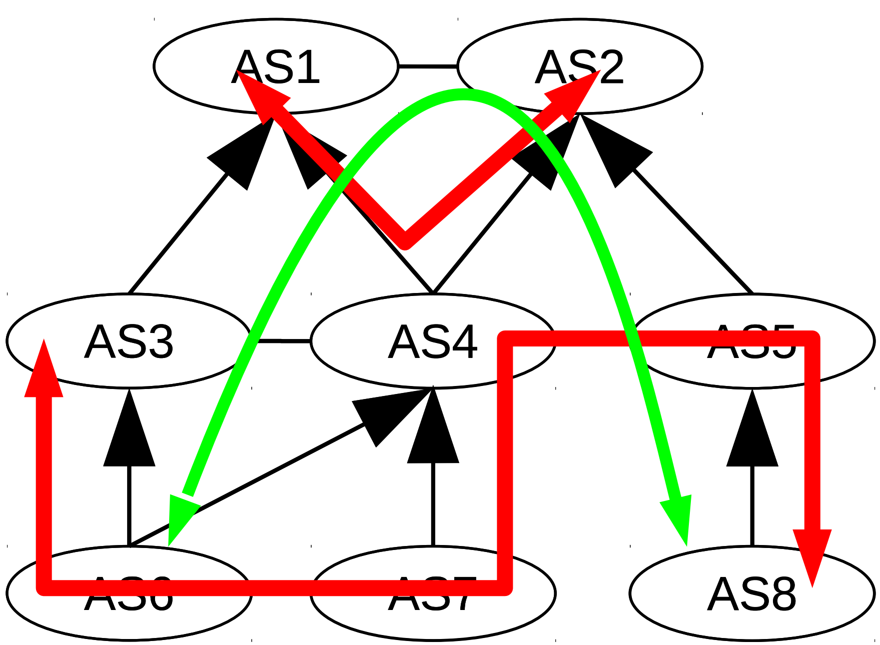
The Internet Ecosystem and Evolution

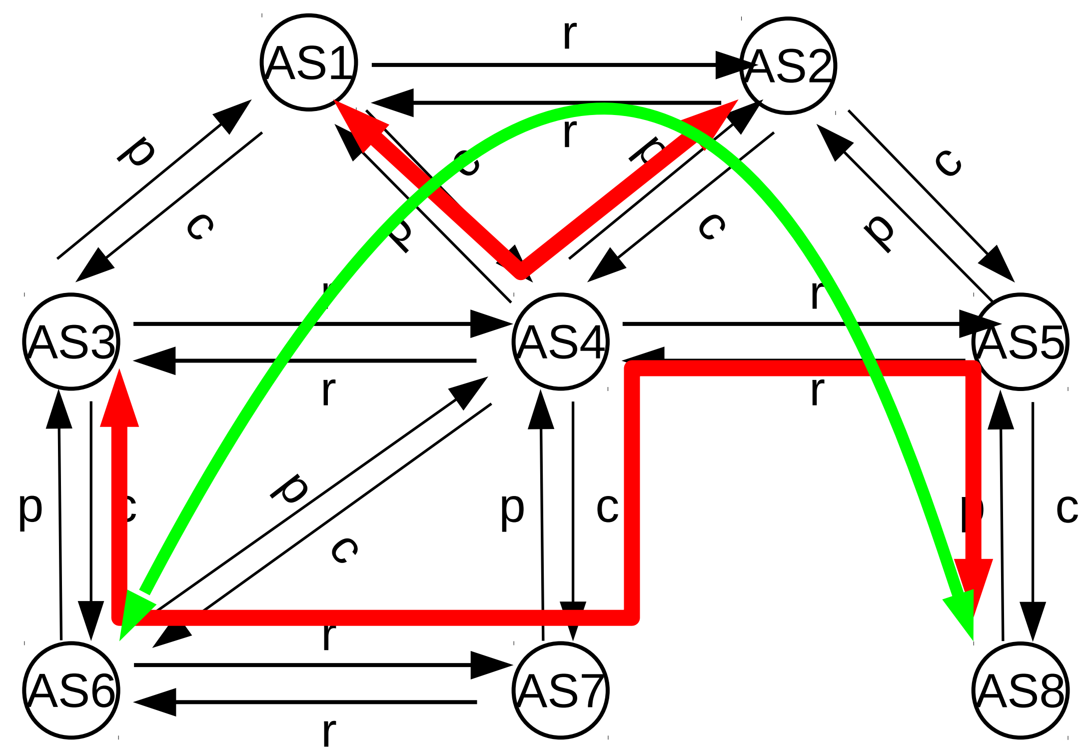
Contents

- Advanced BGP
  - reminder: configuring valley-free routing
  - enforcing "prefer-customer"
  - configuring/non-configuring shortest AS path
  - prefix hijacking and prefix filtering
  - filtering AS paths
  - backup routing and AS-path prepending
  - hot-potato routing
  - traffic engineering

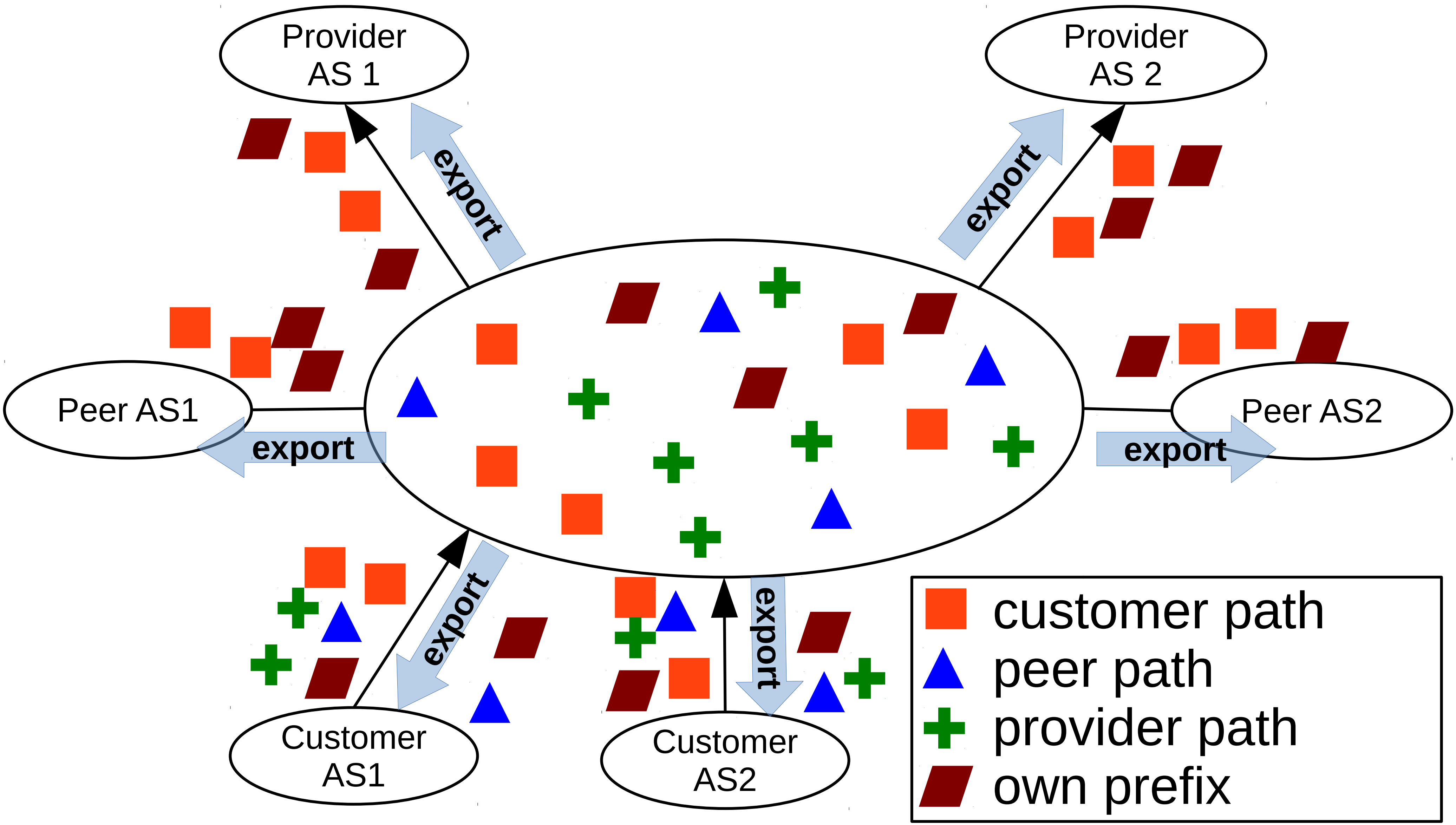
Recall: AS-AS links & services

- AS–AS business relationships: **transit/peer**
- Feasible/prohibited paths: **valley-free routing**
- Must be configured into BGP: **import/export filters**





Recall: „valley-free” BGP filters



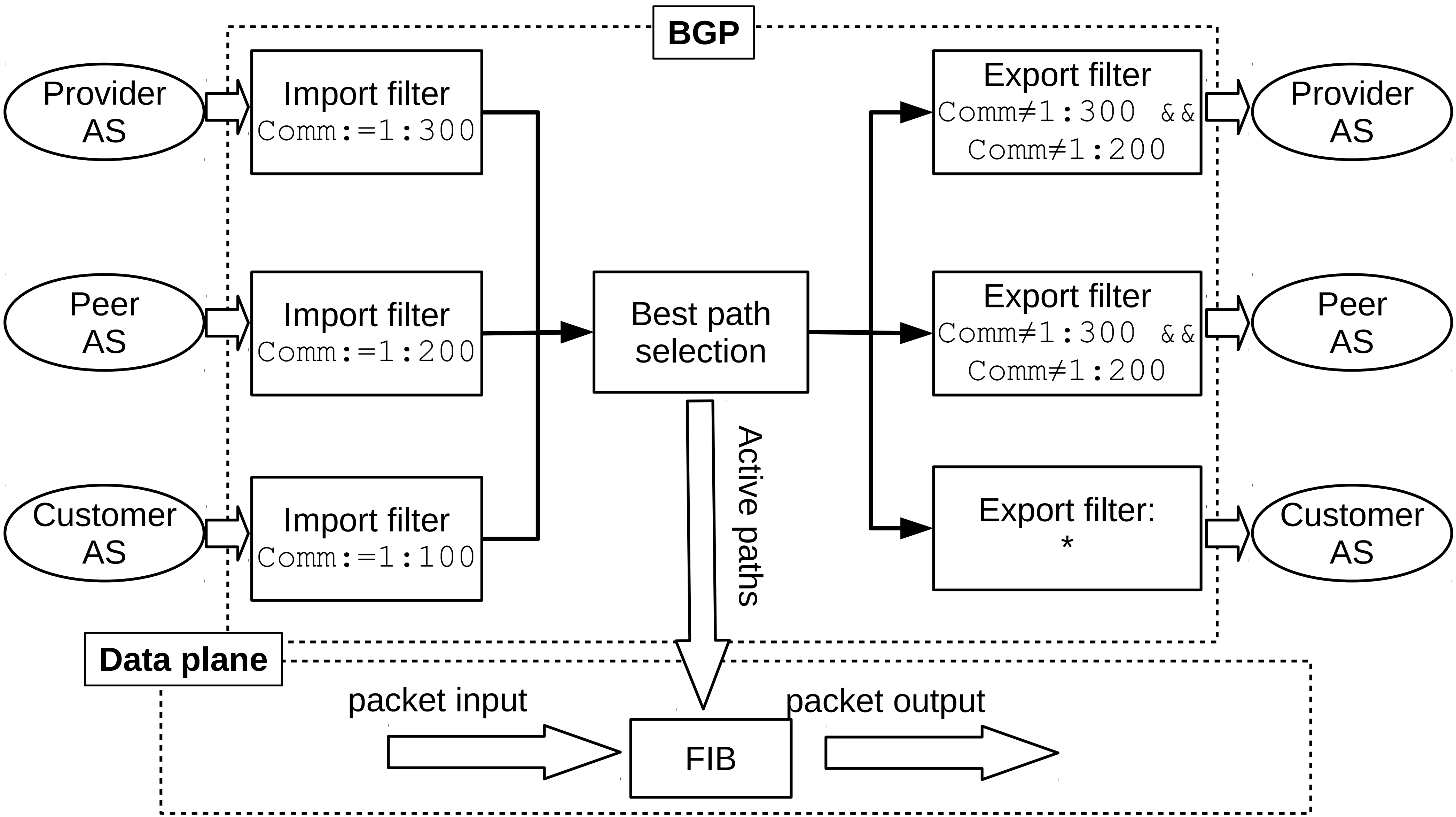
Recall: BGP

- BGP routers generate announcements:

  BGP announcement = prefix + attributes

- Important attributes: AS_PATH, NEXT_HOP, LOCAL_PREF, COMMUNITIES
- Received announcements → **import filters** → BGP RIB → **best path selection: active path** → FIB
  - "destinations" are each prefix
  - prefer announcements with high LOCAL_PREF
  - tie-breaking by AS path length (shorter: better)
- Active paths (and only active paths!) are advertised to neighbors: subject to **export filters**

Recall: BGP valley-free routing



BGP best path selection process

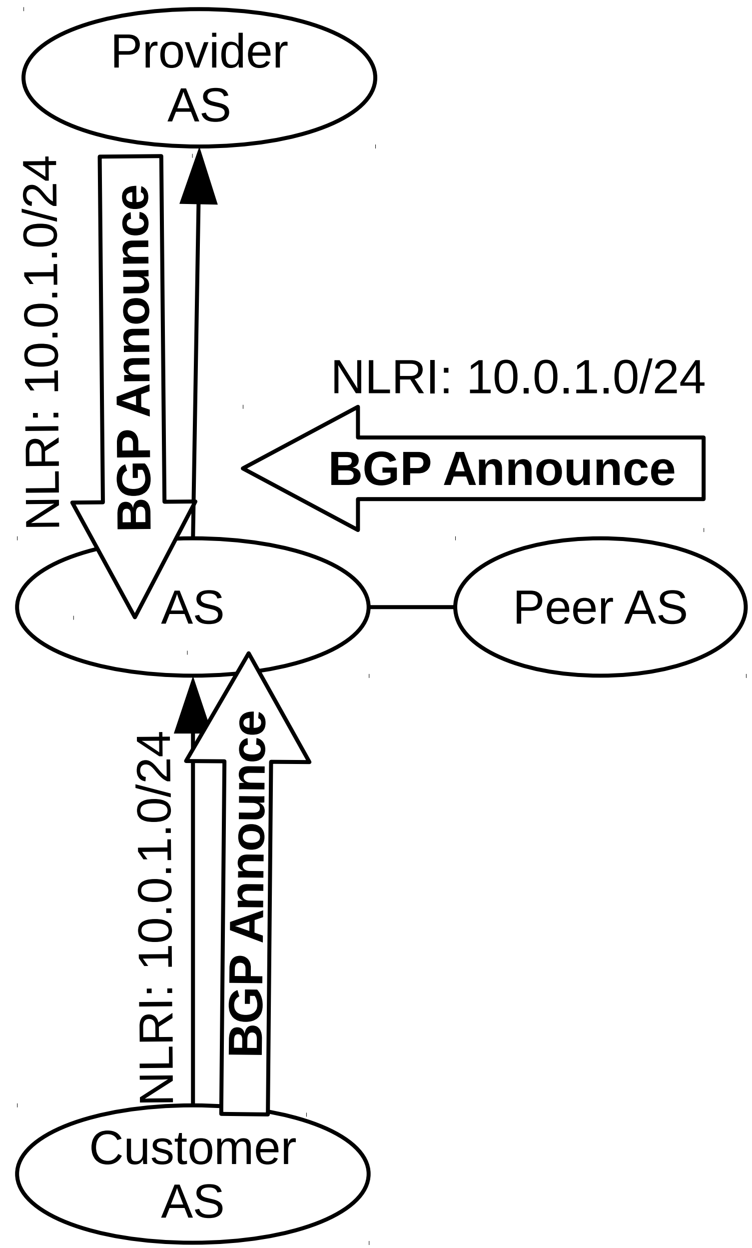
Priority	BGP attribute	Usage
1.	Higher local preference (LOCAL_PREF)	• Prefer customer policy • Choosing a primary provider (see later)
2.	Shorter AS path (AS_PATH)	• Managing traffic inside an AS • Choosing a border router for egress traffic via iBGP
3.	Lower Multi-Exit Discriminator (MED)	
4.	iBGP announcements preferred over eBGP ones	
5.	Smaller IGP administrative cost to BGP border router	
6.	Smaller router-id	• Tie-breaking

Advanced policy routing

- The Internet services market is competitive
  - to stay ahead and maximize profit
  - suboptimal paths lead to profit loss and securoty threats (prefix hijacking)
- ISPs need to **fine-tune routing policies** (beyond valley-free routing)
- Below, we discuss some typical ISP routing policies and the respective BGP configuration

Prefer-customer: BGP config

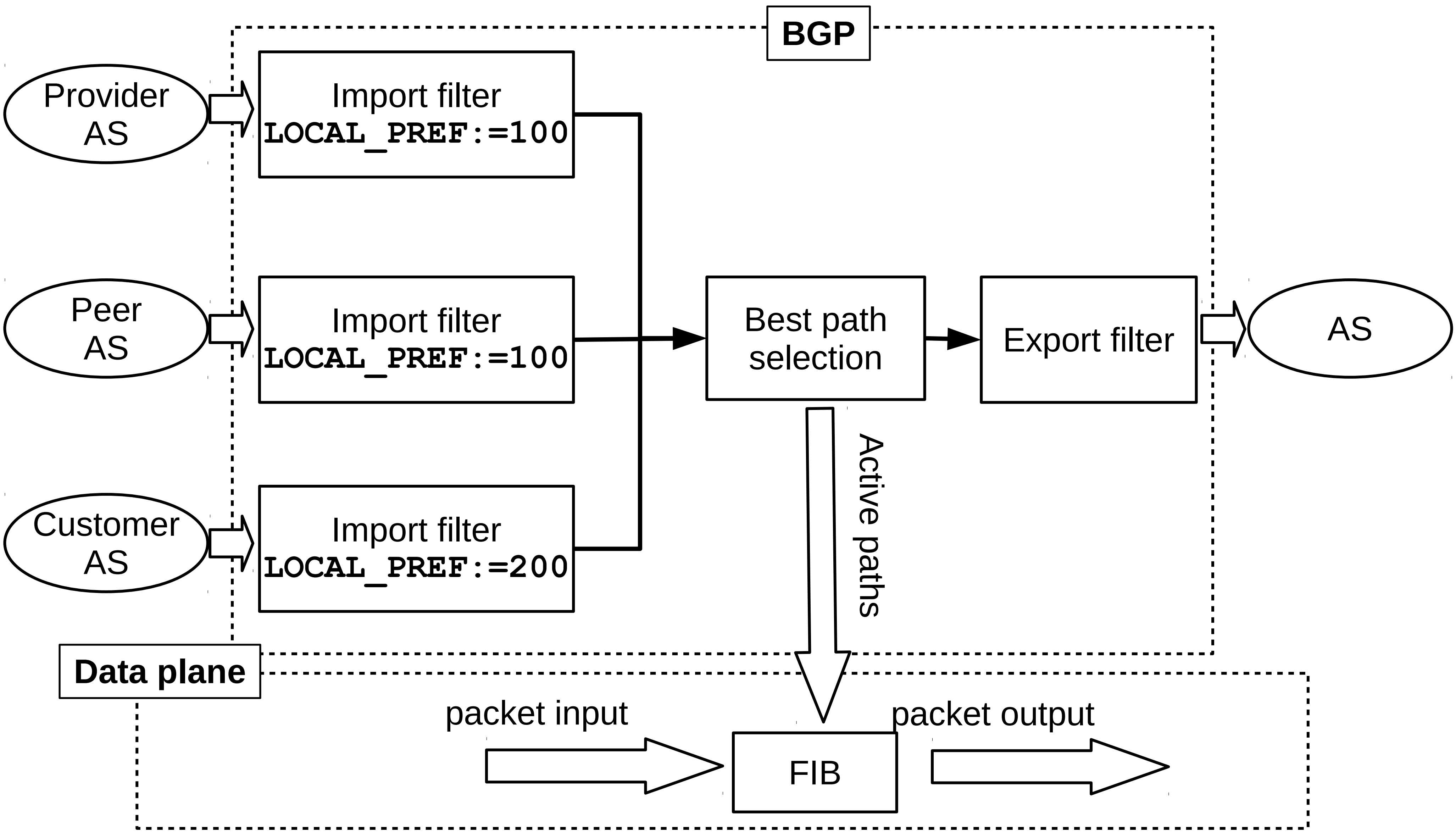
- By the **prefer customer** rule, paths via customers are preferred over paths via peers/ providers (to avoid transit fees)
- Set BGP announcements from customers to high priority
- The **local preference** attribute serves just this purpose
- For now, we do not differentiate providers and peers



Prefer-customer: import filter

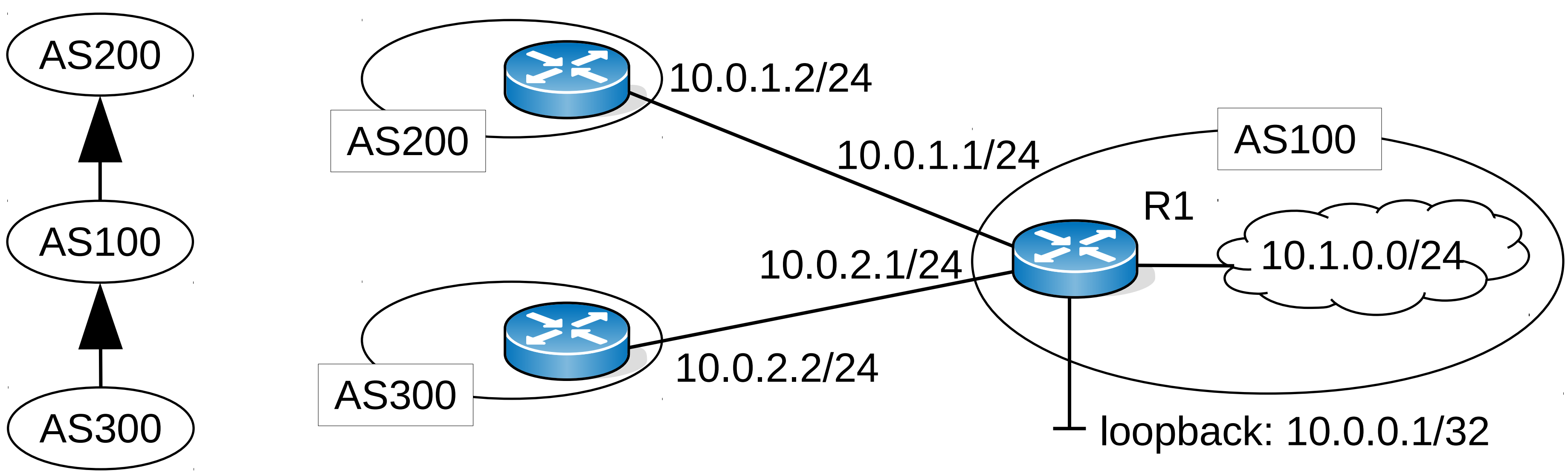
- Recall that the **local preference** attribute is the most important attribute BGP considers during the best path selection process
  - if the BGP RIB contains more than one announcement to a prefix
  - the one with the higher LOCAL_PREF wins
- **BGP configuration for prefer-customer:** set the local preference on BGP announcements received from customers to a high value by an **import filter**
- So BGP will automatically prefer customer paths dutring path selection

Prefer-customer: import filter



Prefer-customer: BGP config.

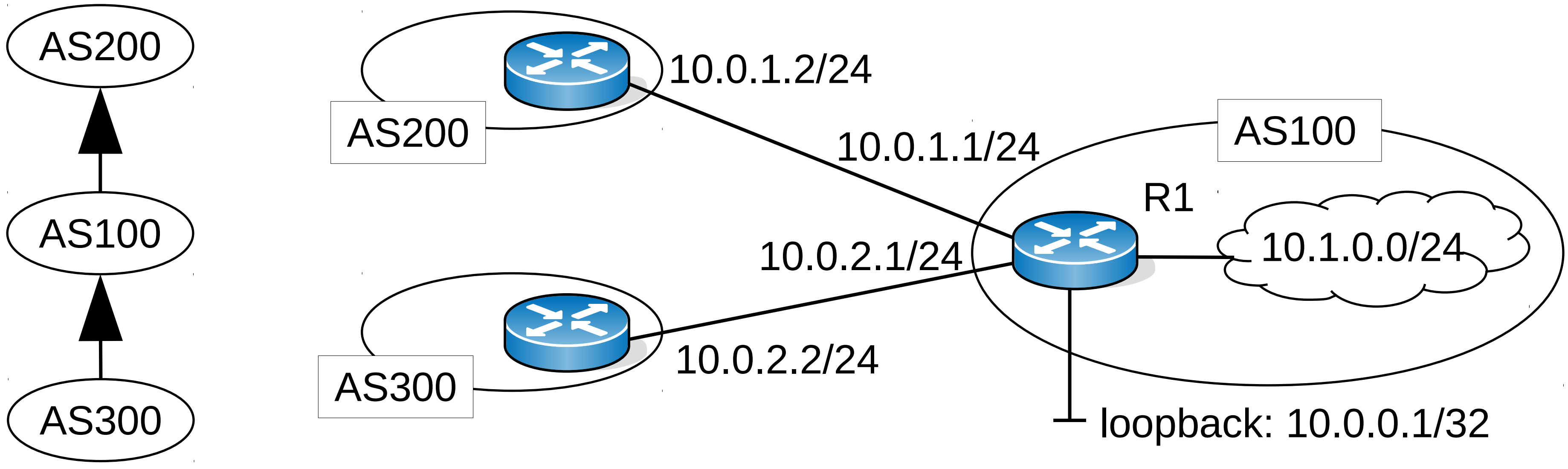
- Suppose now that `AS300` is a customer and `AS200` a provider of `AS100`
- By the “prefer-customer” rule, router `R1` prefers announcements received from `AS300`



Prefer-customer: BGP config.

- Default LOCAL_PREF setting: `100`
- Enough to set the LOCAL_PREF to `200` on customer announcements by an import filter:

```
route-map rm-cust-in permit 10
   set community 1:100
   set local-preference 200
```

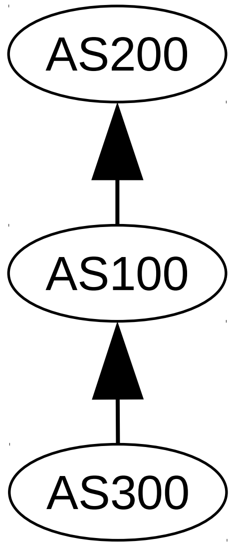


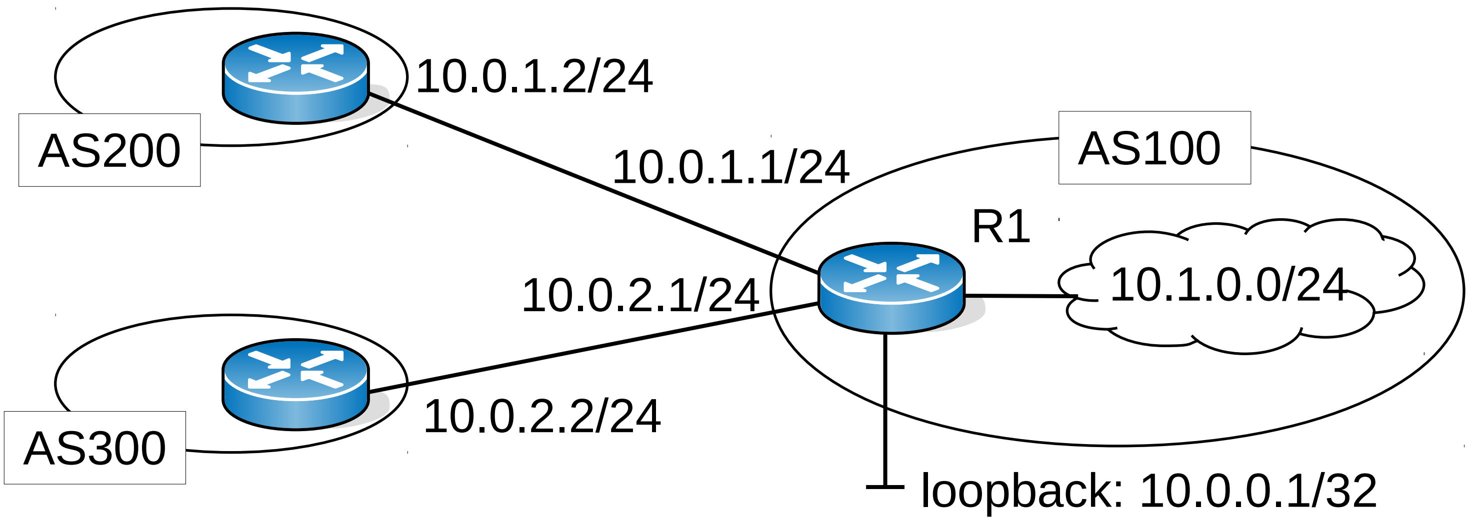
Prefer-customer: BGP config.

- Attach the import filter to the customer AS

```
neighbor 10.0.2.2 route-map rm-cust-in in
```

- The last `in` clause sets the filter's direction
- Now BGP will favor customer routes during path selection





Prefer-customer: Example

```
!!! BGP router configuration
!!! Comunities:
!!!   1:100: customer
!!!   1:200: peer
!!!   1:300: provider
router bgp 100
 bgp router-id 10.0.0.1
 network 10.1.0.0/24
 neighbor 10.0.1.2 remote-as 300
 neighbor 10.0.1.2 route-map rm-prov-set-cm in
 neighbor 10.0.1.2 route-map rm-no-export out
 neighbor 10.0.2.2 remote-as 200
 neighbor 10.0.2.2 route-map rm-cust-in in

!!! cont'd on next page
```

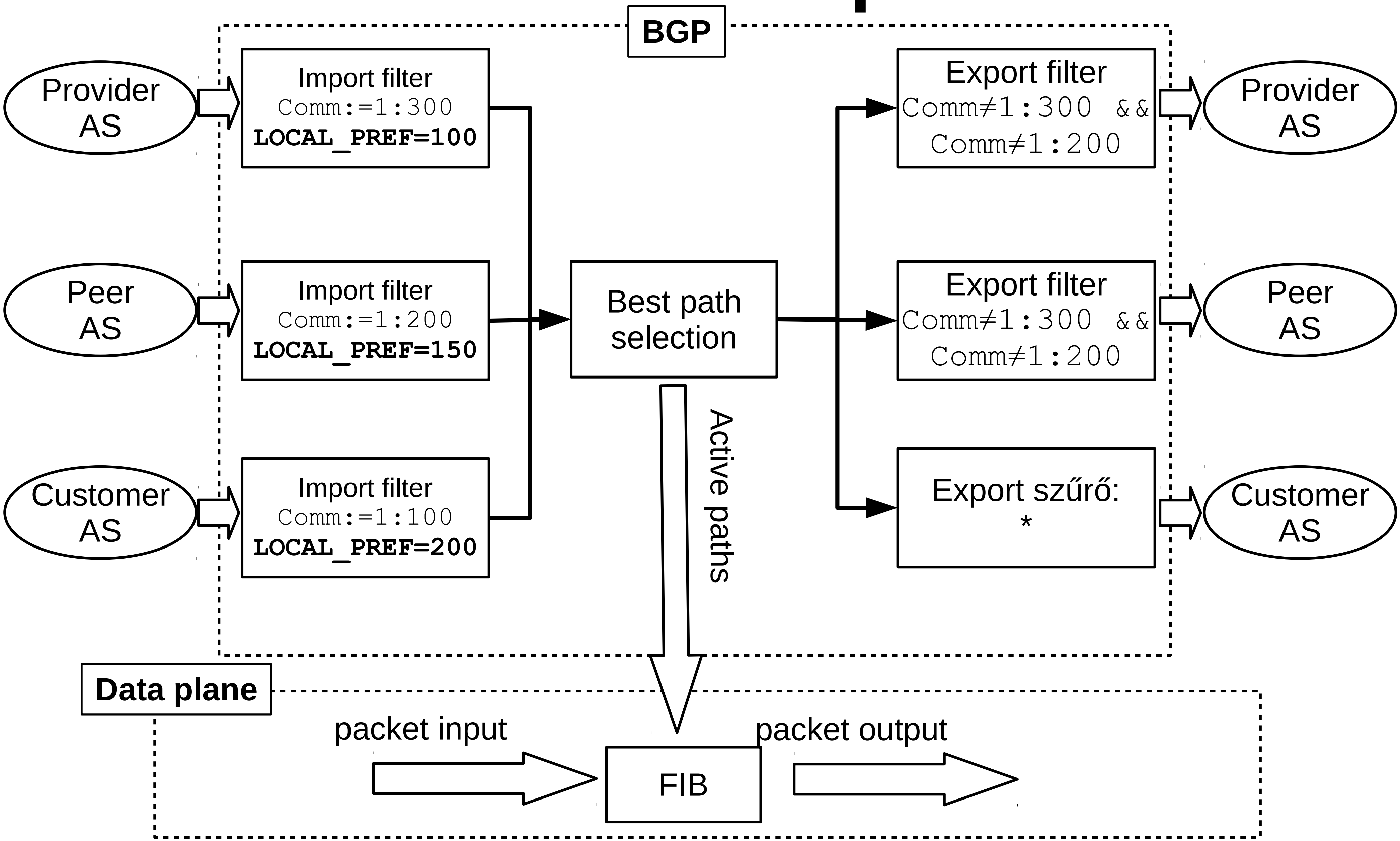
Prefer-customer: Example

```
route-map rm-prov-set-cm permit 10
 set community 1:300

route-map rm-peer-set-cm permit 10
 set community 1:200

route-map rm-cust-in permit 10
 set community 1:100
 set local-preference 200

ip community-list standard cm-no-export permit 1:200
ip community-list standard cm-no-export permit 1:300

route-map rm-no-export deny 10
 match community cm-no-export

route-map rm-no-export permit 20
```

Shortest AS path: BGP config.

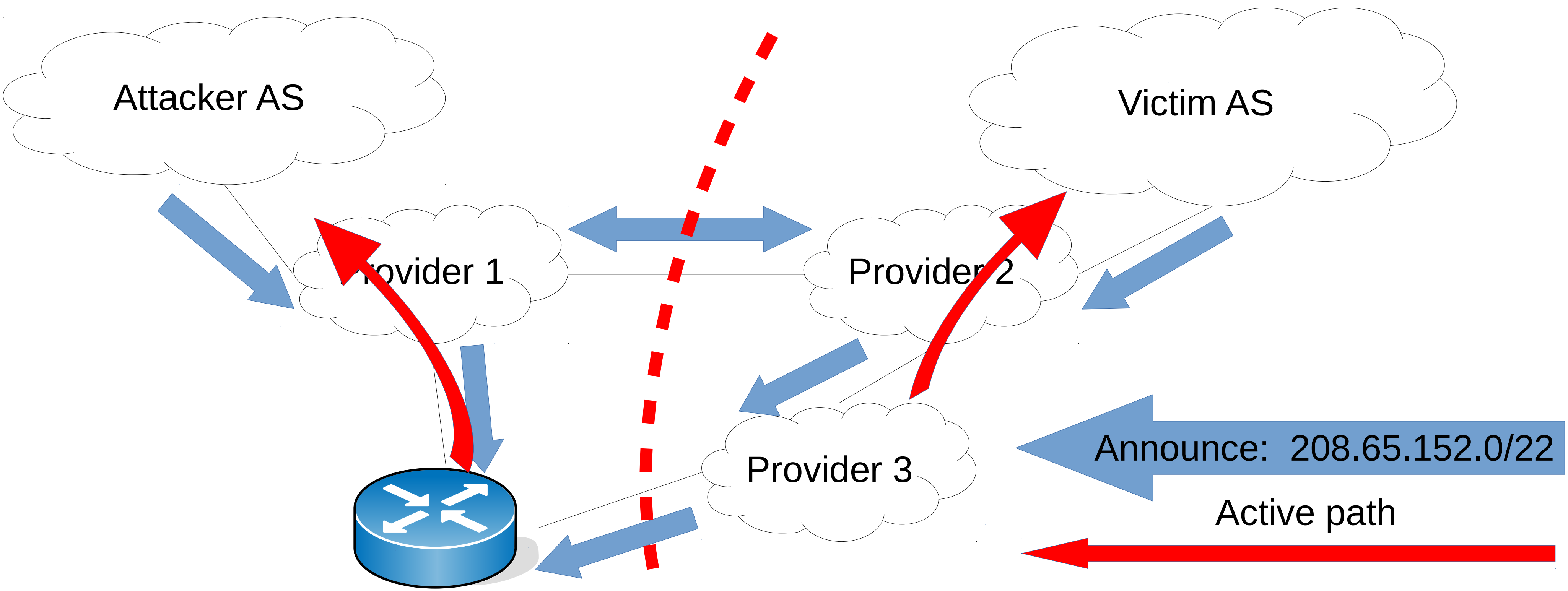
- A router with the above BGP configuration will
  - announce only **valley-free** paths to neighbors
  - implement the **prefer-customer** rule
- Interestingly, it also implements **shortest AS-path** without explicit configuration
  - if more than one customer route is available
  - these will all have the same local preference
  - thusly shorter AS-path length will decide
- So far, we haven't distinguished peer and provider paths: this needs additional configuration

BGP: valley-free+prefer-customer+ shortest AS-path



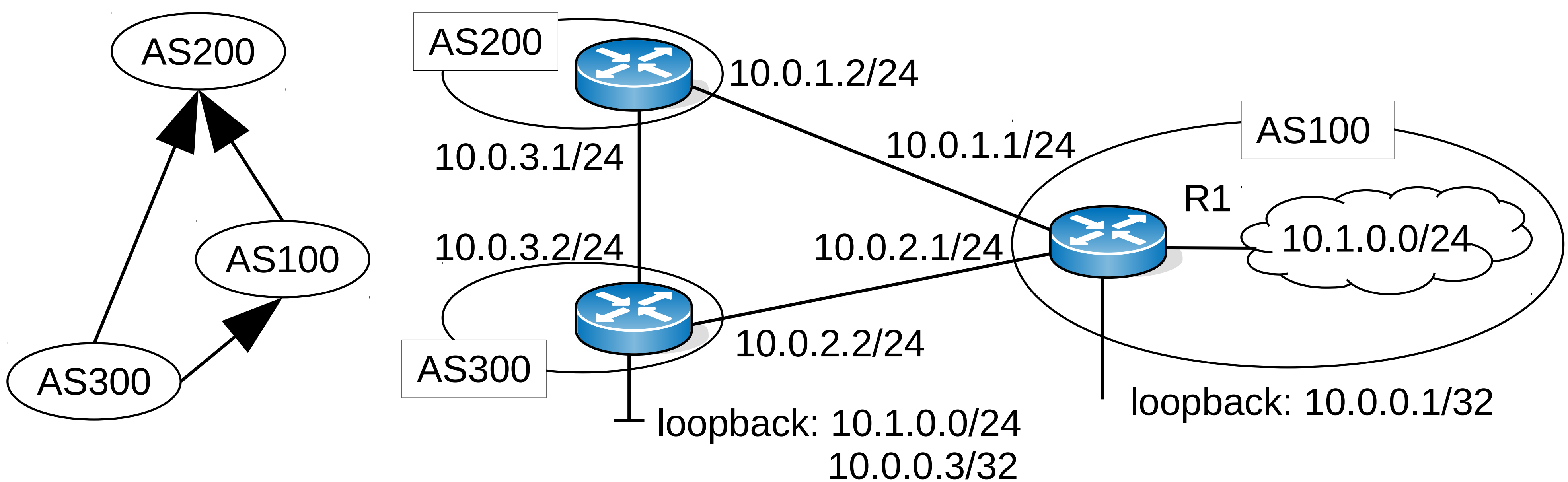
Prefix hijacking

- If a prefix is announced by more than one AS
- No way to decide, which one is authentic
- If path to Attacker is preferred: **prefix hijack**



Prefix hijacking

- Below, `AS100` and `AS300` are customers of `AS200`, and `AS100` „legitimately" announces the prefix `10.1.0.0/24`
- What if `AS300` announces the same prefix (maliciously or mistakenly)?



Prefix hijacking

```
! AS100
router bgp 100
  bgp router-id 10.0.0.1
  network 10.1.0.0/24
  neighbor 10.0.1.2 remote-as 200
  neighbor 10.0.2.2 remote-as 300
```

```
! AS300
router bgp 300
  bgp router-id 10.0.0.3
  network 10.1.0.0/24
  neighbor 10.0.3.1 remote-as 200
  neighbor 10.0.2.1 remote-as 100
```

Prefix hijacking

- `AS200` has no way to determine, which BGP announcement is legitimate
  - if lucky, if prefers the legitimate announcement and reaches `AS100` as normal
  - if unlucky, the illegitimate announcement will be chosen as active path and go into the FIB
  - the entire traffic to prefix `10.1.0.0/24` will be routed to `AS300` (the Attacker)
- This is not the only way to hijack a prefix: e.g., the Attacker may announce a more specific prefix

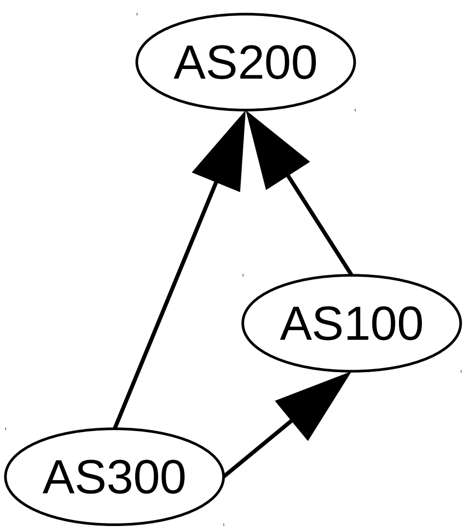
Man-in-the-middle attack: MITM

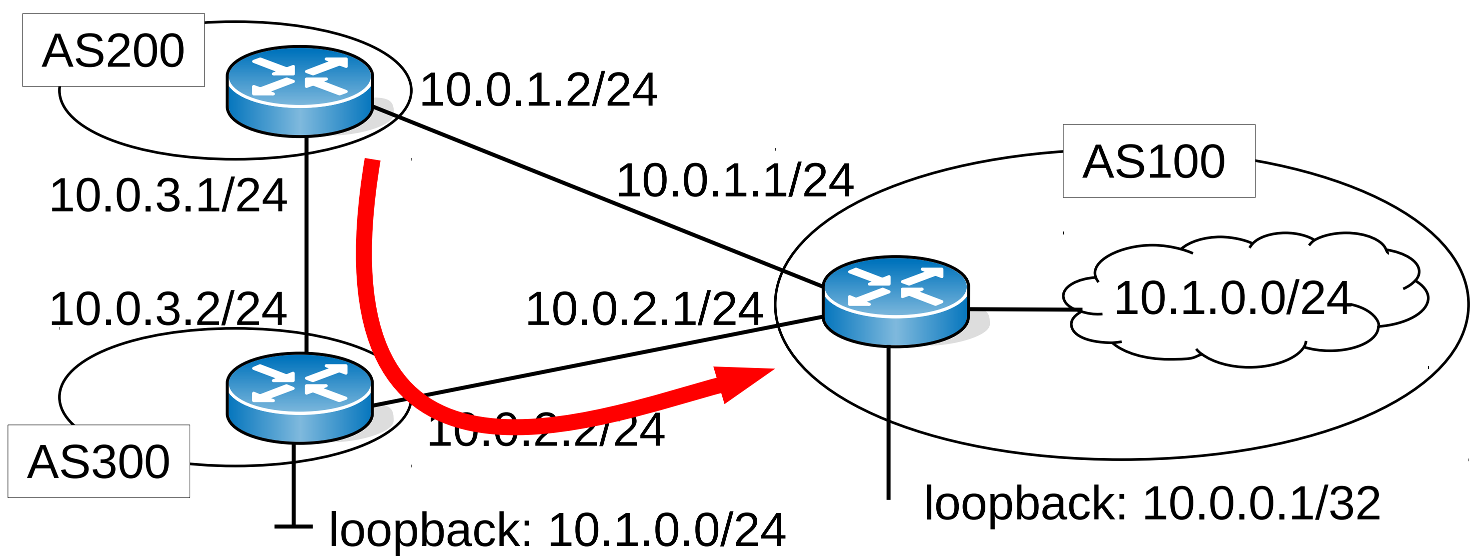
- If the Attacker `AS300` “blackholes” the traffic to `10.1.0.0/24`: it becomes unreachable for the rest of the Internet
- Often, this occurs due to a misconfiguration
- But if the hijack is hostile, the Attacker can pass the packet on to the Victim `AS100`
- Meanwhile, the Attacker can sneak up on/intercept the Victim's traffic, without it noticing this at all
- Or inject malicious payload into the hijacked traffic (e.g., it could infect intercepted emails with a virus)

MITM attack: Example

- AS300 can insert a static route into its FIB:

  `ip route add 10.1.0.0/24 via 10.0.2.1`
- Will forward hijacked traffic to the intended AS
- Can modify passing traffic in any hostile ways





MITM attack: Prevention

1. **Monitoring:** observe the fate of an AS's prefixes
   - data plane: check reachability of our prefixes from looking glasses/route servers/etc. (ping)
   - control plane: check BGP announcements to our prefixes at BGP monitors/looking glasses
2. **SecureBGP:** cryptographically signed BGP msgs
   - can check AS-number–IP-prefix assignments
   - can also sign entire AS paths cryptographically
   - has not spread this far, only of limited use

MITM attack: Prevention

3. **Reclaim prefix by announcing a more specific**

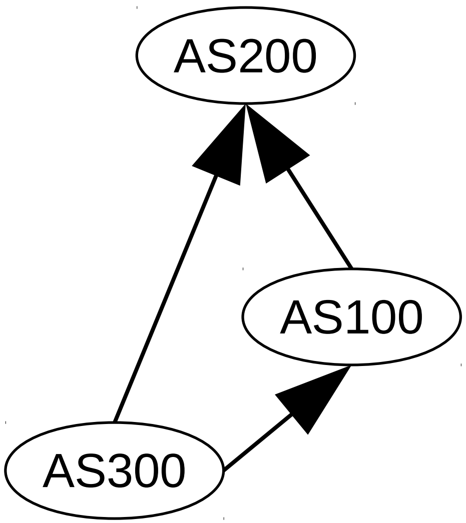
- if prefix `10.1.0.0/24` hijacked: announce more specifics `10.1.0.0/25+10.1.0.128/25` in response
- our legitimate routes will override hijacked entries in the FIBs throughout the Internet
- but /25s are often filtered by inter-domain routing

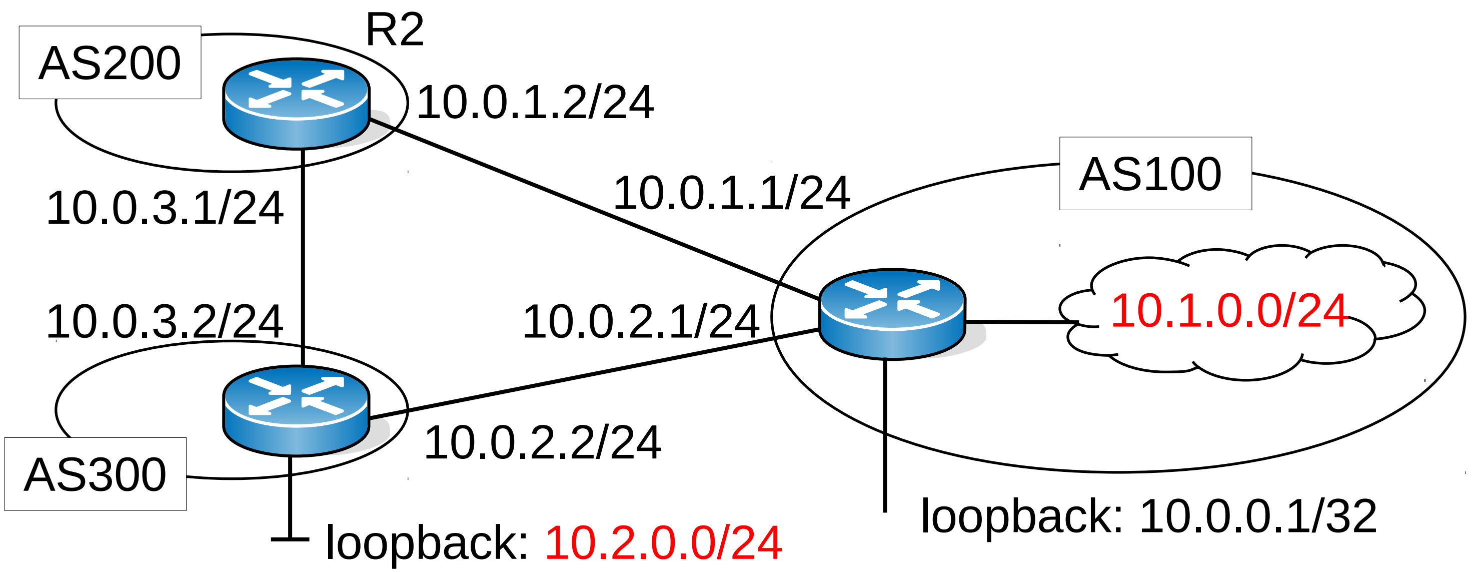
4. **Filter illegitimate BGP announcements**

- publish an AS's all prefixes in a “reliable” database: **Internet Routing Registry (IRR)**
- Routers can filter BGP announcements accordingly

Prefix filtering: BGP

- Suppose `AS100` owns prefix `10.1.0.0/24`, while `AS300` owns prefix `10.2.0.0/24`
- `AS200` wants to accept only these prefixes announced from these ASes
- Of course, import filters are the way to do that





Prefix filtering: BGP

- `R2` (the border router of `AS200`) defines the list of prefixes to be accepted from `AS100`

```
ip prefix-list AS100 seq 5 permit 10.1.0.0/24
ip prefix-list AS100 seq 10 deny 0.0.0.0/0 le 32
```

- `seq` sets the order, first permitted (`permit`) prefixes and then prefixes to reject (`deny`)
- "`0.0.0.0/0 le 32`" means "everything else"
- Similar prefix list for `AS300`:

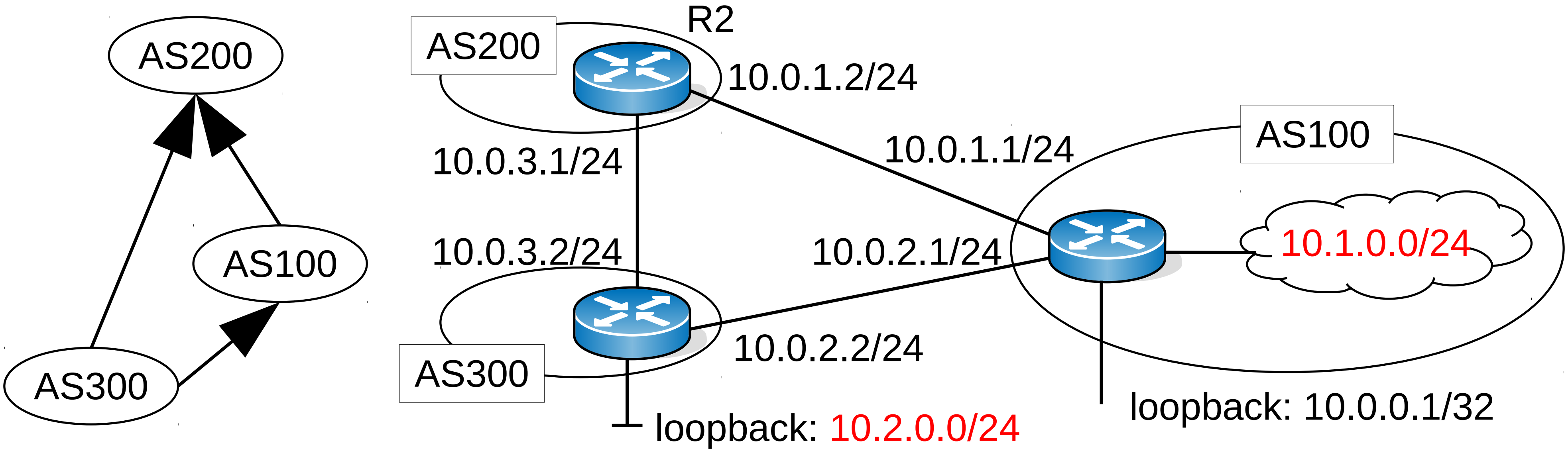
```
ip prefix-list AS300 seq 5 permit 10.2.0.0/24
ip prefix-list AS300 seq 10 deny 0.0.0.0/0 le 32
```

Prefix filtering: BGP

- Attach the prefix list to the neighbors

```
neighbor 10.0.1.1 remote-as 100
neighbor 10.0.1.1 prefix-list AS100 in
…
neighbor 10.0.3.2 remote-as 300
neighbor 10.0.3.2 prefix-list AS300 in
```

- `prefix-list` is in fact a special `route-map`



Prefix filtering: BGP

```
router bgp 200
 bgp router-id 10.0.0.2
 neighbor 10.0.1.1 remote-as 100
 neighbor 10.0.1.1 prefix-list AS100 in
 ...
 neighbor 10.0.3.2 remote-as 300
 neighbor 10.0.3.2 prefix-list AS300 in
 ...

!!! Filter legitimate prefixes from AS100
ip prefix-list AS100 seq 5 permit 10.1.0.0/24
ip prefix-list AS100 seq 10 deny 0.0.0.0/0 le 32

!!! Filter legitimate prefixes from AS300
ip prefix-list AS300 seq 5 permit 10.2.0.0/24
ip prefix-list AS300 seq 10 deny 0.0.0.0/0 le 32
```

Prefix filtering: Martians

- **Martian prefix:** prefix reserved for special purpose
  - `0.0.0.0/8`: „This network” (RFC1122)
  - `127.0.0.0/8`: loopback address range (RFC1122)
  - `192.0.2.0/24`: TEST-NET example networks (doc)
  - `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`: private address ranges for intranets (RFC1918)
  - `169.254.0.0/16`: auto-configuration
  - `224.0.0.0/4`: multicast
  - `240.0.0.0/4`: reserved for future use
- Can never appear in inter-domain routing

Prefix filtering: Bogon filters

- The IANA periodically publishes the list of prefixes officially allocated for ASes
- Everything not on the list: **bogon address/prefix**
  - special addresses, unallocated addresses, etc.
- Worth dropping BGP announcements to bogon prefixes by proper import filters: **bogon filtering**
  - DoS attacks often come from (spoofed) bogon address
  - packets to bogon addresses should also be dropped by firewalls (control-plane+data-plane filtering!)
- Many ISPs expect customers to filter bogons!

AS-path filtering

- Due to political/economical/security reasons, an ISP might want to avoid sending traffic via certain other ASes
  - e.g., the US government might not want to route sensitive traffic through a Chinese ISP
  - and vice versa
- BGP can filter paths via certain other ASes
  - based on the AS numbers appearing in a path
  - allows to bypass insecure AS paths

AS-path filtering: BGP config.

- Ignore every path from the BGP neighbor `10.10.10.10` that contains `AS200`

```
ip as-path access-list 1 deny ^.*200.*$

router bgp X
 …
 neighbor 10.10.10.10 filter-list 1 in
```

- The `ip as-path` construct creates the filter
- “`deny`“ means to reject all AS paths that match the regular expression `^.*200.*$`
- Attach to a neighbor as usual (`neighbor`)

AS-path filtering: BGP config.

- The filter itself defines a regular expression:
  - concatenate AS numbers along the path
  - separate entries by an underscore (_)
  - match resultant string against the regexp in the filter
- Permit the two-hop subpath `AS100-AS200`

```
ip as-path access-list 1 permit 100_200
```

- Reject any path containing `AS100` OR `AS200`

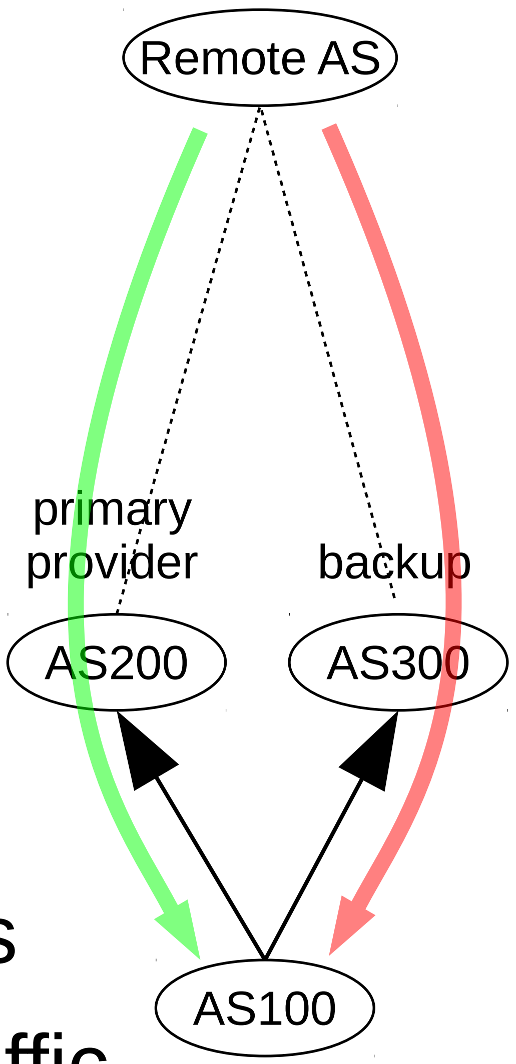
```
ip as-path access-list 1 deny _(100|200)_
```

- Drop `AS300` path-prepending (see later)

```
ip as-path access-list 1 deny _300_300_
```

Backup routing

- Frequent setup for multi-homed ASes:
  - preferred **primary** provider
  - secondary **backup** provider
  - backup used only if primary becomes unreachable
- **Goal:** force all ingress/egress traffic to the primary provider
- Simple for egress traffic: set local-prererence on primary provider's paths
- But it is difficult to influence ingress traffic

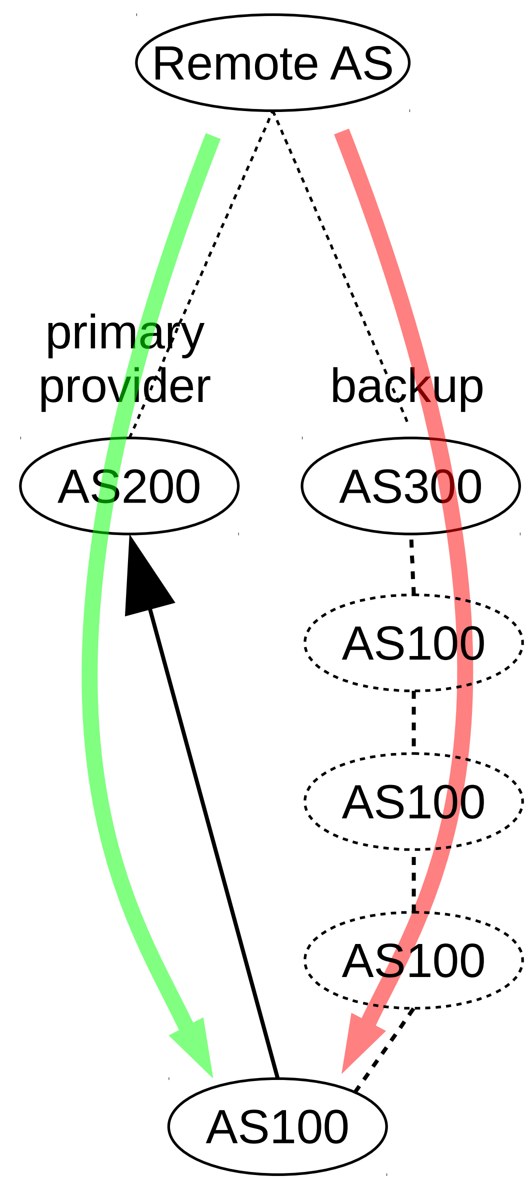


Backup routing

- **Goal:** let remote **ASes prefer the ingress path via the primary provider over the one via the backup**
- But how can `AS100` influence which path a *remote* AS chooses towards it? (action-at-a-distance)
- **Naive idea:** let `AS100` announce its prefixes only to the primary and start announcing to the backup only in case of a failure
- Must wait until new announcements spread throughout the Internet: can take minutes until everyone can reach us again
- It would be better to announce via both providers and somehow make others prefer the path via the primary

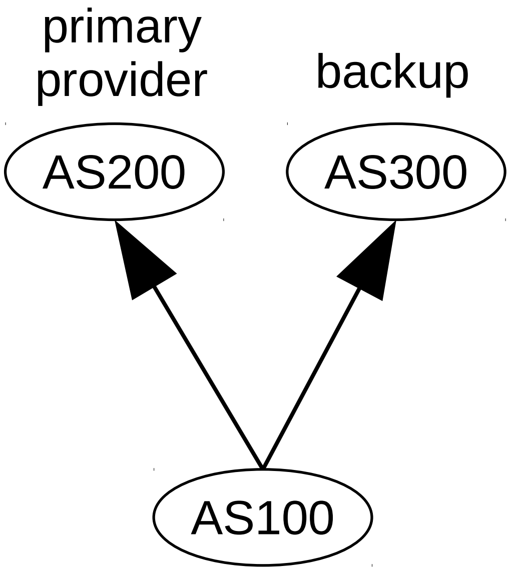
Backup: AS-path prepending

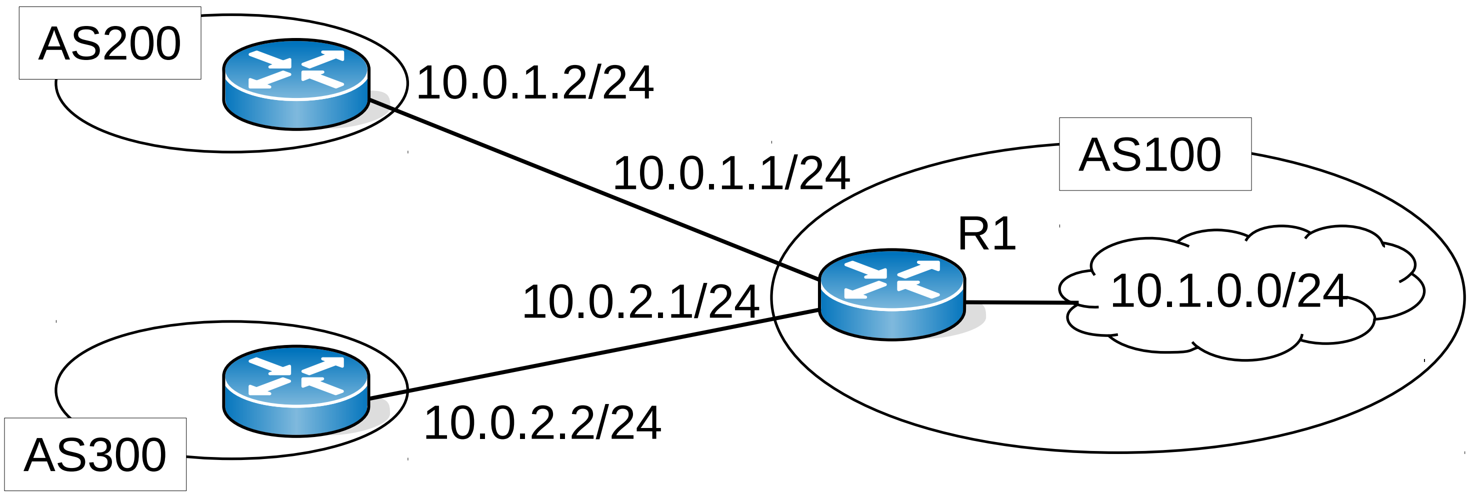
- Tricky BGP configuration: deceive others into thinking that the ingress path via the backup was longer
- **AS-path prepending:** repeat our AS number multiple times in the AS path announced to the backup
  - backup path “looks” longer
  - remote ASes will prefer shorter paths in best path selection
  - shorter path is via the primary!



Backup: AS-path prepending

- Let `AS200` now be the primary provider and `AS300` the backup provider of `AS100`
- `AS100` „lies" to the backup `AS300`: of course, we use export filters in BGP to do this



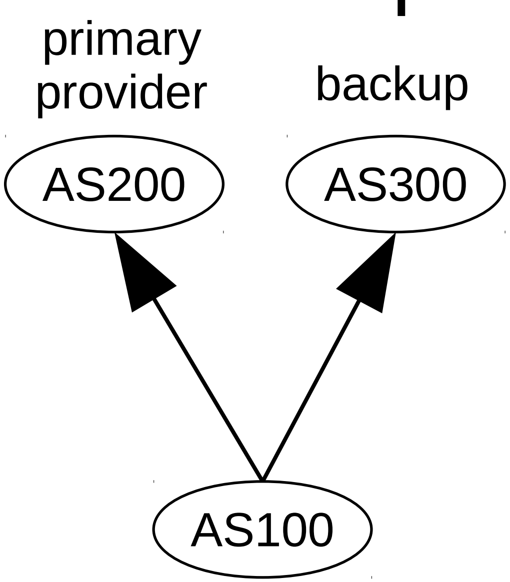


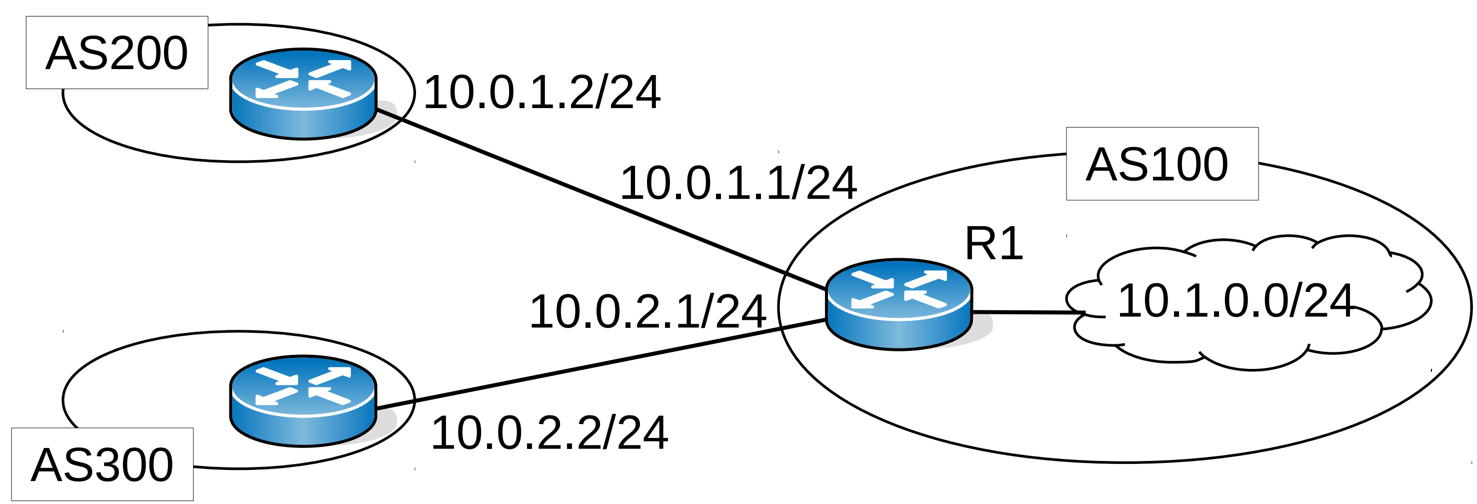
Backup: AS-path prepending

- Create a `route-map` at `R1`

```
route-map rm-as-prepend permit 10
  set as-path prepend 100 100 100
```

- The `set as-path prepend` clause injects the required number of `AS100` ids into the path



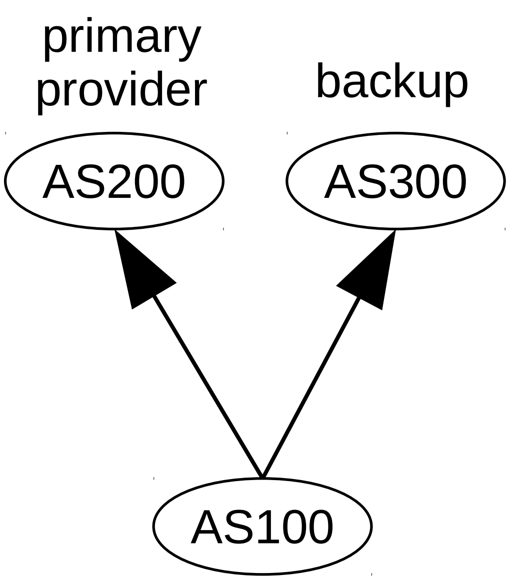


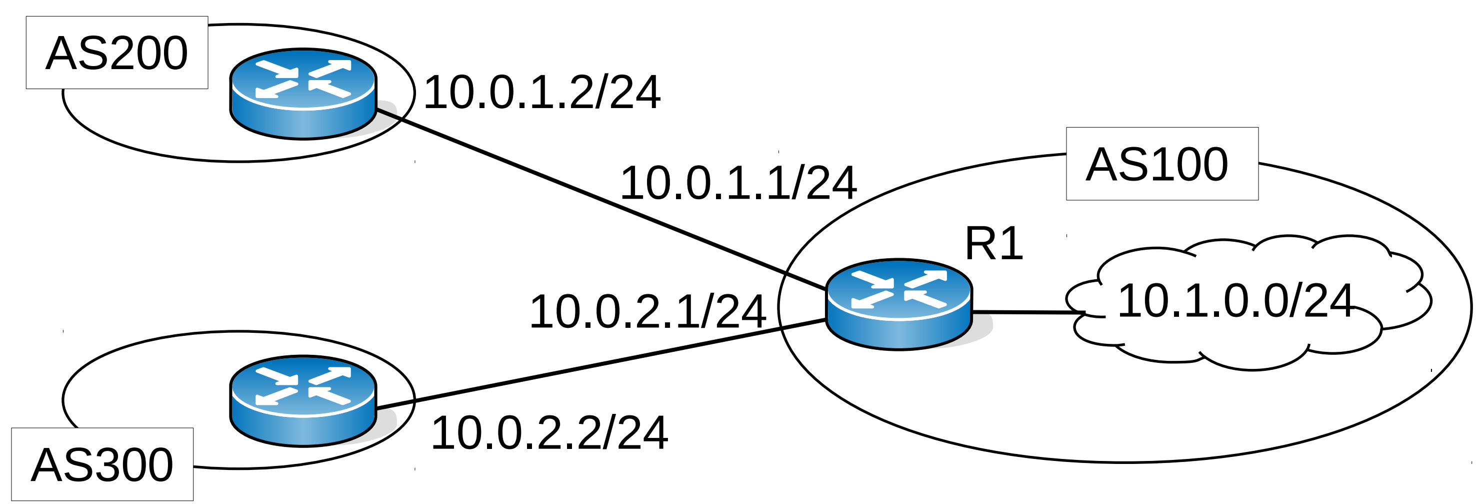
Backup: AS-path prepending

- Attach the new `route-map` to the neighbor with the usual `neighbor` command

```
neighbor 10.0.2.2 route-map rm-as-prepend out
```

- Direction is `out`, since this is an export filter





Backup: AS-path prepending

```
router bgp 100
 bgp router-id 10.0.0.1
 neighbor 10.0.1.2 remote-as 200
 ...
 neighbor 10.0.2.2 remote-as 300
 neighbor 10.0.2.2 route-map rm-as-prepend out
 ...

!!! AS-path prepending filter
route-map rm-as-prepend permit 10
  set as-path prepend 100 100 100
```

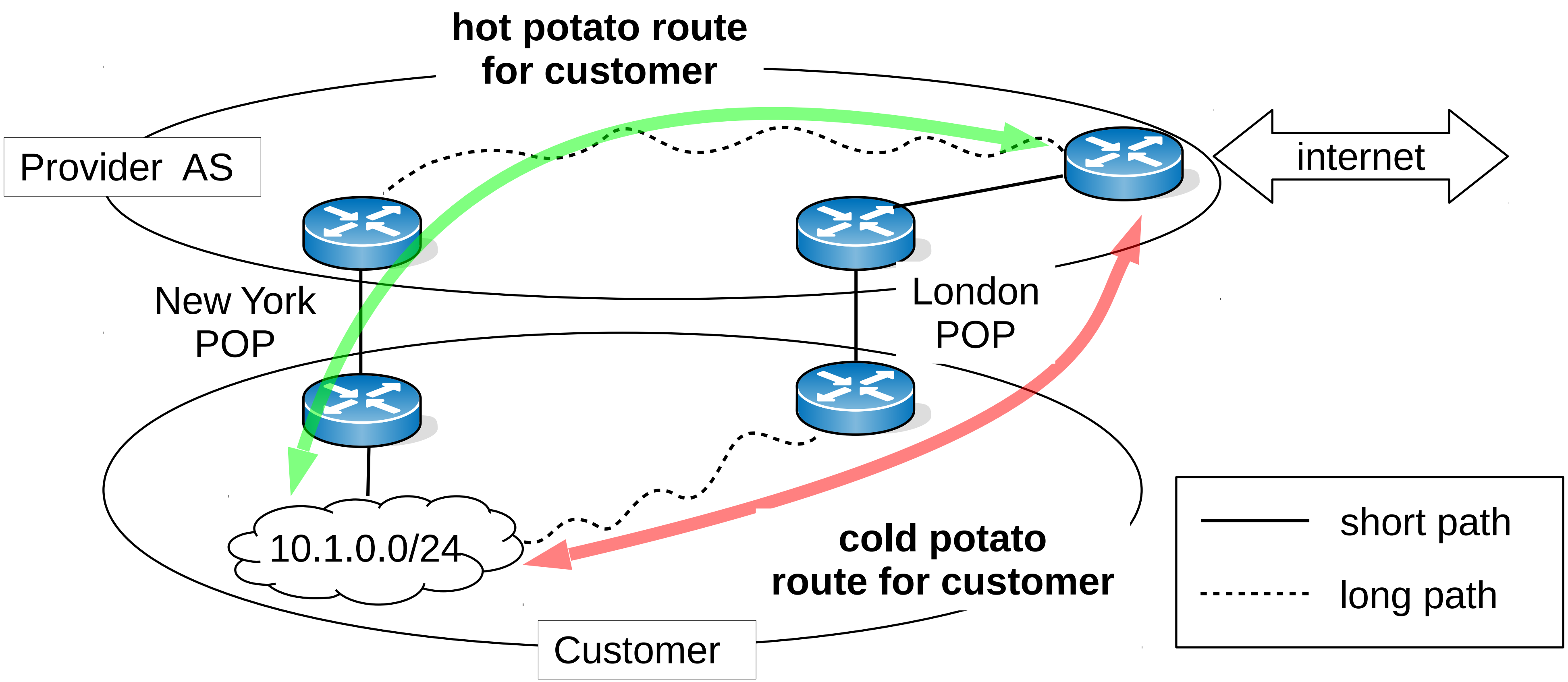
- There can be only one import/export filter active for a neighbor
- Complex BGP filters: write multiple filters, assign the same name, and combine them with `permit X/deny Y` sequence numbers

Backup: AS-path prepending

- But `AS300` still favors the backup path (by the prefer-customer rule)
- Generally, the backup and all its customers and peers will still use the backup to reach `AS100`
- AS path prepending is only a hack!
- **Solution:** explicitly signal to the provider that it is being used as a backup
  - using „well-known” BGP communities (communities neighbors previously agreed on)
  - set the BGP community on backup announcements

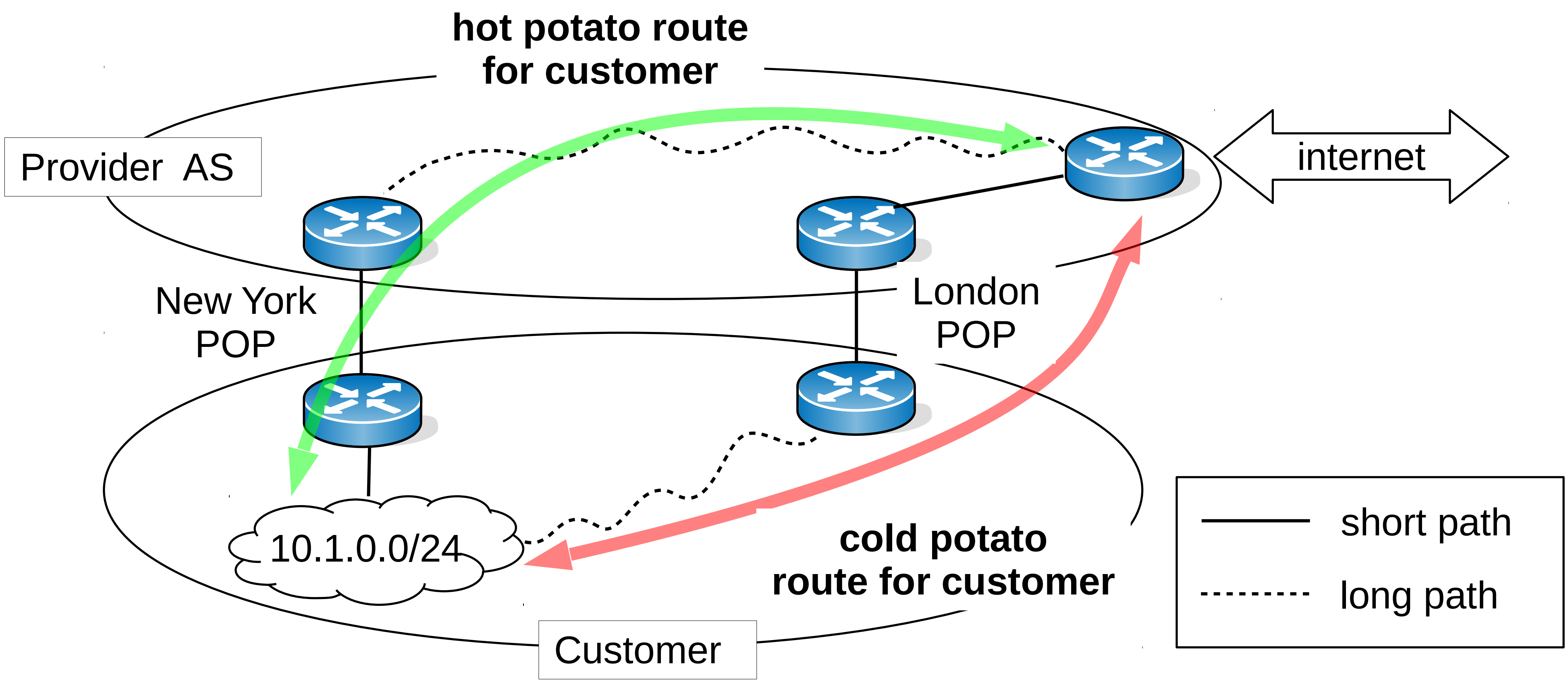
Hot potato routing

- **Hot potato routing:** a routing policy whereas packets are to leave an AS at the earnest
- Causes the least congestion/load **locally**



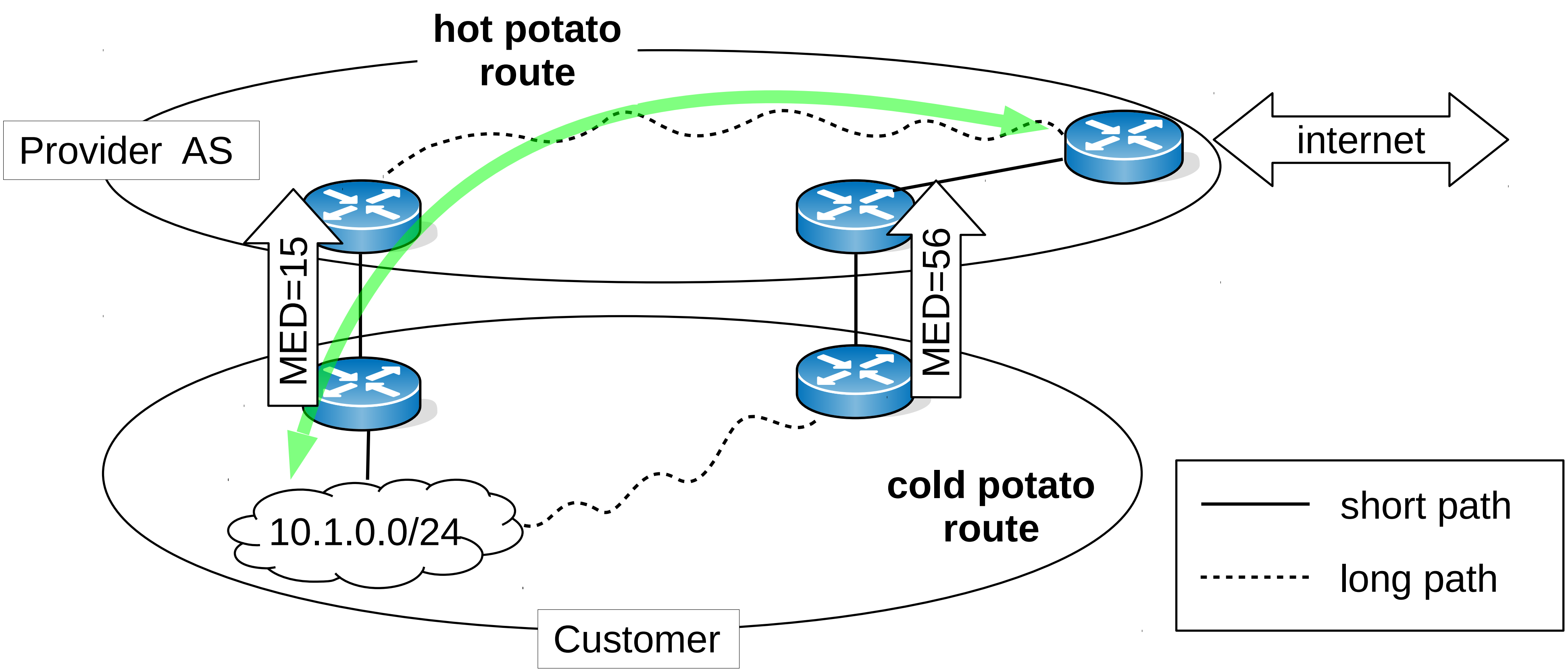
Hot potato routing

- What is a hot potato route for a customer is a cold potato route for the provider
- Needs mutual agreement who bears the costs



Hot potato routing

- Egress paths: iBGP/local-preference setting
- Ingress paths: Multi-Exit Discriminator (MED) BGP attribute (the lower the MED the better)

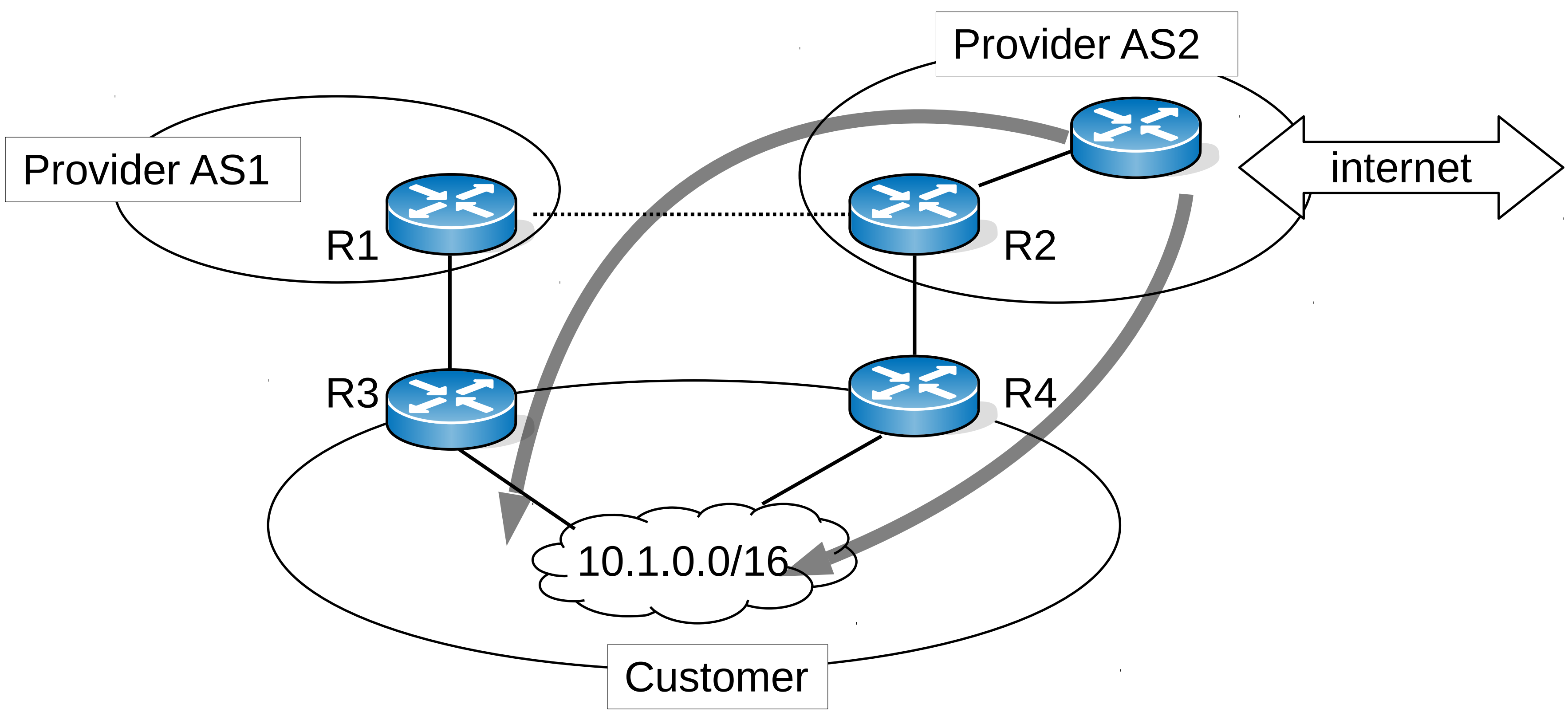


Ingress traffic engineering

- Instead of an all-or-nothing load distribution (all: primary, nothing: backup), balance traffic across providers equally
- Again, egress traffic is easy: set one provider as primary for half of the Internet prefixes and other as the backup, and vice versa for the other half
- Ingress direction is again more difficult (action-at-a-distance)
- **Ingress Traffic Engineering (TE):** influence remote ASes' routing decisions so that ingress traffic via different providers be split equally

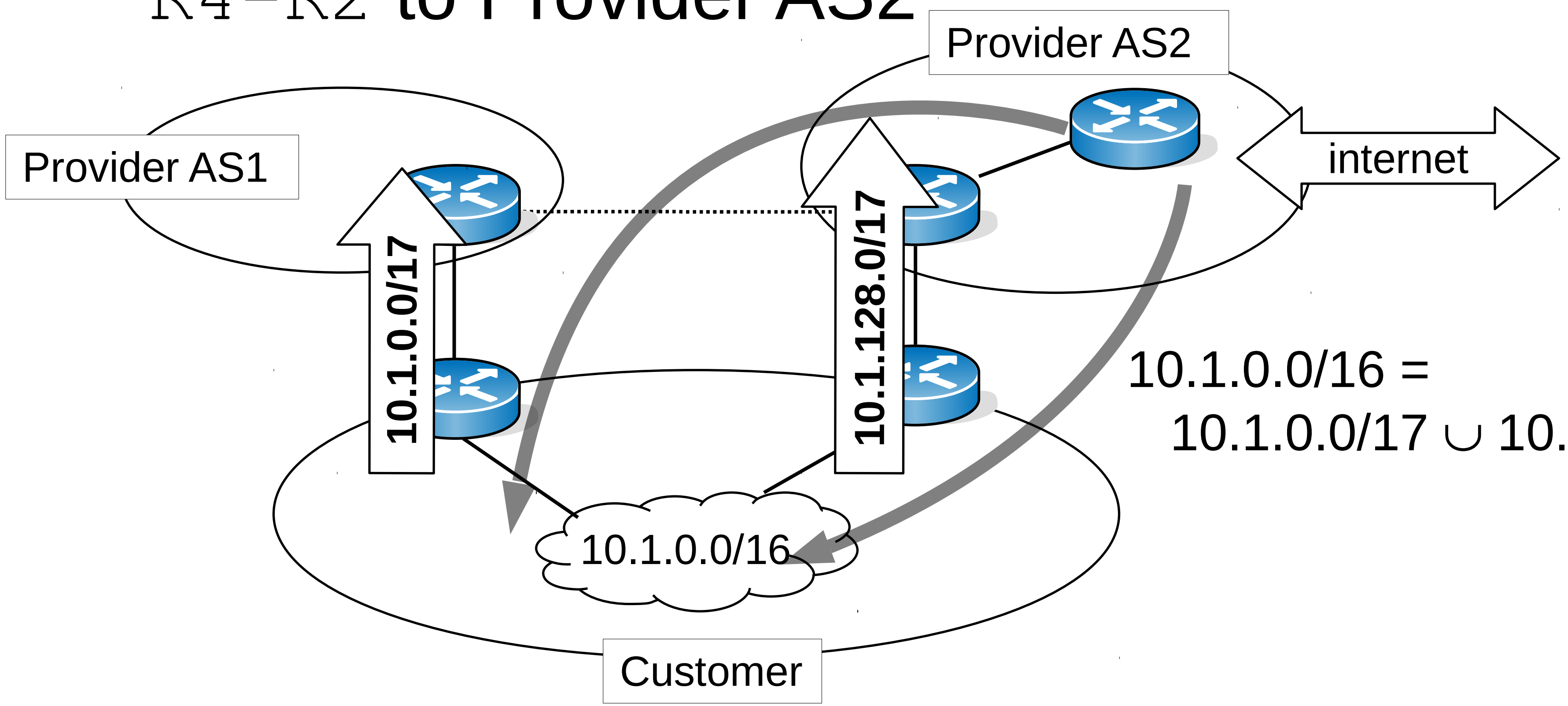
Ingress traffic engineering

- The customer wants to share ingress traffic roughly equally between AS-AS links `R1-R3` and `R2-R4`



Ingress traffic engineering

- **De-aggregation:** the customer splits its prefix into two, one subnet is announced on link `R3-R1` to `Provider AS1` and the other one on `R4-R2` to Provider AS2



10.1.0.0/16 =
10.1.0.0/17 ∪ 10.1.128.0/17

Ingress traffic engineering

- Subnets `10.1.0.0/17` and `10.1.128.0/17` hold the same number of IP addresses
- If they attract the same amount of traffic (ingress traffic uniformly distributed across the IP addresses)
- Then load is balanced equally
  - packets into `10.1.0.0/17` enter `Customer AS` at `R1`
  - packets into `10.1.128.0/17` enter on `R2`
- **But de-aggregation increases FIB size at every router on the Internet** (2 prefixes instead of 1)!
  - about 5–10% of routed prefixes are de-aggregates
  - a major cause of Internet unscalability