

# Naming and Discovery

Buch: Kap. 5

Fragen:

1. Erläutern Sie die Begriffe "Name", "Identifier", "Address" sowie den Bezug zwischen diesen Begriffen in der Praxis.

Unter *Name* versteht man in verteilten Systemen einen String, der auf eine Entität (zB Drucker, Hosts, Prozesse, etc.) verweist. Um mit einer Entität arbeiten zu können, benötigt man einen Zugangspunkt, bzw. eine *Adresse*, die auf einen solchen verweist. Dabei kann eine Einheit durchaus mehrere Adressen aufweisen und diese bei Bedarf wechseln. Beispiele für Name und Adresse sind etwa eine Kombination aus Hostname und IP-Adresse bzw. Name und Telefonnummer. Vorteil dieses Systems ist, dass ein Server, der auf eine andere Maschine transferiert wird, auch danach noch unter dem gleichen, *ortsunabhängigen* Namen erreichbar ist. Unter *Identifier* versteht man einen Namen, der sich auf genau eine Einheit bezieht (bijektiv) und nur einmal vergeben werden kann. Praktischen Zweck hat diese Konvention in Verteilten Systemen v.a. Für das Message Routing, wobei Namen bzw. Identifier zu Adressen aufgelöst werden (*lookup*).

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

Name Spaces sind Strukturen, die eine Gruppe von Namen umfassen. Sie können als gerichtete Graphen durch Blattknoten (enthält Informationen über eine Entität, zB Adresse, oder deren Zustand, zB Datei) und Verzeichnisknoten (enthält eine Tabelle von ausgehenden Kanten sowie deren Knoten). Jene(r) Knoten, der keine eingehenden Kanten aufweist, heißt *Root*. Eine Sequenz aus Kanten lässt sich zu einem *absoluten* (wenn er mit dem Root-Knoten beginnt) oder *relativen Pfad* zusammensetzen.

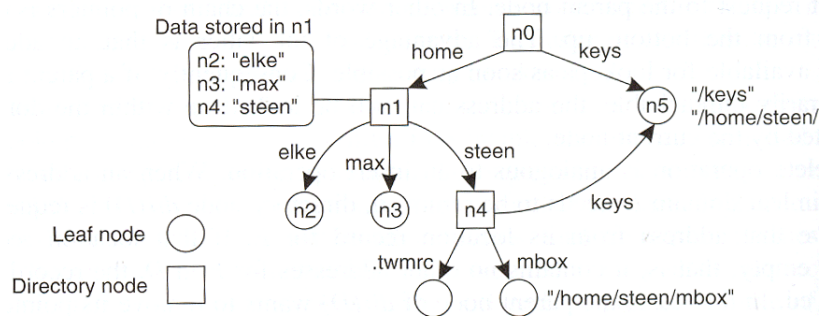


Figure 5-9. A general naming graph with a single root node.

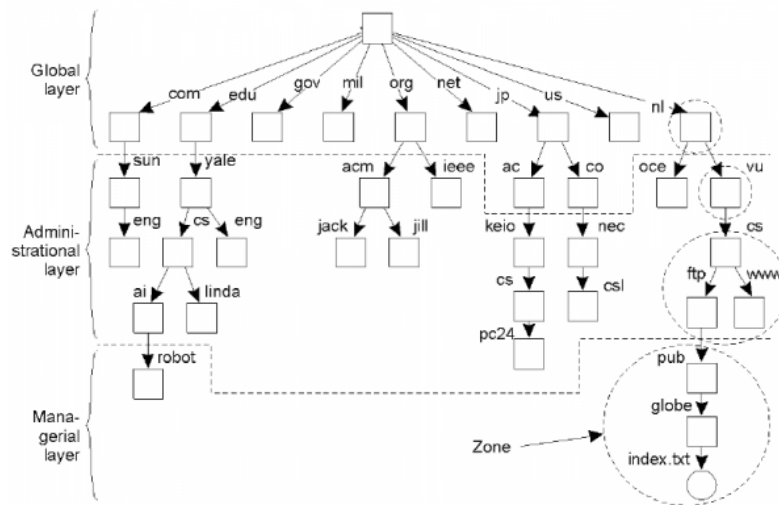
Mithilfe eines solchen Pfades kann man nun mittels *Name Resolution* (entspricht Lookup) auf Informationen des Nodes, auf den sich der Pfad bezieht, zugreifen. Zunächst muss man herausfinden, an welchem Knoten aus dem Name Space mit der Resolution begonnen werden soll (*Closure Mechanismus*). Um in UNIX einen Dateinamen wie */home/student/uni* auflösen zu können, muss dem Dateisystem der Root-Knoten „/“ bekannt sein. Der tatsächliche Offset des Root-Knotens ist im Superblock des logischen Laufwerks kodiert.

3. 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".

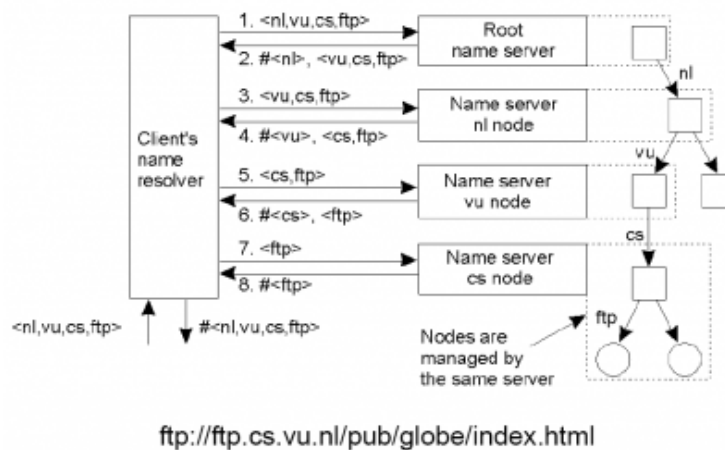
Name Spaces sind gewöhnlicherweise hierarchisch, in logischen Schichten, organisiert. Der *global layer* besteht aus den highest-level Nodes, also dem Root-Node und seinen unmittelbaren Nachfolgern. Die Verzeichnistabellen der Knoten in diesem Layer verändern sich kaum und bilden Organisation oder Gruppen von Organisationen ab (zB .com, .org, .at). Der *administrational layer* enthält seinerseits die Verzeichnisknoten einer einzigen Organisation, während der *managerial layer* ständig wechselnde Knoten enthält (zB Hosts in einem lokalen Netzwerk, freigegebene Dateien). Im Gegensatz zu den beiden oberen Schichten wird der Inhalt des managerial layer weniger von Administratoren, sondern von den einzelnen Usern verändert.

Zusätzlichen Performance-Gewinn und erhöhte Verfügbarkeit im GL kann durch Replikation der Server sowie client-seitiges Caching effizient erreicht werden, da nur wenige Änderungen zu erwarten sind. Erreichbarkeit von Name Servern im AL ist nur dann von Bedeutung, wenn sich der Client in der selben

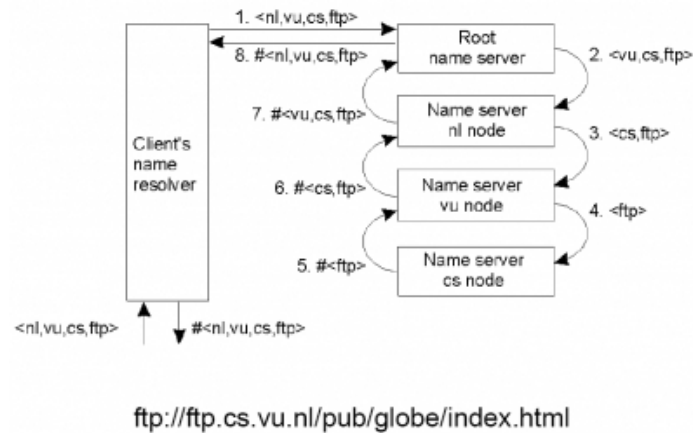
Organisation wie der Name Server befindet. In diesem Fall können Ressourcen aus der eigenen Organisation nicht gesucht werden. Ebenso wie im GL ist Caching ein probates Mittel, um die Performance zu steigern – jedoch sollten Updates schneller durchgeführt und Lookup-Resultate innerhalb weniger Millisekunden zurück gegeben werden was den Einsatz von Hochleistungsrechnern notwendig macht. Eine schnelle Reaktionszeit ist auch im ML unabdingbar, ein Caching ist hier nicht sinnvoll, da mit regelmäßigen Änderungen zu rechnen ist.



In einem Verteilten System hat jeder Client Zugang zu einem lokalen name resolver. Dieser kann die Namensauflösung in 2 verschiedenen Varianten durchführen: Im Zuge der *iterativen* Namensauflösung wird der gesamte Name dem Root-Name-Server übergeben, der nun, soweit ihm möglich, den Namen auflöst und das Resultat zurück gibt. Nun kontaktiert der Client diesen Name-Server, der wiederum einen Teil auflöst, usw. Abschließend löst der FTP Server den letzten Teil der ursprünglichen Pfad-Angabe auf (pub, globe) und gibt das HTML-File zurück. In der Praxis wird dies meist direkt vom Client übernommen und lediglich der Pfad *root:<nl, vu, cs, ftp>* übergeben.



Alternativ dazu gibt es das Modell der *rekursiven Namensauflösung*, bei welcher die Zwischenresultate nicht zum Client gesendet, sondern an den nächsten Name Server weitergeleitet werden. Die Ergebnisse des Lookup werden schließlich über alle beteiligten Server zurück zum Root Name Server und schließlich zum Client übertragen, welcher den FTP Server für den Transfer kontaktiert. Nachteilig wirkt sich bei der rekursiven Methode der höhere Rechenaufwand an den Name Servern aus, die jeweils den gesamten verbleibenden Pfad zu verarbeiten haben. Aus diesem Grund unterstützen Server im Global Layer lediglich iterative Namensauflösung. Als Vorteile sind zu nennen, dass die Kosten der Kommunikation wesentlich geringer ausfallen sowie ein effektiveres Caching, das bei der iterativen Suche auf den Client Name Resolver beschränkt ist.



4. 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?

Das DNS, das eines der größten verteilten Naming Services darstellt, hat die Aufgabe, IP Adressen von Hosts und Mail Servern zu finden (*mapping*). Mit dem DNS ist neben dem gewöhnlichen Lookup von Name zu IP auch eine umgekehrte Auflösung von IP-Adressen in Namen möglich (*reverse lookup*). In Analogie zum Telefonbuch entspricht dies einer Suche nach dem Namen eines Teilnehmers zu einer bekannten Rufnummer.

Der *DNS Name Space* ist hierarchisch als Wurzelbaum organisiert, dessen Labels aus Strings bestehen, die durch einen Punkt getrennt sind und zusammen einen Pfad ergeben. Eine Domain kann man sich als Unterbaum dieses Graphen vorstellen, ein Pfadname wird als *domain name* bezeichnet. Dabei besteht der Name Space lediglich aus einem Global und einem Administrational Layer – der Managerial Layer ist nicht Teil des DNS. Inhalte der Knoten setzen sich aus sogenannten *resource records* zusammen. Diese umfassen u.a. Informationen über die jeweilige Zone, Host IP-Adresse, Host-Name, etc.

Ein konkreter *Lookup* gestaltet sich wie folgt: Der Client stellt eine Anfrage für die IP Adresse von [www.tuwien.ac.at](http://www.tuwien.ac.at) an seinen lokalen DNS Server. Falls er einen entsprechenden Eintrag in seiner Datenbank findet, wird die IP an den Client (bzw. an dessen DNS) geschickt. Andernfalls teilt er den Domain-Namen immer kleinere Teile, die er wiederum aufzulösen versucht (tuwien.ac.at, .ac.at, .at), was einem rekursiven Lookup entspricht. Bringt dies keinen Erfolg, wird einer der Root Name Server kontaktiert. Da dem Root sämtliche Top-Level-Domains bekannt sind, schickt er dem lokalen Name Server am Client einen Verweis zu einem .at-Server. Im nächsten Schritt erfolgt die ursprüngliche Anfrage, diesmal aber an den .at-Server. Nachdem auch der .ac-Server kontaktiert wurde, wird im Idealfall ein DNS für die Domain tuwien.ac.at gefunden, an welchen der lokale DNS wiederum seine Anfrage sendet. Die Rückantwort sollte nun die IP Adresse von [www.tuwien.ac.at](http://www.tuwien.ac.at) enthalten und schlussendlich den Client erreichen.

Ein *reverse lookup* wird benötigt, um von einer IP auf einen Domainnamen zu schließen. Im ersten Schritt wird die IP-Adresse umgeformt, um sie von rechts nach links lesen zu können. In weiterer Folge wird die Domain *in-addr.arpa* hinzugefügt und für gegebene IP Adresse der Pointer Record (PTR) abgefragt, der auf den gesuchten Namen referenziert.

Um eine hohe Ausfallsicherheit zu gewährleisten, werden die Zonendaten nicht nur auf dem Primary Server gehalten, sondern auf Secondary Server repliziert. Bei Änderungen muss sichergestellt sein, dass alle Server den gleichen Datenbestand besitzen. Die Synchronisation zwischen den beteiligten Servern (*primary* → *secondary*) wird durch den *Zonentransfer* realisiert, der in der Regel auf Initiative der Secondary Server hin durchgeführt wird.

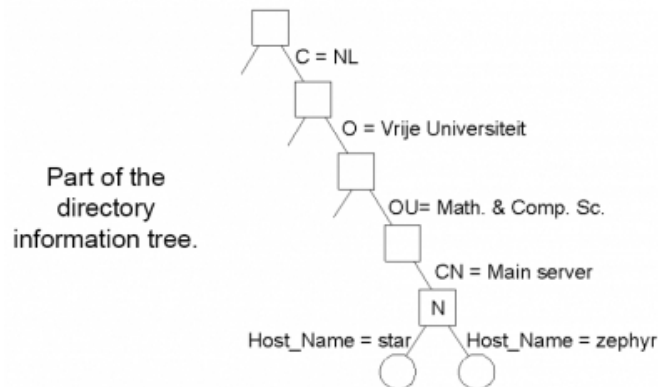
5. Was ist Directory Service bzw. "Attribute-based naming"? Beschreiben Sie den prinzipiellen Aufbau des X.500 Name Space sowie dessen LDAP Implementierung.

Bei *Attribute-based Naming* werden jeder Entität Paare (Attribut, Wert) zugeordnet wobei jedes Attribut etwas über die Entität aussagt. Ein User kann nun nach bestimmten Entitäten suchen, indem er Werte angibt, die die Attribute erfüllen müssen. Das Naming System gibt schließlich eine oder mehrere Entitäten zurück, die der Abfrage entsprechen.

Attributbasierte Services werden oft auch als *Directory Services* bezeichnet, wogegen *Naming Systems*

generell Systeme darstellen, die strukturiertes Naming unterstützen. Die Auswahl geeigneter Attribute ist jedoch schwierig, wenn spezifische Ergebnisse erwartet werden. Zudem ist die Suche rechnerisch aufwändig, da alle Deskriptoren untersucht werden müssen, vor allem bei verteilten Datenbanken.

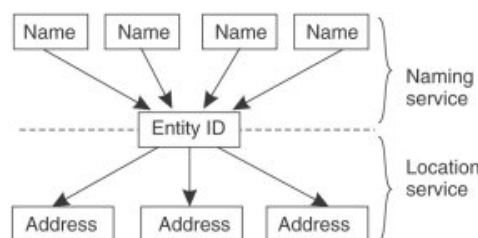
Bei x.500 handelt es sich um einen OSI-Standard, der den Entwurf eines globalen Verzeichnisdienstes beschreibt, der auf strukturiertem und attributbasiertem Naming basiert. Der Aufbau von x.500 besteht aus Objekten, über die Informationen (Attributwerte) im Verzeichnis abgelegt werden, die notwendig oder optional sein können. Alle Objekte sind hierarchisch im *Directory Information Tree (DIT)* miteinander verbunden, wobei jede Ebene von Objekten für die Organisation der darunterliegenden Objekte verantwortlich ist. Jeder Eintrag hat einen eindeutigen Distinguished Name, der eine Kombination aus seinem eigenen RDN (Relative Distinguished Name) und den RDN all seiner übergeordneten Einträgen ist. Die Gesamtheit aller Informationen im Verzeichnis wird als Directory Information Base (DIB) bezeichnet.



Das *lightweight directory access protocol (LDAP)* ist eine effizientere Umsetzung von x.500, die auf dem Client/Server Prinzip basiert und direkt auf TCP aufsetzt. Es gilt als de Facto Standard für internet-basierte Verzeichnisdienste. Ein LDAP Verzeichnisdienst besteht aus verschiedenen Verzeichniseinträgen, die jeweils von mehreren (Attribut, Wert) Paaren beschrieben werden. Das Verzeichnis kann beispielsweise ein E-Mail-Adressbuch enthalten und die gewünschte Adresse zu einer gegebenen Person liefern.

**6. 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?**

Ein Location Service bezeichnet ein Two-Level mapping zum Übersetzen von Namen in Adressen über sogenannte Identities. Es hat in Ad-hoc Netzwerken (zB P2P), deren Ressourcen und Services sich dynamisch ändern, die Aufgabe, Services zu registrieren bzw. aufzufinden (*Discovery*). Dadurch erreicht man einen hohen Grad an *Mobility*, da ein Host nicht mehr über eine fixe IP-Adresse angesprochen wird. Grundlegender Unterschied zu Name- und Directory Services ist, dass sich ein Location Service häufig und schnell anpassen muss.



Ein Ansatz, um Mobility zu erreichen, ist die Verwendung von *Forwarding Pointers*. Wechselt eine Entität ihren Ort, lässt sie eine Referenz (*symbolic link*) zurück, welche auf die neue Position zeigt. Dieses System ist sehr einfach zu implementieren, kann jedoch lange Ketten aufweisen, die viele Zwischenschritte notwendig machen. Kommt es bei einer dieser Schritte zu einer falschen Referenz (*broken link*), wird die gesamte Kommunikation unterbrochen. Daher ist es nötig, die Ketten so kurz wie möglich zu halten. *Homebased Approaches* stellen Mobility in großen Netzwerken sicher. Die Home Location speichert dabei immer den aktuellen Standort einer Entität. Ändert die Entität ihre Position, registriert sie eine Care-Of-Adresse beim Home-Agent. Ein Client, der den aktuellen Standort nicht kennt, schickt eine Nachricht zur

bekannten Adresse des mobilen Client. Der dortige Home-Agent kennt dessen aktuelle Adresse und „tunnelt“ die Pakete an den Empfänger weiter. Gleichzeitig informiert er den Sender über die tatsächliche Adresse des mobilen Clients – nachfolgende Nachrichten werden somit direkt zwischen den beiden Clients ausgetauscht. Nachteil eines solchen Systems ist die hohe Latenz in der Kommunikation.

