

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

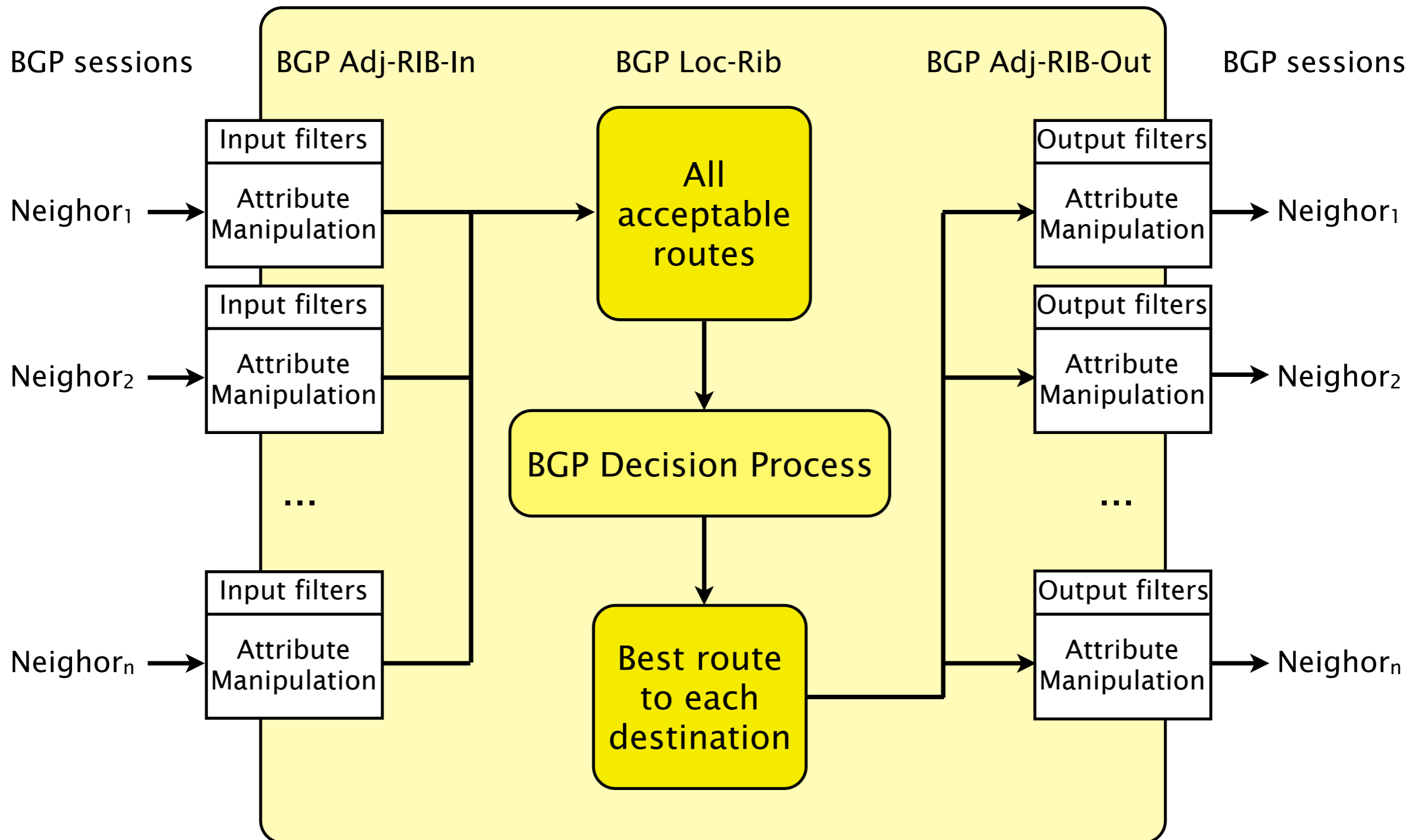
Introduction and Motivation

Towards a *hierarchical* model of routing policies

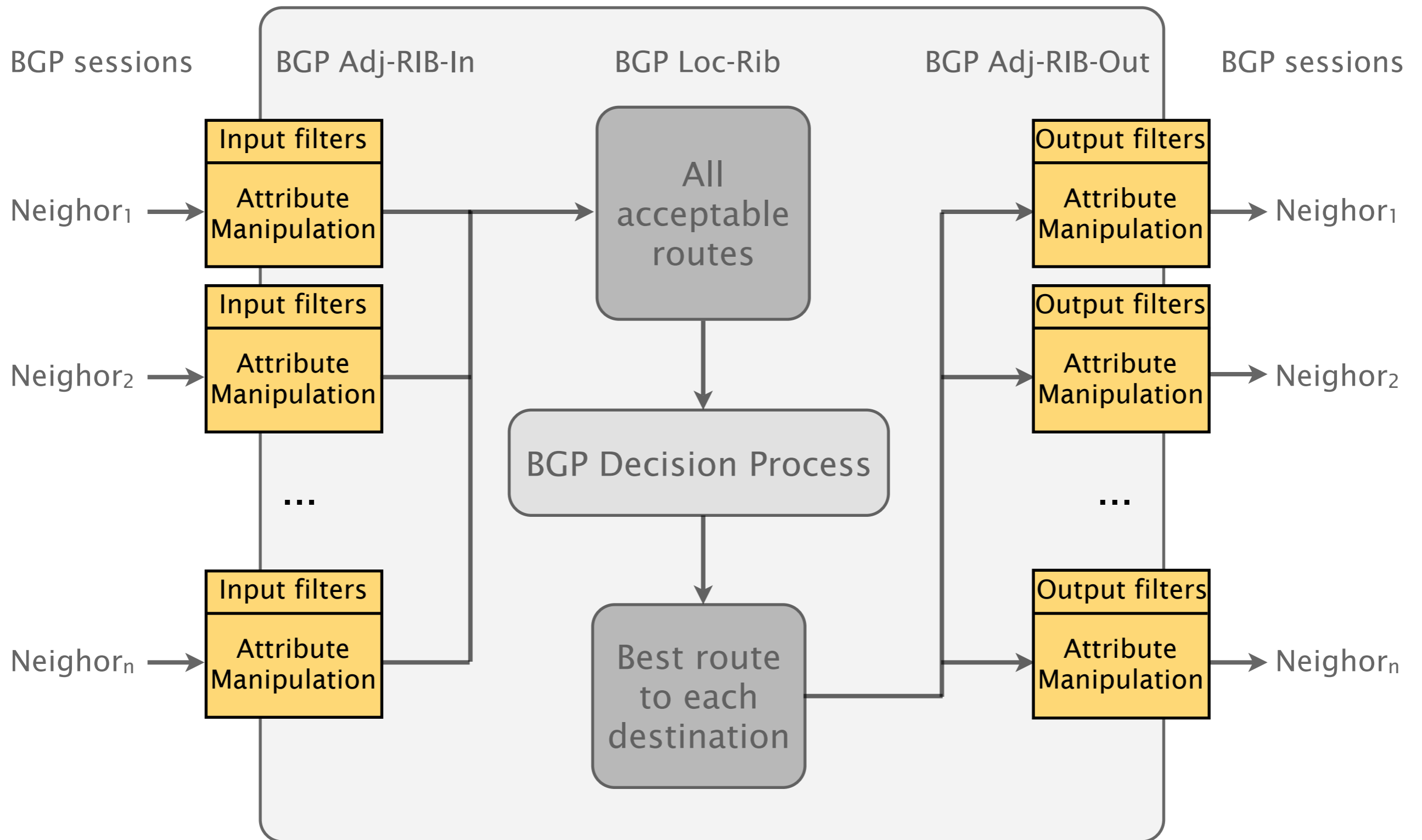
Implementation

Conclusion

A BGP Router at a Glance



Talk is about BGP Policies

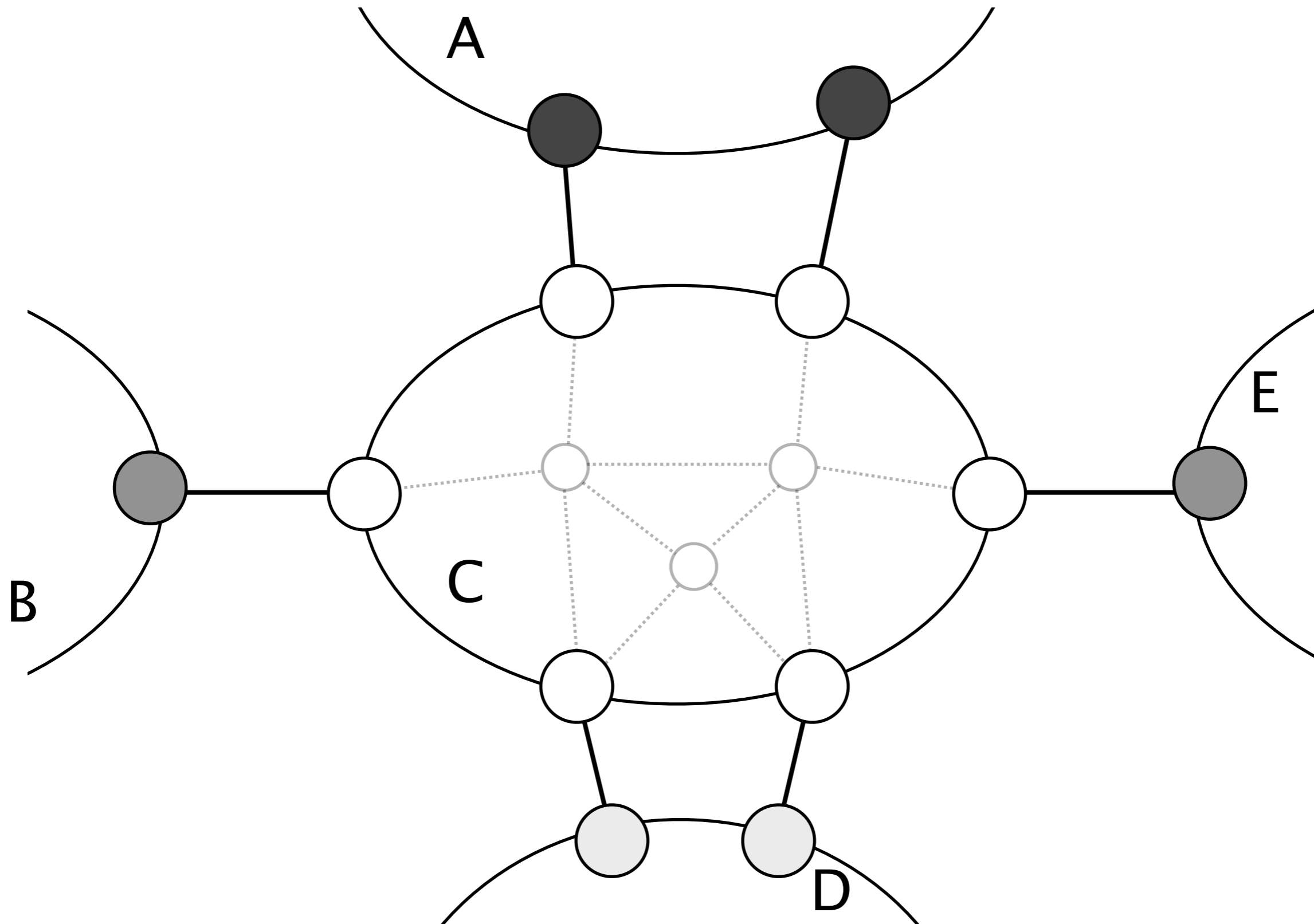


BGP Policies give operators control over routes selection

Policies are mainly used to

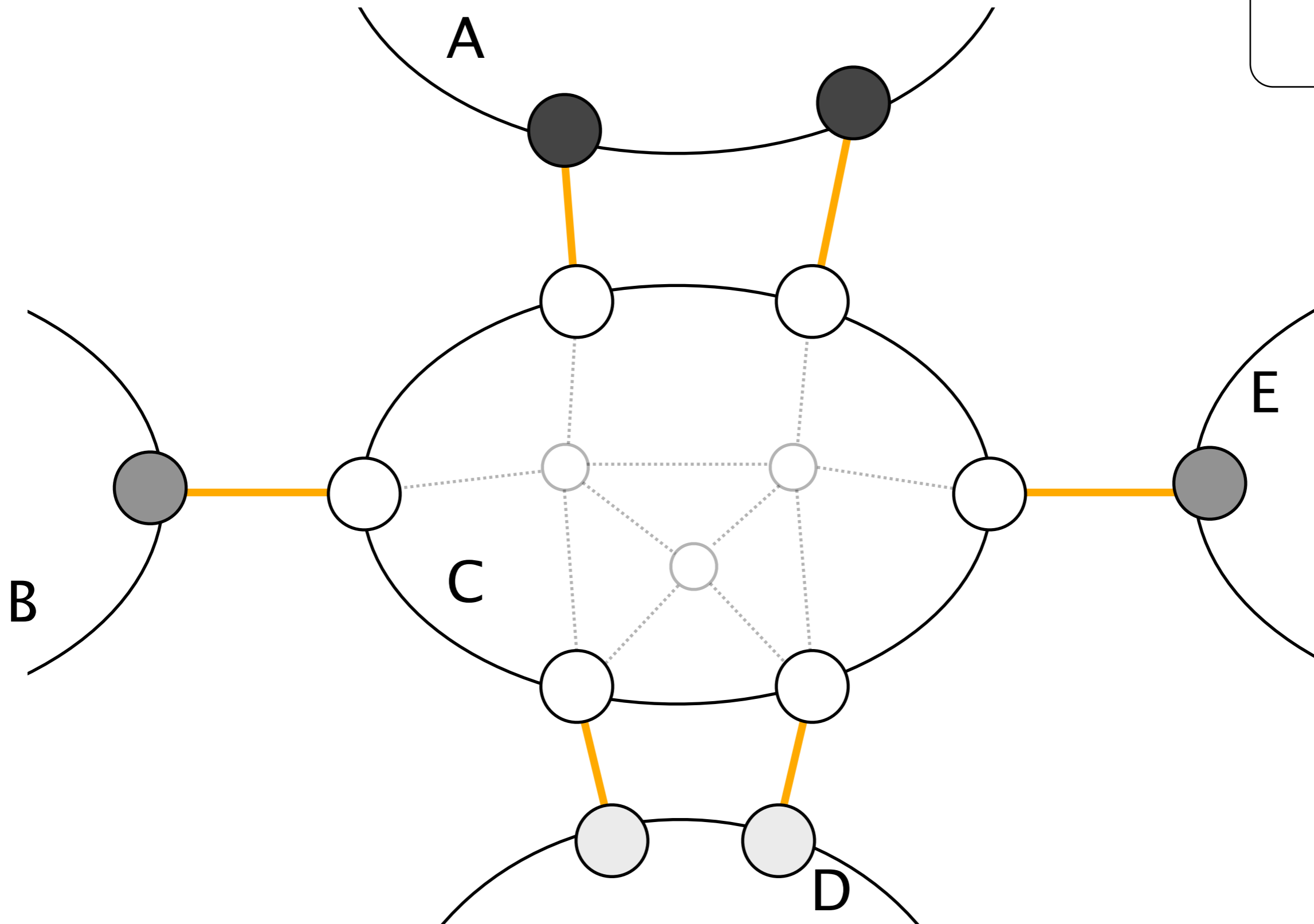
- filter incoming routes
 - ignore* routes you don't want to consider
- modify routes' attributes
 - influence* path selection
 - modify* the way routes are perceived
- filter outgoing routes
 - enforce* business relationships

BGP Policies are defined at different *abstraction* levels



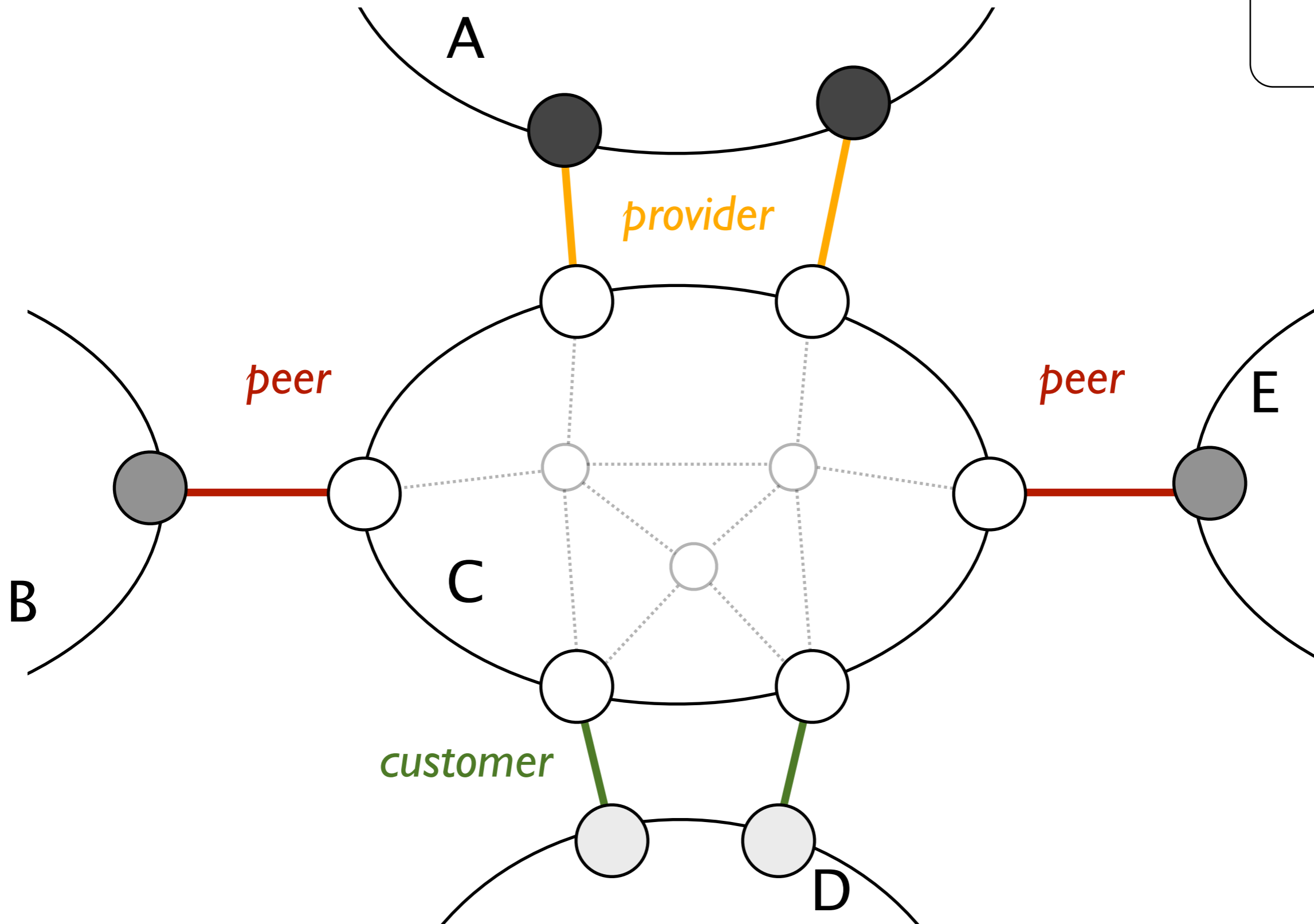
Some BGP Policies are defined on *all* sessions

■ all



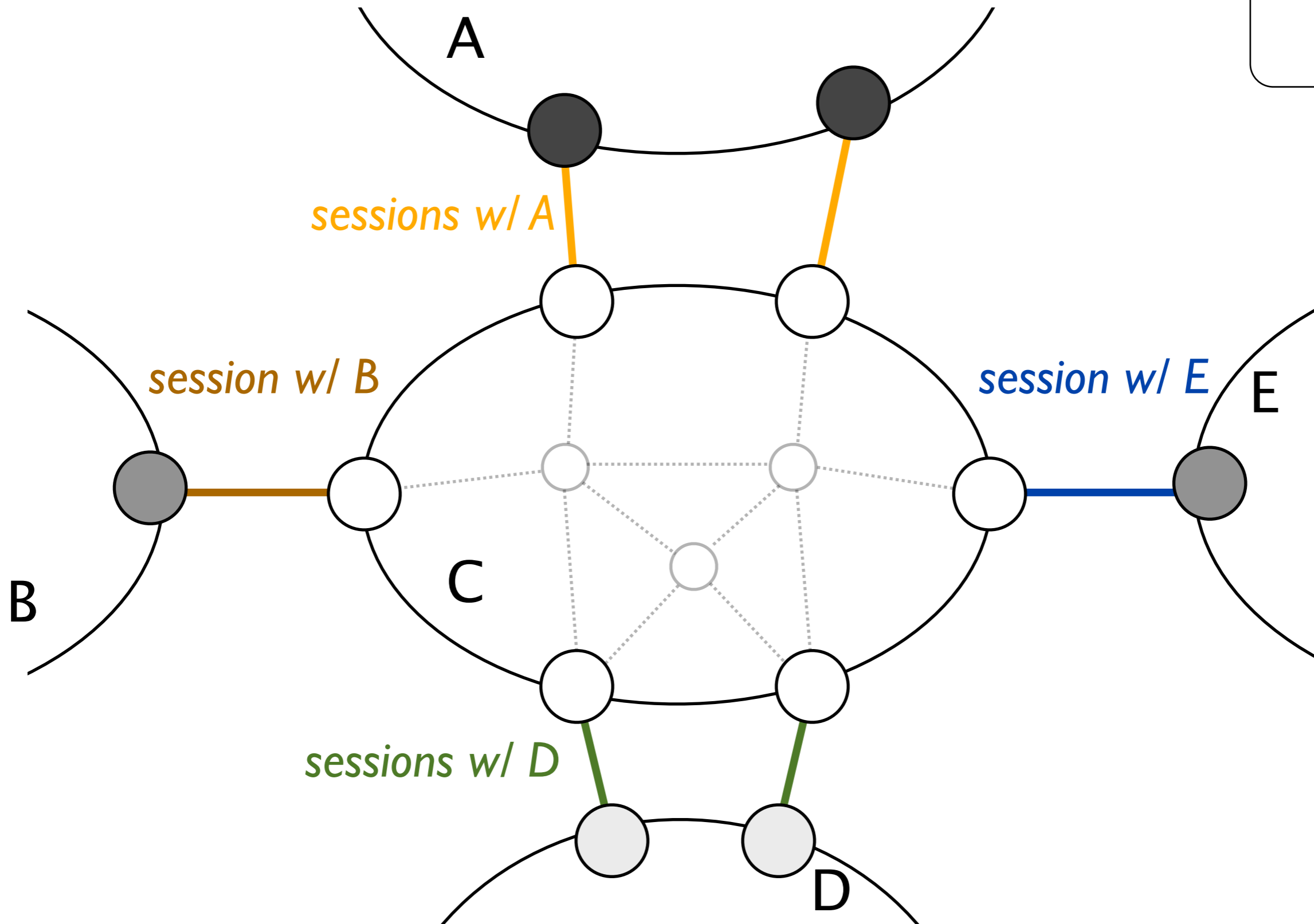
Some BGP Policies are defined on *groups* of sessions

- all
- group



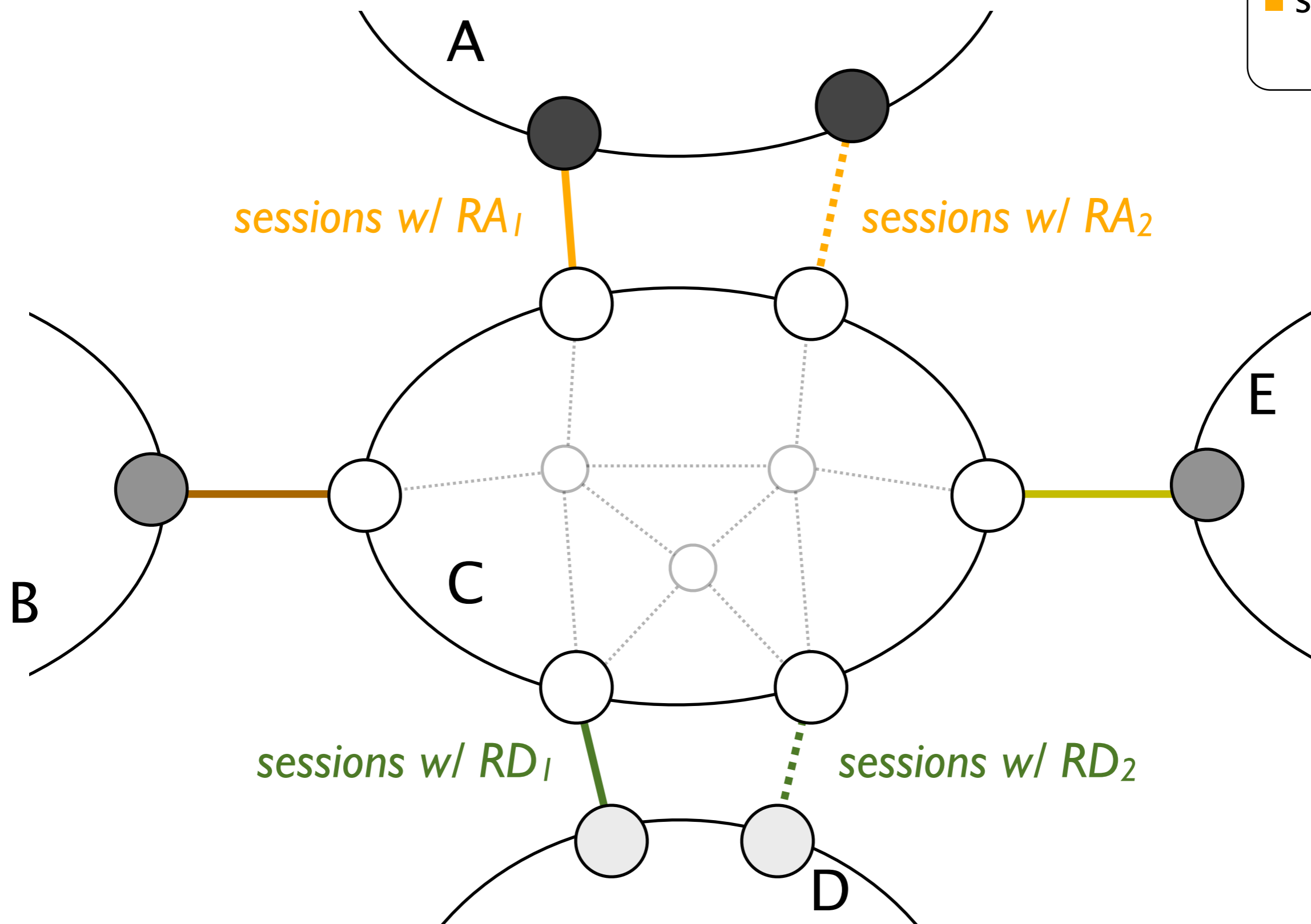
Some BGP Policies are defined on *AS* sessions

- all
- group
- as



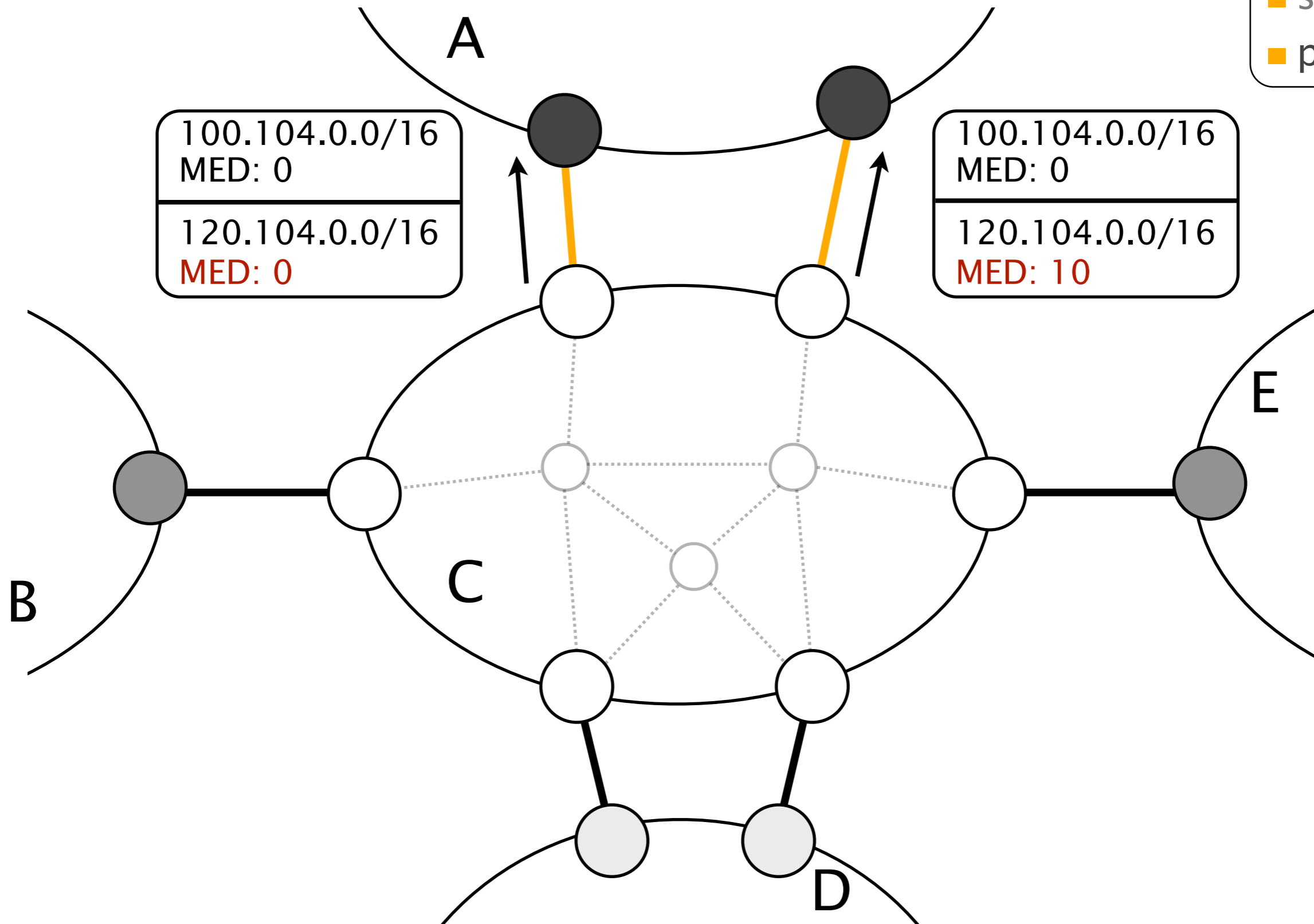
Some BGP Policies are defined on *individual* sessions

- all
- group
- as
- session



Some BGP Policies are defined on *prefixes*

- all
- group
- as
- session
- prefix



However, policies are often defined at *low* level

```
neighbor 206.196.178.45 {
  description "Mid-Atlantic Crossroads (MAX)";
  import [ SANITY-IN SET-CONNECTOR-PREF MAX-IN CONNECTOR-IN ];
  peer-as 10886;
}
neighbor 192.88.192.137 {
  description OSCnet;
  import [ SANITY-IN SET-CONNECTOR-PREF OARNET-IN CONNECTOR-IN ];
  peer-as 3112;
}
neighbor 204.238.76.5 {
  description "Drexel University";
  import [ SANITY-IN SET-CONNECTOR-PREF DREXEL-IN CONNECTOR-IN ];
  peer-as 36412;
}
neighbor 192.88.115.24 {
  description 3ROX;
  import [ SANITY-IN SET-CONNECTOR-PREF PSC-IN CONNECTOR-IN ];
  peer-as 5050;
}
...
neighbor 199.18.156.241 {
  description "OSCnet mcast-only for their non-I2 customers";
  import [ SANITY-IN SET-CONNECTOR-PREF CONNECTOR-IN ];
  peer-as 600;
}
```

Diagram annotations:

- Arrow labeled "all" points to the first neighbor block.
- Arrow labeled "session" points to the "MAX-IN" policy in the first neighbor block.
- Arrow labeled "group" points to the "CONNECTOR-IN" policy in the first neighbor block.

A Hierarchical Model for BGP Routing Policies

Introduction and Motivation

Towards a *hierarchical* model of routing policies

Implementation

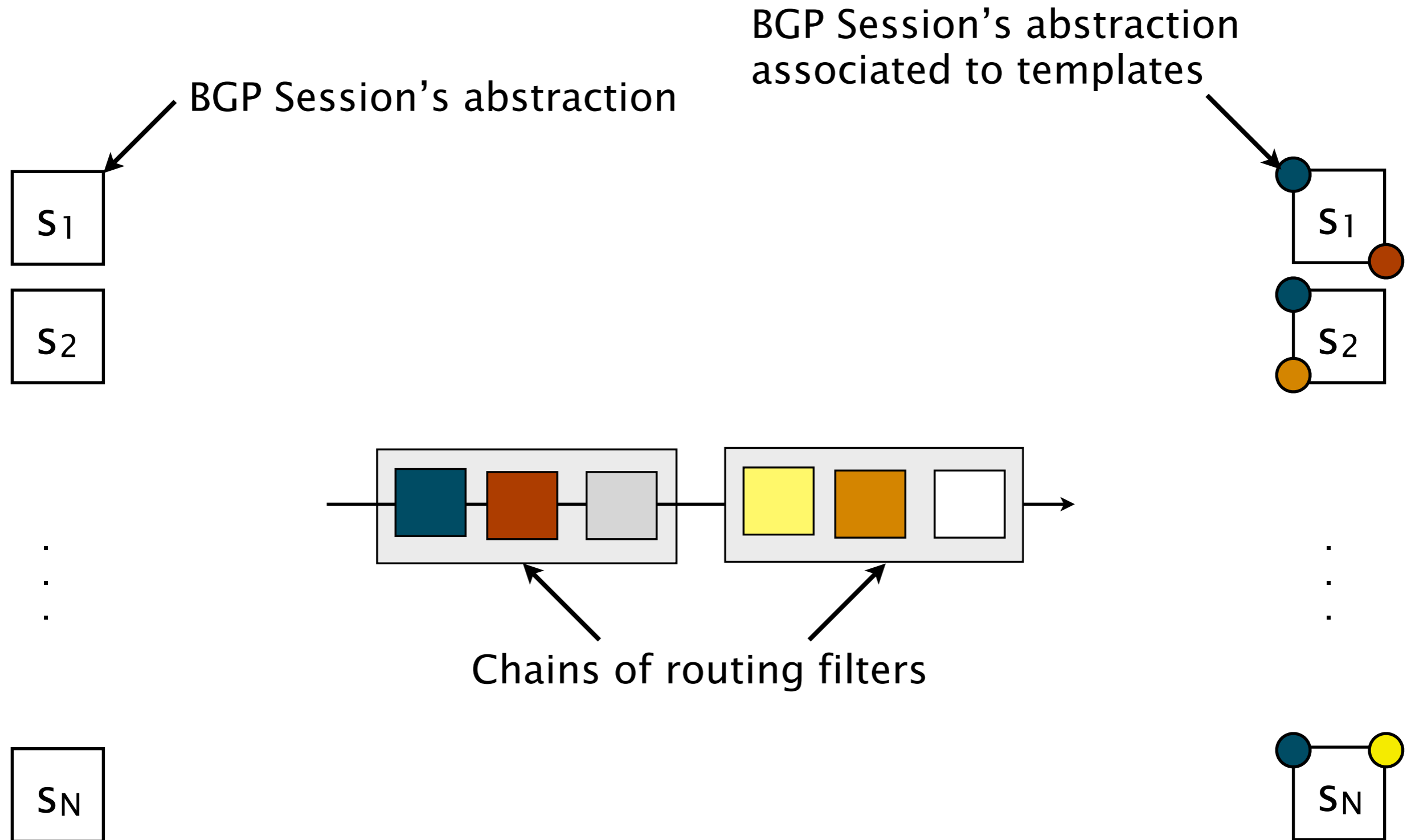
Conclusion

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

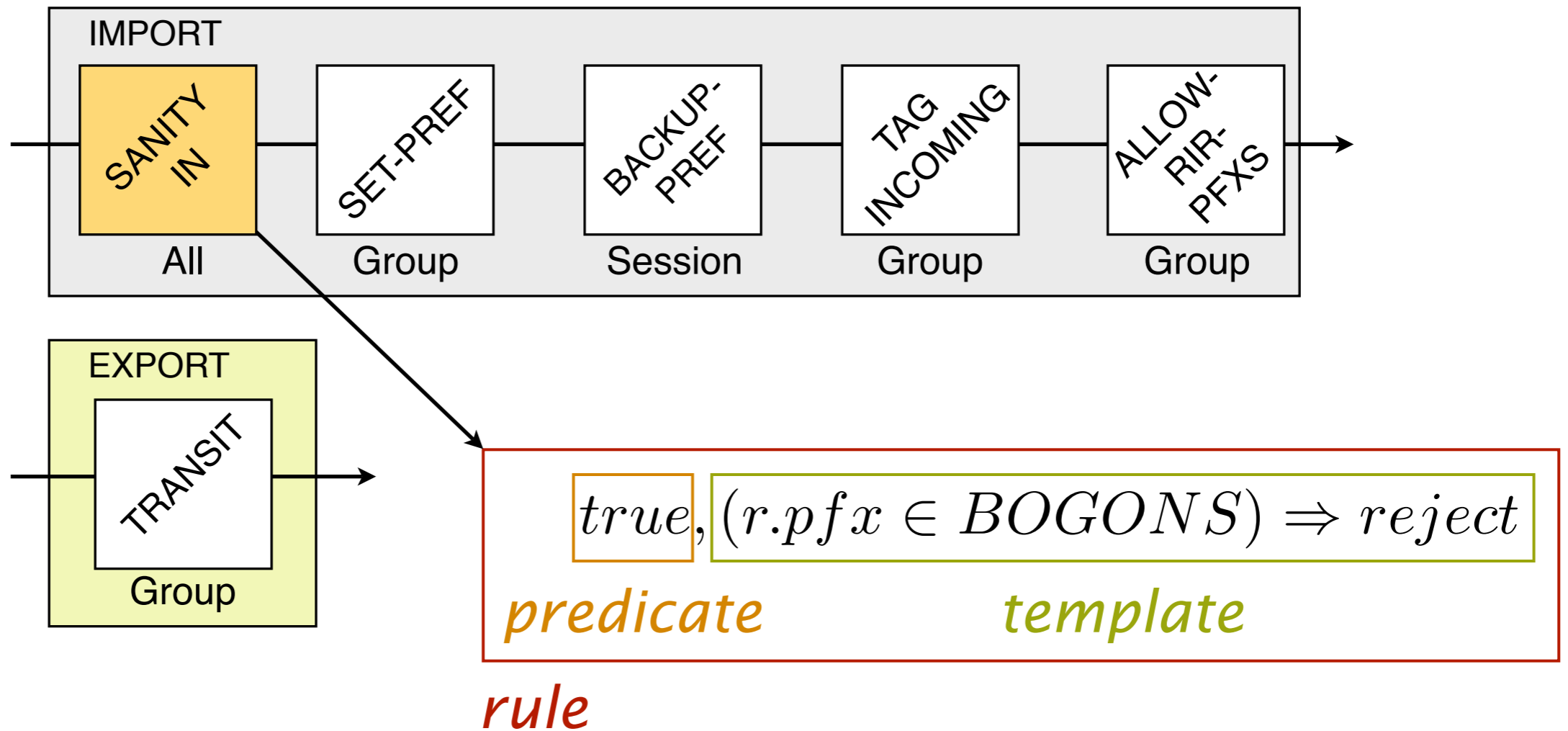


Our model is structured around *chains* of filters

Policies are modeled by chains

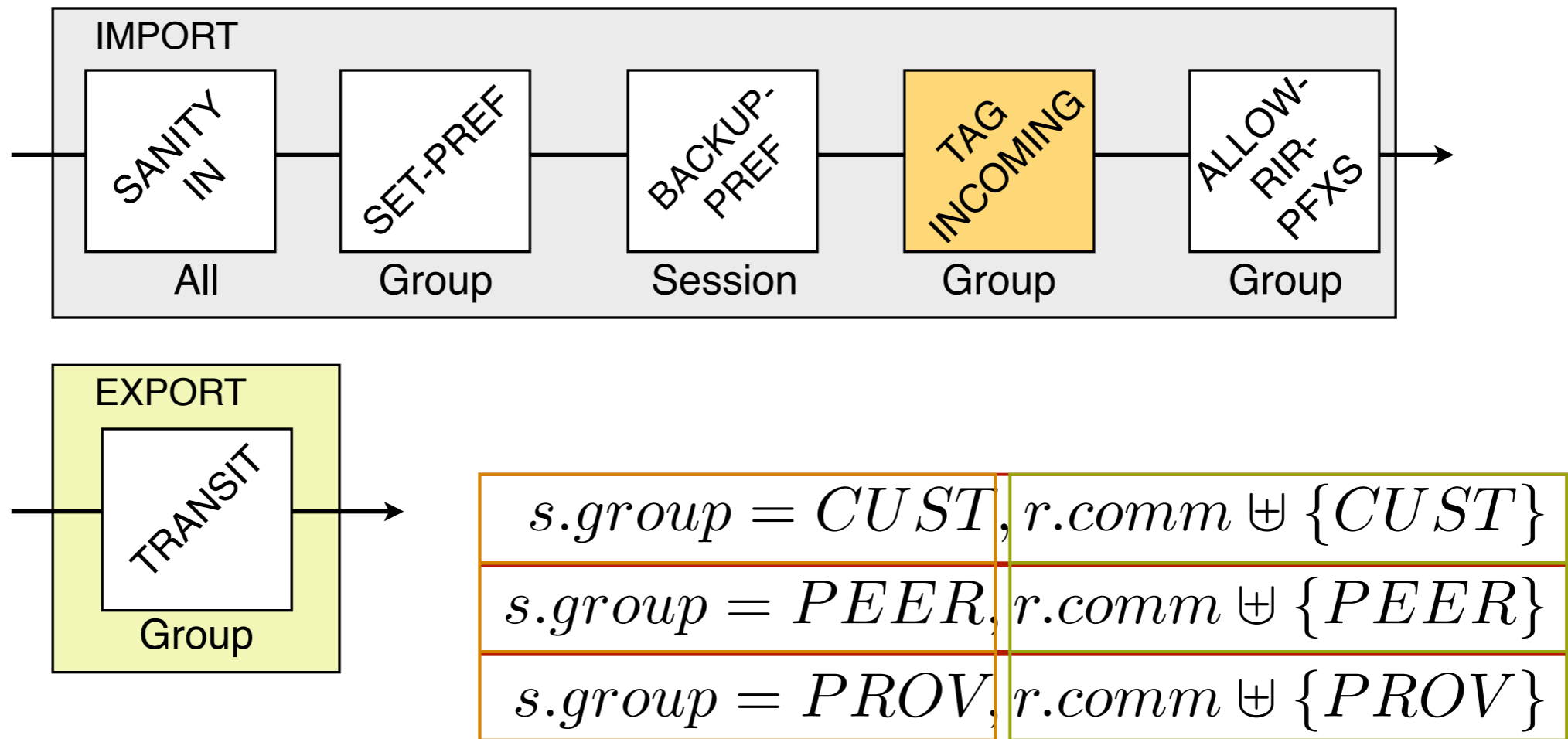
- a node is a sequence of *rules*
- a *rule* is a couple (*predicate*, *template*)
- a *predicate* conditions the association of the template to the session's filters
- a *template* is a sequence of routing filters statements

Our model is structured around *chains* of filters



s : session
 r : route

Our model is structured around *chains* of filters



s : 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
```

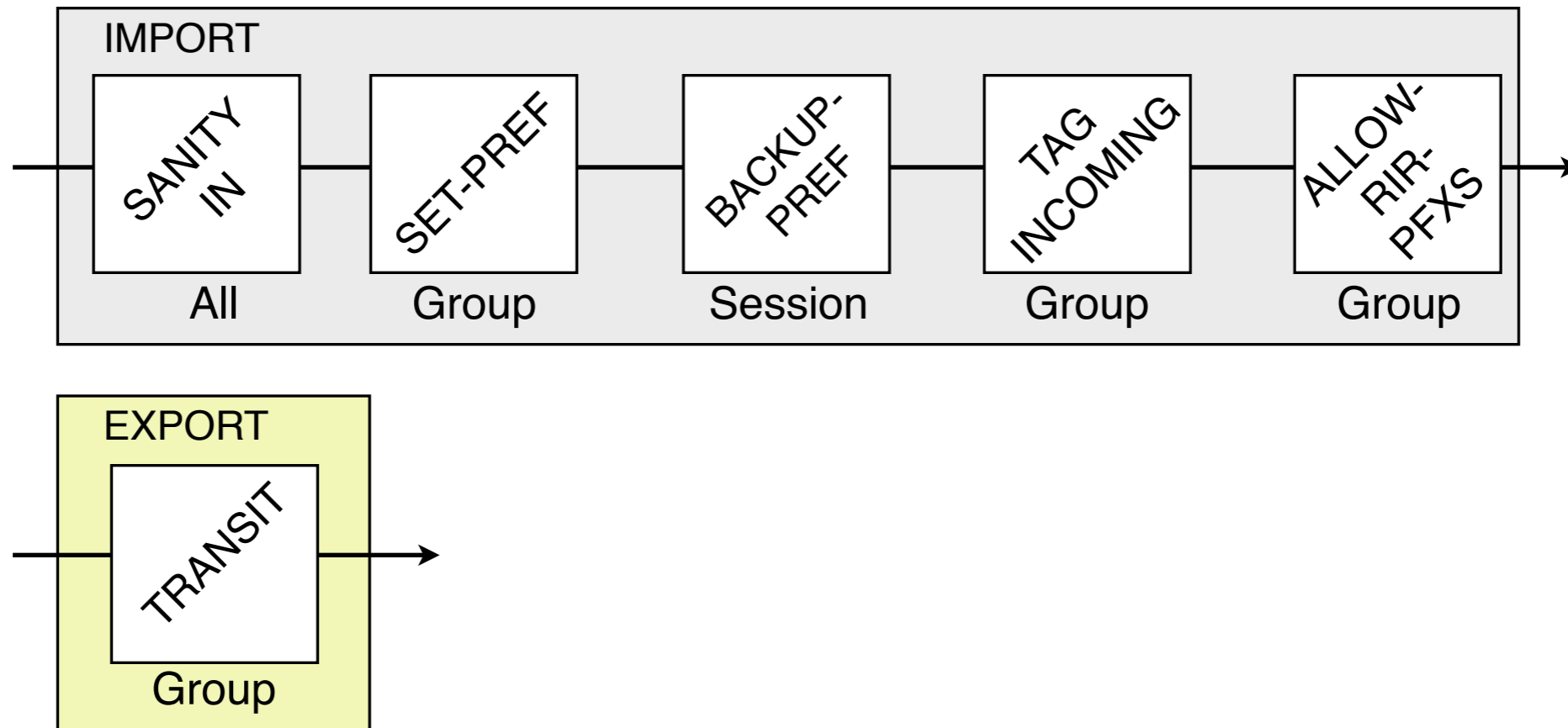
- Predicates are modeled by *regular expressions*

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

- Templates are represented by using *StringTemplate*

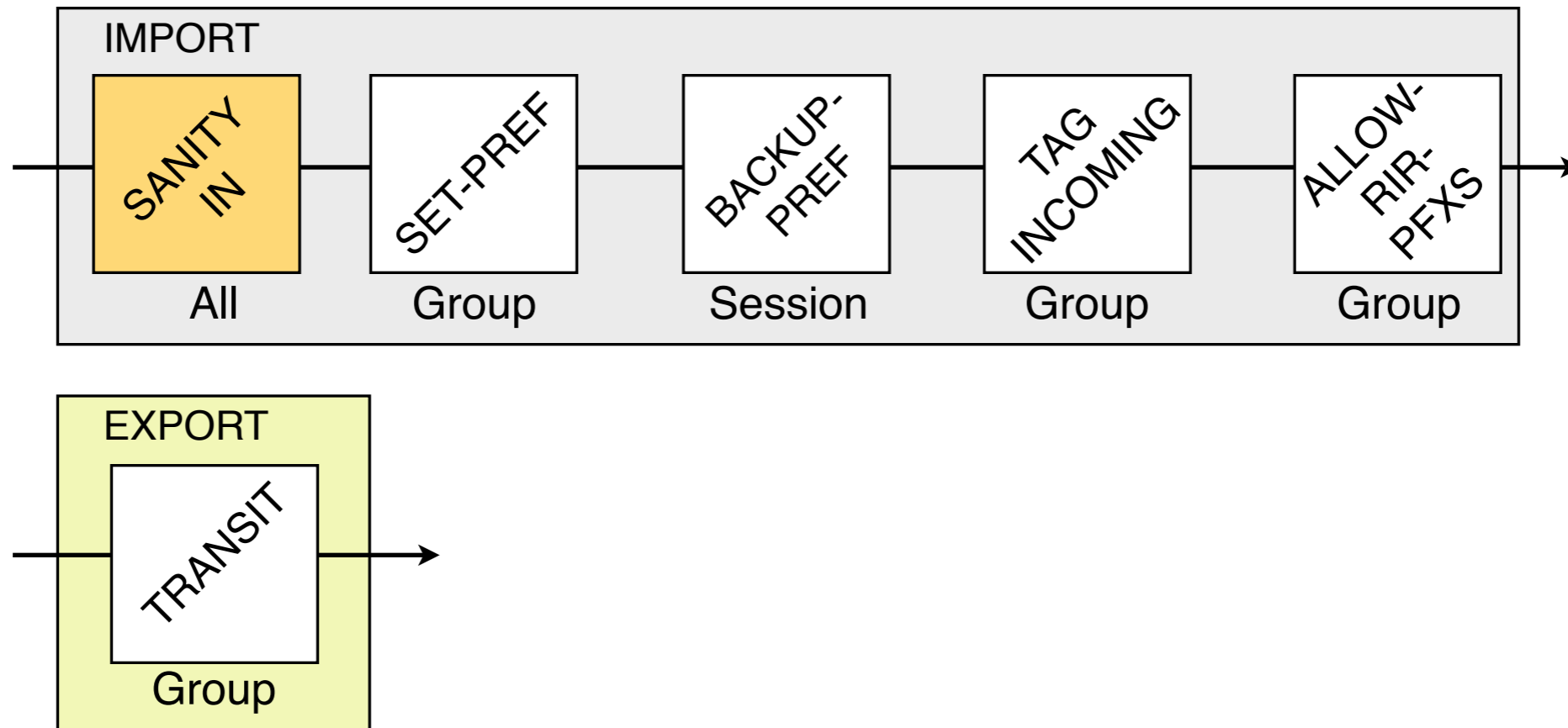
```
policy-statement BACKUP-PREF {  
  term down-pref {  
    then {  
      local-preference subtract $value$;  
      accept;  
    }  
  }  
  ...  
}
```

How does it work ?



BXL : CUST : 2611 : <130.104.0.2> : backup

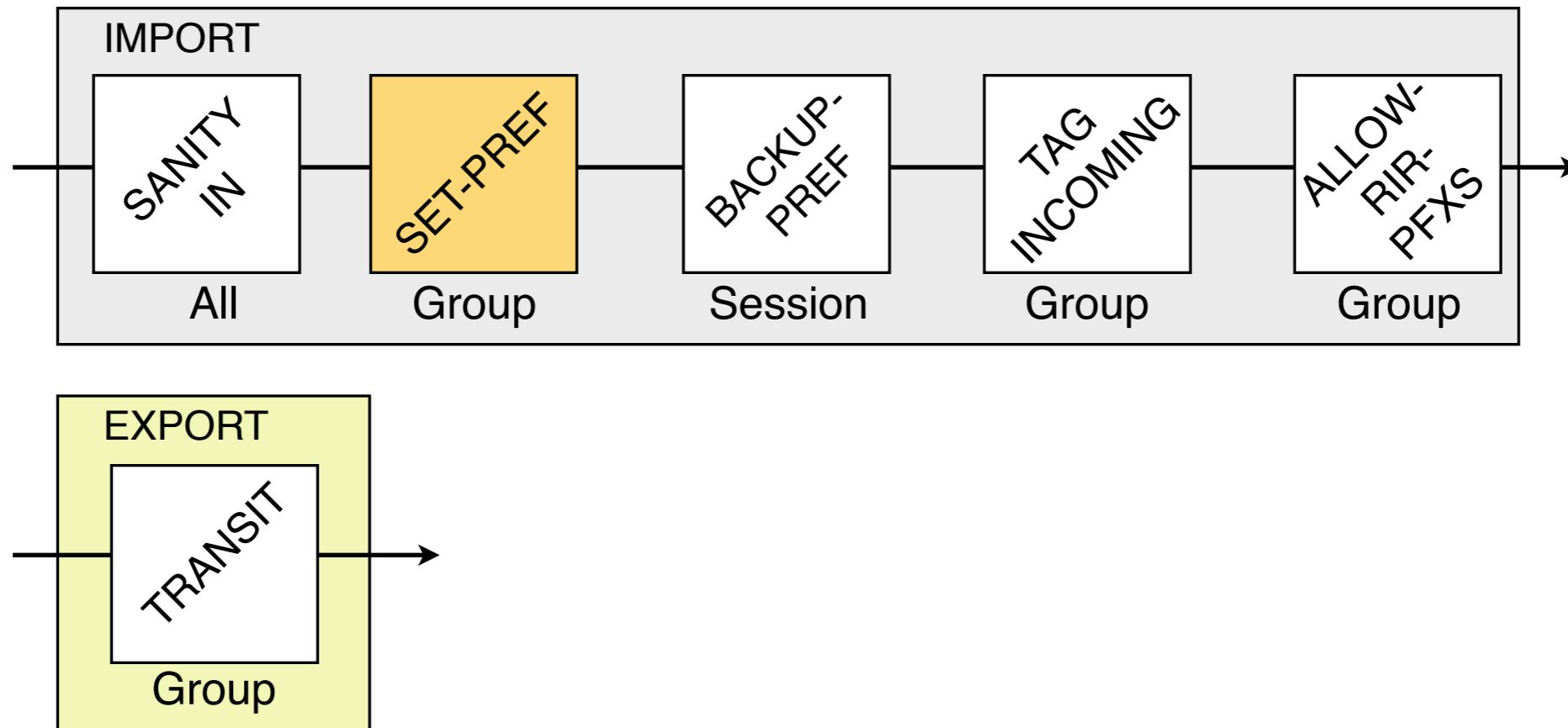
How does it work ?



BXL:CUST:2611:<130.104.0.2>:backup

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

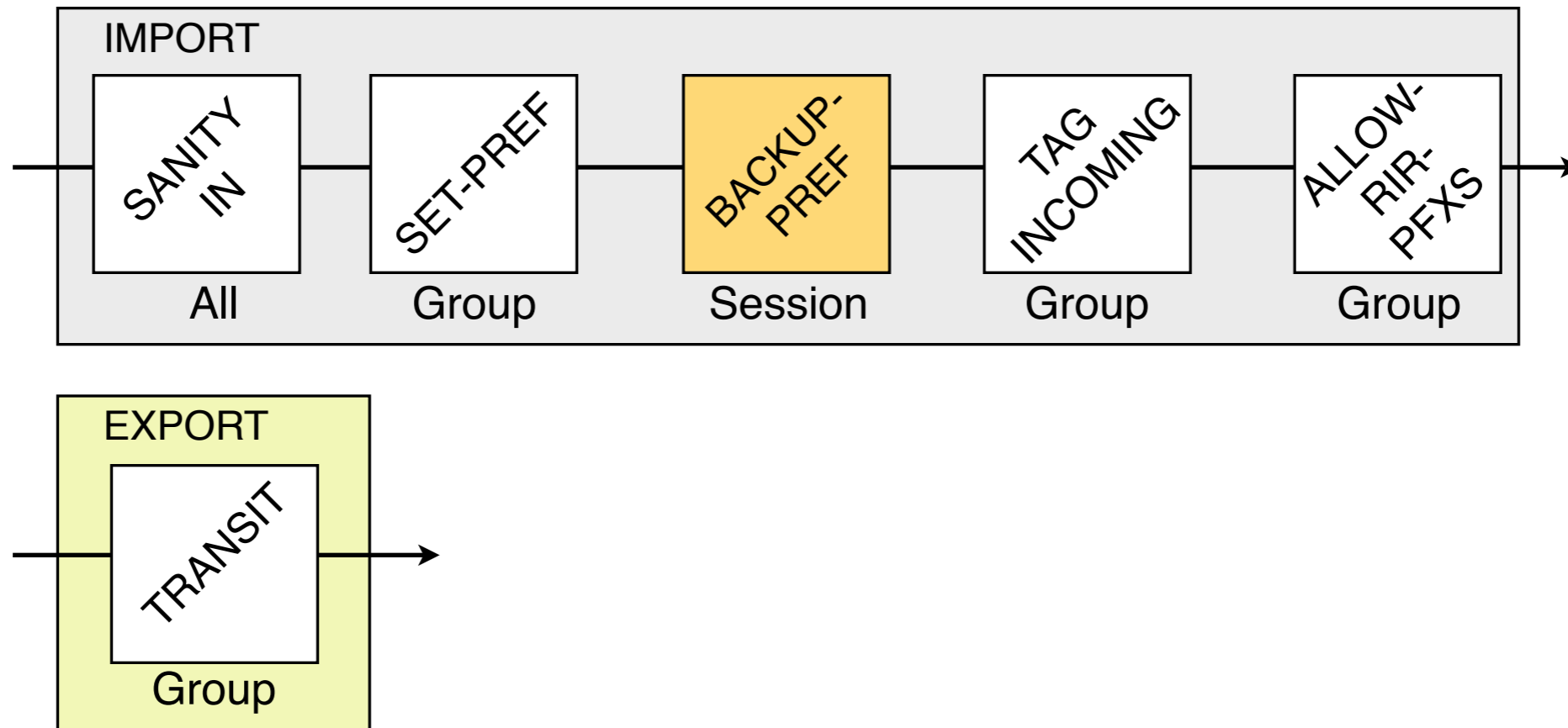
How does it work ?



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

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

How does it work ?



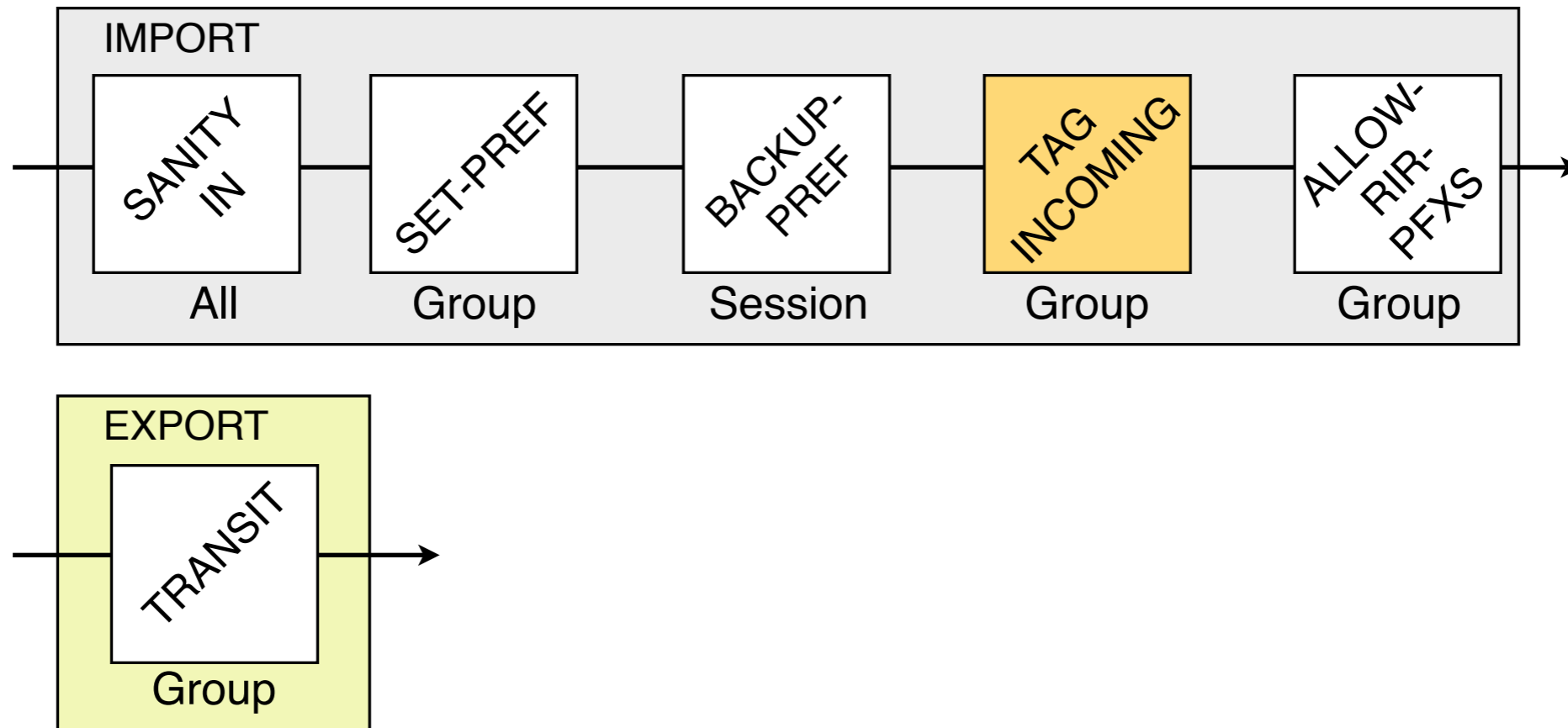
BXL:CUST:2611:<130.104.0.2>: **backup**

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

$r.lp = 5000$

$r.lp = r.lp - 500$

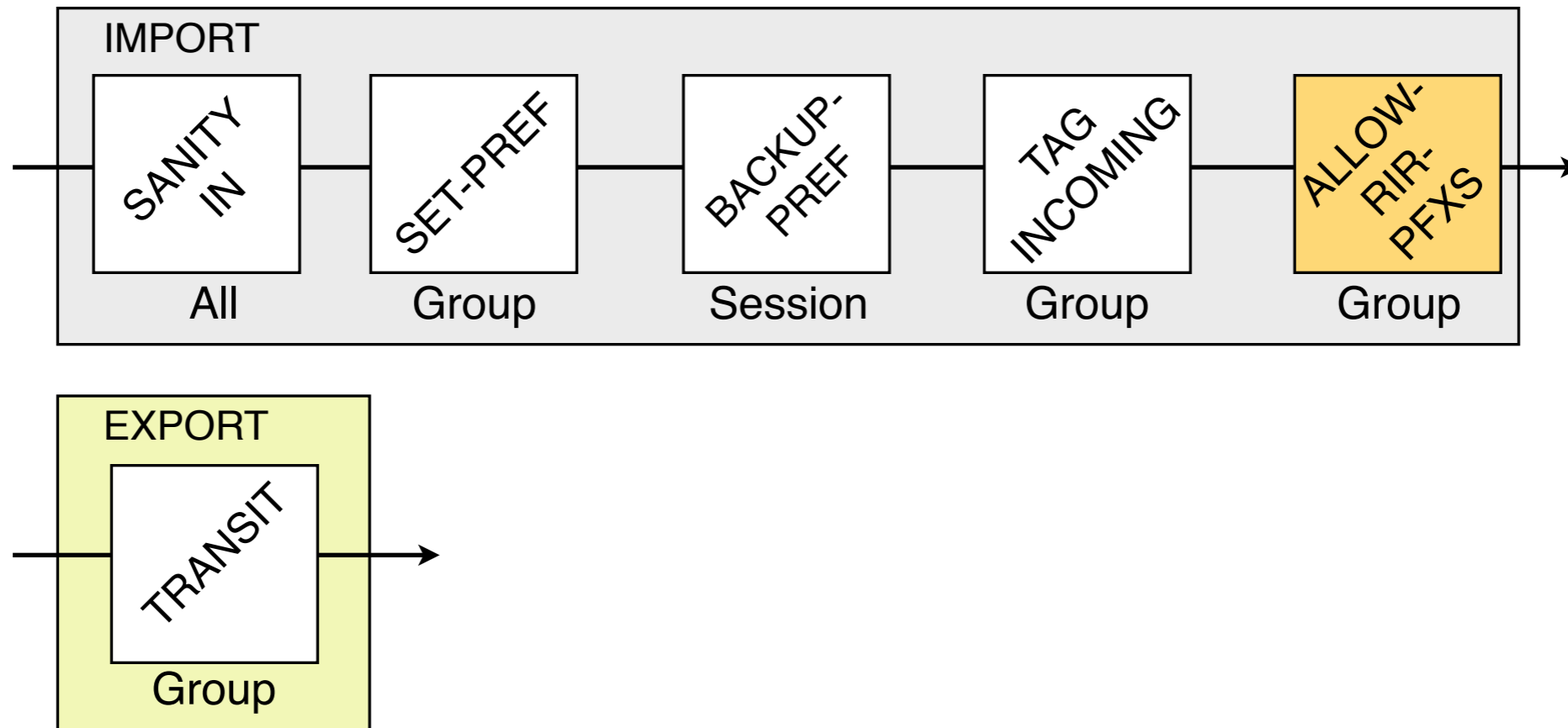
How does it work ?



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

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

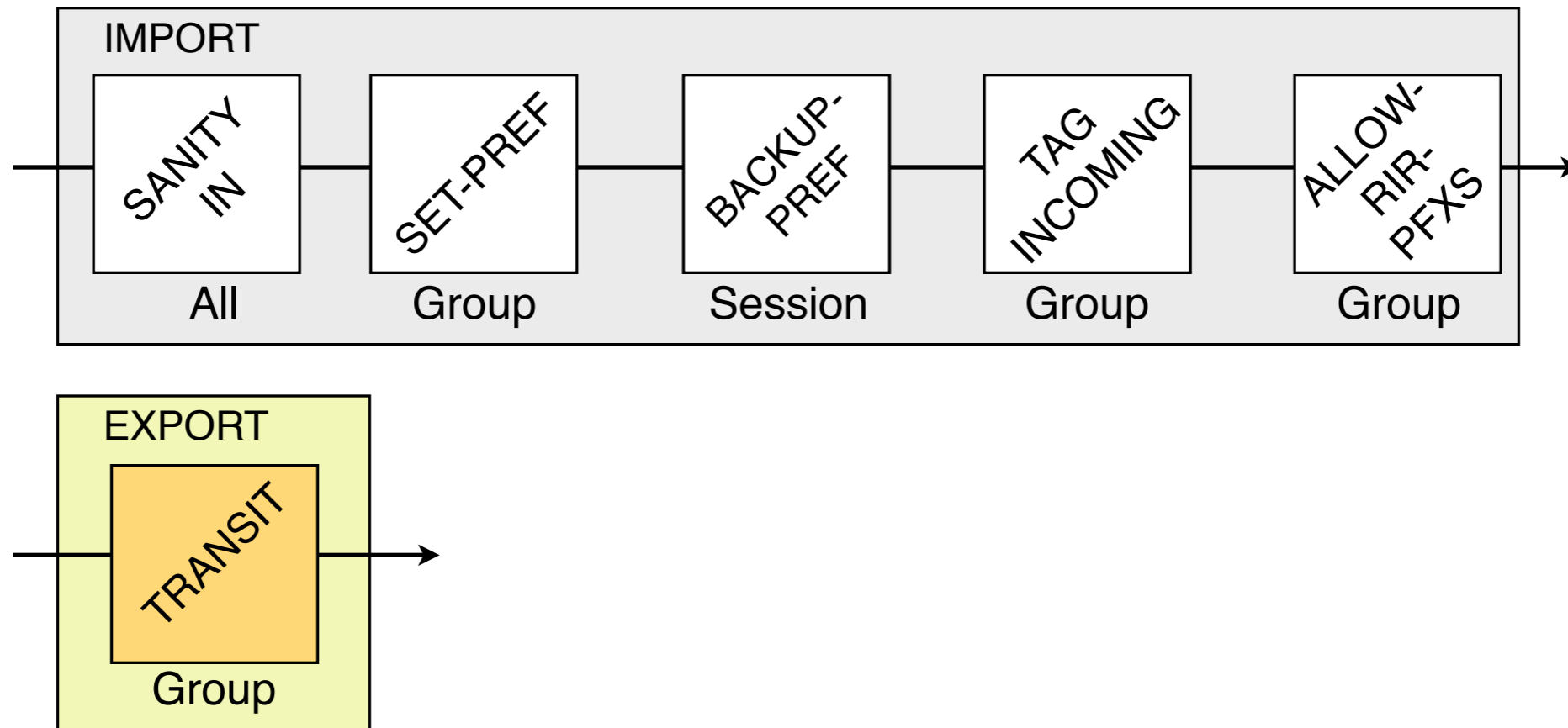
How does it work ?



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 ?

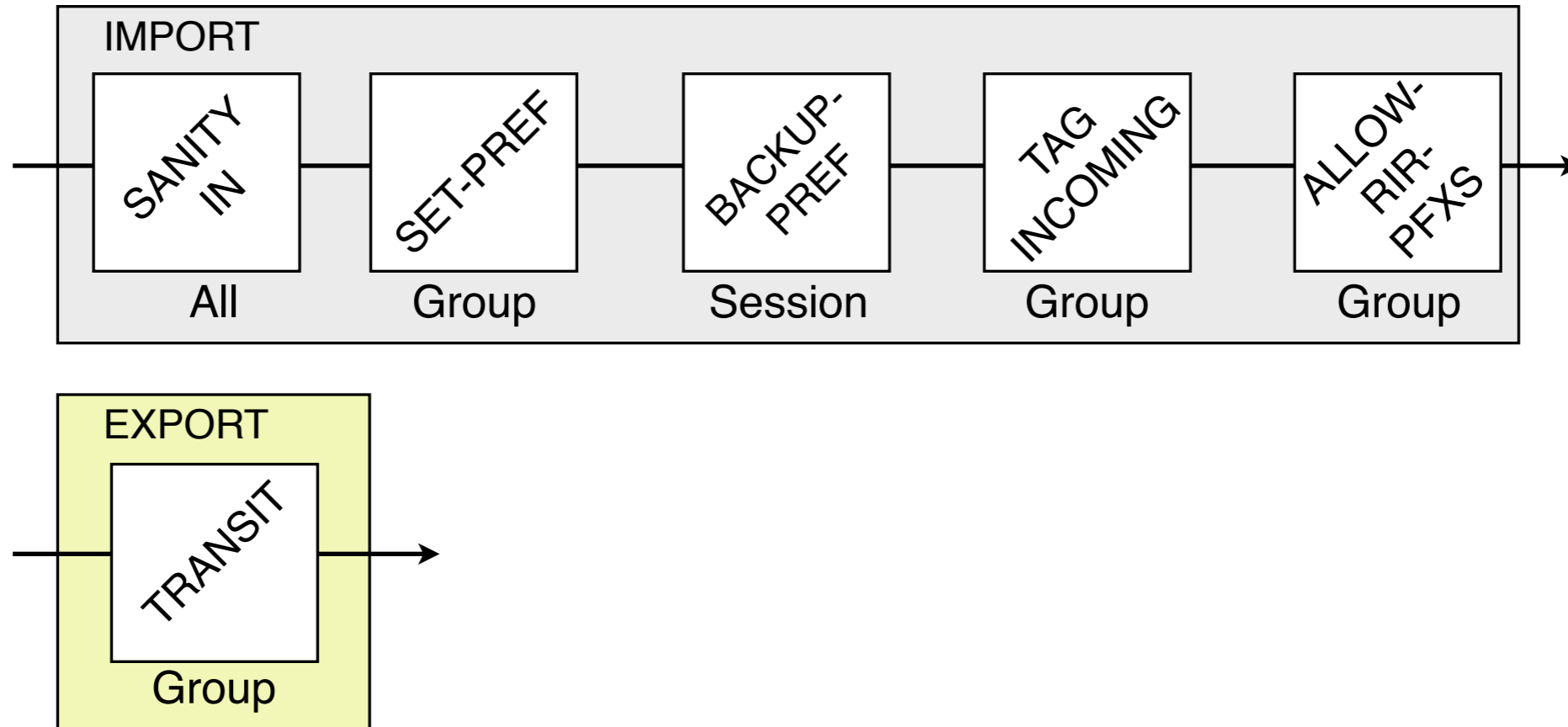


BXL : **CUST** : 2611 : <130.104.0.2> : backup

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

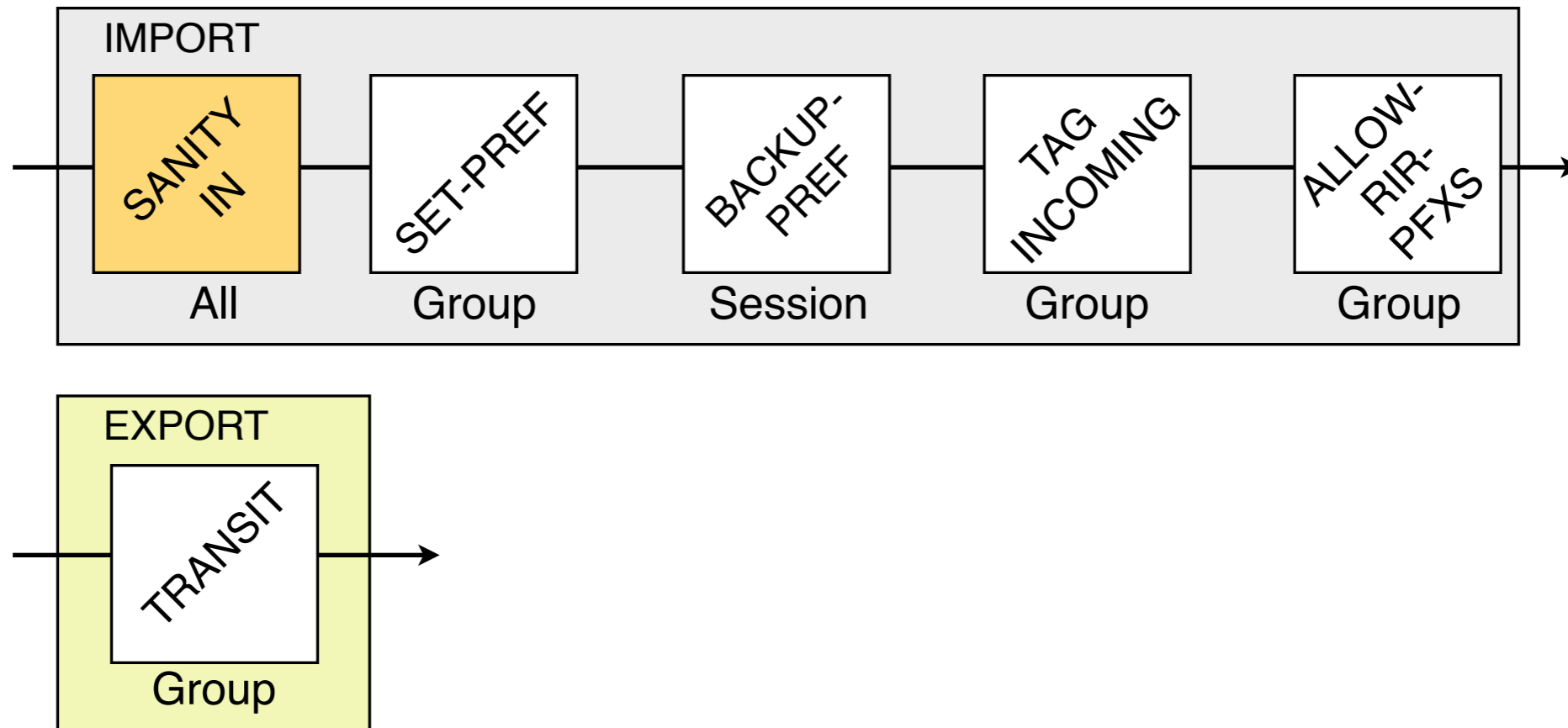
announce default route

How does it work ?



BXL : PROVIDER : 2611 : <130.104.0.2>

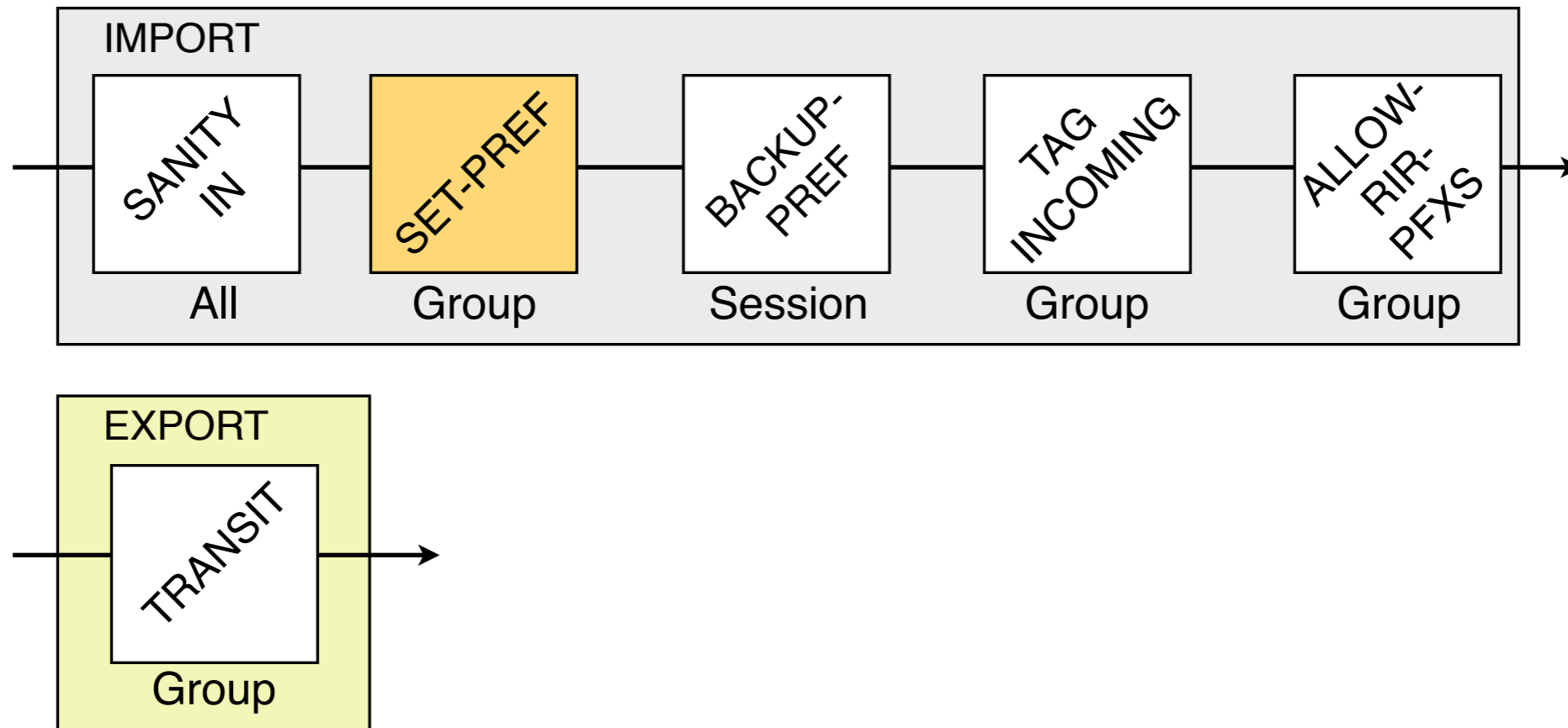
How does it work ?



BXL : PROVIDER : 2611 : <130.104.0.2>

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

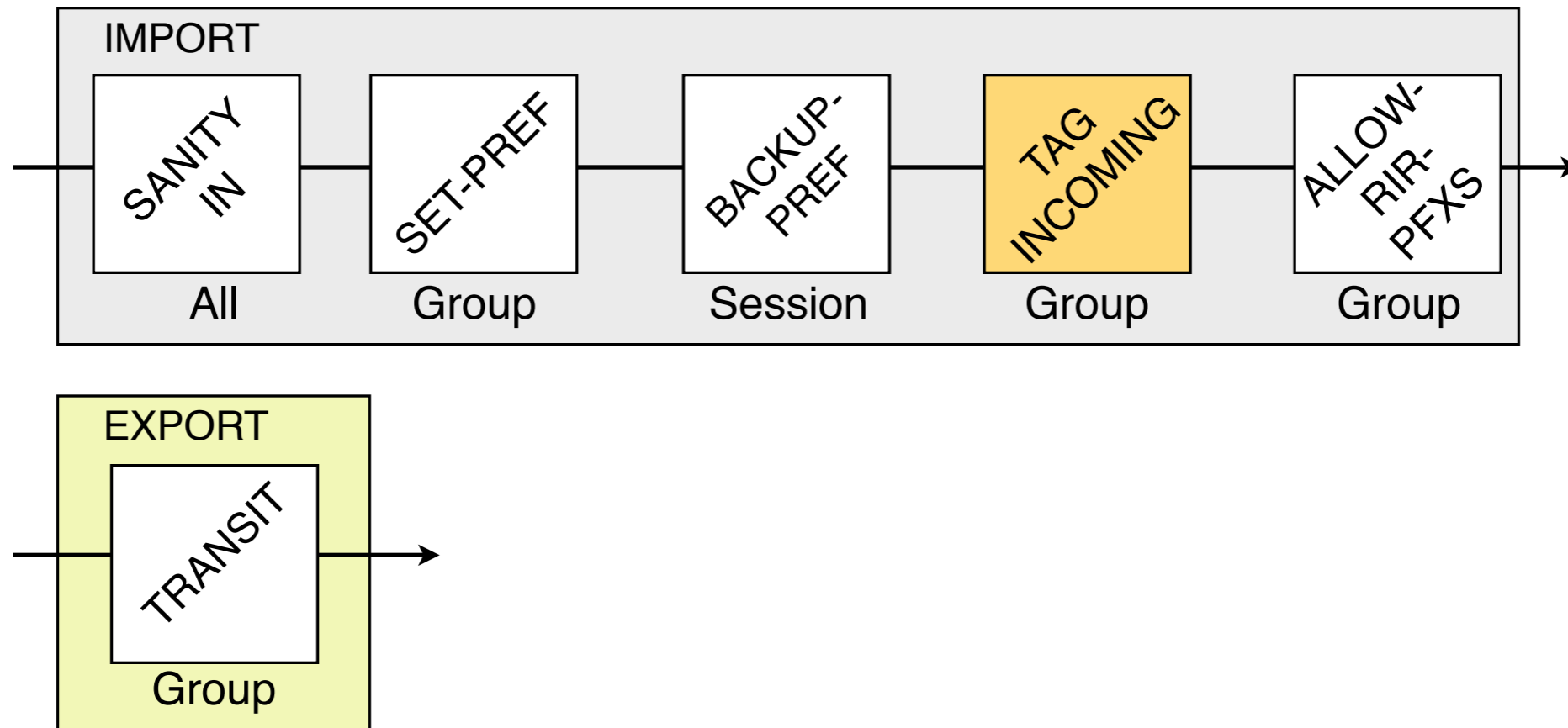
How does it work ?



BXL : PROVIDER : 2611 : <130.104.0.2>

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

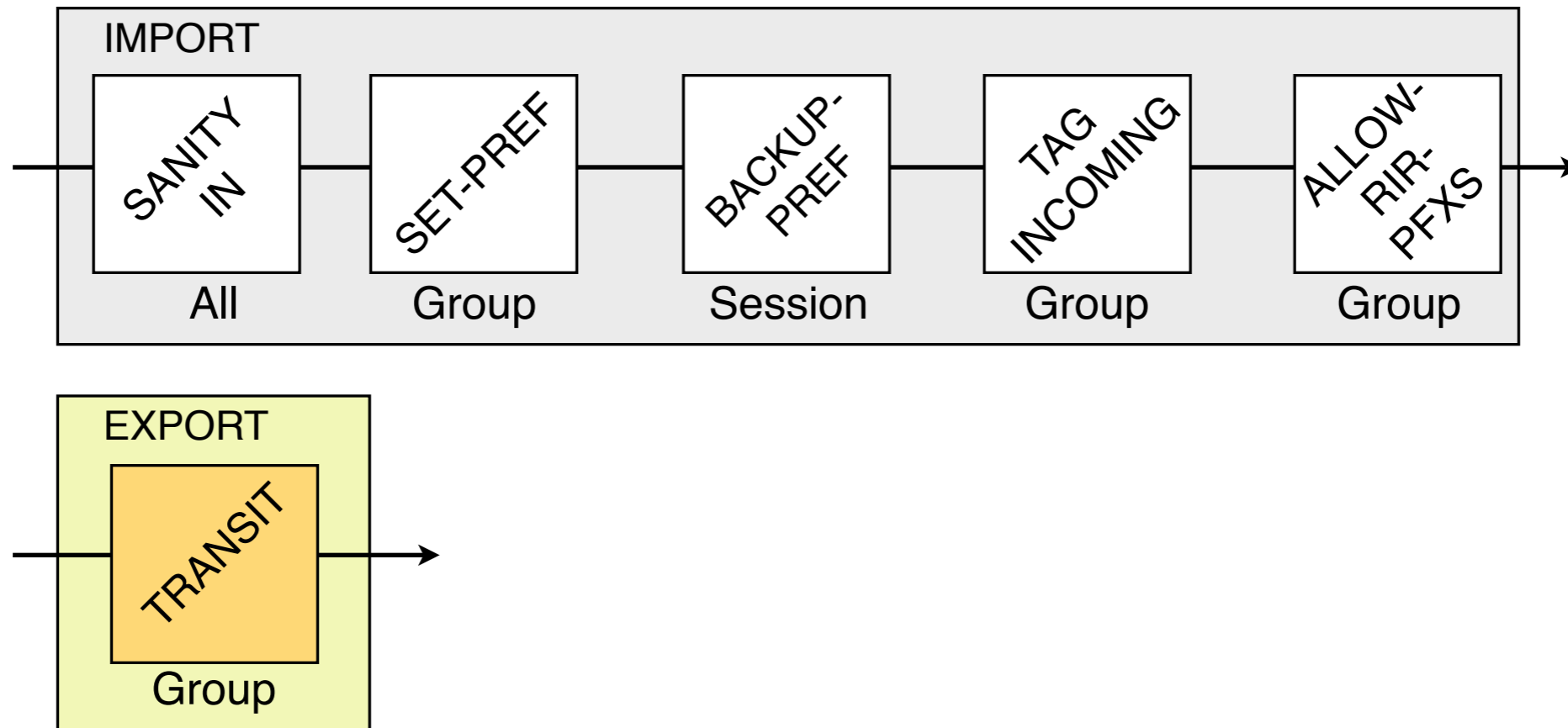
How does it work ?



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>