Networking Basics

06a - Domain Name System (DNS)

Wolfgang Tremmel
academy@de-cix.net

DE-CIX

Where networks meet

Networking Basics

DE-CIX Academy

01 - Networks, Packets, and Protocols

02 - Ethernet, 02a - VLANs

03 - IP, 03a - Routing, 03b - Global routing

04a - User Datagram Protocol (UDP)

04b - TCP

04c - ICMP

05 - Uni-, Broad-, Multi-, and Anycast

06a - Domain Name System (DNS)

DE CIX

Networking Basics

DE-CIX Academy

01 - Networks, Packets, and Protocols

02 - Ethernet, 02a - VLANs

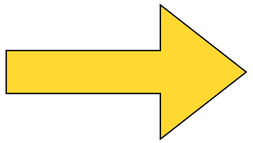
03 - IP, 03a - Routing, 03b - Global routing

04a - User Datagram Protocol (UDP)

04b - TCP

04c - ICMP

05 - Uni-, Broad-, Multi-, and Anycast



06a - Domain Name System (DNS)

DE-CIX

Some history...

Telephone

Some history...

Telephone

- Early telephones only used numbers

Attribution: https://commons.wikimedia.org/wiki/File:Telefon_BW_2012-02-18_13-44-32.JPG

DE CIX

Some history...

Telephone

- Early telephones only used numbers
- If you did not know the number to call, you had to look it up

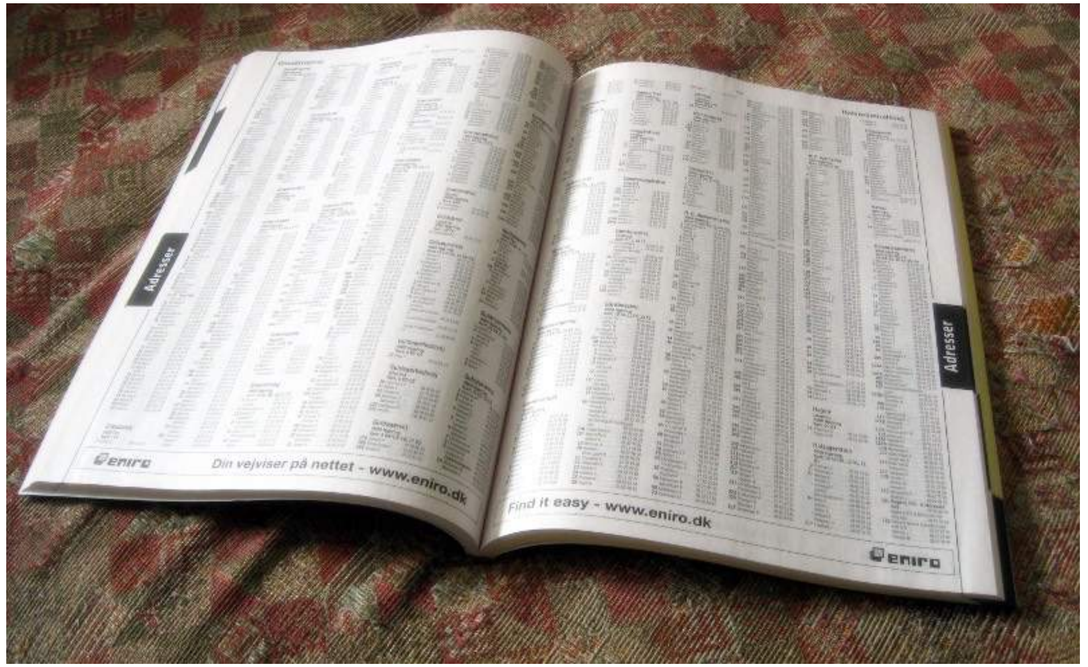
Attribution: https://commons.wikimedia.org/wiki/File:Telefon_BW_2012-02-18_13-44-32.JPG

DE CIX

Some history...

Telephone

- Early telephones only used numbers
- If you did not know the number to call, you had to look it up
- In a phonebook



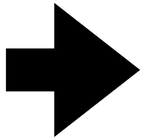
Attribution: https://commons.wikimedia.org/wiki/File:Telefonbog_ubt-0.JPG

DE CIX

Internet Model

IP / Internet Layer

- Data units are called "Packets"
- Provides source to destination transport



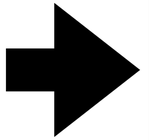
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Internet Model

IP / Internet Layer

- Data units are called "Packets"
- Provides source to destination transport
  - For this we need addresses
  - Like 192.0.2.0.1 or 2001:db8:34:22ff:fe55::1



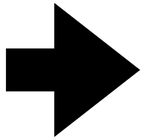
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Internet Model

IP / Internet Layer

- Data units are called "Packets"
- Provides source to destination transport
  - For this we need addresses
  - Like 192.0.2.0.1 or 2001:db8:34:22ff:fe55::1
  - Hard to remember

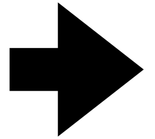


Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Internet Model

Users do not like numbers



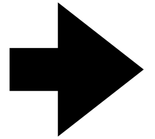
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Internet Model

Users do not like numbers

- In applications like a web browser, users do not want to enter numbers



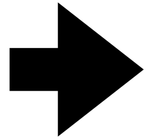
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Internet Model

Users do not like numbers

- In applications like a web browser, users do not want to enter numbers
- They want to enter names, like www.de-cix.net



Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

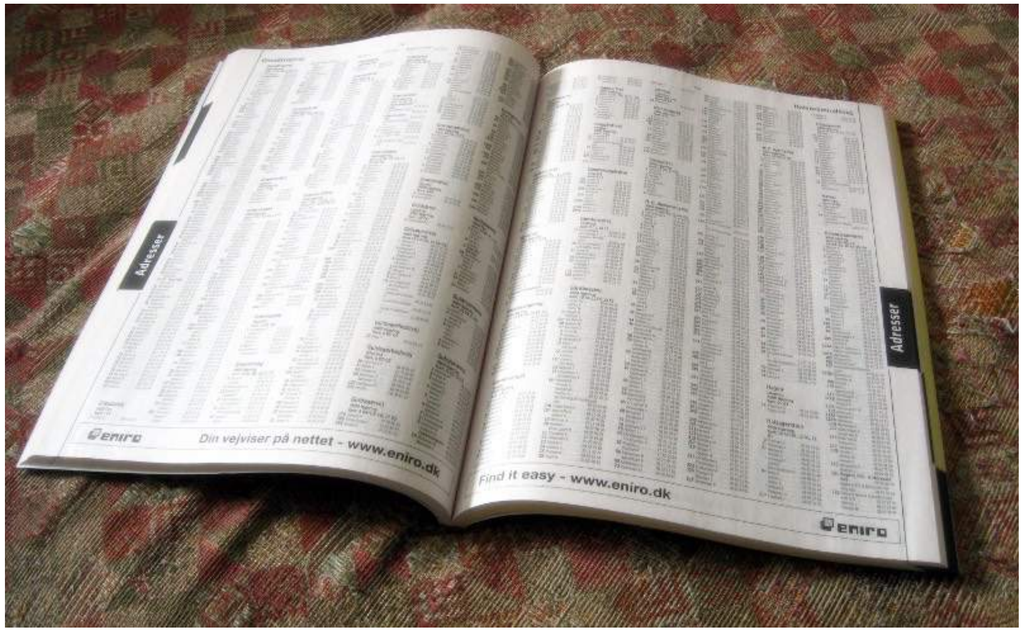
DE CIX

Internet Model

Users do not like numbers

- In applications like a web browser, users do not want to enter numbers
- They want to enter names, like www.de-cix.net
- **So we need a Phonebook for the Internet**

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical



Attribution: https://commons.wikimedia.org/wiki/File:Telefonbog_ubt-0.JPG

Phonebook for the Internet

Some history

Phonebook for the Internet

Some history

- Easy: Just have a text file, with names and addresses

Phonebook for the Internet

Some history

- Easy: Just have a text file, with names and addresses
  - This is how it actually was done in the beginning
  - The file name was */etc/hosts* and it still exists on a lot of systems

DE CIX

Phonebook for the Internet

Some history

- Easy: Just have a text file, with names and addresses
  - This is how it actually was done in the beginning
  - The file name was */etc/hosts* and it still exists on a lot of systems
  - When the file had more than 7000 entries in the 1990s, this was discontinued

Phonebook for the Internet

Some history

- Easy: Just have a text file, with names and addresses
  - This is how it actually was done in the beginning
  - The file name was */etc/hosts* and it still exists on a lot of systems
  - When the file had more than 7000 entries in the 1990s, this was discontinued
- Solution: Have a distributed database mapping names to numbers
  - The Domain Name System (DNS) was specified in 1987

Domain Name System

DNS

DE CIX

Domain Name System

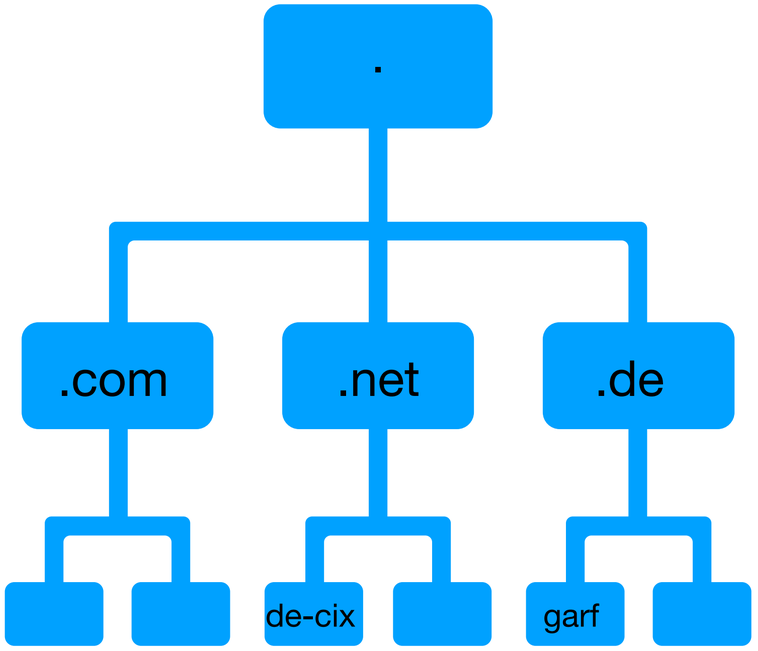
DNS

- Work began earlier: First defined in RFC882 in 1983

DE CIX

Domain Name System
DNS

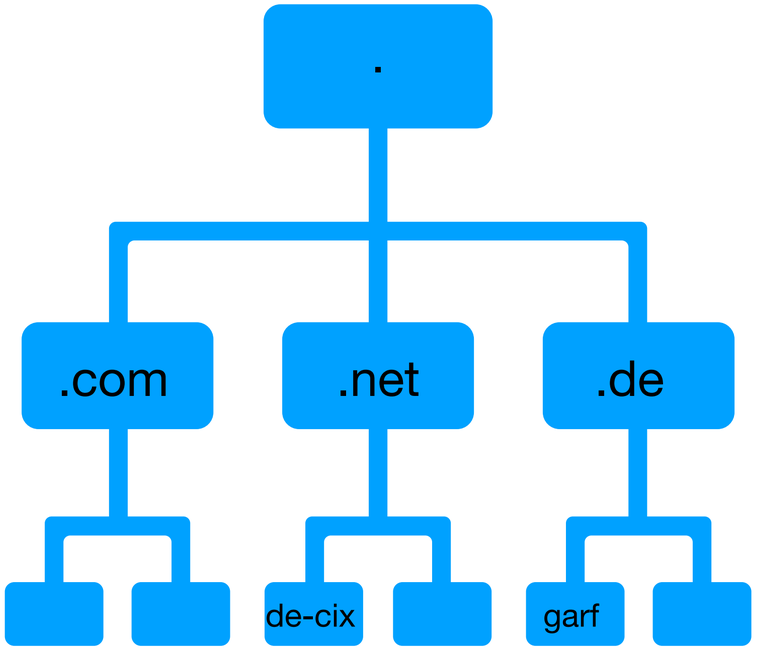
- Work began earlier: First defined in RFC882 in 1983
  - Idea: Not a single name space, but "domains"
  - Distributed (but with local caching)
  - "Tree" structured



DE CIX

Domain Name System
DNS

- Work began earlier: First defined in RFC882 in 1983
  - Idea: Not a single name space, but "domains"
  - Distributed (but with local caching)
  - "Tree" structured
- Original top-level domains were:
  ARPA, COM, EDU, GOV, MIL, and ORG (RFC920)
  - Country domains were also specified, NET came later
  - .de was registered on 1986-11-05 (IANA)



DE CIX

Domain Name System

Where does it fit?

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Domain Name System

Where does it fit?

- Originally, DNS used only UDP

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Domain Name System

Where does it fit?

- Originally, DNS used only UDP
- TCP was added 1987 in RFC1035

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Domain Name System

Where does it fit?

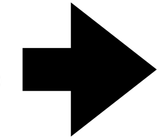
- Originally, DNS used only UDP
- TCP was added 1987 in RFC1035
- Both UDP and TCP are Transport Layer

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

Domain Name System

Where does it fit?

DNS 

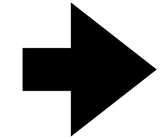
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

- Originally, DNS used only UDP
- TCP was added 1987 in RFC1035
- Both UDP and TCP are Transport Layer
  - → so DNS belongs to the application layer

DE CIX

Domain Name System

Where does it fit?

DNS 

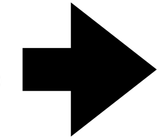
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

- Originally, DNS used only UDP
- TCP was added 1987 in RFC1035
- Both UDP and TCP are Transport Layer
  - → so DNS belongs to the application layer
- New standards also allow
  - HTTPS (RFC8484)
  - TLS (Transport Level Security) - RFC7858

DE CIX

DNS over UDP

As originally defined

DNS 

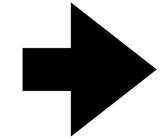
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

DNS over UDP

As originally defined

- Well-known port 53 is used

DNS 

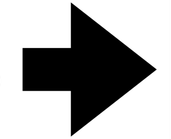
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

DNS over UDP

As originally defined

- Well-known port 53 is used
- Originally the idea was that DNS packets should be *512 bytes or less* (to avoid *fragmentation*)

DNS 

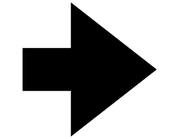
Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

DNS over UDP

As originally defined

- Well-known port 53 is used
- Originally the idea was that DNS packets should be *512 bytes or less* (to avoid *fragmentation*)
- Today larger packets are possible (but need to be fragmented)

DNS 

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

DE CIX

DNS over UDP

As originally defined

- Well-known port 53 is used
- Originally the idea was that DNS packets should be *512 bytes or less* (to avoid *fragmentation*)
- Today larger packets are possible (but need to be fragmented)
    - A lot of stuff was added to DNS over the years - and packet fragmentation still can cause problems

DNS ➡

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

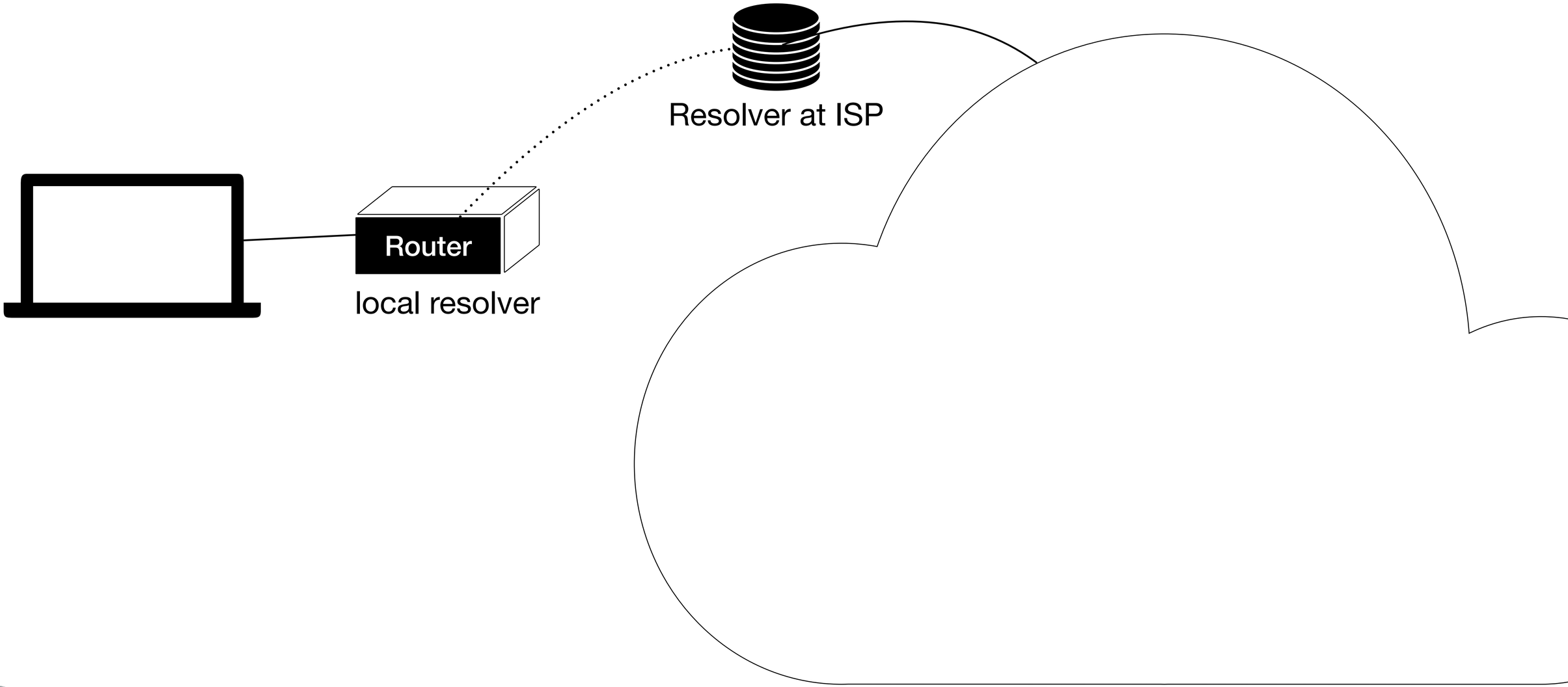
DE CIX

DNS - How does it work?

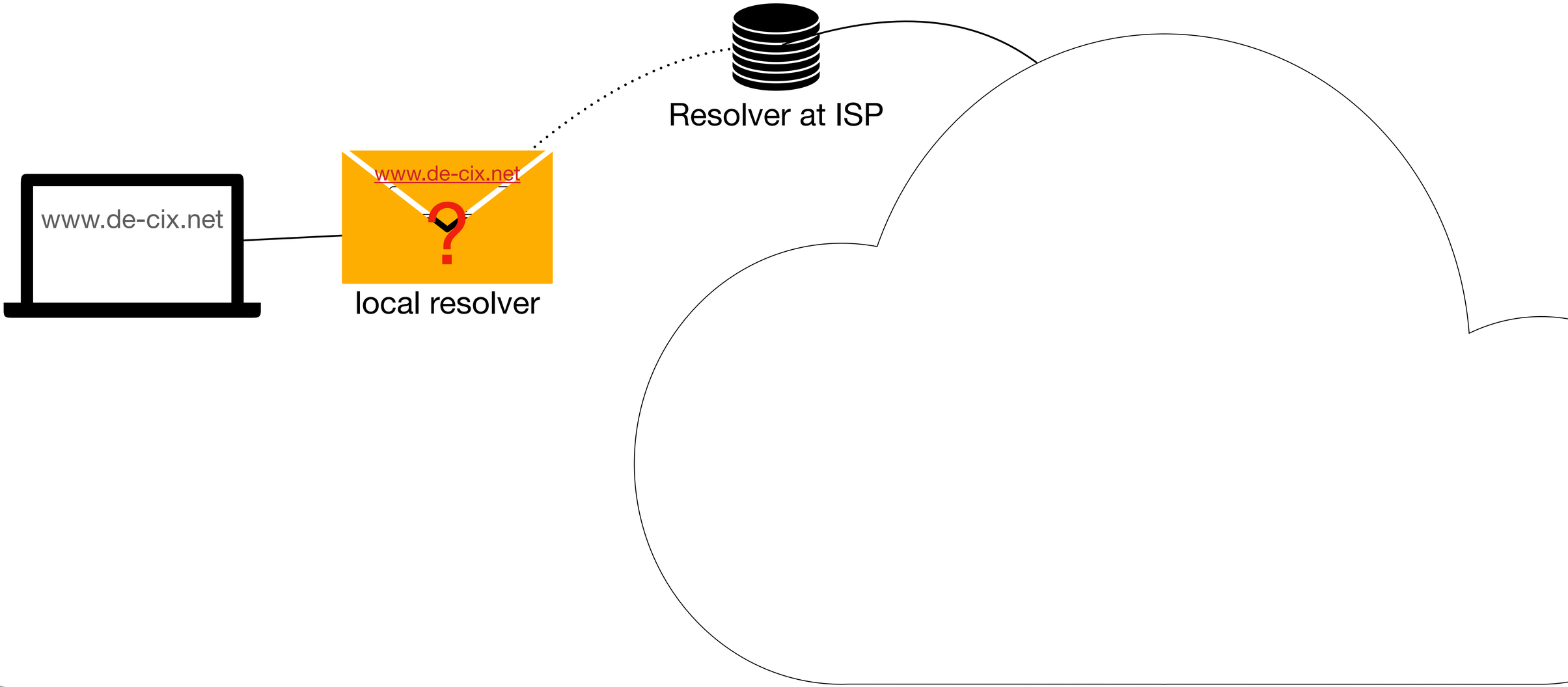
DNS - How does it work?

1. Queries

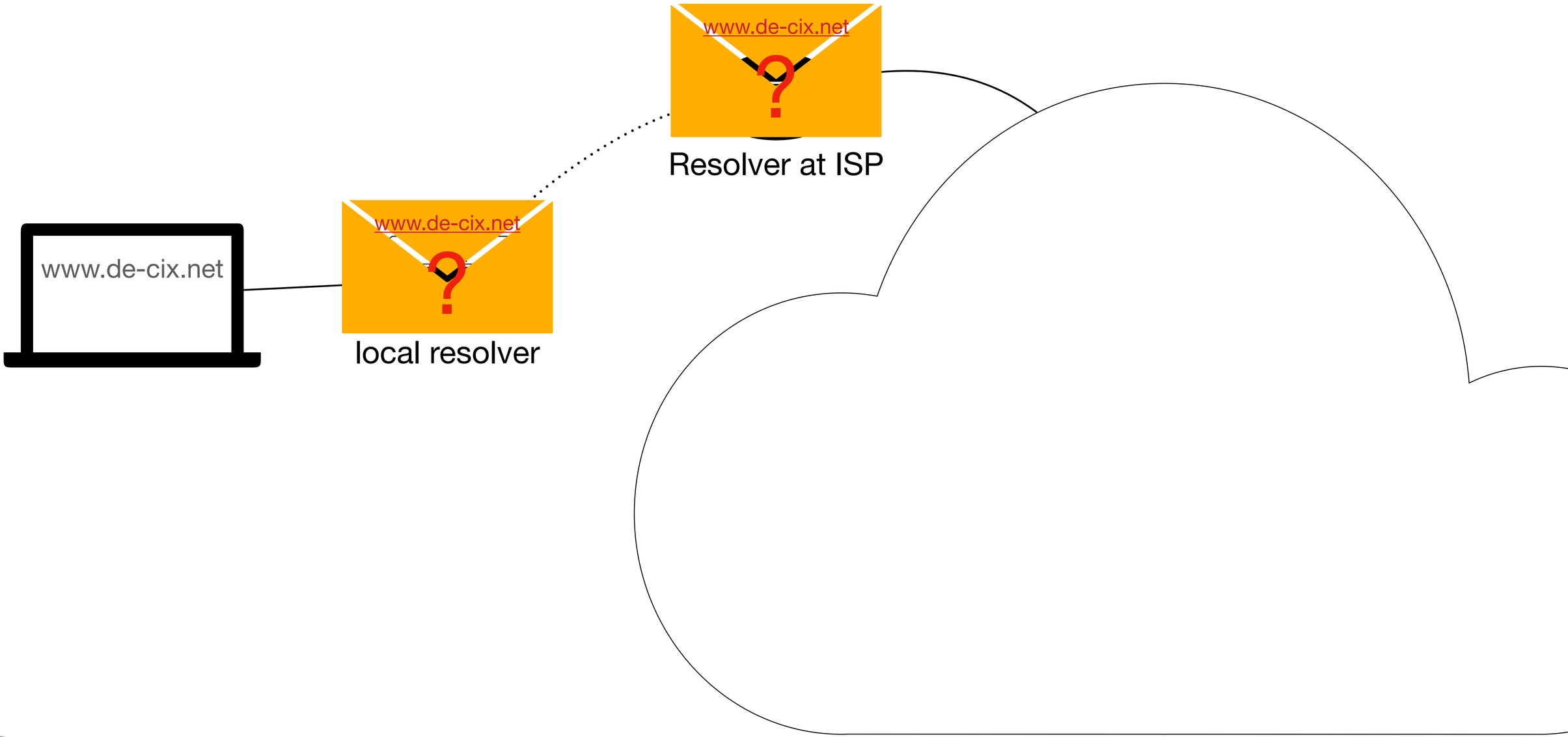
DNS - How does it work?



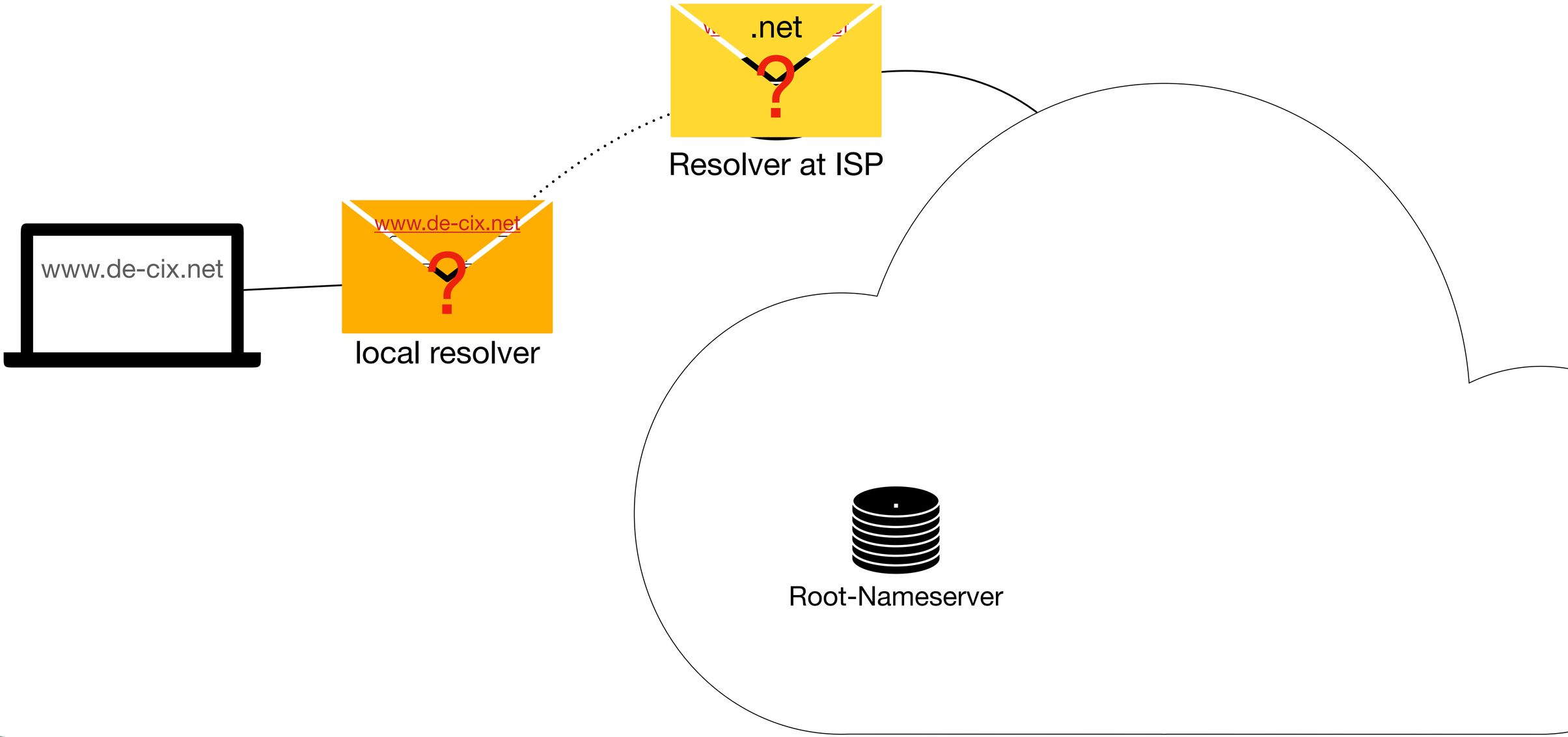
DNS - How does it work?



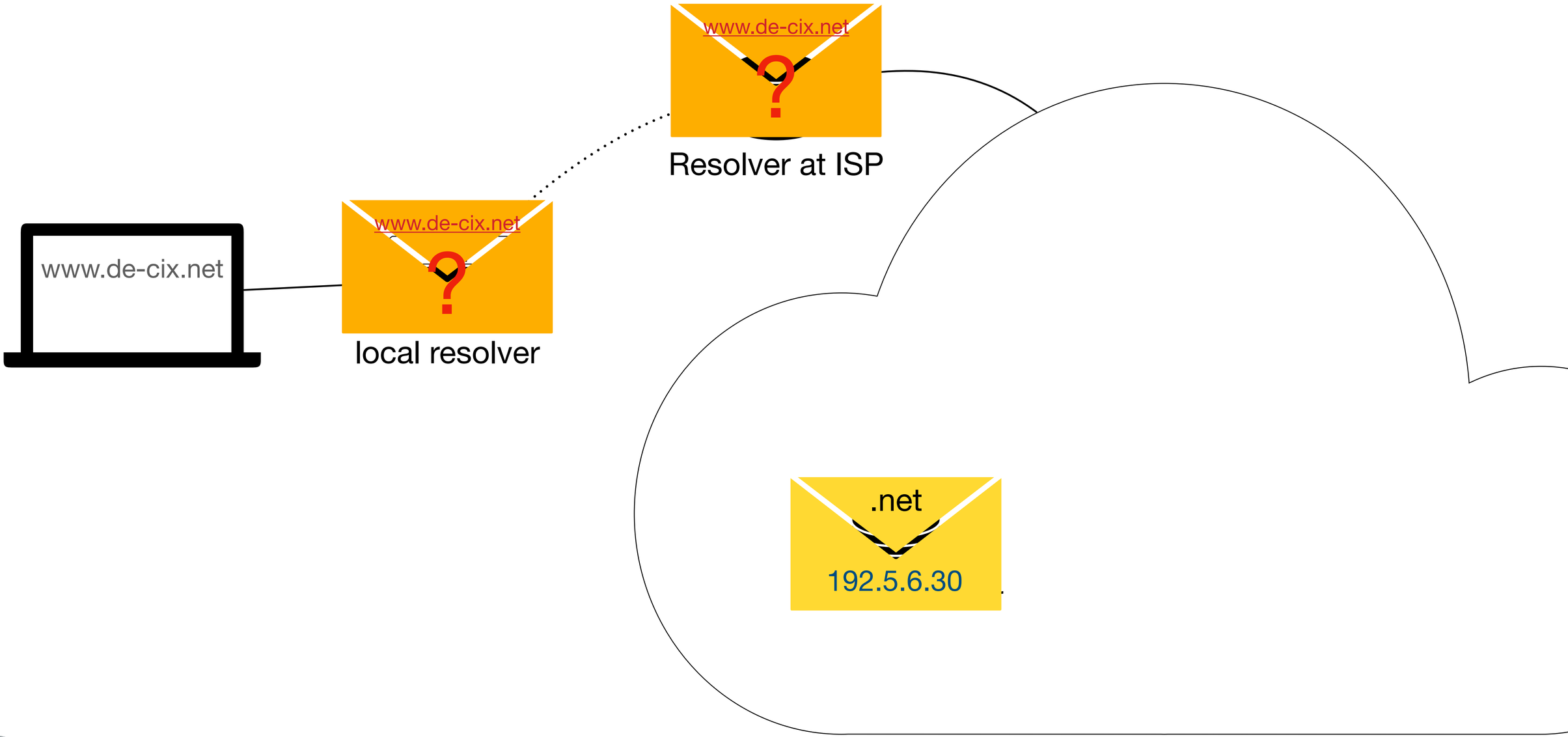
DNS - How does it work?



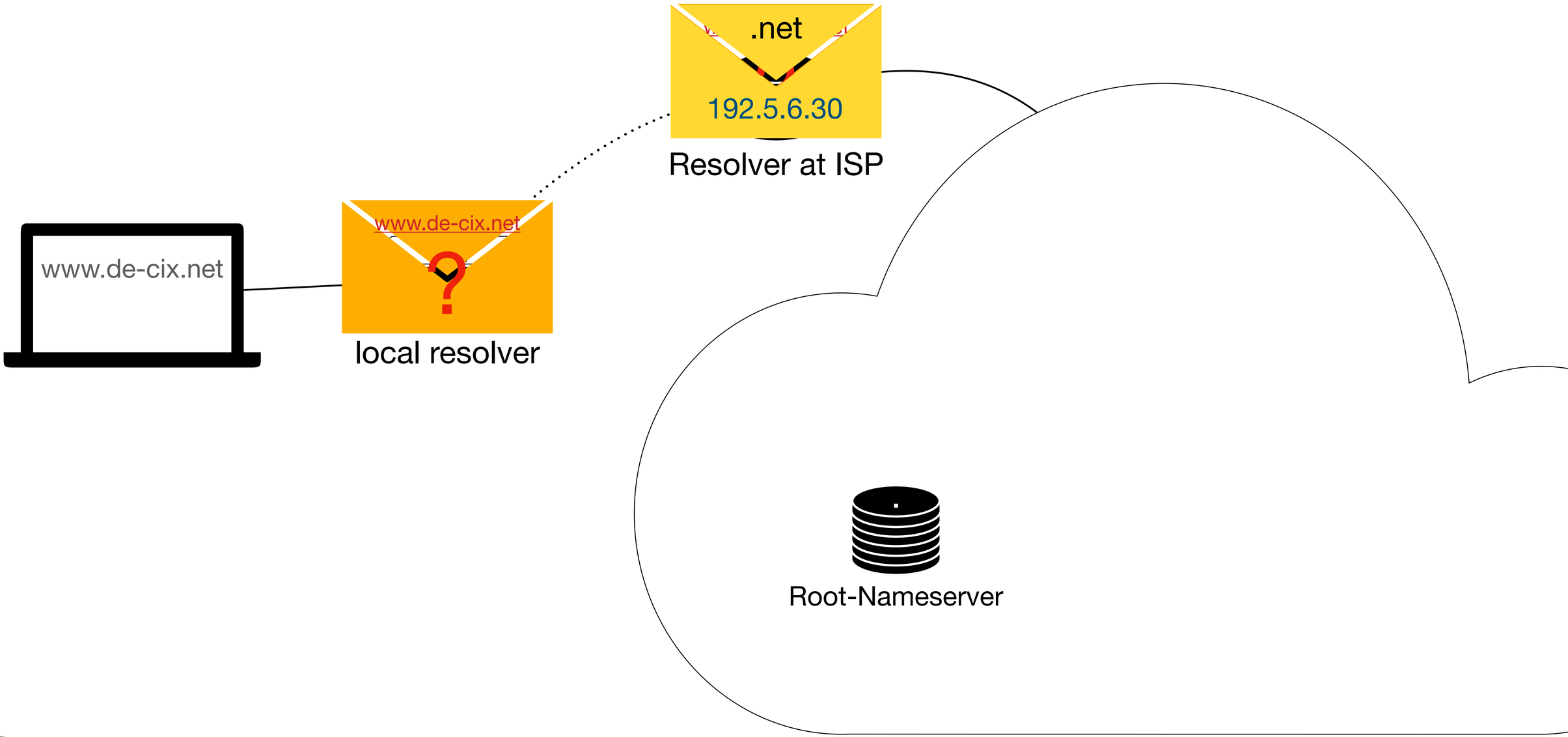
DNS - How does it work?



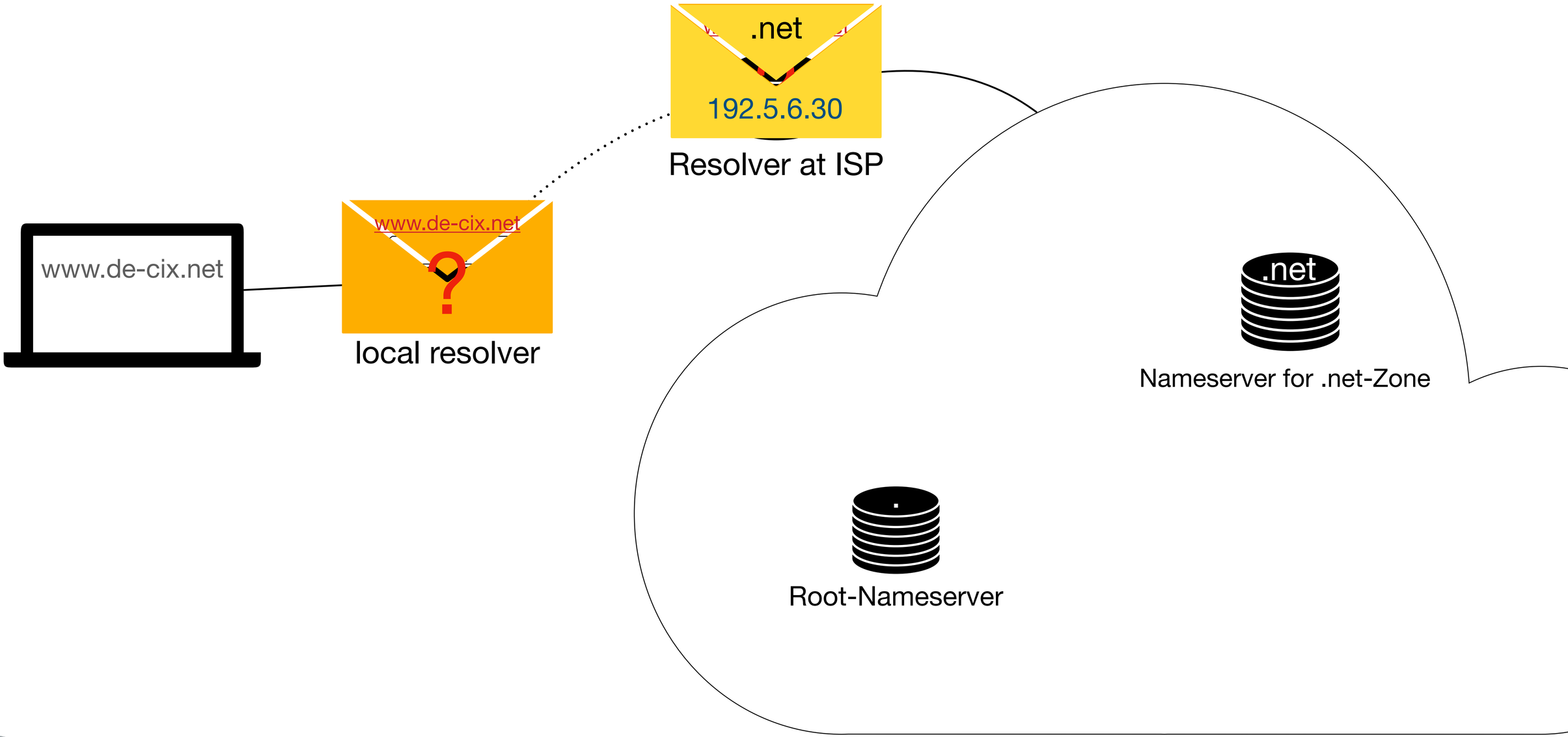
DNS - How does it work?



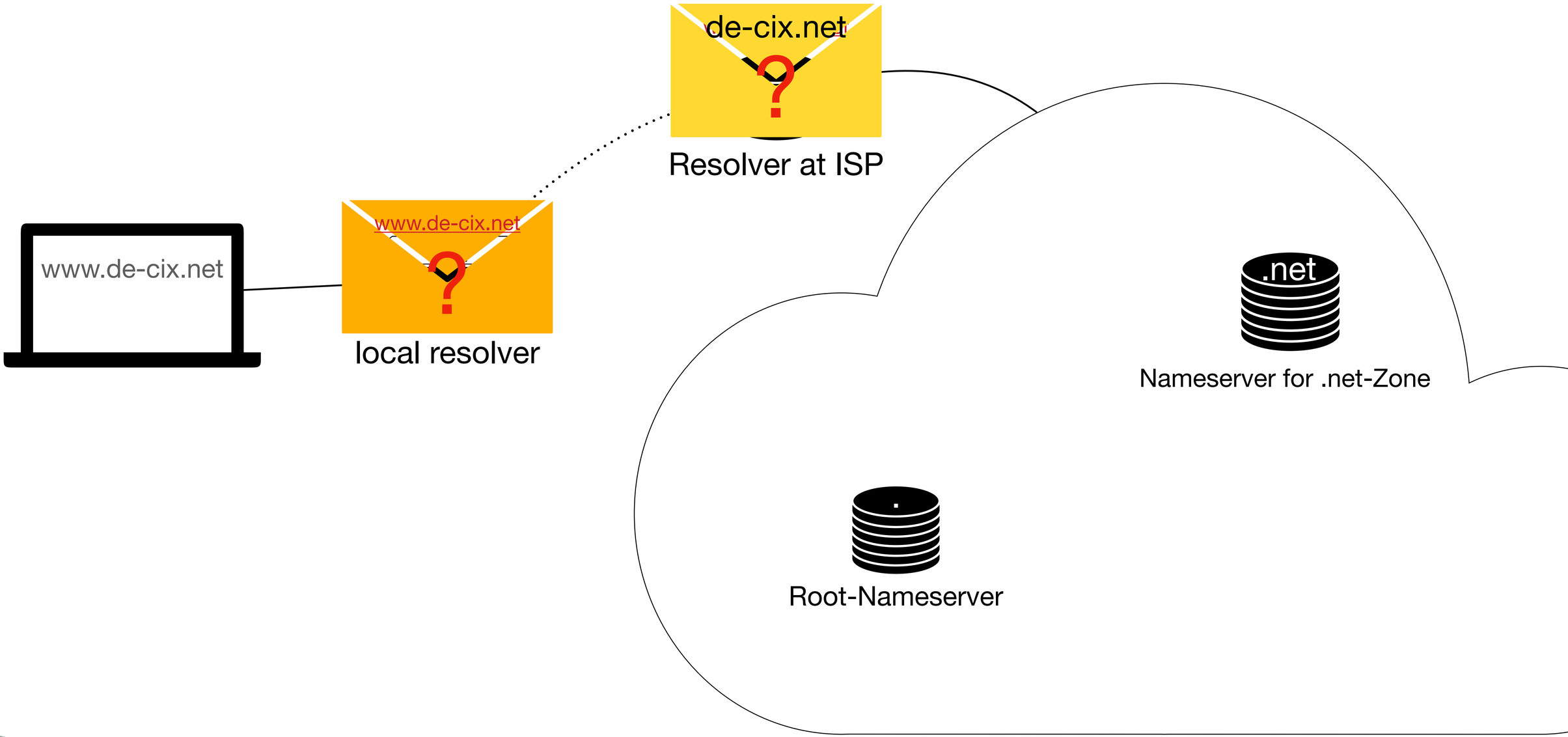
DNS - How does it work?



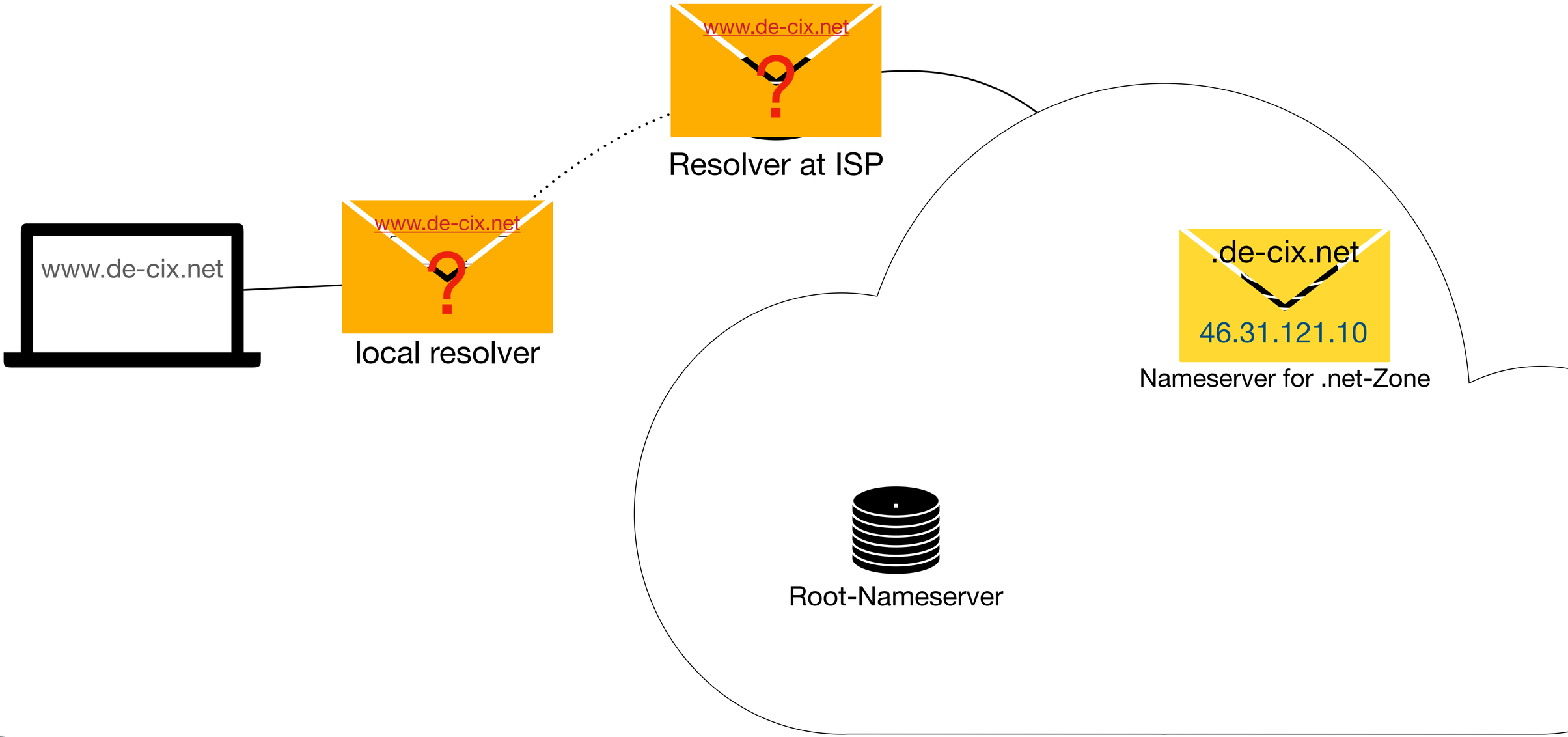
DNS - How does it work?



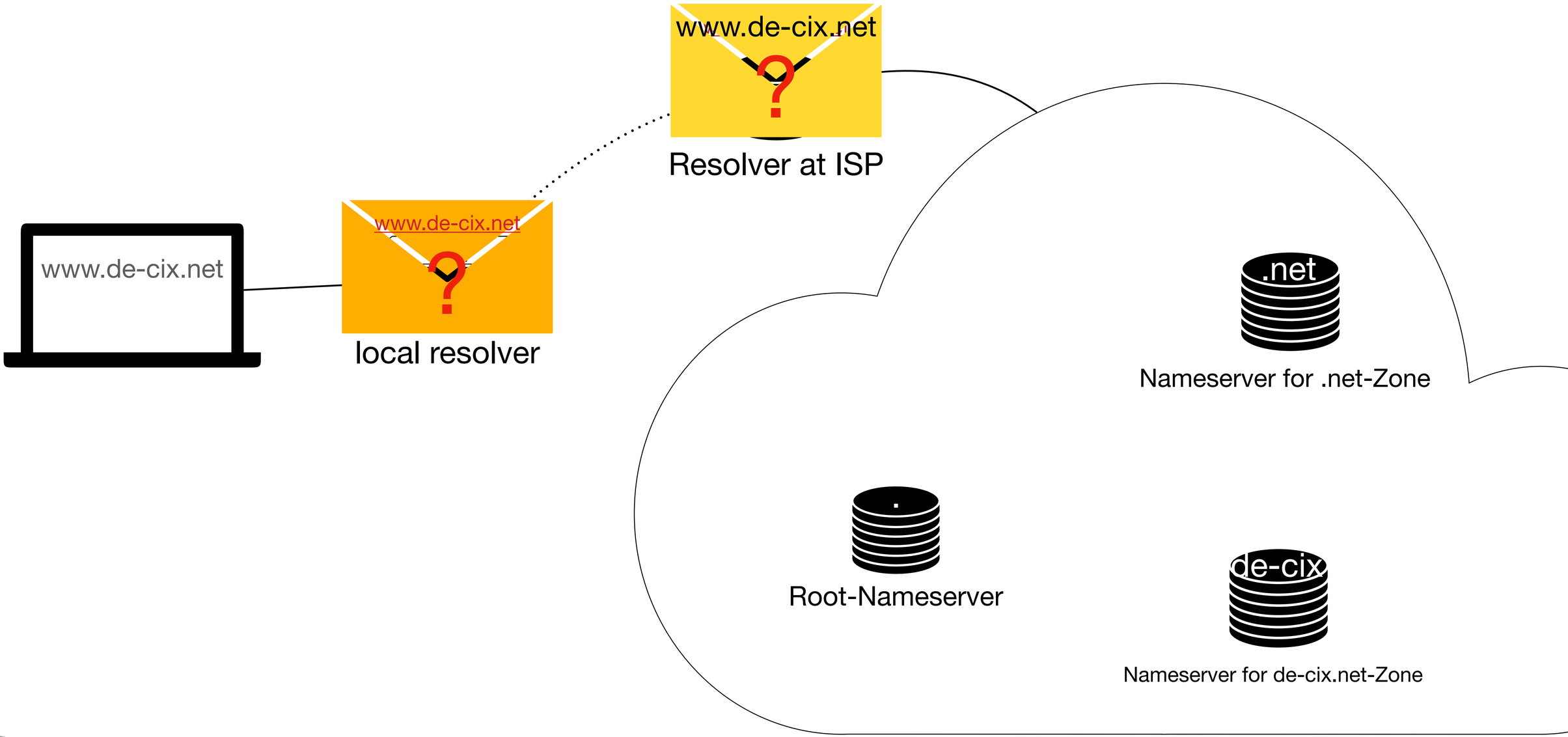
DNS - How does it work?



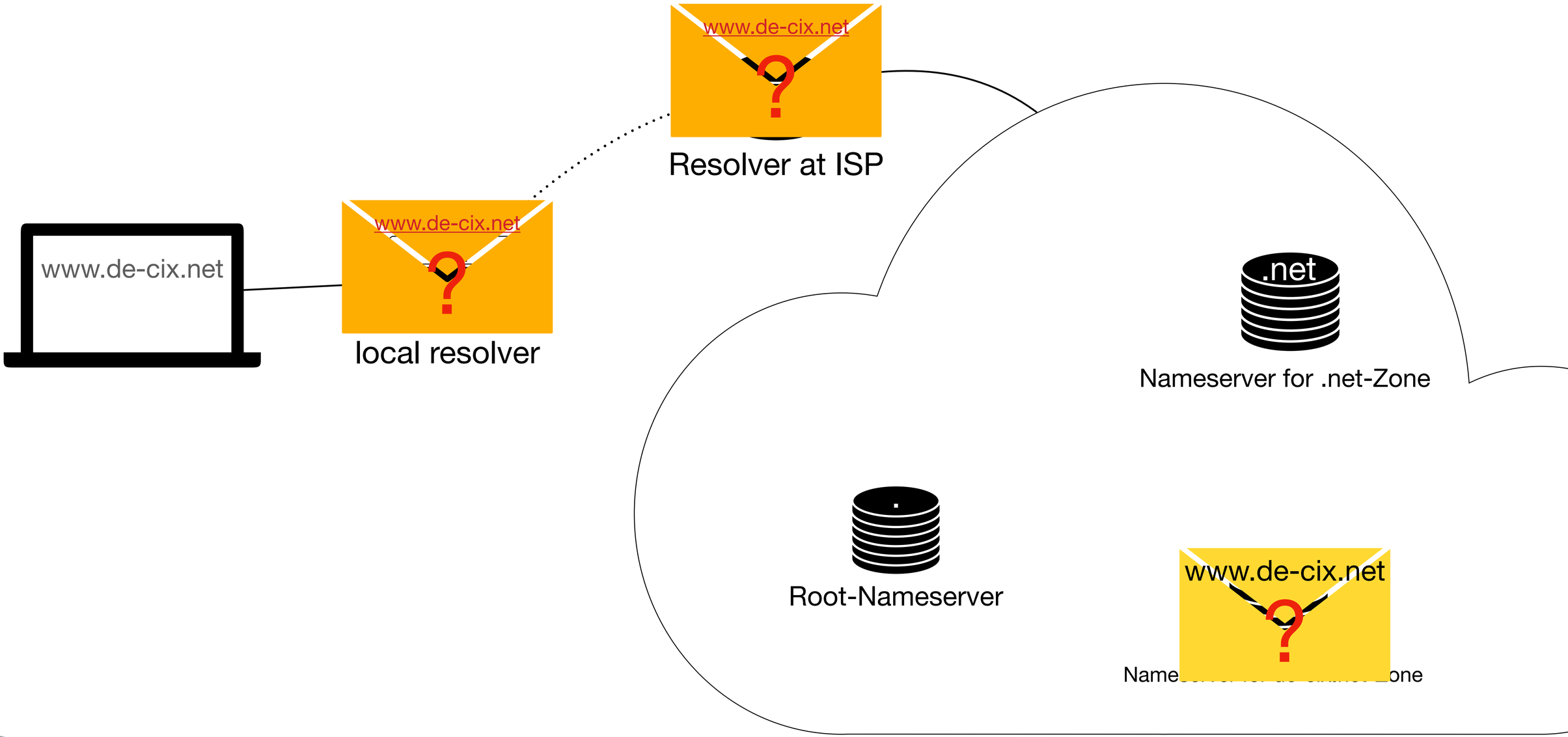
DNS - How does it work?



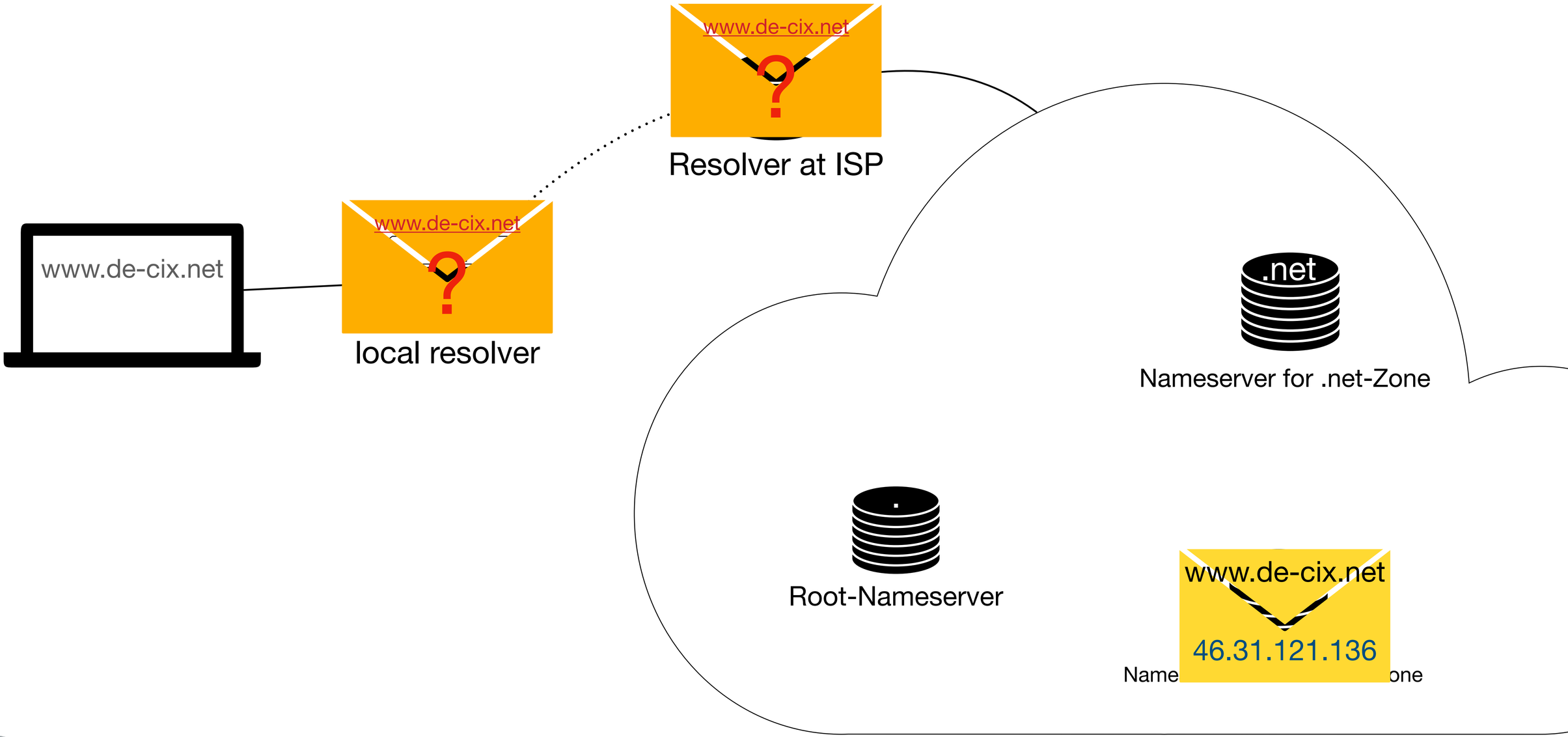
DNS - How does it work?



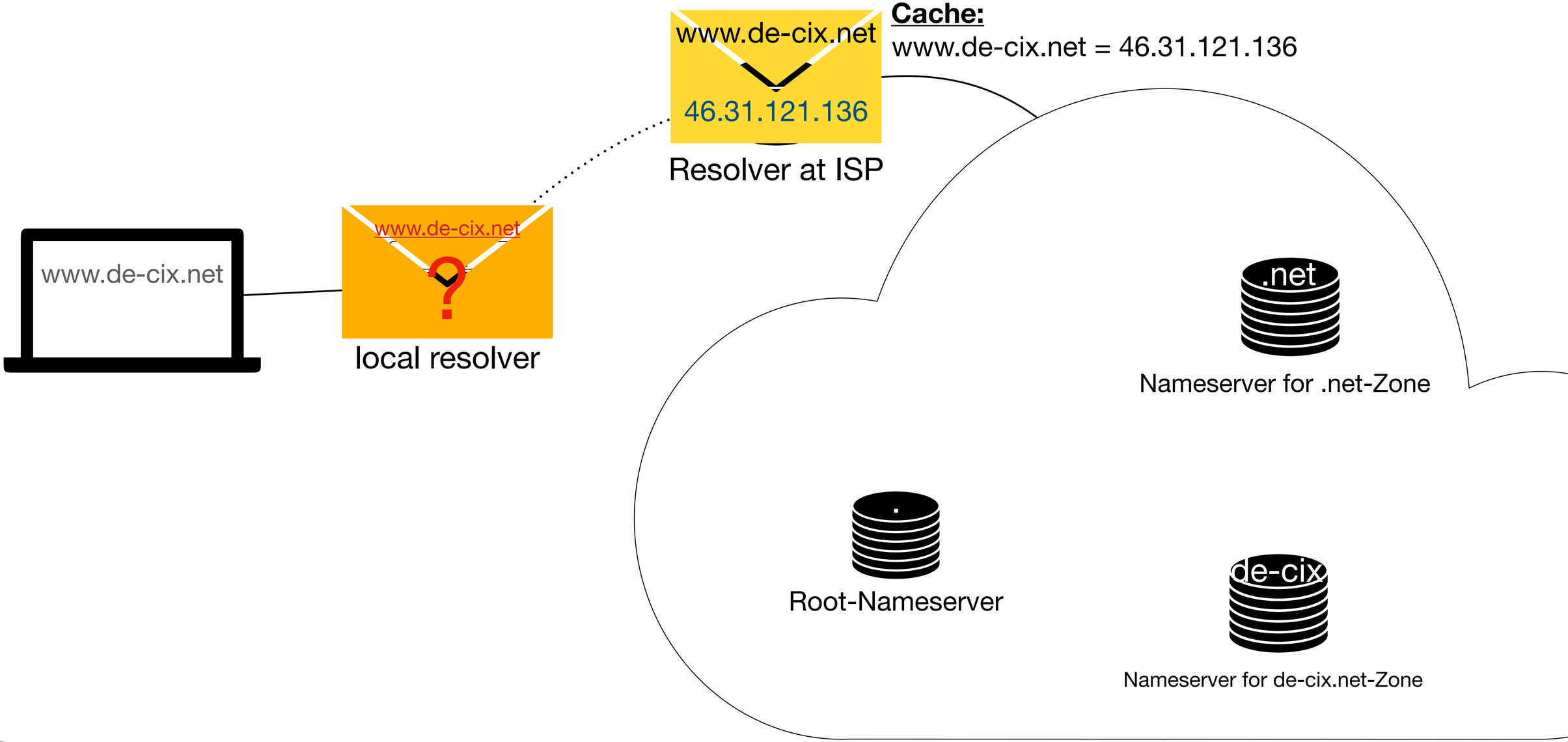
DNS - How does it work?



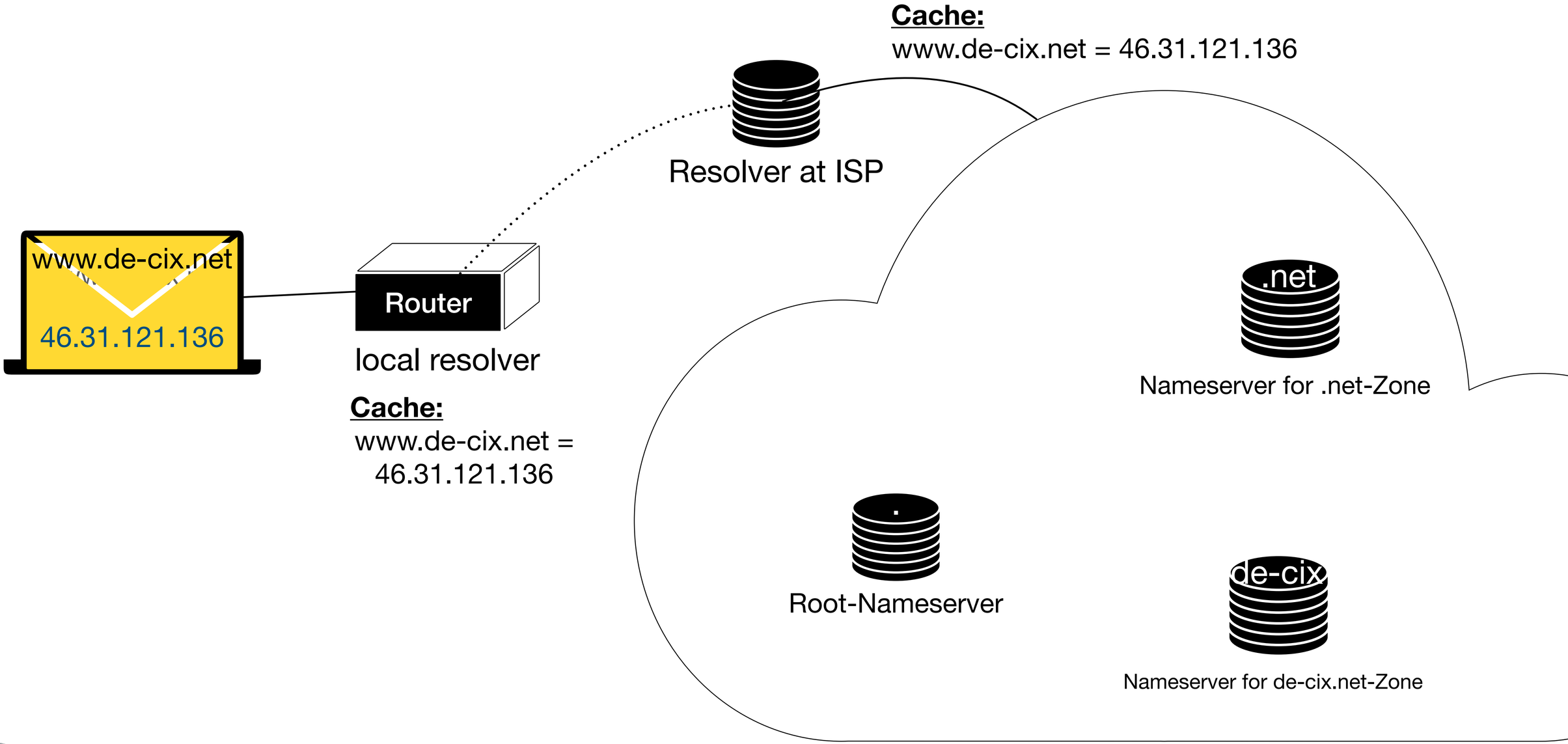
DNS - How does it work?



DNS - How does it work?



DNS - How does it work?



DNS - How does it work?

Two modes of operation - recursive or non-recursive

DE CIX

DNS - How does it work?

Two modes of operation - recursive or non-recursive

- Recursive query
  - "I want to know an IP address, just give me the result"
  - Used by your computer and also your local router

DE CIX

DNS - How does it work?

Two modes of operation - recursive or non-recursive

- Recursive query
  - "I want to know an IP address, just give me the result"
  - Used by your computer and also your local router
- Non-recursive query
  - "Where do I have to look next?"
  - Used by your providers resolver

DE CIX

DNS - How does it work?

Two modes of operation - recursive or non-recursive

- Recursive query
  - "I want to know an IP address, just give me the result"
  - Used by your computer and also your local router
- Non-recursive query
  - "Where do I have to look next?"
  - Used by your providers resolver
- DNS resolvers must be able to do both

DNS - How does it work?

DNS - How does it work?

2. Zones

DNS Zones

What are "Zones"?

DNS Zones

What are "Zones"?

- Units of authoritative information

DE CIX

DNS Zones

What are "Zones"?

- Units of authoritative information
  - Like the zones "example.com" or "de-cix.net"

DE CIX

DNS Zones

What are "Zones"?

- Units of authoritative information
  - Like the zones "example.com" or "de-cix.net"
  - Over which a domain server has complete authoritative information

DE CIX

DNS Zones

What are "Zones"?

- Units of authoritative information
  - Like the zones "example.com" or "de-cix.net"
  - Over which a domain server has complete authoritative information
- In reality often a text file with names and IP addresses in it

DE CIX

DNS Zones

What are "Zones"?

- Units of authoritative information
  - Like the zones "example.com" or "de-cix.net"
  - Over which a domain server has complete authoritative information
- In reality often a text file with names and IP addresses in it
  - Or a table in a database

DE CIX

DNS Zones

What are "Zones"?

- Units of authoritative information
  - Like the zones "example.com" or "de-cix.net"
  - Over which a domain server has complete authoritative information
- In reality often a text file with names and IP addresses in it
  - Or a table in a database
- Each zone has one *primary name server*, but each name server can serve multiple zones

DE CIX

DNS Zones

Example

DNS Zones

Example

- Zones start with a **SOA** record
- SOA means Start Of Authority
- The values configured here are very important so we will cover them all

```
example.com.  3600  SOA  dns.example.com.  hostmaster.example.com.  (
                           2021110200    ; serial YYYYMMDDnn
                           86400         ; refresh (   24 hours)
                           7200          ; retry   (    2 hours)
                           3600000       ; expire  (1000 hours)
                           172800 )      ; minimum (    2 days)
```

DE CIX

DNS Zones

Example

- Zones start with a **SOA** record
- SOA means Start Of Authority
- The values configured here are very important so we will cover them all

- First value is the name of the zone, here "example.com"

```
example.com.  3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Example

- Zones start with a **SOA** record
- SOA means Start Of Authority
- The values configured here are very important so we will cover them all

- First value is the name of the zone, here "example.com"
- The 3600 is the TimeToLive value for the SOA record itself (3600 seconds == one hour)
  - After this time the SOA record is refreshed (anywhere it is cached)

```
example.com.  3600  SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Primary name server and zone contact

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Primary name server and zone contact

- "dns.example.com" is the *primary name server* for this zone
  - There can be only one

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Primary name server and zone contact

- "dns.example.com" is the *primary name server* for this zone
  - There can be only one

- "hostmaster.example.com" is actually an email address
  - replace the first "." with "@"
- So it reads hostmaster@example.com
- This is the responsible contact for the zone

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Serial

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Serial

- "Serial" is an up-counting serial number of the zone file

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DNS Zones

Serial

- "Serial" is an up-counting serial number of the zone file
- If you make a change, you have to increase it, otherwise the software does not pick up the change

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Serial

- "Serial" is an up-counting serial number of the zone file
- If you make a change, you have to increase it, otherwise the software does not pick up the change
- Of course you can simply start with "1" and count upwards

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Serial

- "Serial" is an up-counting serial number of the zone file
- If you make a change, you have to increase it, otherwise the software does not pick up the change
- Of course you can simply start with "1" and count upwards
- But best practise is to encode the date of the last change

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                          2021110200   ; serial YYYYMMDDnn
                          86400        ; refresh (   24 hours)
                          7200         ; retry   (    2 hours)
                          3600000      ; expire  (1000 hours)
                          172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Refresh

- "Refresh" is how often *secondary name servers* will pull the primary name server for changes
  - To indicate a change the serial numbers are compared
  - 24h are a reasonable default
  - Modern DNS software uses a "notify" mechanism

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Refresh

- "Refresh" is how often *secondary name servers* will pull the primary name server for changes
  - To indicate a change the serial numbers are compared
  - 24h are a reasonable default
  - Modern DNS software uses a "notify" mechanism

It is a real good idea to place your 2ndary name servers into different networks.

So in case of a massive outage your domain name is still resolvable.

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                            2021110200   ; serial YYYYMMDDnn
                            86400        ; refresh (   24 hours)
                            7200         ; retry   (    2 hours)
                            3600000      ; expire  (1000 hours)
                            172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Retry

- If a 2ndary name server cannot contact the primary for refresh, it waits "**retry**" time until it tries again

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Expire

- How long a secondary name server will treat its copy of the zone valid if it cannot contact the primary
  - Default of 1000 hours was chosen because it is a nice round value
  - 2-4 weeks are ok here

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                            2021110200   ; serial YYYYMMDDnn
                            86400        ; refresh (   24 hours)
                            7200         ; retry   (    2 hours)
                            3600000      ; expire  (1000 hours)
                            172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Minimum

- This is the default time-to-live value for entries without an explicit TTL
- How long data is cached in other name servers
  - Value depends on how often zone data is changed
  - Two days is a good default for a stable zone
  - Lower if you change entries a lot or plan a change in the near future.

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                            2021110200   ; serial YYYYMMDDnn
                            86400        ; refresh (   24 hours)
                            7200         ; retry   (    2 hours)
                            3600000      ; expire  (1000 hours)
                            172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

Minimum

- This is the default time-to-live value for entries without an explicit TTL
- How long data is cached in other name servers
  - Value depends on how often zone data is changed
  - Two days is a good default for a stable zone
  - Lower if you change entries a lot or plan a change in the near future.
  - AS196610 uses 30 minutes here

```
example.com. 3600 SOA dns.example.com. hostmaster.example.com. (
                           2021110200   ; serial YYYYMMDDnn
                           86400        ; refresh (   24 hours)
                           7200         ; retry   (    2 hours)
                           3600000      ; expire  (1000 hours)
                           172800 )     ; minimum (    2 days)
```

DE CIX

DNS Zones

More values

- "**NS**" - entries list all name servers for a zone

```
IN NS    ns.de-cix.net.
IN NS    dns.de-cix.net.
```

DE CIX

DNS Zones

More values

- "**NS**" - entries list all name servers for a zone
- We already know the primary name server from the SOA entry

```
IN NS    ns.de-cix.net.
IN NS    dns.de-cix.net.
```

DE CIX

DNS Zones

More values

- "**NS**" - entries list all name servers for a zone
- We already know the primary name server from the SOA entry
- Any zone must have at least two name servers

```
IN NS    ns.de-cix.net.
IN NS    dns.de-cix.net.
```

DE CIX

DNS Zones

More values

- "**NS**" - entries list all name servers for a zone
- We already know the primary name server from the SOA entry
- Any zone must have at least two name servers
- Non-primary name servers are called *secondary* name servers

```
IN NS    ns.de-cix.net.
IN NS    dns.de-cix.net.
```

DE CIX

DNS Zones

More values

- "**NS**" - entries list all name servers for a zone
- We already know the primary name server from the SOA entry
- Any zone must have at least two name servers
- Non-primary name servers are called *secondary* name servers
- They must also be listed in the parent zone (in this case ".net")

```
IN NS    ns.de-cix.net.
IN NS    dns.de-cix.net.
```

DE CIX

DNS Zones

Finally - IP addresses

```
academyserver01          IN A            46.31.124.66
                         IN AAAA         2a02:c50:6209:704::2

academyserver02          IN A            91.214.253.1
                         IN AAAA         2a02:c50:db8:3::2

academyserver04          IN A            91.214.253.3
                         IN AAAA         2a02:c50:db8:3::3
```

DE CIX

DNS Zones

Finally - IP addresses

- Reminder - "IN" stands for Internet
- "A" records are for IPv4 addresses
- "AAAA" records are for IPv6 addresses

```
academyserver01          IN A            46.31.124.66
                         IN AAAA         2a02:c50:6209:704::2

academyserver02          IN A            91.214.253.1
                         IN AAAA         2a02:c50:db8:3::2

academyserver04          IN A            91.214.253.3
                         IN AAAA         2a02:c50:db8:3::3
```

DE CIX

DNS Zones - What else?

Email delivery

- For mail delivery information, "MX" (mail exchanger) records are used
  - Here: Email for @example.com is to be delivered to two servers
  - 10 and 20 are priority values (lower means try first)

```
example.com.        IN      MX      10 mailgw01.example.com.
                    IN      MX      20 mailgw02.example.com.
```

DE CIX

DNS Zones - What else?

Aliases (or: Canonical Names)

- "CNAME" records can be used to give a host a nickname
- Here **paul.example.com** has the nickname **peter**
- No other entries for peter are allowed if there is a CNAME entry

```
peter        IN      CNAME      paul.example.com.
```

DE CIX

DNS - names to addresses

Using A and AAAA records

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name
- You query a resolver and ask for an A or AAAA record

DE CIX

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name
- You query a resolver and ask for an A or AAAA record
- The resolver either
  - has the information in its cache

DE CIX

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name
- You query a resolver and ask for an A or AAAA record
- The resolver either
  - has the information in its cache
  - or recursively queries the next resolver

DE CIX

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name
- You query a resolver and ask for an A or AAAA record
- The resolver either
  - has the information in its cache
  - or recursively queries the next resolver
  - or gathers the information itself

DE CIX

DNS - names to addresses

Using A and AAAA records

- So now we know how to get an IP address for a name
- You query a resolver and ask for an A or AAAA record
- The resolver either
  - has the information in its cache
  - or recursively queries the next resolver
  - or gathers the information itself
- But what about the other way?
  - How to get a name to an IP address?

How to get a name to an IP address?

DNS - Reverse mapping

Getting a name to an IP address

DNS - Reverse mapping

Getting a name to an IP address

- There is a special domain for this
  - IN-ADDR.ARPA for IPv4
  - IP6.ARPA for IPv6

DE CIX

DNS - Reverse mapping

Getting a name to an IP address

- There is a special domain for this
  - IN-ADDR.ARPA for IPv4
  - IP6.ARPA for IPv6
- And a special record type
  - PTR

DE CIX

DNS - Reverse mapping

Getting a name to an IP address

- There is a special domain for this
  - IN-ADDR.ARPA for IPv4
  - IP6.ARPA for IPv6
- And a special record type
  - PTR
- But how does a reverse query work?

DNS - Reverse mapping

How does it work?

DNS - Reverse mapping

How does it work?

- From a user standpoint its quite easy:

DE CIX

DNS - Reverse mapping

How does it work?

- From a user standpoint its quite easy:
  - `dig -x 46.31.121.136` or
    `dig -x 2a02:c50:6209:702::8`

DE CIX

DNS - Reverse mapping

How does it work?

- From a user standpoint its quite easy:
  - `dig -x 46.31.121.136` or
    `dig -x 2a02:c50:6209:702::8`
  - Some tool will do the work for you

DE CIX

DNS - Reverse mapping

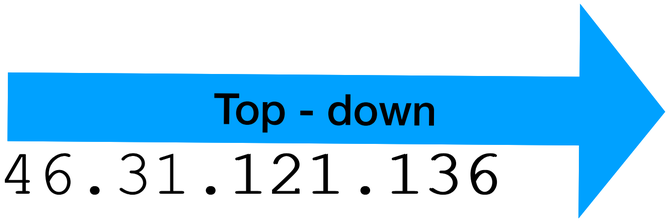
How does it work?

- From a user standpoint its quite easy:
  - `dig -x 46.31.121.136` or
    `dig -x 2a02:c50:6209:702::8`
  - Some tool will do the work for you
  - But what is actually queried?

DE CIX

DNS - Reverse mapping

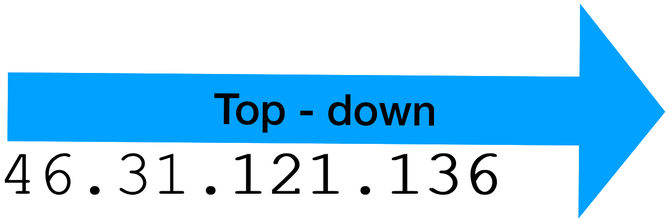
How does it work?




```
dig -x 46.31.121.136
```

DNS - Reverse mapping

How does it work?



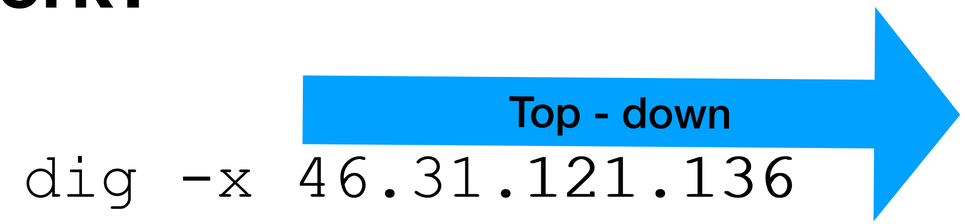

```
dig -x 46.31.121.136
```

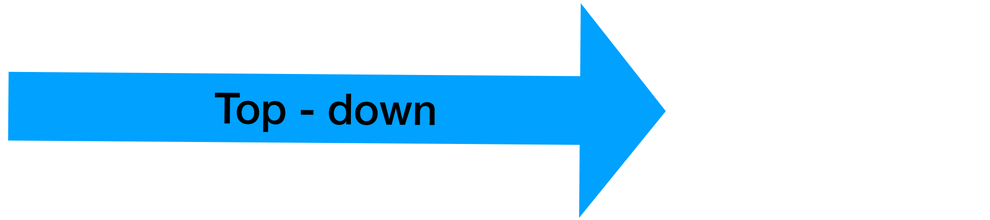
```
.in-addr.arpa.   IN    PTR
```

DE CIX

DNS - Reverse mapping

How does it work?

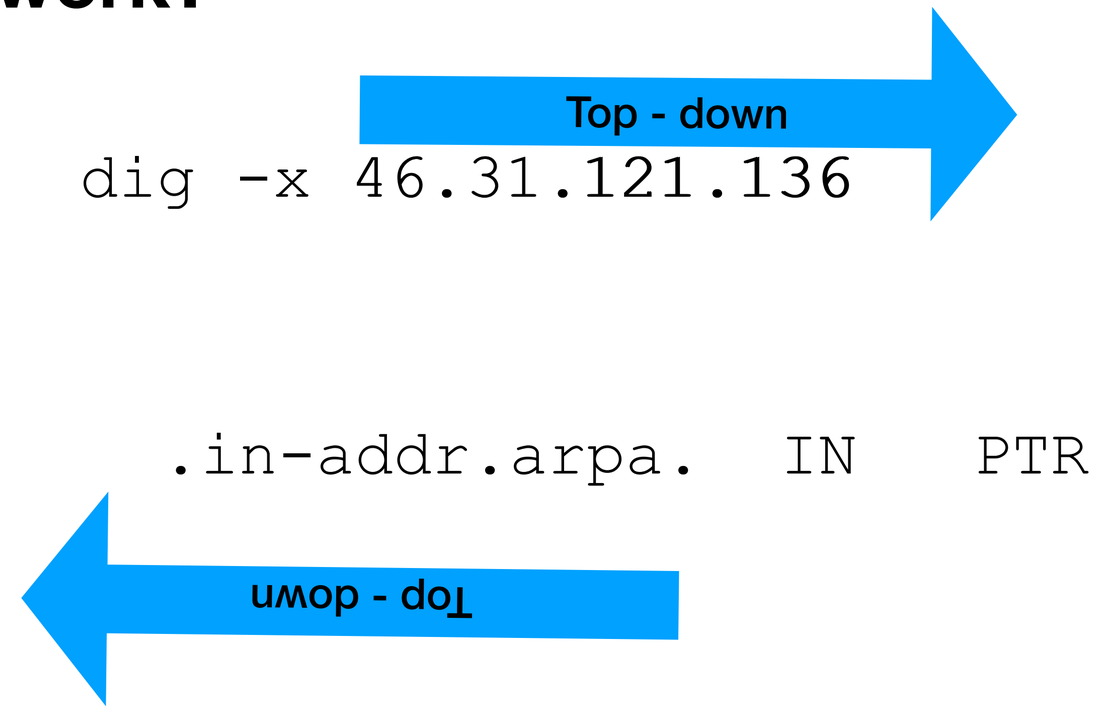




DE CIX

DNS - Reverse mapping

How does it work?



DNS - Reverse mapping

How does it work?


```
dig -x 46.31.121.136
```

```
136.121.31.46.in-addr.arpa.   IN    PTR
```

DNS - Reverse mapping

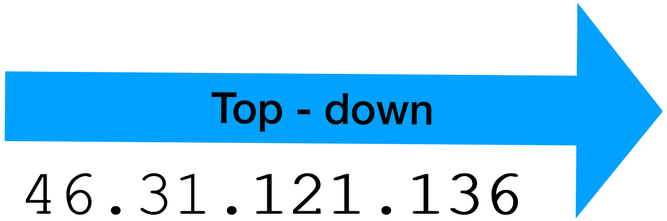
How does it work?


```
dig -x 46.31.121.136
```

```
136.121.31.46.in-addr.arpa.   IN    PTR   myserver.example.com
```

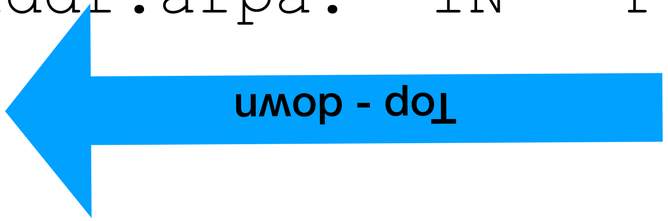
DNS - Reverse mapping

How does it work?



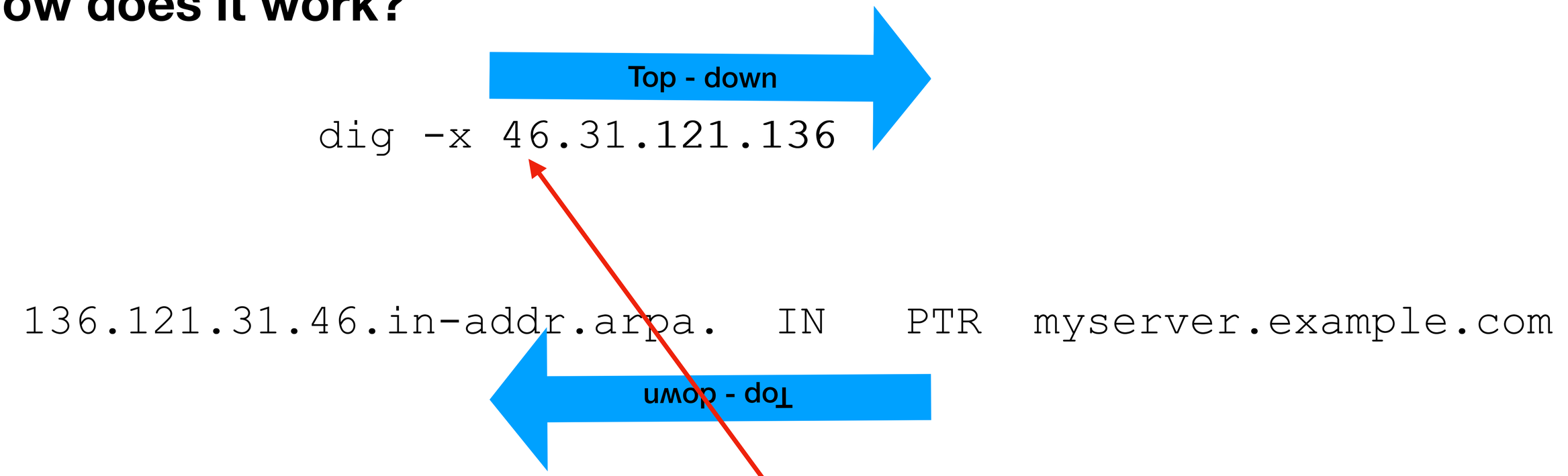

```
dig -x 46.31.121.136
```

```
136.121.31.46.in-addr.arpa.   IN    PTR   myserver.example.com
```



DNS - Reverse mapping

How does it work?

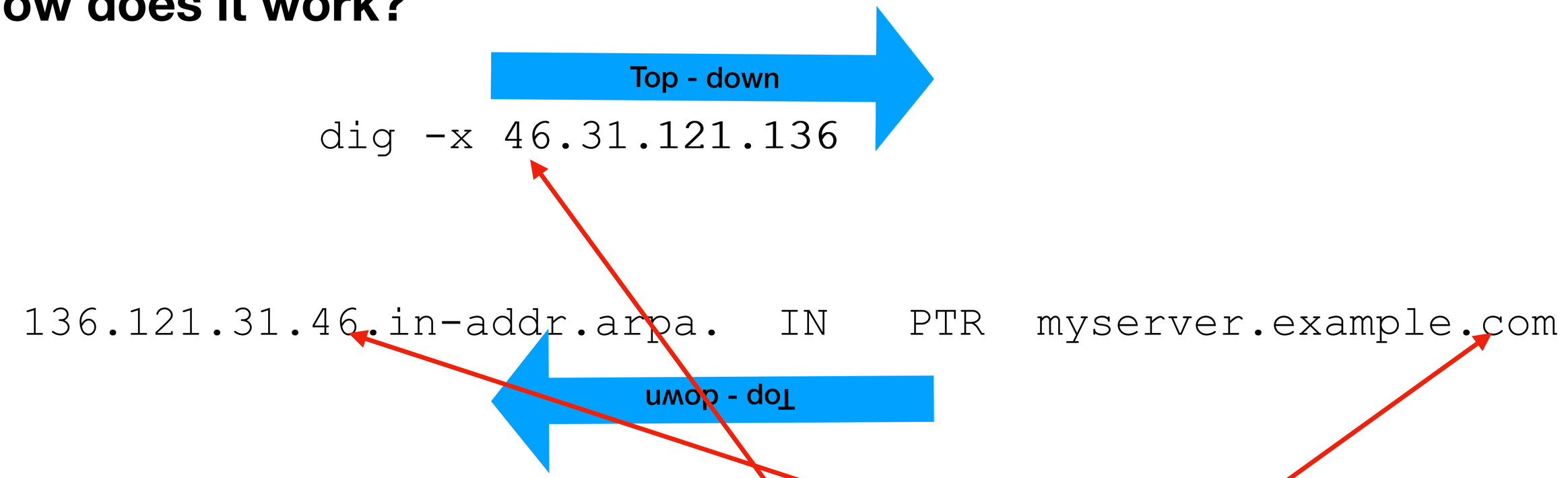


- Hierarchy of IP addresses: most significant part is left

DE CIX

DNS - Reverse mapping

How does it work?

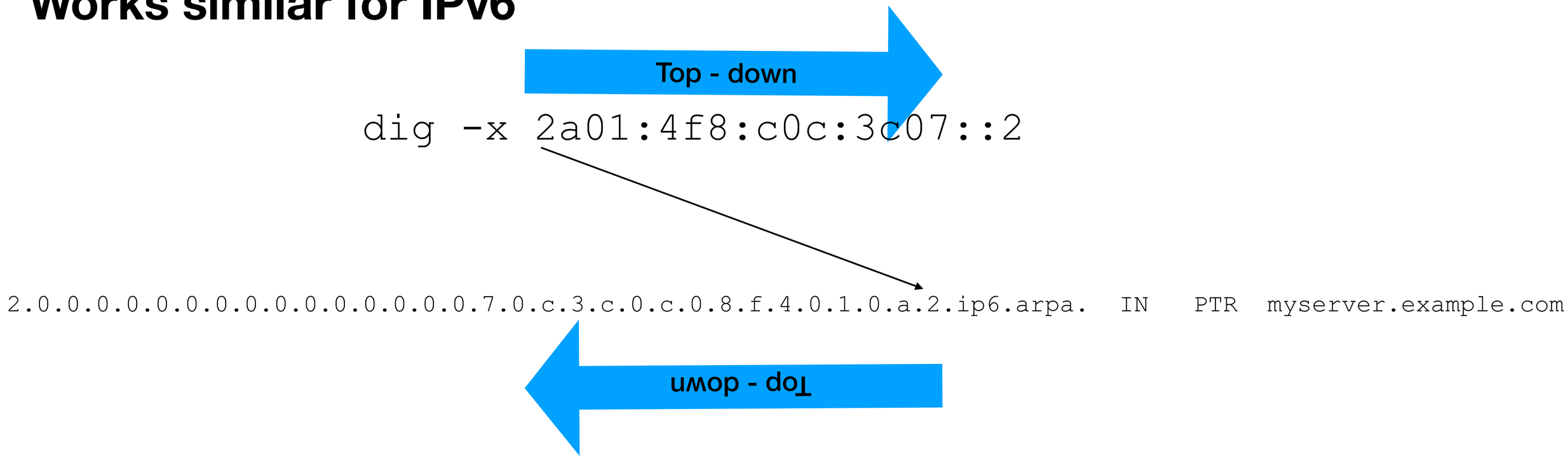


- Hierarchy of IP addresses: most significant part is left
- Hierarchy of Domain Names: most significant part is right (like .net, .com, ...)

DE CIX

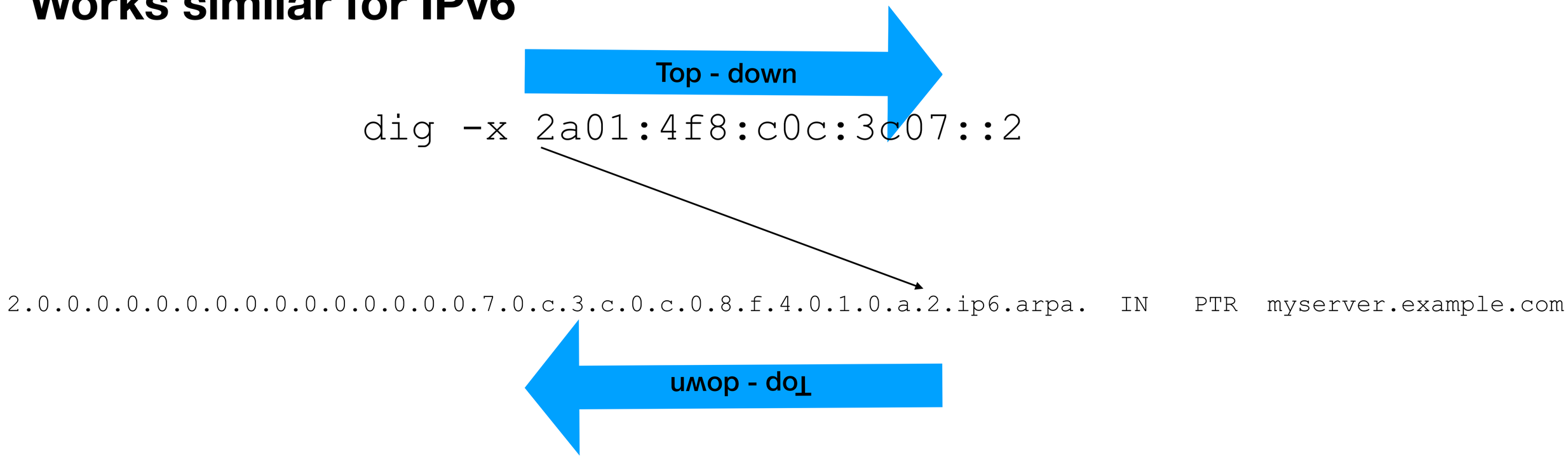
DNS - Reverse mapping

Works similar for IPv6



DNS - Reverse mapping

Works similar for IPv6



- For IPv6, "ip6.arpa" is used as domain
- Each possible digit has its own hierarchy level

DE CIX

DNS Reverse and Forward

Best Practise

- Have a reverse entry for all your IP addresses in use
  - Automate it!
- IF you have a "forward" entry, match the reverse
  - Not all reverse entries need to have a forward entry, especially for network infrastructure

DE CIX

DNS Today

DNS Today

Technical

DE CIX

DNS Today

Technical

- Every ISP offers DNS resolvers for its customers

DE CIX

DNS Today

Technical

- Every ISP offers DNS resolvers for its customers
  - For home-users your router picks them up automatically

DE CIX

DNS Today

Technical

- Every ISP offers DNS resolvers for its customers
  - For home-users your router picks them up automatically
- But also a number of "Public Resolvers" are available from various providers

DE CIX

DNS Today

Technical

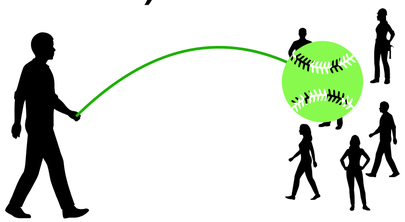
- Every ISP offers DNS resolvers for its customers
  - For home-users your router picks them up automatically
- But also a number of "Public Resolvers" are available from various providers
  - Like Google (at 8.8.8.8), Cloudflare (at 1.1.1.1), Quad9 (at 9.9.9.9)

DE CIX

DNS Today

Technical

- Every ISP offers DNS resolvers for its customers
  - For home-users your router picks them up automatically
- But also a number of "Public Resolvers" are available from various providers
  - Like Google (at 8.8.8.8), Cloudflare (at 1.1.1.1), Quad9 (at 9.9.9.9)
  - All these (and more) use an *Anycast* infrastructure

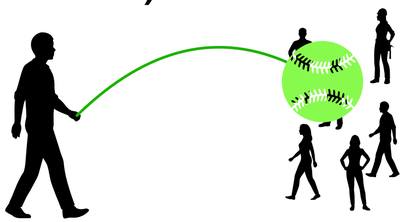


DE CIX

DNS Today

Technical

- Every ISP offers DNS resolvers for its customers
  - For home-users your router picks them up automatically
- But also a number of "Public Resolvers" are available from various providers
  - Like Google (at 8.8.8.8), Cloudflare (at 1.1.1.1), Quad9 (at 9.9.9.9)
  - All these (and more) use an *Anycast* infrastructure
- Recently, DNS over HTTPS (DoH) was defined in RFC8484



DE CIX

DNS Today

Technical

DE CIX

DNS Today

Technical

- DNSSec is a recent initiative to make DNS more secure

DE CIX

DNS Today

Technical

- DNSSec is a recent initiative to make DNS more secure
  - By signing zones

DE CIX

DNS Today

Technical

- DNSSec is a recent initiative to make DNS more secure
  - By signing zones
    - A "chain of trust" is used to authenticate public keys
    - With the DNS root zone as trust anchor

DE CIX

DNS Today

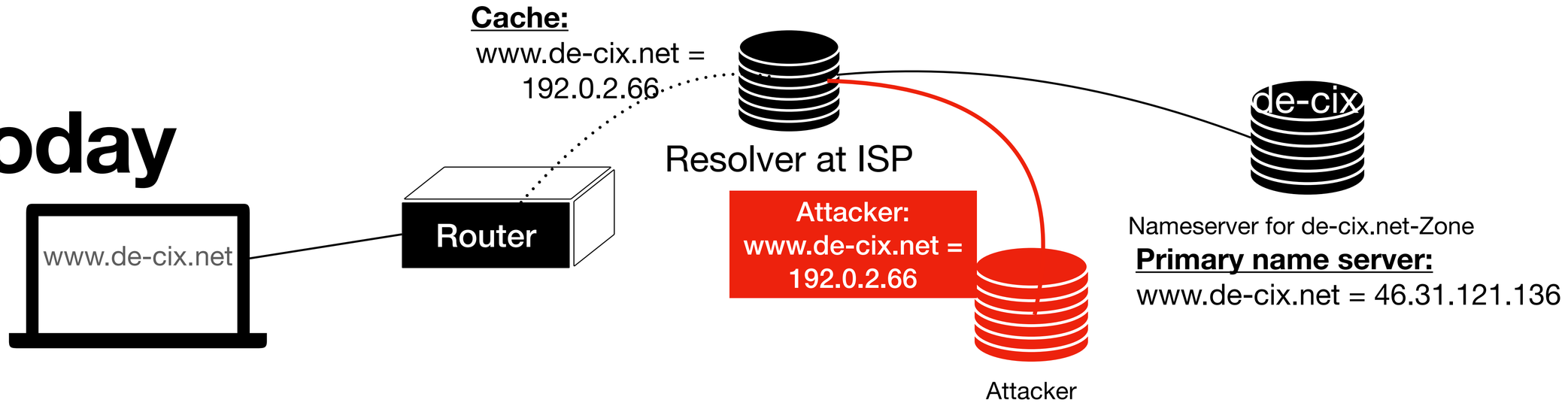
Technical

- DNSSec is a recent initiative to make DNS more secure
  - By signing zones
    - A "chain of trust" is used to authenticate public keys
    - With the DNS root zone as trust anchor
  - Resolvers can use this to check the correctness of (cached) answers

DE CIX

DNS Today

Technical



- DNSSec is a recent initiative to make DNS more secure
  - By signing zones
    - A "chain of trust" is used to authenticate public keys
    - With the DNS root zone as trust anchor
  - Resolvers can use this to check the correctness of (cached) answers
  - This prevents an attack against DNS called "cache poisoning"

DE CIX

DNS Today

Non-Technical

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship

DE CIX

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship
  - To suppress unwanted content

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship
  - To suppress unwanted content
  - To control who can see what

DE CIX

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship
  - To suppress unwanted content
  - To control who can see what

- Remember - DNS only translates a (domain) name to a number (IP address)

DE CIX

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship
  - To suppress unwanted content
  - To control who can see what

- Remember - DNS only translates a (domain) name to a number (IP address)
  - You can do that translation yourself locally if you know the address

DE CIX

DNS Today

Non-Technical

- As DNS is so central to using the Internet, it is often used as an instrument of censorship
  - To suppress unwanted content
  - To control who can see what

- Remember - DNS only translates a (domain) name to a number (IP address)
  - You can do that translation yourself locally if you know the address
  - /etc/hosts still works - or you can run your own resolver!

DE CIX

Conclusion

Conclusion

DNS

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*

DE CIX

Conclusion

DNS

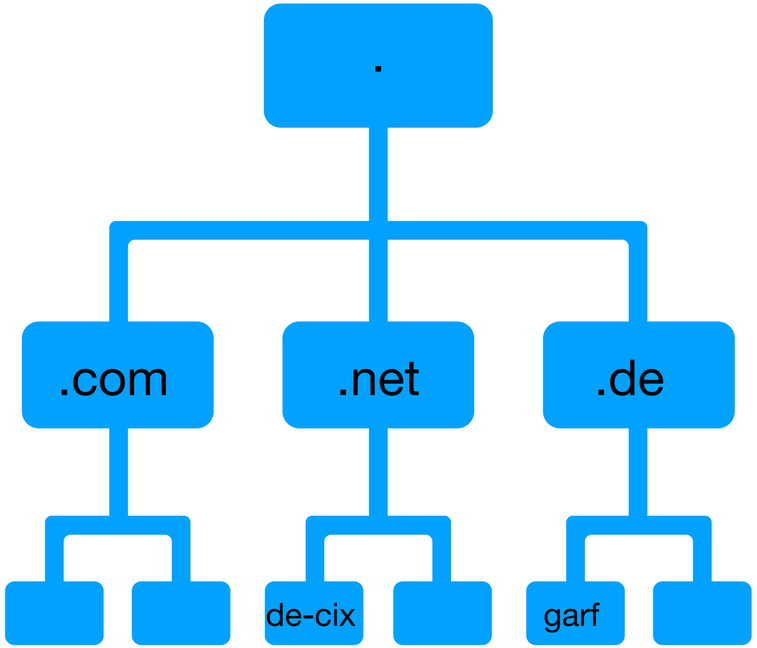
- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8

DE CIX

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure

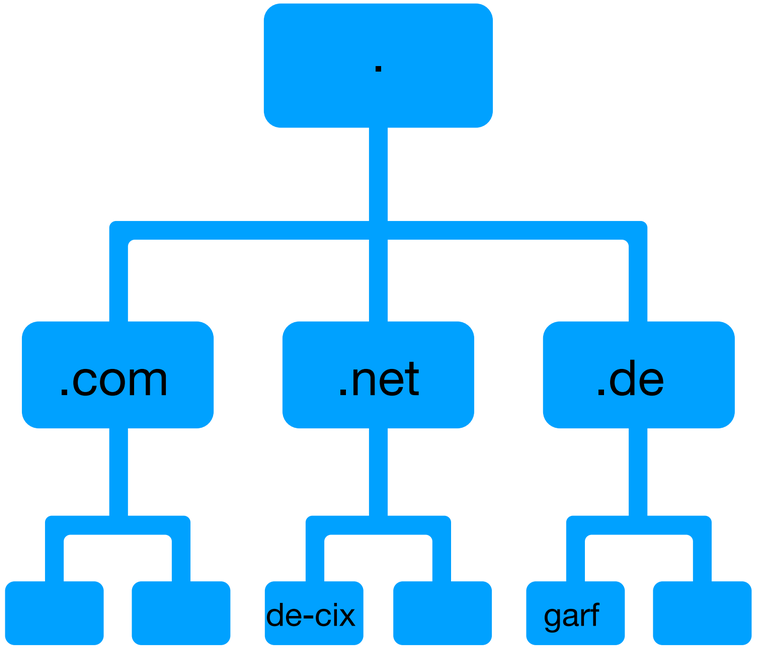


DE CIX

Conclusion

DNS

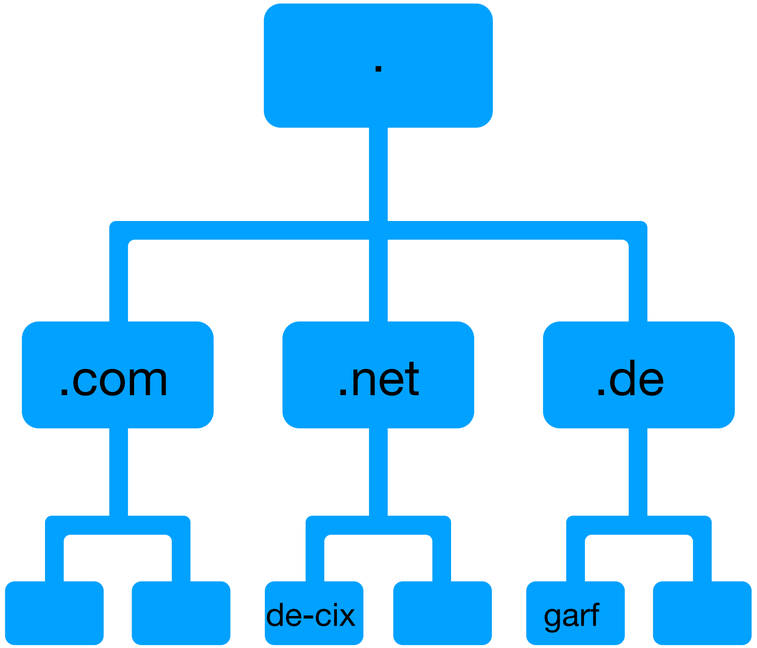
- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net



Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net
  - On 2nd level de-cix - so it is de-cix.net

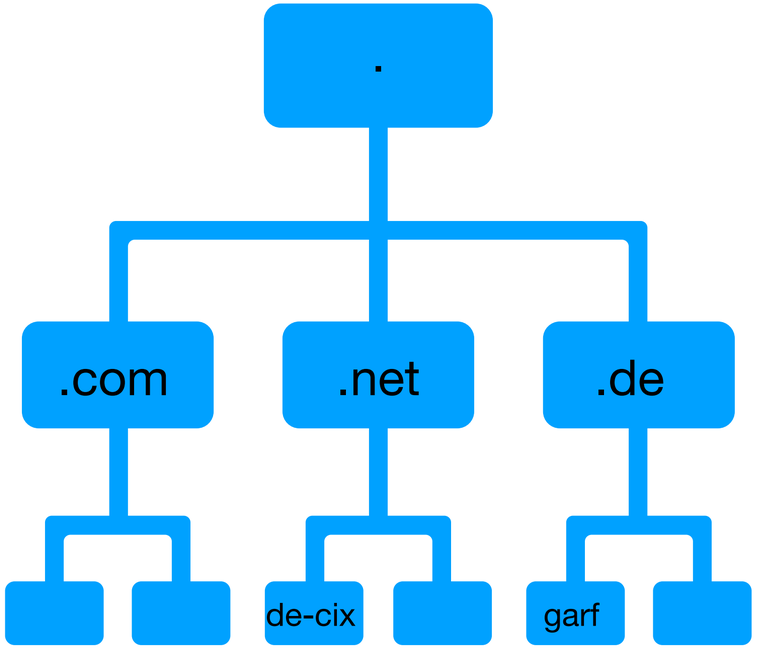


DE CIX

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net
  - On 2nd level de-cix - so it is de-cix.net
  - On the 3rd level service names like www - so it is www.de-cix.net

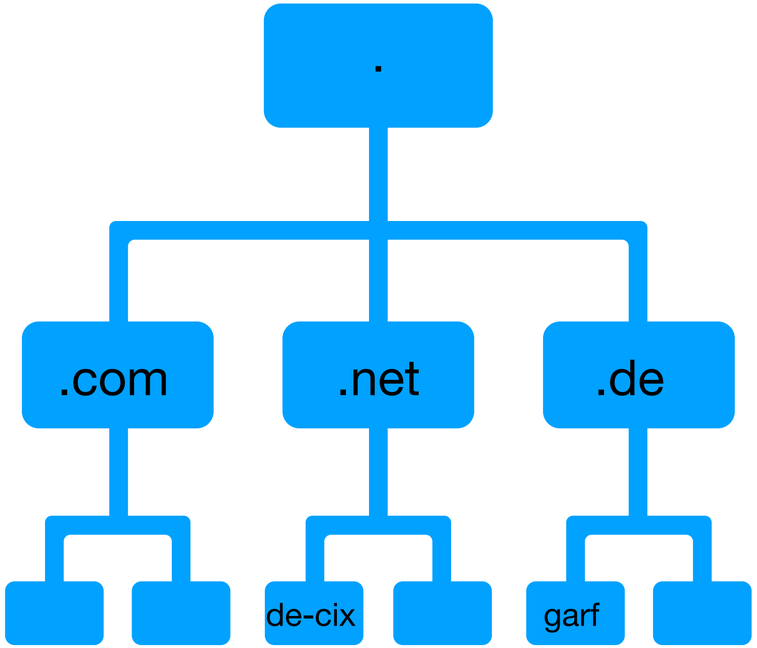


DE CIX

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net
  - On 2nd level de-cix - so it is de-cix.net
  - On the 3rd level service names like www - so it is www.de-cix.net
- DNS can also reverse-map IP addresses back to names

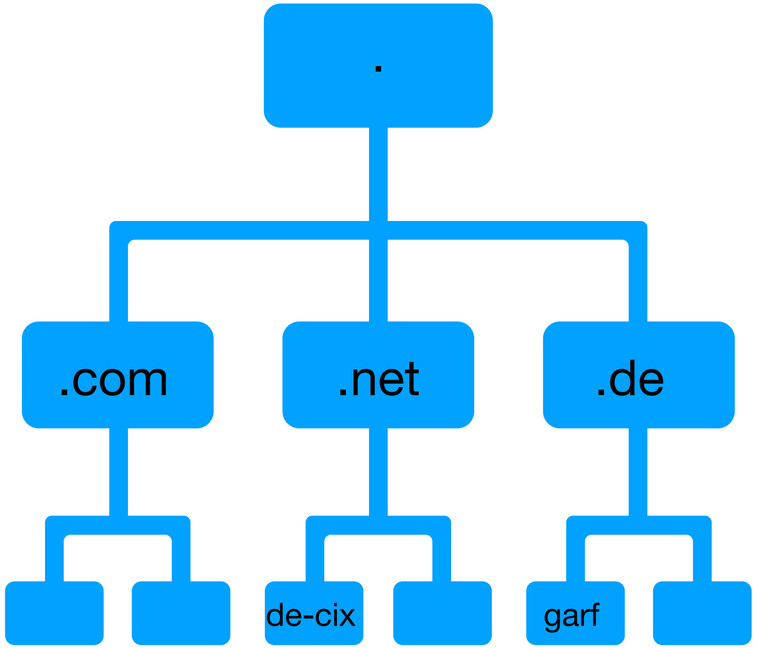


DE CIX

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net
  - On 2nd level de-cix - so it is de-cix.net
  - On the 3rd level service names like www - so it is www.de-cix.net
- DNS can also reverse-map IP addresses back to names
  - like mapping 46.31.121.136 to webprod01.de-cix.net

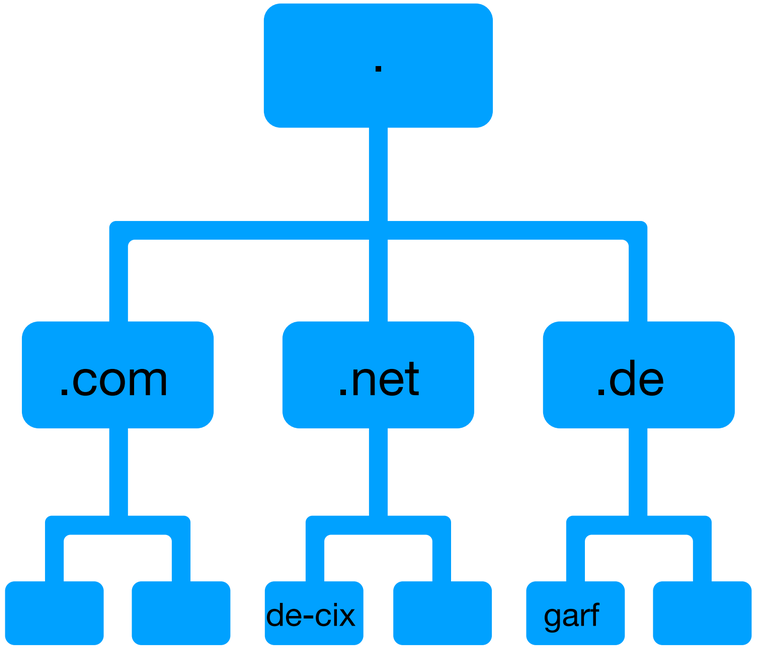


DE CIX

Conclusion

DNS

- DNS - Domain Name System - maps *names* to *numbers*
  - Like www.de-cix.net to 2a02:c50:6209:702::8
- Domains are hierarchical and have a tree-like structure
  - On the top-level (below the root) domains like .net
  - On 2nd level de-cix - so it is de-cix.net
  - On the 3rd level service names like www - so it is www.de-cix.net
- DNS can also reverse-map IP addresses back to names
  - like mapping 46.31.121.136 to webprod01.de-cix.net
- DNS is a complex system which was extended a lot over time



Thank you!

academy@de-cix.net

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

DE-CIX Management GmbH | Lindleystr. 12 | 60314 Frankfurt | Germany
Phone + 49 69 1730 902 0 | sales@de-cix.net | www.de-cix.net

Links and further reading

Links and further reading

- Internet protocol - https://en.wikipedia.org/wiki/Internet_Protocol
- Protocol stack - https://en.wikipedia.org/wiki/Protocol_stack
  - Transport Layer: https://en.wikipedia.org/wiki/Transport_layer
  - Datagram: https://en.wikipedia.org/wiki/Datagram
- IP Network Model: https://en.wikipedia.org/wiki/Internet_protocol_suite
- IPv4
  - IPv4 - https://en.wikipedia.org/wiki/IPv4
- IPv6
  - IPv6 itself - https://en.wikipedia.org/wiki/IPv6
  - IPv6 header - https://en.wikipedia.org/wiki/IPv6_packet
- History of Internet and IP
  - Internet Hall of Fame - https://internethalloffame.org
  - Defense Advanced Research Projects Agency (DARPA) - https://www.darpa.mil
  - ARPANET - https://www.darpa.mil/about-us/timeline/arpanet
  - The "Protocol Wars" - https://en.wikipedia.org/wiki/Protocol_Wars

Links and further reading

- Recommendations for DNS SOA values: https://www.ripe.net/publications/docs/ripe-203
- RFCs - there are too many RFCs about DNS to list them all, here are a few highlights:
  - RFC1034: Domain Names - Concepts and Facilities
  - RFC1035: Domain Names - Implementation And Specification
  - a good overview of all DNS RFCs can be found here: https://rfcs.io/dns

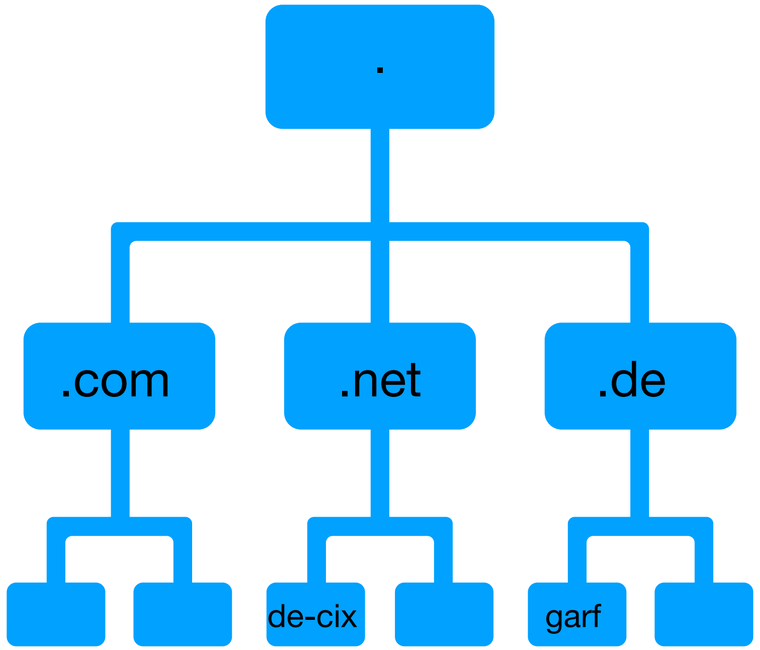
Popular name servers and resolvers

- Bind - old, but well maintained and with all features
- Unbound - fast and lean
- PowerDNS - for large installations

Popular public resolvers

This list may be always out of date...

- Overview: https://en.wikipedia.org/wiki/Public_recursive_name_server
- Cloudflare - 1.1.1.1: https://1.1.1.1
- Google - 8.8.8.8:
  https://developers.google.com/speed/public-dns/docs/using
- Quad9 - 9.9.9.9: https://www.quad9.net



DE CIX


Header
Questions
Answer Resource Records
Authority Resource Records
Additional Resource Records

DE CIX

Byte	0	1	2	3
0	ID		Flags	
4	QDCOUNT (Number of questions)		ANCOUNT (Number of answers)	
8	NSCOUNT (Number of authority records)		ARCOUNT (Number of additional records)	

Byte	0	1	2	3
0	ID = 0x8a79		Flags: R0 0000 0 Recurse=1 0 AD=1 0	
2	QDCOUNT (Number of questions): 1		ANCOUNT (Number of answers): 0	
4	NSCOUNT (Number of authority records): 0		ARCOUNT (Number of additional records): 1	

Byte	0	1	2	3
0	Length	QNAME		
...	...			
	...			0
	QTYPE		QCLASS	

Byte	0	1	2	3
0	3	w	w	w
4	6	d	e	-
8	c	i	x	3
12	n	e	t	0
16	TYPE: A (address)		CLASS: IN (Internet)	

Query

Byte	0	1	2	3
0	0	Type: 41 (OPT)		UDP Size
...	4096	0	Version 0	0
	0	0	0	

DE CIX

Query

Answer

Byte	0	1	2	3
0	ID		Flags	
4	QDCOUNT (Number of questions)		ANCOUNT (Number of answers)	
8	NSCOUNT (Number of authority records)		ARCOUNT (Number of additional records)	

Byte	0	1	2	3
0	Length	QNAME		
...	...			
	...			0
	QTYPE		QCLASS	

Byte	0	1	2	3
0	Length	Name		
	...			
	TYPE		CLASS	
	TTL			
	RDLength		RDATA	
...				

Byte	0	1	2	3
0	ID = 0x8a79		Flags: R1 0000 0 Recurse=1 1 AD=1 0	
4	QDCOUNT (Number of questions): 1		ANCOUNT (Number of answers): 1	
8	NSCOUNT (Number of authority records): 0		ARCOUNT (Number of additional records): 1	

Byte	0	1	2	3
0c	3	w	w	w
10	6	d	e	-
14	c	i	x	3
18	n	e	t	0
1c	TYPE: A (address)		CLASS: IN (Internet)	

Byte	0	1	2	3
20	0xc0	'0x0c	TYPE: A (address)	
24	CLASS: IN (Internet)		TTL	
28	TTL (3246 seconds)		RDLength = 4	
2c	46	31	121	136

Answer

Byte	0	1	2	3
0	0	Type: 41 (OPT)		UDP Size
...	4096	0	Version 0	0
	0	0	0	

DNS - How does it work?