RIPE NCC

RIPE NETWORK COORDINATION CENTRE

RIPE Database

Training Course

April 2025

RIPE NCC Learning & Development

RIPE NCC Training Material

Please find your training material at the following link

https://www.ripe.net/training-material

09:00 - 09:30	**Coffee, Tea**
11:00 - 11:15	**Break**
13:00 - 14:00	**Lunch**
15:30 - 15:45	**Break**
17:30	**End**

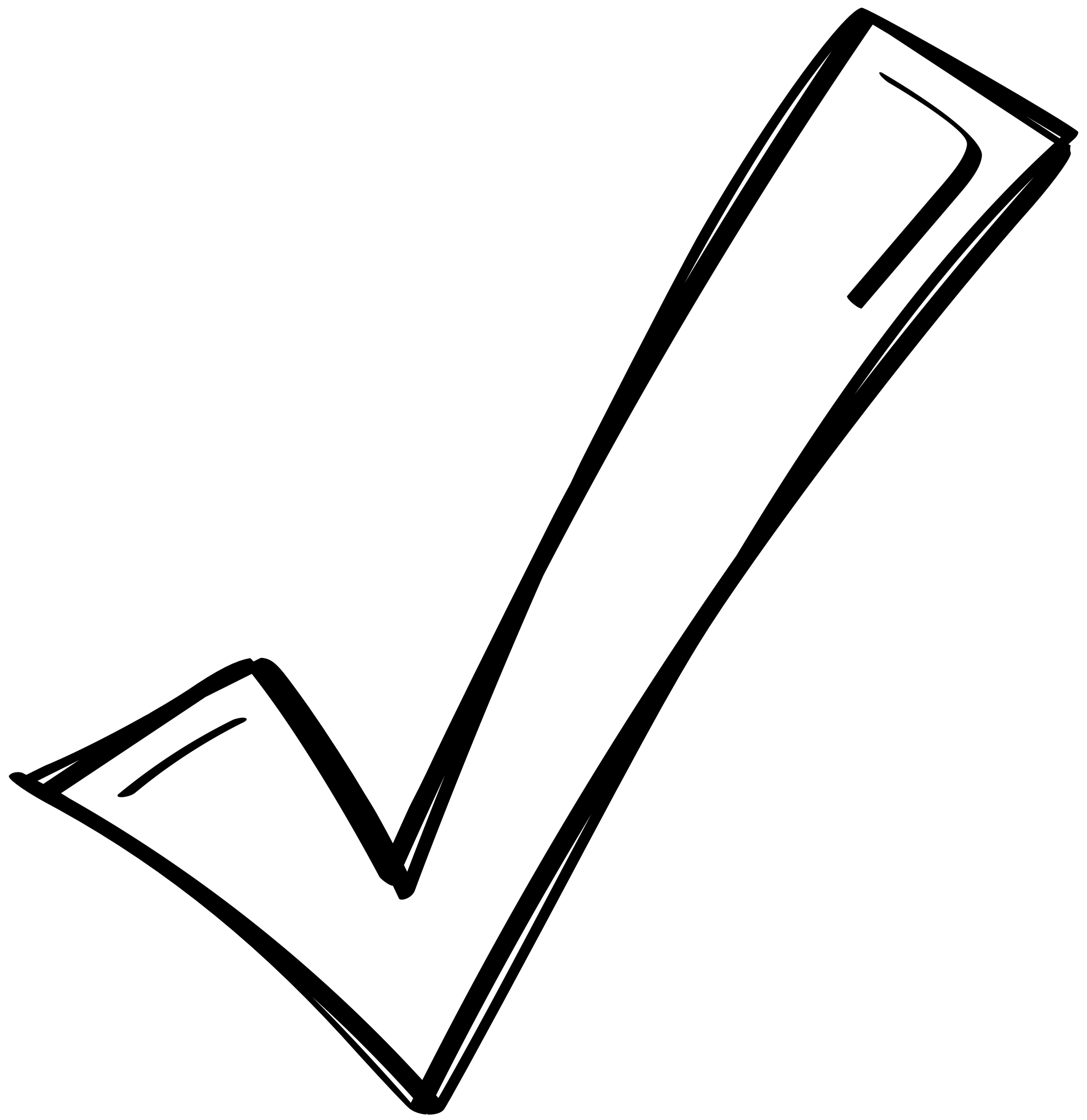
Introductions

- Name
- Experience with:
  - Operating an LIR
  - The RIPE Database
- Goals for today

Hello!

Overview

- What is the RIPE Database?
- How does it work?
- How to update it?
- Delegating address space to others
- RIPE Routing Registry
- Reverse DNS
- More RIPE Database
- Play Time!
- The RESTful API



Getting ready!

- Get your laptop up and running

- Make sure you have an Internet connection
  - **and a RIPE NCC Access account:** https://access.ripe.net

- Associate your Access account with your designated maintainer
  - https://academy.ripe.net/ext/testdb/add_sso.php

- Go to the TEST Database: https://apps-test.db.ripe.net
  - Open several tabs in the browser, if you want

Make sure you are in the **TEST** Database!

- Take out the exercise booklet
- When you see the green square, there is an activity for you to do!

🟩 = Activity time!

- Get ready to type a lot!
- Don’t forget to take notes ;-)

The Story

- Your colleague **Jean Blue** opened an **LIR account**
- Jean Blue already did some things in the **Database**
- You were requested to take over some tasks
- You decided to come to **this training course**!

The RIPE Database

What is it?

Your LIR Account Was Activated

1. Read the email **1**
   - from the RIPE NCC Member Services department
2. Go to the TEST Database: https://apps-test.db.ripe.net
3. Search for the **person** object from the email

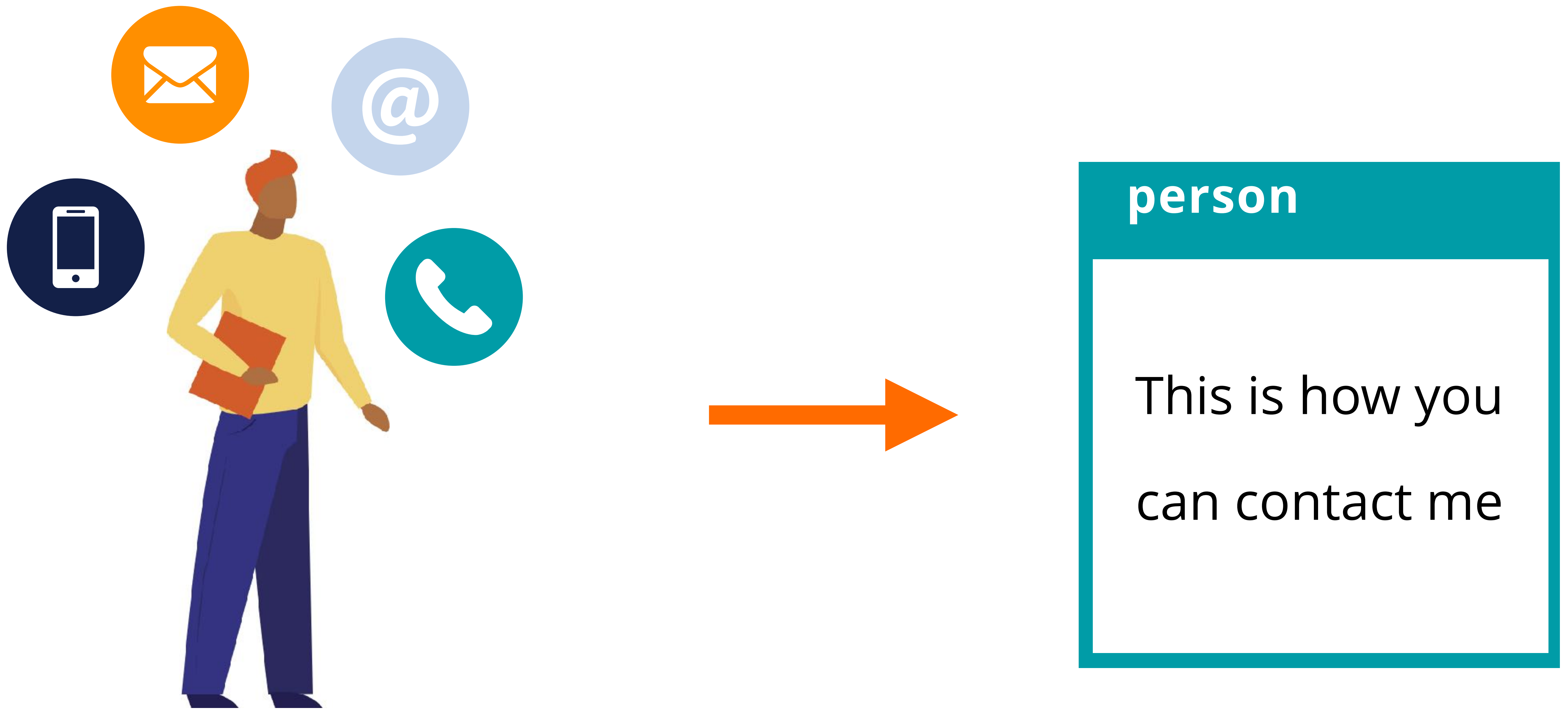
What Do You See?

- What do you get as a result?

- Which lines are not easy to understand?

What You Are Seeing

A **person** object has data that can be used to contact a real person

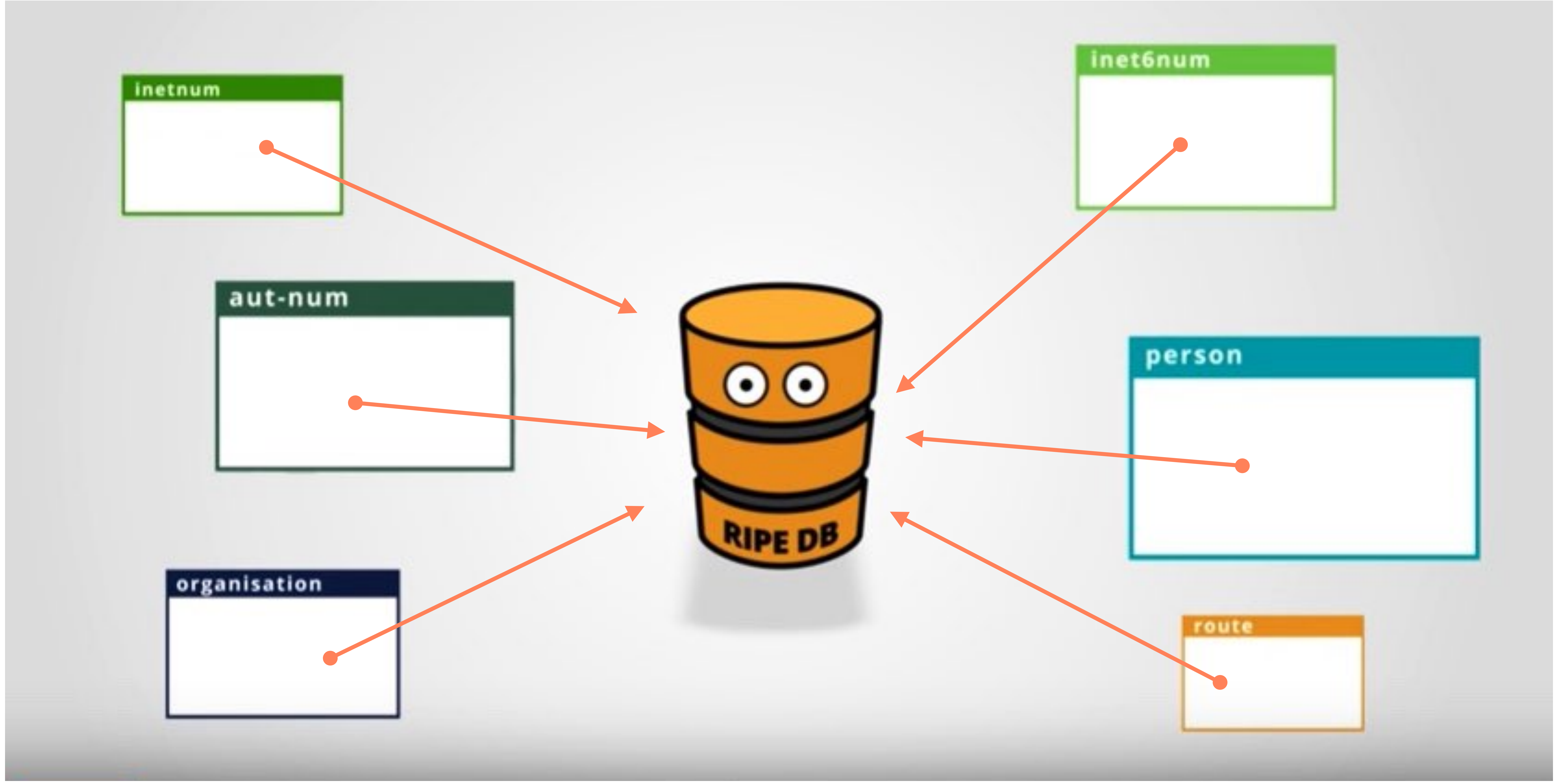


person

This is how you can contact me

The RIPE Database

Public **Internet resource** and **routing registry** database



Purpose of the RIPE Database

- Registry of **WHO** holds IPs and ASNs
- Keep **contact** information
  - For troubleshooting, notifying of outages, etc.
- Publishing **routing** policies
- Provisioning **reverse DNS**

RIPE Database Objects

IPs and ASNs

- inetnum
- inet6num
- aut-num

Contact Information

- organisation
- person
- role

Routing

- route
- route6
- as-set

Reverse DNS

- domain

Object Protection

- mntner

Looking Up Object Templates

1. Go to the TEST Database: http://apps-test.db.ripe.net

2. Search for the following:

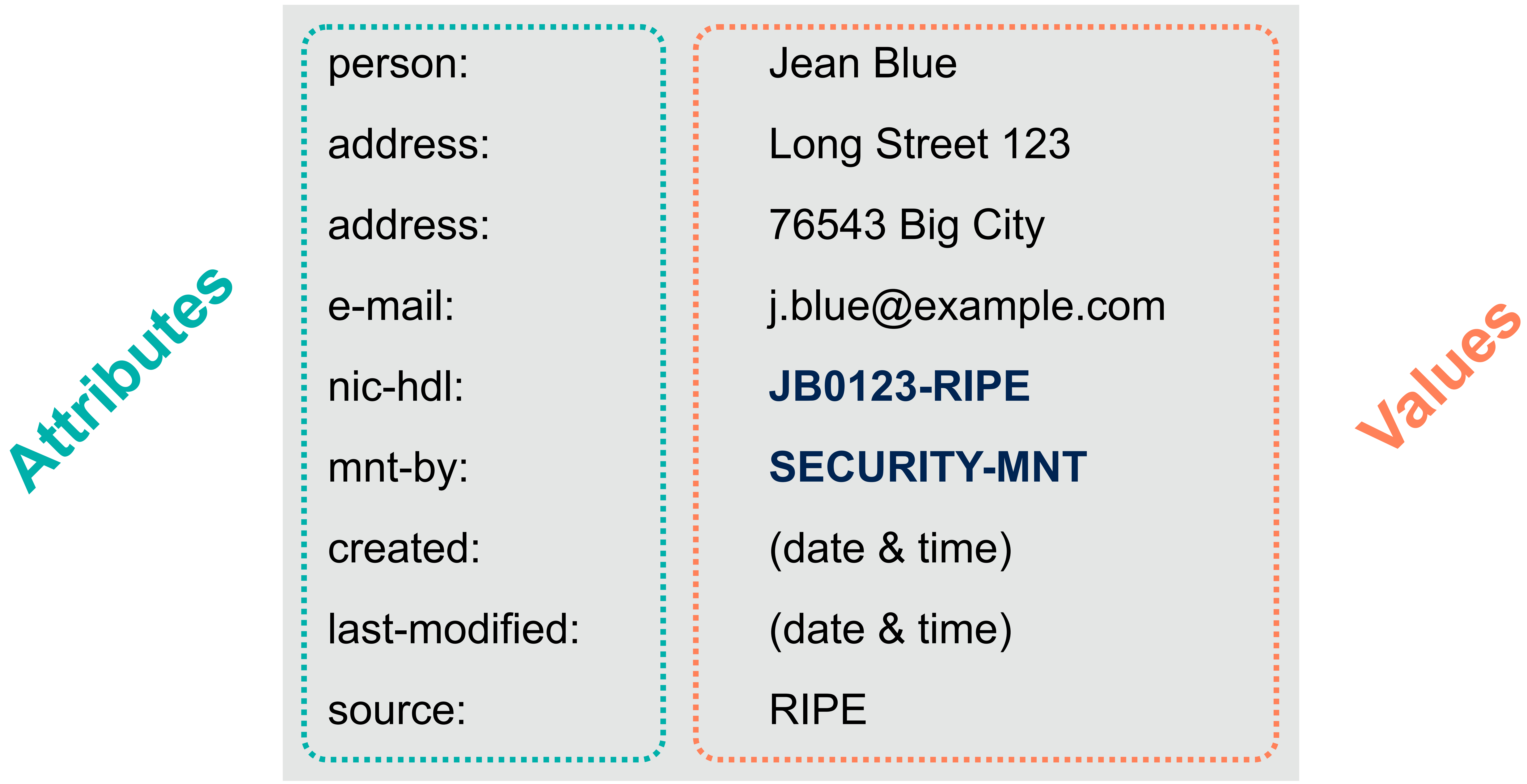
   -t person

- Alternatively, check the database manual:
- https://apps.db.ripe.net/docs/

What Do You See?

- What do you get as a result?
- What is not easy to understand?

Anatomy of an Object



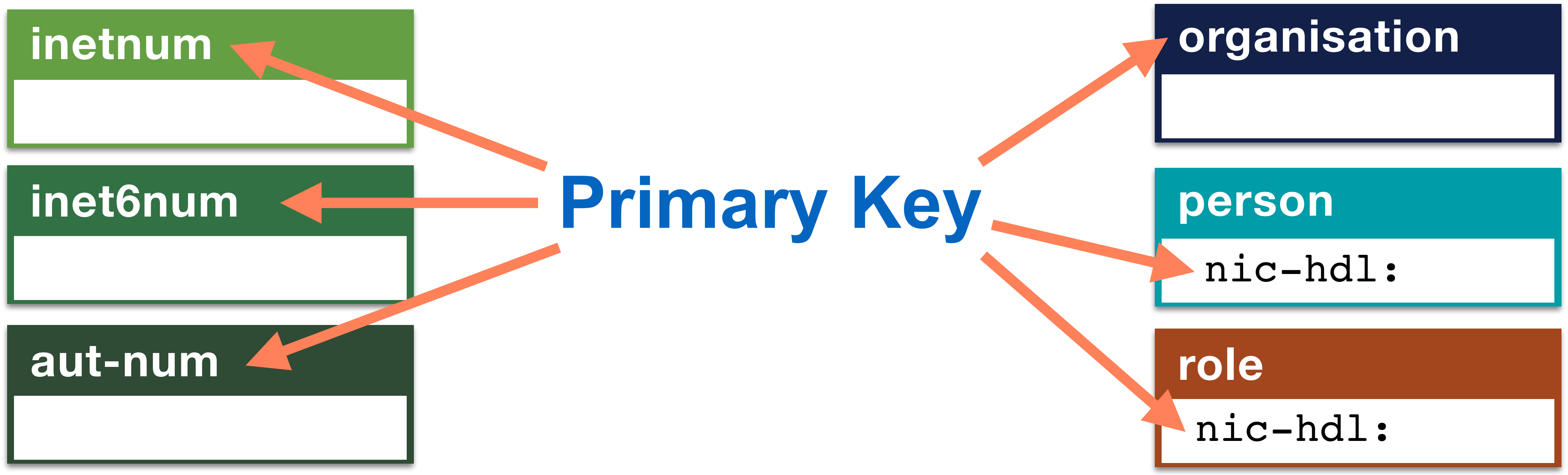
```
person:          Jean Blue
address:         Long Street 123
address:         76543 Big City
e-mail:          j.blue@example.com
nic-hdl:         JB0123-RIPE
mnt-by:          SECURITY-MNT
created:         (date & time)
last-modified:   (date & time)
source:          RIPE
```

Object Templates

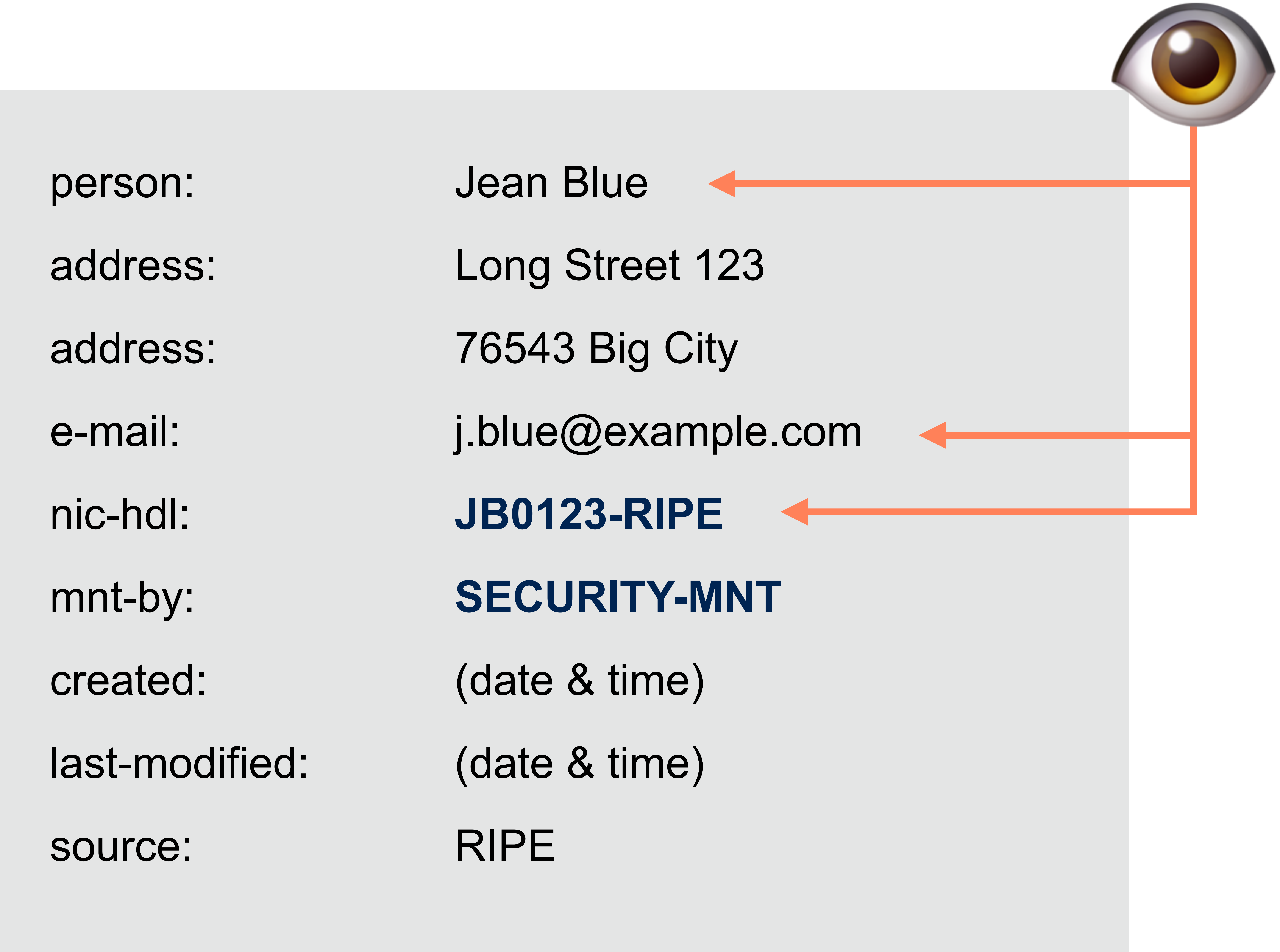
```
person:         [mandatory]   [single]     [lookup key]
address:        [mandatory]   [multiple]   [ ]
phone:          [mandatory]   [multiple]   [ ]
fax-no:         [optional]    [multiple]   [ ]
e-mail:         [optional]    [multiple]   [lookup key]
org:            [optional]    [multiple]   [inverse key]
nic-hdl:        [mandatory]   [single]     [primary/lookup key]
remarks:        [optional]    [multiple]   [ ]
notify:         [optional]    [multiple]   [inverse key]
mnt-by:         [mandatory]   [multiple]   [inverse key]
mnt-ref:        [optional]    [multiple]   [inverse key]
created:        [generated]   [single]     [ ]
last-modified:  [generated]   [single]     [ ]
source:         [mandatory]   [single]     [ ]
```

Primary Key

- Every object has one **Primary Key**
- It makes the object unique
  - Different from other objects of the same type



Lookup Keys



```
person:          Jean Blue
address:         Long Street 123
address:         76543 Big City
e-mail:          j.blue@example.com
nic-hdl:         JB0123-RIPE
mnt-by:          SECURITY-MNT
created:         (date & time)
last-modified:   (date & time)
source:          RIPE
```

Search For Your Organisation

1. Read the email 1 again
2. Go to https://apps-test.db.ripe.net
3. Search for the **organisation** object

What Do You See?

- What does the **organisation** object represent?
- Notice the “**admin-c:**” and “**tech-c:**” attributes
- What are their values?

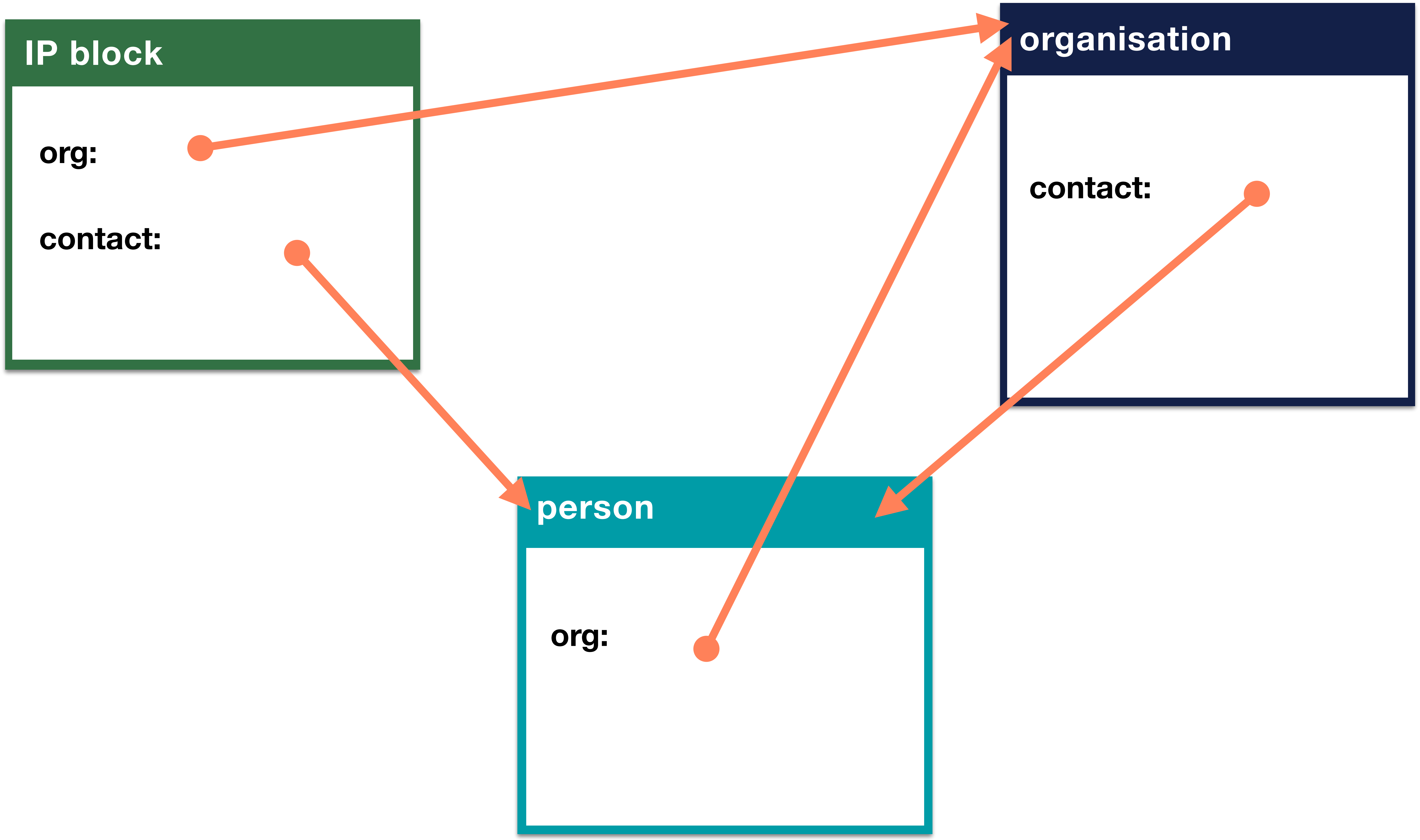
What You Are Seeing

An **organisation** object has data about a company, institution or any other kind of organisation that has IP addresses and AS Numbers

organisation

This is how you can contact ORG and who is responsible

Objects Are Linked To Each Other



admin-c

- Appears in most types of objects
- Name of **administrative** contact person(s)
- This is someone who will be contacted about administrative questions such as network registration, etc.

tech-c

- Appears in most types of objects
- Name of **technical** contact person(s)
- This is someone to be contacted for technical problems such as routing, (mis)behavior of hosts on the net, etc.

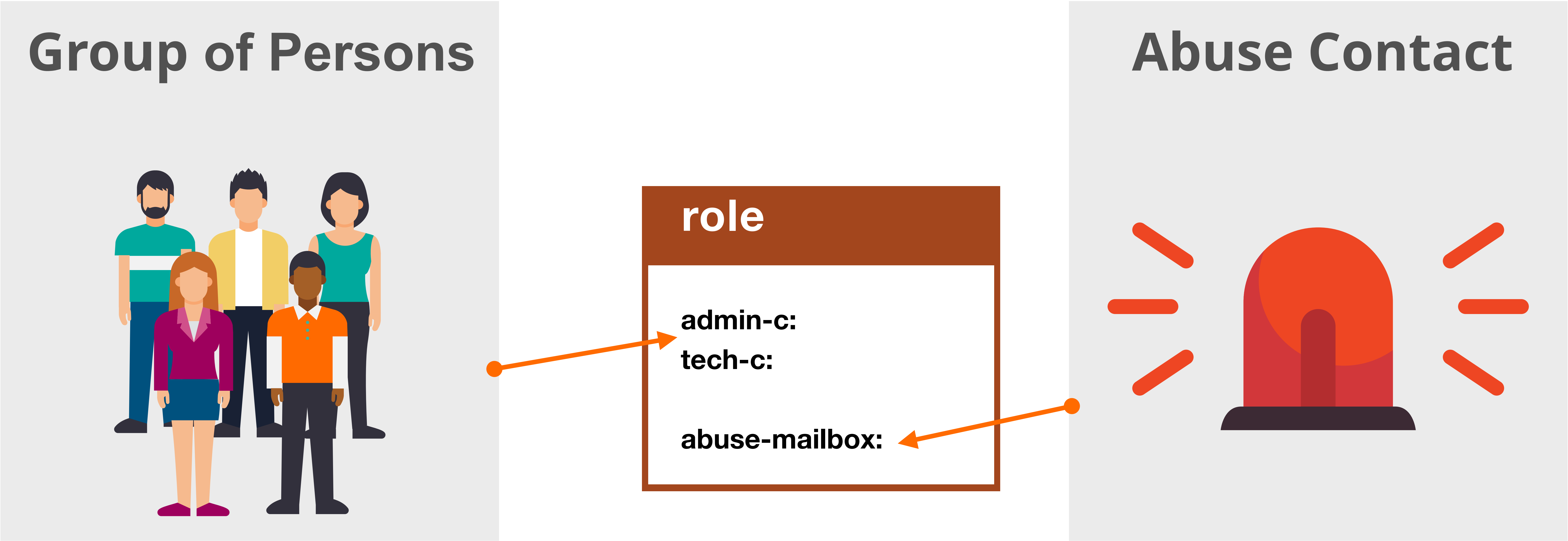
Search For Your ROLE Object

1. Read the email 1 again
2. Go to https://apps-test.db.ripe.net
3. Search for the **role** object

What Do You See?

- Notice the “admin-c:” and “tech-c:” attributes
- What are their values?
- Do you see any attribute that catches the eye?

Two Functions for the ROLE Object



Group of Persons

role

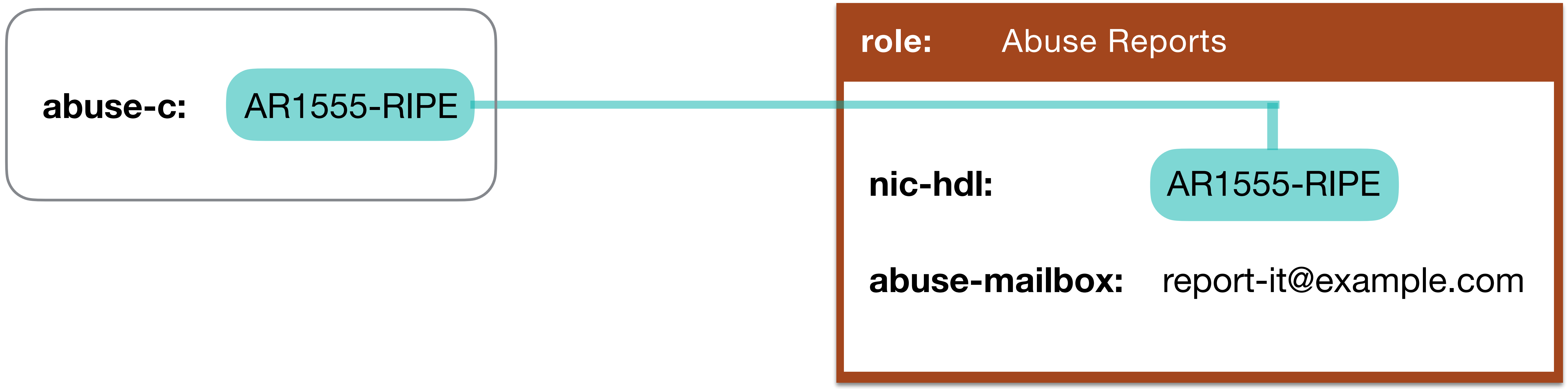
admin-c:
tech-c:

abuse-mailbox:

Abuse Contact

ROLE Object: Abuse Contact

- The **role** object contains the “abuse-mailbox:”
- Objects reference the **role** in “abuse-c:”
- RIPE Database shows the abuse contact in WHOIS query results



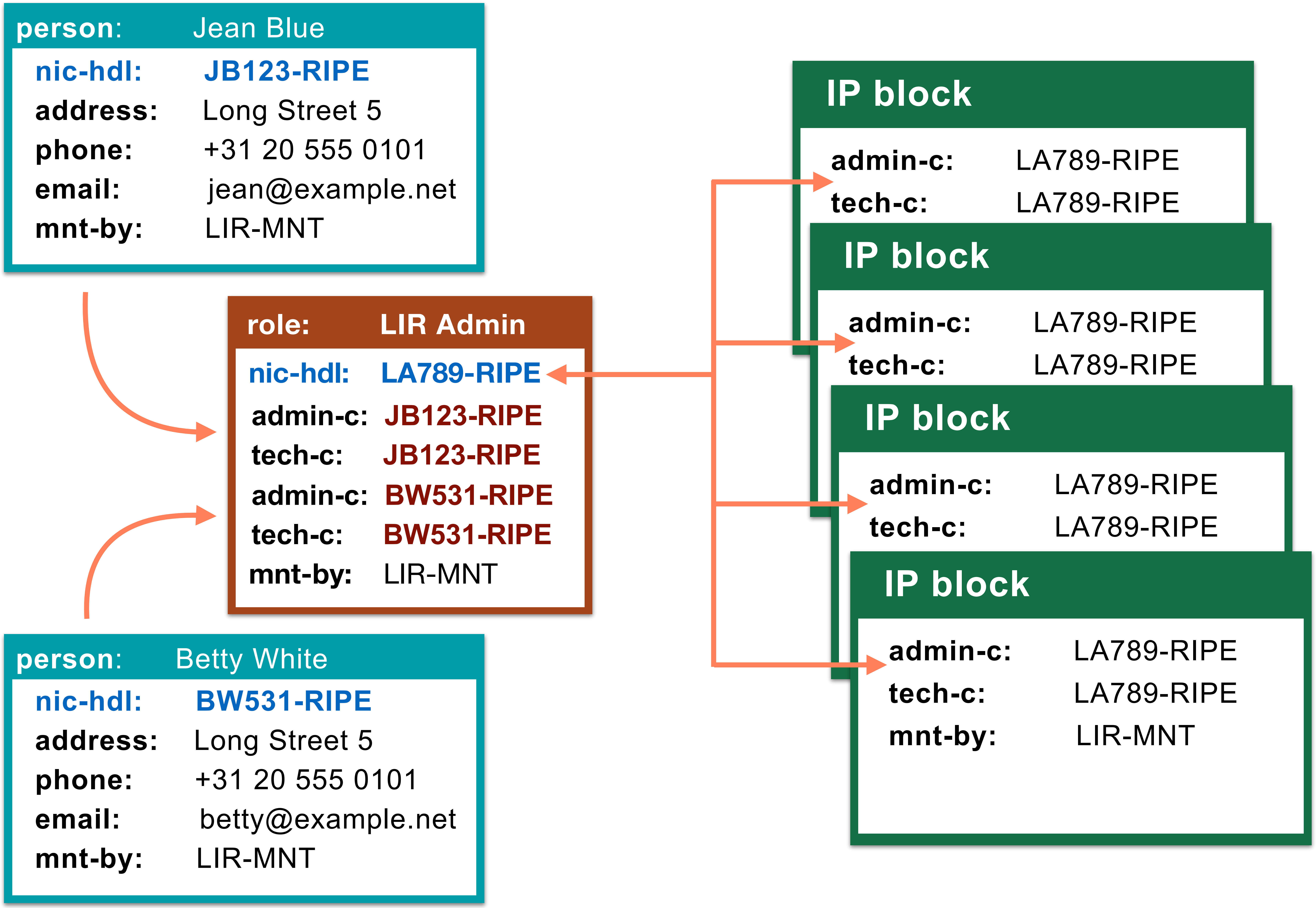
abuse-c: AR1555-RIPE

role: Abuse Reports

nic-hdl: AR1555-RIPE

abuse-mailbox: report-it@example.com

ROLE Object: Group of Persons



```
person:    Jean Blue
nic-hdl:   JB123-RIPE
address:   Long Street 5
phone:     +31 20 555 0101
email:     jean@example.net
mnt-by:    LIR-MNT
```

```
role:      LIR Admin
nic-hdl:   LA789-RIPE
admin-c:   JB123-RIPE
tech-c:    JB123-RIPE
admin-c:   BW531-RIPE
tech-c:    BW531-RIPE
mnt-by:    LIR-MNT
```

```
person:    Betty White
nic-hdl:   BW531-RIPE
address:   Long Street 5
phone:     +31 20 555 0101
email:     betty@example.net
mnt-by:    LIR-MNT
```

```
IP block
admin-c:   LA789-RIPE
tech-c:    LA789-RIPE
```

```
IP block
admin-c:   LA789-RIPE
tech-c:    LA789-RIPE
```

```
IP block
admin-c:   LA789-RIPE
tech-c:    LA789-RIPE
```

```
IP block
admin-c:   LA789-RIPE
tech-c:    LA789-RIPE
mnt-by:    LIR-MNT
```

Questions ?

How Does It Work?

Looking for data in the Database

Search For Your Allocations

- Read emails 2 and 3
  - from the Registry Services department
- Go to http://apps-test.db.ripe.net
- Search for the **inetnum** and **inet6num** objects
  - Open two tabs or windows if needed!
  - Use the text in the “inetnum:” and “inet6num:” lines
    - i.e. 10.**XX**.0.0 - 10.**XX**.3.255
    - i.e. 2002:ff**XX**::/32

What Do You See?

- Look at the first object in the results
- What do you see?
- How many objects did you get?

Network Objects

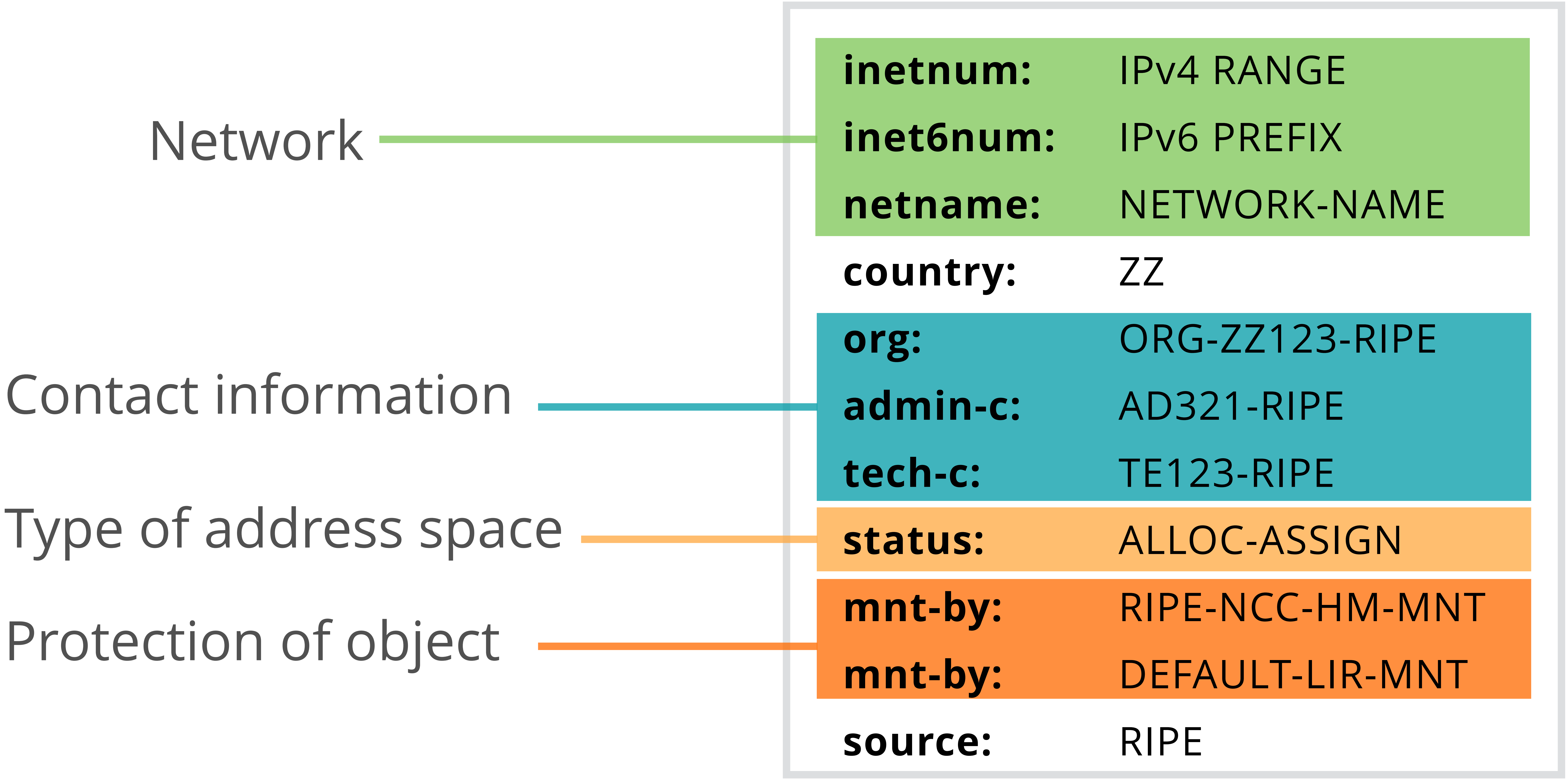
IPv4 = inetnum

```
inetnum: 192.30.0.0 - 192.30.3.255

netname:    NL-NETWORK-20170101
country:    NL
org:        ORG-EE2-RIPE
admin-c:    DV789-RIPE
tech-c:     JS123-RIPE
status:     ALLOCATED PA
mnt-by:     RIPE-NCC-HM-MNT
mnt-by:     DEFAULT-LIR-MNT
source:     RIPE
```

IPv6 = inet6num

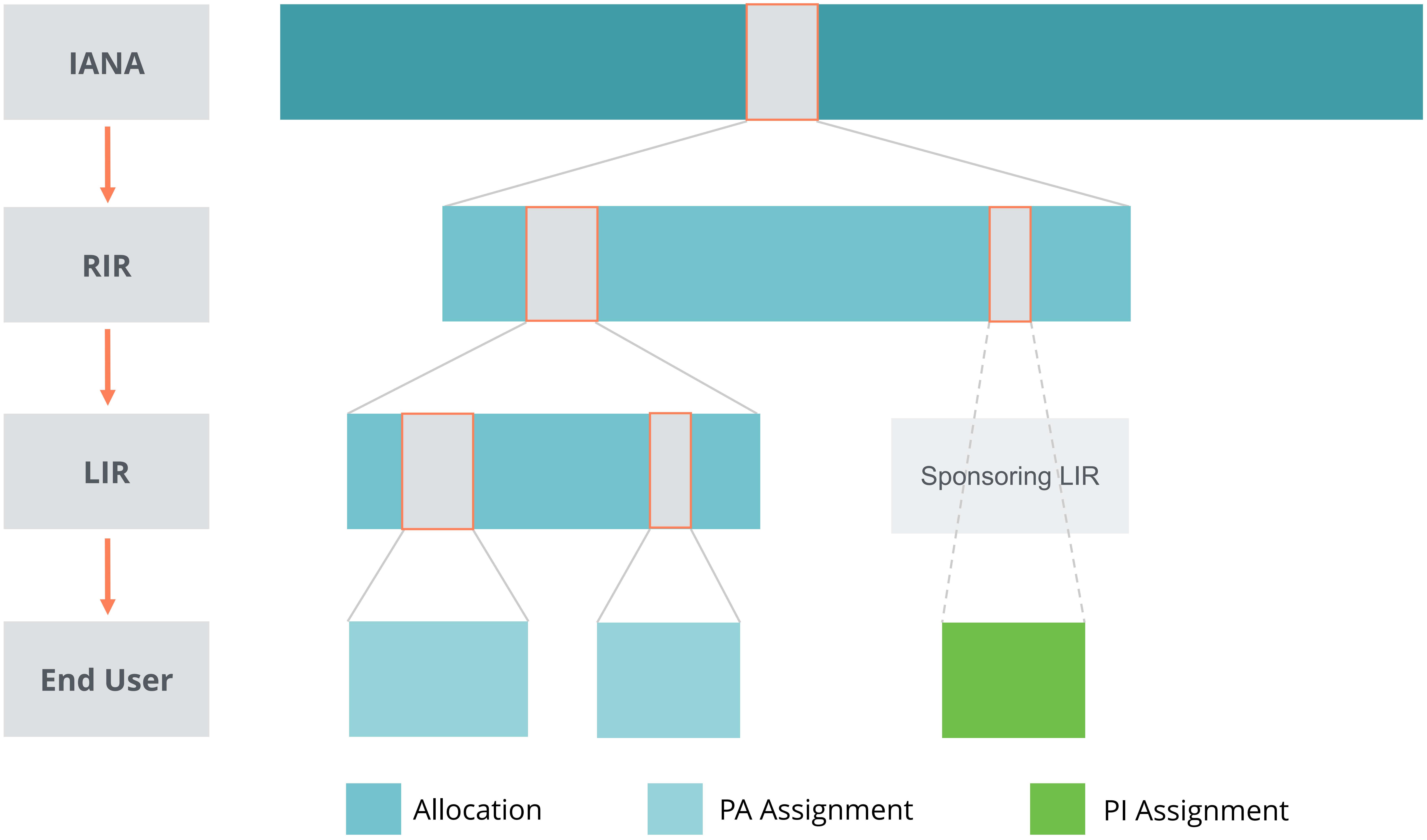
```
inet6num:   2001:db8::/32

netname:    NL-NETWORK-20170101
country:    NL
org:        ORG-EE2-RIPE
admin-c:    DV789-RIPE
tech-c:     JS123-RIPE
status:     ALLOCATED-BY-RIR
mnt-by:     RIPE-NCC-HM-MNT
mnt-by:     DEFAULT-LIR-MNT
source:     RIPE
```

- Same object structure for IPv4 and IPv6



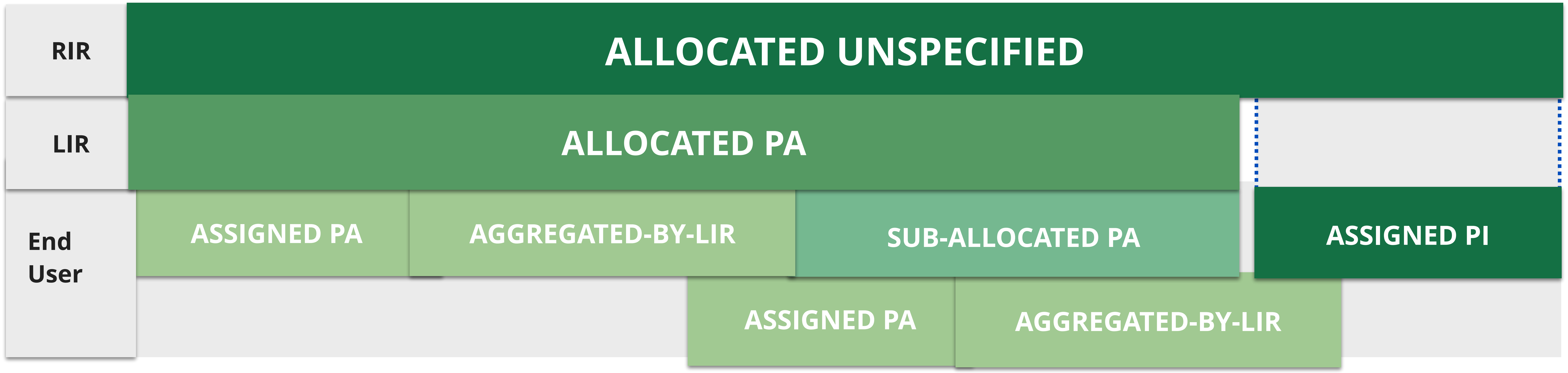
```
inetnum:      IPv4 RANGE
inet6num:     IPv6 PREFIX
netname:      NETWORK-NAME
country:      ZZ
org:          ORG-ZZ123-RIPE
admin-c:      AD321-RIPE
tech-c:       TE123-RIPE
status:       ALLOC-ASSIGN
mnt-by:       RIPE-NCC-HM-MNT
mnt-by:       DEFAULT-LIR-MNT
source:       RIPE
```

Hierarchical Distribution

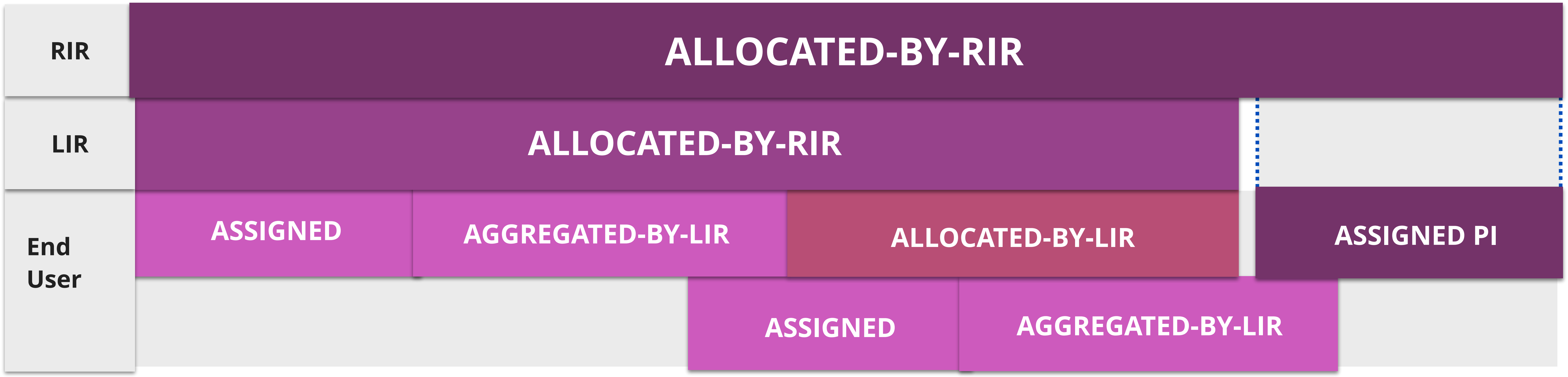


Object Status Hierarchy

IPv4

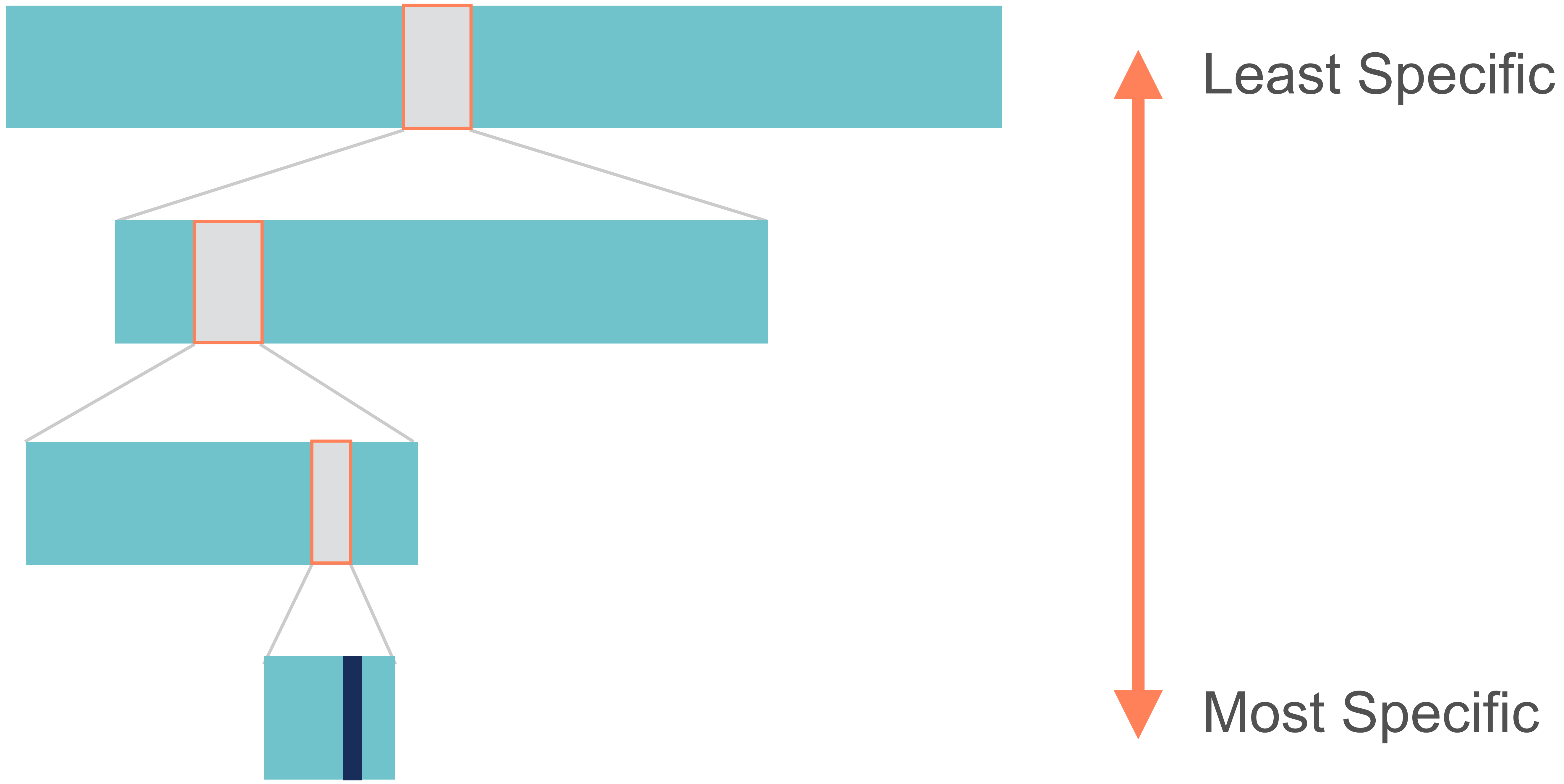


IPv6



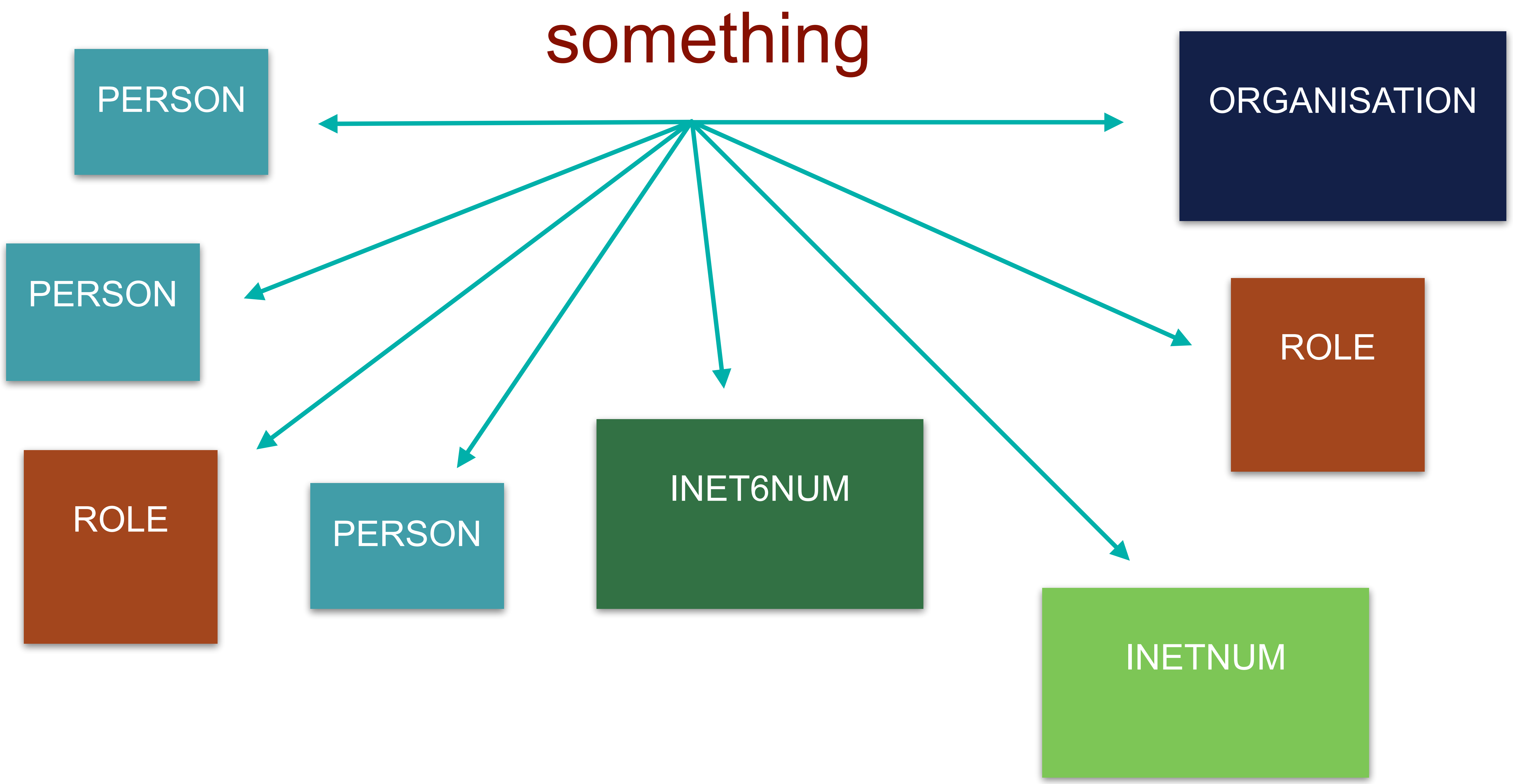
Default Query Results

- When you query for an **IP address** or **prefix**...



Default Query Results

- When you query for simple **text**...



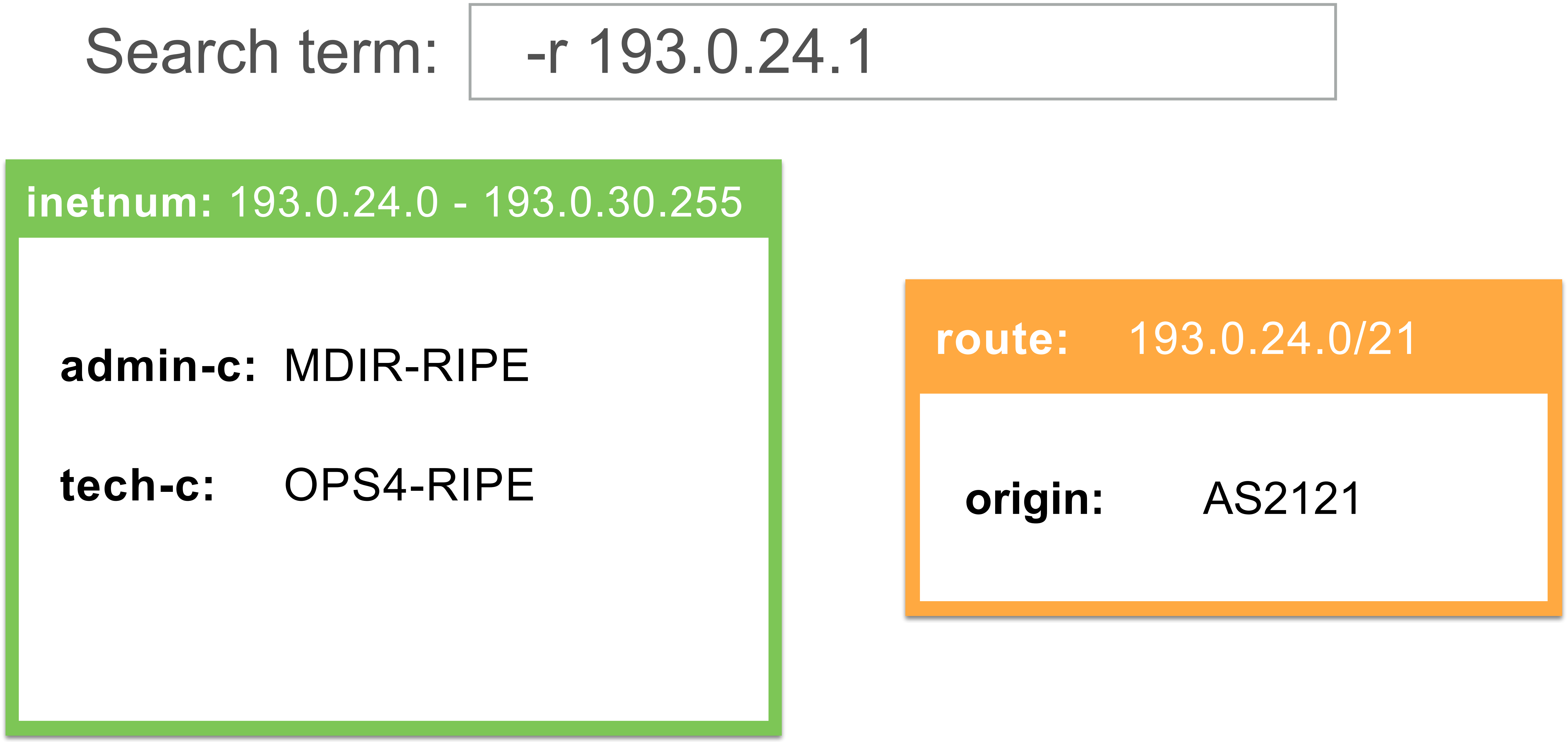
Filtered Query Results

- All email addresses are filtered
- Show them with **-B** flag in query
  - Or turn on “Show full object details”
- “auth:” attribute values are always filtered

```
person:     Jean Blue

nic-hdl:    JB123-RIPE
address:    Long Street 5
phone:      +31 20 555 0101
mnt-by:     LIR-MNT
source:     RIPE # Filtered
```

```
mntner:     LIR-MNT

admin-c:    JB123-RIPE
auth:       SSO # Filtered
auth:       SSO # Filtered
auth:       PGP-KEY-54321
mnt-by:     LIR-MNT
source:     RIPE # Filtered
```

Results Without Related Objects



Search term: -r 193.0.24.1

inetnum: 193.0.24.0 - 193.0.30.255

admin-c: MDIR-RIPE

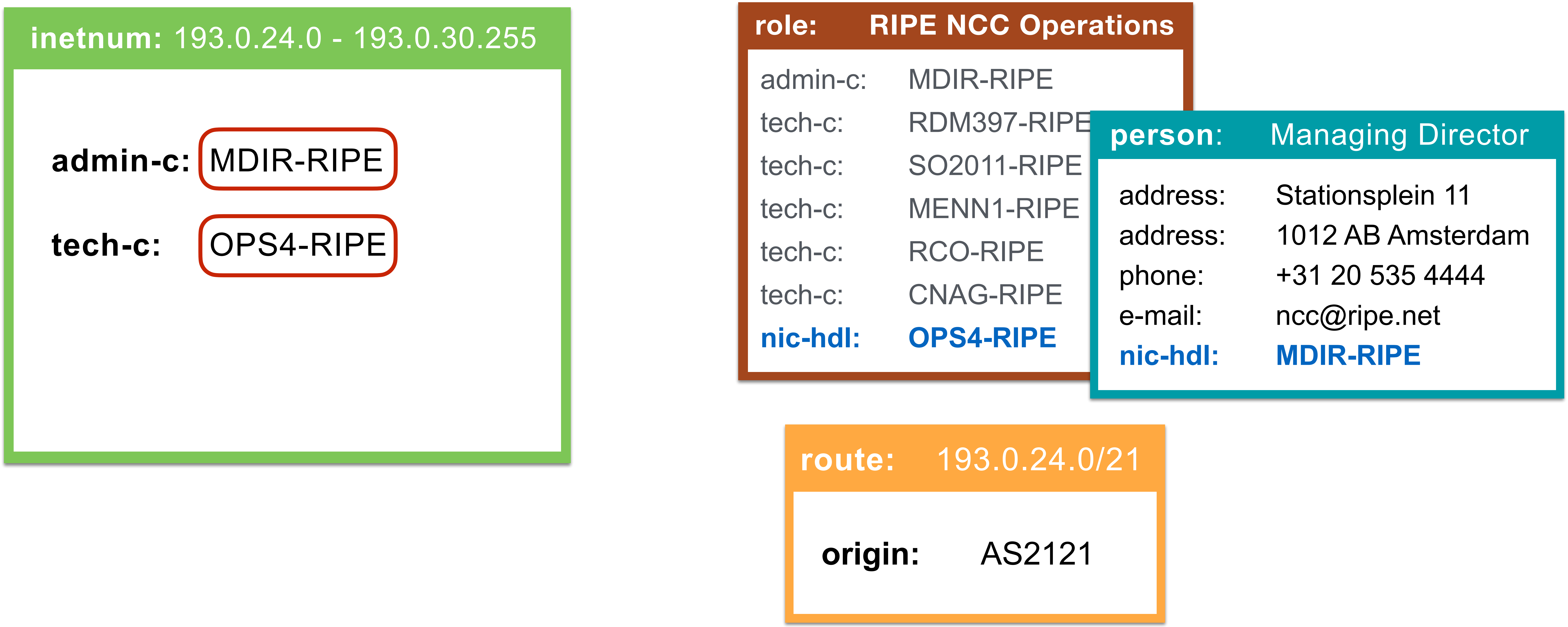
tech-c: OPS4-RIPE

route: 193.0.24.0/21

origin: AS2121

Results With Related Objects

Search term: 193.0.24.1



inetnum: 193.0.24.0 - 193.0.30.255

admin-c: MDIR-RIPE

tech-c: OPS4-RIPE

role: RIPE NCC Operations

admin-c: MDIR-RIPE

tech-c: RDM397-RIPE

tech-c: SO2011-RIPE

tech-c: MENN1-RIPE

tech-c: RCO-RIPE

tech-c: CNAG-RIPE

nic-hdl: OPS4-RIPE

person: Managing Director

address: Stationsplein 11

address: 1012 AB Amsterdam

phone: +31 20 535 4444

e-mail: ncc@ripe.net

nic-hdl: MDIR-RIPE

route: 193.0.24.0/21

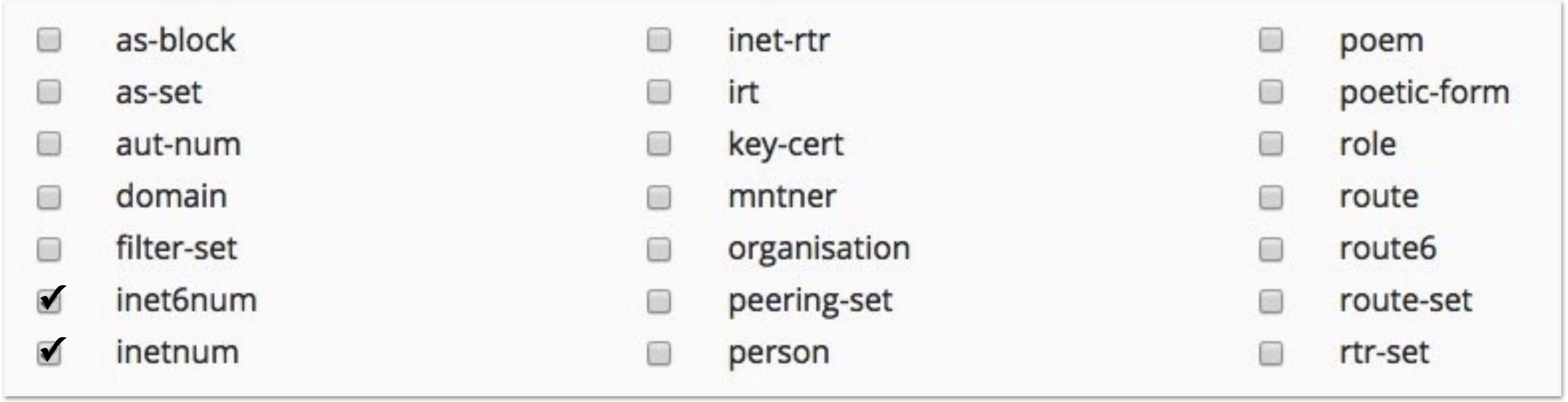
origin: AS2121

Making Better Queries

- Reduce the amount of objects returned
- Use options and flags to optimise the results
- Avoid getting blocked!

Selecting Object Types

- Choose the types of objects you want to see
- This results in fewer objects to process



☐ as-block
☐ as-set
☐ aut-num
☐ domain
☐ filter-set
☑ inet6num
☑ inetnum
☐ inet-rtr
☐ irt
☐ key-cert
☐ mntner
☐ organisation
☐ peering-set
☐ person
☐ poem
☐ poetic-form
☐ role
☐ route
☐ route6
☐ route-set
☐ rtr-set

- Using a flag: **-T inetnum**

Search For Your Allocations Again

1. In the previous query windows, turn **off** “*Do not retrieve related objects*”

2. Search again for the **inetnum** and **inet6num** objects

What Do You See?

- Look at all the objects in the results
- How many objects did you get now?
- Which objects are now in the results?

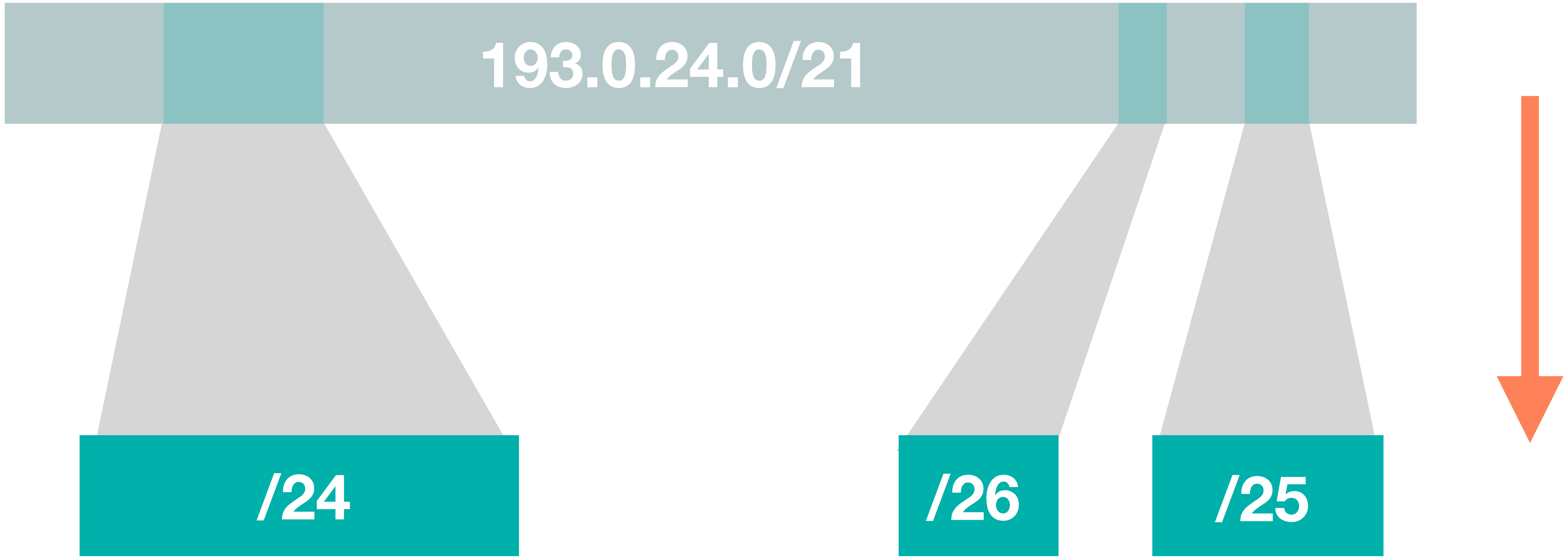
Navigating the Hierarchy

- Using flags, you can find what is under or above an inet(6)num object
  - Under = More Specific
  - Above = Less Specific

- The flags: **-m, -M, -l, -L**
- Also in the “Hierarchy Flags” tab

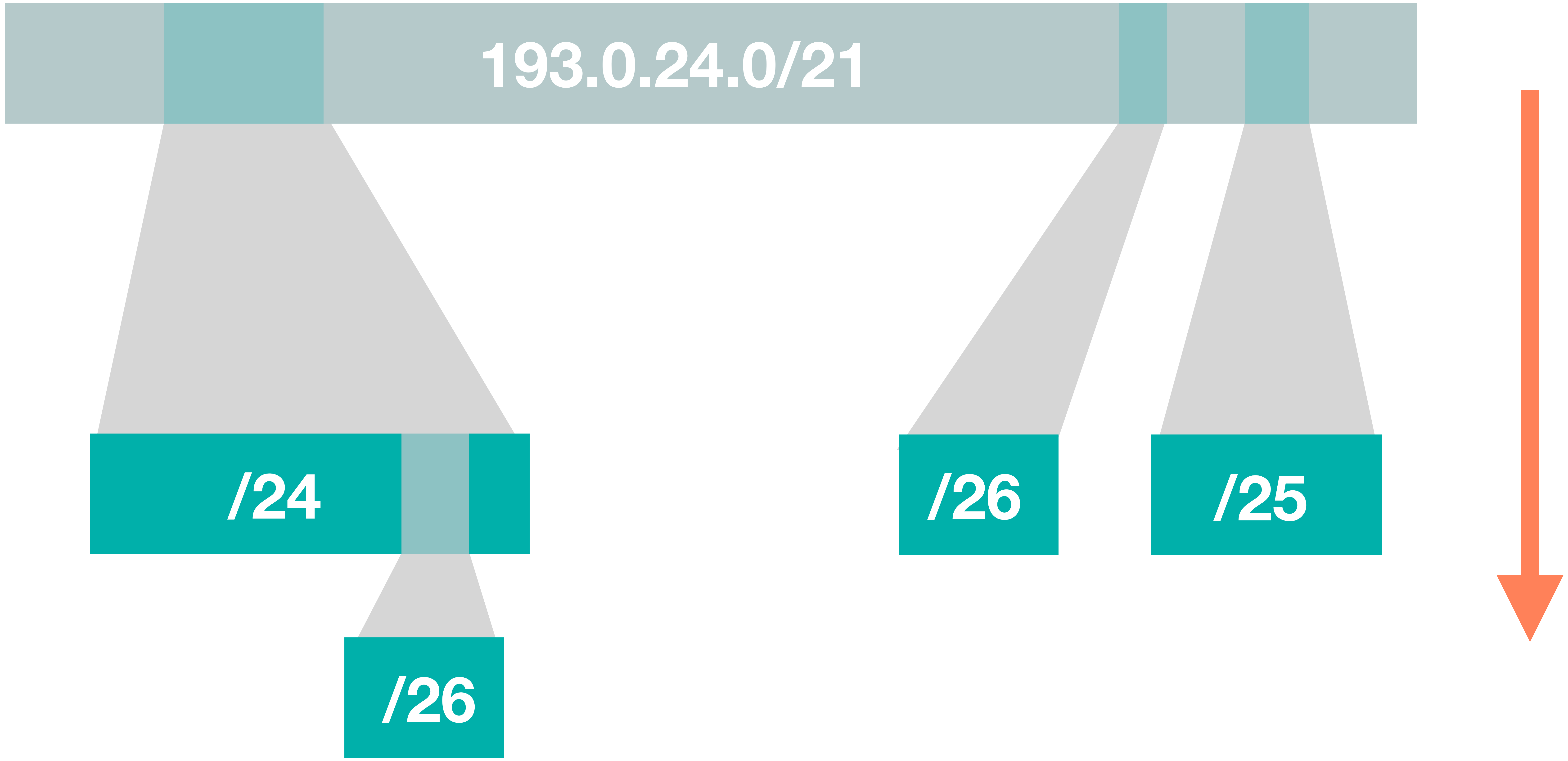
More Specific INETNUMs: -m

-m 193.0.24.0/21



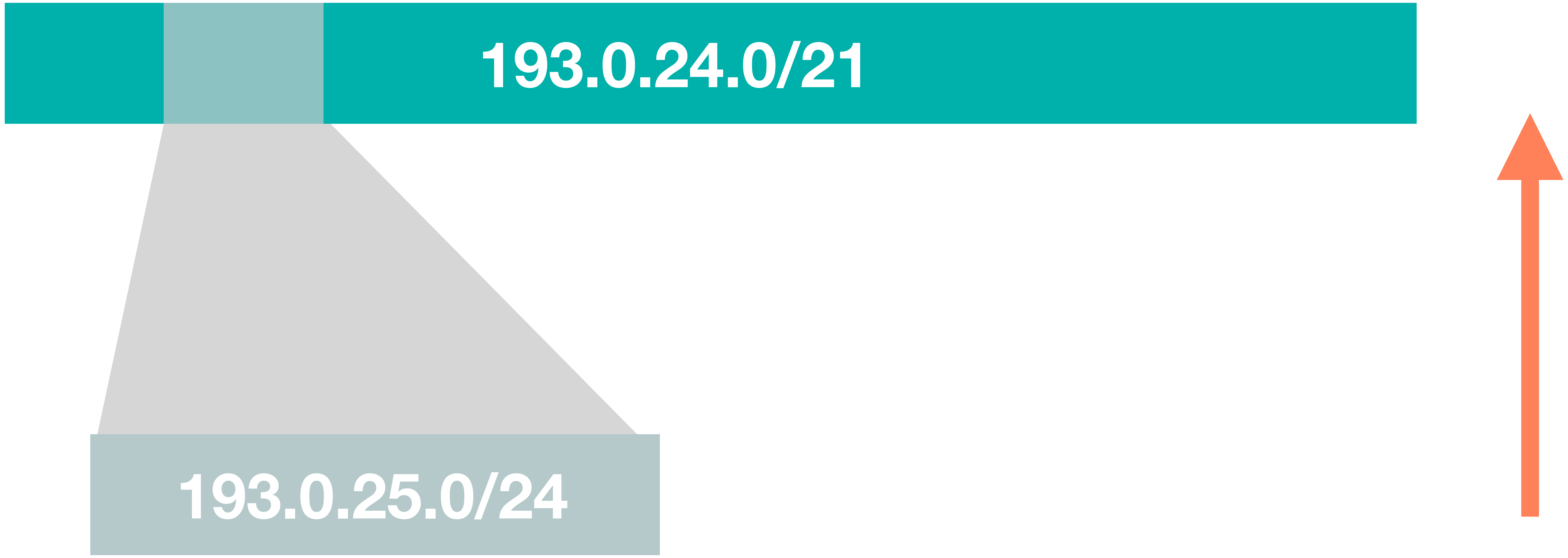
More Specific INETNUMs: -M

-M 193.0.24.0/21



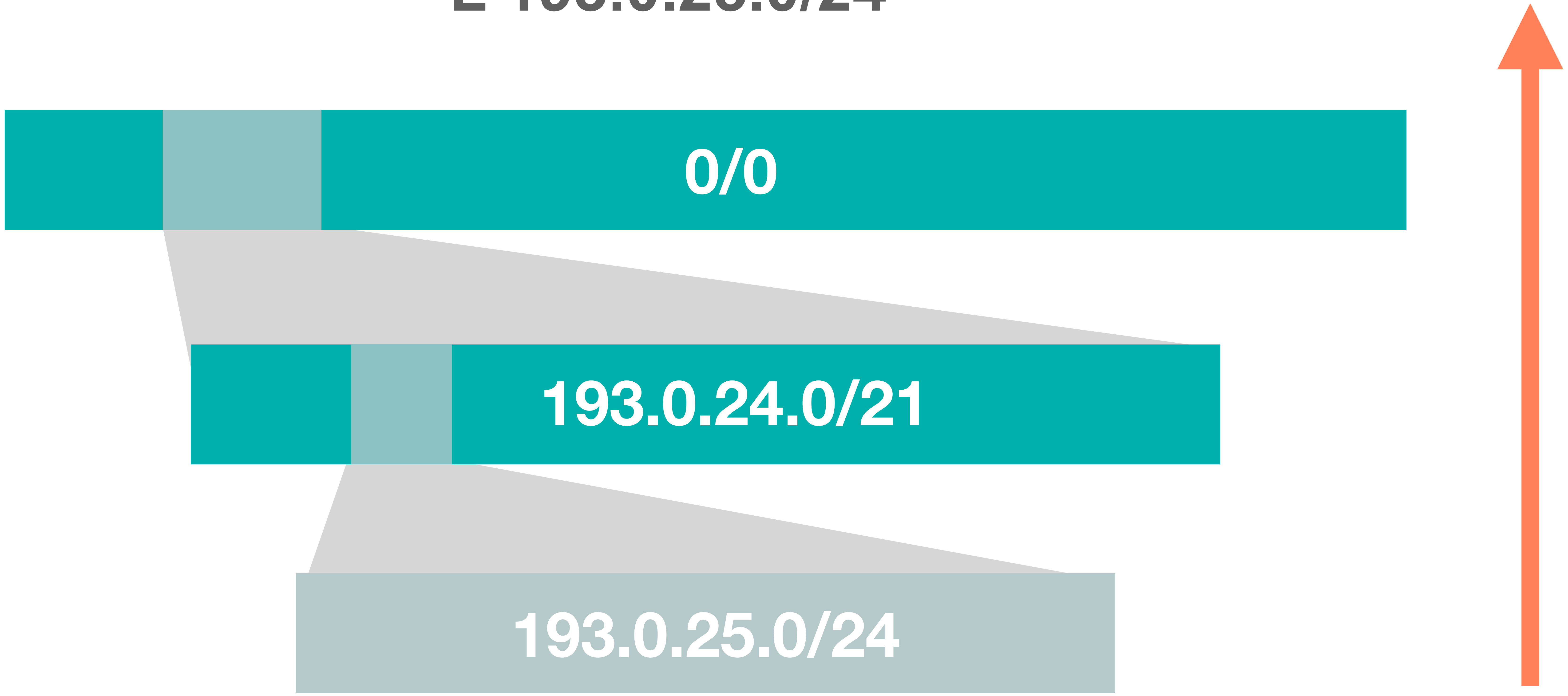
Less Specific INETNUMs: -l

-l 193.0.25.0/24



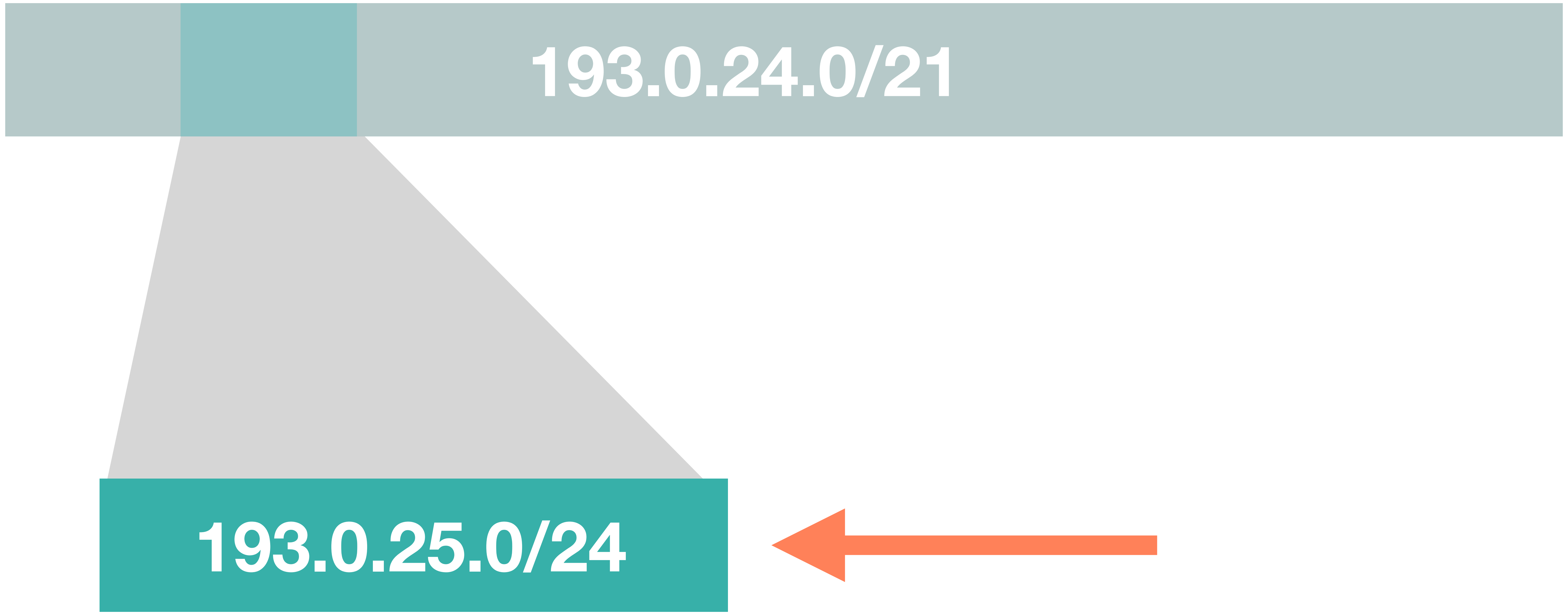
Less Specific INETNUMs: -L

-L 193.0.25.0/24



Exact Match: -x

-x 193.0.25.0/24



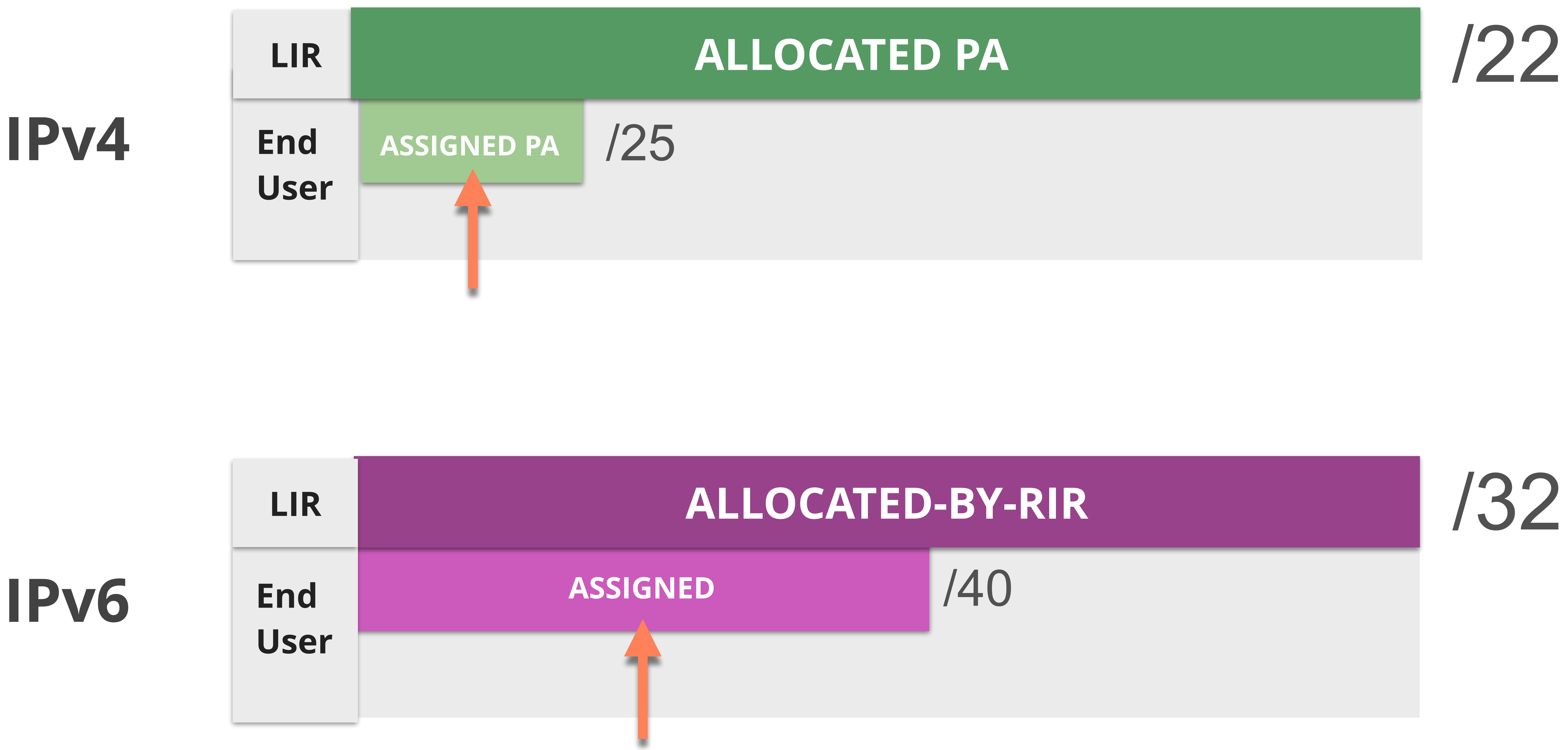
Search For Your Allocations Again

1. In the previous query windows, add “-m” to the search text
    - i.e. **-m** 10.XX.0.0 - 10.XX.3.255
    - i.e. **-m** 2002:ffXX::/32
2. Search again for the **inetnum** and **inet6num** objects

What Do You See?

- Look at the objects in the results
- How many objects did you get now?
- Different from what you got before?
  - Notice the “**status:**” attribute

What You Are Seeing



Questions

How To Update It?

Updating the RIPE Database
Part 1

Updating: What You Need

- To **update** the RIPE Database you must have:
  - a RIPE NCC **Access** account
  - a **maintainer** object
  - the **need** to create, update or delete an object!

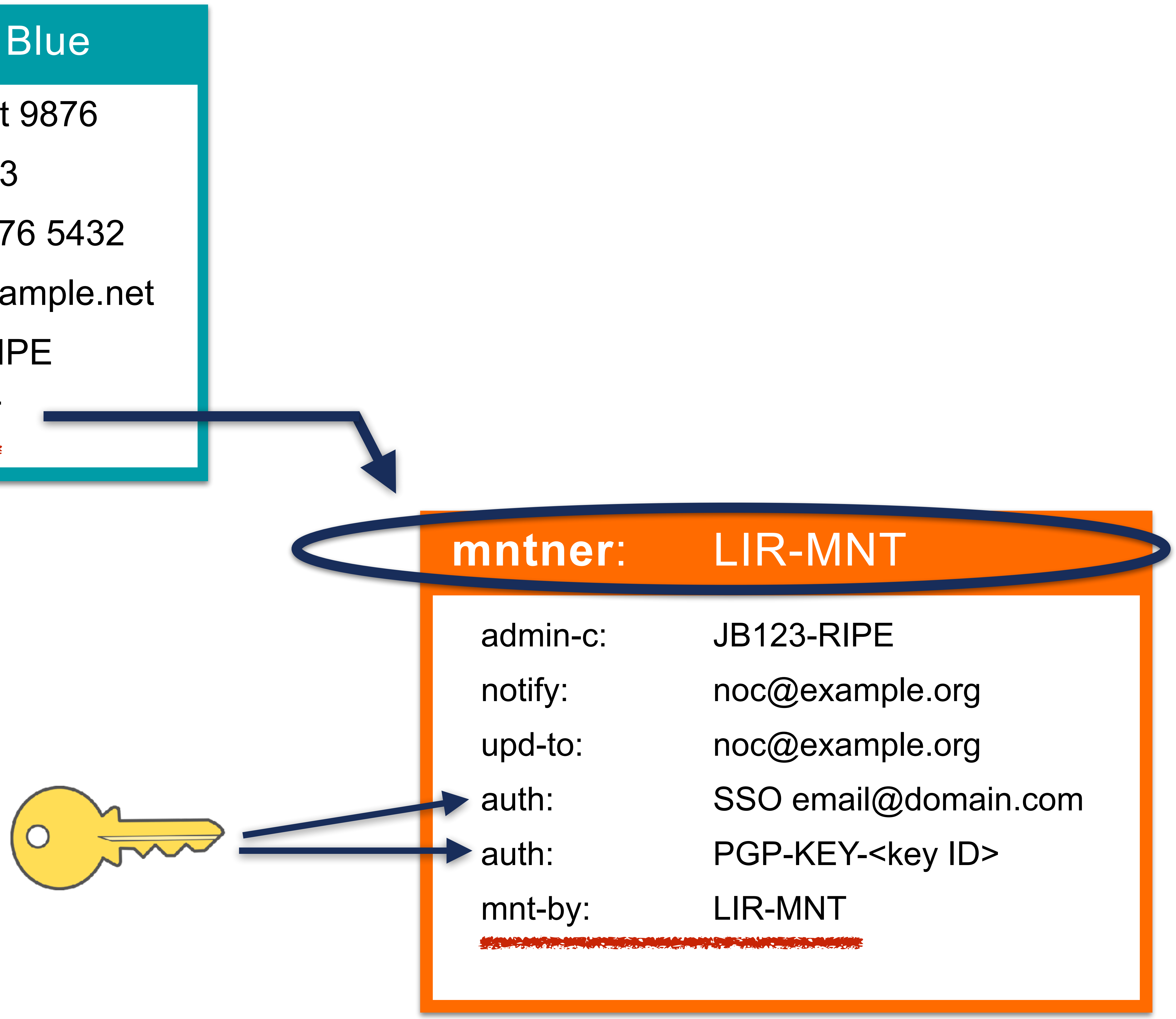
Search for LIR Maintainer Object

1. Read the email **5**
   - from your colleague Jean Blue
2. Go to http://apps-test.db.ripe.net
3. Search for the **maintainer** object
   - i.e. SMXX-MNT

What Do You See?

- Look at the “**mnt-by:**” attribute
- What is the value?
- Look at the “**auth:**” attribute
- What is the value?

Maintainers: Protecting Objects



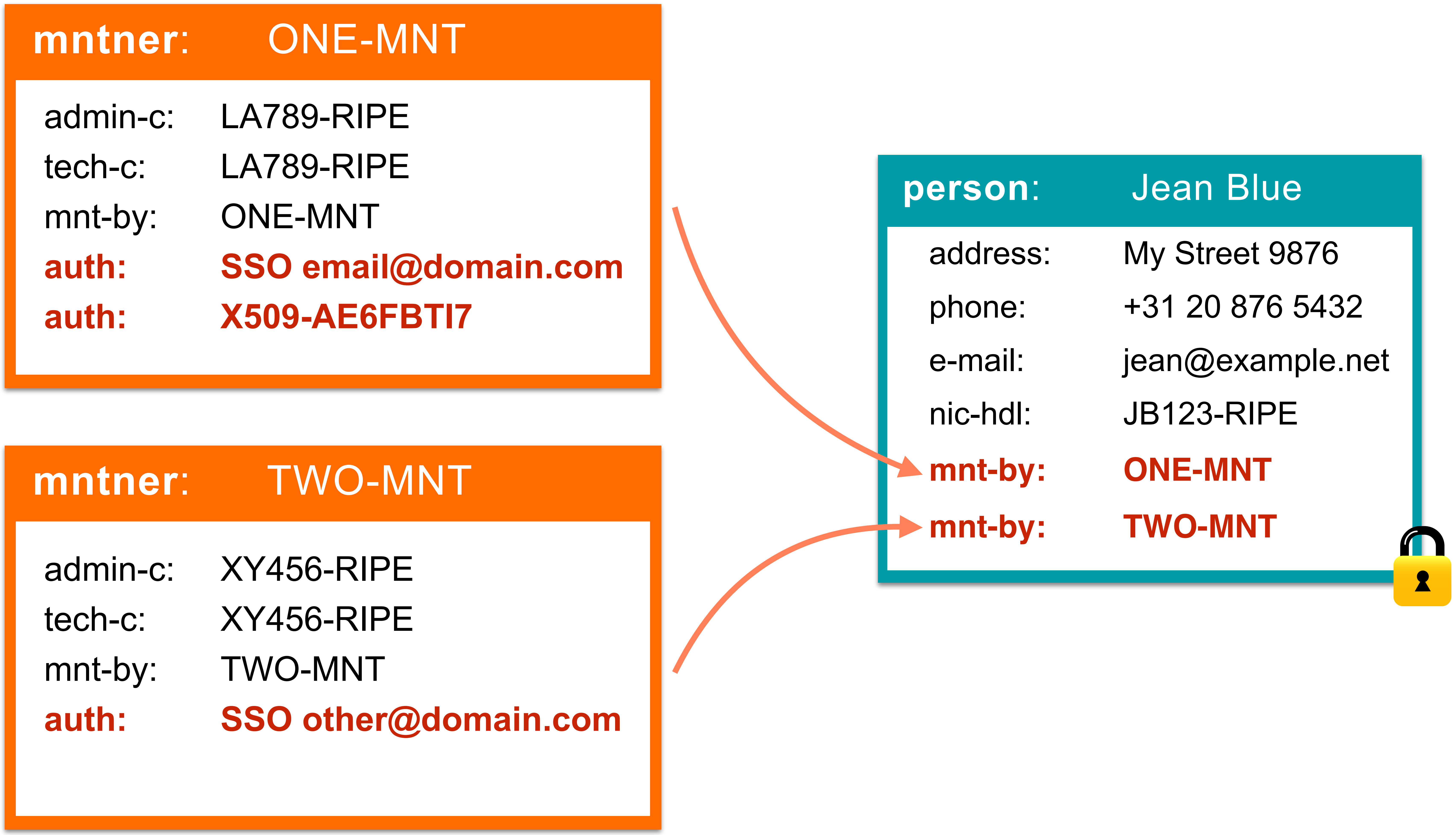
```
person:     Jean Blue
address:    My Street 9876
address:    Office 123
phone:      +31 20 876 5432
e-mail:     jean@example.net
nic-hdl:    JB123-RIPE
mnt-by:     LIR-MNT
```

```
mntner:     LIR-MNT
admin-c:    JB123-RIPE
notify:     noc@example.org
upd-to:     noc@example.org
auth:       SSO email@domain.com
auth:       PGP-KEY-<key ID>
mnt-by:     LIR-MNT
```

Maintainers: Authentication

- **SSO**
  - default authentication mechanism
  - uses RIPE NCC Access account
  - to authenticate: login on RIPE NCC website
- **PGP / x509**
  - uses PGP key pair or x509 certificates
  - to authenticate: sign updates with private key
- **API Keys**
  - not really an authentication mechanism
  - allows you to authenticate with a maintainer that has your SSO
  - requires yearly maintenance

Multiple Maintainers



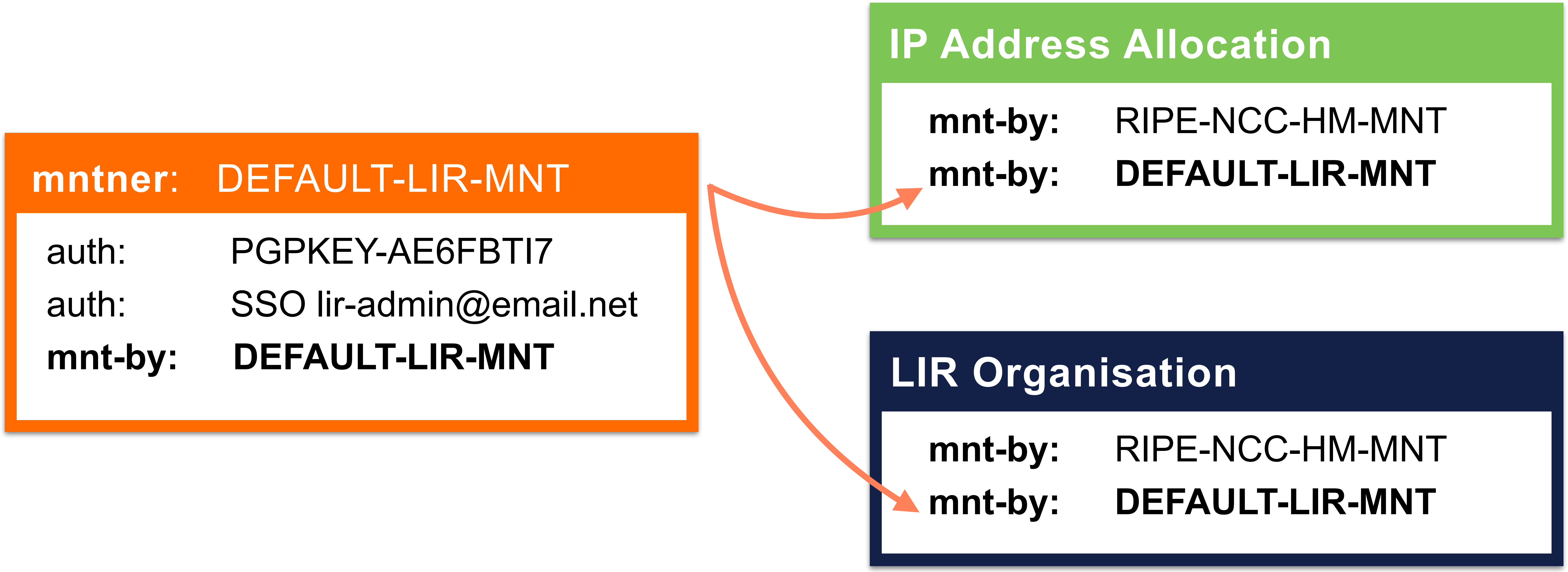
```
mntner:     ONE-MNT
admin-c:    LA789-RIPE
tech-c:     LA789-RIPE
mnt-by:     ONE-MNT
auth:       SSO email@domain.com
auth:       X509-AE6FBTI7
```

```
mntner:     TWO-MNT
admin-c:    XY456-RIPE
tech-c:     XY456-RIPE
mnt-by:     TWO-MNT
auth:       SSO other@domain.com
```

```
person:     Jean Blue
address:    My Street 9876
phone:      +31 20 876 5432
e-mail:     jean@example.net
nic-hdl:    JB123-RIPE
mnt-by:     ONE-MNT
mnt-by:     TWO-MNT
```

Default Maintainer for LIRs

- Allows partial control over Allocation and ORG
- Can be selected in the LIR Account Details
- Automatically reflected in the RIPE Database



mntner: DEFAULT-LIR-MNT

auth: PGPKEY-AE6FBTI7
auth: SSO lir-admin@email.net
mnt-by: DEFAULT-LIR-MNT

IP Address Allocation

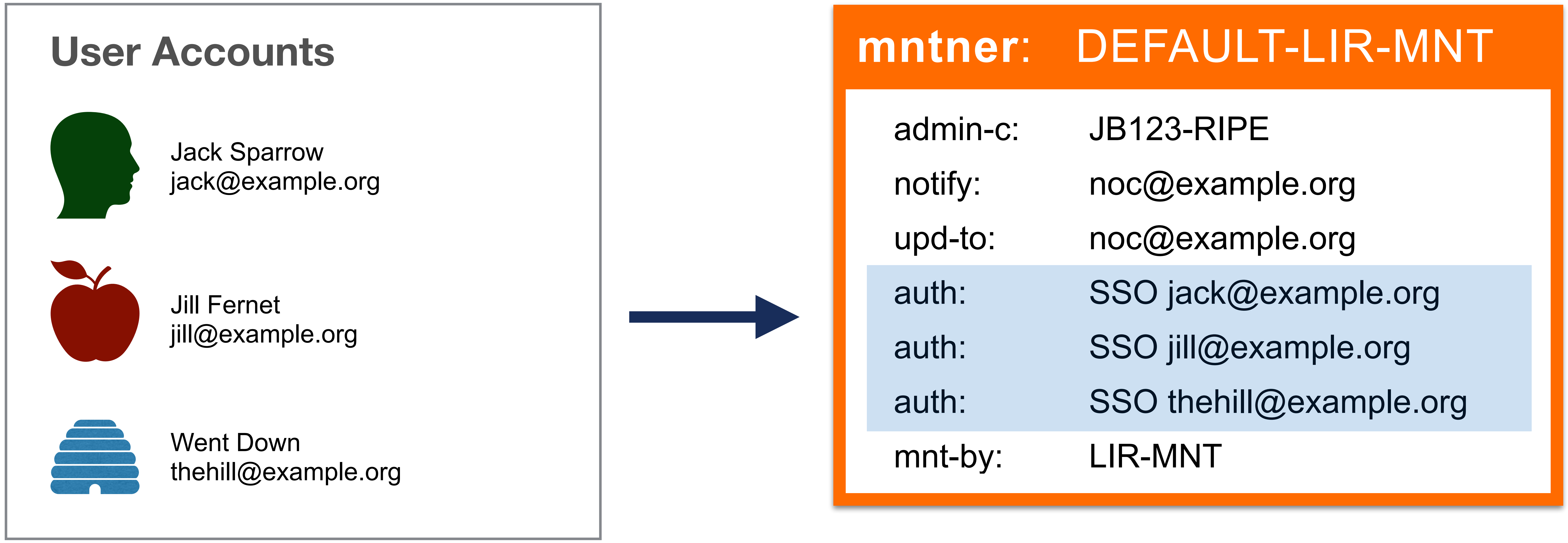
mnt-by: RIPE-NCC-HM-MNT
mnt-by: DEFAULT-LIR-MNT

LIR Organisation

mnt-by: RIPE-NCC-HM-MNT
mnt-by: DEFAULT-LIR-MNT

Synch With LIR Portal

- Default LIR Maintainer can be synchronised with LIR Portal
- Users added as SSO to the maintainer
- Previous “auth:” lines are removed



User Accounts

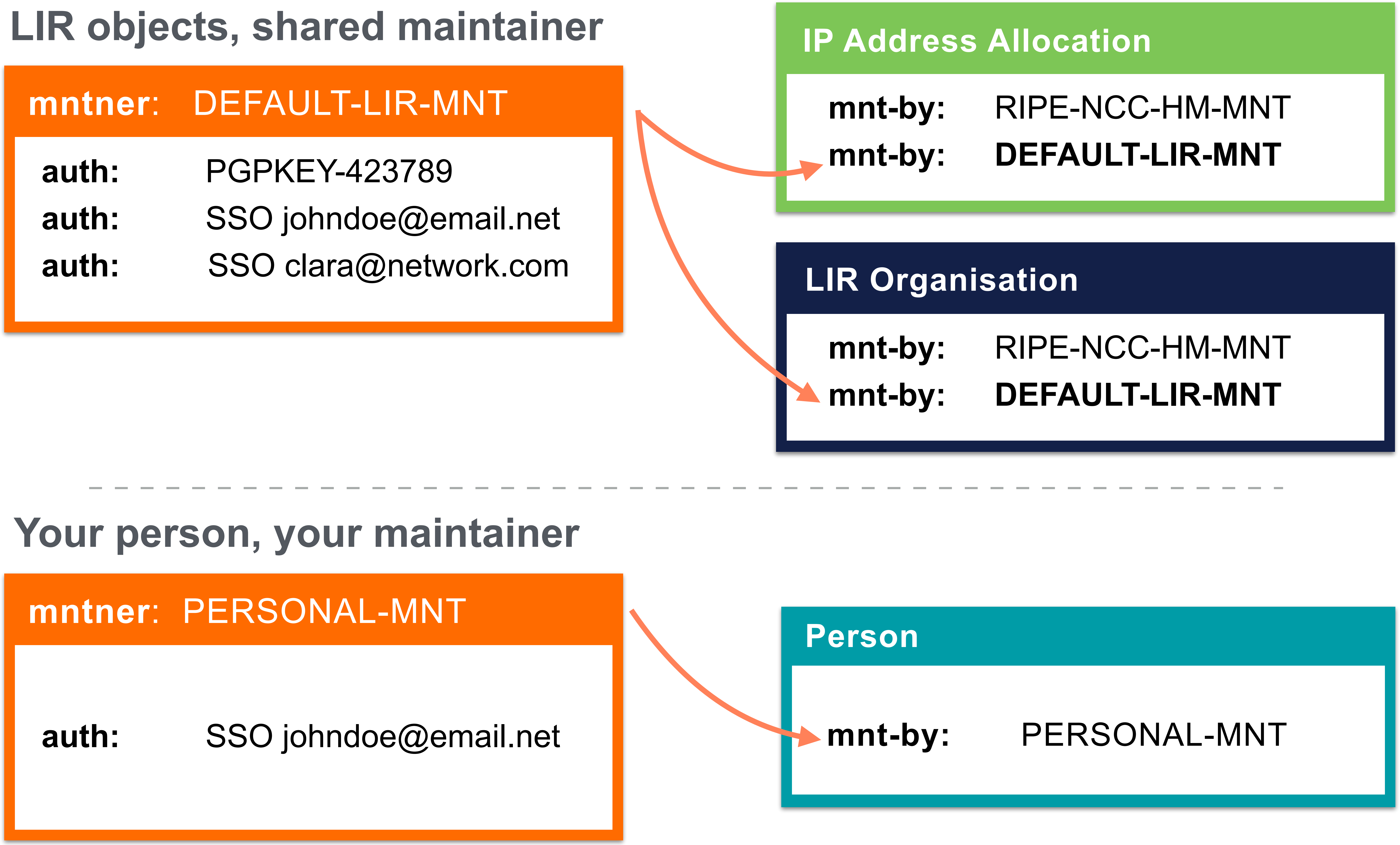
Jack Sparrow
jack@example.org

Jill Fernet
jill@example.org

Went Down
thehill@example.org

```
mntner:   DEFAULT-LIR-MNT
admin-c:  JB123-RIPE
notify:   noc@example.org
upd-to:   noc@example.org
auth:     SSO jack@example.org
auth:     SSO jill@example.org
auth:     SSO thehill@example.org
mnt-by:   LIR-MNT
```

Personal vs Shared



LIR objects, shared maintainer

mntner: DEFAULT-LIR-MNT

auth: PGPKEY-423789
auth: SSO johndoe@email.net
auth: SSO clara@network.com

IP Address Allocation

mnt-by: RIPE-NCC-HM-MNT
mnt-by: **DEFAULT-LIR-MNT**

LIR Organisation

mnt-by: RIPE-NCC-HM-MNT
mnt-by: **DEFAULT-LIR-MNT**

Your person, your maintainer

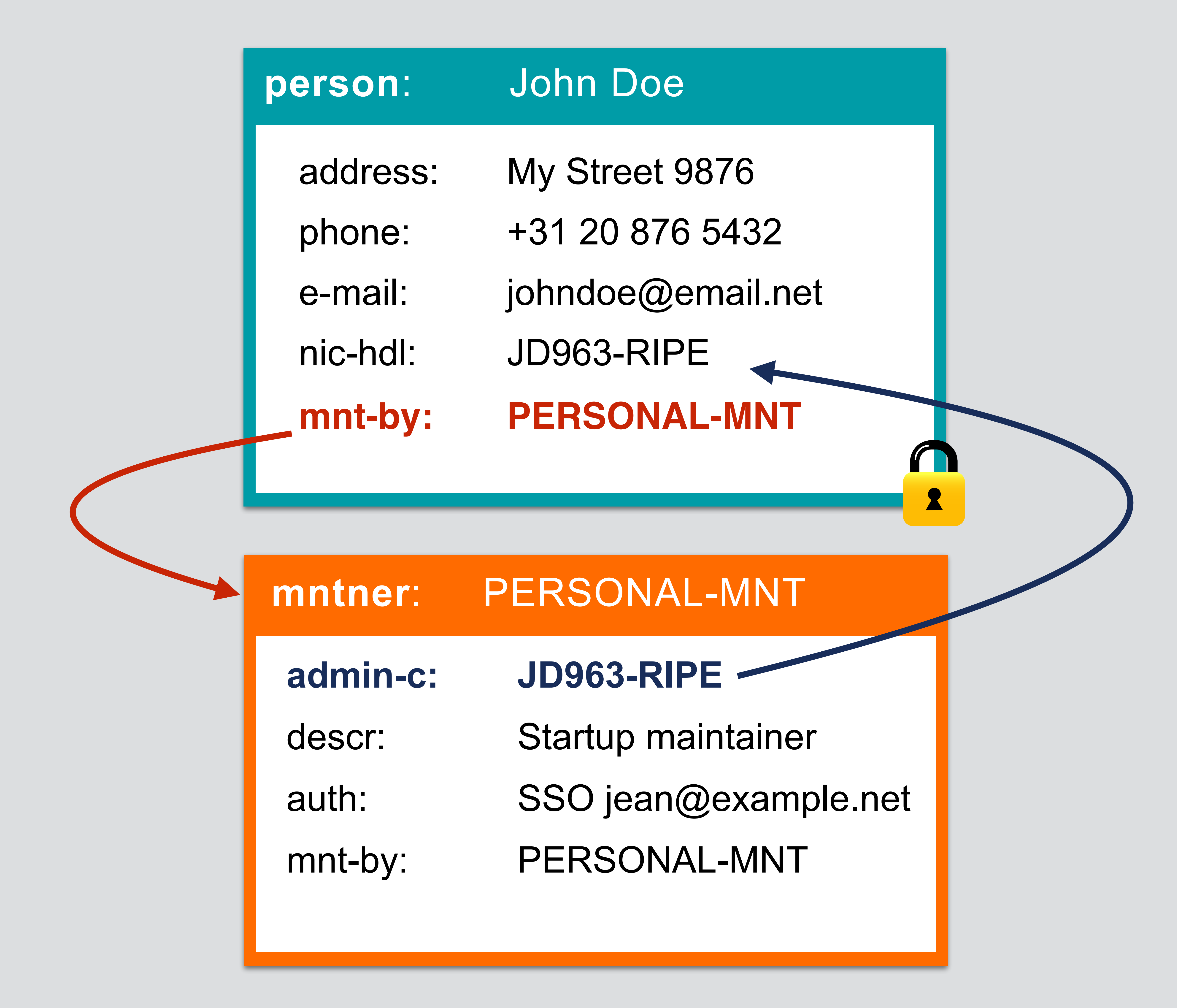
mntner: PERSONAL-MNT

auth: SSO johndoe@email.net

Person

mnt-by: PERSONAL-MNT

MAINTAINER and PERSON



```
person:     John Doe

address:    My Street 9876
phone:      +31 20 876 5432
e-mail:     johndoe@email.net
nic-hdl:    JD963-RIPE
mnt-by:     PERSONAL-MNT
```

```
mntner:     PERSONAL-MNT

admin-c:    JD963-RIPE
descr:      Startup maintainer
auth:       SSO jean@example.net
mnt-by:     PERSONAL-MNT
```

Creating Your PERSON/MNTNER Pair

1. Read again the email **5**
   - from your colleague Jean Blue
2. Go to http://apps-test.db.ripe.net
3. On the left side, click on “**Create an object**”
4. Choose ”**role and maintainer pair**”
5. Click on [**Create**]
6. On the empty **role** template, switch to “**person**”

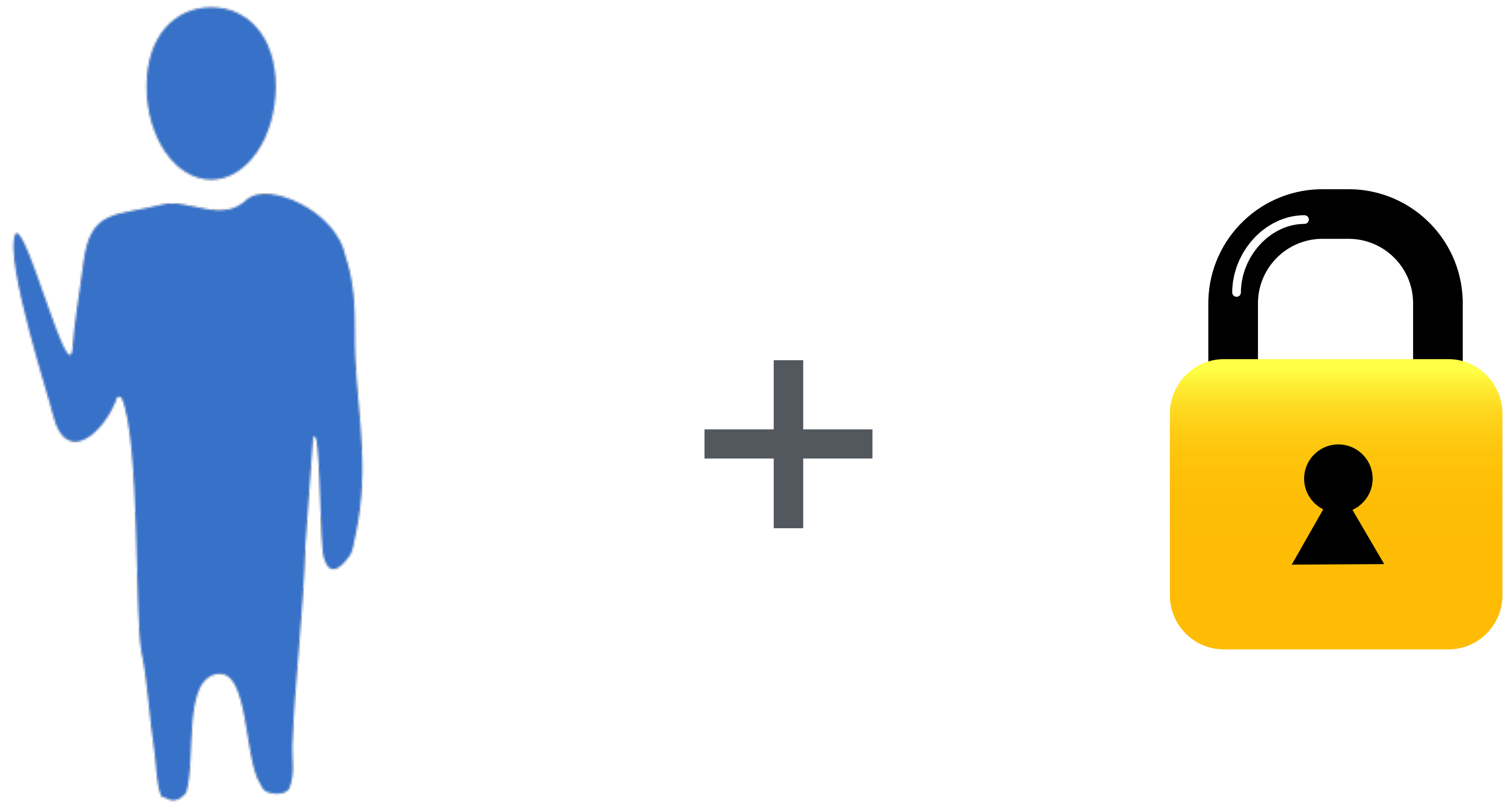
What Do You See?

- Which attributes do you see in the empty template?
- Which lines are not easy to understand?

- Fill in the template and click on [Submit]
- Write down the **nic-hdl** and the **mntner**

What You Are Seeing

- Congratulations! You just created your first objects in the RIPE (TEST) Database!
- You now have your own **person** object and your own personal **maintainer**



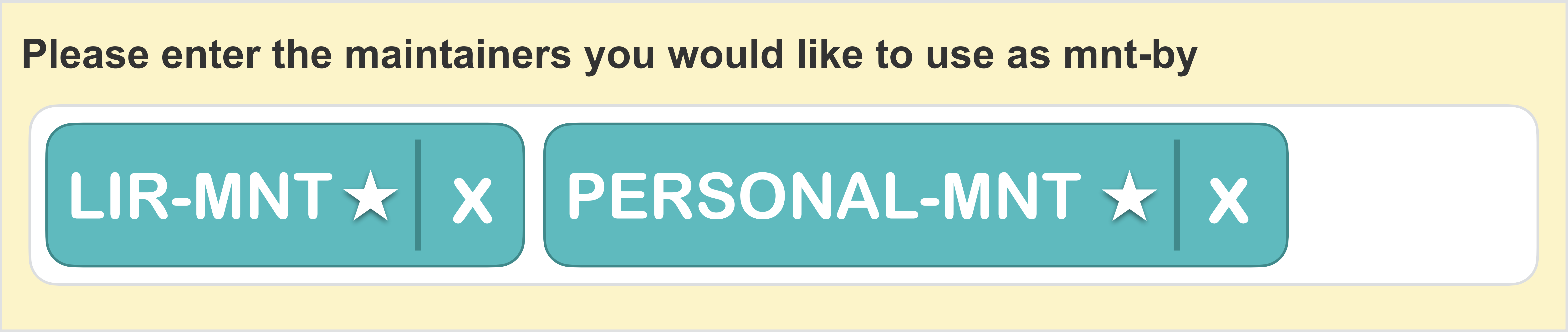
Creating a ROLE Object

It’s a **good habit** to use a **role** for the admin-c and tech-c attributes of LIR objects

1. Go to http://apps-test.db.ripe.net
2. On the left side, click on “**Create an object**”
3. Choose ”**role**” and click on [**Create**]

3. Choose which maintainer will protect the new object

4. Click on the **X** to remove a maintainer



Please enter the maintainers you would like to use as mnt-by

LIR-MNT ★ | x

PERSONAL-MNT ★ | x

☆ = Associated with your Access account

5. Fill in the template with data

- Use your LIR maintainer (**SMXX-MNT**)
- Use **role**:   Tech Team
- Fill in an “**address:**” and “**email:**”
- Leave “**nic-hdl:**” as it is: AUTO-1

6. Click on the [**+**] button next to “**email**”

   - Choose “**admin-c**” from the drop-down list
   - Click on **[Add]**
   - You now have an empty “admin-c:” attribute

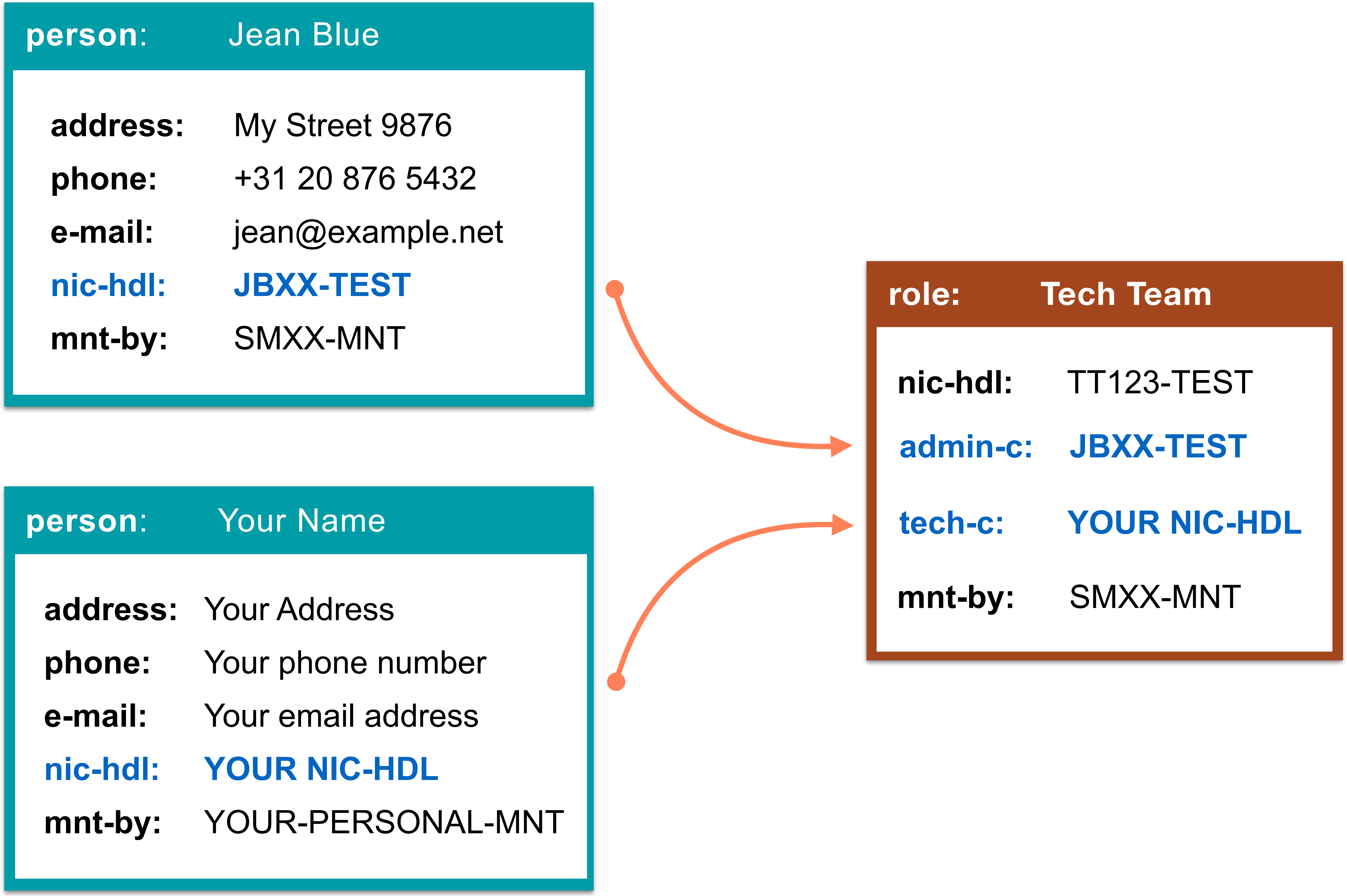
7. Do the same steps in 6) and add a “**tech-c:**”

8. Fill in the admin-c and tech-c with data
   - **admin-c**: JBXX-TEST
   - **tech-c**: YOUR PERSON OBJECT

9. Click on the **[Submit]** button

- If all was correctly filled in, you have a **role** object!
- Write down the **nic-hdl** of the object

What You Just Did



person: Jean Blue

address: My Street 9876
phone: +31 20 876 5432
e-mail: jean@example.net
nic-hdl: JBXX-TEST
mnt-by: SMXX-MNT

person: Your Name

address: Your Address
phone: Your phone number
e-mail: Your email address
nic-hdl: YOUR NIC-HDL
mnt-by: YOUR-PERSONAL-MNT

role: Tech Team

nic-hdl: TT123-TEST
admin-c: JBXX-TEST
tech-c: YOUR NIC-HDL
mnt-by: SMXX-MNT

Questions

How To Update It?

Updating the RIPE Database
Part 2

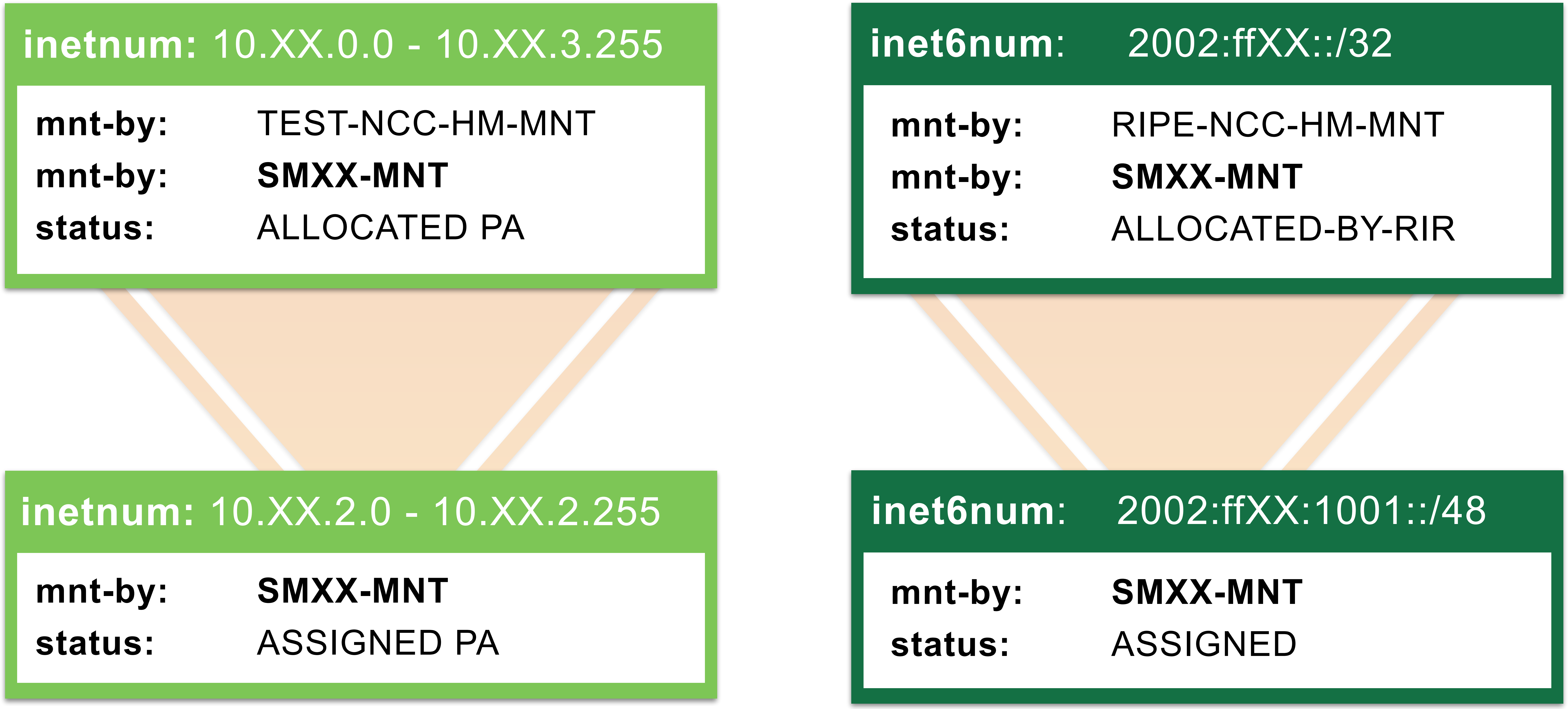
Registering IPv4 and IPv6

1. Let’s go back to the email **5**
   - from your colleague Jean Blue
2. Go to http://apps-test.db.ripe.net
3. On the left side, click on “**Create an object**”
4. Choose ”inetnum” or “inet6num”
5. Click on [**Create**]

What Do You See?

- Which attributes do you see in the template?
- Notice the first line (**mnt-by:**)
- How many maintainers appear here?
- Which lines are not easy to understand?

Registering Assignments



inetnum: 10.XX.0.0 - 10.XX.3.255

mnt-by: TEST-NCC-HM-MNT
mnt-by: **SMXX-MNT**
status: ALLOCATED PA

inetnum: 10.XX.2.0 - 10.XX.2.255

mnt-by: **SMXX-MNT**
status: ASSIGNED PA

inet6num: 2002:ffXX::/32

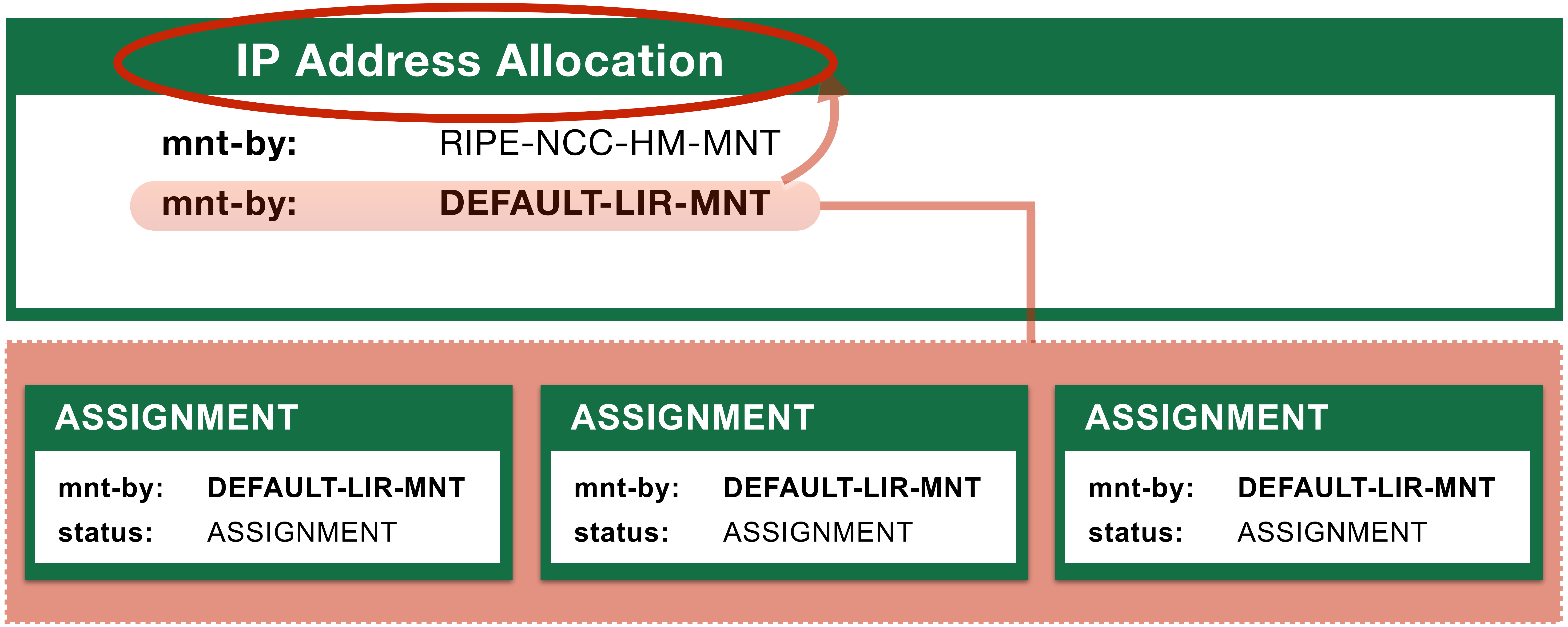
mnt-by: RIPE-NCC-HM-MNT
mnt-by: **SMXX-MNT**
status: ALLOCATED-BY-RIR

inet6num: 2002:ffXX:1001::/48

mnt-by: **SMXX-MNT**
status: ASSIGNED

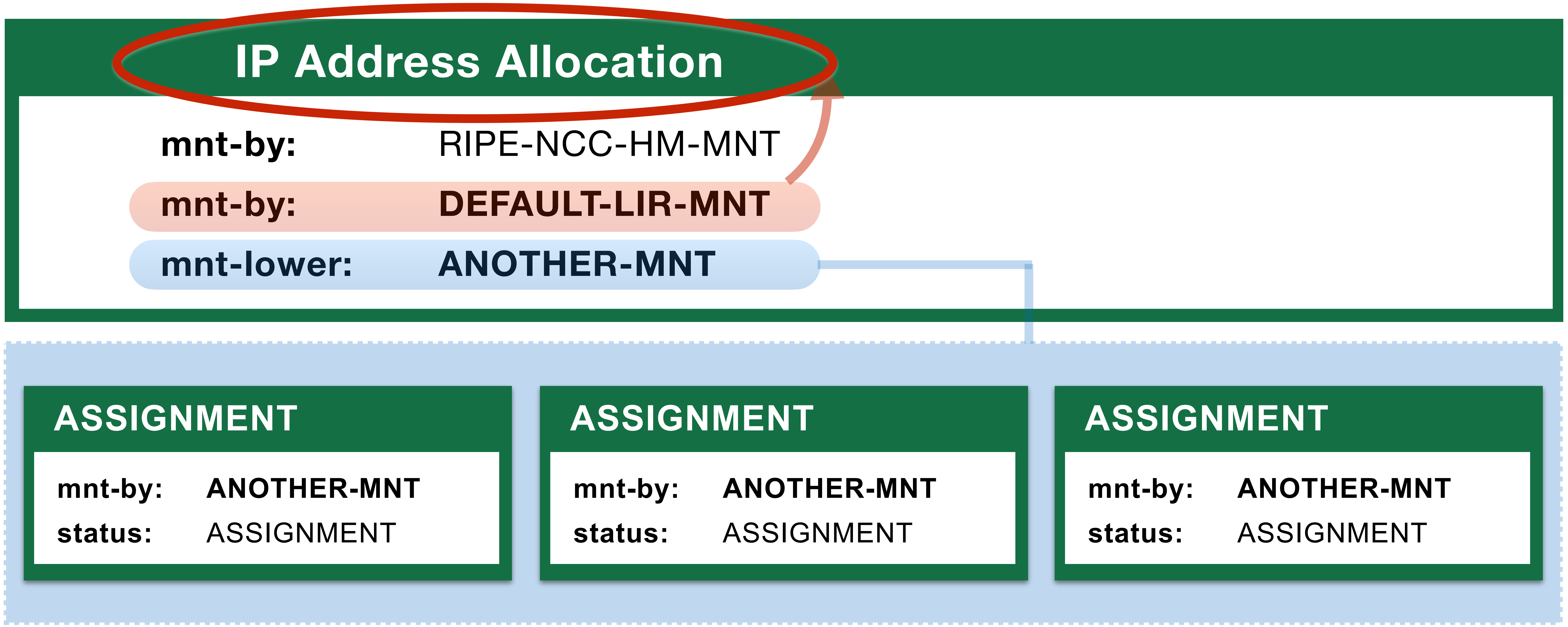
Registering Assignments

- To create an assignment, you must have authorisation from the allocation
- Here, “**mnt-by:**” has control over the allocation object and the space under the object



Registering Assignments

- If “**mnt-lower:**” is present, then it has permission to create objects in the space under the object
  - but it cannot update the allocation! (**mnt-by:**)



Registering Assignments

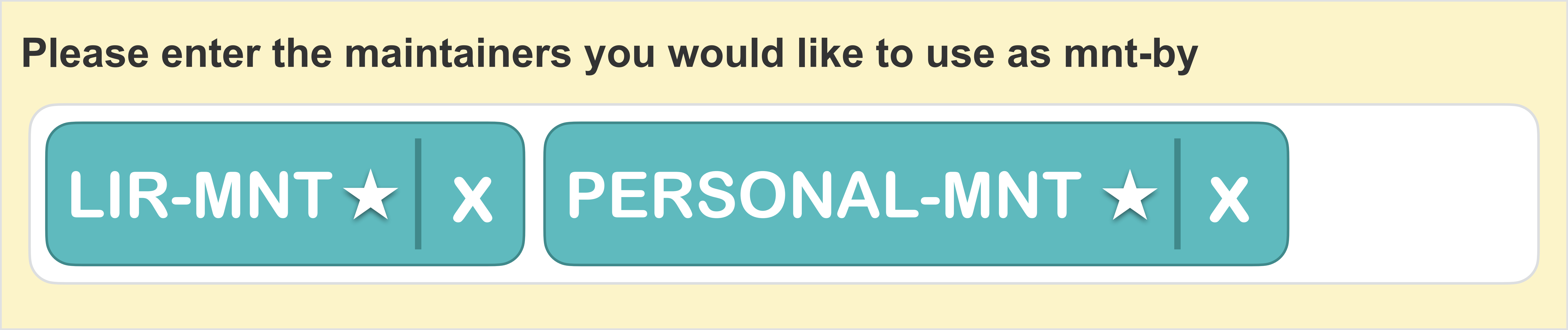
- You can group multiple assignments in one object
- Use "**status: AGGREGATED-BY-LIR**"
- "**assignment-size:**" attribute is
  - optional in IPv4
  - mandatory in IPv6

```
inetnum: 10.XX.0.0 - 10.XX.0.127

mnt-by:          SMXX-MNT
status:          AGGREGATED-BY-LIR
assignment-size: 30
```

```
inet6num:    2002:ffXX:1000::/36

mnt-by:          SMXX-MNT
status:          AGGREGATED-BY-LIR
assignment-size: 56
```

Filling In The Template

- Choose which maintainer will protect the new object
- Click on the **X** to remove a maintainer



Please enter the maintainers you would like to use as mnt-by

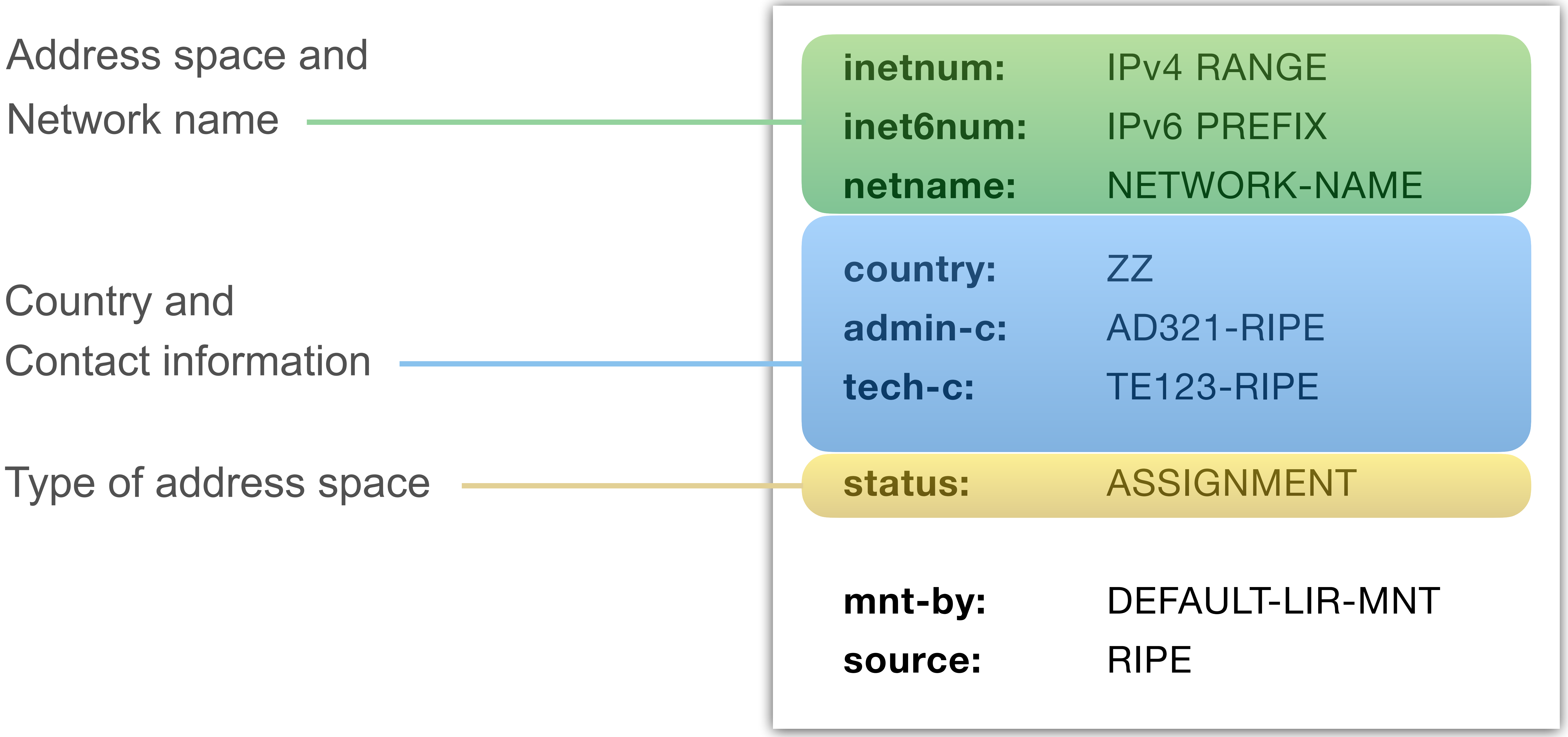
LIR-MNT ★ x

PERSONAL-MNT ★ x

☆ = Associated with your Access account

Filling In The Template

Same object structure for **IPv4** and **IPv6**



Address space and Network name

Country and Contact information

Type of address space

```
inetnum:        IPv4 RANGE
inet6num:       IPv6 PREFIX
netname:        NETWORK-NAME

country:        ZZ
admin-c:        AD321-RIPE
tech-c:         TE123-RIPE

status:         ASSIGNMENT

mnt-by:         DEFAULT-LIR-MNT
source:         RIPE
```

Object Creation Success

If the values in the object template are correct, then the RIPE Database will create the object

```
inetnum:   10.30.2.0 - 10.30.2.255
netname:   LAIKA-NET-01
country:   ZZ
admin-c:   MB54321-TEST
tech-c:    ROLE-NIC-HDL
status:    ASSIGNED PA
mnt-by:    SMXX-MNT
```

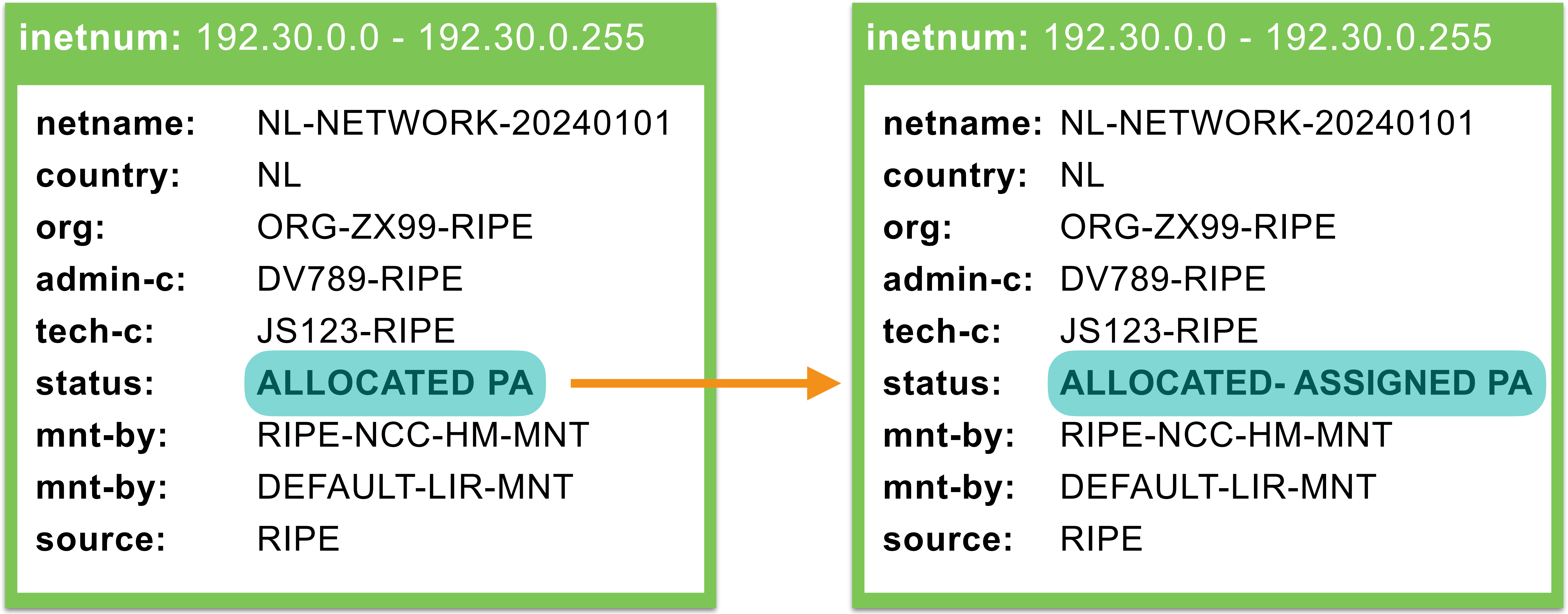
✓

```
inet6num:  2002:ff30:1001::/48
netname:   LAIKA-NET-01
country:   ZZ
admin-c:   MB54321-TEST
tech-c:    ROLE-NIC-HDL
status:    ASSIGNED
mnt-by:    SMXX-MNT
```

✓

status: ALLOCATED-ASSIGNED PA

- Registers a whole assignment for small IPv4 allocations
- LIR changes status in the RIPE Database
- It is seen both as ALLOCATED PA and ASSIGNED PA



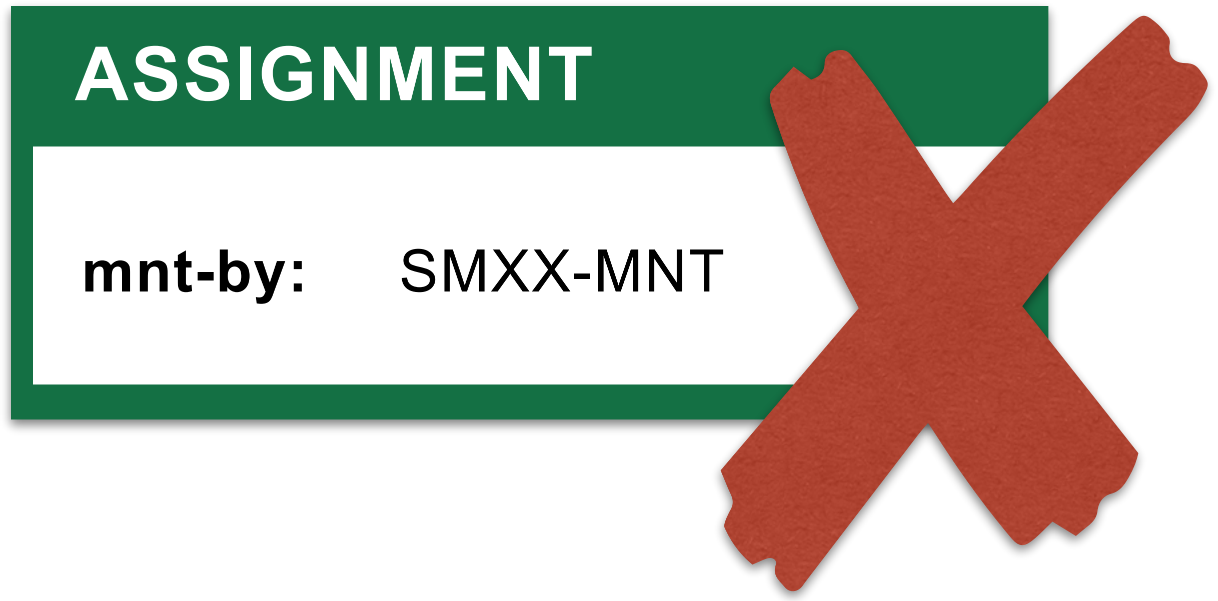
```
inetnum: 192.30.0.0 - 192.30.0.255

netname:   NL-NETWORK-20240101
country:   NL
org:       ORG-ZX99-RIPE
admin-c:   DV789-RIPE
tech-c:    JS123-RIPE
status:    ALLOCATED PA
mnt-by:    RIPE-NCC-HM-MNT
mnt-by:    DEFAULT-LIR-MNT
source:    RIPE
```

```
inetnum: 192.30.0.0 - 192.30.0.255

netname:  NL-NETWORK-20240101
country:  NL
org:      ORG-ZX99-RIPE
admin-c:  DV789-RIPE
tech-c:   JS123-RIPE
status:   ALLOCATED- ASSIGNED PA
mnt-by:   RIPE-NCC-HM-MNT
mnt-by:   DEFAULT-LIR-MNT
source:   RIPE
```

Deleting Objects

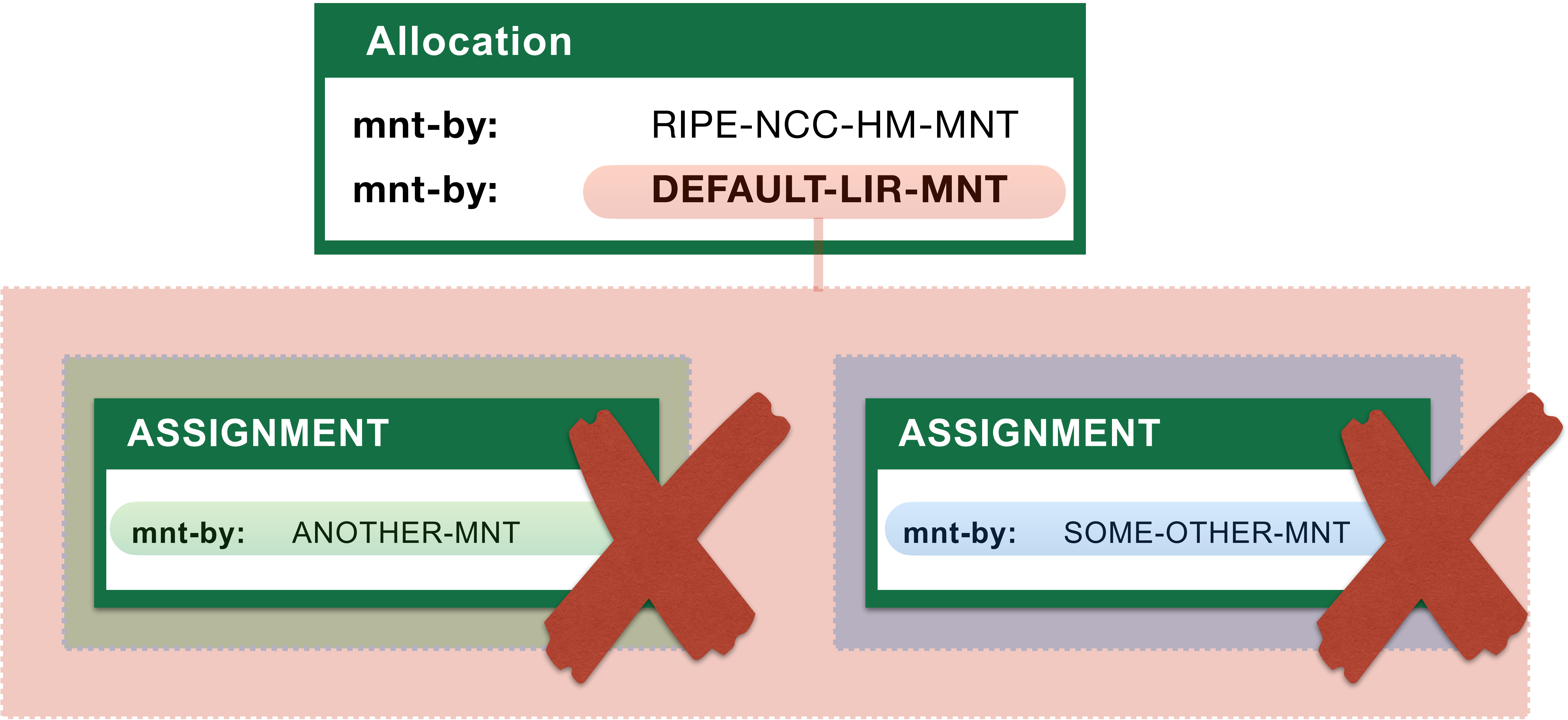
1. Let’s go back to the email **5**
   - from your colleague Jean Blue
2. Go to http://apps-test.db.ripe.net
3. Search for all the assignments:
   - i.e. **-m** 10.XX.0.0 - 10.XX.3.255
   - i.e. **-m** 2002:ffXX::/32

4. You should see Jean Blue’s assignments and your newly registered assignments
5. Look for the **wrong** objects in the results
6. Click on **[Update object]**
7. Click on the **[Delete this object]** button
8. Provide a “reason” and click on **[Confirm delete]**



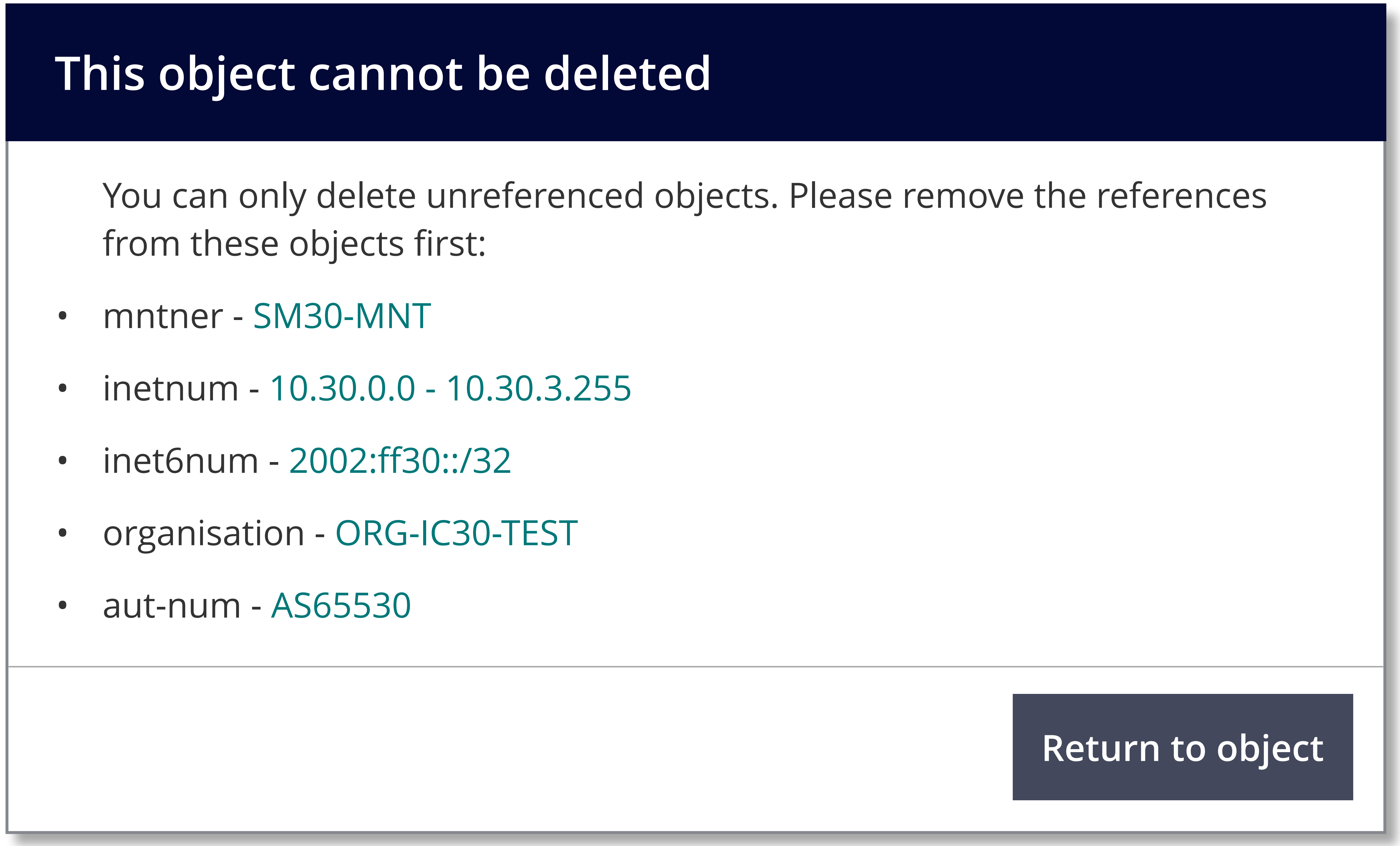
LIR Keeps Control

- LIR Default Maintainer has control over the whole address space
- Use “**Force Delete**” to remove lost objects



When You Cannot Delete

- If an object is referenced in another object, you must first remove the reference



This object cannot be deleted

You can only delete unreferenced objects. Please remove the references from these objects first:

- mntner - SM30-MNT
- inetnum - 10.30.0.0 - 10.30.3.255
- inet6num - 2002:ff30::/32
- organisation - ORG-IC30-TEST
- aut-num - AS65530

Summary

- You have now updated the RIPE Database:
  - Created your own **person**/**maintainer** pair
  - Created a **role** object for the LIR
  - Registered assignments by creating **inet(6)num** objects
  - Deleted the wrong **inet(6)num** objects

✓

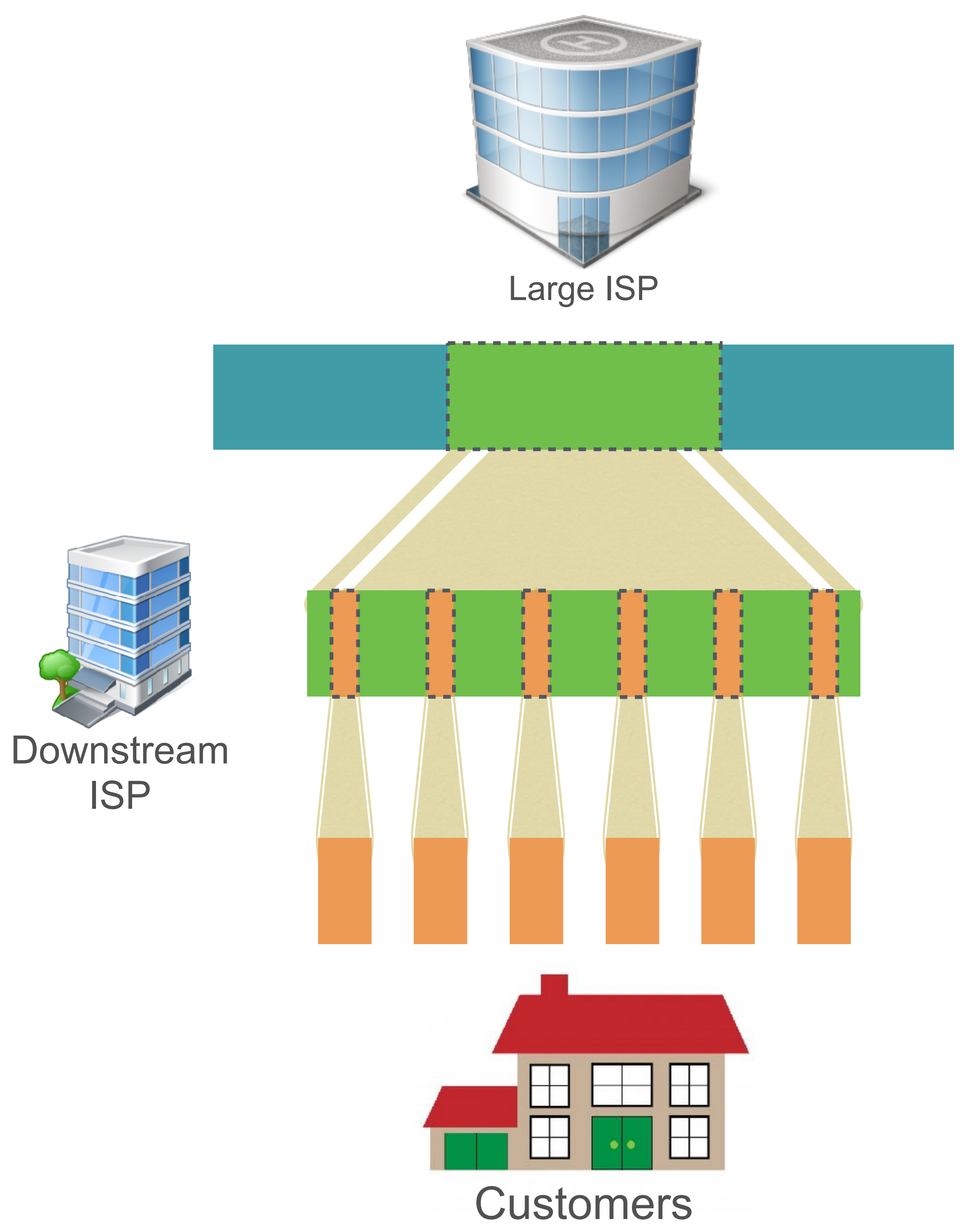
Questions

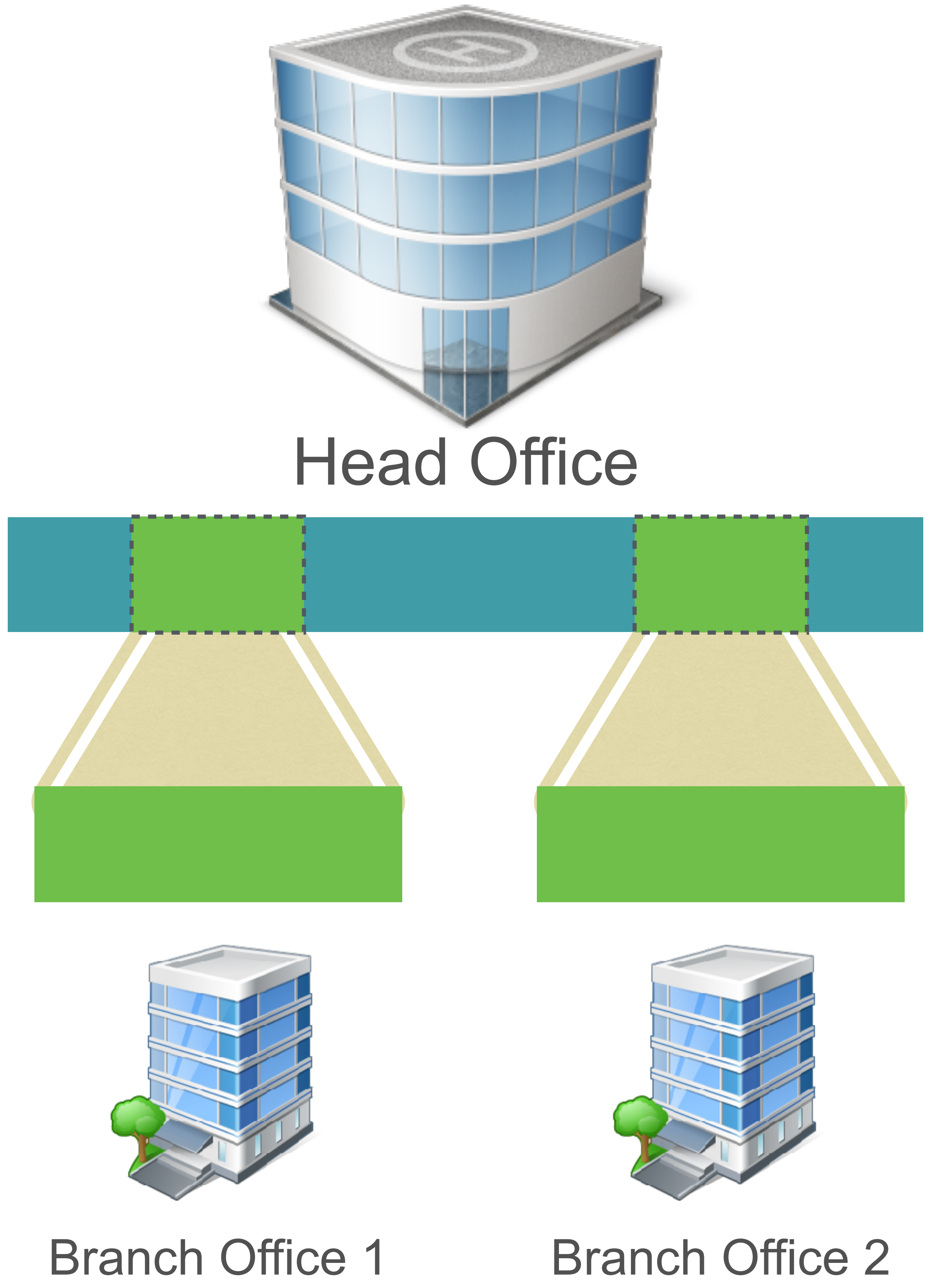
Delegating To Others

Giving control to someone else

Sub-Allocations

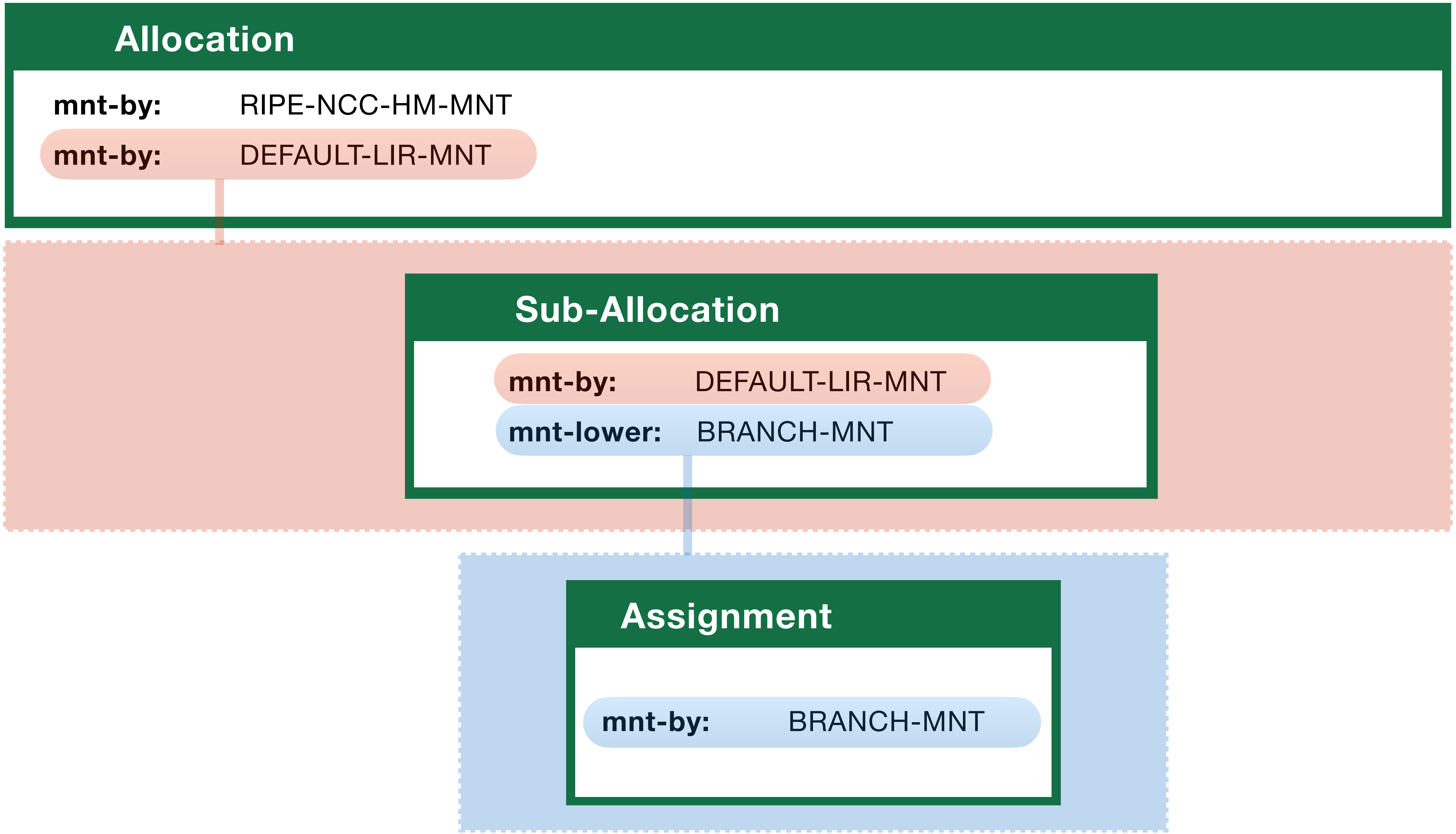
- Block for a downstream customer
- Branch office or department





Delegating Control

- “**mnt-lower:**” attribute gives permission to create more specific objects



Registering Sub-Allocations

Use the appropriate “**status:**”

IPv4 = SUB-ALLOCATED PA

IPv6 = ALLOCATED-BY-LIR

```
inetnum:    10.0.1.0 - 10.0.2.255

netname:    Branch-office-1
country:    NL
admin-c:    LA789-RIPE
tech-c:     LA789-RIPE
status:     SUB-ALLOCATED PA
mnt-by:     LIR-MNT
mnt-lower:  BRANCH-MNT
```

```
inet6num:   2002:ff00:a000::/36

netname:    Branch-office-1
country:    NL
admin-c:    LA789-RIPE
tech-c:     LA789-RIPE
status:     ALLOCATED-BY-LIR
mnt-by:     LIR-MNT
mnt-lower:  BRANCH-MNT
```

Register a IPv6 Sub-Allocation

1. Go to http://apps-test.db.ripe.net

2. On the left side, click on “**Create an object**”

3. Choose “**inet6num**” and click on [Create]

4. Fill in the template:

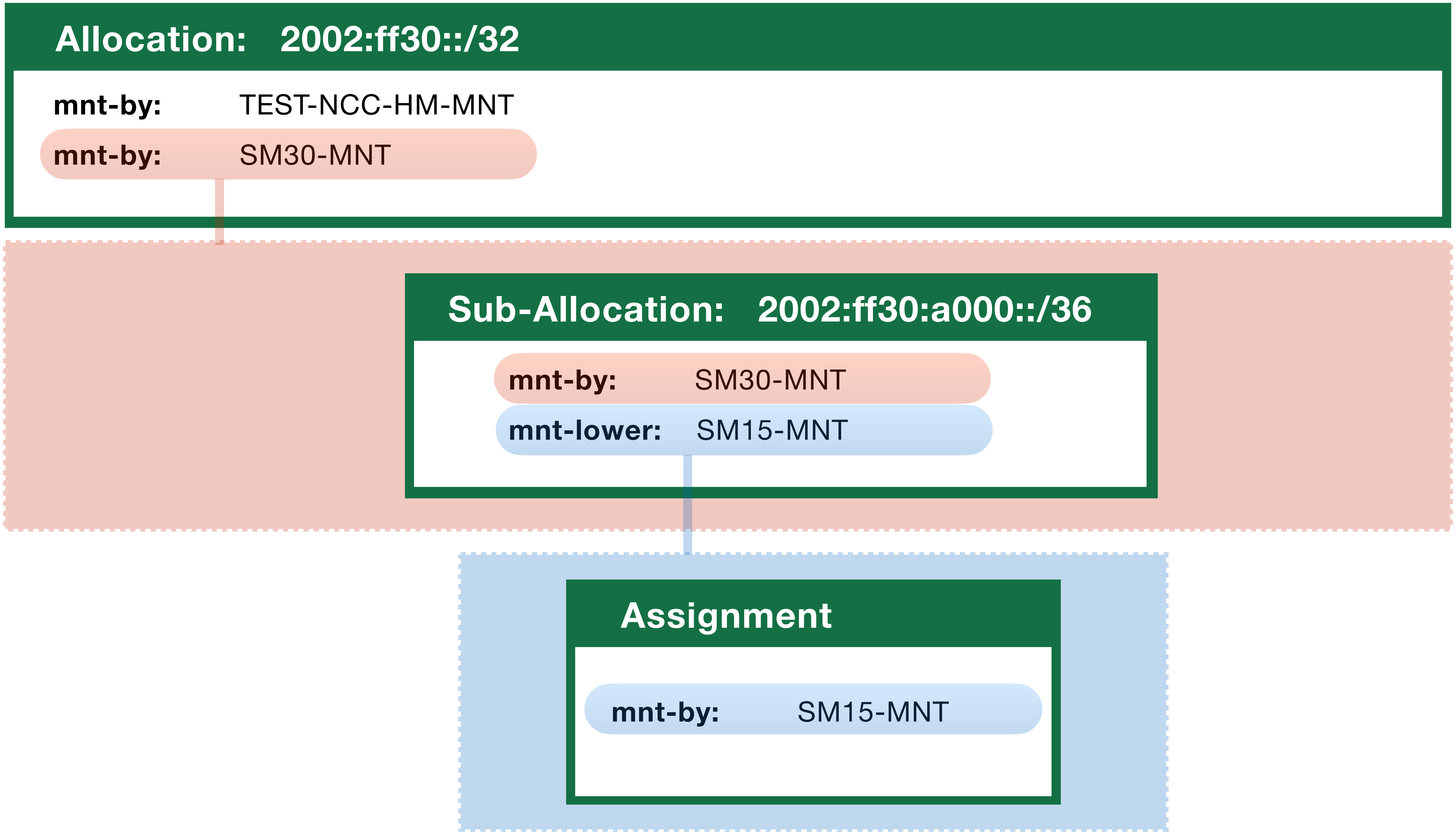
- inet6num: 2002:ffXX:a000::/36
- netname: SUBALLOCATION
- country: your neighbor’s country
- Use your **person** object as “admin-c:”
- Use your neighbor’s **person** object as “tech-c:”

5. Add a “**mnt-lower:**” attribute
    - Use **your neighbor’s maintainer** as value

6. Choose the status **ALLOCATED-BY-LIR**

7. Click on **[Submit]**

What You Want To Do

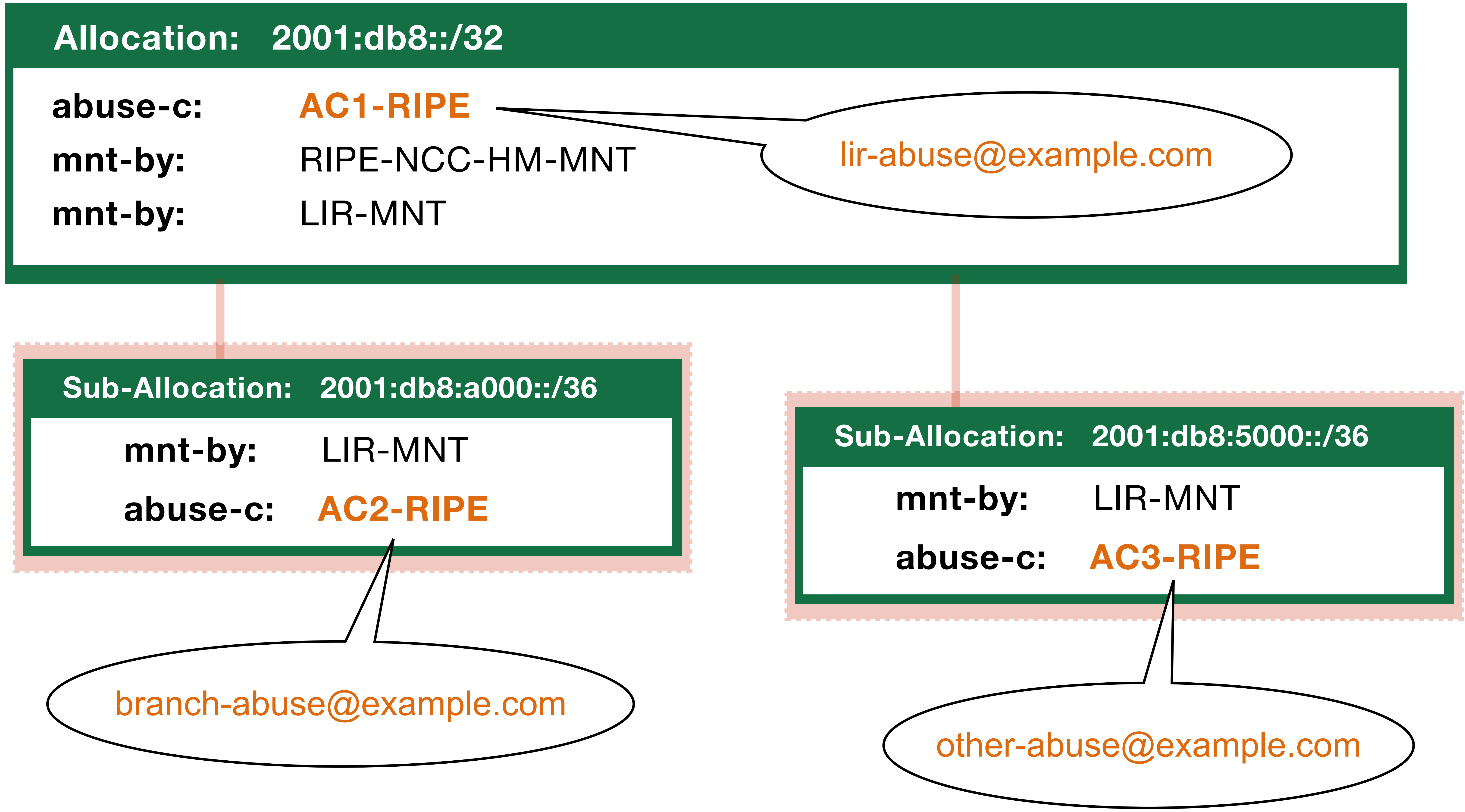


Create an Assignment

1. Go to http://apps-test.db.ripe.net
2. On the left side, click on “**Create an object**”
3. Choose ”**inet6num**” and click on [Create]
4. Fill in the template:
    1. inet6num: 2002:ffzz:a000::/48
    2. zz = number of your neighbor
    3. status: **ASSIGNED**
5. You know how to do the rest! ;-)

Separate Abuse Contact

- Sub-allocations can have a separate “**abuse-c:**”



```
Allocation:   2001:db8::/32
abuse-c:      AC1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-by:       LIR-MNT
```

lir-abuse@example.com

```
Sub-Allocation:   2001:db8:a000::/36
mnt-by:     LIR-MNT
abuse-c:    AC2-RIPE
```

branch-abuse@example.com

```
Sub-Allocation:   2001:db8:5000::/36
mnt-by:     LIR-MNT
abuse-c:    AC3-RIPE
```

other-abuse@example.com

Questions ?

RIPE Routing Registry

aut-num, route and route6
objects

Search For Your AUT-NUM Object

1. Read the email **6**
2. Go to http://apps-test.db.ripe.net
3. Search for **AS655XX**

What Do You See?

- What does this object represent?
- Which attributes call your attention?

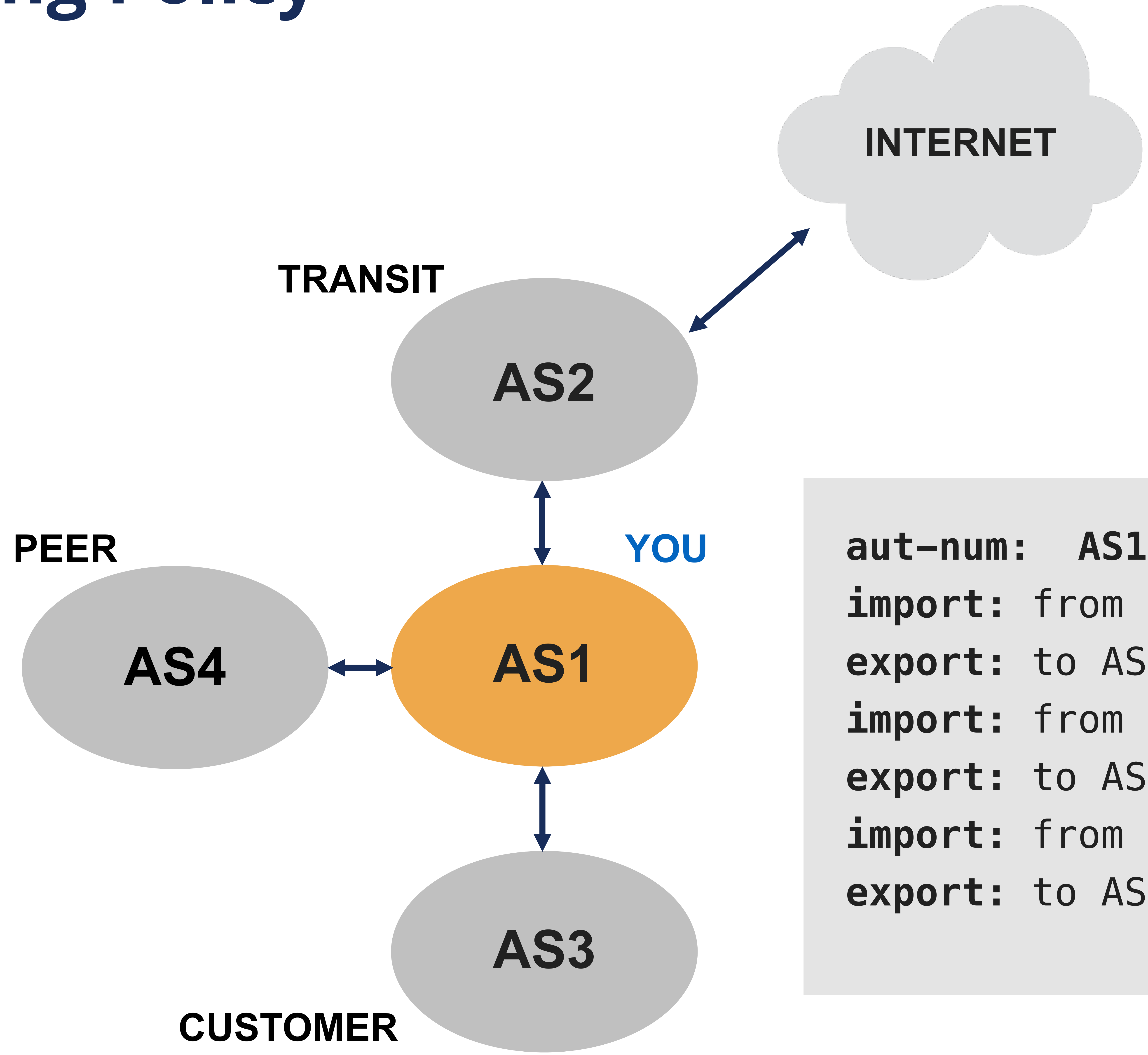
Autonomous System Number Objects

- Known as **aut-num** objects

- Register **who** holds an AS Number and the routing policy for that AS

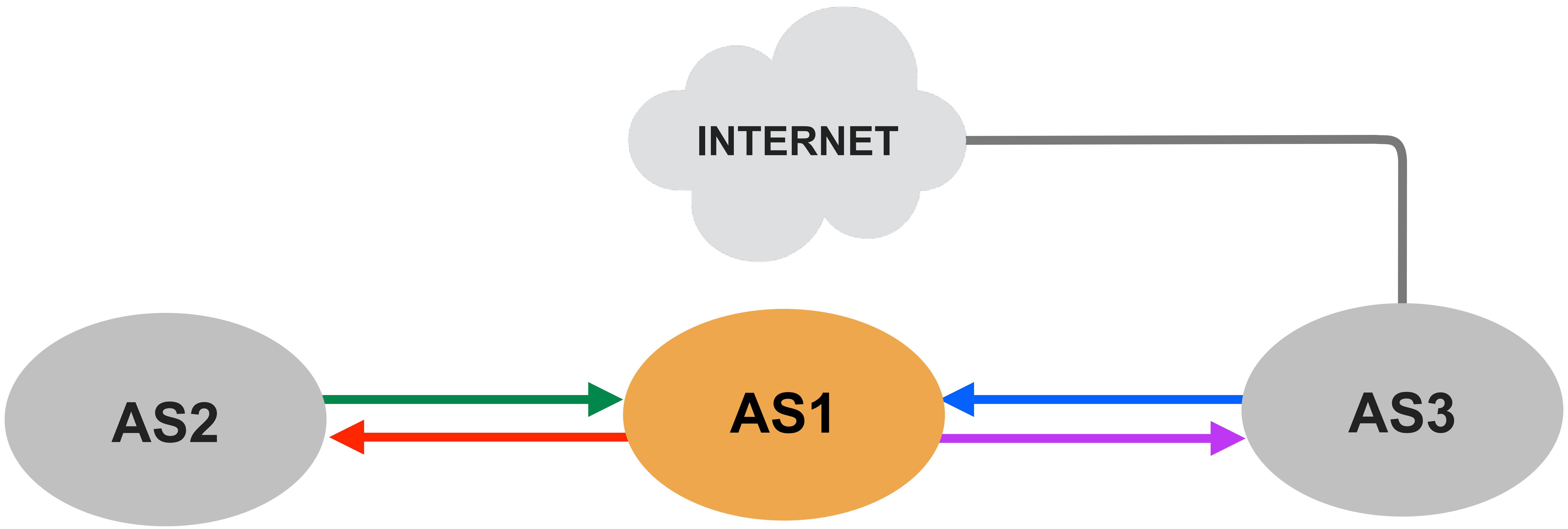
```
aut-num:     AS12345

as-name:     YOUR-AS-NAME
org:         ORG-EE2-RIPE
import:      from AS1010 accept ANY
export:      to AS1010 announce AS12345
import:      from AS987 accept ANY
export:      to AS987 announce AS12345
admin-c:     DV789-RIPE
tech-c:      JS123-RIPE
status:      ASSIGNED
mnt-by:      RIPE-NCC-END-MNT
mnt-by:      DEFAULT-LIR-MNT
source:      RIPE
```

Routing Policy



```
aut-num:  AS1
import: from AS2 accept ANY
export: to AS2 announce AS1 AS3
import: from AS3 accept AS3
export: to AS3 announce ANY
import: from AS4 accept AS4
export: to AS4 announce AS1 AS3
```

Building An AUT-NUM Object



```
aut-num: AS2

import: from AS1 accept AS1

export: to AS1 announce AS2
```

```
aut-num: AS1

export: to AS2 announce AS1

import: from AS2  accept AS2

import: from AS3 accept ANY

export: to AS3 announce AS1
```

```
aut-num: AS3



export: to AS1 announce ANY

import: from AS1 accept AS1
```

Search For ROUTE(6) Objects

1. Read the email **6**

2. Go to http://apps-test.db.ripe.net

3. Search for the **route(6)** objects

   - Use the “-T” flag to show the route(6) objects
   - i.e. -T route 10.xx.0.0/22
   - i.e. -T route6 2002:ffxx::/32

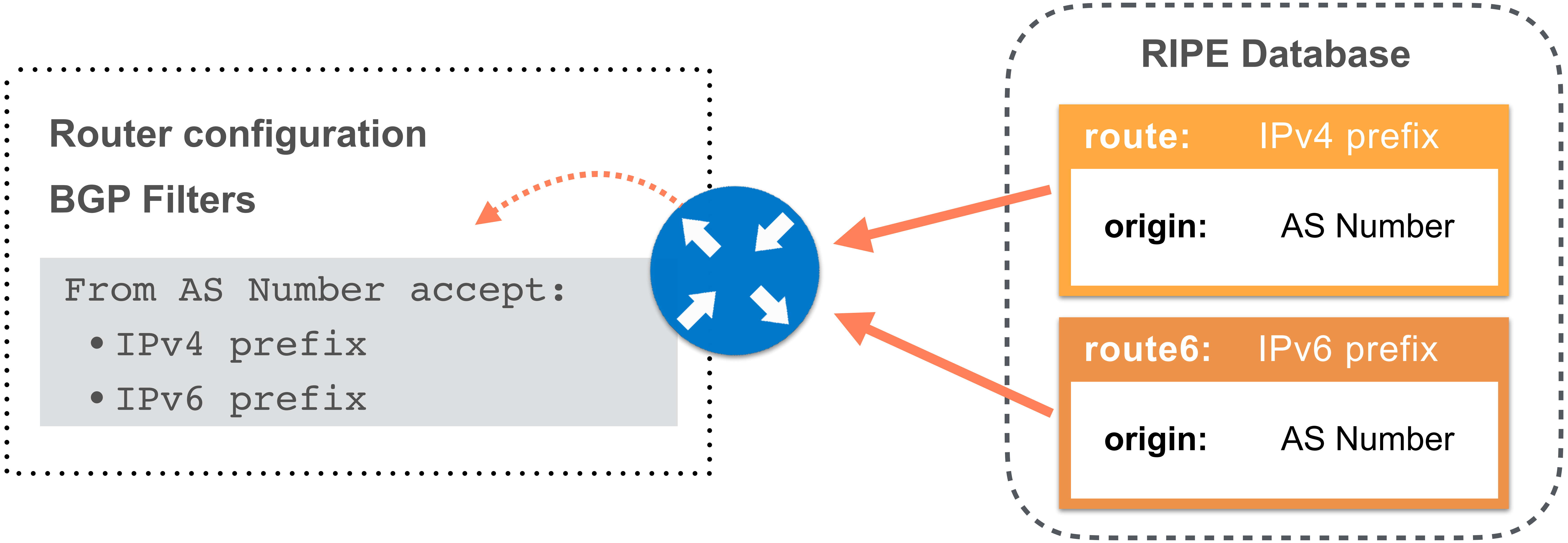
What Do You See?

- Did you get any objects in the results?

- No? Then there are no route(6) objects yet!

What Are ROUTE(6) Objects?

- **route(6)** objects register which IPv4/IPv6 prefix will be announced by which AS number
- Used for creating BGP filters

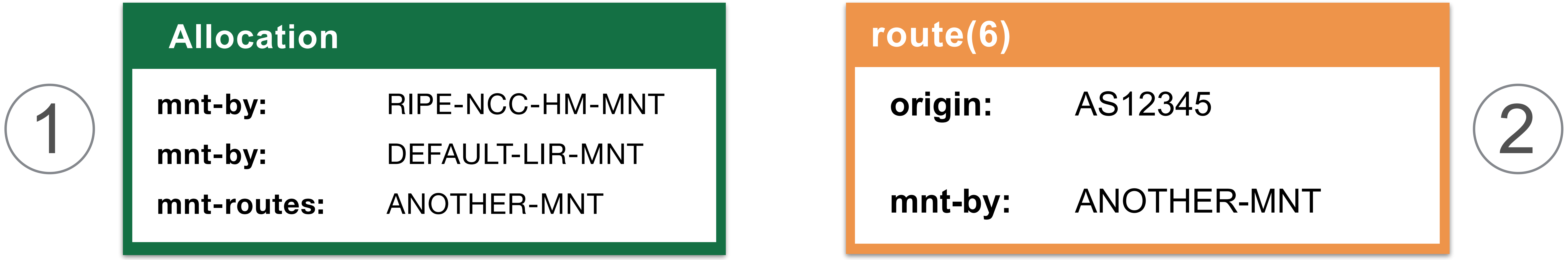


```
From AS Number accept:
 • IPv4 prefix
 • IPv6 prefix
```

How To Create ROUTE(6) Objects

- You need permission from:

1. **inetnum** or **inet6num**
2. **route** or **route6**



1

Allocation

mnt-by: RIPE-NCC-HM-MNT
mnt-by: DEFAULT-LIR-MNT
mnt-routes: ANOTHER-MNT

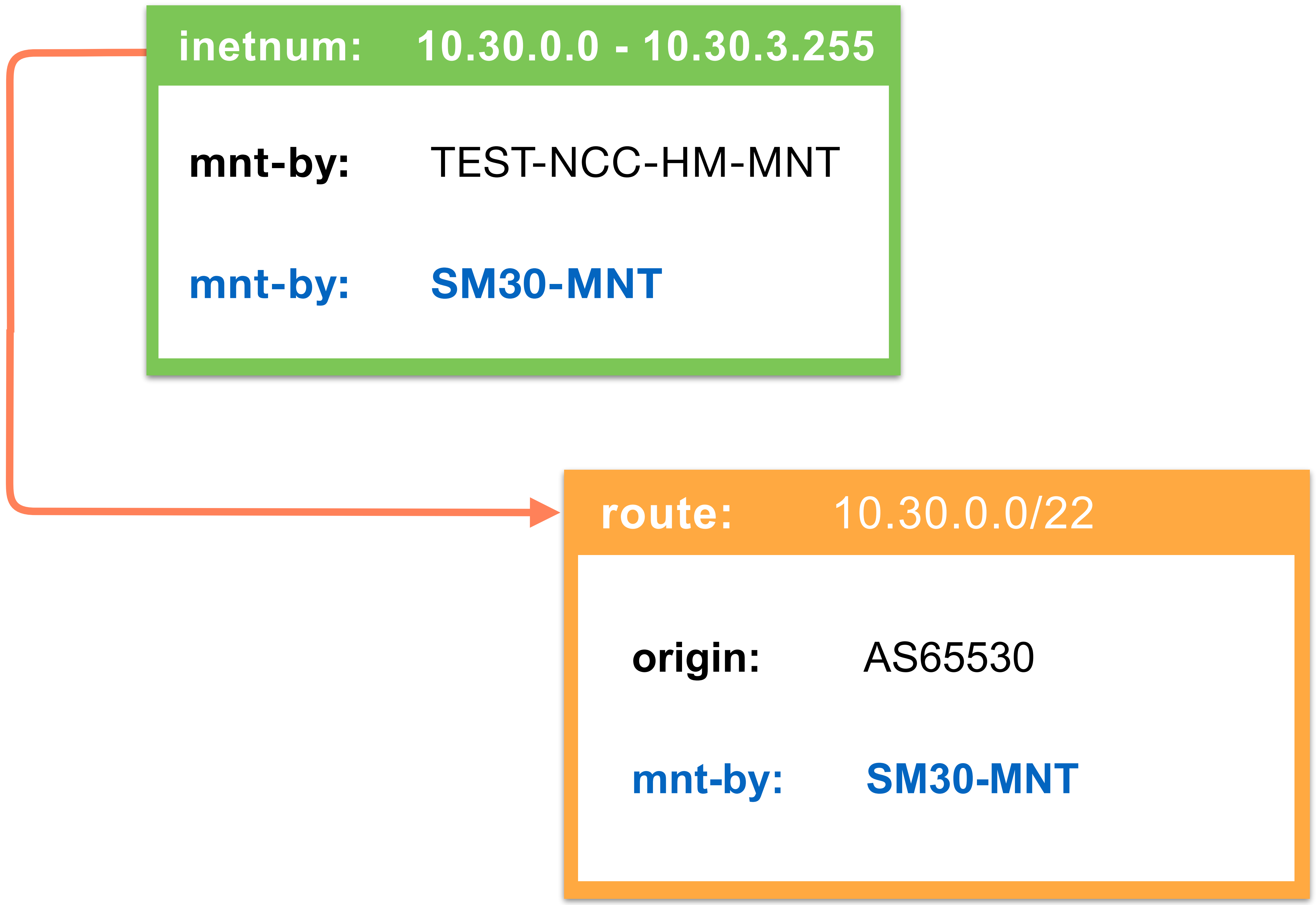
route(6)

origin: AS12345
mnt-by: ANOTHER-MNT

2

* **mnt-routes** delegates the creation of route(6) objects

Registering IPv4 Routes



inetnum: 10.30.0.0 - 10.30.3.255

mnt-by: TEST-NCC-HM-MNT

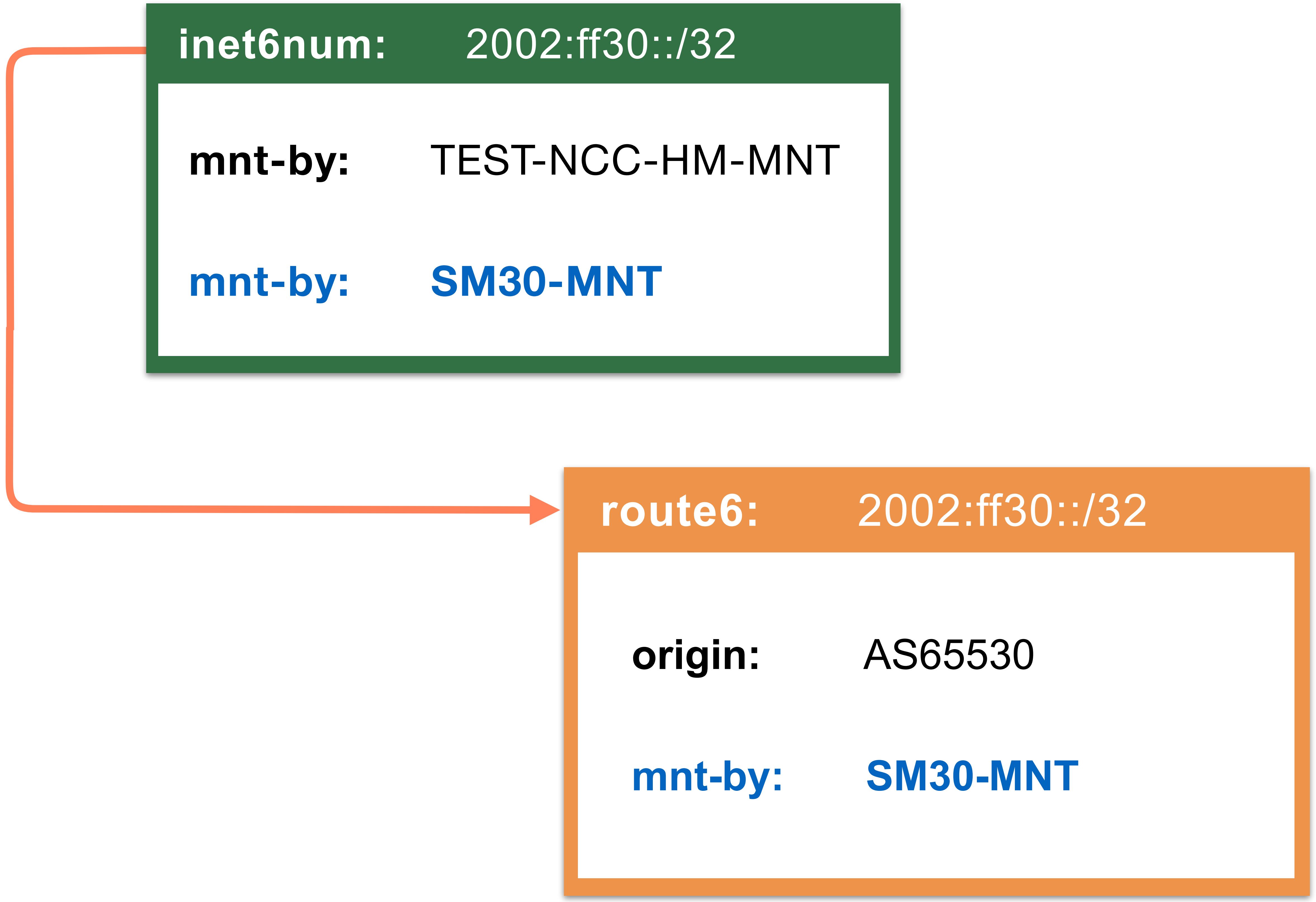
mnt-by: SM30-MNT

route: 10.30.0.0/22

origin: AS65530

mnt-by: SM30-MNT

Registering IPv6 Routes



inet6num: 2002:ff30::/32

mnt-by: TEST-NCC-HM-MNT

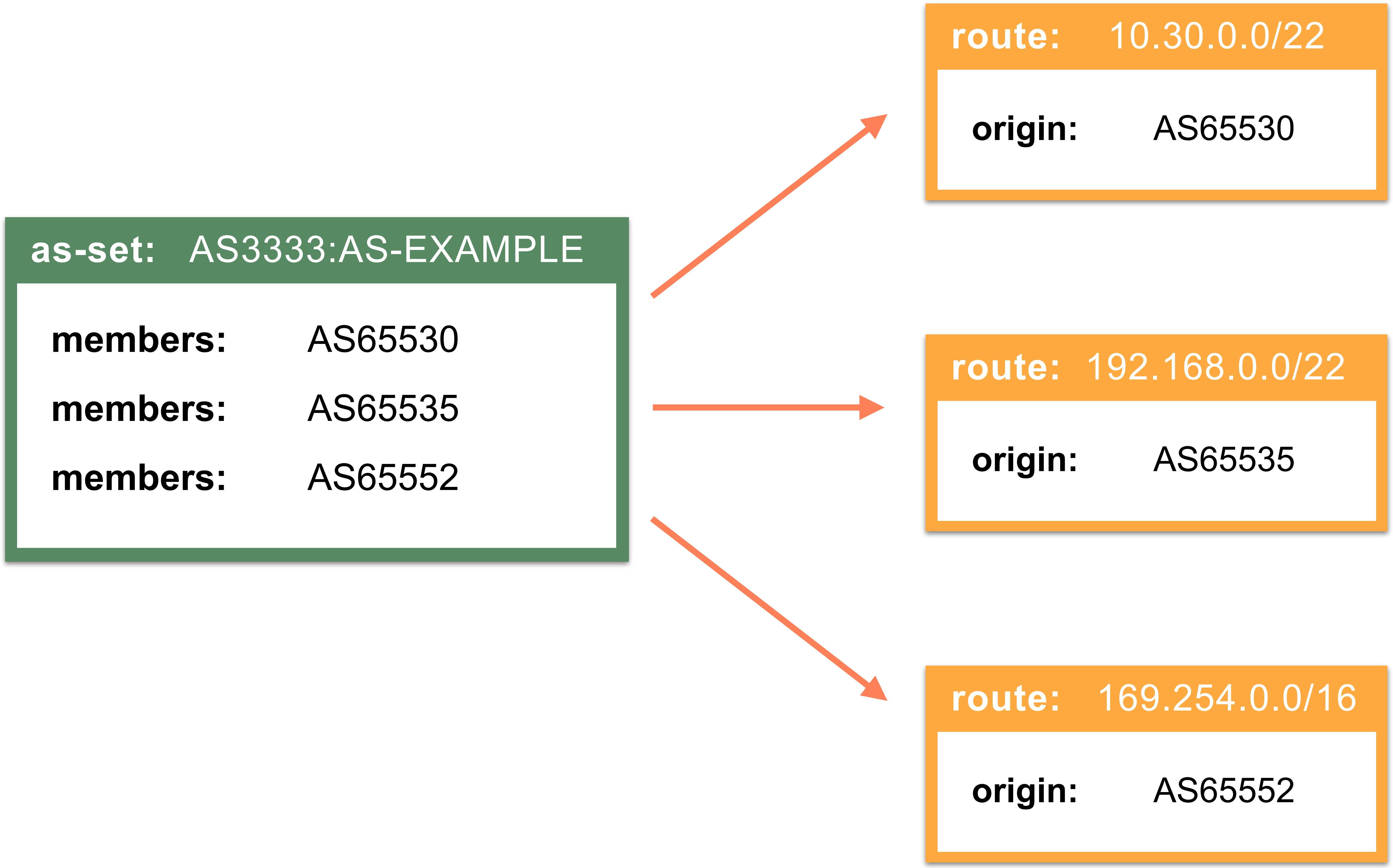
mnt-by: SM30-MNT

route6: 2002:ff30::/32

origin: AS65530

mnt-by: SM30-MNT

AS-SETs



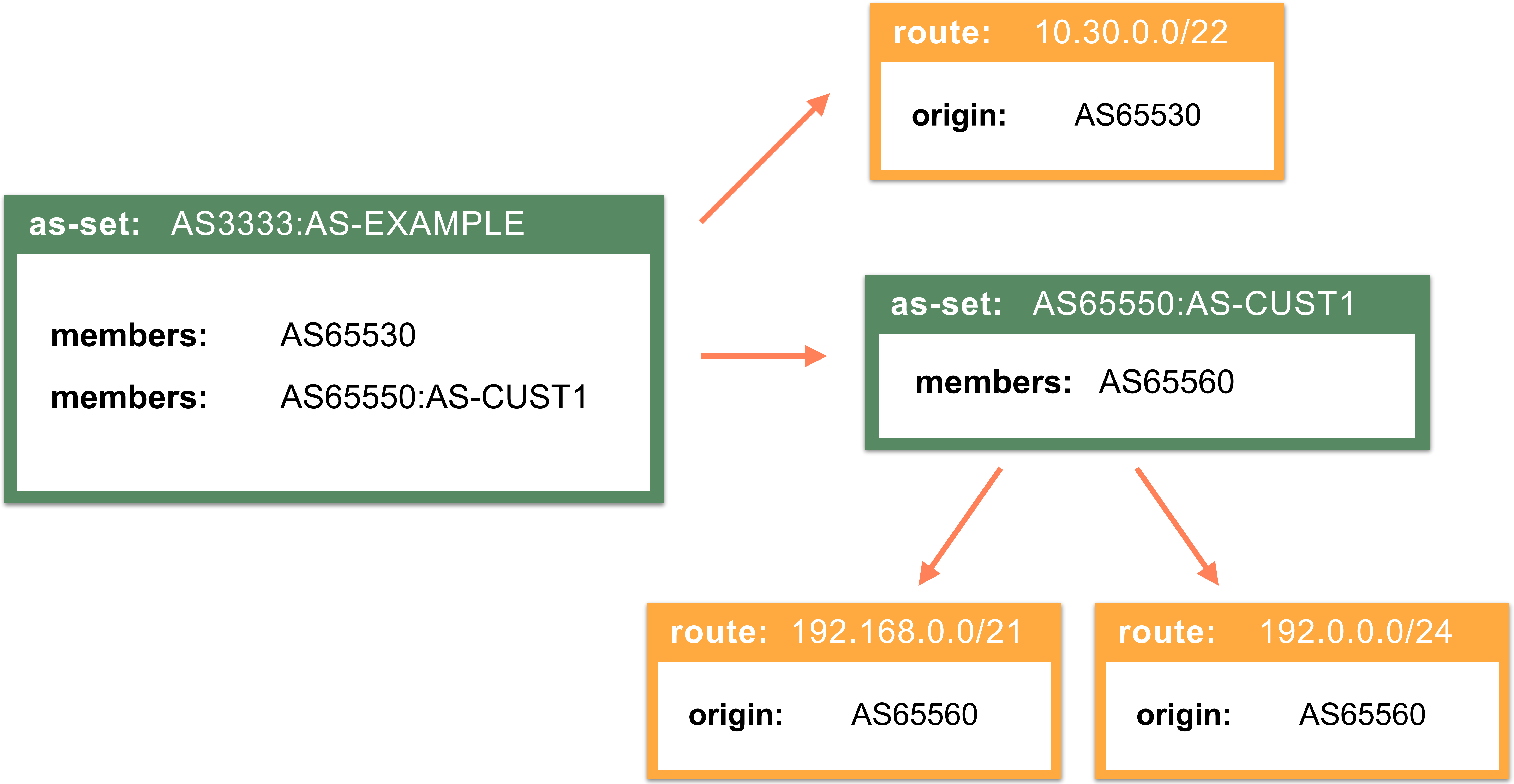
```
as-set:   AS3333:AS-EXAMPLE

members:     AS65530
members:     AS65535
members:     AS65552
```

```
route:   10.30.0.0/22

origin:      AS65530
```

```
route:  192.168.0.0/22

origin:      AS65535
```

```
route:   169.254.0.0/16

origin:      AS65552
```

AS-SETs



as-set: AS3333:AS-EXAMPLE

members: AS65530

members: AS65550:AS-CUST1

route: 10.30.0.0/22

origin: AS65530

as-set: AS65550:AS-CUST1

members: AS65560

route: 192.168.0.0/21

origin: AS65560

route: 192.0.0.0/24

origin: AS65560

Create ROUTE(6) Objects

1. Go to http://apps-test.db.ripe.net

2. On the left side, click on “**Create an object**”

3. Choose “**route**” or “**route6**” and click on [Create]

4. Fill in the template:
   - route: 10.XX.0.0/22
   - route6: 2002:ffXX::/32
   - origin: AS655XX

Questions

Reverse DNS

Setting up reverse delegation

Looking For DOMAIN Objects

1. Read the email **7**
2. Go to http://apps-test.db.ripe.net
3. Search for your IPv4 allocation
4. Use the flags “**-r -m -d**” in the query
   - “**-d**” flag includes domain objects in results
   - i.e. -r -m -d 10.XX.0.0/22

You can try this with your own real allocation!

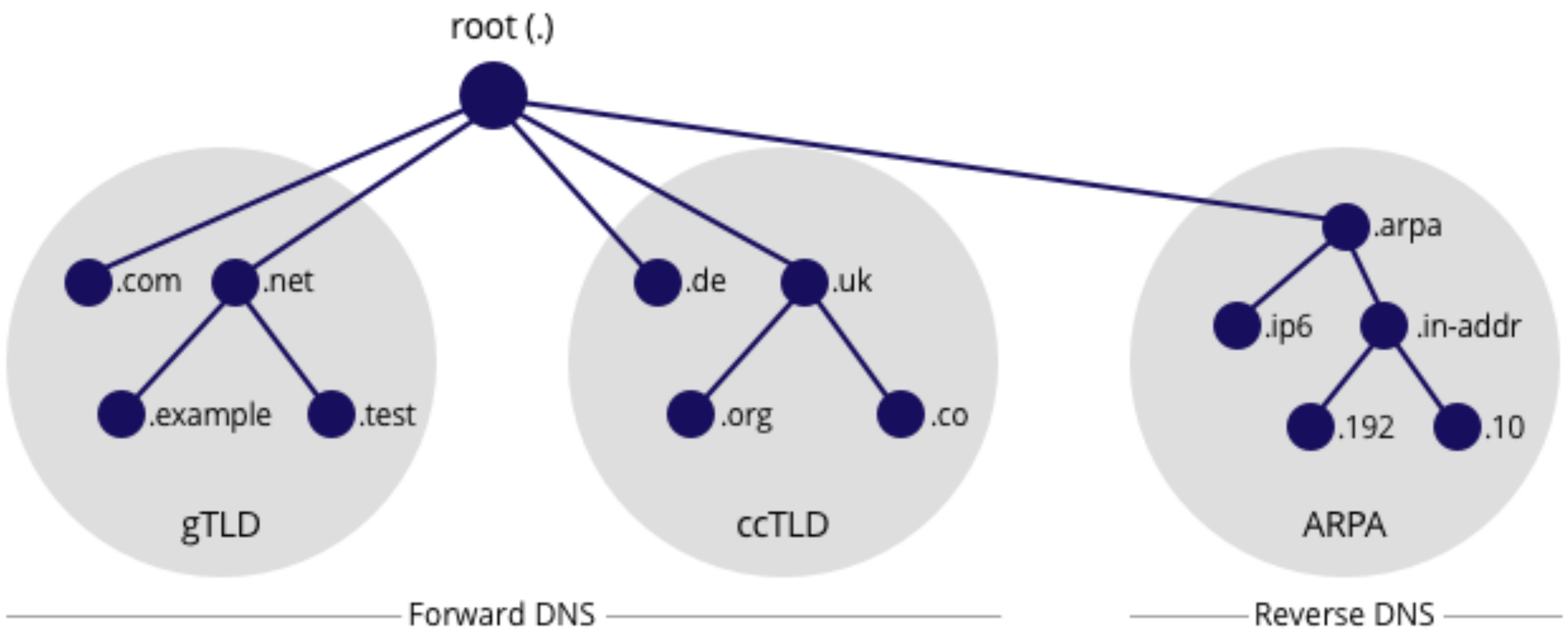
What Do You See?

- Do you see any **domain** objects in the results?

- No? Then Reverse Delegation is not set up yet!

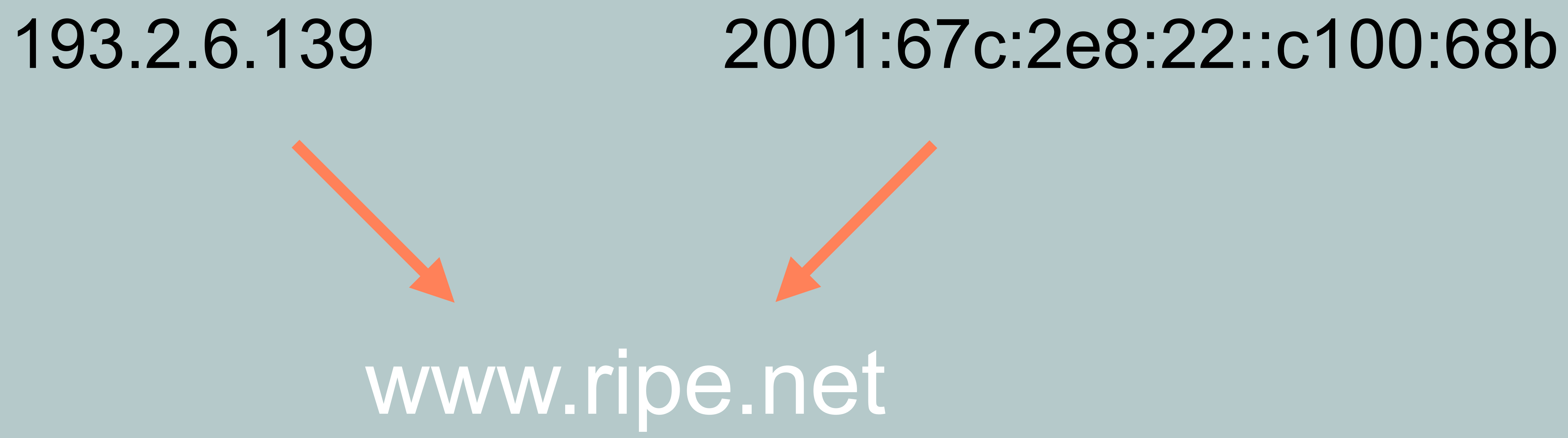
DNS Tree Structure

- At the top is the root (**.**)
- Then the ccTLDs and gTLDs
- Each domain/sub-domain is stored in a DNS zone



What is Reverse DNS ?

Mapping of IP addresses to host names



193.2.6.139

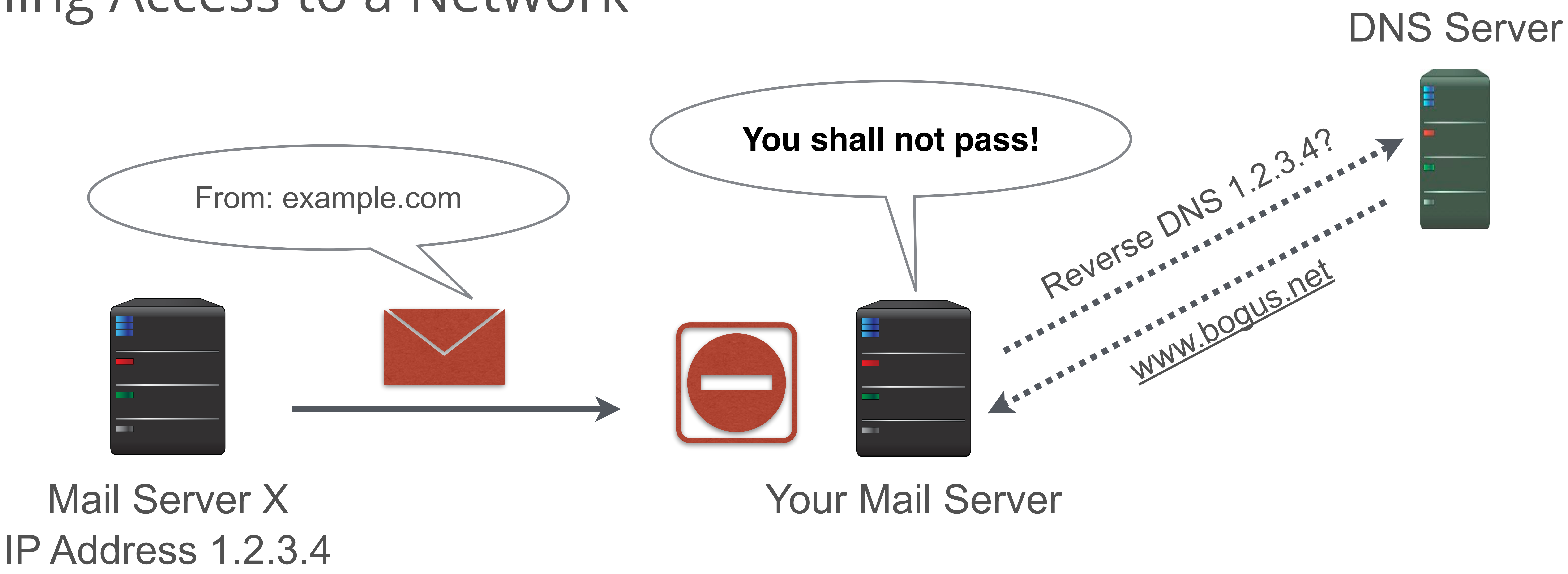
2001:67c:2e8:22::c100:68b

www.ripe.net

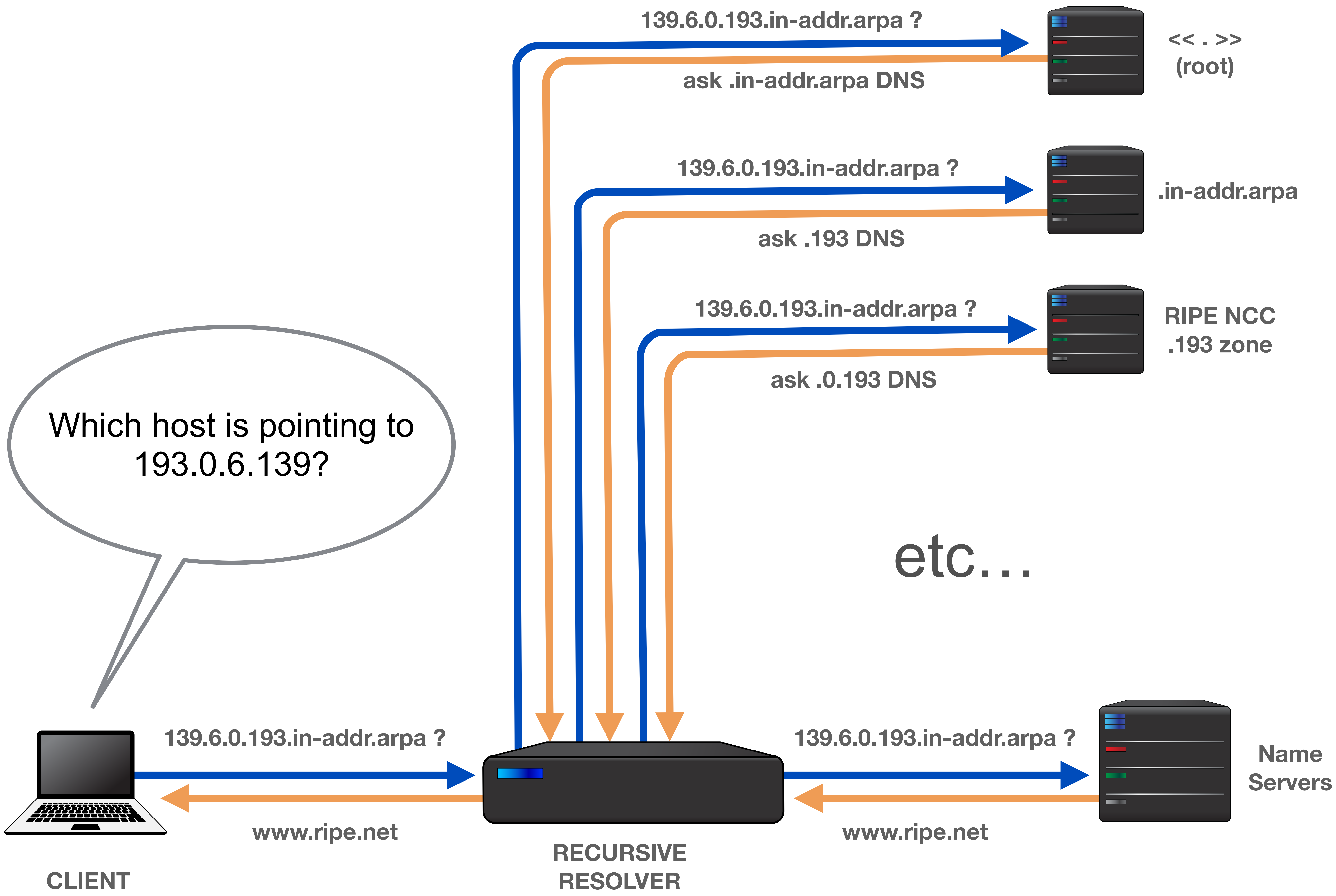
Purpose of Reverse DNS

Reverse DNS is used for:

- Identifying Spam
- Network Diagnostics
- Controlling Access to a Network



How does Reverse DNS Work?


139.6.0.193.in-addr.arpa ?
ask .in-addr.arpa DNS
<< . >>
(root)
139.6.0.193.in-addr.arpa ?
.in-addr.arpa
ask .193 DNS
139.6.0.193.in-addr.arpa ?
RIPE NCC
.193 zone
ask .0.193 DNS
Which host is pointing to 193.0.6.139?
etc…
139.6.0.193.in-addr.arpa ?
www.ripe.net
139.6.0.193.in-addr.arpa ?
www.ripe.net
Name
Servers
RECURSIVE
RESOLVER
CLIENT

Reverse Delegation Basics

IPv4	IPv6
in-addr.arpa zone	ip6.arpa zone
/24 or /16 blocks	Multiple of 4 bits
	/28, /32, /36, /40, /44, /48

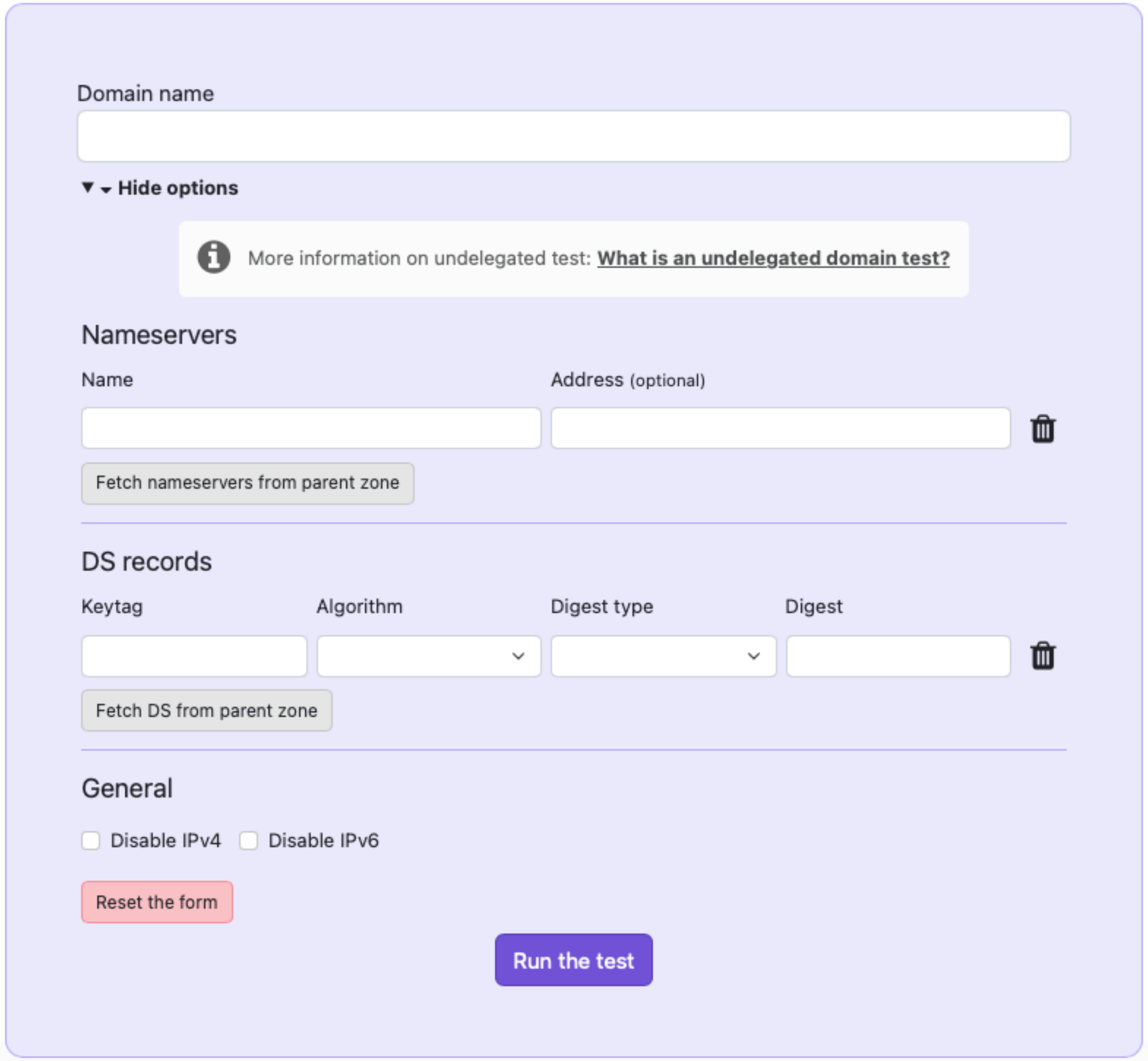
Setting up Reverse Delegation

Configure your DNS servers

- at least two name servers in different subnets
- create a zone file on each for each chunk

Check your zones:

http://dnscheck.ripe.net



Domain name

▼ ▾ Hide options

More information on undelegated test: **What is an undelegated domain test?**

Nameservers

Name	Address (optional)

DS records

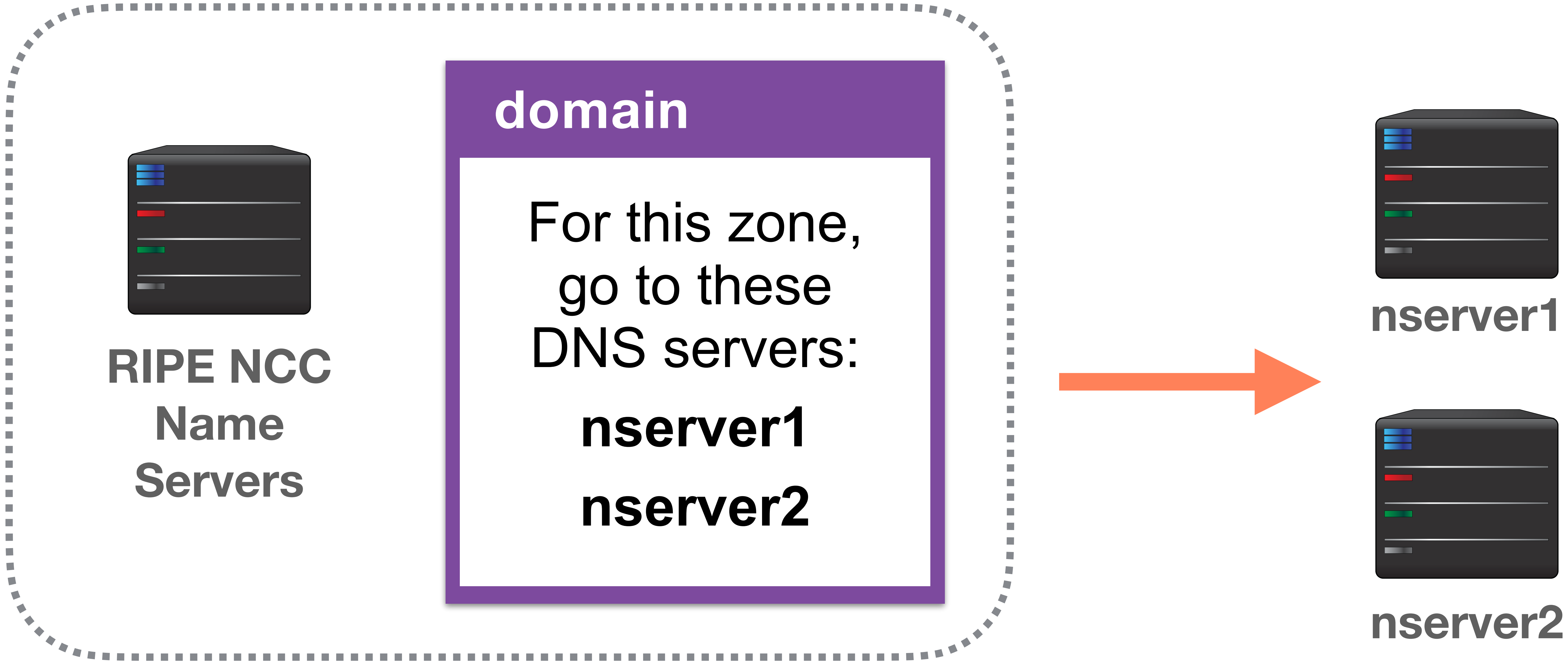
Keytag	Algorithm	Digest type	Digest

General

□ Disable IPv4 □ Disable IPv6

DOMAIN Objects

- Create records on RIPE NCC DNS servers
- They point to name servers that will be authoritative for the zone



Creating DOMAIN Objects

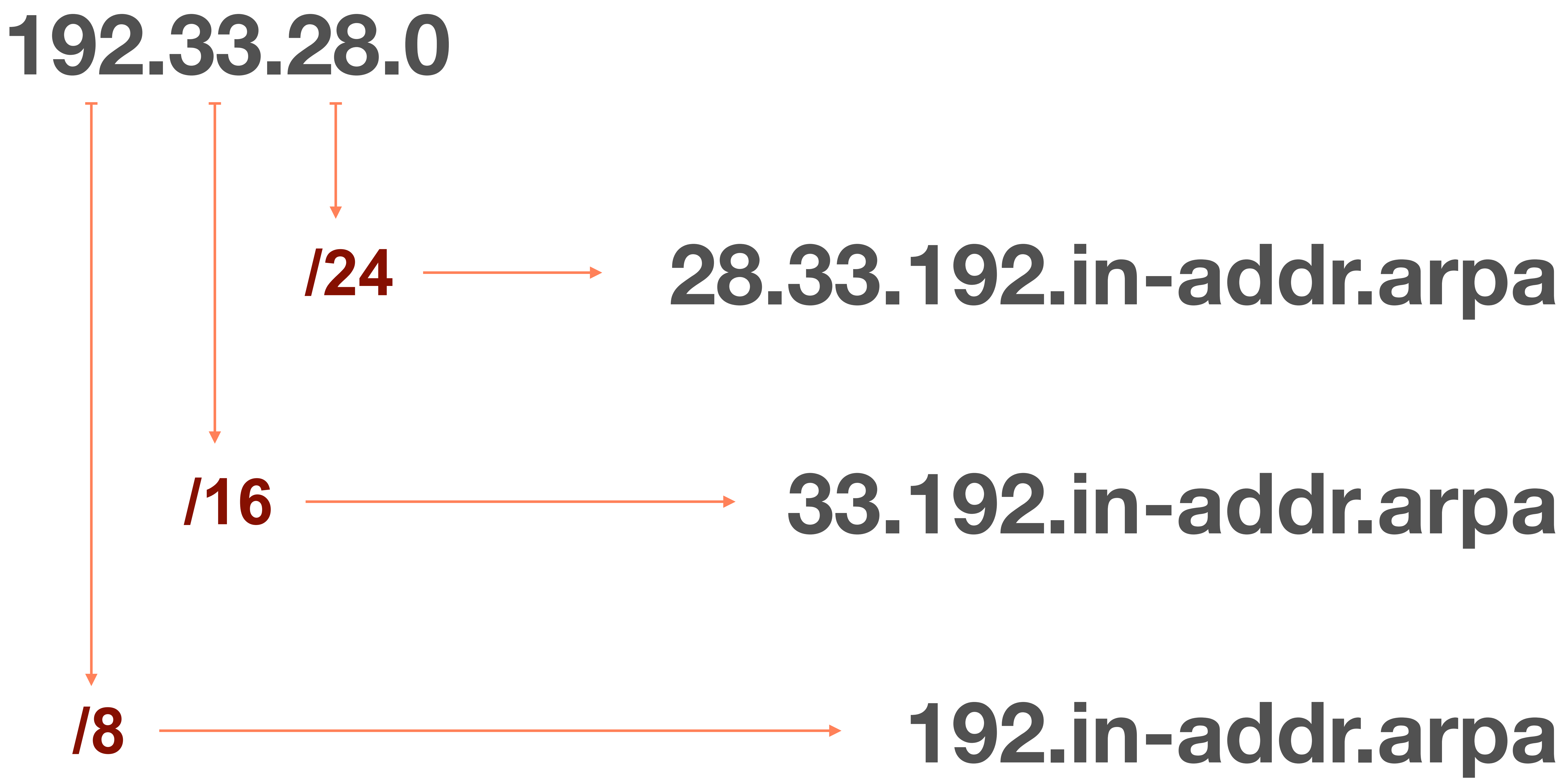
Which maintainers are on the address space?

Address Space

mnt-by:	SOME-BIG-MNT
mnt-lower:	ANOTHER-MNT
mnt-domains:	DNS-ZONE-MNT

mnt-domains allows you to delegate creation of domain objects to another maintainer

Reverse DNS for IPv4



192.33.28.0

/24 → 28.33.192.in-addr.arpa

/16 → 33.192.in-addr.arpa

/8 → 192.in-addr.arpa

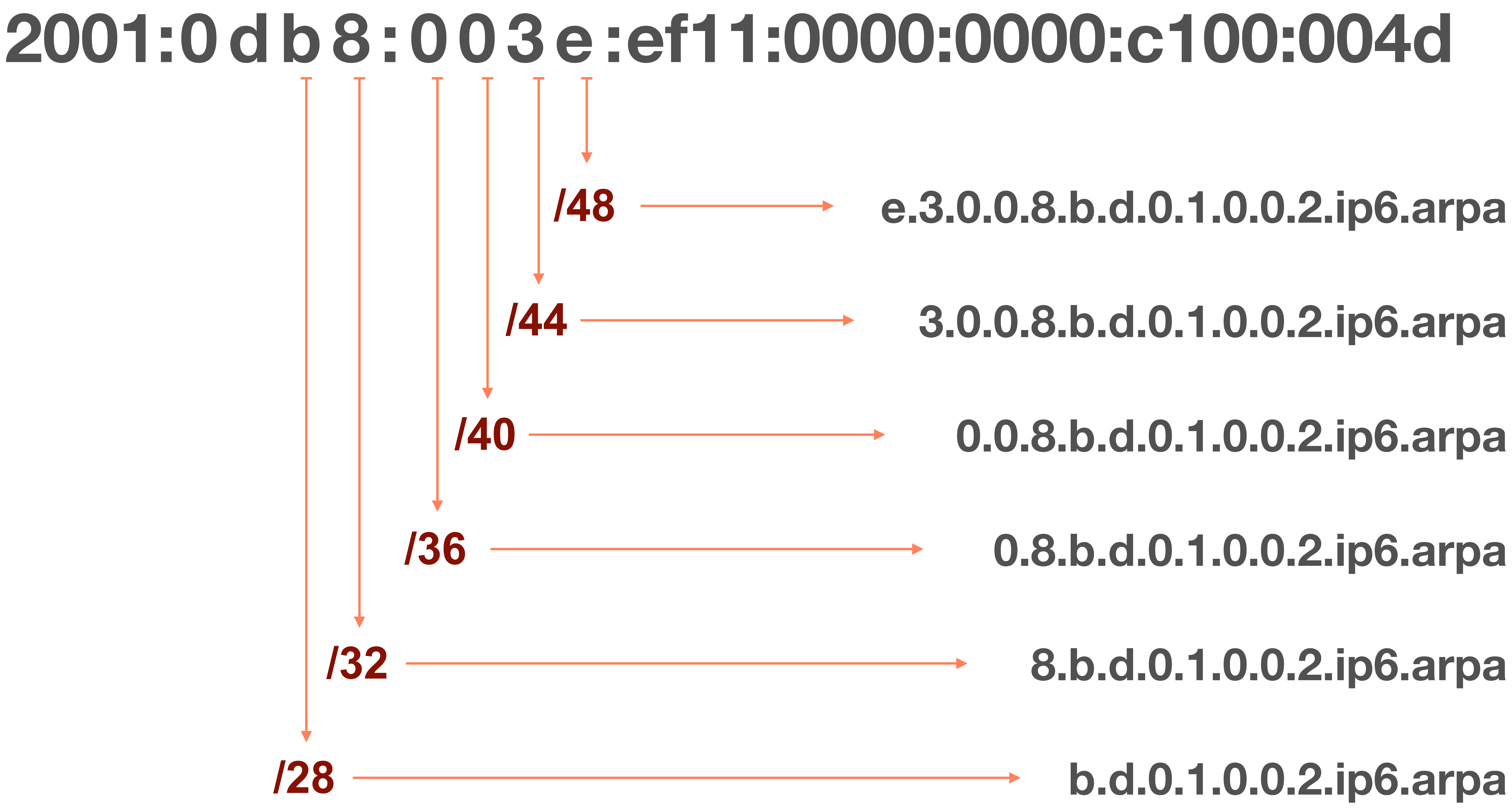
IPv4 and DOMAIN Objects

IPv4 prefix: 192.33.28.0/24

Domain object:

```
domain:          28.33.192.in-addr.arpa
descr:           rDNS for my IPv4 network
admin-c:         NOC12-RIPE
tech-c:          NOC12-RIPE
zone-c:          NOC12-RIPE
nserver:         pri.example.net
nserver:         sns.company.org
ds-rdata:        45062 8 2 275d9acbf3d3fec11b6d6…
mnt-by:          EXAMPLE-LIR—MNT
created:         2015-01-21T13:52:29Z
last-modified:   2016-02-07T15:09:46Z
source:          RIPE
```

Reverse DNS for IPv6



2001:0db8:003e:ef11:0000:0000:c100:004d

/48	e.3.0.0.8.b.d.0.1.0.0.2.ip6.arpa
/44	3.0.0.8.b.d.0.1.0.0.2.ip6.arpa
/40	0.0.8.b.d.0.1.0.0.2.ip6.arpa
/36	0.8.b.d.0.1.0.0.2.ip6.arpa
/32	8.b.d.0.1.0.0.2.ip6.arpa
/28	b.d.0.1.0.0.2.ip6.arpa

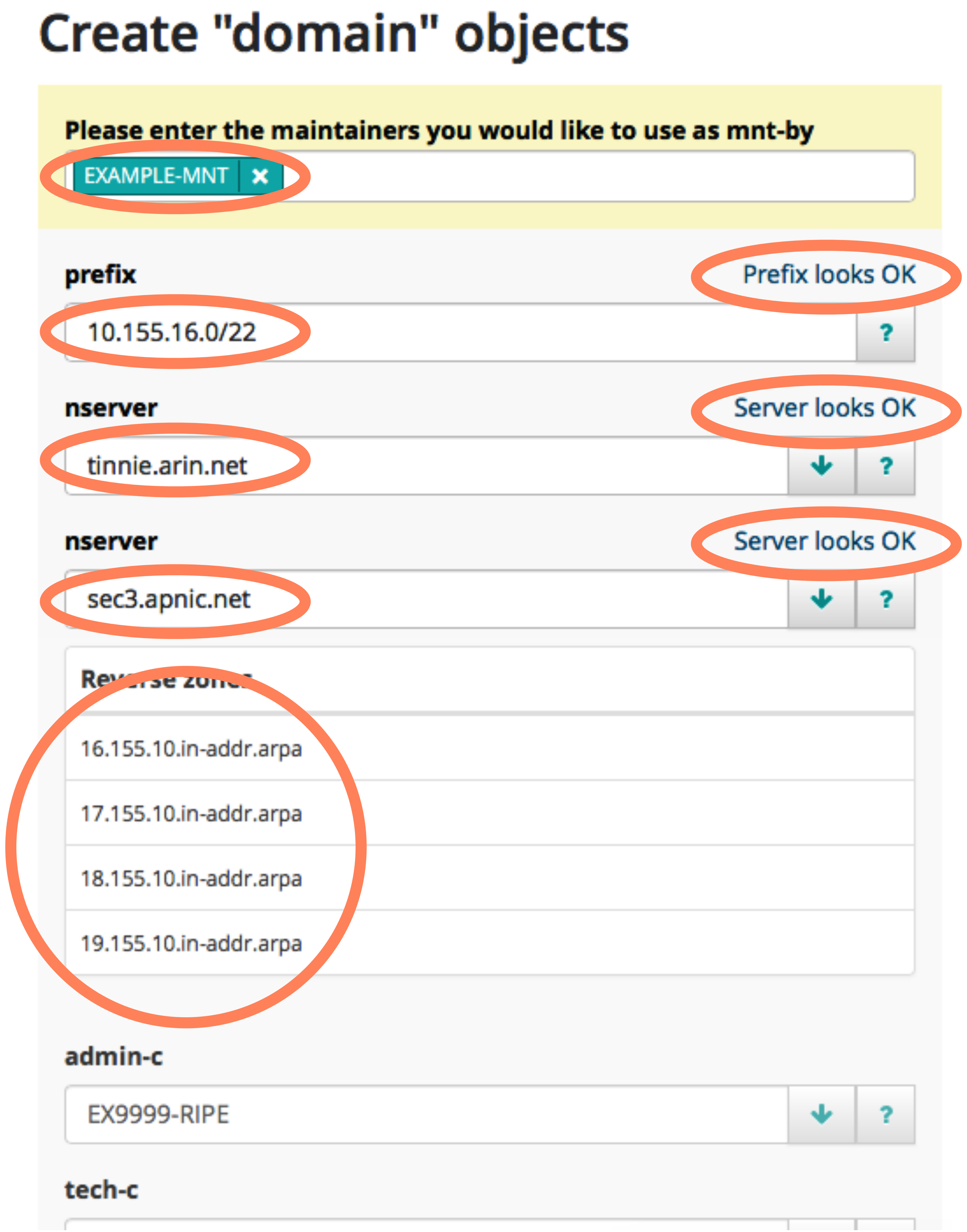
IPv6 and DOMAIN Objects

IPv6 prefix: 2001:db8::/32

Domain object:

```
domain:          8.b.d.0.1.0.0.2.ip6.arpa
descr:           rDNS for my IPv6 network
admin-c:         NOC12-RIPE
tech-c:          NOC12-RIPE
zone-c:          NOC12-RIPE
nserver:         pri.example.net
nserver:         sns.company.org
ds-rdata:        45062 8 2 275d9acbf3d3fec11b6d6…
mnt-by:          EXAMPLE-LIR—MNT
created:         2015-01-21T13:52:29Z
last-modified:   2016-02-07T15:09:46Z
source:          RIPE
```

Create DOMAIN Objects Wizard



Create "domain" objects

Please enter the maintainers you would like to use as mnt-by

EXAMPLE-MNT ✕

prefix — Prefix looks OK

10.155.16.0/22

nserver — Server looks OK

tinnie.arin.net

nserver — Server looks OK

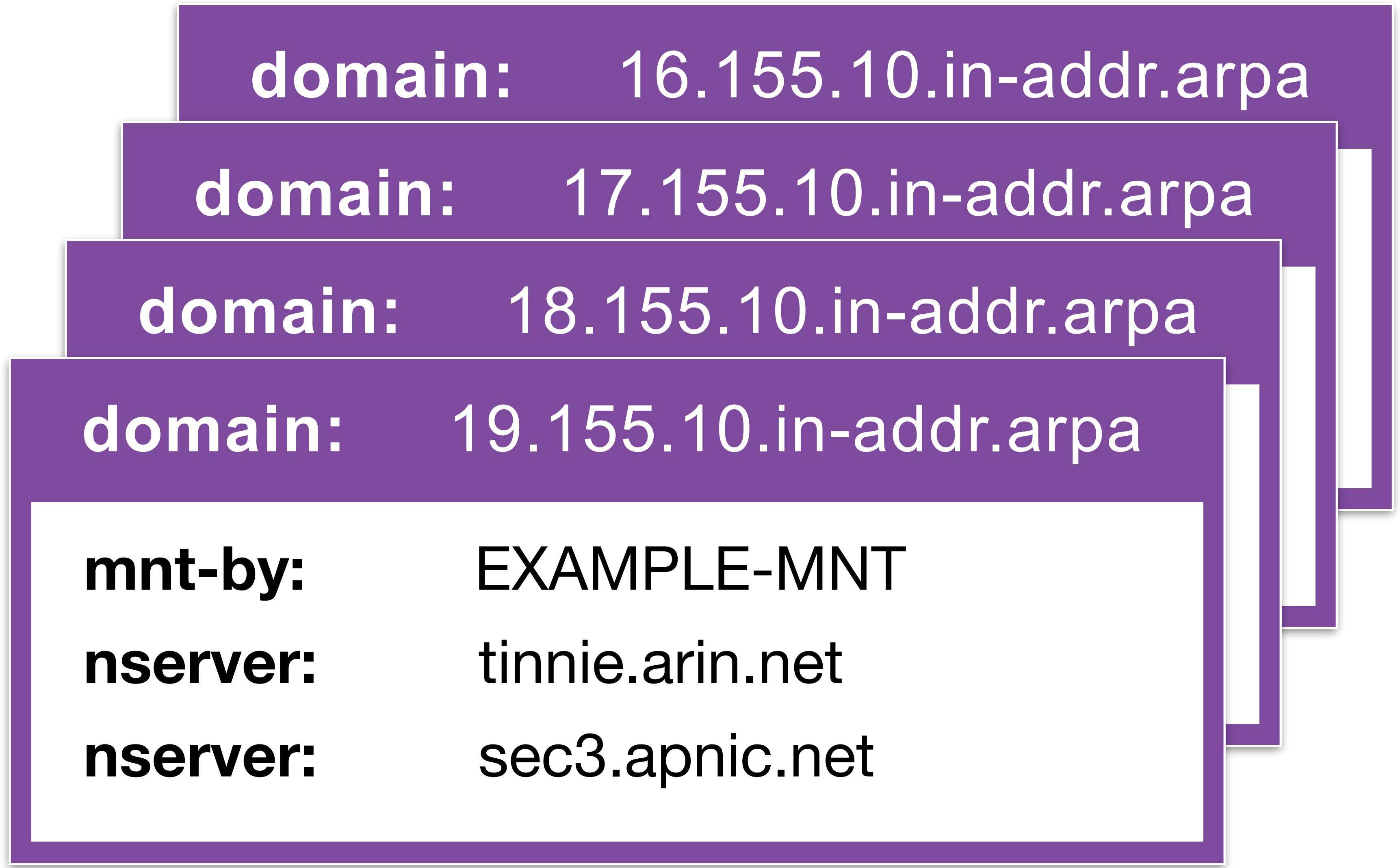
sec3.apnic.net

Reverse zones
16.155.10.in-addr.arpa
17.155.10.in-addr.arpa
18.155.10.in-addr.arpa
19.155.10.in-addr.arpa

admin-c

EX9999-RIPE

tech-c



domain: 16.155.10.in-addr.arpa

domain: 17.155.10.in-addr.arpa

domain: 18.155.10.in-addr.arpa

domain: 19.155.10.in-addr.arpa

mnt-by:	EXAMPLE-MNT
nserver:	tinnie.arin.net
nserver:	sec3.apnic.net

Exercise

How many domain objects?

Calculate How Many Objects

You have the following address space:

- 192.12.32.0/22
- 2a00:38::/29

How many domain objects do you have to create?

- Use the largest block size possible

What are the first and last domain objects for each?

And For The Customer?

What are the two domain objects for Marc Bromski's address space?

IPv4: 10.xx.2.0 – 10.xx.2.255

IPv6: 2002:ffxx:1001::/48

How to query for IPv6?

Which query would you use to find the /32 domain object for the IPv6 allocation 2001:db8::/32?

a) -Md 2001:db8::/32

b) -md 2001:db8::/32

c) -xd 2001:db8::/32

Questions

More RIPE Database

Inverse Lookups, Free Text Search,
Notifications, RIPE Database WG

Looking For References

You want to replace the reference to Jean Blue’s **person** object in all the LIR objects with your new LIR **role** object

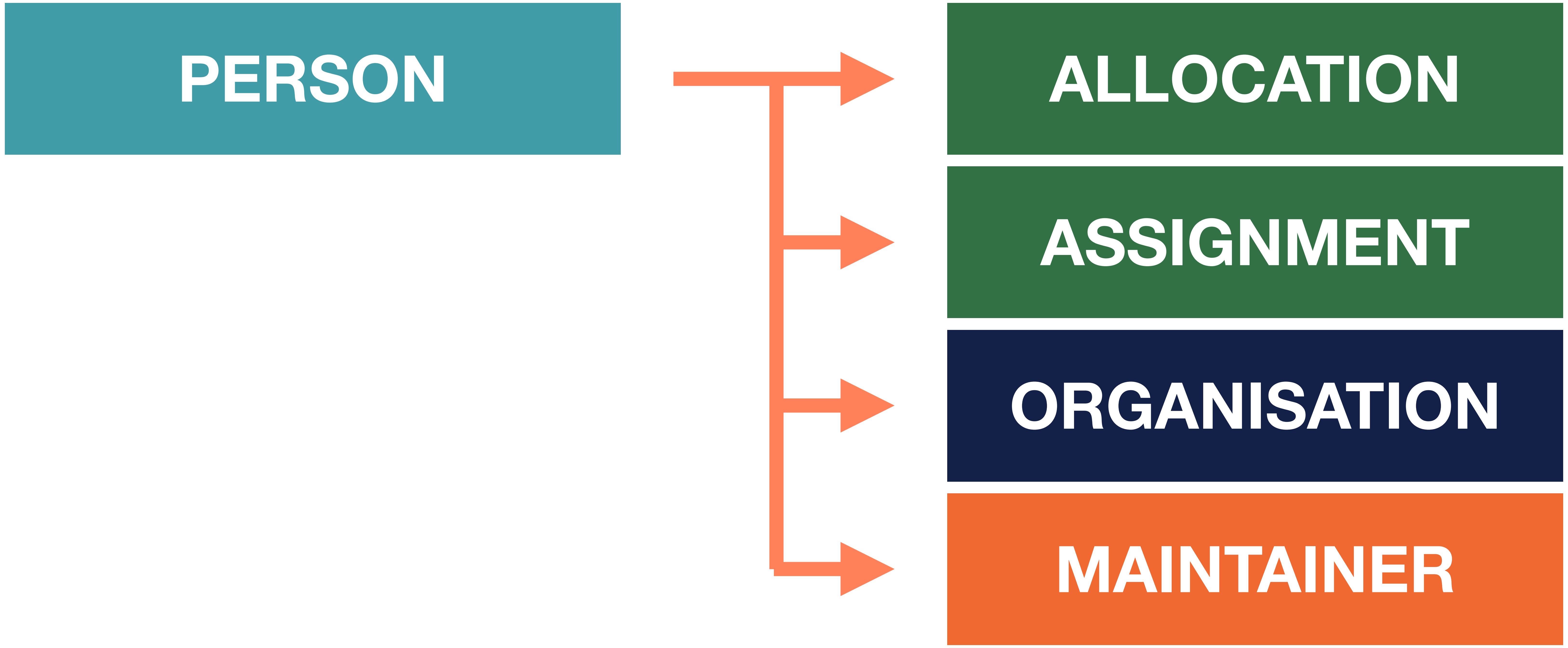
1. Go to http://apps-test.db.ripe.net
2. Search for “**-i person JBXX-TEST**”

What Do You See?

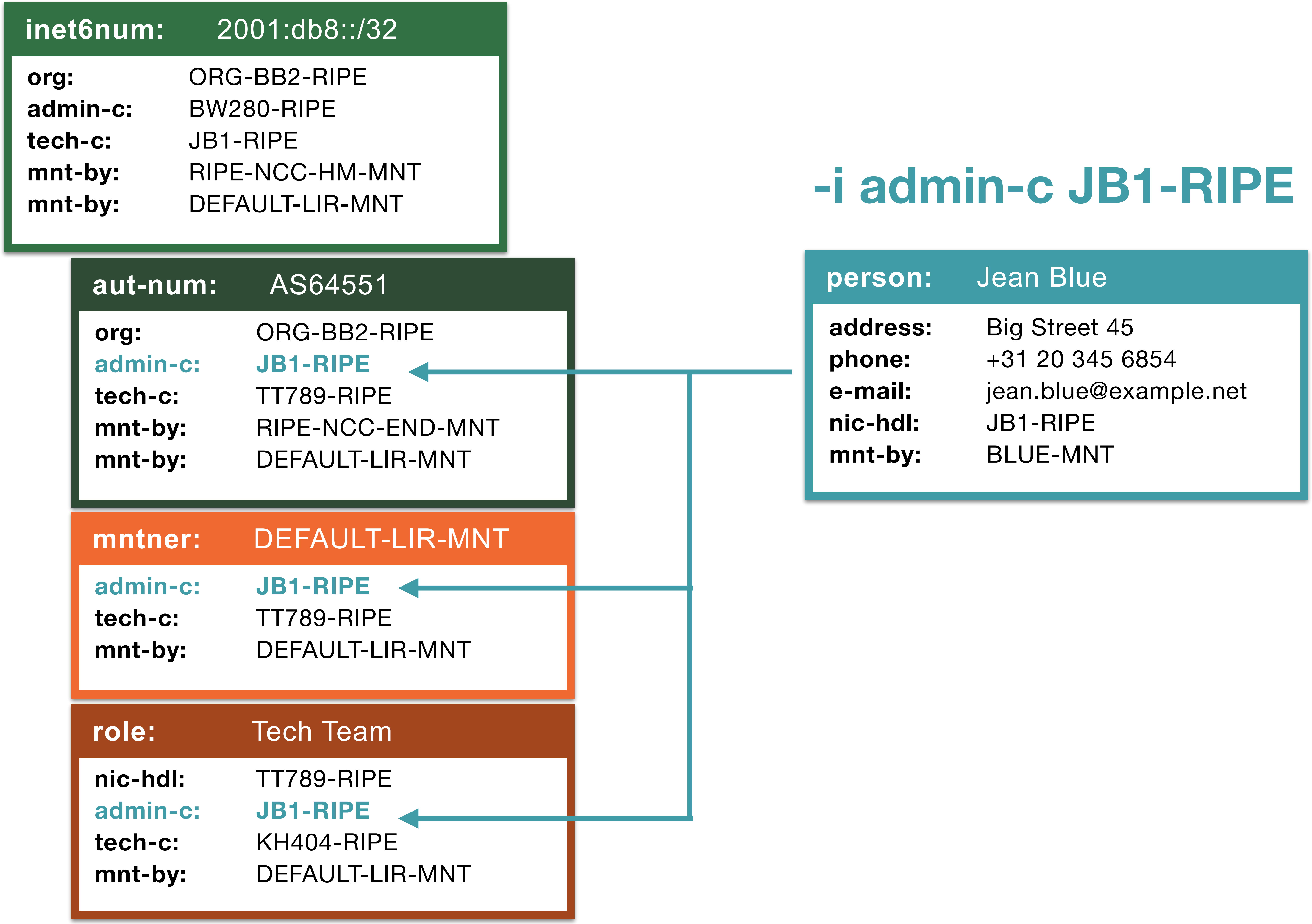
- Which objects are in the query results?
- Where do you see JBXX-TEST?

Inverse Lookups

Finding all objects in which an object is referenced



Inverse Lookup: admin-c



```
inet6num:   2001:db8::/32
org:        ORG-BB2-RIPE
admin-c:    BW280-RIPE
tech-c:     JB1-RIPE
mnt-by:     RIPE-NCC-HM-MNT
mnt-by:     DEFAULT-LIR-MNT
```

```
aut-num:    AS64551
org:        ORG-BB2-RIPE
admin-c:    JB1-RIPE
tech-c:     TT789-RIPE
mnt-by:     RIPE-NCC-END-MNT
mnt-by:     DEFAULT-LIR-MNT
```

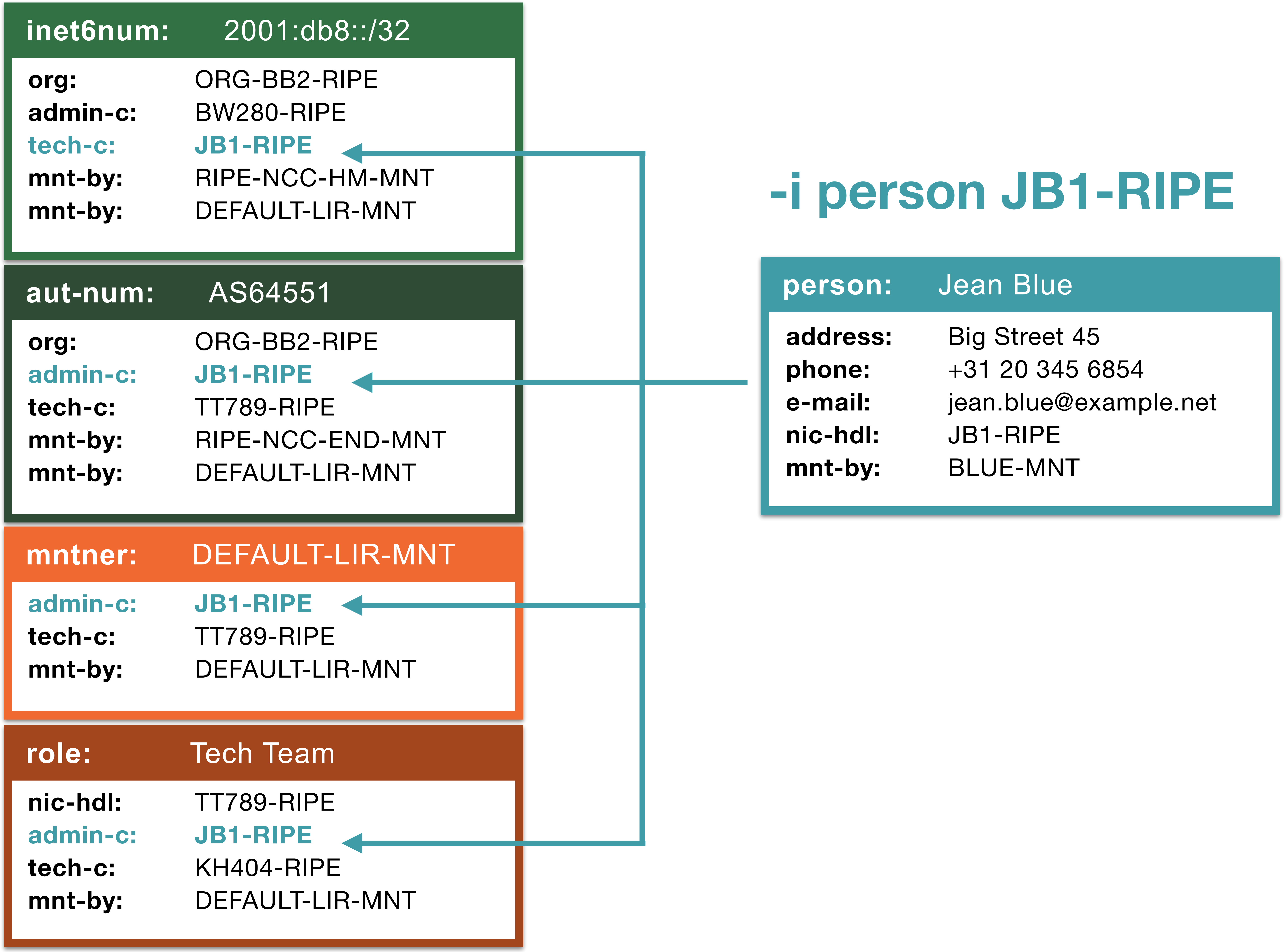
```
mntner:     DEFAULT-LIR-MNT
admin-c:    JB1-RIPE
tech-c:     TT789-RIPE
mnt-by:     DEFAULT-LIR-MNT
```

```
role:       Tech Team
nic-hdl:    TT789-RIPE
admin-c:    JB1-RIPE
tech-c:     KH404-RIPE
mnt-by:     DEFAULT-LIR-MNT
```

```
-i admin-c JB1-RIPE
```

```
person:     Jean Blue
address:    Big Street 45
phone:      +31 20 345 6854
e-mail:     jean.blue@example.net
nic-hdl:    JB1-RIPE
mnt-by:     BLUE-MNT
```

Inverse Lookup: person



```
inet6num:    2001:db8::/32
org:         ORG-BB2-RIPE
admin-c:     BW280-RIPE
tech-c:      JB1-RIPE
mnt-by:      RIPE-NCC-HM-MNT
mnt-by:      DEFAULT-LIR-MNT
```

```
aut-num:     AS64551
org:         ORG-BB2-RIPE
admin-c:     JB1-RIPE
tech-c:      TT789-RIPE
mnt-by:      RIPE-NCC-END-MNT
mnt-by:      DEFAULT-LIR-MNT
```

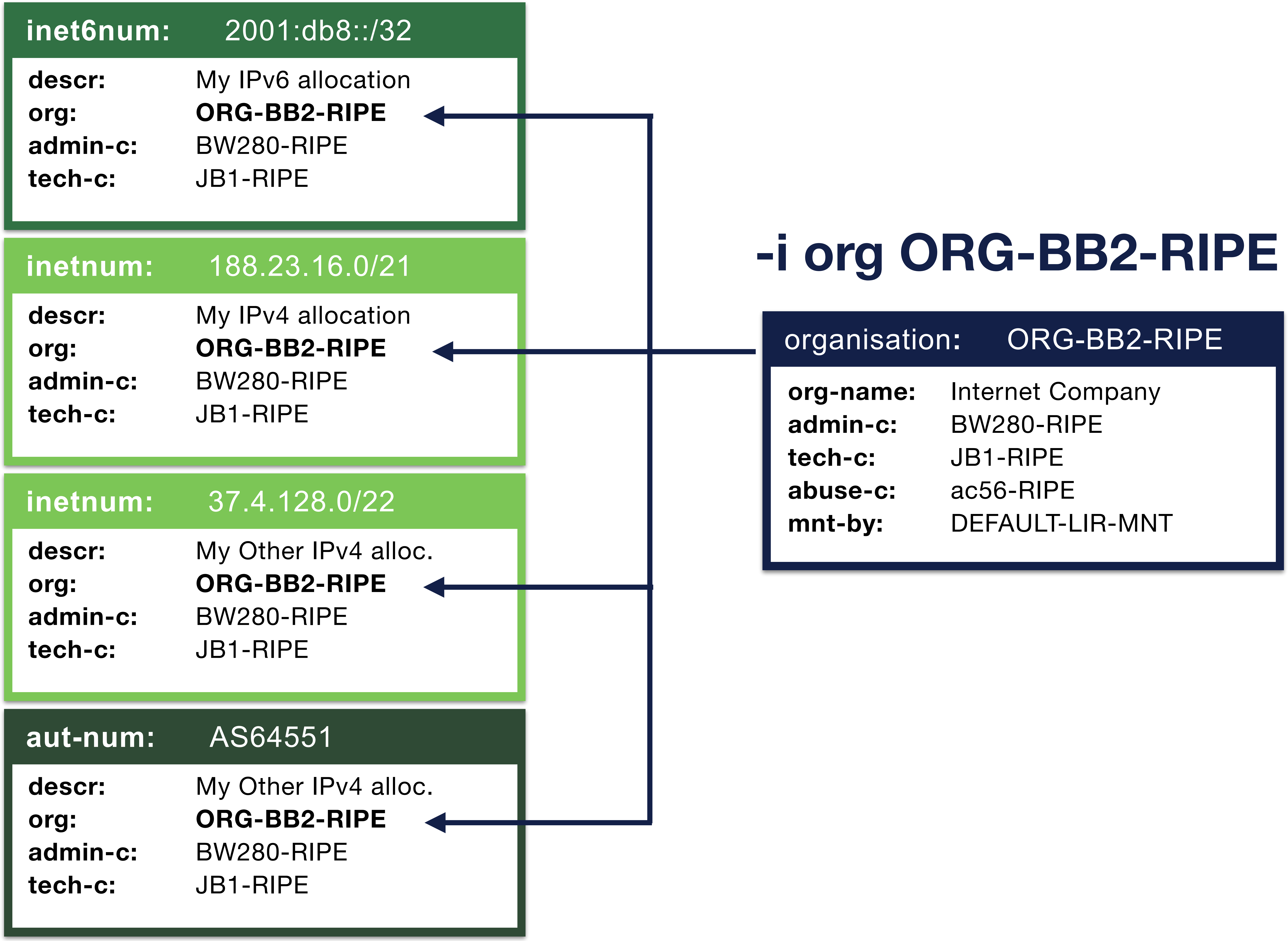
```
mntner:      DEFAULT-LIR-MNT
admin-c:     JB1-RIPE
tech-c:      TT789-RIPE
mnt-by:      DEFAULT-LIR-MNT
```

```
role:        Tech Team
nic-hdl:     TT789-RIPE
admin-c:     JB1-RIPE
tech-c:      KH404-RIPE
mnt-by:      DEFAULT-LIR-MNT
```

-i person JB1-RIPE

```
person:      Jean Blue
address:     Big Street 45
phone:       +31 20 345 6854
e-mail:      jean.blue@example.net
nic-hdl:     JB1-RIPE
mnt-by:      BLUE-MNT
```

Inverse Lookup: organisation



```
inet6num:     2001:db8::/32
descr:        My IPv6 allocation
org:          ORG-BB2-RIPE
admin-c:      BW280-RIPE
tech-c:       JB1-RIPE
```

```
inetnum:      188.23.16.0/21
descr:        My IPv4 allocation
org:          ORG-BB2-RIPE
admin-c:      BW280-RIPE
tech-c:       JB1-RIPE
```

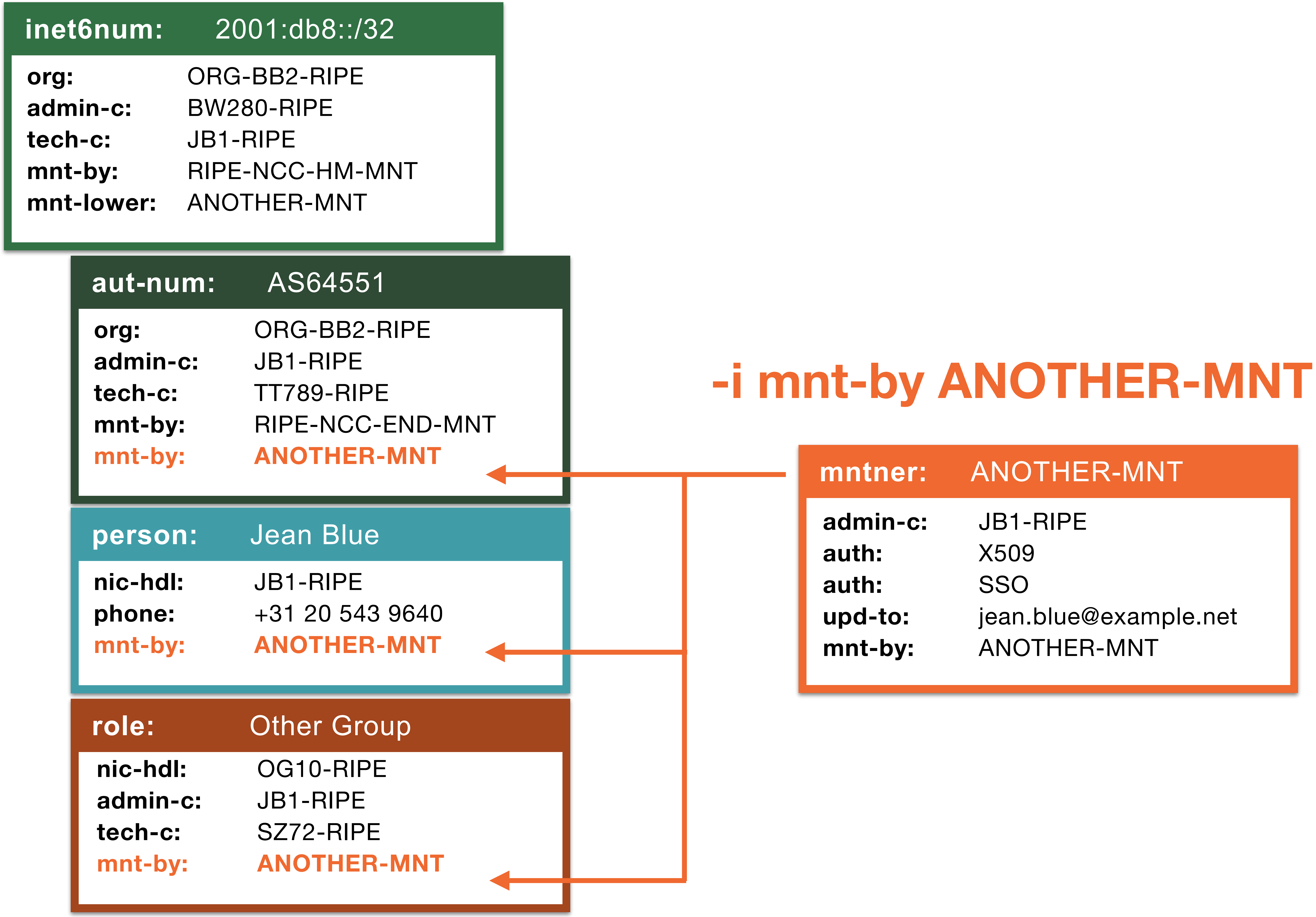
```
inetnum:      37.4.128.0/22
descr:        My Other IPv4 alloc.
org:          ORG-BB2-RIPE
admin-c:      BW280-RIPE
tech-c:       JB1-RIPE
```

```
aut-num:      AS64551
descr:        My Other IPv4 alloc.
org:          ORG-BB2-RIPE
admin-c:      BW280-RIPE
tech-c:       JB1-RIPE
```

-i org ORG-BB2-RIPE

```
organisation:   ORG-BB2-RIPE
org-name:       Internet Company
admin-c:        BW280-RIPE
tech-c:         JB1-RIPE
abuse-c:        ac56-RIPE
mnt-by:         DEFAULT-LIR-MNT
```

Inverse Lookup : mnt-by



```
inet6num:    2001:db8::/32
org:         ORG-BB2-RIPE
admin-c:     BW280-RIPE
tech-c:      JB1-RIPE
mnt-by:      RIPE-NCC-HM-MNT
mnt-lower:   ANOTHER-MNT
```

```
aut-num:     AS64551
org:         ORG-BB2-RIPE
admin-c:     JB1-RIPE
tech-c:      TT789-RIPE
mnt-by:      RIPE-NCC-END-MNT
mnt-by:      ANOTHER-MNT
```

```
person:      Jean Blue
nic-hdl:     JB1-RIPE
phone:       +31 20 543 9640
mnt-by:      ANOTHER-MNT
```

```
role:        Other Group
nic-hdl:     OG10-RIPE
admin-c:     JB1-RIPE
tech-c:      SZ72-RIPE
mnt-by:      ANOTHER-MNT
```

```
-i mnt-by ANOTHER-MNT
```

```
mntner:      ANOTHER-MNT
admin-c:     JB1-RIPE
auth:        X509
auth:        SSO
upd-to:      jean.blue@example.net
mnt-by:      ANOTHER-MNT
```

Search For A Word

You want to look for every object that has the word “**uplink**” in any of the attributes

1. Go to https://apps.db.ripe.net/

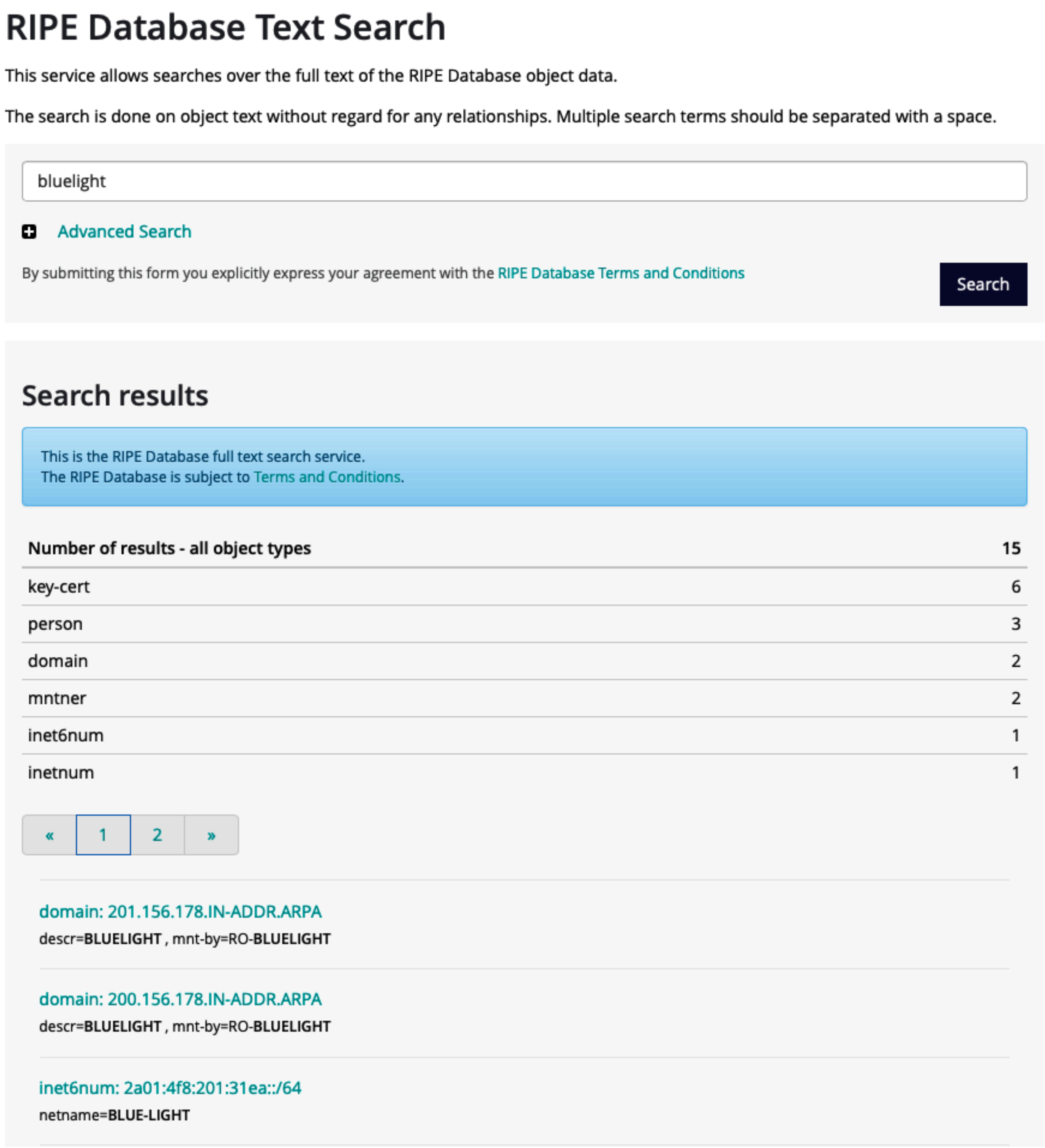
2. Click on the left menu on “**Full Text Search**”

3. Search for “**uplink**”

What Do You See?

- Do you get any objects in the results?
- How many objects do you get?
- Can you see the whole object?

Full Text Search



RIPE Database Text Search

This service allows searches over the full text of the RIPE Database object data.

The search is done on object text without regard for any relationships. Multiple search terms should be separated with a space.

By submitting this form you explicitly express your agreement with the RIPE Database Terms and Conditions

Search results

This is the RIPE Database full text search service.
The RIPE Database is subject to Terms and Conditions.

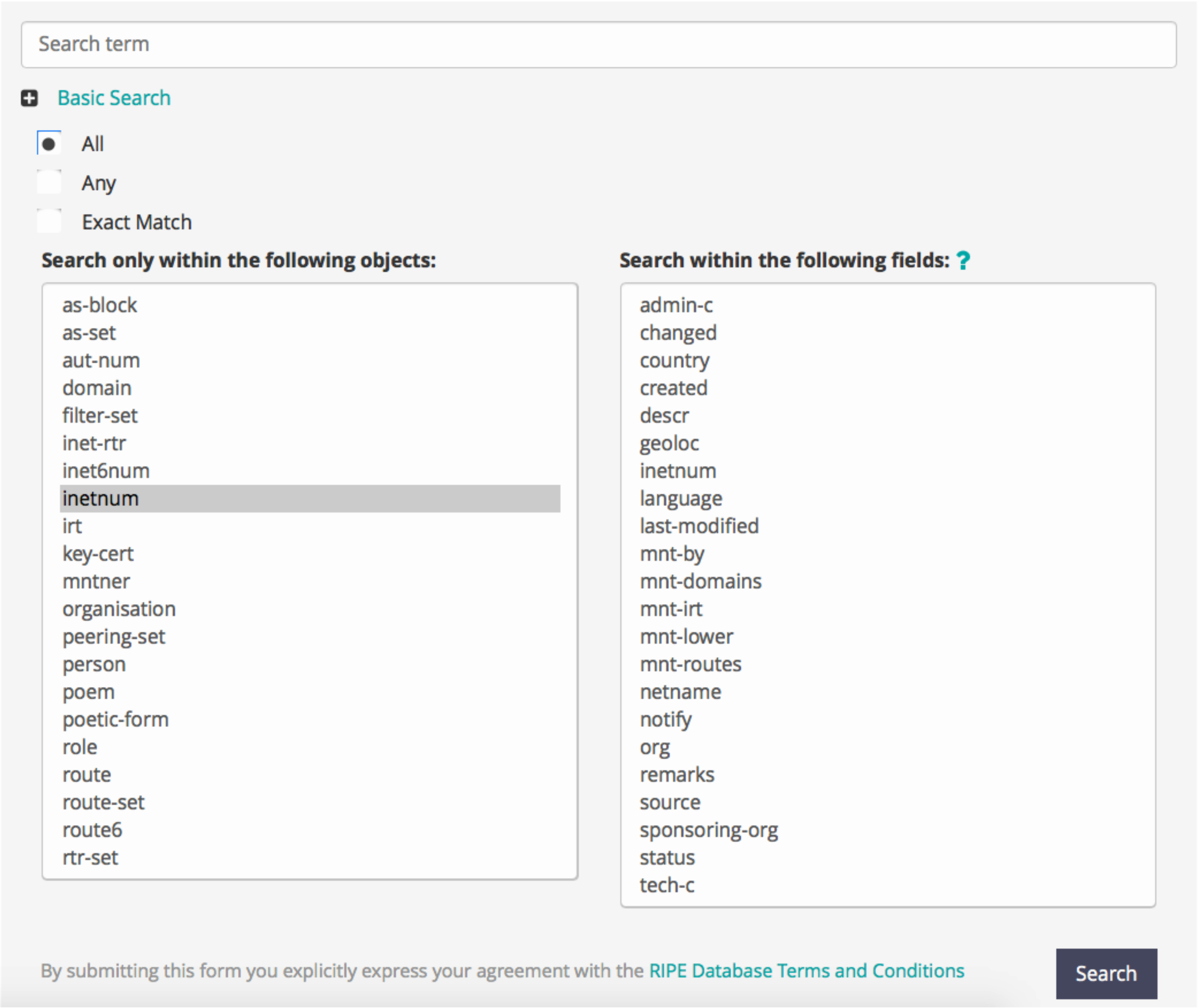
Number of results - all object types	15
key-cert	6
person	3
domain	2
mntner	2
inet6num	1
inetnum	1

domain: 201.156.178.IN-ADDR.ARPA
descr=**BLUELIGHT** , mnt-by=RO-**BLUELIGHT**

domain: 200.156.178.IN-ADDR.ARPA
descr=**BLUELIGHT** , mnt-by=RO-**BLUELIGHT**

inet6num: 2a01:4f8:201:31ea::/64
netname=**BLUE-LIGHT**

Full Text Search - Advanced



Search term

⊞ Basic Search

● All

○ Any

○ Exact Match

Search only within the following objects:

- as-block
- as-set
- aut-num
- domain
- filter-set
- inet-rtr
- inet6num
- inetnum
- irt
- key-cert
- mntner
- organisation
- peering-set
- person
- poem
- poetic-form
- role
- route
- route-set
- route6
- rtr-set

Search within the following fields: ?

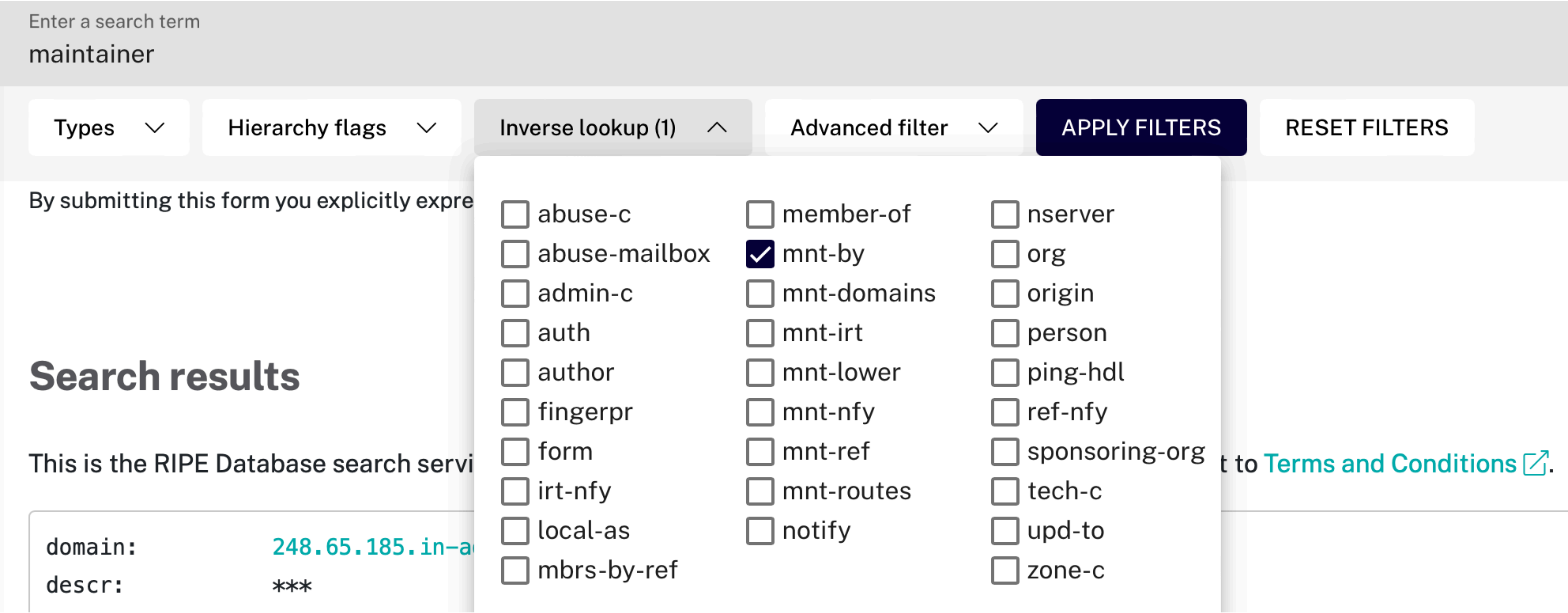
- admin-c
- changed
- country
- created
- descr
- geoloc
- inetnum
- language
- last-modified
- mnt-by
- mnt-domains
- mnt-irt
- mnt-lower
- mnt-routes
- netname
- notify
- org
- remarks
- source
- sponsoring-org
- status
- tech-c

By submitting this form you explicitly express your agreement with the RIPE Database Terms and Conditions

Mass Updates with Syncupdates

- Update multiple objects in one go
- Objects must be prepared beforehand

1. Find all the objects you want to modify
   - For example, using inverse lookups



Enter a search term
maintainer

Inverse lookup (1)
☑ mnt-by

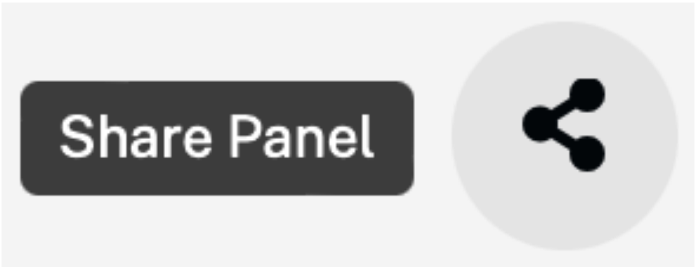
By submitting this form you explicitly expre

Search results

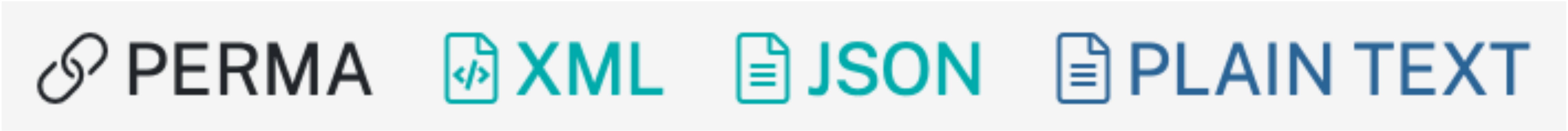
This is the RIPE Database search servi t to Terms and Conditions.

```
domain:         248.65.185.in-a
descr:          ***
```

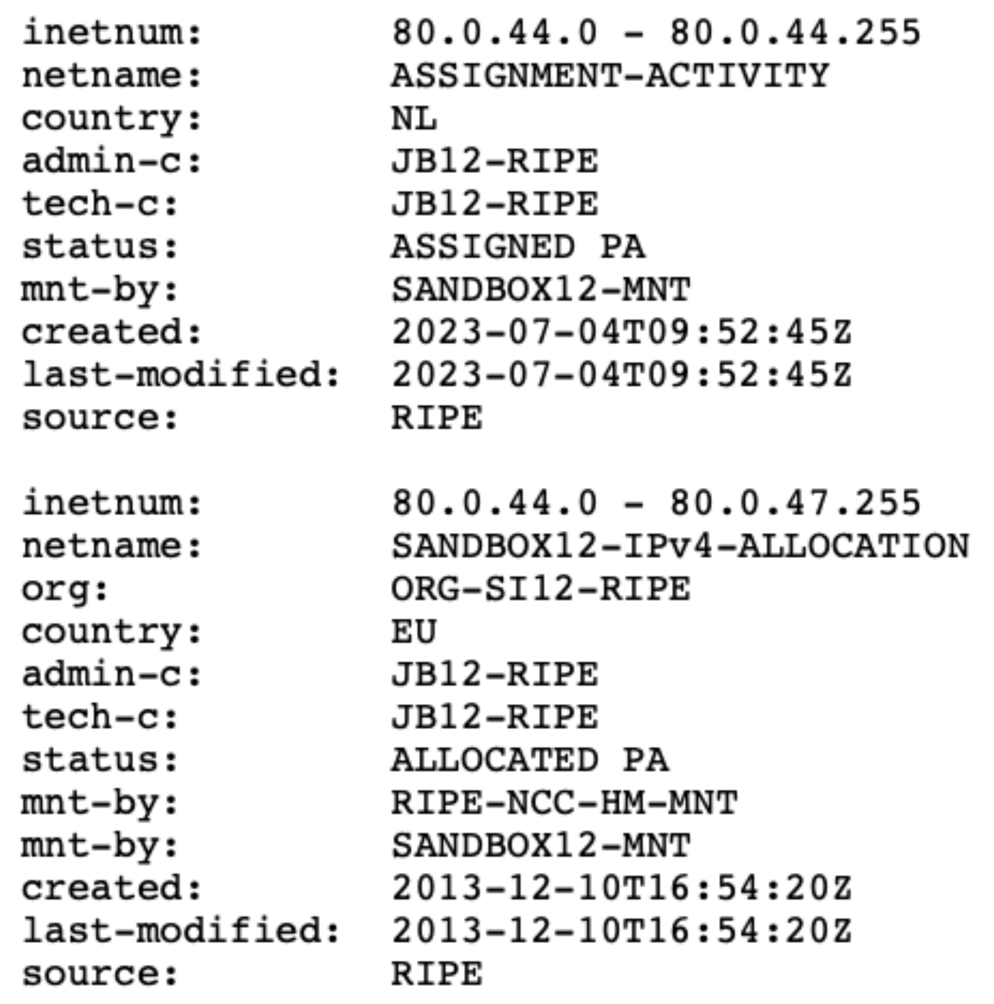
2. Click on the ‘Share Panel’ button



3. Click on the ‘Plain Text’ option

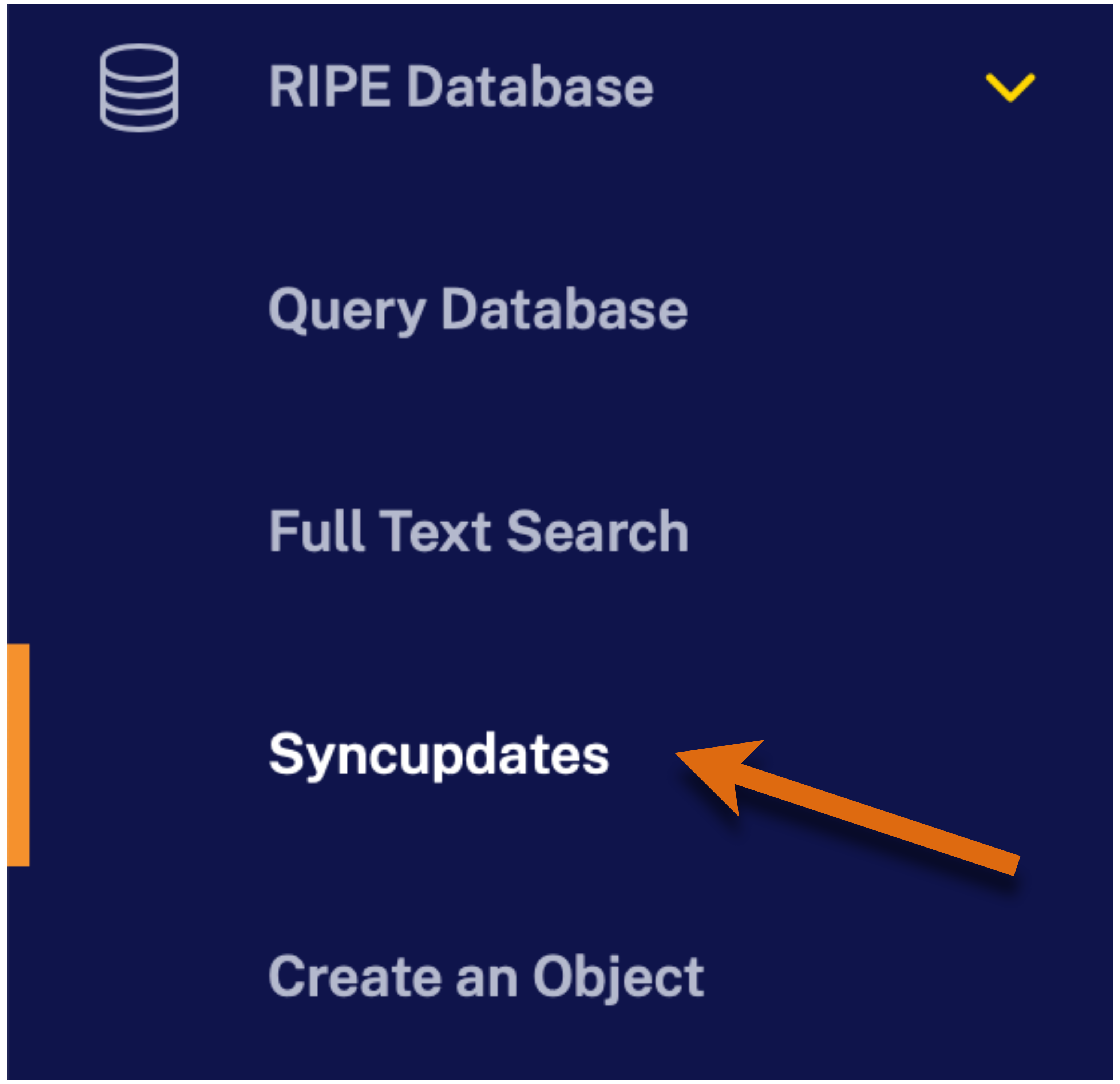


4. This opens a new browser tab with the objects

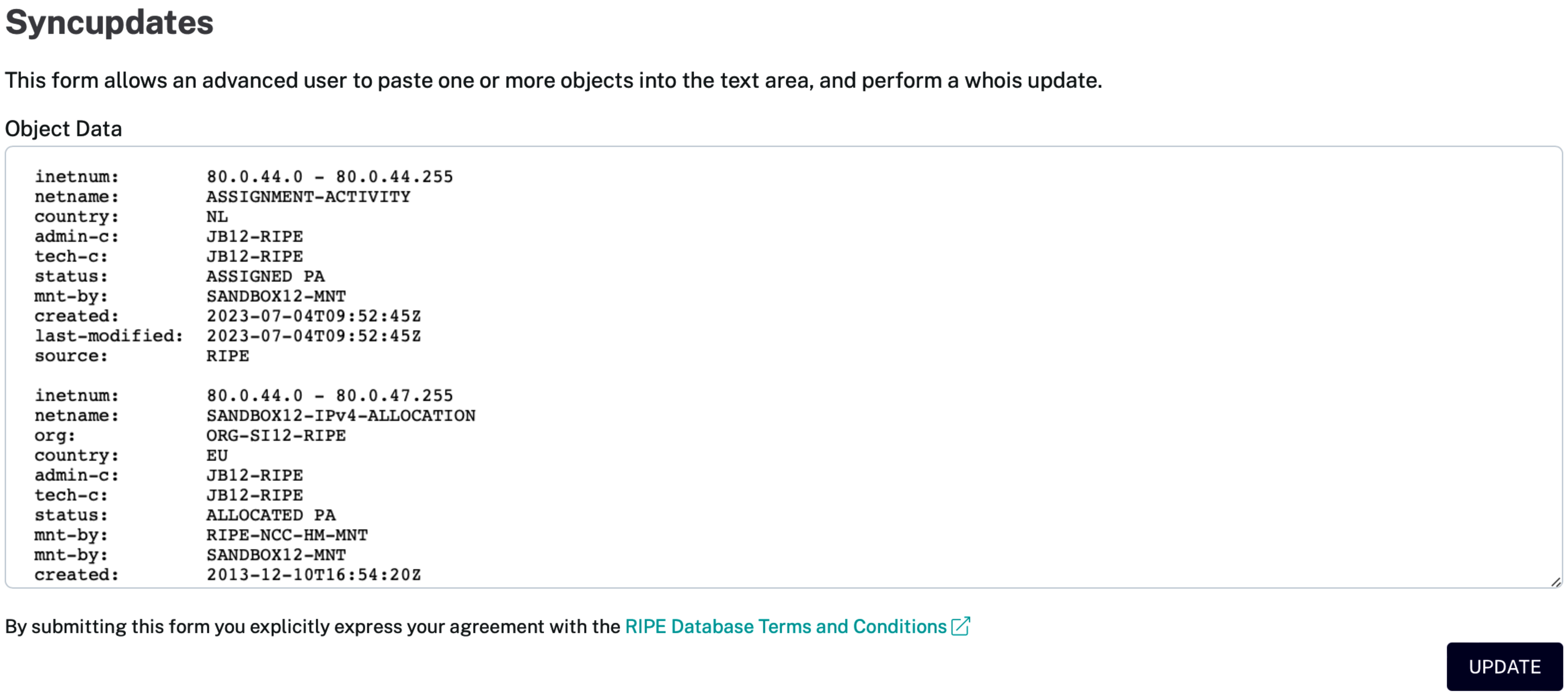


```
inetnum:        80.0.44.0 - 80.0.44.255
netname:        ASSIGNMENT-ACTIVITY
country:        NL
admin-c:        JB12-RIPE
tech-c:         JB12-RIPE
status:         ASSIGNED PA
mnt-by:         SANDBOX12-MNT
created:        2023-07-04T09:52:45Z
last-modified:  2023-07-04T09:52:45Z
source:         RIPE

inetnum:        80.0.44.0 - 80.0.47.255
netname:        SANDBOX12-IPv4-ALLOCATION
org:            ORG-SI12-RIPE
country:        EU
admin-c:        JB12-RIPE
tech-c:         JB12-RIPE
status:         ALLOCATED PA
mnt-by:         RIPE-NCC-HM-MNT
mnt-by:         SANDBOX12-MNT
created:        2013-12-10T16:54:20Z
last-modified:  2013-12-10T16:54:20Z
source:         RIPE
```

5. Copy/paste the objects to a text editor
6. Update them as needed
7. Click on **Syncupdates**



8. Copy/paste the objects into **Syncupdates**



Syncupdates

This form allows an advanced user to paste one or more objects into the text area, and perform a whois update.

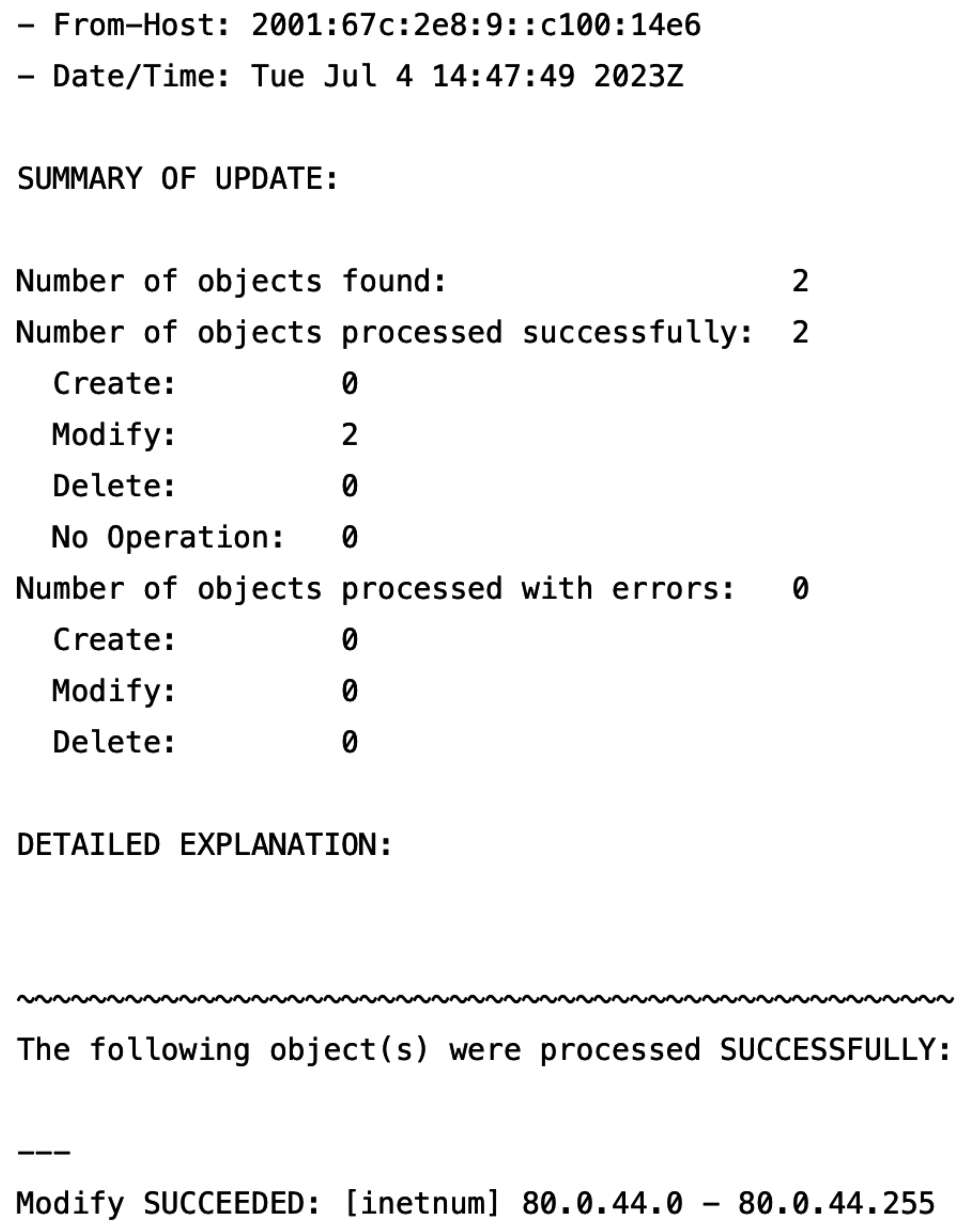
Object Data

```
inetnum:        80.0.44.0 - 80.0.44.255
netname:        ASSIGNMENT-ACTIVITY
country:        NL
admin-c:        JB12-RIPE
tech-c:         JB12-RIPE
status:         ASSIGNED PA
mnt-by:         SANDBOX12-MNT
created:        2023-07-04T09:52:45Z
last-modified:  2023-07-04T09:52:45Z
source:         RIPE

inetnum:        80.0.44.0 - 80.0.47.255
netname:        SANDBOX12-IPv4-ALLOCATION
org:            ORG-SI12-RIPE
country:        EU
admin-c:        JB12-RIPE
tech-c:         JB12-RIPE
status:         ALLOCATED PA
mnt-by:         RIPE-NCC-HM-MNT
mnt-by:         SANDBOX12-MNT
created:        2013-12-10T16:54:20Z
```

By submitting this form you explicitly express your agreement with the RIPE Database Terms and Conditions

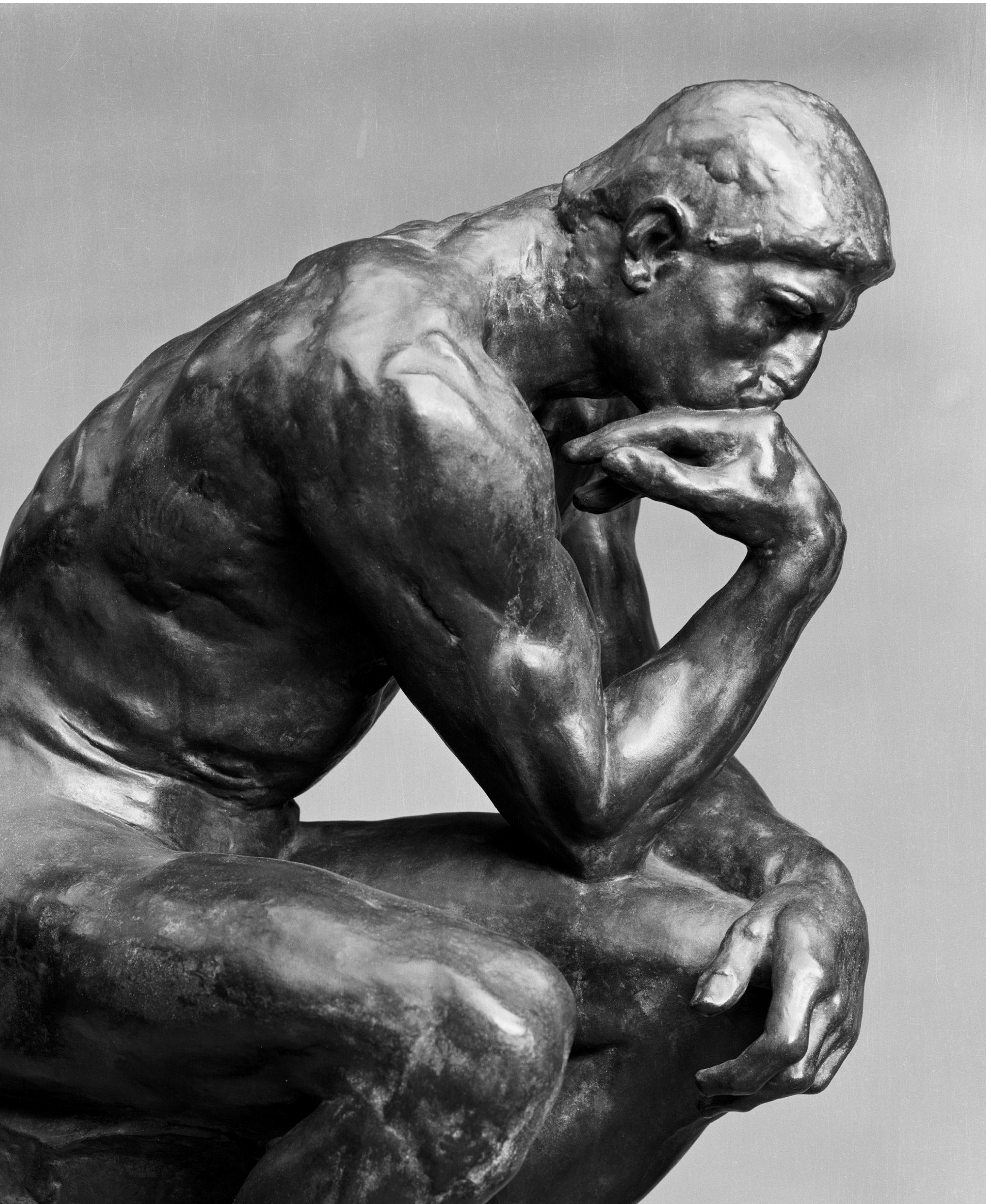
9. Click on the [**Update**] button

10. **Syncupdates** will process the objects and report results



```
- From-Host: 2001:67c:2e8:9::c100:14e6
- Date/Time: Tue Jul 4 14:47:49 2023Z

SUMMARY OF UPDATE:

Number of objects found:                    2
Number of objects processed successfully:   2
  Create:         0
  Modify:         2
  Delete:         0
  No Operation:   0
Number of objects processed with errors:    0
  Create:         0
  Modify:         0
  Delete:         0

DETAILED EXPLANATION:


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The following object(s) were processed SUCCESSFULLY:

---
Modify SUCCEEDED: [inetnum] 80.0.44.0 - 80.0.44.255
```

Think About This...

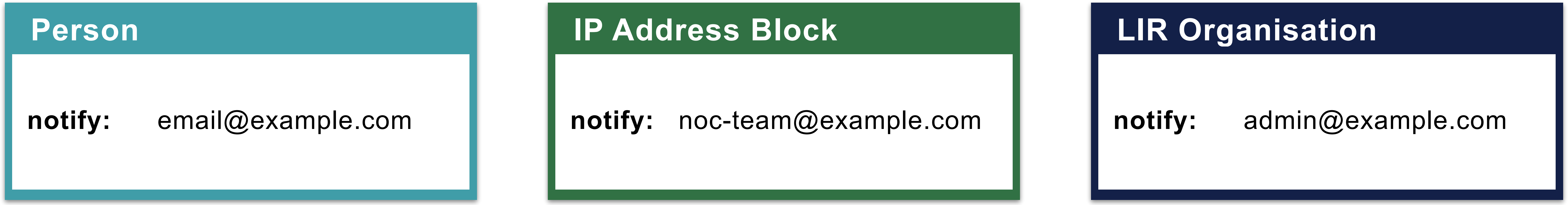


- The RIPE Database is a **public** database
- **Anybody** can search in the database
- **Who** can make updates?
- How can you **know** if somebody updates your objects?

Notifications: “notify:”

The RIPE Database has several ways to trigger notifications about updates to objects

- “**notify:**” attribute
  - Can be used on any object
  - An email is sent when the object is updated



Person

notify: email@example.com

IP Address Block

notify: noc-team@example.com

LIR Organisation

notify: admin@example.com

Notifications: Maintainers

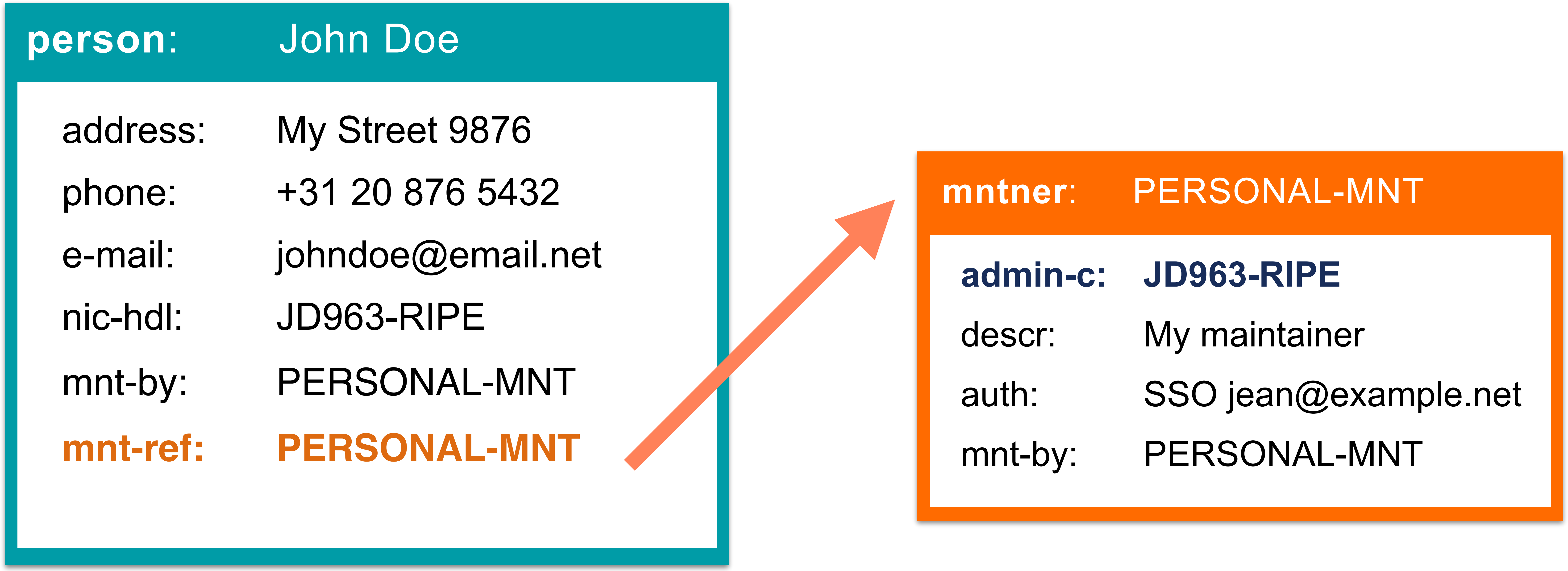
Maintainers have special attributes

- **“upd-to:”**
  - For **failed** attempts to update objects
- **“mnt-nfy:”**
  - For **successful** attempts to update objects

```
mntner:    LIR-MNT
upd-to:    db-alerts@example.com
mnt-nfy:   db-success@example.com
```

Avoid Unauthorised Referencing

- Anybody can reference **person** / **role** objects
- To protect from unauthorised references: **mnt-ref**
- Point to a maintainer that will authorise the reference to the object

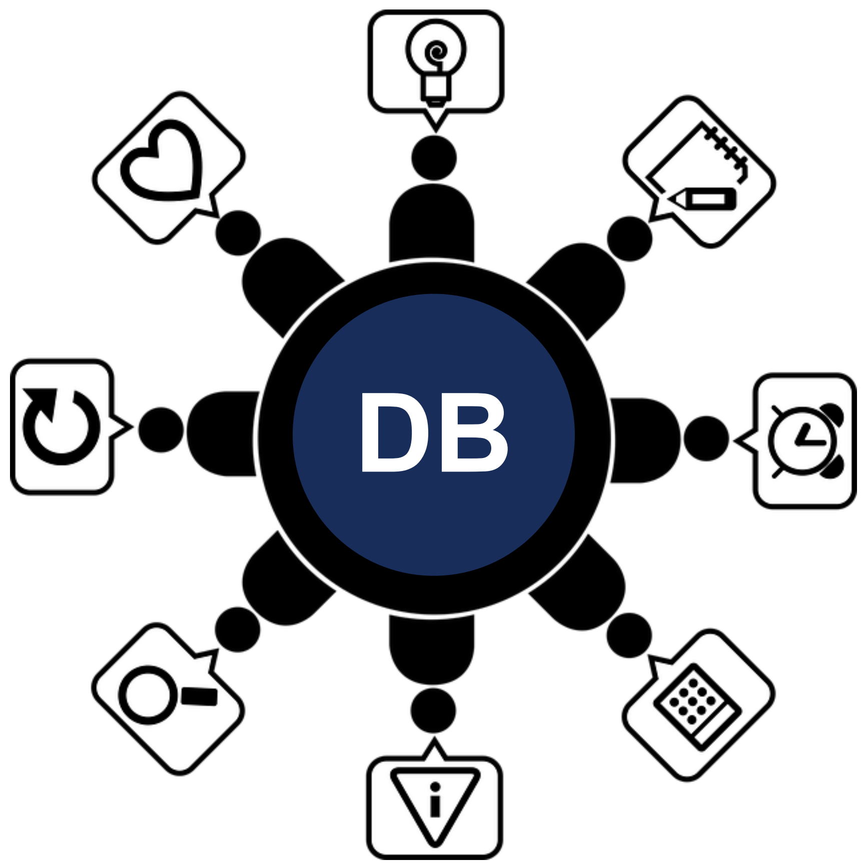


```
person:     John Doe
address:    My Street 9876
phone:      +31 20 876 5432
e-mail:     johndoe@email.net
nic-hdl:    JD963-RIPE
mnt-by:     PERSONAL-MNT
mnt-ref:    PERSONAL-MNT
```

```
mntner:     PERSONAL-MNT
admin-c:    JD963-RIPE
descr:      My maintainer
auth:       SSO jean@example.net
mnt-by:     PERSONAL-MNT
```

RIPE Database Working Group

- Influence the development of the RIPE Database software and operations
- Participate in the Database WG discussions!
- https://www.ripe.net/community/wg/active-wg/db/



More RIPE Database Resources

- The RIPE Database page on ripe.net
  - https://docs.db.ripe.net/
- Other RIPE Database query methods
  - https://docs.db.ripe.net/How-to-Query-the-RIPE-Database/
- Update Methods
  - https://docs.db.ripe.net/Update-Methods/

Questions ?

Play Time!

Practice What You Learned

Choose Your Own Adventure

- From the **Play Time!** list of tasks, choose what you would like to practice
- Review the course slides and your own notes
- Ask the trainers or other participants to assist, if you need help

Beyond The Database

The RESTful API

Problem Statement

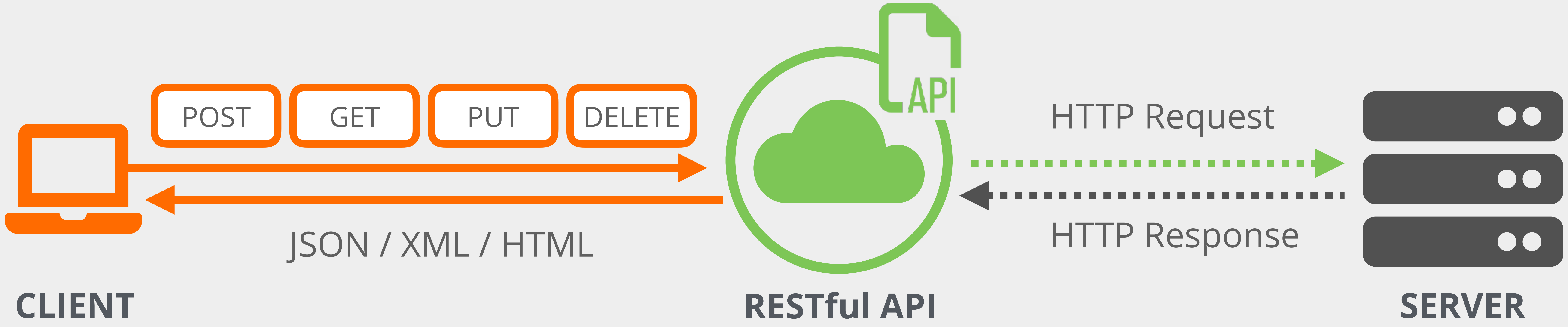
- Your company has a provisioning software that **assigns** address blocks to customers from a pool
- The RIPE policies require you to **register** these blocks with contact data in the RIPE Database
- Can you **save time** by letting the software create the required objects in the RIPE Database?

Why Use the RIPE Database REST API?

- **Automate** IP address block registration with RIPE
- **Integrate** network resource management into your systems
- **Update contact information** across multiple objects
- Ensure **compliance with RIPE policies** automatically
- **Reduce** manual data entry **errors** and save time

What is a REST API?

- REST = Representational State Transfer
- A standardised way for computer systems to communicate over HTTP
- Uses familiar web concepts (URLs, HTTP methods)



Understanding the REST API Structure

URI Format: https://{API endpoint}/{source}/{objecttype}/{key}

Environment	API Endpoint
RIPE	rest.db.ripe.net
TEST	rest-test.db.ripe.net

{source}: Which database to use

- **ripe**: Production RIPE Database
- **test**: Test Database

Understanding the RIPE Database API Structure

URI Format: https://{API endpoint}/{source}/{objecttype}/{key}

{objecttype}: What kind of resource

- person, role, organisation
- inet6num, inetnum (IP address blocks)
- route, aut-num, domain, mntner, etc.

{key}: Unique identifier for the object

- For IP blocks: The IP range (e.g., "193.0.0.0 - 193.0.7.255")
- For contacts: The NIC handle (e.g., "ABC123-RIPE")

REST API Operations

The RIPE Database REST API uses standard HTTP methods:

GET: Retrieve an object

- Use to look up information
- Example: Get details about an IP block

POST: Create a new object

- Use to register new resources
- Example: Register a new IP assignment

PUT: Update an existing object

- Use to modify resource information
- Example: Update contact details

DELETE: Remove an object

- Use to deregister resources
- Example: Remove an obsolete IP block registration

Some operations require appropriate authentication!

Practical Example: Looking Up an IP Block

Task: Find information about an IP block (193.0.0.0 - 193.0.7.255)

- HTTP Method: GET (retrieve)
- URI: https://rest.db.ripe.net/ripe/inetnum/193.0.0.0%20-%20193.0.7.255

Terminal command:

```
curl 'https://rest.db.ripe.net/ripe/inetnum/193.0.0.0%20-%20193.0.7.255'
```

The output will show the database object in the default XML format.

Add '.json' to the URI to get the object in JSON format.

Authentication and Security

- RIPE Database operations require proper authentication
- Write operations (create/update/delete) require an API key or a client certificate
- API keys can be created for all maintainers with a specific Access account or limited to one specific maintainer
- Authentication is provided through the HTTP header

Example:

```
curl -v --header "Accept: application/xml" \
    --header "Authorization: Basic {API KEY}“ \
    'https://rest.db.ripe.net/ripe/mntner/example-mnt?unfiltered'
```

Creating a New Object

Example: Register a new IPv6 block

1. Prepare the data (in XML or JSON)
2. Use HTTP POST method to the appropriate endpoint
3. Include authentication

Example:

```
curl -X POST --header "Accept: application/xml" \
    --header "Authorization: Basic {API KEY}“ \
    --data @myipv6.xml ‘https://rest.db.ripe.net/ripe/inet6num'
```

Updating an Existing Object

Example: Update an existing IPv6 block

1. Prepare the data with modifications (in XML or JSON)
2. Use PUT method to the specific API endpoint
3. Include authentication

Example:

```
curl -X PUT --header "Accept: application/xml"\
    --header "Authorization: Basic {API KEY}" \
    --data @myipv6-update.xml
    'https://rest.db.ripe.net/ripe/inet6num/2001:db8:1234::/48'
```

Deleting an Object

Example: Remove an IPv6 block registration

1. Identify the exact object to delete (source, type, key)
2. Use HTTP DELETE method to the specific object endpoint
3. Include authentication
4. Provide a reason for deletion

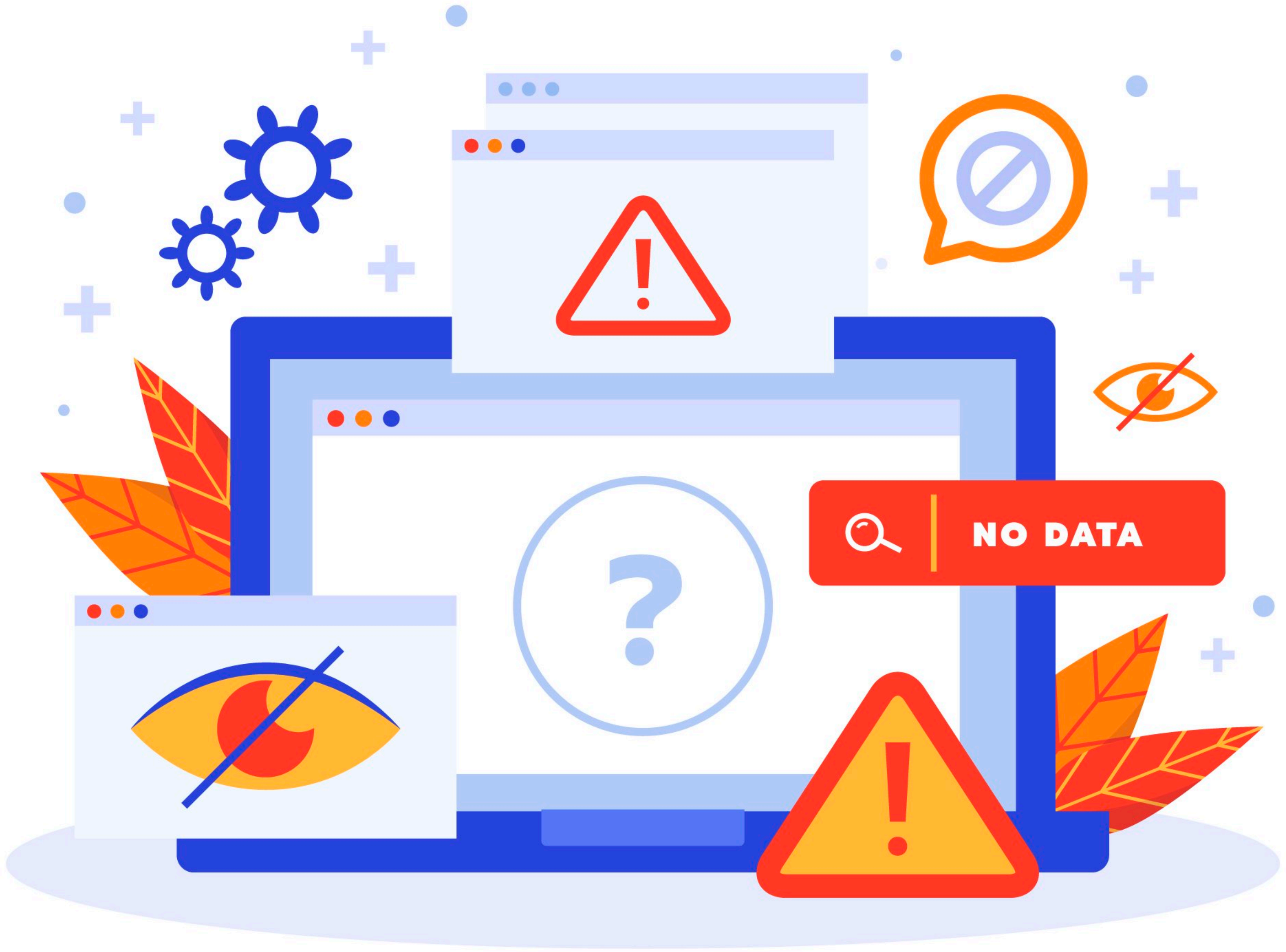
Example:

```
 curl -X DELETE --header "Accept: application/xml" \
    --header "Authorization: Basic {API KEY}" \
    'https://rest.db.ripe.net/ripe/inet6num/2001:db8:1234::/48?
    reason=address-range-no-longer-in-use'
```

Error Handling: Understanding API Responses

Common Error Scenarios:

- Syntax error in your request data
- Missing required attributes
- Authentication failure
- Object already exists
- Referenced objects don't exist



Error Handling: Understanding API Responses

HTTP Status	Code	Description
OK: Operation Successful	**200**	Everything went OK!
Bad Request	**400**	The service is unable to understand and process the request.
Forbidden	**403**	Query limit exceeded.
Not Found	**404**	No results were found (on a search request), or object specified in URI does not exist.
Conflict	**409**	Integrity constraint was violated (e.g. when creating, object already exists).
Internal Server Error	**500**	The server encountered an unexpected condition which prevented it from fulfilling the request.

Tip: Always check the response body for detailed error messages!

Additional Services

Search	RIPE database whois search service
Metadata	List available sources Object type template
Geolocation	Geolocation and language attributes for IPv4/IPv6 Address
Abuse Contact	Lookup abuse contact for Internet Resource

Examples

```
curl -H 'Accept: application/json' 'http://rest-test.db.ripe.net/search?source=test&query-string=tp19-test'
```

```
curl http://rest.db.ripe.net/metadata/templates/person.xml
```

```
curl http://rest.db.ripe.net/abuse-contact/AS3333
```

```
curl https://rest.db.ripe.net/geolocation?ipkey=10.0.0.0
```

References

- RESTful API documentation:
  https://docs.db.ripe.net/Update-Methods/RESTful-API

- REST API Data model:
  https://docs.db.ripe.net/RIPE-Database-Structure/REST-API-Data-model/

- Using API Keys in the RIPE Database microlearning:
  https://academy.ripe.net/mod/scorm/view.php?id=913

Doing it for real!

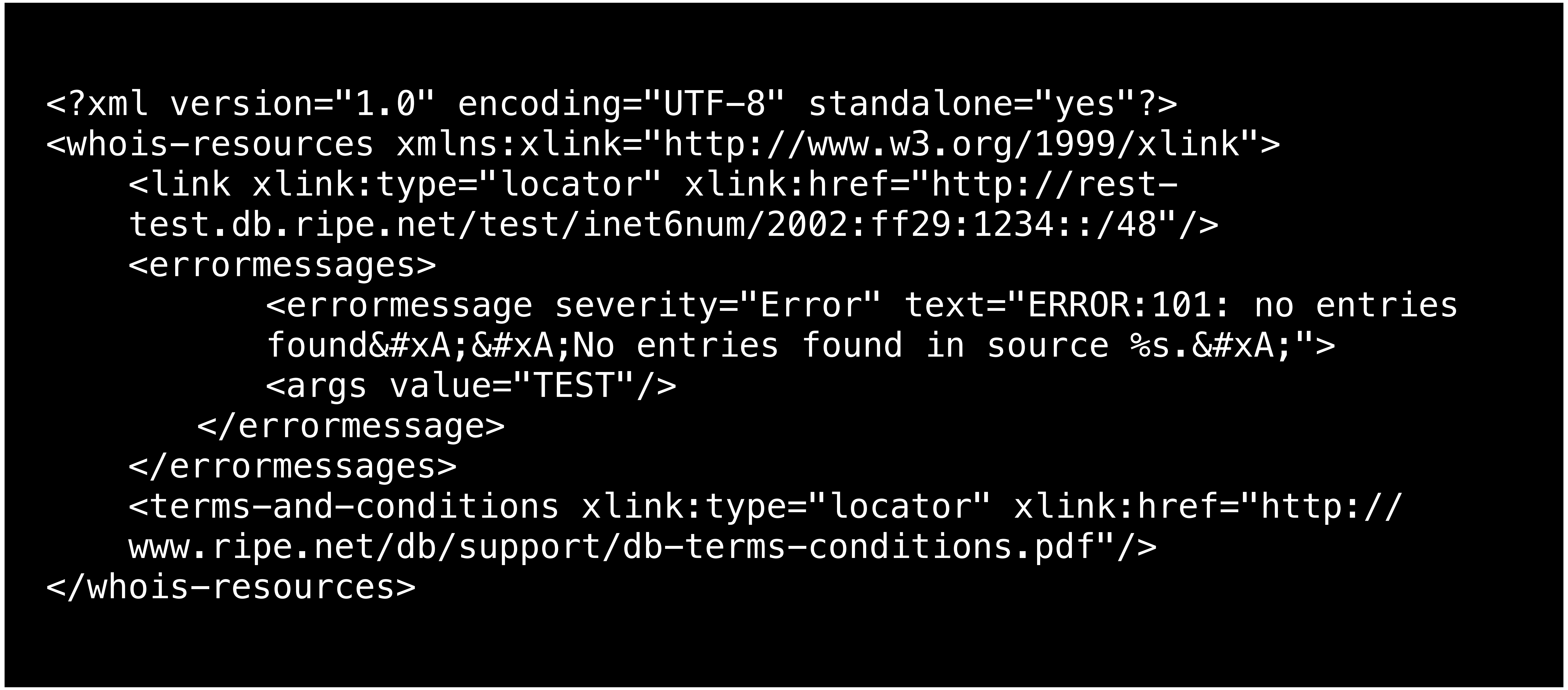
Demo

Create an INET6NUM object

TEST Database	Location: rest-test.db.ripe.net Source: test
Object Type	Type: inet6num (ASSIGNED)
Key	Key: 2002:ff29:1234::/48
Format	XML

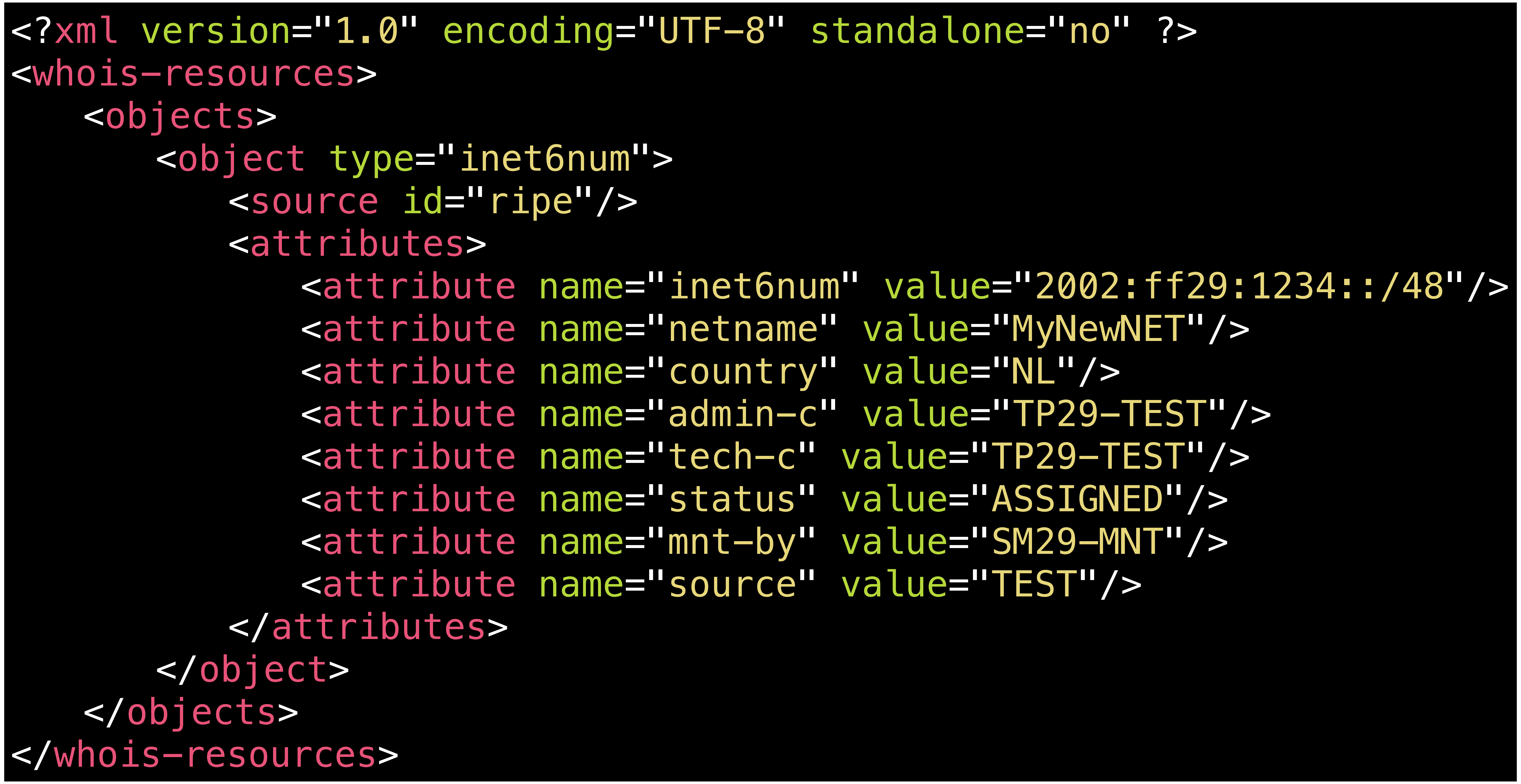
Query and Fail

curl 'http://rest-test.db.ripe.net/test/inet6num/2001:ff29:1234::/48'



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<whois-resources xmlns:xlink="http://www.w3.org/1999/xlink">
    <link xlink:type="locator" xlink:href="http://rest-
    test.db.ripe.net/test/inet6num/2002:ff29:1234::/48"/>
    <errormessages>
          <errormessage severity="Error" text="ERROR:101: no entries
          found&#xA;&#xA;No entries found in source %s.&#xA;">
          <args value="TEST"/>
       </errormessage>
    </errormessages>
    <terms-and-conditions xlink:type="locator" xlink:href="http://
    www.ripe.net/db/support/db-terms-conditions.pdf"/>
</whois-resources>
```

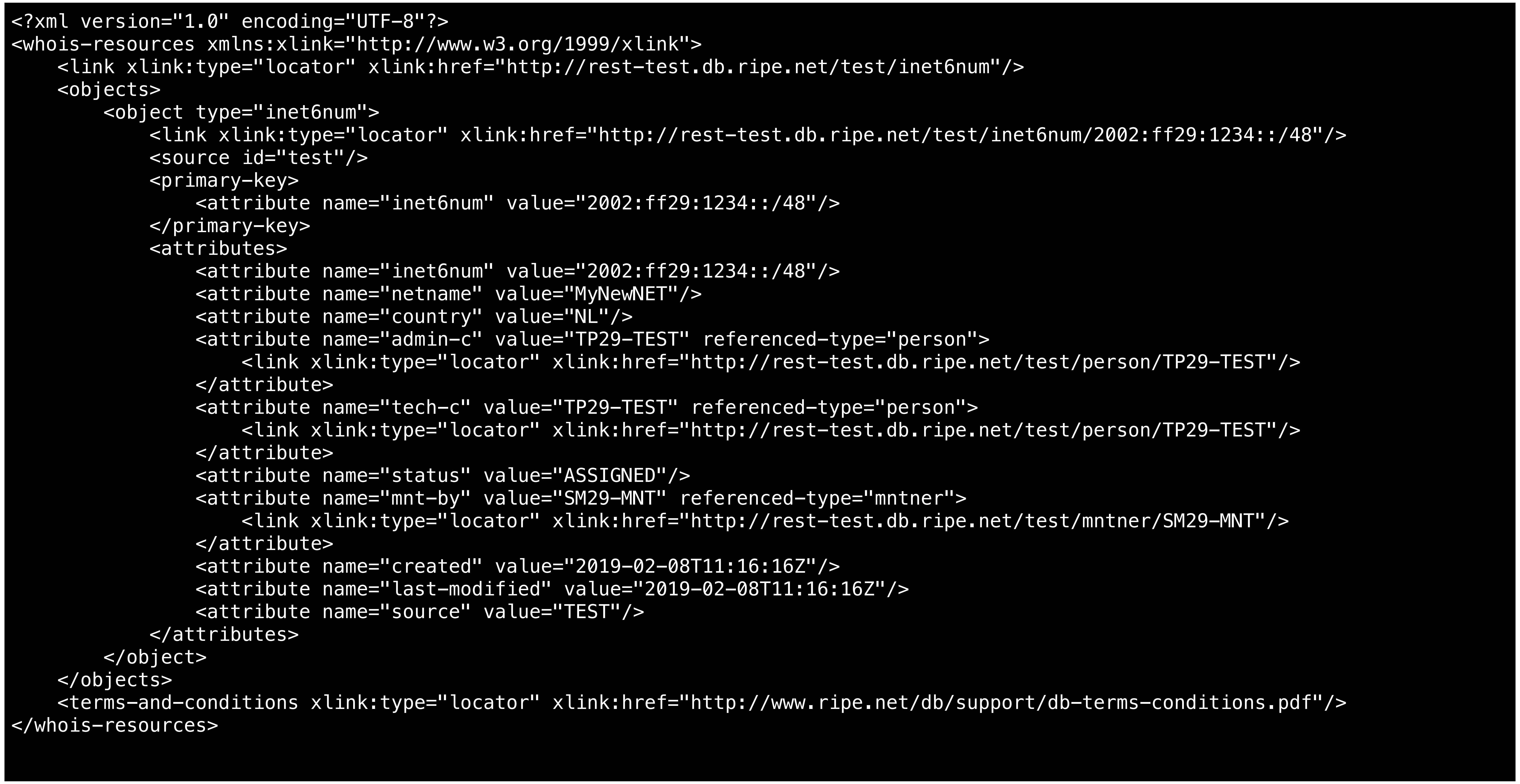
XML Template



```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<whois-resources>
    <objects>
        <object type="inet6num">
            <source id="ripe"/>
            <attributes>
                <attribute name="inet6num" value="2002:ff29:1234::/48"/>
                <attribute name="netname" value="MyNewNET"/>
                <attribute name="country" value="NL"/>
                <attribute name="admin-c" value="TP29-TEST"/>
                <attribute name="tech-c" value="TP29-TEST"/>
                <attribute name="status" value="ASSIGNED"/>
                <attribute name="mnt-by" value="SM29-MNT"/>
                <attribute name="source" value="TEST"/>
            </attributes>
        </object>
    </objects>
</whois-resources>
```

Create INET6NUM Object

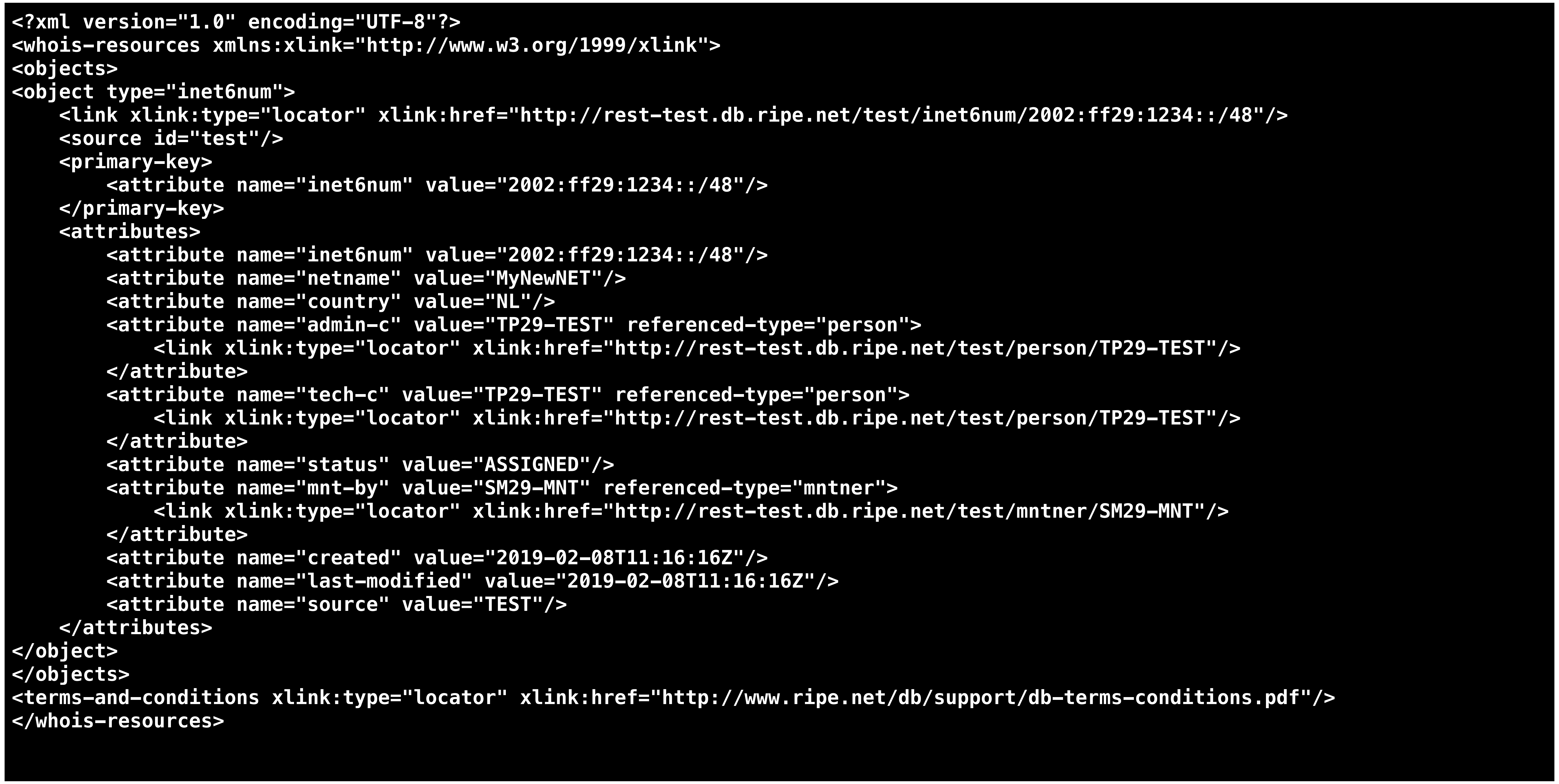
```
curl -X POST -H 'Content-Type: application/xml' --header "Authorization: Basic YOUR-API-KEY"
--data @my_object.xml 'https://rest-test.db.ripe.net/test/inet6num'
```



```
<?xml version="1.0" encoding="UTF-8"?>
<whois-resources xmlns:xlink="http://www.w3.org/1999/xlink">
    <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/inet6num"/>
    <objects>
        <object type="inet6num">
            <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/inet6num/2002:ff29:1234::/48"/>
            <source id="test"/>
            <primary-key>
                <attribute name="inet6num" value="2002:ff29:1234::/48"/>
            </primary-key>
            <attributes>
                <attribute name="inet6num" value="2002:ff29:1234::/48"/>
                <attribute name="netname" value="MyNewNET"/>
                <attribute name="country" value="NL"/>
                <attribute name="admin-c" value="TP29-TEST" referenced-type="person">
                    <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/person/TP29-TEST"/>
                </attribute>
                <attribute name="tech-c" value="TP29-TEST" referenced-type="person">
                    <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/person/TP29-TEST"/>
                </attribute>
                <attribute name="status" value="ASSIGNED"/>
                <attribute name="mnt-by" value="SM29-MNT" referenced-type="mntner">
                    <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/mntner/SM29-MNT"/>
                </attribute>
                <attribute name="created" value="2019-02-08T11:16:16Z"/>
                <attribute name="last-modified" value="2019-02-08T11:16:16Z"/>
                <attribute name="source" value="TEST"/>
            </attributes>
        </object>
    </objects>
    <terms-and-conditions xlink:type="locator" xlink:href="http://www.ripe.net/db/support/db-terms-conditions.pdf"/>
</whois-resources>
```

Query and Succeed!

curl 'http://rest-test.db.ripe.net/test/inet6num/2001:ff29:1234::/48'



```
<?xml version="1.0" encoding="UTF-8"?>
<whois-resources xmlns:xlink="http://www.w3.org/1999/xlink">
<objects>
<object type="inet6num">
    <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/inet6num/2002:ff29:1234::/48"/>
    <source id="test"/>
    <primary-key>
        <attribute name="inet6num" value="2002:ff29:1234::/48"/>
    </primary-key>
    <attributes>
        <attribute name="inet6num" value="2002:ff29:1234::/48"/>
        <attribute name="netname" value="MyNewNET"/>
        <attribute name="country" value="NL"/>
        <attribute name="admin-c" value="TP29-TEST" referenced-type="person">
            <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/person/TP29-TEST"/>
        </attribute>
        <attribute name="tech-c" value="TP29-TEST" referenced-type="person">
            <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/person/TP29-TEST"/>
        </attribute>
        <attribute name="status" value="ASSIGNED"/>
        <attribute name="mnt-by" value="SM29-MNT" referenced-type="mntner">
            <link xlink:type="locator" xlink:href="http://rest-test.db.ripe.net/test/mntner/SM29-MNT"/>
        </attribute>
        <attribute name="created" value="2019-02-08T11:16:16Z"/>
        <attribute name="last-modified" value="2019-02-08T11:16:16Z"/>
        <attribute name="source" value="TEST"/>
    </attributes>
</object>
</objects>
<terms-and-conditions xlink:type="locator" xlink:href="http://www.ripe.net/db/support/db-terms-conditions.pdf"/>
</whois-resources>
```

Questions

We want your feedback!

What did you think about this session? Take our survey at:

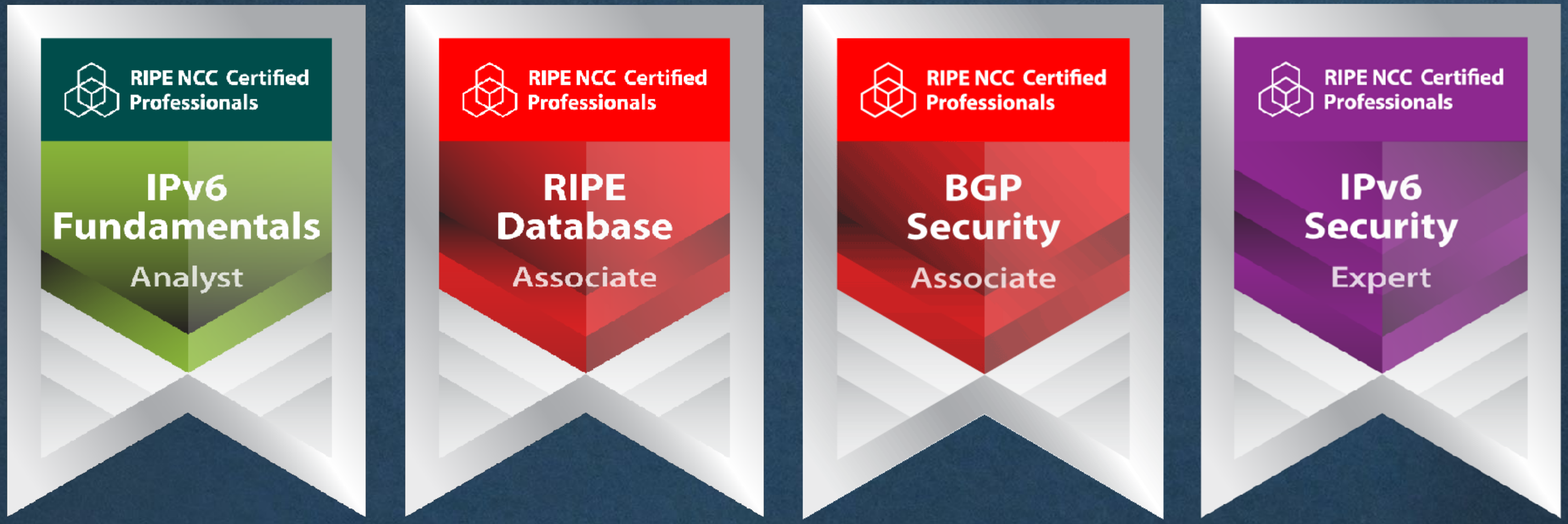
https://www.ripe.net/feedback/db

RIPE NCC
Academy

Learn something new today!

academy.ripe.net

RIPE NCC Certified Professionals


RIPE NCC Certified Professionals
IPv6 Fundamentals
Analyst
RIPE NCC Certified Professionals
RIPE Database
Associate
RIPE NCC Certified Professionals
BGP Security
Associate
RIPE NCC Certified Professionals
IPv6 Security
Expert

https://getcertified.ripe.net/

Have more questions? Ask us!

academy@ripe.net

Ënn

Соңы

An Críoch

پایان

Ende

Y Diwedd

Vége

Endir

Koniec

Finvezh

վերջ

Кінець

Son

დასასრული

Finis

הסוף

Tmiem

Liðugt

Lõpp

Amaia

Loppu

Крај

Slutt

Kraj

Sfârşit

النهاية

Конец

Konec

Fund

Fine

Fin

Fí

Einde

Край

Beigas

Τέλος

Fim

Slut

Pabaiga

What’s Next in Internet Registry

Webinars

Attend another webinar live wherever you are.

- LIRs and the Internet Ecosystem (2 hrs)
- LIRs: Managing IP Addresses and ASNs (2 hrs)
- Internet Governance (1 hr)
- Policy Development Process (1 hr)
- Webinar for New LIRs (1 hr)

Face-to-face

Meet us at a location near you for a training session delivered in person.

- LIR (8.5 hrs)
- RIPE Database (8.5 hrs)

For more info click the link below

learning.ripe.net

E-learning

Learn at your own pace at our online Academy.

- Internet Governance (3 hrs)
- RIPE Database (16 hrs)

For more info click the link below

academy.ripe.net

Examinations

Learnt everything you needed? Get certified!

- RIPE Database Associate

For more info click the link below

getcertified.ripe.net

Copyright Statement

[…]

The RIPE NCC Materials may be used for **private purposes, for public non-commercial purpose, for research, for educational or demonstration purposes**, or if the materials in question specifically state that use of the material is permissible, and provided the RIPE NCC Materials are not modified and are properly identified as RIPE NCC documents. Unless authorised by the RIPE NCC in writing, any use of the RIPE NCC Materials for advertising or marketing purposes is strictly forbidden and may be prosecuted. The RIPE NCC should be notified of any such activities or suspicions thereof.

[…]

Find the full copyright statement here:
https://www.ripe.net/about-us/legal/copyright-statement