

## Network Training Workshop

---

### Routing Policy Specification Language

RIPE NCC Database Group  
<ripe-dbm@ripe.net>

RPSL/1

## RPSL Tutorial

---

### ⇒Goal

⇒Awareness of RPSL

⇒Examples

### ⇒Target Audience

⇒knowledge of Internet Routing

⇒no need to know Internet Routing Registry

### ⇒Tutorial

⇒not a substitute for reading the drafts

RPSL/2

## RPSL Drafts

---

⇒draft-ietf-rps-rpsl-v2-03

⇒Routing Policy Specification Language (RPSL)

⇒draft-ietf-rps-appl-rpsl-04

⇒Using RPSL in Practice

RPSL/3

## Contents of the tutorial

---

### ⇒Introduction

⇒Internet Routing Registry

⇒IETF RPS Working Group

⇒RPSL; background

### ⇒RPSL

⇒Objects, Reserved Words/Names

⇒Autonomous Systems and Routes

⇒Sets and Operators

⇒Specifying Policies

RPSL/4

## Introduction

---

- ⇒ Internet Routing Registry
  - ⇒ History
  - ⇒ Structure
- ⇒ IETF RPS Working Group
  - ⇒ Activities
  - ⇒ Charter
- ⇒ RPSL background
  - ⇒ Comparison with RIPE-181
  - ⇒ Goals and Milestones
  - ⇒ Current Status

RPSL/5

## Internet Routing Registry

---

- ⇒ Established in 1995
- ⇒ Stability and consistency of routing
  - ⇒ network operators share information
- ⇒ Public databases:
  - ⇒ Cable & Wireless
  - ⇒ ANS
  - ⇒ Merit (RADB)
  - ⇒ Bell Canada (formerly CA\*net)
  - ⇒ RIPE

RPSL/6

## Internet Routing Registry 2

---

- ⇒ These databases are independent
  - ⇒ but they exchange data
  - ⇒ can see all these databases at any one site
  - ⇒ only register your data in one database
- ⇒ Private: Cable & Wireless, ANS, Bell Canada
- ⇒ Public: RADB, RIPE
- ⇒ Routing policies expressed in RIPE-181
  - ⇒ RPSL is based on RIPE-181

RPSL/7

## IETF RPS Working Group

---

- ⇒ Routing Policy System Working Group
- ⇒ Standardization:
  - ⇒ protocols and recommended practices
- ⇒ Activities:
  - ⇒ define a language
  - ⇒ define a distributed registry model
  - ⇒ provide a forum for discussion
- ⇒ <http://www.ietf.org/html.charters/rps-charter.htm>

RPSL/8

## RPSL

---

- ⇒RIPE-181
  - ⇒some policies cannot be specified
- ⇒Internet Routing Registry
  - ⇒needed a more powerful language
- ⇒RPSL
  - ⇒more expressive than RIPE-181
  - ⇒policies can be specified at the AS level
  - ⇒policies can be detailed => router configurations
  - ⇒extensible (new routing protocols possible)
  - ⇒structured, vendor-neutral

RPSL/9

## RPSL Features

---

- ⇒Language features:
  - ⇒AS and route objects, object sets, policy as relations
  - ⇒structured, vendor-neutral, extensible
  - ⇒RPS IETF WG

### ⇒History

PRDB ⇒ RIPE 81 ⇒ RIPE 181 ⇒ RPSL

RPSL/10

## Goals and Milestones of RPSL

---

- ⇒RPSL Coding: 9/1997 – 5/1998
- ⇒Testing and training: 5/1998 – now
- ⇒Deployment: 1999
- ⇒April 1999
  - ⇒submit RPSL to IESG as a Proposed Standard

RPSL/11

## Current Status

---

- ⇒RPSL and RIPE-181 servers in parallel
  - ⇒rpsl.merit.edu
  - ⇒rpslii.ripe.net
    - ⇒<auto-rpsl@ripe.net>
    - ⇒whois -h rpslii.ripe.net
- ⇒RIPE Database
  - ⇒RPSL in new implementation
- ⇒RPSL training
  - ⇒Wide scale testing
  - ⇒Updating in-house tools/extensions

RPSL/12

## RPSL Training

---

- ⇒ <http://www.isi.edu/ra/rps/training/>
- ⇒ <http://www.ripe.net/db/rps/training/>
- ⇒ Tutorials in the RIPE region
  - ⇒ Edinburgh (September 1998, RIPE-31)
  - ⇒ Amsterdam (January 1999, RIPE-32)
  - ⇒ Vienna (May 1999, RIPE-33)
  - ⇒ Local Internet Registry Training, Quarter 3, 1999

RPSL/13

## Tutorial

---

- ⇒ Objects, Reserved Names
- ⇒ Contact Information
- ⇒ Specifying Policy
- ⇒ Set Objects
- ⇒ Autonomous Systems

RPSL/14

## Tutorial 2

---

- ⇒ Internet Routers
- ⇒ Advanced Topics

RPSL/15

## Part I

---

### RPSL Objects

RPSL/16

## Objects in RPSL

- ⇒ RPSL is based on objects
- ⇒ Objects and Attributes
- ⇒ Attributes and Values
- ⇒ Object Names
- ⇒ Reserved Names

RPSL/17

## Object Based

- ⇒ Each object describes an entity in real world
- ⇒ Object classes
  - ⇒ route
  - ⇒ autonomous system
  - ⇒ router
  - ⇒ person, role, maintainer
  - ⇒ set objects
  - ⇒ dictionary

RPSL/18

## Object

The diagram shows an RPSL object with several attributes. Annotations with boxes and lines point to specific parts of the object:

- Attribute name:** points to `person:`
- Attribute value:** points to `Clare Lancers`
- Comment:** points to `# day time`
- continuation:** points to the `remarks:` attribute

```
• person:  Clare Lancers
• address: Corofin
• phone:   +1 23 123    # day time
• fax-no:  +1 23 121
• e-mail:  clancers@ripe.net
• nic-hdl: CL123-TEST
• remarks: This object is automatically
•          converted from RIPE181
• mnt-by:  RIPE-NCC-MNT
• changed: clancers@ripe.net 19990122
• source:  TEST
```

RPSL/19

## Objects in RPSL

- ⇒ Similar to RIPE-181
- ⇒ Object
  - ⇒ set of attributes
- ⇒ Attributes
  - ⇒ mandatory or optional
  - ⇒ values: single, list, multiple

RPSL/20

## Objects in RPSL 2

---

- ⇒Class "key"
  - ⇒set of attributes
  - ⇒uniquely identify each object
- ⇒Attributes
  - ⇒Case insensitive
  - ⇒ASCII
  - ⇒type

RPSL/21

## Objects in RPSL 3

---

- ⇒Attribute types
  - ⇒<object-name>
  - ⇒<as-number>
  - ⇒<ipv4-address>
  - ⇒<address-prefix>
  - ⇒<address-prefix-range>
  - ⇒<date>
  - ⇒<nic-handle>
  - ⇒.....
- ⇒Complete list in the draft

RPSL/22

## Objects in RPSL 4

---

- ⇒Address-prefix-range
  - ⇒address prefix followed by a range operator
- ⇒Range operators
  - ⇒<sup>+</sup>: inclusive more specifics
    - ⇒5.0.0.0/8<sup>+</sup>
  - ⇒<sup>~</sup>: exclusive more specifics
    - ⇒128.9.0.0/16<sup>~</sup>
  - ⇒<sup>n</sup>: length **n** more specifics
    - ⇒30.0.0.0/<sup>16</sup>
  - ⇒<sup>n-m</sup>: length **n-m** more specifics
    - ⇒30.0.0.0/<sup>24-32</sup>

RPSL/23

## Objects in RPSL 5

---

- ⇒list of attribute-value pairs
  - ⇒order is significant
- ⇒the 'class' attribute should be first
- ⇒line-continuation
  - ⇒space, tab, '+'
- ⇒comments
  - ⇒may be anywhere
  - ⇒begin with '#'

RPSL/24

## Object Names

---

- ⇒ Object names can have – or \_ in the middle
  - ⇒ RGNET-MAINT-MCI
- ⇒ Can have digits
  - ⇒ RGNET-MAINT-MCI\_1
- ⇒ Case insensitive
  - ⇒ rgnet-MaInT-MCI

RPSL/25

## Reserved Names

---

- ⇒ any as-any rs-any peeras
- ⇒ and or not
- ⇒ atomic from to at action accept
- announce except refine
- ⇒ networks into inbound outbound

RPSL/26

## Part II

---

### Contact Information

RPSL/27

## Contact Information

---

- ⇒ Person
- ⇒ Role
- ⇒ Mntner

RPSL/28

## Person Object

```
• Person: Clare Lancers
• address: Corofin
• phone: +1 23 123 # day time
• fax-no: +1 23 121
• e-mail: clancers@ripe.net
• nic-hdl: CL123-TEST
```

Person object  
information

```
• remarks: This object is automatically
• converted from RIPE181
• mnt-by: RIPE-NCC-MNT
• changed: clancers@ripe.net 19990122
• source: TEST
```

Auxiliary  
information

RPSL/29

## Auxiliary Information

**descr:** short free text description of the object  
**remarks:** free text comment attribute  
**tech-c:** Technical contact nic handles  
**admin-c:** Administrative contact nic handles  
**notify:** emails to send notification of changes  
**mnt-by:** maintainer authorized to do changes  
**changed:** <email> <date>  
**Source:** registry

RPSL/30

## Role object

```
• role: RIPE NCC Operations
• address: Singel 258
• 1016 AB Amsterdam
• The Netherlands
• phone: +31 20 535 4444
• fax-no: +31 20 545 4445
• e-mail: ops@ripe.net
• admin-c: OK65
• tech-c: RW488-RIPE
• tech-c: CF124
• nic-hdl: OPS4-RIPE
• notify: ops@ripe.net
• changed: roderik@ripe.net 19981208
• source: RIPE
•
```

RPSL/31

## Mntner Objects

• Example:  
•  
• **mntner:** RIPE-DBM-MNT  
• **descr:** Mntner for RIPE DBM objects  
• **admin-c:** AMRM1-RIPE  
• **tech-c:** RD132-RIPE  
• **upd-to:** ripe-dbm@ripe.net  
• **mnt-nfy:** ripe-dbm@ripe.net  
• **auth:** CRYPT-PW sprZvgymRKygk  
• **notify:** ripe-dbm@ripe.net  
• **mnt-by:** RIPE-DBM-MNT  
• **changed:** ripe-dbm@ripe.net 19980211  
• **source:** RIPE

Defines access control for other objects in database!

RPSL/32



## Template of a mntner object

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>mntner:</b>	<object-name>	mandatory, single, class key
<b>descr:</b>	<free-form>	mandatory, single
<b>auth:</b>	Described later	mandatory, multiple
<b>upd-to:</b>	<email-address>	mandatory, multiple
<b>mnt-nfy:</b>	<email-address>	optional, multiple
<b>tech-c:</b>	<nic-handle>	mandatory, multiple
<b>admin-c:</b>	<nic-handle>	optional, multiple
<b>remarks:</b>	<free-form>	optional, multiple
<b>notify:</b>	<email-address>	optional, multiple
<b>mnt-by:</b>	list of <mntner-name>	mandatory, multiple
<b>changed:</b>	<email-address> <date>	mandatory, multiple
<b>source:</b>	<registry-name>	mandatory, single

RPSL/33

## Auth Attribute

```
⇒auth: PGPKEY-<PGP Key ID>
⇒auth: CRYPT-PW lz1A7/JnfkTI
⇒auth: MAIL-FROM ripe-dbm@ripe.net
⇒auth: MAIL-FROM .*@ripe.net
⇒auth: NONE

⇒RIPE NCC implementation
⇒RIPE-157
⇒RIPE-189
⇒draft-ietf-rps-dbsec-pgp-authent-01.txt
```

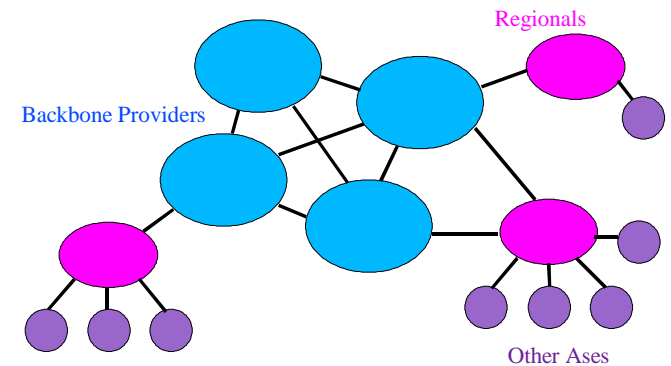
RPSL/34

## Part III

### Specifying Policy

RPSL/35

## Inter-AS Topology



RPSL/36

## AS Relationships

⇒ Customer–Regional Provider

- ⇒ Provider forwards traffic
- ⇒ advertises routes

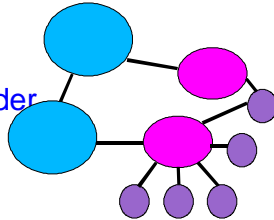
⇒ Peer–Peer

- ⇒ Mutual benefit

⇒ Regional Provider–Backbone Provider

- ⇒ Similar to Customer–Regional Provider

⇒ Typical routing policies implement these



RPSL/37

## Policies of an AS

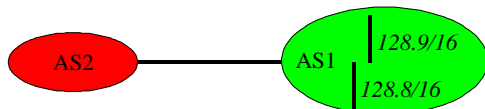
⇒ Use **aut-num** object

⇒ AS1 **originates** two routes

- ⇒ 128.8.0.0/16
- ⇒ 128.9.0.0/16

RPSL/38

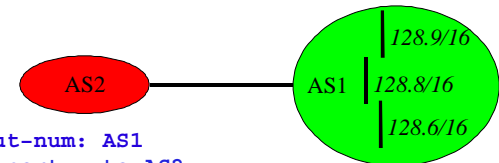
## Prefix Based



- aut-num: AS1
- export: to AS2
- announce {128.9.0.0/16, 128.8.0.0/16}
- 
- aut-num: AS2
- import: from AS1
- accept {128.9.0.0/16, 128.8.0.0/16}
- 
- **N.B.** Filtering based on Address-Prefix Set

RPSL/39

## Originate more routes?



- aut-num: AS1
- export: to AS2
- announce {128.9.0.0/16, 128.8.0.0/16,
- 128.6.0.0/16}
- 
- aut-num: AS2
- import: from AS1
- accept {128.9.0.0/16, 128.8.0.0/16,
- 128.6.0.0/16}
- 

RPSL/40

## Route Template

⇒Attribute	Value	Type
⇒		
⇒route:	<address-prefix>	mandatory, single, class key
⇒origin:	<as-number>	mandatory, single, class key
⇒member-of:	list of	
⇒	<route-set-names>	optional, multiple
⇒inject:		optional, multiple
⇒components:		optional, single
⇒aggr-bndry:		optional, single
⇒aggr-mtd:		optional, single
⇒export-comps:		optional, single
⇒holes:		optional, multiple
⇒		

RPSL/41

## route-set Objects

```

• route-set: rs-foo
• members: 128.9.0.0/16, 128.9.0.0/24,
            128.8.0.0/16
•
• descr:    some address prefixes
• mnt-by:   MAINT-RGNET
• tech-c:   RB366
• changed:  randy@psg.com 19960829
• source:   RADB
•
•
• route-set: rs-bar
• members:  128.7.0.0/16, rs-foo
•
•

```

RPSL/42

## Route-Set Template

⇒Attribute	Value	Type
⇒		
⇒route-set	<object-name>	Mandatory, single, class key
⇒		
⇒members	list of	optional, multi-values
⇒	<address-prefix-range>	
⇒	<route-set-name> or	
⇒	<route-set-name><range-operator> or	
⇒	rs-any	
⇒		
⇒mbrs-by-ref	list of <mntner-names>	optional, multi-valued
⇒	or ANY	
⇒		

RPSL/43

## Indirect Members of Route-Set

```

• route-set:  RS-ANS-IGP_ONLY
• descr:      ANS IGP aggregates
• mbrs-by-ref: any
•
•
• route:      207.25.17.0/24
• origin:     AS1675
• member-of:  RS-ANS-IGP_ONLY
• mnt-by:     MNT-ANS
•
• route:      192.157.69.0/24
• origin:     AS1675
• member-of:  RS-ANS-IGP_ONLY
• mnt-by:     MNT-ANS
•
•

```

RPSL/44

## Restricted Indirect Members

```
• route-set: RS-ANS-IGP_ONLY
• descr:    ANS IGP aggregates
• mbrs-by-ref: MNT-ANS, MNT-CENGIZ
•
•
• route:    207.25.17.0/24
• origin:   AS1675
• member-of: RS-ANS-IGP_ONLY
• mnt-by:    MNT-ANS
•
• route:    192.157.69.0/24
• origin:   AS1675
• member-of: RS-ANS-IGP_ONLY
• mnt-by:    MNT-CURTIS
•
```

RPSL/45

## Direct & Indirect Members

```
• route-set: RS-ANS-IGP_ONLY
• descr:    ANS IGP aggregates
• members:  207.25.17.0/24, 207.25.16.0/24,
            207.25.20.0/24
• mbrs-by-ref: MNT-ANS
•
• route:    207.25.17.0/24
• origin:   AS1675
• member-of: RS-ANS-IGP_ONLY
• mnt-by:    MNT-ANS
•
• route:    192.157.69.0/24
• origin:   AS1675
• member-of: RS-ANS-IGP_ONLY
• mnt-by:    MNT-ANS
•
```

RPSL/46

## Direct Members

The "**member-of**" attribute of the **route** object is an additional mechanism for specifying the members indirectly.

If an address prefix is listed in the **members** attribute of a route-set, it is a member of that route set.

The **route** object corresponding to this address prefix does not need to contain a **member-of** attribute referring to this set name.

RPSL/47

## Per route-set

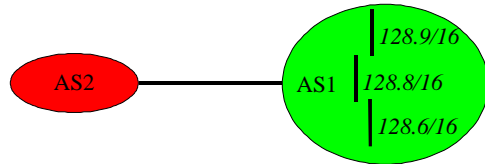
```
route-set:  rs-red
members:    128.6.0.0/16, 128.9.0.0/16,
            128.8.0.0/16
```

```
aut-num: AS1
export:  to AS2    announce rs-green
```

```
aut-num: AS2
import:  from AS1  accept  rs-green
```

RPSL/48

## Policy and route-set



aut-num: AS1  
⇒ export: to AS2  
⇒ announce rs-green  
⇒ aut-num: AS2  
⇒ import: from AS1

RPSL/49

## Range Operators & route-sets

```
route-set: rs-martians
descr:    most ASes do not import these routes
members:  127.0.0.0/8^+,
          10.0.0.0/8^+,      172.16.0.0/20^+,
          192.168.0.0/16^+,  192.0.2.0/24^+,
          128.0.0.0/16^+,    191.255.0.0/16^+,
          192.0.0.0/24^+,    223.255.255.0/24^+,
          224.0.0.0/3^+
```

Inclusive more specifics: ^+  
Exclusive more specifics: ^-  
Length n more specifics: ^n  
Length n-m more specifics: ^n-m

RPSL/50

## Route Object 1

⇒ Subset of a route !  
⇒ The **route** and **origin** attributes == class key

RPSL/51

## Route Object 2

```
Route: 193.0.0.0/23
origin: AS3333

route: 128.8.0.0/16
origin: AS1

route: 128.8.0.0/16
origin: AS2
```

RPSL/52

## Route Object 3

```
route: 156.36.0.0/16
origin: AS2914
```

Policy  
information

```
member-of: RS-VERIO
```

```
descr: my routes
```

```
mnt-by: MAINT-RGNET
```

```
tech-c: RB366
```

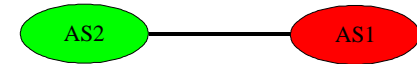
```
changed: randy@psg.com 19960829
```

```
source: RADB
```

- Route 156.36.0.0/16
  - is originated by AS2914
  - is a member of set RS-VERIO

RPSL/53

## AS Numbers in Policy

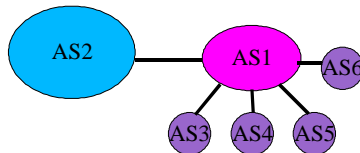


- route: 128.9.0.0/16 route: 128.8.0.0/16
- origin: AS1 origin: AS1
- 
- aut-num: AS1
- export: to AS2 announce AS1
- 
- aut-num: AS2
- import: from AS1 accept AS1
- 

AS1 == {128.9.0.0/16, 128.8.0.0/16}

RPSL/54

## Cumbersome?



- aut-num: AS1
- export: to AS2 announce AS1 OR AS3 OR ... AS6
- 
- aut-num: AS2
- import: from AS1 accept AS1 OR AS3 OR ... AS6

RPSL/55

## as-set Objects

- as-set: AS-SESQUISTUB
- descr: Single Homed Sesquinet Customer ASs
- members: AS1832, AS2712, AS302, AS3526, AS8
- tech-c: SB98
- mnt-by: MAINT-AS114
- source: RADB
- 

Same flexibility as route-set objects

RPSL/56

## AS-Set Template

Attribute	Value	Type
⇒ <b>as-set</b>	<object-name>	mandatory, single, class key
⇒ <b>members</b>	list of	optional, multiple
	<as-numbers> or	
	<as-set-names> or	
	as-any	
⇒ <b>mbrs-by-ref</b>	list of	optional, multiple
	<mntner-names> or	
	ANY	

RPSL/57

## Indirect as-sets

```

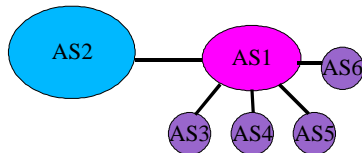
as-set:      as-aads-mlpa
descr:       MLPA participants at the AADS NAP
mbrs-by-ref: any
admin-c:     Andrew Schmidt
tech-c:      Mark Cnota
notify:      mlpa-participants@aads.net
mnt-by:      MAINT-RSPEER
changed:     auto-mlpa@aads.net 19971123
source:      RADB

aut-num:     AS4550
member-of:   as-aads-mlpa

aut-num:     AS683
member-of:   as-aads-mlpa
    
```

RPSL/58

## Using as-set objects

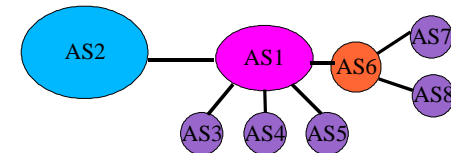


```

⇒ As-set:  AS1:AS-Customers
⇒ members: AS1, AS3, AS4, AS5, AS6
⇒
⇒ aut-num: AS1
⇒ export:  to AS2  announce AS1:AS-Customers
AS1:AS-Customers
⇒
    
```

RPSL/59

## Using as-set objects 2

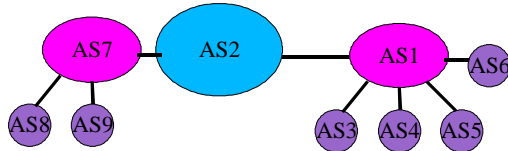


```

• as-set:  AS1:AS-Customers
• members: AS1, AS3, AS4, AS5, AS6:AS-Customers
•
• aut-num: AS1
• export:  to AS2  announce AS1:AS-Customers
•
• aut-num: AS2
• import:  from AS1  accept AS1:AS-Customers
    
```

RPSL/60

## Cumbersome?

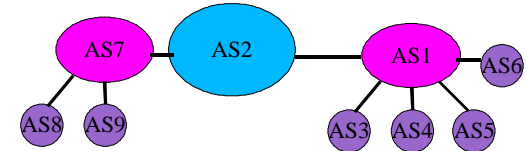


```

• as-set: AS1:AS-Customers
• members: AS1, AS3, AS4, AS5, AS6
•
• as-set: AS7:AS-Customers
• members: AS7, AS8, AS9
•
• aut-num: AS2
• import: from AS1 accept AS1:AS-Customers
• import: from AS7 accept AS7:AS-Customers
  
```

RPSL/61

## PeerAS



```

as-set: AS2:AS-Customers
members: AS1, AS7

aut-num: AS2
import: from AS2:AS-Customers
       accept PeerAS:AS-Customers
  
```

Same as:

```

from AS1 accept AS1:AS-Customers
from AS7 accept AS7:AS-Customers
  
```

RPSL/62

## PeerAS

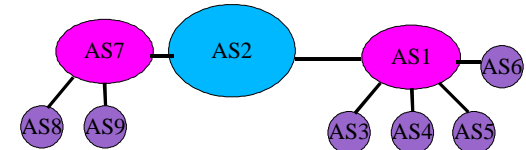
⇒ Keyword: **PeerAS**

⇒ Used in **import** attribute  
 ⇒ instead of the AS number of the peer AS

⇒ Useful when using AS expression

RPSL/63

## PeerAS



```

as-set: AS2:AS-Customers
members: AS1, AS7
  
```

```

aut-num: AS2
import: from AS2:AS-Customers
       accept PeerAS:AS-Customers
  
```

Same as:

```

from AS1 accept AS1:AS-Customers
from AS7 accept AS7:AS-Customers
  
```

RPSL/64



## Notations

AS Numbers	<b>AS</b> 2914
Address prefixes	156.36.0.0/16
Route set names	<b>RS-VERIO</b>
AS set names	<b>AS-REEL</b>

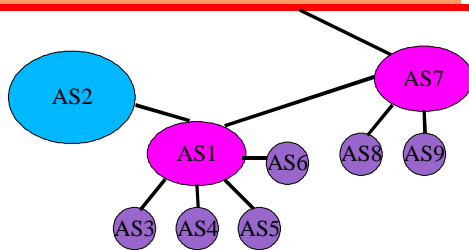
RPSL/65

## Predefined Set Objects

- ⇒RS-ANY, rs-any
- ⇒AS-ANY, as-any
- ⇒Route-set context
  - ⇒AS number: **AS**x == routes originated by **AS**x
  - ⇒as-set: **AS-X** == routes originated by the AS's in **AS-X**

RPSL/66

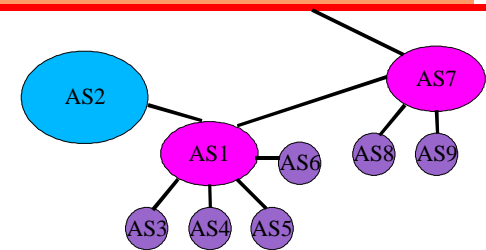
## Complex Example



Solution ?

RPSL/67

## AS Path Based



- import: from AS1 accept <^AS1 .\* AS8\$>
- 
- import: from AS1 accept <^AS1 AS1:AS-Customers\*\$>

No route prefixes here!

RPSL/68

## AS Path Regular Expressions

AS1	AS1
as-foo	any AS in as-foo
X*	0 or more occurrence of X
X+	1 or more occurrence of X
X?	0 or 1 occurrence of X
^	beginning of path
\$	end of path
X Y	X or Y
X Y	X followed by Y

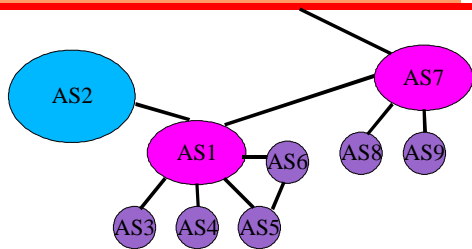
RPSL/69

## AS Path Regular Expressions

- ⇒ Policy filter
  - ⇒ only when the expression is between '<' and '>'
- ⇒ Regular expressions
  - ⇒ the alphabet of AS numbers
- ⇒ Router can check
  - ⇒ BGP: AS\_PATH
  - ⇒ IDRP: RD\_PATH
- ⇒ Regular Expression Operators

RPSL/70

## AS Path RE Example



- ⇒ `<^AS1+ AS1:AS-Customers* $>` matches
  - ⇒ AS1
  - ⇒ AS1 AS3
  - ⇒ AS1 AS4
  - ⇒ AS1 AS5 AS6
  - ⇒ AS1 AS1 AS5 AS5 AS6

RPSL/71

## Composite Policy Filters

- ⇒ NOT
  - ⇒ negation
- ⇒ AND
  - ⇒ intersection
- ⇒ OR
  - ⇒ union

RPSL/72

## Composite Policy Filters 2



⇒AS1 == {128.8.0.0/16, 128.9.0.0/16}  
⇒rs-red == {128.6.0.0/16, 128.9.0.0/16}  
⇒  
⇒AS1 OR rs-red == {128.6.0.0/16, 128.8.0.0/16, 128.9.0.0/16}  
⇒  
⇒AS1 AND rs-red == {128.9.0.0/16}  
⇒  
⇒AS1 AND NOT rs-red == {128.8.0.0/16}  
⇒

RPSL/73

## AS numbers = routes

```
• aut-num: AS1
• import: from AS1
• accept (AS1 OR rs-red) AND NOT
  {0.0.0.0/0}
•
•
•
• AS1 == {128.9.0.0/16, 128.9.0.0/16}
• rs-red == {128.6.0.0/16, 128.9.0.0/16}
•
• AS1 OR rs-red == {128.6.0.0/16, 128.9.0.0/16, 128.9.0.0/16}
•
• AS1 AND rs-red == {128.9.0.0/16}
•
• AS1 AND NOT rs-red == {128.8.0.0/16}
•
• N.B. AS numbers & as-set names == routes
```

RPSL/74

## Prefix Length Based Policy

```
aut-num: AS1
import: from as-any
      accept ANY AND NOT { 192.168.0.0/16^+ }
```

N.B. Filter == Address-Prefix Set;  
Composite Policy

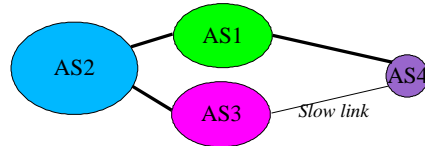
RPSL/75

## Example of Actions

⇒Two links:  
⇒one is fast  
⇒the other is slow

RPSL/76

## Choice of links

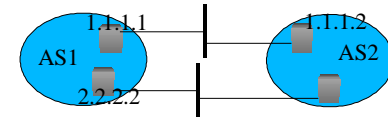


- aut-num: AS4
- import: from AS1 action pref = 10; accept ANY
- import: from AS3 action pref = 15; accept ANY

Smaller the number, higher the preference!

RPSL/77

## Peering Choice



- aut-num: AS1
- import: from AS2 at 2.2.2.2
- action pref = 10;
- accept AS2
- import: from AS2 1.1.1.2 at 1.1.1.1
- action pref = 5;
- accept AS2

RPSL/78

## Specifying Actions

- ⇒ RPSL policy actions
  - ⇒ set or modify route attributes (BGP attributes)
  - ⇒ instruct routers to do special operations
    - ⇒ route flap dampening
- ⇒ Route attributes ?
  - ⇒ RPSL dictionary

RPSL/79

## Specifying Actions 2

- ⇒ Syntax of a policy action
  - ⇒ x.method(arguments)
  - ⇒ x "op" argument
- ⇒ Terminated by semicolon ';'
  - ⇒ Composite policy actions possible
    - ⇒ evaluated left-to-right

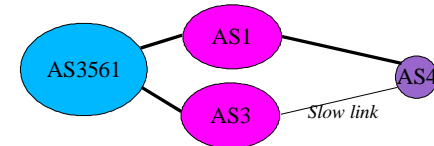
RPSL/80

## Actions

```
⇒import: from ... action XXX; accept ...
⇒export: to ... action XXX; announce ...
⇒
⇒med = 0;
⇒med = igp_cost;
⇒
⇒community.append(NO_EXPORT, 10250,
3561:90);
⇒community.delete(NO_EXPORT);
⇒
⇒aspath.prepend(AS1, AS1, AS1);
```

RPSL/81

## Example; community



⇒AS4 wants AS3561 to prefer AS1 path  
⇒AS3561 prefers routes with

- ⇒no community
- ⇒with community 3561:90
- ⇒with community 3561:80
- ⇒with community 3561:70

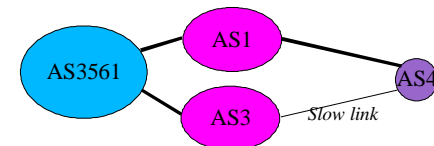
RPSL/82

## AS3561's Policies

```
aut-num: AS3561
import: from AS-ANY
      action pref = 30;
      accept community(3561:70)
import: from AS-ANY
      action pref = 20;
      accept community(3561:80)
import: from AS-ANY
      action pref = 10;
      accept community(3561:90)
import: from AS-ANY
      action pref = 0;
      accept ANY
```

RPSL/83

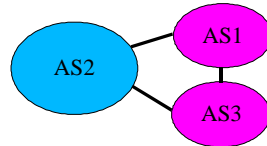
## AS4's Policies



- aut-num: AS4
- export: to AS1 action community.= {3561:90};
- to AS3 action community.= {3561:80};
- announce AS4
- 

RPSL/84

## Policy Example

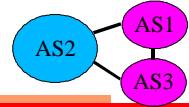


⇒ Possibilities

- 1 each link is private (i.e. no transit)
- 2 case 1 + AS2 as backup transit provider
- 3 case 2 + AS2 can use the link as backup to AS1
- 4 case 3 + AS2 can use the link as backup to AS3

RPSL/85

## Case 1



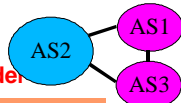
```

• aut-num: AS1
• import: from AS3 accept AS3
• import: from AS2 accept AS2
• export: to AS3 announce AS1
• export: to AS2 announce AS1
•
• aut-num: AS3
• import: from AS1 accept AS1
• import: from AS2 accept AS2
• export: to AS1 announce AS3
• export: to AS2 announce AS3
•
• aut-num: AS2
• import: from AS1 accept AS1
• import: from AS3 accept AS3
• export: to AS1 announce AS2
• export: to AS3 announce AS2
  
```

RPSL/86

## Case 2

AS2 as backup transit provider



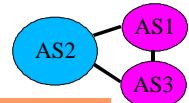
```

• aut-num: AS2
• import: from AS1 accept AS1
• import: from AS3 accept AS3
• export: to AS1 announce ANY
• export: to AS3 announce ANY
•
• aut-num: AS1
• import: from AS2 action pref = 10; accept ANY
• import: from AS3 action pref = 5; accept AS3
• export: to AS2 announce AS1
• export: to AS3 announce AS1
•
• aut-num: AS3
• import: from AS2 action pref = 10; accept ANY
• import: from AS1 action pref = 5; accept AS1
• export: to AS2 announce AS3
• export: to AS1 announce AS3
  
```

RPSL/87

## Case 3

AS3 has backup to AS1



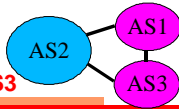
```

• aut-num: AS2
• import: from AS1 action pref = 10; accept AS1
• import: from AS3 action pref = 20; accept AS1
• import: from AS3 accept AS3
•
• aut-num: AS3
• export: to AS2 announce AS3 or AS1
•
•
  
```

RPSL/88

## Case 4

AS2 is backup for AS1 to AS3



```
• aut-num: AS2
• import: from AS1 action pref = 10; accept AS1
• import: from AS3 action pref = 20; accept AS1
• import: from AS3 action pref = 10; accept AS3
• import: from AS1 action pref = 20; accept AS3
•
• aut-num: AS3
• export: to AS2 announce AS3 or AS1
•
• aut-num: AS1
• export: to AS2 announce AS1 or AS3
•
•
```

RPSL/89

## Ambiguity Resolution

⇒ Two or more peering expressions

⇒ describe the same peering

⇒ Which is used ?

⇒ Specification–order rule

⇒ the first peering specification is always used

RPSL/90

## Ambiguity Resolution

Example:

```
aut-num: AS1
import: from AS2 action pref = 2; accept AS4
import: from AS2 action pref = 1; accept AS4 OR AS5
```

AS4's routes are accepted from AS2 with preference 2

AS5's routes are accepted from AS2 with preference 1

RPSL/91

## RPSL

⇒ RPSL

⇒ compact representation of policy

⇒ accurate representation of policy

⇒ Training materials

⇒ <http://www.isi.edu/ra/rps/training>

⇒ <http://www.ripe.net/db/rps/training>

RPSL/92

## RIPE NCC RPSL server

---

- ⇒ [rpslii.ripe.net](http://rpslii.ripe.net)
- ⇒ **Statistics:**
  - ⇒ Updates
  - ⇒ Queries
- ⇒ **Demonstration**

RPSL/93

## Part IV

---

- ⇒
- ⇒
- ⇒
- ⇒ **Set Objects**

RPSL/94

## Set Objects

---

- ⇒ Sets of routes, autonomous systems, etc.
  - ⇒ **route-set**
  - ⇒ **as-set**
  - ⇒ **filter-set**
  - ⇒ **peering-set**
  - ⇒ **rtr-set**
- ⇒ **Specify members**
  - ⇒ directly
  - ⇒ indirectly

RPSL/95

## Set Objects 2

---

- ⇒ **Set names**
  - ⇒ **as-customers**
  - ⇒ **rs-partner**
  - ⇒ .....
- ⇒ **Hierarchical set names**
  - ⇒ **authorisation**
    - ⇒ AS1:AS-CUSTOMERS
    - ⇒ AS1:RS-EXPORT:AS2

RPSL/96



## Set Objects 3

- ⇒Route-set
- ⇒as-set
- ⇒filter-set
- ⇒peering-set
- ⇒rtr-set

RPSL/97

## Filter-Set Objects

Examples:

```
filter-set:  fltr-mine
filter:      {5.0.0.0/8, 6.0.0.0/8}

filter-set:  fltr-yours
filter:      (AS1 or fltr-mine) and <AS2>
```

RPSL/98

## Filter-Set Objects 2

Set of routes that are matched by a filter

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>filter-set</b>	<i>&lt;object-name&gt;</i>	mandatory, single, class key
<b>filter</b>	<i>&lt;filter&gt;</i>	mandatory, single

The **filter** attribute == policy filter.

Policy filter => matches a subset of routes.

Filter set names: "**fltr-**"

RPSL/99

## Filter-Set Objects 3

- ⇒Policy filters
  - ⇒ANY
  - ⇒Address-Prefix Set
  - ⇒Route Set Name
  - ⇒AS Path Regular Expressions
  - ⇒Composite Policy Filters
  - ⇒Routing Policy Attributes
  - ⇒Filter Set Name

RPSL/100

## Address Prefix Set

---

- ⇒ List of address-prefixes between '{', '}'
- ⇒ Policy filter matches those routes in the set
- ⇒ range operators are optional
- ⇒ Example:
  - ⇒ filter-set: fltr-mine
  - ⇒ filter: {128.8.0.0/16, 128.9.0.0/16}
  - ⇒ filter-set: fltr-default
  - ⇒ filter: {0.0.0.0/^19-32}

RPSL/101

## Route Set Name

---

- ⇒ Route set name
  - ⇒ matches the routes that are members of the set
- ⇒ Route set name
  - ⇒ name of a **route-set** object
  - ⇒ AS number
  - ⇒ name of an **as-set** object
- ⇒ Can be followed by a range operator
  - ⇒ {5.0.0.0/8, 6.0.0.0/8}^+
  - ⇒ AS1^~

RPSL/102

## Routing Policy Attributes

---

- ⇒ Can use the values of other attributes
- ⇒ See the RPSL dictionary
- ⇒ Example: BGP community attribute
  - ⇒ filter-set: fltr-mci-communities
  - ⇒ filter: community(3561:100, 3561:90, 3561:80)
- ⇒ Evaluated before **AND, OR, NOT**

RPSL/103

## Filter Set Name

---

- ⇒ Matches a set of routes
  - ⇒ matched by **filter** of filter set
- ⇒
  - ⇒ Inclusion possible
    - ⇒ can use name of a filter set in a filter

RPSL/104

## Rtr-Set Objects

Router-Set Name: "**rtrs-**"

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>rtr-set</b>	<object-name>	mandatory, single, class key
<b>members</b>	list of < <b>inet-rtr</b> -names> or < <b>rtr-set</b> -names> or <ipv4_addresses>	optional, multiple
<b>mbrs-by-ref</b>	list of < <b>mntner</b> -names> ANY	

RPSL/105

## Peering Set Object

Router 9.9.9.1 imports 128.9.0.0/16 from 9.9.9.2 and 9.9.9.3

**peering-set:** prng-one  
**peering:** AS3 at 9.9.9.1

**peering-set:** prng-two  
**peering:** prng-one  
**peering:** AS2 at 9.9.9.1

**aut-num:** AS1  
**import:** from prng-two accept {128.9.0.0/16}

RPSL/106

## Peering-Set Object 2

Defines a set of peerings

Peering Set name: "**prng-**"

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>peering-set</b>	<object-name>	mandatory, single, class key
<b>peering</b>	<peering>	mandatory, multiple

The **peering** attribute defines a peering  
=> used to import or export routes

RPSL/107

## Peering Specification

<as-expression> [<router-expression-1>] [at  
<router-expression-2>]

| <peering-set-name>

where

<as-expression>: AS numbers, AS sets, **AND**, **OR**, **EXCEPT**

<router-expression-\*>: IP addresses, **inet-rtr**'s, **rtr-set**'s, **AND**, **OR**,  
**EXCEPT**

RPSL/108

## Part V

---

### Autonomous Systems

RPSL/109

## aut-num Objects

---

```
• aut-num: AS4591
• as-name: Syra-NET
• import: from AS4590
•         action pref=1;
•         accept AS4590
• export: to AS4590
•         announce AS4591
• default: to AS4590
•         action pref=1;
•         networks {140.222.0.0/16}
• admin-c: Warren Lavallee
• tech-c: Warren Lavallee
• mnt-by: MAINT-AS4591
• changed: warren@Syra.NET 19950522
• source: RADB
•
```

RPSL/110

## Aut-num Object

---

Specifies routing policies

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>aut-num</b>	<as-number>	mandatory, single, class key
<b>as-name</b>	<object-name>	mandatory, single
<b>member-of</b>	list of <as-set-names>	optional, multiple
<b>import</b>	defined later	optional, multiple
<b>export</b>	defined later	optional, multiple
<b>default</b>	defined later	optional, multiple

RPSL/111

## Specifying Policies

---

- ⇒RPSL allows policies based on:
  - ⇒prefix
  - ⇒AS Path
  - ⇒community
  - ⇒prefix-length
  - ⇒future attributes thru its dictionary
- ⇒Structured Policy possible

RPSL/112

## Specifying Policy in Aut-Num

- ⇒import
- ⇒export
- ⇒default

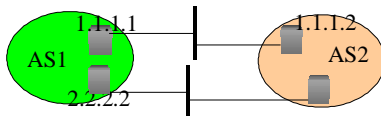
RPSL/113

## Import

- ⇒import:
  - from <peering-1> [action <action-1>]
  - ...
  - from <peering-N> [action <action-N>]
  - accept <filter>
- ⇒Set of routes matched by **filter**
  - ⇒imported from all peers in **peerings**
- ⇒While importing routes at <peering-M>
  - ⇒<action-M> is done

RPSL/114

## Choosing a Peering



- aut-num: AS1
- import: from AS2 at 2.2.2.2
- action pref = 10;
- accept AS2
- import: from AS2 1.1.1.2 at 1.1.1.1
- action pref = 5;
- accept AS2

RPSL/115

## Export

- ⇒export:
  - to <peering-1> [action <action-1>]
  - ...
  - to <peering-N> [action <action-N>]
  - announce <filter>
- ⇒Set of routes matched by **filter**
  - ⇒exported to all peers in <peerings>
- ⇒While exporting routes at <peering-M>
  - ⇒<action-M> is done

RPSL/116

## Default

- ⇒ default:
- ⇒ to <peering> [action <action>] [networks <filter>]
- ⇒ Local AS defaults to the AS in <peering>
- ⇒ <action> == attributes of defaulting
- ⇒ <filter> == policy filter
- ⇒ Router only uses the default policy
  - ⇒ if it received the routes matched by <filter>

RPSL/117

## Examples of Default

AS1 defaults to AS2 and uses 128.9.0.0/16

```
aut-num:      AS1
default:      to AS2 networks { 128.9.0.0/16 }
```

AS1 defaults to AS2 and AS3, but prefers AS2 over AS3

```
aut-num:      AS1
default:      to AS2 action pref = 1;
default:      to AS3 action pref = 2;
```

RPSL/118

## Import, Export

- ⇒ Other Routing Protocols
- ⇒ Multi-Protocol Routing Protocols
- ⇒ Injecting Routes between protocols

RPSL/119

## Import, Export 2

Complete syntax of **import**, **export** attributes:

```
import: [protocol <protocol-1>] [into <protocol-2>]
        from <peering-1> [action <action-1>]
        .....
        from <peering-N> [action <action-N>]
        accept <filter>
```

```
export: [protocol <protocol-1>] [into <protocol-2>]
        to <peering-1> [action <action-1>]
        .....
        to <peering-N> [action <action-N>]
        announce <filter>
```

RPSL/120

## Import, Export 3

---

- ⇒ Valid protocols
  - ⇒ in RPSL dictionary
- ⇒ Default protocol is Exterior Gateway Protocol
  - ⇒ BGP

RPSL/121

## Part VI

---

## Internet Router

RPSL/122

## Inet-Rtr Object

---

```
• inet-rtr:      c56-11.t3.ans.net
• local-as:      AS1664
• ifaddr:        140.222.56.200 masklen 26
• ifaddr:        140.222.56.65  masklen 26
• ifaddr:        204.151.29.9   masklen 30
• peer:          BGP4 140.222.56.199
  asno (AS1673)
• peer:          BGP4 140.222.56.66
  asno (AS1673)
• admin-c:       Steve Heimlich
• tech-c:        Selina Priestley
• mnt-by:        ANS
• changed:       configs@ans.net 19970320
• source:        ANS
•
```

RPSL/123

## Inet-Rtr Object 2

---

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>inet-rtr</b>	<dns-name>	Mandatory, single, class key
<b>alias</b>	<dns-name>	Optional, multiple
<b>local-as</b>	<as-number>	Mandatory, single
<b>ifaddr</b>	Described below	Mandatory, multiple
<b>peer</b>	Described later	Optional, multiple
<b>member-of</b>	list of <rtr-set-names>	Optional, multiple
<b>ifaddr:</b>	<ipv4-address> masklen <integer> [action <action>]	

RPSL/124

## Inet-Rtr Object 3

---

The **peer** attribute:

```
<protocol> <ipv4-address>    <options>
| <protocol> <inet-rtr-name>    <options>
| <protocol> <rtr-set-name>     <options>
| <protocol> <peering-set-name> <options>
```

**<protocol>** == defined in RPSL dictionary

RPSL/125

## Inet-Rtr Object 4

---

Example:

```
rtr-set:    rtrs-ibgp-peers
members:    1.1.1.1, 2.2.2.2, 3.3.3.3
```

```
peering-set: prng-ebgp-peers
peering:     AS3334 192.87.45.195
peering:     AS3335 192.87.45.196
```

RPSL/126

## Inet-Rtr Object 5

---

Example of inet-rtr object with peering groups:

```
inet-rtr:    Amsterdam.ripe.net
alias:       amsterdam1.ripe.net
local-as:    AS3333
ifaddr:      192.87.45.190 masklen 24
ifaddr:      192.87.4.28   masklen 24
ifaddr:      193.0.0.222   masklen 27
ifaddr:      193.0.0.158   masklen 27
peer:        BGP4 rtrs-ibgp-peers asno(AS3333), flap_damp()
peer:        BGP4 prng-ebgp-peers asno(PeerAS), flap_damp()
```

RPSL/127

## Part VII

---

## Advanced Features

RPSL/128



## Advanced Features

- ⇒ Extending RPSL
  - ⇒ "dictionary" object
- ⇒ Aggregating Routes
- ⇒ Static Routes
- ⇒ Structured Policy

RPSL/129

## Extending RPSL

- ⇒ "dictionary" class
- ⇒ New attributes
- ⇒ New object types (classes)
- ⇒ Changing the syntax of attributes

RPSL/130

## "dictionary" Object

```
• dictionary: RPSL
• rp-attribute: pref # smaller values are preferred
•               operator=(integer[0, 65535])
• rp-attribute: med # BGP multi_exit_discriminator
  attribute
•               operator=(integer[0, 65535])
•               operator=(enum[igp_cost])
• typedef:      community_elm union
•               integer[1, 4294967200],
•               enum[internet, no_export,
  no_advertise],
•               list[2:2] of integer[0, 65535]
• rp-attribute: community # BGP community attribute
•               operator.=(community_elm)
•               operator()(community_elm, ...)
• ...
```

RPSL/131

## "dictionary" Object 2

- ⇒ "dictionary" object
  - ⇒ main way to extend RPSL
- ⇒ Dictionary objects define
  - ⇒ routing policy attributes
  - ⇒ types
  - ⇒ routing protocols
- ⇒ Routing policy attribute
  - ⇒ rp-attribute

RPSL/132

## "dictionary" Object 3

- ⇒ The `rp`-attribute
  - ⇒ actual protocol attributes
    - ⇒ e.g. BGP path attributes
  - ⇒ router features
    - ⇒ e.g. BGP route flap dampening
- ⇒ `rp`-attributes
  - ⇒ accessed using access methods
  - ⇒ describe policy filters and actions
- ⇒ Well-known dictionary object: **RPSL**
  - ⇒ All tools use this by default

RPSL/133

## Aggregation

- `route:` 128.8.0.0/15
- `origin:` AS1
- `components:` {128.8.0.0/15^-}
- `aggr-mtd:` outbound AS-ANY
- `inject:` at 1.1.1.1 action dpa = 100;
- `inject:` at 1.1.1.2 action dpa = 110;
- 

RPSL/134

## Static Routes

- `route:` 128.7.0.0/16
- `origin:` AS1
- `inject:` at 7.7.7.1
  - `action next-hop = 7.7.7.2; cost = 10;`
  - `upon static`
- `inject:` at 7.7.7.1
  - `action next-hop = 7.7.7.3; cost = 20;`
  - `upon static`
- 

RPSL/135

## Structured Policy

- `aut-num:` AS3561
- `import:` { from AS-ANY action pref = 30;
  - `accept community(3561:70);`
  - `from AS-ANY action pref = 20;`
  - `accept community(3561:80);`
  - `} refine {`
    - `from AS1 accept`
    - `AS1:AS-Customers;`
    - `from AS2 accept AS2;`
    - `from AS3 accept ...;`
  - `}`
- 
- 
- 

RPSL/136

## Structured Policy

```
• aut-num: AS3561
• import: {
•     from AS-ANY
•     accept any and not
•     rs-martians;
•     } refine {
•         from AS-ANY action pref = 30;
•         accept community(3561:70);
•         from AS-ANY action pref = 20;
•         accept community(3561:80);
•     } refine {
•         from AS1 accept
•         AS1:AS-Customers;
•         from AS2 accept AS2;
•         from AS3 accept ...;
•     }
```

RPSL/137

## Structured Policy

```
• aut-num: AS3561
• import: { from AS-ANY action pref = 30;
•     accept community(3561:70);
•     from AS-ANY action pref = 20;
•     accept community(3561:80);
•     } refine {
•         from AS1 accept AS1:AS-Customers;
•     } except {
•         from AS2 accept AS2;
•         from AS3 accept AS3;
•     }
• AS1:AS-Customers contains AS2 and AS3
```

RPSL/138

## More Complete Example

```
aut-num: AS2764
as-name: ASN-CONNECT-NET
descr: connect.com.au Pty Ltd
import: {
    from AS-ANY action community .= {2764:65408};
    accept ANY AND NOT { 0.0.0.0/0 };
} refine {
    from AS-ANY action community={internet}; pref=0;
    accept community(2764:65280,...,2764:65412);
    from AS-ANY action pref=25;
    accept community(2764:3)
        AND NOT AS2764:RS-PROVIDER^-,
    ...
    from AS-ANY action pref=0; accept ANY;
} refine {
    from AS2764:AS-GLOBAL
    accept PeerAS AND <^PeerAS$>;
    ...
}
```

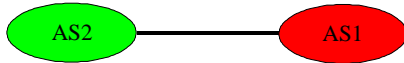
RPSL/139

## Practical Example

```
⇒RAToolSet
⇒http://www.isi.edu/ra/RAToolSet/
⇒RtConfig
```

RPSL/140

## AS Numbers in Policy



⇒ route: 128.9.0.0/16    route: 128.8.0.0/16  
 ⇒ origin: AS1            origin: AS1  
 ⇒  
 ⇒ aut-num: AS1  
 ⇒ export: to AS2    announce AS1  
 ⇒  
 ⇒ aut-num: AS2  
 AS1 := {128.9.0.0/16, 128.8.0.0/16}  
 ⇒ import: from AS1 accept AS1

RPSL/141

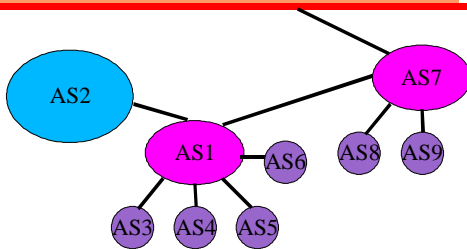
## RtConfig: A tool to configure routers

```

% RtConfig
RtConfig> @RtConfig export AS1 1.1.1.1 AS2 2.2.2.2
no access-list 1
access-list 1 permit ip 128.8.0.0 0.0.0.0 255.255.0.0 0.0.0.0
access-list 1 permit ip 128.9.0.0 0.0.0.0 255.255.0.0 0.0.0.0
access-list 1 deny ip 0.0.0.0 255.255.255.255 0.0.0.0
255.255.255.255
!
no route-map foo
route-map foo permit 1
match ip address 1
!
!
  
```

RPSL/142

## AS Path Based



⇒ import: from AS1 accept <^AS1 .\* AS8\$>  
 ⇒  
 ⇒ import: from AS1 accept <^AS1  
 AS1:AS-Customers\* \$>    **No prefixes here!**

RPSL/143

## RtConfig

```

• RtConfig> @RtConfig import AS2 1.1.1.1 AS1
  2.2.2.2
• !
• no ip as-path access-list 1
• ip as-path access-list 1 permit
  ^_1((_[0-9]+))*_8$
• !
• no route-map foo
• route-map foo permit 1
• match as-path 1
• !
• router bgp 2
• neighbor 2.2.2.2 route-map foo in
•
•
  
```

RPSL/144

## RtConfig

```
• aut-num: AS1
• export: to AS2
      announce AS2764 AND NOT { 0.0.0.0/0 }
      AND <^AS2764*$>
      AND NOT community({2764,1})
•
•
•
•
• RtConfig> @RtConfig export AS1 1.1.1.1 AS2
  2.2.2.2
• access-list 1 permit ip 210.8.248.0 0.0.0.0 255.255.248.0
  0.0.0.0
• ...
• access-list 1 permit ip 210.9.2.0 0.0.0.0 255.255.254.0
  0.0.0.0
• access-list 1 deny ip 0.0.0.0 255.255.255.255 0.0.0.0
  255.255.255.255
• !
• no ip as-path access-list 1
```

RPSL/145

## RtConfig (cont)

```
• ip bgp-community new-format
• no community-list 1
• ip community-list 1 deny 2764:1
• ip community-list 1 permit internet
• !
• no route-map foo
• route-map foo permit 1
• match as-path 1
• match community 1
• match ip address 1
• !
• router bgp 1
• neighbor 0.0.0.0 route-map foo out
•
```

RPSL/146

## RtConfig

```
• @RtConfig import AS1 1.1.1.1 AS2 2.2.2.2
• @RtConfig export AS1 1.1.1.1 AS2 2.2.2.2
• @RtConfig networks AS1
• @RtConfig default AS1 AS2
• @RtConfig pkt_filter "eth0" AS1 1.1.1.1 AS2
  2.2.2.2
• @RtConfig outbound_pkt_filter "eth0" AS1 1.1.1.1 AS2
  2.2.2
• @RtConfig static2bgp AS1 1.1.1.1
•
• Not yet available:
• @RtConfig inet-rtr c56-11.t3.ans.net
•
• Configuration formats:
• cisco, bay, gated, rsd
```

RPSL/147

## Acknowledgements

- ⇒ Cengiz Alaettinoglu
  - ⇒ USC Information Sciences Institute
  - ⇒ Provided slides from which these slides are derived
  - ⇒ But any errors are the responsibility of A. M. R. Magee
- ⇒ RIPE NCC
  - ⇒ Eng Group
  - ⇒ Feedback & Comments

RPSL/148

# Network Training Workshop

---

## Routing Policy Specification Language

**RIPE NCC Database Group**  
**`<ripe-dbm@ripe.net>`**

# RPSL Tutorial

---

⇒ Goal

⇒ Awareness of RPSL

⇒ Examples

⇒ Target Audience

⇒ knowledge of Internet Routeing

⇒ no need to know Internet Routeing Registry

⇒ Tutorial

⇒ not a substitute for reading the drafts

# RPSL Drafts

---

⇒ draft-ietf-rps-rpsl-v2-03

⇒ Routing Policy Specification Language (RPSL)

⇒ draft-ietf-rps-appl-rpsl-04

⇒ Using RPSL in Practice



# Contents of the tutorial

---

## ⇒ Introduction

- ⇒ Internet Routing Registry
- ⇒ IETF RPS Working Group
- ⇒ RPSL; background

## ⇒ RPSL

- ⇒ Objects, Reserved Words/Names
- ⇒ Autonomous Systems and Routes
- ⇒ Sets and Operators
- ⇒ Specifying Policies

# Introduction

---

⇒ Internet Routing Registry

⇒ History

⇒ Structure

⇒ IETF RPS Working Group

⇒ Activities

⇒ Charter

⇒ RPSL background

⇒ Comparison with RIPE–181

⇒ Goals and Milestones

⇒ Current Status

# Internet Routing Registry

---

- ⇒ Established in 1995
- ⇒ Stability and consistency of routing
  - ⇒ network operators share information
- ⇒ Public databases:
  - ⇒ Cable & Wireless
  - ⇒ ANS
  - ⇒ Merit (RADB)
  - ⇒ Bell Canada (formerly CA\*net)
  - ⇒ RIPE

# Internet Routing Registry 2

---

- ⇒ These databases are independent
  - ⇒ but they exchange data
  - ⇒ can see all these databases at any one site
  - ⇒ only register your data in one database
- ⇒ Private: Cable & Wireless, ANS, Bell Canada
- ⇒ Public: RADB, RIPE
- ⇒ Routing policies expressed in RIPE-181
  - ⇒ RPSL is based on RIPE-181

# IETF RPS Working Group

---

⇒ Routing Policy System Working Group

⇒ Standardization:

⇒ protocols and recommended practices

⇒ Activities:

⇒ define a language

⇒ define a distributed registry model

⇒ provide a forum for discussion

⇒ <http://www.ietf.org/html.charters/rps-charter.html>

# RPSL

---

⇒ RIPE–181

⇒ some policies cannot be specified

⇒ Internet Routing Registry

⇒ needed a more powerful language

⇒ RPSL

⇒ more expressive than RIPE–181

⇒ policies can be specified at the AS level

⇒ policies can be detailed => router configurations

⇒ extensible (new routing protocols possible)

⇒ structured, vendor–neutral

# RPSL Features

---

## ⇒ Language features:

- ⇒ AS and route objects, object sets, policy as relations
- ⇒ structured, vendor-neutral, extensible
- ⇒ RPS IETF WG

## ⇒ History

PRDB ➡ RIPE 81 ➡ RIPE 181 ➡ RPSL

# Goals and Milestones of RPSL

---

⇒ RPSL Coding: 9/1997 – 5/1998

⇒ Testing and training: 5/1998 – now

⇒ Deployment: 1999

⇒ April 1999

⇒ submit RPSL to IESG as a Proposed Standard



# Current Status

---

⇒ RPSL and RIPE-181 servers in parallel

⇒ `rpsl.merit.edu`

⇒ `rpslii.ripe.net`

⇒ `<auto-rpsl@ripe.net>`

⇒ `whois -h rpslii.ripe.net`

⇒ RIPE Database

⇒ RPSL in new implementation

⇒ RPSL training

⇒ Wide scale testing

⇒ Updating in-house tools/extensions

# RPSL Training

---

⇒ <http://www.isi.edu/ra/rps/training/>

⇒ <http://www.ripe.net/db/rps/training/>

⇒ Tutorials in the RIPE region

⇒ Edinburgh (September 1998, RIPE-31)

⇒ Amsterdam (January 1999, RIPE-32)

⇒ Vienna (May 1999, RIPE-33)

⇒ Local Internet Registry Training, Quarter 3, 1999

# Tutorial

---

- ⇒ Objects, Reserved Names
- ⇒ Contact Information
- ⇒ Specifying Policy
- ⇒ Set Objects
- ⇒ Autonomous Systems

# Tutorial 2

---

⇒ Internet Routers

⇒ Advanced Topics

# Part I

---

## RPSL Objects

# Objects in RPSL

---

⇒ RPSL is based on objects

⇒ Objects and Attributes

⇒ Attributes and Values

⇒ Object Names

⇒ Reserved Names

# Object Based

---

⇒ Each object describes an entity in real world

⇒ Object classes

⇒ route

⇒ autonomous system

⇒ router

⇒ person, role, maintainer

⇒ set objects

⇒ dictionary

# Object

- 
- The diagram illustrates the structure of an RPSL object. It features a list of fields with their corresponding values. Four purple boxes with labels are connected to specific parts of the list by lines:
- Attribute name:** Points to the 'person:' field.
  - Attribute value:** Points to the 'Clare Lancers' value.
  - Comment:** Points to the '# day time' comment.
  - continuation:** Points to the 'remarks:' field.
- The list of fields and values is as follows:
- **person:** Clare Lancers
  - **address:** Corofin
  - **phone:** +1 23 123 # day time
  - **fax-no:** +1 23 121
  - **e-mail:** clancers@ripe.net
  - **nic-hdl:** CL123-TEST
  - **remarks:** This object is automatically converted from RIPE181
  - **mnt-by:** RIPE-NCC-MNT
  - **changed:** clancers@ripe.net 19990122
  - **source:** TEST



# Objects in RPSL

---

⇒ Similar to RIPE–181

⇒ Object

⇒ set of attributes

⇒ Attributes

⇒ mandatory or optional

⇒ values: single, list, multiple

# Objects in RPSL 2

---

⇒ Class "key"

⇒ set of attributes

⇒ uniquely identify each object

⇒ Attributes

⇒ Case insensitive

⇒ ASCII

⇒ type

# Objects in RPSL 3

---

⇒ Attribute types

⇒ <object-name>

⇒ <as-number>

⇒ <ipv4-address>

⇒ <address-prefix>

⇒ <address-prefix-range>

⇒ <date>

⇒ <nic-handle>

⇒ .....

⇒ Complete list in the draft

# Objects in RPSL 4

---

⇒ Address–prefix–range

⇒ address prefix followed by a range operator

⇒ Range operators

⇒  $\wedge+$ : inclusive more specifics

⇒ 5.0.0.0/8 $\wedge+$

⇒  $\wedge-$ : exclusive more specifics

⇒ 128.9.0.0/16 $\wedge-$

⇒  $\wedge n$ : length **n** more specifics

⇒ 30.0.0.0/ $\wedge 16$

⇒  $\wedge n-m$ : length **n–m** more specifics

⇒ 30.0.0.0/ $\wedge 24-32$

# Objects in RPSL 5

---

- ⇒ list of attribute–value pairs
  - ⇒ order is significant
- ⇒ the ‘class’ attribute should be first
- ⇒ line–continuation
  - ⇒ space, tab, ‘+’
- ⇒ comments
  - ⇒ may be anywhere
  - ⇒ begin with ‘#’

# Object Names

---

- ⇒ Object names can have – or \_ in the middle
  - ⇒ RGNET-MAINT-MCI
- ⇒ Can have digits
  - ⇒ RGNET-MAINT-MCI\_1
- ⇒ Case insensitive
  - ⇒ rgnet-MaInT-MCI

# Reserved Names

---

- ⇒ any as-any rs-any peers
- ⇒ and or not
- ⇒ atomic from to at action accept  
announce except refine
- ⇒ networks into inbound outbound

# Part II

---

## Contact Information



# Contact Information

---

⇒ Person

⇒ Role

⇒ Mntner

# Person Object

---

- **Person:** Clare Lancers
- **address:** Corofin
- **phone:** +1 23 123 # day time
- **fax-no:** +1 23 121
- **e-mail:** clancers@ripe.net
- **nic-hdl:** CL123-TEST

Person object  
information

- **remarks:** This object is automatically
- converted from RIPE181
- **mnt-by:** RIPE-NCC-MNT
- **changed:** clancers@ripe.net 19990122
- **source:** TEST

Auxiliary  
information

# Auxiliary Information

---

<b>descr:</b>	short free text description of the object
<b>remarks:</b>	free text comment attribute
<b>tech-c:</b>	Technical contact nic handles
<b>admin-c:</b>	Administrative contact nic handles
<b>notify:</b>	emails to send notification of changes
<b>mnt-by:</b>	maintainer authorized to do changes
<b>changed:</b>	<email> <date>
<b>Source:</b>	registry

# Role object

---

- **role:** RIPE NCC Operations
- **address:** Singel 258
- 1016 AB Amsterdam
- The Netherlands
- **phone:** +31 20 535 4444
- **fax-no:** +31 20 545 4445
- **e-mail:** ops@ripe.net
- **admin-c:** OK65
- **tech-c:** RW488-RIPE
- **tech-c:** CF124
- **nic-hdl:** OPS4-RIPE
- **notify:** ops@ripe.net
- **changed:** roderik@ripe.net 19981208
- **source:** RIPE
-

# Mntner Objects

---

- Example:
- 
- **mntner:** RIPE-DBM-MNT
- **descr:** Mntner for RIPE DBM objects
- **admin-c:** AMRM1-RIPE
- **tech-c:** RD132-RIPE
- **upd-to:** ripe-dbm@ripe.net
- **mnt-nfy:** ripe-dbm@ripe.net
- **auth:** CRYPT-PW sprZvgymRKygk
- **notify:** ripe-dbm@ripe.net
- **mnt-by:** RIPE-DBM-MNT
- **changed:** ripe-dbm@ripe.net 19980211
- **source:** RIPE

Defines access control for other objects in database!

# Template of a mntner object

---

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>mntner:</b>	<i>&lt;object-name&gt;</i>	mandatory, single, class key
<b>descr:</b>	<i>&lt;free-form&gt;</i>	mandatory, single
<b>auth:</b>	Described later	mandatory, multiple
<b>upd-to:</b>	<i>&lt;email-address&gt;</i>	mandatory, multiple
<b>mnt-nfy:</b>	<i>&lt;email-address&gt;</i>	optional, multiple
<b>tech-c:</b>	<i>&lt;nic-handle&gt;</i>	mandatory, multiple
<b>admin-c:</b>	<i>&lt;nic-handle&gt;</i>	optional, multiple
<b>remarks:</b>	<i>&lt;free-form&gt;</i>	optional, multiple
<b>notify:</b>	<i>&lt;email-address&gt;</i>	optional, multiple
<b>mnt-by:</b>	list of <i>&lt;mntner-name&gt;</i>	mandatory, multiple
<b>changed:</b>	<i>&lt;email-address&gt;</i> <i>&lt;date&gt;</i>	mandatory, multiple
<b>source:</b>	<i>&lt;registry-name&gt;</i>	mandatory, single

# Auth Attribute

---

- ⇒auth: PGPKEY-<PGP Key ID>
- ⇒auth: CRYPT-PW lz1A7/JnfkTI
- ⇒auth: MAIL-FROM ripe-dbm@ripe.net
- ⇒auth: MAIL-FROM .\*@ripe.net
- ⇒auth: NONE
  
- ⇒RIPE NCC implementation
  - ⇒RIPE-157
  - ⇒RIPE-189
  - ⇒draft-ietf-rps-dbsec-pgp-authent-01.txt

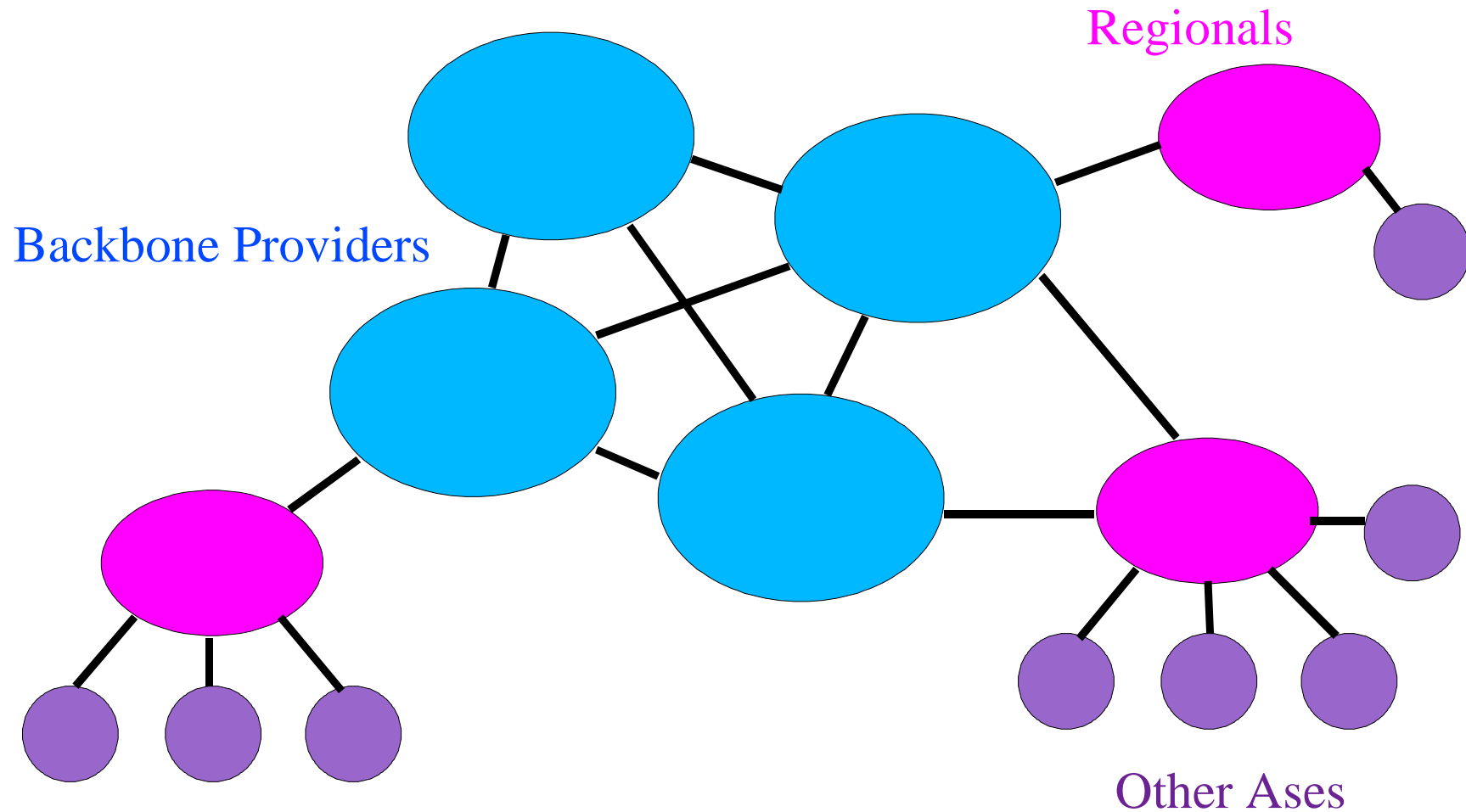
# Part III

---

## Specifying Policy



# Inter-AS Topology



# AS Relationships

---

⇒ Customer–Regional Provider

⇒ Provider forwards traffic

⇒ advertises routes

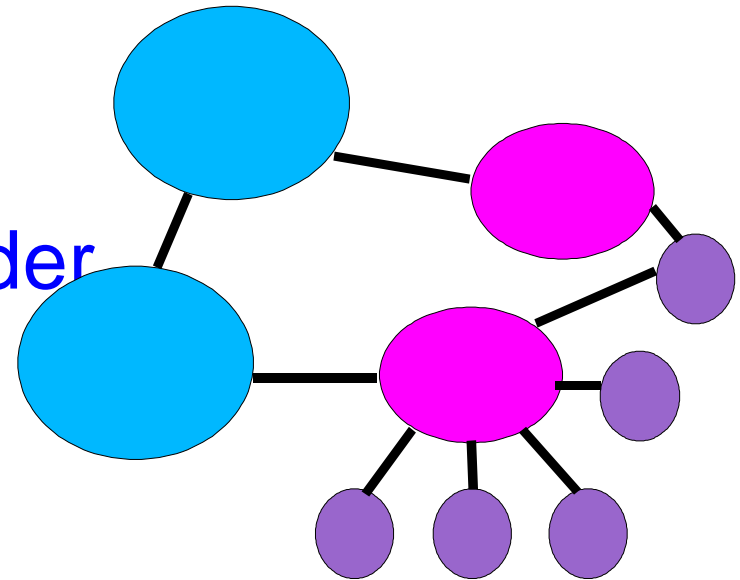
⇒ Peer–Peer

⇒ Mutual benefit

⇒ Regional Provider–Backbone Provider

⇒ Similar to Customer–Regional Provider

⇒ Typical routing policies implement these



# Policies of an AS

---

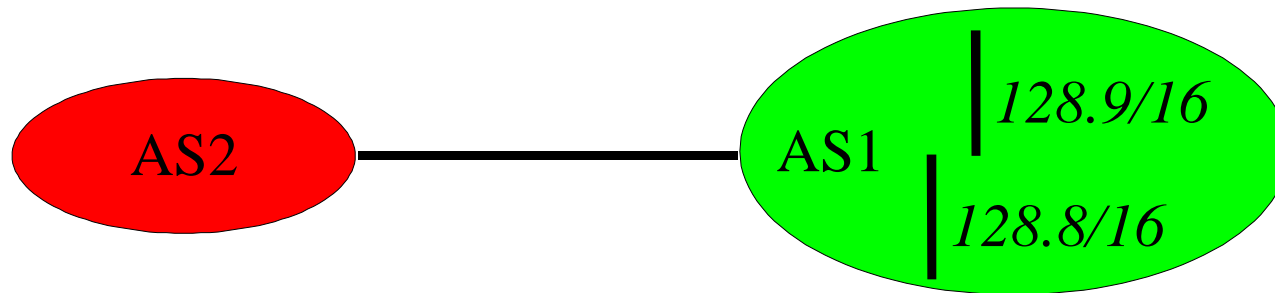
⇒ Use aut-num object

⇒ AS1 originates two routes

⇒ 128.8.0.0/16

⇒ 128.9.0.0/16

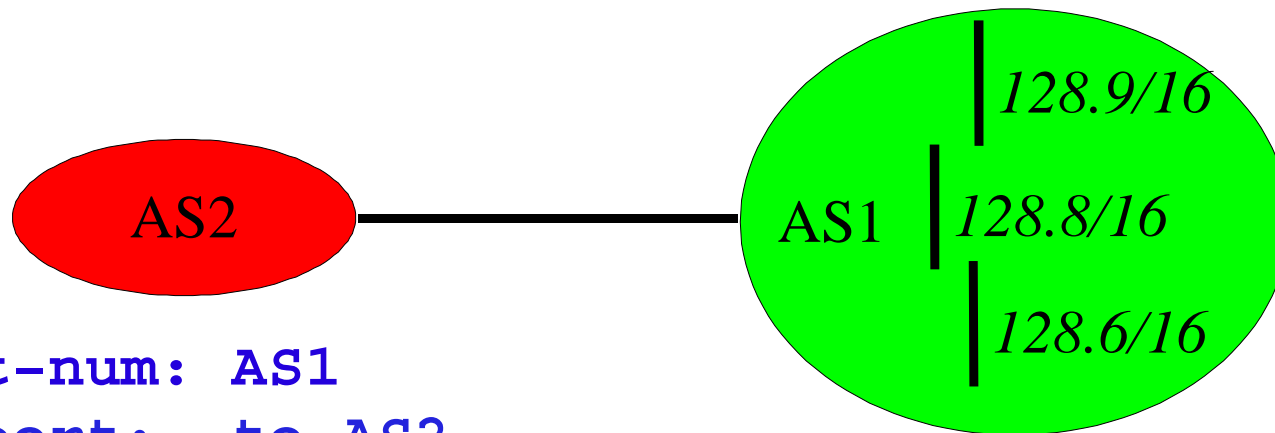
# Prefix Based



- `aut-num: AS1`
- `export: to AS2`
- `announce {128.9.0.0/16, 128.8.0.0/16}`
- 
- `aut-num: AS2`
- `import: from AS1`
- `accept {128.9.0.0/16, 128.8.0.0/16}`
- 
- **N.B.** Filtering based on Address-Prefix Set

# Originate more routes?

---



- `aut-num: AS1`
- `export: to AS2`
- `announce {128.9.0.0/16, 128.8.0.0/16,`
- `128.6.0.0/16}`
- `aut-num: AS2`
- `import: from AS1`
- `accept {128.9.0.0/16, 128.8.0.0/16,`
- `128.6.0.0/16}`

# Route Template

⇒ <u>Attribute</u>	<u>Value</u>	<u>Type</u>
⇒		
⇒route: key	<address-prefix>	mandatory, single, class
⇒origin: key	<as-number>	mandatory, single, class
⇒member-of:	list of	
⇒	<route-set-names>	optional, multiple
⇒inject:		optional, multiple
⇒components:		optional, single
⇒aggr-bndry:		optional, single
⇒aggr-mtd:		optional, single
⇒export-comps:		optional, single
⇒holes:		optional, multiple
⇒		

# route-set Objects

---

- route-set: rs-foo
- members: 128.9.0.0/16, 128.9.0.0/24,
- 128.8.0.0/16
- descr: some address prefixes
- mnt-by: MAINT-RGNET
- tech-c: RB366
- changed: randy@psg.com 19960829
- source: RADB
- 
- 
- route-set: rs-bar
- members: 128.7.0.0/16, rs-foo
- 
-

# Route-Set Template

⇒ <u>Attribute</u>	<u>Value</u>	<u>Type</u>
⇒		
⇒ <b>route-set</b>	<i>&lt;object-name&gt;</i>	Mandatory, single,
⇒		class key
⇒ <b>members</b>	list of	optional,
multi-values		
⇒	<i>&lt;address-prefix-range&gt;</i>	
⇒	<b>&lt;route-set-name&gt;</b> or	
⇒	<b>&lt;route-set-name&gt;</b> <i>&lt;range-operator&gt;</i> or	
⇒	<b>rs-any</b>	
⇒		
⇒ <b>mbrs-by-ref</b>	list of <b>&lt;mntner-names&gt;</b>	optional,
multi-valued		
⇒	or <b>ANY</b>	
⇒		



# Indirect Members of Route-Set

---

- route-set: RS-ANS-IGP\_ONLY
- descr: ANS IGP aggregates
- mbrs-by-ref: any
- 
- 
- route: 207.25.17.0/24
- origin: AS1675
- member-of: RS-ANS-IGP\_ONLY
- mnt-by: MNT-ANS
- 
- route: 192.157.69.0/24
- origin: AS1675
- member-of: RS-ANS-IGP\_ONLY
- mnt-by: MNT-ANS
-

# Restricted Indirect Members

---

- route-set: RS-ANS-IGP\_ONLY
- descr: ANS IGP aggregates
- mbrs-by-ref: MNT-ANS, MNT-CENGIZ
- 
- 
- route: 207.25.17.0/24
- origin: AS1675
- member-of: RS-ANS-IGP\_ONLY
- mnt-by: MNT-ANS
- 
- route: 192.157.69.0/24
- origin: AS1675
- member-of: RS-ANS-IGP\_ONLY
- mnt-by: MNT-CURTIS
-

# Direct & Indirect Members

---

- **route-set:** RS-ANS-IGP\_ONLY
- **descr:** ANS IGP aggregates
- **members:** 207.25.17.0/24, 207.25.16.0/24,  
207.25.20.0/24
- **mbrs-by-ref:** MNT-ANS
- 
- **route:** 207.25.17.0/24
- **origin:** AS1675
- **member-of:** RS-ANS-IGP\_ONLY
- **mnt-by:** MNT-ANS
- 
- **route:** 192.157.69.0/24
- **origin:** AS1675
- **member-of:** RS-ANS-IGP\_ONLY
- **mnt-by:** MNT-ANS
-

# Direct Members

---

The "**member-of**" attribute of the **route** object is an additional mechanism for specifying the members indirectly.

If an address prefix is listed in the **members** attribute of a route-set, it is a member of that route set.

The **route** object corresponding to this address prefix does not need to contain a **member-of** attribute referring to this set name.

# Per route-set

---

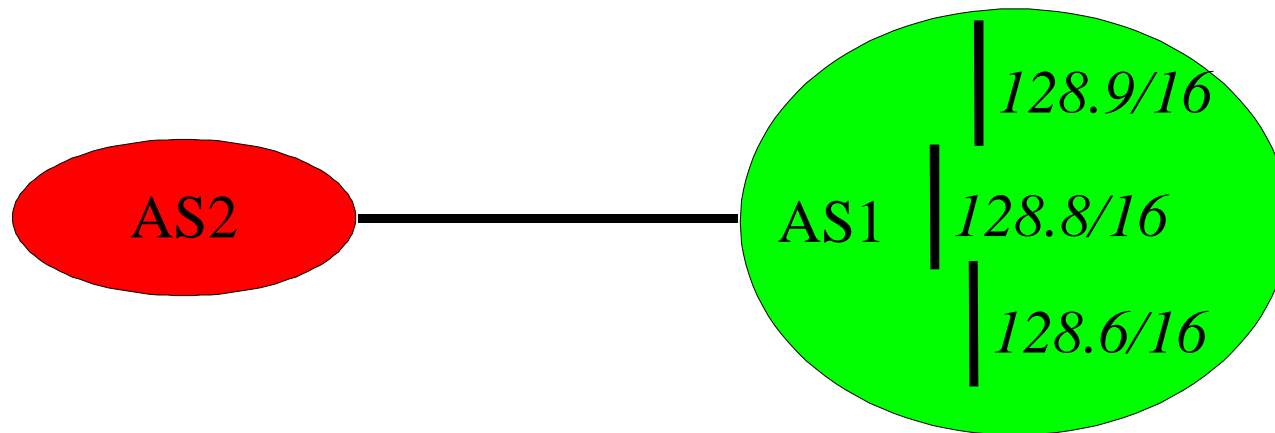
```
route-set:    rs-red
members:      128.6.0.0/16, 128.9.0.0/16,
               128.8.0.0/16
```

```
aut-num: AS1
export:  to AS2      announce rs-green
```

```
aut-num: AS2
import:  from AS1    accept    rs-green
```

# Policy and route-set

---



☰ aut-num: AS1

⇒ export: to AS2

⇒ announce rs-green

⇒

⇒ aut-num: AS2

⇒ import: from AS1

→

# Range Operators & route-sets

- `route-set:` `rs-martians`
- `descr:` most ASes do not import these routes
- `members:` `127.0.0.0/8^+,`
- `10.0.0.0/8^+,` `172.16.0.0/20^+,`
- `192.168.0.0/16^+,` `192.0.2.0/24^+,`
- `128.0.0.0/16^+,` `191.255.0.0/16^+,`
- `192.0.0.0/24^+,` `223.255.255.0/24^+,`
- `224.0.0.0/3^+`

Inclusive more specifics:  $\wedge_+$

Exclusive more specifics:  $\wedge_-$

Length n more specifics:  $\wedge_n$

Length n-m more specifics:  $\wedge_{n-m}$

# Route Object 1

---

⇒ Subset of a route !

⇒ The **route** and **origin** attributes == class key



# Route Object 2

---

Route: 193.0.0.0/23  
origin: AS3333

route: 128.8.0.0/16  
origin: AS1

route: 128.8.0.0/16  
origin: AS2

# Route Object 3

---

route: 156.36.0.0/16

origin: AS2914

member-of: RS-VERIO

descr: my routes

mnt-by: MAINT-RGNET

tech-c: RB366

changed: randy@psg.com 19960829

source: RADB

Policy  
information

- Route 156.36.0.0/16
  - is originated by AS2914
  - is a member of set RS-VERIO

# AS Numbers in Policy

---

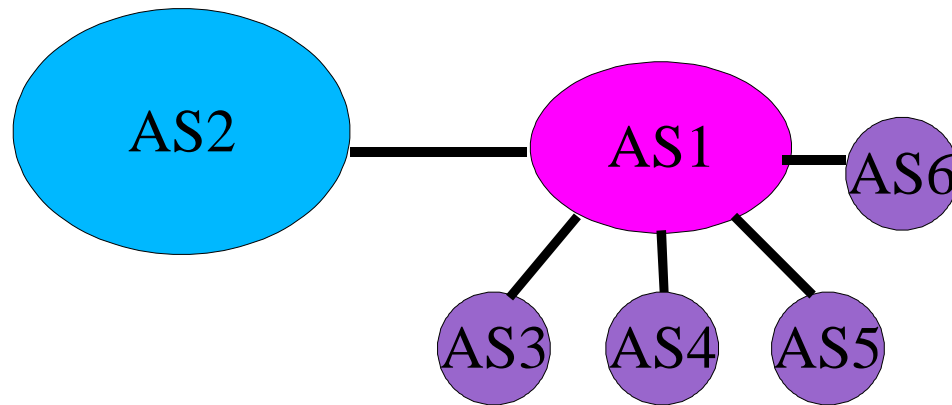


- route: 128.9.0.0/16      route: 128.8.0.0/16
- origin: AS1              origin: AS1
- 
- aut-num: AS1
- export: to AS2      announce AS1
- 
- aut-num: AS2
- import: from AS1 accept AS1

AS1 == {128.9.0.0/16, 128.8.0.0/16}

# Cumbersome?

---



- **aut-num:** AS1
- **export:** to AS2      announce AS1 OR AS3 OR ...  
AS6
- 
- **aut-num:** AS2
- **import:** from AS1      accept AS1 OR AS3 OR ...  
AS6

# as-set Objects

---

- **as-set:** AS-SESQUISTUB
- **descr:** Single Homed Sesquinet Customer ASs
- **members:** AS1832, AS2712, AS302, AS3526, AS8
- **tech-c:** SB98
- **mnt-by:** MAINT-AS114
- **source:** RADB
- 

Same flexibility as route-set objects

# AS-Set Template

---

⇒ <u>Attribute</u>	<u>Value</u>	<u>Type</u>
⇒		
⇒ <b>as-set</b>	<i>&lt;object-name&gt;</i>	mandatory, single,
⇒	class key	
⇒ <b>members</b>	list of	optional, multiple
⇒	<i>&lt;as-numbers&gt;</i> or	
⇒	<b>&lt;as-set-names&gt;</b> or	
⇒	<b>as-any</b>	
⇒ <b>mbrs-by-ref</b>	list of	optional, multiple
⇒	<b>&lt;mntner-names&gt;</b> or	
⇒	<b>ANY</b>	

# Indirect as-sets

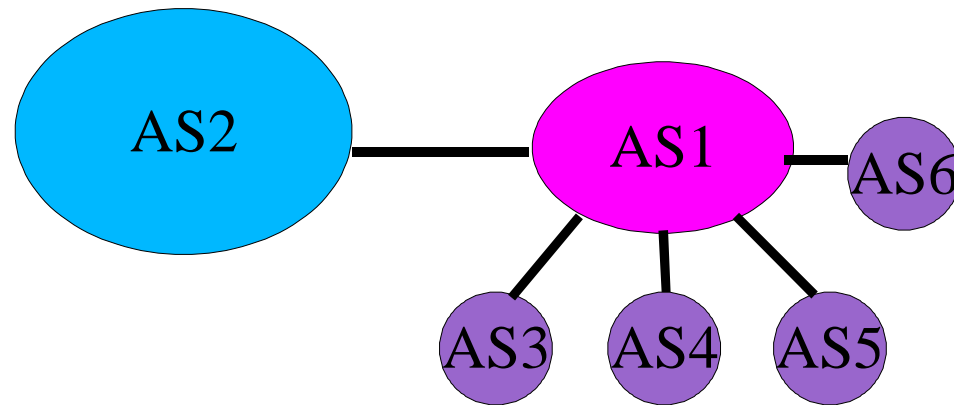
---

as-set: as-aads-mlpa  
descr: MLPA participants at the AADS NAP  
mbrs-by-ref: any  
admin-c: Andrew Schmidt  
tech-c: Mark Cnota  
notify: mlpa-participants@aads.net  
mnt-by: MAINT-RSPEER  
changed: auto-mlpa@aads.net 19971123  
source: RADB

aut-num: AS4550  
member-of: as-aads-mlpa

aut-num: AS683  
member-of: as-aads-mlpa

# Using as-set objects

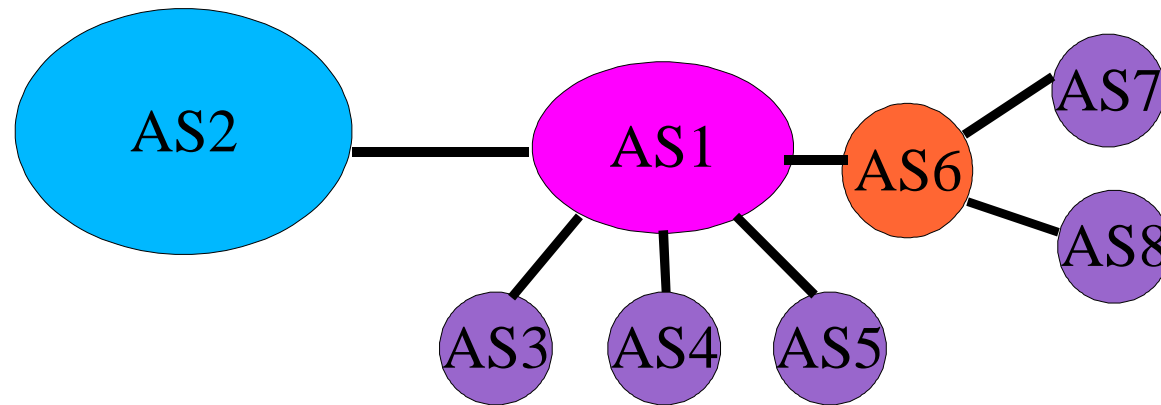


- ⇒ As-set: AS1:AS-Customers
- ⇒ members: AS1, AS3, AS4, AS5, AS6
- ⇒
- ⇒ aut-num: AS1
- ⇒ export: to AS2 announce
- AS1:AS-Customers
- ⇒



# Using as-set objects 2

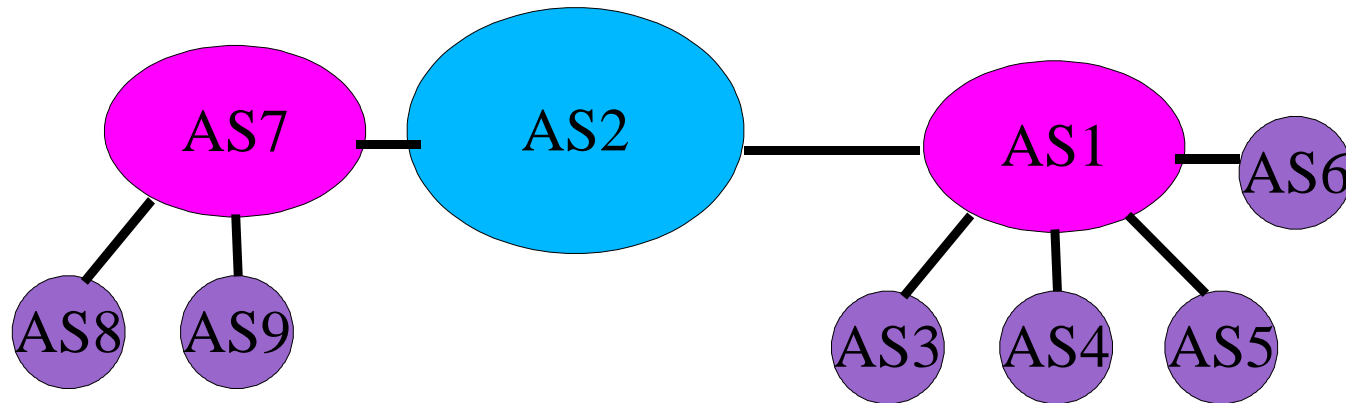
---



- **as-set:** **AS1:AS-Customers**
- **members:** AS1, AS3, AS4, AS5, AS6:AS-Customers
- 
- **aut-num:** AS1
- **export:** to AS2      announce **AS1:AS-Customers**
- 
- **aut-num:** AS2
- **import:** from AS1      accept **AS1:AS-Customers**

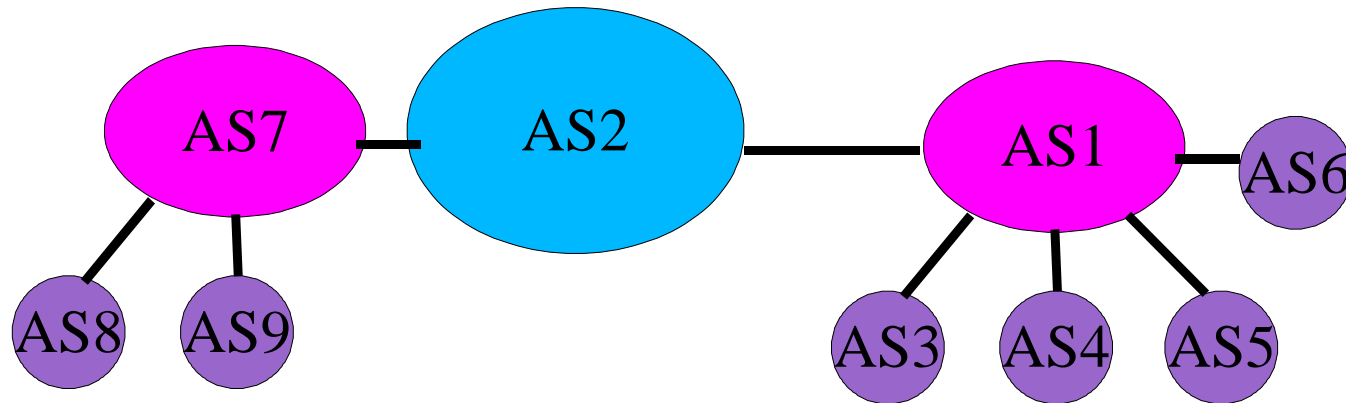
# Cumbersome?

---



- **as-set:** AS1:AS-Customers
- **members:** AS1, AS3, AS4, AS5, AS6
- 
- **as-set:** AS7:AS-Customers
- **members:** AS7, AS8, AS9
- 
- **aut-num:** AS2
- **import:** from AS1 **accept AS1:AS-Customers**
- **import:** from AS7 **accept AS7:AS-Customers**
-

# PeerAS



```
as-set: AS2:AS-Customers  
members: AS1, AS7
```

```
aut-num: AS2  
import: from AS2:AS-Customers  
accept PeerAS:AS-Customers
```

Same as:

```
from AS1 accept AS1:AS-Customers  
from AS7 accept AS7:AS-Customers
```

# PeerAS

---

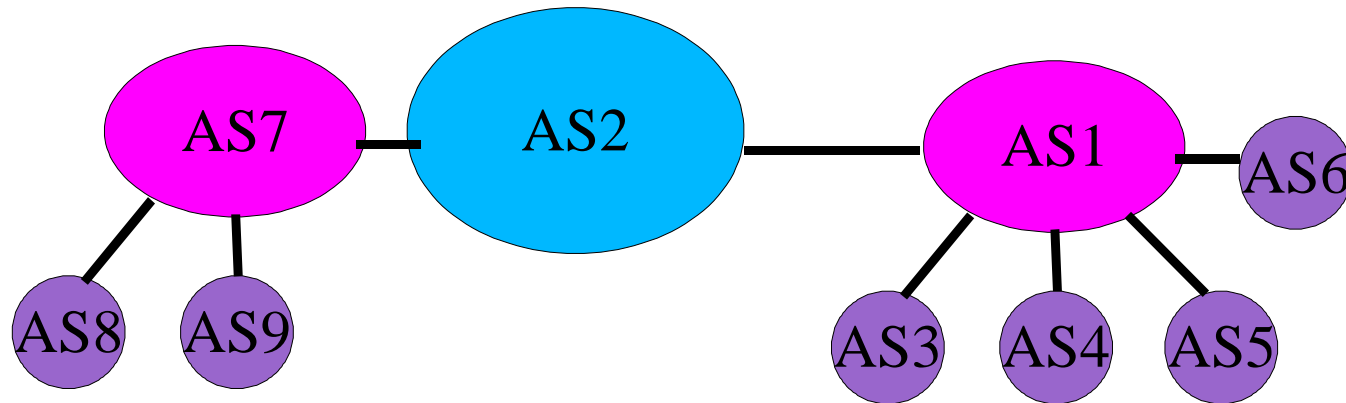
⇒ Keyword: **PeerAS**

⇒ Used in **import** attribute

⇒ instead of the AS number of the peer AS

⇒ Useful when using AS expression

# PeerAS



as-set: AS2:AS-Customers

members: AS1, AS7

aut-num: AS2

import: from AS2:AS-Customers

accept *PeerAS*:AS-Customers

Same as:

from AS1 accept AS1:AS-Customers

from AS7 accept AS7:AS-Customers

# Notations

---

AS Numbers            **AS**2914

Address prefixes      156.36.0.0/16

Route set names       **RS**–VERIO

AS set names           **AS**–REEL

# Predefined Set Objects

---

⇒ RS-ANY, rs-any

⇒ AS-ANY, as-any

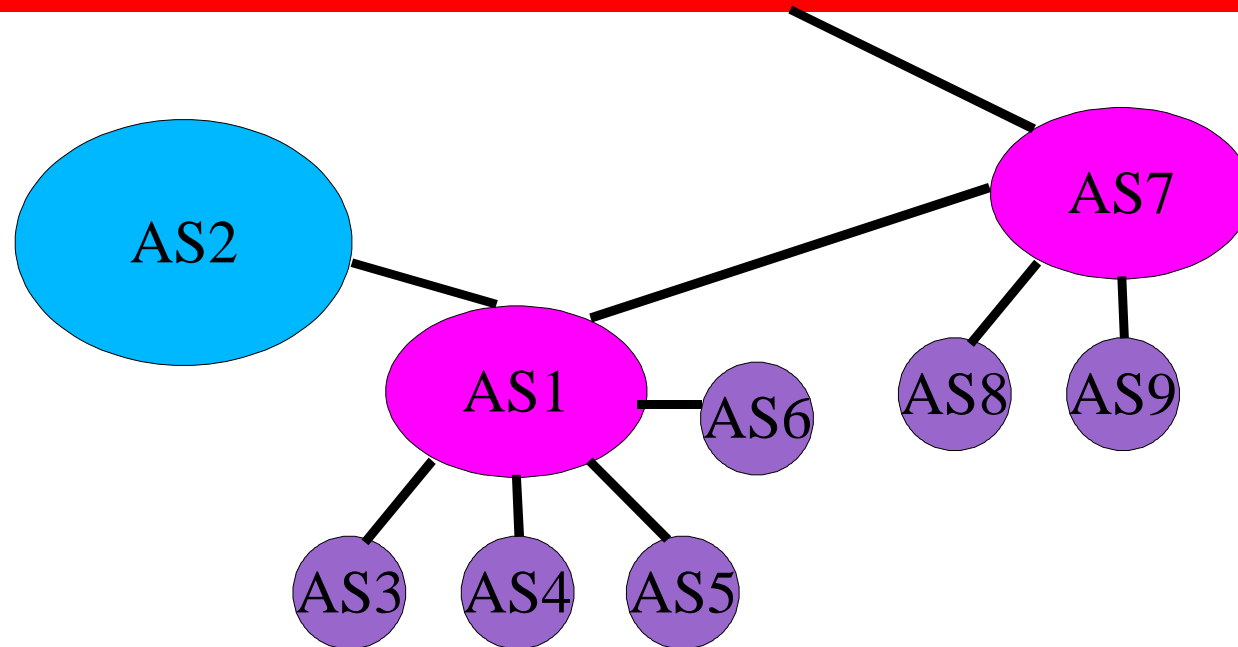
⇒ Route-set context

⇒ AS number: **AS<sub>x</sub>** == routes originated by **AS<sub>x</sub>**

⇒ as-set: **AS-X** == routes originated by the AS's  
in **AS-X**

# Complex Example

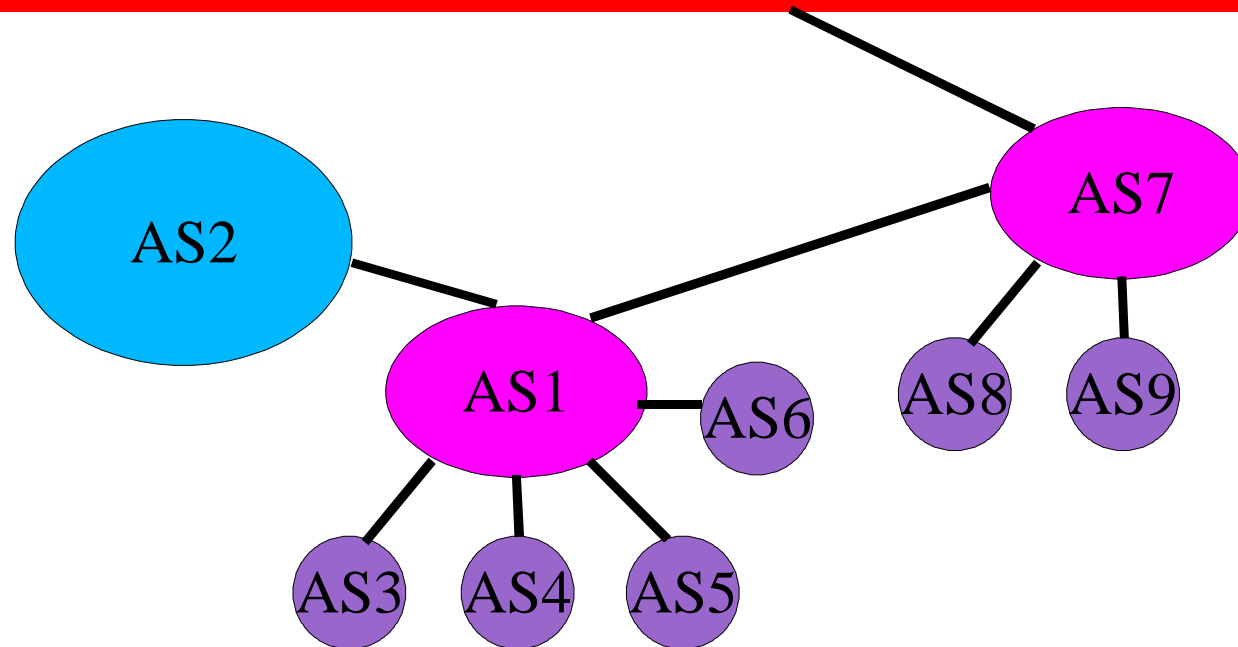
---



Solution ?



# AS Path Based



- `import: from AS1 accept <^AS1 .* AS8$>`
- 
- `import: from AS1 accept <^AS1 AS1:AS-Customers*$>`

**No route prefixes here!**

# AS Path Regular Expressions

---

AS1

AS1

as-foo

any AS in as-foo

x\*

0 or more occurrence of x

x+

1 or more occurrence of x

x?

0 or 1 occurrence of x

^

beginning of path

\$

end of path

x | y

x or y

x y

x followed by y

# AS Path Regular Expressions

---

⇒ Policy filter

⇒ only when the expression is between '<' and '>'

⇒ Regular expressions

⇒ the alphabet of AS numbers

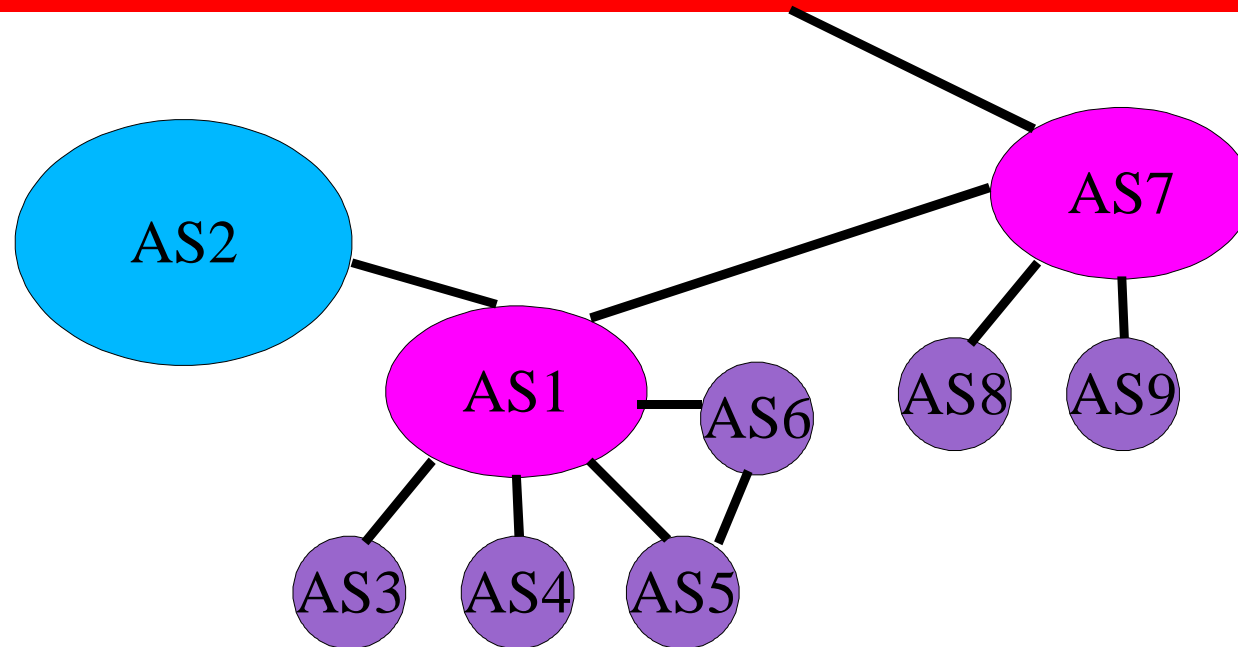
⇒ Router can check

⇒ BGP: AS\_PATH

⇒ IDRP: RD\_PATH

⇒ Regular Expression Operators

# AS Path RE Example



⇒ `<^AS1+ AS1:AS-Customers* $>` matches

⇒ AS1

⇒ AS1 AS3

⇒ AS1 AS4

⇒ AS1 AS5 AS6

⇒ AS1 AS1 AS5 AS5 AS6

# Composite Policy Filters

---

⇒ NOT

⇒ negation

⇒ AND

⇒ intersection

⇒ OR

⇒ union

# Composite Policy Filters 2

---



⇒ AS1 == {128.8.0.0/16, 128.9.0.0/16}

⇒ rs-red == {128.6.0.0/16, 128.9.0.0/16}

⇒

⇒ AS1 OR rs-red == {128.6.0.0/16, 128.8.0.0/16, 128.9.0.0/16}

⇒

⇒ AS1 AND rs-red == {128.9.0.0/16}

⇒

⇒ AS1 AND NOT rs-red == {128.8.0.0/16}

⇒

# AS numbers = routes

---

- `aut-num: AS1`
- `import: from AS1`
- `accept (AS1 OR rs-red) AND NOT`  
`{0.0.0.0/0}`
- 
- 
- 
- `AS1 == {128.9.0.0/16, 128.9.0.0/16}`
- `rs-red == {128.6.0.0/16, 128.9.0.0/16}`
- 
- `AS1 OR rs-red == {128.6.0.0/16, 128.9.0.0/16, 128.9.0.0/16}`
- 
- `AS1 AND rs-red == {128.9.0.0/16}`
- 
- `AS1 AND NOT rs-red == {128.8.0.0/16}`
- 
- **N.B.** `AS numbers & as-set names == routes`

# Prefix Length Based Policy

---

```
aut-num: AS1
import: from as-any
       accept ANY AND NOT { 192.168.0.0/16^+ }
```

**N.B.** Filter == Address-Prefix Set;  
Composite Policy

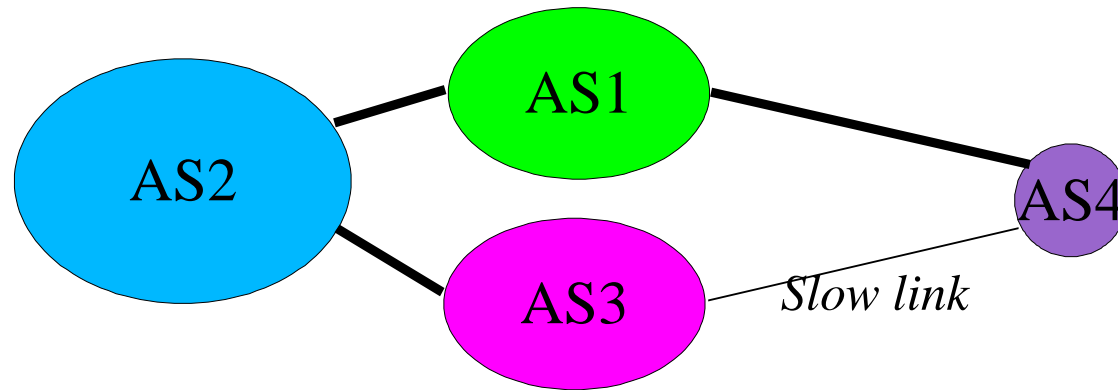


# Example of Actions

---

- ⇒ Two links:
  - ⇒ one is fast
  - ⇒ the other is slow

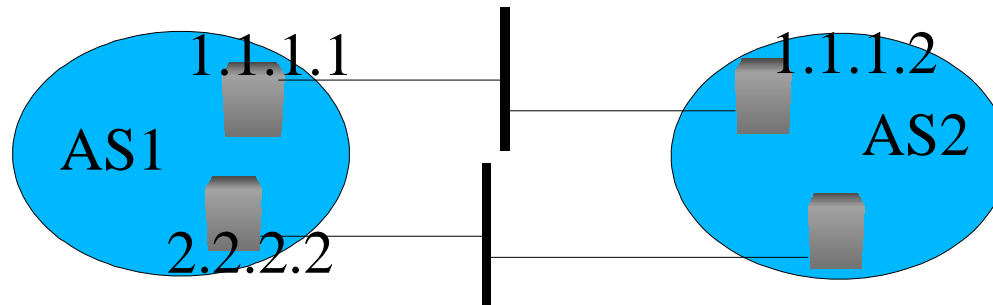
# Choice of links



- `aut-num:` AS4
- `import:` from AS1 action `pref = 10;` accept ANY
- `import:` from AS3 action `pref = 15;` accept ANY

**Smaller the number, higher the preference!**

# Peering Choice



- `aut-num: AS1`
- `import: from AS2 at 2.2.2.2`
- `action pref = 10;`
- `accept AS2`
- `import: from AS2 1.1.1.2 at 1.1.1.1`
- `action pref = 5;`
- `accept AS2`

# Specifying Actions

---

⇒ RPSL policy actions

⇒ set or modify route attributes (BGP attributes)

⇒ instruct routers to do special operations

⇒ route flap dampening

⇒ Route attributes ?

⇒ RPSL dictionary

# Specifying Actions 2

---

⇒ Syntax of a policy action

⇒ `x.method(arguments)`

⇒ `x "op" argument`

⇒ Terminated by semicolon ‘;’

⇒ Composite policy actions possible

⇒ evaluated left-to-right

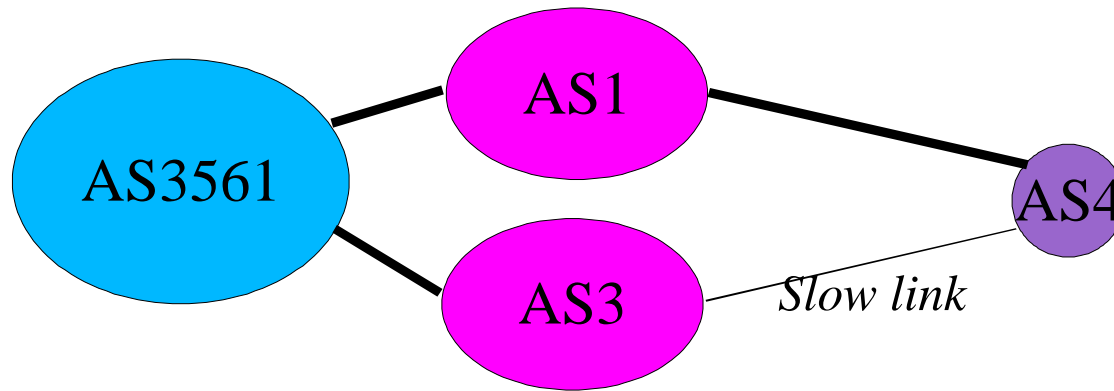
# Actions

---

⇒ import: from ... action XXX; accept ...  
⇒ export: to ... action XXX; announce ...  
⇒  
⇒ med = 0;  
⇒ med = igp\_cost;  
⇒  
⇒ community.append(NO\_EXPORT, 10250, 3561:90);  
⇒ community.delete(NO\_EXPORT);  
⇒  
⇒ aspath.prepend(AS1, AS1, AS1);  
⇒

# Example; community

---



⇒ AS4 wants AS3561 to prefer AS1 path

⇒ AS3561 prefers routes with

⇒ no community

⇒ with community 3561:90

⇒ with community 3561:80

⇒ with community 3561:70

# AS3561's Policies

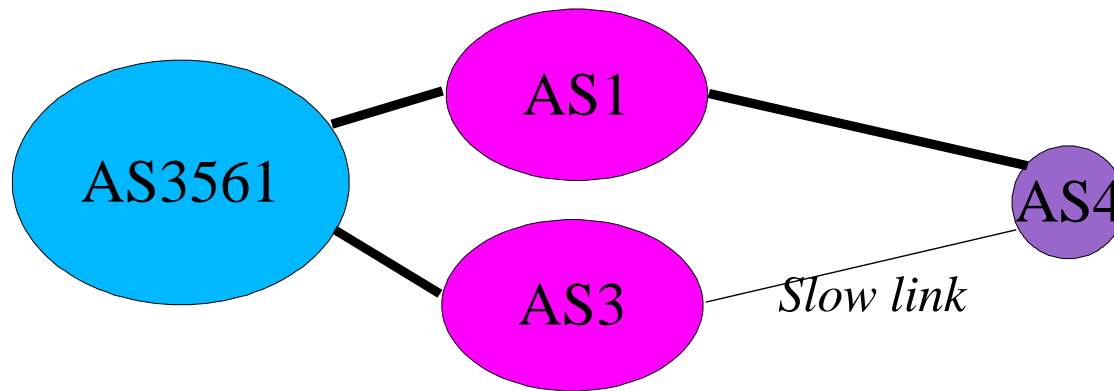
---

```
aut-num: AS3561
import: from AS-ANY
      action pref = 30;
      accept community(3561:70)
import: from AS-ANY
      action pref = 20;
      accept community(3561:80)
import: from AS-ANY
      action pref = 10;
      accept community(3561:90)
import: from AS-ANY
      action pref = 0;
      accept ANY
```



# AS4's Policies

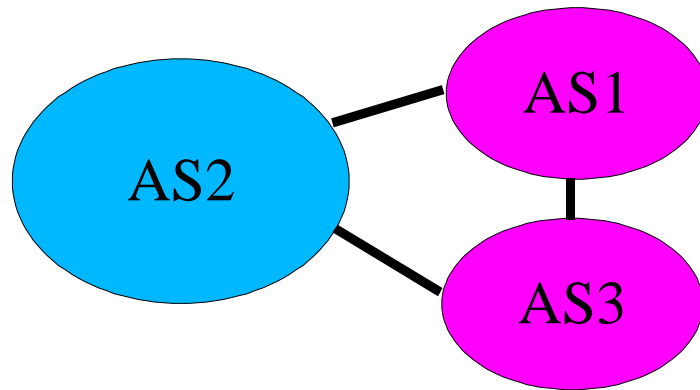
---



- `aut-num: AS4`
- `export: to AS1 action community.={3561:90};`
- `to AS3 action community.={3561:80};`
- `announce AS4`
-

# Policy Example

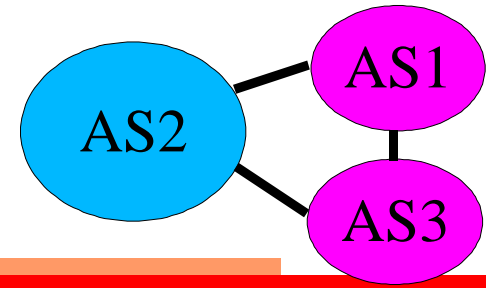
---



## ⇒ Possibilities

- 1 each link is private (i.e. no transit)
- 2 case 1 + AS2 as backup transit provider
- 3 case 2 + AS2 can use the link as backup to AS1
- 4 case 3 + AS2 can use the link as backup to AS3

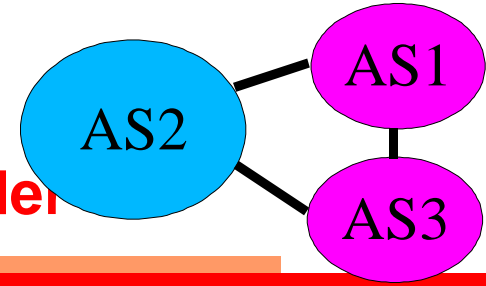
# Case 1



- **aut-num:** AS1
- **import:** from AS3 accept AS3
- **import:** from AS2 accept AS2
- **export:** to AS3 announce AS1
- **export:** to AS2 announce AS1
- 
- **aut-num:** AS3
- **import:** from AS1 accept AS1
- **import:** from AS2 accept AS2
- **export:** to AS1 announce AS3
- **export:** to AS2 announce AS3
- 
- **aut-num:** AS2
- **import:** from AS1 accept AS1
- **import:** from AS3 accept AS3
- **export:** to AS1 announce AS2
- **export:** to AS3 announce AS2
-

# Case 2

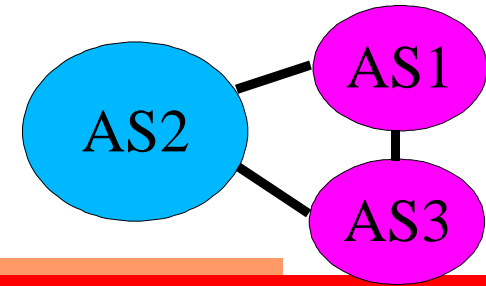
AS2 as backup transit provider



- aut-num: AS2
- import: from AS1 accept AS1
- import: from AS3 accept AS3
- export: to AS1 announce ANY
- export: to AS3 announce ANY
- 
- aut-num: AS1
- import: from AS2 action pref = 10; accept ANY
- import: from AS3 action pref = 5; accept AS3
- export: to AS2 announce AS1
- export: to AS3 announce AS1
- 
- aut-num: AS3
- import: from AS2 action pref = 10; accept ANY
- import: from AS1 action pref = 5; accept AS1
- export: to AS2 announce AS3
- export: to AS1 announce AS3
-

# Case 3

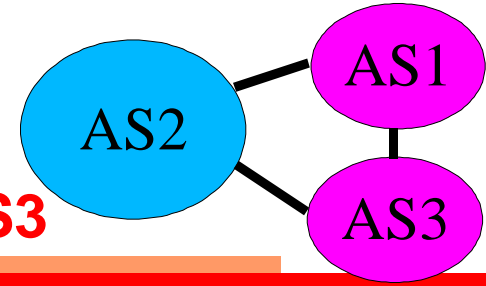
AS3 has backup to AS1



- aut-num: AS2
- import: from AS1 action pref = 10; accept AS1
- import: from AS3 action pref = 20; accept AS1
- import: from AS3 accept AS3
- 
- aut-num: AS3
- export: to AS2 announce AS3 or AS1
- 
-

# Case 4

AS2 is backup for AS1 to AS3



- aut-num: AS2
- import: from AS1 action pref = 10; accept AS1
- import: from AS3 action pref = 20; accept AS1
- import: from AS3 action pref = 10; accept AS3
- import: from AS1 action pref = 20; accept AS3
- 
- aut-num: AS3
- export: to AS2 announce AS3 or AS1
- 
- aut-num: AS1
- export: to AS2 announce AS1 or AS3
- 
-

# Ambiguity Resolution

---

- ⇒ Two or more peering expressions
  - ⇒ describe the same peering
- ⇒ Which is used ?
- ⇒ Specification–order rule
  - ⇒ the first peering specification is always used

# Ambiguity Resolution

---

Example:

**aut-num: AS1**

**import: from AS2 action pref = 2; accept AS4**

**import: from AS2 action pref = 1; accept AS4 OR AS5**

**AS4's routes are accepted from AS2 with preference 2**

**AS5's routes are accepted from AS2 with preference 1**



# RPSL

---

⇒ RPSL

⇒ compact representation of policy

⇒ accurate representation of policy

⇒ Training materials

⇒ <http://www.isi.edu/ra/rps/training>

⇒ <http://www.ripe.net/db/rps/training>

# RIPE NCC RPSL server

---

⇒ [rpslii.ripe.net](http://rpslii.ripe.net)

⇒ Statistics:

⇒ Updates

⇒ Queries

⇒ Demonstration

# Part IV

---

⇒

⇒

⇒

⇒ **Set Objects**

# Set Objects

---

⇒ Sets of routes, autonomous systems, etc.

⇒ **route-set**

⇒ **as-set**

⇒ **filter-set**

⇒ **peering-set**

⇒ **rtr-set**

⇒ Specify members

⇒ directly

⇒ indirectly

# Set Objects 2

---

⇒ Set names

⇒ as-customers

⇒ rs-partner

⇒ .....

⇒ Hierarchical set names

⇒ authorisation

⇒ AS1:AS-CUSTOMERS

⇒ AS1:RS-EXPORT:AS2

# Set Objects 3

---

⇒Route-set

⇒as-set

⇒filter-set

⇒peering-set

⇒rtr-set

# Filter-Set Objects

---

Examples:

filter-set: fltr-mine

filter: {5.0.0.0/8, 6.0.0.0/8}

filter-set: fltr-yours

filter: (AS1 or fltr-mine) and <AS2>

# Filter-Set Objects 2

---

Set of routes that are matched by a filter

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>filter-set</b>	<i>&lt;object-name&gt;</i>	mandatory, single, class key
<b>filter</b>	<i>&lt;filter&gt;</i>	mandatory, single

The **filter** attribute == policy filter.

Policy filter => matches a subset of routes.

Filter set names: "**fltr-**"



# Filter–Set Objects 3

---

⇒ Policy filters

⇒ ANY

⇒ Address–Prefix Set

⇒ Route Set Name

⇒ AS Path Regular Expressions

⇒ Composite Policy Filters

⇒ Routing Policy Attributes

⇒ Filter Set Name

# Address Prefix Set

---

- ⇒ List of address–prefixes between '{', '}'
- ⇒ Policy filter matches those routes in the set
- ⇒ range operators are optional

## ⇒ Example:

⇒ filter–set: fltr–mine

⇒ filter: {128.8.0.0/16, 128.9.0.0/16}

⇒ filter–set: fltr–default

⇒ filter: {0.0.0.0/^19–32}

# Route Set Name

---

⇒ Route set name

⇒ matches the routes that are members of the set

⇒ Route set name

⇒ name of a **route-set** object

⇒ AS number

⇒ name of an **as-set** object

⇒ Can be followed by a range operator

⇒ {5.0.0.0/8, 6.0.0.0/8}^+

⇒ AS1^\_

# Routing Policy Attributes

---

- ⇒ Can use the values of other attributes
- ⇒ See the RPSL dictionary
- ⇒ Example: BGP community attribute
  - ⇒ filter-set: fltr-mci-communities
  - ⇒ filter: community(3561:100, 3561:90, 3561:80)
- ⇒ Evaluated before **AND, OR, NOT**

# Filter Set Name

---

- ⇒ Matches a set of routes
  - ⇒ matched by **filter** of filter set
- ⇒
- ⇒ Inclusion possible
  - ⇒ can use name of a filter set in a filter

# Rtr-Set Objects

---

Router-Set Name: "rtrs-"

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>rtr-set</b>	<i>&lt;object-name&gt;</i>	mandatory, single, class key
<b>members</b>	list of <i>&lt;inet-rtr-names&gt;</i> or <i>&lt;rtr-set-names&gt;</i> or <i>&lt;ipv4_addresses&gt;</i>	optional, multiple
<b>mbrs-by-ref</b>	list of <i>&lt;mntner-names&gt;</i> ANY	

# Peering Set Object

---

Router 9.9.9.1 imports 128.9.0.0/16 from 9.9.9.2 and 9.9.9.3

peering-set: prng-one  
peering: AS3 at 9.9.9.1

peering-set: prng-two  
peering: prng-one  
peering: AS2 at 9.9.9.1

aut-num: AS1  
import: from prng-two accept {128.9.0.0/16}

# Peering-Set Object 2

---

Defines a set of peerings

Peering Set name: "**prng-**"

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>peering-set</b>	<i>&lt;object-name&gt;</i>	mandatory, single, class key
<b>peering</b>	<i>&lt;peering&gt;</i>	mandatory, multiple

The **peering** attribute defines a peering  
=> used to import or export routes



# Peering Specification

---

**<as-expression> [<router-expression-1>] [at  
<router-expression-2>]**

**| <peering-set-name>**

where

**<as-expression>**: AS numbers, AS sets, **AND**, **OR**, **EXCEPT**

**<router-expression-\*>**: IP addresses, **inet-rtr**'s, **rtr-set**'s, **AND**, **OR**,  
**EXCEPT**

# Part V

---

## Autonomous Systems

# aut-num Objects

---

- aut-num: AS4591
- as-name: Syra-NET
- import: from AS4590
- action pref=1;
- accept AS4590
- export: to AS4590
- announce AS4591
- default: to AS4590
- action pref=1;
- networks {140.222.0.0/16}
- admin-c: Warren Lavallee
- tech-c: Warren Lavallee
- mnt-by: MAINT-AS4591
- changed: warren@Syra.NET 19950522
- source: RADB
-

# Aut-num Object

---

Specifies routing policies

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>aut-num</b>	<i>&lt;as-number&gt;</i>	mandatory, single, class key
<b>as-name</b>	<i>&lt;object-name&gt;</i>	mandatory, single
<b>member-of</b>	list of <i>&lt;as-set-names&gt;</i>	optional, multiple
<b>import</b>	defined later	optional, multiple
<b>export</b>	defined later	optional, multiple
<b>default</b>	defined later	optional, multiple

# Specifying Policies

---

⇒ RPSL allows policies based on:

⇒ prefix

⇒ AS Path

⇒ community

⇒ prefix-length

⇒ future attributes thru its dictionary

⇒ Structured Policy possible

# Specifying Policy in Aut-Num

---

- ⇒ Import
- ⇒ export
- ⇒ default

# Import

---

⇒import:

from <peering-1> [action <action-1>]

. . . .

from <peering-N> [action <action-N>]

accept <filter>

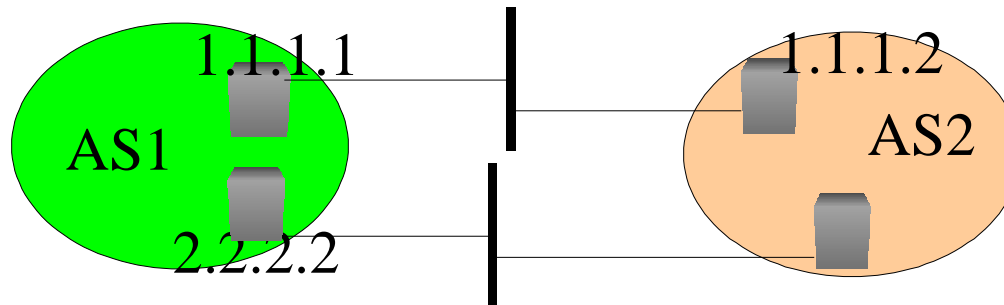
⇒Set of routes matched by **filter**

⇒imported from all peers in **peerings**

⇒While importing routes at <**peering-M**>

⇒<**action-M**> is done

# Choosing a Peering



- `aut-num: AS1`
- `import: from AS2 at 2.2.2.2`
- `action pref = 10;`
- `accept AS2`
- `import: from AS2 1.1.1.2 at 1.1.1.1`
- `action pref = 5;`
- `accept AS2`



# Export

---

⇒ export:

to <peering-1> [action <action-1>]

. . .

to <peering-N> [action <action-N>]

announce <filter>

⇒ Set of routes matched by **filter**

⇒ exported to all peers in <peerings>

⇒ While exporting routes at <peering-M>

⇒ <action-M> is done

# Default

---

⇒ default:

⇒ to <peering> [action <action>] [networks <filter>]

⇒ Local AS defaults to the AS in <peering>

⇒ <action> == attributes of defaulting

⇒ <filter> == policy filter

⇒ Router only uses the default policy

⇒ if it received the routes matched by <filter>

# Examples of Default

---

**AS1** defaults to **AS2** and uses **128.9.0.0/16**

**aut-num:**        **AS1**

**default:**        to **AS2** networks { **128.9.0.0/16** }

**AS1** defaults to **AS2** and **AS3**, but prefers **AS2** over **AS3**

**aut-num:**        **AS1**

**default:**        to **AS2** action pref = 1;

**default:**        to **AS3** action pref = 2;

# Import, Export

---

⇒ Other Routing Protocols

⇒ Multi-Protocol Routing Protocols

⇒ Injecting Routes between protocols

# Import, Export 2

---

Complete syntax of **import**, **export** attributes:

**import:** [protocol <protocol-1>] [into <protocol-2>]  
from <peering-1> [action <action-1>]  
.....  
from <peering-N> [action <action-N>]  
accept <filter>

**export:** [protocol <protocol-1>] [into <protocol-2>]  
to <peering-1> [action <action-1>]  
.....  
to <peering-N> [action <action-N>]  
announce <filter>

# Import, Export 3

---

⇒ Valid protocols  
⇒ in RPSL dictionary

⇒ Default protocol is Exterior Gateway Protocol  
⇒ BGP

# Part VI

---

## Internet Router

# Inet-Rtr Object

---

- `inet-rtr:` `c56-11.t3.ans.net`
- `local-as:` `AS1664`
- `ifaddr:` `140.222.56.200 masklen 26`
- `ifaddr:` `140.222.56.65 masklen 26`
- `ifaddr:` `204.151.29.9 masklen 30`
- `peer:` `BGP4 140.222.56.199`  
`asno (AS1673)`
- `peer:` `BGP4 140.222.56.66`  
`asno (AS1673)`
- `admin-c:` `Steve Heimlich`
- `tech-c:` `Selina Priestley`
- `mnt-by:` `ANS`
- `changed:` `configs@ans.net 19970320`
- `source:` `ANS`
-



# Inet-Rtr Object 2

<u>Attribute</u>	<u>Value</u>	<u>Type</u>
<b>inet-rtr</b>	<i>&lt;dns-name&gt;</i>	Mandatory, single, class key
<b>alias</b>	<i>&lt;dns-name&gt;</i>	Optional, multiple
<b>local-as</b>	<i>&lt;as-number&gt;</i>	Mandatory, single
<b>ifaddr</b>	Described below	Mandatory, multiple
<b>peer</b>	Described later	Optional, multiple
<b>member-of</b>	list of <i>&lt;rtr-set-names&gt;</i>	Optional, multiple
<b>ifaddr:</b>	<i>&lt;ipv4-address&gt; masklen &lt;integer&gt; [action &lt;action&gt;]</i>	

# Inet-Rtr Object 3

---

The **peer** attribute:

```
<protocol> <ipv4-address>    <options>  
| <protocol> <inet-rtr-name>    <options>  
| <protocol> <rtr-set-name>    <options>  
| <protocol> <peering-set-name> <options>
```

<protocol> == defined in RPSL dictionary

# Inet-Rtr Object 4

---

## Example:

**rtr-set:        rtrs-ibgp-peers**  
**members:        1.1.1.1, 2.2.2.2, 3.3.3.3**

**peering-set:    prng-ebgp-peers**  
**peering:        AS3334 192.87.45.195**  
**peering:        AS3335 192.87.45.196**

# Inet-Rtr Object 5

---

Example of inet-rtr object with peering groups:

```
inet-rtr:      Amsterdam.ripe.net
alias:         amsterdam1.ripe.net
local-as:      AS3333
ifaddr:        192.87.45.190 masklen 24
ifaddr:        192.87.4.28   masklen 24
ifaddr:        193.0.0.222   masklen 27
ifaddr:        193.0.0.158   masklen 27
peer:          BGP4 rtrs-ibgp-peers asno(AS3333), flap_damp()
peer:          BGP4 prng-ebgp-peers asno(PeerAS), flap_damp()
```

# Part VII

---

## Advanced Features

# Advanced Features

---

⇒ Extending RPSL

⇒ "dictionary" object

⇒ Aggregating Routes

⇒ Static Routes

⇒ Structured Policy

# Extending RPSL

---

⇒ **"dictionary"** class

⇒ New attributes

⇒ New object types (classes)

⇒ Changing the syntax of attributes

# "dictionary" Object

---

- **dictionary:** RPSL
- **rp-attribute:** pref # smaller values are preferred
  - operator=(integer[0, 65535])
- **rp-attribute:** med # BGP multi\_exit\_discriminator attribute
  - operator=(integer[0, 65535])
  - operator=(enum[igp\_cost])
- **typedef:** community\_elm union
  - integer[1, 4294967200],
  - enum[internet, no\_export, no\_advertise],
  - list[2:2] of integer[0, 65535]
- **rp-attribute:** community # BGP community attribute
  - operator.=(community\_elm)
  - operator()(community\_elm, ...)
- ...



# "dictionary" Object 2

---

- ⇒ "dictionary" object
  - ⇒ main way to extend RPSL
- ⇒ Dictionary objects define
  - ⇒ routing policy attributes
  - ⇒ types
  - ⇒ routing protocols
- ⇒ Routing policy attribute
  - ⇒ rp-attribute

# "dictionary" Object 3

---

⇒ The rp–attribute

⇒ actual protocol attributes

⇒ e.g. BGP path attributes

⇒ router features

⇒ e.g. BGP route flap dampening

⇒ rp–attributes

⇒ accessed using access methods

⇒ describe policy filters and actions

⇒ Well–known dictionary object: **RPSL**

⇒ All tools use this by default

# Aggregation

---

- route: 128.8.0.0/15
- origin: AS1
- components: {128.8.0.0/15^-}
- aggr-mtd: outbound AS-ANY
- inject: at 1.1.1.1 action dpa = 100;
- inject: at 1.1.1.2 action dpa = 110;
-

# Static Routes

---

- route: 128.7.0.0/16
- origin: AS1
- inject: at 7.7.7.1
  - action next-hop = 7.7.7.2; cost = 10;
  - upon static
- inject: at 7.7.7.1
  - action next-hop = 7.7.7.3; cost = 20;
  - upon static
-

# Structured Policy

---

- `aut-num: AS3561`
- `import: { from AS-ANY action pref = 30;`
- `accept community(3561:70);`
- `from AS-ANY action pref = 20;`
- `accept community(3561:80);`
- `} refine {`
- `from AS1 accept`
- `AS1:AS-Customers;`
- `from AS2 accept AS2;`
- `from AS3 accept ...;`
- `}`
- 
- 
-

# Structured Policy

---

```
• aut-num: AS3561
• import: {
•     from AS-ANY
•         accept any and not
rs-martians;
•     } refine {
•         from AS-ANY action pref = 30;
•         accept community(3561:70);
•         from AS-ANY action pref = 20;
•         accept community(3561:80);
•     } refine {
•         from AS1 accept
AS1:AS-Customers;
•         from AS2 accept AS2;
•         from AS3 accept ...;
•     }
```

# Structured Policy

---

- `aut-num: AS3561`
- `import: { from AS-ANY action pref = 30;`
- `accept community(3561:70);`
- `from AS-ANY action pref = 20;`
- `accept community(3561:80);`
- `} refine {`
- `from AS1 accept AS1:AS-Customers;`
- `} except {`
- `from AS2 accept AS2;`
- `from AS3 accept AS3;`
- `}`
- `AS1:AS-Customers contains AS2 and AS3`

# More Complete Example

```
aut-num:      AS2764
as-name:      ASN-CONNECT-NET
descr:        connect.com.au Pty Ltd
import: {
    from AS-ANY action community .= {2764:65408};
    accept ANY AND NOT { 0.0.0.0/0 };
} refine {
    from AS-ANY action community={internet}; pref=0;
    accept community(2764:65280,...,2764:65412);
    from AS-ANY action pref=25;
    accept community(2764:3)
        AND NOT AS2764:RS-PROVIDER^-;
    ...
    from AS-ANY action pref=0; accept ANY;
} refine {
    from AS2764:AS-GLOBAL
        accept PeerAS AND <^PeerAS$>;
    ...
}
```



# Practical Example

---

⇒ RAToolSet

⇒ <http://www.isi.edu/ra/RAToolSet/>

⇒ RtConfig

# AS Numbers in Policy

---



⇒ route: 128.9.0.0/16      route: 128.8.0.0/16

⇒ origin: AS1                      origin: AS1

⇒

⇒ aut-num: AS1

⇒ export: to AS2    announce AS1

⇒

AS1 == {128.9.0.0/16, 128.8.0.0/16}  
⇒ aut-num: AS2  
⇒ import: from AS1 accept AS1

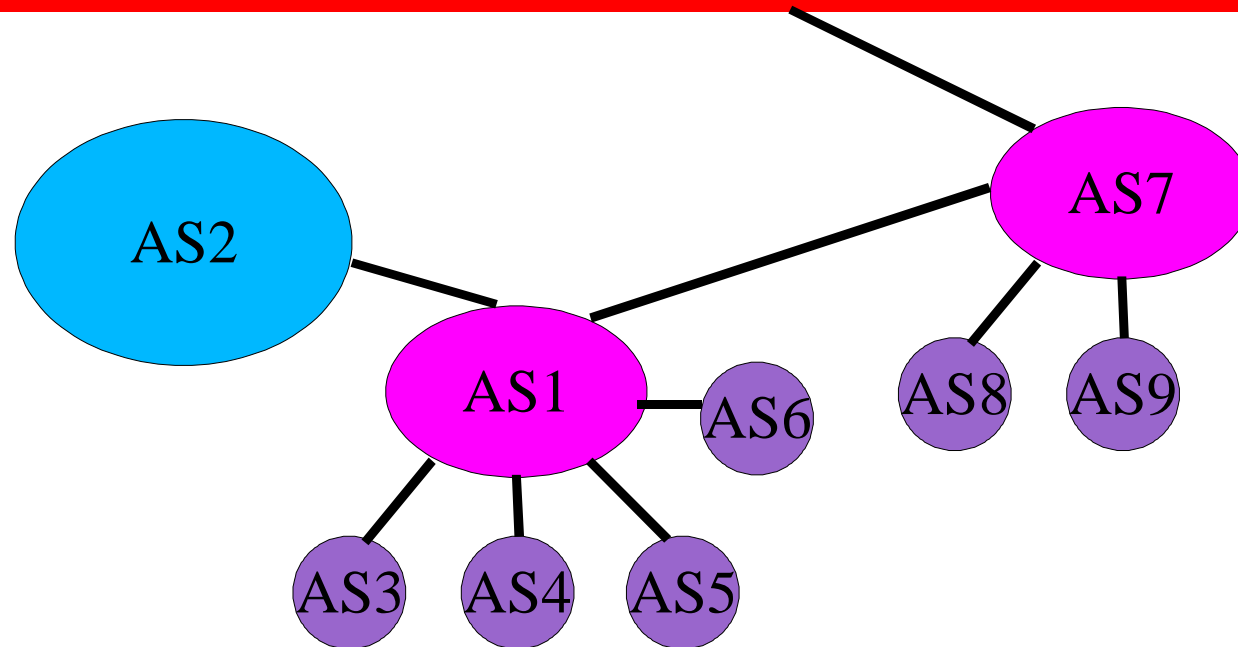
# RtConfig:

## A tool to configure routers

---

```
% RtConfig
RtConfig> @RtConfig export AS1 1.1.1.1 AS2 2.2.2.2
no access-list 1
access-list 1 permit ip 128.8.0.0 0.0.0.0 255.255.0.0 0.0.0.0
access-list 1 permit ip 128.9.0.0 0.0.0.0 255.255.0.0 0.0.0.0
access-list 1 deny ip 0.0.0.0 255.255.255.255 0.0.0.0
    255.255.255.255
!
no route-map foo
route-map foo permit 1
    match ip address 1
!
!
```

# AS Path Based



⇒import: from AS1 accept <^AS1 .\* AS8\$>

⇒

⇒import: from AS1 accept <^AS1

AS1:AS-Customers\* \$> **No prefixes here!**

# RtConfig

---

- RtConfig> @RtConfig import AS2 1.1.1.1 AS1 2.2.2.2
- !
- no ip as-path access-list 1
- ip as-path access-list 1 permit ^\_1((\_[0-9]+))\*\_8\$
- !
- no route-map foo
- route-map foo permit 1
- match as-path 1
- !
- router bgp 2
- neighbor 2.2.2.2 route-map foo in
- 
-

# RtConfig

- aut-num: AS1
- export: to AS2
- announce AS2764 AND NOT { 0.0.0.0/0 }
- AND <^AS2764\*\$>
- AND NOT community({2764,1})
- 
- RtConfig> @RtConfig export AS1 1.1.1.1 AS2 2.2.2.2
- access-list 1 permit ip 210.8.248.0 0.0.0.0 255.255.248.0 0.0.0.0
- ...
- access-list 1 permit ip 210.9.2.0 0.0.0.0 255.255.254.0 0.0.0.0
- access-list 1 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
- !
- no ip as-path access-list 1
- ip as-path access-list 1 permit ^(\s2764)\*\$

# RtConfig (cont)

---

- `ip bgp-community new-format`
- `no community-list 1`
- `ip community-list 1 deny 2764:1`
- `ip community-list 1 permit internet`
- `!`
- `no route-map foo`
- `route-map foo permit 1`
- `match as-path 1`
- `match community 1`
- `match ip address 1`
- `!`
- `router bgp 1`
- `neighbor 0.0.0.0 route-map foo out`
-

# RtConfig

---

- @RtConfig import AS1 1.1.1.1 AS2 2.2.2.2
- @RtConfig export AS1 1.1.1.1 AS2 2.2.2.2
- @RtConfig networks AS1
- @RtConfig default AS1 AS2
- @RtConfig pkt\_filter "eth0" AS1 1.1.1.1 AS2 2.2.2.2
- @RtConfig outbound\_pkt\_filter "eth0" AS1 1.1.1.1 AS2 2.2.2
- @RtConfig static2bgp AS1 1.1.1.1
- 
- Not yet available:
- @RtConfig inet-rtr c56-11.t3.ans.net
- 
- Configuration formats:
- cisco, bay, gated, rsd



# Acknowledgements

---

⇒ Cengiz Alaettinoglu

⇒ USC Information Sciences Institute

⇒ Provided slides from which these slides are derived

⇒ But any errors are the responsibility of A. M. R. Magee

⇒ RIPE NCC

⇒ Eng Group

⇒ Feedback & Comments