Strukturierung von Betriebssystemen

Betriebssysteme
WS 2006/2007

Jörg Kaiser
IVS – EOS

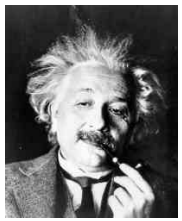
Otto-von-Guericke-Universität Magdeburg

Allgemeines Entwurfsprinzip: "Weniger ist mehr"

Occam's Razor: "Plurality should not be assumed without necessity."
(William of Ockham, ca. 1285-1349)

C.A.R. Hoare: "There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult."

A. Einstein: "Mache die Dinge so einfach wie möglich - aber nicht einfacher"



A. de Saint Exupérie: ".. la perfection n'est pas atteinte quand il n'y a plus rien à ajouter mais quand il n'y a plus rien à enlever!"

Allgemeine Entwurfs- und Strukturierungsprinzipien und -techniken

Abstraktion: Reduktion der Komplexität

Trennung von Strategie und Mechanismus: Flexibilisierung

Orthogonalität: Unabhängigkeit und Komponierbarkeit

} Prinzipien

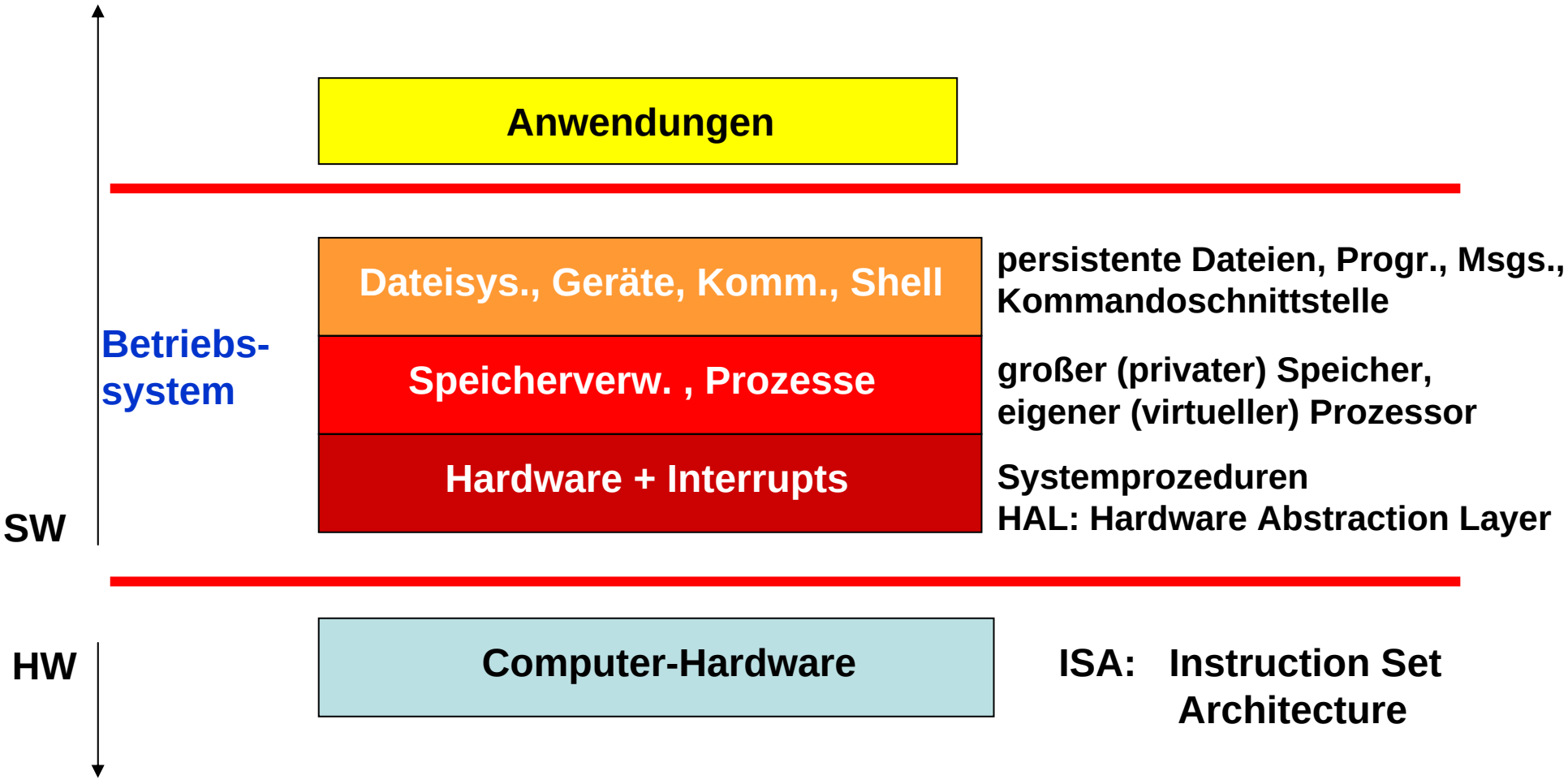
Modularisierung: Kapselung und Schnittstelle

Hierarchisierung: Beziehungen zw. Komponenten

Schichtung: gängige Form der Hierarchisierung

} Techniken

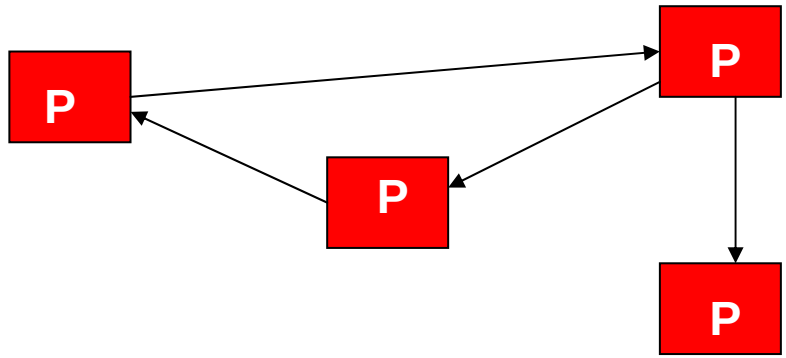
Schichtenmodell



Abstraktion: Prozess

Prozesse sind die Komponenten zur Repräsentation und Strukturierung der Aktivität im System !

Prozessbereich: Funktion wird hier erbracht



: Interaktion

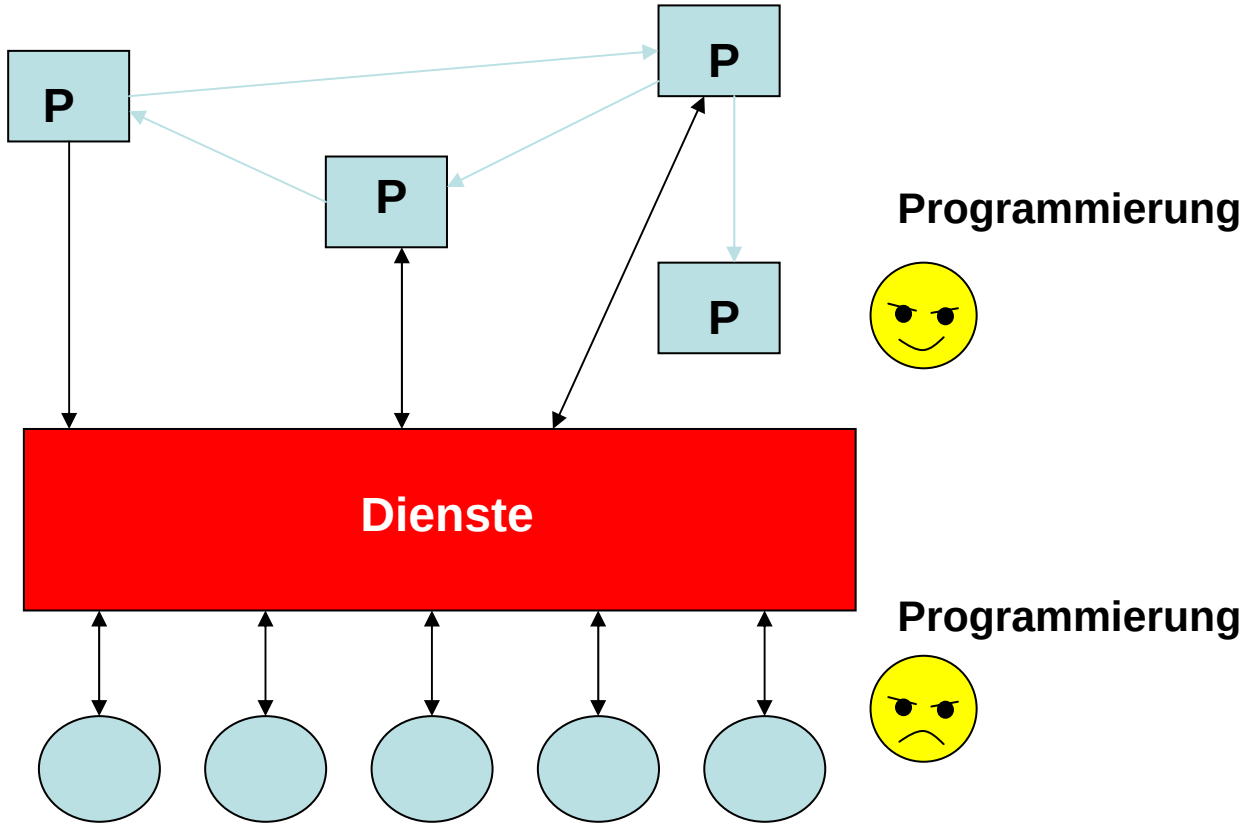
Abstraktion der Betriebsmittel

Kernbereich — **Infrastruktur zur Realisierung von Prozessen**

Abstraktion: Dienst

Dienste dienen dem Betrieb und der Verwaltung von Ressourcen (Betriebsmittel). Sie stellen eine Benutzergerechte Schnittstelle zu Betriebsmittel zur Verfügung.

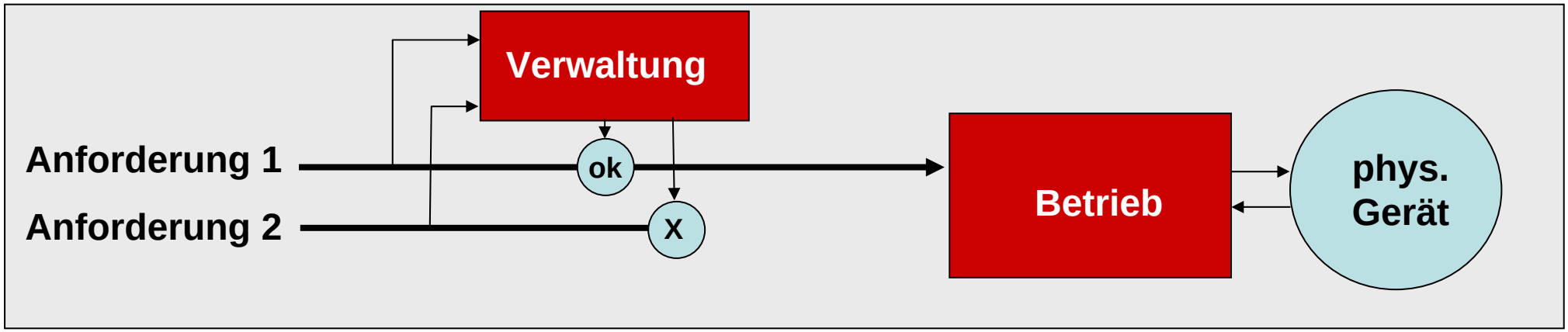
Logische Betriebsmittel:
Datei, Fenster, Mouse-Zeiger,..



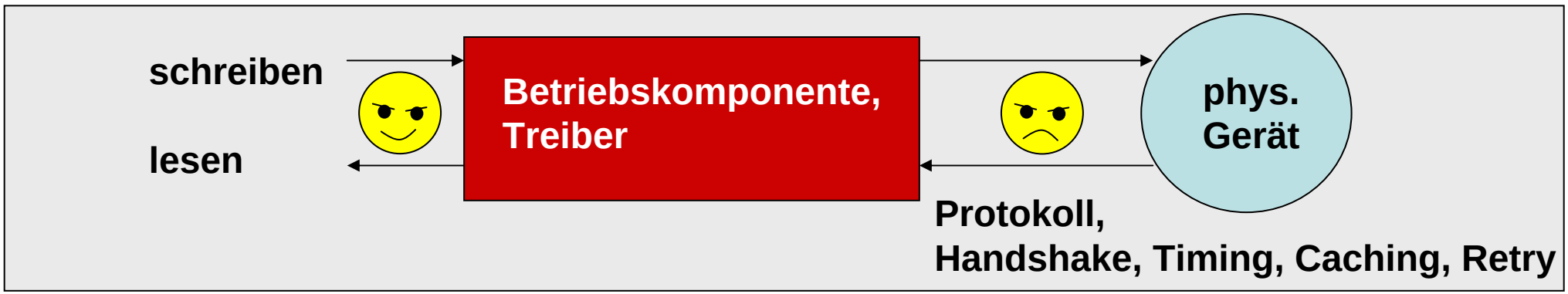
Reale Betriebsmittel:
Platte, Video RAM, Serielles E/A Gerät, ...

Dienste: Betrieb und Verwaltung

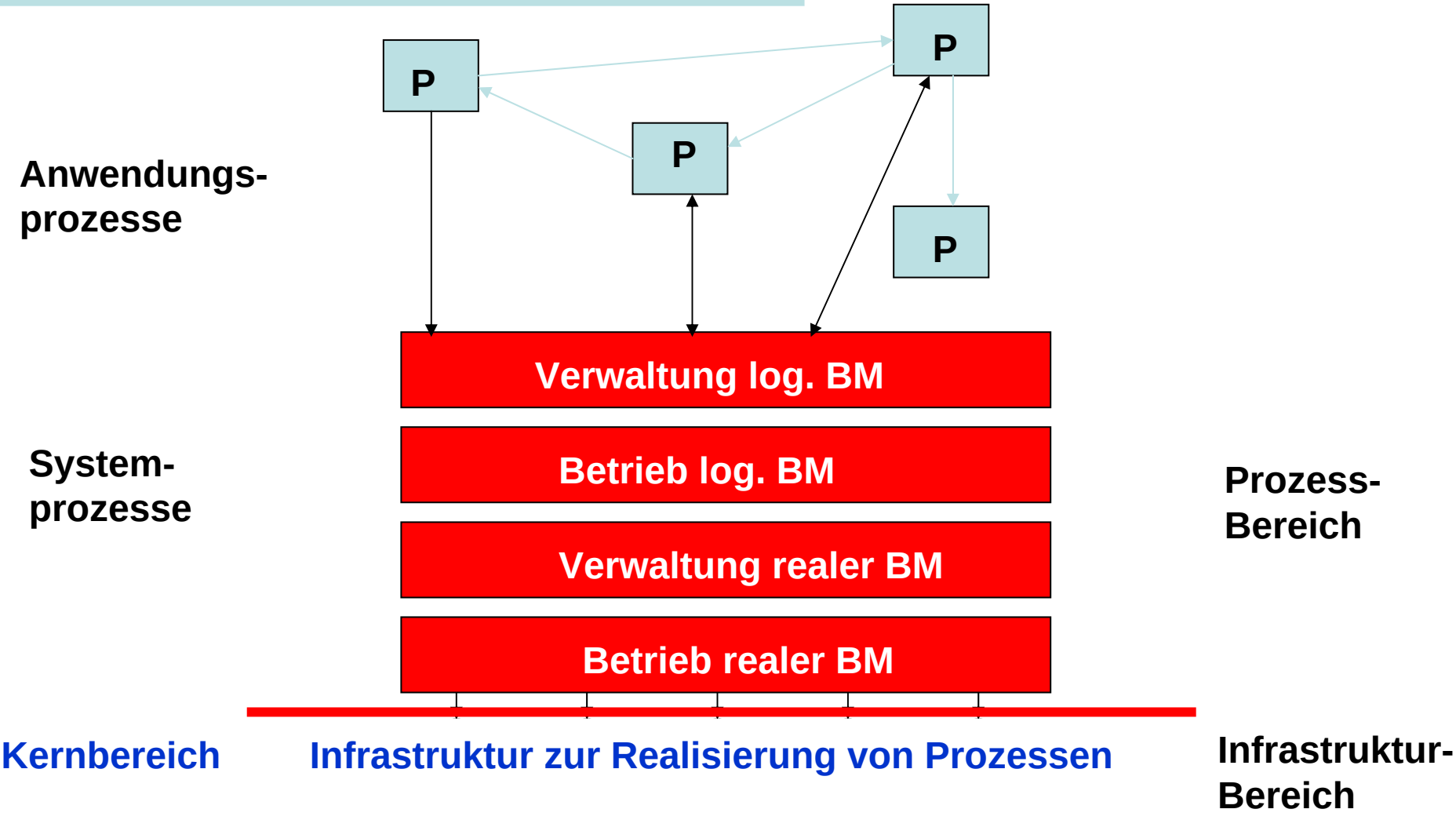
BM-Verwaltung: Zuteilung von Nutzungsanforderungen.



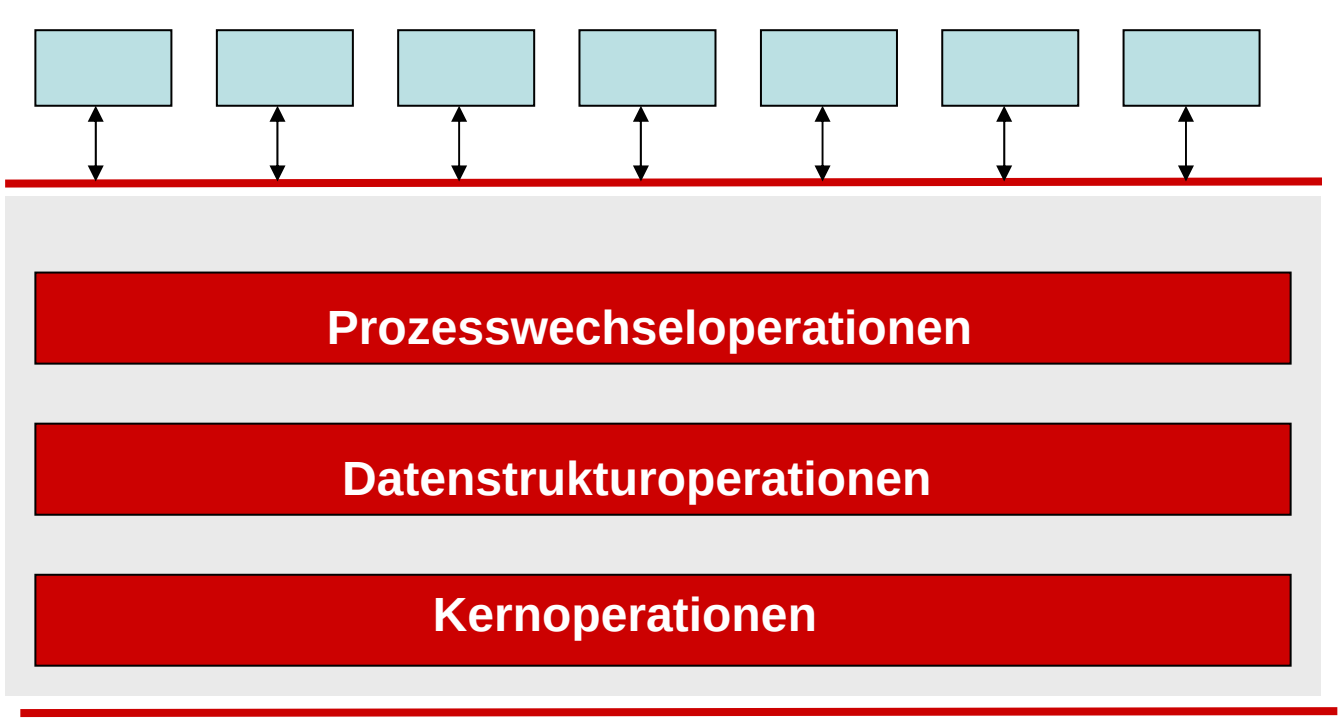
BM-Betrieb: Steuerung und Kontrolle von Betriebsmitteln



Schichtung der Dienste



Strukturierung des Kernbereichs



Prozess-Bereich

Kern-Bereich

Hardware Abstraktionen (HAL)

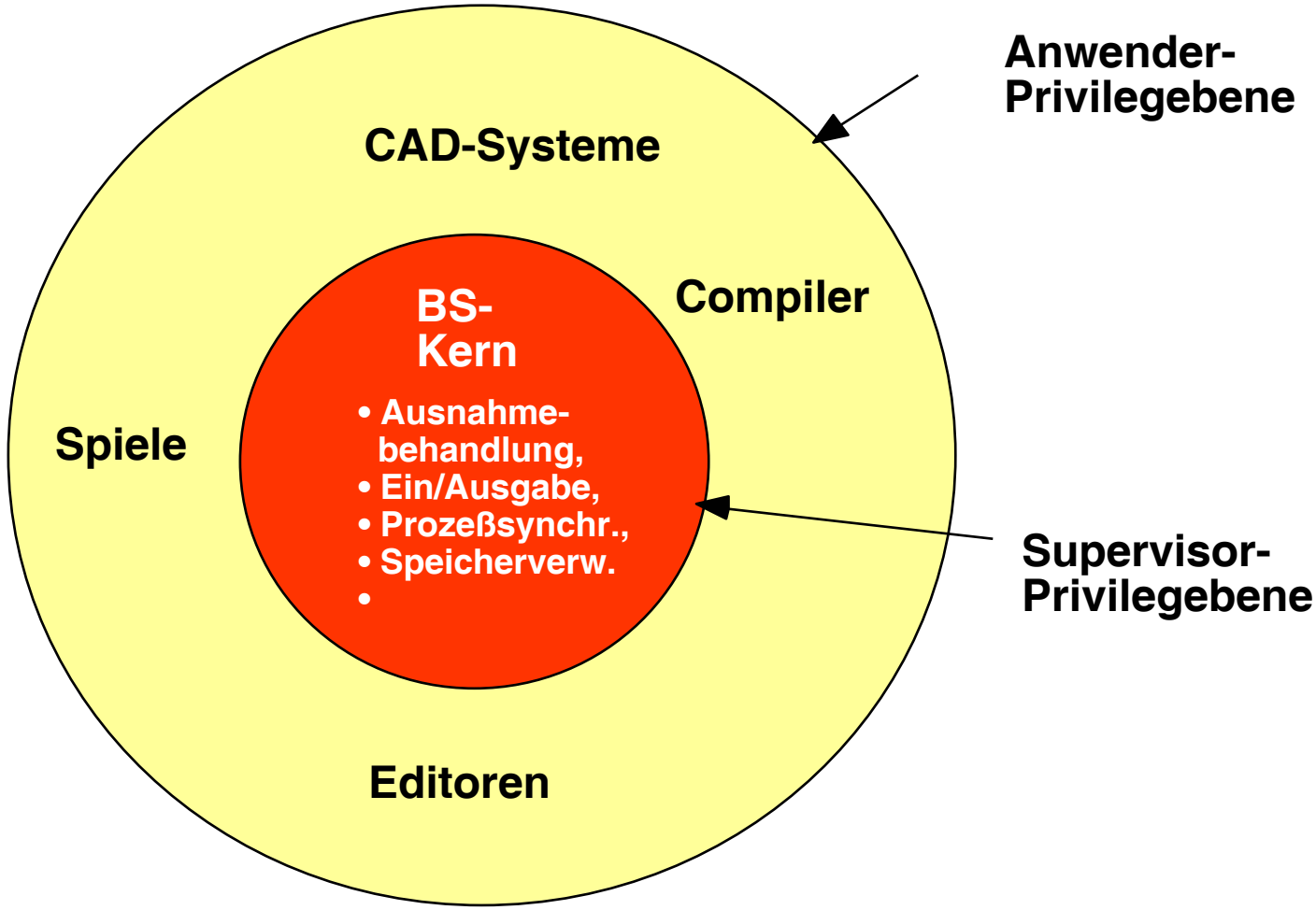
z.B. DMA, Interrupt Handler, Systemuhr, Speichermodule

Firmware-Bereich

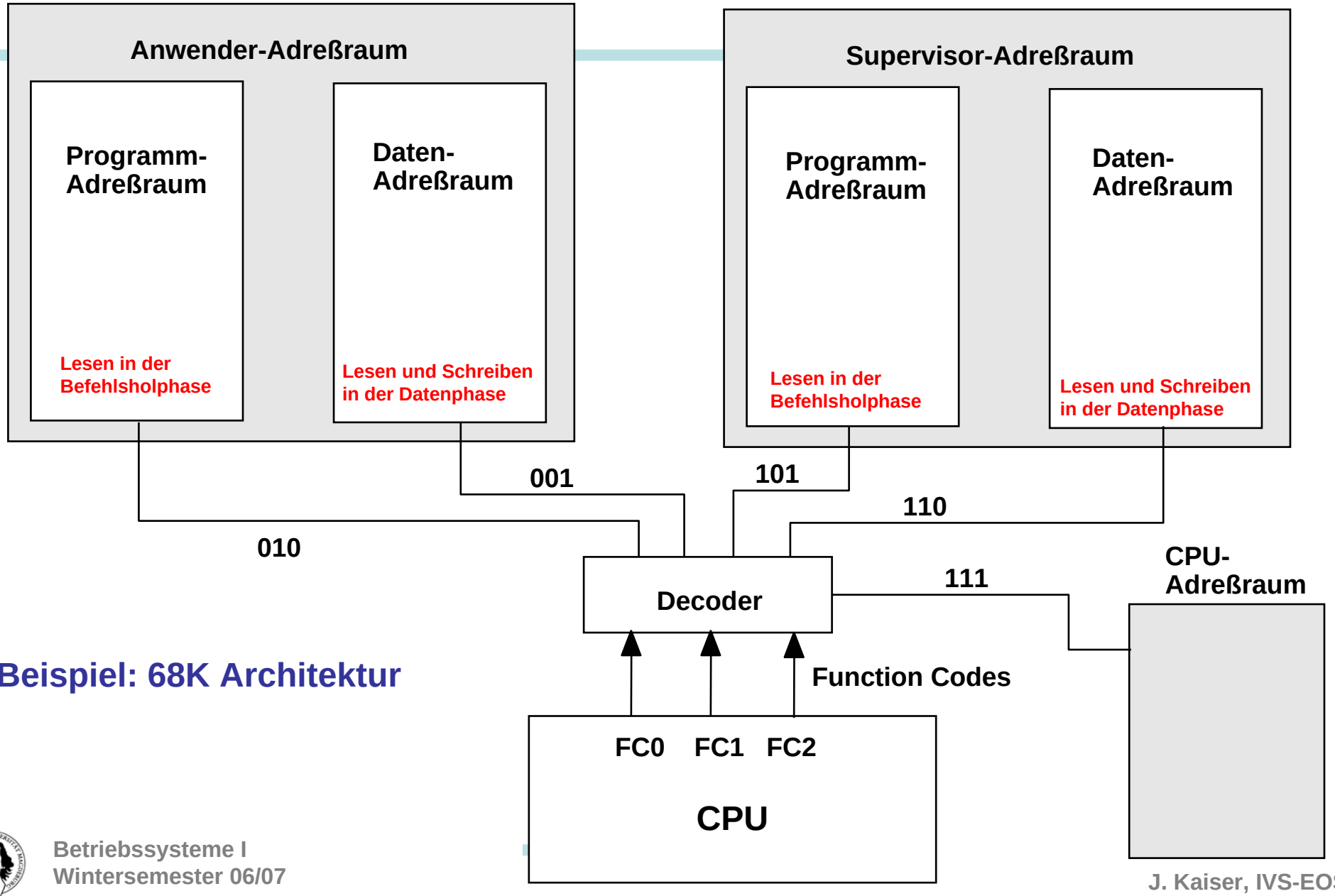
Was ist Anwendung ?
Was ist System ?
Was ist Kern ?

Trennung von Anwendermodus und Systemmodus

Motivation:
Zugriffsschutz!



Trennung der Adreßräume durch die CPU-Hardware



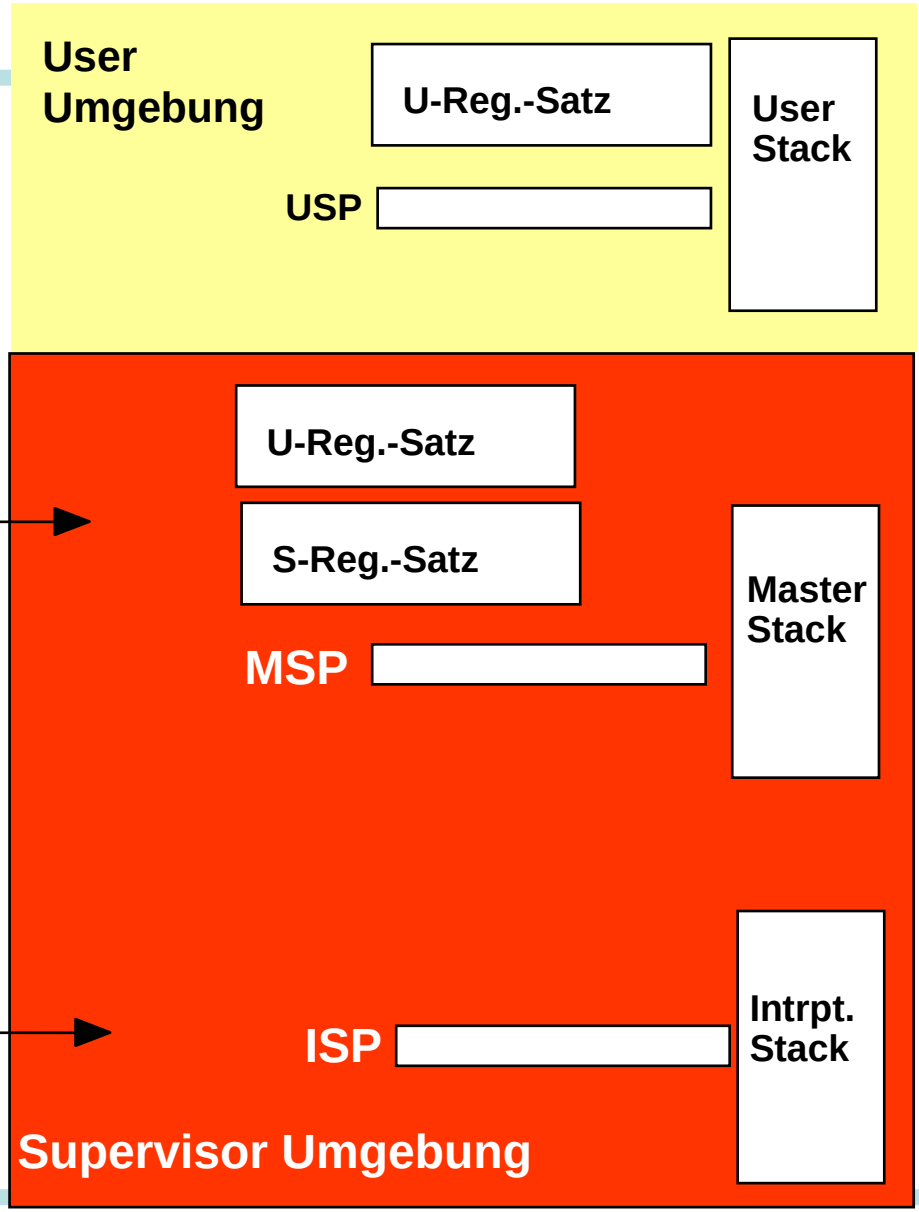
Beispiel: 68K Architektur

Übergang vom Anwender in den Supervisor Zustand

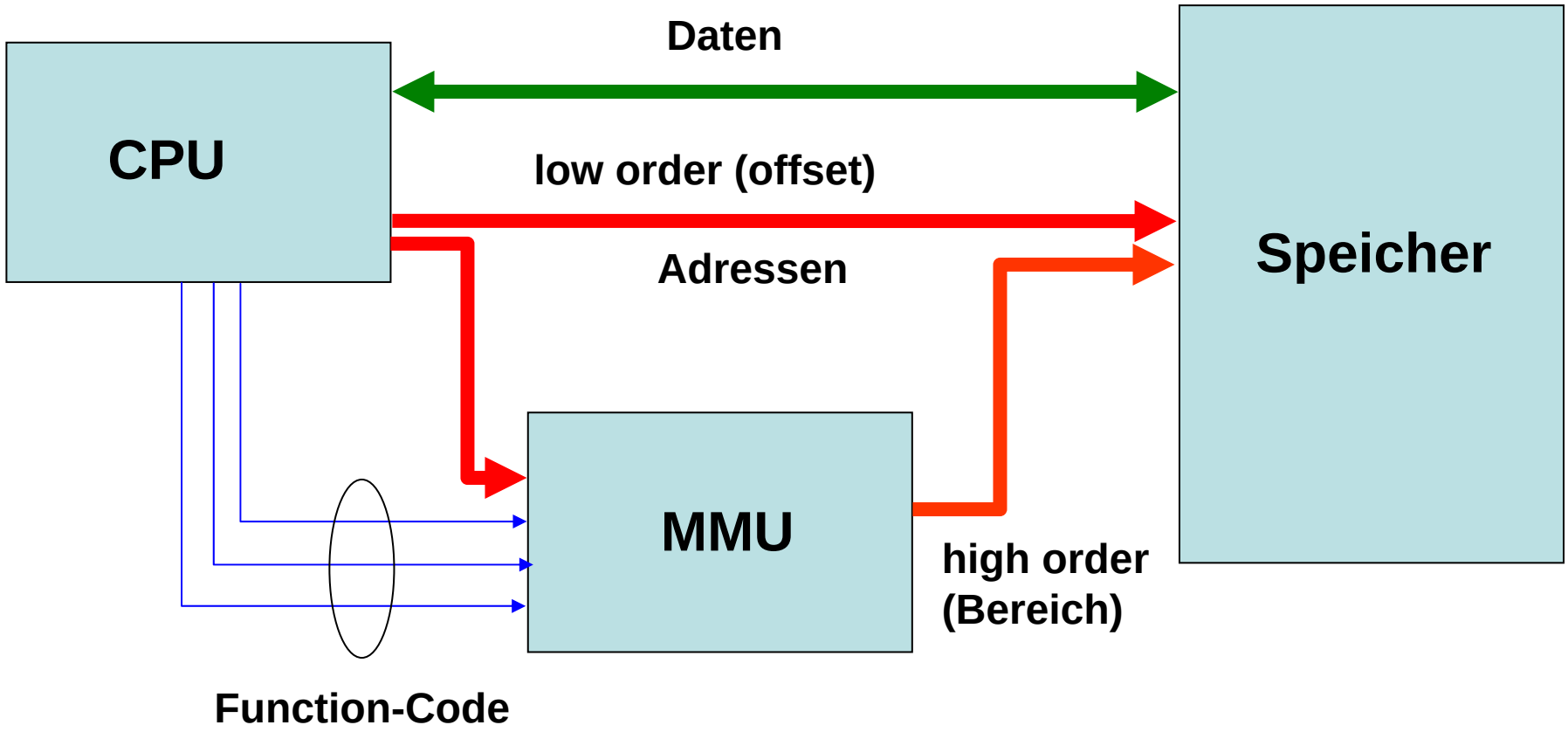
Der Übergang vom Anwendermodus in den Supervisormodus ist nur durch Interrupts oder Ausnahmebehandlung möglich.

Ausnahmen sind:

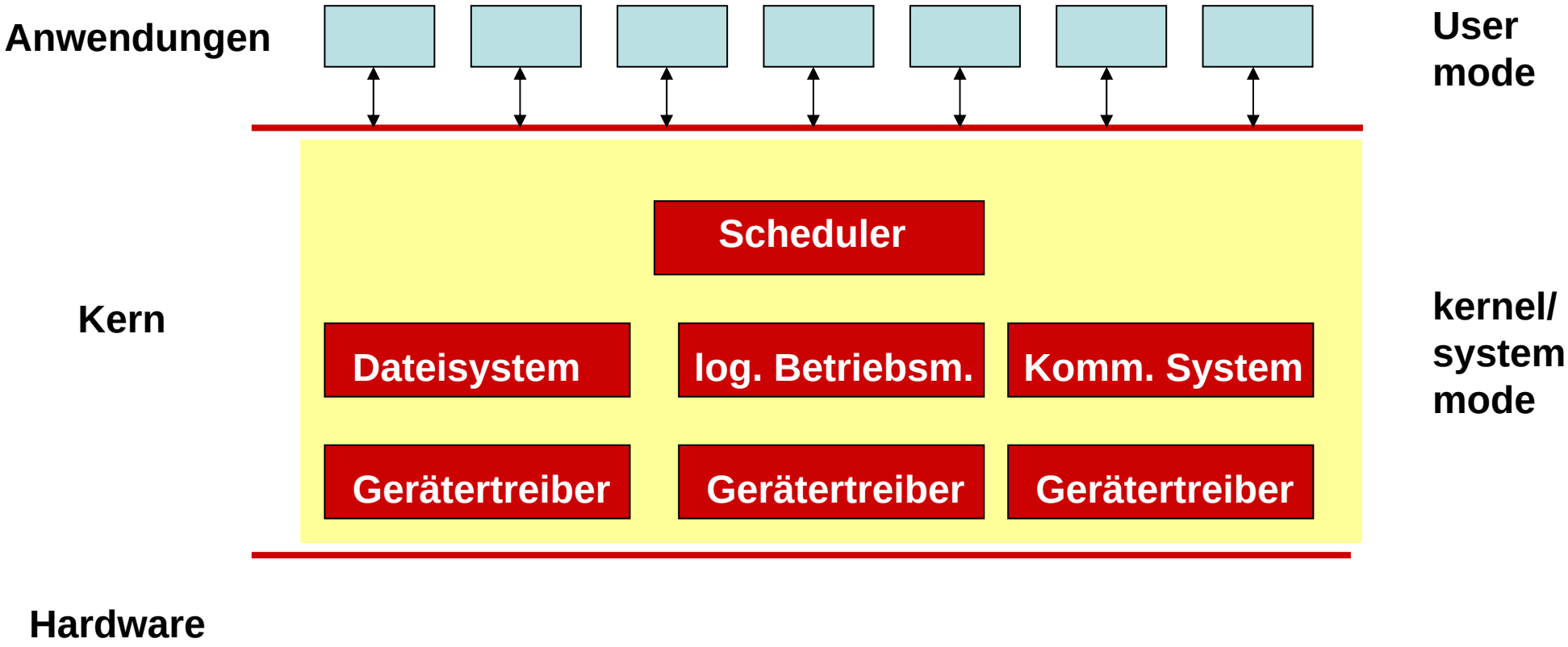
- explizite Software Traps:
  z.B. TRAP, CHK,
- Ausnahme, die bei der Befehlsabarbeitung auftreten:
  z.B. Illegale Instruktion,
  Division durch 0
  Adressierungsfehler
  Privilegverletztung
- Interrupts



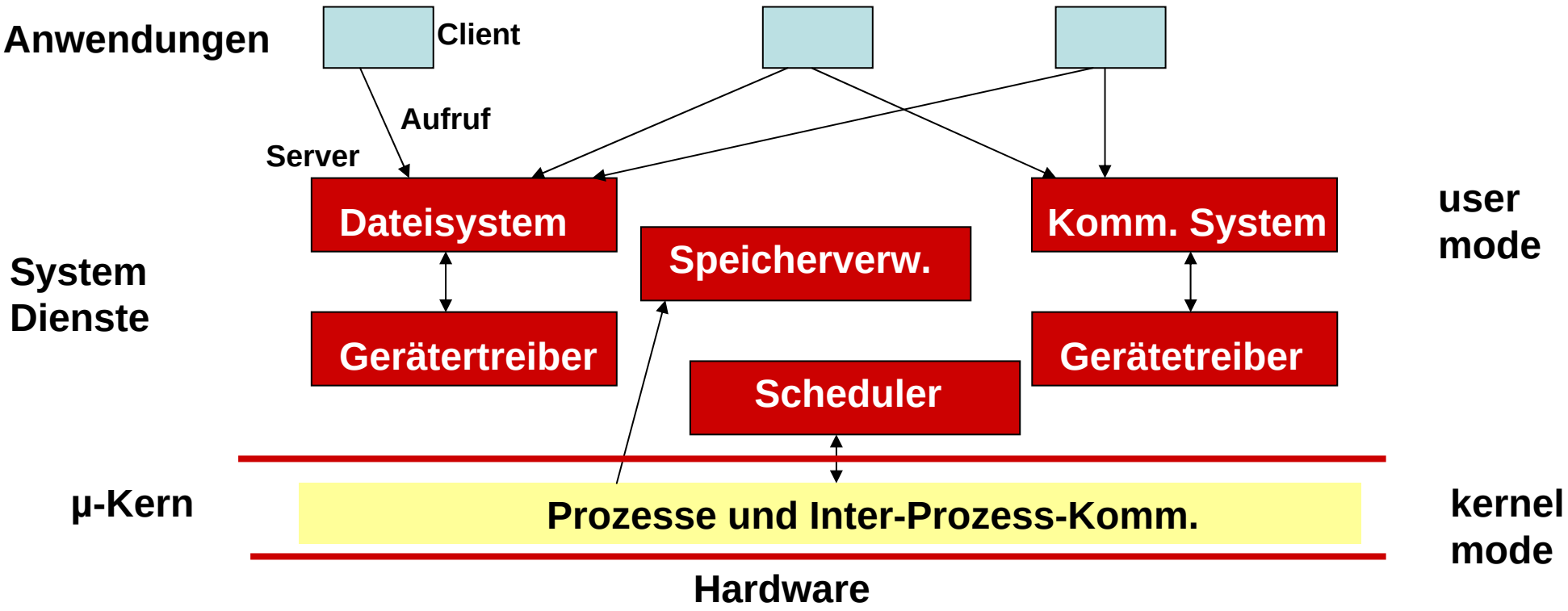
Trennung der Adressräume durch MMU



Monolithische Betriebssystemstruktur



Micro-Kern Betriebssystemstruktur

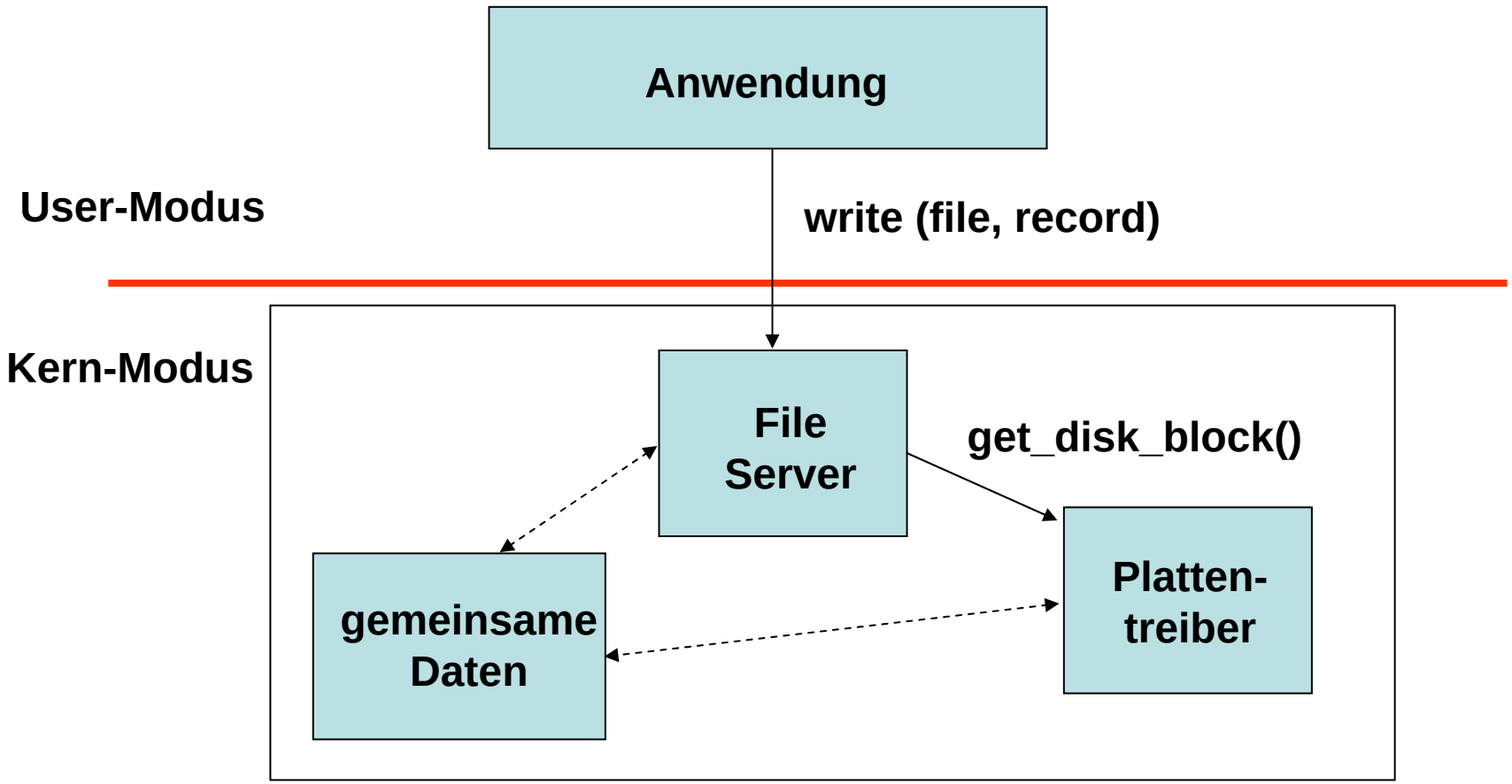


The Unix kernel is an I/O Multiplexer more than a complete operating system. This is as it should be.

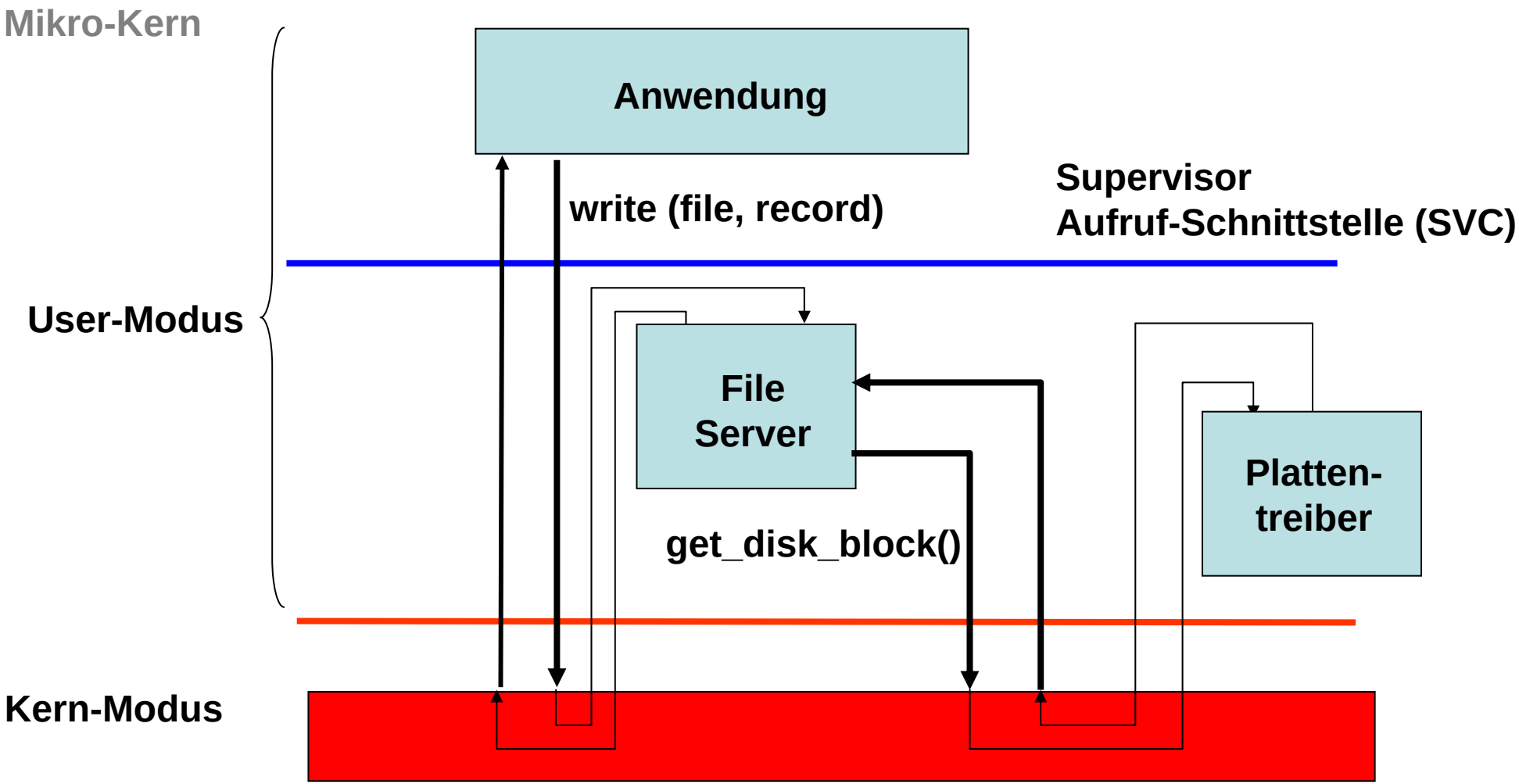
Ken Thompson: Unix Implementation, Bell Systems Technical Journal, 57(6), 1978

Kosten des Aufrufs einer Systemfunktion

Monolithischer Kern



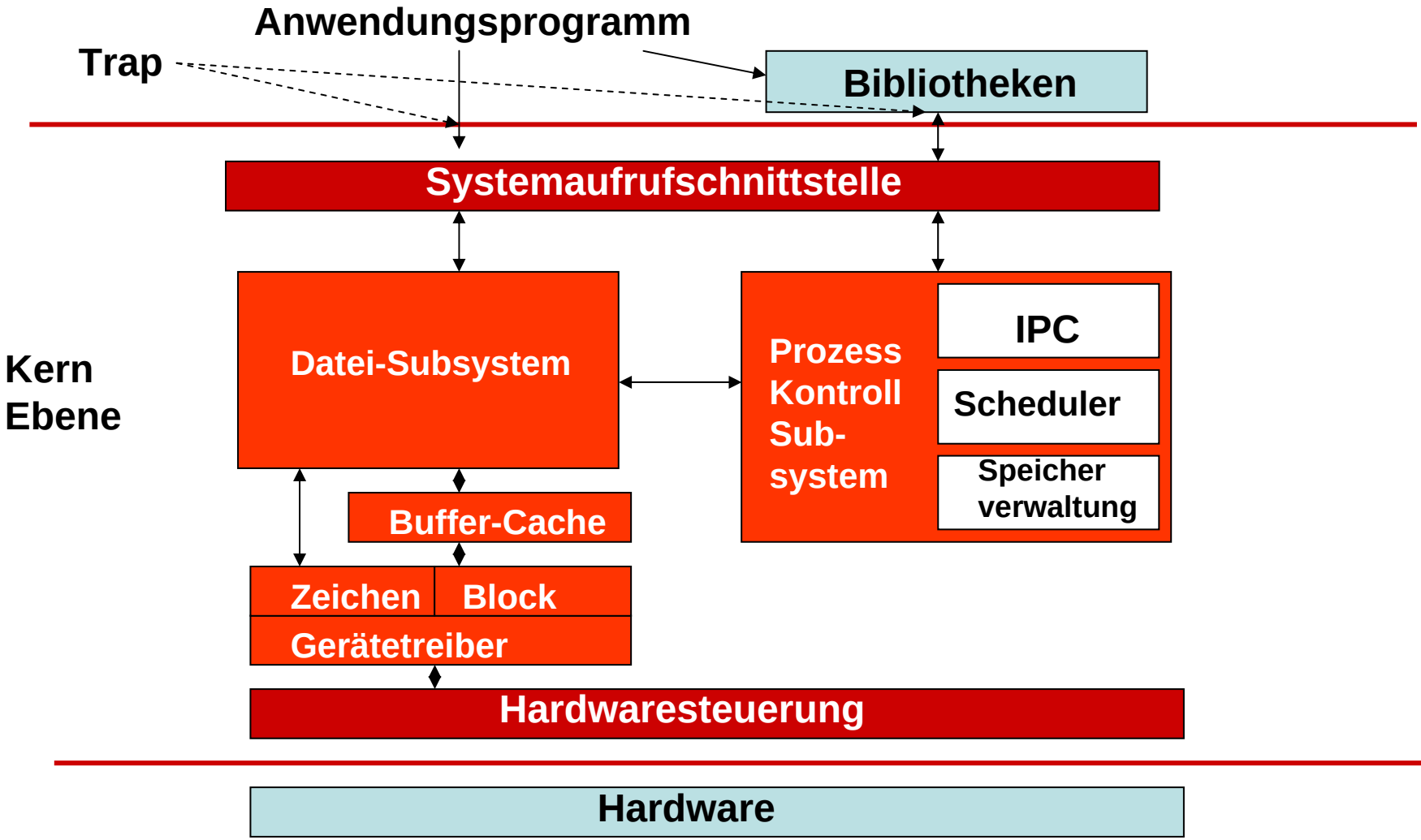
Kosten des Aufrufs einer Systemfunktion



Monolitisch vs. µ-Kern: Pro und Con

	Monolith	µ-Kern
Modularität		 Fehlereingrenzung
Erweiterbarkeit	 	 neue Dienste können einfach hinzugefügt werden.
Schutz	User/System. Kein Schutz der Kernfkt. gegenein-ander	User-Prozesse gegeneinander geschützt. kl. Kern (besser verifizierbar)
Flexibilität		Emulation versch. BS.
Performance		 viele Prozess- und Moduswechsel

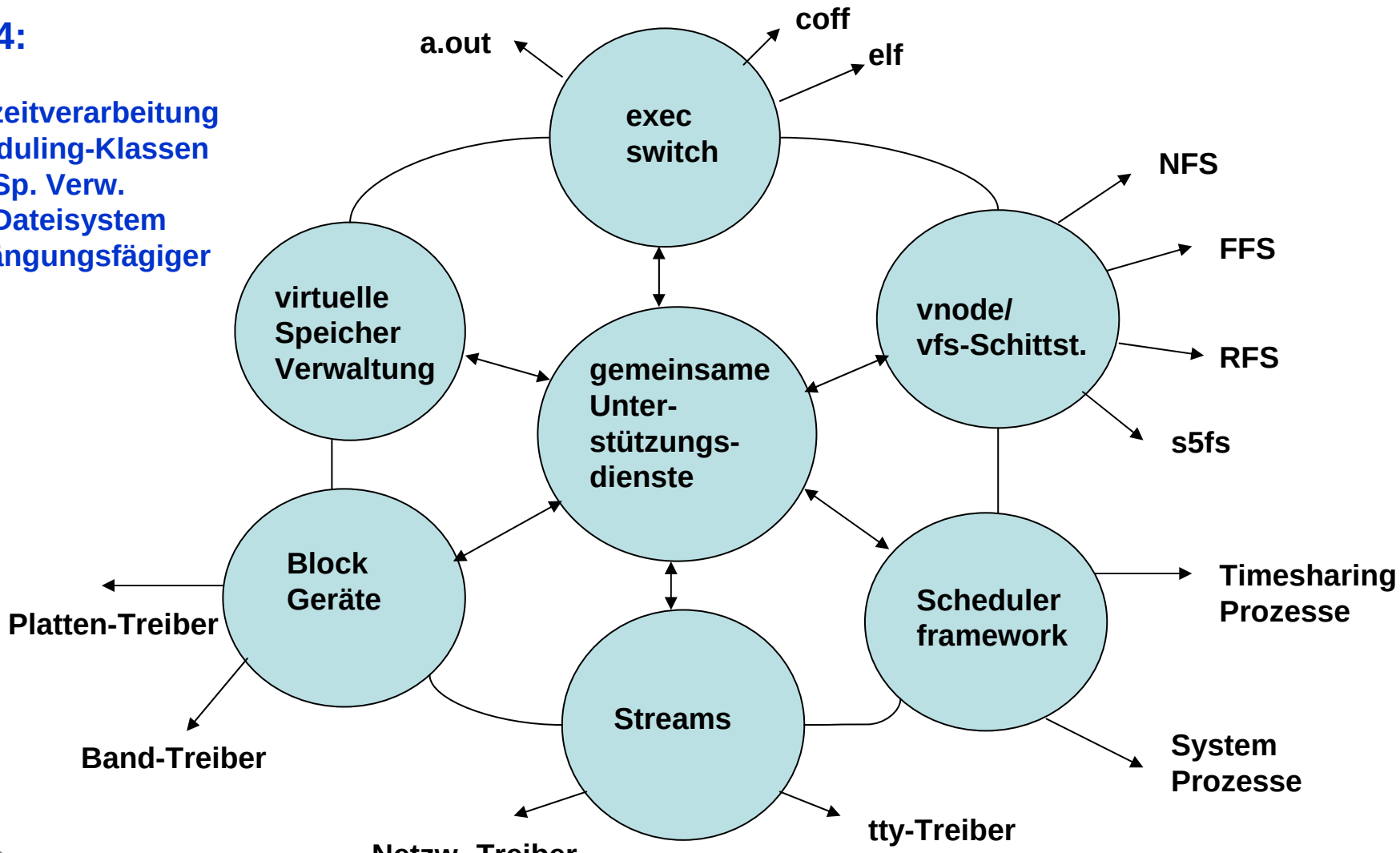
Herkömmlicher Unix-Kern (Bach 1986)



Moderner Unix-Kern (Vahalia 1996)

SVR4:

- Echtzeitverarbeitung
- Scheduling-Klassen
- Virt. Sp. Verw.
- Virt. Dateisystem
- verdängungsfägiger Kern



Betriebssystemstrukturen

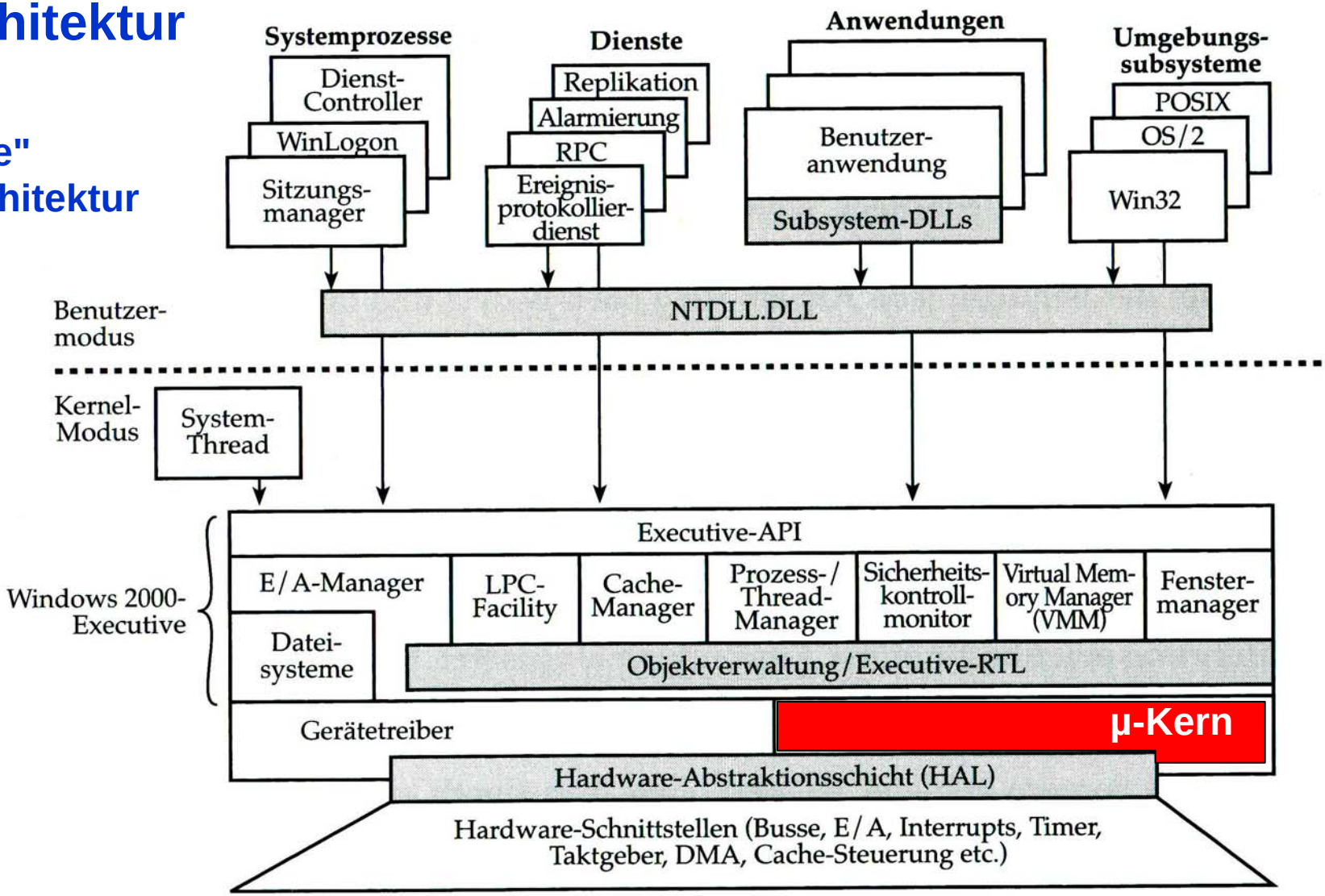
Probleme mit einem monolithschen Kern:

- **alle funktionalen Komponenten des Kerns haben Zugriff auf sämtliche internen Datenstruktren und Routinen.**
- **werden an einem beliebingen Teil Änderungen vorgenommen, müssen 1. alle Module und Routinen neu gebunden und installiert werden und 2. das System neu gebootet werden.**
- **Problembehebung (z.B. bei Linux):**
  Strukturierung des BS in relativ unabhängige Funktionsblöcke, sogenannte "Ladbare Moduln", die dynamisch geladen und gebunden werden können.

Es bleibt das Problem des nicht vorhandenen Schutzes der Funktionsblöcke untereinander.

W2K Architektur

"modifizierte"
µ-Kern Architektur



Stallings: Betriebssysteme

"modifizierte" µ-Kern Architektur

µ-Kern: Scheduling von Threads, Prozessumschaltung, Ausnahme- und Interrupt-Behandlung, Synchronisation (resident im Speicher, wird nicht verdrängt).

Moduln der Exekutive (laufen als Threads im Systemprozess):

E/A-Manager: Einheitlicher Rahmen für E/A, API für alle E/A, Sicherheitsziele, Zuordnung logischer Namen.

Objektmanager: Verwaltung, Erstellung, Löschen von Objekten der Exekutive wie: Prozesse, Threads, Dateien, Adressräume, E/A-Geräte, Sync.-Obj. , Namenszuordnung, Sicherheitsaspekte.

Prozess/Thread Manager: verwaltet Prozesse und Threads.

Sicherheitskontrollmonitor: Durchsetzung von Zugriffskontrollen.

LPC Facility: Local Procedure Call unterstützt lokale Client/Server Beziehungen.

Virtual Memory Manager: unterstützt virtuellen Speicher.

Cache Manager: verbessert die Leistung dateibasierter E/A .

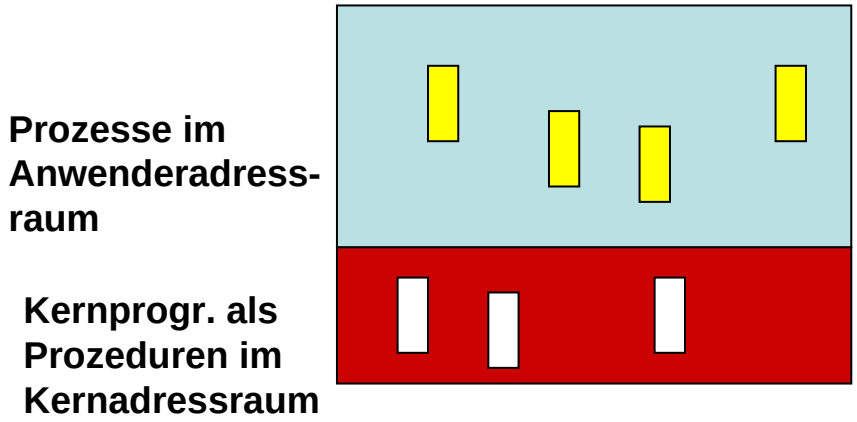
Grafik-Modul: unterstützt die fensterorientierte Bildschirmschnittstelle, verwaltet Grafikgeräte.

Realisierungsformen

Residente Realisierung:

Betriebssystem besitzt einen eigenen Adressraum und ist (mehr oder weniger) vollständig im Speicher unabhängig von Anwendungsprogrammen.

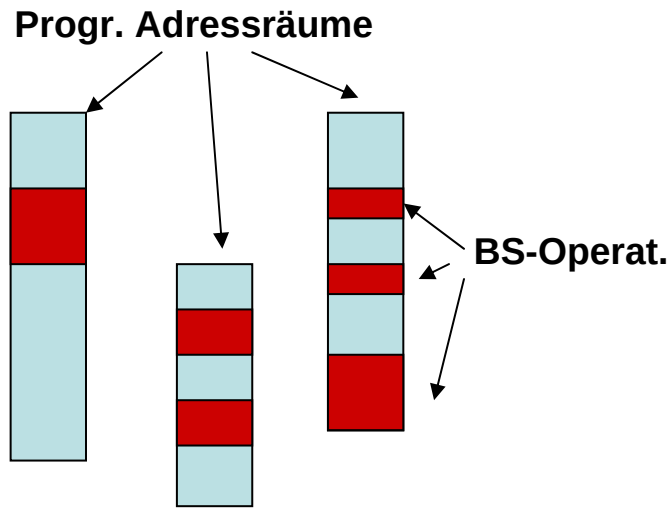
Standard bei General Purpose BS.



Realisierung als Funktionsbibliothek:

Benötigten Betriebssystemfunktionen werden beim Compilieren oder Binden zur Anwendung hinzugebunden.

Standard bei BS für eingebettete Systeme



Diskussion und Zusammenfassung

Ein Betriebssystem ist ein komplexes Softwaresystem und Abstraktion und Strukturierungsmassnahmen sind wesentlich zum Verständnis und zur Realisierung.

Abstraktion wird eingesetzt, um den Umgang mit Betriebsmitteln zu erleichtern.

Schichtung ist eine Prinzip, um Abstraktionen zu strukturieren.

Modularisierung ist ein wesentliches Konzept, um eine adäquate Realisierung zu erreichen.

Qualitätsziele:

Dynamische Erweiterbarkeit und Einschränkung, Flexibilität, Testbarkeit, Wartbarkeit, Sicherheit und Zuverlässigkeit, Effizienz, Portierbarkeit.