



# Verteilte Systeme

## 4. Namensdienste

Sommersemester 2011

Institut für Betriebssysteme und  
Rechnerverbund

TU Braunschweig

Dr. Christian Werner

– Bundesamt für Strahlenschutz –

INSTITUT FÜR **B**ETRIEBSSYSTEME  
UND **R**ECHNERVERBUND

Prof. Dr.-Ing. L. Wolf | Prof. Dr. S. Fekete



- Namen, Identifier und Adressen
- Verzeichnisdienste
- Namensräume
- Namensauflösung
- Implementierung von Namensräumen und Namensauflösungen
- Beispiel: Domain Name System (DNS)

- Namen in verteilten Systemen
  - Sind eine **Folge von Bits oder Zeichen**
  - Werden verwendet, um eine **Instanz menschenlesbar** zu referenzieren
- Instanzen:
  - Hosts, Drucker, Platten, Dateien, ...
  - Prozesse, Benutzer, Mailboxen, Web-Seiten, ...
- Um auf identifizierte Instanzen zugreifen zu können, benötigt man einen **Zugangspunkt**, die **Adresse**.

## Spezielle Namen: Identifier

- Identifier haben spezielle Eigenschaften:
  - Ein Identifier bezieht sich auf höchstens eine Instanz.
  - Jede Instanz wird von höchstens einem Identifier referenziert.
  - Ein Identifier bezieht sich immer auf dieselbe Instanz.
- Zweck:
  - Eindeutige Identifikation von Instanzen.
- Adressen und Identifier sind meist nicht „lesbar“.

- Verzeichnisdienste werden generell genutzt, um einem Namen Attribut-/Wert-Paare zuzuordnen.
- Beispiele:
  - Telefonnummern- und Adressenverzeichnis in einer Firma
  - Namen und Verfügbarkeit von Ressourcen (Drucker, Platten, Modems)
  - Verfügbare Anwendungsdienste
- Dabei sind nicht nur Abfragen von Name zu Attribut, sondern auch umgekehrt möglich („Yellow Pages“).
- Die Verknüpfung zwischen einem Namen und seinen Attributen wird als *Bindung* („binding“) bezeichnet.

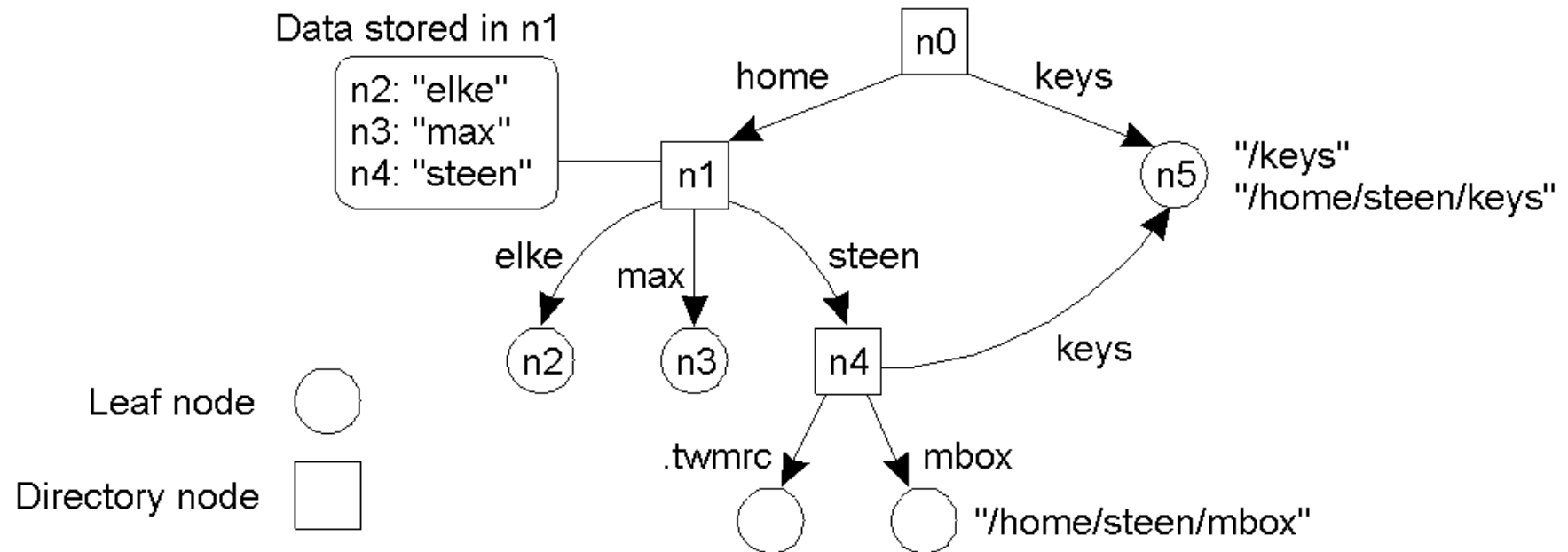
- Namensräume werden zur Ordnung von Namen in VS verwendet.
- Sie enthalten alle gültigen Namen und geben Syntaxregeln zu deren Definition vor.
- Ein Namensraum lässt sich als markierter gerichteter Graph mit zwei Arten von Knoten darstellen:
  - Blattknoten zur Repräsentation von benannten Instanzen
  - Verzeichnisknoten speichert Tabellen mit Verweisen auf andere Knoten
- Kanten werden ebenfalls markiert
- Typische Namensräume haben genau einen Wurzelknoten.

- Pfade werden typischerweise durch die Folge der Kantenlabels beschrieben:

$N: \langle \text{label-1}, \text{label-2}, \dots, \text{label-n} \rangle$

wobei  $N$  der erste Knoten des Pfades ist. (Root wird oft durch „/“ repräsentiert, ebenso die Label-Abgrenzung).

- Wenn  $N = \text{Root}$  ist, dann ist der Pfad ein **absoluter Pfad**, sonst ein **relativer**.





- Ziel: finde heraus, ob ein gegebener Name definiert (=gebunden) ist und liefere die damit verbundenen Informationen zurück
- Zur Auflösung wandert man durch den Graph, bis man auf den entsprechenden Blattknoten stößt oder nicht weiterkommt (Name existiert dann nicht)
- Der Inhalt des Blattknotens wird dann zurückgegeben.
- Es ist die Aufgabe von **Verzeichnisdiensten**, solche Auflösungen vorzunehmen.

# Anforderungen an Verzeichnisdienste

- Die Möglichkeit, eine im wesentlichen unbeschränkte Zahl von Namen zu verwalten
- Lange Lebensdauer des Dienstes (Investitionssicherung)
- Hohe Verfügbarkeit
- Isolierung von Fehlern (um nicht den ganzen Service zu beeinflussen)
- Tolerierung von Misstrauen (in einem großen System sind nicht alle Komponenten gleich vertrauenswürdig)

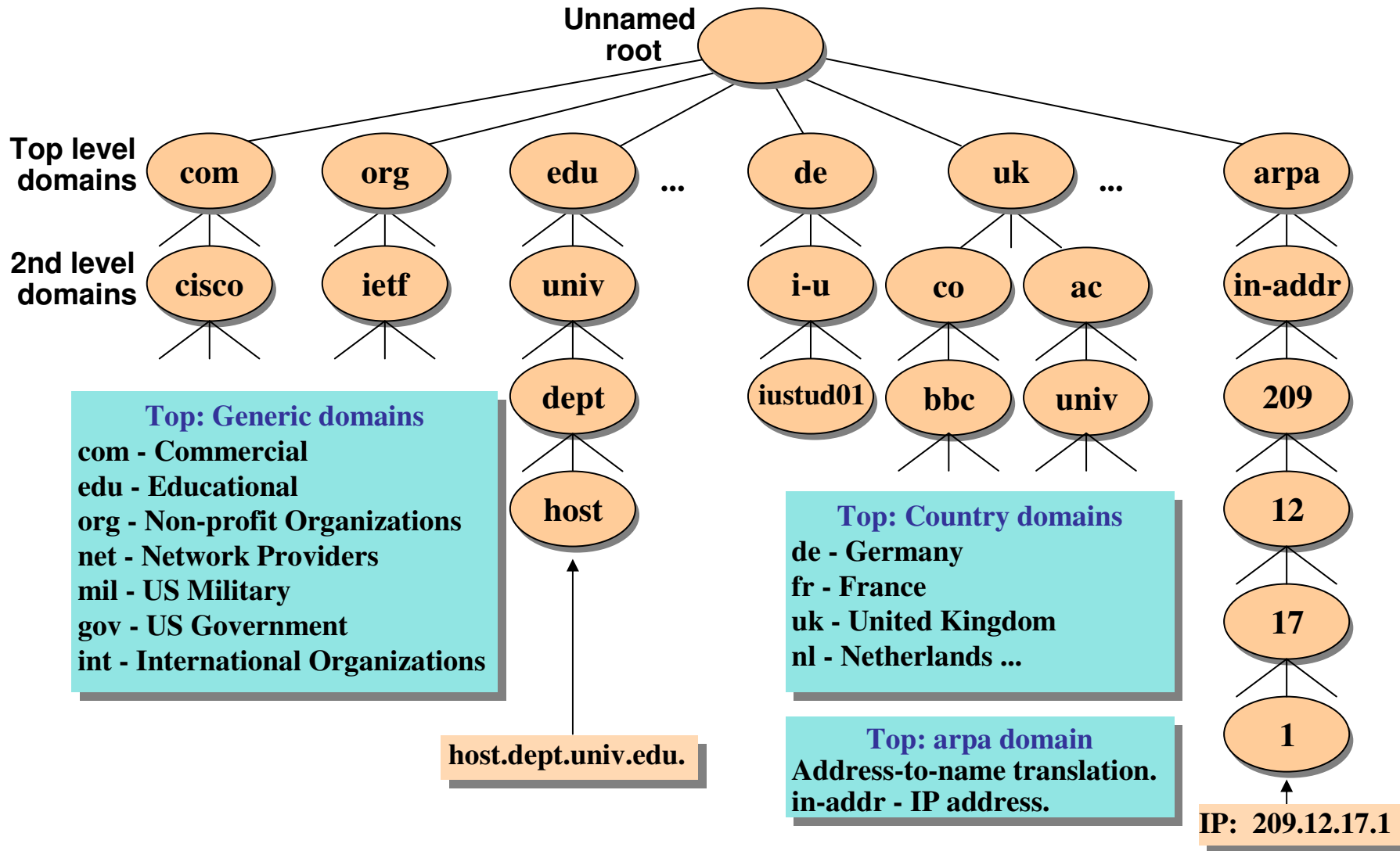
- Hauptkomponente: Datenbank von Bindungen
- Hauptfunktion: Namensauflösung
- Ansonsten:
  - Erzeugung von Bindungen
  - Löschen von Bindungen
  - Auflistung gebundener Namen
  - Suche von Namen anhand von Attributen

- Wird üblicherweise in großen Namensräumen dezentralisiert durchgeführt
- Für jede Naming Domain (ein Ausschnitt des Namensraums) gibt es eine *administrative authority*, die
  - die Kontrolle darüber hat, welche Namen in der Domain gebunden werden können,
  - diese Kontrolle delegieren kann (für *sub-domains*)
  - die Datenbank der Bindings administriert.

- Das Telefonbuch
- Die „Gelben Seiten“
- **Internet DNS**
- Jini lookup service
- OSI X.500
- CORBA Name Service
- **Java JNDI und LDAP als Integratoren**
- UDDI
- ...

- Aufgabe von DNS
  - Übersetze Namen in Adressen und lasse Anwendungen so Netzdienste nutzen: `www.cs.tu-bs.de` ↔ `134.169.34.18`.
- Internet Name Service
  - DNS: Domain Name System. Frühe 80er Jahre.
  - Grundlegende Definition: RFC 1034 und 1035, 1987
  - Zahllose weitere RFCs für die Nutzung von DNS, Updates etc.
  - Hierarchisches Namensschema
  - Verteilte Namensdatenbank
  - DNS-Protokoll: query-response protocol.

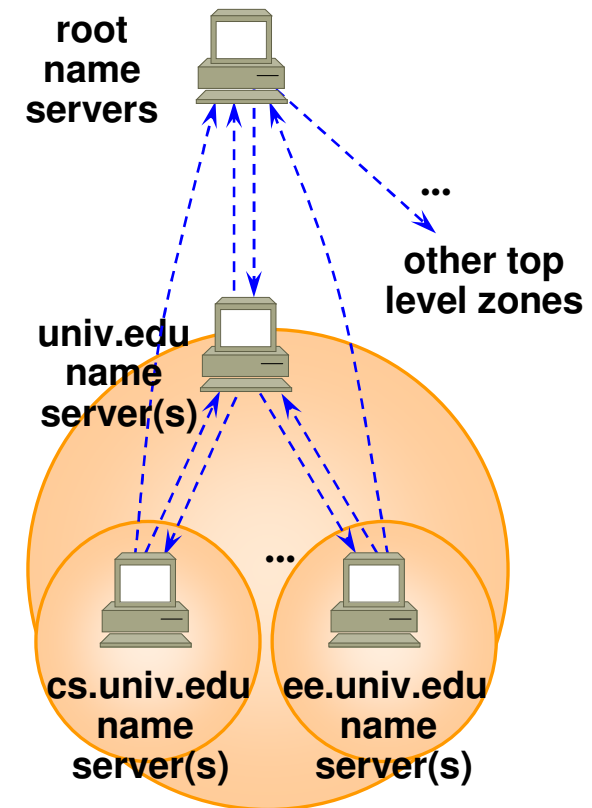
# Namenshierarchie



- „Zone of authority“
  - Eine Zweig des Namensraums, der getrennt verwaltet wird
  - Der Namensraum ist rekursiv in kleinere Zonen aufgeteilt.

## Delegation der Verantwortung

Der Administrator einer Zone verwaltet den/die Name Server (Primär-, Sekundär-)  
 Jeder Name Server verwaltet die Namensinformation für seine Zone und kennt die Name Server der Unterzonen.

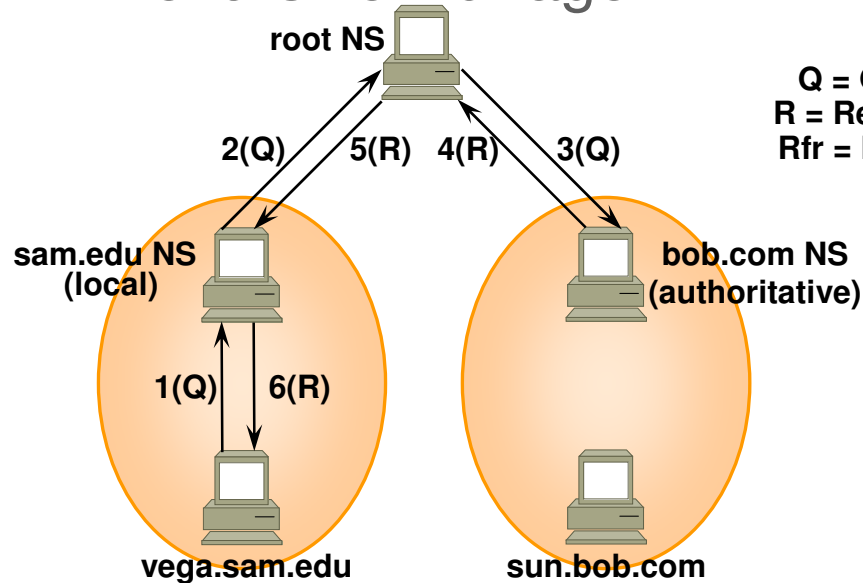


Eine Gruppe von „root name servers“ stellen Name-Server-Adressen für die Top-Level-Domains zur Verfügung. Alle anderen Server kennen die Adresse der Root-Server und einiger Name Server der oberen Ebenen.

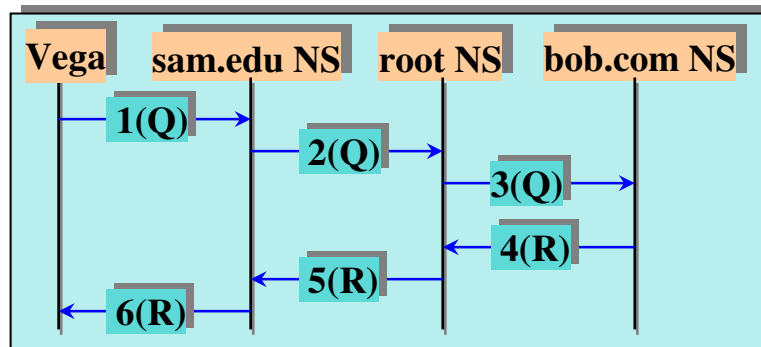


# DNS-Abfragen

## Rekursive Abfragen

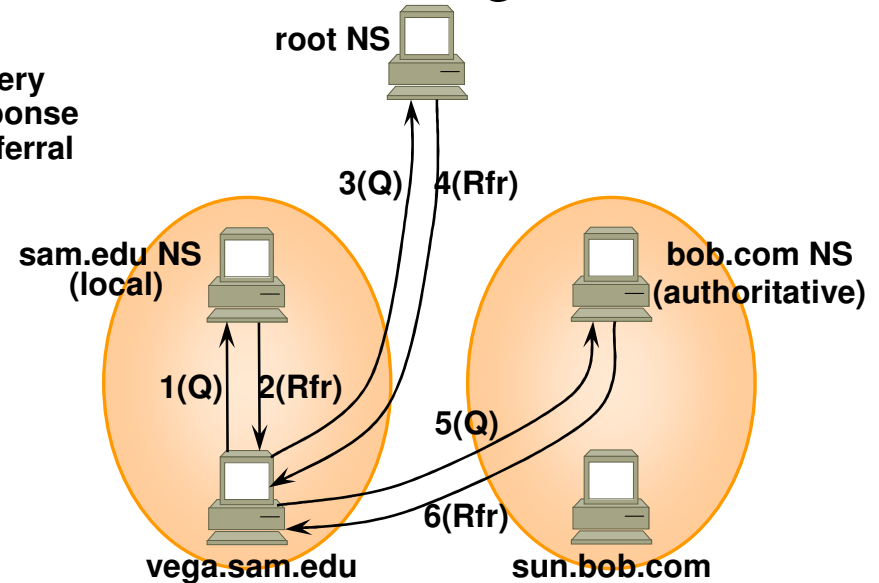


Query: "Address of sun.bob.com"

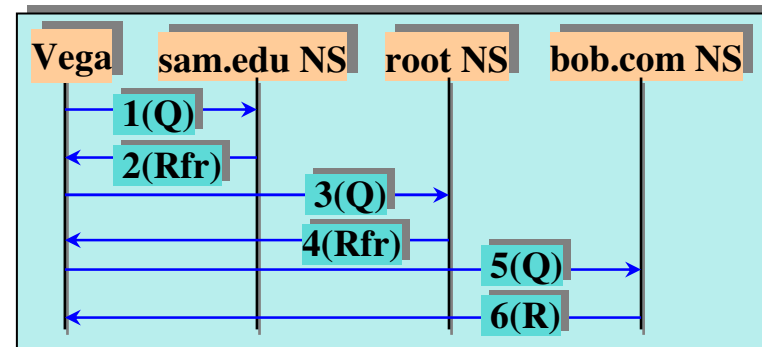


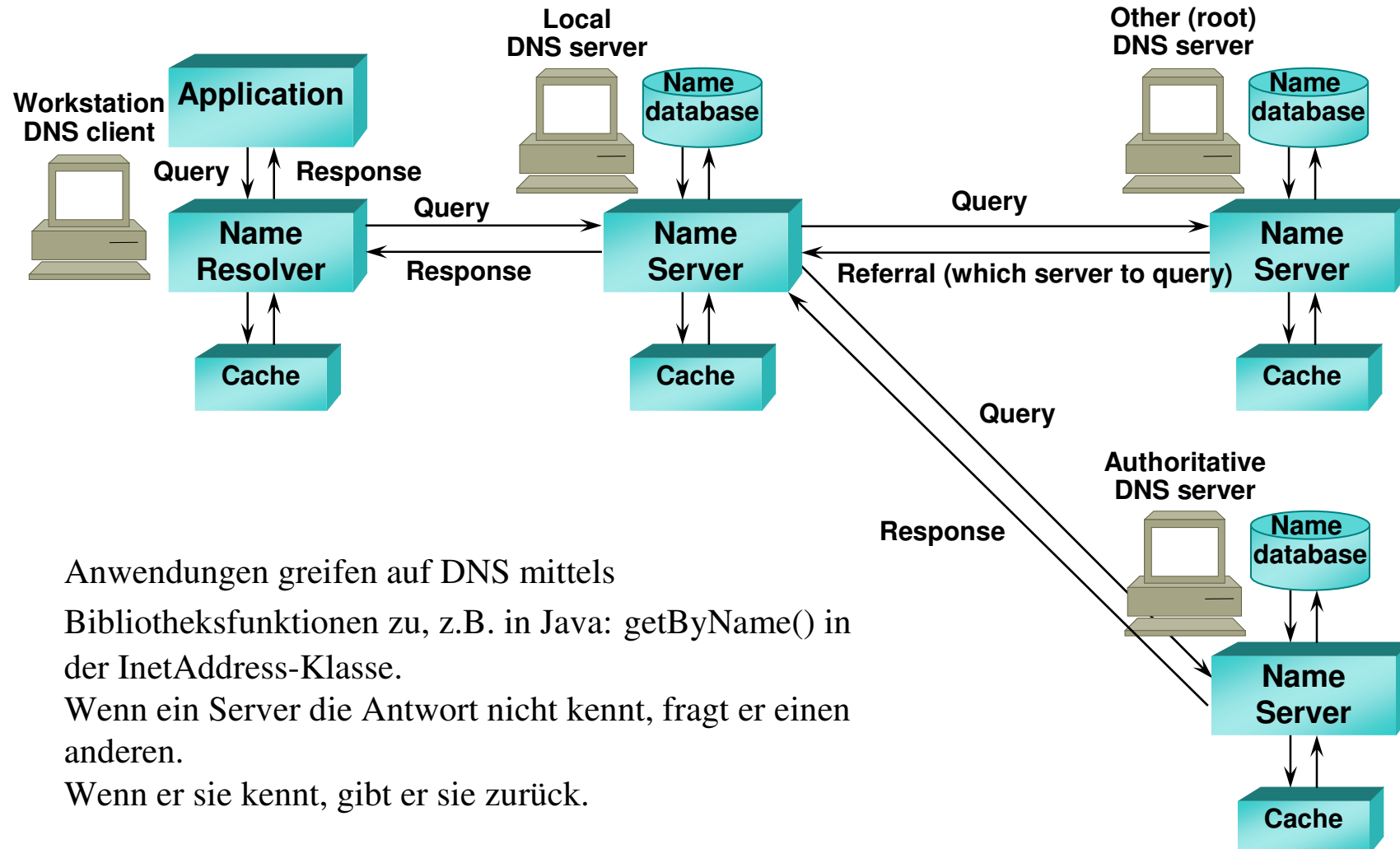
## Iterative Abfragen

Q = Query  
R = Response  
Rfr = Referral



Query: "Address of sun.bob.com"





Anwendungen greifen auf DNS mittels

Bibliotheksfunktionen zu, z.B. in Java: `getByName()` in der `InetAddress`-Klasse.

Wenn ein Server die Antwort nicht kennt, fragt er einen anderen.

Wenn er sie kennt, gibt er sie zurück.

# DNS Datensatztypen

Type of record	Associated entity	Description
SOA	Zone	Holds information on the represented zone
A	Host	Contains an IP address of the host this node represents
MX	Domain	Refers to a mail server to handle mail addressed to this node
SRV	Domain	Refers to a server handling a specific service
NS	Zone	Refers to a name server that implements the represented zone
CNAME	Node	Symbolic link with the primary name of the represented node
PTR	Host	Contains the canonical name of a host
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any entity-specific information considered useful

# Beispiel: DNS am IBR in Braunschweig

```

; Start of Authority:
;
@           IN           SOA      ibr.cs.tu-bs.de. root.ibr.cs.tu-bs.de. (
                                2001103001      ; Serial YYYYMMDDff
                                10800          ; Refresh          3H
                                3600           ; Retry            1H
                                604800        ; Expire 1W
                                86400 )        ; Minimum          1D
;
; Nameserver:
;
@           IN           NS       agitator.ibr.cs.tu-bs.de.

; Mail Exchanger fuer ibr.cs.tu-bs.de:
;
@           IN           MX       10 mumm.ibr.cs.tu-bs.de.

becks      IN           A         134.169.34.65
           IN           MX       10 mumm.ibr.cs.tu-bs.de.
koepi      IN           A         134.169.34.66
           IN           MX       10 mumm.ibr.cs.tu-bs.de.
tuborg     IN           A         134.169.34.67
           IN           MX       10 mumm.ibr.cs.tu-bs.de.

```

# Discovery Services

- Ein Verzeichnisdienst, der Service-Registrierungen in Spontaneous-Networking-Umgebungen vornimmt.
- Discovery-Dienste besitzen Möglichkeiten zur automatischen Registrierung und De-Registrierung von Diensten. Dienstangebote sind häufig schwankend.
- Es müssen keine menschlichen Benutzer eingebunden sein!
- Beispiele
  - Jini Lookup Service
  - IETF Service Location Protocol (RFC 2608)



- Lightweight Directory Access Protocol
- Spezifiziert im [RFC 2251](#)
- Definiert ein vereinfachtes Internet-fähiges Protokoll zum Zugriff auf Verzeichnisdienste, die auf der Basis von OSI X.500 arbeiten
- Namensbeschreibung hierarchisch, in etwas anderer Form, Beispiel:  
`cn=Christian Werner, ou=cs, o=tu-bs, c=de`
- LDAP gestattet Authentisierung; wichtig für Updates von Bindungen

- Java Naming and Directory Service
- Javas Schnittstelle für alle Arten von Namens- und Verzeichnisdiensten
- Mit JNDI können
  - Java-Objekt-basierte Verzeichnisdienste implementiert werden
  - vorhandene Dienste in Java-Anwendungen eingebunden werden
- Beispiel: LDAP, DNS, NIS
- s. <http://java.sun.com/products/jndi>



Das API findet sich unter [javax.naming.\\*](#).  
Beispiel: die Klasse Context:

```
public interface Context {
    public Object lookup(Name name) throws NamingException;
    public void bind(Name name, Object obj) throws NamingException;
    public void rebind(Name name, Object obj) throws
        NamingException;
    public void unbind(Name name) throws NamingException;
    public void rename(Name old, Name new) throws NamingException;
    public NamingEnumeration listBindings(Name name) throws
        NamingException;
    ...
    public Context createSubcontext(Name name) throws
        NamingException;
    public void destroySubcontext(Name name) throws NamingException;
    ...
};
```

- Namen werden benötigt, um Instanzen in einem VS in eindeutiger Weise zu identifizieren und Informationen damit zu verbinden.
- Wichtiger Teilaspekt: Namensauflösung
- DNS als zentraler Verzeichnisdienst des Internet
- Für verschiedene Middleware-Lösungen gibt es unterschiedliche Namensdienste

# Diskussion

