

Verteilte Systeme

Verteilte Betriebssysteme

Verteilte Betriebssysteme

Definition gemäß DIN 44300:

Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.

Fragen:

- Was ist ein verteiltes Betriebssystem?
- Welche der Abstraktionen, die ein Betriebssystem bietet, kann man Verteilen?
- Welche Garantien bietet ein verteiltes Betriebssystem?

Verteilte Betriebssysteme

Abstraktionsebenen für Verteilung:

- Programmiersprache -> RPC
- Objektmodell -> DCOM, CORBA, .NET Remoteing
- Arbeitsspeicher -> Orca, Linda, Beowulf
- Prozess -> Mosix
- Hintergrundspeicher -> Verteilts Dateisystem, Plan 9

Verteilte Programmiersprache

Programmiersprache bietet Konstrukte, die es ermöglichen bestimmte Operationen auf entfernten Rechner auszuführen, wie z.B.

- Funktionsaufrufe
- Serialisierung von Objekten bzw. Datenstrukturen

Probleme:

- Semantik?
- Adressierung?
- Umgang mit Kommunikationsfehlern?

Verteiltes Objektmodell

Programmiersprachenunabhängiges Objektmodell (Extreme Datenrepräsentation, Middleware) -> vgl. ALP 5

- Funktionsaufrufe
- Serialisierung von Objekten

Probleme:

- Objektmodell muss eine Untermenge der Objektmodelle aller beteiligten Programmiersprachen sein.
- Wie erfolgt der Zugriff auf geteilte Objekte?
- Welche Garantien werden gegeben (z.B. Konsistenz)?
- Wie wird mit Referenzen umgegangen (z.B. verteilte Garbage-Collection).

Verteilter Arbeitsspeicher

Bestimmte Bereiche des Arbeitsspeichers werden zwischen verschiedenen Prozessen geteilt (fließender Übergang zum verteilten Objektmodell!) - Programmierung einfacher als Nachrichtenaustausch.

Implementierung in der Programmiersprache/Objektmodell:

- Zugriffe passen zu Datenstrukturen
- Fein granularer Zugriff auf Daten regelbar

Implementierung als Teil der Seitenverwaltung:

- Einfach zu implementieren
- Beschränkt auf Vielfaches der Seitengröße
- Anordnung der Daten muss manuell erfolgen

Verteilter Arbeitsspeicher

Lokal:

- Schneller Datenaustausch zwischen Prozessen
- Geringer Kommunikationsaufwand durch Zugriff auf den gemeinsamen physischen Speicher.
- Konsistenz durch Cache-Kohärenz-Protokolle gegeben.

Verteilt:

- Hoher Kommunikationsaufwand durch Umsetzung von gemeinsamem Zugriff auf Nachrichtenaustausch.
- Konsistenz ist teuer - Möglichkeiten:
 - Konsistenz einschränken
 - langsamen Austausch in Kauf nehmen

Verteilte Dateisysteme

Erste Ideen zum Verteilen von Betriebssystemressourcen.

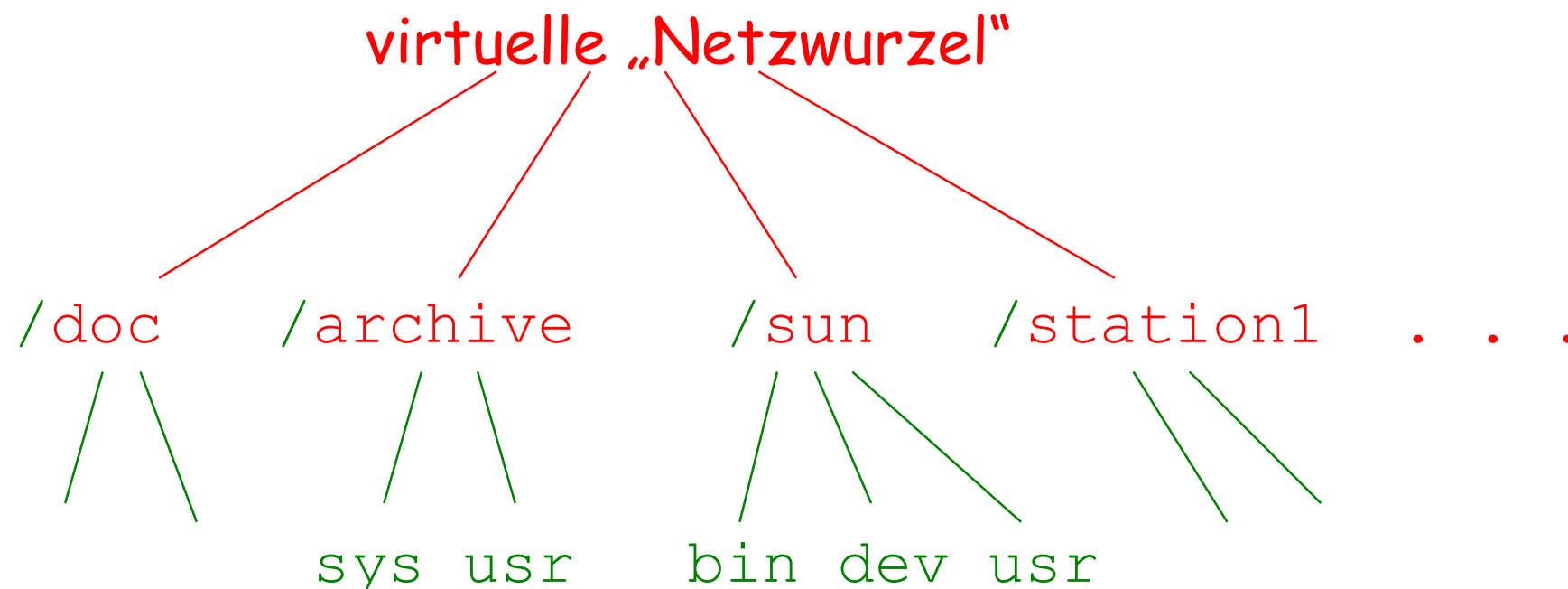
Idee:

- Dateisystemoperationen sind die älteste Methode zum Austausch von Daten zwischen Prozessen.
- Zugriffe tendenziell langsam (verglichen mit Memory)
- Klare Schnittstelle, in den meisten Betriebssystemen sehr ähnlich realisiert.

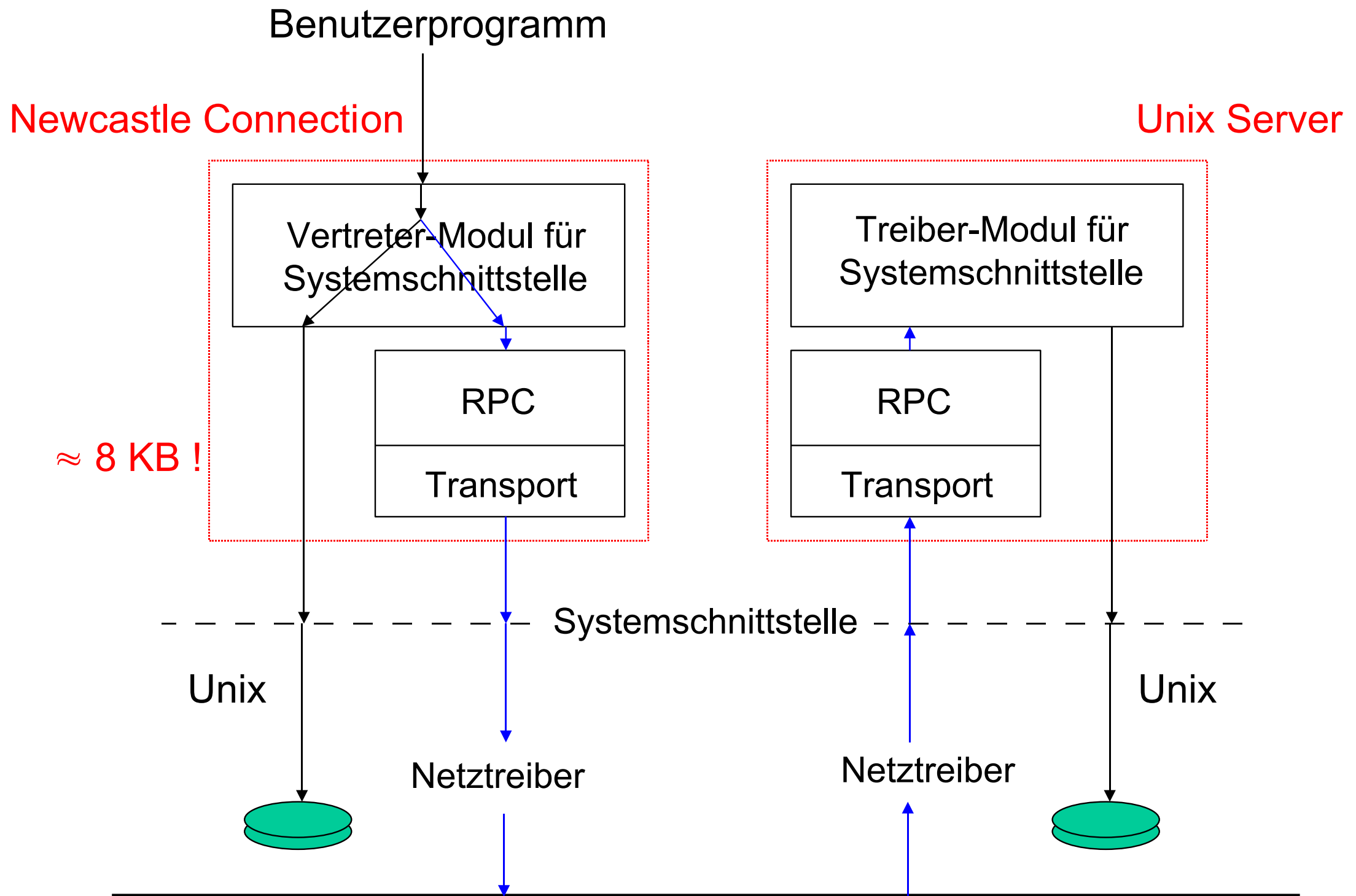
Verteilte Dateisysteme - United Unix

Idee:

- Modifikation der Systembibliotheken um ein gemeinsames, Verteiltes Dateisystem für Unix-Rechner zu realisieren.
- Entfernte Dateioperationen werden als RPC realisiert
- Zugriffe auf Gemeinsame Ressourcen erfolgen Durch spezielle Pfade, z.B. `/../doc/`



Verteilte Dateisysteme - United Unix



Verteilte Dateisysteme - United Unix

Vorteile:

- Unmodifizierter Betriebssystemkern
- Einheitlicher Namensraum für alle Netz-Ressourcen

Nachteile:

- Veränderung der Dateisystemsemantik (Netz-Wurzel über der Host-Wurzel)
- Damit keine echte Verteilungsabstraktion

Verteilte Dateisysteme - Sun NFS

Idee:

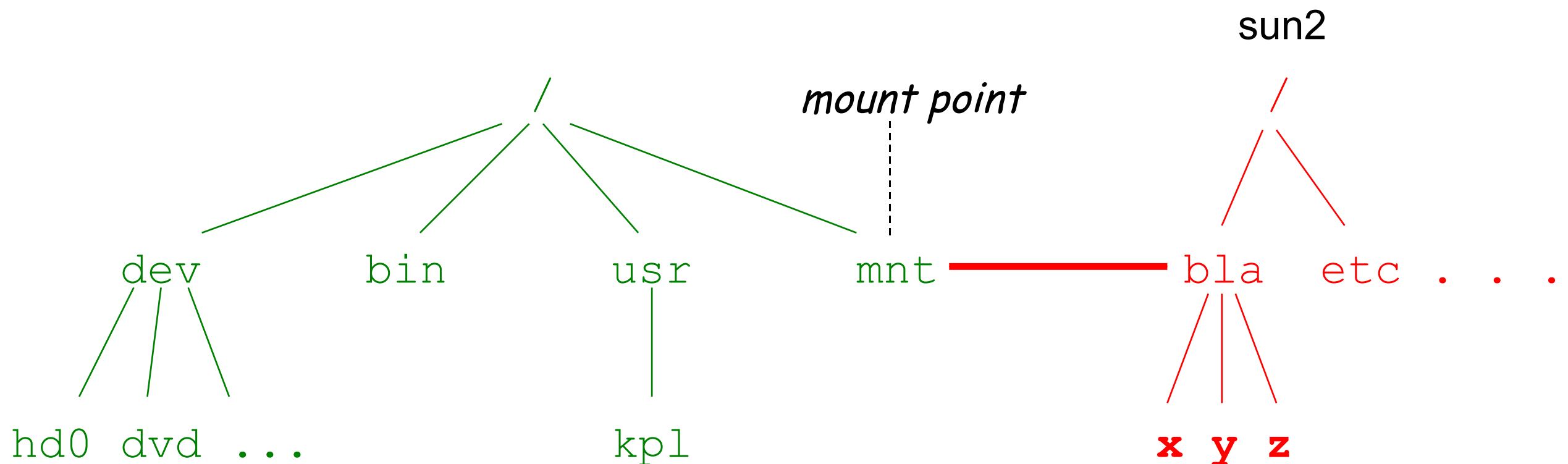
- Erweiterung der Mount-Point Idee aus dem Unix-FS auf Netzwerkressourcen.
- Zusätzliche Schicht im Systemkern Virtual Filesystem (VFS) abstrahiert zwischen Plattenzugriffen und Zugriffen auf entfernte Daten
- Realisierung der Operation per RPC im Systemkern

Verteilte Dateisysteme - Sun NFS

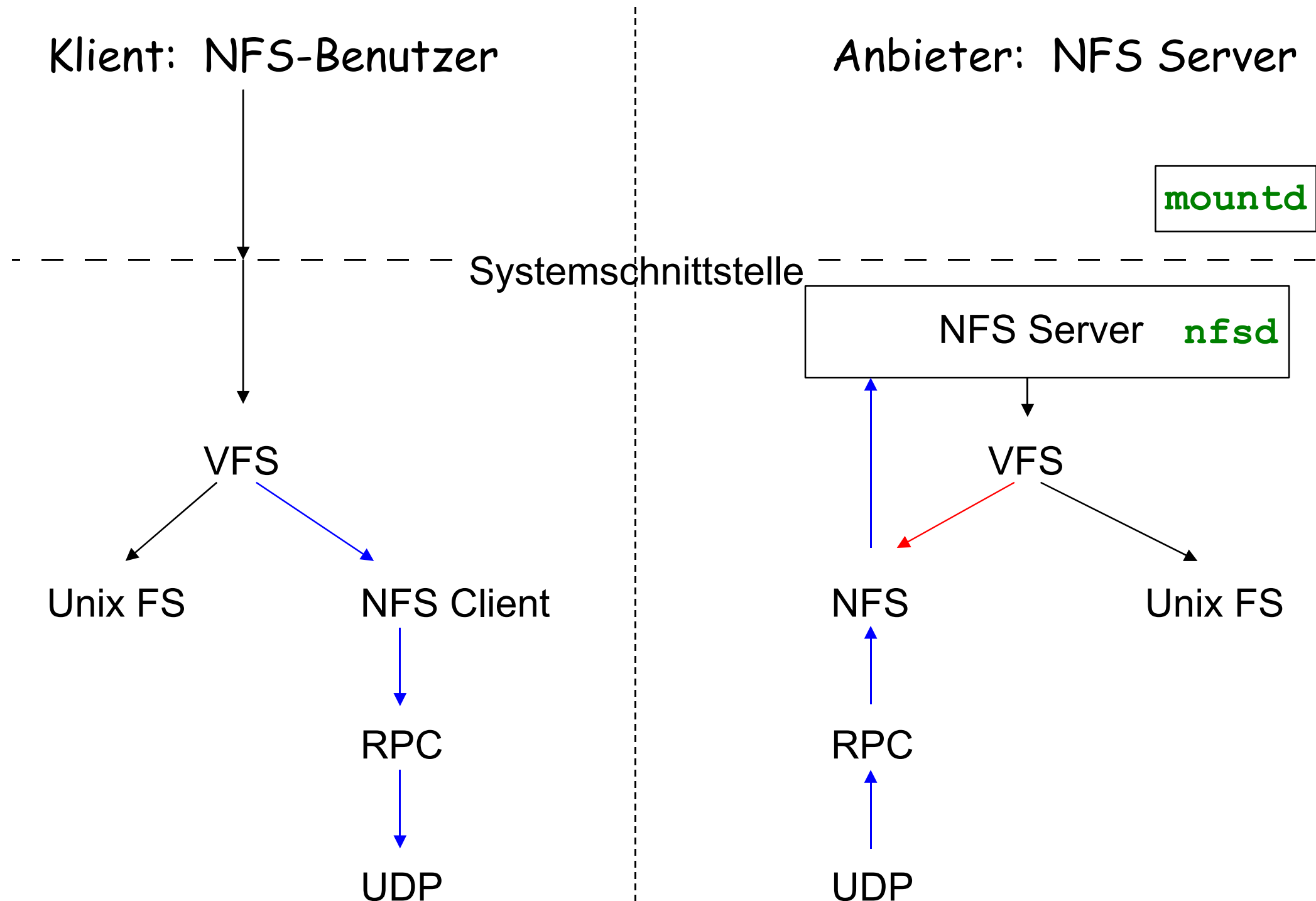
Erweiterung der Mount-Operation:

statt `mount /dev/hd0a /home` für lokale Platte

`mount sun2:/bla /mnt` für entfernte Daten



Verteilte Dateisysteme - Sun NFS



Verteilte Dateisysteme - Sun NFS

Vorteile:

- Flaches Dateisystem analog zu mehreren Platten im Rechner
- Zustandsloses RPC auf Server-Seite
- Liest und schreibt Blockweise
- Extrem stabil (auf Serverseite)

Nachteile:

- Keine exakte Einhaltung der Dateisystemsemantik per zustandslosem RPC möglich (z.B. offene File-Handles auf gelöschten Dateien)

Verteilte Dateisysteme - Andrew File System

Idee:

- Große Menge an Clients und Servern
- Directory-Server liefern eine globale Sicht für alle Rechner des Campus
- Lastverteilung durch Replikation
- Dauerhaftes Caching auf den Clients

Nachteil:

- Schwache Konsistenz durch modifizierte Dateisystemsemantik (write-invalidate)

Verteilte Betriebssysteme - Plan 9

Idee:

- Generalisierung der Unix-Idee “everything is a File”
- Dateisemantik z.B. auch für TCP-Sockets (/net/tcp)
- Redesign vieler Unix-Ideen
 - Modifikation von Symlinks
 - Mount-Points pro Prozess änderbar
- Verteiltes Dateisystem “Fossil” mit Snapshots und Archivierungsfunktion
- Erzeugung entfernter Prozesse mittels rfork()
- Automatische Verteilung mittels rfork() erzeugter Prozesse auf Compute-Knoten

Verteilte Betriebssysteme - Mosix

Idee:

- Singel-System-Image Cluster für Linux
- Erweiterung des Kernels um verteilten Arbeitsspeicher.
- Virtualisierung vieler Systemcalls, damit Transparent über das Netz funktionieren (vor allem IPC)
- Benutzt NFS als verteiltes Dateisystem
- Prozessmigration zur Lastverteilung (optional)

Verteilte Betriebssysteme - Amoeba

Idee:

- Echt Verteiltes Betriebssystem
- Nutzer soll nicht mehr sehen, wo Prozesse laufen.
- Alle Interprozesskommunikation ist RPC, damit ist es unerheblich wo Prozesse laufen
- Aller persistenter Speicher wird als verteiltes Dateisystem "Bullet-Fileserver" angeboten.
- Zugriff auf Ressourcen mittels Capabilities.
- Directory Server ermöglicht das Auffinden von Capabilities, die nicht vererbt wurden.

Verteilte Systeme

Klausur

Klausur

Termin:

12. Juli 2011, 8:00, ZIB-Hörsaal

Aufbau:

- 90 Minuten
- 90 Punkte
- Vier Aufgaben sind an Teile der Übungsaufgaben und Vorrechnenaufgaben angelehnt.
- Zwei Aufgaben an die Klausurübung

Hilfsmittel:

Ein handbeschriebenes DIN A4-Blatt, Stifte