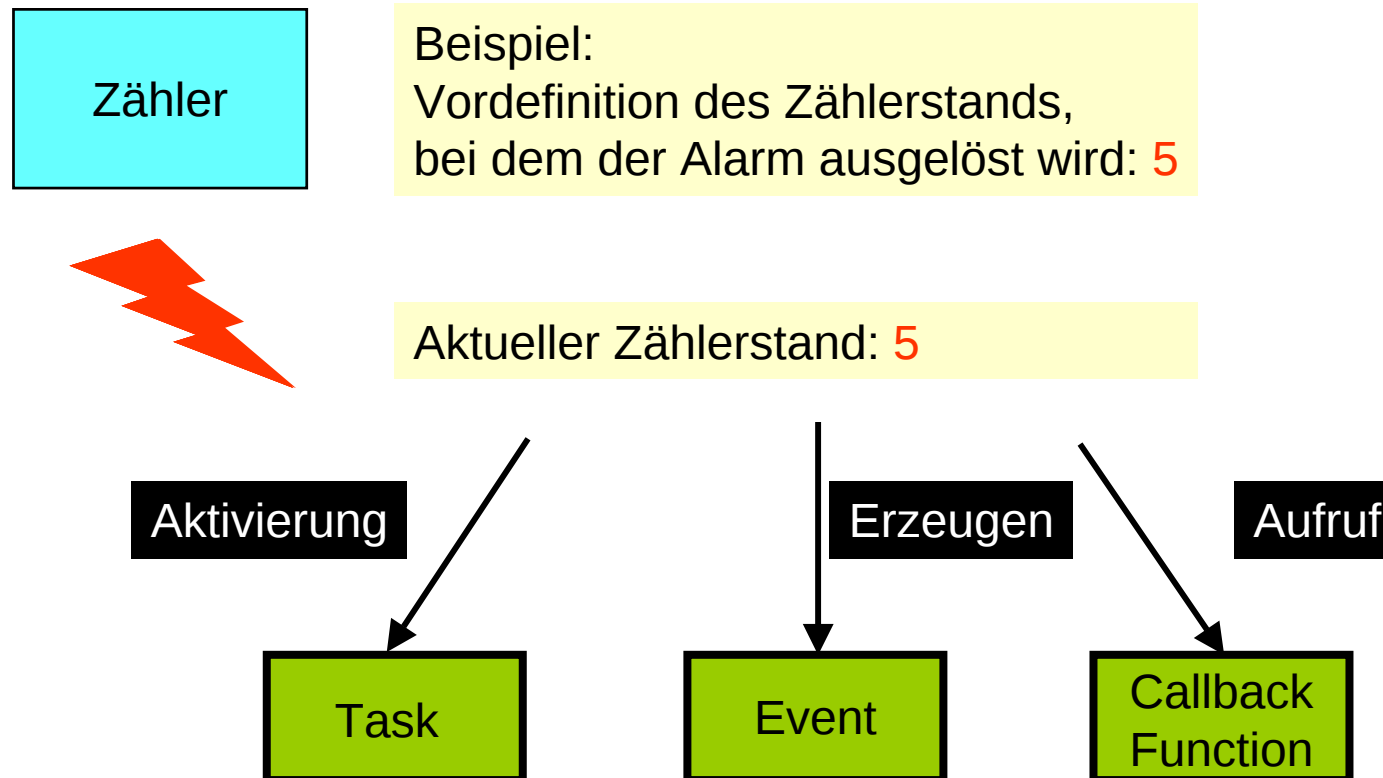
- Koordinationsmechanismus: Semaphore (Mutex)
- Schutz vor gleichzeitigem Mehrfachzugriff auf gemeinsam genutzte Ressourcen

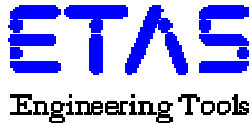


Forderung: Keine Task im Waiting Zustand wegen belegter Ressource  
Lösung: Priority Ceiling Protocol (Prioritätsschranken-Algorithmus)  
Bedingung:  $\text{Ceiling Priorität} \geq \text{max. Priority [Taskpool]}$



- Mechanismus für die Behandlung von wiederkehrenden Ereignissen
- Realisierung durch die Implementierung von Zählern



|                                                                                                                                   |                                                                                      |  |
|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--|
| <b>ERCOS<sup>EK</sup></b><br>ETAS GmbH<br><a href="http://www.etas.de">www.etas.de</a>                                            |    |  |
| <b>OSEKturbo</b><br>Metrowerks<br><a href="http://www.metrowerks.com">www.metrowerks.com</a>                                      |    |  |
| <b>osCAN, osCANopen</b><br>Vektor Informatik<br><a href="http://www.vector-informatic.com">www.vector-informatic.com</a>          |    |  |
| <b>OSEKWorks</b><br>WindRiver<br><a href="http://www.windriver.com">www.windriver.com</a>                                         |    |  |
| <b>Nucleus OSEK</b><br>Accelerated Technology<br><a href="http://www.acceleratedtechnology.com">www.acceleratedtechnology.com</a> |   |  |
| <b>ProOSEK</b><br>3Soft GmbH<br><a href="http://www.3soft.de">www.3soft.de</a>                                                    |  |  |

|                                                                                                                |                                                                                    |                                                                                           |
|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>Real-Time Architect</b><br>Livedevices (ETAS Group)<br><a href="http://www.realogy.com">www.realogy.com</a> |  |                                                                                           |
| <b>OSEKplus</b><br>EUROS Embedded Systems GmbH<br><a href="http://www.kaneff.de">www.kaneff.de</a>             |  |                                                                                           |
| <b>P4/OSEK</b><br>SYSGO AG<br><a href="http://www.sysgo.de">www.sysgo.de</a>                                   |  | <b>PikeOS</b> integriert die Ausführung<br>verschiedener Betriebssysteme<br>auf einer CPU |

## Achtung:

**Nicht jedes Produkt, welches im Namen „...OSEK...“ enthält, ist auch nach OSEK zertifiziert.  
Eine aktuelle Liste OSEK-zertifizierter Systeme ist im Internet bereitgestellt unter:**

[www.osek-vdx.org](http://www.osek-vdx.org)

OSEK/VDX ist ein Standard, nach dem ein OSEK/VDX-konformes Betriebssystem implementierbar ist.

Geschwindigkeit, geringer Ressourcenverbrauch, Zuverlässigkeit und Sicherheit sind die wichtigsten Anforderungen an ein Echtzeitbetriebssystem für den Einsatz im Kraftfahrzeug.

Ursprung des Standards sind die Kraftfahrzeugfirmen und deren Zulieferer.

Eingesetzt werden OSEK/VDX-konforme Betriebssysteme vorwiegend in Steuersystemen im Kraftfahrzeug.

Die vorteilhaften Eigenschaften eines OSEK/VDX-konformen Betriebssystems haben inzwischen aber auch andere Industriezweige für sich entdeckt.

## OSEK-Grundlagen

### Trainingsinhalt:

- Allgemeine Einführung in Echtzeitbetriebssysteme
- Mechanismen und Betriebsmittel eines nach dem OSEK-Standard implementierten Echtzeitbetriebssystems
- Network Management
- Communication
- OIL (OSEK Implementation Language)
- Analyse und Design von Embedded- und Echtzeitapplikationen unter Berücksichtigung des Echtzeitbetriebssystems
- Zusammenstellung von aktuellen Echtzeitbetriebssystemen, die nach dem OSEK-Standard implementiert sind
- Praktischen Übungen mit einem nach dem OSEK-Standard implementierten Betriebssystem und einer Embedded Hardware

Weitere Informationen finden Sie im Internet: <http://www.microconsult.de>

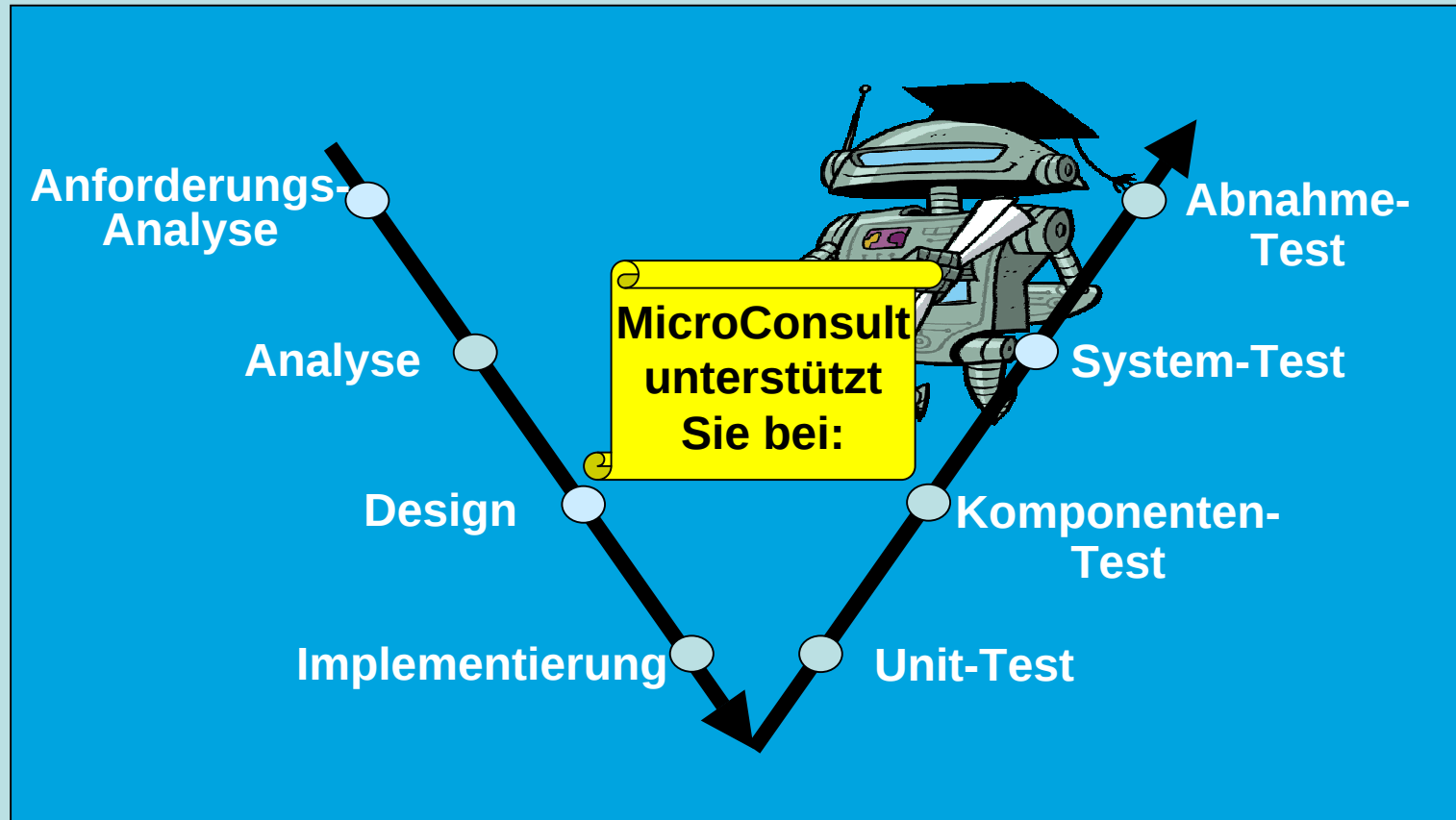
## RTOS-Grundlagen

### Trainingsinhalt:

- Allgemeine Einführung in Echtzeitbetriebssysteme
- Multitasking-Mechanismen
- Interrupt Management
  - Interrupt-Serviceroutinen, Threads
- Time Mangement
  - Systemzeit, Timer- und Watchdog- Dienste
- Task-Koordination
  - Eventflags, Signals, Semaphore
- Task-Kommunikation
  - Mailboxes, Pipes
- Speichermanagement
  - Memory-, Buffer-, Mega- und Shared-Memory-Pools
- Cluster-Architektur, Objektverwaltung
- Port-Treiber-Konzept, Bit-Processing

Weitere Informationen finden Sie im Internet: <http://www.microconsult.de>

## Training, Coaching, Engineering



HW-/SW-Technologien, Tools, Methoden, Prozess, Team