



# CyPhyControl

Virtualisierte Ausführungsplattform für die zuverlässige Steuerung  
cyber-physikalischer Systeme

---

Olaf Spinczyk  
Markus Buschhoff  
Boguslaw Jablkowski



AG Eingebettete Systemsoftware  
Informatik 12, TU Dortmund



# Agenda

- Einführung
- Demo
- Durchführung
- Seminarthemen
- Fragen



# Cyber-physikalische Systeme

- Definition:
  - “Cyber-Physical Systems (CPS) are integrations of computation with physical processes.” - Edward A. Lee
- Technologien:
  - Eingebettete Systeme
  - Kommunikationstechnik





# Cyber-physikalische Systeme



- Anwendungsbereiche:
  - Automotive- und Avionik-Systeme
  - Produktionsanlagen
  - Energienetze
  - Robotik und viele weitere



# Cyber-physikalische Systeme

- Herausforderungen:
  - Echtzeitfähigkeit
    - Automobilsystem: Airbag muss zwischen 20 und 50 ms voll entfaltet werden
    - Energietechnik: Distanzschutz, Antwortzeit  $< 30$  ms
  - Fehlertoleranz
    - Softwarefehler: Mars Climate Orbiter (MCO)- zerbrach in der marsianischen Atmosphäre aufgrund eines Einheitenfehlers.
    - Hardwarefehler: Ausfall eines Servers mit kritischen Diensten.



NASA – Mars Climate Orbiter

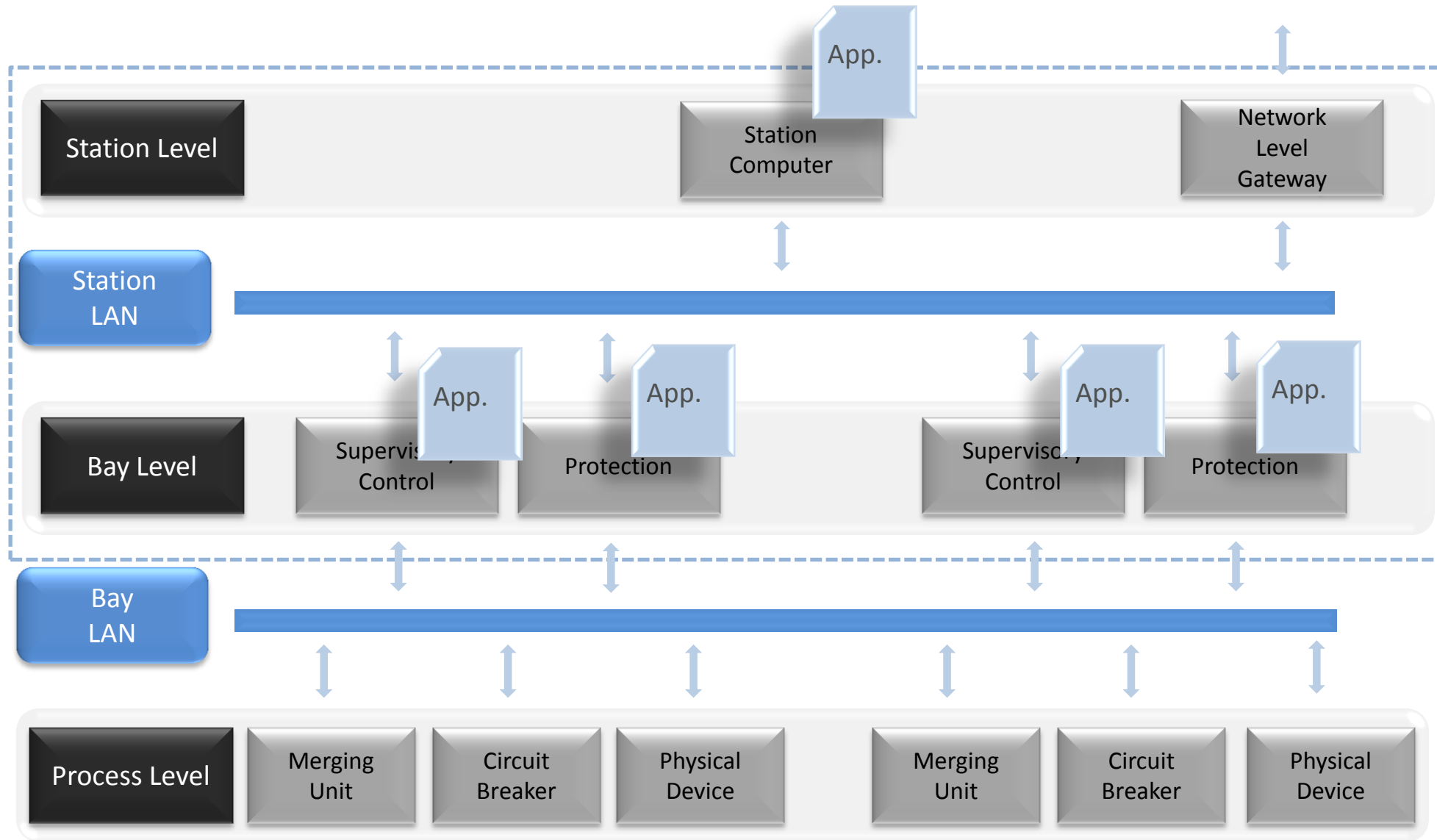


# CPS - Schaltanlage

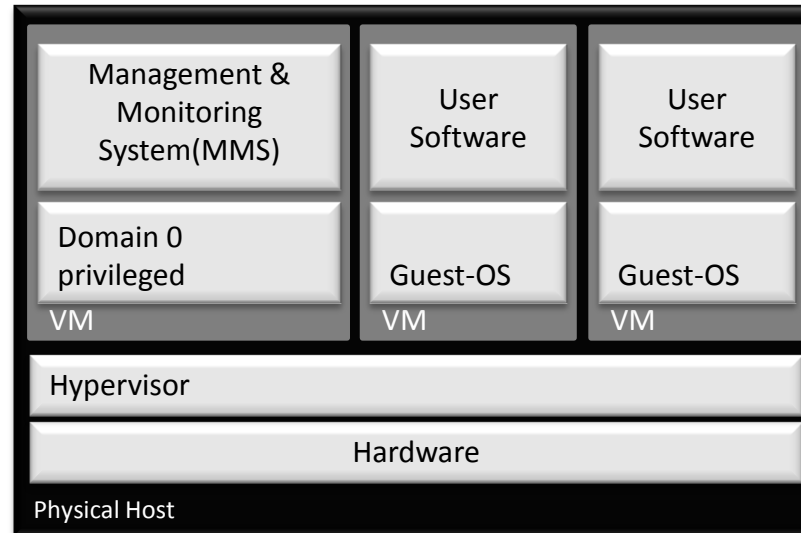
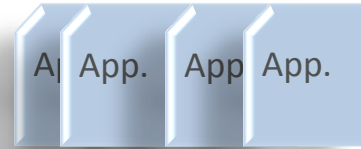


- Verteilung und Umspannung elektrischer Energie
- Netzschutzsysteme

# CPS - Schaltanlage



# Ansatz Virtualisierung







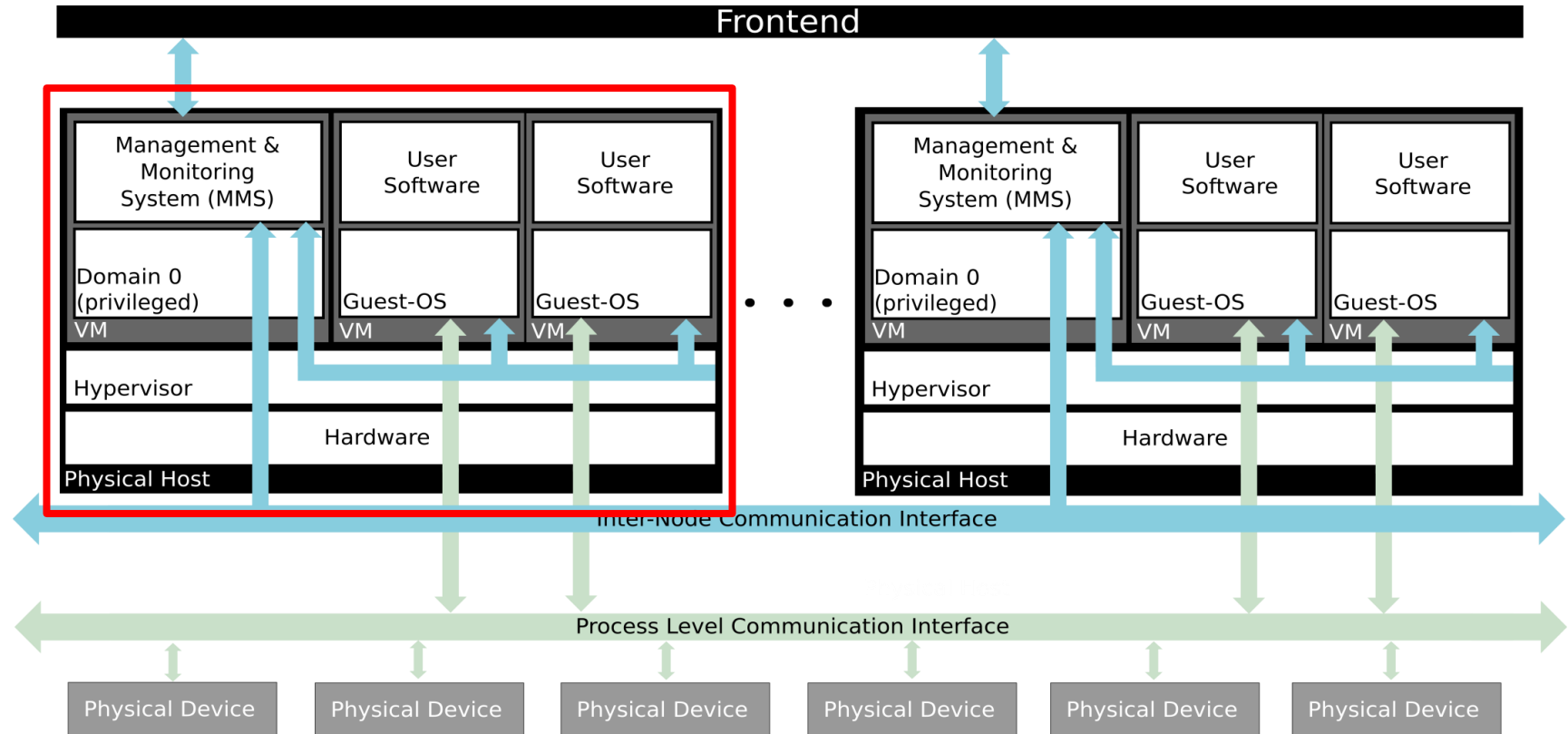
# Virtualisierung

- eine Technik, welche das Abstrahieren von Ressourcen erlaubt
- **virtuell** in Duden „nicht echt, nicht in Wirklichkeit vorhanden, aber echt erscheinend“

## Systemvirtualisierung

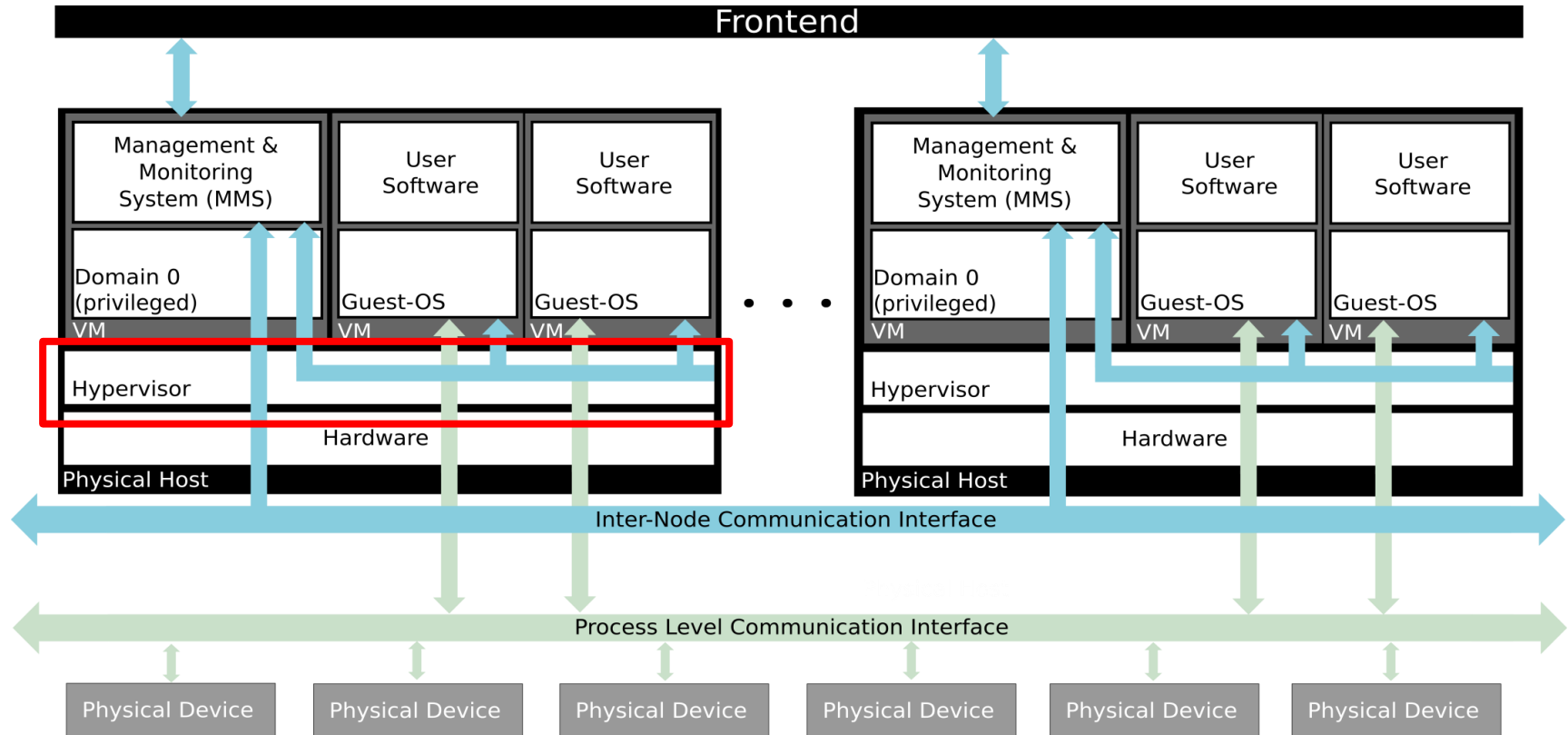
- mehrere Systeminstanzen auf einem Rechner
- Virtuelle Maschine (VM)
  - Betriebssystem
  - Anwendung(en)
- einzelnen VMs (Gastsystemen) wird ein gesamter Rechner mit allen Hardwarekomponenten vorgespiegelt
- Ausführungsumgebung für die VMs wird *Virtual Machine Monitor* oder auch *Hypervisor* genannt

# Beispielarchitektur



- Physischer Host/Hardware
  - x86-Plattform

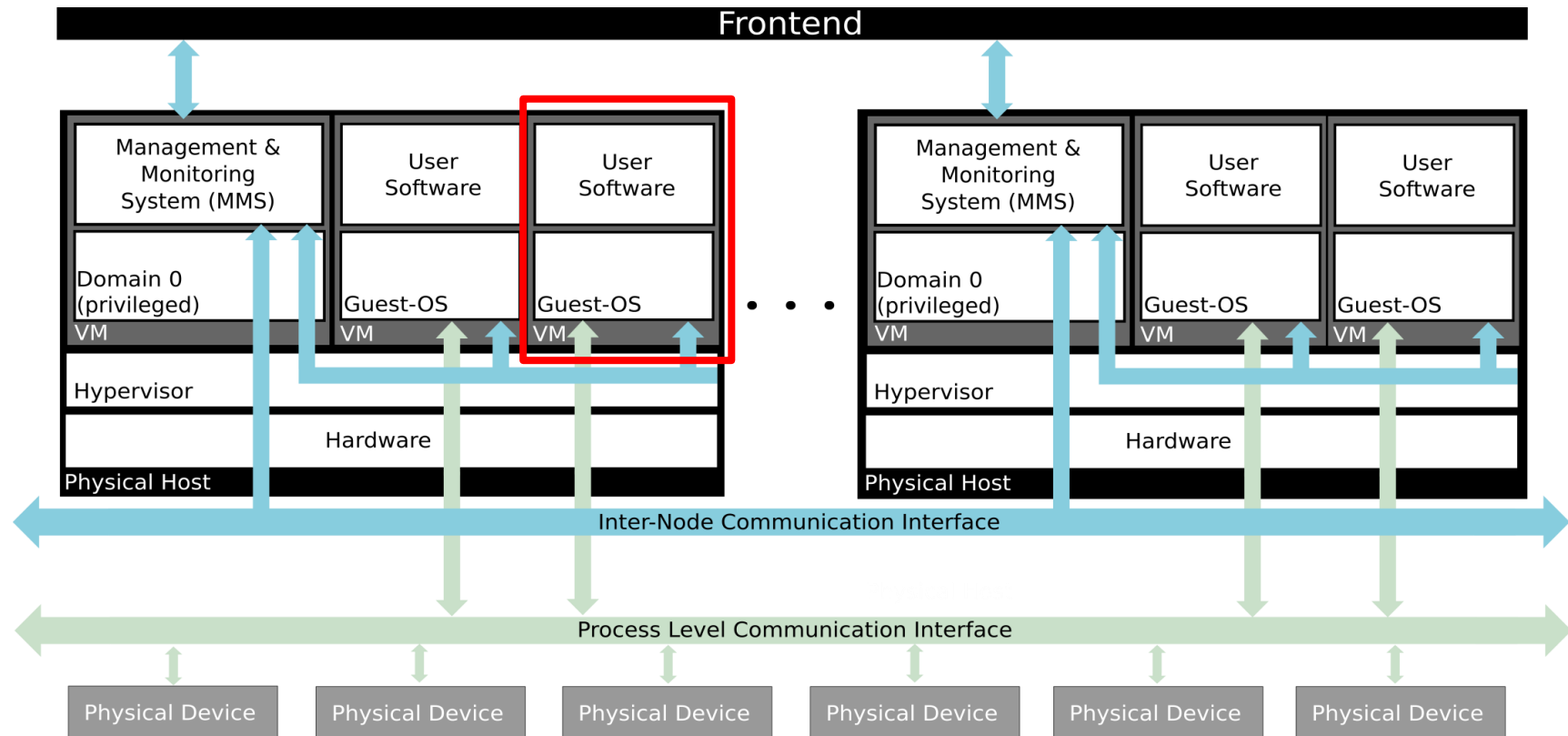
# Beispielarchitektur



- Hypervisor
  - Xen 4.1

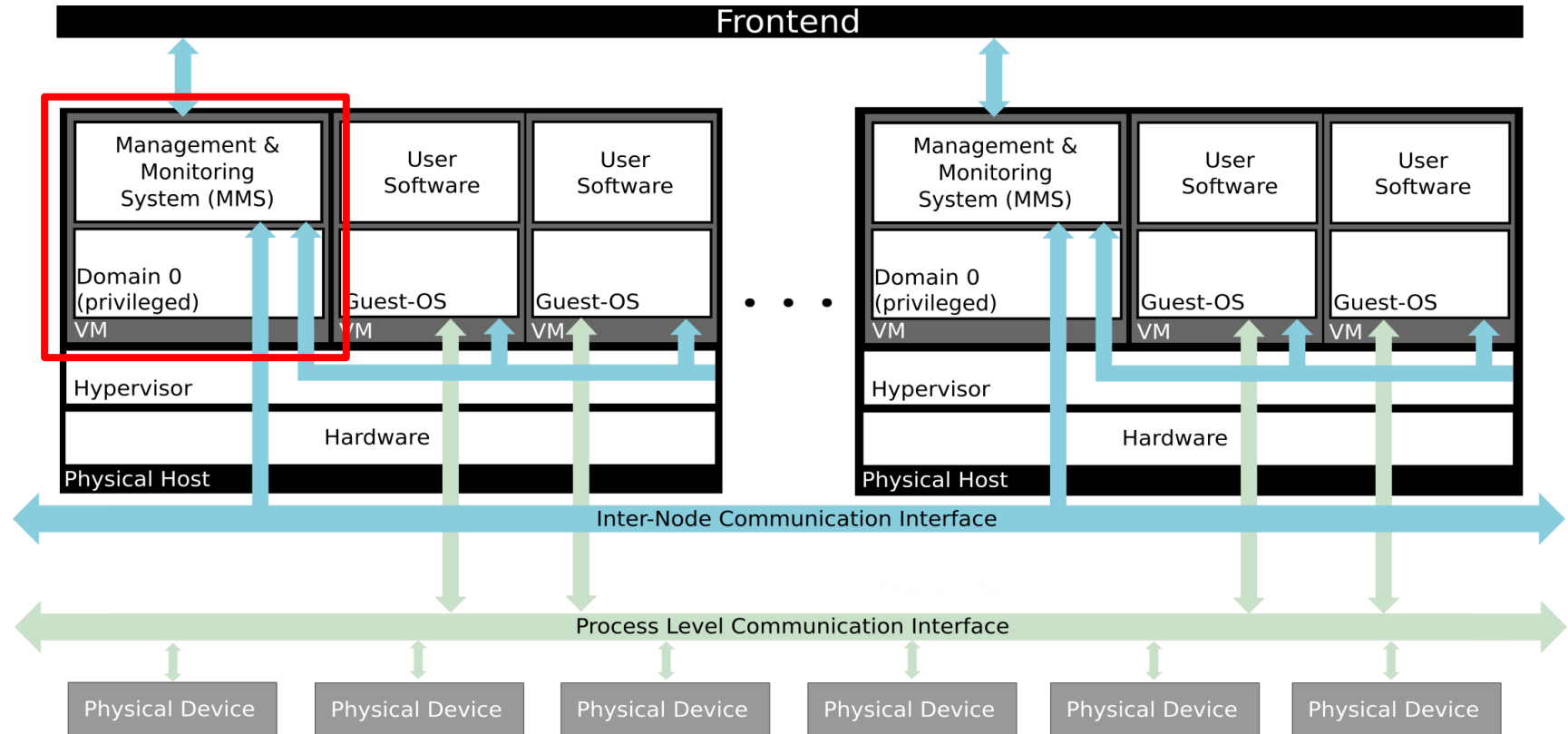


# Beispielarchitektur



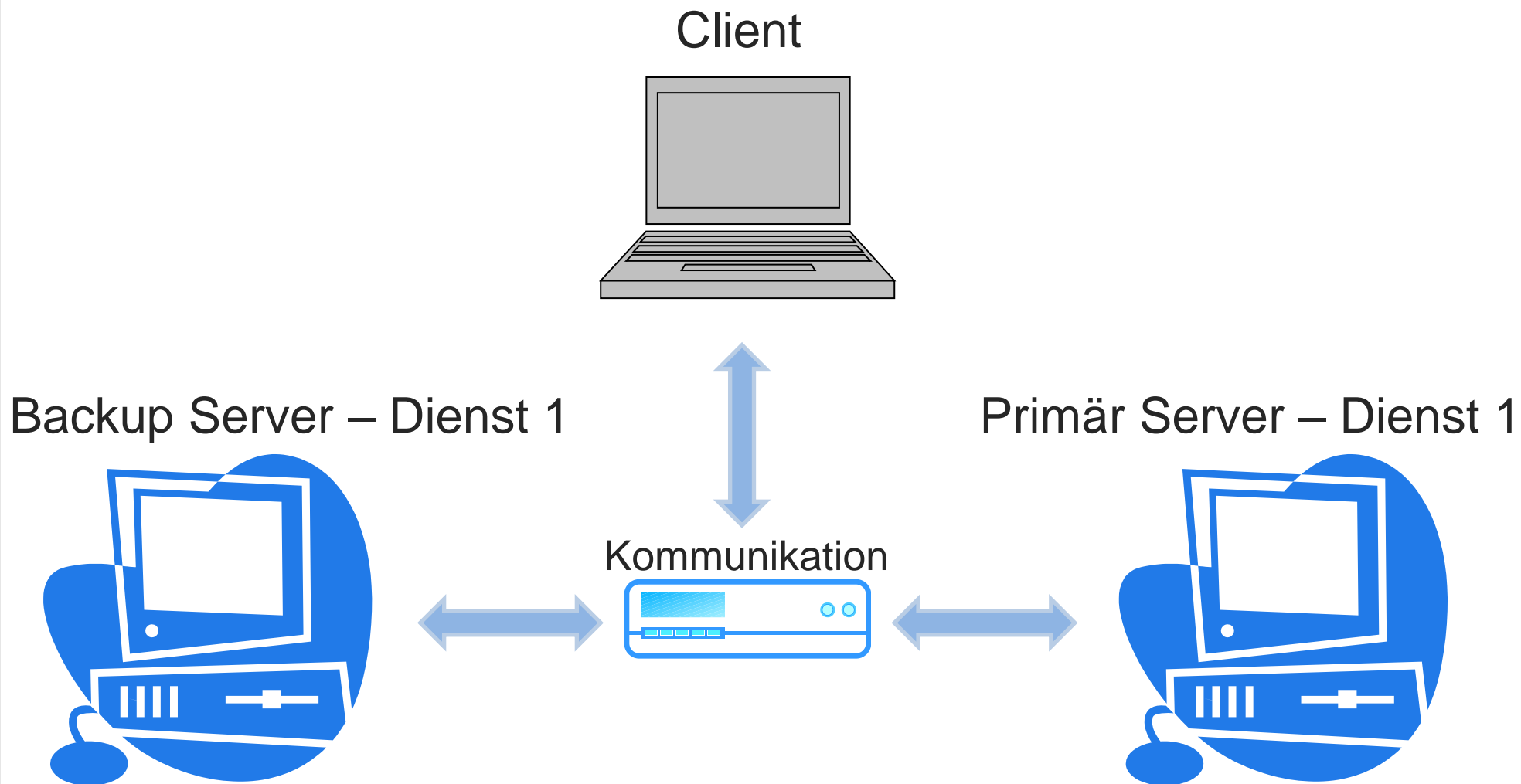
- Virtuelle Maschine (Gastsystem)
  - Linux Betriebssystem
  - Beispielanwendung

# Beispielarchitektur



- Domain 0
- MMS

# Demo Hardware-Fehlertoleranz mit Remus





# Agenda

- Einführung
- Demo
- Durchführung
- Seminarthemen
- Fragen



# Ziele der Projektgruppe

- Entwurf der Ausführungsplattform
- Erstellung der Virtualisierungsinfrastruktur
- Erstellung von Gastsystemen
  - maßgeschneiderte Betriebssysteme
  - Beispielanwendungen
- Entwicklung des Managementsystems
  - Entwurf eines Fehlererkennungskonzeptes
  - Entwurf und Implementierung von Fehlertoleranzmechanismen
  - Entwurf und Implementierung von Protokollen für die Kommunikation, Überwachung und Steuerung der Host-Systeme
- Umsetzung einer GUI für das Managementsystem
- Implementierung und Evaluation eines Prototyps der Plattform



# Forschungsaspekt



# Zeitplan

Aufgabe	Anfang	Ende	Wochen
<i>Einarbeitungsphase</i>			
Einstieg in Literatur, Seminar	KW 15	KW 16	2
Ideensammlung, Umreißung des Grobkonzeptes	KW 17	KW 18	2
Erstellung des Projektplanes	KW 19	KW 19	1
<i>Analyse- und Entwurfsphase</i>			
Analyse der Anforderungen	KW 20	KW 20	1
Auswahl und Bewertung der Techniken	KW 21	KW 21	1
Ausarbeitung des Konzeptes	KW 22	KW 23	2
Entwurf der Architekturkomponenten: Frontend Steuer-/Überwachungskomponente Protokolle	KW 24	KW 28	5
Dokumentation (parallel)	KW 17	KW 29	13

Aufgabe	Anfang	Ende	Wochen
<i>Umsetzungsphase</i>			
Konsolidierung	KW 42	KW 42	1
Implementierung der Architekturkomponenten	KW 43	KW 2	10
<i>Evaluationsphase</i>			
Test und Bewertung des Gesamtsystems	KW 3	KW 5	3
Dokumentation (parallel)	KW 43	KW 6	14



# Dokumentation

## Was muss dokumentiert werden?

- Ausarbeitungen der Seminarthemen
- Protokolle aus den Gruppen- und Plenarsitzungen
- Zwischen- und Endbericht

## Wie soll dokumentiert werden?

- Qualität statt Quantität
  - Auf das Wichtigste reduzieren
- kontinuierlich erweitern



# Infrastruktur

- Labor OH16, Raum E07
- Hardware:
  - Server mit Unterstützung für Hardware-Virtualisierung (VT)
    - 8 Core Xeon mit 16 GB RAM
  - Arbeitsplätze: SunRay Thin Client
  - Desktop-PCs mit VT
- Kooperations-Werkzeuge:
  - Mailingliste
  - Wiki
  - SVN (Versionsverwaltung)



# Agenda

- Einführung
- Demo
- Durchführung
- Seminarthemen
- Fragen



# Seminarthemen

- Virtualisierung
  - Virtualisierung auf Systemebene
  - Ansätze: Xen, Linux KVM, L4 Microkernel
- Maßgeschneiderte Betriebssysteme
- Echtzeitsysteme
  - Scheduling
- Fehlertoleranz
  - Grundbegriffe (fault, dependability, availability, reliability...)
  - fehlertolerante Systeme
- Leistungsbewertung
  - Techniken: simulationsbasierte Methoden, formale Methoden

# Highlights

- großes Projekt im Team realisieren
- Entwicklung einer komplexen Architektur
- breites Spektrum an Fragestellungen und Aufgaben:
  - Virtualisierung, Betriebssysteme, Fehlertoleranz, Echtzeitsysteme, Leistungsbewertung
- PG-Thema relevant für Forschung und Industrie
  - Forschergruppe: “Schutz- und Leitsysteme zur zuverlässigen und sicheren elektrischen Energieübertragung”
- bei Interesse Exkursion zur Schaltleitung Arnsberg (RWE)
  - Überwachung- und Steuerungssysteme in Stromnetzen
- Themen für Diplom- und Masterarbeiten



# Fragen?

## Kontakt

[olaf.spinczyk@tu-dortmund.de](mailto:olaf.spinczyk@tu-dortmund.de)

(0231) 755-6322

[markus.buschhoff@tu-dortmund.de](mailto:markus.buschhoff@tu-dortmund.de)

(0231) 755-6317

[boguslaw.jablkowski@tu-dortmund.de](mailto:boguslaw.jablkowski@tu-dortmund.de)

(0231) 755-6330