Virtualisierung im Echtzeitbereich

Andreas Hollmann
FH Landshut
EADS Military Air Systems

Überblick

- Hintergrund und Motivation
- Vorstellung von Lösungsansätzen
- Auswahl und Evaluierung
- Einschränkungen

Echtzeit-Virtualisierung

- "gleichzeitige" Ausführung mehrerer (echtzeitfähiger) Betriebssysteme auf einem physikalischen System unter Beibehaltung der Echtzeitfähigkeit des Gesamtsystems
  - mehrere Instanzen von RTOS (auch heterogen)
  - Kombinationen aus GPOS und RTOS

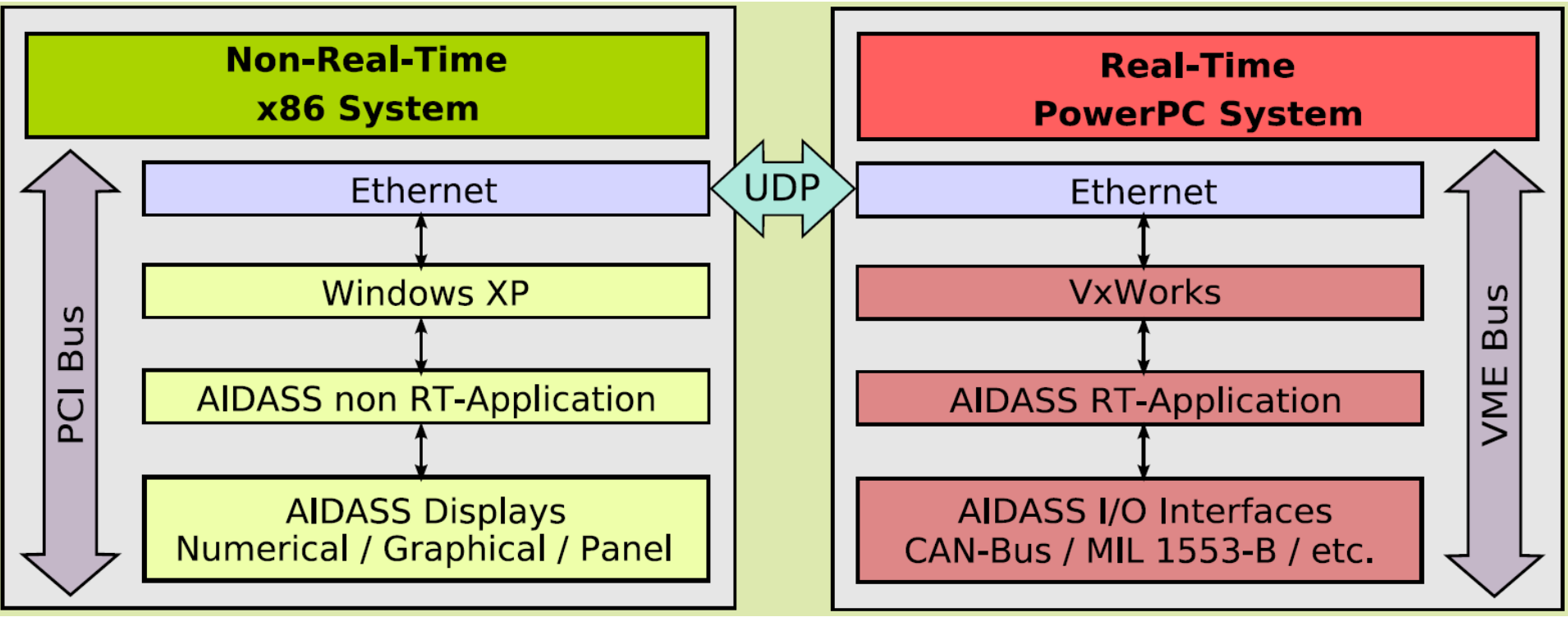
Anwendungsziele

- Konsolidierung echtzeitfähiger Systeme auf einem physikalischen Zielsystem
  - Verringern der Ausgaben für Hardware und Wartung
  - Minimierung des Platz- und der Energiebedarfs
- Beispiele
  - Consumer-Hardware (Handy)
  - Medizinische Systeme (Ultraschall)
  - Testsysteme
    - EADS AIDASS 2000

AIDASS 2000

- **A**dvanced **I**ntegrated **D**ata **A**cquisition and **S**timulation **S**ystem
- Bodentestsystem (Entwicklung bis Wartung)
- Datenerfassung, Stimulation, Simulation, Visualisierung und Auswertung
- besteht aus 2 Teilsystemen – echtzeitfähigem System (VxWorks / PowerPC / VMEbus) + Windows Applikation (Windows XP / x86 PC)

AIDASS 2000


Non-Real-Time
x86 System
Ethernet
Windows XP
AIDASS non RT-Application
AIDASS Displays
Numerical / Graphical / Panel
PCI Bus
UDP
Real-Time
PowerPC System
Ethernet
VxWorks
AIDASS RT-Application
AIDASS I/O Interfaces
CAN-Bus / MIL 1553-B / etc.
VME Bus

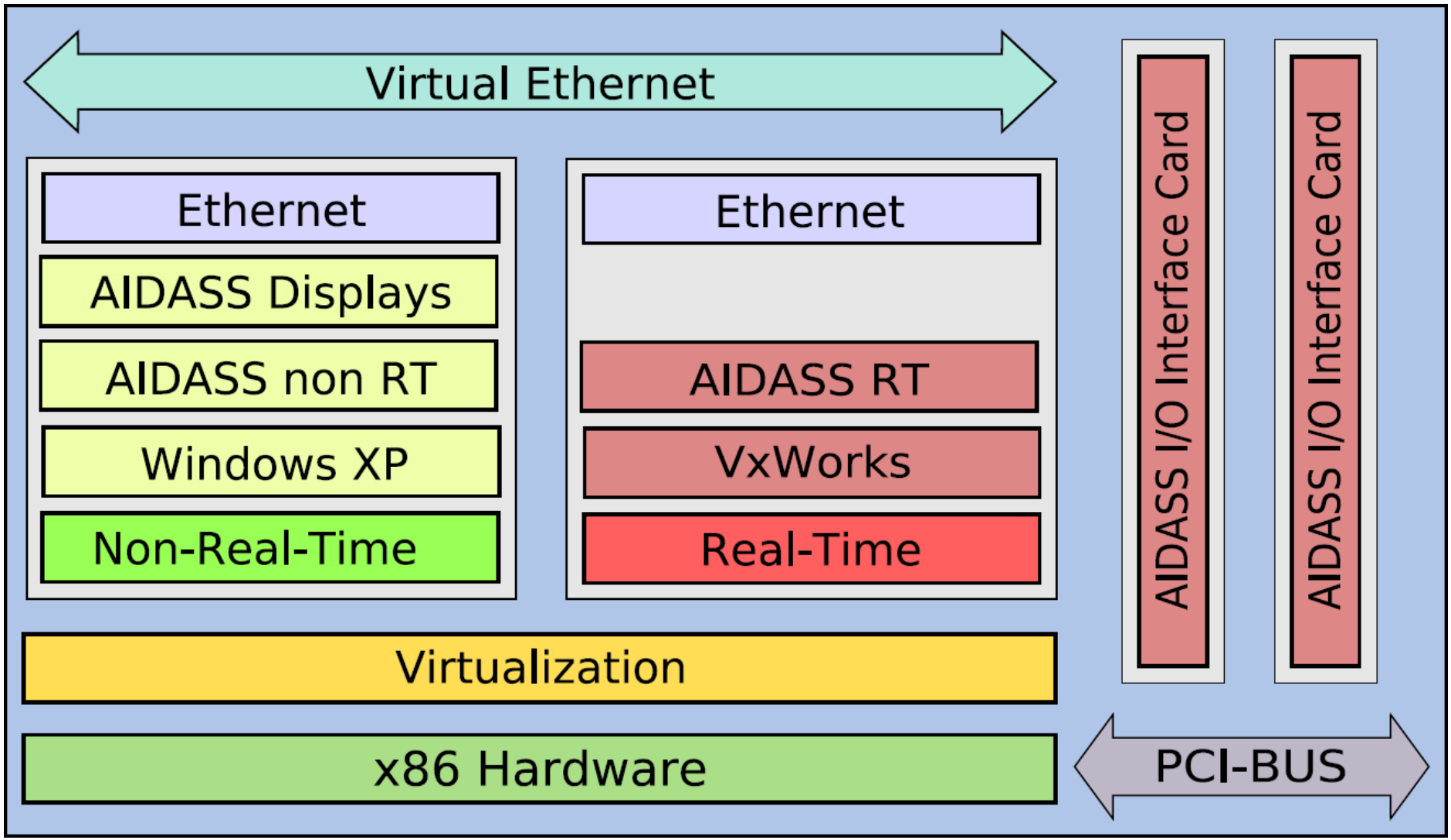
AIDASS 2000

AIDASS 2000

- sehr flexible, gut erweiterbare Lösung aus hochwertigen Komponenten
- hohe Kosten aufgrund geringer Stückzahlen
- zwei physikalische Systeme notwendig, selbst bei kleinen Testsystemen

- Optimierung durch Echtzeit-Virtualisierung möglich ???

AIDASS 2000 - Virtualisiert



Vorzüge der x86-Plattform

- rasant steigende Performance
- sinkende Leistungsaufnahme
  - Intel Atom (2 Watt TDP)
- Verfügbarkeit von Multi-Core Prozessoren
  - erlaubt echte parallele Verarbeitung
- Virtualisierungs-Erweiterungen in Hardware
  - ermöglichen es Hardwarezugriffe zu kontrollieren
- Embedded Initiative von Intel
  - verlängerte Verfügbarkeit der Hardware

Ziel der Virtualisierung

- „konventionelle“ Virtualisierung
  - sichere Abschottung der Gastsysteme hat die höchste Priorität – auf Kosten der Leistungsfähigkeit
- Echtzeit-Virtualisierung
  - möglichst geringe Latenzzeiten als wichtigstes Kriterium
  - unter Umständen auch unter Aufweichung der sicheren Abschottung

Konzepte der Virtualisierung

- Systemvirtualisierung (VMware, VirtualBox, Xen)
  - Illusion eines vollständigen x86-Systems
    - CPU und Speicher können direkt genutzt werden
    - restliche Hardware (z. B. I/O) muss in Software emuliert werden
- Paravirtualisierung (Xen, RTOS-Virtualsierung)
  - anpassen der Gastbetriebssysteme
    - Schaffung effizienter Schnittstellen zwischen Wirtssystem und Gastbetriebssystem
    - Quelltext der Betriebssysteme muss verfügbar sein

Lösungsansätze

- Windows XP als Host-System
- Microkernel als Hypervisor
- Partitionierung auf Multi-Core-Systemen

Windows XP als Host-System

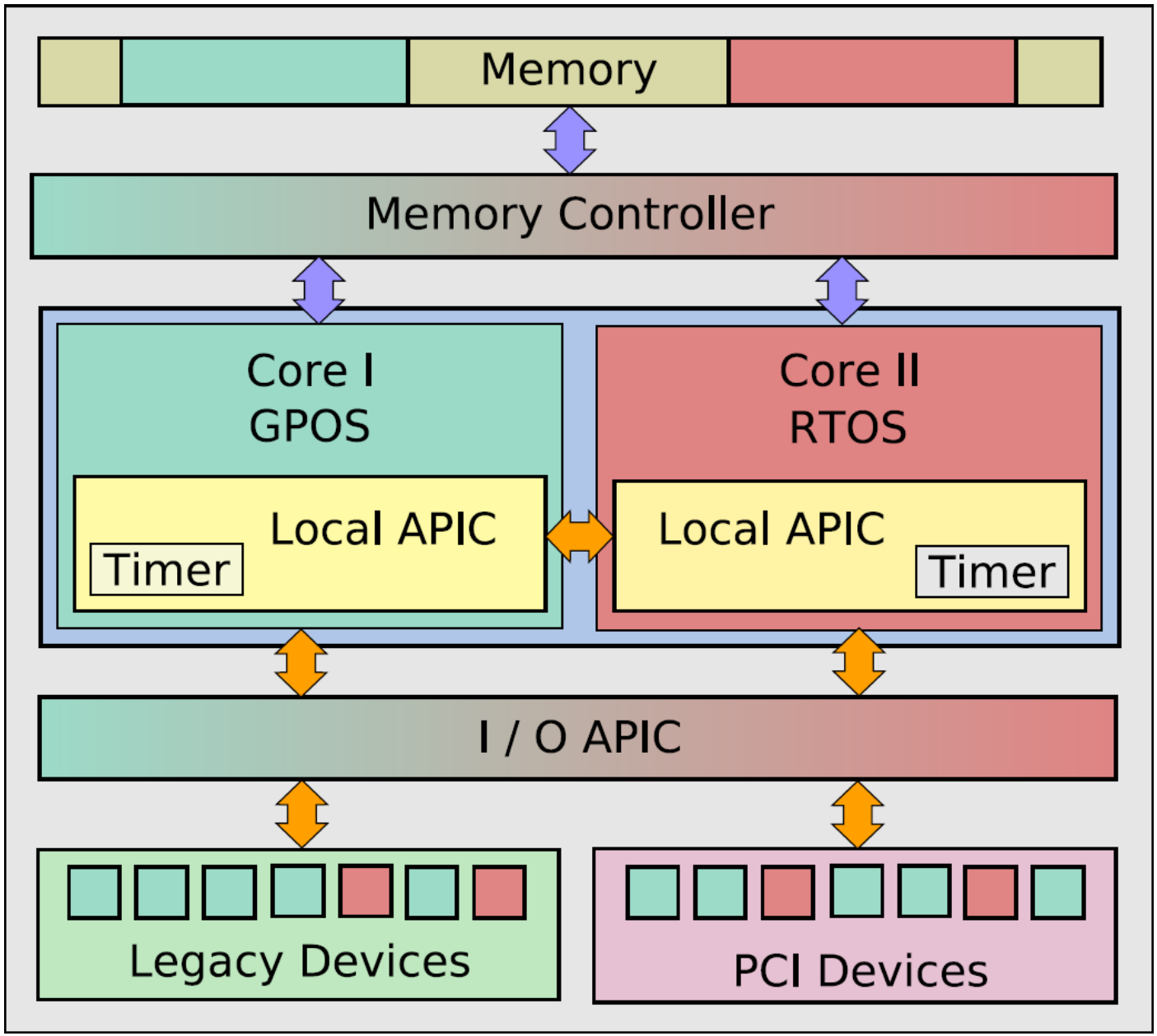
- Anpassung von Windows und RTOS - Zugriff vom RTOS auf Hardware wird über Schnittstellen ermöglicht
- Ablaufsteuerung
  - 1 CPU-Kern – Windows als Idle-Task des RTOS
  - 2+ Kerne – exklusiver Kern für das RTOS
- KUKA RTOSWin Product Family
  - VxWin, CeWin, QWin, (On Time) RTOS32Win
  - keine Weiterentwicklung bis Mitte 2008
- TenAsys eVM Virtualization Platform (2008)

Microkernel als Hypervisor

- Abstraktion und Verwaltung wichtiger Hardware (z. B. IRQ-Controller) durch einen relativ kleinen Kernel
- Zugriff auf abstrahierte Hardware über Schnittstellen
- RTOS läuft so weit wie möglich nativ auf der Hardware
- Interrupts werden vom Hypervisor entgegengenommen und an das Ziel weitergeleitet
  - zusätzliche Latenzen !
- VirtualLogix VLX for Network Infrastructure

Partitionierung mit Multi-Core-CPUs

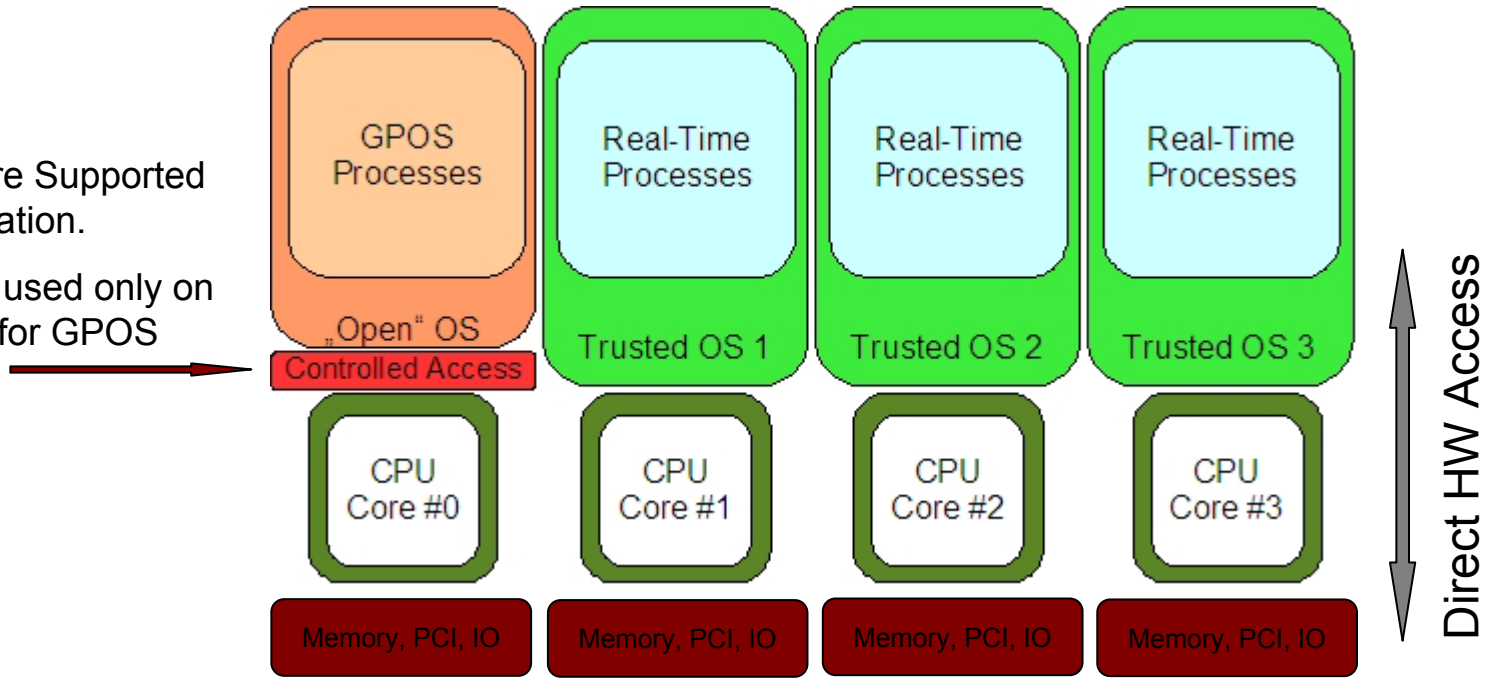
- Partitionierung des Systems auf Hardware-Ebene
  - CPU – mindestens eine CPU pro Betriebssystem
  - Speicher – disjunkte Adressbereiche
  - Timer – pro CPU 1x TSC + APIC Timer für Interrupts
  - Geräte und Interrupts – zur Boot-Zeit wird der APIC-Controller konfiguriert und die Interrupt-Line an CPU-Kerne zugewiesen
- Kommunikation über Shared Memory oder Virtual Network
- Real-Time Systems Hypervisor


Memory
Memory Controller
Core I
GPOS
Local APIC
Timer
Core II
RTOS
Local APIC
Timer
I / O APIC
Legacy Devices
PCI Devices

Direkter Hardwarezugriff

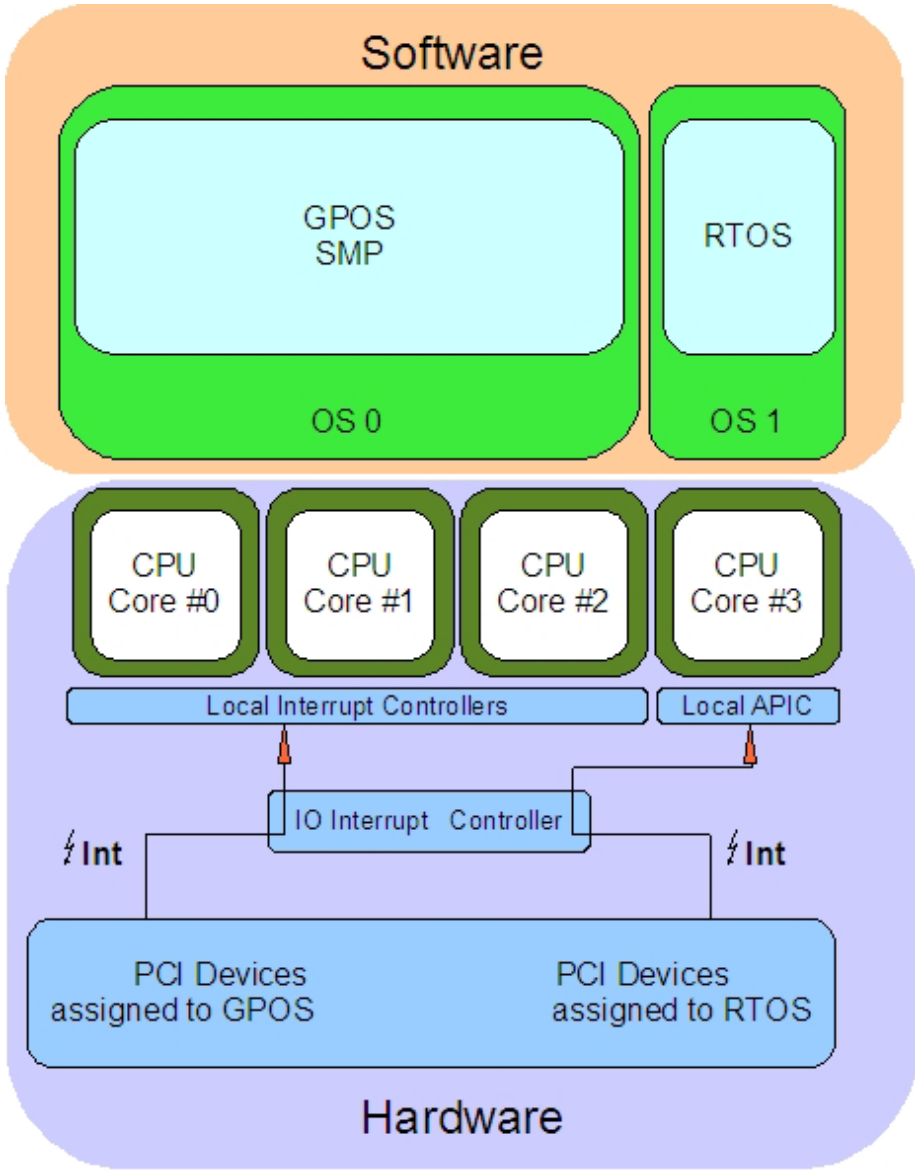
Hardware Supported Virtualization.

Intel-VT used only on Core(s) for GPOS



Example of a Quad-Core Processor System

Interrupt Behandlung



Einschränkungen

- gemeinsame Ressourcen als Flaschenhals
  - Bus-Systeme (PCI, Chipsatzbus)
  - Last-Level-Cache
  - Speicherbandbreite
- alle Geräte an einer Interrupt-Line, werden einem Betriebssystem zugeordnet
- „Triple-Fault“ in einem der Betriebssysteme führt zu einem Reboot des gesamten Systems

Fazit

- Latenzen sehr stark abhängig vom verwendeten Mainboard (Firmware)
  - bei geeigneter Hardware keine messbare zusätzliche Latenz
- Interrupt-Routing des Mainboards entscheidend
- Komplexere Analyse des Systems
- bei geeigneter Hardware für weniger kritische Applikationen geeignet

Vielen Dank für die Aufmerksamkeit!