

# z/OS Grundlagen

## Einführung und Überblick

**cps4it**

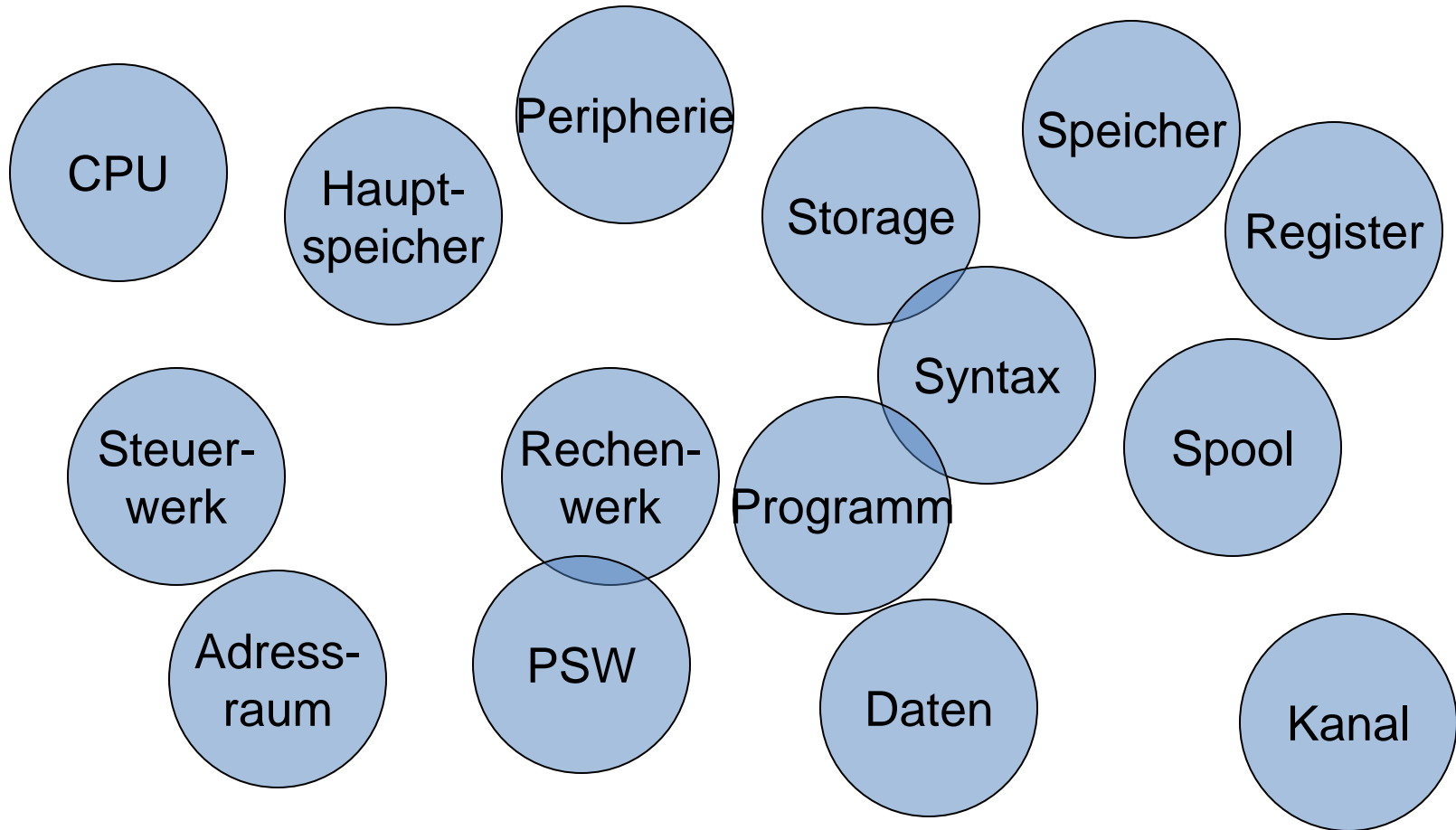
consulting, projektmanagement und seminare für die informationstechnologie

Ralf Seidler, Stromberger Straße 36A, 55411 Bingen

Fon: +49-6721-992611, Fax: +49-6721-992613, Mail: [rseidler@cps4it.de](mailto:rseidler@cps4it.de)

Internet: <http://www.cps4it.de>

- 
- ➔ • Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)



- P. Herrmann, W.G. Spruth: „Einführung in z/OS und OS/390: Web-Services und Internet-Anwendungen für Mainframes “. Oldenbourg 2017, ca. ~~55~~ 65 €
- M. Teuffel, R. Vaupel: „Das Betriebssystem z/OS und die zSeries“. Oldenbourg 2004, ca. ~~65~~ / 75€!

- W. Greis: „Die IBM-Mainframe-Architektur: z/OS, z/VM und Linux “. Open Source Press, 2005, ca. 50€ -> derzeit nicht verfügbar
- M. Teuffel: „TSO Time Sharing Option im Betriebssystem OS/390“. Oldenbourg, ca. 95€ - > derzeit nicht verfügbar
- <http://www-03.ibm.com/systems/z/> Einstiegsseite der IBM zur z-Series, 0€
- <http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp> Knowledge Center der IBM, 0€
- <http://wiki.cps4it.de/> 0€

- einige Seiten der Uni Leipzig
  - <https://ti.informatik.uni-leipzig.de/workgroup/mainframes-und-enterprise-computing/>
  - <http://jedi.informatik.uni-leipzig.de/de/> (leider veraltet, die Informationsseiten selbst sind aber Spitze)

## Terminologie

---

- Hardware: zSeries
- Betriebssystem: z/OS, früher OS/390, MVSxxx
- Rechner (alt): S/390, S/370, S/360
- Betriebssysteme (alt): OS/390, OS/360, MVS
- **z**Series und **z**/OS mit 64/128 Bit-Unterstützung
- aktuelle zSeries-Implementierungen: z16, z15,...
  - z16 vom April 2022
- kleine Modelle z900, z990, z10
- Umgangssprache: Mainframe oder Host
- [https://de.wikipedia.org/wiki/Z\\_Systems](https://de.wikipedia.org/wiki/Z_Systems)

**Zero Downtime**

## einige wenige Zahlen zu zEC12

---

- fast 100% Verfügbarkeit
- 5,5 GHz
- > 78.000 Mio MIPS
- 3 TB Memory
- 32 GB HSA (Hardware System Area – z.B. I/O)
- CMOS-Prozessoren und Intel-Prozessoren
- bis 101 Prozessoren
  - unterschiedlich definierbar als zIIP, zAAP, IFL, CP, SAP
- „ein System der Superlative“



CP	Central Processors (CPs)
IFL	Integrated Facility for Linux
zAAP:	System z Application Assist Processors
zIIP	System z Integrated Information Processors
ICF	Internal Coupling Facilities
SAP	System Assist Processors



einige wenige Zahlen zu z13 (mehr als 500 neue Patente)

- 13. Generation CMOS (1994)
- fast 100% Verfügbarkeit
- 5,0 GHz
- > 111.000 Mio MIPS
- 10,0 TB Memory
- 64 GB HSA (Hardware System Area – z.B. I/O)
- CMOS-Prozessoren und Intel-Prozessoren
- bis 141 Prozessoren
  - unterschiedlich definierbar als zIIP, zAAP, IFL, CP, SAP

Model	Processor Drawers	Installed Pcs	Standard SAPs	Maximum Characterized Pcs	Min – Max Customer Memory (GB)
N30	1	39	6	30	64 – 2464
N63	2	78	12	63	64 – 5024
N96	3	117	18	96	64 – 7584
N129	4	156	24	129	64 – 10144
NE1	4	168	24	141	64 – 10144



CP Central Processors (CPs)  
IFL Integrated Facility for Linux  
zAAP: System z Application Assist Processors  
zIIP System z Integrated Information Processors  
ICF Internal Coupling Facilities  
SAP System Assist Processors  
IFP Integrated Firmware Processor

einige wenige Zahlen zu z14 ([Link](#))

---

- 14. Generation CMOS (1994)
- fast 100% Verfügbarkeit
- 5,2 GHz
- ? Mio MIPS
- 32 TB Memory
- 192 GB HSA (Hardware System Area – z.B. I/O)
- CMOS-Prozessoren und Intel-Prozessoren
- bis 170 Prozessoren
  - unterschiedlich definierbar als zIIP, zAAP, IFL, CP, SAP



CP	Central Processors (CPs)
IFL	Integrated Facility for Linux
zAAP:	System z Application Assist Processors
zIIP	System z Integrated Information Processors
ICF	Internal Coupling Facilities
SAP	System Assist Processors
IFP	Integrated Firmware Processor

## einige Aussagen zu z16

---

- KI und Cyber-Ausfallsicherheit
- Hybrid Cloud
- On-Chip-KI-Beschleunigung
- Quantensichere Kryptoerkennung



- Quellen
  - <https://www.ibm.com/de-de/products/z16>
  - <https://www.ibm.com/docs/en/systems-hardware/z/Z16M-A01>

## Hat der Host, hat z/OS eine Zukunft?

- Anteil der MIPS
- Anteil der Daten
- Anteil des Programmcodes
- Anteil der Patente
- Umfragen / Reports: [Link](#) [Link](#) [Link](#)
- Neue Begriffe
  - Blockchain, Cloud, DevOps
  - Encryption, Machine Learning
  - Open Source, Operations
  - Docker, Container ...

The world's businesses run on IBM Z

- ✓ The 10 top insurers
- ✓ 44 of the top 50 banks
- ✓ 18 of the top 25 retailers
- ✓ 90% of the largest airlines

[Link 2011: 100 Jahre IBM](#)

[Link 2014: IBM-Patente für 2013](#)



## Umfragen / Veröffentlichungen aktuell

---

- FTD Februar 2009 
  - [http://www.ftd.de/technik/it\\_telekommunikation/:Schnelster-Rechner-der-Welt-IBM-baut-den-Super-Computer/469428.html](http://www.ftd.de/technik/it_telekommunikation/:Schnelster-Rechner-der-Welt-IBM-baut-den-Super-Computer/469428.html)
- Computerwoche April 2010: 
  - <http://www.computerwoche.de/mittelstand/1892930/>
- Computerwoche April 2010: 
  - <http://www.computerwoche.de/subnet/oracle/1934491>
- ComputerWeekly Mai 2020:
  - <https://www.computerweekly.com/feature/Mainframe-storage-Three-players-in-a-market-thats-here-to-stay>

## Aussagen

---

- A fairly well accepted notion in computing is that the mainframe is going the way of the dinosaur.  
Forbes, March 20, 1989
- The mainframe computer is rapidly being turned into a technological Dinosaur... New York Times, April 4, 1989
- *On March 15, 1996, an InfoWorld Reader will unplug the last mainframe. InfoWorld 1991*
- ...the mainframe seems to be hurtling toward extinction.  
New York Times, Feb. 9, 1993
- *Its the end of the end for the mainframes*  
George Colony, Forrester Research, Business Week,  
Jan. 10, 1994

## Zahlen

---

- 95% der weltweit größten 2000 Unternehmen setzen OS/390 oder z/OS als ihren zentralen Server ein. Insgesamt 20 000 Unternehmen verfügen über einen S/390- oder z-Rechner.
- Zwischen 65 und 70 % aller geschäftsrelevanten Daten werden im EBCDIC Format auf z-Rechnern gespeichert.
- 60% aller geschäftsrelevanten Daten, auf die mittels des World Wide Web zugegriffen werden kann, sind in Mainframe Datenbanken gespeichert, hauptsächlich DB2, IMS und VSAM
- etc.

## Zahlen (27.07.2015)

---

- 71% of all Fortune 500 companies have their core business on the mainframe
- 23 of the world's top 25 retailers use a mainframe
- 92% of the top 100 banks use a mainframe
- 10 out of 10 of the top insurers use a mainframe
- More than 225 state and local governments worldwide rely on a mainframe
- 9 of the top 10 global life and health insurance providers process their high-volume transactions on mainframe.

Quelle: <http://enterprisesystemsmedia.com/article/rehabilitating-the-perception-of-mainframes>



## Zahlen (19.08.2015) – 1

---

- 88 percent of CIOs believe the mainframe will be a key business asset over the next decade
- 89 percent of CIOs consider their organization's mainframe code to be valuable intellectual property
- Despite this, investments in mainframe security outpace investment in mainframe intellectual property by a factor of more than 3:1
- 78 percent of CIOs see the mainframe as a key enabler of innovation
- 81 percent of CIOs say the mainframe can deliver greater Big Data throughput than commodity hardware alone, with 61 percent using it for this purpose.

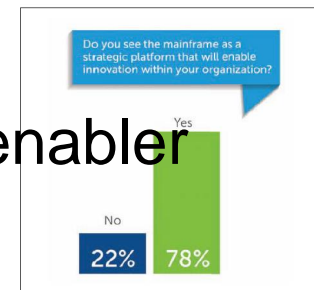


Figure 1: Far From Seeing The Mainframe as Merely a "Legacy" Platform, CIOs Believe It Plays a Central Role in the Digital Transformation of Their Organizations.

## Zahlen (19.08.2015) – 2

---

- 70 percent of CIOs are concerned that lack of mainframe documentation will hinder knowledge transfer and create risk
- 39 percent of CIOs have no explicit plans for addressing mainframe developer shortages
- 75 percent of CIOs admit that their distributed application developers have little understanding of the mainframe

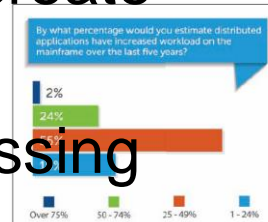
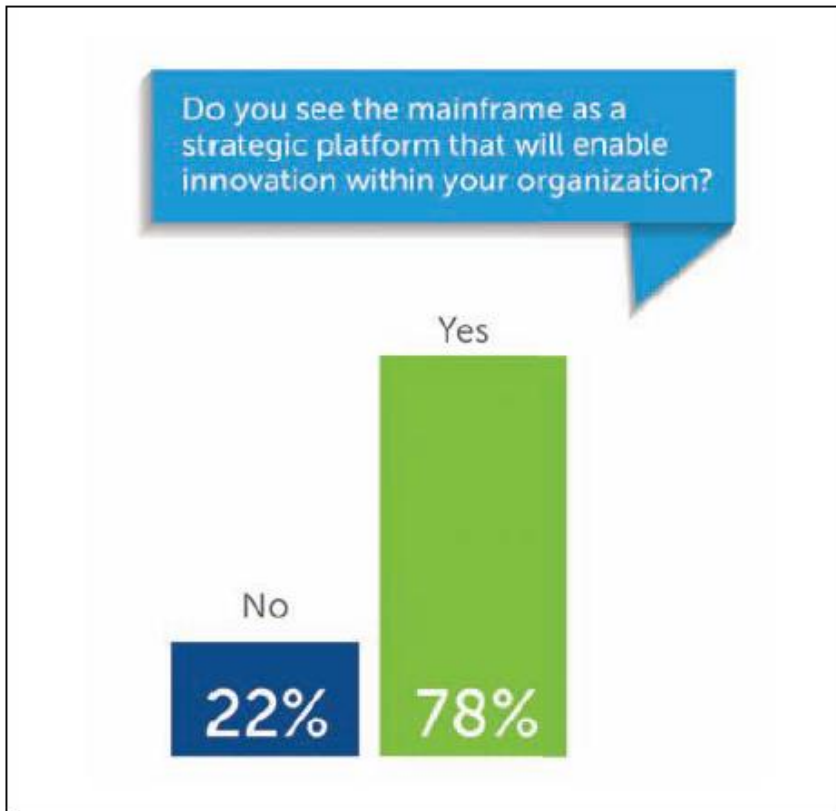


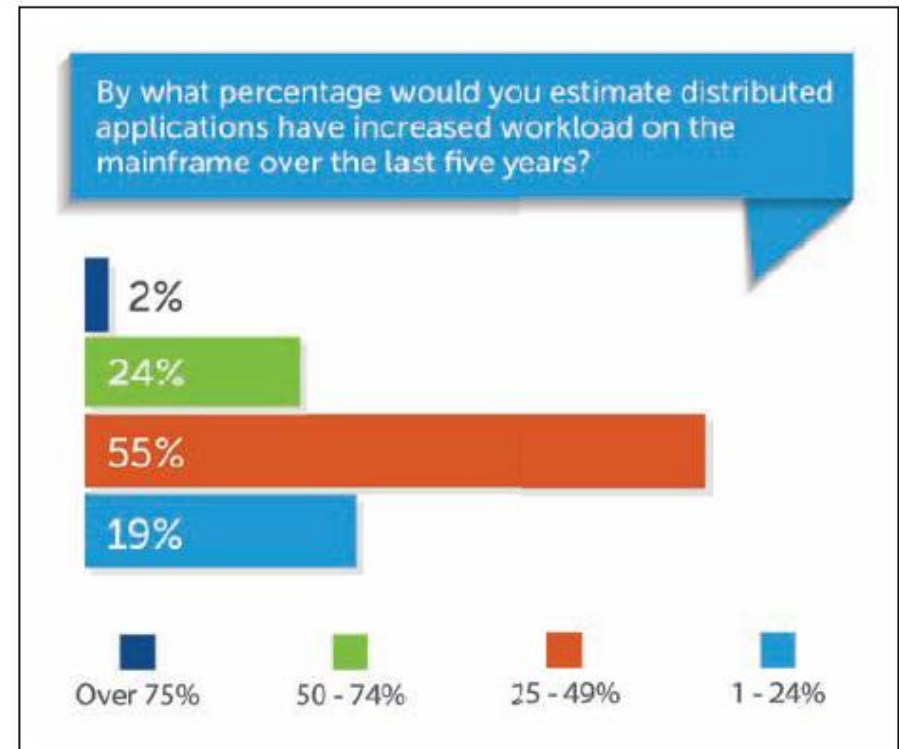
Figure 2: Surveyed CIOs Confirmed That Their Mainframe Workloads Continue to Grow With an Overall Average Growth of Approximately 39 Percent.

Quelle: <http://ourdigitalmags.com/publication/?i=269252> / These findings and others are the result of a new study of 350 CIOs at large companies conducted by Vanson Bourne, covering a cross-section of vertical markets in Europe, Australia and the U.S.

## Bilder (19.08.2015) – 1 (identische Quelle)

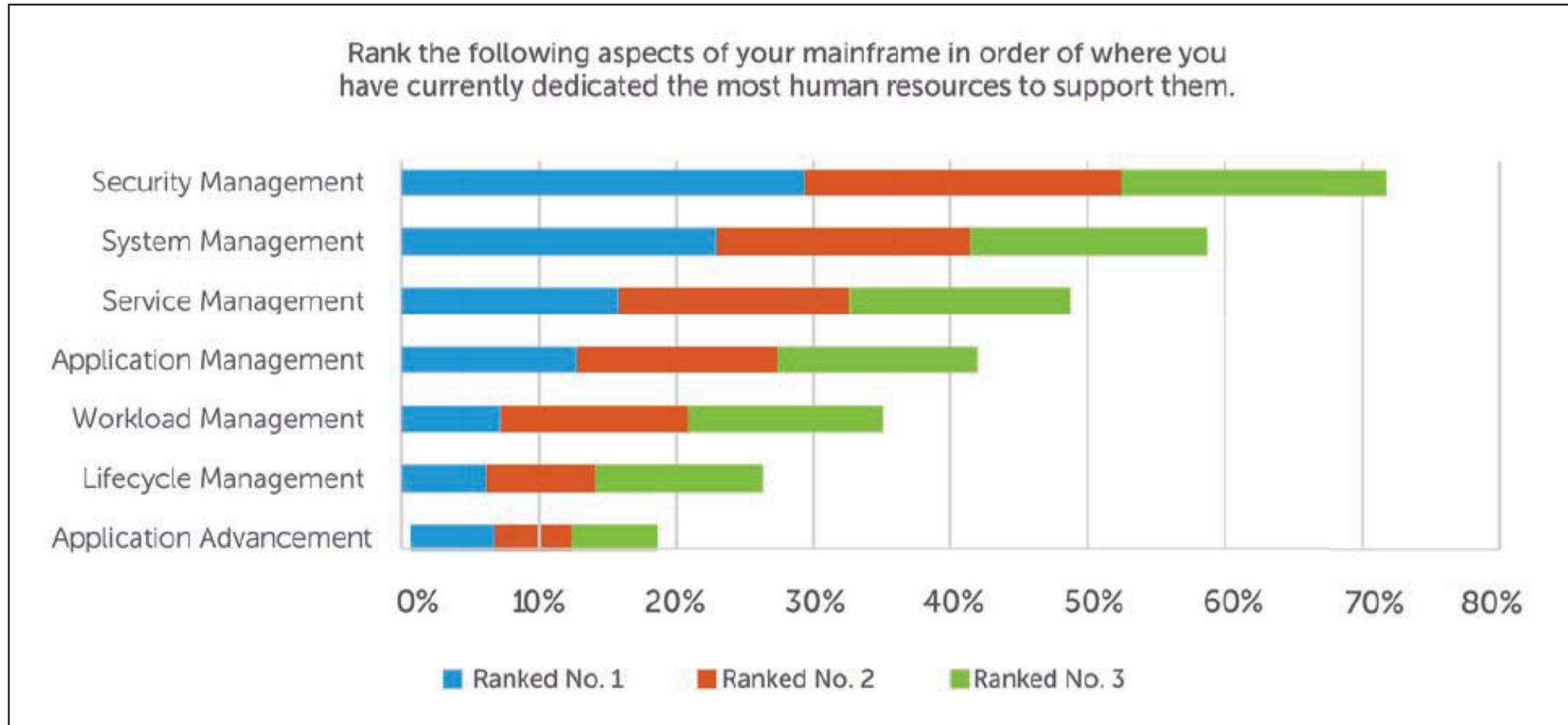


**Figure 1: Far From Seeing The Mainframe as Merely a “Legacy” Platform, CIOs Believe It Plays a Central Role in the Digital Transformation of Their Organizations.**



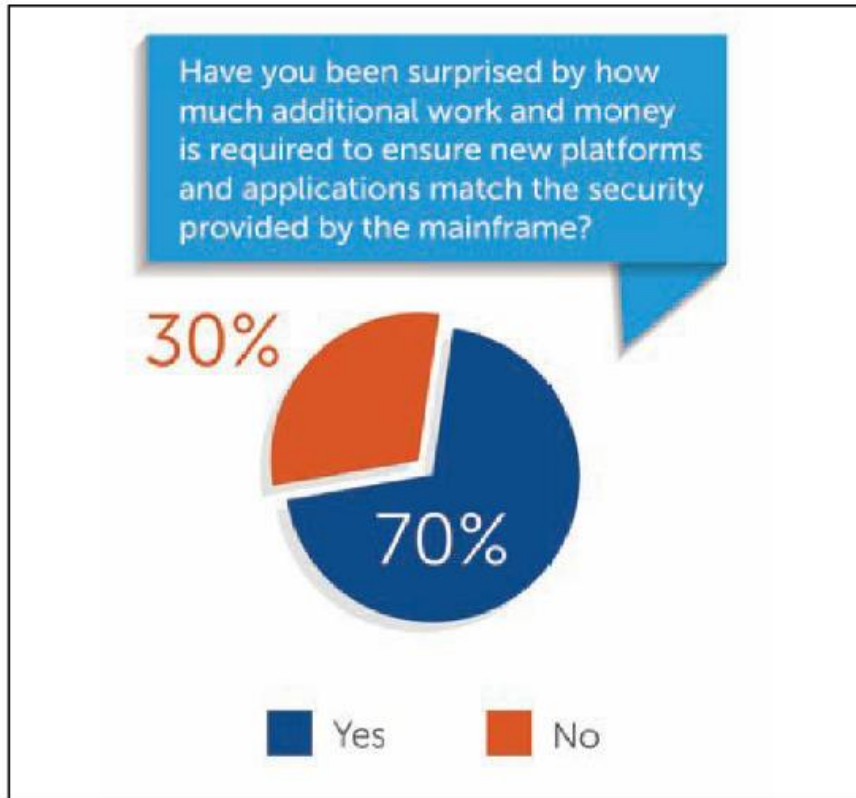
**Figure 2: Surveyed CIOs Confirmed That Their Mainframe Workloads Continue to Grow With an Overall Average Growth of Approximately 39 Percent.**

## Bilder (19.08.2015) – 2 (identische Quelle)

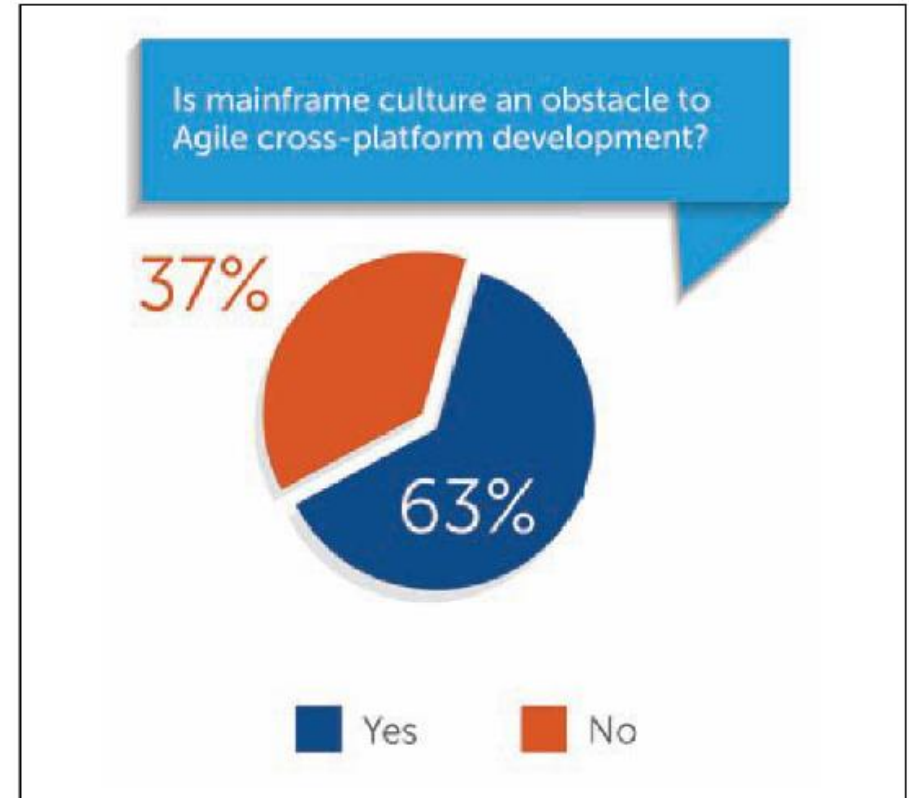


**Figure 3: While CIOs Acknowledge the Strategic Role of the Mainframe, They Tend to Prioritize Investment in Non-Strategic Platform Attributes Such as Security and System Management Over the Applications That Actually Comprise Their Companies' IP.**

## Bilder (19.08.2015) – 3 (identische Quelle)



**Figure 4: Non-Mainframe Platforms Often Disappoint CIOs When It Comes to Achieving Security Within Projected Budgets.**



**Figure 5: Integration and Optimization of Mainframe Development Is a Cultural Issue as Much as a Technical One.**

## Zahlen (26.05.2020)

---

- Mainframes are in use by
  - 92 of the world's top 100 banks
  - 23 of the 25 top airlines
  - all of the world's top 10 insurers
  - 71% of Fortune 500 companies

Quelle: <https://www.computerweekly.com/feature/Mainframe-storage-Three-players-in-a-market-thats-here-to-stay> abgerufen am 26.05.2020 um 01:40 Uhr

# Einführung

## Zahlen (von einer Universität)

---

	<b>Class of 9s</b>	<b>Outage</b>	<b>Example</b>
<b>Continuous Availability</b>	<b>99,999 %</b>	<b>5 min/year</b>	<b>z/OS Parallel Sysplex</b>
<b>Fault Tolerant</b>	<b>99,99%</b>	<b>53 min/year</b>	<b>S/390 Parallel Sysplex</b>
<b>High Availability</b>	<b>99,9%</b>	<b>8,8 hrs/year</b>	<b>ES/9000 XRF Fault Tolerant Sys.</b>
<b>General Purpose</b>	<b>99%</b>	<b>88 hours/year</b>	<b>ES/9000 High Avail. Cluster SMP</b>
<b>Campus LAN</b>	<b>90%</b>	<b>876 hours/year</b>	

## Vergleich Mainframe – Server-Farm – 1

---

- Verfügbarkeit / Ausfallsicherheit? – Mainframe
- Sicherheit gegen Missbrauch? – Mainframe
- Know-How / Nachwuchs – Server
- Schnelligkeit? – ??
  - Schnelligkeit bezogen auf was?
- geringere Kosten? – ??
  - Kosten bezogen auf was?
  - Wie kann ich eine TCO (im wahrsten Sinne des Wortes) berechnen?



## Vergleich Mainframe – Server-Farm – 2

---

- Beispiel Versicherung: 1 Mio Sätze kopieren
  - SAP – x Minuten / z/OS – x Sekunden
- Beispiel Versicherung: 1 Mio Sätze aus DB lesen
  - Oracle – x Sekunden / z/OS(DB2) – x Sekunden
  - Was ist mit verteilten Datenbanken?
  - Was ist mit Übertragung?
  - Wo ist die Zeit zum Endbenutzer?
- Was sagen diese Beispiele?
- Es gibt keine verlässliche (unabhängige) Quelle, die einen nachvollziehbaren Vergleich zwischen Mainframe und z.B. Intel-Server zulässt.

und ...

---

**Alle Unix Anwendungen (nach Rekompilation) und alle Java Anwendungen sind unter den z/OS „Unix System Services“ und/oder unter zLinux lauffähig.**

**Die Auslastung für ein typisches zSeries System besteht aus:**



- 55%** „Legacy „ Anwendungen (Anwendungen die vor längerer Zeit entstanden sind)
- 35 %** Anwendungen, die in den letzten 1 - 2 Jahren geschrieben wurden
- 10%** Anwendungen, die im Rahmen von Konsolidierungsmaßnahmen übernommen wurden, davon viele von Unix Rechnern

<http://www-03.ibm.com/servers/eserver/zseries/library/literature/papers.html>

## Was ist ein Betriebssystem?

---

- Menge von Programmen
- zwischen Hardware und Anwendung
- Aufgaben
  - Bereitstellen von Steuerungs- und Hilfsfunktionen
  - Bestmögliche Nutzung der Betriebsmittel
  - Ermöglichen der Planung der Arbeitsabläufe
  - Erleichterung der Programmierung
  - Erleichterung der Systembedienung
  - Fehlerbehandlung, Datenschutz, etc.

- Rechner (Mainframe)
  - CPU(s), Hauptspeicher, Zusatzspeicher, Kanalsubsystem
- E/A-Geräte (Peripherie) für Kommunikation
  - mit Bildschirmen, mit Rechnern
  - Massenspeicher, Drucker
- Datenträger
  - Platten, Bänder, optische Datenträger

- Steuerwerk
  - Gehirn, Aktionen werden interpretiert und veranlasst
- Rechenwerk
- ausführen der Instruktionen
  - angestoßen durch Steuerwerk
- Register
  - speichern von Informationen
  - PSW ist spezielles Register
- zusätzliche Rechenregister
  - wie Fließkomma, En-/Decryption etc.

- Arbeit nur im Speicher
- Basiseinheit für Adressierung 1 Byte
  - Halbwort            16 Bit
  - Vollwort            32 Bit
  - Doppelwort        64 Bit
  - Quadword        128 Bit
- EBCDIC
- Transfer von Daten in 4k-Blöcken (page)
- Transfer von Daten in 4GB-Blöcken !!!

## Hardware - Kontrolleinheiten und Kanäle

---

- 1 Kontrolleinheit pro Gerätetyp
  - z.B. für Platte zur Positionierung und Read/Write
- gleichartige logische Funktionen
  - Kanäle - sind eigene Rechner
    - Blockmultiplex – schnell
    - Bytemultiplex – langsam
- Kommunikation über
  - „Flexe“
  - Glasfaser

- BUS
  - Verbindung zwischen CPU, Hauptspeicher und Register
  - Breite abhängig von Rechner
  - bis 64 bit
  - Frage: Was ist die Lieblingsbeschäftigung von Bits und Bytes?
- PSW
  - Adresse für nächste Instruktion



## Verbindungen mit Peripheriegeräten

---

- Command Bus
- Daten Bus
- Ablauf
  - CPU sendet SSCH (Start Sub channel) an Kanalsubsystem
  - KSS stellt Verbindung zu Gerät her
  - KSS schickt Daten über Daten Bus in HS
  - KSS schickt am Ende I/O Interrupt über Command BUS an CPU

## Adressierung der Peripheriegeräte

---

- Verbindung von Gerät über CU (Channelunit) mit Kanal
  - Kanal Adressierung x00 aufsteigend eindeutig
  - CU hat ebenfalls eindeutige Adresse
  - Adressierung ist Kombination der beiden
- Beispiel
  - Kanal Adresse 3
  - CU-Adresse 60
  - Adresse 1. Gerät 360
  - Adresse 2. Gerät 361 etc.

- Hardware ist Physik
- Software ist Logik und Ablauf
- Daten sind Beschreibung der Realität
- Programme und Daten sind Bitmuster
  - Bedeutung der Bits wird durch einen logischen Prozess bestimmt

- Addition von zwei Zahlen als Bitmuster

01011000001100001100000000000000

010110000100000011000000000000100

0001101000110100

010100000011000011000000000001000

- später Addition als Hexwerte

58 30 C0 00

- später Assembler

AH R5,=H'1' für Addiere 1 auf Registerinhalt 5

- höhere Programmiersprachen

C = A + B bzw. ADD A,B GIVING C

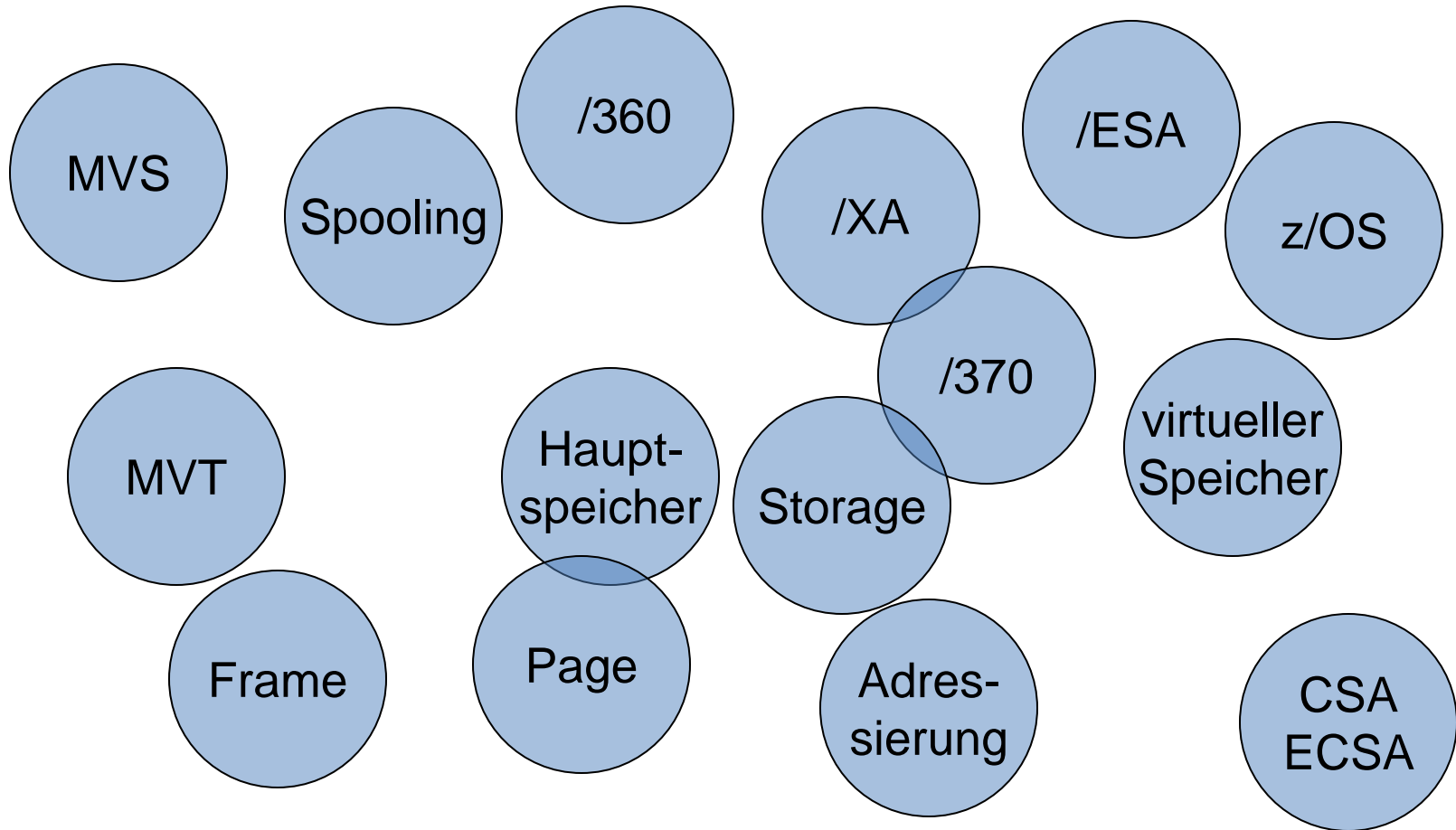
Übrigens: Es gibt 10 Arten von Menschen: Die, die das Binärsystem verstanden haben und die, die es nicht verstehen.



- 
- Einführung
  - • Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

## Begriffe

---



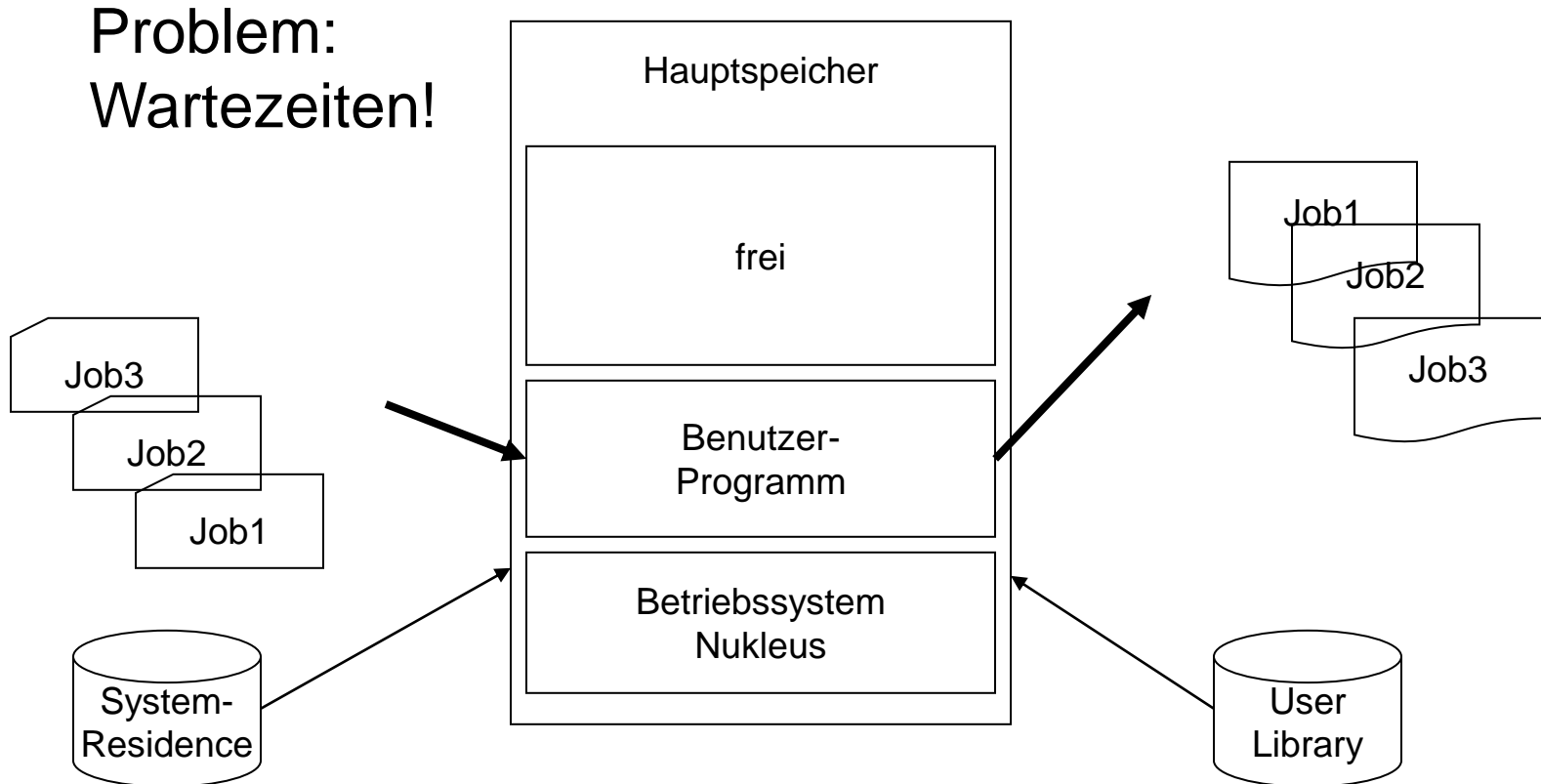
## Monoprogramming

---

- Monoprogramming
  - sequentielles Arbeiten
  - jeder hat Hoheit über alle Ressourcen
  - zunächst nur Batch
  - 1964 /360 von IBM als Rechnerfamilie
    - Objektmodule neu linken
    - Hilfsprogramme
    - Jobabläufe
    - Speicherung auf Platten
    - Peripheriegeräte zur Ausführungszeit zuweisen
    - Protokolle

**360-Revolution**

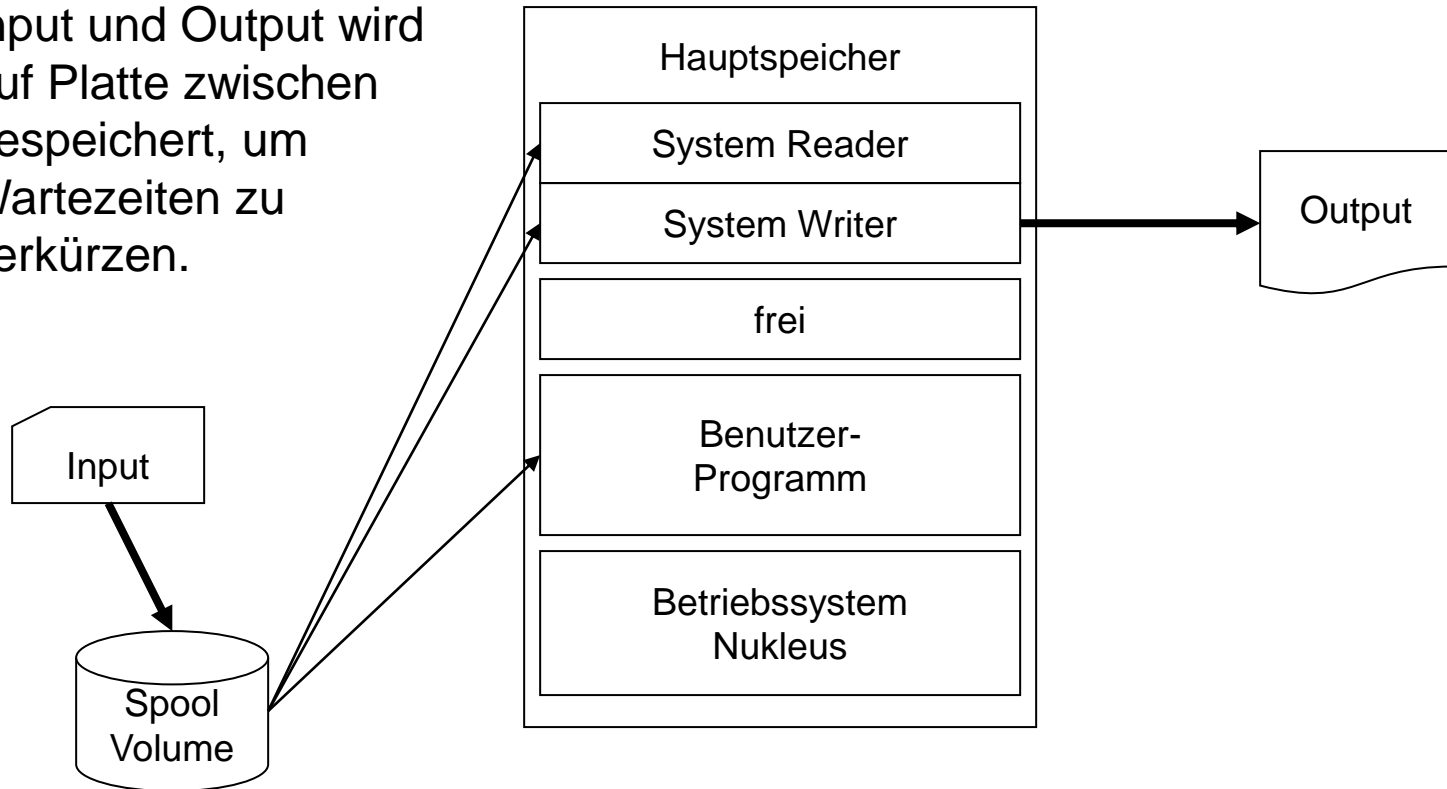
## Monoprogramming





## Spooling

Idee:  
Input und Output wird  
auf Platte zwischen  
gespeichert, um  
Wartezeiten zu  
verkürzen.



## Multiprogramming

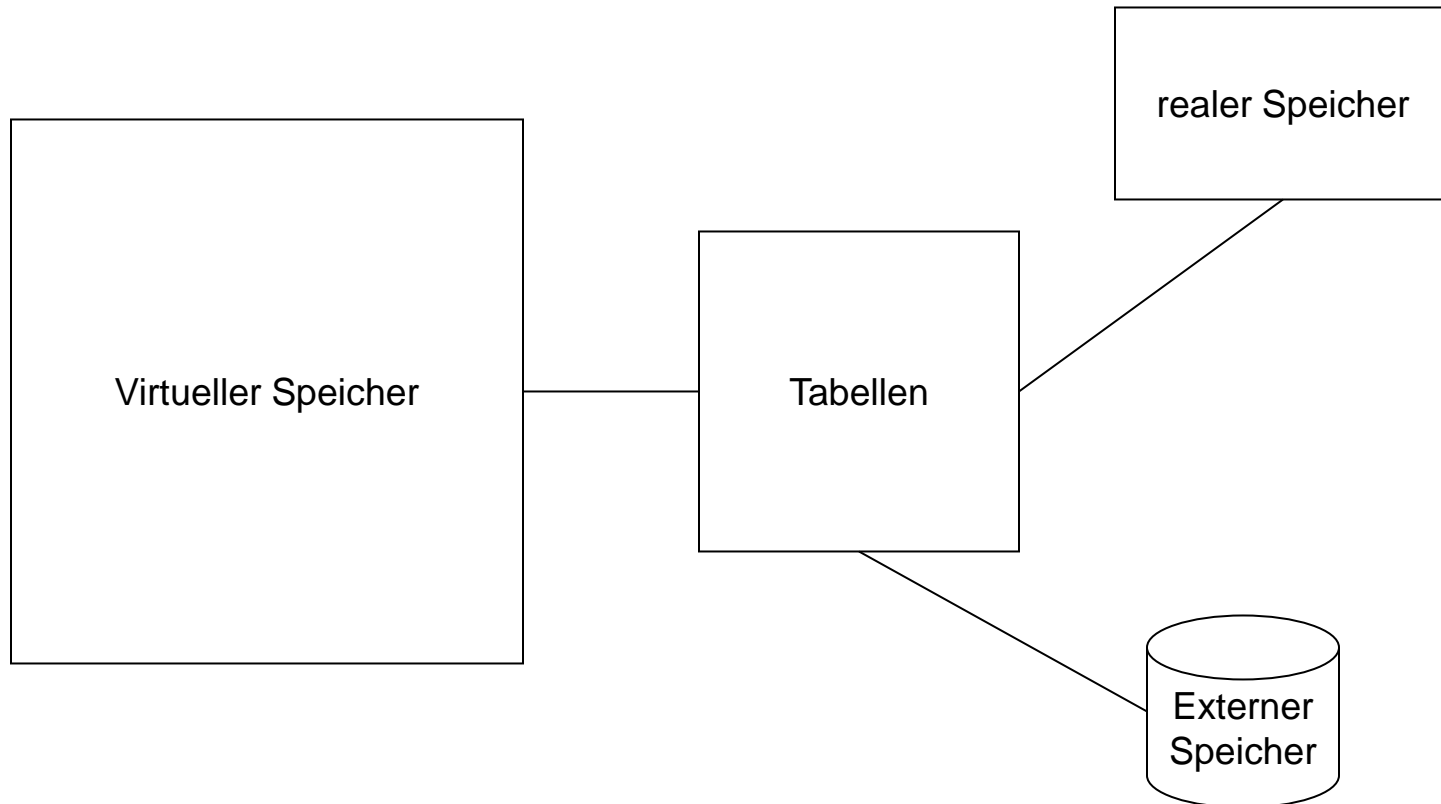
---

- Idee: Resource-Sharing
- drei Modelle
  - MFT: Multiprogramming with a fixed number of tasks
  - MVT: Multiprogramming with a variable number of tasks
  - MVS: Multiprogramming und mehrfach virtueller Speicher
- Voraussetzungen
  - Speicherschutz
  - Privilegierte Befehle
  - Interrupt möglich
  - Zeitgeber

- Herausforderungen
  - freier HS gibt es nicht am Stück
  - HS ist teuer und daher immer zu klein
- Idee: Speicher virtuell adressieren
  - Aufteilung des HS in 4k Frames
  - virtueller Speicher in 4k Pages
  - alle Programme und Daten im virtuellen Speicher
- zur Ausführung Pages in Frames laden
  - Paging mit page-in und page-out

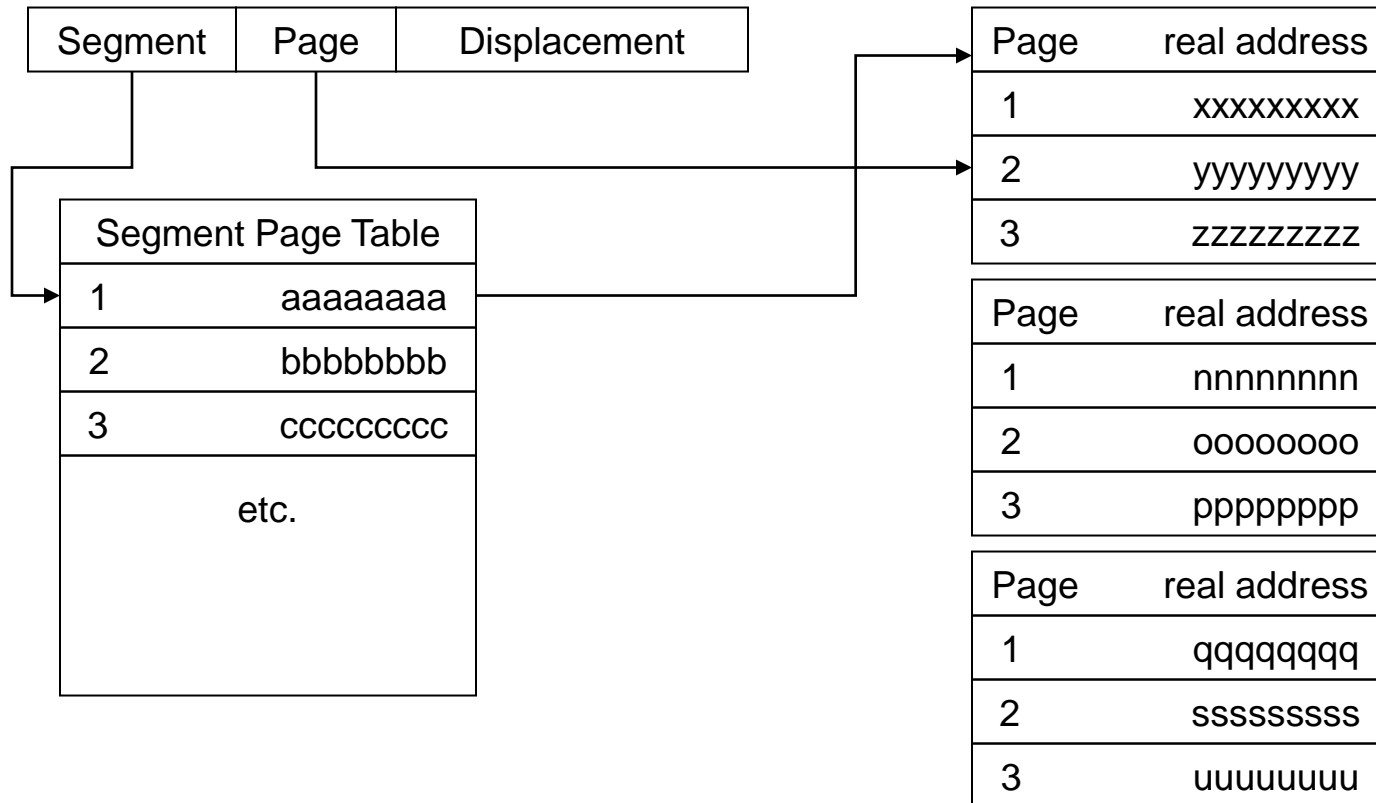
## MVS Prinzipien

---



## MVS Prinzipien

Adressierung über dynamic address translation



# Entwicklung der Betriebssysteme

## MVS/370 Storage Layout

---

SQA		
PLPA		
CSA		
LSQA	LSQA	LSQA
SWA	SWA	SWA
Region	Region	Region
Nukleus		

SQA System Queue Area  
PLPA Pageable Link Pack Area  
CSA Common System Area  
LSQA Local System Queue Area  
SWA Scheduler Work Area

## MVS/XA Prinzipien

---

- Herausforderungen
  - HS ist immer noch / wieder zu klein
- warum?
  - Programme mit höherem Komfort
  - neue Funktionen
  - zusätzliche E/A Geräte, Dateien, Datenbanken
  - mehr Benutzer, Programme, Transaktionen
  - Daten im HS halten, um I/O zu minimieren
- Adressierung von 16MB auf 2GB !

# Entwicklung der Betriebssysteme

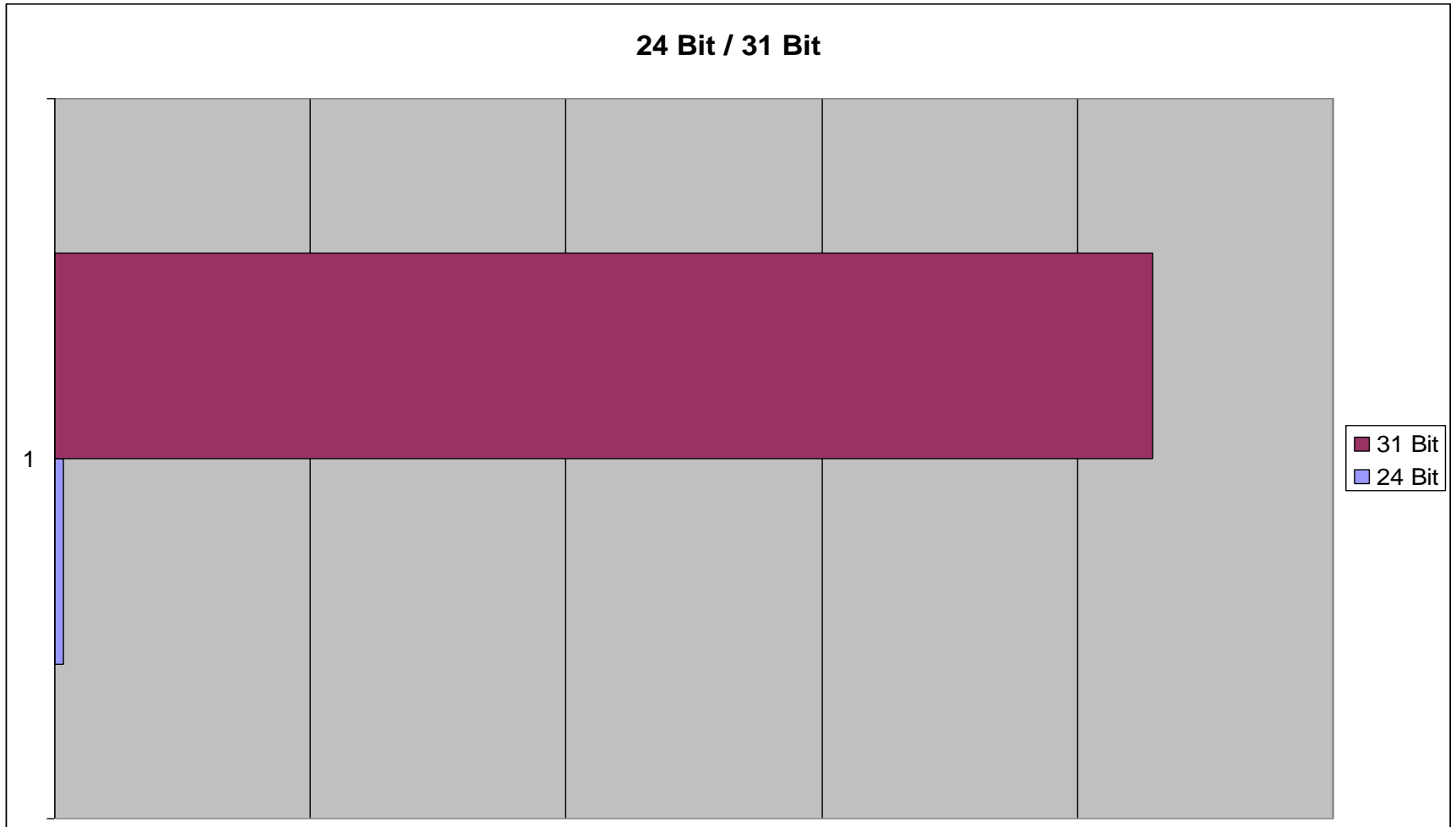
## MVS/XA Storage Layout

---

Extended Private	Extended LSQA	2 GB
	Extended User Region	
Extended Common	Nucleus	
	SQA	
	PLPA / FLPA / MLPA	
	CSA	
Common	Nucleus	16 MB
	SQA	
	PLPA / FLPA / MLPA	
	CSA	
Private	LSQA	
	User Region	
Common	PSA	4 kB



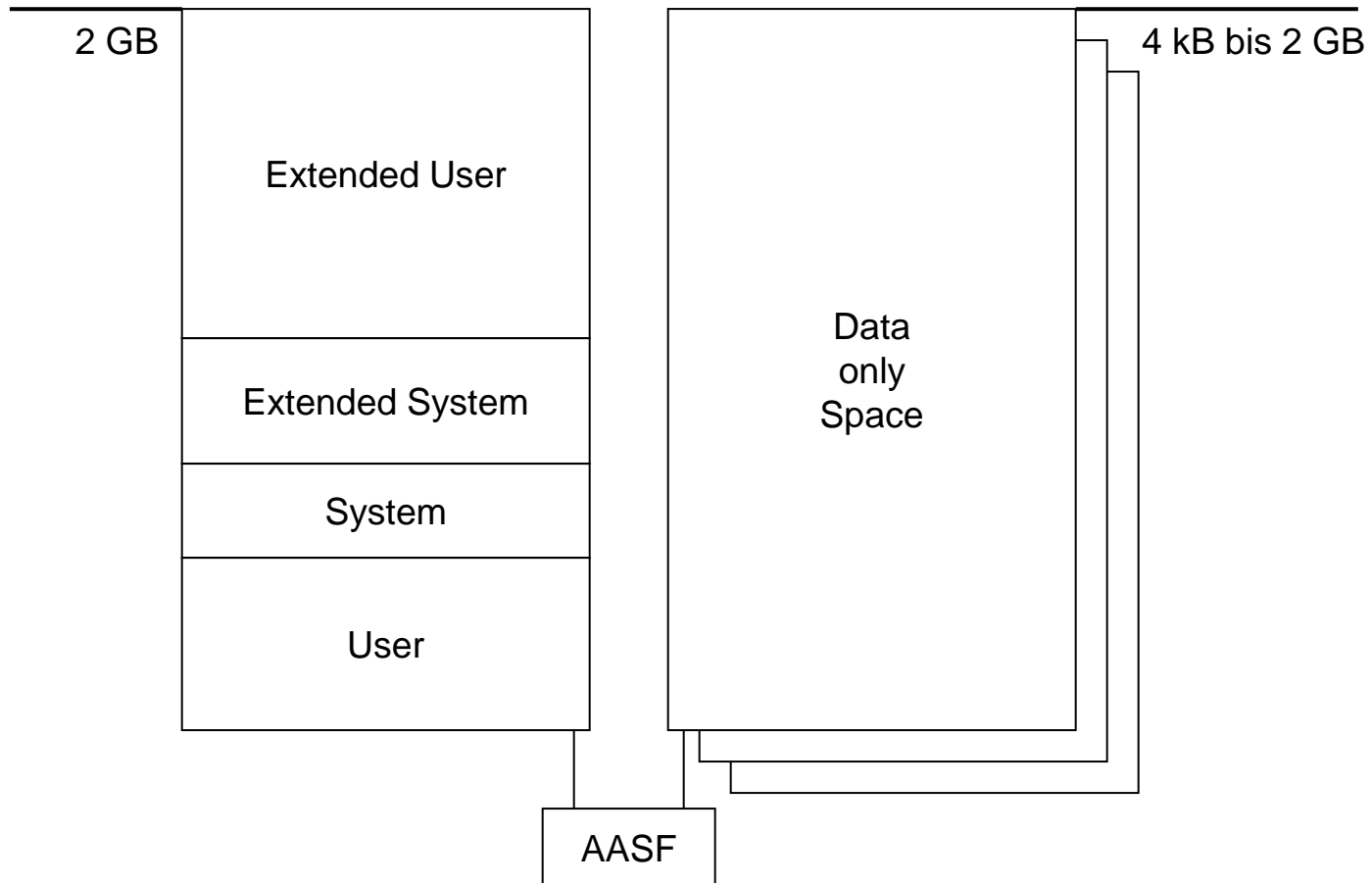
## MVS/XA Prinzipien - Speicher



- Enterprise System Architecture
- für Anwender kaum Auswirkungen
- Begriff: Advanced Address Space Facilities
- Programme und Daten in unterschiedlichen Adressräumen
- paralleler Zugriff auf Daten in unterschiedlichen Adressräumen

## MVS/ESA

---



## LPAR

---

Vergleich VM/ESA und PR/SM (LPAR)

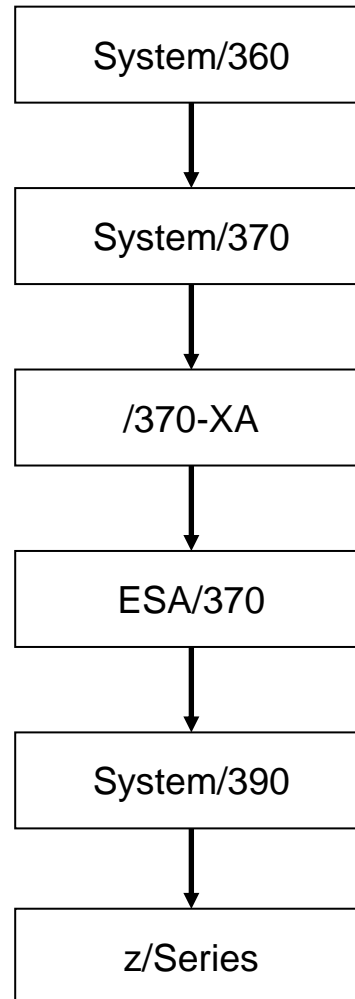
Virtual Machine (VM/ESA)							
MVS/ESA	MVS/ESA	MVS/XA	VSE/SP	CMS	CMS	CMS	CMS
Test	Prod	Prod-alt	Prod-uralt	User 1	User 2	User 3	User 4

Processor Resource / System Manager oder LPAR			
MVS/ESA	MVS/ESA	MVS/XA	VSE/SP
Test	Prod	Prod-alt	Prod-uralt



## Schritte

---



**Zero Downtime**

Das Ende des Mainframe ;-)

## Schritte

1964	S/360	Uniprocessors, single-user batch (PCP), and 24-bit addressing (16 MB). Real memory only (typically less than 128 KB)
1968		First two-way SMP. Concurrent batch (up to 15 jobs). Printer spooling.
1970	S/370	Interactive time sharing (TSO).
1971	OS/VS1	Virtual storage (up to 16 MB). Workload management of mixed workloads. Transaction managers and database management.
1974	MVS/370	Multiple 16 MB address spaces. Tools for measurement (RMF), security (RACF). Clustered systems.
1981	MVS/XA	31-bit addressing (2 GB) real and virtual. Multiple 31-bit address spaces. ynamic Channel Architecture (up to 8 paths per device). Dynamic path reconnection. Alternate Path Retry.
1983	3084	.First four-way SMP. First relational database (DB2).
1985	3090	Expanded storage (up to 16,000 TB addressability)
1987		First six-way SMP. PR/SM
1988	MVS/ESA	B1 security rating.
1990	S/390	Fiber optic channels (ESCON). I/O configuration management (ESCON Manager). Base sysplex. APPC for interprogram communication.
1993	ES/9000	First eight-way SMP.
1993		MVS OpenEdition (Version 4.3).
1994	ES/9000	First ten-way SMP. Parallel query systems (Parallel DB2). RAID disk (RAMAC). Parallel Sysplex. MVS workload manager. MVS OpenEdition (Version 5.1).
1994	9672	Technology change bipolar to CMOS.
1995		MVS OpenEdition (Version 5.2).
1996	OS/390	OS/390 Release 2 branded XPG4 UNIX compliant by X/OPEN.
1998	9672	Generation 5 - First S/390 system > 1,000 MIPS
1999	9672	Generation 6 - 200 MIPS per processor, 12 Way system, 1600 MIPS System
2000	zSeries, z/OS	64 Bit Erweiterung

## Adressierungsgrenzen

---

- 16 MB
- 2 GB
- 16 EB
- ??
- 24 Bit
- 31 Bit
- 64 Bit
- 128 Bit
- 16.777.215
- 2.147.483.647
- ca.  $18 * 10^{18}$
- ca.  $340 * 10^{36}$

Power to the people - 64-Bit-Leistung für PCs nutzbar gemacht

SUSE LINUX 9.1 Professional unterstützt serienmäßig - also ohne extra zu erwerbende Zusätze - AMDs Athlon(tm) 64 sowie Intels® bald erhältliche Extended Memory 64 Technology. Damit präsentiert SUSE ein 64-Bit-Betriebssystem mitsamt Anwendungen für Heimanwender, das die spürbar höhere Performance und Schnelligkeit dieser Prozessoren auch voll ausschöpft.

Gigantische Speichergrößen

Die maximale Speicheradressierung (Nutzungsmöglichkeit von Hauptspeicher) der derzeitigen 32Bit Prozessoren liegt bei 4 GB. 64Bit Prozessoren hingegen ermöglichen einen physischen Speicherplatz von bis zu 1 Terabyte und virtuellen Speicheradressraum von 512 Terabyte. Damit ermöglichen 64-Bit Systeme Computing-Technologien, die bislang auf herkömmlichen PCs auf Grund der zu hohen Rechenzeit nicht realisiert werden konnten. Außerdem verfügen 64-Bit-Computer über größere Caches und eine effizientere Speicheranbindung, was die Geschwindigkeit des Systems weiter erhöht. Ein Beispiel: Ein Computer mit einem AMD Athlon(tm) 64-Prozessor mit 1,8 GHz ist schneller als ein 32-Bit Computer mit einem Pentium(tm) 4 mit 3,2 GHz.

Umstieg leicht gemacht

Die AMD Athlon(tm) 64-Architektur ist für maximale Performance optimiert und unterstützt den x86-64 Befehlssatz. Auf Grund der doppelten Datenbreite auf dem Prozessor profitieren Sie mit dem Athlon(tm) 64 gleichermaßen von Performancegewinnen bei der Ausführung von 32-Bit- und 64-Bit-Programmen. Und das Beste: AMD Athlon(tm) 64 ermöglicht durch die Unterstützung des 32-Bit-Codes den nahtlosen Schritt in die 64-Bit-Welt.

Quelle: [http://www.suse.de/de/private/products/suse\\_linux/prof/64bit.html](http://www.suse.de/de/private/products/suse_linux/prof/64bit.html)

## Bilder

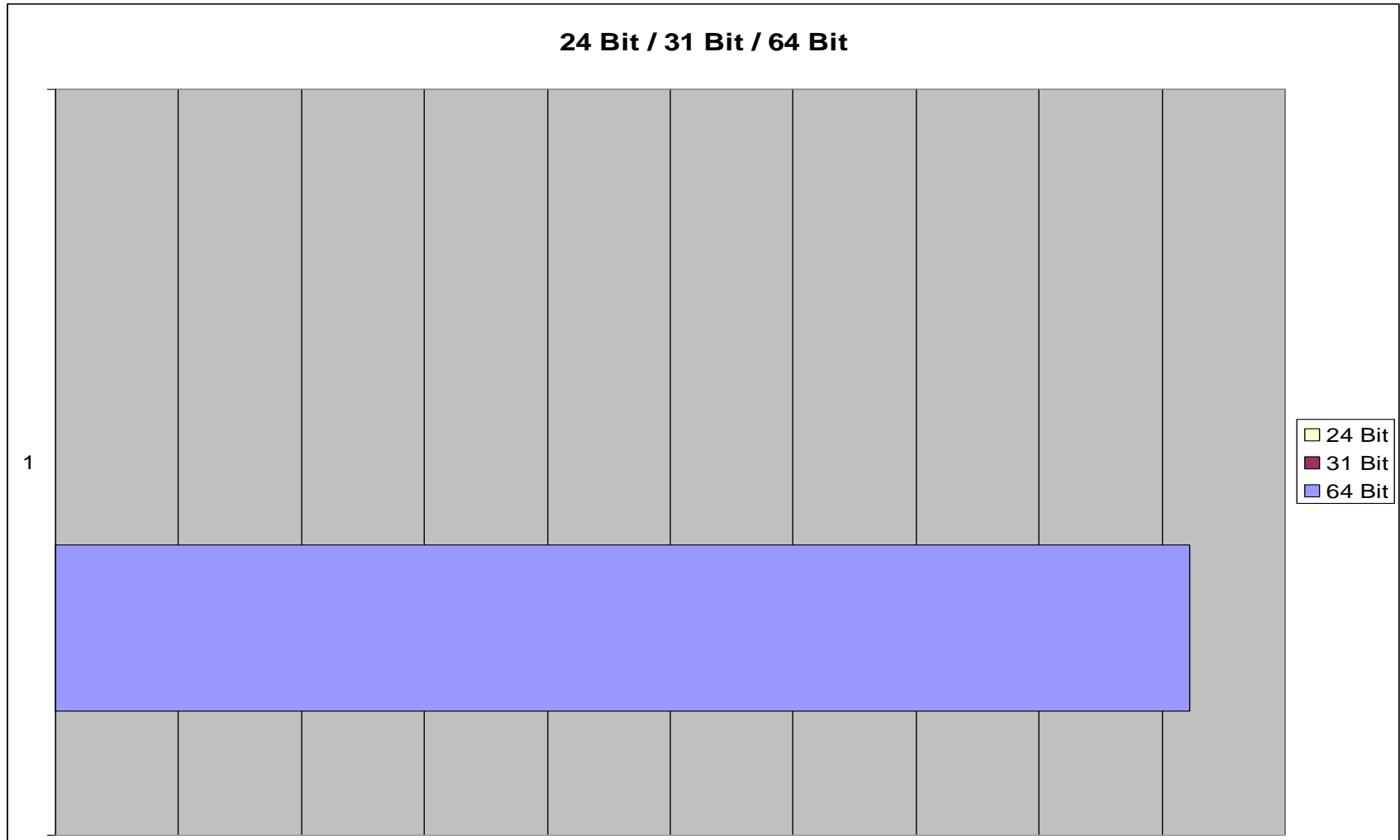
---

- S/360
- Lockkarterkopierer
- Bildschirmarbeitsplatz
- z/Series (z10)





## Adressierungsgrenzen



## Adressierungsgrenzen - Grenzen?

---

### IP Adressierung

Wie in dem vorigem Kapitel bereits erwähnt muss jeder Rechner bzw. Ressource im Netz adressierbar sein. Jeder Rechner erhält eine IP - Adresse. Dies geschieht hierarchisch, d.h. ein Kunde bekommt nötige IP-Adressen von seinem Provider, dieser mietet seine IP-Adressen vom Netzwerk (Carrier), an das er angeschlossen ist, während die Betreiber des Netzwerks ihre IP-Adressen blockweise bei den sogenannten IP Numbering Authorities zeitlich unbefristet "ausleihen".

Diese Adresse besteht aus 32 Bits die byteweise in sogenannten Quads zusammengefasst werden. Der gesamte Adressbereich wird in 4 Klassen aufgeteilt. Neuerdings reichen die IP – Adressen nicht mehr aus, so dass eine 128 Bit lange IP – Adressierung eingeführt wurde. Die alten Adressen können einfach in die neue IP – Adressierung eingebettet werden. Die neue IP – Adressen werden in 8 Quads je 4 hexadezimalen Zahlen angeordnet.

aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh.

Quelle: <http://www.informatik.hu-berlin.de/~haas/adressierung.htm>

### Authentifizierung durch 128-Bit-Schlüssel

eingebaute 1T 128-bit BITBLT Grafikengine

Änderungen von IPv4-Adressierung wächst von 32 auf 128 Bit an

### IBM-Infos:

#### CoreConnect 128-bit Implementation

Related links: [CoreConnect Bus Architecture](#)

The IBM CoreConnect bus architecture eases the integration and reuse of processor, system, and peripheral cores within standard product and custom system-on-a-chip (SOC) designs.

#### Processor Local Bus (128-bit )

Related links: [CoreConnect Bus Architecture](#)

#### Specifications

This book begins with an overview followed by detailed information on 128-bit Processor Local Bus signals, interfaces, timing and operations. This book is for hardware, software, and application developers who need to understand Core+ASIC development and system-on-a-chip (SOC) designs. The audience should understand embedded system design, operating systems, and the principles of computer organization.

Quelle: <http://www.ibm.com>



## Speicherkapazität in der Datenverarbeitung (Wikipedia)

---

### Bit

- 1 Bit - (2 mögliche Zustände), z. B. Ja/Nein
- 5 Bit - ( $2^5 = 32$  mögliche Zustände), z. B. ein Großbuchstabe des lateinischen [Alphabetes](#)
- 7 Bit - ( $2^7 = 128$  mögliche Zustände), z. B. ein Zeichen im [ASCII](#)-Zeichensatz

### Oktett (8 Bit, 1Byte)

- 1 Oktett - ( $2^8 = 256$  mögliche Zustände) ein Schriftzeichen (erweitertes lateinisches [Alphabet](#))
- 2 Oktetts - ( $2^{16} = 65536$  mögliche Zustände) ein Schriftzeichen im [Unicode](#)-Format
- 4 Oktetts - ( $2^{32} =$  etwa 4,3 Milliarden mögliche Zustände)

### Kilobyte ( $2^{10} = 1.024$ Bytes ca. $10^3$ Bytes)

- 0,5 KB Eine Buchseite als Text
- 1440 Kilobytes - eine High Density 3,5 Zoll [Diskette](#)

### Megabyte ( $2^{20} = 1.048.576$ Bytes ca. $10^6$ Bytes)

- 5 MB Die [Bibel](#) als Text
- 650 bis 700 Megabyte - eine [CD-ROM](#)

### Gigabyte ( $2^{30} = 1.073.741.824$ Bytes ca. $10^9$ Bytes)

- 5 GB - Ein komprimierter Spielfilm

### Terabyte ( $2^{40} = 1.099.511.627.776$ Bytes ca. $10^{12}$ Bytes)

- 20 TB Textumfang der Bestände der [Library of Congress](#) mit rund 20 Millionen Büchern (1963 – heute ca. 80 TB liegen)

### Petabyte ( $2^{50} = 1.125.899.906.842.624$ Bytes ca. $10^{15}$ Bytes)

- Die Speicherkapazitäten der weltweit größten Rechenzentren lagen Ende 2002 bei 1 bis 10 Petabyte

### Exabyte ( $2^{60} = 1.152.921.504.606.846.976$ Bytes ca. $10^{18}$ Bytes)

- Die Gesamtheit aller gedruckten Werke wird auf 0,2 Exabyte geschätzt

### Zettabyte ( $2^{70} = 1.180.591.620.717.411.303.424$ Bytes ca. $10^{21}$ Bytes)

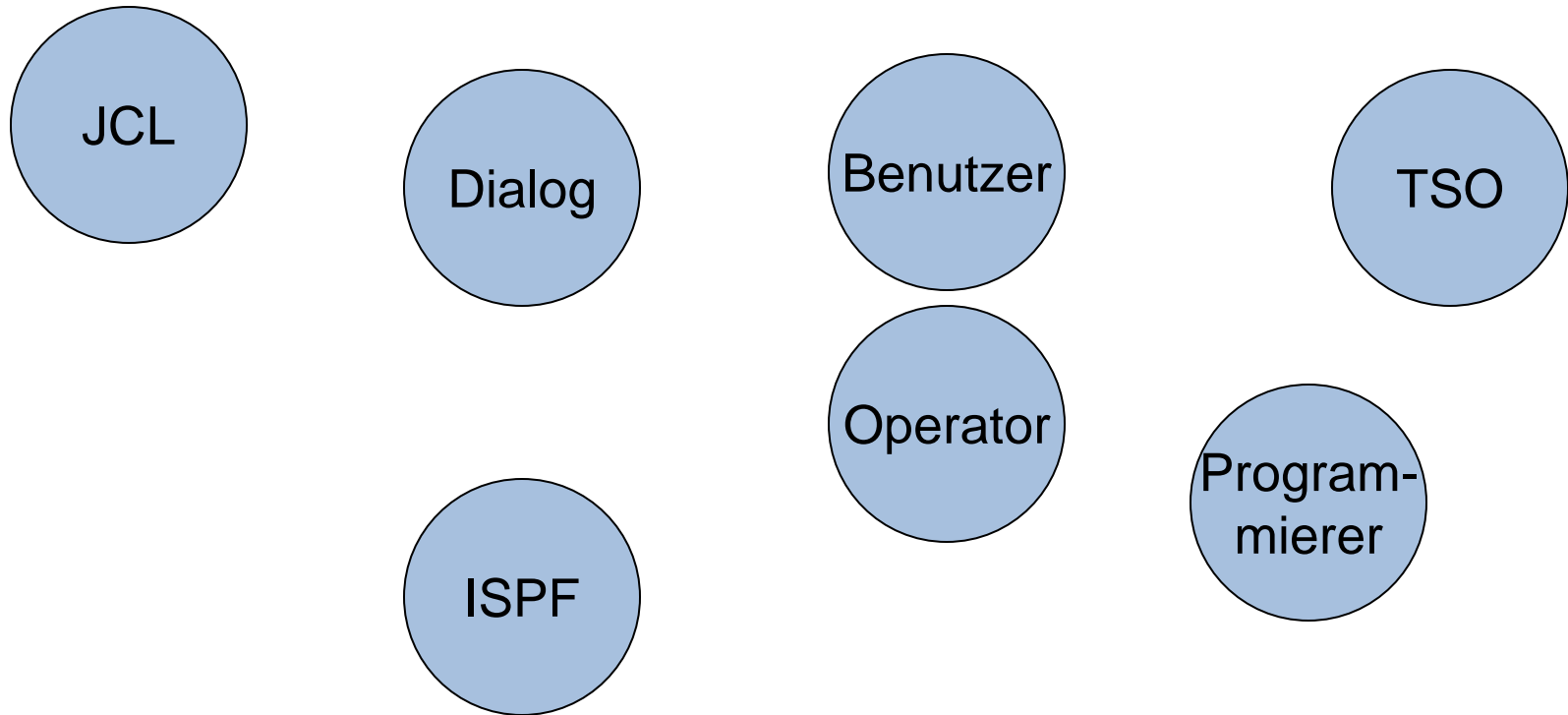
### Yottabyte ( $2^{80} = 1.208.925.819.614.629.174.706.176$ Bytes ca. $10^{24}$ Bytes)

- 1 Yottabyte sind etwa doppelt so viele Bytes, wie Teilchen in einem [Mol](#) vorhanden sind (zum Beispiel sind 12 [Gramm Kohlenstoff](#)  $^{12}\text{C}$  genau 1 Mol)

- 
- Einführung
  - Entwicklung des Mainframes
  - • Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

## Begriffe

---



## Kommunikationsebenen

---

- Steuerung des BS durch Operator
- Beschreibung von Jobs über JCL
- Dialog zwischen Anwender und System

## Operatorkommandos

---

- Das Betriebssystem regelt alles. Woher weiß es, was es tun soll?
- Also sind Fragen zu beantworten:
  - Woher weiß das BS, welches Programm ausgeführt werden soll?
  - Was soll geschehen, wenn ein Programm loopt?
  - Woher weiß ein BS, welche Daten auf welcher Peripherie in dem Programm benötigt werden?
- Kommandosprache für Operator

## Job Control

---

- ~~Hunderte~~ Millionen von Programmen müssen organisiert und kontrolliert ablaufen.
- Job Control
  - JOB-Anweisung mit allgemeinen Definitionen
  - EXEC-Anweisung mit Programm
  - DD-Anweisung mit Datengeräteinformationen
  - siehe Kapitel “Job Management”



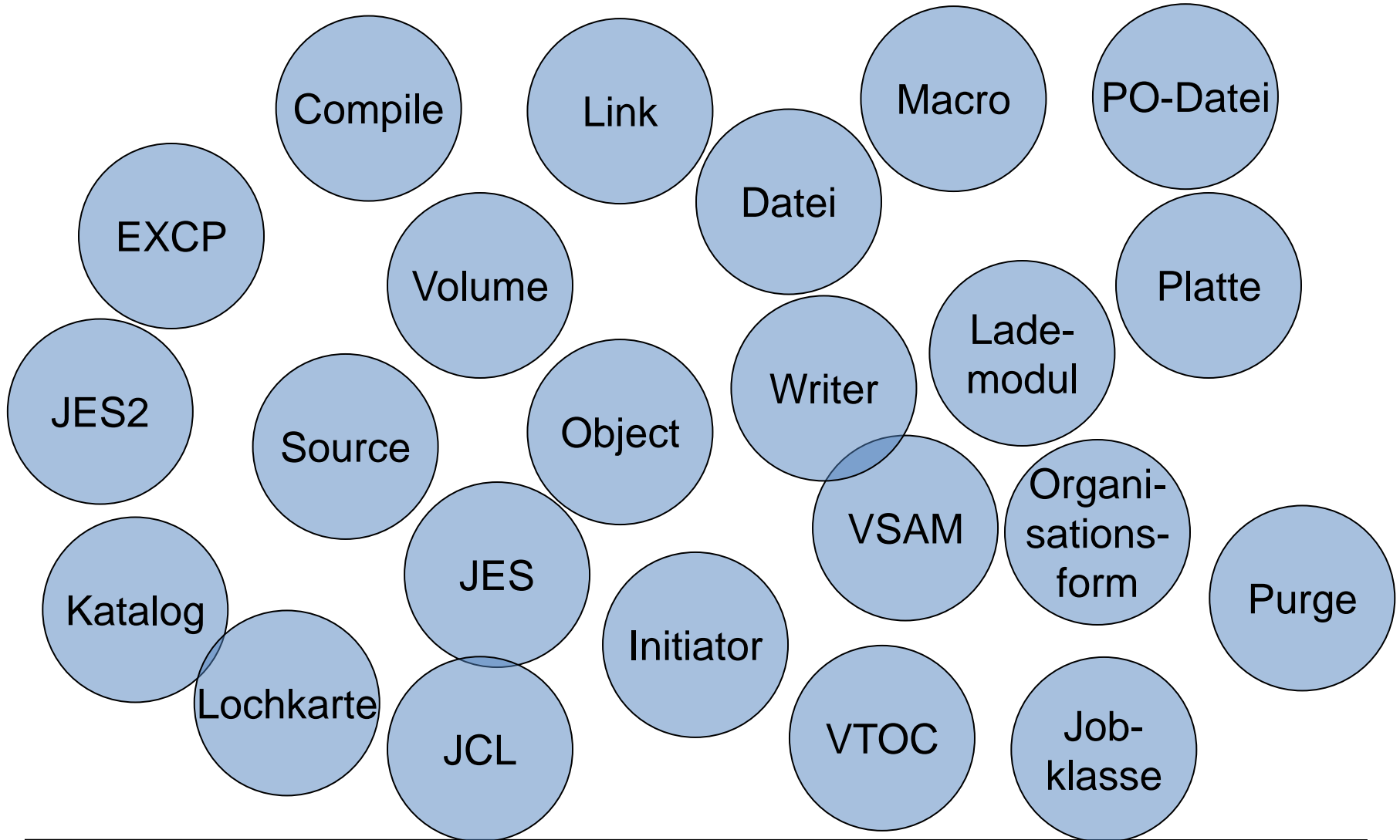
- TSO (Time Sharing Option)
  - Dateiverwaltung und Dateipflege
  - Erstellen von Programmen
  - Ausführen kleinerer Programme
  - Ausführen von Kommandoprozeduren
  - siehe Kapitel “TSO und ISPF”



- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - ➔ • Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

## Begriffe

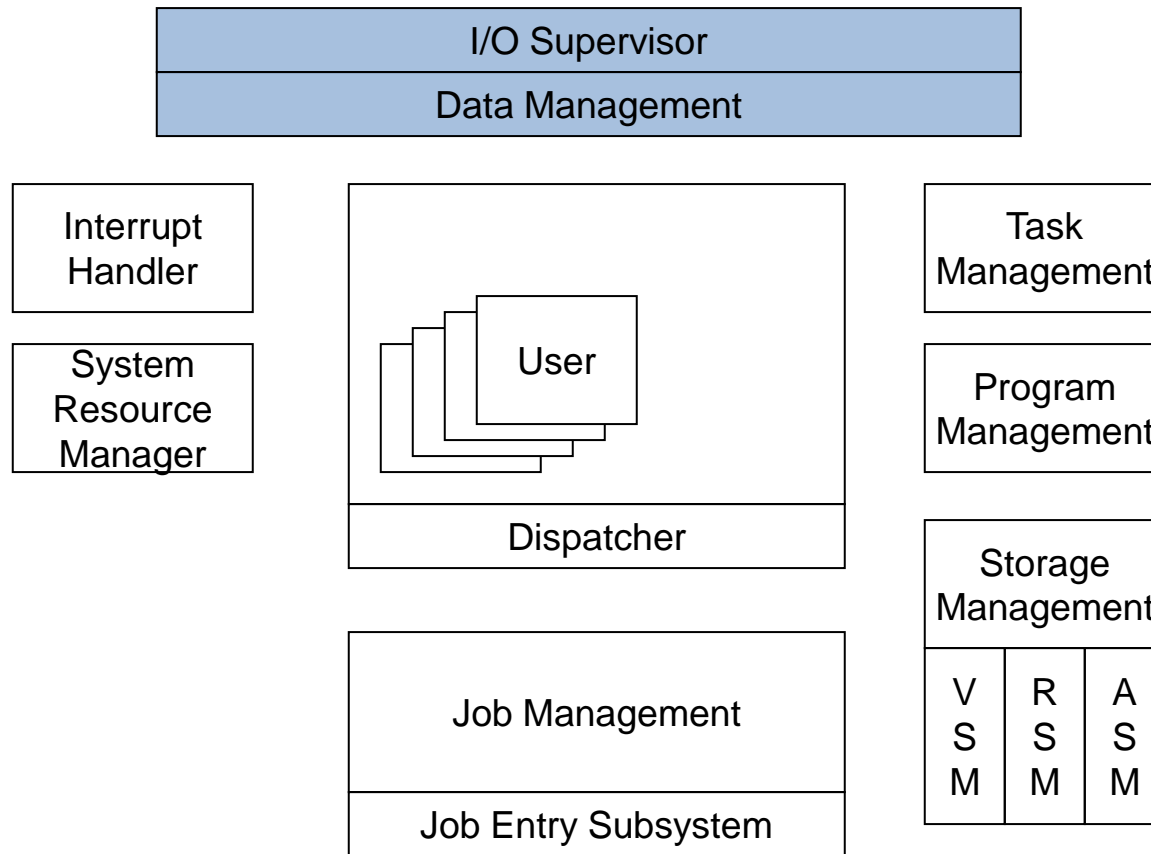
---



# Data Management

## z/OS Komponenten

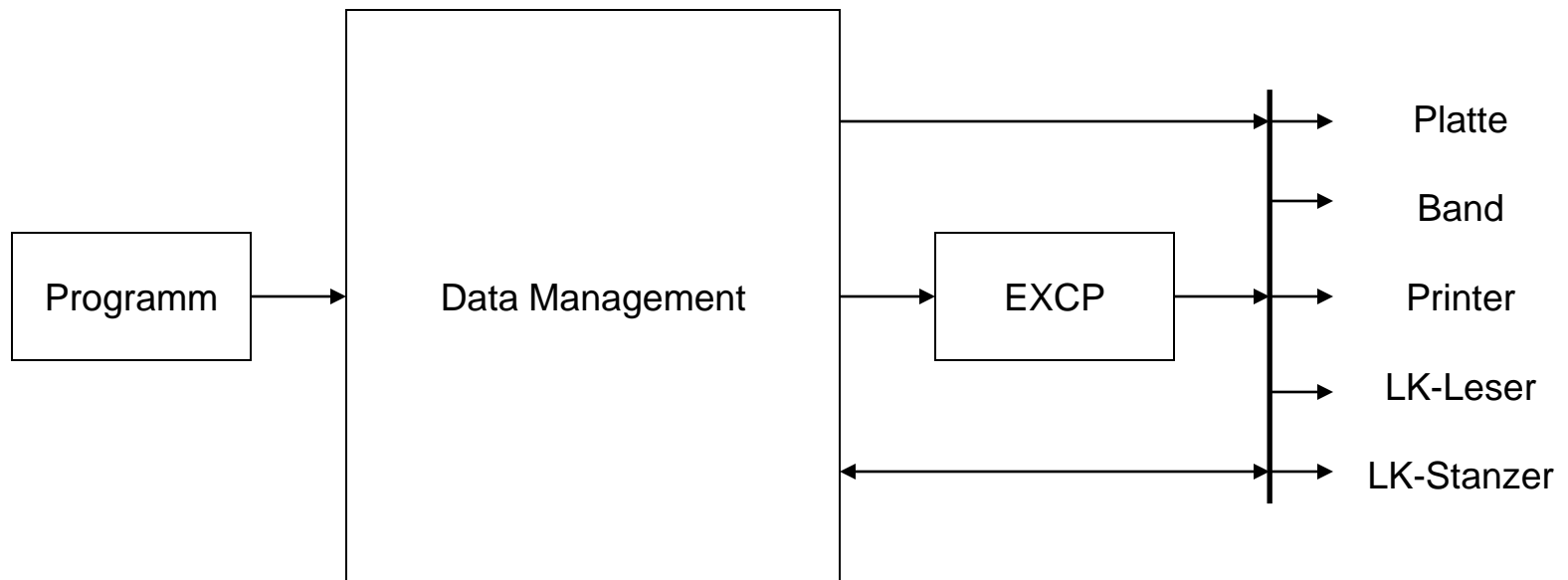
---



## Aufgaben und Funktionen

---

- steuern und überwachen der Ein/Ausgaben
- verwalten von Speicherplatz
- Katalogverwaltung für Dateien
- Verwaltung von Datenträgern
- Schutz der Dateien
- Bindung zwischen Programm und Datei
  - Puffer, Blockung, Gerätetypen, Geräteadressen müssen nicht bei Programmierung bekannt sein



## Aufgabe der Steuereinheit (EXCP) – 1

---

- E/A - Kommandos (CCW) ausführen, z.B.
  - SEEK
  - SEARCH
  - READ
  - WRITE
- Command Chanining
- Fehlerkorrektur (permanente Fehler sind normal)
- E/A – Befehlswiederholung

## Aufgabe der Steuereinheit (EXCP) – 2

---

- Statusinformation sammeln und an Zentraleinheit weitergeben
- Unterbrechungssignale erzeugen und an Zentraleinheit weitergeben (CEDE)
- Eine von mehreren Festplatten selektieren
- Cache - Non Volatile Cache
- RAID (Redundant Array of Independent Disks)



## Aufgabe der Festplattenelektronik

---

- Umsetzen der magnetischen Lese / Schreibsignale in Folgen von Bits ( R / W Channel)
- Spuranfangssignal
- Steuerung des Zugriffsmechanismus
- Lese / Schreibkopf selektieren (Plattenoberfläche)
- Fehler Erkennung (Syndrom Checking, Syndrom = 5 - 6 Bytes)
- Status setzen

ES war einmal ;)

- Eigenschaften
  - Satzformat: Record Format F, V, U
  - Blockung: ja oder nein
  - Satzlänge: LRECL
  - Blocklänge: BLKSIZE
- Identifizierung
  - Dateiname: DSN
  - Datenträger
  - physische Position auf dem Datenträger

## Datenträger

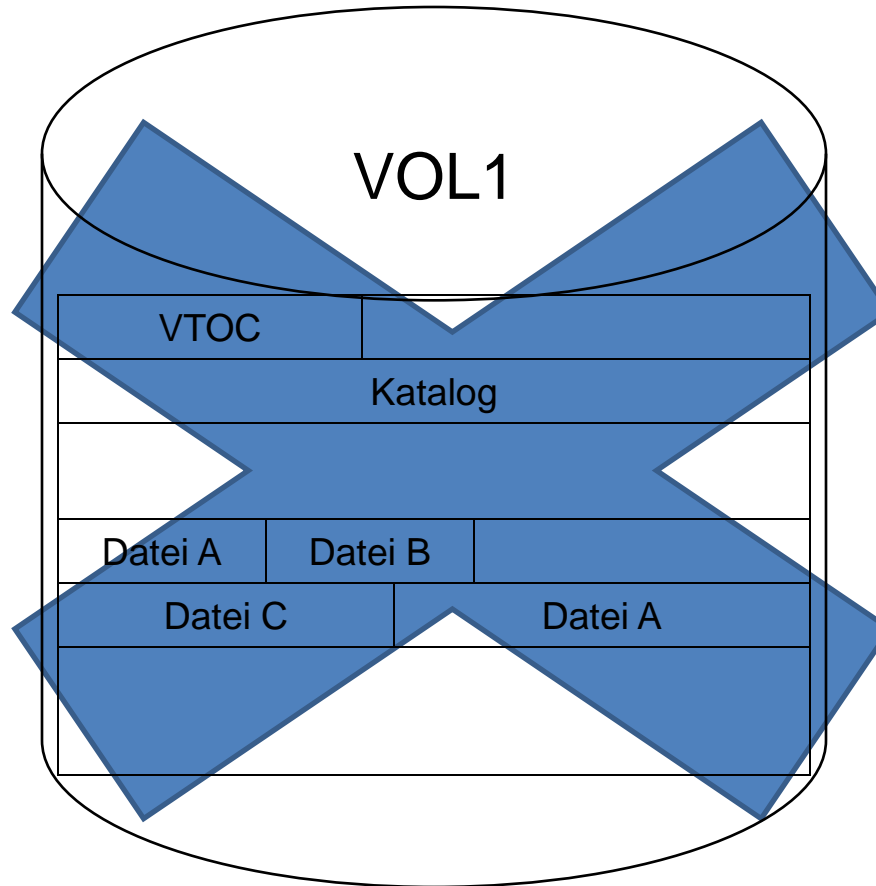
---

- Magnetplatten
- optische Platten, magneto-optische Platten
- Magnetbänder, Magnetkassetten
- Papier
- anno dunnemals:
  - Lochkarten, Lochstreifen
- Verarbeitungsarten
  - sequentiell
  - direkt

- Volume
  - Magnetplatten
  - optische Datenträger
  - benötigen Verwaltungsinformationen
  - VOLSER ist auf VOL1-Kennsatz
  - trägt Eigenschaften jeder Datei auf ihr
    - DSN, RECFM etc. im sog. HDR1-Satz
  - VTOC (Volume Table of Contents)
    - Informationen des HDR1-Satzes und physischer Platz
    - Information über freien Platz

## Struktur einer Magnetplatte

---



## Organisationsformen von Dateien

---

- Sequentielle Dateien
  - Speicherung “überall”
- Direct Access Dateien
  - mit physikalischer Adresse wie Zylinder, Spur, Satznummer gespeichert
- indexsequentielle Dateien
- relative Dateien
- Bibliotheken oder PO-Datei

## Zugriffsmethoden

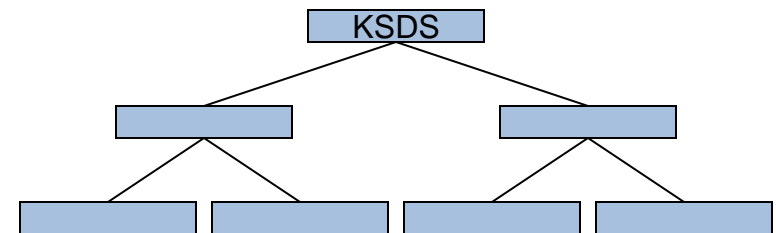
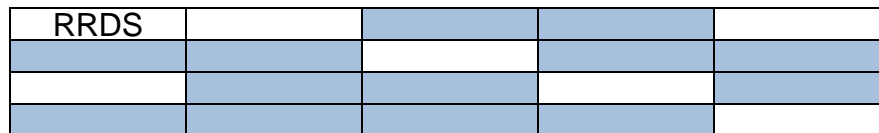
---

- Pro Dateiform eine Methode
    - Umwandlung von Anforderung eines Programms in EXCP-Instruktionen
  - Dateiformen
    - BSAM oder QSAM
    - BDAM
    - BISAM oder QISAM
    - VSAM
- (“B” für blockweise Verarbeitung)

## Zugriffsmethode VSAM

---

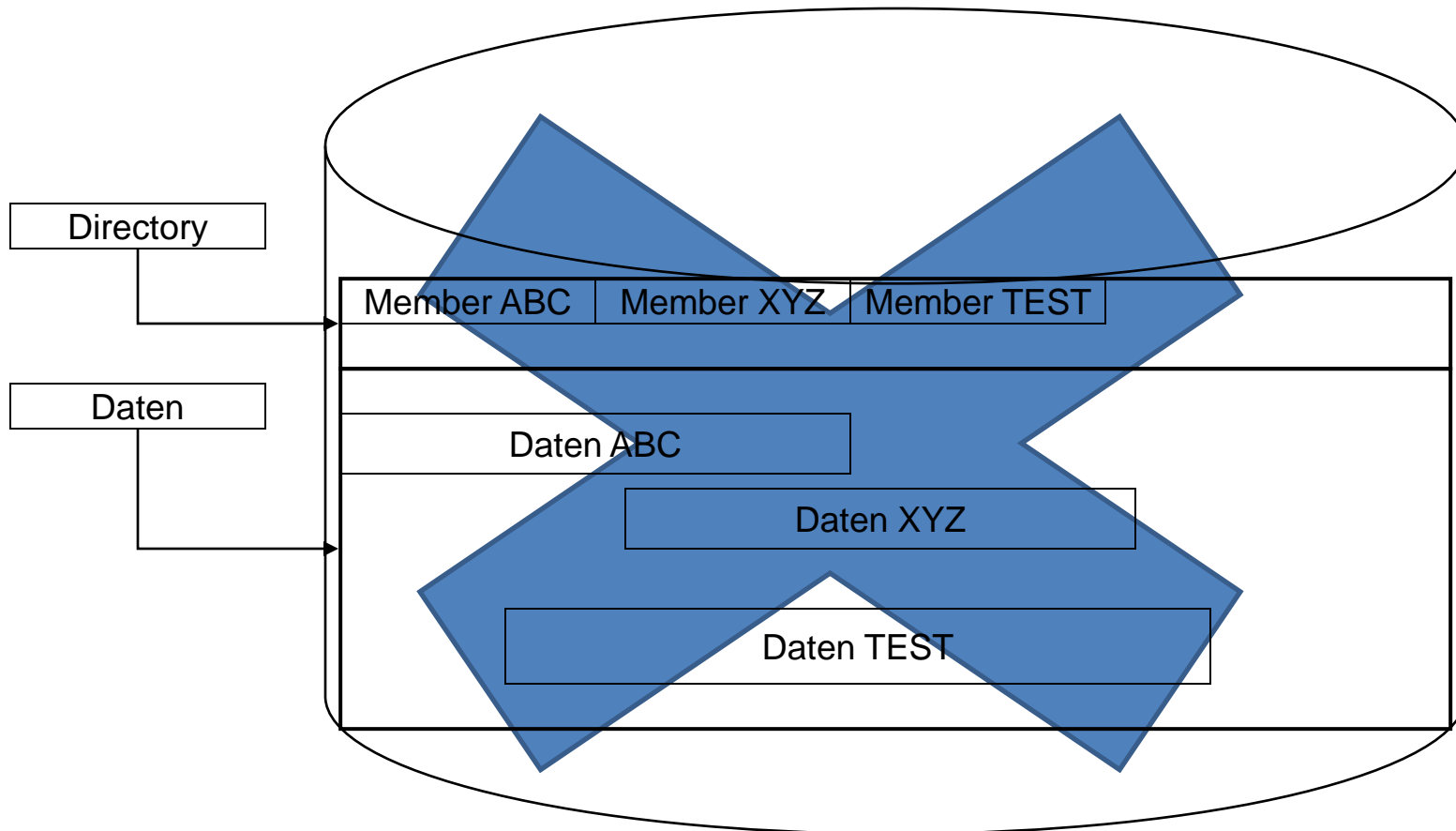
- ESDS - Entry Sequenced Data Set
- KSDS - Key Sequenced Data Set
- RRDS - Relative Record Data Set
- LDS - Linear Data Set





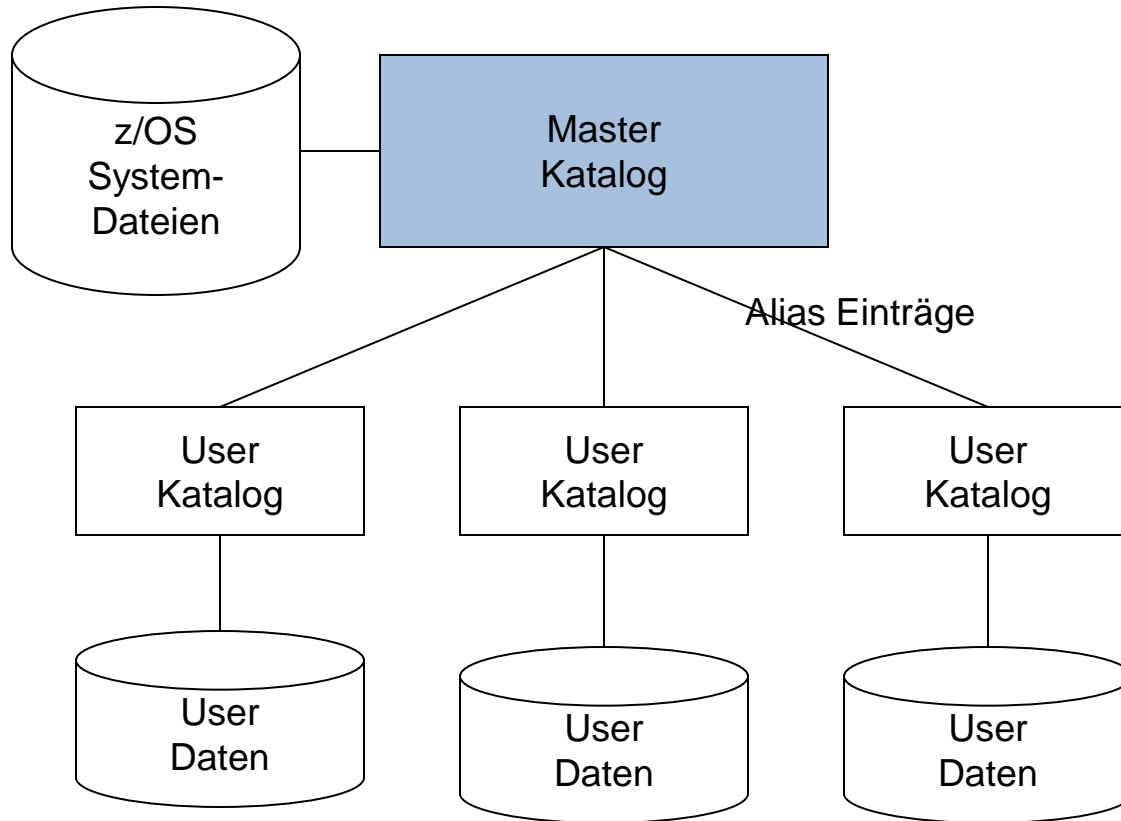
## Struktur einer PO-Datei

---



## Katalogstruktur im z/OS

---



## Dienstprogramme für Dateien

---

- IEBCOPY
- IEBGENER
- IEHLIST
- IDCAMS
- ICEMAN

## Vergleich Unix – z/OS in Auswahl

---

- Unix
  - Dateien sind strukturlose Zeichenketten
  - Zugriffsmethode READ (fileid, buffer, length)
    - Ende des Buffers zeigt auf nächstes Offset
    - feste Blocklänge (fixed block architecture)
  - Zugriffsmethode raw für Datenbankanwendungen
- z/OS
  - Satz orientierter Zugriff
  - Zugriffsmethode GET (recordid, buffer)
    - Teil des BS
    - sequentiell, indiziert, random
    - EXCP mit exakter direkter physischer Adressierung

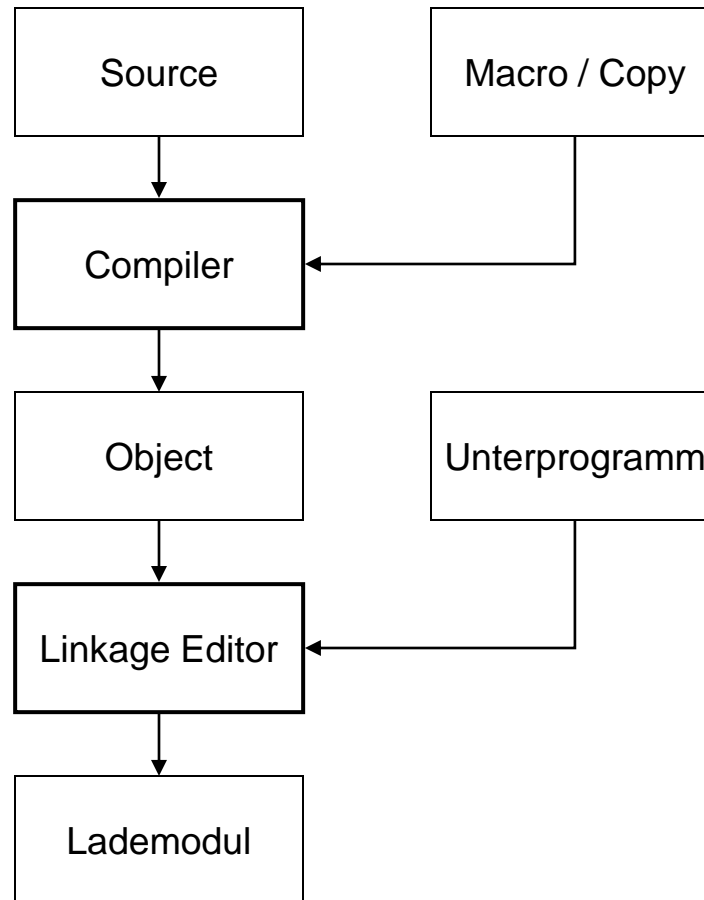


## Vergleich Unix – z/OS - Fazit

---

- schnellste Zugriffe in Größenordnungen
  - z/OS mit mehreren 10.000 E/A-Zugriffe pro Sekunde
  - Unix mit mehreren 100 E/A-Zugriffe pro Sekunde
- Was ist „besser“?



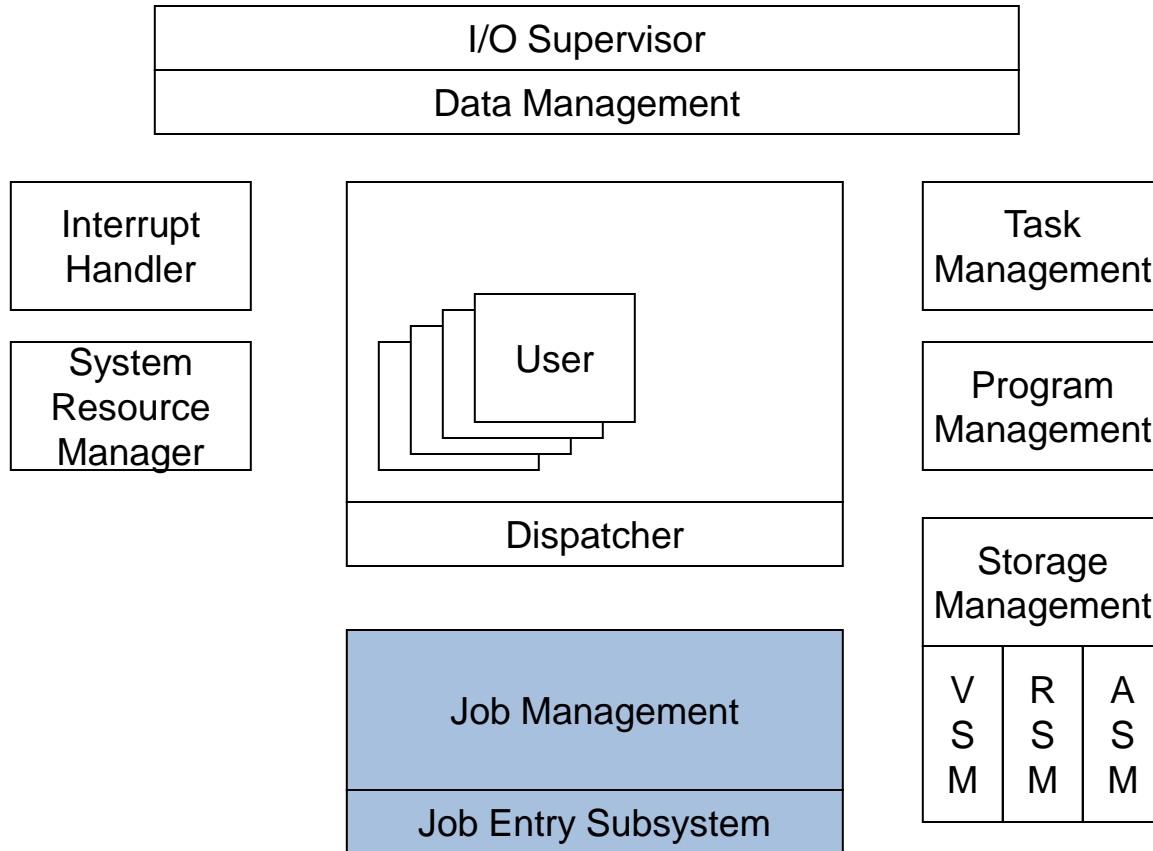


- alles ist unterstützt mit Tools wie
  - SCM: SCLM (IBM) / Endevor (CA)
  - IDz (IBM) / M4Workbench (Macro4) / Workbench (Compuware)
  - Debugger
  - File-Zugriffe
  - DB-Zugriffe
  - Abend Tools
  - etc.
- viele Produkte basieren auf Eclipse bzw. gibt es als Eclipse Plugins

# Job Management

## z/OS Komponenten

---





## Definition

---

- Ein Job ist ein Auftrag an das Betriebssystem, einen oder mehrere Arbeitsschritte (Steps) mit jeweils einem Programm unter Benutzung von einer oder mehreren Dateien abzuarbeiten.
- Batchjob, wenn das mit JCL beschrieben ist
- Achtung! Lochkarten lassen grüßen!

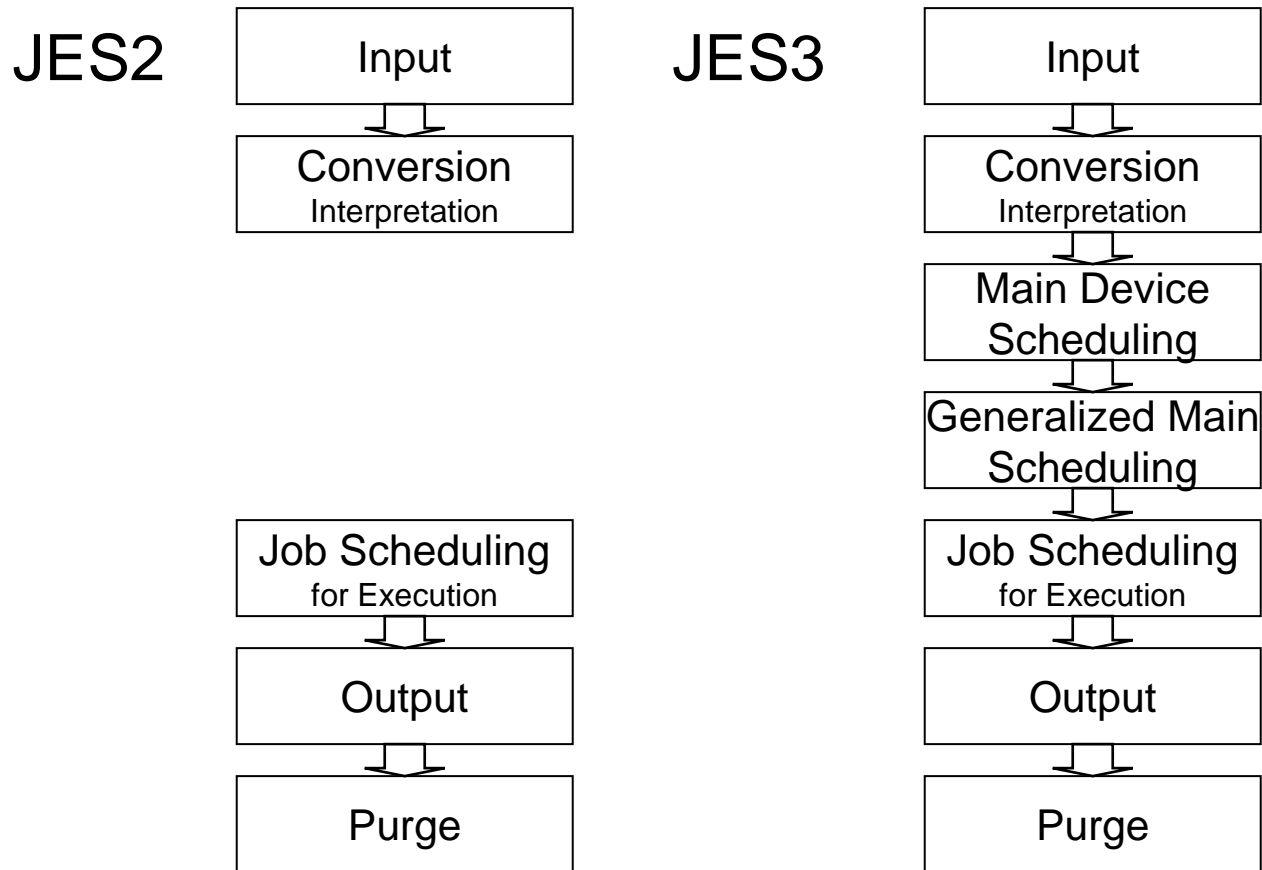
## Funktionen – Batch

---

- JES, Job Entry System, liest JCL ein
- JES interpretiert mit Converter / Interpreter
- Initiator kontrolliert Ausführung der Steps
- Initiator weist (mit Hilfe Allocation / Unallocation Routine) Geräte und Datenträger zu und gibt sie frei

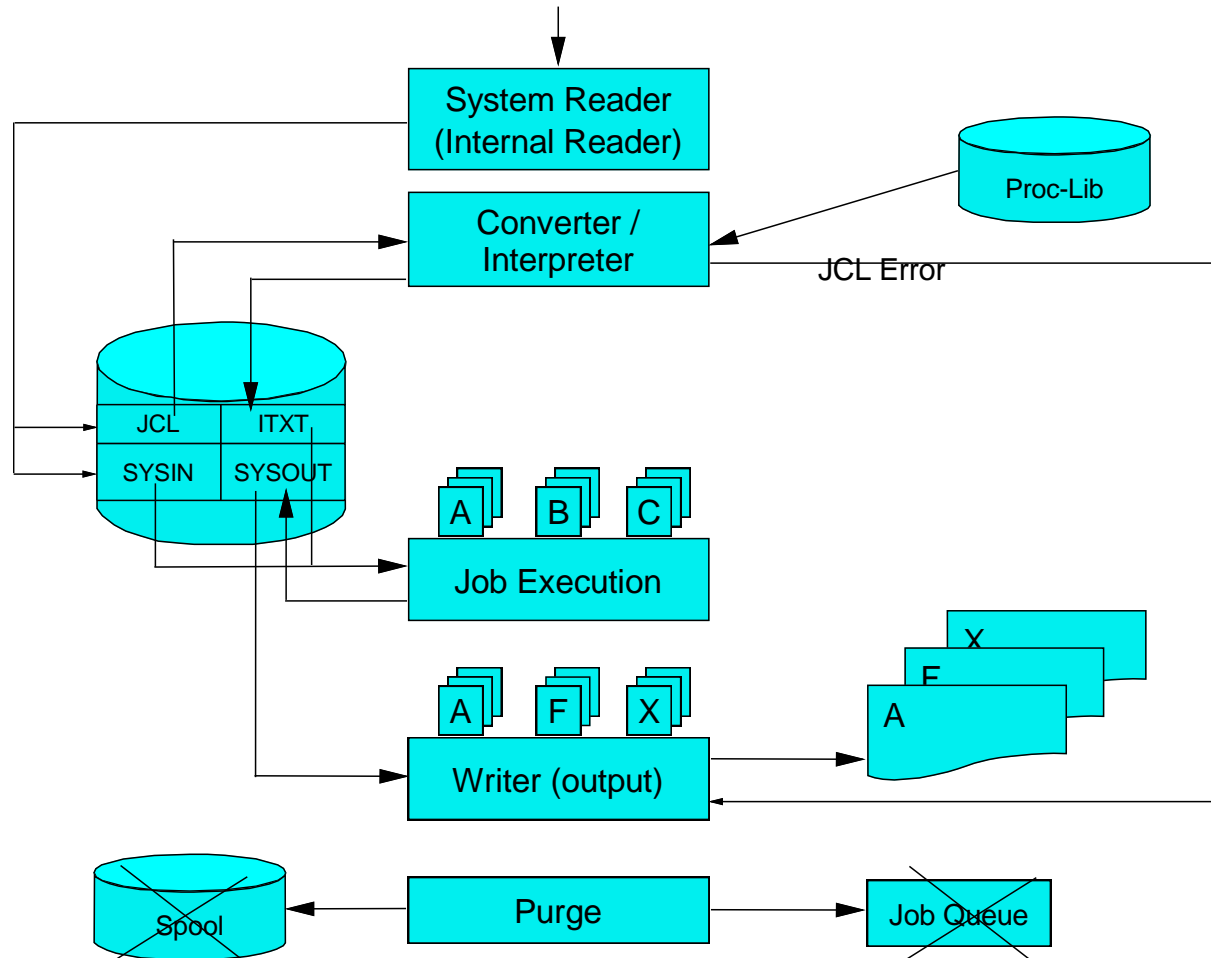
## JES

---



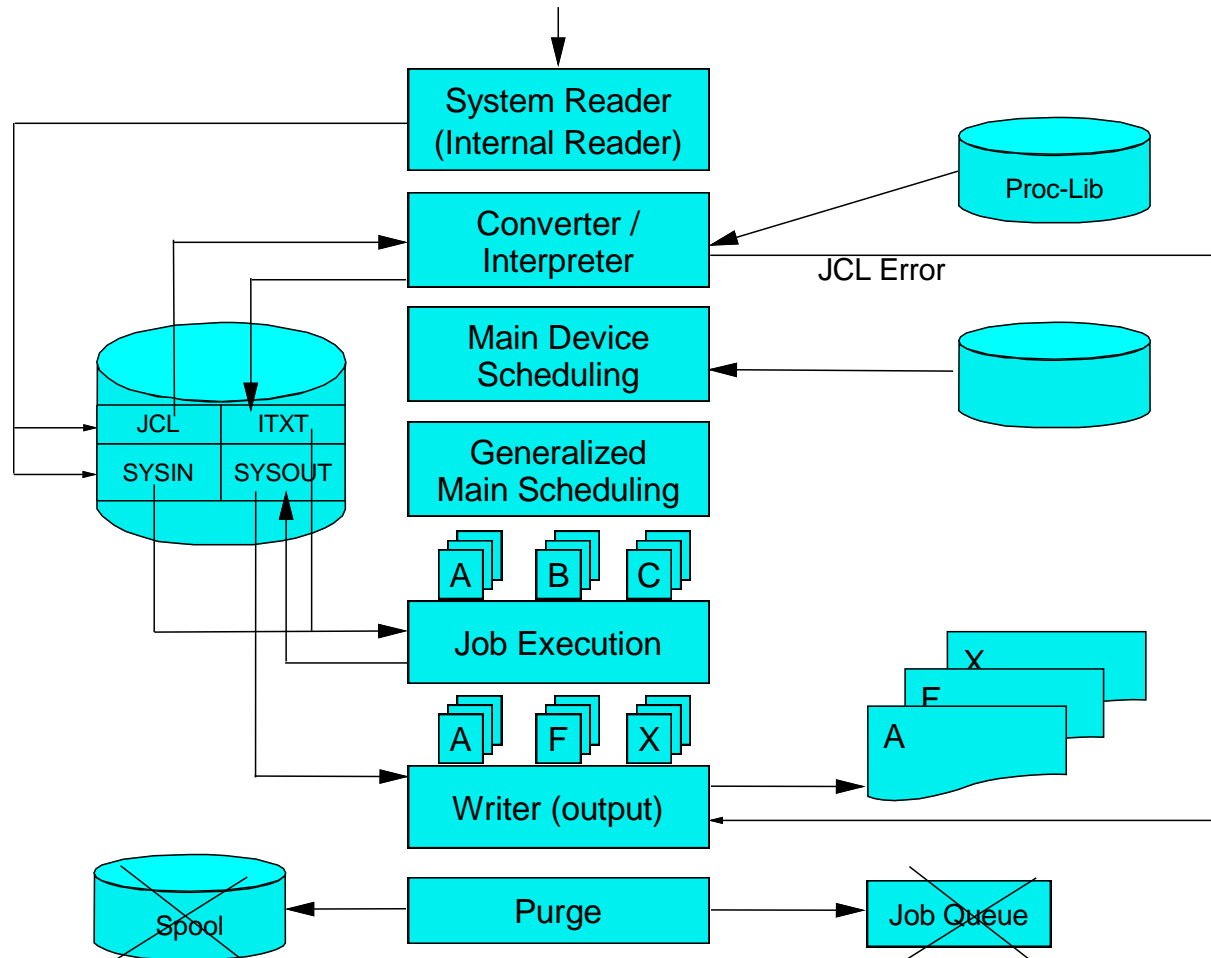
# Job Management

## Ablauf eines Batchjobs – JES2



# Job Management

## Ablauf eines Batchjobs – JES3



## Jobcontrol Language

---

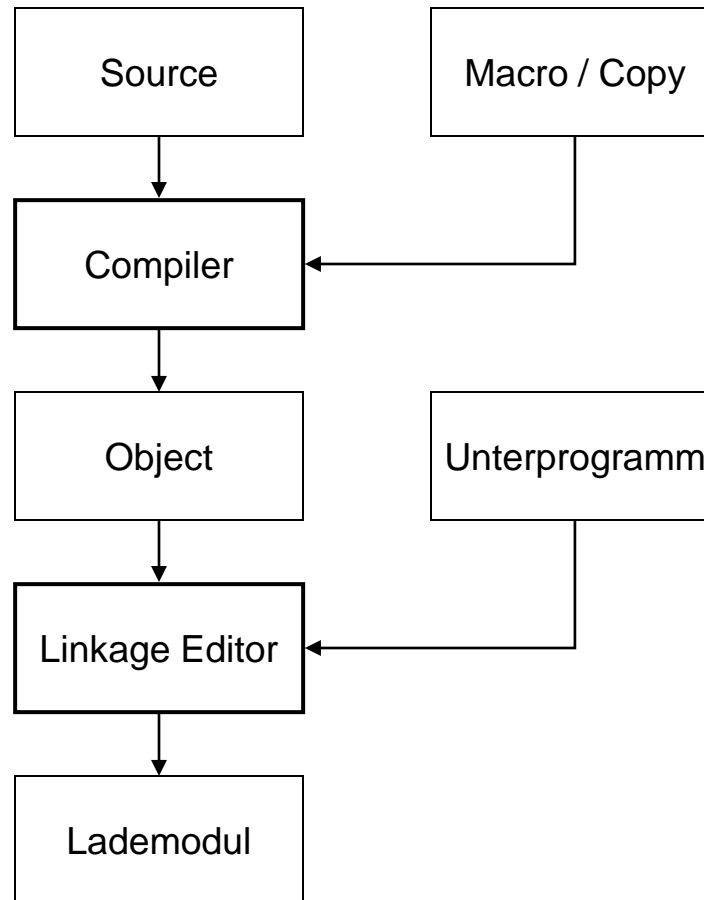
- Es müssen vorhanden sein:
  - für jeden Job 1 JOB Anweisung
  - für jeden Step 1 EXEC Anweisung
  - für jede Datei 1 DD-Anweisung

- Beispiel

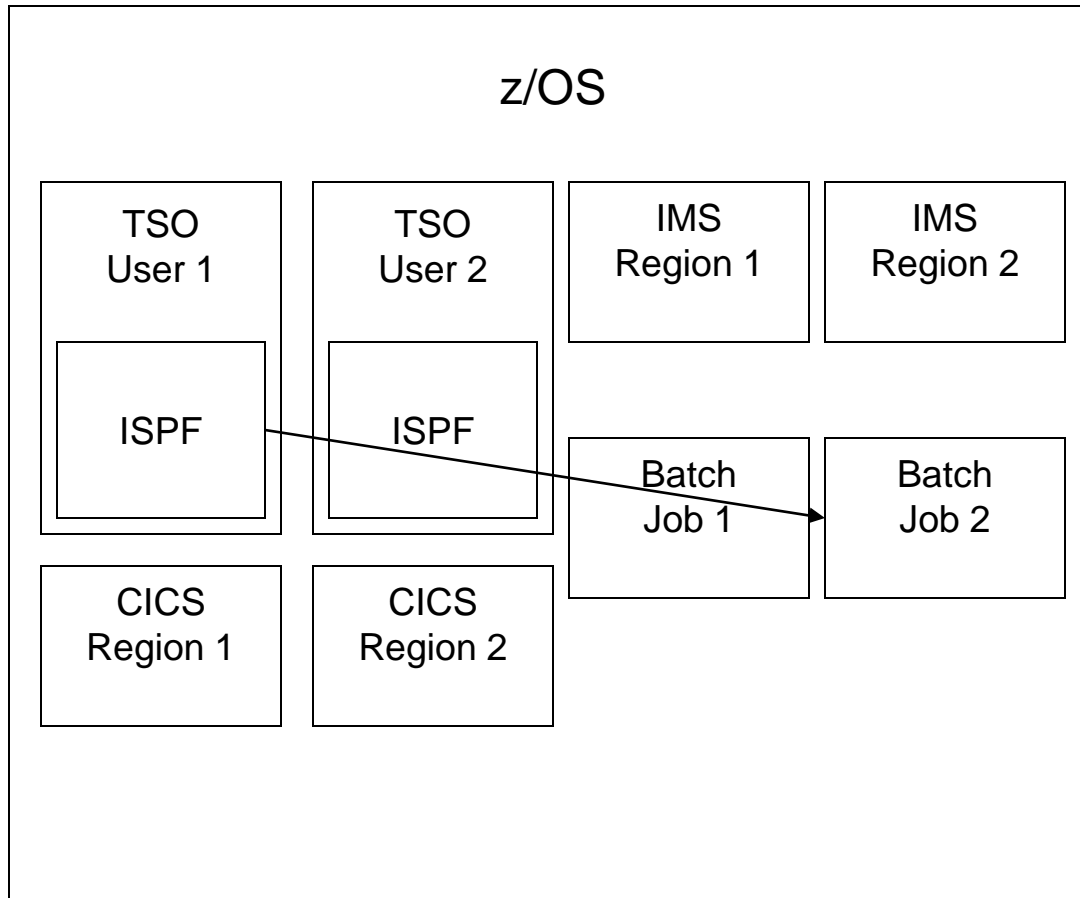
```
//JOBX      JOB   (3SLX510,000,00T NR0003) , . . .
//STEP01    EXEC  PGM=ZINS
//EINGABE   DD   DSN=XV10733.ispf.datei,DISP=SHR
//AUSGABE   DD   DSN=XV10733.ispf.out,DISP=(,CATLG) ,
//          UNIT=SYSDA,SPACE=(TRK,5) ,
//          DCB=(LRECL=80,BLKSIZE=0,RECFM=FB)
//SYSPRINT  DD   SYSOUT=*
```

## Beispiel Compile und Link

---



## z/OS Ausschnitt aus dem Speicher



TSO

Time Sharing Option

ISPF

Interactive System Productivity Facility

CICS

Customer Information Control System

IMS

Information Management System

OS

Operation System

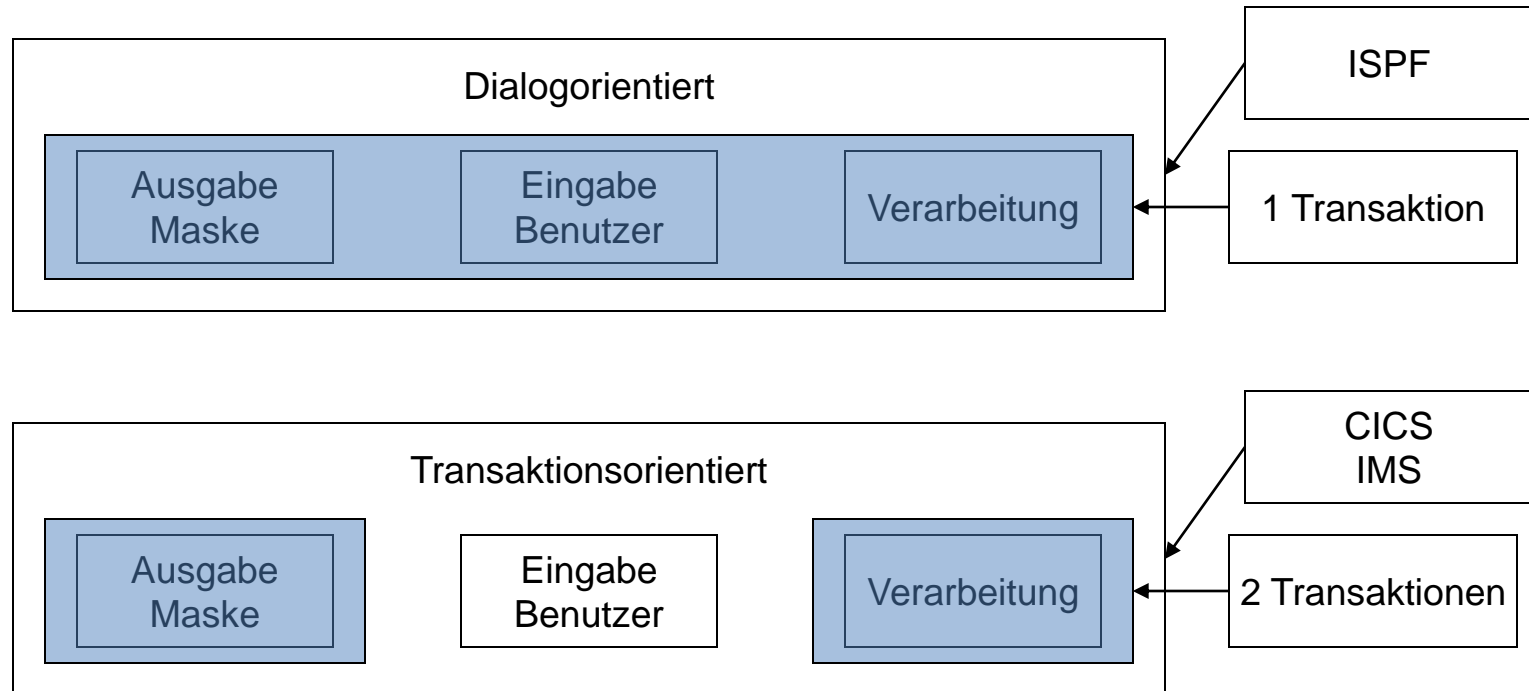


## Wozu ein Dialogsystem?

---

- Batchverarbeitung
  - sequentielle Satzverarbeitung
  - Datenmodifikation
  - Druck
  - Mengen Verarbeitung
- Dialog
  - Einzelverarbeitung
  - sofortige Antwort auf Fragen
  - genaue und aktuelle Informationen

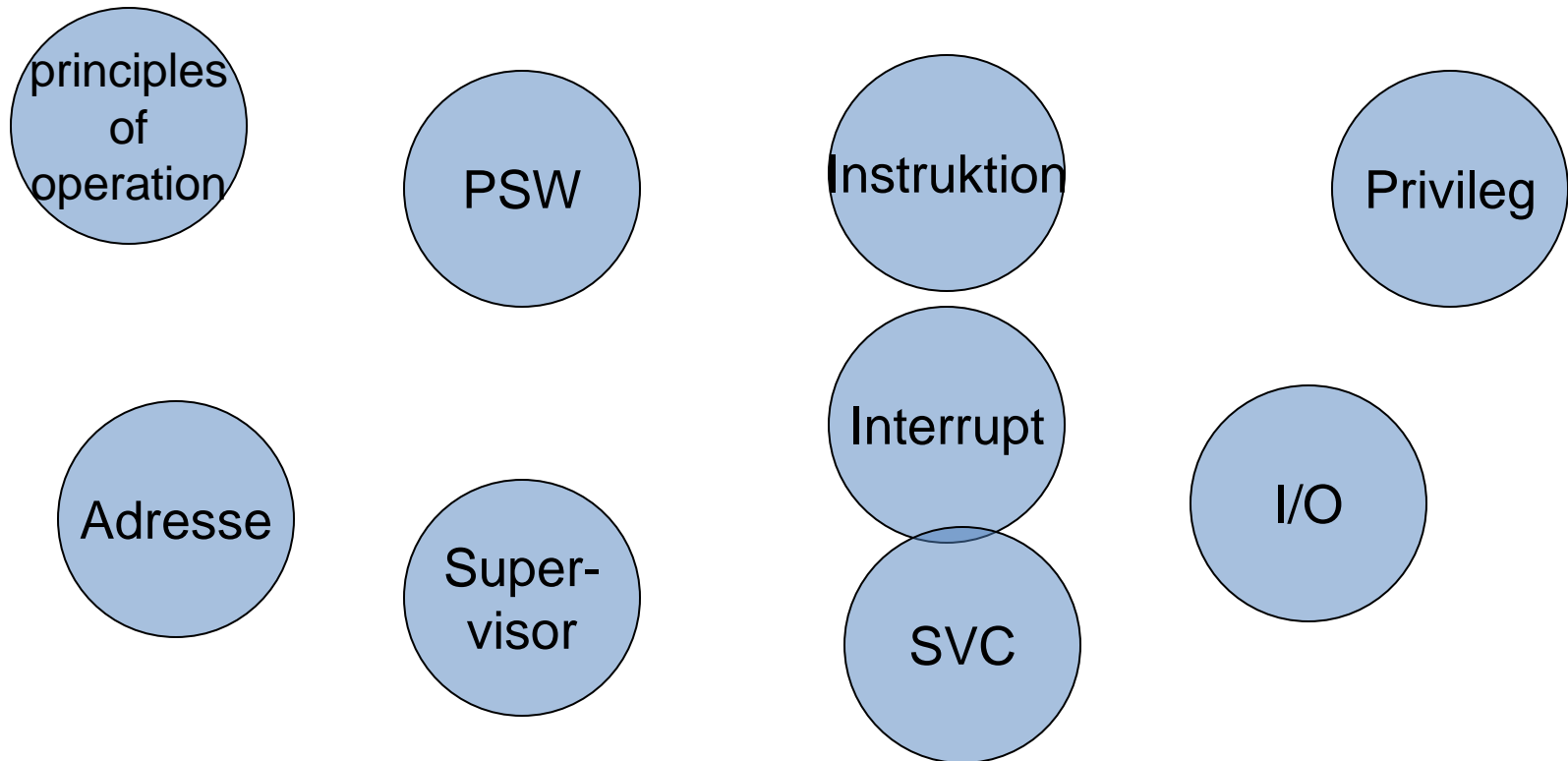
## Arten der Dialogverarbeitung



- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - ➔ • Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

## Begriffe

---

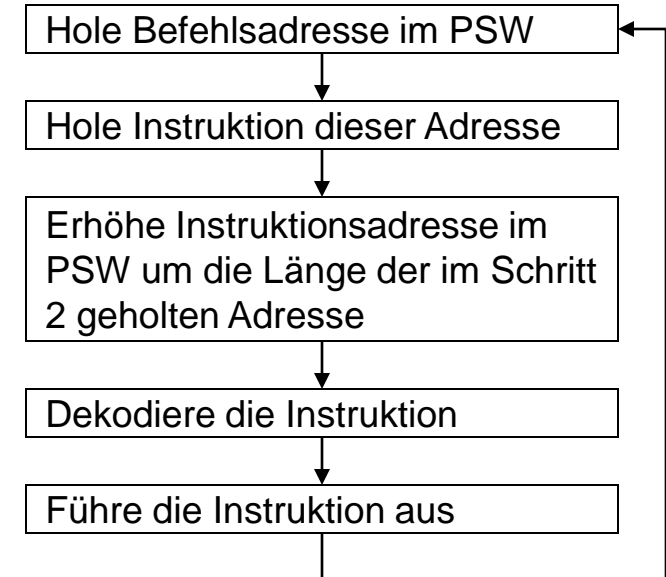


- interrupts
- PSW Program Status Word
  - Doppelwort Register
  - enthält Informationen für Programm Ausführung
    - Befehlszähler
    - Status des aktiven Programms
    - Steuerung Instruktionsfolge
  - Aufbau
    - Statusfelder 40 Bit
    - Befehlsadresse 24 bit / 32 bit / 64 bit / 128 bit

## Principles Of Operation – Beispiel

- Adresse                      Instruktion
- 500                            L 3,X
- 504                            L 4,Y
- 508                            AR 3,4
- 50C                            ST 3,Z

Ablauf



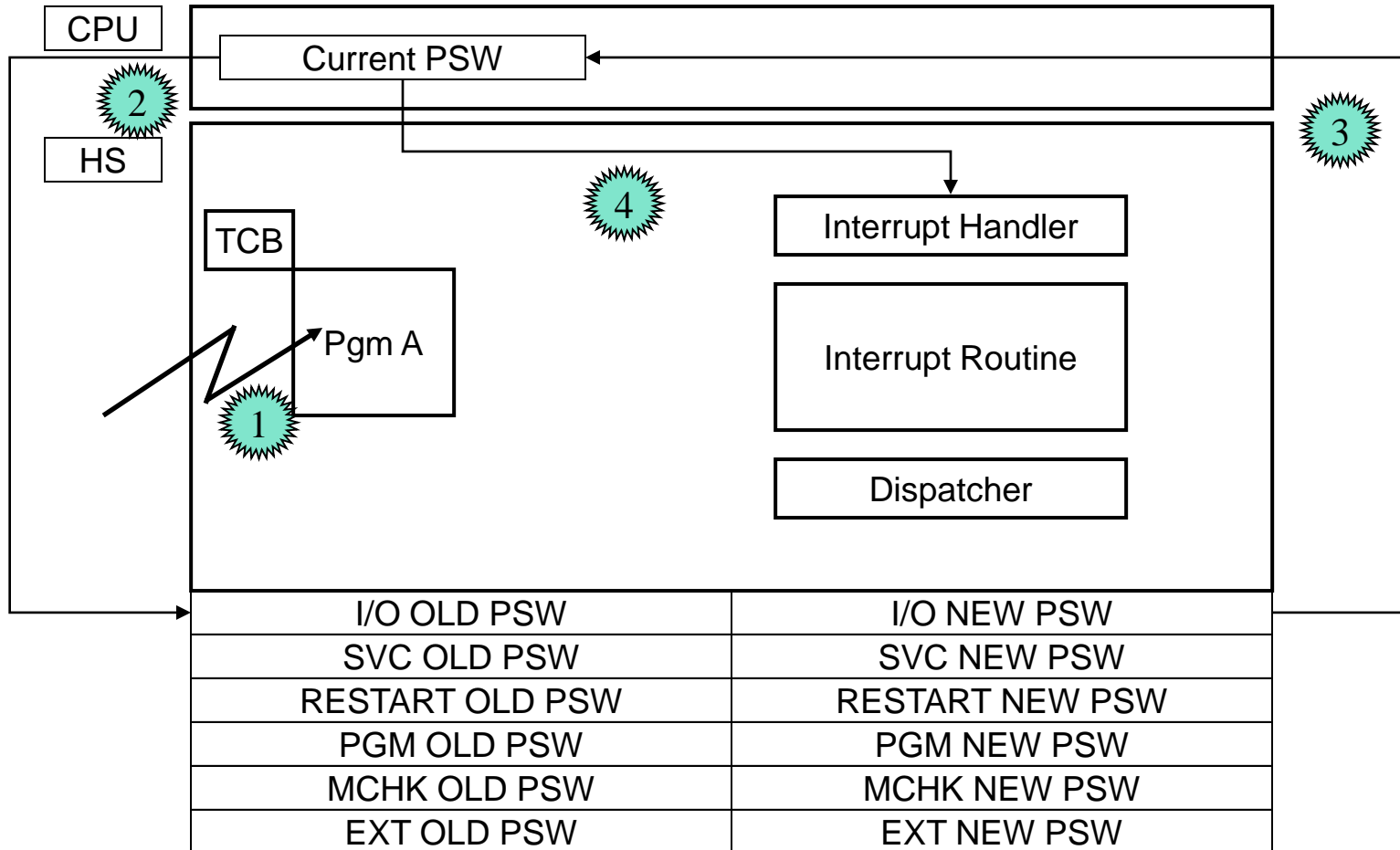
- reagieren auf Anforderungen des Systems
- Kontrolle geht automatisch an FLIH
  - First Level Interrupt Handler
  - je Interrupt Typ ein FLIH
  - eventuell Weitergabe an SLIH
- Synchroner Interrupts von Programm selbst
- Asynchrone Interrupts von “außen”

- I/O Interrupt
- SVC, Supervisor Call, Interrupt
- External Interrupt
- Operator, andere CPU, Timer
- Program Interrupt
- Fehler, führt zum Abend
- Machine Check Interrupt
- Hardwarefehler
- Restart Interrupt



# Funktionen des Betriebssystems

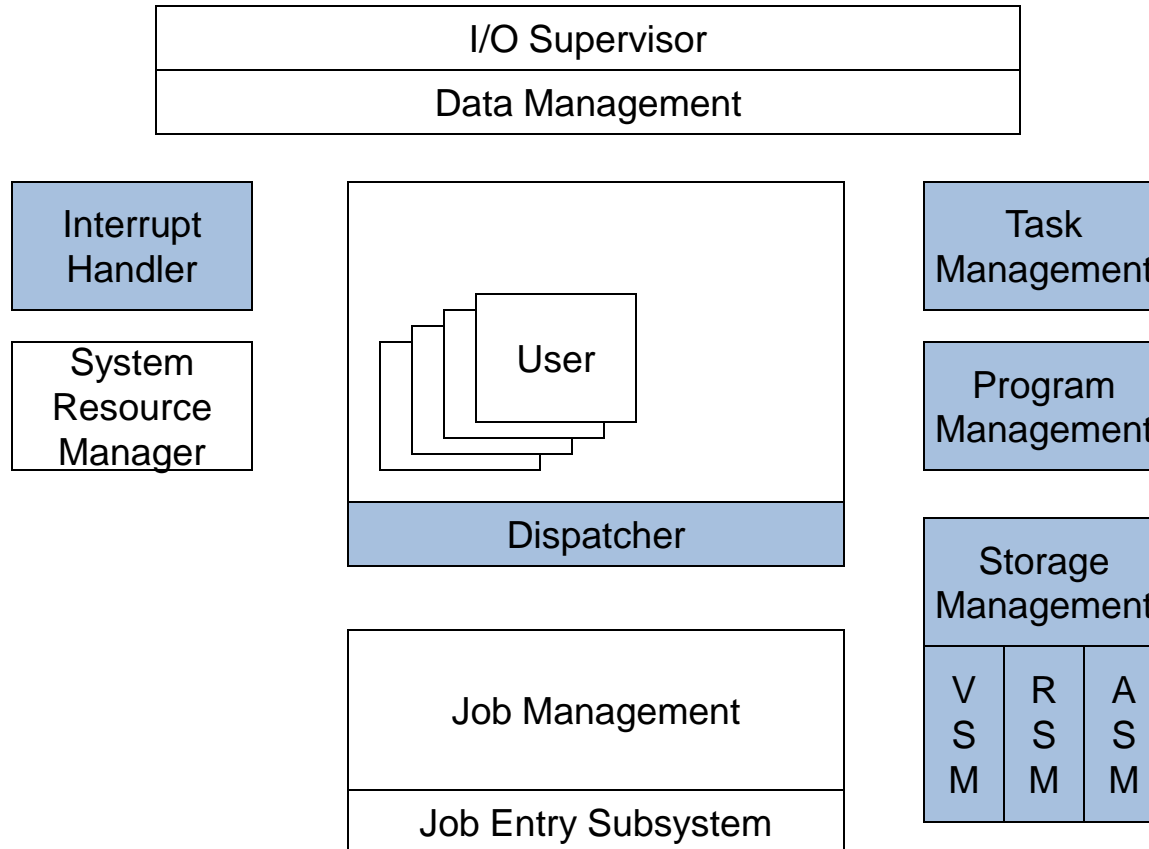
## Principles Of Operation - Ablauf eines Interrupts



- Interrupt während Interrupt arbeitet
  - mal erlaubt, mal nicht
- System Masken im PSW (Bit Masken)
  - I/O Interrupt darf keinen 2. nach sich ziehen (Bit 6)
  - externe Interrupt darf keinen 2. nach sich ziehen (Bit 7)
  - Machine Check Interrupt lässt gar nichts mehr zu
- sog. CMWP-Feld
  - Bit 12 BC-Mode oder EC-Mode
  - Bit 14 Zustand ready oder wait
  - Bit 15 zeigt supervisor state an

# Funktionen des Betriebssystems

## z/OS Komponenten - Supervisor



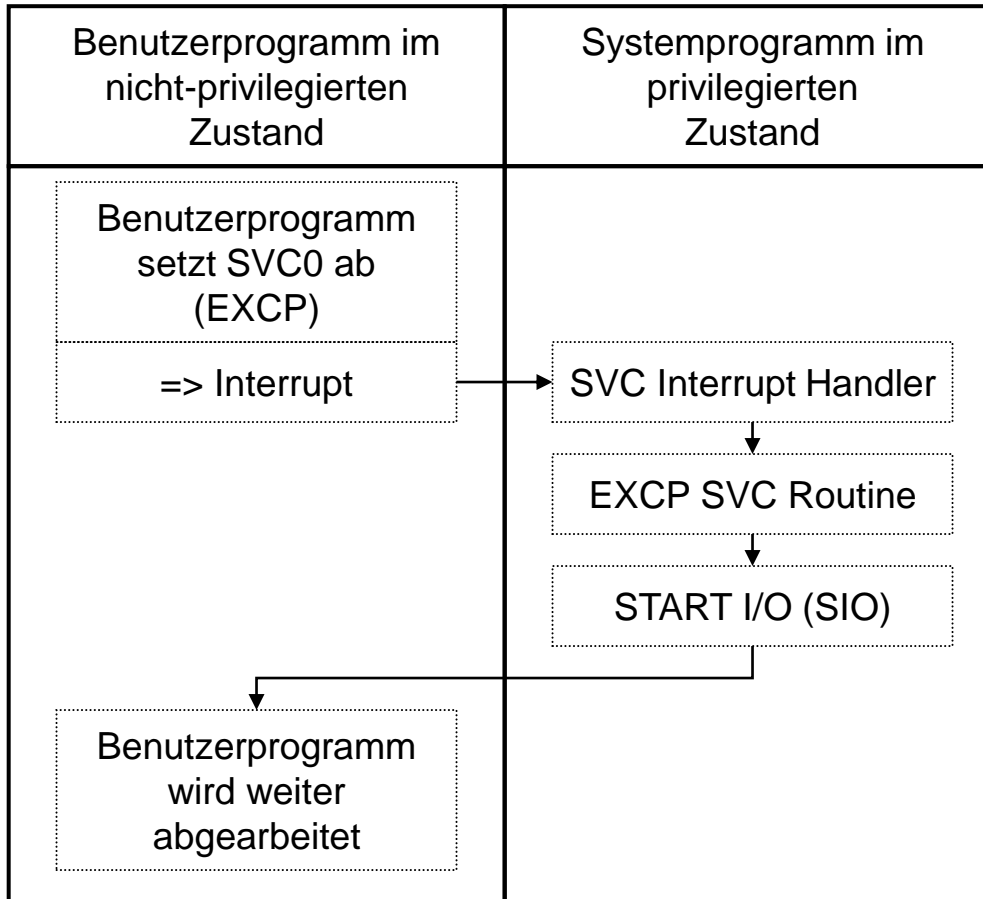
## Aufgaben des Supervisor

---

- Interrupt Handler
- Task Management
- Virtual Storage Management
- Real Storage Management
- Program Management
- Serially Reusable Resource Management
- Timer Management
- Program Interrupt Exit and Dump
- Input/Output Supervisor
- Recovery Management

# Funktionen des Betriebssystems

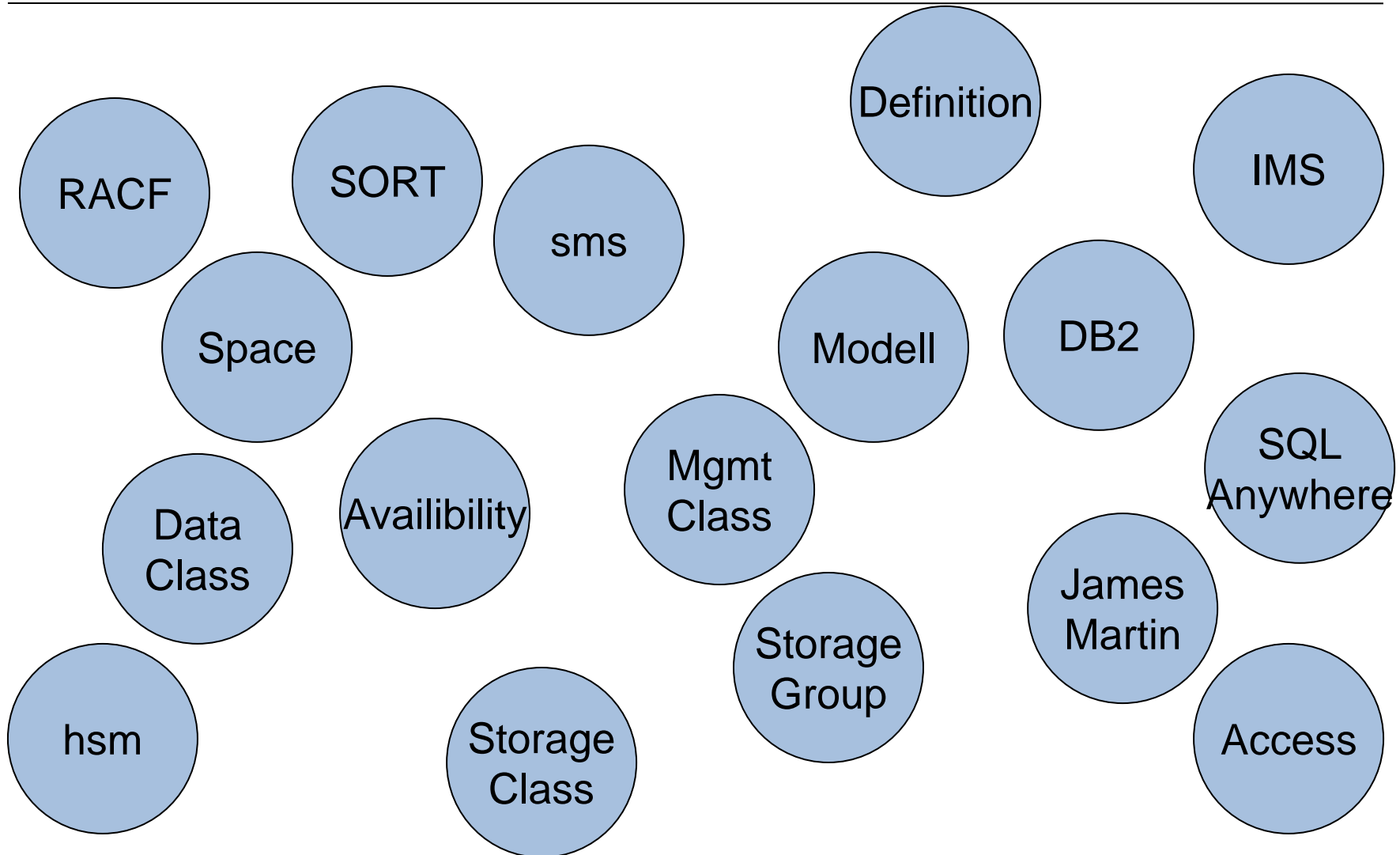
## Supervisor - Ablauf eines Interrupts



- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - • Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

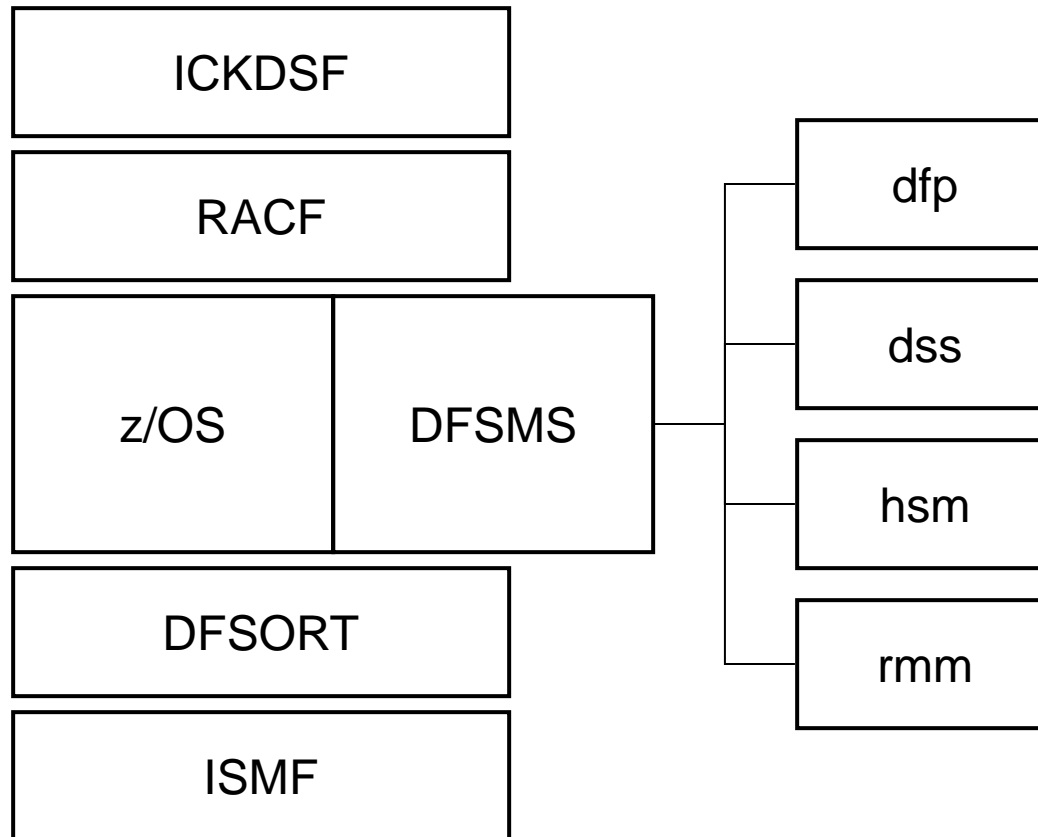
## Begriffe

---



## Überblick

---





## SMS

---

- Nachfolger von DFP, DFHSM, DFDSS
- Komponenten
  - Data Facility Product dfp
    - Zugriffsmethoden, Organisationsformen.  
Datenträgerverwaltung, Katalogverwaltung etc.
  - Data Set Services dss
    - Daten übertragen, Dump, Restore, Defrag
  - Hierarchical Storage Manager hsm
    - Migration, Recall, Dump-, Backup-, Restoreverwaltung
  - Removable Media Manager rmm
    - Verwaltung Wechseldatenträger in Katalogen

## weitere Produkte

---

- Data Facility Sort DFSORT
  - sortieren, mischen von Daten
- Device Support Facility ICKDSF
  - DASD initialisieren, formatieren, analysieren, reparieren
- Resource Acces Control Facility RACF
  - Datenschutz
- Interactive Storage Management Facility ISMF
  - ISPF Schnittstelle zu SMS
- etc.

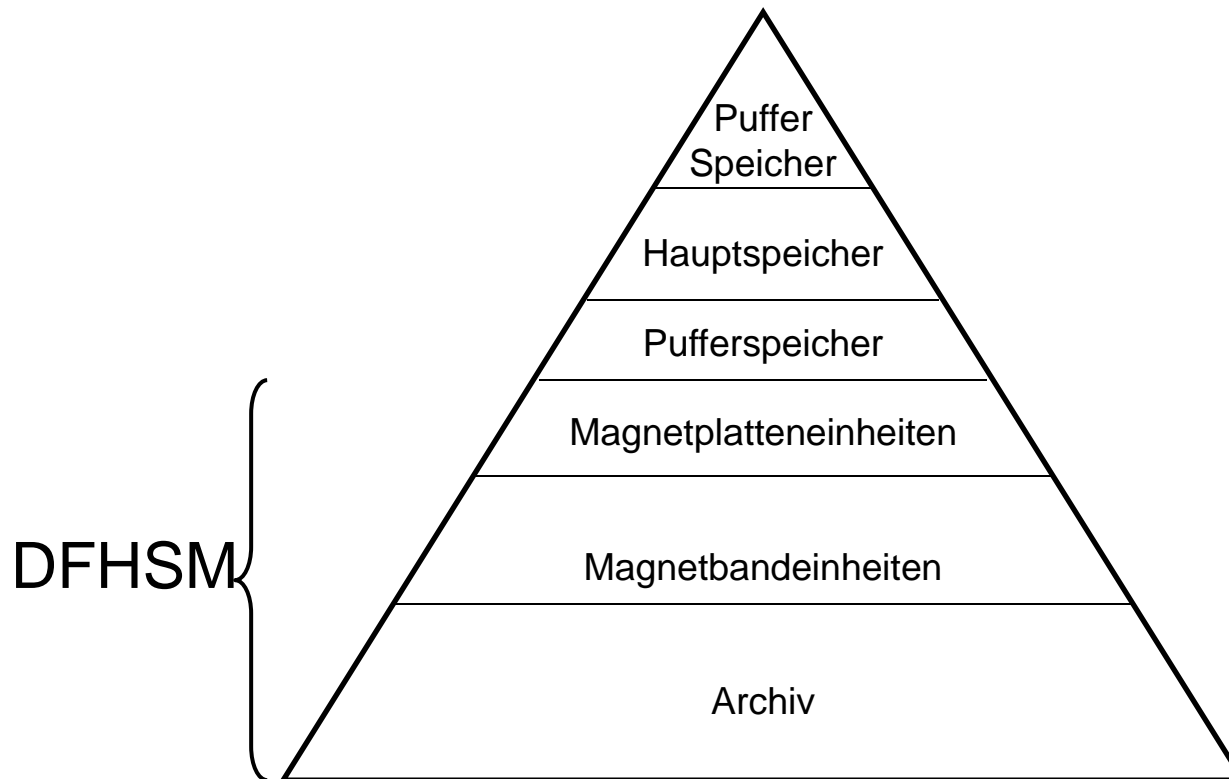
## SMS im Detail

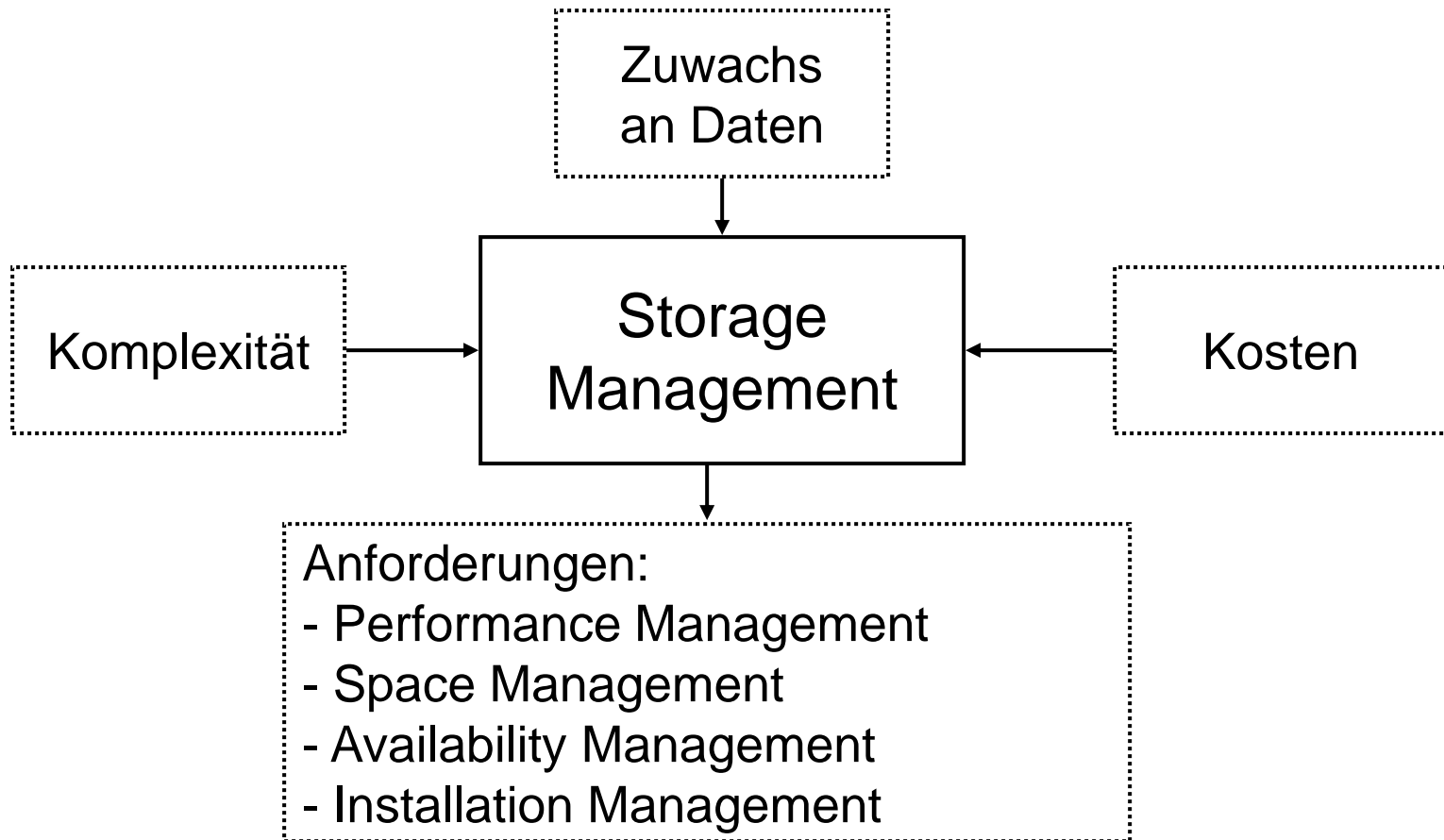
---

- logische Sicht
  - Data Class DC - Art der Daten
  - Storage Class SC - benötigter Hardware Service
  - Management Class MC - benötigter Management Service
- physikalische Sicht
  - Storage Group SG - wo die Daten liegen
  - Steuerung DC z.B. über last-level-qualifier
- Beispiele siehe ISPF

## SMSHsm – Speicherhierarchie im z/OS

---





- Interaktive Dateien
  - erstellen auf Anforderung
  - undefinierte Lebensdauer
  - sporadische Benutzung
  - bei Nutzung ist diese intensiv
  - backup notwendig

- Batch Dateien
  - erstellen auf Anforderung
  - genau definierte Lebensdauer
  - sporadische Benutzung
  - bei Nutzung ist diese intensiv
  - backup notwendig

- System Dateien
  - werden selten verändert
  - konstante Anzahl
  - definierte Stellen
- Datenbanken
  - sehr groß
  - Anzahl relativ konstant
  - spezielle Recovery Techniken



## SMShsm – Ziele

---

- optimieren der Produktivität
- optimieren der Speicherauslastung
- sichern der Datenverfügbarkeit
- gewährleisten Datensicherheit
- einfache Benutzerschnittstelle
- unterstützen von Konvertierungen

## SMSHsm – Hauptfunktionen

---

- Space Management
  - migrate, recall, delete
- Availability Management
  - Backup, Recover, Dump, Restore
- weitere Funktionen
  - modifizieren Primary Allocation, komprimieren, reduzieren Extents, Löschen, ausschließen von migrate etc.

## Dienstprogramme für Dateien

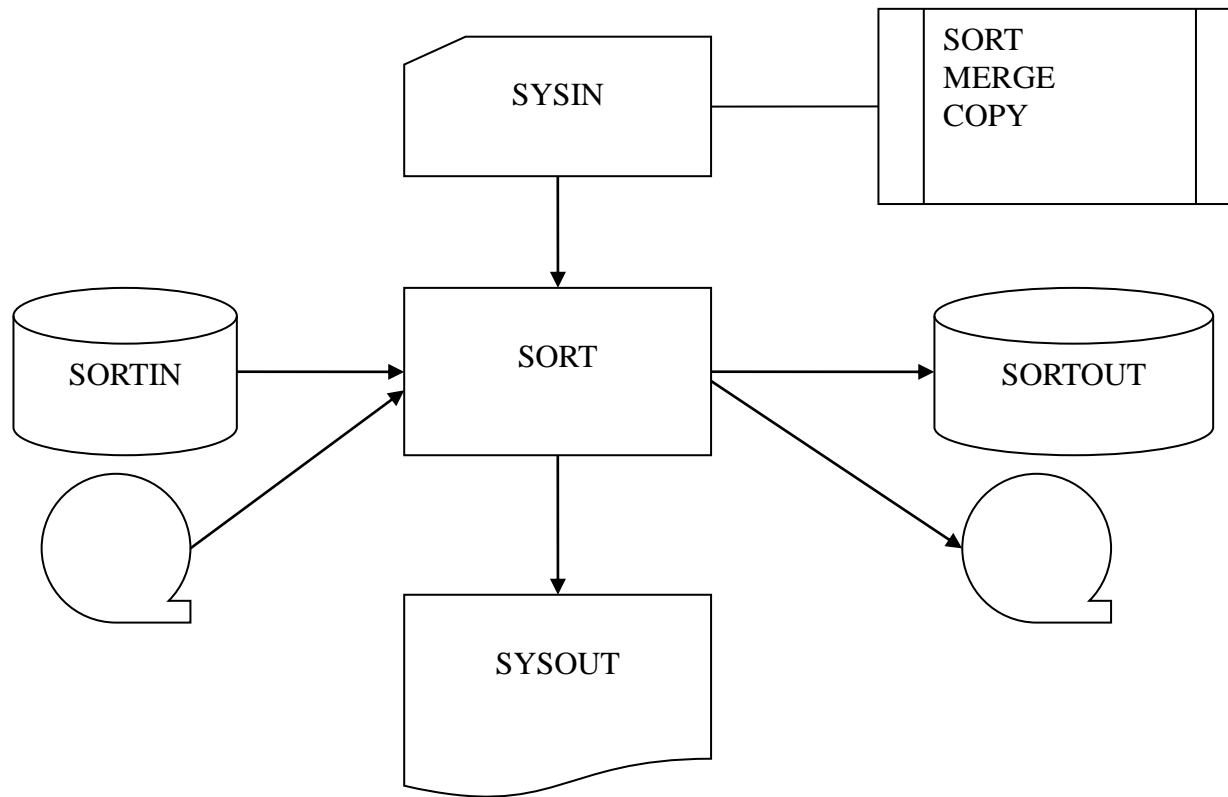
---

- IEBCOPY
- IEBGENER
- IEHLIST
- IDCAMS
- ICEMAN



## SORT

---



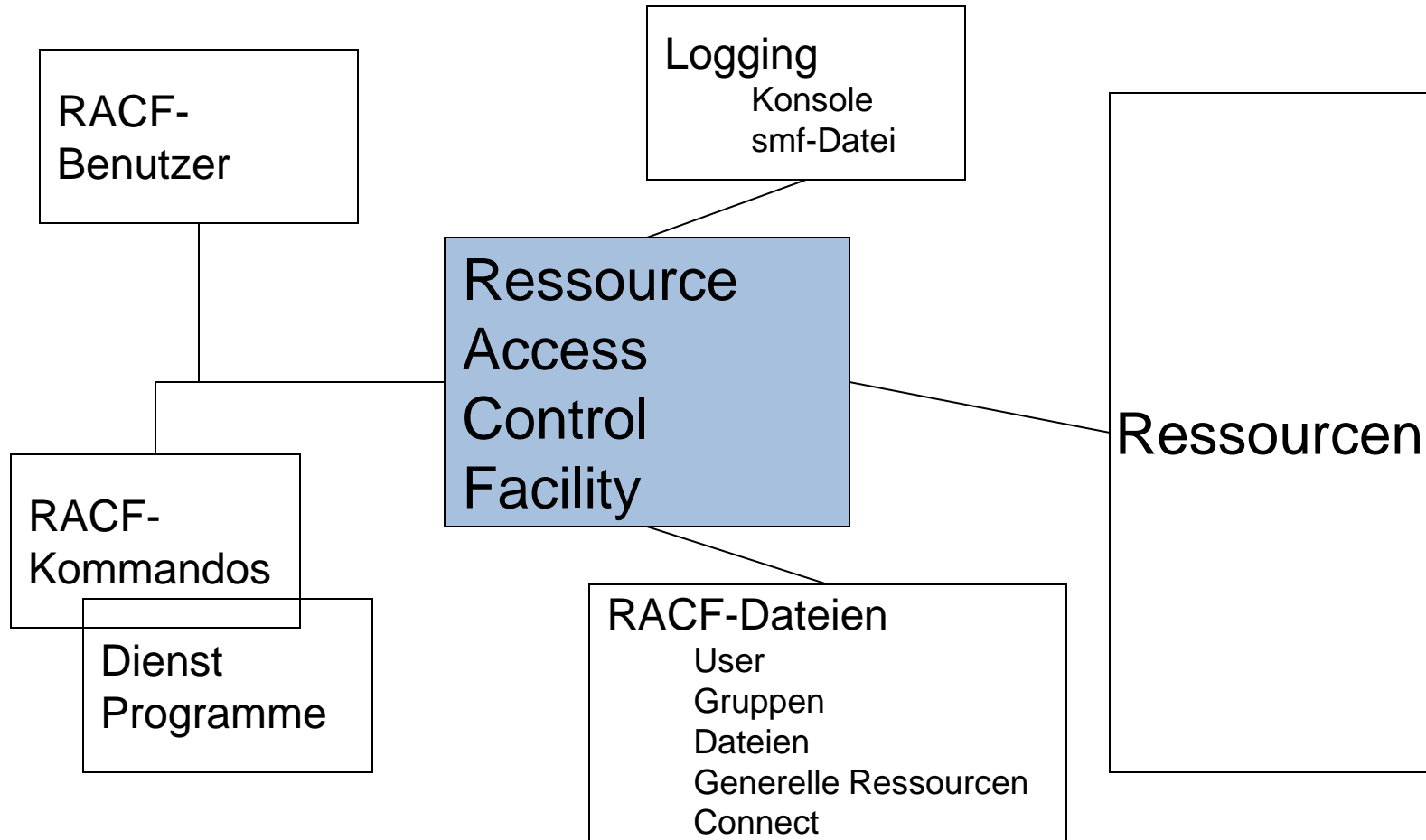
## RACF - Anforderungen

---

- Identifikation von Benutzern
- Verifizierung
- Zugriffsberechtigung
- Kontrolle und Protokollierung
- Benutzerschnittstelle
- klare Strukturen
- einfache Handhabung



## RACF – Überblick

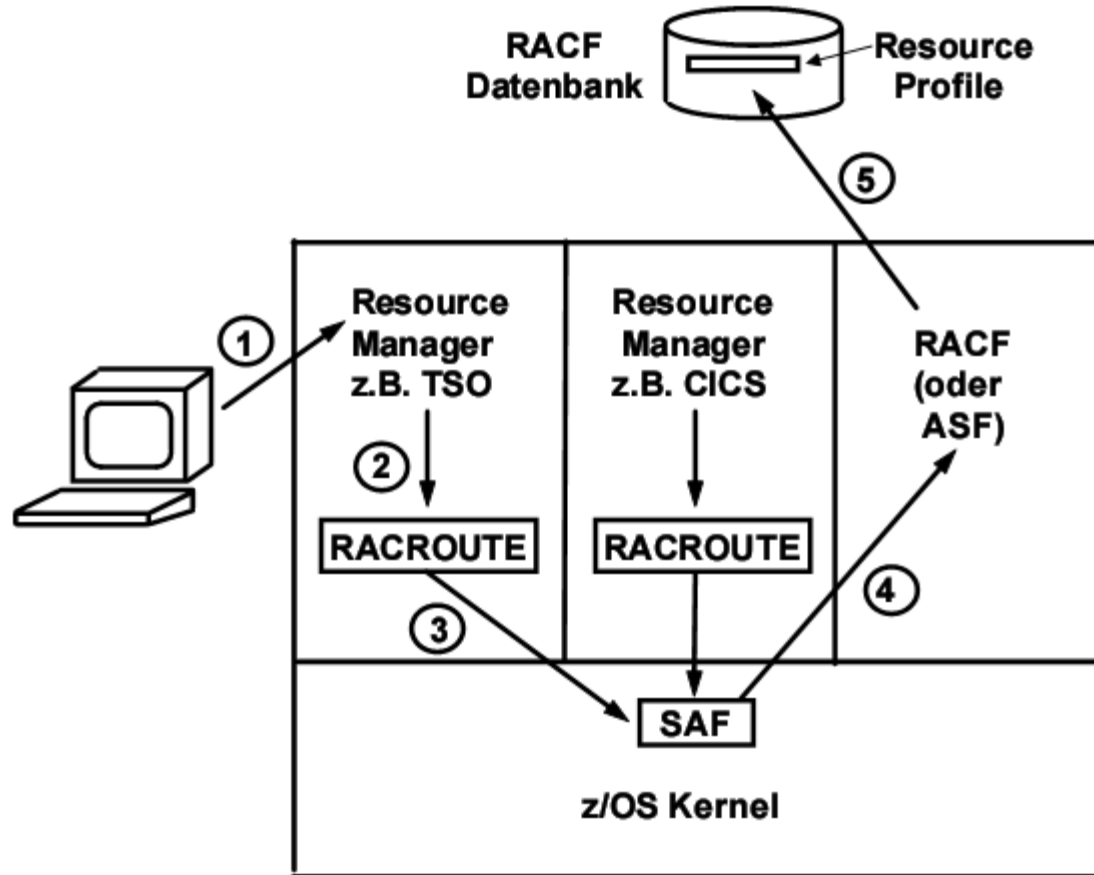


## RACF – Prinzip

---

- Was nicht definiert ist darf nicht.
- Ressourcen werden in „Klassen“ aufgeteilt
  - Benutzer
  - Dateien
  - CICS-Transaktionen
  - Datenbank-Rechte
  - Terminals
- Jedem Element wird ein „Profil“ zugeordnet
  - Benutzerprofile beinhalten „Capabilities“
  - Dateiprofile enthalten „Access Control Listen“
- Herausforderung: Pflege der Profile

## RACF – Arbeitsweise



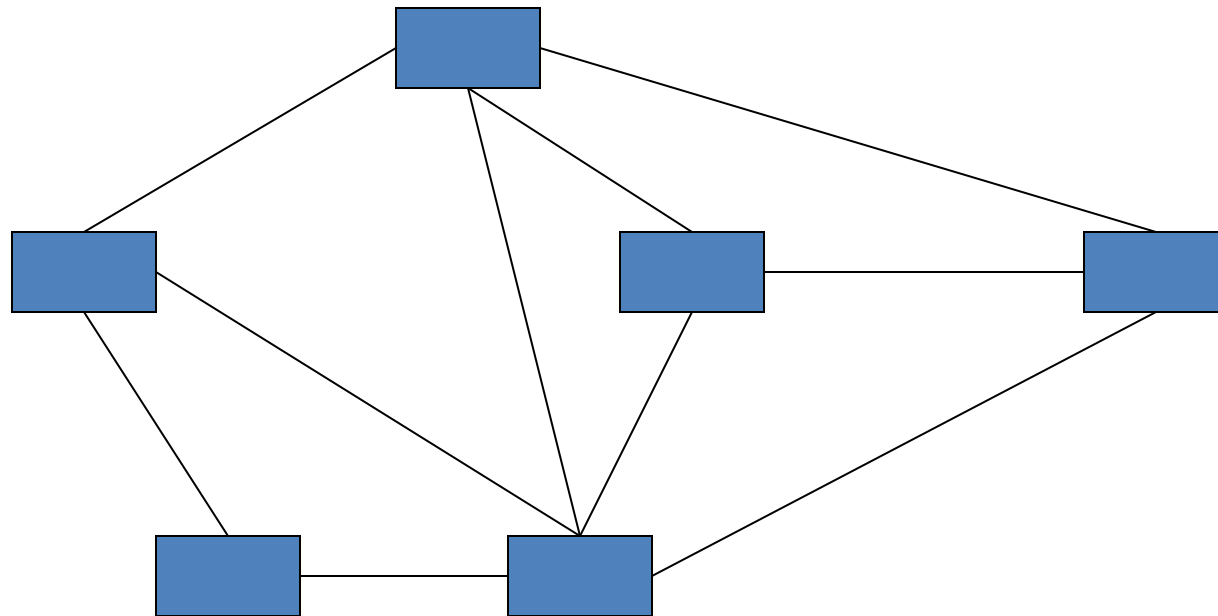


## Datenbank – Definition

---

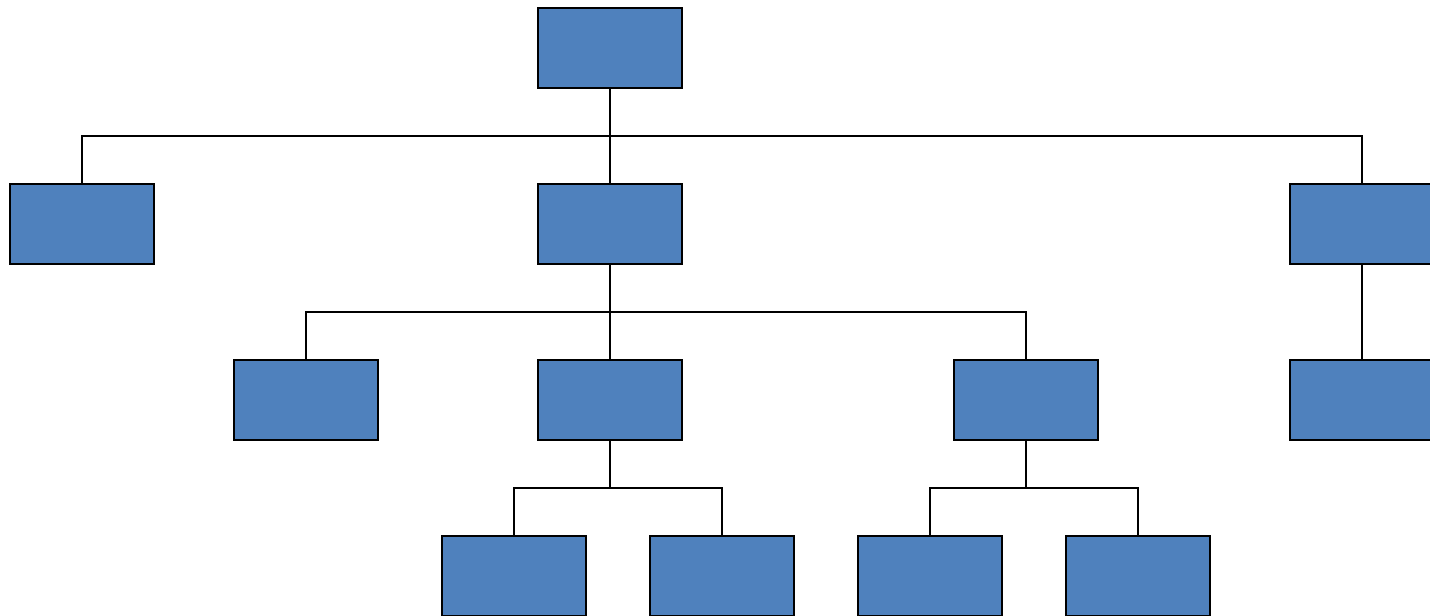
- Eine Sammlung von Daten, die von verschiedenen Programmierern verwendet werden kann, wird Datenbank genannt. Wir definieren sie als eine Sammlung von inhaltlich zusammen hängenden Daten, um für eine oder mehrere Anwendungen in optimaler Art und Weise verwendbar zu sein.
- Die Daten werden so abgespeichert, dass sie unabhängig von den Programmen sind, von denen sie benutzt werden. Eine allen gemeinsame und kontrollierte Lösung wird für das Hinzufügen, das Modifizieren und das Abfragen abgespeicherter Daten benutzt. Ein System kann mehrere Datenbanken enthalten, wenn jede für sich eine eigenständige, von den anderen verschiedene Datenstruktur hat.

Autor: James Martin

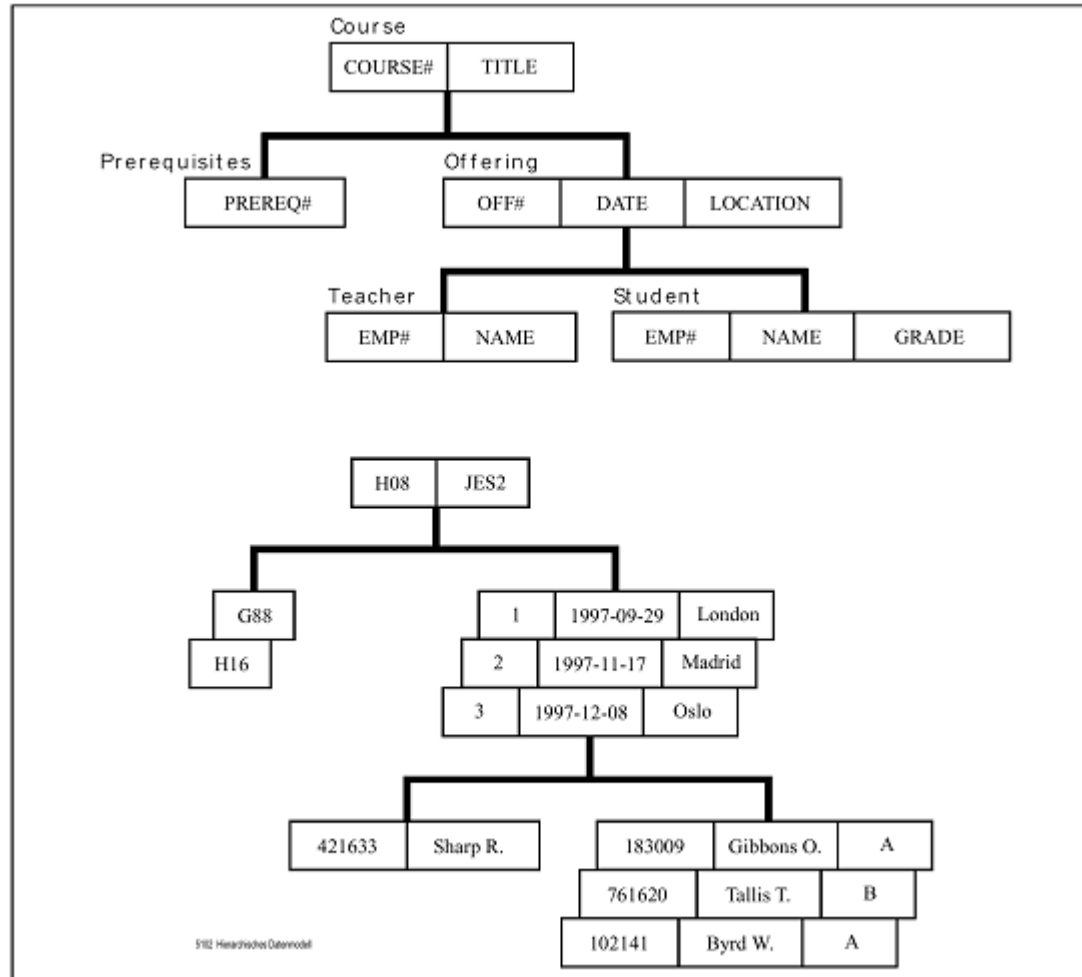


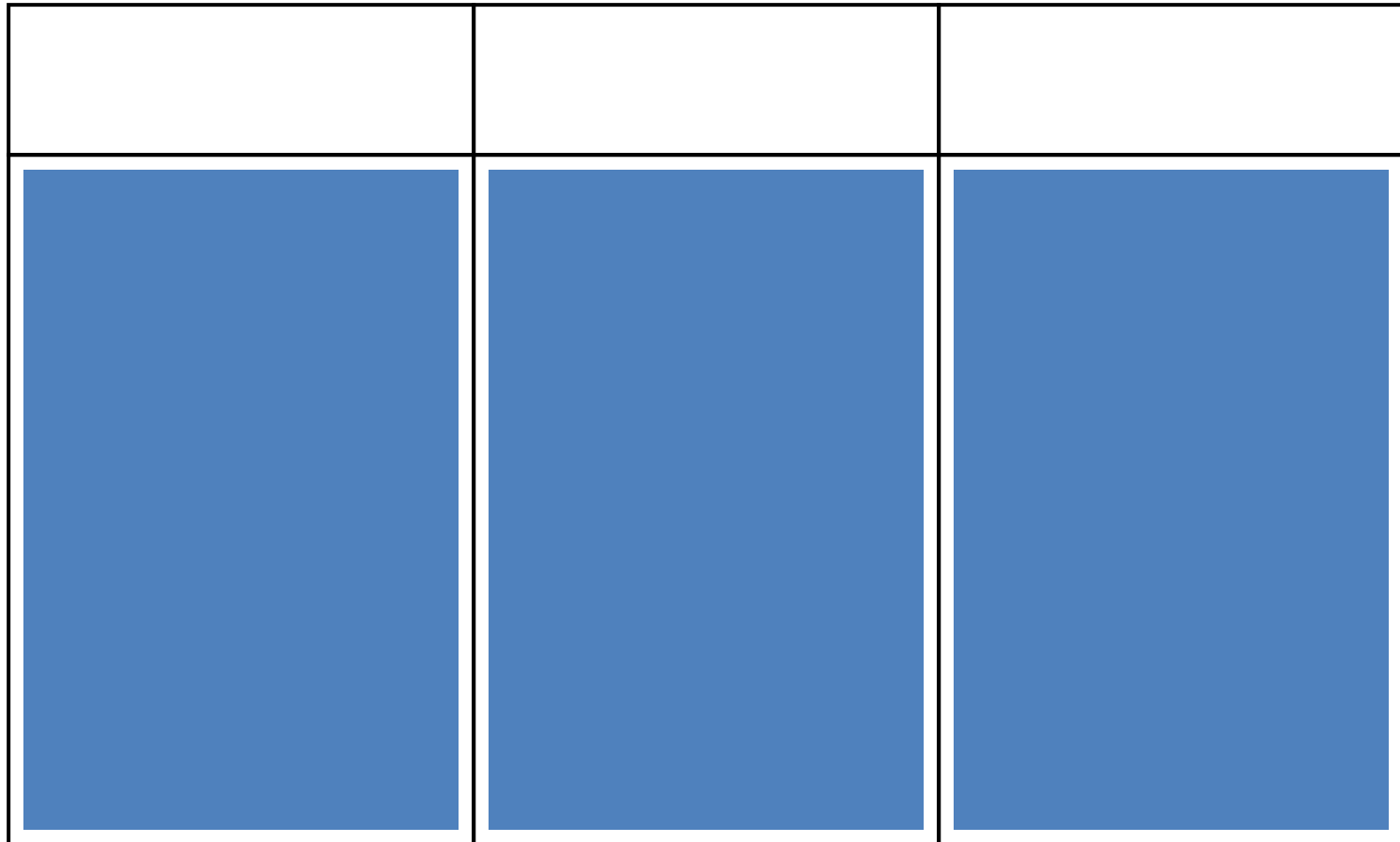
## hierarchisches Modell

---

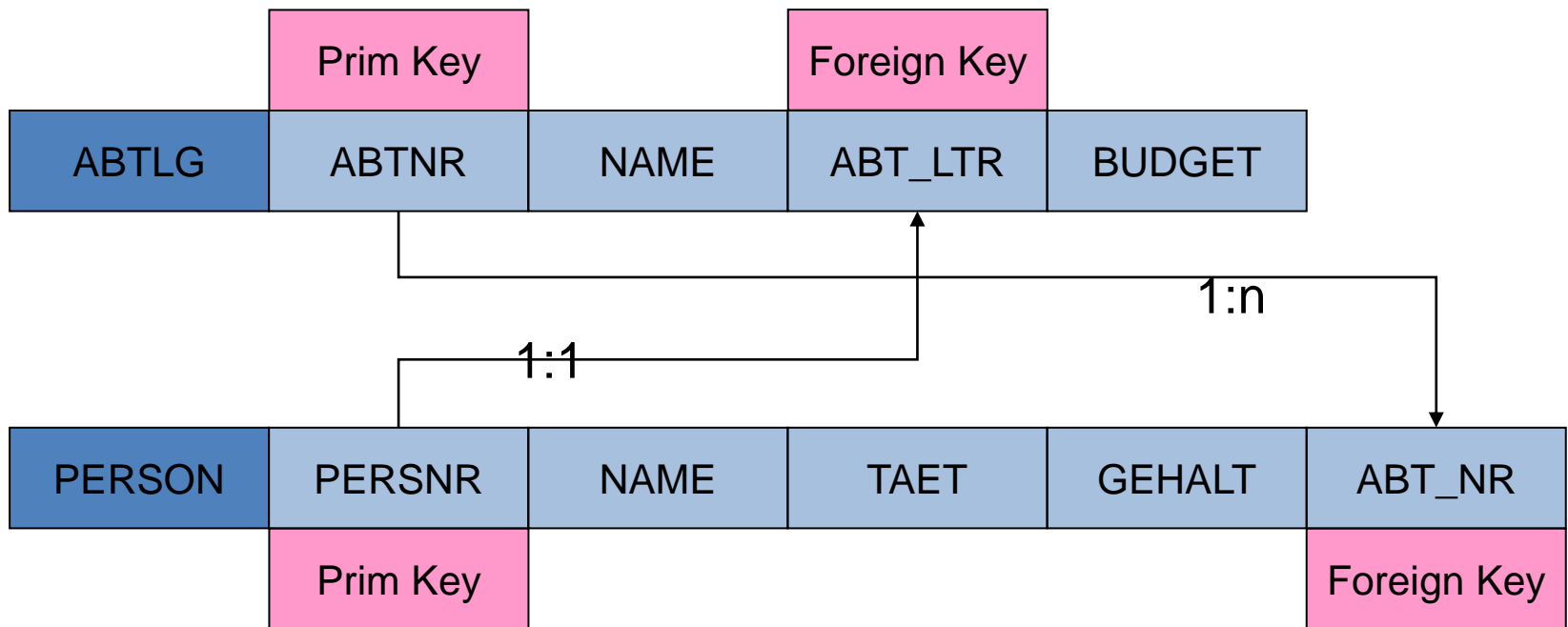


## Beispiel – IMS

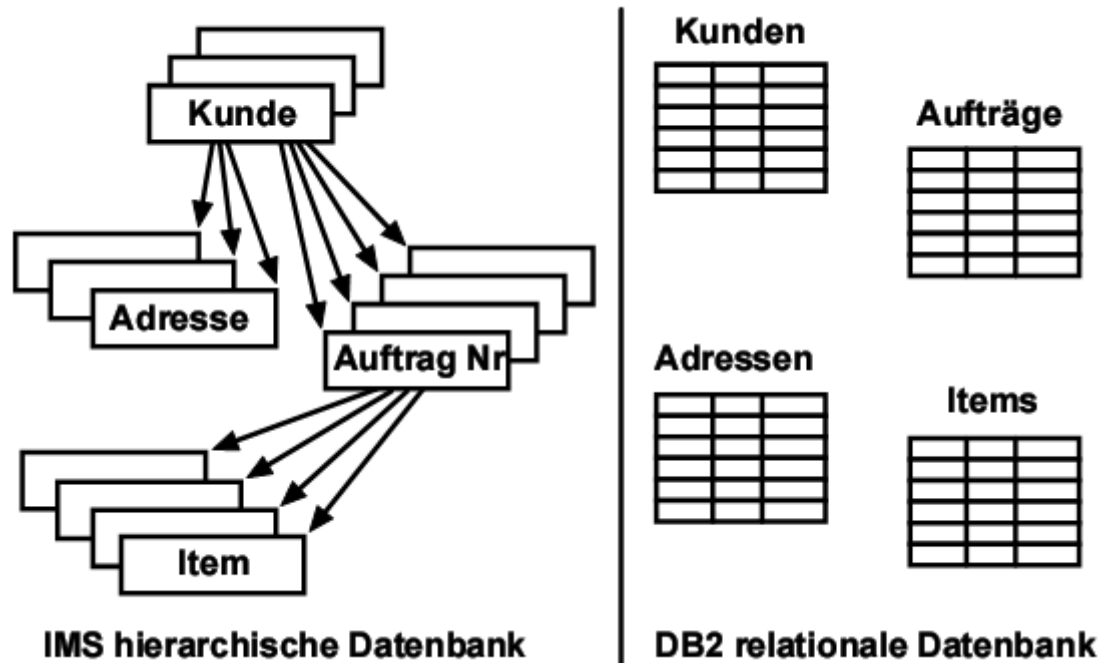




## Beziehung zwischen Tabellen



## IMS vs. DB2

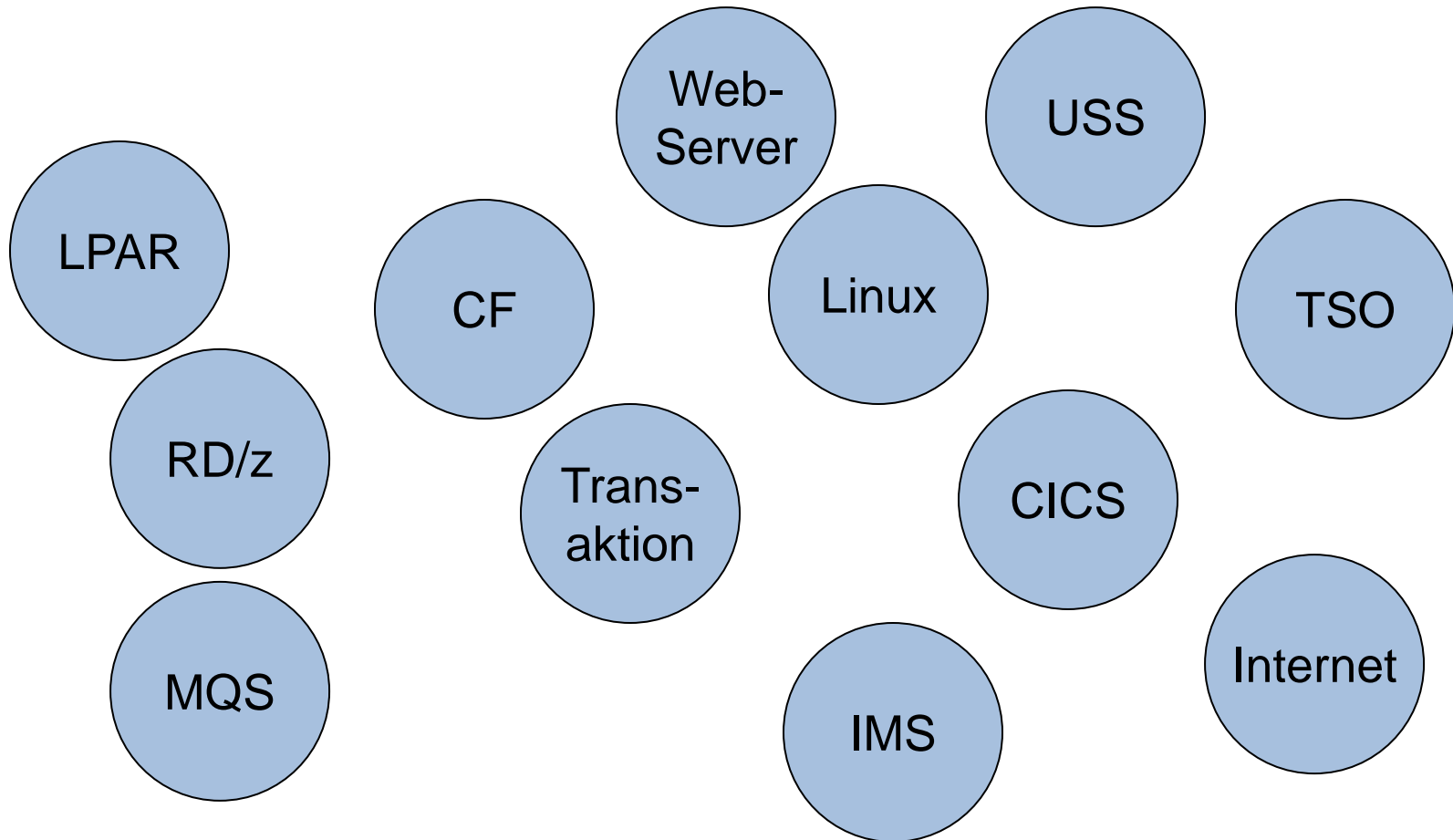


- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - • weitere Subsysteme und Features
  - Begriffe ;-)



## Begriffe

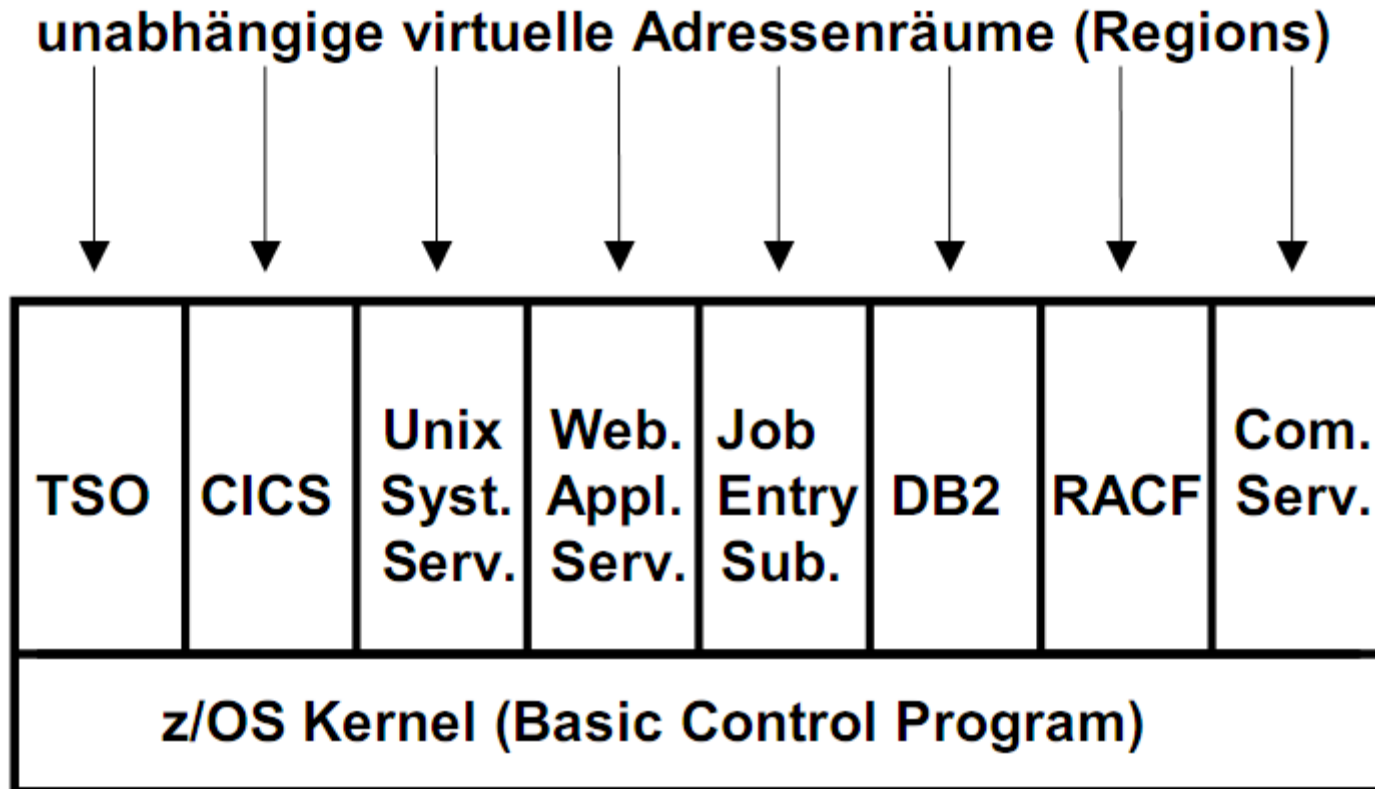
---



# weitere Subsysteme und Features

## z/OS Grundstruktur

---



## Subsysteme auf einen Blick – 1

---

- CICS
- IMS (IMS/DB und IMS/TM)
- DB2
- VSAM
- XCF
- JES2 / JES3
- Security Server (RACF, Firewall, DCE Security)
- Netview, Systemview
- WebSphere
- UNIX Services

## Subsysteme auf einen Blick – 2

---

- Distributed Computing Services (DCE, NFS, DFS, FTP)
- Lan Server
- Runtime Language Support (COBOL, C++ etc.)
- C/C++ Open Class Library
- JDK, JVM
- etc.

## DB2

---

- relationales Datenbanksystem
- verfügbar auf verschiedenen Plattformen
- Kommunikation via SQL
  - Standard und mächtige Erweiterungen
  - einbindbar in Programmcode
  - dynamischer / statischer SQL
- offen für Zugriff via TCP/IP
- mit Java-Konnektoren
- Stored Procedures sind möglich
- logging / locking / rollback etc.



## IMS/DB

---

- hierarchisches Datenbanksystem
- proprietäres IBM-System
- Kommunikation via IMS-Calls
  - GU, GN, GHN, DLTE, ISRT etc.
  - Positionierung erforderlich
- offen für Zugriff via TCP/IP
- mit Java-Konnektoren
- logging / locking / rollback etc.
- verschiedene technische Restriktionen

## IMS/DB und DB2

---

- IMS – es wird kodiert, wie IMS lesen soll
- DB2 – es wird kodiert, was DB2 lesen soll
- beide DB-Systeme parallel nutzbar
- IMS ist der „Chef“ bzgl. locking/rollback/sync
- DB2 variabler
- IMS schneller
- ...

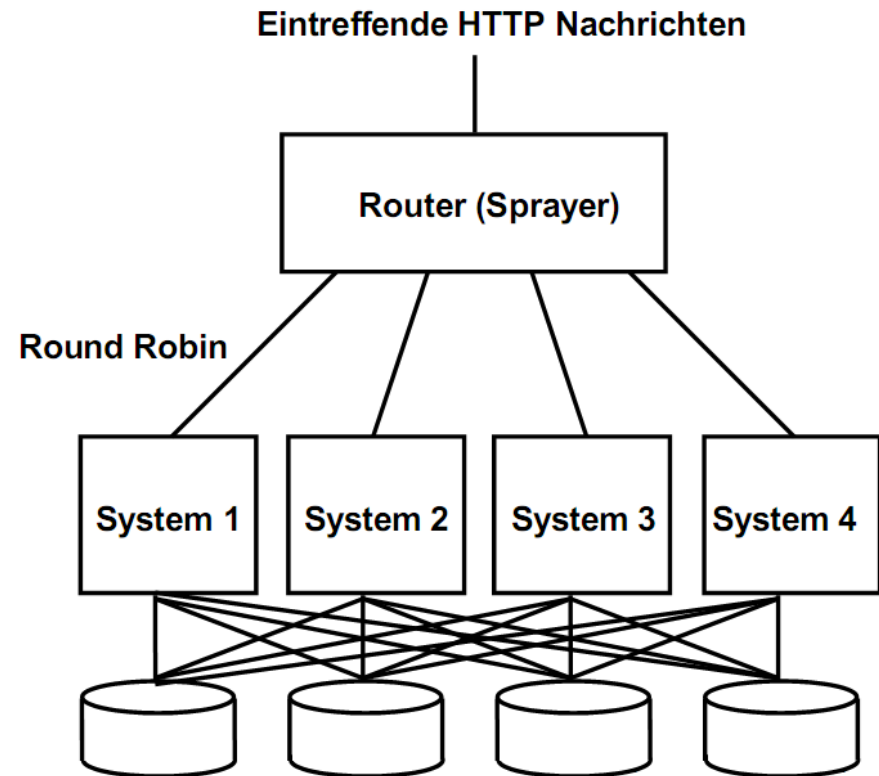
## IMS/TM

---

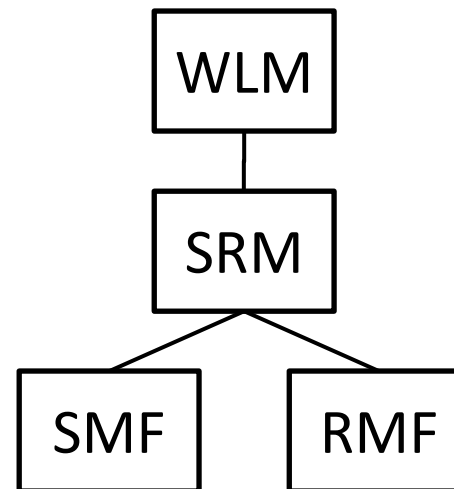
- Transaktionsmonitor des IMS
- Kontrolle unter IMS
- Kommunikation mit DB2 möglich
- offen für Zugriff via TCP/IP
- mit Java-Konnektoren
- weitere Eigenschaften siehe CICS



- Beispiel eines WLM



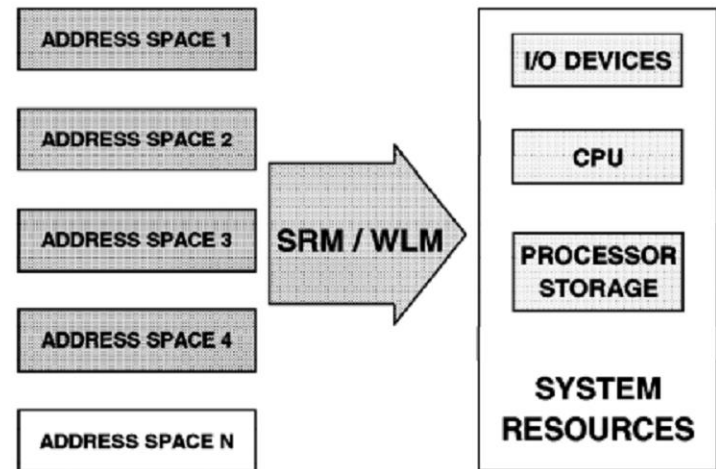
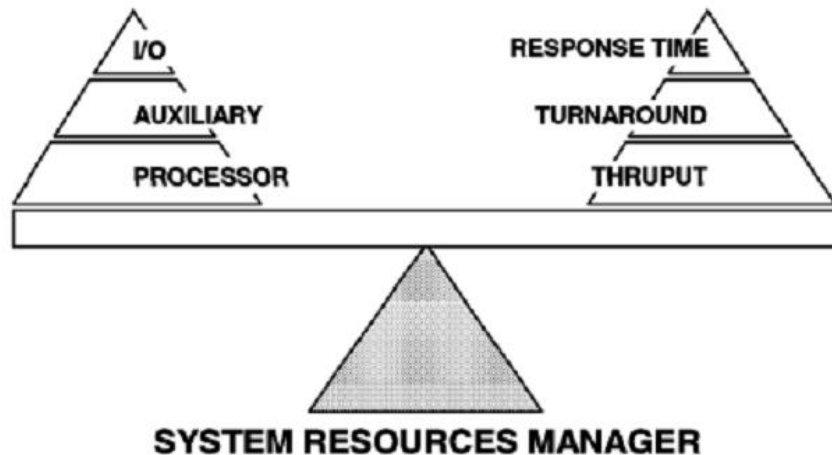
- WLM („Ziel-orientiert“)
- SRM (System Resource Manager)
- SMF (System Management Facility)
- RMF (Resource Management Facility)



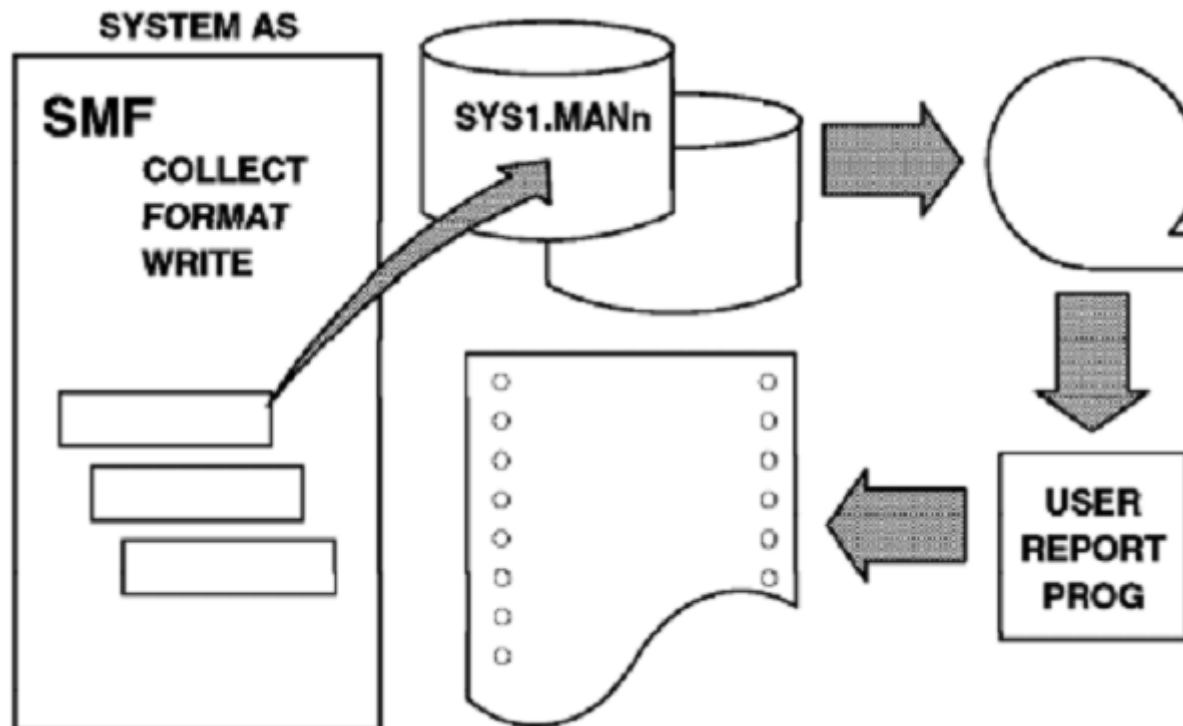
# weitere Subsysteme und Features

## Work Load Manager im z/OS – Komponente SRM

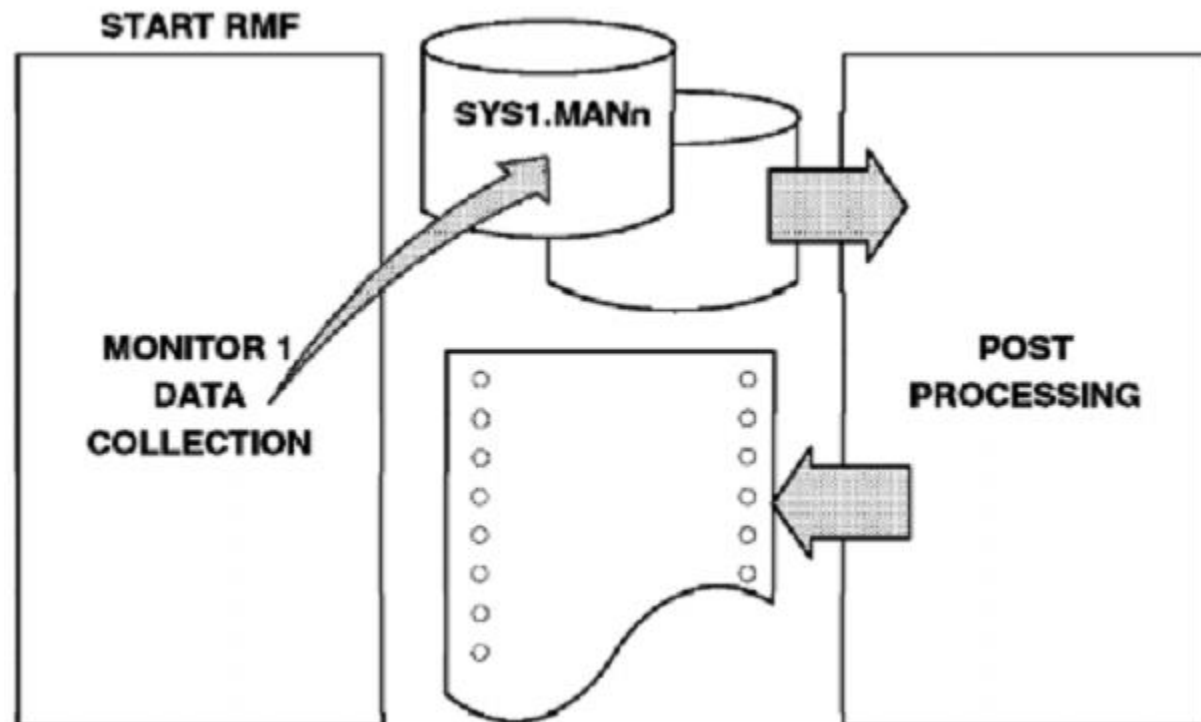
- CPU-Auslastung
- Hauptspeicher Nutzung
- E/A Belastung



- loggen aller System relevanter Daten wie
  - Konfiguration, Workload, Paging, Swapping, JES-Infos

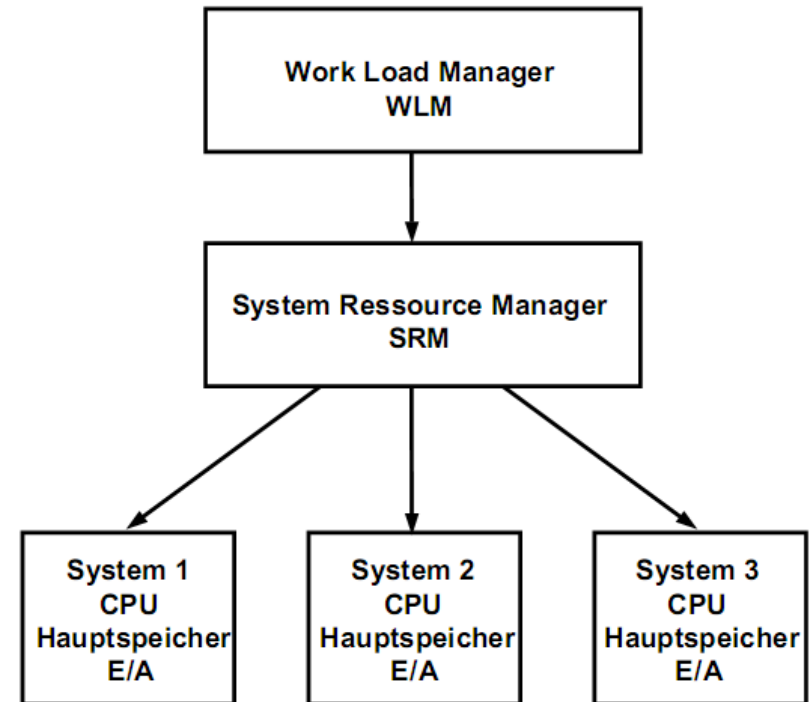
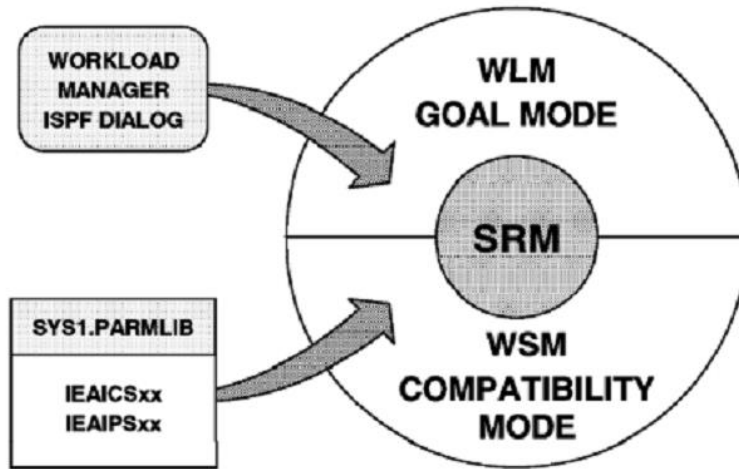


- loggen aller System Daten wie
  - CPU-Verbrauch, DASD-Aktivitäten, Hauptspeichernutzung

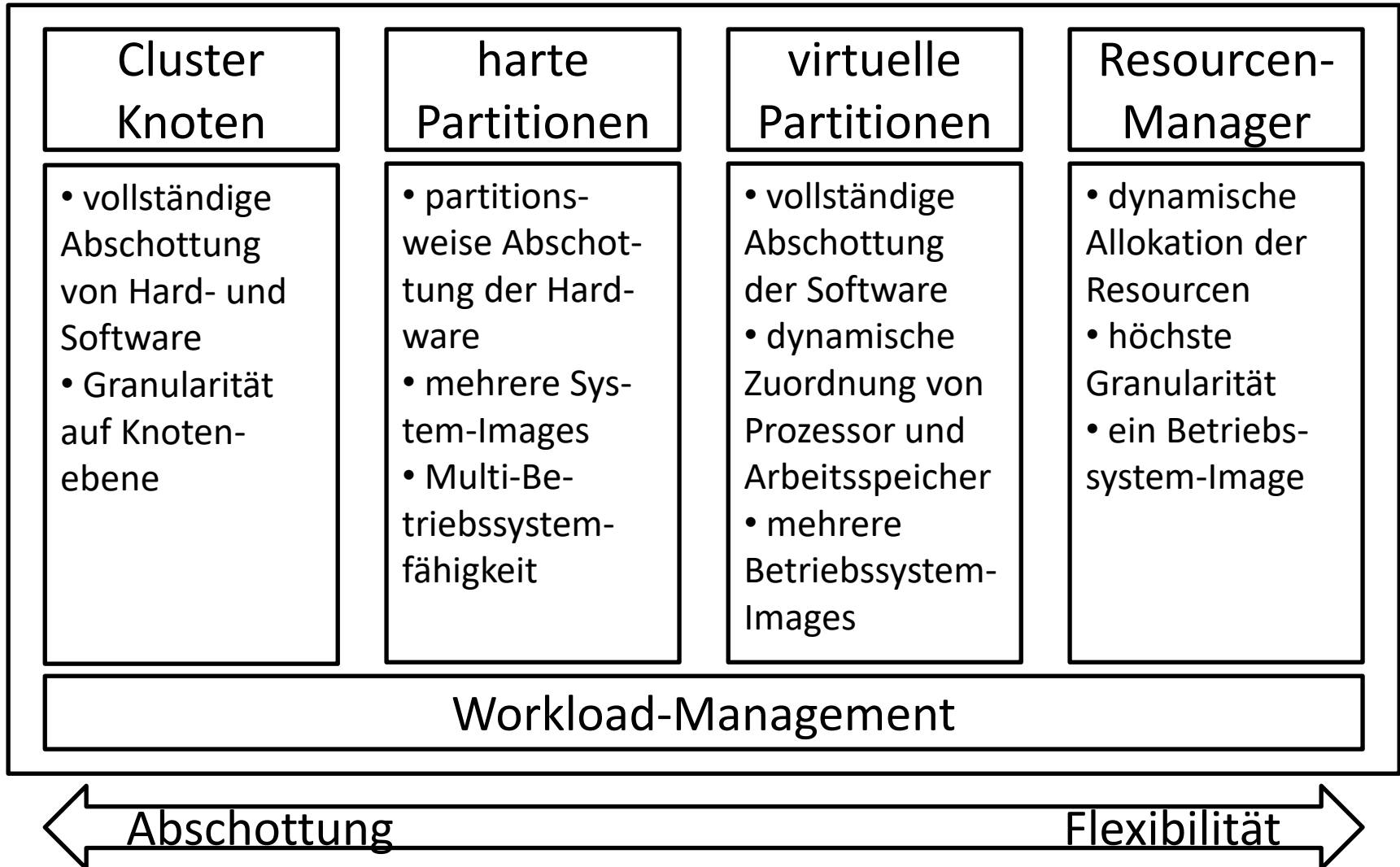


## Work Load Manager im z/OS – Grenzen

- WLM über 1 Betriebssystem hinweg



## Partitionskonzepte



## LPAR

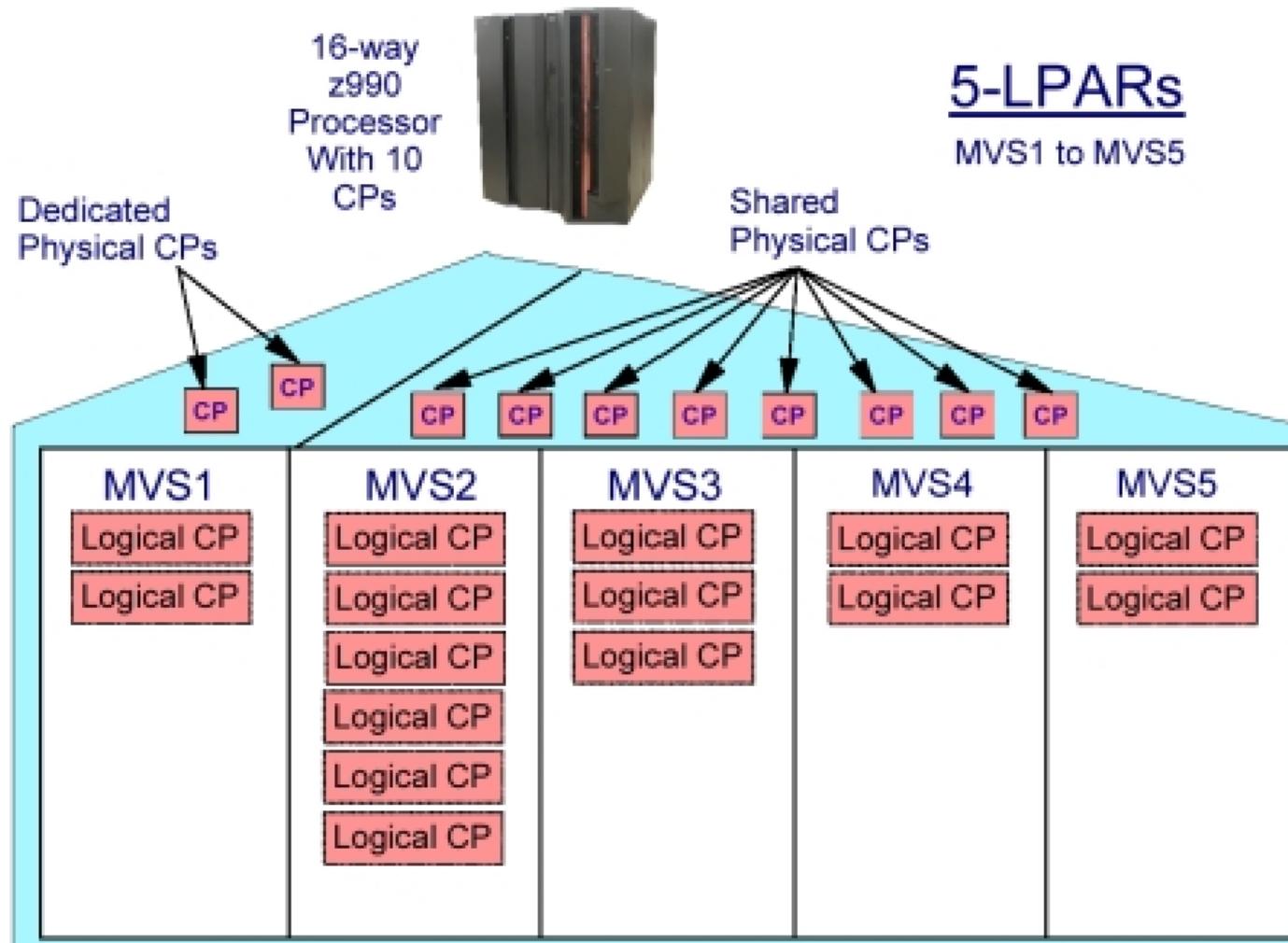
---

- Aufteilung in virtuelle Systeme
- LPAR <-> Betriebssystem
- Umsetzung durch PR/SM (Processor Resource System Manager)
- LPARs mit Zusammenfassung als SysPlex (system complex) möglich
- Ressourcen werden dynamisch durch Workload Manager verwaltet
- auf LPAR kann „beliebiges“ Betriebssystem laufen
- bis 255? LPARs auf z16 (1 für System)



# weitere Subsysteme und Features

## LPAR – Beispiel



## Parallel Sysplex

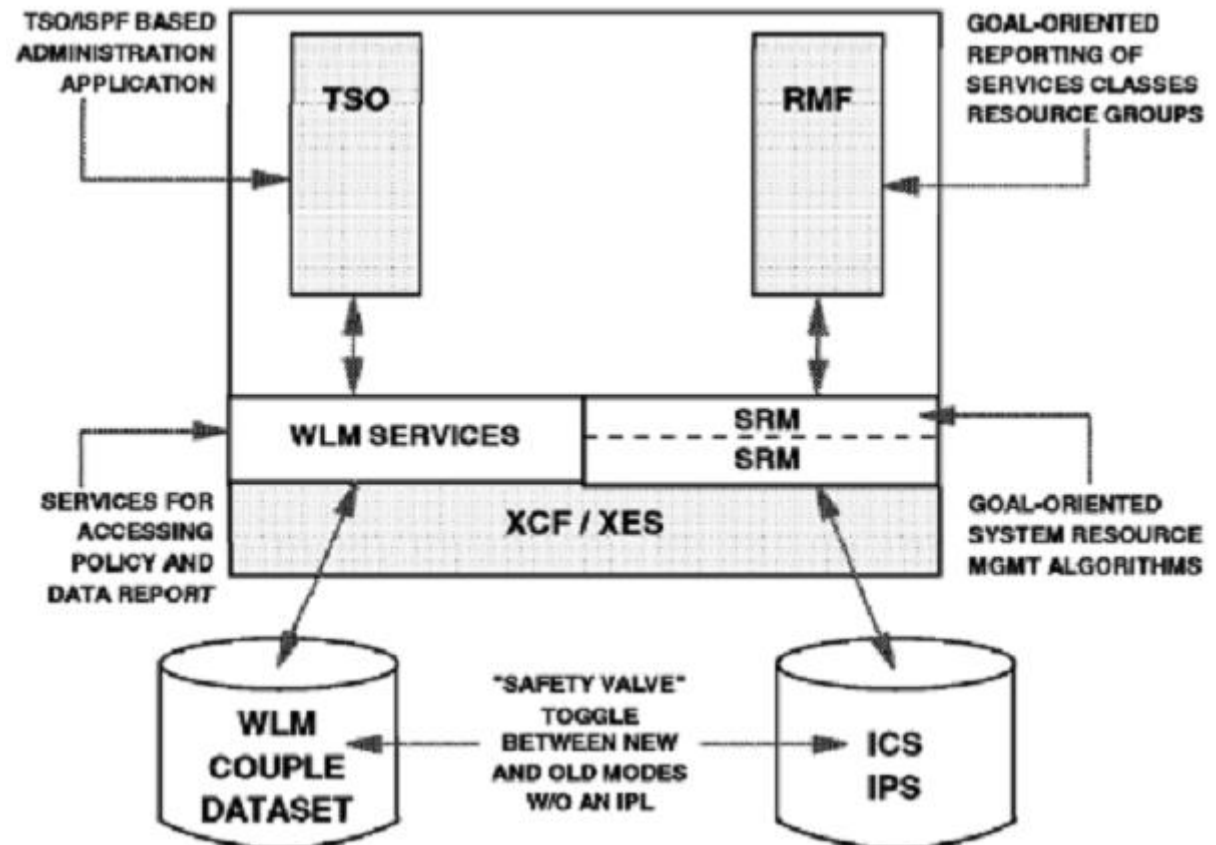
---

- Cluster von „Knoten“
- Knoten ist SMP (symmetrischer Multiprozessor)
- 1 SMP enthält bis 24(?) CPUs + 3 E/A-Einheiten
  - SMP -> LPAR (logische Partition)
  - SMP -> Betriebssystem
- Kommunikation der SMPs mit E/A via
  - ESCON / FICON
- Synchronisierung von Daten via CF (Coupling Facility)

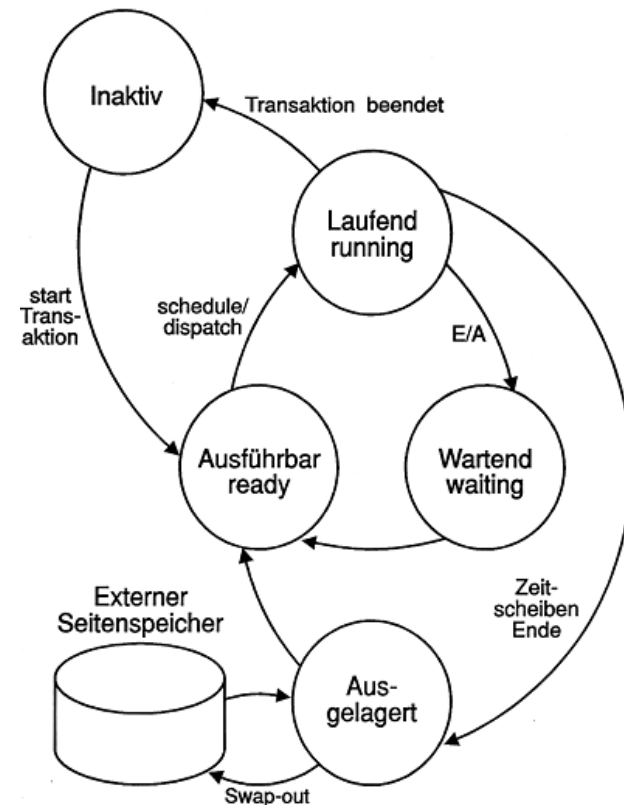
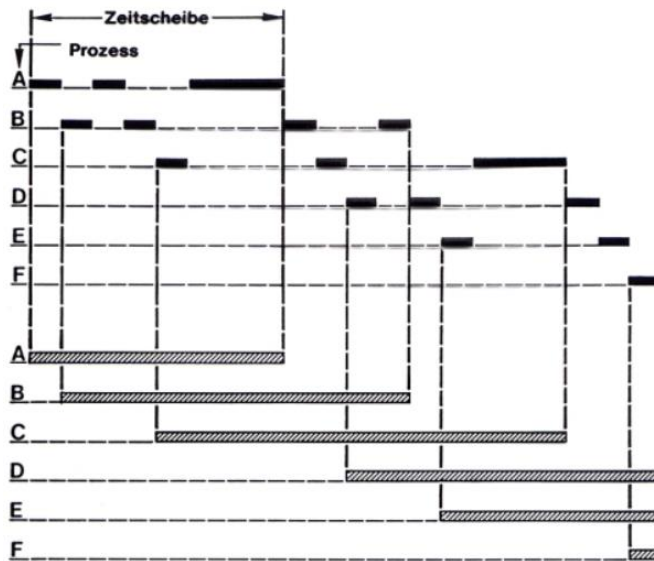


## Work Load Manager im Sysplex

- Ausbalancieren aller Systeme und Subsysteme durch SRM

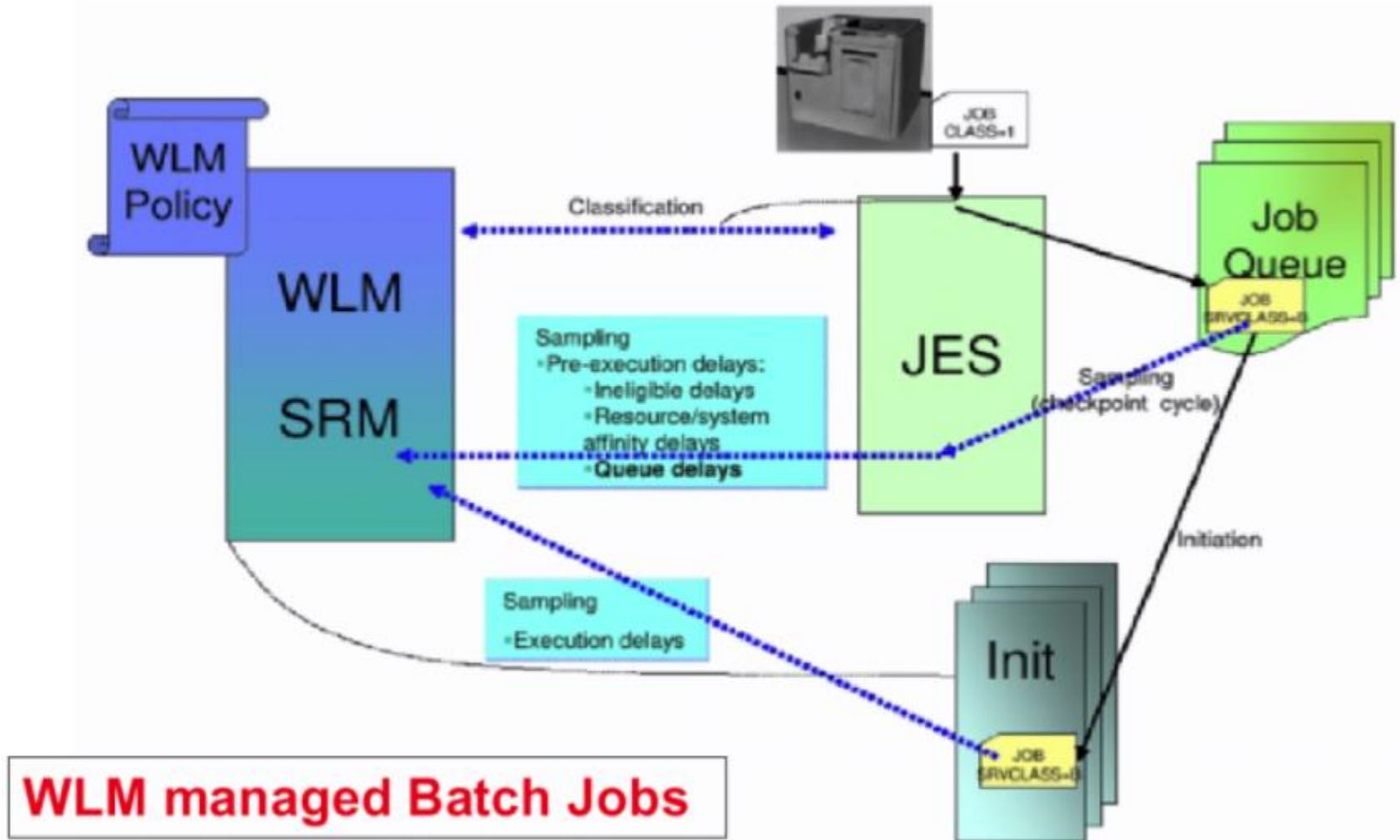


- swapping (auslagern von Prozessen)
- einstellen Multi-Programming-Level
- HS-Vergabe
- Prioritätenvergabe



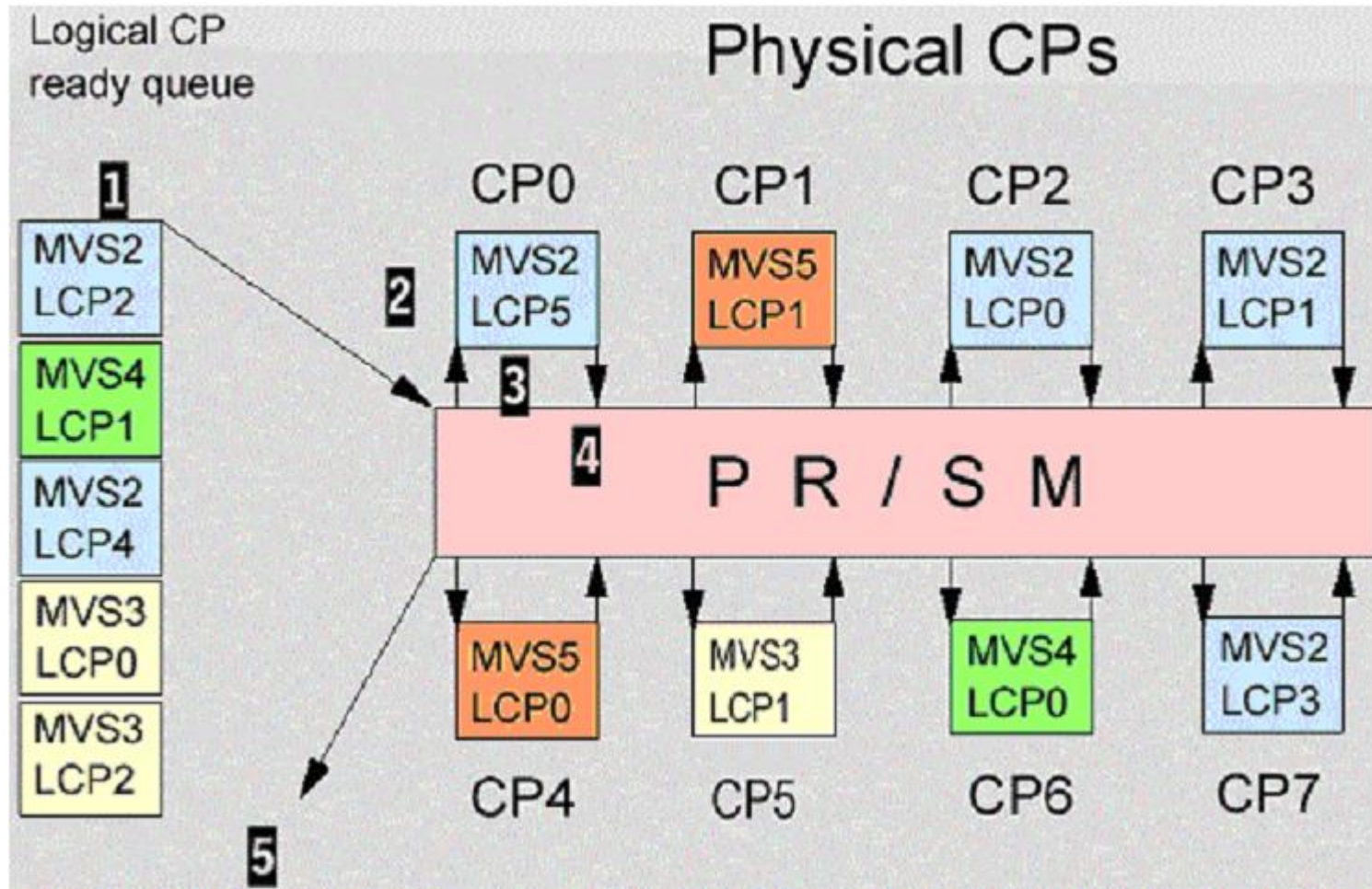
# weitere Subsysteme und Features

## Work Load Manager im Sysplex – Beispiel Batchjobs



# weitere Subsysteme und Features

## LPAR – mit PR / SM



## LPAR – mit PR / SM – Beschreibung – 1

---

- Den vier LPARs mit den Betriebssystemen MVS2, MVS3, MVS4 und MVS5 stehen 8 physische CPUs zur Verfügung (CP0 .. CP7).
- Jedes der Betriebssysteme ist als Multiprozessor konfiguriert und glaubt, über eine bestimmte Anzahl (logischer) CPUs zu verfügen. Die Anzahl der logischen CPUs übertrifft die Anzahl der physisch vorhandenen CPUs.
- Der PR/SM Hypervisor ordnet die logischen CPUs den physisch vorhandenen CPUs zu.

## LPAR – mit PR / SM – Beschreibung – 2

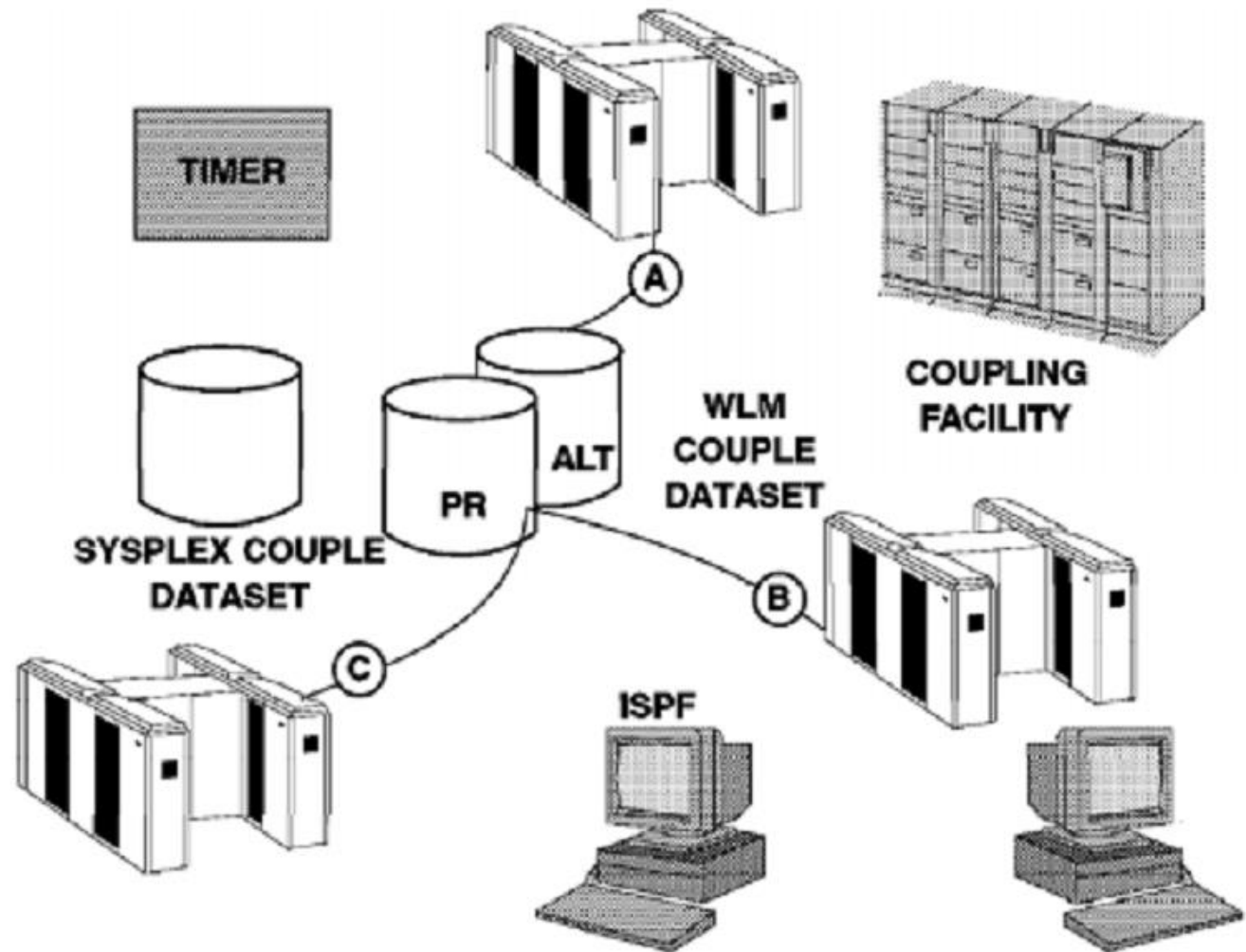
---

1. The next logical CP to be dispatched is chosen from the logical CP ready queue based on the logical CP weight.
2. LPAR LIC dispatches the selected logical CP (LCP5 of MVS2 LPAR) on a physical CP in the CPC (CP0 above).
3. The z/OS dispatchable unit running on that logical processor (MVS2 logical CP5) begins to execute on physical CP0. It executes until its time slice (generally between 12.5 and 25 milliseconds) expires, or it enters a wait, or it is intercepted for some reason.
4. the logical CP keeps running until it uses all its time slice. At this point the logical CP5 environment is saved and control is passed back to LPAR LIC, which starts executing on physical CP0 again.
5. LPAR LIC determines why the logical CP ended execution and requeues the logical CP accordingly. If it is ready with work, it is requeued on the logical CP ready queue and step 1 begins again.



## Work Load Manager im Sysplex – Überblick

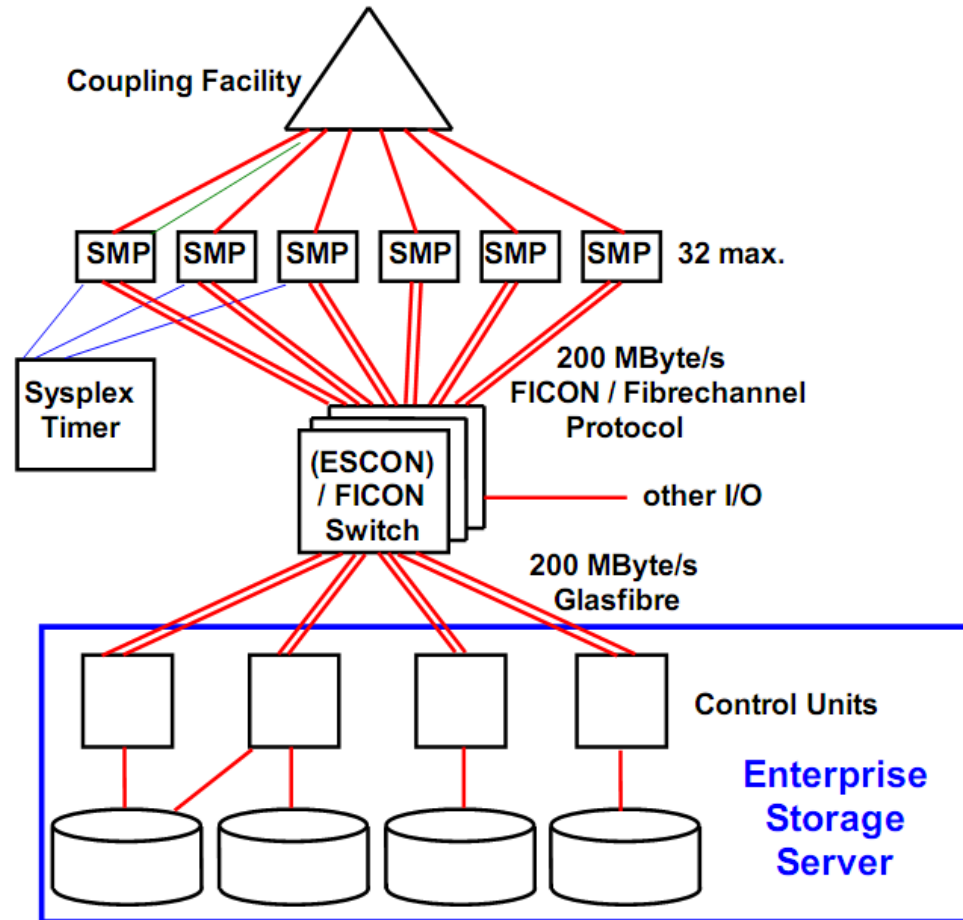
- und was ist mit Dateien und Datenbanken?



# weitere Subsysteme und Features

## CF – Coupling Facility (auch XCF – cross-system CF)

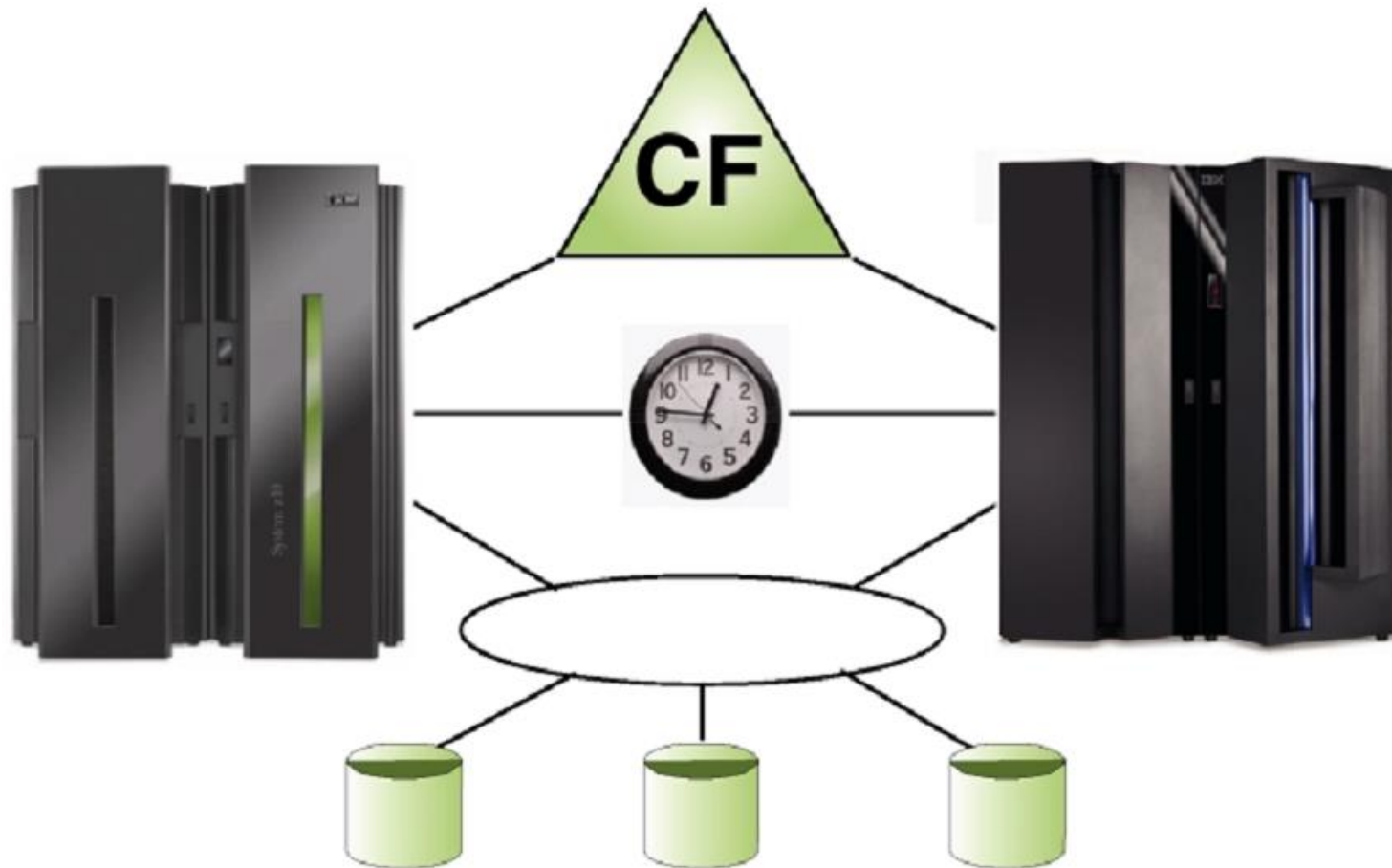
- Aufgabe
  - locking
  - caching
  - listing



**Sysplex  
with  
Coupling  
Facility**

## CF – Bausteine

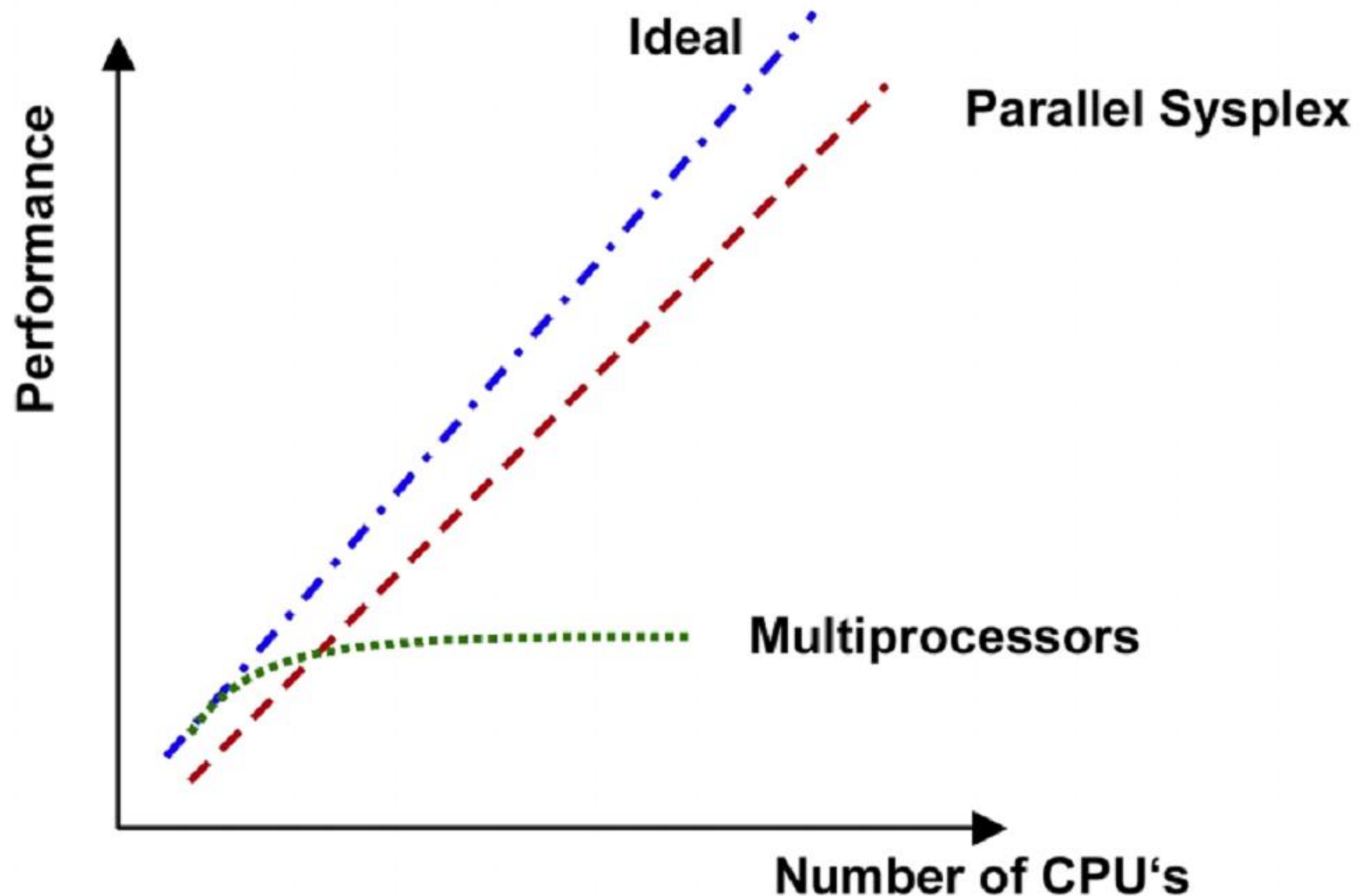
---



**zSeries Coupling Facility**

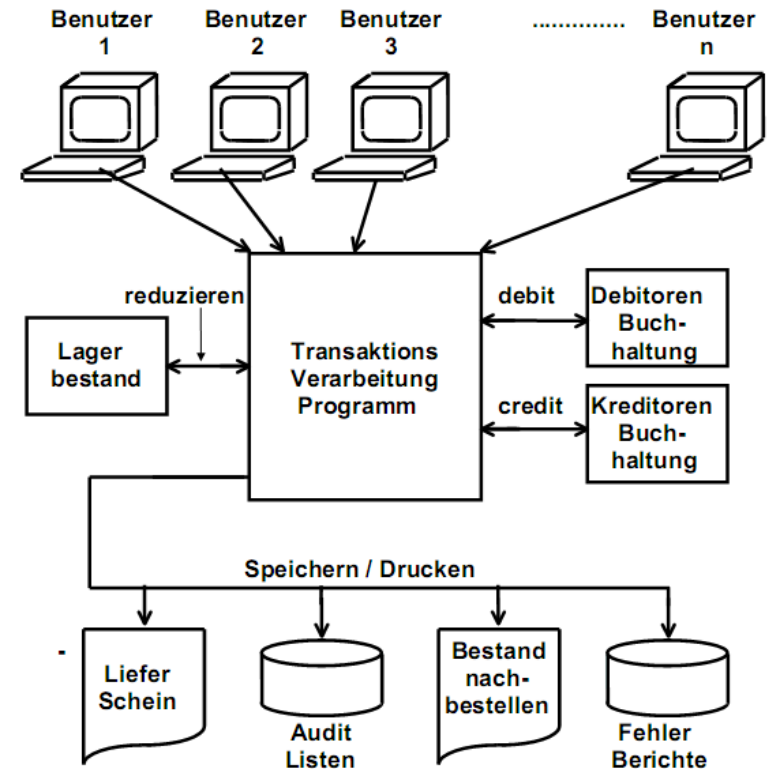
# weitere Subsysteme und Features

## Sysplex mit CF etc. – Performance



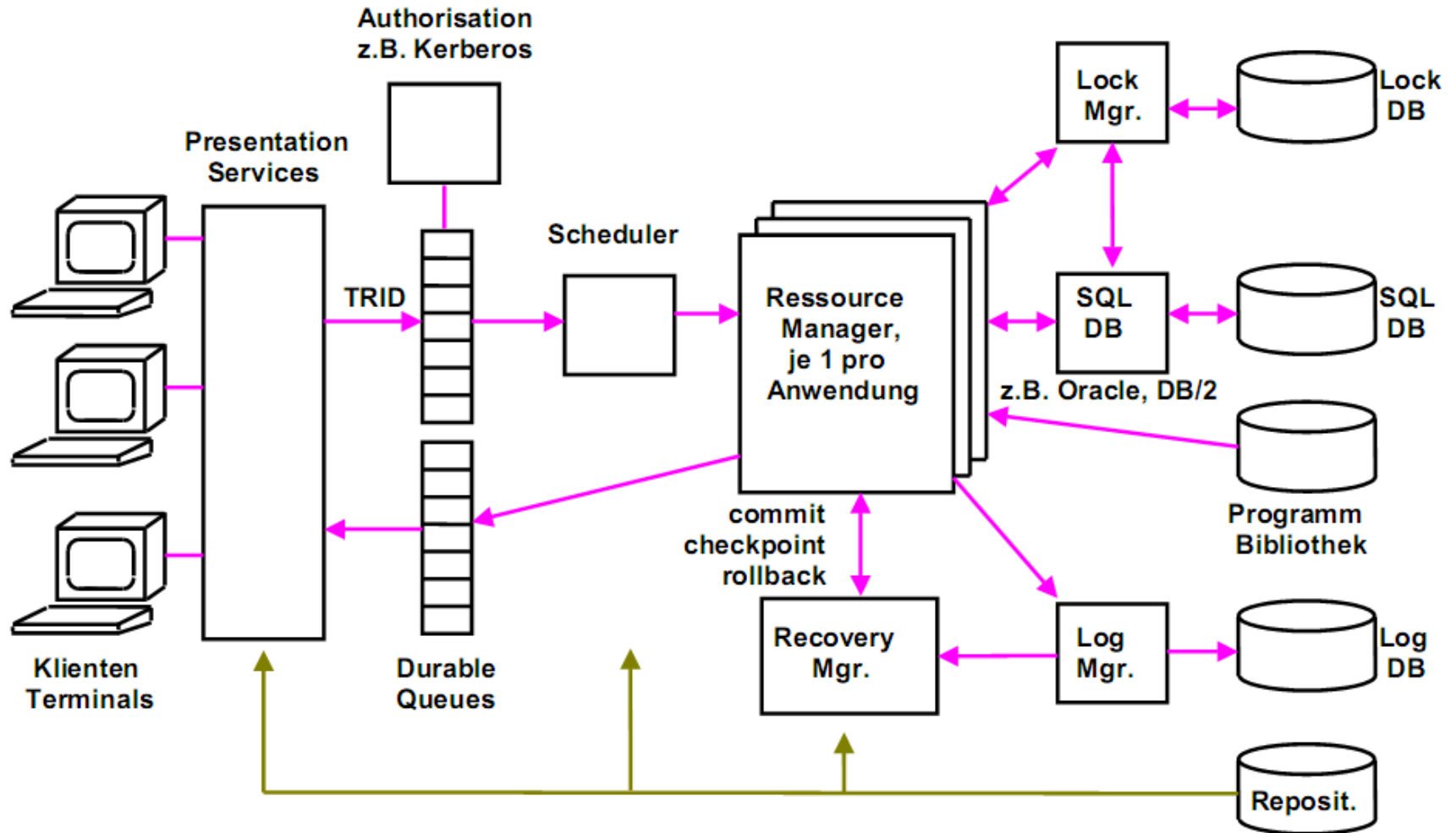
## CICS – Transaktionsmonitor (Customer Information Control System)

- ACID-Eigenschaften
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Kernfunktionen
  - Message Queueing
  - Lock-Verwaltung
  - Log-Verwaltung
  - 2-Phase-Commit / Rollback
  - Load Balancing



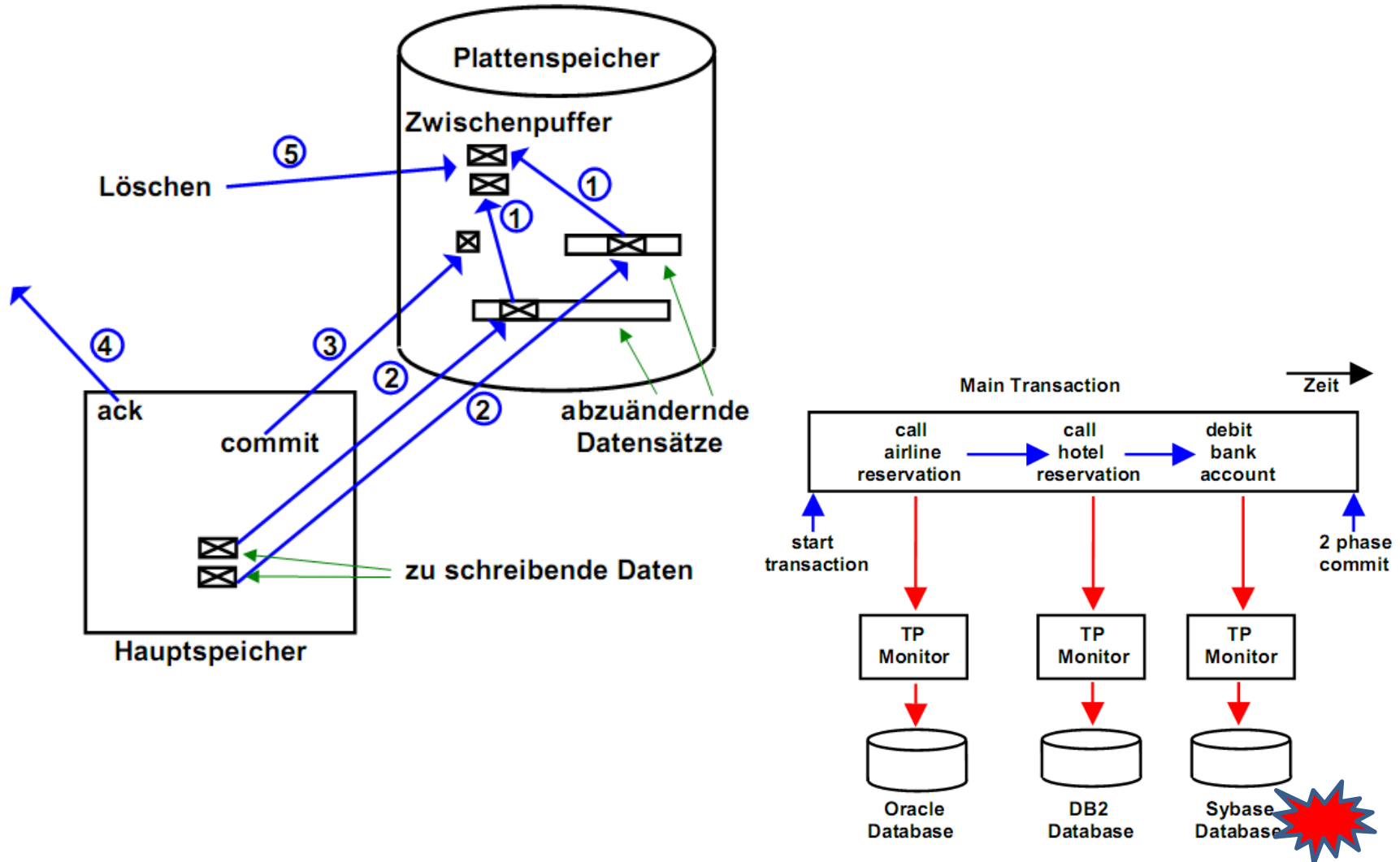
# weitere Subsysteme und Features

## CICS – prinzipielle Struktur



# weitere Subsysteme und Features

## CICS – Backward Recovery und 2-Phase-Commit



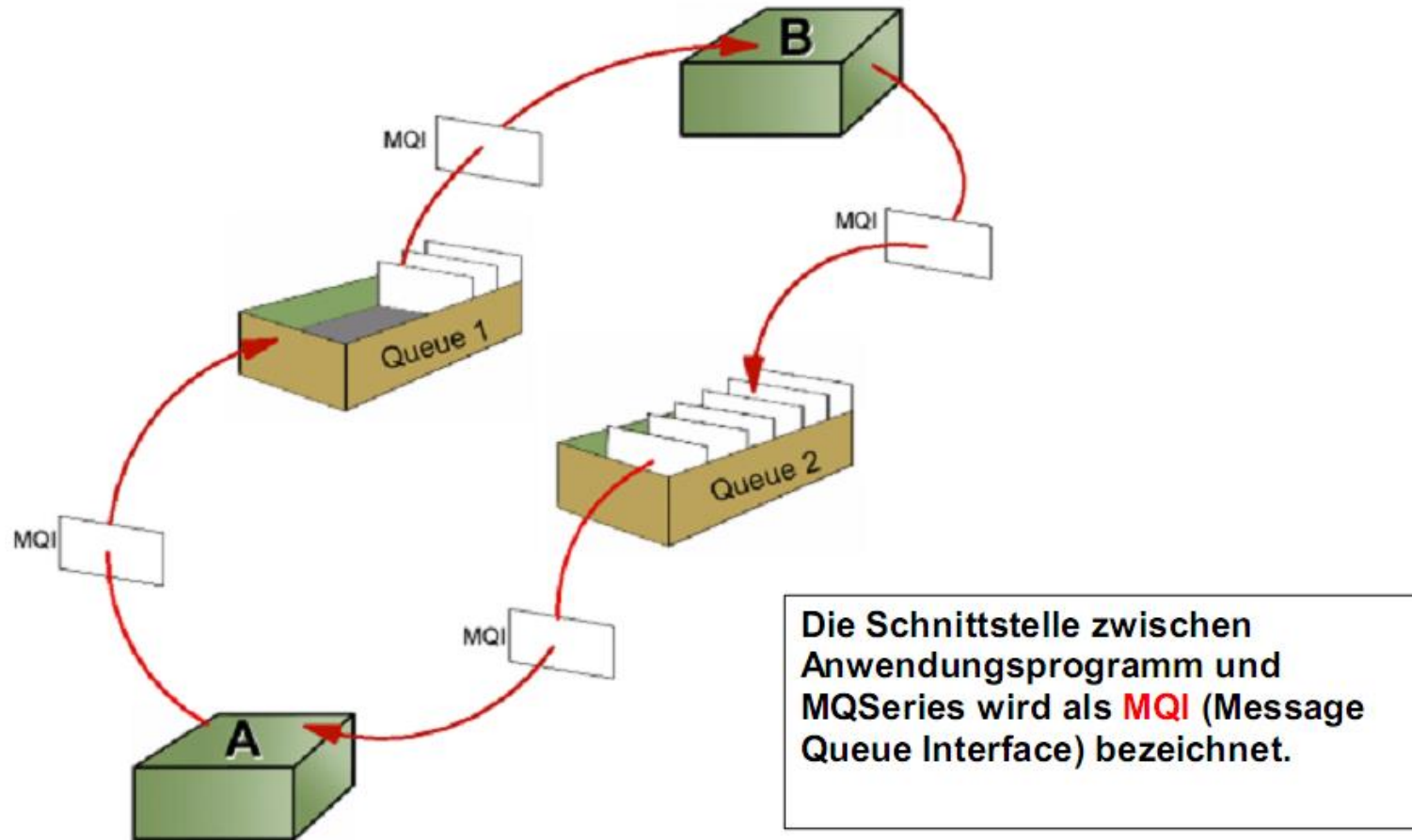
## MQSeries

---

- Eigenschaften
  - Nachrichten werden bis zur endgültigen Auslieferung an die Zielanwendung in Warteschlangen zwischengespeichert
  - Asynchron (im Gegensatz zum RPC, Store-and-Foreward Prinzip)
  - Recovery Mechanismen beim Versagen von Knoten oder Verbindungen
  - Auslieferung wird garantiert
  - Message Tracking (lokale Platte, entfernte Platte, Annahme der Nachricht durch Anwendung)
  - Steuerung durch Queue Manager auf jedem Rechner



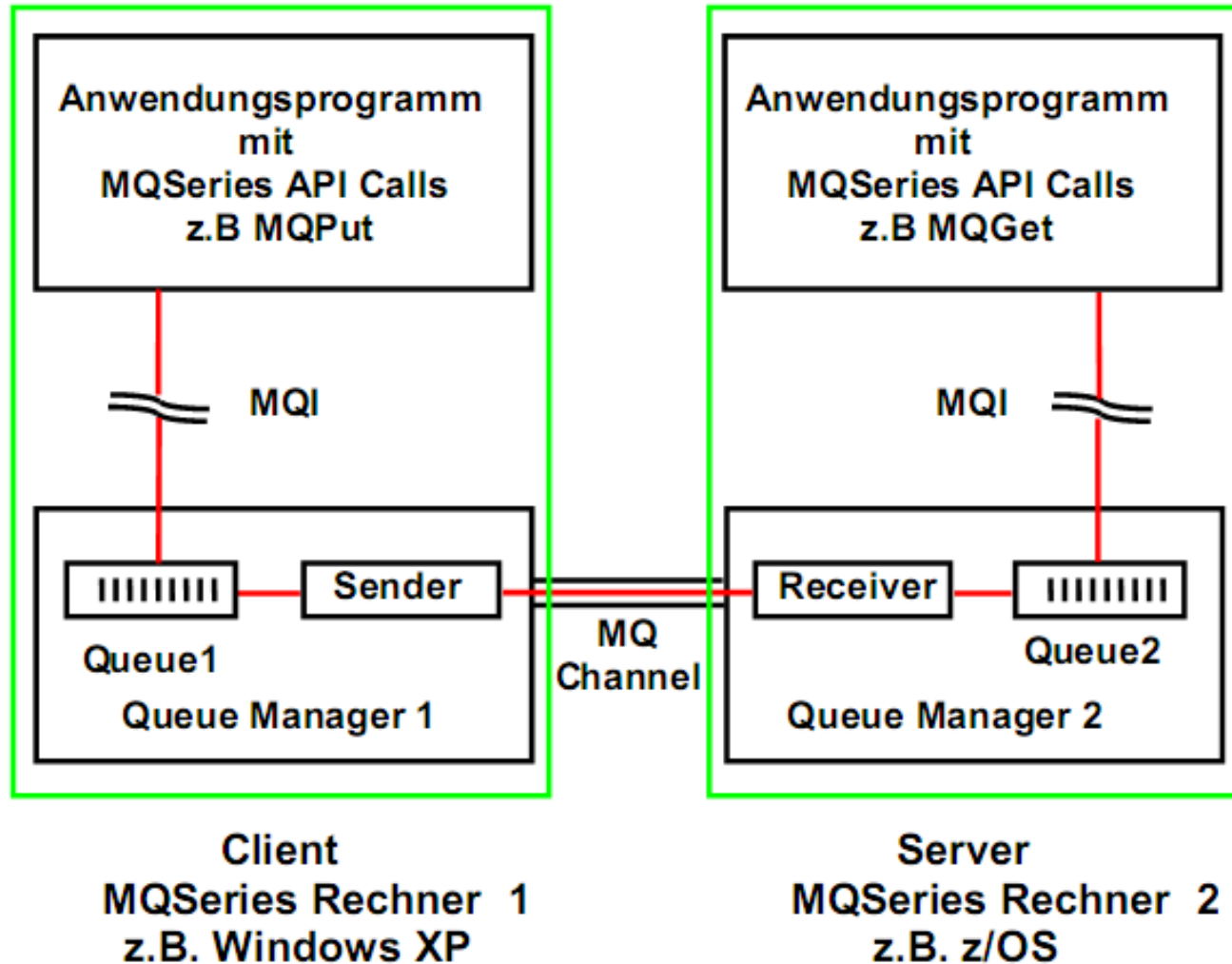
## MQSeries – Prinzip



## MQSeries

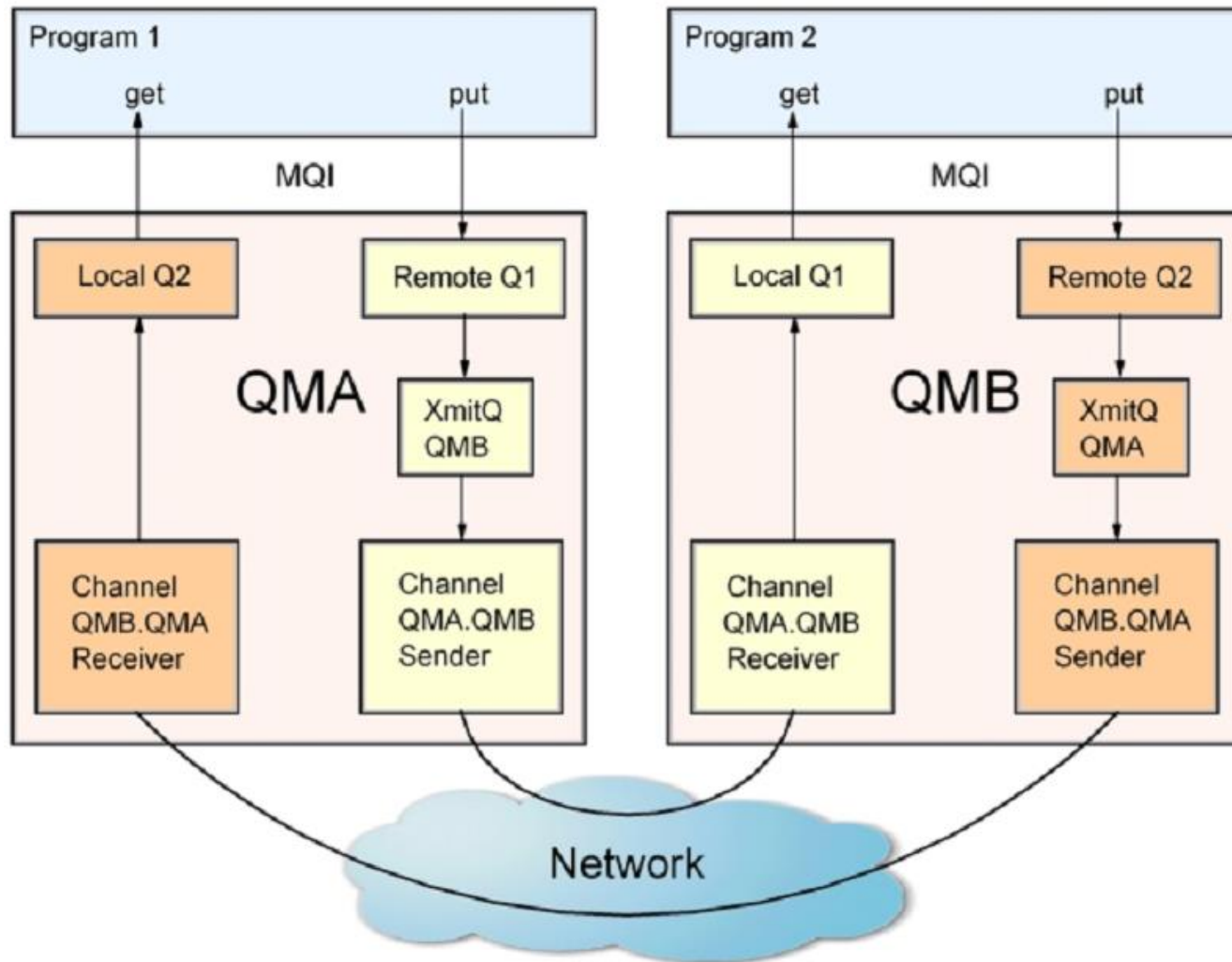
# weitere Subsysteme und Features

## MQSeries – auf vielen Plattformen verfügbar



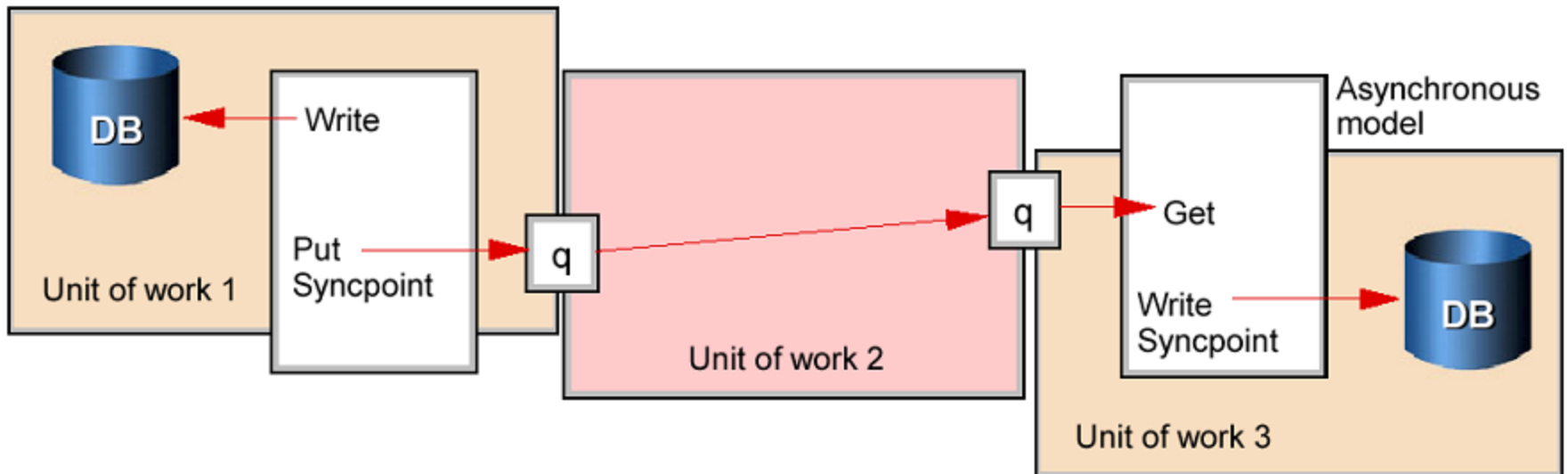
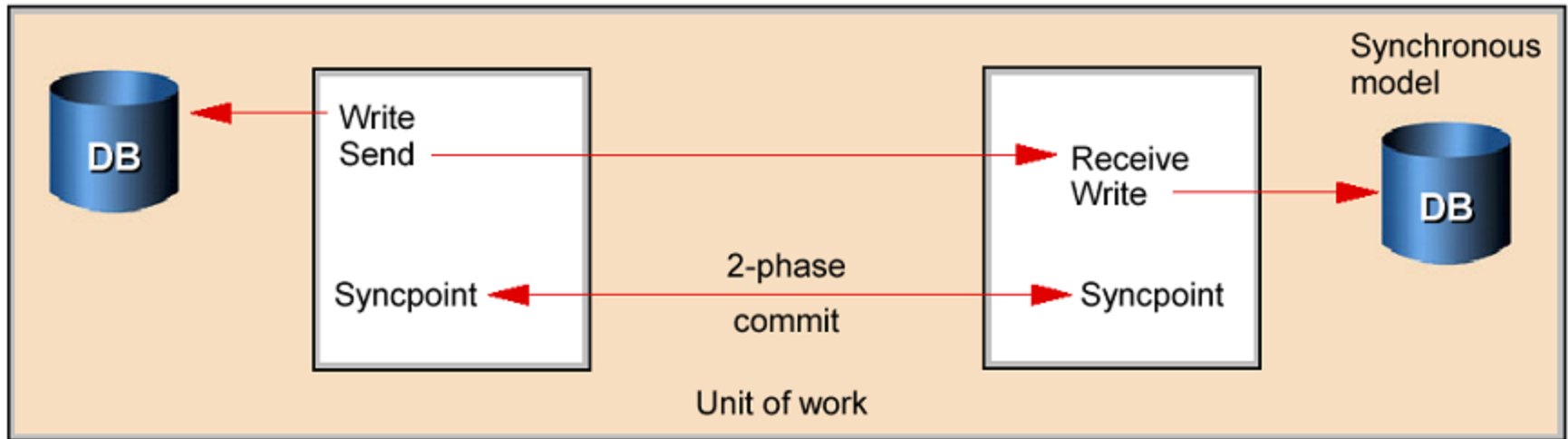
# weitere Subsysteme und Features

## MQSeries – über Netzwerke hinweg (auch Internet!)



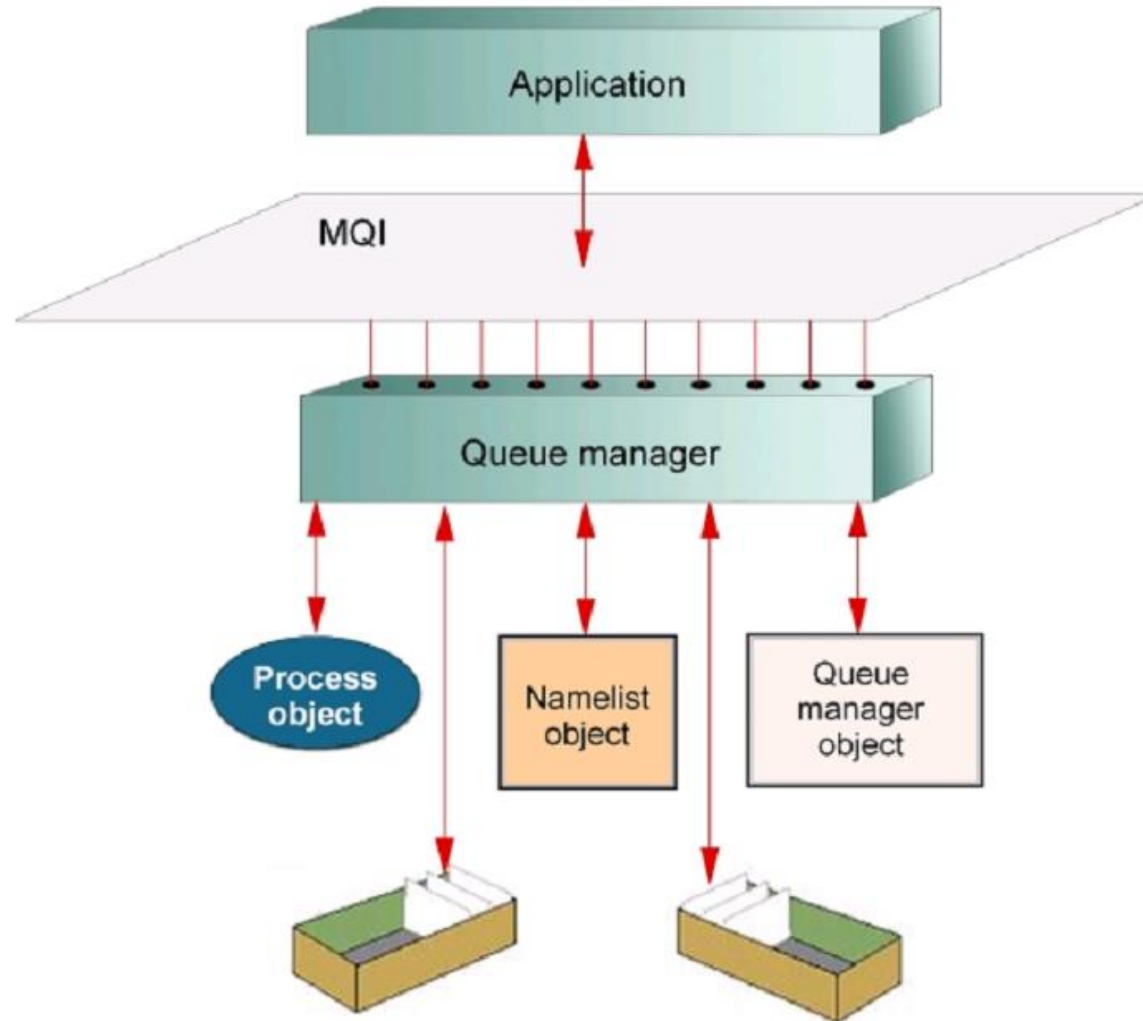
# weitere Subsysteme und Features

## MQSeries – Arbeitsweise



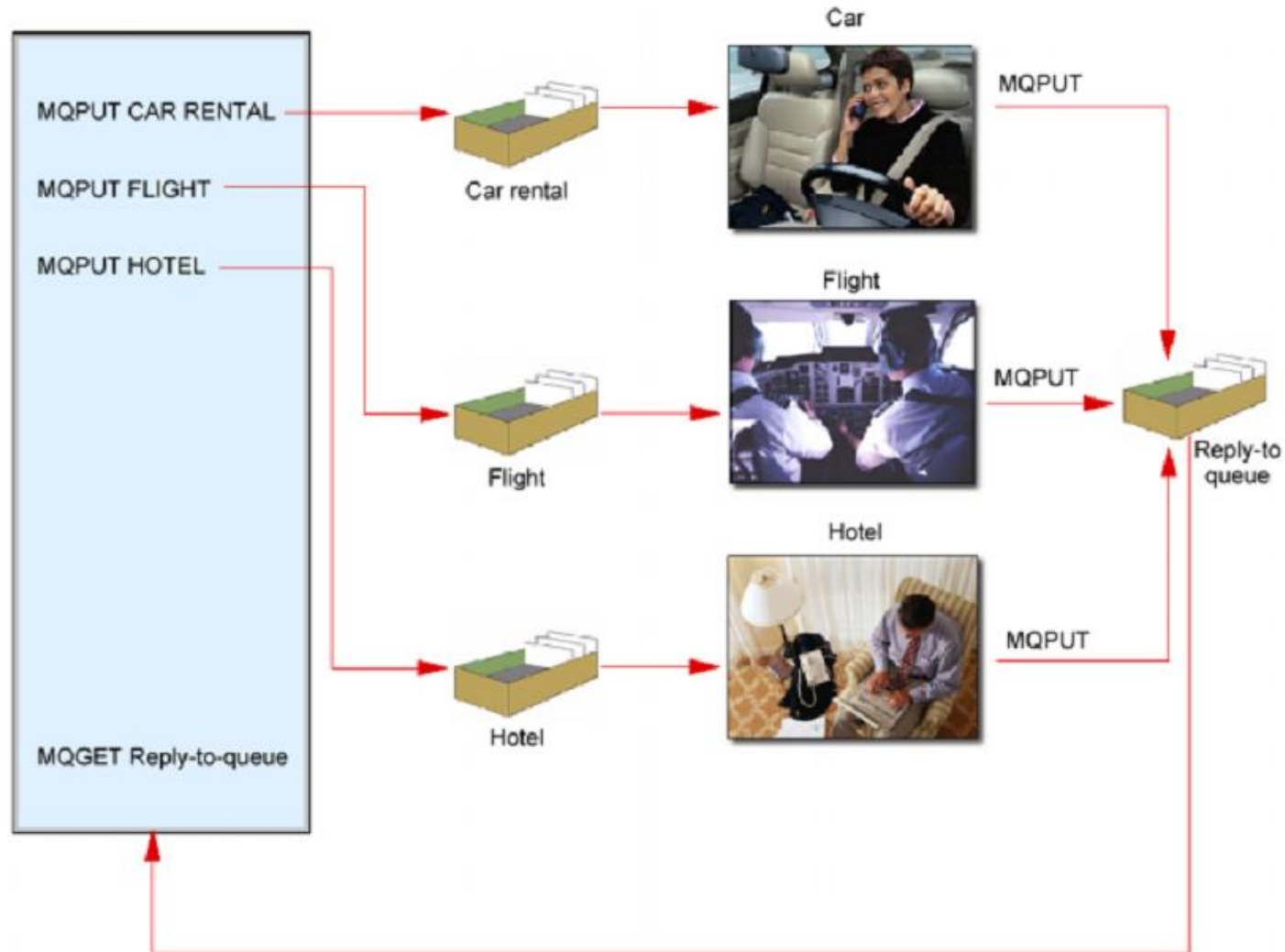
## MQSeries – Kommunikation mit Anwendung

---



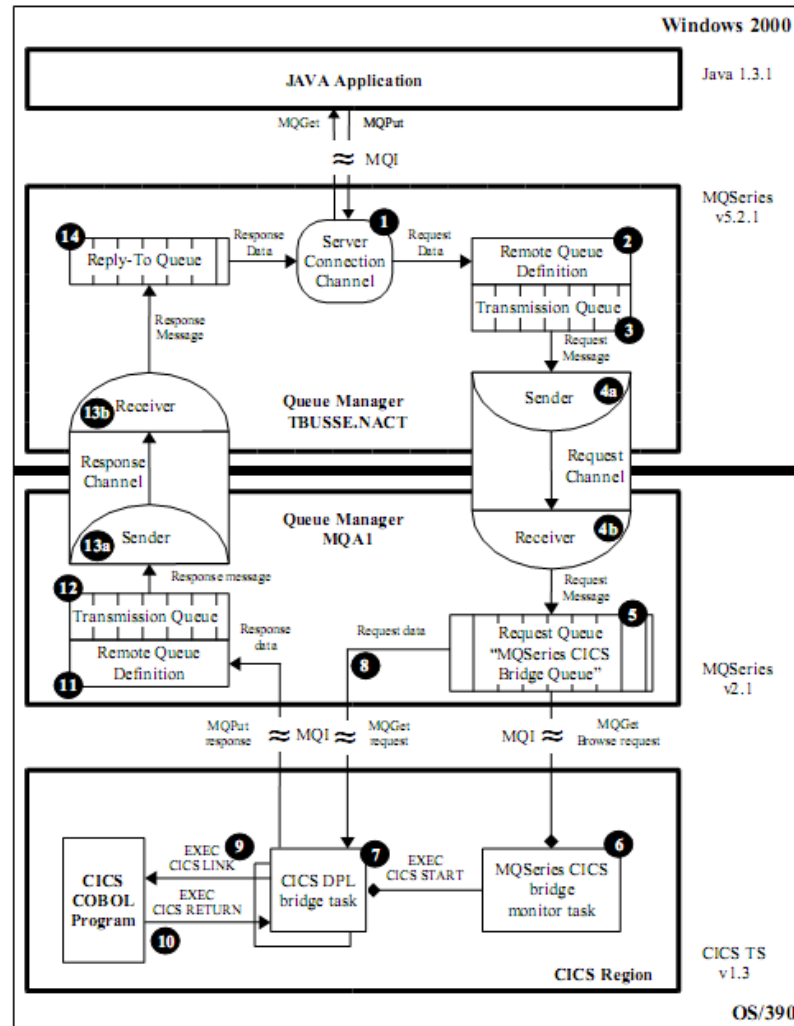
# weitere Subsysteme und Features

## MQSeries – Beispiel im Prinzip



# weitere Subsysteme und Features

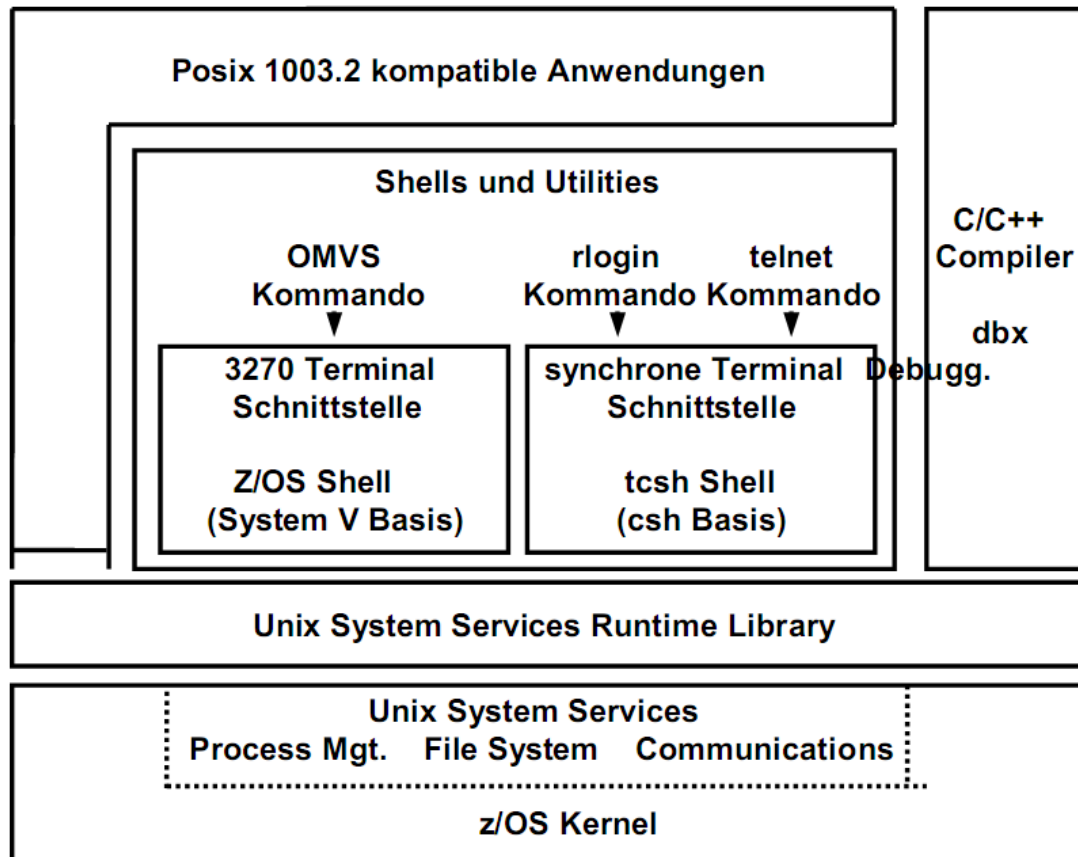
## MQSeries – oder ...



# weitere Subsysteme und Features

## USS – Unix System Services

- eigener virtueller Adressraum

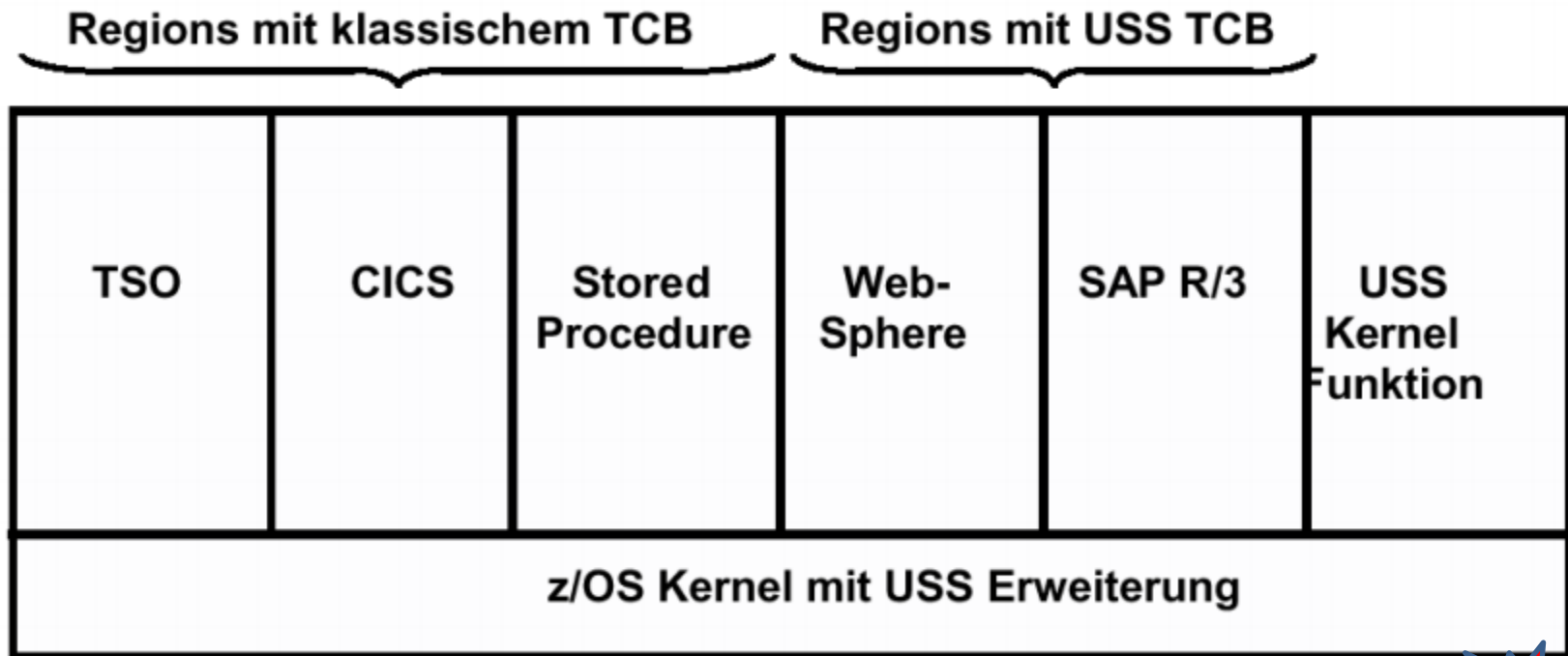




# weitere Subsysteme und Features

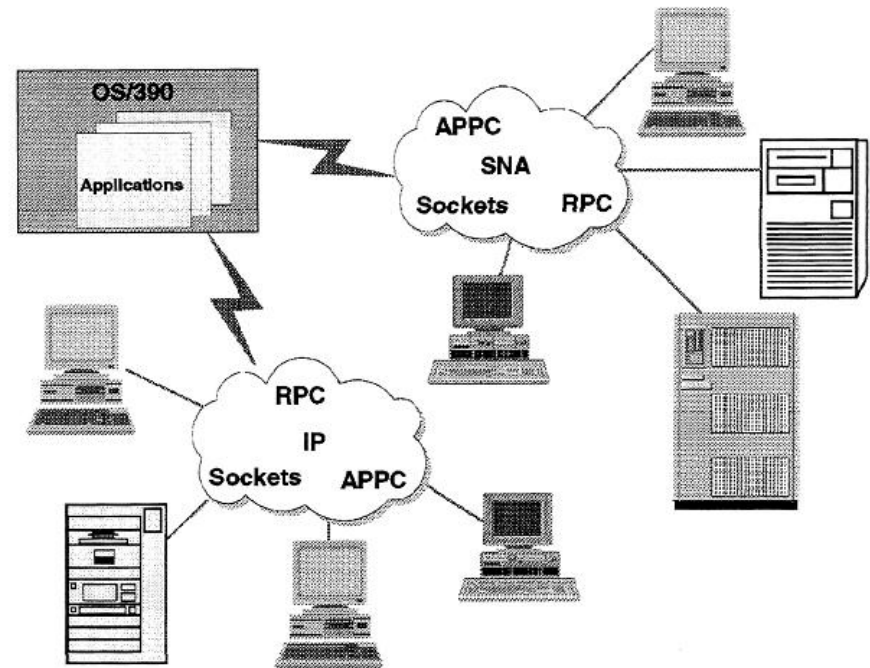
## USS – Unix System Services - Umgebung

- unterschiedliche TCBs (task control block)
- Kommunikation über den z/OS Kernel



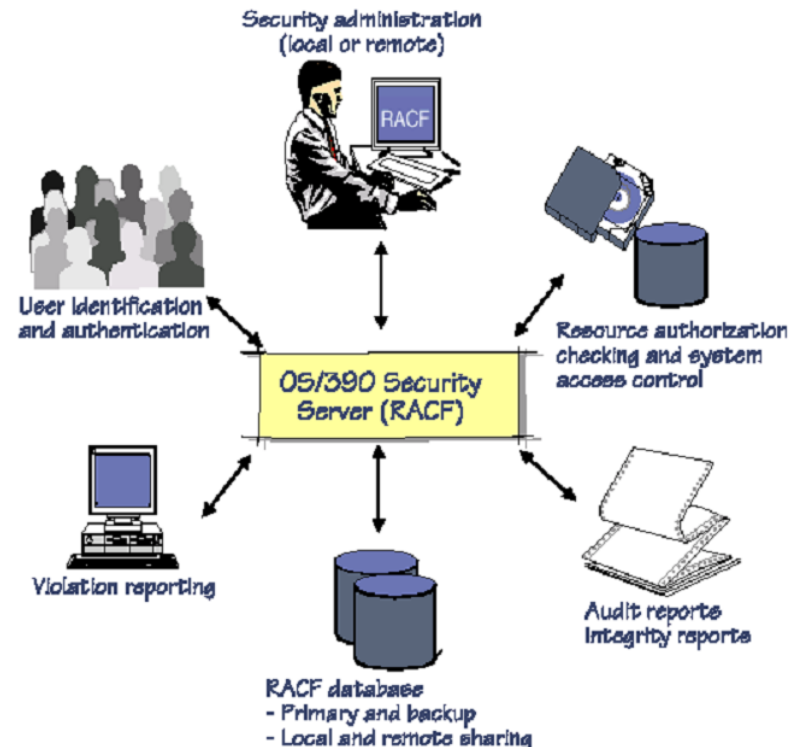
## z/OS Communication Server

- eigenständiges Subsystem
- eigener virtueller Adressraum
- implementiert TCP/IP, SNA Netzwerk Stacks



## z/OS „Secure Way“ Security Server

- LDAP-Server – Secure Directory Server
- Kerberos Network Authentication Service
  - Kryptographie
    - eigene Hardwarebefehle
    - Krypto-Prozessoren
  - Firewall Unterstützung
- RACF



## virtuelle Maschinen – z/VM

---

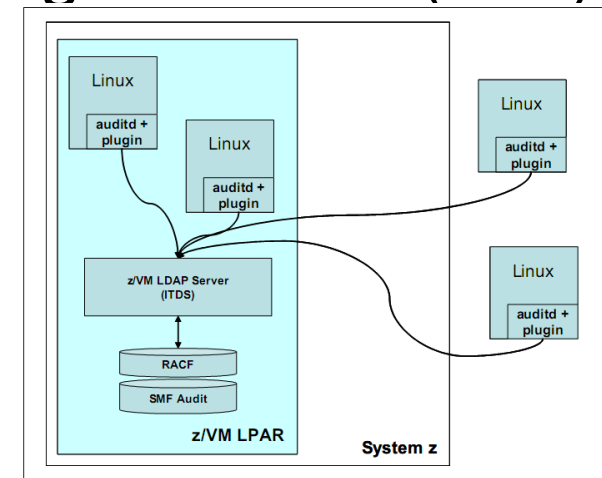
- Wurzel aus 50-er Jahre
- Vorbild für heutige PC-basierte VM
- beliebig viele VM erzeugbar
- Begrenzung nur durch Ressourcen
- Fokus vor allem z/Linux – LDAP-Server



## z/Linux – 1



- Beginn in 90-er Jahren
- alles, was unter Linux geht, geht unter z/Linux
  - Anwendungen portierbar
  - Anwendungen skalierbar
  - kostenlose Tools
- direkte und schnelle Verbindung zu allen z(/OS)-Ressourcen

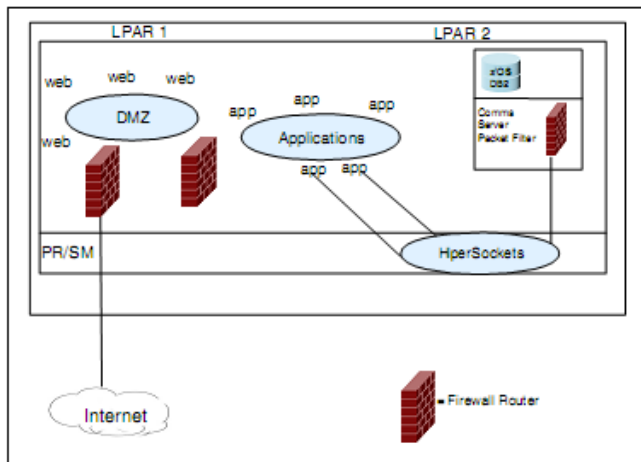
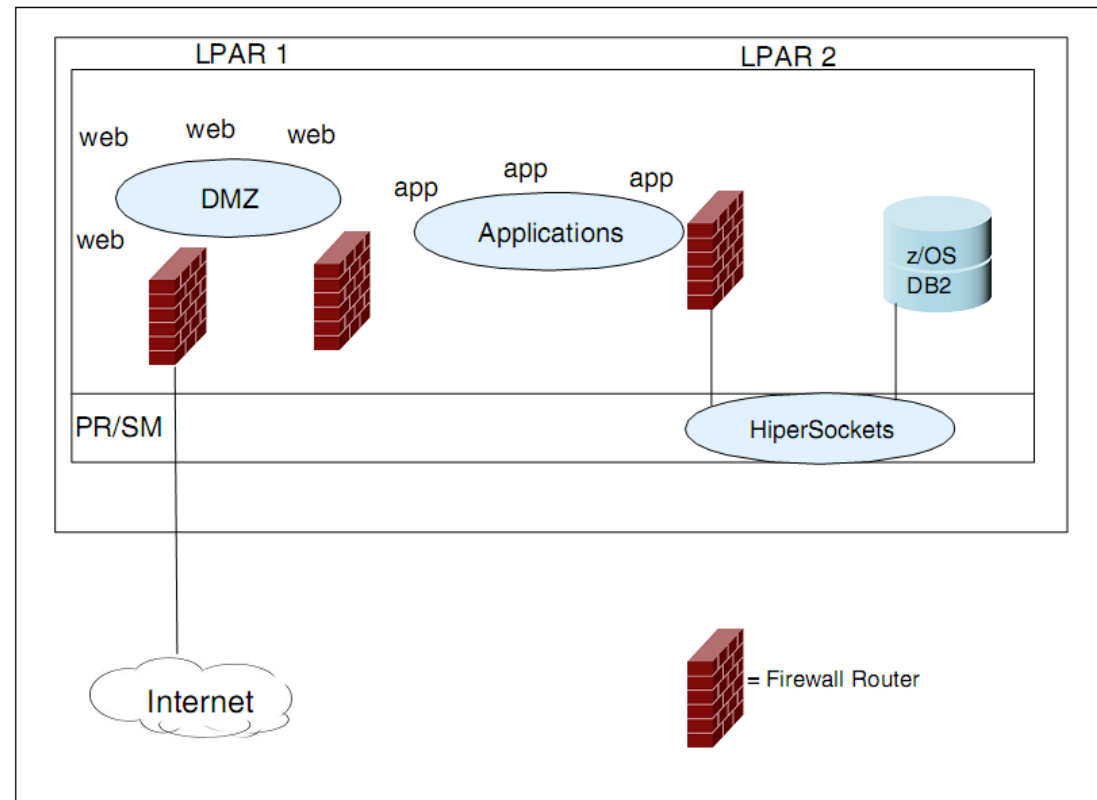
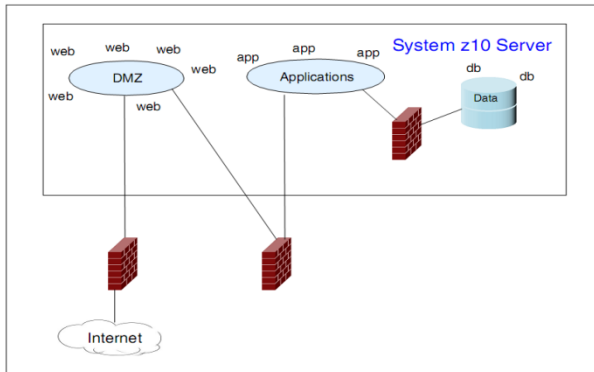


# weitere Subsysteme und Features

## z/Linux – 2



- Multizonenkonzept / Hypersockets

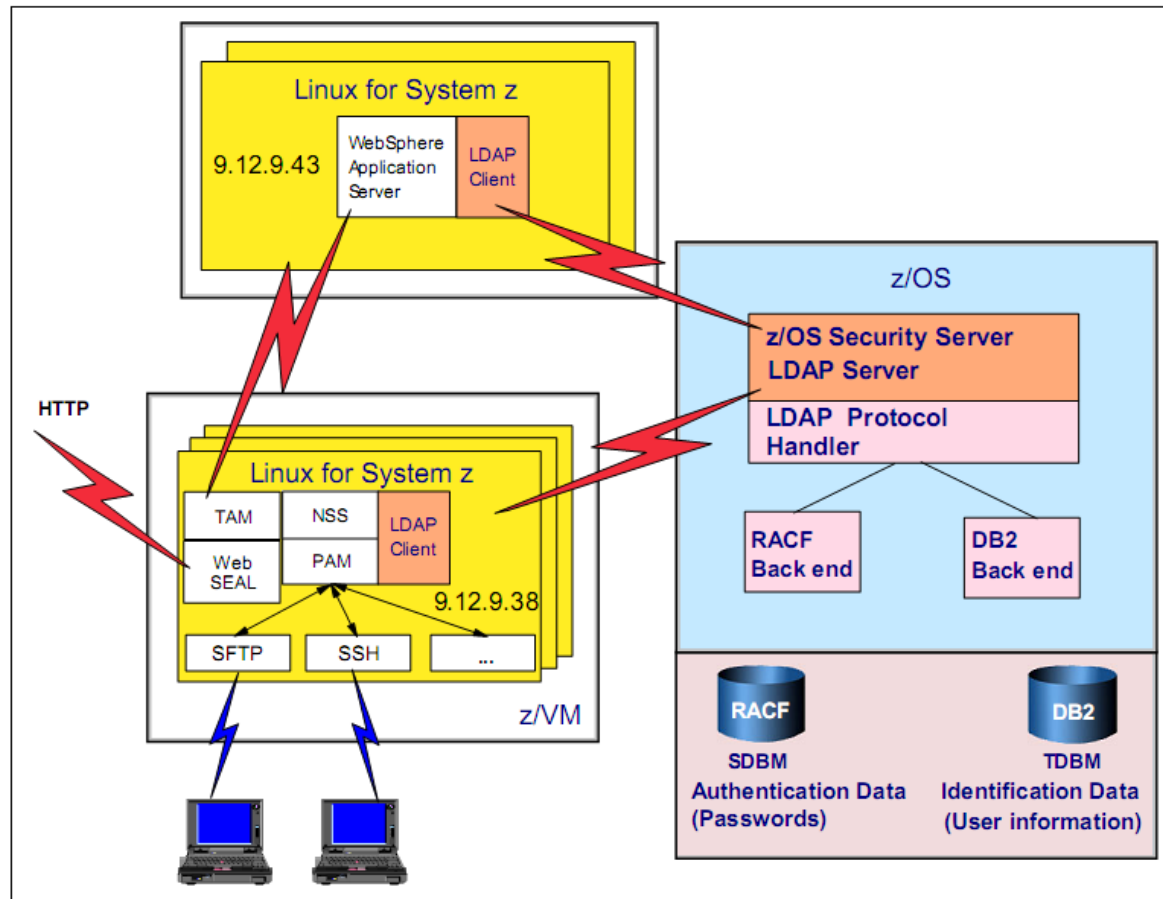


# weitere Subsysteme und Features

## z/Linux – 3

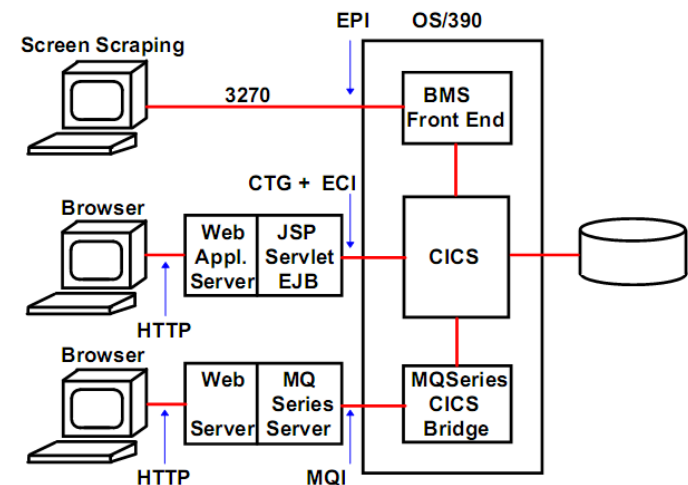


- Rechte



## Web Application Server

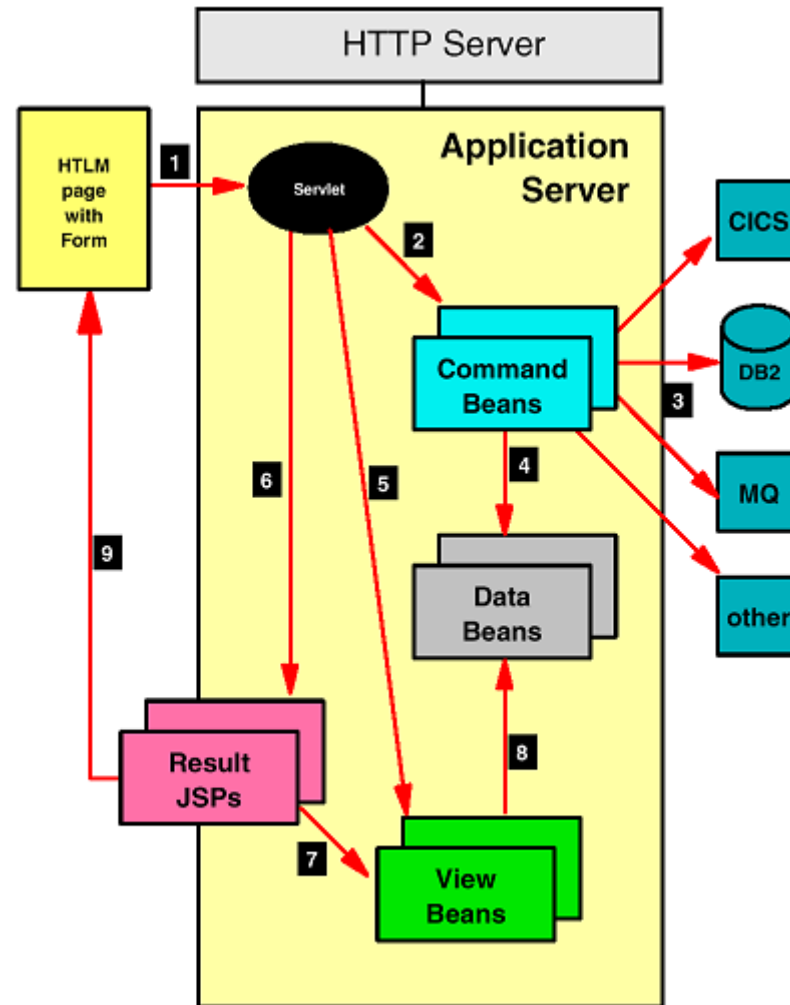
- (mögliche) Schnittstelle zwischen Dinosaurier und der „bunten“ Welt
- http-Server
- Servlet-Engine mit Java Servlets
- Datenbank-Server via JDBC





# weitere Subsysteme und Features

## Web Application Server – mögliche Architektur



und ...

---

- Java-Konnektoren in Subsystemen
  - Konnekten an Subsysteme mit TCP/IP-Adressen
  - etc...
- 
- unter der vollen Kontrolle des z/OS
  - unter der vollen Kontrolle der Security-Server
  - unter der vollen Kontrolle von wenigen Admin
  - unter der vollen Kontrolle

- Anwendungsentwicklungsumgebung auf Client
- direkte Kommunikation mit z/OS
- Basis Eclipse
- viele Plugins möglich
  
- Bitte nicht vom Namen irritieren lassen. ;-)

# IBM Rational Developer for System z V7.5

## JES and PD Tools

- Read/Write/Update VSAM datasets via integration with IBM File Manager
- Access IBM Fault analyzer reports for analyzing ABENDS and associating back to source code
- Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output
- Debug zOS applications from workstation as they execute live in the remote runtime

## Integration with EGL using RBD

- Quick and easy development of modern enterprise applications for procedural programmers
- Simplify and speed up creation of Web applications and services without having to learn Java or J2EE

## Traditional Development

### Development Environment

- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/1, C, C++, JCL, etc.
- Perform dataset management actions like allocating datasets and migrating datasets
- Perform typical edit, compile, and debug tasks on remote z/OS resources from the workstation
- Create, build, and catalog DB2 stored procedures on zOS
- Compile and test programs locally to ensure correctness

### Screen design

- Visually create, modify, build, and deploy BMS maps sets or MFS/IMS maps remotely or on the local workstation

### Code Generation

- Generate CRUD DB2 program code from UML, which can also be easily integrated into web service applications

## IBM Rational Developer for System z

### Host Tooling Integration

[JES, FA, FM, Debug Tool]

### zOS Application Development

[COBOL, PL/1, C/C++, JCL, Screens, Stored Procedures, etc]

### Enterprise Service Tools

[Web Services For CICS/IMS]

RBD

### Host / Distributed SCM Integration

### IBM Rational Application Developer

## zOS Web Service and Flow Creation

- Implements SOA and Web Services
- SOA access to CICS V3.2 and IMS V10 COBOL, PLI applications
- Bottom-up/Top-down or meet-in-the-middle COBOL, PLI to XML mapping support
- Integrated COBOL, PLI XML converters, XML schemas, and WSDL generation
- Service Flow Modeler to build/deploy service flows out of your existing Commarea, Channel, MQ, and Terminal CICS applications.

## SCM Support

- Access to host SCMs such as SCLM
- Framework for writing/deploying custom SCM integration code
- Support for storing zOS resources in distributed SCMs such as ClearCase

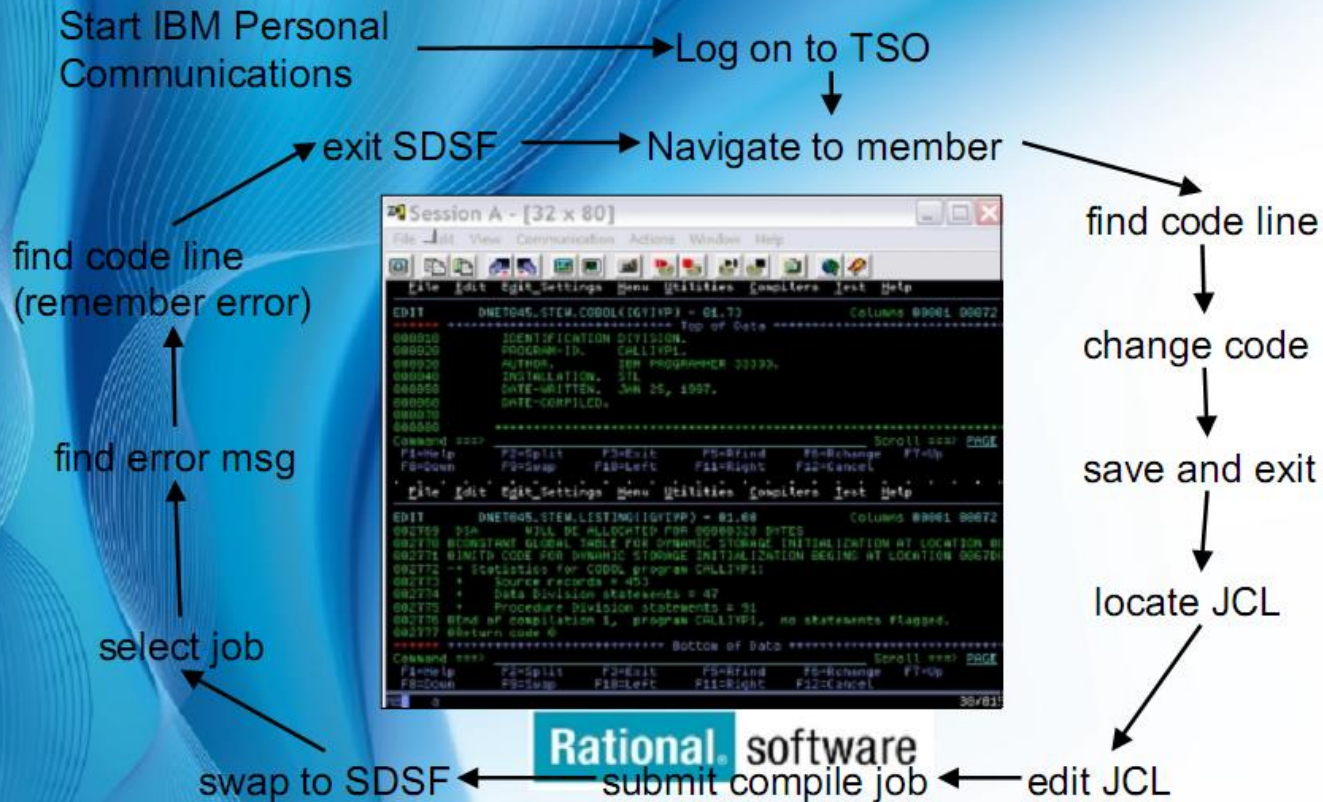
## Web and JEE Development

- Create Web Pages / JSF / Struts
- JEE/Java Development
- JCA Connectors
- Distributed debugger
- Web Services and Test environment

# weitere Subsysteme und Features

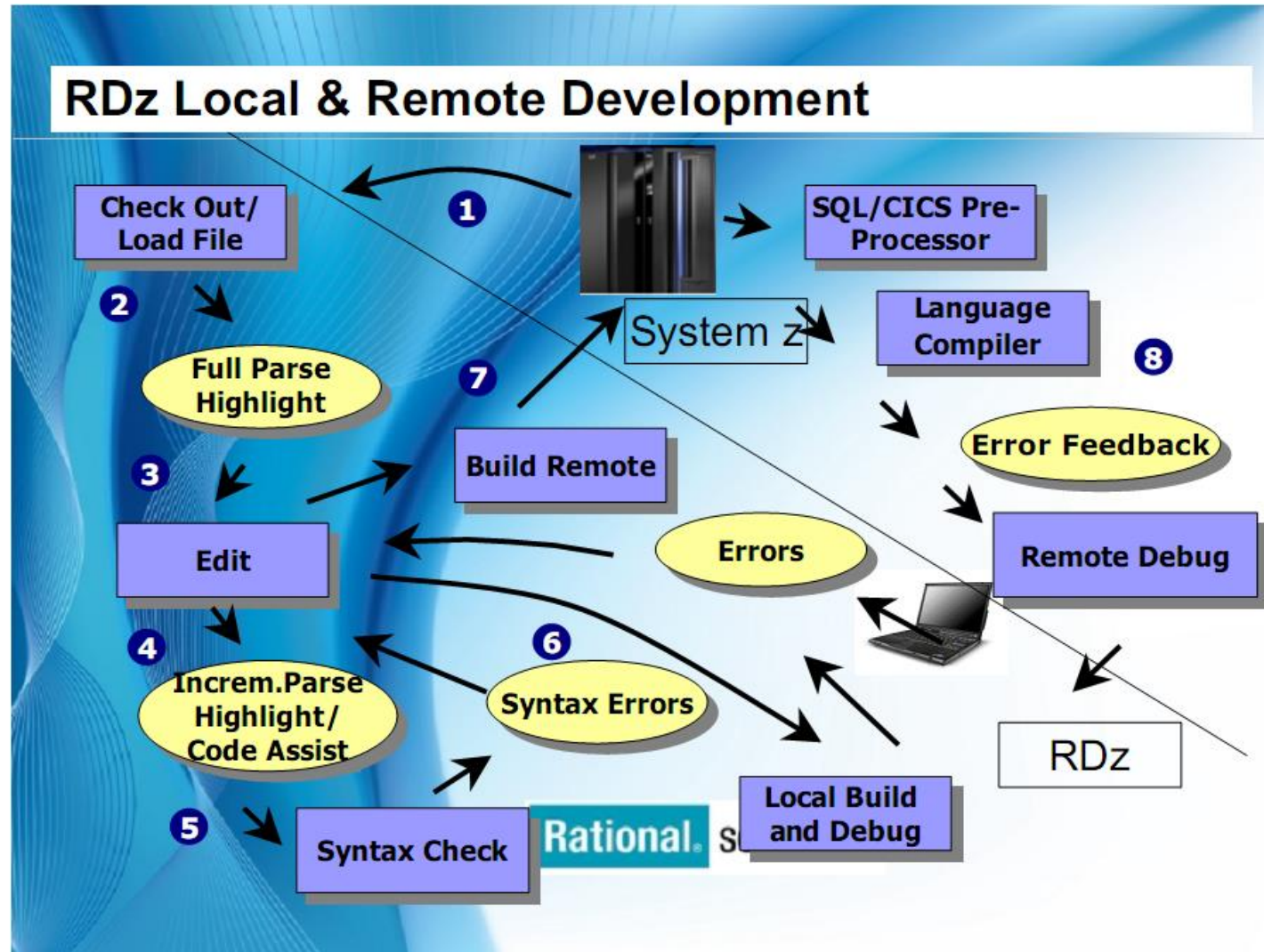
## RD/z / IDz – heutige Programmentwicklung in z/OS

### Remote development – so far



# weitere Subsysteme und Features

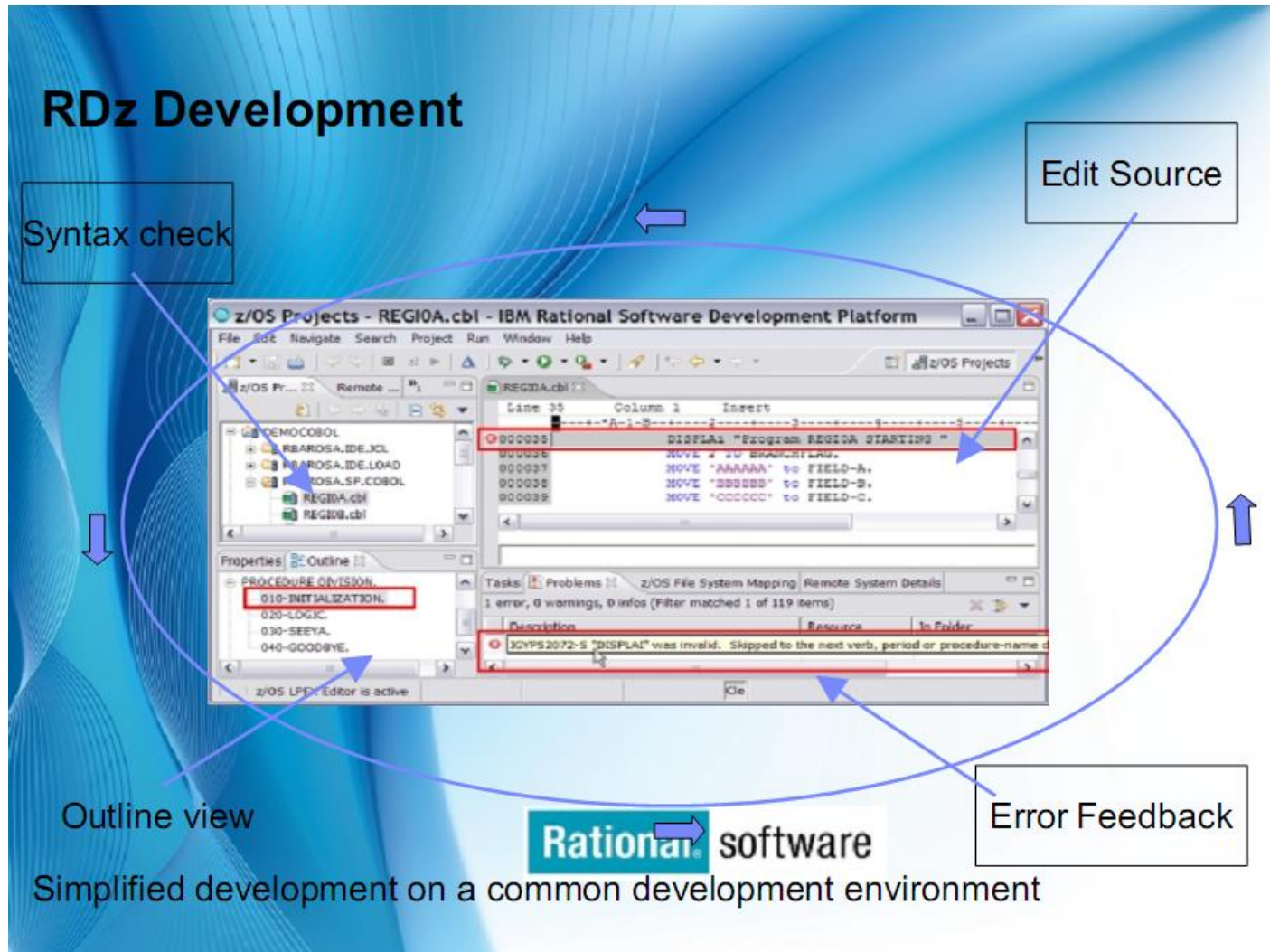
## RD/z – moderne Programmmentwicklung für z/OS



# weitere Subsysteme und Features

## RD/z – wie es aussieht

### RDz Development



The screenshot displays the IBM Rational Software Development Platform interface for z/OS development. The main editor window shows a COBOL program with the following code:

```
000005 DISPLAY "Program REGIOA STARTING "  
000006 MOVE * TO BRANCHFLAG.  
000007 MOVE "AAAAA" TO FIELD-A.  
000008 MOVE "BBBBB" TO FIELD-B.  
000009 MOVE "CCCCC" TO FIELD-C.
```

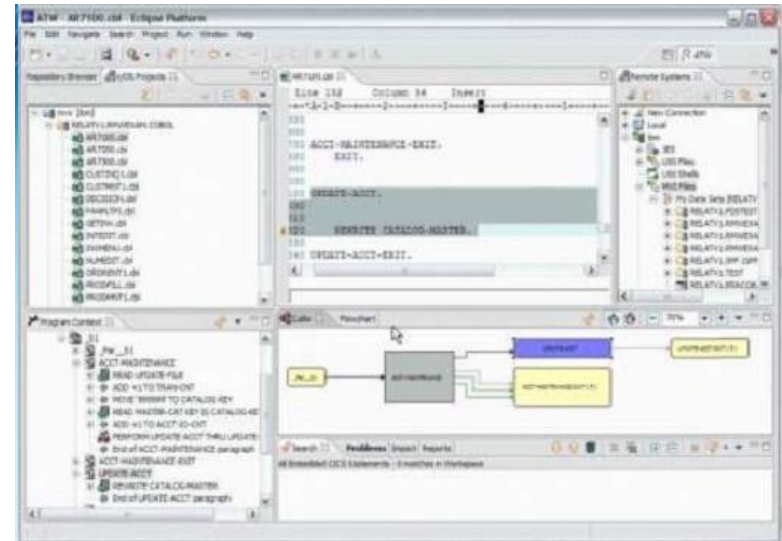
Annotations in the image include:

- Syntax check:** A box pointing to the top of the editor window.
- Edit Source:** A box pointing to the code in the editor.
- Outline view:** A box pointing to the left-hand pane showing a project tree with folders like DEMOCOBOL and REGIOA.CBL.
- Error Feedback:** A box pointing to the bottom pane showing a task description: "IGYPS2072-S 'DISPLAY' was invali... Skipped to the next verb, period or procedure-name d...".

At the bottom of the screenshot, the Rational software logo is visible, and the text reads: "Simplified development on a common development environment".

## RD/z – weitere Features – 1

- end-2-end-debugging mit IBM Debug-Tool
- end-2-end-debugging mit JPDA (Java Plattform Debugger Architecture)
- RTW Analyzer for Eclipse
  - Verbindungen
  - Visualisierung
  - Impacts erkennen
  - Suchmechanismen
- Integrierte Testumgebung

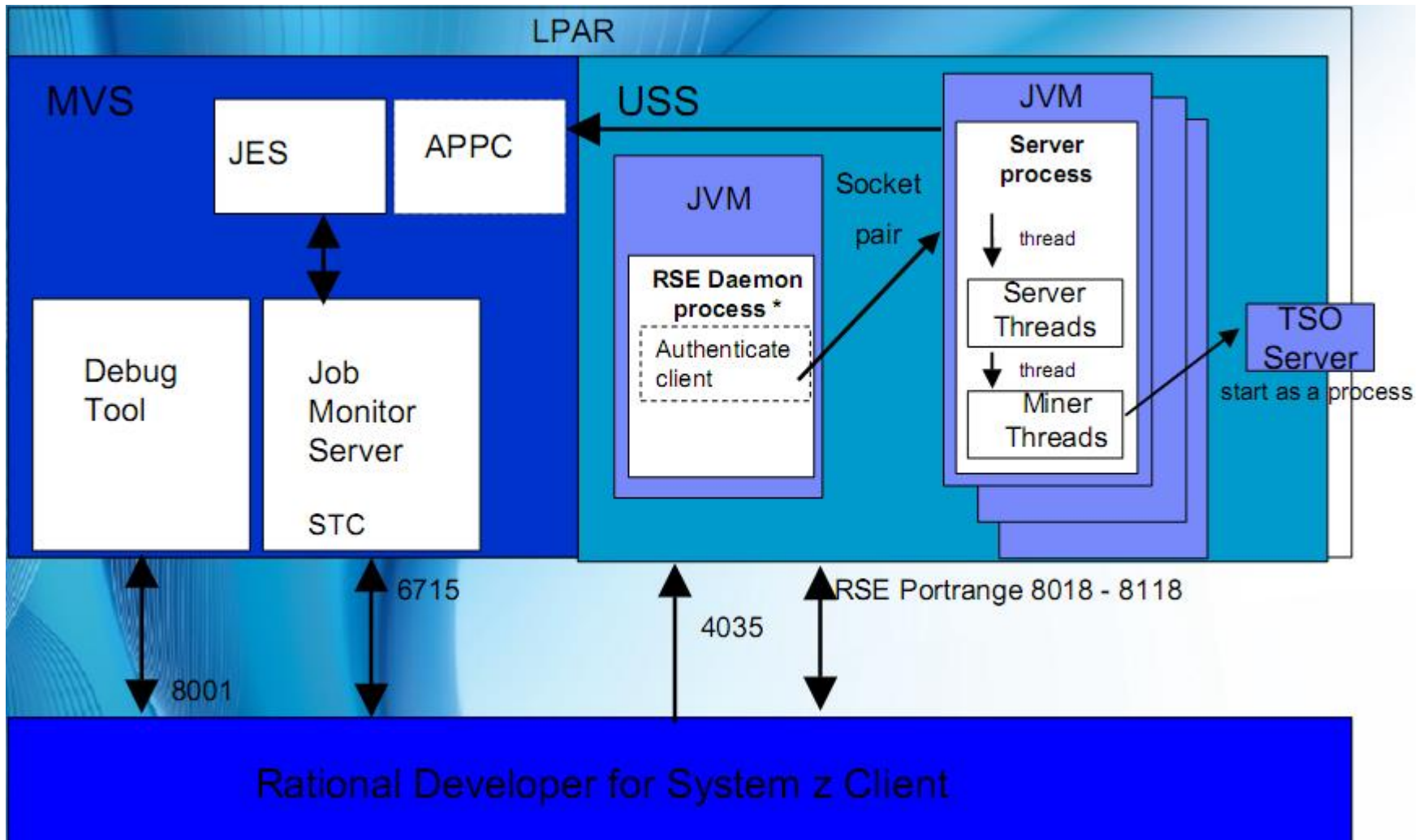




- Visualisierung IMS-DB
- CICS Explorer
- CICS Resource Definition Editor
- COBOL editieren
- UML nach COBOL
- Integration Fault Analyzer
- Integration File Manager
- Plugins für fast alles (auch SCLM ;-)

# weitere Subsysteme und Features

## RD/z – Einbindung



## einige wenige Zahlen zu zEC10

---

- fast 100% Verfügbarkeit
- 5,5 GHz
- > 78.000 Mio MIPS
- 3 TB Memory
- 32 GB HSA (Hardware System Area – z.B. I/O)
- CMOS-Prozessoren und Intel-Prozessoren
- bis 101 Prozessoren
  - unterschiedlich definierbar als zIIP, zAAP, IFL, CP, SAP
- „ein System der Superlative“



CP	Central Processors (CPs)
IFL	Integrated Facility for Linux
zAAP:	System z Application Assist Processors
zIIP	System z Integrated Information Processors
ICF	Internal Coupling Facilities
SAP	System Assist Processors

# weitere Subsysteme und Features

## einige wenige Zahlen zu z13

- 13. Generation CMOS (1994)
- fast 100% Verfügbarkeit
- ? 5,5 GHz
- > 111.000 Mio MIPS
- 10,0 TB Memory
- 64 GB HSA (Hardware System Area – z.B. I/O)
- CMOS-Prozessoren und Intel-Prozessoren
- bis 141 Prozessoren
  - unterschiedlich definierbar als zIIP, zAAP, IFL, CP, SAP

Model	Processor Drawers	Installed PUs	Standard SAPs	Maximum Characterized PUs	Min - Max Customer Memory (GB)
N30	1	39	6	30	64 - 2464
N63	2	78	12	63	64 - 5024
N96	3	117	18	96	64 - 7584
N129	4	156	24	129	64 - 10144
NE1	4	168	24	141	64 - 10144



CP Central Processors (CPs)  
IFL Integrated Facility for Linux  
zAAP: System z Application Assist Processors  
zIIP System z Integrated Information Processors  
ICF Internal Coupling Facilities  
SAP System Assist Processors  
IFP Integrated Firmware Processor

## einige wenige Zahlen zu z16

---

- leider kommt man nicht (mehr) auf diese Seiten der IBM
- Die IBM-ID muss speziell freigeschaltet werden.
- <https://www.ibm.com/de-de/products/z16>
- <https://www.ibm.com/docs/en/systems-hardware/z/Z16M-A01>

# weitere Subsysteme und Features

## mögliches Fazit



- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - • Begriffe ;-)

- OPC / TWS / IWS
- EJES
- Beta91 / Beta92
- Workplace
- SDF II
- Omegamon
- XINFO
- Repository
- Tabellensystem



- Cleanup
- DSS
- Mainview
- Strobe
- REXX / Clist
- Apache / Tomcat
- STC

- Container (zCX) / Docker
  - <https://www.ibm.com/support/z-content-solutions/container-extensions/>
  - <https://www.ibm.com/cloud/learn/docker>
- Blockchain
  - <https://www.ibm.com/blockchain>
- Cloud
  - <https://cloud.ibm.com/login>
  - <https://www.ibm.com/cloud>

- DevOps
  - <https://www.ibm.com/cloud/learn/devops-a-complete-guide>
- Encryption
  - <https://www.ibm.com/topics/encryption>
- AI / Machine Learning / Watson
  - <https://www.ibm.com/cloud/learn/machine-learning>
- Open Source
  - <https://www.ibm.com/opensource/>

- 
- Einführung
  - Entwicklung des Mainframes
  - Kommunikation mit dem Betriebssystem
  - Data / Program / Job Management
  - Funktionen des Betriebssystems
  - Data Facility und Datenbanken
  - weitere Subsysteme und Features
  - Begriffe ;-)

