Kap. 5:
Namens- und Verzeichnisdienste

Folien dieses Kapitels basieren überwiegend auf Folien von Prof. Dr. Gergeleit

5.1 Einführung

- In Kap. 4 RPC wurde besprochen:
  - Naming und Locating/Addressing von Diensten durch Binder
- Hier Verallgemeinerung:
  - Namensdienste
  - Verzeichnisdienste

5.2 Namen und Adressen

- **Namen:**
  - **Namen werden genutzt, um Objekte (z.B. eine Ressource oder einen Service) zu identifizieren.**
  - **Ein Name ist ein Bit- oder Zeichenstring**
  - **Binding: der Prozess, der einen Namen an ein Objekt bindet**

- **Eigenschaften von Namen**
  - ***unique*: ein Name identifiziert eindeutig (höchstens) ein Objekt**
  - ***pure*: ein Name ist nur ein Bit-Muster und enthält keinerlei weitere Information**
  - ***impure*: ein Name impliziert zusätzliche Information über das bezeichnete Objekt**

Beispiele

- unique
  - „Erika Mustermann“ ist nicht unique
    - » Name mit Geburtstag und Geburtsort sollte für Menschen unique sein
  - UUID (Globally Unique Identifier) (vgl. DCE Service-Namen) sind unique
    - » 128 Bit-Zahl
    - » enthält Netzwerk-Adressinformationen (z.B. Ethernet MAC-Adresse) und Zeitmarke
    - » generiert durch Tool `uuidgen`
- pure
  - UUIDs als Namen von DCOM-Objekten oder Klassen sind pure
- impure
  - DNS-Namen implizieren zusätzliche Information
    - » mail.informatik.fh-wiesbaden.de

Namensräume

- Namen werden in Namensräumen strukturiert
- Flache Namensräume (heute eher selten, z.B. Unix UIDs)
- Hierarchische Namensräume werden üblicherweise als gerichtete Graphen mit Labels organisiert
  - Verzeichnisknoten und Blätter
  - Absolute und relative Pfade
  - Globale und lokale Namen
- Beispiel: Unix-Dateinamensraum

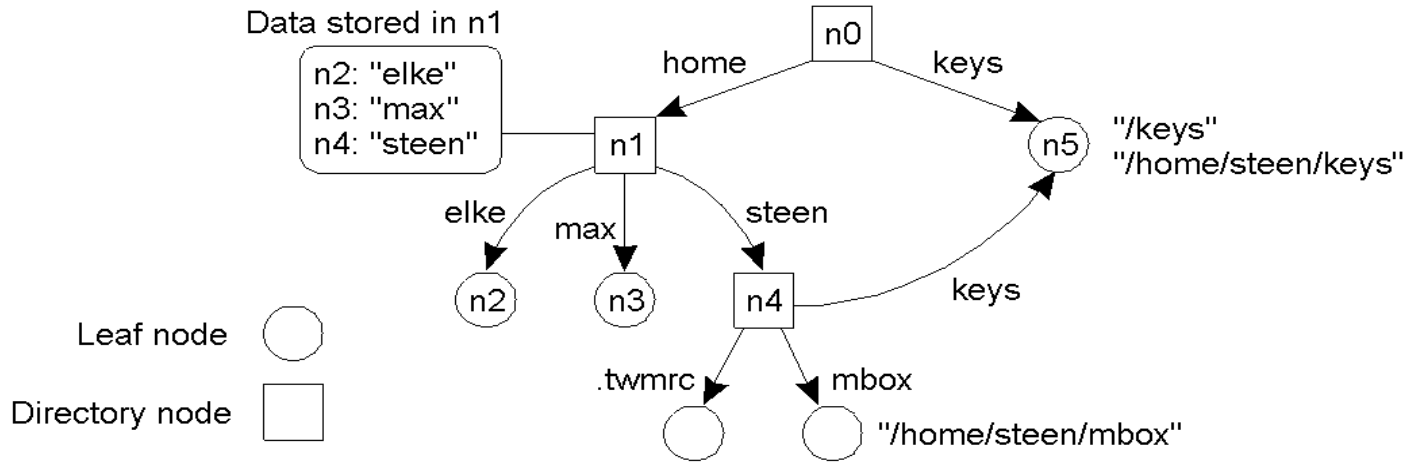


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Beispiel: MIB-2 - Namensraum

- **MIB: Management Information Base**

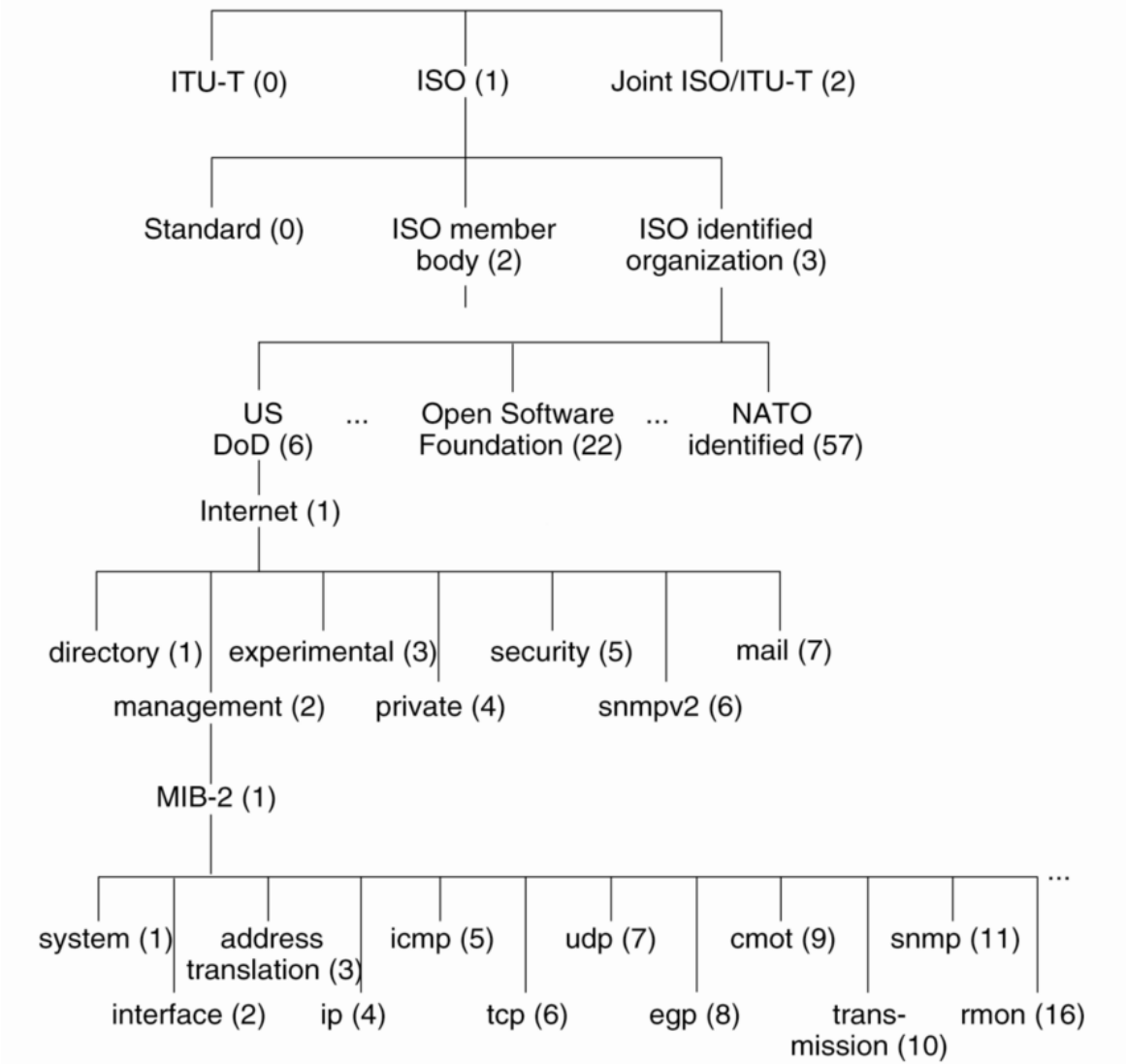


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Links in einem Namensraum

- **Das Objekt eines Namens ist ein weiterer Name**
- **Weiterleitung bzw. Abbildung eines Namens auf einen anderen Namen**
- **Beispiel: Unix Soft-Link**

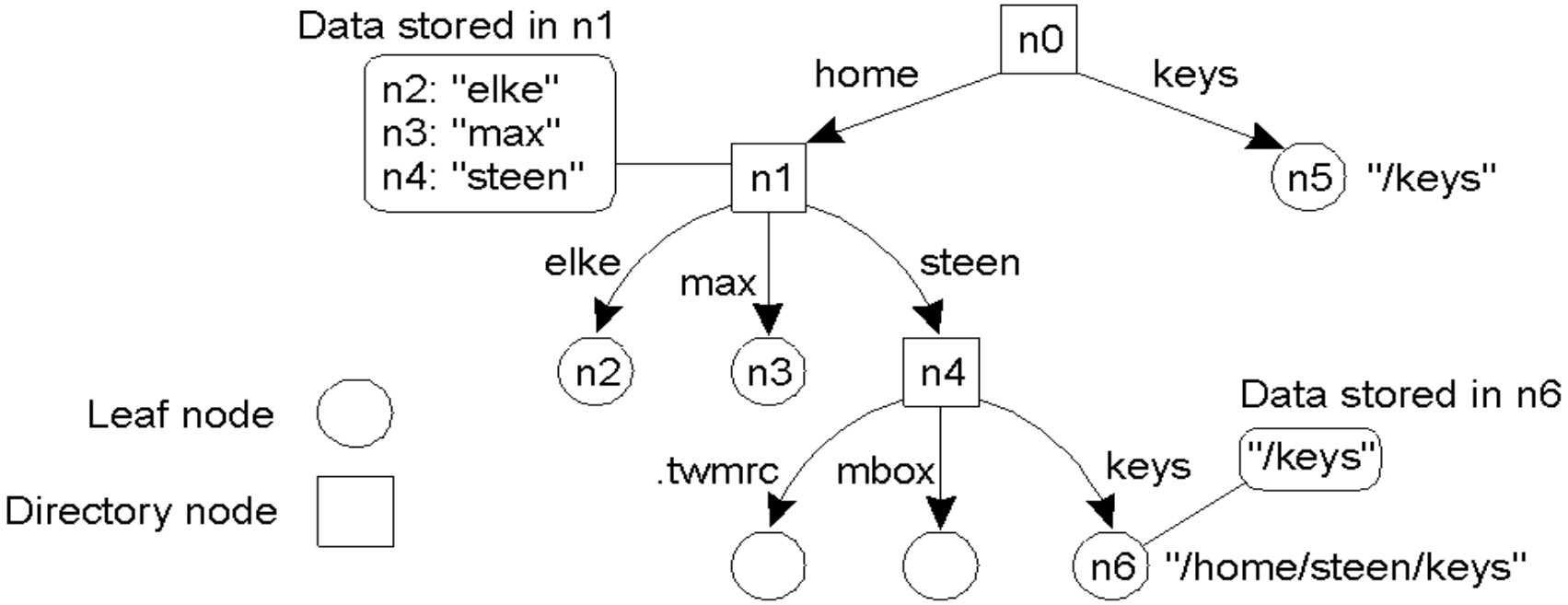


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Links in einen anderen Namensraum

- **Mounting**

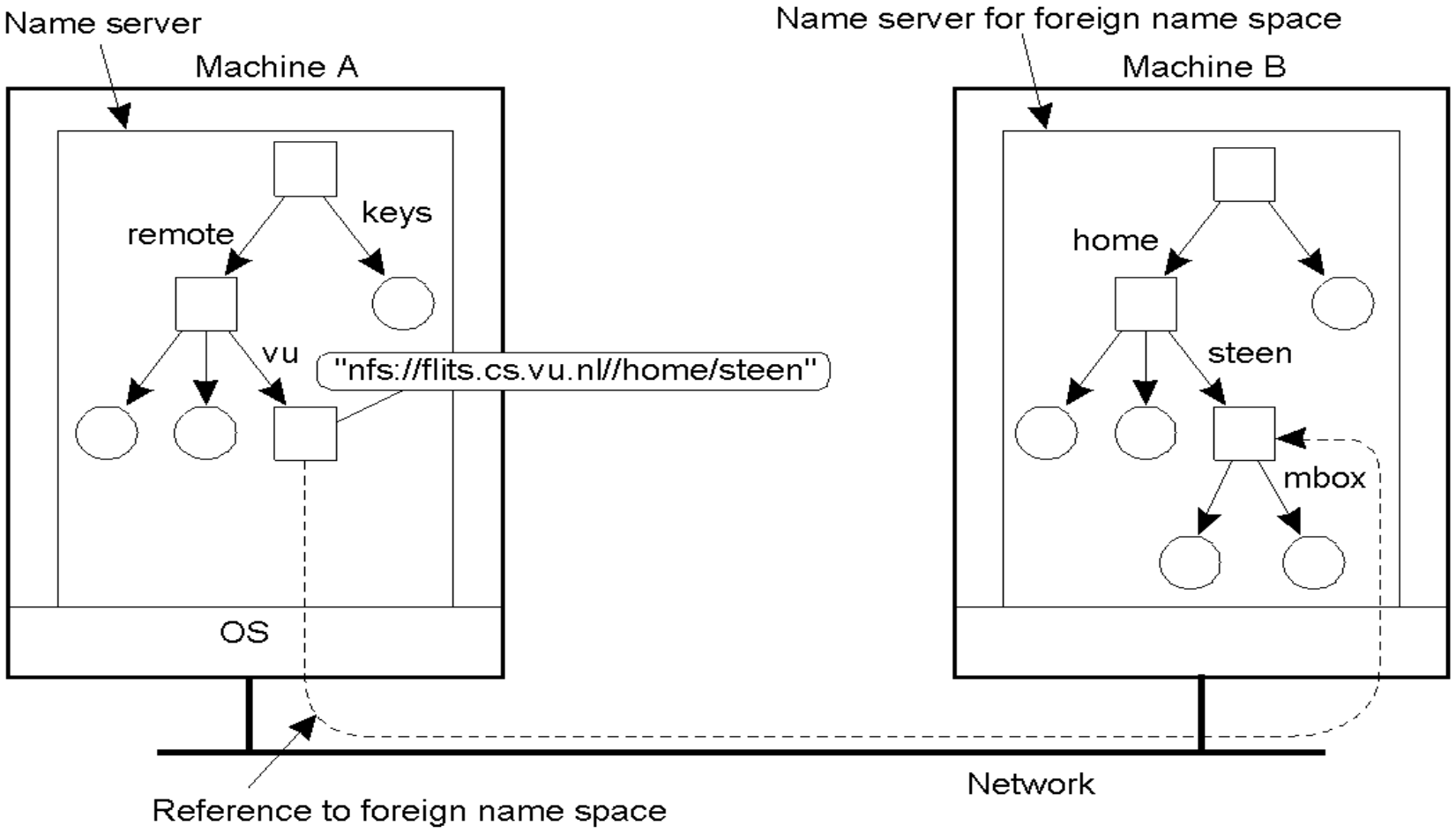


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Beispiel: Namensraum des DEC Global Name Service 5.2

- **Idee: Unter einer neuen Root werden bestehende Namensräume gemountet**
- **Probleme:**
  - **Aus /home/steen/mbox wird n0:/home/steen/mbox d.h. Namen ändern sich**

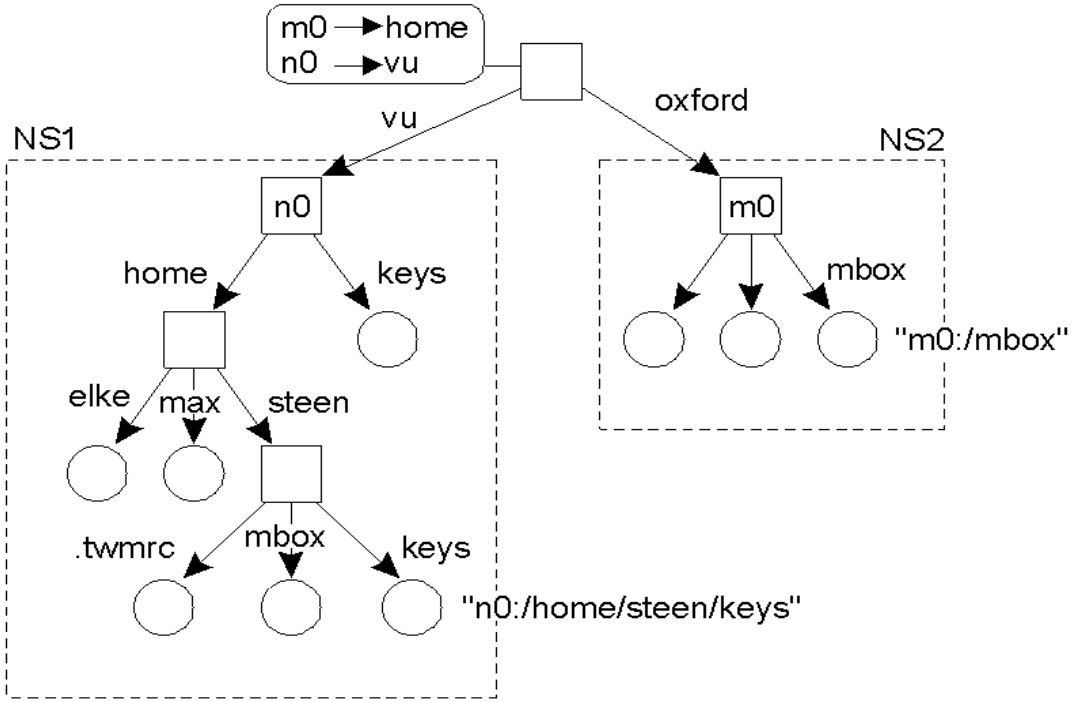


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

- **Weiterführung: URL Universal Resource Locator protocol://server/name**
  - **Access-Protokoll**
  - **Server**
  - **Name**

  http://wwwvs.cs.hs-rm.de:80/lehre/vm11vs

Organisation von großen Namensräumen

- Um große Namensräume effektiv verwalten zu können, sind diese typischerweise in drei Layer geteilt:
  - Global Layer
    - » High-level Knoten (Einstiegspunkte)
  - Administrative Layer
    - » Namensräume innerhalb einer Organisation
  - Managerial Layer
    - » Namensräume mit Namen, die sich häufig ändern
- Eigenschaften:

	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

Beispiel: DNS (Domain Name Service)

Vgl. LV Rechnernetze

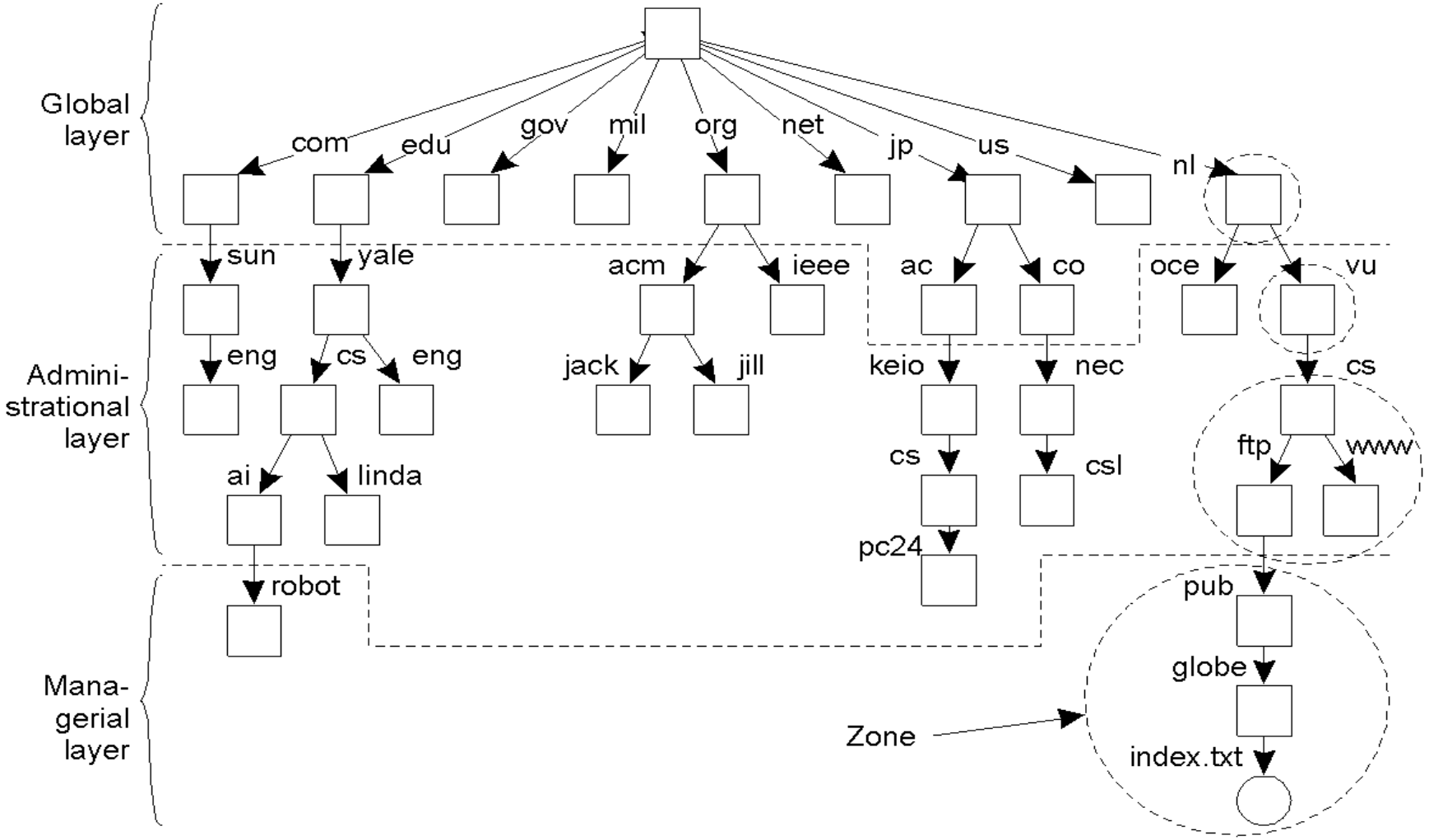


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Adressen

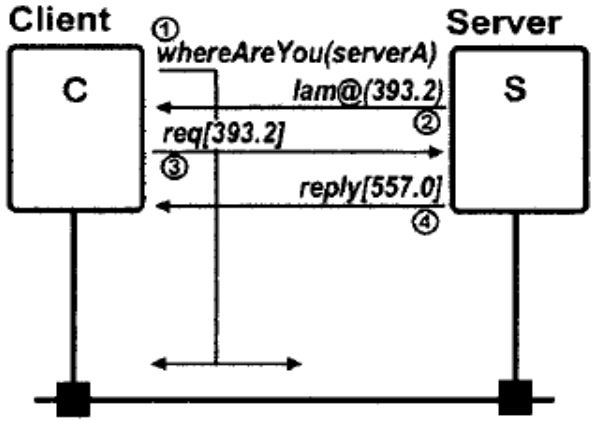
- Adressen sind Attribute von Namen, die genutzt werden können, um mit dem Objekt zu interagieren / zuzugreifen
  - Beispiele von Adressen
    - » Straße, Hausnr., Ort
    - » Telefonnr.
    - » (IP-Adresse, Portnummer)
    - » Speicheradresse
- Vorteile der Nutzung von Namen gegenüber Adressen
  - Ortunabhängig (wünschenswert)
  - Besser zu merken
  - Abstrahiert von vielen (Protokoll)-Details der Adresse

5.3 Namensdienste

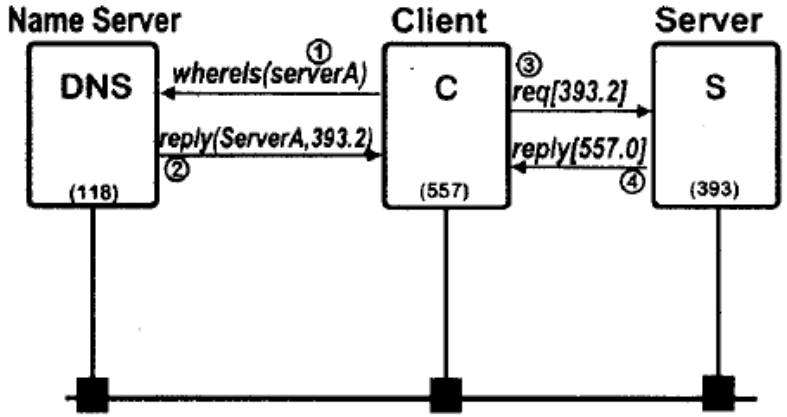
- **Namensauflösung:**
  Vorgang, um von gegebenem Namen eines Objekts zu seinem Adress-Attribut zu kommen
- **Namensdienst (Name Server):**
  Realisierung der Namensauflösung für anfragende Clients
  - **Im Falle von RPC-Systemen auch Binder genannt**
  - **Typische Operationen:**
    - **Register / Bind**
    - **Deregister / Unbind**
    - **Resolve / Lookup**

Arten der Namensauflösung

- **Suche durch Broadcast**
  - Anfrage wird an alle gesendet; nur die Einheit antwortet, die den Namen auflösen kann.
  - Nachteil: Skaliert nicht
  - Beispiel: ARP zur Auflösung von IP-Adressen (Namen) in MAC-Adressen



- **Nutzung Name Server**
  - Es wird ein dedizierter Server gefragt, der die Abbildung hält
  - Nachteil: benötigt well-known Adr.
  - Beispiel: DNS



Namensauflösung mit Name-Servern (1)

- **Iterative Namensauflösung**
  - **vom Client aus**
  - **Caching nur beim Client**

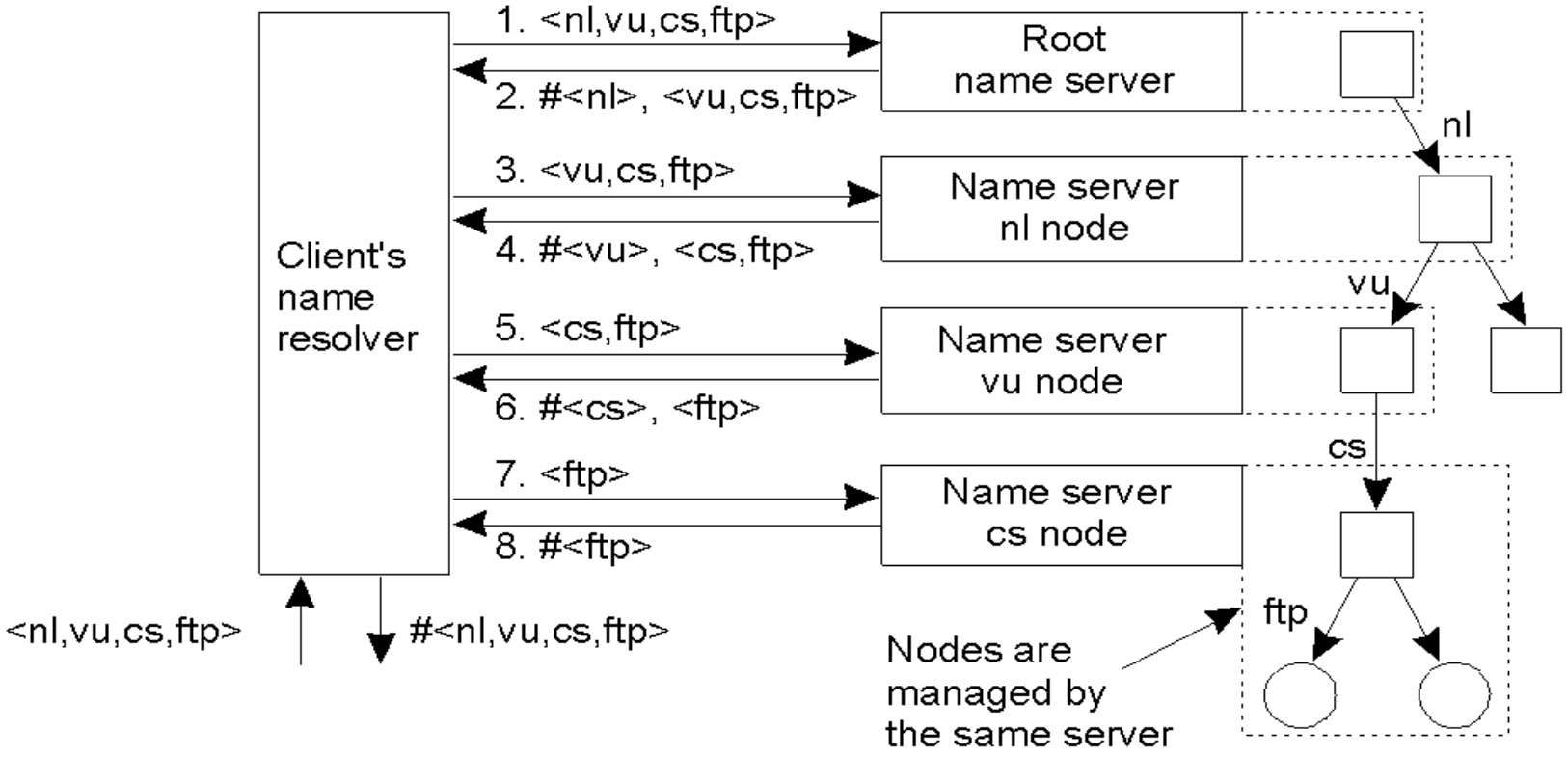


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Namensauflösung mit Name-Servern (2)

- **Rekursive Namensauflösung**
  - **Caching auf den Servern möglich**
  - **Weniger Kommunikation**
  - **Mehr Last auf den Root-Servern**

- **DNS**
  - **Rekursiv**
  - **Außer Root-Server**

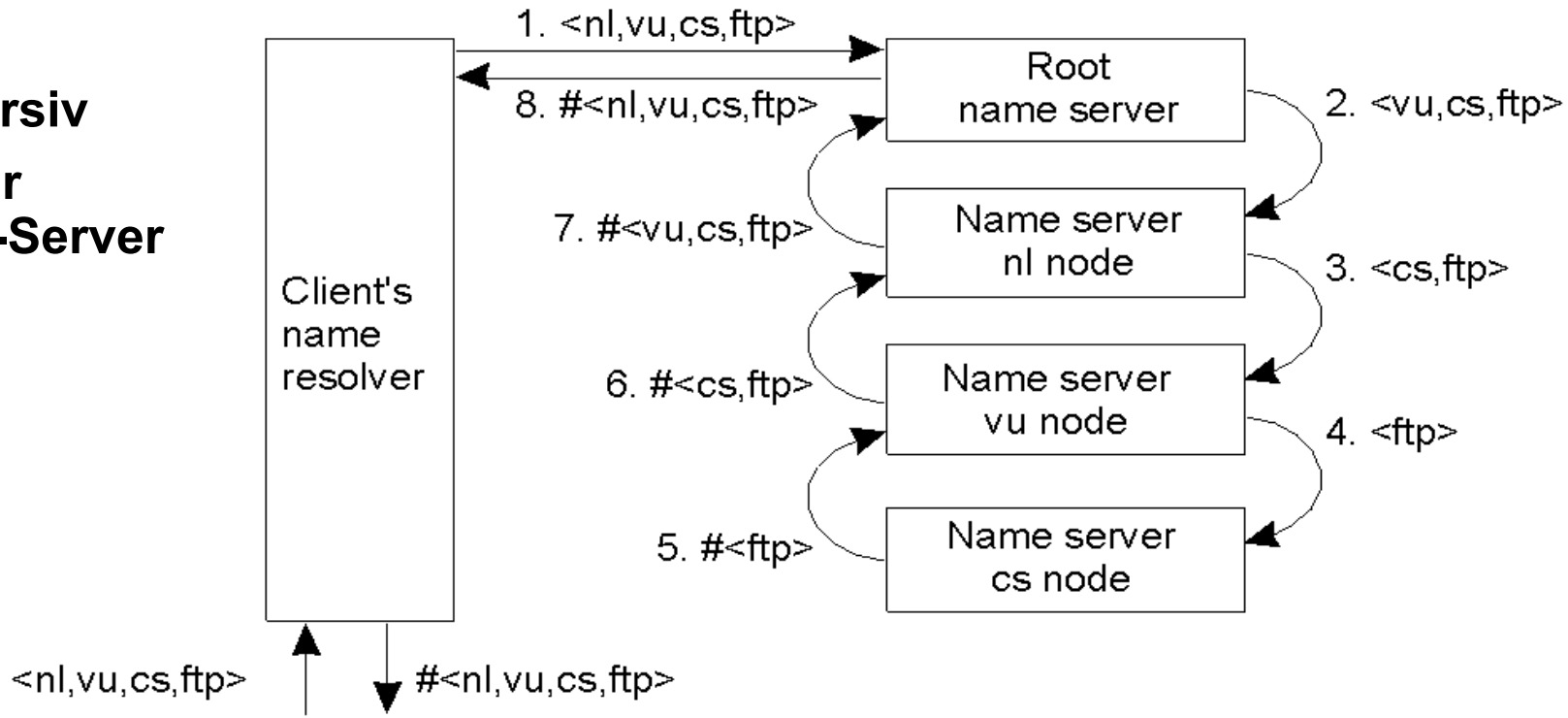


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Verbreitete Namensdienste

- DNS (Internet Domain Name Service)
  - Vgl. LV Rechnernetze
- JNDI (Java Naming and Directory Interface)
- Java RMI Registry
  - Vgl. Praktikum
- CORBA INS (Interoperable Naming Service)
  - URLs als Namen für CORBA-Objekte

5.4 Verzeichnisdienste

- Verzeichnisdienst = Directory Service
- Unterschied zu Namensdienst
  - Erweiterung
  - Analogie: „Gelben Seiten“ zu Telefonbuch
  - Im Verzeichnisdienst werden Einträge nicht in erster Linie über ihren Namen gesucht, sondern über Eigenschaften
- Standards
  - X.500 (ITU-T ehem. CCITT)
    - » Komplex, nutzte ISO/OSI-Stack und Directory Access Protocol (DAP)
  - LDAP (Lightweight Directory Access Protocol)
    - » Implementiert nur einen Teil des X.500-Standards
    - » Setzt auf TCP/IP auf
    - » LDAP = Lightweight-Version von DAP
    - » Heute versteht man unter LDAP nicht nur das Zugriffsprotokoll sondern auch den Server des Verzeichnisses (LDAP-Server)

LDAP Verzeichnis

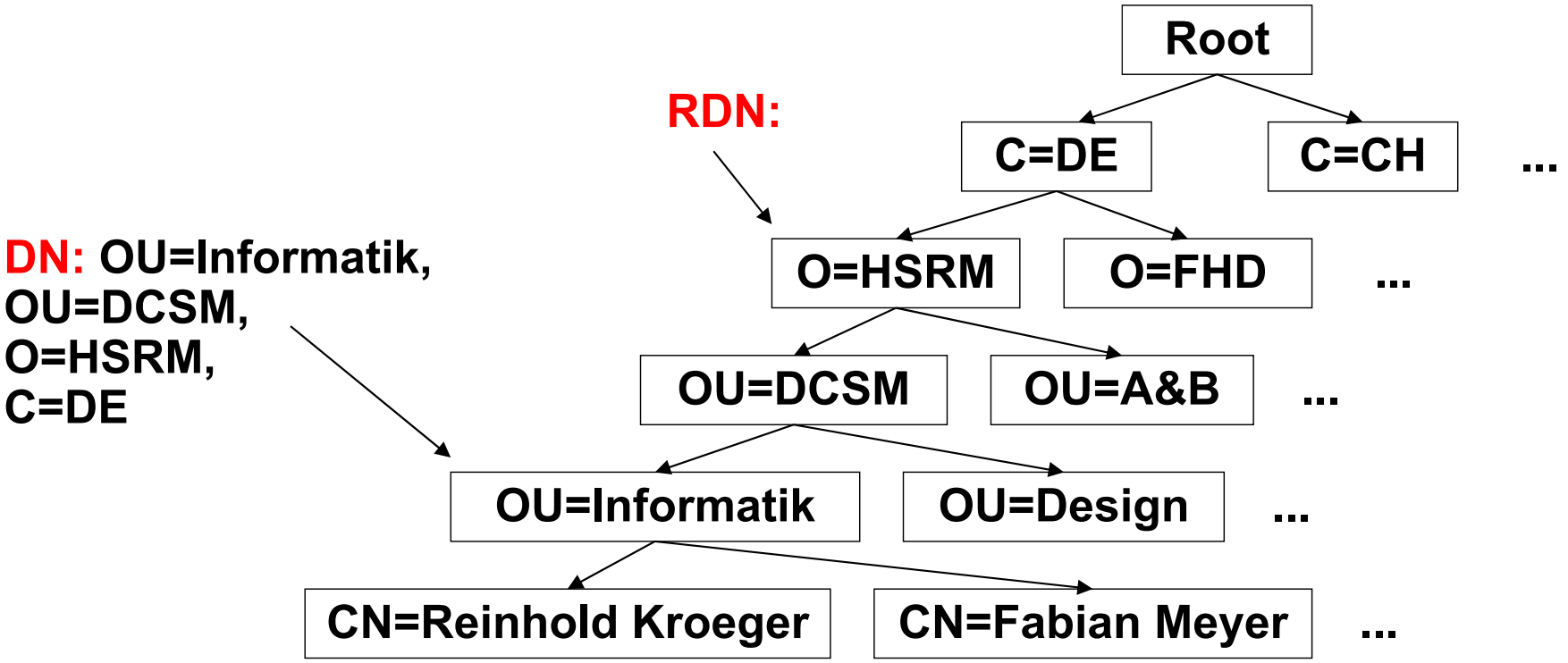
- Hierarchischer Namensraum: Directory Information Tree (DIT)
- Einträge (Knoten des Baums) können beliebige LDAP-Objekte sein
- LDAP-Objekt besteht aus Menge von <Attribut, Wert>-Paaren
- Klassen definieren Objekttypen mit bestimmten Attributmengen und Wertmengen
- Jedes Objekt gehört zu mindestens einer Klassen
- Es gibt Schemata für vordefinierte Klassen (z.B. Person, Organisation)
- Vererbung möglich
- Anwendungsspezifisch erweiterbar

Beispiel

Attribute	Abbr.	Value
Country	C	DE
Locality	L	Wiesbaden
Organization	O	HSRM
OrganizationalUnit	OU	DCSM
OrganizationalUnit	OU	Informatik
CommonName	CN	Reinhold Kroeger

RDN und DN

- Ausgangspunkt Directory Information Tree DIT
- Jeder Knoten hat in seiner Ebene einen eindeutigen Namen, genannt Relative Distinguished Name (RDN)
- Zusammensetzung der RDNs von Knoten bis zur Wurzel heisst Distinguished Name (DN) (vgl. Pfadnamen)



Anfragen

- Nutzung als Namensdienst
  - Finde Objekt bei gegebenem Distinguished Name
  - z.B. read(/C=DE/O=HSRM/OU=Informatik/CN=Reinhold Kroeger), damit Zugriff auf alle Attribute des Objekts
- Suche von Objekten mit bestimmten Attributwerten
  - Anfragen können mehrere Ergebnisse liefern
- Anfragen können komplex sein
  - Wildcards, Logische Ausdrücke: z.B. &(C=DE)(CN=*Kroeger)
- DIT kann über mehrere LDAP-Server partitioniert sein (Forwarding auf anderen Server)
- Clients und Server dürfen Teile der Abbildung cachen
  - Kein Cache-Koherenz-Protokoll
  - Nur Time-to-Live(TTL)-Zeiten
  - „Authorative Read“ unter Umgehung aller Caches möglich

Replikation

- Teile des Namensraums sind i.d.R. auf mehreren Servern repliziert
  - Aus Gründen der Fehlertoleranz und der Performance
  - Insbesondere die zentralen Teile
  - Replikation kann Stunden dauern
  - Master-Slave Konfiguration
    - » Änderungen nur auf Master
    - » Propagation an Slaves
- Problem: Update-Zeiten bei Änderungen
  - Updates sind nicht sofort global sichtbar
  - Vertretbar nur, wenn
    - » Read/Write-Verhältnis groß
    - » Das Lesen veralteter Einträge unkritisch ist

Anwendungen

- Benutzerverwaltung (Identity Management)
  - Schema: inetOrgPerson (RFC 2798)
- Adressbücher von Mail-Systemen
  - Z.B. Thunderbird-Schnittstelle zu LDAP
- Unternehmensorganisation
  - Information entsprechend Organigrammen
- Inventarsysteme / Infrastrukturverwaltung

Produkte LDAP/X.500

- OpenLDAP (Open Source)
- NetIQ eDirectory (früher Novell eDirectory, davor Novell Directory Services NDS)
- MS Active Directory (mit LDAP Interface)
- Atos DirX (früher Siemens DirX)
- Oracle Directory Server (früher Sun Directory Server)
- ...

5.5 Lokationsdienste

- **Bisherige Architektur kritisch, wenn Objekte schnell ihre (physikalische) Adresse ändern können**
  - **Jedes Mal müssten die Einträge im Name Server geändert werden (Problem Replikation/Caching)**
- **Lösung:**
  - **Aufteilung in Naming und Location-Service**
  - **Abbildung: Name → unique Entity ID → Location**
  - **Nur ein Update erforderlich**

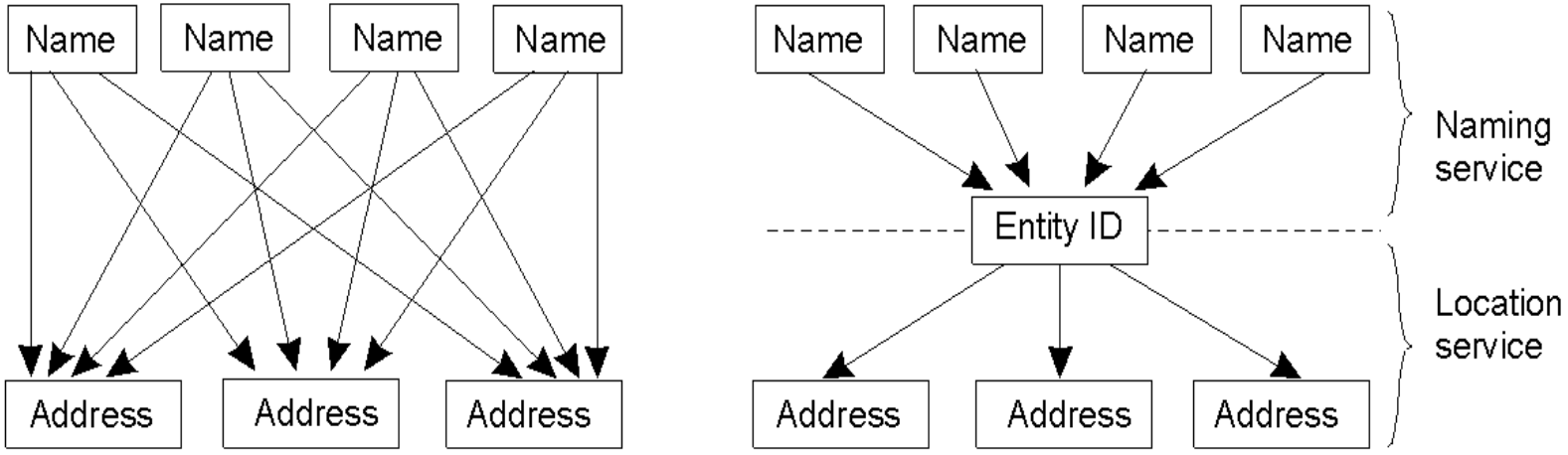


Abb. aus Tanenbaum, van Steen: Verteilte Systeme