barkhauseninstitut.org

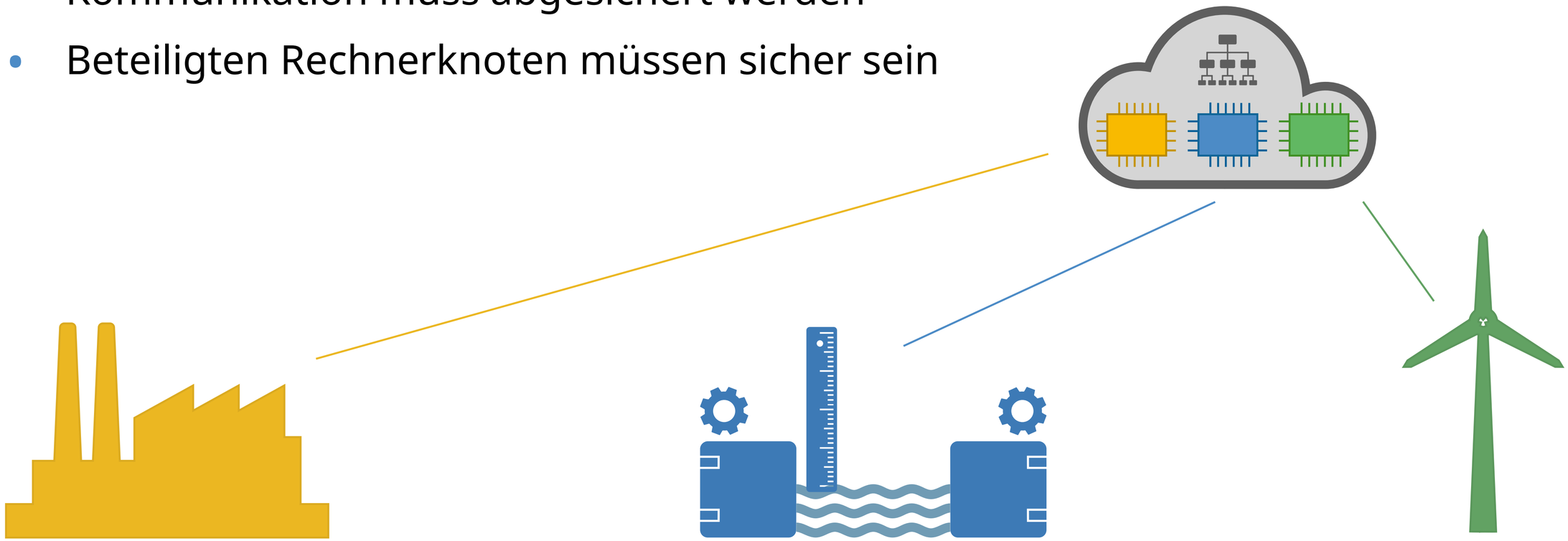
barkhausen institut

Hardware/Software Co-Design für eine modulare Systemarchitektur

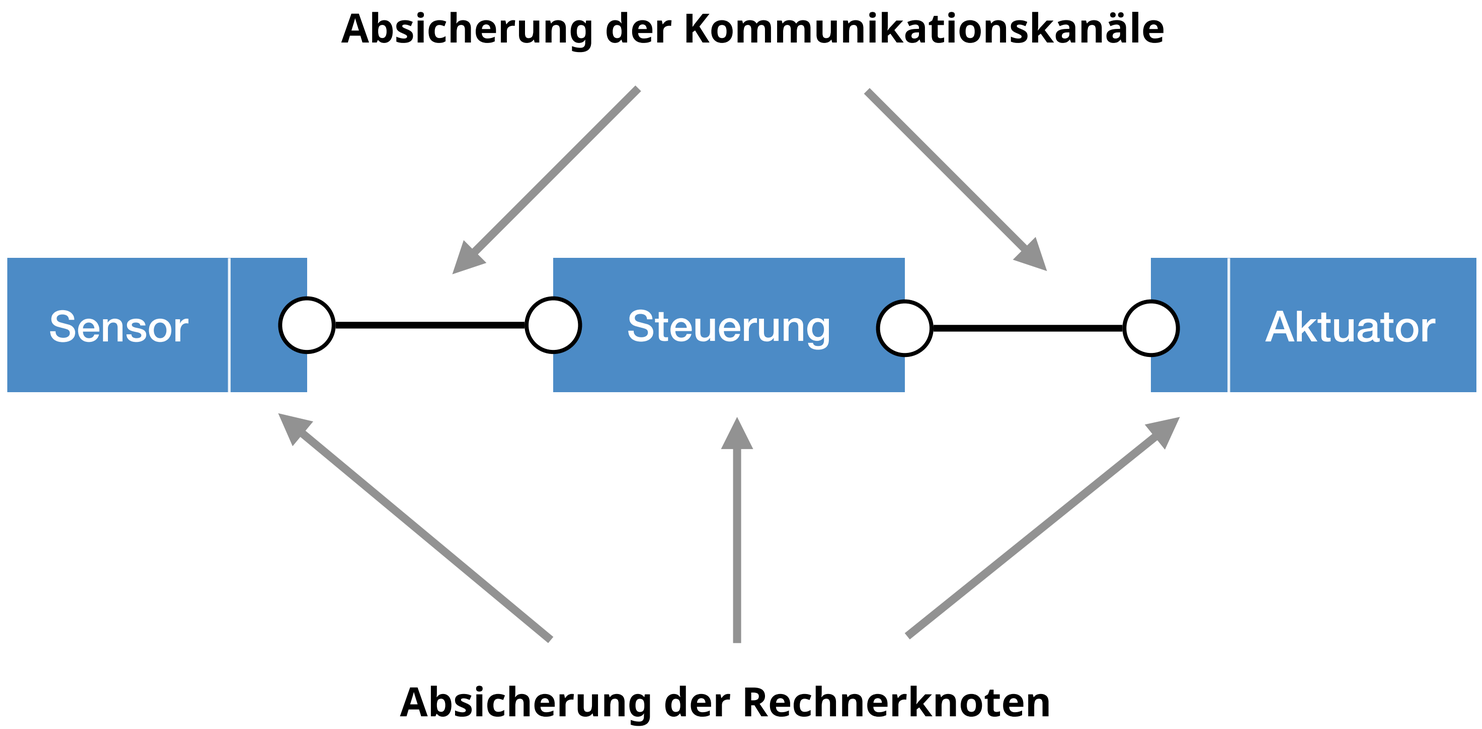
Carsten Weinhold, Nils Asmussen, Michael Roitzsch

Cyber-Physikalische Systeme und das Internet der Dinge

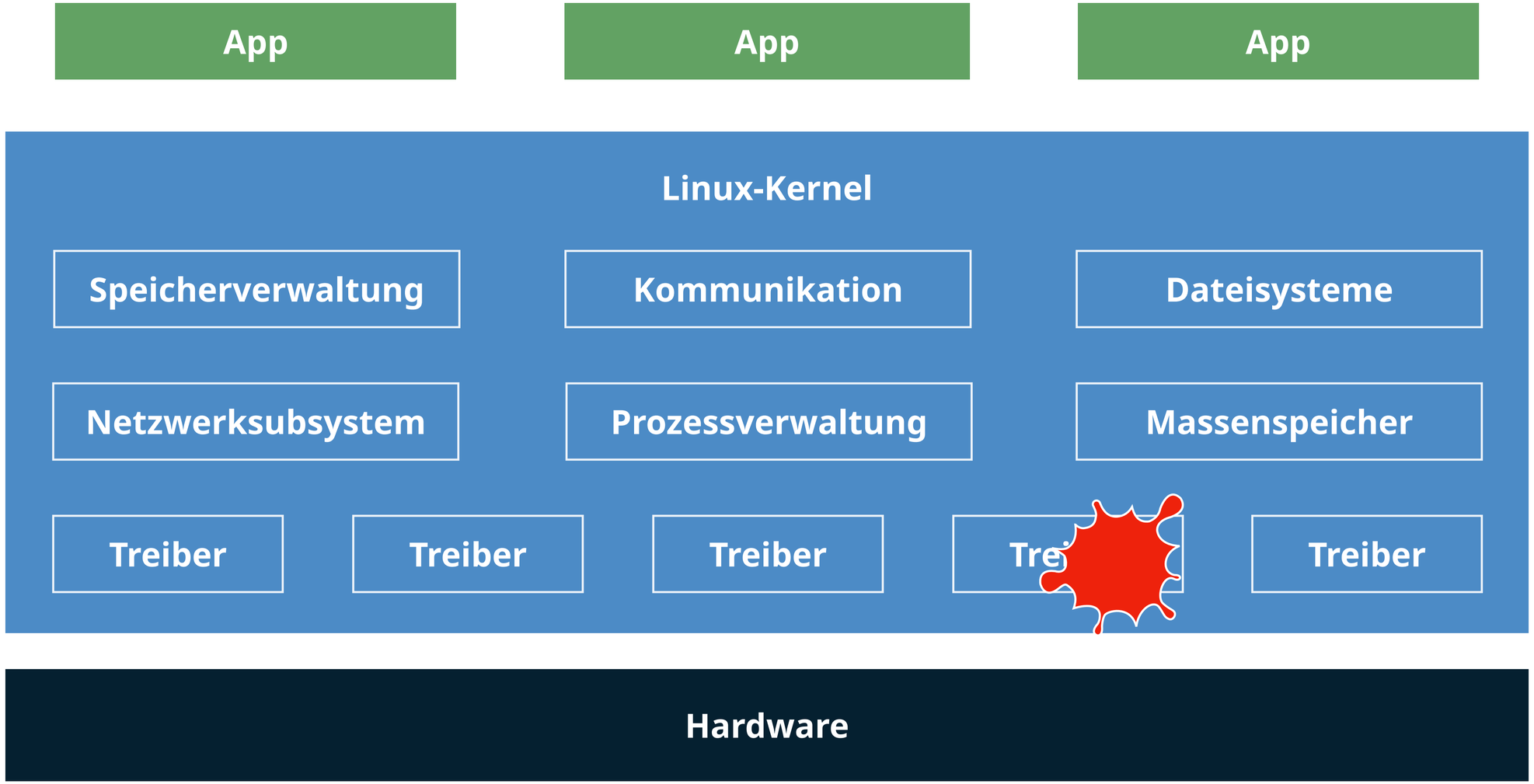
- Cyber-Physikalische Systeme (CPS) haben Einfluss auf Umwelt und Leben
- Internet der Dinge (IoT) verbindet CPS und exponiert sie gegenüber Angreifern
- Kommunikation muss abgesichert werden
- Beteiligten Rechnerknoten müssen sicher sein



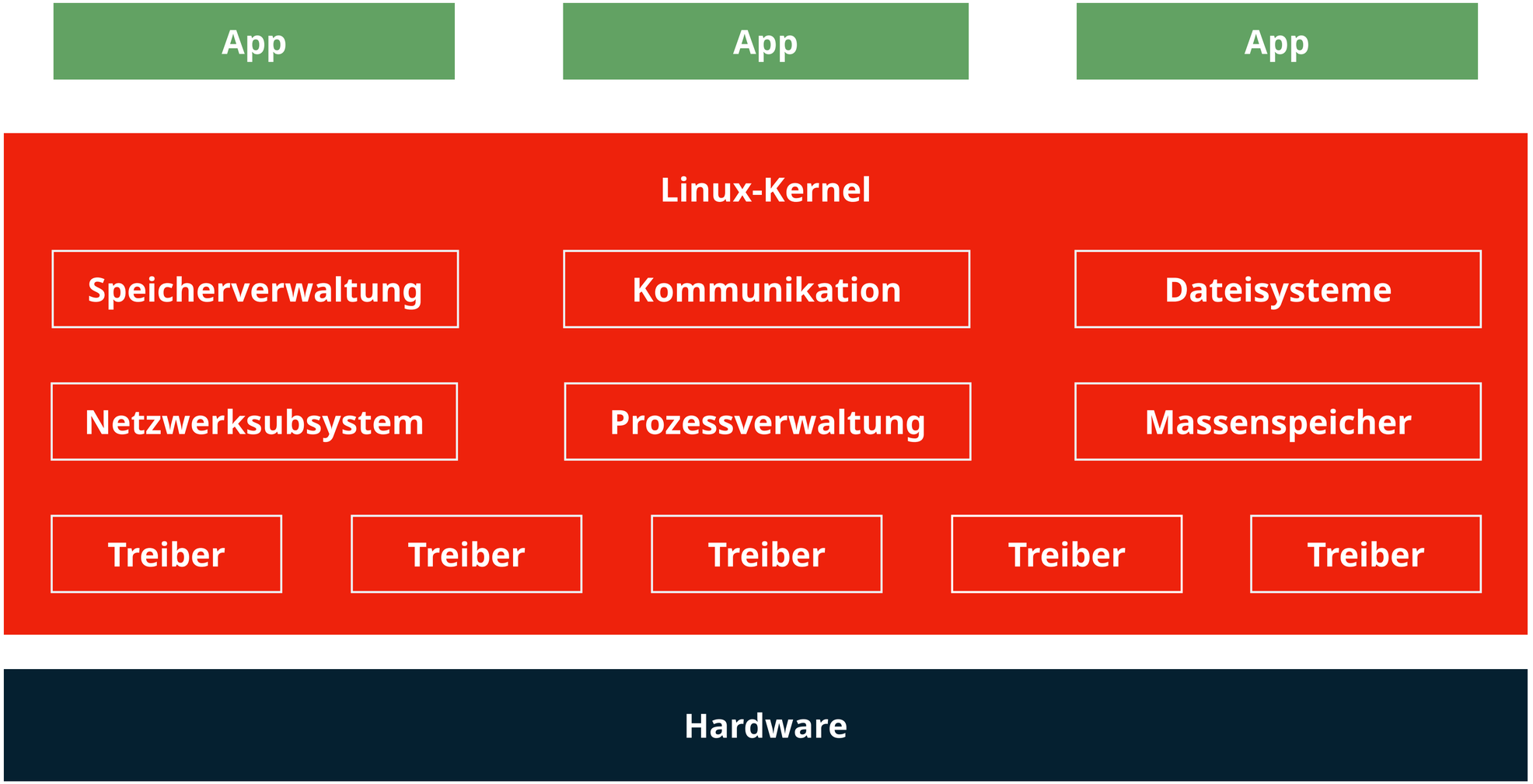
Internet der Dinge: Sicherheit in einem verteiltem System



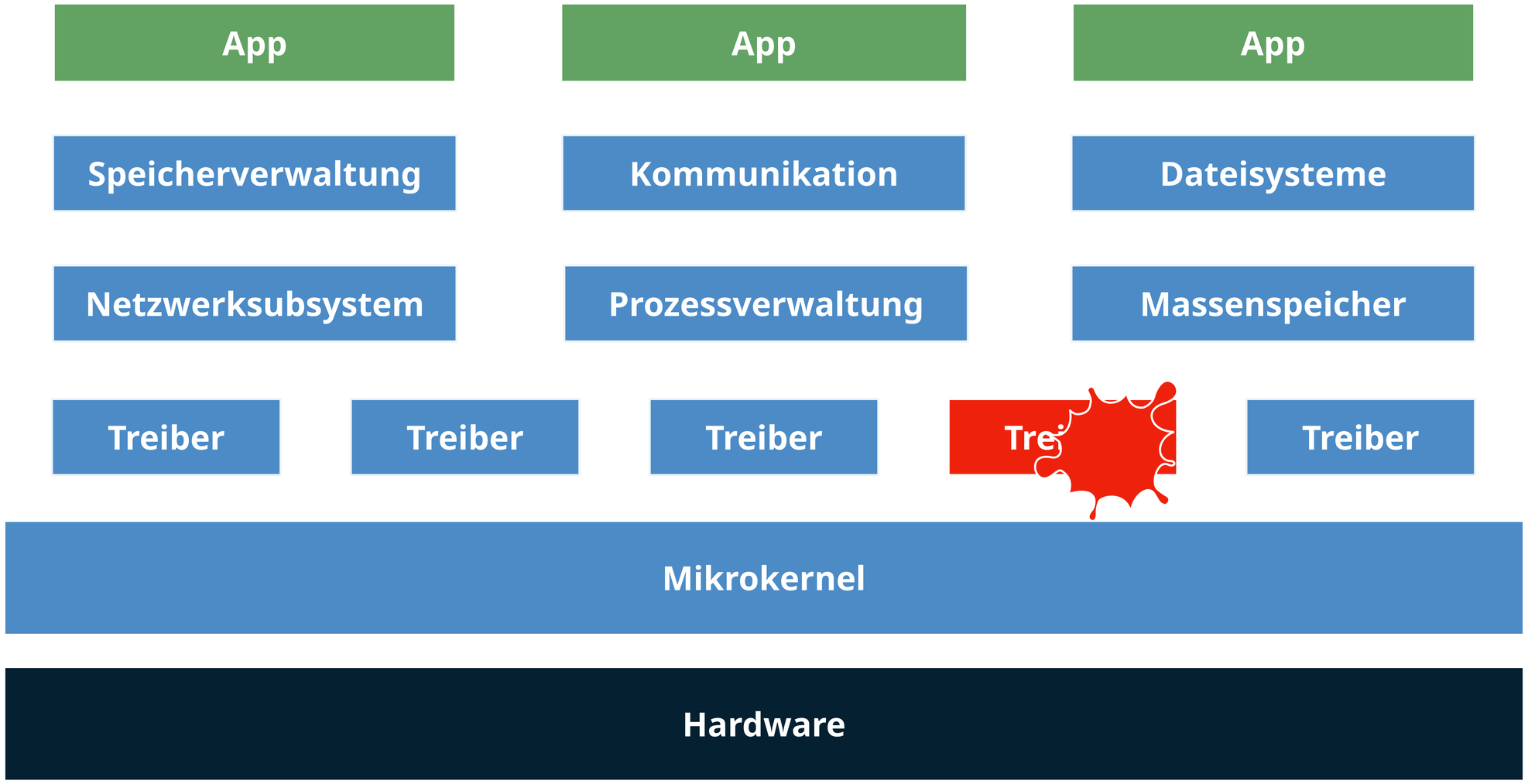
Software-Komplexität: Monolithische Systeme



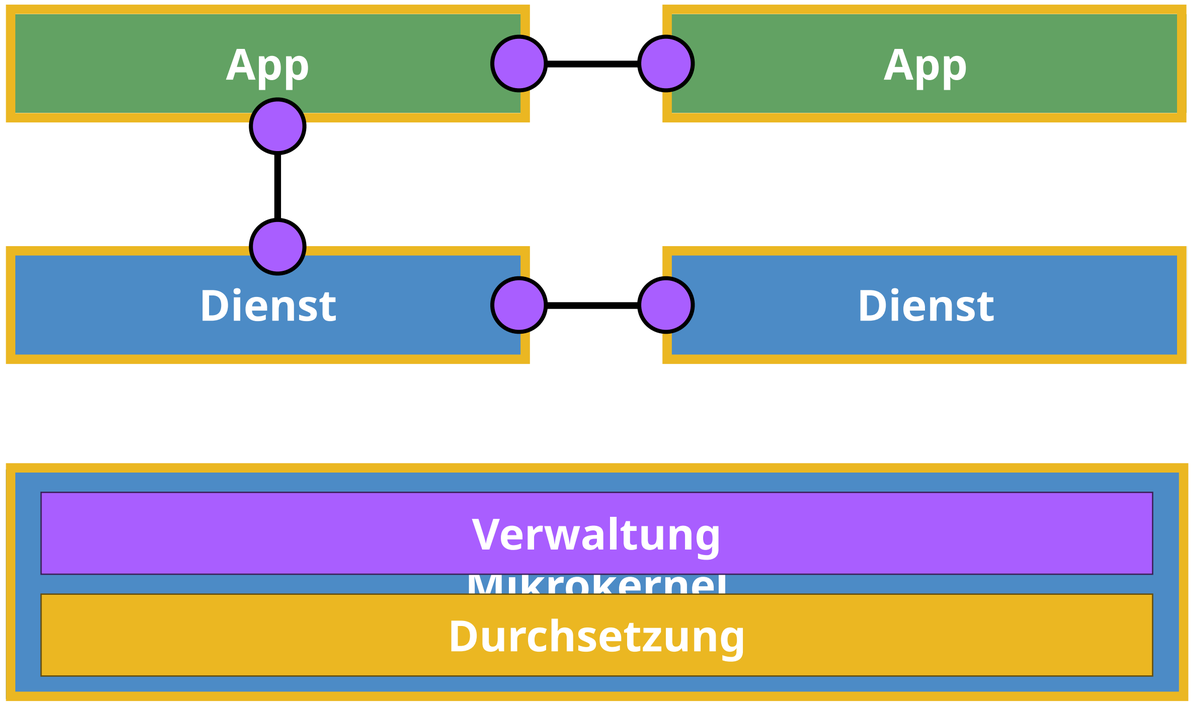
Software-Komplexität: Monolithische Systeme



Software-Komplexität: Mikrokernel-basierte Systeme



Software-Komplexität: Monolithische Systeme

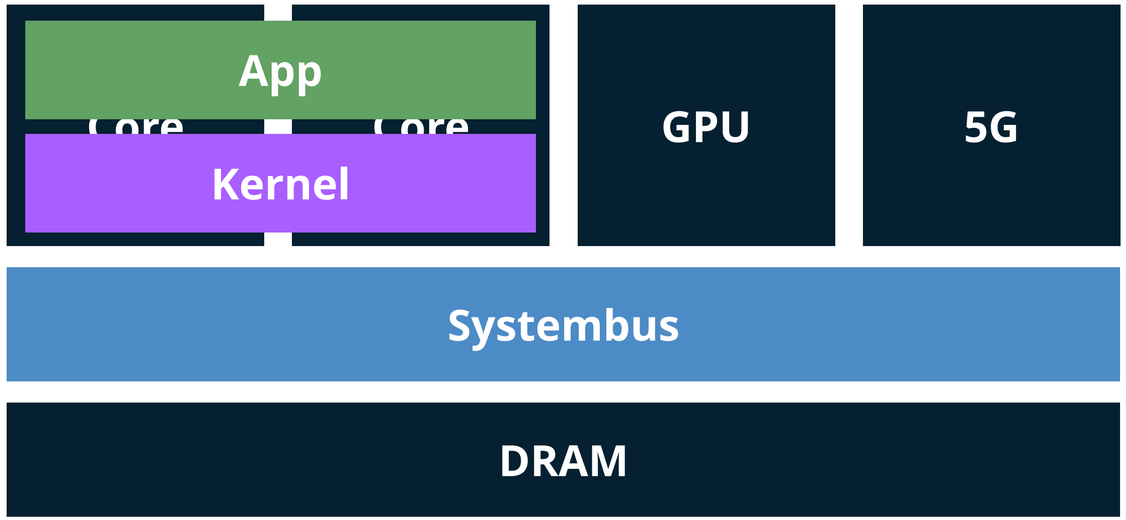


Verwaltung von Zugriffsrechten auf Basis von **Capabilities**

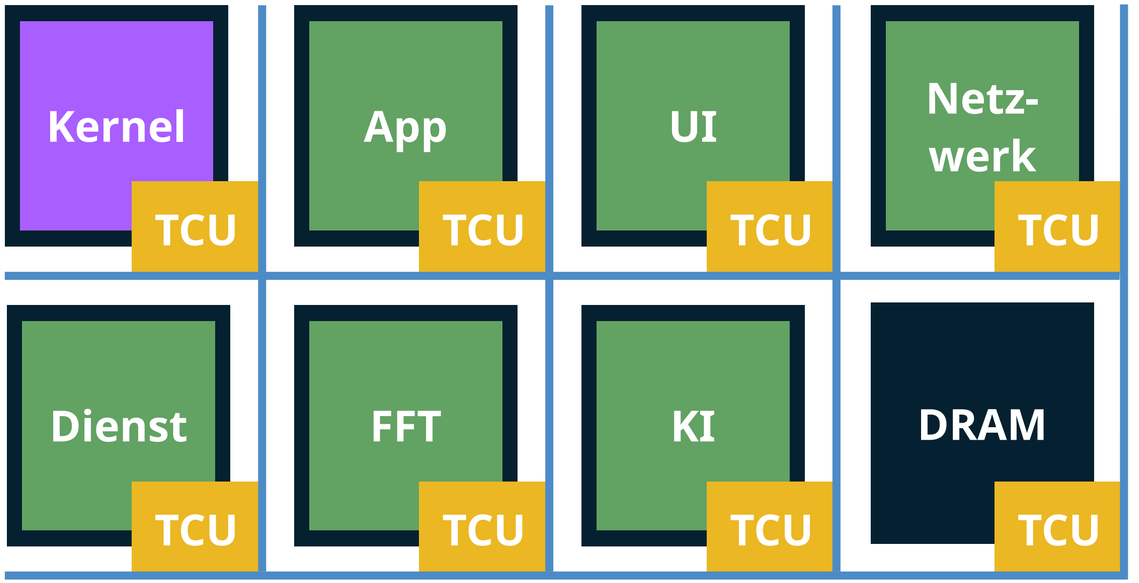
Heterogenität auch in eingebetteten Systemen im IoT

Wie sollen **Beschleuniger** behandelt werden?

Systemarchitektur

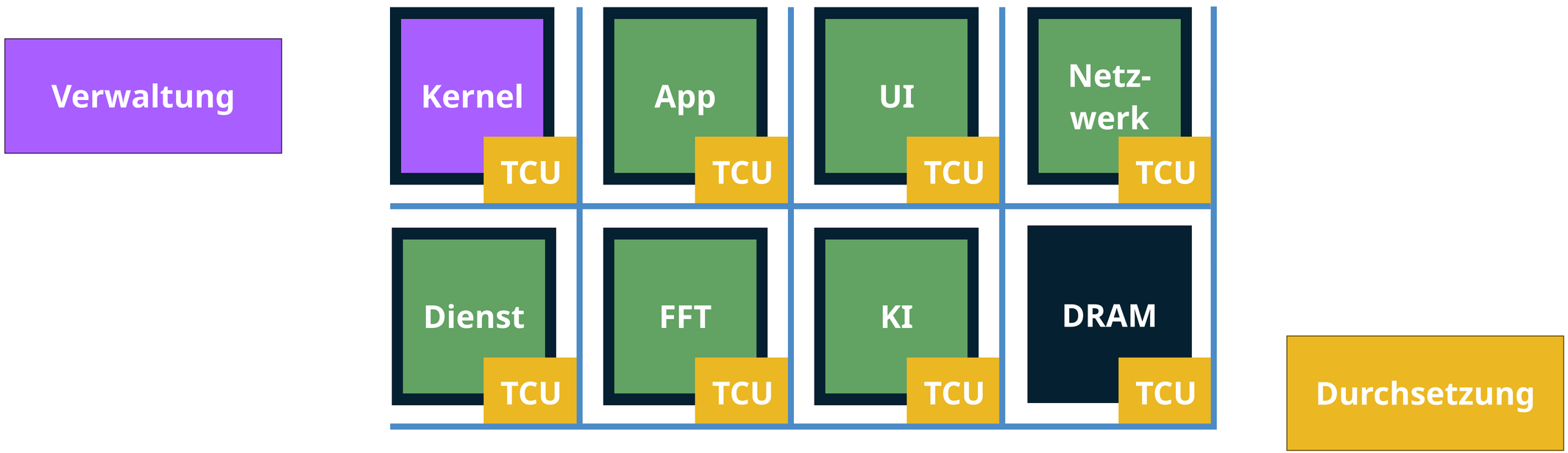


Traditionelle Hardware-Architektur

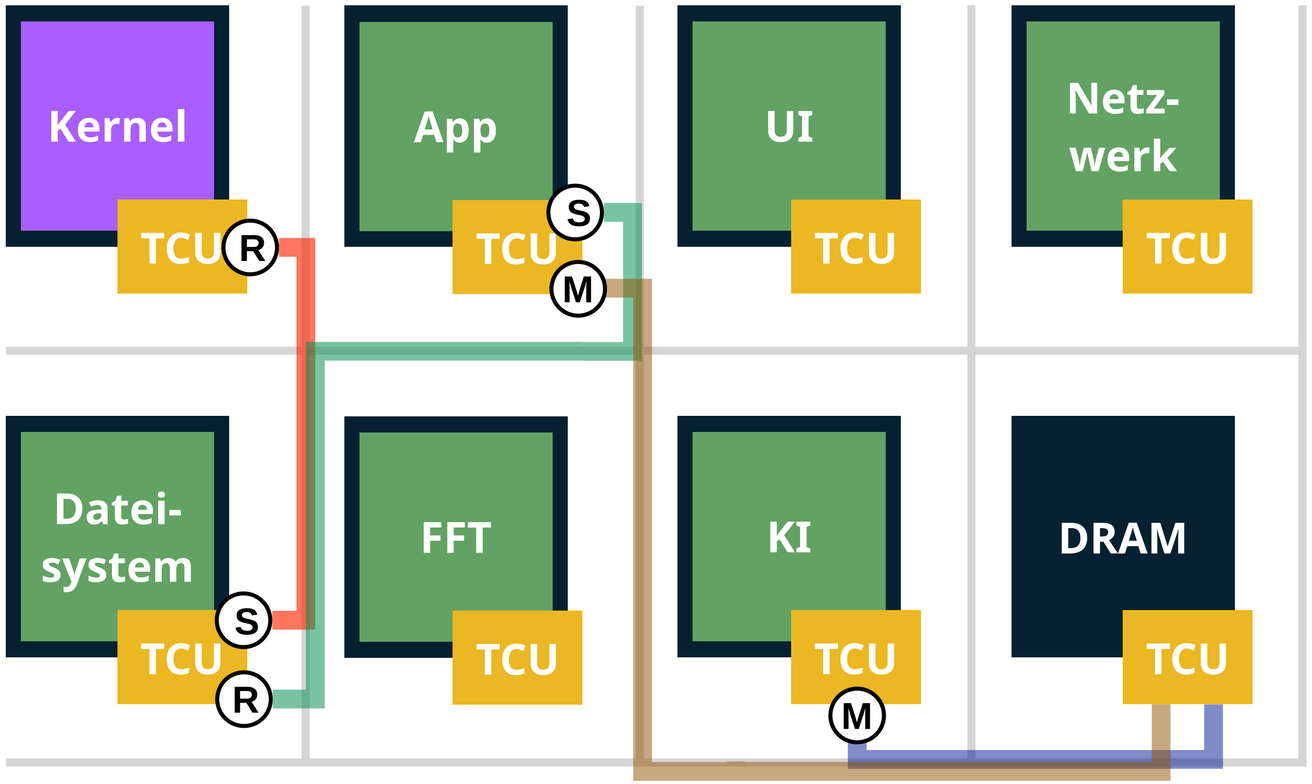


M³-Architektur

M³ Hardware/Software Co-Design



Zugriffskontrolle und Einbindung von Beschleunigern



- **TCU** hat konfigurierbare Endpunkte für:
  - Speicherzugriff
  - Senden / Empfangen von Nachrichten
- Konfiguration von außen:
  - durch **Kernel** auf privilegierter Kachel
  - auf Geheiß von **Anwendungen** (benötigen Capability)

Gleichbehandlung aller Kacheln

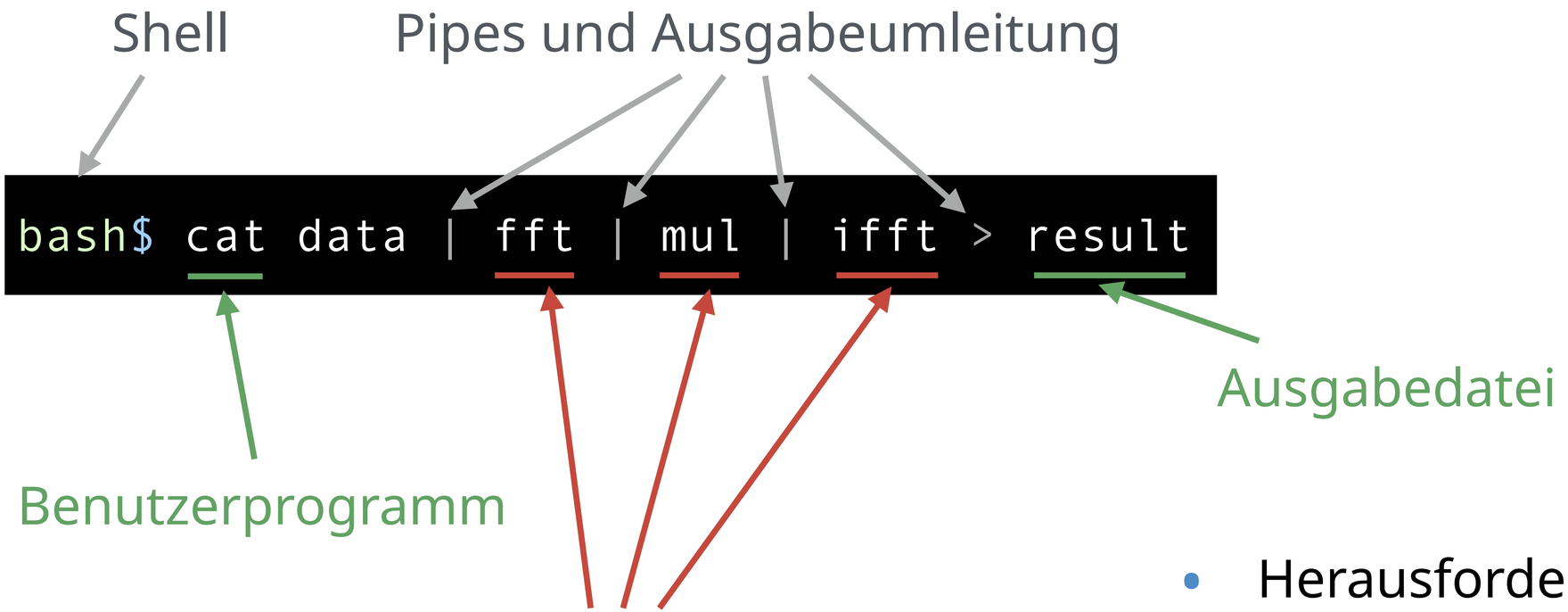
Verwaltung und **Durchsetzung** von
Zugriffsrechten **identisch für alle Kacheln**

Cores
(Software)

Beschleuniger
(Hardware)

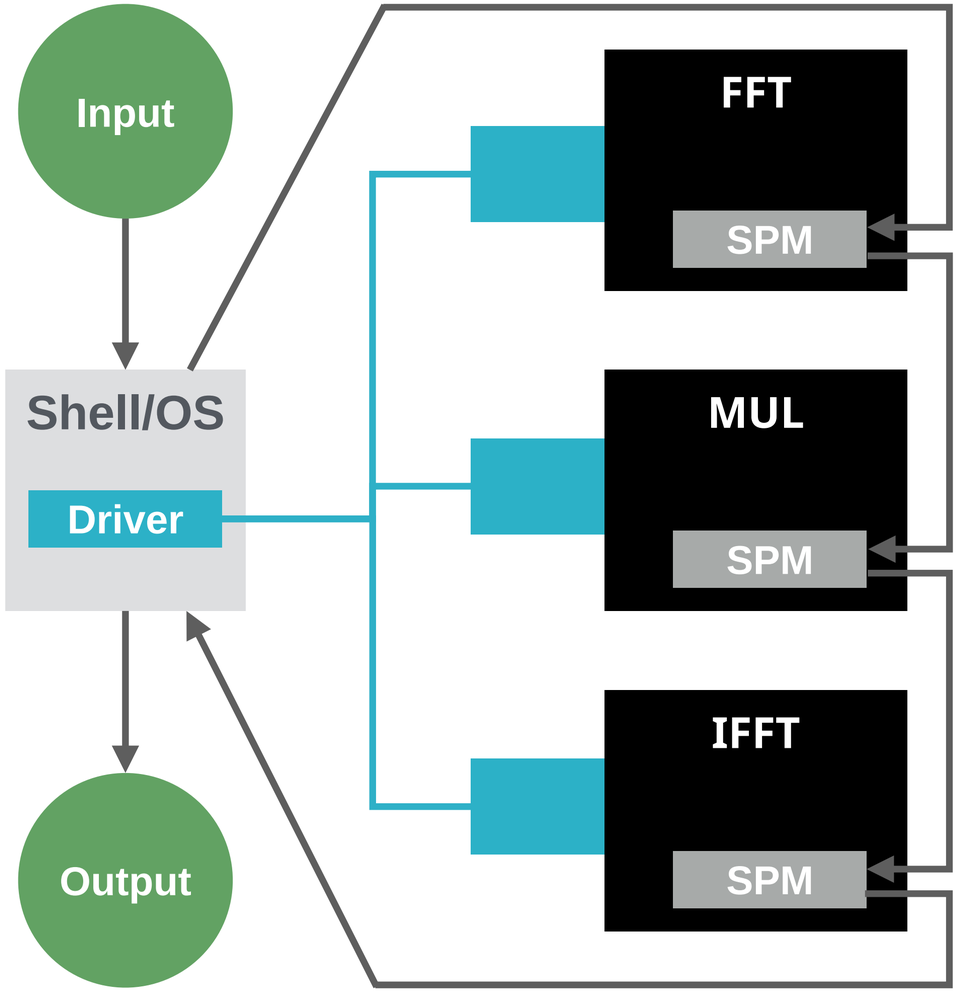
Autonome Beschleuniger

Bürger Erster Klasse: Kooperation zwischen allen Kacheln



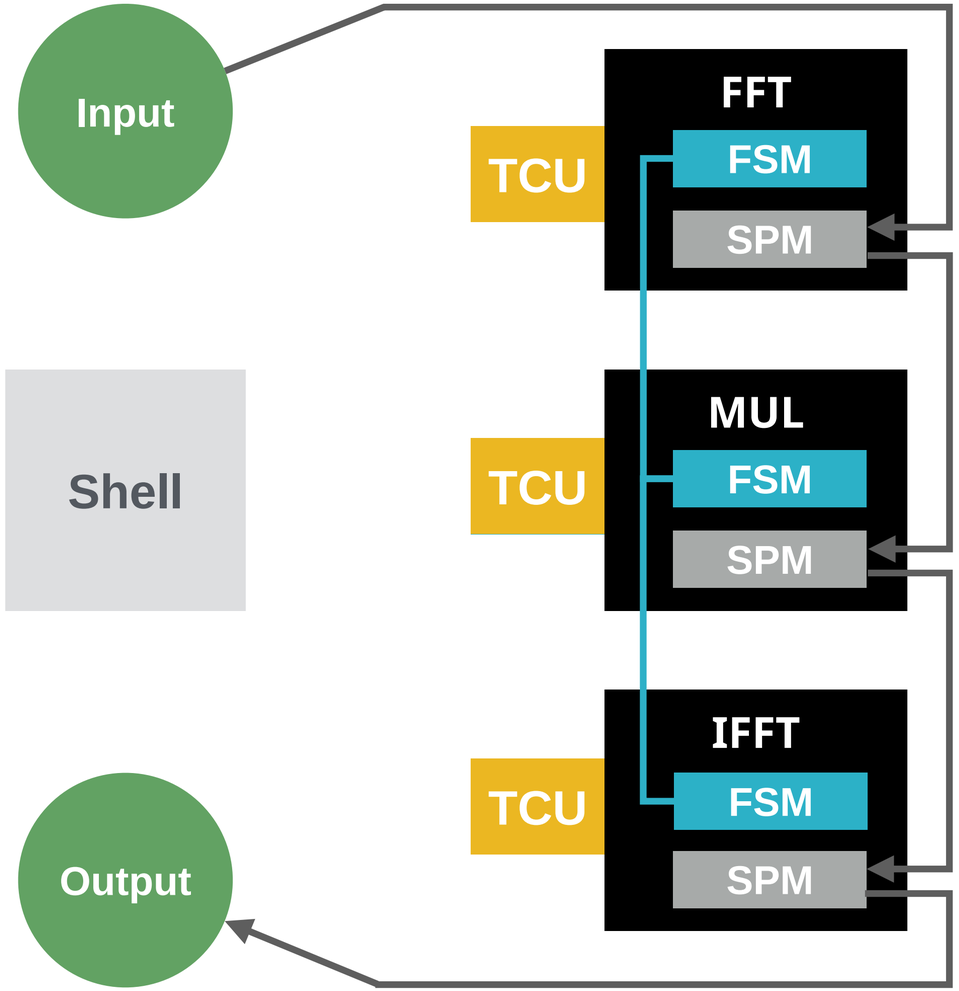
- Herausforderungen:
  - Generische **Protokolle**
  - **Protokollunterstützung** in Beschleunigern

Beschleunigerketten: Unterstützung durch Betriebssystem



- Unterstützt durch Betriebssystem:
  - Betriebssystem **treibt** Kopieren in/aus Scratchpad-Speicher des Beschleunigers
  - **Einfaches DMA** genügt
  - Wie in traditionellen Systemen, hohe CPU-Last für Betriebssystem

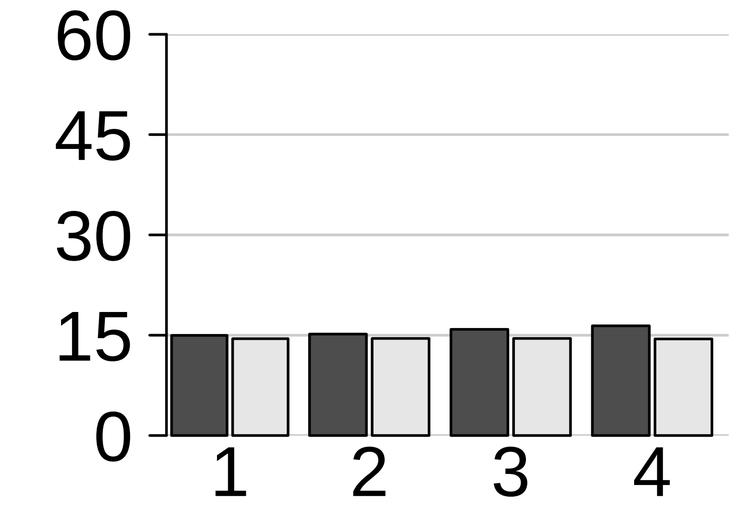
Beschleunigerketten: Komplette Autonomie



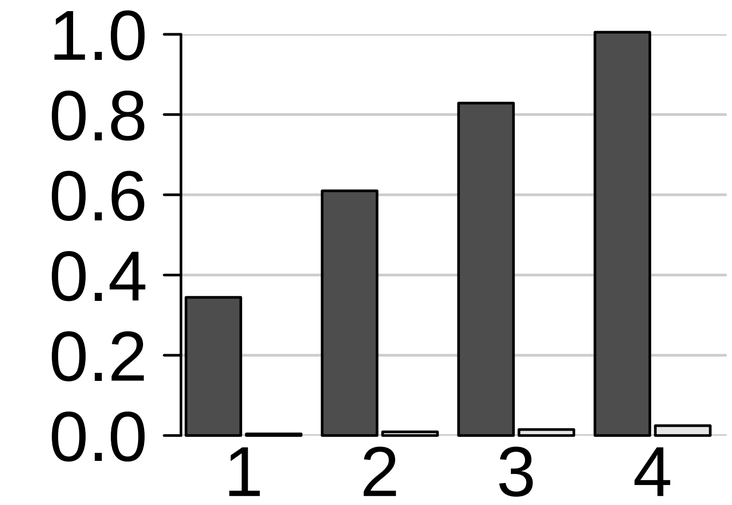
- Autonome Beschleuniger:
  - Shell konfiguriert **Endpunkte**
  - **Finite State Machines** der Beschleuniger treiben **TCUs** für autonomen Datentransfer
  - **TCU** setzt Zugriffsrechte durch
  - Offloading: keine CPU-Last für Betriebssystem

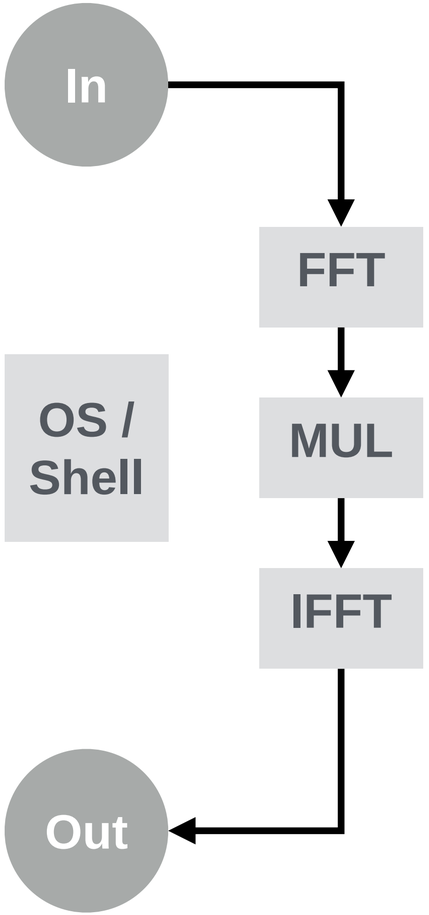
Beschleunigerketten: Leistungsbewertung

Laufzeit in ms



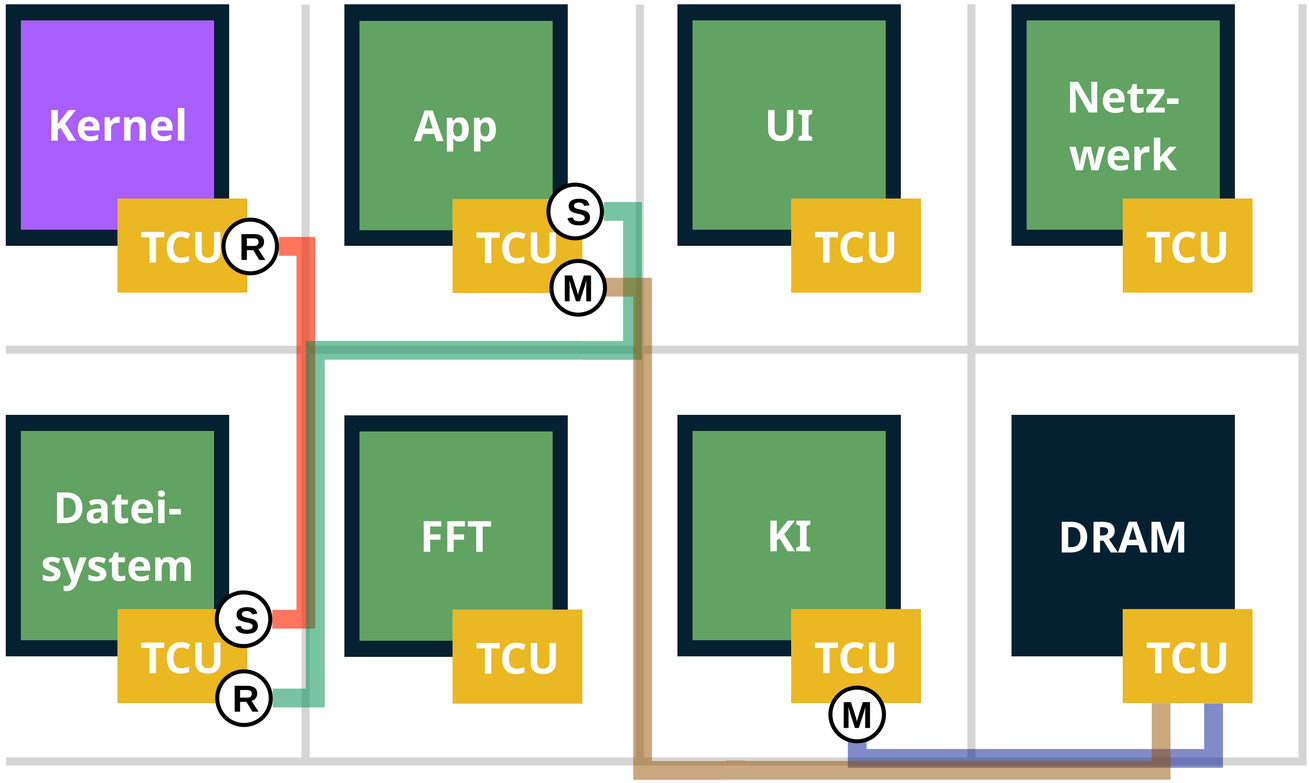
CPU-Last (Betriebssystem)





1..4 Beschleunigerketten arbeiten parallel

Echtzeitfähigkeit?



Herausforderungen:

- Garantiert **niedrige Latenz** bei Kommunikation zwischen Kacheln
- **Scheduling** von Network-on-Chip und Kacheln
- Inseln von **Hardware-assistiertem** Scheduling ohne Kernel
- **Leicht einsetzbares** Framework für Entwickler

Caladan: Datacenter-Architektur

Capability-basierte Datacenter-Architektur

- Kooperation **Prof. Mark Silbersteins** Gruppe am **Technion (Haifa)**
- Komponentenarchitektur für **Disaggregated Datacenters**
- **Remote Direct Memory Access** (RDMA) notwendig für hohe Leistung
- … aber schlecht für Sicherheit

Capability-basierte Datacenter-Architektur

- Kooperation **Prof. Mark Silbersteins** Gruppe am **Technion (Haifa)**
- Komponentenarchitektur für **Disaggregated Datacenters**
- **Remote Direct Memory Access** (RDMA) notwendig für hohe Leistung
- … aber schlecht für Sicherheit
- Prinzipien der M³-Architektur angewendet: **SmartNICs** als **„TCUs“**

Aktuelle Forschungsthemen

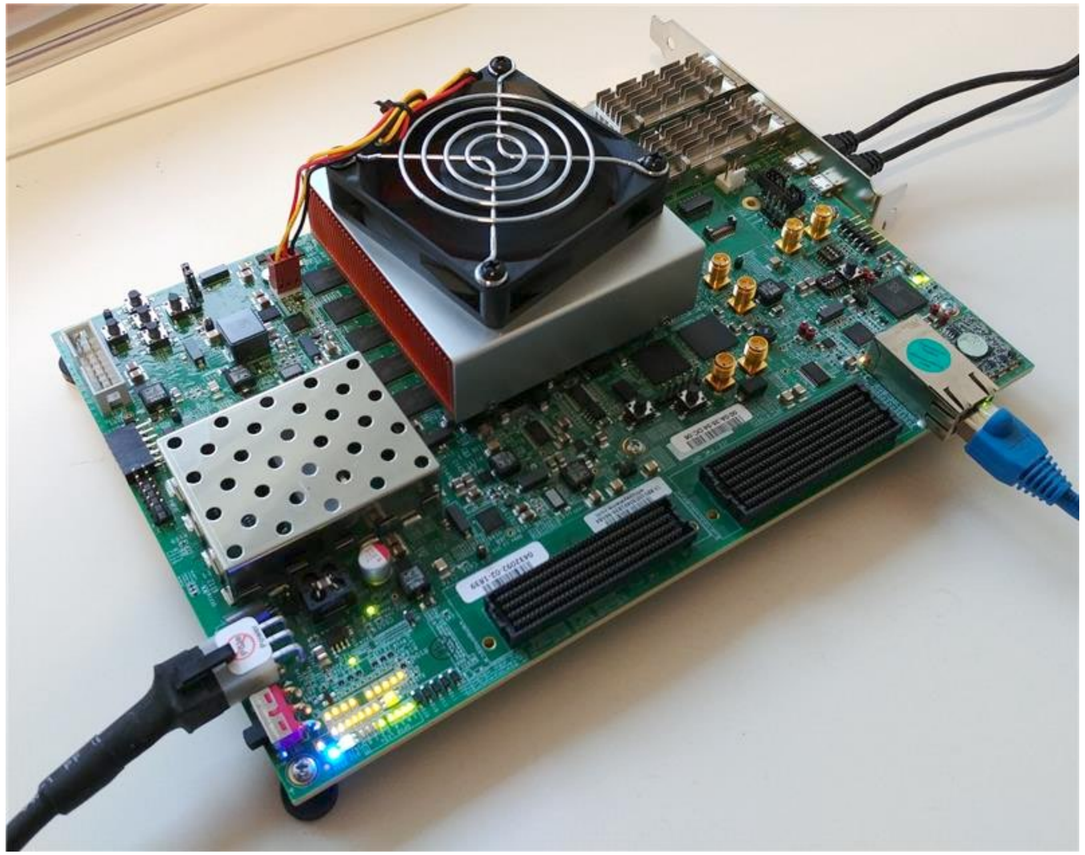
Simulation mit gem5 und FPGA-Prototyp

```
      L1i$ =32 KiB (2-way assoc, 4 cycles)
      L1d$ =32 KiB (2-way assoc, 4 cycles)
      L2$  =256 KiB (8-way assoc, 12 cycles)
      Comp =Core -> DTU+AT -> L1$ -> L2$

PE03: build/gem5-x86_64-release/bin/rctmux
      Core =TimingSimpleCPU x86_64 @ 1GHz
      DTU  =eps:16, bufsz:1024 B, blocksz:64 B, count:4, tlb:128, walker:1
      L1i$ =32 KiB (2-way assoc, 4 cycles)
      L1d$ =32 KiB (2-way assoc, 4 cycles)
      L2$  =256 KiB (8-way assoc, 12 cycles)
      Comp =Core -> DTU+AT -> L1$ -> L2$

PE04: build/gem5-x86_64-release/default.img x 1
      DTU  =eps:16, bufsz:1024 B, blocksz:1024 B, count:8, tlb:0, walker:0
      imem =3145728 KiB
      Comp =DTU -> DRAM

Global frequency set at 1000000000000 ticks per second
info: kernel located at: build/gem5-x86_64-release/bin/kernel
info: kernel located at: build/gem5-x86_64-release/bin/rctmux
info: kernel located at: build/gem5-x86_64-release/bin/rctmux
info: kernel located at: build/gem5-x86_64-release/bin/rctmux
warn: DRAM device capacity (49152 Mbytes) does not match the address range assigned (4096 Mbytes)
info: No kernel set for full system simulation. Assuming you know what you're doing
info: No kernel set for full system simulation. Assuming you know what you're doing
platform.com_1.device: Listening for connections on port 3456
0: pe00.remote_gdb: listening for remote gdb on port 7000
0: pe01.remote_gdb: listening for remote gdb on port 7001
0: pe02.remote_gdb: listening for remote gdb on port 7002
0: pe03.remote_gdb: listening for remote gdb on port 7003
warn: CoherentXBar pe04.xbar has no snooping ports attached!
info: Loaded 'root' to 0x8400000080000000 .. 0x8400000080043868
info: Loaded 'hello' to 0x8400000080044000 .. 0x840000008016eb60
info: Loaded 'hello' to 0x840000008016f000 .. 0x8400000080299b60
info: Loaded 'rctmux' to 0x840000008029a000 .. 0x84000000802aed08
info: Entering event queue @ 0.  Starting simulation...
[kernel  @0] Kernel is ready
Hello World
Hello World
[kernel  @0] Shutting down
Exiting @ tick 6355915000 because m5 exit instruction encountered
```



Aktuelle Forschungsthemen

- **M³-basiertes Hardware/Software Co-Design:**
  - **Hardware:** FPGA-Prototyp mit RISC-V Cores, TCUs, Network-on-Chip
  - **Software:** Virtueller Speicher, Kontextwechsel beliebiger Recheneinheiten
- **Anwendung auf Rechenzentren:** Caladan-Architektur
- **Vertrauen zwischen verteilten Rechnerknoten:**
  - Trusted Execution Environment
  - Remote Attestation
- **Ohua-Compiler:** Zerlegung von Programmen in isolierte Komponenten
- **Echtzeit:** Scheduling von Cores, Beschleunigern, Network-on-Chip