ERNST MORITZ ARNDT
UNIVERSITÄT GREIFSWALD

Wissen
lockt.
Seit 1456

OpenLDAP, das Kernstück einer kompletten IT-Infrastruktur

Universitätsrechenzentrum Greifswald
Gordon K. Grubert

25. Juni 2015

SLAC Secure Linux Administration Conference 2015

24.-26. Juni 2015 | Berlin

Die Ausgangslage im Jahre 2010

Folgende Verwaltungssysteme waren im Einsatz:

- Sun ONE Directory Server (iPlanet) für das Mailsystem (indirekt)
- separater Windows-Server für VPN & W-LAN
- Active Directory für zentrale Windows-Anmeldungen
- dezentrale Anmeldungen für alle weiteren zentralen Dienste (Webserver, ftp, etc.)

Die Ausgangslage im Jahre 2010

Folgende Verwaltungssysteme waren im Einsatz:

- Sun ONE Directory Server (iPlanet) für das Mailsystem (indirekt)
- separater Windows-Server für VPN & W-LAN
- Active Directory für zentrale Windows-Anmeldungen
- dezentrale Anmeldungen für alle weiteren zentralen Dienste (Webserver, ftp, etc.)

Aufgabe: **Erneuern Sie das Mailsystem!**

Die strategische Grundsatzentscheidung

Variante A

Beibehaltung der „IdMS-Infrastruktur“ und sture Erneuerung des Mailsystems

Die strategische Grundsatzentscheidung

Variante B

Vorherige Konsolidierung der „IdMS-Infrastruktur“ auf ein bereits vorhandenes IdMS

Die strategische Grundsatzentscheidung

Variante C

Vorherige Konsolidierung der „IdMS-Infrastruktur“ unter Verwendung eines komplett neuen IdMS

Die strategische Grundsatzentscheidung: Variante C

Entgegen aller Widerstände und Vorurteile bei OpenLDAP ...

- OpenLDAP ist zu langsam
- OpenLDAP ist zu kompliziert
- OpenLDAP ist zu instabil
- außerdem ist es Open Source $\Rightarrow$ das kann nicht gut sein!

...

Die strategische Grundsatzentscheidung: Variante C

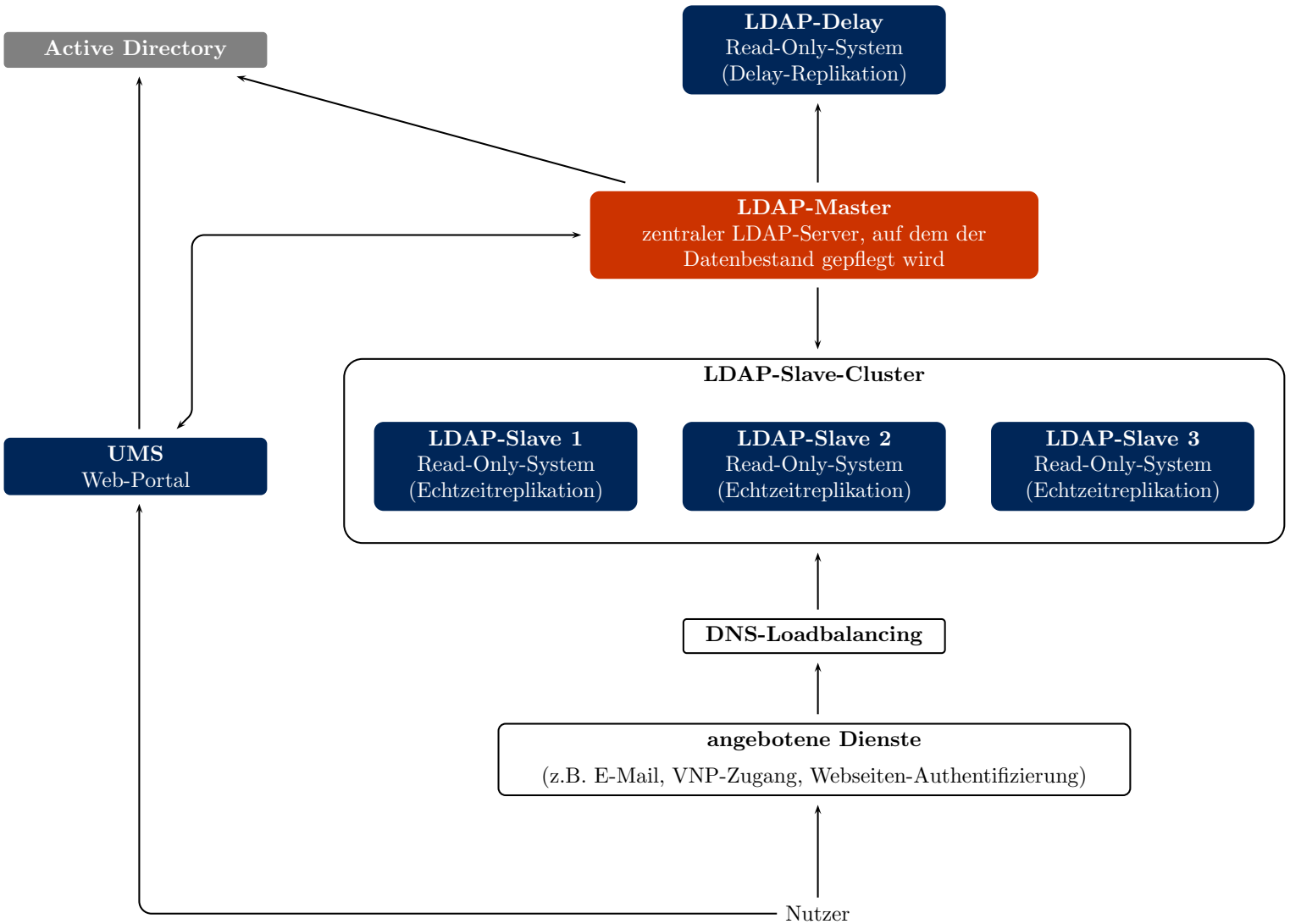
... wurde die Entscheidung für OpenLDAP getroffen, und das ist auch gut so!

- direkter Random-Lesezugriffs-Vergleich OpenLDAP vs. iPlanet: OpenLDAP bei gleicher Datenstruktur über 70% schneller
- direkter Random-Lesezugriffs-Vergleich OpenLDAP vs. iPlanet: OpenLDAP mit „logisch angepasster“ Datenstruktur über 30% schneller
- wer LDAP versteht, versteht auch OpenLDAP
- Open Source $\Rightarrow$ dies war bis heute nur von Vorteil

Randbedingungen für das Systemdesign

1. größtmögliche Verfügbarkeit für den Lesezugriff
2. Verfügbarkeitseinbußen beim Schreibzugriff sind akzeptabel
3. Einhaltung der sehr hoher Sicherheitsstandards des URZ
   (u.a. strikte Verschlüsselung aller Verbindungen)
4. maximale Leseperformance, da alle Systeme des URZ in Echtzeit auf das LDAP-System zugreifen sollen
5. Abbildung der Verwaltungsstruktur der Universität im LDAP-System
6. es müssen individuelle Anpassungen für die Bedürfnisse der Universität von Beginn an vorgesehen werden

Systemaufbau



Systemaufbau

Vorteile

- der Lesezugriff für alle IT-Systeme (und Nutzer) ist hoch performant und hochverfügbar
- Verteilung der IdMS-Daten auf 2 Standorte mit voneinander unabhängigen Storage-Backends
- komplettes Desaster Recovery des IdMS-Datenbestandes in weniger als 3 Minuten möglich
- im Ernstfall beträgt der Datenverlust idealerweise wenige Minuten, maximal 4 Stunden

Systemaufbau

Nachteile

- Datenänderungen nur auf dem singulären LDAP-Master-System
  ⇒ bewusste Entscheidung gegenüber den Risiken

  http://www.openldap.org/doc/admin24/replication.html
- Applikationen müssen RR-DNS unterstützen und einen automatischen Re-Connect einleiten können, wenn der LDAP-Server nicht mehr verfügbar ist
  (z.B. Probleme mit Tomcat bekannt)

Individuelle Anpassungen: Eigenes LDAP-Schema

- das Gesamtsystem ist nicht mehr weltweit „transportabel“
- ermöglicht extrem hohe Flexibilität
- unbedingt die Dokumentation beachten

http://www.openldap.org/doc/admin24/schema.html

- Verwendung einer OID mit eigener **P**rivate **E**nterprise **N**umber der IANA ist Pflicht
- Objektnamen sollten mit „x-“ beginnen

http://www.rfc-editor.org/rfc/rfc4520.txt

Individuelle Anpassungen: Eigenes LDAP-Schema

- das Gesamtsystem ist nicht mehr weltweit „transportabel“
- ermöglicht extrem hohe Flexibilität
- unbedingt die Dokumentation beachten

http://www.openldap.org/doc/admin24/schema.html

- Verwendung einer OID mit eigener **P**rivate **E**nterprise **N**umber der IANA ist Pflicht
- Objektnamen sollten mit „x-“ beginnen

http://www.rfc-editor.org/rfc/rfc4520.txt

Tip bei Änderungen des eigenen Schemas

Wenn das genutzte Schema geändert werden muss, unbedingt die Datenkonsistenz prüfen!

Individuelle Anpassungen: Eigenes LDAP-Schema (Auszug)

```
...
attributetype ( 1.3.6.1.4.1.34867.1.1 NAME 'UniHGW-HISNr'
    DESC 'Eindeutige HIS-Nummer der Hochschulangestellten'
    EQUALITY numericStringMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.36
    SINGLE-VALUE )
...
attributetype ( 1.3.6.1.4.1.34867.1.18 NAME 'UniHGW-ServiceAgreement'
    DESC 'Services, zu deren Nutzung der Nutzer explizit zugestimmt hat'
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
...
objectclass ( 1.3.6.1.4.1.34867.2.8 NAME 'UniHGW-Resource'
    DESC 'URZ-Ressource'
    AUXILIARY
    MUST ( UniHGW-RealUID $ UniHGW-EndOfValidity $ UniHGW-ResourceID $
           UniHGW-StartOfValidity $ UniHGW-Approval )
    MAY ( UniHGW-CleanupInformation $ description ) )
...
```

Kleiner Tip

Boolean-Werte müssen exakt „TRUE“ oder „FALSE“ sein!

https://www.ietf.org/rfc/rfc2252.txt (Abs. 6.4)

LDAP-Server konfigurieren

Allgemein

- Nutzung des config-Schemas
  - Speicherung der Konfiguration im LDAP selbst
  - Änderungen live möglich
  - Änderungen können an andere LDAP-Server repliziert werden

LDAP-Server konfigurieren

Allgemein

- Nutzung des config-Schemas
  - Speicherung der Konfiguration im LDAP selbst
  - Änderungen live möglich
  - Änderungen können an andere LDAP-Server repliziert werden
- wer all diese Dinge nicht benötigt, nutzt eine „klassische“ Konfigurationsdatei

LDAP-Server konfigurieren

Die Wahl des Backends entscheidet über Qualität eines LDAP-Systems

- Hierarchische Berkeley Database:
  „The hdb backend to slapd(8) is the recommended primary backend for a normal slapd database.“
  - muss speziell getuned werden, um gute Performance zu erreichen
  - Caching ist zwingend erforderlich
  - DB-Parameter nicht änderbar
- Lightning Memory-Mapped Database:
  „The mdb backend to slapd(8) is the upcoming primary backend for a normal slapd database.“
  - benötigt kein Caching
  - kein Tuning notwendig
  - es ist „nur“ eine maximale Größenangabe erforderlich

http://www.openldap.org/doc/admin24/backends.html

LDAP-Server konfigurieren

Die Wahl des Backends entscheidet über Qualität eines LDAP-Systems

- Hierarchische Berkeley Database:
  „The hdb backend to slapd(8) is the recommended primary backend for a normal slapd database.“
  - muss speziell getuned werden, um gute Performance zu erreichen
  - Caching ist zwingend erforderlich
  - DB-Parameter nicht änderbar
- Lightning Memory-Mapped Database:
  „The mdb backend to slapd(8) is the upcoming primary backend for a normal slapd database.“
  - benötigt kein Caching
  - kein Tuning notwendig
  - es ist „nur“ eine maximale Größenangabe erforderlich

http://www.openldap.org/doc/admin24/backends.html

⇒ Das Berkeley Backend ist ausgereift und absolut performant!

Grundkonfiguration (Auszug slapd.conf)

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/UniHGW.schema

# Log facility: local4
loglevel 64 256

moduleload back_hdb
moduleload back_monitor
moduleload pw-sha2
moduleload sssvlv

TLSCertificateFile cert-file.pem
TLSCertificateKeyFile key-file.pem
TLSCACertificateFile chachain-file.pem
TLSDHParamFile /etc/ldap/dh_2048.pem

backend hdb
disallow bind_anon
```

Berkeley Datenbank

```
database hdb
overlay sssvlv
suffix "dc=TLD"
rootdn "DN_OF_ROOT"
directory "/var/lib/ldap/TLD"

cachesize 10000
idlcachesize 30000

dbconfig set_data_dir data
dbconfig set_lg_dir log
dbconfig set_cachesize 0 100000000 0
dbconfig set_lk_max_objects 3000
dbconfig set_lk_max_locks 6000
dbconfig set_lk_max_lockers 3000

index objectClass,entryCSN,entryUUID eq
index uid,mail eq
overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100
lastmod on
checkpoint 512 30
```

Berkeley Datenbank

- separates data/log dir idealerweise auf verschiedenen Partitionen/Festplatten
- Indizes nur sinnvoll verwenden
  bdb_equality_candidates: (<ATTRIBUT-NAME>) not indexed
- Kontrolle der fest eingestellen DB-Parameter:
  db_stat -c -h <DB-HOME>
- „Problem“ der Datenbank Transaktionslogs:
  db_archive -d -h <DB-HOME>

Replikation: syncrepl

```
# Master
moduleload syncprov
overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100

# Slave
syncrepl rid=123
        provider=ldap://master
        starttls=critical
        type=refreshAndPersistent
        retry="60 60 300 +"
        searchbase="dc=TLD"
        scope=sub
        bindmethod=simple
        binddn="BINDDN"
        credentials=PASSWORD
```

Sicherheit ist auch maßgeblich Aufgabe des Clients: ldap.conf

```
TLS_CACERT      cachain.pem
TLS_REQCERT     demand
TLS_CRLFILE     crl-file.pem
TLS_CRLCHECK    all
```

Tip: Security Strength Factor (SSF) ist auch eine gute Idee

Direkt angeschlossene Dienste

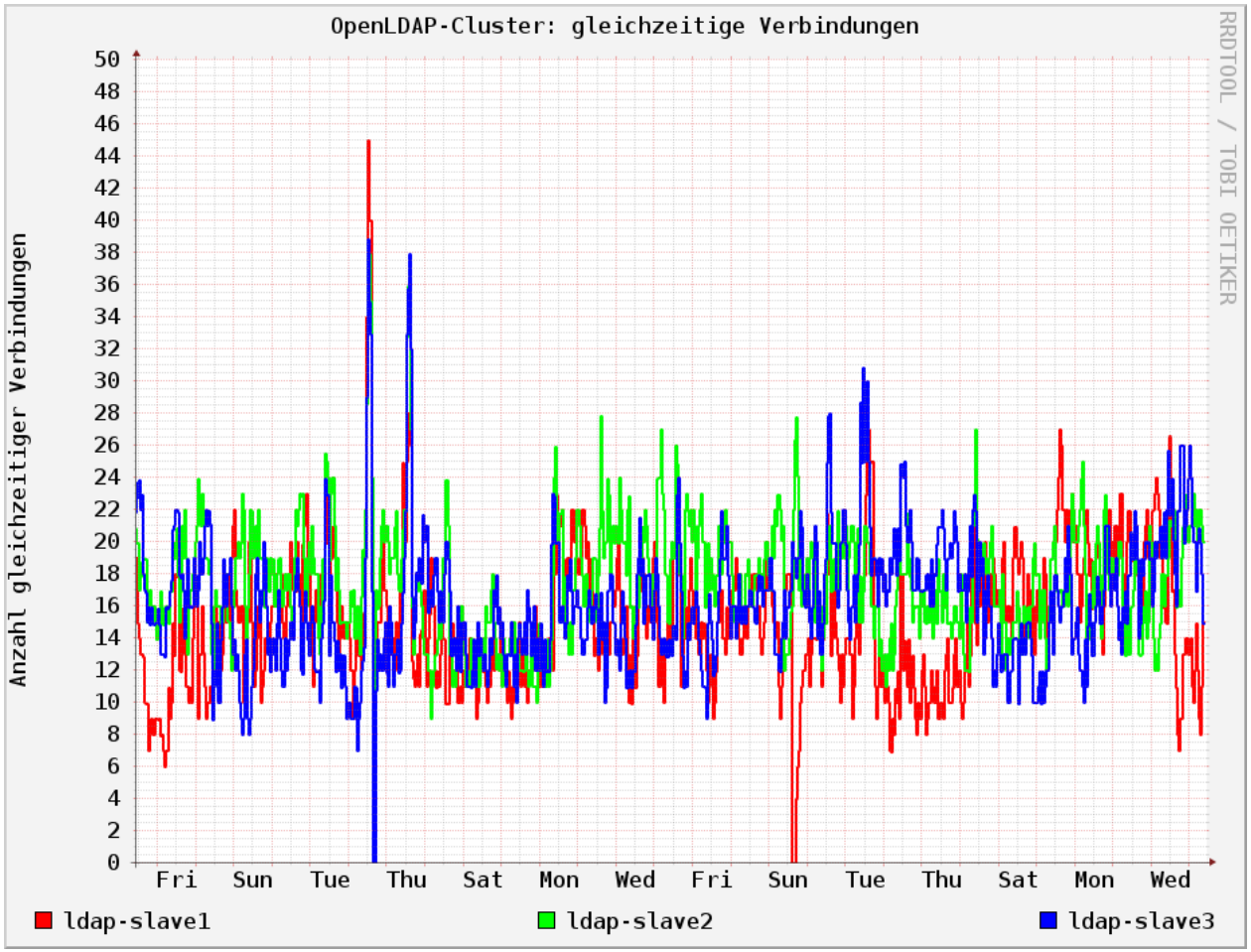
- E-Mailsystem
- W-LAN
- VPN
- Webseitenanmeldungen
- Shibboleth
- Netzwerkanmeldungen
- zentrale administrative Anmeldungen
- elektronisches Wählerverzeichnis

Direkt angeschlossene Dienste

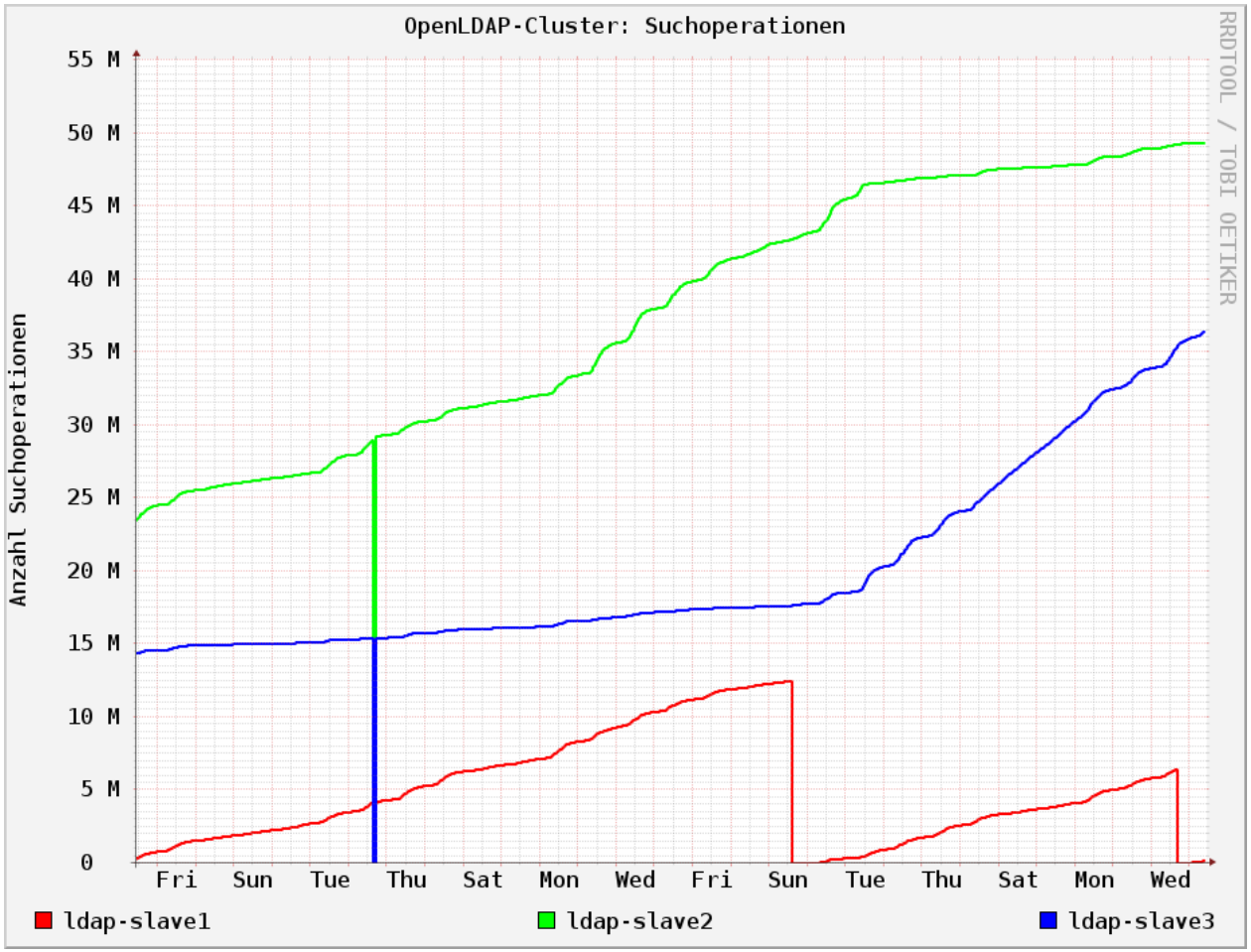
- E-Mailsystem
- W-LAN
- VPN
- Webseitenanmeldungen
- Shibboleth
- Netzwerkanmeldungen
- zentrale administrative Anmeldungen
- elektronisches Wählerverzeichnis

$\Rightarrow$ alle zentralen Dienste des URZ bis 2011 umgestellt

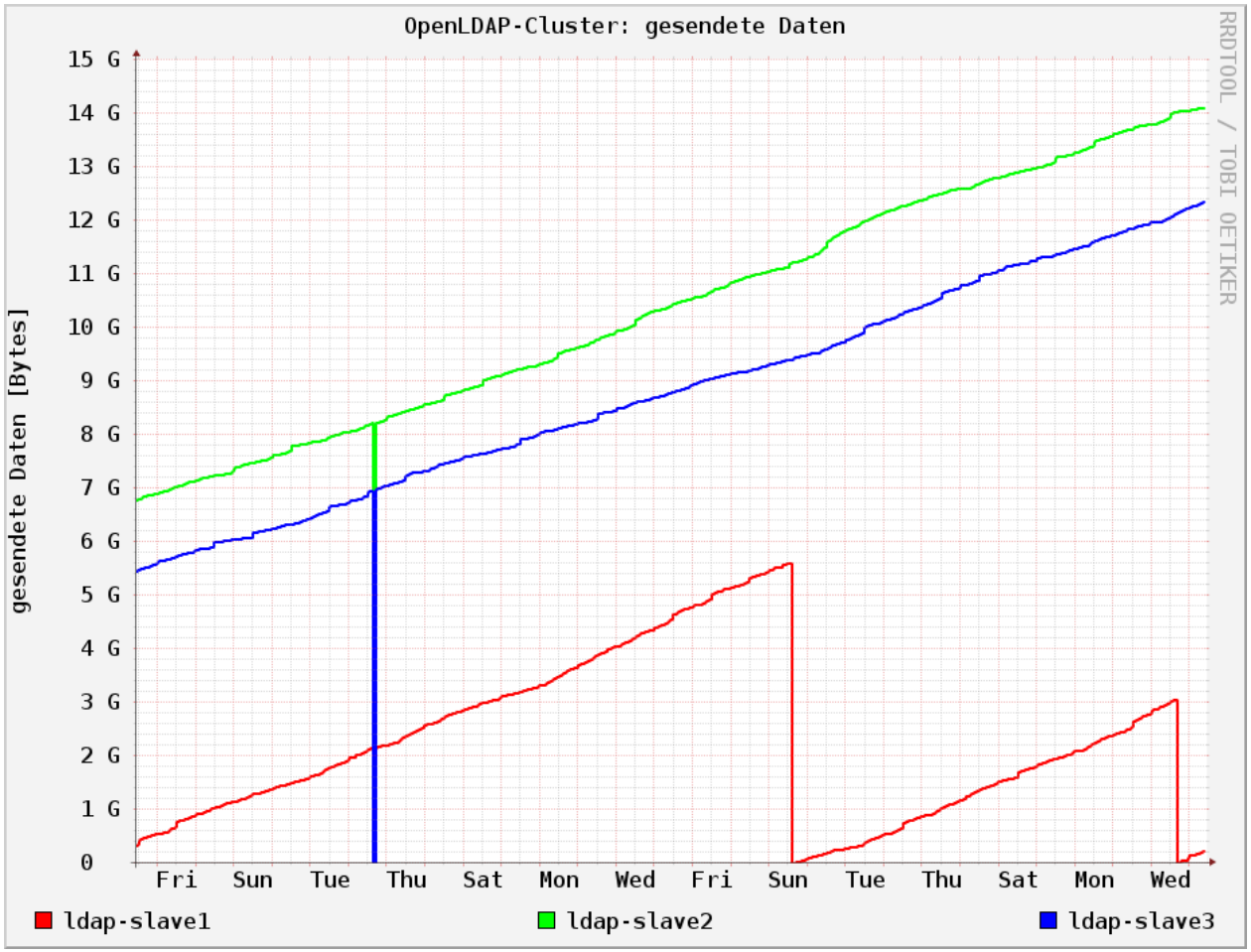
Statistiken



Statistiken



Statistiken



DHCP für ein Class B Netz – wie sinnvoll organisieren?

- strikte Umstellung aller Netzbereiche der Universität auf DHCP in den Jahren 2012/2013
- über 100 Netzsegmente $\Rightarrow$ es muss handhabbar bleiben
- feste IP-Adresszuweisungen sollten Personen zugeordnet werden können

DHCP für ein Class B Netz – wie sinnvoll organisieren?

Entscheidung für OpenLDAP als DHCP-Storage-Backend

- ISC-DHCP-Server bringt LDAP-Support mit; sehr gut dokumentiert
- zentraler Punkt für die DHCP-Konfiguration
- DHCP-Cluster sehr einfach zu verwalten
- Änderungen für Hostobjekte der DHCP-Konfiguration im LDAP-Backend erfordern keinen Reload/Restart des DHCP-Servers

DHCP-Server: Konfigurationsdatei

```
ldap-ssl              ldaps;
ldap-server           "ldapserver";
ldap-port             636;
ldap-tls-ca-file      "cachain.pem";
ldap-tls-reqcert      demand;
ldap-tls-crlcheck     all;
ldap-username         "BindDN";
ldap-password         "BindPW";
ldap-base-dn          "dc=DHCP,...,dc=TLD";
ldap-method           dynamic;
ldap-debug-file       "/var/lib/dhcp/dhcpd_startup.conf";
```

Anpassung des OpenLDAP-Servers

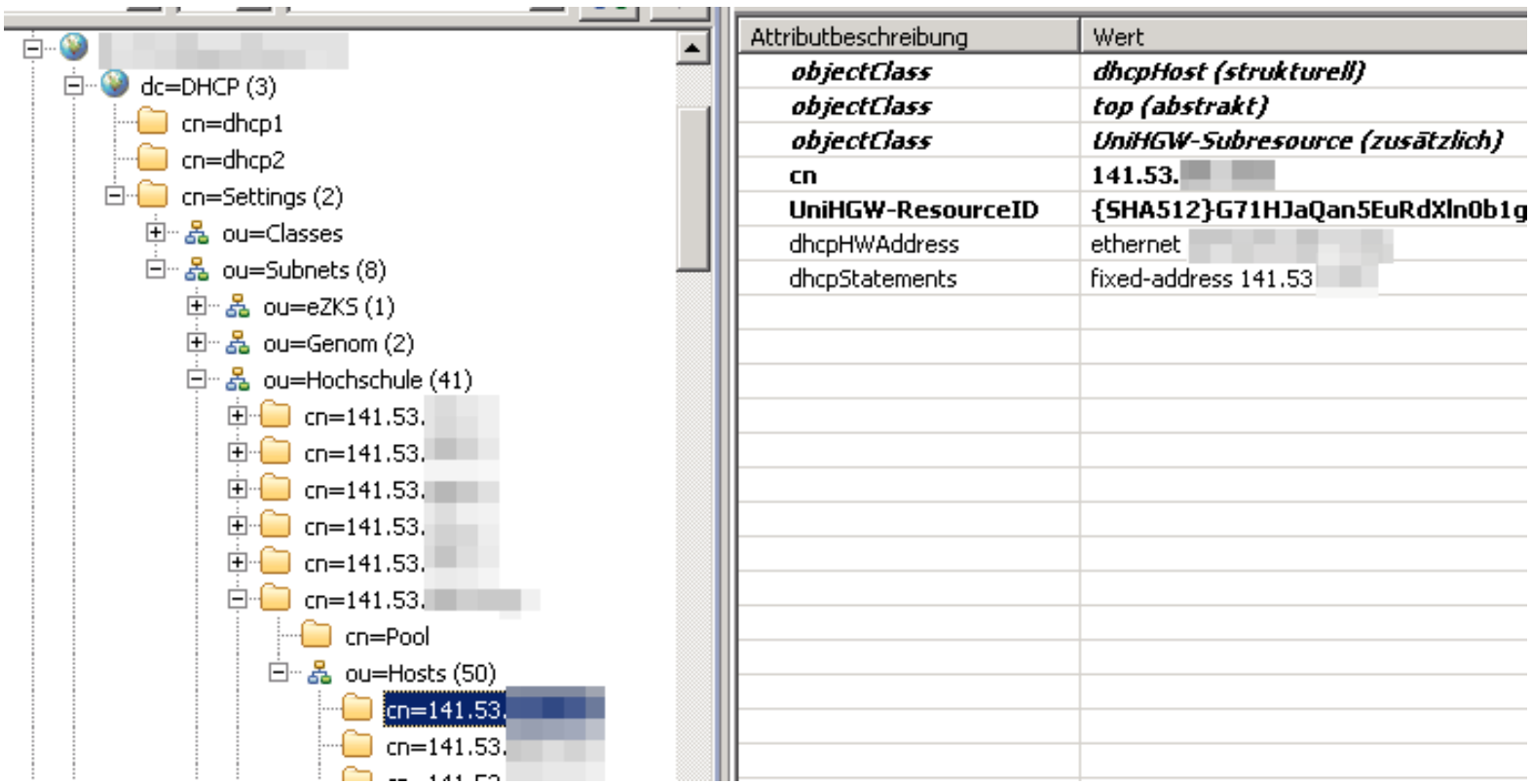
```
include /etc/ldap/schema/dhcp.schema
```

⇒ Schema-Datei wird beim ISC DHCP-Server mitgeliefert

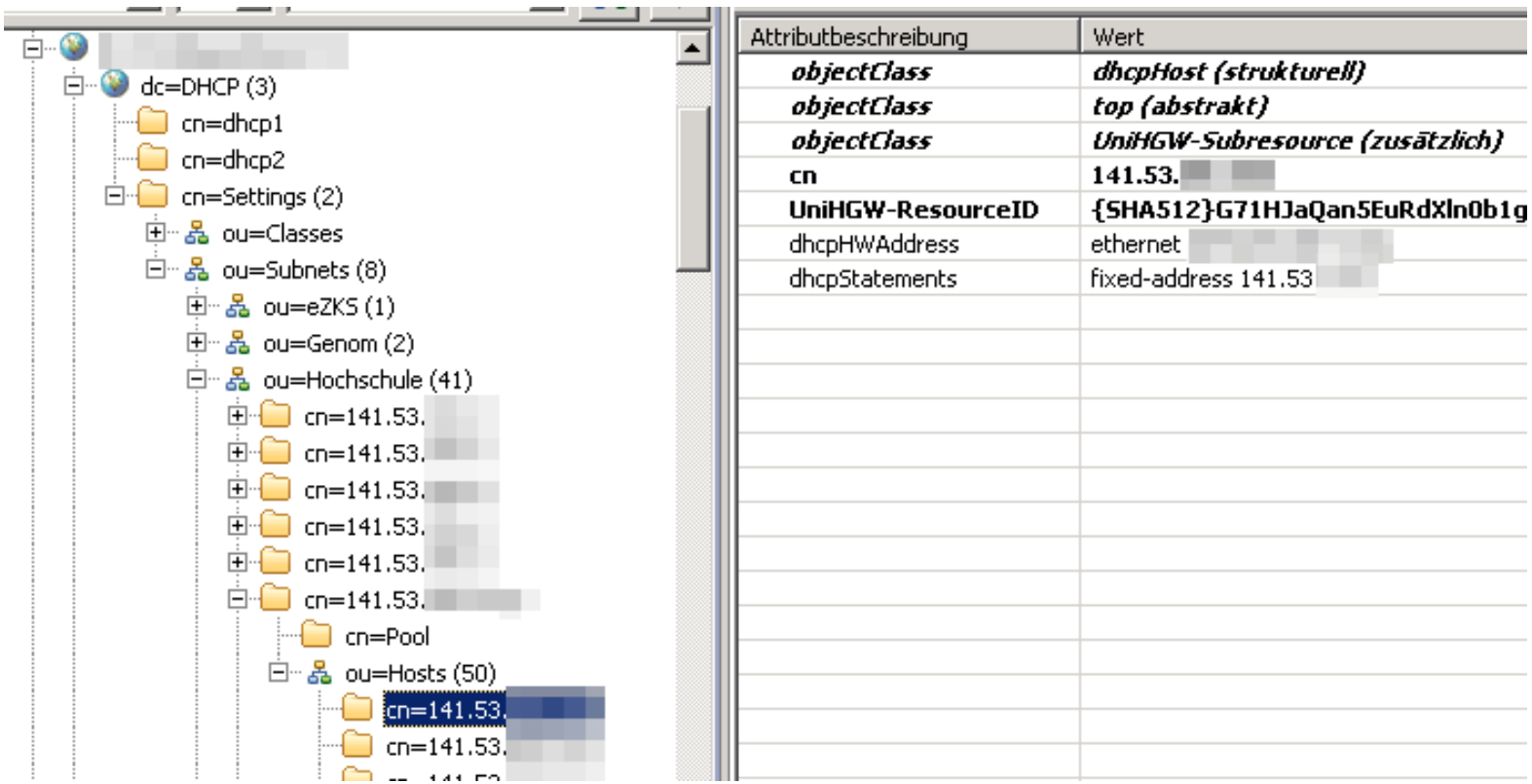
DHCP-Konfiguration in LDAP-Syntax

1. globaler Konfigurationsteil
2. Definition der beiden DHCP-Server für den Failover-Cluster
   $\Rightarrow$ Betrieb im dynamischen Modus
3. Definition eines Subnetzbereiches
   - Subnetz-Parameter
   - dynamische Pools
   - feste IP-Adressreservierungen

DHCP-Konfiguration in LDAP-Syntax



DHCP-Konfiguration in LDAP-Syntax



- beliebige Strukturierung im LDAP-Baum möglich
- „nur“ Objektklassen **dhcp*** haben bzgl. der Strukturierung einen direkten Einfluss auf die DHCP-Serverkonfiguration

DHCP-Konfiguration in LDAP-Syntax: Global

```
dn: cn=Settings,dc=DHCP,...,dc=TLD
objectClass: dhcpService
objectClass: top
cn: Settings
dhcpOption: domain-name "uni-greifswald.de"
dhcpOption: ntp-servers 141.53.x1.y1,141.53.x2.y2
dhcpOption: domain-name-servers 141.53.x1.y1,141.53.x2.y2
dhcpPrimaryDN: cn=dhcp1,dc=DHCP,...,dc=TLD
dhcpSecondaryDN: cn=dhcp2,dc=DHCP,...,dc=TLD
dhcpStatements: ping-check true
dhcpStatements: one-lease-per-client true
dhcpStatements: deny bootp
dhcpStatements: authoritative
dhcpStatements: ignore client-updates
dhcpStatements: log-facility local5
dhcpStatements: ddns-update-style none
dhcpStatements: max-lease-time 86400
dhcpStatements: default-lease-time 86400
```

DHCP-Konfiguration in LDAP-Syntax: Failover

Primärer DHCP-Server

```
dn: cn=dhcp1,dc=DHCP,...,dc=TLD
objectClass: dhcpServer
objectClass: top
cn: dhcp1
dhcpServiceDN: cn=Settings,dc=DHCP,...,dc=TLD
dhcpStatements: failover peer "dhcp-failover" { primary; address 141.53.x1.y1
 ; port 519; peer address 141.53.x2.y2; peer port 520; max-response-delay 30;
  max-unacked-updates 10; load balance max seconds 3; mclt 3600; split 128;
 }
```

Sekundärer DHCP-Server

```
dn: cn=dhcp2,dc=DHCP,...,dc=TLD
objectClass: dhcpServer
objectClass: top
cn: dhcp2
dhcpServiceDN: cn=Settings,dc=DHCP,...,dc=TLD
dhcpStatements: failover peer "dhcp-failover" { secondary; address 141.53.x2
 .y2; port 520; peer address 141.53.x1.y1; peer port 519; max-response-delay
  30; max-unacked-updates 10; load balance max seconds 3; }
```

Der CN des DHCP-Server-Objektes muss mit dem eigentlichen Hostnamen des DHCP-Servers übereinstimmen.

DHCP-Konfiguration in LDAP-Syntax: Subnetz

Allgemeine Subnetzkonfiguration

```
dn: cn=141.53.00x.0,ou=Hochschule,ou=Subnets,cn=Settings,dc=DHCP,...,dc=TLD
objectClass: dhcpOptions
objectClass: dhcpSubnet
objectClass: top
cn: 141.53.00x.0
dhcpNetMask: 24
dhcpOption: subnet-mask 255.255.255.0
dhcpOption: routers 141.53.x.1
dhcpOption: domain-name "rz.uni-greifswald.de"
dhcpOption: broadcast-address 141.53.x.255
```

Definition eines dynamischen Adresspools

```
dn: cn=Pool,cn=141.53.00x.0,ou=Hochschule,ou=Subnets,cn=Settings,dc=DHCP,...
 ,dc=TLD
objectClass: dhcpPool
objectClass: top
cn: Pool
dhcpRange: 141.53.x.103 141.53.x.190
dhcpStatements: deny dynamic bootp clients
dhcpStatements: failover peer "dhcp-failover"
```

DHCP: Anmerkungen

- Attribut dhcpComment sollte nur ASCII-Zeichen enthalten
- im dynamischen Modus wird die Startup-Konfiguration sporadisch ausgenullt (Bug!)
  - neben Debugzwecken wird diese Datei auch zum Monitoring der dynamischen Adresspools benötigt
  - Problem lässt sich über kleinen Workaround umgehen

    (Sicherheitskopie erstellen, wenn Datei nicht Null ist)
- Inhalt der dhcpd.leases wird aktuell nicht im LDAP hinterlegt und wegen diverser technischer Gründe auch für die Zukunft nicht geplant

DNS: Wie am besten machen?

Was bisher geschah ...

- drei Primary- / Secondary-DNS-Päarchen zur Abbildung diverser Sichten auf das DNS
- klassisches Konstrukt mit Bind und Zonenfiles

DNS: Wie am besten machen?

... und überdacht wurde

- verschiedene Sichten auf das DNS lassen sich mittels Views realisieren
- ein zentraler Datenstamm ist sehr einfach zu verwalten
- LDAP-Replikation bietet merkliche Vorteile gegenüber DNS-Zonentransfers

Alternative DNS Servers, Jan-Piet Mens (ISBN 978-0-9544529-9-5)

DNS: Wie am besten machen?

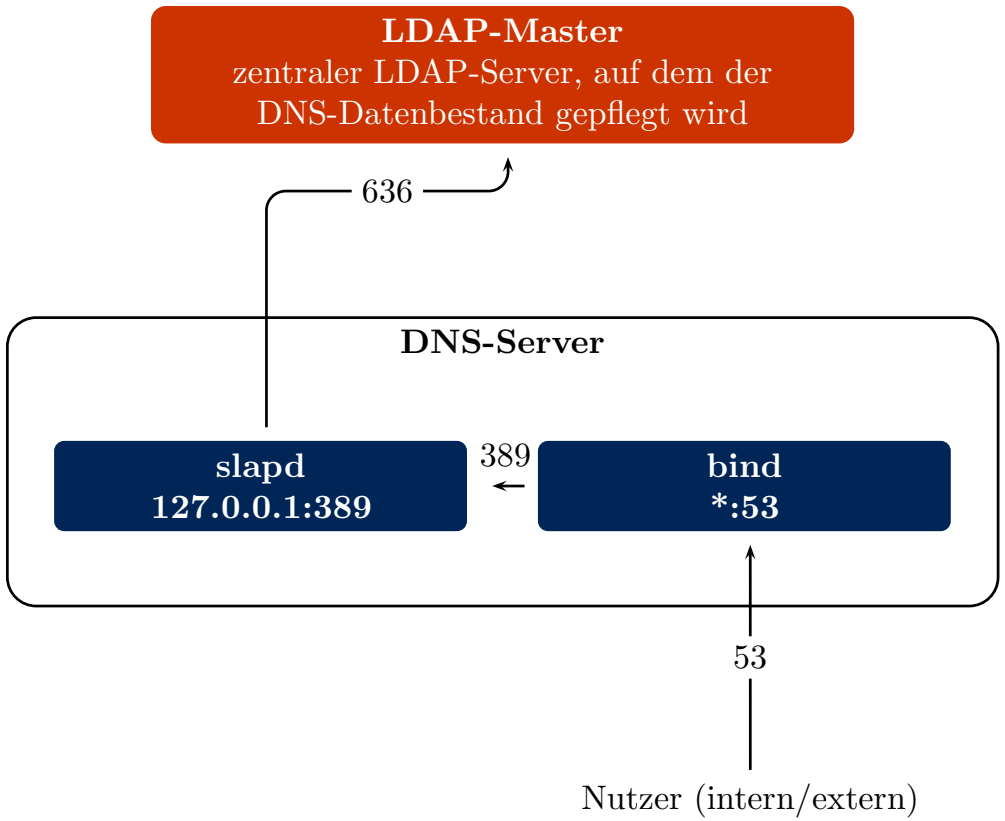
2014: Entscheidung für OpenLDAP als Storage-Backend

- Bind bringt via DLZ (Dynamically Loadable Zones) ein LDAP-Backend mit (ist seit Debian 7 in den Standardpaketen nicht mehr einkompiliert)

  http://bind-dlz.sourceforge.net/

- zentraler Punkt für die DNS-Konfiguration
- Änderung der DNS-Einträge sind ohne Zonenreload verfügbar
- Fehlkonfiguration in einer Zone beeinträchtigt nicht die ganze Zone

DNS: Systemaufbau



DNS: Systemaufbau

DNS-Server werden als LDAP-Slave betrieben

- „Henne-Ei-Problem“, wenn DNS-Server seine DNS-Daten sicher via TLS aus LDAP beziehen soll
- DNS-Anfragen erzeugen im Datennetz die meisten Sessions
  $\Rightarrow$ sehr hohe Last am LDAP zu erwarten
- mit lokalem LDAP-Server kann zwischen Bind und OpenLDAP auf Verschlüsselung verzichtet werden
- lokaler anonymer LDAP-Zugriff möglich
- lokaler OpenLDAP-Server kann individuell justiert werden

Bind: Vorbereitungen

- DLZ beim configure aktivieren:

  `--with-dlz-ldap=yes`
- klassische Bind-Konfiguration

Bind: Zonenkonfiguration – ACLs

```
acl "intern" {
    141.53.0.0/16;
    10.0.0.0/8;
    127.0.0.1;
};

acl "extern" {
    any;
};
```

⇒ beliebige ACLs definierbar

⇒ analog sind die Views erweiterbar

Bind: Zonenkonfiguration – Interner View

```
view "intern" IN {
    match-clients { "intern"; };

    dlz "ldap zone" {
        database "ldap 1
        v3 simple {} {} {127.0.0.1}
        ldap:///dlzZoneName=$zone$,dc=DNS,...,dc=TLD???objectclass=dlzZone
        ldap:///dlzHostName=$record$,dlzZoneName=$zone$,dc=DNS,...,dc=TLD?\
 dlzTTL,dlzType,dlzPreference,dlzData,dlzIPAddr,dlzPrimaryNS,dlzAdminEmail,dlzSerial,\
 dlzRefresh,dlzRetry,dlzExpire,dlzMinimum?sub?\
 (&(objectclass=dlzAbstractRecord)(!(dlzType=soa))(dlzACL=intern))
        ldap:///dlzHostName=@,dlzZoneName=$zone$,dc=DNS,...,dc=TLD?\
 dlzTTL,dlzType,dlzData,dlzPrimaryNS,dlzAdminEmail,dlzSerial,dlzRefresh,dlzRetry,\
 dlzExpire,dlzMinimum?sub?\
 (&(objectclass=dlzAbstractRecord)(dlzType=soa)(dlzACL=intern))
        ldap:///dlzZoneName=$zone$,dc=DNS,...,dc=TLD?\
 dlzTTL,dlzType,dlzHostName,dlzPreference,dlzData,dlzIPAddr,dlzPrimaryNS,dlzAdminEmail,\
 dlzSerial,dlzRefresh,dlzRetry,dlzExpire,dlzMinimum?sub?\
 (&(objectclass=dlzAbstractRecord)(!(dlzType=soa))(dlzACL=intern))
        ldap:///dlzZoneName=$zone$,dc=DNS,...,dc=TLD??sub?\
 (&(objectclass=dlzXFR)(dlzIPAddr=$client$))";
    };

    include "/etc/bind/named.conf.zones.rootservers";
    recursion yes;
};
```

http://bind-dlz.sourceforge.net/ldap_driver.html

Bind: Zonenkonfiguration – Externer View

```
view "extern" IN {
    match-clients { "extern"; };

    dlz "ldap zone" {
        database "ldap 1
        v3 simple {} {} {127.0.0.1}
        ldap:///dlzZoneName=$zone$,dc=DNS,...,dc=TLD???objectclass=dlzZone
        ldap:///dlzHostName=$record$,dlzZoneName=$zone$,dc=DNS,...,dc=TLD?\
 dlzTTL,dlzType,dlzPreference,dlzData,dlzIPAddr,dlzPrimaryNS,dlzAdminEmail,dlzSerial,\
 dlzRefresh,dlzRetry,dlzExpire,dlzMinimum?sub?\
 (&(objectclass=dlzAbstractRecord)(!(dlzType=soa))(dlzACL=extern))
        ldap:///dlzHostName=@,dlzZoneName=$zone$,dc=DNS,...,dc=TLD?\
 dlzTTL,dlzType,dlzData,dlzPrimaryNS,dlzAdminEmail,dlzSerial,dlzRefresh,dlzRetry,\
 dlzExpire,dlzMinimum?sub?\
 (&(objectclass=dlzAbstractRecord)(dlzType=soa)(dlzACL=extern))";
    };

    recursion no;
};
```

Anpassung des OpenLDAP-Servers

```
include /etc/ldap/schema/dlz.schema
```

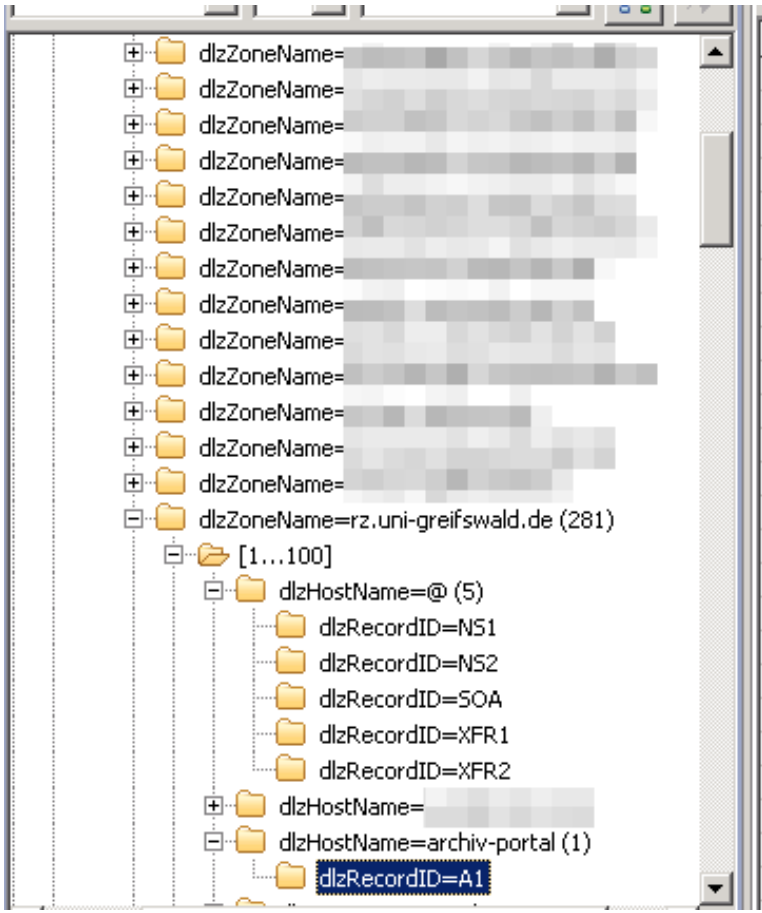
Individuelle Anpassungen des DLZ-Schemas

- Attribute **dlzACL** definiert, um verschiedene Rechte in Views über LDAP-Abfragen zu realisieren
- Objektklasse **dlzSRVRecord** definiert, um SRV-Einträge zu realisieren
- Objektklasse **dlzLOCRecord** definiert, um LOC-Einträge zu realisieren
- Attribut **dlzHostName** in vielen Klassen zusätzlich als **MUST** definiert, da dieser Wert für einen Zonentransfer zwingend erforderlich ist (das Tool Sleuth basiert auf Zonentransfers)

DNS-Konfiguration in LDAP-Synatx

- Zonenobjekt
  - darunter ein Hostobjekt
    - darunter wiederum eigentliche Hosteinträge

DNS-Konfiguration in LDAP-Synatx



Attributbeschreibung	Wert
objectClass	***dlzAbstractRecord (abstrakt)***
objectClass	***dlzARecord (strukturell)***
objectClass	***top (abstrakt)***
dlzHostName	**archiv-portal**
dlzIPAddr	**141.53.15.40**
dlzRecordID	**A1**
dlzTTL	**3600**
dlzType	**A**
dlzACL	extern
dlzACL	intern

DNS-Konfiguration in LDAP-Synatx

Zonenobjekt

```
dn: dlzZoneName=rz.uni-greifswald.de,dc=DNS,...,dc=TLD
objectClass: dlzZone
objectClass: top
dlzZoneName: rz.uni-greifswald.de
```

Hostobjekt

```
dn: dlzHostName=archiv-portal,dlzZoneName=rz.uni-greifswald.de,dc=DNS,...,dc
 =TLD
objectClass: UniHGW-Subresource
objectClass: dlzHost
objectClass: top
dlzHostName: archiv-portal
UniHGW-ResourceID: {SHA512}NkbrtQOKnt9WvTkAjAfahxPez/4imUxA9aXrHbSyYYdeEzsps
 Hb8qRyUV/OUlgfjvQx5FNG+C47VwShP11XOmw==
```

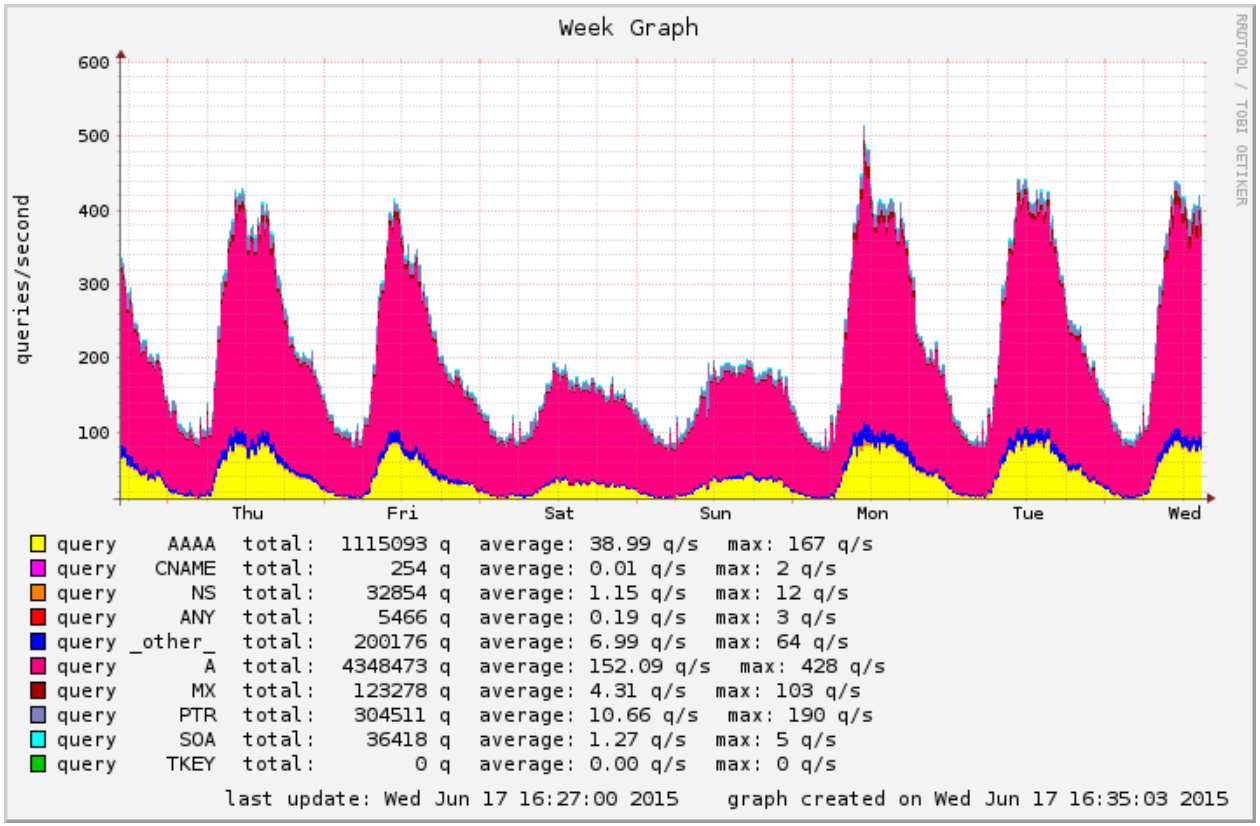
DNS: Anmerkungen

- erste Tests erfolgten mit hdb Backend (bisherige Optimierung)
  $\Rightarrow$ viele sporadische Abstürze verzeichnet
- Wechsel auf lmdb (keine Optimierung notwendig)
  $\Rightarrow$ extrem langsam & permanente Abstürze
- nach 2 Tagen Wechsel zurück auf hdb mit verbesserter Optimierung für „DNS-LDAP-Server“
  $\Rightarrow$ die Welt ist in Ordnung

DNS: Anmerkungen

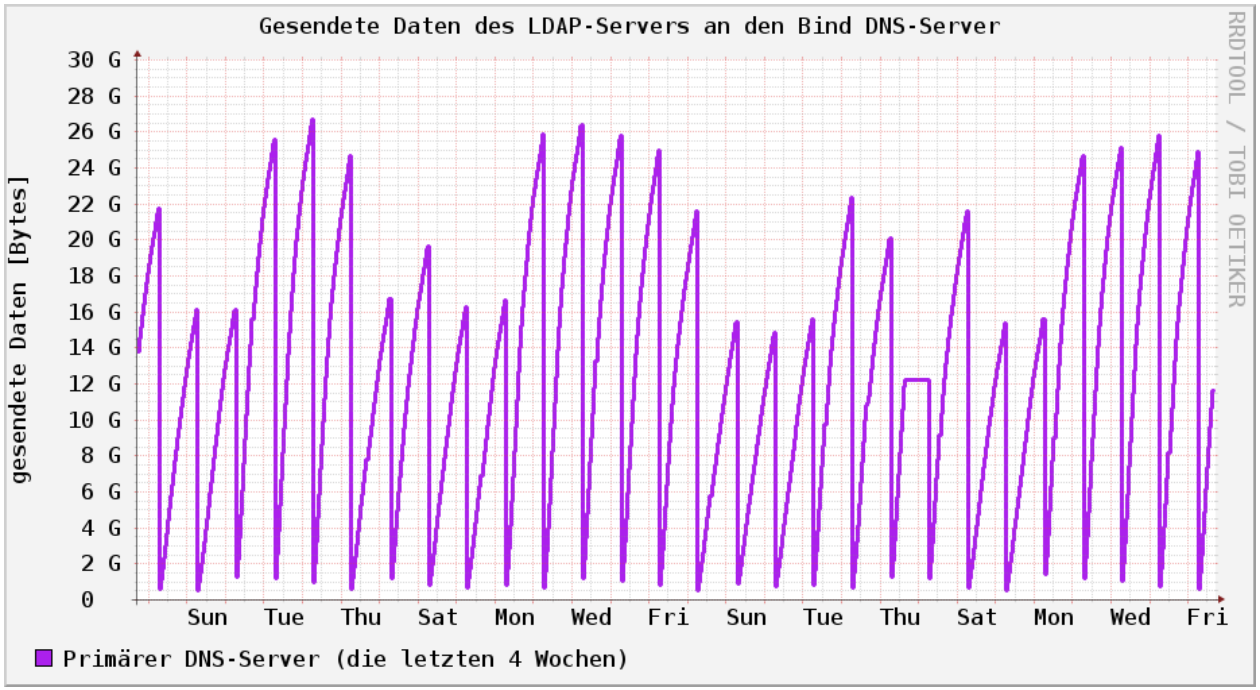
- erste Tests erfolgten mit hdb Backend (bisherige Optimierung)
  $\Rightarrow$ viele sporadische Abstürze verzeichnet
- Wechsel auf lmdb (keine Optimierung notwendig)
  $\Rightarrow$ extrem langsam & permanente Abstürze
- nach 2 Tagen Wechsel zurück auf hdb mit verbesserter Optimierung für „DNS-LDAP-Server“
  $\Rightarrow$ die Welt ist in Ordnung
- DNSSec
  - macht die Strategie des Echtzeit-DNS „kaputt“
  - ist sicherlich möglich $\Rightarrow$ wir werden es angehen

DNS: Anmerkungen



⇒ LDAP-basiertes Echtzeit-DNS funktioniert für unsere Anforderungen

DNS: Anmerkungen



⇒ OpenLDAP leistet hohe Datendurchsatzanforderungen

I have a dream

Selbstbereinigende IT-Infrastruktur

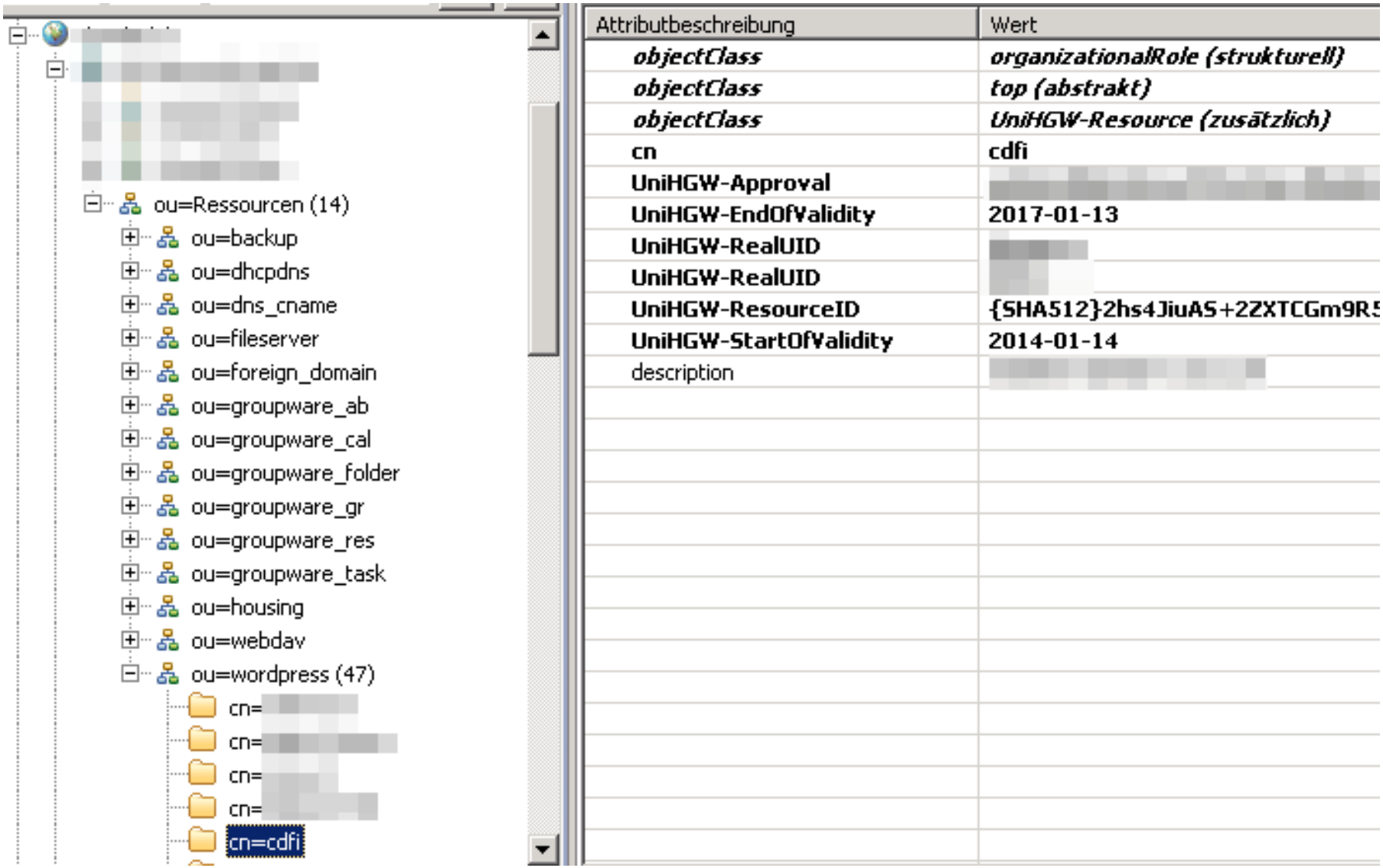
- Zuordnung aller „Ressourcen“ zu Nutzern
- Nutzer werden im IdMS automatisch bereinigt
  $\Rightarrow$ verknüpfte Ressourcen automatisch mit bereinigt
  - Daten im OpenLDAP selbst werden direkt gelöscht
  - Daten in externen Systemen müssen durch Systembetreuer gelöscht werden, jedoch entscheidet das OpenLDAP-System, welche Daten gelöscht werden
- durch wenige Metadaten kann sehr effizienter Workflow etabliert werden

Verfügbare Ressourcen

Die Wichtigsten

- IP-Adressreservierungen (inkl. DNS-Einträgen)
- DNS CNAMEs
- Webspace
- Fileserver-Space
- groupwareinterne Ressourcen
- zentrales Backupsystem

Verfügbare Ressourcen



Metadaten

- mindestens **2** Verantwortliche
- Genehmiger
- Datum der Genehmigung und des Ressourcenablaufs
- Beschreibung
- eindeutige ID für eine Ressource
- Informationen über Ablaufinformationen

Workflow (für Nutzer)

- Antrag im Online-System stellen
  (Nutzer benennt einen Genehmiger)
- Genehmiger genehmigt Antrag
- URZ genehmigt Antrag technisch und richtet Ressource ein

Workflow (für Nutzer)

- Antrag im Online-System stellen
  (Nutzer benennt einen Genehmiger)
- Genehmiger genehmigt Antrag
- URZ genehmigt Antrag technisch und richtet Ressource ein

⇒ Implementierung der Weboberfläche ist eigentlicher Aufwand
(wurde parallel zum Dienstausbau durchgeführt)

Nutzerportal

Ressourcenlöschung

Gründe für die Löschung

1. Ressource hat Ablaufdatum überschritten
2. Anzahl der Verantwortlichen auf 1 gesunken

Ressourcenlöschung

Vollautomatisierter Workflow

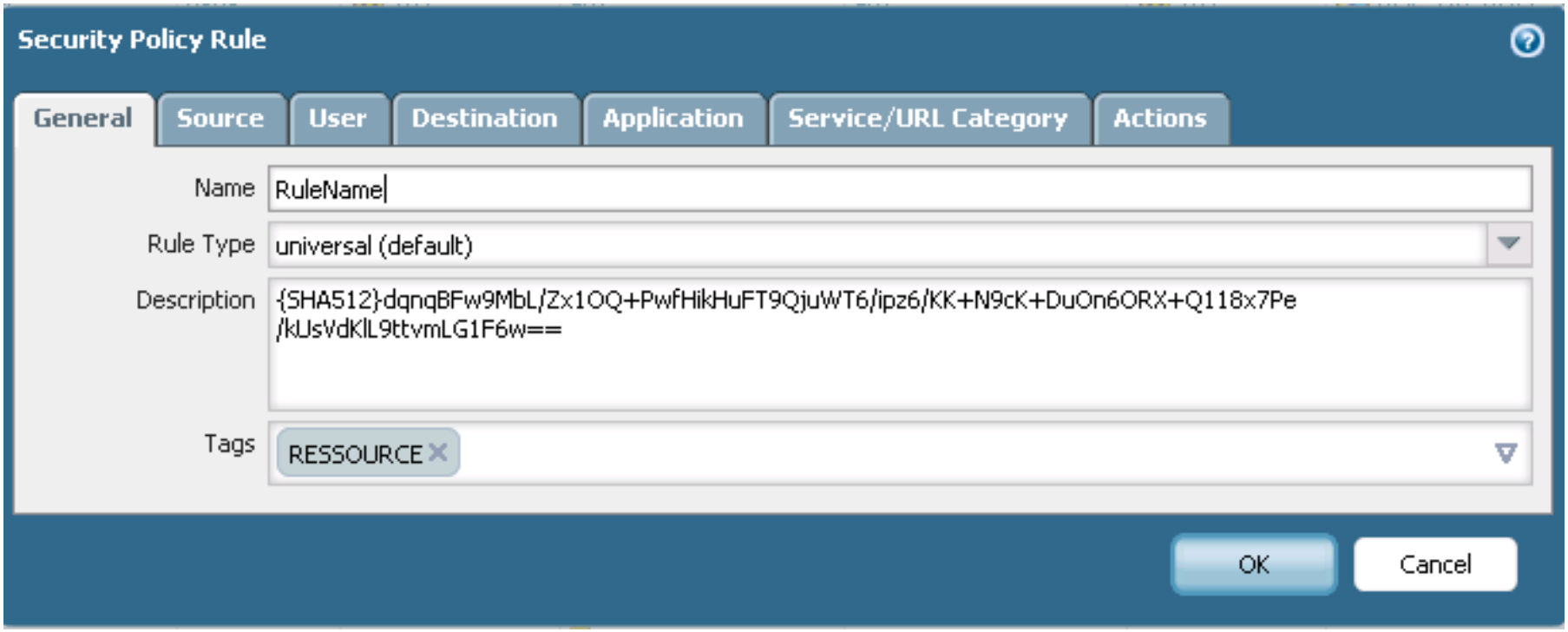
- alle Verantwortliche erhalten **SMIME-signierte** E-Mails
- 3 E-Mails im Abstand von 4 Wochen
- Ablaufdatum + 12 Wochen überschritten und 3 Ablaufinformationen verschickt
  $\Rightarrow$ Löschung des Metadatensatzes im LDAP
- Löschung aller zugehörigen Daten im LDAP
  (Verantwortung beim OpenLDAP-System)
- Löschung aller Daten in externen Systemen
  (Verantwortung beim jeweiligen Systembetreiber)

Neue Ressourcen

Demnächst sind dabei

- Mailinglisten des Sympa-Systems
- Regelwerk der zentralen Firewall

Neue Ressourcen



Nutzerfeedback

Negativ

- „Warum läuft die Webseite ab? Die brauche ich für immer!“
- Nutzer ignorieren 3 automatische Warnungen vor der Löschung
  $\Rightarrow$ das hat wirklich Konsequenzen

Nutzerfeedback

Positiv

- rein elektronischer Workflow
- schnellere Bearbeitung durch das URZ möglich
- gute Übersicht im Nutzerportal
- selbständige Löschung möglich

Nutzerfeedback

Positiv

- rein elektronischer Workflow
- schnellere Bearbeitung durch das URZ möglich
- gute Übersicht im Nutzerportal
- selbständige Löschung möglich

$\Rightarrow$ Für uns überwiegen die Vorteile!

Fazit

1. Man kann ohne OpenLDAP auskommen, muss es aber nicht.
2. OpenLDAP hat uns durch
   - Hochverfügbarkeit
   - sehr gute Performance
   - extreme Flexibilität

   überzeugt.

Fazit

1. Man kann ohne OpenLDAP auskommen, muss es aber nicht.
2. OpenLDAP hat uns durch
   - Hochverfügbarkeit
   - sehr gute Performance
   - extreme Flexibilität

   überzeugt.
3. Sie wurden hoffentlich durch die komplett selbstbereinigende, karteileichenfreie IT-Infrastruktur überzeugt, die mit OpenLDAP realisiert wurde.