Verteilte Systeme

R. Kaiser, R. Kröger, O. Hahm

(HTTP: http://www.cs.hs-rm.de/~kaiser
E-Mail: eobert.kaiser@hs-rm.de)
Kai Beckmann
Sebastian Flothow
Alexander Schönborn

Sommersemester 2021

5. Namens- und Verzeichnisdienste

http://www.sa-23e.com/wordpress/wp-content/uploads/2013/05/telefonbuch.jpeg

Inhalt

Hochschule RheinMain

Einführung

Hochschule RheinMain

- Numerische Identifier und Adressen (z. B. IP-Adressen) sind für den Menschen schwer zu merken.
- Namen dienen zur Abstraktion von konkreten Diensten oder Objekten
- Aber wie komme ich von einem Namen zu einer Adresse?
- In Kapitel 3 (RPC) wurde besprochen:
  - Naming und Locating/Addressing von Diensten durch Binder
- Hier Verallgemeinerung:
  - Namensdienste
  - Verzeichnisdienste
- Auch in anderen Kontexten haben wir es mit Namensauflösung zu tun (z. B. Variablenname $\rightarrow$ Speicheradresse)

Namen

Hochschule RheinMain

- 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/rein*: ein Name ist nur ein Bit-Muster und enthält keinerlei weitere Information
  - *impure/unrein*: 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 (Universally Unique Identifier) sind unique
    - 128 Bit-Zahl
    - spezifiziert in RFC 4122 und ITU-T Rec. X.667 | ISO/IEC 9834-8:2005
    - unterschiedliche Versionen
    - generiert durch Tool `uuidgen`
    - in der Microsoft-Welt auch *Global Unique Identifier (GUID)*
    - Beispiel: `123e4567-e89b-12d3-a456-426614174000`
- pure
  - UUIDs als Namen von DCOM-Objekten oder Klassen sind pure
- impure
  - DNS-Namen implizieren oft zusätzliche Information
    - z. B. `mail.hs-rm.de`

URI, URL und URN

Hochschule RheinMain

- Ein *Uniform Resource Identifier* (URI) identifiziert eine konkrete Ressource,
  z. B. `ISBN 978-1543057386`.
- Ein *Uniform Resource Locator* (URL) gibt darüber hinaus an wie auf die Ressource zugegriffen werden kann,
  z. B.
  `https://www.libra-buchhandlung.de/shop/item/9780131217867`.
- Ein *Uniform Resource Name* (URN) identifiziert eine konkrete Ressource persistent und ortsunabhängig,
  z. B. `urn:isbn:978-1543057386`.

Namensräume (1)

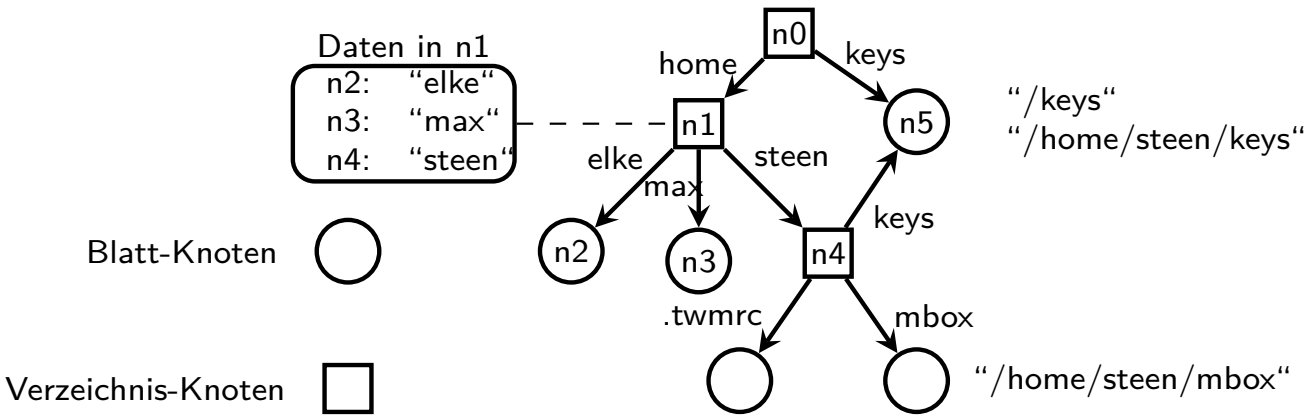
Hochschule RheinMain

- Namen haben nur in einem bestimmten Kontext eine Bedeutung.
- Namen werden in Namensräumen strukturiert.
- Namensräume geben die Syntaxregeln für die Namen vor.
- Beispiele: Namespaces in C++, DNS, ISBN...

Namensräume (2)

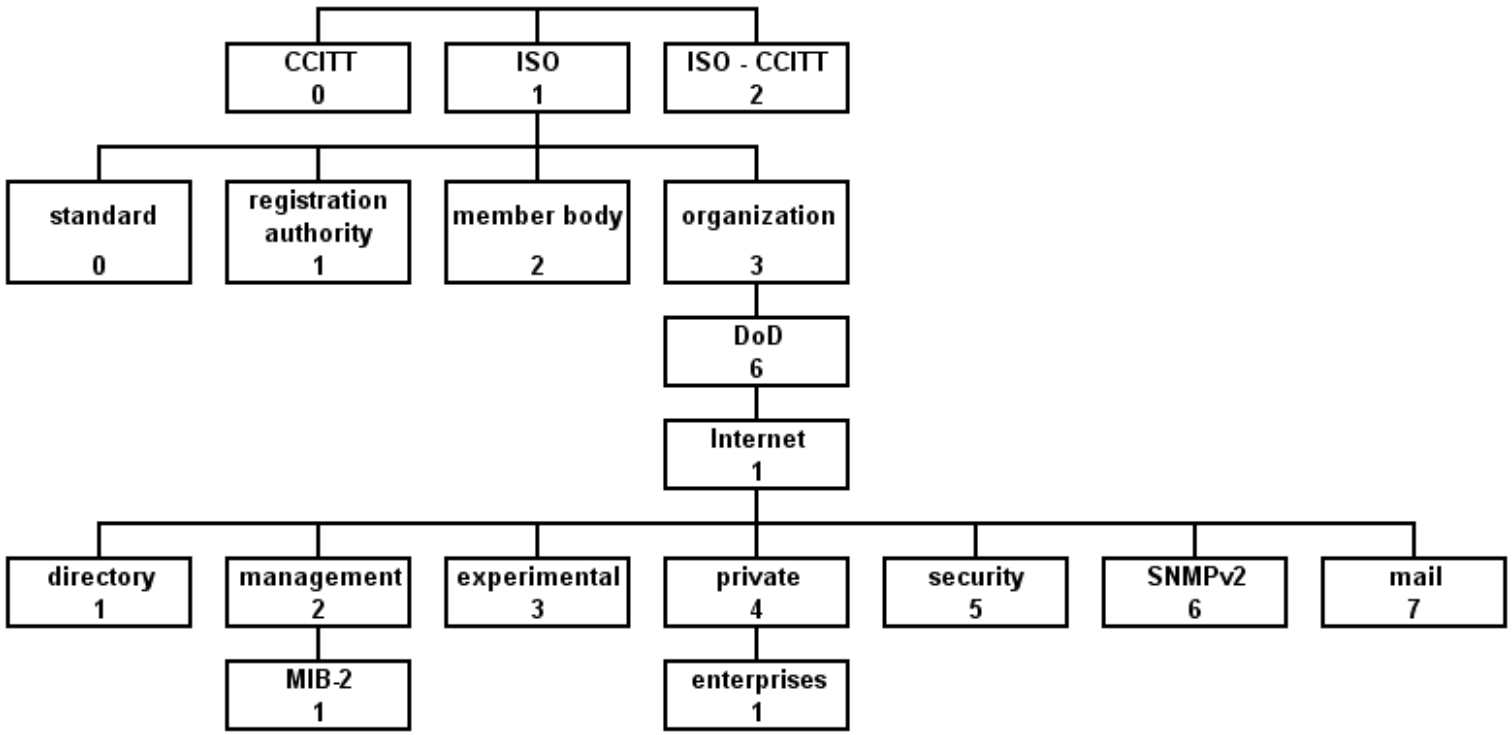
Hochschule RheinMain

- Flache Namensräume (heute eher selten, z.B. Unix UIDs)
- Hierarchische Namensräume werden üblicherweise als gerichtete Graphen mit Labels organisiert.
- In solchen Namensräumen liefert dann der Präfix den Kontext.
  - Verzeichnisknoten und Blätter
  - Absolute und relative Pfade
  - Globale und lokale Namen
- Beispiel: Unix-Dateinamensraum



Beispiel: MIB-2 - Namensraum

- MIB: Management Information Base

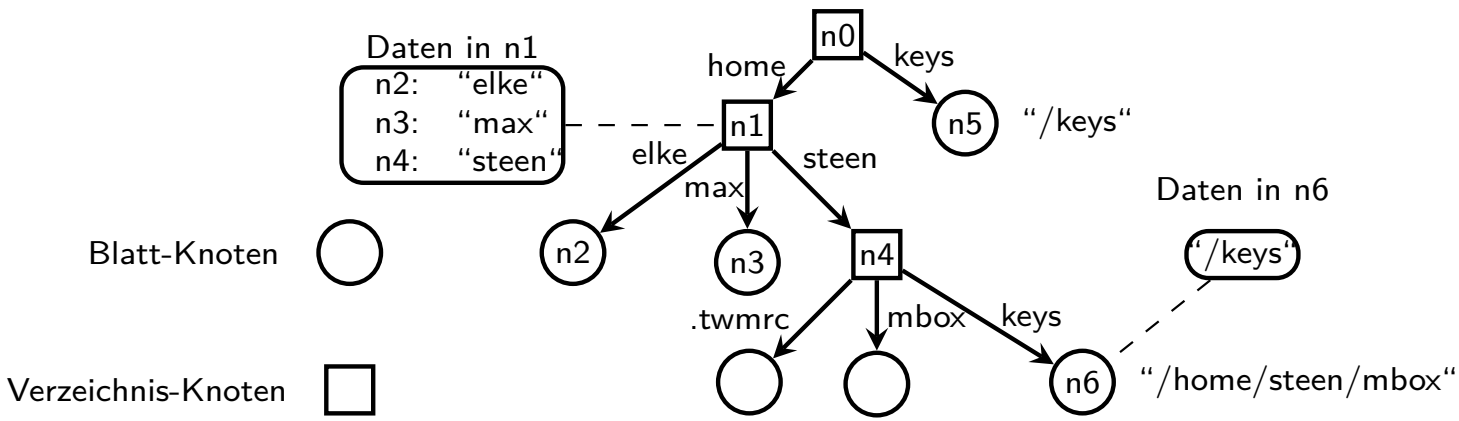


https://upload.wikimedia.org/wikipedia/commons/1/1c/SNMP.MIB-Tree.PNG

Links in einem Namensraum

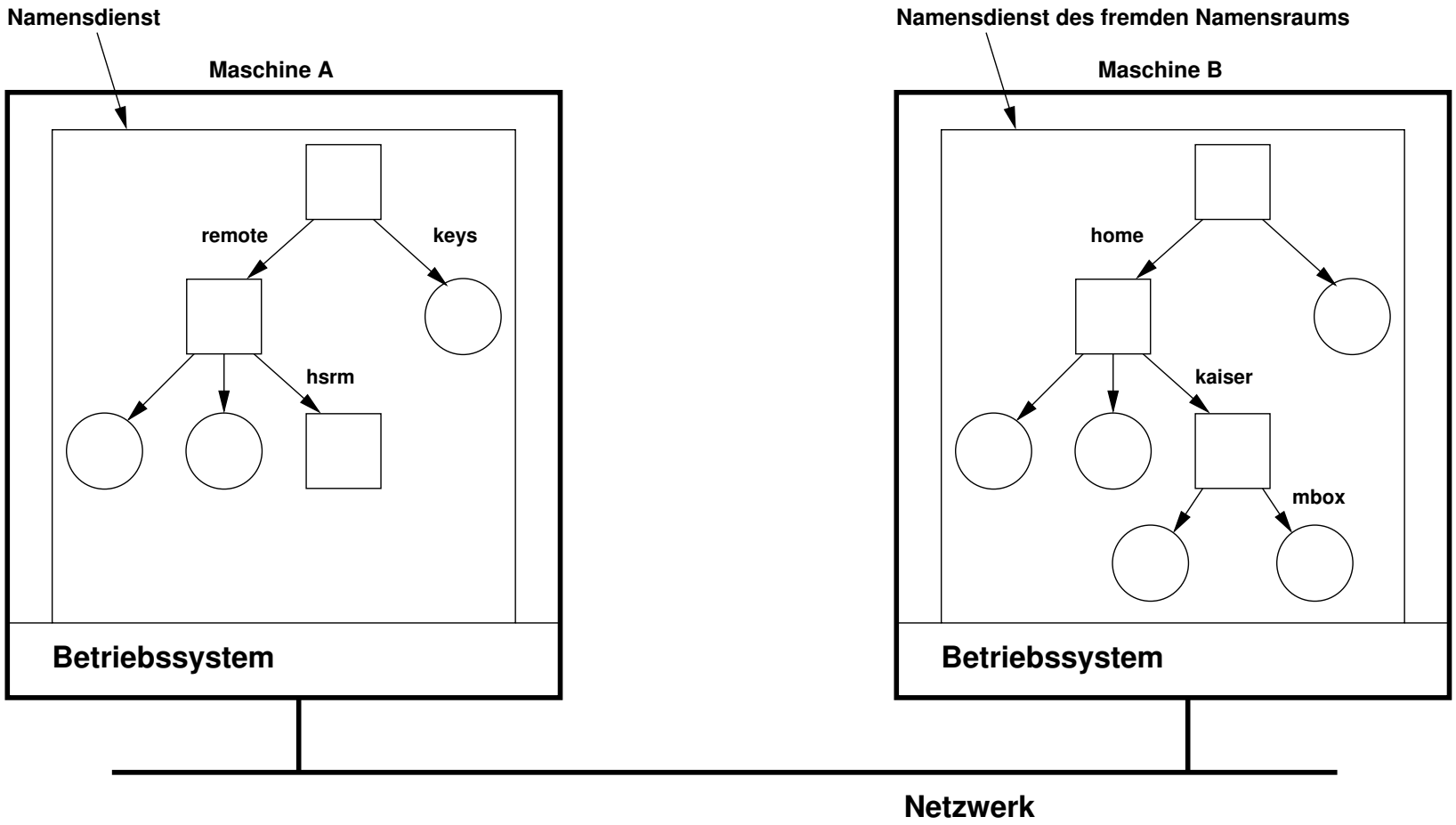
Hochschule RheinMain

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



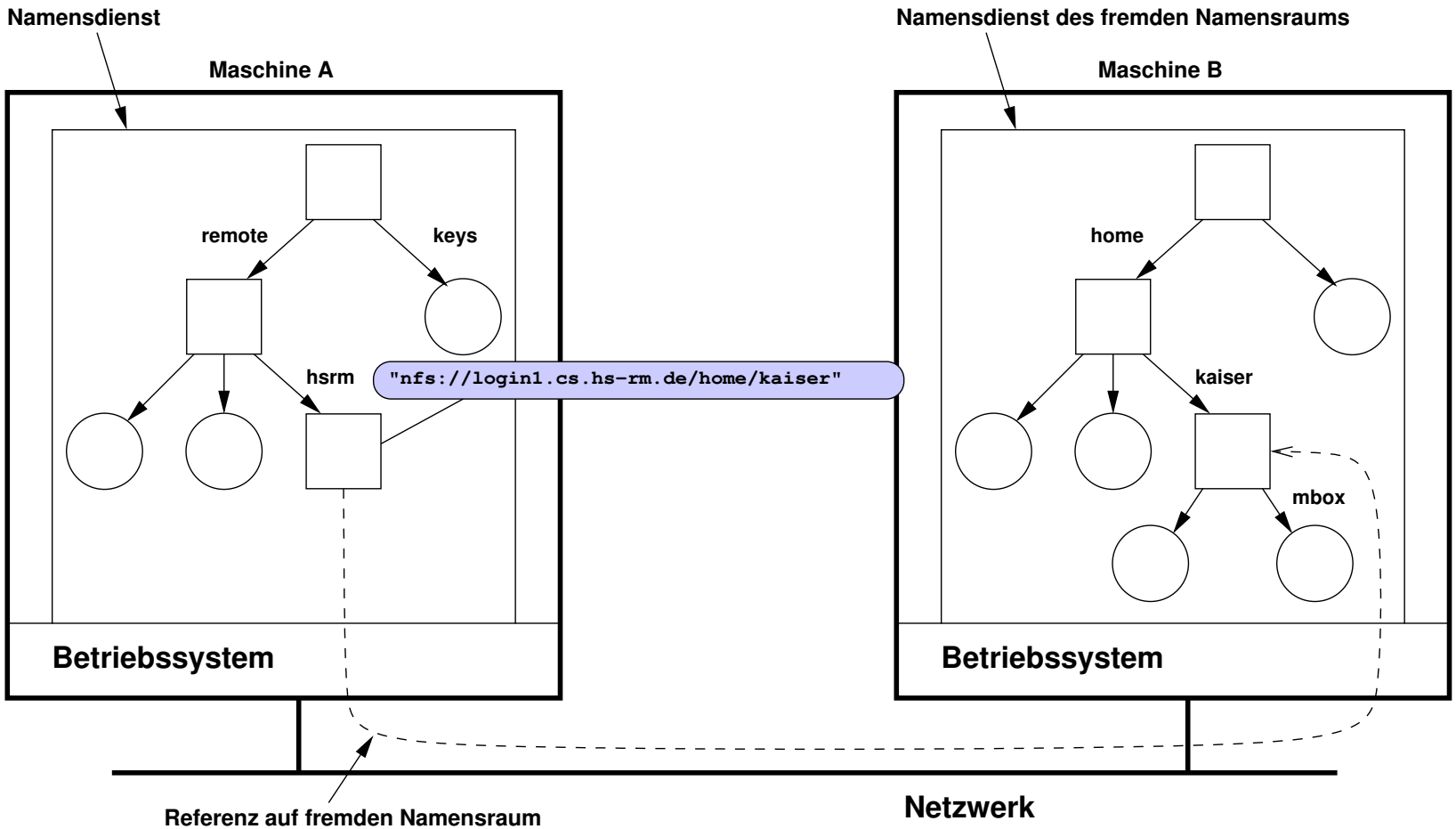
Links in einen anderen Namensraum

- Mounting



Links in einen anderen Namensraum

- Mounting



Organisation von großen Namensräumen

Hochschule RheinMain

- 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)

Hochschule RheinMain

Vgl. LV Rechnernetze

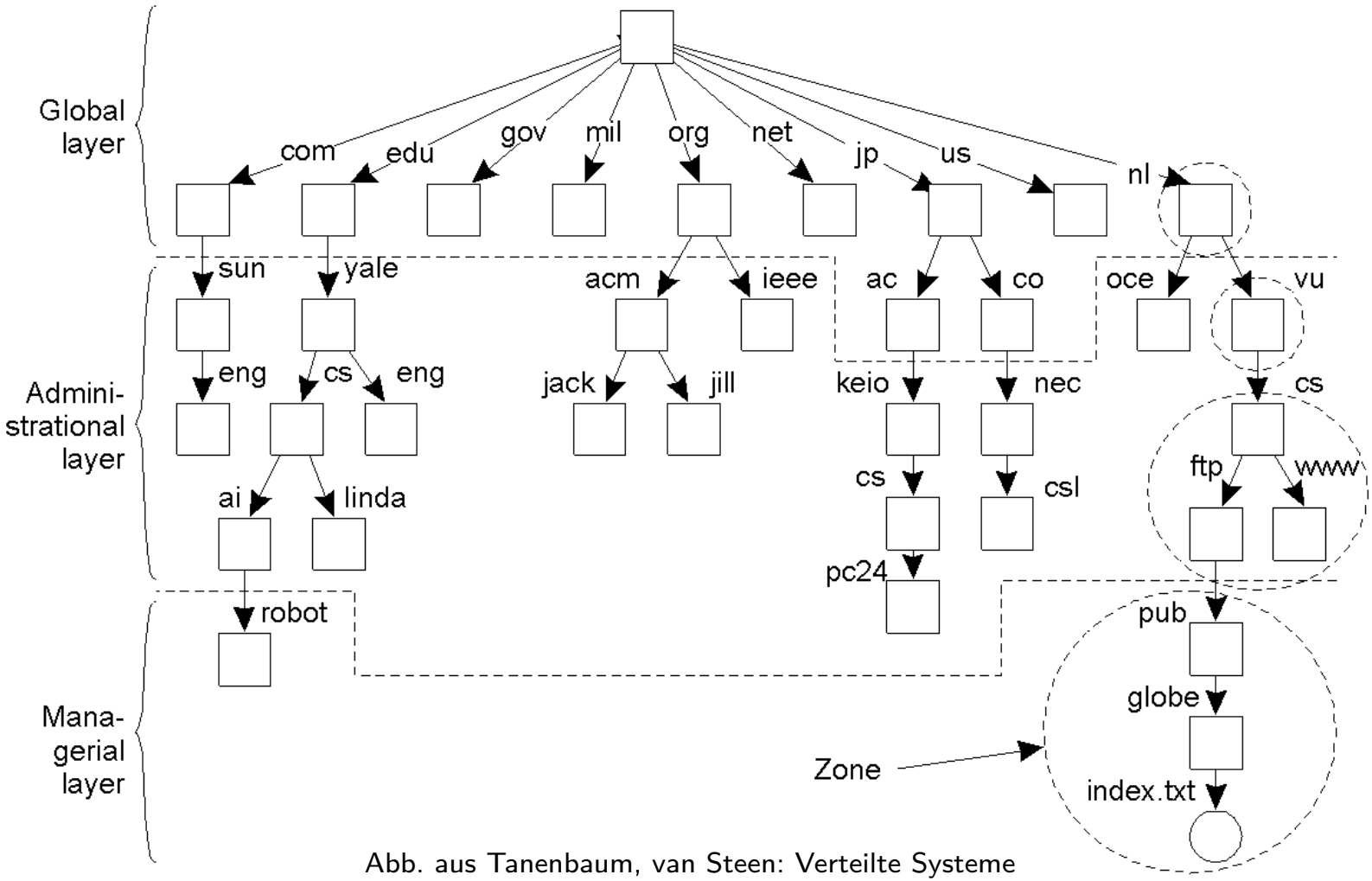


Abb. aus Tanenbaum, van Steen: Verteilte Systeme

Adressen

Hochschule RheinMain

- 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

Namensdienste

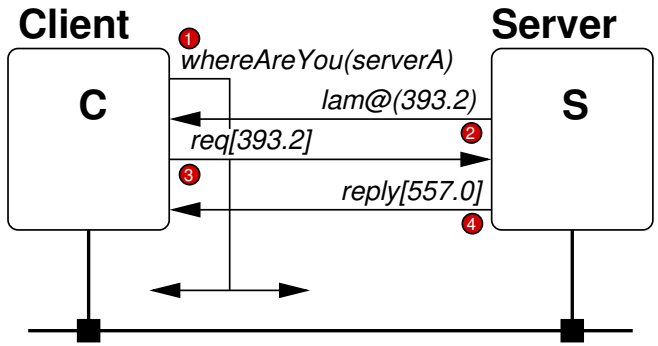
Hochschule RheinMain

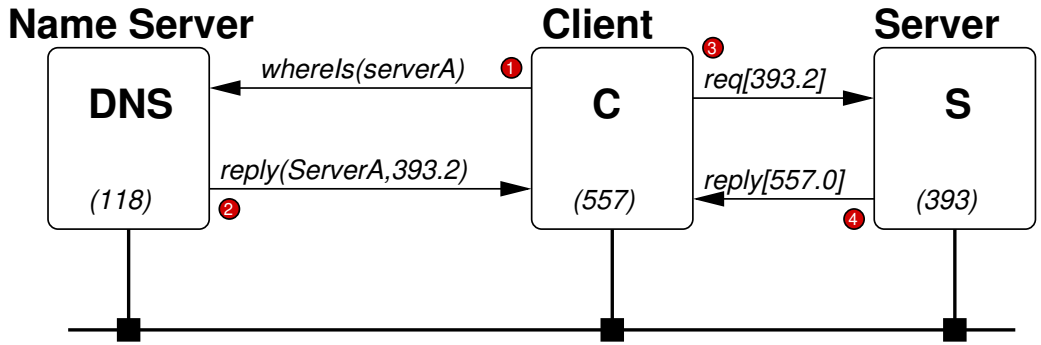
- Namensauflösung:
  Vorgang, um von gegebenem Namen eines Objekts zu seinem Adress-Attribut zu kommen
- Namensdienst (Name Server):
  Realisierung der Namensauflösung für anfragende Clients
  - Im Falle von RPC-Systemen auch Binder genannt
  - Typische Operationen:
    - Register / Bind
    - Deregister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

Arten der Namensauflösung

Hochschule RheinMain

- 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 Adresse
  - Beispiel: DNS

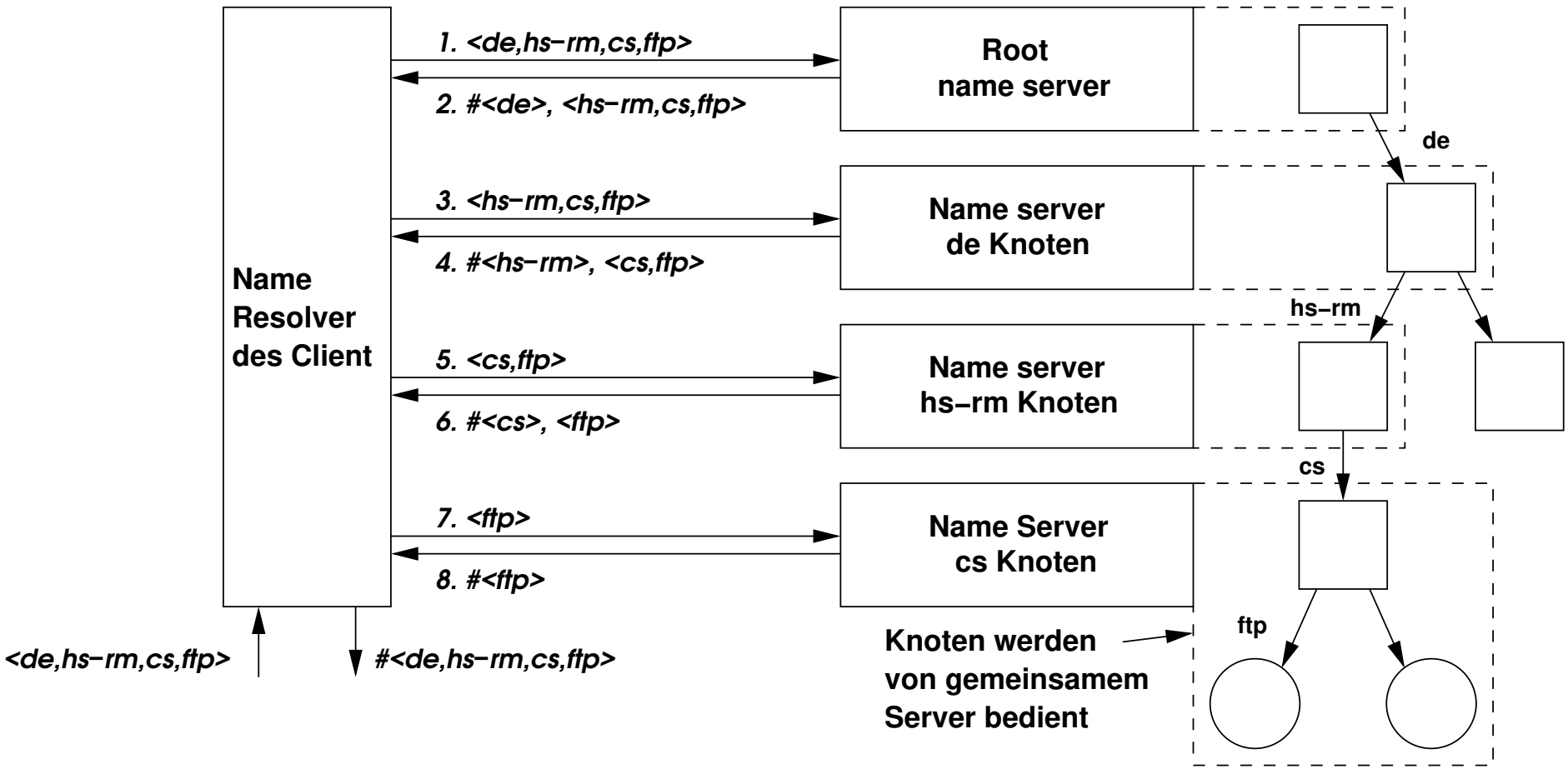




Namensauflösung mit Name-Servern (1)

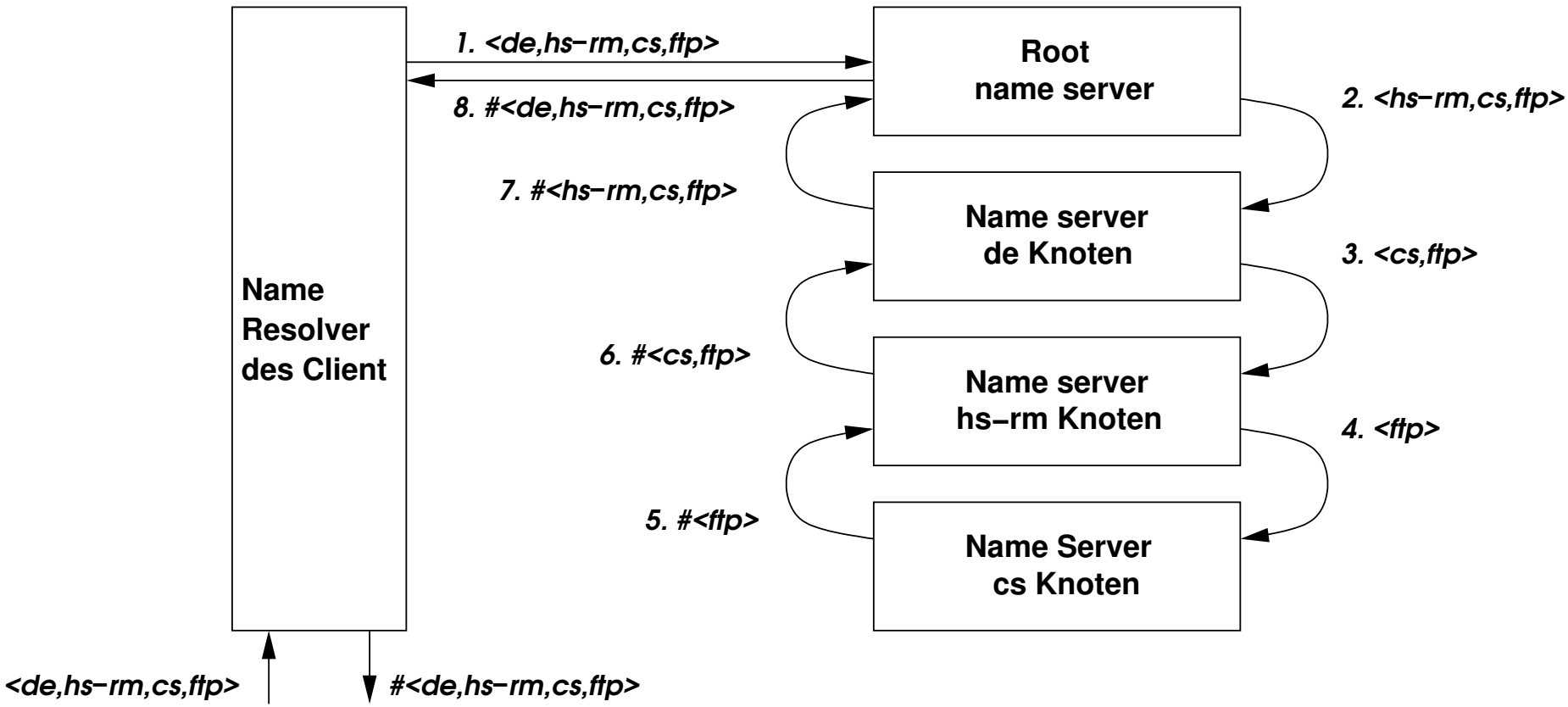
Hochschule RheinMain

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



Namensauflösung mit Name-Servern (2)

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



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

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 (Protokoll)

- Aktuelle Version 3 ist spezifiziert im RFC 4511.
- Unterstützung unter vielen Betriebssystemen.
- LDAP - Server beherrschen Replikation und Delegation (Referral).
- LDAPv3 unterstützt SSL, SASL und Kerberos-Authentisierung.
- Die meisten LDAP-Daten sind Textstrings (einfache Kodierung bei Netzübertragung), Binärdaten können verarbeitet werden.

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 Klasse.
- Es gibt Schemata für vordefinierte Klassen (z.B. Person, Organisation)
- Vererbung möglich
- Anwendungsspezifisch erweiterbar

Beispiel

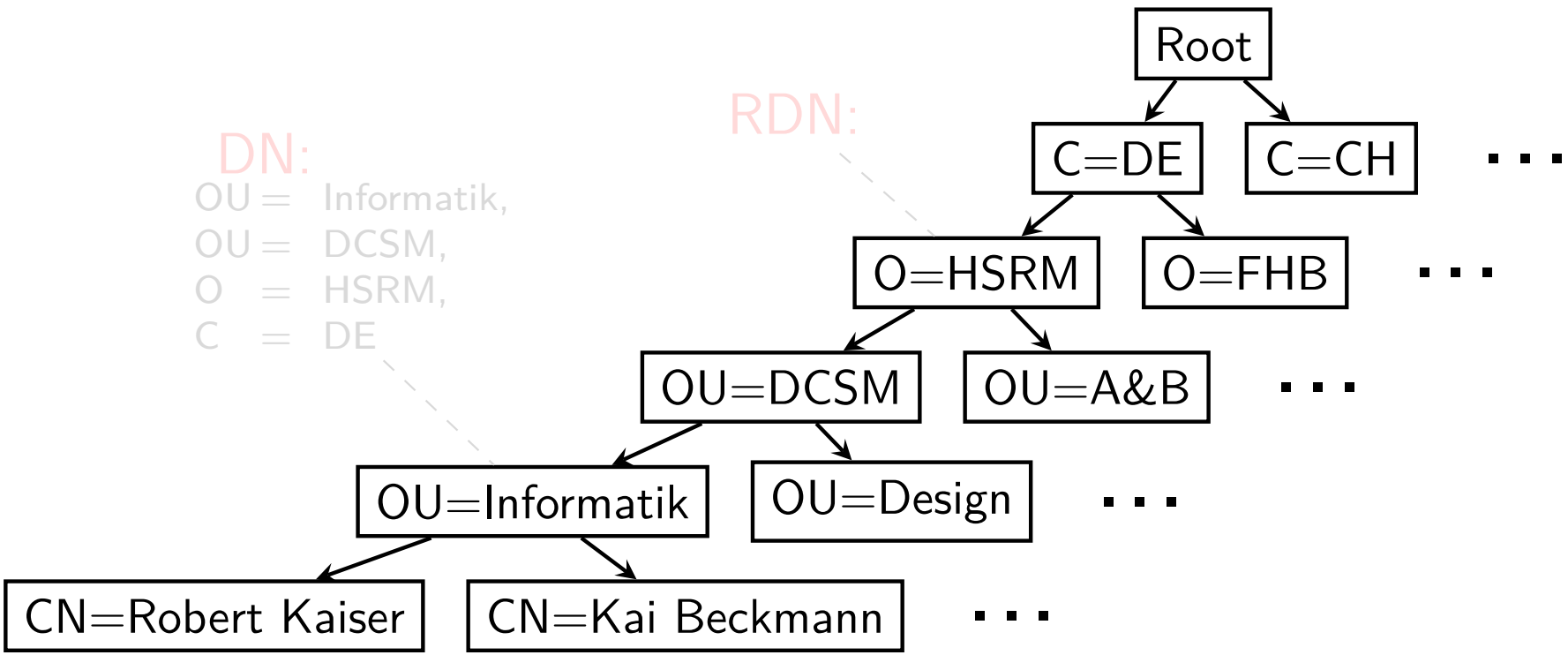
Hochschule RheinMain

Attribut	**Kürzel**	**Wert**
Country	C	DE
Locality	L	Wiesbaden
Organization	O	HSRM
OrganizationalUnit	OU	DCSM
OrganizationalUnit	OU	Informatik
CommonName	CN	Robert Kaiser

RDN und DN

Hochschule RheinMain

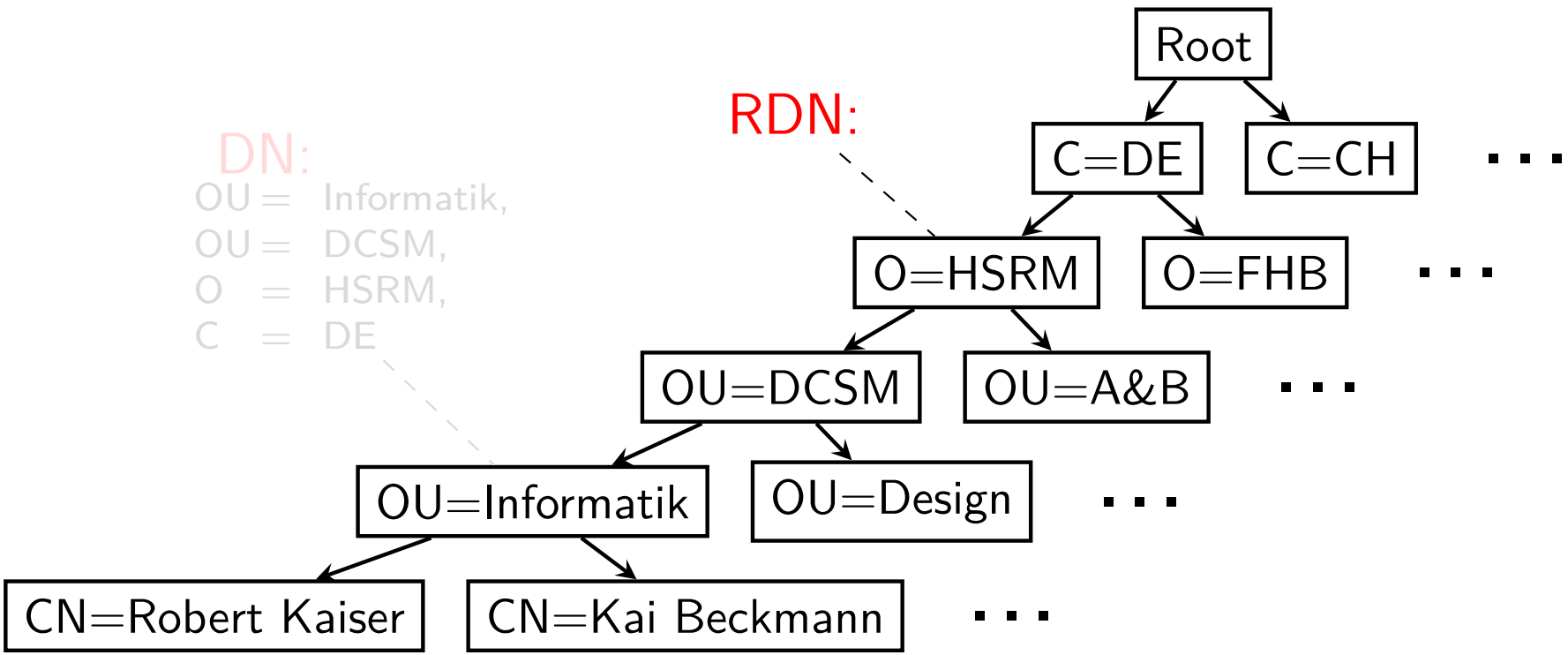
- Ausgangspunkt: DIT Wurzel → Base Object
- Jeder Knoten hat in seiner Ebene einen eindeutigen Namen, genannt Relative Distinguished Name (RDN)
- Zusammensetzung der RDNs von Knoten bis zur Wurzel heisst Distinguished Name (DN) (vgl. Pfadnamen)



RDN und DN

Hochschule RheinMain

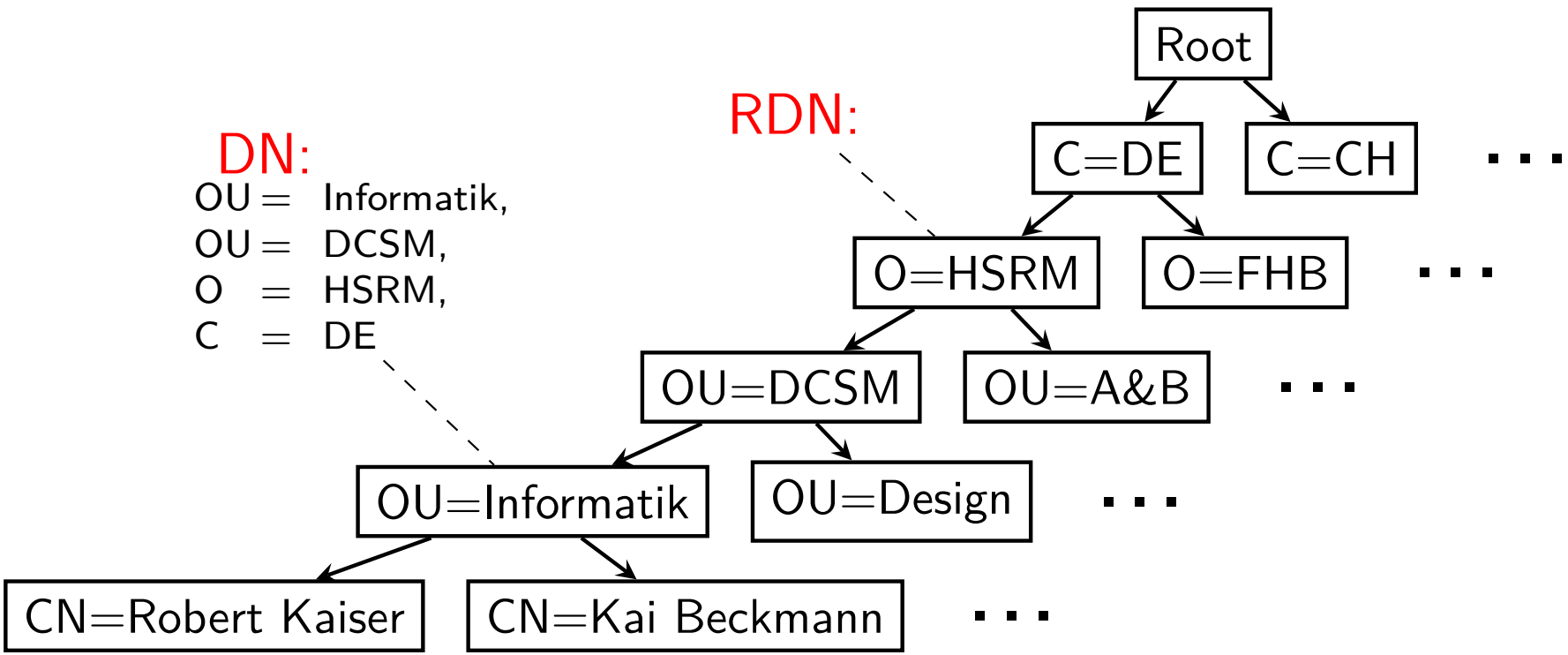
- Ausgangspunkt: DIT Wurzel → Base Object
- Jeder Knoten hat in seiner Ebene einen eindeutigen Namen, genannt Relative Distinguished Name (RDN)
- Zusammensetzung der RDNs von Knoten bis zur Wurzel heisst Distinguished Name (DN) (vgl. Pfadnamen)



RDN und DN

Hochschule RheinMain

- Ausgangspunkt: DIT Wurzel → Base Object
- 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)



Operationen

- Bind — Authentifizierung
- Add — Hinzufügen eines Eintrags
- Delete — Löschen eines Eintrags
- Search — Suchen von Einträgen
- Compare — Vergleichen von LDAP-Objekten
- Modify — Bearbeiten eines LDAP-Objekts
- ModifyRDN — Verschieben/umbenennen eines Objekts
- Abandon — Abbrechen einer laufenden Anfrage
- Unbind — Abmeldung eines Clients

Anfragen

Hochschule RheinMain

- Nutzung als Namensdienst
  - Finde Objekt bei gegebenem Distinguished Name.
  - z.B. read(/C=DE/O=HSRM/OU=Informatik/CN=Robert Kaiser), 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=*Kaiser)

Anfragen

Hochschule RheinMain

- Nutzung als Namensdienst
  - Finde Objekt bei gegebenem Distinguished Name.
  - z.B. read(/C=DE/O=HSRM/OU=Informatik/CN=Robert Kaiser), 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=*Kaiser)

Replikation

Hochschule RheinMain

- 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

Hochschule RheinMain

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

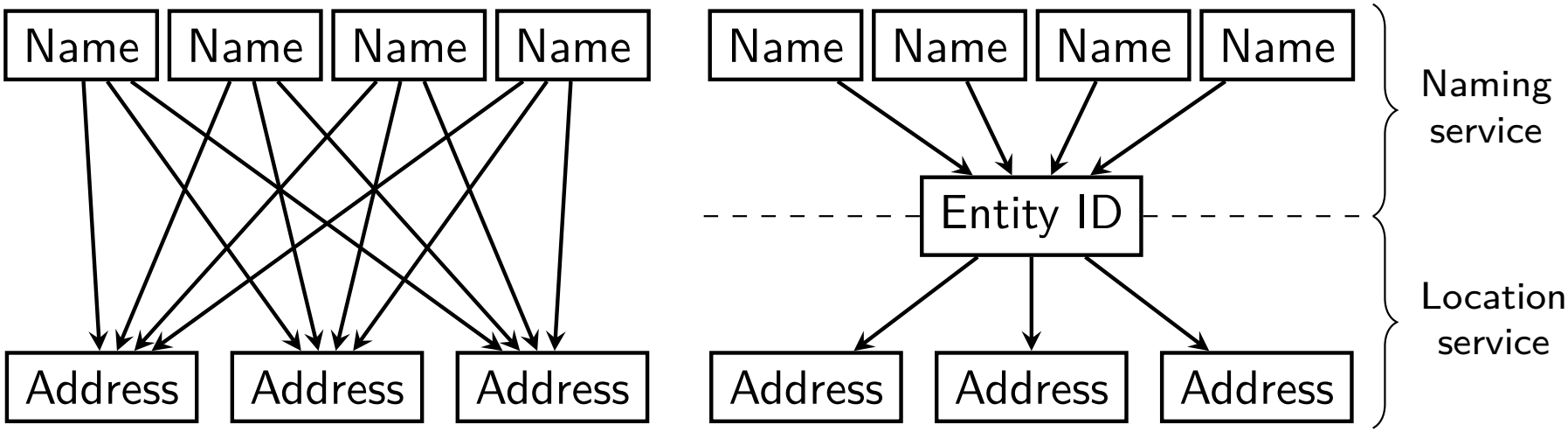
Produkte LDAP/X.500

- 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)
- ...

Lokationsdienste

Hochschule RheinMain

- 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

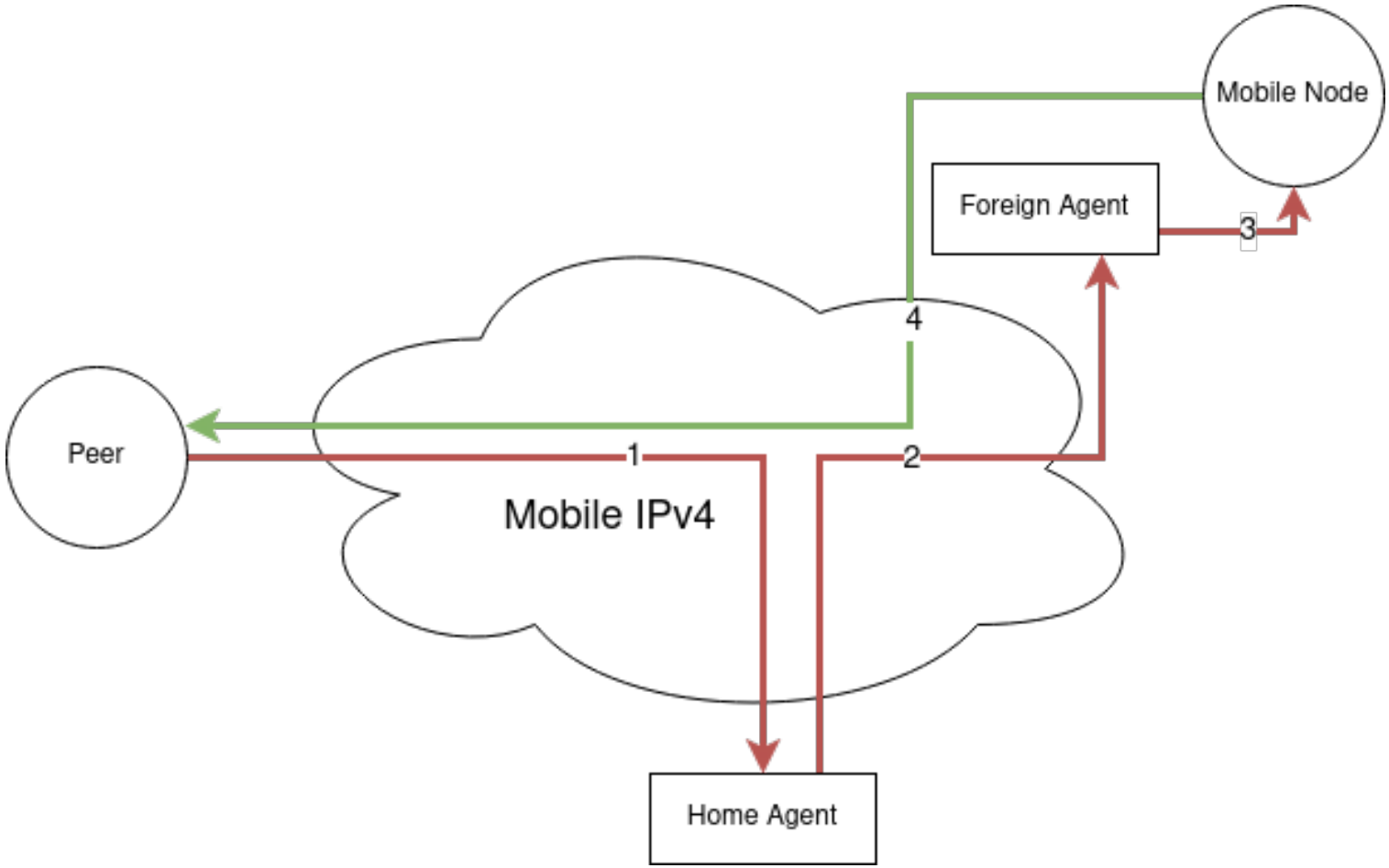


Identifier-Locator Split

Hochschule RheinMain

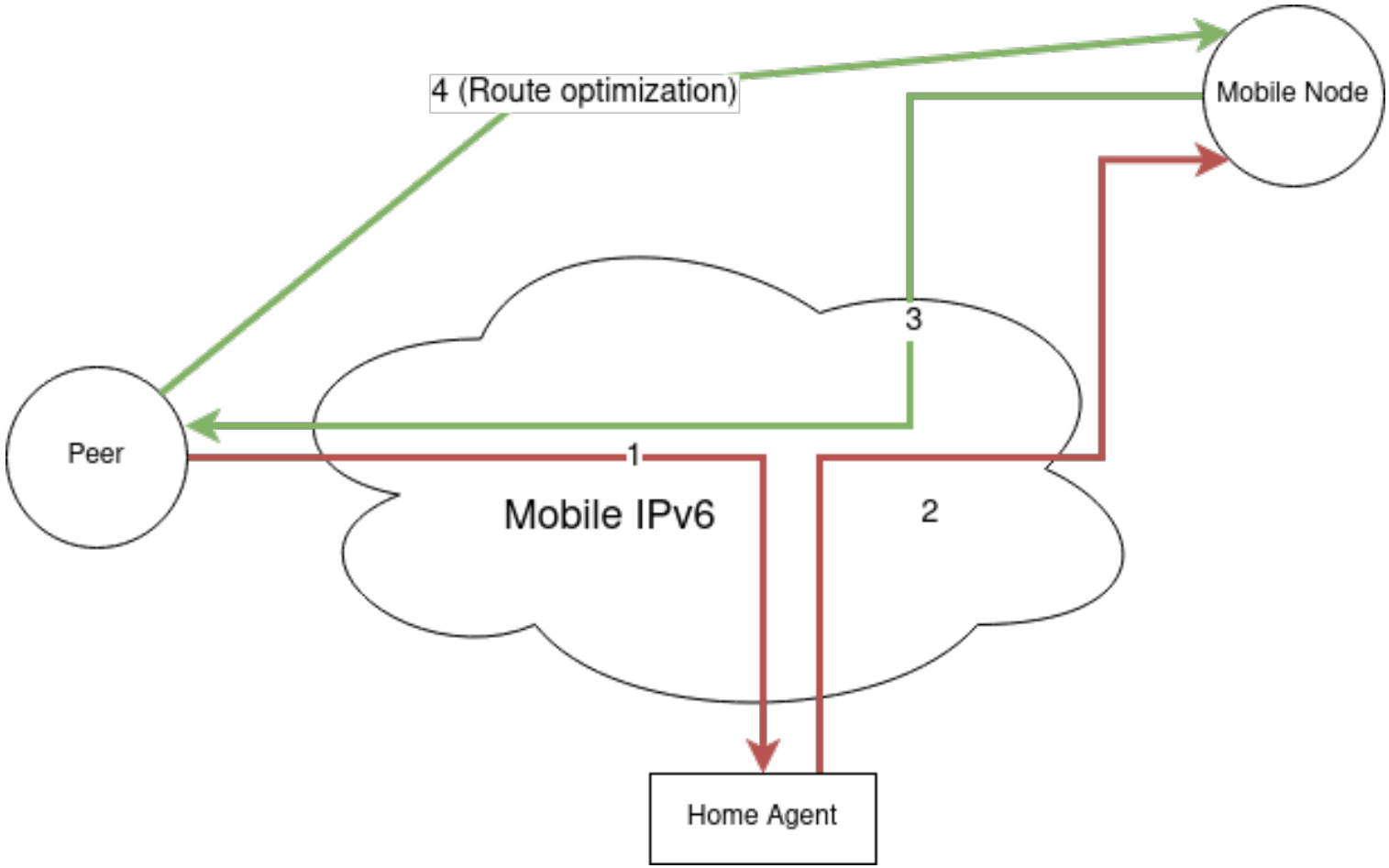
- Jeder Teilnehmer benötigt zwei Bezeichner (Adressen)
  - Ein Identifier (ID) bezeichnet, **wer** er ist.
  - Ein Locator (Loc) bezeichnet, **wo** er ist.
- In statischen Netzen sind beide Bezeichner statisch:
  - Können sich deshalb auf eine Adresse reduzieren.
  - Im Internet ist die ‚reguläre' IP Adresse ID und Loc
- In mobilen Netzen ändert sich der Locator
  - ID und Loc divergieren: ID-Loc Split
  - Netzwerk und Endsysteme müssen Dualität handhaben

Mobile IPv4



Mobile IPv6

Hochschule RheinMain



Zusammenfassung

- Namen können unique und pure/impure sein und beziehen ihre Bedeutung aus ihrem Kontext.
- Namensdienste dienen zur Auflösung von Namen zu Adressen.
- Verzeichnisdienste erweitern diesen Ansatz und ermöglichen eine Suche über weitere Eigenschaften.