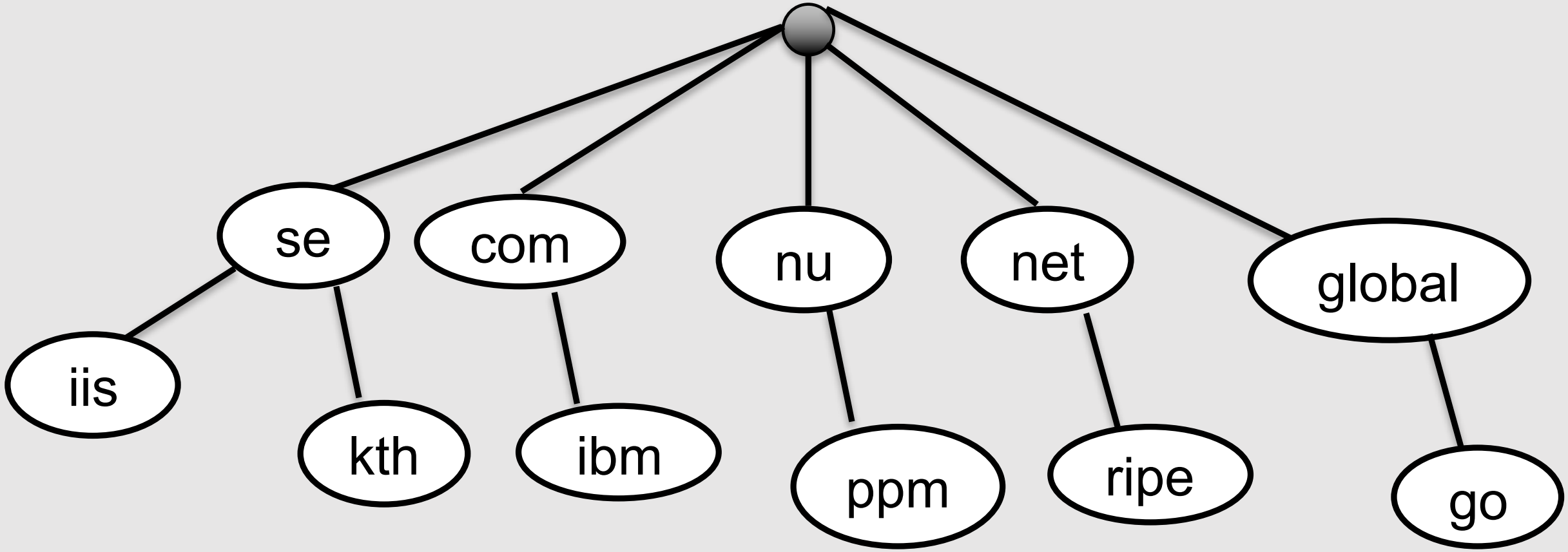

se
com
nu
net
global
iis
kth
ibm
ppm
ripe
go

INTERNET STIFTELSEN

Internets domännamnssystem*

Föreläsning FL13, HT2023

Mats Dufberg

* Se “Internets domännamnssystem”

Innehåll

▶Wildcard "*" i domännamn

[Till Innehåll]

Enkel zonfil

Antag att följande är en komplett zonfil.

```
$ORIGIN exempel.se.
@                       SOA     (...)
                        NS      ns1.exempel.com.
www                     A       192.0.2.20
```

De namn som finns i zonen är "exempel.se" (NS och SOA) och "www.exempel.se" (A).

Om namnet inte finns → NXDOMAIN.

Namnet finns, men inte posttypen → NODATA.

Exempel 1 – zonfil med "*"

Om första ”label” i ”owner name” i en zonefil är en ”*” så har det en speciell betydelse.

```
$ORIGIN exempel.se.
@                       SOA     (...)
                        NS      ns1.exempel.com.
www                     A       192.0.2.20          ; Egen post, matchar inte wildcard
*                       A       192.0.2.100         ; Wildcard för A-post
```

Förutom namnen på förra bilden så har följande namn tillkommit i zonen:

- "*.exempel.se" (med A-post)
- Vilket namn som helst under "exempel.se" (med A-post)

Exempel 1 – namn från "*"

Fråga efter vilket namn som helst under exempel.se kommer att matcha namnet "*.exempel.se" utom just fråga efter ”www.exempel.se” som har en egen post. Det motsvarar följande svar om vi har frågat efter A-post:

```
www.exempel.se.          A      192.0.2.20  ; Egen post, inte wildcard
a.exempel.se.            A      192.0.2.100 ; Matchar wildcard
a.b.exempel.se.          A      192.0.2.100 ; Matchar wildcard
a.b.c.exempel.se.        A      192.0.2.100 ; Matchar wildcard
```

Fråga efter annan posttyp → NODATA, inte NXDOMAIN.

Exempel 2 – zonfil med "*"

Bara en A-post, med "*".

```
$ORIGIN exempel.se.
@                       SOA     (...)
                        NS      ns1.exempel.com.
*.info                  A       192.0.2.100         ; Wildcard för A-post
```

Följande namn får vi svar på från zonen:

- exempel.se (NS, SOA)
- info.exempel.se (bara NODATA)
- *.info.exempel.se (A)
- Vilket namn som helst under info.exempel.se, t.ex. kth.info.exempel.se (A)

Exempel 2 – uppslagningar mot zonen

Fråga efter vilket namn som helst under info.exempel.se kommer att matcha namnet "*.info.exempel.se", men alla namn finns inte.

```
www.exempel.se.          A?     →  NXDOMAIN
mail.exempel.se.         A?     →  NXDOMAIN
info.exempel.se.         A?     →  NODATA oavsett posttyp
www.info.exempel.se.     A?     →  192.0.2.100 ; Matchar wildcard
a.info.exempel.se.       A?     →  192.0.2.100 ; Matchar wildcard
a.b.c.info.exempel.se.   A?     →  192.0.2.100 ; Matchar wildcard
www.info.exempel.se.     AAAA?  →  NODATA, matchar wildcard-namn, men inte posttyp
a.info.exempel.se.       AAAA?  →  NODATA, matchar wildcard-namn, men inte posttyp
```

Användning av wildcard

Om man har många tjänster på en webbserver, t.ex.

tomat.exempel.se
gurka.exempel.se
potatis.exempel.se

och vill kunna lägga till nya genom bara uppdatering i webbservern utan att ändra dns så kan hantera det med en wildcard-post.

Alla okända eller nya namn under exempel.se kommer då att få samma IP-adress som wildcard-posten.

Owner name med *

Begränsningar på domännamn med "*" för att vara wildcard:

- Måste vara en hel *label*.
- Måste vara första *label* i *owner name*.
- Gäller inte för definierade namn
- Används bara i *owner name*, inte i RDATA.

Owner name med *

```
$ORIGIN   exempel.se.

@                   SOA     (...)
                    NS      ns1.exempel.com.
                    A       192.0.2.10
www                 A       192.0.2.20
local.*             TXT     "Ej wildcard" ; Ej wildcard
```

Det måste vara **första** ”label”.

Owner name med *

```
$ORIGIN   exempel.se.

@                   SOA     (...)
                    NS      ns1.exempel.com.
                    A       192.0.2.10
www                 A       192.0.2.20
*.local             A       192.0.2.100
```

Nu är det istället namn under ”local.exempel.se”, t.ex. "a.local.exempel.se".

Fråga efter "*"

```
; <<>> DiG 9.10.6 <<>> *.dnskurs.narnia.pp.se txt +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15133
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 0


;; QUESTION SECTION:
;*.dnskurs.narnia.pp.se. IN TXT


;; ANSWER SECTION:
*.dnskurs.narnia.pp.se. 3600 IN TXT "DNSkurs wildcard exempel"
```

Om det finns ett "*" som wildcard och vi frågar efter det så får vi det i svaret. Mest för att ta reda på om det finns någon sådan post eller inte. Används inte av tjänster.

Fråga efter namn som matchar "*"

```
; <<>> DiG 9.10.6 <<>> tenta.dnskurs.narnia.pp.se txt +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39875
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 0


;; QUESTION SECTION:
;tenta.dnskurs.narnia.pp.se. IN TXT


;; ANSWER SECTION:
tenta.dnskurs.narnia.pp.se. 3600 IN TXT "DNSkurs wildcard exempel"
```

Zonen är samma som föregående bild, men nu frågade vi efter ett specifikt namn.

Wildcard gäller inte för definierade namn

```
$ORIGIN exempel.se.
@                       SOA     (...)
                        NS      ns1.exempel.com.
www4                    A       192.0.2.100
www6                    AAAA    2001:db8::1
*                       TXT     "www4 or www6"
```

Olika frågor ger olika resultat:

```
www4.exempel.se.        A?      →  192.0.100
www4.exempel.se.        TXT?    →  NODATA
www6.exempel.se.        AAAA?   →  2001:db8::1
www6.exempel.se.        TXT?    →  NODATA
www5.exempel.se.        TXT?    →  "www4 or www6"
www.exempel.se.         TXT?    →  "www4 or www6"
```

Wildcard och DNSSEC

```
$ORIGIN wildcard.xa.
@                       SOA     (...)
                        NS      ns1.exempel.com.
www                      A      192.0.2.100
*                       A       192.0.2.30
```

När zonen signeras med NSEC blir det NSEC-poster för de namn som finns i zonen och RRSIG för alla poster.

Vilka namn finns i zonen när den signeras?

- wildcard.xa
- *.wildcard.xa
- www.wildcard.xa

Wildcard och DNSSEC

Wildcard kan matcha många namn, det blir inga RRSIG eller NSEC för de namn som wildcard kan matcha, t.ex. web.wildcard.xa.

Wildcard och DNSSEC

```
$ORIGIN wildcard.xa.
@                       SOA     (...)
@                       RRSIG   SOA (...)
@                       NS      ns1.exempel.com.
@                       RRSIG   NS (...)
@                       DNSKEY  (...)
@                       RRSIG   DNSKEY (...)
@                       NSEC    *.wildcard.xa. NS SOA RRSIG NSEC DNSKEY
*                       A       192.0.2.30
*                       RRSIG   A (...)
*                       NSEC    www.wildcard.xa. A RRSIG NSEC
*                       RRSIG   NSEC (...)
www                     A       192.0.2.100
www                     RRSIG   A (...)
www                     NSEC    wildcard.xa A RRSIG NSEC
www                     RRSIG   NSEC (...)
```

web.wildcard.xa finns inte direkt i zonen, men zonen kan ge svaret via wildcard "*".

Wildcard och DNSSEC

```
*.wildcard.xa.          A       192.0.2.30
*.wildcard.xa.          NSEC    www.wildcard.xa. A RRSIG NSEC
```

Posterna ovan finns i zonen, och finns tillgängliga via frågor.

Hur kan vi då visa att följande genererade post (från wildcard) är giltig?

```
web.wildcard.xa.        A       192.0.2.30
```

Wildcard och DNSSEC

```
*.wildcard.xa.          A       192.0.2.30
*.wildcard.xa.          NSEC    www.wildcard.xa. A RRSIG NSEC
```

- A-posten visar att det finns en wildcardpost (A) med det RDATA.
- NSEC-posten visar att det inte kan finnas någon specifik post som matchar namnet "web.wildcard.xa".

RRSIG på posterna ovan gör det möjligt att validera.

web.wildcard.xa via wildcard

```
; <<>> DiG 9.16.25 <<>> web.wildcard.xa +dns +mult
(...)
;; QUESTION SECTION:
;web.wildcard.xa.  IN A

;; ANSWER SECTION:
web.wildcard.xa.   3600 IN A 192.0.2.30
web.wildcard.xa.   3600 IN RRSIG A 13 2 3600 (
                20220307185732 20220223095041 51609 wildcard.xa.
                NeaC9+IdGDhvdwhqCCM+5JVFXnW4E9YdwtDFUcDWQmAu
                pn9vtIxLMRNLzSDTMBs+uTFh6rYzyLoOR+LmJrDueA== )

;; AUTHORITY SECTION:
*.wildcard.xa.     300 IN NSEC www.wildcard.xa. A RRSIG NSEC
*.wildcard.xa.     300 IN RRSIG NSEC 13 2 300 (
                20220307185732 20220223095041 51609 wildcard.xa.
                axJuhricGBqzhgjeGeK3j4iZV8qVNb0sxoJdzYy788WR
                cLo2RmTN7IwSVcJxb3Fnw+a7FJAp4zKcX11nJTxSJA== )
```

”2” betyder att RRSIG gäller ett ”owner name” som har 2 ””lablar” före rot, bortsett från ev. initial wildcard-label.

Dessa två poster finns inte i zonen, utan är genererade från "*.wildcard.xa"

NSEC-posten visar att det finns en "*.wildcard.xa" med A och att det inte finns någon "web.wildcard.xa".

Jämför web.wildcard.xa med *.wildcard.xa

```
;; QUESTION SECTION:
;web.wildcard.xa.  IN A

;; ANSWER SECTION:
web.wildcard.xa.   3600 IN A 192.0.2.30
web.wildcard.xa.   3600 IN RRSIG A 13 2 3600 (
                20220307185732 20220223095041 51609 wildcard.xa.
                NeaC9+IdGDhvdwhqCCM+5JVFXnW4E9YdwtDFUcDWQmAu
                pn9vtIxLMRNLzSDTMBs+uTFh6rYzyLoOR+LmJrDueA== )

================================================================
;; QUESTION SECTION:
;*.wildcard.xa.        IN A

;; ANSWER SECTION:
*.wildcard.xa.     3600 IN A 192.0.2.30
*.wildcard.xa.     3600 IN RRSIG A 13 2 3600 (
                20220307185732 20220223095041 51609 wildcard.xa.
                NeaC9+IdGDhvdwhqCCM+5JVFXnW4E9YdwtDFUcDWQmAu
                pn9vtIxLMRNLzSDTMBs+uTFh6rYzyLoOR+LmJrDueA== )
```

Svar på två olika frågor. Se resp. ***Question Section***.

Samma RDATA i båda RRSIG-posterna. Den för "web.wildcard.xa" kommer från "*.wildcard.xa".

▶localhost och dess revers

[Till Innehåll]

Localhost och revers av 127.0.0.1 och ::1

Varken ”localhost” eller baklängesuppslagning av 127.0.0.1 finns i det publika DNS-trädet.

Samma sak gäller IPv6-adressen ::1.

Uppslagning av localhost

```
; <<>> DiG 9.10.6 <<>> @192.58.128.30 localhost +noedns +norec +mult
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 5261
;; flags: qr aa; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;localhost.     IN A

;; AUTHORITY SECTION:
.           86400 IN SOA a.root-servers.net. nstld.verisign-grs.com. (
                2020021501 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                604800     ; expire (1 week)
                86400      ; minimum (1 day)
                )

(...)
```

Fråga till en rotnamnserver.

Namnet ”localhost” finns inte i det publika DNS-trädet.

Uppslagning av revers av 127.0.0.1

```
; <<>> DiG 9.10.6 <<>> -x 127 @199.180.182.53 +noedns +mult +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 15408
;; flags: qr aa; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;127.in-addr.arpa. IN PTR

;; AUTHORITY SECTION:
in-addr.arpa.      3600 IN SOA b.in-addr-servers.arpa. nstld.iana.org. (
                  2020012104 ; serial
                  1800       ; refresh (30 minutes)
                  900        ; retry (15 minutes)
                  604800     ; expire (1 week)
                  3600       ; minimum (1 hour)
                  )

(...)
```

Fråga till en namnserver för in-addr.arpa-zonen.

Namnet ”127.in-addr.arpa” finns inte i det publika DNS-trädet, NXDOMAIN.

Därmed finns heller inte reversen för 127.0.0.1, 1.0.0.127.in-addr.arpa, i det publika Internet.

Uppslagning av revers av ::1

```
; <<>> DiG 9.10.6 <<>> 0.ip6.arpa. PTR @199.180.182.53 +noedns +mult +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 45252
;; flags: qr aa; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;0.ip6.arpa.            IN PTR

;; AUTHORITY SECTION:
ip6.arpa.       3600 IN SOA b.ip6-servers.arpa. nstld.iana.org. (
                2020012241 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                604800     ; expire (1 week)
                3600       ; minimum (1 hour)
                )

(...)
```

Fråga till en namnserver för in-addr.arpa-zonen.

Namnet ”0.ip6.arpa” finns inte i det publika DNS-trädet, NXDOMAIN.

Därmed finns heller inte reversen för ::1, 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa, i det publika Internet.

Localhost och revers av 127.0.0.1 resp ::1

Trots det så får vi ofta svar på ”localhost” resp. baklängesuppslagning av 127.0.0.1 och ::1 från resolvern.

Frågor till resolvern 9.9.9.9

```
; <<>> DiG 9.10.6 <<>> -x 127.0.0.1 @9.9.9.9
(...)
;; ANSWER SECTION:
1.0.0.127.in-addr.arpa. 86343 IN PTR localhost.
```

```
; <<>> DiG 9.10.6 <<>> -x ::1 @9.9.9.9
(...)
;; ANSWER SECTION:
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa. 10800 IN PTR localhost.
```

```
; <<>> DiG 9.10.6 <<>> localhost A @9.9.9.9
(...)
;; ANSWER SECTION:
localhost. 10800 IN A 127.0.0.1
```

```
; <<>> DiG 9.10.6 <<>> localhost AAAA @9.9.9.9
(...)
;; ANSWER SECTION:
localhost. 10800 IN AAAA ::1
```

Svaren kan inte ha hämtats från det publika DNS-trädet.

Localhost och revers av 127.0.0.1 resp ::1

En resolver är ofta, men inte alltid, direktkonfigurerad med följande DNS-poster, som inte kommer från det publika DNS-trädet.

```
localhost.                  A       127.0.0.1
1.0.0.127.in-addr.arpa.     PTR     localhost.
localhost.                  AAAA    ::1
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa. PTR localhost.
```

Resolvrarna 1.1.1.1 och 8.8.8.8 ger inga svar, 9.9.9.9 ger svar. Olika policybeslut.

Lokal ”localhost” i bind

Skapa en zonfil för ”localhost”.

```
$TTL        604800
$ORIGIN     localhost.
@       SOA     localhost. root.localhost. (
                      2     ; Serial
                 604800     ; Refresh
                  86400     ; Retry
                2419200     ; Expire
                 604800 )   ; Negative Cache TTL
        NS      localhost.
        A       127.0.0.1
        AAAA    ::1
```

Exakta värden i SOA-posten kommer att spela mindre roll eftersom vi ska ladda den i resolvern.

- Den kommer aldrig att föras över med zonöverföring.
- Vi kommer troligen aldrig att uppdatera den.

Lokal revers för 127.0.0.1 i bind

Skapa en zonfil för ”1.0.0.127.in-addr.arpa”.

```
$TTL        604800
$ORIGIN     1.0.0.127.in-addr.arpa.
@       SOA     localhost. root.localhost. (
                      2     ; Serial
                 604800     ; Refresh
                  86400     ; Retry
                2419200     ; Expire
                 604800 )   ; Negative Cache TTL
        NS      localhost.
        PTR     localhost.
```

Exakta värden i SOA-posten kommer att spela mindre roll.

Reverszon för exakt en IP-adress.

Lokal revers för ::1 i bind

Skapa en zonfil för revers av ::1.

```
$TTL        604800
$ORIGIN     1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa.
@       SOA     localhost. root.localhost. (
                      2     ; Serial
                 604800     ; Refresh
                  86400     ; Retry
                2419200     ; Expire
                 604800 )   ; Negative Cache TTL
        NS      localhost.
        PTR     localhost.
```

Exakta värden i SOA-posten kommer att spela mindre roll.

Reverszon för exakt en IP-adress.

Lokal zonfiler i bind

Lägg in dessa ”zone statement” i named.conf (named.conf.local).

```
zone "localhost" {
    type master;
    file "localhost";
};

zone "1.0.0.127.in-addr.arpa" {
    type master;
    file "revers-localhost-IPv4";
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" {
    type master;
    file "revers-localhost-IPv6";
};
```

Filnamnet bestämmar man själv vad det ska vara. Filen ska ligga på rätt plats så att bind kan ladda den.

▶Globalt eller privat DNS-träd

[Till Innehåll]

Globalt DNS-träd eller inte

I grunden så har vi ett gemensamt DNS-träd för hela Internet.

I praktiken så har vi privata hörn eller områden som inte nås från det allmänna Internet.

Det finns också det som vi inte vill visa upp för allmänna Internet.

▶Globala namn mot privata adresser

[Till Innehåll]

Privata IP-adresser

Vissa IP-adresser är utpekade för att användas i privata områden.

10.0.0.0/8	172.16.0.0/12	192.168.0.0/16	fc00::/7

Privat adress betyder att samma adress kan återanvändas på olika, oberoende nät. Men också att adresserna inte routas på Internet.

Privata IP-adresser

Förteckning över IP-adressområden med olika användning, IPv4 resp IPv6.

https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml

Globala namn mot privata adresser

Om vi har ett globalt namn som pekar ut en tjänst med privat IP-adress så blir detta bara meningsfullt på rätt nät.

Globala namn mot privata adresser

Om namnet används av mobil utrustning (t.ex. laptop) så finns det risk att den kommer att försöka komma åt tjänsten från fel nät där det kan finnas något annat på den adressen, vilket kanske inte uppskattas.

Globala namn mot privata adresser

Det kan också vara så att adressen aldrig svarar utan den mobila användaren alltid får vänta på timeout. Det kan skapa problem.

Globala namn bör inte peka ut privata adresser.

Webbserver med privat IP-adress

```
intra.exempel.se.       A       10.3.0.10
```

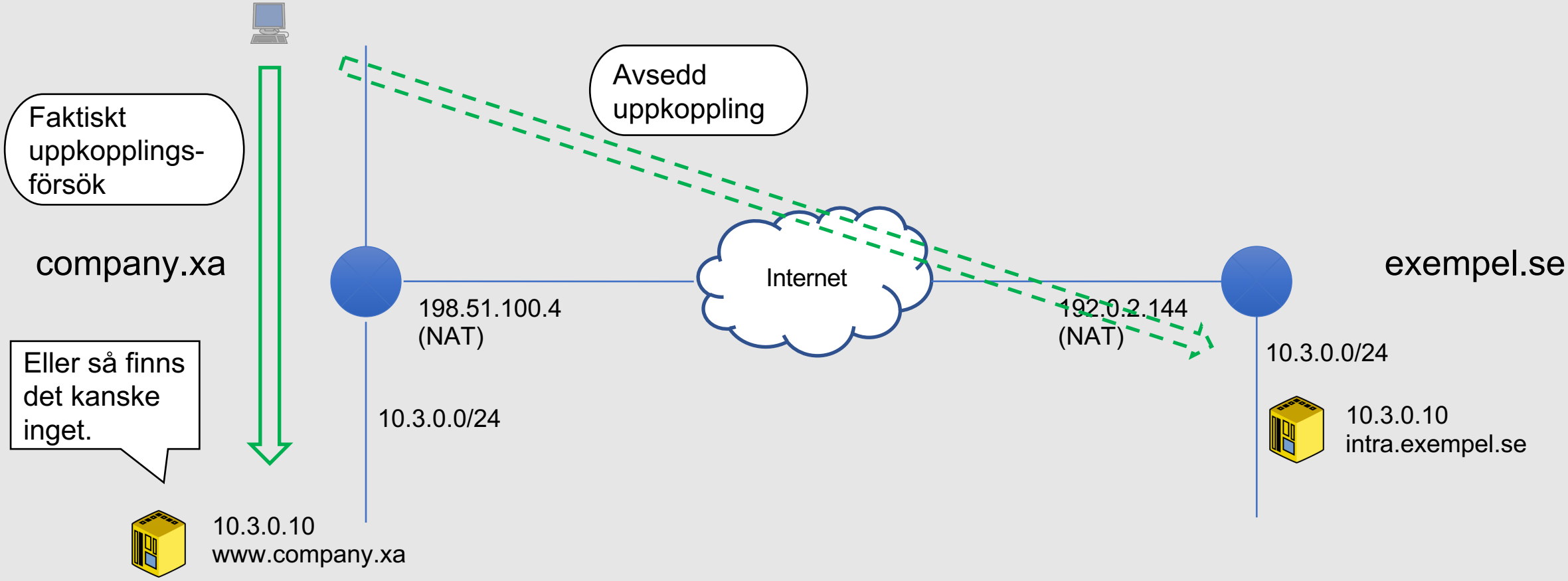
DNS-posten sitter i det publika DNS. När en mobil användare sitter utanför det interna nätet så tillhör 10.0.0.0/8 ingen eller någon annan.

Av vana eller automatiskt så gör användaren kanske ett försök till uppkoppling.

Webbserver med privat IP-adress

Besökare från exempel.se hos company.xa försöker ansluta till intra.exempel.se

Både exempel.se och company.xa använder de privata adresserna 10.3.0.0/24 men bakom olika NAT.

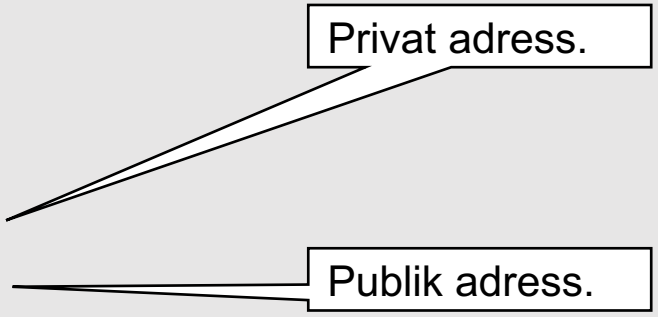


Webbserver med privat IP-adress

1. Om 10.0.0.0/8 inte finns där datorn är uppkopplad så kan det bli ett snabbt felmeddelande.
2. Om 10.0.0.0/8 finns där, men inte 10.3.0.10 (eller porten) så får användaren vänta på timeout.
3. Om 10.3.0.10 finns där och lyssnar på samma port så kan det bli oönskat uppkopplingsförsök eller oönskat felmeddelande.

Mailserver med privat IP-adress

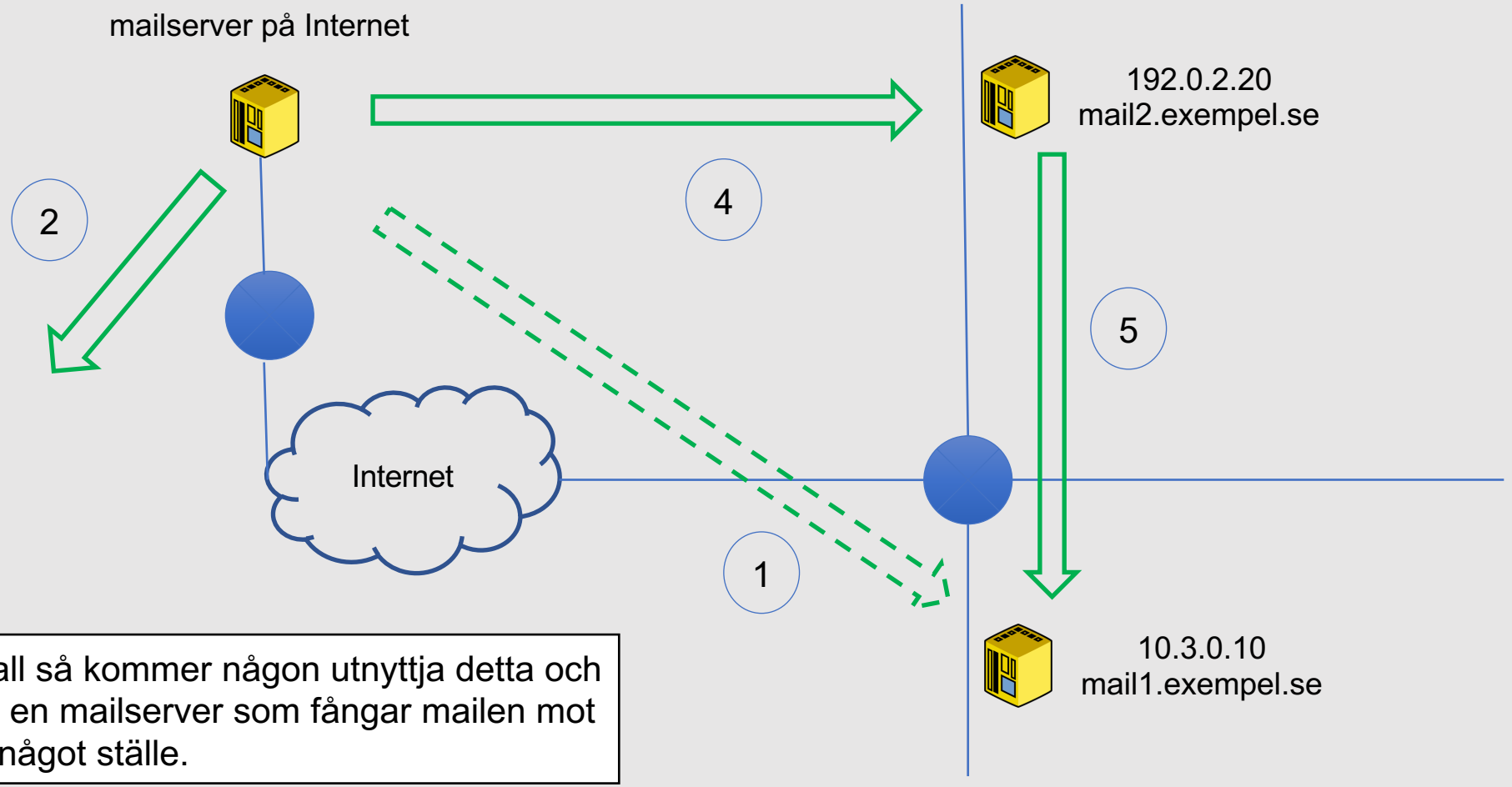
```
$ORIGIN exempel.se.
@               MX      10  mail1
                MX      20  mail2
www             A       192.0.2.10
mail1           A       10.3.0.10
mail2           A       192.0.2.144
```



Mail ska i första hand levereras till mail1 (prio 10 i mx) och i andra hand till mail2 (prio 20). Mail som hamnar på mail2 kommer automatiskt att skickas till mail1 med SMTP.

Mailserver med privat IP-adress

1. Den avsedda men omöjliga vägen mot mail1.
2. Det verkliga försöket till leverans.
3. Timeout? Felmeddelande? (Beror på om något svarar eller inte.)
4. Skickar mailet till mail2 istället.
5. Mailet förs över till mail1.

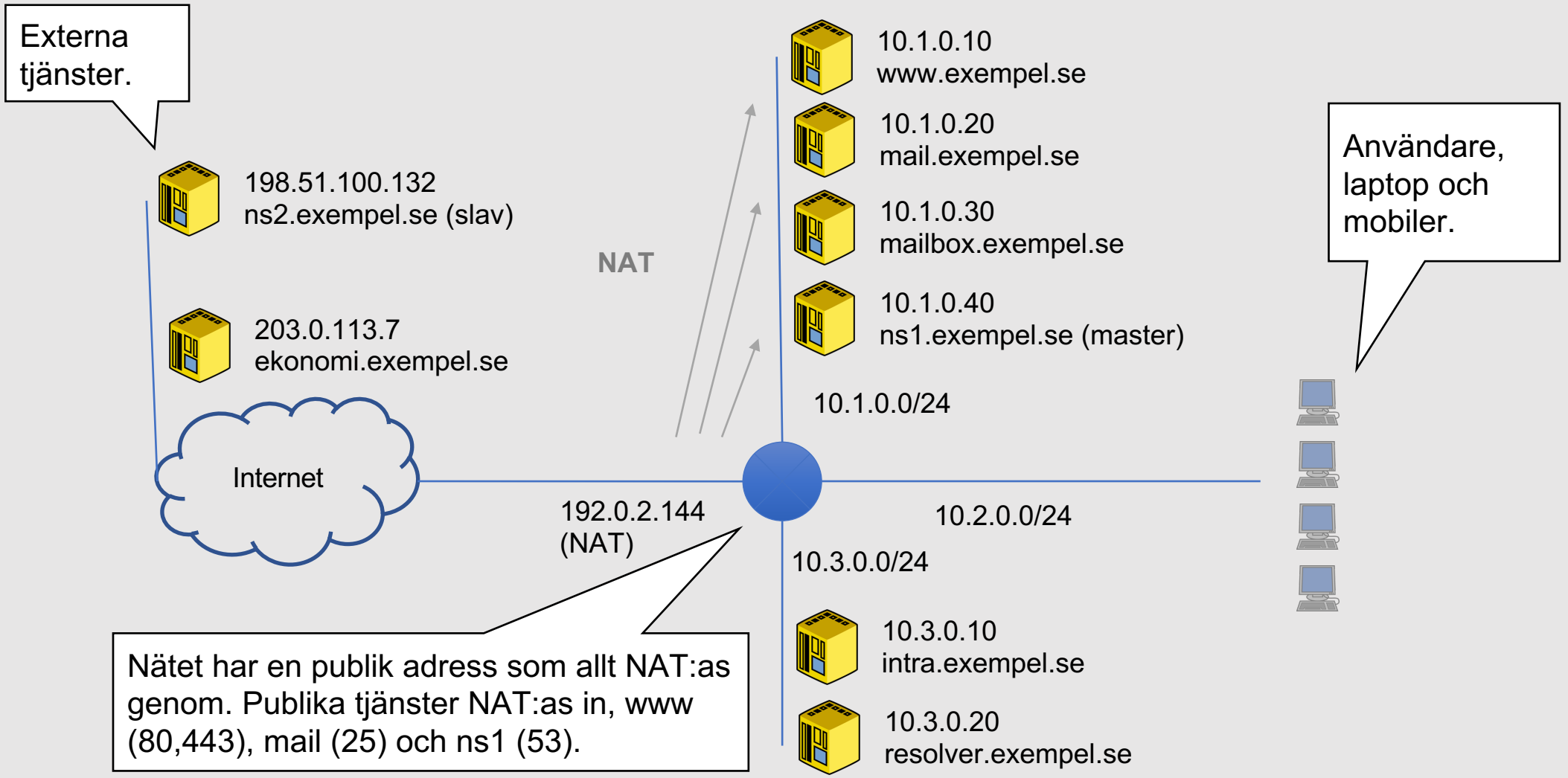


I värsta fall så kommer någon utnyttja detta och sätta upp en mailserver som fångar mailen mot mail1 på något ställe.

▶Hos exempel.se

[Till Innehåll]

Hos exempel.se



exempel.se

```
$ORIGIN exempel.se.
@                MX       1  mail
www              A        192.0.2.144       ; Webb för extern åtkomst
www-host         A        10.1.0.10         ; Webb för intern åtkomst
mail             A        192.0.2.144       ; Inkommande mail
mail-host        A        10.1.0.20         ; För att kunna logga in på mail med ssh
mailbox          A        10.1.0.30         ; Skicka och läsa mail
ns1              A        192.0.2.144
ns1-host         A        10.1.0.40         ; För att kunna logga in på ns1 med ssh
intra            A        10.3.0.10         ; Interna dokument och intranät
resolver         A        10.3.0.20         ; DNS-resolver
ns2              A        198.51.100.132    ; På extern server
ekonomi          A        203.0.113.7       ; Extern tjänst
```

- Mail och dokument bara från kontoret
- Olika namn för webb internt resp publikt
- Speciella servernamn för intern åtkomst

Utanför kontoret hos exempel.se

Några medarbetarna har bärbara datorer och jobbar hemifrån. Mailen är inte åtkomlig.

Alla har mobiler, men kan inte använda den för mail om de inte går via lokalt WIFI på kontoret.

Utanför kontoret hos exempel.se

Intranätsservern, intra.exempel.se, är bara åtkomlig internt. Om någon försöker komma åt den externt av misstag så kan det bli lång timeout.

På kontoret hos exempel.se

Webbservern har olika namn internt och externt, vilket leder till komplex lösning i DNS och irritation när de skriver fel.

På kontoret hos exempel.se

Andra servrar har också olika namn internt och externt vilket leder till felskrivningar åt båda hållen. (Externa namnet används internt och interna namnet externt.)

Hur ska man gå vidare?

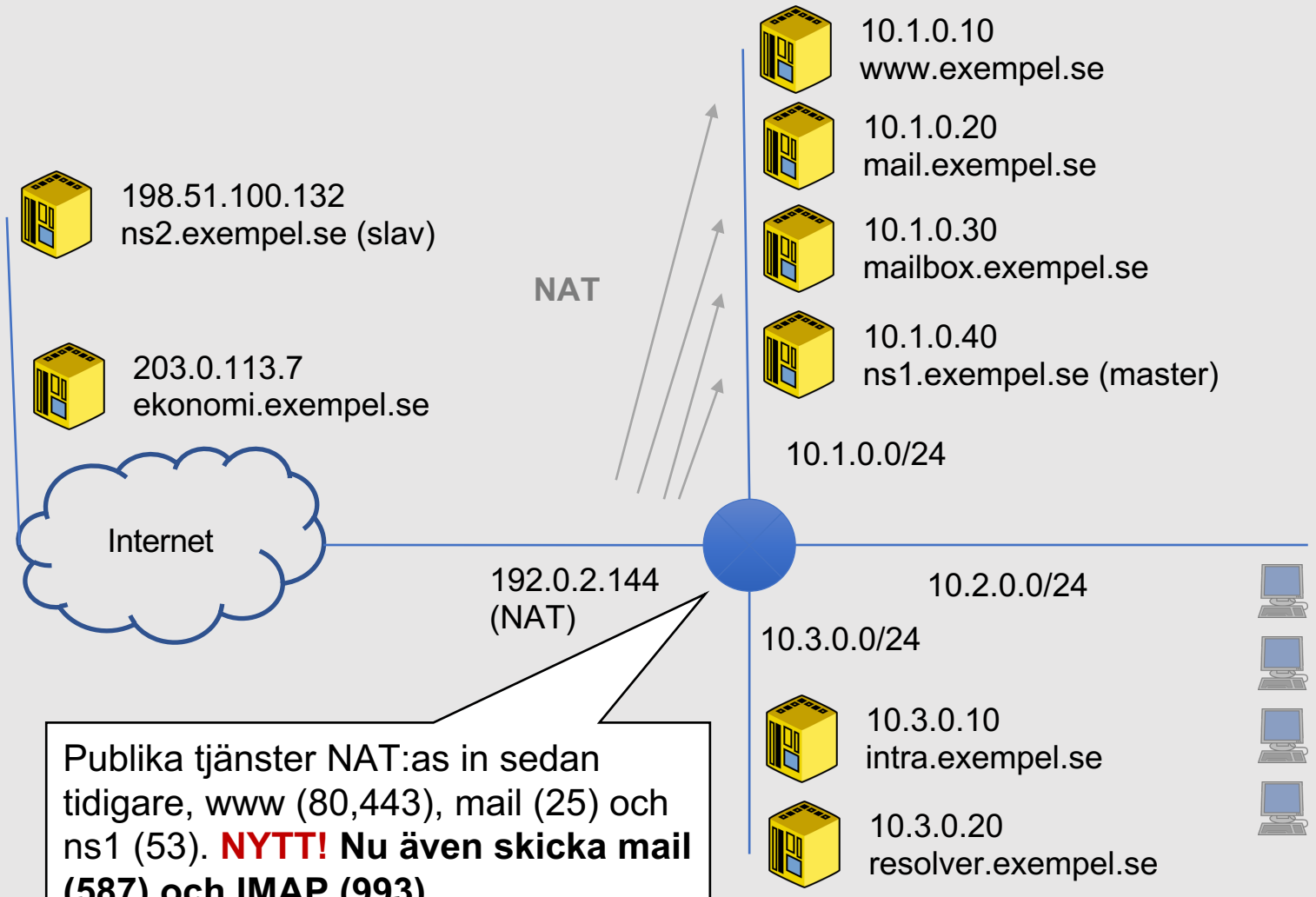
Utanför kontoret hos exempel.se

Kraven på att komma åt mailen från utsidan leder till att det öppnas upp.

▶Fler tjänster externt hos exempel.se

[Till Innehåll]

Hos exempel.se



exempel.se uppdaterat för mail

```
mailbox         A       10.1.0.30    ; Skicka och läsa mail internt
mailbox-ext     A       192.0.2.144  ; Skicka och läsa mail externt
```

Kan komma åt mail utanför kontoret, men måste ange annat namn.

Lösningen är inte bra eftersom detta kommer leda till supportärenden.

Problemet med webb m.m. kvarstår.

▶Dela upp internt/externt hos exempel.se

[Till Innehåll]

Lösa problemet

Man ska lösa problemet och först identifiera de tjänster och DNS-poster som hör till extern resp intern uppslagning.

exempel.se för internt resp. externt

```
; Gemensamt internt och externt:
exempel.se.     MX      1  mail
ns2             A       198.51.100.132 ; Extern slav
ekonomi         A       203.0.113.7

; För extern åtkomst:
www             A       192.0.2.144    ; Samma namn som internt
mail            A       192.0.2.144    ; Samma namn som internt
mailbox         A       192.0.2.144    ; Samma namn som internt
ns1             A       192.0.2.144    ; Samma namn som internt

; För intern åtkomst:
www             A       10.1.0.10      ; Samma namn som externt
mail            A       10.1.0.20      ; Samma namn som externt
mailbox         A       10.1.0.30      ; Samma namn som externt
ns1             A       10.1.0.40      ; Samma namn som externt
intra           A       10.3.0.10      ; Interna dokument och intranät
resolver        A       10.3.0.20
```

Styra uppslagningen

Nu har vi en specifikation på interna resp externa uppslagningar. Vi har samma namn både internt och externt.

Hur ska vi styra uppslagningen så att man bara får det ena? – Intern adress internt och extern adress externt.

Styra uppslagningen

Interna användare (mobil, laptop eller desktop) får inställning via DHCP. DNS-resolver ställs till den lokal resolvern, 10.3.0.20.

Interna servrar använder samma resolver.

Styra uppslagningen

Externa användare går mot extern resolver (t.ex. Googles resolver) som hittar uppslagningen på ns1 eller ns2.

▶Bind views

[Till Innehåll]

Vald lösning

Vi väljer att utnyttja funktionen ”views” i bind och att lägga lösningen på ns1 för att hålla all namnsättning på ett ställe.

Det finns andra sätt att lösa problemet.

Dubbla zonfiler

Vi behöver två zonfiler för exempel.se, en för den interna vyn och en annan för den externa vyn.

exempel.se (intern vy)

```
; Filnamn exempel.se_internt
;
exempel.se.     SOA     (...)           ; Både externt och internt
                NS      ns1             ; Både externt och internt
                NS      ns2             ; Både externt och internt
                MX      1   mail        ; Både externt och internt
www             A       10.1.0.10
mail            A       10.1.0.20
mailbox         A       10.1.0.30
ns1             A       10.1.0.40
intra           A       10.3.0.10
resolver        A       10.3.0.20
ns2             A       198.51.100.132 ; Både externt och internt
ekonomi         A       203.0.113.7    ; Både externt och internt
```

exempel.se (extern vy)

```
; Filnamn exempel.se_externt
;
exempel.se.     SOA     (...)           ; Både externt och internt
                NS      ns1             ; Både externt och internt
                NS      ns2             ; Både externt och internt
                MX      1   mail        ; Både externt och internt
www             A       192.0.2.144     ; Port 80 och 443 via NAT
mail            A       192.0.2.144     ; Port 25 via NAT
mailbox         A       192.0.2.144     ; Port 587 och 993 via NAT
ns1             A       192.0.2.144     ; Port 53 via NAT
intra           CNAME   www             ; Kommentar på nästa bild
ns2             A       198.51.100.132  ; Både externt och internt
ekonomi         A       203.0.113.7     ; Både externt och internt
```

Intranätet från extern åtkomst

Externt låter vi namnet ”intra” gå mot den externa webbservern som antingen ger proxy-access till vissa delar av intranätet eller bara ger snabbt och vettigt felmeddelande.

Samma namn internt och externt

Nu finns alla namn, utom resolver.exempel.se, både internt och externt.

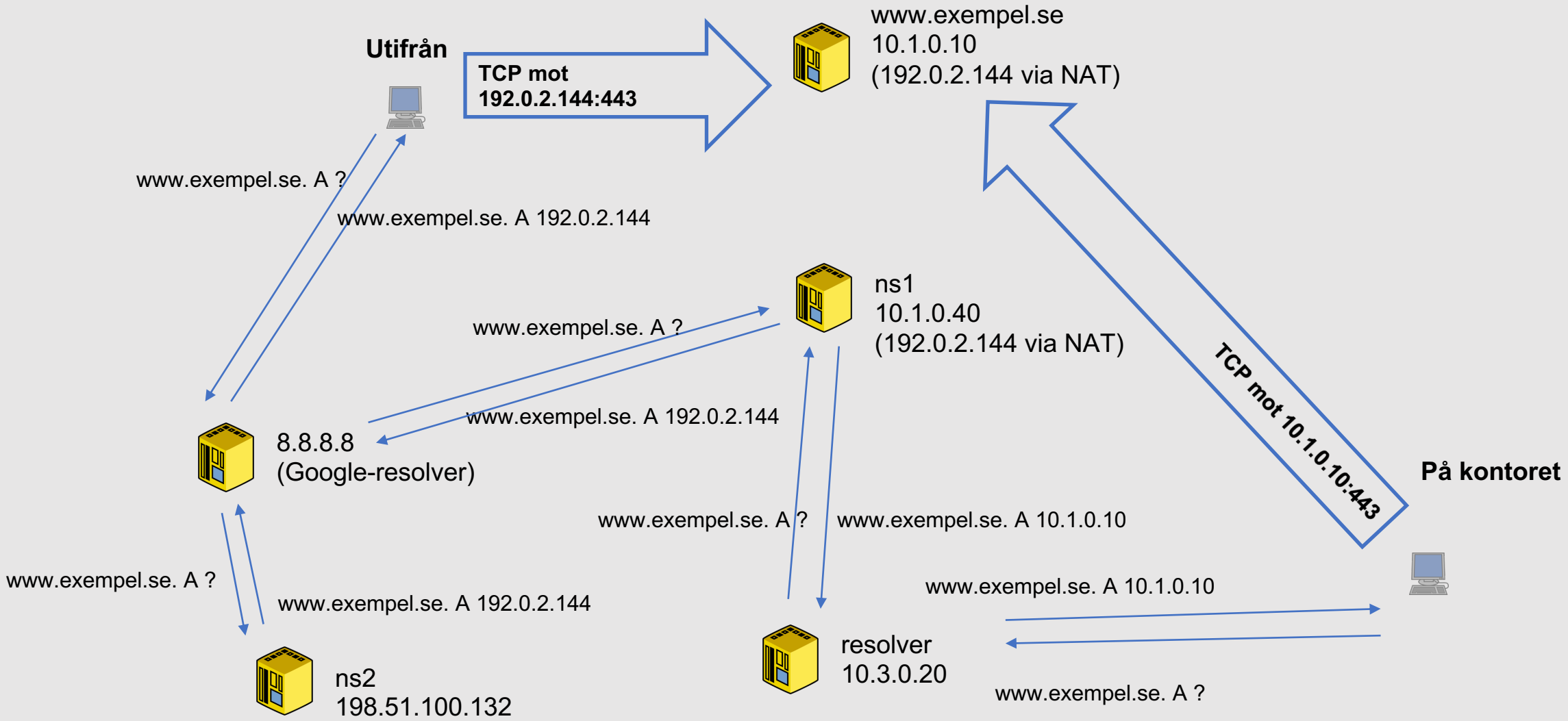
Bind ”views” i named.conf på ns1

```
view "internt" {
    match-clients { 10.0.0.0/8; 127.0.0.1; }; # Specifika adresser
    zone "exempel.se" {
        type master;
        file "exempel.se_internt";
    };
};

view "externt" {
    match-clients { any; }; # Ofta fångar sista view ”alla andra”
    zone "exempel.se" {
        type master;
        file "exempel.se_externt";
    };
};

# Ordningen är avgörande. Den första vyn som ger match kommer
# att användas för anropet.
```

Hur styr vyerna www.exempel.se?



Vad är vyer i Bind?

Se vyer som en virtualisering. Varje vy blir som en egen namnserver.

- Varje anrop (fråga, ”query”) styrs till en specifik vy – eller ingen vy om det inte finns någon vy som matchar klienten.
- Alla "zone statement" måste tillhöra en specifik vy. Man får dubblera om den ska tillhöra båda.
- Man kan ha många vyer, inte bara två.
- Annan konfiguration utanför vyerna blir defaultvärden för båda vyerna.

Vad är vyer i Bind?

Match i en vy kan baseras på

- Klientens IP-adress (source IP)
  - Som i vårt exempel.
- Serverns IP-adress (target IP)
  - Kräver att servern har flera adresser. Adress genom NAT räknas inte in.
- TSIG-nyckel i ”query”

Vyer på ns2

- Vilken vy hamnar på ns2 (slav)?
  - ”Query” från ns2 hamnar i den externa vyn på ns1, vilket gör att zonfilen som hämtas är den externa vyn och den externa version av zonen.
- Kan vi ha båda vyerna på ns2?
  - Ja, men det kräver lösning med TSIG.
- Vad kan hända när ns2 inte har båda vyerna?
  - Om ”resolver” ställer frågan till ns2 så får den externt svar och klienten får problem.

Alternativ

1. Se till att ”resolver” alltid går till ns1
2. Se till att ns2 har båda vyerna

För detta scenario så är det rimligast att välja alternativ 1.

I andra fall kan alternativ 2 vara bäst.

▶Styra till intern vy via resolvern

[Till Innehåll]

named.conf på resolver

```
zone ”exempel.se." {
    type static-stub;
    server-addresses { 10.3.0.20; };
};

# Istället för att gå den vanliga vägen för exempel.se,
# gå till denna specifika adress (ns1).
```

exempel.se (intern vy) – utan ns2 som NS

```
; Filnamn exempel.se_internt
;
$ORIGIN exempel.se.
@               SOA     (...)
                NS      ns1             ; Bara en NS, ns1, på intern vy
                MX      1   mail
www             A       10.1.0.10
mail            A       10.1.0.20
mailbox         A       10.1.0.30
ns1             A       10.1.0.40
intra           A       10.3.0.10
resolver        A       10.3.0.20
ns2             A       198.51.100.132
ekonomi         A       203.0.113.7
```

Hur får man baklängesuppslagning?

Baklängesuppslagning för publika adresser följer den vanliga vägen från rot.

Hur ser det ut för privata adresser, t.ex. ur 10.0.0.0/8?

Baklänges för privata adresser

Baklänges för privata adresser är bara meningsfullt på det egna nätet.

Nu har vi en vy för det egna nätet i ns1.

10.in-addr.arpa

```
; Filnamn 10.0.0.0-8.rev
;
$ORIGIN 10.in-addr.arpa.
@                   SOA     (...)
                    NS      ns1.exempel.se.         ; Bara en NS
10.0.1              PTR     www.exempel.se.
20.0.1              PTR     mail.exempel.se.
30.0.1              PTR     mailbox.exempel.se.
40.0.1              PTR     ns1.exempel.se.
10.0.3              PTR     intra.exempel.se.
20.0.3              PTR     resolver.exempel.se.
101.0.2             PTR     dhcp101.exempel.se.
; alla mellan 101 och 150
150.0.2             PTR     dhcp150.exempel.se.
```

named.conf på ns1

```
view "internt" {
    match-clients { 10.0.0.0/8; 127.0.0.1; };
    zone "exempel.se" {
        type master;
        file "exempel.se_internt";
    };
    zone ”10.in-addr.arpa" {
        type master;
        file ”10.0.0.0-8.rev";
    };
};

view "externt" {
    match-clients { any; };
    zone "exempel.se" {
        type master;
        file "exempel.se_externt";
    };
};
```

Reversen är oåtkomlig om man inte kommer från de interna näten. Det betyder att ingen externt kan ställa frågor om revers på interna adresser även om frågan ställs direkt mot ns1.

named.conf på resolver

```
zone ”exempel.se." {
    type static-stub;
    server-addresses { 10.3.0.20; };
};

zone ”10.in-addr.arpa." {
    type static-stub;
    server-addresses { 10.3.0.20; };
};
# Styr revers av 10.0.0.0/8 till lokal server istället för
# globala internet.
```

exempel.se (intern vy)

```
; Filnamn exempel.se_internt
;
$ORIGIN exempel.se.
@               SOA     (...)
                NS      ns1
                MX      1   mail        ; Både externt och internt
www             A       10.1.0.10
mail            A       10.1.0.20
mailbox         A       10.1.0.30
ns1             A       10.1.0.40
intra           A       10.3.0.10
resolver        A       10.3.0.20
ns2             A       198.51.100.132 ; Både externt och internt
ekonomi         A       203.0.113.7    ; Både externt och internt
dhcp101         A       10.2.0.101     ; Alla 101-150 ska läggas in
(...)
dhcp150         A       10.2.0.150     ; Alla 101-150 ska läggas in
```

Manuellt ändrad resolver

Om en användare på kontoret ändrar resolver till 8.8.8.8 så kommer användaren inte att kunna komma åt interna tjänster längre.

Beroende på NAT-lösning så kan det vara så att den publika adressen inte fungerar för att komma åt tjänsterna heller så länge som enheten sitter på det interna nätet.

▶Vyer och "views"

[Till Innehåll]

Olika sätt att åstadkomma vyer

”Views” i bind är ett avancerat sätt att skapa vyer, men det är inte alltid som en sådan lösning behövs eller finns tillgänglig.

Alla namnservrar tillhandahåller sätt att konfigurera en resolver med A-, AAAA- och PTR-poster för att antingen

1. Skriva över publika DNS-poster med lokala värden (normalt privata IP-adresser).
2. Skjuta in lokala namn som inte finns publikt.

Vyer för att dölja

Även om nätverket är helt baserat på globala (publika) adresser så kan det finnas användning för vyer.

På insidan så kan DNS via baklängesuppslagning ge full information om enheterna, d.v.s. namn om avslöjar vad det är för tjänst. På utsidan så kan DNS ge anonyma namn.

En sådan lösning gör att enheterna har namn vid anrop mot externa tjänster utan att berätta för mycket om dem, men på insidan ge full information.

▶Reverser på privata och dynamiska adresser

[Till Innehåll]

Tomma reverser i bind

Bind skapar **tomma** baklängeszoner (reverszoner) för en lång rad IP-block som ändå inte kan slås upp i det publika DNS inklusive för 127.0.0.0/8 och ::1/128 om man inte

1. stänger av funktionen, eller
2. skapar egen zonfil (se tidigare bilder)

Tomma reverser i bind

Tomma baklängeszoner betyder att man inte får något svar på PTR-frågan. Det är bättre att få svar om svaret är meningsfullt, t.ex. för ”localhost”, och revers av 127.0.0.1 resp. ::1.

ULA – privata IPv6-adresser

Motsvarigheten till privata adresser under IPv6 heter ”Unique Local Addresses” och finns under prefixet fc00::/7

Om de används och man vill ha baklängesuppslagning så får man göra på motsvarande sätt som för 10.0.0.0/8.

Behovet av ULA är dock inte lika stort som för privata IPv4-adresser eftersom det finns gott om globala (publika) IPv6 adresser. Om man väljer ULA så är det annan orsak, inte adressbrist.

Dynamiska IPv6-adresser

Tilldelning av IPv6-adress för klienter (sådant som inte är servrar) är ofta mera dynamiskt från en stor adressrymd.

Det är en utmaning att skapa reverser – och framlängesnamn – för dynamiska adresser.

Minsta subnät för IPv6 är /64 vilket ger $1{,}8 \cdot 10^{19}$ adresser.

Dynamiskt tilldelade adresser kan inte hanteras med statiskt skapade revers- och framlängeszoner.

Dynamiska IPv6-adresser

Om man ska skapa reverser – och motsvarande framlängesnamn – för dynamiska IPv6-adresser så måste det vara ”on the fly”.

Det finns teknik för det i namnservrarna, inkl. signering ”on the fly” för DNSSEC.

▶Filtrering i resolvern

[Till Innehåll]

Resolvern kan "ljuga"

Den som kontrollerar resolvern kontrollerar också vad resolvern ska svara.

- Styra om allt till en loginsida.
- Blockera utvalda domäner genom att ge SERVFAIL eller annat svar.
- Skriva om utvalda namn och data.
- Skjuta in lokal baklängesuppslagning.

Resolvern kan "ljuga"

I vilken mån klienten kan se "lögnen" beror på om zonen man frågar om är signerad med DNSSEC eller inte.

Och om klienten själv validerar eller inte, vilket fortfarande är ovanligt.

Resolvern kan rensa bort

Klienten kan vilja ha en resolver som "ljuger" om det betyder att skadlig kod och fishing-sidor motas bort.

Barnporrfiltrering

Ett enkelt sätt att filtrera bort barnporr är att stoppa in de domänerna som man vill stoppa i resolvern och skicka klienten till en portal.

Det är inte ett felsäkert sätt, men kanske tillräckligt bra.

Gemensam zonfil för alla barnporrsdomäner

```
; Filnamn "block-domain"
;
$TTL        604800
; $ORIGIN
@       SOA     localhost. root.localhost. (
                      2     ; Serial
                 604800     ; Refresh
                  86400     ; Retry
                2419200     ; Expire
                 604800 )   ; Negative Cache TTL
        NS      localhost.
        A       203.0.113.74
        AAAA    2001:DB8:10:A:C::80
*       A       203.0.113.74
*       AAAA    2001:DB8:10:A:C::80
```

$ORIGIN är bortkommenterad för vi ska använda samma zonfil till många zoner.

Exakta värden i SOA-posten kommer att spela mindre roll eftersom vi ska ladda zonen i resolvern.

Vi använder wildcard här för att fånga alla namn under den konfigurerade barnporrsdomänen. IP-adresserna går till en portal med information om att domänen är blockerad och varför.

Lokala "zone statement" i named.conf

```
zone "barnporr.xa" {   type master;   file "block-domains"; };
zone "barnporr2.aa" {  type master;   file "block-domains"; };
zone "barnporr5.oo" {  type master;   file "block-domains"; };
zone "barnporr.qm" {   type master;   file "block-domains"; };
zone "barnporr.xo" {   type master;   file "block-domains"; };
```

Samma zonfil används till alla zoner. Listan kan lätt genereras. Kan vara en include-fil till named.conf.

RPZ, Response Policy Zone

RPZ är ett avancerat sätt att i resolvern filtrera DNS-svaret.

Förutom att RPZ kan användas för att åstadkomma vyer så kan det även – och kanske framförallt – användas för att tvätta bort oönska domäner från resolvern.

RPZ har utvecklats i Bind men finns även i Unbound och kanske även i andra namnservrar.

▶CNAME och revers

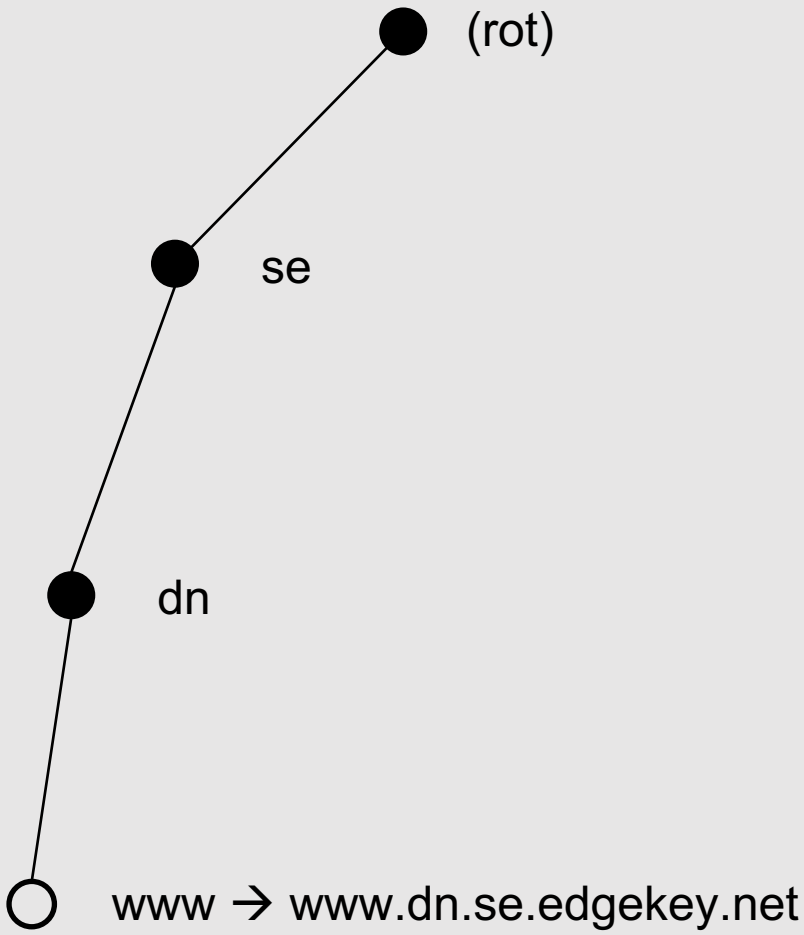
[Till Innehåll]

www.dn.se. A ?

```
www.dn.se.          CNAME www.dn.se.edgekey.net.
```

Träd från owner name

● = nod som är start av zon
○ = nod som *inte* är start av zon

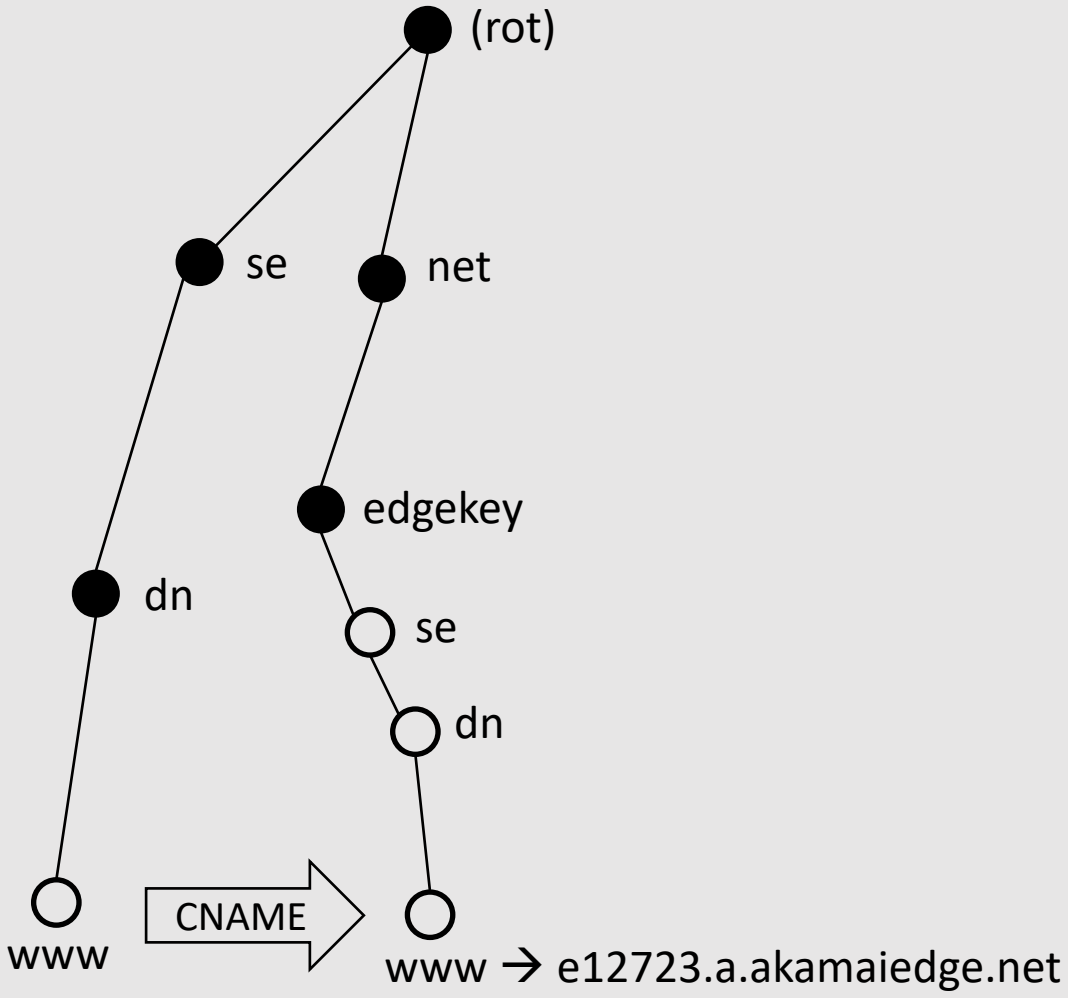


www.dn.se. A ?

```
www.dn.se.                CNAME www.dn.se.edgekey.net.
www.dn.se.edgekey.net.    CNAME e12723.a.akamaiedge.net.
```

Träd från owner name

● = nod som är start av zon
○ = nod som *inte* är start av zon

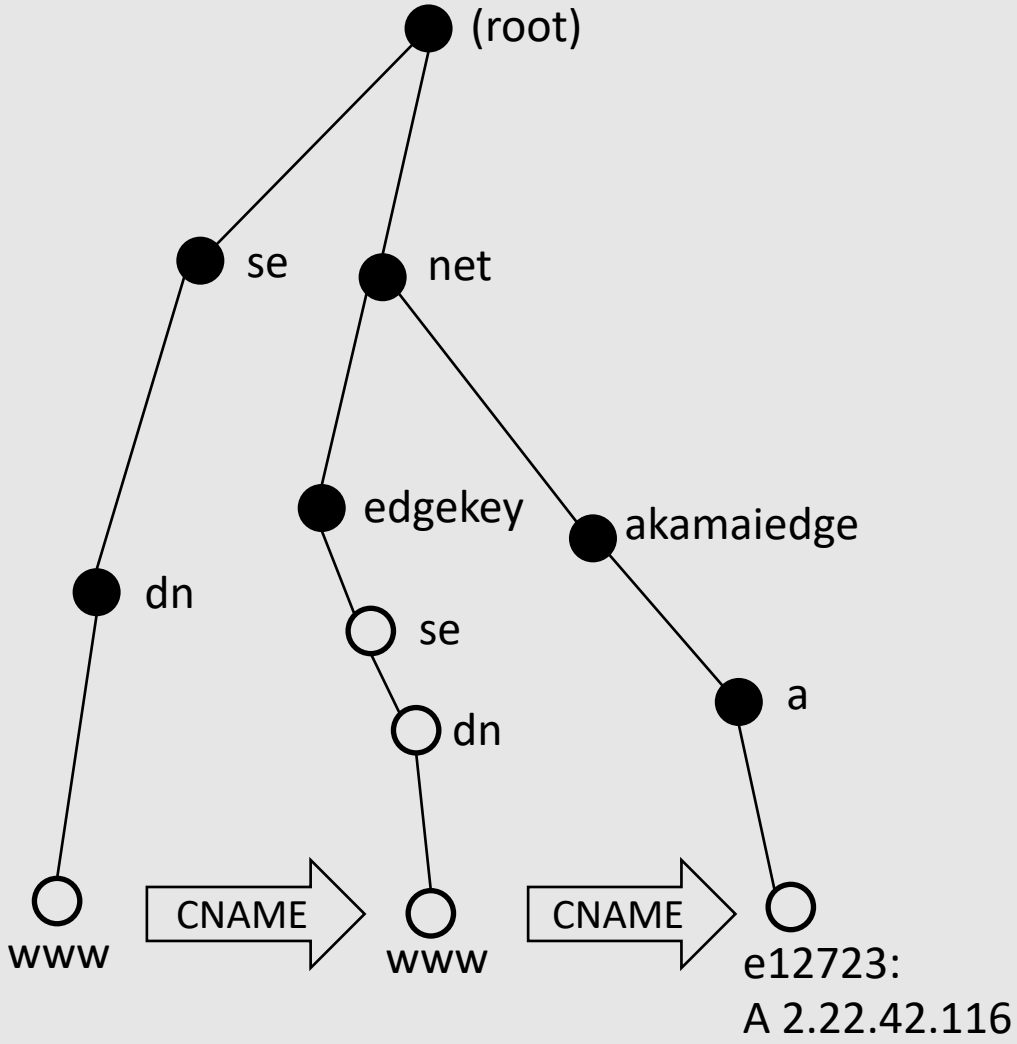


www.dn.se. A ?

```
www.dn.se.                  CNAME www.dn.se.edgekey.net.
www.dn.se.edgekey.net.      CNAME e12723.a.akamaiedge.net.
e12723.a.akamaiedge.net.    A 2.22.42.116
```

Träd från owner name

● = nod som är start av zon
○ = nod som *inte* är start av zon



PTR för revers av 45.155.99.226

226.99.155.45.in-addr.arpa. PTR ?

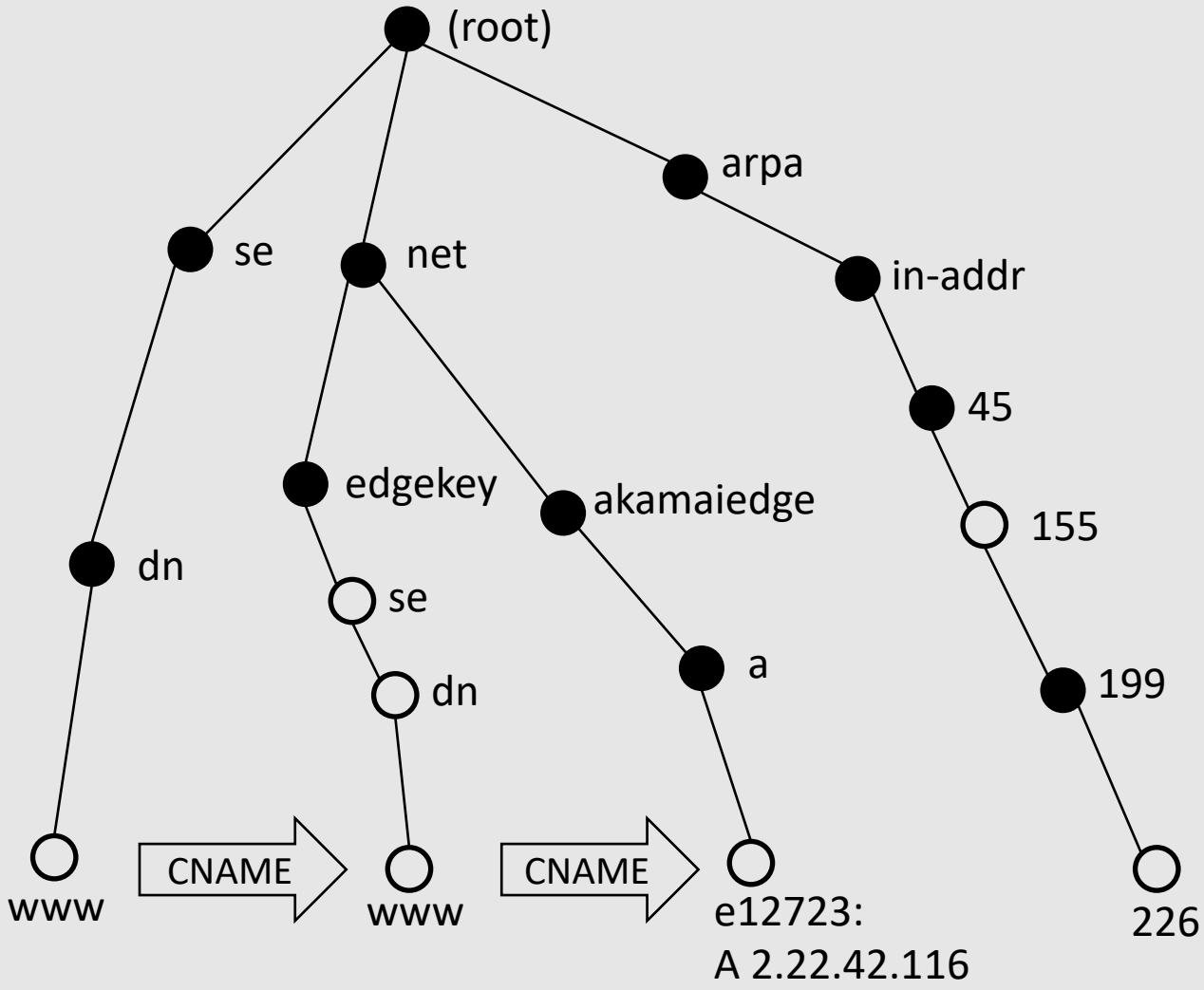
"dig -x 45.155.99.226" ger rätt namn

226.99.155.45.in-addr.arpa. PTR ?

```
226.99.155.45.in-addr.arpa. CNAME 226.224-231.99.155.45.in-addr.arpa.
```

Träd från owner name

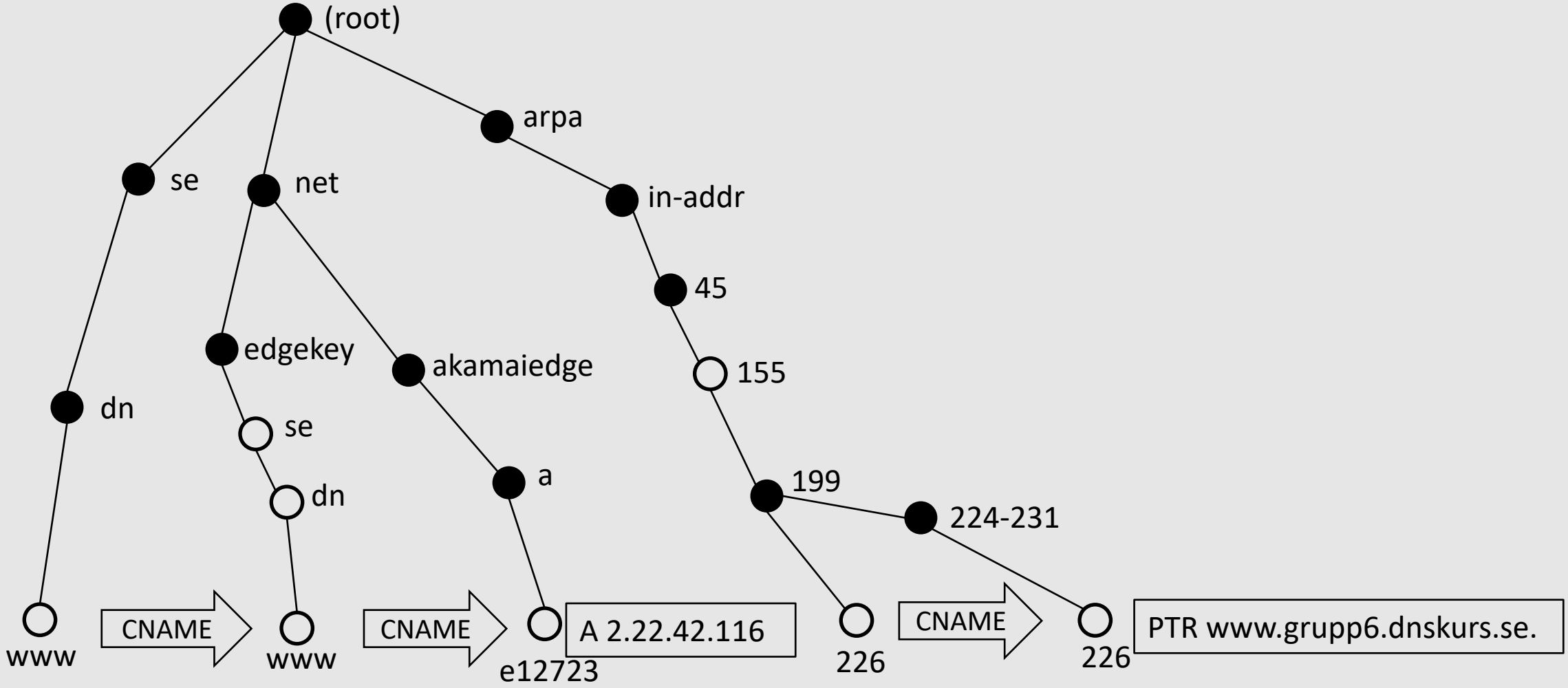
● = nod som är start av zon
○ = nod som *inte* är start av zon



226.99.155.45.in-addr.arpa. PTR ?

```
226.99.155.45.in-addr.arpa.   CNAME   226.224-231.99.155.45.in-addr.arpa.
226.224-231.99.155.45.in-addr.arpa.   PTR   www.grupp6.dnskurs.se.
```

Träd från owner name



▶Om presentationen

[Till Innehåll]

Internets domännamnssystem

Denna presentation är framtagen 2019–2023 av Mats Dufberg (mats.dufberg@internetstiftelsen.se) på Internetstiftelsen (https://internetstiftelsen.se/). Den är en del av undervisningsmaterialet för kursen ”Internets domännamnssystem” vid Kungliga tekniska högskolan, KTH (kurskod HI1037) resp. Karlstads universitet, KAU (kurskod DVGC28).

Licens

Dokumenthistorik

- Rev A: Ursprunglig version HT 2023

Slut.