BGP Optimierungen fuer Anycast

Atnog Meetup, 16.07.2019

Unicast

- 1 Host hat die Unicast IP Adresse konfiguriert
- Jeder Client weltweit landet beim gleichen Server

Anycast

- Viele Host haben die gleiche IP Adresse konfiguriert
- Clients landen bei irgendeinem dieser Server

Warum Anycast

- Lastverteilung – fast beliebige Skalierung in die Breite
- Kürzere RTT zwischen Client und Server
  - Dadurch schneller DNS Antworten
  - Aber nur, wenn das Anycastnetz so aufgebaut ist, dass Clients zu einem nahe gelegenen Server geroutet werden

Gutes Anycast

- Routing zu einem nahem Server

Schlechtes Anycast

- Routing zu einem weit entfernten Server

Anycast Routing

- So wie normales Routing im Internet → BGP
- Ein Router kann für ein Prefix mehrere Routen haben
  - Unicast: alles landet letztendlich beim gleichen Server
  - Anycast: unterschiedliche Routen können auf unterschiedlichen Servern landen

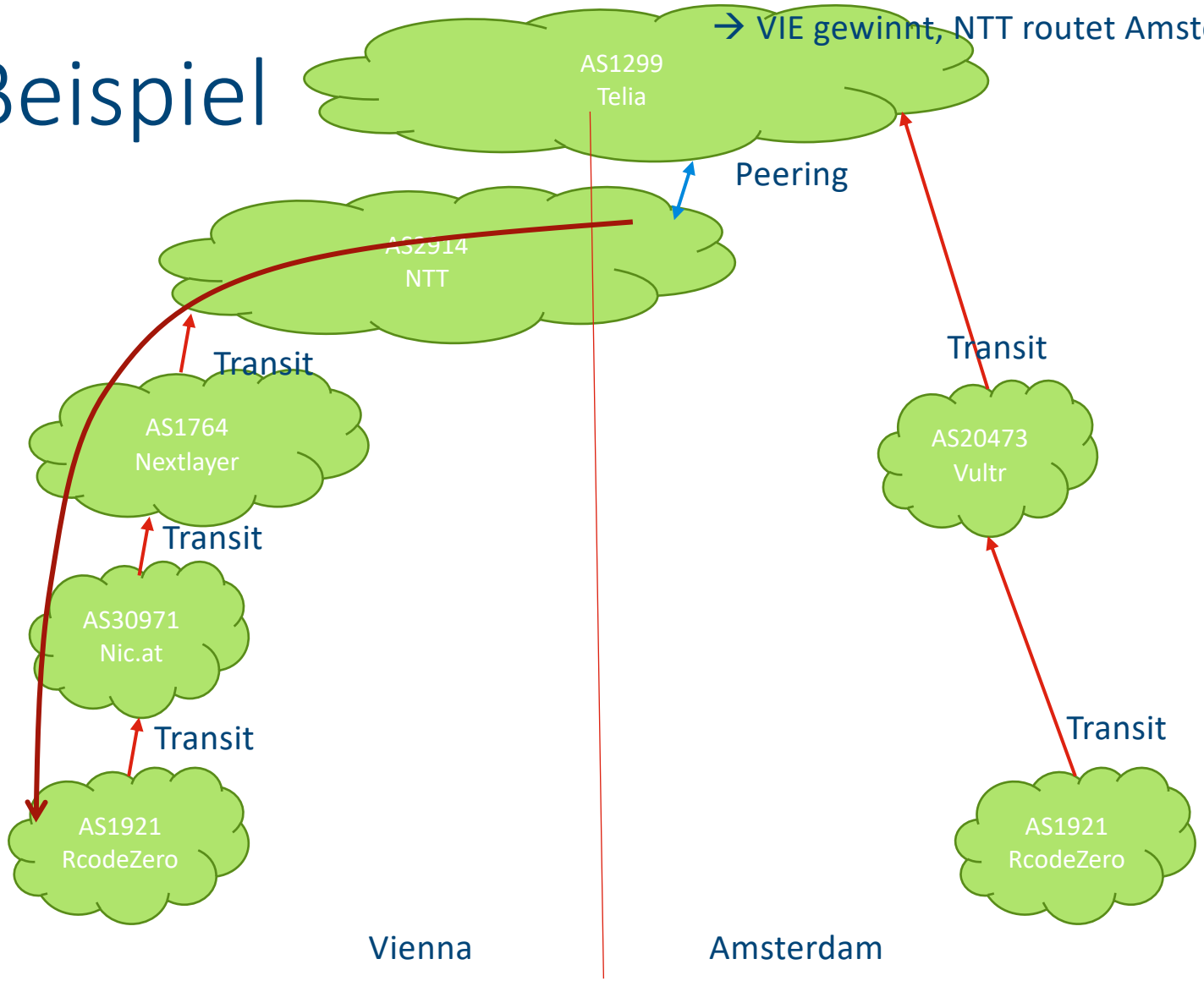
Internet Routing (zwischen ASen)

1. Best prefix match (more specific wins)
   - 192.174.68.0/24 ist besser als 192.174.0.0/16
2. Local Preference
   - $$$: die Geschäftsbeziehung wird abgebildet
     - Route kommt von einem Kunden +$$$: höchste LocalPref
     - Route kommt von einem Peer: mittlere LocalPref (mehrere Abstufungen)
     - Route kommt von einem Transit Provider -$$$: niedrigste LocalPref
3. AS-Pfad Länge
   - Je kürzer der AS-Pfad umso besser

- Interne Kosten
  - Wie weit ist es zum Interconnect Point?
- Route Stabilität
  - Eine ältere Route (stabiler) wird bevorzugt

Beispiel

NTT sieht 2 Routen für 192.174.68.0/24:
AMS: localPref=100, AS-Pfad len=3: 1299 20473 1921
VIE: localPref=120, AS-Pfad len=3: 1764 30971 1921
→ VIE gewinnt, NTT routet Amsterdam nach VIE



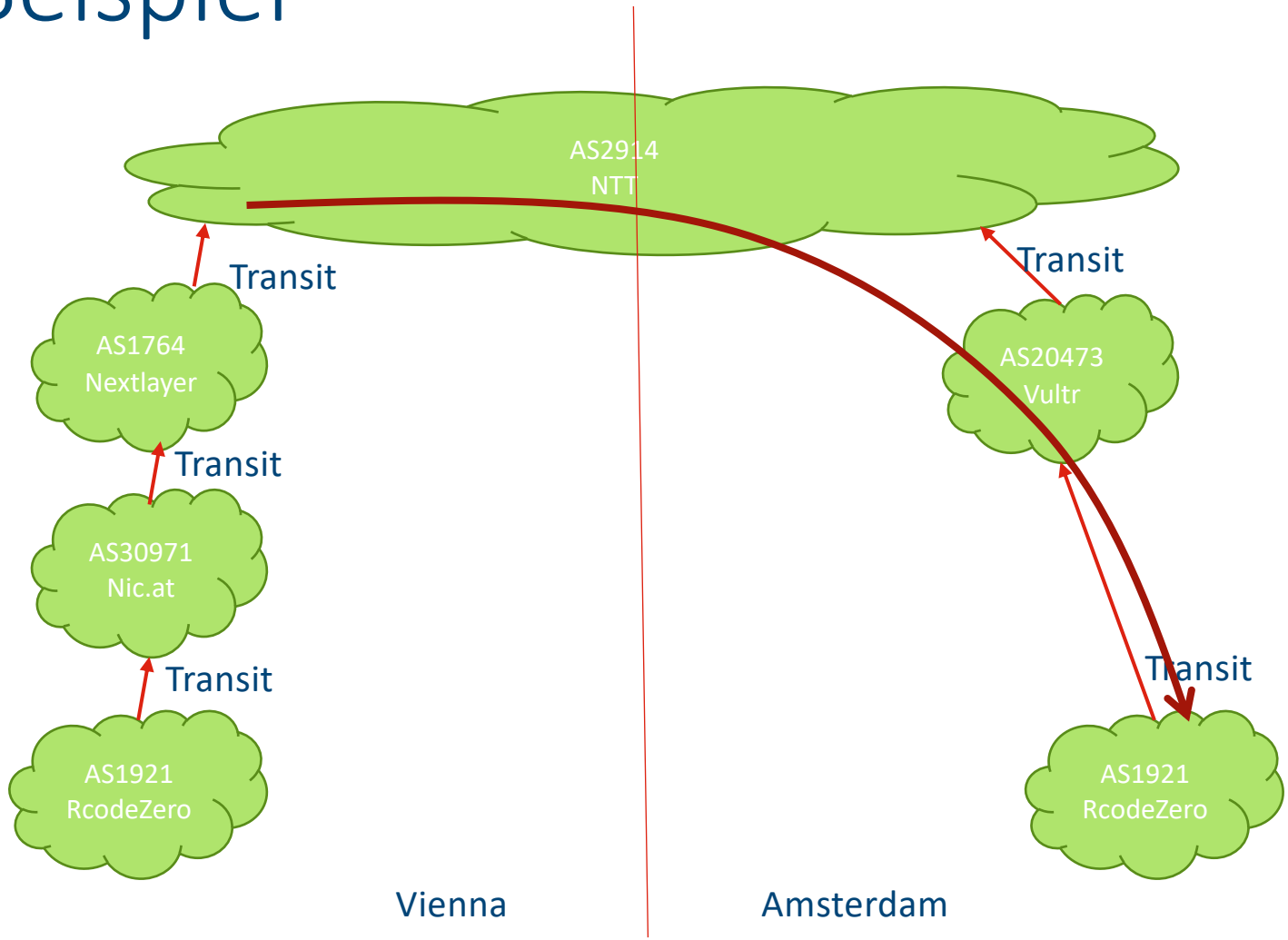
Optimierung

- Globale Transit Provider:
  - Transit auf jedem Standort (zumindest Kontinent) oder gar nicht
  - Sonst routet dieser Provider weltweiten Traffic immer nur zu diesem einen Standort

  → besser überall den gleichen Tier 1 als auf jeden Standort einen anderen Tier 1 Provider

NTT sieht 2 Routen für 192.174.68.0/24:
AMS: localPref=120, AS-Pfad len=2: 20473 1921
VIE: localPref=120, AS-Pfad len=3: 1764 30971 1921
→ NTT routet immer nach

Beispiel



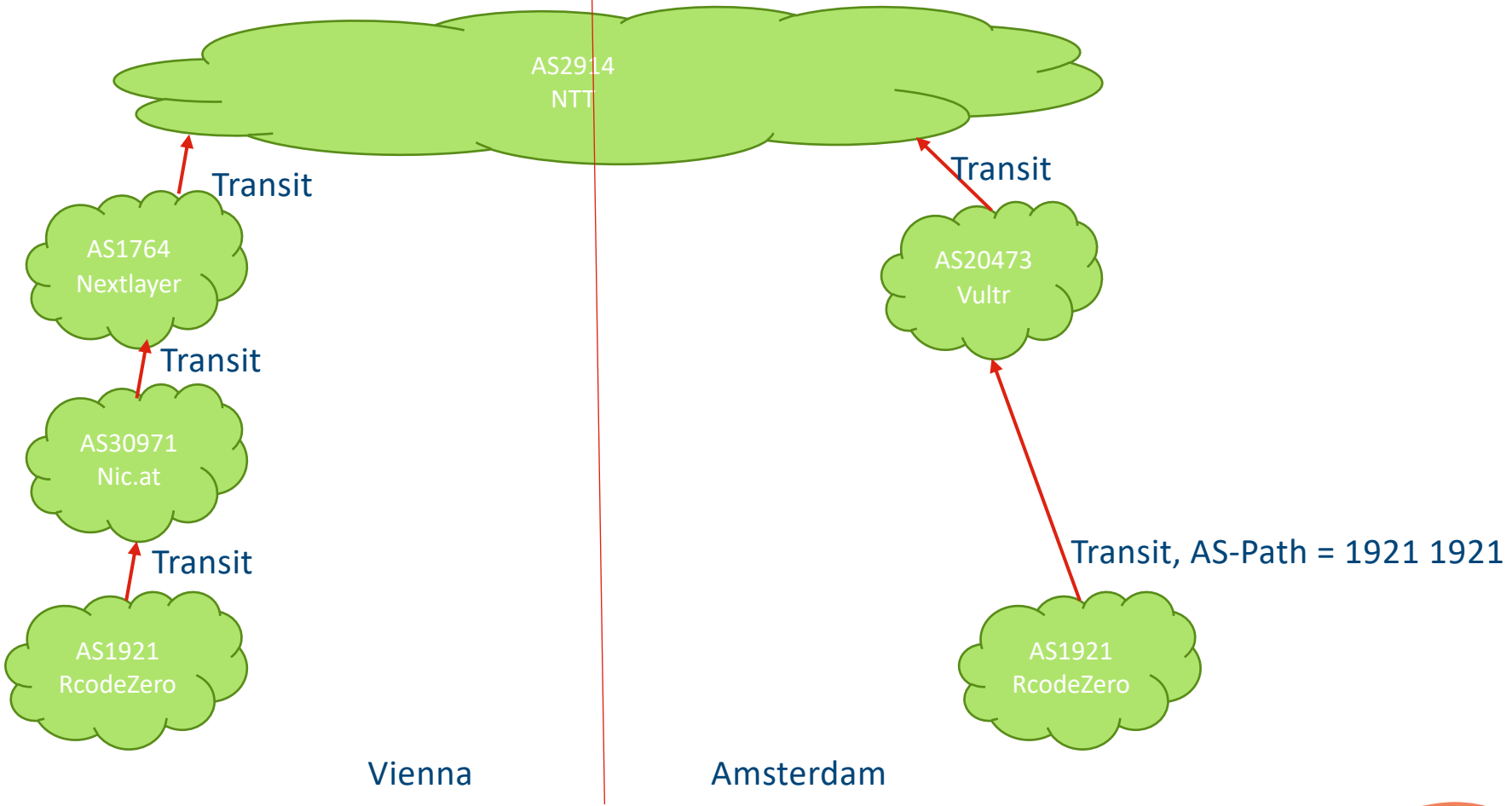
Optimierung

- AS Pfad zu großen Providern soll gleich lang sein

→ AS Pfad prepending

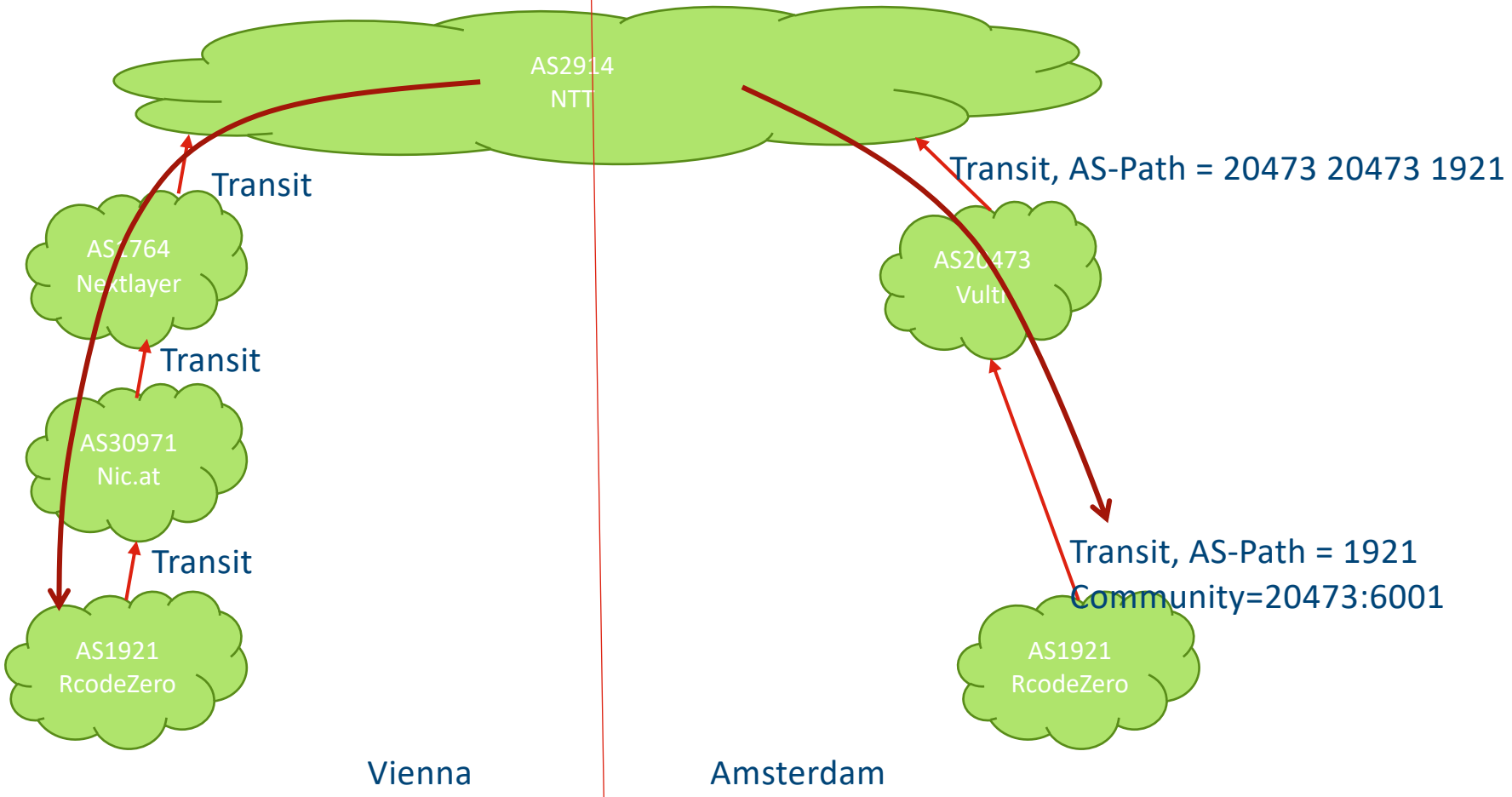
NTT sieht 2 Routen für 192.174.68.0/24:
AMS: localPref=120, AS-Pfad len=3: 20473 1921 1921
VIE: localPref=120, AS-Pfad len=3: 1764 30971 1921
→ NTT routet zum nächsten Standort

Prepending Variante 1



NTT sieht 2 Routen für 192.174.68.0/24:
AMS: localPref=120, AS-Pfad len=3: 20473 20473 1921
VIE: localPref=120, AS-Pfad len=3: 1764 30971 1921
→ NTT routet zum nächsten Standort

Prepending Variante 2



Prepending via „prepend“

- Juniper

```
policy-statement SELECTIVE-PREPEND-V6 {
        term TLD1-V6 {
            from {
                prefix-list tld1-v6;
            }
            then as-path-prepend "1921";
        }
```

- Cisco

```
route-map transit-out-v4 permit 10
 description Prepend 1x
 match ip address prefix-list tld1-v4
 set as-path prepend 1921
```

Prepending via upstream community

- Juniper

```
policy-statement SELECTIVE-COMMUNITIES-V4 {
  term TLD1-V4 {
    from {
      prefix-list tld1-v4;
    }
  then {
    community add VULTR-PREPEND-1x-ALL-PEERS;
  }
}
community VULTR-PREPEND-1x-ALL-PEERS 20473:6001;
```

- Cisco

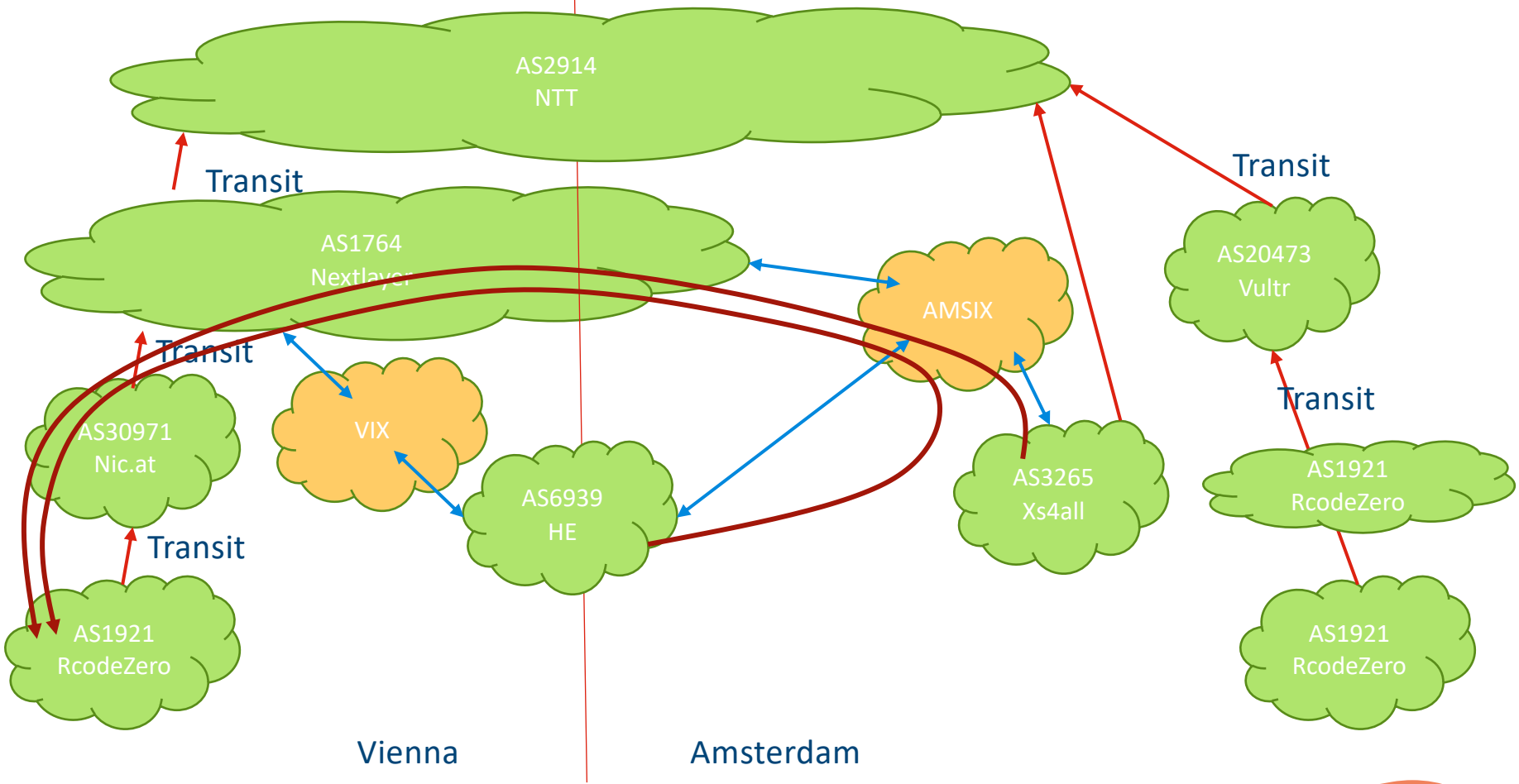
```
route-map transit-out-v4 permit 10
 description Ask Vultr to Prepend 1x
 match ip address prefix-list tld1-v4
 set community 20473:6001
```

Wo findet man BGP Communities

- Websuche: auch alte ISP-Namen verwenden(CTBC)
- Foren
- Whois
- Reverse Engineering

Probleme:
Nextlayer announct uns am AMSIX und zieht Traffic von Amsterdam nach Wien

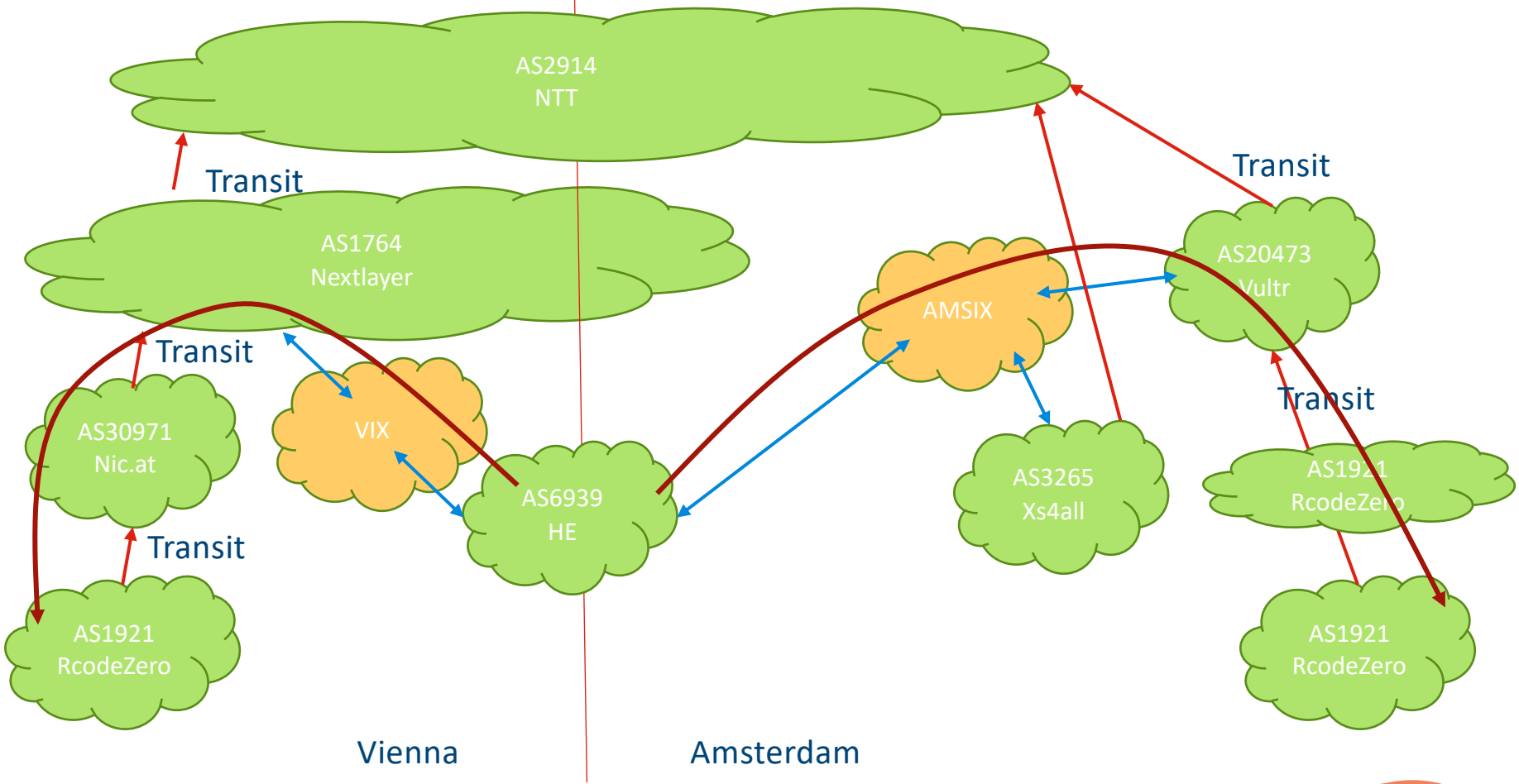
Beispiel Peering schlecht



Optimierung

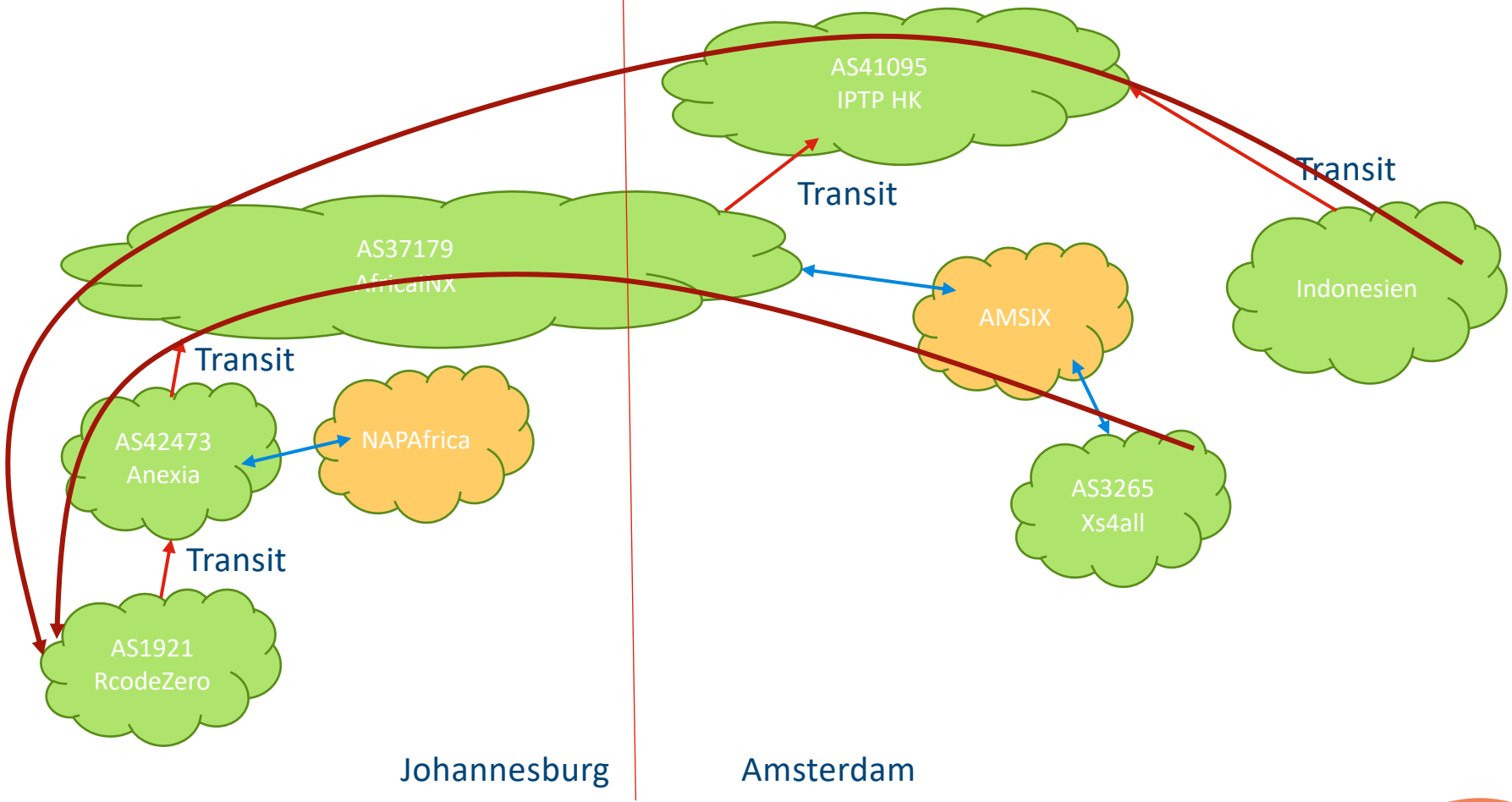
- Route darf nur zu lokalem Exchange announct werden
  - Nur regionale Tier 2 verwenden
  - Keine Tier 2 verwenden
  - Tier 2 mit guten BGP Communities: do not announce AMSIX
- Entweder jeder Standort am lokalem Exchange oder gar keine Exchanges verwenden
- AS-Pfad-Länge zu jedem Exchange muss ident sein

Beispiel Peering gut



Probleme:
AfricaINX Backhaul to Amsterdam für Transit und Peering
Keine Communities
Blockt Transitive Communities

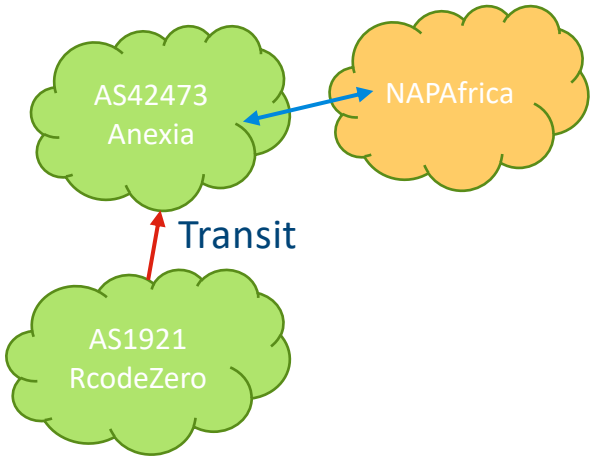
Beispiel Transit schlecht



Optimierung

- Schlechte Transits deaktivieren
  - Anexia Community: 47147:3200 REJECT_TYPE_UPSTREAM
- Anycast Standort ohne Transit
  - → „Local Node“

Beispiel Local Node



Johannesburg

Amsterdam

Regeln für gutes Routing

- Globale Transit Provider:
  - Auf jedem Standort (zumindest Kontinent) oder gar nicht
  - Globale/Große Tier 2 sind suboptimal als Transit, da sie Traffic von vielen Exchanges ansaugen (HE, Core-Backbone)
- Announce to Exchanges?
  - Auf jedem Standort (zumindest Kontinent) oder gar nicht
  - Nur zu lokalem Exchange
- Idente AS-Pfad Länge: Zu Exchanges und globalen Providern (Tier 1, große Tier 2)
- Schlechte Transits deaktivieren
  - „Local Node“

Tools

- RIPE RIS (Routing Information Service)
  - (Fast) Echtzeit Looking Glass: https://stat.ripe.net/data/looking-glass/data.json?preferred_version=2.0&resource=192.174.68.100
  - Analyse von Standorten
  - Analyse von RcodeZero
- ISP LGs: HE, NTT sind top (zeigen alle Routen), andere OK bis schlecht

Benchmark

- Dnsperf.com
  - Peropfs.net: kommerzieller Dienst
- RIPE Atlas
  - Für Debugging und finden von schlechten Routen