

# Rechnergestützter DNS-Management-Prozess mit DoctorDNS

Dr. Stefan Bier, fgn GmbH  
bier@fg-networking.de

Brian Worden, RHRK  
worden@rhrk.uni-kl.de

Maurice Massar, RHRK  
massar@unix-ag.uni-kl.de

Thomas Esselen, RHRK  
esselen@rhrk.uni-kl.de

**Abstract:** Das Domain Name System (kurz DNS genannt) ist ein integraler Bestandteil des Internets (vielleicht **der** integrale Bestandteil).

Vorgestellt wird ein Softwaresystem zur komfortablen Konfiguration nicht nur dieses Dienstes, sondern auch zur Lösung vieler Aufgaben im Rahmen des IP Address Managements (IPAM).

## 1 Historie

### 1.1 Host-NIC Tabelle

Als das Internet (als Arpanet) Ende der sechziger Jahre des 20. Jahrhunderts aus der Taufe gehoben wurde, existierte nur eine Handvoll Rechner, die an diesem Verbund teilnahmen. Mit Aufkommen des TCP/IP-Protokolls wuchs die Zahl der Rechner sprunghaft an. Um diese Rechner nicht immer über ihre IP-Adresse ansprechen zu müssen, wurde eine Tabelle auf jedem Rechner gehalten, welche die Abbildung IP-Adresse  $\longleftrightarrow$  Rechnername zum Inhalt hatte (vgl. Abbildung 1).

Damit alle Rechner des Internets die gleiche Tabelle benutzen, wurde diese an einer zentralen Stelle gepflegt, dem NIC. Alle Rechner holten nun in bestimmten Zeitabständen eine Kopie dieser Tabelle von den Servern des NIC.

Als die Anzahl der teilnehmenden Rechner im Internet immer größer wurde, erwies sich dieses Vorgehen als nicht mehr praktikabel. Der Aufwand zur Pflege der hosts.txt stieg stark an, ebenso die Zeitspanne, bis ein neuer Rechner allen anderen Rechnern bekannt war.

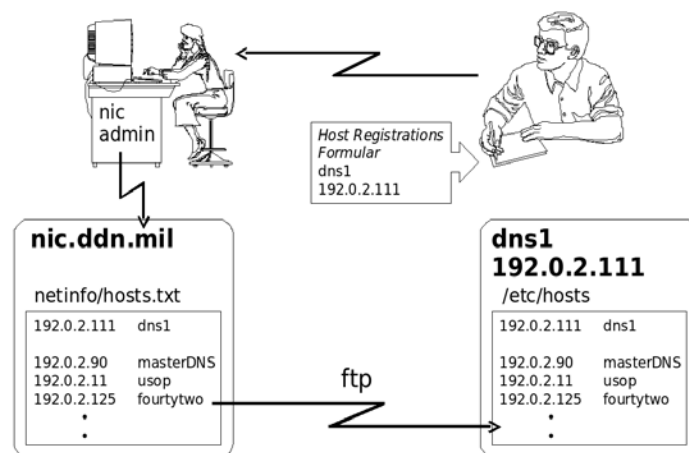


Abbildung 1: Szenario mit Host-Nic Tabelle

## 1.2 DNS

Nach einem Zwischenschritt ([Pos79]) im Jahre 1979 wurde dann 1984 ein neues System vorgeschlagen, das die Einschränkungen nicht mehr haben sollte. Das Domain Name System beruht auf einer verteilten Datenbank, die auf dedizierten Rechnern des Internets, sogenannten Nameservern, verwaltet wird. Ein Rechner, der den Namen bzw. die IP-Adresse eines Kommunikationspartners erfahren will, wendet sich dann an einen ihm bekannten Nameserver. Diese Nameserver verwalten jeweils nur einen Ausschnitt aus dem Internet, ein Protokoll sorgt dafür, dass nicht vorhandene Informationen aber von dem entsprechenden Nameserver beschafft werden können.

Ein Nameserver ist eigentlich ein Programm, das auf einem bestimmten Rechner ausgeführt wird. Dieses Programm wird über Konfigurationsdateien gesteuert. Diese Konfigurationsdateien enthalten für jeden verwalteten Rechner sogenannte ResourceRecords, in denen Informationen zu Name, IP-Adresse, Betriebssystem, EMail usw. gespeichert sind. Weitere Informationen in diesen Konfigurationsdateien betreffen die Struktur des DNS-Systems (root-server, zonen, primary und secondary nameserver usw.). Eine detaillierte Beschreibung des DNS-Systems liefert [CL02].

## 1.3 Konfiguration des DNS

Für das Erstellen der Konfigurationsdateien eines Nameservers (DNS-Server) existieren verschiedene Möglichkeiten:

### **1.3.1 Manuell**

Ein Systemadministrator erstellt die Konfigurationsdateien mittels Editor. Dieser Ansatz bietet die größtmögliche Flexibilität. Der Systemadministrator kann bestimmen, welche Informationen zu einem Rechner in welcher Form in die Konfigurationsdateien geschrieben werden. Er kann die Struktur der Konfigurationsdateien, natürlich im Rahmen der Syntax des Programms, frei bestimmen. Die Topologie seines Netzwerkes und deren Abbildung in Konfigurationsdateien ist nahezu beliebig.

### **1.3.2 Software**

Diverse Firmen bieten Softwarelösungen zum Erstellen der Konfigurationen. Beispiele hierfür sind VitalQIP von Lucent [Luc], DNSOne von Infoblox [Inf], IP Address Domain Manager von Nortel [Nor], und viele mehr.

Im Open Source – Bereich existiert eine ganze Reihe von Lösungen zur Verwaltung von DNS-Domains. Diese setzen in der Regel den Einsatz eines relationalen Datenbanksystems wie MySQL oder PostgreSQL voraus.

Diese Systeme befreien den Administrator von der Notwendigkeit, sich mit der Syntax der Nameserverkonfiguration auseinander setzen zu müssen.

## **1.4 Probleme dieser Ansätze**

### **1.4.1 Manuelle Konfiguration**

Diese Methode setzt einen Spezialisten für Nameserverkonfiguration voraus. Bei grossen Netzen (> 20 Hosts) ist es schwierig, Konsistenzen z.B. zwischen Forward- und Reversezonen zu gewährleisten. Da die Syntaxprüfung durch das Nameserverprogramm erfolgt, sind eventuell mehrere Anläufe bis zum erfolgreichen Neustart zu tätigen.

### **1.4.2 Closed Source Software**

Im Allgemeinen lässt sich eine käuflich erworbene Software nicht an eine vorhandene Netzwerkstruktur anpassen. Der Anpassungsprozeß läuft in der Regel umgekehrt.

Manche Lizenzmodelle führen zu erheblichen Investitionskosten.

Fehler oder Sicherheitslücken in der Software können nicht direkt behoben werden, sondern setzen ein Update des Herstellers voraus.

### **1.4.3 Open Source Software**

Bei quelloffenen Systemen fallen andere Probleme an. Es ist fast immer eine Konfiguration und Wartung mehrerer Softwareprodukte nötig (z.B PHP und MySQL). Da viele Open

Source Projekte keinen besonderen Wert auf Dokumentation legen. ist der Einarbeitungsaufwand sehr gross. Eventuelle Erweiterungen müssen in Abstimmung mit dem Autor der Software vorgenommen werden. Das Ändern eines Attributes kann einen erheblichen Aufwand bedeuten, wenn sowohl Datenbankschema als auch grafische Oberfläche angepasst werden muss.

## 2 DoctorDNS

Zur Vermeidung dieser Probleme wurde unter der GPL (GNU General Public License) ein eigenes System entwickelt, das folgende Anforderungen erfüllt:

1. Mehrbenutzerfähig
2. Gleichermassen geeignet für DNS/DHCP – Spezialisten und „normale“ Rechnerbenutzer
3. Client–Server Architektur
4. Erweiterbar
5. Einsetzbar in heterogener Systemumgebung

### 2.1 Ganymede

Als Basis des DNS–Managementsystems dient Ganymede [Abb].

Dieses Directory–Management–System wird seit 1985 an der University of Texas at Austin unter der Federführung von Jonathan Abbey entwickelt und ist dort im täglichen Einsatz.

Es deckt einen Teil der gestellten Anforderungen wie Client–Server Architektur, Benutzer- und Rechteverwaltung und Mehrbenutzerfähigkeit bereits ab. Ebenso integriert ist eine grafische Oberfläche (GUI). Durch das Schema–Konzept ist Ganymede erweiterbar. Da es in Java implementiert wurde, ist auch eine weitgehende Systemunabhängigkeit gewährleistet.

Die Erweiterung von Ganymede geschieht durch Implementierung eines Schema-Kits. Ein Schema–Kit besteht aus einer Schemadefinition und diversen Custom–Klassen zur Anpassung der Funktionalität.

Zur Erstellung der Schemadefinition stellt Ganymede einen Schema–Editor zur Verfügung. Jede Änderung des Schemas wird automatisch in die Darstellung der GUI–Clients übernommen.

Abbildung 3 zeigt einen Auszug aus der Schemadefinition und das zugehörige Layout des Clients.

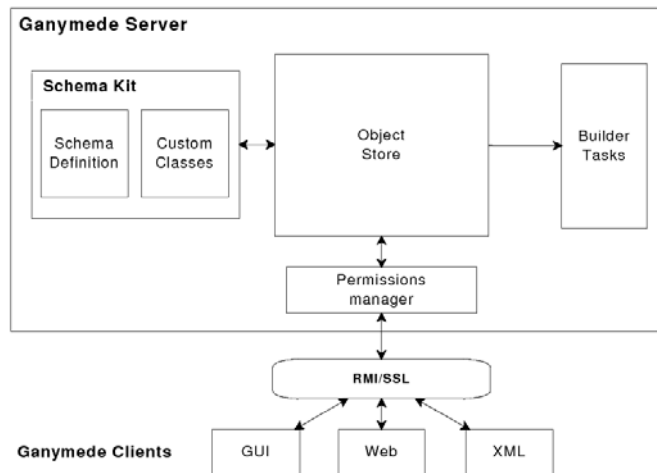


Abbildung 2: Übersicht der Ganymede-Komponenten

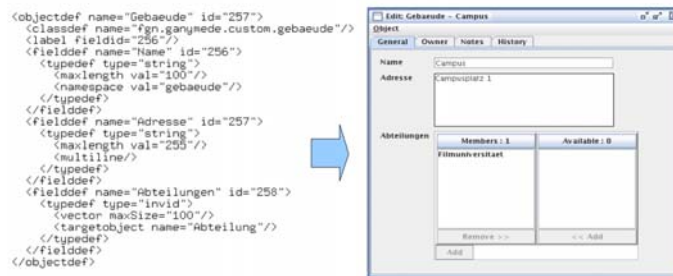


Abbildung 3: Layout aus Schemadefinition

## 2.2 DoctorDNS als Schema-Kit

DoctorDNS bildet ein solches Schema-Kit. Mit seiner Hilfe ist man in der Lage, Ganymede zum Verwalten von DNS bzw DHCP Daten einzusetzen. Die Eingabe der Daten erfolgt entweder über den grafischen Client oder über die XML-Schnittstelle (siehe Abbildung 2).

### 2.2.1 Komponenten

DoctorDNS selbst ist in mehrere Komponenten aufgeteilt. Die erste Komponente, das Schema, definiert die Datenstruktur der Anwendung, die Beziehungen der einzelnen Datenkomponenten untereinander (Assoziation, Komposition) sowie die Reihenfolge der Eingabefelder in der grafischen Oberfläche.

Eine weitere Komponente von DoctorDNS ist eine Sammlung von Klassen, welche die Funktionalität von Ganymede erweitern. Ganymede selbst stellt schon diverse Operationen bereit, die beim Verändern des Datenbestands ausgeführt werden können. Zu Erwähnen ist hier z.B. eine Validierung der Eingabe in ein Feld gegen eine Regular Expression, gegen eine Liste von erlaubten bzw. verbotenen Zeichen oder, bei numerischen Feldern, gegen einen Wertebereich.

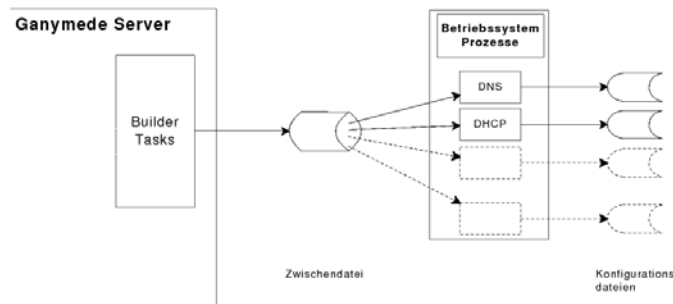


Abbildung 4: Komponenten von DoctorDNS

Möchte man aber eine weitergehende Funktionalität realisieren, z.B. Erzeugen eines neuen Objekts bei dem Editieren eines anderen Objekts oder Beschränkung der Auswahlmöglichkeiten aus einer Liste in Abhängigkeit des Wertes eines bestimmten Feldes, so muss man, in Java, eine Klasse schreiben. Eine Instanz dieser Klasse wird beim Editieren eines Objekts automatisch erzeugt. Methoden dieser Instanz, die beim Editieren von Feldern des Objekts aufgerufen werden, können dann das gewünschte Verhalten erzeugen.

Die dritte Komponente von DoctorDNS ist eine Taskklasse (BuilderTask), die bei bestimmten Ereignissen im Ganymede-Server instanziiert wird. Diese Ereignisse können sein:

- Abspeichern (commit) eines Objektes
- Erreichen eines Zeitpunktes
- Auslösung durch den Benutzer (Administrator)

Die Taskklasse von DoctorDNS wird im Allgemeinen durch den Administrator instanziiert, indem er die Ausführung der zugehörigen Task anfordert. Bei der Aktion wird der Teil des Datenbestandes, der zur Erstellung der jeweiligen Konfigurationsdatei benötigt wird, in eine Zwischendatei geschrieben. Dies kann auch der gesamte Datenbestand sein. Dieser Teil der Konfigurationserstellung ist in Java implementiert. er ist integraler Teil des Ganymede-Servers.

Der zweite Teil des Erstellungsprozesses, d.h. die aktuelle Generierung der Konfigurationsdatei ist schließlich als Betriebssystemprozess realisiert, der parallel und unabhängig

zum Ganymede-Server ausgeführt wird. Während dieser Zeit können die Benutzer mit dem System weiterarbeiten.

Da dieser zweite Teil unabhängig von Ganymede abläuft, bleibt die Wahl der Implementierung frei, das heisst man ist hier nicht mehr an Java gebunden.

### 2.3 Quasiautomatisierter Prozess zur Vergabe von IP-Adressen

Eine Möglichkeit zur Bedienung von DoctorDNS/Ganymede ist die Benutzung der grafischen Oberfläche. Sie ist im Allgemeinen den Netzwerkadministratoren vorbehalten. Um den Workflow zur Vergabe einer IP-Adresse noch weiter zu automatisieren, wurde ein Web-Eingabeformular geschaffen, das allen Mitarbeitern der Uni KL über die Homepage zur Verfügung steht. Abbildung 5 zeigt das Layout dieser Webpage.

Hier muss der Antragsteller einige Angaben machen, die zu seiner Identifikation und zur Bestimmung des Netzwerkes und der Domain dienen, in denen die neue IP-Adresse angelegt werden soll. Durch den Einsatz von List-Feldern bleiben die Möglichkeiten der Falscheingabe beschränkt.

**IP-Adresse beantragen (Festnetz)**

**Antragssteller:**

Name:  Telefon:

E-Mail:

**Gerät:**

Hostname:  Domain:

Abteilung:

Gebäude:

Telefon:

Gerät:

Kommentar:

Name: Pflichtfeld  
 E-Mail: Pflichtfeld  
 Hostname: Pflichtfeld  
 Domain: Pflichtfeld  
 Abteilung: Pflichtfeld  
 Gebäude: Pflichtfeld  
 Raum: Pflichtfeld

Für Einträge in Domains die nicht in der Auswahl enthalten sind, bitte den Hostnamen inklusive Domain angeben. Selbstverständlich können wir nur Einträge für Domains vornehmen die wir auch verwalten.

Wenn Sie neue Domains beantragen wollen, oder für weitere Fragen, setzen Sie sich bitte mit uns in Verbindung:

- Brian Worden, 2448
- Claudia Baltes, 3180
- Thomas Esselen, 3380
- E-Mail: dns@uni-kl.de

Abbildung 5: IP-Adresse beantragen — User-Sicht

Sendet der Antragsteller das Formular ab, wird einem Netzwerkadministrator eine Kopie des Antrags zugestellt und der Administrator entscheidet nach Sichtprüfung über Annahme oder Ablehnung des Antrags. In beiden Fällen wird der Antragsteller benachrichtigt, im Falle der Annahme gleich mit einer Liste der von ihm einzustellenden Netzwerkparameter. Abbildung 6 zeigt das Formular des Antrags in der Version für einen Netzwerkadministrator.

**IP-Adressen Antragsverwaltung**

Nach dem Eintragen, DNS-Lauf nicht vergessen!

1. host1.rhrk.uni-kl.de (Hans Mustermann)

ID: addf1513-cbf-4593-85c4-246ff5248b08

**Antragssteller:**

Name: Hans Mustermann      Telefon: 20557-1234  
E-Mail: hansl@rhrk.uni-kl.de

**Gerät:**

Hostname: host1      Domain: rhrk.uni-kl.de  
Abteilung: Regionales Hochschulrechenzentrum  
Gebäude: Bau 34      Raum: 210  
IP-Adresse:      Netzname:  
Telefon: 24448      MAC-Adresse:  
Gerät: PC      System:

Kommentar:  
Admin  
Kommentar:

• Netzname: Pflichtfeld

Abbildung 6: IP-Adresse beantragen — Admin-Sicht

Bei Annahme des Antrags wird über die XML-Schnittstelle von Ganymede ein neues System mit den entsprechenden Erweiterungen (Interface, IP-Adresse) erzeugt und in den Datenbestand übernommen.

Nach einem Lauf der DNS-BuilderTask steht dann der Eintrag zur allgemeinen Verfügung.

### 3 Erweiterungen

Geplante Erweiterungen des Systems Ganymede/DoctorDNS sind:

- Dynamic DNS
- Classless Delegation
- Integration von Realtime-Monitoring Werkzeugen



- DHCP Option 82

## 4 Fazit

Durch den Einsatz von DoctorDNS zusammen mit dem Web-Formular hat sich der Aufwand, der bei der Eintragung einer neuen IP-Adresse von Netzwerkadministratorsseite bereitzustellen ist, erheblich reduziert.

Es ist nicht nur DNS-Spezialisten möglich, einen Eintrag vorzunehmen. Gleichzeitig bleibt die Konsistenz der Konfiguration gewahrt. Dadurch hat sich die Turnaround-Zeit, das heisst, die Zeit bis zum Wirksamwerden eines neuen Eintrags, vom Stundenbereich in den Sekundenbereich verschoben. Da weniger Aufwand bei der Eintragung zu erbringen ist, hat auch die Bereitschaft zugenommen, Einträge sofort zu bearbeiten und nicht erst zu sammeln.

Es wird erwartet, dass ähnliche Ergebnisse auch bei den anderen Netzdiensten, insbesondere DHCP und Monitoring, zu erzielen sind.

## Literatur

- [Abb] Jonathan Abbey. Ganymede 1.0.12. <http://tools.arlut.utexas.edu/gash2/>.
- [CL02] Paul Albitz Cricket Liu. *DNS und BIND*. O'Reilly, 3 edition, 2002.
- [Inf] Infoblox. DNSone. <http://www.infoblox.com>.
- [Luc] Lucent. Vital QIP. <http://www.alcatel-lucent.com>.
- [Nor] Nortel. IP Address Domain Manager. <http://www.nortelnetworks.com>.
- [Pos79] J. Postel. Internet Name Server. IEN 116, USC/Information Sciences Institute, August 1979.