RIPE NCC
RIPE NETWORK COORDINATION CENTRE

# Secure Internet Routing

Cigdem Senol | 22 June 2023 | **nog.fi**

# BGP has some challenges …

- It is only based on **trust**, no built-in security
- **No verification** of the correctness of prefixes or AS paths

**RFC 4272 -** “BGP Security Vulnerabilities Analysis”

# Due to these vulnerabilities …

- Any AS can announce any prefix
- Any AS can prepend any ASN to the AS path
- BGP announcements are accepted without validation
- Fake routing information may disrupt Internet routing!

# For secure Internet Routing …

- Do not be the cause!
  - Announce the right prefixes to the right peers
- Do not distribute others’ mistakes or attacks!
  - Validate the routing information you receive
- Do not be the victim!
  - Take all the measures you can to protect your network

# Have proper filters in place!

- Inbound filters
  - Detects configuration mistakes and attacks
  - Particularly from customer networks
- Outbound filters
  - Eliminates route leaks
- Filter routes with prefix or AS path filters
  - Manually or automatically with data from IRRs

# Validate received routes!

*Is the AS authorised to **originate** a certain IP prefix?*

- The **IRR** system is in place to make informed routing decisions
  - Many transit providers and IXPs perform IRR filtering
  - Automation relies on the IRR being complete
- **RPKI** aims to complement and expand this effort
  - Validates the routes based on trusted, accurate and up-to-date RPKI data

# Validate received routes!

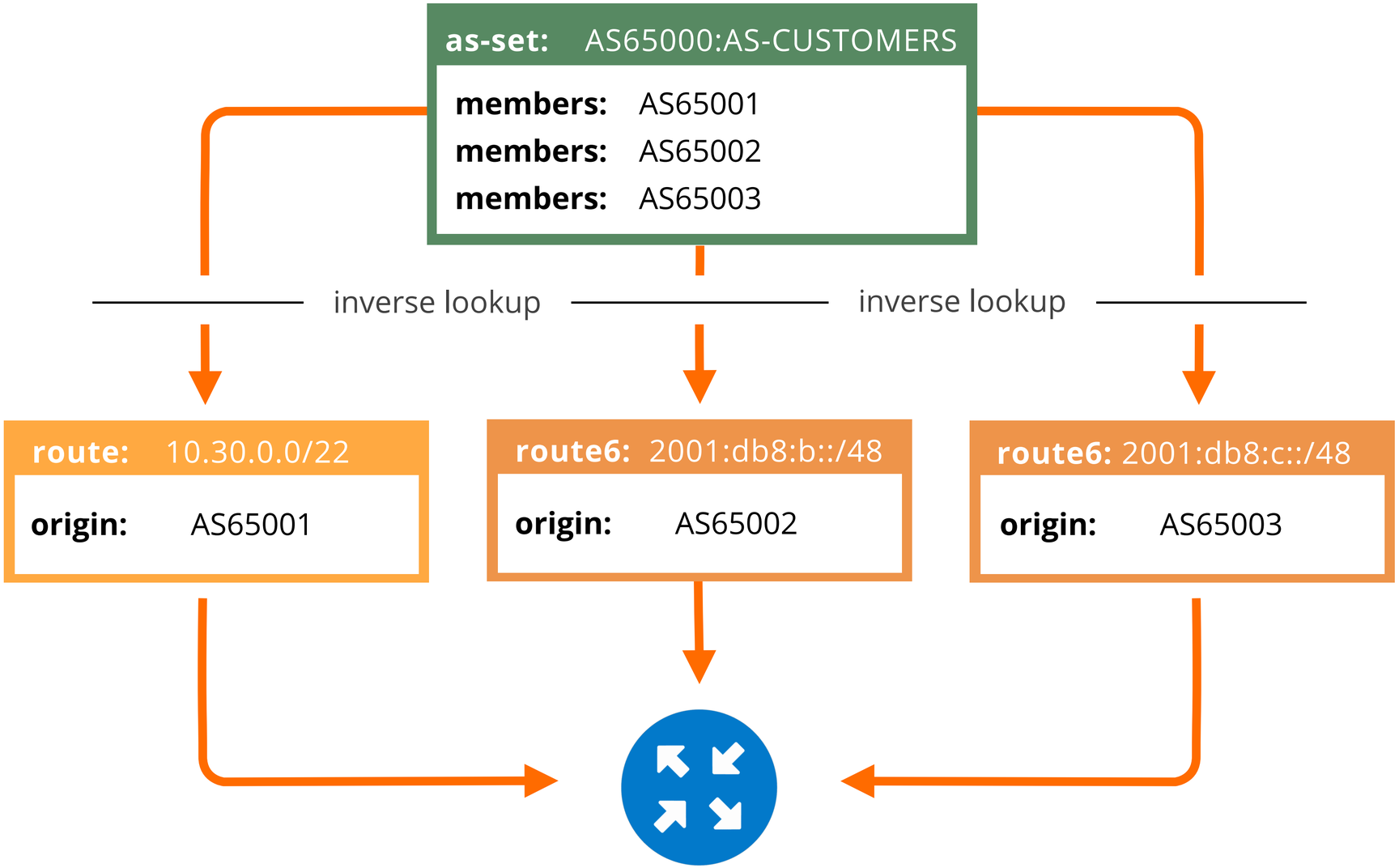
*Are BGP path attributes legitimate and correct?*

- Requires validation of whole BGP path
- RPKI is a stepping stone to path validation!
- **BGPsec** (RFC 8205)
- **ASPA** (Autonomous System Provider Authorisation) (draft)

# Internet Routing Registries (IRRs)

- Public routing policy databases
  - Declarations of BGP announcements, connected peers and routing policies
- Many IRR databases exist, mostly mirroring each other
  - RIPE, APNIC, RADB, JPIRR, Level3, NTTCom, others
- Tools available that get the policy data from IRRs
  - IRRToolset , IRRPT, bgpq4

# Generating prefix filters from IRRs



# DEMO: Generating BGP filter with bgpq4

Cisco by default

```
$ bgpq4 -l AS3333-v4-policy AS3333
no ip prefix-list AS3333-v4-policy
ip prefix-list AS3333-v4-policy permit 193.0.0.0/21
ip prefix-list AS3333-v4-policy permit 193.0.10.0/23
ip prefix-list AS3333-v4-policy permit 193.0.12.0/23
ip prefix-list AS3333-v4-policy permit 193.0.18.0/23
ip prefix-list AS3333-v4-policy permit 193.0.20.0/23
ip prefix-list AS3333-v4-policy permit 193.0.22.0/23
ip prefix-list AS3333-v4-policy permit 193.230.194.0/24
```

```
$ bgpq4 -6 -l AS3333-v6-policy AS3333
no ipv6 prefix-list AS3333-v4-policy
ipv6 prefix-list AS3333-v4-policy permit 2001:610:240::/42
ipv6 prefix-list AS3333-v4-policy permit 2001:67c:2e8::/48
ipv6 prefix-list AS3333-v4-policy permit 2a13:27c0::/29
ipv6 prefix-list AS3333-v4-policy permit 2a13:27c0:10::/44
```

# DEMO: Generating BGP filter with bgpq4

```
$ bgpq4 -6 -Jl AS3333-v6-policy AS3333
policy-options {
replace:
 prefix-list AS3333-v6-policy {
     2001:610:240::/42;
     2001:67c:2e8::/48;
     2a13:27c0::/29;
     2a13:27c0:10::/44;
 }
}
```

Juniper Junos

```
$ bgpq4 -6 -Bl AS3333-v6-policy AS3333
AS3333-v6-policy="prefix {
  2001:610:240::/42
  2001:67c:2e8::/48
  2a13:27c0::/29
  2a13:27c0:10::/44
  }"
```

OpenBSD

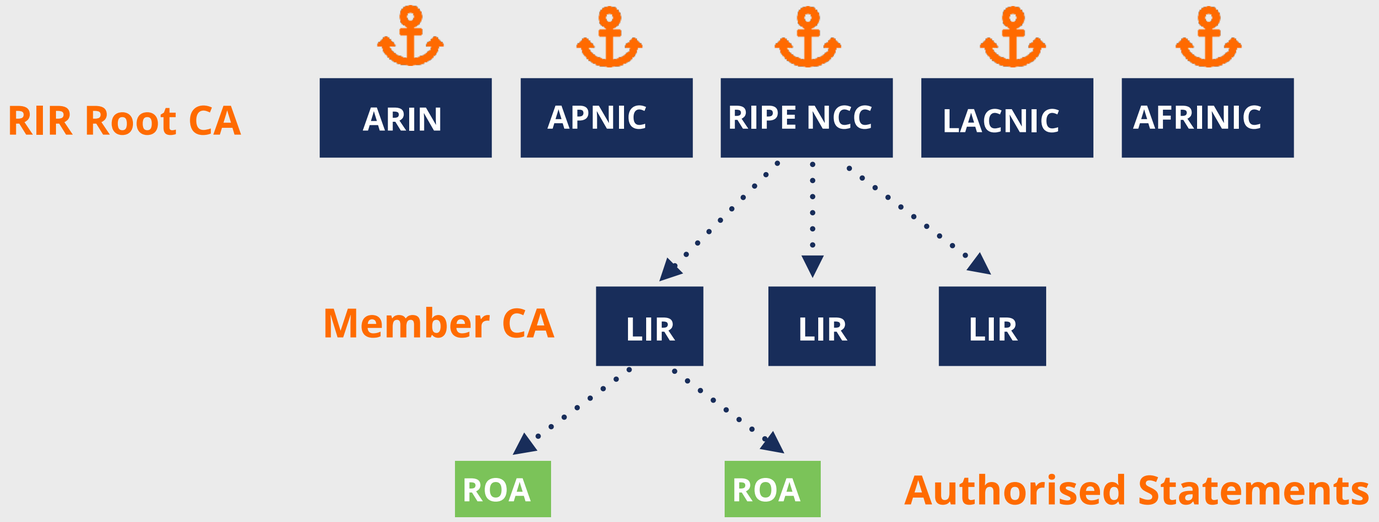
MikroTik

```
$ bgpq4 -6 -Kl AS3333-v6-policy AS3333
/routing filter add action=accept chain="AS3333-v6-policy-V6" prefix=2001:610:240::/42
/routing filter add action=accept chain="AS3333-v6-policy-V6" prefix=2001:67c:2e8::/48
/routing filter add action=accept chain="AS3333-v6-policy-V6" prefix=2a13:27c0::/29
/routing filter add action=accept chain="AS3333-v6-policy-V6" prefix=2a13:27c0:10::/44
```

IRR filters are good **only if the IRR entries are correct!**

# RPKI complements routing security efforts!

- Public key infrastructure for Internet number resources
  - Attaches digital certificate to IP addresses and AS numbers
- Hierarchy with
  - 5 RIR trust anchors
  - 2 AS0 trust anchors from APNIC and LACNIC



# RPKI complements routing security efforts!

- Signed objects with different payloads
  - ROA with VRP (used for ROV)
  - ASPA with VAP (in development)
- Currently only ROAs are of practical use

# How does RPKI enable routing security?

## SIGNING

Create ROAs for your prefixes in the RPKI system

+

## VALIDATION

Verify the information provided by others

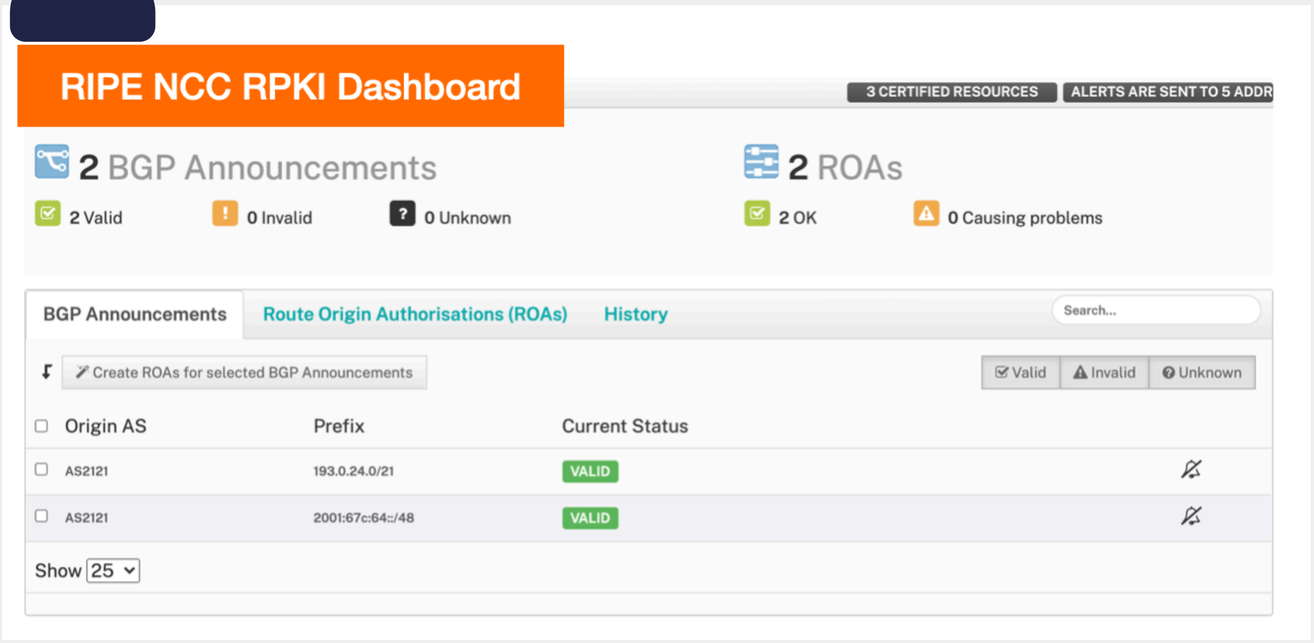
# SIGNING

Create ROAs for your prefixes in the RPKI system

+

# VALIDATION

Verify the information provided by others




RIPE NCC RPKI Repository

Publication

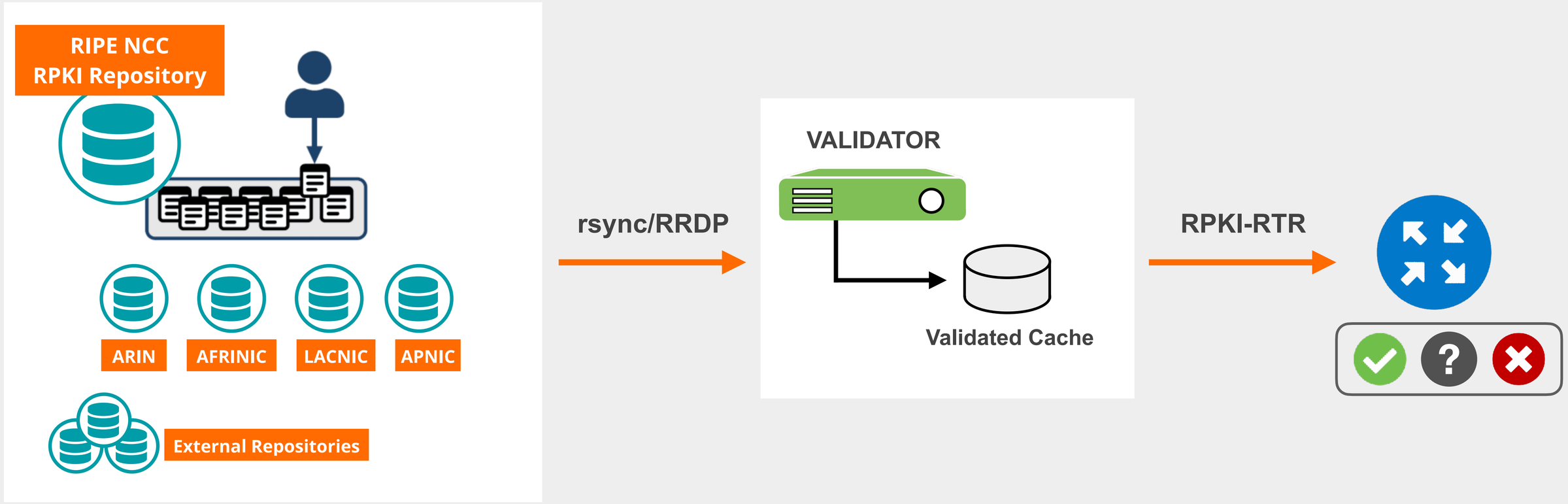
# SIGNING

Create ROAs for your prefixes in the RPKI system

+

# VALIDATION

Verify the information provided by others



# RPKI Validators

- **Routinator**
  - Built by NLNetlabs
- **OctoRPKI**
  - Cloudflare’s relying party software
- **FORT**
  - Open source RPKI validator
- **rpki-client**
  - Integrated in OpenBsd

**Links for RPKI Validators**

https://github.com/NLnetLabs/routinator.git

https://github.com/cloudflare/cfrpki#octorpki

https://github.com/NICMx/FORT-validator/

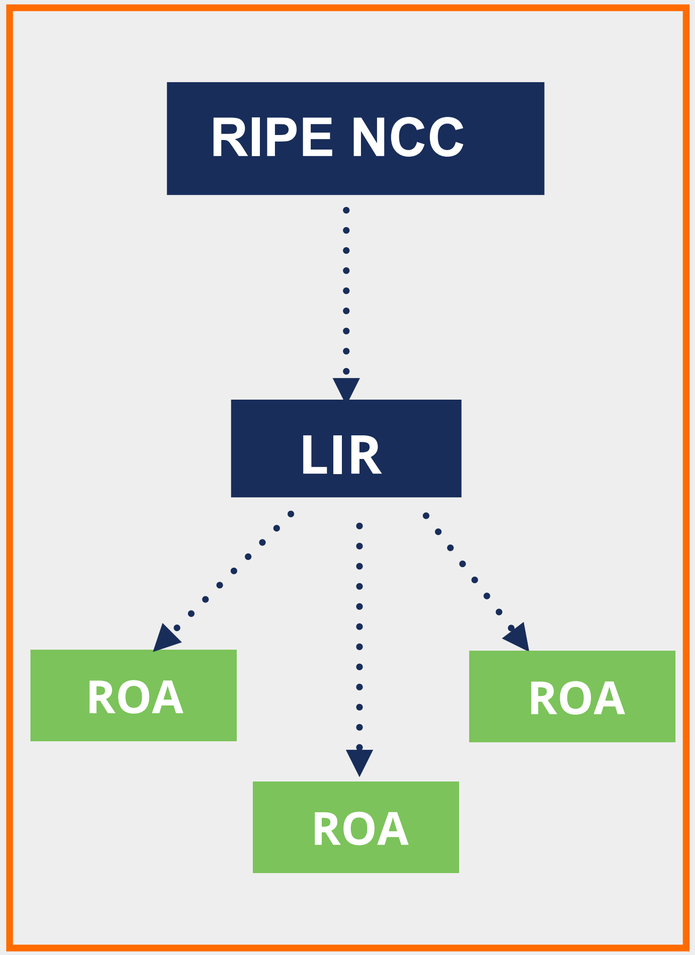
https://www.rpki-client.org/

**For more info…** https://rpki.readthedocs.io

RPKI has two flavours: **Hosted** and **Delegated RPKI**

# Hosted RPKI

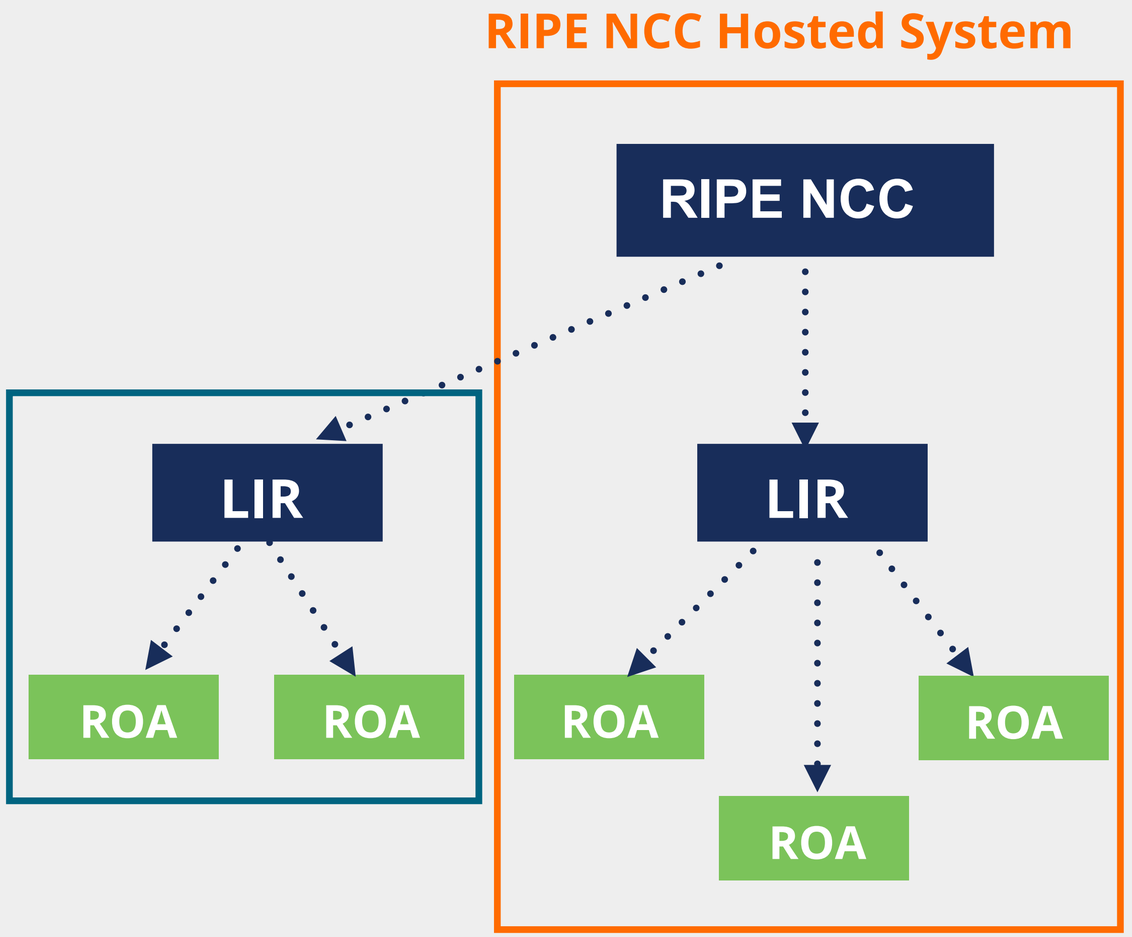
- ROAs are created and published using the RIR’s member portal
- RIR hosts CA and signs all ROAs
- Automated signing and key rollovers
- Allows LIRs focus on creating and publishing ROAs



# Delegated RPKI

- LIR manages full RPKI system
  - Runs its own CA, manages keys/key rollovers
  - Creates, signs and publishes ROAs

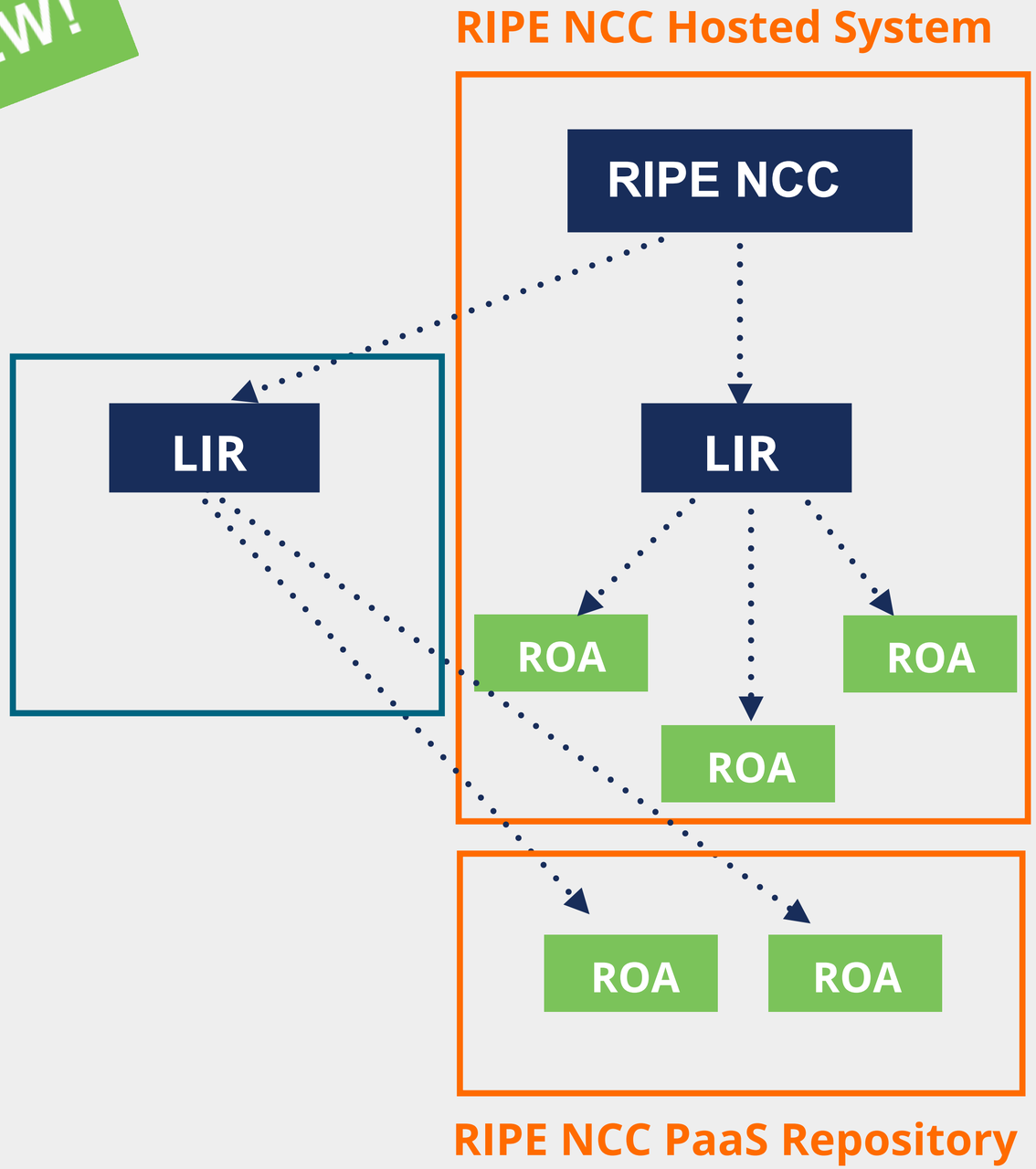
- Certificate Authority (CA) Software
  - **Krill** (NLnet Labs)
  - **rpkid** (Dragon Research Labs)



# Publication as a Service

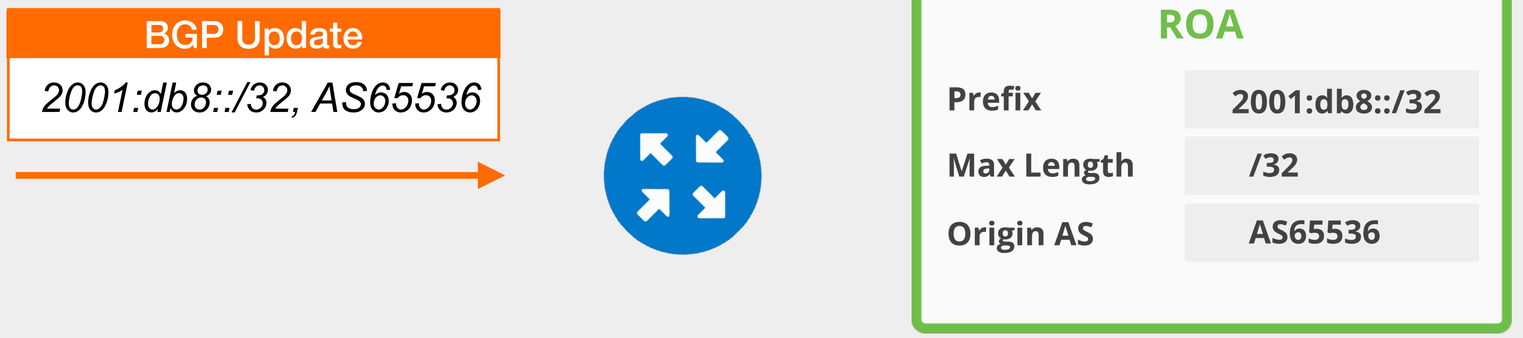
NEW!

- aka “Publication in parent” or “Hybrid RPKI”
- In-between hosted and delegated RPKI
  - LIR maintains key-pairs and ROAs
  - RIR publishes your ROAs in its repository
- Supported by APNIC, ARIN and RIPE NCC



# RPKI & BGP Route Origin Validation (ROV)

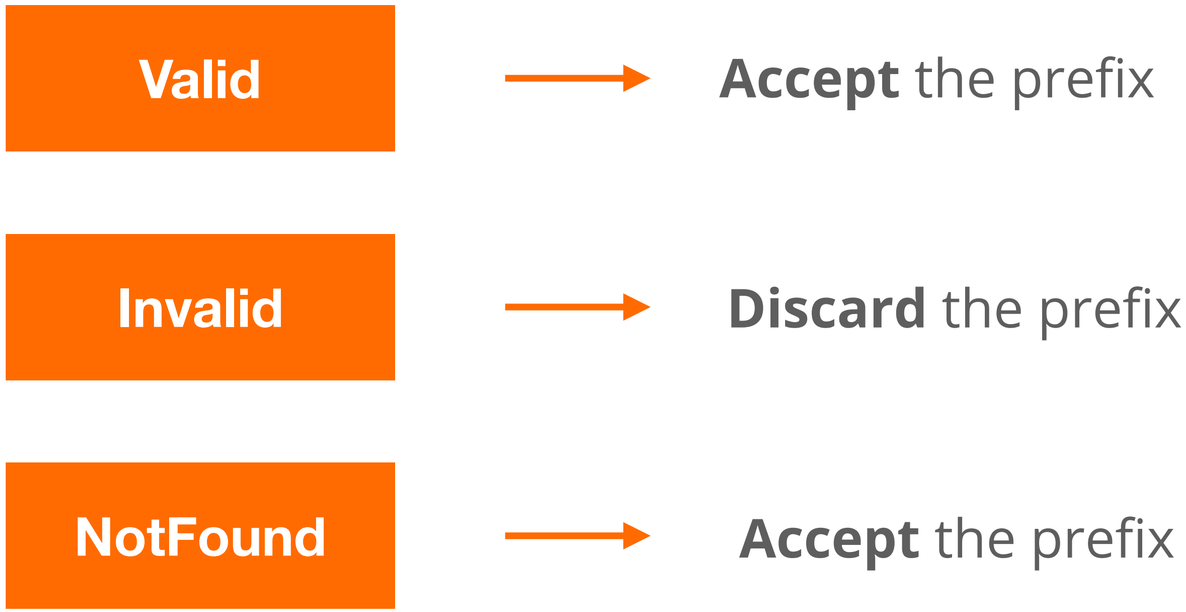
- RPKI based route filtering, RFC 6811
- BGP announcements are compared against the **valid** ROAs
  - **origin ASN** and **max-length** must match!
- Router validates the  origin of received routes: **Valid**, **Invalid** and **Not Found**




**RFC 6811 -** “BGP Prefix Origin Validation”

# After Validating …

- You have to make a decision : “Accept” or “Discard”



# After Validating …

- You have to make a decision : “Accept” or “Discard”

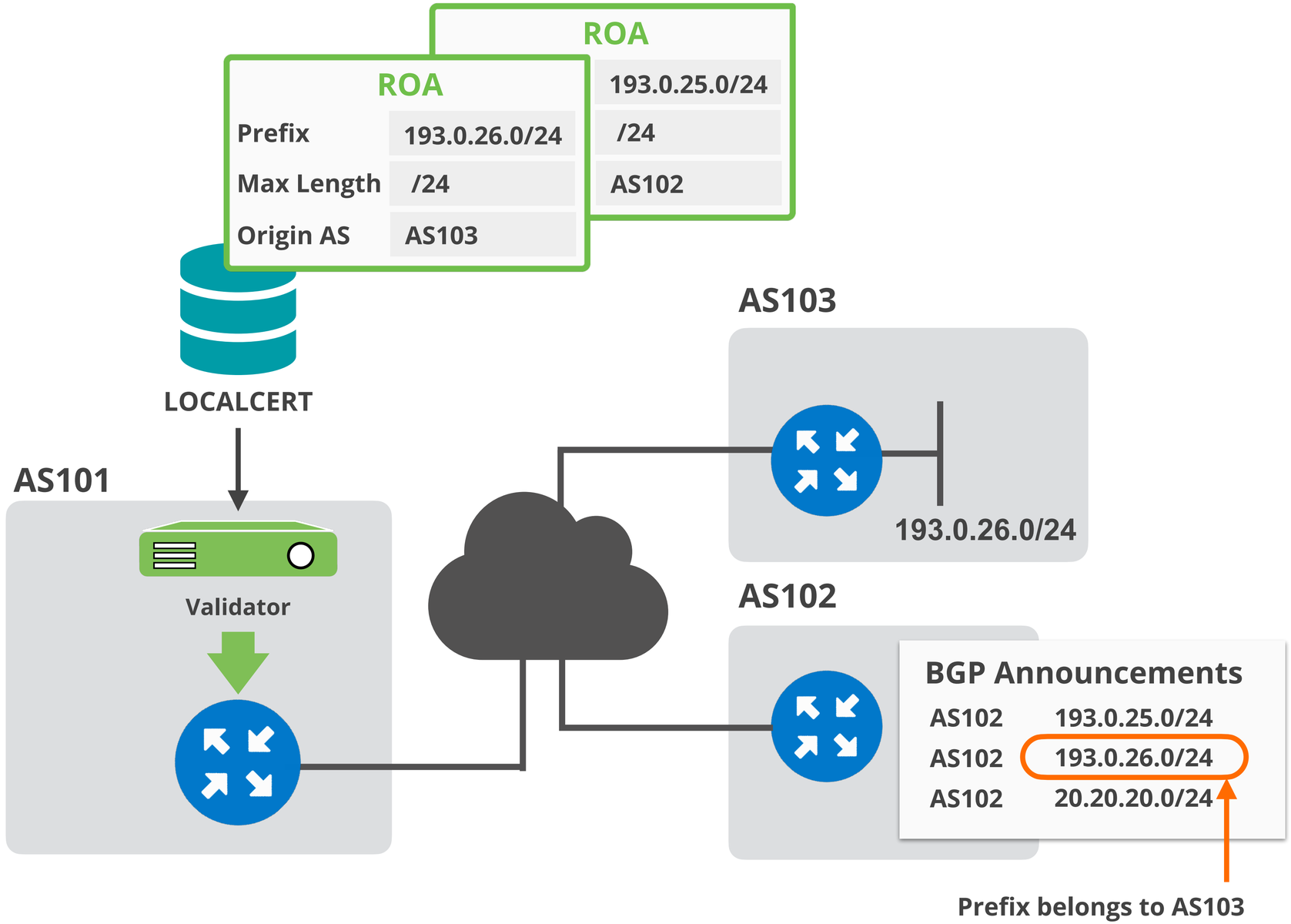
| | | |
|---|---|---|
| **Valid** | → | **Accept** the prefix |
| **Invalid** | → | **Discard** the prefix |
| **NotFound** | → | **Accept** the prefix |

**Do not consider dropping prefixes with “NotFound” RPKI validation state!**

# Discarding BGP Invalids

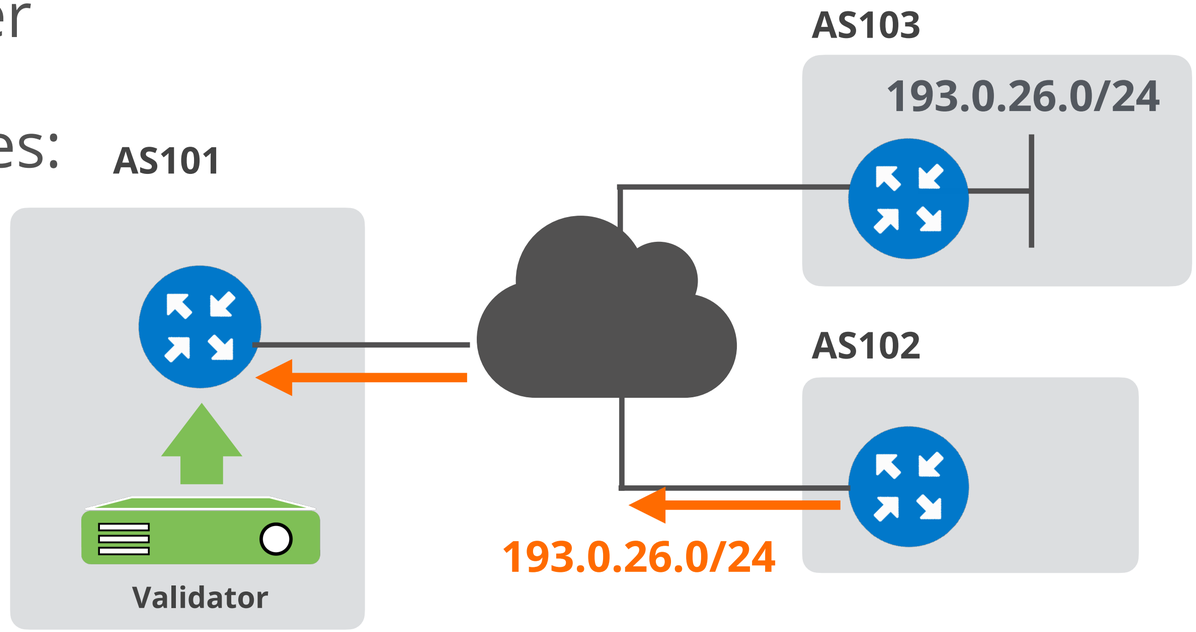
- Major networks are dropping invalid BGP prefixes!
  - Telia, AT&T, Cloudflare, Netflix, Swisscom, Cogent, …
- April 2021, RIPE NCC (AS3333) started dropping invalids too!
  - only networks with RPKI **Valid** or **Unknown** announcements are allowed
  - K-Root (AS25152) is not part of AS3333

# DEMO: ROV with RPKI



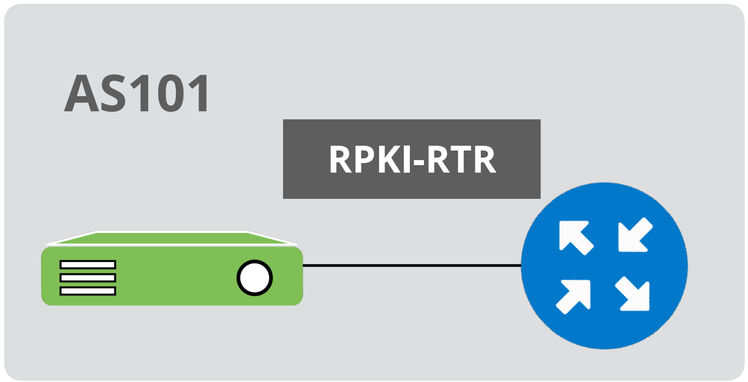
# Demo Setup

- Validators : **FORT** and **Routinator**
  - Both are installed, preconfigured and running!
- ROV will be configured on AS101 router
- AS102 announces the following prefixes:
  - its own prefix (**193.0.25.0/24**)
  - AS103 prefix (**193.0.26.0/24**)
  - a prefix without a ROA (**20.20.20.0/24**)




https://rpki.readthedocs.io/en/latest/ops/tools.html#relying-party-software

# Step-1: Set up validator connection




**On AS101 router**

```
(config)# conf t
(config)# router bgp 101
(config-router)# bgp rpki server tcp 100.64.1.1 port 3323 refresh 300
(config-router)# bgp rpki server tcp 100.64.1.1 port 323 refresh 300
```

Routinator

FORT

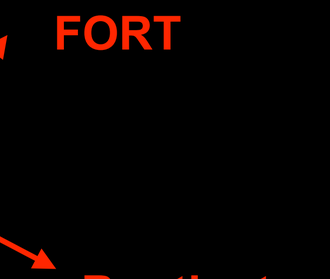
**RPKI Router Configurations...**

https://www.ripe.net/manage-ips-and-asns/resource-management/rpki/router-configuration

# Step-2: Verify  Validator connection and VRPs

```
U1_Router#show ip bgp rpki servers | i ESTAB

Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

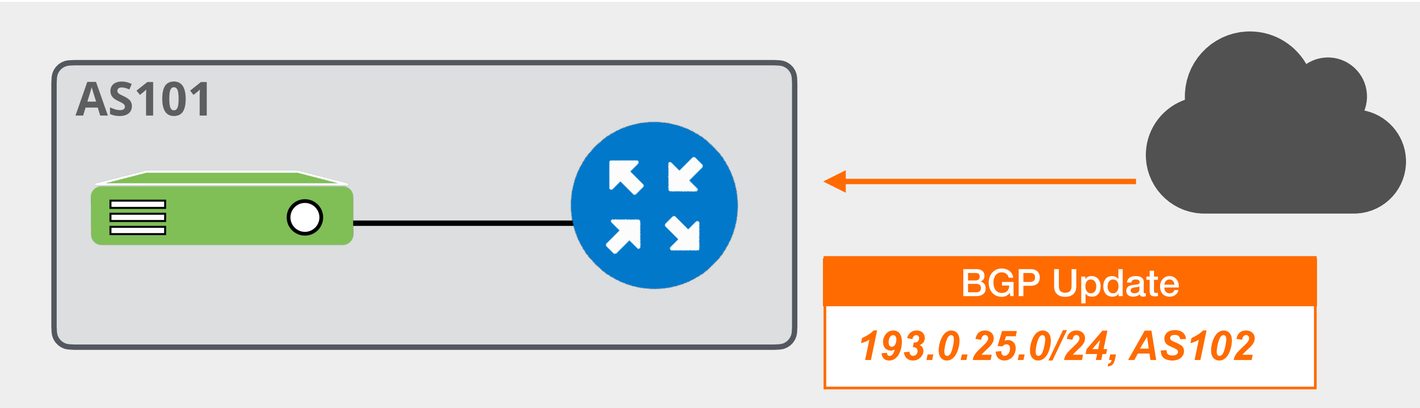
```
U1_Router#sho ip bgp rpki table
1547 BGP sovc network entries using 247520 bytes of memory

3851 BGP sovc record entries using 123232 bytes of memory

Network            Maxlen  Origin-AS  Source  Neighbor

5.32.168.0/21      21      15836      0       100.64.1.1/323
5.32.168.0/21      21      15836      0       100.64.1.1/3323
5.35.224.0/19      24      8972       0       100.64.1.1/323
5.35.224.0/19      24      8972       0       100.64.1.1/3323
5.35.224.0/19      24      29066      0       100.64.1.1/323
5.35.224.0/19      24      29066      0       100.64.1.1/3323
```



# Step-3: Check validation result

```
U1_Router#show  ip bgp 193.0.25.0/24
BGP routing table entry for 193.0.25.0/24, version 1598443
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      path 7FD8EAB30678 RPKI State valid
      rx pathid: 0, tx pathid: 0x0
```

| ROA | |
|---|---|
| Prefix | 193.0.25.0/24 |
| Max Length | /24 |
| Origin AS | AS102 |

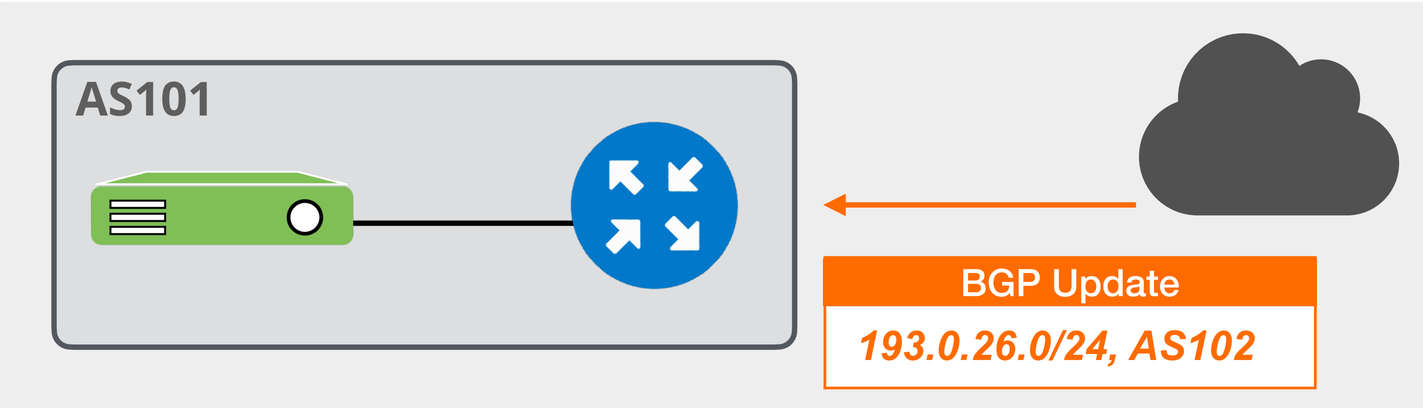


# Step-3: Check validation result

Prefix belongs to AS103!

```
U1_Router#show  ip bgp 193.0.26.0/24
BGP routing table entry for 193.0.26.0/24, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external
      path 7FD8EAB30708 RPKI State invalid
      rx pathid: 0, tx pathid: 0
```

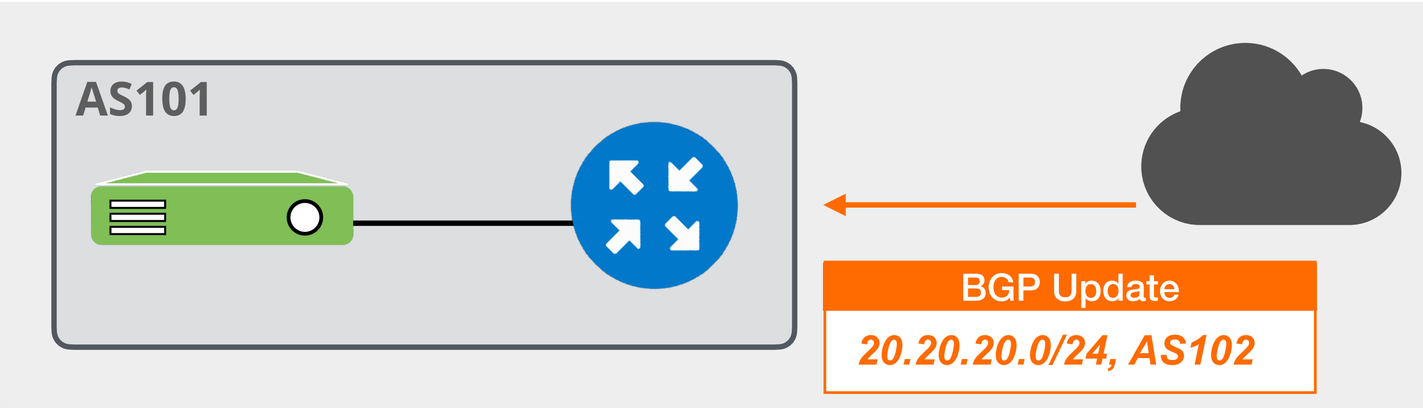
| ROA | |
|---|---|
| Prefix | 193.0.26.0/24 |
| Max Length | /24 |
| Origin AS | AS103 |



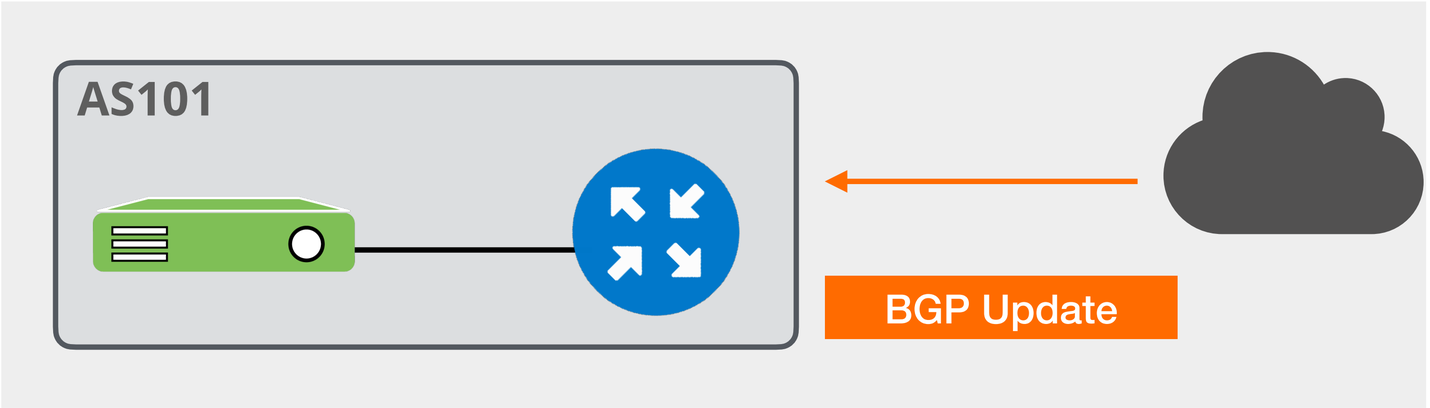
# Step-3: Check validation result

```
U1_Router#show  ip bgp 20.20.20.0/24
BGP routing table entry for 20.20.20.0/24, version 1598444
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      path 7FD8EAB305E8 RPKI State not found
      rx pathid: 0, tx pathid: 0x0
```

No ROA exits for this prefix!



# Step-4: Discard Invalids




**On AS101 router**

```
(config-router)# route-map rpki-reject deny 10
(route-map)# match rpki invalid
(route-map)# route-map rpki-reject permit 20
```

# Conclusion

- Have proper filters in place!
  - IRR based filters particularly for customer routes
- Protect your prefixes with ROAs
- ROV prevents large fraction of hijacks and route leaks
- Deploying RPKI is not that difficult and brings big benefits
- Go for it if you haven't yet!

# Questions