

Übungsblatt 4

zur Vorlesung

Verteilte Systeme/Ubiquitous Computing

Wintersemester 2007/2008

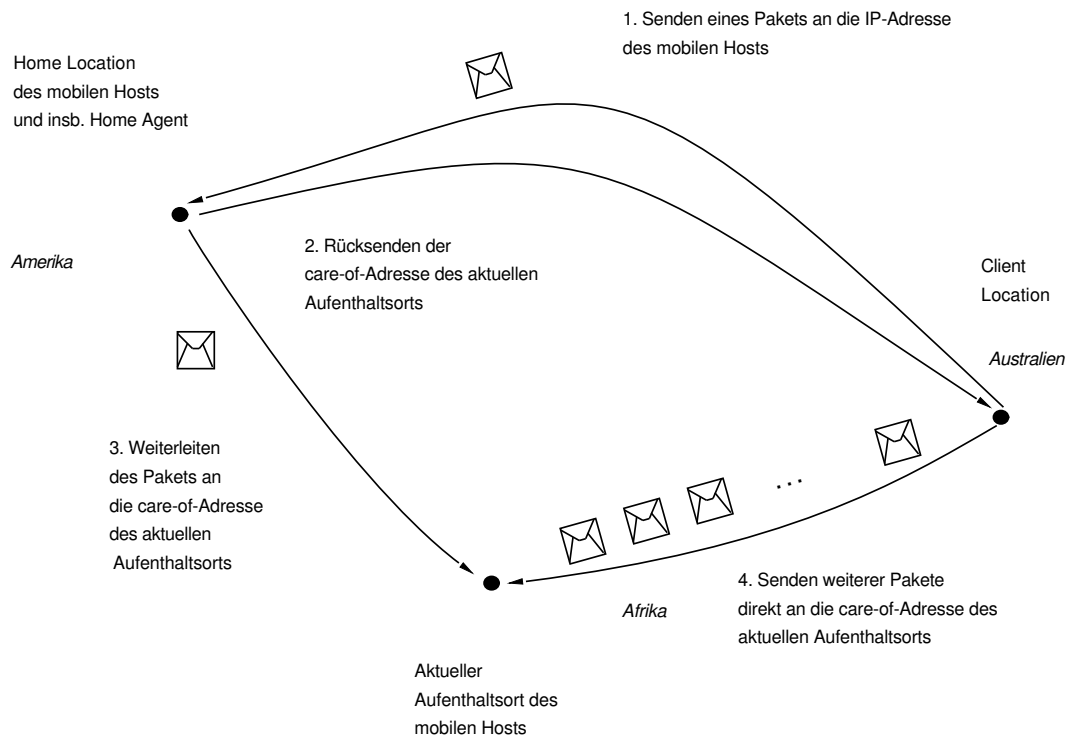
Achtung: Bitte beachten Sie auch die Web-Site der Vorlesung:
<http://www.mobile.ifi.lmu.de/Vorlesungen/ws0708/vs/>

Aufgabe 11: (H) Forwarding Pointers und Mobile IPv4

- a. Beschreiben Sie, wie Lokalisierung mittels Forwarding Pointers funktioniert.
- b. Welche Nachteile haben Forwarding Pointers?
- c. Beschreiben Sie, wie Mobile IPv4 funktioniert.

Antwort:

- a. Wenn eine Entität von Rechner A nach Rechner B migriert, so wird auf A eine Referenz auf den neuen Ort auf B hinterlassen. Dadurch ergeben sich Ketten von Forwarding Pointers.
- b. Die Kette von Forwarding Pointers kann sehr lang werden. Je länger die Kette ist, umso höher ist die Wahrscheinlichkeit, dass die Kette bricht. Außerdem sind bei langen Ketten auch die Lokalisierungszeiten lang.
- c. Mobile IPv4



Zusätzlich ist zu beachten, dass das mobile Objekt nach jeder Migration seine Care-of-Adresse beim Home-Agent registrieren muss.

Aufgabe 12: (T) Mobile IPv6

- Was ist an Mobile IPv4 problematisch bzw. nachteilig?
- Die folgenden RFCs beschreiben Mobile IPv6:
 - RFC 3775 - Mobility Support in IPv6
 - RFC 3776 - Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents

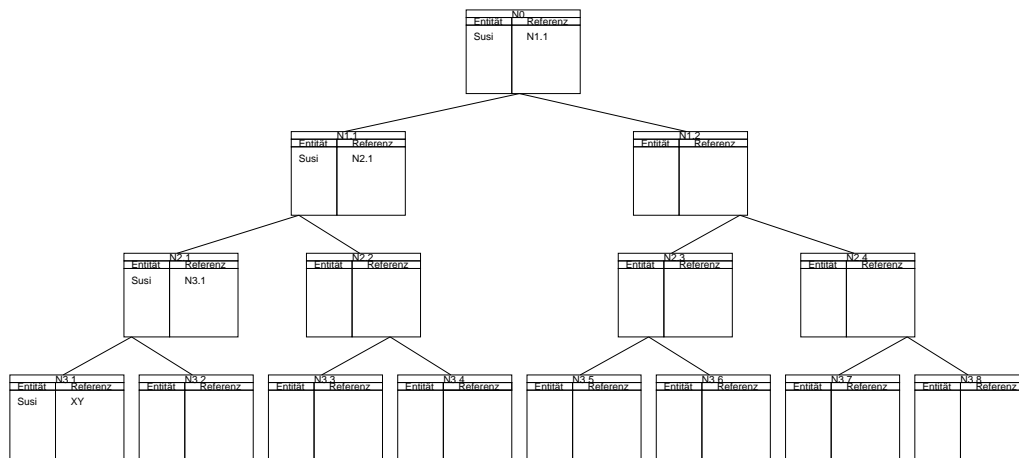
Schildern Sie die Funktionsweise von Mobile IPv6 und stellen Sie dabei die Unterschiede zur Vorgängerversion (Mobile IPv4) dar.

Antwort:

- Selbst wenn sich Client und mobiler Host im selben Netz befinden, wird erst der Home Agent kontaktiert. Problematisch ist weiterhin die Existenz und Erreichbarkeit des Home Agents (sowohl für den Anfrager als auch für das mobile Objekt, das eine Care-of-Adresse registrieren möchte).
- Siehe Folien.

Aufgabe 13: (H) Globe Location Service

- a. Erklären Sie, warum traditionelle Namenssysteme schlecht für die Lokalisierung mobiler Einheiten geeignet sind.
- b. Der Wetterfrosch Anton bietet einen Wetterdienst an. Da er viel herumhüpft, hat er sich entschieden, für seine Kunden über einen Globe Location Service (GLS) auffindbar zu sein. Dieser Service verwaltet, wie in der GLS-Grafik dargestellt, 15 Knoten. Die Knoten der untersten Ebene, Knoten N3.1 - N3.8, sind den Lokalisierungsgebieten 1 - 8 zugeordnet.
 - (i) Der Wetterfrosch befindet sich in Gebiet 1, das von N3.1 verwaltet wird. Füllen Sie die Tabellen der GLS-Grafik entsprechend aus.
 - (ii) Er hüpft hinüber zu Gebiet 2 (N3.2). Passen Sie die Tabellen hieran an.
 - (iii) Anschließend bewegt er sich ins Gebiet 5 (N3.5). Führen Sie alle notwendigen Änderungen aus.
 - (iv) Anton kann die anfallende Arbeit nicht mehr allein bewältigen und stellt die Wetterfröschin Berta ein. Berta befindet sich in Gebiet 4 (N3.4). Fügen Sie dies in die GLS-Grafik ein.
 - (v) Susi ist eine Kundin des Wetterdienstes. Sie hält sich in Gebiet 1 (N3.1) auf. Sie fragt nach dem Wetter, so dass das System versucht, den nächsten Wetterfrosch zu orten. Stellen Sie dar, wie diese Suchanfrage abläuft. Wird diese Anfrage von Anton oder von Berta beantwortet?

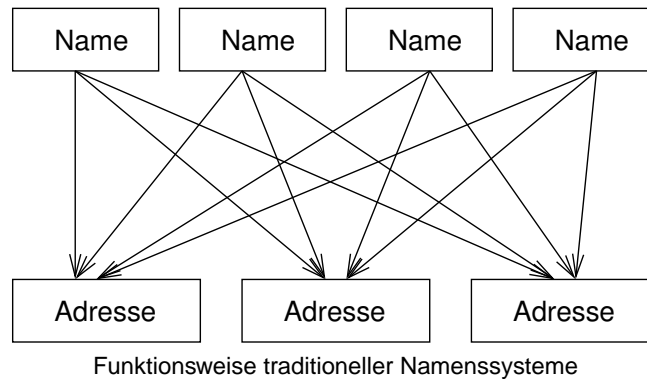


GLS-Grafik

- c. Inwiefern ist der Globe Location Service leichter skalierbar als eine vollständig zentrale Lösung?

Antwort:

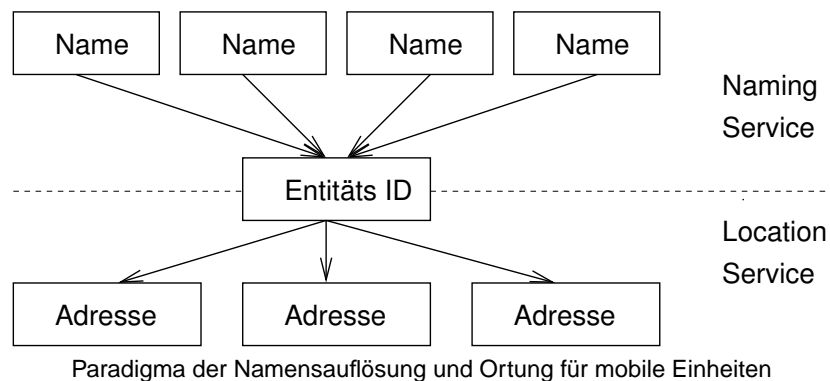
- a. Traditionelle Namenssysteme wie z.B. das Domain Name System bilden einen Namen direkt auf eine Adresse ab.



Dies führt zu der Versuchung, den Namen bereits implizit eine Ortsinformation mitzugeben. Anfänglich ließen die Top-Level-Domains des DNS keine Rückschlüsse auf den Ort der benannten Entität zu (.edu, .gov, .com, .org etc.) Dies änderte sich mit der Einführung von Endungen wie .de. Endet ein Name mit .de, so liegt die Vermutung nahe, dass sich die zugehörige Entität in Deutschland befindet.

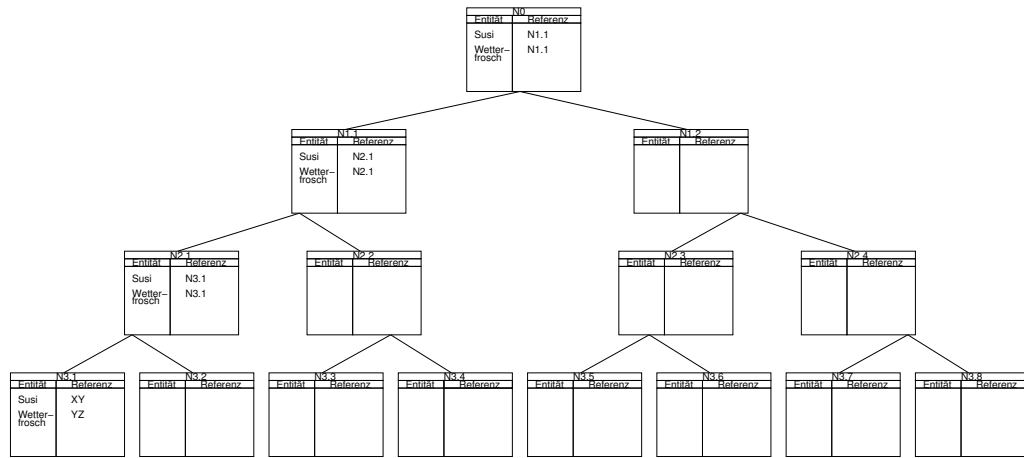
Es ist möglich, den Namen *Imu.de* auf einen Server in den USA zeigen zu lassen, doch dies führt einerseits zu Verwirrung und andererseits bei hoher Mobilität zu Ineffizienz. Die Ineffizienz rührt daher, dass bei einer erneuten Verlagerung der *Imu.de*-Entität in den USA entweder am vorherigen Ort ein symbolic link hinterlegt werden muss, wodurch dem Ortungsvorgang ein weiterer Schritt zugefügt wird, oder aber der Eintrag im Namensregister (DENIC) in Deutschland von den USA aus verändert werden muss, was aufgrund der hohen geografischen Entfernung zwischen Deutschland und den USA recht lange dauern kann.

Deswegen gehen Namenssysteme für mobile Einheiten dazu über, die Auflösung von Namen in Identifier und die Ortung der Entitäten zu entkoppeln.

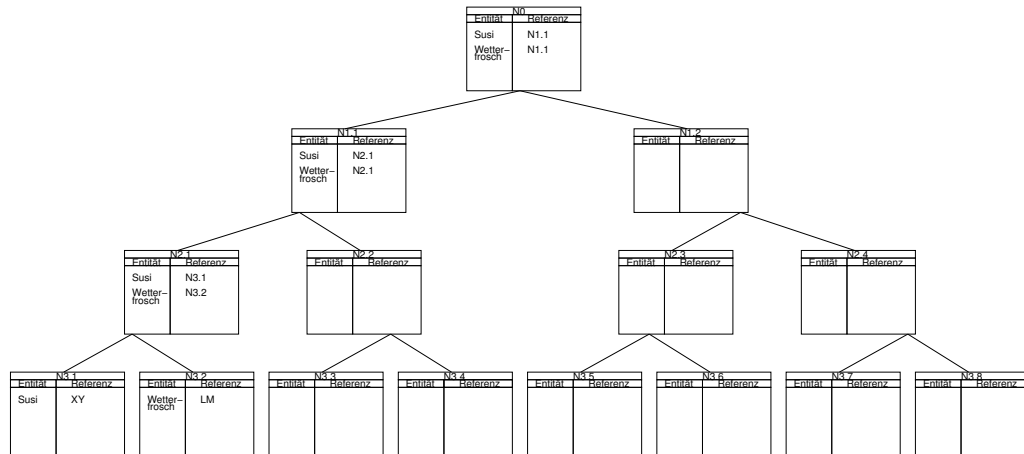


Dies ermöglicht es dem Ortungssystem, die Datenstrukturen für die Ortung unabhängig von den anfänglichen Orten der Entitäten aufzubauen. Die Struktur des "location systems" ist nicht mehr implizit durch die Namen der zu verwaltenden Einheiten vorgegeben. Unabhängig von den Namen kann die Datenstruktur rein nach Effizienzgesichtspunkten organisiert und verändert werden.

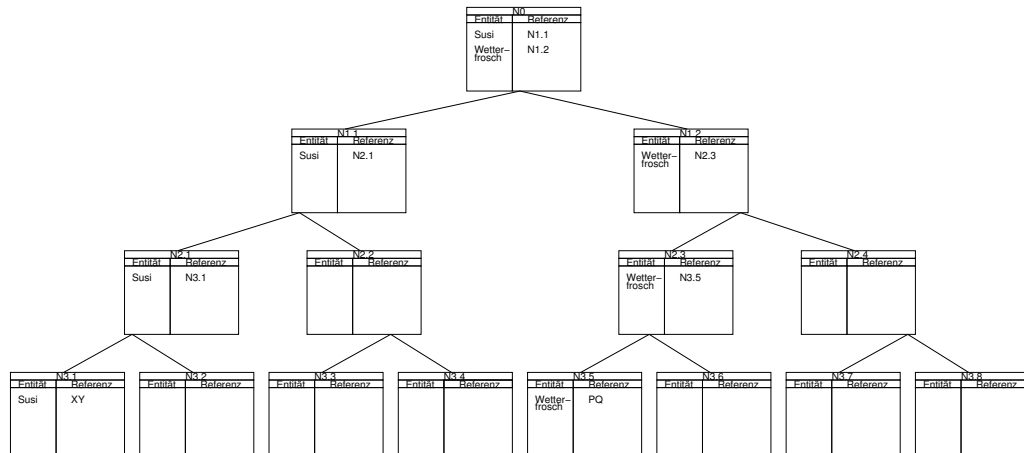
b. (i)



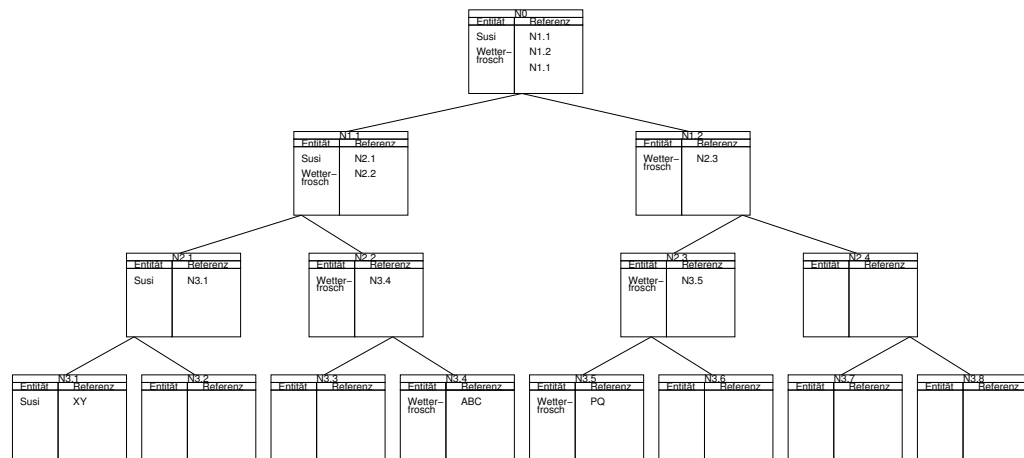
(ii)



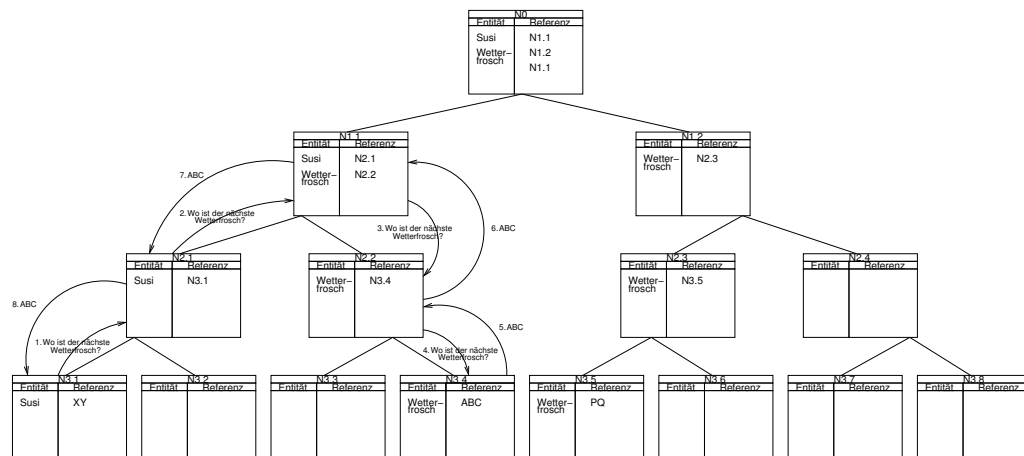
(iii)



(iv)



(v)



- c. Bei oberflächlicher Betrachtung wird durch den “Globe Location Service” gegenüber einem zentralen Server, der alle Aufenthaltsorte der zu ortenden Entitäten kennt, nichts gewonnen. Denn in der Wurzel des “Globe Location Service” muss für jede Entität ein Eintrag gehalten werden. Hierbei ist zu bedenken, dass nicht der Speicherplatz das Skalierungsproblem erzeugt, sondern die Arbeitslast des Servers. Ein Eintrag in der Wurzel muss einen Identifier für die Entität enthalten und eine oder mehrere Referenzen auf Knoten in der darunter liegenden Hierarchieebene. Nehmen wir an, dass ein solcher Eintrag 1 KB an Speicherplatz benötigt. Dann könnten eine Milliarde Entitäten mit einem Tera-Byte verwaltet werden. Dies stellt bereits bei heutiger Technologie kein größeres Problem dar.

Wenn ein zentraler Server jedoch sämtliche Ortungsanfragen und Ortsänderungen bearbeiten müsste, wäre er hoffnungslos überfordert. Somit bietet der “Globe Location Service” gegenüber einer zentralen Lösung dadurch eine bessere Skalierbarkeit, dass der Wurzelknoten mit der Bearbeitung der meisten Ortsänderungen und Ortsanfragen nicht belastet wird.

Aufgabe 14: (H) Dienste: Grundlagen

- Wie hängen Diensttyp, Dienst, Dienstangebot und Dienstigenschaften zusammen?
- Erklären Sie das Grundprinzip der Dienstvermittlung mittels eines Traders und nennen Sie Beispiele, wo dies in der Praxis eingesetzt wird!
- Bewerten Sie die Verfahren „Caching“ und „Polling“ für den Zugriff auf dynamische Dienstigenschaften im Vergleich zu einander!

Antwort:**a. Abgrenzungen**

- **Diensttyp:**
Diensttypen bestimmen sich durch einen Rechnerschnittstellentyp. Sie sind ein abstraktes Konzept eines Dienstes, legen Funktionalität, Operationssignaturen und Rechnerverhalten fest.
- **Dienst:**
Ein Dienst ist eine Funktion, die von einem Objekt an einer Schnittstelle zur Verfügung gestellt wird. In einer objektorientierten Sichtweise sind Dienste Instanzen eines Diensttyps. Neben den vom Diensttyp festgelegten Charakteristiken bestehen auch Unterschiede zwischen den Diensten, nämlich bei den *Diensteigenschaften*:
- **Diensteigenschaften:** Diensteigenschaften (oder auch Service Properties) können technologische, informationsbezogene, technische oder auch unternehmensbezogene Aspekte des Dienstes betreffen, also allgemein: rechner- oder verhaltensunabhängige Aspekte eines Diensttypen. Diensteigenschaften gehören zu Diensten und sind Instanzen von Diensteigentenschaftstypen. Es werden statische und dynamische Diensteigenschaften unterschieden.
- **Dienstangebot:**
Wird ein Dienst angeboten, so spricht man von einem Dienstangebot. Es handelt sich also Beschreibung eines Schnittstellenbezeichners, der vom Trader vermittelt wird. Zu einem Dienstangebot gibt es wiederum Dienstangebotseigenschaften, die Werte von Diensteigentenschaftstypen sind.

b. Grundprinzip eines Traders

Ein *Exporteur* muss sich beim Trader registrieren, damit das Dienstangebot (wozu auch die Adresse gehört) abgerufen werden kann. Ein potenzieller Nutzer, der *Importeur*, stellt eine Anfrage nach einem gewünschten Dienst, worauf der Trader mit den Daten aus seinem Register (nach *Search* und *Selection*) antwortet. Je nach Algorithmus für dynamische Diensteigenschaften sind noch zusätzliche Schritte nötig, siehe unten. Über die Adresse kann der Importeur dann Kontakt zum Exporteur herstellen und den Dienst nutzen.

Ein sehr bekanntes Modell eines Dienstvermittlungssystems für Web-Services ist z.B. UDDI.

c. Caching ↔ Polling

Beim Caching werden in festgelegten Abständen bei den Diensten die dynamischen Eigenschaften abgefragt und zwischengespeichert. Diese Werte werden dann den Importeuren weitergegeben.

Beim Polling werden keine Werte zwischengespeichert, sondern bei einer Anfrage die in Frage kommenden Dienste abgefragt und diese Werte dann zurückgegeben.

Beide Verfahren müssen anhand der verursachten Netzlast und der Aktualität der Diensteigenschaften verglichen werden: Beim Polling ist absolute Aktualität garantiert. Bei wenigen Anfragen an den Trader ist auch die entstehende Netzlast gering, bei vielen steigt sie aber stark an. Dies ist beim Caching anders. Die Anzahl der Abfragen beim Dienst bleibt konstant, unabhängig von der Nutzerzahl, was bei vielen Anfragen von Vorteil ist. Beim Caching muss der richtige Trade-Off zwischen Abfragehäufigkeit und Netzlast gefunden werden. Bei wenigen Änderungen der Eigenschaften, sind viele Abfragen nur unnötige Netzlast, bei vielen sind entsprechend oft veraltete Werte auf dem Server zwischengespeichert.

Die Entscheidung, welches Verfahren zum Einsatz kommen soll, ist damit vom konkreten Fall abhängig. Auch ein abwechselnder Einsatz muss in Betracht gezogen werden.