

## Naming and Discovery

Buch: Kap. 5

### 27. Erläutern Sie die Begriffe "Name", "Identifizier", "Address" sowie den Bezug zwischen diesen Begriffen in der Praxis.

#### **Name:**

String, der eine Entität benennt (meistens benutzerfreundlich)

Beispiel: [www.tuwien.ac.at](http://www.tuwien.ac.at)

Vorteile:

- Meist „Human readable“
- unabhängig von Adresse (ortsunabhängig) durch Namensauflösung
- für Benutzer sind (hierarchische) Namen leichter zu merken als z.B. reine (IP) Adressen

Eigenschaften:

- Location independent
- Unique

#### **Address:**

Name eines Zugriffspunktes (Access Point). Direkte Verwendung einer Adresse ist problematisch, da dadurch die Location-, Replication- und Migration-Transparenz nicht unterstützt wird und Adressen auch oft für Menschen schwer lesbar sind. Gute Namen sollten daher die Adresse weder direkt, noch versteckt codieren.

Beispiel: IP-Adresse, zB 80.224.3.67

Kann sich ändern, wenn das Gerät an einen anderen Ort gebracht wird (keinen Location-Transparency)

#### **Identifizier:**

Identifizier ist „sehr direkt“ mit einer Entität verdrahtet. Mögliches Beispiel (?): MAC-Adresse

für eindeutige Kennzeichnung einer Entität, die folgenden Kriterien entspricht:

- Jeder Identifizier verweist höchstens auf eine Entität
- Auf jede Entität verweist höchstens ein Identifizier
- Ein Identifizier verweist immer auf die gleiche Entität

Identifizier können durch Counter oder Zufallszahlgeneratoren erzeugt werden. Wobei bei letzterem sichergestellt werden muss, dass sich Zahlen mit hoher Wahrscheinlichkeit NICHT wiederholen

### 28. Was ist ein "Name Space"? Erläutern Sie das Grundprinzip des "Closure Mechanismus" anhand eines Beispiels (zB Unix File System).

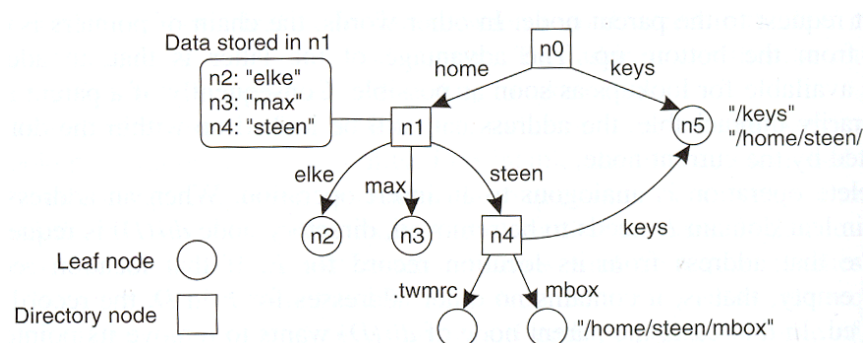


Figure 5-9. A general naming graph with a single root node.

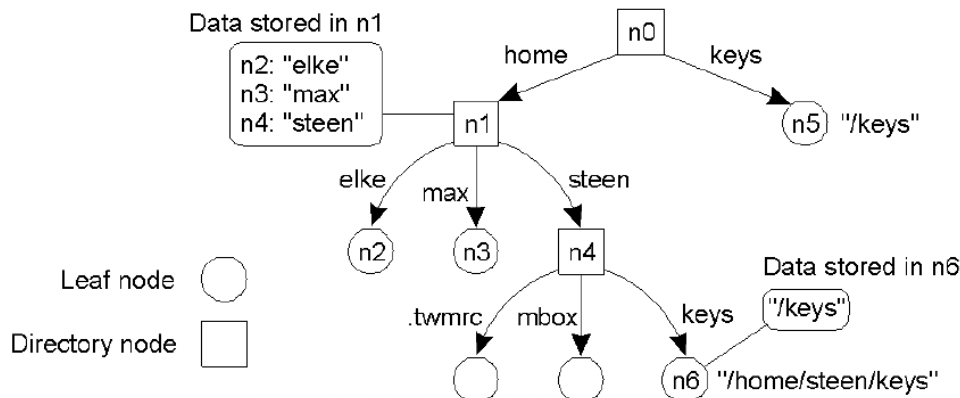
**Name Space:**

Ein Name Space organisiert Namen in einem benannten gerichteten Graph. Jeder Name ist immer relativ zu einem Verzeichnisknoten. Zur eindeutigen Zuordnung ist daher der entsprechende Kontext (eben der Namensraum) zu beachten.

- Blattknoten = benannte Entität
- Verzeichnisknoten (hat ausgehende Kanten): speichert eine Verzeichnistabelle --> Kantenbeschriftung, Knotenbezeichner Paare
- Root Node: Äußerster Namespace; Einstiegspunkt
- Path: Sequence of Labels. Absolute vs Relative (bezogen auf root-knoten); global vs. local

Durch **Aliases** können mehrere Namen auf die gleiche Entität zeigen.

- Hardlinks: mehrere absolute Pfade verweisen auf den selben Knoten im Graphen
- Symbolic link: speichern Pfad zu einer Entität



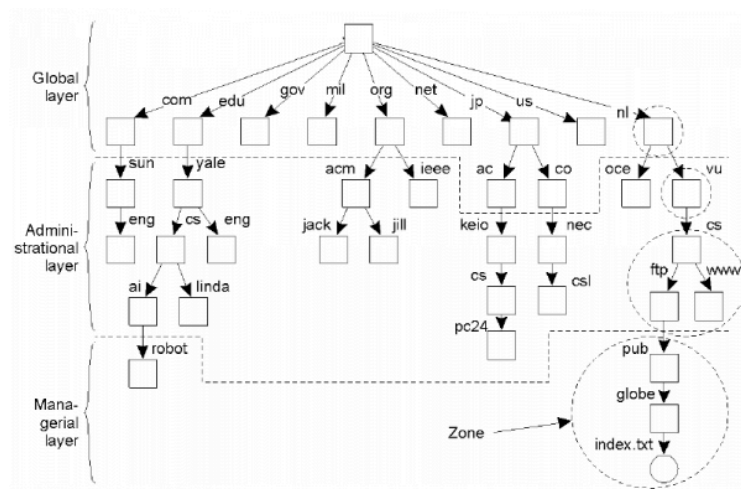
Durch **Mounting** können Namensräume miteinander kombiniert werden. (benötigt: access protocol, server, mounting point).

**Closure Mechanismus:**

zu wissen, wie und wo die Namensauflösung beginnen soll--> Auswahl des ersten Knotens.

- Bsp. Unix: um absoluten Verzeichnispfad (/users/home) aufzulösen, muss dem Dateisystem der Root-Knoten **"/"** bekannt sein. Der tatsächliche Offset des Root-Knotens ist im Superblock des logischen Laufwerks kodiert.
- Bsp. DNS: Der Closure Mechanismus bei DNS sind die IP-Adressen der 13 DNS-Root-Server

**29. Erklären Sie die Schichten der Verteilung von Name Spaces. Erläutern Sie die Einsatzmöglichkeiten von Replication und Caching in den verschiedenen Schichten. Erklären Sie verschiedene (hierarchische) Möglichkeiten der iterativen/rekursiven "name resolution".**

**Schichten:****Globale Schicht:**

- wenige Knoten der höchsten Ebene (Root-Knoten und seine Kind-Knoten)
- sehr stabil, da Änderungen sehr selten sind
- können Organisation oder Gruppen von Organisationen abbilden (z.B. com, org, Länder: at, de)

- geographische Ausdehnung: weltweit
- Reaktionszeit im Sekundenbereich
- Viele Replikate (da wenig Änderungen leicht zu replizieren)
- Effektives Caching durch Clients (da wenig Änderungen zu erwarten sind)

### **Administrations Schicht:**

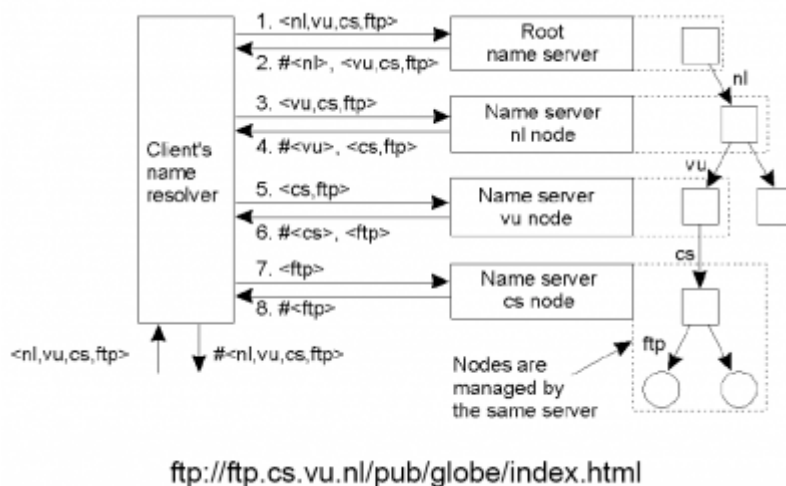
- Knoten, die innerhalb einer Organisation verwaltet werden
- Besteht aus vielen Knoten (im Vergleich zur globalen Schicht)
- Reaktionszeit im Millisekundenbereich
- keine bis wenige Replikate
- Caching durch Clients möglich

### **Management Schicht**

- Knoten die innerhalb einer Abteilung verwaltet werden
- besteht aus zahllosen Knoten
- Reaktionszeit: sofort
- keine Replikate
- Caching durch Clients manchmal möglich

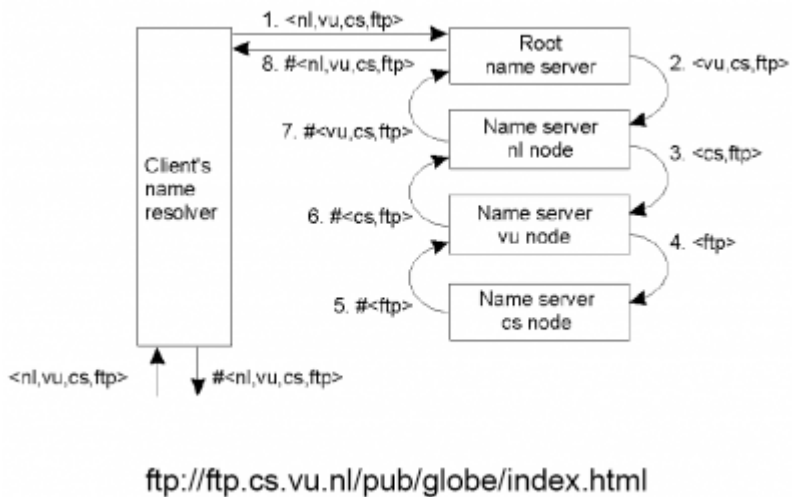
### **Iterative Namensauflösung:**

7. Der Resolver übergibt den vollständigen Namen an Root-Server.
8. Root-Server löst Name soweit wie möglich auf und gibt als Zwischenergebnis die Adresse des nächsten NS und dem verbleibenden Pfadnamen an den Client/Resolver zurück.
9. Client fragt den retournierten NS nach dem verbleibenden Pfadnamen.
10. Schritt 3 wiederholt sich solange, bis der gesamte Name aufgelöst wurde.



### **Rekursive Namensauflösung:**

1. Der Resolver übergibt den vollständigen Namen an den Root-Server.
  2. Wenn der NS den Namen nicht vollständig auflösen kann (oder aus dem Cache lesen kann) fragt er automatisch den nächsten NS ohne ein Zwischenergebnis an den Client zu senden.
  3. Sobald ein NS den Namen vollständig aufgelöst hat, wird das Endergebnis an den übergeordneten NS weitergegeben.
  4. Der Root-Server übergibt das vollständige Resultat an den Client.
- Vorteil: effektiveres Caching, geringere Kommunikationskosten
  - Nachteil: benötigt mehr Leistung am Server



Daher: Globale Schicht --> iterativ, Administrationsschicht --> rekursiv

- **Caching** steigert die Geschwindigkeit der Namensauflösung bei einem Cachehit und reduziert die Netzwerklast.
- **Replikation** verbessert die Availability, steigert die Geschwindigkeit durch „nähere“ Replikate und reduziert Hot-Spots.

### 30. Erläutern Sie das Domain Name System DNS, sowie den Ablauf bei der Namens-Auflösung anhand der DNS Database (Resource Records). Was ist reverse lookup? Was ist ein zone-transfer?

DNS ist das Namesystem für das Internet. Es wird verwendet, um Domainnamen auf IP-Adressen zu mappen. Eine Domain kann man sich als „Unterbaum“ eines hierarchischen Graphen vorstellen. Pfadnamen nennt man „Domainname“.

Fixe Liste mit 13 Root-Servern (einige dieser Server sind repliziert – d.h. unter einer IP Adresse sind mehrere Server erreichbar)

Knoten speichern ihre Daten in Ressource Records. Ressource Record ist die kleinste Informationseinheit im DNS.

#### Ausgewählte Ressource Record Typen:

- SOA: Start of Authority: Informationen zur dargestellten Zone
- A: IP eines Hosts
- MX: Mailserver
- NS: Name eines NS der die dargestellte Domain implementiert
- CNAME: kanonischer Name des Hosts (für reverse lookup)
- PTR: symbolischer Link auf Primären Namen des dargestellten Knoten

Caching und Replikation werden zur Steigerung der Effektivität verwendet.

(Ein Secondary NS kann ein Primary NS einer anderen Zone sein. Beide sind authoritative).

#### Reverse-Lookup

Wird benötigt, um von einer IP auf einen Domainname zu schließen. Dazu wurde eine eigene Domain (mit 3 Ebenen von) Subdomains) begründet: in-addr.arpa-Domäne. Somit lassen sich z.B. alle IP Adressen, die mit 193 beginnen in der 193.in-addr.arpa Domäne finden.

#### Zone-Transfer:

Zonenübertragung (Übertragung von Ressource Records) auf einen anderen NS also z.B. von Primary NS auf Secondary NS. Dies dient der Ausfallsicherheit und Performance.

### 31. Was ist Directory Service bzw. "Attribute-based naming"? Beschreiben Sie den prinzipiellen Aufbau des X.500 Name Space sowie dessen LDAP Implementierung.

#### Directory Service bzw. Attribute-based naming:

- Bietet einen Suchdienst nach Entitäten über deren Attribute
- Suche einer Entität über (Attribut,Wert)-Paare, in der Form /C=NL (Country=Netherlands)
- Attributbasierte Namensgebung
- „Yellow Pages“
- z.B. Wildcards Suche

**X.500:**

zB:

read /C=NL/L=Amsterdam/O=Vrije Universiteit/OU=Math.&amp;Comp. Sc./CN=Main server/

Attribute	Value
C	NL
L	Amsterdam
O	Vrije Universiteit
OU	Math & Comp. Sc
...	...

- Versuch eines umfassenden weltweiten Verzeichnissystems
- Zu komplex, war daher nicht erfolgreich
- Basierend auf OSI-Modell
- hierarchisch organisiertes Directory Service.
- Die Benennung einer Entität basiert auf Attribut-Wert Paaren (=attributbasierte Benennung).
- Es gibt einen einzigen Directory Information Tree (DIT), welcher Einträge hierarchisch speichert (Verteilung über mehrere Server möglich). Jeder Eintrag kann mehrere Attribute-Wert-Paare speichern.

Jeder Eintrag hat einen eindeutigen Distinguished Name, der eine Kombination aus seiner eigenen RDN (Relative Distinguished Name) und der RDN all seiner übergeordneten Einträgen ist.

Vorteil: effektives Suchen wird möglich (Besucher gibt Beschreibung dessen ein, was er sucht).

**LDAP (Lightweight Directory Access Protocol)**

Ist eine effizientere, einfachere Implementierung von X.500 auf TCP Basis. Verwendung: z.B. zum Beschreiben von Zugriffsberechtigungen für Benutzerauthentifizierung (MS Active Directory), oder als Verzeichnis für Personen und E-Mailadressen.

**32. Wie funktioniert in flachen Namensräumen das Location Service? Geben Sie das Grundprinzip möglicher Lösungen an und gehen Sie dabei auch auf die Begriffe Mobility und Discovery ein. Erläutern Sie Vor- und Nachteile bei der Verwendung von "Forwarding Pointers". Wie funktionieren "Home-based approaches" für mobile Geräte?**

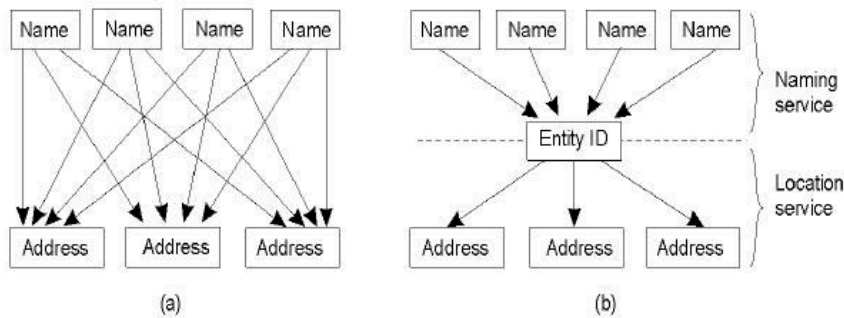
**Location Service:**

Ein Location Service bezeichnet ein Two-Level mapping zum Übersetzen von Namen in Adressen über sogenannte Identities. Es dient dazu, Services zu registrieren und aufzufinden. Beispielsweise bei P2P Netzen, müssen Rechner Services und Ressourcen ankündigen bzw. Ressourcen anderer Rechner auffinden können.

Abgrenzung zu Discovery Services: Bei Location Services ändern sich die Anforderungen oft und schnell. Zudem wird ein hoher Grad an Mobility erreicht, da ein Host nicht mehr über eine fixe Adresse (IP) angesprochen wird.

**Lösungsansätze: Broadcast und Multicast**

Wird nach einer Entität gesucht, sendet man per Broadcast Anfragen an alle Geräte im Netzwerk, ob sie diese Entität haben. Wenn ja, wird die Adresse des Eintrittspunkts zurückgeliefert, ansonsten die Anfrage einfach ignoriert. Bsp.: ARP. Dies sorgt jedoch für jede Menge unnötigen Traffic und die Hosts werden ständig mit Anfragen nach Entitäten belastigt. Eine Lösung wäre der Wechsel auf Multicast, durch die nur eine begrenzte Anzahl an Hosts die Anfrage erhält.



- a) Direct, single level mapping between names and addresses.
- b) Two-level mapping (e.g., using identities).  
**Location service** maps identifier to address.

### **Forwarding Pointers:**

Ein Ansatz, um Mobility zu erreichen. Wenn eine Entität ihren Ort ändert, lässt sie eine Referenz zurück, welche auf ihren neuen Ort zeigt.

- Vorteil: sehr einfach
- Nachteil: Lange Ketten, möglicherweise viele Zwischenschritte notwendig, eine falsche Referenz unterbricht gesamte Kommunikation (broken link), daher müssen die Ketten so kurz wie möglich gehalten werden.

### **Homebased Approaches:**

Ein Ansatz, um Mobility in großen Netzwerken zu erreichen. Der Home-Agent speichert dabei immer den aktuellen Standort einer Entität. Ändert das mobile Gerät (Entität) ihren Standort, registriert sie eine Care-Of-Address beim Home-Agent. Ein Client, der den aktuellen Standort nicht kennt, kann ihn also immer beim Home Agent erfragen und anschließend direkt kommunizieren.