RIPE NCC
RIPE NETWORK COORDINATION CENTRE

BGP Operations and Security

Training Course

RIPE NCC Learning & Development

Please Follow Our Safety Measures

Self-test

Please test yourself and keep everyone safe

Get a mask

Use a mask if this is required at the location

Sanitise your hands

Particularly when entering and leaving rooms

Respect each other's space

Red sticker: Please keep 1.5 m distance
Yellow sticker: Elbow bumps only please
Green sticker: You can stand closer

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**

Introduction

- Name
- Experience
  - Routing
  - BGP
- Does your organisation have an AS number ?
- Do you have a RIPE NCC Access account ?
- Goals

Overview

Day 1

- Introduction to BGP
- Running BGP
- BGP Attributes
- Traffic Engineering
- iBGP Scalability
- Multiprotocol BGP

Day 2

- BGP & Routing Security
- Internet Routing Registry
- Filtering
- RPKI
- What else?
- BGP Tips & Tricks

Overview

Day 1

- Introduction to BGP
- Running BGP
- BGP Attributes
- Traffic Engineering
- iBGP Scalability
- Multiprotocol BGP

Day 2

- BGP & Routing Security
- Internet Routing Registry
- Filtering
- RPKI
- What else?
- BGP Tips & Tricks

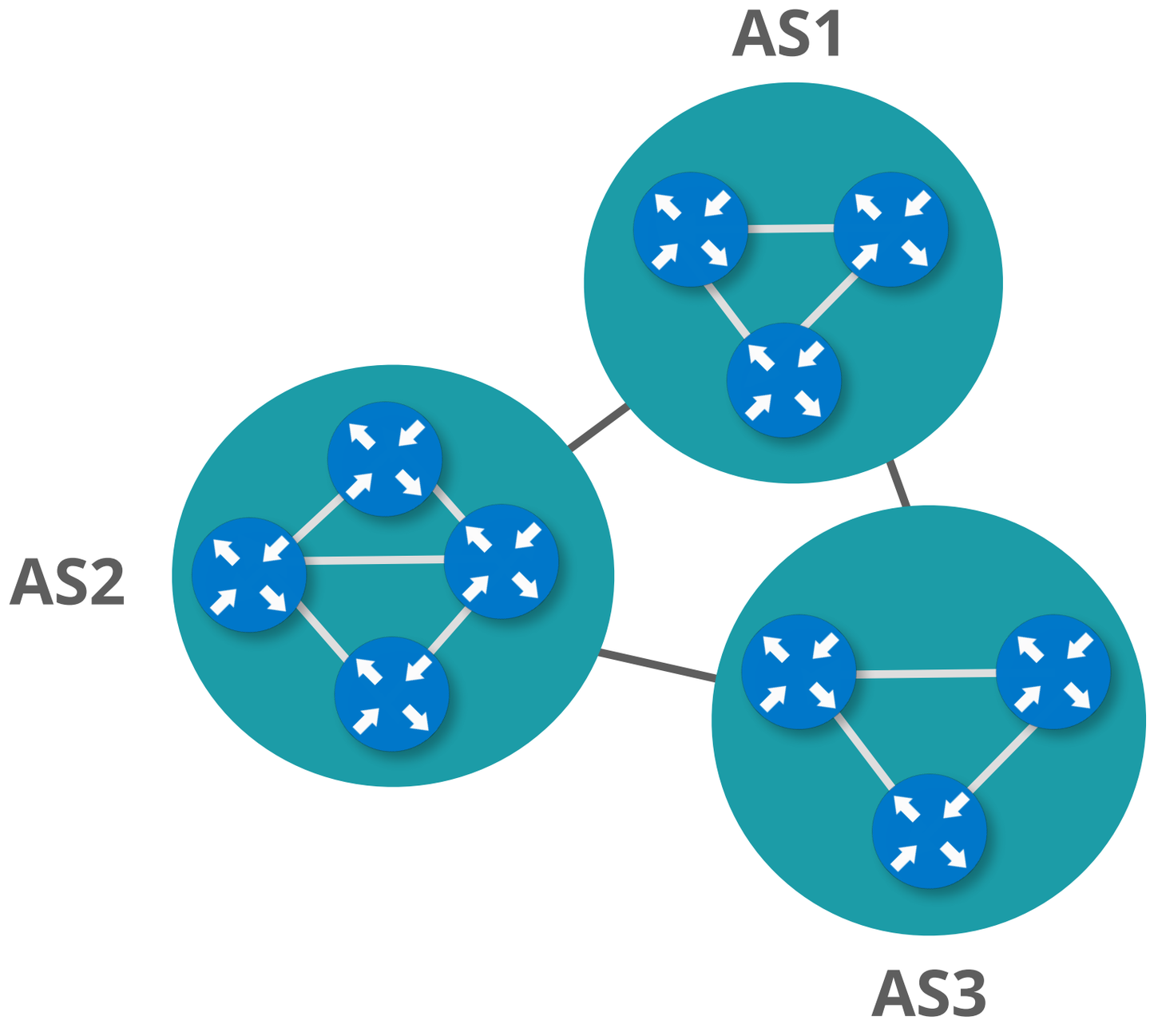
Introduction to BGP

Section 1

The Internet

- Who runs the Internet?
  - No one (in particular), not ICANN, nor the RIRs, nor the EU
- How does it keep working?
  - Internet by and large functions for the common good
  - Business relationships and the need for reachability
- Any help to keep it working?
  - No central coordination
  - Many individuals and organisations

Internet relations



IGP - Interior Gateway Protocol

vs

EGP - Exterior Gateway Protocol

Interior Gateway Protocols

- **IGP (OSPF, IS-IS, EIGRP)**
  - Reachability and path info **within** a network domain
  - Provides Next hop address and an egress interface to any known destination address

Exterior Gateway Protocols

- **EGP**
  - Reachability and path info **between** network domains
  - Only provides a Next Hop address to a destination prefix
  - This has to be resolved to an egress interface using a second route lookup

BGP

Border Gateway Protocol

Routing Protocol for exchanging information between networks

RFC 4271

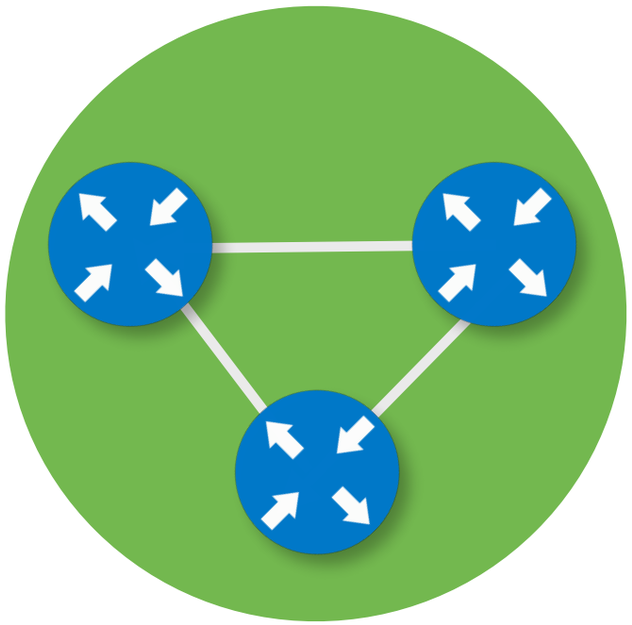
Sources of information

RFC4276 - implementation report

RFC4277 - operational experiences

Autonomous Systems

- A network controlled by a single entity
  - Same interior and exterior routing policy
  - Can also be a group of networks



Autonomous System
AS12345

Who distributes AS numbers?

- AS numbers are distributed by Regional Internet Registries
- In our region: RIPE NCC
  RIPE NETWORK COORDINATION CENTRE

AS Numbers

16-bit ASN

0 to 65535

32-bit ASN

65536 to 4294967295

Unlike IPv4 and IPv6, they are **interoperable**

ASNs - Special Use

Documentation

64496 to 64511

65536 to 65551

Private Use

64512 to 65534

4200000000 to 4294967294

Reserved

0

23456

65535

4294967295

AS23456

- All software now supports 32-bit ASNs
- Placeholder for 32 bit AS numbers
  - where AS32 is not supported

Path Vector Protocol

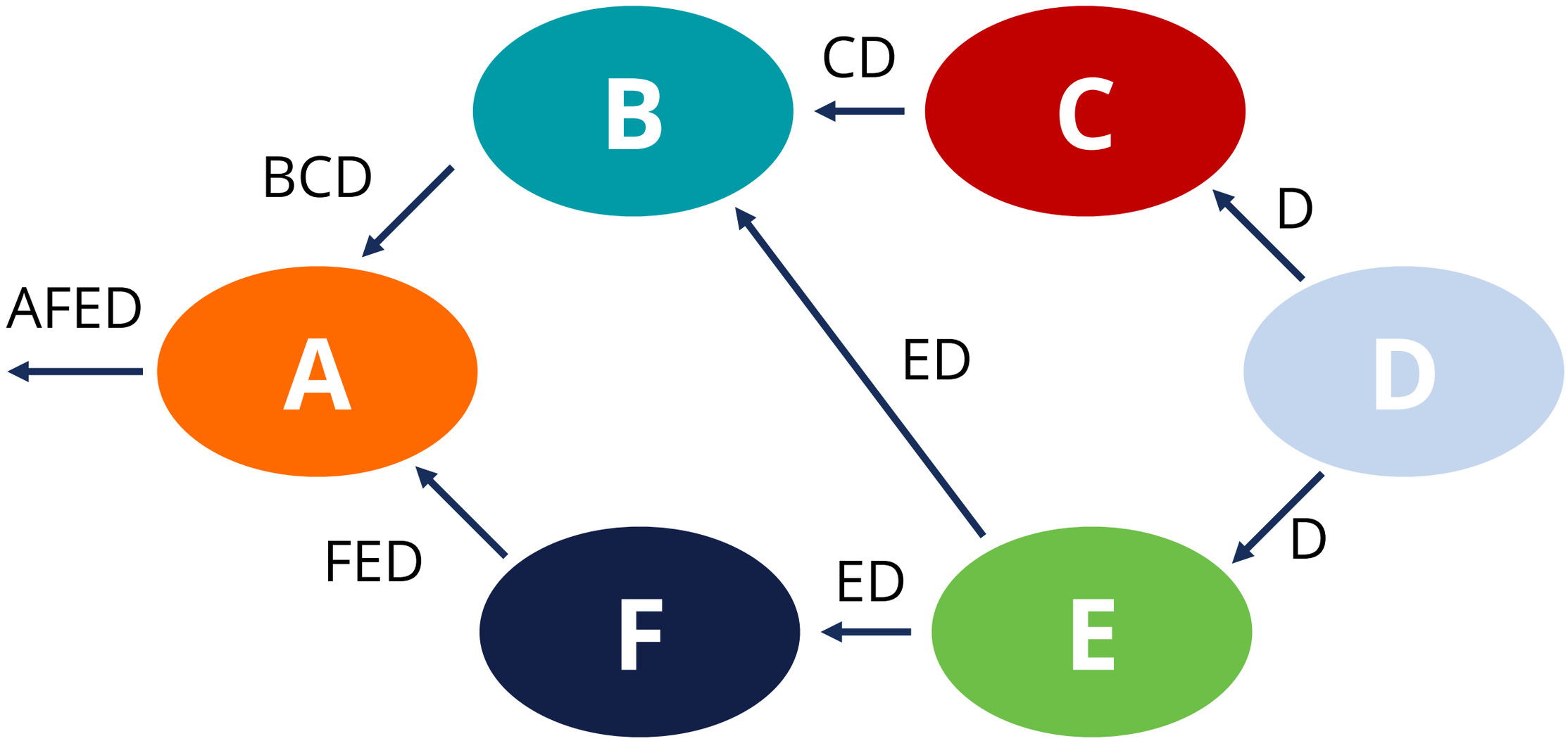
- Maintains and dynamically updates the path information
- Uses the **AS_PATH** attribute

AS777, AS3232, AS9843247, AS23242

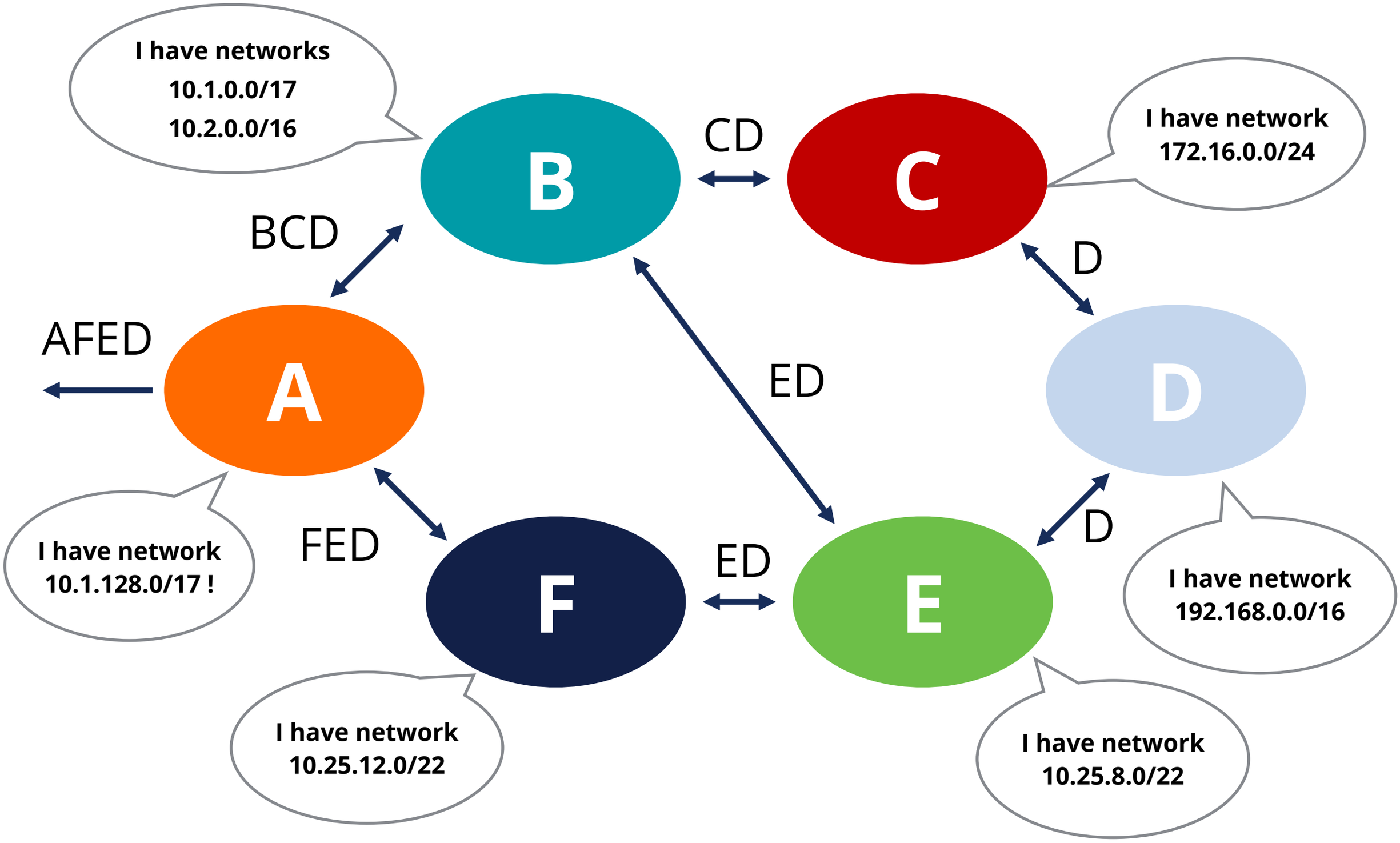
Path Vector Protocol Features

- If own AS is detected, then path is discarded
  - simple loop prevention mechanism
- Shorter paths are preferred

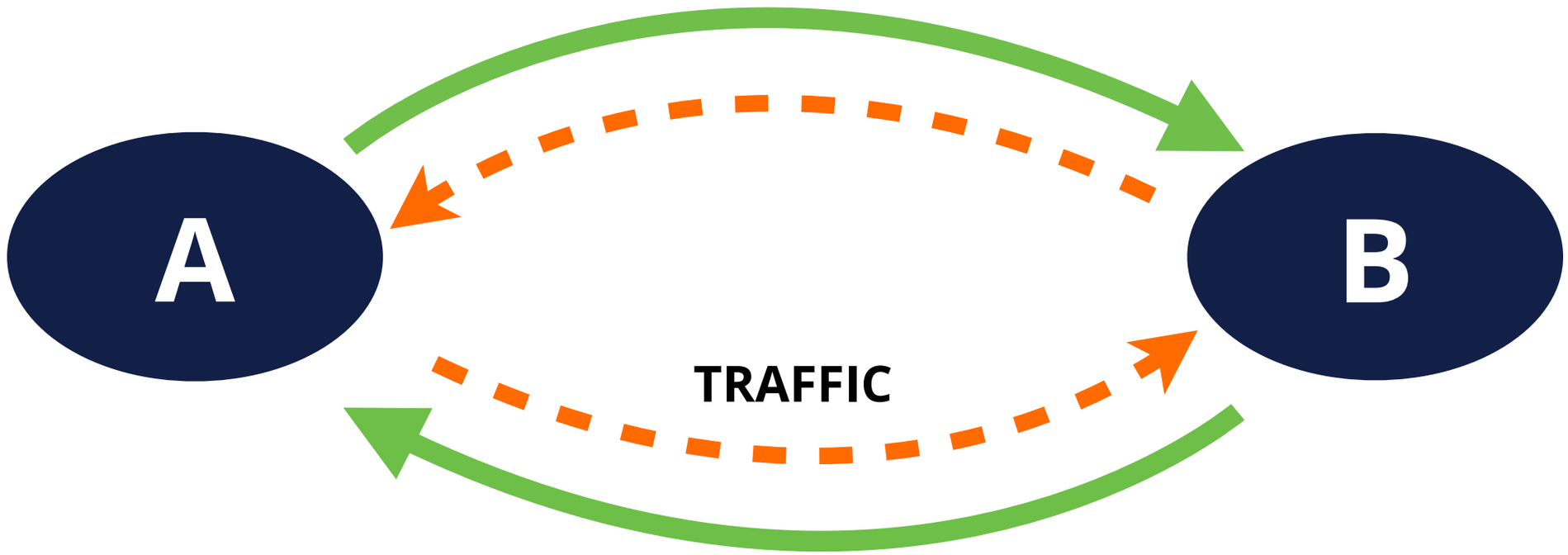
AS Path



Announcements

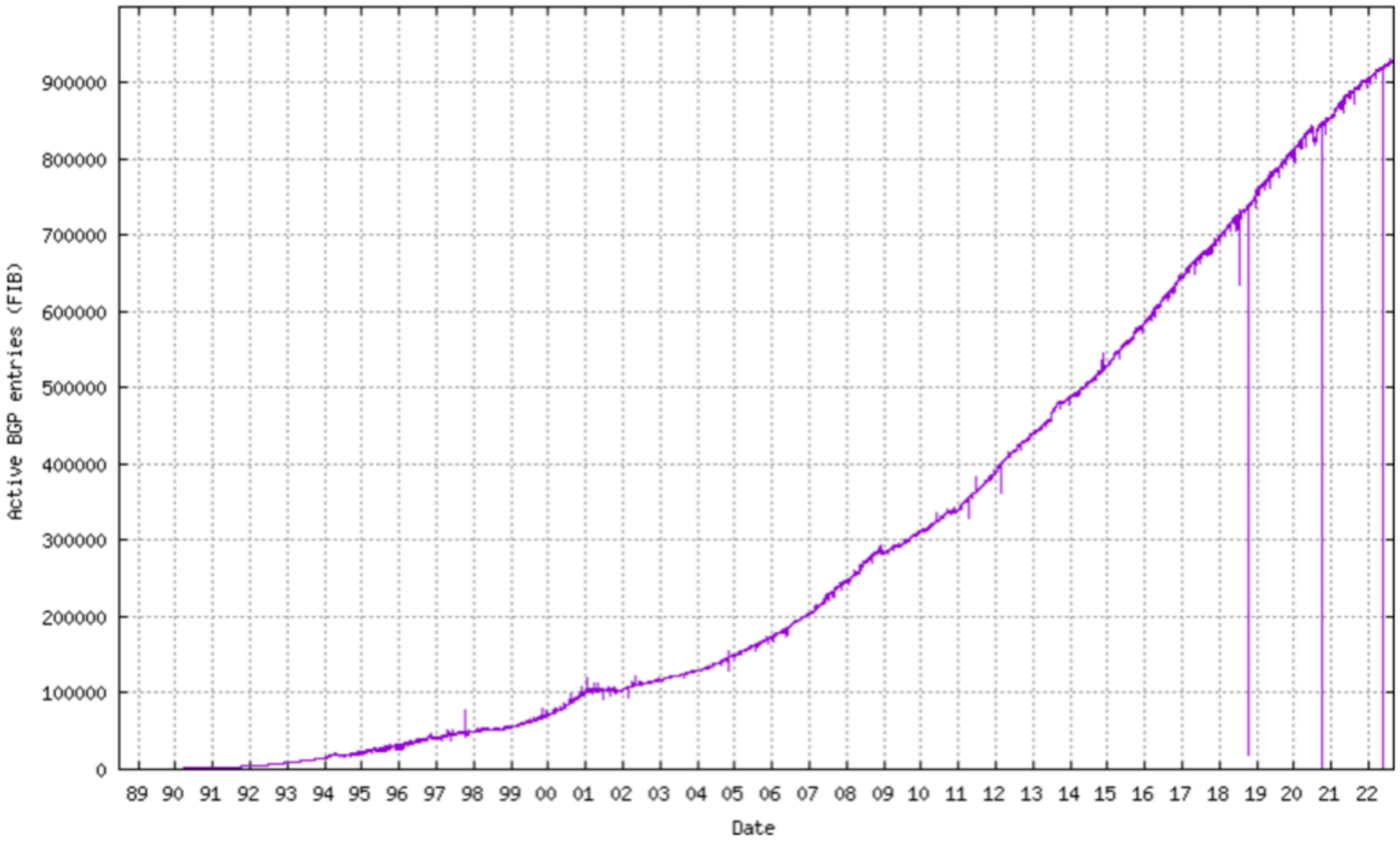


Traffic Direction vs Announcement



Default-Free Zone

- Internet routers which have explicit routing information about the rest of the Internet



eBGP - External BGP

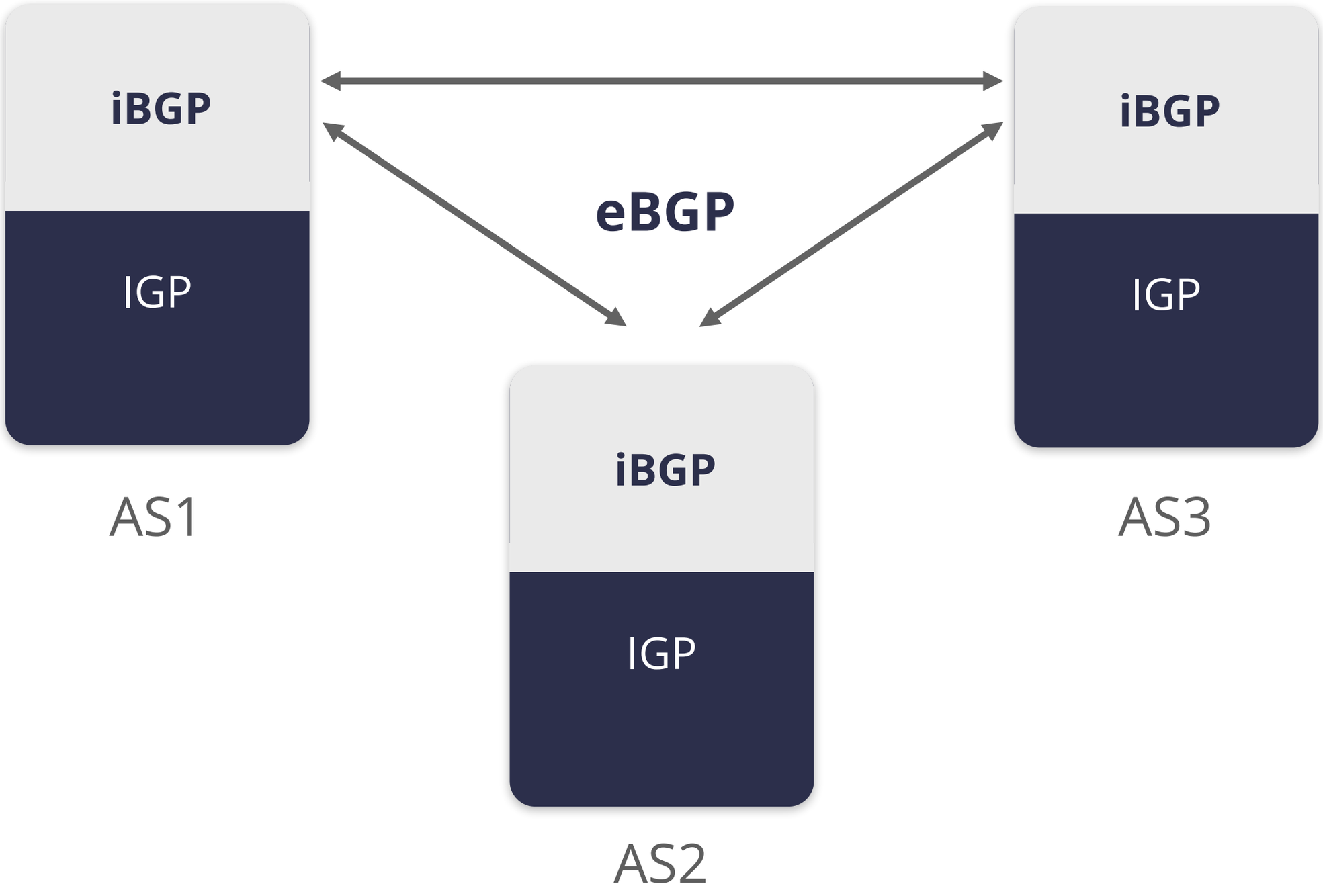
vs

iBGP - Internal BGP

- **eBGP -** External BGP
  - BGP neighbour relationship between two peers belonging to different AS
  - Prefix interchange with external peers and upstreams
  - Most routing policy located here

- **iBGP -** Internal BGP
  - BGP neighbour relationship within the same AS
  - Routes customer prefixes around internal infrastructure

Operator Model



Do I need BGP?

- Single homed
  - Static default route or distributed via IGP or private ASN
  - Operator takes responsibility for **reachability** of your prefix

Do I need BGP?

- Multi homed with the **same transit**
  - Multiple default routes to different networks
  - **Required** on downstream and upstream

Do I need BGP?

- Multiple upstreams, **no transit**
  - BGP is **optional**
  - No need to receive full global routing table
  - Still control over its own routing policy

Do I need BGP?

- Multiple upstreams, **providing transit**
  - BGP is **required**
  - Need to announce customer prefixes

Your first BGP Session

Activity

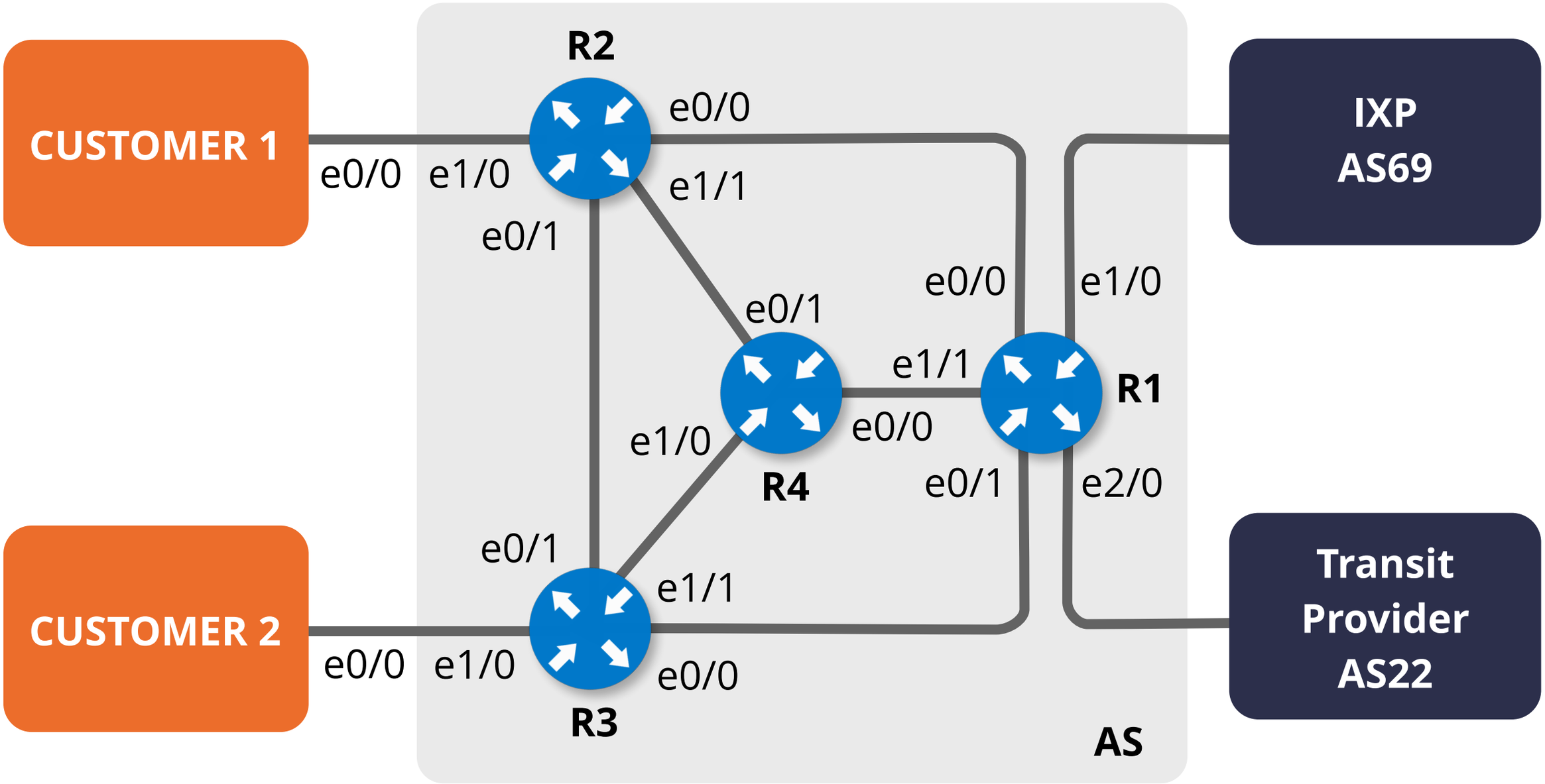
Login to the Labs

- Make sure you have connectivity
- Go to: workbench.ripe.net
  - Choose the correct lab, trainer will provide info
  - Your login is your number
  - Trainer will provide you with the password
- No password required to login to the routers!

Discover the Network

- Routing Protocol
  - IGP (OSPF) is used for loopback addresses and point-to-point links
  - No EGP (BGP) configuration
- R1 announces a default route via OSPF
  - Keeps routing tables in the area smaller
  - All inter-area traffic must pass R1

Network Diagram



Assignment

- Connect your network to Transit Provider
- Connect your network to Internet Exchange
- Data you will need:
  - Your AS number
  - Your IP address space (IPv4 and IPv6)
  - The AS number of your neighbours
  - The IP address of your neighbours BGP routers

Preparation (on R1)

- Insert static Null route
  - Before BGP advertises a network, it checks for an exact match in the router's routing table

```
(config)# ip route 10.X.0.0 255.255.252.0 null0 250
```

Configure IXP Interface (on R1)

- Identify and enable your IXP interface

```
(config)# interface Ethernet1/0
(config-if)# no shutdown
```

- Configure IXP interface IP address

```
(config)# interface Ethernet1/0
(config-if)# ip address 172.16.0.X 255.255.255.0
```

- Test if IXP routers are reachable

```
# ping 172.16.0.66
# ping 172.16.0.99
```

Configure Transit Interface (on R1)

- Identify and enable your transit interface

```
(config)# interface Ethernet2/0
(config-if)# no shutdown
```

- Configure transit interface IP address

```
(config)# interface Ethernet2/0
(config-if)# ip address 10.132.X.2 255.255.255.252
```

- Test if transit provider router is reachable

```
# ping 10.132.X.1
```

Create a filter (on R1)

- BGP sends the best paths to all neighbours
  - updated by RFC 8212 “Route Propagation Behavior without Policies” for eBGP

```
(config)# ip prefix-list TRANS-OUT-V4 seq 5 permit 10.X.0.0/22
(config)# ip prefix-list IXP-OUT-V4 seq 5 permit 10.X.0.0/22
```

Configure Transit Session (on R1)

- Configure BGP session with AS22

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor 10.132.X.1 remote-as 22
(config-router)# neighbor 10.132.X.1 prefix-list TRANS-OUT-V4 out
```

- How to advertise a route
  - redistribution
  - network statement

```
(config-router)# network 10.X.0.0 mask 255.255.252.0
```

Configure IXP Sessions

- Configure BGP session with AS69

```
(config)# router bgp 1XX
(config-router)# neighbor 172.16.0.66 remote-as 69
(config-router)# neighbor 172.16.0.66 prefix-list IXP-OUT-V4 out
(config-router)# neighbor 172.16.0.99 remote-as 69
(config-router)# neighbor 172.16.0.99 prefix-list IXP-OUT-V4 out
```

Verify

- Check sessions summary

```
# show ip bgp summary
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
# show ip bgp neighbors <peer IP> advertised-routes
```

- Verify reachability

```
# ping 10.132.32.1
# ping <your colleague R1 IP>
```

Questions

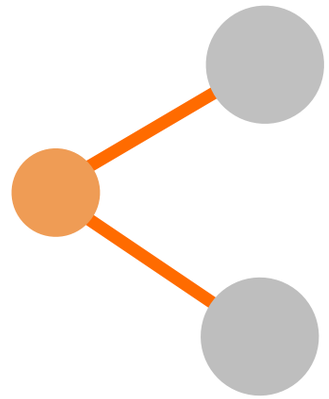
Running BGP

Section 2

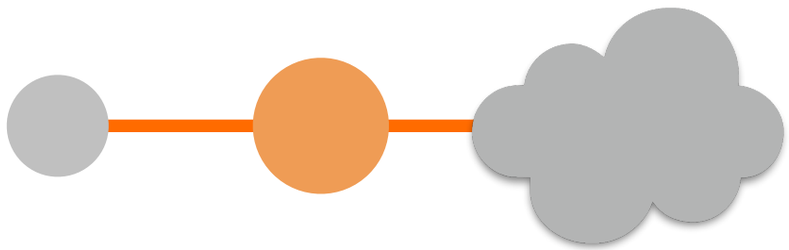
AS Types

Stub

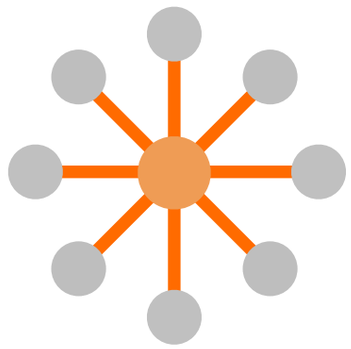
Multi homed



Transit



IXP - Internet Exchange Point



BGP Operations - 1

Neighbours open **TCP** connection (*port 179*)

BGP session is established

BGP exchanges routes with neighbours

...

BGP Operations - 2

Best BGP path is selected and installed into BGP table

The best path is sent to BGP neighbours

BGP neighbours periodically exchange **keep-alive** messages

BGP Messages

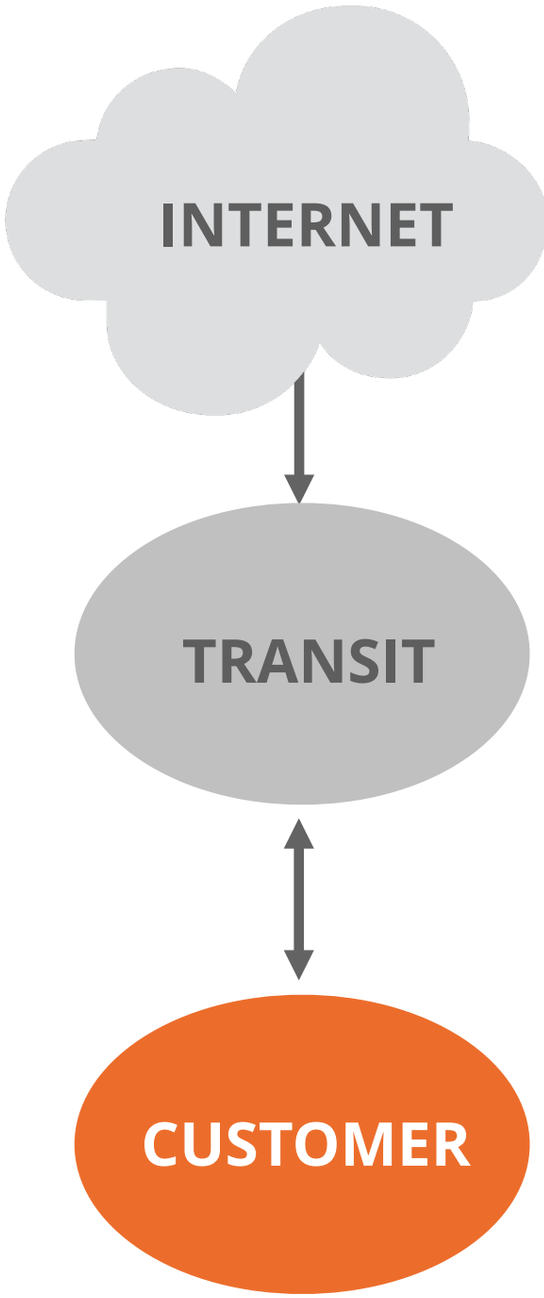
- **Open**
  - **Information about the local BGP speaker**
    - Version and Hold time
    - AS number and Router-ID
  - **BGP Capabilities Advertisement -** *RFC 2842*
    - Multiprotocol
    - Route Refresh
    - 32 bit ASN

BGP Messages

- **Keepalive**
  - Verify BGP session
- **Update**
  - New or unreachable routes and path attributes
- **Notification**
  - Indicate an error condition

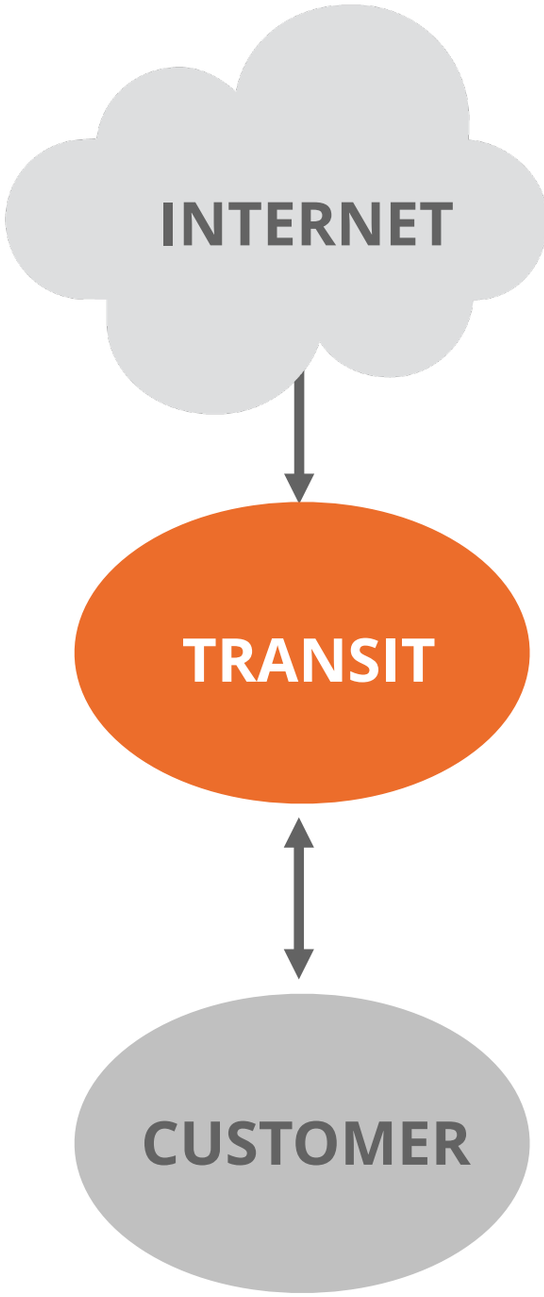
You receive a BGP Transit

- **“Upstream**” network
- Connects you to the rest of the internet
  - by giving a **full BGP routing table**
  - or just the **default route**
- You announce them your prefixes



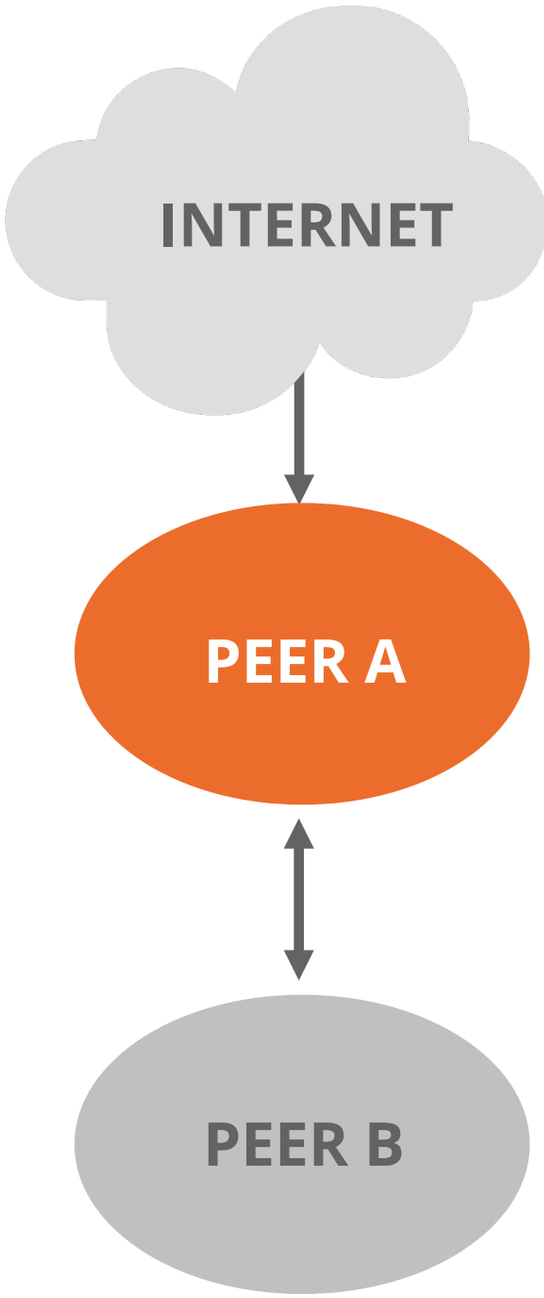
You have a BGP Customer

- “**Downstream**” network
  - You connect them to the internet
  - You give them
    - a **full BGP table**
    - or **a default route**
- You receive your customers’ routes
  - And, in specific cases, their customers’



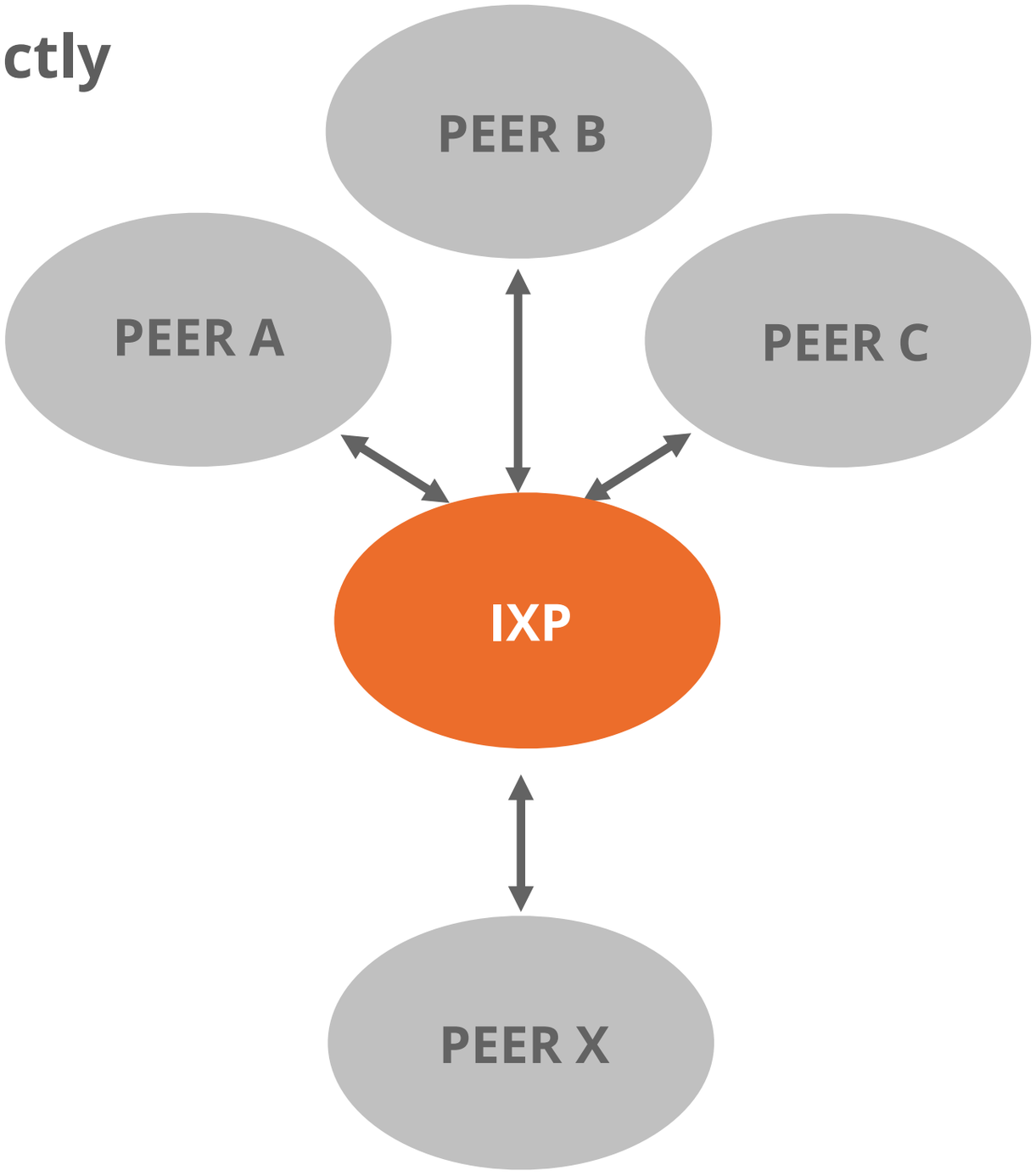
BGP Peering

- Usually peer with you at IXPs
  - Gives you access to their network
  - And/or their customers
- You announce them only your route
  - And your customers’



Internet Exchanges (IX or IXP)

- A switch (or set of switches) that allows members to exchange traffic **directly**
- Many countries have at least one
  - AMS-IX, LINX, VIX, MIX, etc



IXPs - Why

- Traffic remains **local**
- Improve **routing efficiency**
- Reduce costs (less transit)

IXPs - Architecture

- A switch, or a group of switches
  - Range is generally from 100Mb to 100Gb ports
- Switches are in colocation facilities
  - Easy to reach them
  - Can be spread in different facilities across a city or region
- Some IXPs have two LANs for redundancy

IXPs - Route Servers

- A server running a BGP Daemon
- Helps networks who peer at many IXPs
  - Avoids setting up a meshed environment
  - Eases management

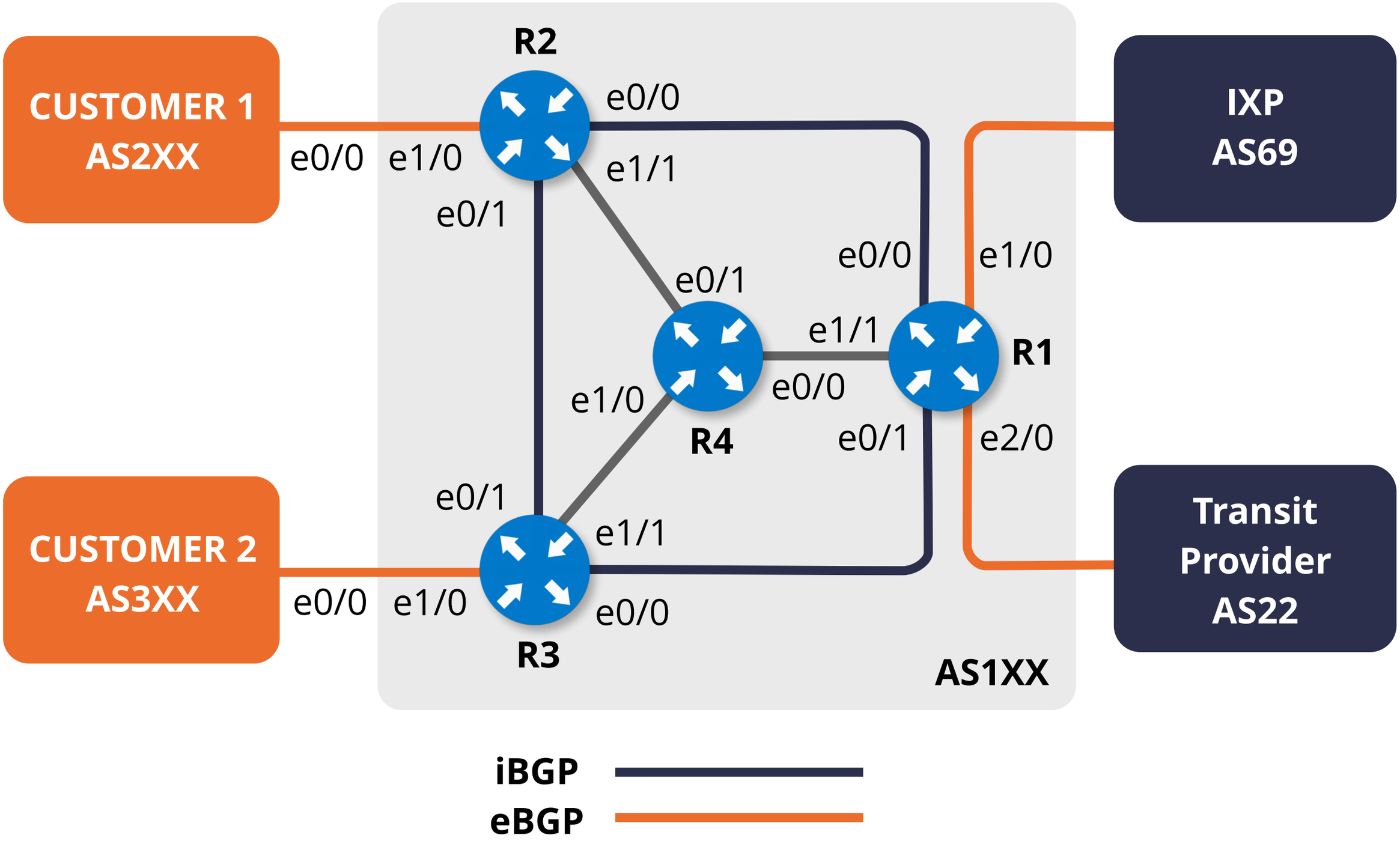
IXPs - Route Servers

- Sets **next-hop** as announcer, leaving itself out
- Traffic does not flow through the route server

Connecting BGP Customers

Activity

Network Diagram



Assignment

- Connect Customer 1 and 2 using BGP
  - Customers will use prefixes from your address space
- Data you will need
  - Your AS number
  - Your IP address space
  - The AS number of your customers
  - The IP address of your customers’ BGP routers

Preparation

- Remove default routes from C1 and C2

```
(config)# no ip route 0.0.0.0 0.0.0.0
```

Using loopbacks

- Better to use Loopback address than Interface address
  - Session is not dependent on state of a single interface
  - Session is not dependent on physical topology
- Can be propagated by IGP
  - IS-IS or OSPF

iBGP Configuration R1

- BGP configuration of Router 1 on top of IP core

```
(config)# router bgp 1XX
(config-router)# neighbor 172.X.255.2 remote-as 1XX
(config-router)# neighbor 172.X.255.2 update-source lo0
(config-router)# neighbor 172.X.255.2 next-hop-self
(config-router)# neighbor 172.X.255.3 remote-as 1XX
(config-router)# neighbor 172.X.255.3 update-source lo0
(config-router)# neighbor 172.X.255.3 next-hop-self
```

BGP Configuration R2 and C1

- BGP configuration of Router 2

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor 172.X.255.1 remote-as 1XX
(config-router)# neighbor 172.X.255.1 update-source lo0
(config-router)# neighbor 172.X.255.1 next-hop-self
(config-router)# neighbor 172.X.255.3 remote-as 1XX
(config-router)# neighbor 172.X.255.3 update-source lo0
(config-router)# neighbor 172.X.255.3 next-hop-self
(config-router)# neighbor 10.X.0.26 remote-as 2XX
```

- BGP configuration of Customer 1

```
(config)# router bgp 2XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.1.0 mask 255.255.255.0
(config-router)# neighbor 10.X.0.25 remote-as 1XX
```

BGP Configuration R3 and C2

- BGP configuration of Router 3

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor 172.X.255.1 remote-as 1XX
(config-router)# neighbor 172.X.255.1 update-source lo0
(config-router)# neighbor 172.X.255.1 next-hop-self
(config-router)# neighbor 172.X.255.2 remote-as 1XX
(config-router)# neighbor 172.X.255.2 update-source lo0
(config-router)# neighbor 172.X.255.2 next-hop-self
(config-router)# neighbor 10.X.0.30 remote-as 3XX
```

- BGP configuration of Customer 2

```
(config)# router bgp 3XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.2.0 mask 255.255.255.0
(config-router)# neighbor 10.X.0.29 remote-as 1XX
```

Verify

- Check sessions in summary

```
# show ip bgp neighbors | include BGP
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
```

- Verify reachability from customer

```
# ping 10.132.32.1
# ping 1.1.1.1
```

- Show logged events

```
# show logging
```

Create a filter for customers (on R1)

- Allow BGP to send paths of the customers

```
(config)# ip prefix-list TRANS-OUT-V4 seq 10 permit 10.X.1.0/24
(config)# ip prefix-list TRANS-OUT-V4 seq 15 permit 10.X.2.0/24
(config)# ip prefix-list IXP-OUT-V4 seq 10 permit 10.X.1.0/24
(config)# ip prefix-list IXP-OUT-V4 seq 15 permit 10.X.2.0/24
```

- And clear all the sessions

```
# clear ip bgp 10.132.X.1 out
# clear ip bgp 172.16.0.66 out
# clear ip bgp 172.16.0.99 out
```

Questions

BGP Attributes

Section 3

BGP Attributes

- Every prefix has a number of attributes
  - BGP puts multiple prefixes in a single update packet associated with the same attributes
- Used by local AS and remote AS for traffic engineering

BGP Attributes

Well known		Optional	
Mandatory	**Discretionary**	**Transitive**	**Non-Transitive**
AS-PATH	LOCAL-PREF	COMMUNITY	MED
NEXT-HOP			ORIGINATOR ID
ORIGIN			CLUSTER-LIST

BGP Attributes Classification

- Well-known mandatory
  - Must be supported by every BGP implementation
  - Must exist in every update, propagated to other neighbours
- Well-known discretionary
  - Must be supported by every BGP implementation
  - Doesn’t have to be present in every update

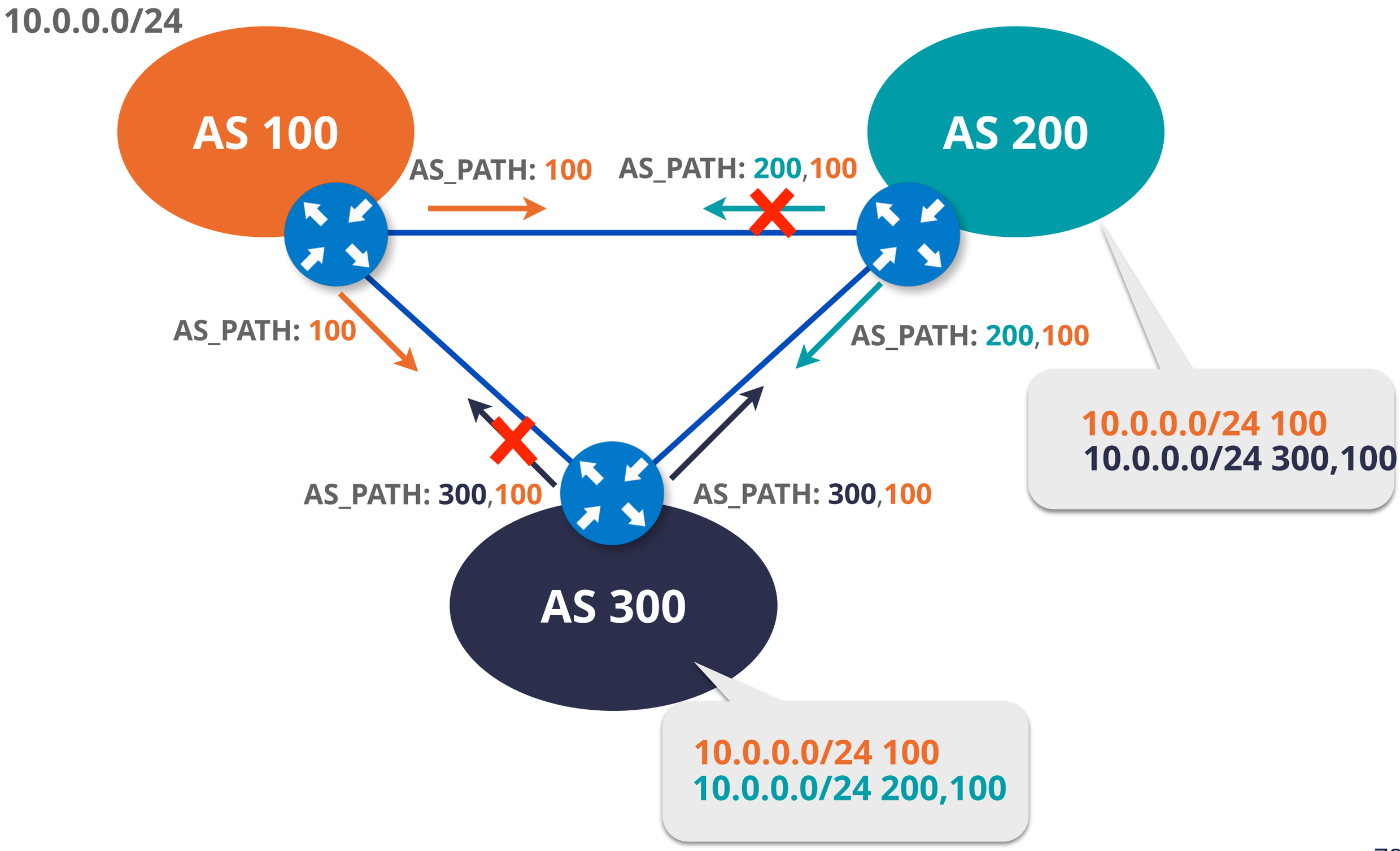
BGP Attributes Classification

- Optional transitive
  - Not necessary to be supported by all BGP implementations,
  - Doesn’t have to exist in every update
  - Propagated to other neighbours
- Optional non-transitive
  - Not necessary to be supported by all BGP implementations,
  - Doesn’t have to be in every update, discarded if not recognized

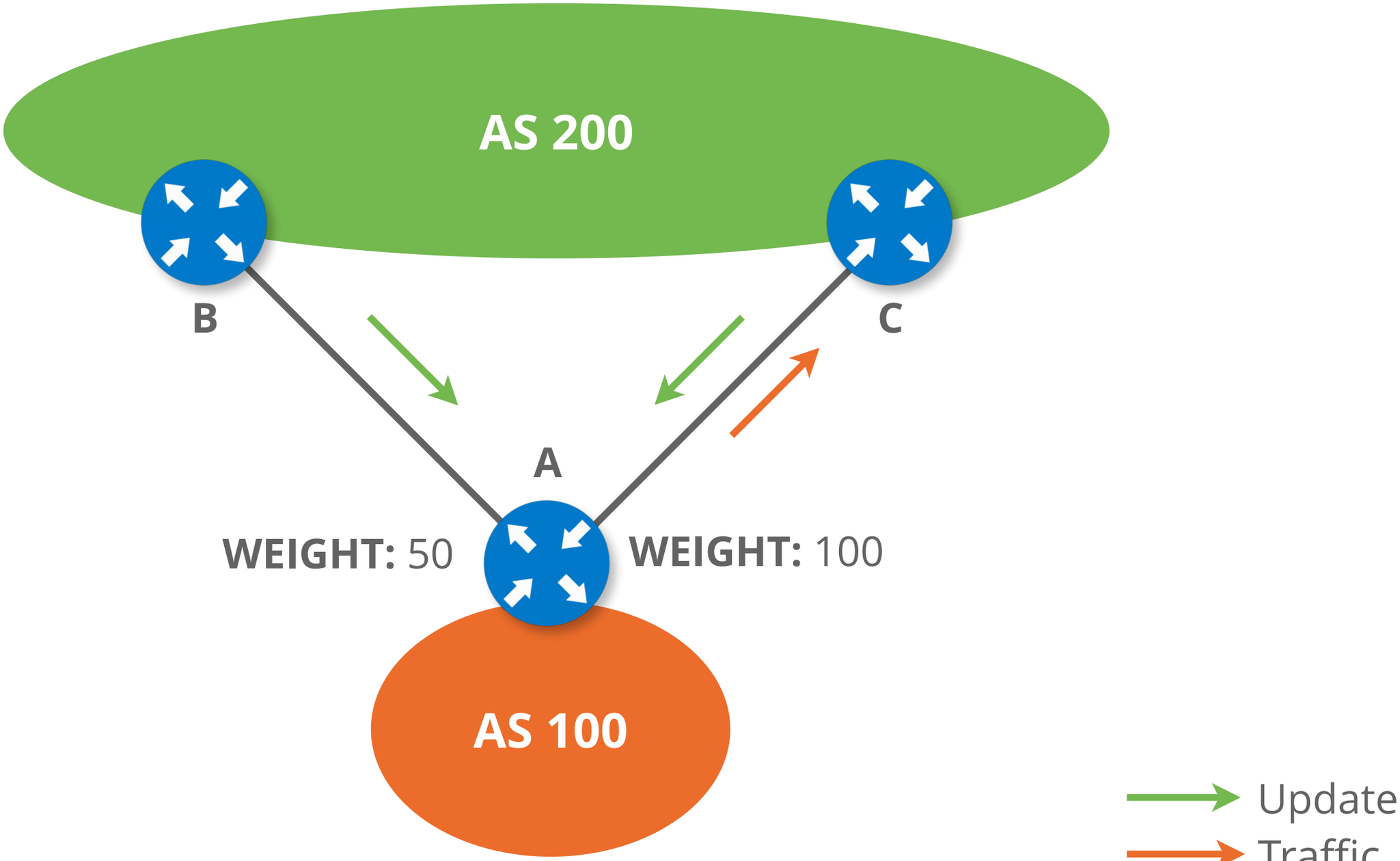
Attribute Propagation

Router:	Local AS:	Local AS + Neighbor:	Not limited:
Weight	Local-pref	MED	Origin communities

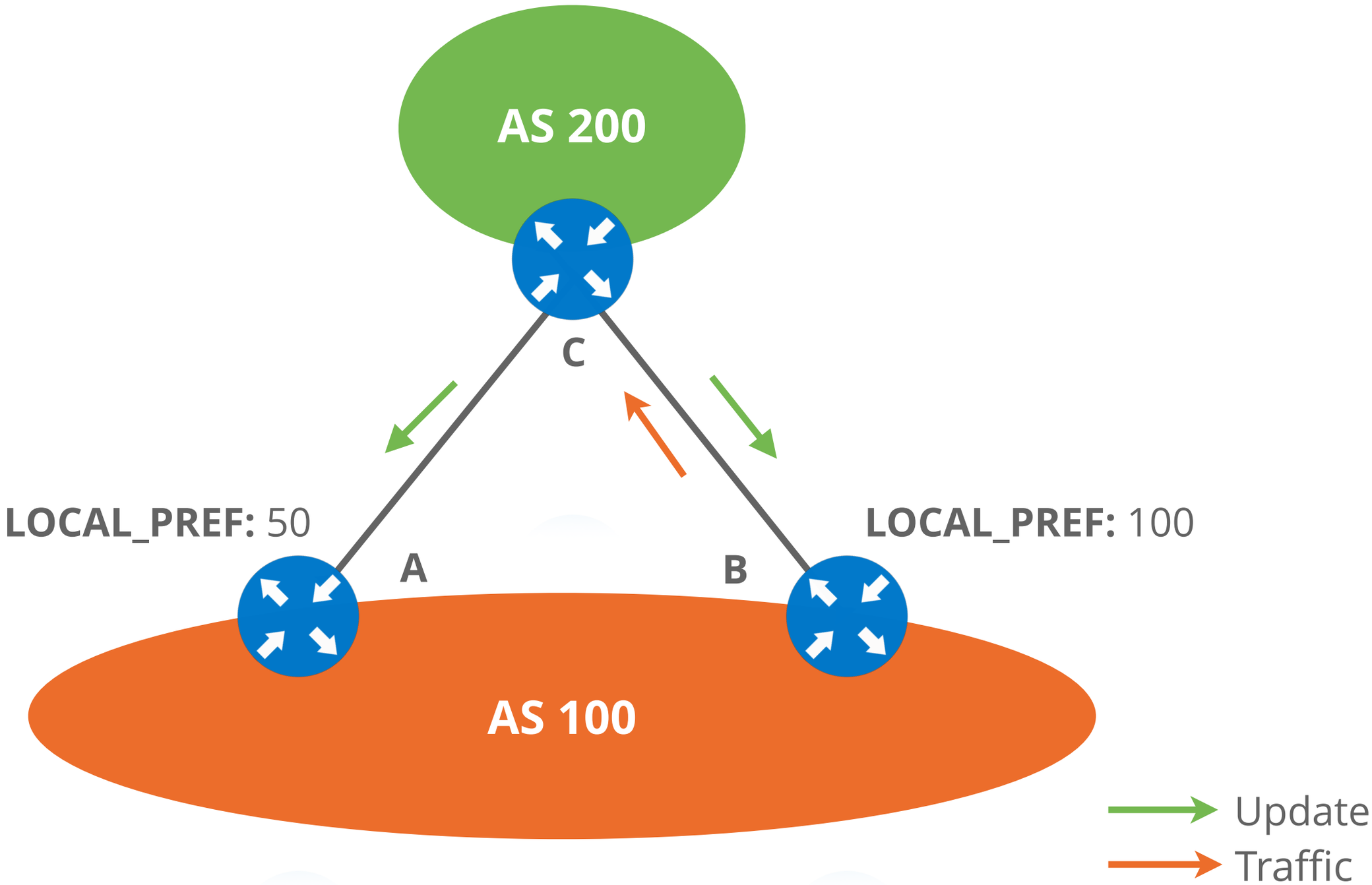
AS Path


10.0.0.0/24
AS 100
AS 200
AS_PATH: 100
AS_PATH: 200,100
AS_PATH: 100
AS_PATH: 200,100
AS_PATH: 300,100
AS_PATH: 300,100
AS 300
10.0.0.0/24 100
10.0.0.0/24 300,100
10.0.0.0/24 100
10.0.0.0/24 200,100

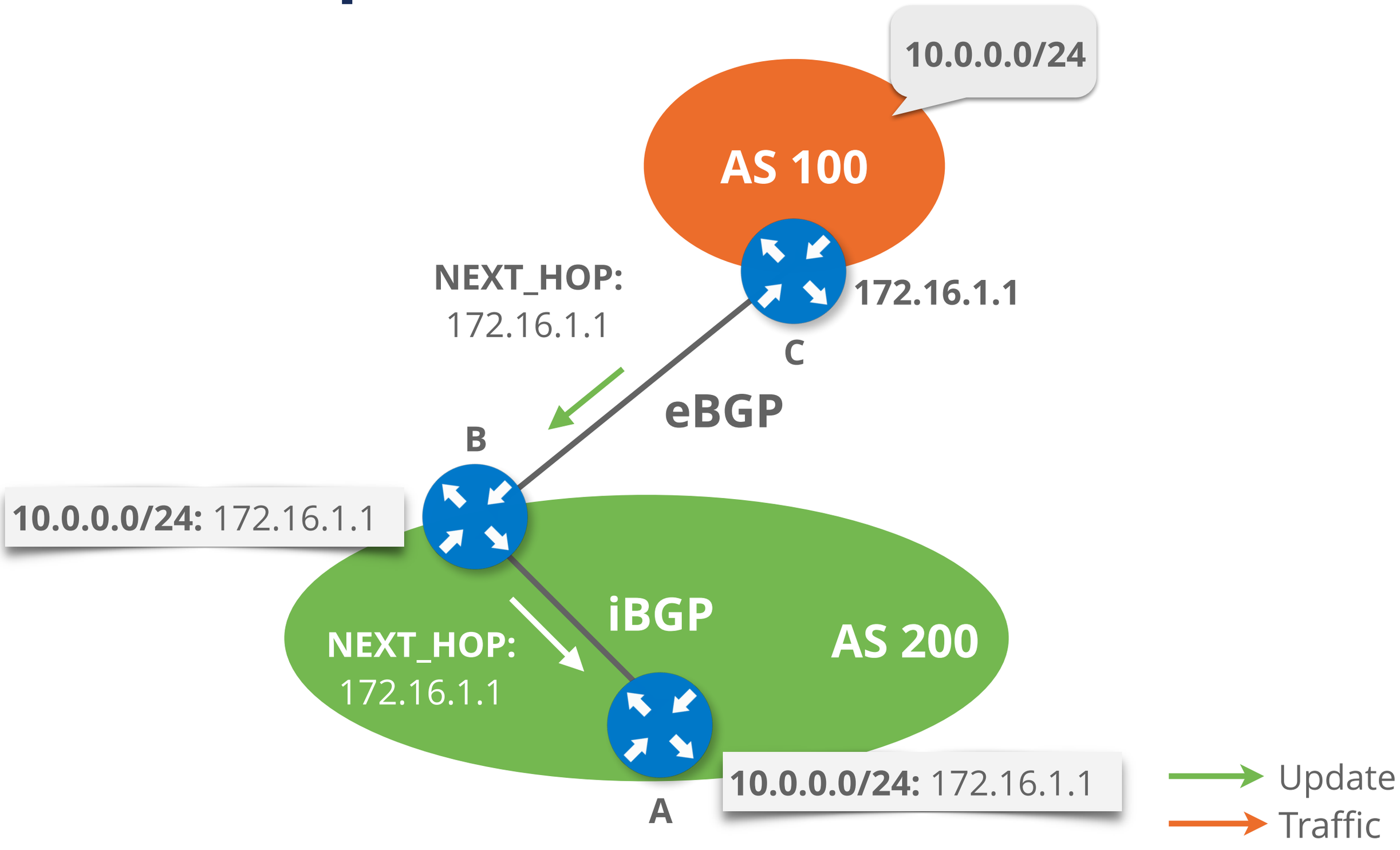
Weight



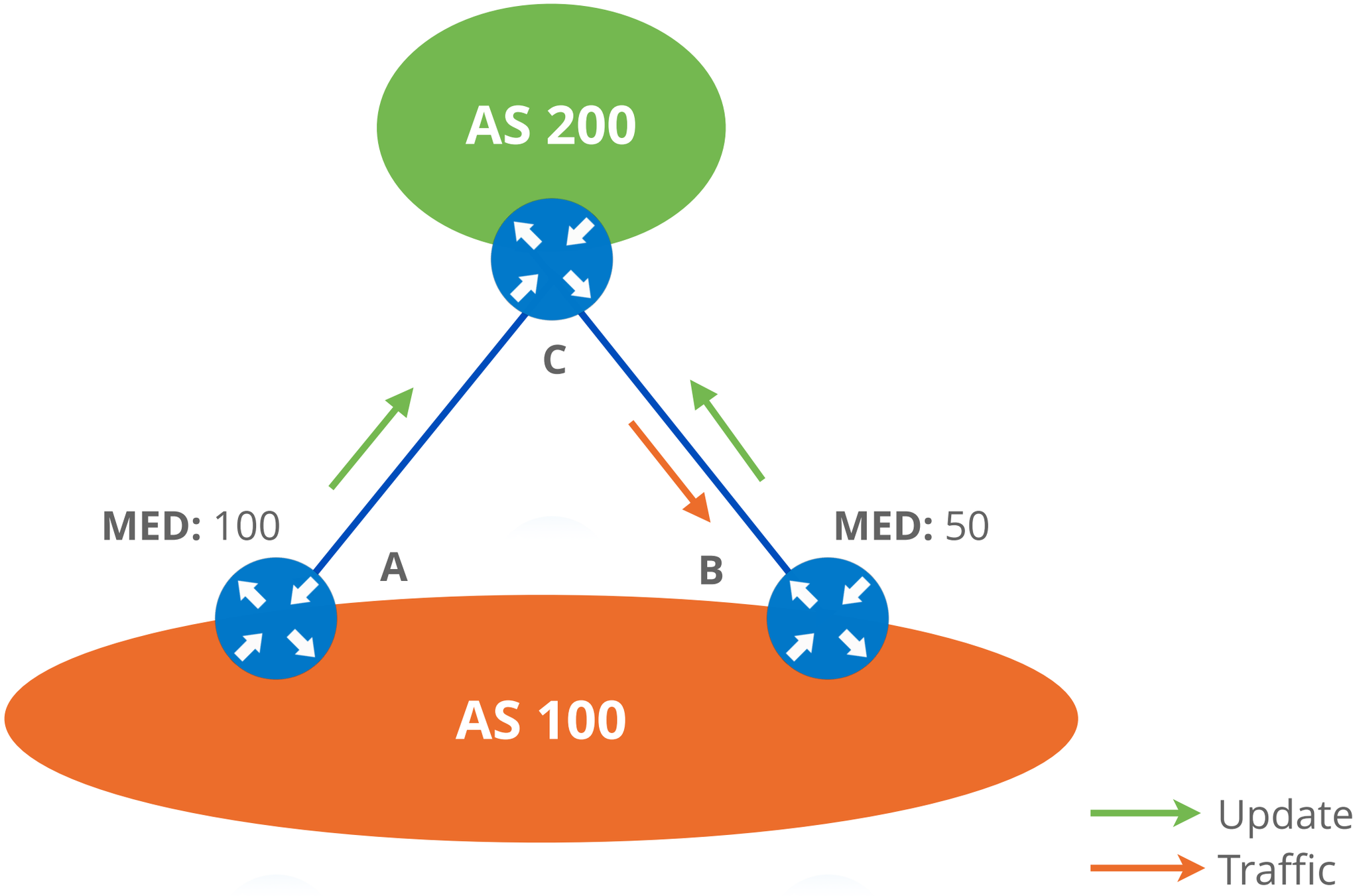
Local Preference



Next Hop


10.0.0.0/24
AS 100
NEXT_HOP:
172.16.1.1
172.16.1.1
C
eBGP
B
10.0.0.0/24: 172.16.1.1
iBGP
AS 200
NEXT_HOP:
172.16.1.1
10.0.0.0/24: 172.16.1.1
A
Update
Traffic

MED



Origin

- Mandatory
  - **IGP** – generated by BGP network statement
  - **EGP** – generated by EGP
  - **Incomplete** – redistributed from another routing protocol

Communities

- Community is a tagging technique to mark a set of routes
  - 32-bit integer which is attached to a BGP route as an optional transitive attribute

  as-number:community-value

  - Multiple communities can be attached to one route
  - Well-known (hard-coded) communities exist: www.iana.org/assignments/bgp-well-known-communities

Communities categories

- Attach information
  - Providing information (Informational tags) to other Autonomous Systems
  - Requesting actions (Action tags) by other Autonomous Systems
- Informational Communities
  - On the edge and within Autonomous System network
  - BGP Attribute manipulation
  - Import and Export control (do or do not announce the prefix to network X)

Large Communities

- A simple approach continuing along the standard communities
  - 96-bit integer which is attached to a BGP route as an optional transitive attribute

  as-number:function:parameter

  - Larger fields, more fields, and a clean namespace separation
  - 32-bit ASN clean solution
  - Canonical representation is $Me:$Action:$You
  - **Ex:***70000:4:80000* , AS70000 requests AS80000 to do 4 times prepend of that prefix while advertising it to its peers

Questions

Traffic Engineering

Section 4

Why do Traffic Engineering?

- Manage your capacity
- Ensure service quality
- Manage service cost
- Recover from failures

Intra-domain Traffic Engineering

- **You** control your network:
  - You know how reliable it is
  - You know the cost of all the paths

Inter-domain Traffic Engineering

- You **DO NOT** control the network
  - BGP has no metrics, capacity or cost
  - High volume of traffic and number of routes combined with the simplicity of the protocol imposes some limitations
- Large volume of data passes through a small number of ASNs
  - Tier 1,2,3 operators
  - IXPs

The BGP Decision Algorithm

- BGP router learns several paths to the destination
- For each prefix, BGP selects the best route
  - The decision process is based on attributes
  - The best path will get propagated to the neighbours

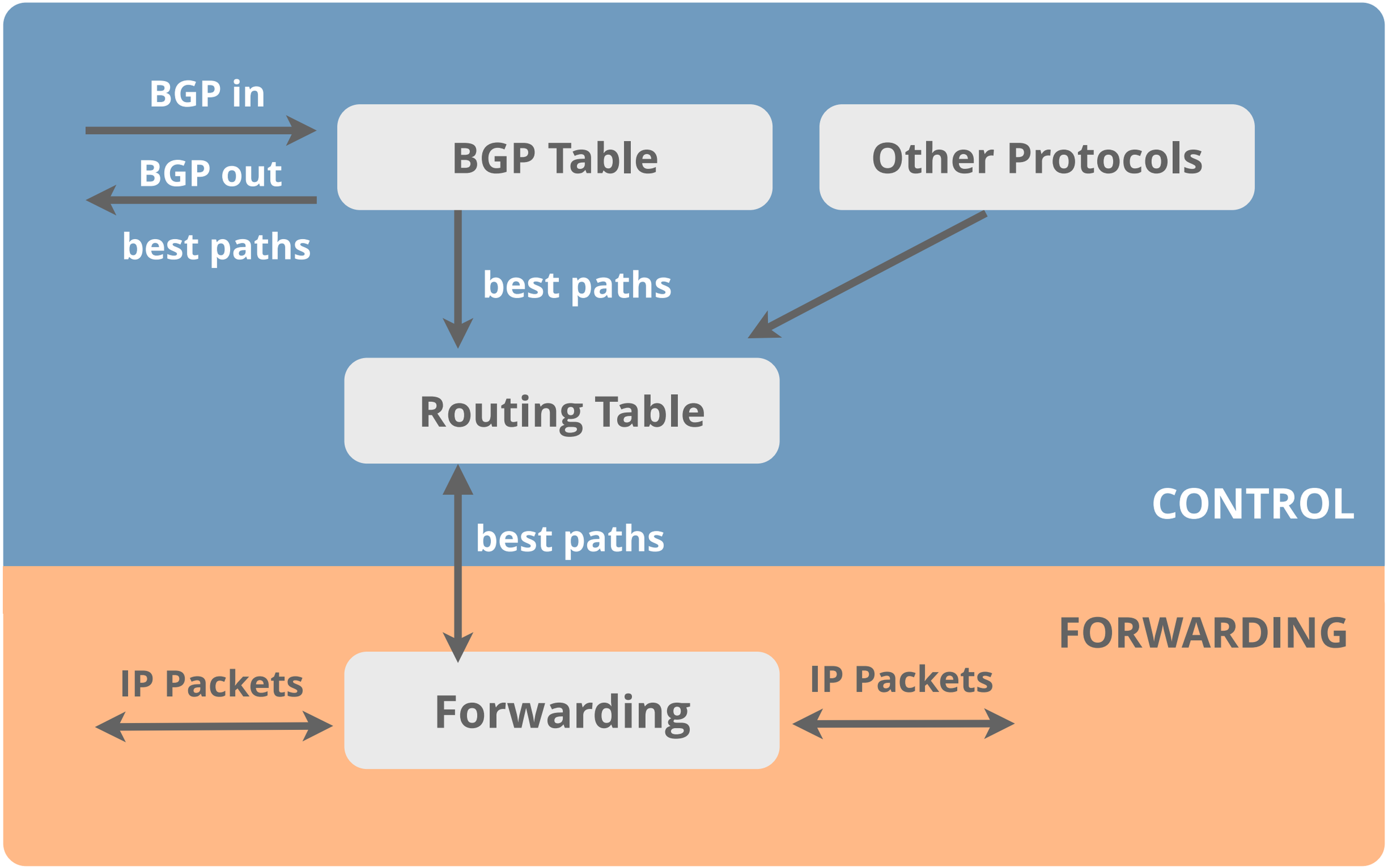
BGP Path Selection

- Prefer path with the highest **WEIGHT**
- Highest **LOCAL_PREF**
- Prefer the path that was locally originated
- Shortest **AS_PATH**
- Lowest **IGP ORIGIN**
- Lowest **MED**
- Prefer **eBGP** over **iBGP**

BGP Path Selection

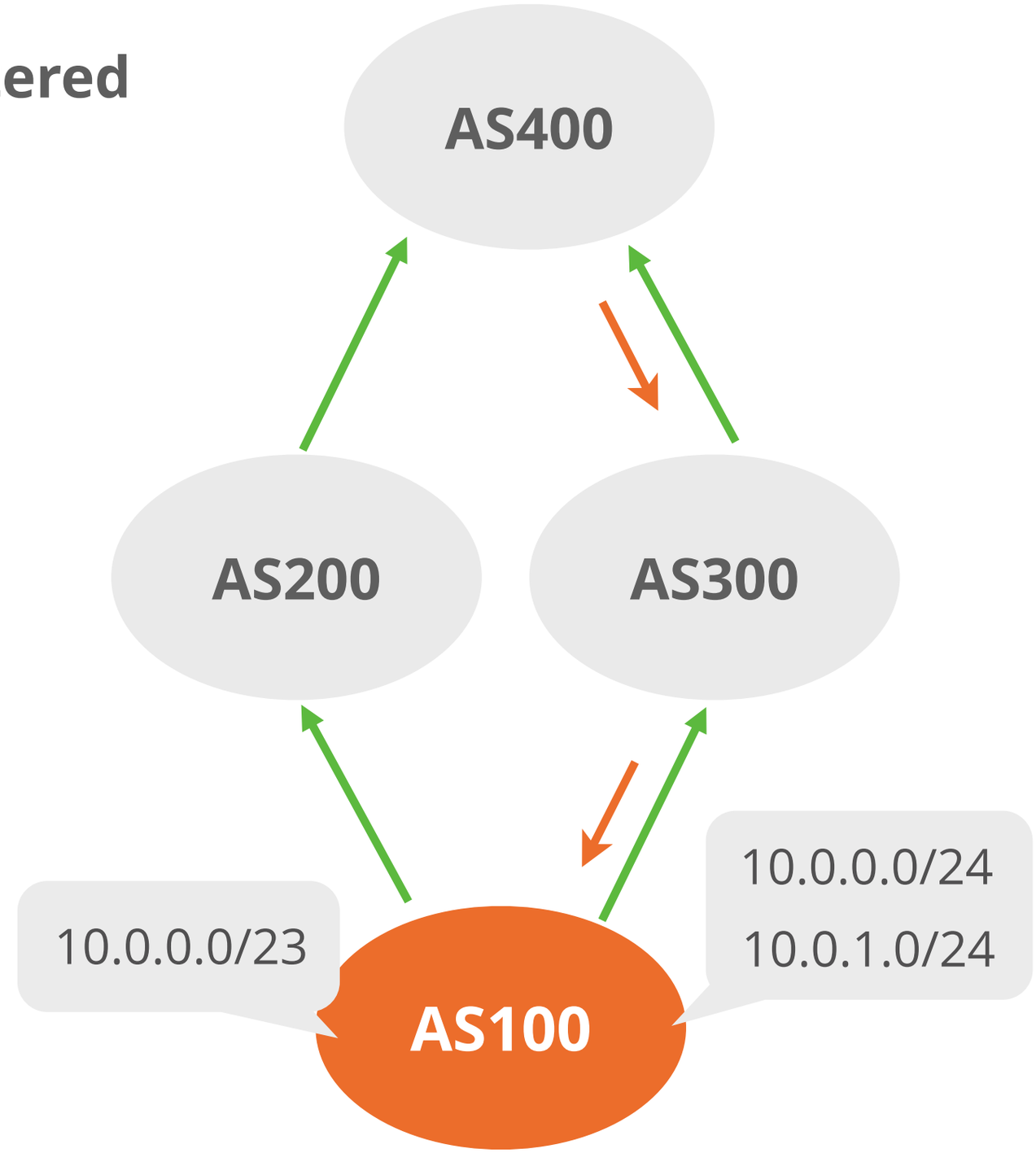
- Lowest IGP metric (shortest IGP path to BGP next hop)
- **Oldest** received path
- Lowest **router ID**
- Path from **lowest** neighbour address

RIB and FIB


BGP in
BGP out
best paths
BGP Table
Other Protocols
best paths
Routing Table
best paths
CONTROL
FORWARDING
IP Packets
Forwarding
IP Packets

More specific announcements

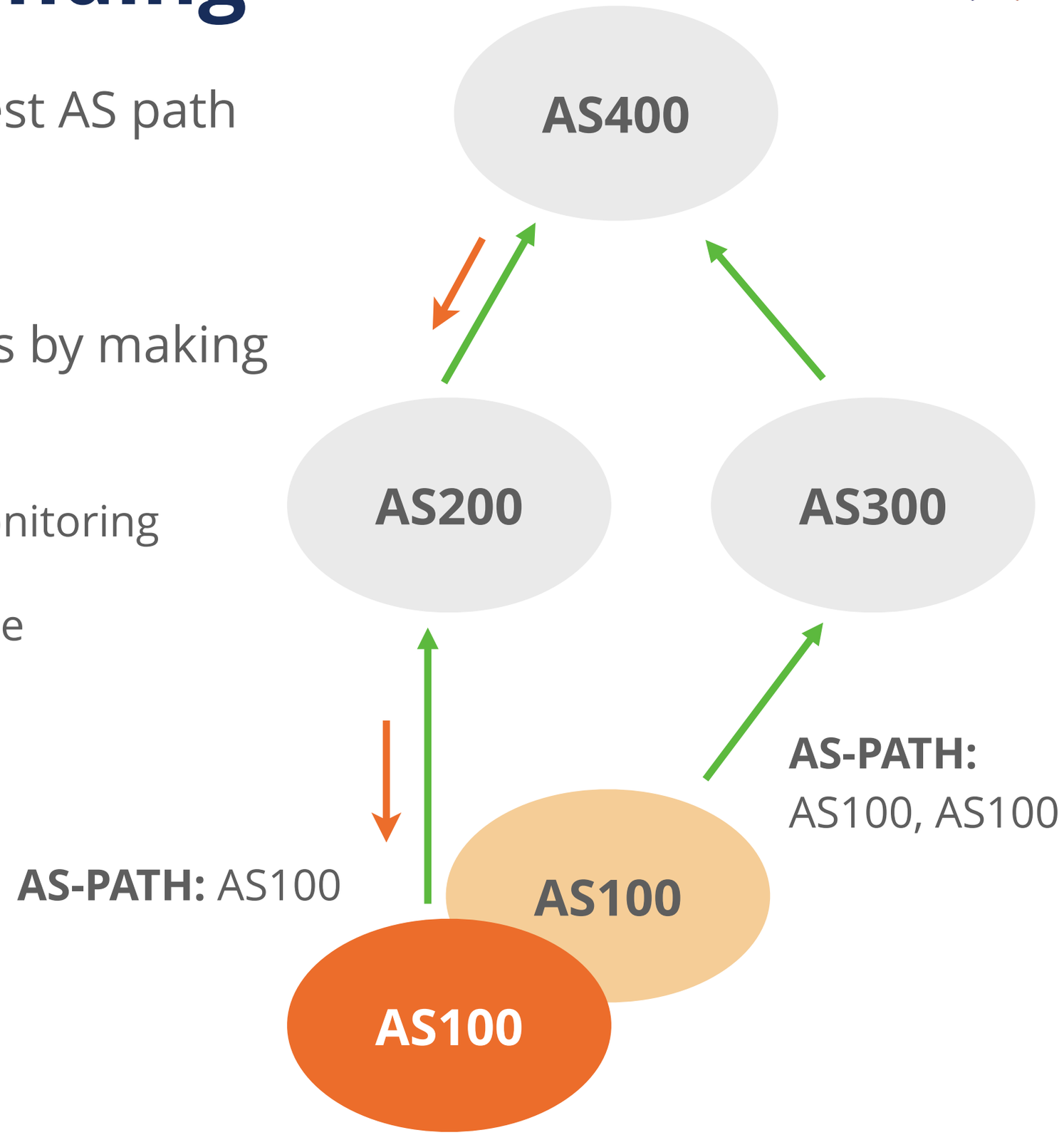
- Considered rude and often **filtered**
- Can be an effective tool
  - Might be used to announce a regional prefix
  - But should **never** be announced globally



AS Path Prepending

- BGP prefers the shortest AS path to get to a destination

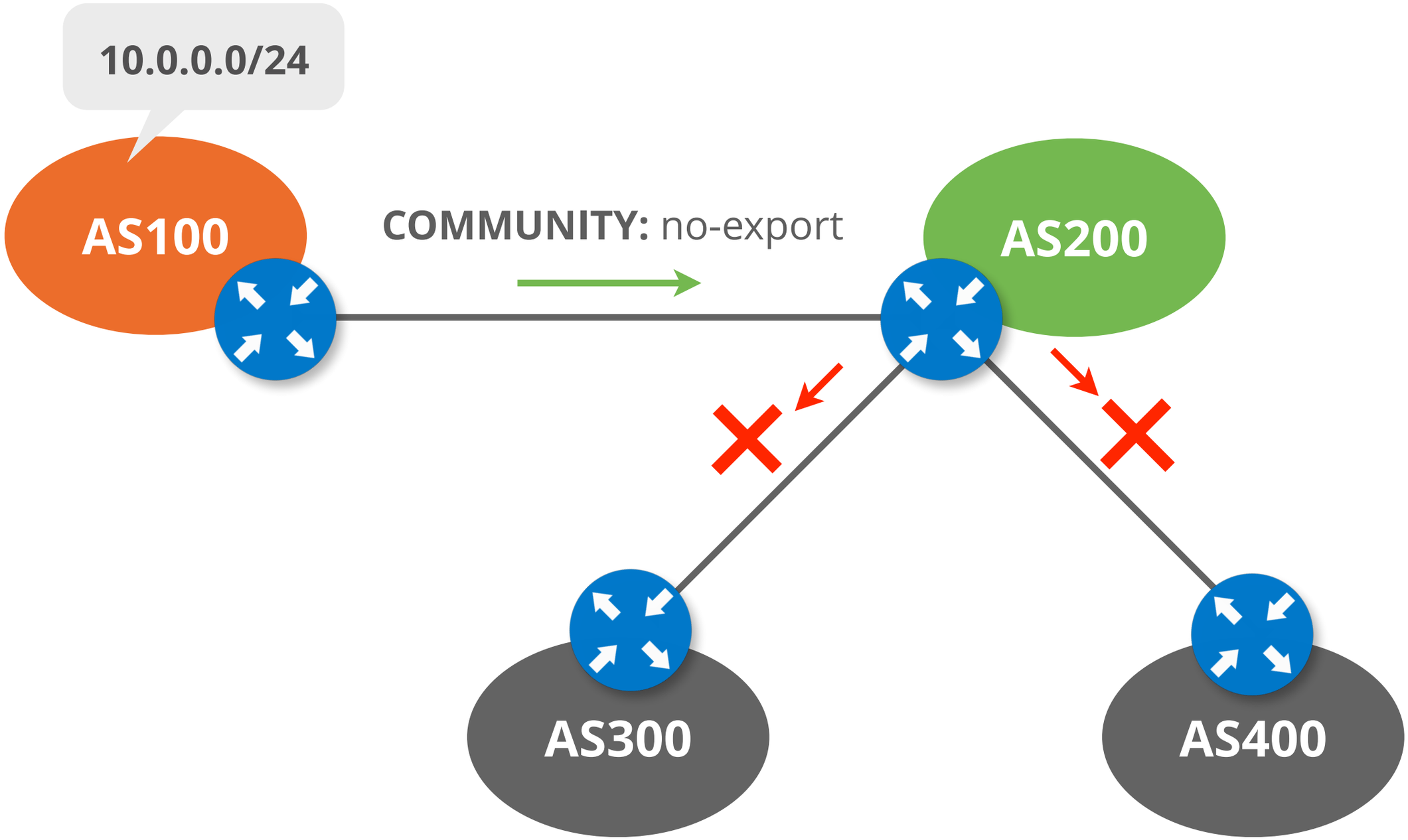
- We can manipulate this by making the **AS path** longer
  - Requires continuous monitoring
  - Very hard to load balance



Communities Usage

- Assign prefixes to pre-defined groups
  - Local significance only
- Control how prefix is advertised by peer
  - Control your neighbours **LOCAL_PREF** for the specific prefix
  - Signal neighbour to prepend multiple ASNs to **AS_PATH**
  - Blackhole all traffic to specific prefix

Community Example



Example Use of Communities

- Design your community system with character positions
- Community **1234:5678** (customer information)
  - Field #1, Value 5 (Type of Relationship)
  - Field #2, Value 6 (Continent Code)
  - Field #3, Value 7 (Region Code)
  - Field #4, Value 8 (POP Code)

Well-Known Communities

- **65535:65281** - no-export
  - do not advertise to any eBGP peers
- **65535:65282** - no-advertise
  - do not advertise to any BGP peer
- **65535:65283** - no-export-subconfed
  - do not advertise outside local AS (confederations)
- **65535:65284** - no-peer
  - do not advertise to bi-lateral peers *(RFC3765)*

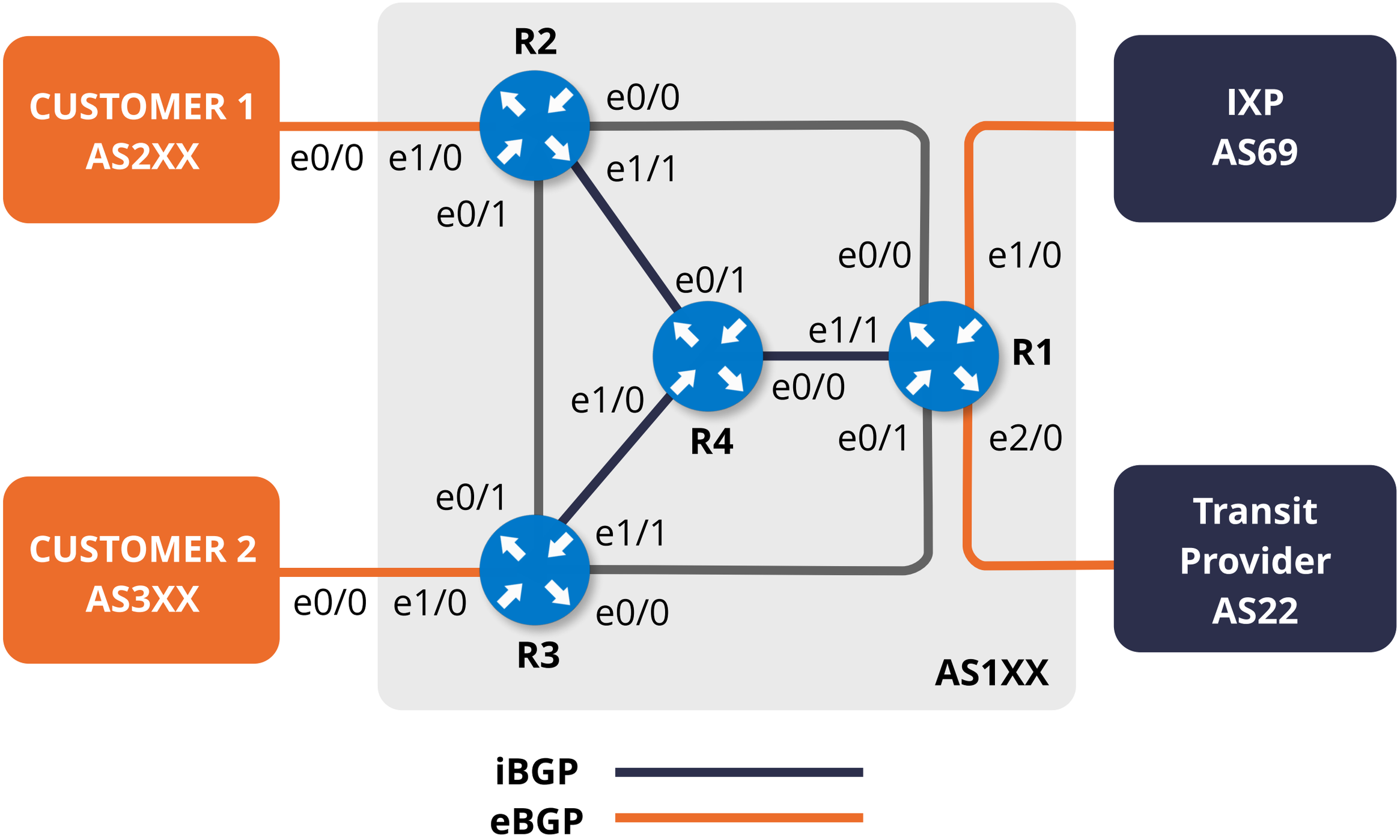
BGP Multipath

- Enables load balancing between “equal” paths
  - All attributes must be the same
- Multiple entries for the same destination appear in the routing table

Using Attributes

Activity

Network Diagram



Assignment

- Prefer routes received from Internet Exchange
  - Use local-preference
  - Use AS path prepending
- Data you will need
  - The IP address of IXP BGP routers
  - The AS number of IXP
  - Routing policy

Preparation (on R1)

- Examine routing tables

```
# show ip route
```

```
# show ip bgp
```

```
# show ip bgp 10.66.0.1
```

- Which routes are you using to reach other Internet Exchange members?

Outgoing Traffic (on R1)

- Create a route map

```
(config)# route-map LOCAL-PREF-150 permit 5
(config-route-map)# set local-preference 150
```

- Apply map to incoming routes from IXP

```
(config)# router bgp 1XX
(config-router)# neighbor 172.16.0.66 route-map LOCAL-PREF-150 in
(config-router)# neighbor 172.16.0.99 route-map LOCAL-PREF-150 in
```

- Session must be cleared, for the new policy

```
# clear ip bgp 172.16.0.66 in
# clear ip bgp 172.16.0.99 in
```

Incoming Traffic (on R1)

- Create a route map

```
(config)# route-map PREPEND permit 5
(config-route-map)# match ip address prefix-list TRANS-OUT-V4
(config-route-map)# set as-path prepend 1XX 1XX 1XX
```

- Add route map outgoing routes to Transit router

```
(config)# router bgp 1XX
(config-router)# neighbor 10.132.X.1 route-map PREPEND out
```

- Session must be cleared, for the new policy

```
# clear ip bgp 10.132.X.1 out
```

Verification (on R1)

- Examine routing tables

```
# show ip route
```

```
# show ip bgp
```

```
# show ip bgp 10.66.0.1
```

- Make sure that routes received from Internet Exchange are preferred

- Ask your colleague to show route to your network

Questions

iBGP Scalability

Section 5

Networks Grow

- How to scale iBGP mesh beyond a few peers?
- How to implement new policy without causing flaps and route churning?
- How to reduce the overhead on the routers?
- How to keep the network stable, scalable, as well as simple?

Scaling Techniques

- Current best practice:
  - Route Refresh capability
  - Templating
  - Route Reflectors
  - Confederations
- Deprecated practice:
  - Soft Reconfiguration
  - Route Flap Damping

BGP Configuration change

- Historically, hard BGP peer reset required after every policy change
  - Brings down BGP peering
  - Consumes CPU
  - Disrupts connectivity for all networks
  - Impact equivalent to a router reboot

Route Refresh

- Non-disruptive policy changes
- No configuration needed
  - Automatically negotiated at peer establishment
  - Requires support “route refresh capability” on all routers
- No additional resources used

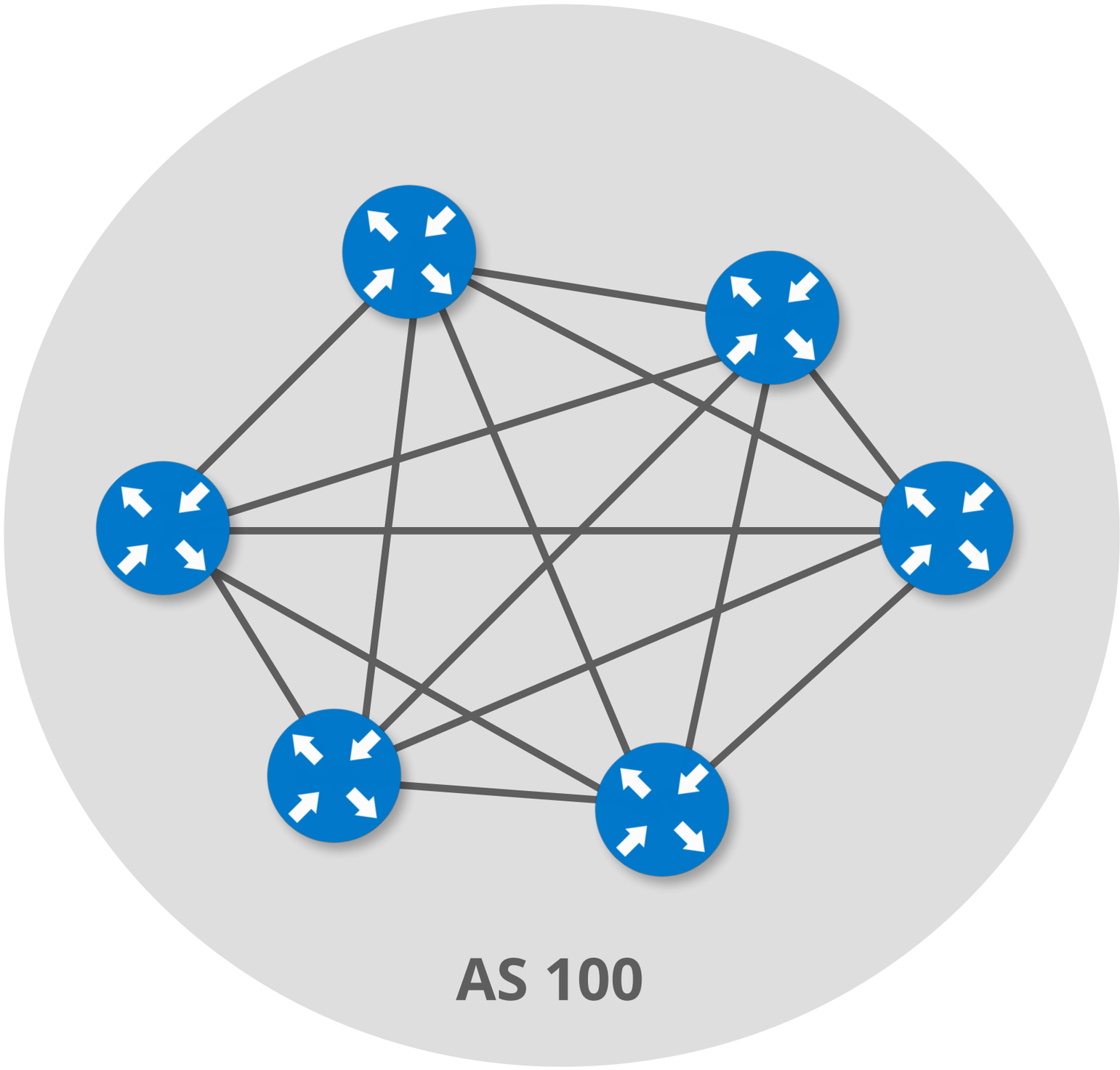
Templating

- Easier and more readable configuration
  - Groups peers with the same outbound policy
  - Updates generated once per template (Lower router CPU)
  - iBGP mesh builds more quickly
  - Members can have different inbound policy
- Can be used for eBGP neighbours
  - Consider using templates for peering at IXPs

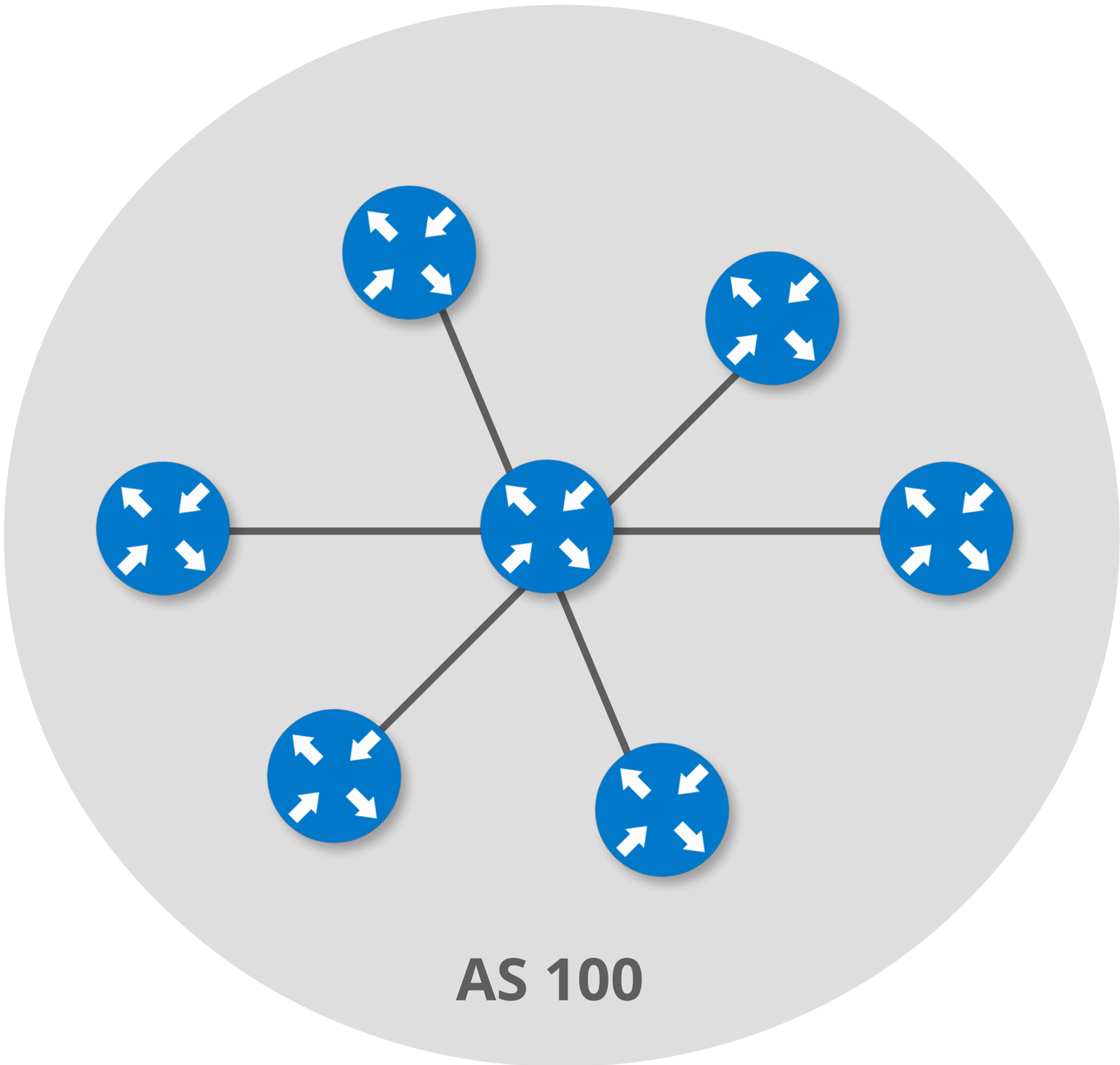
Network growth

- iBGP needs a full mesh
  - Slow to build
  - iBGP neighbours receive the same update
  - Router CPU wasted on repeated calculations
  - Growing number of sessions

iBGP without Route Reflector



iBGP with Route Reflector



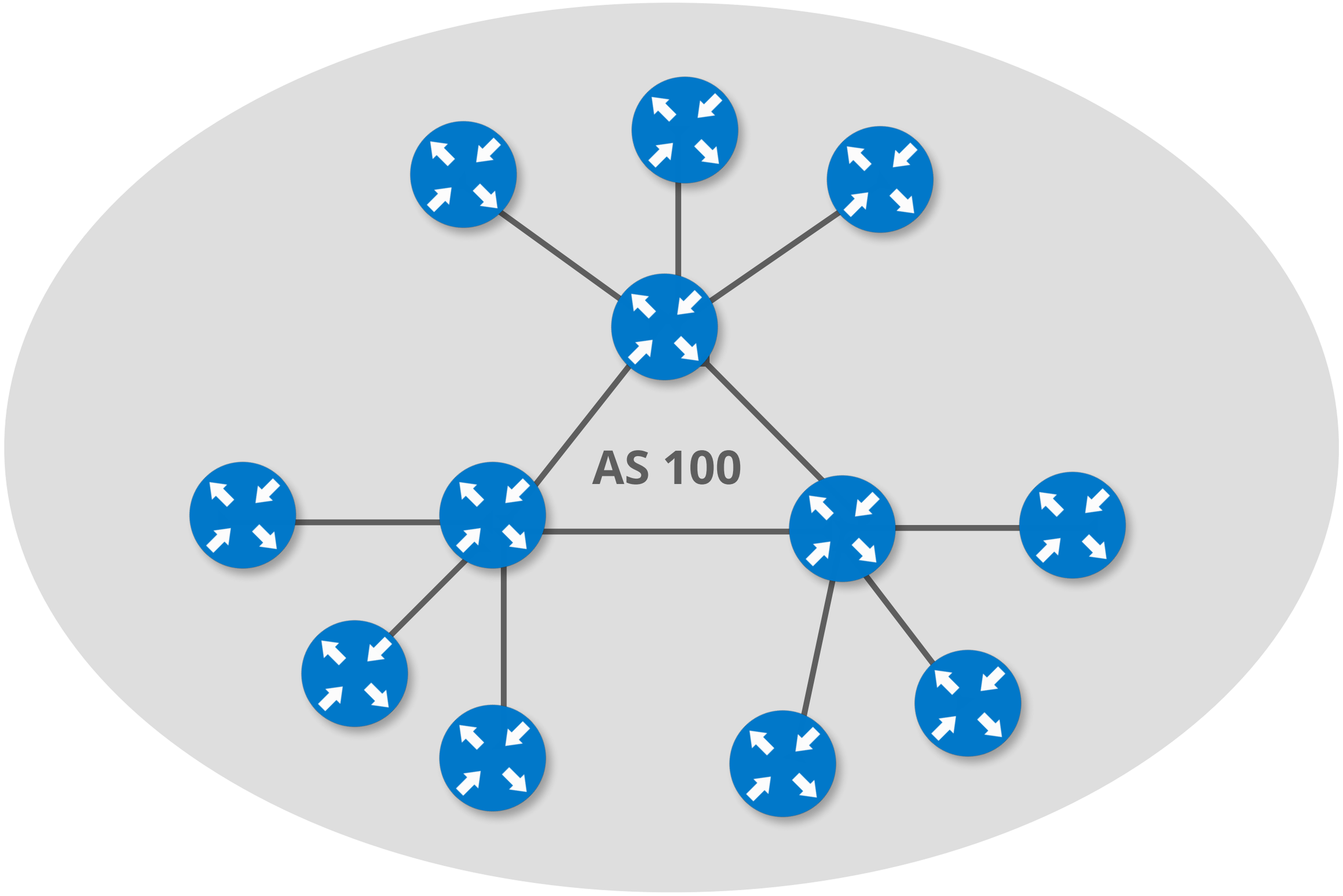
What is a route reflector?

- Each router in the ASN establishes one iBGP session with the route reflector
  - Solves iBGP mesh problem
  - Reduce number of BGP Sessions

Route Reflector Operations

- Receives path from clients
- Selects best path
  - If best path is from client, reflect to other clients and non- clients
  - If best path is from non-client, reflect to clients only non-meshed clients

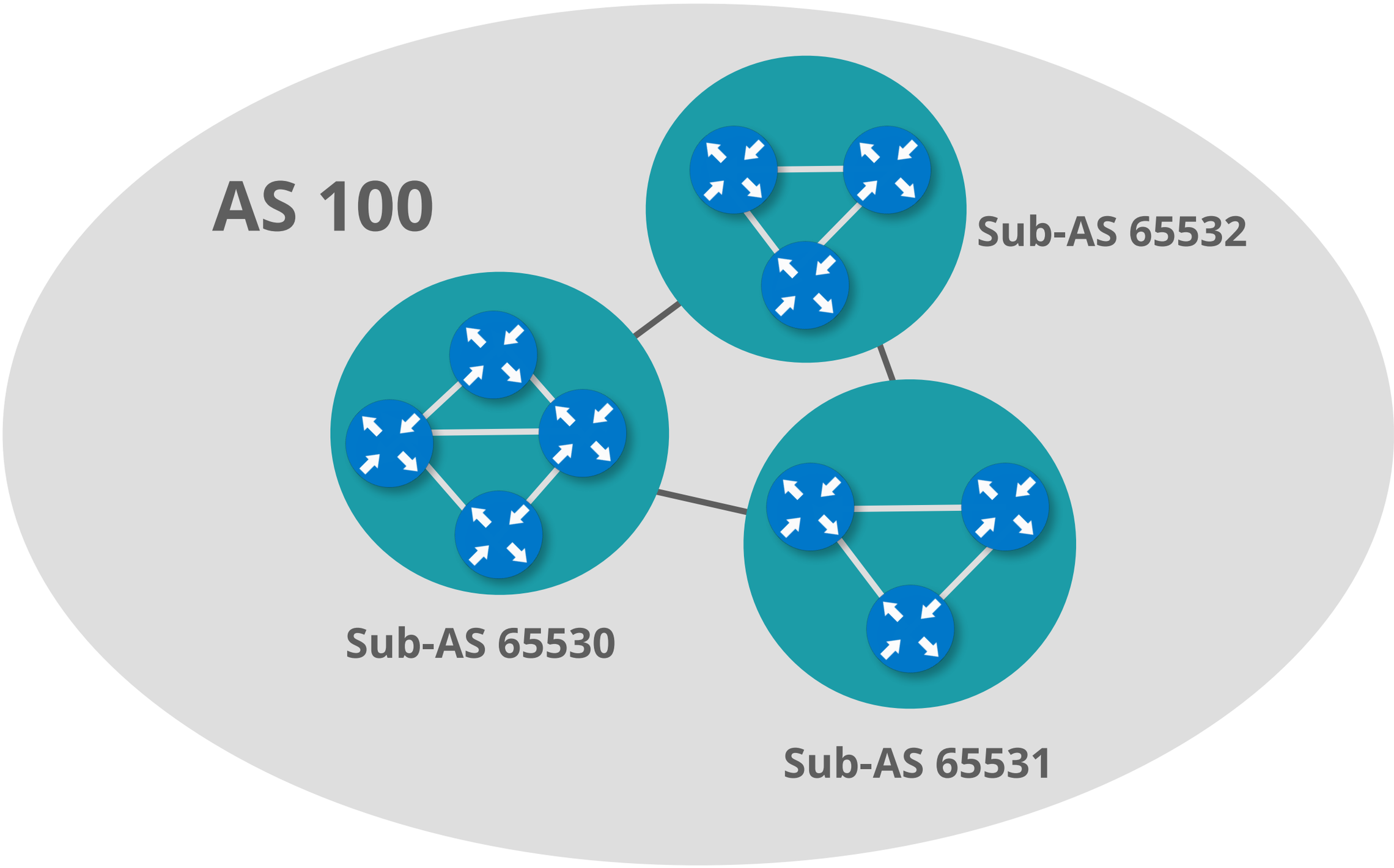
Route Reflectors Topology



Route Reflector Best Practice

- Divide the backbone into multiple clusters
  - One route reflector and a few clients per cluster
- Route reflectors must be fully meshed between each other

Confederations



Confederations

- Divide the AS into sub-ASns
  - Use private ASNs
- eBGP between sub-ASns
- iBGP speakers in sub-ASns are fully meshed

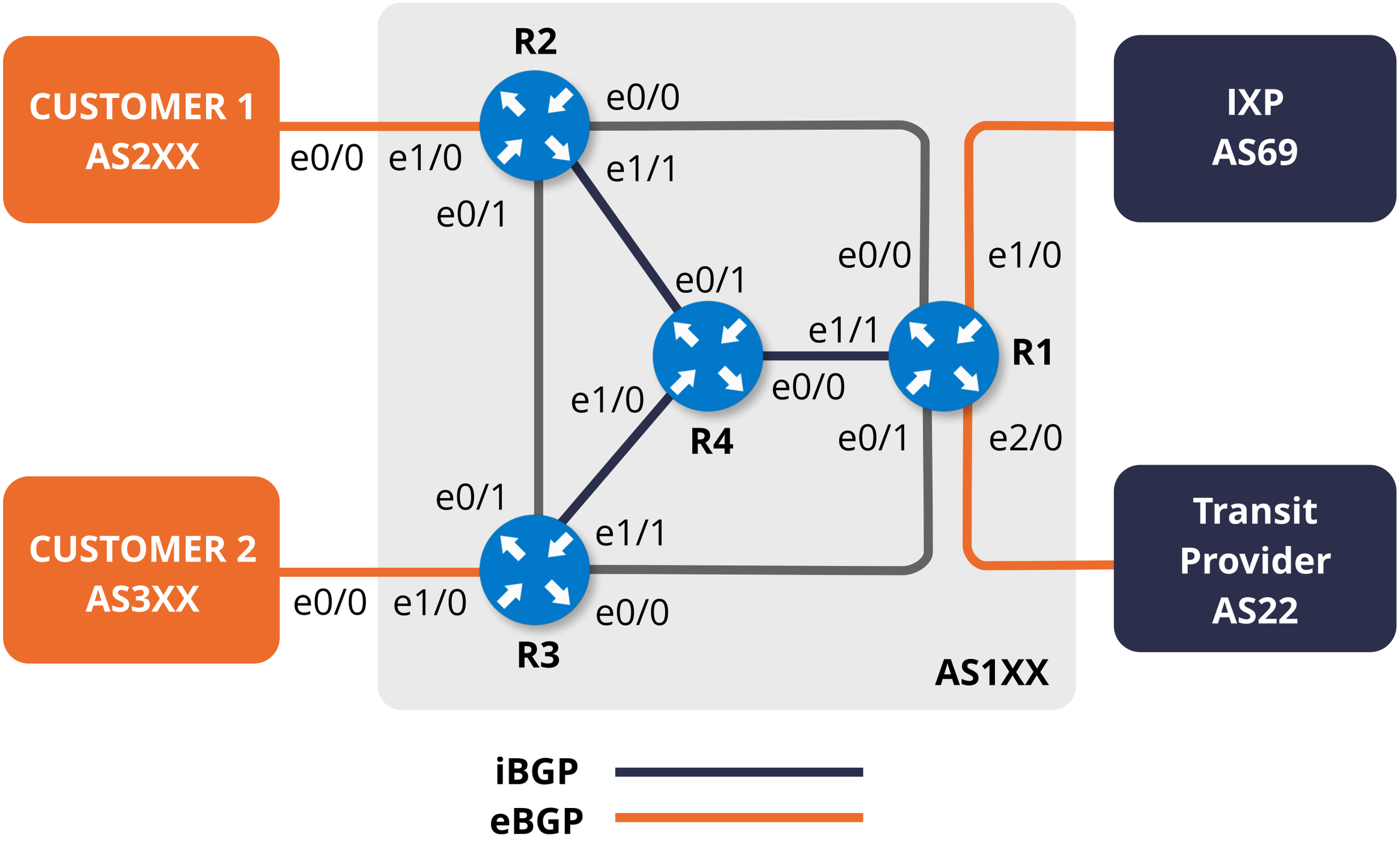
Confederations + Route Reflectors

- You can mix confederations and route reflectors
- Sub-ASns can have their own route reflector

Using a Route Reflector

Activity

Network Diagram



Assignment

- Simplify your internal BGP network by using Router 4 as a Route Reflector
- Data you will need
  - Your AS number
  - Your IP address space
  - Loopback address of the Route Reflector

Configure Route Reflector (on R4)

- Router 4 will reflect routes to other iBGP speakers - Route Reflector Clients

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor RR-GROUP peer-group
(config-router)# neighbor RR-GROUP remote-as 1XX
(config-router)# neighbor RR-GROUP update-source lo0
(config-router)# neighbor RR-GROUP route-reflector-client
(config-router)# neighbor 172.X.255.1 peer-group RR-GROUP
(config-router)# neighbor 172.X.255.2 peer-group RR-GROUP
(config-router)# neighbor 172.X.255.3 peer-group RR-GROUP
```

- Configuration simplified with peer-group

Add Route Reflector Clients

- Router 1, Router 2, Router3

```
(config)# router bgp 1XX
(config-router)# neighbor 172.X.255.4 remote-as 1XX
(config-router)# neighbor 172.X.255.4 next-hop-self
(config-router)# neighbor 172.X.255.4 update-source lo0
```

Remove iBGP Mesh

- Router 1

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.2
(config-router)# no neighbor 172.X.255.3
```

- Router 2

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.1
(config-router)# no neighbor 172.X.255.3
```

- Router 3

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.1
(config-router)# no neighbor 172.X.255.2
```

Verify

- Check sessions in summary

```
# show ip bgp neighbors | include BGP
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
```

- Verify reachability from customer

```
# ping 10.132.32.1
# ping <your colleague Customer 1 or 2 IP>
```

- Show logged events

```
# show logging
```

Questions

Multiprotocol BGP

Section 6

Multiprotocol BGP (MP-BGP)

- Extension to the BGP protocol
- Negotiated during sessions set up
  - Using the **BGP OPEN** message
  - When **CAPABILITIES** contain Multiprotocol Extensions

MP-BGP

- New BGP Capabilities in OPEN message:
  - Address Family Identifier (**AFI**)
  - Subsequent Address Family Identifier (**SAFI**)

MP-BGP Path Attributes

- Attribute contains one or more
  - Address Family Identifier (AFI) with SAFI
- Identifies the protocol information carried in the **Network Layer Reachability Info (NLRI)** field
- Next-hop address must be of the same family

Address Families

- Address Family Identifier (AFI)
  - Identifies Address Type
  - **AFI = 1 (IPv4)**
  - **AFI = 2 (IPv6)**
- Subsequent Address Family Identifier (SAFI)
  - Sub category for AFI Field
  - **Address Family Identifier (AFI)**
    - Sub-AFI = 1 (NLRI is used for unicast)
    - Sub-AFI = 2 (NLRI is used for multicast RPF check)
    - Sub-AFI = 3 (NLRI is used for both unicast and multicast RPF check)
    - Sub-AFI = 4 (label)
    - Sub-AFI = 128 (VPN)

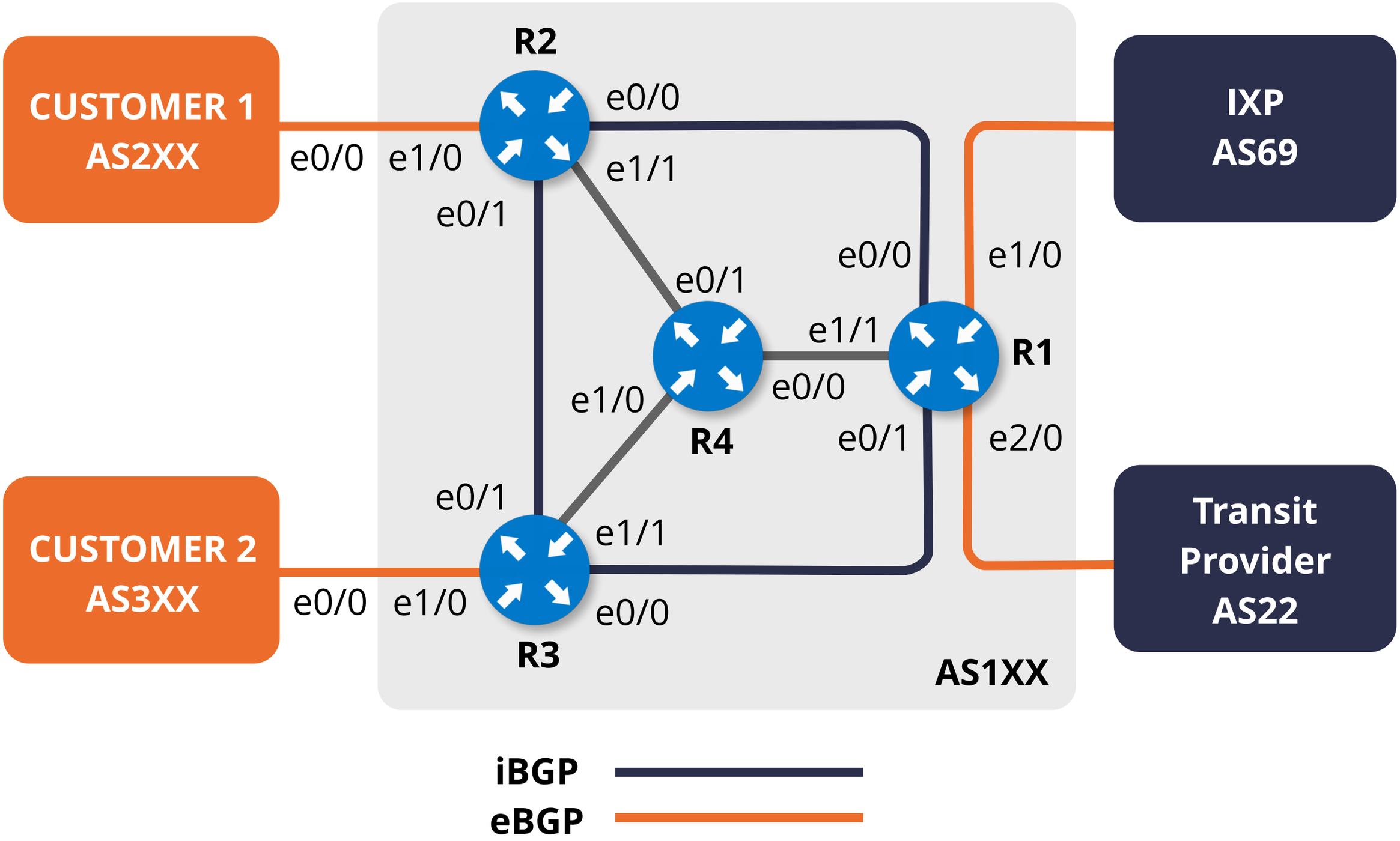
Multiprotocol BGP

Activity

Assignment

- Enable Multiprotocol BGP
- Using IPv6
  - Connect your network to Transit Provider
  - Connect you network to Internet Exchange
- Data you will need
  - Your AS number
  - Your IPv6 address space
  - The AS number of your neighbours
  - The IPv6 address of your neighbours BGP routers

Network Diagram



Preparation (on R1)

- Insert static Null route
  - Before BGP advertises a network, it checks for an exact match in the router's routing table

```
(config)# ipv6 route 2001:ffXX::/32 null0 250
```

Enabling Multiprotocol BGP (on R1)

- Examine your router BGP configuration

```
# show running-config | section router bgp
```

- Enable MP-BGP

```
(config)# router bgp 1XX
(config-router)# no bgp default ipv4-unicast
```

- Examine your router BGP configuration again

```
# show running-config | section router bgp
```

Interface IPv6 Settings (on R1)

- Your network is already dual stacked
  - IGP and Loopbacks
- Configure IPv6 on your IXP interface

```
(config)# interface Ethernet1/0
(config-if)# ipv6 address 2001:ff69::XX/64
(config-if)# no ipv6 redirects
(config-if)# ipv6 nd ra suppress all
```

- Configure IPv6 on your Transit interface

```
(config)# interface Ethernet2/0
(config-if)# ipv6 address 2001:ff32:0:XX::b/64
(config-if)# no ipv6 redirects
(config-if)# ipv6 nd ra suppress all
```

Create a filter (on R1)

- Create a filter to announce your IPv6 prefix to both Transit provider and IXP

```
(config)# ipv6 prefix-list TRANS-OUT-V6 seq 5 permit 2001:ffXX::/32
(config)# ipv6 prefix-list IXP-OUT-V6 seq 5 permit 2001:ffXX::/32
```

Configure Transit Session (on R1)

- Configure BGP session with AS22

```
(config)# router bgp 1XX
(config-router)# neighbor 2001:ff32:0:XX::a remote-as 22
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff32:0:XX::a activate
(config-router-af)# neighbor 2001:ff32:0:XX::a prefix-list TRANS-OUT-V6
out
```

- Advertise your IPv6 prefix

```
(config-router-af)# network 2001:ffXX::/32
```

Configure IXP Sessions (on R1)

- Configure BGP sessions with AS69

```
(config)# router bgp 1XX
(config-router)# neighbor 2001:ff69::66 remote-as 69
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff69::66 activate
(config-router-af)# neighbor 2001:ff69::66 prefix-list IXP-OUT-V6 out
(config-router)# exit
(config-router)# neighbor 2001:ff69::99 remote-as 69
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff69::99 activate
(config-router-af)# neighbor 2001:ff69::99 prefix-list IXP-OUT-V6 out
```

Verify

- Check sessions summary

```
# show bgp ipv6 unicast
```

- Check BGP and routing table

```
# show bgp ipv6
# show ipv6 route
```

- Verify reachability

```
# ping 2001:ff32::a
# ping <your colleague R1 IPv6>
```

- Show logged events

```
# show logging
```

Questions

Overview

Day 1

- Introduction to BGP
- Running BGP
- BGP Attributes
- Traffic Engineering
- iBGP Scalability
- Multiprotocol BGP

Day 2

- BGP & Routing Security
- Internet Routing Registry
- Filtering
- RPKI
- What else?
- BGP Tips & Tricks

BGP & Routing Security

Section 7

BGP has some challenges …

- BGP has some challenges from the perspective of routing security
  - It is only based on trust, no built-in security
  - No verification of the correctness of prefixes or AS paths
- These challenges are discussed in RFC#4272, “BGP Security Vulnerabilities Analysis”

RFC#4272,“BGP Security Vulnerabilities Analysis”

Vulnerabilities of BGP

- Based on RFC, BGP has three fundamental vulnerabilities:

1. No internal mechanism to protect the integrity and source authenticity of BGP messages

2. No mechanism specified to validate the authority of an AS to announce NLRI

3. No mechanism to verify the authenticity of the attributes of a BGP update message

- These vulnerabilities can be exploited either **maliciously** or **accidentally**

Due to these vulnerabilities …

- Attacks can be conducted by exploiting TCP or BGP messages
- Any AS can announce any prefix
  - BGP prefix hijacks due to malicious activity / mis-origination
- Any AS can prepend any ASN to the AS path
  - Path hijacks, MITM
- Fake routing information could be propagated over the Internet and disrupt overall Internet behaviour

Some BGP incidents are malicious!

- Attackers may …
  - Hijack a BGP session and break peer to peer connection
  - Spoof the IP address of one of the BGP speaker’s peer routers
  - Initiate a DoS attack and exhaust victim’s resources
  - Manipulate BGP and reroute packets
  - Intercept and modify the traffic
  - Blackhole the entire network etc.

Sometimes, just human errors …

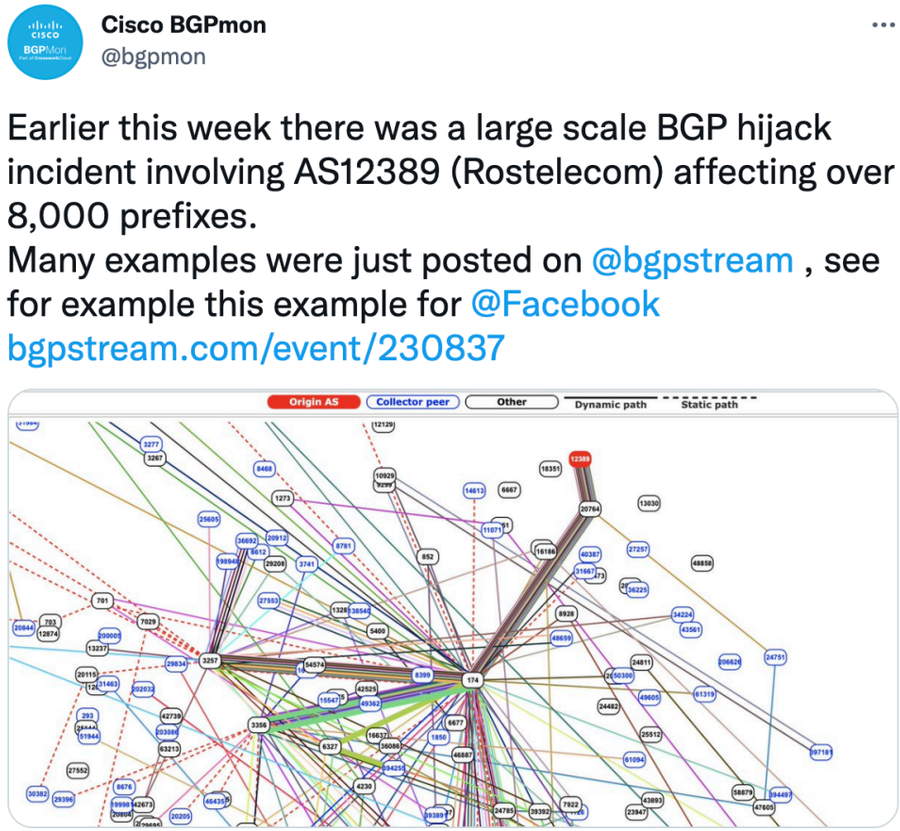
- Typo errors
  - Also known as “fat fingers”
  - May cause mis-origination
- Configuration errors
  - Faulty BGP filter configuration
  - AS path prepending mistake

April 2021: Vodafone Idea, AS55410

- What happened?
  - 34,000+ prefixes hijacked!
  - Impacted major network operators, cloud and CDN providers
  - 13 times more traffic than usual
- Why did it happen?
  - Caused by wrong advertisement
  - Lack of good filtering by upstream providers

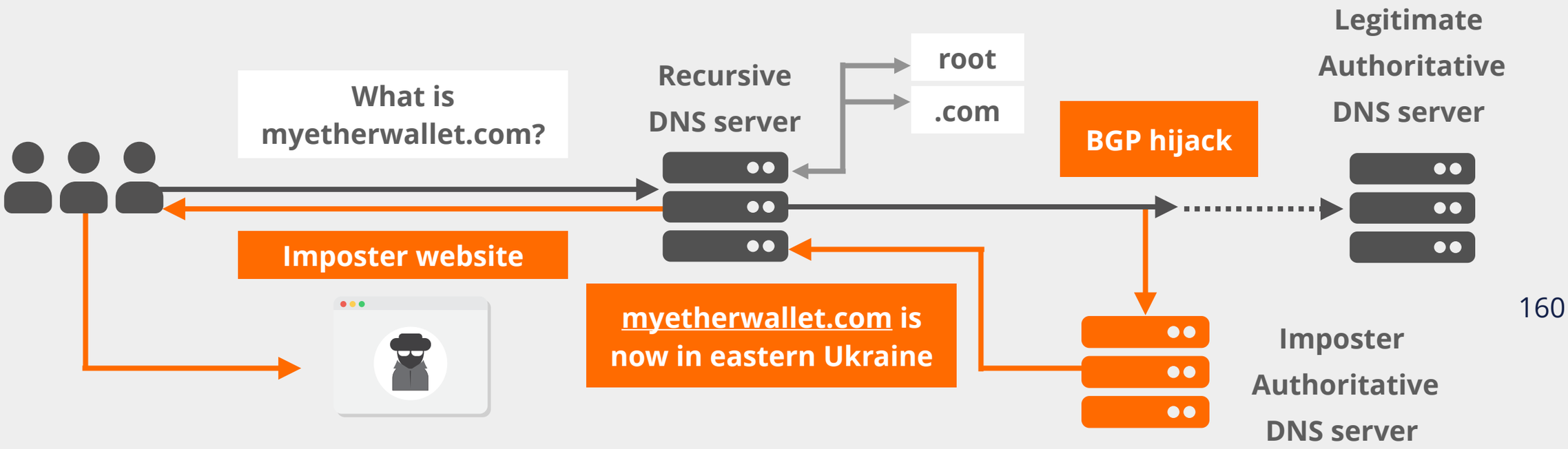
April 2020: Akamai, Amazon, …

- What happened?
  - 8k+ routes hijacked by Rostelecom (AS12389)
  - 200+ CDNs and cloud providers impacted
  - Not known how much data leaked
- Why did it happen?
  - Malicious activity
  - Lack of good filtering by upstream providers/peers



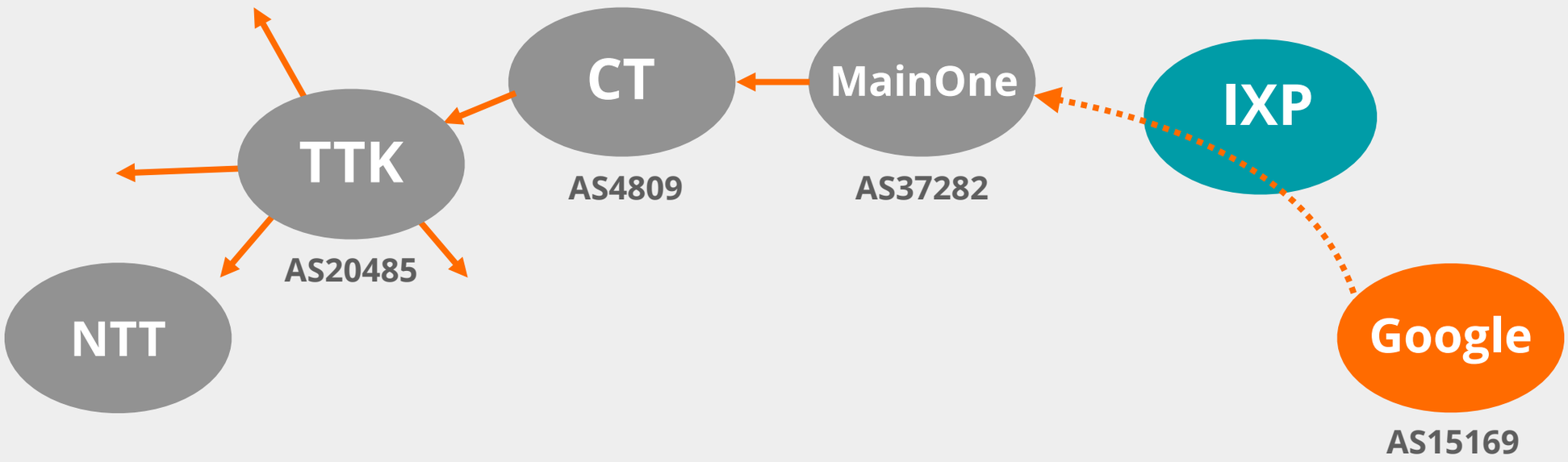
April 2018: Amazon-MyEtherWallet

- BGP hijack of Amazon DNS
- How did it happen?
- Why?
  - Attack to steal cryptocurrency



November 2018: Google Prefix Leak

- MainOne leaked Google routes to CT
- CT propagated them to several transit ISPs
- Google services (G Suite and Google Search) affected by the leak
- Due to misconfigured filters



For Secure Internet Routing …

- Do not be the cause!
  - Announce the right prefixes to the right peers
  - Have proper filters in place to eliminate route leaks
- Do not spread others' mistakes or attacks!
  - Validate the routing information you receive
- Do not be the victim!
  - Get all measures to protect your network

Measures to Prevent BGP Incidents

- RFC 7454 documents major countermeasures for BGP Operations and Security
- According to RFC, you should
  - Protect your BGP speaker (control plane and data plane filters)
  - Protect your BGP sessions (MD5, TCP-AO)
  - Implement Route Filtering
  - Register your routing information in IRR system
  - Implement RPKI and validate the origin of your BGP routes

Questions

Internet Routing Registry

Section 8

Introduction to the IRR

- Collection of **databases** for routing purposes
  - Declarations of BGP announcements
  - Declarations of BGP connected customers
- Many different databases exist
  - Mostly **mirroring** each other
  - RIPE, APNIC, RADB, JPIRR, Level3, NTTCom, others

Link to the IRR:

http://www.irr.net

RIPE Database Objects

IPs and ASNs

- **inetnum**
- **inet6num**
- **aut-num**

Routing

- **route**
- **route6**

Object Protection

- **mntner**

Benefits of the IRR

- Allows you to make informed routing decisions
  - “Shall I accept this route from my peering?”
- Some of the IRRs verify who is the **rightful holder** of the IPs and ASNs

Why Publish Your Routing Policy?

- Many transit providers and IXPs 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?

Autonomous Number Objects

Known as **aut-num** objects

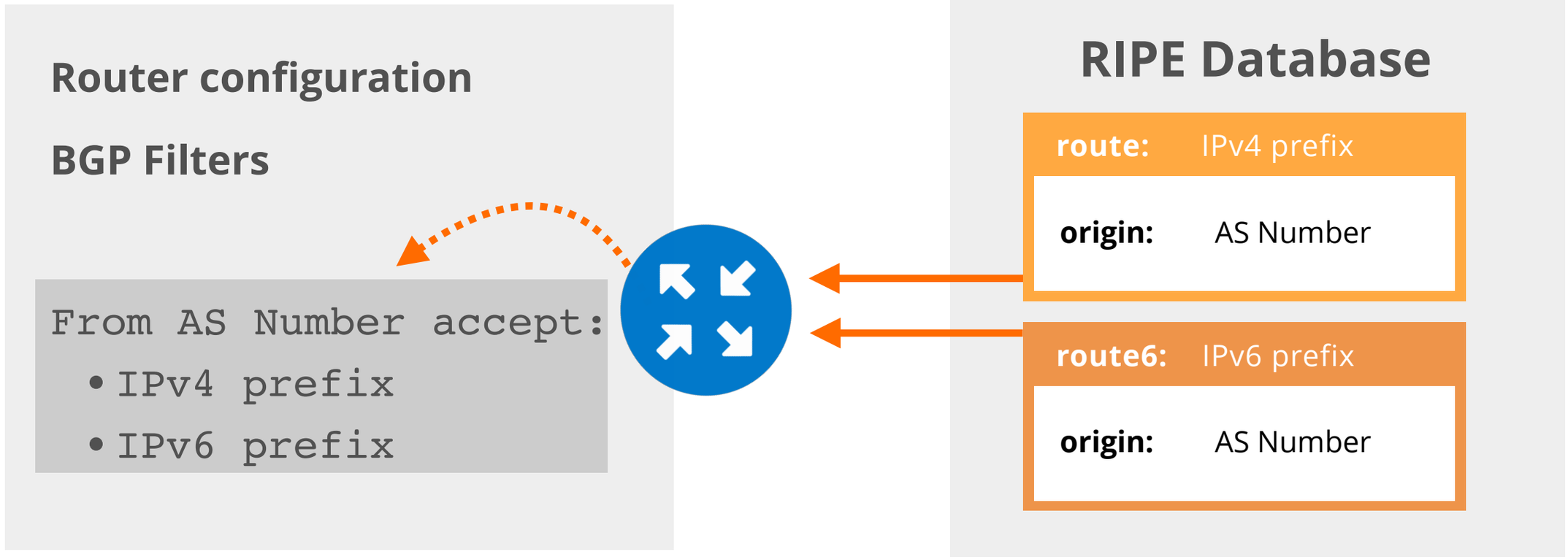
aut-num:	AS12345
as-name:	YOUR-AS-NAME
org:	ORG-EE2-RIPE
import:	**from AS1010 accept ANY**
export:	**to AS1010 announce AS12345**
import:	**from AS987 accept ANY**
export:	**to AS987 announce AS12345**
admin-c:	DV789-RIPE
tech-c:	JS123-RIPE
status:	ASSIGNED
mnt-by:	RIPE-NCC-END-MNT
mnt-by:	DEFAULT-LIR-MNT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

Registers **who** holds an AS Number

Registers the routing policy for that AS

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



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:	ANTOHER-MNT

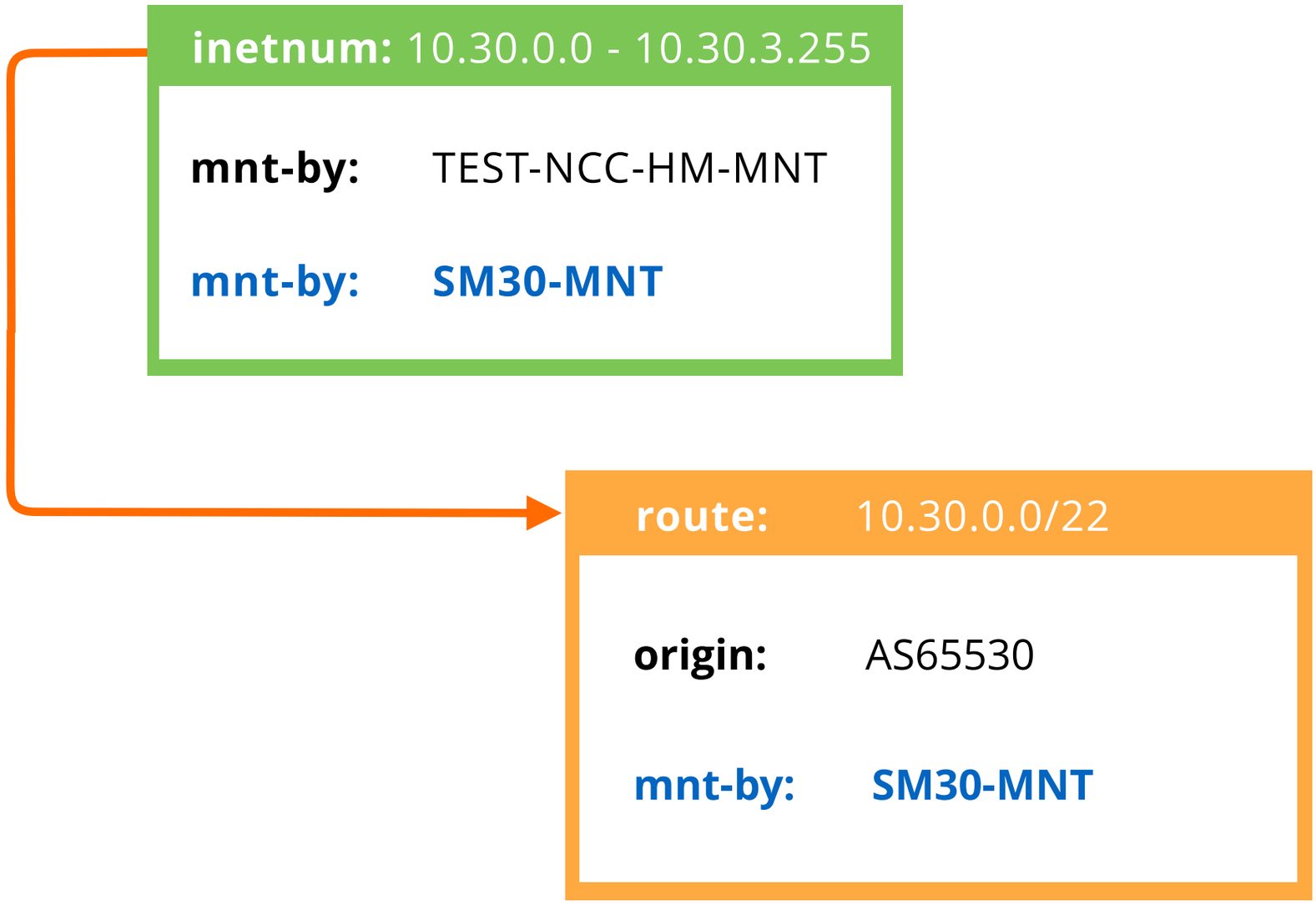
2

route(6)

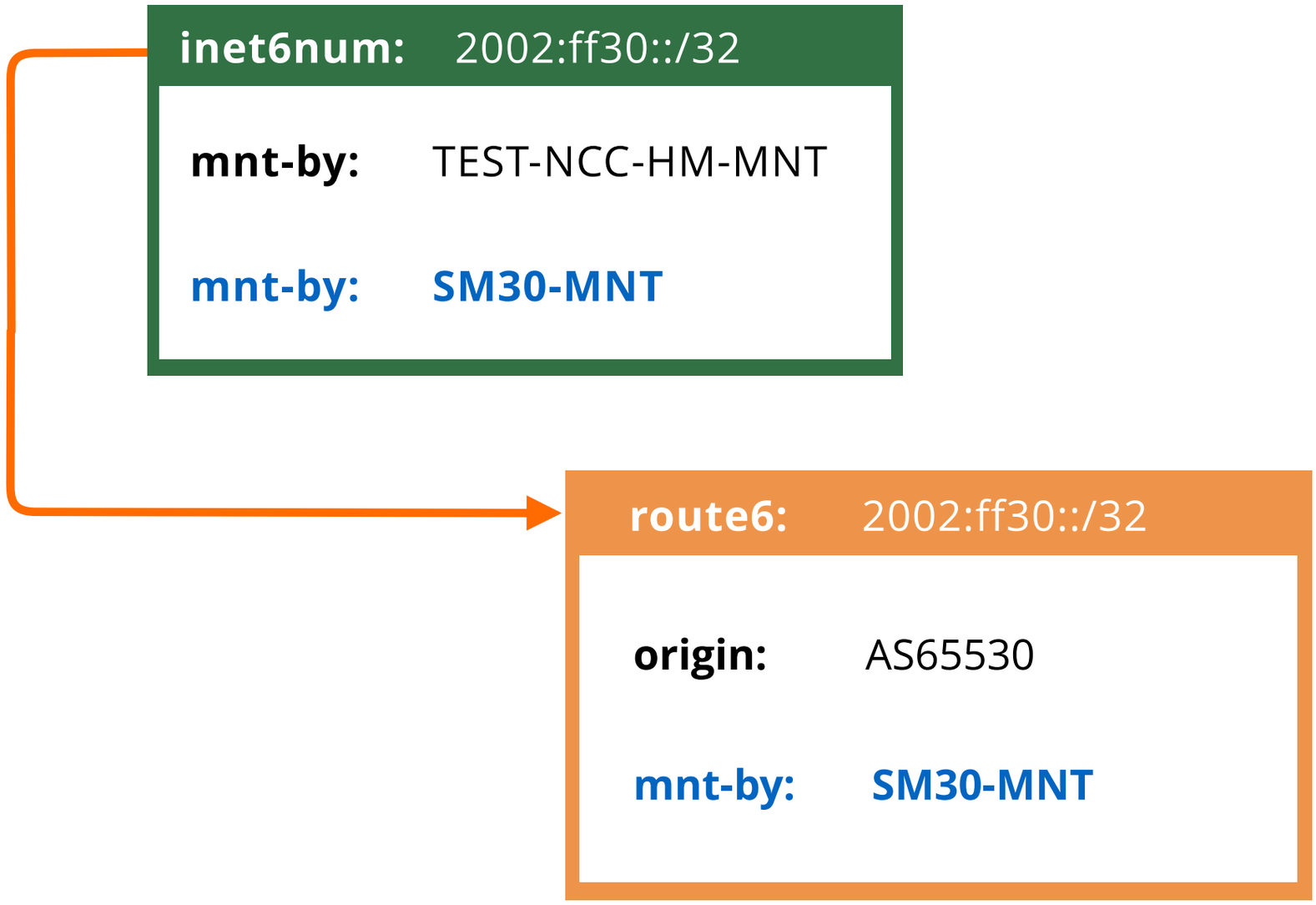
origin:	AS12345
mnt-by:	ANOTHER-MNT

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

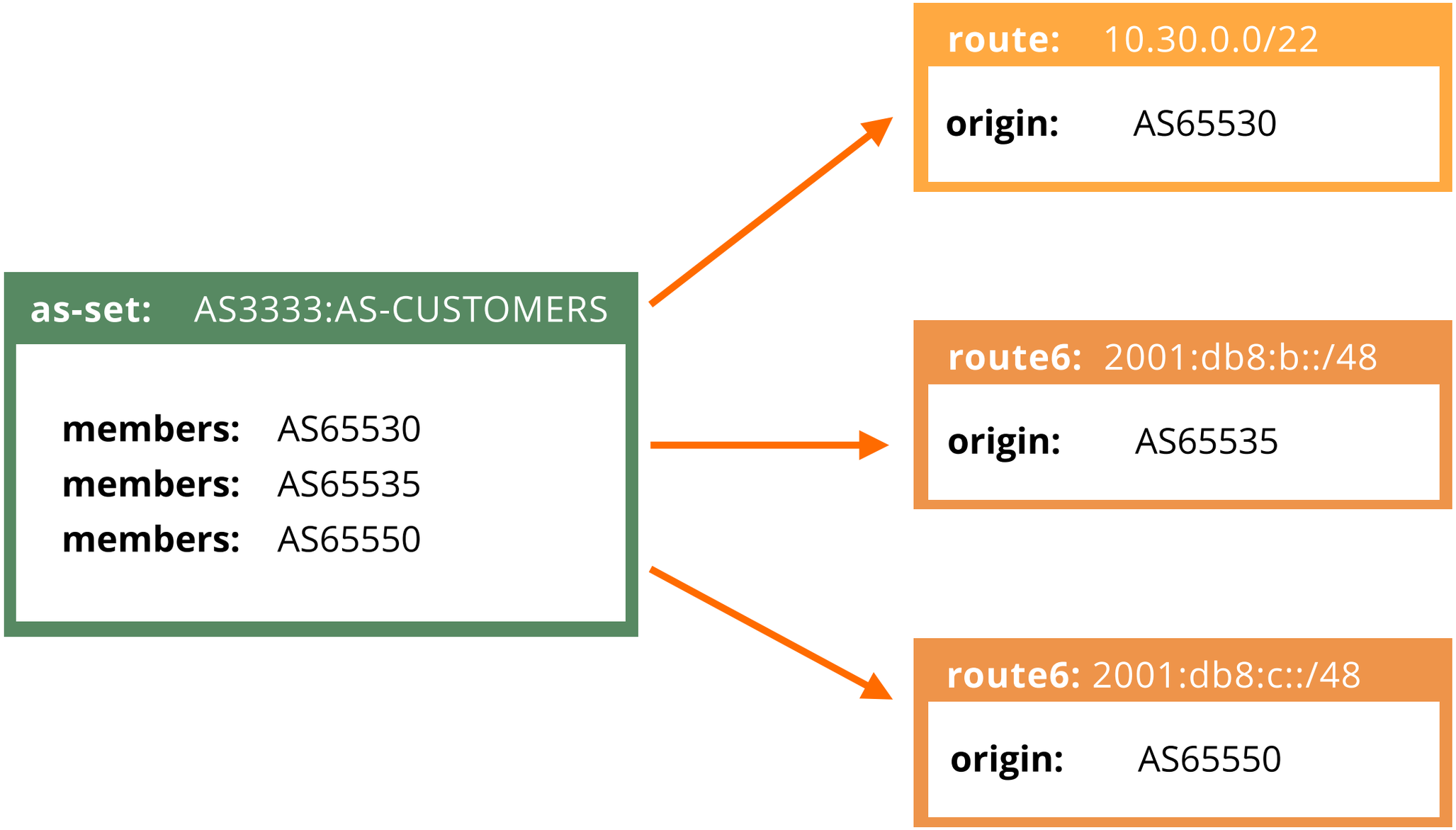
Registering IPv4 Routes

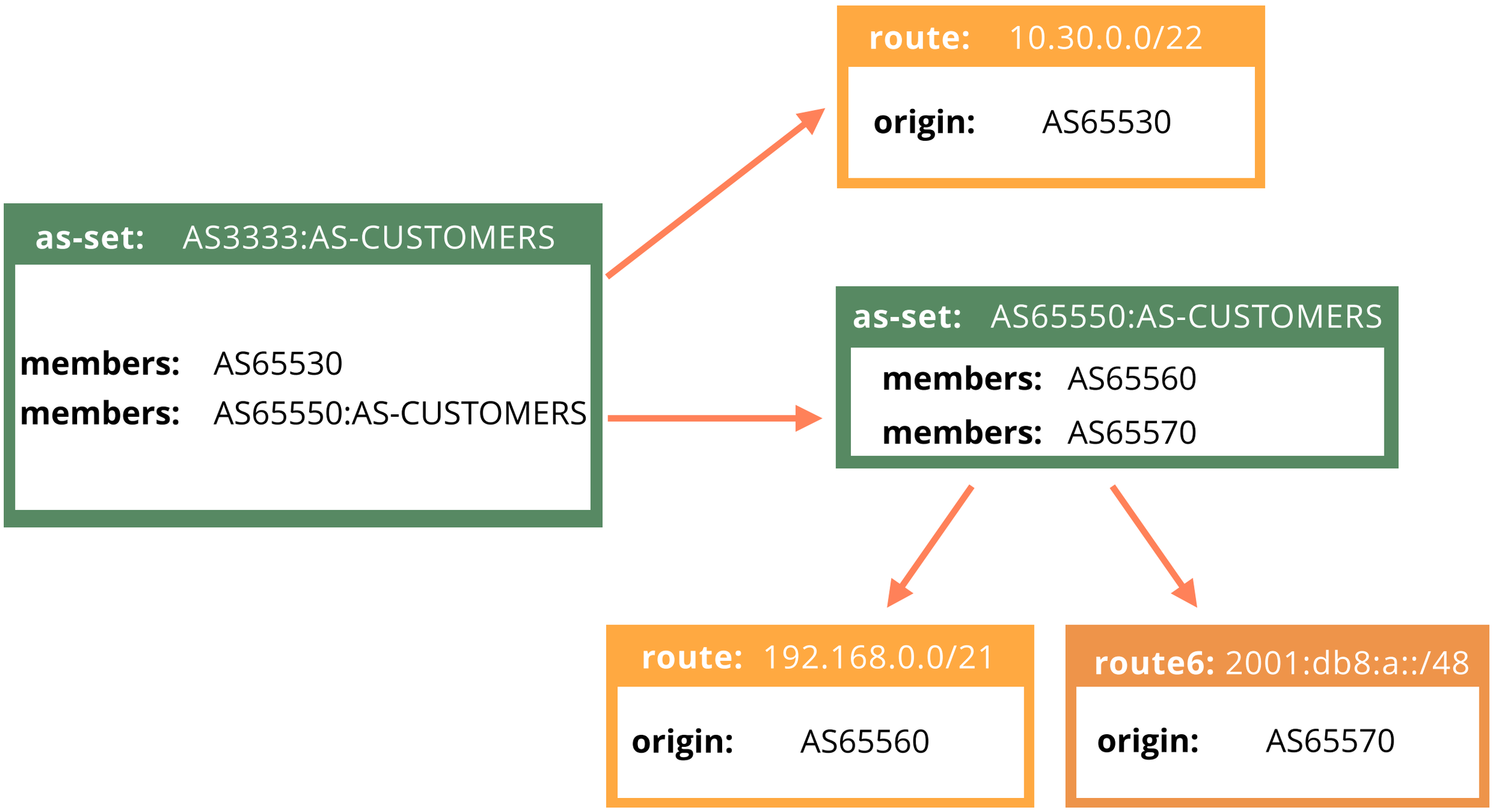


Registering IPv6 Routes



AS-Sets


route: 10.30.0.0/22
origin: AS65530
as-set: AS3333:AS-CUSTOMERS
members: AS65530
members: AS65535
members: AS65550
route6: 2001:db8:b::/48
origin: AS65535
route6: 2001:db8:c::/48
origin: AS65550


route: 10.30.0.0/22
origin: AS65530
as-set: AS3333:AS-CUSTOMERS
members: AS65530
members: AS65550:AS-CUSTOMERS
as-set: AS65550:AS-CUSTOMERS
members: AS65560
members: AS65570
route: 192.168.0.0/21
origin: AS65560
route6: 2001:db8:a::/48
origin: AS65570

Limitations of the IRR System

1. **Not globally deployed** (Just distributed databases)
2. **No central authority** (Who will verify the accuracy of the data?)
3. **No verification of holdership** (Anyone can input anything)
4. **Not updated properly** (Information is missing, outdated or incorrect)

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

Create Route(6) Object

Activity

Preparation

- Create a RIPE NCC Access account

  https://access.ripe.net/registration

- Go to RIPE **Test** Database

  https://apps-test.db.ripe.net

- Search for your IPv4 and IPv6 allocations and your AS number

- You can find the name and password of your maintainer object in the exercise booklet

Create route(6) Objects

- On the left side, click on “Create an Object”
- Choose “route” or “route6” and click on [Create]
- Enter the maintainer created for you (CMX-MNT)
- Fill in the template:

```
route:   192.XX.0.0/22
route6:  2001:ffXX::/32
origin:  AS1XX
```

Questions

Filtering

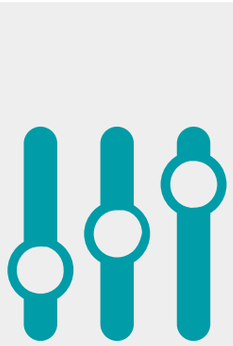
Section 9

What is Filtering

- Techniques used to decide:
  - Which routes to **allow** inside your routing table or network
  - And what you **announce** to your neighbours

Filtering is important, because...

It is your first line of **defence**

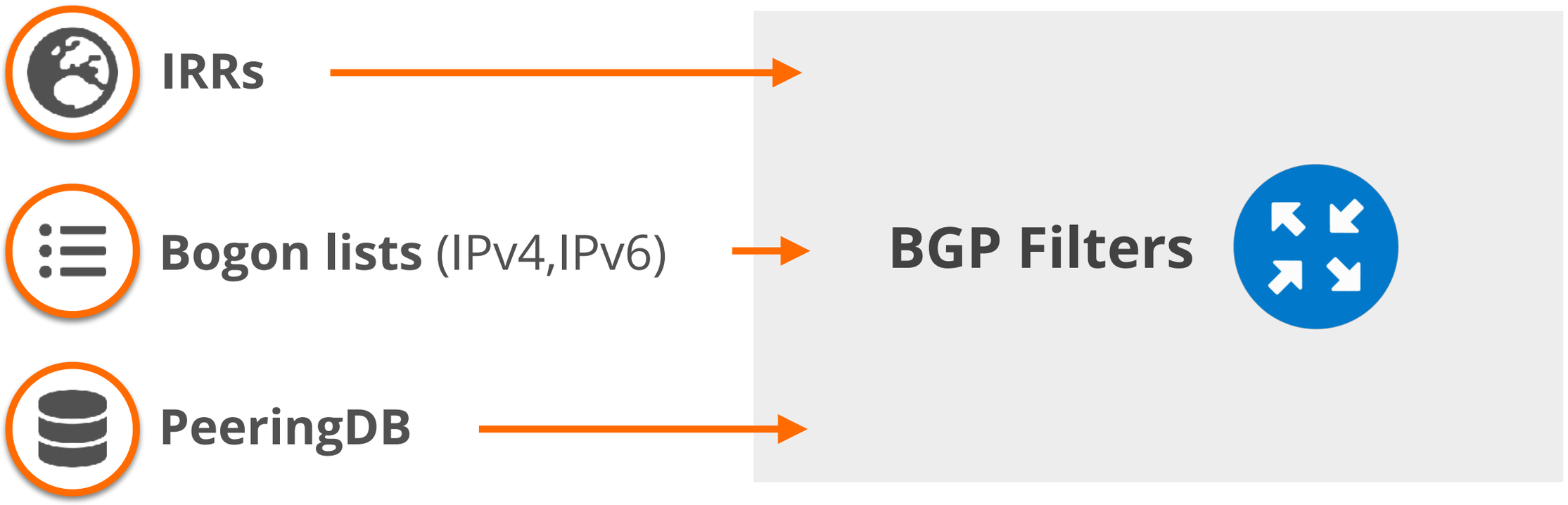


You **control** what you are announcing

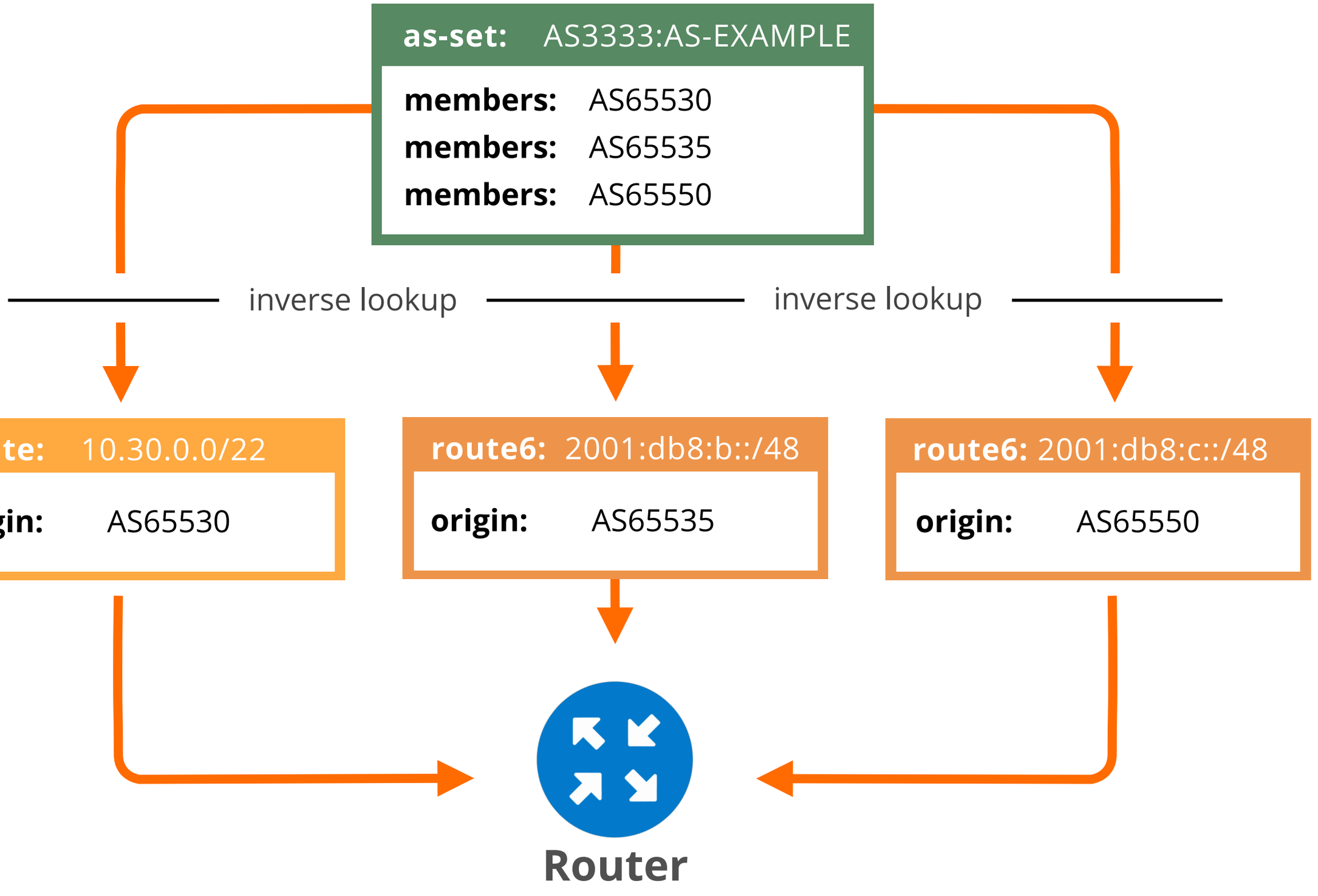
- You have no control over what other networks announce

To avoid issues, you have to decide what to **accept** from other networks

Data Sources



Generating a Prefix Filter



Filtering Principles

- Filter **as close to the edge** as possible
- Filter **as precisely** as possible
- Two filtering techniques:
  - Explicit Permit (permit then deny any)
  - Explicit Deny (deny then permit any)

Best Practices of Ingress Filters

- Don’t accept BOGON ASNs
- Don’t accept BOGON prefixes
- Don’t accept your own prefix
- Don’t accept default (unless you requested it)
- Don’t accept prefixes that are too specific
- Don’t accept if AS Path is too long
- Create filters based on Internet Routing Registries

What are Bogons

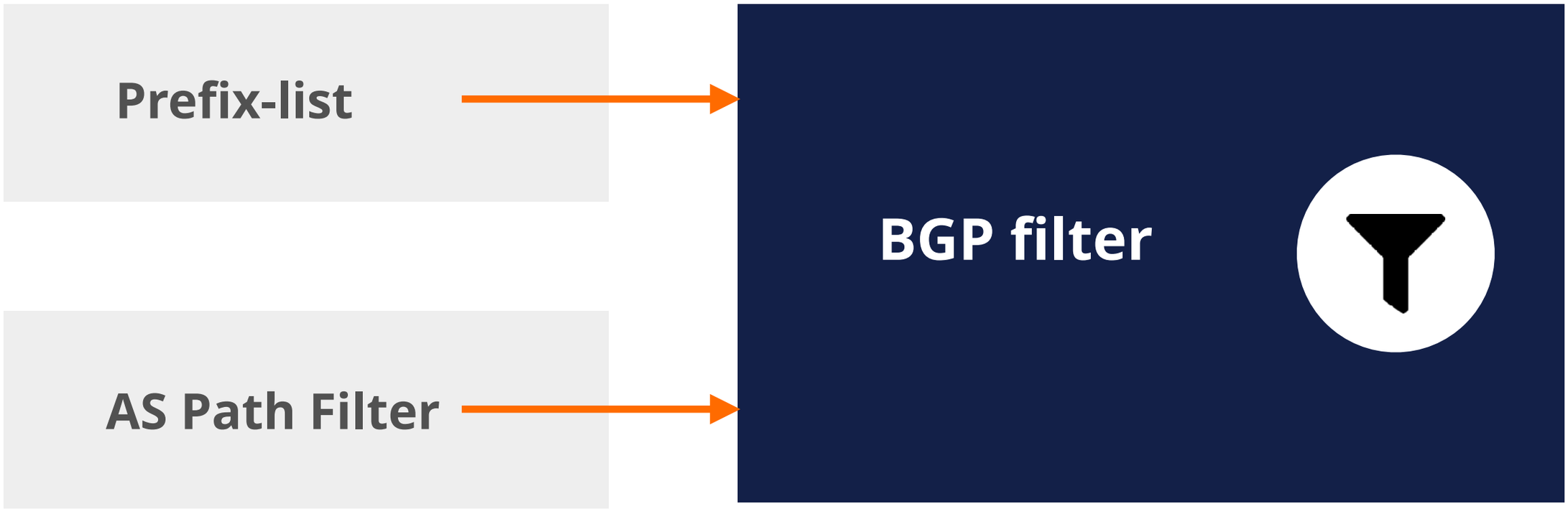
- Routes you **should not see** in the routing table
  - Private addresses
  - Non-allocated space
  - Reserved space (documentation, multicast, etc.)

Link: Team Cymru provides lists of bogons:
http://www.team-cymru.com/bogon-reference.html

ASN Bogons

ASNs	Status
0	Reserved - RFC7607
23456	AS_TRANS - RFC6793
64496-64511 and 65536-65551	Reserved for use in docs and code - RFC5398
64512-65534 and 4200000000-4294967294	Reserved for Private Use - RFC6996
65535 and 4294967295	Last 16 and 32 bit ASNs - RFC 7300
65552-131071	Reserved - IANA

BGP Prefix Filtering



Prefix-lists

- Lists of routes you want to **accept** or **announce**
- You can create them **manually** or **automatically** with data from IRRs
- Using a tool
  - Level3 Filtergen
  - bgpq4
  - peval

Longest Accepted Prefixes

- Small prefixes should not be a part of global routing
  - /24 (IPv4)
  - /48 (IPv6)
- Those prefixes generally neither announced nor accepted in the Internet

```
ip prefix-list SMALL-V4 permit 0.0.0.0/0 le 24
ipv6 prefix-list SMALL-V6 permit 2000::/3 le 48
```

AS Path Filter

- Filtering routes based on AS path information
- Widely used and highly scalable
- Applied same way as prefix-list filters

```
router bgp 65564
  network 10.0.0.0 mask 255.255.255.0
  neighbor 172.16.1.1 remote-as 65563
  neighbor 172.16.1.1 filter-list 1 out
  neighbor 172.16.1.1 filter-list 2 in

ip as-path access-list 1 permit 65564
ip as-path access-list 2 permit 65563
```

Defining Filters

Activity

Preparation (on R1)

- Examine your routing table

```
# show ip route bgp
# show ip bgp
# show ipv6 route bgp
# show bgp ipv6
```

- Do you see any prefix that is too specific?

Filter More Specifics (on R1)

- Create a routing policy to filter too specific prefixes from transit provider and IXP

```
(config)# ip prefix-list TRANS-IN-V4 seq 10 permit 0.0.0.0/0 le 24
(config)# ip prefix-list IXP-IN-V4 seq 10 permit 0.0.0.0/0 le 24
(config)# ipv6 prefix-list TRANS-IN-V6 seq 10 permit  2000::/3 le 48
(config)# ipv6 prefix-list IXP-IN-V6 seq 10 permit  2000::/3 le 48
```

Filter More Specifics (on R1)

- Apply the policy on inbound

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.132.X.1 prefix-list TRANS-IN-V4 in
(config-router-af)# neighbor 172.16.0.66 prefix-list IXP-IN-V4 in
(config-router-af)# neighbor 172.16.0.99 prefix-list IXP-IN-V4 in
(config-router-af)# address-family ipv6
(config-router-af)# neighbor 2001:ff32:0:XX::a prefix-list TRANS-IN-V6 in
(config-router-af)# neighbor 2001:ff69::66 prefix-list IXP-IN-V6 in
(config-router-af)# neighbor 2001:ff69::99 prefix-list IXP-IN-V6 in
```

Clear the BGP Sessions (on R1)

```
# clear bgp ipv4 unicast 172.16.0.66 in
# clear bgp ipv4 unicast 172.16.0.99 in
# clear bgp ipv4 unicast 10.132.X.1 in
# clear bgp ipv6 unicast 2001:ff69::66 in
# clear bgp ipv6 unicast 2001:ff69::99 in
# clear bgp ipv6 unicast 2001:ff32:0:XX::a in
```

Verify (on R1)

- Check BGP and routing table

```
# show bgp ipv4 unicast
# show bgp ipv6 unicast
# show ip route bgp | include /25
# show ipv6 route bgp | include /64
```

- Do you see any prefix that is too specific?

Filter Customer-1 Prefixes (on R2)

- Create a prefix-list for Customer-1 routes

```
(config)# ip prefix-list C1-IN-V4 seq 5 permit 10.X.1.0/24
```

- Apply the inbound policy to the neighbour

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.X.0.26 prefix-list C1-IN-V4 in
```

Filter Customer-2 Prefixes (on R3)

- Create a prefix-list for Customer-2 routes

```
(config)# ip prefix-list C2-IN-V4 seq 5 permit 10.X.2.0/24
```

- Apply the inbound policy to the neighbour

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.X.0.30 prefix-list C2-IN-V4 in
```

Questions

RPKI

Section 10

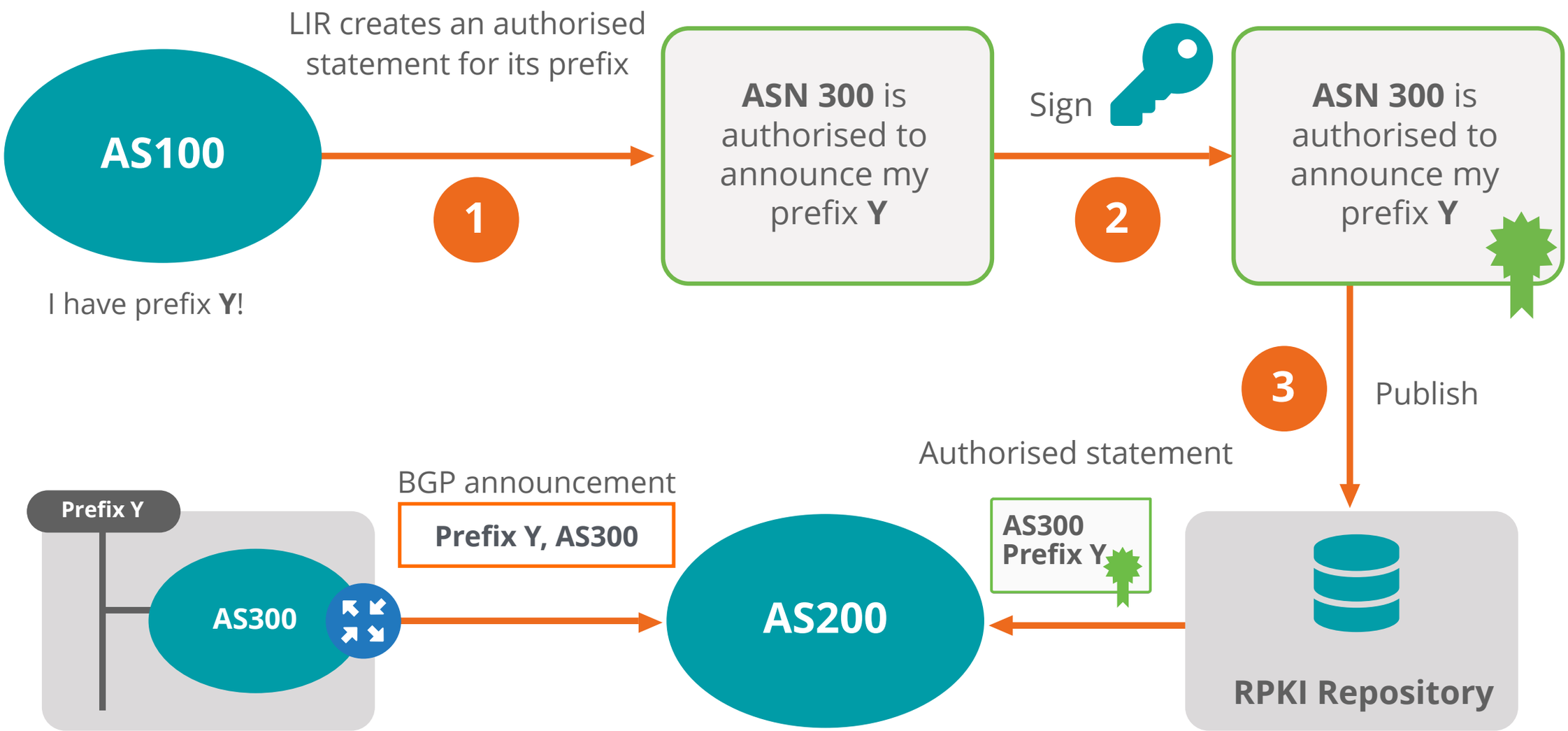
What is RPKI?

- RPKI is …
  - **Resource certification** (X.509 PKI certificates)
  - A security framework
- It is used to make Internet routing more secure and reliable

How RPKI helps for Routing Security

- Verifies the association between resource holders and their resources.
  - Proves holdership through a public key and certificate infrastructure
- Used to validate the origin of BGP announcements
  - Is the originating ASN authorised to originate a particular prefix?
- Stepping stone to “Path Validation”

How does it work?



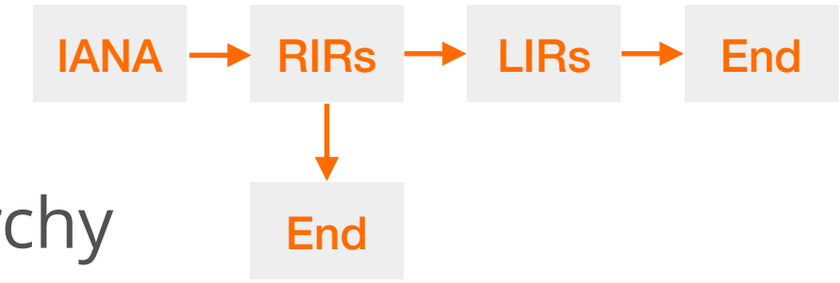
4 Others use those statements to make better routing decisions!

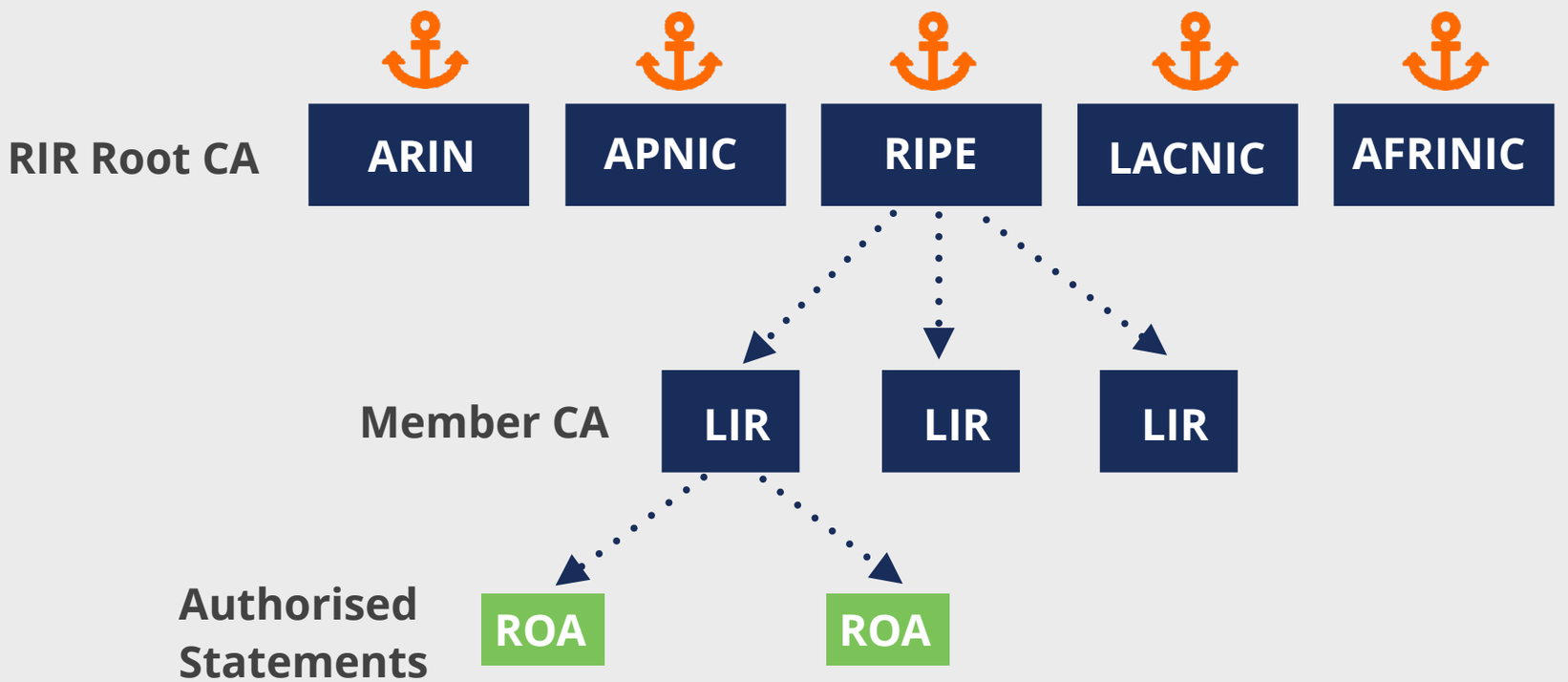
Implementing RPKI helps to prevent…

- BGP Origin Hijacks
  - Caused by malicious activities
- Mis-origination
  - Due to typos/fat fingers
- Route leaks
  - Caused by configuration mistakes

Trust in RPKI

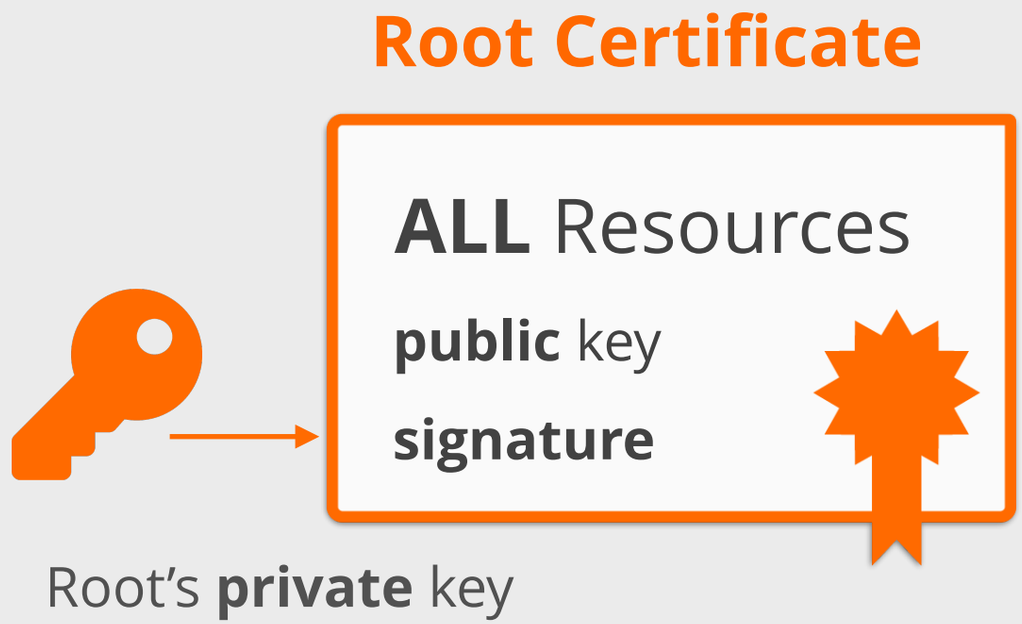
- RPKI relies on five RIRs as Trust Anchors
- Certificate structure follows the RIR hierarchy
- RIRs issue certificates to resource holders





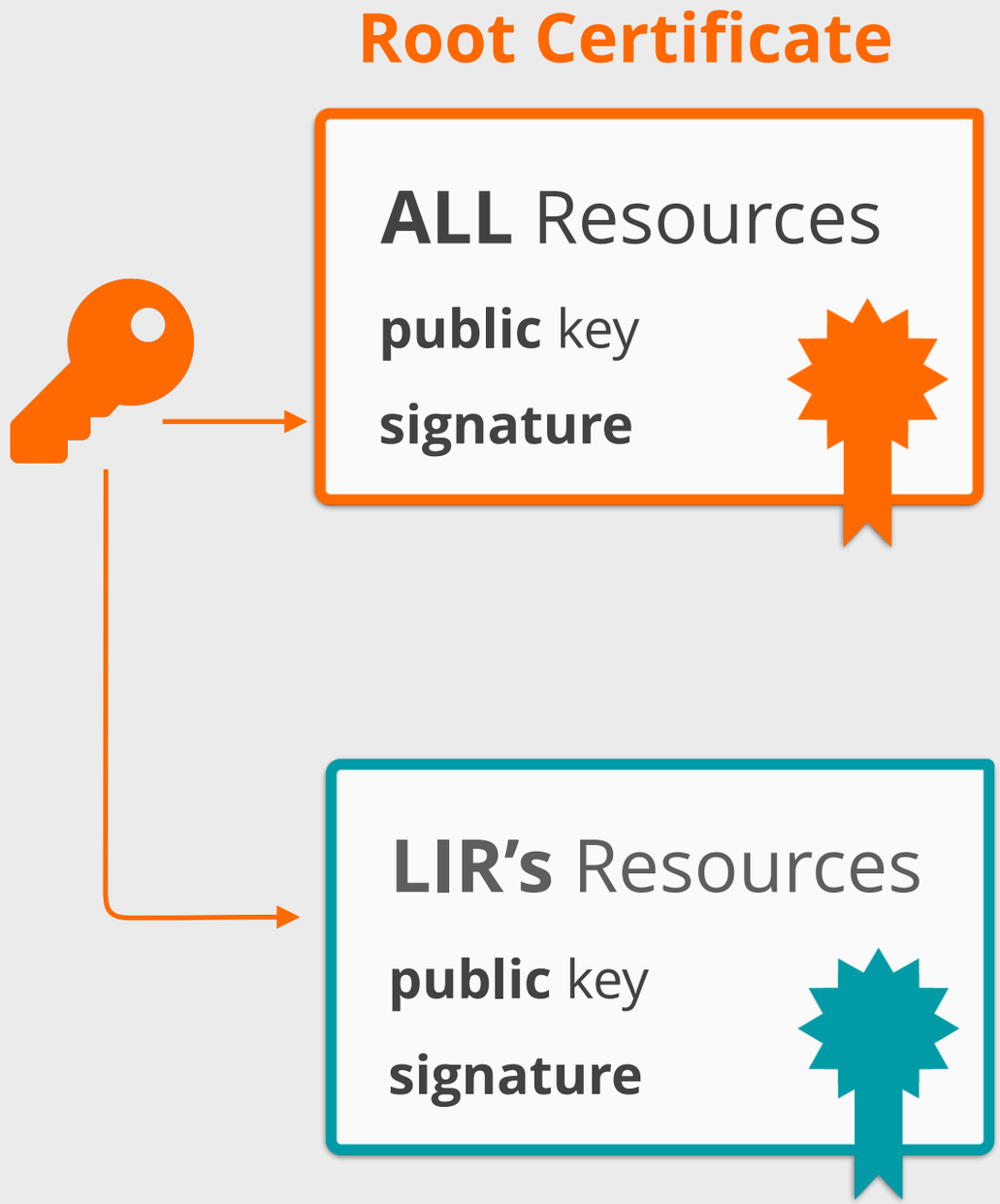
Trust in RPKI

- **Root certificate**
  - **Self-signed**
  - RIRs use root certificate to sign LIRs' certificates



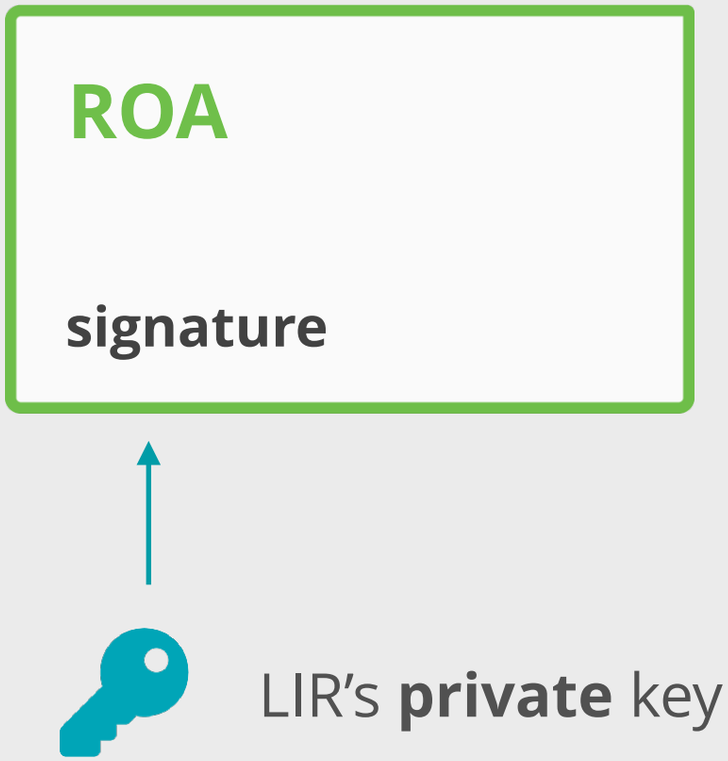
Trust in RPKI

- **Root certificate**
  - **Self-signed**
  - RIRs use root certificate to sign LIRs' certificates
- **LIR certificate**
  - Resource certificate for member allocations
  - Binds LIR's resources to LIR's public key
  - Proves legitimate holdership for the LIR's resources

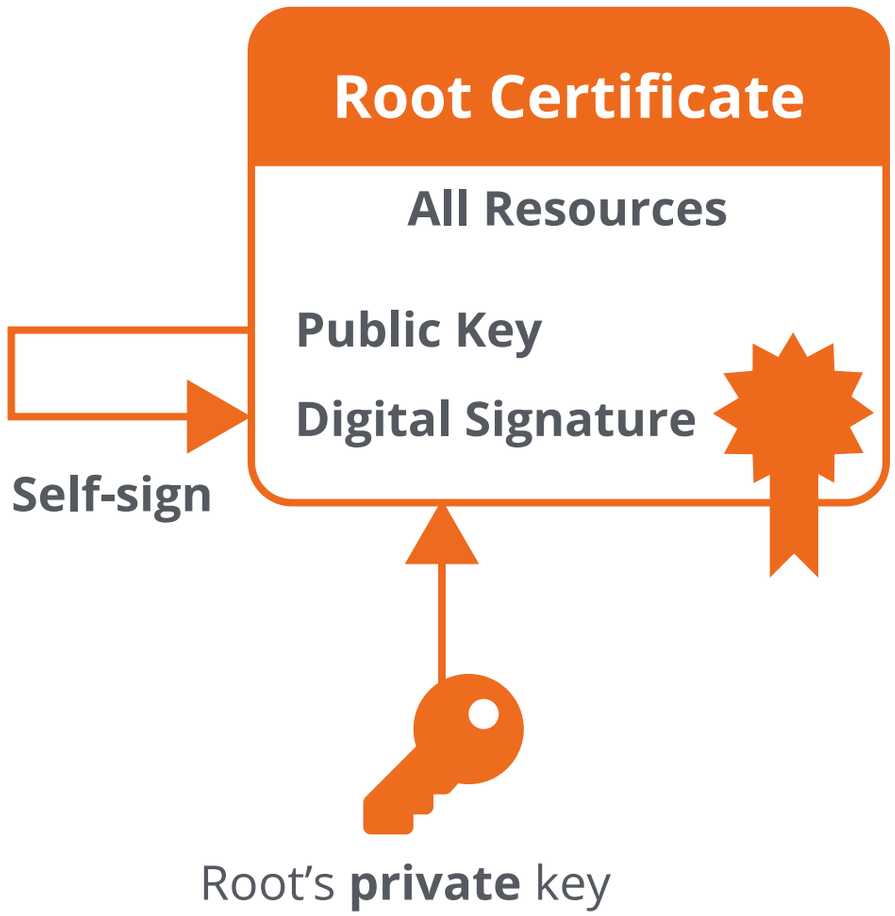


Trust in RPKI

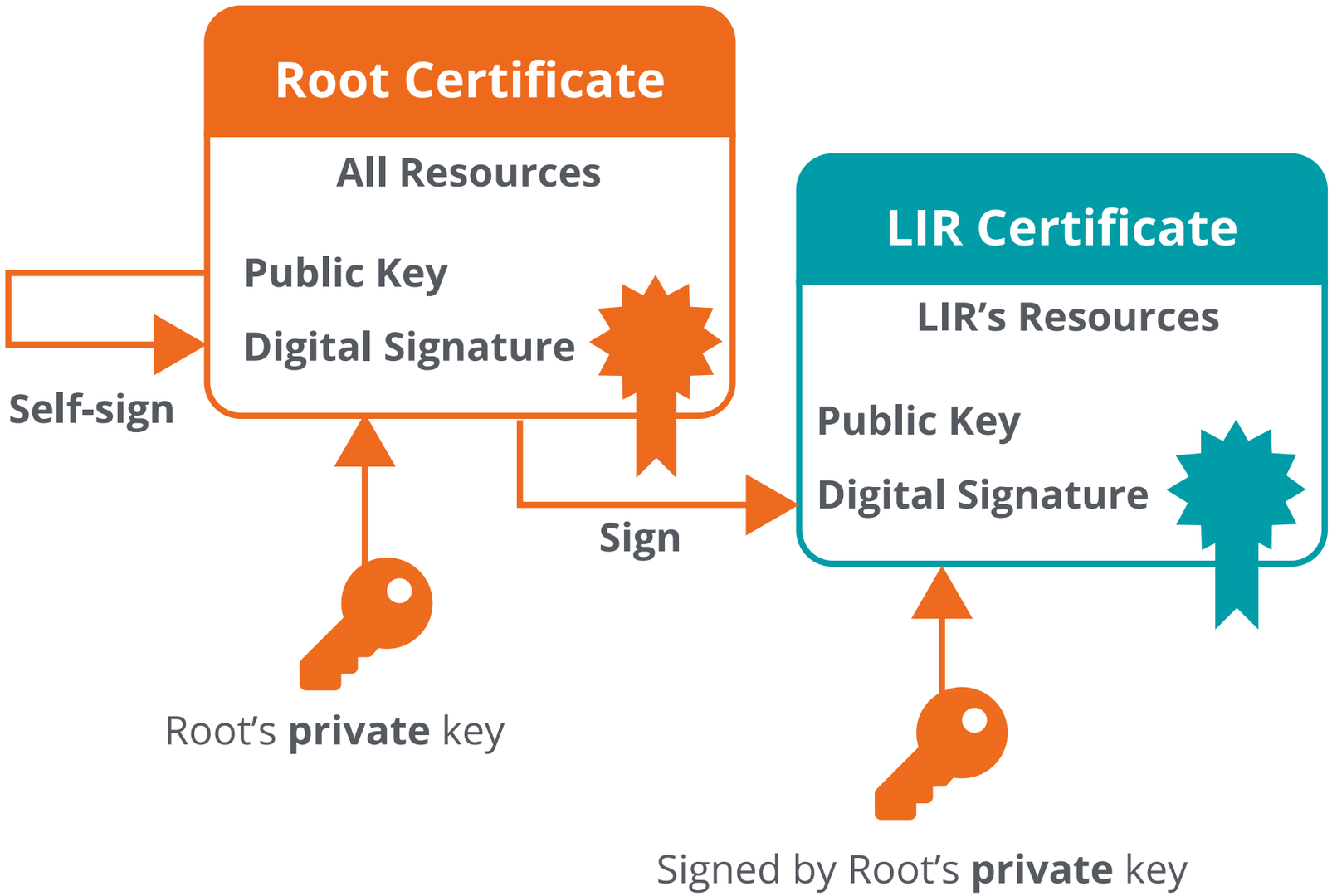
- **Authorised statements**
  - Known as a ROA (Route Origin Authorisation)
  - Cryptographically signed object
  - Signed by LIR’s private key



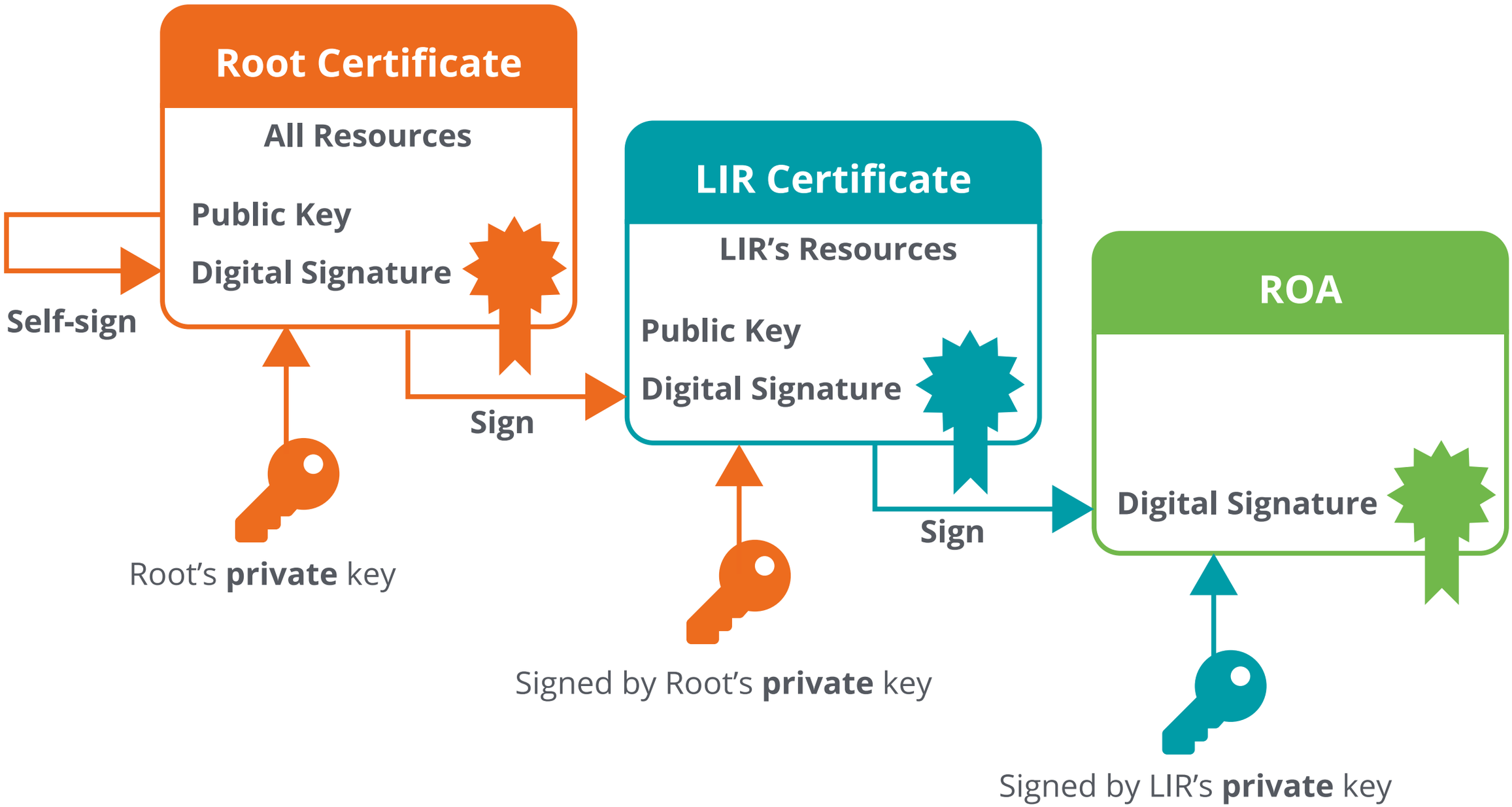
RPKI Chain of Trust



RPKI Chain of Trust



RPKI Chain of Trust



Elements of RPKI

- RPKI system consists of two parts…

SIGNING

Create ROAs for your prefixes in the RPKI system

+

VALIDATION

Verify the information provided by others

Elements of RPKI

- RPKI system consists of two parts…

SIGNING

Create ROAs for your prefixes in the RPKI system

+

VALIDATION

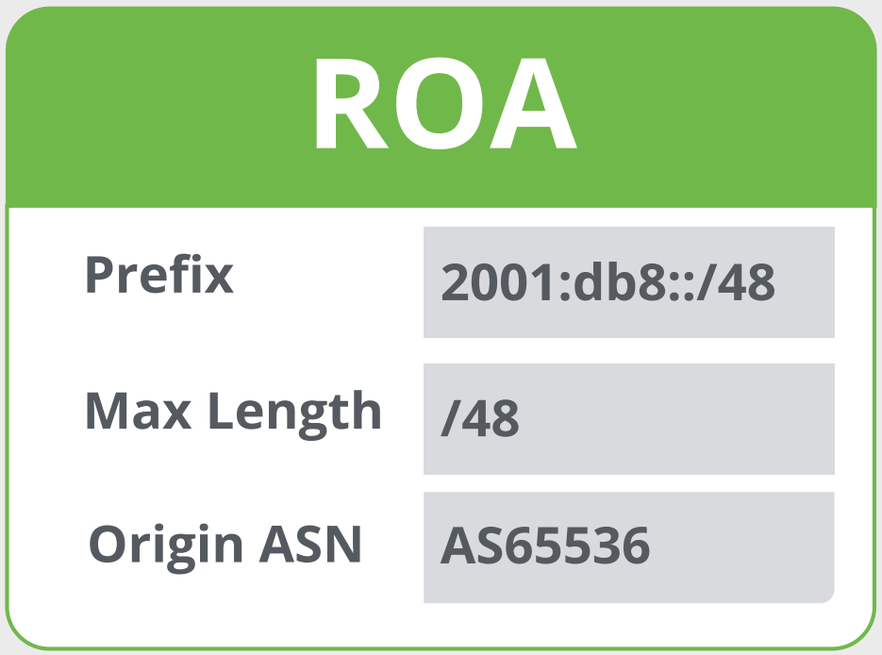
Verify the information provided by others

Registering in the RPKI system

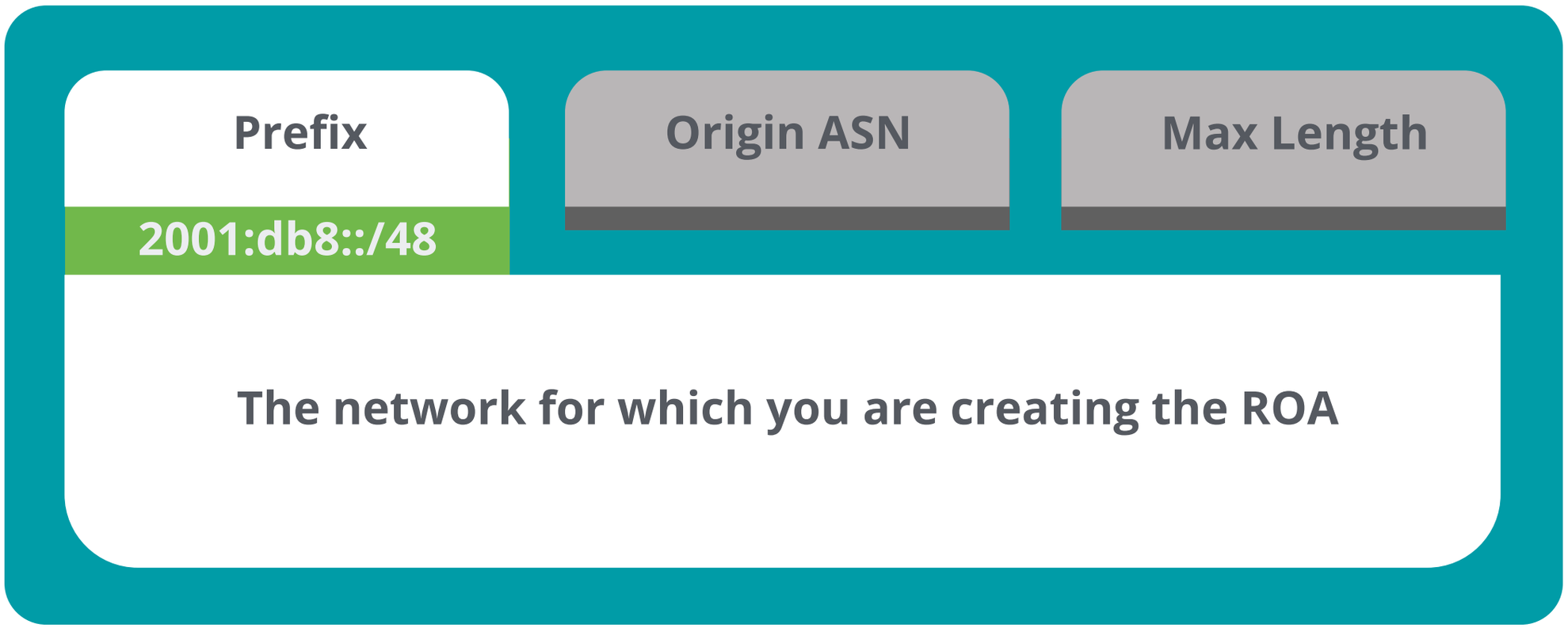
Route Origin Authorisation

What are ROAs?

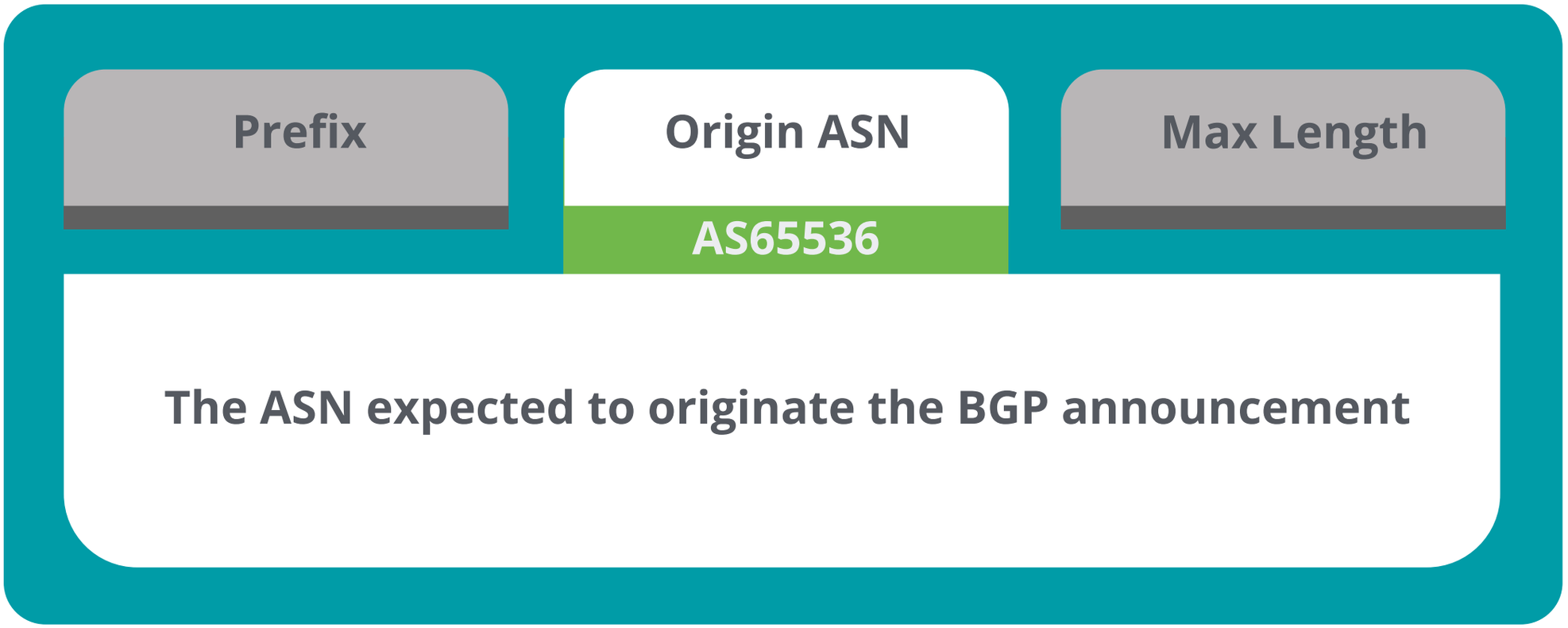
- An **authorised statement** created by the resource holder
- It states that a certain prefix can be originated by a certain AS
- LIRs can create ROAs for their resources
- Multiple ROAs can exist for the same prefix
- ROAs can overlap



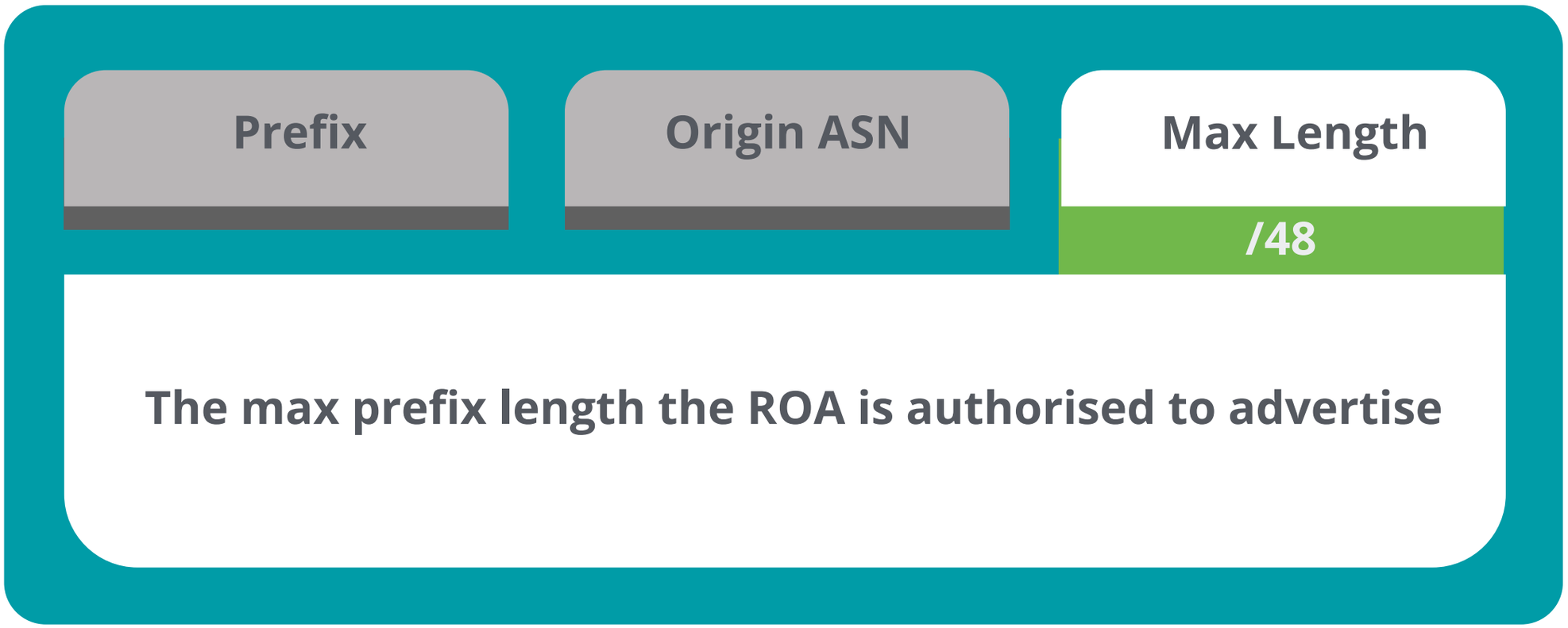
What is in a ROA?



What is in a ROA?



What is in a ROA?



Max-Length

RIPE NCC (AS3333) has an IP address allocation

193.0.0.0/21

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

193.0.0.0/21

ROA

Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

According to ROA;

/21

193.0.0.0/21

ROA

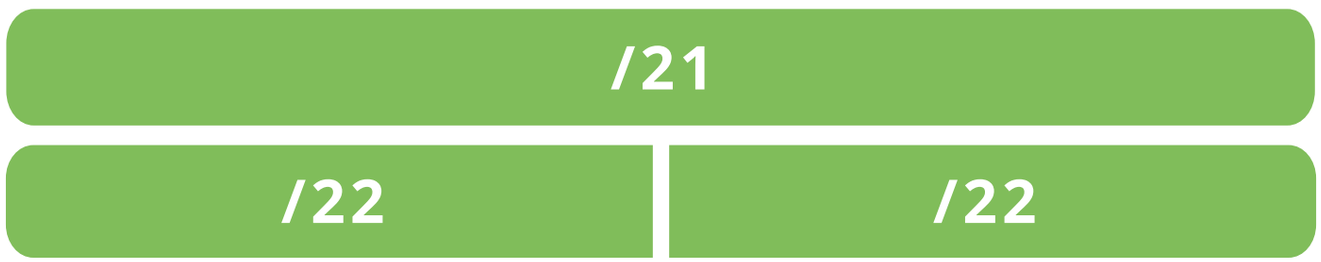
Prefix	**193.0.0.0/21**
Max Length	**/22**
Origin ASN	**AS3333**

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

According to ROA;



193.0.0.0/21

ROA

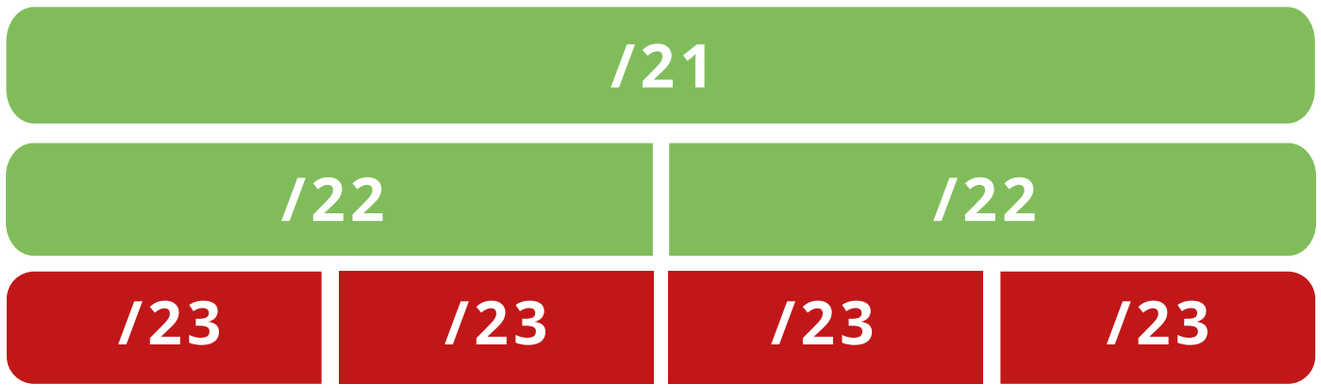
Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

According to ROA;



193.0.0.0/21

ROA

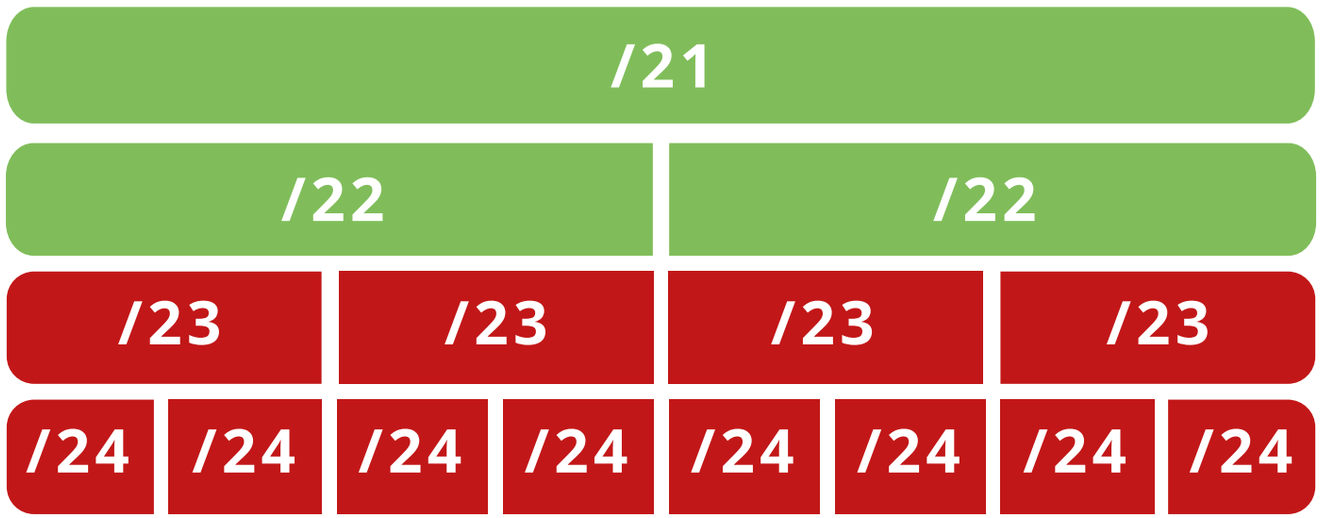
Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

According to ROA;



193.0.0.0/21

ROA

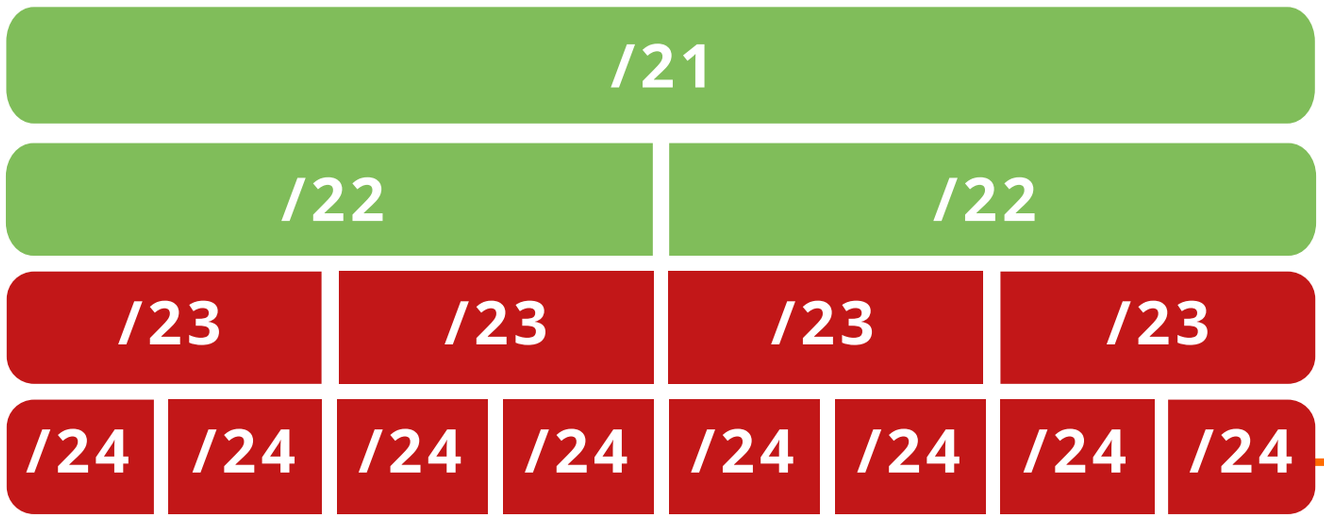
Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333

Max-Length

RIPE NCC (AS3333) has an IP address allocation

AS3333 creates this ROA

According to ROA;



193.0.0.0/21

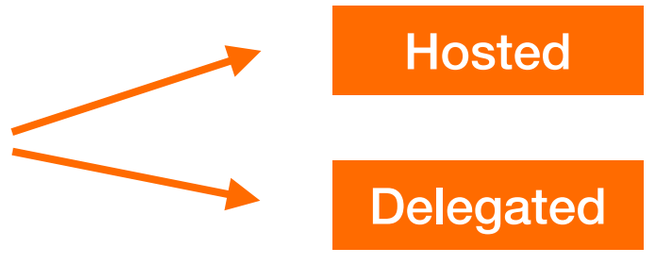
ROA

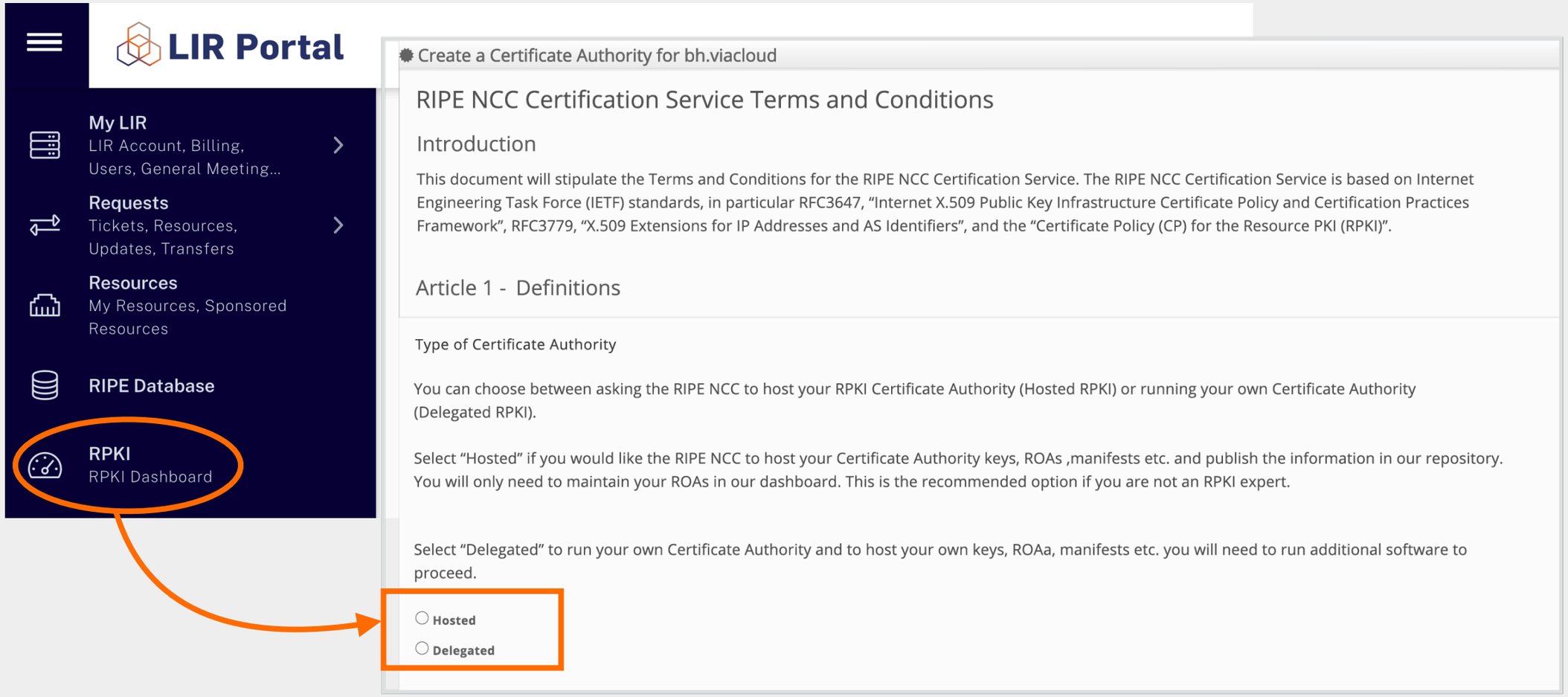
Prefix	**193.0.0.0/21**
Max Length	**/22**
Origin ASN	**AS3333**

Any more specific announcements are unauthorised by the ROA.

How to create a ROA?

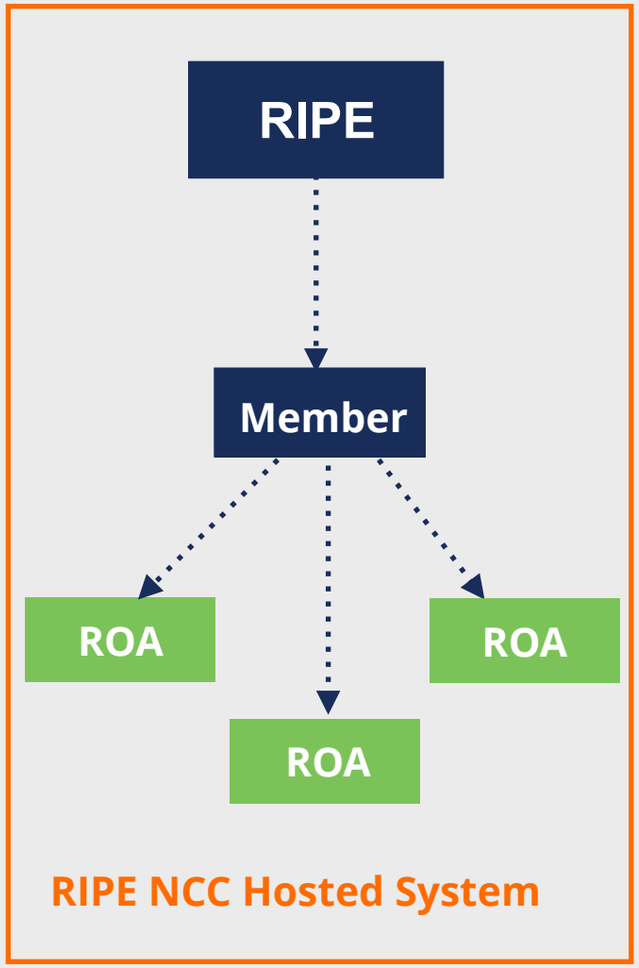
- Login to LIR Portal (my.ripe.net)
- Go to the RPKI Dashboard
- Choose which RPKI model to use





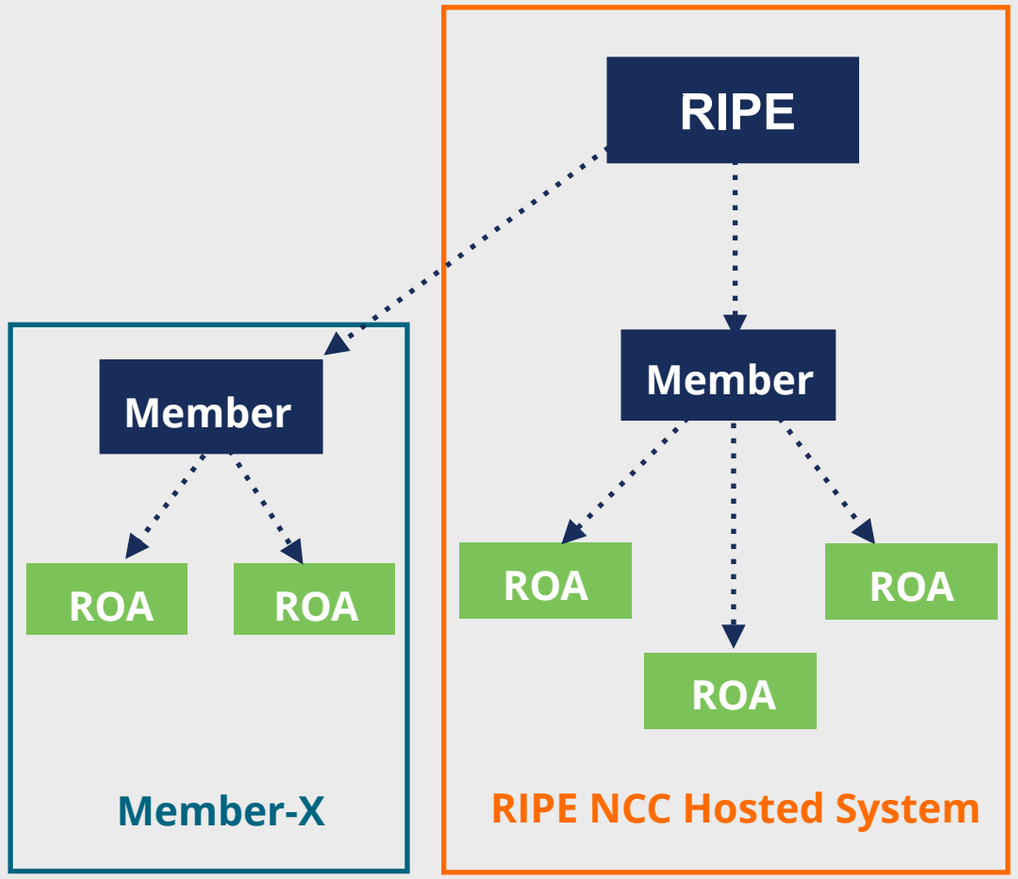
Hosted RPKI

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



Delegated RPKI

- Each LIR manages its part of the RPKI system
  - Runs its own CA as a child of the RIR
  - Manages keys/key rollovers
  - Creates ROAs in its own platform
  - Signs and publishes ROAs



RIPE NCC Hosted Solution

RPKI Dashboard

3 CERTIFIED RESOURCES | NO ALERT EMAIL CONFIGURE

2 BGP Announcements

0 Valid | **0** Invalid | **2** Unknown

0 ROAs

0 OK | **0** Causing problems

BGP Announcements | Route Origin Authorisations (ROAs) | History

Search...

Create ROAs for selected BGP Announcements

Valid | Invalid | Unknown

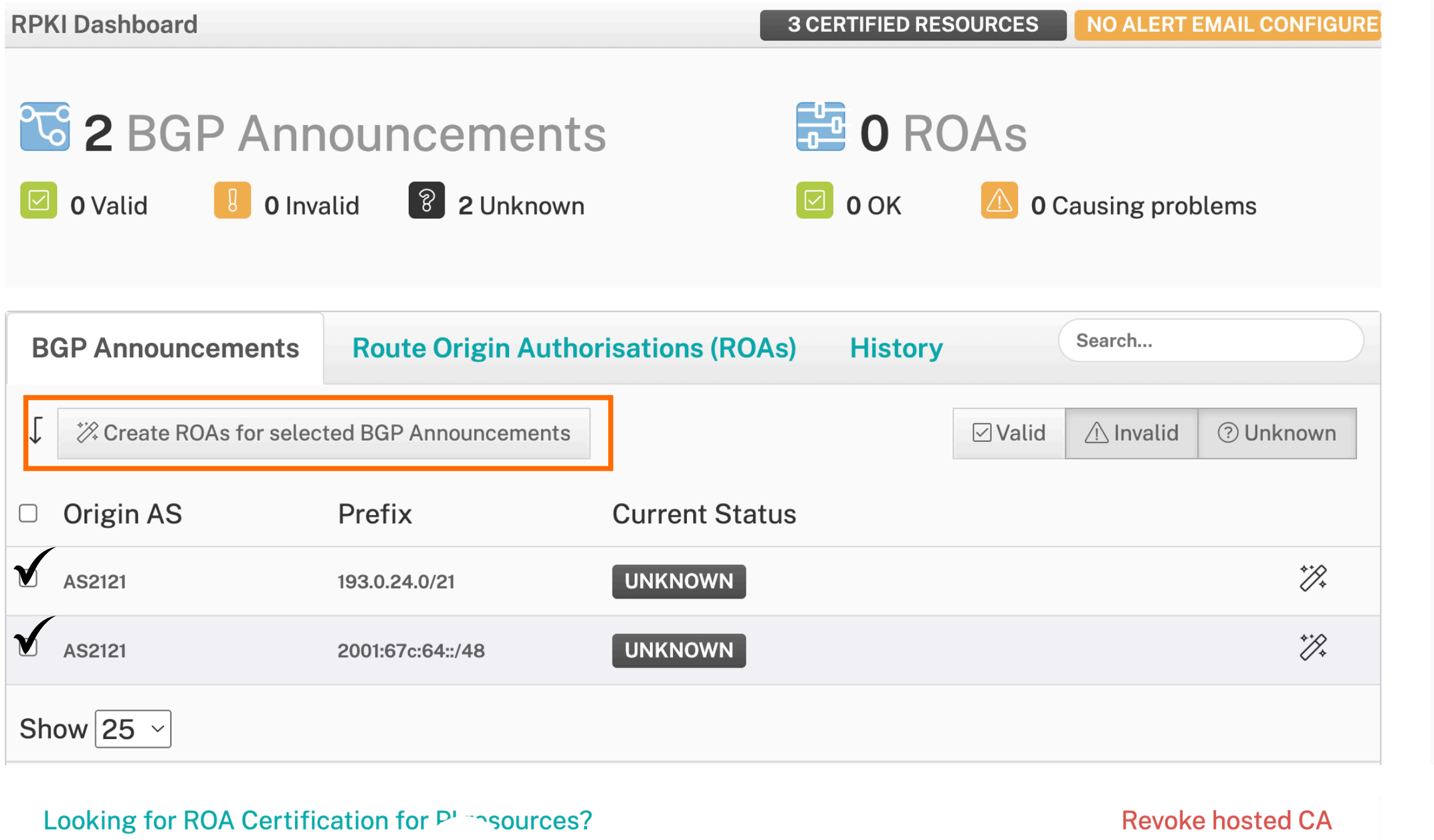
Origin AS	Prefix	Current Status
AS2121	193.0.24.0/21	UNKNOWN
AS2121	2001:67c:64::/48	UNKNOWN

Show 25

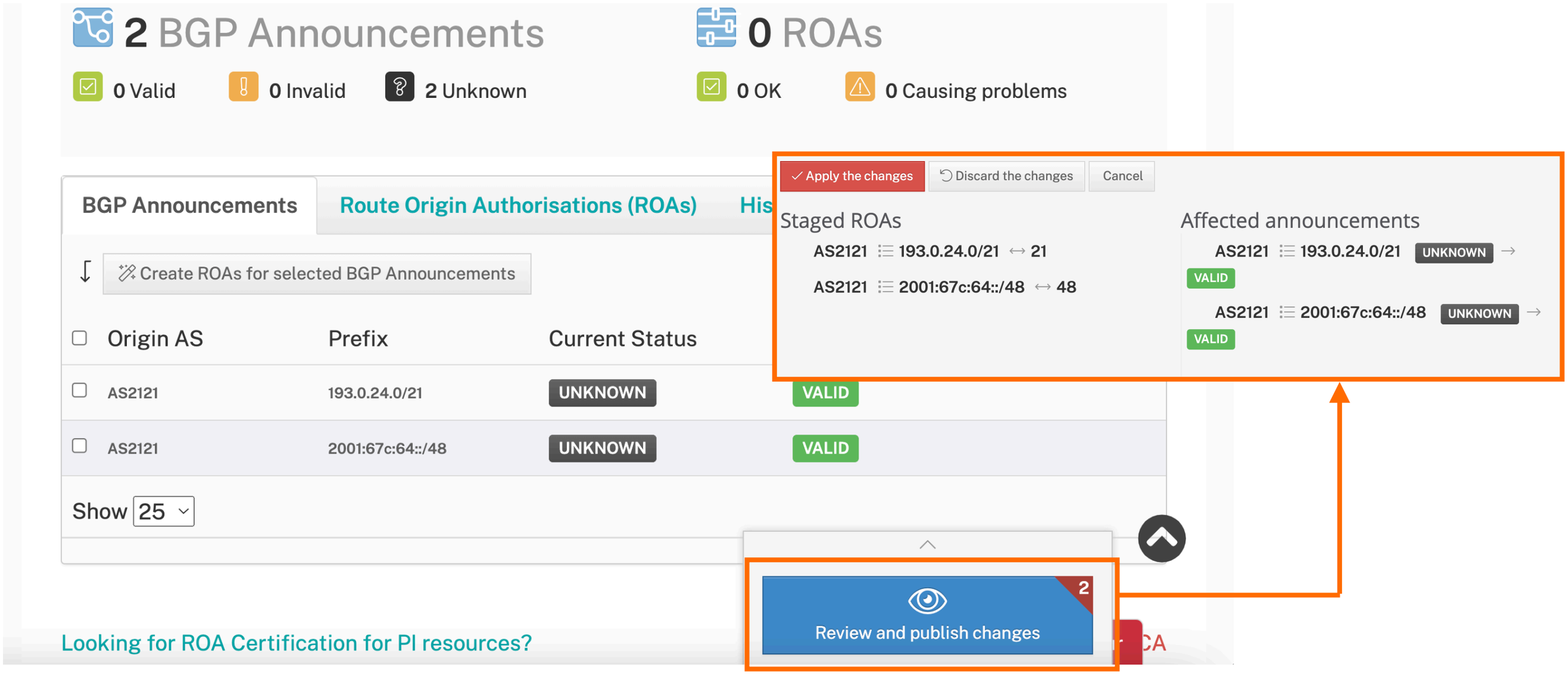
Looking for ROA Certification for PI resources?

Revoke hosted CA

RIPE NCC Hosted Solution



RIPE NCC Hosted Solution



RIPE NCC Hosted Solution

2 BGP Announcements

2 Valid **0** Invalid **0** Unknown

2 ROAs

2 OK **0** Causing problems

BGP Announcements | Route Origin Authorisations (ROAs) | History

Search...

Create ROAs for selected BGP Announcements

Valid | Invalid | Unknown

Origin AS	Prefix	Current Status
AS2121	193.0.24.0/21	VALID
AS2121	2001:67c:64::/48	VALID

Show 25

Looking for ROA Certification for PI resources?

Revoke hosted CA

RIPE NCC Hosted Solution

2 BGP Announcements

2 Valid **0** Invalid **0** Unknown

2 ROAs

2 OK **0** Causing problems

BGP Announcements | Route Origin Authorisations (ROAs) | History

Search...

Create ROAs for selected BGP Announcements

Valid | Invalid | Unknown

Origin AS	Prefix	Current Status
AS2121	193.0.24.0/21	VALID
AS2121	2001:67c:64::/48	VALID

Show 25

Looking for ROA Certification for PI resources?

Revoke hosted 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

Activity

Create ROAs for your LIR

- In this exercise, you are going to create ROAs for your LIR's prefixes in a **test environment**
- Login to test RPKI Dashboard with your LIR account

https://localcert.ripe.net/

- On the left side, click on "Resource Certification" and then "Dashboard"
- **Activity #1 :** Create ROAs for your BGP announcements
- **Activity #2 :** Create a more specific ROA for one of your BGP prefixes

Activity #1

- Choose all the BGP Announcements for which you would like to create a ROA
- Click on “Create ROAs for selected BGP Announcements”
- Review and publish changes
- Verify that the ROAs have been created

Activity #2

- Create a **more specific** ROA for one of your IPv4/IPv6 prefixes
- On RPKI Dashboard, go to “Route Origin Authorisations (ROA)“ tab, and click “New ROA”
- Pick one of your networks and decide on a more specific
  - i.e.: /24 or even /25 for IPv4
  - i.e.: /36 or /40 for IPv6
- Create a ROA with your assigned ASN as the origin
  - i.e.: Number 13 == AS113 or Number 9 == AS109

Questions

RPKI Validation

Deploying RPKI Validators

Elements of RPKI

- RPKI system consists of two parts…

SIGNING

Create ROAs for your prefixes in the RPKI system

+

VALIDATION

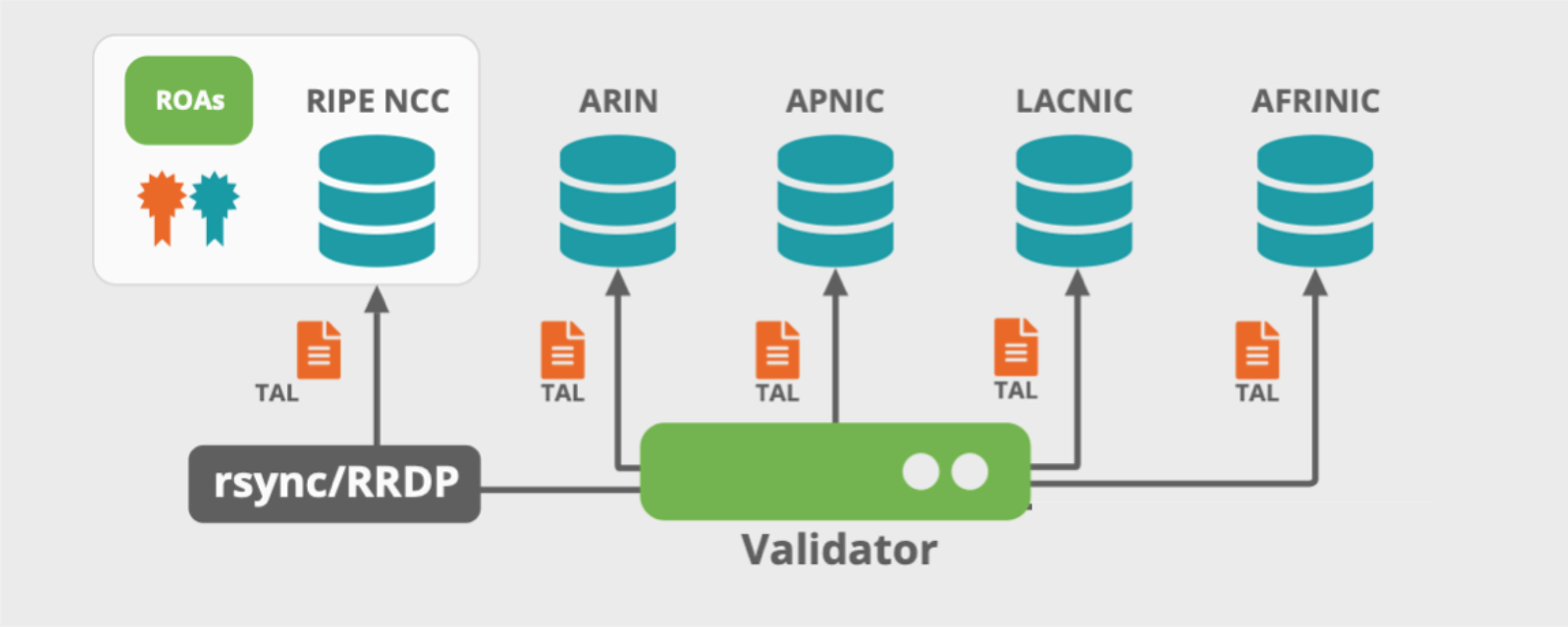
Verify the information provided by others

RPKI Validation

- Verifying the information provided by others
  - Proves holdership through a public key and certificate infrastructure
- In order to validate RPKI data, you need to …
  - install a validator software locally in your network
- Goal is to validate the “origin of BGP announcements”
  - Known as BGP Origin Validation (BGP OV) or Route Origin Validation (ROV)

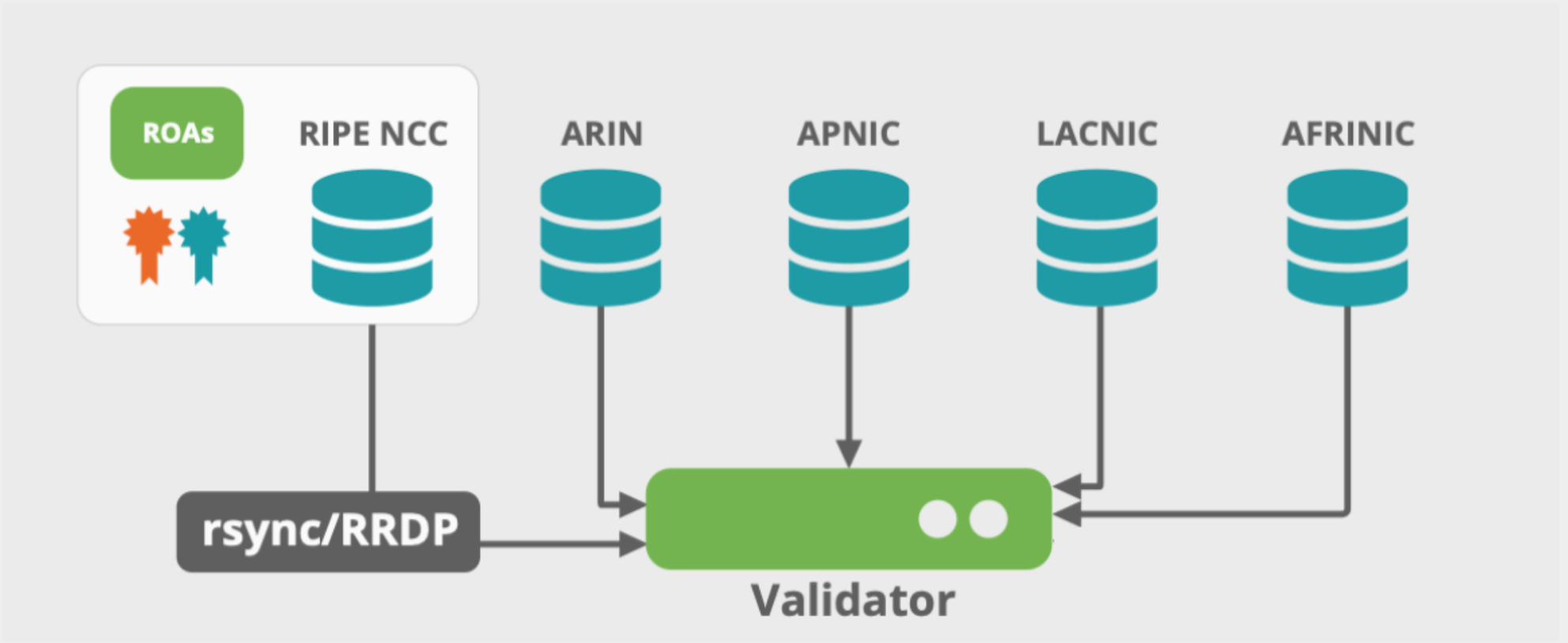
RPKI Validators

- Also known as Relying Party Software
- Connects to RPKI repositories via rsync or RRDP protocol
- Checks the information in TALs to connect to the repositories



RPKI Validators

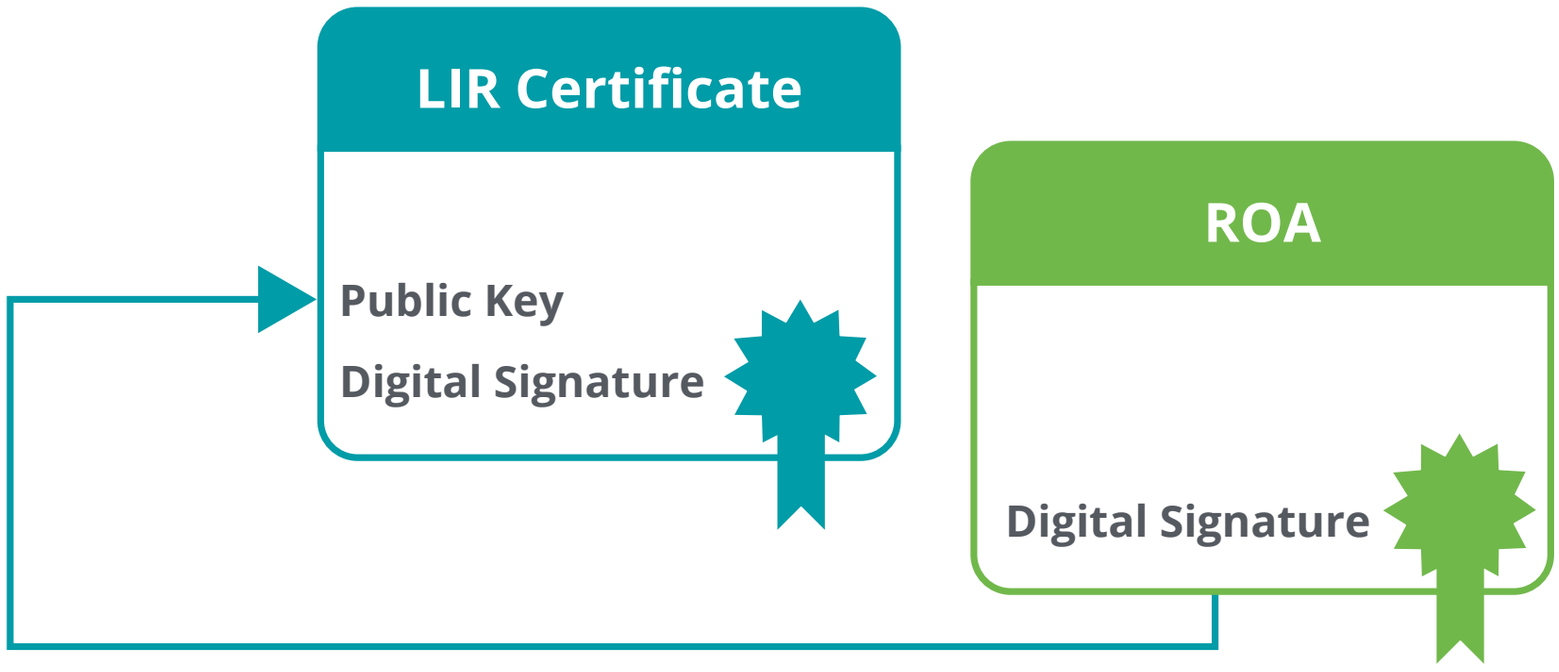
- Validator
  - Downloads all ROAs from RPKI repositories (from RIRs and external repos)
  - Validates the chain of trust for all ROAs and associated CAs
  - Creates a local “validated cache” with all the valid ROAs



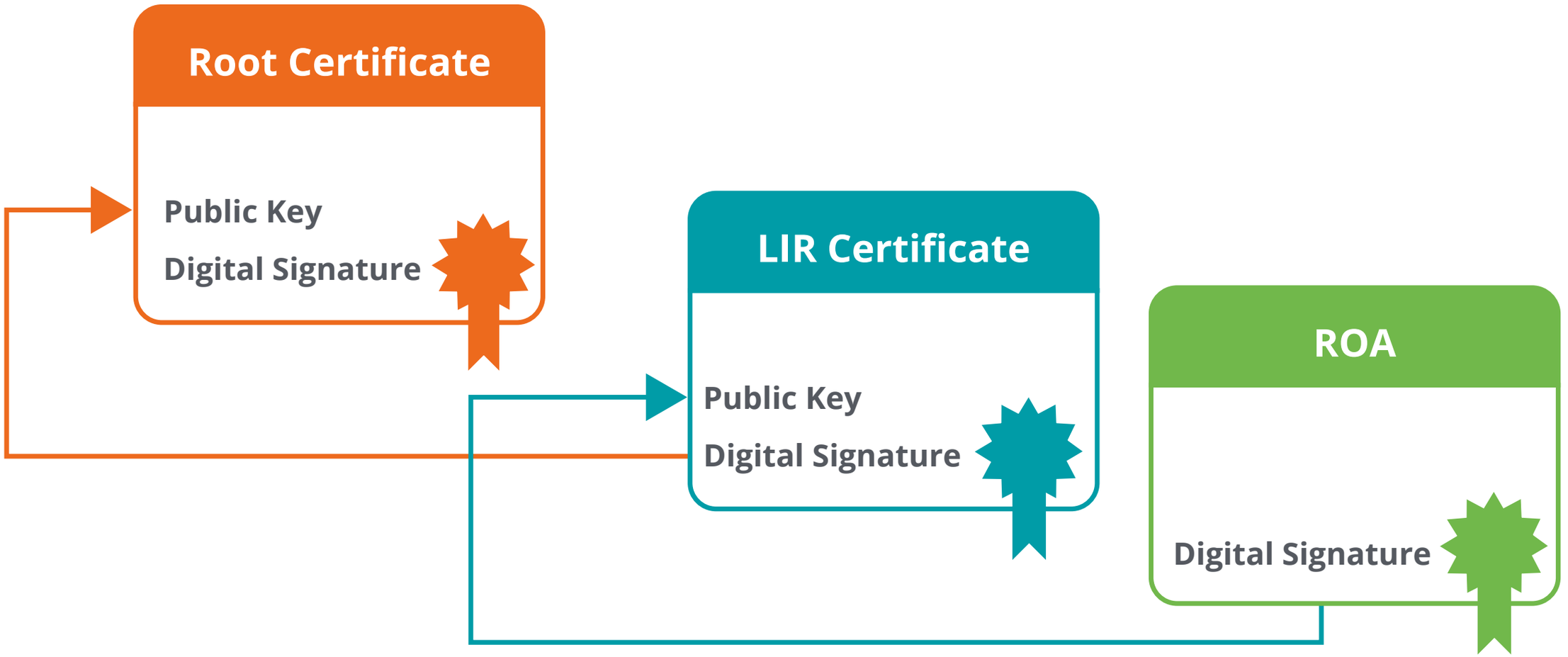
ROA Validation Process

ROA Validation Process

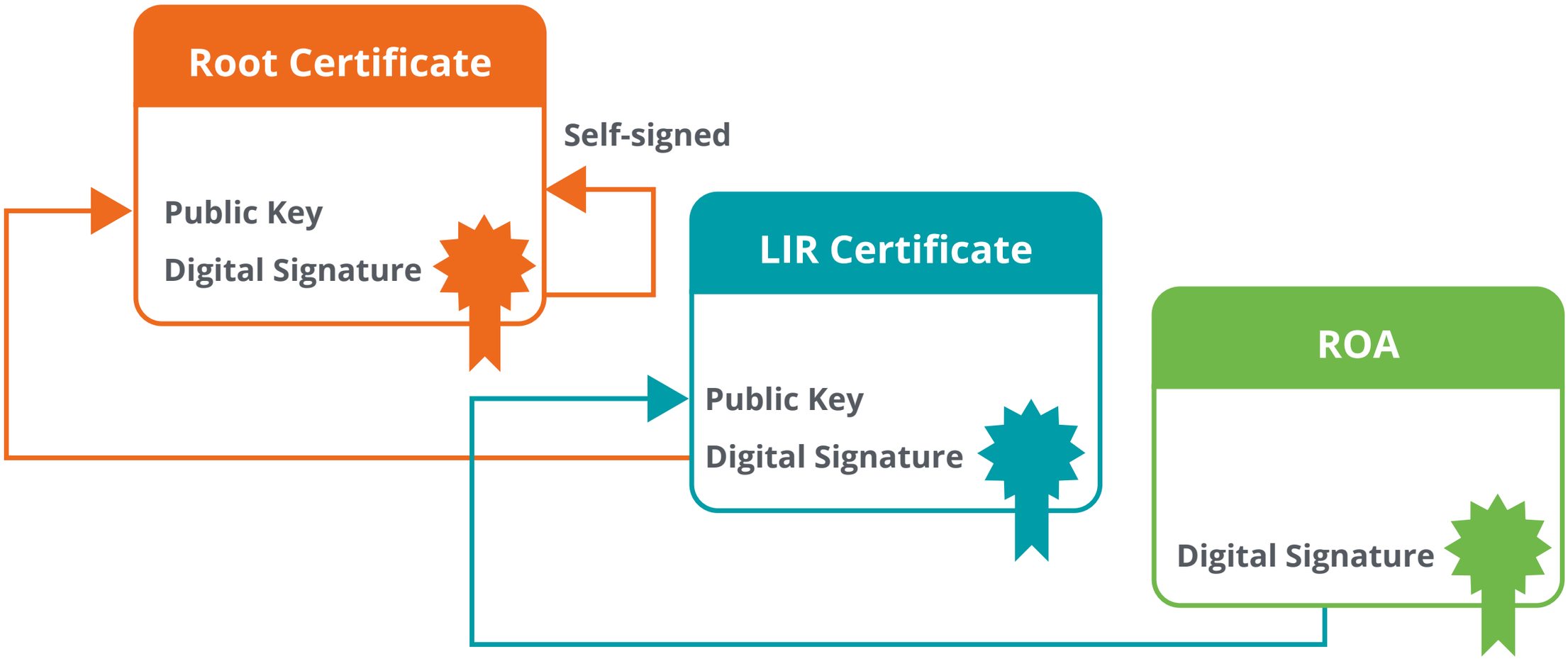
ROA Validation Process



ROA Validation Process

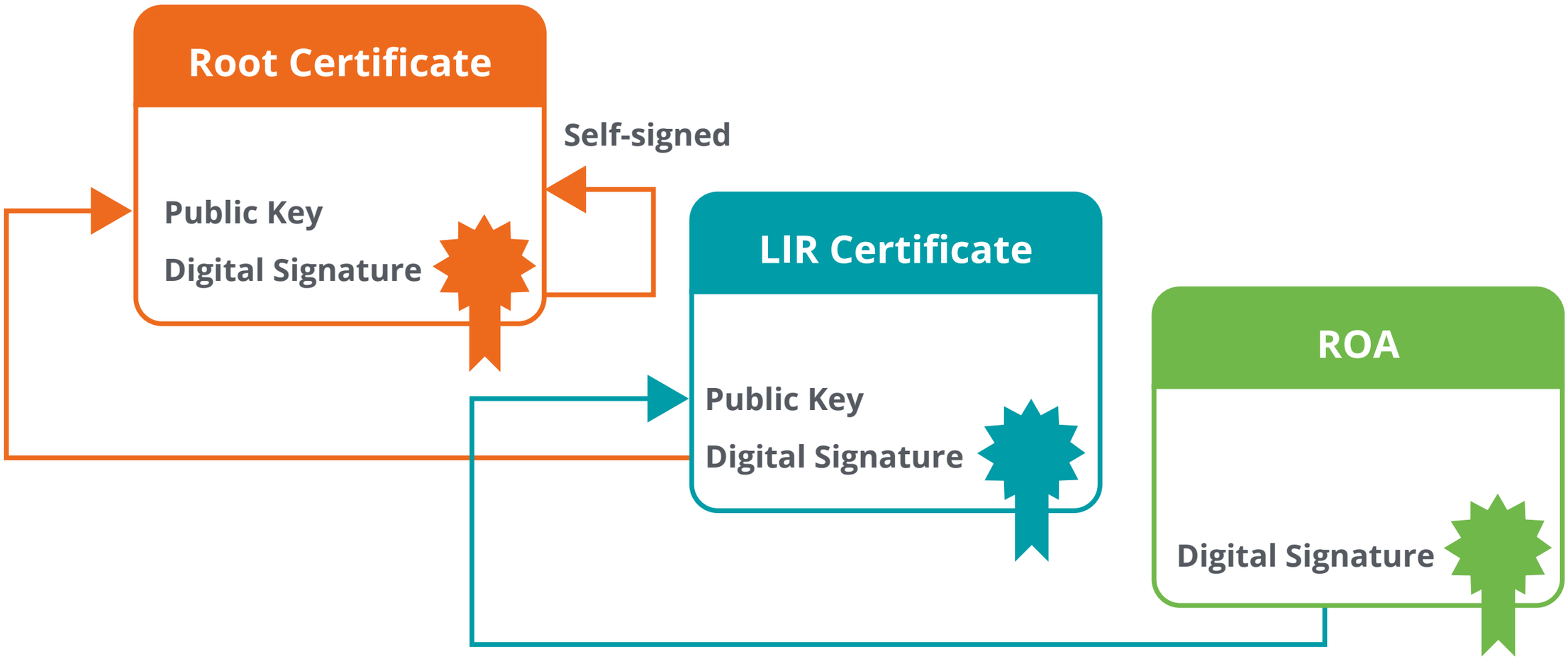


ROA Validation Process



ROA Validation Process

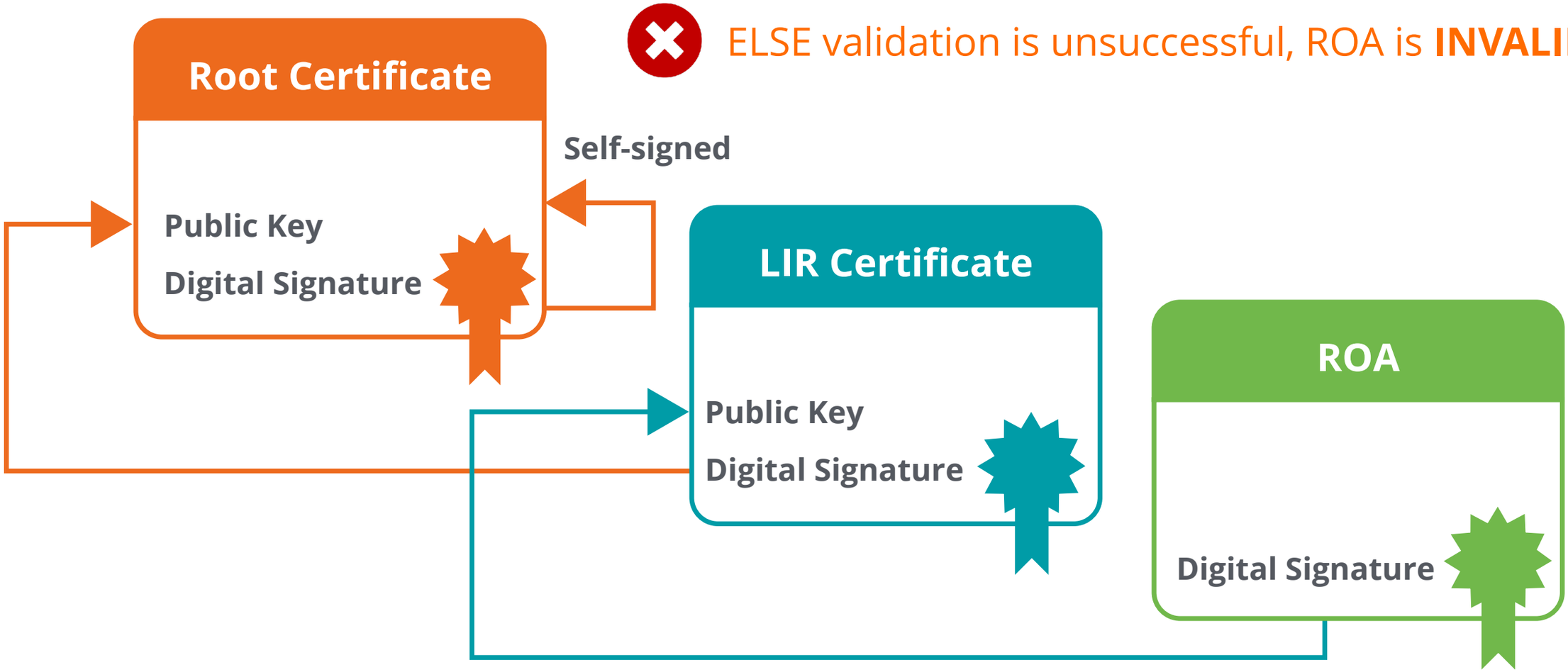
IF chain is complete, it means ROA is **VALID!**



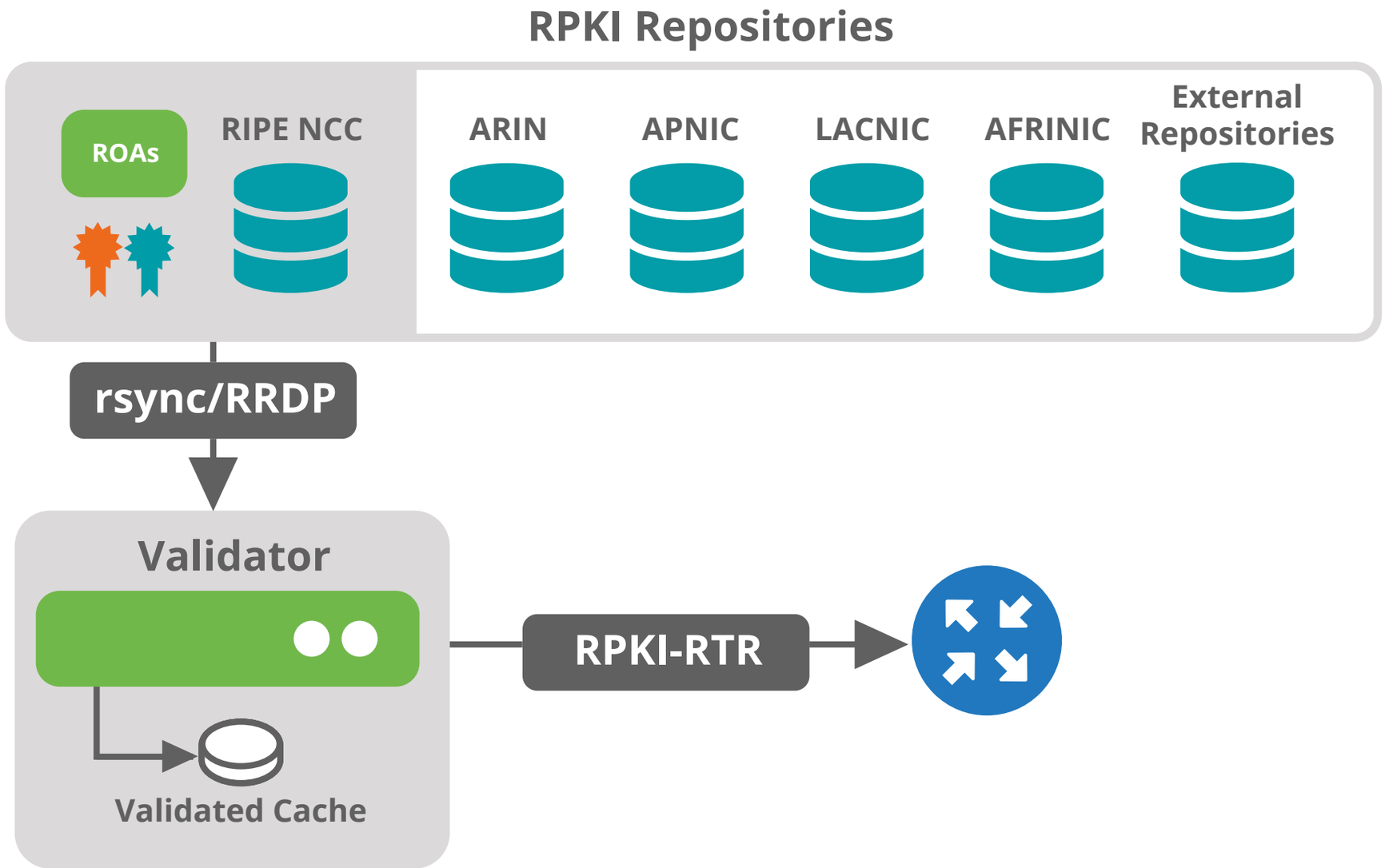
ROA Validation Process

IF chain is complete, it means ROA is **VALID!**

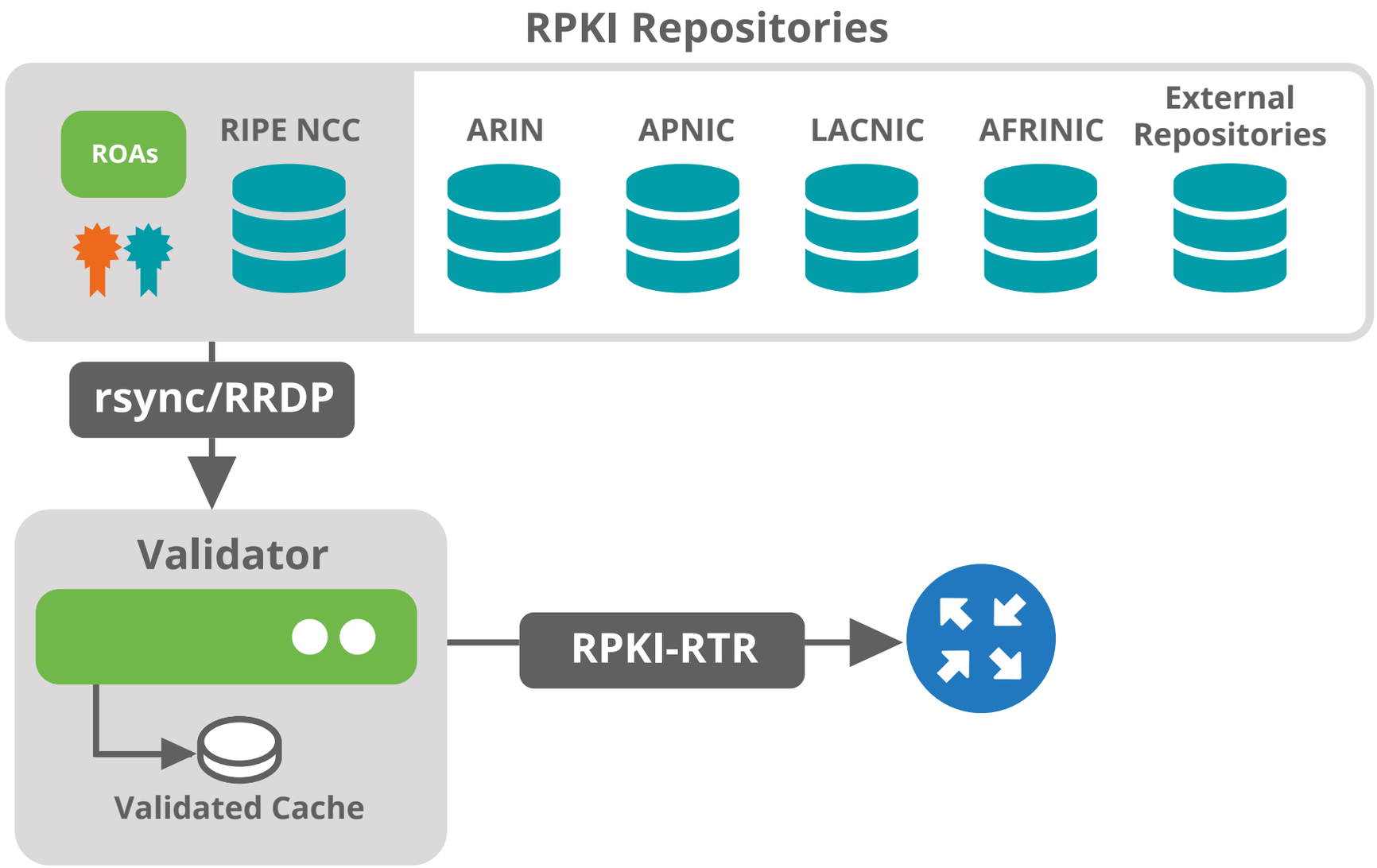
ELSE validation is unsuccessful, ROA is **INVALID!**



Valid ROAs are sent to the router!



Valid ROAs are sent to the router!



Router uses this information to make better routing decisions!

OR

RPKI Validator Options

- **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/

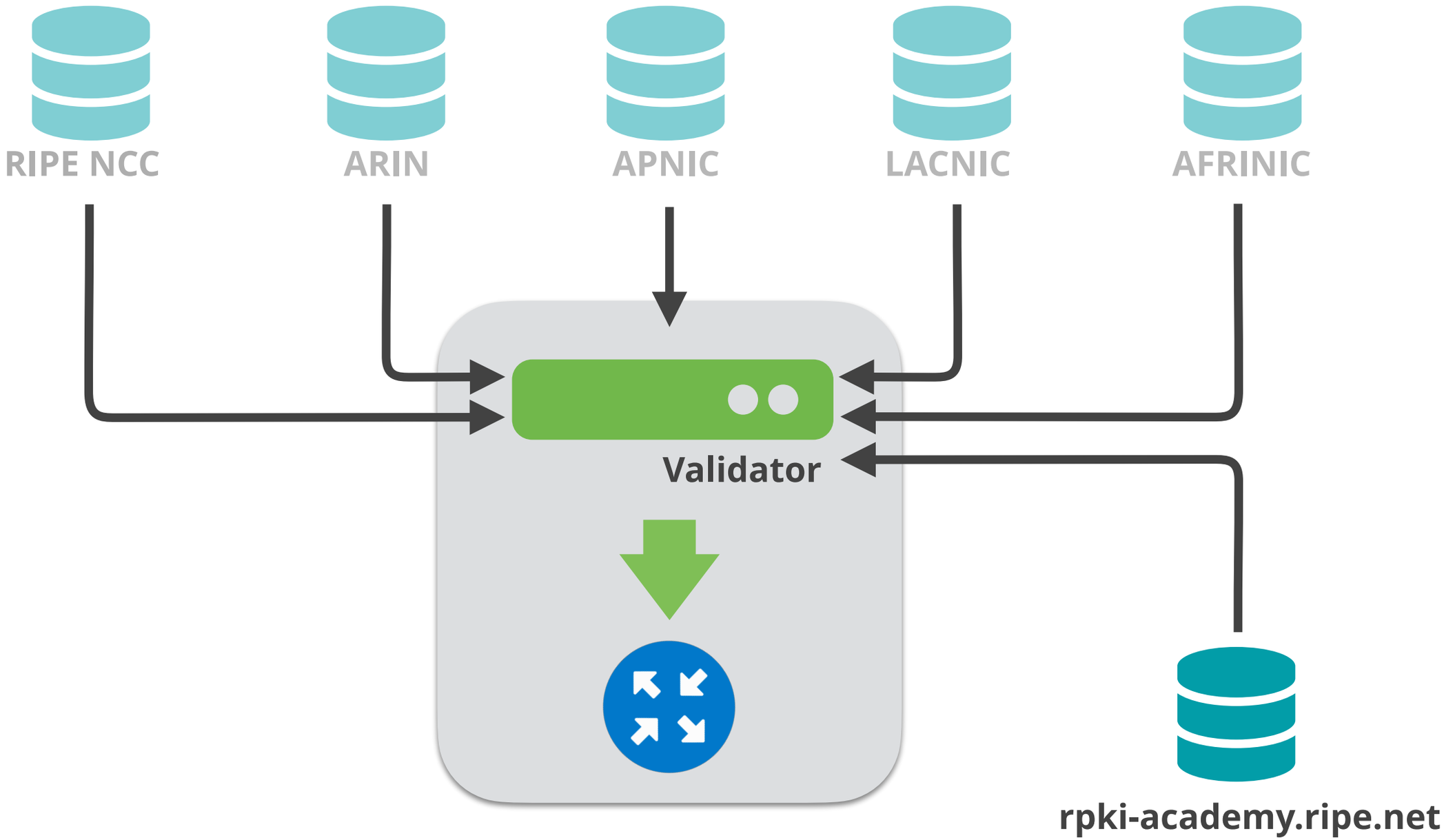
For more info...

https://rpki.readthedocs.io

Running Validators

Activity

Lab Setup


RIPE NCC
ARIN
APNIC
LACNIC
AFRINIC
Validator
rpki-academy.ripe.net

Environment

- Every user has its own dedicated lab setup
  - Server (where validators will run)
  - Router (where BGP announcements will be validated)
- There is a CentOS7 Server in the labs
  - The network configuration is done
  - The validators have been pre-installed

Login to the Labs

- You'll use a new lab setup for RPKI
- Lab supports 15 user and you'll work in pairs
- Go to: workbench.ripe.net
  - Choose the correct lab, trainer will provide info
  - Your login is your number
  - Trainer will provide you the password
- Login info for the server (validator)
  - user/pass : **root/rpki**
- No password required to login to the router!

Running Validators

- You will run the following validators on the server:
  - Routinator (0.12.1)
  - FORT (1.5.3)
- After running validators, configure the correct TALs
  - First, do the standard initialisation for 5 production TALs
  - Then, remove those TALs and replace with the TAL of rpki-academy.ripe.net
- Check that validators are up and running

Install Testbed TAL and start it

```
[root@validator ~]# cd /var/lib/routinator/tals/
[root@validator tals]# wget https://rpki-academy.ripe.net/testbed.tal
[root@validator tals]# cd
```

```
[root@validator ~]# systemctl enable --now routinator
Created symlink from /etc/systemd/system/multi-user.target.wants/
routinator.service to /usr/lib/systemd/system/routinator.service.
```

Check the status and VRPs

```
[root@validator ~]# curl -s http://localhost:8323/status
version: routinator/0.12.1
serial: 0
…
last-update-done-ago:  PT1.873342638S
last-update-duration:  PT0.047253776S
valid-roas: 110
vrps: 200
rtr-connections: 0 current
http-connections: 1 current, 9 total
…
[root@validator ~]# curl -s http://localhost:8323/csv | grepcidr
2001:ff01::/32
AS101,2001:ff01::/32,32,testbed
```

Initialize the FORT validator

```
[root@validator ~]# fort --init-tals --tal=/etc/fort/tal/
…
Successfully fetched '/etc/fort/tal/afrinic.tal'!
…
Successfully fetched '/etc/fort/tal/apnic.tal'!
Attention: ARIN requires you to agree to their Relying Party
Agreement (RPA) before you can download and use their TAL.
Please download and read https://www.arin.net/resources/mrty
Agreement (RPA) before you can download and use their TAL.
Please download and read https://www.arin.net/resources/manage/rpki/
rpa.pdf
If you agree to the terms, type 'yes' and hit Enter: yes
…
Successfully fetched '/etc/fort/tal/arin.tal'!
…
Successfully fetched '/etc/fort/tal/lacnic.tal'!
…
Successfully fetched '/etc/fort/tal/ripe-ncc.tal'!
```

Replace the TALs and start it

```
[root@validator ~]# cd /etc/fort/tal/
[root@validator tal]# rm -f *.tal
[root@validator tal]# wget https://rpki-academy.ripe.net/testbed.tal
[root@validator tal]# cd
```

```
[root@validator ~]# systemctl enable --now fort
Created symlink from /etc/systemd/system/multi-user.target.wants/
fort.service to /usr/lib/systemd/system/fort.service.
```

Check the status

- FORT will not start RTR server before it does the validation for the first time.
- It listens on port **323** by default.
- Configuration is in **/etc/fort/config.json**

To check whether FORT is listening

```
[root@validator ~]# ss -tlnp | grep fort
LISTEN     0      128    100.64.1.1:323                 *:*
users:(("fort",pid=1009,fd=4))
```

See the logs

```
[root@validator ~]# journalctl -u fort -f
Aug 12 13:33:59 validator fort[9708]: INF: Attempting to bind socket
to address '100.64.1.1', port '323'.
Aug 12 13:33:59 validator fort[9708]: INF: Success; bound to address
'100.64.1.1', port '323'.
Aug 12 13:33:59 validator fort[9708]: WRN: First validation cycle has
begun, wait until the next notification to connect your router(s)
Aug 12 13:33:59 validator fort[9708]: INF: Starting validation.
Aug 12 13:34:00 validator fort[9708]: INF: Checking if there are new
or modified SLURM files
Aug 12 13:34:00 validator fort[9708]: INF: Applying configured SLURM
Aug 12 13:34:00 validator fort[9708]: INF: Validation finished:
Aug 12 13:34:00 validator fort[9708]: INF: - Valid ROAs: 200
Aug 12 13:34:00 validator fort[9708]: INF: - Valid Router Keys: 0
Aug 12 13:34:00 validator fort[9708]: INF: - Serial: 1
Aug 12 13:34:00 validator fort[9708]: INF: - Real execution time: 1
secs.
Aug 12 13:34:00 validator fort[9708]: WRN: First validation cycle
successfully ended, now you can connect your router(s)
<Press Ctrl+C to exit>
```

Check the VRPs

```
[root@validator ~]# grepcidr 2001:ff01::/32 /var/lib/fort/roas.csv
AS101,2001:ff01::/32,32
```

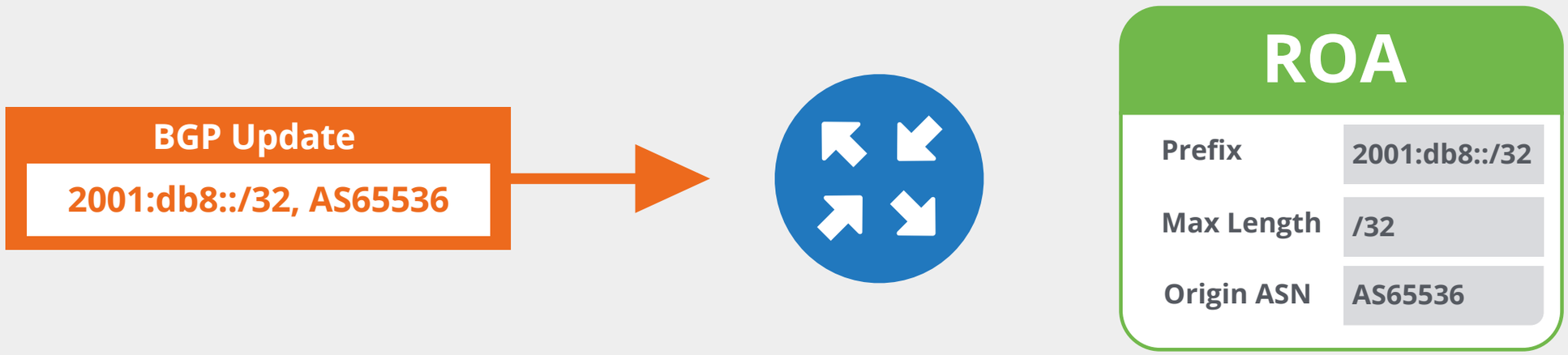
Questions

RPKI Validation

Validating BGP Announcements

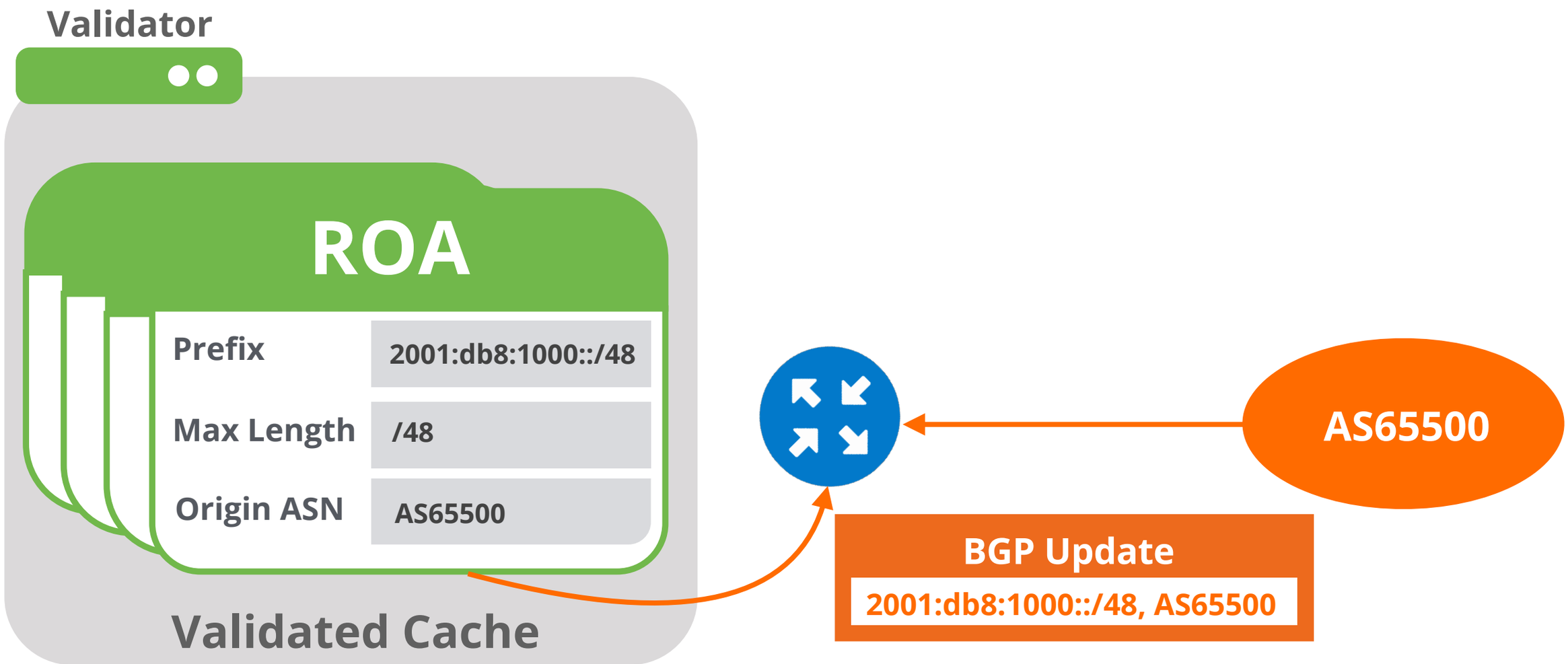
BGP Origin Validation (BGP OV)

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

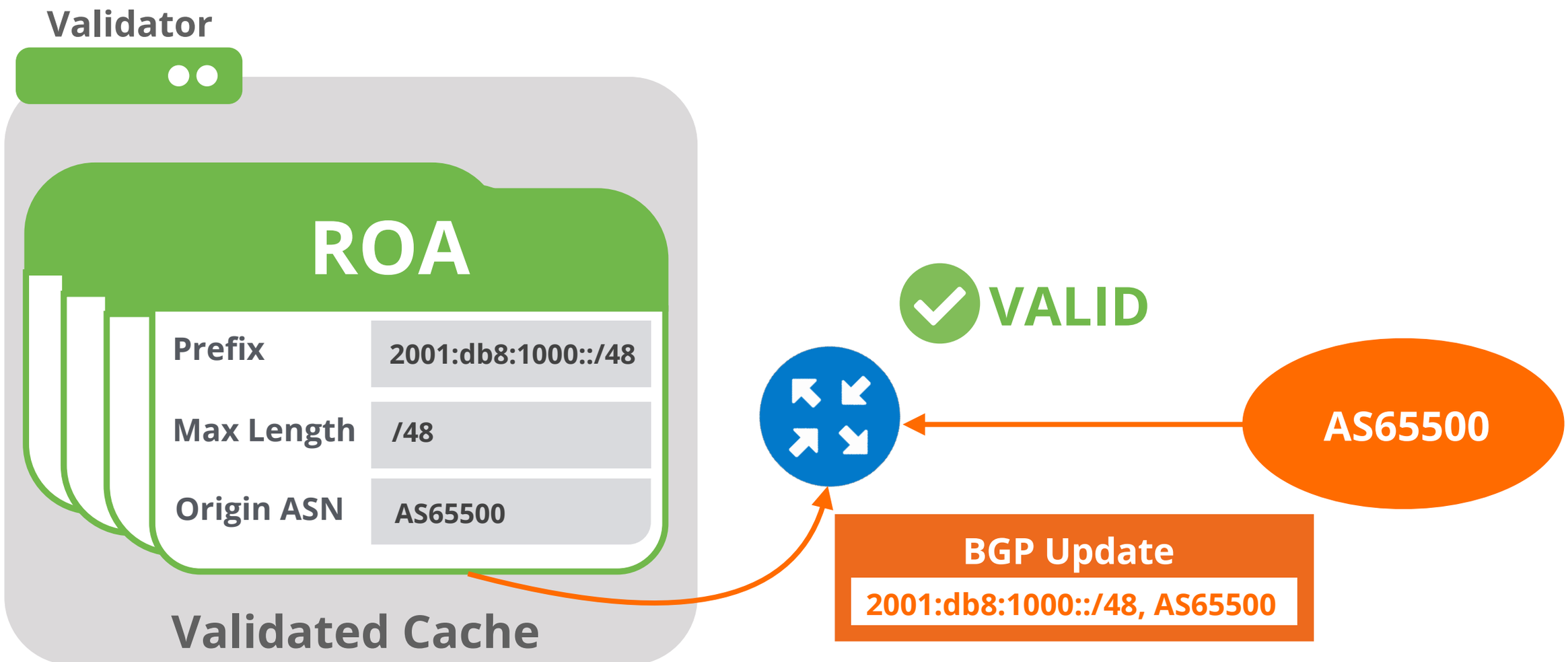


RFC#6811-BGP Prefix Origin Validation https://datatracker.ietf.org/doc/html/rfc6811

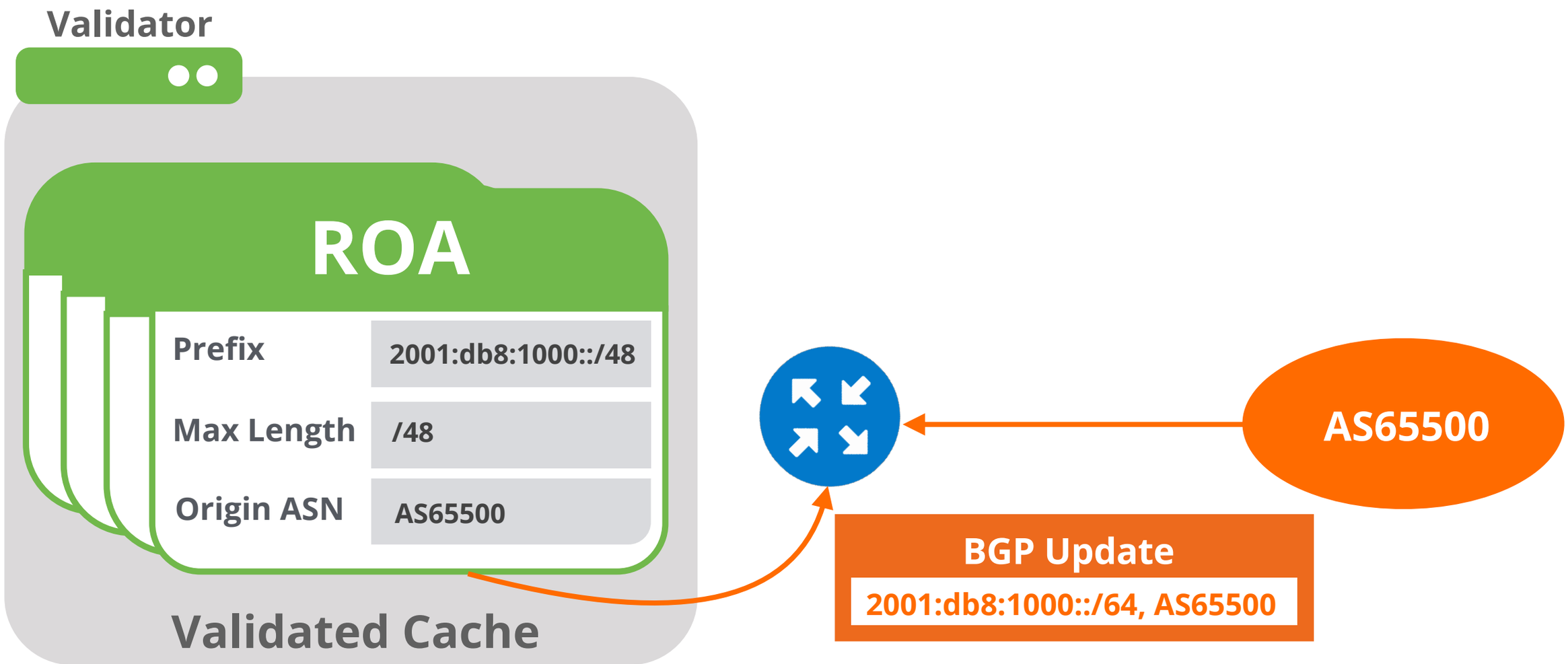
How does RPKI validate the origin?



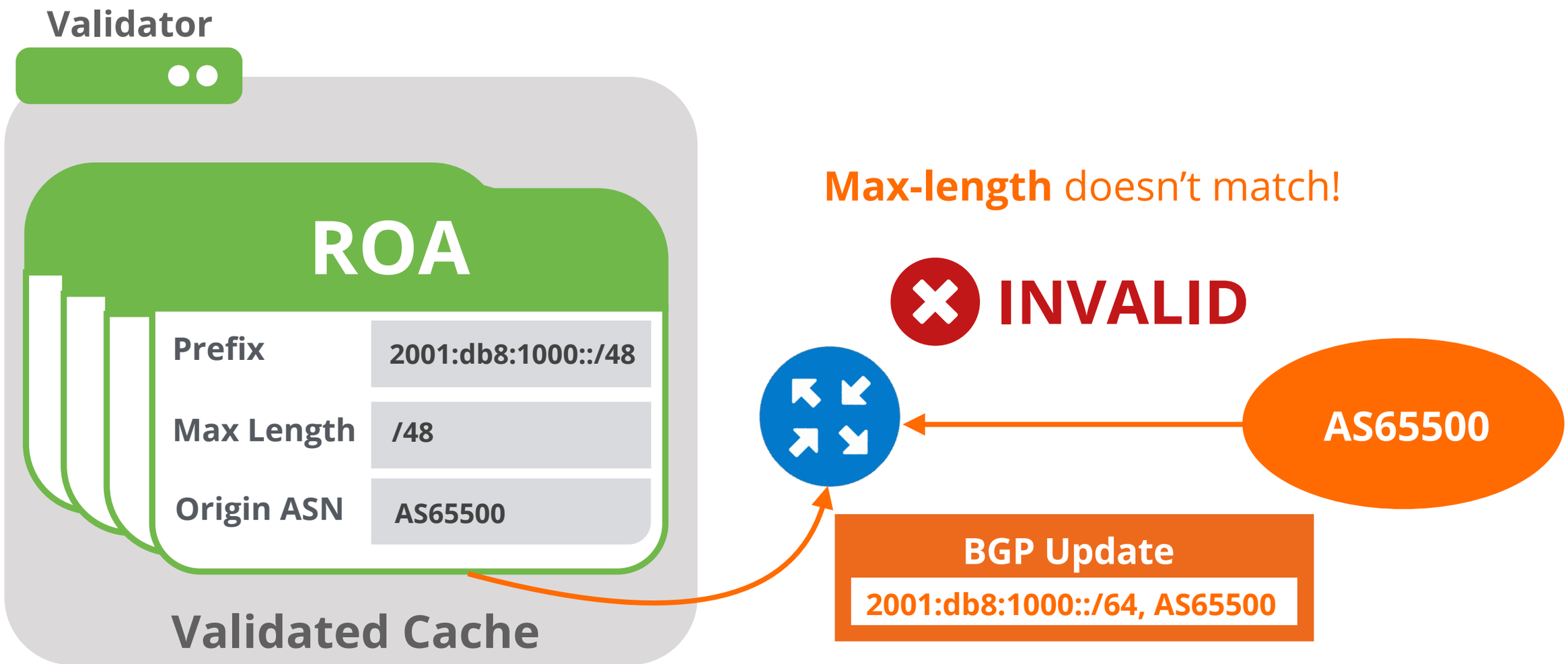
How does RPKI validate the origin?



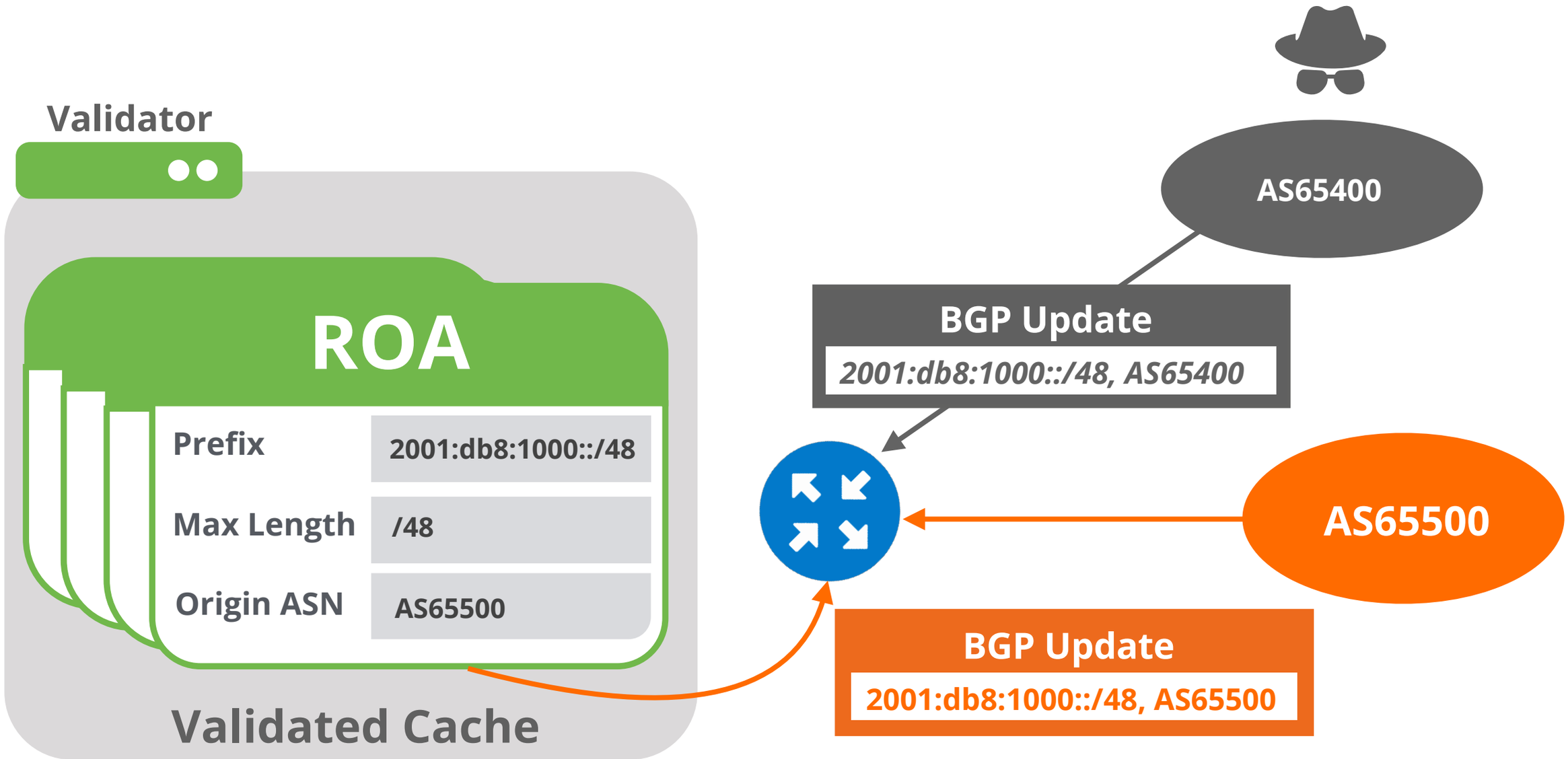
How does RPKI validate the origin?



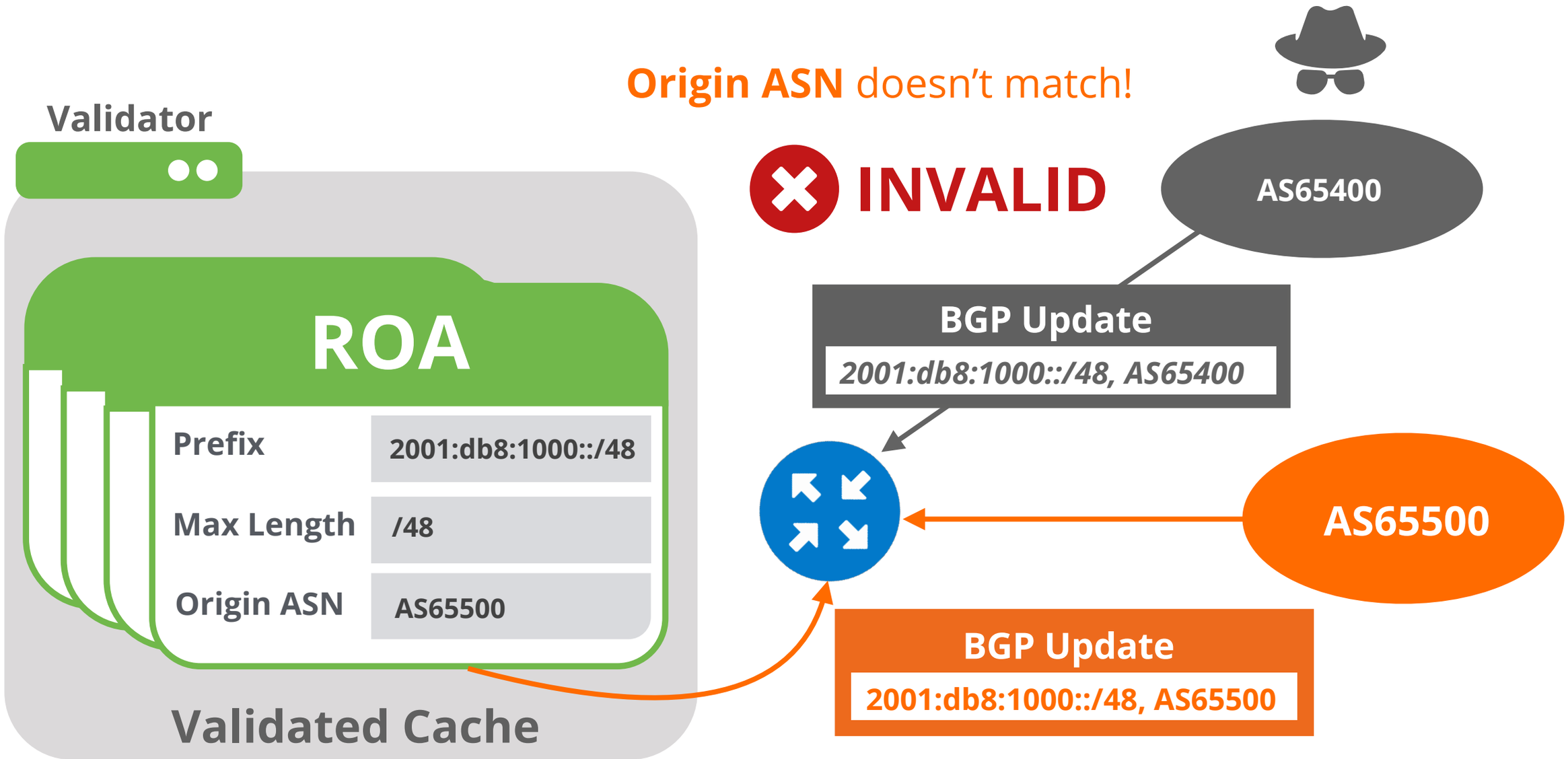
How does RPKI validate the origin?



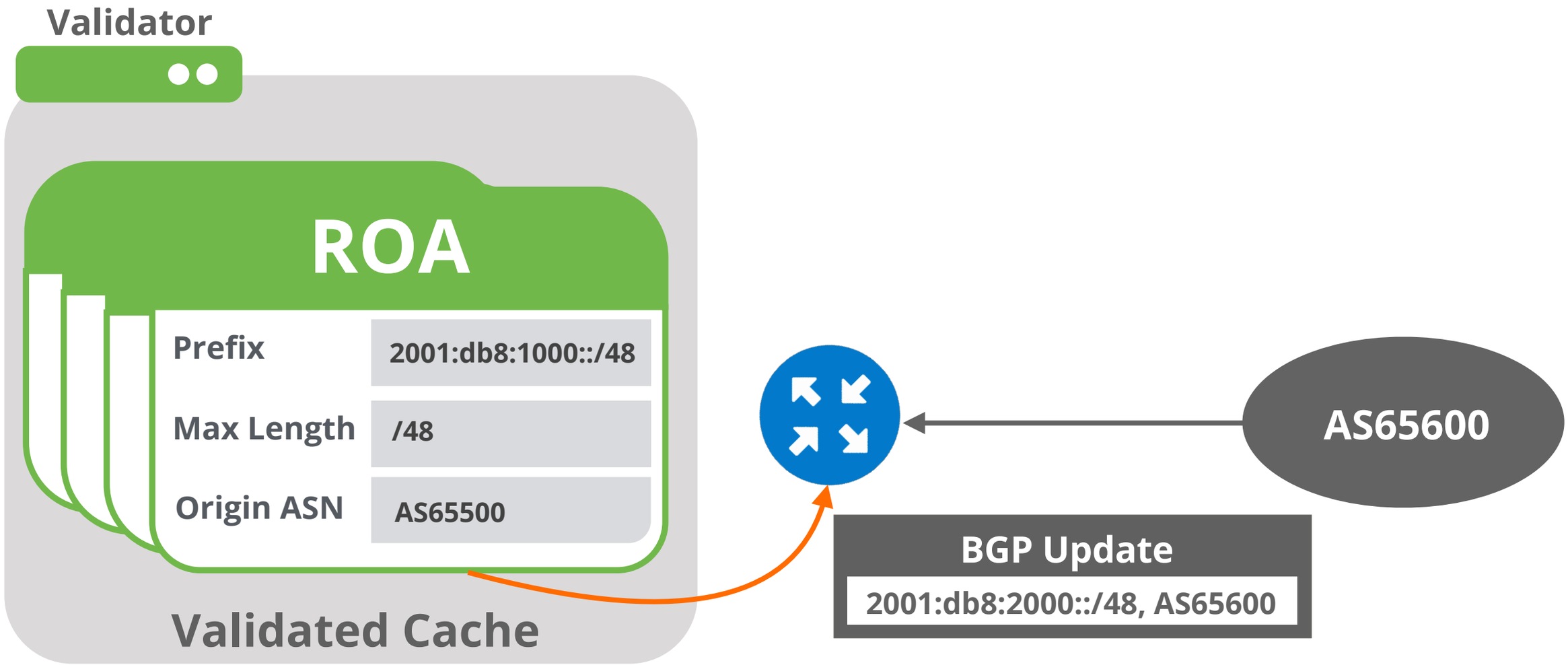
How does RPKI validate the origin?



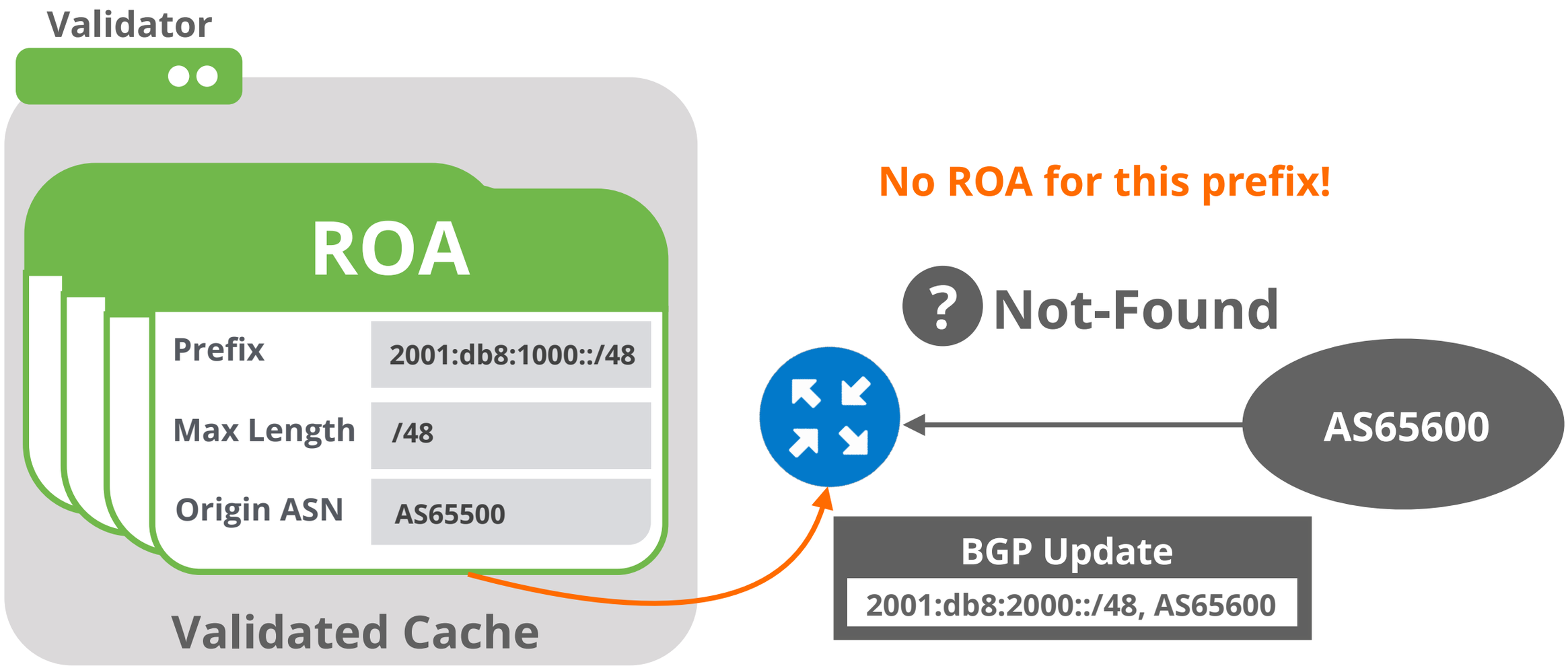
How does RPKI validate the origin?



How does RPKI validate the origin?



How does RPKI validate the origin?



After Validating…

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

Valid → Accept the prefix (set higher local preference)

Invalid → Discard the prefix

NotFound → Accept the prefix (may set lower local preference)

After Validating…

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

Valid → Accept the prefix (set higher local preference)

Invalid → Discard the prefix

NotFound → Accept the prefix (may set lower local preference)

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

Discarding BGP Invalids

- For BGP origin validation (BGP OV) to achieve its goal…
  - Invalids should be dropped!
- Tag the invalids with a BGP communities
  - or set lower local preference (not a long term solution)
- After analysing the effect, you can start dropping invalids

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

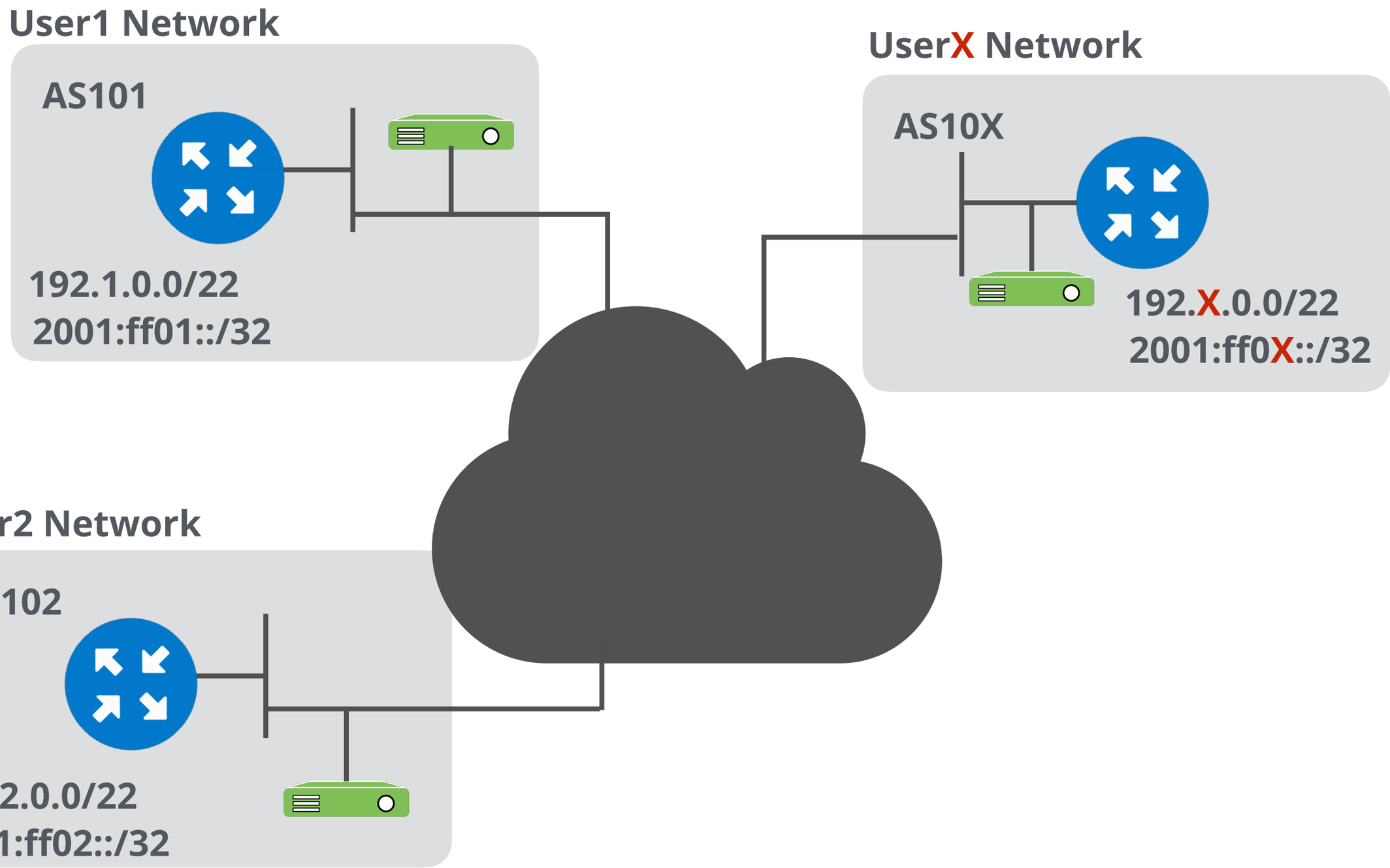
BGP Origin Validation with RPKI

Activity

Goals

- Validate the origin of BGP announcements
  - Check RPKI validation states, Valid, Invalid, Not Found
- Discard Invalid BGP announcements

Network Diagram



Lab Steps

- **Step-1:** Set up validator connection
  - Configure RPKI-RTR protocol on your router
  - Use both validators you run in the previous exercise
  - Remember that Routinator is running on port **3323** and Fort is on **323**
- **Step-2:** Check RPKI prefix table
  - Verify that you download valid ROAs
- **Step-3:** Originate BGP Hijack
  - Announce another user's IPv4/IPv6 address blocks (i.e. for user **X**, you can announce the prefix of user (**X+1**)) (or just ask for your neighbor's prefix)
  - for user **X** , 192.**X**.0.0/22 & 2001:ff**XX**/32 ip address blocks are allocated

Lab Steps

- **Step-4:** Check validation result
  - Check RPKI validation state for any user's address block (except yours and your neighbor's)
- **Step-5:** Originate the following routes that has no ROA in RPKI repository
  - For IPv4, 33.33.**X**.0/24
  - For IPv6, 2001:ff33:**X**::/48
- **Step-6:** Check validation result for the prefixes in Step-5
- **Step-7:** Discard BGP invalids

Step-1: Set up Validator Connection

- Configure validators as “‘RPKI servers” on the router
  - Router talks to validator via RPKI-RTR (RPKI to Router Protocol)

```
(config)# conf t
(config)# router bgp 1XX
(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

Step-2: Check RPKI prefix table

- Verify the connection to the RPKI Validator

```
U1_Router#show ip bgp rpki servers
BGP SOVC neighbor is 100.64.1.1/323 connected to port 323
Flags 64, Refresh time is 300, Serial number is 80, Session ID is
31990
.
.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 100.64.1.2, Local port: 31795
Foreign host: 100.64.1.1, Foreign port: 323 ⟶ FORT
.
.
BGP SOVC neighbor is 100.64.1.1/3323 connected to port 3323
Flags 64, Refresh time is 300, Serial number is 0, Session ID is 31627
.
.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 100.64.1.2, Local port: 29760
Foreign host: 100.64.1.1, Foreign port: 3323 ⟶ Routinator
```

Step-2: Check RPKI prefix table

- Check VRPs

```
U1_Router#sh ip bgp rpki table
90 BGP sovc network entries using 14400 bytes of memory
90 BGP sovc record entries using 2880 bytes of memory

Network                Maxlen  Origin-AS  Source  Neighbor
10.1.1.0/24            24      201        0       100.64.1.1/3323
10.1.1.0/24            24      201        0       100.64.1.1/323
10.1.2.0/24            24      301        0       100.64.1.1/3323
10.1.2.0/24            24      301        0       100.64.1.1/323
10.2.1.0/24            24      202        0       100.64.1.1/3323
10.2.1.0/24            24      202        0       100.64.1.1/323
10.2.2.0/24            24      302        0       100.64.1.1/3323
10.2.2.0/24            24      302        0       100.64.1.1/323
10.3.1.0/24            24      203        0       100.64.1.1/3323
10.3.1.0/24            24      203        0       100.64.1.1/323
10.3.2.0/24            24      303        0       100.64.1.1/3323
10.3.2.0/24            24      303        0       100.64.1.1/323
10.4.1.0/24            24      204        0       100.64.1.1/3323
10.4.1.0/24            24      204        0       100.64.1.1/323
10.4.2.0/24            24      304        0       100.64.1.1/3323
--More--
```

Step-3: Originate BGP Hijack

- Announce your neighbour's IPv4&IPv6 prefixes

```
(config)# ip route 192.(X+1).0.0 255.255.252.0 null0
(config)# ipv6 route 2001:ff(X+1)::/32 null0
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# network 192.(X+1).0.0 mask 255.255.252.0
(config-router-af)# address-family ipv6
(config-router-af)# network 2001:ff(X+1)::/32
```

- Be careful with ipv6 announcement! (i.e.: user 1 will announce 2001:ff02::/32, not 2001:ff2::/32)
  - Remember ipv6 address notation! Each field has 4 hex digits, and you can not omit leading zeros!

Step-4: Check Validation Result

- Check RPKI validation states for the routes in BGP table

```
# show ip bgp
# show ip bgp ipv6 unicast
```

- Which routes are RPKI Valid (V) and which ones are Invalid (I)? Why?
- Do you see any RPKI Not Found routes?

Step-5: Routes with no ROA

- Originate the following routes that has no ROA in RPKI repository
  - for IPv4, 33.33.**X**.0/24 & for IPv6, 2001:ff33:**X**::/48

```
(config)# ip route 33.33.X.0 255.255.255.0 null0
(config)# ipv6 route 2001:ff33:X::/48 null0
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# network 33.33.X.0 mask 255.255.255.0
(config-router-af)# address-family ipv6
(config-router-af)# network 2001:ff33:X::/48
```

Step-6: Check Validation Result

- Check RPKI validation state for the routes in BGP table again

```
# show ip bgp
# show ip bgp ipv6 unicast
```

- Do you see RPKI Not Found routes now?
- Why are they tagged as RPKI not found?

Step-7: Discard BGP invalids

- Create a route-map on your router
- Route map will
  - set local-preference value to **110** for RPKI “Valid“ and to **70** for RPKI “NotFound “ routes
  - discard RPKI invalids

```
(config)# route-map rpki-accept permit 10
(route-map)# match rpki valid
(route-map)# set local-preference 110
(route-map)# route-map rpki-accept permit 20
(route-map)# match rpki not-found
(route-map)# set local-preference 70
```

Step-7: Discard BGP invalids

- Apply the route map to eBGP session with your upstream on inbound

```
(config)# router bgp 1XX
(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
```

Step-7: Discard BGP invalids

- Clear your BGP session

```
# clear bgp ipv4 unicast 192.168.1.254 in
# clear bgp ipv6 unicast 2002:eeee:ffff::a in
```

- Check your BGP table again

```
# show ip bgp
# show ip bgp ipv6 unicast
```

- Do you see RPKI Invalid routes in your BGP table? If not, why?
- Check the local-preference for RPKI valid and not-found routes in BGP table.

Questions

What else?

Section 11

Where do we go from here ?

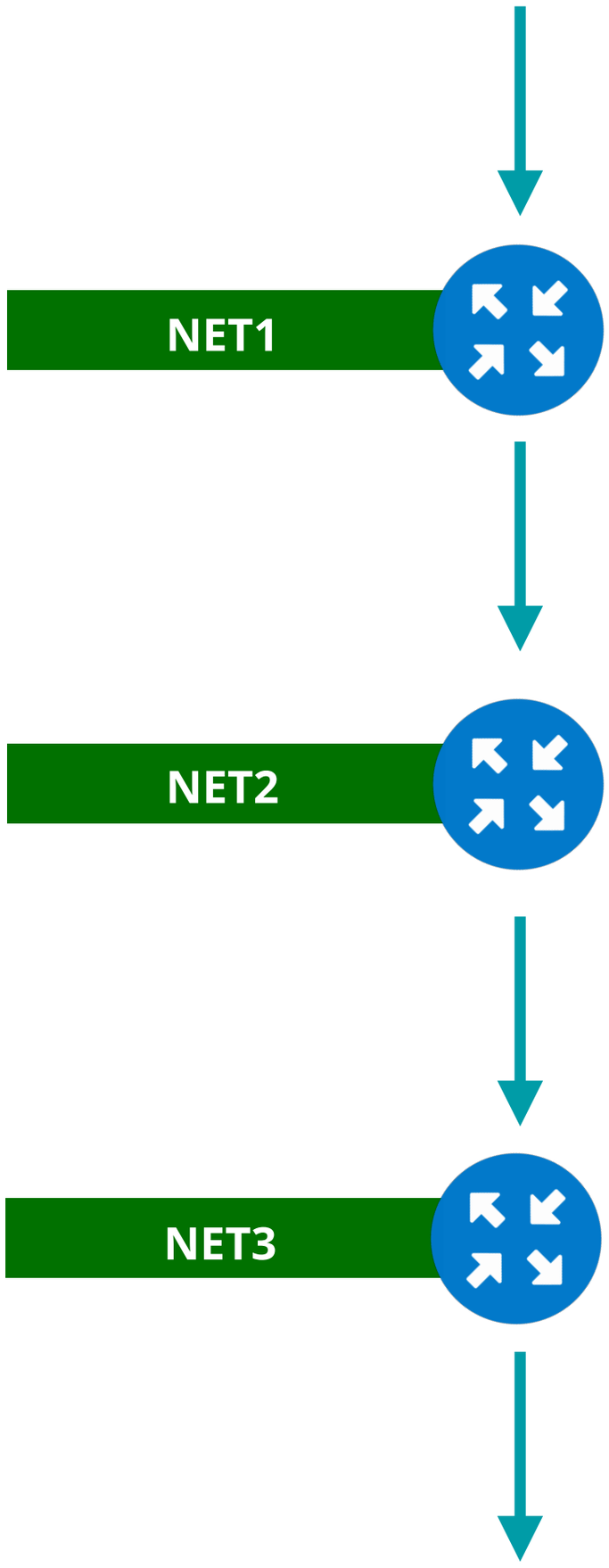
- RPKI is only one of the steps towards full BGP Validation
  - Paths are not validated
- We need more building blocks
  - BGPSec (RFC)
  - ASPA (draft)
  - AS-Cones (draft)

BGPSec

- RPKI does not protect against path redirection attacks
- We need a way to verify the AS-Path of a given BGP Announcement
  - And understand if anyone tampered with the data on the way to our routers

BGPSec Path Validation

- With BGPSec, the AS-Path attribute is cryptographically signed
  - Using the operator's certificate from RPKI
- In order to validate an AS-Path, routers verify the chain of trust of all the signatures of the AS-Path



Network: 192.168.0.0/16
AS Path: NET1, …
BGPSEC: (key1, signature1)

Network: 192.168.0.0/16
AS Path: NET2, NET1, …
BGPSEC: (key1, signature1)
(key2, signature2)

Network: 192.168.0.0/16
AS Path: NET3, NET2, NET1, …
BGPSEC: (key1, signature1)
(key2, signature2)
(key3, signature3)

BGPSec Operations

- Optional, non-transitive BGP path attribute
- Carries digital signatures
- Support is negotiated between routers
  - non BGPSEC router will not be burdened by big UPDATE messages
- Incremental deployment is possible

ASPA

- Additional object in RPKI to define upstreams for a defined ASN
- Provides infrastructure to do lightweight path validation
- Still in draft state

AS-Cones

- Additional objects in RPKI to define
  - Announcements upstream/downstream
  - List of customer ASNs and/or cones
- Similar to AS-Sets in RPSL
- Still in draft state

BGP Tips & Tricks

Section 12

Protect your BGP Routers

- Allow only BGP neighbours to send packets to TCP 179
  - Implement Control Plane Policing (CoPP)
  - Use data plane filters (ACLs) (If CoPP is not supported)
- Limit accepted BGP traffic
- Implement uRPF to mitigate DoS/DDoS attacks

Protect your BGP Sessions

- Authenticate your BGP sessions and ensure the integrity of BGP messages
- MD5
  - Legacy solution, still widely deployed in many networks
  - Not a strong authentication mechanism, obsoleted by TCP-AO
- TCP-AO
  - Supports multiple stronger authentication algorithms
  - Provides better key management and agility
  - Supported by Nokia, Cisco and Juniper
  - No open source implementation yet!

Prevent Route Leaks

- Use ingress policy
  - Tag routes received from eBGP peers
- Use egress policy
  - Utilize the tagged information
- Do not forward routes
  - Received from a transit provider to another transit provider or a lateral peer
  - Received from a lateral peer to another lateral peer or a transit provider

Apply BCP38

- RFC#2827: Network Ingress Filtering
- Use inbound and outbound packet filters to protect your network
  - Outbound: only allow your network source addresses out
  - Inbound: only allow specific ports to specific destinations in
- Principles:
  - Filter as close to the edge as possible
  - Filter as precisely as possible
  - Filter both source and destination where possible

Register your routing in IRR

- Create route, route6 objects in IRR database
- Update your routing registry information regularly
- Create filters based on IRR data
  - Automation relies on the IRR being complete
  - Check your output before using it
- Help others by documenting your policy

Create ROAs in the global RPKI system

- Contribute to routing security
- Protect your prefixes
- To minimize forged origin attacks
  - max-length should not exceed the length of the most specific prefix
  - Or list your prefix and more-specific prefixes explicitly in multiple ROAs (i.e., one ROA per prefix or more-specific prefix)

Validate your BGP Routes

- Implement BGP Origin Validation (BGP OV)
- Use BGP OV results in path selection
  - Accept “Valid” prefixes
  - Accept “Not Found” prefixes and set lower local-pref
  - Never discard “Not-Found” BGP routes!
- What to do with invalids?
  - Tag “invalid” prefixes with BGP communities
  - May set lower local preference (not a long term solution)
  - After analysing the effect, you can start dropping invalids

Check Your Routing

- RIPEstat
  - https://stat.ripe.net/
- IXP Country Jedi
  - https://jedi.ripe.net/latest/
- Bgpmon
  - http://routeviews.org/
  - http://traceroute.org
- RIPE Atlas
  - http://atlas.ripe.net/
- NLNOG Ring
  - http://ring.nlnog.net/
- HE BGP Toolkit
  - http://bgp.he.net/

Questions

We want your feedback!

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

RIPE NCC
Academy

Learn something new today!

academy.ripe.net

RIPE NCC Certified Professionals

https://getcertified.ripe.net/


Ënn
Соңы
An Críoch
پایان
Y Diwedd
Vége
Endir
Finvezh
վերջ
Ende
Koniec
Son
დასასრული
הסוף
Кінець
Finis
Lõpp
Amaia
Tmiem
Liðugt
Kpaj
Sfârşit
Loppu
Slutt
Fund
Kraj
النهاية
Конец
Konec
Τέλος
Fin
Fine
Fí
Край
Pabaiga
Einde
Slut
Fim
Beigas
E1 N1 D2

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