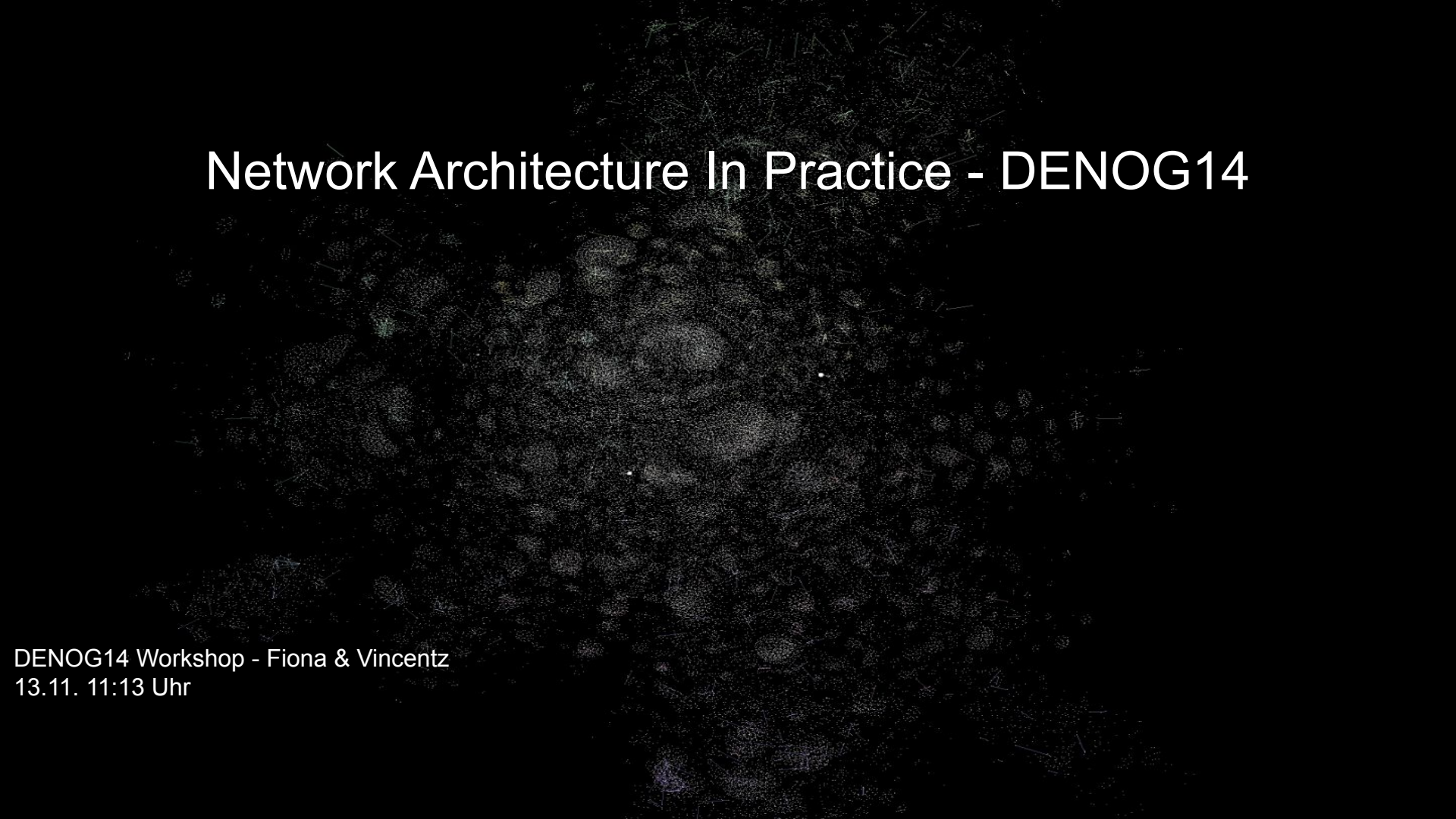
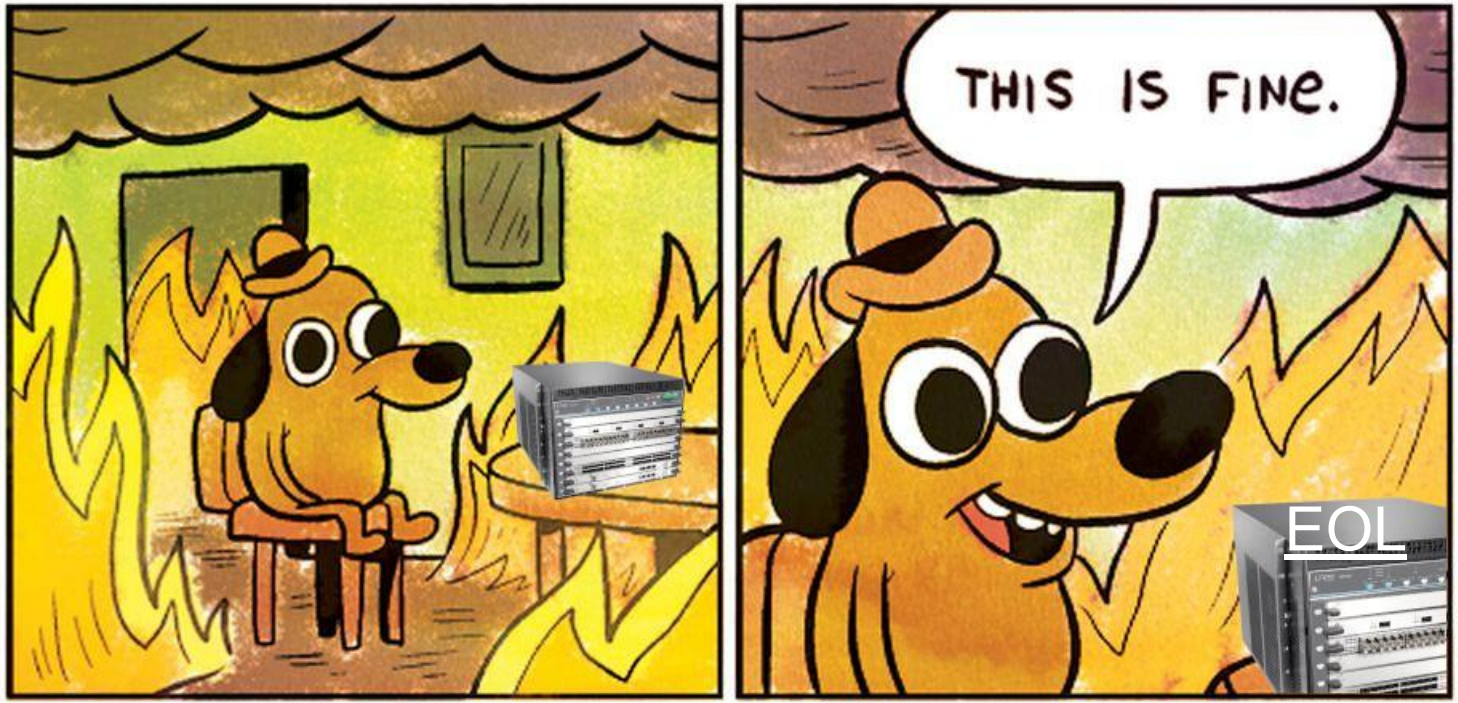
Network Architecture In Practice - DENOG14



DENOG14 Workshop - Fiona & Vincentz
13.11. 11:13 Uhr



OR how to NOT screw up your network

Introducing

Fiona

Network Engineer / SRE

WDZ / Wobcom AS9136
infra.run AS213027
DENOG e.V. Vorstand

Vincentz

Head of Network

Inter.link - AS5405 / AS25291
(former SysEleven Network)
BCIX e.V. Beirat

Who is this workshop for?

If you've configured a bgp session before, this workshop is for you.

Goal of the workshop?

Theoretical background as far as necessary ...

Experience and best practices from real life

Housekeeping

- Thank you for your participation and attention!
- All slides will be available as PDF later (yes, also the config).
- Config examples are available for Junos (more to come)
- Use of any Config is at your own responsibility!
- Please do not use anything in "Headless 🐔 Mode".
- Please write down your questions for the Q&A.
- Quiz and Q&A after Session 3!

Time schedule

Design Basics and Guidelines

- 11:00 - 11:45 = Session 1
- 11:45 - 12:00 = 15 min Break

Route Filtering

- 12:00 - 12:45 = Session 2
- 12:45 - 13:30 = 45 min Lunch 

Forwarding Filtering

- 13:30 - 14:15 = Session 3
- 14:15 - 15:00 = Quiz + Q&A

Agenda - Session 1

- **Interconnection - ~10 min**
  - The Internet
  - Interconnection
  - Communities
  - Route Aggregates

- **Network Design - ~10 min**
  - Topology / Strategy (Island vs. Backbone)
  - iBGP
  - IGP

- **Traffic Engineering - ~25 min**
  - Best Path Selection
  - How about Local Pref/MED/More Specifics
  - Do not announce/redistribute to, prepend to?
  - Please prepend to peer X, Do not announce to, Do not redistribute to
  - Origin Community tagging (Like Country, City, IXP, Upstream)
  - Multipathing
  - Remote Peering / Impact of Remote Peering?

Bio Break - 15 min

Agenda - Session 2

- **RIPE, RPKI & PeeringDB - Basics - ~5 min**
  - RPKI ROA Signing
  - DB Housekeeping (AS-SET, ROUTE)
  - DB Query Hints
  - Inetnum Automation (RIPE Updater)
  - PeeringDB
  - MANRS

- **Route Filtering - Building Blocks - ~25 min**
  - Filtering BCPs
    - RPKI, PeeringLANs, Prefixlen
    - Bogon ASes/Prefixes, AS Pathlen, Tier1 ASNs
    - My Prefixes, Direct Peer Prefixes
    - Communities, Prefix Limit
    - AS Path, Prefix List
  - Maintenance Switch, Graceful shutdown

- **Route Filtering - Policy Building - ~15 min**
  - Buildings blocks for a public peer
  - Out Policy Example

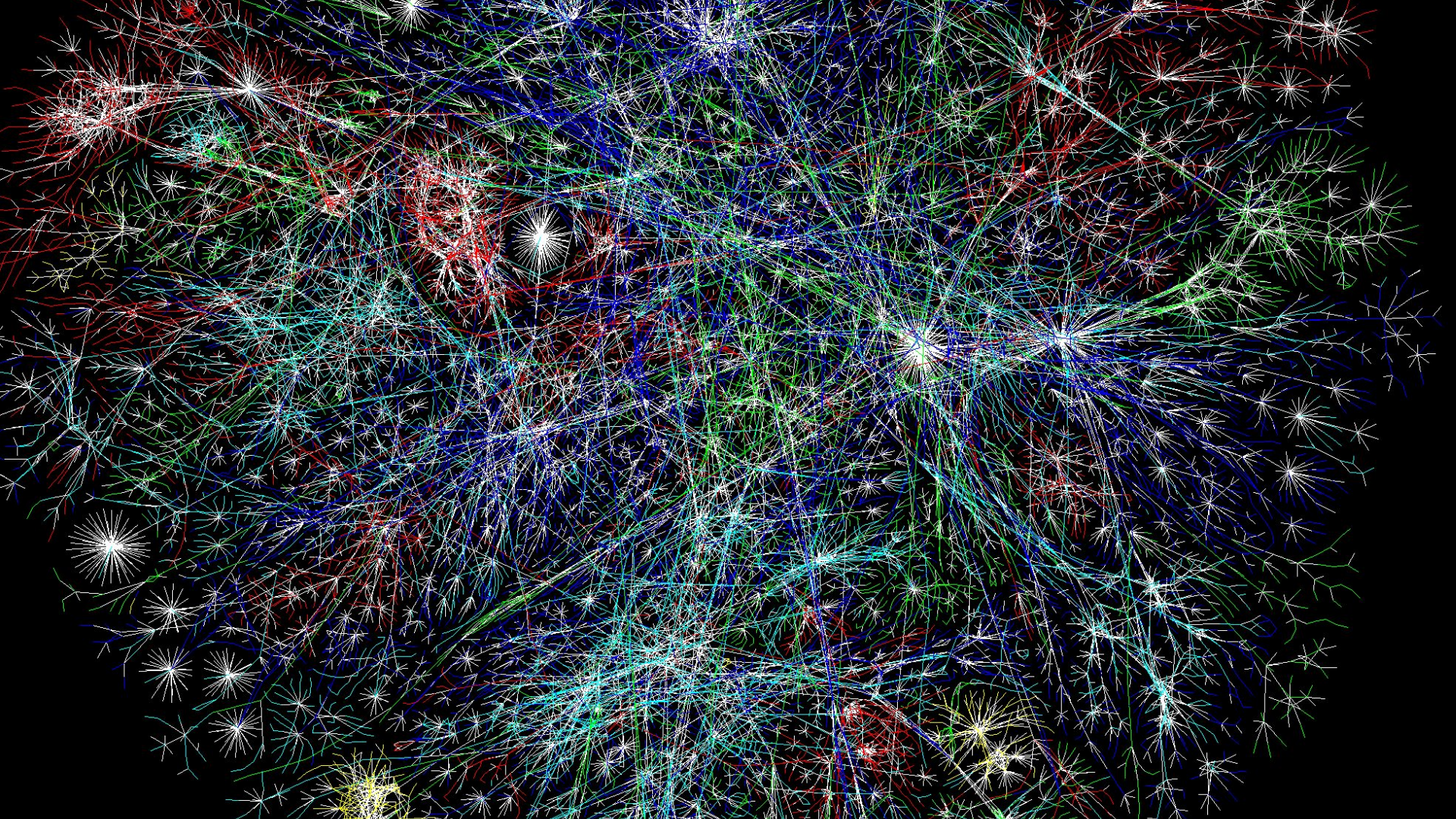
Lunch - 45 min

Agenda - Session 3

- **Protecting the Control Plane**
  - Control Plane vs. Forwarding Plane, Punting, DoS
  - Automatched/Wildcard Apply Groups/Lists
  - lo0 Protection ACL for Juniper
- **Forwarding Plane Filtering**
  - Why filter?
  - BCP 38
  - Where to filter what?
  - Traffic Ingress filtering
    - Customer
    - Edge/Peering/Upstream
    - Internal “Services”
  - Traffic Egress filtering
  - Reverse Path Filtering
    - Loose vs. Strict
    - Where? where not?
    - By feature or statically generated

Quiz + Q&A - 45 min

Session 1
The fellowship of the (token) ring



Interconnection

Types

- Transit/Upstream
  - Access to the whole Internet™
  - “Full Table”
- Peering
  - Exchanging traffic between both parties (and their downstreams)
  - Usually settlement-free
  - Flavors:
    - Public (Internet Exchange)
    - Private (PNI)
- Customer/Downstream
  - We announce their routes to the world
  - We give them our Full Table

Why is peering important?

- Without (direct) peering, the Internet would be a rather hierarchical structure
- Peering shortens the paths (latency)*
- Reduces the "points of failure" or its impact
- Simplifies capacity planning
- Saves money*

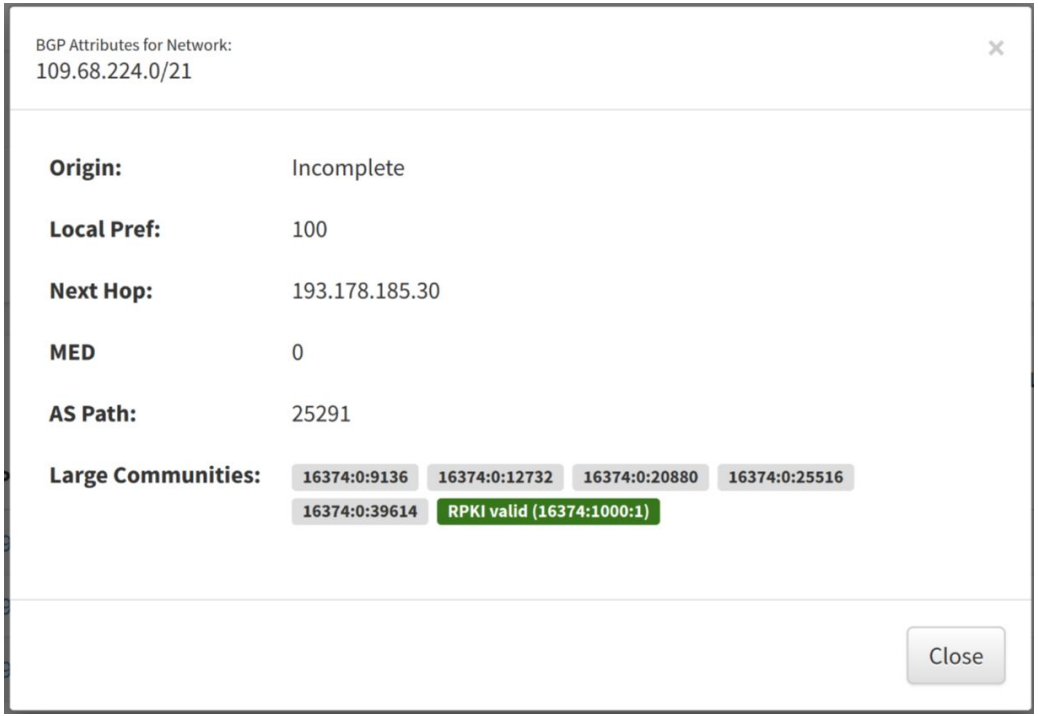
*** hopefully**

Route Export

	To Customer	To Peer	To Transit
Own Aggregates	Yes	Yes	Yes
From Customer	Yes	Yes	Yes
From Peer	Yes	No	No
From Upstream	Yes	No	No

Communities

Sticky notes on routes

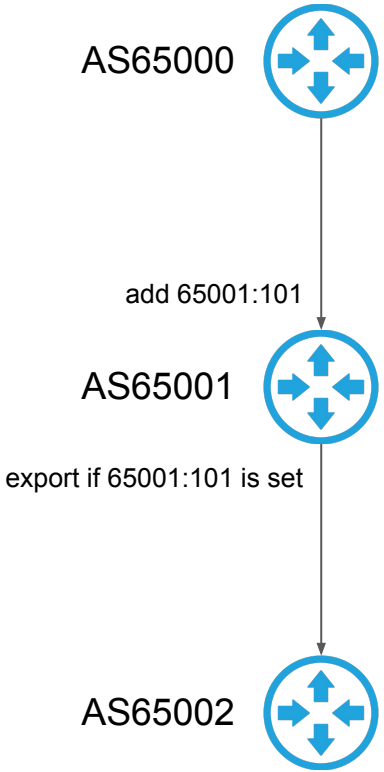


Communities

Control export on ingress

Example:

<yourASN>:100	To Internal (iBGP Export)
<yourASN>:101	To Upstream
<yourASN>:102	To Customer
<yourASN>:103	To Peer



Ingress Policy Example

```
policy-statement CUSTOMER_IMPORT {
      then {
            community add TO_INTERNAL;
            community add TO_UPSTREAM;
            community add TO_CUSTOMER;
            community add TO_PEERING;
            accept;
      }
}
```

Egress Policy Example

```
policy-statement UPSTREAM_EXPORT {
  term EXPORT_TO_UPSTREAM {
      from community TO_UPSTREAM;
      then {
              next-hop self;
        accept;
      }
      }
      then reject;
}
```

Route Aggregates

Our prefixes that we announce to the world

Route Aggregates

```
[edit routing-options aggregate]
route 192.0.2.0/24 {
    community [ TO_INTERNAL TO_PEERING TO_CUSTOMER TO_UPSTREAM ];
    discard;
}
route 203.0.113.0/24 {
    community [ TO_INTERNAL TO_PEERING TO_CUSTOMER TO_UPSTREAM ];
    discard;
}
```

Route Aggregate Placement

- If the aggregates are gone, we are offline.
- If a router is isolated and continues to announce aggregates to the outside, we blackhole parts of the traffic.

👉 Aggregates should reside as closely to the services as possible

👉 If necessary, split up the networks so that the respective locations work autonomously

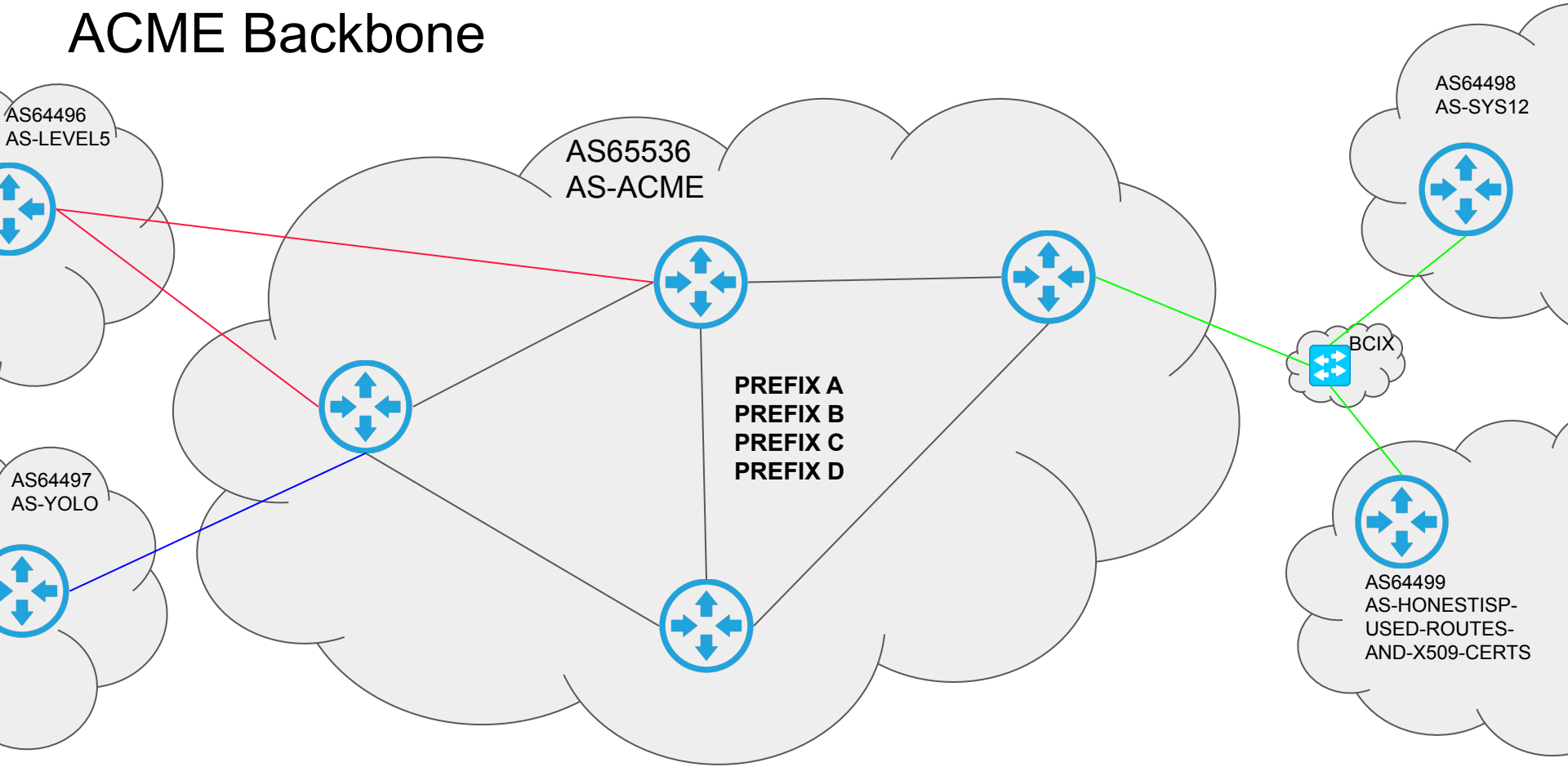
Topology

Backbone vs. Island

Backbone

- Locations are connected to each other
- Announce all aggregates everywhere
- Usually "hot potato routing”

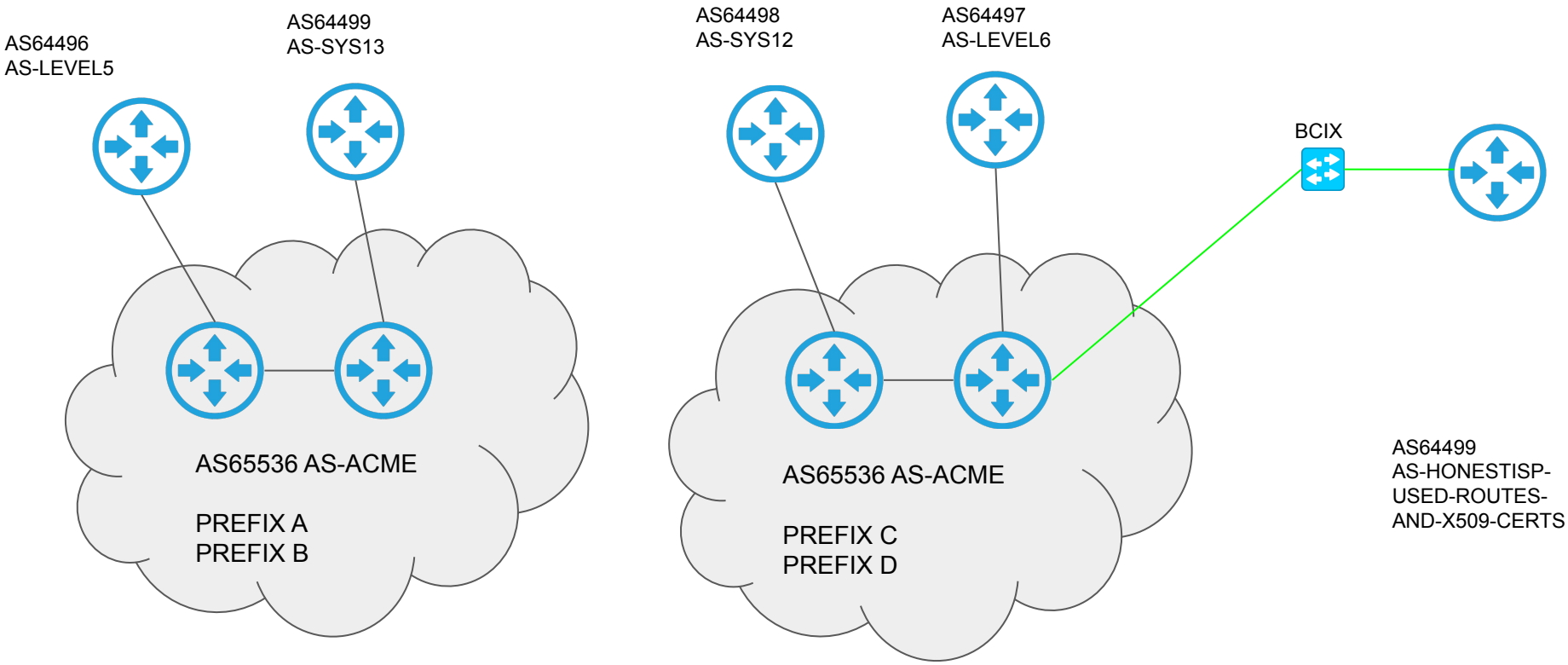
ACME Backbone



Island

- Locations are not connected to each other
- Announce only local aggregates
- Connectivity between islands "through the internet

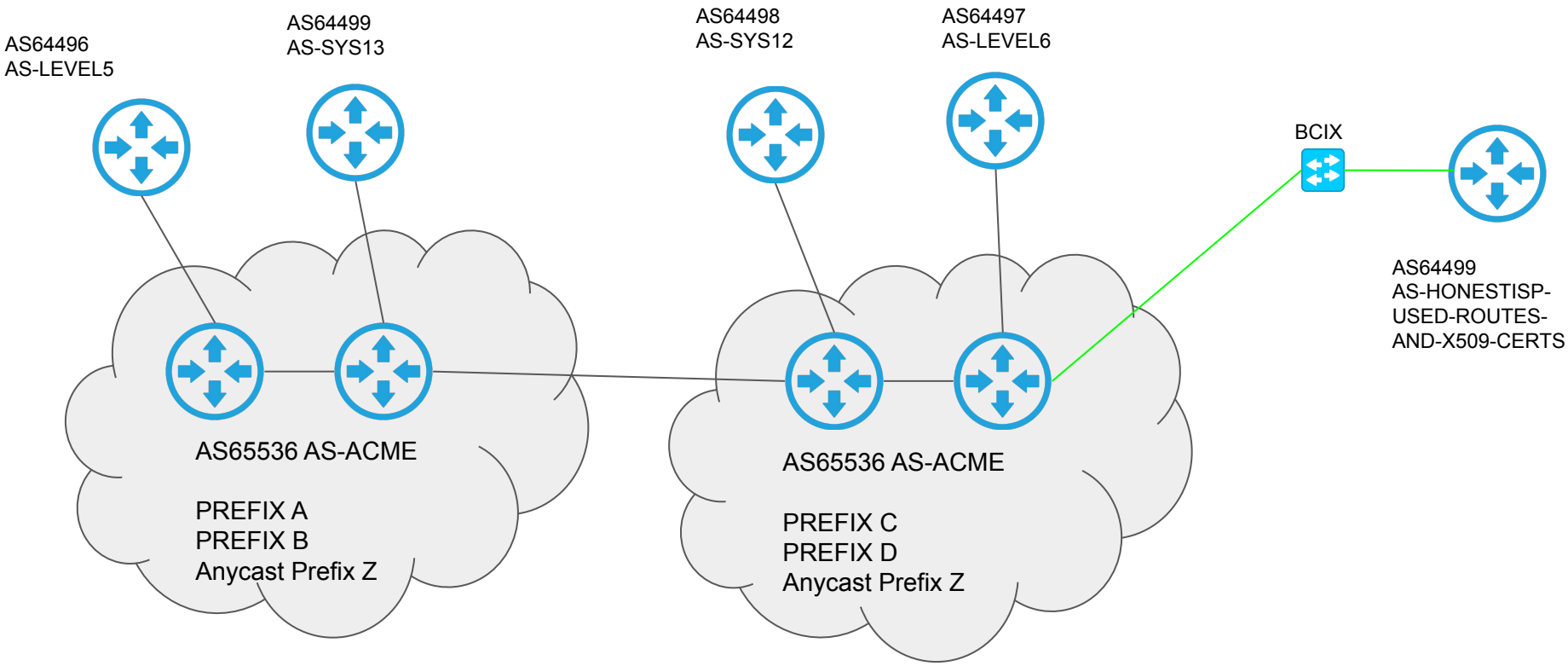
ACME Islands



Island with backbone

- Sites are connected to each other
- Announce only their own aggregates to eBGP peers
- Connectivity between islands through backbone links

ACME Interconnected Islands



iBGP + IGP

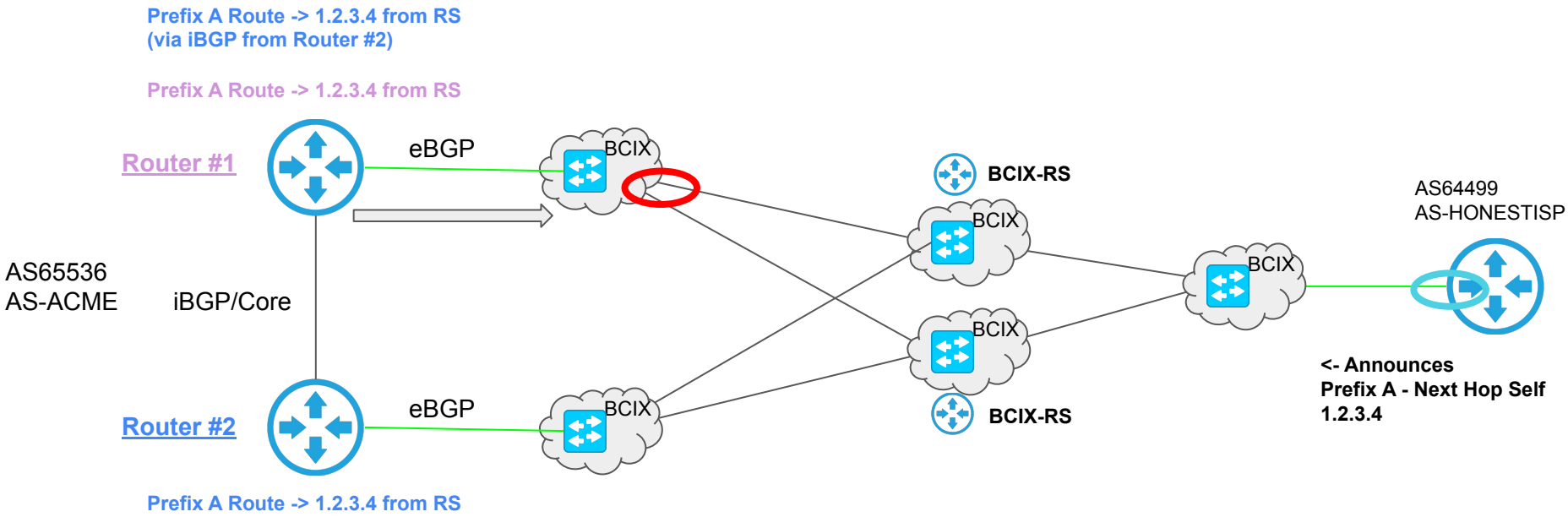
iBGP

All routers send their routes to **all** other routers in the network.

The loopback IP of the router is set as the next hop.

👉 This way, each router has a complete view of everything it can reach via the respective other routers.

Why Next-Hop-Self?



Problem: Traffic from Router #1 towards BCIX peers is blackholed

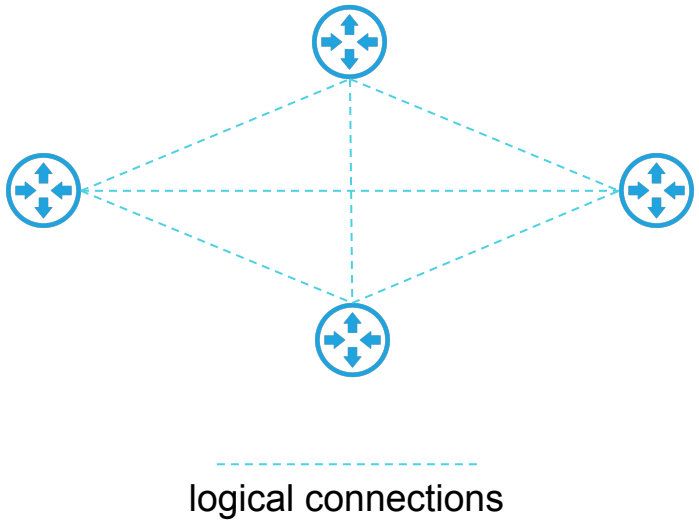
Solution: Setting "next-hop-self" in iBGP Policy!

Next-Hop-Self

```
[edit policy-options policy-statement IBGP-OUT]
term ANNOUNCE-TO-INTERNAL {
    from community ANNOUNCE_TO_INTERNAL;
    then {
        next-hop self;
        accept;
    }
}
term ANNOUNCE-DIRECT-ROUTES {
    from protocol direct;
    then {
        next-hop self;
        accept;
    }
}
```

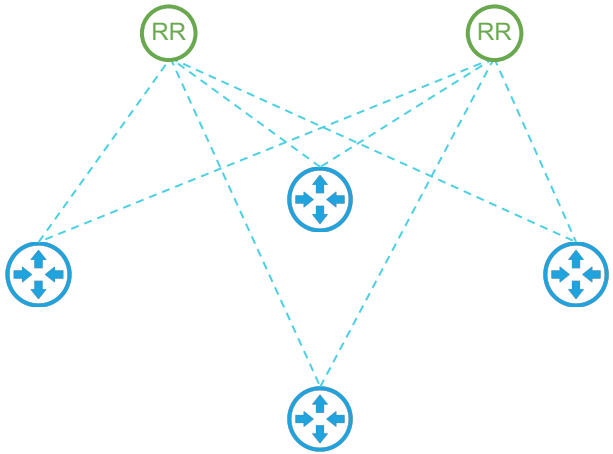
iBGP

- iBGP Full Mesh
  - Routers establish BGP sessions to all other routers
  - 👍 Resilient
  - 👎 A lot of configuration work (not the case with proper automation!)
  - 👎 Scales only up to a certain number of routers



iBGP

- iBGP Route Reflectors
  - Router establish sessions to route reflectors
  - Hybrid setups are possible (hierarchy, regions, etc.)



  - 👍 Less BGP sessions to maintain
  - 👍 Better scaling in large network
  - 👎 Potential SPOF

IGP

Exchange of loopback IPs via IGP

Which routes are distributed via the IGP?

- Router Loopback IPs
- Backbone transfer networks
- **NOTHING else**

Which IGP?

IS-IS or OSPF

What happens if you redistribute BGP full view into OSPF

https://routingcraft.net/what-happens-if-you-redistribute-bgp-full-view-into-ospf/

IGP

- Routers have routes for the loopback IPs of the other routers.
- If a link falls over, the IGP calculates a new path

👉 Routers can now establish iBGP sessions with each other via Loopback IPs, no matter what the topology looks like.

Recursive Lookup

👉 To resolve routes, a recursive lookup is performed.

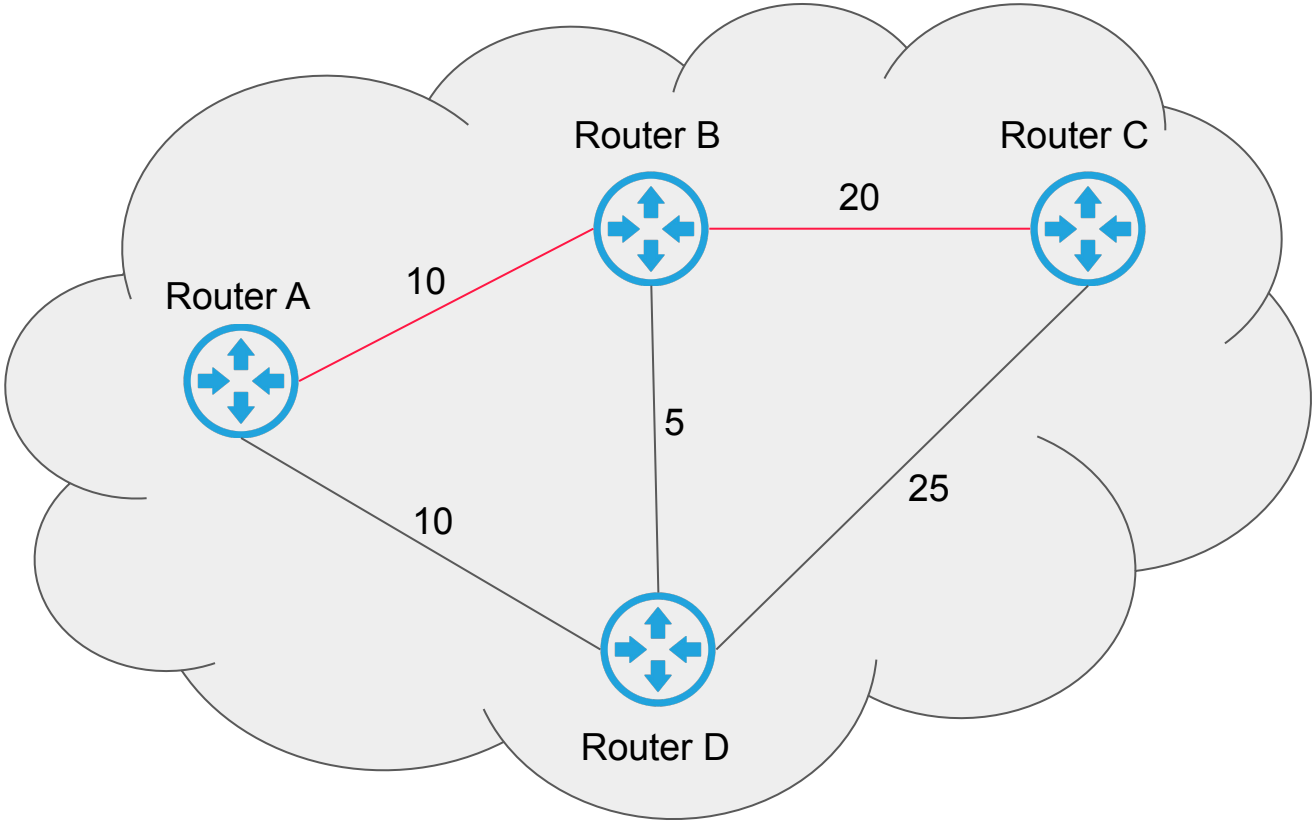
This way we always use the shortest path to the next router calculated by the IGP.

```
> show route 1.1.1.1

1.1.1.0/24          *[BGP/170] 7w5d 03:50:13, MED 1000, localpref 100, from 62.176.224.232
                       AS path: 13335 I, validation-state: unverified
                     >  to 62.176.224.232


> show route 62.176.224.232

62.176.224.232/32   *[IS-IS/18] 22w6d 11:15:32, metric 350
                     >  to 62.176.251.15 via et-0/0/2.0
```

IGP Metrics

- Links have configured "metric" or "cost".
- IGP tries to calculate the "cheapest" path (lowest metric).
- Cost of a path is calculated by adding the cost of the links.

ACME Backbone



Router A to Router C

Route:

Router A -> Router B	10
Router B -> Router C	+20
	= 30

IGP Metrics

What to set as a metric? It quickly becomes confusing with many links...

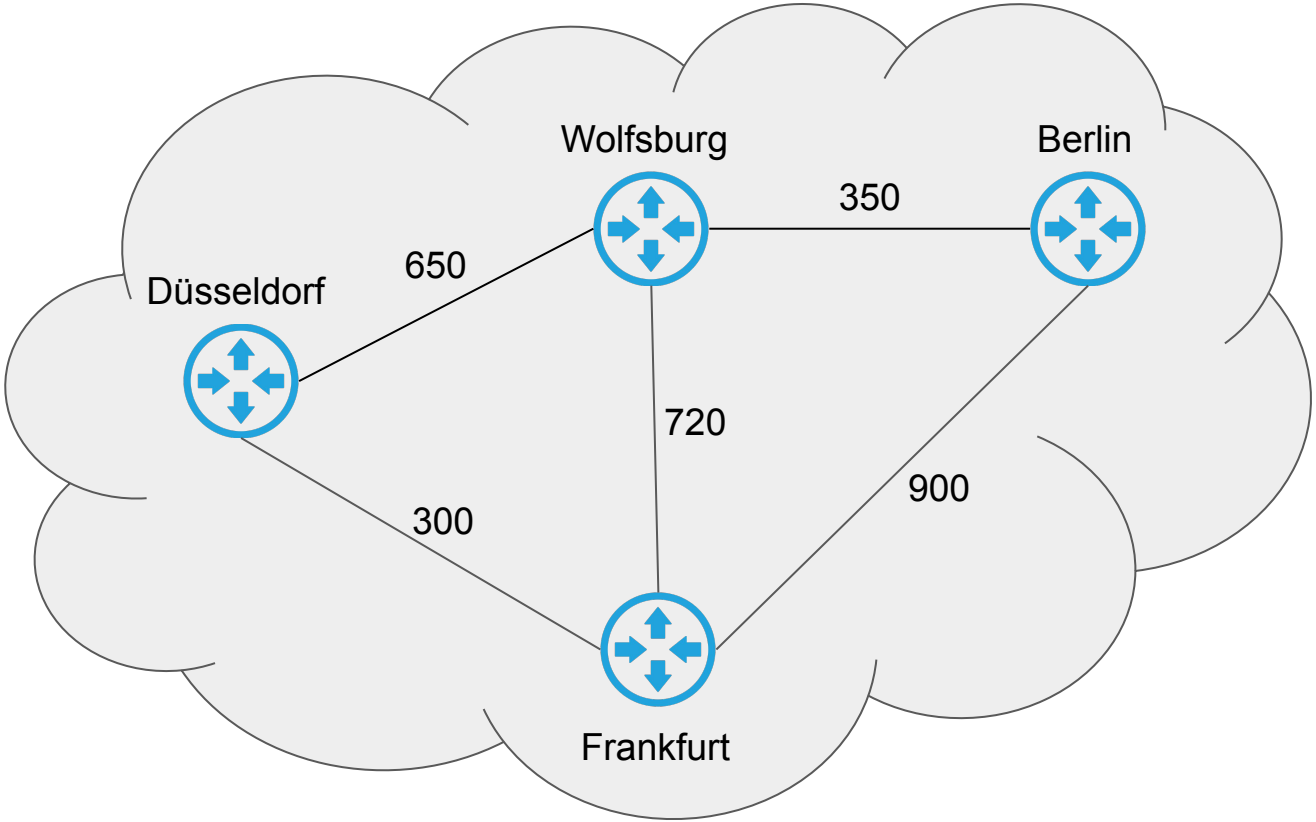
- Hops
  - The same metric everywhere
  - Can lead to extremely suboptimal routing 💩
- Link bandwidth
  - Example:
    - 10G: Metric 1000
    - 100G: Metric 100
  - Can lead to extremely suboptimal routing 💩
- Latency... (next slide)

IGP Metrics

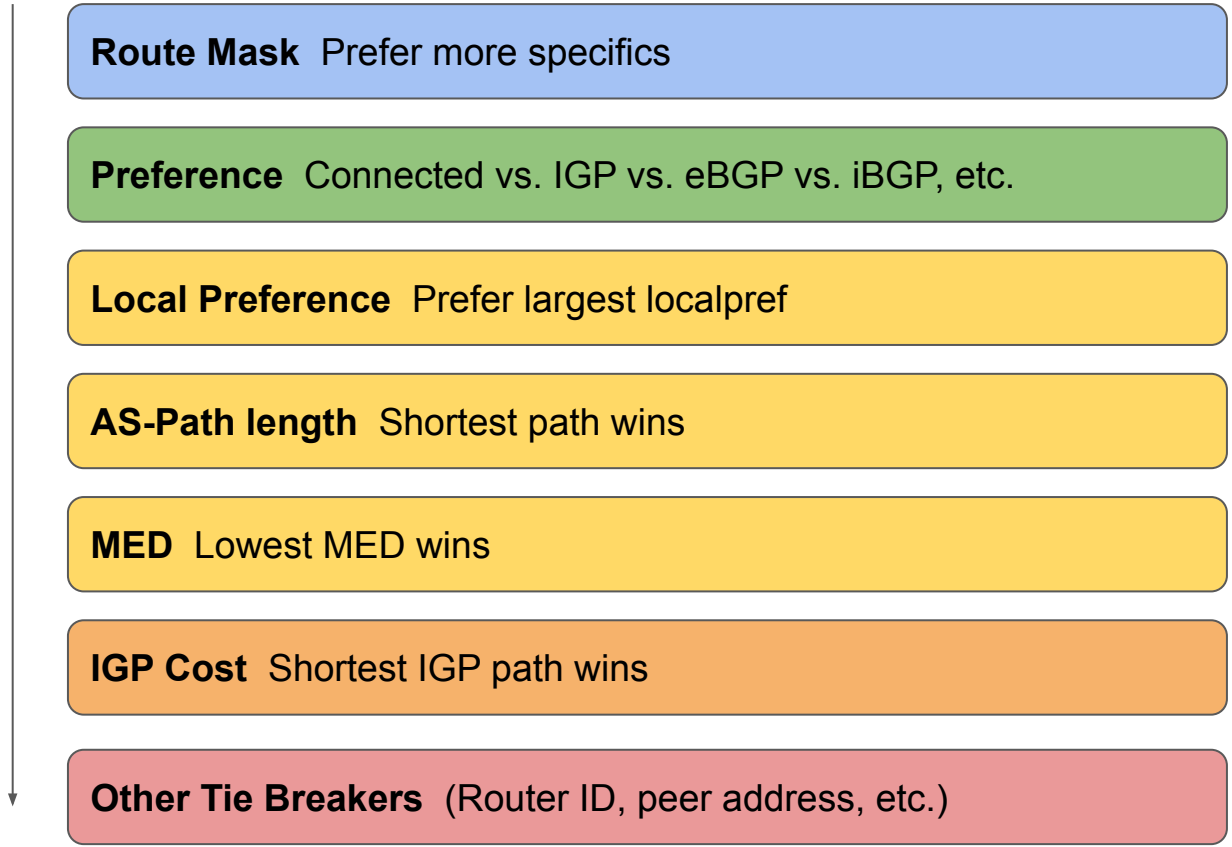
Latency!

- Example:
  - RTT ms * 10
  - 17,34 ms = 173
  - Links within a metro fixed at 10
- Takes the shortest path (in terms of latency) ✨
- (hopefully) saves money 💰
  - The longer the line, the more expensive
  - Metro links can be upgraded almost at will
- What if I have varying link speeds? Use a multiplier!
  - How much extra latency do I tolerate to prefer a thicker pipe?
  - Example: Multiply old 10G lines by a factor of 2 if the majority of the backbone is 100G.

Example Backbone



Crash course: Best Path Selection (Simplified)



Crash course: Best Path Selection (Simplified)

Route Mask Prefer more specifics

Preference Connected vs. IGP vs. eBGP vs. iBGP, etc.

Local Preference Prefer largest localpref

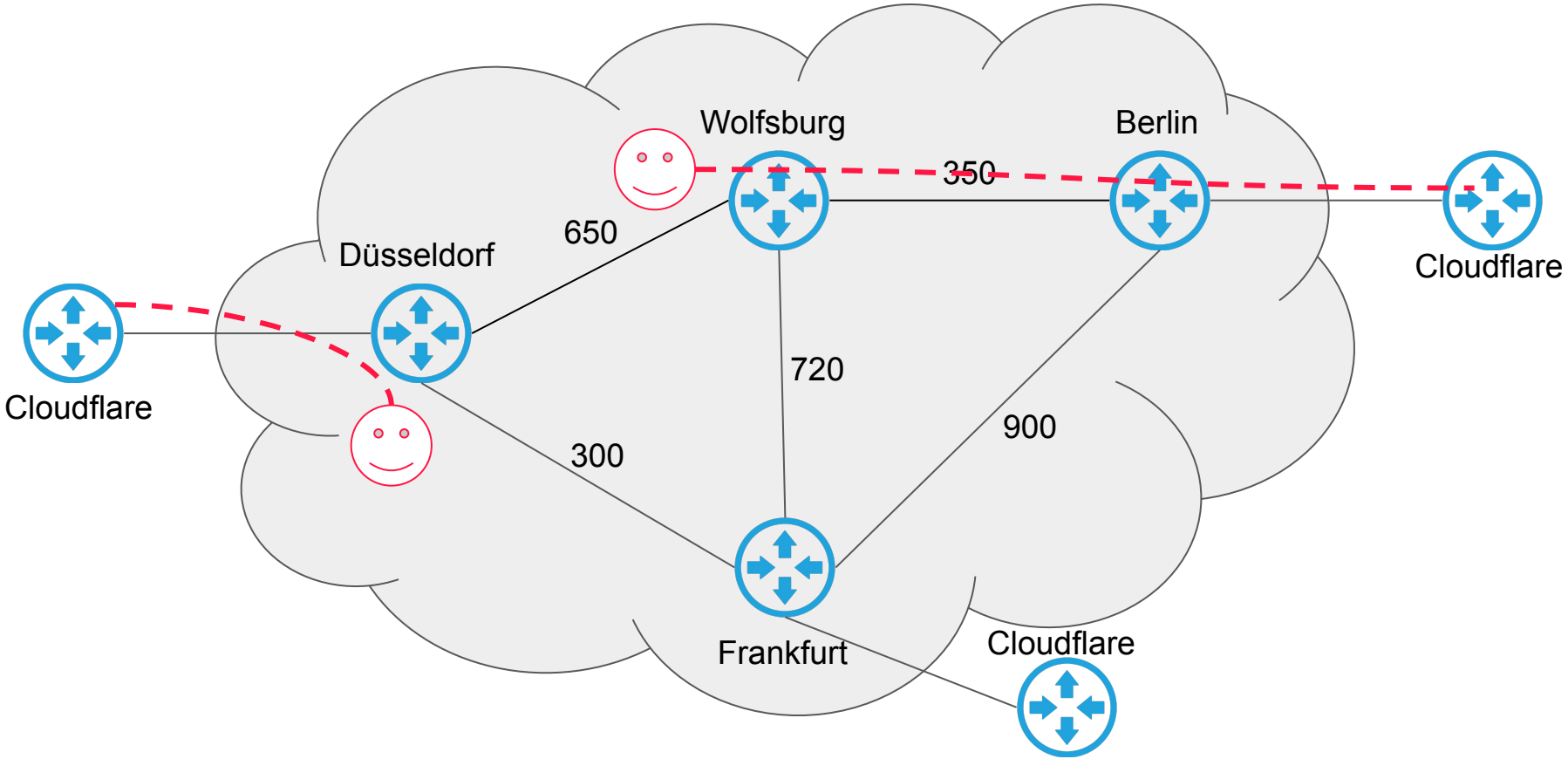
AS-Path length Shortest path wins

MED Lowest MED wins

IGP Cost Shortest IGP path wins

Other Tie Breakers (Router ID, peer address, etc.)

Example Backbone



Example

```
> show route 1.1.1.1

1.1.1.0/24         *[BGP/170] 7w5d 03:50:13, MED 1000, localpref 100, from 62.176.224.232
                      AS path: 13335 I, validation-state: unverified
                    >  to 62.176.224.232


> show route 62.176.224.232

62.176.224.232/32  *[IS-IS/18] 22w6d 11:15:32, metric 350
                    >  to 62.176.251.15 via et-0/0/2.0



> ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: icmp_seq=0 ttl=62 time=3.930 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=62 time=3.928 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=62 time=3.995 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=62 time=3.906 ms
^C
--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.906/3.940/3.995/0.033 ms
```

Traffic Engineering

Get packets from A to B via γ

Less is more

Preference? MED? Local preference? more specifics ???

aaaaaaaaaaaaaaaaaaaaaaaa ???

In? Out?

Attention:

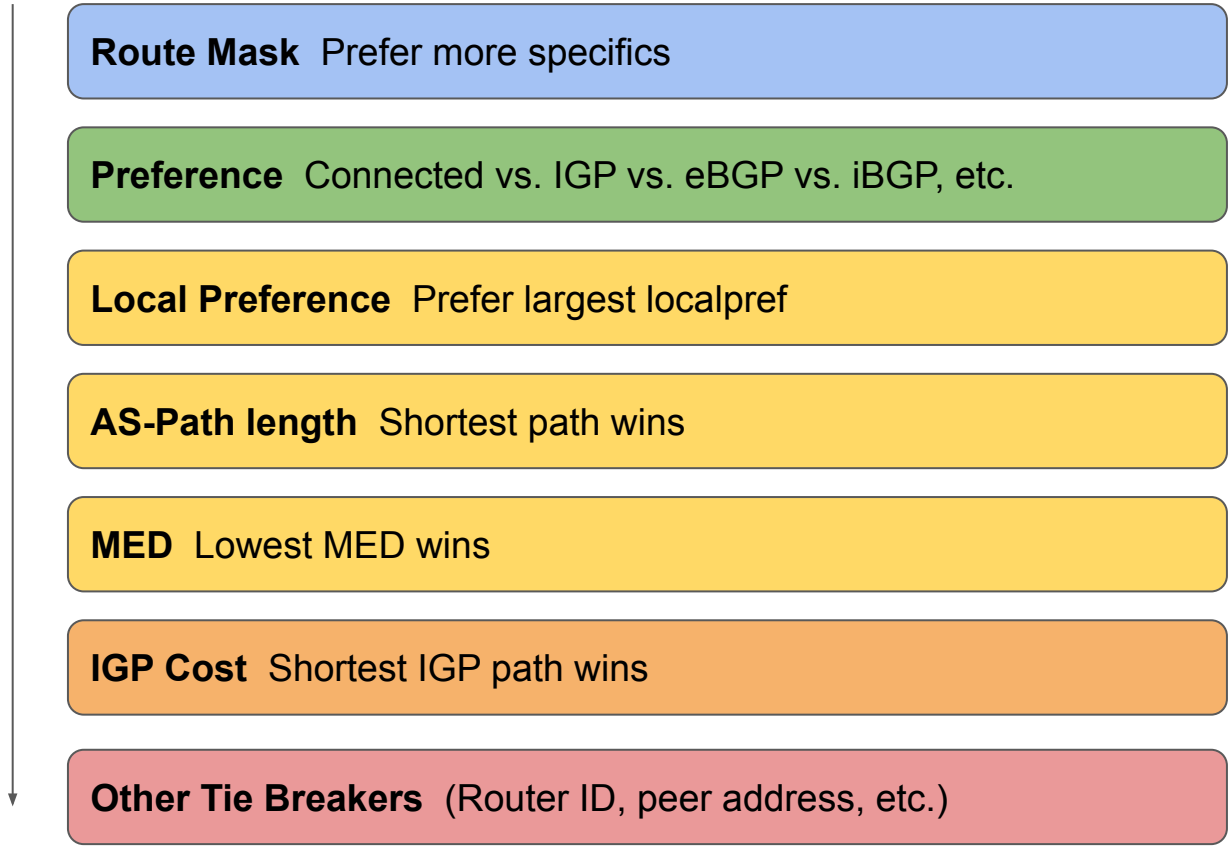
Incoming routes:

👉 Control outgoing traffic

Outgoing routes:

👉 Control incoming traffic

Crash course: Best Path Selection (Simplified)



Inbound Traffic: More specifics 🔨

- Nuclear option
- Always use this path, no matter what.

💀 Some CDNs ignore more specifics

💀 Breaks graceful shutdown

💀 Difficult with RPKI ROAs

👉 Not really suitable for traffic engineering

👉 Can be used for traffic-pull for DDoS mitigation

Outbound Traffic: Administrative Distance

- Preference by protocol
- Varies by vendor
- Lower wins
- Don’t touch this for traffic engineering

Junos defaults:

Local/Connected	0
Static	5
OSPF	10
ISIS	18
BGP	170

Outbound Traffic: Localpref

Higher value wins

Overrides AS-Path
Overrides MED
Overrides IGP Metrik

Example

Graceful Shutdown	0
Default	100
Customer	200

Our opinion: Don’t.

Better use MED for outbound traffic engineering.

Exception: Customer prefixes

Why higher local pref for customer prefixes?

Customer also on Internet Exchange?

Worst case: IXP route wins -> customer offline.
The route learned via the IXP has no TO_UPSTREAM community.

Alternative: De-peer customer -> Can also backfire

AS-Path Prepend

Selectively extend the path for some peers/prefixes by appending your own ASN again.

Before: **9136** 208942 i
After: **9136 9136** 208942 i

Example:

- Backup path
- Push most of the traffic away from saturated link

💀 Doesn’t work when others networks mess with local pref

💡 Please don’t prepend more than 2-3 times

Outbound Traffic: Selective Path Prepend

What?

I would like to send traffic to DTAG only via Telia if there is no other way.

How?

AS paths starting with 1299 3320 [...] become 9136 1299 3320 [...]

```
as-path DTAG_VIA_TELIA "1299 3320.*";

policy-statement TELIA_IMPORT {
        term PREPEND_DTAG_IMPORT {
            from as-path DTAG_VIA_TELIA;
            then as-path-prepend "9136";
        }
}
```

Inbound Traffic: Selective Prepend

Many providers allow selective prepend via BGP Community.

By ASN, region, etc.

BCIX Route Server:

0:<xxx> or 16374:0:<xxx>	No export to ASxxx
16374:101:<xxx>	Prepend once, only to ASxxx

Lumen:

65001:<xxx>	Prepend to ASxxx
64981:0	Prepend once to all EU peers

Inbound Traffic: Path Prepend

What?

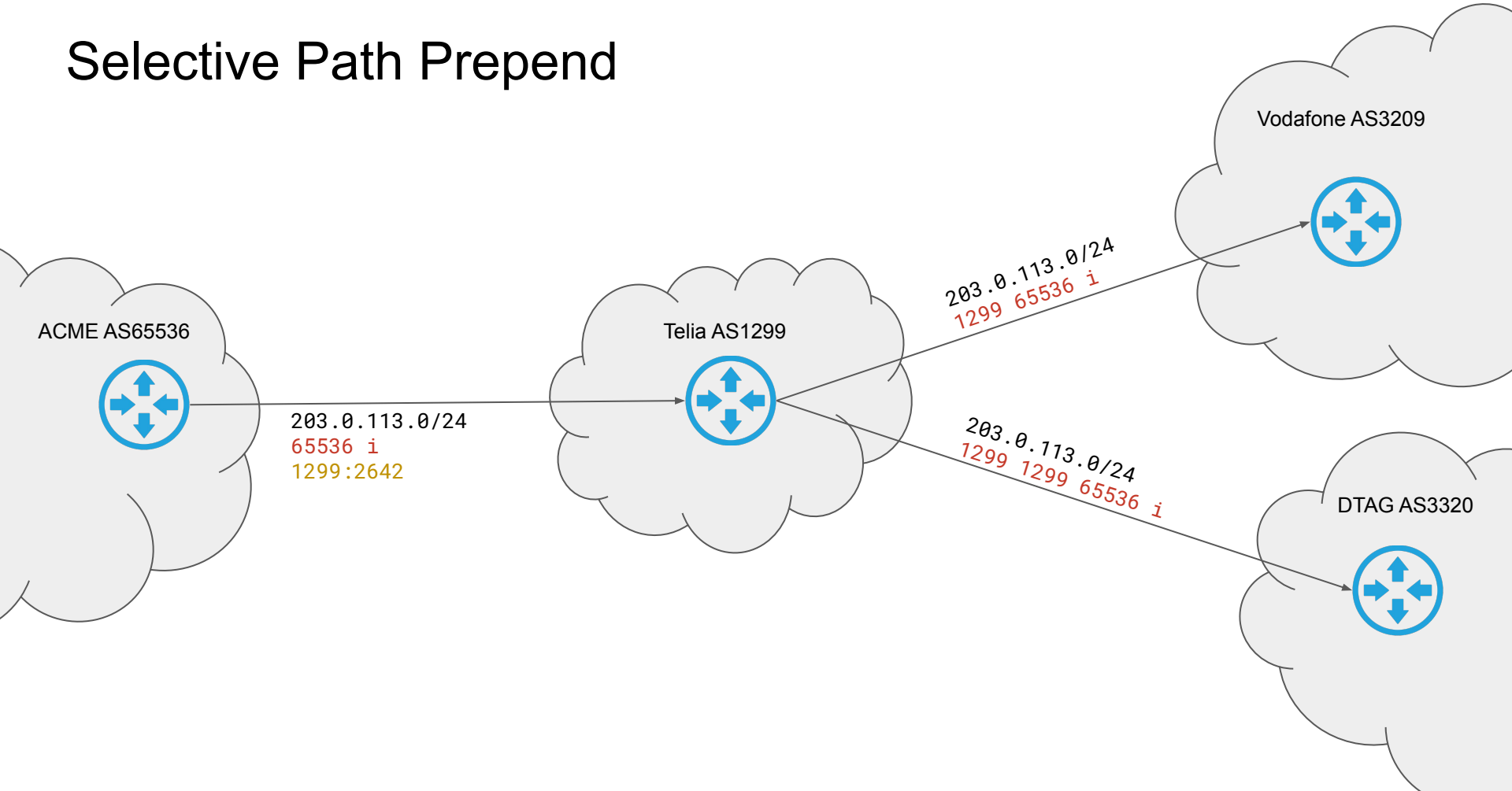
I want to get traffic from DTAG only via Telia if there is no other way.

How?

Add the community 1299:2642 to the prefix to make the Telia router prepend to DTAG.

```
policy-options {
     community TELIA_PREPEND_DTAG_TWICE {
        members 1299:2642;
     }

    policy-statement TELIA_EXPORT {
        term PREPEND_DTAG_EXPORT {
            then {
                community add TELIA_PREPEND_DTAG_TWICE;
            }
        }
    }
}
```

Selective Path Prepend



MED

- Lower wins
- Considers AS path
- Overwrites IGP metric

Default:

Compares MED if first AS in path is same

Outbound Traffic: MED

Example:

Use the PNI to Twitch, no matter if it's on the other end of my network.

Set to 1000 everywhere, set to 900 on the PNI.

👉 Default is 0

We have to increase the MED everywhere to be able to be able to prefer paths

Inbound Traffic: MED

Good idea, but many peers ignore the MED value.

But you can try!

💀 Can cost you downstream traffic if peers have always-compare-med enabled

💡 Works with some transit providers if you have multiple circuits there

Something is strange?
Traffic where it shouldn't be?

Ask for help!

IRC

#networker https://spodder.com/networker/

#denog https://www.denog.de/de/chatterliste_iframe.html

#ix on irc.terahertz.net

Telegram

t.me/bgpde

Multipathing

We want to use all available paths of the same length.

Multipathing is even more important for peering:

-> Peers with multiple routers.

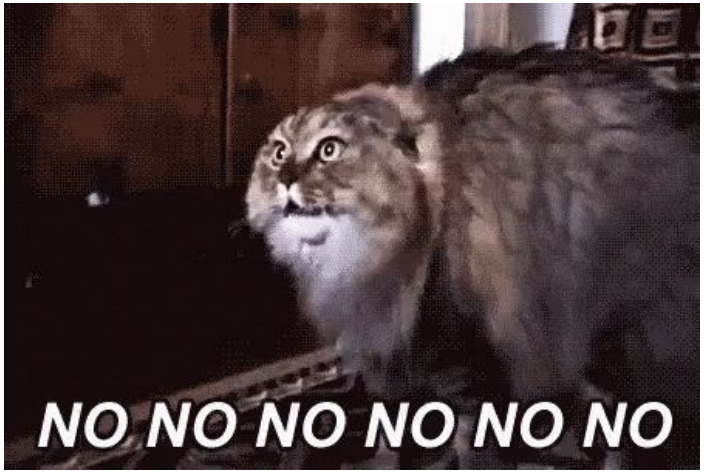
Optionally: multipath multiple-as for multipathing across multiple next-hop ASNs.

```
# In FIB einschalten
set policy-options policy-statement MULTIPATHING then load-balance per-packet

set routing-options forwarding-table export MULTIPATHING


# Für BGP einschalten
set protocols bgp multipath / multipath multiple-as?
```

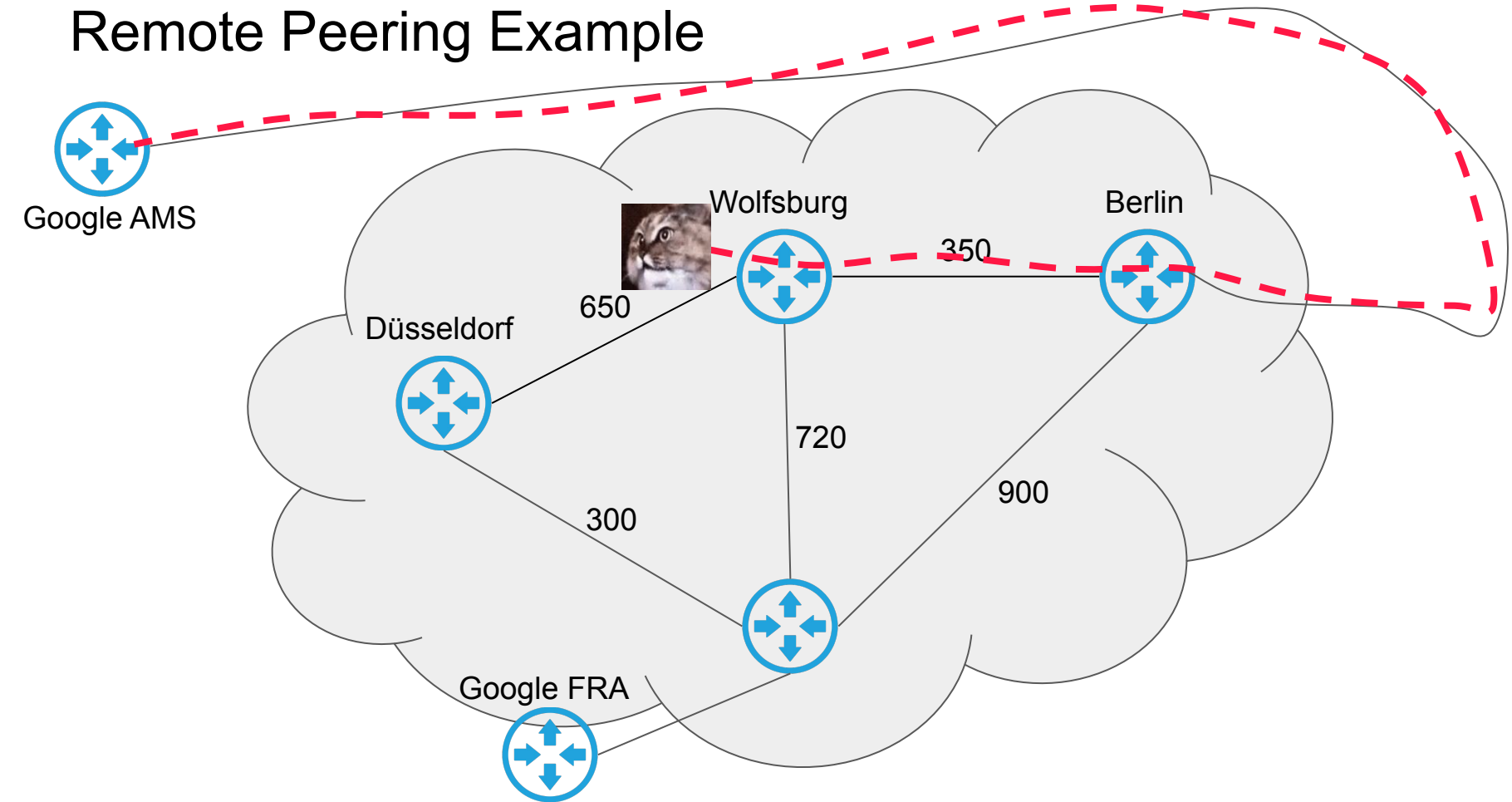
One more topic before the bio break...

Remote Peering?

- Router of other peer is geographically located somewhere else than the Internet Exchange
- Our router cannot know about it
- Whoops, Scenic Routing

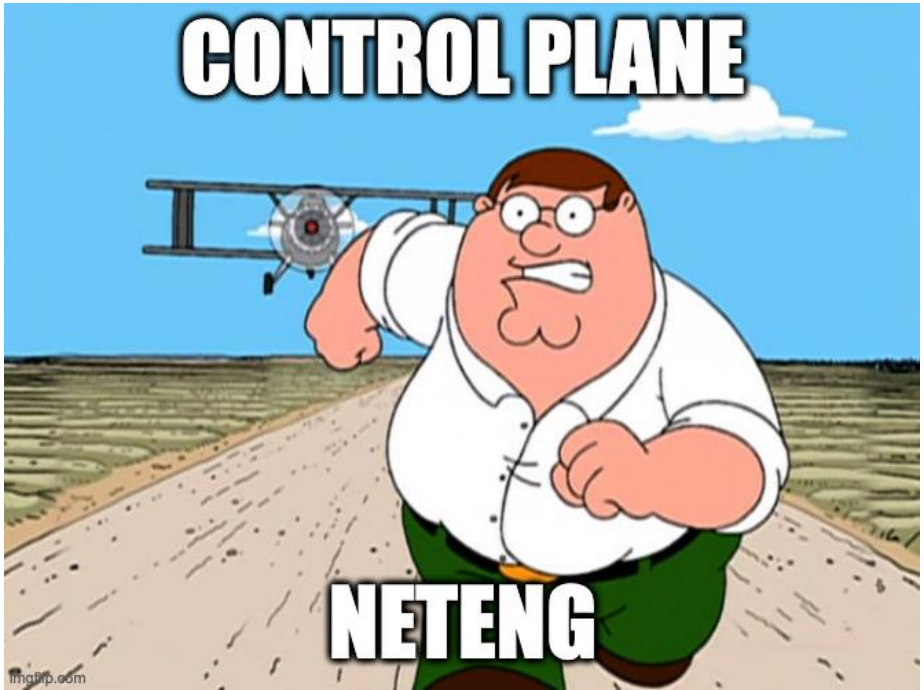


Remote Peering Example



BREAK NOW ☕

Session 2
Route filtering



Agenda

- **RIPE, RPKI & PeeringDB - Basics - ~5 min**
  - RPKI ROA Signing
  - DB Housekeeping (AS-SET, ROUTE)
  - DB Query Hints
  - Inetnum Automation (RIPE Updater)
  - PeeringDB
  - MANRS

- **Route Filtering - Building Blocks - ~25 min**
  - Filtering BCPs
    - RPKI, PeeringLANs, Prefixlen
    - Bogon ASes/Prefixes, AS Pathlen, Tier1 ASNs
    - My Prefixes, Direct Peer Prefixes
    - Communities, Prefix Limit
    - AS Path, Prefix List
  - Maintenance Switch, Graceful shutdown

- **Route Filtering - Policy Building - ~15 min**
  - Buildings blocks for a public peer
  - Out Policy Example

Bio Break - 45 min

RIPE, RPKI & PeeringDB

The usual RPKI drill

Sign your prefixes!

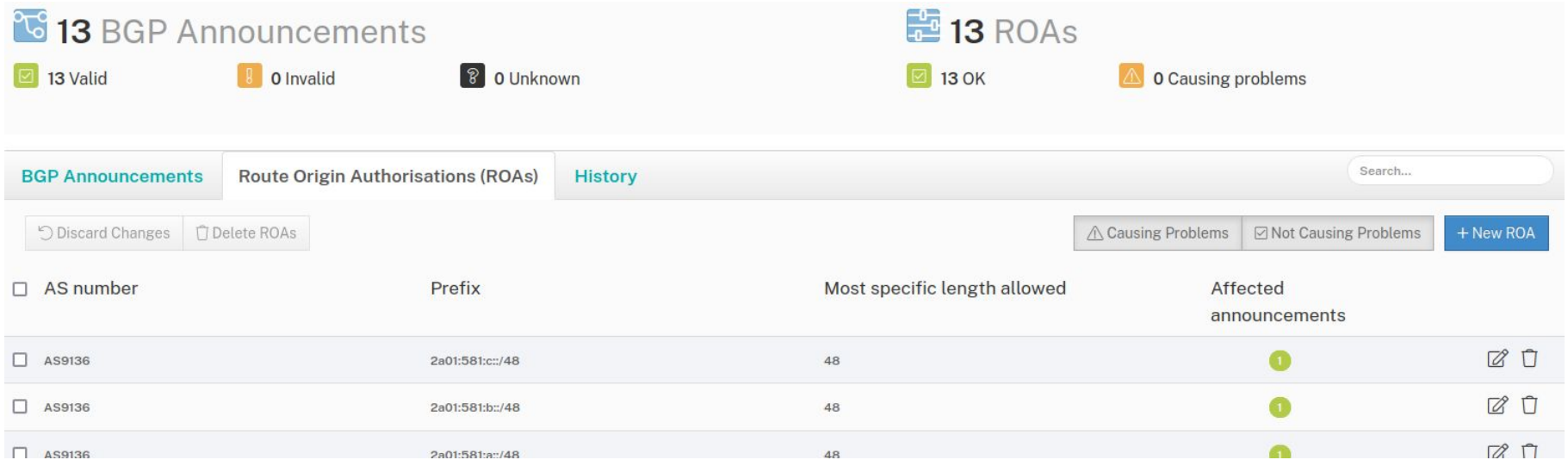
LIR Portal

https://access.ripe.net/

RPKI Dashboard

https://my.ripe.net/#/rpki

The usual RPKI drill



AS number: ASXXXXX

Prefix: 172.16.0.0/12

Max length: 20

Keep your AS-SET up-to-date!

- The AS-SET plays a major role for filtering
- Automate it or maintain it manually
- You don't know your AS-SET?

```
# Search for AS-SETs for which you are maintainer
whois -h whois.ripe.net -i mnt-by <YOUR MAINTAINER HANDLE> -Br -T as-set
```

- You don't know your maintainer?

https://my.ripe.net/#/account-details -> **Sektion: Maintainer**

AS-SET

```
vpetzholtz@area51:~$ whois AS-SYSELEVEN
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to 'AS-SYSELEVEN'

as-set:         AS-SYSELEVEN
descr:          SysEleven GmbH
descr:          Boxhagener Strasse 80
descr:          10245 Berlin
descr:          Germany
members:        AS25291
members:        AS43902
members:        AS49130
members:        AS201066
members:        AS202499
members:        AS-BER
members:        AS-CHAOS
```

Know your route(6) objects!

- It is recommended to have one route object per allocation and all /24’s in it
- e.g. for traffic engineering (in case of DDoS)

```
# Legacy IP
whois -h whois.ripe.net -i mnt-by <YOUR MAINTAINER HANDLE> -Br -T route
# IPv6
whois -h whois.ripe.net -i mnt-by <YOUR MAINTAINER HANDLE> -Br -T route6
```

👉 **Otherwise you have to assume that you will not be able to "get through" everywhere with an announcement.**

route(6)

```
vpetzholtz@area51:~$ whois 2a00:13c8::/32 -T route6
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '2a00:13c8::/32AS25291'

route6:         2a00:13c8::/32
descr:          SysEleven Global Network - Aggregate route object
origin:         AS25291
remarks:        Managed by SysEleven RIPE static templates
mnt-by:         SYS11-MNT
created:        2010-02-04T15:19:51Z
last-modified:  2019-07-31T12:25:30Z
source:         RIPE # Filtered
```

Usefull RIPE DB / Whois Queries

```
- All objects of one maintainer
whois -h whois.ripe.net -i mnt-by <MAINTAINER HANDLE> -Br
- AS-SETs
whois -h whois.ripe.net -i mnt-by <MAINTAINER HANDLE> -Br -T as-set
- route objects
whois -h whois.ripe.net -i mnt-by <MAINTAINER HANDLE> -Br -T route
- Look for all inetnum objects at/within 1.2.0.0/20
whois -h whois.ripe.net -T inetnum -MBr 1.2.0.0/20

Some good flags:
-h specify database
-i inverse lookup by itemfield
-B full contact details
-r do NOT show related db items
-T specify item type

Common types: inetnum, inet6num, aut-num, as-set, route, route6, domain, mntner, organisation, role, person
```

Keep Whois (IRR) DB up-to-date!

Remember to keep your RIPE/ARIN/RADB up to date!

inet(6)num, ASN, AS-SET, route(6) etc.

If you use Netbox the RIPE Updater could help you to maintain the RIPE DB!

Github

https://github.com/interdotlink/ripe-updater

Keep PeeringDB up-to-date!

Remember to keep your PeeringDB up to date!

Name, ASN, AS-SET, Exchange Points, Facilities, Contact Data!

https://peeringdb.com/asn/<your asn>

M.A.N.R.S.

“Mutually Agreed Norms for Routing Security (MANRS) is a global initiative that helps reduce the most common routing threats.”

You become a member!

https://www.manrs.org/

It’s also a good source of information (about routing security):

MANRS-Network-Implementation-Guide.pdf

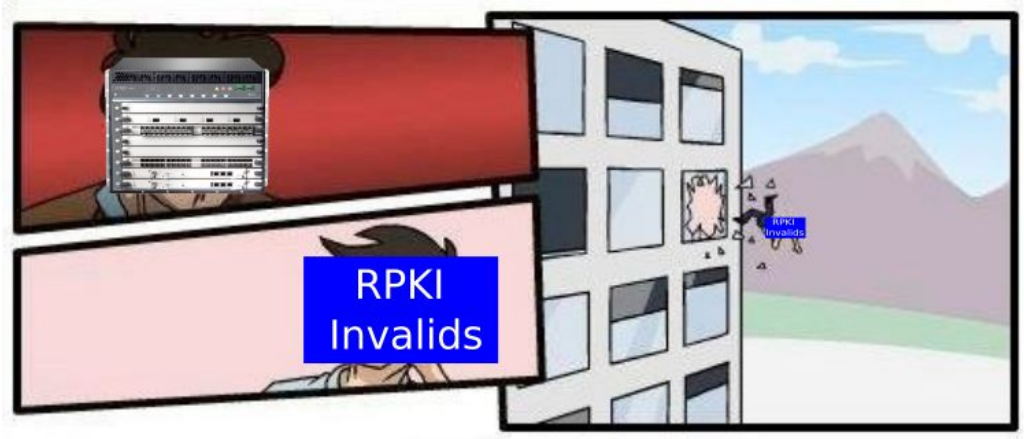
Route Filtering Building Blocks

Route Filtering

- Trust unfortunately **not** enough
- Good filtering makes the Internet a better place
  - It limits the expansion of leaks and hijacks
  - Increases stability and leads to more trust (will it ever be enough tho?)
- Filtering should be done on all "Edges" (IN/OUT)
  - Upstream
  - Downstream
  - Peering
- Traditionally you filter IN "a little" more than OUT ;-)

How to deal with RPKI invalids

- Invalids = prefixes with wrong origin AS and/or allowed prefix length.
- Dropping Invalids means to "respect" the ROA.
- This provides some protection against prefix hijacks and other route leaks



👉 Invalids don’t belong in the routing table

ROUTINATOR

Hint: Routinator & OctoRPKI (+RTR Server)

Use of the validators

```
# Validator Sessions
[edit routing-options validation]
group RPKI {
    max-sessions 4;
    session 2c01:dead:1:2::1 {
        port 3323;
        local-address 2c01:dead:2:2::1;
    }
    session 2c01:dead:1:2::2 {
        port 3323;
        local-address 2c01:dead:2:2::1;
    }
}

# Allow traffic to control plane (lo0)
term RPKI-ALLOW {
    from {
        source-prefix-list {
            RPKI-RTR-SERVERS;
        }
        protocol tcp;
        source-port [ 8282 3323 323 ];
    }
    then accept;
}
```

```
[edit policy-options policy-statement 4-BASE-IN]
term RPKI-MARK-VALID {
    from validation-database valid;
    then {
        validation-state valid;
    }
}
term RPKI-REJECT-INVALID {
    from validation-database invalid;
    then {
        validation-state invalid;
        reject;
    }
}
term RPKI-MARK-UNKNOWN {
    then {
        validation-state unknown;
    }
}
```

Direction: IN - Relevant for: All eBGP sessions

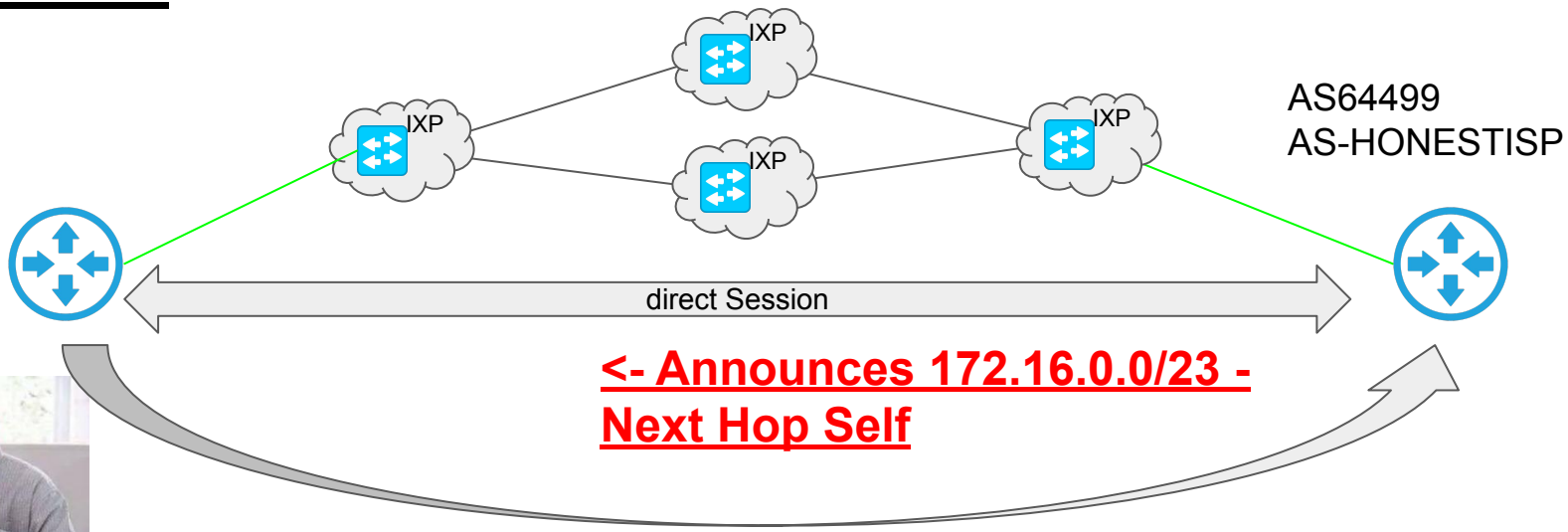
Peering LANs

- You should avoid accepting peering LANs (prefixes).

- This applies especially to the peering LANs to which your own network is connected!

Peering LANs

Peering LAN: 172.16.0.0/22



Problem: ACME router loses the peers in the /23 and blackholed traffic at the next hop if necessary.

Solution: Filter Peering LAN Prefixes (including more specifics)!

```
[edit policy-options]
prefix-list 4-PEERING-LANS {
    80.81.192.0/21;      # DE-CIX FRA
    193.178.185.0/25;    # BCIX
    80.249.208.0/21;     # AMS-IX
}
[edit policy-options]
prefix-list 6-PEERING-LANS {
    2001:7f8::/64; # DE-CIX FRA
    2001:7f8:1::/64;     # AMS-IX
    2001:7f8:19:1::/64;  # BCIX
}
[edit policy-options policy-statement 4-BASE-IN]
term FILTER-PEERING-LANS {
    from {
        prefix-list-filter 4-PEERING-LANS orlonger;
    }
    then reject;
}
[edit policy-options policy-statement 6-BASE-IN]
term FILTER-PEERING-LANS {
    from {
        prefix-list-filter 6-PEERING-LANS orlonger;
    }
    then reject;
}
```

Direction: IN - Relevant for: All eBGP sessions

Filter Prefix Length

- Mask too long:
  - Legacy IP: </24
  - IPv6: </48
- Mask too short:
  - Legacy IP: 0/0 default OR /0-/7
  - IPv6: ::/0
- Exceptions:
  - e.g. for own customers
  - Remote Triggered Blackholing (RTBH)

```
[edit policy-options policy-statement 4-BASE-IN]
term FILTER-PREFIX-LENGTH {
    from {
        route-filter 0.0.0.0/0 prefix-length-range /0-/7;
        route-filter 0.0.0.0/0 prefix-length-range /25-/32;
    }
    then reject;
}

[edit policy-options policy-statement 6-BASE-IN]
term FILTER-PREFIX-LENGTH {
    from {
        route-filter ::/0 prefix-length-range /0-/0
        route-filter ::/0 prefix-length-range /49-/128;
    }
    then reject;
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

BOGON ASNs

- Private ASNs are used for the lab and internal sessions.
- From and to external peers these ASNs shouldn’t be present

-> Reserved (64512-65534)

-> (RFC) Examples (64496-64511, 65536-65551)

...

```
[edit policy-options]
as-path-group BOGON-ASNS {
    as-path ZERO ".* 0 .*";
    as-path AS_TRANS ".* 23456 .*";
    as-path EXAMPLES1 ".* [64496-64511] .*";
    as-path EXAMPLES2 ".* [65536-65551] .*";
    as-path RESERVED1 ".* [64512-65534] .*";
    as-path RESERVED2 ".* [4200000000-4294967294] .*";
    as-path LAST32 ".* 65535 .*";
    as-path LAST64 ".* 4294967295 .*";
    as-path IANA_RESERVED ".* [65552-131071] .*";
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-BOGON-ASNS {
    from as-path-group BOGON-ASNS;
    then reject;
}

[edit policy-options policy-statement 6-BASE-IN]
term REJECT-BOGON-ASNS {
    from as-path-group BOGON-ASNS;
    then reject;
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

BOGON Prefixes

- There are prefixes you should never receive as announcements
- You should not announce them yourself :-D
- These are usually blocks that are used for certain RFCs or are reserved for other reasons

Special thanks to Team CYMRU!

https://team-cymru.com/community-services/bogon-reference/

```
[edit policy-options]
prefix-list 4-BOGON-PREFIXES {
    0.0.0.0/8;
    10.0.0.0/8;
    100.64.0.0/10;
    127.0.0.0/8;
    169.254.0.0/16;
    172.16.0.0/12;
    192.0.0.0/24;
    192.0.2.0/24;
    192.88.99.0/24;
    192.168.0.0/16;
    198.18.0.0/15;
    198.51.100.0/24;
    203.0.113.0/24;
    224.0.0.0/4;
    240.0.0.0/4;
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-BOGONS {
    from {
        prefix-list-filter 4-BOGON-PREFIXES orlonger;
    }
    then reject;
}
```

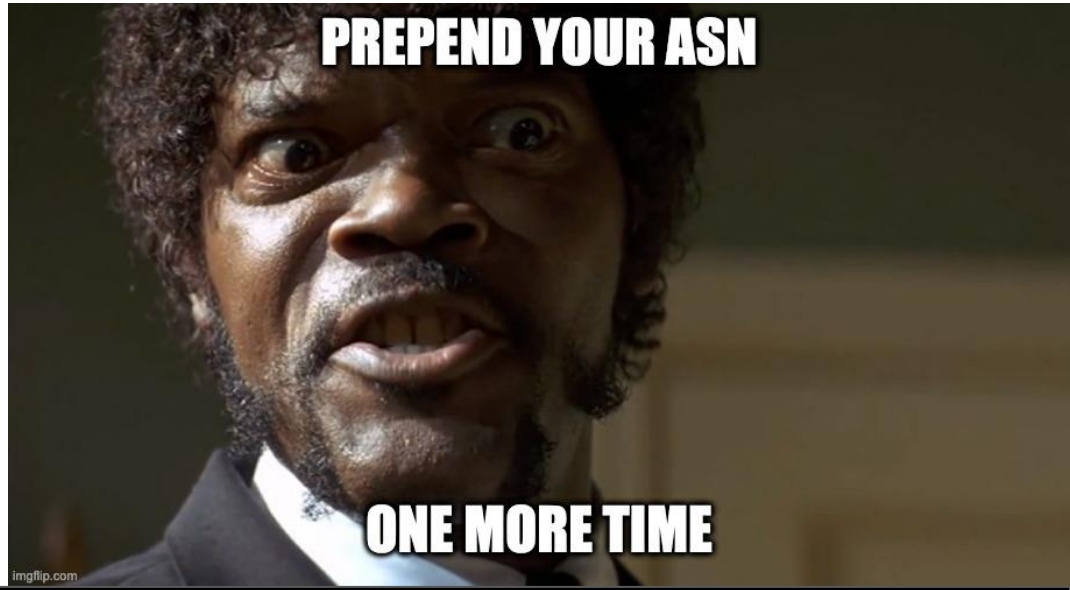
Direction: IN / OUT - Relevant for: All eBGP sessions

```
[edit policy-options]
prefix-list 6-BOGON-PREFIXES {
    ::/8;
    100::/8;
    200::/7;
    400::/6;
    800::/5;
    1000::/4;
    2000::/16;
    2001::/16;
    2002::/16;
    4000::/2;
    8000::/1;
}

[edit policy-options policy-statement 6-BASE-IN]
term REJECT-BOGONS {
    from {
        prefix-list-filter 6-BOGON-PREFIXES orlonger;
    }
    then reject;
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

AS Path too long

- AS Path Prepend can lead to extreme path lengths
- Too long paths waste memory and could trigger bugs (and did so in the past)




```
> show route aspath-regex ".{37,}" table inet.0

inet.0: 887212 destinations, 6177645 routes (887143 active, 0 holddown, 74 hidden)
+ = Active Route, - = Last Active, * = Both

45.162.104.0/22     [BGP/170] 3d 00:58:53, MED 100, localpref 100
                      AS path: 1299 52320 11432 11432 11432 11432 11432 11432 11432 11432 11432 11432 11432 11432
11432 11432 11432 11432 11432 11432 11432 11432 11432 11432 11432 264216 264216 264216 264216 264216 264216 264216
264216 264216 264216 264216 268516 I, validation-state: unknown
                    >  to 213.248.81.225 via ae1.0
```

```
[edit policy-options]
as-path AS-PATH-MAX-LENGTH ".{50,}";

[edit policy-options policy-statement 4-BASE-IN]
term AS-PATH-WAY-TOO-LONG {
    from {
        as-path AS-PATH-MAX-LENGTH;
    }
    then reject;
}
```

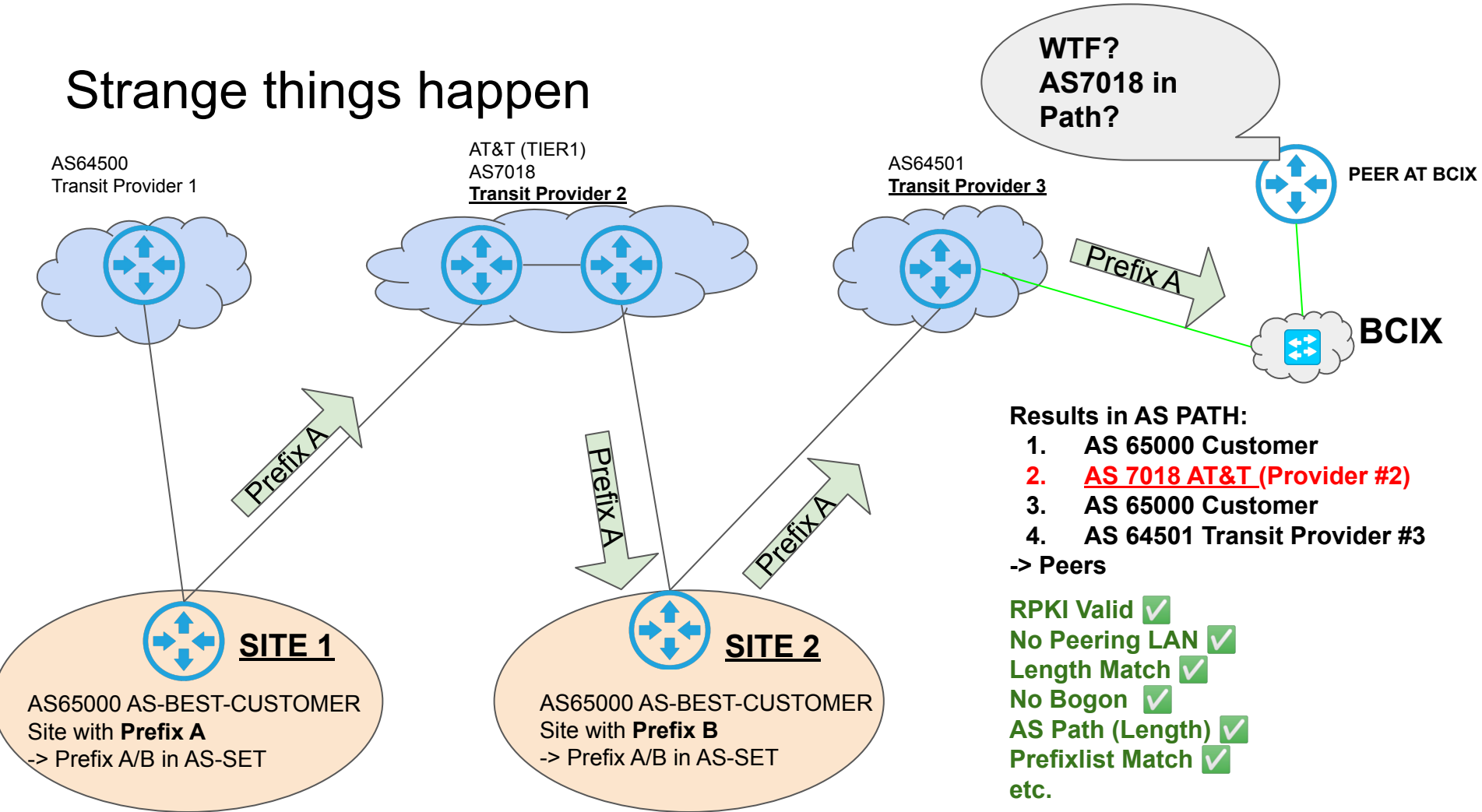
Direction: IN - Relevant for: All eBGP sessions

Prefixes with "Tier1" ASNs

- If someone announces paths with Tier1 ASNs to me (e.g. via peering) there's most likely something wrong
- But what or who is Tier1 anyway? (don't get us started :-)
- Why do you need it?

💀 Do not apply to upstream sessions!

Strange things happen



Results in AS PATH:

1. **AS 65000 Customer**
2. **AS 7018 AT&T (Provider #2)**
3. **AS 65000 Customer**
4. **AS 64501 Transit Provider #3**

-> Peers

RPKI Valid ✅
No Peering LAN ✅
Length Match ✅
No Bogon ✅
AS Path (Length) ✅
Prefixlist Match ✅
etc.

```
[edit policy-options]
as-path-group TIER1-ASNS {
    as-path ATT ".* 7018 .*";
    as-path CENTURYLINK ".* 3356 .*";
    as-path CENTURYLINK1 ".* 3549 .*";
    as-path CHINATELECOM ".* 4134 .*";
    as-path CHINAUNICOM ".* 4837 .*";
    as-path COMCAST ".* 7922 .*";
    as-path COGENT ".* 174 .*";
    as-path DTAG ".* 3320 .*";
    as-path GTT ".* 3257 .*";
    as-path HE ".* 6939 .*";
    as-path LIBERTYGLOBAL ".* 6830 .*";
    as-path NTT ".* 2914 .*";
    as-path ORANGE ".* 5511 .*";
    as-path PCCW ".* 3491 .*";
    as-path RETN ".* 9002 .*";
    as-path SPRINT ".* 1239 .*";
    as-path TATA ".* 6453 .*";
    as-path TELECOMITALIA ".* 6762 .*";
    as-path TELEFONICA ".* 13184 .*";
    as-path TELIA ".* 1299 .*";
    as-path VERIZON ".* 701 .*";
    as-path VODAFONECARRIER ".* 1273 .*";
    as-path ZAYO ".* 6461 .*";
}
```

```
[edit policy-options policy-statement 4-PEER-IN]
term REJECT-TIER1-IN-PATH {
    from as-path-group TIER1-ASNS;
    then reject;
}
```

Direction: IN - Relevant for: Peering sessions

Accept my own prefixes? Maybe ... maybe not

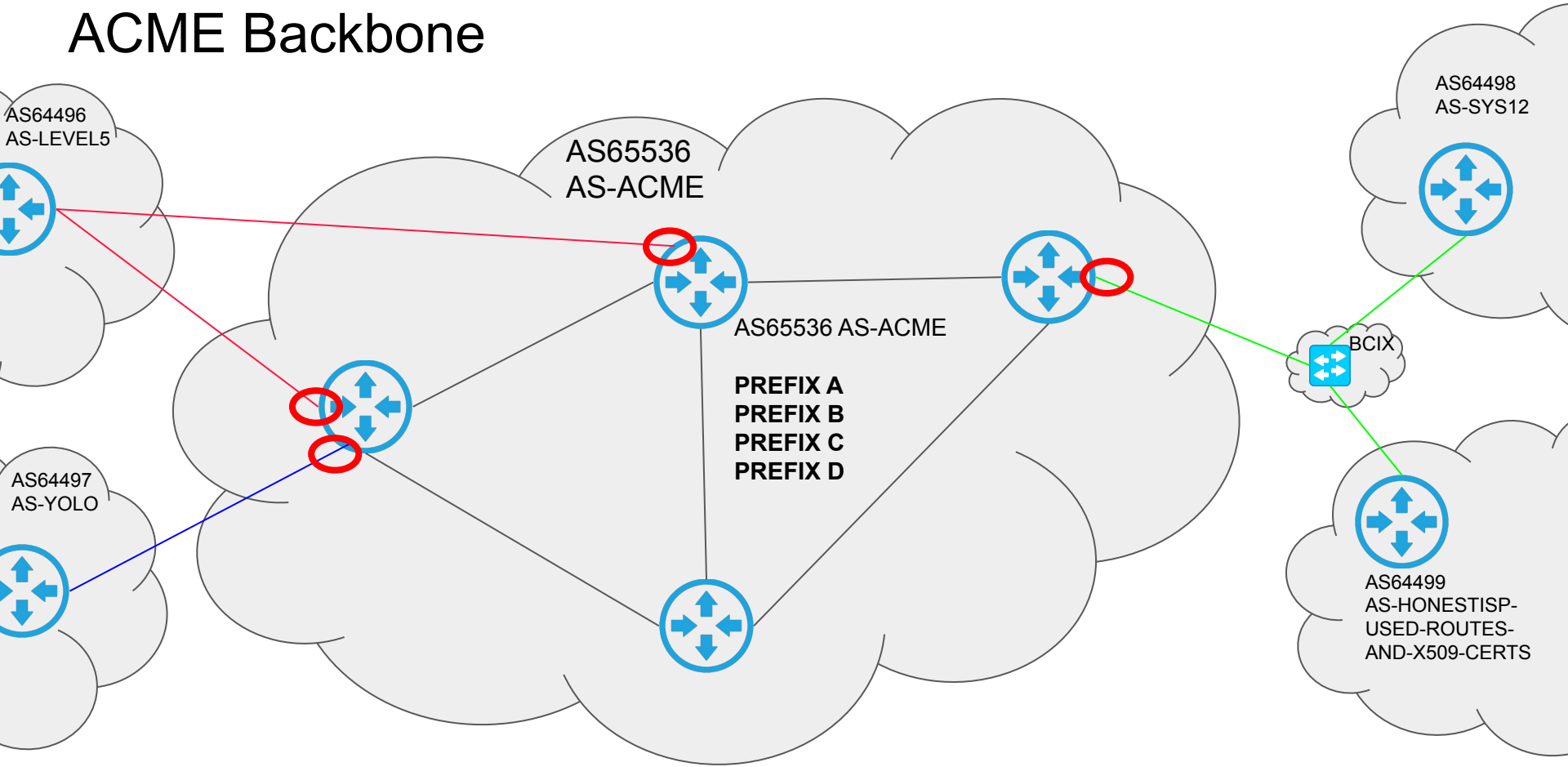
Topology: Island

-> You should only drop the prefixes of/at the respective Island

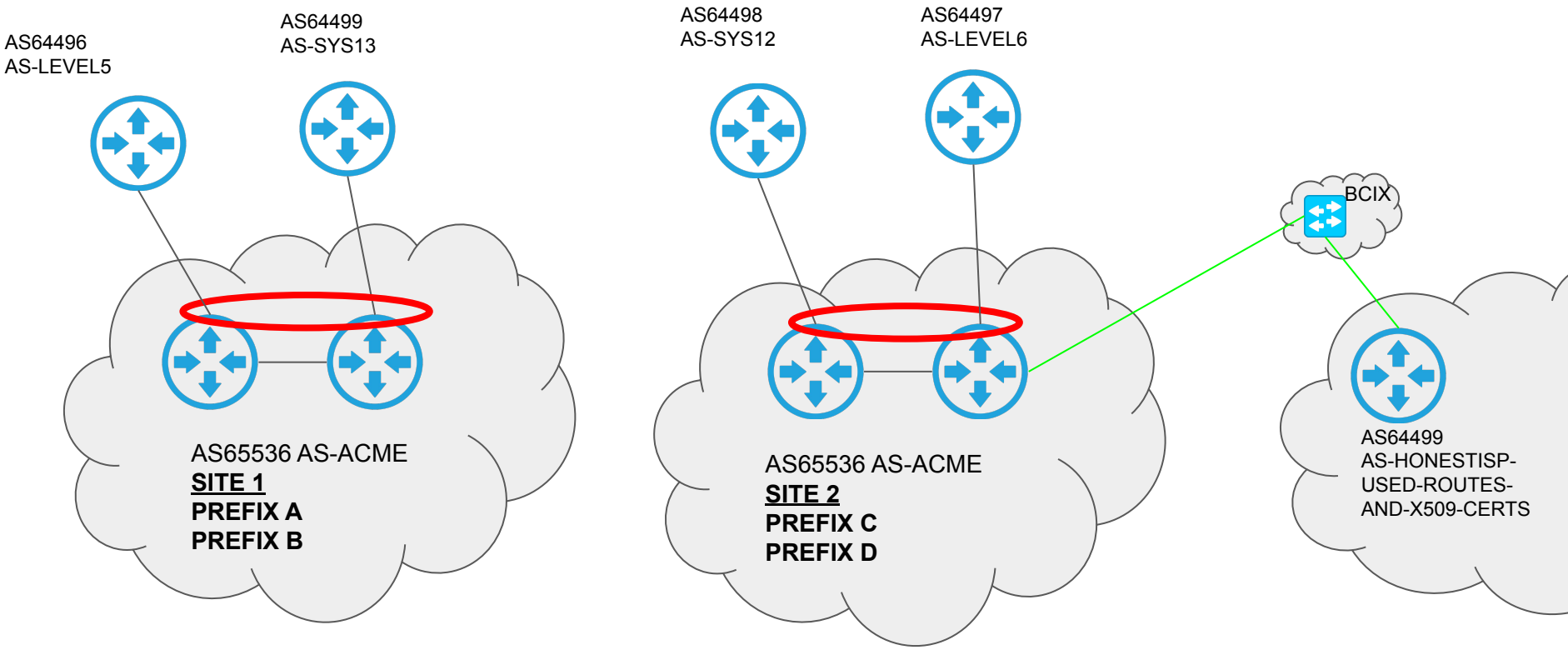
Topology: Backbone / Region(s)

-> Filter on all edges (Worldwide, Per Region/Cluster etc.)

ACME Backbone



ACME Islands



```
[edit policy-options]
prefix-list 4-MY-PREFIXES-GLOBAL {
    192.168.0.0/22;
    192.168.4.0/22;
    192.168.8.0/22;
    192.168.12.0/22;
}

[edit policy-options policy-statement 4-PEER-IN]
term REJECT-MY-PREFIXES {
    from {
        prefix-list-filter 4-MY-PREFIXES-GLOBAL orlonger;
    }
    then reject;
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

Reject Direct Peer Prefixes from RS [optional]

- Route Server (RS) peering has its advantages and disadvantages
- If you have a lot of direct peerings it makes sense NOT to accept prefixes of these peers via the route servers anymore
- But how? -> AS Path Filtering

💀 Do not apply to upstream sessions!

```
# Deny ASNs to which you already have direct Sessions
[edit policy-options as-path-group ASP-LOCAL-DIRECT-PEERINGS]
as-path PATH0 "(1234|5678|9012|3456|7890|12345|67890|1337) .*";
as-path PATH1 "(12340|56780|120|34560|38900|12045|890|31337) .*";

[edit policy-options policy-statement 4-PEER-IN]
term REJECT-RS-ROUTES-TO-DIRECT-PEERS {
    from as-path-group ASP-LOCAL-DIRECT-PEERINGS;
    then reject;
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

Reject and clean prefixes with own communities

- As soon as communities control the behavior of the network, they should be "protected".
- "foreign" prefixes with these communities must be rejected
- e.g. the communities that control prefix export

```
[edit policy-options]
community MY-COMMUNITIES members <MY ASN>:*;

[edit policy-options policy-statement 4-PEER-IN]
term SCRUB-COMMUNITIES {
    then community delete MY_COMMUNITIES;
}
```

Direction: IN / OUT - Relevant for: All eBGP Sessions

Add Origin Communities [optional]

- Some customers and peers appreciate certain "extra info"
- These can be easily attached to the routes in the form of Origin Communities
- There are no limits for creativity
- Nevertheless, one should not exaggerate

```
[edit policy-options]
community ORIGIN_TYPE_PEER members <YOUR ASN>:100;
community ORIGIN_REGION_EUROPE members <YOUR ASN>:110;
community ORIGIN_COUNTRY_DE members <YOUR ASN>:120;
community ORIGIN_CITY_BER members <YOUR ASN>:130;
community ORIGIN_IXP_BCIX members <YOUR ASN>:140;

[edit policy-options policy-statement 4-PEER-IN]
term ADD-ORIGIN-COMMUNTIY {
    then {
        community add ORIGIN_TYPE_PEER;
        community add ORIGIN_REGION_EUROPE;
        community add ORIGIN_COUNTRY_DE;
        community add ORIGIN_CITY_BER;
        community add ORIGIN_IXP_BCIX;
    }
}
```

Direction: IN / OUT - Relevant for: All eBGP sessions

Prefix Count and Limits

- No policy is perfect
- Even with a perfect filter policy a prefix limit is useful to save memory and CPU cycles
- Remember to set a "reasonable" time for automatic recovery.

👉 The prefix limit should be applied to received and NOT accepted prefixes

```
[edit groups BCIX protocols bgp group 4-BCIX neighbor XXX.XXX.XXX.XX]
description "BCIX PEER";
import [ MAINTENANCE-MODE 4-BASE-IN 4-PEER-IN DENY-ALL ];
family inet {
    unicast {
        prefix-limit {
            maximum 200;
            teardown {
                idle-timeout 1440;
            }
        }
    }
}
export [ MAINTENANCE-MODE 4-BASE-OUT 4-PEER-OUT DENY-ALL ];
peer-as <PEER ASN>;
```

Check AS-Path and Prefix List

- Finally ... are we almost done?
- AS Path and Prefix lists are the recommended acceptance criteria
- Prefix list Match has priority (AS Path is a nice plus)

You have to decide:

-> If you trust **ONLY** the peer (-> filter on ASnum of the peer)

-> Trust the **peer + downstream** (-> filter on AS-SET of the peer)

```
[edit policy-options]
as-path-group AS-PATH-AS65536 {
    as-path PATH0 "^65536(65536)*$";
}

[edit policy-options]
prefix-list 4-AS65536 {
    1.2.3.0/24;
    4.5.6.0/24;
}

[edit policy-options policy-statement 4-PEER-IN]
term ACCEPT-PEER {
    from {
        as-path-group AS-PATH-AS65536;
        prefix-list-filter 4-AS65536 orlonger;
    }
    then {
        metric XXX;
        local-preference 100;
        community add ANNOUNCE_TO_INTERNAL;
        community add ANNOUNCE_TO_CUSTOMER;
        accept;
    }
}
```

Direction: IN / OUT - Relevant for: All BGP sessions

```
# ACME AS (65536) + Kunden AS (65537)
[edit policy-options]
as-path-group AS-PATH-AS-ACME {
    as-path PATH0 "^65536(65536)*$";
    as-path PATH1 "^65536(.)*(65537)$";
}

[edit policy-options]
prefix-list 4-AS-ACME {
    1.2.3.0/24;
    4.5.6.0/24;
    7.8.9.0/24;
    9.8.7.0/24;
}

[edit policy-options policy-statement 4-PEER-IN]
term ACCEPT-PEER {
    from {
        as-path-group AS-PATH-AS-ACME;
        prefix-list-filter 4-AS-ACME orlonger;
    }
    then {
        metric XXX;
        local-preference 100;
        community add ANNOUNCE_TO_INTERNAL;
        community add ANNOUNCE_TO_CUSTOMER;
        accept;
    }
}
```

Direction: IN - Relevant for: All BGP sessions

Also useful

Maintenance Switch

- It can be very handy to take an IXP port or peer offline as "Graceful" as possible

- For example, due to maintenance or an incident

Killing me …

Softly: Graceful Shutdown (Local Pref 0)

Medium: Import/Export DENY

Hard: BGP Session shut/disable

Hardest: Shut the link or Box

```
# Config
show policy-options policy-statement MAINTENANCE-MODE
inactive: term ACTIVATE-MAINTENANCE {
    then reject;
}

# Config (SET)
set policy-options policy-statement MAINTENANCE-MODE term ACTIVATE-MAINTENANCE then reject
deactivate policy-options policy-statement MAINTENANCE-MODE term ACTIVATE-MAINTENANCE


# How to activate
activate policy-options policy-statement MAINTENANCE-MODE term ACTIVATE-MAINTENANCE
```

Direction: IN / OUT - Relevant for: Sessions that are to be drained in case of maintenance

Support of Graceful Shutdown (RFC 8326)

- In addition to the "Import/Export Deny" you can implement RFC8326

- How it works
  Community = 65535:0?
  -> Local-Pref 0

- This leads to the route not being used (anymore)

👉 Default with Junos >19.1!

```
[edit policy-options]
community GRACEFUL_SHUTDOWN members 65535:0;

# Set Community (Trigger Graceful Shutdown to Peers)
set policy-options policy-statement 4-PEER-OUT term ANNOUNCE then community add GRACEFUL_SHUTDOWN

# Add support for Pre Junos 19.1 Router
[edit policy-options policy-statement MAINTENANCE-MODE]
term GRACEFUL-SHUT {
    from {
        community GRACEFUL_SHUTDOWN;
    }
    then {
        local-preference 0;
        next term;
    }
}
```

Direction: IN - Relevant for: All eBGP sessions

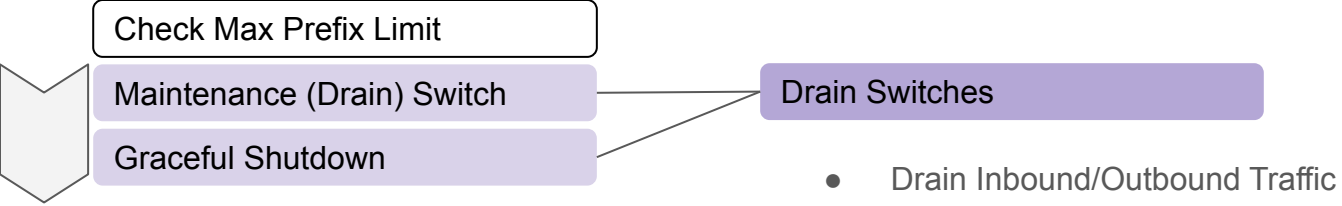
Policy Building

Sequence and selection

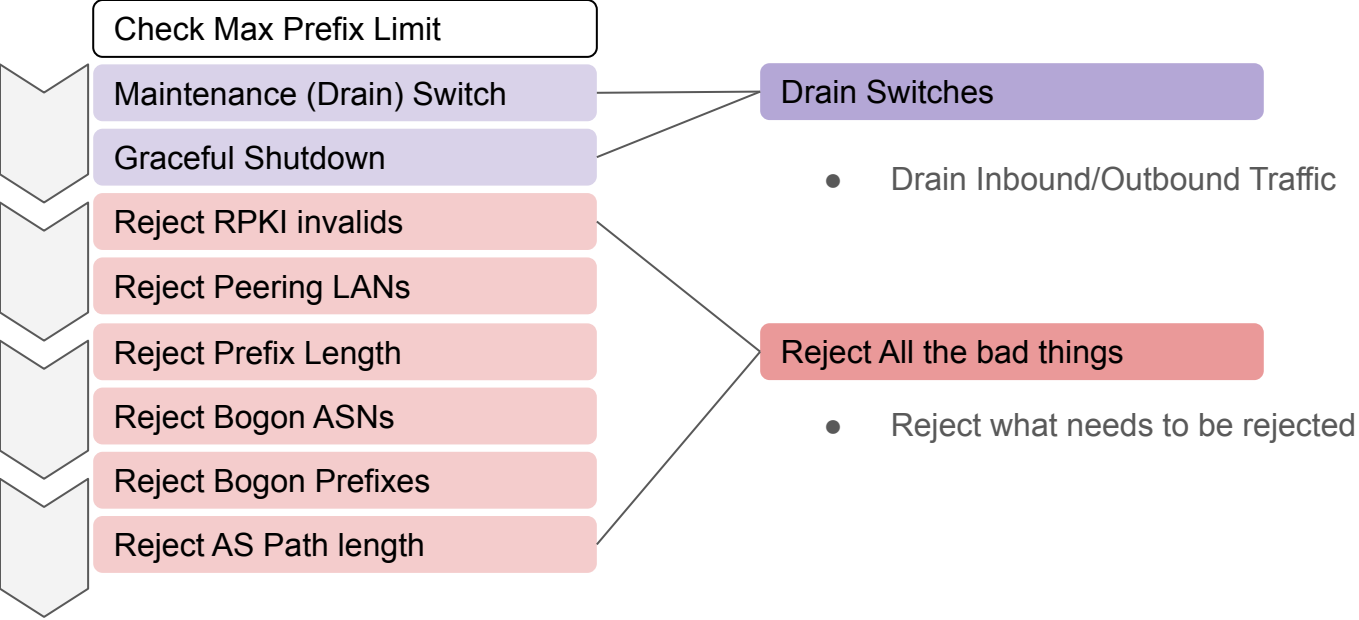
Buildings Blocks by:

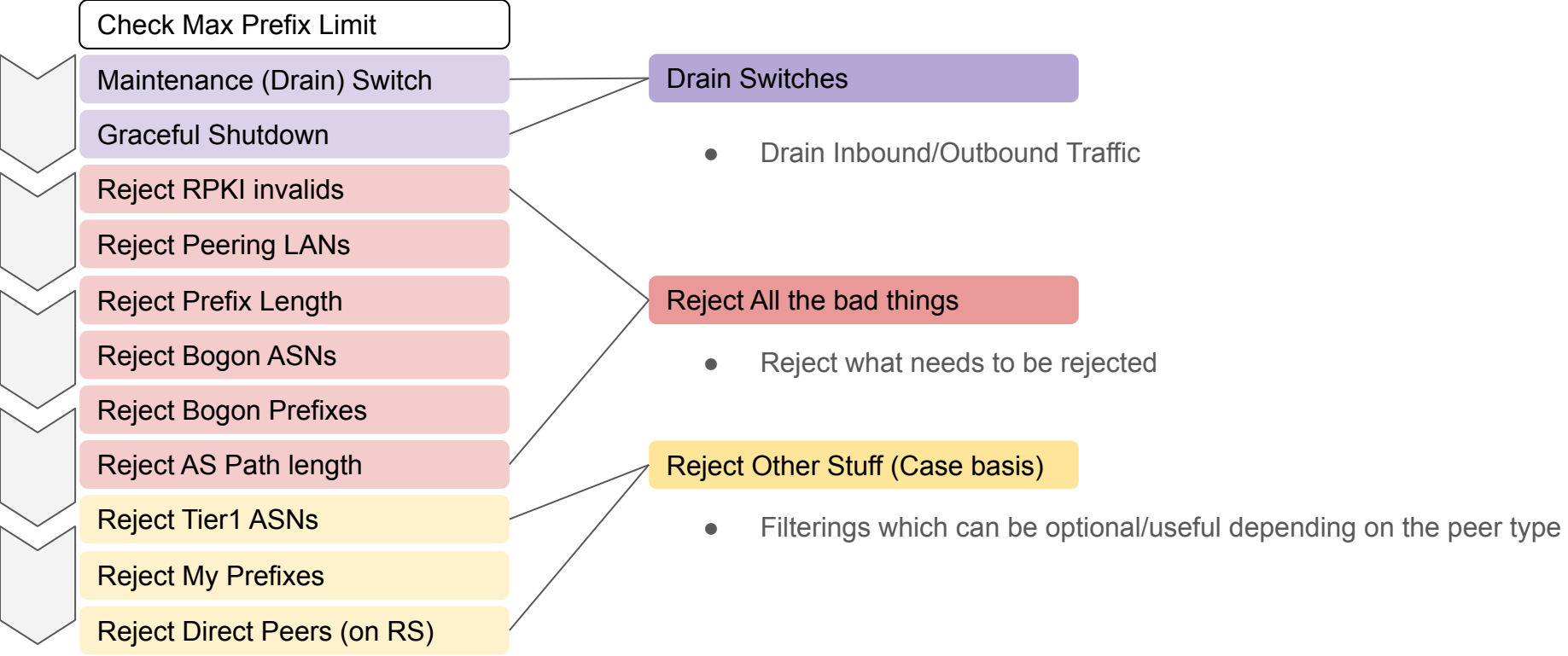
- Same principle / type of peer (downstream, upstream, peering, etc.).
- Policy Action
- Performance / "hit" probabilities
- Automation templates / logic

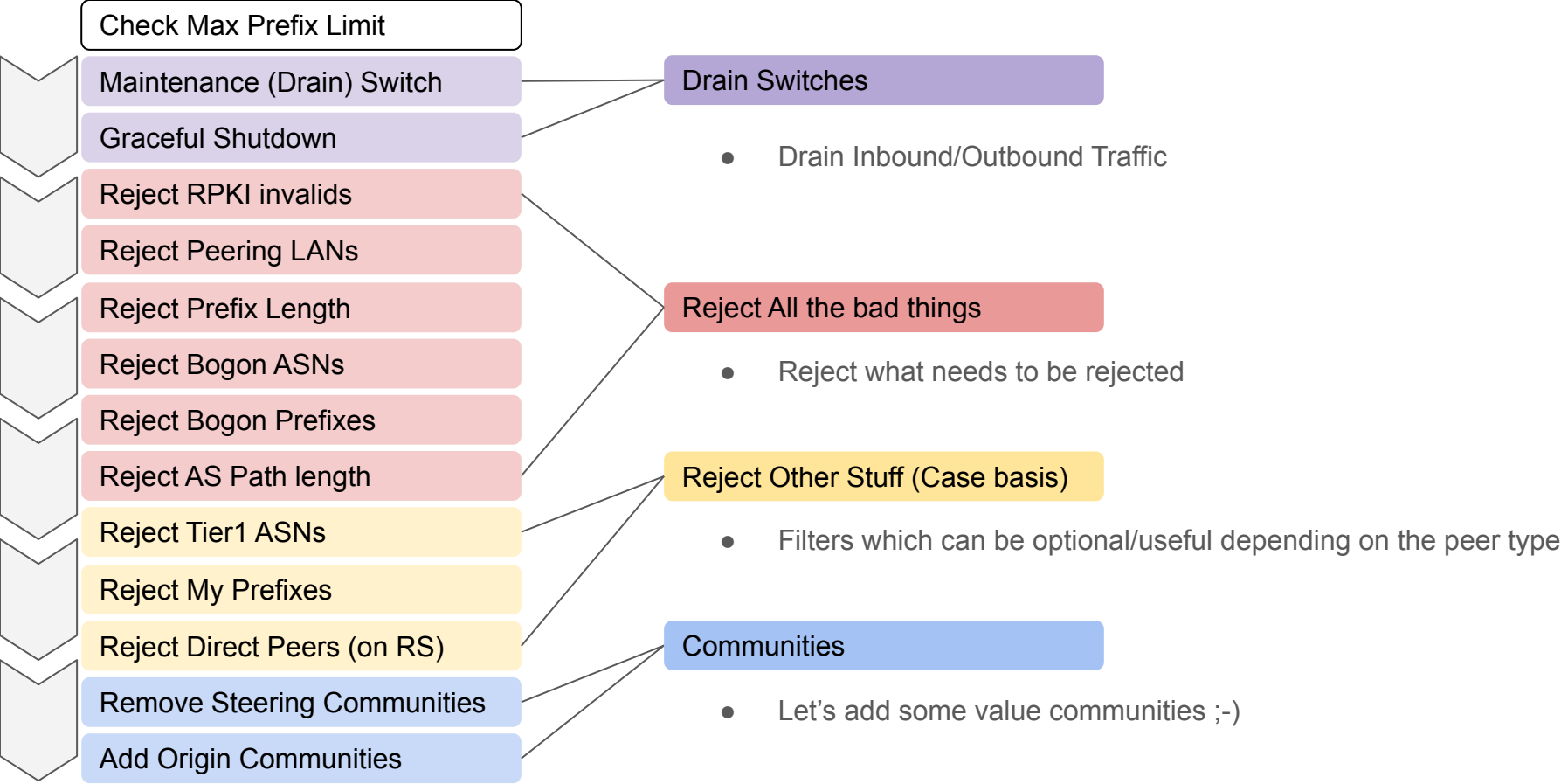
👉 “Think before committing”

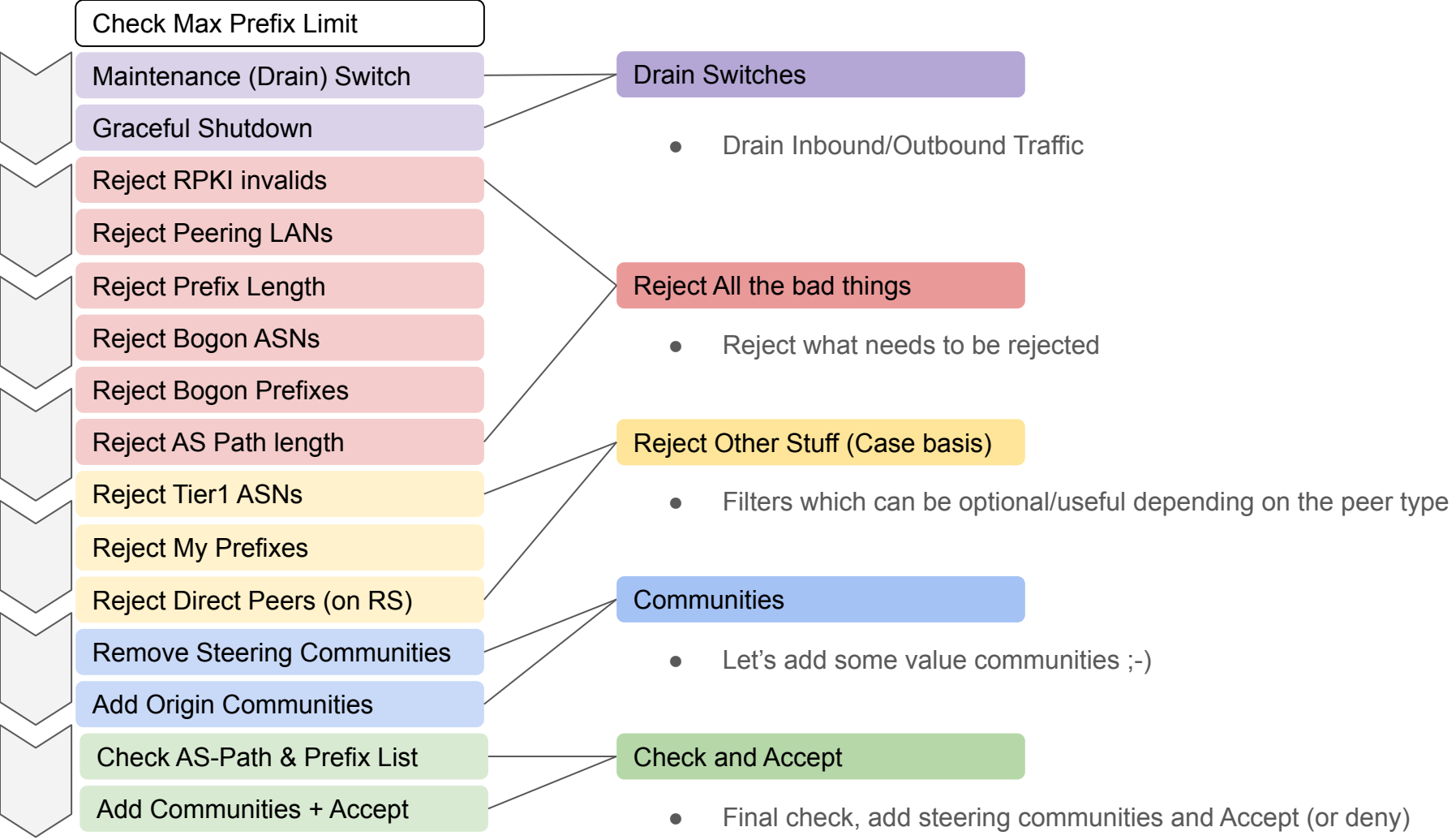


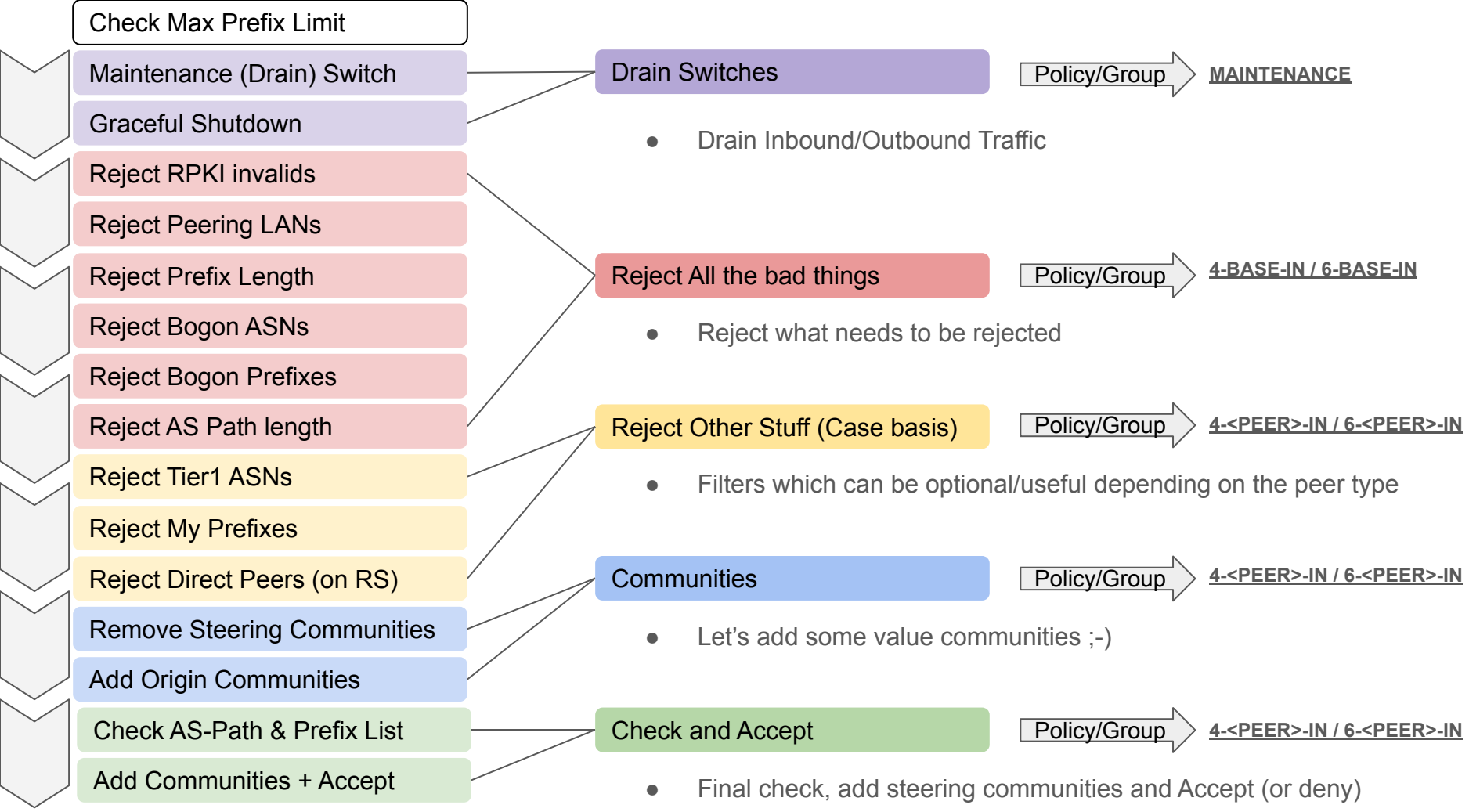
- Drain Inbound/Outbound Traffic


Check Max Prefix Limit
Maintenance (Drain) Switch
Graceful Shutdown
Reject RPKI invalids
Reject Peering LANs
Reject Prefix Length
Reject Bogon ASNs
Reject Bogon Prefixes
Reject AS Path length
Drain Switches
Drain Inbound/Outbound Traffic
Reject All the bad things
Reject what needs to be rejected


Check Max Prefix Limit
Maintenance (Drain) Switch
Graceful Shutdown
Reject RPKI invalids
Reject Peering LANs
Reject Prefix Length
Reject Bogon ASNs
Reject Bogon Prefixes
Reject AS Path length
Reject Tier1 ASNs
Reject My Prefixes
Reject Direct Peers (on RS)
Drain Switches
Drain Inbound/Outbound Traffic
Reject All the bad things
Reject what needs to be rejected
Reject Other Stuff (Case basis)
Filterings which can be optional/useful depending on the peer type


Check Max Prefix Limit
Maintenance (Drain) Switch
Graceful Shutdown
Reject RPKI invalids
Reject Peering LANs
Reject Prefix Length
Reject Bogon ASNs
Reject Bogon Prefixes
Reject AS Path length
Reject Tier1 ASNs
Reject My Prefixes
Reject Direct Peers (on RS)
Remove Steering Communities
Add Origin Communities
Drain Switches
Drain Inbound/Outbound Traffic
Reject All the bad things
Reject what needs to be rejected
Reject Other Stuff (Case basis)
Filters which can be optional/useful depending on the peer type
Communities
Let’s add some value communities ;-)


Check Max Prefix Limit
Maintenance (Drain) Switch
Graceful Shutdown
Reject RPKI invalids
Reject Peering LANs
Reject Prefix Length
Reject Bogon ASNs
Reject Bogon Prefixes
Reject AS Path length
Reject Tier1 ASNs
Reject My Prefixes
Reject Direct Peers (on RS)
Remove Steering Communities
Add Origin Communities
Check AS-Path & Prefix List
Add Communities + Accept
Drain Switches
Drain Inbound/Outbound Traffic
Reject All the bad things
Reject what needs to be rejected
Reject Other Stuff (Case basis)
Filters which can be optional/useful depending on the peer type
Communities
Let’s add some value communities ;-)
Check and Accept
Final check, add steering communities and Accept (or deny)


Check Max Prefix Limit
Maintenance (Drain) Switch
Graceful Shutdown
Reject RPKI invalids
Reject Peering LANs
Reject Prefix Length
Reject Bogon ASNs
Reject Bogon Prefixes
Reject AS Path length
Reject Tier1 ASNs
Reject My Prefixes
Reject Direct Peers (on RS)
Remove Steering Communities
Add Origin Communities
Check AS-Path & Prefix List
Add Communities + Accept
Drain Switches
Policy/Group
MAINTENANCE
Drain Inbound/Outbound Traffic
Reject All the bad things
Policy/Group
4-BASE-IN / 6-BASE-IN
Reject what needs to be rejected
Reject Other Stuff (Case basis)
Policy/Group
4-<PEER>-IN / 6-<PEER>-IN
Filters which can be optional/useful depending on the peer type
Communities
Policy/Group
4-<PEER>-IN / 6-<PEER>-IN
Let's add some value communities ;-)
Check and Accept
Policy/Group
4-<PEER>-IN / 6-<PEER>-IN
Final check, add steering communities and Accept (or deny)

```
[edit protocols bgp group 4-BCIX]
remove-private {
    all;
}
multipath;
neighbor 193.178.185.5 {
    description "BCIX BCIX-RS";
    import [ MAINTENANCE 4-BASE-IN 4-BCIX-RS-IN DENY-ALL ];
    family inet {
        unicast {
            prefix-limit {
                maximum 160000;
                teardown {
                    idle-timeout 1440;
                }
            }
        }
    }
    export [ MAINTENANCE-MODE 4-BASE-OUT 4-BCIX-RS-OUT DENY-ALL ];
    peer-as 16374;
}
```

But what about the OUT policy?

Many of the "IN" filters can also be used "OUT"

- Basic BCP's / "Features"
  - No-export (do NOT export to other networks)
  - No-advertise (do NOT advertise)
  - No-advertise-to (do NOT advertise to AS XXXXX)
  - Prepend once (add your ASN one time)
  - Prepend twice (add your ASN two times)
- Export Blackhole Routes
- Export Filter (similar to "IN")
- Prefixes should be announced on a community basis

```
[edit policy-options]
community GRACEFUL_SHUTDOWN members 65535:0;
community BLACKHOLE_DEFAULT members 65535:666;
community <PEER ASN>_NO_ADVERTISE members 64512:<PEER ASN>;
community <PEER ASN>_1PREPEND members 64513:<PEER ASN>;
community <PEER ASN>_2PREPEND members 64514:<PEER ASN>;

[edit policy-options policy-statement 4-BASE-OUT]
term NO-EXPORT {
    from community no-export;
    then reject;
}
term NO-ADVERTISE {
    from community no-advertise;
    then reject;
}
term REJECT-BOGON-ASNS {
    from as-path-group BOGON-ASNS;
    then reject;
}
term REJECT-BOGONS {
    from  prefix-list-filter 4-BOGON-PREFIXES orlonger;
    then reject;
}
```

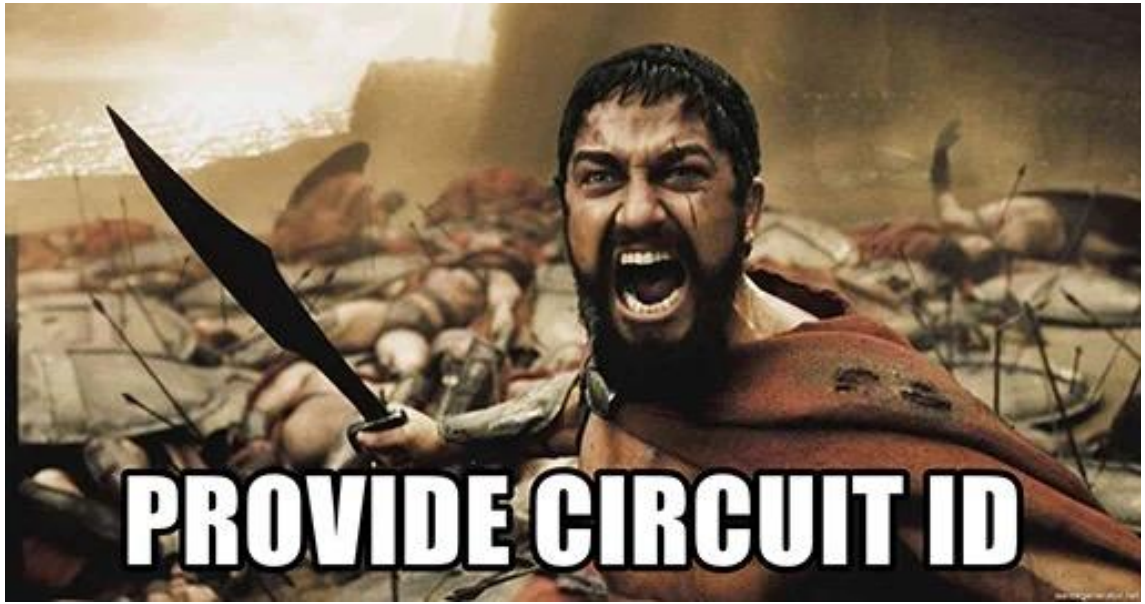
```
term ANNOUNCE-BLACKHOLE {
    from {
        community BLACKHOLE_DEFAULT;
        route-filter 0.0.0.0/0 prefix-length-range /32-/32;
    }
    then {
        community delete MY_COMMUNITIES;
        community add IXP-PEER-SPECIFIC-BLACKHOLE-COMMUNITY-IF-NEEDED
        accept;
    }
}
term FILTER-PREFIX-LENGTH {
    from {
        route-filter 0.0.0.0/0 prefix-length-range /0-/7;
        route-filter 0.0.0.0/0 prefix-length-range /25-/32;
    }
    then reject;
}
term <PEER ASN>-NO-ADVERTISE {
    from community <PEER ASN>_NO_ADVERTISE;
    then reject;
}
```

```
term PREPEND1 {
    from community <PEER ASN>_1PREPEND;
    then {
        as-path-prepend <MY ASN>;
        next term;
    }
}
term PREPEND2 {
    from community <PEER ASN>_2PREPEND;
    then {
        as-path-prepend "<MY ASN> <MY ASN>";
        next term;
    }
}
term ANNOUNCE {
    from community ANNOUNCE_TO_PEER;
    then {
        community delete MY_COMMUNITIES;
        next-hop self;
        accept;
    }
}
```

Lunch

Session 3 start -> 13:30

Session 3
Forwarding filtering



Agenda

- **Protecting the Control Plane**
  - Control Plane vs. Forwarding Plane, Punting, DoS
  - Automatched/Wildcard Apply Groups/Lists
  - lo0 Protection ACL for Juniper
- **Forwarding Plane Filtering**
  - Why filter?
  - BCP 38
  - Where to filter what?
  - Traffic Ingress filtering
    - Customer
    - Edge/Peering/Upstream
    - Internal “Services”
  - Traffic Egress filtering
  - Reverse Path Filtering
    - Loose vs. Strict
    - Where? where not?
    - By feature or statically generated

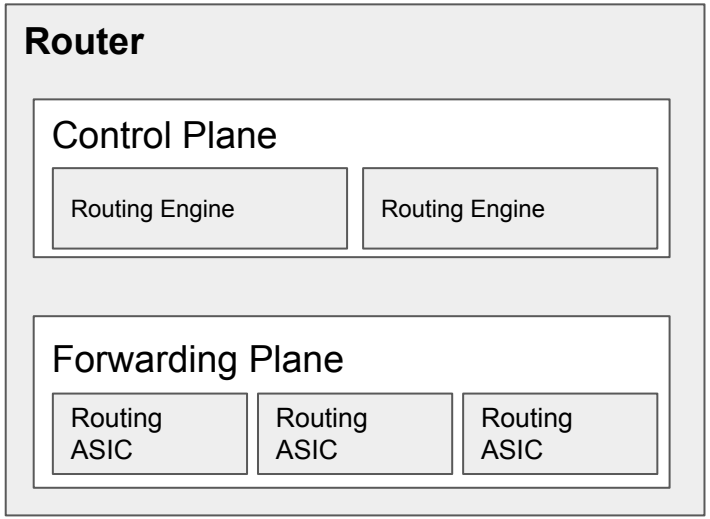
Control Plane vs. Forwarding Plane, Punting, DoS (FW)

Control Plane:

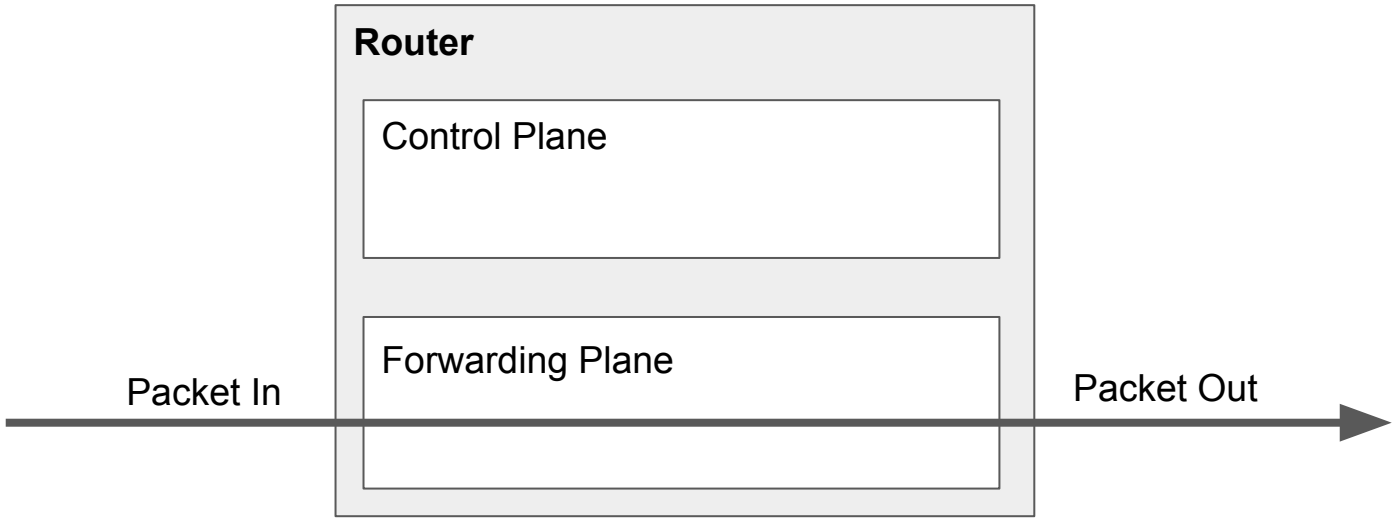
- “Brain” of our Router
- General purpose computing
- Runs our routing protocols
- (Almost) no packet handling
- Programs the forwarding plane

Forwarding Plane:

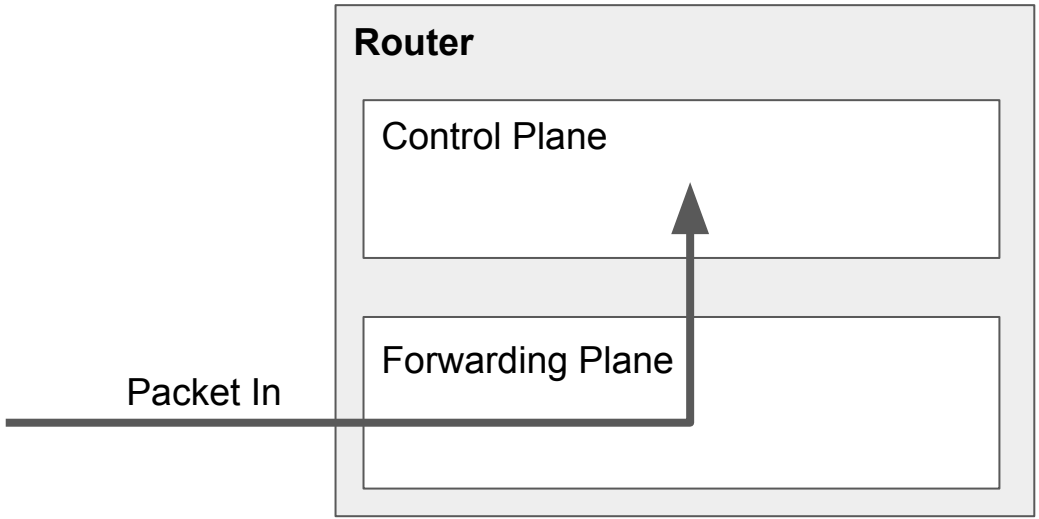
- Specialized routing chips
- Handles all our packets
- Connects Front Ports



Normal Packet Flow



Punted Packet Flow



We need to protect our control plane

- Control plane is a critical component
- Can act as a target for DDoS attacks
  - If successful -> router down
- We can let the forwarding plane filter “in hardware”
- Implemented as
  - CoPP ACL on Cisco IOS or
  - lo0 firewall filter at JunOS

We need to protect our control plane

```
set interfaces lo0 unit 0 family inet filter input-list [ filters belong here ]
set interfaces lo0 unit 0 family inet6 filter input-list [ filters belong here ]
```

We need to protect our control plane

Rules

- *Reject all* at the end
- Selectively allow used protocols (SSH, ICMP, BGP, BFD, etc.)
  - Allow only from hosts that need to talk to our router
  - *apply-path* is your friend
- Apply strict policer!

Automatched/Wildcard Apply Path/Lists

```
prefix-list ROUTER_RPKI_SERVER {
      apply-path "routing-options validation group <*> session <*>";
}
```

```
> show policy-options prefix-list ROUTER_RPKI_SERVER | display inheritance
##
## apply-path was expanded to:
##    62.176.246.69/32;
##    62.176.246.165/32;
##    62.176.246.166/32;
##
apply-path "routing-options validation group <*> session <*>";
```

Juniper lo0 ACL - Allow Local

- Sometimes local traffic (within control plane needs to be permitted)

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term LOCAL-ALLOW
term LOCAL-ALLOW {
    from {
        source-address {
            127.0.0.0/8;
        }
    }
    then accept;
}
```

Juniper lo0 ACL - SSH Access

- Allow ssh access from privileged IP’s/prefixes
- No rate limiting for maximum image copy speed
  - or go for a pps limit

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term SSH-ALLOW
term SSH-ALLOW {
    from {
        source-prefix-list {
            4-MGMT-HOSTS;
        }
        protocol tcp;
        destination-port ssh;
    }
    then accept;
}
```

Juniper lo0 ACL - Allow jumping between routers

- This is optional but maybe useful in corner cases

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term SSH-COREJUMP-ALLOW
term SSH-COREJUMP-ALLOW {
    from {
        source-address {
            1.2.3.0/23;
            4.5.6.0/28;
            7.8.9.0/28;
            10.11.12.0/28;
        }
        protocol tcp;
        port ssh;
    }
    then accept;
}
```

Juniper lo0 ACL - Accept BGP connections

- Allow BGP connections but only from configured peers!

```
set policy-options prefix-list BGP-NEIGHBORS apply-path "protocols bgp group <*> neighbor <*>"

{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term BGP-ALLOW
term BGP-ALLOW {
    from {
        source-prefix-list {
            BGP-NEIGHBORS;
        }
        protocol tcp;
        port bgp;
    }
    then accept;
}
```

Juniper lo0 ACL - Allow monitoring

- For legacy monitoring allow some hosts to do SNMP

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term SNMP-ALLOW
term SNMP-ALLOW {
    from {
        source-prefix-list {
            SNMP-CLIENTS;
        }
        protocol udp;
        port snmp;
    }
    then {
        policer RATE-LIMIT-50;
        count SNMP-ALLOW;
        accept;
    }
}
```

Juniper lo0 ACL - What about DNS?

- Of course you know all the IP addresses but just in case …

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term DNS-ALLOW
term DNS-ALLOW {
    from {
        source-prefix-list {
            DNS-HOSTS;
        }
        protocol udp;
        port domain;
    }
    then {
        policer RATE-LIMIT-5;
        accept;
    }
}
```

Juniper lo0 ACL - What about time?

- To allow NTP responses

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term NTP-ALLOW
term NTP-ALLOW {
    from {
        source-prefix-list {
            NTP-HOSTS;
        }
        protocol udp;
        port ntp;
    }
    then {
        policer RATE-LIMIT-5;
        accept;
    }
}
```

Juniper lo0 ACL - VRRP

- Allow VRRP Multicast Adress

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term VRRP-ALLOW
term VRRP-ALLOW {
    from {
        destination-address {
            224.0.0.18/32;
        }
    }
    then accept;
}
```

Juniper lo0 ACL - RSVP-MPLS Allow

- Allow RSVP IP's/Prefixes to talk to each other

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term RSVP-ALLOW
term RSVP-ALLOW {
    from {
        source-prefix-list {
            4-RSVP-HOSTS;
        }
        protocol rsvp;
    }
    then accept;
}
```

Juniper lo0 ACL - MPLS-SP Allow

- Allow MPLS Self ping for LSP testing

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term MPLS-SP-ALLOW
term MPLS-SP-ALLOW {
    from {
        source-prefix-list {
            4-RSVP-HOSTS;
        }
        protocol udp;
        destination-port 8503;
    }
    then accept;
}
```

Juniper lo0 ACL - Allow ICMP

- Give internal ICMP traffic it’s own rate limiter bucket

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term ICMP-LIMIT-INTERNAL
term ICMP-LIMIT-INTERNAL {
    from {
        source-prefix-list {
            SNMP-CLIENTS;
        }
        protocol icmp;
        icmp-type [ echo-request echo-reply unreachable time-exceeded source-quench ];
    }
    then {
        policer RATE-LIMIT-1;
        accept;
    }
}
```

Juniper lo0 ACL - Allow ICMP (external)

- Allow specific ICMP for everyone … it won't hurt (especially not rate limited)

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term ICMP-LIMIT
term ICMP-LIMIT {
    from {
        protocol icmp;
        icmp-type [ echo-request echo-reply unreachable time-exceeded source-quench ];
    }
    then {
        policer RATE-LIMIT-1;
        accept;
    }
}
```

Juniper lo0 ACL - Allow BFD Speakers

- Using BFD? You may want to allow it.

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term BFD-ALLOW
term BFD-ALLOW {
    from {
        source-prefix-list {
            BGP-NEIGHBORS;
            BFD-NEIGHBORS;
        }
        protocol udp;
        port [ 3784 ];
    }
    then accept;
}
```

Juniper lo0 ACL - What RPKI / RTR Sessions?

- Guess we need to cover that as well

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term RPKI-ALLOW
term RPKI-ALLOW {
    from {
        source-prefix-list {
            RPKI-RTR-SERVERS;
        }
        protocol tcp;
        source-port [ 8282 3323 323 ];
    }
    then accept;
}
```

Juniper lo0 ACL - Need some GRE Tunnel?

- Guess we need to cover that as well

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term GRE-ALLOW
term GRE-ALLOW {
    from {
        source-prefix-list {
            4-GRE-DESTINATIONS;
        }
        protocol gre;
    }
    then accept;
}
```

Juniper lo0 ACL - Do not forget

- Deny the rest

```
{master}[edit firewall family inet filter 4-MGMT-ACL-ROUTER]
vpetzholtz@area51# show term DENY-OTHER
term DENY-OTHER {
    then {
        count DENY-OTHER;
        discard;
    }
}
```

Juniper lo0 ACL - Overview

Allow Loopback - Allow Local (127.0.0.0/X) depends on $Vendor

Management - Allow SSH/HTTPs for privileged Hosts/Networks

Allow BGP & BFD - Accept BGP/BFD Session from peers

RPKI / RTR - Allow RTR Sessions for Validator Sessions

NTP & DNS - Allow NTP/DNS Responses

Tunneling? - Allow GRE/$Tunnel Traffic from peers

ICMP - Please allow this (rate limited)

DENY THE REST!

Forwarding Filter

Why filter?

- Prohibit packet spoofing
  - Do not trust your clients :-D
- Filter forged traffic
  - At the edge
- Protection against DDoS
  - As much as possible

- It relies on the providers to search, find and fix loopholes in configs
- Can be accomplished by various techniques
  - by feature (uRPF)
  - “manually” configured

BCP38 aka. RFC2827

Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source Address Spoofing.

http://bcp38.info

https://www.rfc-editor.org/rfc/rfc2827

MANRS Guide:

MANRS-Network-Implementation-Guide.pdf

Where to filter what?

Inbound

- Edge (Peers, Transit, Multi-Homed Customers, etc.)
  - Discard bogon traffic
  - Discard (spoofed) traffic from own prefixes
  - Optional: Policer for known attack traffic
- Single-Homed Customers/Server Networks
  - Only allow traffic from the respective prefixes

Discard Bogon Traffic

```
term BOGON-SRC {
    from {
        source-prefix-list {
            4-BOGON-PREFIXES;
        }
    }
    then discard;
}
term BOGON-DST {
    from {
        destination-prefix-list {
            4-BOGON-PREFIXES;
        }
    }
    then discard;
}
```

Our own prefixes

- We want to drop spoofed traffic with our own src addresses
- We want to accept traffic from directly connected transfer networks that we hand out
  - apply-path comes in handy again

Allow connected transfer networks

```
    prefix-list 4-ROUTER-CONNECTED {
        apply-path "interfaces <*> unit <*> family inet address <*>";
    }
```

```
[...]
    term ALLOW-TRANSFER {
        from {
            source-prefix-list {
                4-ROUTER-CONNECTED;
            }
        }
        then accept;
    }
[...]
```

Discard edge inbound traffic from own prefixes

```
term OWN-SRC {
    from {
        source-prefix-list {
            4-MY-PREFIXES-GLOBAL;
        }
    }
    then discard;
}
```

Where to filter what?

Outbound

Theoretically not needed if you filter properly on **all** in-bound interfaces

If that’s not possible:

- Drop packets from bogon IPs (Both src and dst)
- Or, if you have no downstreams: Drop everything but own src IPs

uRPF (Unicast Reverse Path Filtering)

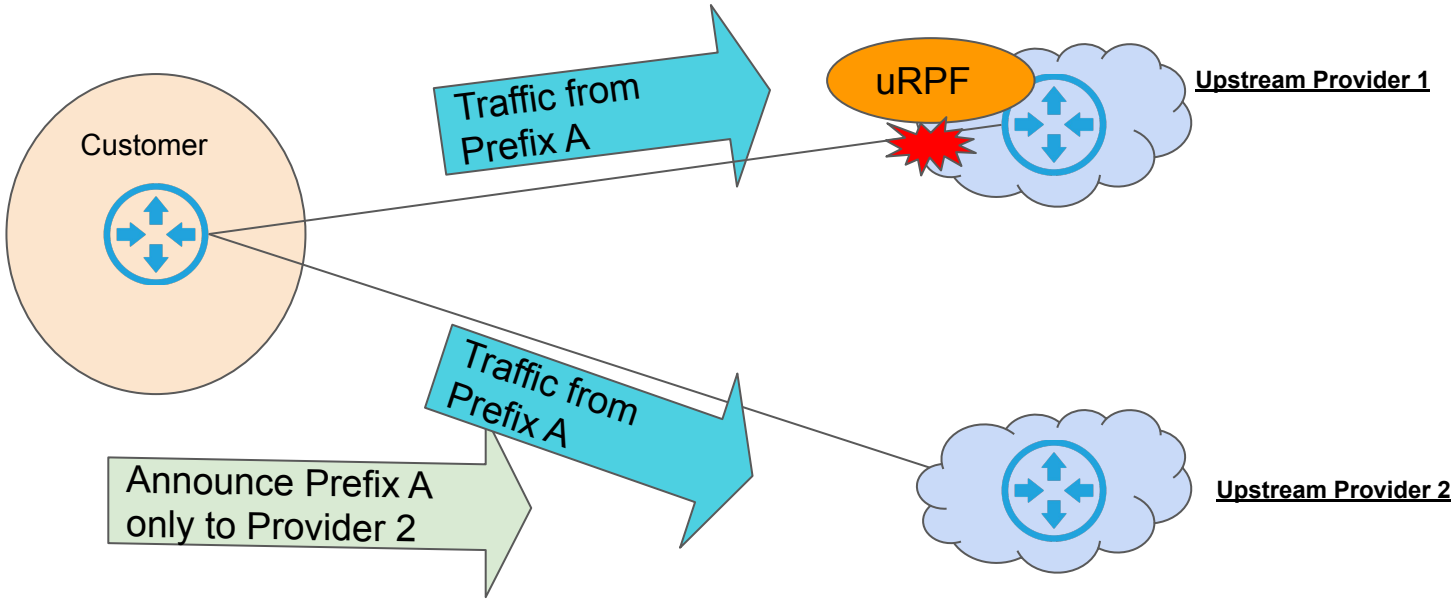
- Performs a lookup in the routing table to check if there is a valid route for the source IP
- Simplifies filtering a lot
- Modes
  - Strict
  - Loose

uRPF Strict Mode

- The router checks if there is any active route in the routing table **that points towards the interface** from which the packet is coming.
- If there are multiple selected routes (ECMP) all of those interfaces are allowed to originate the packets.
- Useful for:
  - Server Networks
  - Single-Homed Customers
  - Access Networks/Subscribers

Strict Mode (VP)

- What would strict mean for Transit Customers?



uRPF - Strict Mode

uRPF - Loose Mode

- The router checks if there is **any active** matching entry in the routing table, independent of what the next hop interface is.
- Useful for:
  - Peering
  - Transit
  - Multi-Homed Customers

Enable uRPF

```
interfaces {
    ae0 {
        unit 0 {
            family inet {
                rpf-check {
                    mode strict;
                }
                address <addr>;
            }
[...]
```

Optional: Police evil traffic

There are protocols that are notoriously vulnerable to reflection attacks.

If you want you can configure a policer to protect your internal links from congestion.

Examples:

- UDP Fragments
- DNS
- SNMP
- LDAP
- Memcached
- etc.

Police Evil Traffic

```
policer POLICER_10KPPS {
    if-exceeding-pps {
        pps-limit 10k;
        packet-burst 5k;
    }
    then discard;
}

policer POLICER_100KPPS {
    if-exceeding-pps {
        pps-limit 100k;
        packet-burst 20k;
    }
    then discard;
}
```

```
filter DROP_EVIL {
    term POLICE-FRAGMENTS {
        from {
            is-fragment;
        }
        then {
            policer POLICER_100KPPS;
            loss-priority high;
            accept;
        }
    }
    term POLICE-NTP {
        from {
            protocol udp;
            source-port 123;
        }
        then {
            policer POLICER_10KPPS;
            loss-priority high;
            accept;
        }
    }
```

Add exceptions for certain IPs

```
term POLICE-DNS {
    from {
        destination-prefix-list {
            DNS_RESOLVER_SOURCE_V4 except;
        }
        protocol udp;
        source-port 53;
    }
    then {
        policer POLICER_10KPPS;
        loss-priority high;
        accept;
    }
}
```

```
    term POLICE-DNS-RESOLVER {
        from {
            destination-prefix-list {
                DNS_RESOLVER_SOURCE_V4;
            }
            protocol udp;
            source-port 53;
        }
        then {
            policer POLICER_100KPPS;
            accept;
        }
}
```

Assemble your interface config

```
interfaces {
    ae0 {
        unit 0 {
            family inet {
                  rpf-check {
                mode loose;
                  }
                  filter {
                      input-list [ DROP_BOGONS DROP_EVIL ALLOW_TRANSFER DROP_OWN ];
                      output-list [ DROP_BOGONS ];
                  }
            address <addr>;
            }
[...]
```

Thank you for your Attention!

Quiz starts after a short break! (5 Mins)

Quiz starts now!

https://forms.gle/gshq17NpgsGv1KHu5

Q & A (fix me)

Slides:
https://drive.google.com/file/d/1QwLeiCSRsxkJ9CKHuvZHyQcOxs2QXCh9/view

Thank you for your Attention!

QR-Code / Link für PDF Download

Sourcen/Links

https://access.ripe.net/ / https://my.ripe.net/#/rpki

https://www.peeringdb.com

https://github.com/interdotlink/ripe-updater

https://github.com/denog/routing-bcp

http://bgpfilterguide.nlnog.net/

https://tools.ietf.org/html/bcp194

Our own networks

Meme Storage