RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Routing Security

Training Course

Training Services | RIPE NCC | November 2015

Schedule

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
- Number in the list
- Experience
  - BGP Routing
  - RIPE Database and Routing Registry
  - Resource Certification
- Goals

Overview

- Internet Routing Insecurity
- BGP and Routing Basics
- Introduction to the Routing Registry
  - Routing Policy Specification Language (RPSL)
  - RPSL in Practice
  - Tools and Automation
- Introduction to the Resource Certification
  - RPKI: Setting it up
  - RPKI: Using it. Relying Party’s side. Validation
  - RPKI: Router Integration

Internet Routing Insecurity

Section 1

The Importance of the Internet

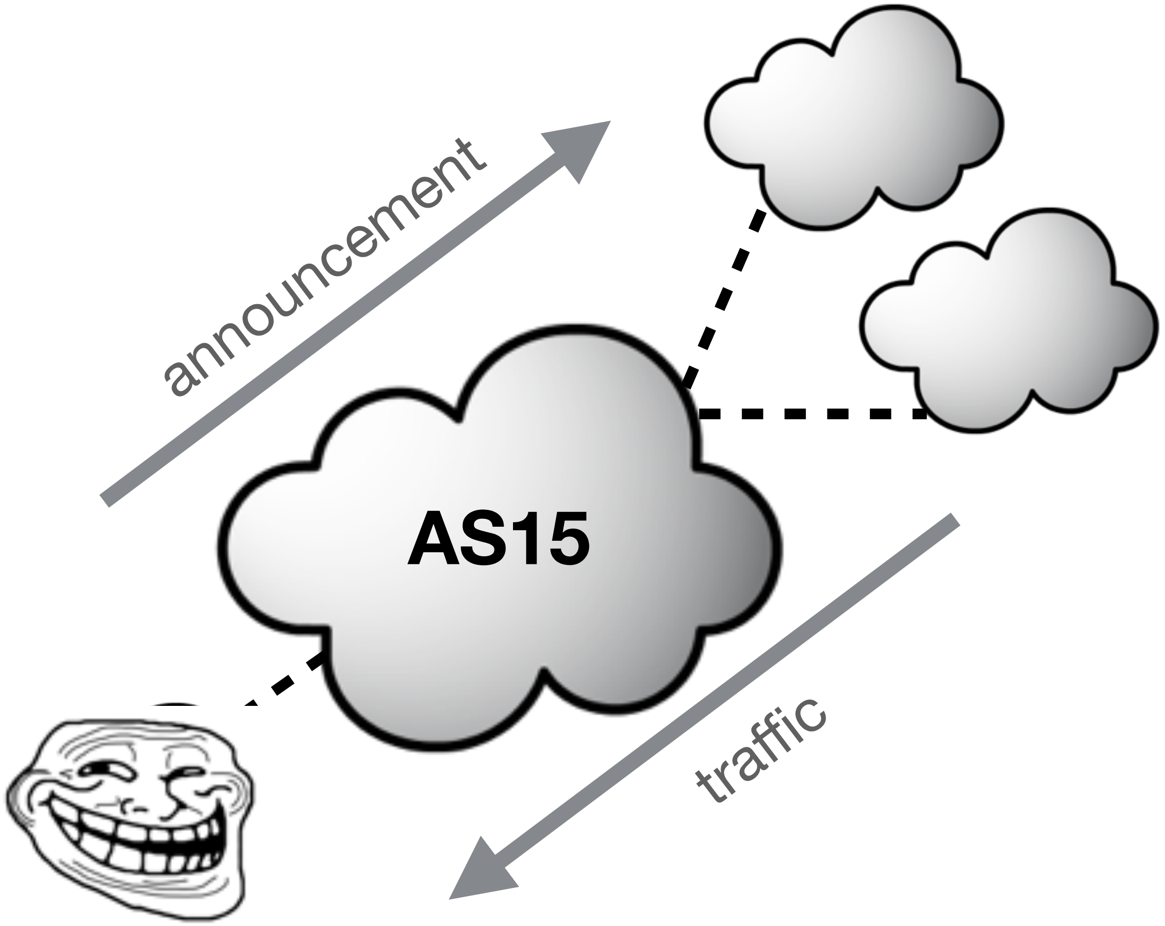
Internet has taken on an important role and facilitates nearly every aspect of modern life

- Communication
- Publishing
- Support
- Research

- Personal
- Commercial
- Governmental
- Internet of Things

Border Gateway Protocol 101

- **Individual networks (Autonomous Systems) identified by number (ASN) interconnect and announce prefixes to each other**



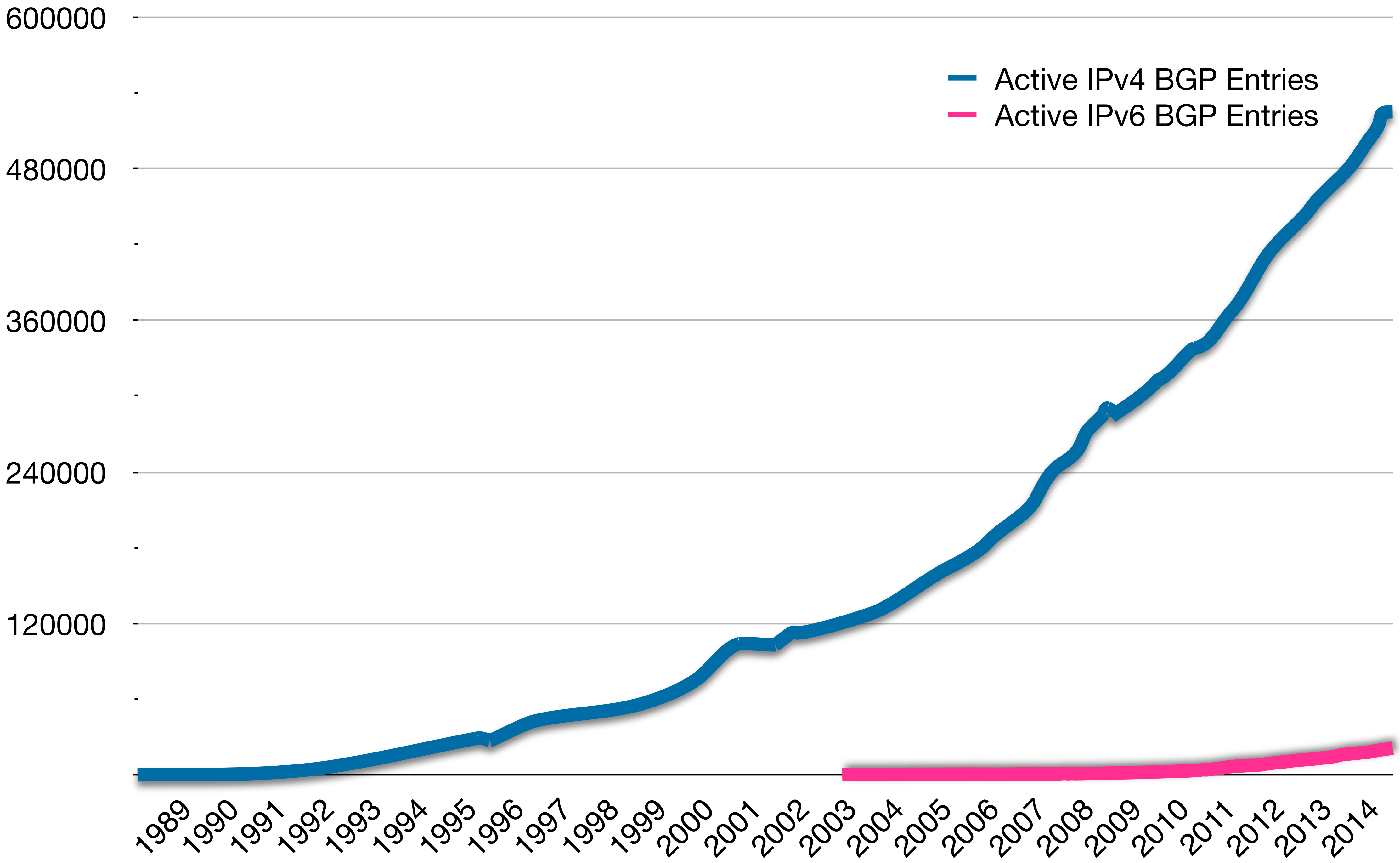
- No central "core"
- No "chain of trust" in IP allocation / assignment
- No association between ASN and IP

The State of The Global Routing

- Largely a trust-based system
  - Maximum prefix lists
  - Static prefix lists
  - IRR sourced
  - Often unfiltered
  - Often unauthenticated

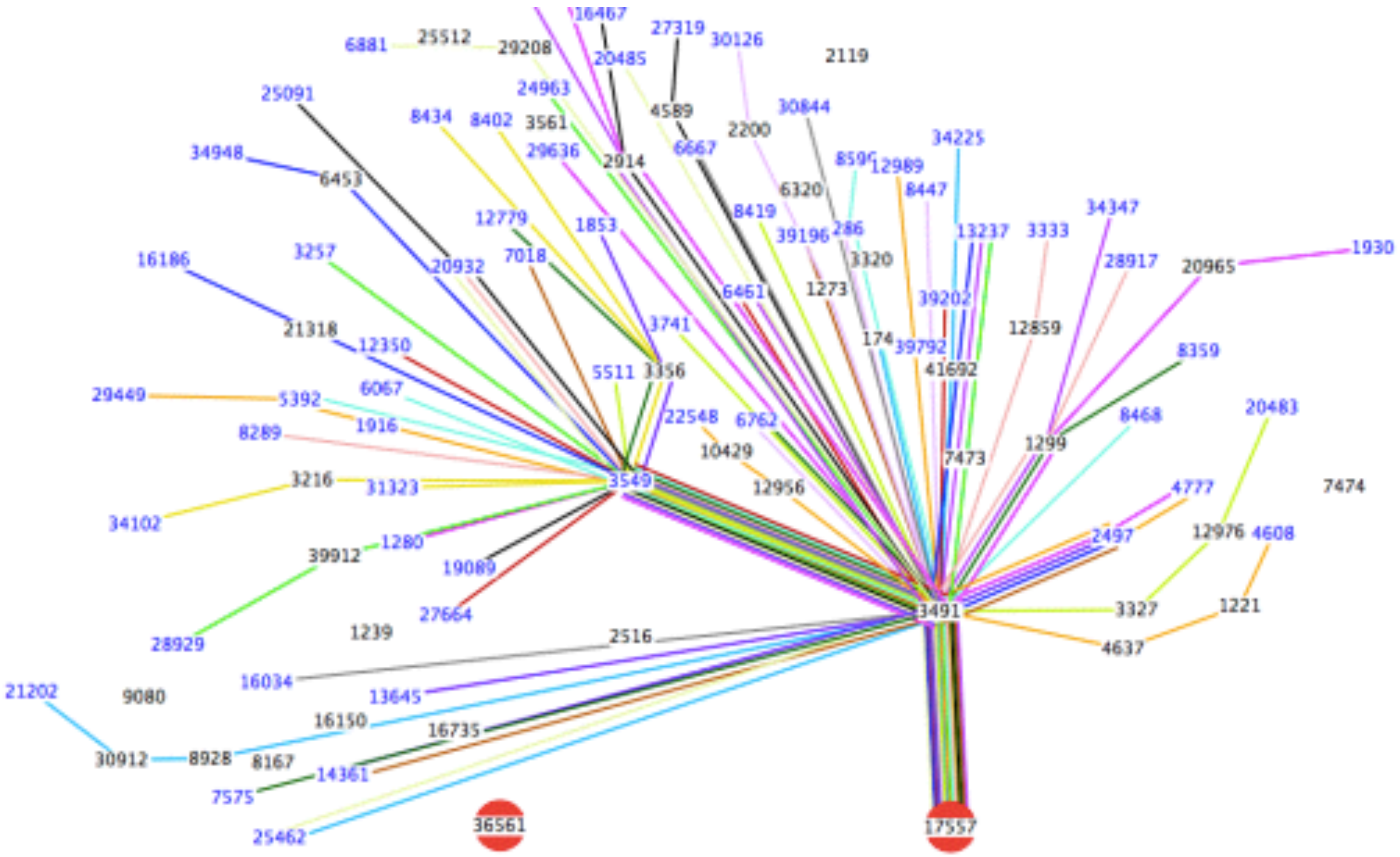
- Auditing is almost impossible

Global Routing Table Size

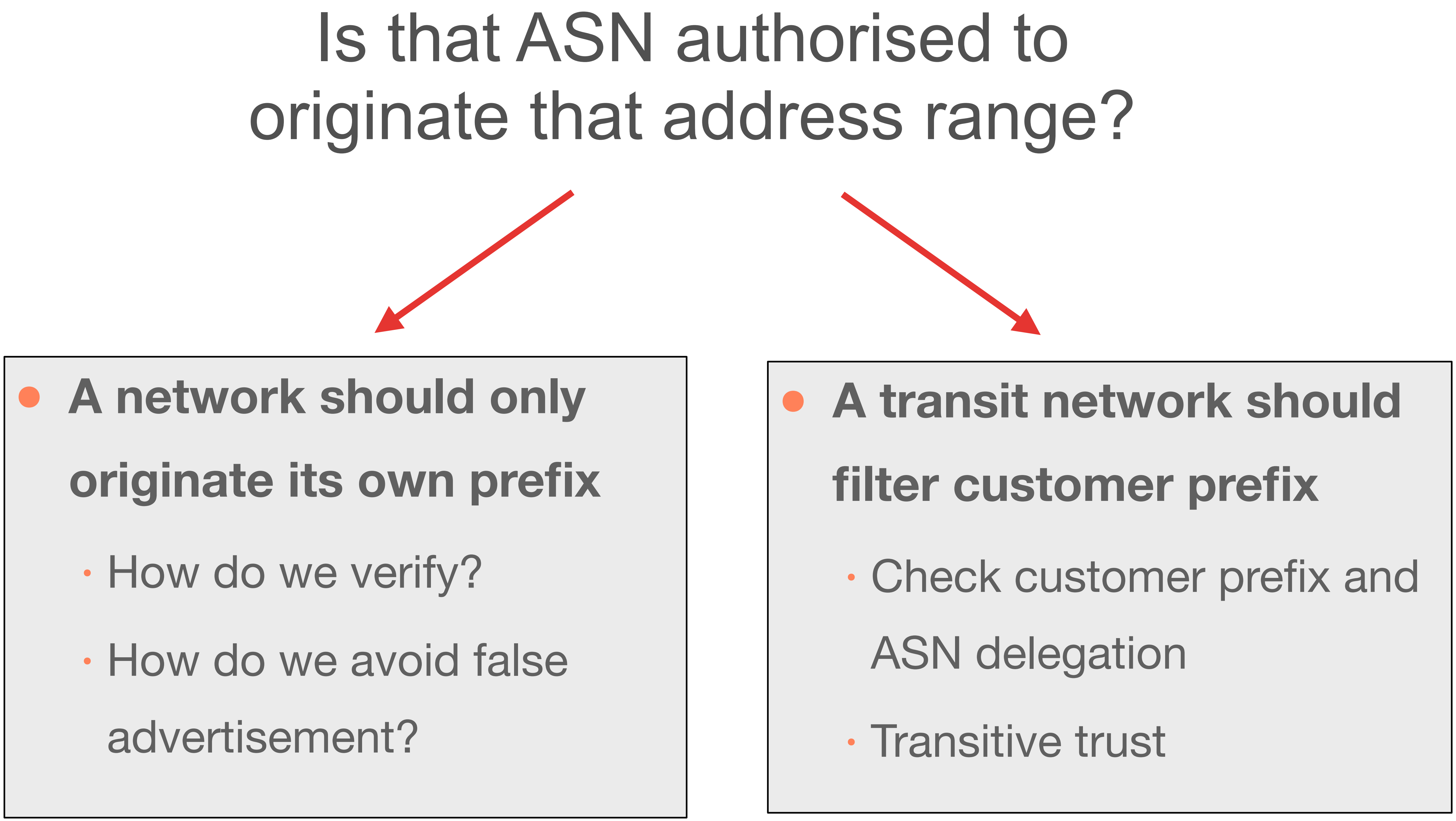


Routing Incidents Types

- Misconfiguration
  - No malicious intentions
  - Software bugs
- Malicious
  - Competition
  - Claiming “unused” space
- Targeted Traffic Misdirection
  - Collect and/or temper with data



Routing Incidents Mitigation



Is that ASN authorised to originate that address range?

- **A network should only originate its own prefix**
  - How do we verify?
  - How do we avoid false advertisement?

- **A transit network should filter customer prefix**
  - Check customer prefix and ASN delegation
  - Transitive trust

Origin Validation

- Organisation gets their resources from the RIR
  - Allocated resource is in RIR whois database
- Organisation notifies its upstream of the prefix to be announced
  - Usually email or phone
- Upstream must check the RIR whois database before accepting prefix
  - Need to be able to authoritatively prove who owns a prefix and which ASN may announce it

External Origin Validation Tools

- Internet Routing Registry
  - Public database viewable and parsable by anyone
  - Needs validation for publishing information

- Resource Public Key Infrastructure
  - Framework for automation
  - Integration with routers

End Goal: BGP Security (BGPsec)

- Extension to BGP
- Currently an IETF Internet draft
- Implemented via a new optional non-transitive BGP path attribute that contains a digital signature
- Features:
  - BGP Prefix Origin Validation (using RPKI)
  - BGP Path Validation

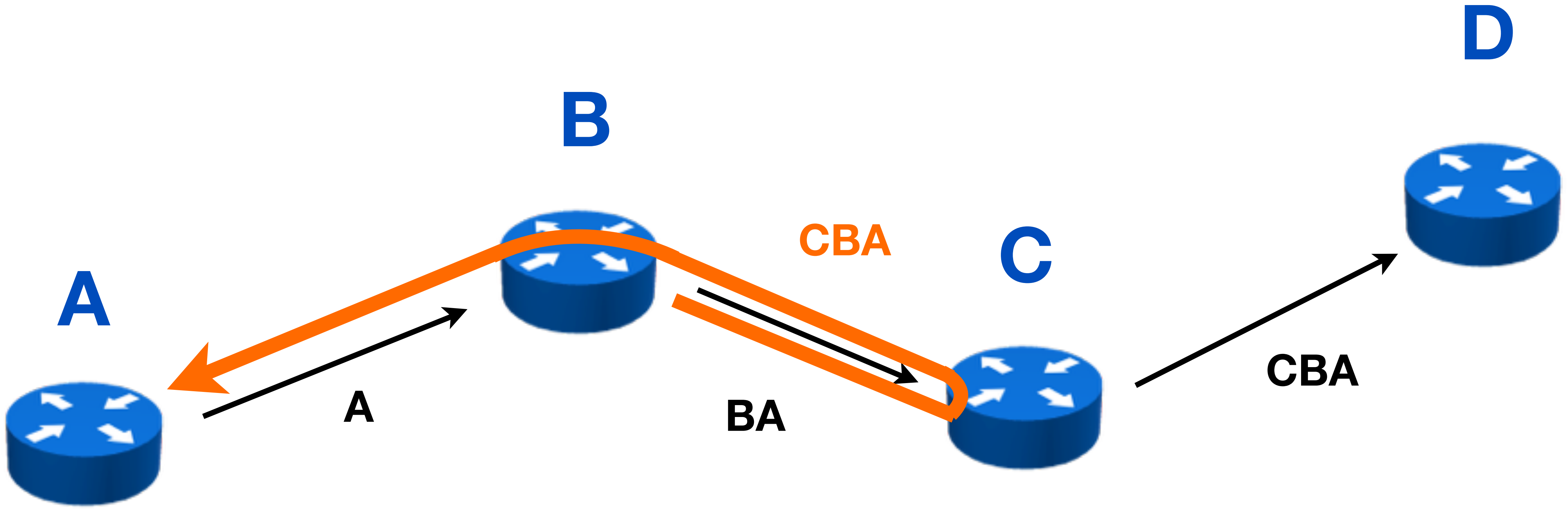
BGP and Routing Basics

Section 2

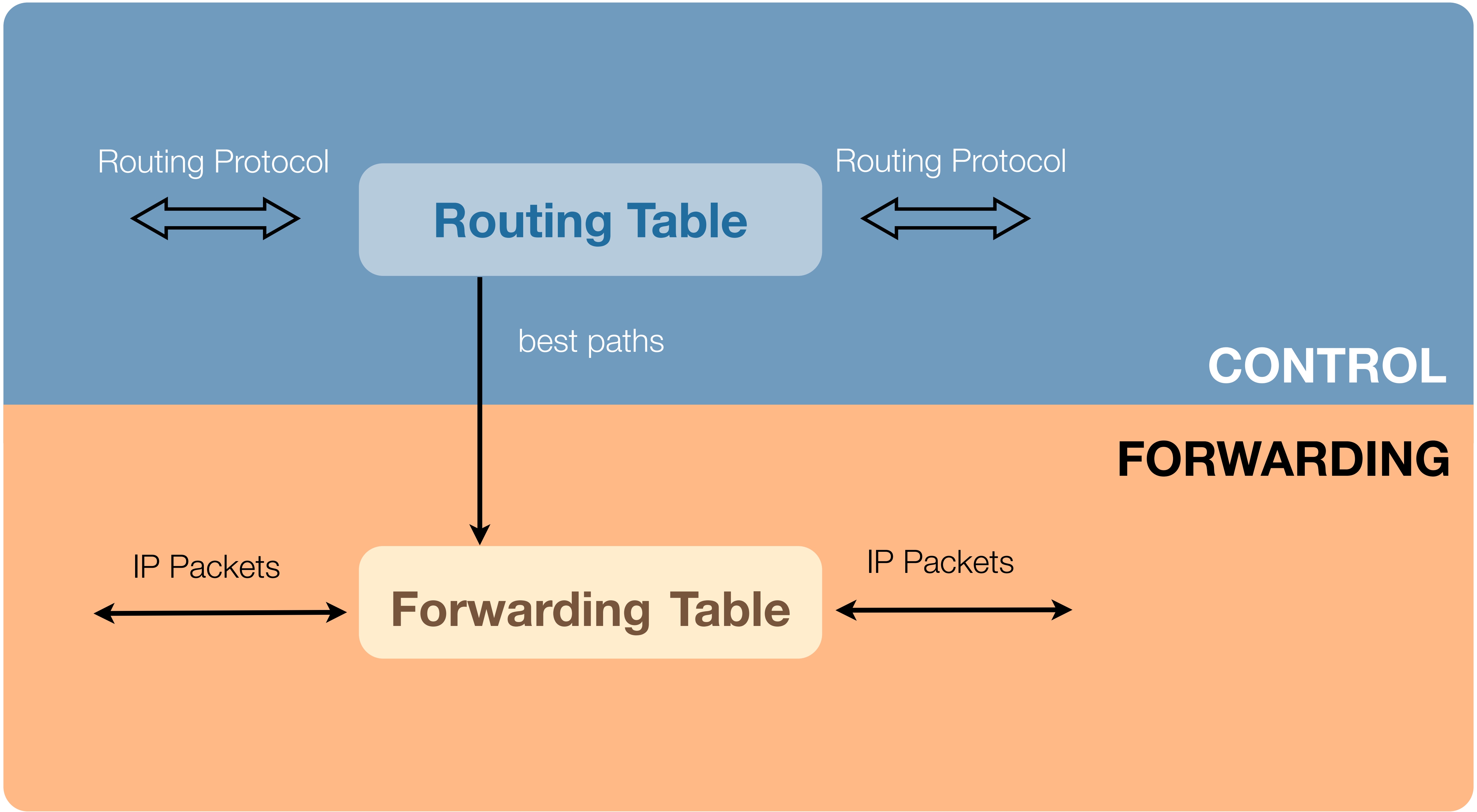
Border Gateway Protocol (BGP)

- The routing protocol of the Internet
- Routing between AS-es
- Uses AS Paths

AS-Path Prevents Loops



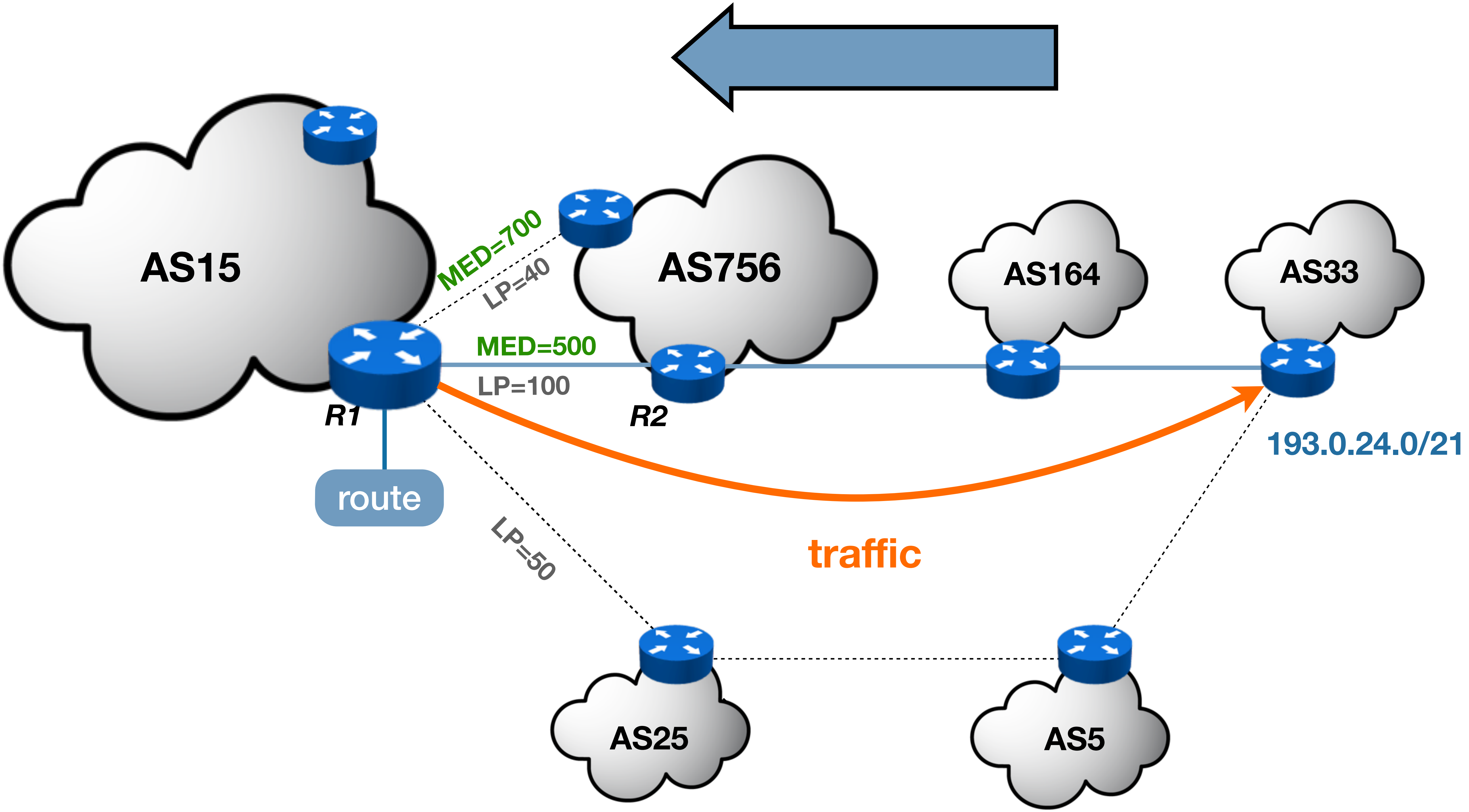
Control and Forwarding Planes



A Route and its Attributes

Prefix (NLRI)	next hop	MED	origin	weight	Local-pref	AS-path	communities	
66.2.9.0/23	95.3.12.68	500	IGP	200	100	756 164 33	756:205 337:52	...

Route Propagation



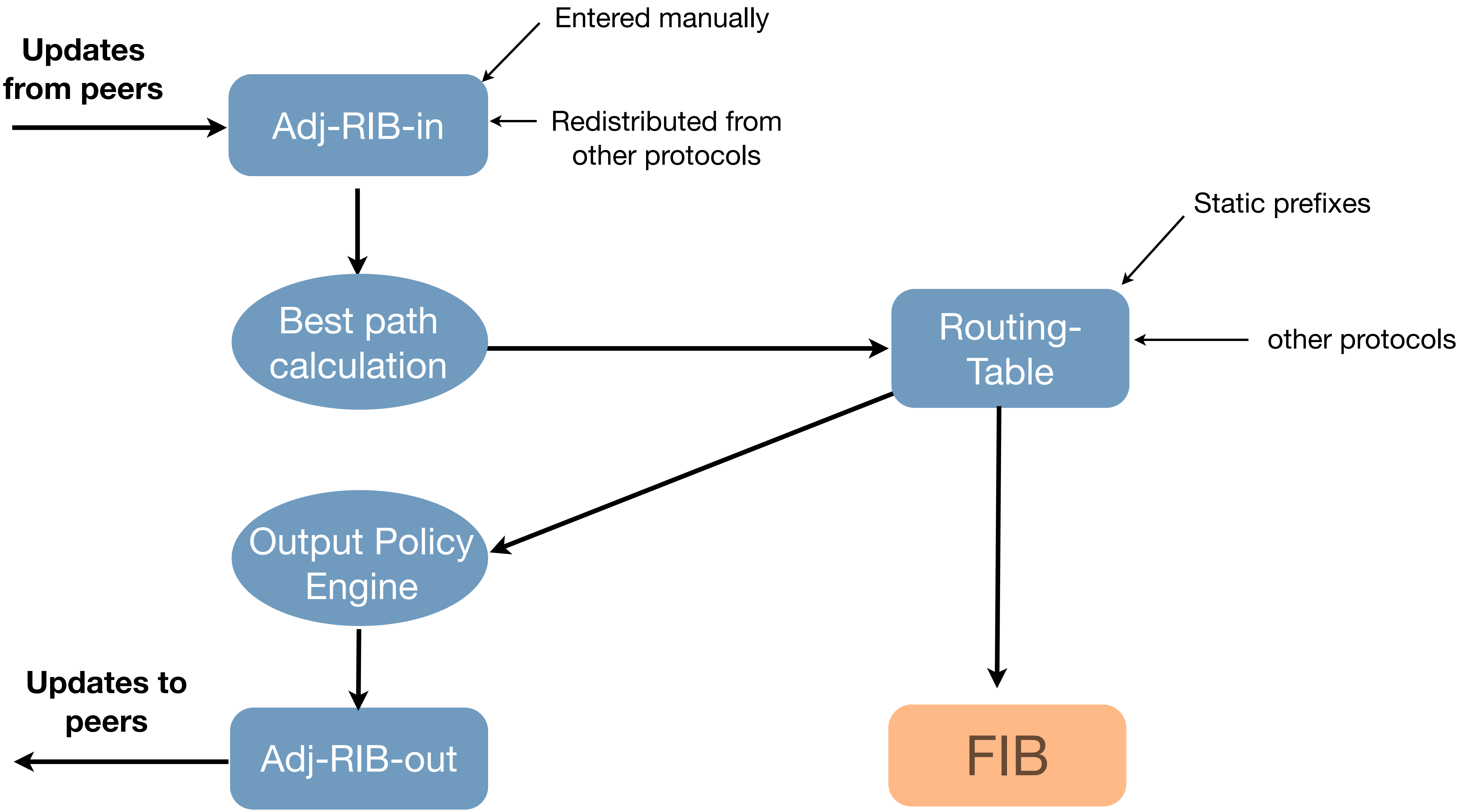
Route Attributes Limited To

Router:	Local AS:	local AS + neighbour:	not limited:
weight	local-pref updated: Next-hop AS-Path	MED	origin communities

Update Messages

- Withdrawn prefixes
- New prefixes
  - with attributes
- Also Keep-alive messages

Routing Tables in a Router



Updates from peers

Adj-RIB-in

Entered manually

Redistributed from other protocols

Best path calculation

Routing-Table

Static prefixes

other protocols

Output Policy Engine

FIB

Updates to peers

Adj-RIB-out

Adj-RIB-In

Prefix	Next Hop	MED	Origin	Weight	Local Pref	AS-Path	Communities	...
66.249.0.0/16	92.65.185.42	0	IGP	0	100	203 89 151		
66.249.0.0/16	98.3.23.146	0	IGP	0	100	34 151	34:102 34:123	
66.249.0.0/16	91.67.47.102	100	IGP	0	100	456 1436 151	456:30 1436:78	
66.249.0.0/20	95.23.129.30	0	IGP	100	40	2344 151		
198.45.16.0/21	81.23.45.2	500	IGP	0	100	3456 2119 8289		
198.45.16.0/21	84.5.167.85	0	IGP	0	80	4561 2356 8289	4561:180 2356:90	
198.45.16.0/20	82.46.10.182	40	IGP	0	200	341 8289		
213.4.78.0/23	85.196.44.23	0	IGP	0	20	7895 1299		
...	...	...	...	...	...	...		...

BGP Entries in the Routing-Table

Prefix	Next Hop	MED	Origin	Weight	Local Pref	AS-Path	Communities	...
66.249.0.0/16	98.3.23.146	0	IGP	0	100	34 151	34:102 34:123	
66.249.0.0/20	95.23.129.30	0	IGP	100	40	2344 151		
198.45.16.0/21	81.23.45.2	500	IGP	0	100	3456 2119 8289		
198.45.16.0/20	82.46.10.182	40	IGP	0	200	341 8289		
213.4.78.0/23	85.196.44.23	0	IGP	0	20	7895 1299		
...	...	...	...	...	...	...		...

FIB - Forwarding Table

Prefix	Interface
66.249.0.0/16	2
66.249.0.0/20	4
198.45.16.0/21	1
198.45.16.0/20	3
213.4.78.0/23	5
...	...

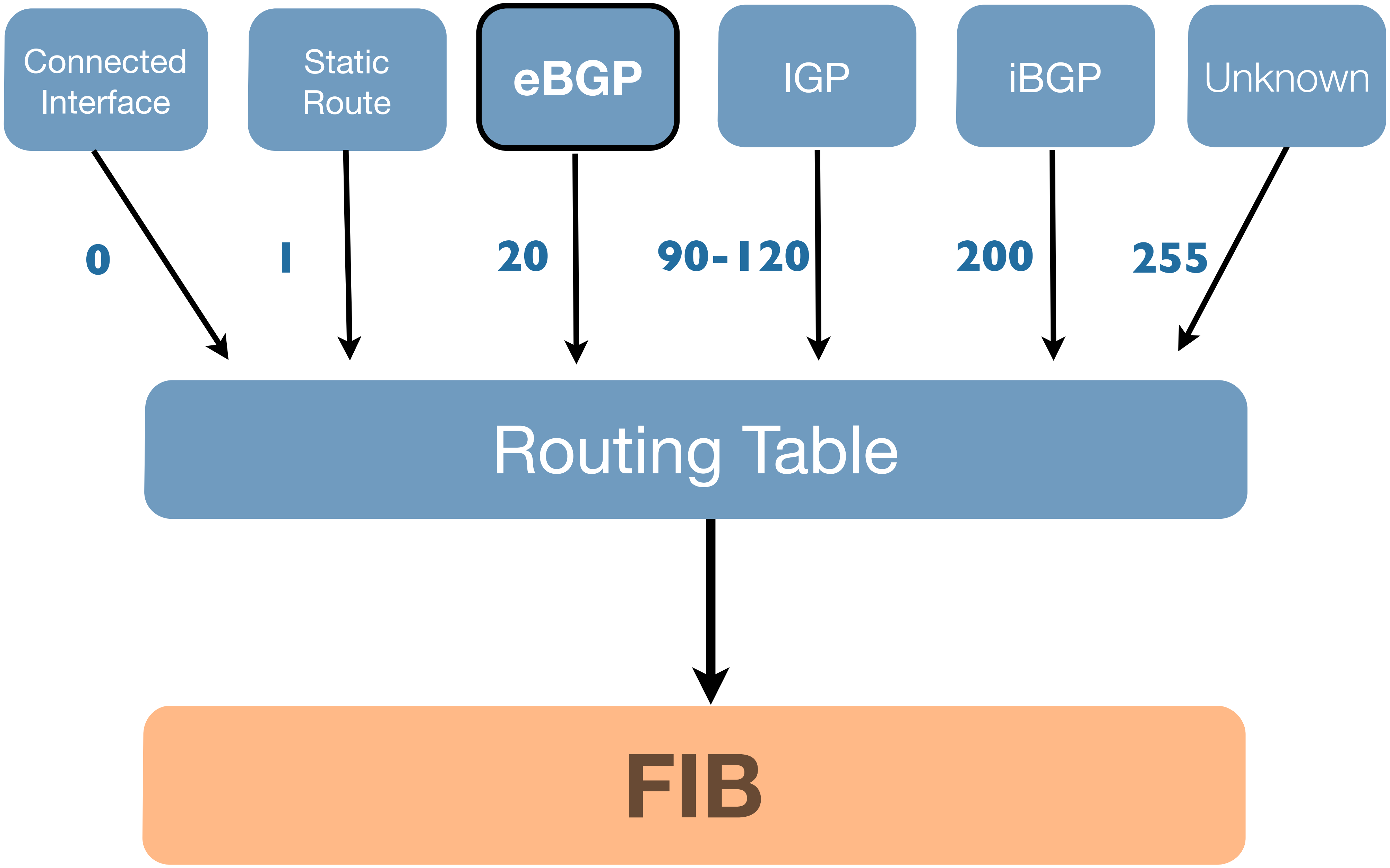
Best Path Calculation

- Drop if own AS in AS-Path
- Prefer path with highest Weight
- Highest Local Preference
- Shortest AS-Path
- Lowest MED

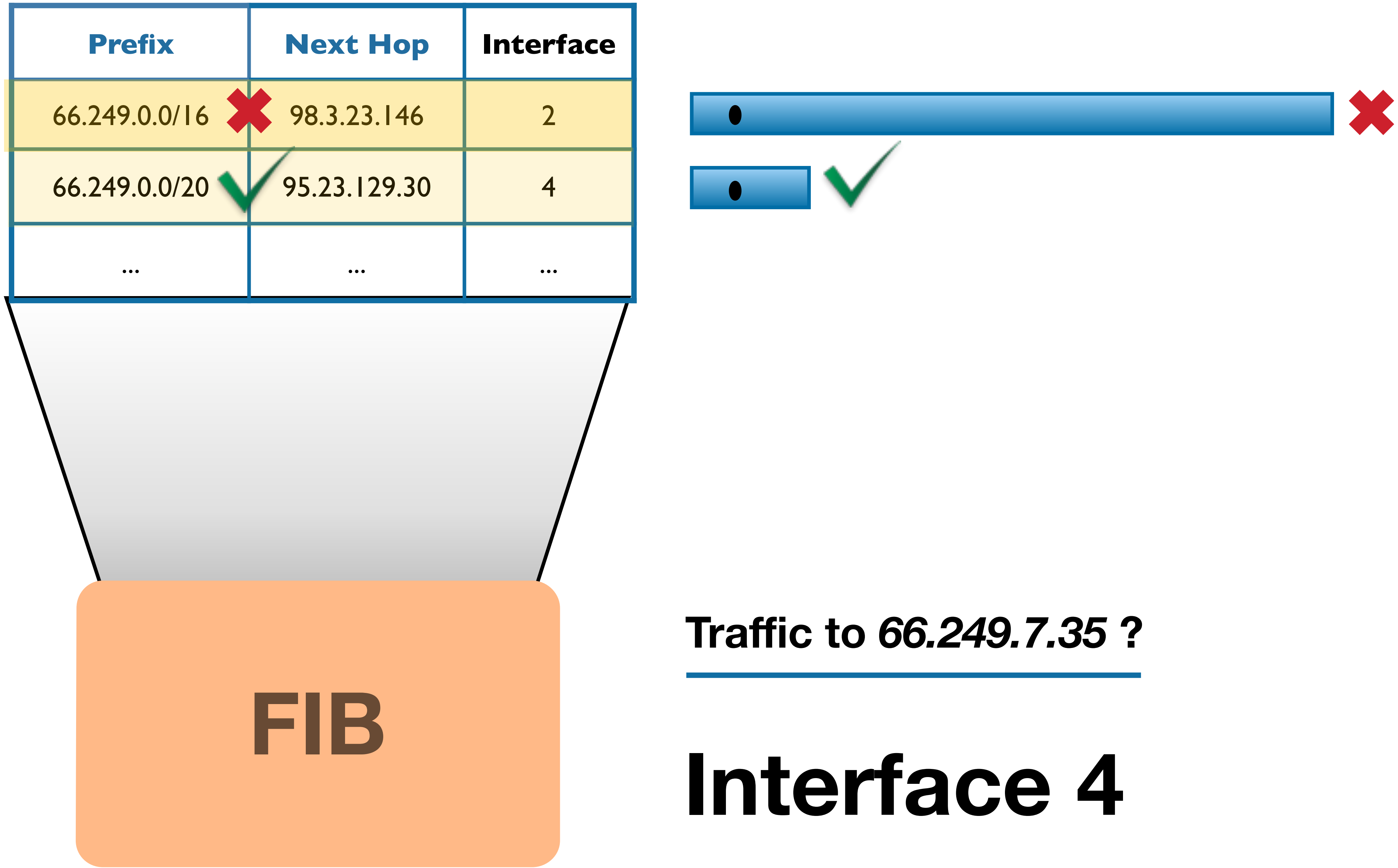
Best Path Calculation - Tiebreakers

- Path with shortest next hop metric (minimum IGP cost)
- Oldest received path
- Path from lowest neighbour address

Administrative Distance



More Specific Wins



Prefix	Next Hop	Interface
66.249.0.0/16 ✗	98.3.23.146	2
66.249.0.0/20 ✓	95.23.129.30	4
...	...	...

Traffic to *66.249.7.35* ?

Interface 4

Introduction to the Routing Registry

Section 3

Why Routing Registry ?

To be able to answer the question:

Is that ASN authorised to originate that address range?

Internet Routing Registry

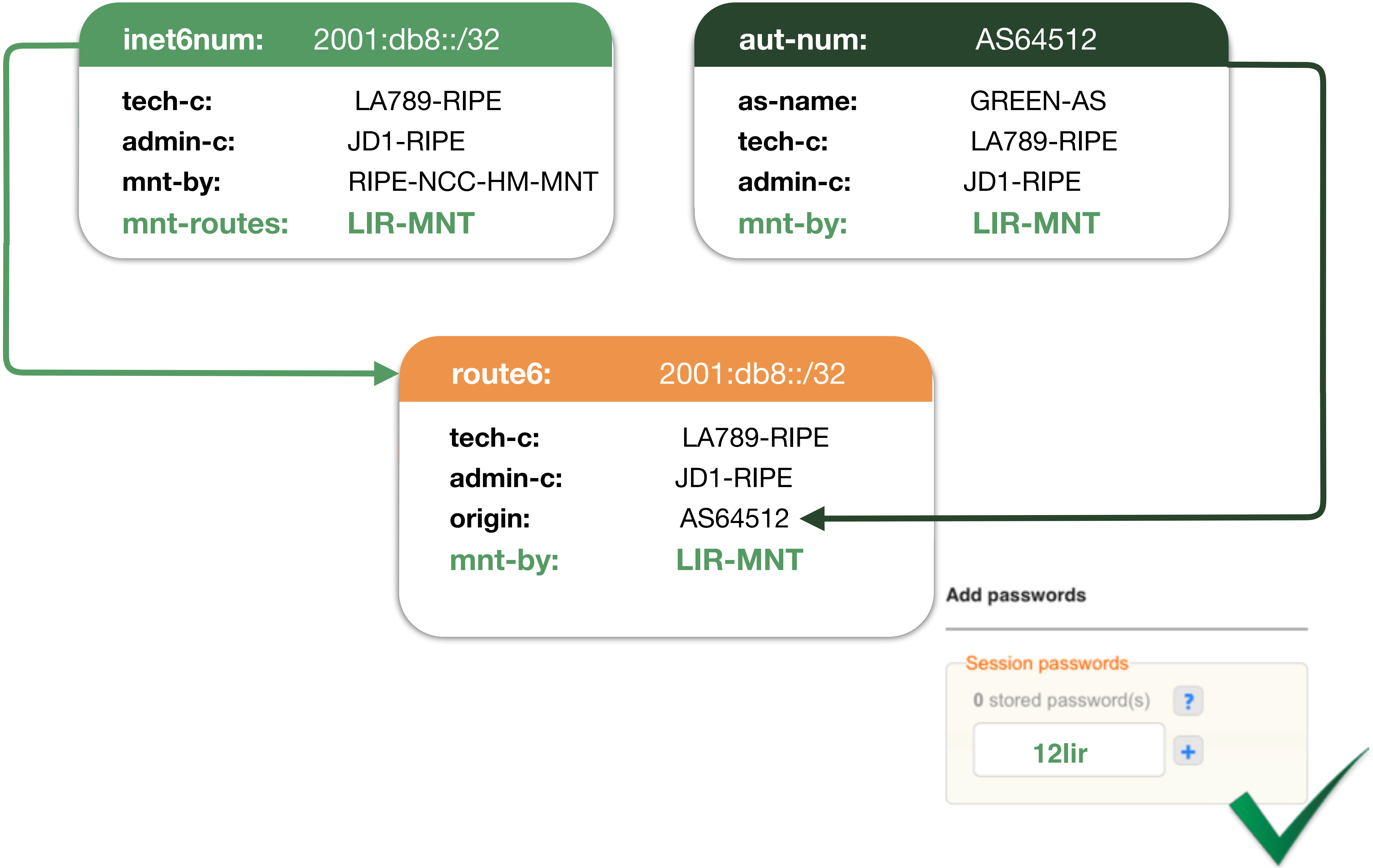
- Number of public databases that contain routing policy information which mirror each other:
  - RIPE, APNIC, RADB, JPIRR, Level3, …
  - http://www.irr.net
- RIPE NCC operates the RIPE Routing Registry
  - Part of the RIPE Database
  - Part of the Internet Routing Registry

RIPE Database Objects

- inetnum ➡ IPv4 address range
- inet6num ➡ IPv6 address range
- **aut-num** ➡ single AS number and routing policy
- **route, route6** ➡ glue between IP address range and an AS number announcing it

- person ➡ contact info for other objects
- role ➡ group of person objects
- maintainer ➡ protects all other objects

Registering Routes



```
inet6num:     2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-routes:   LIR-MNT
```

```
aut-num:      AS64512
as-name:      GREEN-AS
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       LIR-MNT
```

```
route6:       2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
origin:       AS64512
mnt-by:       LIR-MNT
```

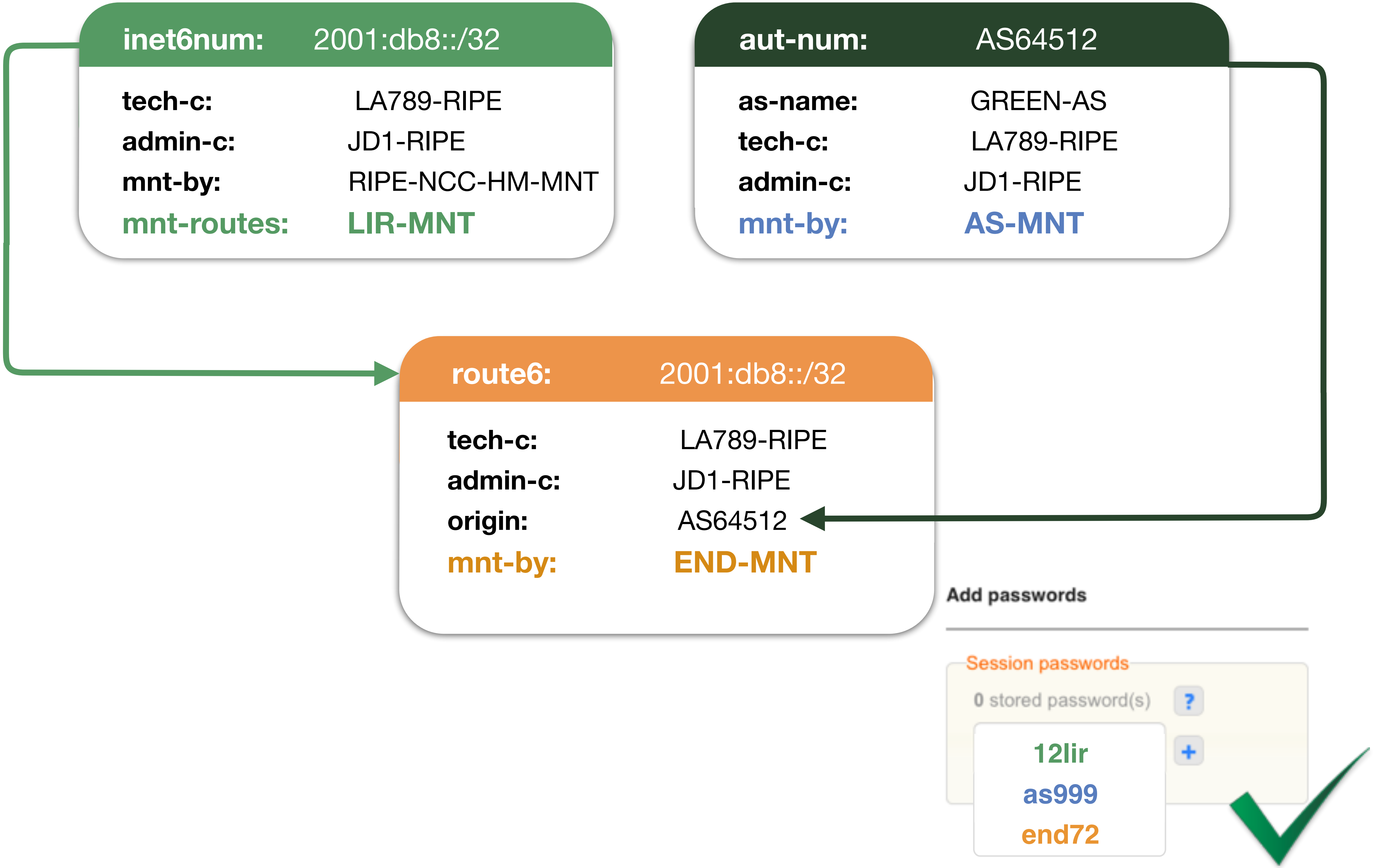
Add passwords

Session passwords

0 stored password(s)

12lir

Registering Routes



```
inet6num:     2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-routes:   LIR-MNT
```

```
aut-num:      AS64512
as-name:      GREEN-AS
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       AS-MNT
```

```
route6:       2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
origin:       AS64512
mnt-by:       END-MNT
```

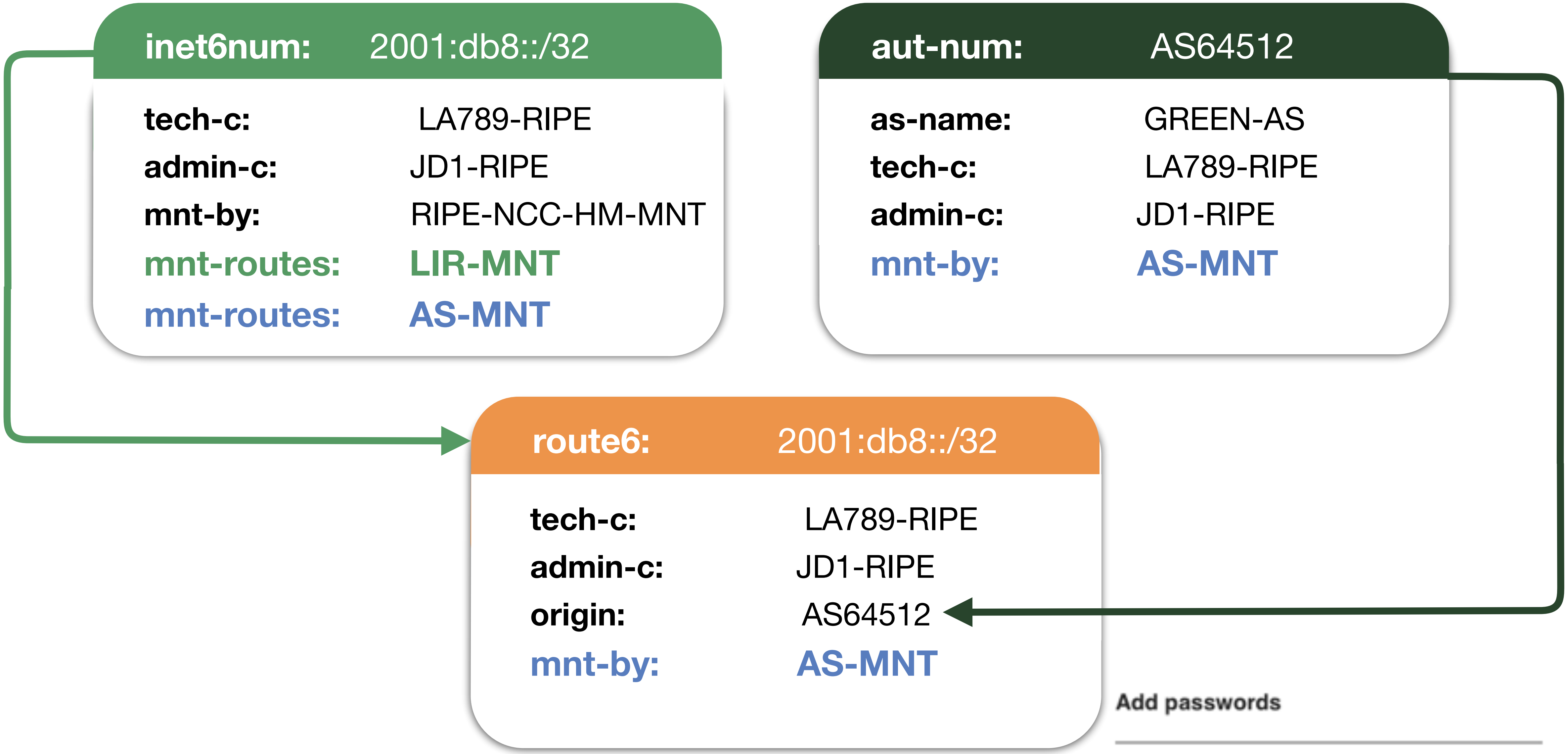
Add passwords

Session passwords

0 stored password(s)

12lir
as999
end72

Registering Routes

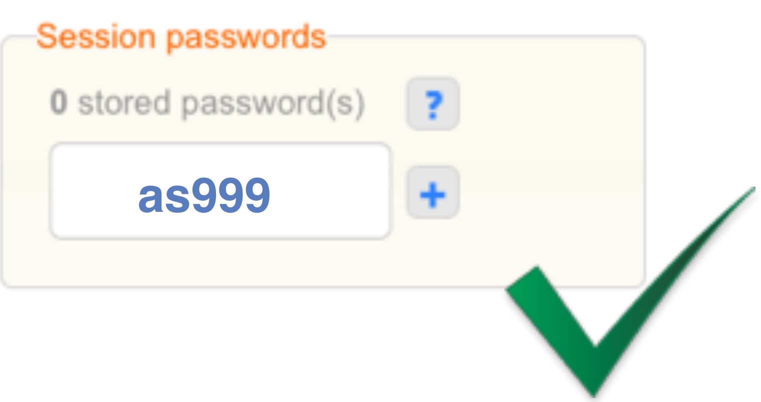


```
inet6num:     2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-routes:   LIR-MNT
mnt-routes:   AS-MNT
```

```
aut-num:      AS64512
as-name:      GREEN-AS
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       AS-MNT
```

```
route6:       2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
origin:       AS64512
mnt-by:       AS-MNT
```

Add passwords

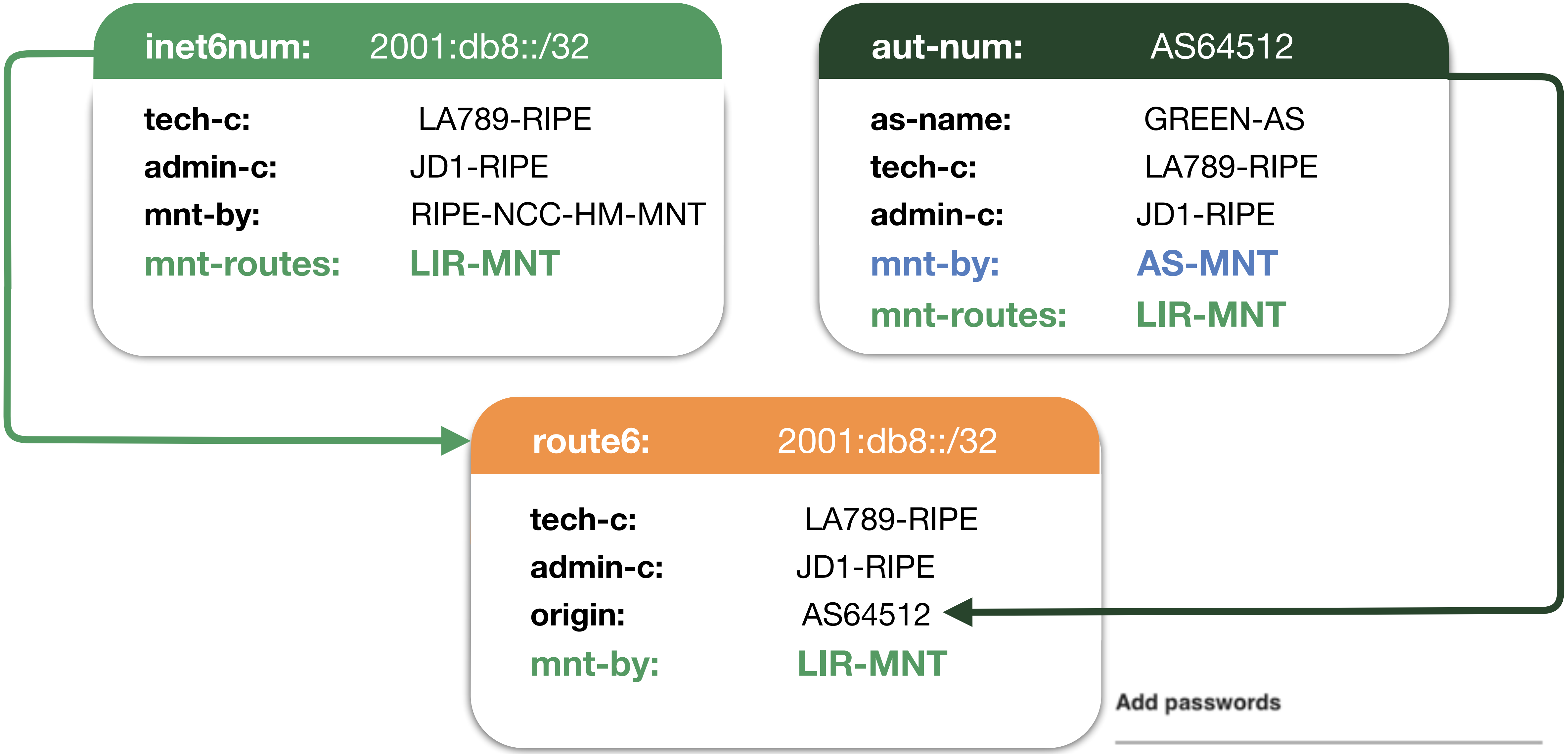


Session passwords

0 stored password(s) ?

as999 +

Registering Routes

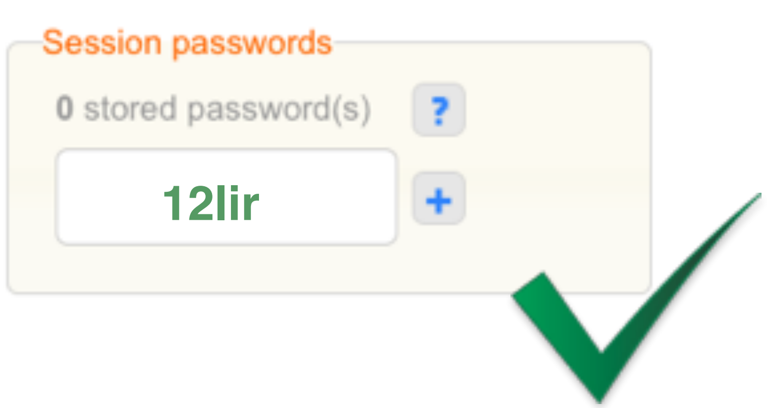


inet6num: 2001:db8::/32
tech-c: LA789-RIPE
admin-c: JD1-RIPE
mnt-by: RIPE-NCC-HM-MNT
mnt-routes: LIR-MNT

aut-num: AS64512
as-name: GREEN-AS
tech-c: LA789-RIPE
admin-c: JD1-RIPE
mnt-by: AS-MNT
mnt-routes: LIR-MNT

route6: 2001:db8::/32
tech-c: LA789-RIPE
admin-c: JD1-RIPE
origin: AS64512
mnt-by: LIR-MNT

Add passwords



Session passwords

0 stored password(s)

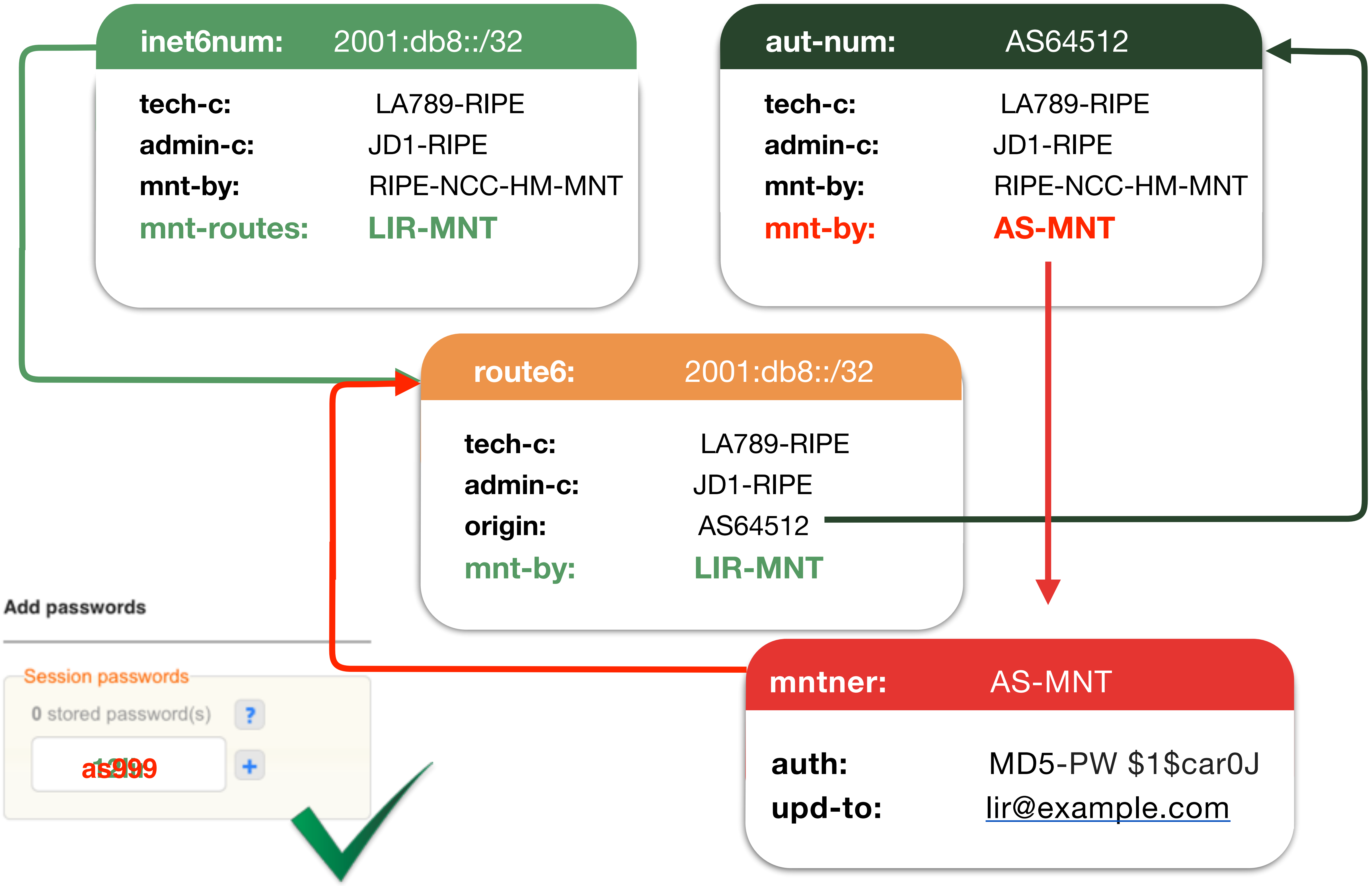
12lir

Registering Routes

- Creating route object
  - Sharing passwords
  - Adding other users’ maintainers to your objects
- New approach
  - For any missing authorisation, object is queued and notification is sent to the maintainer

```
mntner:     LIR-MNT

auth:       MD5-PW $1$car0J
upd-to:     lir@example.com
```

Registering Routes



```
inet6num:     2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-routes:   LIR-MNT
```

```
aut-num:      AS64512
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
mnt-by:       RIPE-NCC-HM-MNT
mnt-by:       AS-MNT
```

```
route6:       2001:db8::/32
tech-c:       LA789-RIPE
admin-c:      JD1-RIPE
origin:       AS64512
mnt-by:       LIR-MNT
```

```
mntner:       AS-MNT
auth:         MD5-PW $1$car0J
upd-to:       lir@example.com
```

Add passwords

Session passwords

0 stored password(s)

as999

What is a Routing Policy?

- What prefixes do you announce?
- Who are your neighbours?
  - Peers, transits and customers
- Which prefixes do you accept from them?
- What are your preferences?

aut-num Object and Routing Policy

```
aut-num:        AS64512
descr:          RIPE NCC Training Services
as-name:        GREEN-AS
tech-c:         LA789-RIPE
admin-c:        JD1-RIPE
import:         from AS64444 accept ANY
import:         from AS64488 accept ANY
export:         to AS64444 announce AS64512
export:         to AS64488 announce AS64512
mnt-by:         LIR-MNT
source:         RIPE
```

Why Publish Your Routing Policy?

- Some transit providers and IXPs (Internet Exchange Points) require it
  - They build their filters based on the routing registry
- Contributes to routing security and stability
  - Let people know about your intentions
- Can help in troubleshooting
  - Which parties are involved?

RIPE Database

- Close relation between registry information and routing policy
  - The holder of the resources knows how they should be routed

- The Routing Policy Specification Language (RPSL) originates from a RIPE Document
  - Shares attributes with the RIPE Database

Routing Registries Challenges

- Accuracy and completeness
- Not every Routing Registry is linked directly to an Internet Registry
  - Offline verification of the resource holder is needed
- Different authorisation methods
- Mirrors are not always up to date

Create a route or a route6 Object

Exercise 1

Exercise 1

- Create a **route** object for your IPv4 allocation
- Create a **route6** object for your IPv6 allocation
- List your AS Number (**aut-num**) as the origin for both objects

Routing Policy Specification Language

Section 4

Routing Policy

- A routing policy describes how a network works
  - Who do you connect with
  - Which prefixes or routes do you announce
  - Which routes do you accept from others
  - What are your preferences
- In your router, this is your BGP configuration
  - neighbours
  - route-maps
  - prefix lists
  - localpref

RPSL

- Language used by the IRRs
- Not vendor-specific
- Documented in RFC 2622
  - and RFC 2650 “Using RPSL in practice”

- Can be translated into router configuration

Objects Involved

- **route** or **route6** object
  - Connects a prefix to an origin AS
- **aut-num** object
  - Registration record of an AS Number
  - Contains the routing policy
- Sets
  - Objects can be grouped in sets, i.e. as-set, route-set
- Keywords
  - “ANY” matches every route

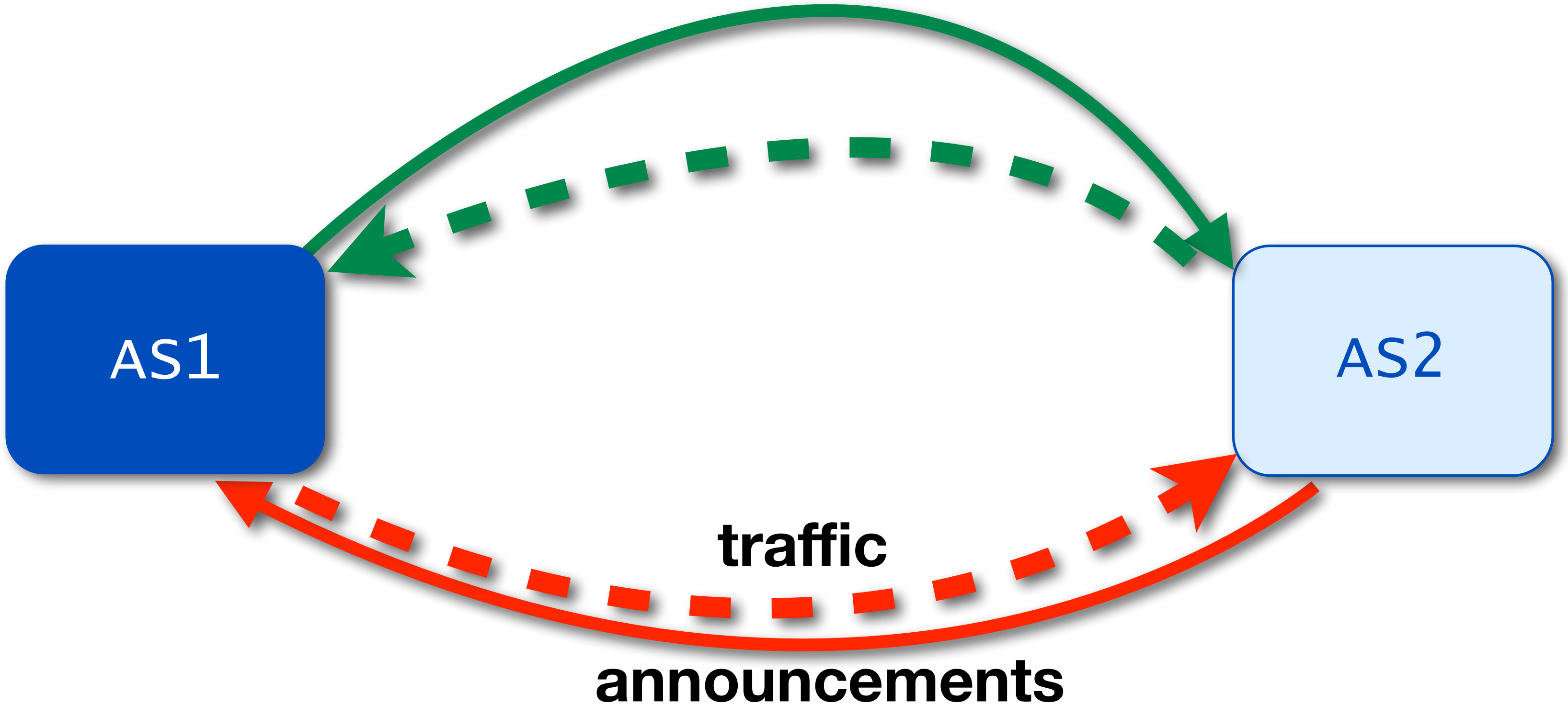
Notation

- AS Numbers are written as ASxxx
- Prefixes are written in CIDR notation
  - i.e.193.0.4.0/24
- Any value can be replaced by a list of values of the same type
  - AS1 can be replaced by “AS1 AS2 AS3”
- You can reference a set instead of a value
  - “...announce AS1” or “...announce as-myname”

Import and Export Attributes

- You can document your routing policy in your aut-num object in the RIPE Database:
  - Import lines describe what routes you accept from a neighbour and what you do with them
  - Export lines describe which routes you announce to your neighbour

Traffic Direction vs Announcement

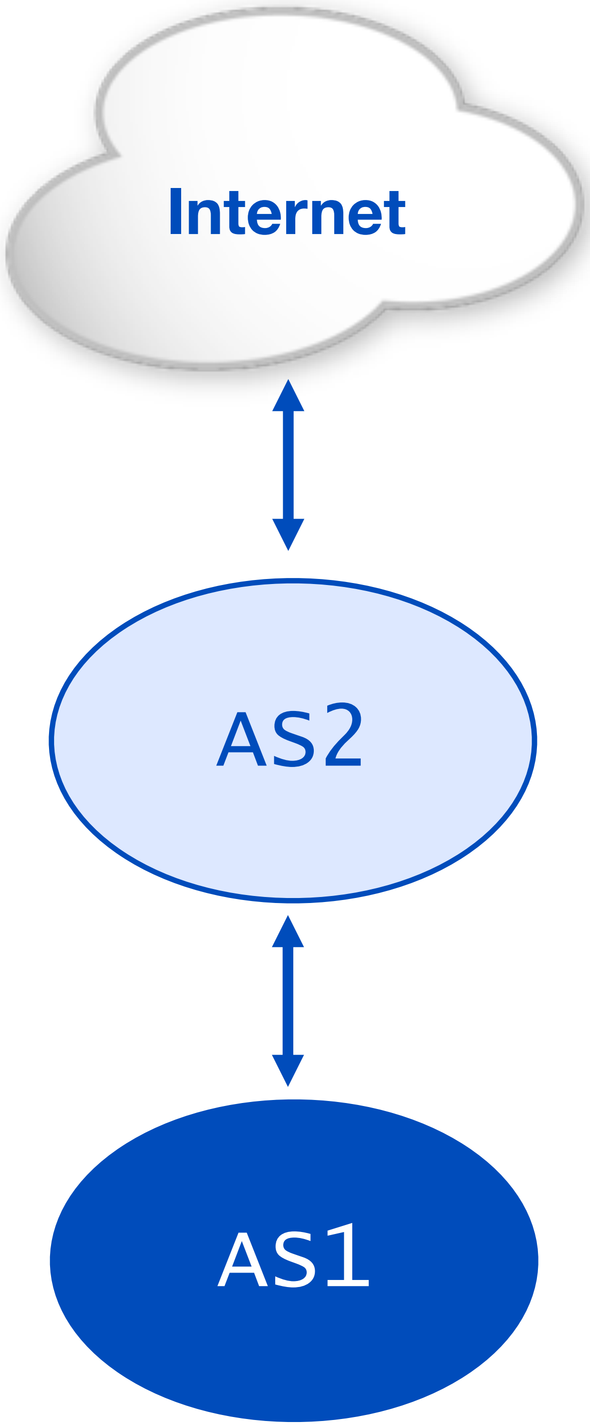


```
aut-num: AS1

  import: from AS2 accept AS2


  export: to AS2 announce  AS1
```

AS1 accepting those prefixes **from** AS2 that originate in AS2 so that the **outbound** traffic for AS2 can go **towards** the AS2

AS1 announcing prefixes (originating in AS1) **to** AS2, so that the **incoming** traffic for AS1 can flow **away** from the AS2

Example: You Are Downstream

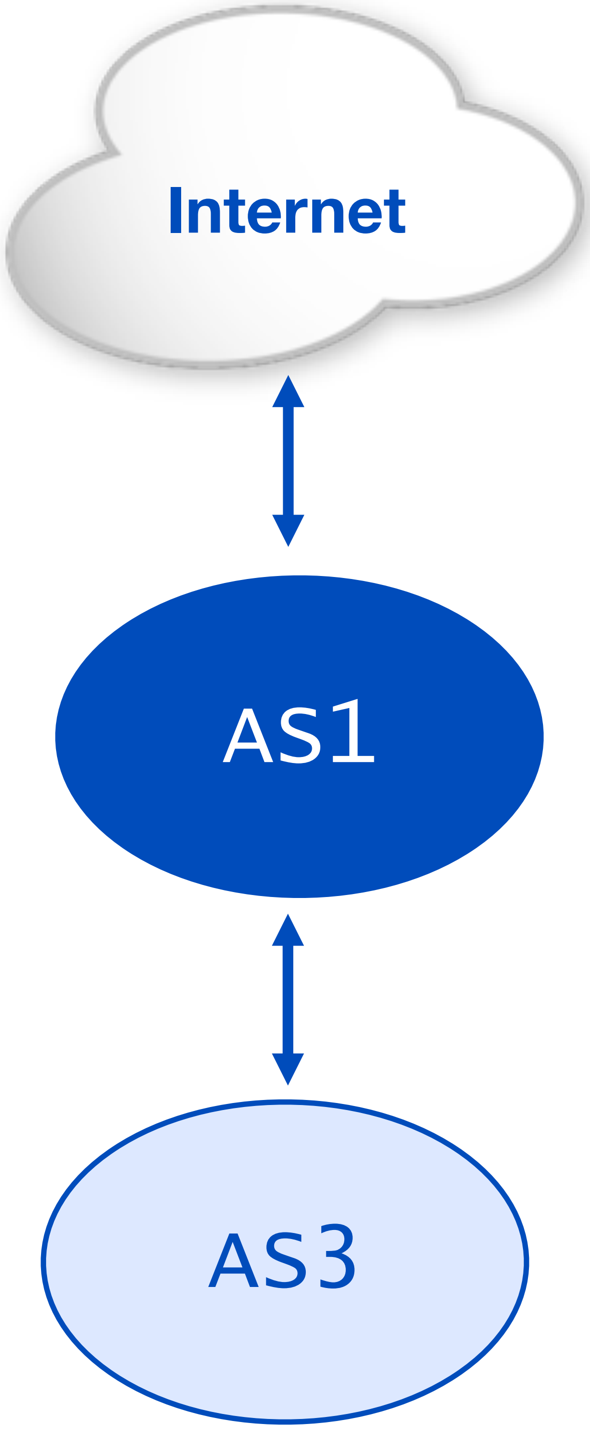


Transit provider

You

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

Example: You Are Upstream

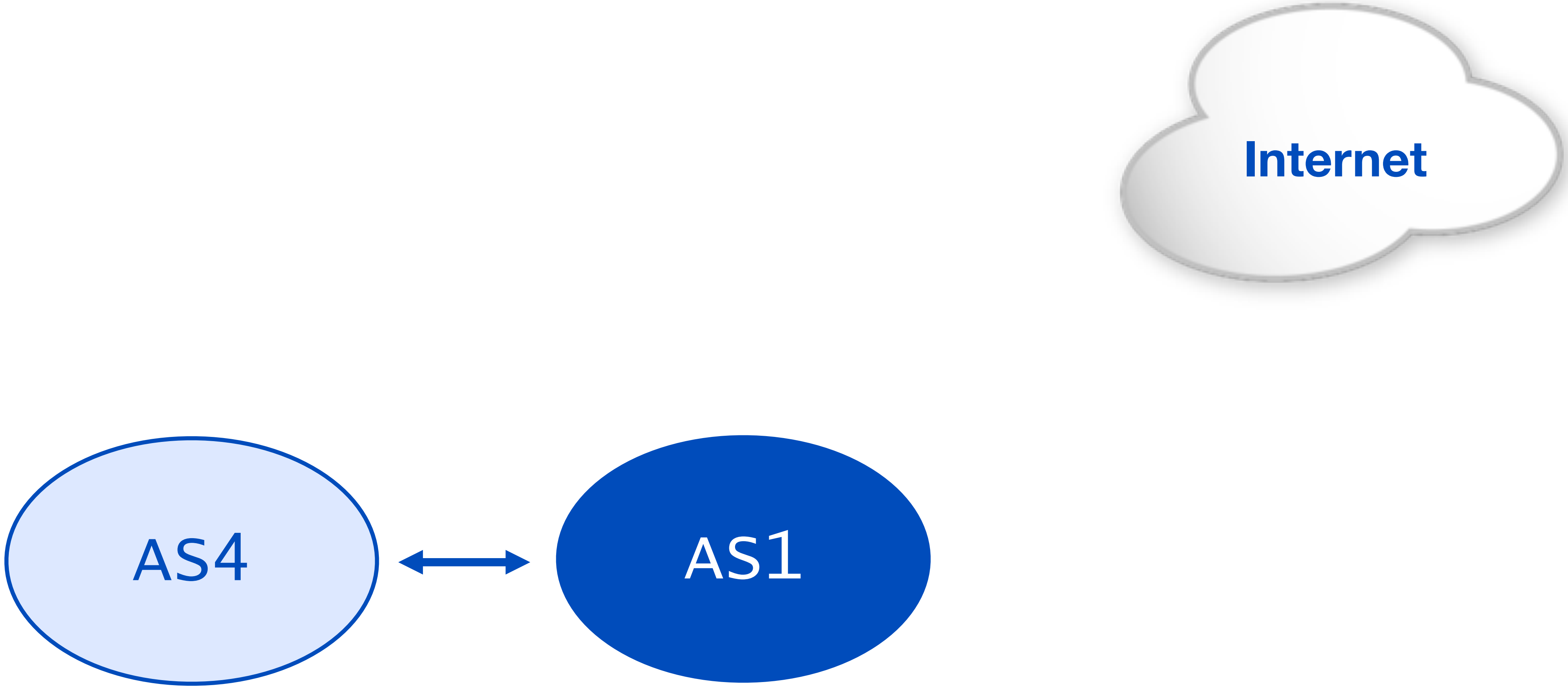


You

Downstream customer

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

Example: Peering

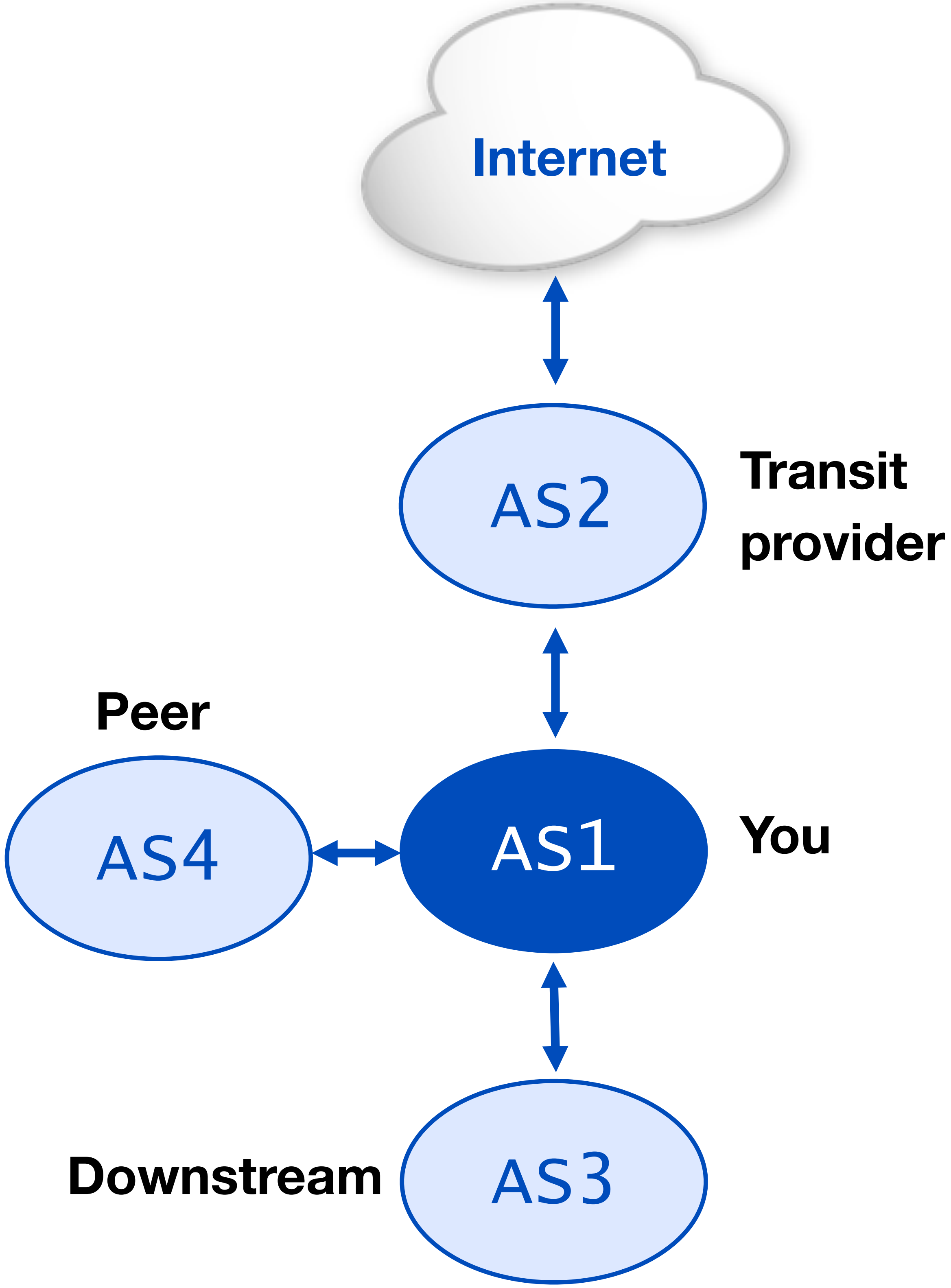


Peer

You

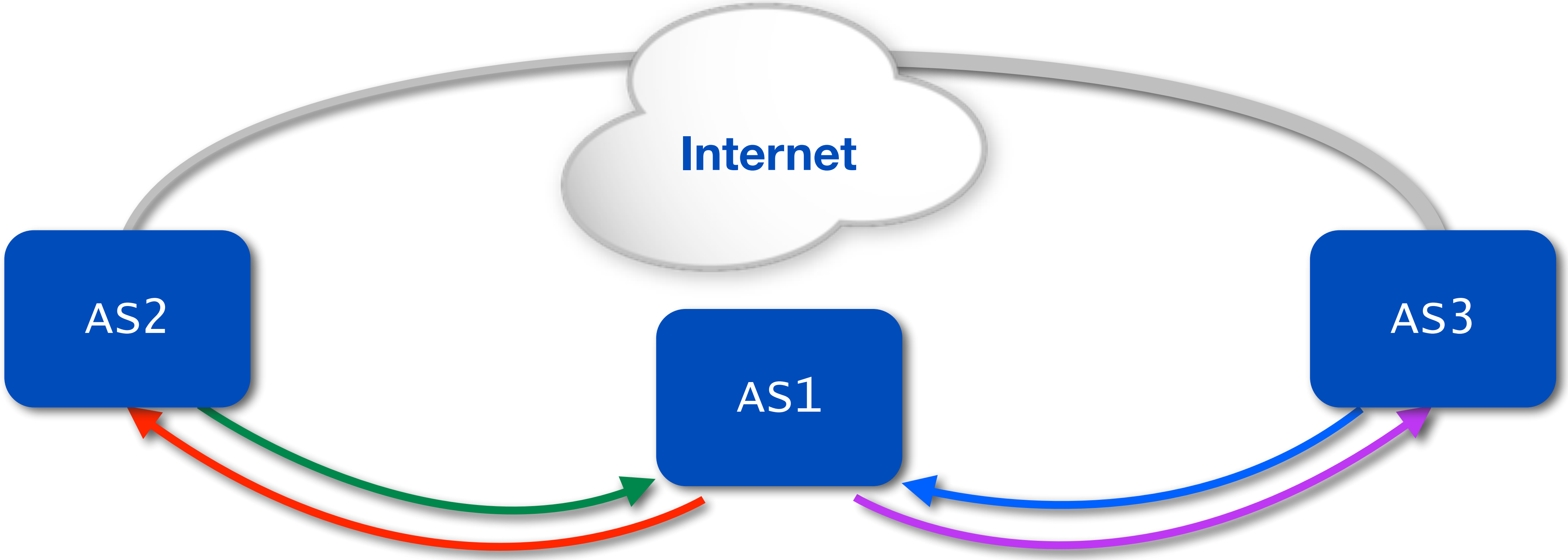
```
aut-num:   AS1
import: from AS4 accept AS4
export: to AS4 announce AS1
```

Example: Summary



```
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
```

RPSLng

- RPSL is older than IPv6, the defaults are IPv4
- IPv6 was added later using a different syntax
- You have to specify that it’s IPv6

```
mp-import:  afi ipv6.unicast from AS201 accept AS201
mp-export:  afi ipv6.unicast to AS201 announce ANY
```

- More information in RFC 4012 RPSLng

Retrieving Information from the IRR

Exercise 2

A Look at the Real World

- Have a look at AS 3333 in the RIPE Database
  - Which prefixes would you accept from AS 3333 if it was your customer?

- Remember to use the real database!

- Optionally verify the results using the tools at http://stat.ripe.net

RPSL in Practice

Section 5

Example Routing Policy

```
aut-num:     AS99
as-name:     SMALL-ISP-EU
descr:       My network
remarks:     ***   Transit via 101   ***
import:      from AS101 accept ANY
export:      to AS101 announce AS99 AS201 AS202
remarks:     ***   Transit via 102   ***
import:      from AS102 accept ANY
export:      to AS102 announce AS99 AS201 AS202
remarks:     ***   AS201 is a customer   ***
import:      from AS201 accept AS201
export:      to AS201 announce ANY
remarks:     ***   AS202 is a customer   ***
import:      from AS202 accept AS202
export:      to AS202 announce ANY
```

Using as-set

- Adding and removing customers can become time consuming

- Create a set to list them all at once

```
as-set:    AS-SMALLISP
descr:     Customers’ ASNs of a small ISP
members:   AS99
members:   AS201
members:   AS202
```

- And use that to describe your policy

```
export:    to AS101 announce AS-SMALLISP
export:    to AS102 announce AS-SMALLISP
```

Use Keywords for as-sets

```
as-set:     AS4:AS-CUSTOMERS
members:    AS7, AS5, AS8
```

```
aut-num: AS4
export: to AS3 announce AS4 AS4:AS-customers
export: to AS4:AS-CUSTOMERS announce ANY
import: from AS4:AS-CUSTOMERS accept PeerAS
```

- PeerAS means:
  - from AS5 accept AS5
  - from AS7 accept AS7
  - from AS8 accept AS8

Indicating Your Preferences

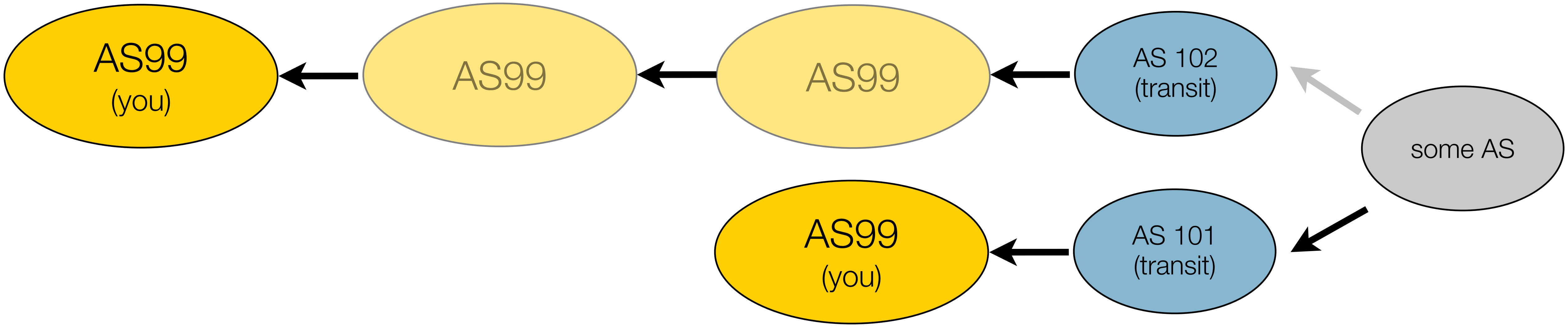
- BGP uses the “**localpref**” to influence which received routes you want to prefer
- In RPSL you can use the “**pref**” action on your import attributes
- Important: lower value means more preferred!

```
import:    from AS101 action pref=20;
           accept ANY
import:    from AS102 action pref=30;
           accept ANY
```

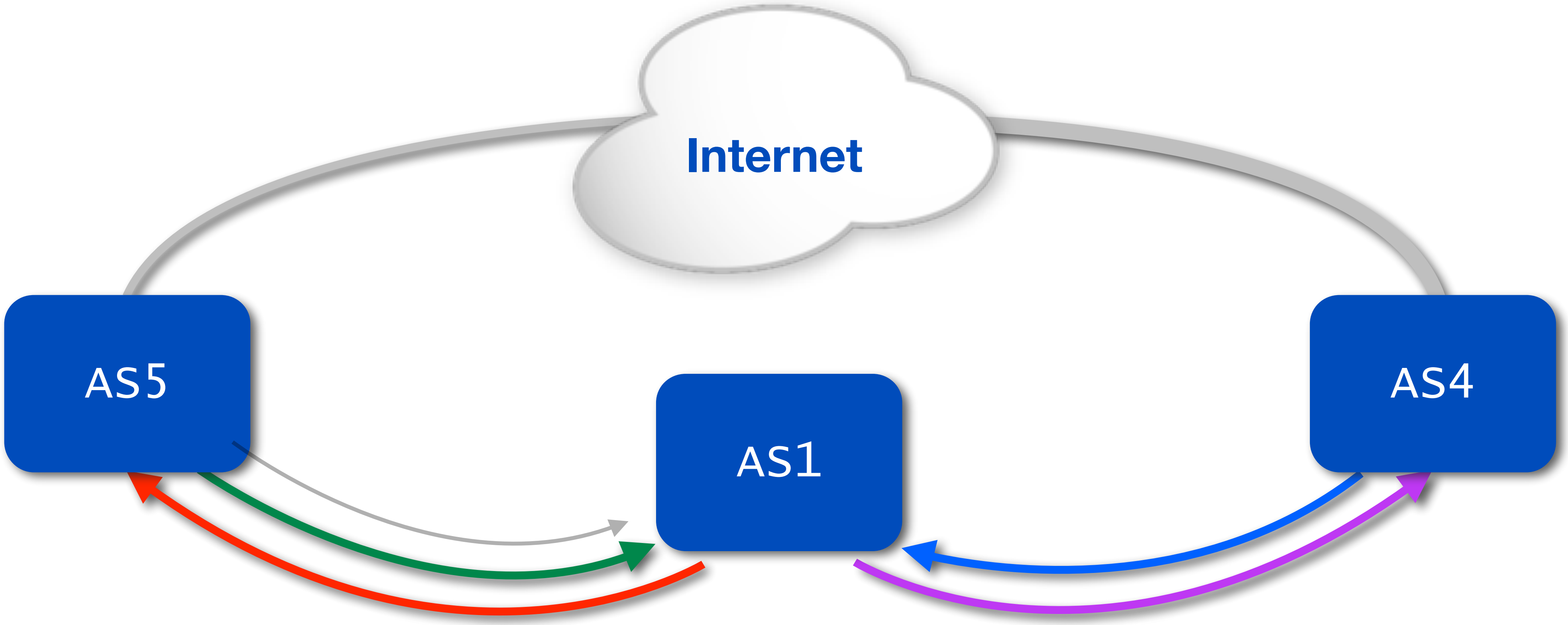
Describing AS Path Prepending

- AS Path prepending is used to influence other people's preferences
- Prepending can also be notated in RPSL using another action statement:

```
export:     to AS102 action aspath.prepend
             (AS99, AS99); announce AS-SMALLISP
```



Building an aut-num Object



```
aut-num: AS5

import: from AS1 accept AS1
export: to AS1 announce ANY
```

```
aut-num: AS1

import: from AS4  action pref=80;
        accept ANY
export: to AS4 announce AS1
import: from AS5  action pref=90;
        accept ANY
import: from AS5  action pref=70;
        accept AS5
export: to AS5
 action aspath.prepend (AS1, AS1);
 announce AS1
```

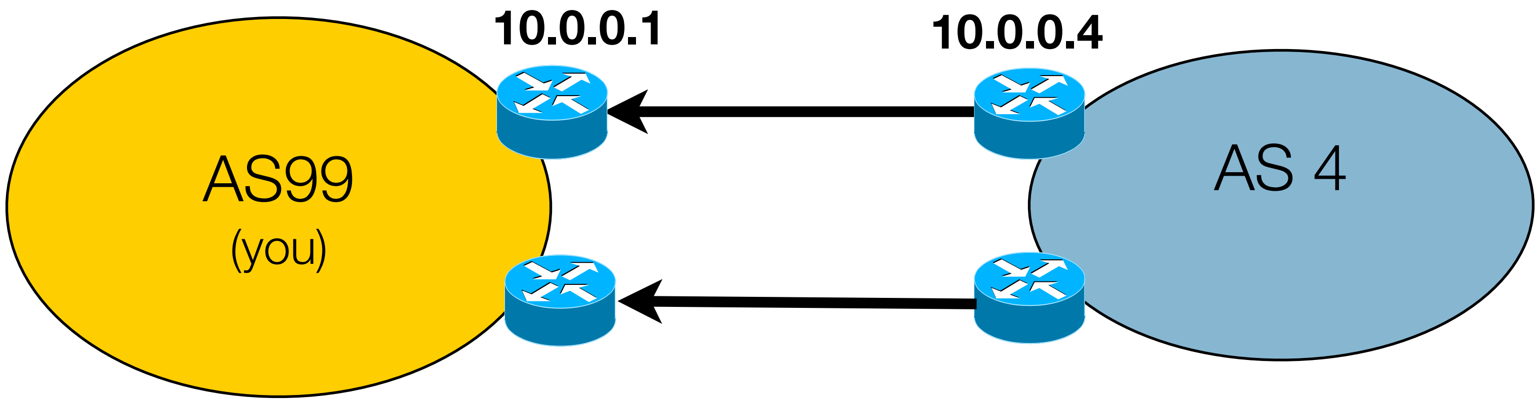
```
aut-num: AS4

import: from AS1 accept AS1
export: to AS1 announce ANY
```

MED (Multi Exit discriminator)

- Multiple Exit Discriminator
  - Differentiates connections to same peer
  - “Which inbound connection do I prefer?”
  - Doesn’t go beyond neighbour
- Local Pref has precedence over MED
  - To honour your neighbours MED:
  - Don’t set different prefs

Example: Using MED

```
export:    to AS4
           10.0.0.4 at 10.0.0.1
           action med=1000;
          announce AS99
export:    to AS4
           10.0.0.5 at 10.0.0.2
           action med=2000;
          announce AS99
```



Communities

- Optional tags
  - Can go through many peers
- Can be used for advanced filtering
- Not a routing parameter
- Enables customers to control their own routing policy
  - Publish your communities, and what you do with them
  - Filter incoming announcements accordingly

Example: Using Communities

- Set a community

```
import:      from AS6
             action community = { 99:100 };
             accept AS6
```

- Append a community

```
import:      from AS7
             action community.append(99:51);
             accept AS7
```

```
export:      to AS3
             action community .= { 99:100 };
             announce ANY
```

- Delete a community

```
import:      from AS201 action community.delete
             (99:100); accept AS201
```

Example: Communities Filtering

```
import:     from AS21
             accept AS6 AND
             community.contains = (21:32)
```

```
import:     from AS17
             accept community(68:2)
```

```
import:     from AS1:AS-CUSTOMERS
             accept PeerAS AND
             community.contains (202:3)
```

```
export:     to AS3
             announce AS1:AS-CUST AND
             community == {1:113}
```

```
export:     to AS1:AS-PEERS
             announce ANY AND
             community.contains (1:75)
```

AS Path Regular Expressions

- You can use regular expressions in your filters
  - they are always enclosed in “< >”
  - import: from AS201 accept <^AS201+$>
- Uses the standard posix notation
  - “^” start of path
  - “$” end of path
  - “*” zero or more
  - “+” one or more
  - “?” zero or one

Literal Prefixes

- Instead of AS Numbers you can use prefixes
  - import: from AS2121 accept {193.0.24.0/21}

- Operators can be used to define ranges
  - “^-” all more specifics excluding the prefix itself
  - “^+” all more specifics including the prefix itself
  - “^n” all routes of length n in this prefix
  - “^n-m” all routes of length n to length m

Using a route-set

- Groups literal prefixes
- Can include other route-sets and even ASNs

```
route-set: RS-BAR
descr:     All ASNs of a small ISP
members:   5.0.0.0/8^+, 30.0.0.0/8^24-32
members:   rs-foo^+
members:   AS2
```

- And use that to describe/simplify your policy

```
export:    to AS101 announce RS-BAR
```

Default Routes

- Next to import and export there can also be a default line to describe your default policy

```
export:    to AS99 announce AS201
import:    from AS202 accept AS202
export:    to AS202 announce AS201
default:   to AS99 action pref=150
```

- Instead of all routes, you can also announce a default route

```
export:    to AS101 announce RS-BAR
```

The Simplified Object

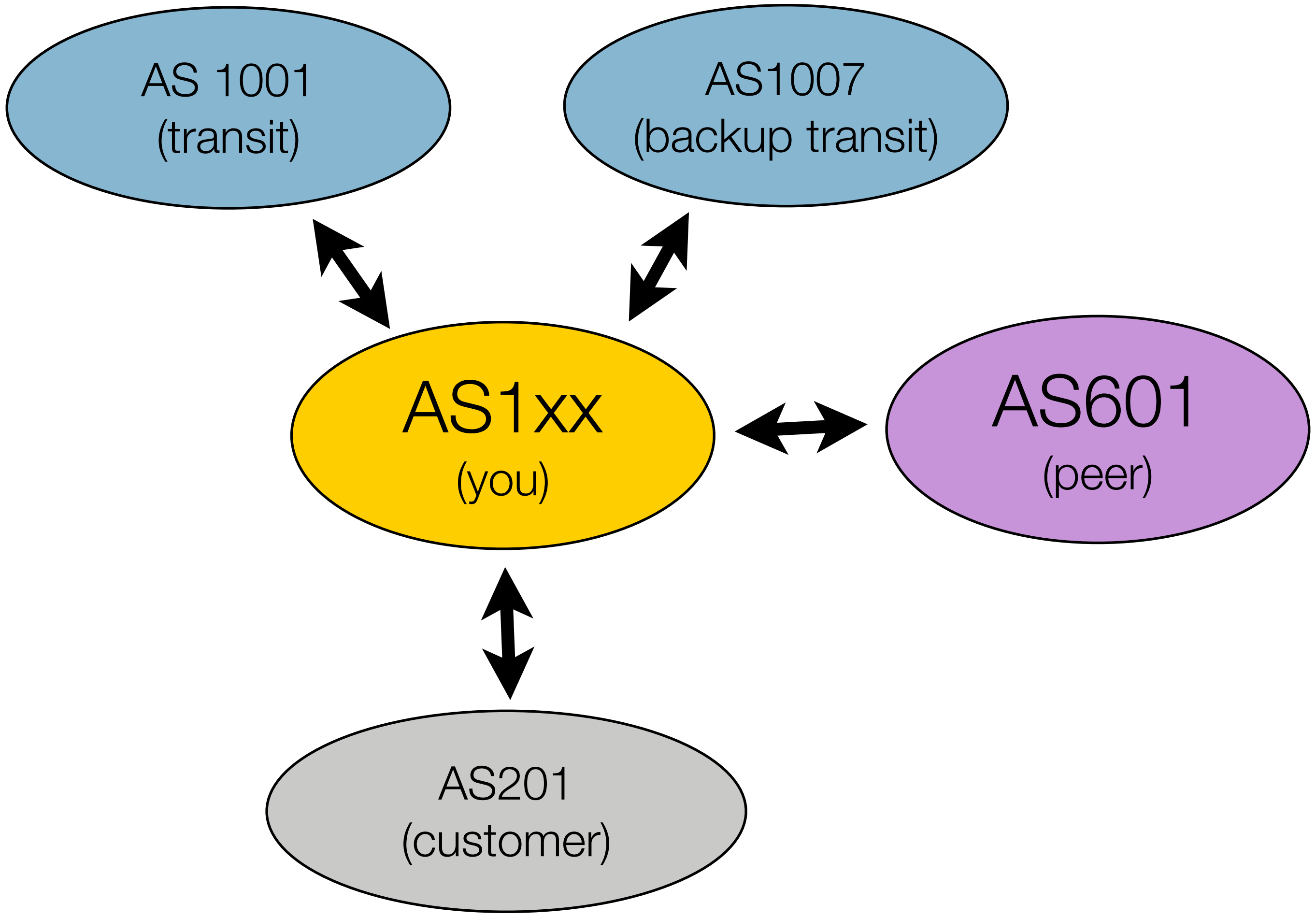
```
aut-num:    AS99
as-name:    SMALL-ISP-EU
descr:      My network
remarks:    *** Announcements are grouped ***
import:     from AS101 accept ANY
export:     to AS101 announce AS-SMALLISP
import:     from AS102 accept ANY
export:     to AS102 announce AS-SMALLISP
remarks:    *** My Customers are grouped ***
import:     from AS99:Customers accept PEERAS
export:     to AS99:Customers announce ANY
```

Describing Your Routing Policy

Exercise 3

Modifying aut-num Object

- Take the scenario as presented



  - In the TEST RIPE Database update your AS (**aut-num**), adding **import**, **export**, **mp-import**, **mp-export** attributes to describe your policy towards these neighbours

Tools and Automation

Section 6

Making Life Easier

- There are a lot of tools around that use information in the Routing Registry
- Some can generate complete router configurations like the IRRToolset
- Most are open source tools
  - You can modify them to your needs
  - Some are not very well maintained

Example Tools

- IRRToolkit (written in C++)
  - http://irrtoolset.isc.org/
- Rpsltool (perl)
  - http://www.linux.it/~md/software
- IRR Power Tools (PHP)
  - http://sourceforge.net/projects/irrpt/
- BGPQ3 (C)
  - http://snar.spb.ru/prog/bgpq3/
- Filtergen (Level 3)
  - whois -h filtergen.level3.net RIPE::ASxxx
- IRR Explorer (web)
  - http://irrexplorer.nlnog.net

Building Your Own

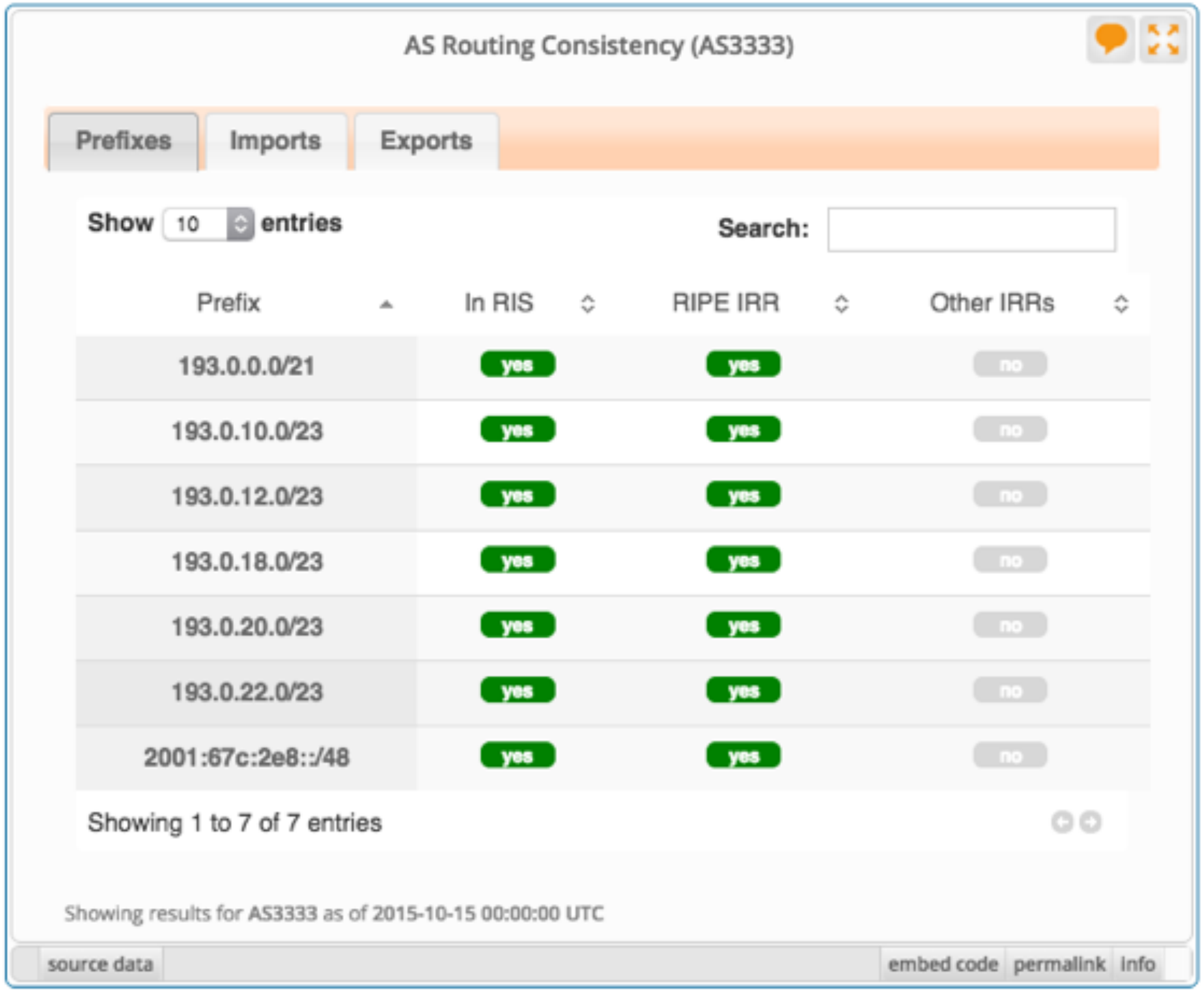
- A couple of things to keep in mind
  - The RIPE Database has limits on the number of queries you can do per day
  - Query flags or output format can change over time
- Instead of the whois interface, you can use the RESTful API for the RIPE Database
  - Uses XML or JSON for output
  - See **https://ripe.net/developer**
  - Also visit **https://labs.ripe.net** for more information

Getting the Complete Picture

- Automation relies on the IRR being complete
  - Not all resources are registered in an IRR
  - Not all information is correct
- Small mistakes can have a big impact
- Check your output before using it
  - Be prepared to make manual overrides
- Help others by documenting your policy

RIPEstat

- You can compare the Routing Registry and the Internet routing table using http://stat.ripe.net



AS Routing Consistency (AS3333)

Prefix	In RIS	RIPE IRR	Other IRRs
193.0.0.0/21	yes	yes	no
193.0.10.0/23	yes	yes	no
193.0.12.0/23	yes	yes	no
193.0.18.0/23	yes	yes	no
193.0.20.0/23	yes	yes	no
193.0.22.0/23	yes	yes	no
2001:67c:2e8::/48	yes	yes	no

Showing 1 to 7 of 7 entries

Showing results for AS3333 as of 2015-10-15 00:00:00 UTC

Using a Tool

Exercise 4

Using Filtergen

- Use a tool to retrieve the same information from the exercise 2
- “whois -h filtergen.level3.net RIPE::AS3333”
  - Syntax is “RIPE::” followed by the AS you want information about
- Do you get the same answers?
  - What is the result of AS-RIPENCC?
  - If you have time, try AS-TELIANET

Questions

Introduction the the RPKI

Section 7

Why RPKI ?

To be able to answer the question:

Is that ASN authorised to originate that address range?

RPKI and IRR

- Why yet another system?
  - Lots of Routing Registries
  - Not all mirroring each other
  - Different levels of trustworthiness and authentication

- RPKI replaces IRR or lives side by side?
  - Side by side: different advantages
  - Security, almost real time, simple interface: RPKI
  - More info in: IRR

The Advantages of RPKI

- Useable toolset
  - No installation required
  - Easy to configure manual overrides
- Tight integration with routers
  - Supported routers have awareness of RPKI validity states
- Stepping stone for AS-Path Validation
  - Prevent Attacks on BGP

RPKI
The announcers side

Section 8

Resource Certificates

- RIPE NCC issues digital certificates
    - To LIRs
    - To PI end users

- Upon request

- Certificate lists all resources held by the member

Which Resources Are Certified?

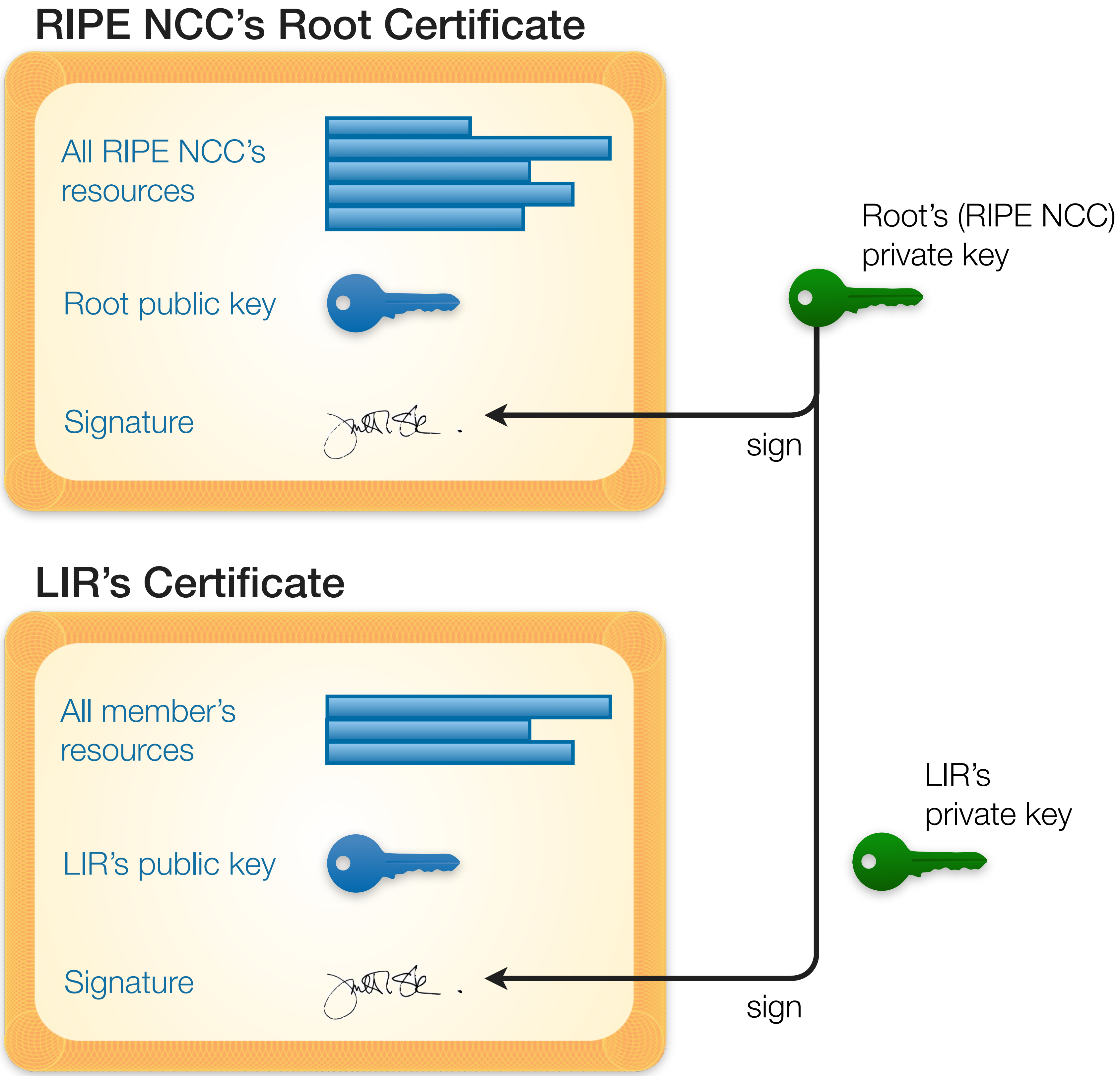
- Everything for which we are 100% sure who the holder is
  - Provider Aggregatable (PA) addresses
  - Provider Independent (PI) addresses
    - marked as LIR "Infrastructure"
    - for which we have a contract (Policy 2007-01)
  - Legacy Resources

RPKI Chain of Trust

- RIPE NCC holds self-signed root certificate for all resources they have in the registry
  - Signed by the root’s private key

- The root certificate is used to sign all certificates for members listing their resources
  - Signed by the root’s private key

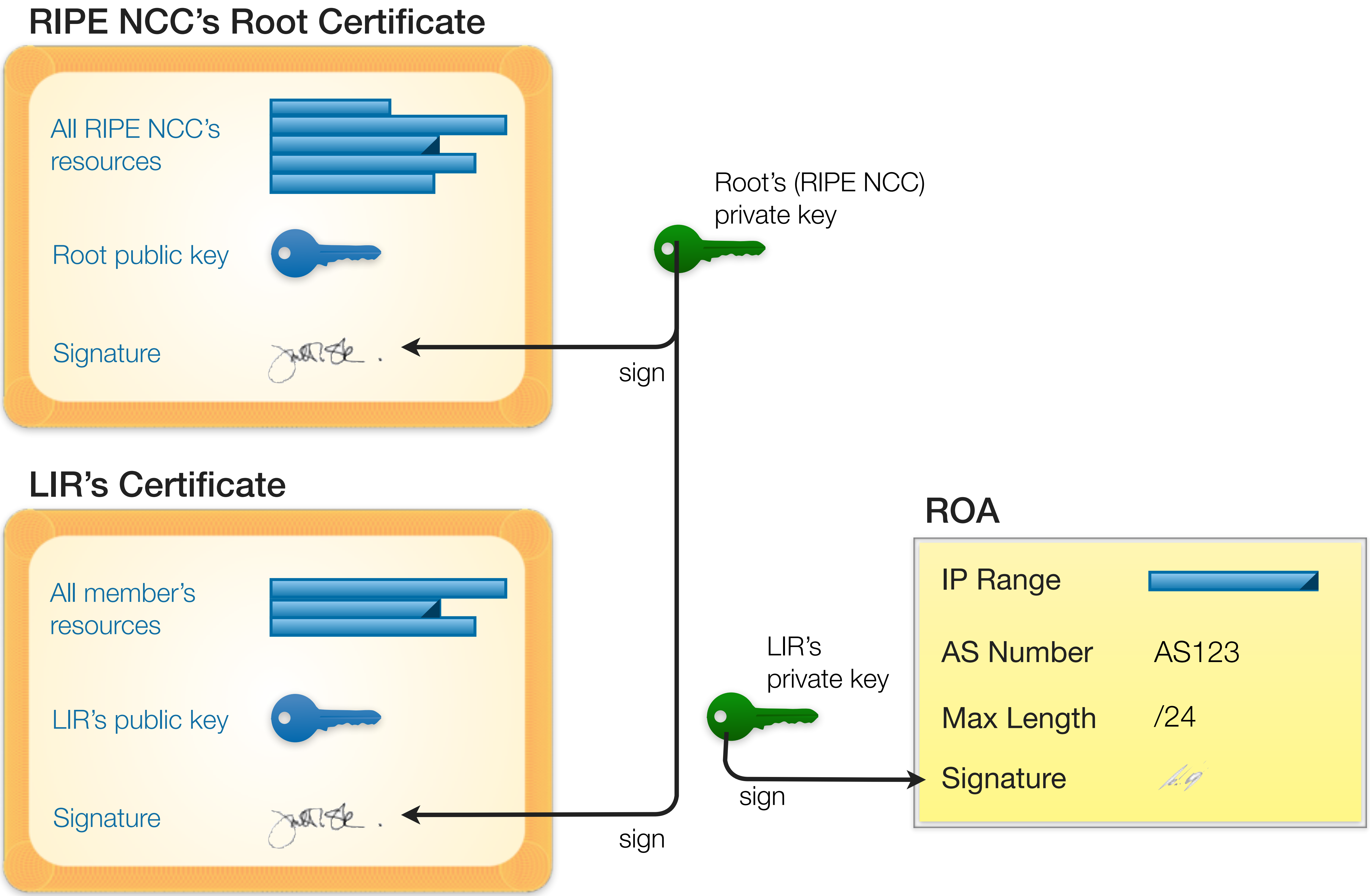
RPKI Chain of Trust



ROA (Route Origin Authorisation)

- LIRs can use their certificate to create a ROA for each of their resources (IP address ranges)
  - Signed by the LIR’s private key
- ROA states
  - Address range
  - Which AS this is announced from (freely chosen)
  - Maximum length (freely chosen)
- You can have multiple ROAs for an IP range
- ROAs can overlap

ROA Chain of Trust



RIPE NCC's Root Certificate

All RIPE NCC's resources

Root public key

Signature

Root's (RIPE NCC) private key

sign

LIR's Certificate

All member's resources

LIR's public key

Signature

sign

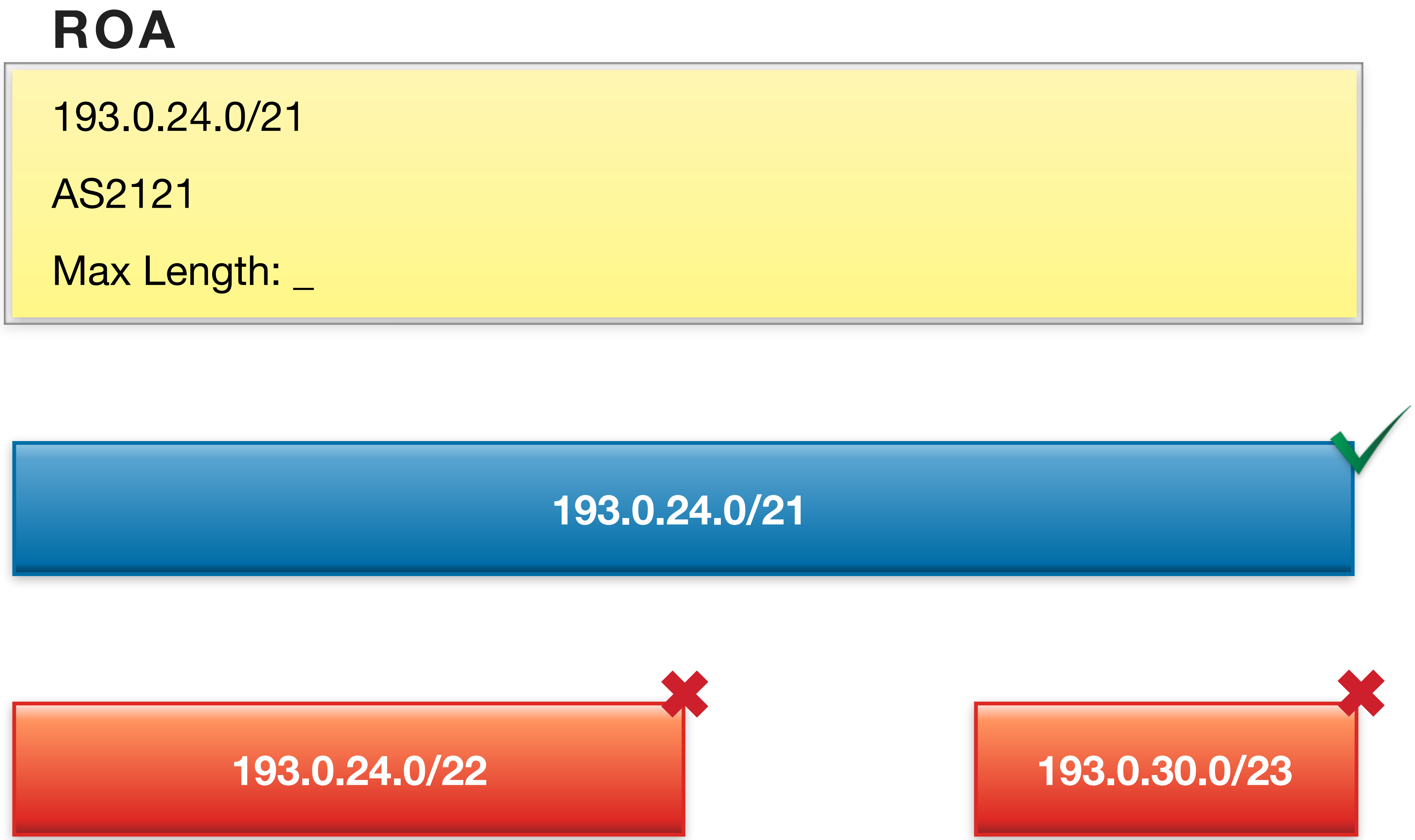
LIR's private key

sign

ROA

IP Range	
AS Number	AS123
Max Length	/24
Signature	

Example: ROA



ROA

193.0.24.0/21

AS2121

Max Length: _

193.0.24.0/21 ✓

193.0.24.0/22 ✗

193.0.30.0/23 ✗

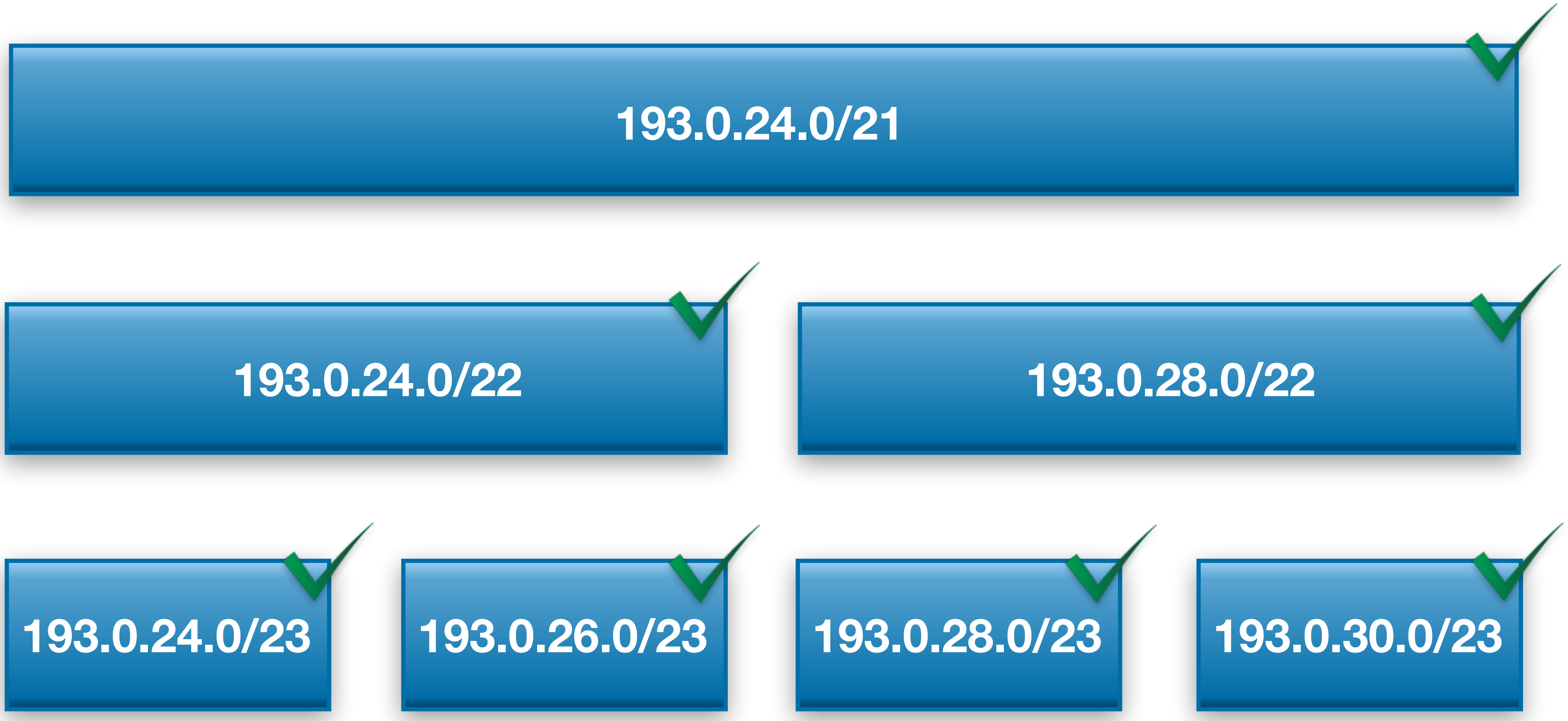
Example: ROA

ROA

193.0.24.0/21

AS2121

Max Length: /23

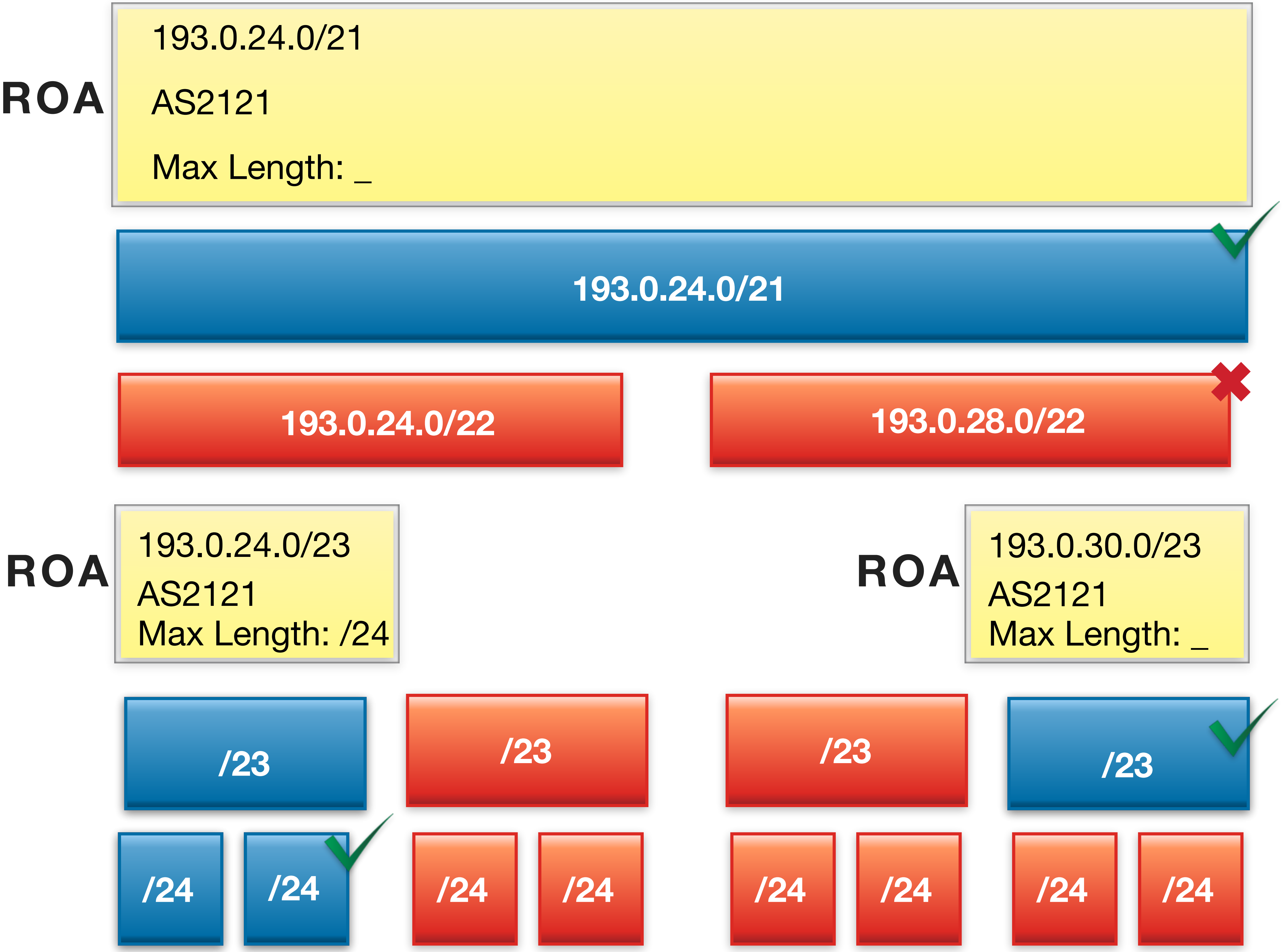


193.0.24.0/21 ✓

193.0.24.0/22 ✓ 193.0.28.0/22 ✓

193.0.24.0/23 ✓ 193.0.26.0/23 ✓ 193.0.28.0/23 ✓ 193.0.30.0/23 ✓

Example: ROA



ROA

193.0.24.0/21

AS2121

Max Length: _

193.0.24.0/21 ✓

193.0.24.0/22

193.0.28.0/22 ✗

ROA

193.0.24.0/23

AS2121

Max Length: /24

ROA

193.0.30.0/23

AS2121

Max Length: _

/23

/23

/23

/23 ✓

/24

/24 ✓

/24

/24

/24

/24

/24

/24

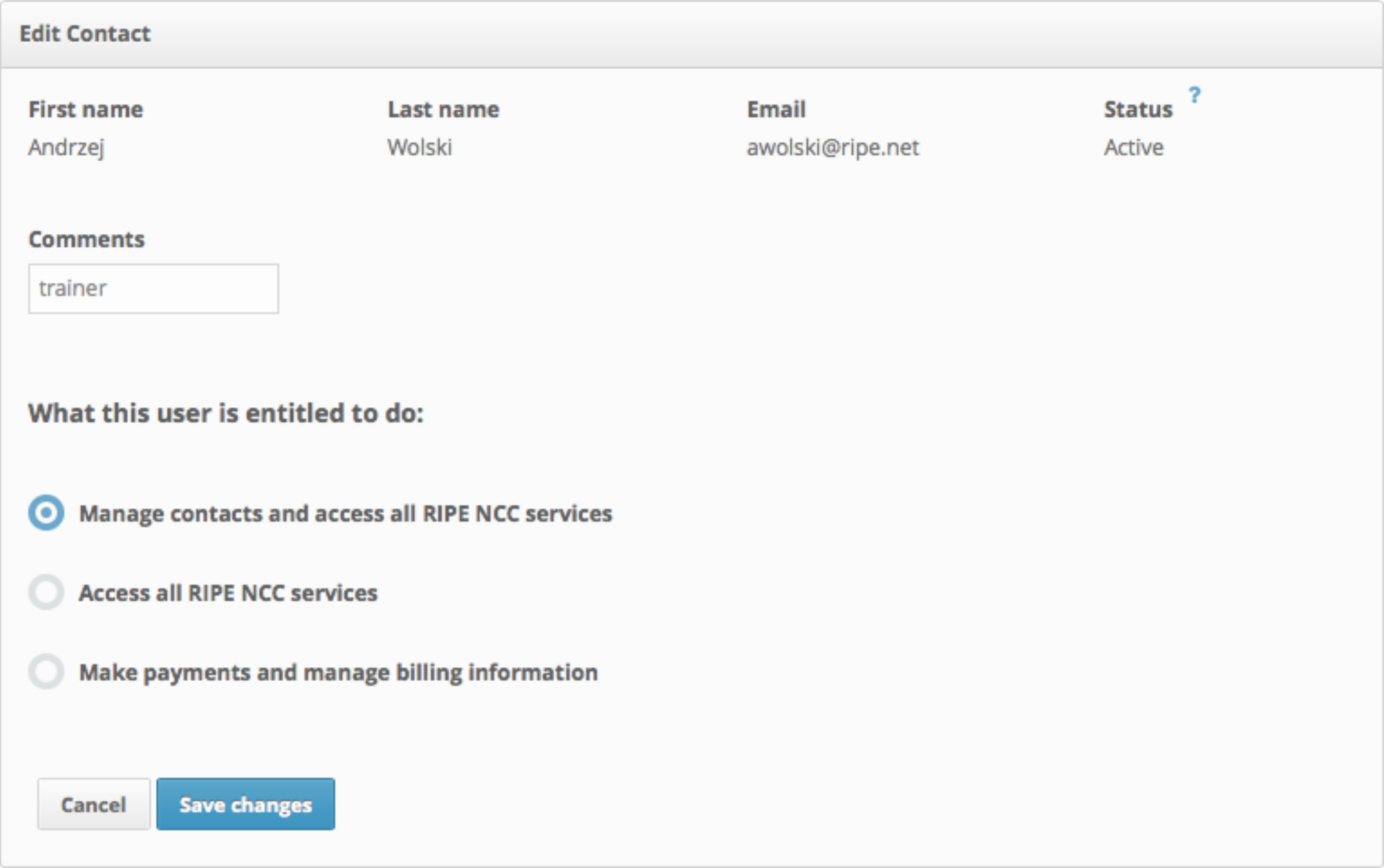
Public Repository

- RIPE NCC maintains a Certificate Repository containing
  - All the certificates
  - All the public keys
  - All the ROAs

RPKI Certification

Section 9

Enabling Access in the LIRPortal



Edit Contact

First name	Last name	Email	Status ?
Andrzej	Wolski	awolski@ripe.net	Active

Comments

trainer

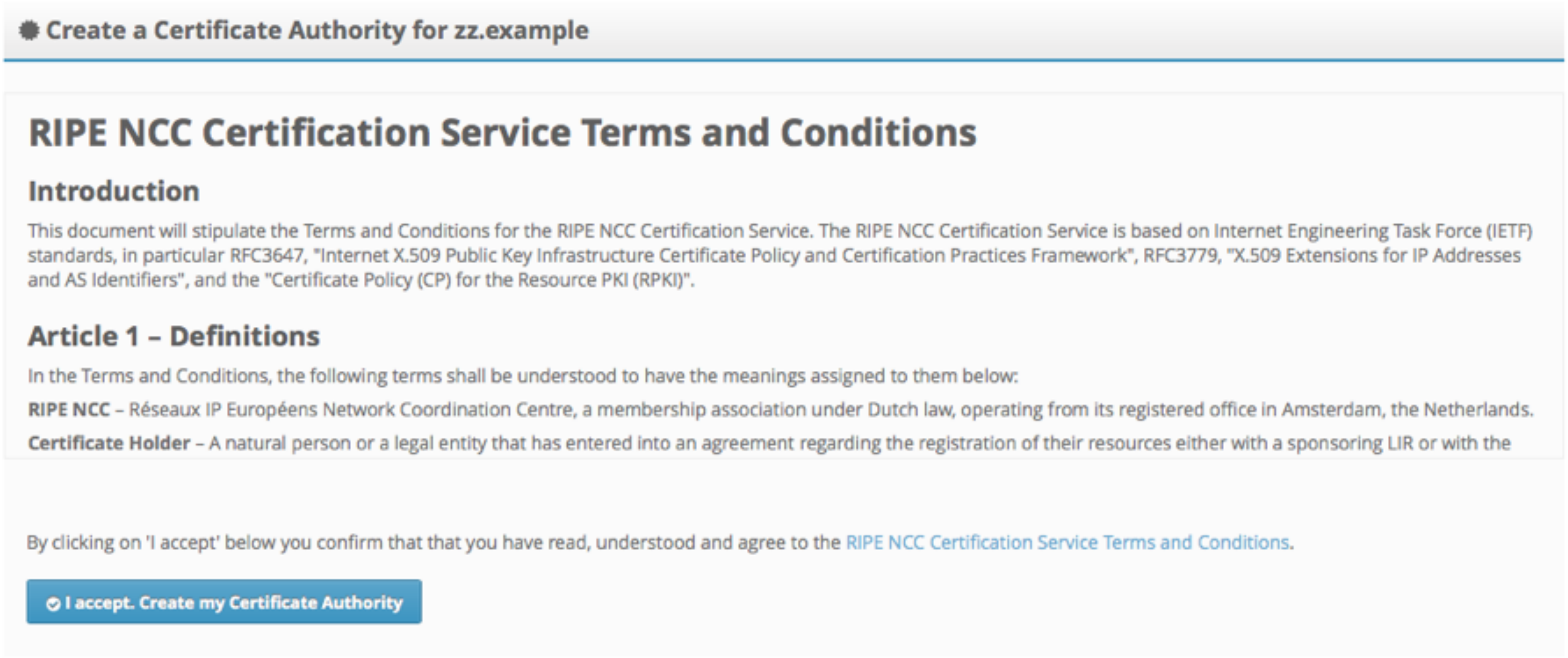
What this user is entitled to do:

● Manage contacts and access all RIPE NCC services

○ Access all RIPE NCC services

○ Make payments and manage billing information

Setting up Certificate Authority



Create a Certificate Authority for zz.example

RIPE NCC Certification Service Terms and Conditions

Introduction

This document will stipulate the Terms and Conditions for the RIPE NCC Certification Service. The RIPE NCC Certification Service is based on Internet Engineering Task Force (IETF) standards, in particular RFC3647, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC3779, "X.509 Extensions for IP Addresses and AS Identifiers", and the "Certificate Policy (CP) for the Resource PKI (RPKI)".

Article 1 – Definitions

In the Terms and Conditions, the following terms shall be understood to have the meanings assigned to them below:

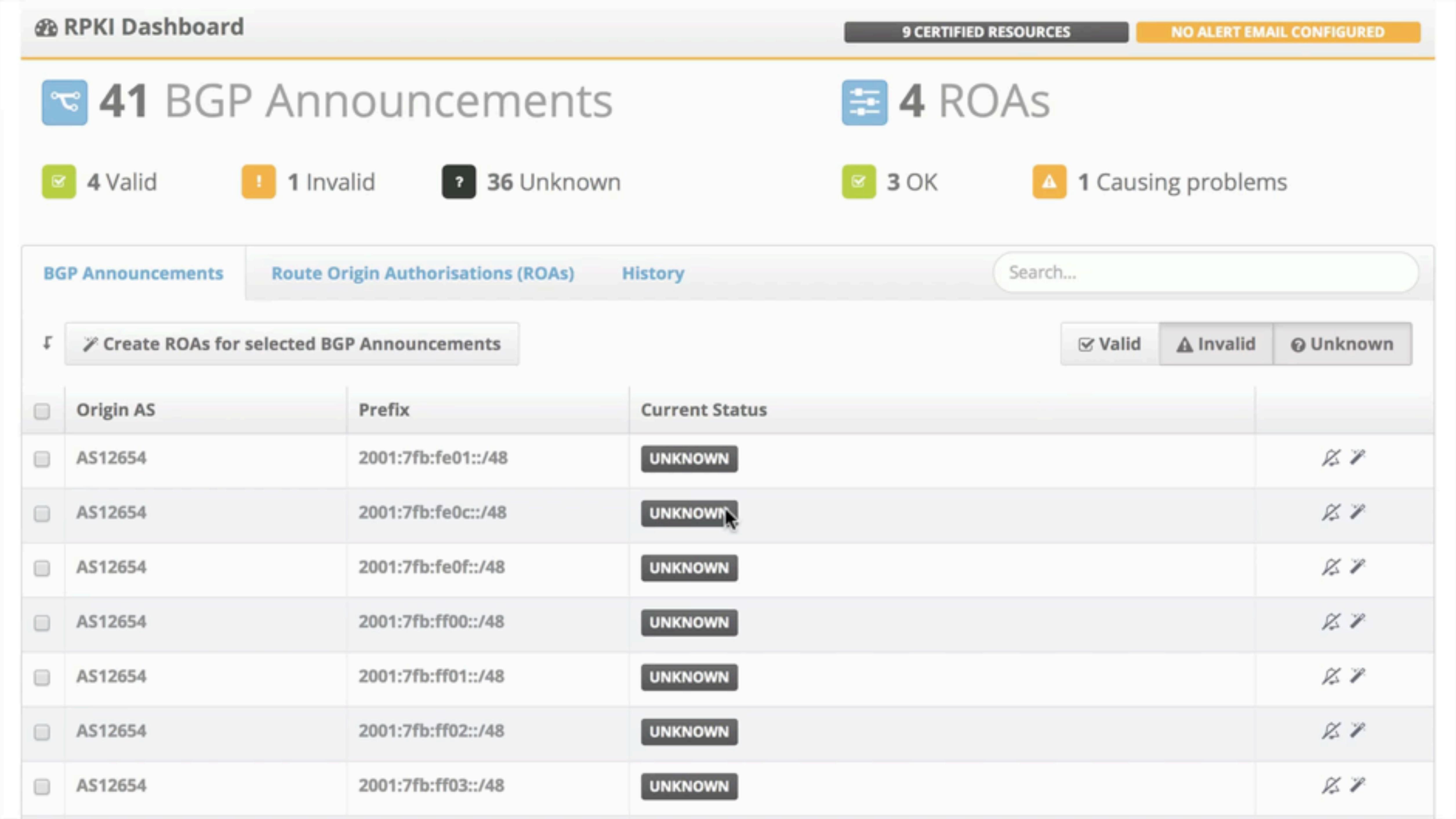
RIPE NCC – Réseaux IP Européens Network Coordination Centre, a membership association under Dutch law, operating from its registered office in Amsterdam, the Netherlands.

Certificate Holder – A natural person or a legal entity that has entered into an agreement regarding the registration of their resources either with a sponsoring LIR or with the

By clicking on 'I accept' below you confirm that that you have read, understood and agree to the RIPE NCC Certification Service Terms and Conditions.

https://localcert.ripe.net

Managing ROAs



RPKI Dashboard

9 CERTIFIED RESOURCES | NO ALERT EMAIL CONFIGURED

41 BGP Announcements

4 Valid | 1 Invalid | 36 Unknown

4 ROAs

3 OK | 1 Causing problems

BGP Announcements
Route Origin Authorisations (ROAs)
History
Search...
Create ROAs for selected BGP Announcements
Valid
Invalid
Unknown

□	Origin AS	Prefix	Current Status
□	AS12654	2001:7fb:fe01::/48	UNKNOWN
□	AS12654	2001:7fb:fe0c::/48	UNKNOWN
□	AS12654	2001:7fb:fe0f::/48	UNKNOWN
□	AS12654	2001:7fb:ff00::/48	UNKNOWN
□	AS12654	2001:7fb:ff01::/48	UNKNOWN
□	AS12654	2001:7fb:ff02::/48	UNKNOWN
□	AS12654	2001:7fb:ff03::/48	UNKNOWN

RPKI
Relying Party’s side

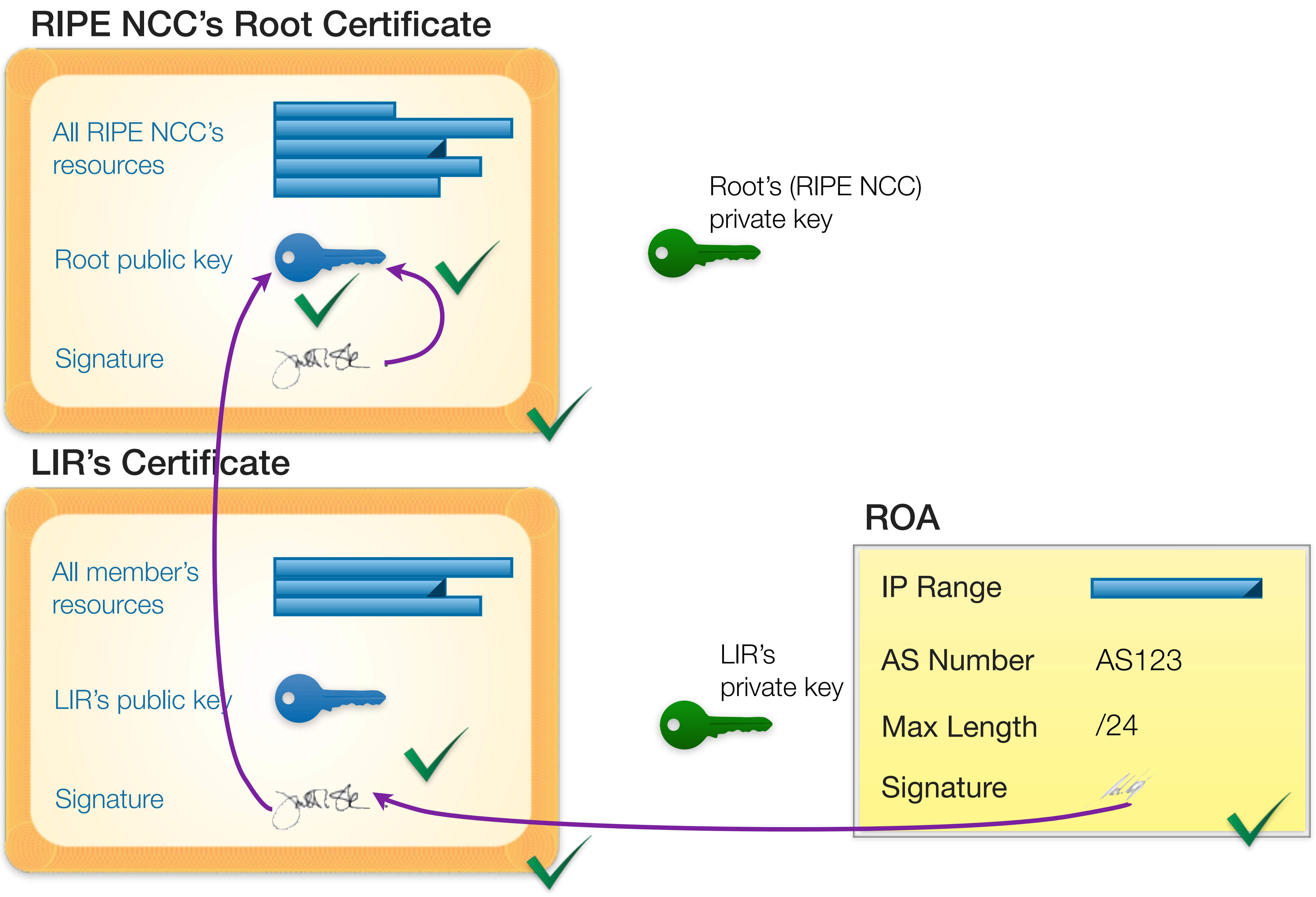
Section 10

Validator

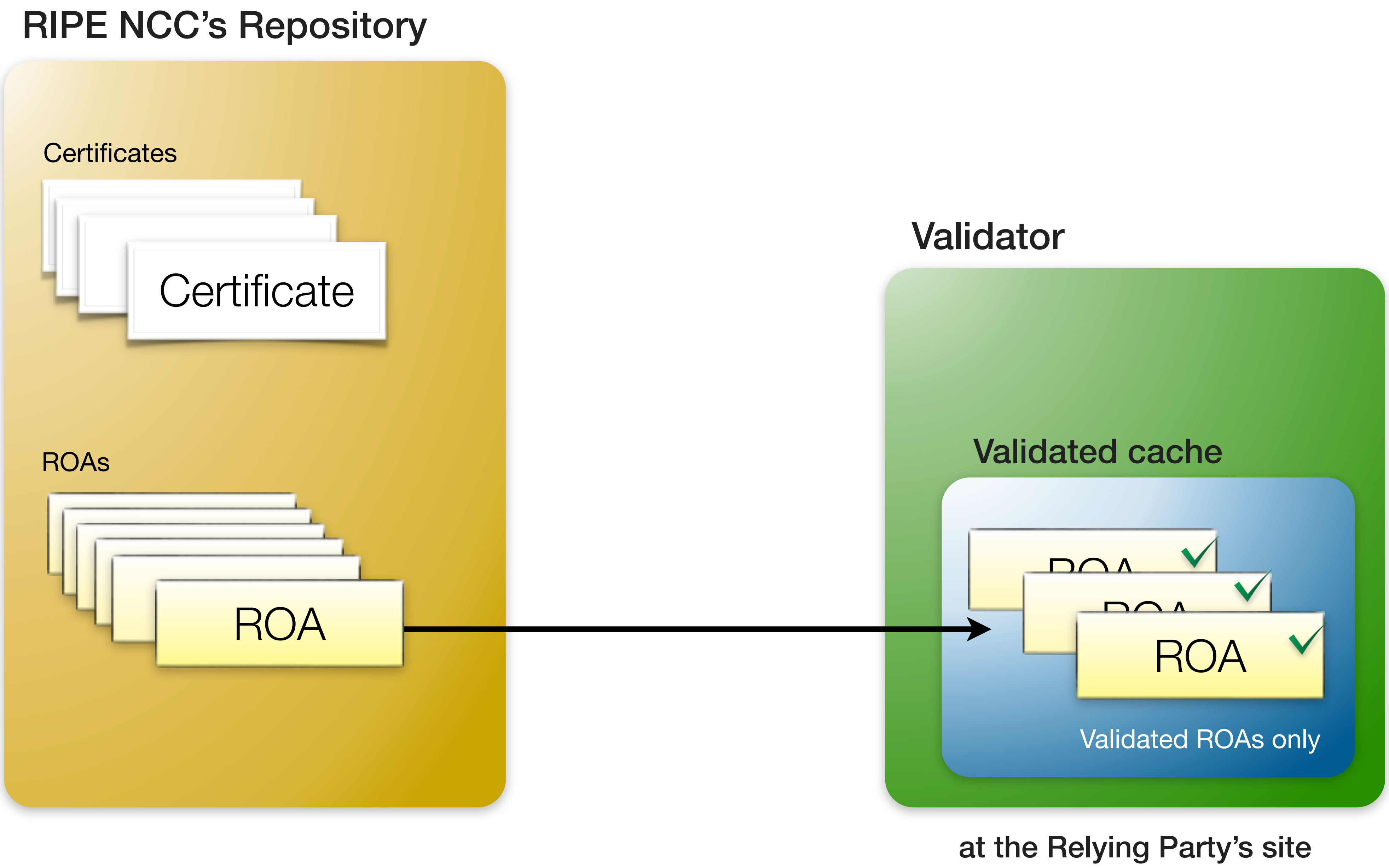
- The validator of the client can access RIPE NCC's Repository with all the certificates, public keys, ROAs

- It downloads everything and then performs validation, checking whether the certificates and ROAs are valid. Then it constructs a list of valid ROAs, which is its "validated cache"

ROA Chain of Trust



Validated Cache



Invalid ROAs

- Invalid ROAs are simply not included in the list of validated ROAs when the validator of the client computes them

- Reasons for a ROA to be invalid
  - The signing certificate or key pair has expired or has been revoked
  - It does not validate back to a configured trust anchor
  - The LIR’s resource has been returned to the RIPE NCC

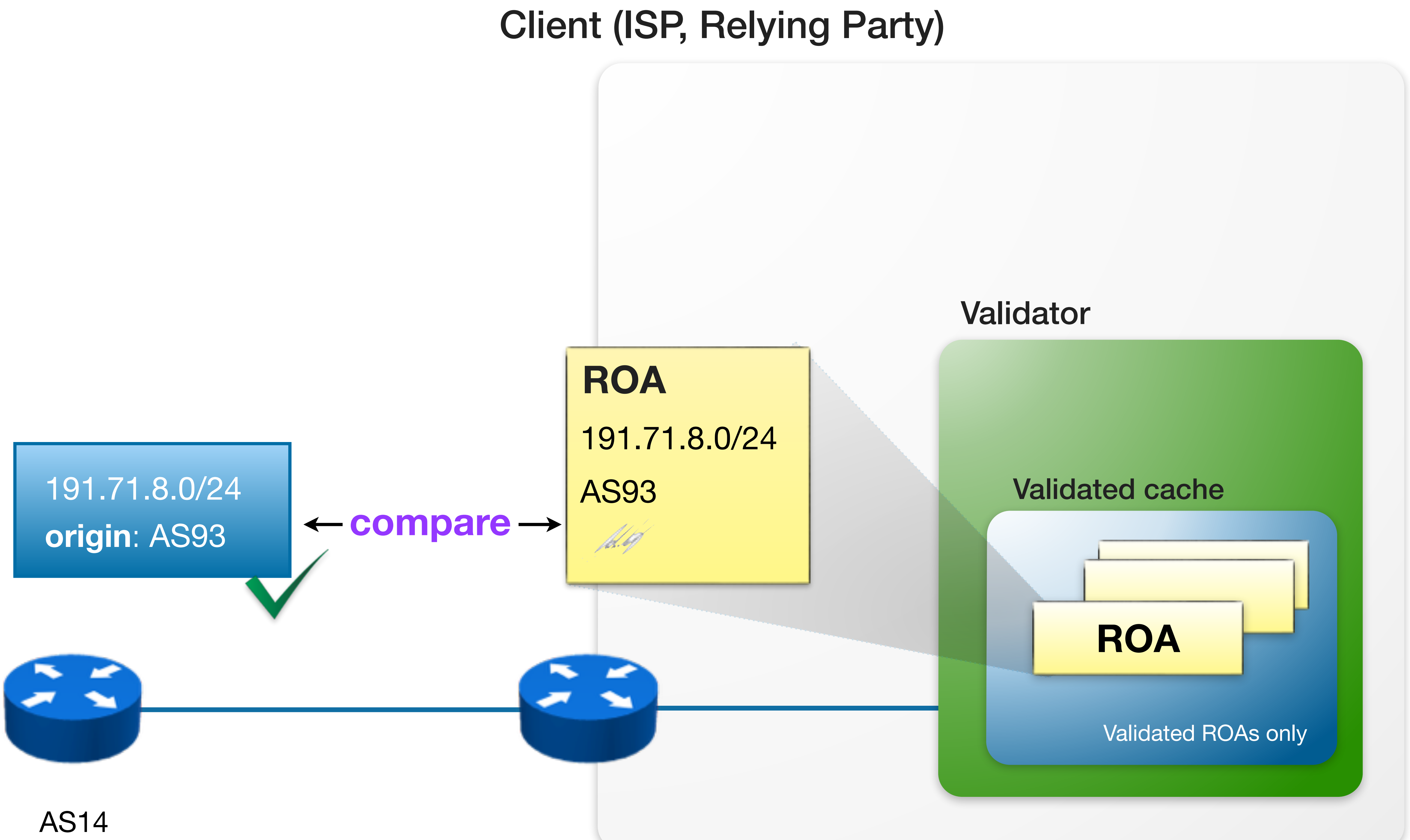
Modifying the Validated Cache

- The RIPE NCC Validator allows you to manually override the validation process
- Adding an ignore filter will ignore all ROAs for a given prefix
  - The end result is the validation state will be “unknown”
- Creating a whitelist entry for a prefix and ASN will locally create a valid ROA
  - The end result is the validation state becomes “valid”

Router Integration

- The Relying Party's router can connect and download the cache from the validator
  - Router can then compare any BGP announcements to the list of valid ROAs in the validated cache

BGP Verification



Results of BGP Verification

- valid
  - There is a ROA in the validated cache that matches the BGP announcement of the peer, size matches too
- unknown
  - There is no ROA for that prefix in the cache
- invalid
  - There is a ROA for the prefix, but for a different AS
  - The size doesn’t match

ROA vs Announcement

- Invalid ROA
  - The ROA in the repository cannot be validated by the client (ISP) so it is not included in the validated cache
- Invalid BGP announcement
  - There is a ROA in validated cache for that prefix but for a different AS.
  - Or the max length doesn’t match.
- If no ROA in the cache then announcement is “unknown”

You are in control

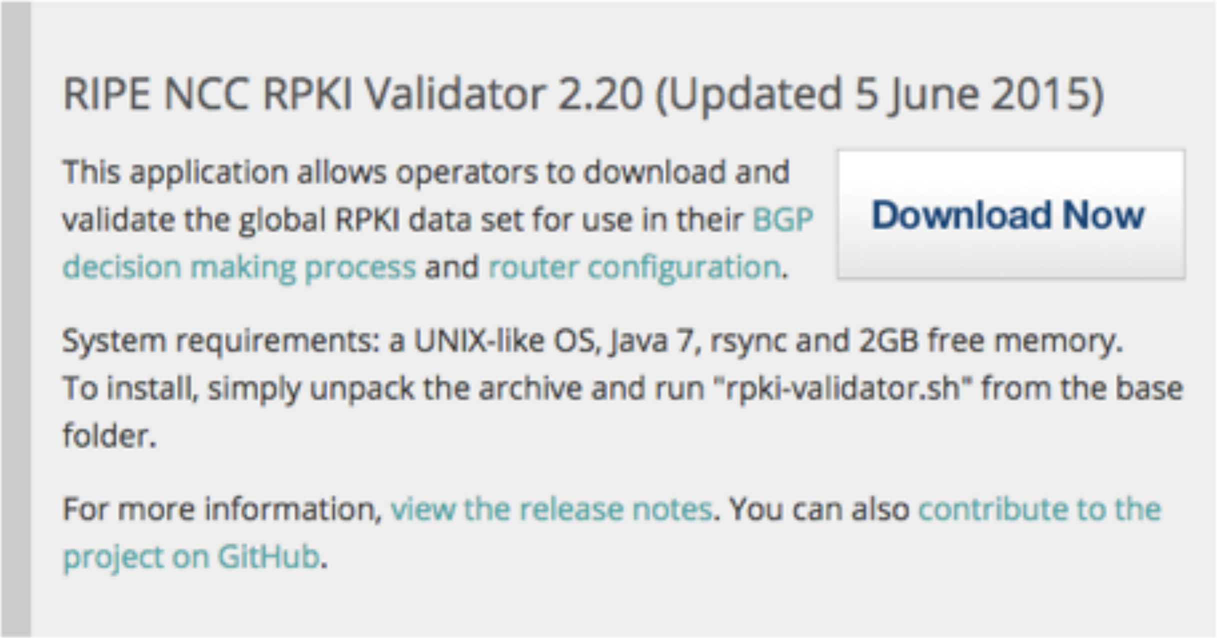
- As an announcer/LIR
  - You choose if you want certification
  - You choose if you want to create ROAs
  - You choose AS, max length
- As a Relying Party
  - You can choose if you use the validator
  - You can override the lists of valid ROAs in the cache, adding or removing valid ROAs locally
  - You can choose to make any routing decisions based on the results of the BGP Verification (valid/invalid/unknown)

RPKI
RIPE NCC Validator

Demo

Download the Validator

- http://www.ripe.net/certification



RIPE NCC RPKI Validator 2.20 (Updated 5 June 2015)

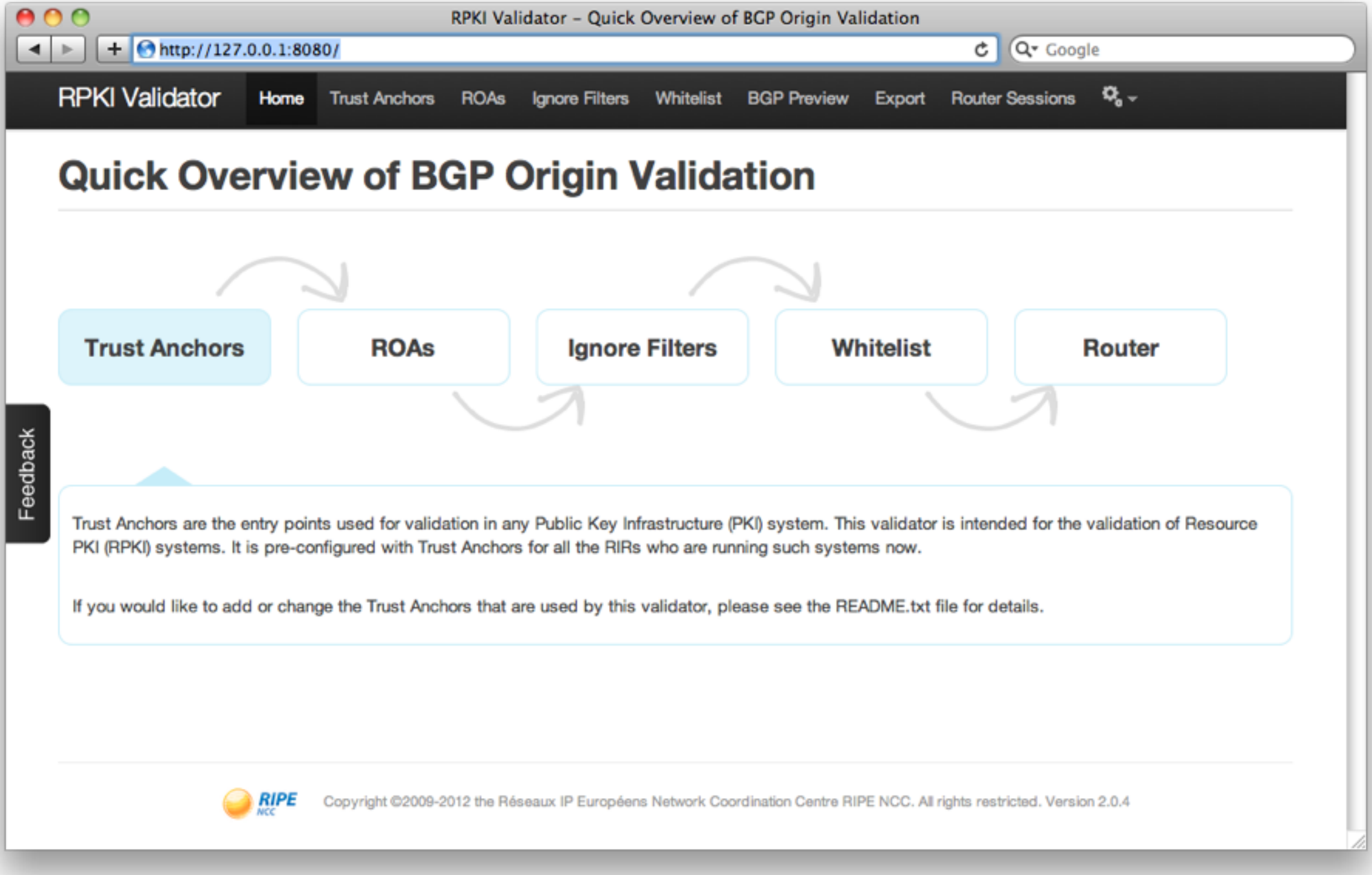
This application allows operators to download and validate the global RPKI data set for use in their BGP decision making process and router configuration.

System requirements: a UNIX-like OS, Java 7, rsync and 2GB free memory. To install, simply unpack the archive and run "rpki-validator.sh" from the base folder.

For more information, view the release notes. You can also contribute to the project on GitHub.

- No Installation required
  - Unzip the package
  - Run the program: rpki-validator.sh start
- Interface available on localhost port 8080

The Web Interface



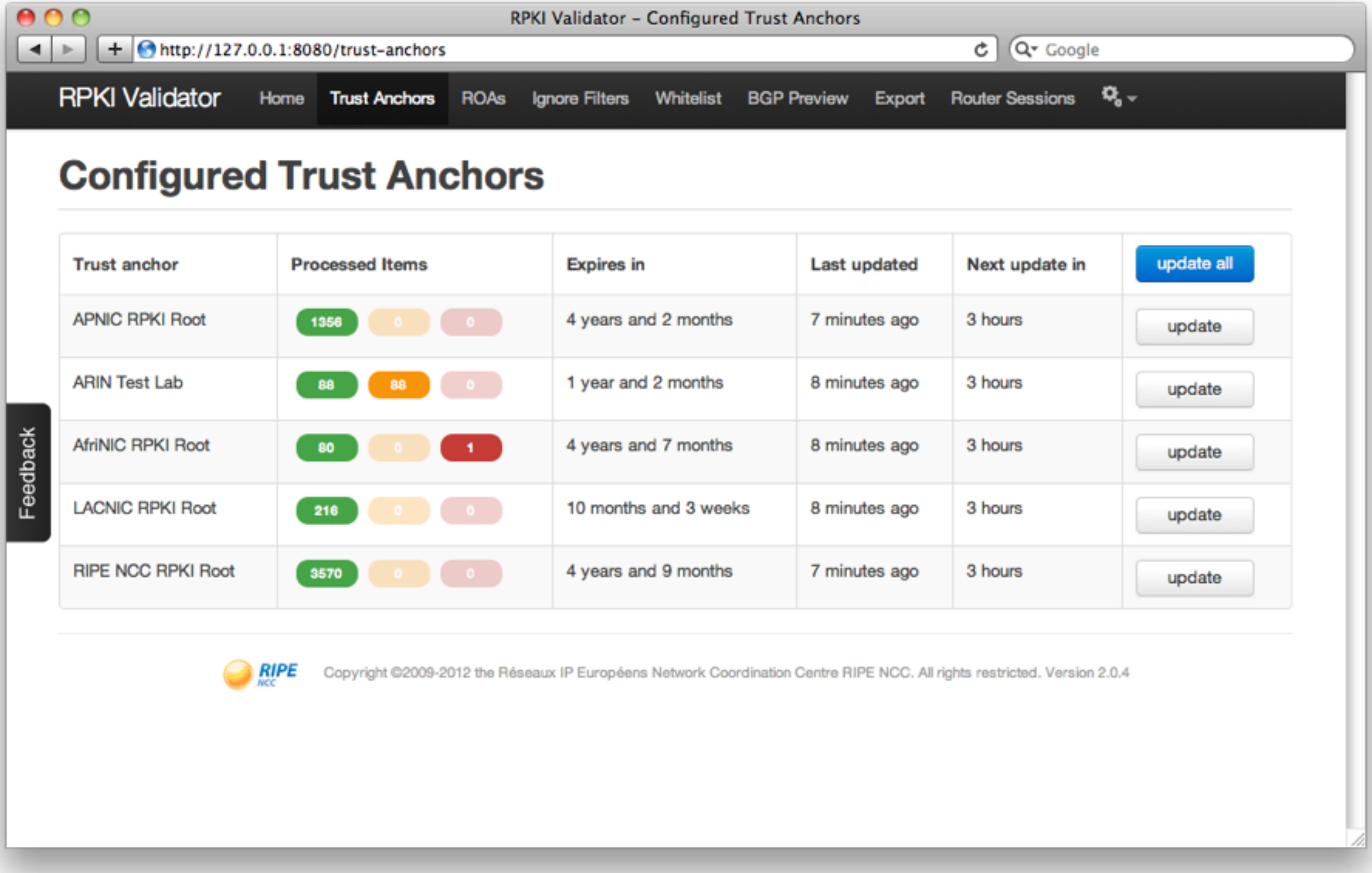
Quick Overview of BGP Origin Validation

Trust Anchors are the entry points used for validation in any Public Key Infrastructure (PKI) system. This validator is intended for the validation of Resource PKI (RPKI) systems. It is pre-configured with Trust Anchors for all the RIRs who are running such systems now.

If you would like to add or change the Trust Anchors that are used by this validator, please see the README.txt file for details.

Copyright ©2009-2012 the Réseaux IP Européens Network Coordination Centre RIPE NCC. All rights restricted. Version 2.0.4

Trust Anchors

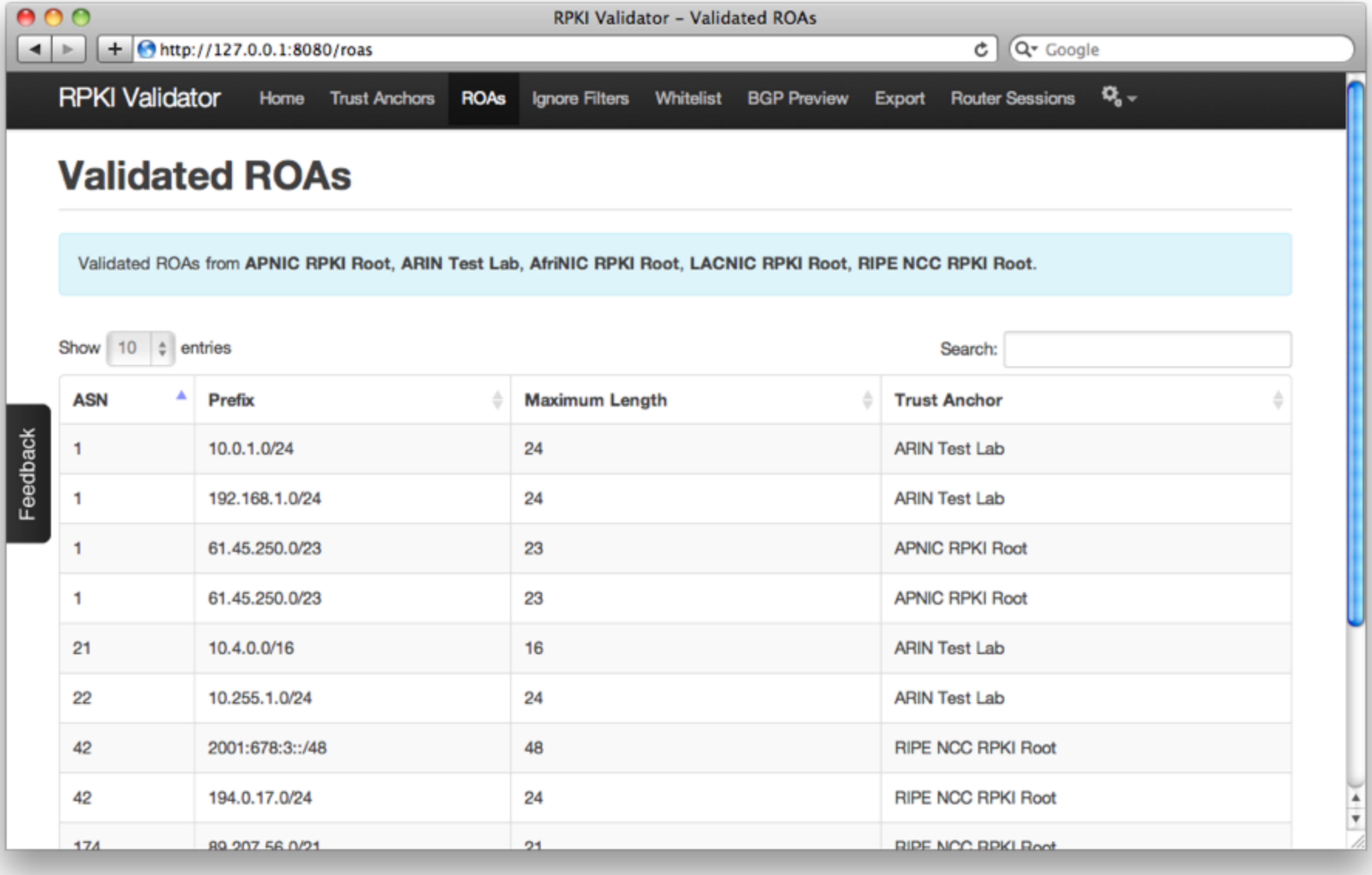


Configured Trust Anchors

Trust anchor	Processed Items	Expires in	Last updated	Next update in	update all
APNIC RPKI Root	1356 0 0	4 years and 2 months	7 minutes ago	3 hours	update
ARIN Test Lab	88 88 0	1 year and 2 months	8 minutes ago	3 hours	update
AfriNIC RPKI Root	80 0 1	4 years and 7 months	8 minutes ago	3 hours	update
LACNIC RPKI Root	216 0 0	10 months and 3 weeks	8 minutes ago	3 hours	update
RIPE NCC RPKI Root	3570 0 0	4 years and 9 months	7 minutes ago	3 hours	update

RIPE NCC Copyright ©2009-2012 the Réseaux IP Européens Network Coordination Centre RIPE NCC. All rights restricted. Version 2.0.4

Validated Cache

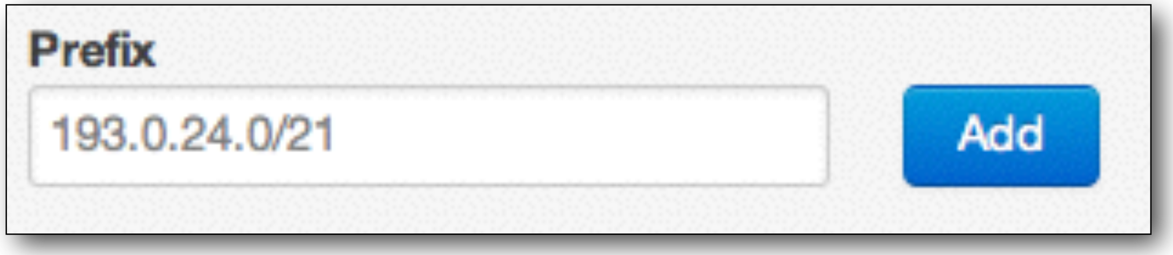


Validated ROAs

Validated ROAs from **APNIC RPKI Root, ARIN Test Lab, AfriNIC RPKI Root, LACNIC RPKI Root, RIPE NCC RPKI Root**.

ASN	Prefix	Maximum Length	Trust Anchor
1	10.0.1.0/24	24	ARIN Test Lab
1	192.168.1.0/24	24	ARIN Test Lab
1	61.45.250.0/23	23	APNIC RPKI Root
1	61.45.250.0/23	23	APNIC RPKI Root
21	10.4.0.0/16	16	ARIN Test Lab
22	10.255.1.0/24	24	ARIN Test Lab
42	2001:678:3::/48	48	RIPE NCC RPKI Root
42	194.0.17.0/24	24	RIPE NCC RPKI Root
174	89.207.56.0/21	21	RIPE NCC RPKI Root

Creating a Whitelist

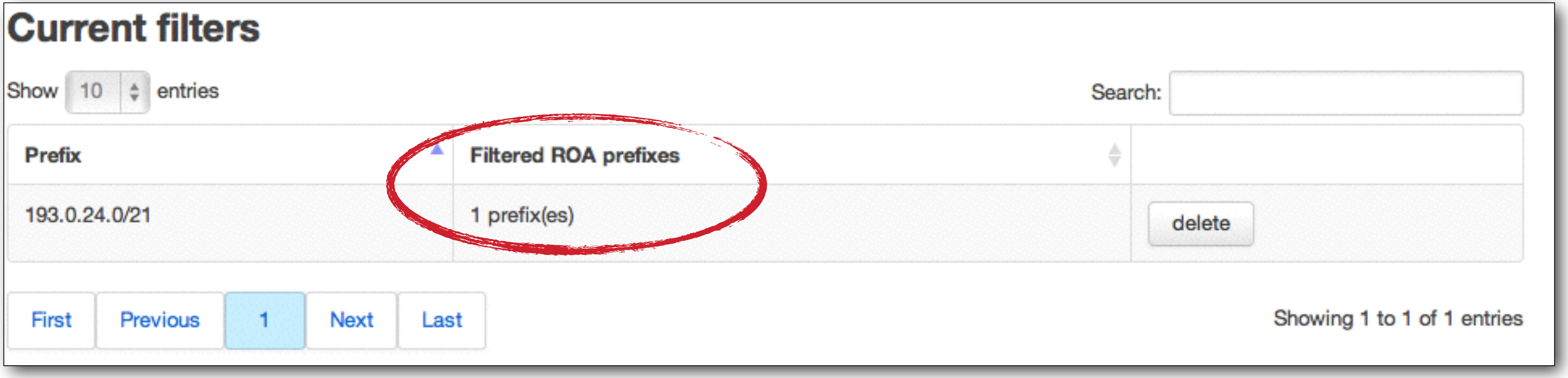


Prefix

193.0.24.0/21

Insert the prefix and click “Add”

This locally creates a valid (but fake) ROA



Current filters

Show 10 entries

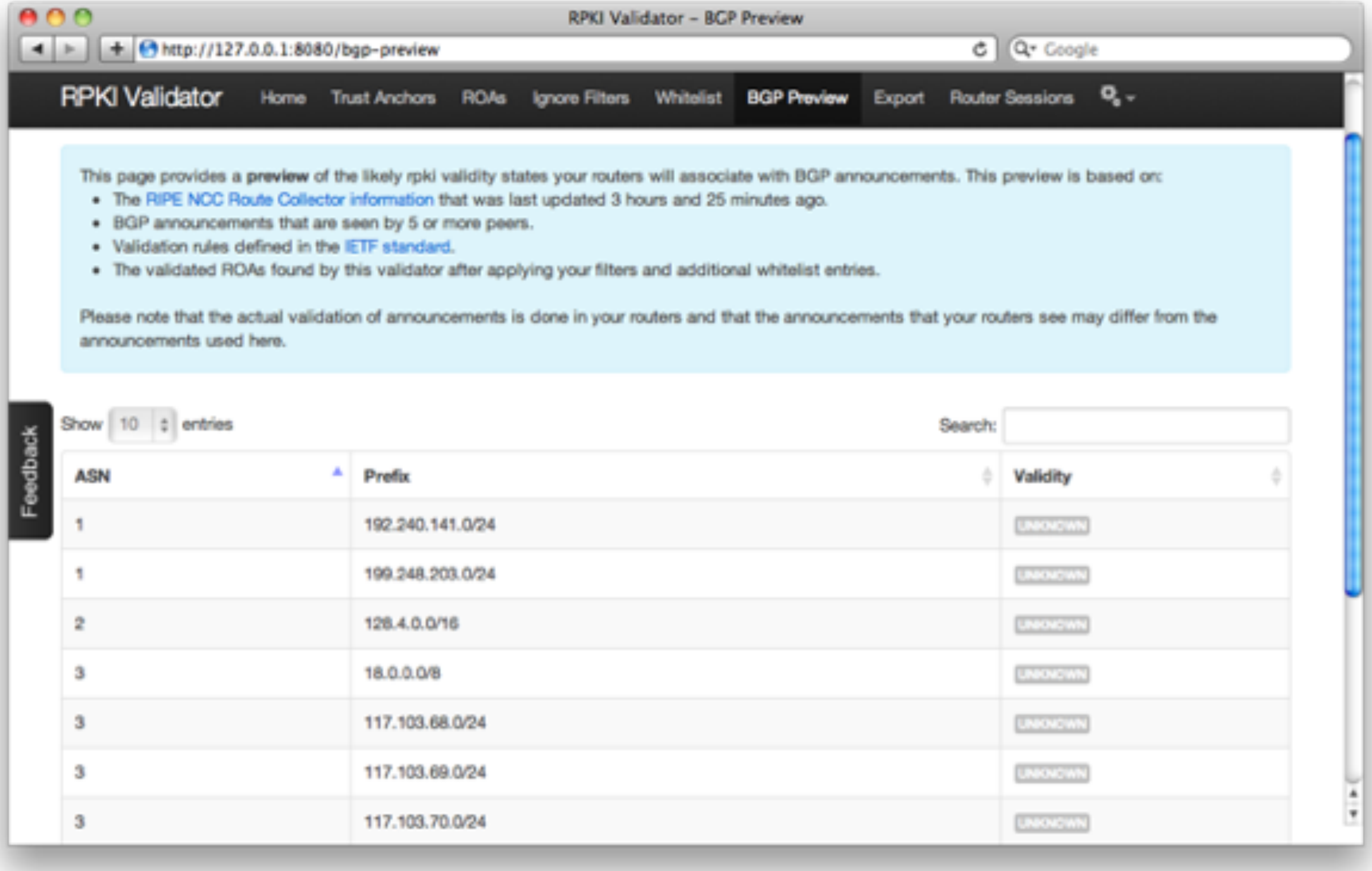
Search:

Prefix	Filtered ROA prefixes	
193.0.24.0/21	1 prefix(es)	delete

Showing 1 to 1 of 1 entries

BGP Preview

- The validator downloads a copy of the RIS
  - Allows you to get a hint of what would happen
  - RIS view might be different from your routing table



This page provides a **preview** of the likely rpki validity states your routers will associate with BGP announcements. This preview is based on:

- The RIPE NCC Route Collector information that was last updated 3 hours and 25 minutes ago.
- BGP announcements that are seen by 5 or more peers.
- Validation rules defined in the IETF standard.
- The validated ROAs found by this validator after applying your filters and additional whitelist entries.

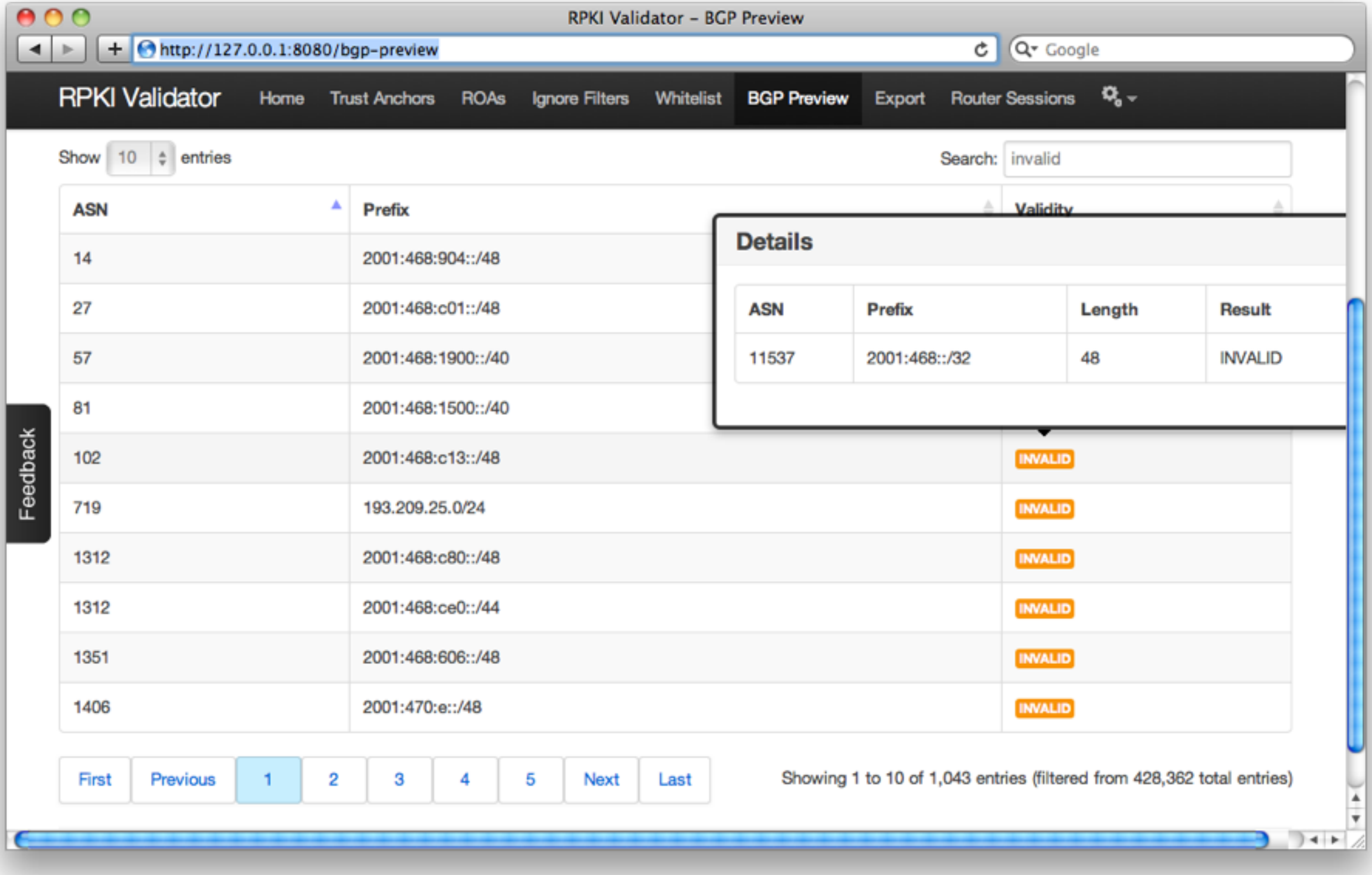
Please note that the actual validation of announcements is done in your routers and that the announcements that your routers see may differ from the announcements used here.

Show 10 entries

Search:

ASN	Prefix	Validity
1	192.240.141.0/24	UNKNOWN
1	199.248.203.0/24	UNKNOWN
2	128.4.0.0/16	UNKNOWN
3	18.0.0.0/8	UNKNOWN
3	117.103.68.0/24	UNKNOWN
3	117.103.69.0/24	UNKNOWN
3	117.103.70.0/24	UNKNOWN

BGP Preview Detail



Show 10 entries

Search: invalid

ASN	Prefix	Validity
14	2001:468:904::/48	
27	2001:468:c01::/48	
57	2001:468:1900::/40	
81	2001:468:1500::/40	
102	2001:468:c13::/48	INVALID
719	193.209.25.0/24	INVALID
1312	2001:468:c80::/48	INVALID
1312	2001:468:ce0::/44	INVALID
1351	2001:468:606::/48	INVALID
1406	2001:470:e::/48	INVALID

Details

ASN	Prefix	Length	Result
11537	2001:468::/32	48	INVALID

Showing 1 to 10 of 1,043 entries (filtered from 428,362 total entries)

RPKI Quiz

Exercise 5

RPKI
Router Integration

Section 11

Exporting the Validated Cache

- Router sessions
  - Validator listens on 8282 for RPKI-RTR Protocol
  - Routers can connect and download the cache

- Export function
  - Allows you to download a CSV with the cache
  - Can be integrated with your internal workflow
  - Use for statistics or spotting anomalies

RPKI Support in Routers

- **RPKI** and **RPKI-RTR** are an IETF standards
  - All router vendors can implement it
- **Cisco** support:
  - XR 4.2.1 (CRS-x, ASR9000, c12K) / XR 5.1.1 (NCS6000, XRv)
  - XE 3.5 (C7200, c7600, ASR1K, CSR1Kv, ASR9k, ME3600…)
  - IOS15.2(1)S
- **Juniper** has support since version 12.2
- **Alcatel Lucent** has support since SR-OS 12.0 R4
- **Quagga** has support through BGP-SRX
- **BIRD** has support for ROA but does not do RPKI-RTR

Public Testbeds

- Cisco (hosted by the RIPE NCC)
  - Telnet to rpki-rtr.ripe.net
  - User: ripe, no password
- Juniper (hosted by Kaia Global Networks)
  - Telnet to 193.34.50.25 or 193.34.50.26
  - Username: rpki, password: testbed

http://www.ripe.net/certification

Community Activity

- Open source RPKI Tools
  - rpki.net
- SURFnet RPKI Dashboard
  - rpki.surfnet.nl
- BGPMon Route Monitoring
  - bgpmon.net/services/route-monitoring/
- RIPE NCC Github
  - github.com/RIPE-NCC

Questions

RIPE NCC Academy

Graduate to the next level!

http://academy.ripe.net

Feedback

http://www.ripe.net/training/rs/survey

Follow us!

@TrainingRIPENCC

The End!

Край

Y Diwedd

Fí

Finis

النهاية

Соңы

Վերջ

Liðugt

Ende

Finvezh

Кінець

Konec

Kraj

Ënn

Fund

پایان

Kpaj

Lõpp

Beigas

Vége

Son

An Críoch

הסוף

Fine

Endir

Sfârşit

Fin

Τέλος

Einde

Конец

Slut

Slutt

დასასრული

Pabaiga

Tmiem

Koniec

Fim

Amaia

Loppu