

# SSM - Service & System-Management

## Stoffabgrenzung HS 2013

Kapitel 01 Einführung .....	2
Kapitel 02 Was ist IT? .....	2
Kapitel 03 Übersicht IT – Betriebsprozesse .....	3
Kapitel 04 IT Cloud Computing .....	5
Kapitel 05 ITIL & IT-Service Management .....	7
Kapitel 06 IT Service Level Management .....	8
Kapitel 07 SLA Fallstudie .....	9
Kapitel 08 Outsourcing & Offshoring .....	10
Kapitel 09 Vertiefung IT-Betriebsprozesse und ITIL V3 .....	13
Kapitel 10 Monitoring & Event Management .....	18
Kapitel 11 Data Center .....	21
Kapitel 12 Parallele Rechnerarchitekturen .....	24
Kapitel 13 Unix Systemsoftware .....	28
Kapitel 14 File Systeme .....	31
Kapitel 15 Cluster & Grid Systeme .....	33
Kapitel 17 Server Virtualisierung .....	36
Kapitel 19 Storage Systems .....	40

## Kapitel 01 Einführung

1. Nicht prüfungsrelevant.



## Kapitel 02 Was ist IT?

1. Sie können die grundlegenden Prozesse eines Unternehmens erläutern und in die drei Hauptkategorien einteilen.

- **Managementprozesse**

- Strategieplanungsprozess
- Qualitätsmanagementprozess
- Controllingprozess

- **Kernprozesse**

*Beschreiben die Erstellung und Verwertung der Leistungen, die den Zweck des Unternehmens darstellen. Fangen beim Kunden an und hören beim Kunden auf.*

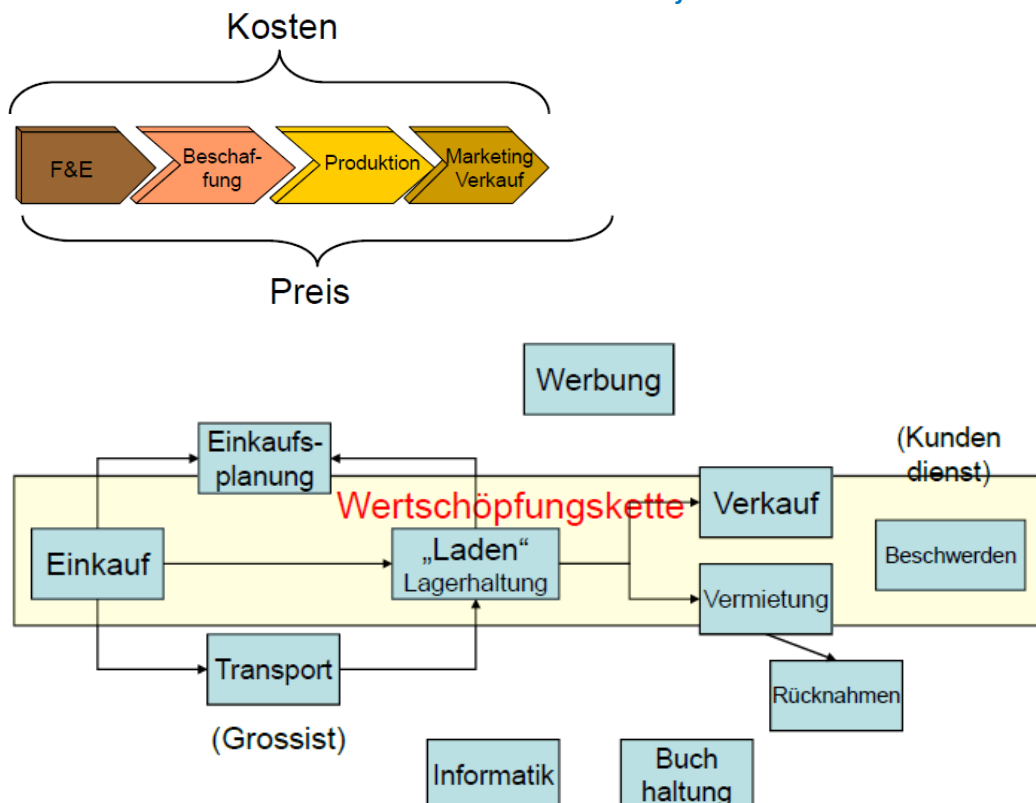
- Innovationsprozess
- Vertriebsprozess (Leistungsverwertung)
- Auftragsabwicklungsprozess (Leistungserstellung)
- Serviceprozess

- **Unterstützungsprozesse**

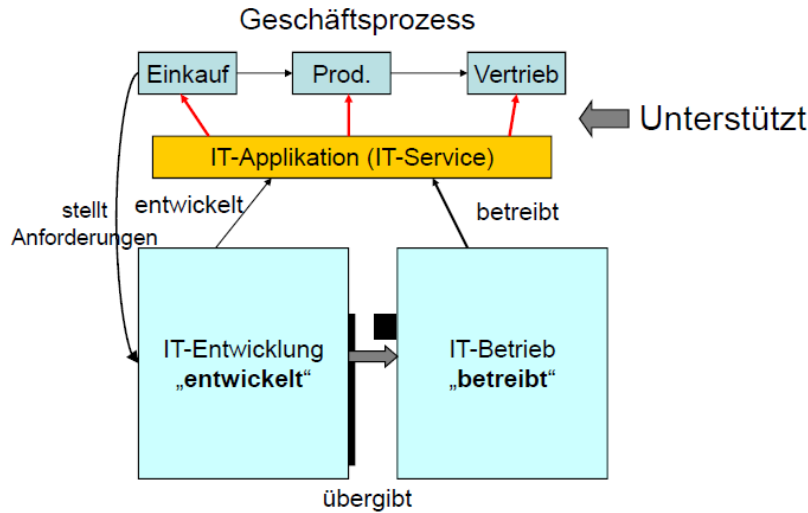
- Personal
- Informatik
- Finanz- und Rechnungswesen
- Einkauf
- Gebäudemanagement
- Qualitätsmanagement

2. Sie wissen, was Wertschöpfung ist und können eine Wertschöpfungskette erklären.

*Abfolge der Aktivitäten eines Betriebes, um marktfähige Güter zu erstellen und abzusetzen, deren **Verkaufswert höher ist als die Summe der Estandskosten aller Produktionsfaktoren.***



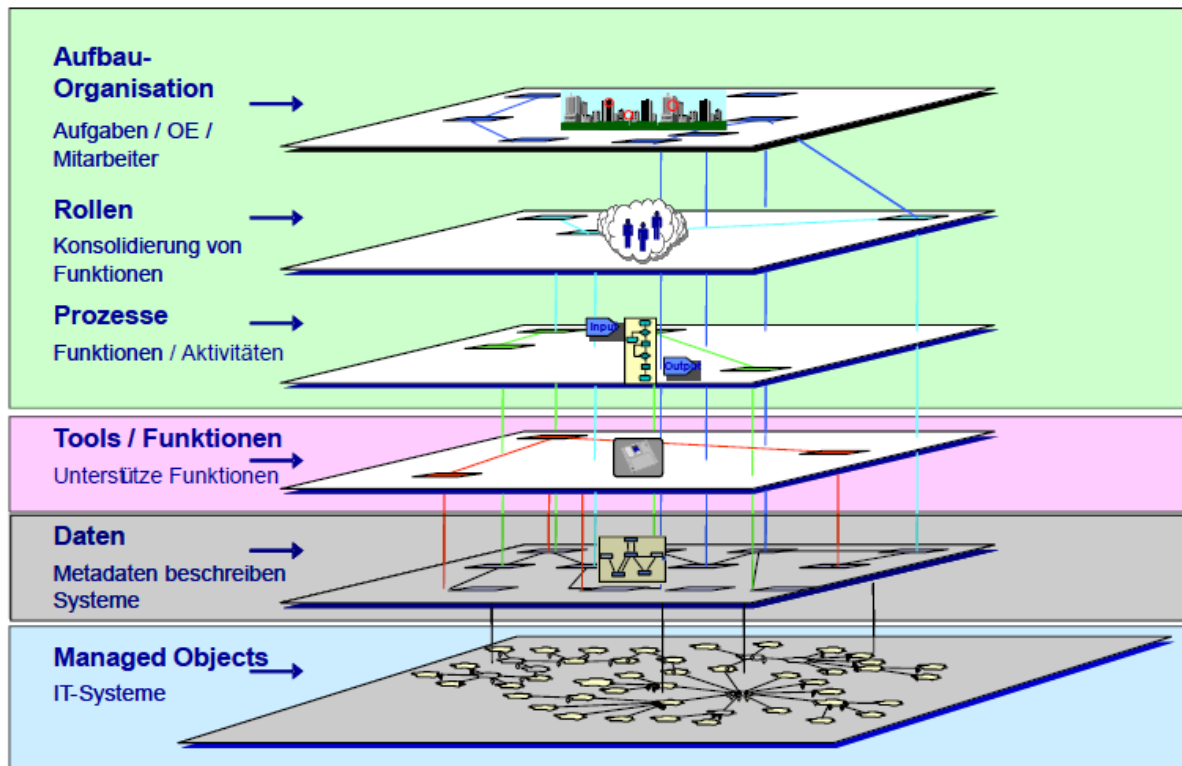
3. Sie kennen den Zweck der IT im Unternehmen und kennen deren Hauptaufgaben.

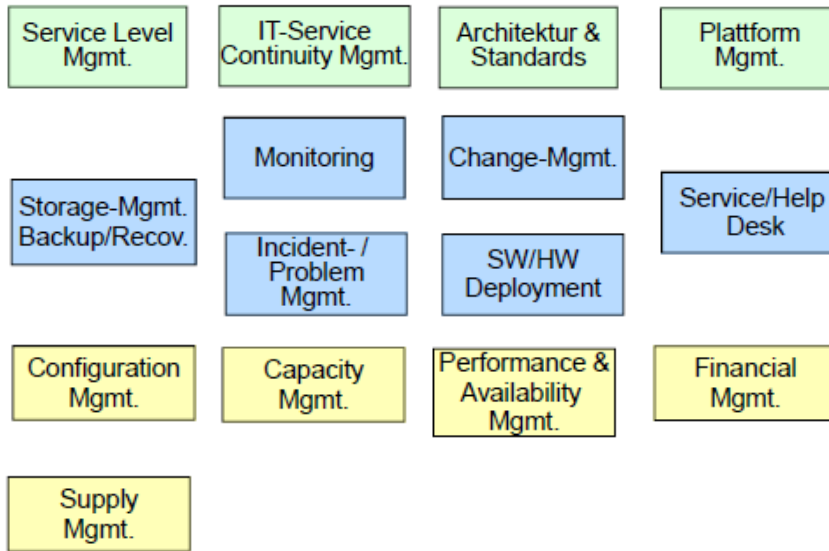


- **Hauptaufgabe der IT-Abteilung**
  - Unterstützung von Geschäftsprozessen Entwicklung & Betrieb von IT-Applikationen
- **Hauptaufgabe der IT-Entwicklung**
  - Entwicklung und Unterhalt von IT-Applikationen im Rahmen von Projekten
- **Hauptaufgabe des ICT-Betriebs**
  - Betrieb von IT-Services einer Unternehmung in genügender Verfügbarkeit in einer Linienorganisation: „Fabrik“

**Kapitel 03 Übersicht IT – Betriebsprozesse**

1. Sie kennen die wichtigsten IT-Betriebsprozesse und verstehen die Bedeutung des IT-Betriebs für ein Unternehmen.





### Kernprozesse

- **Monitoring**  
*Effektive und potentielle Störungen erkennen. Automatische Massnahmen zur Behebung auslösen.*
- **Incident Management**  
*Stellt bei Beeinträchtigung eines Services diesen so schnell als möglich wieder her.*
- **Problem Management**  
*Hält Fehlerhäufigkeit möglichst gering und verhindert das wiederholte Auftreten von Störungen.*
- **Change Management**  
*Managed alle Aenderungen an Software, Hardware und Netzen mittels standardisierten Verfahren und möglichst geringen Auswirkungen.*
- **SW/HW Deployment**  
*Plant und führt den Rollout von getesteter Hard- und Software durch.*
- **Service Desk**  
*Der Service Desk (Help Desk) ist die zentrale Anlaufstelle für den Kunden.*

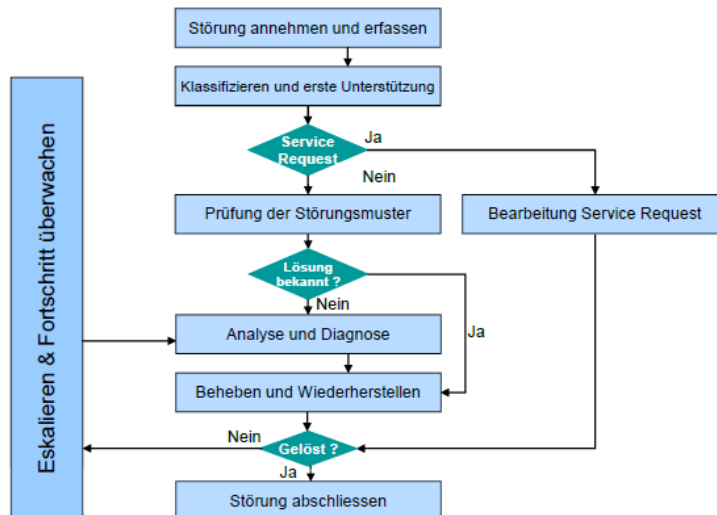
### Unterstützungsprozesse

- **Configuration Management**  
*Führt den Status aller IT-Komponenten und deren Beziehungen und stellt diese Informationen allen anderen ICT-Prozessen zur Verfügung.*
- **Capacity Management**  
*Plant und verwaltet die für den IT Betrieb benötigten technischen Ressourcen*
- **Availability Management**  
*Stellt die vereinbarte Verfügbarkeit und die Performance der IT Services sicher.*
- **Financial Management of IT Services**  
*Regelt und wickelt die Bezahlung von IT Services ab.*

### Management Prozesse

- **Service Level Management**  
*Definiert, überwacht und optimiert die IT-Service Vereinbarungen mit dem Business*
- **IT Service (Business) Continuity Management**  
*Stellt die Wiederherstellung von Anwendungen, Plattformen und Daten durch Vorsorgemassnahmen sicher und überprüft und testet sie periodisch.*
- **Architektur & Standards**  
*Legt die Architektur und Standards für den IT-Betrieb fest. Dies umfasst Hardware, Software (z.B. Datenbanken) und IT-Betriebsapplikationen.*
- **Plattform-Management**  
*Erstellt und wartet Runtime-Plattformen für die IT. Eine Plattform umfasst Hardware und Software.*

## 2. Sie kennen den Incident-Management Prozesse und können Key Performance Indikatoren und Critical Success Factors erläutern.



Der **Zweck** des Incident Management ist einen (IT) **Dienstes so schnell wie möglich wieder für den Anwender herzustellen.**

Dabei werden oft **provisorische bzw. temporäre Lösungen (Workarounds) eingesetzt, statt eine dauerhafte Lösung zu suchen.**

### KPI – Key Performance Indicators

- Gesamtzahl der Störungen
- Durchschnittliche Lösungszeit mit Bezug auf SLA
- Prozentsatz der vom First-Level Support behobenen Störungen (Lösung in erster Instanz, ohne Weiterleitung)

### CSF – Critical Success Factors

- Aktuelle und sorgfältig gepflegte CMDB
- gepflegte Datenbank für Probleme und bekannte Fehler
- ausreichend automatisiertes System für die Erfassung, die Verfolgung und die Überwachung von Störungen
- enge Beziehung zum Service-Level Management

## Kapitel 04 IT Cloud Computing

### 1. Sie verstehen die grundlegende Theorie des Cloud Computing (IaaS, PaaS und SaaS). Cloud Computing ist ein Modell respektive beschreibt den Ansatz, um IT-Applikationen (Services) aber auch Plattformen und IT-Infrastrukturen nach Bedarf dynamisch erweiterbar über ein Netzwerk zur Verfügung zu stellen.

Eine Cloudeinfrastruktur respektive ein Cloud-Dienst muss fünf wichtige Eigenschaften aufweisen:

- On Demand Self Service
- Rapid Elasticity / Scalability
- Broad network access
- Resource pooling
- Measured Service

### PUBLIC Cloud

- Im Eigentum eines IT-Dienstleisters befindliche und von diesem betriebene Cloud-Umgebung.
- Zugriff über Internet
- Flexible und schnelle Nutzung durch Subskription.
- Stellt eine Auswahl von hoch-standardisierten Geschäftsprozessen, Anwendungen und/oder Infrastrukturservices auf einer variablen "pay per use"-Basis zur Verfügung.

**PRIVATE (Enterprise) Cloud**

- Kundeneigene, vom Kunden selbst betriebene Cloud-Umgebung.
- Zugang beschränkt; nur für den Kunden selbst, autorisierte Geschäftspartner, Kunden und Lieferanten.
- Zugriff über firmeneigenes Intranet
- Effiziente, standardisierte und sichere IT-Betriebsumgebung unter Kontrolle des Kunden, die individuelle Anpassung erlaubt. (firmeneigene IT)

**HYBRID Cloud**

- Diese Cloud Art ist eine Mischung der Nutzung einer herkömmlichen IT Infrastruktur, private und public Cloud.
- Vermutlich eine am häufigsten in Praxis anzutreffende Form der Cloud-Nutzung.

## 2. Sie kennen die wichtigsten Prinzipien für eine Cloud-Architektur und können diese bewerten und in der IT-Landschaft einordnen.

**SaaS Software as a Service**

Dropbox, Flickr, Office 365, Google Apps for Business

**PaaS Platform as a Service**

Windows Azure, Google App Engine

**IaaS Infrastructure as a Service**

Amazon EC2, AppNexus, HP Cloud Enabling Computing, Sun Cloud, Root Server (Hoststar)

**SaaS - Software as a Service**

Die oberste Ebene umfasst Anwendungen. Diese werden im Markt als Software as a Service (SaaS) bezeichnet. Diese Dienste richten sich an Anwender. Geschäftsanwendungen werden als standardisierte Services von einem Dienstleister bereitgestellt. Dabei sind ihre Anpassungs- und Integrationsmöglichkeiten oft eingeschränkt. Bekannte Beispiele sind Speicherservices wie Dropbox, aber auch Desktop-, Kollaborations- und Kommunikations-Anwendungen sowie industriespezifische Geschäftsabläufe (ERP-Systeme), die vollständig von der Technologie abstrahiert sind, fallen in diesen Bereich.

Zielgruppen

Privatkunden, Firmenkunden (Fachabteilungen), Business Analysten

**PaaS – Platform as a Service**

Eine Ebene darunter liegen IT-Leistungen für Entwickler-Plattformen (PaaS). Mit den Cloud Services dieser Ebene befassen sich System-Architekten und Anwendungsentwickler. PaaS beschreibt Services auf der Anwendungs-Infrastruktur-Ebene (Middleware-Layer), die auf Basis von technischen Frameworks, also Entwicklungs-Plattformen, angeboten werden. Mit ihnen lassen sich Anwendungskomponenten entwickeln und integrieren. Hier finden sich je nach Hersteller Cloud Services für Datenbank Funktionalitäten, Zugriffskontrolle, Workflow-Steuerung und für die Synchronisation von Anwendungen und Endgeräten in vernetzten Systemen.

Zielgruppen

Architekten, Anwendungsentwickler/-integration

**IaaS - Infrastructure as a Service**

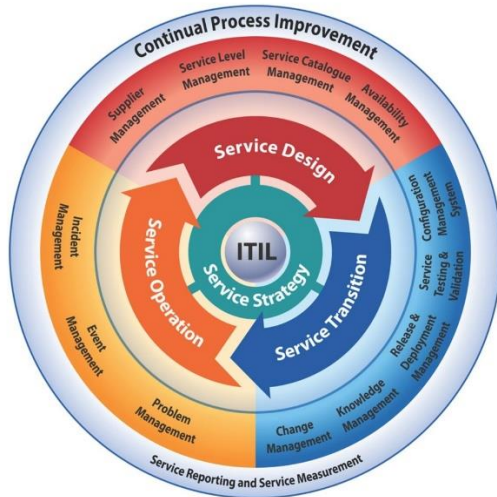
Auf der untersten Ebene sind die IT-Leistungen der Basisinfrastruktur (IaaS) angesiedelt. Sie bilden das Tätigkeitsfeld der Spezialisten für den IT-Betrieb sowie der IT-Dienstleister. Auf technologischer Ebene wird hier im Wesentlichen wenig veredelte Rechen- und Speicherleistung auf virtualisierten Servern sowie Netzwerkinfrastruktur-Funktionalität mit hohem Standardisierungsgrad und intelligentem System-Management als Service bereitgestellt. Dabei können die einzelnen Funktionalitäten auch eng verbunden sein und als integrierter Service angeboten werden.

Zielgruppen

IT-Betrieb/IT-Dienstleister, Cloud-Provider

# Kapitel 05 ITIL & IT-Service Management

1. Sie kennen die Bedeutung von ITIL und wissen was ITIL ist.



**ITIL ist**

- *IT Infrastructure Library*
- Ein "best practice" Framework um IT Applikationen/Services zu managen
- De facto Standard
- Mehrere Bücher / CD

**... ein generisches Prozess- und Rollenmodell**

- Rollenmodell
- Organisationsneutral
- Umfassend und vollständig (alle Aspekte und Funktionsbereiche sind berücksichtigt)
- Auf grössere IT-Organisationen ausgelegt
- Fokus: Weg von Technik-Orientierung hin zu KUNDEN-Orientierung

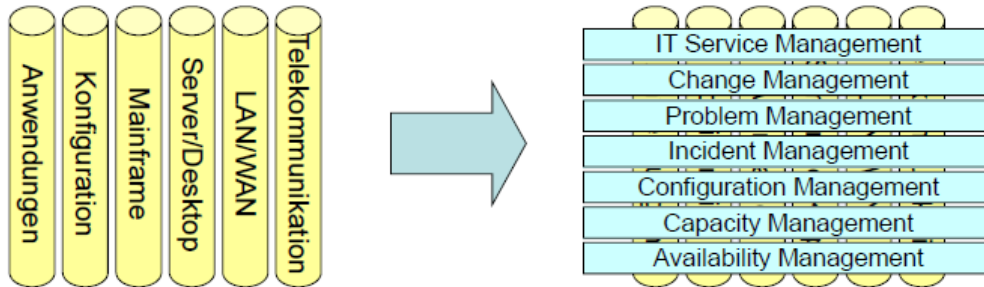
- IT Services / Applikationen stehen im Vordergrund
- IT Service
  - Ist eine Menge von IT-Funktionen, für die Unterstützung eines Business-Prozesses: Applikation!
- ITIL
  - Ist ein de facto Standard für den ICT-Betrieb
  - Viele Toolhersteller (BMC, HP, CA, IBM) richten sich danach aus
  - Weg von TECHNIK Orientierung -> hin zu KUNDEN Orientierung
- Kernbereiche sind:
  - Service Support: effiziente Erbringung von IT-Services
  - Service Delivery: Planung und Bereitstellung von ITServices

2. Sie verstehen die Kernidee von ITIL, das IT-Service Management.

Idee: Strukturiertes Zusammenführen von IT-Dienstleistungen und Business-Prozessen

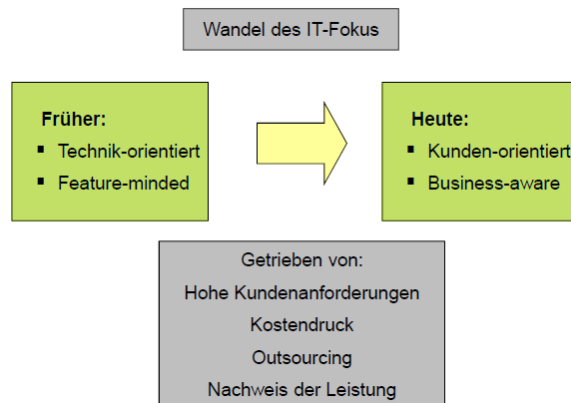
**Systemorientierte Sicht**

**Prozessorientierte Sicht**



Prozessorientiertes Service Management fasst gleichartige und übergreifende Funktionen und Aufgaben für verschiedene IT-Komponenten in **Prozessen** zusammen.

Dies verbessert die **Definition von Verantwortlichkeiten** und verlagert die Ausrichtung von einer **Systemkultur** hin zu einer **Service-Kultur**.





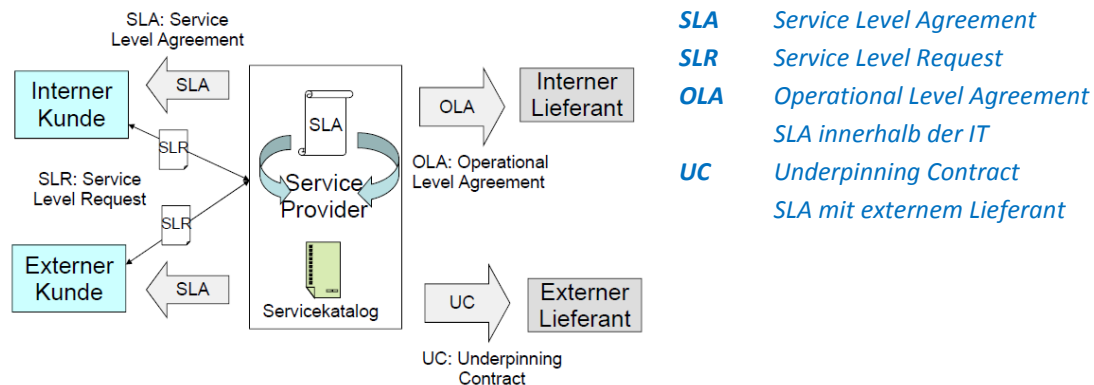
# Kapitel 06 IT Service Level Management

1. Sie kennen das IT Service Level Management und können ein Service Level Agreement erläutern.

Service Level Management stellt sicher, dass IT Services auf einem geforderten und akzeptablen Mass erbracht werden.

Dies beinhaltet das Verständnis der Anforderungen der Benutzer und eine Vereinbarung über die zu erwartende Service Qualität.

- Ein Service Katalog hilft die angebotenen Dienste zu definieren und entsprechende Ressourcen bereitzustellen
- Legt die Erwartungshaltung der Anwender entsprechend fest
- Rapportierung über Dienste hilft beim Vergleich, der Zielbestimmung und der Planung von Verbesserungen



- SLA** Service Level Agreement
- SLR** Service Level Request
- OLA** Operational Level Agreement  
SLA innerhalb der IT
- UC** Underpinning Contract  
SLA mit externem Lieferant

IT Service FrontNet	<b>SLA Key Figures:</b> ■ Service Class: <b>B19</b> ■ Availability: <b>99%</b> ■ Performance: <b>6 TOP30 use cases, 10 sec</b> ■ BCP Class: <b>11</b> ■ Monitoring/Reporting: <b>TOP30, Availability Cockpit planned</b> ■ Related OLAs: <b>JAP Silver, FN Addendum</b> ■ # of assigned APPO applications: <b>47</b> ■ For <b>SLA 1.0 Performance Measured with 16 FN Servlets (Approved with Business)</b>
	<b>Current Status / Outlook:</b> ■ SLA version 0.9, under review with business ■ Monitoring/Reporting: Extension to 30 use cases planned (pilot by 09/06, in production by 11/06) ■ SLA to be approved and put in place by 15.7.06

Extended	<b>A</b>	<b>7 x 24h</b> <ul style="list-style-type: none"> <li>Verfügbarkeit 99.0 %</li> <li>Bedingungen: Spezielle Voraussetzungen an HW/SW</li> </ul>
Standard	<b>B</b>	<ul style="list-style-type: none"> <li>Verfügbarkeit 99.0 % (gemessen MO-FR 06:00 – 23.00 Uhr, resp. 23.00 Uhr)</li> <li><b>B23 06h - 23h</b> ■ Zusätzliches Wartungsfenster jeden Donnerstag</li> <li><b>B19 06h - 19h</b> ■ SA/SO und nachts ausserhalb der Wartungsfenster Betrieb nach „best effort“</li> </ul>
Light	<b>C</b>	<b>Best Effort</b> <ul style="list-style-type: none"> <li>Keine spezielle Verfügbarkeit (nicht gemessen)</li> <li>Keine Pikett-Organisation</li> <li>Nur Infrastruktur (HW, Betriebssystem, Floorspace, Klima, etc.)</li> </ul>

## Bestandteile eines SLA

- **Servicebeschreibung**
  - o Beschreibung der Leistungen eines Service und der zugehörigen Optionen
- **Betriebsfenster**
  - o In welchem Zeitfenster wird eine Applikation supported und eine bestimmte Verfügbarkeit garantiert, z.B. 08.00 – 18.00 Uhr
- **Verfügbarkeit**
  - o in Prozent, z.B. 99%
  - o 99% Verfügbarkeit bei einem 24h-Betrieb heisst, dass die Applikation immer noch 87.6 Stunden oder ca. 3.6 Tage im Jahr ausfallen kann.
- **Performance**
  - o In Sekunden für eine bestimmte Auswahl von wichtigen Funktionen oder Antwortzeiten eines Services, welche den IT-Service gut repräsentieren, z.B.: <= 3 Sek. für eine Saldoabfrage eines Kontos
- **Business Continuity Planning Klasse (Desaster Recovery)**
  - o Datenverfügbarkeit (RPO)
    - **RPO: Recovery Point Objective:** Bestimmt "wieviel" der Datenverlust sein kann in Stunden, sprich wie alt maximal die letzten katastrophensicher gespeicherten Daten sind; z.B. 0 Sekunden, 12 Stunden.
  - o Wiederherstellungszeit (RTO)
    - **RTO: Recovery Time Objective:** Zeit die maximal benötigt wird, eine Applikation wieder Lauffähig zu machen, z.B. 24h



- **Operational Level Agreement**
    - o *Das oder die zugrundeliegende OLAs, um den Service zu erbringen (OLA: internes SLA)*
  - **Preis**
    - o *Der Preis für den IT-Service (Pauschal, pro Benutzer, pro Abfrage, etc.)*
2. Sie können einen Service Level Katalog erstellen und sind in der Lage einen IT-Service zu beschreiben.  
*Siehe Übung 5.4 Service-Katalog*
  3. Sie können die Budgetierung der Kosten für ein Betriebsteam oder für einen kleinen Servicedienstleister durchführen.  
*Siehe Übung 6.1 SLA & Budget erstellen*

## Kapitel 07 SLA Fallstudie

1. Sie kennen und verstehen die Beziehungen zwischen einem ICT-Betreiber und dessen Kunden.
  - *Verantwortlichkeiten definieren*
  - *Ansprechperson / Prozesse definieren*
  - *SLA*
  -
2. Sie kennen die wichtigsten rechtlichen Aspekte im Spannungsfeld ICT-Betreiber und Kunde.
  - *Abgrenzung zwischen Betreiber und Kunde*
  - *Kündigungsbedingungen*
  - *Datenschutz*
  - *Haftung, Haftungsausschluss (Konventionalstrafe)*
  - *Abrechnungsperiode / Dauer*
  - *Anwendbares Recht und Gerichtsstand*
3. Sie können die Risiken für eine ICT-Betreiber Firma abschätzen und kennen die wichtigsten vertragsrechtlichen (SLA) und technischen Mechanismen für deren Minderung.
  - *Verfügbarkeit Netzwerk / Server*
  - *Anbindungskapazität*
  - *Support / Antwortzeiten im Support*
  - *SLA Preise*
  - *Inhalt SLA: Siehe Kapitel 7*
  - *Verantwortung, Haftung grundsätzlich ablehnen*
  - *Maximal Betrag festlegen*
  - *Zweiter Raum / Gebäude für Backup / Disaster-Infrastruktur*
  - *Kunde muss sich selber um Business Continuity kümmern, z.B. mittels Notebook*
  - *Rechnerleistung / Server mieten*
  - *RZ @Container mieten*
  - *Vertrag mit HW-Lieferant für Ersatz-HW*
  - *Disaster-RZ*

## Kapitel 08 Outsourcing & Offshoring

### 1. Sie verstehen was Outsourcing ist und kennen die möglichen Ausprägungen.

#### **Definition: Outsourcing**

- *Herkunft: Outside + resource + using*
- *Outsourcing ist die signifikante Beteiligung von externen Firmen bei der Erbringung von IT Services mittels Mitarbeiter und IT Komponenten innerhalb Firmen.*
- *Outsourcing beinhaltet die Frage, ob bestimmte Aktivitäten intern oder durch einen externen Anbieter erbracht werden soll: längerfristiger externer Bezug von IT-Dienstleistungen*

#### **Definition: Offshoring**

- *Herkunft: Offshoring: Off + shore*
- *"ausserhalb/vor der Küste"*
- *Durchführen von Tätigkeiten/Aufgaben im Ausland*

### 2. Sie kennen die Beweggründe sowie Risiken und Strategien des Outsourcing

#### **Gründe für Outsourcing: Sicht Management**

- *Reduktion der IT-Kosten*
- *Erhöhung der Transparenz*
  - *Das Business-Management misstraut oft der internen IT bezüglich Effizienz und Kostentransparenz*
- *Konzentration auf die Kernkompetenzen*
  - *Umlenkung von Finanzmitteln auf das Kerngeschäft*
  - *Das tun, was man gut kann und womit man "gutes" Geld verdient*
- *Freisetzung von Kapazitäten*
- *Konzentration der IT auf die unternehmensrelevanten Aufgaben (Fokussierung auf die strategischen Aufgaben)*
- *Steigerung der Qualität und Geschwindigkeit hinsichtlich eingesetzte IT-Infrastruktur und IT-Dienstleistungen*

#### **Gründe für Outsourcing: Kosten**

- *Kostenreduktion durch Ausschöpfung von Grössen und Breitenvorteilen des Outsourcing-Anbieters*
  - *Erreichen einer "kritischen Grösse" um kostengünstig zu produzieren*
  - *Beispiel: dezentrale Desktop-Services der "Swisscom IT-Services"*
- *Umwandlung von Fixkosten in variable Kosten*
  - *Beispiel: Ersatz des eigenen Telefonie Systems (fixe und hohe "Sockel"-Kosten) durch Tarif pro Minute (variable Kosten)*
- *Verbesserung der Planbarkeit der Kosten und der Kostentransparenz*

#### **Gründe für Outsourcing: Technologie & Verminderung von Risiken**

- *Zugang zu speziellem Know-how (Technologie und Personal)*
- *Stärkung Innovationskraft / Erhöhung Innovationspotential*
- *Möglichkeit der Nutzung moderner Technologien ohne eigene Investitionen*
- *Flexibilität bezüglich Kapazitätsanpassungen / -änderungen*
- *Verringerung / Verlagerung von Risiken*
  - *bezüglich der Technologieentwicklung*
  - *bezüglich der zunehmenden Komplexität des Einsatzes moderner IT*
- *Vertraglich geregelte Abwälzung von Risiken und möglicher Gefahren an den Outsourcing-Anbieter*
- *Erhöhung der Datensicherheit (z.B. durch Ausweich-Rechenzentrum)*

#### **Gründe für Outsourcing: Personal**

- *Verringerung der Abhängigkeit von einzelnen Mitarbeitern mit Spezial-Know-how*
- *Verringerung des Personalbestands im IT-Bereich*
- *Entschärfung Personalbeschaffungsproblematik im IT-Bereich*
  - *Brechung von Spitzen*

- *Entlastung der internen IT von Routinearbeiten*
  - *Z.B.: Auslagerung der Entwicklung nach „Billiglohn-Ländern“, Konzentration auf Geschäfts-analyse und Design.*
- *Risikovorsorge bezüglich einer zukünftigen Verknappung qualifizierter IT-Kräfte*
  - *Z.B.: Fachleute in „älteren“ IT-Bereichen; Cobol oder PL/1 Fachleute und Programmierer*

#### **Gründe für Outsourcing: Finanzen**

- *Vermeidung hoher Investitionen für neue Informationstechnologien oder für kapazitativ bedingte Erweiterungen der bestehenden IT-Infrastruktur (Investitionen fallen beim Outsourcing-Anbieter an).*
  - *Z.B.: Ersatz einer alten Telefonanlage durch ein Servicevereinbarung mit einem Telecom-Anbieter*
- *Erhöhung der Zahlungsfähigkeit durch Zuführung liquider Mittel aus dem Verkauf der IT-Infrastruktur (insbesondere Hardware) an den Outsourcing-Anbieter.*
- *Möglichkeiten zur positiven Beeinflussung des Jahresabschlusses z.B. durch Reduktion des gebundenen Kapitals.*

#### **Risiken des Outsourcing: Sicht IT-Management**

- *Einseitige und grosse Abhängigkeit vom Outsourcing-Anbieter*
- *Langfristigkeit der Bindung*
  - *bis 6 Jahre üblich*
- *bei völligem Outsourcing kann der Auslagerungsentscheid nach einigen Jahren evtl. nicht mehr rückgängig gemacht werden*
- *räumliche Distanz*
  - *fehlender Kontakt zum Outsourcing-Anbieter*
  - *Z.B. nach Osteuropa oder Indien*

#### **Risiken des Outsourcing: Kosten**

- *hohe Anbahnungskosten (Kosten der Evaluation)*
  - *Sehr hoher Aufwand bei Festlegung und Ausgestaltung der Services*
  - *Sehr hoher Aufwand bei Evaluation der Anbieter*
- *hohe, einmalige Umstellungskosten: Switching Costs*
  - *"Switching Costs" können sehr hoch sein, z.B. bei Auslagerung von Applikationen mit komplexen Schnittstellen*
- *Vendorlocking: Risiko des Preisdiktats nach der ersten Phase durch Provider*
- *erhöhte Kommunikation- und Koordinationskosten*
  - *Z.B. SW-Entwicklungsprojekt: Analyse und Spezifikation in CH, Entwicklung in Indien*
- *Intransparenz resp. Unkontrollierbarkeit der vom Outsourcing Anbieter verlangten Preise*
  - *Risiken der vertraglichen Preisfixierung*
  - *Zu wenig Wettbewerb am Markt*

#### **Risiken beim Outsourcing: Technologie**

- *Starre Bindung an die Technologie des Outsourcing-Anbieters*
  - *Z.B. Storage bei Desktopplattformen: dezentral vs zentral*
- *Einsatz von veralteter Technologie zur Erbringung „billiger“ Services*
- *Verlust der flexiblen und kundennahen Serviceerbringung durch eine zu grosse Standardisierung*
  - *Einschränkung bei Wahl von Standard-Applikationen wegen vorgegebenen Plattformen*
- *Verlust IT-spezifischer Kompetenzen und Know-how, die bei einem gescheitertem Outsourcing nicht wieder rasch aufgebaut werden können*

#### **Risiken beim Outsourcing: Daten**

- *Gewährleistung des gesetzlichen Datenschutzes bei vertraulichen Daten*
  - *Z.B.: bankfachliche Daten bei CH-Banken dürfen nicht im Ausland gespeichert werden respektive ein Zugriff vom Ausland auf CH-Daten ist verboten*
  - *CH-Banken haben beim Outsourcing von Entwicklungsleistungen ins Ausland grösseres Problem beim Testen. In der CH werden teilweise anonymisierte bankfachliche Daten verwendet*

- *HR-Outsourcing: Datenhaltung und -Verarbeitung im Ausland benötigt Zustimmung des einzelnen Mitarbeiters*
- *Risiko eines Datenverlustes*
  - *Sicherstellung und Kontrolle der Datensicherung beim externen IT-Provider*
  - *Daten bilden das „Kapital“ vieler Firmen und deren endgültiger Verlust könnte zu grossen betrieblichen und finanziellen Problemen führen.*

#### **Risiken des Outsourcing: Personal**

- *Verlust von Schlüsselpersonen und deren Know-how*
  - *„gute“ Leute sind an der Arbeit interessiert und verlassen unter Umständen die Firma vor oder während des Outsourcings*
- *Widerstand des Personals gegenüber grundlegenden Veränderungen*
  - *Personalprobleme in der Umstellungs- und Übergangsphase*
- *Die im Unternehmen verbleibenden Restaufgaben der Informationsverarbeitung schaffen keine ausreichende Motivation mehr für das verbleibende Personal*
  - *Wandel der Aufgaben: Weg von der Serviceerstellung und –Erbringung hin zum reinen Lieferanten- und Service-Management.*
- *Personalpolitische und arbeitsrechtliche Probleme*
- *unternehmenskulturelle Unverträglichkeiten zwischen Outsourcing- Anbieter und –Nehmer*
  - *Angelsächsische oder indische Firma (Serviceerbringer) vs CH/Europäische Firma*

#### **Risiken des Outsourcing: Re-/Insourcing**

- *Re-Insourcing bedeutet Rückkehr zur eigenen IT-Verarbeitung*
- *Wiederaufbau von Know-how nach gescheiterten Outsourcing-Projekten*
- *Langfristige Bindung an Outsourcing-Verträge ermöglicht es nur schwer, diese Verträge zu lösen*
  - *Sehr teure Ausstiegszenarien*
- *Grosser Aufwand für den Wiederaufbau einer IT-Abteilung mit Rechenzentrum*
- *Bei völliger Aufgabe der Informationsverarbeitungs-Knowhow-Basis durch Outsourcing ist es fast unmöglich, nach mehreren Jahren den Auslagerungsschritt rückgängig zu machen.*

#### **Zusammenfassung: Motivation für Outsourcing**

*Outsourcing ist vielfach eine Antwort auf Marktveränderungen, wobei sich die Motivation in drei Hauptgruppen einteilen lässt:*

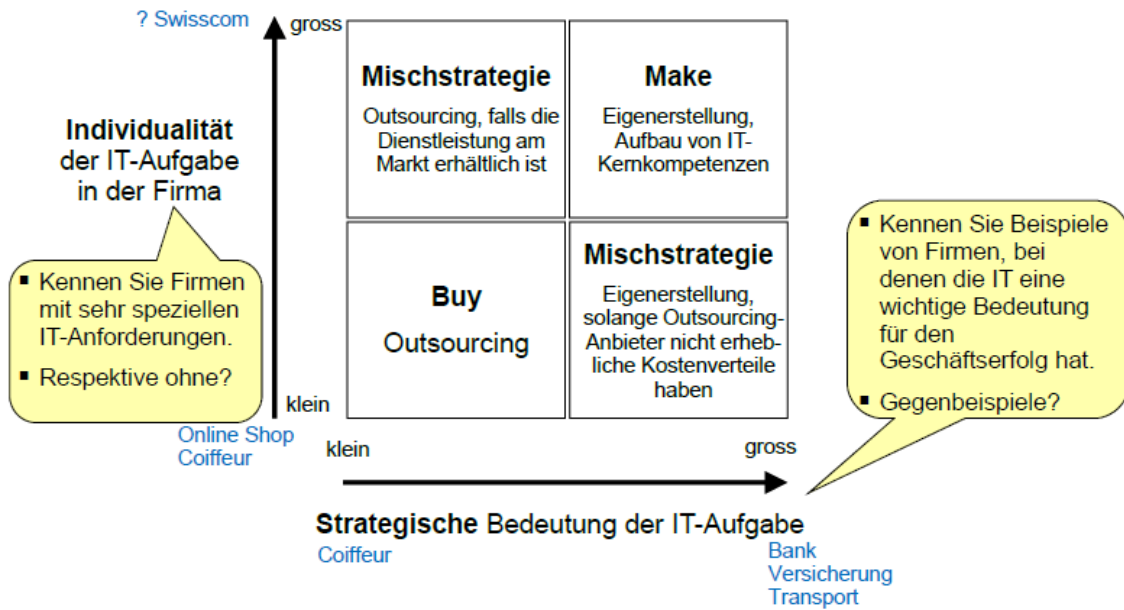
- *Kosten und Flexibilität:*
  - *z.B. Kosteneinsparungen, Gewinn an unternehmerischer Flexibilität*
- *Finanzielle Überlegungen:*
  - *Fix vs. Variable Kosten: Bsp. Telefonanlage*
  - *z.B. Reduktion des gebundenen Kapitals bei investitionsintensiven Aktivitäten, Minimieren des unternehmerischen Risikos durch Delegation der Verantwortung auf den Zulieferer*
- *Technische Überlegungen:*
  - *verbessertes Beherrschen von Kapazitätsschwankungen durch flexible Outsourcing-Verträge, lediglich Belastungsspitzen werden outgesourced (Teiloutsourcing)*

#### **Zusammenfassung: Risiken beim Outsourcing**

*Die Gefahren und Probleme von Outsourcing lassen sich zu folgenden Gruppen zusammenfassen:*

- *Abhängigkeit:*
  - *z.B. Furcht vor Verlust des unternehmerischen Handlungsspielraums und vor einem einseitigen Abhängigkeitsverhältnis, Abhängigkeit vor allem bei am Markt nicht breit verfügbaren Leistungen oder bei hohen "Switching Costs" zu einem anderen Outsourcing-Anbieter geben*
- *Sicherheit:*
  - *z.B. Abgabe der Kontrolle über sensitive Informationen kann mit Sicherheitsrisiken verbunden sein*
- *Know-how-Verlust:*
  - *z.B. bezüglich Produkte, Technologien und Verfahren im Informationsmanagement und in der Informationsverarbeitung*
- *Personalpolitik:*
  - *z.B. Aktivitätsauslagerung kann zu Vertrauensverlust und Widerständen beim Personal führen*

3. Sie sind sensibilisiert für das Thema „Make“ or „Buy“.



4. Sie kennen, was heute auf dem Markt im Bereich IT-Outsourcing angeboten wird.

Grosse Outsourcing-Anbieter in der Schweiz sind:

- HP Enterprise Services (ex EDS)
- IBM
- Swisscom

Beispiel für einen KMU- Outsourcing - Anbieter

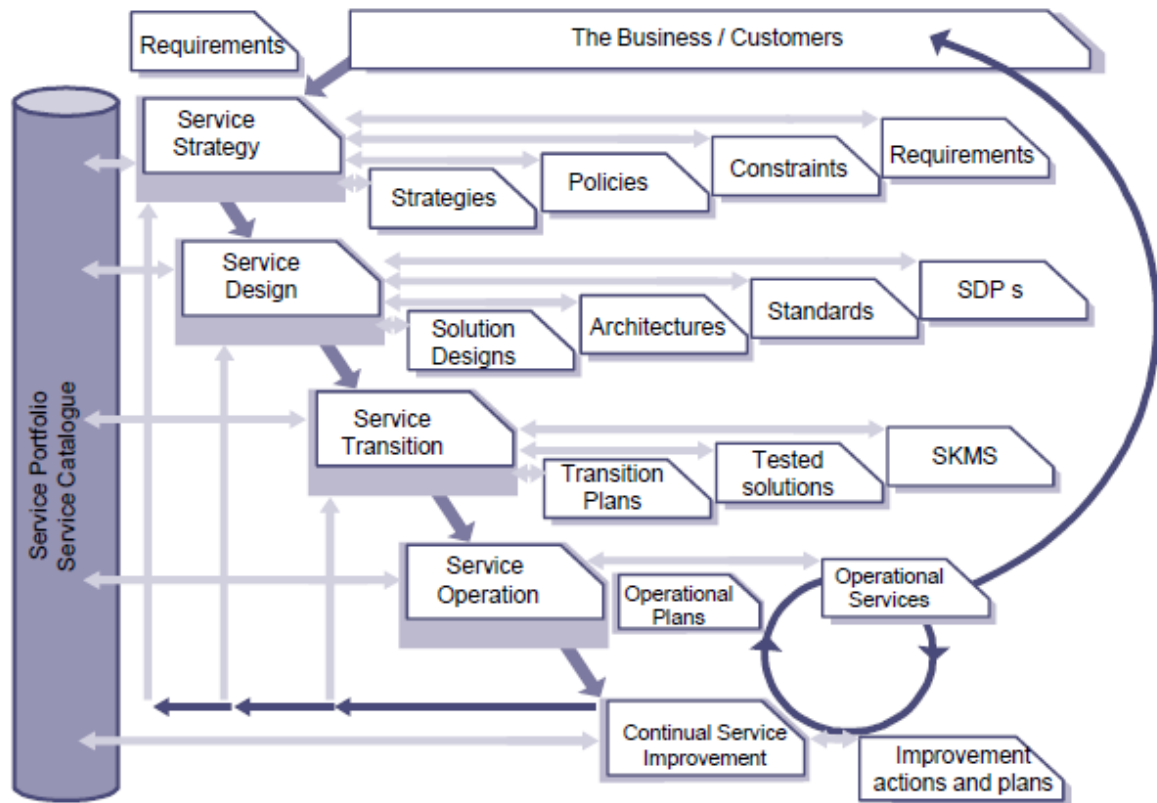
- Firma In4U: Sie bietet folgende Outsourcing Services an:
  - Projekt-Management
  - Betrieb von Standard Software (Abacus, SAP, etc.)
  - Server-Hosting
  - „IT-komplett“ Betrieb (ICT Outsourcing)
  - Application Service Providing
  - Individual ICT-Services (z.B. Migrationen)

### Kapitel 09 Vertiefung IT-Betriebsprozesse und ITIL V3

1. Sie kennen ITIL Vers. 3 und dessen wichtigste Prozessgruppen



- Service Strategy
- Service Design
- Service Transition
- Service Operation



2. Sie kennen und verstehen die wichtigsten IT-Betriebsprozesse aus ITIL V3 und deren Bedeutung für das Unternehmen.

**Ziel der Service Strategy**

*Beschreibt die Gestaltung, Entwicklung und Implementation von IT Service Management als organisatorische Fähigkeit und strategischen Wert für die Organisation.*

- *Entwurf, Entwicklung und Implementation von Service Management als organisatorische Fähigkeit und strategisches Asset.*
- *Beherrschen und bewältigen der mit den Service Portfolios verbundenen Kosten und Risiken.*

*Prozesse:*

- *Strategy Generation*
- *Service Portfolio Management*
- *Demand Management*
- *Financial Management*

**Ziel des Service Design**

- *Beschreibt den Entwurf und die Entwicklung von IT Services und die dazu notwendigen Designgrundsätze und Methoden.*
- *Sicherstellen, dass Services den aktuellen und zukünftigen Anforderungen des Business entsprechen.*
- *Erstellung und zukünftiger Betrieb der IT Services (Policies, Architektur, Risikomanagement, Dokumentation).*
- *Entwurf und Entwicklung von Service Management Prozessen.*

*Prozesse:*

- *Service Catalogue Management*
- *Service Level Management*
- *Capacity Management*
- *Availability Management*
- *IT Service Continuity Management*
- *Information Security Management*
- *Supplier Management*

**Ziel der Service Transition**

- *Beschreibt die kontrollierte Überführung von neuen und geänderten IT Services in den IT-Betrieb.*
- *Strukturierte Überführung neuer und geänderter Services in den operativen Betrieb.*
- *Sicherstellen, dass der Service entsprechend den Anforderungen und Rahmenbedingungen genutzt werden kann.*
- *Management der Komplexität von Änderungen an Services und Service Management Prozessen: verhindern unerwünschter Konsequenzen bei gleichzeitiger Unterstützung und Förderung von Innovation.*

**Prozesse:**

- *Transition Planning & Support*
- *Change Management*
- *Service Asset & Configuration Management*
- *Release & Deployment Management*
- *Service Validation & Testing*
- *Evaluation*
- *Knowledge Management*

**Ziel des Service Operation**

- *Beschreibt den effizienten und effektiven Betrieb, Support und Unterhalt der IT Services.*
- *Sicherstellung eines effektiven und effizienten Servicebetriebs aus einer End-zu-End Perspektive.*
- *Koordination und Ausführung der zur Bereitstellung und zum Betrieb der Services notwendigen Aktivitäten und Prozesse.*
- *Leistungsüberwachung, Beurteilung von Metriken und systematische Datenerfassung um eine kontinuierliche Verbesserung der Services zu ermöglichen.*

**Prozesse:**

- *Event Management (Monitoring)*
- *Incident Management*
- *Request Fulfillment*
- *Problem Management*
- *Access Management*

**Ziel von CSI / Continual Service Improvement**

- *Verbesserung der Wertschöpfung für die Kunden durch besseren Entwurf, bessere Einführung und besseren Betrieb von Services.*
- *Qualitätsmanagement: Sicherstellung der Konsistenz, Wiederholbarkeit, Transparenz und Verbesserungsfähigkeit.*

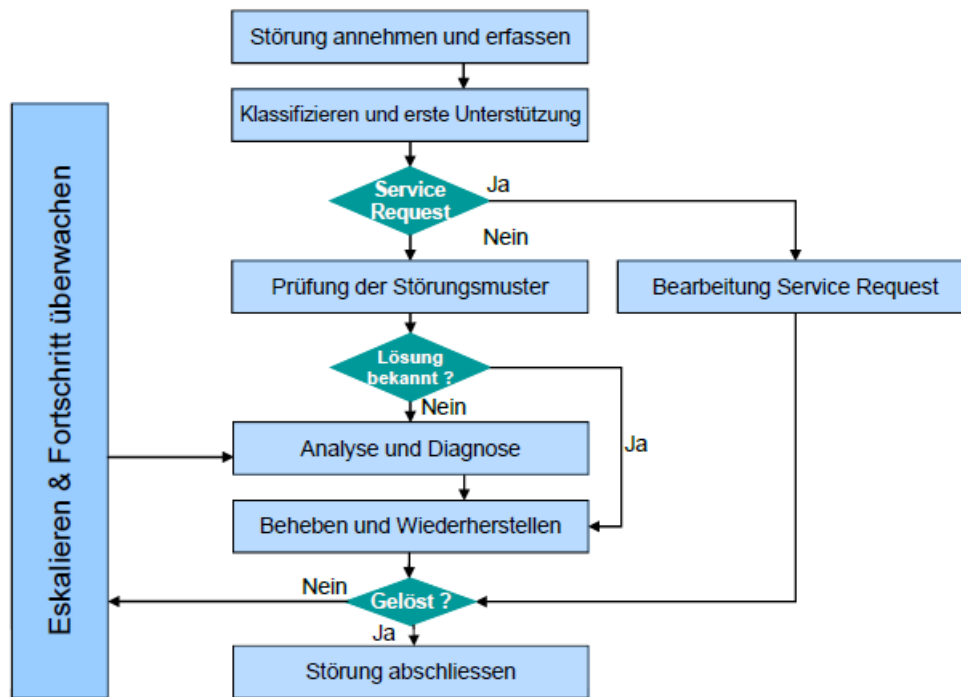
**Prozesse (Meta-Prozesse, keine operative Prozesse)**

- *Service Improvement*
- *Service Reporting*
- *Service Measurement*



3. Sie kennen das Incident-Management, das Configuration Management und zwei ausgewählte IT-Betriebsprozesse vertieft.

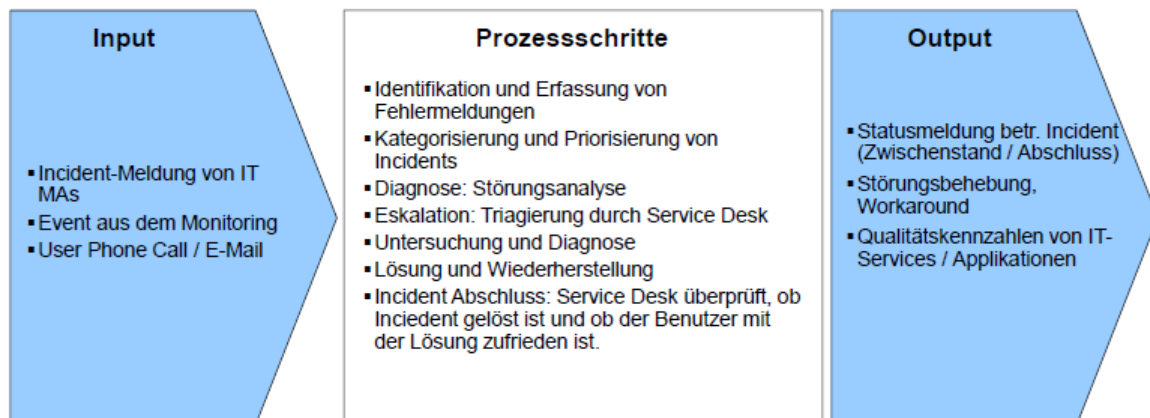
**Incident Management (Störungsmanagement)**



- Der **Zweck** des Incident Management ist einen (IT) **Dienstes so schnell wie möglich wieder** für den Anwender **herzustellen**.
- Dabei werden oft *provisorische bzw. temporäre Lösungen (Workarounds)* eingesetzt, statt eine dauerhafte Lösung zu suchen.
  - Störungen werden schnell bearbeitet, um ihre Auswirkung und Schwere abzumildern
  - Nutzt verfügbares Wissen –kein Neuerfinden des Rades.
  - Schnelle Wiederherstellung von Diensten produziert zufriedene Nutzer
  - Erhöhte Effizienz bei der Entdeckung, Diagnose und Lösung von Störungen

KPI – Key Performance Indicators:

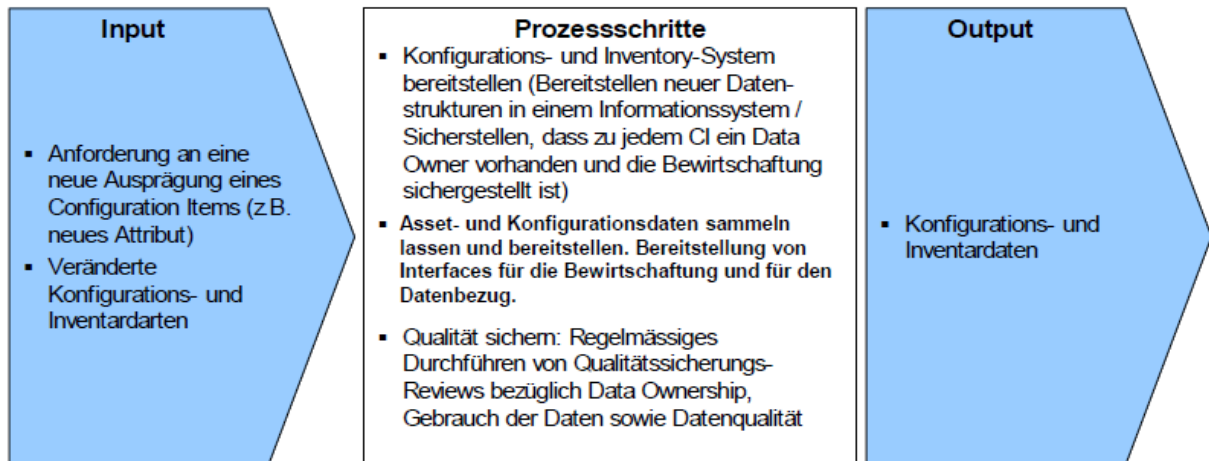
- Gesamtzahl der Störungen
- Durchschnittliche Lösungszeit mit Bezug auf SLA
- Prozentsatz der vom First-Level Support behobenen Störungen (Lösung in erster Instanz, ohne Weiterleitung)



## Configuration Management

Configuration Management hilft dabei den Überblick darüber zu behalten welches Equipment vorhanden ist, wo es ist und wofür es verwendet wird.

- Das Configuration Management verwaltet gemeinsame Konfiguration- und Inventardaten und stellt sie allen anderen Prozessen sowie dem IT Management zur Verfügung.
- Es umfasst alle notwendigen Hardware-, Software-, Netzwerk und Konfigurations-Daten sowie die dazugehörigen administrativen Informationen.
- Es stellt in Zusammenarbeit mit dem Change Management und anderen Betriebsprozessen die Kontrolle der Infrastruktur und die Qualität der Metadaten sicher, durch Überwachung und Pflege von Informationen über:
  - alle für die Lieferung von IT-Services benötigten (IT)Ressourcen
  - Status und Historie dieser Ressourcen
  - das Zusammenspiel (die Relationen) dieser Ressourcen



### CI – Configuration Item

Alle in einem Configuration Mgmt.-System verwalteten Daten sind Configuration Items – CIs. Diese können wie folgt klassifiziert werden:

- Hardware
- Software
- Dokumente
- Konfigurationen
  - Beziehungen SW und HW
  - Software-Konfiguration (SW- und Instanz-bezogen)

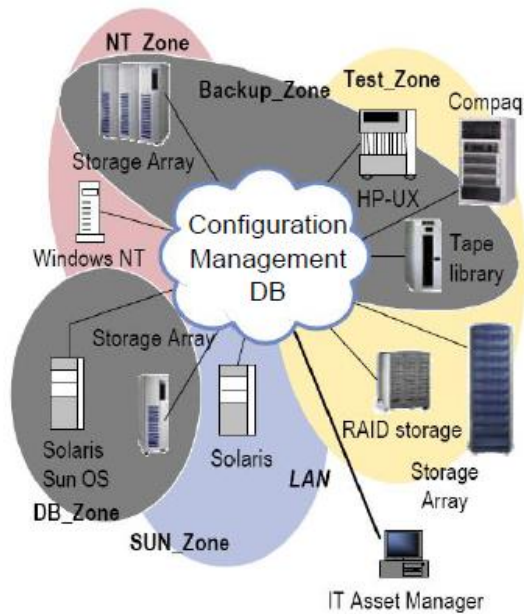
### CMDB - Configuration Management Database

Die CMDB verwaltet Daten, die in mehr als einem Systems Management Bereich bzw. mehr als einem Prozess benutzt werden, in einer (möglicherweise föderierten) Datenbank und stellt diese Daten über definierte Schnittstellen zur Verfügung.

### CM - Key Performance Indicators

- Dimension Qualität:
  - Datenaktualität und –qualität der Konfigurationsdaten
  - Abweichung Verfügbarkeit der CMDB
- Dimension Wirtschaftlichkeit:
  - Stückkosten pro Konfigurationsitem für Erfassung und Pflege
- Dimension Leistung:
  - Abweichung Planzeiten: Soll-/Ist-Vergleich der geplanten und effektiven Zeiten zur Bereitstellung bzw. Änderung von Strukturen, Daten und Schnittstellen bei der CMDB

**Summary & wichtige Erfolgsfaktoren im CM**



1. **Automatisierung:** Ohne CM ist keine Automatisierung, im IT-Betrieb möglich
2. **Beschränkung Daten:** Die Schwierigkeit besteht darin, die Daten aktuell zu halten: Weniger ist hier mehr, dafür aktuell.  
Nur die Daten speichern, die wirklich notwendig sind und für die ein Owner (Pfleger) vorhanden ist.
3. **BO Datenpflege:** Aufsetzen und Durchsetzen einer Betriebsorganisation (Prozesse) für die Datenpflege (Data- und Structure-Ownership) wichtig.
4. **Sicherung Datenqualität:** Periodische Prüfung der Datenqualität (Stichproben) und konsequentes Ergreifen von Massnahmen zur Verbesserung der Datenqualität.
5. **5. Aktives CM:** Eine gute Datenqualität erhält man quasi „automatisch“, wenn die Konfigurationsdaten die Basis für die Deploymentprozesse sind und sichergestellt ist, dass die realen Systemzustände nur über das CM-System verändert werden können.

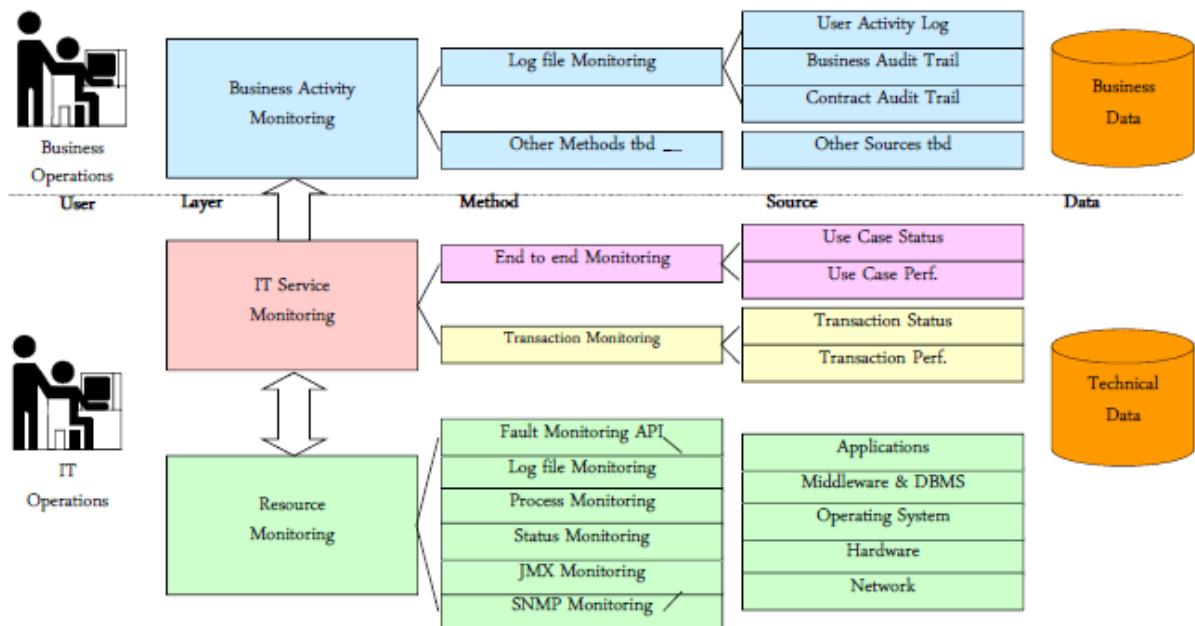
**Kapitel 10 Monitoring & Event Management**

1. Sie kennen den Monitoring & Event Management Prozess vertieft und können deren wichtigste Eigenschaften erläutern.

**Monitoring (Event Management)**

Der Monitoring-Prozess überwacht die IT Services und die zugehörigen IT Komponenten und erkennt Fehlerzustände.

- Das Monitoring zeigt jederzeit plattformübergreifend den Zustand aller Hardwaresysteme und Applikationen auf, welche zu einem bestimmten IT Service gehören.
- Er liefert Informationen zur präventiven Erkennung möglicher Störungen (Trends) bzw. zur Analyse und Fehlerbehebung.
- Er konfiguriert Instrumente zur automatischen Überwachung und Steuerung von Systemen und zur Vermeidung von Störungen.
- Beinhaltet auch die automatische Behebung von Störungen mittels Scripts (Automation)

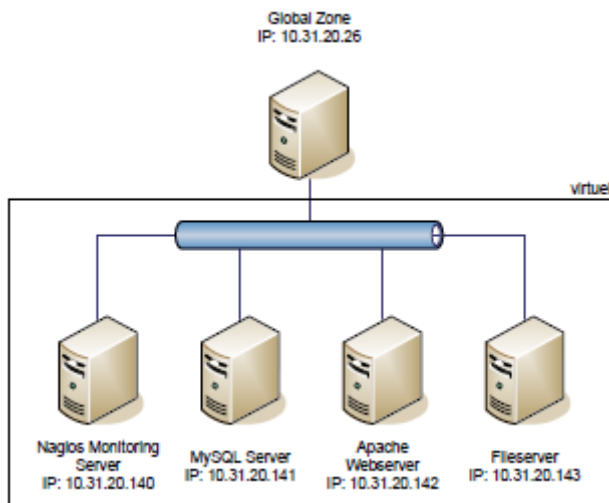


### Resource Monitoring & Event Management

- *Resource / Komponenten Monitoring*
  - *Erkennung von Fehlern und Performance Problemen bei IT-Systemen: Server, Applikationen, etc*
- *Monitored Object*
  - *Ein System oder ein Teil eines Systems, welches überwacht werden soll über welches man eine Aussage machen will, zumBeispiel: Server, Disk, Applikationskomponenten*
- *Messpunkt*
  - *Ein konkretes Charakteristika eines Monitored Objects, welches konkret GEMESSEN wird. Ein Monitored Object wird durch einen oder mehrere Messpunkte definiert.*
  - *Beispiel: Bei einer Disk wird mit Hilfe von zwei Messpunkten – Füllgrad und Lesezeit – eine Verfügbarkeitsaussage gemacht.*

## 2. Sie können die wichtigsten Funktionen eines Monitoring-Tools am Beispiel von NAGIOS erläutern.

### NAGIOS Infrastruktur



### Globale Zone in vier Zonen unterteilt

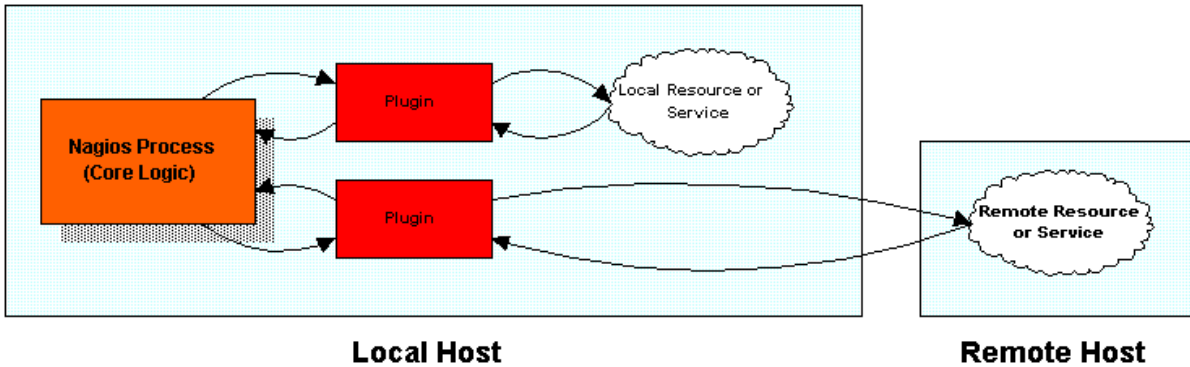
- *Nur globale Zone ist physisch vorhanden*
- *Die Unterzonen sind virtuelle Maschinen*
- *Jede Zone bietet spezifischen Dienst an*
- *Zugriff auf Zonen per SSH (putty)*

### Nagios – Features

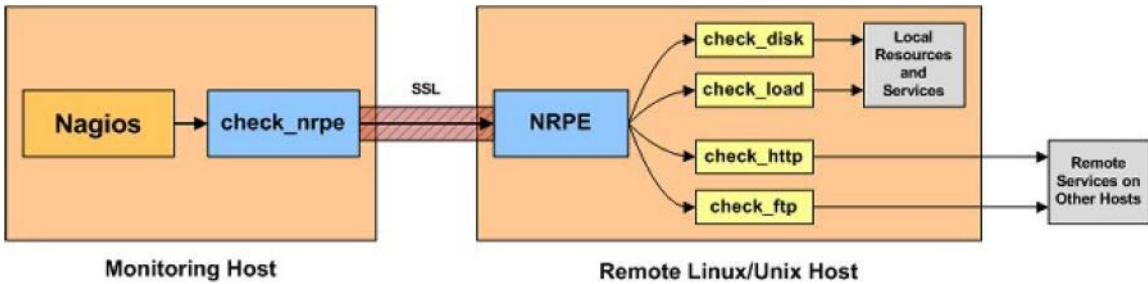
- *Viele Monitoring Möglichkeiten*
  - *Windows, Linux/Unix*
  - *Routers, Switches, Firewalls*
  - *Drucker*
  - *Applikationen*
- *Verschiedene Benachrichtigungsmöglichkeiten*
  - *E-Mail*
  - *SMS*
  - *Web Interface*
- *Event Handlers*
  - *Definierbare Reaktionen nach Problemerkennung (Bsp. Neustart eines Webservers)*
- *Reporting Options*
  - *SLA availability reports*
  - *Alert and notification reports*
  - *Reporting Addons*
- *Multi-User Capabilities*
  - *Jeder User kann eigene View aufs Web Interface haben*
- *Skalierbar und bewährt*
  - *Skalierbar auf bis zu 100'000 zu überwachende Nodes*
  - *Failover protection*
- *Huge community*
  - *Über 200 Community Plugins*

**Nagios – Arten der Überwachung**

- *Aktive Überwachung*
  - *Überprüfung wird durch Nagios initiiert*
- *Passive Überwachung (wird nicht behandelt)*
  - *Überprüfung wird durch externe Programme initiiert*
  - *Antwort wird an Nagios gesendet und verarbeitet*
- *Direkte Überwachung*
  - *Nagios kann direkt, ohne Zwischenstellen, die Dienste prüfen*
    - *„öffentliche“ Dienste wie HTTP, FTP, Ping etc. auf lokalen und entfernten Rechnern*
    - *„lokale“ Dienste wie CPU Load, disk usage etc. auf lokalen Rechnern*



- *Indirekte Überwachung*
  - *Nagios muss über Zwischenstellen, wie der NRPE Agent, die Dienste überprüfen*
    - *„lokale“ Dienste auf entfernten Rechnern*



**Nagios – Konfiguration**

- *Die Nagios Konfiguration nagios.cfg verweist auf weitere Konfigurationsdateien*
  - *timeperiods.cfg – Enthält Zeitangaben wann überwacht werden soll*
  - *contacts.cfg – Enthält Kontakte, welche benachrichtigt werden*
  - *commands.cfg – Enthält die Überprüfungscommandos*
  - *localhost.cfg – Enthält die zu überwachenden Dienste für den localhost*
  - *templates.cfg – Commandos und Services können von Vorlagen erben*
  - *my.cfg – Alle Konfigurationen für die Übung kommen hier rein*

# Kapitel 11 Data Center

1. Sie kennen die Building Blocks eines Datacenters.

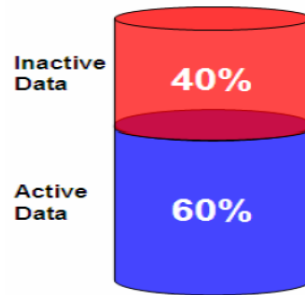
- Klima
- Strom
- Storage
- Netzwerk
- Server
- 

2. Sie wissen wie man Daten klassifiziert und sie in SLAs einbindet.

**Daten Charakteristik:**

1. Mission Critical Data
2. Business Critical Data
3. Accessible Data Online
4. Nearline Data
5. Offline Data

**Aktive vs. Inaktive Daten**



- Konsumieren teuren Speicherplatz
- Werden gepflegt, gesichert, repliziert, etc.
- Unterliegen trotzdem Rechtlichen- und Datenhaltungs-Ansprüchen
- Müssen im Katastrophenfall wiederhergestellt werden





	Mission Critical	Business Critical	Business Important	Development
Requirement	SLA 1	SLA 2	SLA 3	SLA 4
Availability	99.99%	99.9%	99%	97%
Threshold based Automatisch Laden	bis 20% vom Filesystem alloziert während 1 Arbeitstag	bis 20% vom Filesystem alloziert während 2 Arbeitstagen	bis 10% vom Filesystem alloziert während 4 Arbeitstagen	bis 20% vom Filesystem alloziert während 1 Arbeitswoche
RTO	15 Minuten	1 Stunde	8 Stunden	24 Stunden
RPO	1 Stunde	12 Stunden	48 Stunden	96 Stunden
Restore Anfragen	100 Anfragen/Woche	100 Anfragen/Woche	50 Anfragen/Woche	50 Anfragen/Woche
Backup Erfolgsrate	97%	95%	90%	90%
Archivierung - Policy	Kein Zugriff während 90 Tagen	Kein Zugriff während 30 Tagen	Kein Zugriff während 90 Tagen	Kein Zugriff während 180 Tagen
Zugriffszeit zum Archiv	Sekunden	Sekunden	Bis 4 Stunden	24-48 Stunden
Prognose	Monatlich	vierteljährlich	jährlich	jährlich
Kosten	Hoch	Mittel	Niedrig	Billig

RTO: Recovery Time Objective (Wiederherstellungszeit)

RPO: Recovery Point Objective (Maximaler Datenverlust)



3. Sie sind in der Lage Data-Tiers und deren Aufgaben zu erklären.

	Tier 1 	Tier 2 	Compliance 	Tape vault 
<b>Charakteristik</b>	Hohe Geschwindigkeit Hohe Verfügbarkeit	Mittlere Geschwindigkeit, Hohe Speicherdichte	Einmalbeschreibbar, Sicher, unveränderlich	Geringe Geschwindigkeit, Entfernbar
<b>Gebrauch</b>	Produktionsdaten	Referenzdaten	Compliance Data (vom Gesetzgeber vorgeschrieben)	Offsite Disaster Recovery
<b>Relative Kosten</b>	10	2	3	1
<b>% der gesamten Daten</b>	20%	35%	30%	15%

**Information Lifecycle Management (ILM)**

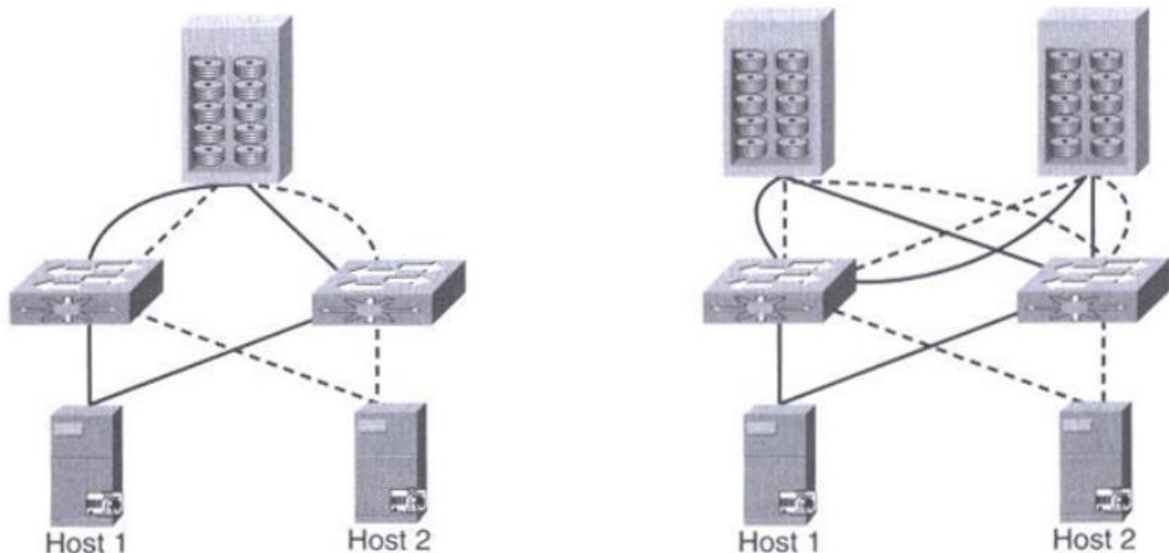


4. Sie sind fähig die kritischen Punkte eines Datacenters zu adressieren und Massnahmen vorzuschlagen.

**Ausfall Storage (SAN)**

Lösung: Redundanter Storage

Kosten: Hoch

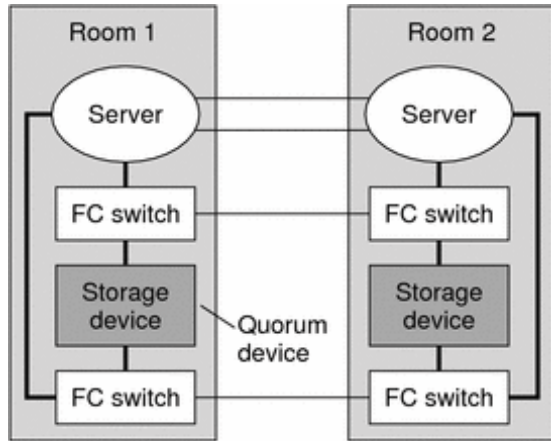




**Ausfall Server**

Lösung: *Redundante Server + Infrastruktur und asynchroner/synchroner Abgleich der Daten*

Kosten: *Mittel-Hoch*

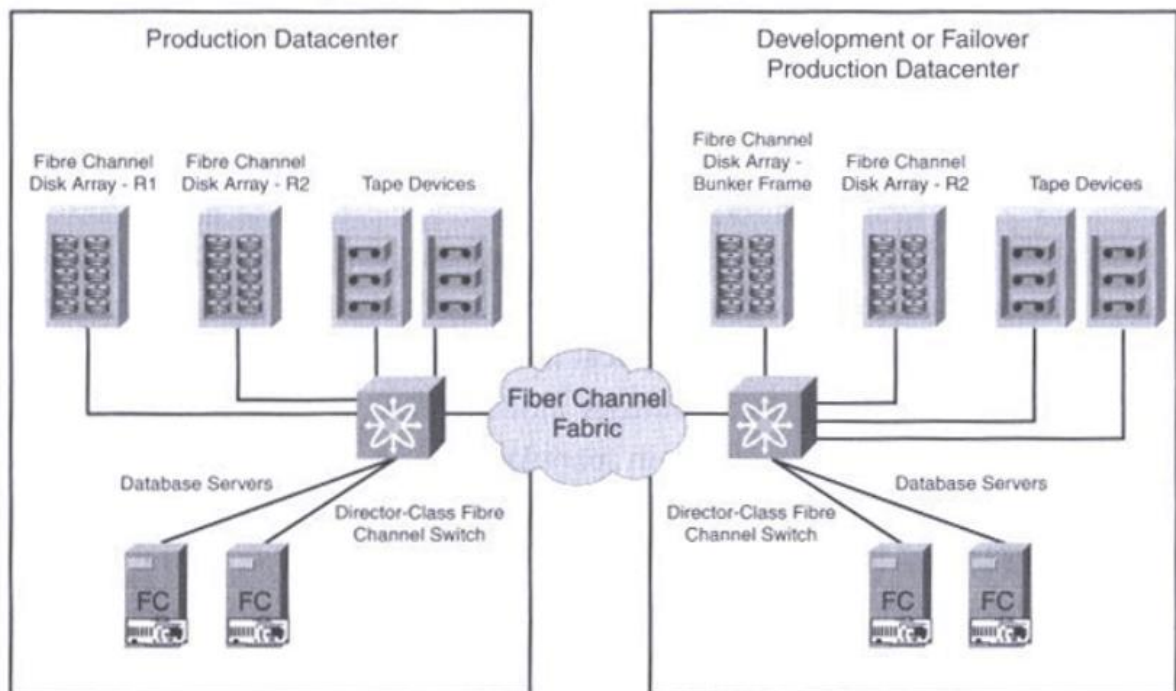


— Single-mode fiber  
— Multimode fiber

**Komplettausfall Datacenter**

Lösung: *Redundantes DataCenter*

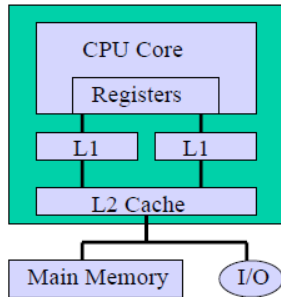
Kosten: *Sehr hoch*



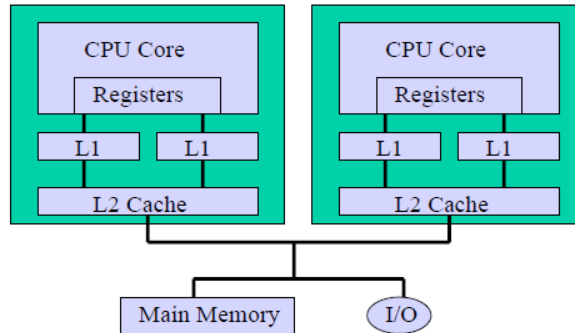
# Kapitel 12 Parallele Rechnerarchitekturen

1. Sie kennen und verstehen moderne multi Core Architekturen, Memory Architekturen, und Cache Verwaltungen.

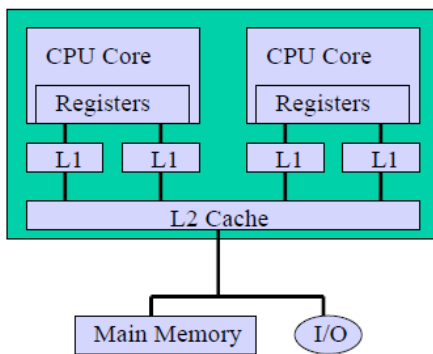
Conventional Microprocessor



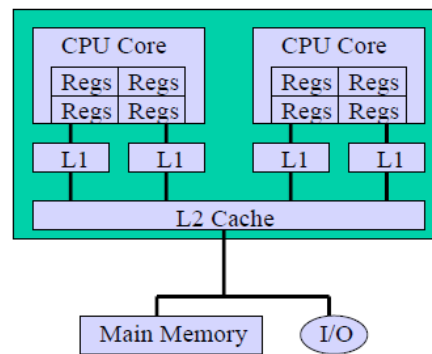
Simple Chip Multiprocessor



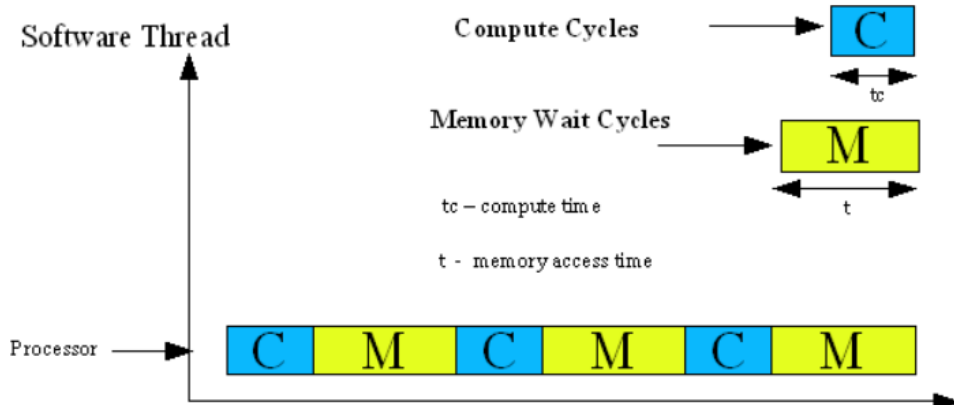
Shared Cache Chip Multiprocessor



Multithreaded, Shared Cache Chip Multiprocessor

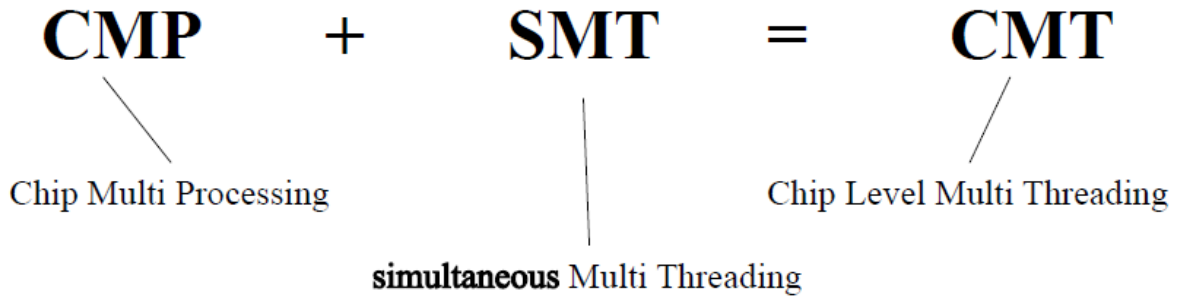


50% von der Prozessor Zeit ist Warten auf das Memory



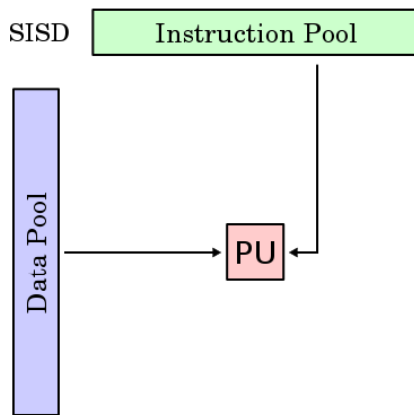
**Chip Level Multi Threading (CMT)**

→ single Thread Application wird langsamer laufen



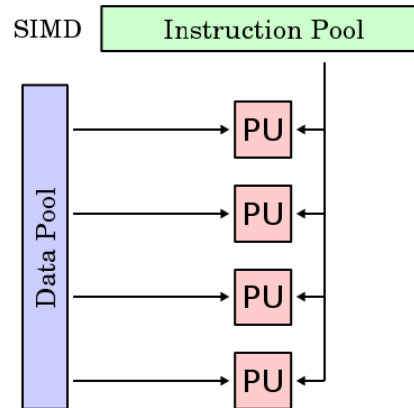
**Single Instruction Single Data:**

Von-Neumann-Rechner



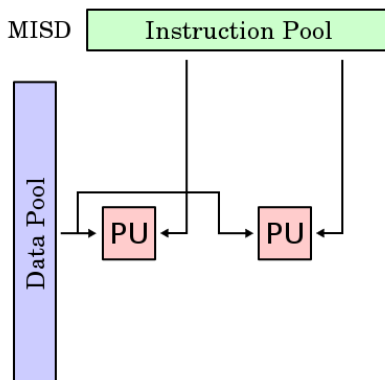
**Single Instruction Multiple Data:**

Vektorrechner, VLIW-Rechner



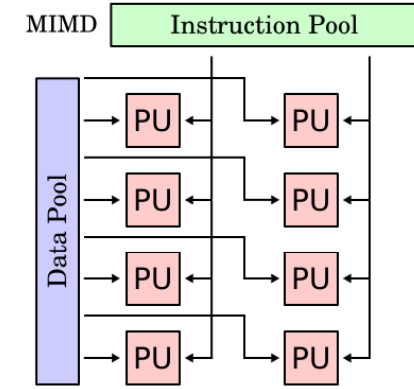
**Multiple Instruction Single Data:**

sehr selten, Systolic Array ist ein Vertreter davon



**Multiple Instruction Multiple Data:**

Parallelrechner, unsere modernen SMPs oder Computer Cluster



**Speicherorganisation von MIMD Rechnern**

Shared-Memory: ein globaler linearer Adressraum

- **UMA - Uniform Memory Access**
  - Speicherzugriffszeit gleich für alle Speicheradressen
  - Üblicherweise zentraler gemeinsamer Speicher
- **NUMA - Non-Uniform Memory Access**
  - Speicherzugriffszeit = f(Speicheradresse)
  - Üblicherweise verteilter Speicher (Distributed Shared Memory)
- **Symmetrischer Multiprozessor:**
  - jeder Prozessor hat gleiche Zugriffsrechte auf I/OEinheiten, üblicherweise UMA
  - Architektur heutiger „intern paralleler“ Workstations

**Speicherorganisation von MIMD Rechnern**

**Shared Memory:**

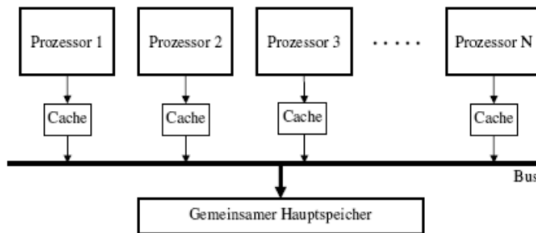
- hohe Kommunikationsleistung, kleine Latenz, kleine Hardwareaufwand, einfache Programmierung (speziell bei dynamischen Kommunikationsmustern), hardwareunterstütztes Caching.

**Private Memory:**

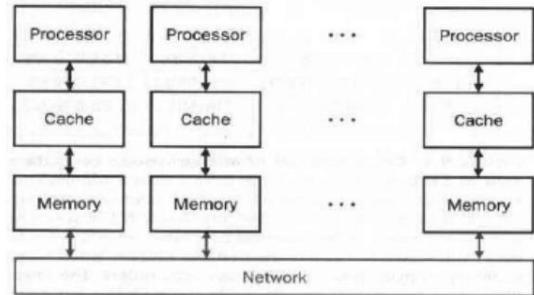
- Knoten sind "Massenware", hohe Administrationskosten, Netzwerk ist mit I/O-Bus verbunden (langsam), leicht erweiterbar,

**Shared Memory Architektur (UMA) Shared Memory Architektur (NUMA)**

- Alle Prozessoren (P) haben direkten Zugriff auf alle Speicherstellen (M)
- einfacher und schneller Kommunikationsmechanismus
- Bus-Architektur:



**Netzwerk Architektur**



Das **Amdahlsche Gesetz** ist ein Modell in der Informatik über die Beschleunigung von Programmen durch parallele Ausführung. Nach Amdahl wird der Geschwindigkeitszuwachs vor allem durch den sequentiellen Anteil des Problems beschränkt, da sich dessen Ausführungszeit durch Parallelisierung nicht verringern lässt.

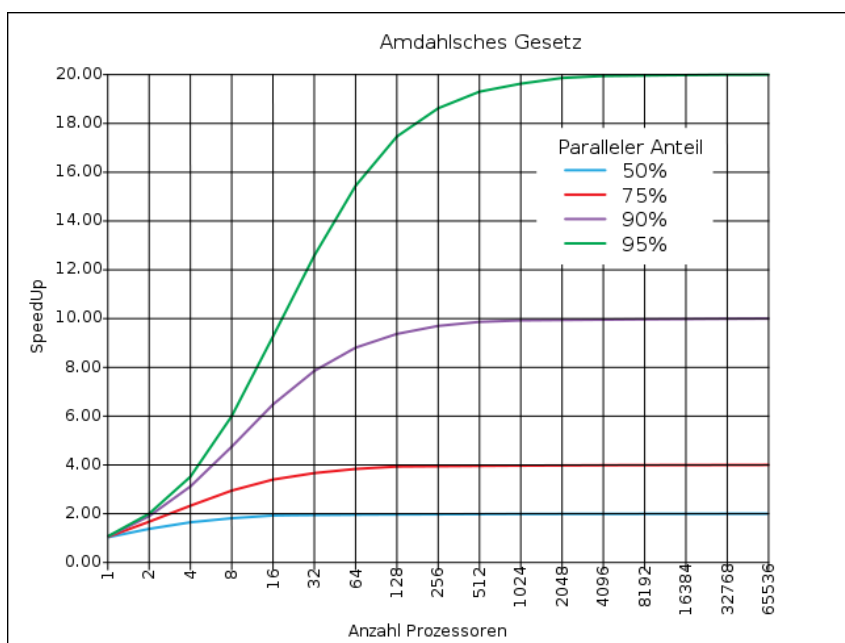
Ein Programm kann nie vollständig parallel ausgeführt werden, da einige Teile wie Prozess-Initialisierung oder Speicher-Allokation nur einmalig auf einem Prozessor ablaufen oder der Ablauf von bestimmten Ergebnissen abhängig ist.

Gesamte Rechenzeit = 1

Nicht Parallelisierter Anteil = a = 20%

Parallelisierbar = 1-a = 80%

$$T = a + \frac{(1-a)}{n}$$



**Cache Kohärenz**

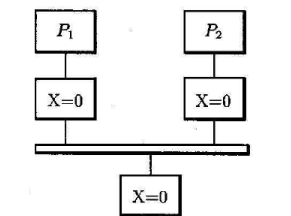
Durch die Sicherstellung von Cache-Kohärenz wird bei Mehrprozessorsystemen mit mehreren CPU-Caches verhindert, dass die einzelnen Caches für dieselbe Speicheradresse unterschiedliche (inkohärente) Daten zurückliefern.

Eine temporäre Inkonsistenz zwischen Speicher und den Caches ist zulässig, sofern diese spätestens bei lesenden Zugriffen identifiziert und behoben wird. Inkonsistenzen werden z. B. durch das Rückschreibeverfahren (Write-Back) erzeugt, das im Gegensatz zu einem Durchschreibeverfahren (Write-Through) beim Schreiben in den Cache-Speicher nicht sofort den Hauptspeicher aktualisiert

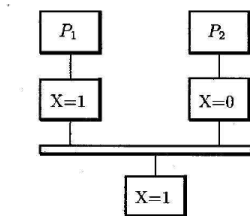
- Write trough: Cache und Speicher werden aktualisiert
- Write Back: Speicher wird nicht aktualisiert

Cachekohärenz (Beispiel 1: write through cache)

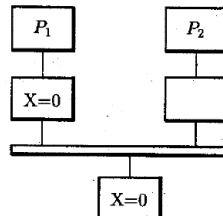
Cachekohärenz (Beispiel 2: write back cache)



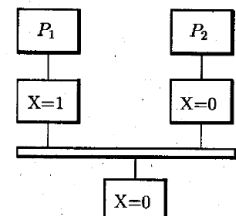
Die Variable X befindet sich in den Caches von P1 und P2 sowie im Hauptspeicher



P1 schreibt X=1. Der Hauptspeicher wird beim **write through Cache** ebenfalls verändert. P2 liest den alten (inkohärenten) Wert aus dem eigenen Cache.



Die Variable X befindet sich im Cache von P1 und im Hauptspeicher



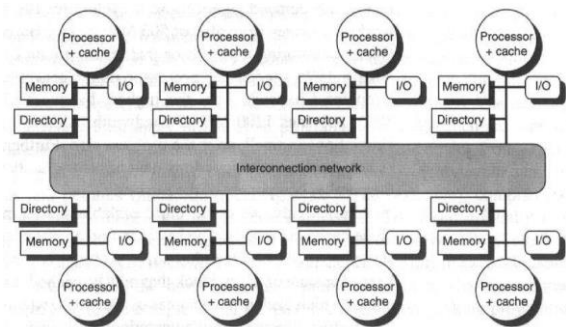
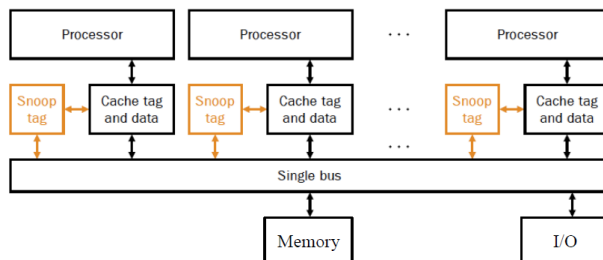
P1 schreibt X=1. Der Hauptspeicher wird nicht sofort verändert (**write back cache**). P2 liest nun den alten, inkohärenten Wert aus dem Hauptspeicher.

**Cachekohärenz - Protokolle**

- **Verzeichnisbasiert (Directory)**
  - Zentrale Liste mit Status aller Cache Blöcke. Welcher Prozessor hat was?
    - Read-Only = Shared, exclusive write = Exclusive
- **Snoopingbasiert (Snoop)**
  - Da alle Zugriffe über gleiches Medium verlaufen (Bus, Switch) können die Cachecontroller beobachten und erkennen welche Blöcke sie selbst gespeichert haben.
    - Shared (Block ist in einem oder mehreren Caches, Wert ist up-to-date),
    - Uncached (Block ist in keinem Cache),
    - Exclusive (Block gehört einem Cache, Wert ist nicht up-to-date).

**Cachekohärenz: Snoopy**

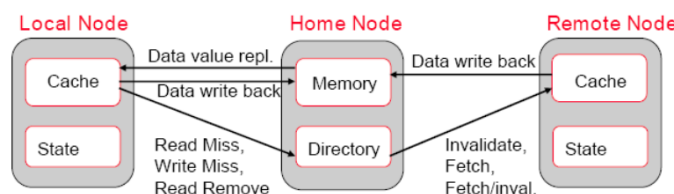
Alle Prozessoren beobachten Datenübertragungen zwischen jedem Cache und dem gemeinsamen Speicher



**Beteiligte Einheiten:**

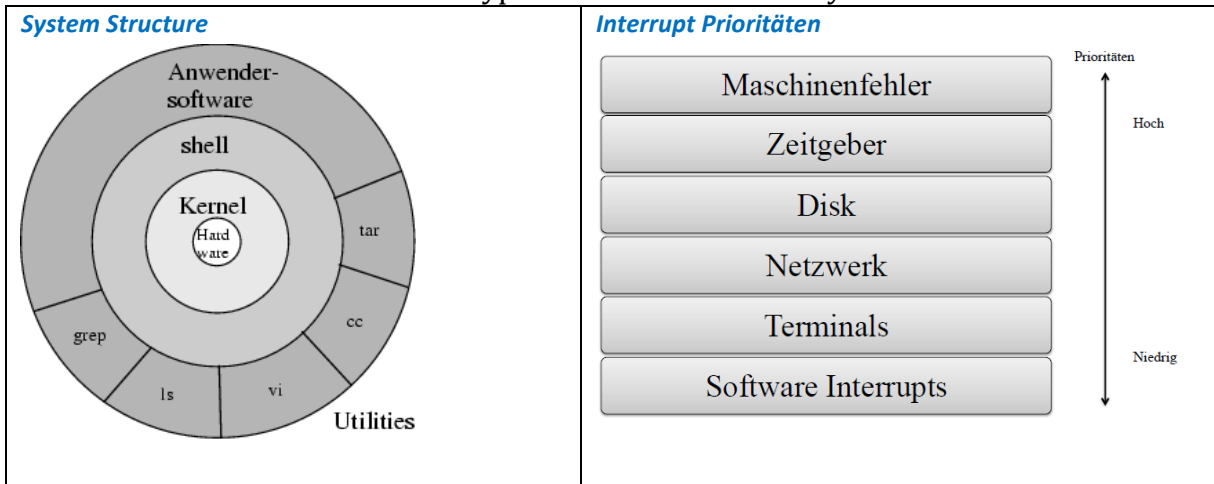
- Local Node: Ursache der gesamten Transaktion.
- Home Node: Hauptspeicher und Directory, in denen die Adresse der Schreib- oder Leseanforderung liegt.
- Remote Node: Knoten, der eine Kopie des jeweiligen Cacheblocks besitzt.

**Nachrichten:**

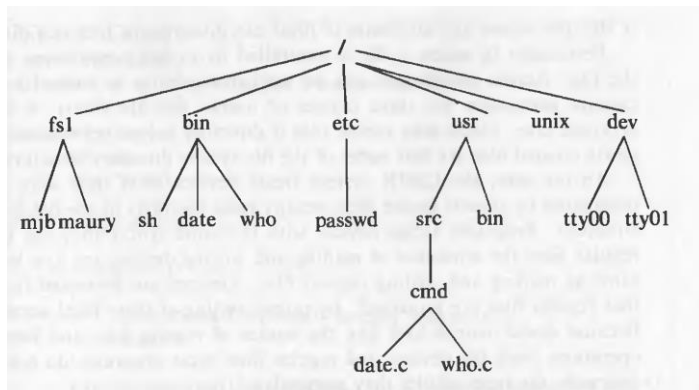


# Kapitel 13 Unix Systemsoftware

1. Sie sind mit der Architektur eines typischen Serverbetriebssystems vertraut.



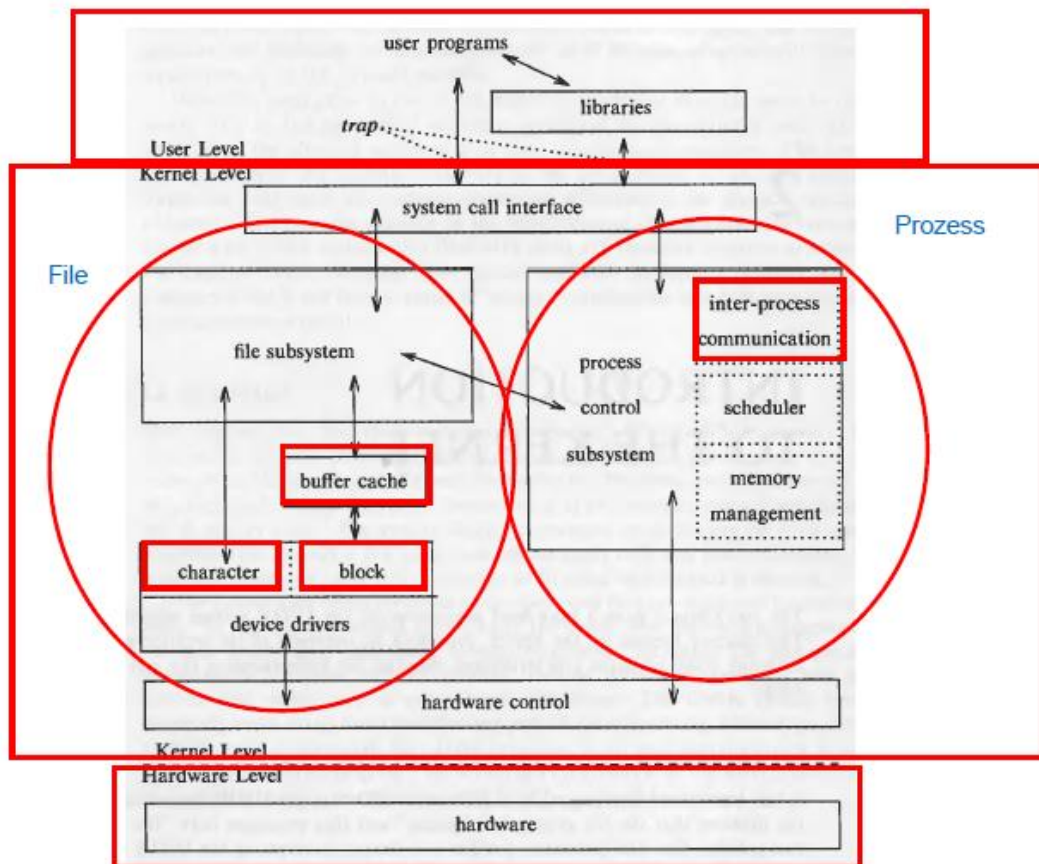
Sample System Tree



*etc: Systemconfig, ssh-daemon*

*unix: teile des Kernel*

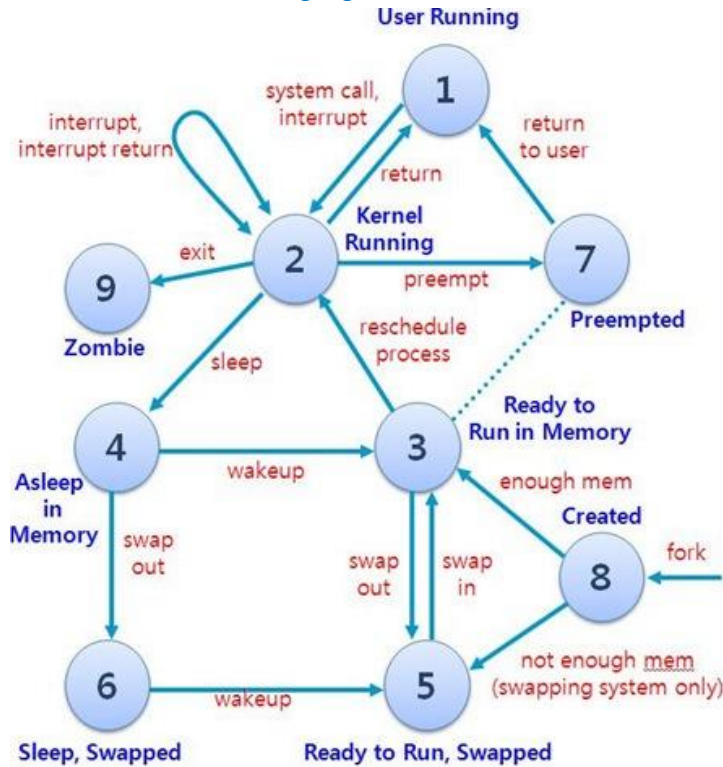
*dev: (device) tty=Keyboard*





2. Sie kennen die Relationen zwischen Prozessen und Threads im Betriebssystemkontext.

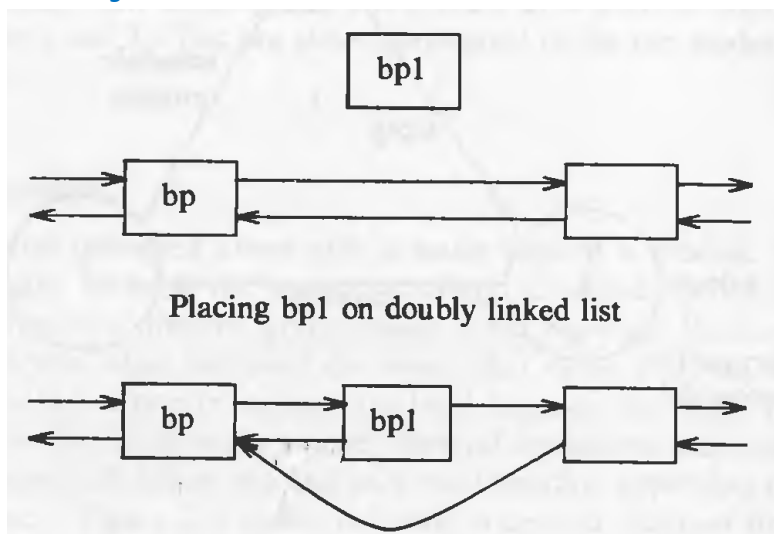
**Prozesszustände und Übergänge**



**Prozess Kontext**

- Benutzer Kontext
  - zugewiesener Adressraum und Daten
- Hardware Kontext (multitasking relevant)
  - Inhalte CPU Register
  - weitere wichtige Infos wie z.B: Seitentabelle
- System Kontext (Sicht OS)
  - Prozessnummer, geöffnete Dateien, Info Eltern und Kind Prozess, Prioritäten, etc.

**Auswirkung Kontextswitch**



Kernel: Damit kein Kontextswitch stattfindet,

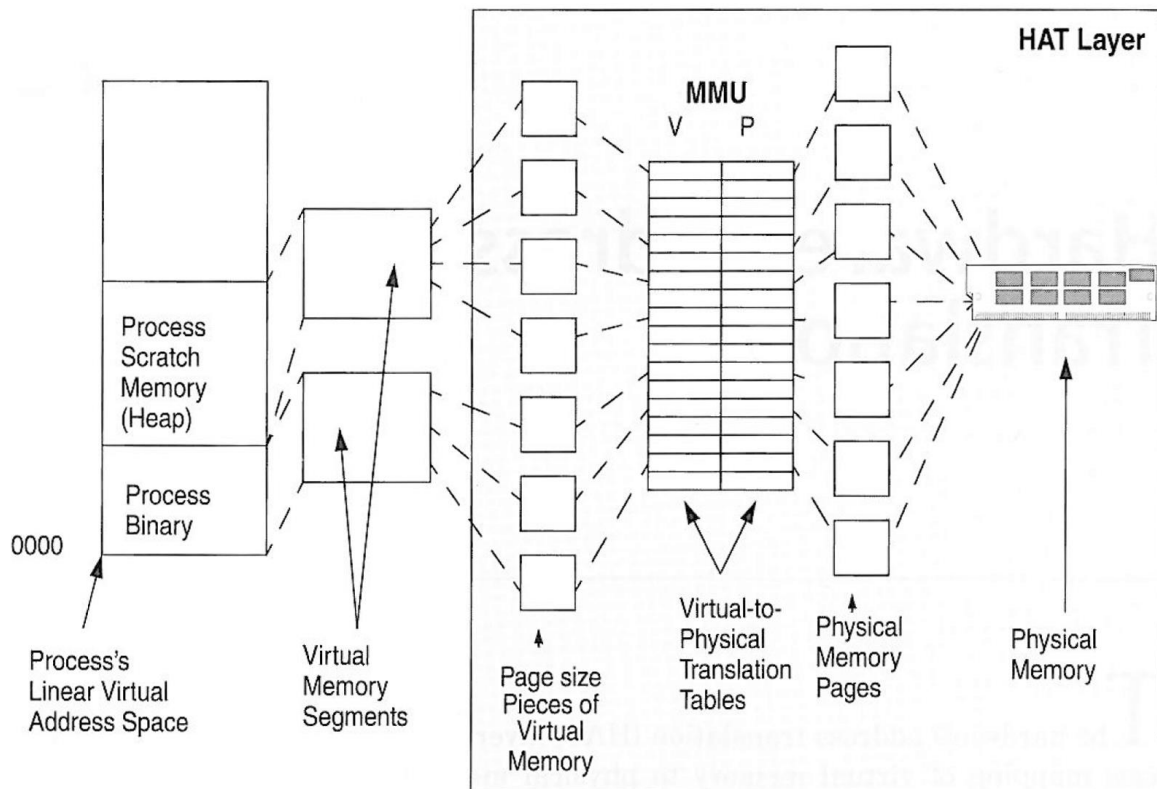
wird die Priorität vom Prozess auf Maximum gesetzt

So wird der Prozess nicht unterbrochen und die

Pointers können korrekt und alle angepasst werden



## Virtuelle zu Physikalische Adress-Übersetzung



HAT: Hardware Adress-Translation

### Thread Kontext

- Ein Prozess besteht aus mindestens 1 Thread
- Threads vom gleichen Prozess teilen sich:
  - Codesegment
  - Datensegment
  - Dateideskriptoren
- Jeder Thread hat einen eigenen Stack
- Jeder Thread hat seinen eigenen Code Pointer (Programm Pointer)
- Bei einem Thread-Wechsel ist kein vollständiger Prozesskontext-Wechsel nötig. (Prozesskontext-Wechsel ist teuer!)
- Die Kommunikation zwischen Threads ist durch den gemeinsamen Prozess Kontext auf natürlicher Weise gegeben.

### User Thread

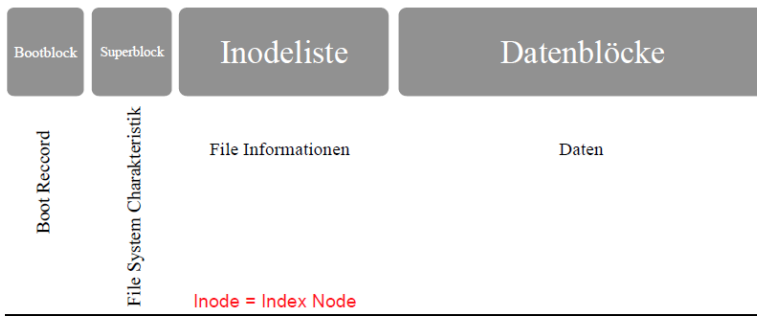
- Werden durch eine Library (vom Programmierer) implementiert
- Der Kern hat keine Kenntnis davon
- Müssen Kontrolle von sich aus abgeben
- Neigen mehr zu kooperativem als zu preemptivem Scheduling
- Implementieren manchmal das Scheduling selber (z.B: frühe Versionen von Java)

### Kernel Threads

- Fundamentale Einheit die zum Prozessor verlegt wird.
- Switching Kernel Threads geht sehr schnell.
- Werden für Kernel Tasks verwendet (process execution, interrupt handling)

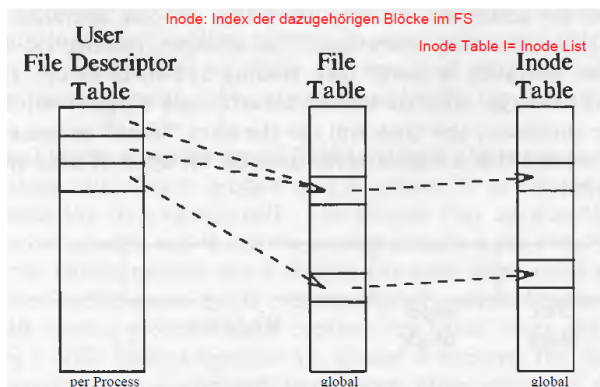
# Kapitel 14 File Systeme

## 1. Aufbau Dateisystem



### Superblock

- Grösse des Dateisystems (Grösse der Blöcke von Inodeliste / Datenblöcke)
- Anzahl der freien Blöcke im Dateisystem
- Liste der freien Blöcke im Dateisystem
- Index auf den nächsten freien Block in der Liste
- Grösse der Inodeliste
- Anzahl der freien Inodes im Dateisystem
- Liste der freien Inodes im Dateisystem
- Index auf die nächste freie Inode in dieser Liste
- Flag für durchgeführte Änderungen im Superblock



Ein **Inode** (auch I-Node, Kurzwort von englisch index node) ist ein Eintrag in einem Unix-Dateisystem, der **Metadaten einer Datei** enthält. In den meisten Unix-Betriebssystemen werden Dateien nur über ihre Inodes verwaltet.

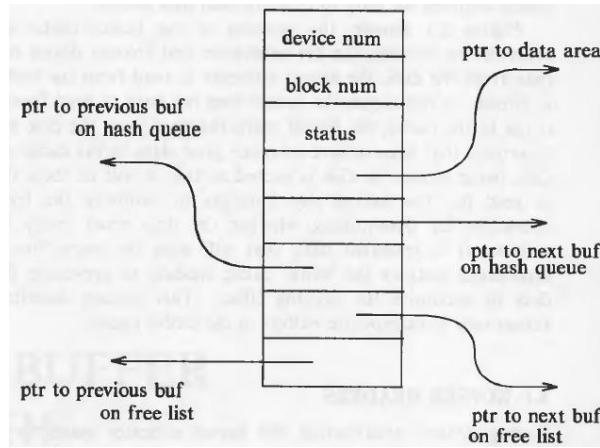
Jeder Inode ist durch eine eindeutige Inode-Nummer gekennzeichnet und verweist auf genau eine Datei. Umgekehrt gehört zu jeder Datei auch ein einziger Inode, der Attribute der betreffenden Datei enthält, die Zuordnung ist daher eineindeutig. Namen und Inhalt werden unabhängig abgelegt: **Dateinamen verweisen auf den Inode, dieser verweist auf den Speicherplatz des Datei-Inhalts**, ausgenommen für sehr kleine Dateien (bis zu etwa hundert Bytes): Deren Inhalt wird von den meisten heutigen Dateisystem-Implementierungen direkt im Inode abgespeichert.

### Ein Inode enthält

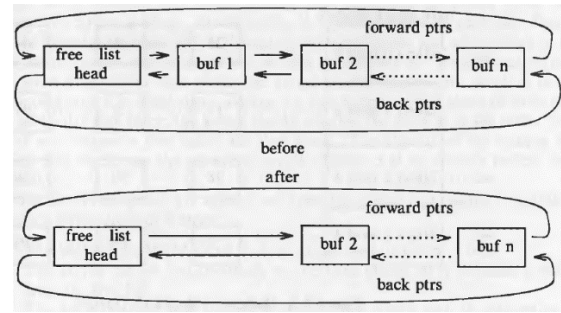
- die Zugriffsrechte auf die Datei (Eigentümer-, Gruppen- und sonstige Rechte)
- den Eigentümer der Datei und die bevorrechtigte Gruppe (UID, GID)
- den Dateityp (einfache Datei, Verzeichnis, Link, ...)
- die Größe der Datei (in Bytes)
- einen Referenzzähler (Anzahl der Verweise auf die Datei)
- das Datum der letzten Inode-Änderung (change time, ctime), des letzten Zugriffs auf die Datei (letzte Dateiöffnung/-ausführung, access time, atime) und der letzten Änderung der Datei (modification time, mtime)
- einen oder mehrere Verweise auf die Cluster (Blöcke), in denen der Inhalt der Datei liegt.

Ein Dateiname (samt Pfad) verweist stets eindeutig auf einen Inode, ist aber nicht in diesem gespeichert. Die Speicherung des Dateinamens wiederum findet in speziellen Verzeichnisdateien des Systems statt.

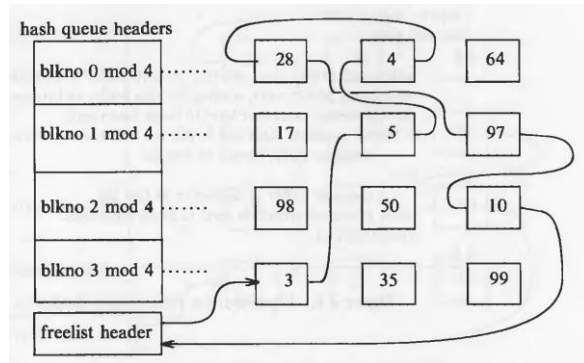
**Buffer Header**



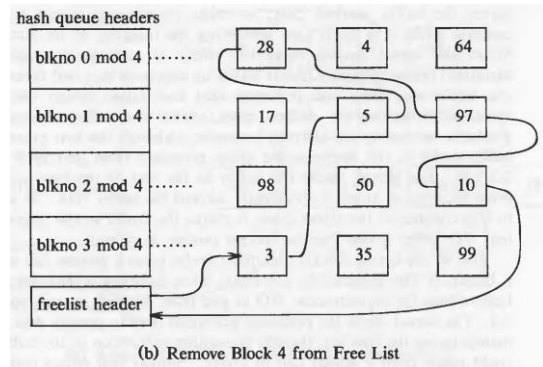
**Freie Liste**



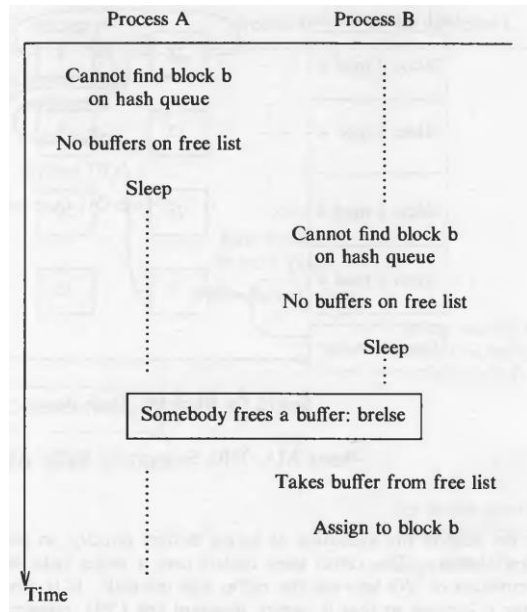
**Suche für Block 4 in der ersten Hash Queue**



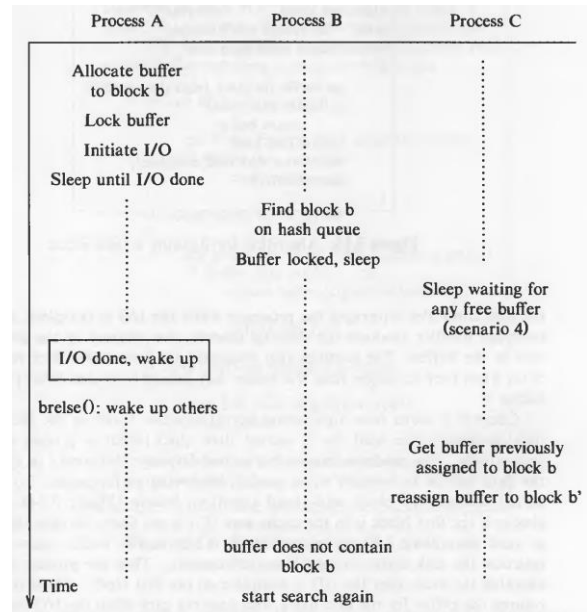
**Entfernen von Block 4 von der freien Liste**



**Rennen um einen freien Buffer**



**Rennen für einen locked Buffer**



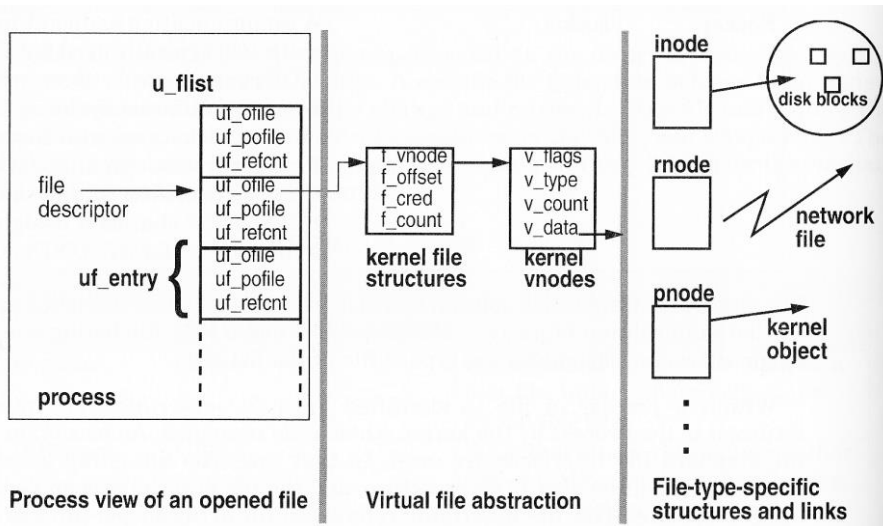
**Beispiel anhand des Dateisystems ext2**

Bei ext2 verweisen standardmäßig bis zu 12 Einträge im Inode auf je einen Cluster, in dem der Inhalt der Datei gespeichert ist.

Reichen diese 12 Cluster nicht aus, zeigt ein Eintrag im Inode auf einen Cluster, der dann die Verweise zu den eigentlichen Datenclustern enthält. Ein solcher Verweis wird als **einfach indirekter Block** bezeichnet. Bis zu **dreifach indirekte Blöcke sind möglich**, so dass die maximale Dateigröße je nach Blockgröße zwischen 16 GiB und 4 TiB liegen kann.

2. Sie kennen die Prozedur eines Kernel-Filezugriffs.

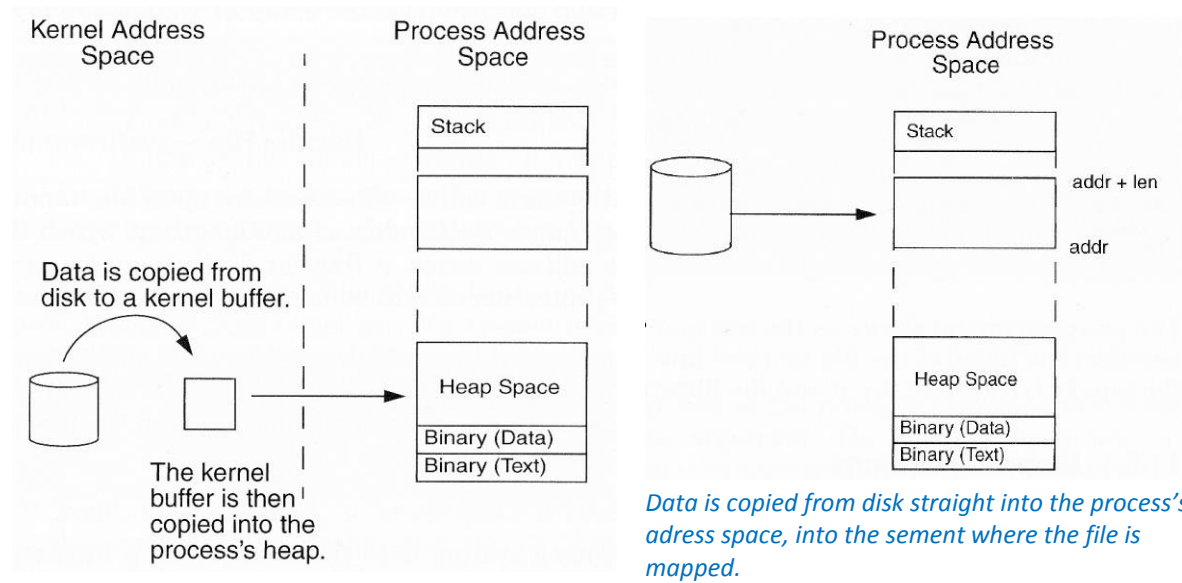
*File relevante Strukturen*



3. Sie wissen wie der Kern die Filezugriffe beschleunigt.

*File lesen mit read()*

*Memory mapped File I/O*



**Kapitel 15 Cluster & Grid Systeme**

1. Sie kennen die Einsatzgebiete von hochverfügbaren Systemen.

$$C_{\text{downtime}(t)} = t_{\text{down}} * P_{\text{uptime}} / t_{\text{up}}$$

- $C_{\text{downtime}(t)}$ : Die Kosten vom Ausfall
- $t_{\text{down}}$ : Zeitdauer des Ausfalles
- $P_{\text{uptime}}$ : Gewinn während des Normalbetriebes  $t_{\text{up}}$
- $t_{\text{up}}$ : Zeitdauer wo das System Normal funktionierte

*Gute Gründe für High Availability*

- Für Geschwindigkeit und Hochverfügbarkeit (oder Beides)
- Erhöhen meistens die Service Verfügbarkeit
- Ausfall ist kritisch für das Business

## 2. Sie wissen wie hochverfügbare Systeme aufgebaut sind.

### Arbitration

Der Arbitrer oder die Arbitrationslogik (von lat. arbiter „Richter“) ist eine Funktionseinheit in Form einer elektrischen, digitalen Schaltung oder einer Softwareroutine, die Zugriffskonflikte oder Zugriffskollisionen löst oder priorisiert.

### Asymmetric Arbitration

- Angewendet wenn die Priorität der Kandidaten klar ermittelt werden kann.
- Beispiel: SCSI 2 Protokoll setzt die höchste Priorität auf Target ID=7, CDROM auf ID=6 und die schnelle Boot Disk startet auf ID=0

### Symmetric Arbitration

- Verwendet wenn die Kandidaten (Peers, Teilnehmer) sind und ein Prioritäten basiertes Auswahlverfahren nicht durchführbar ist.
- Sample: SMP oder 100BASE-T (zwei oder mehr Nodes teilen sich ein gemeinsames Medium. (Schwierige Vorhersage)

### Wahl und Beschlussfähigkeit (Quorum)

- Meist verbreitete Methode der Arbitration. Für die Auswahl der Zugehörigkeit zu einem Cluster verwendet. Auswahl durch die Majorzwahlverfahren der Mitglieder.
- Basiert auf: Mehrheit > Hälfte von allen Stimmen
- Jeder Node hat eine Stimme
- Eine Mehrheit der Nodes müssen fähig sein miteinander zu kommunizieren.
- Mehrheit = Hälfte der Nodes + 1
- Wenn diese Kondition nicht erreicht wird geht der Node in Panik.
- Abschalten eines Nodes in einem 2 Node Cluster bringt den Anderen auch runter.
- Quorum erledigt dieses Problem weil er  $V_n-1$  Stimmen verteilt.

## 3. Sie kennen die theoretischen und praktischen Probleme mit hochverfügbaren Systemen.

### Split Brain

- ein Fehler verursacht neue Cluster Rekonfiguration in mehrere Partitionen und keine der Partition weiss etwas von der Anderen.
- Der neu formierte Cluster erzeugt Kollisionen in den gemeinsam benutzten Ressourcen.
- Netzwerk Adressen sind dupliziert, weil er denkt er sei alleine.

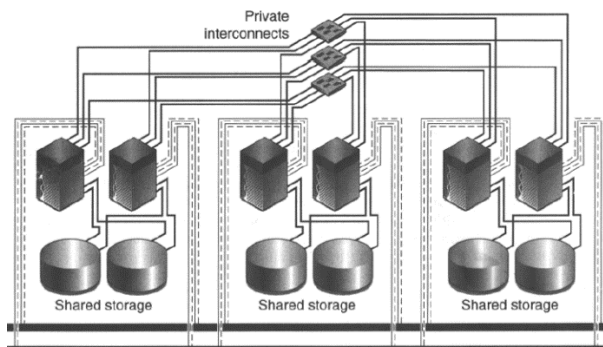
### Amnesia

- Nach einem Fehler (re)startet einer der Cluster Nodes neu mit alten Cluster Konfigurationsdaten.
- Diese alten Konfigurationsdaten scheinen korrekt und ein neuer Cluster wird mit diesen veralteten Daten erzeugt.

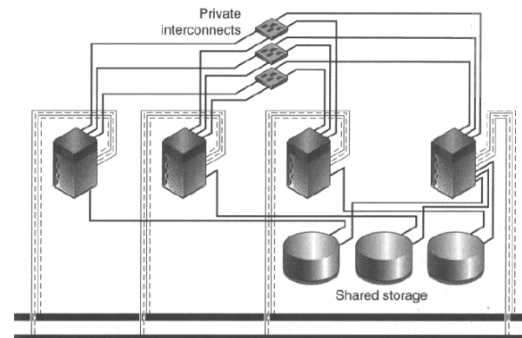
## 4. Sie kennen die Architekturen von hochverfügbaren Systemen.

- **Activ / Passive**
- **Active / Active**

### Pair Topology

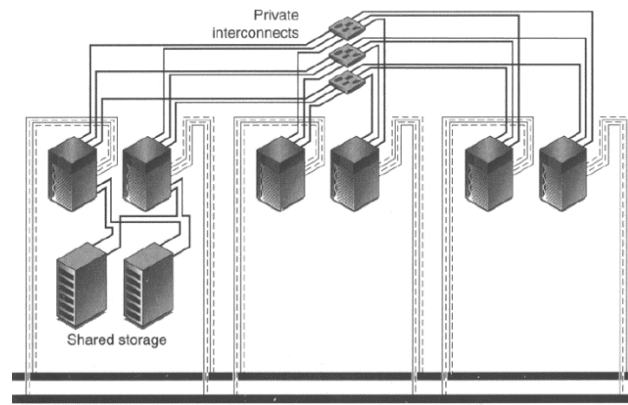


### N+1 Topology

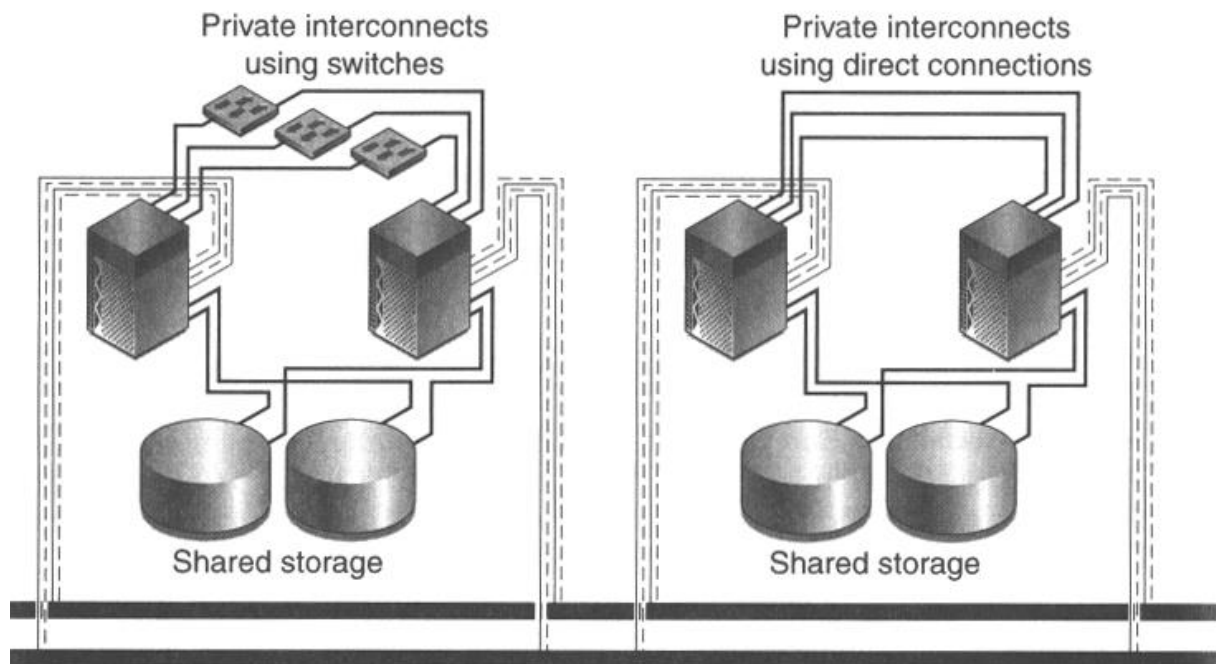




**Pair+M Topology**



**Cluster Interconnects**



**Oracle Real Application Cluster**

**Pro**

- Verfügbarkeit
- Skalierbarkeit
- Investitionsschutz

**Kontra**

- Komplexität
- Lizenzkosten
- Unterhaltskosten

# Kapitel 17 Server Virtualisierung

1. Es können auch Fragen bezüglich der Übung „Lab 17“ gestellt werden. Speziell wenn es um Konzepte geht.

*Siehe LAB 17*

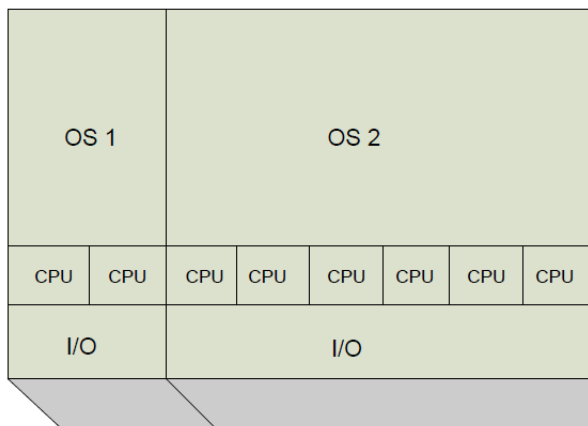
2. Sie kennen die Gründe, warum virtualisiert wird.

*Virtualisierung ist Ressourcenmanagement*

3. Sie kennen die verschiedenen Virtualisierungsarten.

- *Hardware Partitionierung (auf Hardwareebene)*
- *Ressource Management (auf Softwareebene im OS)*
- *Hardware Virtualization*
  - *Binary Translation*
  - *Para Virtualization (SW Assist)*
  - *Full Virtualization (HW Assist)*
- *OS Virtualization*

## Hardware Partitioning

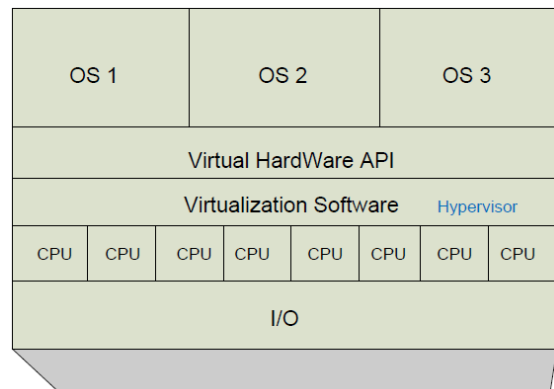


Nachteil: keine fein granulare Unterteilung möglich.

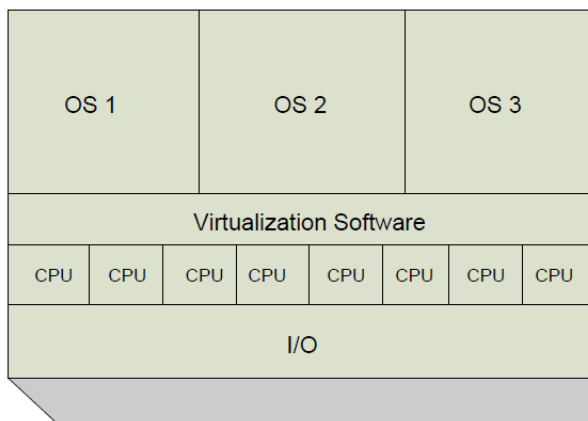
*IBM: LPAR (Logical Partition)*

*Sun: LDom (Logical Domain)*

## Paravirtualization

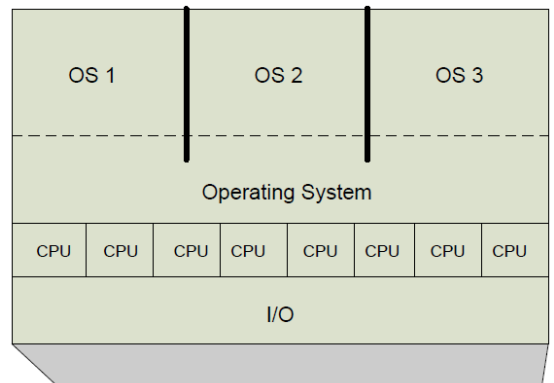


## Full Virtualization



*Intel VT oder AMD-V*

## Containers



*Solaris Zones, BSD Jails or Linux OpenVZ*



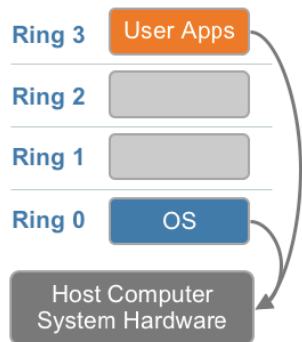
Übersicht Virtualisierungsarten

<p><b>Hardware Partitioning</b></p> <p>Server ist weiter unterteilt, wobei Jeder Teil wieder ein Os laufen hat</p>	<p><b>Bare Metal Hypervisor</b></p> <p>Hypervisor stellt time sharing fuer alle Ressourcen zur Veruegung</p>	<p><b>Hosted Hypervisor</b></p> <p>Hypervisor verwendet OS Services fuer time sharing aller Ressourcen</p>
<p><b>Physical partitioning</b> S/370 SI-to-PP and PP-to-SI, Sun Domains, HP nPars</p> <p><b>Logical partitioning</b> pSeries LPAR, HP (PA) vPars</p>	<p><b>Hypervisor Software/Firmware</b> läuft direkt auf dem Server</p> <p>System z PR/SM and z/VM POWER Hypervisor VMware ESX Server Xen Hypervisor</p>	<p><b>Hypervisor Software</b> läuft auf einem Host Betriebssystem</p> <p>VMware GSX Microsoft Virtual Server HP Integrity VM User Mode Linux</p>

SMP: Symmetric Multiprocessing

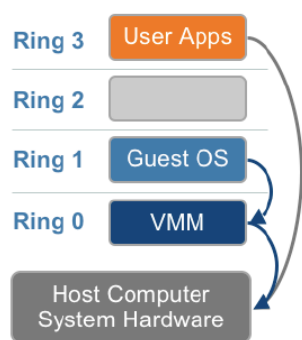
- Hardware Partitionierung unterteilt den Server weiter wobei jeder Teil sein eigenes OS laufen lassen kann.
- Hypervisor bestehen aus einer dünnen Schicht Code um fein-granulares dynamisches Ressourcen Sharing zu ermöglichen.
- Typ 1 Hypervisor ist dominant bei Servern. Typ 2 Hypervisor ist dominant bei Workstation.

Direct Execution



- x86 kennen 4 Levels von Privilegien.
- User Level Programme laufen nicht Privilegiert Ring 3
- Das OS braucht HW Zugriff Ring 0
- Ein Virtualisierungsschicht muss unter Ring 0 gelegt werden!

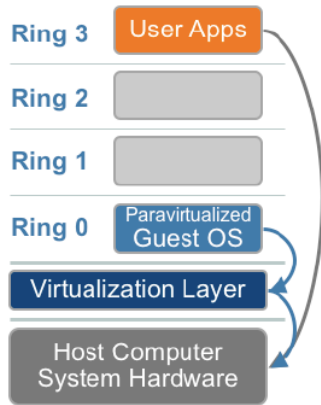
Full Virtualization mit Binary Translation



- Kombination zwischen direkter Ausführung und binary translation.
- User Level Code wird direkt ausgeführt.
- Übersetzter Kernel Code ersetzt nicht virtualisierbare Instruktionen.
- Keine HW Unterstützung nötig.
- Gast OS ist voll abstrahiert, hat keine Ahnung dass es in einer VM läuft.

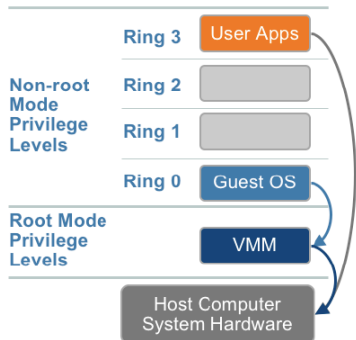
User Applikationen werden nicht übersetzt, sie laufen direkt. Binary Translation werden nur angewendet wenn der Gast OS Kernel aufgerufen wird.

**OS Unterstützte (Para) Virtualisierung**



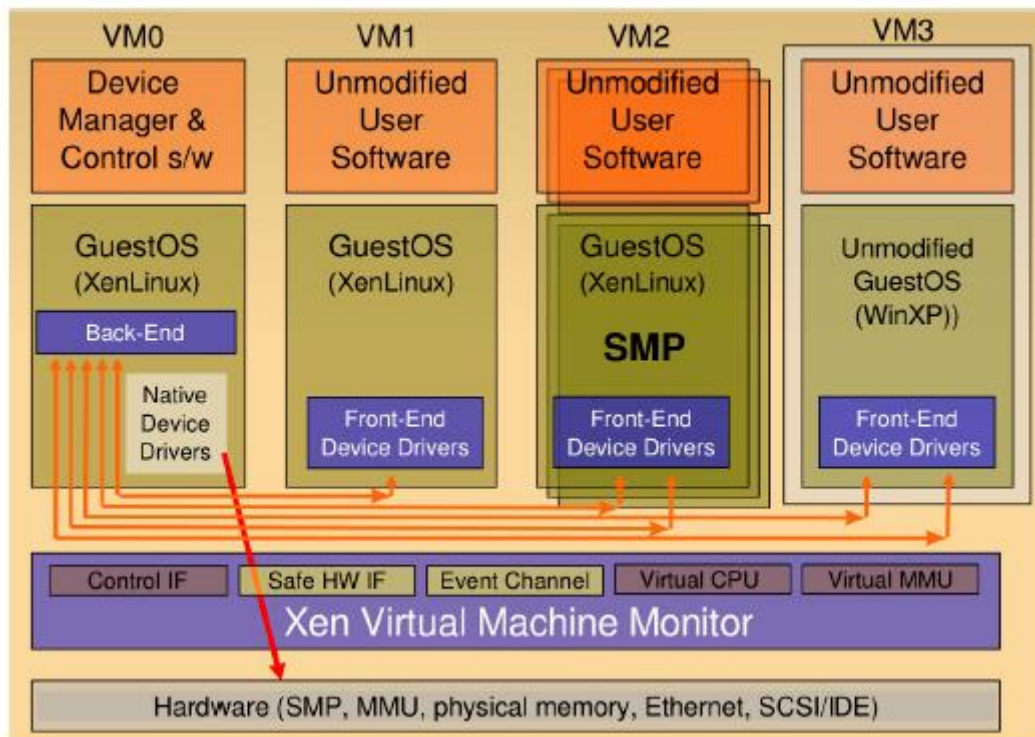
- Para (griechisch = neben) bezieht sich auf Kommunikation zwischen Gast OS und Hypervisor um die Geschwindigkeit zu erhöhen.
  - Kernel vom OS muss modifiziert werden um nicht virtualisierbare Instruktionen mit Hypercalls zu ersetzen die direkt mit dem Virtualization Layer Hypervisor kommunizieren.
  - Der Hypervisor stellt Hypercall Interfaces zur Verfügung.
  - Keine unmodifizierte Gast OS
  - Paravirtualisierung ist nicht so verschieden von Binary Translation.
  - BT übersetzt „kritischen“ in „harmlosen“ Code.
  - Paravirtualisierung macht dasselbe aber im Source Code.
  - Änderungen im Source Code erlaubt grössere Flexibilität.
  - Para Virtualisierung braucht keine „Laufzeit-Übersetzung“ und wird daher schneller ausgeführt.
- ➔ Unveränderte Betriebssysteme können nicht ausgeführt werden.

**HW unterstützte Virtualisierung**



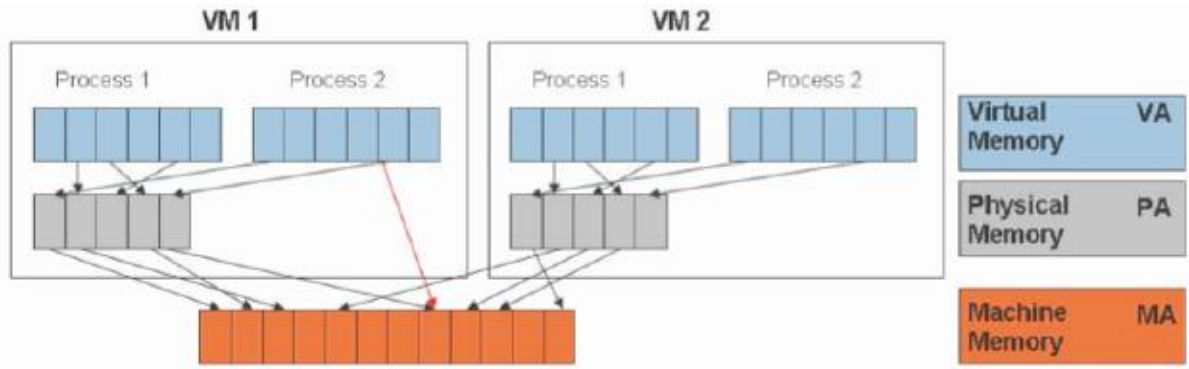
- Neuer CPU Modus eingeführt.
- Erlaubt Ausführung unterhalb Ring 0.
- Privilegierte und sensitive Calls trap automatisch zum Hypervisor ohne binary translation.
- Der Gast Status wird in virtual machine control structures abgespeichert.
- CPUs seit 2006 erhältlich.
- Neu 3Bit anstelle von 2Bit für die Ringe

**Vereinfachte Front-End Driver**



Der Hypervisor stellt Interfaces für kritische Kern Operationen zur Verfügung. I/O Devices in der VM sind „nur“ Pointer zu reellen „native“ Drivers in einer privilegierten VM (genannt Domain 0). Keine Emulation oder Übersetzung erforderlich!

**Memory Virtualisation**



**ohne HW Virtualisierung**

- Applikation sieht kontinuierlichen Adressbereich.
- OS mapped virtual Page # zu physikalischen Page #.

**mit Hypervisor**

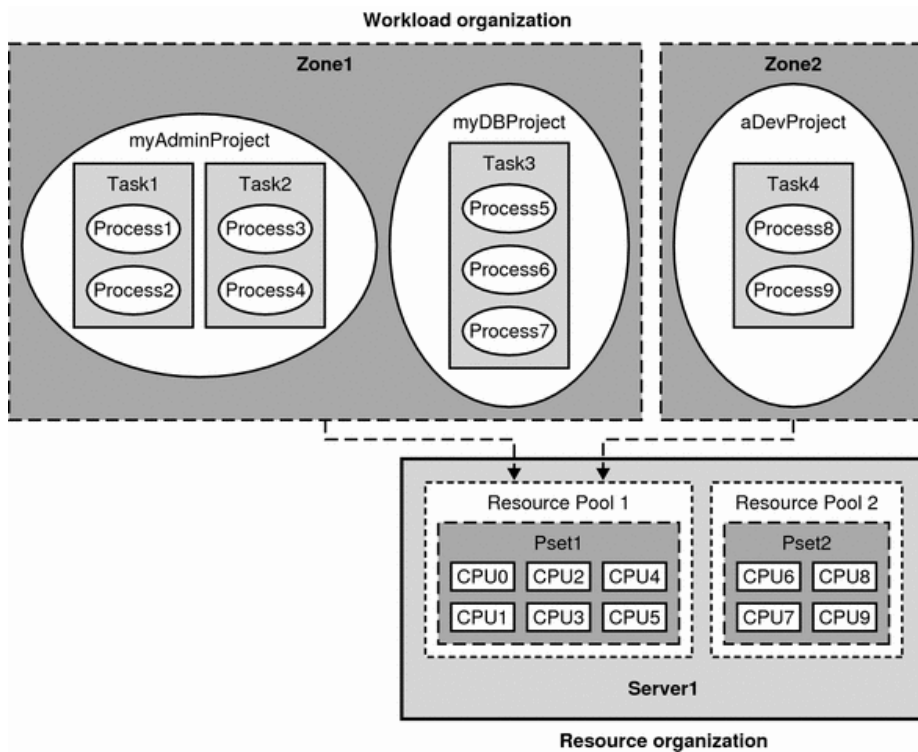
- mehrere Gast OS auf einem Hypervisor benötigen eine weitere Memory Virtualisierung.
- Das Gast OS hat keinen direkten Zugriff mehr auf das Memory.

Problem: Memory Verbrauch -> 3 Möglichkeiten um Memory zu sparen:

- Page Sharing
- Page Patching
- Page Compression

**Ressourcen Management in Zonen**

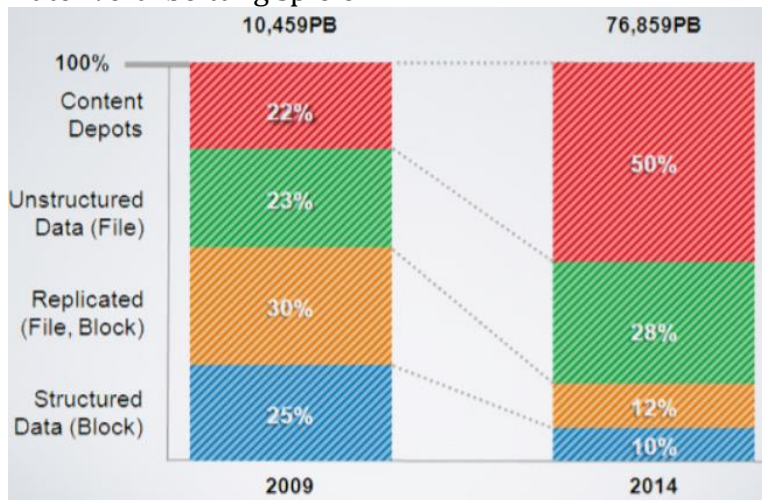
Das Ressourcen Management verhindert ein Gegenseitige Beeinflussung der Zonen, sowie das Verhungern einer Ressource.



## Kapitel 19 Storage Systems

Massgebend für den Stoff ist die Vorlesung von Gaetano Bisaz, HDS. Die Slides sind auf ILIAS und sie wurden vor der Vorlesung ausdrücklich darauf hingewiesen Notizen zu machen.

1. Sie können Daten klassifizieren und wissen welche Rolle diese in der zukünftigen Datenverarbeitung spielen.



2. Sie wissen aus welchen Teilen ein klassisches Disksubsystem besteht.

*Disk-Controller + Disks*

3. Was ist DAS / NAS / SAN?

**DAS (Direct Attached Storage)**

*Bezeichnet an einen einzelnen Host angeschlossene Festplatten, die sich in einem separaten Gehäuse befinden.*

**NAS (Network Attached Storage)**

*Bezeichnet einfach zu verwaltende Dateiserver. Allgemein wird NAS eingesetzt, um ohne hohen Aufwand unabhängige Speicherkapazität in einem Rechnernetz bereitzustellen.*

**SAN (Storage Area Network)**

*Als Storage-Area-Network (SAN) bzw. Speichernetzwerk bezeichnet man im Bereich der Datenverarbeitung ein Netzwerk zur Anbindung von Festplattensubsystemen (Disk-Array) und Tape-Libraries an Server-Systeme. Storage Area Networks sind für serielle, kontinuierliche Hochgeschwindigkeitsübertragungen großer Datenmengen konzipiert worden. Sie basieren heute für hochverfügbare, hochperformante Installationen auf der Implementierung des Fibre-Channel-Standards, bei KMU aus Kostenüberlegungen auch auf IP.*

4. Sie können die Block/File Speicherprotokolle beschreiben.

*DAS und SAN: Block-Speicherzugriff*

*NAS: File-Speicherzugriff*

5. Sie kennen Vor-/Nachteile von Cloud Storage.

### Vorteile

- *Kostenersparnisse*
- *Komfort*
- *Flexibilität*
- *Skalierbarkeit*

### Nachteile

- *Sicherheit*
- *Abhängig von einem Provider*
- *Benötigt schnelle Anbindung an das Internet*
- *Benutzer weiss nicht, wo die Daten gespeichert sind*
- *Kontinuität des Providers*

## 6. Sie können „scale out“ v. „scale up“ erklären.

### **vertikale Skalierung (scale up)**

- *Unter vertikaler Skalierung versteht man ein Steigern der Leistung durch das Hinzufügen von Ressourcen zu einem Knoten/Rechner des Systems. Beispiele dafür wären das Vergrößern von Speicherplatz, das Hinzufügen einer CPU, oder das Einbauen einer leistungstärkeren Grafikkarte.*

### **horizontale Skalierung (scale out)**

- *Im Gegensatz zur vertikalen Skalierung sind der horizontalen Skalierung keine Grenzen (aus Sicht der Hardware) gesetzt. Horizontale Skalierung bedeutet die Steigerung der Leistung eines Systems durch das Hinzufügen zusätzlicher Rechner/Knoten. Die Effizienz dieser Art der Skalierung ist jedoch stark von der Implementierung der Software abhängig, da nicht jede Software gleich gut parallelisierbar ist.*

## 7. Sie können die richtige Storage Architektur bestimmen (block/file)

### **Recovery Time Objective (RTO)**

- *Wie lange darf der Ausfall dauern?*
- *Was sind meine Zeiten?*

### **Recovery Point Objective (RPO)?**

- *Wo fahre ich weiter?*
- *Welche Daten darf ich verlieren?*