RPKI Tutorial

MANRS RPKI Week

July 2022

Massimiliano Stucchi

stucchi@isoc.org

Agenda

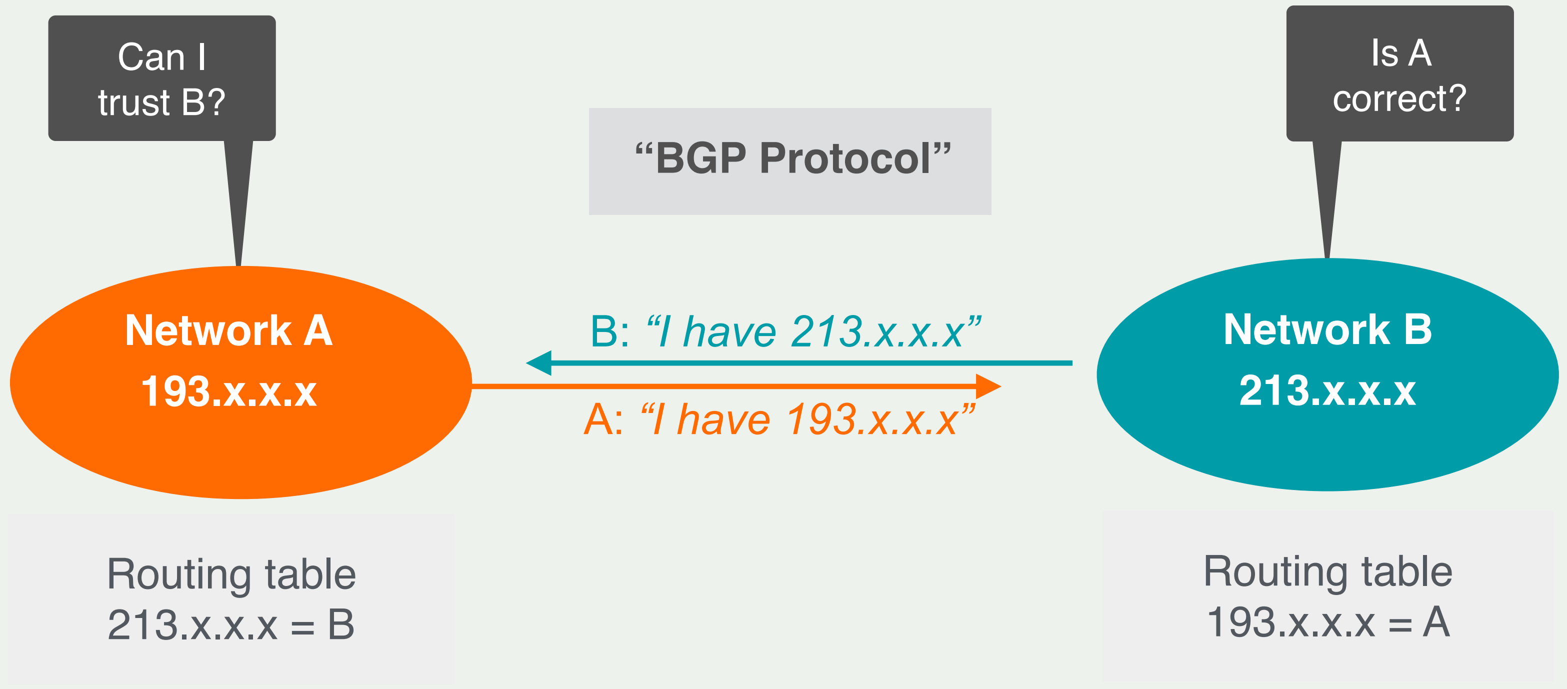
Introduction

ROAs
- **Demo:** Create ROAs

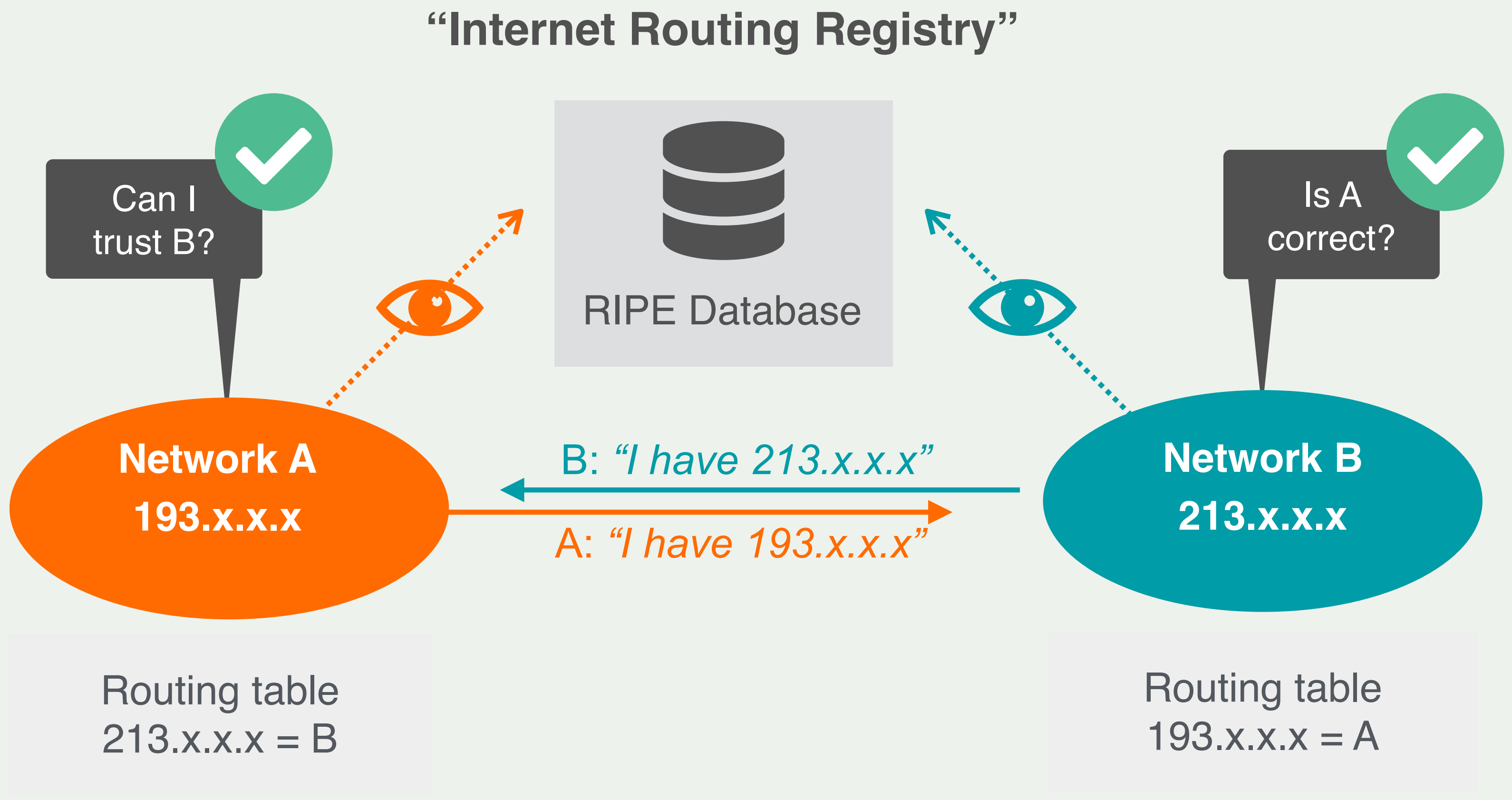
Deploying RPKI Validators
- **Demo:** Running Validators

Validation
- **Demo:** Setting up BGP Origin Validation
- **Demo:** Discarding BGP Invalids

Routing on the Internet



How can you have secure routing?



Problem Statement

- Some IRR data cannot be fully trusted
  - Accuracy
  - Incomplete data
  - Lack of maintenance

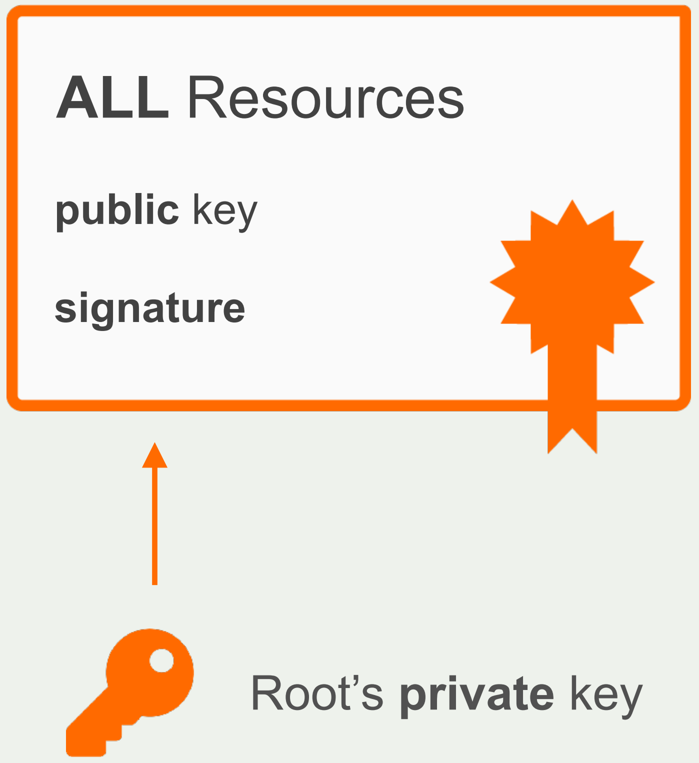
- Third party databases are widely used
  - No verification of who holds IPs/ASNs

A Short History

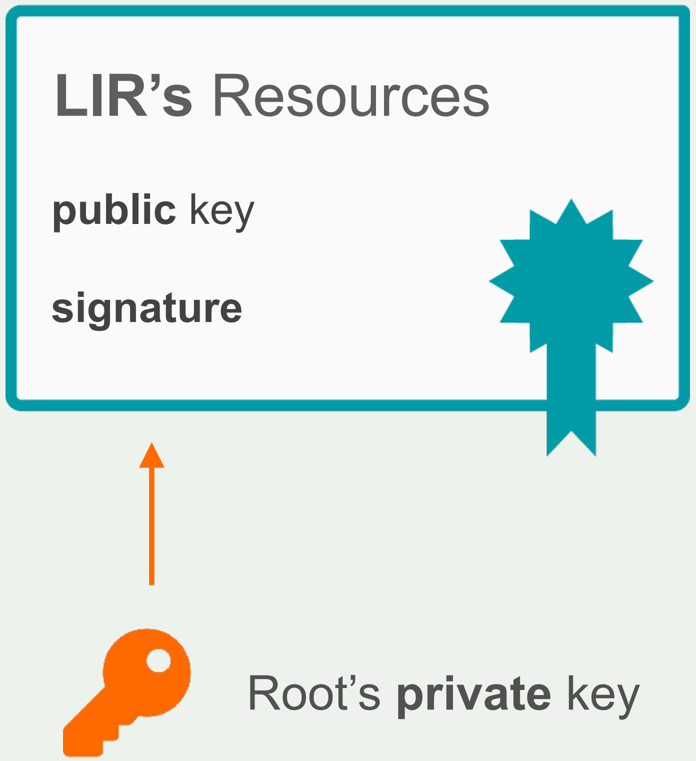
- Operated since 2008 by all RIRs
  - Community-driven standardisation (IETF)

- Adds crypto-security to IP addresses and ASNs
  - Provides data you can trust

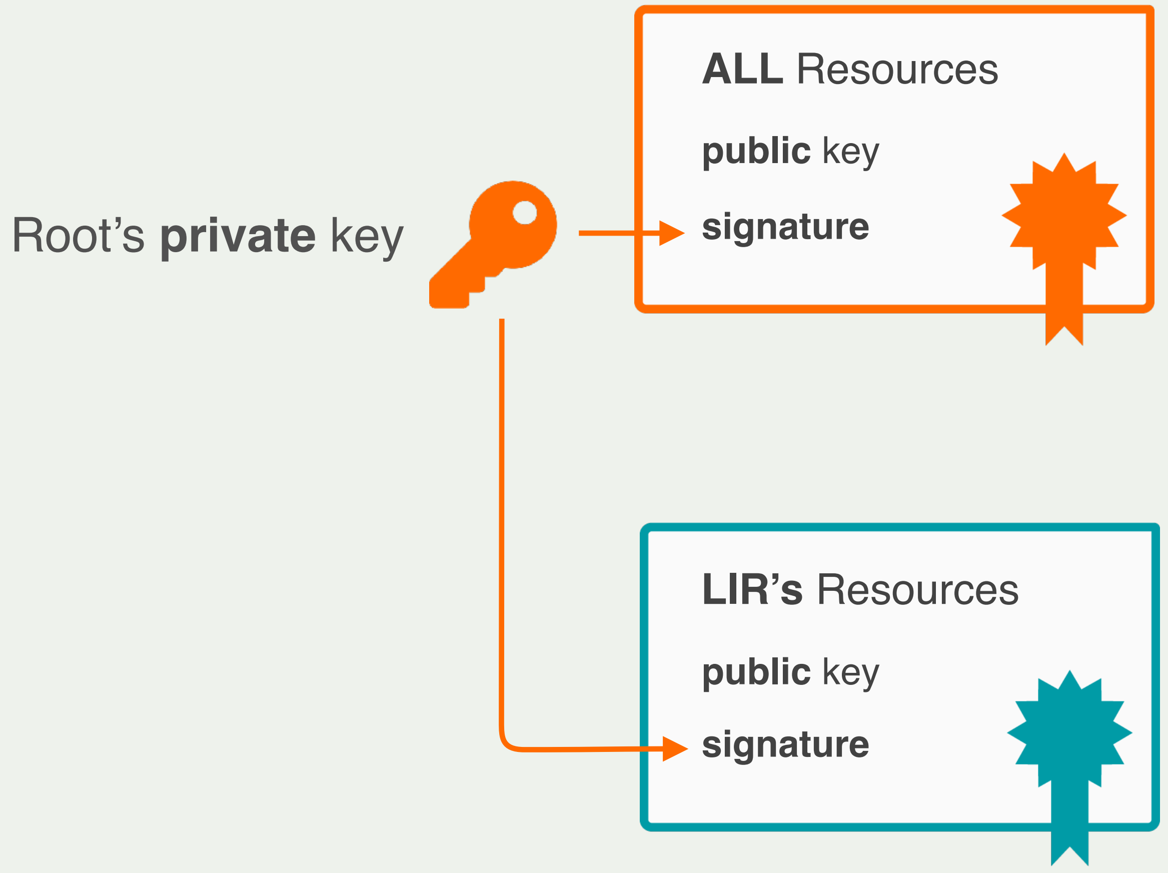
RPKI Chain of Trust



RPKI Chain of Trust



RPKI Chain of Trust



Resource Public Key Infrastructure

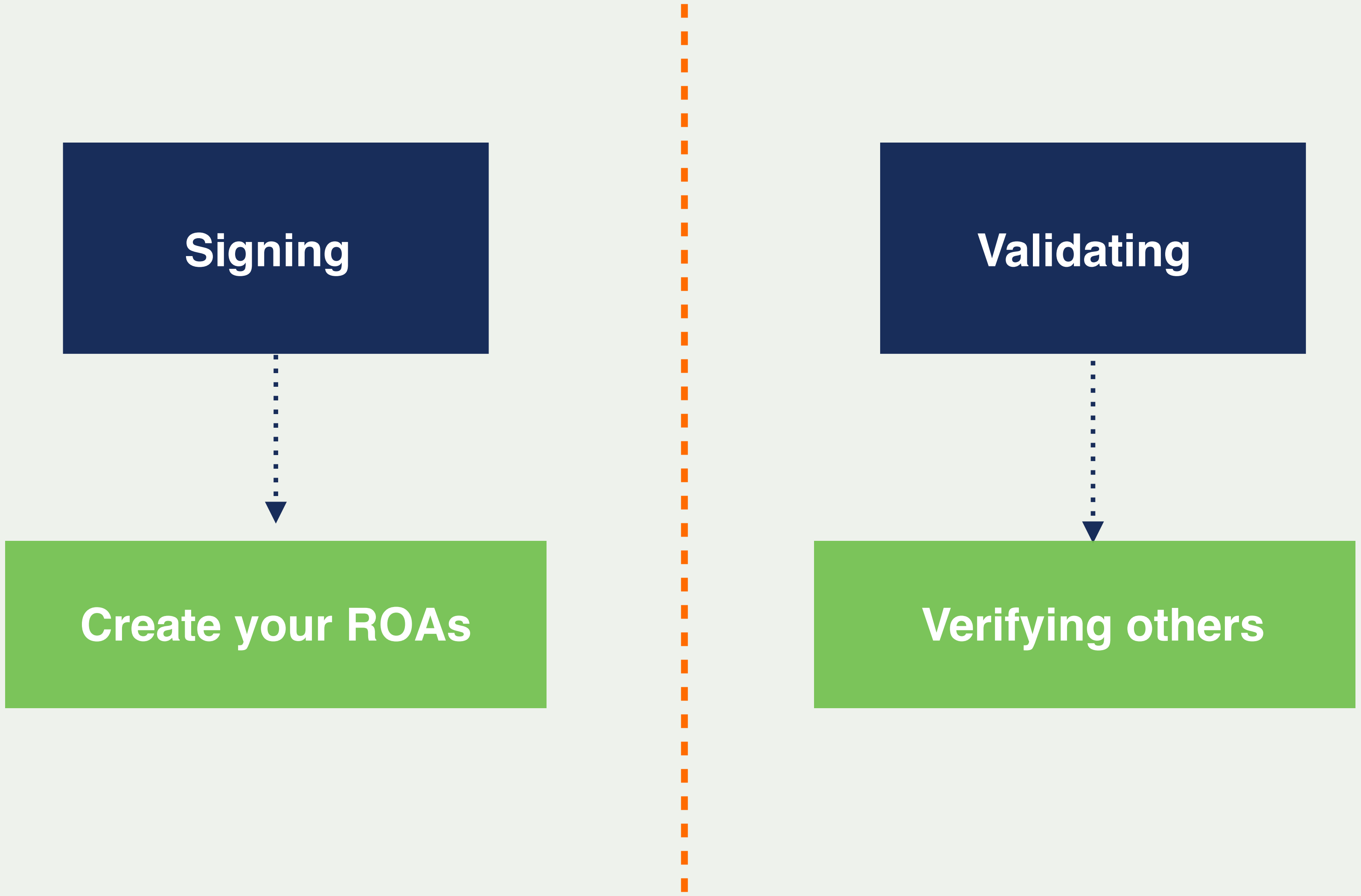
Ties IP addresses and ASNs to public keys

Follows the hierarchy of the registries

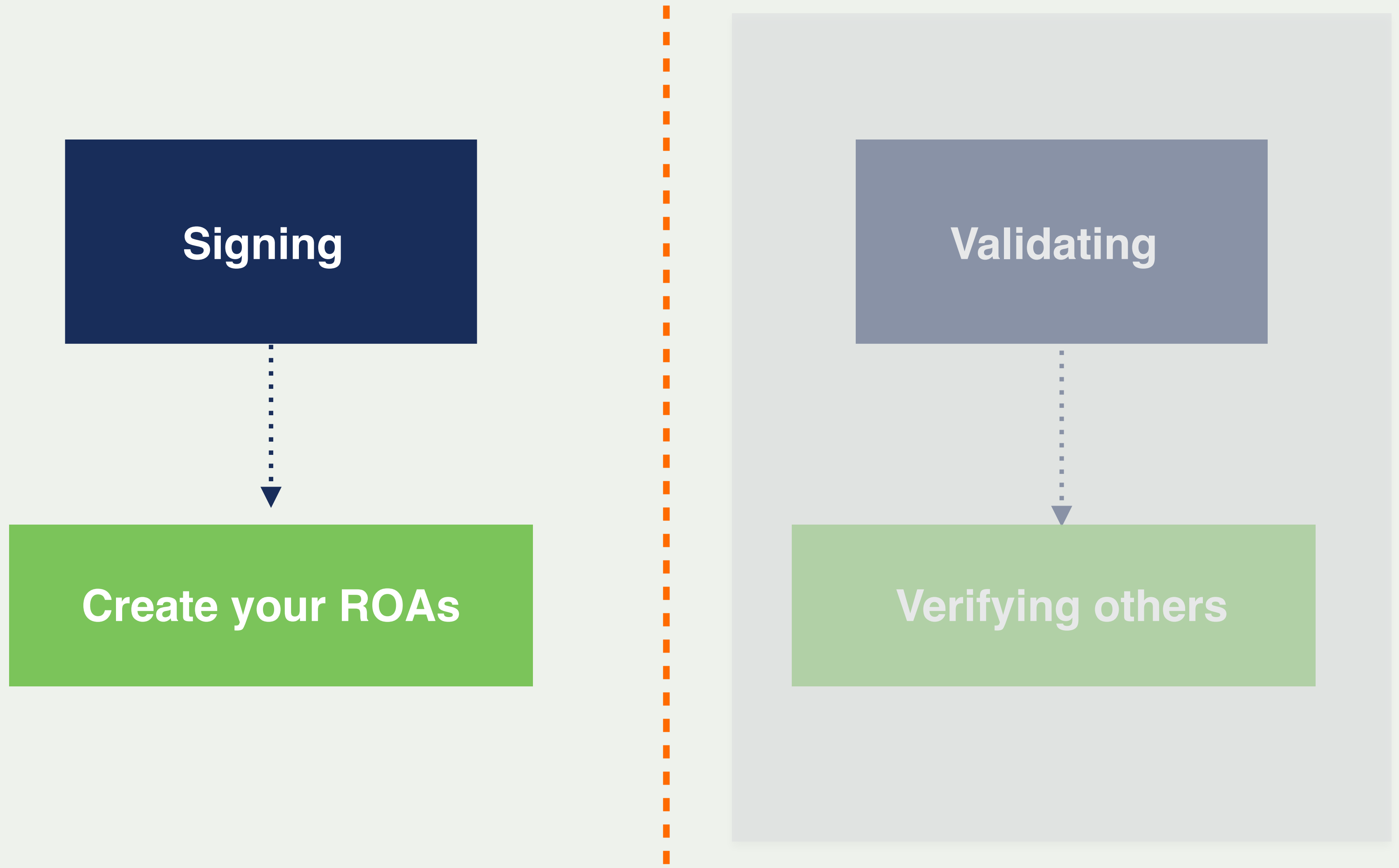
Authorised statements from resource holders
“ASN X is authorised to announce my Prefix Y”
Signed, holder of Y

ROAs

Elements of RPKI

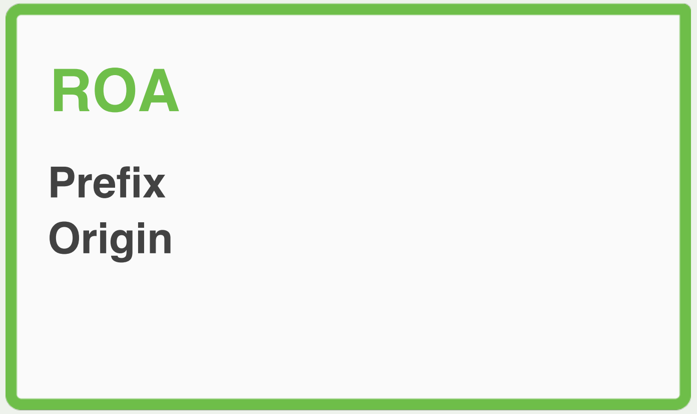


Elements of RPKI



What is a ROA ?

An **authorised statement** from a resource holder



AS Number
is authorised to announce
Prefix

What is a ROA ?

- LIRs can create a ROA for their resources

- Multiple ROAs can exist for the same prefix
    - With different origin/maxlength

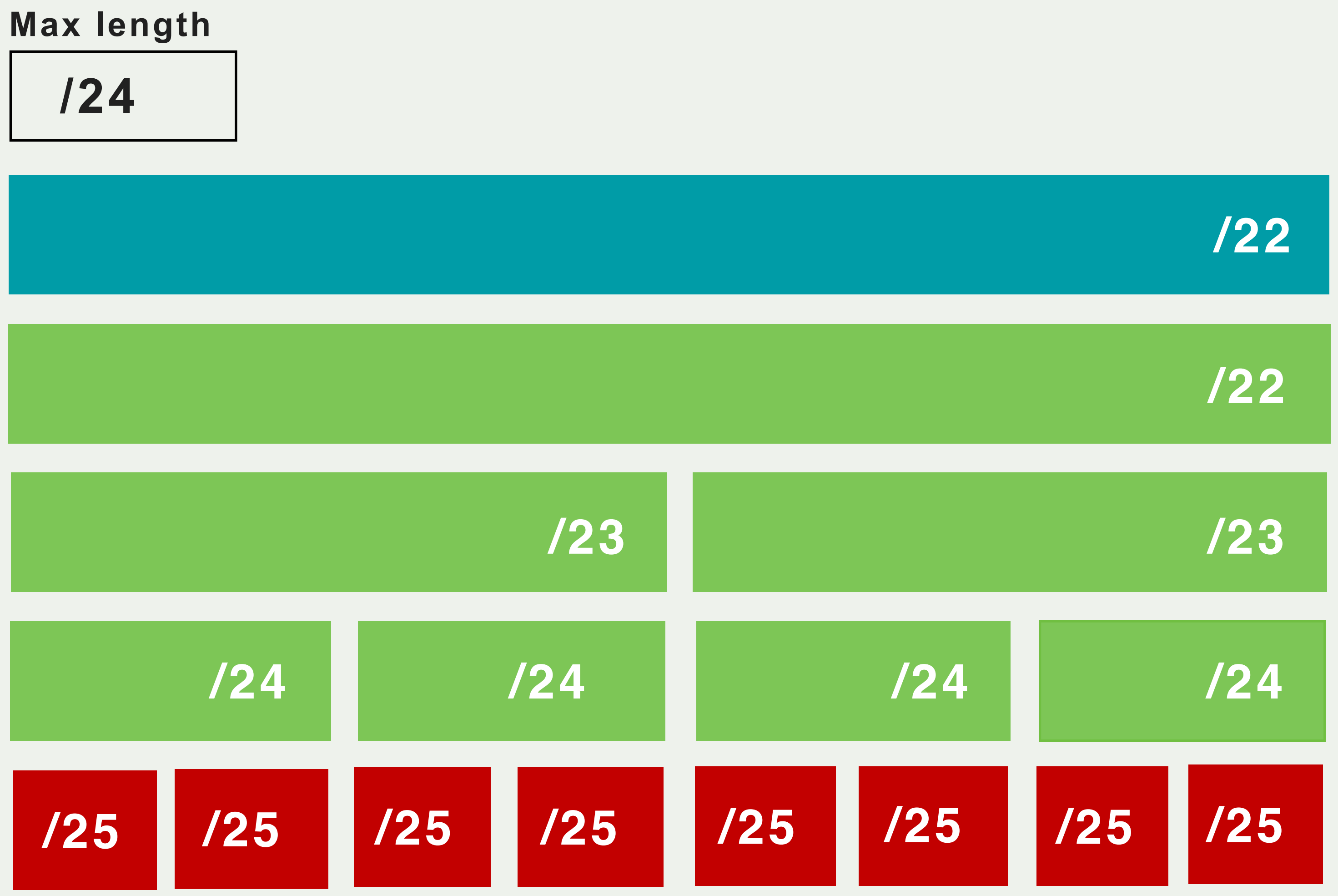
- ROAs can overlap

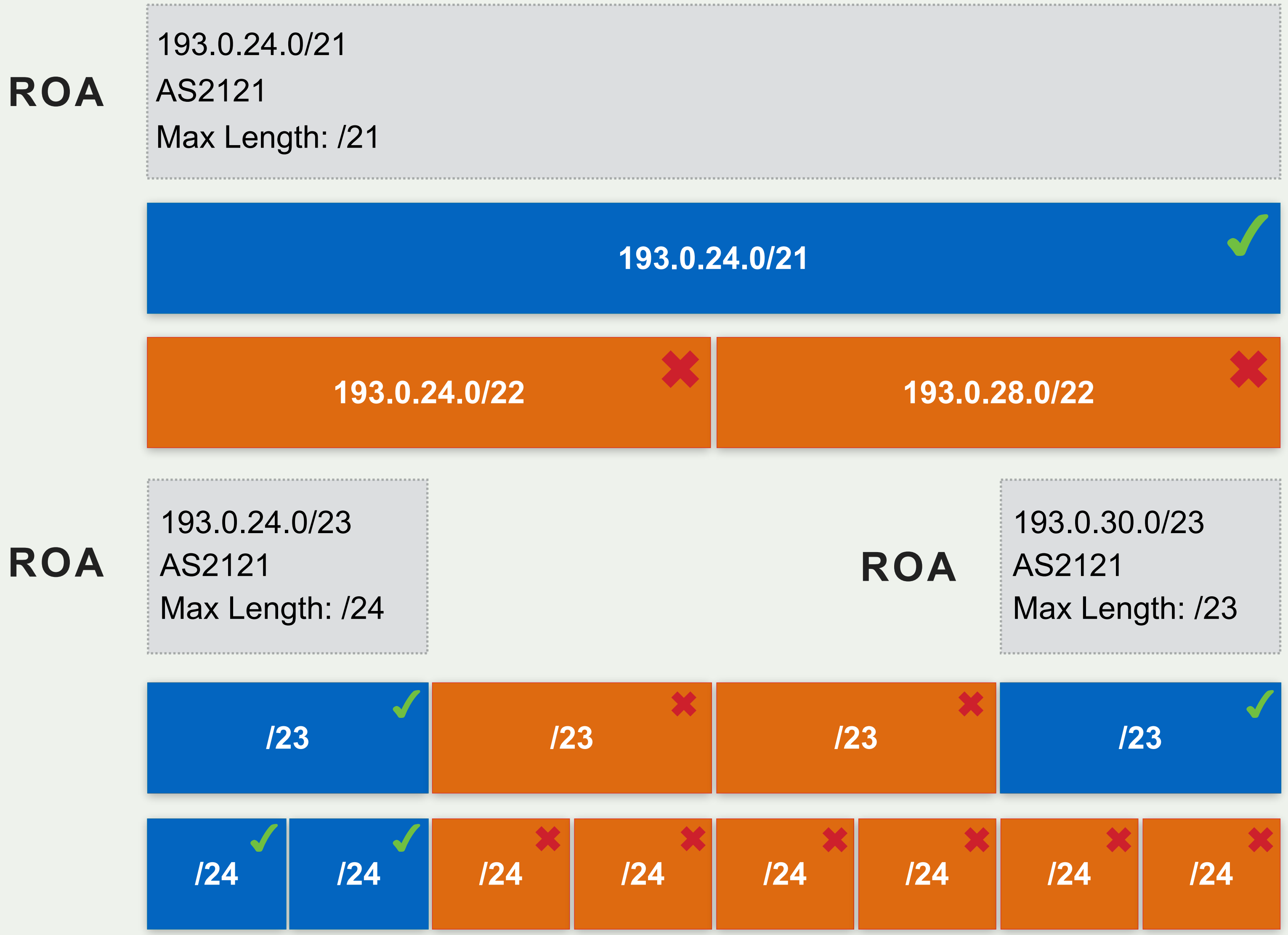
What is in a ROA ?

Route
Origin
Authorisation

Prefix	The network for which you are creating the ROA
Origin ASN	The ASN supposed to originate the BGP Announcement
Max Length	The maximum prefix length that ROA is authorised to advertise

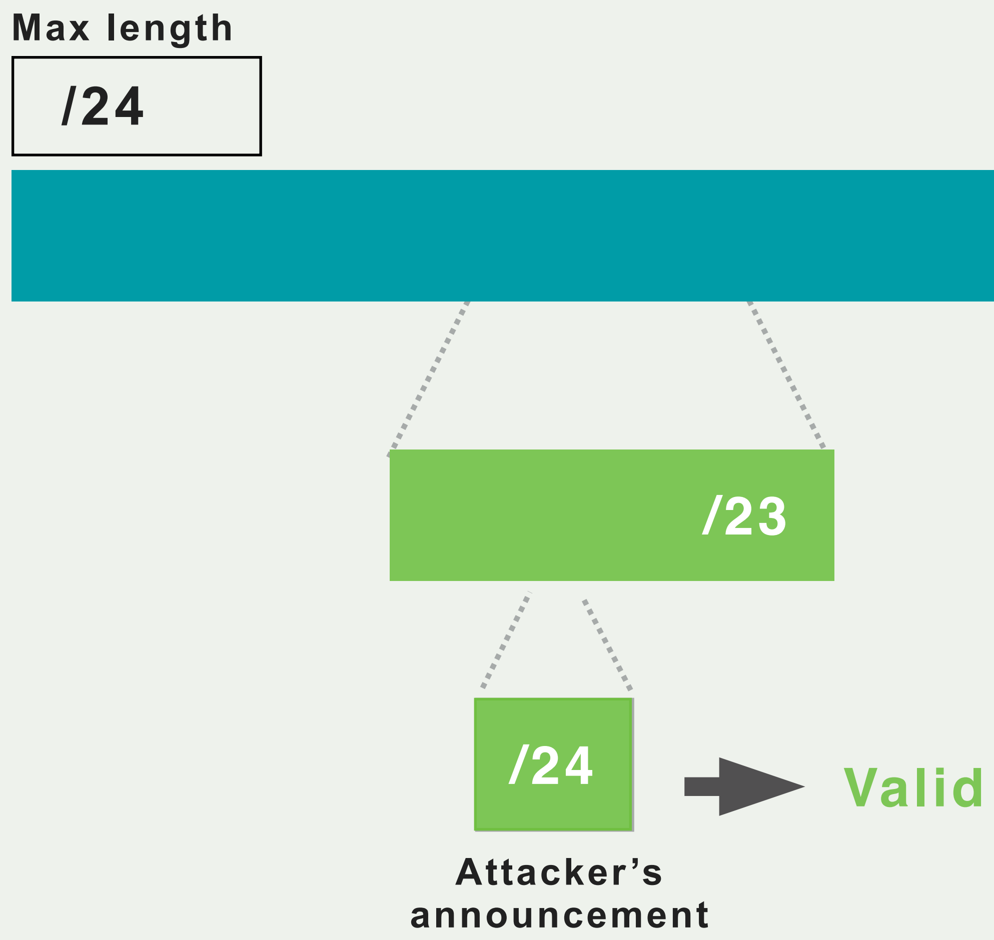
What is max-length?




ROA
193.0.24.0/21
AS2121
Max Length: /21
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 ✗
/23 ✓
/24 ✓
/24 ✓
/24 ✗
/24 ✗
/24 ✗
/24 ✗
/24 ✗
/24 ✗

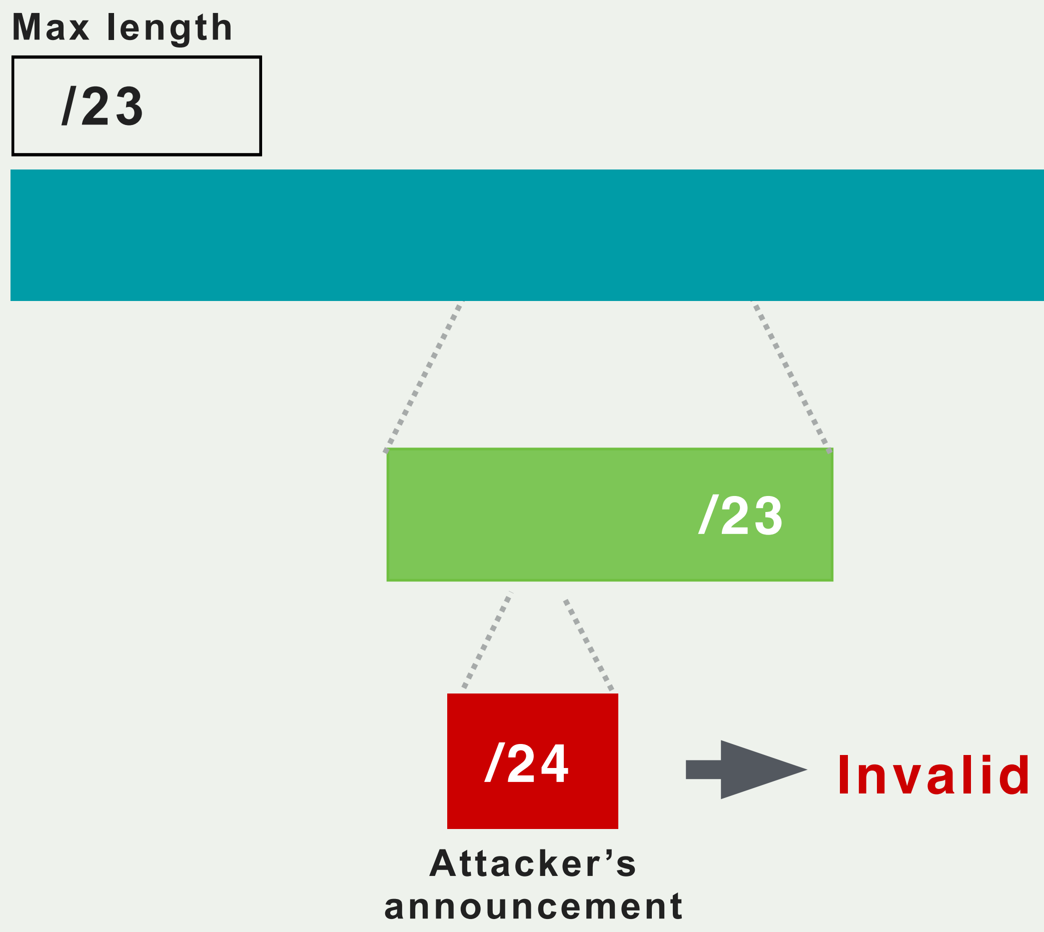
How should we use max-length?

You created a single ROA authorising the entire /22

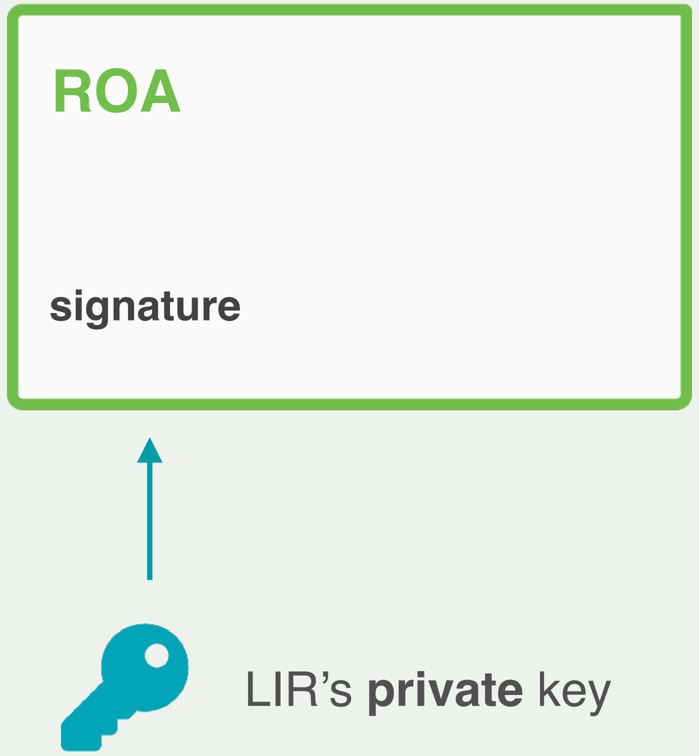


How should we use max-length?

Create ROAs for BGP announcements only



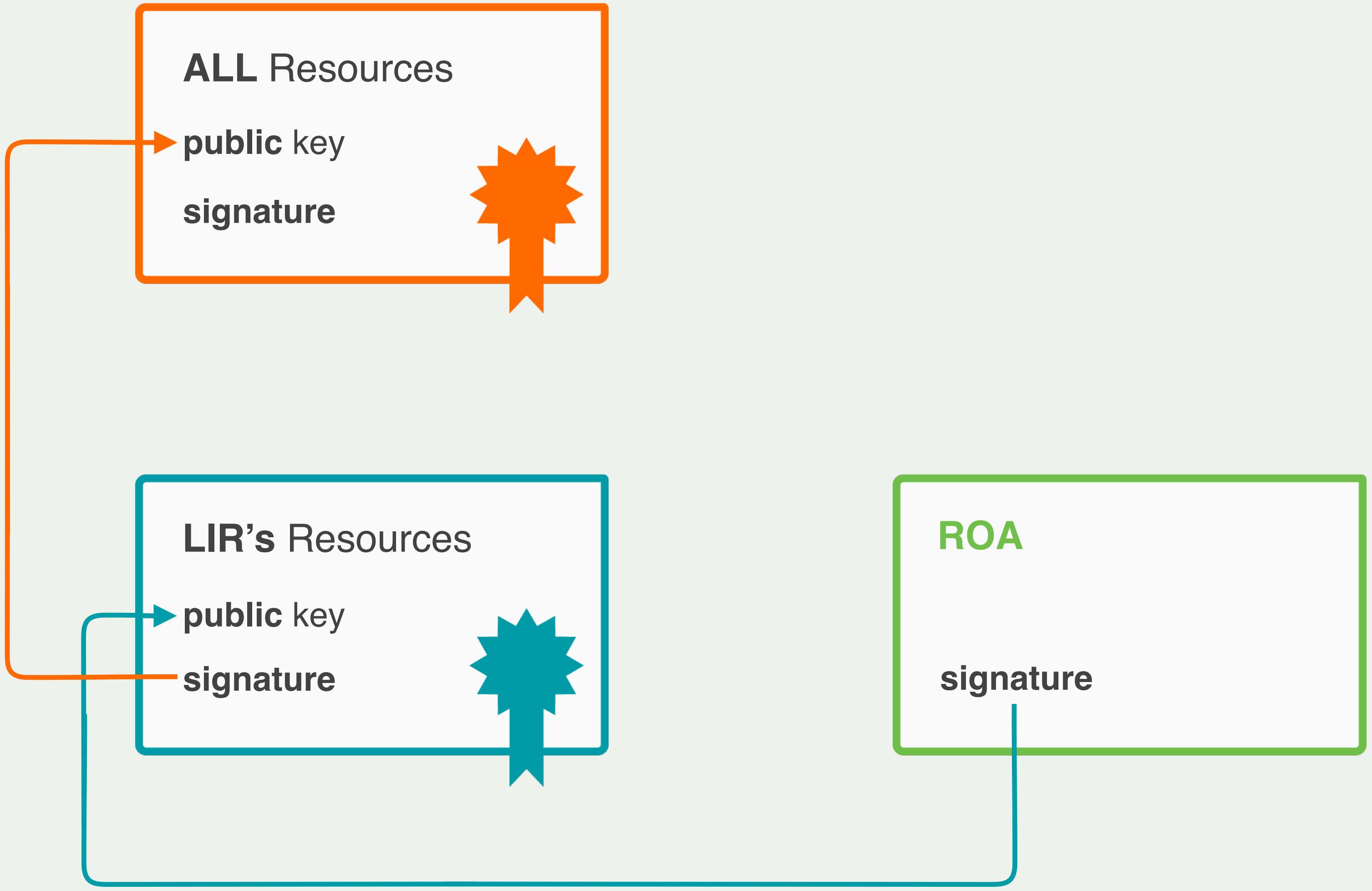
ROA Signature



LIR's **private** key

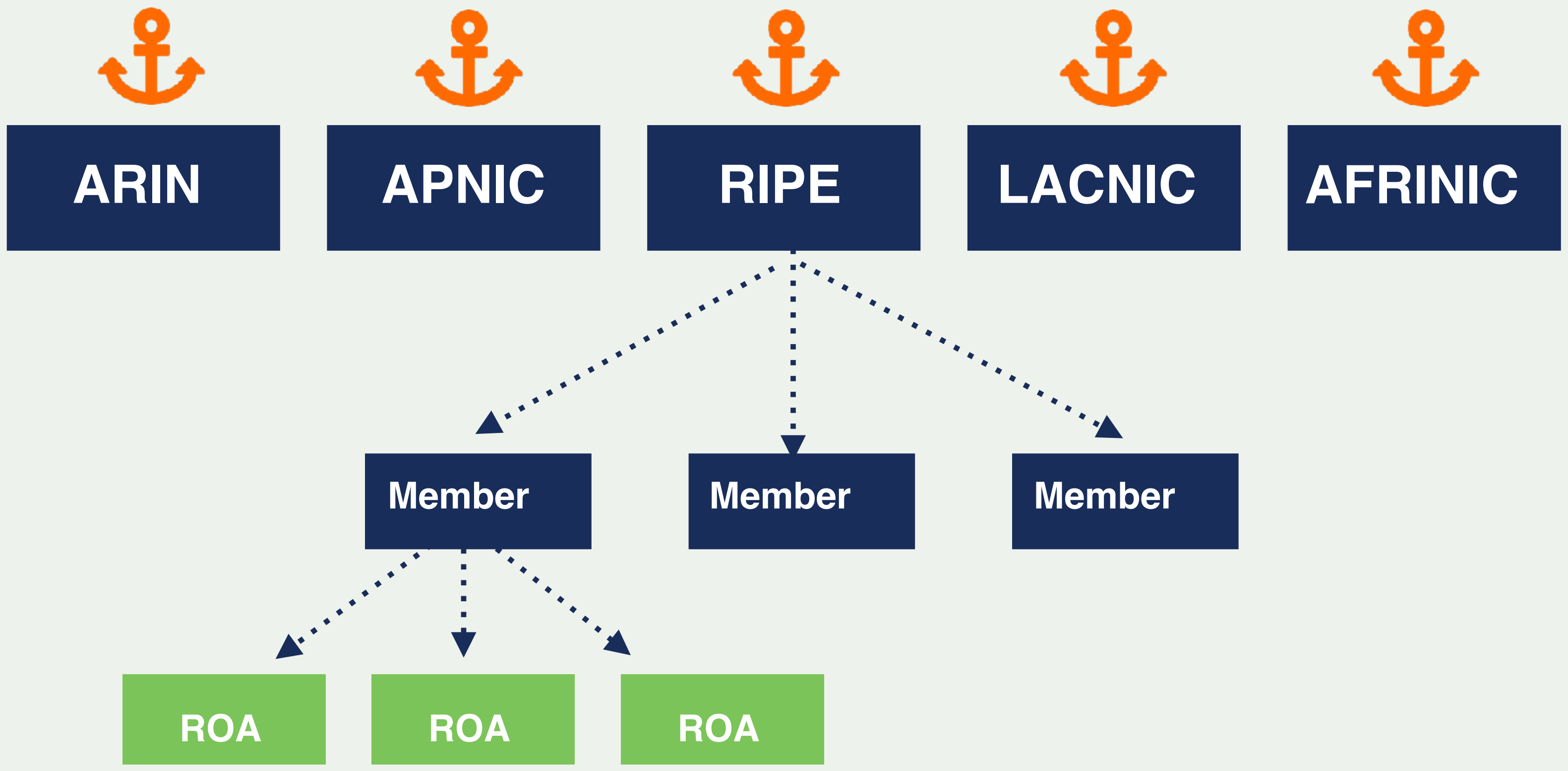
Prefix
is authorised to be announced by
AS Number

RPKI Chain of Trust

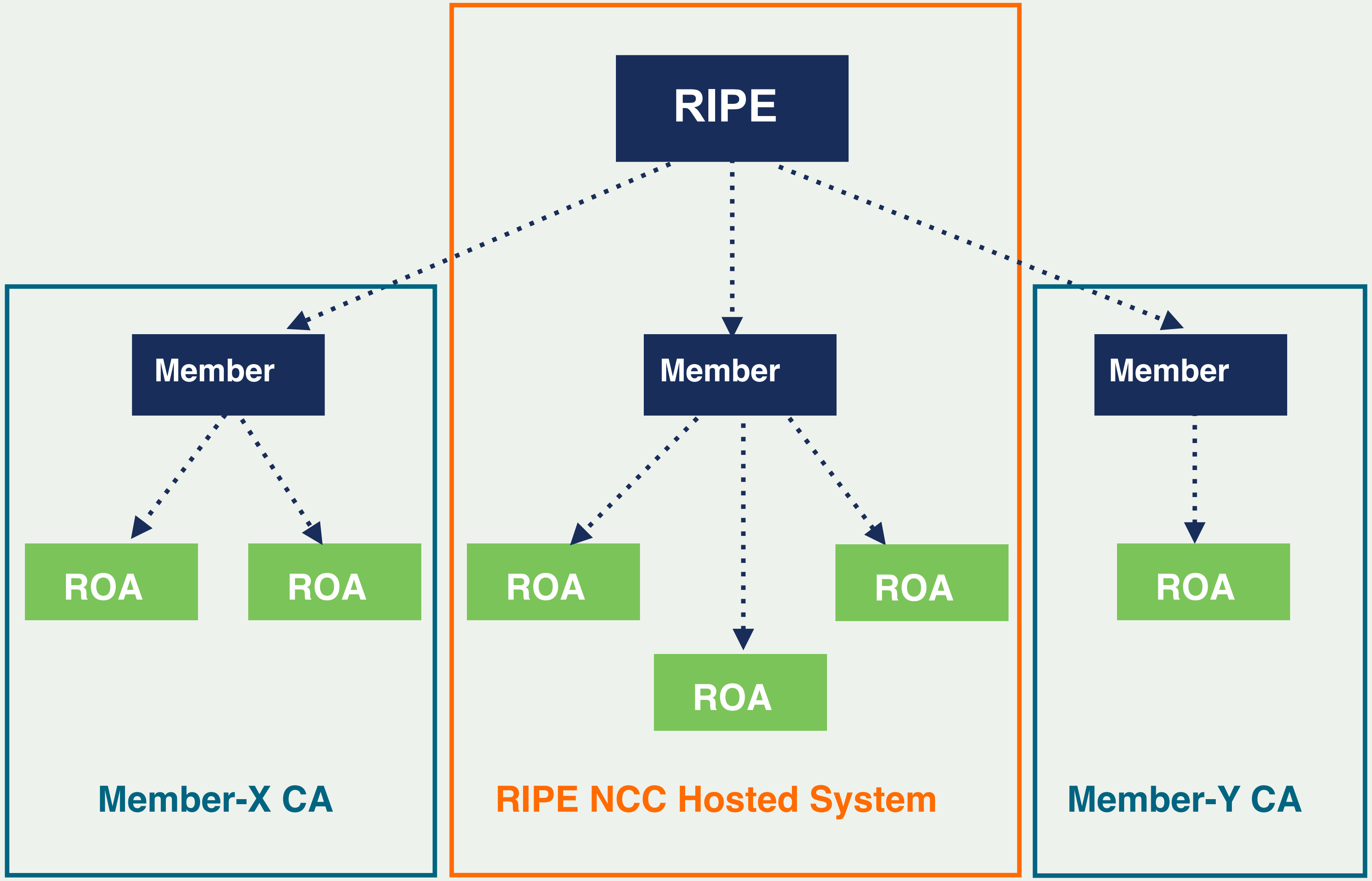


RPKI Certificate Structure

Certificate hierarchy follows allocation hierarchy



Hosted or Delegated RPKI



Hosted RPKI

- RIR hosts a CA and signs all ROAs
- Automate signing and key rollovers
- Allows you focus on creating and publishing ROAs

Delegated RPKI

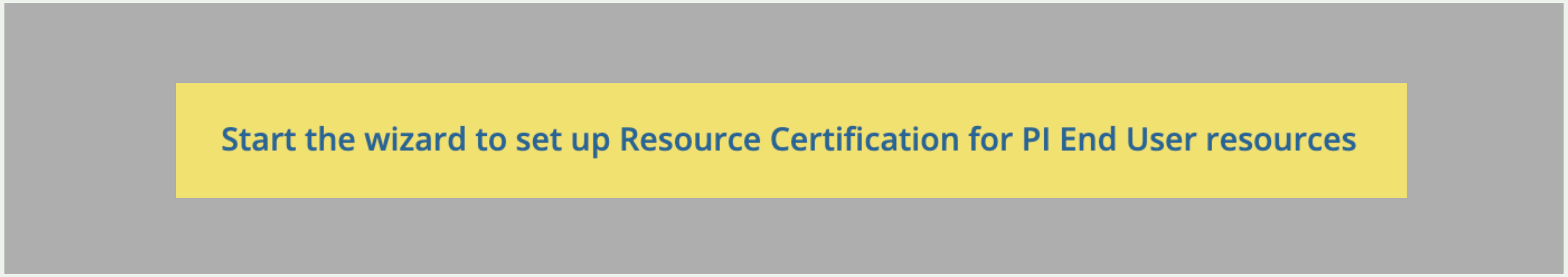
- Run your own Certificate Authority software
  - Dragon Research Labs, RPKI Toolkit
  - NLnetlabs, Krill

- Setup connection with RIR CA
- Generate your LIR certificate and get it signed by parent CA

Certifying PI Resources

Requested and managed by PI End User or by Sponsoring LIR

1. Complete the wizard successfully


Start the wizard to set up Resource Certification for PI End User resources

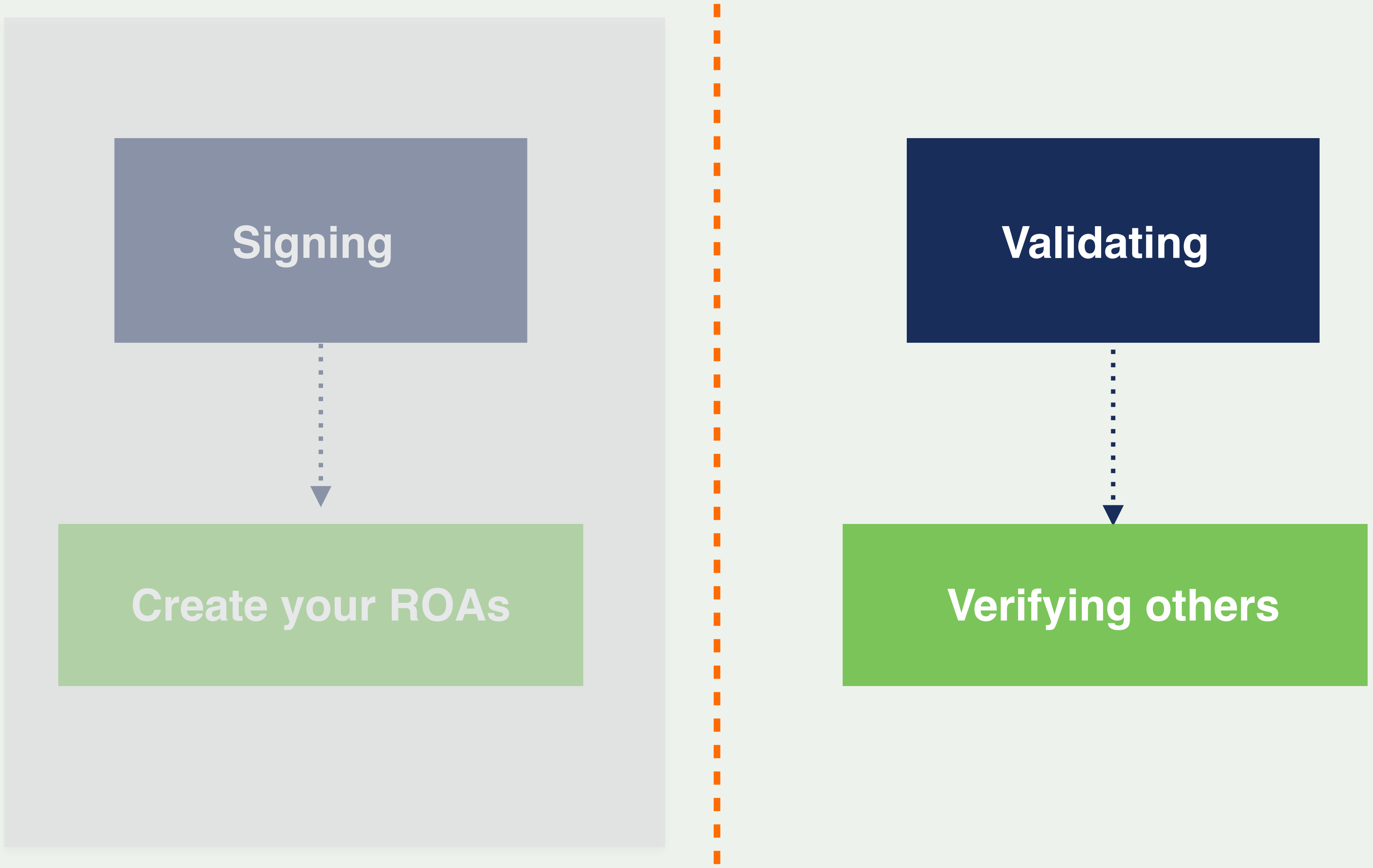
2. Login to https://my.ripe.net and request a certificate
   - Sign in with your RIPE NCC Access account
3. Manage your ROAs

Creating ROAs

Demo on the RIPE NCC LIR Portal

RPKI Validators

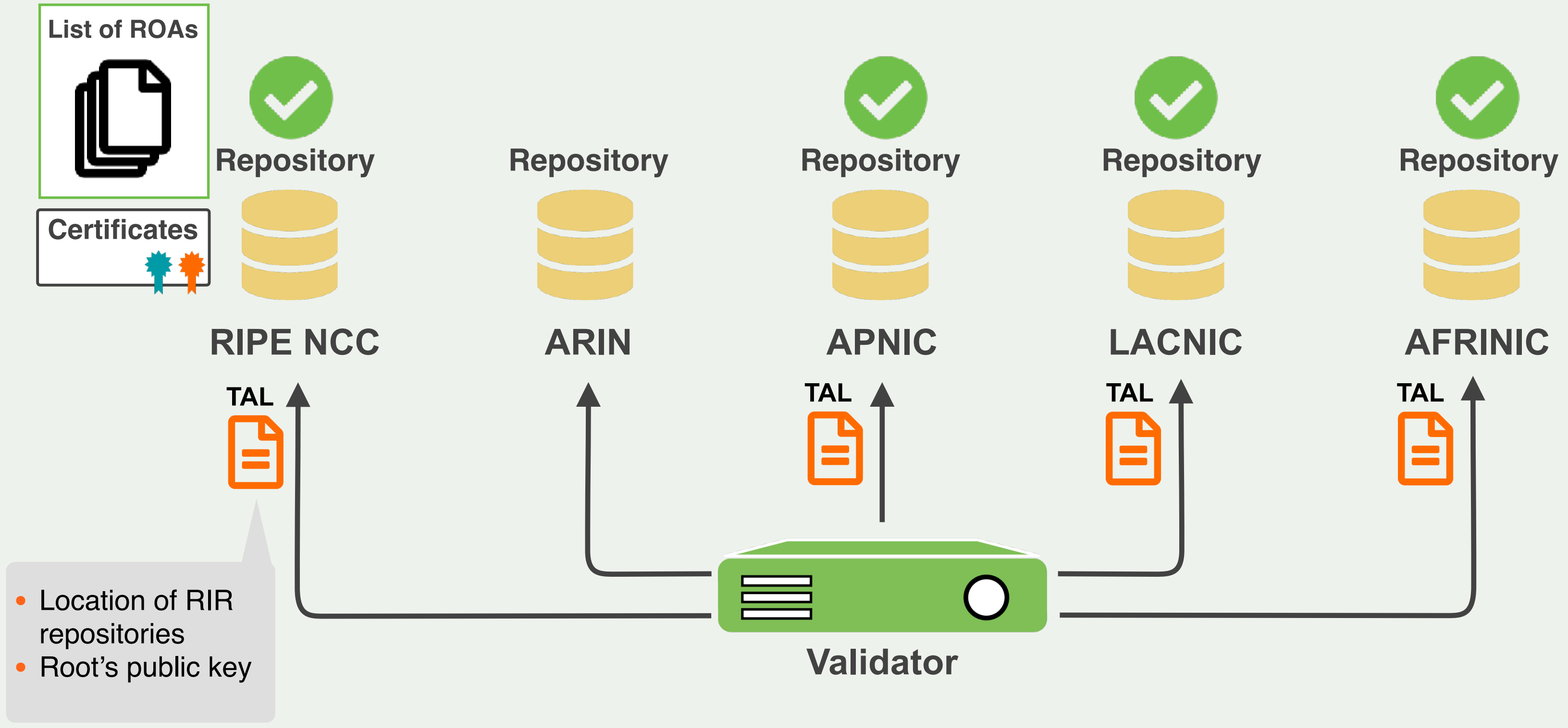
Elements of RPKI



RPKI Validators

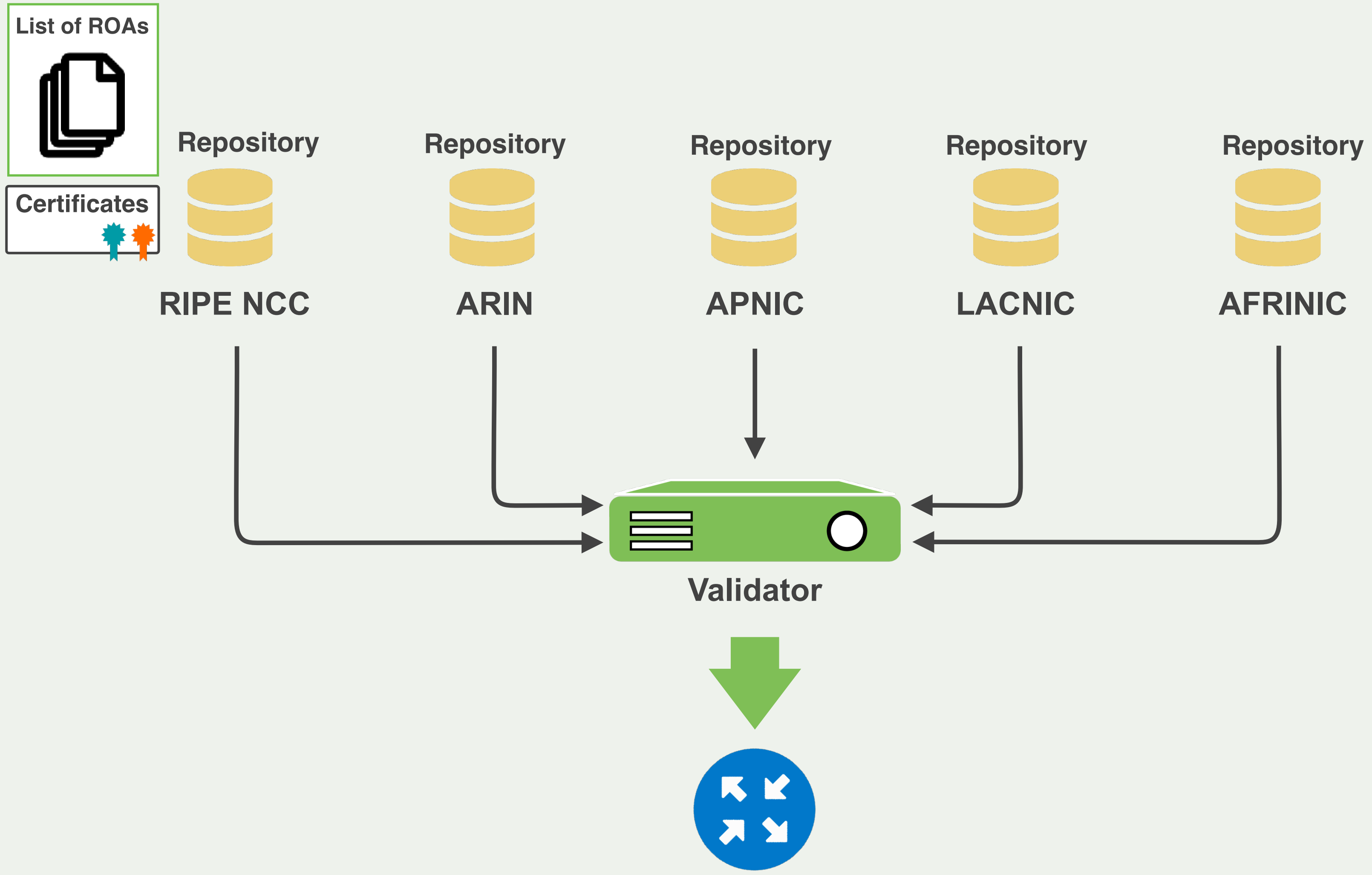
- Software that creates a local **“validated cache”** with all the **valid ROAs**
    - Downloads the RPKI repository from the RIRs
    - Validates the chain of trust of all the ROAs and associated CAs
    - Talks to routers using the RPKI-RTR Protocol

Trust Anchor Locator (TAL)

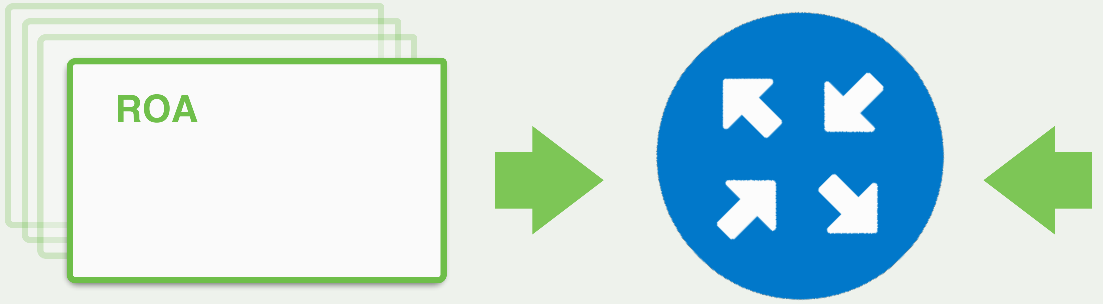


- Location of RIR repositories
- Root's public key

Relying Party

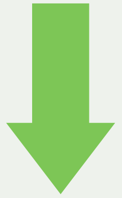


Relying Party



BGP Announcements

AS111	10.0.8.0/22
AS222	10.0.6.0/24
AS333	10.4.16.0/20
AS111	10.0.12.0/22
AS111	10.0.16.0/22
AS111	10.0.20.0/22



BETTER ROUTING DECISIONS

RPKI Validator Options

- **Routinator**
  - Built with Rust, built by NLNetlabs
- **rpki-client**
  - Part of OpenBSD project, written in C
- **OctoRPKI**
  - Cloudflare’s Relying Party software, written in Go
- **FORT**
  - Open source RPKI validator, Written in C

Configuring and Running Validators

Demo

How to Configure Validators

Run at least two validators

- Routinator
- FORT

Configure the correct TALs

- They have already been downloaded
- ARIN TAL needs to be “acknowledged” separately

Start Routinator

On the Server:

```
routinator server --rtr 100.64.1.1:3323
```

➡ TAL directory is **missing!**

➡ We need to initialize via **init command!**

```
[root@server1 ~]# routinator server --rtr 100.64.1.1:3323
Missing TAL directory /root/.rpki-cache/tals.
You may have to initialize it via 'routinator init'.
```

```
[root@server1 ~]# routinator init
Before we can install the ARIN TAL, you must have read
and agree to the ARIN Relying Party Agreement (RPA).
It is available at



If you agree to the RPA, please run the command
again with the --accept-arin-rpa option.
```

```
[root@server1 ~]# routinator init --accept-arin-rpa
Created local repository directory /root/.rpki-cache/repository
Installed 5 TALs in /root/.rpki-cache/tals
```

Start Routinator

On the Server:

```
routinator server --rtr 100.64.1.1:3323
```

Check if it’s running

```
ps aux | grep routinator
```

Start FORT validator

On the Server:

```
systemctl start fort
```

Check if it is running and the logs (exit with ctrl-c):

```
Systemctl status fort

journalctl –u fort
```

Start FORT validator

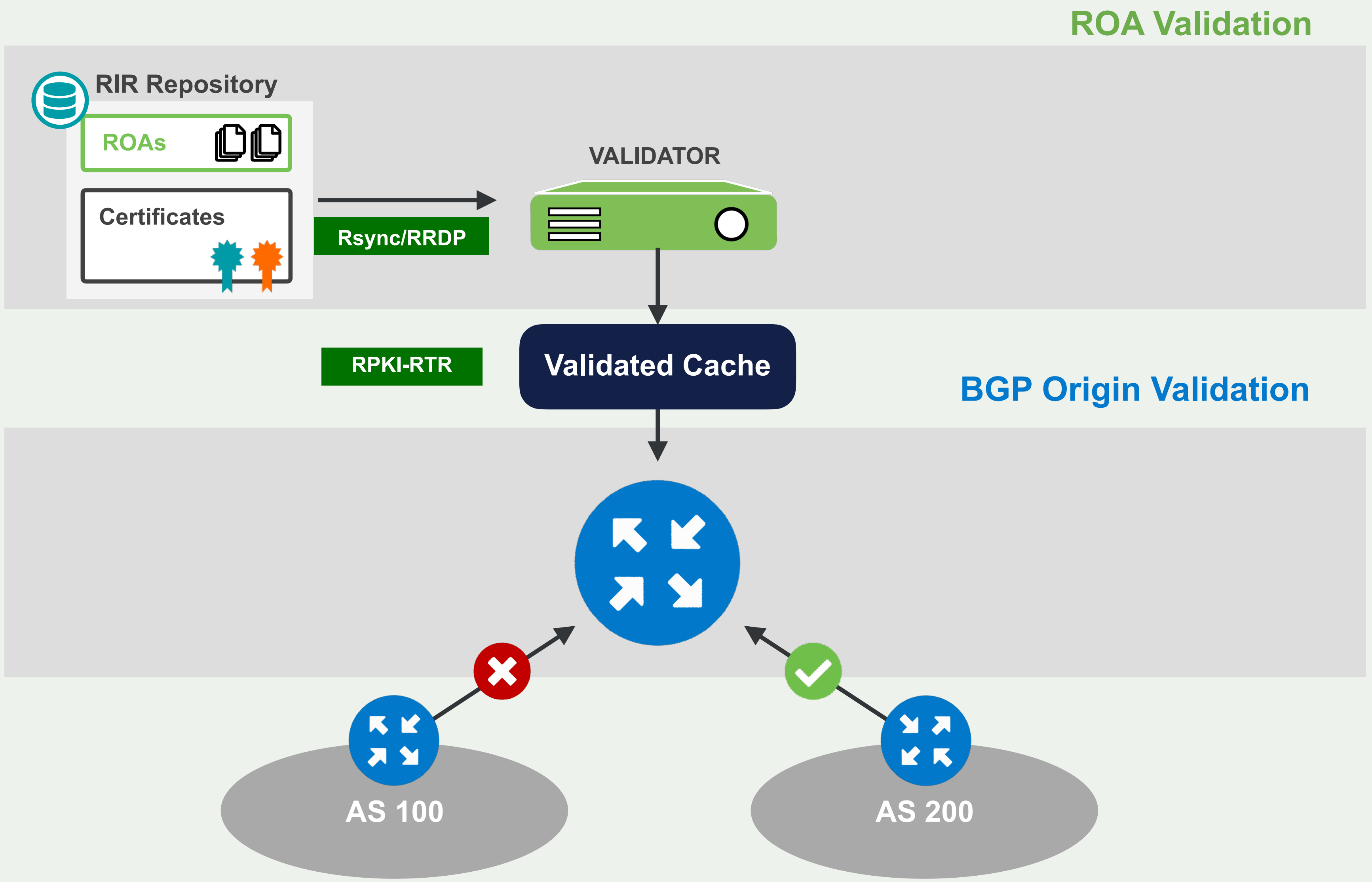
On the Server:

```
fort --init-tals -tal=/etc/fort/tal
```

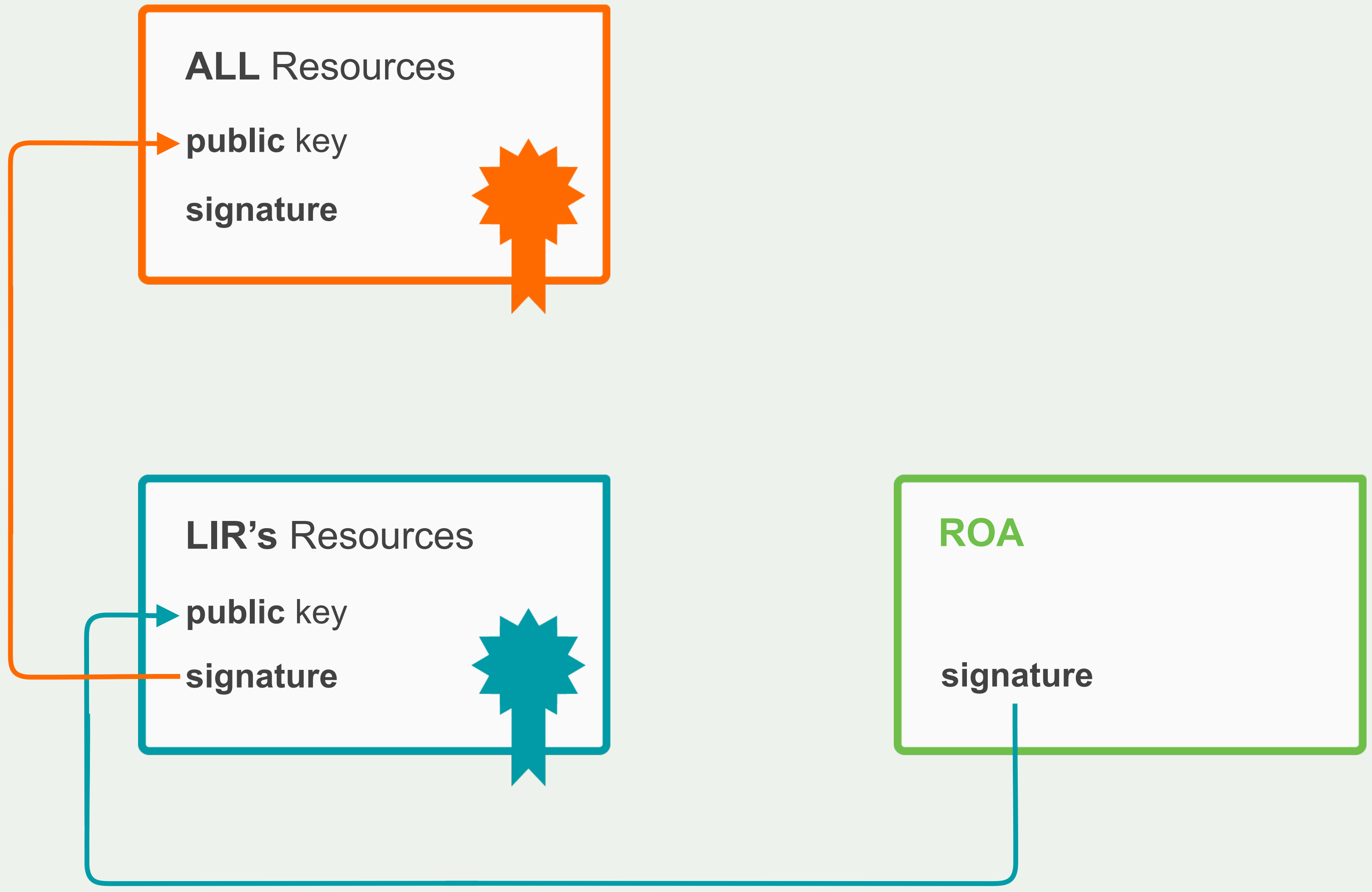
```
[root@server1 ~]# fort --init-tals --tal=/etc/fort/tal
Please download and read ARIN Relying Party Agreement (RPA) from https://
www.arin.net/resources/manage/rpki/rpa.pdf. Once you've read it and if you
agree ARIN RPA, type 'yes' to proceed with ARIN's TAL download:
yes
Successfully fetched '/etc/fort/tal/arin.tal'!
Successfully fetched '/etc/fort/tal/apnic.tal'!
Successfully fetched '/etc/fort/tal/afrinic.tal'!
Successfully fetched '/etc/fort/tal/ripe.tal'!
Successfully fetched '/etc/fort/tal/lacnic.tal'!
```

Validation (ROV)

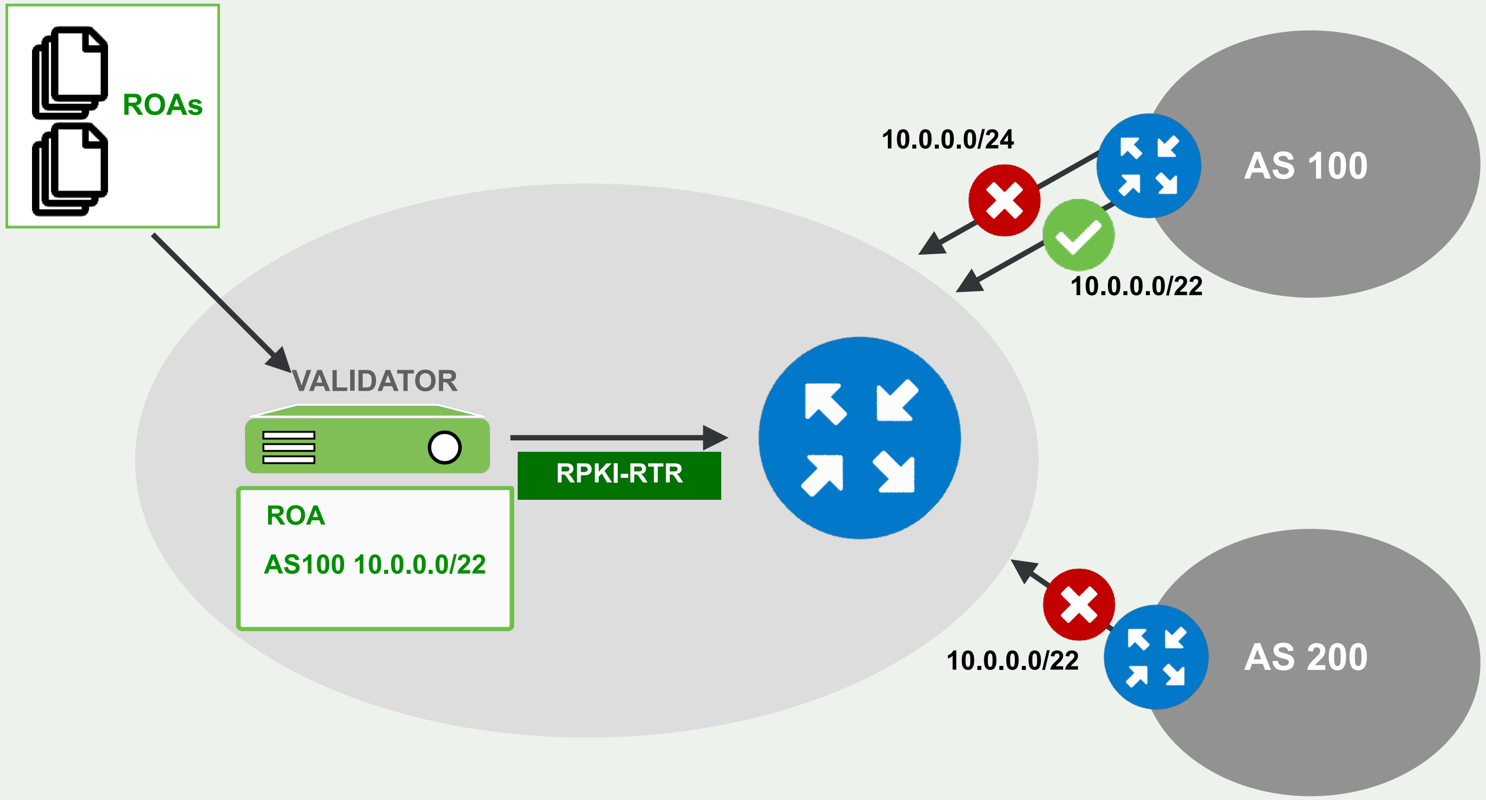
Validation


ROA Validation
RIR Repository
ROAs
Certificates
Rsync/RRDP
VALIDATOR
RPKI-RTR
Validated Cache
BGP Origin Validation
AS 100
AS 200

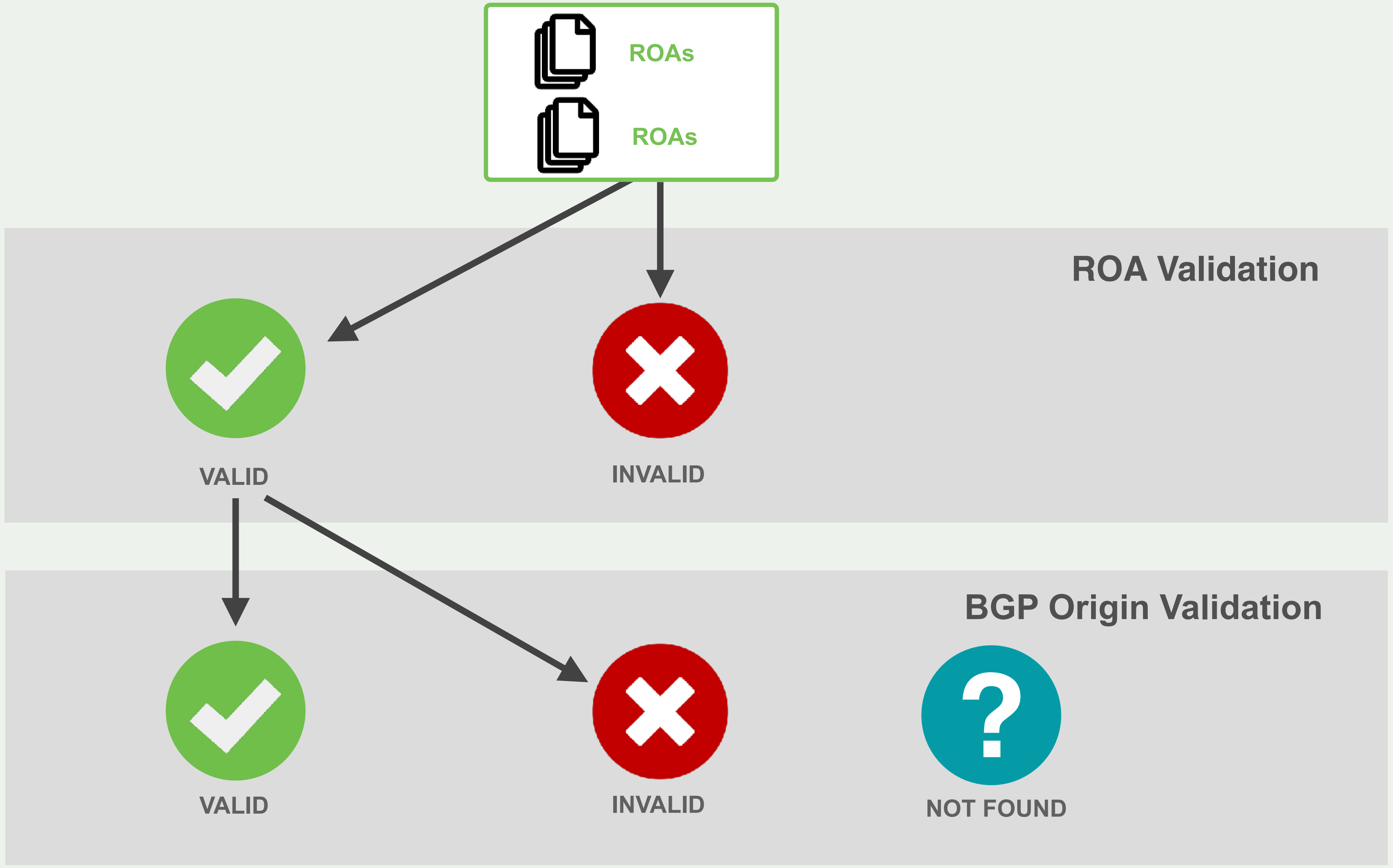
ROA Validation



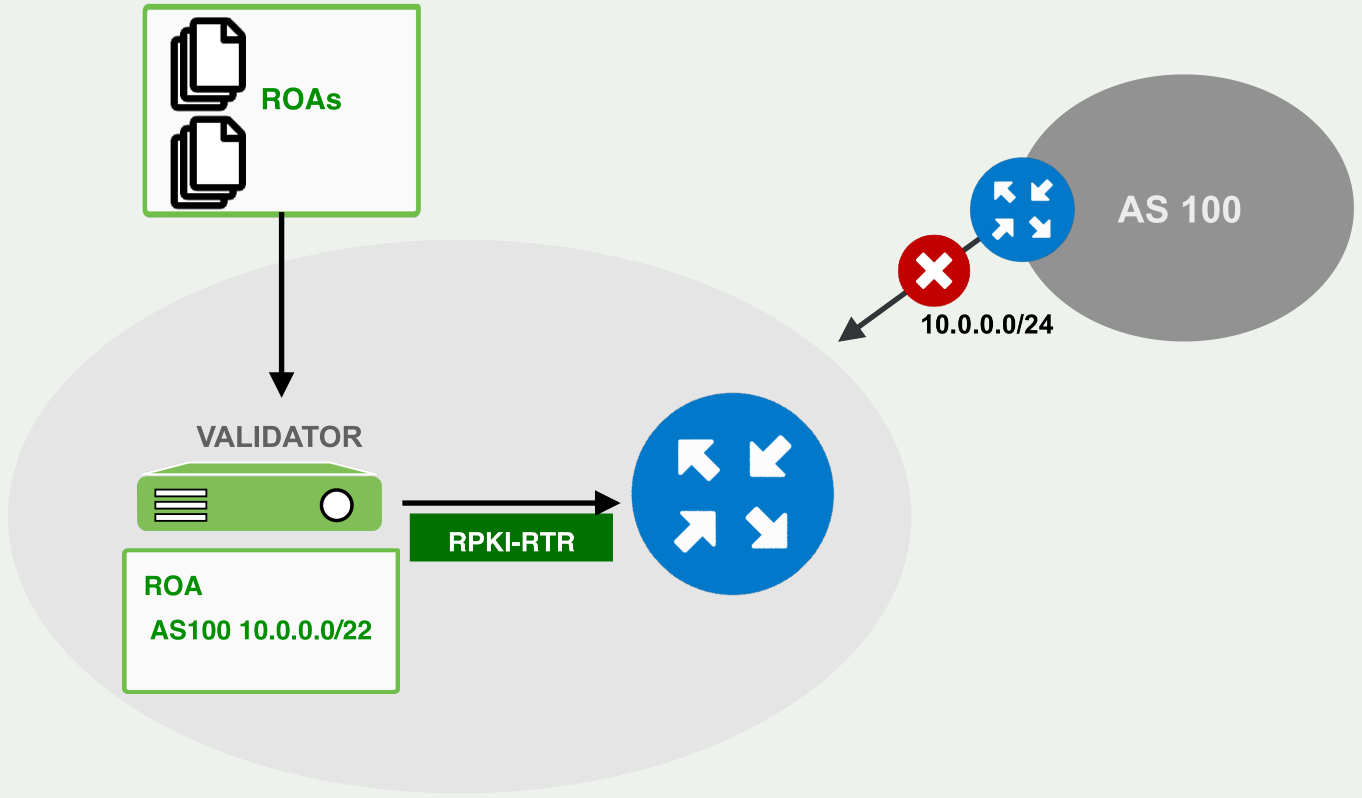
BGP Prefix Origin Validation-RFC6811



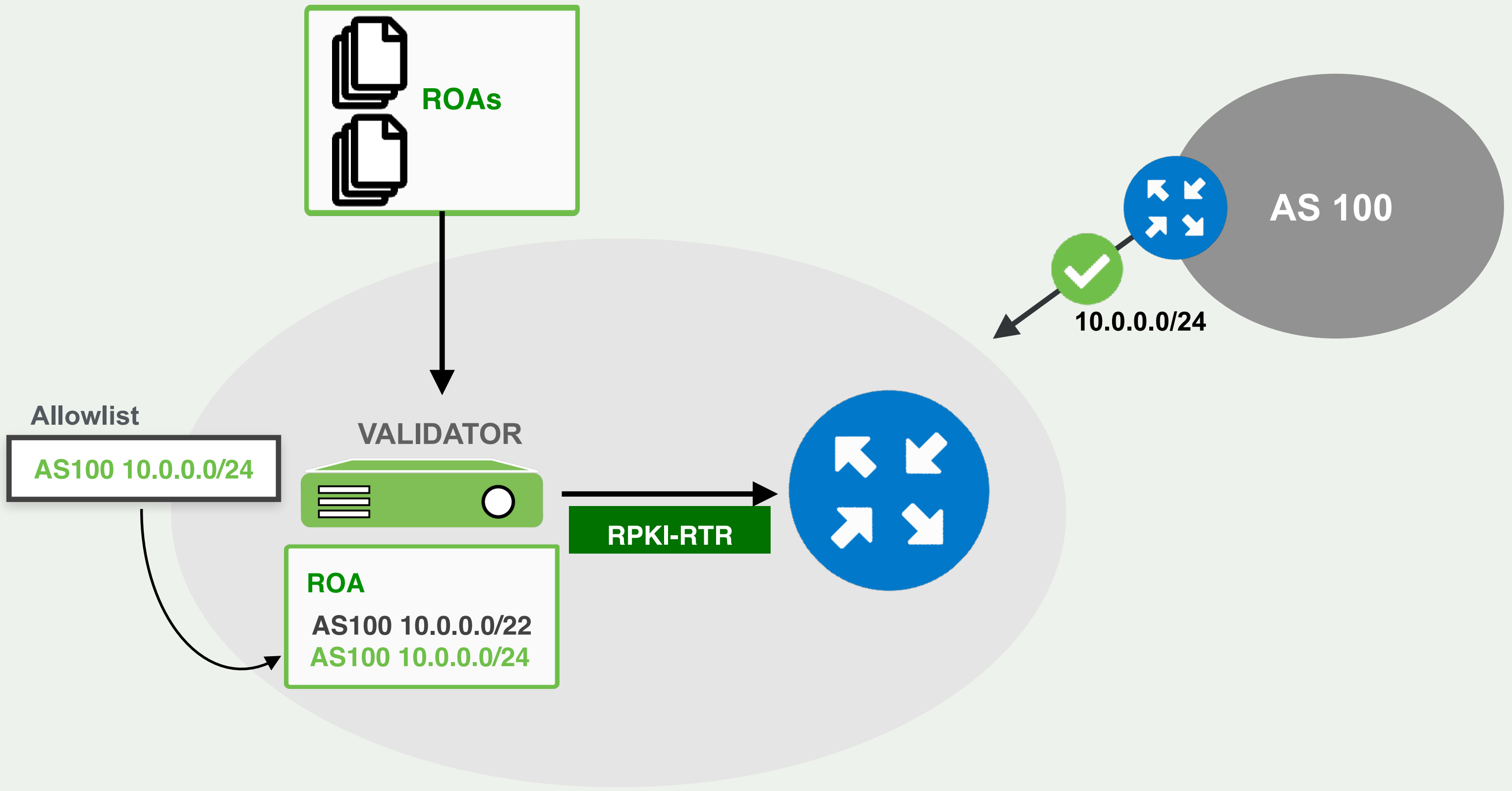
RPKI Validation States



Local Changes to ROA DB (using SLURM)



Local Changes to ROA DB (using SLURM)



Cisco Origin Validation configuration

```
(config)# conf t
(config)# router bgp $ASN
(config-router)# bgp rpki server tcp 100.64.1.1 port 8323 refresh 300
(config-router)# bgp rpki server tcp 100.64.1.1 port 3323 refresh 300
```

Cisco Origin Validation configuration

```
(config-router)# route-map rpki-accept permit 10
(route-map)# match rpki valid
(route-map)# set local-preference 100
(route-map)# route-map rpki-accept permit 20
(route-map)# match rpki not-found
(route-map)# set local-preference 80
```

Cisco Origin Validation configuration

```
(config)# router bgp $ASN
(config)# address-family ipv4
(config)# neighbor 192.168.1.254 route-map rpki-accept in
(config)# address-family ipv6
(config)# neighbor 2002:eeee:ffff::a route-map rpki-accept in
```

Juniper Origin Validation configuration

```
routing-options {
    autonomous-system 64511;
    validation {
        group rpki-validator {
            session 100.64.1.1 {
                refresh-time 120;
                hold-time 180;
                port 8282;
                local-address 100.64.1.2;
}}}}
```

Juniper Origin Validation configuration

```
policy-statement send-direct {
  from protocol direct;
  then accept;}
policy-statement validation {
  term valid {
    from {
      protocol bgp;
      validation-database valid; }
    then {
      local-preference 110;
      validation-state valid;
      community add origin-validation-state-valid;
      accept;
      }}
```

Juniper Origin Validation configuration

```
term invalid {
  from {
    protocol bgp;
    validation-database invalid;}
  then {
    local-preference 90;
    validation-state invalid;
    community add origin-validation-state-invalid;
    accept;
}}
```

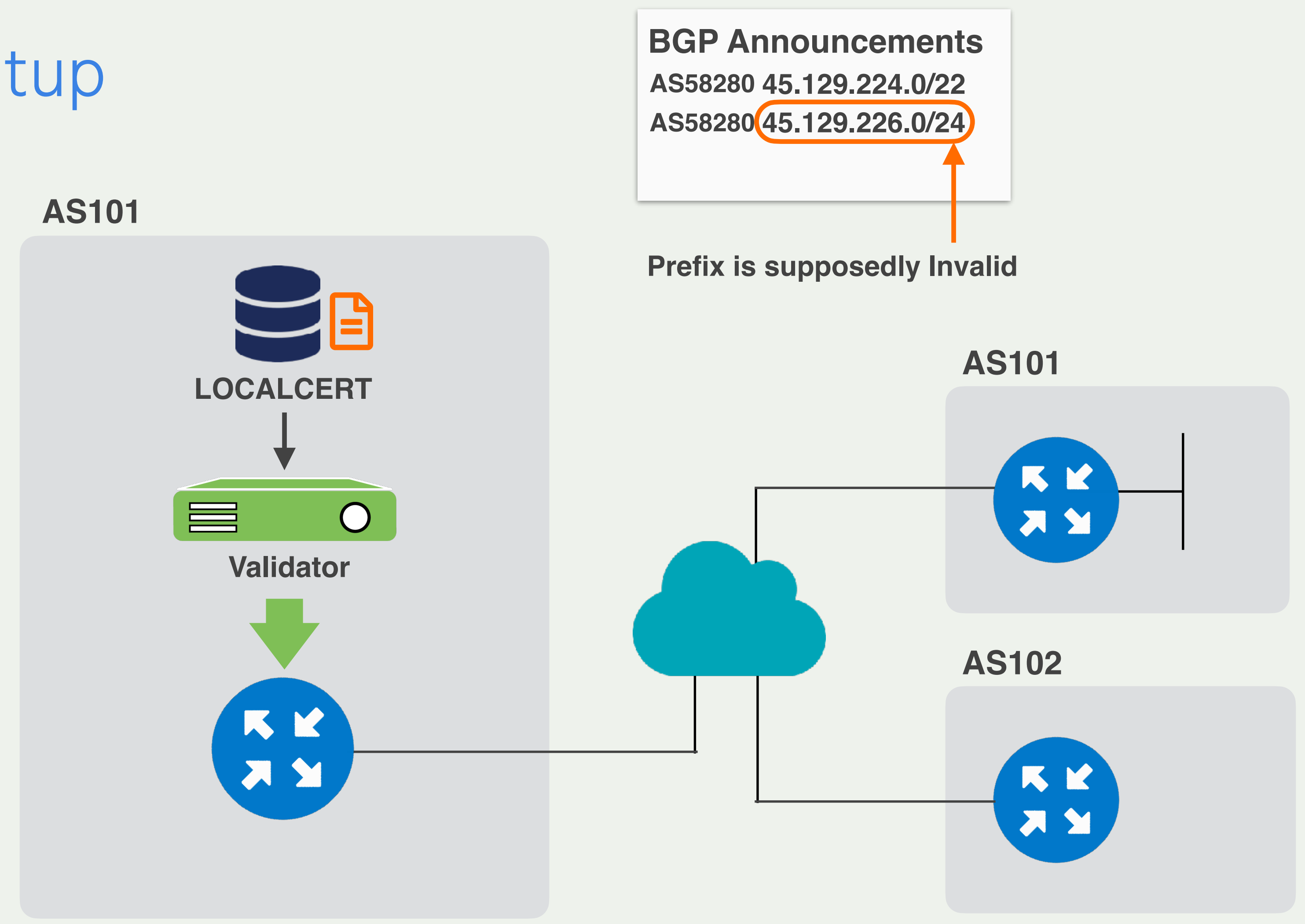
Juniper Origin Validation configuration

```
term **unknown** {
  from protocol bgp;
  then {
  validation-state **unknown**;
  community add origin-validation-state-unknown;
  accept;
}}
```

Route Origin Validation (ROV)

Demo

Demo Setup


BGP Announcements
AS58280 45.129.224.0/22
AS58280 45.129.226.0/24
Prefix is supposedly Invalid
AS101
LOCALCERT
Validator
AS101
AS102

Validators

There are instances of Fort and Routinator running

We now need to configure them on our router

```
rpki
rpki cache 103.162.143.28 3323 preference 1
rpki cache 103.162.143.29 8323 preference 2
```

Validators

And then we can check if it worked

```
Show rpki prefix-table
```

Configuring ROV

First step is to create a route-map

```
route-map rpki permit 20
 match rpki notfound
 set local-preference 100
!
route-map rpki permit 30
 match rpki valid
 set local-preference 120
```

Configuring ROV

And then we apply it to the bgp neighbor

```
router bgp 101
neighbor 123.123.123.0 route-map rpki in
```

… and 45.129.226.0/24 should disappear from our table

Other actions

- You could tag the validated routes with a specific community

  … and the not found ones with another community

Origin Validation Check

- Go with your browser to

http://www.ripe.net/s/rpki-test

- And check if your network applies Origin Validation

Wrapping up

Some suggestions

- Run multiple validators, both in type and location

- Monitor them

  Check that the serial number keeps increasing

BGPAlerter

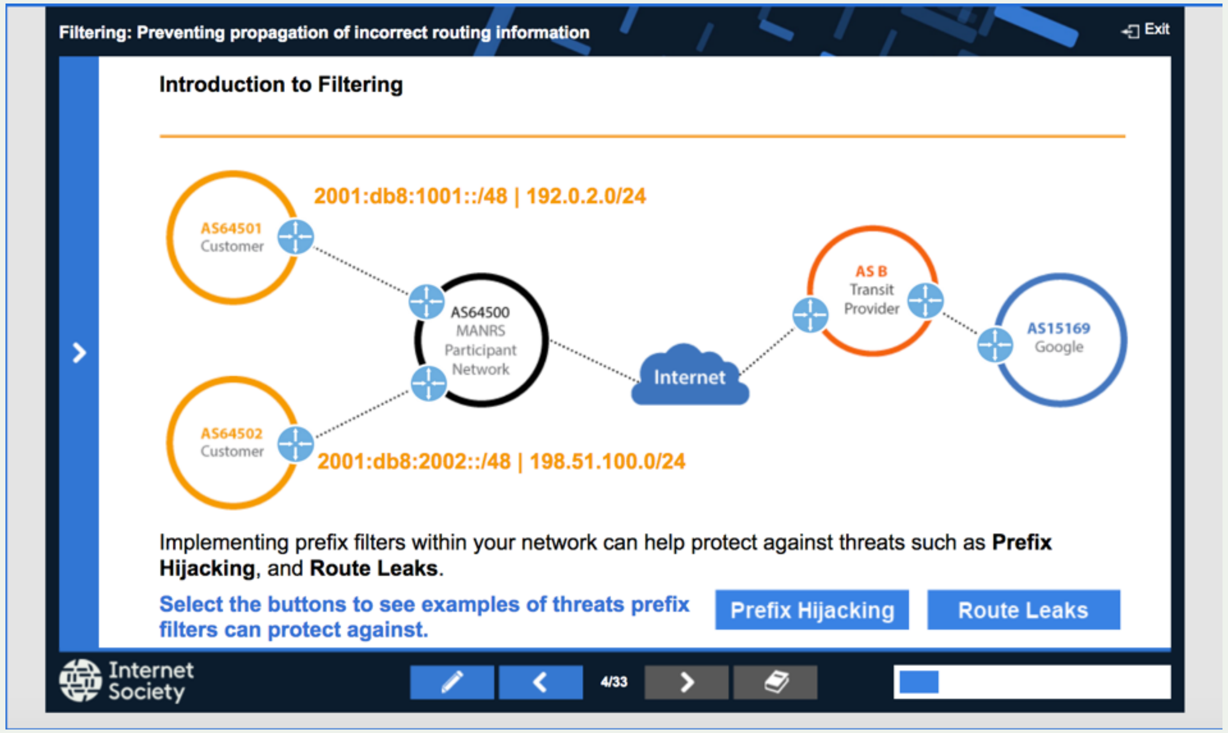
- You can use it to monitor your announcements
- https://github.com/nttgin/BGPalerter
- Very quick setup, monitors changes in your announcements based on RIS data
-

MANRS Training Tutorials

6 training tutorials based on information in the Implementation Guide.

A test at the end of each tutorial.

About to begin training moderators for online classes (43 applications received!)


Filtering: Preventing propagation of incorrect routing information

Introduction to Filtering

Exit
2001:db8:1001::/48 | 192.0.2.0/24
AS64501 Customer
AS64500 MANRS Participant Network
Internet
AS B Transit Provider
AS15169 Google
AS64502 Customer
2001:db8:2002::/48 | 198.51.100.0/24

Implementing prefix filters within your network can help protect against threats such as **Prefix Hijacking**, and **Route Leaks**.

Select the buttons to see examples of threats prefix filters can protect against.

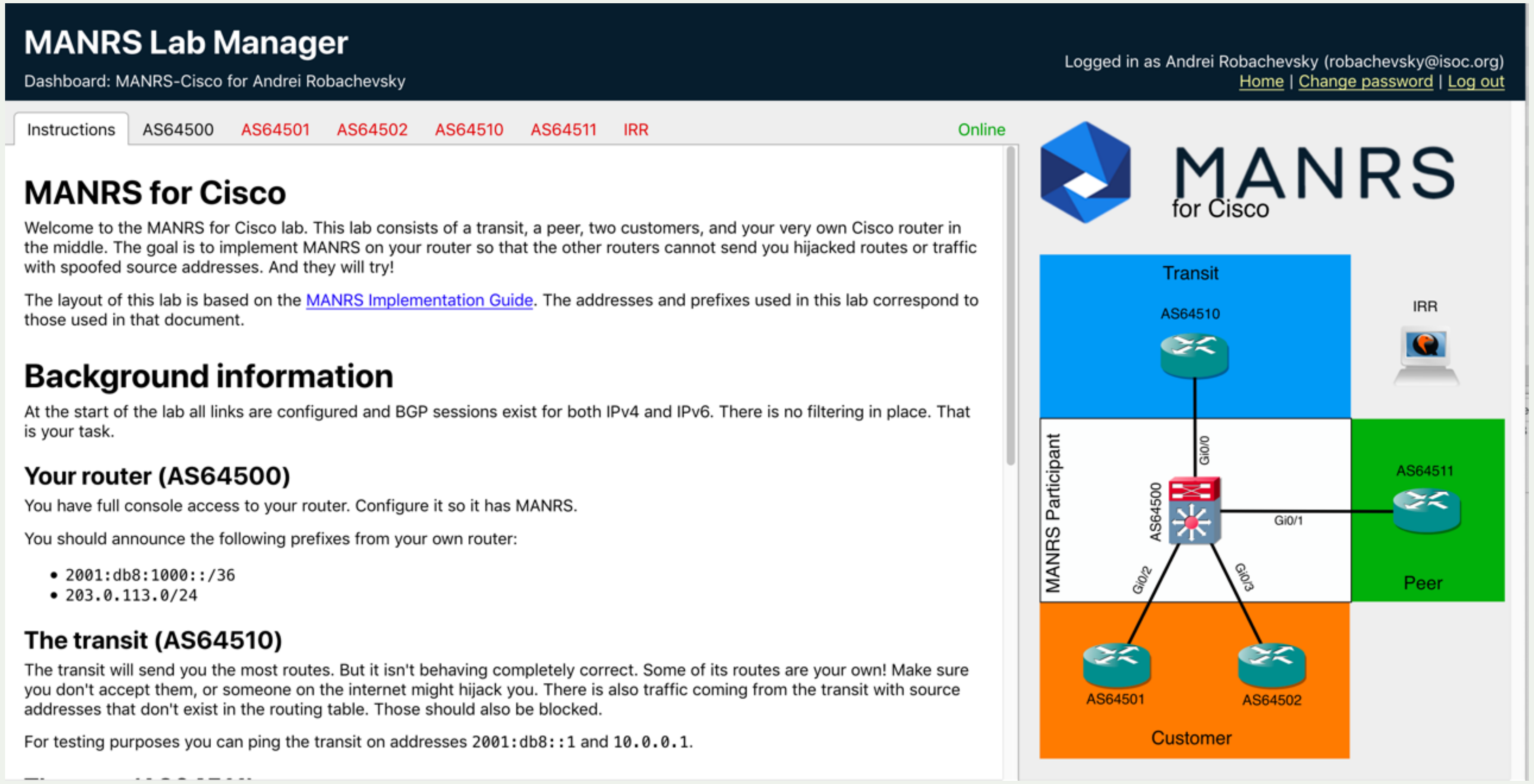
https://www.manrs.org/tutorials

MANRS Hands-on Lab

The prototype lab is ready, finalising the production version.

- Cisco
- Juniper
- Mikrotik

Can be used as a standalone lab or as a final exam



MANRS Lab Manager

Dashboard: MANRS-Cisco for Andrei Robachevsky

Logged in as Andrei Robachevsky (robachevsky@isoc.org)
Home | Change password | Log out

Instructions AS64500 AS64501 AS64502 AS64510 AS64511 IRR Online

MANRS for Cisco

Welcome to the MANRS for Cisco lab. This lab consists of a transit, a peer, two customers, and your very own Cisco router in the middle. The goal is to implement MANRS on your router so that the other routers cannot send you hijacked routes or traffic with spoofed source addresses. And they will try!

The layout of this lab is based on the MANRS Implementation Guide. The addresses and prefixes used in this lab correspond to those used in that document.

Background information

At the start of the lab all links are configured and BGP sessions exist for both IPv4 and IPv6. There is no filtering in place. That is your task.

Your router (AS64500)

You have full console access to your router. Configure it so it has MANRS.

You should announce the following prefixes from your own router:

- `2001:db8:1000::/36`
- `203.0.113.0/24`

The transit (AS64510)

The transit will send you the most routes. But it isn't behaving completely correct. Some of its routes are your own! Make sure you don't accept them, or someone on the internet might hijack you. There is also traffic coming from the transit with source addresses that don't exist in the routing table. Those should also be blocked.

For testing purposes you can ping the transit on addresses `2001:db8::1` and `10.0.0.1`.

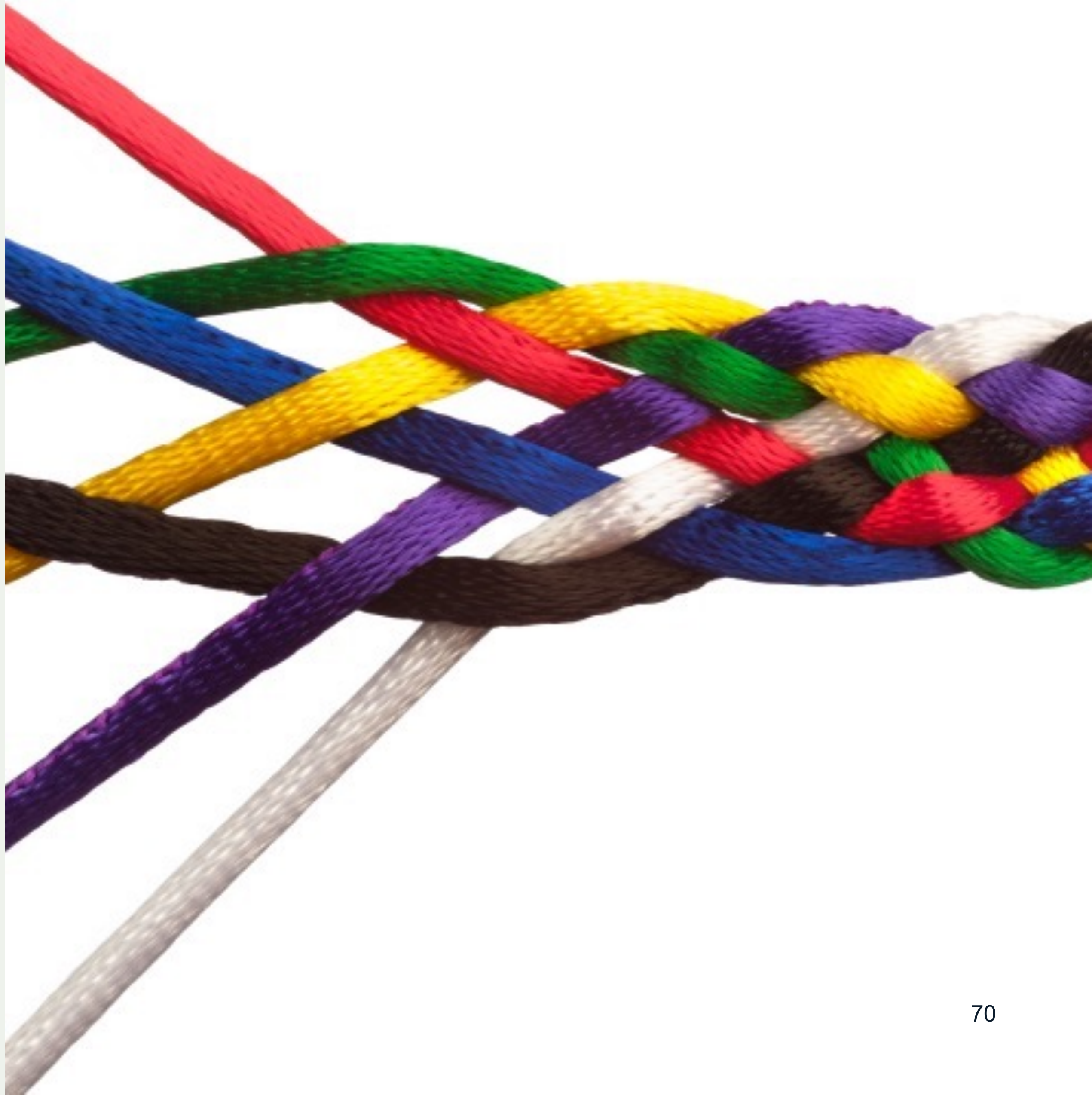
Join MANRS

Visit https://www.manrs.org

- Fill out the sign up form with as much detail as possible.
- We may ask questions and request tests

Get Involved in the Community

- Participants support the initiative and implement the actions in their own networks and encouraging MANRS adoption
- Participants are engaged in substantive activities – developing MANRS requirements and guidance, assisting with capacity and awareness building activities



Questions ?

stucchi@isoc.org

@stucchimax