# Welcome to "Introduction to BGP"

## DENOG18 Essen

- Please find a seat
- **Download** the training material:
  - https://go.de-cix.net/bgpzip
- Please take a router sheet (in front)
- We are starting at 11:00
- Lunchbreak 12:30-14:00

DE CIX

# *BGP for networks who peer*

## DENOG18

**Wolfgang Tremmel**
**academy@de-cix.net**



DE-CIX

*Where networks meet*

# About me



- Wolfgang Tremmel
- studied Informatik (Uni Karlsruhe)
  - Degree: Diploma (1994)
- Network Engineer at XLink
  - Since 1996 Director NOC
  - Since 2000 Senior Network Planner DSL at kpn/Qwest
- 2001 - 2005 Director Network Planning at VIA NET.WORKS
- 2006 - 2016 Manager Customer Support at DE-CIX
- since 2016: Head of DE-CIX Academy

I am a **Network Engineer**

DE CIX

wolfgangtremmel1966

@wtremmel@hessen.social

# DE-CIX Academy

## Learn from the experts

- Videos about topics related to IP networks, routing, peering
- Seminar(s) and workshops about BGP
- White Papers for download
- de-cix.net/academy

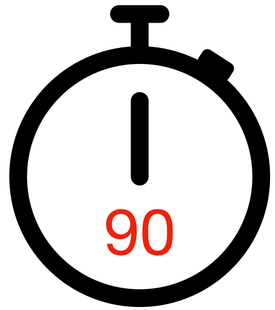



DE CIX

# Other versions of this presentation

# BGP in 90 minutes

## The short version




- Length: 90-120 minutes
- Features:
  - me talking
  - you asking questions
- Covers:
  - The very basics of BGP
  - Up and including BGP best path selection






- Now available on YouTube!

DE CIX

# BGP 4-5 hour workshop

## What we do today

- Length: 4-5 hours, including at least one break
- Happened a number of times at workshop Sunday at DENOG
- Features:
  - Me talking
  - You asking questions
  - Limited number of lab experiments using FRRouting
- Covers:
  - The very basics of BGP
  - Up and including BGP best path selection





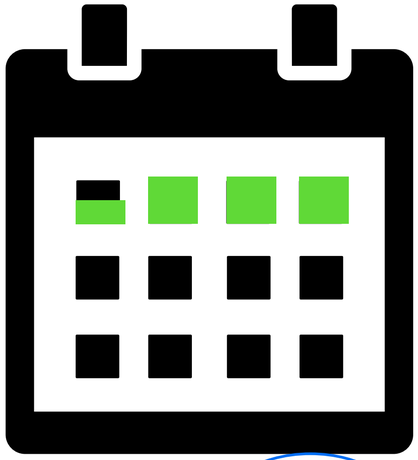






# 3.5 Day BGP Seminar

## All and everything






- Length: 3.5 days, starting Monday noon, finishing Thursday late afternoon,
- Classroom seminar, max. 14 attendees
- Features:
  - Me talking
  - You asking questions
  - Extensive number of lab experiments using FRRouting
- Covers:
  - All of BGP
  - Including BGP Security, Traffic Engineering, Peering Relationships
  - Tools useful for BGP and peering



DE CIX

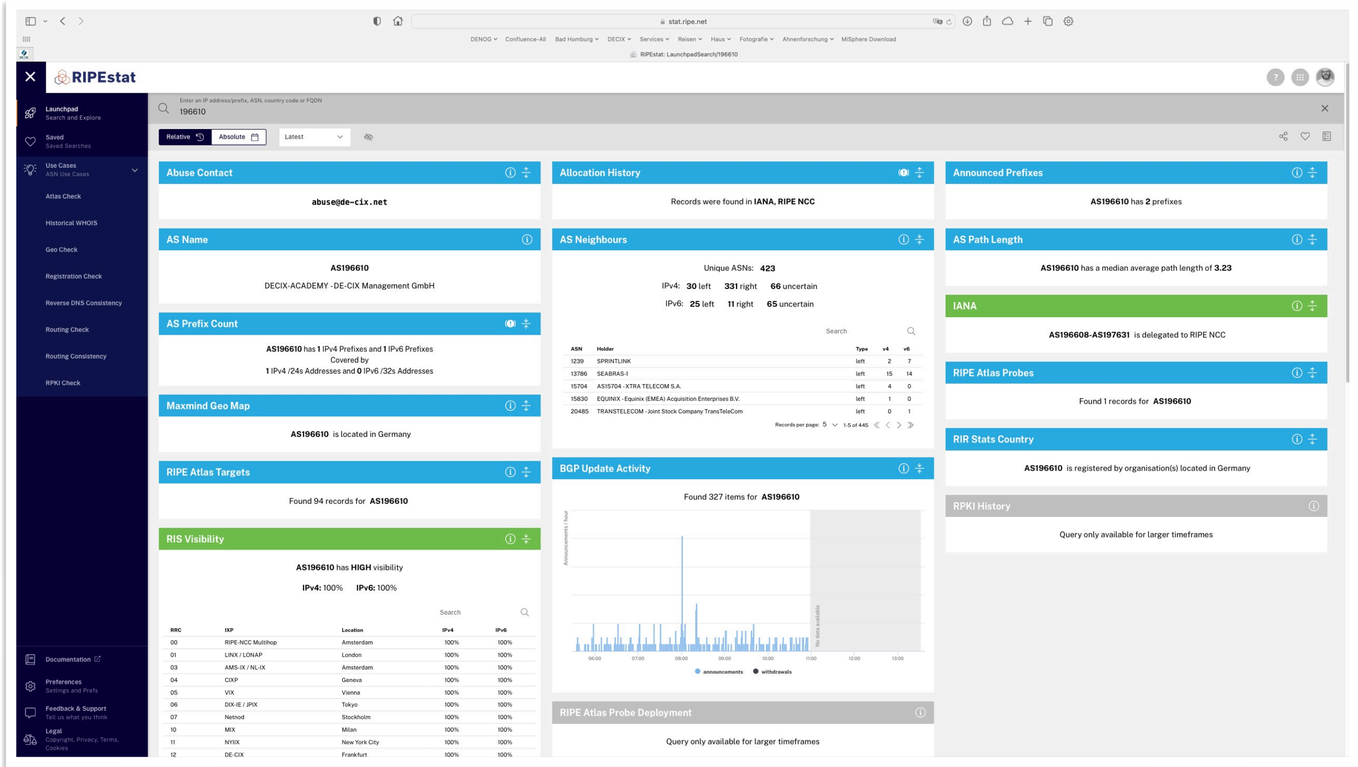
# Some resources for BGP



# Public tools for BGP

## RIPE Stat

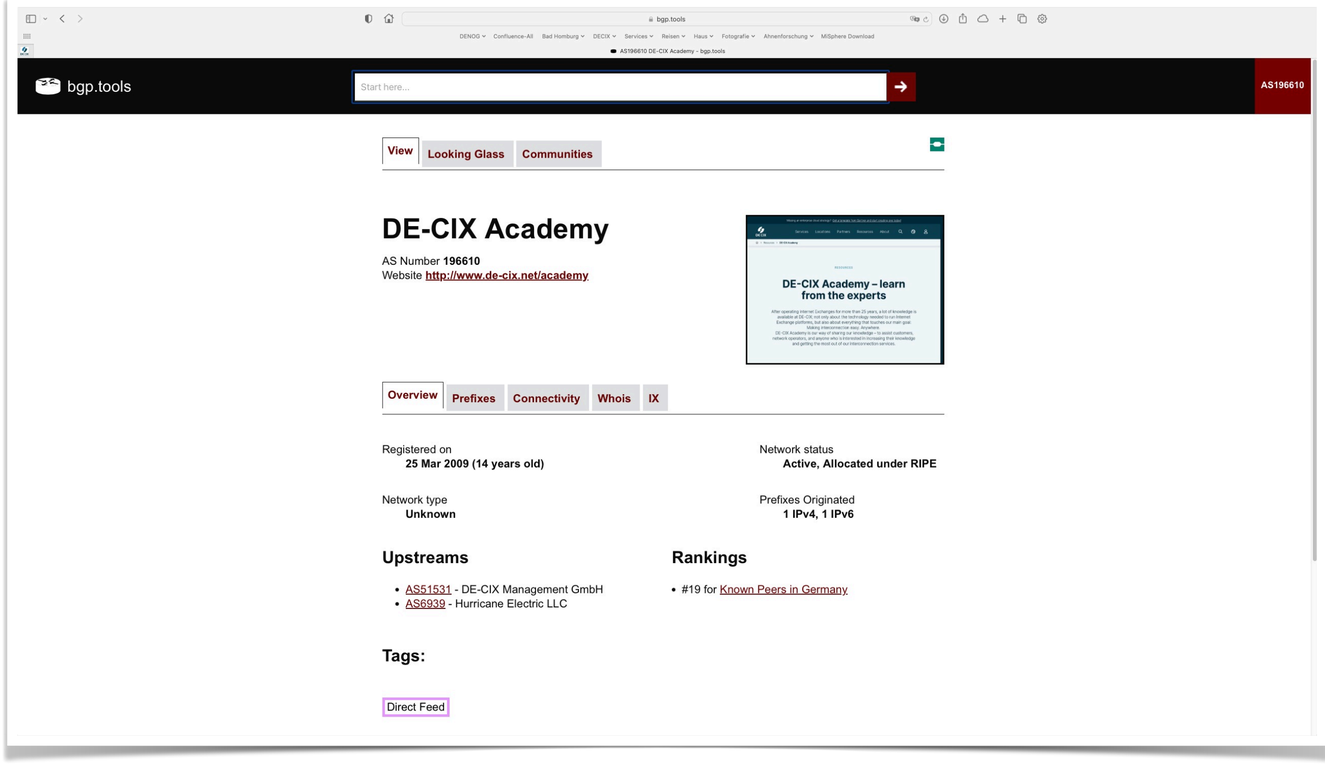
- Operated by the RIPE NCC (same entity handing out AS numbers in this region)
- Details about prefixes, ASes and more
- just check it out at https://stat.ripe.net



# Public tools for BGP

## bgp.tools

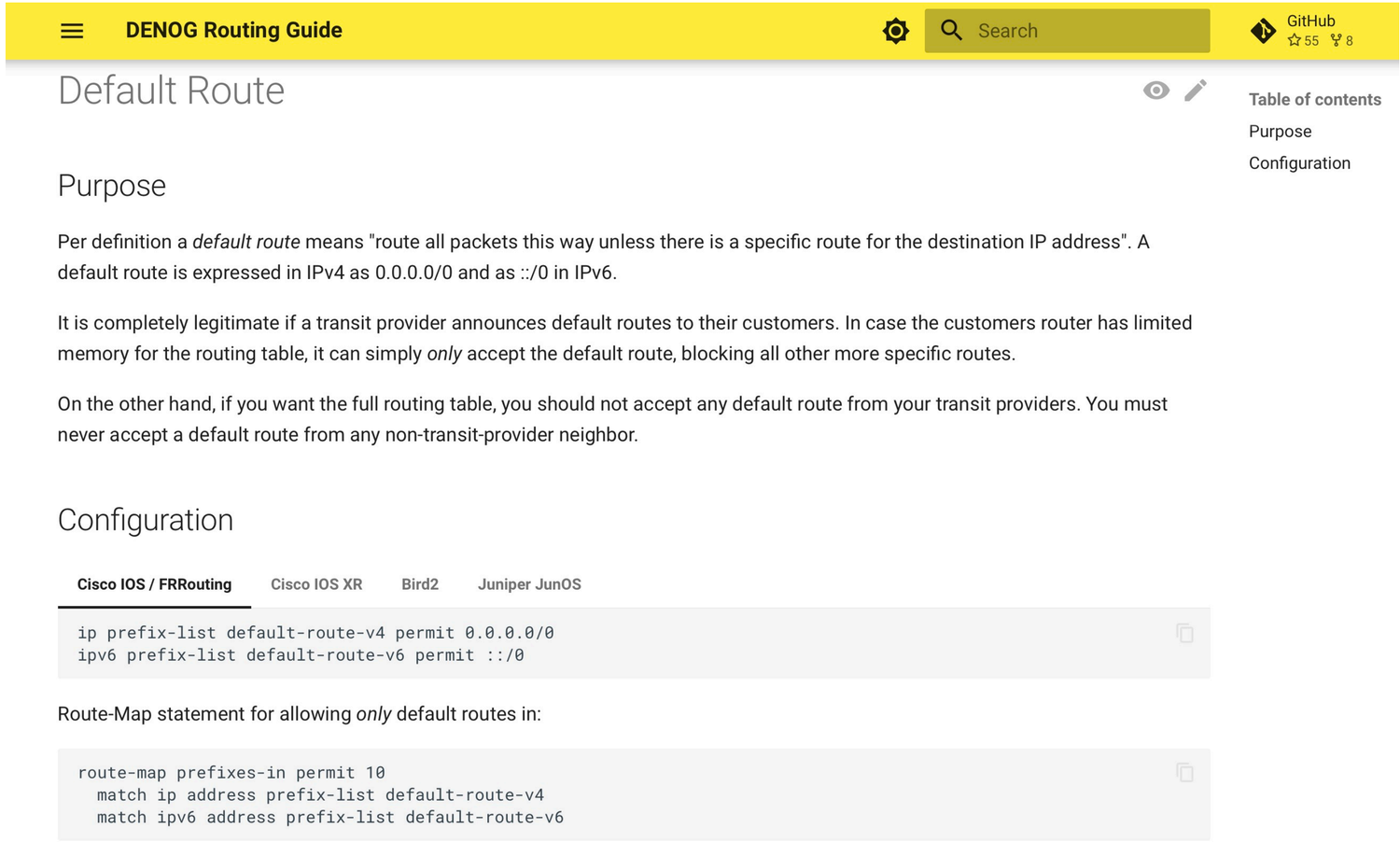
- Private initiative
- Free, offer premium monitoring service for a fee
- just check it out at https://bgp.tools



# DENOG Routing Guide

https://routing.denog.de

- Written by the Community
- Hosted and initiated by DENOG e.V.
- Talk to me or any other group member if you want to contribute



# Where to meet people doing BGP



# DENOG e.V.

**German Network Operators Group**




- Started as a chat-channel
- First in-person meeting in 2009
- Registered Association (e.V.) since 2017
- Yearly 3-day meeting
  - Next one ?
- Regular local meetups

Welcome!

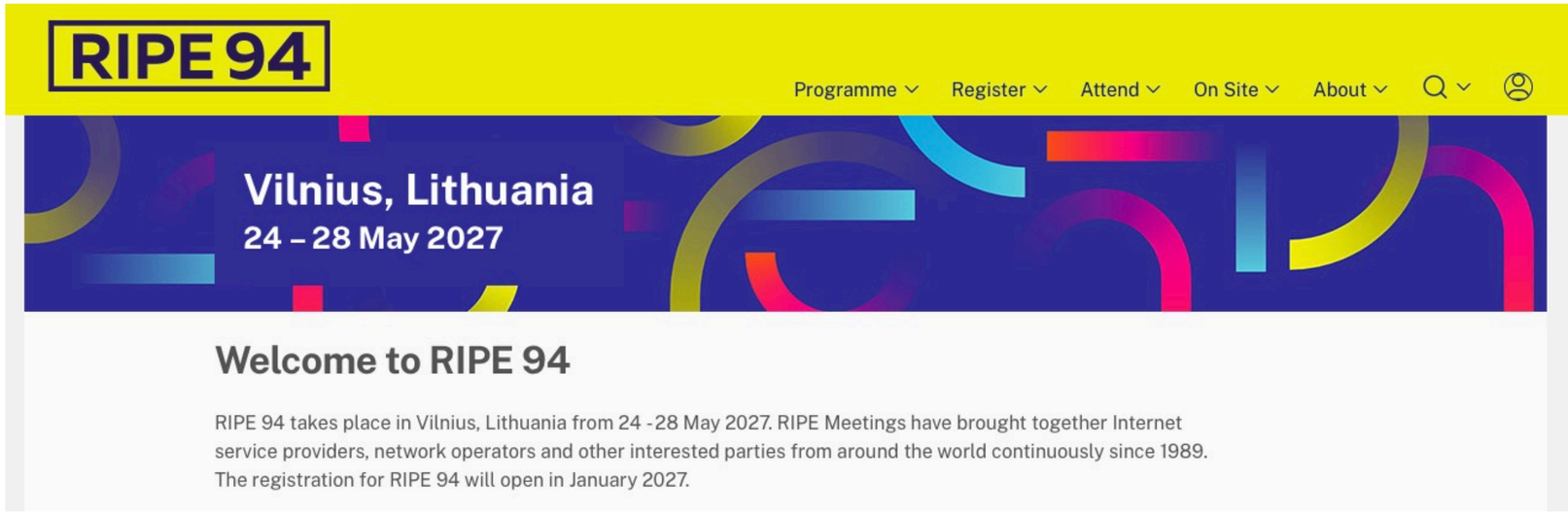
Website: https://denog.de

Also visit https://routing.denog.de/

# RIPE

## Réseaux IP Européens

- RIPE is the community
- RIPE NCC hands out IP addresses
- Two meetings per year
  - 5 day long
  - Different locations
- No travel budget?
  All meetings are live streamed!




- Next RIPE meetings:
  - **Vilnius, May 24th 2027**
  - Madrid, November 15th 2027

DE CIX

# Housekeeping

- End of workshop around 17:00
- Lunch break 12:30
- Coffeebreak
- Short bio breaks when needed
- Help yourself with water etc. any time

DE CIX

# *Prefixes and Netmasks*

## IPv4 and IPv6

**Wolfgang Tremmel**
wolfgang.tremmel@de-cix.net

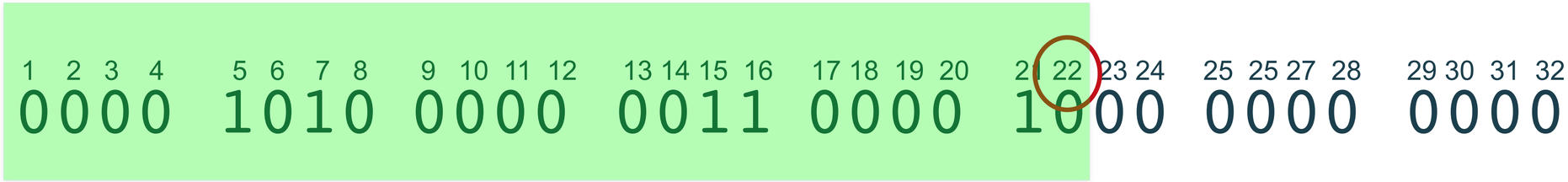


# What is BGP about?



# IPv4 Prefixes

10.3.8.0/22




- IPv4 and IPv6 addresses have a network and a host part
- A **prefix** is just the network part
- Important:
  - **The boundary between network and host can be anywhere!**
  - **Class-A, -B, -C : No such thing since 1993!**

# Characteristics of Prefixes: IPv4

10.3.8.0/22




Prefix-Length: 0-32

Notation:
- 4 Numbers 0-255
- Separated by "."
- a "/", followed by

Host-part all zero

32 Bits long

# Characteristics of Prefixes: IPv6

Prefix-Length: 0-128

2003:de:274f:400::/64




Notation:

- 4 digit hex numbers (0-9,a-f)
- Separated by ":"
- "::" = fill up with zeros

Host-part all zero

128 Bits long

# What is an Autonomous System?

And why do I need one?



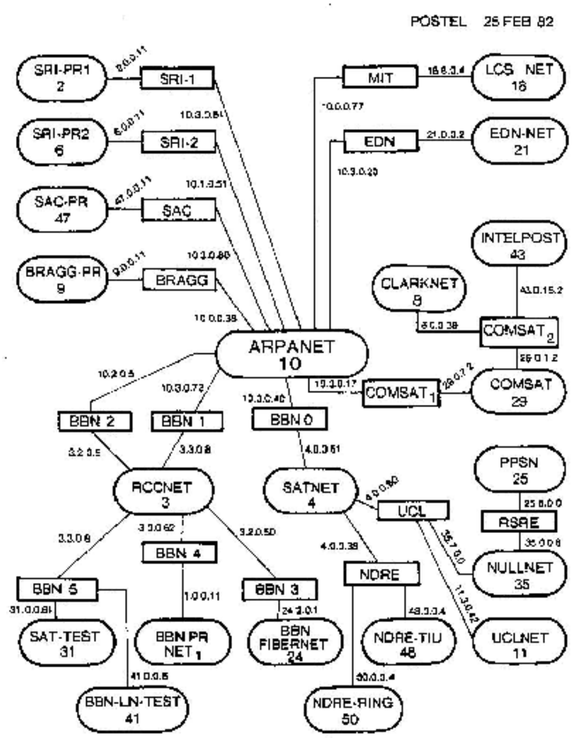
Wolfgang Tremmel
academy@de-cix.net

# A brief history of the Internet

*According to the Internet Hall of Fame*

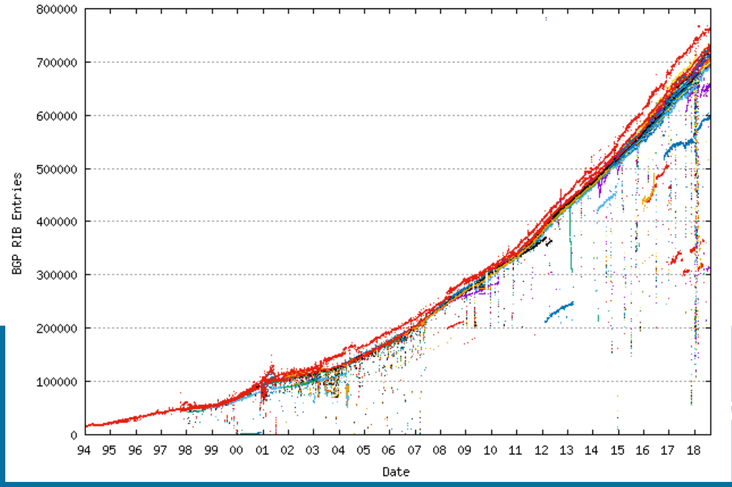
- 1982 – Arpanet (successor of Internet)
- 1982: RFC827 defines Exterior Gateway Protocol:

**"Autonomous systems will be assigned 16-bit identification numbers (in much the same ways as network and protocol numbers are now assigned)"**



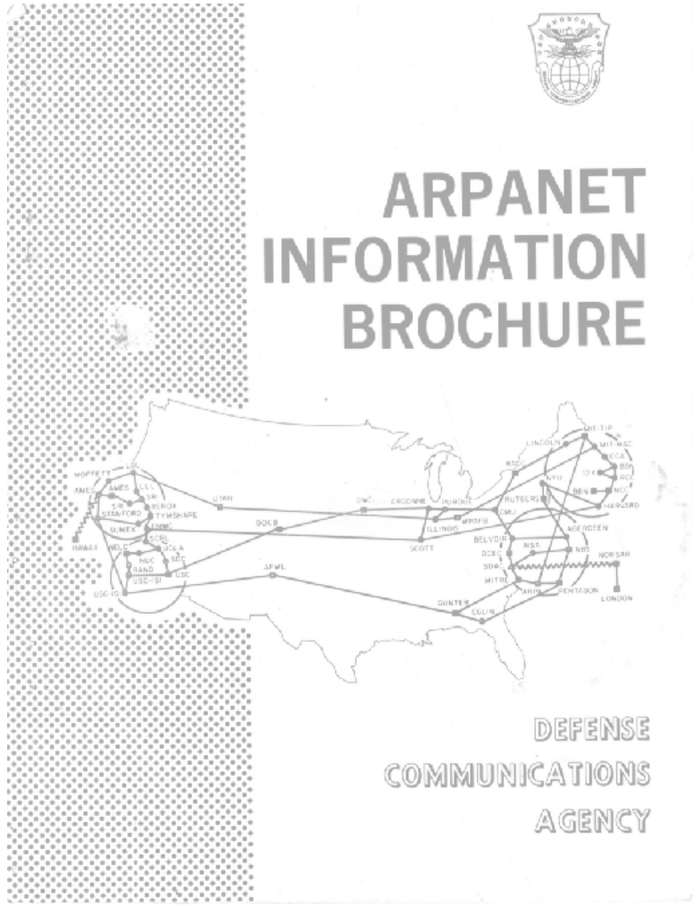
# *Some years later...*

- ➔ June 2025: There are 78667 active ASs
  (source: http://bgp.he.net/report/prefixes#_networks)
- ➔ In 2001, planning to extend AS numbers to 4 bytes began
- ➔ This was finalized as a RFC (standard) in May 2007 (see RFC 6793)
- ➔ Today, 4-byte AS number are assigned
  - ➔ They are supported by all modern routers
  - ➔ You can no longer request a 2-byte AS number
  - ➔ There is also no reason for requesting one



# But what is an Autonomous System?

The classic definition of an Autonomous System is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASes.




➔1996 – Defined in RFC1930 (earlier definitions exist)

➔Other routing protocols call these "areas" for example

➔The inside of an AS is "invisible" to the outside

# *So this is an Autonomous System!*

"An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."

- **So now you know:**
  - You do not need a router
  - However, you need prefixes to be routed
- **Most commonly:**
  - you do have a router
  - … or more than one
  - and it "belongs" to an AS if it runs BGP

# What is an Autonomous System good for?

| | If you have an AS | Without an AS |
|---|---|---|
| **Redundancy** | You can have multiple upstream ISPs and Peering | You only can have one upstream ISP |
| **Control** | You have full control over your outgoing traffic | Your upstream ISP controls your traffic |
| **Cost** | You can optimize your traffic for cost | You just pay your upstream ISP |
| **Peering** | You can setup your own peering policy and have full control | Your upstream ISP makes all decisions |

# *Sounds good, I want an AS, where can I get one?*

- ➔AS numbers are globally unique
- ➔So some sort of authority must exist for handing them out
- ➔This authority is IANA - the Internet Assigned Numbers Authority
- ➔But no, you cannot go to IANA and just ask for an AS – they delegated the task
- ➔to five Regional Internet Registries (RIRs)
- ➔For Europe, the RIPE NCC in Amsterdam hands out AS Numbers



# *Getting an AS number from RIPE NCC: The easy way*

- →Just become a customer
  - →You have to be a legal entity
  - →Fill out the forms
  - →Pay the sign-up fee (and annual fee)
- →Request your AS number
  - →You have to be/want to be multi-homed (peering counts!)
  - →AS numbers from RIPE NCC cost €50 / ASN / year
  - →RIPE Academy offers lots of online / offline trainings to help you get started.

# *Getting an AS number without becoming a RIPE NCC member*

- ➔You can also get an AS from someone who already is a RIPE NCC member
- ➔This is called a "sponsoring LIR"
- ➔Basically they request the AS from RIPE NCC for you
- ➔… and may charge you for this (cost from RIPE NCC is €50 / ASN / year)

# *Now I have an AS – how can I route my IP prefix?*

- ➔Hmm, this depends where you have your IP space from
- ➔In general, IPv4 prefixes of /24 or larger are announcable via BGP
- ➔In IPv6 you can announce /48 or larger
- ➔If you have just become a new RIPE NCC member, you can also request IP space
  - ➔~~As there is not much IPv4 left, you get a /22 once (and not more)~~
  - ➔**IPv4 is out! No more IPv4 addresses (except by transfers)**
    - ➔Yes - new RIPE NCC members can still request one /24 via the waiting list
  - ➔But plenty of IPv6 available...
- ➔To check whether your current space is routable from your new AS, the best way is to check with whom you got that IP space from

in 2026: ~€9000 for a /24 IPv4

# Open Questions?

# *BGP - an introduction*

## BGP for networks who peer: Part 1

**Wolfgang Tremmel**
wolfgang.tremmel@de-cix.net



# BGP

# *BGP*

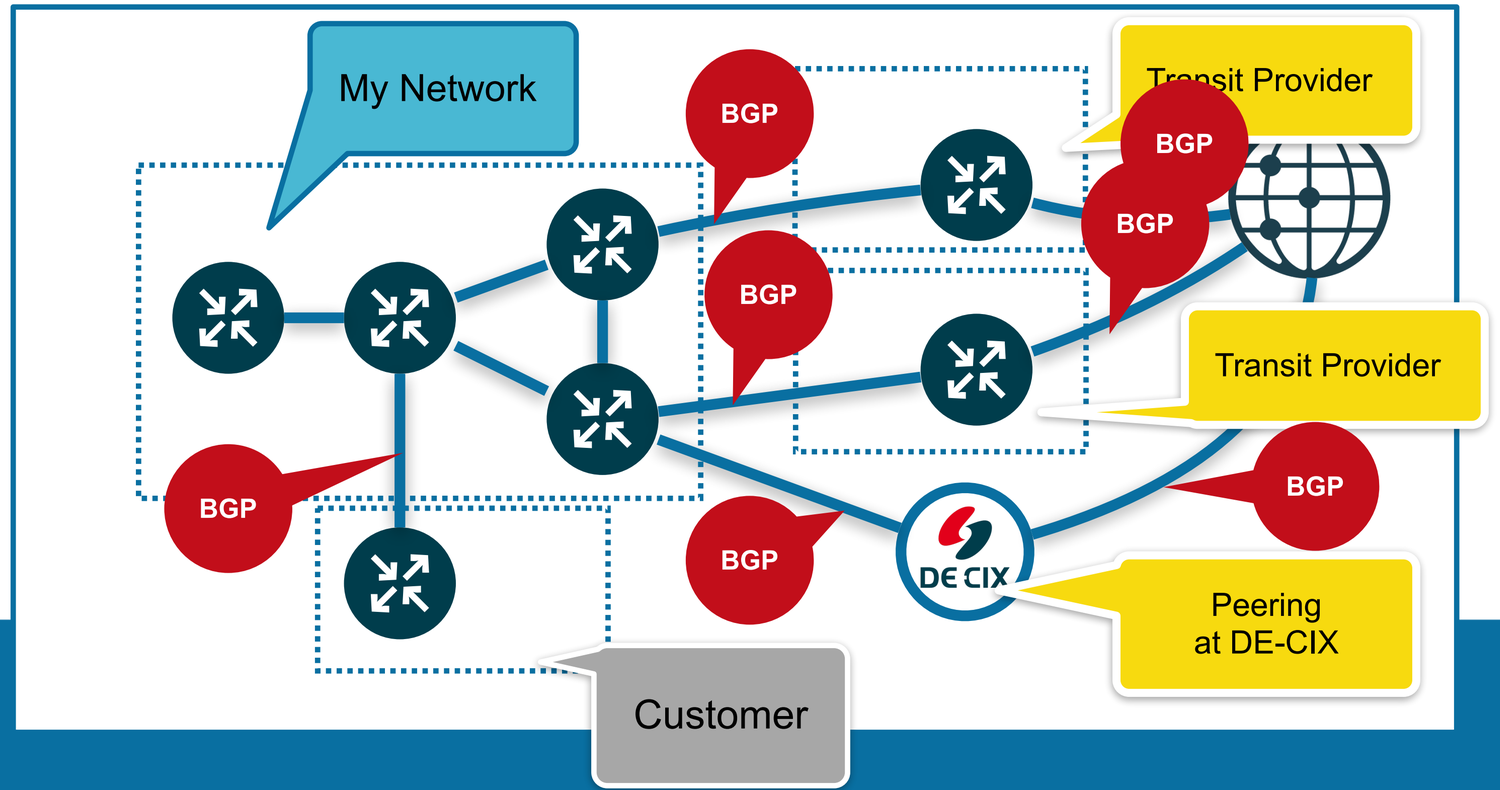
Border

Gateway

Protocol

# BGP

- → P - a **PROTOCOL**
  - spoken between Internet routers
- → B - spoken on the **BORDER** between two providers
- → G - on the **GATEWAYs** - the routers connecting two providers



# *BGP Key Concepts*

- →IPv4 and IPv6 prefixes
- →Autonomous Systems (AS)
- →BGP - the protocol itself
- →running BGP in your network
- →using BGP to connect to Peers and Transit Providers

# *BGP Training Overview*

- →01 - Prefixes and AS numbers
- →02 - BGP Introduction
- →03a - Setting up iBGP
- →03b - Setting up eBGP
- →04 - Becoming multi-homed
- →05 - BGP Best Path Selection

---

- →06 - BGP Communities
- →07 - BGP Traffic Engineeri
- →08 - BGP Security

Basic Part (today)

Advanced Part

If time permits

# How does BGP work?



# BGP is a protocol to announce prefixes

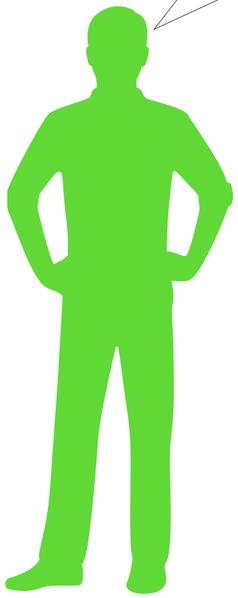
## Everybody has Neighbors

I am **AS196610**, DE-CIX Academy, and I announce prefix **2a02:c50:db8::/48**

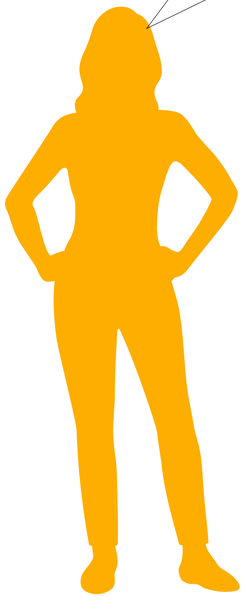


DE-CIX Academy
AS196610

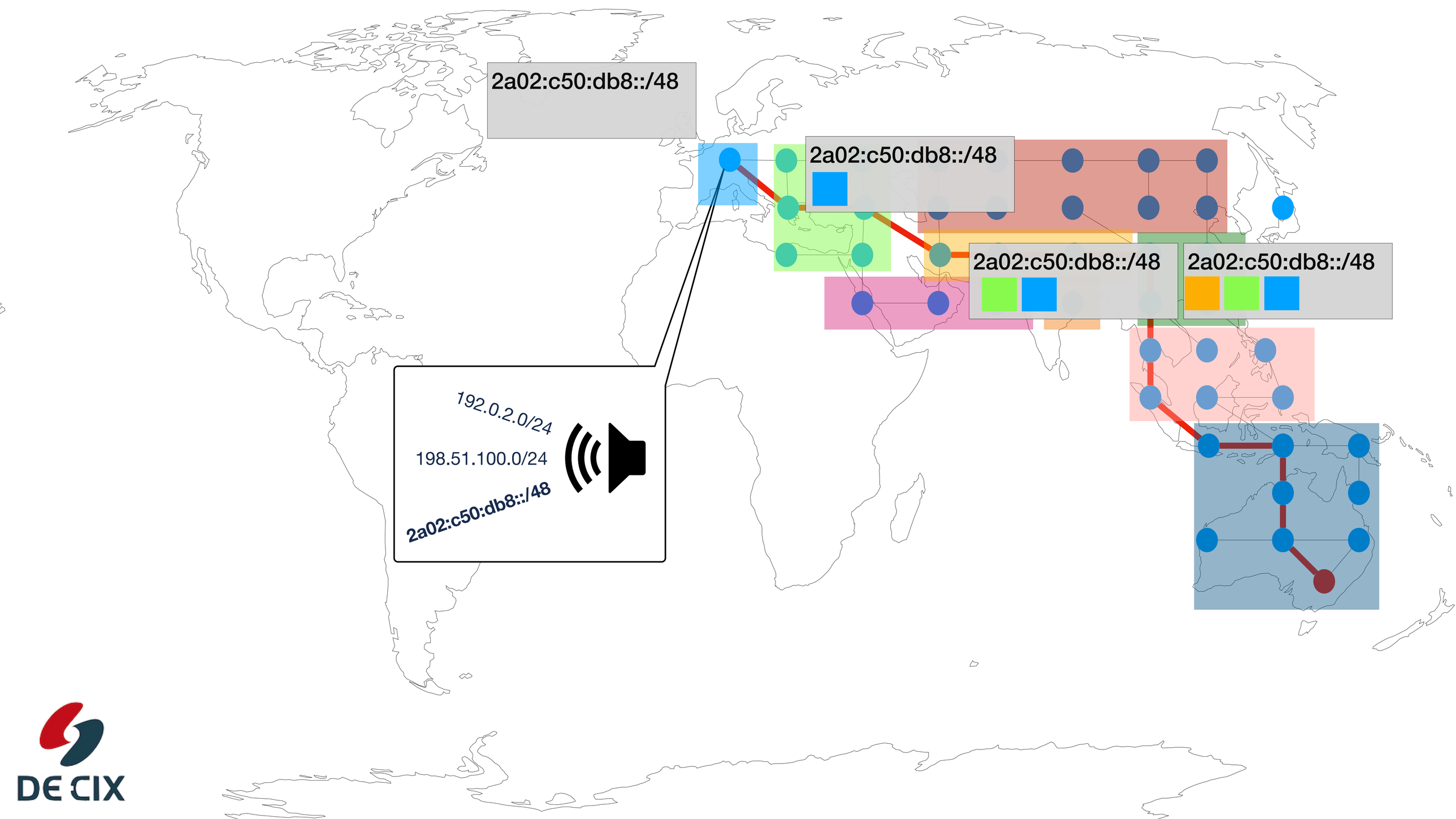
My neighbor AS196610 announces prefix 2a02:c50:db8::/48



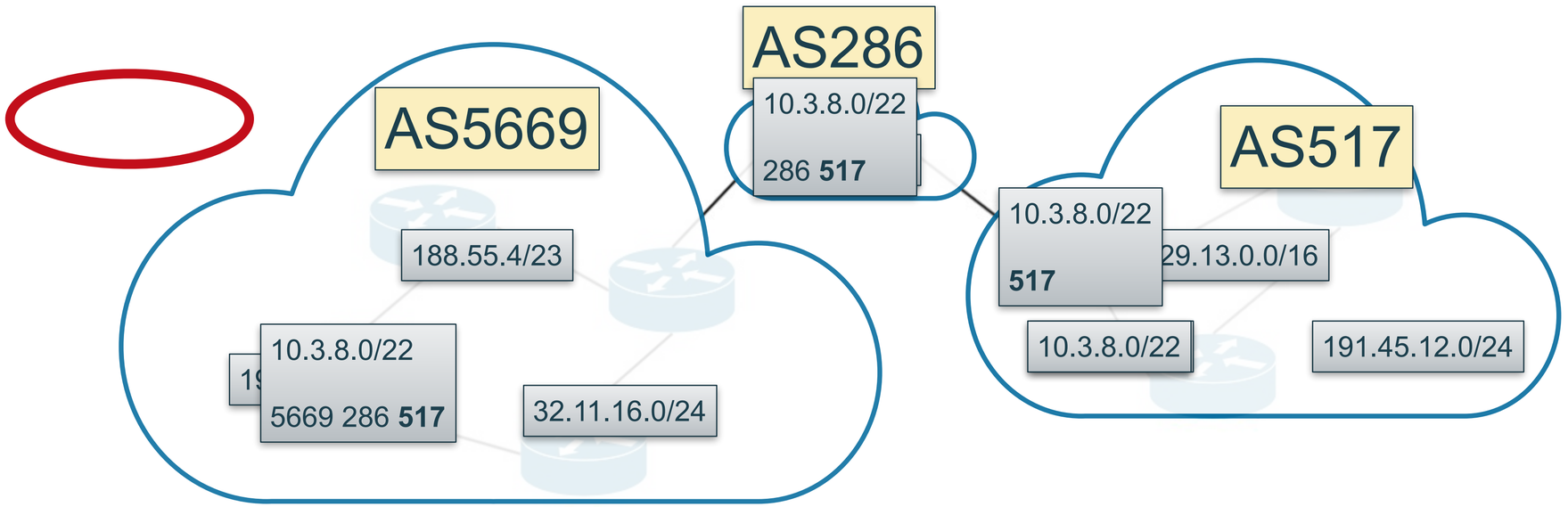
My green neighbor told me, his neighbor AS196610 announces prefix 2a02:c50:db8::/48



DE-CIX


2a02:c50:db8::/48
2a02:c50:db8::/48
2a02:c50:db8::/48
2a02:c50:db8::/48
192.0.2.0/24
198.51.100.0/24
2a02:c50:db8::/48

# BGP - Key Concepts: The AS Path



# A real live example



```
asd2-rs-02>show bgp ipv4 unicast 129.13.0.0
Load for five secs: 1%/0%; one minute: 4%; five minutes: 5%
Time source is NTP, 09:14:07.268 UTC Thu Aug 17 2017
BGP routing table entry for 129.13.0.0/16, version 2944571
Paths: (13 available, best #10, table default)
....
  5669 286 517
    134.222.85.126 from 134.222.85.126 (134.222.85.126)
      Origin IGP, metric 0, localpref 80, valid, internal
      Community: 286:18 286:19 286:28 286:29 286:49 286:800 286:888
```

Prefix

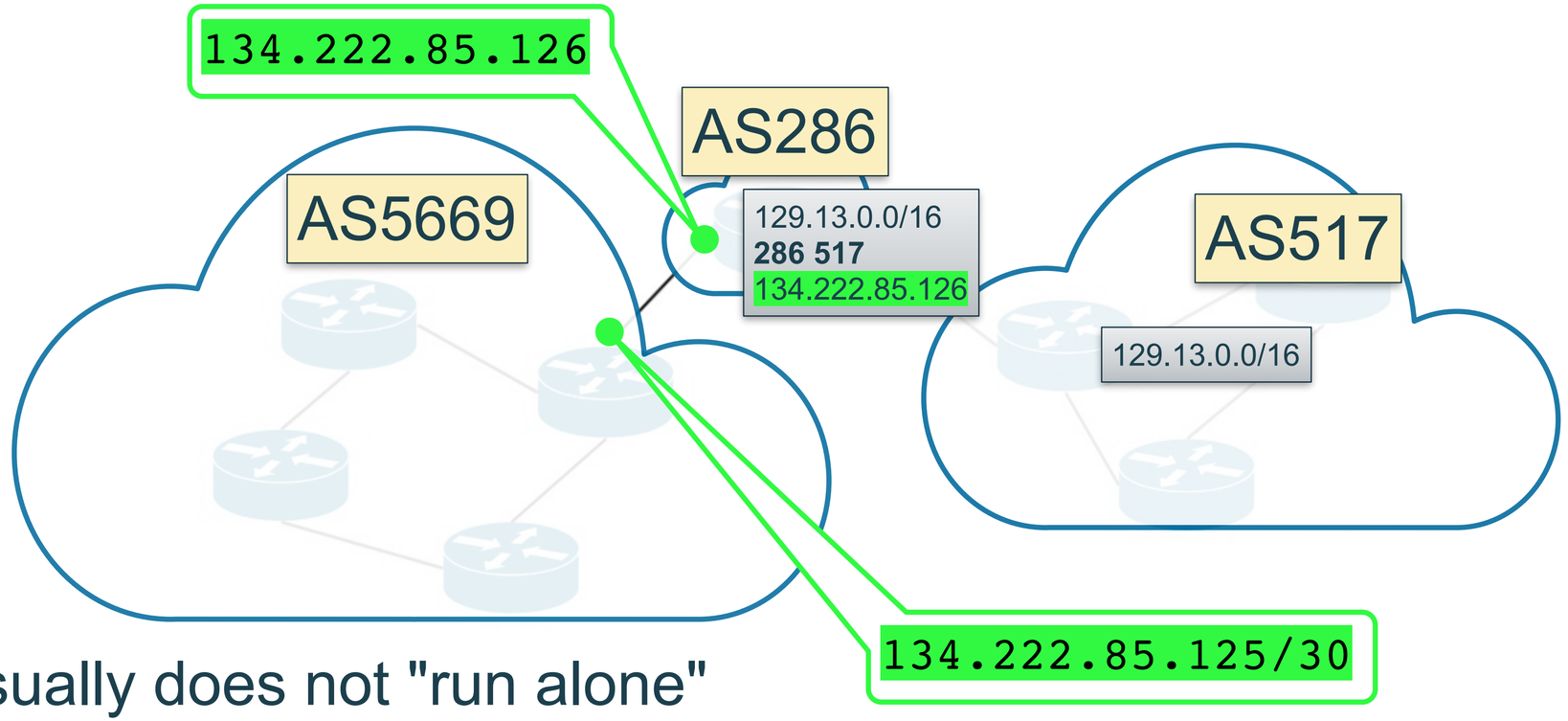
AS-Path

Next Hop IP

Originator AS

DE CIX

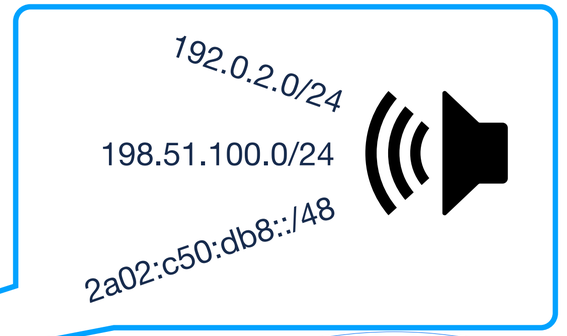
# BGP - Key Concepts: Next Hop Address




BGP usually does not "run alone"
Another routing protocol is needed to distribute interface addresses

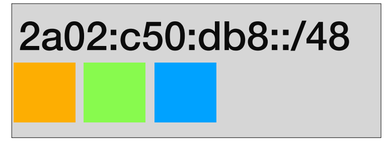
# BGP announces prefixes

## To neighbors




- BGP **announces** IP prefixes to **neighbors**
  - These neighbors have to be **configured**
  - Each BGP speaking device is part of an **Autonomous System**
  - The path these announcements take is recorded - this is called the **Autonomous System Path**
  - The AS Path shows which Autonomous Systems have forwarded the prefix announcement
  - The rightmost AS in the AS Path is called the "**Originator**"

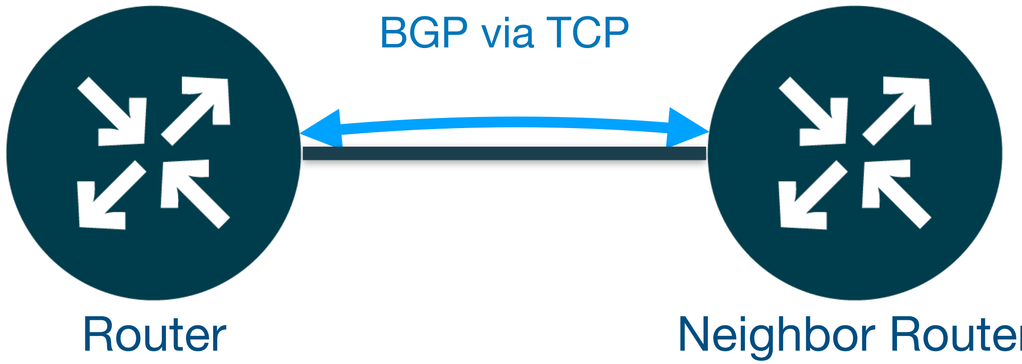
I am **AS196610**, DE-CIX Academy, and I announce prefix 2a02:c50:db8::/48




DE CIX

# BGP Neighbors

## Directly connected neighbors

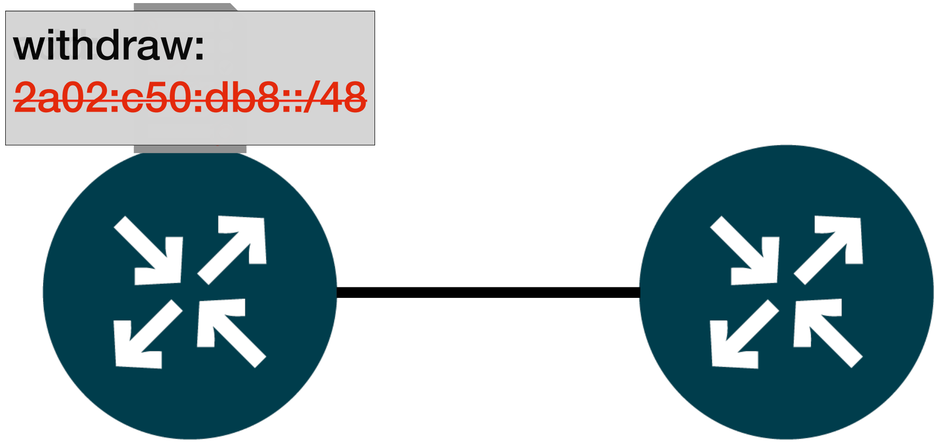



- BGP uses **TCP** to connect to a neighbor
- TCP brings already:
  - **Reliable transport** (sender knows that receiver got it)
  - **Flow control** (do not send faster than the receiver can receive)
  - **Framing** (putting BGP messages into packets)

DE CIX

# BGP works incremental

## Using add- / withdraw- messages

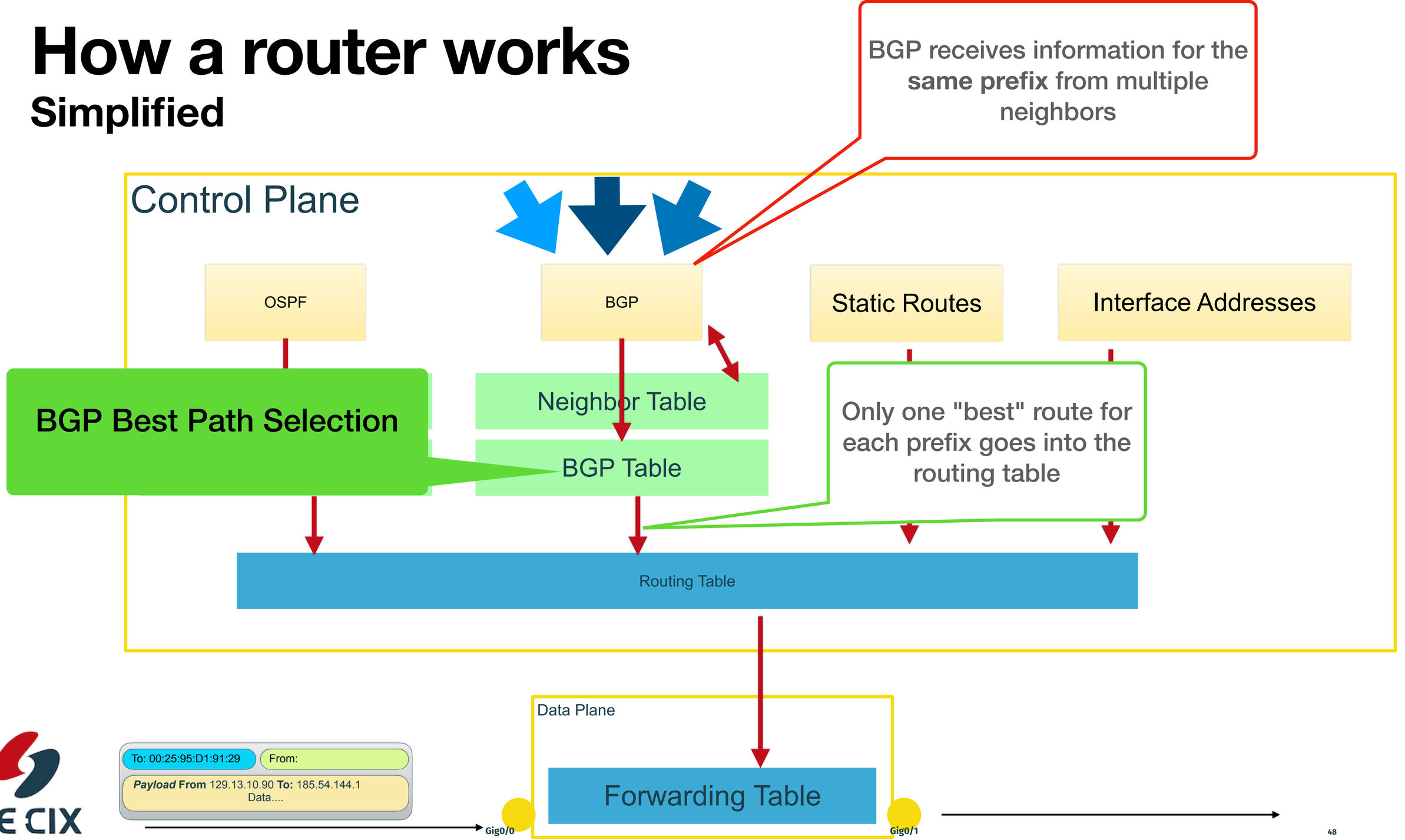



- At session setup, BGP announces "everything" to its neighbor
- After that, updates are **incremental**:
  - If BGP learns about a new prefix, it sends an **add**-message to neighbors
  - If a prefix goes away, it sends a **withdraw** message to neighbors
- As long as the BGP session is "up", a router assumes its neighbors are "in sync" (= did not forget anything it sent)

DE CIX

# How a router works

## Simplified



# Open Questions?

# *BGP for networks who peer*

## A lab for BGP

**Wolfgang Tremmel**
academy@de-cix.net



# First idea: Use RaspberryPI

## It can run BGP, really!

- It is reasonably priced
- Open Source
- Flexible
- There are too many things that can go wrong
- I would spend too much time on debugging.

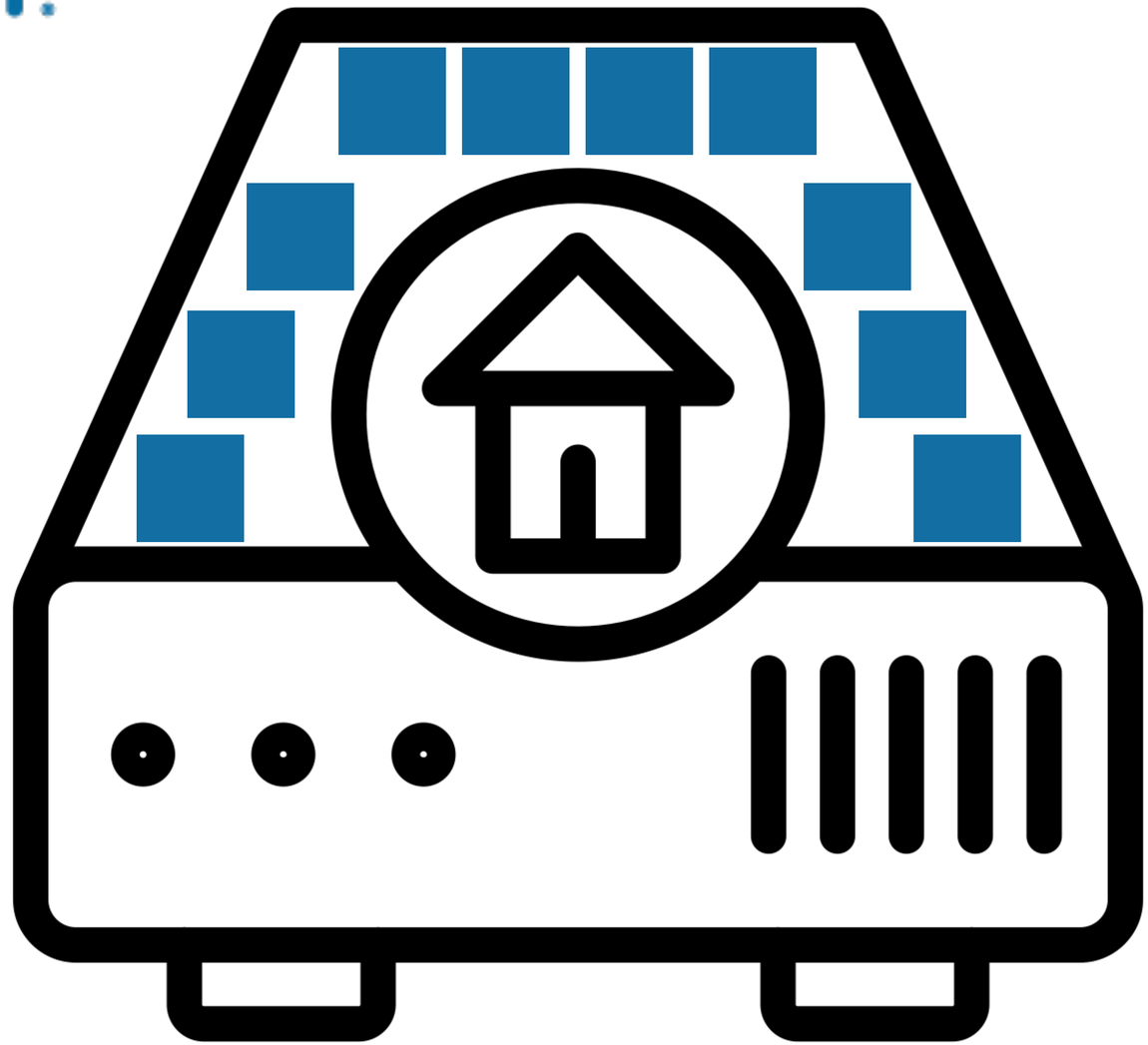



DE CIX

# Idea:

**Docker, Docker-Compose and FRR**

- Generate a *Docker* container for each participant
  - Accessible via web browser
  - Running OpenSource router FRRouting
- With Docker scaling is easy
  - Scale using a custom script



# FRRouting

## Open Source Router



- Open Source routing daemon
  - based on Quagga
- Actively developed
- "Cisco-IOS-like" user interface
- Not only BGP, but a lot of other routing protocols as well
- See frrouting.org

## Docker Container

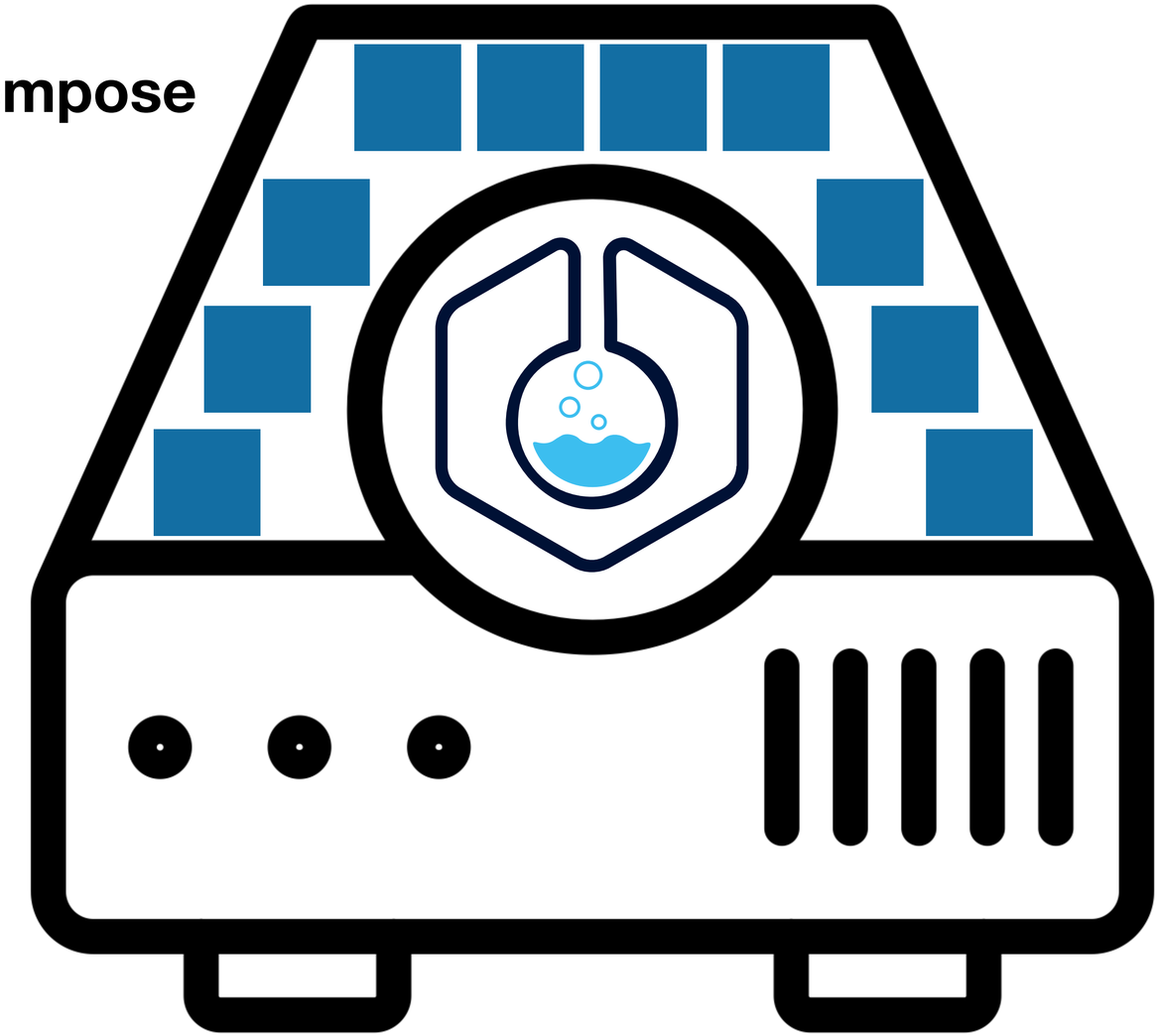
- Alpine Linux
- Supervisord
- TTYd
- FRRouting Software

DE CIX

# New: Containerlab

## Use Containerlab instead of Docker-Compose

- Similar setup
- Script builds containerlab file
- Docker runs containers
- But now using containerlab instead of docker-compose
- Old lab still available
- All Open-Source!



https://gitlab.com/de-cix-public/team-academy/bgp/BGPLab

DE CIX

# Setting up the DE-CIX Academy lab

## Some Linux experience required...







- The lab runs on any Debian-like Linux host or virtual machine
  - Only modest resources required (~100kB RAM / lab seat)
  - Other Linux distributions possible, but not tested
- Setup via Ansible



- Presents lab seats on ports 9001, 9002, ...
- If HTTPS is needed, just add a proxy





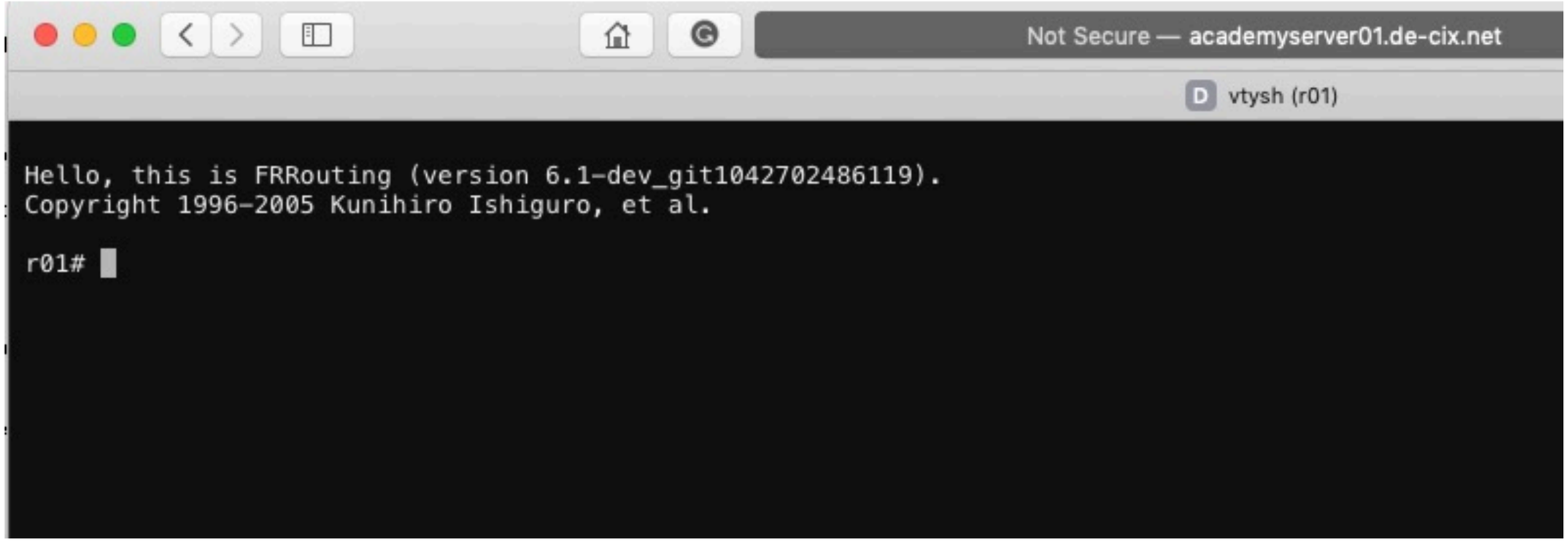

# *Connect now*

➔Your router:

➔using a Browser:

**https://containerlab.as196610.net:70XX**

➔:7002, 7003, ...


Not Secure — academyserver01.de-cix.net
vtysh (r01)
Hello, this is FRRouting (version 6.1-dev_git1042702486119).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
r01#

# *Experiment: Connecting to your router*



Open file: experiment 00

# Open Questions?

# Welcome to "Introduction to BGP" Part 2

## DENOG18 Essen

- Please find a seat
- **Download** the training material:
  - https://go.de-cix.net/bgpzip
- We continue at 14:00
- We finish at 17:00 +/-
- There will be a short bio-/coffeebreak

DE CIX

# *BGP for networks who peer*

## 01 - IGP and iBGP



**Wolfgang Tremmel**
academy@de-cix.net

DE CIX

*Where networks meet*

www.de-cix.net

# *BGP - not re-inventing the wheel*

→BGP uses TCP for transport

→so no need to re-implement features TCP already provides, like

→reliable transport

→flow control

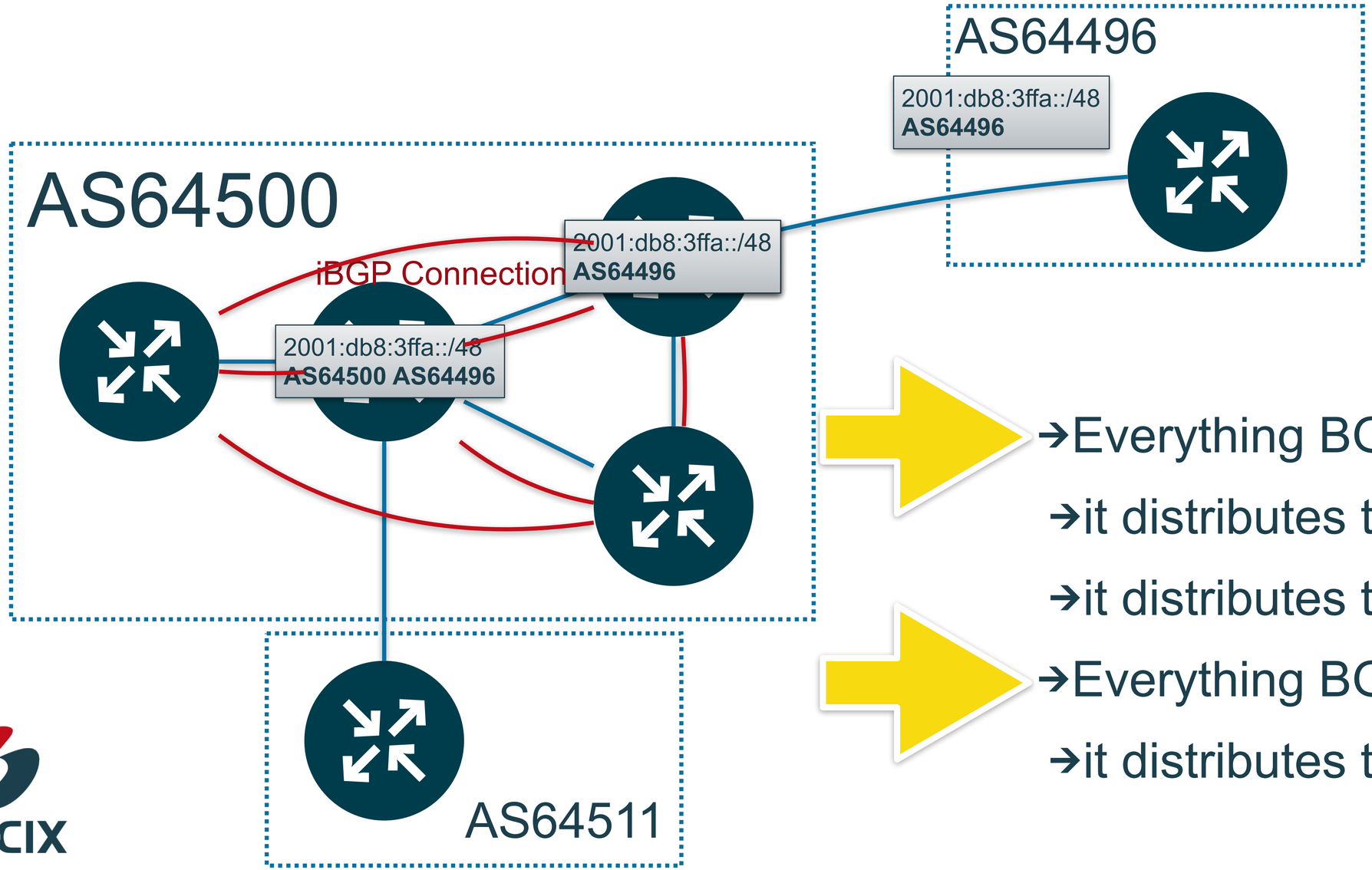
→framing

→as long as the TCP session is up, BGP assumes its neighbors are up

→and have all the information sent to them

# *BGP - (re)distributing prefixes*

- a BGP speaking router
  - learns prefixes
  - distributes prefixes to its BGP neighbors
- Everything BGP learns from extern
  - it distributes to intern
  - it distributes to extern
- Everything BGP learns from intern
  - it distributes to extern



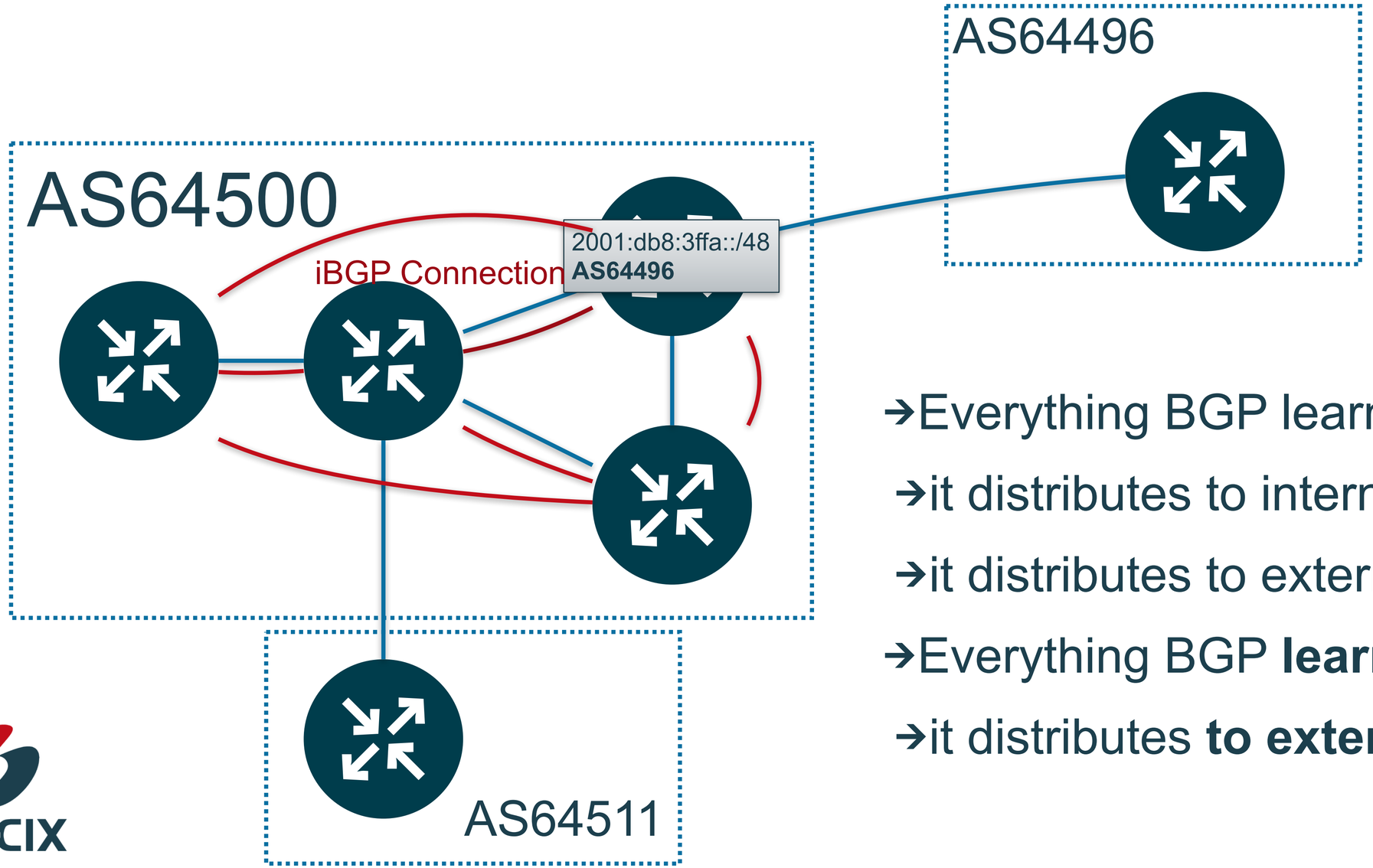

→Everything BGP learns from extern

→it distributes to intern

→it distributes to extern

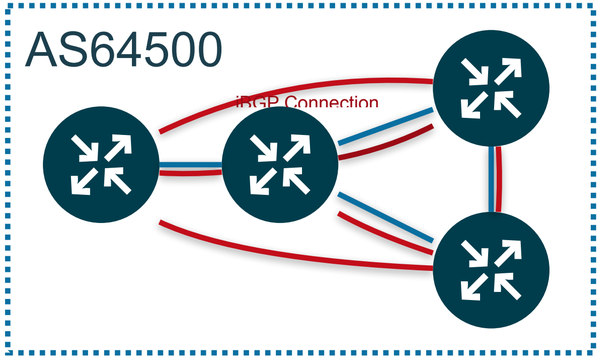
→Everything BGP learns from intern

→it distributes to extern




- Everything BGP learns from extern
  - it distributes to intern
  - it distributes to extern
- Everything BGP **learns from intern**
  - it distributes **to extern**

# *Which routers do need to speak BGP?*




- ➔Thats a hard question
- ➔Depends on your network design
- ➔In general:
  - ➔all routers with an **outside connection** to another AS
  - ➔big core routers inside your network
    - ➔but they might not need the full routing table
    - ➔a default-route to the nearest exit might do
    - ➔you can also distribute a default route via (i)BGP

# *BGP - not re-inventing the wheel*

→BGP uses **TCP for transport**

→so no need to re-implement features TCP already provides, like

→reliable transport

→flow control

→framing

→iBGP needs for session setup (via TCP)

→a **source IP** address

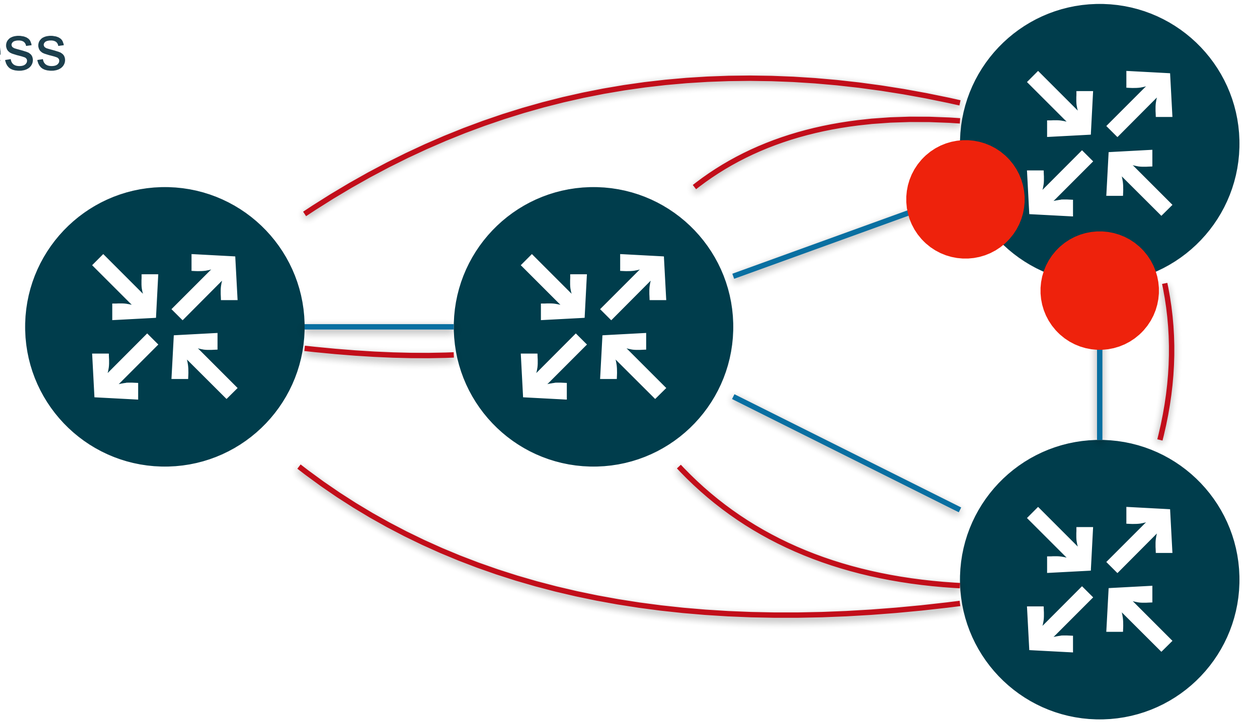
→a **destination IP** address

# *iBGP Session Setup - addressing*

➔iBGP needs for session setup (via TCP)

➔a **source IP** address

➔a **destination IP** address

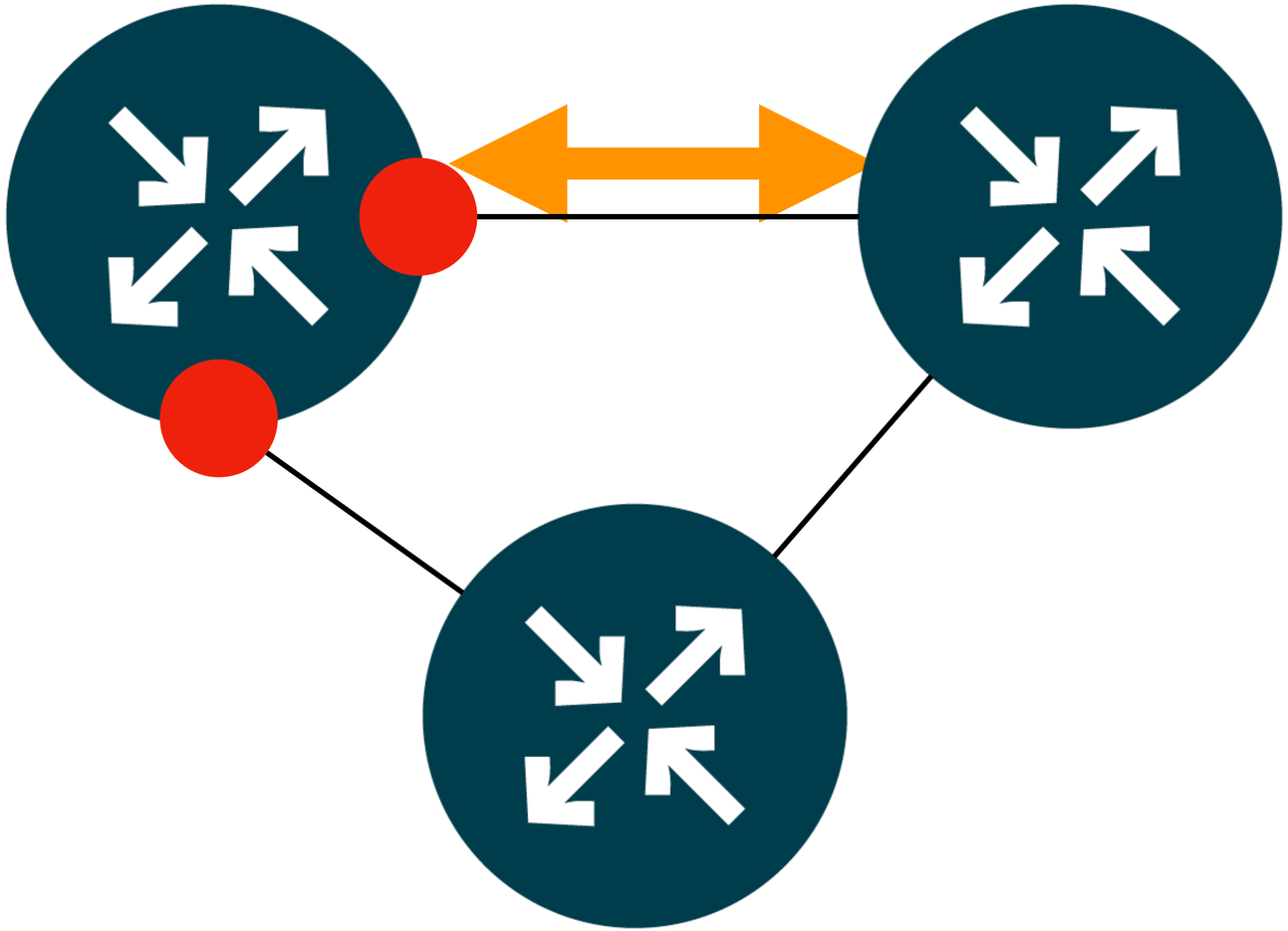


➔**Source IP**: Which one?

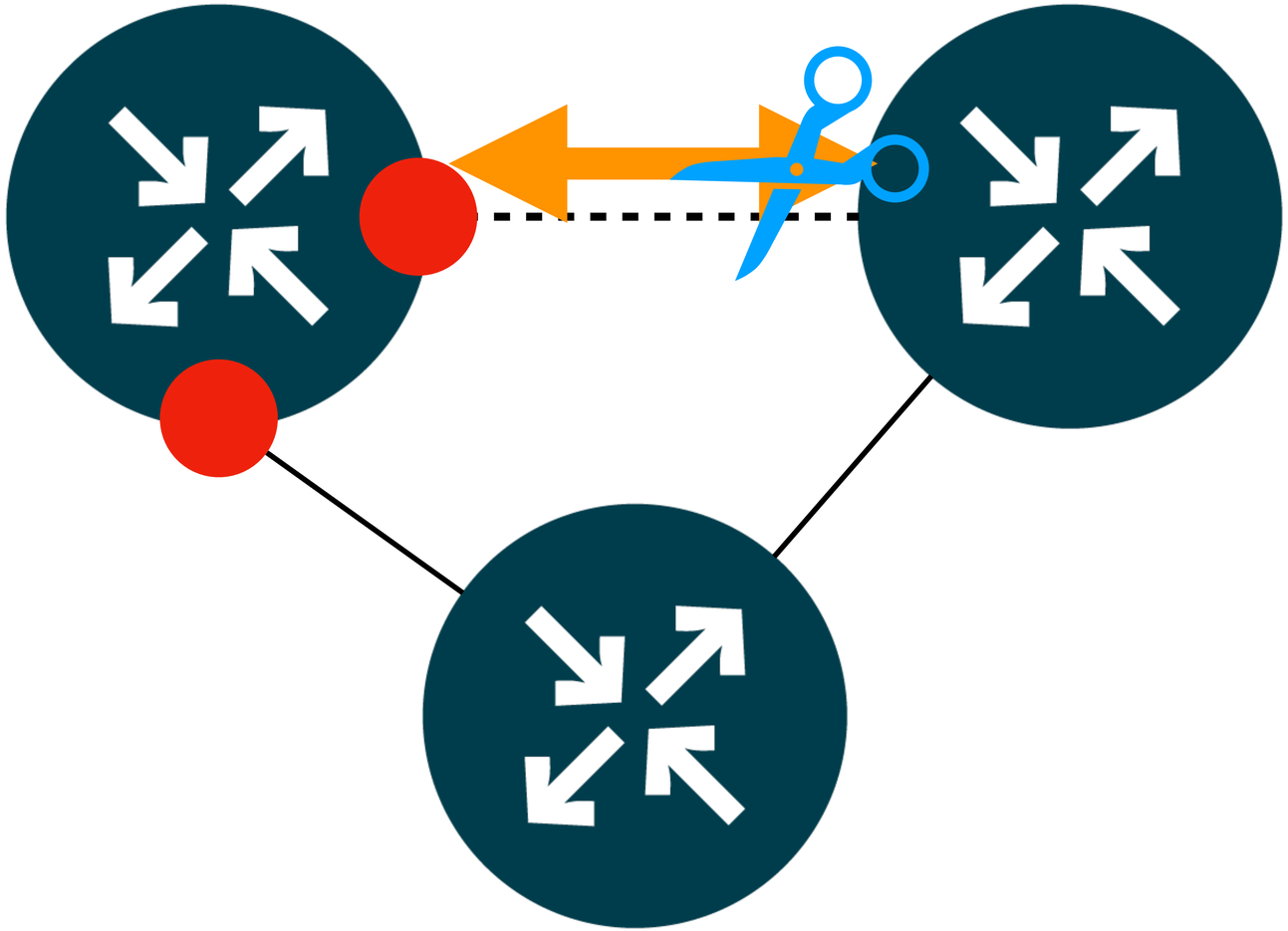
➔ Default:

➔address of the interface on which the **packets leave the router**

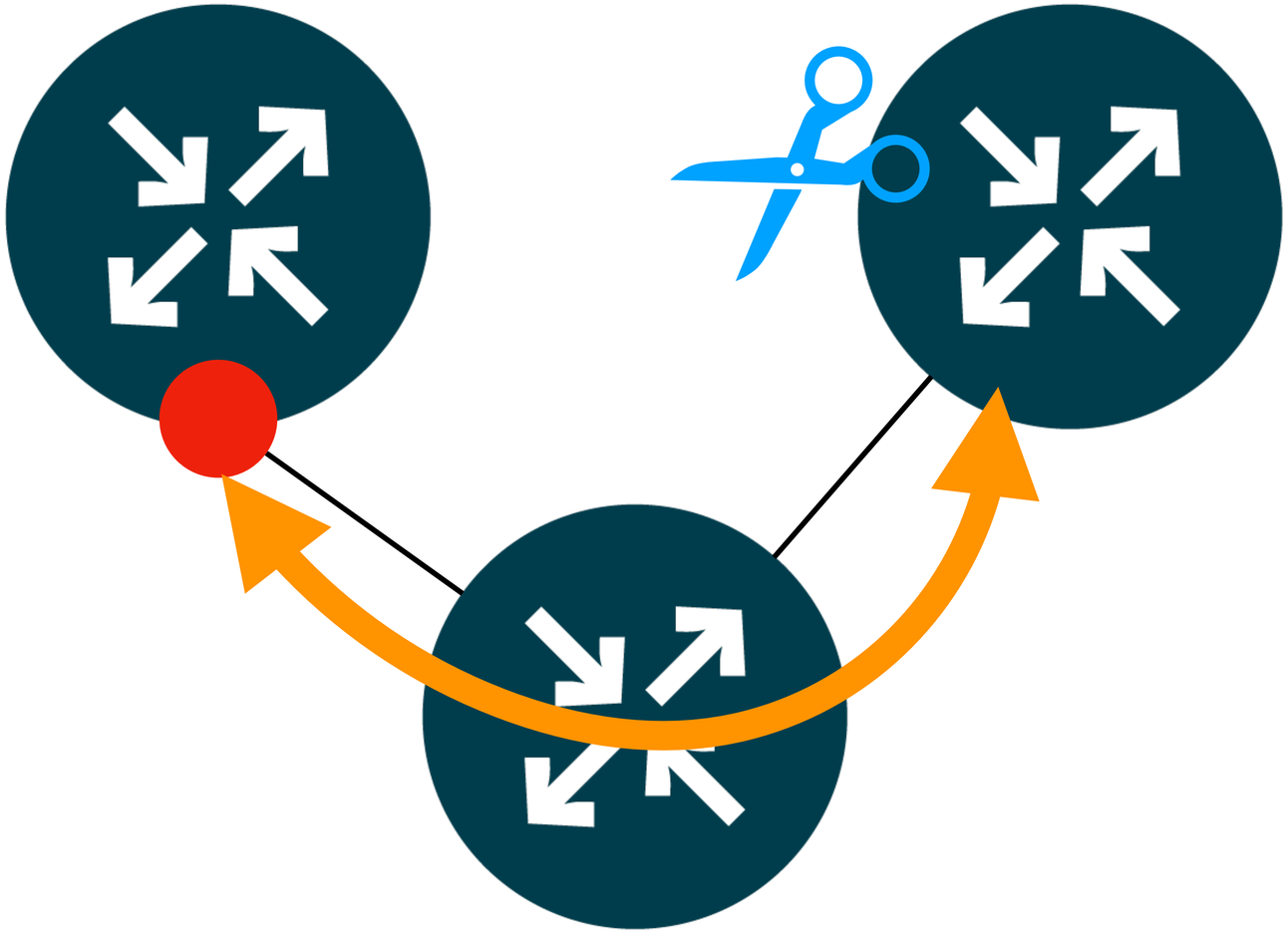
# iBGP - Which IP addresses to use?



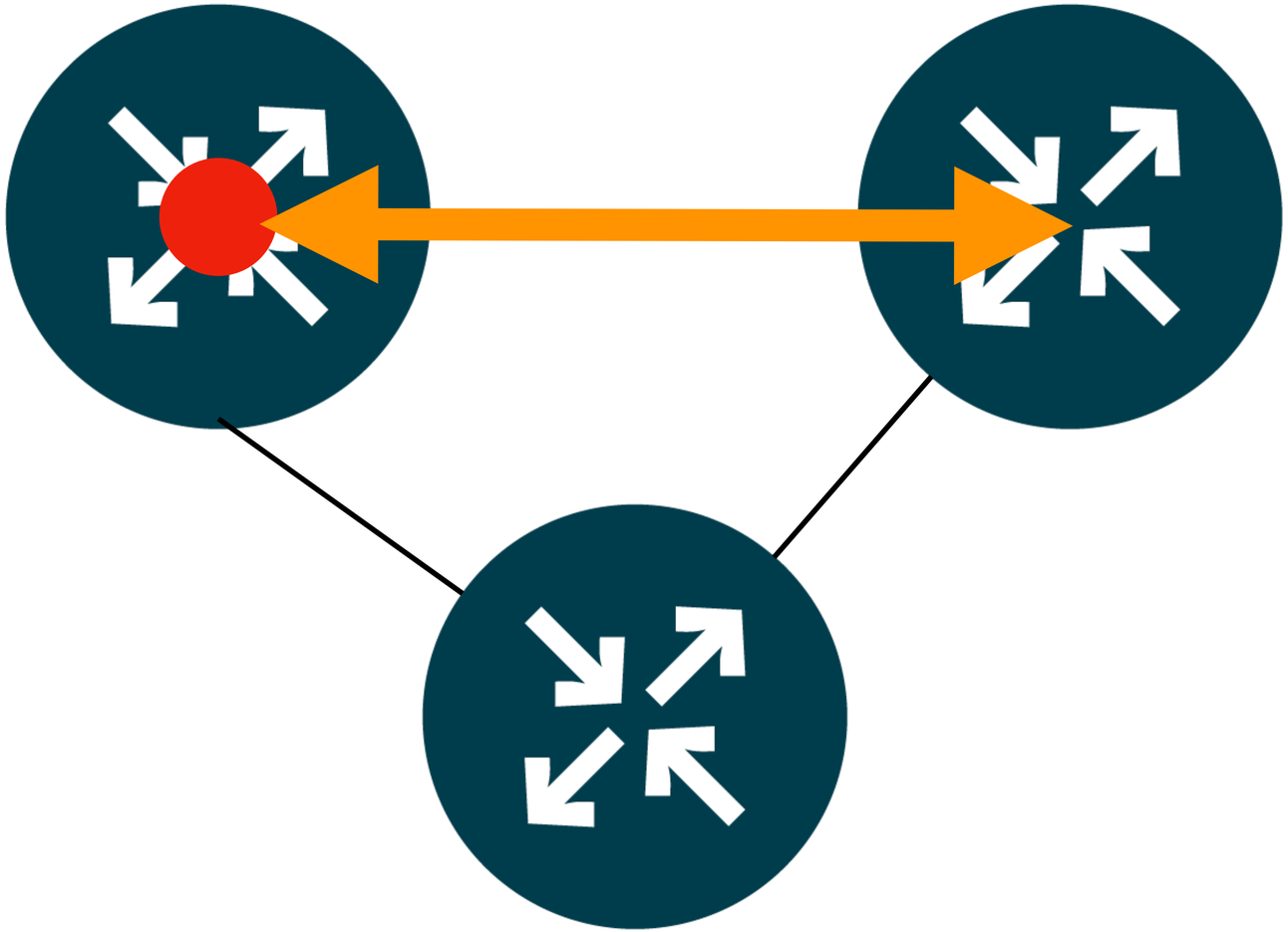
# iBGP - Which IP addresses to use?



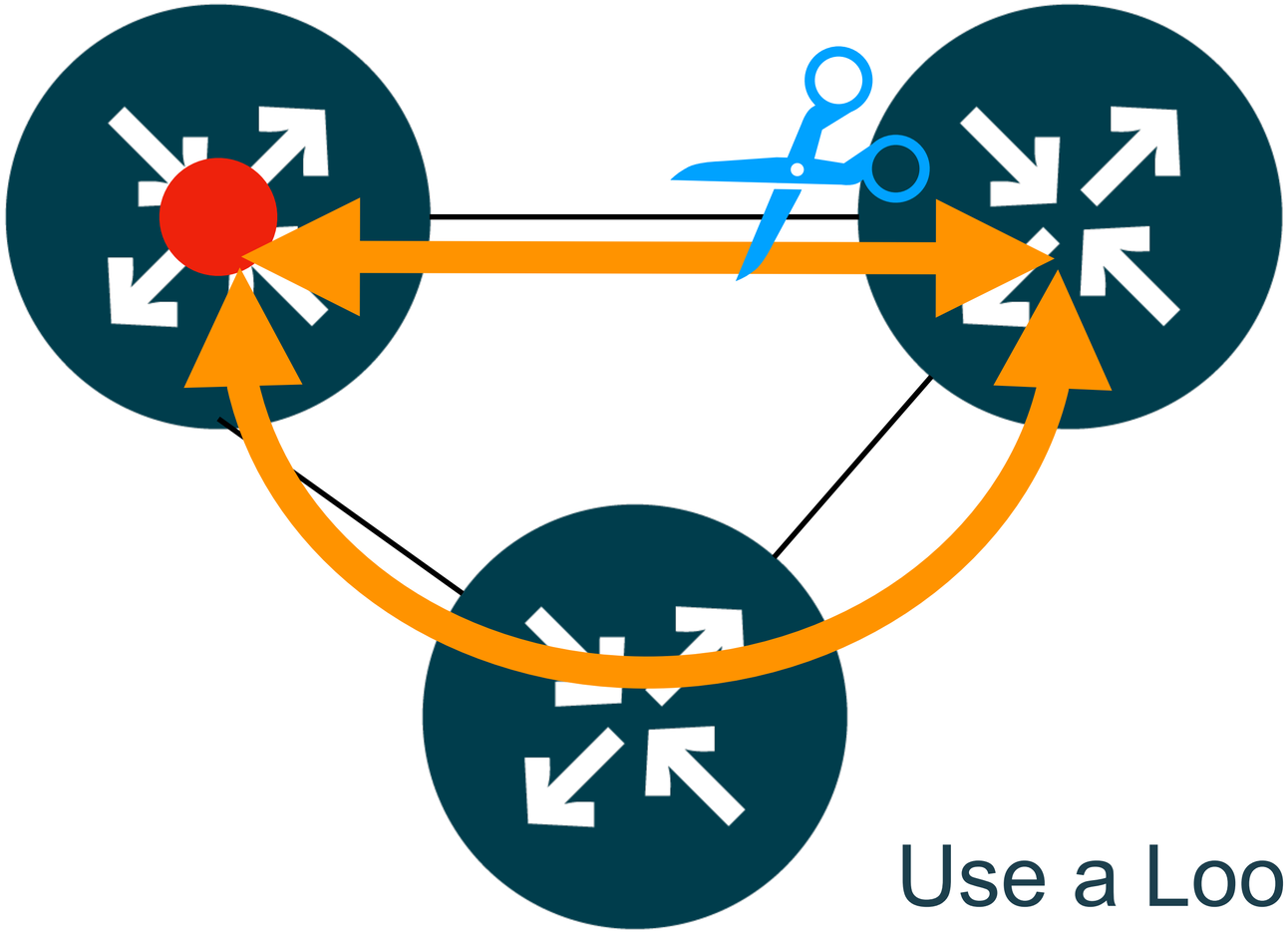
# iBGP - Which IP addresses to use?



# iBGP - Which IP addresses to use?

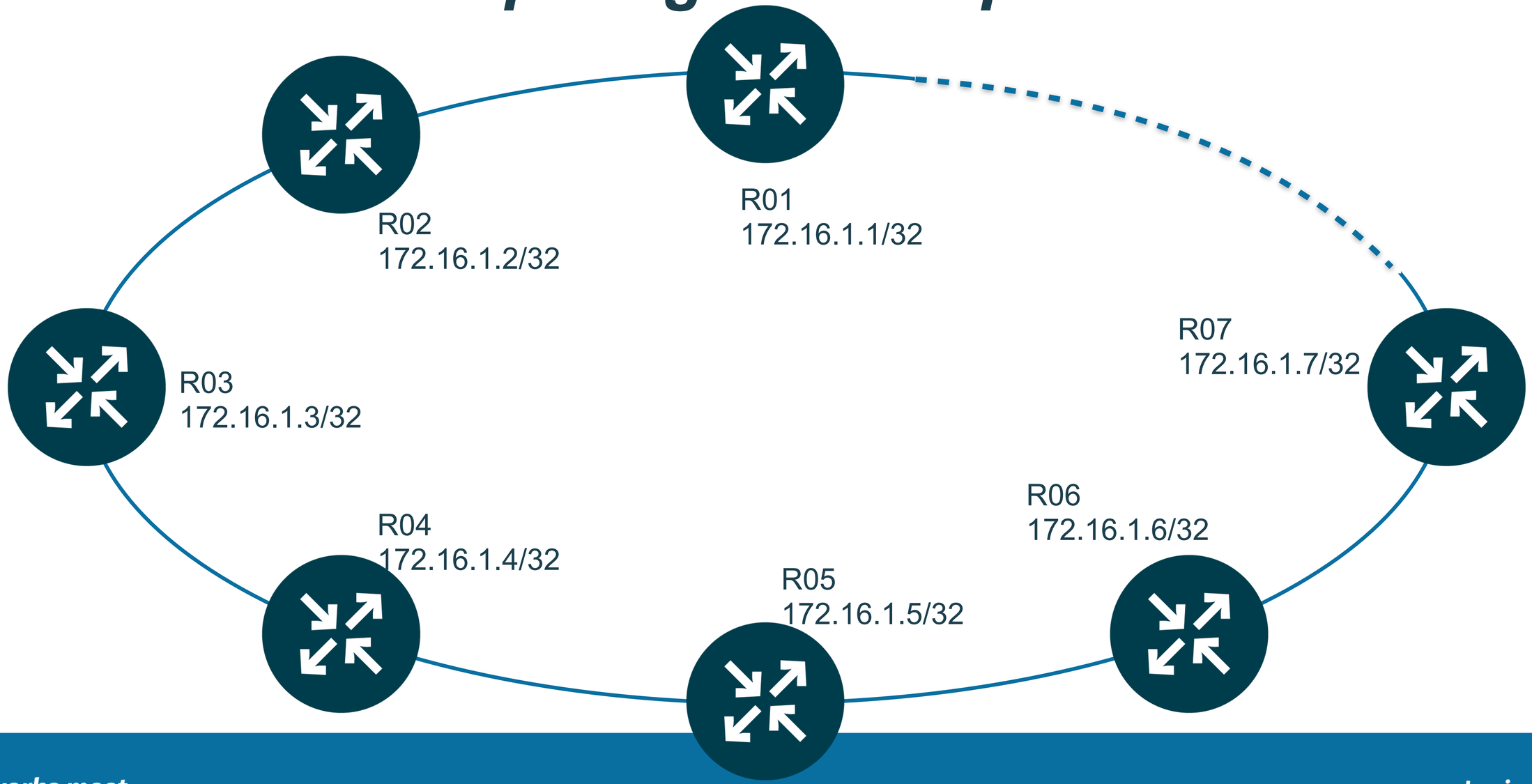


# iBGP - Which IP addresses to use?

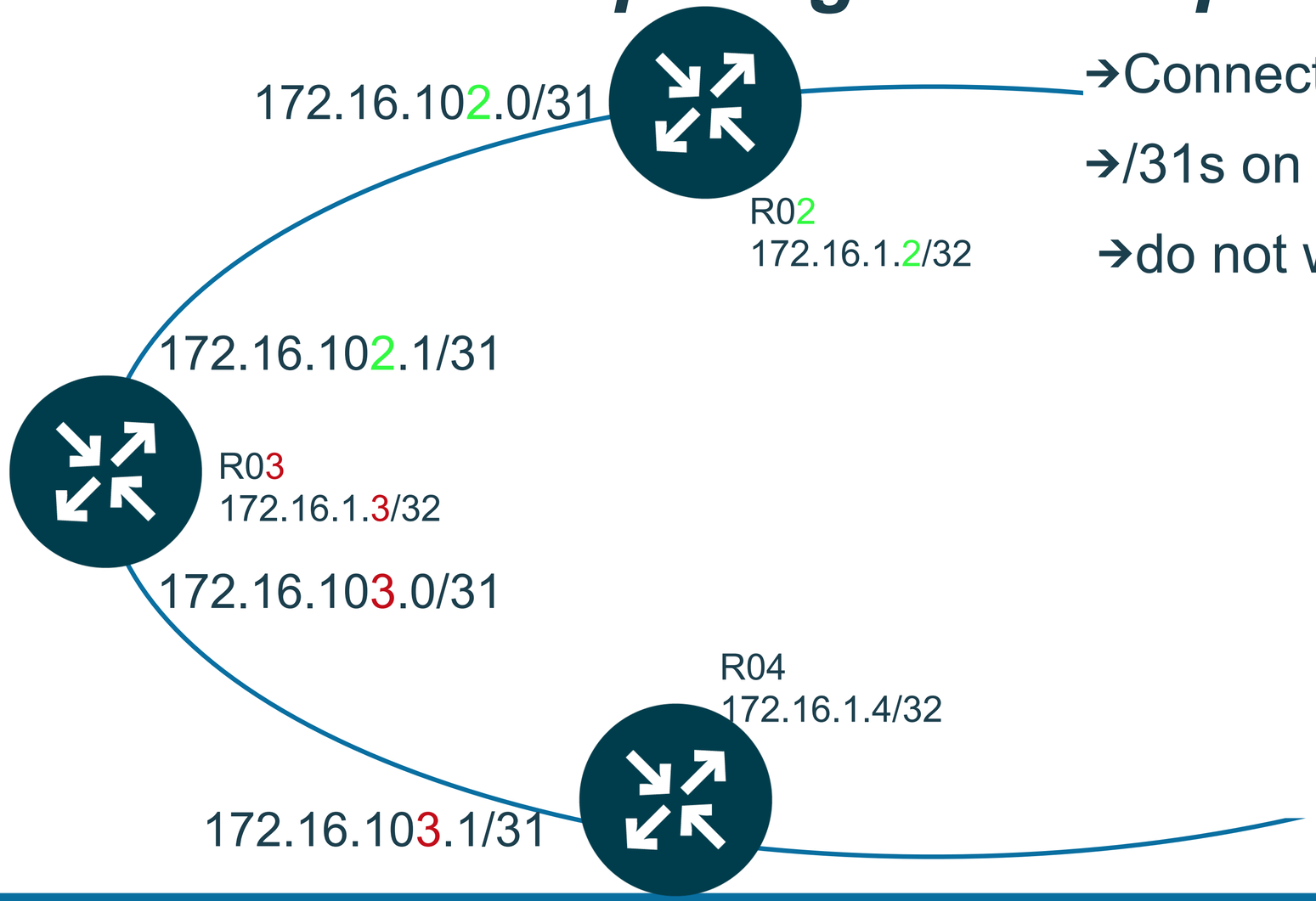


Use a Loopback address!

# Network setup: Logical Setup



# Network setup: Logical Setup

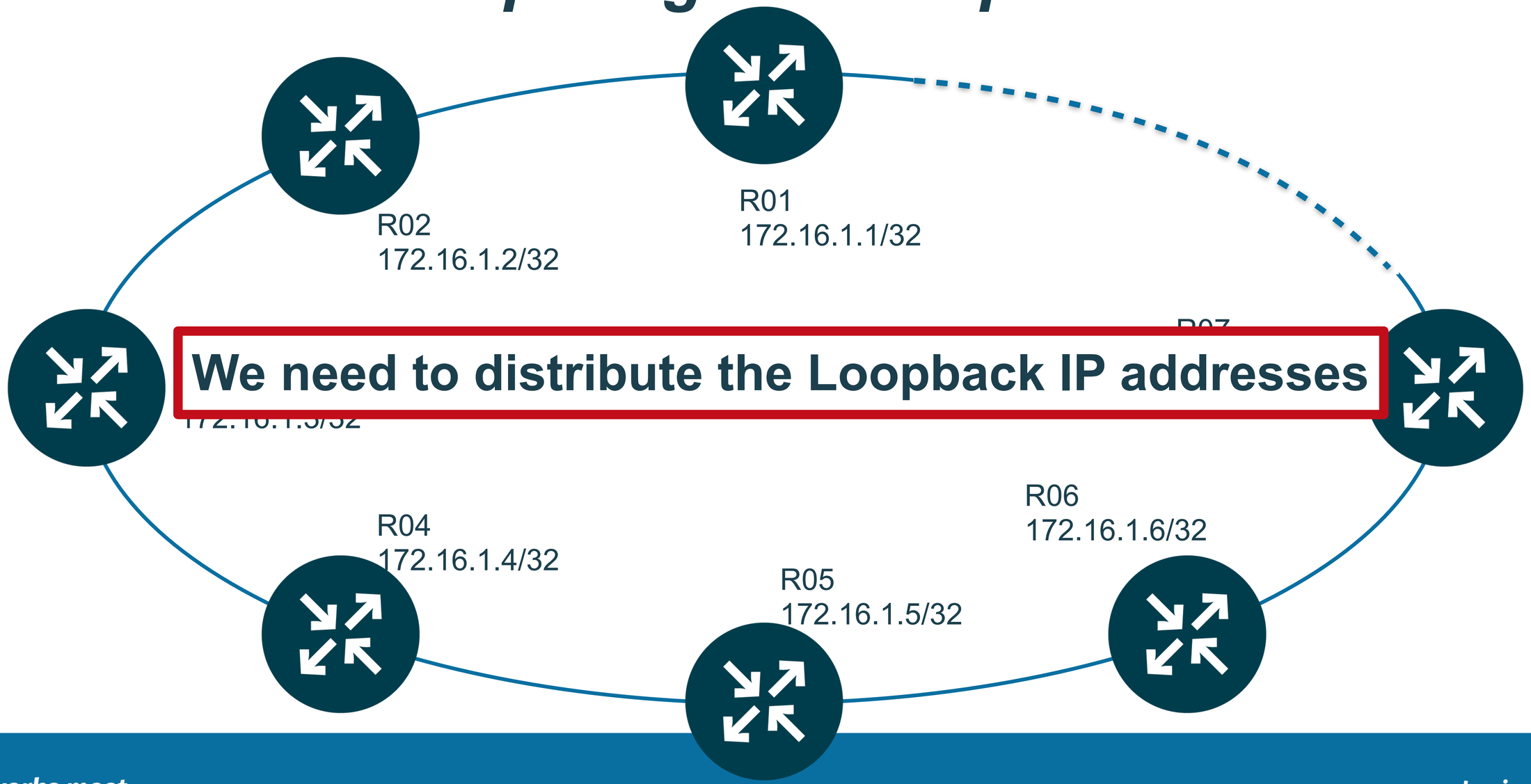



→Connecting Interfaces

→/31s on IPv4

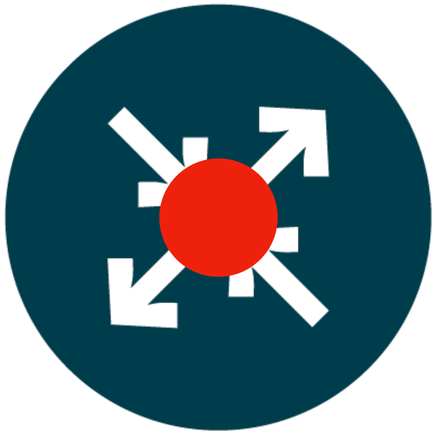
→do not waste IPv4 addresses

# Network setup: Logical Setup



# We need to distribute the Loopback IP addresses

- ➔For this we need (another) routing protocol
- ➔OSPF - Open Shortest Path First
  - ➔works on IPv4 only
  - ➔still widely used
- ➔OSPFv3
  - ➔works on IPv6 link-local addressing
- ➔IS-IS
  - ➔is truly protocol independent, works on Layer 2 directly
- ➔Static routing, RIP, EIGRP
  - ➔yes, this also works but is not recommended



# *Use OSPFv2 + OSPFv3 or IS-IS*

➔Most of the time not your choice

➔In an existing network you have to use whats there

➔...and what is supported best by your routers...

➔Clean slate installation: Use IS-IS

# *BGP Session Setup*

→BGP uses **TCP for transport**

→TCP already provides **reliable** transport

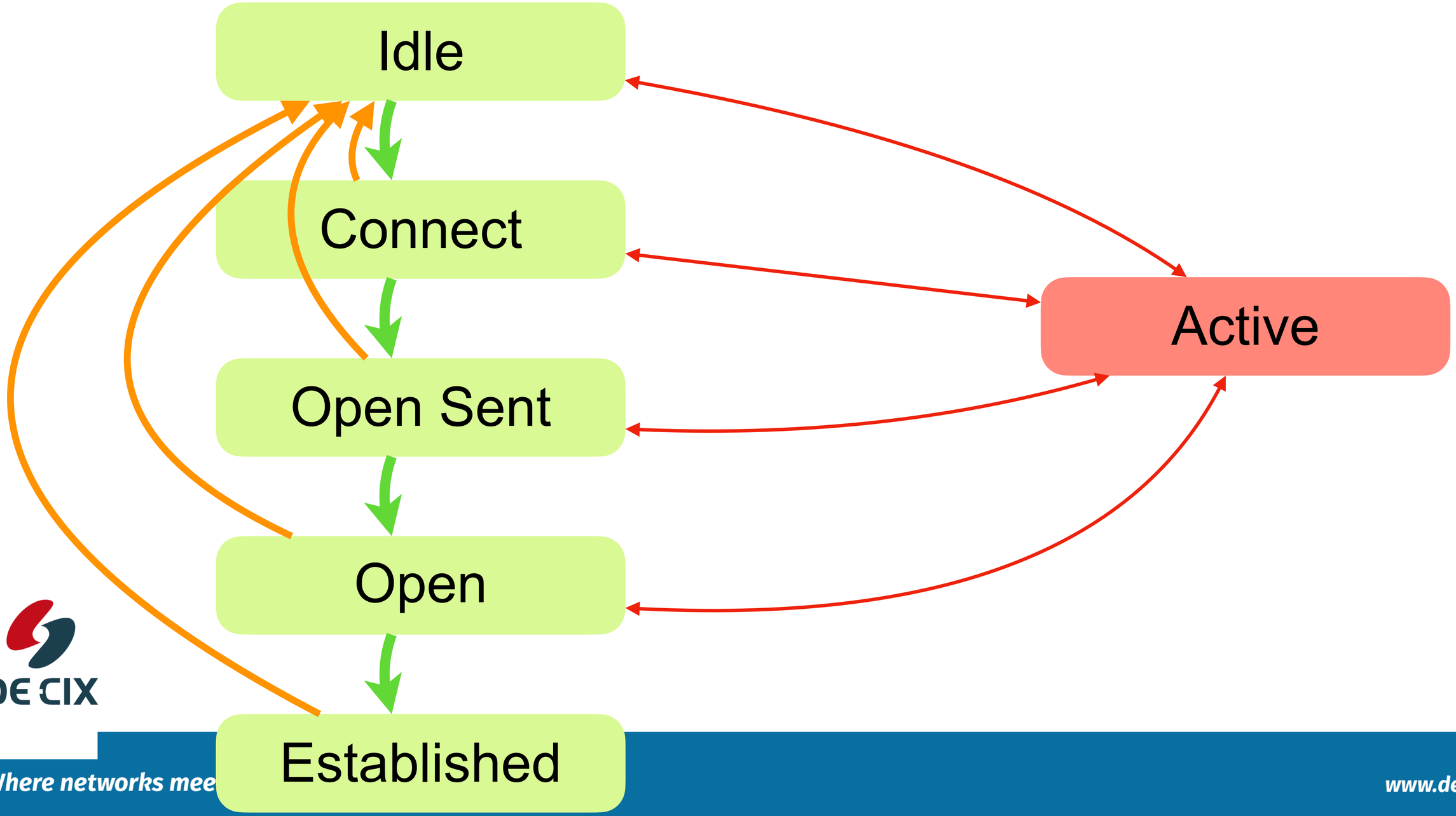
→but a bit more is needed

→some information exchange at **setup**

→some mechanism for **keepalive**

→a **state model** and timers

# *Life cycle of a BGP session (incomplete)*



# Experiment: Setup iBGP



experiment 01d, 01e
**You need file: 01d**

# *iBGP - why fully meshed?*

→BGP receives prefixes from external - eBGP

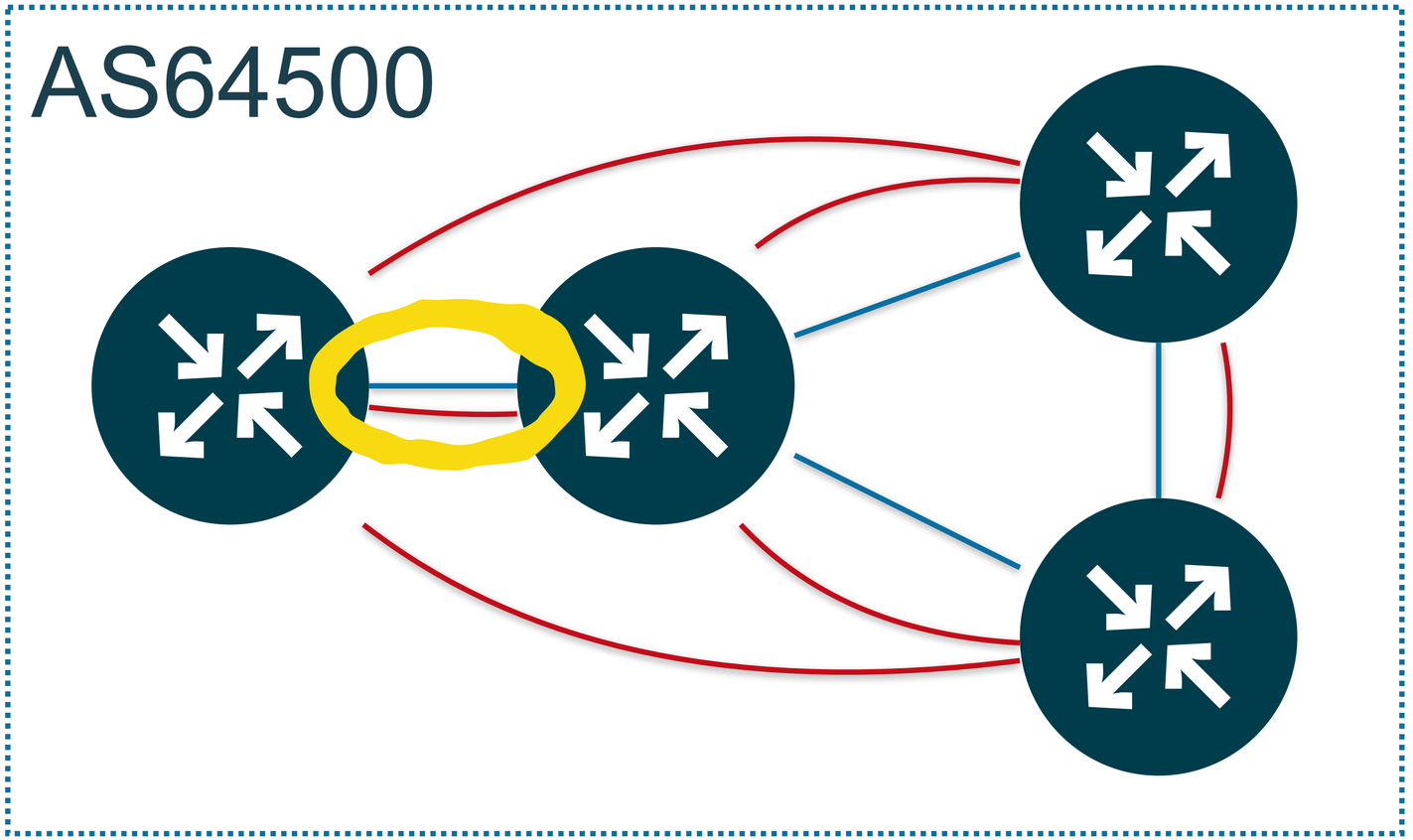
→BGP sends **all** prefixes to external (unless filtered)

→BGP sends prefixes **received from external** to internal

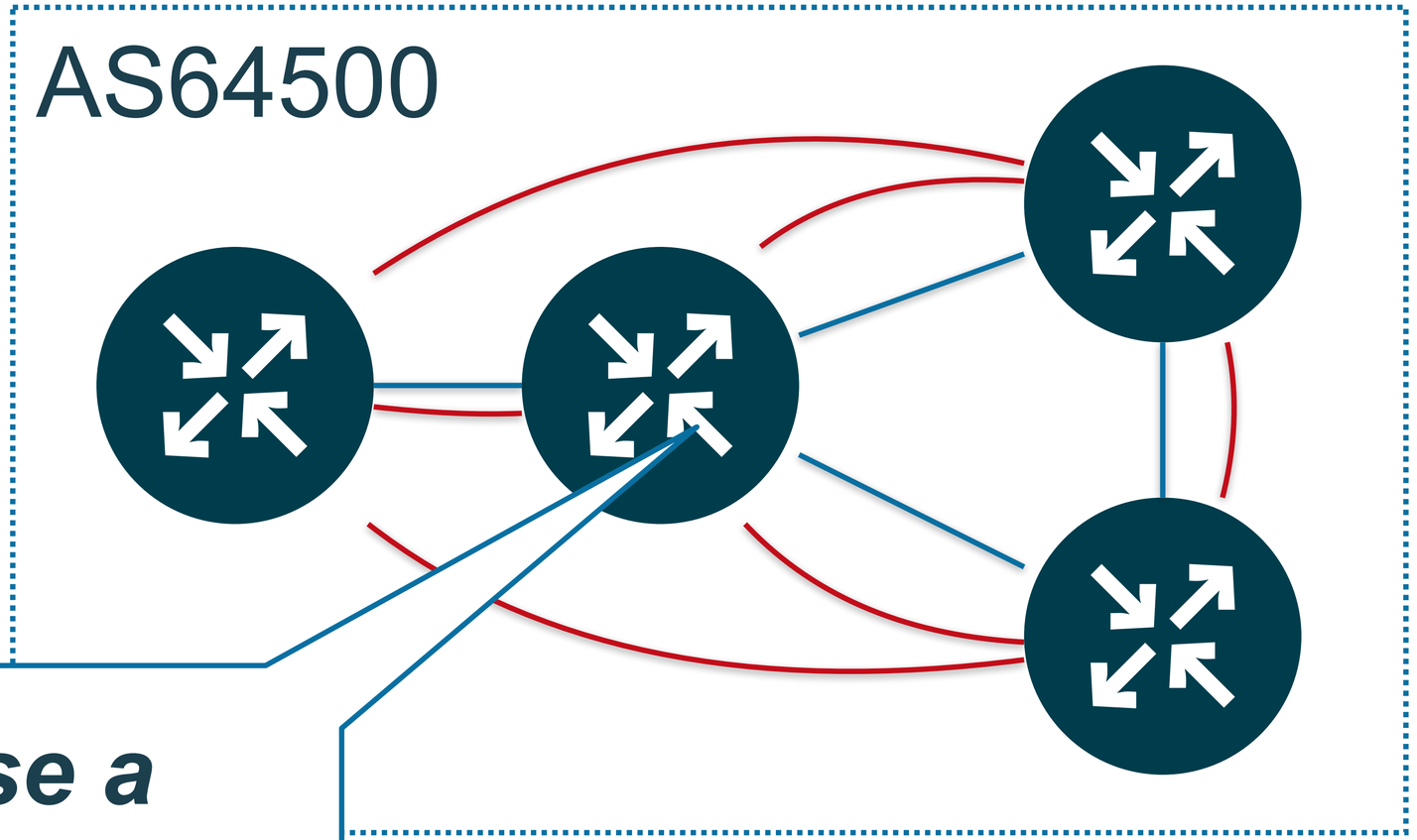
→BGP does **not** send prefixes received from internal to internal

→**unless...**

# *Example Network: Fully meshed iBGP?*

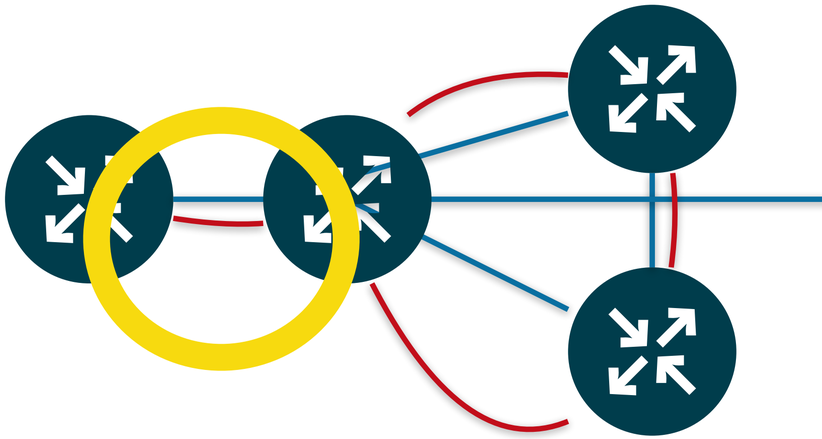


# *Example Network: ~~Fully meshed~~ iBGP?*




***Use a route reflector***

# *iBGP: Route Reflector*

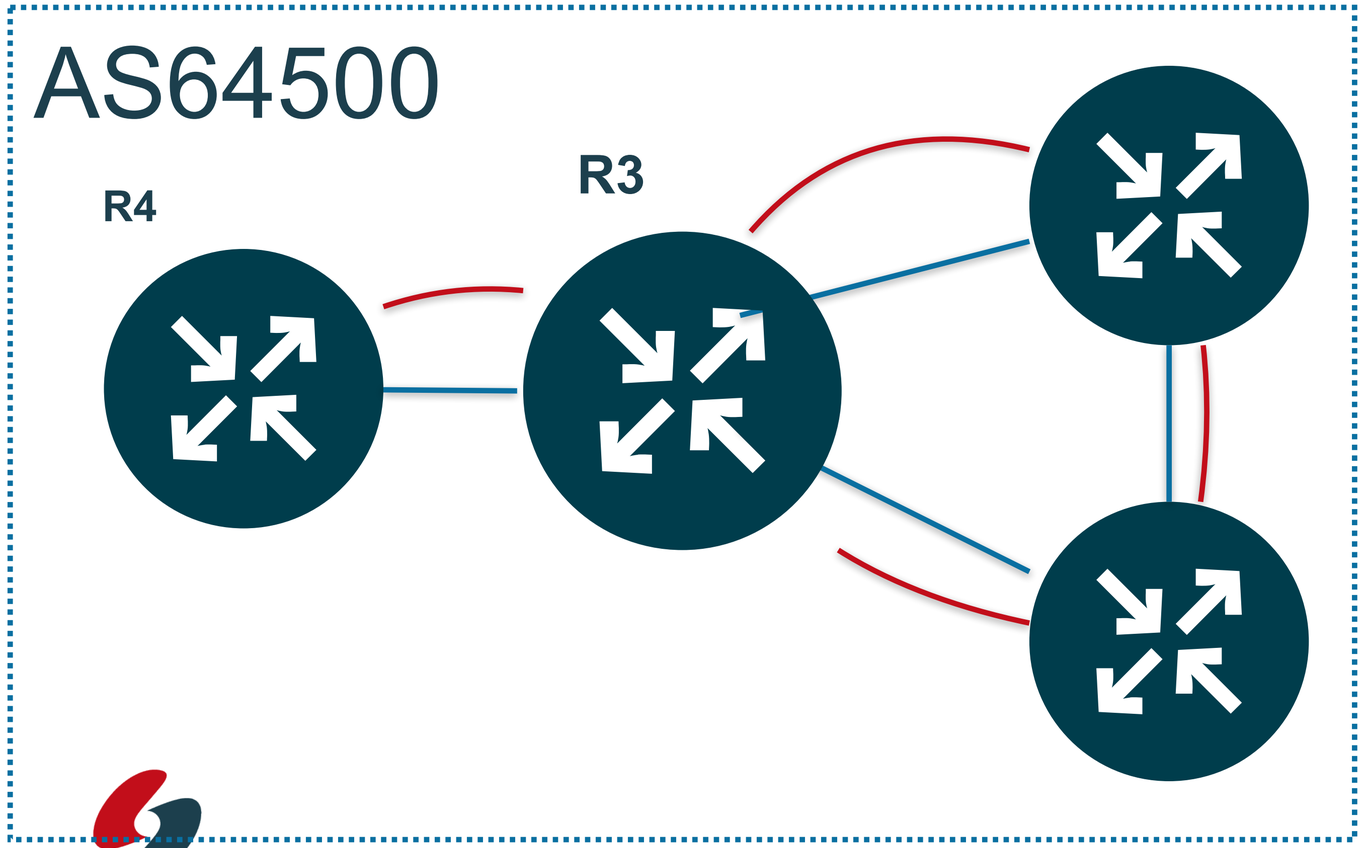


- "Normal" prefix forwarding rule for iBGP
  - do **not** send out anything learned via iBGP
- route-reflector
  - defined in RFC4456
  - send out one best path of all prefixes to each route-reflector client
- how to configure

```
neighbor x.x.x.x route-reflector-client
```

  - no special config on client side

# iBGP: Config (route-reflector)




```
! r3
router bgp 64500
 neighbor internal peer-group
 neighbor internal remote-as 64500
 neighbor internal update-source Loopback0
 neighbor internal next-hop-self
 neighbor internal send-community both
 neighbor internal-rr peer-group
 neighbor internal-rr remote-as 64500
 neighbor internal-rr route-reflector-client
 neighbor internal-rr next-hop-self all
 neighbor internal-rr send-community both
 !
 neighbor 192.168.1.1 peer-group internal
 neighbor 192.168.1.2 peer-group internal
 neighbor 192.168.1.4 peer-group internal-rr
```

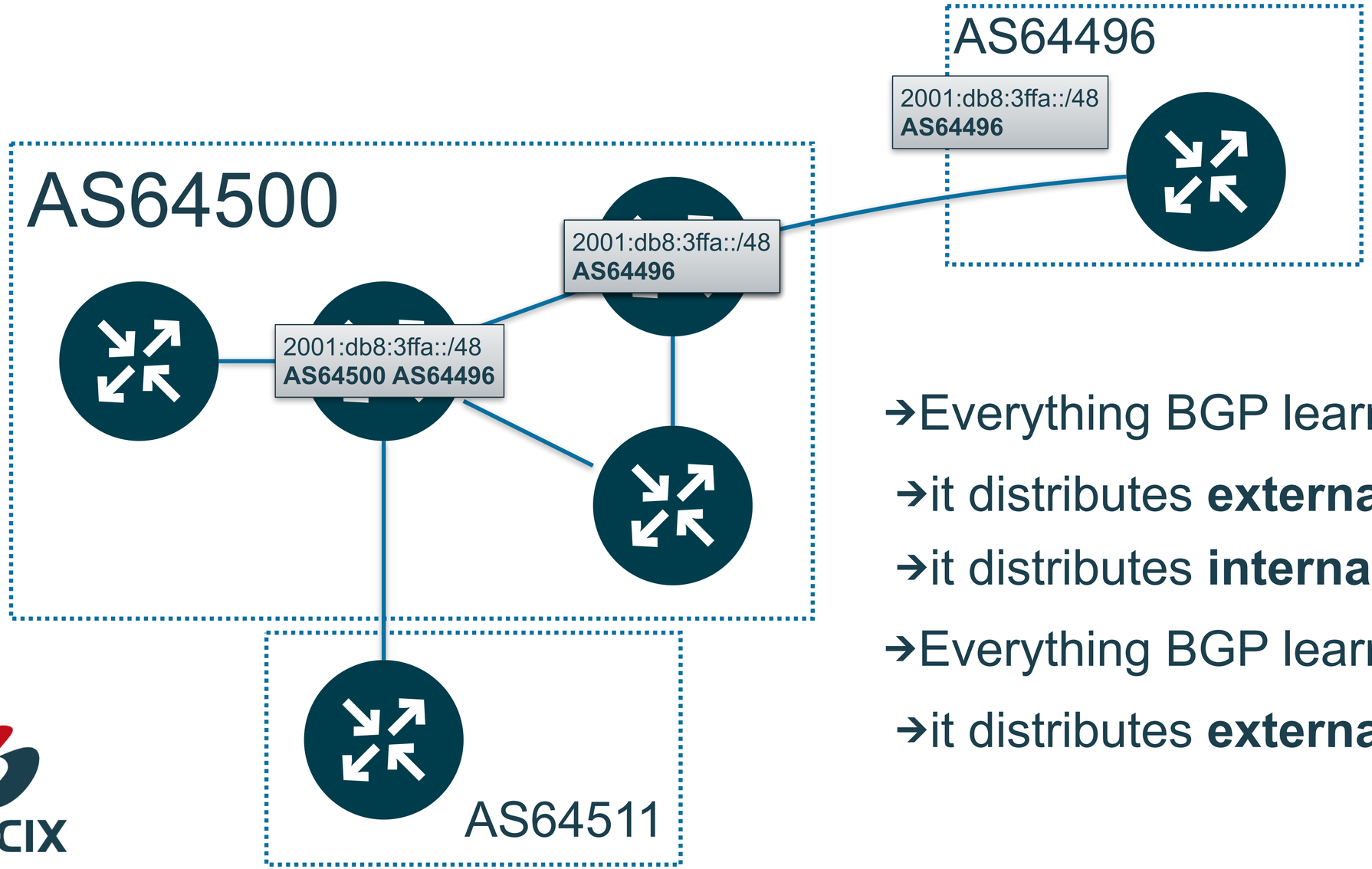
# Open Questions?

# *BGP for networks who peer*

## 02a - Setup eBGP



**Wolfgang Tremmel**
academy@de-cix.net




→Everything BGP learns from **external**

→it distributes **external**

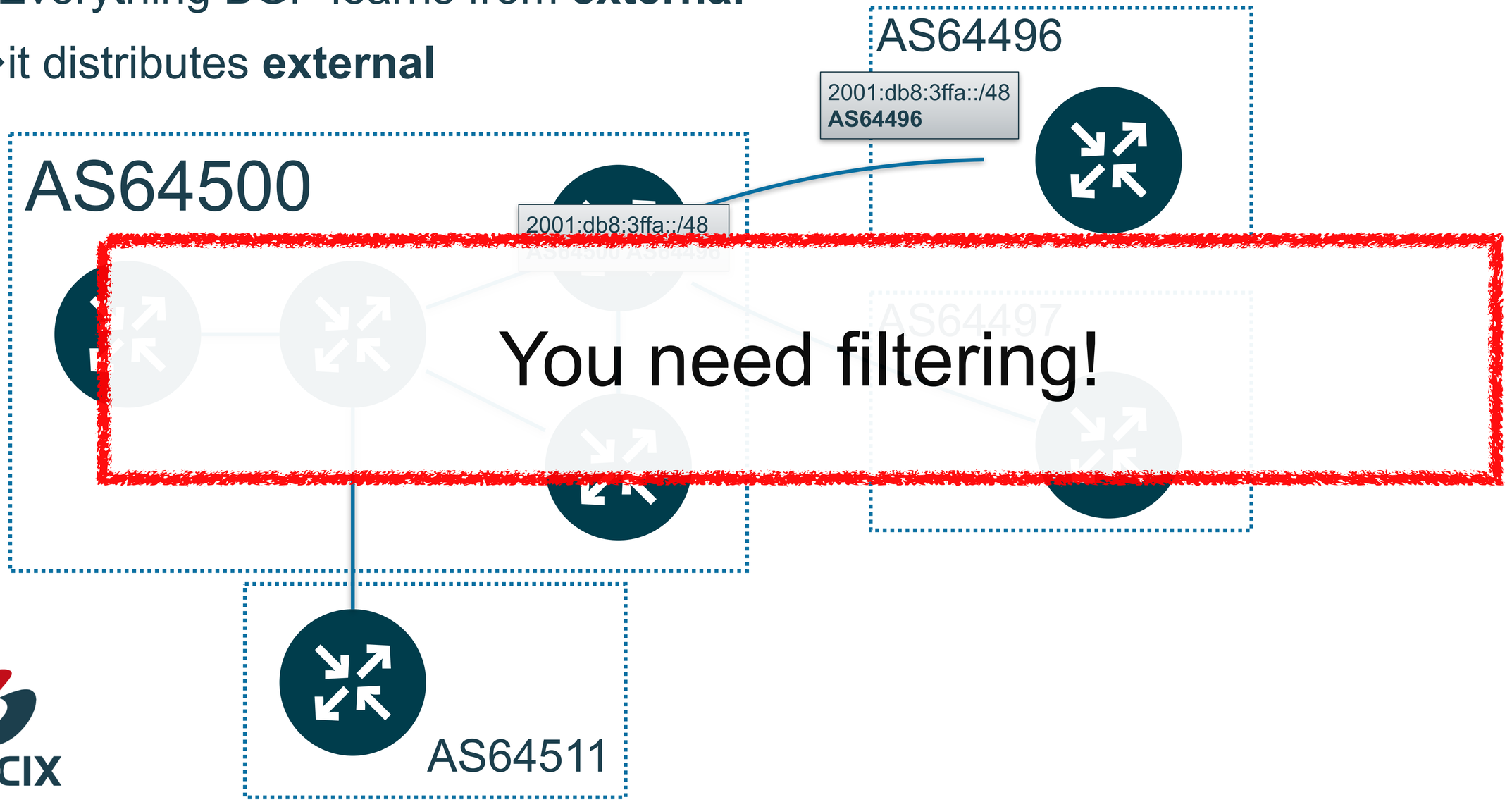
→it distributes **internal**

→Everything BGP learns from internal

→it distributes **external**

➔Everything BGP learns from **external**

➔it distributes **external**



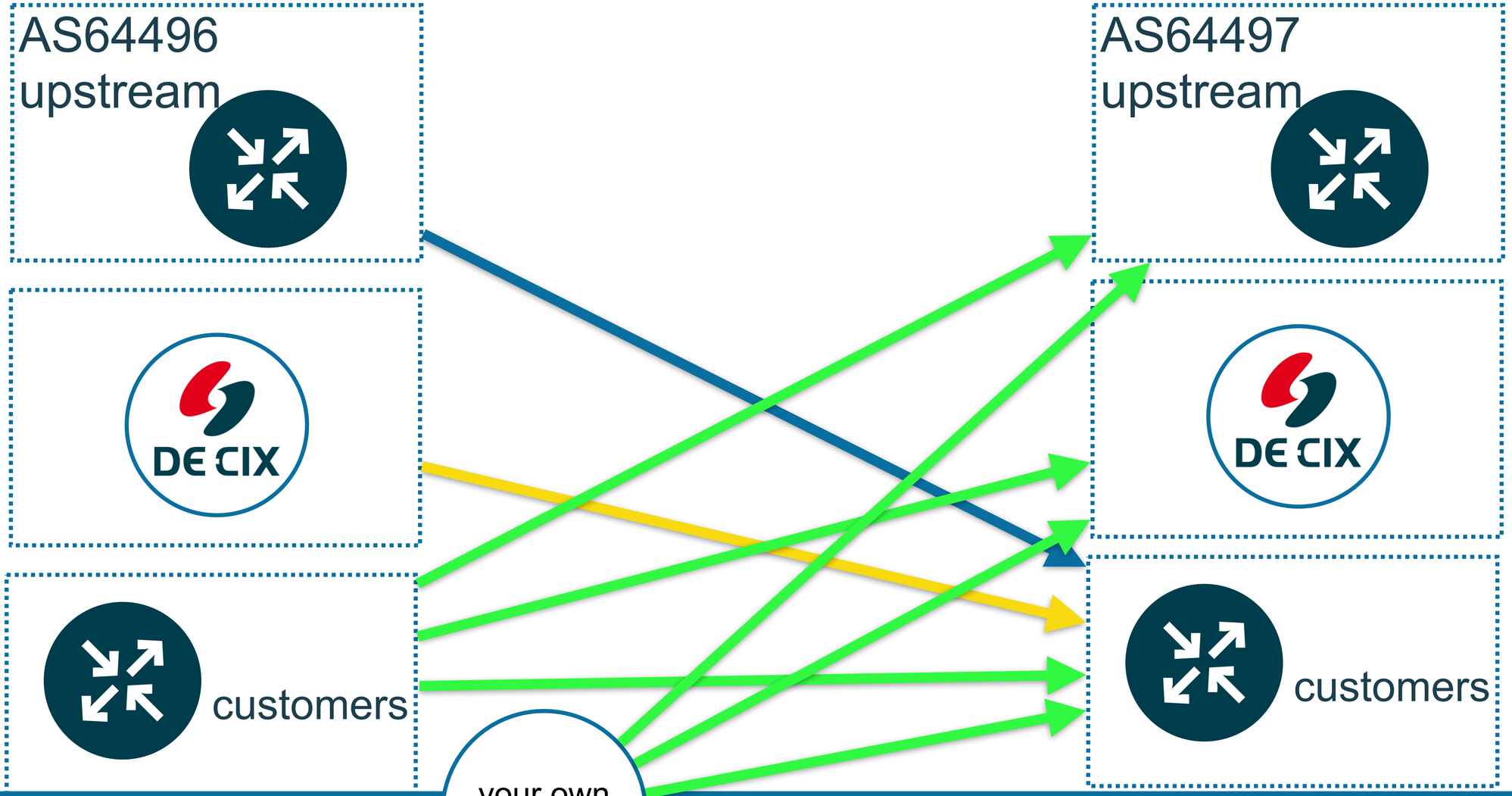
# You need filtering!

- You have multiple sources of prefixes
  - upstream provider(s)
  - peering(s)
  - customer(s)
  - your own prefixes!
- And destinations to which you announce prefixes
  - upstream providers
  - peerings
  - customers






# Prefixes
# Sources and Destinations



# *Easy filtering for beginners*

➔Deny everything outgoing

➔Allow everything incoming

```
route-map upstream-out deny 100
!
route-map upstream-in permit 100
!
```

➔Open filters step by step to allow certain prefixes through

```
ip prefix-list my-networks permit 198.51.100.0/24
!
route-map upstream-out permit 50
 match ip address prefix-list my-networks
!
route-map upstream-out deny 100
```

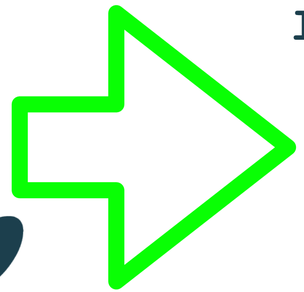
# About: Route Maps (in terms of Cisco and FRR)

- Each route-map has a name (and there is no check against typos)
- Each route-map consists of a ordered list of statements
  - Just like a BASIC program (with line numbers)
- Each statement has a result of either permit or deny

```
route-map my-great-filter permit 10
```

# *About: Route Maps Statements*

- ➔Each statement has a result of either permit or deny
- ➔also zero to many "*match" clauses*
  - ➔no *match clause* = **always true**
  - ➔more then one *match clause* are "**and**"ed together
- ➔If match(es) evaluate true, route-map is terminated and result returned



```
route-map my-great-filter permit 10
 match ip address prefix-list my-list
```

# About: Route Maps Statements

→route-maps also can have none to many set-statements

→if match-statements evaluate true (or if there are no match statements)

→all set-statements are executed

→route-map terminates and result is returned

```
route-map my-great-filter permit 10
 match ip address prefix-list my-list
 set local-preference 1000
```



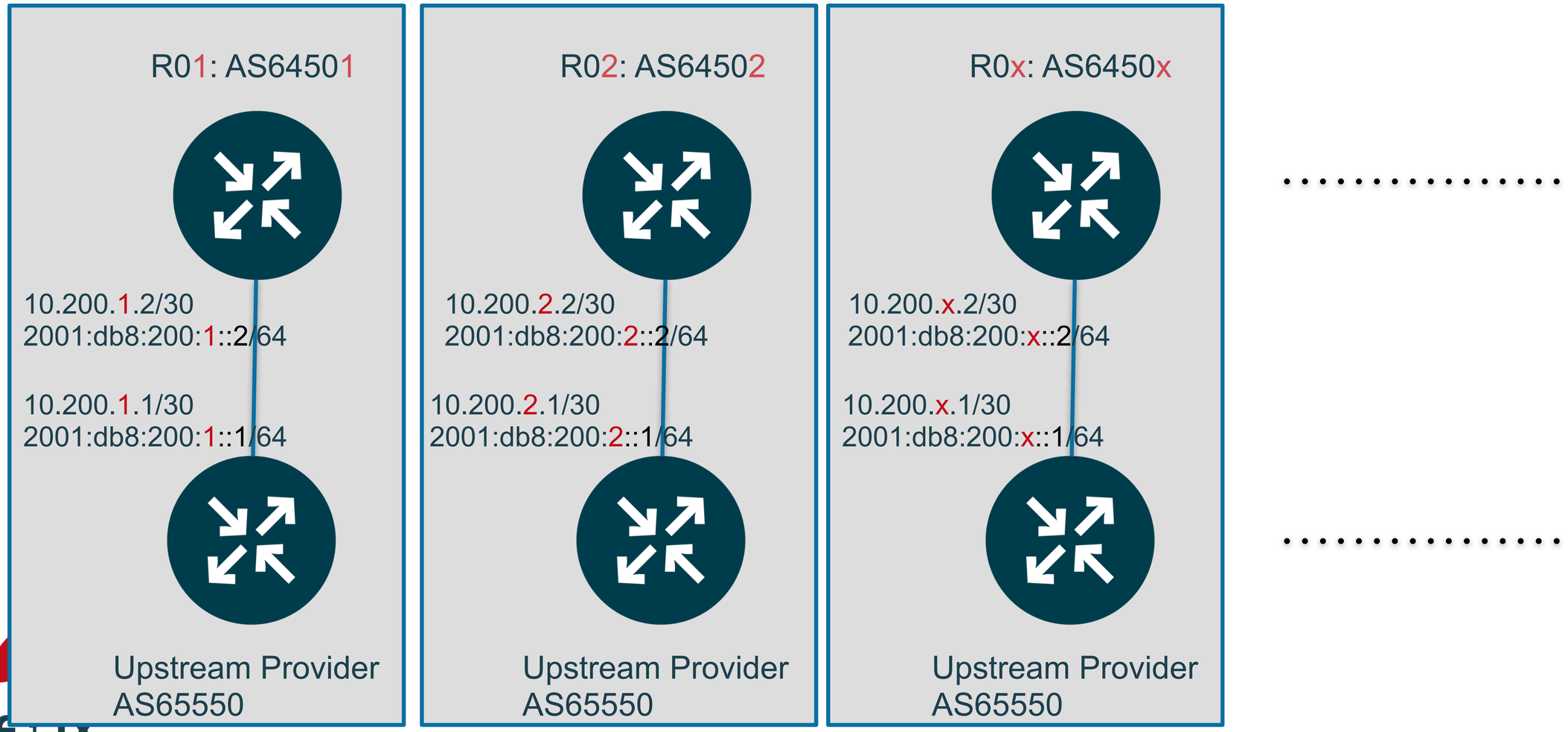
# Example: Filter for receiving prefixes

```
route-map upstream-in deny 10
 match ip address prefix-list ipv4-unwanted
 match ipv6 address prefix-list ipv6-unwanted

route-map upstream-in deny 20
 match as-path 100

route-map upstream-in permit 1000
 set local-preference 10
```

# Network setup: Logical Setup



# Configure eBGP: Peer Group

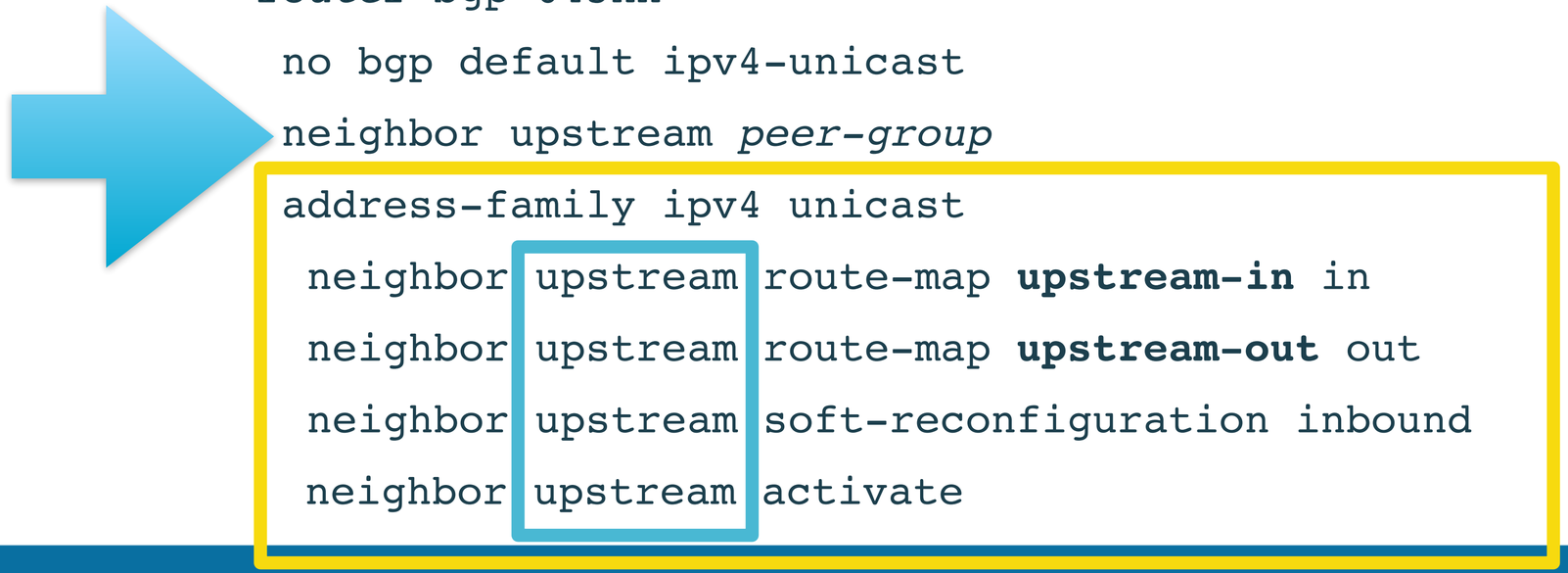
➔we group common commands in a *peer group*

➔we might have multiple upstreams with multiple AS numbers,

➔ so we do not add the AS number to the peer-group

➔Remember our filters? **upstream-in** for in, **upstream-out** for out

➔Everything related to the **announced address family** goes into the address-family context



# *Configure eBGP: Neigbor(s)*

- We have a peer-group, so we only need what is **unique** to each neighbor
  - statements configured in the peer-group are **inherited** by each member
- In this case, only the AS number is unique
  - (and the neighbor IP address)

```
router bgp 645XX
  neighbor 10.200.XX.1 remote-as 65550
  neighbor 10.200.XX.1 peer-group upstream
```

# *We start with simple filters for eBGP*

➔Configure filters (route-map) for in and out

```
route-map upstream-in permit 100
route-map upstream-out deny 100
```

➔The **in** filter lets everything through

➔no match statement = always true

➔result "permit" is returned for every prefix

➔The **out** filter blocks everything

➔result "deny" is returned for every prefix

# Experiment: Configure eBGP



experiment 02a - Setup eBGP
start exabgp!

# Open Questions?

# *BGP for networks who peer*

## 02b - Become Multi-Homed

**Wolfgang Tremmel**
academy@de-cix.net

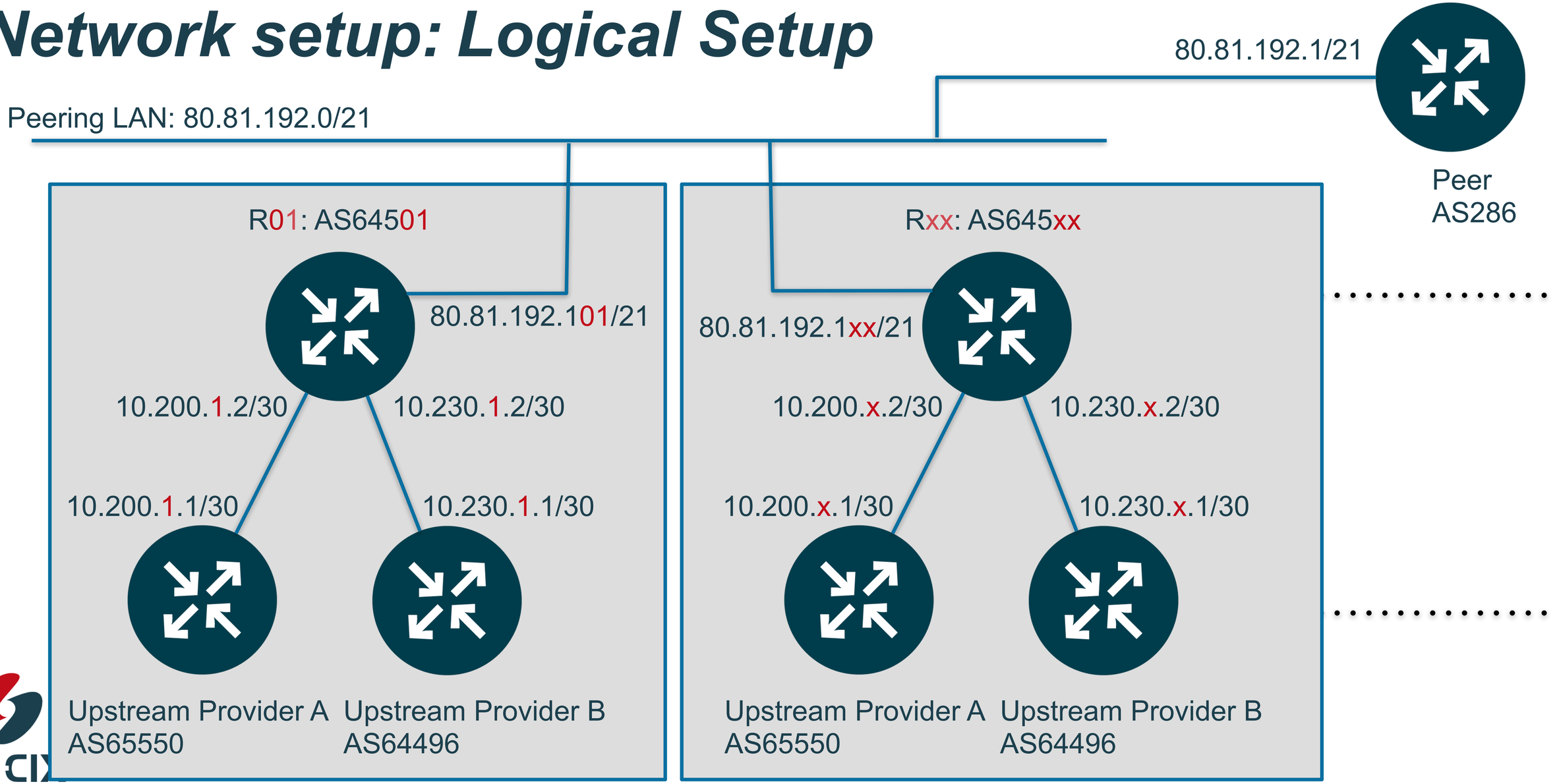


DE CIX

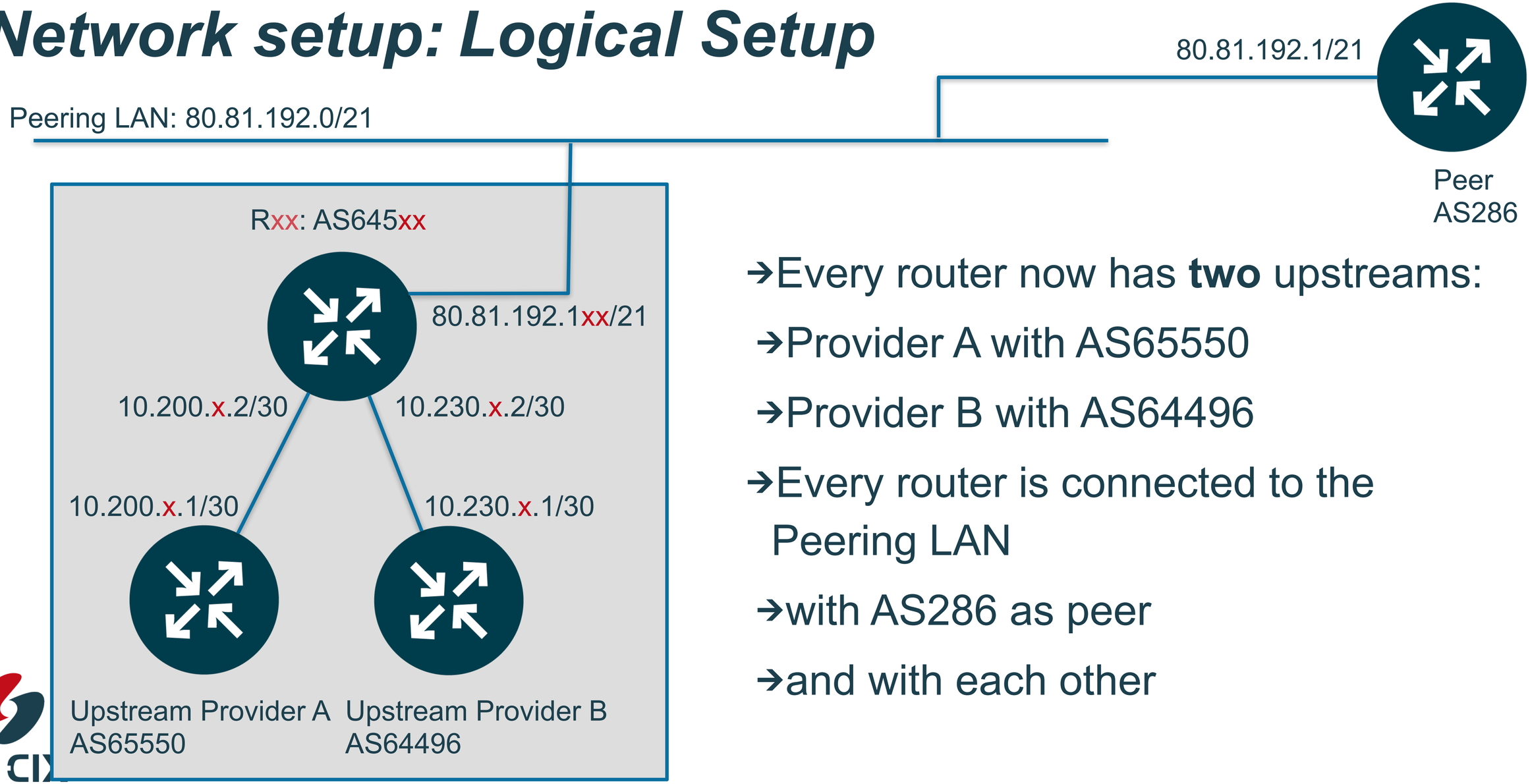
*Where networks meet*

www.de-cix.net

# Network setup: Logical Setup



# Network setup: Logical Setup




- Every router now has **two** upstreams:
  - Provider A with AS65550
  - Provider B with AS64496
- Every router is connected to the Peering LAN
  - with AS286 as peer
  - and with each other

# Experiment: Configure eBGP



experiment 02b - become multi homed (2 upstreams)

# Open Questions?

# *BGP - becoming multihomed*

## 02c - Adding multiple upstreams and peering

**Wolfgang Tremmel**
wolfgang.tremmel@de-cix.net





DE CIX

*Where networks meet*

www.de-cix.net

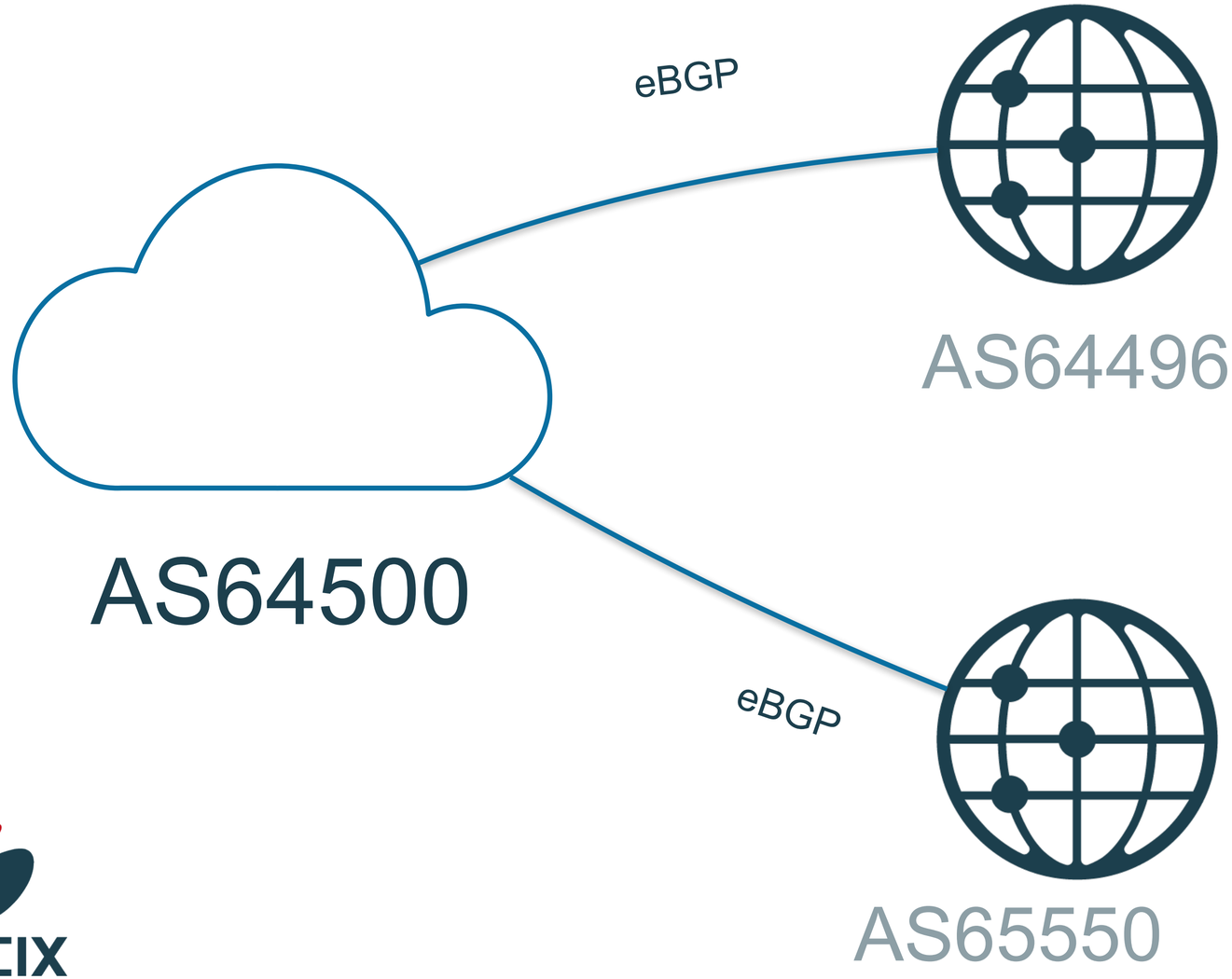
# Multihomed

- → Multiple Upstreams
  - For redundancy
  - For cost optimization
- → Peering
  - For even better performance
  - For even more resilience




DE CIX

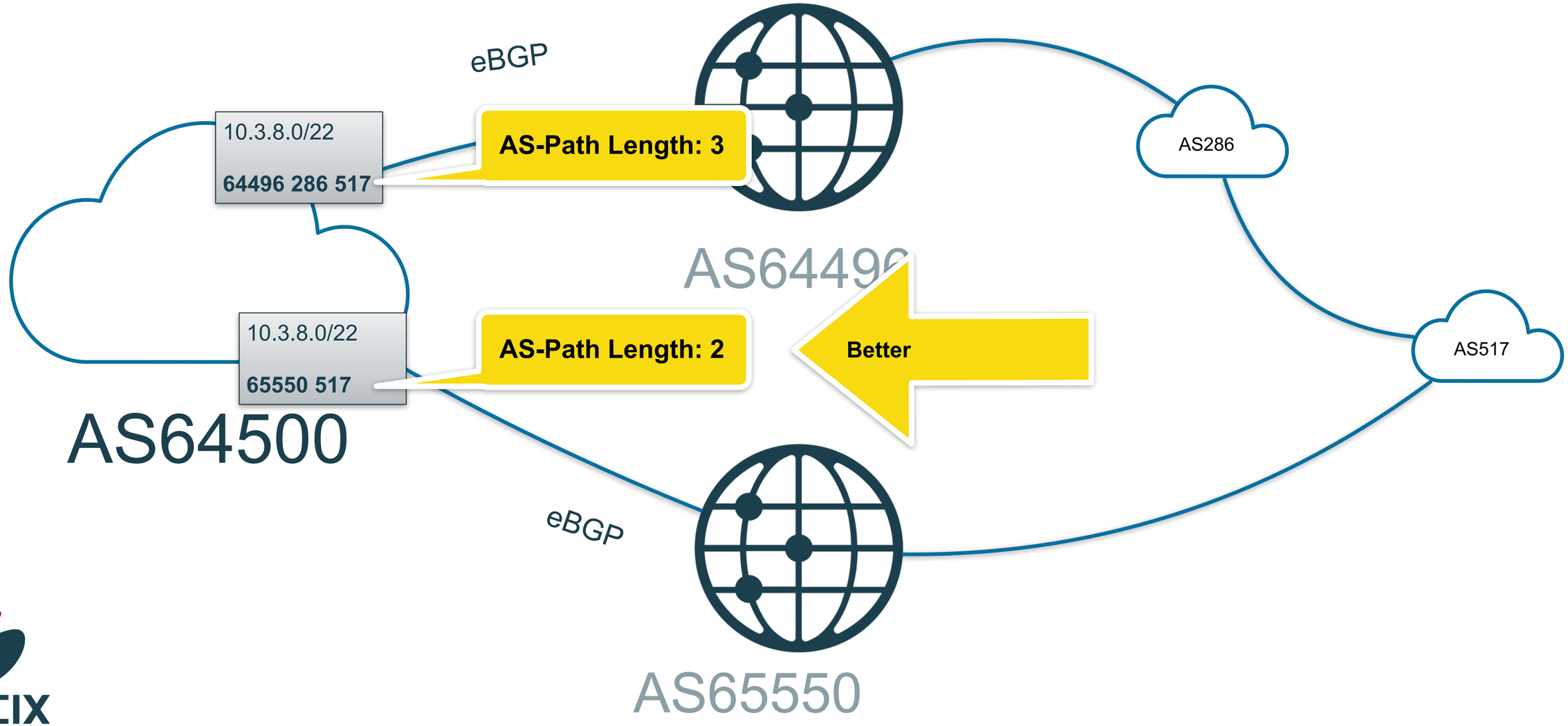
# Let's get started.... with two upstreams



# Let's get started.... with two upstreams



# Let's get started.... with two upstreams



# The BGP Best Path Selection Algorithm

| | | |
|---|---|---|
| 1 | NextHop reachable? | Continue if "yes" |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

AS-Path Length: 3

AS-Path Length: 2

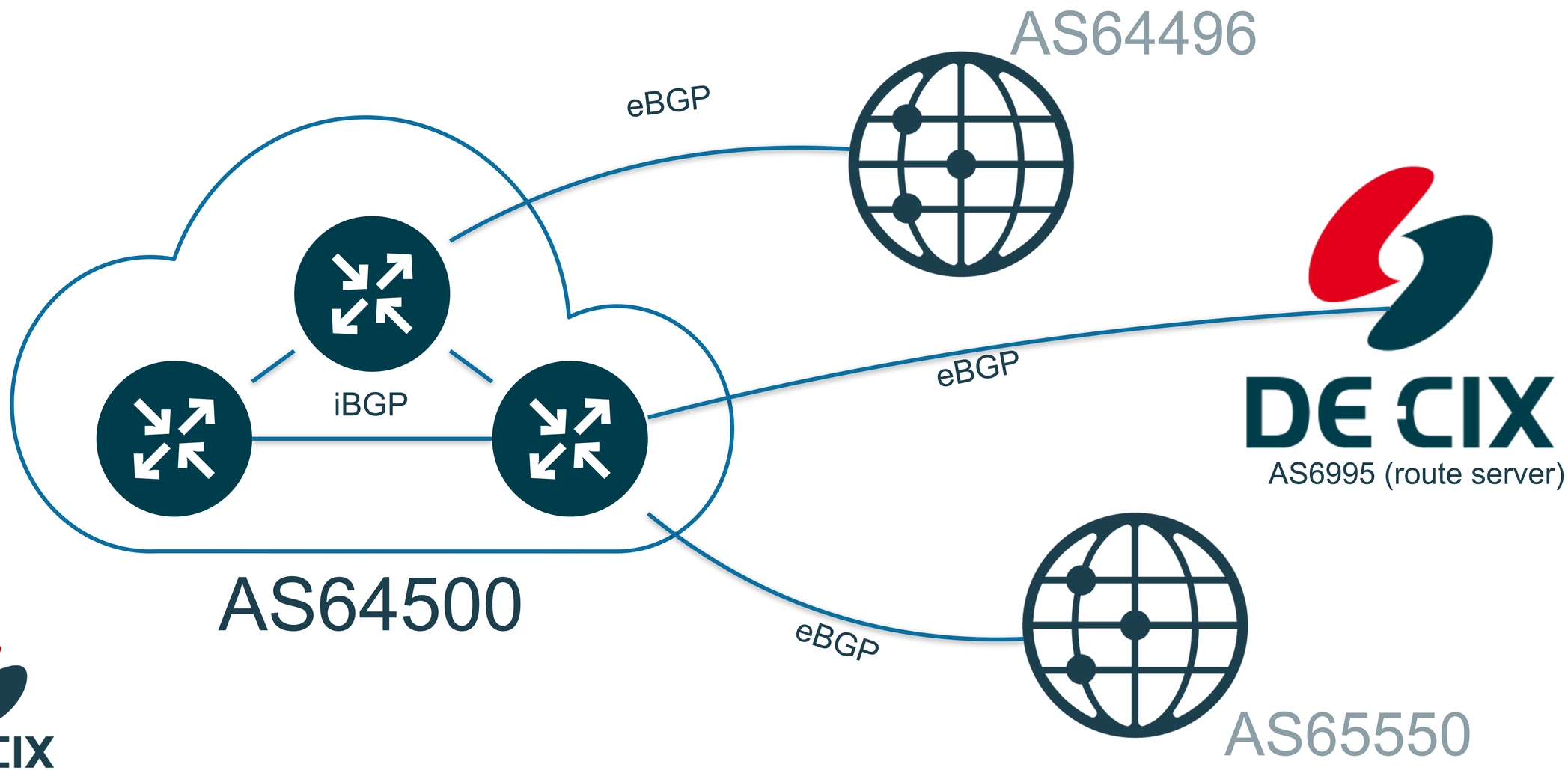
Better

DE CIX

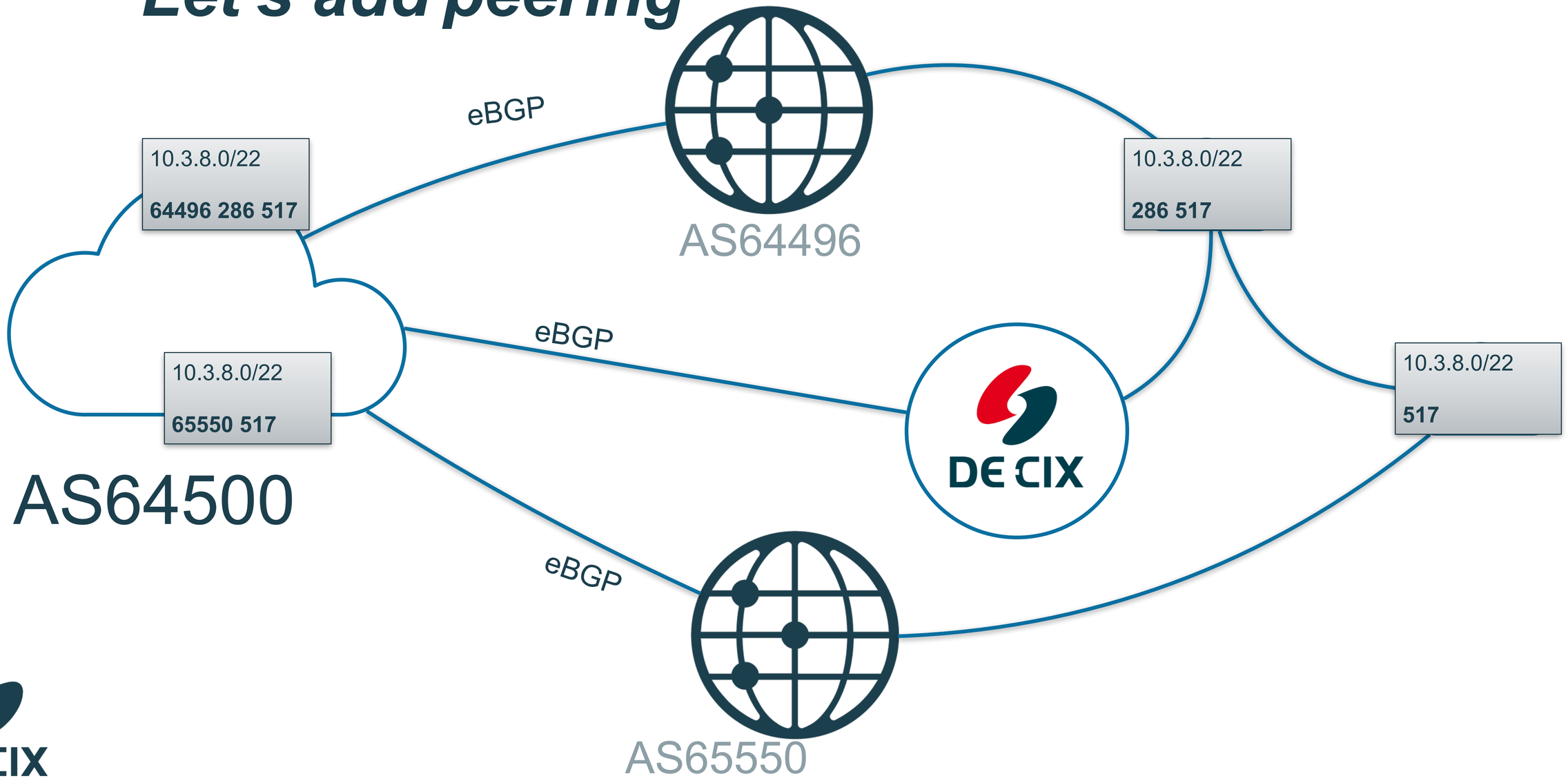
Where networks meet

www.de-cix.net

# Let's continue...

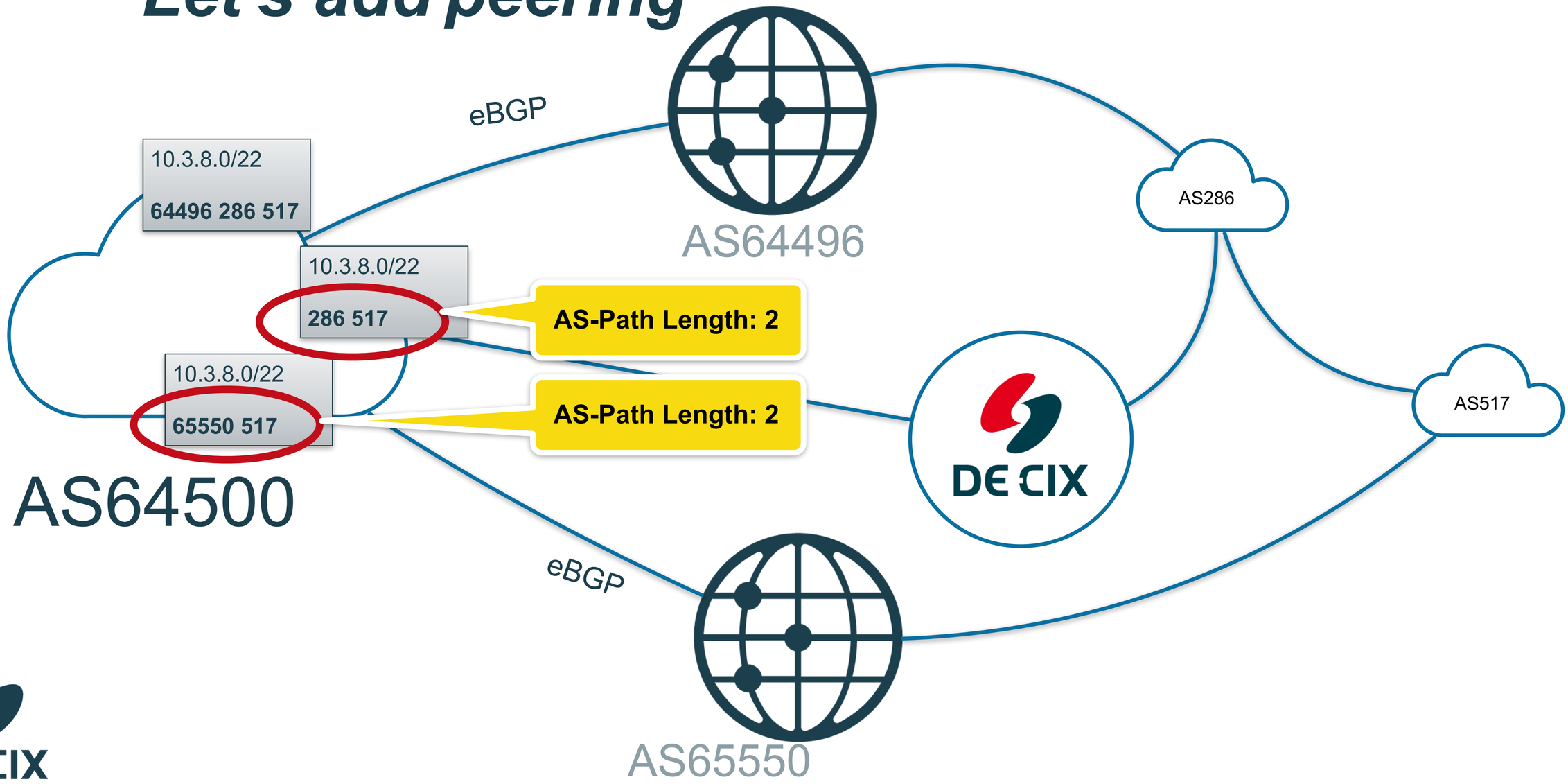


# Let's add peering



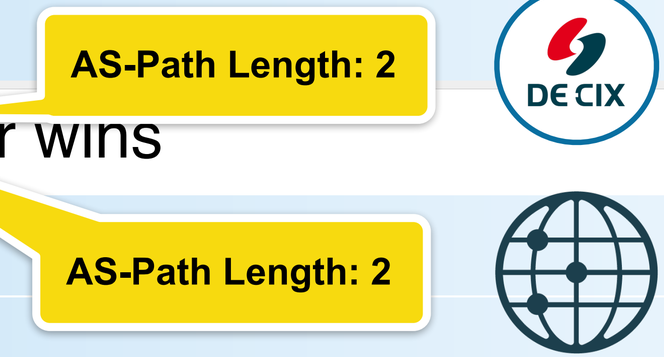

DE CIX

# Let's add peering



# The BGP Routing Algorithm

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | | |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |




DE CIX

# *Local Preference*

- Higher wins
- Integer value (32bit, 0-4294967295)
- Propagated via iBGP inside an Autonomous System
- Set using a route-map when receiving prefixes
- Typical values:
  - Customer prefixes: 10000
  - Peering prefixes: 1000
  - Upstream prefixes: 10

Why am I not using "100" here?

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

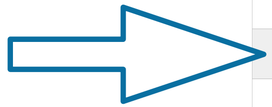
# *Local Preference - how to set*

- High level:

```
if (prefix received from customer)
 then set local-preference of prefix = 10000
else if (prefix received from peer)
 then set local-preference of prefix = 1000
else
 set local-preference of prefix = 10
```

- Our experiment

```
route-map peering-in permit 100
 set local-preference 1000
route-map upstream-in permit 100
 set local-preference 10
```



| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# Experiment: Configure eBGP



experiment 02b - become multi homed (add peering)
experiment 02c - Announce Prefixes

# *Summary*

- When connecting to multiple upstreams ISPs and peering, you need to define a routing policy
- This policy changes attributes of **received** prefixes
- This policy defines how your **outgoing** traffic is routed
- *Local Preference* can be used to influence this
- Otherwise *AS Path Length* is used to find the best path
- BGP has a complex route selection algorithm

# BGP route selection algorithm

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# Open Questions?

# *Summary*

- We learned about route-maps
- We configured eBGP for IPv4 and IPv6
  - we added two upstream providers to our lab router
  - we set up peering between each other
- We learned the first two criteria of BGP Best Path Selection:
  - Local Preference
    - Configured
    - Highest wins
  - AS path length
    - shortest wins

# *BGP - Best Path Selection*

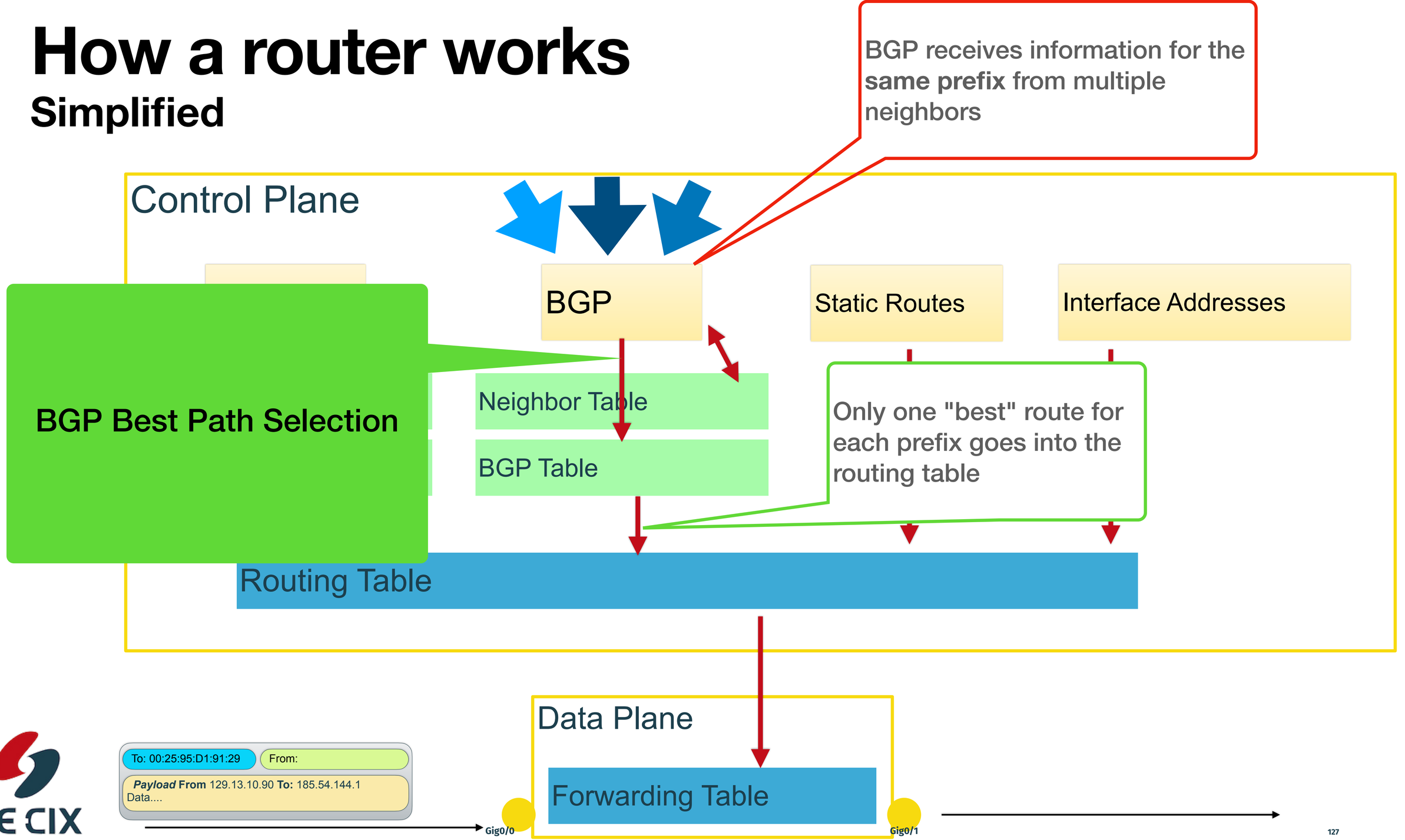
## Beyond LocalPref and AS-Path Length

**Wolfgang Tremmel**
wolfgang.tremmel@de-cix.net

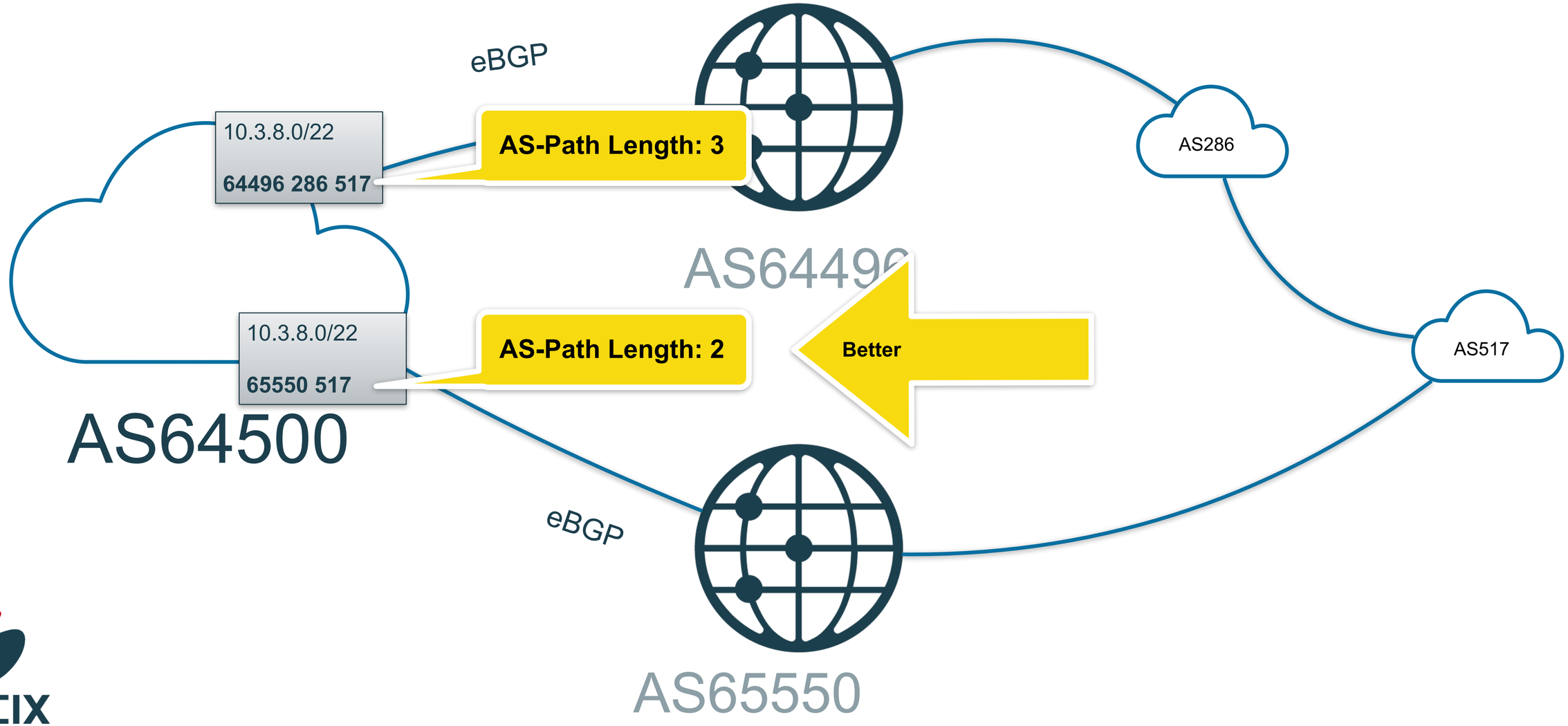


# How a router works

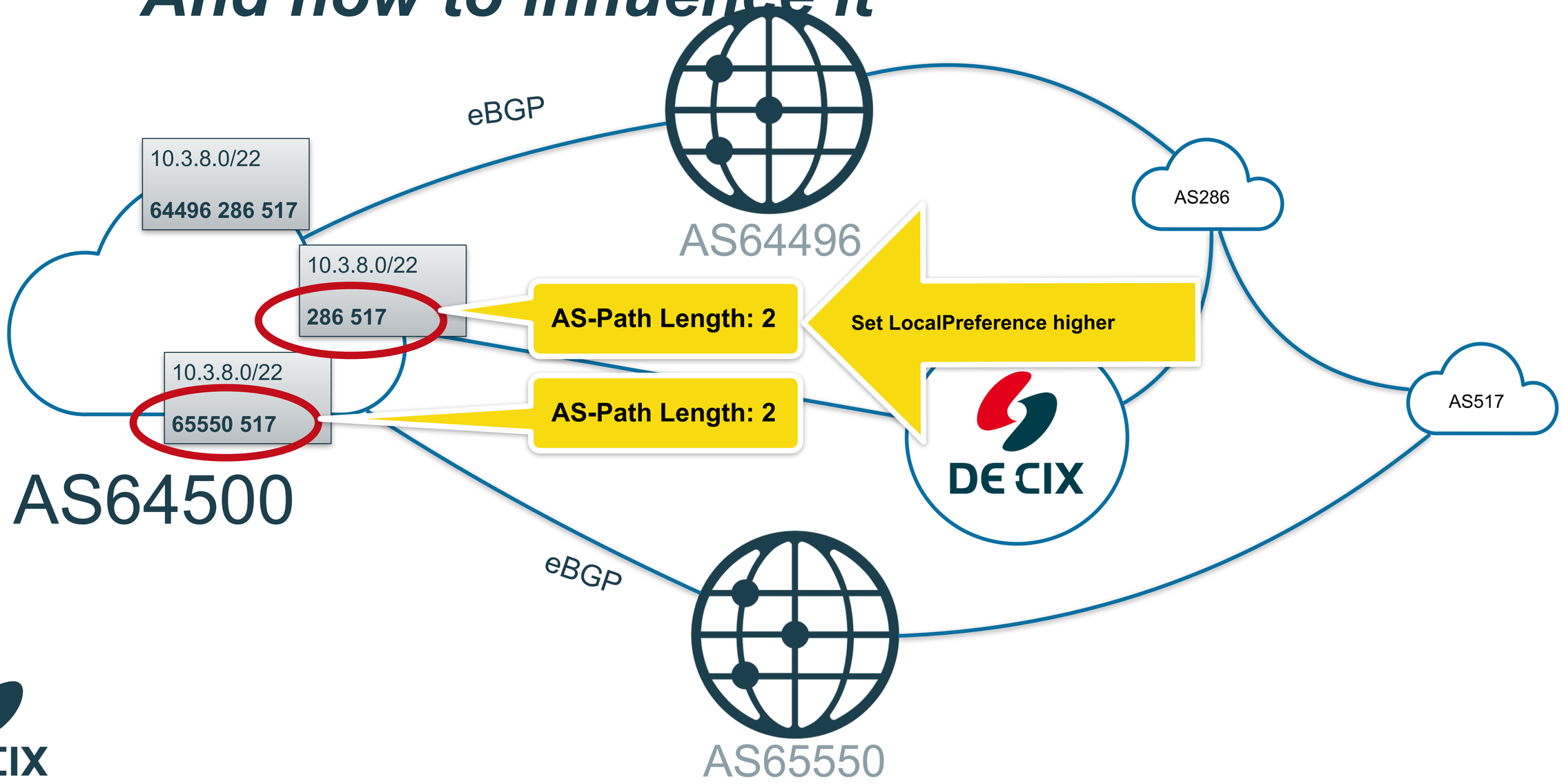
## Simplified



# *In Part 2 we talked about path selection*



# And how to influence it

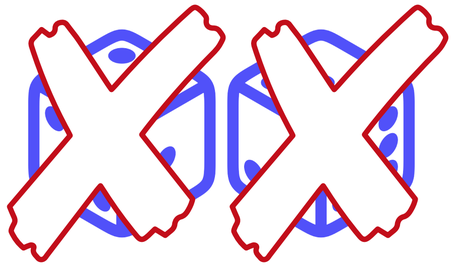


# BGP Route Selection Algorithm

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# *BGP Route Selection Algorithm: Motivation*

- ➔ Only one single path for each destination is needed (and wanted)
- ➔ Decision must be based on attributes
- ➔ And must not be random, but deterministic
- ➔ Some of the criteria will sound strange
- ➔ Some are really outdated
- ➔ and sound strange





| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# Experiment: best path selection



Experiment 3.01: Local Preference
Experiment 3.02: AS Path Length

# *BGP Route Selection: Origin Type*

- Origin Type is a "historical" attribute
- Three possible values:
  - IGP - route is generated by BGP network statement
  - EGP - route is received from EGP
  - incomplete - redistributed from another protocol
- ***This rule is not really important*** ? really?
- Fun fact: There are prefixes in the global routing table marked "e"

**E**xterior **G**ateway **P**rotocol

Predecessor of BGP which is no longer used

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# BGP Route Selection: Origin Type Examples

```
show ip bgp 1.0.4.0/22

Path #22: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.10 0.11
  Advertised to peers (in unique update groups):
    46.31.120.208
  6939 4826 38803 56203
    206.130.10.8 from 206.130.10.252 (206.130.10.2
      Origin IGP, metric 634, localpref 200, val
import-candidate, import suspect
      Received Path ID 0, Local Path ID 1, versi
      Community: 51531:35214 65101:0 65102:200 6
      Origin-AS validity: not-found
```

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# BGP Route Selection: Origin Type Examples

```
show ip bgp 1.0.137.0/24

Path #6: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.10 0.11
  Advertised to peers (in unique update groups):
    46.31.120.208
  9318 38040 23969
    80.81.[illegible] from 80.81.192.157 (80.81.192.157)
      Origin incomplete, metric 5000, localpref 200, valid, external, best, group-best,
import-candidate, import suspect
      Received Path ID 0, Local Path ID 1, version 3322655[illegible]
      Community: 9318:120 9318:8300 9318:8330 9318:902
65103:276 65104:150
      Origin-AS validity: not-found
```

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# BGP Route Selection: Origin Type Examples

```
show bgp 23.89.224.0/21

  Path #3: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    91.214.253.230  80.81.196.222   2a02:c50:db8:7:1:0:ed5f:0
    2a02:c50:db8:2:0:a30                    91.214.253.223  185.230.223.51
    91.214.253.4    91.                02:c50:db8:3:a066:b1ff:fe64:7515
    80.81.196.223
  15169 396982
    80.                 108 from 80.81.192.157 (80.81.192.157)
      Origin EGP, metric 0, localpref 1000, valid, external, best, group-best
      Re                Path ID 0, Local Path ID 1, version 72809868
      Community: 65101:1002 65102:1000 65103:276 65104:150
      Large Community: 6695:1000:1 196610:0:11276 196610:0:
      Origin-AS validity: not-found
```

Weird!

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# *Experiment: best path selection*



Experiment 3.03: Origin Type

# BGP Best Path Selection

## Origin Type

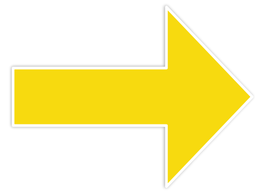
- Origin Type is a "historical" attribute (or is it not?)
- Set by Originator AS
- **SHOULD NOT** be changed by forwarding ASes
- Three possible values:
  - IGP - route is generated by BGP network statement
  - EGP - route is received from EGP
  - incomplete - redistributed from another protocol

| | | |
|---|---|---|
| 1 | NextHop reachable? | Continue if "yes" |
| 2 | Local Preference | higher wins |
| 3 | AS Path Length | shorter wins |
| 4 | Origin Type | IGP over EGP over Incomplete |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

DE CIX

# Tier1 providers rewriting Origin-Type

## From the BGP-dirty-tricks-department



| | | |
|---|---|---|
| 1 | NextHop reachable? | Continue if "yes" |
| 2 | Local Preference | higher wins |
| 3 | AS Path Length | shorter wins |
| 4 | Origin Type | IGP over EGP over Incomplete |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

- If you have two or more transit providers
  - And set the same Local Preference for all
  - You receive the same **prefixes** from all of them
  - With the **same AS path length** for a lot of them
    - The one who rewrites Origin-Type to "IGP" gets more traffic

DE CIX

# Countermeasures?

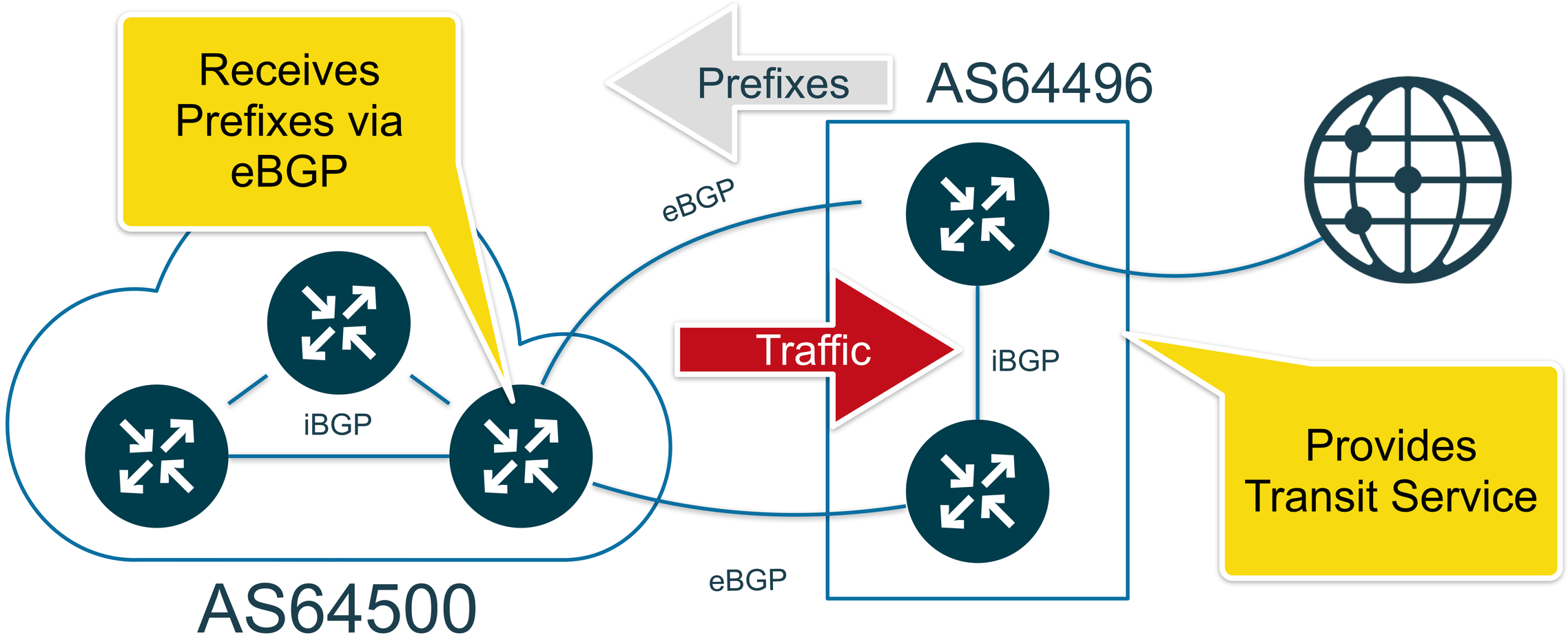
## Or simply give up?

- With rewriting origin-type, the attribute has lost its meaning
- Can any BGP implementation turn off Origin-Type evaluation? No (?)
- If you have two or more transit providers and no BGP customers
  - you can simply re-write origin-type for all incoming prefixes
- If you have BGP customers
  - talk to them first before doing something like that

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | ~~Origin Type~~ | ~~IGP over EGP over Incomplete~~ |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

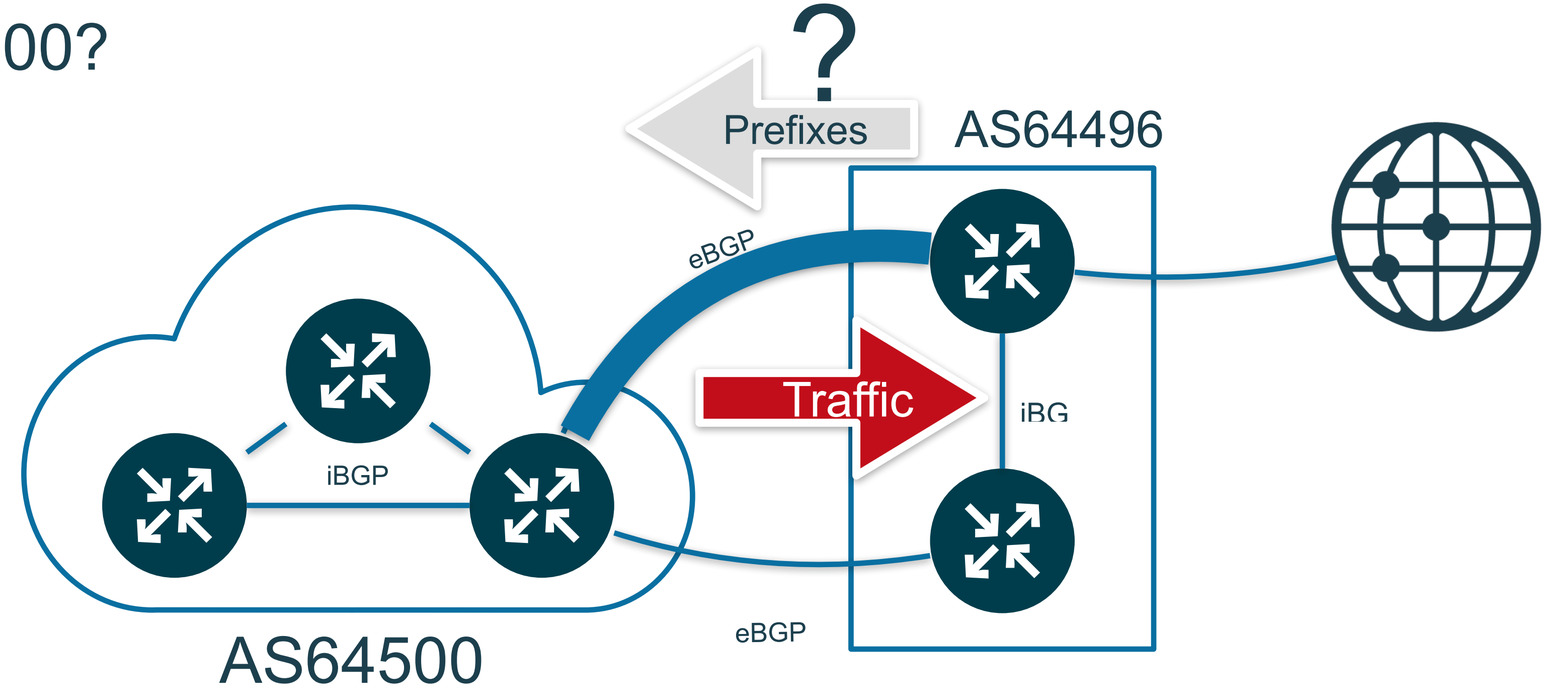
DE CIX

# Consider the following network



# *Consider the following network*

- There are two circuits
- AS64496 wants one of them preferred
- **How to tell** AS64500?



# BGP Route Selection Algorithm:

## How to tell your neighbor where you prefer traffic?

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# BGP Route Selection Algorithm: MED




- MED = **M**ulti-**E**xit **D**iscriminator
- Only compared if next-hop AS is the same
- 32bit value (0..4294967294)
- Lower wins
- Optional (does not have to be there)
- A missing MED can be treated as "best" (=0, default) or "worst" (=4294967294)
- Option "always-compare-med" **- only if you know what you are doing**!
- You can override any MED you receive

# Experiment: best path selection



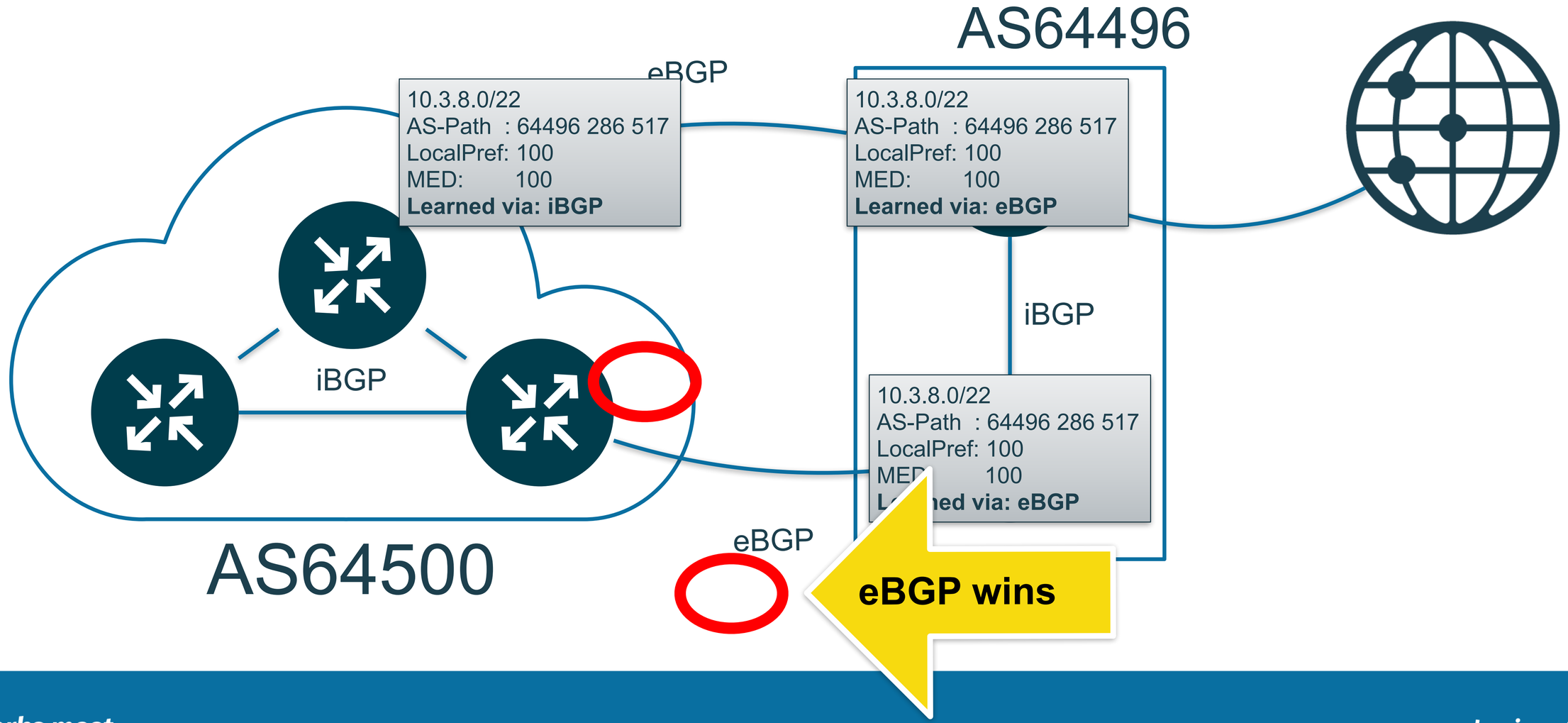
Experiment 3.04a: MED (same first AS)
Experiment 3.04b:MED (different first AS)

# BGP Route Selection : Hot Potato Rules

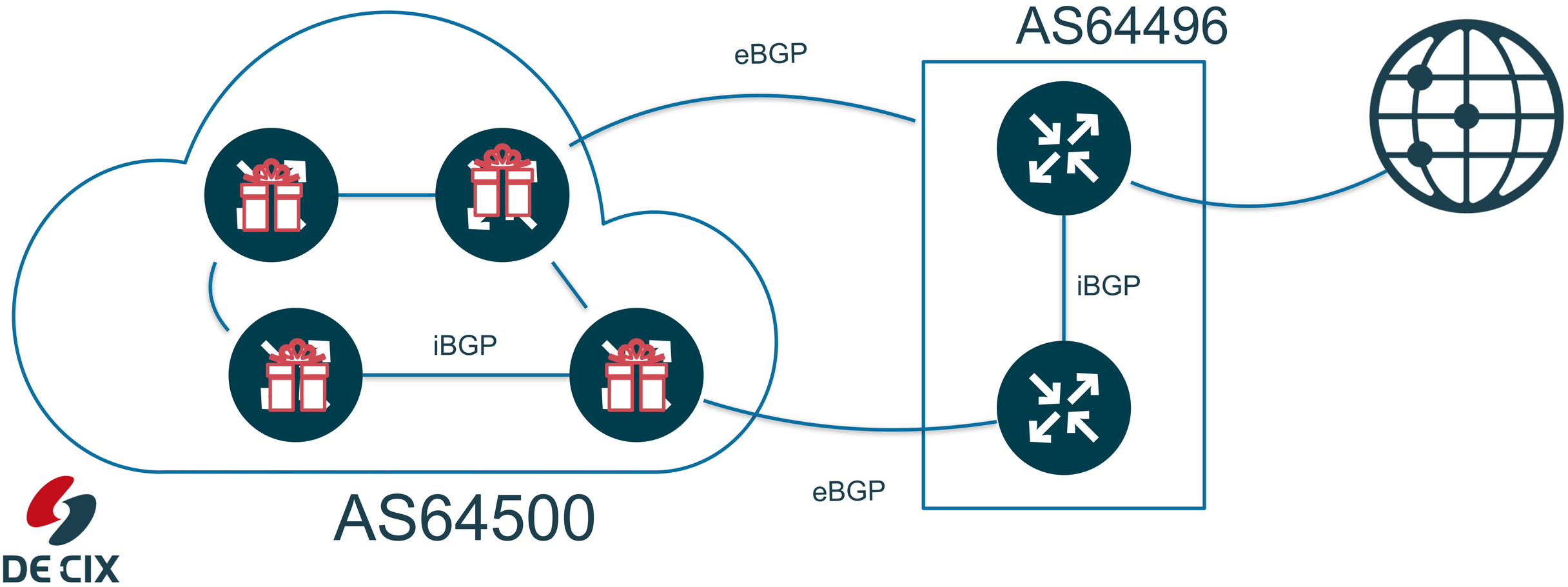
| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |



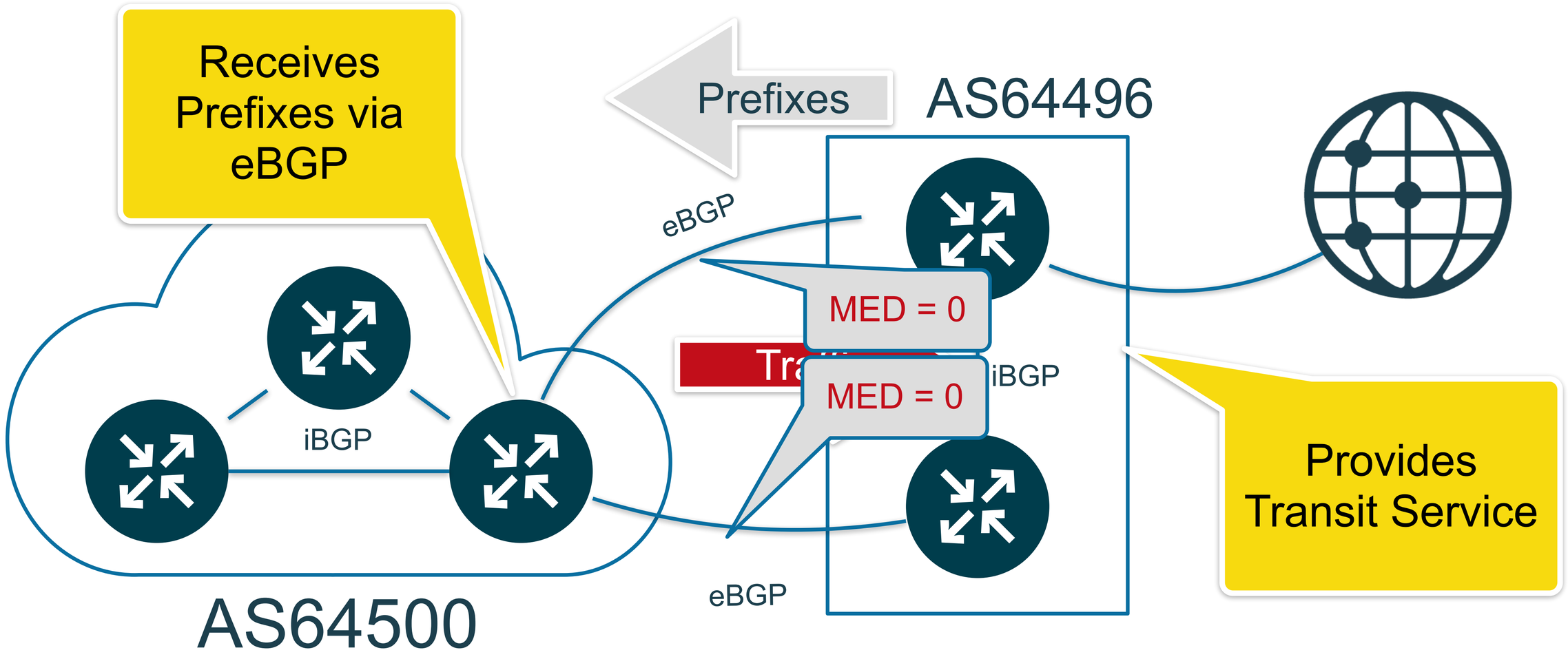
# BGP Route Selection : eBGP wins



# BGP Route Selection : nearest exit wins



# *Let's go back to our sample network*

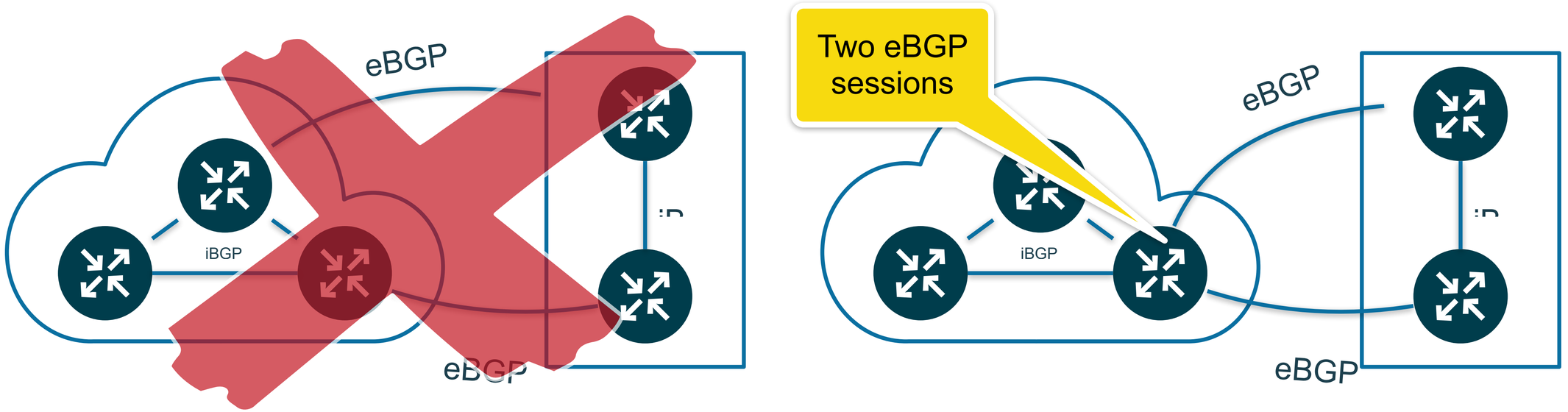


# BGP Route Selection : Age / Stability

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | | |
| 9 | | | |
| 10 | | | |

# BGP Route Selection : Age / Stability

- Exact phrasing is (Cisco):
  "When both paths are external, prefer the path that was received first"
- So this applies only if a router has two (or more) eBGP sessions
- Which happens quite often when connecting to Internet Exchanges



# Experiment: best path selection



Experiment 3.05: older wins
Experiment 3.08: own AS
Experiment 3.09: more specific

# BGP Route Selection : Last Resort

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | Age of route | older wins |
| 9 | | | |
| 10 | | | |

# *BGP Route Selection : Last Resort*

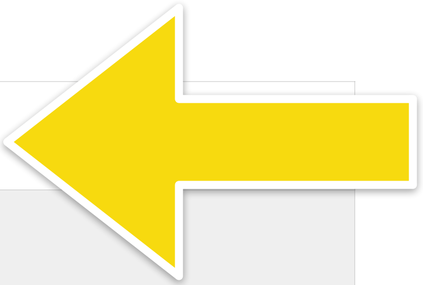
- Router ID: lower wins
- Neighbor IP: lower wins
- Rules of last resort
- ...because at the end one and only one best path has to be selected
- Usually path selection stops before it gets to these two rules....




| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | Age of route | older wins |
| 9 | | Router ID | lower wins |
| 10 | | Neighbor IP | lower wins |

# BGP Route Selection : Summary

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | Age of route | older wins |
| 9 | | Router ID | lower wins |
| 10 | | Neighbor IP | lower wins |



# A word about....

"Weight"



DE CIX

# *"Weight"*



- "Weight" is not a BGP attribute, it is a Cisco invention
- This is why it was not mentioned - I used to skip it completely
- But there are always questions about it
- So to be clear: **Do not use it**
  - Weight overrides everything, even Local Preference
  - It is only local to the router where it is applied
  - *It is the best way to **routing loops**!*

Weight - higher wins

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | Age of route | older wins |
| 9 | | Router ID | lower wins |
| 10 | | Neighbor IP | lower wins |

# *Is that all?* No!

# BGP Multipath

- Some times you do not only want one best path
- Have multiple "equal good" paths for load sharing
  - This can be enabled with the "maximum-path" config option
- To be considered as "equal good", a candidate must have the following equal to the best path:
  - local preference
  - as-path (can be relaxed)
  - origin
  - MED
  - neighbor AS (can be switched off)

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path Length | shorter wins |
| 4 | | Origin Type | IGP over EGP over Incomplete |
| 5 | | MED | lower wins |
| 6 | | eBGP, iBGP | eBGP wins |
| 7 | | Exit | nearest wins |
| 8 | | Age of route | older wins |
| 9 | | Router ID | lower wins |
| 10 | | Neighbor IP | lower wins |

# Open Questions?

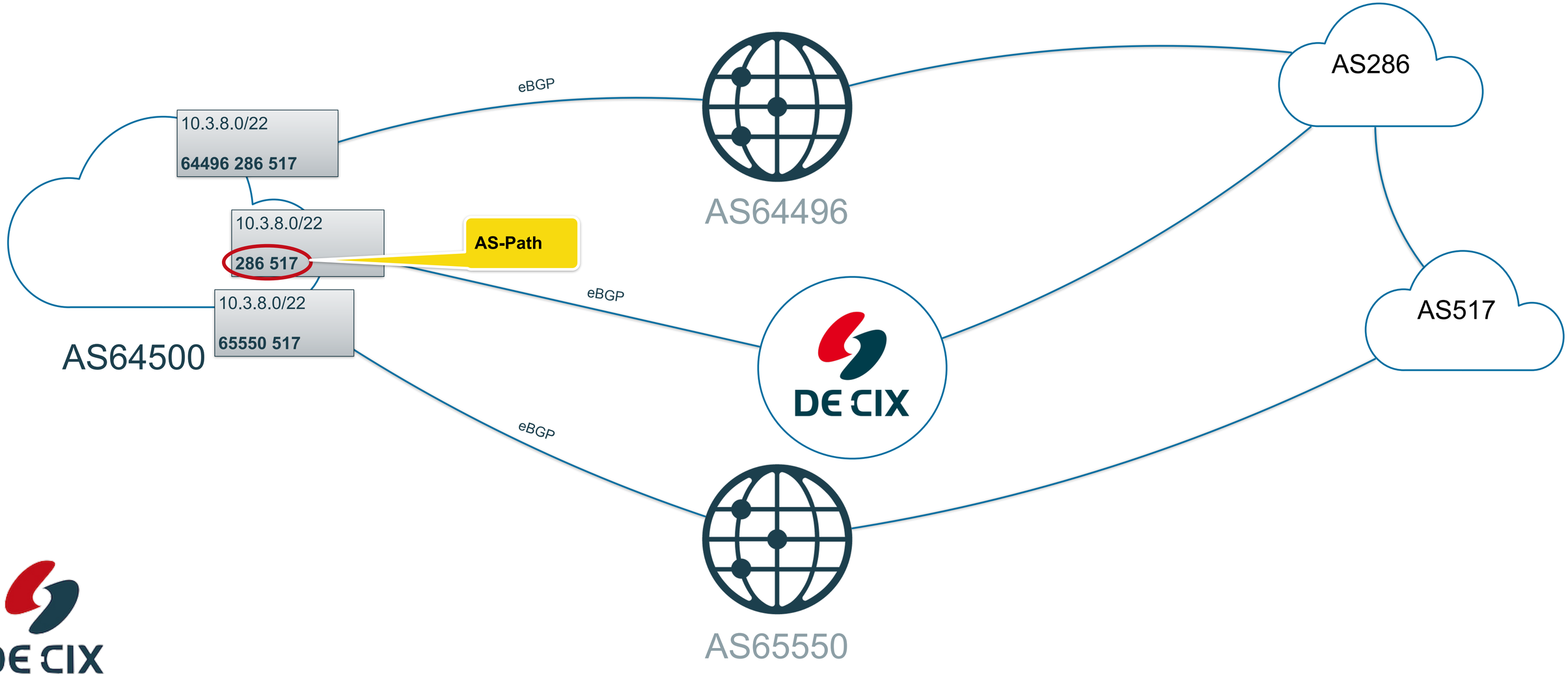
# *BGP - Best Path Selection*

## Beyond LocalPref and AS-Path Length

**Wolfgang Tremmel**
wolfgang.tremmel@de-cix.net



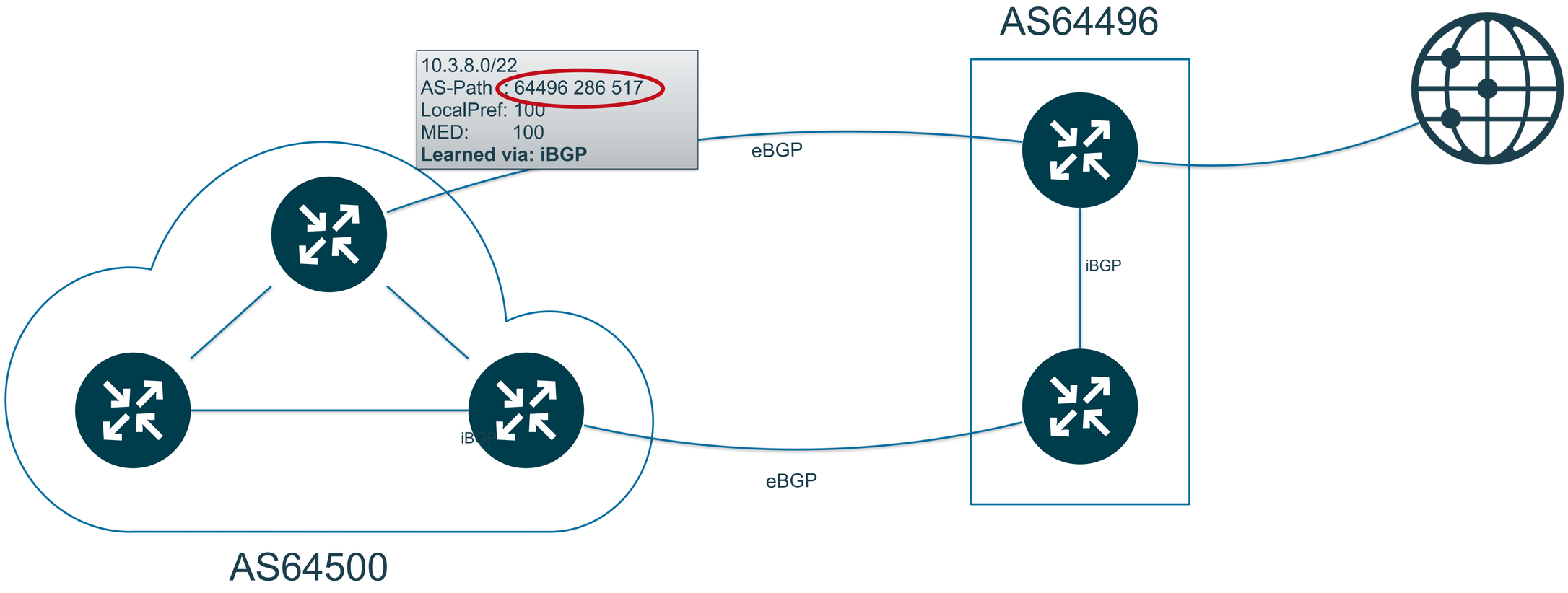
# We learned about BGP announcing prefixes




DE CIX

# ...and about BGP attributes

## Attributes of prefix announcements




DE CIX

# Attributes of BGP prefixes

## Mandatory, optional, (non-)transitive

10.3.8.0/22
AS-Path : 64496 286 517
LocalPref: 100
MED: 100
**Learned via: iBGP**

- **Mandatory** attributes: have to be there
  - Example: AS-Path
- **Optional** attributes: are, well, optional
  - Example: MED
- **Transitive** attributes:
  - are kept on the prefix and are forwarded via eBGP
- **Non-Transitive** attributes
  - are added to the prefix and not forwarded by the receiver

DE CIX

# Some BGP attributes

## Examples

- **AS-Path**: mandatory
- **Origin**: mandatory
- **Next-Hop**: mandatory
- **MED**: optional, non-transitive
- **Local Preference**: mandatory for iBGP, not sent via eBGP

- *IANA* keeps track of attribute codes
  - Currently there are more than 40 attributes registered

DE CIX

Introducing:

# BGP Communities

DE CIX

# BGP Communities

## A transitive, optional BGP attribute

- Transitive: Once attached, it stays until removed
- Optional: It does not have to be there
- "BGP Communities are like stickers on a suitcase"




Source: https://openclipart.org/detail/273877/suitcase-with-stickers-no-trademarks-remix

DE CIX

# "Original" BGP Communities

## First defined in 1996




Source: https://openclipart.org/detail/273877/suitcase-with-stickers-no-trademarks-remix

- Definition:
  *"A community is a group of destinations which share some common property"*
- Introduced in RFC1997 in year 1996
- An original BGP community is expressed by a 32Bit number
- High 16 bits are the AS defining the low 16 bits
  - Notation: "6695:1000", "517:286"
- You can attach as many BGP communities as you like (within reason)
  - BGP max message size ist 4096 bytes

DE CIX

# What are BGP communities useful for?

## Information!

**198.51.100.0/24** **80.81.192.15** **from 80.81.192.15**

**Path: 1301 286 517**

**Origin IGP, metric 0, localpref 100, valid, external**






DE CIX

DE CIX

# Informational BGP Communities

## Add information to prefixes

198.51.100.0/24 80.81.192.15 from 80.81.192.15

Path: 1301 286 517

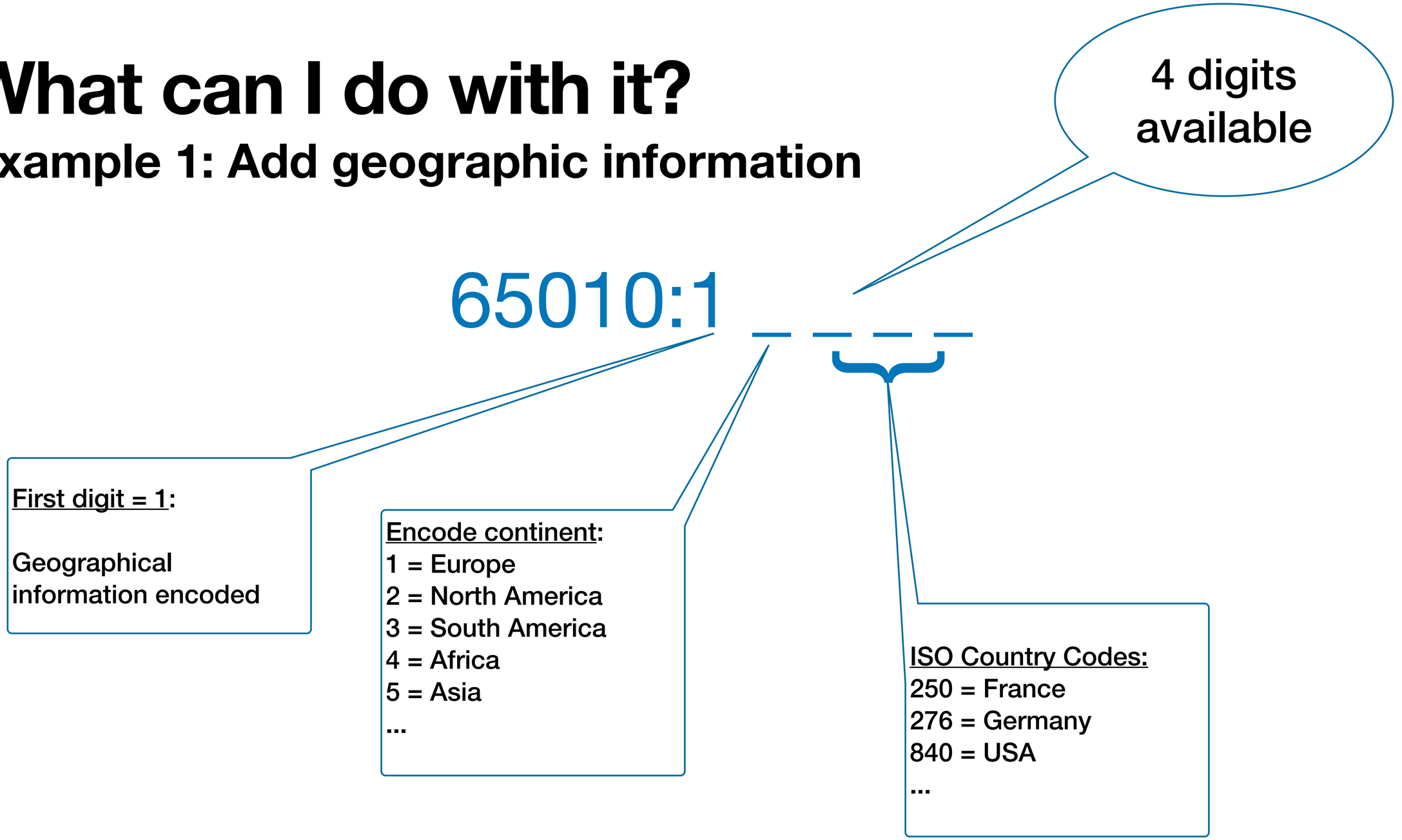
Origin IGP, metric 0, localpref 100, valid, external

Received from: Customer Peering Upstream

DE CIX

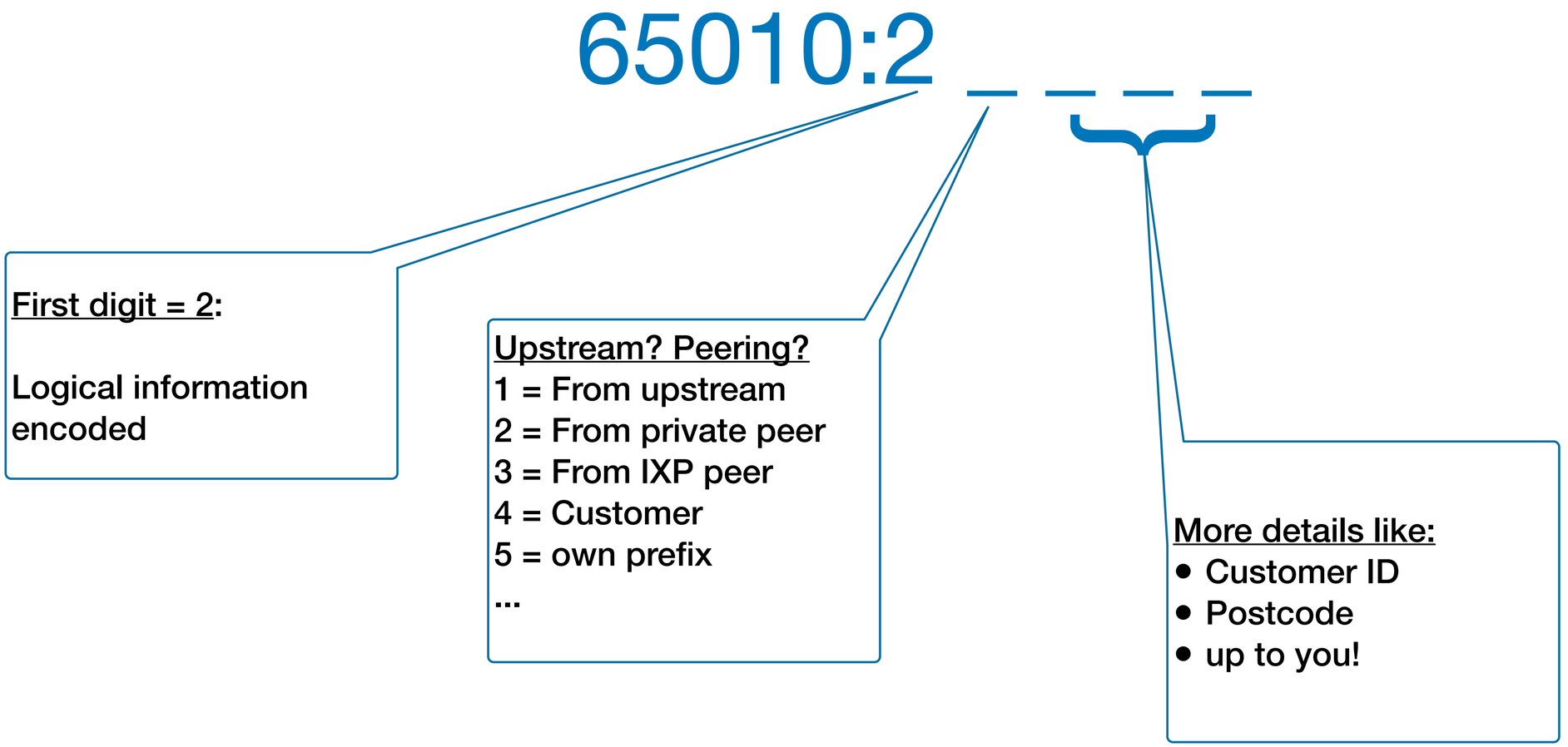
# What can I do with it?

## Example 1: Add geographic information



# What can I do with it?

## Example 2: Add logical information



# What are BGP communities useful for?

## Action!

**198.51.100.0/24** 80.81.192.15 from 80.81.192.15

Path: 1301 286 517

Origin IGP, metric 0, localpref 100, valid, external

Announce this prefix to: 


DE CIX

# BGP Action Communities

## "Well known" = predefined BGP Communities



- A number of BGP communities are pre-defined by RFCs
- NO-EXPORT
  - Do not send the prefix to eBGP neighbors (other ASes)
- NO-ADVERTISE
  - Do not send the prefix to anyone (not even internal via iBGP)
- GRACEFUL-SHUTDOWN
  - Set Local-Preference of so tagged prefixes to a low value
- BLACKHOLE
  - Sink all traffic to prefixes tagged with this communitiy
  - Most commonly used with host routes (/32 or /128)
  - Implies NO-EXPORT

DE CIX

# Do you have an 32bit AS-Number?

## No luck with original BGP Communities

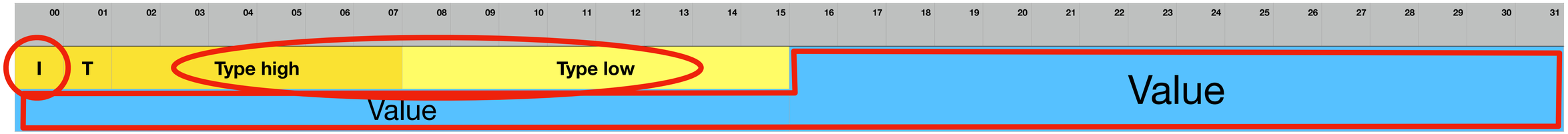
65010:12345

- Two 16-bit numbers
- No way to encode a 32-bit AS number and something else
  - RFC4360 - Extended BGP Communities
- **Extended BGP Communities** - Lots of new features
  - In total 64 bit long
  - Introducing a "type" field
  - Flag for private/public community
  - Flag for transitive  / non-transitive

DE CIX

# Extended BGP Communities

## Usefulness limited




- I: Type is IANA assigned or private
- T: Transitive or not. T=0: remove before forwarding to another AS.
- Type: IANA assigned or experimental. For list of types see RFC
- Value: 48 bits in total
  - 32-bit AS and 16-bit value ✓
  - 16-bit AS and 32-bit value ✓
  - 32-bit AS and 32-bit value ✗

DE CIX

# Extended BGP Communities

## Use cases

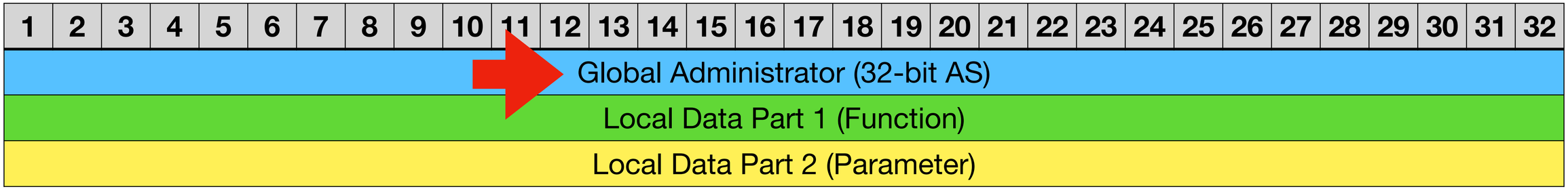
- Notation:
  - Similar to original BGP Communities: RT:6500000:1234
- Disadvantages
  - Only 48 bits in total
  - Only one 32 bit value is possible (and one 16 bit value)
  - RT, RO and other types confusing to many operators
- Where are they really needed?
  - Add information to EVPN BGP announcements
- Conclusion:
  - For peering / routing we need something different
  - It took the IETF a while to realize that (11 years)

DE CIX

# Introducing: Large BGP Communities

## Finally something useful

- Very simple - three 32-bit values
- Global Administrator:
  - an AS number (in 32-bit notation)
  - Defined the meaning of the other two fields
  - May have published that meaning
- Local Data
  - Can be seen as "just two 32-bit numbers"
  - Or as "Function" + "Parameter"



# Large BGP Communities

## Notation and definition

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Global Administrator (32-bit AS) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Local Data Part 1 (Function) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Local Data Part 2 (Parameter) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- Notation:
  - Similar to Original BGP Communities: **65010:100:65000042**
- Defined in two RFCs:
  - RFC8092: BGP Large Community Attribute
  - RFC8195: Use of BGP Large Communities
- A dedicated website exists: http://largebgpcommunities.net/

DE CIX

# Recommendation

If you are starting fresh with BGP communities

only use **LARGE** communities

DE CIX

# Experiment: Working with BGP Communities

## Experiment 04



# BGP Action Communities

## Easy with LARGE BGP Communities

- Reminder: You now have two 32-bit numbers
  - Use your AS number as Global Administrator
- Ideas: "Announce this prefix..."
    - to upstream providers
    - only in Europe
    - not in North America
- You might also allow your customers to send you "actions"
  - Do not only filter on communities
  - Make sure a prefix is also "allowed" to carry this community (this is hard, I know)
- Encoding: Up to you! There are enough possibilities

DE CIX



DE CIX

https://de-cix.net/academy

DE CIX
DE-CIX Management GmbH | Lindleystr. 12 | 60314 Frankfurt | Germany
Phone + 49 69 1730 902 0 | sales@de-cix.net | www.de-cix.net

# Links and further reading



# Links visited

- RFCs

  RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)

  - RFC4632: Classless Inter-domain routing (CIDR)
  - RFC4291: IPv6 addressing architecture
  - RFC827: Exterior Gateway Architecture (EGP) (historical, obsolete)
  - RFC1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)
  - RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space

- AS Numbers
  - Giving AS numbers to the RIRs: iana.org
  - Requesting an AS number, links for:
    - ARIN
    - Lacnic
    - APNIC
    - RIPE NCC
    - Afrinic

# Links visited

- RFCs
  RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)
  - RFC4271 - A Border Gateway Protocol 4 (BGP-4)
    - see 5.1.5 for a definition of *Local Preference*
    - see 9.1 for the BGP best path selection algorithm

- BGP Best Path Selection by vendor
  - Cisco
  - Juniper
  - Mikrotik
  - Nokia
  - BIRD
  - FRRouting

- If you peering at any Internet Exchange - please use PeeringDB

# Links and further reading

- Definition of terms (all from RFC4271):
  - *Next Hop* is defined in Section 5.1.3
  - *AS Path* is defined in Section 5.1.2
  - *Local Preference:* Section 5.1.5
  - Origin: Section 5.1.1
  - *Multi Exit Discriminator (MED):* Section 5.1.4
- Best Path Selection process: Section 9.1
- BGP Route Selection Algorithm by vendor:
  - Cisco
  - Juniper
  - Mikrotik
  - Nokia
  - BIRD
  - FRRouting
  - Quagga

DE CIX

# BGP Best Path Selection Algorithm

**Bold items** were covered in this webinar

| | | | |
|---|---|---|---|
| 1 | | NextHop reachable? | Continue if "yes" |
| 2 | | Local Preference | higher wins |
| 3 | | AS Path | shorter wins |
| **4** | | **Origin Type** | **IGP over EGP over Incomplete** |
| **5** | | **MED** | **lower wins** |
| **6** | | **eBGP, iBGP** | **eBGP wins** |
| **7** | | **Exit** | **nearest wins** |
| **8** | | **Age of route** | **older wins** |
| **9** | | **Router ID** | **lower wins** |
| **10** | | **Neighbor IP** | **lower wins** |

# BGP Best Path Selection Algorithm

**Local Preference** is...

- → a 32bit integer value (0-4294967295)
- → Propagated via iBGP inside an Autonomous System
- → Usually set using rules when receiving prefixes
  - According to your routing policy
- → Typical values
  - 10000 (high value) for customer prefixes
  - 1000 (medium value) for prefixes received via peering
  - 100 (low value) for prefixes received via upstream
- → Rules to adjust local preference can be as complex as your router software allows it to be.

**AS Path** is...

- → an ordered list of AS numbers...
- → ...with the originator AS at the rightmost side
- → automatically built when prefixes are sent via eBGP
- → length of the path is used for selection (shorter wins)

# BGP Best Path Selection Algorithm

**Origin Type** is...

- → a historic, but mandatory attribute
- → set by originator AS and forwarded unchanged
- → can have the values (in order of preference):
  - IGP - prefix was originated via a network statement
  - EGP - prefix was learned from Exterior Gateway Protocol (RFC904, historic)
  - incomplete - prefix was learned by another protocol

**Multi Exit Discriminator (MED)** is...

- → a 32Bit value, lower wins
- → optional, if it is not there it's either treated as zero (best) or as 2^32-1 (worst)
- → non-transitive (set by an eBGP speaker and only sent to the next-hop AS)
- → usually set using rules when sending prefixes (according to the sender's routing policy)
- → only compared between eBGP speakers if next-hop AS is the same

**Router ID** is...

- → also called **BGP Identifier**
- → a 4 byte, unsigned integer (mostly it's the IPv4 loopback address of a router)
- → unique within one AS
- → set at startup and stays unchanged
- → the same for all BGP sessions

**Neighbor IP** is...

- → the last tie-breaker in the BGP Best Path Selection
- → the IP address of the eBGP speaker a prefix was learned from

# Links and further readings

- BGP attribute types:
  - Registering new types: RFC2042
  - Published in BGP Parameters database at IANA
- Well-known communities:
  - Standard: see IANA website
  - Extended: see IANA website
- Provider examples:
  - DE-CIX Communities:
    - Action: https://docs.de-cix.net/article/fh5syii5dk-de-cix-globepeer-route-server-action-bgp-communities
    - Information: https://docs.de-cix.net/article/oqsxcr457r-de-cix-globepeer-route-server-informational-bgp-communities
  - DE-CIX Routeserver Guides: https://docs.de-cix.net/article/traw7kch04-de-cix-frankfurt-globepeer-route-server-guide
  - KPN - AS286 - Community page: https://as286.net/AS286-communities.html
  - NTT - AS2914 - Community page: https://www.us.ntt.net/support/policy/routing.cfm
- Lab: Download the DE-CIX Academy lab here: https://gitlab.com/de-cix-public/team-academy/bgp/BGPLab



# BGP Communities

## Facts and RFCs

| | Original Communities | Extended Communities | Large Communities |
|---|---|---|---|
| **Defined in** | RFC1997 | RFC4360 | RFC8092 |
| **Published** | August 1996 | February 2006 | February 2017 |
| **Additional RFCs** | RFC1998, RFC3765, RFC7999 | RFC4384, RFC5668, RFC5701, RFC7153, RFC8097 | RFC8195 also: http://largebgpcommunities.net |
| **Size** | 32bit | 64bit | 96bit |
| **Commonly used** | 16-bit AS : 16-bit Value | Type : 32-bit AS : 16-bit Value | 32-bit AS: 32-bit Value: 32-bit Value |
| **Example** | 6695:65010 | RT:6695:20250707 | 6695:196610:19660520 |
| **Intention** | | Replace original standard | Supplement original standard |

DE CIX

# License


```
Copyright 2026 DE-CIX Management GmbH

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```