

## Verteilte Systeme – Fragen und Antworten v0.5

Nennen Sie 2 Vorteile von Verteilten Systemen und zentralisierten Systemen. S3

<b>PRO Distributed</b> <ul style="list-style-type: none"><li>• Einfacher zu modellieren/programmieren</li><li>• Scales better</li><li>• No "single point of failure" (Fault tolerant systems)</li></ul>	<b>PRO Centralized</b> <ul style="list-style-type: none"><li>• Einfaches Management One point für Konfiguration, Wartung, etc.</li><li>• Billige Ressourcen</li><li>• Konsistente Zustände No communication inside</li></ul>
<b>CON Distributed</b> <ul style="list-style-type: none"><li>• Complex side effects State machine with permanent "race conditions"?</li><li>• Komplexes Management (Netzwerk, Konfiguration, etc.)</li><li>• Teure Ressourcen N times power supply, memory, casing, etc.</li></ul>	<b>CON Centralized</b> <ul style="list-style-type: none"><li>• Single point of failure</li><li>• Scales badly</li><li>• komplexe Algorithmen/Modelle</li></ul>

Nennen Sie 4 Eigenschaften/Charakteristika von „Parallel Computing Systems“. S9

Gehört zu Distributed Computing Systems (DCS):

„Tightly coupled“ system ->

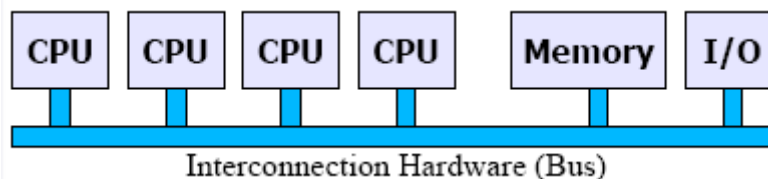
- **Common Clock**
  - ✓ Gemeinsame „Zeit“
  - ✓ keine Synchronisationsprobleme
- **hohe Bandbreiten-Kommunikation**
  - ✓ zwischen Systemteilen
  - ✓ zwischen Systemteilen und Speicher
- **physikalisch „tight“**
  - ✓ **one casing**, Motherboard, Backplane ....
- **logisch „tight“**
  - ✓ parallele Algorithmen, gemeinsamer source code

Performance ->

- Calculation performance
- Durchsatz
- Load balancing
- Number of clients-Distr. Denial of service attacks

CPUs share ->

- Memory resources
- Bus
- Clocking



Was ist ein Flow bei Quality of Service (QoS)? S22

Flow: Sequenz von verwandten Paketen von der Quelle zum Ziel.

-> Internet Protocol (IPv6) mit 24 bits flow label.

Version	Prio./ Class 4 bits	Flow Label 24 bits	Rest...
---------	---------------------------	-----------------------	---------

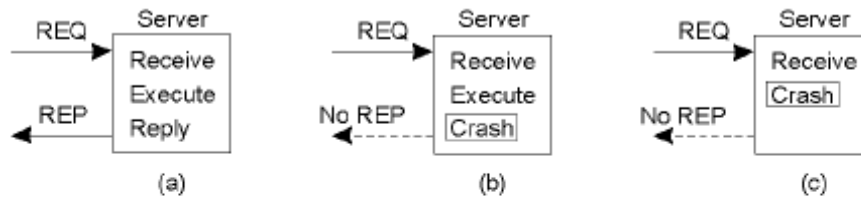
## Welches Betriebssystem befindet sich am LonWorks NEURON Chip? S168,169

„Scheduler“ oder „NEURON Chip Firmware“

- Neuron Scheduler -> „Round Robin“

## Wie kann man den Fehler "The request message from the client to the server is lost" am besten erkennen und beheben? S206

Fehler erkennbar aufgrund des Failure Models. -> in diesem Fall: Omission failure  
Lost request messages Server Crashes



A server in client-server communication

- b) Normal case
- c) Crash after execution
- d) Crash before execution

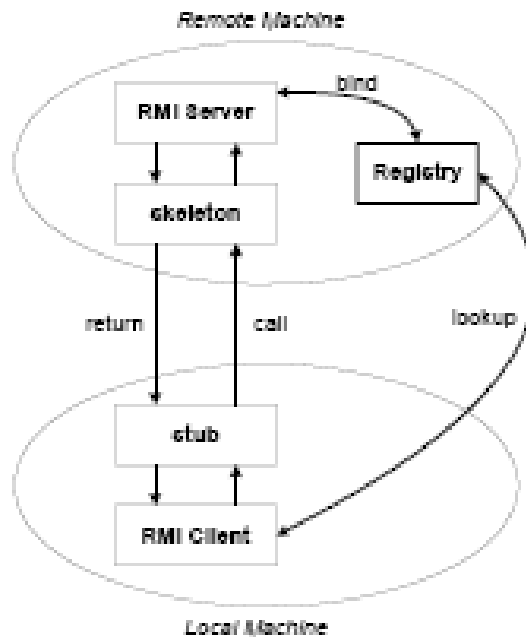
## Was ist die Besonderheit am menschlichen Körper gegenüber einem technischen System im Bezug auf die Verfügbarkeit und Zuverlässigkeit seiner Komponenten? S217

- **Availability (Verfügbarkeit):**  
The organic system have ANY TIME in state that is capable to guarantee life functions.
- **Reliability (Zuverlässigkeit):**  
Essential functions have ALWAYS be guaranteed.

*The human body has to be more reliable and available than its components.*

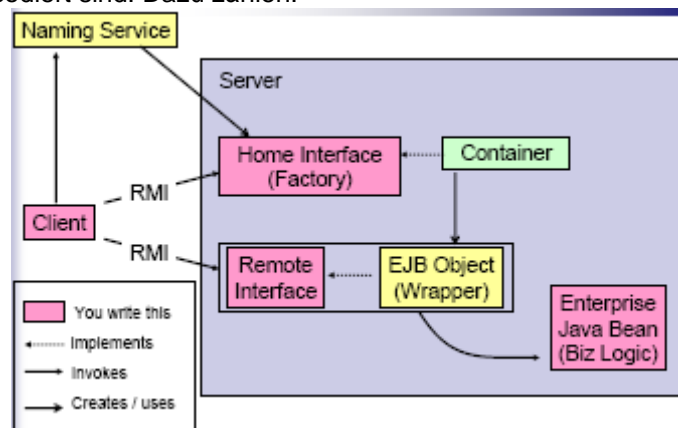
## Erklären Sie die RMI Architektur. S58

- Server muss zuerst seinen Name mit der Registry verbinden (bzw. einbinden)
- Client schlägt in der Registry den Servernamen nach um Remote Referenzen aufzubauen
- Ein Client ruft eine Remote method auf, der Aufruf wird zuerst zum Stub weitergeleitet.
- Der Stub ist für die Übertragung des Remote Calls zum Server-side Skeleton verantwortlich.
- Der Stub öffnet einen Socket zum Remote Server, führt (marshalling – wandelt Datenelemente in ein Format um, um sie in einer Nachricht an den Empfänger (Skeleton) zu schicken) die Objekt Parameter zum Skeleton und leitet den Datenstrom (data stream) an diesen weiter.
- Ein Skeleton beinhaltet eine Methode, die die Remote Calls empfängt, unmarshals – wandelt die Nachricht wieder so um, das die gleiche Datendstruktur wieder hergestellt werden kann - die Parameter und ruft die eigentliche Remote Object Implementation auf. Die Returnwerte werden wieder an den Stub zurückgeschickt.



## Erklären Sie die EJB Architektur. S68

- An **Enterprise Java Bean (EJB)** is a component that provides reusable business logic functionality and/or a representation of a persistent business entity
  - **Enterprise Java Beans (EJB)** sind standardisierte Komponenten innerhalb eines J2EE-Servers (Java 2 Enterprise Edition). Sie vereinfachen die Entwicklung komplexer mehrschichtiger verteilter Softwaresysteme mittels Java.
- An **EJB Container** executes an EJB due to a client request.
  - Provides the plumbing necessary to execute the EJB including
  - Client uses an interface to access the Bean indirectly
  - **EJB-Container** definieren zusätzliche containerspezifische Eigenschaften. Zum Zeitpunkt der Installation können diese Eigenschaften je nach Container auf unterschiedliche Art angegeben werden – in Java-Properties-Dateien, XML-Dateien oder interaktiv.
- A **deployment descriptor (Konfiguration)** describes the structure of the Bean and how to execute the Bean as part of an application
  - Dieser **Deployment Descriptor** ist eine XML-Datei, in der Eigenschaften von EJBs definiert werden, die nicht hart codiert sind. Dazu zählen:

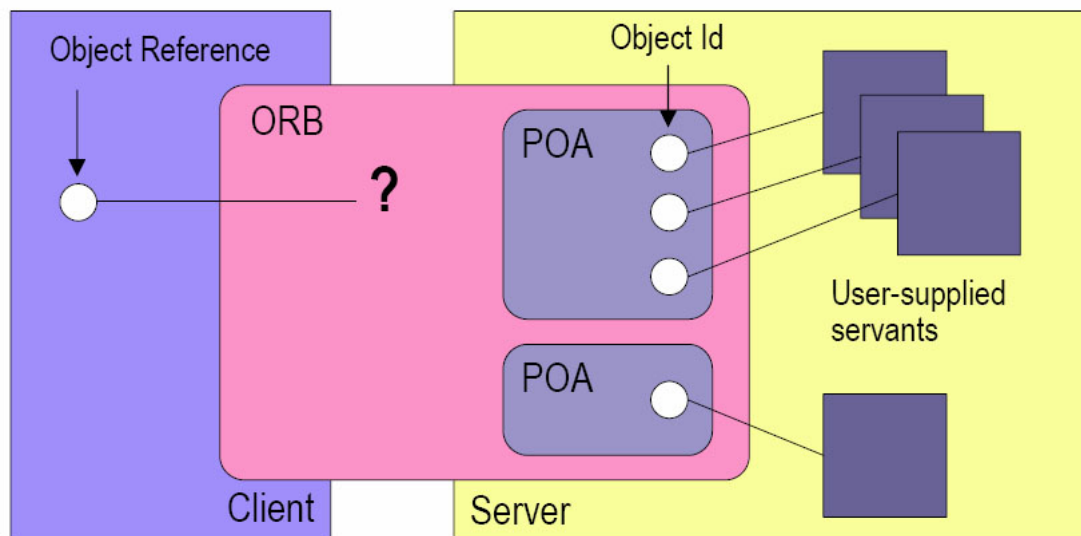


## Welche Aufgaben hat der Portable Object Adapter in CORBA? S83

- Wurde in CORBA 2.2 vorgestellt
- Ersetzt unter-specified BOAs (Basic Object Adapter)
- **Allow construction of object implementations that are portable between different ORB products**
- **Provide support for objects with persistent identities**
- **Provide support for transparent activation of objects**
- **Provide support for transient objects**
- Avoid requiring the ORB to maintain persistent state describing object properties
- Provide an extensible mechanism for associating policy information with objects implemented in the POA

- Allow a single servant to support multiple object identities simultaneously
- **Allow multiple POAs to exist in a server**

#### Abstract POA Model:



#### Nenne 4 Arten von Transparenz bei Middleware. S50

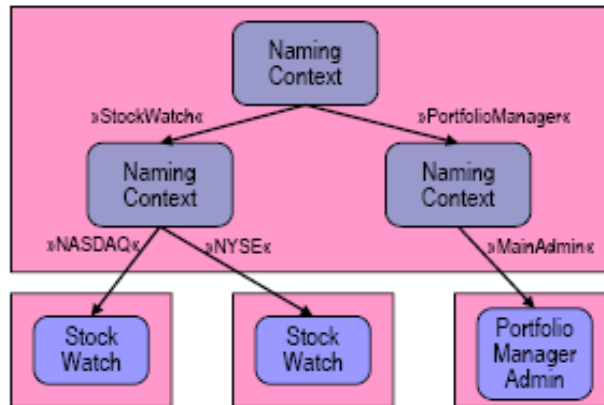
- **Access transparency** – verbirgt Unterschiede in der Datendarstellung und wie der Zugriff auf eine Ressource erfolgt
- **Location transparency** – verbirgt, wo sich eine Ressource befindet
- **Structure transparency** – mehrere File Server erkennt man als einen File Server
- **Replication transparency** – verbirgt die Existenz und den Ort von Kopien
- **Naming transparency** – Ort sollte nicht wichtig sein

#### Welche Schwierigkeiten gibt es bei verteilter Garbage Collection? S55

- Approaches:
  - Managed by the underlying platform (e.g., Distributed OS)
  - The broker identifies the number of live remote references (active connections count)
  - The server counts the number of references and the clients need to cooperate
- **Distributed reference counting: es ist schwierig die zyklischen Referenzen zu identifizieren**

#### Welche Aufgabe hat das CORBA Naming Service? S84

- Der wichtigste CORBA-Dienst ist der **Naming Service**, der Serverobjekten ermöglicht, mittels eines festgelegten Namens angesprochen zu werden. Der Namensdienst liefert dann die IOR zu einem registrierten Objektnamen. Der Naming Service ist eine Art "Telefonbuch" für Corba Objekte.
- Simple example of an object directory
- Stores a name with each object reference
- Name consists of two strings: id and kind
- Provides hierarchical naming space structure
- Hierarchy is made up of naming contexts
- Naming context can store multiple object references or other naming contexts



## Welche Aufgabe hat das CORBA Trading Service? S58

- Der **Trading Service** ermöglicht es ebenfalls, Objekte zur Laufzeit zu finden. Allerdings werden Objekte hier über ihre Eigenschaften identifiziert und nicht durch einen Namen. Das Ergebnis einer solchen Suche können auch mehrere Objekte sein.
- Each object (offer) can have multiple properties of any type
- Provides flexible mechanism for clients to look up objects based on any subset of these properties
- Is not structured in any formal way
- Based on the concept of **service type**
  - contains IDL interface identifier
  - data defining the attributes that can be associated

### Aufgabe:

Allows us to locate objects based on information that is important to us, rather than information that is important to the ORB

- Using Object Reference Strings
- Using Factory Objects
- Bootstrapping

## Was ist CCM? S89

Das **CORBA Component Model** ist ein auf [CORBA](#) aufsetzendes [Komponentenmodell](#). Die Spezifikation des CCM wurde von der [Object Management Group](#) (OMG) veröffentlicht.

Kernpunkt des CCM ist eine Komponente. Dafür führt das CORBA-Komponentenmodell den neuen Metatyp *CORBAComponent* in [CORBA](#) ein. Eine CORBA-Komponente kapselt ihren inneren Aufbau durch *Interfaces*. Diese Interfaces werden über *Ports* angeboten. Momentan wird in folgende Portarten unterschieden.

- A CORBA Component Model (CCM) application is “really” distributed
  - Could be deployed and run on several distributed nodes simultaneously
- A CORBA component could be segmented into several classes

### Spezifikation:

- **Abstract Component Model**
  - Extensions to IDL and the object model
- **Component Implementation Framework**
  - Component Implementation Definition Language (CIDL)
- **Component Container Programming Model**
  - Component implementer and client view
  - Integration with Security, Persistence, Transactions, and Events
- **Packaging and deployment facilities**
- **Interoperability with EJB 1.1**
- **Component Metadata & Metamodel**

## Welche zwei Formen of Remoting unterstützt .NET? S101

- **Web Services**
  - An entry point into application specified by an URL
  - Expose WebService endpoints from any process over any transport using any payload encoding
  - Process types include console apps, graphical applications, NT Services, IIS
- **CLR Object Remoting**
  - Builds on Web Services

- Uses native CLR data types
- Full CLR type system fidelity
- Maintains distributed object identity
- Provides object activation semantics
- Allows control over object lifetime using leases
- Permits out of band info using CallContext

### Was ist ein Skeleton / Server-side Proxy? S58

### Was ist ein Stub? (10/19) S64

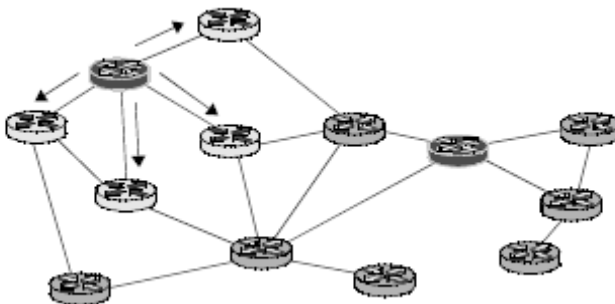
### Was ist ein Stub/Skeleton proxy? S58

- Ein Client ruft eine Remote method auf, der Aufruf wird zuerst zum Stub weitergeleitet.
- Der Stub ist für die Übertragung des Remote Calls zum Server-side Skeleton verantwortlich.
- Der Stub öffnet einen Socket zum Remote Server, führt (marshalling – wandelt Datenelemente in ein Format um, um sie in einer Nachricht an den Empfänger (Skeleton) zu schicken) die Objekt Parameter zum Skeleton und leitet den Datenstrom (data stream) an diesen weiter.
- Ein Skeleton beinhaltet eine Methode, die die Remote Calls empfängt, unmarshals – wandelt die Nachricht wieder so um, das die gleiche Datendtruktur wieder hergestellt werden kann - die Parameter und ruft die eigentliche Remote Object Implementation auf. Die Returnwerte werden wieder an den Stub zurückgeschickt.

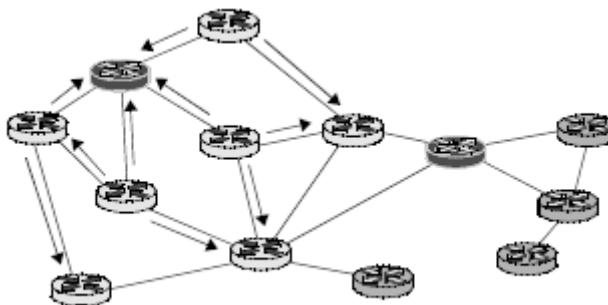


### Beschreiben Sie den Flooding Algorithmus zum Austausch von Informationen. (S124)?

- Broadcast der Nachrichten an alle Nachbarn
- einmaliges Weiterleiten aller empfangenen Datenpakete
  - bis zum Empfänger

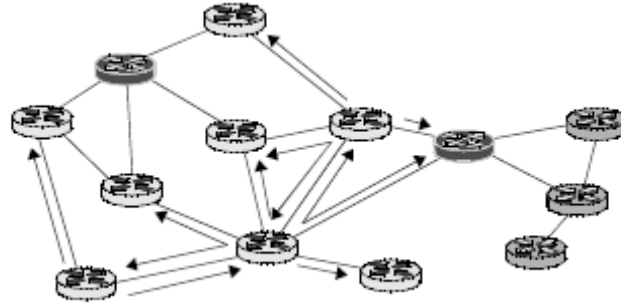


- empfangens Paket wird weitergesendet
  - Nachricht wird auch zurückgesendet
  - immer an alle Nachbarn

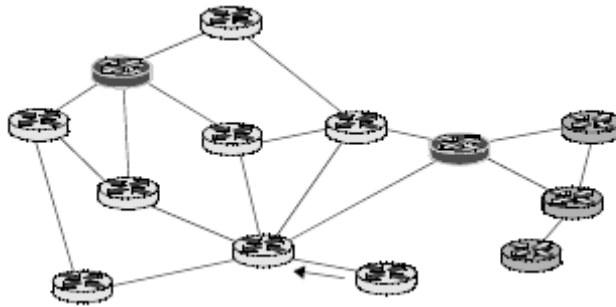


- Empfänger leitet Paket nicht mehr weiter
- andere Knoten schon
  - keine Information ob das Paket schon angekommen ist

- sehr viele Knoten empfangen Nachricht



- selbst nachdem der Empfänger die Nachricht erhalten hat, können noch Nachrichten im Netzwerk gesendet werden
  - im Beispiel nur für eine Nachricht der Fall
- jeder Knoten/Router besitzt nur sehr beschränktes Wissen



#### Eigenschaften:

- Sehr hoher Overhead
- Zuverlässig
- Scheint wenig geeignet für Datenpakete

#### Nennen Sie 4 Anforderungen, die Routing in mobilen Adhoc-Netzwerken von kabelgebundenen Netzwerken unterscheidet. S148,149

Drahtlose Verbindung, selbstorganisierend, ohne vorgegebene Netzwerkstruktur, Knoten können ihre Position verändern.

- Lokaler Broadcast
- Hoher Packet loss
- Geringe Bandbreite
- Hohe Mobilität – Routenänderung, Verbindungsverlust
- Energieversorgung
- Sicherheit

#### Wodurch zeichnen sich symmetrische kryptographische Algorithmen aus? Geben Sie 2 Beispiele für solche Algorithmen. S255

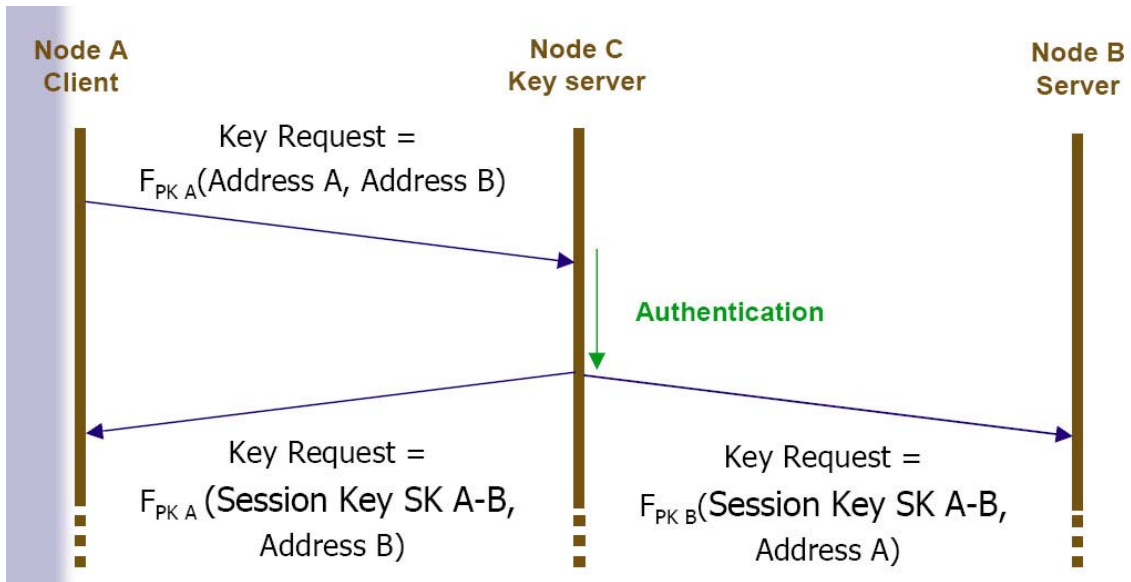
- Blockchiffren sind symmetrische Verfahren => Sender und Empfänger nutzen **gemeinsamen** Schlüssel
- Blockchiffren beschreiben Verschlüsselung eines Blocks
- Zusammenspiel bei mehreren Blöcken: Verarbeitungsmodi (= Grundchiffre + Rückkoppelung + einfache Operationen)
- Verschlüsseln in 8 Byte Blocklänge
- Vier Modi für Blockchiffren: Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB) und Output Feedback (OFB)

Bsp1: Data Encryption Standard (DES) mit 64Bit Blocklänge

Bsp2: Nachfolger von DES -> Advanced Encryption Standard (AES), mind 128bit Blocklänge

#### Wie wird in BACnet ein Sessionkey zwischen zwei Knoten ausgetauscht? (3/19) S261

BACnet = Building Automation and Control Networks ist ein [Netzwerkprotokoll](#) für die [Gebäudeautomation](#).



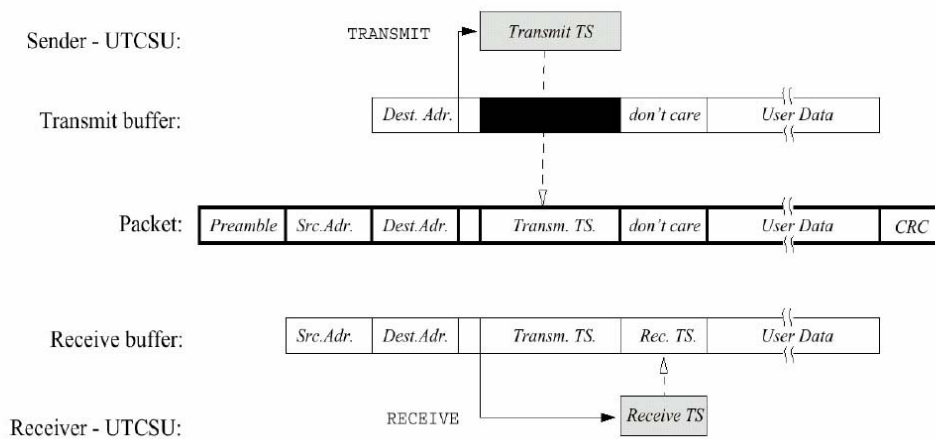
Welche 3 Parameter beeinflussen die netzwerkbasierende Zeitsynchronisation und wodurch werden sie verursacht? S232

-

Was versteht man unter Hardware Time-Stamping? (6/19) S233

## On-The-Fly Hardware Packet Time Stamping

- CPU assembles CSP - Clock Sync Packet
- Hardware generated time stamps are copied into the packet while it is actually sent (received)



Nennen Sie Unterschiede von NOS und DOS. (6/19) S6



	System images	Autonomy of the "nodes"	Fault tolerance
NOS	n	higher	low
DOS	1	lower	high

Network Operating System:	Distributed Operating System:
<ul style="list-style-type: none"> <li>• <b>User kennt n machines</b></li> <li>• <b>Explicit</b> <ul style="list-style-type: none"> <li>○ File transfer</li> <li>○ remote login</li> </ul> </li> <li>• <b>N kernels running</b> <ul style="list-style-type: none"> <li>○ Probably identical</li> </ul> </li> <li>• <b>Knoten responsible for</b> <ul style="list-style-type: none"> <li>○ <b>Themselves</b></li> <li>○ <b>Network services</b></li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• „Transparent“</li> <li>• <b>Single system image</b></li> <li>• <b>Organisiert and verteilt</b> <ul style="list-style-type: none"> <li>○ Jobs, resources, ...</li> </ul> </li> <li>• <b>Dynamische allocation (Bereitstellung) von Ressourcen</b></li> <li>• <b>1 kernel</b></li> <li>• <b>Fault tolerance</b> <ul style="list-style-type: none"> <li>○ degradable Systemperformance möglich</li> </ul> </li> <li>• „True“ DOS</li> </ul>

## Bustopologie S13-15



Alle Endgeräte sind an den Bus angeschlossen.

Eine **Bus-Topologie** (Linien- oder Strangtopologie) besteht aus einem Hauptkabel, dem **Bus**, an das alle Geräte und zwei Endwiderstände angeschlossen sind. Diese Abschlußwiderstände mit dem Leitungswellenwiderstand ( $Z = 50 \text{ Ohm}$  bei Koaxialkabel) dienen zu Verhinderung von Reflexionen. Der Anschluss zwischen den Geräten (also **Netzkarten**) und Hauptkabel erfolgt über **T-Stücke**.

Zugriffsverfahren (z.B. **CSMA/CD**) versuchen zu verhindern, dass sich die Teilnehmer gegenseitig stören. Sie regeln, welcher Teilnehmer die gemeinsame Leitung – den **Bus** – zu welchem Zeitpunkt zur Verfügung hat.

### Bei diesem Verfahren treten folgende Probleme auf:

- Während des Datenverkehrs muss jeder Teilnehmer jede Sendung mithören. Dadurch steigt die Belastung (Strom) der Ausgangsbaugruppen des Senders mit der Anzahl der Teilnehmer am Bus.
- Feldbussysteme können sich über einen Bereich von mehreren hundert Metern erstrecken. Hier ist die Leitungslänge im Vergleich mit der Wellenlänge der Übertragung nicht mehr vernachlässigbar klein. Um störende Reflexionen zu vermeiden, werden Busabschlusswiderstände benötigt, die die Ausgänge des Senders ebenfalls mit höheren Strömen belasten. Kleinere Feldbussysteme können dennoch sehr gut nach dem Bus-Prinzip vernetzt werden.

### Vorteile:

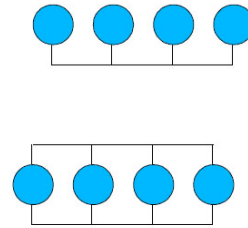
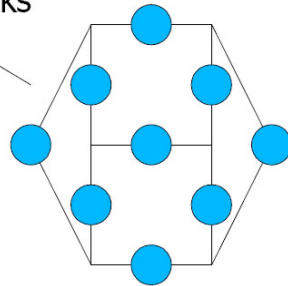
- Der Ausfall eines Rechners hat keine Konsequenzen
- Nur geringe Kosten, da nur geringe Kabelmengen erforderlich sind
- Einfache Verkabelung und Netzerweiterung
- Es werden keine weiteren Rechner zur Übermittlung der Daten benötigt

## Nachteile:

- Alle Daten werden über ein einziges Kabel übertragen
- Datenübertragungen können leicht abgehört werden
- Eine Störung des Übertragungsmediums an einer einzigen Stelle im Bus (defektes Kabel, lockere Steckverbindung, defekte Netzwerkkarte) blockiert den gesamten Netzstrang (die Suche nach der Fehlerquelle ist dann oft sehr aufwändig)
- Es kann immer nur eine Station Daten senden. Während der Sendung sind alle anderen blockiert (Datenstau)

### ▪ Bus Networks

- Classical Bus
- Multibus
- Bus-Hypercubes
- Beta Networks



## Welche sind die 6 wichtigsten Sicherheitsaspekte? (6/19) S251

1. Confidentiality
2. Integrity
3. Availability
4. Authentication
5. Access Control
6. Non-Repudiation

Bei 1. bis 3. spricht man auch von CIA.

## Was ist Flooding? S118-120

Eine sehr einfache Möglichkeit, damit ein Router weiß, wohin er ein Datenpaket senden soll bzw. wie eine Nachricht von Knoten A zu Knoten B gelangt ist Flooding.

Flooding von Nachrichten:

- Sehr hoher Overhead
- Zuverlässig
  - viele Pfade
  - Reliable Broadcast
- Scheint wenig geeignet für Datenpakete
  - i.A. werden sehr viele Datenpakete getauscht
  - Netzwerk nicht mit Datenpaketen überfluten
- Ist Flooding dennoch sinnvoll?

## Unterschied zw. RMI, CORBA und COM+? (14/19) S107,108

### Vergleiche die 3 Standards von CEE`S? S105

Java Technologie (J2EE, EJB, JRMI-Java Remote Method Invocation):

- Viele Programmiersprache, viele Plattformen
  - verwendet die Java Philosophie „write once, read everywhere“
  - muss in Java implementiert sein
  - der EJB Standard ist seit April 1999 ein Teil von CCM

CORBA Standards:

- Eine Programmiersprachen, viele Plattformen
  - durch die Object Management Group (OMG) definiert und gemanagt

- Anbieterunabhängige Spezifikation (Vendor-neutral specification)
- dominiert infrastrukturellen „Backbone“ von distributed Services

### COM+ (MS Technologie - .NET):

- Viele Programmiersprachen, eine Plattform
  - binäre Spezifikation, Sprachenunabhängig
  - Vorherrschend am Desktop
  - Mittelpunkt ist die Windows Umgebung

	Java	CORBA	COM+
<b>Protocol</b>	JRMI utilizing Java Remote Method Protocol (JRMP)	IIOP	Object Remote Procedure Call (ORPC)
<b>Interfaces</b>	Home, Remote	Home, Remote, Service, Metadata	Multiple
<b>Middleware Component Model</b>	EJB	CCM (in CORBA 3.0)	MTS
<b>Implementation</b>	Java only	Binds to any conforming implementation	Language neutral

### Was passiert wenn ein immutable file geschrieben wird S37

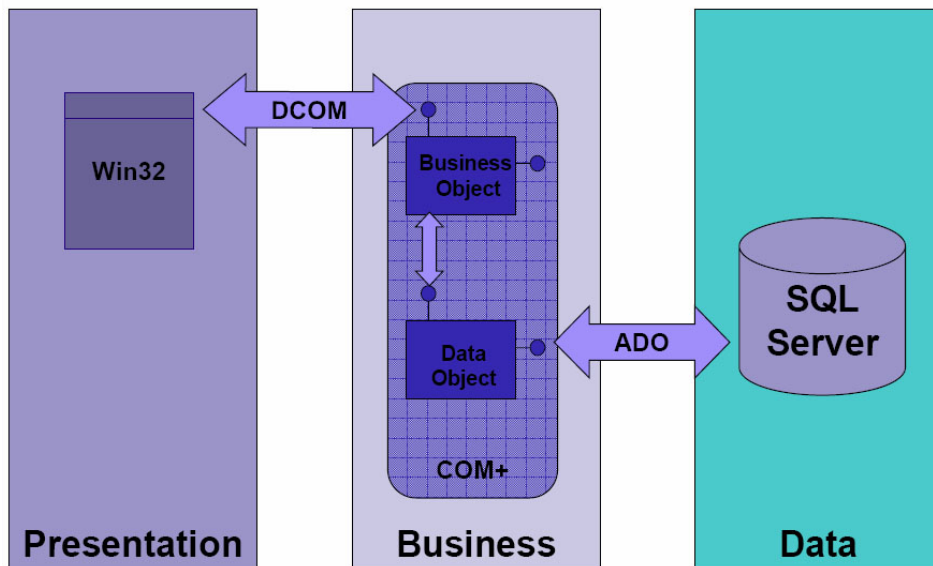
-> Distributed File Systems:

- CFS Cedar File System
- Dateien werden nie verändert, sobald sie angelegt werden
- Update Mechanismus
  - ✓ test.txt!1 gespeichert, vom Client gelesen
  - ✓ Client verändert Inhalte
  - ✓ speichert test.txt!2
- Performanceerhöhung
  - ✓ nur Veränderungen speichern
  - ✓ (sehr) alte Versionen löschen

### Was ist persistence Referenz (11/19) S56

- die Lebenszyklen von Servern und Clients sind nicht synchron
- automatische Reaktivierung von einem schlafenden Server
- Server-Objekte speichern ihren Status vor dem Timeout
- träges Laden im Aktivierungsprozess

## COM Anwendung? S95

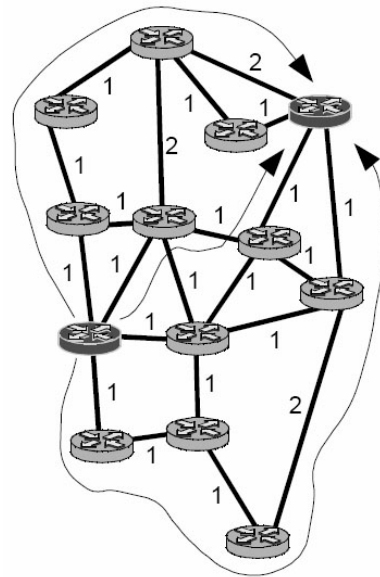


## Was macht ein Metrik? S127

Eine **Routing-Metrik** ist ein Wert, mit dessen Hilfe ein Routing-Algorithmus feststellen kann, ob eine Route im Vergleich zu einer anderen besser ist (Bei mehreren möglichen Routen wird eine Route mit kleiner Distanz im Sinne der Metrik bevorzugt.).

Metriken können Informationen wie z. B. Bandbreite, Verzögerung, Hop Count, Pfadkosten, Last, MTU, Verlässlichkeit und Kommunikationskosten berücksichtigen.

- Jede Verbindung wird bemessen
- Routingalgorithmus wertet diese Maße aus (Metrik)
- Unterschiedliche Routen werden vergleicht
- Optimalität
- Teilweise abstrakte Kosten
  - Entfernung (Hop Count)
  - Reale finanzielle Kosten
  - Netzwerklast
  - Verzögerung
  - Bandbreite



## Welche Arten von Persistentmanagement gibt es? S66

### Bean managed:

- Bean provider schreibt Routinen um einen Zugang zum Data Store herzustellen
  - deklariert Instance Variablen, die die persistenten Daten beinhalten
  - Finder method implementation wird vom Bean provider geschrieben
- Container ruft diese Routinen zu einem angemessenen Zeitpunkt im Lifecycle auf
  - ist gebräuchlicher, wenn der zu Grunde liegende Speicher (store) eine Applikation ist.

### Container managed:

- Erlaubt Bean logisch unabhängig von der Datenquelle zu sein
  - z.b. gleicher Code für relationale Datenbank, IMS Datenbank, etc.

- Container erzeugt Code um den Datenspeicher (data store) zu erreichen
  - Deployer maps the fields of the Bean to the columns of the data base
    - o Bean provider describes Bean's fields and relationships to other Bean in development descriptor
  - Container may use lazy access methods and caching
  - Finder methods are described in the deployment descriptor
    - o Description is in EJB QL
    - o Implementation is generated when Bean is deployed
- Virtual fields are used in the Bean to contain the persitant data
  - Access is via getXXX/setXXX methods.

## Welche Eigenschaften kann ein Programm besitzen, um ein intelligenter Agent zu sein? S223,224

Ein MAS (Multi Agent System) sollte „intelligenter“ als alle seine Komponenten zusammen sein. es sollte ein Software Programm sein, das:

- **Ziele** hat
- **unabhängig reagiert** um es zu erreichen
- **Strategien** und **Pläne** entwickelt, um
- mit anderen Agenten zu **kooperieren**

*INPUTS -> intelligent behavior -> OUTPUTS*

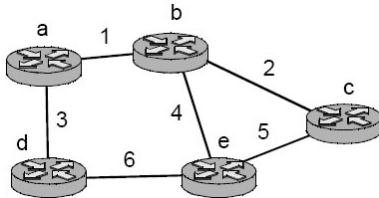
## Warum verwendet Link State/E.W.Dijkstra SPF? S135,136

**OSPF** ist ein dynamisches Routing-Protokoll innerhalb eines autonomen Systems. Es hat das Routing Information Protocol (RIP) als das Standard Interior Gateway Protocol (IGP) abgelöst, insbesondere bei großen Netzen. OSPF verwendet die Kosten eines Pfades als Metrik und kann bei gleichen Kosten lastverteilt arbeiten. Kosten werden bei OSPF standardmäßig aus der verfügbaren Bandbreite berechnet.

Jeder Knoten fügt die empfangenen Informationen der Link State Pakete zu einer Tabelle zusammen dieselbe Tabelle in allen Knoten.

Mit Hilfe dieser Daten wird der kürzeste Pfad berechnet

- Shortest Path First
- E.W.Deijkstra



Von	Nach	Verbindung	Entfernung
a	b	1	1
a	d	3	1
b	a	1	1
b	c	2	1
b	e	4	1
c	b	2	1
c	e	5	1
d	a	3	1
d	e	6	1
e	b	4	1
e	c	5	1
e	d	6	1

### E.W.Dijkstra:

- Der Algorithmus von Dijkstra dient der Berechnung eines kürzesten Pfades zwischen einem Startknoten und einem beliebigen Knoten in einem kantengewichteten Graphen.

#### Merkmale:

- OSPF garantiert ein schleifenfreies Routing im Gegensatz zu RIP (= verhindert Kreisrouting)
- Hello-Protokoll für die Überwachung der Nachbarn
- OSPF ist für große skalierbare Netze gut geeignet
- Das Area-Konzept vereinfacht die Kommunikation und Wartung
- OSPF ist ein offener Standard

## Welche Rolle gibt es beim COM Anwendungsentwicklung? ab S97

- **Design components as a client application**
  - Single user, single thread, no process concern
- **Write components in ANY language**
  - VB, VJ, VC, Cobol, whatever
- **Deploy as 1, 2, 3 or N-tier Application**
  - Drag and drop simplicity
- 

## Faulttolerance auf Zellebene? S219

- Massive redundancy
  - ✓ Of cells and cell groups
- Preemptive replacement (präventive Erneuerung)

## Was sind die 4 grundlegenden Funktionen eines System-Agents? S223

Ein Softwareprogramm das:

- **Ziele** hat
- **unabhängig reagiert** um es zu erreichen
- **Strategien** und **Pläne** entwickelt, um
- mit anderen Agenten zu **kooperieren**

## Schlüsselverteilung im BACnet? S263

- Initial key distribution – plain text
- Consecutive key distribution
  - ✓ Increment to the key (update key)
- Nur ein Schlüssel pro Knoten
- Symmetrischer 48 bit Schlüssel
- Vital areas of the Node are not protected
  - ✓ SNVT2 structures and the application data area can be read/written over the network

## Was ist ein LonMark-Objekt?(2/19) S173,174 (S179)

An object defines

- what NVs are where (NV#)
- how the NVs are called
- what NVs do what (semantics)
- behavior of the program that implements this object (but not the source code!)

**Ein Betriebssystem A läuft ununterbrochen, fällt aber jede Minute eine Millisekunde aus. Ein Betriebssystem B fällt niemals im Betrieb aus, muss aber jeden Tag 1 Stunde lang außer Betrieb genommen und gewartet werden.**

**F1: Welches System ist am Tag mehr verfügbarer?**

**F2: Welches System ist (während des Betriebs) zuverlässiger?**

- Das Betriebssystem A ist verfügbarer, da eine Millisekunde Ausfallzeit bei den meisten Arbeiten kaum auffällt.
- Das Betriebssystem B ist zuverlässiger, da es niemals im Betrieb ausfällt. Außerdem müsste der Zeitpunkt der Wartung bekannt sein und optimal gewählt sein.

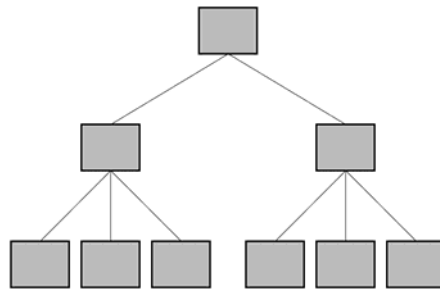
## Wie sieht die generelle Architektur eines Verzeichnisdienstes aus?

Ein Verzeichnisdienst (englisch: Directory Service (DS)) ist eine im Netzwerk **verteilte hierarchische Datenbank** (es bildet die Realwelt durch eine hierarchische Baumstruktur ab. Jeder Record hat also genau einen Vorgänger und somit genau ein Satz an der Wurzel der so entstehenden Baumstruktur), die auf dem Client-Server Prinzip basiert. In dieser Datenbank können beliebige Informationen gespeichert werden.

Die Einträge in der Datenbank können verglichen, gesucht, erstellt, modifiziert und gelöscht werden.

Meistens wird lesend auf die Daten eines Verzeichnisdienstes zugegriffen. Veränderungen an den Einträgen dieser Datenbank sind sehr selten. Aus diesem Grund bieten Verzeichnisdienste lesend eine wesentlich geringere Zugriffszeit als andere Datenbanken.

Der Aufbau der Directory Services erfolgt prinzipiell nach dem sogenannten X.500-Standard.



**Welche Aufgabe hat das Node-Objekt in einem LonMark-konformen LonWorks-Knoten?  
S173,174**