A Hierarchical Model for BGP Routing Policies

Laurent Vanbever, Bruno Quoitin and Olivier Bonaventure

UCL, Belgium

PRESTO'09

Friday, 21 Aug 2009

http://inl.info.ucl.ac.be

Human factors are responsible for ***50*** to ***80*** percent of network device outages

Juniper Networks, *What's Behind Network Downtime?*, 2008

A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion

A Hierarchical Model for BGP Routing Policies

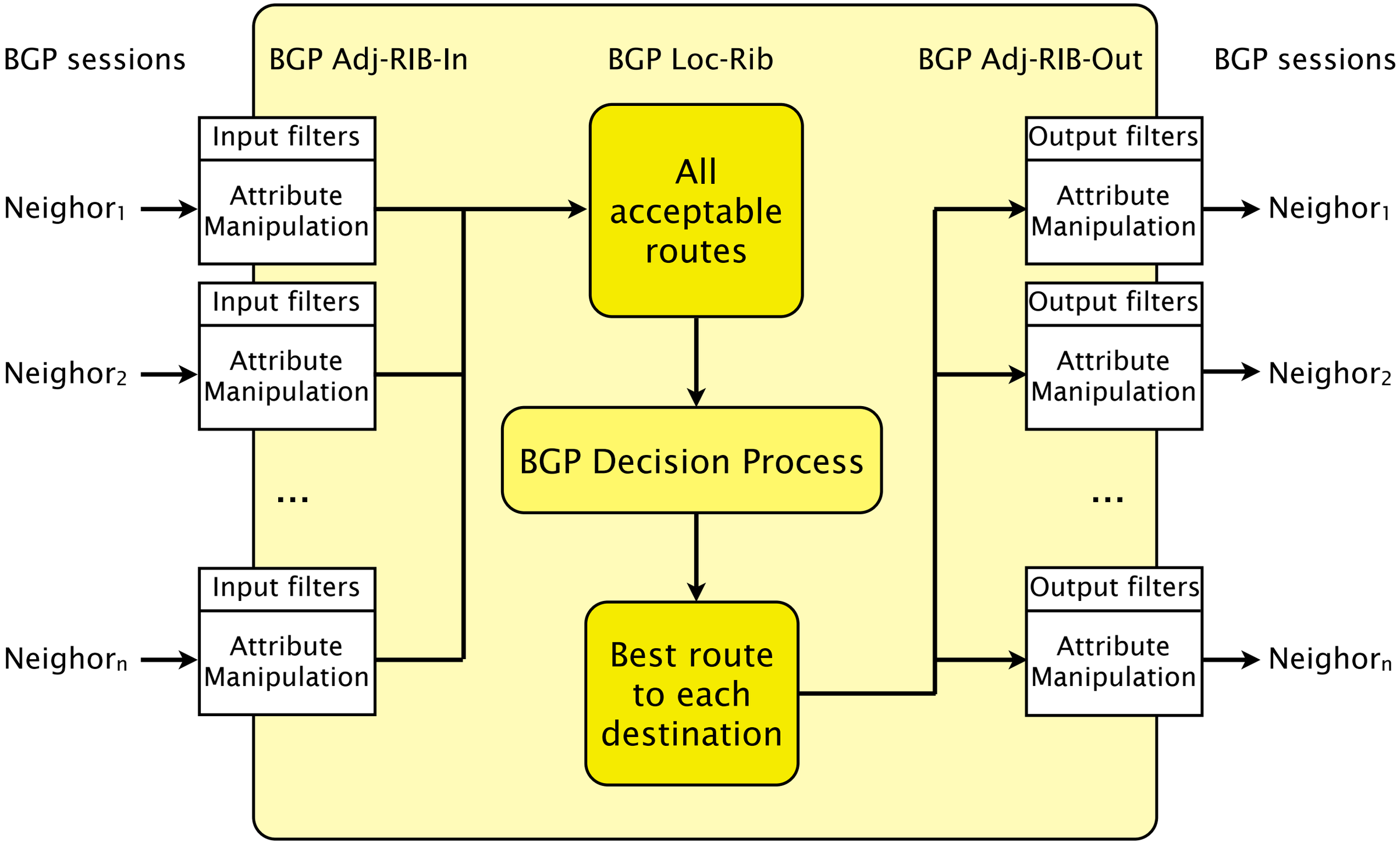
Introduction and Motivation

Towards a *hierarchical* model of routing policies

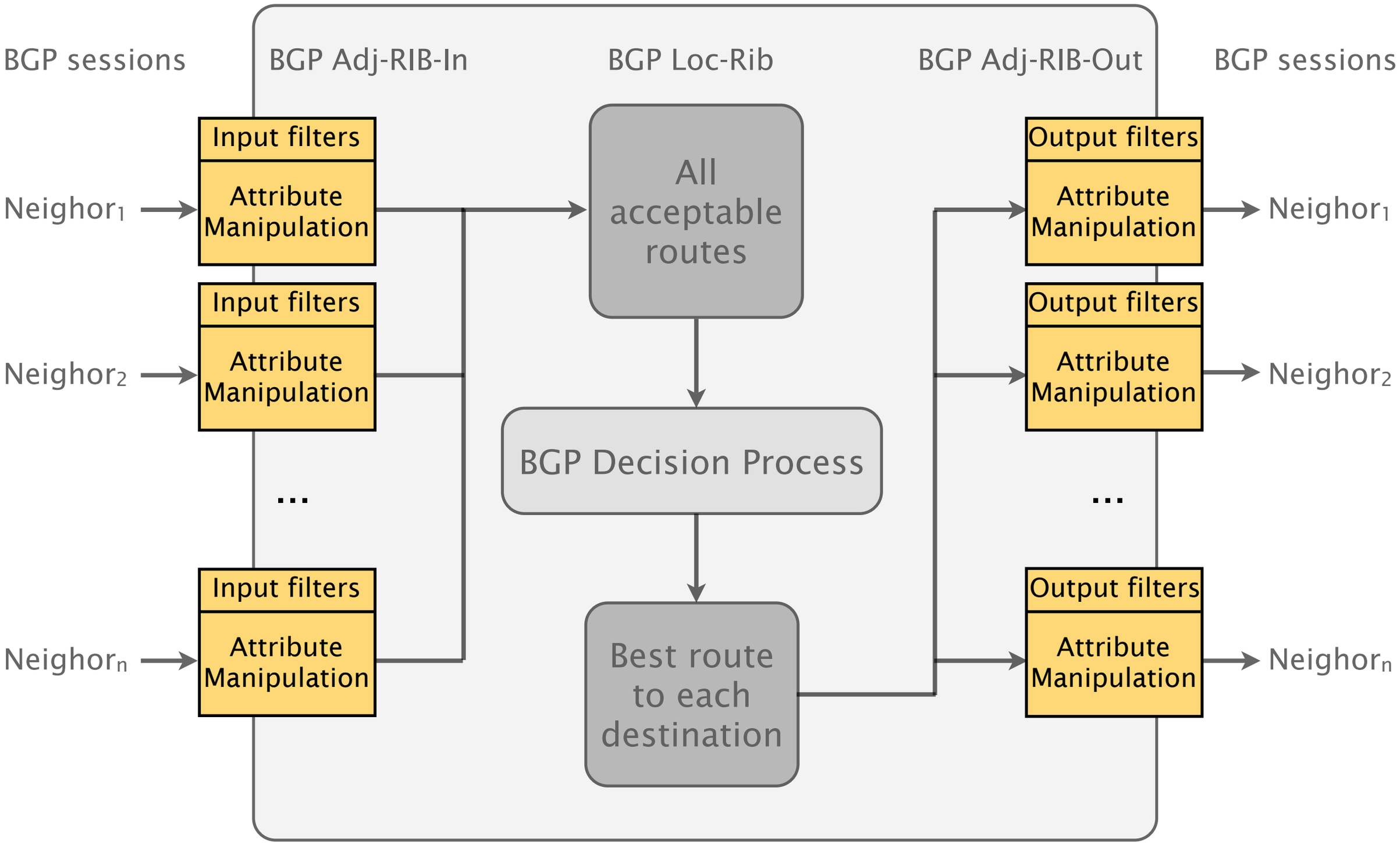
Implementation

Conclusion

A BGP Router at a Glance



Talk is about BGP Policies

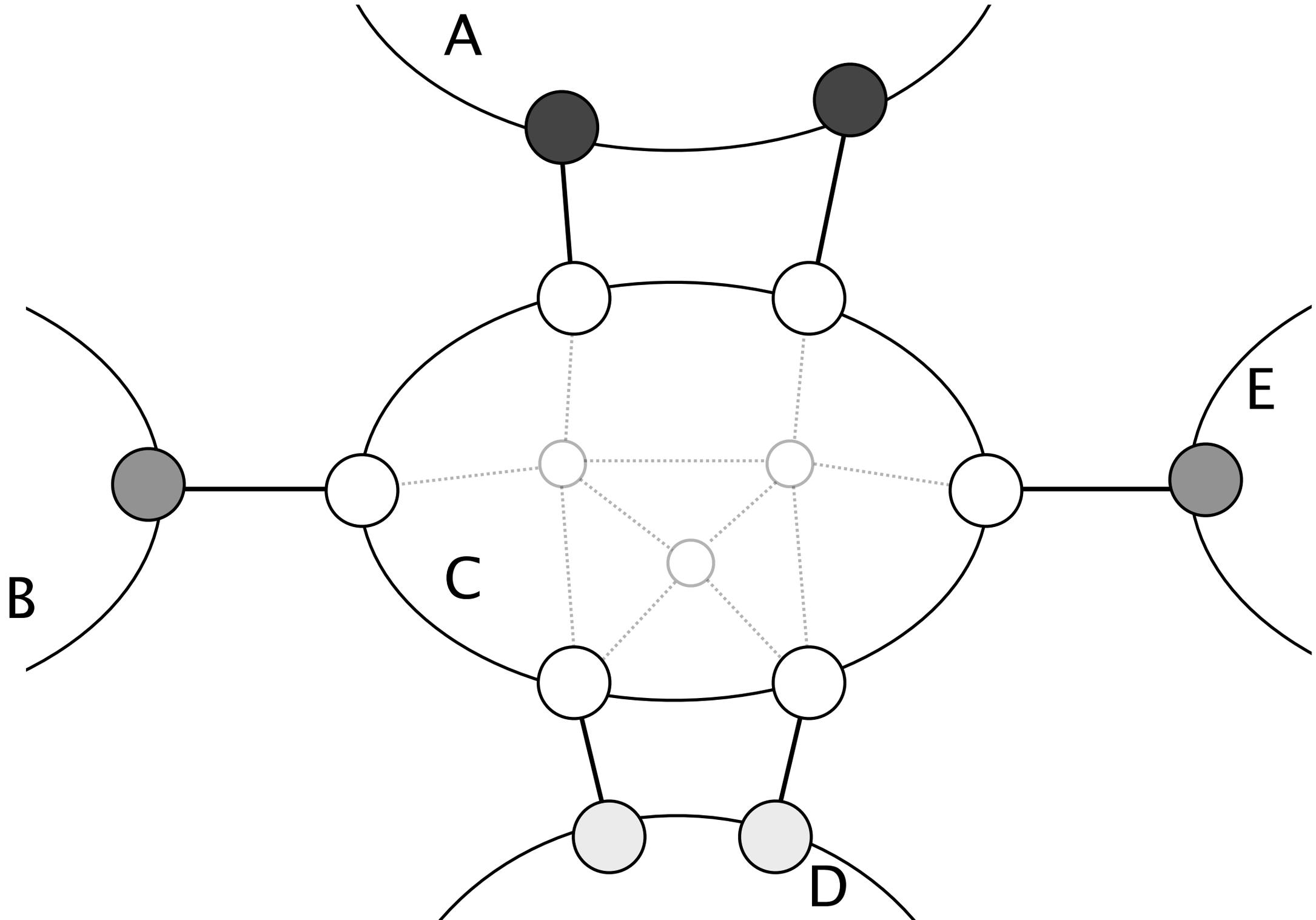


BGP Policies give operators control over routes selection

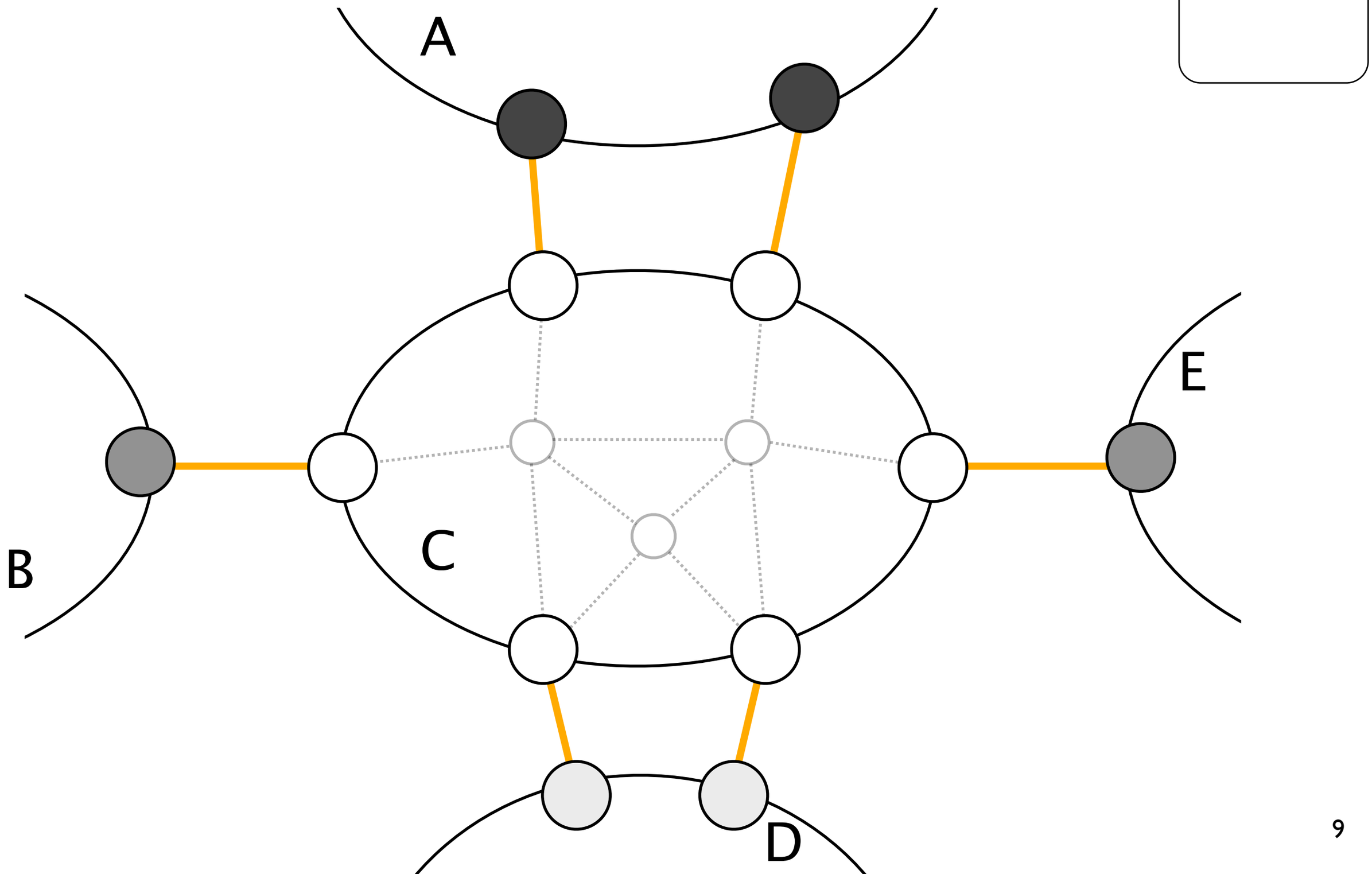
Policies are mainly used to

- filter incoming routes
  - *ignore* routes you don't want to consider
- modify routes' attributes
  - *influence* path selection
  - *modify* the way routes are perceived
- filter outgoing routes
  - *enforce* business relationships

BGP Policies are defined at different *abstraction* levels

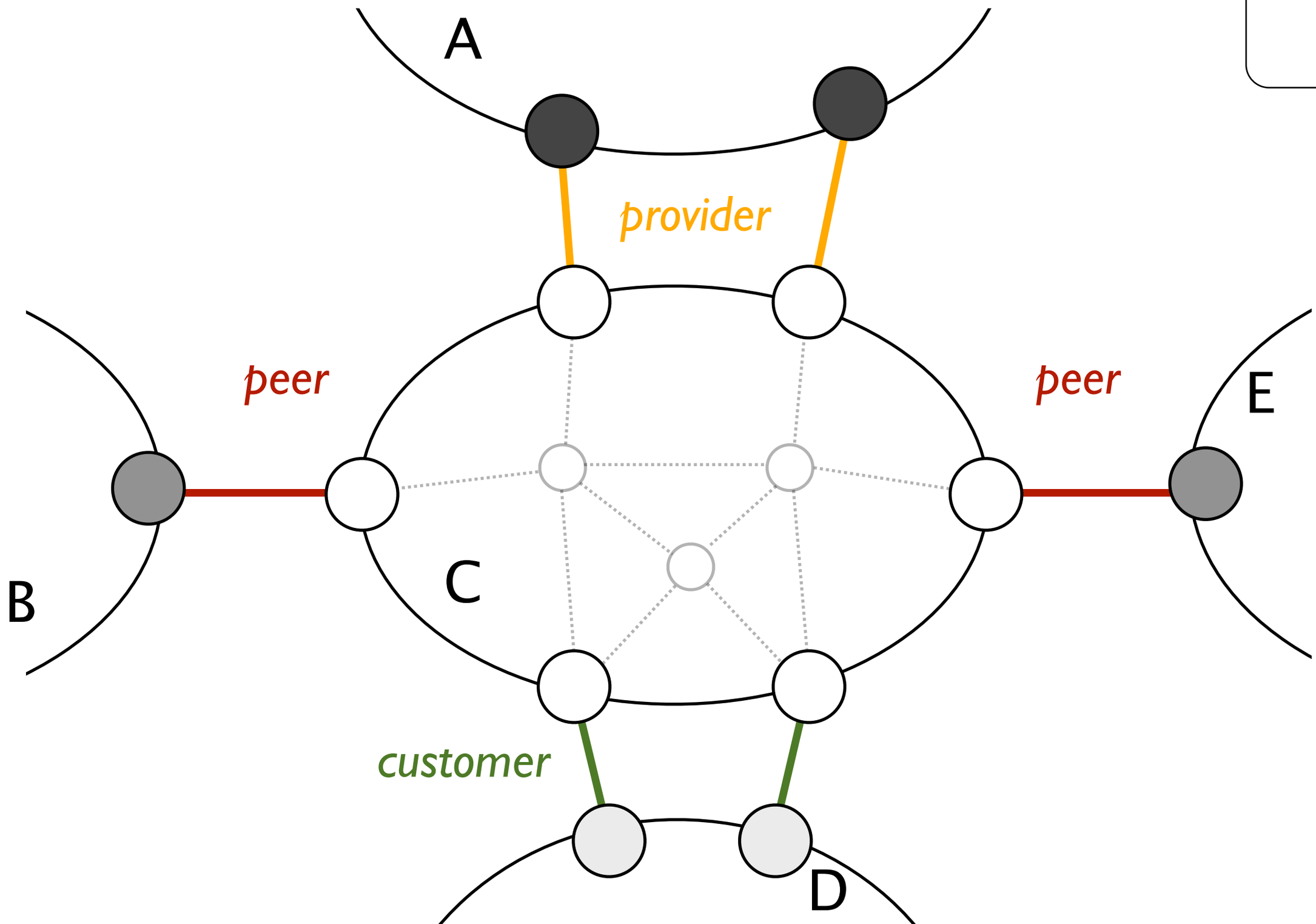


Some BGP Policies are defined on *all* sessions



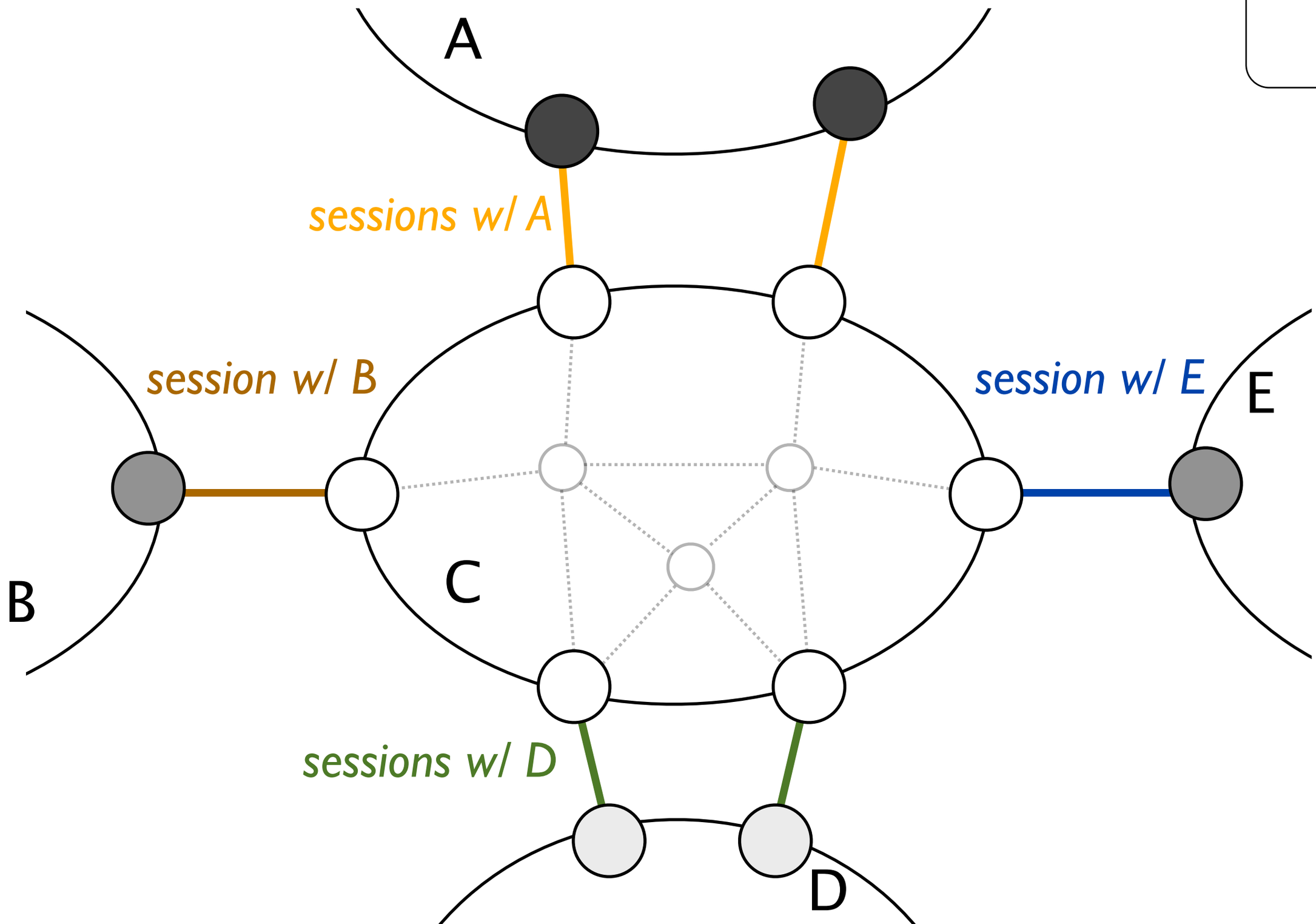
Some BGP Policies are defined on *groups* of sessions

- all
- group



Some BGP Policies are defined on *AS* sessions

- all
- group
- as



Some BGP Policies are defined on *invidivual* sessions

- all
- group
- as
- session

A

sessions w/ RA_1

sessions w/ RA_2

B

C

E

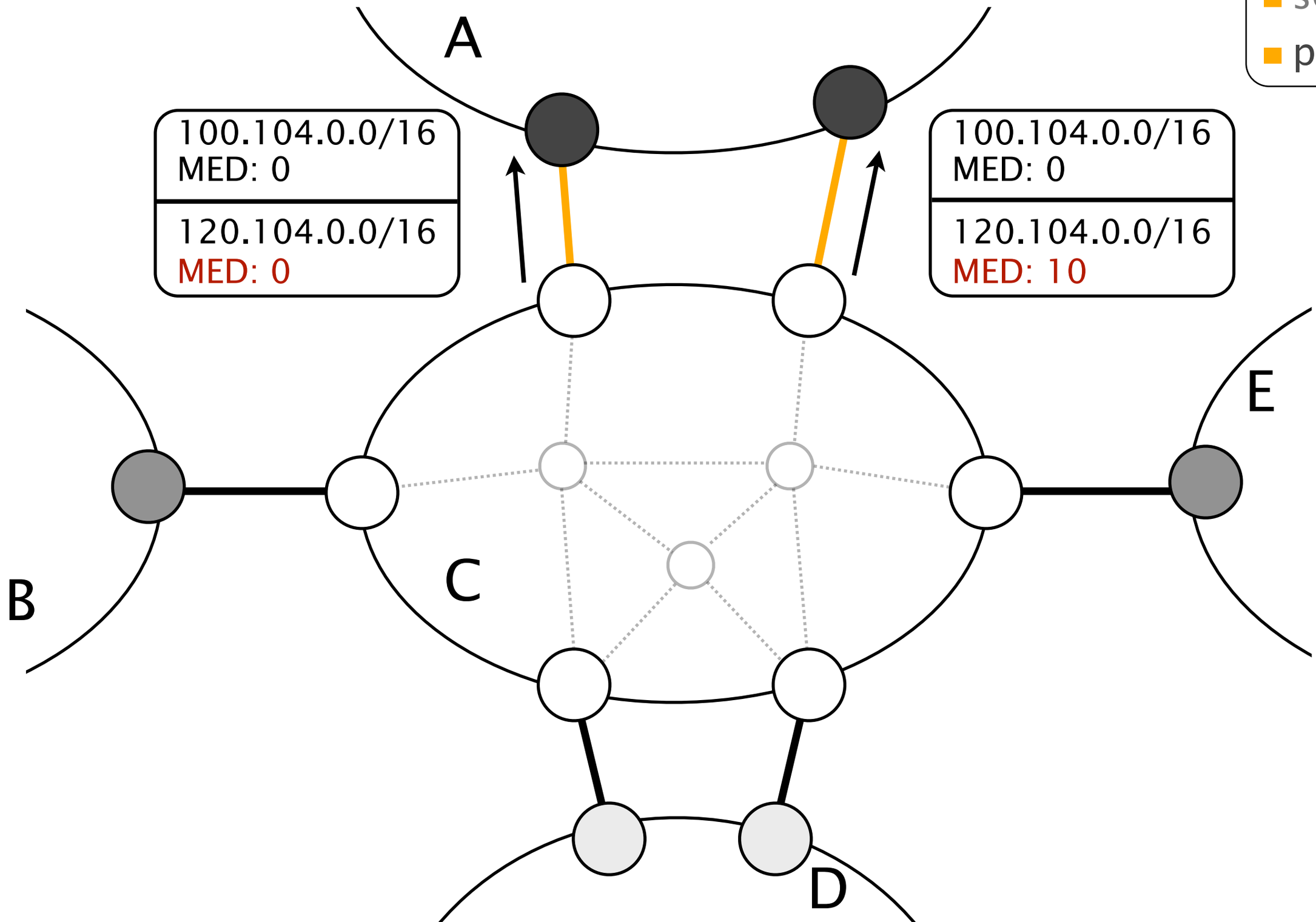
sessions w/ RD_1

sessions w/ RD_2

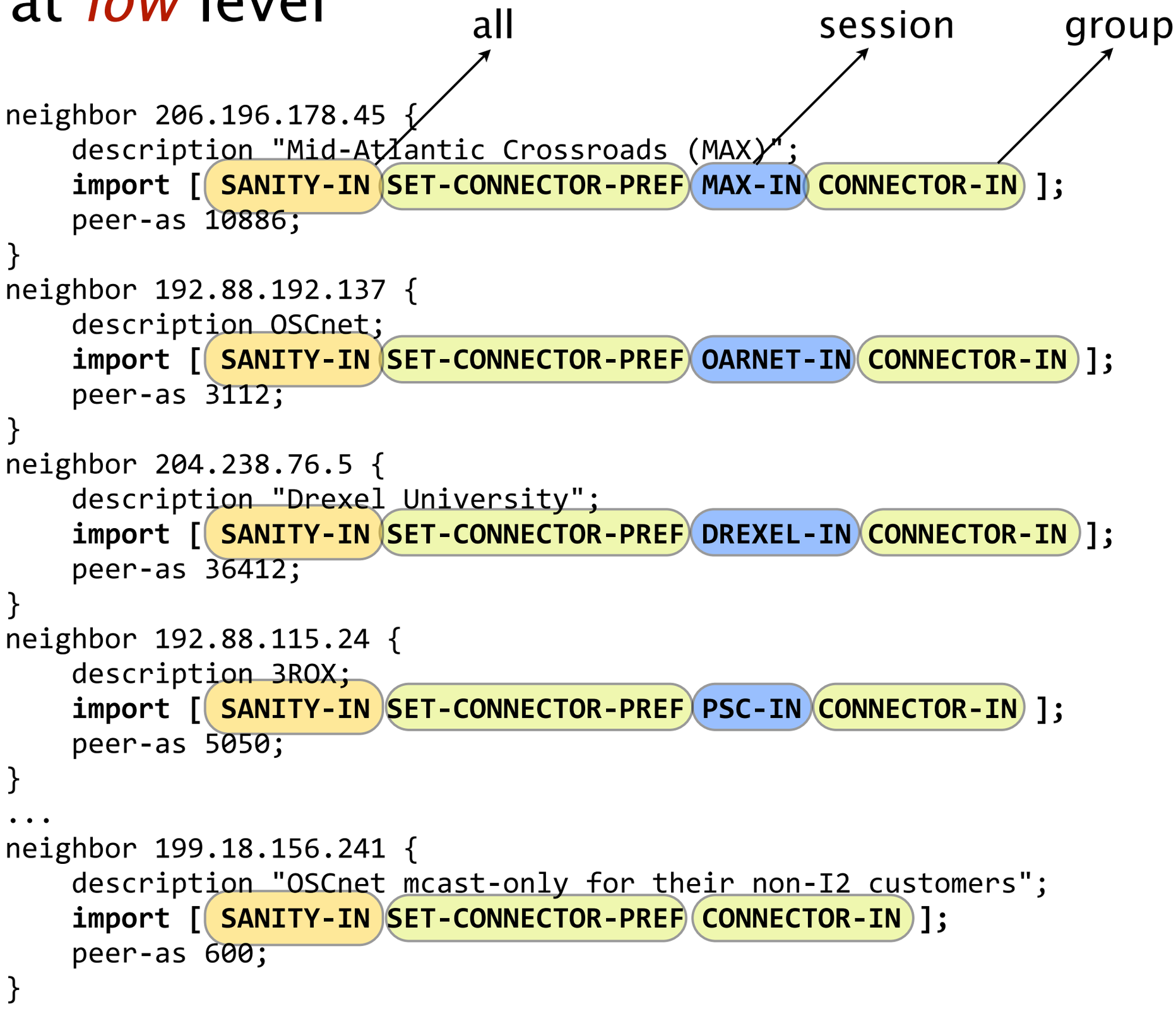
D

Some BGP Policies are defined on *prefixes*

- all
- group
- as
- session
- prefix



However, policies are often defined at *low* level



A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a ***hierarchical*** model of routing policies

Implementation

Conclusion

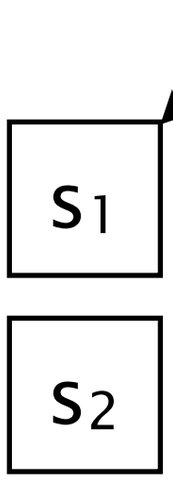
Towards a *hierarchical* model of routing policies

Our model aims to

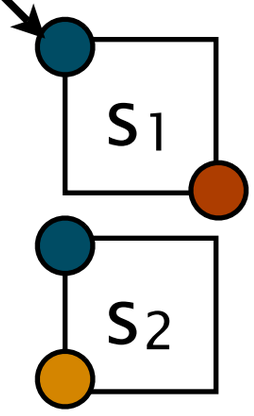
- express a policy at the appropriate level
- represent *network-wide* policies
- ease policy addition and modification
- be vendor agnostic

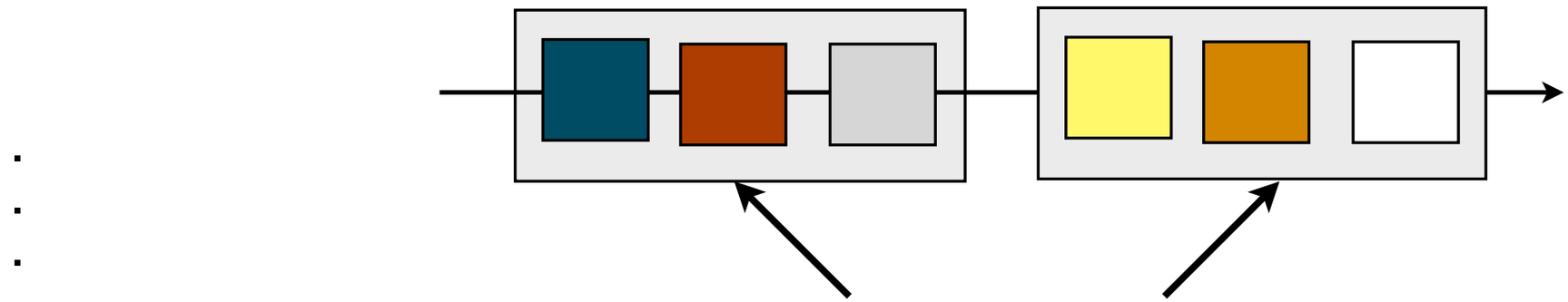
Our model at a glance

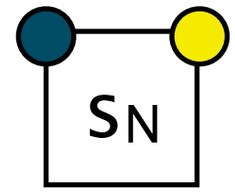
BGP Session's abstraction



BGP Session's abstraction
associated to templates





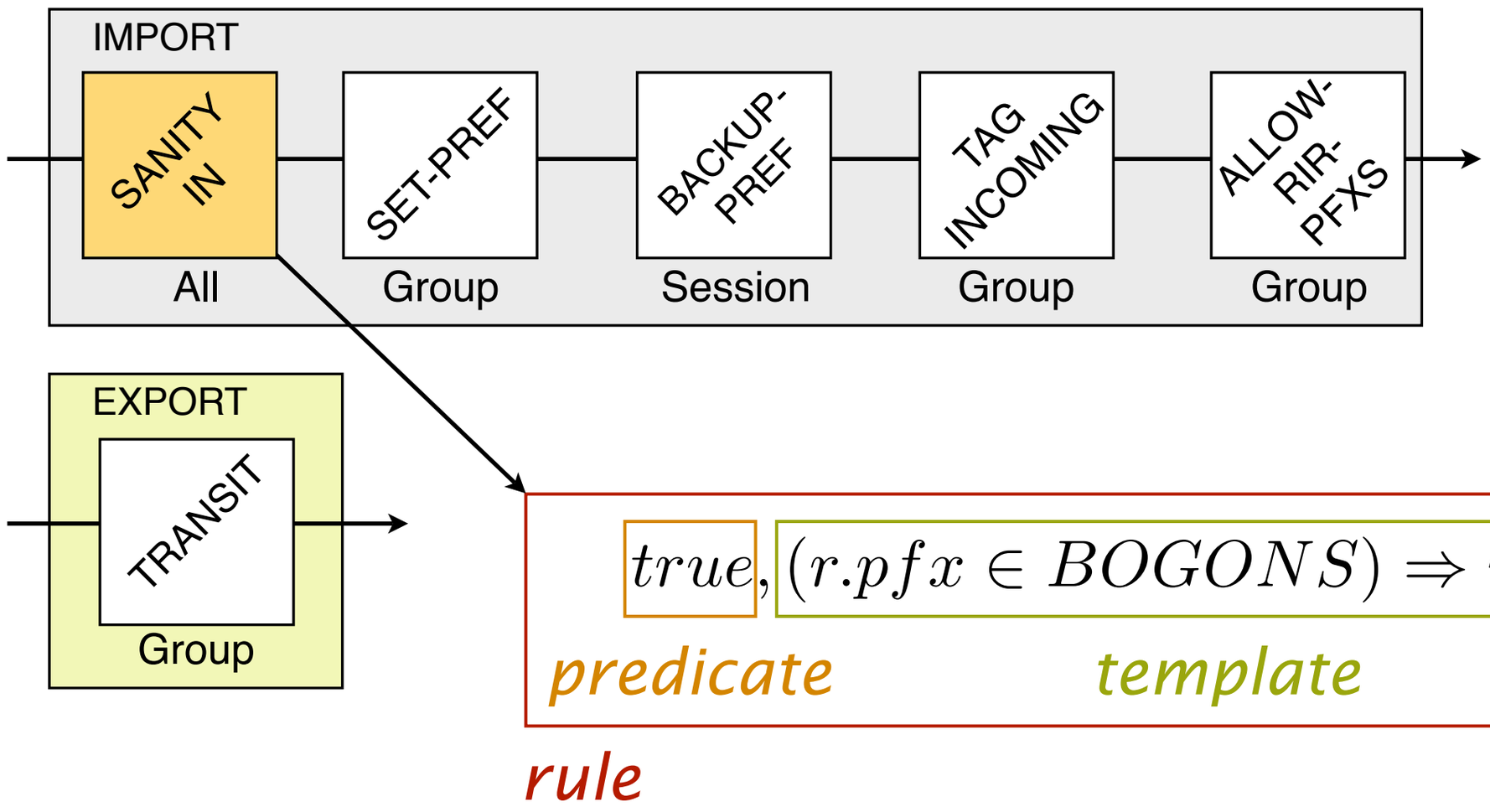


Our model is structured around *chains* of filters

Policies are modeled by chains

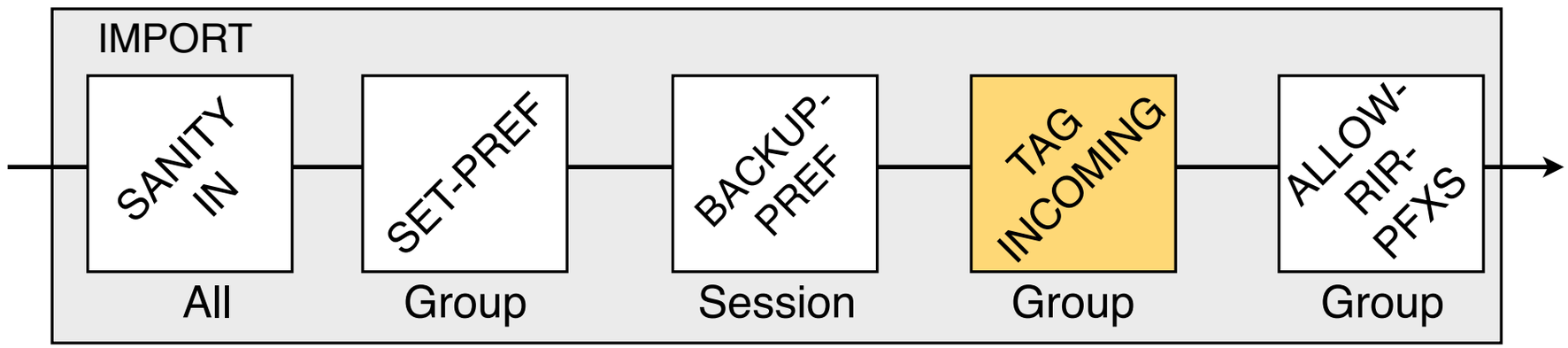
- a node is a sequence of *rules*
- a *rule* is a couple (*predicate*,*template*)
- a *predicate* conditions the association of the template to the session's filters
- a *template* is a sequence of routing filters statements

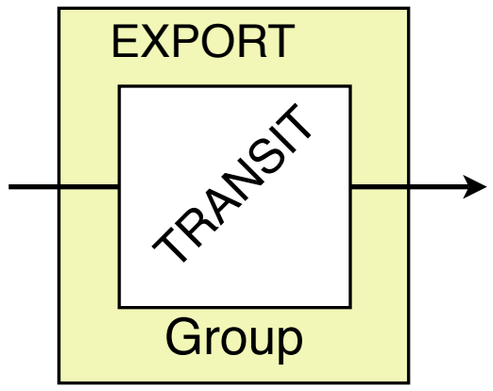
Our model is structured around *chains* of filters



s : session
r : route

Our model is structured around *chains* of filters





$s.group = CUST$,	$r.comm \uplus \{CUST\}$
$s.group = PEER$,	$r.comm \uplus \{PEER\}$
$s.group = PROV$,	$r.comm \uplus \{PROV\}$

s : session
r : route

A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

Conclusion

How is it implemented ?

- Each BGP session is specified with a *textual representation*

  `BXL:CUST:2611:<130.104.0.2>:backup`

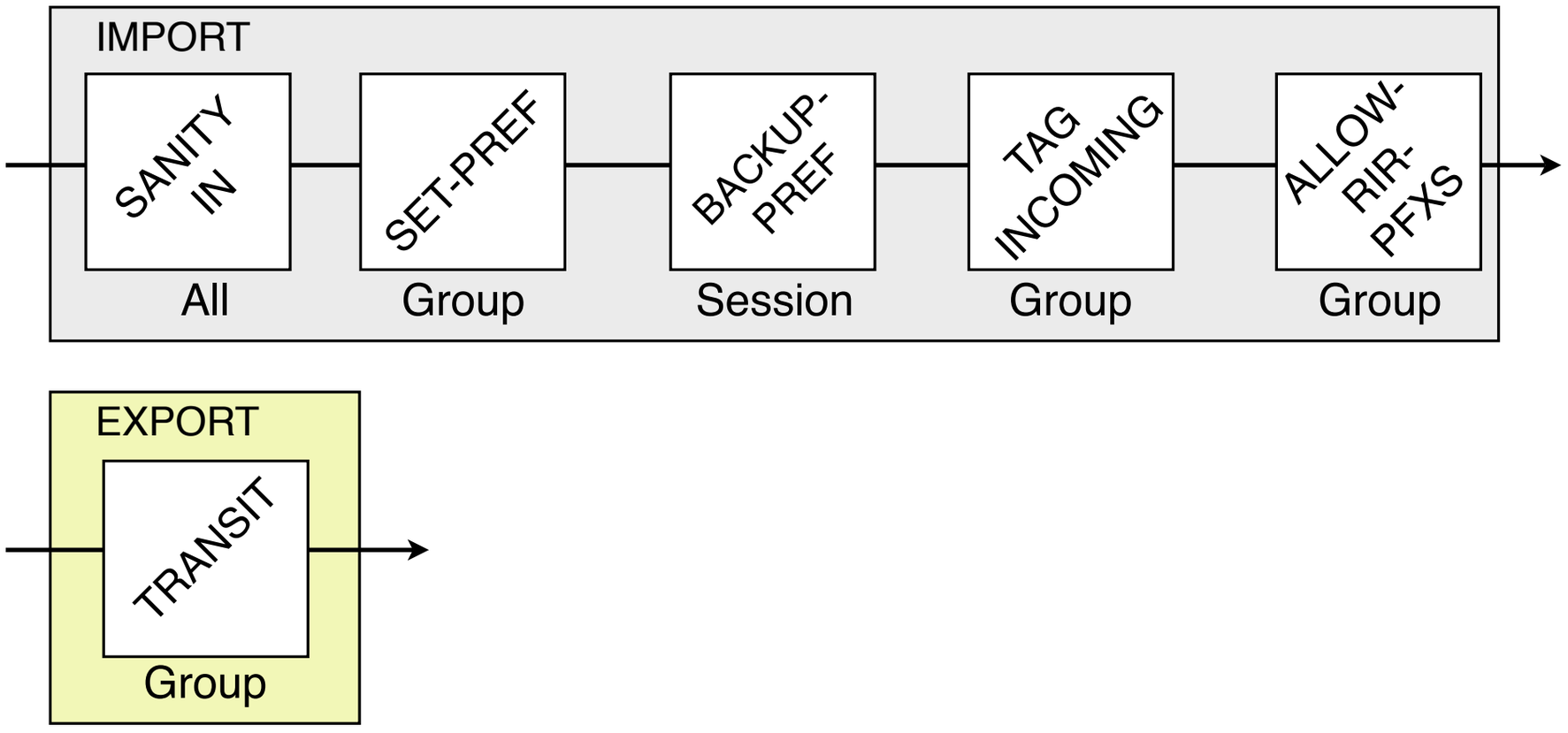
- Predicates are modeled by *regular expressions*

  `s.type=backup modeled *.backup$`

- Templates are represented by using *StringTemplate*

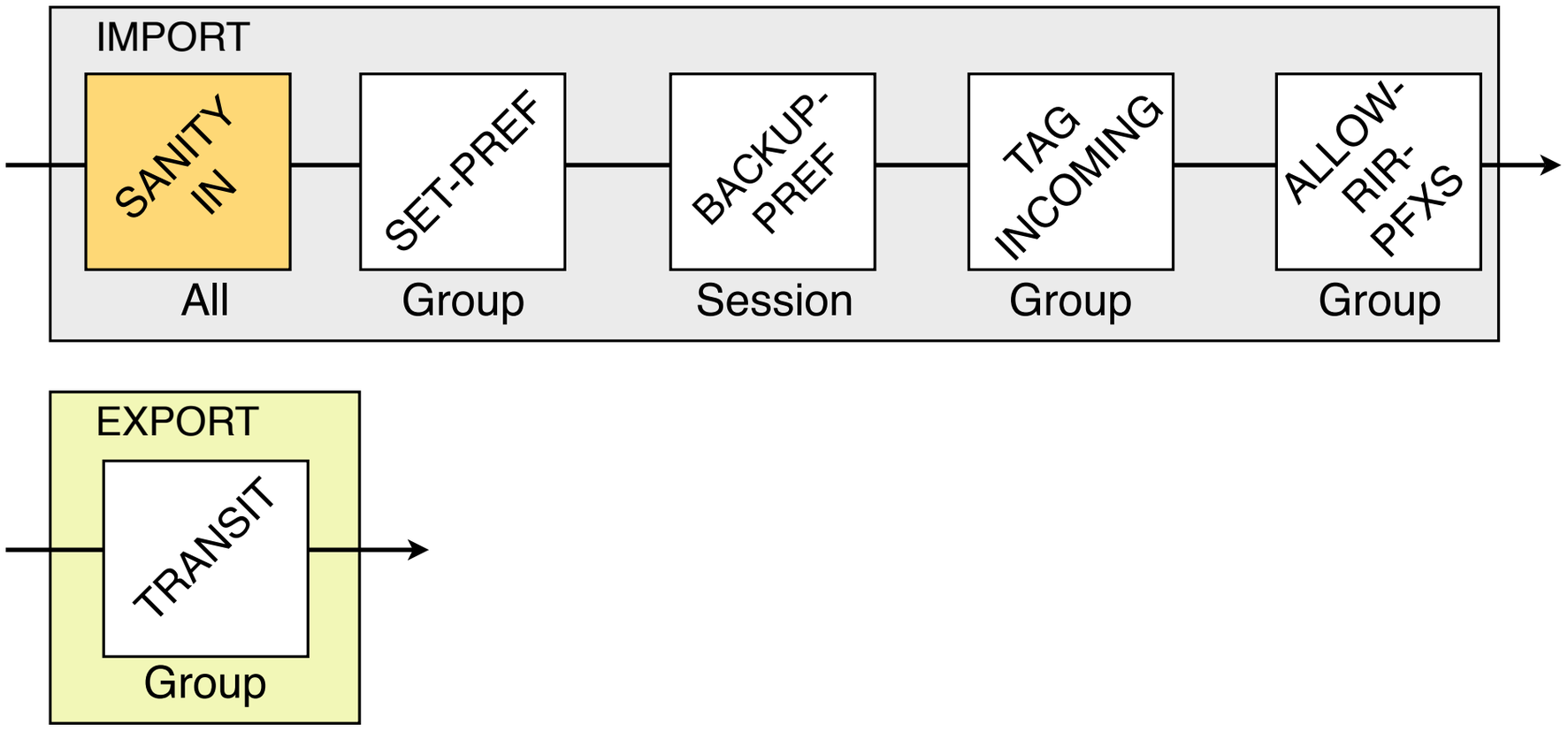
```
policy-statement BACKUP-PREF {
  term down-pref {
    then {
      local-preference subtract $value$;
      accept;
...
```

How does it work ?




```
BXL:CUST:2611:<130.104.0.2>:backup
```

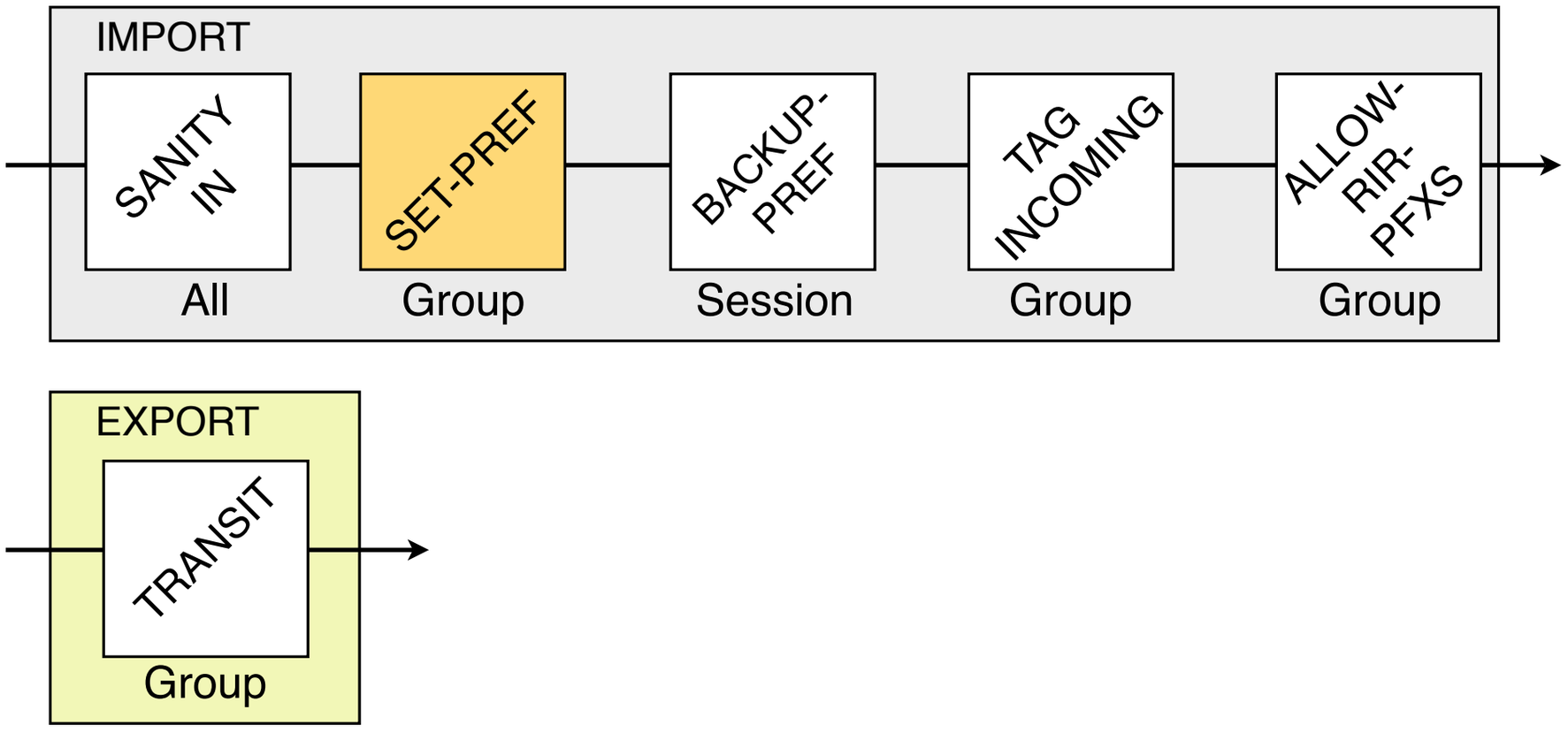
How does it work ?



BXL:CUST:2611:<130.104.0.2>:backup

$$(r.pfx \in BOGONS) \Rightarrow reject$$

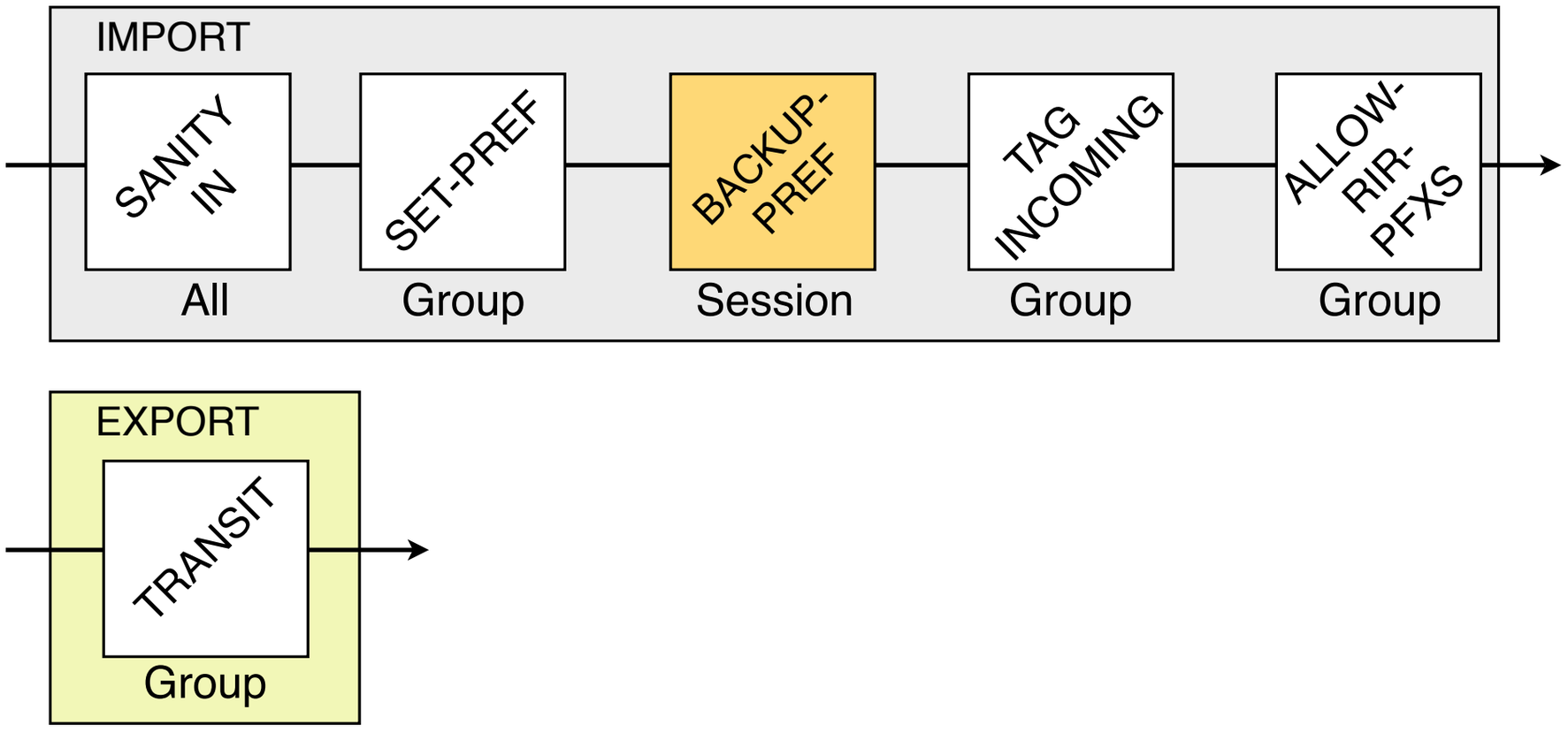
How does it work ?



BXL:**CUST**:2611:<130.104.0.2>:backup

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 5000$

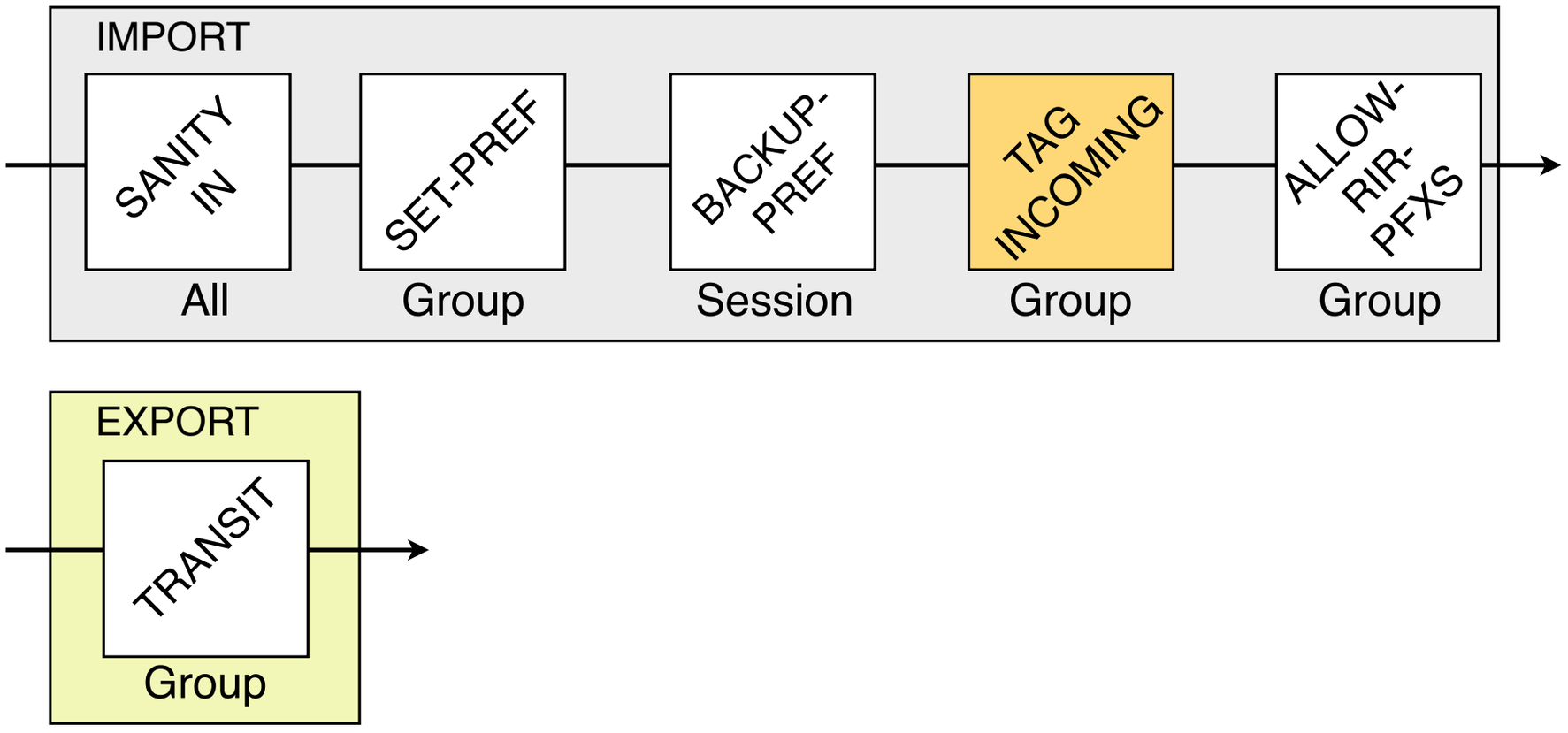
How does it work ?



BXL:CUST:2611:<130.104.0.2>:**backup**

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 5000$
$r.lp = r.lp - 500$

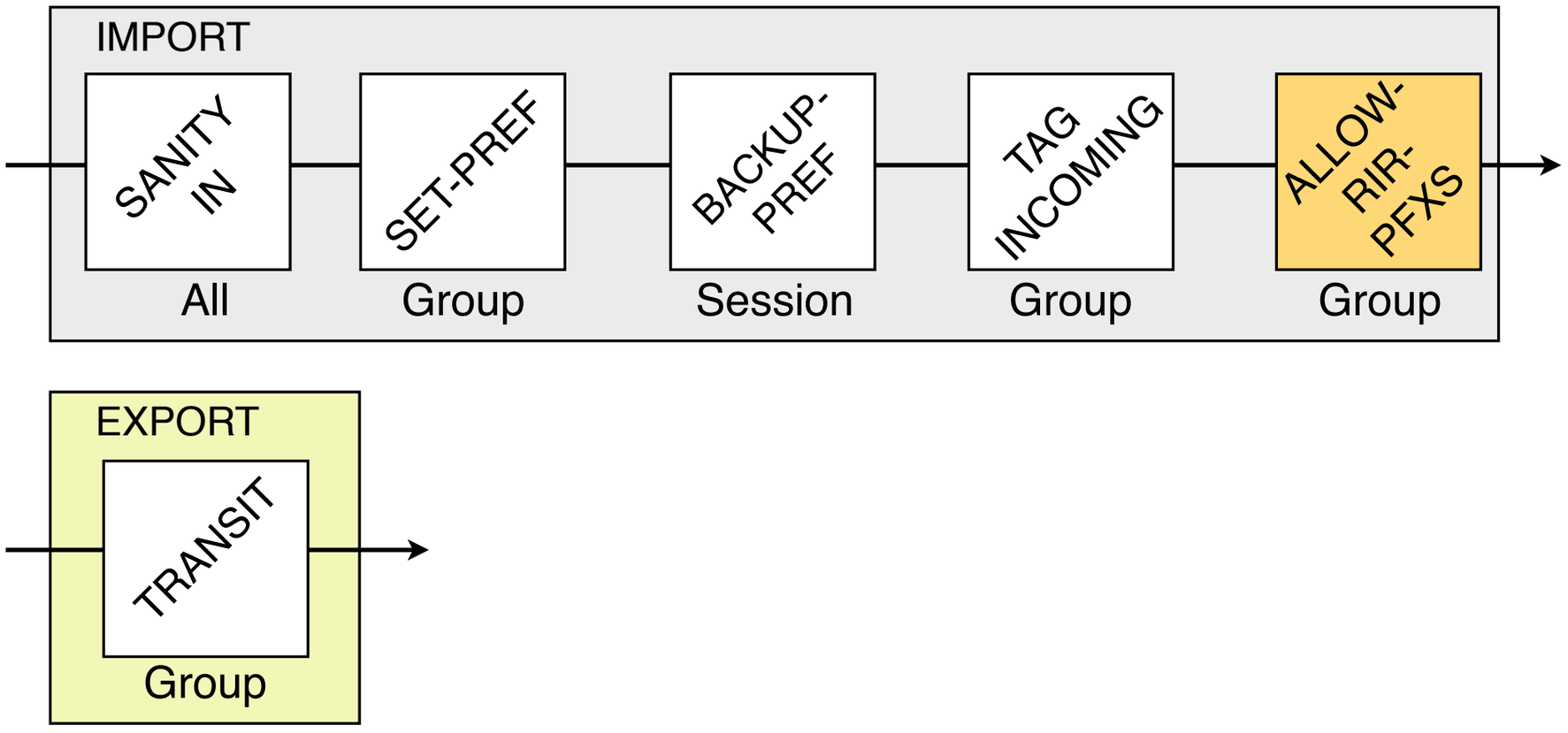
How does it work ?



`BXL:CUST:2611:<130.104.0.2>:backup`

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 5000$
$r.lp = r.lp - 500$
$r.comm \uplus \{CUST\}$

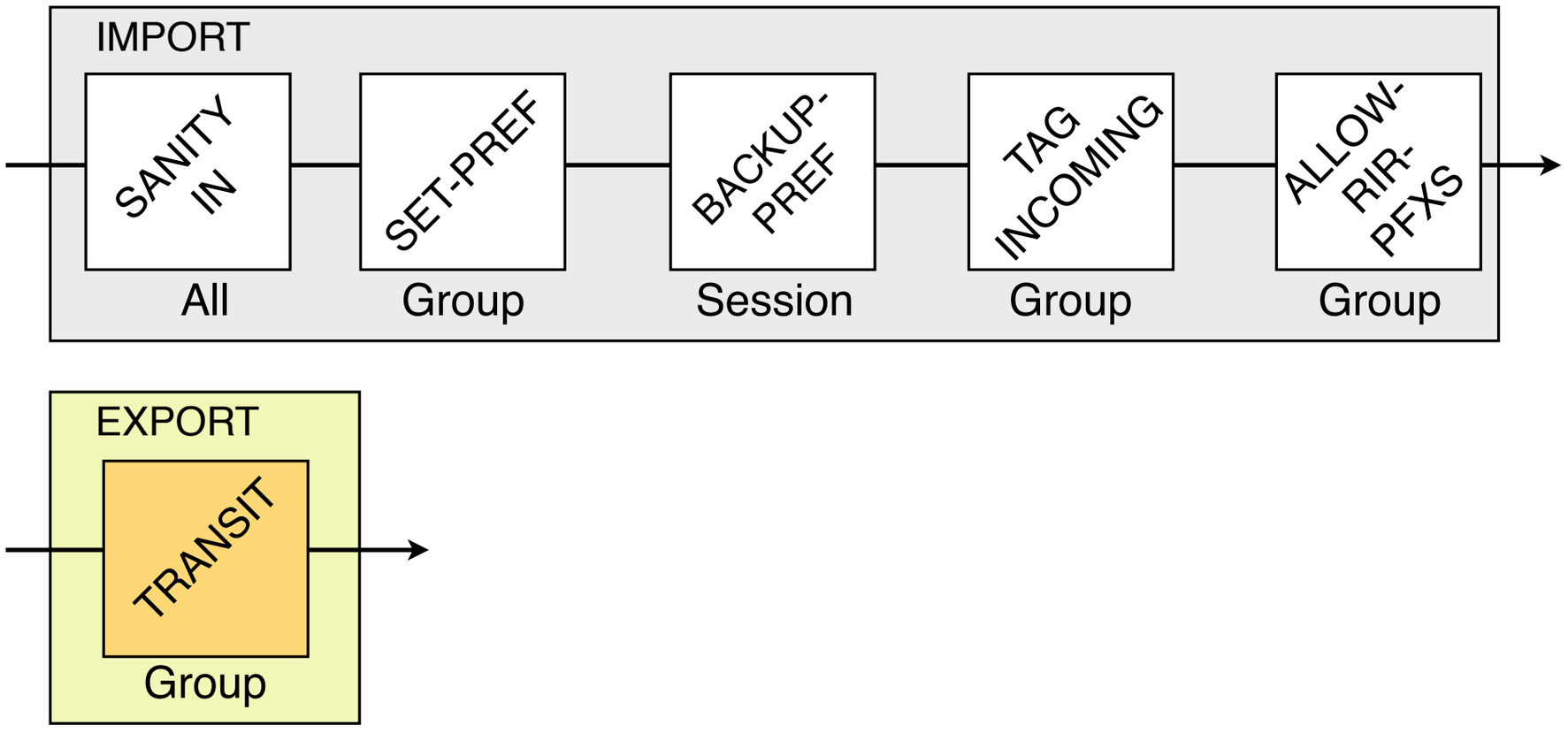
How does it work ?



BXL:**CUST**:2611:<130.104.0.2>:backup

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 5000$
$r.lp = r.lp - 500$
$r.comm \uplus \{CUST\}$
$(r.pfx \notin RIR\_PFX(s.asn)) \Rightarrow reject$

How does it work ?

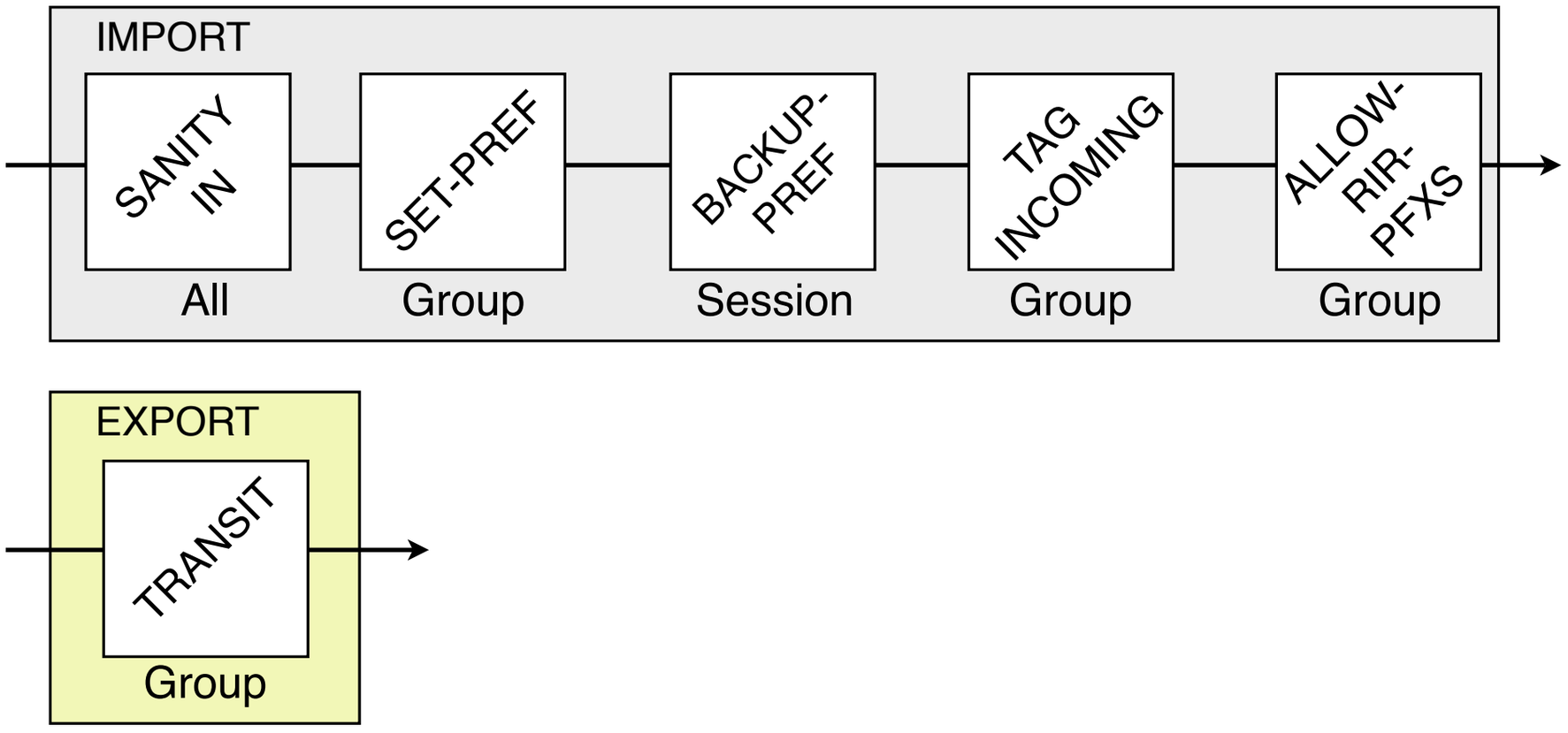


`BXL:CUST:2611:<130.104.0.2>:backup`

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 5000$
$r.lp = r.lp - 500$
$r.comm \uplus \{CUST\}$
$(r.pfx \notin RIR\_PFX(s.asn)) \Rightarrow reject$

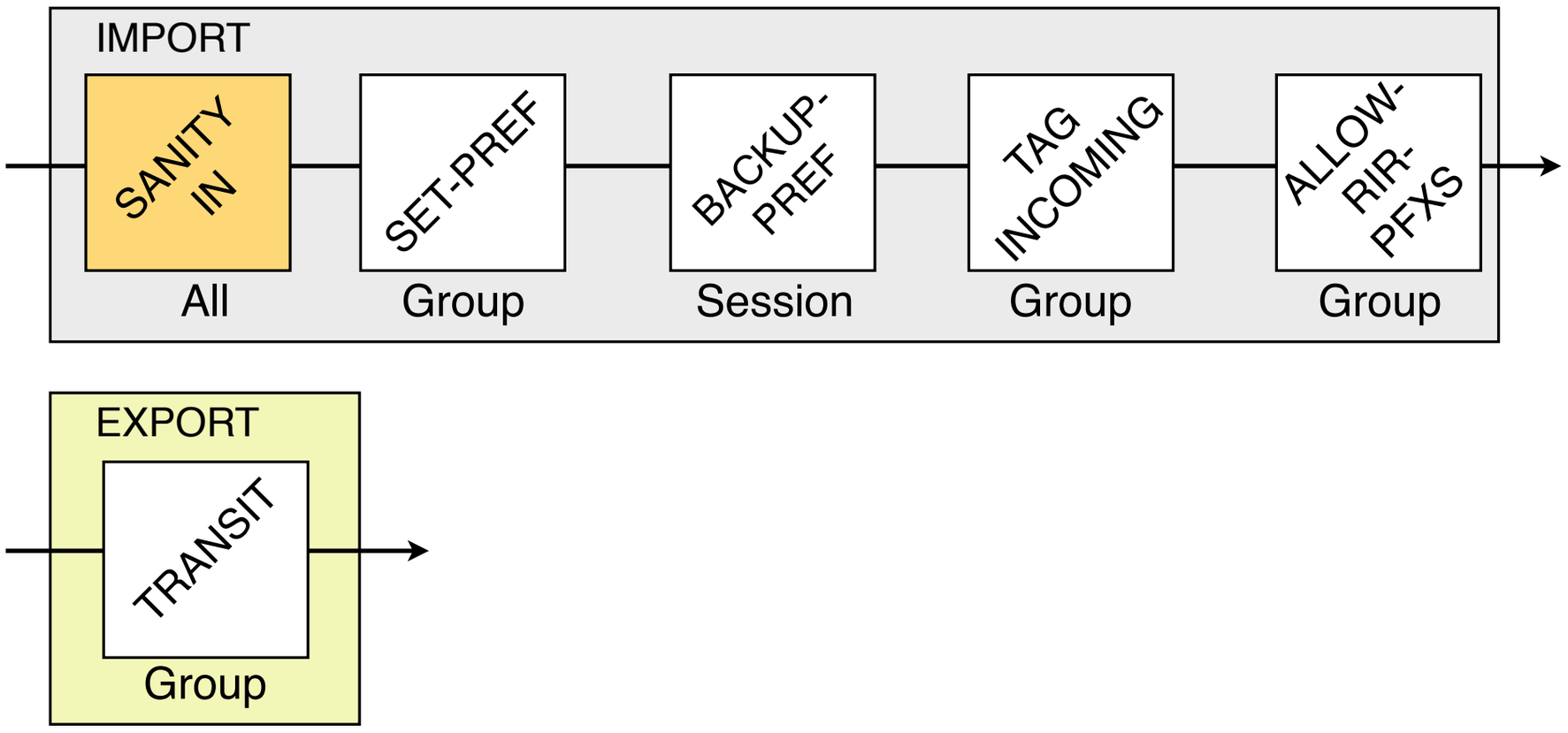
$announce\ default\ route$

How does it work ?



BXL:PROVIDER:2611:<130.104.0.2>

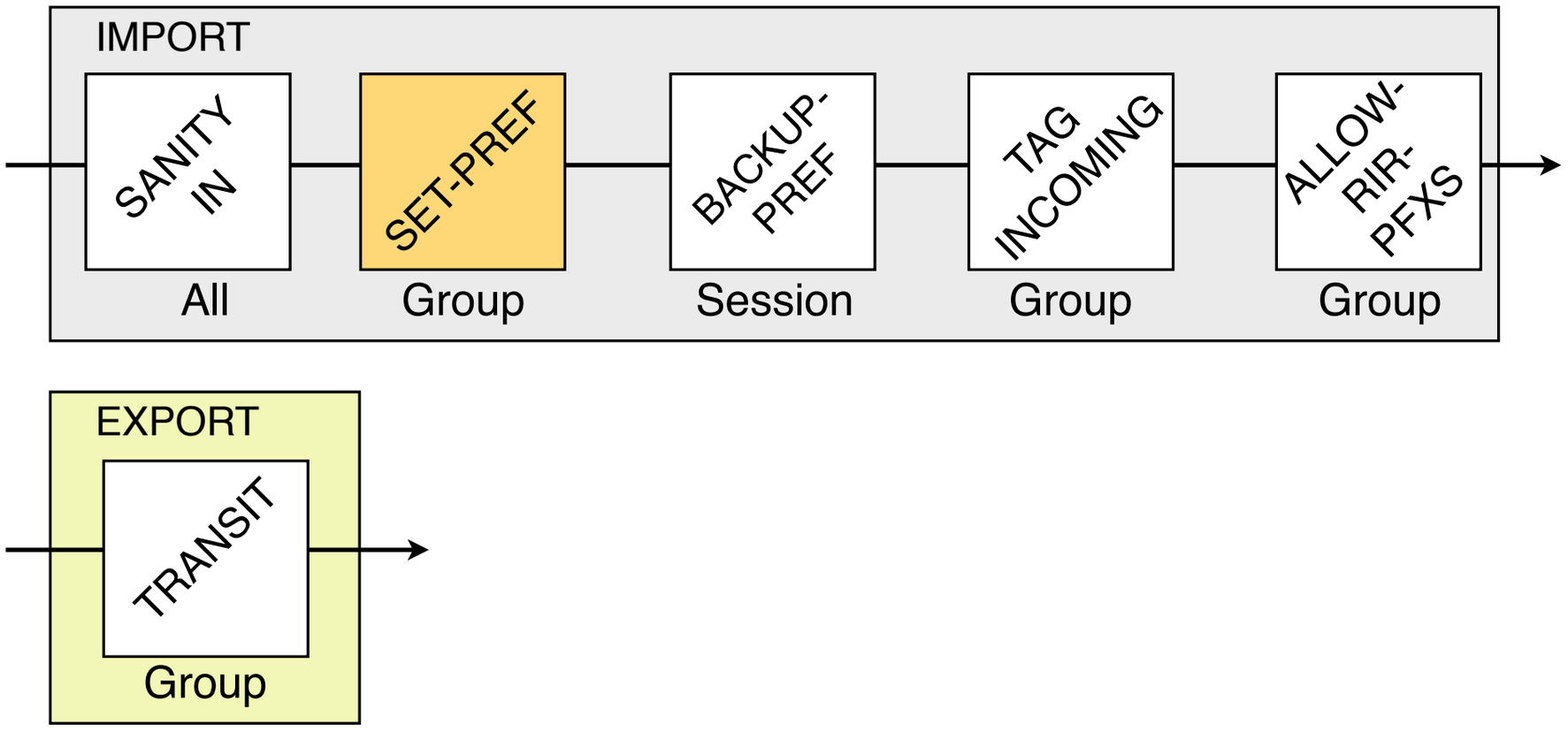
How does it work ?



BXL:PROVIDER:2611:<130.104.0.2>

$$(r.pfx \in BOGONS) \Rightarrow reject$$

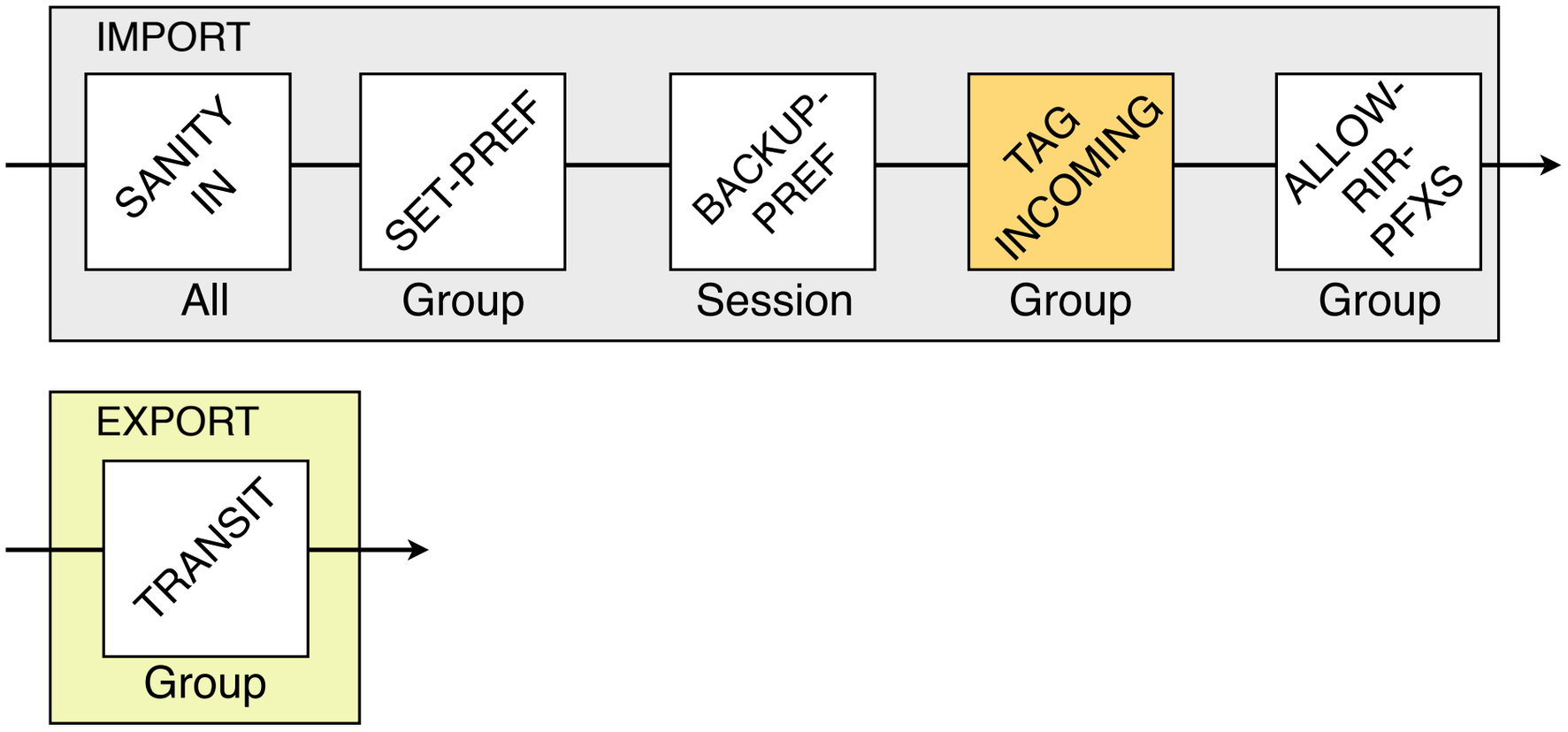
How does it work ?



BXL:**PROVIDER**:2611:<130.104.0.2>

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 3000$

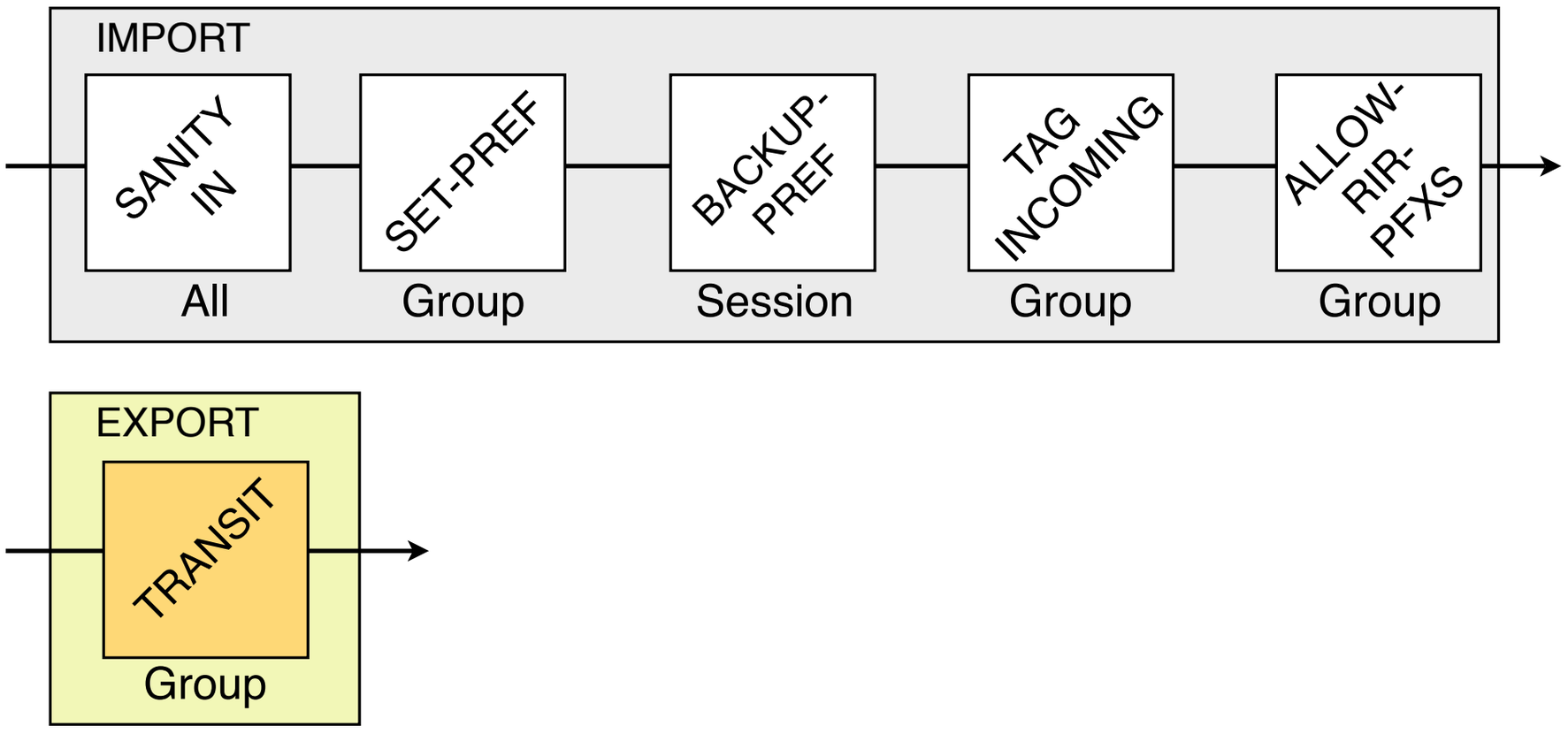
How does it work ?



BXL:**PROVIDER**:2611:<130.104.0.2>

$$(r.pfx \in BOGONS) \Rightarrow reject$$
$$r.lp = 3000$$
$$r.comm \uplus \{PROV\}$$

How does it work ?



BXL:**PROVIDER**:2611:<130.104.0.2>

$(r.pfx \in BOGONS) \Rightarrow reject$
$r.lp = 3000$
$r.comm \uplus \{PROV\}$

$((r.comm \ni CUST) \vee (r.pfx \in INTERNAL)) \Rightarrow accept$

A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation and Evaluation

Conclusion

To Conclude

Our model offers

- a *network-wide* and *vendor-agnostic* way of configuring routing policies
- detailed documentation
- quick and safe modifications/additions

A Hierarchical Model for BGP Routing Policies

Laurent Vanbever, Bruno Quoitin and Olivier Bonaventure

UCL, Belgium

Questions ?

PRESTO'09

Friday, 21 Aug 2009

http://inl.info.ucl.ac.be