DENOG 10
GERMAN NETWORK OPERATORS GROUP

SpaceNet
Internet Business Produkte

BGP Communities 101

Gert Döring, SpaceNet AG

DENOG 10, 21./22. November 2018

BGP Communities

* who? (I'll spare you the SpaceNet marketing slides)
* what?
* why?
* how?

What are BGP Communities?

- 32 bit numbers that can be „attached“ to a prefix in BGP
- usually written in 16:16 bit form
- to avoid clashes, upper 16 bit value usually = AS number
- 5539:nnn = „this is relevant in context of AS5539“
- very few numbers have built-in significance (NO_EXPORT)
- used as „BGP remote control“
  - router A tags prefix „set community XX“
  - BGP transports prefix plus community attributes
  - decision at router B „if (community = YY) then <do something>“
  - announce, prepend, alter attributes (med, next-hop) …

first impressions…

```
RP/0/RSP0/CPU0:Cisco#sh ip b 193.31.7.0
…
 5404 8481
    80.81.193.127 from 80.81.192.157 (80.81.192.157)
      Origin IGP, metric 1, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 0, version 324988009
      Community: 5539:100 5539:101 65101:1001 65102:1000 65103:276 65104:150
…
 1273 6830 5404 8481
    62.208.255.157 from 62.208.255.157 (195.2.1.91)
      Origin IGP, metric 1, localpref 100, valid, external, group-best
      Received Path ID 0, Local Path ID 0, version 0
      Community: 1273:22000 5539:200 6830:13000 6830:23001 6830:34319
```

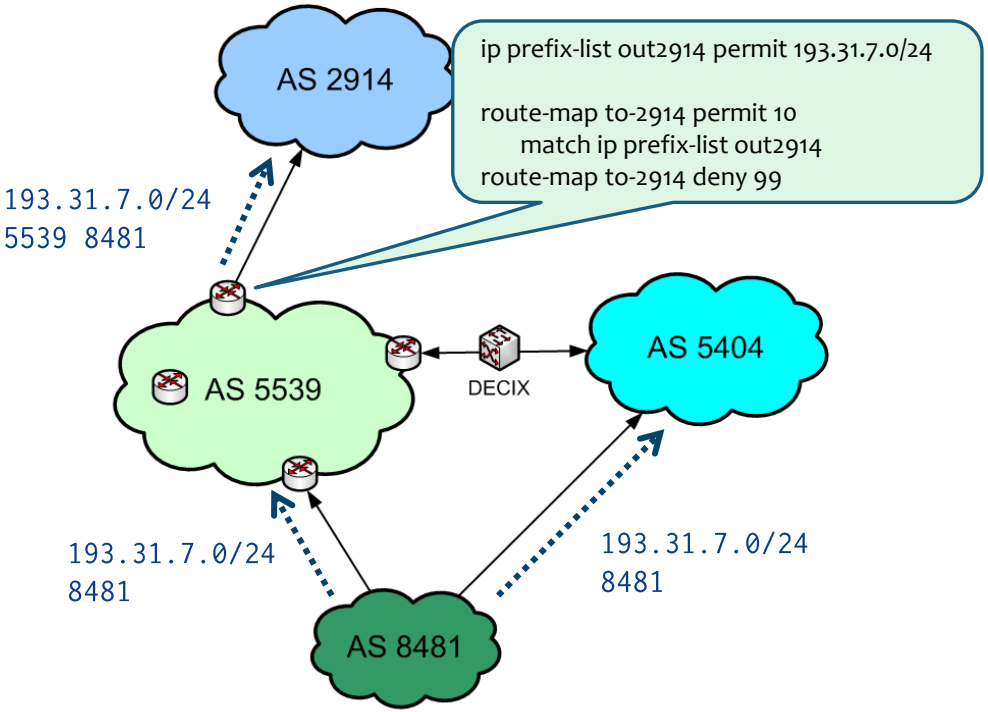
do I really need this?

* No
* Most things you would do with communities can be done in other ways, like
  * manually configuring lots of routers for simple changes
  * using your fancy puppet/salt automatization layer to auto-reconfigure half your network if you add a new customer
  * spending lots of hours on config, research, or tool writing
* strictly speaking, you do not *need* BGP communities
* … but they are a very nice way to get things done

do I really need this?

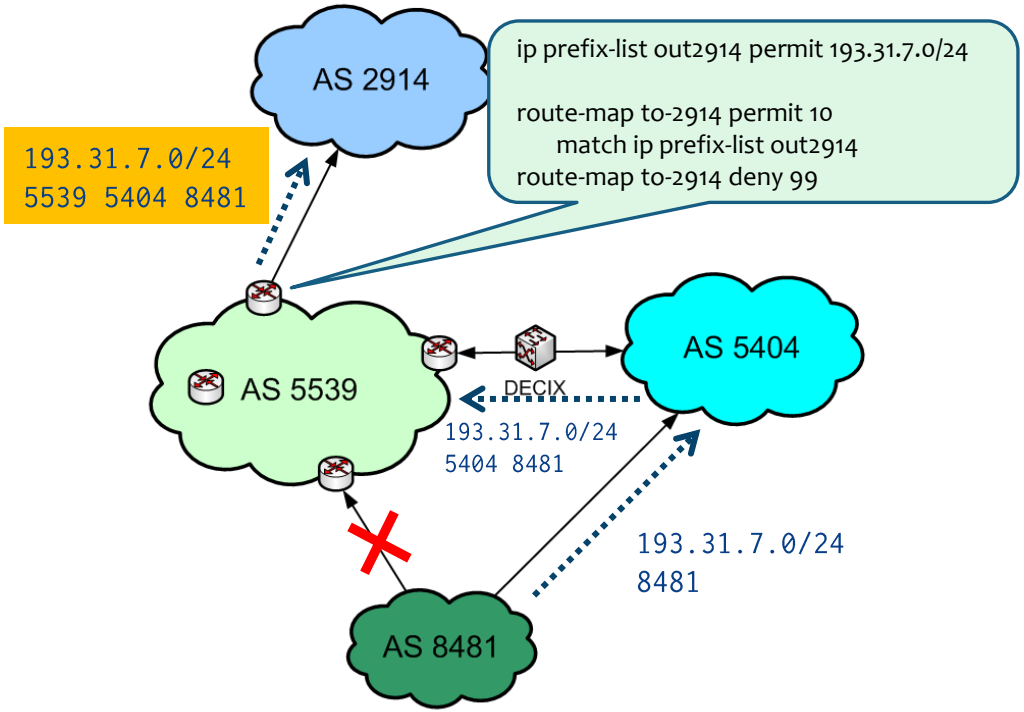
* No
* Most thin communities can be d
  * manu nges
  * using r to auto-reco customer
  * spending lots o ool writing
* strictly speaking, you do no unities
* … but they are a very nice way to get s done

Example 1: prefix leaking



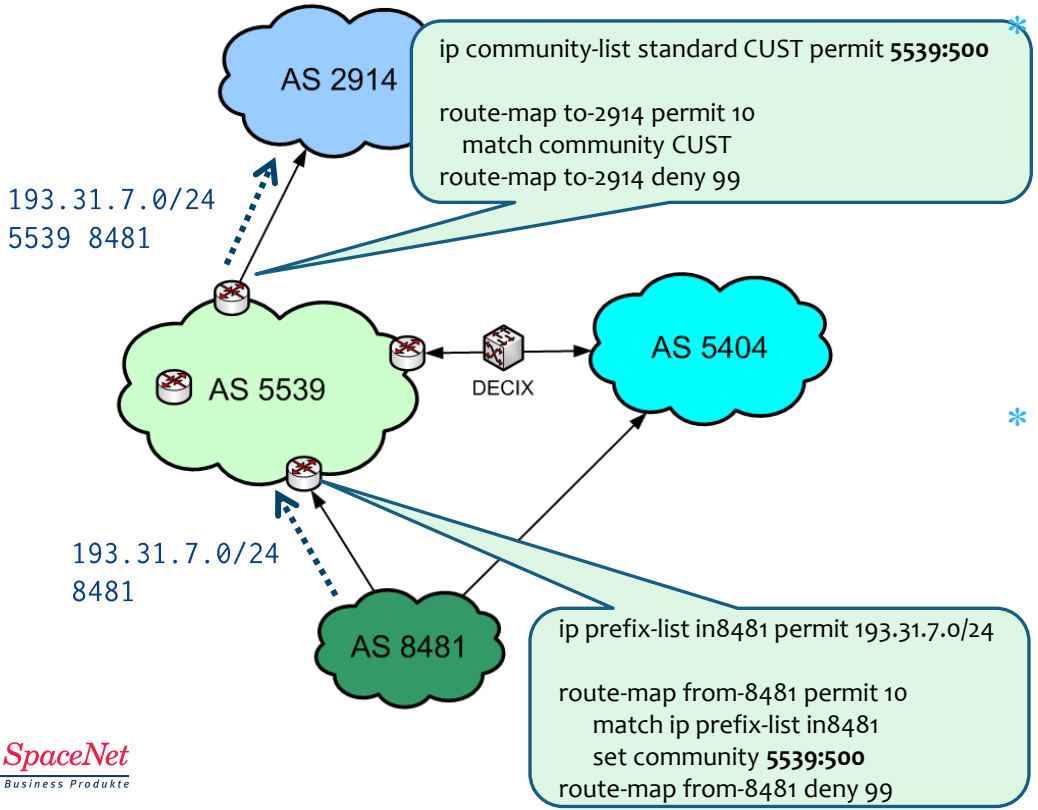
* AS8481 is customer of AS5539
* AS8481 is also customer of AS5404, who peers with AS5539
* AS5539 has very strict export prefix filters towards AS2914
* prefix announced to AS2914 with correct AS-path „5539 8481“

Example 1: prefix leaking



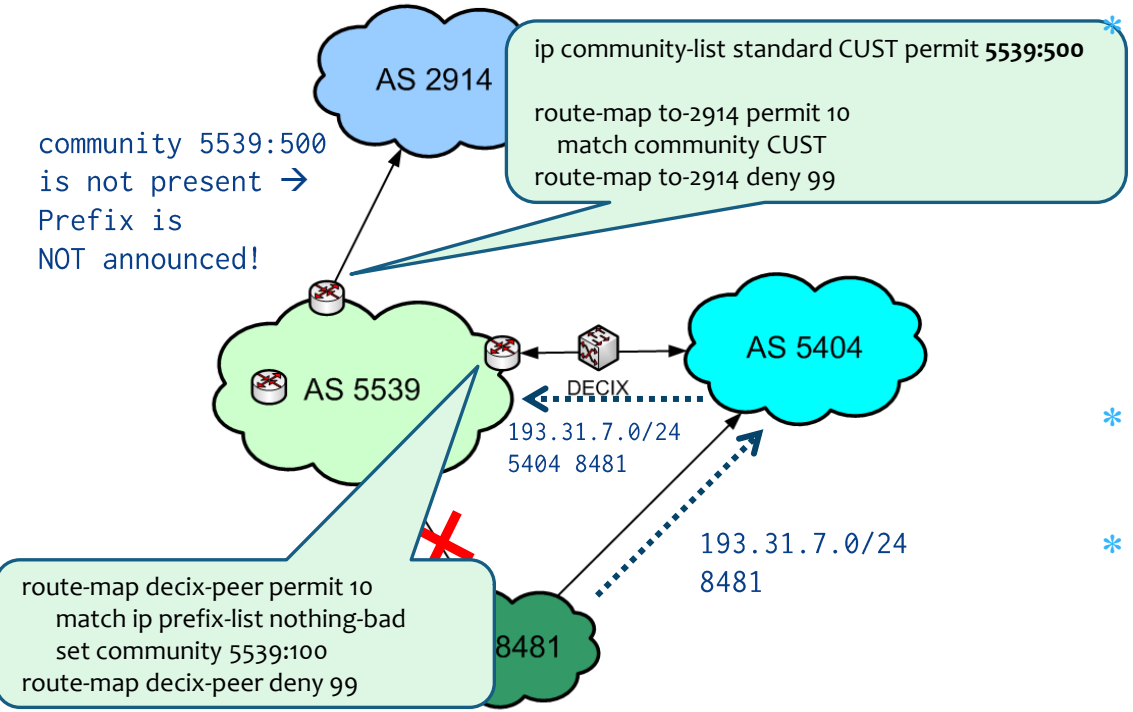
* AS8481 is customer of AS5539
* AS8481 is also customer of AS5404, who peers with AS5539
* AS5539 has very strict export prefix filters towards AS2914
* what if 8481-5539 link fails?
* prefix filters will leak peer routes to upstream!
* could be fixed, of course, by adding strict AS-path filters...
  * do not forget prepends
  * update for each new customer!

Example 1: with BGP communities



* all prefixes accepted into AS5539 get tagged with a community value:
  * 5539:500 = customer
  * 5539:100 = peering (decix)
  * 5539:250 = upstream (NTT)
* on export, the „customer“ community must be present

Example 1: with BGP communities

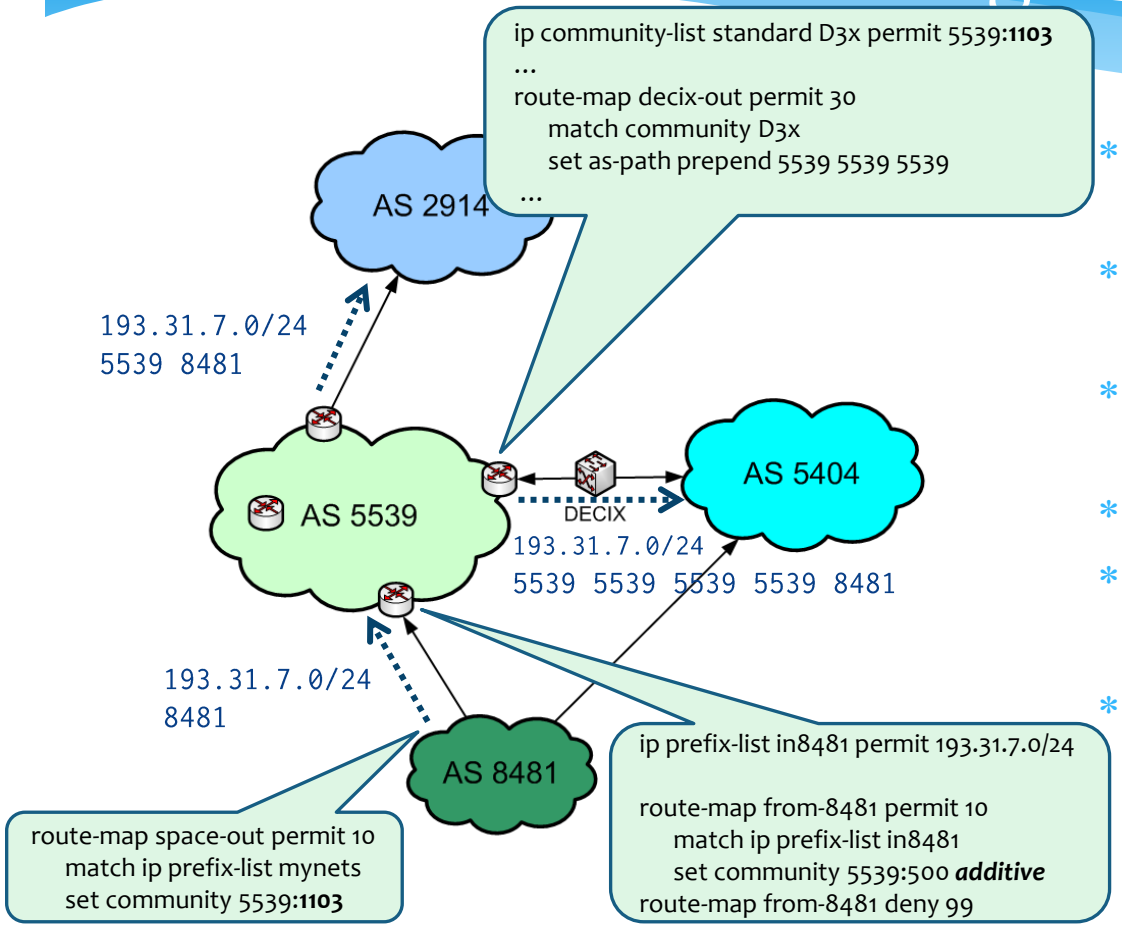


* all prefixes accepted into AS5539 get tagged with a community value:
  * 5539:500 = customer
  * 5539:100 = peering (decix)
  * 5539:250 = upstream (NTT)
* on export, the „customer“ community must be present
* if a prefix has no „this is my customer“ community, it will never leak(!)
* fails *safely*

Nice! More of this!

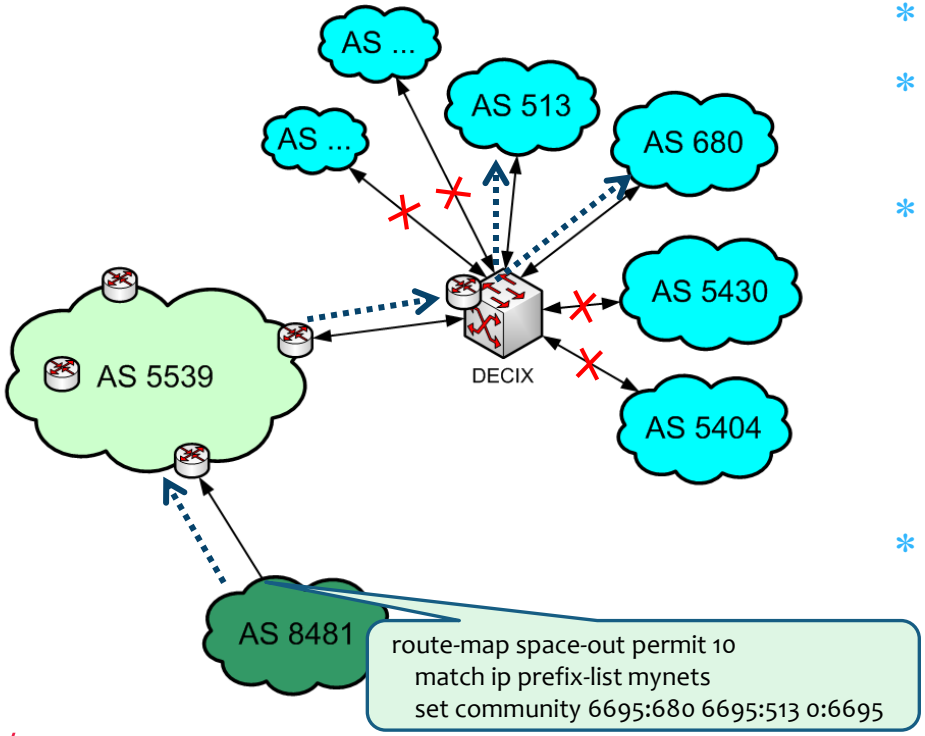
* Fine grained export control
  * do not export to AS 2914
  * prepend 3x 5539 towards all peers @DECIX
  * …
* Tag prefix origin – where did your network learn this prefix?
  * 5539:100 route learned from peering @DECIX Frankfurt
  * 5539:250 route learned from upstream AS2914
  * …
* BGP-based black holing (DDoS protection)
  * 5539:3000 „please null-route traffic to this IP address“
  * …

Fine-grained export control



* BGP customers sometimes need finegrained control on route export
* „For DECIX traffic, I prefer to not use AS5539“
* look up „prepend to DECIX“ value in 5539‘s community documentation
* tag announcements with **5539:1103**
* 3x AS prepend to DECIX!
* *ingress* at AS8481: match on 5539:100 aka „route from DECIX“ and prepend (not shown here)

Fine-grained export control



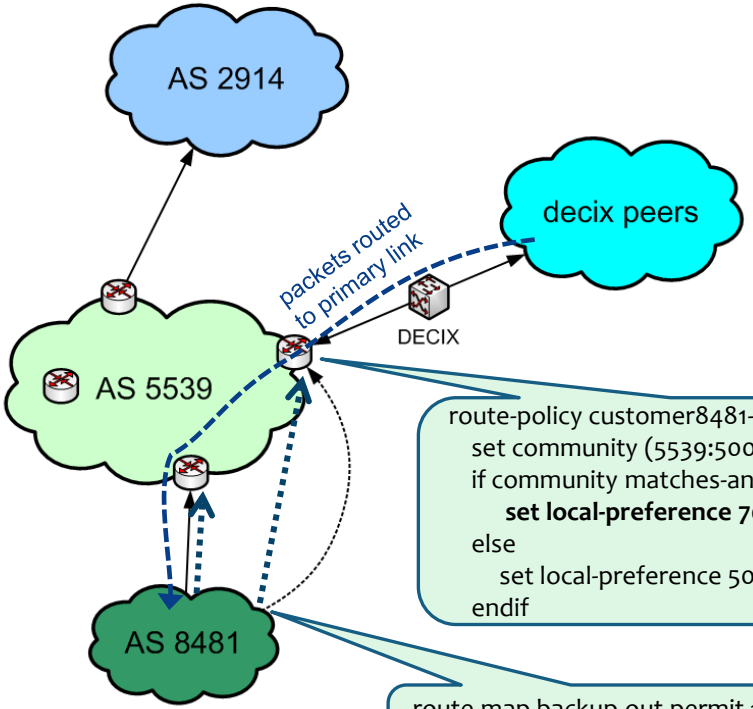
* DECIX FRA: 900+ peers
* reach them all in one go via DECIX route server
* announce to „only a subset“ via well-defined communities*
  * 0:5404 – not to AS 5404
  * 6695:680 – only to AS 680
* what about peers with 32bit AS numbers??

* https://www.de-cix.net/de/resources/route-server-guides/operational-bgp-communities

32bit is never enough

* 16:16 form with „AS#:xx“ does not work with 32bit AS#s
  * it does not even work well for all 16bit cases (0:5404)
* RFC4360 specified „*pfx*:16:32“ „extended communities“
  * complicated and still not large enough
* RFC8092, RFC8195 specify „large community“ standard
  * 96 bit „32:32:32“ format with no special semantics
  * use like: 6695:<what>:<peer_AS> with full 32bit AS support
* the underlying concepts are the same:
  * something *sets* the community („glues the number to the prefix“)
  * something else can *look* at the number, and possibly do things
* more details: http://largebgpcommunities.net/examples/

Fine-grained import control



* control local preference inside AS5539
  * default: 500
  * 5539:3070 -> local-pref 70
  * 5539:3130 -> local-pref 130
* control black holing
  * 5539:3000 -> blackhole
  * 5539:3001 -> blackhole remotely
  * (not shown here)

Numbering Scheme?

* Effectively there is no „gold standard“, every AS re-invents
* Keep numbers easy for humans to „understand“
  * 5539:1xx = peering (:100 = decix, :120 = INXS, :170 = ECIX, …)
  * 5539:2yy = upstream
  * 5539:500 = customer (-> „local pref 500“)
  * 5539:1xxz = „do something with peering xx“ (prepend, no-announce)
  * 5539:2yyz = „do something with upstream yy“ (prepend, …)
  * z = 0 / 1 / 3 -> no announce, prepend 1x, prepend 3x
* clearly document ranges and structure!
* use distinct ranges for „customers can set these“ (5539:aaaa) and „only your network can set these“ (5539:bbb)

Community Scrubbing

* why use distinct number ranges?
  * correctness of your BGP prefix announcements depends on correct communities set on prefix
  * what happens if someone sends you the „I am your customer“ community (5539:500) at a peering point?
  * Answer: you re-announce this prefix to all your other peers and upstreams – route leak again!

* Easy fix: reset (override) all community values on import
* More flexible fix: selectively remove (+set) all values that must not be controlled by remote end

Community Scrubbing: IOS

```
! inverted logic: “deny” = “do not delete” = “keep”.  “permit” = “do delete”
!
ip community-list 100 deny 5539:[0-9][0-9][0-9][0-9]$          ! keep 5539:<4-digit>
ip community-list 100 permit 5539:.*                           ! remove all other 5539:*
ip community-list 100 deny .*                                  ! keep everything else
!
ip prefix-list customer8481 permit 193.31.7.0/24
!
ip as-path access-list 72 permit ^(8481_)+$
!
route-map customer8481-in permit 10
    match ip address prefix-list customer8481
    match as-path 72
    set comm-list 100 delete
    set community 5539:500 additive
!
route-map customer8481-in deny 11
```

Community Scrubbing: IOS XR

```
prefix-set customer8481
    193.31.7.0/24
end-set
!
as-path-set ASP-customer8481
    ios-regex '^(8481_)+$‘
end-set
!
route-policy customer8481-in
    if not (destination in customer8481 and as-path in ASP-customer8481) then
       drop
    endif
!
!                                                    scrubbing: delete non-4-digit 5539:*, leave rest alone
    delete community in (5539:[0..999], 5539:[10000..65535])
    set community (5539:500) additive
end-policy
```

Example: IOS (customer in)

```
ip prefix-list customer8481-msp permit 193.31.7.0/24 le 32
ip prefix-list customer8481     permit 193.31.7.0/24
!
ip as-path access-list 71 permit ^(8481_)+$
!
ip community-list standard SPACE:rtbh permit 5539:3000
ip community-list standard SPACE:rtbh permit 65535:666
!
ip community-list standard SPACE:3070 permit 5539:3070
 […]
!
route-map customer8481-in permit 10
  match ip address prefix-list customer8481-msp
  match community SPACE:rtbh
  set community 5539:3000 no-export
  set ip next-hop 192.0.2.1
!
route-map customer8481-in permit 20
  match ip address prefix-list customer8481
  match as-path 71
  set comm-list 100 delete
  set community 5539:500 additive
  continue 100
!
```

```
…
!
route-map customer8481-in deny 21
!
route-map customer8481-in permit 100
  match community SPACE:3070
  set local-preference 70
!
route-map customer8481-in permit 110
  match community SPACE:3100
  set local-preference 100
!
route-map customer8481-in permit 120
  match community SPACE:3130
  set local-preference 130
!
route-map customer8481-in permit 200
  set local-preference 500
!
route-map customer8481-in deny 999
```

Example: IOS XR (decix out)

```
community-set BGP-BLACKHOLE
  5539:3001,
  65535:666
end-set
!
community-set SPACE:Customer
  5539:500
end-set
!
community-set DECIX:no
  5539:1100
end-set
!
community-set DECIX:1x
  5539:1101
end-set
!
community-set DECIX:500
  5539:1105
end-set
!
```

```
route-policy decix-peer-out
# filter out “no announce” routes
  if community matches-any DECIX:no or not
         (community matches-any SPACE:Customer) then
    drop
  endif
# translate internal to DECIX blackhole community
  if community matches-any BGP-BLACKHOLE then
    set community (65535:666) additive
    done
# prepend?
  elseif community matches-any DECIX:1x then
    prepend as-path 5539
    done
# […]
  elseif community matches-any DECIX:500 then
    set med 500
    done
  endif
# nothing special, just announce
  done
end-policy
```

SpaceNet
Internet Business Produkte

References

* http://largebgpcommunities.net/examples/ (config examples)
  * https://tools.ietf.org/html/rfc8092
  * https://tools.ietf.org/html/rfc8195 (inspirations!!)
* https://www.de-cix.net/de/resources/route-server-guides/operational-bgp-communities
* http://www.space.net/static/bgp_communities.html (AS5539)
* http://www.us.ntt.net/support/policy/routing.cfm (AS2914)
* https://www.de-cix.net/en/about-de-cix/academy/videos-and-webinars

* gert@space.net – questions welcome