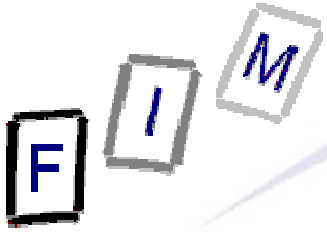


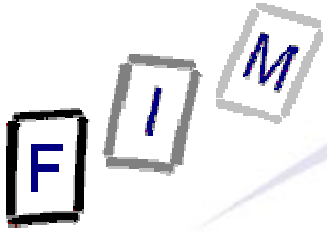
Mag. iur. Dr. techn. Michael Sonntag

Domain Names

Registration, BIND

Institute for Information Processing and
Microprocessor Technology (FIM)
Johannes Kepler University Linz, Austria

E-Mail: sonntag@fim.uni-linz.ac.at
http://www.fim.uni-linz.ac.at/staff/sonntag.htm



Questions?

Please ask immediately!

Introduction

- Basics
  - → Guidelines for selection; dangers
  - → Authoritative vs. caching, iterative vs. recursive
- Registration
  - » Kinds of providers and selection: see other lecture!
- BIND
  - → Installation
  - → Basic configuration
    - » Caching OR authoritative ONLY
  - → Name files
  - → Dynamic updates
  - → Security

The DNS system

- Origin: Distributed name system
  - → Additional information available now, e.g.
    - » Mail exchangers
    - » Location information
    - » SPAM-protection information
- Hierarchical structure
  - → Every domain is "responsible" for those directly below
    - » No consent needed from "above" for new domains "below"!
- Most vulnerable point: Root of hierarchy
  - → These are the root servers (currently 13)
    - » These actually consist of numerous computers physically distributed over the world
    - » All of them provide identical content (Responsible for it: ICANN)
  - → Alternative hierarchies exist (e.g. OpenNIC)!

TLDs

- Two kinds of Top Level Domains (TLDs):
  - → Generic: .com, .info, .edu, ...
    - » Seven new ones were introduced 2000
    - » 10 additional ones are currently under consideration
  - → Country code: .at, .uk, .hu, .de, ...
    - » Dependent on the UN list of countries
- Most are unsponsored, i.e. available to the public
  - → Some are sponsored (e.g. .museum) and therefore available only to restricted user groups (e.g. all kinds of museums)
- Most desirable TLDs for companies:
  - → .com: Worldwide commercial activity
  - → ."country-code": Home country registration
    - » Not important any more: .co."country-code" (too long, ...)

Guidelines for selection

- Use your company and your product name
  - → To be easily found by name
  - → Use separate websites: You might change your product!
- Register several names
  - → They are not that expensive; point them to the same server
- Register different writings (with/-out hyphens, ...), slight misspellings and related names
  - → Avoids problems and eases finding you
- Register under several TLDs (.com, ."cc", .org, ...)
  - → For future extensions, to be found more easily
- If applicable, use both "ordinary" and internationalized DN
  - → Internationalized DN are not yet widely known or supported
- Try to use short and memorable names

Caveats

- In which country is the registrar located?
  - → Depending on this problems, court decisions, etc. might be hard to implement in reality
- Dispute resolution policies
  - → Which, if any, are applicable? Arrangement, costs, judges, ...
- Is it a trademark, service mark, etc.?
  - → Does your domain infringe on any of these?
  - → Search in a search engine before!
- Does the DN have a meaning in other languages?
- Similar to existing DNs?
- Several classes of names are "off-limit": city names, famous company/product names, ...
- Make sure who is the actual owner of the DN

Terminology

- Root domain: "." Origin of the name hierarchy (root servers)
  - → All absolute domain names end with this ("www.jku.at.")
    - » Relative names end without a "."
    - » Most clients automatically add a "." at the end
  - → Absolute domain names are called "FQDN" (Fully Qualified DN)
- Nameserver: A server providing resolution DN ↔ IP address
- Delegation: Creating a "sub"-domain (e.g. "ac.at" from "at"
- Zone: A certain domain name
  - → Includes also all subdomains that have not been delegated
- "Glue" record: "A" record that is part of delegation
  - → Required if NS of the delegated domain is within that domain
    - » Chicken-egg-problem!

Requirements for passive/active use

- Passive use: Resolving foreign DNs to IP addresses
  - → At least one namserver required
    - » Usually a "caching" nameserver (see later)
    - » Provided by the ISP
- Active use: Offering to resolve the own DNs to IP addresses
  - → DNs must be registred with "upstream" provider
    - » E.g. Nic.at for *.at; EDVZ of university for *.uni-linz.ac.at
  - → At least one nameserver
    - » Nic.at and most others require at least two NS, however
    - » These should be independent (power, internet connection, ...)
      - – But need not be (not easy to verify!)
    - » This must be an authoritative one
- Nameservers for both are fundamentally different
  - → But can be the same physical server (incl. program)

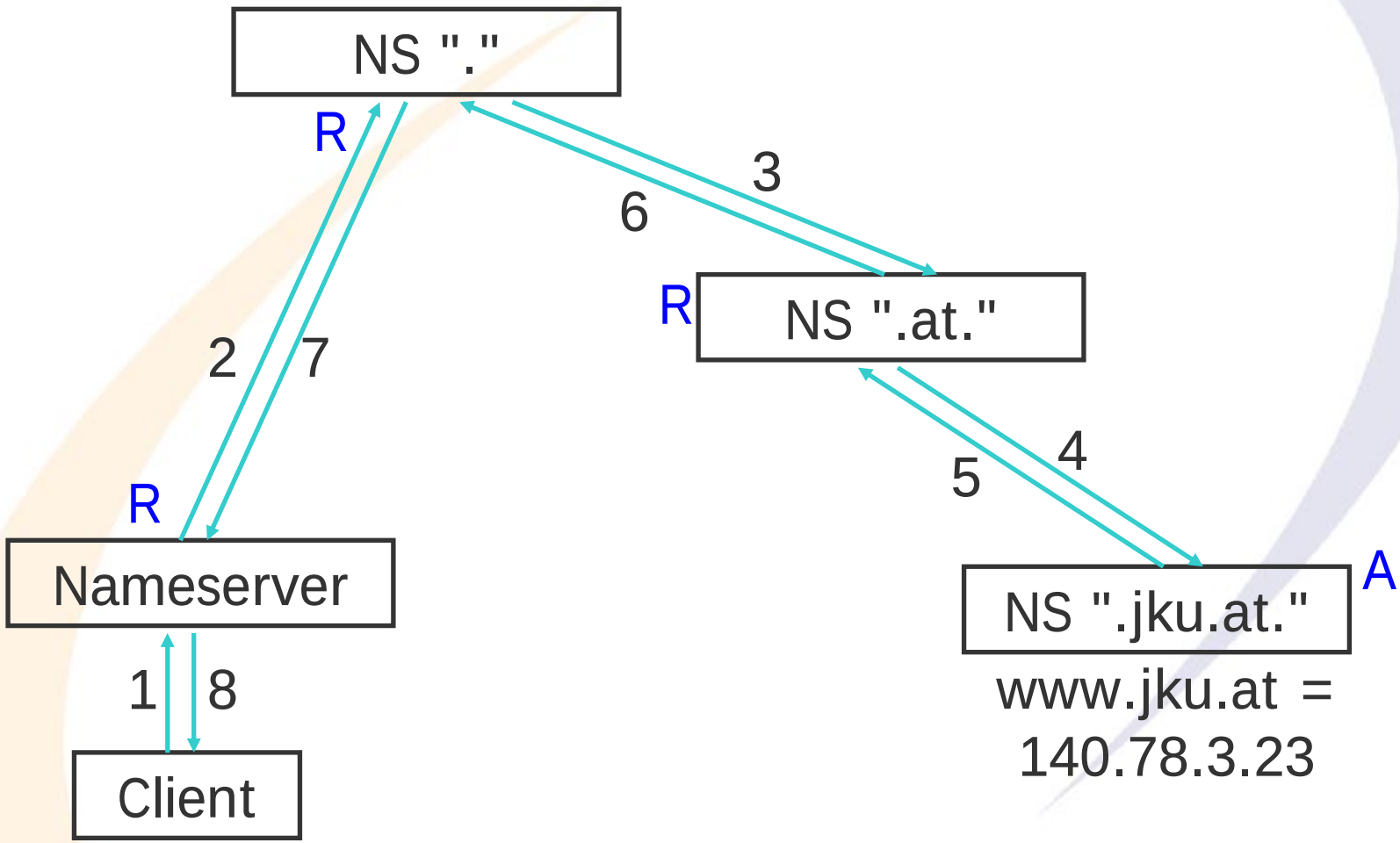
Authoritative vs. caching

- Caching nameserver
  - → Ask other nameservers (caching or authoritative) for resolution of DN and store the information for later use
    - » Time to store is provided along with the information
      - – Therefore any DNS changes might take 3-5 days to "spread" around the whole world!
  - → Caches positive and negative answers
- Authoritative nameserver
  - → Provides the definitive mapping (or other information) for the names within a certain domain
  - → Must actually know all the IP adresses
    - » Therefore a connection to DHCP might be needed
      - – See DNS updates later!
  - → Only one allowed worldwide per domain!

Iterative vs. recursive

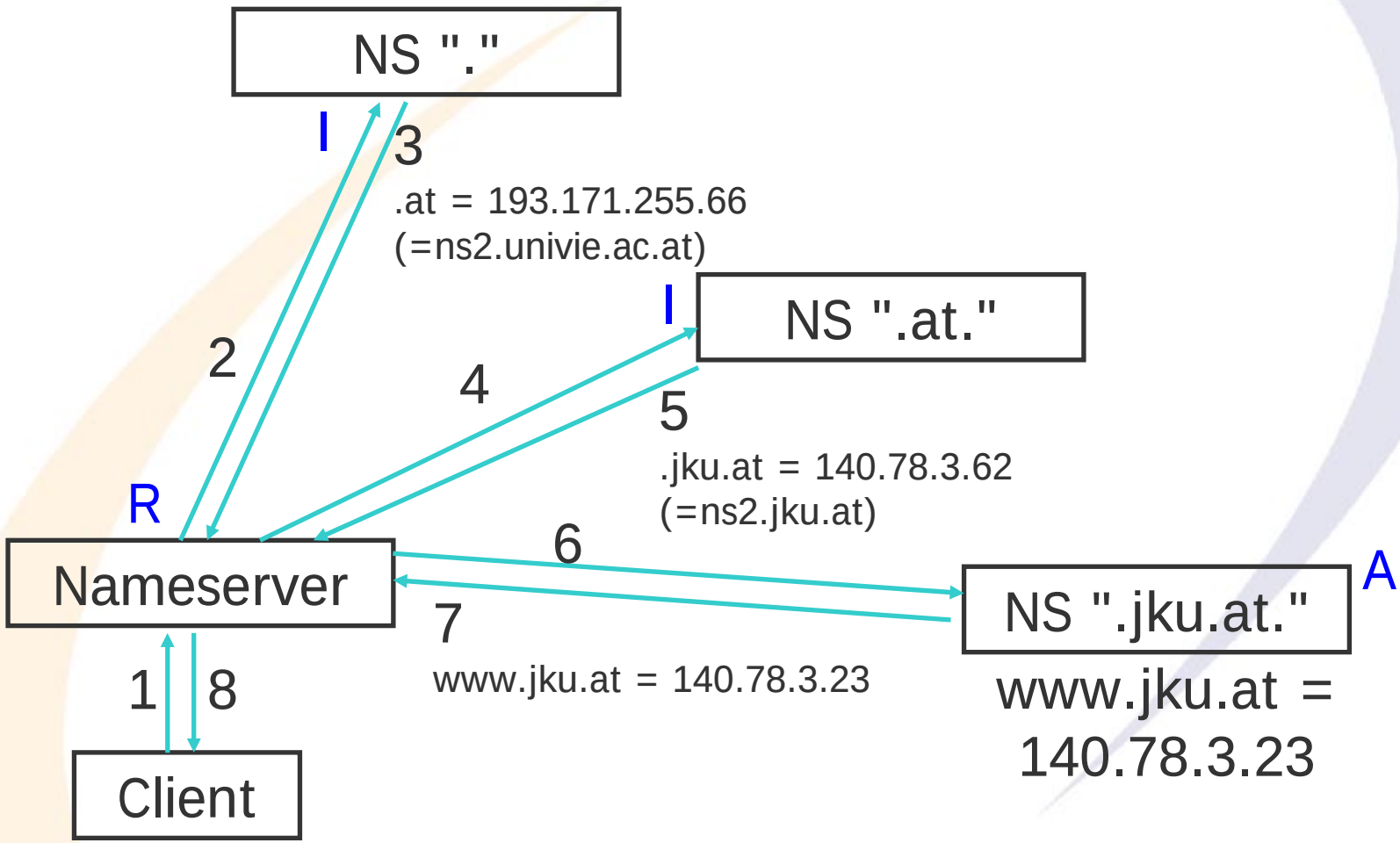
- Recursive query:
  - → Nameserver will give the full answer or return an error
  - → Will ask other nameservers till the answer has been found
  - → This mode is optional
    - » The common mode for the "last" nameserver, which is queried directly by the individual clients!
- Iterative query:
  - → Nameservers will give partial (or full) answers or return errors
  - → Client will have to do lookup with more details himself
    - » Root server tells about ".at", NIC.AT tells about ".ac.at", ...
  - → This mode is required
    - » The common mode for all "higher level" nameservers
- Reverse mapping: Finding the name for a given IP address

DNS lookup: Fully recursive



Please note: This is **not** what the usual configuration is!

DNS lookup: Recursive and iterative



This is the usual configuration!

Static vs. dynamic

- Static DNS: Records stay the same
  - → Changes: Modify the configuration and restart the server
  - → Suitable for many smaller companies
- Dynamic DNS: Records are updated at runtime
  - → Required if servers receive their IP by DHCP
    - » E.g. dialup lines or other low-cost providers!
  - → Or if changes occur frequently (ISPs, ...)
  - → Several dedicated providers available for free
    - » Restricted number of domains usually
  - → Large security risks!
    - » "Poisoning": Injecting wrong data through updates
    - » Use cryptographic means to identify allowed updaters

Registration process & services

- Depends on the registrar
  - → Registrar=Organization responsible for the parent domain
    - » Usually resellers which themselves register at the registrar
      - – Volume discounts, therefore often cheaper than the registrar itself!
- Usually completely online today; procedure itself trivial
- Owner information required
  - → To remain anonymous, trustee needed
- Billing information required
  - → Billing models vary widely: Check and compare!
- Check:
  - → Who is inserted in the other person fields?
  - → Is a nameserver included?
  - → Subdomains possible?

Information needed for registration

- Owner:
  - → Name, Physical address (but might sometimes be a PO box!), Telephone, E-Mail (sometimes optional)
  - → The owner information is published through WHOIS
    - » This is required by the ICANN, so the owner can be identified for inquiries, technical problems, legal proceedings, etc.
- At least two different nameservers must be provided
  - → Could also be done by the registrar
- Except the E-Mail, usually nothing is actually verified
  - → But invoices, information, etc. might also "disappear" then!
- Sometimes additional requirements
  - → E.g. DENIC (.de) requires a german address and a natural person for the administrative contact
    - » The admin-c also MUST provide an E-Mail address

BIND

- Berkeley Internet Name Domain
  - → Domain Name Server (named) (=server side part)
  - → Resolver library (=client side part)
  - → Additional tools for management of the server
- De-facto standard for name server in the internet
- Available for Linux, Unix variants, Windows >=NT
- Current version: 9.3.x
  - → Versions 8 and 4 should not be used any more
    - » Security vulnerabilities, update problems
- Hardware requirements: Very low!
  - → DNSSEC and IPv6 might require some CPU power
  - → Memory requirements generally higher
    - » All the data should fit into memory for fast answers

BIND Installation

- The practical part will cover the Linux version only
  - → Other Unix versions are almost identical
  - → Windows version is very similar
    - » Uses the same data format (but different linefeeds!)
- Installation is extremely simple:
  - → Either use a preconfigured RPM
  - → Or compile and install it manually:
    - » ./configure
    - » make
    - » make install
- The main work is in the actual configuration!

Resolver configuration

- A bit off topic.....
- /etc/resolv.conf
  - → domain mydomain.com Local domain name of the client
  - → nameserver 192.168.2.240 Nameserver to use
  - → nameserver 140.78.100.31 Another nameserver
- Nameservers MUST consist of IP addresses!
  - → Names cannot yet be resolved!
- Optional: "search" provides some other domain names to try to add to relative domain names
- Can also be assigned by DHCP (if configured)!
- Nameserver itself:
  - → Either "nameserver 127.0.0.1"
  - → Or just an empty file

Caching only: /etc/named.conf

```
acl "internal_nets" { 192.168.1.0/24; 192.168.2.0/24; };
options {
    directory "/var/named";
    pid-file "named.pid";
    allow-query { "internal_nets"; };
};

zone "." {
    type hint;
    file "named.ca";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "0.0.127.in-addr.arpa.zone";
    notify no;
};
```

Internal networks, which are served by this server instance

Where the other (zone) configuration files are

Non-authoritative for "."

List of root server names and their IPs

Reverse mapping for the loopback address (127.0.0.1)

Authoritative for "0.0.127.in-addr.arpa"

Name of file with information (="var/named/0.0.127. ...)

Data changes are not propagated to other servers

Authoritative only: /etc/named.conf

```
options {
    directory "/var/named"; pid-file "named.pid";
    allow-query { any; };
    recursion no;
};
zone "." { type hint; file "named.ca"; };
zone "0.0.127.in-addr.arpa" { type master;
    file "0.0.127.in-addr.arpa.zone "; notify no;
};

zone "mydomain.com" {
    type master;
    file "mydomain.com.zone";
    allow-transfer {
        192.168.2.240;
    };
};
```

As authoritative server we must allow the world to query us (or mydomain.com is invisible in some areas!)

We don't do recursive lookups

Local configuration as before

The domain this zone is about

We provide (the one and only) authoritative answers

Name of file with information

Slaves allowed to retrieve the zone file (typically the secondary nameservers for this domain)

BIND: Basic configuration
Split DNS

- Different view on the DNS from two (or more) points
  - → Commonly: Internet view vs. internal view
- Reasons for this:
  - → Hiding the internal structure from the outside
    - » Not necessarily as easy: See e.g. E-Mail headers!
  - → Resolving questions from the internal network and providing authoritative answers to the outside at the same time
    - » Caching and recursive lookup for internal computers
    - » Authoritative and iterative lookup for external computers
- Implementation possibilities
  - → Two different DNS servers
    - » Usuall one internal/DMZ and one on firewall
  - → One server with different views (BIND >=9)

BIND
Zone files

- Zone files contain the actual information DN ↔ IP
  - → Some kind of database (other software uses real databases!)
- For each name there exists several pieces of data
  - → This is called a RR (Resource Record)
  - → The order is undefined and unimportant
    - » Queries are expected to decide upon the ordering within the same class (these are deemed equivalent or contain additional information) or request specific classes
- RRs consist of
  - → Owner name: The Domain Name it belongs to
  - → Type: Type of record (see next slide)
  - → TTL: Time To Live (allowed maximum time of caching)
  - → Class: Protocol or instance (currently only one: IN = Internet)
  - → RDATA: Specific data for the record

Kinds of records: RR types

- Common records:
  - → A Host address
  - → A6 IPv6 host address
  - → CNAME canonical name for an alias
  - → MX Mail exchanger for the domain
  - → NS Auhtoritative nameserver for the domain
  - → PTR Pointer to another part of DN space
  - → SOA Start Of Authority
  - → TXT Text record
- Uncommon records:
  - → CERT, KX: Public key cryptography settings
  - → SIG, NXT, KEY: DNSSEC information
  - → LOC GPS info of computer
- Other records also available

Zone file example

```
$ORIGIN .                   ; Default "extension" for all names
$ TTL       86400           ; Default TTL: 1 day
; This is the SOA (=authoritative info) for "msv.at"
msv.at      IN      SOA   dns.msv.at.  michael.sonntag.msv.at.  (
        0509101827 ; Serial number: Important for updates!!!!!
        28800      ; Refresh after 8 hours (check for updates)
        7200       ; Retry after 2 hours (secondary NS will retry first NS)
        604800     ; Expires after 1 week (secondary NS stops answering)
        86400      ; Minimum TTL is 1 day (how long queries are "valid")
)
            NS          dns.msv.at.         ; Nameserver of domain
            MX    10    mail.msv.at.        ; Mail exchanger for domain
            TXT         "MSV Handels- & Dienstleistungs GmbH"
$ORIGIN msv.at.
firewall    A           192.168.2.240       ; IP address of firewall
pdc         A           192.168.2.3         ; Another computer
mail        CNAME       firewall            ; Just another name
```

MX records

- MX records are special records specifying the mail server(s) for a domain
  - → This contains an additional priority
  - → Priority decides the order in which servers are tried
    - » Lowest priority is tried first
    - » Equal priority: Chosen randomly
- The name of the actual server must be an A record!
  - → Specifying a CNAME is not allowed!
- Example: Random selection of mail or mail2; if neither works, mail.backup.org is tried

example.com.	IN	MX	10	mail.example.com.
	IN	MX	10	mail2.example.com.
	IN	MX	20	mail.backup.org.
mail.example.com.	IN	A	192.170.0.1	
mail2.example.com.	IN	A	192.170.0.2	

Reverse lookup files

- For retrieving the name for a certain IP address
  - → Supported within the DNS through a specific naming scheme
  - → Important for E-Mail: Most mail servers only accept mail from computers where the IP resolves to the name they stated upon opening the connection!
- Method: in-addr.arpa domain
  - → mail.msv.at. ⇒ 192.168.2.240
  - → 240.2.168.192.in-addr.arpa. ⇒ mail.msv.at.
    - » 240.2. ... is a domain name and not an IP address!
- Reverse mapping is done by "PTR" records:
  - → 240.2.168.192.in-addr.arpa. PTR mail.msv.at.
- Zone file looks the same, is just for a "different" domain
  - → 2.168.192.in-addr.arpa.

Reverse lookup file example

```
$ORIGIN .                       ; Default "extension" for all names
$ TTL          86400            ; Default TTL: 1 day
; This is the SOA (=authoritative info) for "msv.at"
2.168.192.in-addr.arpa  IN SOA  firewall.msv.at. michael.sonntag.msv.at. (
        243             ; Serial number: Important for updates!!!!!
        28800           ; Refresh after 8 hours (check for updates)
        7200            ; Retry after 2 hours (secondary NS will retry first NS)
        604800          ; Expires after 1 week (secondary NS stops answering)
        86400           ; Minimum TTL is 1 day (how long queries are "valid")
)
                NS              dns.msv.at.             ; Nameserver of domain
$ORIGIN 2.168.192.in-addr.arpa.                         ; Note reverse order of IP!
240             PTR             mail.msv.at.            ; IP .240 has several names!
                PTR             firewall.msv.at.
3               PTR             pdc.msv.at.             ; Just another mapping
```

Reverse lookup for CIDR

- This backwards resolution of IP addresses works only for complete class subnets (i.e. class A, B or C)
  - → This is not desirable, as many companies only get smaller IP address space parts (e.g. 8, 16 or 32 IP addresses)!
  - → Every change in reverse lookup would then be made by the ISP, although the forward lookup is done by the client itself!
- Requires a CNAME record in the ISP reverse zone file
  - → Instead of a PTR to a name, insert a CNAME to s specially crafted reverse domain name
    - » Example: Subnet 192.168.23.64/27 (=.64-.95)
    - » 64/27 IN NS dns.example.com.
    - » 65 IN CNAME 65.64/27.23.168.192.in_addr.arpa.
  - → In the local reverse zone file is then the final resolution
    - » 65 IN PTR host.example.com.

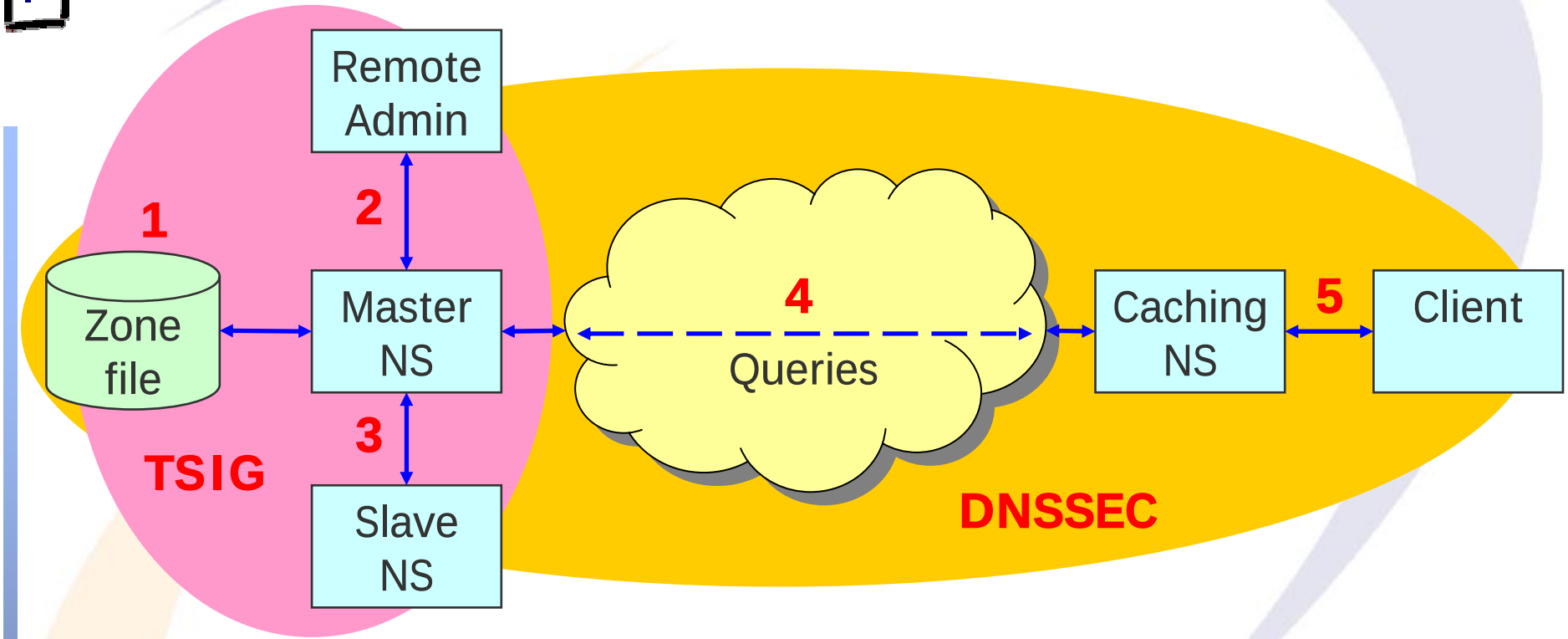
Split DNS

- Split (or stealth, DMZ) DNS: Two separate DNS views
  - → Outside:
    - » Can see only public computers: Typically mail and web server
    - » No recursion allowed, no caching done
    - » Typically an "authoritative only" NS
  - → Inside:
    - » Can see internal and public computers
    - » Recursion allowed, caches responses from the Internet
    - » Typically a "master" NS
- Using BIND both can be combined on one computer/NS
  - → Use different views
    - » This might pose some security risks
  - → This nameserver must run on the firewall or in the DMZ
    - » Must be accessible from both inside and outside

Security considerations

- DNS is a very important service and therefore a prime target
  - → Imagine e.g. redirecting requests to "amazon.com" to your own web server....
  - → Denial of service: Almost every service uses DN and not IPs!
  - → Gathering network information: Which computers exist, gues their function from their name, impersonating them, ...
- Even more important when considering replication:
  - → Many (esp. secondary) NS receive their data from other NSs
    - » What about modifying/preventing this transfer?
- Security measures:
  - → Only root needs access to configuration files
  - → Restricting queries (this is not foolproof!)
  - → Chroot environment
  - → Encrypting/Signing transfers

Security considerations



1: Zone file: Corruption, modification (local administration)

2: Dynamic updates: Unauthorized, IP address spoofing (TSIG)

3: Zone transfer: IP address spoofing (TSIG)

4: Remote queries: Cache poisoning, interception, subverted master/slave (DNSSEC)

5: Resolving: (As remote queries), IP spoofing (DNSSEC)

Restricting queries

- Access control lists determine which computers may do what: query, transfer, notifications, recursion, etc.
  - → Note: This is based on IP addresses!
    - » If IP spoofing is possible, this won't help a bit!
    - » Firewall configuration therefore important
- This is defined in the configuration, not in the zone files
- Example:

```
acl "internal_nets" { 192.168.0.0/16; localhost; };
acl "bogusnets" {10.0.0.0/8; 172.16.0.0/12; 169.254.0.0/16; 0.0.0.0/8; };
options {
        allow-query { "internal_nets"; };        # Querying allowed
        allow-recursion { "internal_nets"; };  # Recursive requests allowed
        blackhole { "bogusnets"; };     # No queries, not used for resolving
}
```

Dynamic updates

- When computers receive IP addresses dynamically (e.g. DHCP), these values must also be sent to the NS
  - → This is again a possible security hole: Anyone could introduce any record into a nameserver!
  - → Also important for non-stop nameservers: Modifying the zone file usually requires restarting the NS
- Slave NS receive their data from master NS
  - → Is it really the master talking?
  - → Ist it really the slave listening?
- Dynamic updates covers two orthogonal concepts
  - → TSIG: Content update and communication between NS
  - → DNSSEC: NS to client communication

Please note: This is different fom DynDNS!

Signing DNS : DNSSEC

- DNSSEC= DNS SECurity
  - → For secure communication between nameservers and clients
- Solves:
  - → Data integrity: No modifications possible in transit
  - → Source spoofing: Originator of the information is verified
- Limitations:
  - → No encryption: All transmitted data is public
  - → No protection against DoS, buffer overruns, etc.
- Rarely used:
  - → Server support missing or complicated
    - » E.g. requires the use of NTP for exact time!
  - → Key rollover (changing the keys) difficult
  - → Client support? ...
- Still in flux (new standard from 15.10.2004)

Signing DNS: DNSSEC

- DNSSEC is a public-private key method
  - → The master nameserver signs the data with his private key
  - → Clients can verify the data authenticity
- It does not use certificates (no PKI)!
  - → Therefore the public/secret key must be retrieved securely....
  - → This key is another resource record ("KEY")
    - » Manual configuration or other sources (e.g. LDAP) possible
  - → Hierrachically: Parent zone signs key of child zone
    - » Root server keys are therefore VERY important!
    - » Root keys must be known; similar to root server IPs
- Each single resource record is signed with the private key
  - → Perhaps only a single record will be queried!
  - → This are again resource records ("SIG")
    - » This is automatically attached to each response

Dynamic updates: TSIG

- TSIG = Transaction SIGnatures
  - → For secure communication between nameservers
  - → Uses symmetric key to sign data exchanged
- Solves: Modifying data in transit
  - → From DHCP or during zone transfer
  - → For dynamic updates/administration
- Limitations:
  - → Shared secret: Only usable in small and closed systems
  - → Open to brute force attacks: Regularly change the key
    - » No key exchange provided for!
      - – See TKEY or other alternatives
- Implementation: Standard
  - → Also quite easy to set up
    - » Generate the keys and add a few configuration statements

DNS and firewalls

- DNS usually uses single UDP packets and a single port: 53
  - → Still, this might pose difficulties!
  - → E.g. different versions of BIND use different local ports to query other nameservers
    - » BIND <=8 connects port 53 to port 53
    - » BIND >=9 connects from an unprivileged port (>=1024) to 53
- If the query/response is too large, TCP is used however!
  - → On the same ports, at least!
  - → Note, depending on the type of server only outgoing connections might be required
- Domain name servers have a reputation for being buggy
  - → Many bugs found in almost all implementations
  - → Restricting the communication might not be sufficient!

Literature

- BIND
  http://www.isc.org/index.pl?/sw/bind/
  http://www.bind9.net/manuals
- DNS for Rocket Scientists
  http://www.zytrax.com/books/dns/
- Linux Home Networking (Chapters 6 and 7)
  http://www.siliconvalleyccie.com/#Linux
- DNS Security Extensions Website
  http://www.dnssec.net/
- Haddad/Gordon: The Basics of DNSSEC
  http://www.onlamp.com/pub/a/onlamp/2004/10/14/dnssec.html
- DNSSEC Operational HOWTO
  http://www.ripe.net/disi/Course/TechCourse.pdf