12.–15.09.2010
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

# Hochverfügbar

## Transaktionssysteme auf Basis von Java und UNIX

Matthias Schorer
CSC

# Über den Sprecher

- **Seit 2009**
  Strategic Management Consultant
  bei CSC

- **1998-2008**
  Technischer Chefarchitect der FIDUCIA IT AG

- **1993 – 1998**
  Technical Director CIMCO Ltd., Bangkok, Thailand

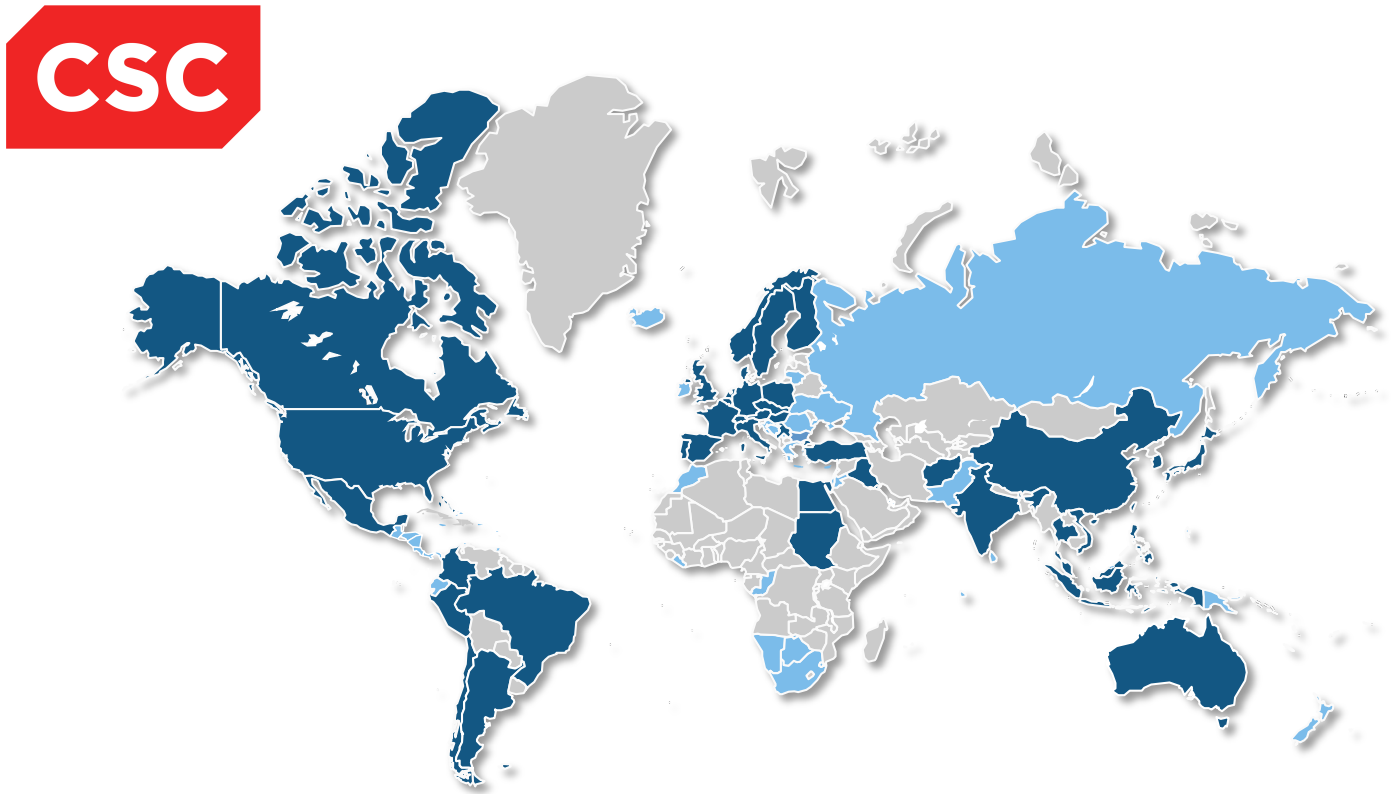
# Agenda

- **Wer ist CSC (schamloser Werbeblock)**
- Wer früher stirbt ist länger tot
- Von XML, dem Teufel und dem Weihwasser
- Alles voll normal oder was?
- Caching, aber richtig!
- Schlangestehen ist cool!
- Hochstapler

# CSC

CSC zählt zu den weltweit führenden Dienstleistungsunternehmen im Bereich der Informationstechnologie (IT). Mit seinen maßgeschneiderten Lösungen und Services sowie den Managed Services unterstützt CSC Kunden in allen wichtigen Wirtschaftszweigen. Das Angebot von CSC umfasst Systemdesign und Systemintegration, Informationstechnologie und Business Process Outsourcing, Applikationssoftwareentwicklung, Web und Applikation Hosting, Auftragsunterstützung und Managementberatung.
Das Unternehmen mit Hauptsitz in Falls Church, Virginia, hat rund 95.000 Mitarbeiter und erwirtschaftete in den zwölf Monaten bis zum 2. Juli 2010 einen Umsatz von 16,2 Milliarden US-Dollar. Weitere Informationen über CSC in Deutschland finden Sie auf der deutschen Website von CSC unter www.csc.com/de oder unter www.csc.com.

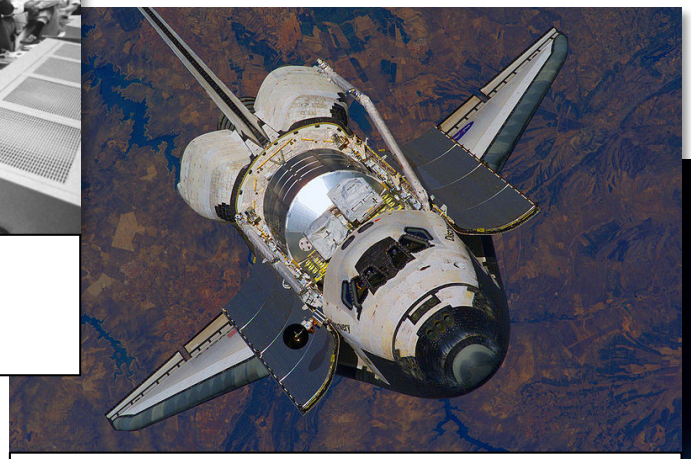
# CSC - 50 Jahre, 90 Länder, 95.000 Mitarbeiter



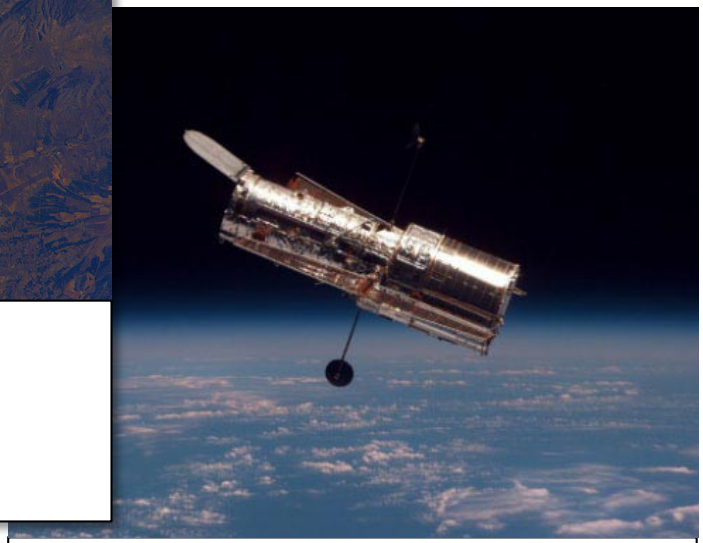

Consulting, Systems-Integration, IT- und BP-Outsourcing, Cloud Computing

# Wir wissen was „Mission Critical“ heißt

Seit 1961 Partner der NASA. CSC entwickelte Software und Systeme für das Apollo Programm



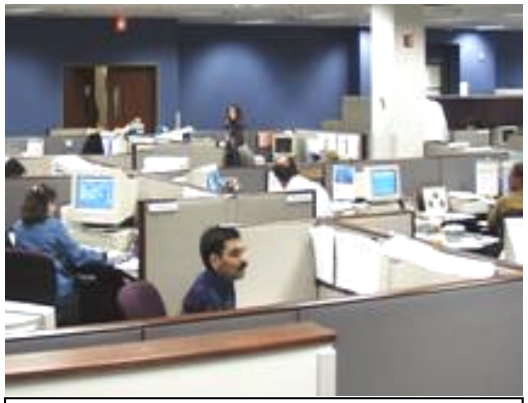
Seit 1971 sind CSC Ingenieure maßgeblich am Space Shuttle Programm beteiligt. CSC Payload Specialist Ron. A. Parise (+ 2008) flog zwei Shuttle Missionen.



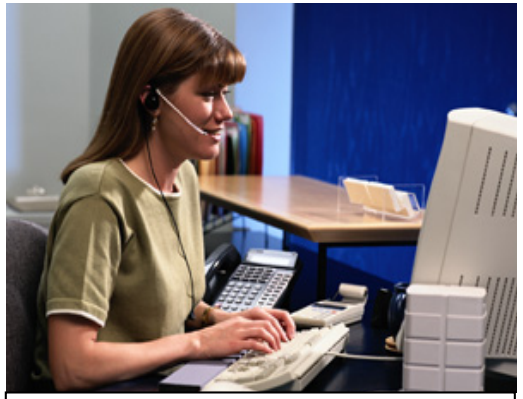
1990 - die Software für das Hubble Weltraumteleskop kommt von CSC.

# Global aufgestellte Computing Ressourcen

- 40 Rechenzentren
- 112.740+ MIPS
- 5.600+ Terabytes Storage
- 1.000.000 Desktops
- 66.000+ Server



- 51.200 WAN- Ports
- 1,46 Mio.+ LAN- Ports
- 860.000+ Voice- Ports



- Mehr als 12 Mio. Help-Desk-Calls in ca. 50 Ländern jährlich
- in 26 Sprachen

# CSC in Deutschland

- Deutsche Großbanken
  - Post Merger Unterstützung
- Bundesinnenministerium
  - IT-Konsolidierung, Green-IT
  - elektronischer Personalausweis


- BMW
  - Erweiterung des Testmanagementsystems
- Zürich Financials
  - Betrieb der kompletten IT für mehr als 55.000 Mitarbeiter

# Agenda

- Wer ist CSC (schamloser Werbeblock)
- **Wer früher stirbt ist länger tot**
- Von XML, dem Teufel und dem Weihwasser
- Alles voll normal oder was?
- Caching, aber richtig!
- Das Geheimnis hinter linearer Skalierbarkeit
- Hochstapler

# IBM Mainframes – totgesagte leben länger

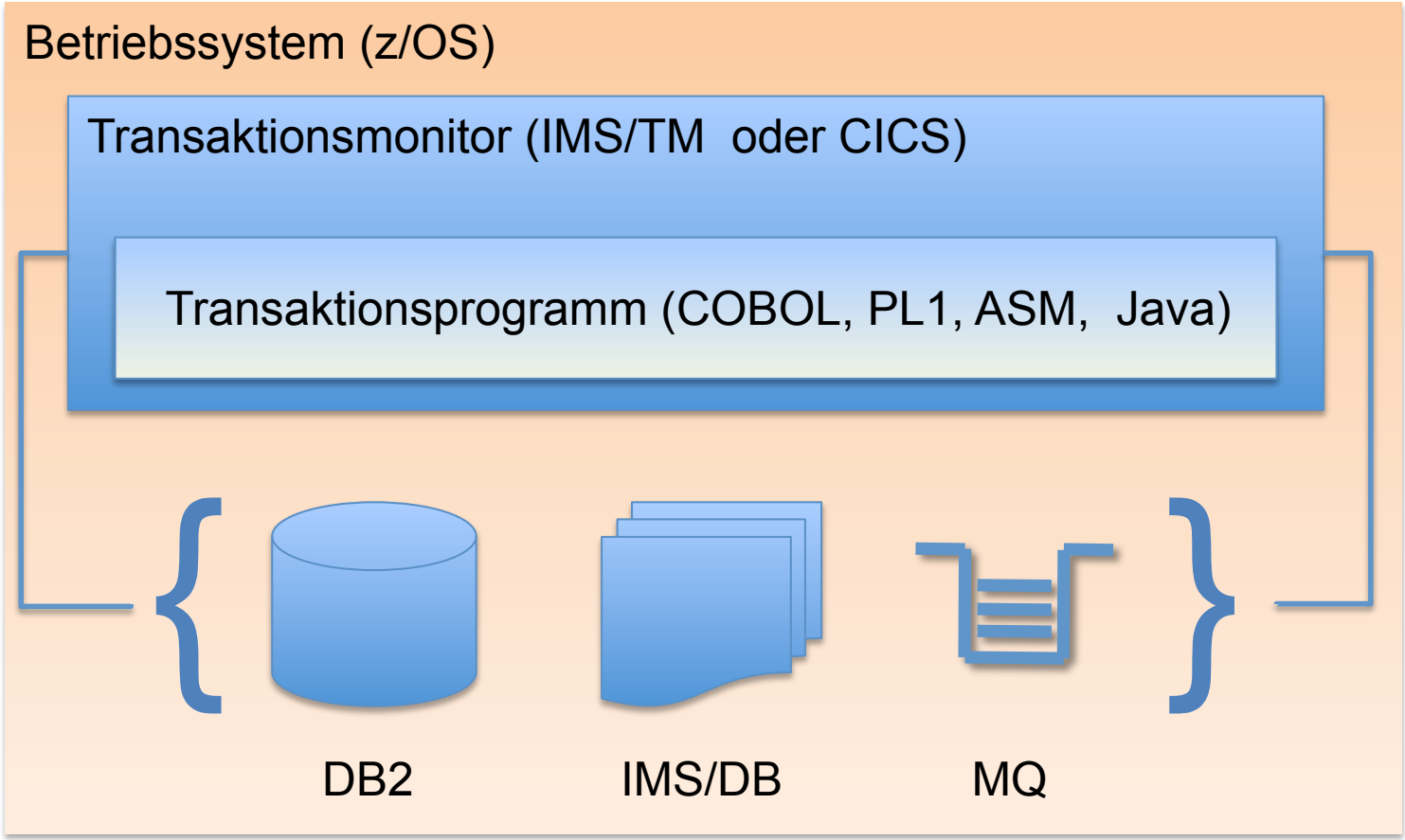

- **Seit ca. 1960 !**
- **Virtualisierung seit 1972!**
- **Fehlertoleranz**
  - CPU (Lockstep)
  - Speicher (Hardware-Speicherschutz)
  - Physikalische Systeme (parallel Sysplex)
- **Skalierbarkeit**
- **Verfügbarkeit (99.999% ~ < 6 Stunden downtime / Jahr)**
- **Riesige Datenbanken (Terrabytes)**
  - Hierarchisch (IMS/DB) und relational (DB2)
- **Extrem hohe Transaktionsraten (zehntausende / Sek.)**
  - Transaktionsmonitore IMS/TM und CICS
- **Batchprocessing**

# Mainframes – ein paar Zahlen

- CICS Transaktionsmonitor seit 1968
- Wird von mehr als 16.000 Unternehmen eingesetzt
- Von den 2.000 weltweit größten Unternehmen setzen 90% auf CICS
- CICS verarbeitet pro Tag weltweit 20 Milliarden Transaktionen (dies sind 231.481 Transaktionen pro Sekunde)

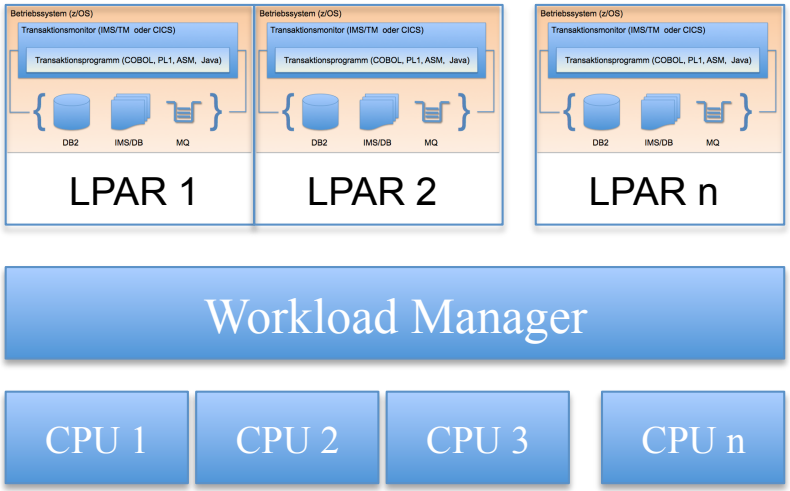
Quelle: Einführung in z/OS und OS/390, Herrmann, Krebschull, Spruth

# Mainframe – ACIDity



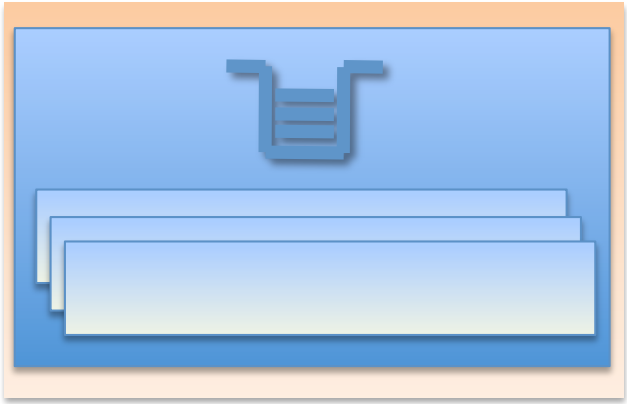

ACID: Atomicity, Consistency, Isolation, Durability

# Mainframe – Skalierung



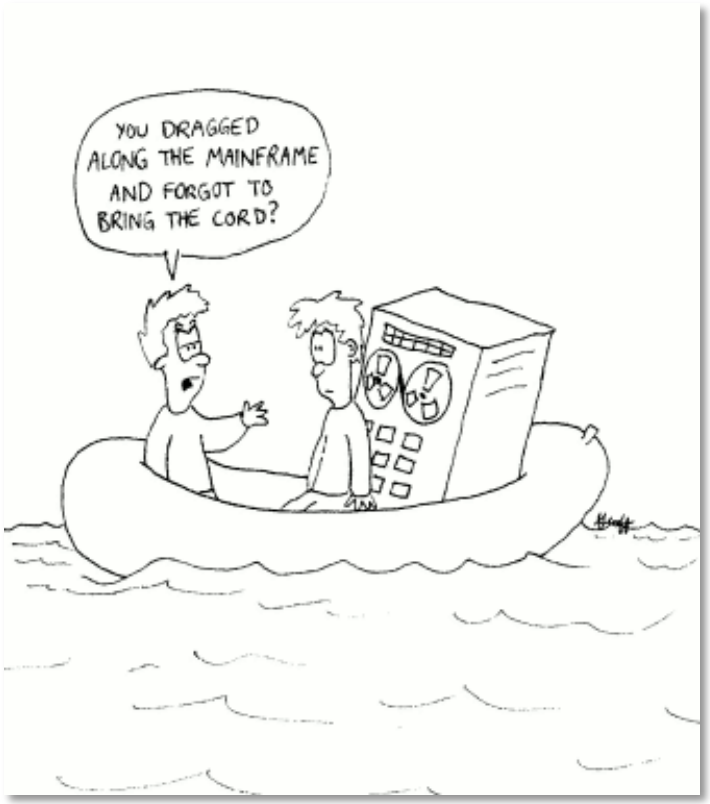

Horizontale Parallelisierung der Verarbeitung

Vertikale Parallelisierung der Verarbeitung



# Faktum

- Alle Probleme über die wir mit modernen IT-Systemen stolpern, sind auf Mainframes schon einmal gelöst worden....
- Lernt von den Leuten die dabei waren als es passierte...
- ... so lange es sie noch gibt!



# Die Aufgabe:

- Baue ein IT System das folgende Aufgaben erfüllt:
  - Verarbeitet mehr als 2.000 Transaktionen pro Sekunde (24*7)
  - Verarbeitet gleichzeitig große Batchjobs
  - Greife performant auf eine Datenbank > 25TB zu
  - Beliebig horizontal und vertikal skalierbar
  - 100% transaktionssicher
  - Verteilte Verarbeitung
  - Verfügbarkeit >= 99.999%
- Nutze Java, UNIX und Open Source

# Agenda

- Wer ist CSC (schamloser Werbeblock)
- Wer früher stirbt ist länger tot
- **Von XML, dem Teufel und dem Weihwasser**
- Alles voll normal oder was?
- Das Geheimnis hinter linearer Skalierbarkeit
- Schlangestehen ist cool!
- Hochstapler

# Da fliegt mir doch das Blech weg...

- „Es ist mir unbegreiflich warum Entwickler XML für Zwecke einsetzen, für die es einfach nie gedacht war...“
- „Wenn ich zwei Systeme die mir gehören miteinander verbinden muss, nehme ich auf keinen Fall XML sondern etwas kompakteres...“

Tim Bray Co-Author of the XML and Atom-Web Standard

# SOAP = XML²

- **Aufgabe:** Rufe einen Webservice der eine Liste zurückgibt.

- **Eingabe:**
  - custID CHAR 50

Summe 50 Bytes

- **Ausgabe ist eine Liste mit den Feldern:**
  - custID CHAR 50
  - userID CHAR 50
  - requestID CHAR 20
  - fileName CHAR 50
  - fileStatus CHAR 1
  - uploadDate DATE (8 BYTES)

Summe 179 Bytes

# Eingabe a la SOAP

<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"><SOAP:Header xmlns:SOAP="http://
schemas.xmlsoap.org/soap/envelope/">\n\t\t\n\t<i18n:international xmlns:i18n="http://www.w3.org/2005/09/ws-i18n"><locale
xmlns="http://www.w3.org/2005/09/ws-i18n">en-gb</locale></i18n:international><wsse:Security xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"><Signature xmlns:SOAP="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns="http://www.w3.org/2000/09/
xmldsig#"><SignedInfo><CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><Reference
URI="#A0050568B-1364-71DF-9EBF-996D04C85A9E"><Transforms><Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></Transforms><DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>zkdy1Ok5td+wgJY6/
oY60Izq87M=</DigestValue></Reference></SignedInfo><SignatureValue>lbkuZYzj+T/
kUjwUG0CwhPONDqc9cSQGjqwAoGiMil5ZGtQJ6EiVtSLNZHf2aHkPR8WY0umh2d1mpz
j23Zi3Rzksu9LbWYiVhc7xpM6m69n6plEu8UMsM6riQCj8jzERJWAtPwaujN5jnlVE1GTF[redacted]hqb7RiqErPll=</
SignatureValue><KeyInfo><X509Data><X509Certificate>MIIB3DCCAUWgAwIBAgIPl[redacted]YK
+eFPzMA0GCSqGSIb3DQEBBQUAMDAxDzANBgNVBAoTBnN5c3RlbTEdMBsGA1UE[redacted]bjc2NzZXBhY2xvdW
QwHhcNMTAwNjEwMTUxMzQwWhcNMjAwNjA3MTUxMzQwWjAqMRcwFQYDVQQDEw5z[redacted]bjEPMA0GA1UE
ChMGc3lzdGVtMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCwh8Rz7+dyHsh4VrQqH[redacted]X5eZpZHKCPi
l7YaRDn5A6m/
oLVUi92BBqSMYBCxknptHLFeHhaJixpC2cayGdyc9rdmKa5shUHqmw0tj44fTSAjzhcd5VLUQaiRh7v9YW[redacted]7QX+
+WFviD/RrjmOjTQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBABzY/kAbopNX0/Hi9Z3NYTvpl00MwNk5N
+CK99pT7ia278xJY5qkoFfS644QRLmyv6Etbb2/MozXy6h0wTm7iCkPqSb52Yp
+YyMhZPsK9HcudjC0djxwARf7OWQiSGrfea81JXU6VZkJuM74yFONuQFxun0ad3mllxMw+6gB5Mbm</X509Certifica[redacted]></
X509Data></KeyInfo></Signature><saml:Assertion xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:SOAP="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
AssertionID="A0050568B-1364-71DF-9EBF-996D04C85A9E" MajorVersion="1" MinorVersion="1"
IssueInstant="2010-07-27T10:52:16.512Z" Issuer="https://www.CSC.com/SSO"><saml:Conditions
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" NotBefore="2010-07-27T10:47:16.512Z"
NotOnOrAfter="2010-07-27T18:52:16.512Z"/><saml:AuthenticationStatement xmlns:saml="urn:oasis:names:tc:SAML:
1.0:assertion" AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
AuthenticationInstant="2010-07-27T10:52:16.512Z"><saml:Subject xmlns:saml="urn:oasis:names:tc:SAML:
1.0:assertion"><saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">joe.doe@CSC.com</
saml:NameIdentifier></saml:Subject></saml:AuthenticationStatement></saml:Assertion></wsse:Security></
SOAP:Header><SOAP:Body><args xmlns="http://webservice.csc.com/SMVGetResultList"><item xmlns="http://
webservice.csc.com/SMVGetResultList">--context_param custID=CSC</item><item xmlns="http://webservice.csc.com/
SMVGetResultList">--context=Production</item></args></SOAP:Body></SOAP:Envelope>

3182 Bytes

# Eingabe a la SOAP

<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"><SOAP:Header xmlns:SOAP="http://
schemas.xmlsoap.org/soap/envelope/">\n\t\t\n\t<i18n:international xmlns:i18n="http://www.w3.org/2005/09/ws-i18n"><locale
xmlns="http://www.w3.org/2005/09/ws-i18n">en-gb</locale></i18n:international><wsse:Security xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"><Signature xmlns:SOAP="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns="http://www.w3.org/2000/09/
xmldsig#"><SignedInfo><CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><Reference
URI="#A0050568B-1364-71DF-9EBF-996D04C85A9E"><Transforms><Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></Transforms><DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>zkdy1Ok5td+wgJY6/
oY60Izq87M=</DigestValue></Reference></SignedInfo><SignatureValue>lbkuZYzj+T/
kUjwUG0CwhPONDqc9cSQGjqwAoGiMil5ZGtQJ6EiVtSLNZHf2aHkPR8WY0umh2d1mpz
j23Zi3Rzksu9LbWYiVhc7xpM6m69n6plEu8UMsM6riQCj8jzERJWAtPwaujN5jnlVE1GTF[illegible]hqb7RiqErPll=</
SignatureValue><KeyInfo><X509Data><X509Certificate>MIIB3DCCAUWgAwIBAgIPl[illegible]YK
+eFPzMA0GCSqGSIb3DQEBBQUAMDAxDzANBgNVBAoTBr[illegible]lbTEdMBsGA1UE[illegible]bjc2NzZXBhY2xvdW
QwHhcNMTAwNjEwMTUxMzQwWhcNMjAwNjA3MTUxMzQw[illegible]RcwFQYDVQQDEw5z[illegible]vbjEPMA0GA1UE
ChMGc3lzdGVtMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCB[illegible]QCwh8Rz7+dyHsh4VrQqH[illegible]X5eZpZHKCPi
I7YaRDn5A6m/
oLVUi92BBqSMYBCxknptHLFeHhaJixpC2cayGdyc9rdmKa5sl[illegible]w0tj44fTSAjzhcd5VLUQaiRh7v9YW[illegible]7QX+
+WFviD/RrjmOjTQIDAQABMA0GCSqGSIb3DQEBBQUAA4G[illegible]kAbopNX0/Hi9Z3NYTvpl00MwNk5N
+CK99pT7ia278xJY5qkoFfS644QRLmyv6Etbb2/MozXy6h0wT[illegible]PqSb52Yp
+YyMhZPsK9HcudjC0djxwARf7OWQiSGrfea81JXU6VZkJuM[illegible]luQFxun0ad3mllxMw+6gB5Mbm</X509Certifica[illegible]></
X509Data></KeyInfo></Signature><saml:Assertion xmlns:sam[illegible]:oasis:names:tc:SAML:1.0:protocol" xmlns:SOAP="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:saml="urn:oasis[illegible]tc:SAML:1.0:assertion"
AssertionID="A0050568B-1364-71DF-9EBF-996D04C85A9E"[illegible]ersion="1" MinorVersion="1"
IssueInstant="2010-07-27T10:52:16.512Z" Issuer="https://ww[illegible]com/SSO"><saml:Conditions
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" NotBefo[illegible]0-07-27T10:47:16.512Z"
NotOnOrAfter="2010-07-27T18:52:16.512Z"/><saml:Authentic[illegible]atement xmlns:saml="urn:oasis:names:tc:SAML:
1.0:assertion" AuthenticationMethod="urn:oasis:names:tc:SAM[illegible]m:password"
AuthenticationInstant="2010-07-27T10:52:16.512Z"><saml:Su[illegible]nlns:saml="urn:oasis:names:tc:SAML:
1.0:assertion"><saml:NameIdentifier Format="urn:oasis:na[illegible].1.1:nameid-format:unspecified">joe.doe@CSC.com</
saml:NameIdentifier></saml:Subject></saml:AuthenticationSta[illegible]></saml:Assertion></wsse:Security></
SOAP:Header><SOAP:Body><args xmlns="http://webservice.csc[illegible]com/SMVGetResultList"><item xmlns="http://
webservice.csc.com/SMVGetResultList">--context_param **custID=CSC<**/item><item xmlns="http://webservice.csc.com/
SMVGetResultList">--context=Production</item></args></SOAP:Body></SOAP:Envelope>

**3182 Bytes**

# Ausgabe a la SOAP

<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><soapenv:Body><runJobResponse xsi:type="ns[illegible]nJobResponse" xmlns="http://webservice.csc.com/SMVGetResultList" xm[illegible]"http://webservice.csc.com/SMVGetResultList"><ns1:item xsi:type="ns1:ArrayOf_xsd_string"><custID xsi:type="xsd:strin[illegible]custID><userID xsi:type="xsd:string">joe.doe@CSC.com</userID[illegible]xsi:type="xsd:string">10</requestID><fileName xsi:type="xsd:string">[illegible]mfoto 2010-07-07 um 19.47.46.png</fileName><comment xsi:type="xsd:string">[illegible]st file upload test of Friday</comment><fileStatus xsi:type="xsd:string">0</fileStatus><uploadDate xsi:type="xsd:string">12-07-2010</uploadDate></ns1:item></runJobResponse></soapenv:Body></soapenv:Envelope>

**891 Bytes**

# Machen wir die Rechnung nochmals auf:

| | | | |
|---|---|---|---|
| • Eingabe netto | 50 Bytes, | brutto | 3.182 Bytes |
| • Ausgabe netto | 179 Bytes, | brutto | 891 Bytes |
| | ======= | | ========== |
| • Gesamt: | 229 Bytes, | brutto | 4.673 Bytes |

- Overhead: **4.444 Bytes**

- **Bei 2.000 Transaktionen / Sekunde macht dies:**

  **8.888.000 Bytes, die pro Sekunde unnötig durch die Gegend geschoben werden!**

# Besser is‘ das...

- JSON
- CSV
- Gezippte Java Objekte

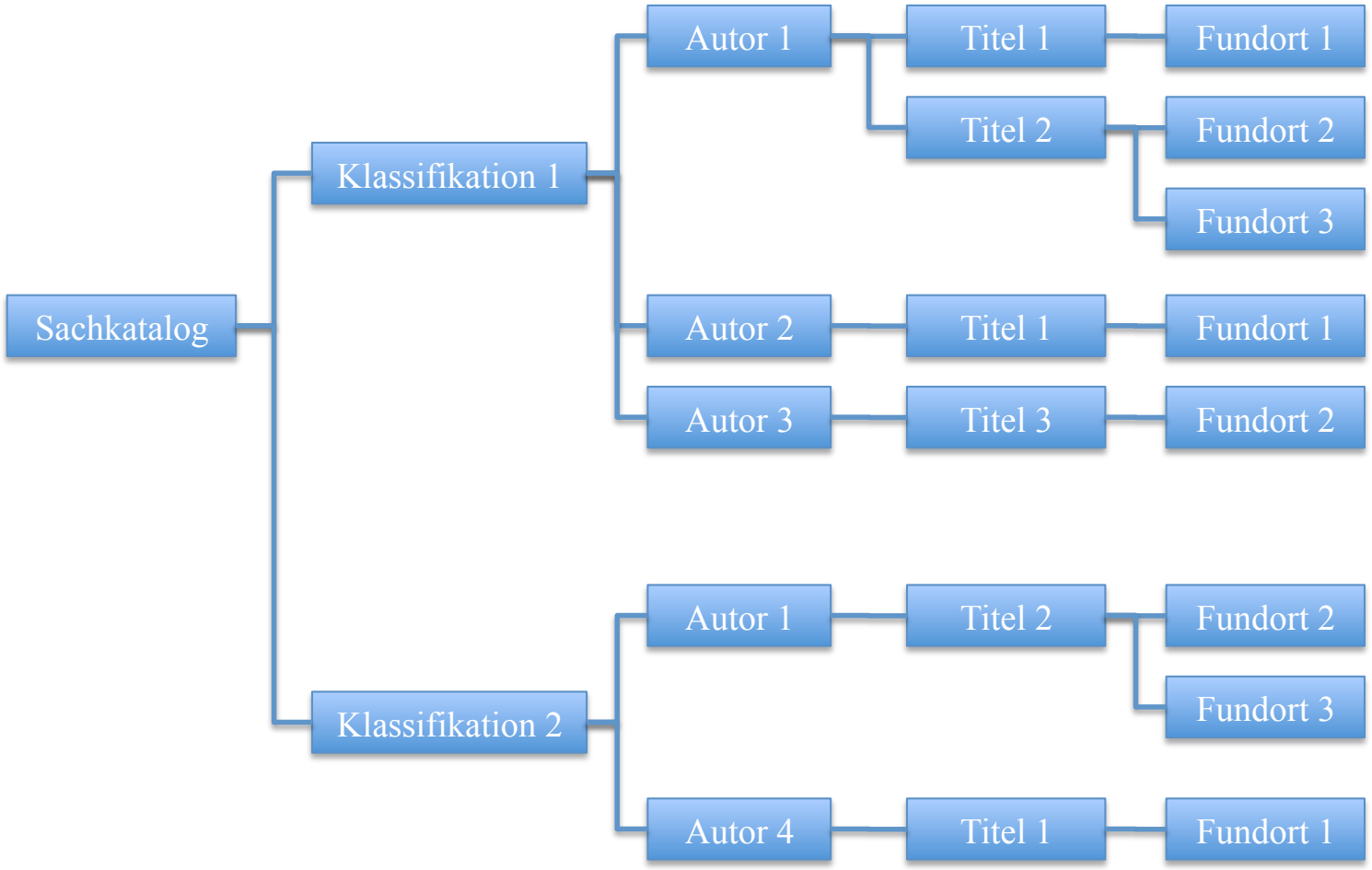
# Agenda

- Wer ist CSC (schamloser Werbeblock)
- Wer früher stirbt ist länger tot
- Von XML, dem Teufel und dem Weihwasser
- **Alles voll normal oder was?**
- Das Geheimnis hinter linearer Skalierbarkeit
- Schlangestehen ist cool!
- Hochstapler

# Datenbank != Datenbank

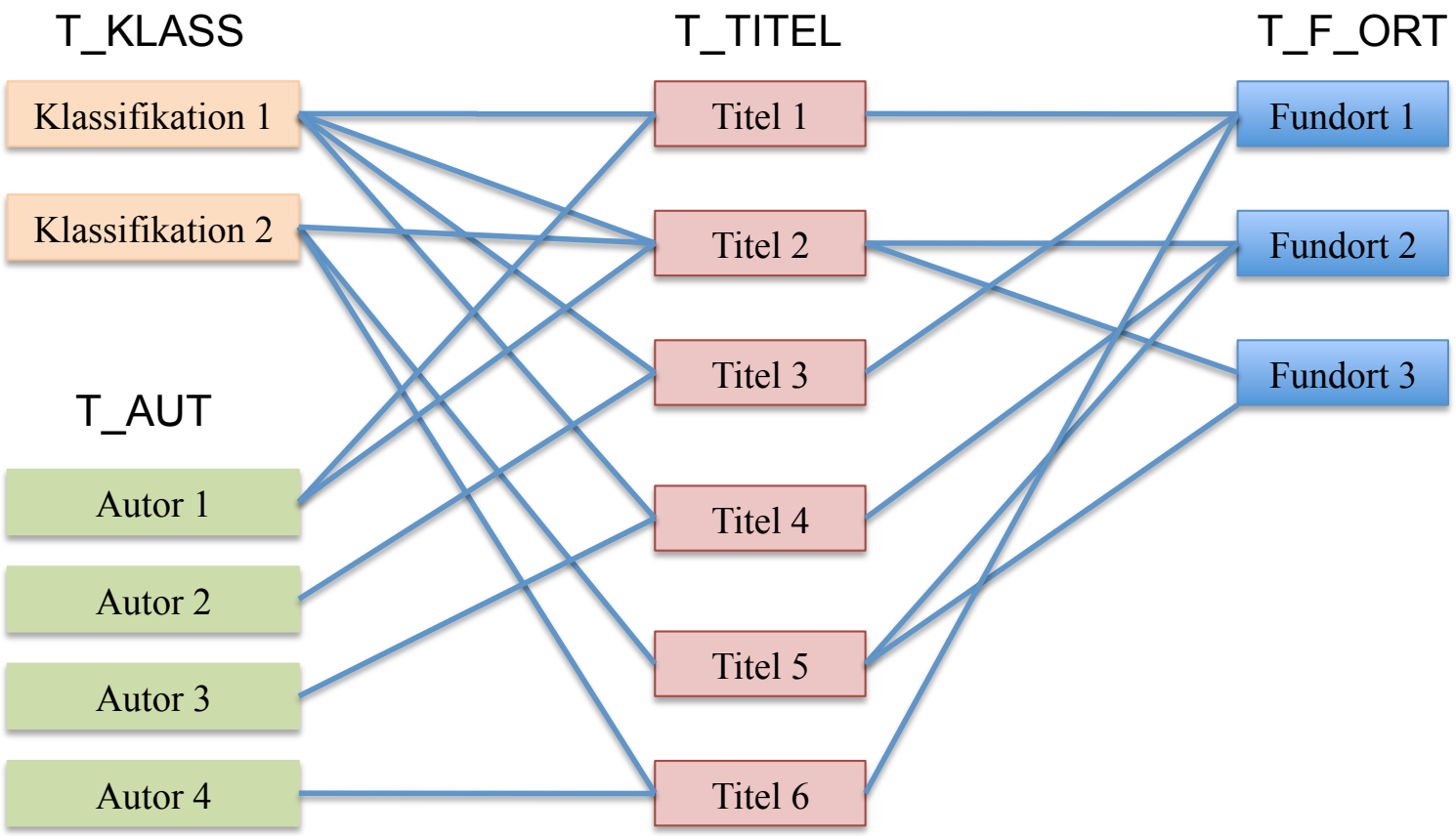
- Auf Mainframes sind in hoch transaktionalen Systemen meist hierarchische Datenbanken zu finden. Ein Vertreter ist IMS/DB (Information Management System)
- Neue IT-Systeme werden i.d.R mit relationalen Datenbanken gebaut.
- Es gibt jedoch noch weitere, die Performance beeinflussende Unterschiede...

# **hierarchisch** vs. relational




Quelle: http://www.payer.de/dbaufbau/dbauf01.html

# hierarchisch vs. **relational**




Quelle: http://www.payer.de/dbaufbau/dbauf01.html

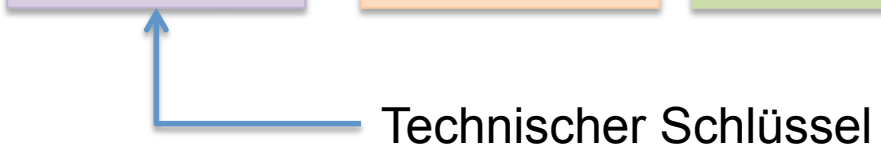
# Besser is‘ das...

- Versuche nie ein ehemals hierarchisches Datenmodell relational abzubilden!
- Die notwendigen Joins bringen die Performance in den Keller!
- Vorsicht bei automatischen OR-Mappern – die Tabellen die erzeugt werden bringen jeden DBA zur Verzweiflung!
- Verwende ein denormalisiertes Datenmodell. Das ist zwar nicht schön, aber schnell!
- **Es bedingt aber auch zwingend die Einführung von technischen Primary Keys!**

# Denormalisiertes Modell

T_SACHKATALOG

| | | | | |
|---|---|---|---|---|
| Primary Key 1 | Klassifikation 1 | Autor 1 | Titel 1 | Fundort 1 |
| Primary Key 2 | Klassifikation 1 | Autor 2 | Titel 2 | Fundort 2 |
| Primary Key 3 | Klassifikation 1 | Autor 2 | Titel 2 | Fundort 3 |
| Primary Key 4 | Klassifikation 1 | Autor 3 | Titel 4 | Fundort 2 |
| Primary Key 5 | Klassifikation 2 | Autor 1 | Titel 2 | Fundort 1 |
| Primary Key 6 | Klassifikation 2 | Autor 1 | Titel 2 | Fundort 2 |
| Primary Key 7 | Klassifikation 2 | Autor 4 | Titel 1 | Fundort 1 |



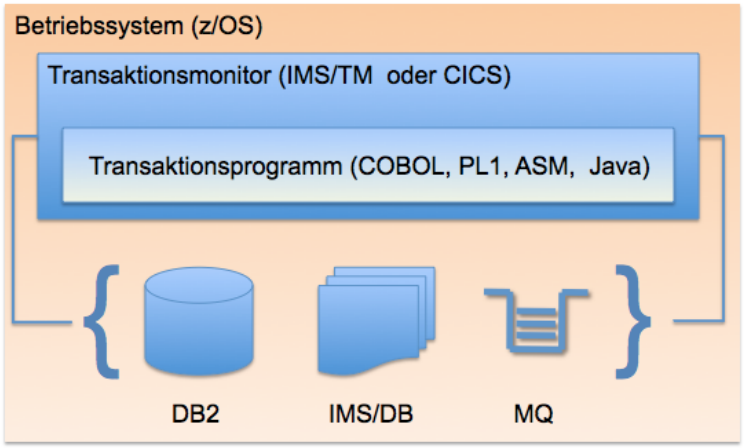
# Besser is‘ das...

- Vielleicht muss es auch gar nicht relational sein!

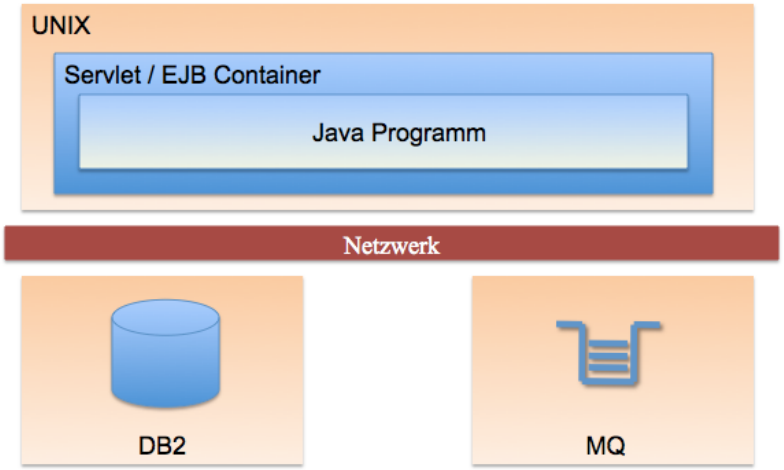
- Es gibt Alternativen:
  - Apache Hadoop
  - Google Bigtable
  - Twitter Gizzard
  - BerkeleyDB (Oracle)

Quelle: http://www.pro-linux.de/artikel/2/1455/nosql-jenseits-der-relationalen-datenbanken.html

# Zugriffszeit




- Zugriffe erfolgen im Prinzip mit der Geschwindigkeit des Speichers.
- Es gibt **kein** Netzwerk zwischen Business Logik und Datenbanken.
- Latenz-Zeit im **µs** Bereich




- Zugriffe erfolgen über das Netzwerk
- Latenz-Zeit im **ms** Bereich

# Noch ´ne Rechnung

- Mal angenommen, eine Transaktion macht 4 relationale Datenbankzugriffe (open, select, update, close)

- Weiter angenommen auf dem Mainframe dauert ein Zugriff 50 µs. Dann dauert die Transaktion **200 µs = 0.2 ms = 0.0002 s.**
  Es können also bis zu 5.000 Transaktionen pro Sekunde ausgeführt werden.

- Im Verteilten System dauert ein Zugriff 4 ms, somit dauert die Transaktion **16 ms**. -> 62 TX/Sek.

# Agenda

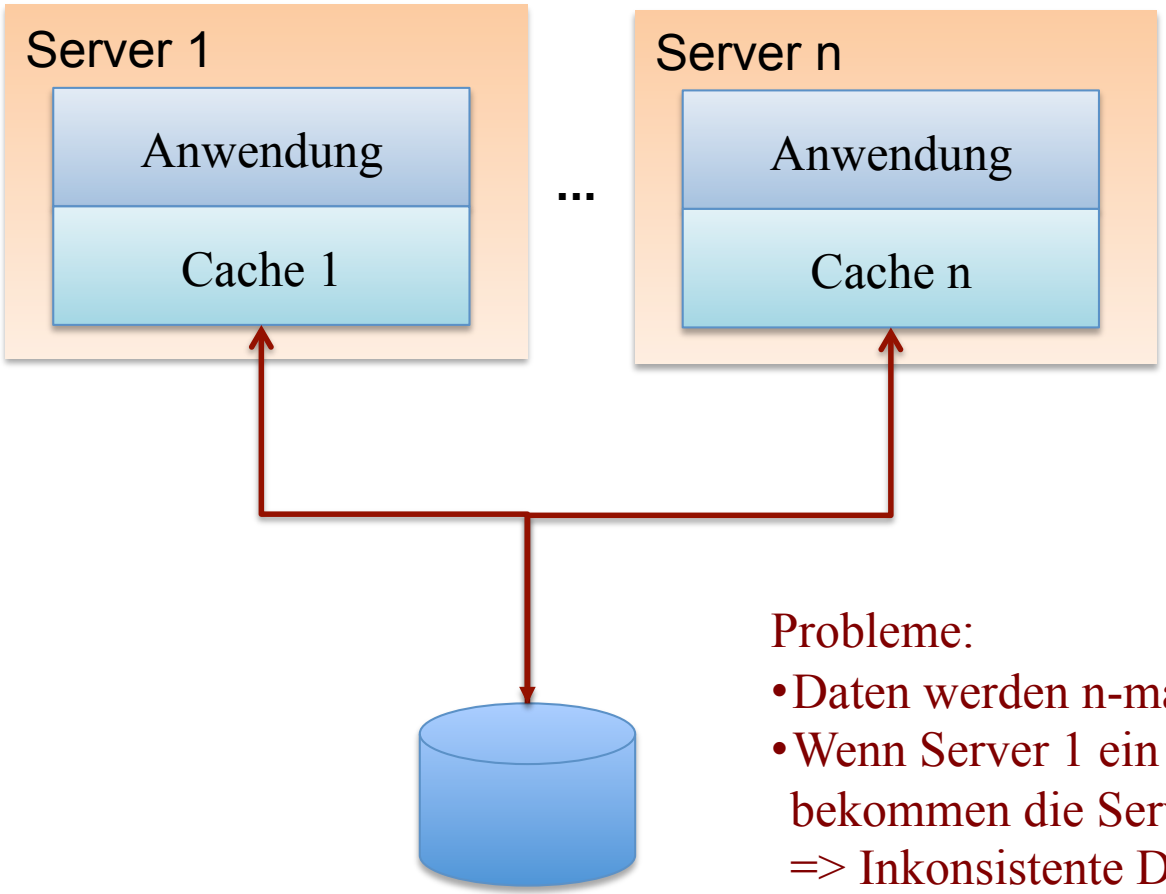
- Wer ist CSC (schamloser Werbeblock)
- Wer früher stirbt ist länger tot
- Von XML, dem Teufel und dem Weihwasser
- Alles voll normal oder was?
- **Caching, aber richtig!**
- Das Geheimnis hinter linearer Skalierbarkeit
- Hochstapler

# Die Lösung – Caching!

# Caching, aber richtig!

- Es gibt Daten die prädestiniert dafür sind gecached zu werden:
- Lesende Zugriffe auf statische Daten die sich gar nicht oder nur selten ändern.
  - Adressdaten
  - Steuerungsdaten (z.B. User-Preferences)
- Es gibt Daten die man nicht cachen sollte:
  - Daten die von verschiedenen Systemen gleichzeitig verändert werden können.
  - Sich häufig ändernde Daten (z.B. Kontostände)

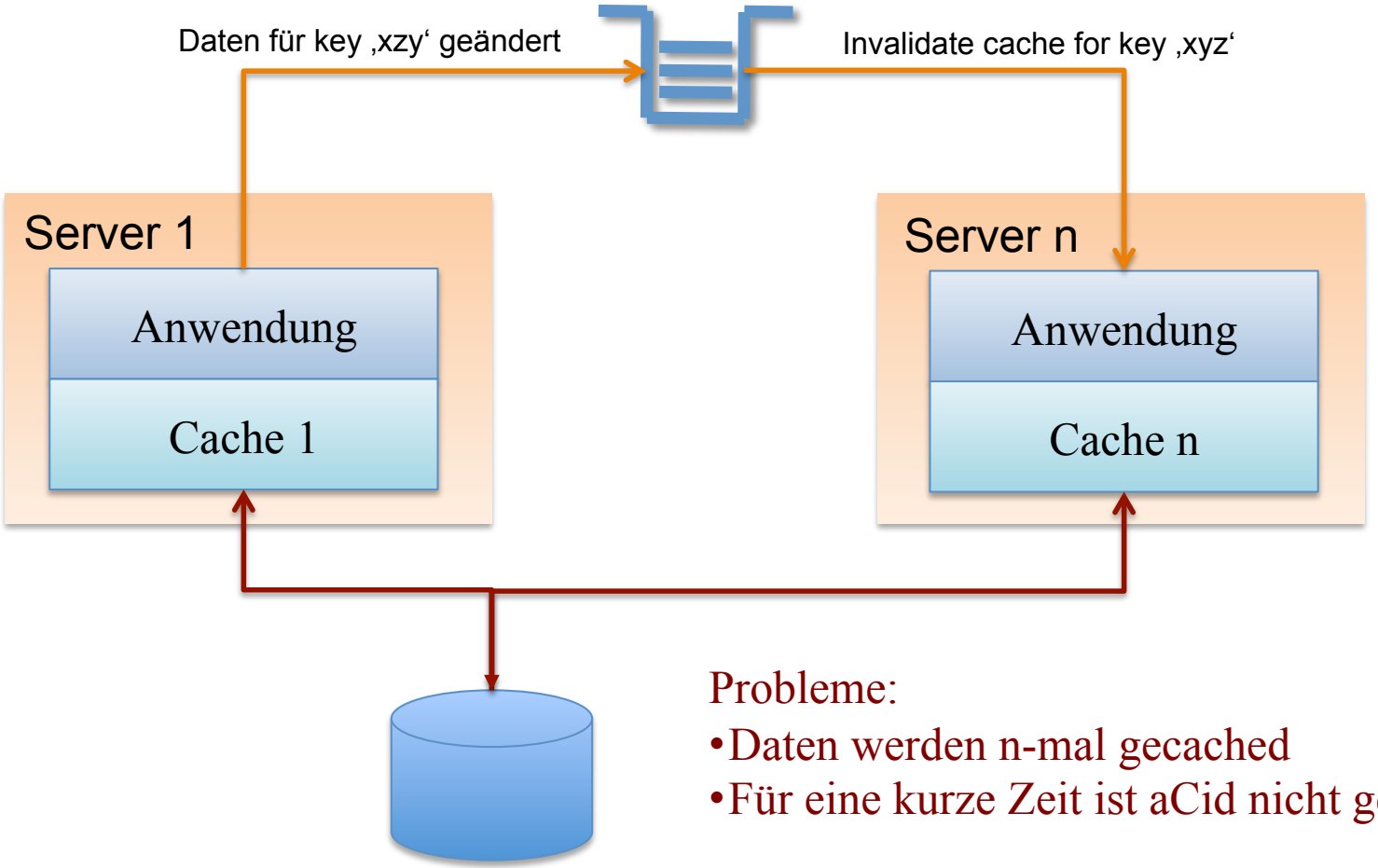
# Das caching Dilemma




Probleme:

- Daten werden n-mal gecached
- Wenn Server 1 ein Update macht, bekommen die Server 2 – n dies nicht mit => Inkonsistente Daten

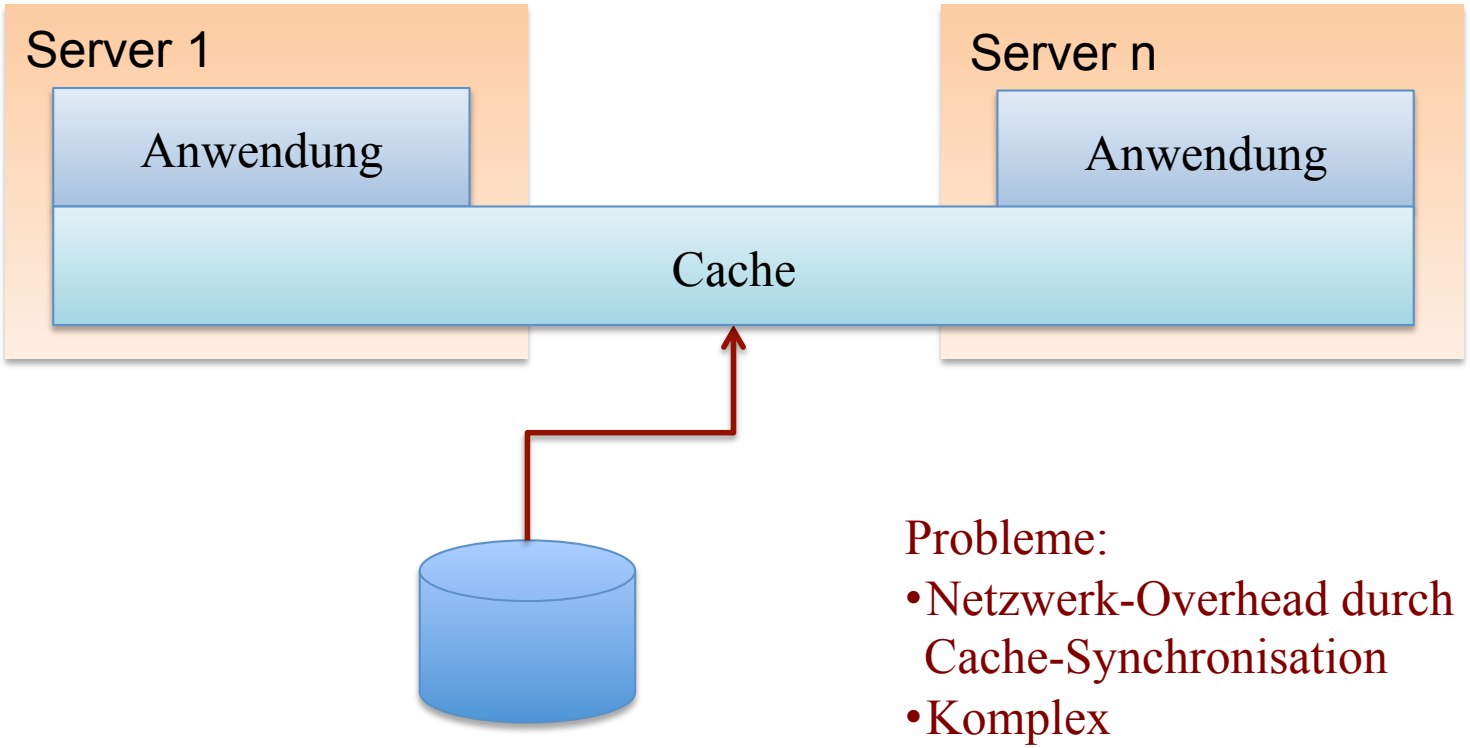
# Lösungsmöglichkeit 1 - Messaging




Probleme:

- Daten werden n-mal gecached
- Für eine kurze Zeit ist aCid nicht gewährleistet

# Lösungsmöglichkeit 2 – verteilter Cache




Probleme:
- Netzwerk-Overhead durch Cache-Synchronisation
- Komplex

# Besser is‘ das...

- Prämissen für verteiltes Caching:
  - Alle Updates müssen ebenfalls über den Cache erzeugt werden.
  - Das Netzwerk muss für die zusätzliche Last ausgelegt sein die das Synchronisieren der Caches erzeugt.
  - Es muss sichergestellt sein, dass die Updates immer in der Datenbank landen.

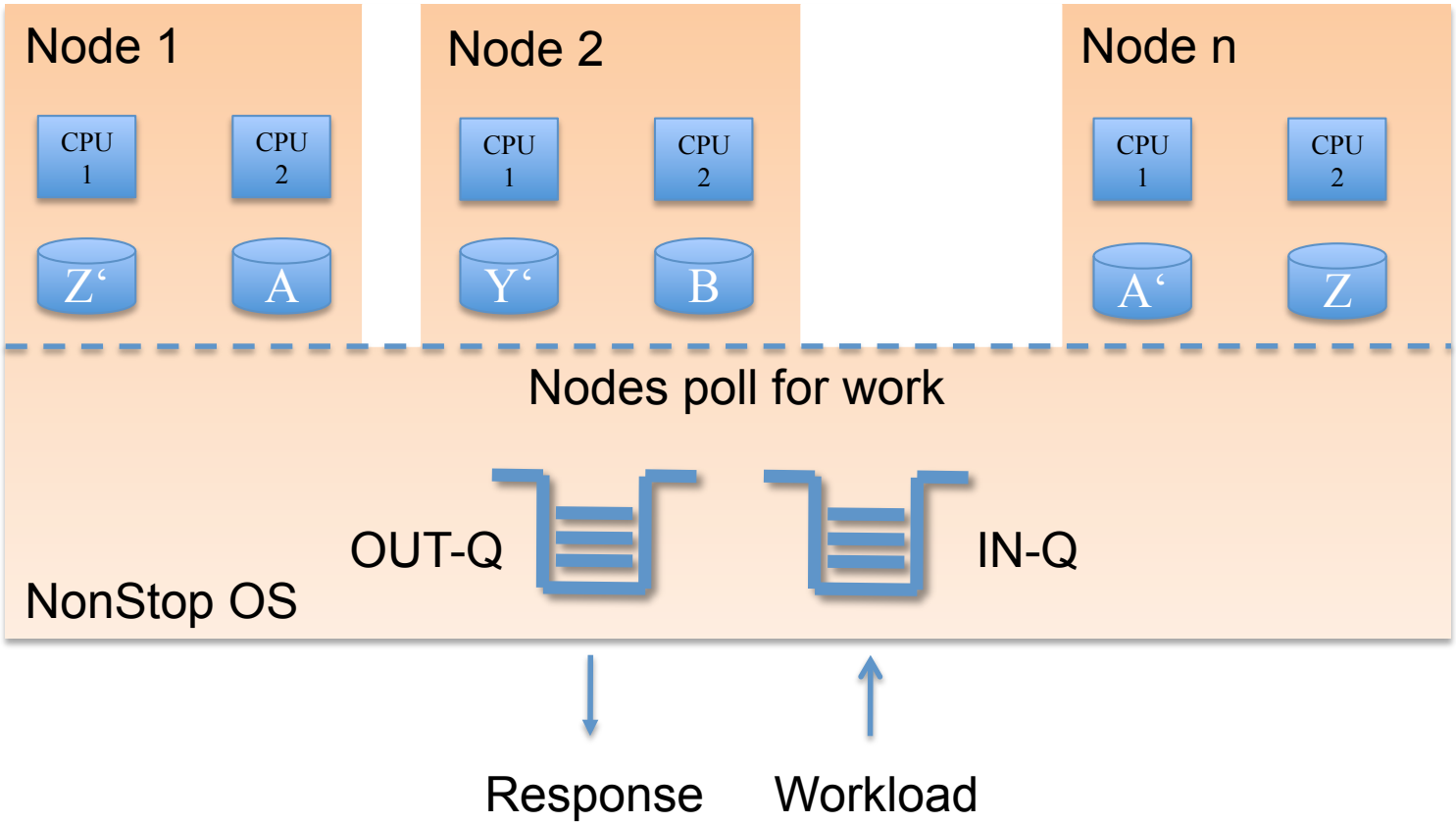
- Produkte
  - Oracle Coherence
  - IBM Websphere Extreme Scale
  - GigaSpaces XAP In Memory Data Grid
  - Terracotta Enterprise EH-Cache

# Agenda

- Wer ist CSC (schamloser Werbeblock)
- Wer früher stirbt ist länger tot
- Von XML, dem Teufel und dem Weihwasser
- Alles voll normal oder was?
- Caching, aber richtig!
- **Das Geheimnis hinter linearer Skalierbarkeit**
- Hochstapler

# Skalierung MPP* Style – shared nothing

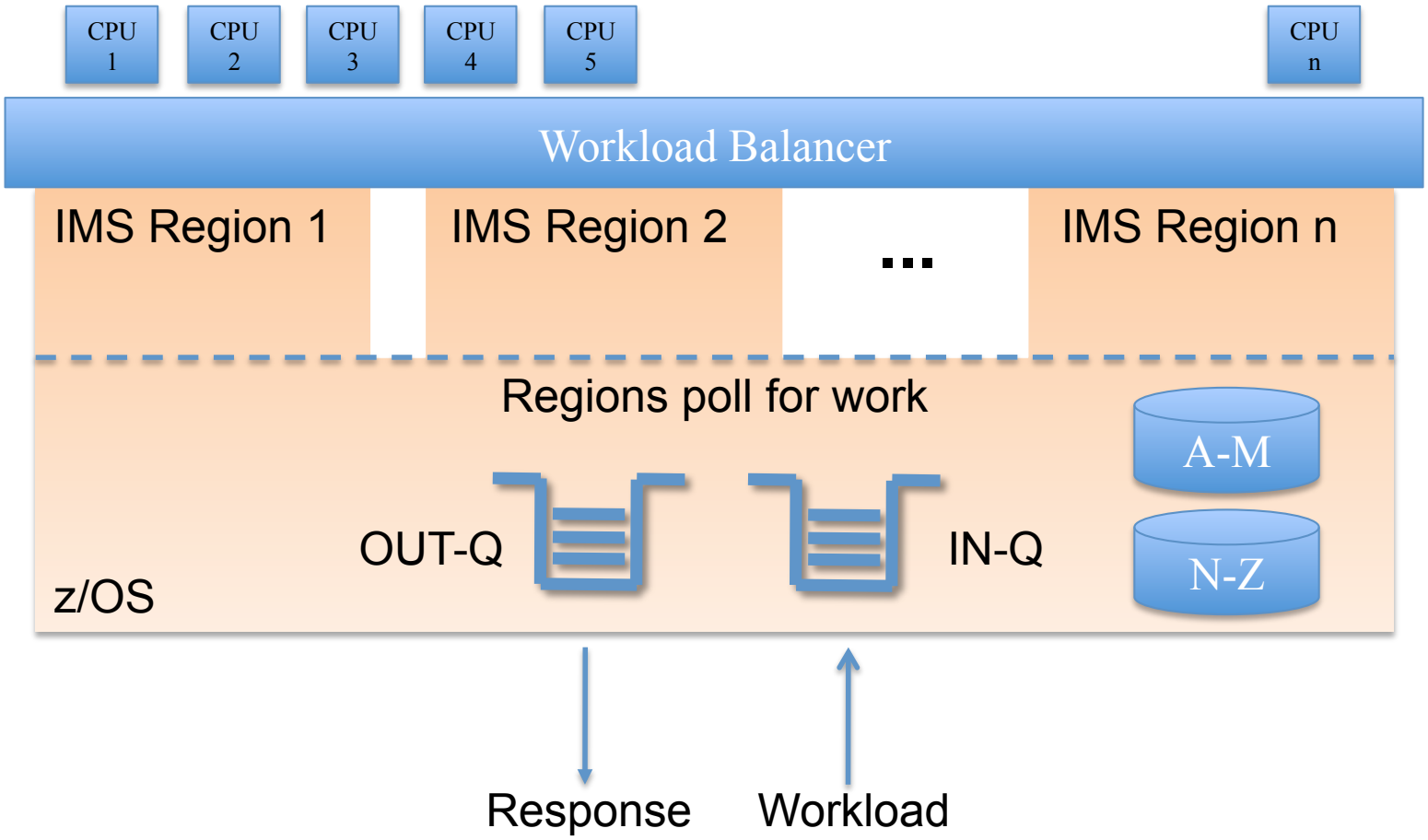
Beispiel: HP NonStop Server mit 99.9999% Verfügbarkeit




*MPP = Massive Parallel Processing

# Skalierung IMS Style

Beispiel: IBM Mainframe IMS/TM



# Viele Gemeinsamkeiten

- Nutzung von Queues für Input und Output
- Single Threaded Verarbeitung
- Nodes bzw. Regions pollen die Queues
- Requests gehen nicht verloren
- **Vorteile**
  - Der Node / die Region kann nie überlastet werden
  - Zusätzliche Nodes / Regions bringen fast 100% zusätzlichen Durchsatz.

# Tipps für Queues

- Queuemanager und Datenbank müssen über einen Transaktionsmanager ansteuerbar sein.
- Auch das Queueing-System muss redundant sein.
- Nur Pointer auf die Daten in der Queue halten!
  - Schnellere Verarbeitung in der Queue
  - Die Daten liegen da wo sie hingehören, in der DB

# Agenda

- Wer ist CSC (schamloser Werbeblock)
- Wer früher stirbt ist länger tot
- Von XML, dem Teufel und dem Weihwasser
- Alles voll normal oder was?
- Caching, aber richtig!
- Das Geheimnis hinter linearer Skalierbarkeit
- **Hochstapler**

# Batchverarbeitung? Gibt‘s das noch???

- DTA-Verarbeitung in Banken
  - Täglich mehrere Millionen Records, fixed length
  - Mehrere tausend Dateien
- SEPA-Verarbeitung in Banken
  - Täglich mehrere Millionen Records, **XML**
  - Täglich mehrere tausend Dateien
- Ticketing Systeme von Airlines
- Versicherungen

DTA = **D**aten**T**räger**A**ustausch Format

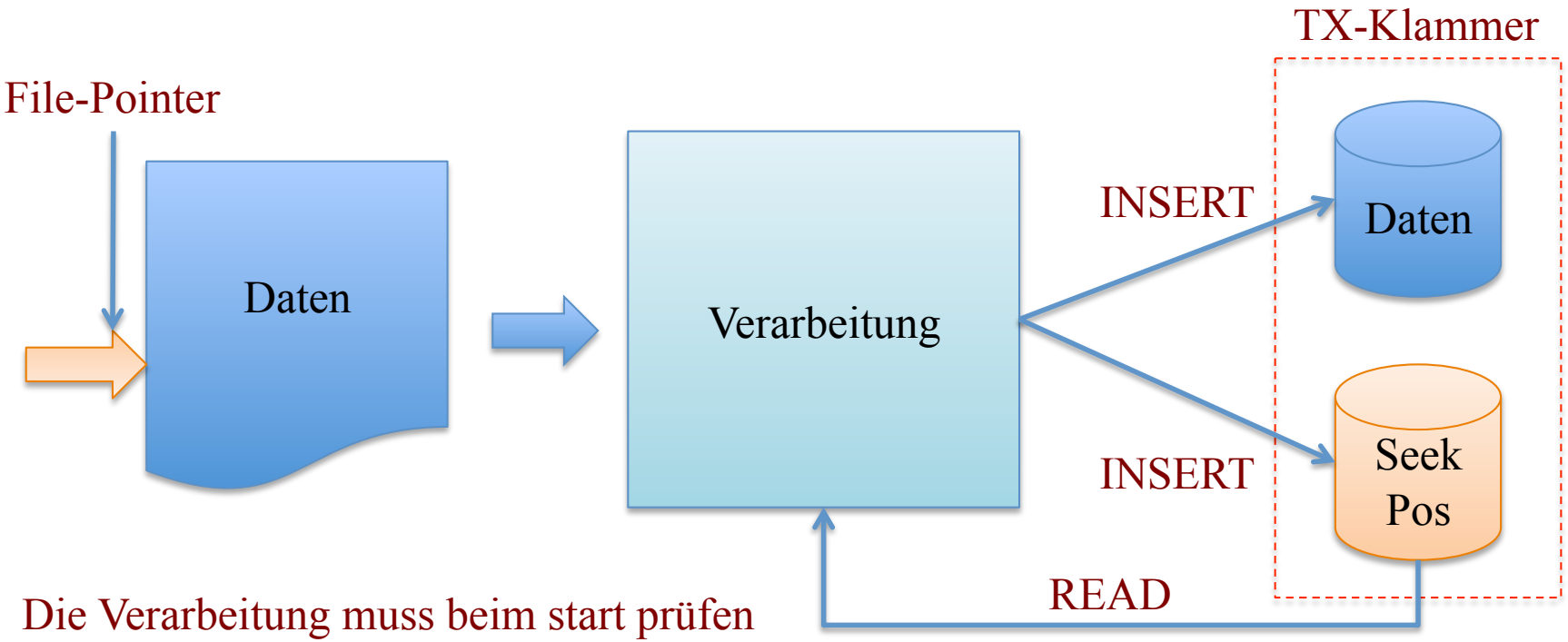
SEPA = **S**ingle **E**uro **P**ayments **A**rea

# Stapelverarbeitung - Herausforderungen

- Die Verarbeitung muss nach Unterbrechung an der Stelle fortgesetzt werden bei der die Unterbrechung stattfand. (Wie war das im Mittelteil?...)

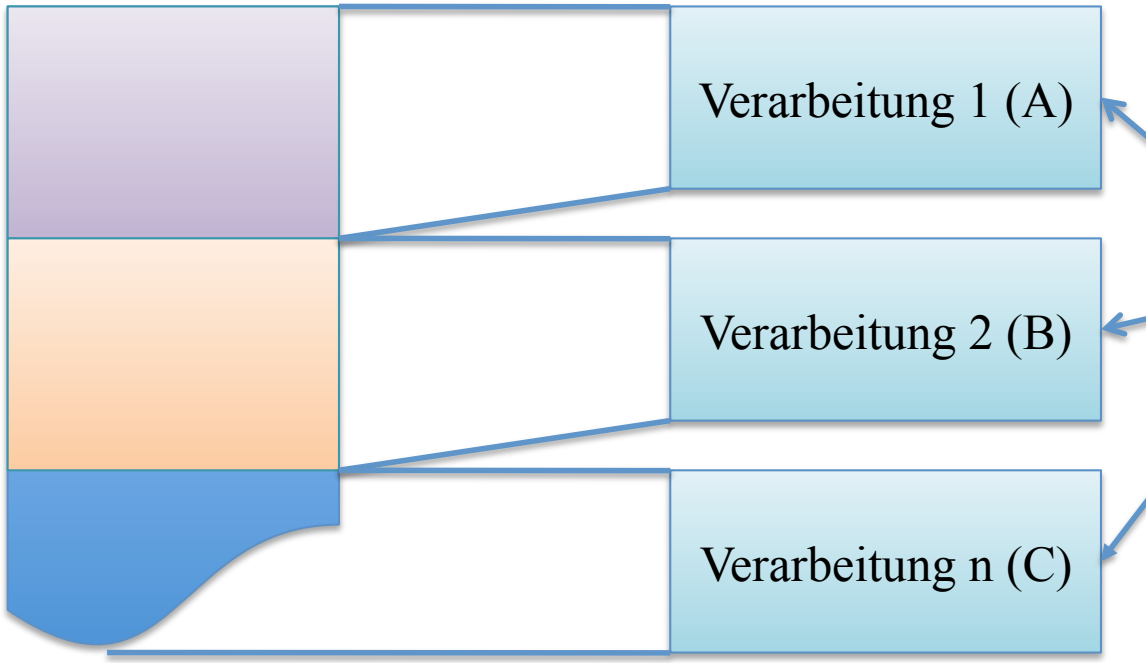
- Möglichst viele Dateien in möglichst kurzer Zeit verarbeiten (Batchfenster)

# Verfahren zum Wiederaufsetzen




Die Verarbeitung muss beim start prüfen ob es eine Unterbrechung gab. Wenn ja ist die zuletzt verarbeitete Datei und die Seek Position aus der DB zu lesen und die Verarbeitung beim nächsten Record fortzusetzen.
Zur Performanceverbesserung ist ein, für den Anwendungsfall noch vertretbares Commit-Interval zu wählen, z.B. alle 100 oder 1000 Records.

# Parallelisierung bei fixed length files




| Key | Start | End |
|---|---|---|
| A | 0 | 999 |
| B | 1000 | 1999 |
| C | 2000 | 2500 |

Vor der Verarbeitung wird die Datei analysiert und die Einträge in der Steuertabelle vorgenommen. Dann werden die Aufträge auf die Queue gelegt.

# Was noch übrig ist...

Datenbanksysteme

Fehlertoleranz

Continuous Operation

Rapid Failover

Grid

Netzwerke

Versionierung

Koexistenz

# Fragen?

- eMail: mschorer@csc.com
- Phone: +49 172 302 1954
- Skype: matze.schorer
- Twitter: mschorer
- XING: mschorer

12.–15.09.2010
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

# Vielen Dank!

Matthias Schorer
CSC