# APNIC Training

## Internet Routing Registry

20 July 2009 – Chennai, India

In conjunction with

SANOG

# Introduction

- Presenters
  - Champika Wijayatunga
    - Team Leader (Training)
    - champika@apnic.net

# Objectives

- To provide an introduction to the APNIC Routing Registry
  - Explain basic concepts of the global RR
  - Outline the benefits of the APNIC Routing Registry

- NOT to:
  - Teach basic routing
  - Explain Internet resource policy and procedures
  - Provide advise on network configuration

# Internet Routing Registry

Overview

# Overview

- APNIC database recap
- What is IRR?
- Why use an IRR?
- APNIC database and the IRR
- Using the Routing Registry
- Benefit of using IRR
- Using RPSL in practice

# APNIC database recap

# APNIC database

- Public network management database
  - APNIC whois database contains:
    - Internet resource information and contact details
  - APNIC Routing Registry (RR) contains:
    - routing information
- APNIC RR is part of IRR
  - Distributed databases that mirror each other

# Database object

- An object is a set of attributes and values
- Each attribute of an object...
    - Has a value
    - Has a specific syntax
    - Is mandatory or optional
    - Is single- or multi-valued

- Some attributes ...
    - Are primary (unique) keys
    - Are lookup keys for queries
    - Are inverse keys for queries

– Object “templates” illustrate this structure

# Person object example

- Person objects contain contact information

Values

```
person:     Ky Xander
address:    ExampleNet Service Provider
address:    2 Pandora St Boxville
address:    Wallis and Futuna Islands
country:    WF
phone:      +680-368-0844
fax-no:     +680-367-1797
e-mail:     kxander@example.com
nic-hdl:    KX17-AP
mnt-by:     MAINT-ENET-WF
changed:    kxander@example.com 20020731
source:     APNIC
```

# Querying whois db

- Unix
  - Whois –h whois.apnic.net <lookup key>
    - E.g. whois –h whois.apnic.net whois AS2000
- Whois web interface
  - http://www.apnic.net/apnic-bin/whois.pl
- Keys for querying
  - Primary key, other lookup keys
    - E.g. whois EX91-AP
  - Inverse key “-i {attribute} {value}”
    - E.g. whois -i mnt-by MAINT-EXAMPLE-AP
- APNIC whois db query options:
  - http://www.apnic.net/db/search/all-options.html

# Advanced database queries

– Flags used for inetnum queries

| | |
|---|---|
| None | find exact match |
| - l | find one level less specific matches |
| - L | find all less specific matches |
| - m | find first level more specific matches |
| - M | find all More specific matches |
| - x | find exact match (if no match, nothing) |
| - d | enables use of flags for reverse domains |
| - r | turn off recursive lookups |

# Database update process

- Update transactions
  - – Create a new object
  - – Change an object
  - – Delete an object

Template

- Updates are submitted by email
  - E-mail to: **<auto-dbm@apnic.net>**
- Email message contains template representing new or updated object

# Database protection

- Authorisation
  - “mnt-by” references a mntner object
    - Can be found in all database objects
    - “mnt-by” should be used with every object!

- Authentication
  - Updates to an object must pass authentication rule specified by its maintainer object

# Authentication methods

- 'auth' attribute
  - Crypt-PW
    - Crypt (Unix) password encryption
    - Use web page to create your maintainer
  - PGP – GNUPG
    - Strong authentication
    - Requires PGP keys
  - MD5
    - Available

# Hierarchical authorisation

- ‘mnt-by’ attribute
  - Can be used to protect any object
  - Changes to protected object must satisfy authentication rules of ‘mntner’ object

- ‘mnt-lower’ attribute
  - Also references mntner object
  - Hierarchical authorisation for inetnum & domain objects
  - Creation of child objects must satisfy this mntner
  - Protects against unauthorised updates to an allocated range - highly recommended!

# Prerequisite for updating objects

- Create person objects for contacts
  - To provide contact info in other objects
- Create a mntner object
  - To provide protection of objects
- Protect your person object

# What is an IRR?

# What is a Routing Registry?

- A repository (database) of Internet routing policy information
  - Autonomous Systems exchanges routing information via BGP
  - Exterior routing decisions are based on policy based rules
  - However BGP does not provides a mechanism to publish/communicate the policies themselves
  - RR provides this functionality
- Routing policy information is expressed in a series of objects

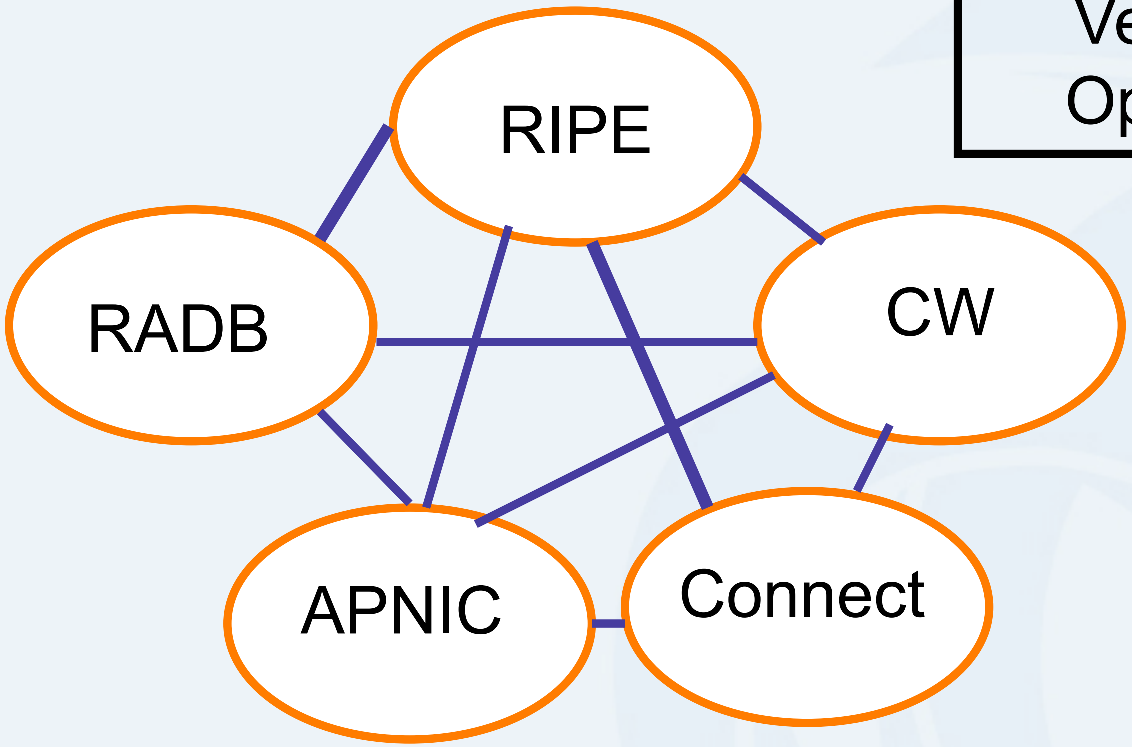
# Routing registry objects

- Route, aut-num, inet-rtr, peering-set, AS-set, rtr-set, filter-set
  - Each object has its own purpose
  - Together express routing policies
- More details covered later

# What is a Routing Registry?

- Global Internet Routing Registry database
  - http://www.irr.net/
    - Uses RPSL
- Stability and consistency of routing
  - network operators share information
- Both public and private databases
  - These databases are independent
    - but some exchange data
    - only register your data in one database

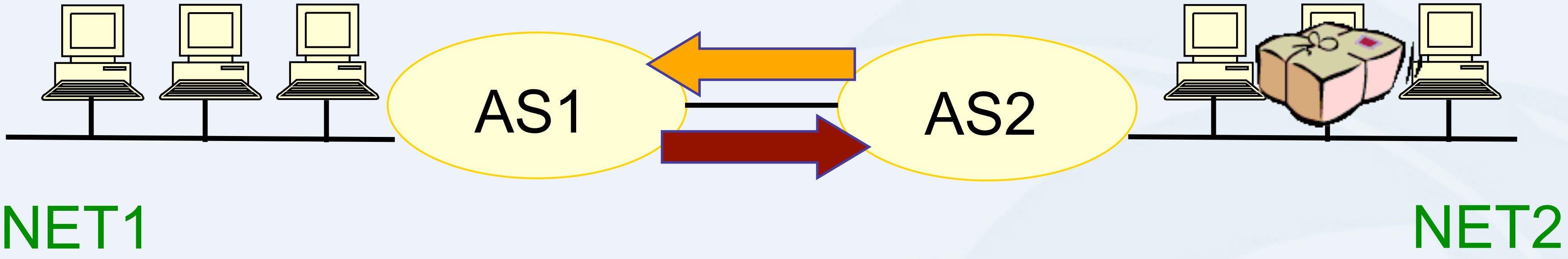
# What is a Routing Registry?



ARIN, ArcStar, FGC,
Verio, Bconnex,
Optus, Telstra, ...

IRR = APNIC RR + RIPE DB + RADB + C&W + ARIN + …

# Representation of routing policy



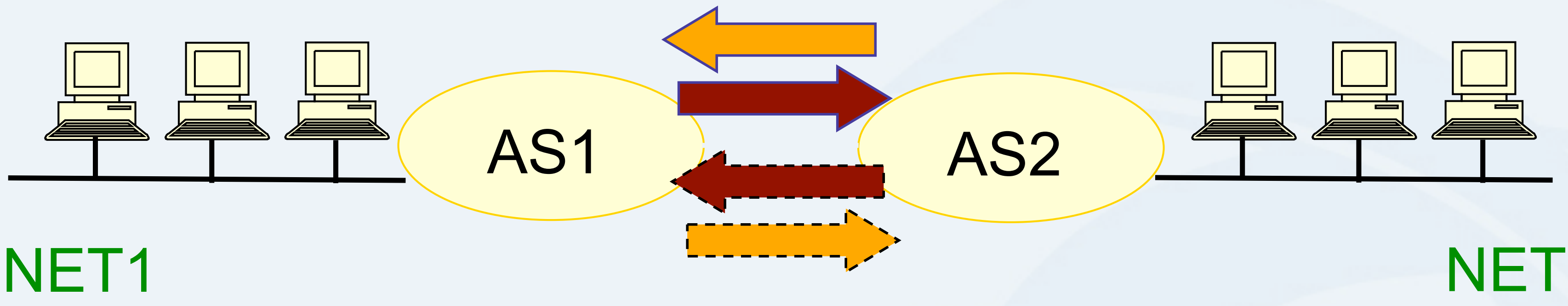
In order for traffic to flow from NET2 to NET1 between AS1 and AS2:

AS1 has to announce NET1 to AS2 via BGP

And AS2 has to accept this information and use it

Resulting in packet flow from NET2 to NET1

# Representation of routing policy (cont.)



In order for traffic to flow towards from NET1 to NET2:

AS2 must announce NET2 to AS1

And AS1 has to accept this information and use it

Resulting in packet flow from NET 1 to NET2

# What is routing policy?

- Description of the routing relationship between autonomous systems
  - Who are my BGP peers?
    - Customer, peers, upstream
  - What routes are:
    - Originated by each neighbour?
    - Imported from each neighbour?
    - Exported to each neighbour?
    - Preferred when multiple routes exist?
  - What to do if no route exists?
  - What routes to aggregate?

# Why use an IRR?

# Information to share

- Routes and AS objects give an abstract specification of the policy of an AS
    - Provides device independent view of routing policy
    - Neighbouring ASes can lookup, verify and understand the other party's policy
    - Provides a clear picture where this AS fits into the Internet

# Information to share (cont.)

- Information – if every AS registers its policy and routes….
  - a global view of routing policy could be mapped
    - This global picture has the ability to improve the integrity of global Internet routing
  - Provides LIR/ISP with a mechanism to find all possible paths between any two points in the Internet
- Provides a high level of abstraction

# Router configuration and network troubleshooting

- Router configuration
  – By using IRRToolSet
    - https://www.isc.org/software/irrtoolset-485
    - Extract information from IRR to create a router readable configuration file
    - Vendor independent
    - Protect against inaccurate routing info distribution
    - Verification of Internet routing
- Network troubleshooting
  – Easier to locate routing problems outside your network

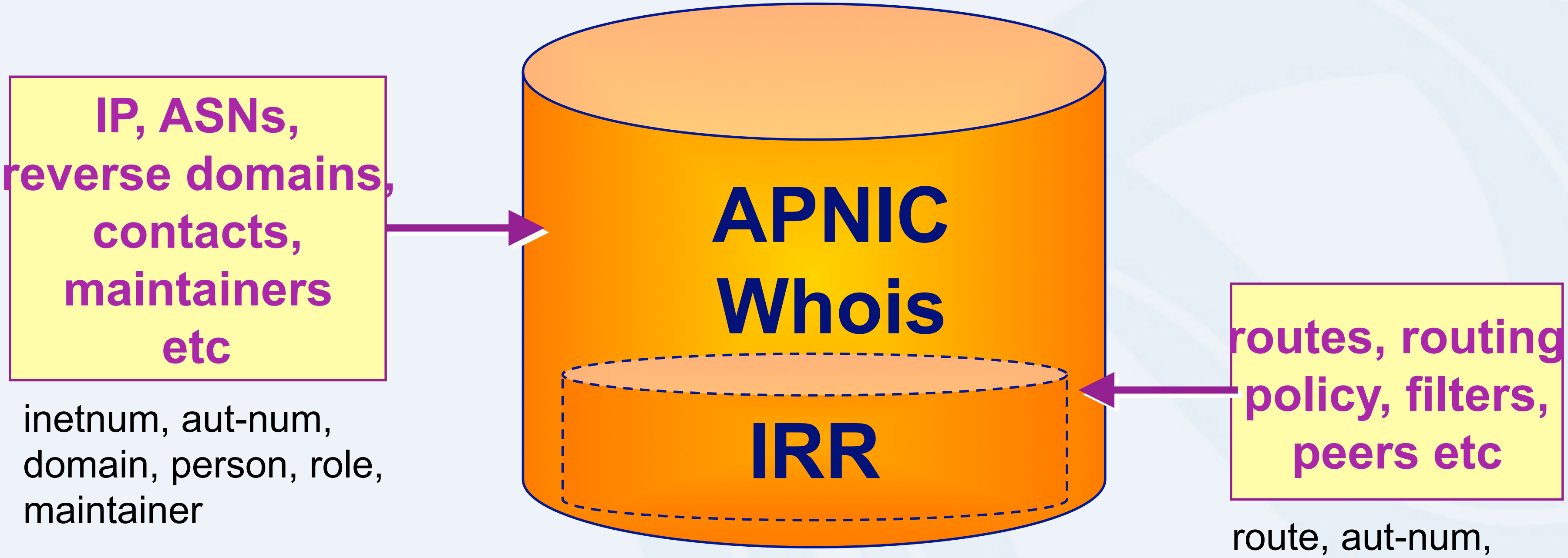
# APNIC database and the IRR

# APNIC Database & the IRR

- APNIC whois Database
  - Two databases in one
- Public Network Management Database
  - "whois" info about networks & contact persons
    - IP addresses, AS numbers etc
- Routing Registry
  - contains routing information
    - routing policy, routes, filters, peers etc.
  - APNIC RR is part of the global IRR

# Integration of Whois and IRR

- Integrated APNIC Whois Database & Internet Routing Registry


IP, ASNs, reverse domains, contacts, maintainers etc
APNIC Whois
IRR
routes, routing policy, filters, peers etc

inetnum, aut-num,
domain, person, role,
maintainer

route, aut-num,
as-set, inet-rtr,
peering-set etc.

Internet resources & routing information

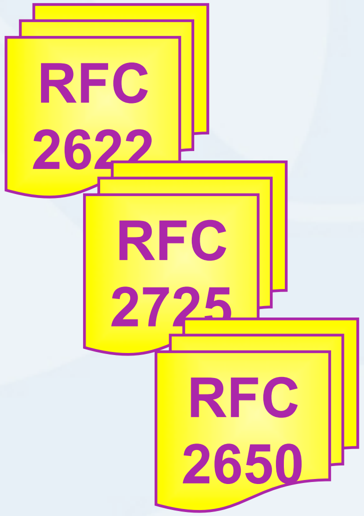
# RPSL

- Routing Policy Specification Language
  - Object oriented language
    - Based on RIPE-181
  - Structured whois objects

- Higher level of abstraction than access lists

- Describes things interesting to routing policy:
  - Routes, AS Numbers …
  - Relationships between BGP peers
  - Management responsibility

- Relevant RFCs
  - Routing Policy Specification Language
  - Routing Policy System Security
  - Using RPSL in Practice

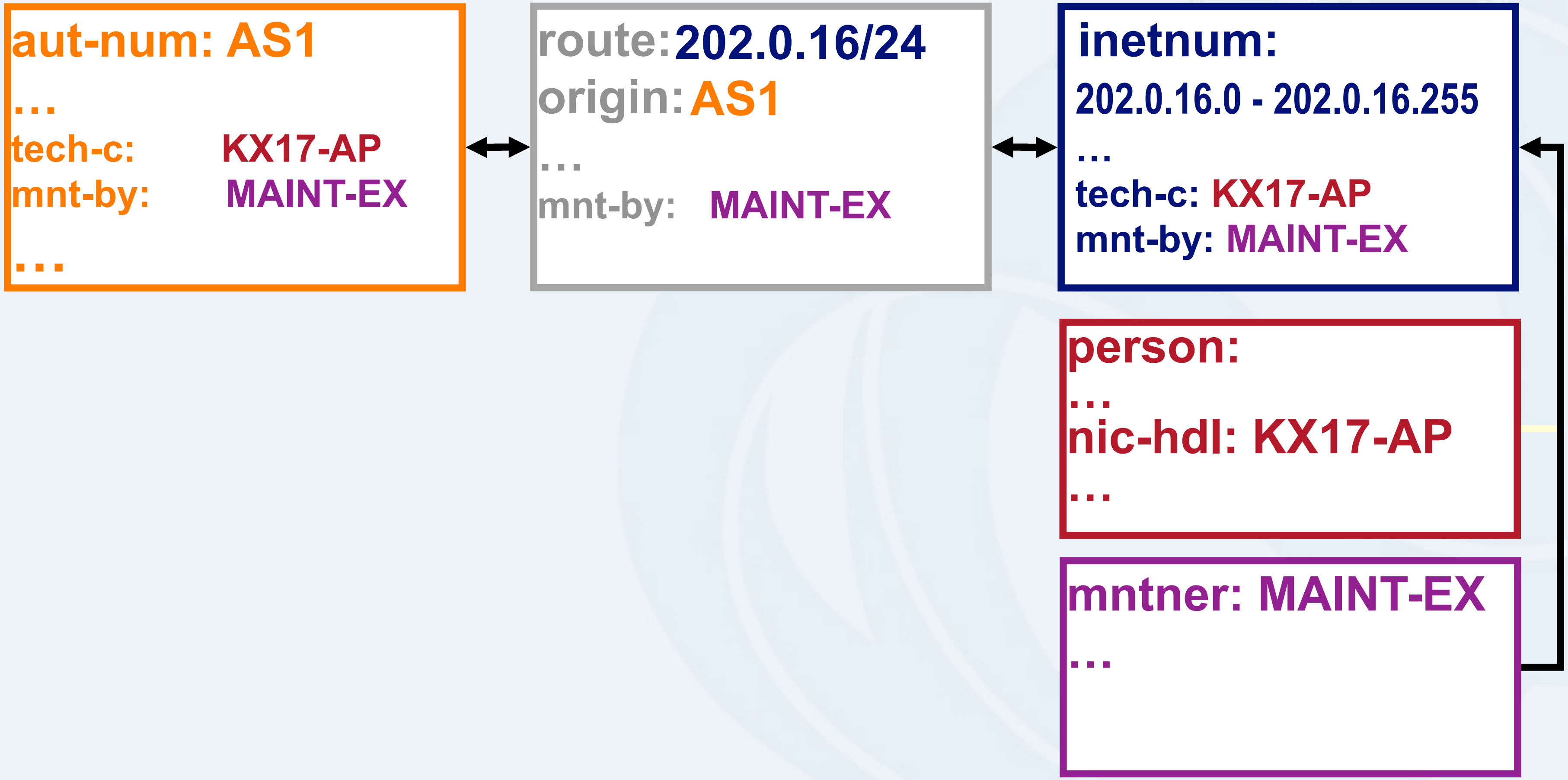


# IRR objects

- route
  - Specifies interAS routes
- aut-num
  - Represents an AS. Used to describe external routing policy
- inet-rtr
  - Represents a router
- peering-set
  - Defines a set of peerings
- route-set
  - Defines a set of routes
- as-set
  - Defines a set of **aut-num** objects
- rtr-set
  - Defines a set of routers
- filter-set
  - Defines a set of routes that are matched by its filter

www.apnic.net/db/ref/db-objects.html

# Inter-related IRR objects



aut-num: AS1
…
tech-c: KX17-AP
mnt-by: MAINT-EX
…

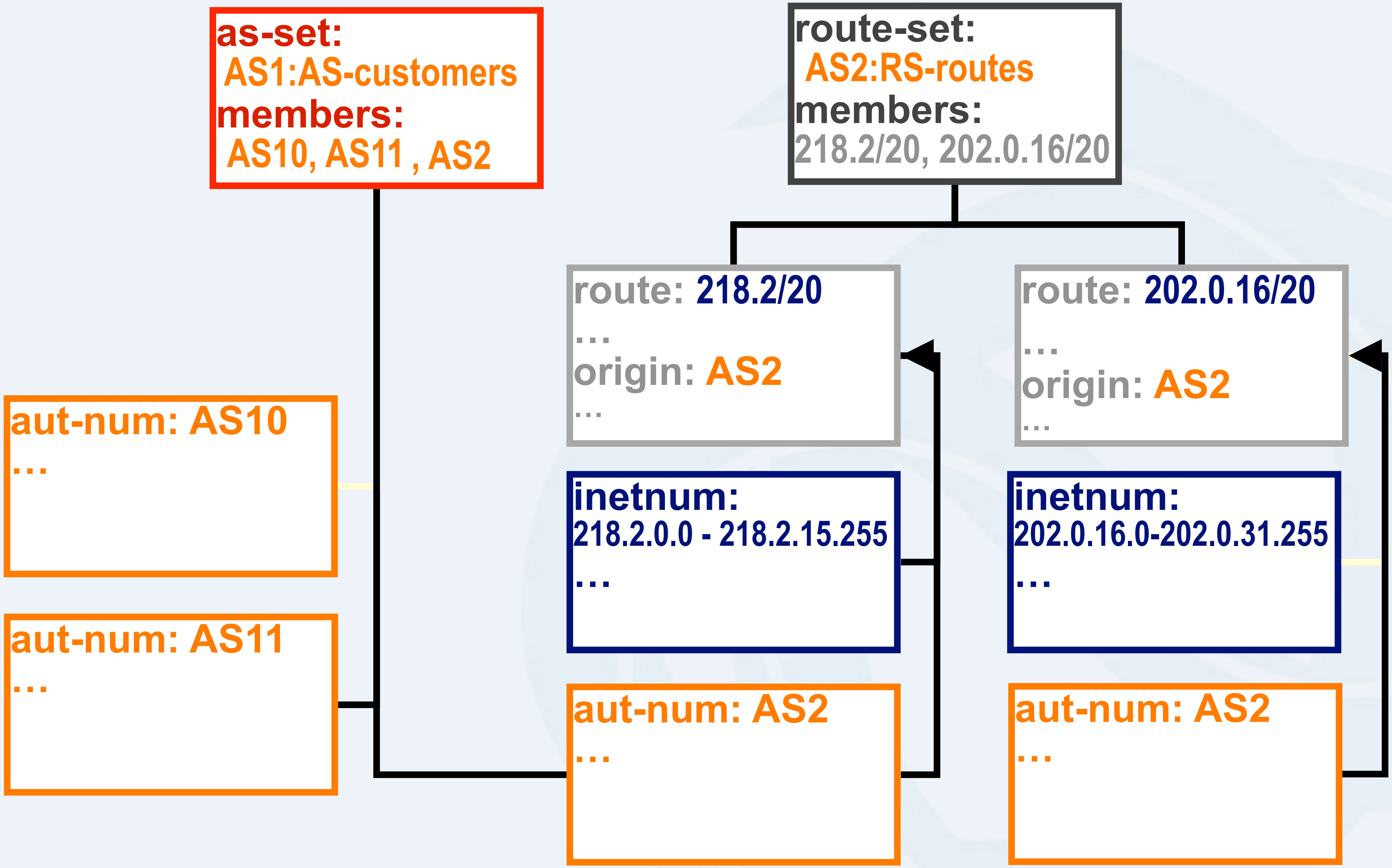
route: 202.0.16/24
origin: AS1
…
mnt-by: MAINT-EX

inetnum:
202.0.16.0 - 202.0.16.255
…
tech-c: KX17-AP
mnt-by: MAINT-EX

person:
…
nic-hdl: KX17-AP
…

mntner: MAINT-EX
…

# Inter-related IRR objects



as-set:
AS1:AS-customers
members:
AS10, AS11 , AS2

route-set:
AS2:RS-routes
members:
218.2/20, 202.0.16/20

route: 218.2/20
…
origin: AS2
…

route: 202.0.16/20
…
origin: AS2
…

aut-num: AS10
…

aut-num: AS11
…

inetnum:
218.2.0.0 - 218.2.15.255
…

inetnum:
202.0.16.0-202.0.31.255
…

aut-num: AS2
…

aut-num: AS2
…

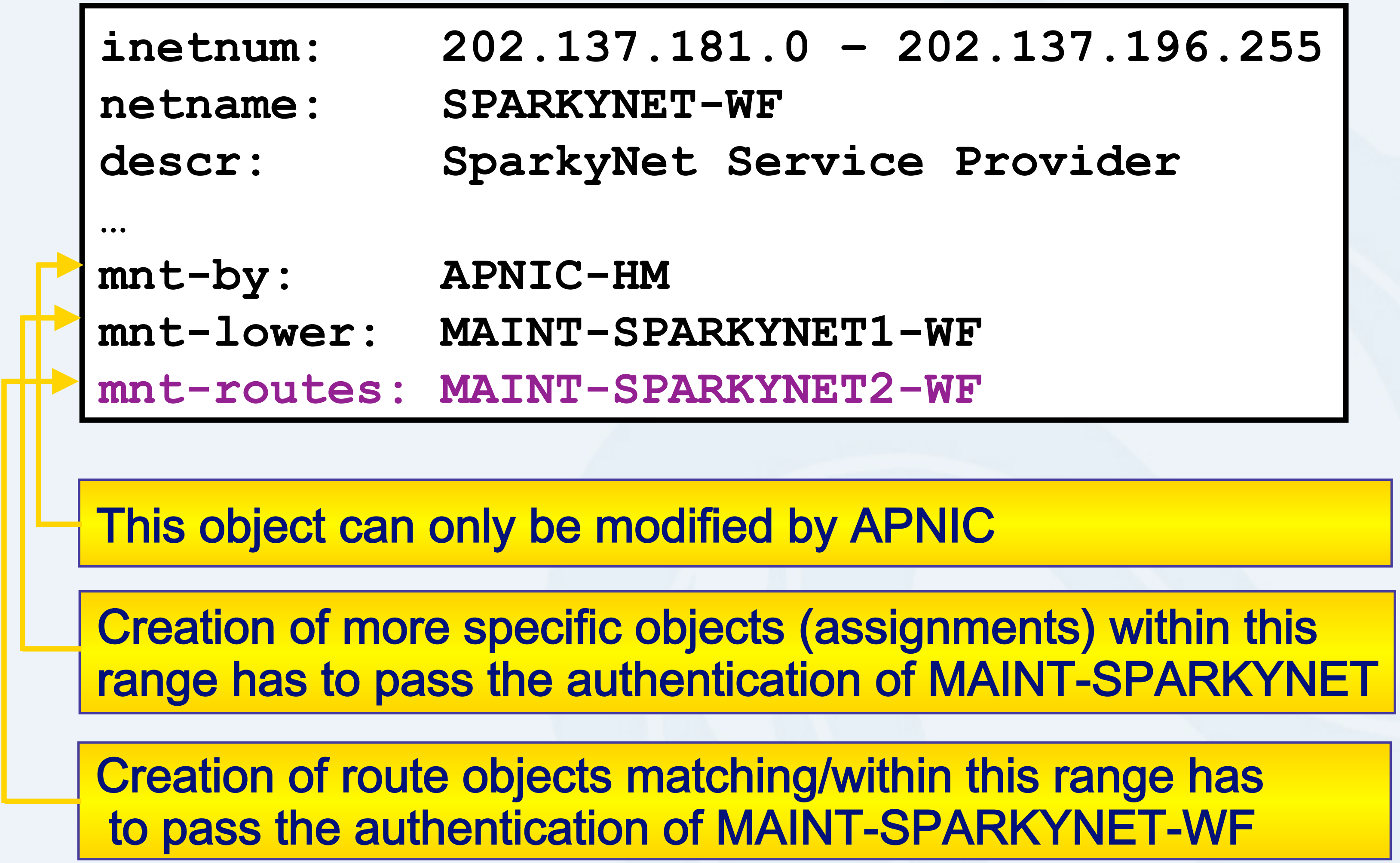
# Hierarchical authorisation

- **mnt-routes**
  - authenticates ***creation*** of route objects
    - creation of route objects must pass authentication of mntner referenced in the mnt-routes attribute
  - Format:
    - `mnt-routes:    <mntner>`

<u>In:</u>

**inetnum** , **aut-num** and **route** objects

# Authorisation mechanism



```
inetnum:     202.137.181.0 - 202.137.196.255
netname:     SPARKYNET-WF
descr:       SparkyNet Service Provider
…
mnt-by:      APNIC-HM
mnt-lower:   MAINT-SPARKYNET1-WF
mnt-routes:  MAINT-SPARKYNET2-WF
```

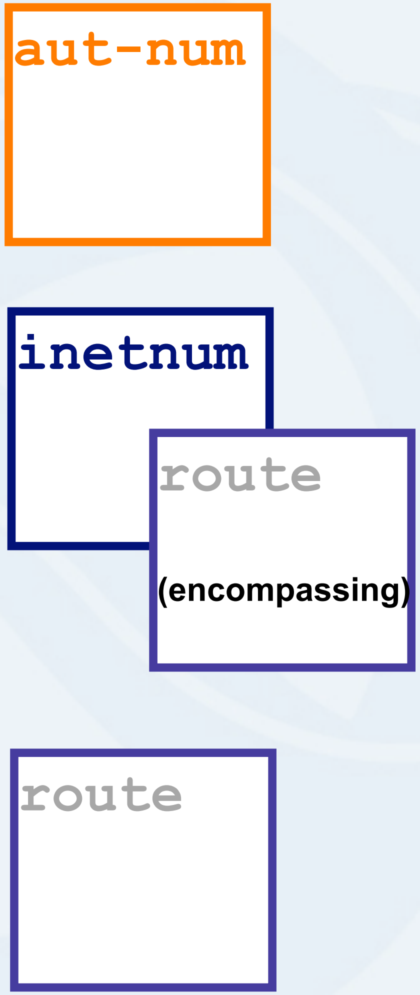
This object can only be modified by APNIC

Creation of more specific objects (assignments) within this range has to pass the authentication of MAINT-SPARKYNET

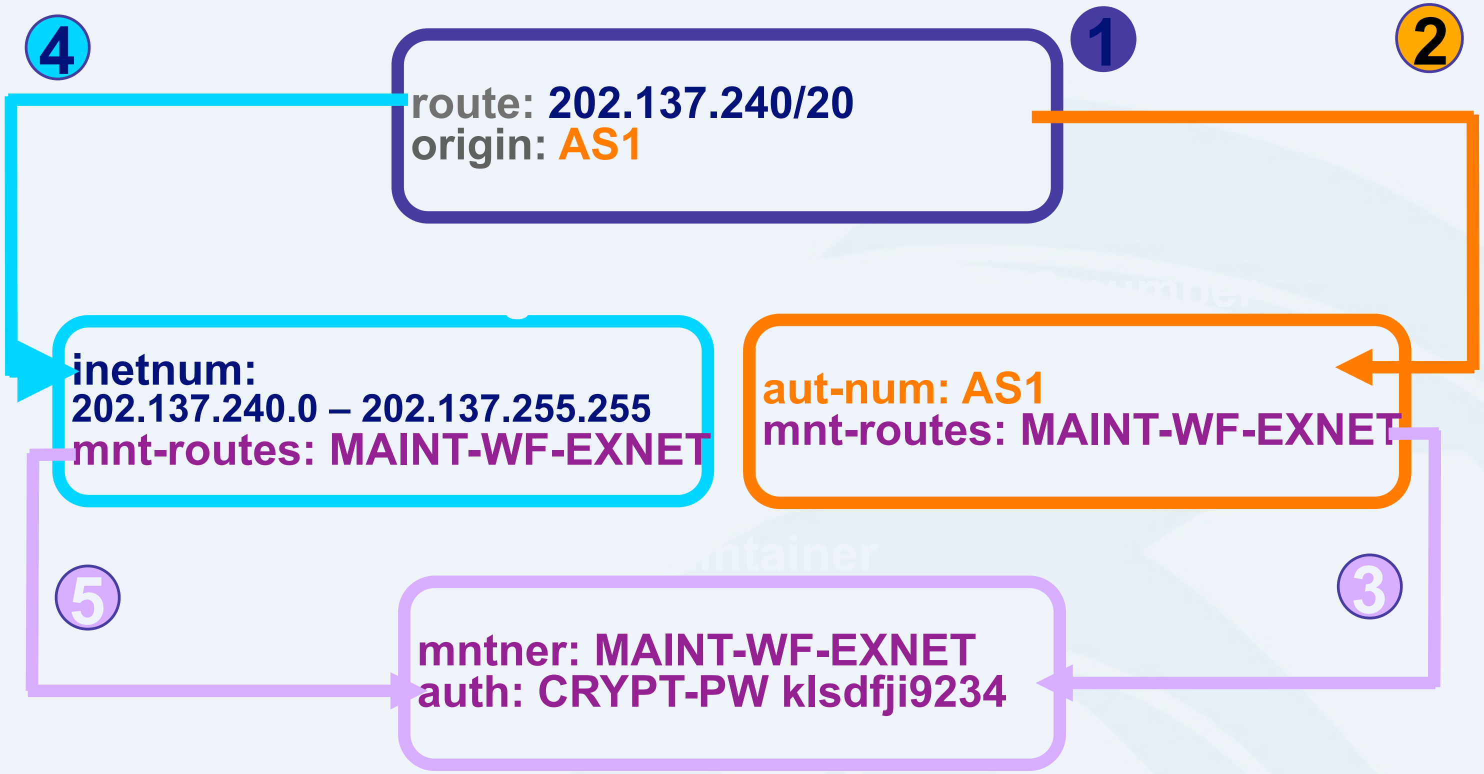
Creation of route objects matching/within this range has to pass the authentication of MAINT-SPARKYNET-WF

# Creating route objects

- Multiple authentication checks:
  - – Originating ASN
    - mntner in the mnt-routes is checked
    - If no mnt-routes, mnt-lower is checked
    - If no mnt-lower, mnt-by is checked
  - – AND the address space
    - Exact match & less specific route
      - - mnt-routes etc
    - Exact match & less specific inetnum
      - - mnt-routes etc
  - – AND the route object mntner itself
    - The mntner in the mnt-by attribute



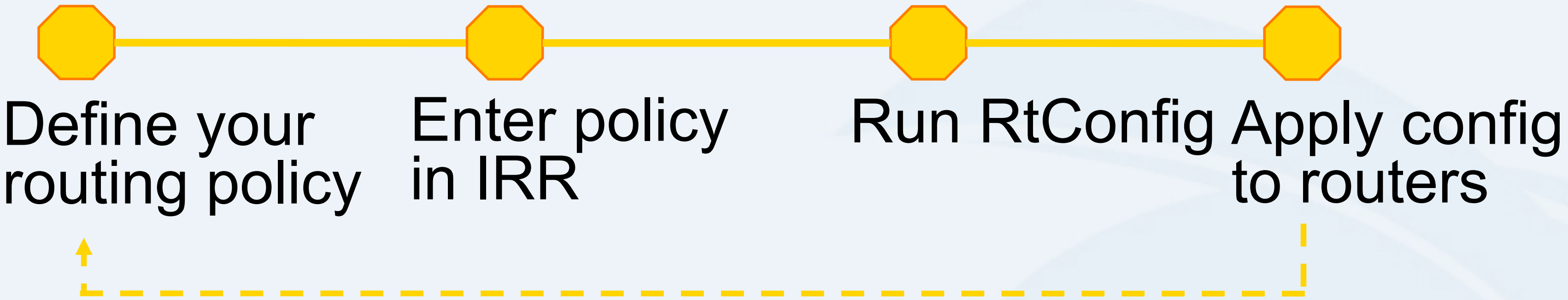
# Creating route objects



1. Create route object and submit to APNIC RR database
2. DB checks aut-num obj corresponding to the ASN in route obj
3. Route obj creation must pass auth of mntner specified in aut-num *mnt-routes* attribute.
4. DB checks inetnum obj matching/encompassing IP range in route obj
5. Route obj creation must pass auth of mntner specified in inetnum *mnt-routes* attribute.

# Benefit of using IRR

# Using the Routing Registry



Define your routing policy

Enter policy in IRR

Run RtConfig

Apply config to routers

## Costs

- Requires some initial planning
- Takes some time to define & register policy
- Need to maintain data in RR

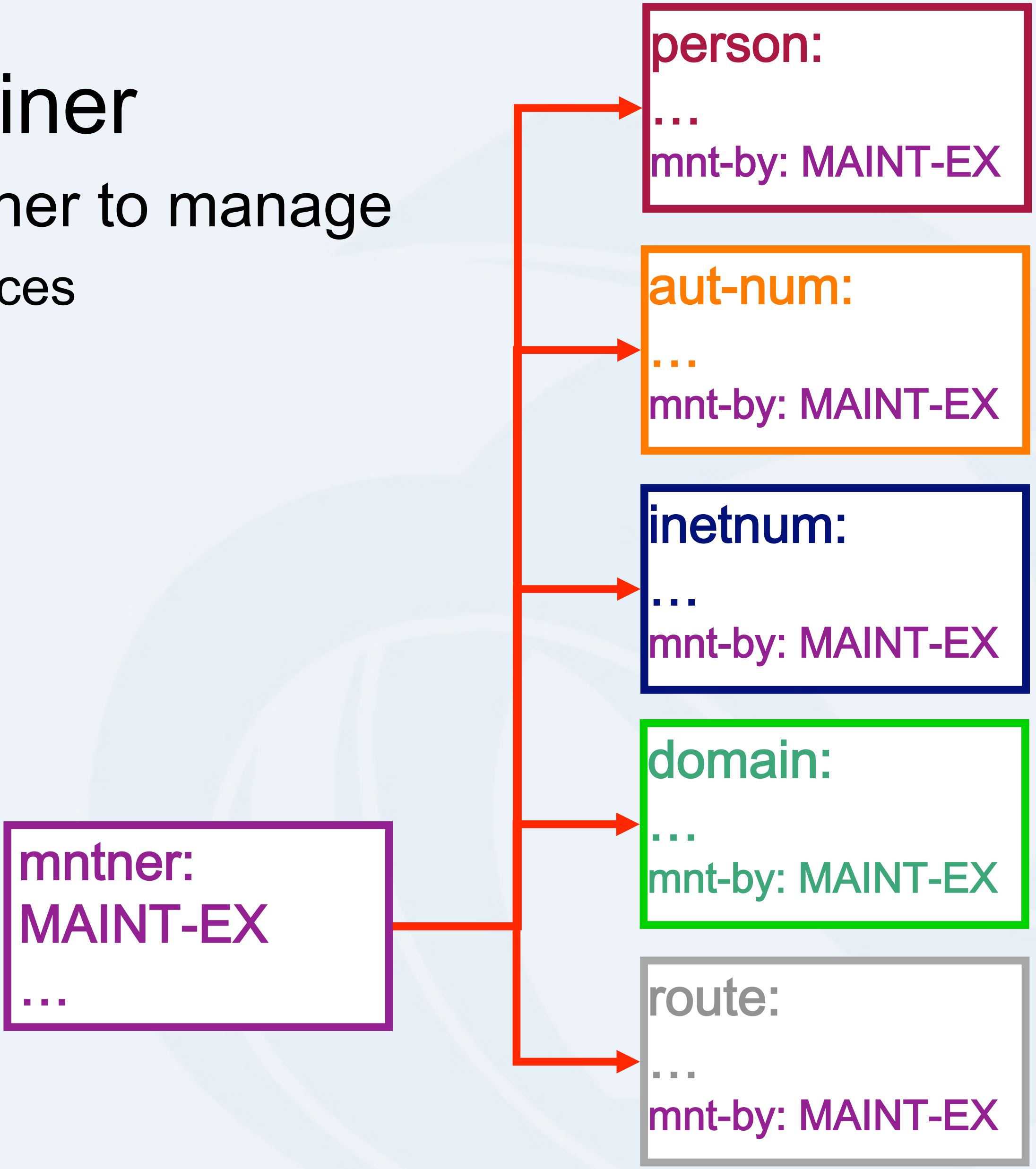
## Benefits

- You have a clear idea of your routing policy
- Consistent config over the whole network
- Less manual maintenance in the long run

# Benefits of APNIC RR

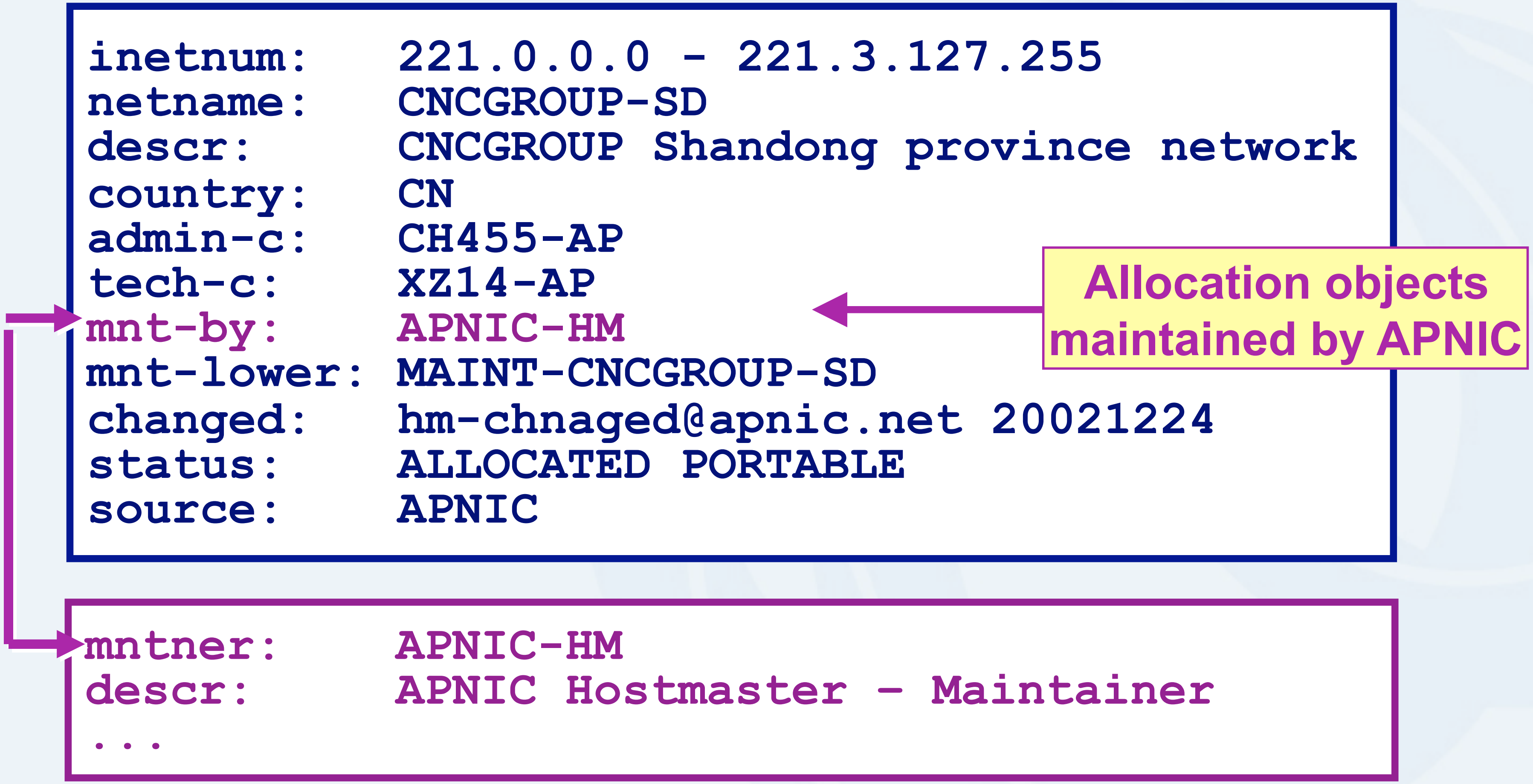
- Single maintainer
  - Use same mntner to manage
    - internet resources
    - reverse DNS
    - routing policy
    - contact info
    - etc

(Single person object can also be used)



# Benefits of APNIC RR

– APNIC able to assert resources for a registered route within APNIC ranges.



```
inetnum:    221.0.0.0 - 221.3.127.255
netname:    CNCGROUP-SD
descr:      CNCGROUP Shandong province network
country:    CN
admin-c:    CH455-AP
tech-c:     XZ14-AP
mnt-by:     APNIC-HM
mnt-lower:  MAINT-CNCGROUP-SD
changed:    hm-chnaged@apnic.net 20021224
status:     ALLOCATED PORTABLE
source:     APNIC
```

Allocation objects maintained by APNIC

```
mntner:     APNIC-HM
descr:      APNIC Hostmaster - Maintainer
...
```

# RPSL

Objects, syntax and semantics

# Overview

- Review of some of RR objects
- Useful queries
- Address prefix range operator
- AS-path regular expression
- Syntax of policy actions and filters

# RPSL

- Purpose of RPSL
  - Allows you to specify your routing configuration in the public IRR
    - Allows you to check "Consistency" of policies and announcements
  - Gives the opportunity to consider the policies and configuration of others
  - There are required syntax and semantics which need to be understood before using RPSL

# RR objects review

- Aut-num object

| Attribute | Value | Type |
|---|---|---|
| aut-num | <as-number> | mandatory, single-valued, class key |
| as-name | <object-name> | mandatory, single-valued |
| member-of | List of <as-set-name> | optional, multi-value |
| import | see next slide | optional, multi-value |
| export | see next slide | optional, multi-value |

# Aut-num object import attribute

- Each import policy expression is specified using an import attribute
- Syntax

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

The action specification is optional.

- Semantics
  - the set of routes that are matched by <filter> are imported from all the peers in <peerings>
  - importing routes at <peering-M>, <action-M> is executed

# Aut-num object export attribute

- Each export policy expression is specified using an export attribute
- Syntax

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

The action specification is optional

- Semantics
  - the set of routes that are matched by <filter> are exported to all the peers specified in <peerings>
  - exporting routes at <peering-M>, <action-M> is executed

# RR objects review

- route object

| Attribute | Value | Type |
|---|---|---|
| route | Prefix of the InterAS route | mandatory, single-valued, class key |
| origin | <AS-number> originates the route | mandatory, single-valued |
| member-of | List of <route-set-name> | optional, multi-value |
| mnt-routes | see slide# 40 | optional, multi-value |

# RR object review

- As-set object

| Attribute | Value | Type |
|---|---|---|
| as-set | <object-name> | mandatory, single-valued, class key |
| members | List of <as-numbers> or <as-set-names> | optional, multi-value |
| Mbrs-by-ref | List of <mntner-names> | optional, multi-value |

- As-set attribute starts with “as-”

# RR object review

- Route-set object

| Attribute | Value | Type |
|---|---|---|
| route-set | <object-name> | mandatory, single-valued, class key |
| members | List of <address-prefix-range> or <route-set-name><range-operator> | optional, multi-value |
| Mbrs-by-ref | List of <mntner-names> | optional, multi-value |

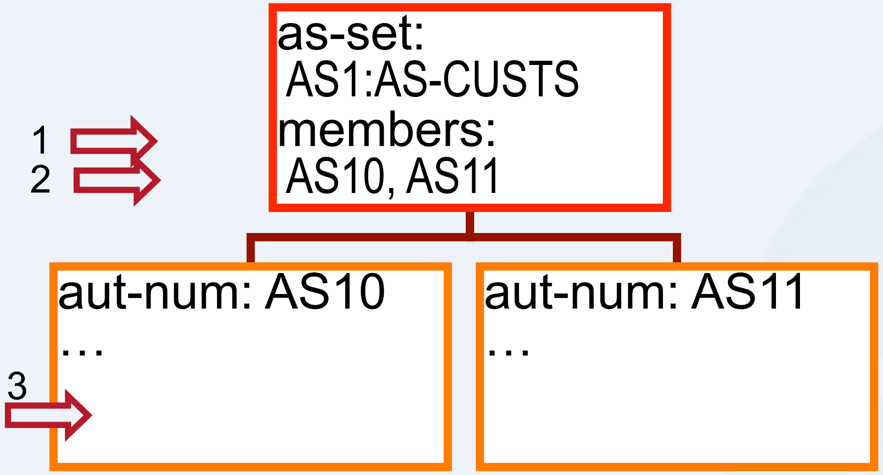
- Route-set attribute starts with “rs-”

# ‘Set-’ objects and their members

- Two ways of referencing members

## members

*- members specified in the ‘set-’ object*



as-set:
 AS1:AS-CUSTS
members:
 AS10, AS11

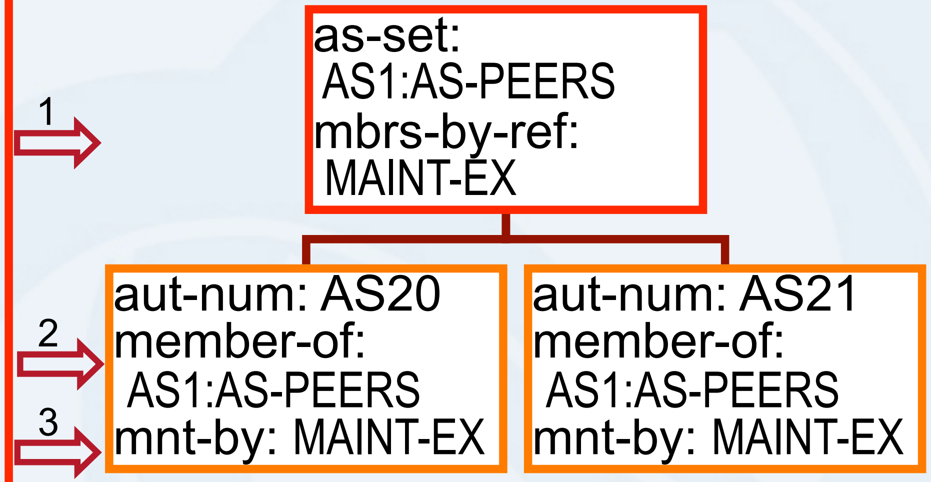
aut-num: AS10
…

aut-num: AS11
…

1. ‘members’ specifies members of the set
2. Members added in the ‘set-’ object
3. No need to modify the member object when adding members

## mbrs–by–ref

*- ‘set’ specified in the member objects*



as-set:
 AS1:AS-PEERS
mbrs-by-ref:
 MAINT-EX

aut-num: AS20
member-of:
 AS1:AS-PEERS
mnt-by: MAINT-EX

aut-num: AS21
member-of:
 AS1:AS-PEERS
mnt-by: MAINT-EX

1. ‘mbrs-by-ref’ specifies the maintainer of the members.
2. Members reference the ‘set-’ object in the ‘member-of’ attribute
3. Members are maintained by the maintainer specified in the ‘set-’

# Useful IRR queries

- *What routes are originating from my AS?*
  - whois -i origin <ASN>
    - route objects with matching origin
- *What routers does my AS operate?*
  - whois -i local-as <ASN>
    - inet-rtr objects with a matching local-as
- *What objects are protecting "route space" with my maintainer?*
  - whois -i mnt-routes <mntner>
    - aut-num, inetnum & route objects with matching mnt-routes

*(always specify host. e.g. 'whois –h whois.apnic.net')*

# Useful IRR queries (cont'd)

- *What '-set objects' are the objects protected by this maintainer a member of?*
  - whois –i  mbrs-by-ref <mntner>
    - set objects (as-set, route-set and rtr-set) with matching mbrs-by-ref
- *What other objects are members of this '-set object'?*
  - whois -i  member-of <set name>
    - Objects with a matching member-of
      - provided the membership claim is validated by the mbrs-by-ref of the set.

# Address prefix range operator

| Operator | Meanings |
|---|---|
| ^- | Exclusive more specifics of the address prefix:<br>E.g. 128.9.0.0/16^- contains all more specifics of 128.9.0.0/16 excluding 128.9.0.0/16 |
| ^+ | Inclusive more specific of the address prefix:<br>E.g. 5.0.0.0/8^+ contains all more specifics of 5.0.0.0/8 including 5.0.0.0/8 |

# Address prefix operator (cont.)

| Operator | Meanings |
|---|---|
| ^n | n = integer, stands for all the length “n” specifics of the address prefix:<br>E.g. 30.0.0.0/8^16 contains all the more specifics of 30.0.0.0/8 which are length of 16 such as 30.9.0.0/16 |
| ^n-m | m = integer, stands for all the length “n” to length “m” specifics of the address prefix:<br>E.g. 30.0.0.0/8^24-32 contains all the more specifics of 30.0.0.0/8 which are length of 24 to 32 such as 30.9.9.96/28 |

# AS-path regular expressions

- Regular expressions
  - A context-independent syntax that can represent a wide variety of character sets and character set orderings
  - These character sets are interpreted according to the current The Open Group Base Specifications (IEEE)
- Can be used as a policy filter by enclosing the expression in “<“ and “>”.

# AS-path regular expression

| Operator | Meanings |
|---|---|
| <AS3> | Route whose AS-path contains AS3 |
| <^AS1> | Routes whose AS-path starts with AS1 |
| <AS2$> | Routes whose AS-path end with AS2 |
| <^AS1 AS2 AS3$> | Routes whose AS-path is exactly “1 2 3” |
| <^AS1 . * AS2$> | AS-path starts with AS1 and ends in AS2 with any number ASN in between |
| <^AS3+$> | AS-path starts with AS3 and ends in AS3 and<br>AS3 is the first member of the path and AS3 occurs one or more times in the path and no other AS can be present in the path after AS3 |

# AS-path regular expression (cont.)

| Operator | Meanings |
|---|---|
| <AS3\|AS4> | Routes whose AS-path is with AS3 or AS4 |
| <AS3 AS4> | Routes whose AS-path with AS3 followed by AS4 |

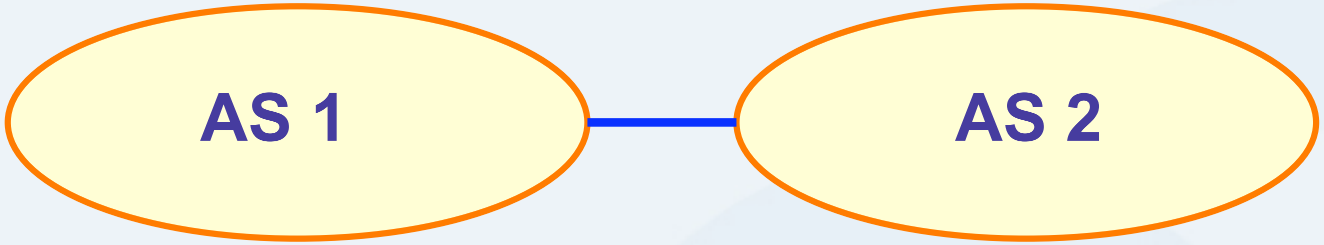
# Using RPSL in practice

# Overview

- Review examples of routing policies expression
  - Peering policies
  - Filtering policies
  - Backup connections

# Representation of routing policy

Basic concept

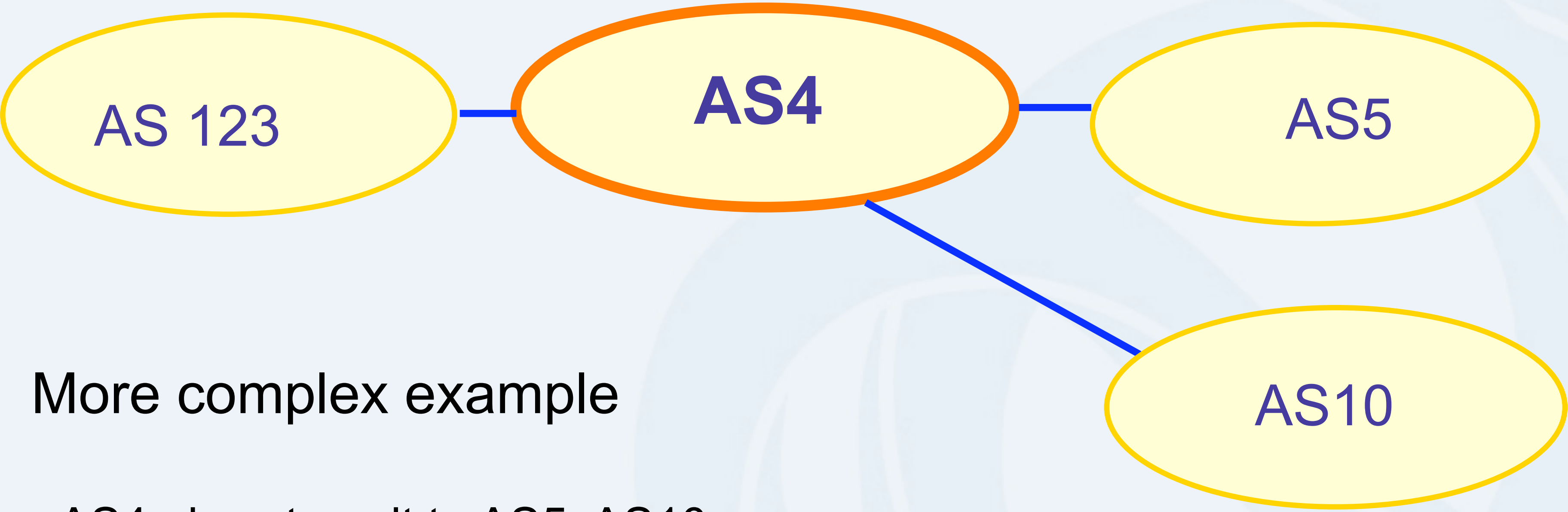


*“action pref” - the lower the value, the preferred the route*

```
aut-num: AS1
…
import:  from AS2
         action pref=100;
         accept AS2
export:  to AS2 announce AS1
```

```
aut-num: AS2
…
import:  from AS1
         action pref=100;
         accept AS1
export:  to AS1 announce AS2
```

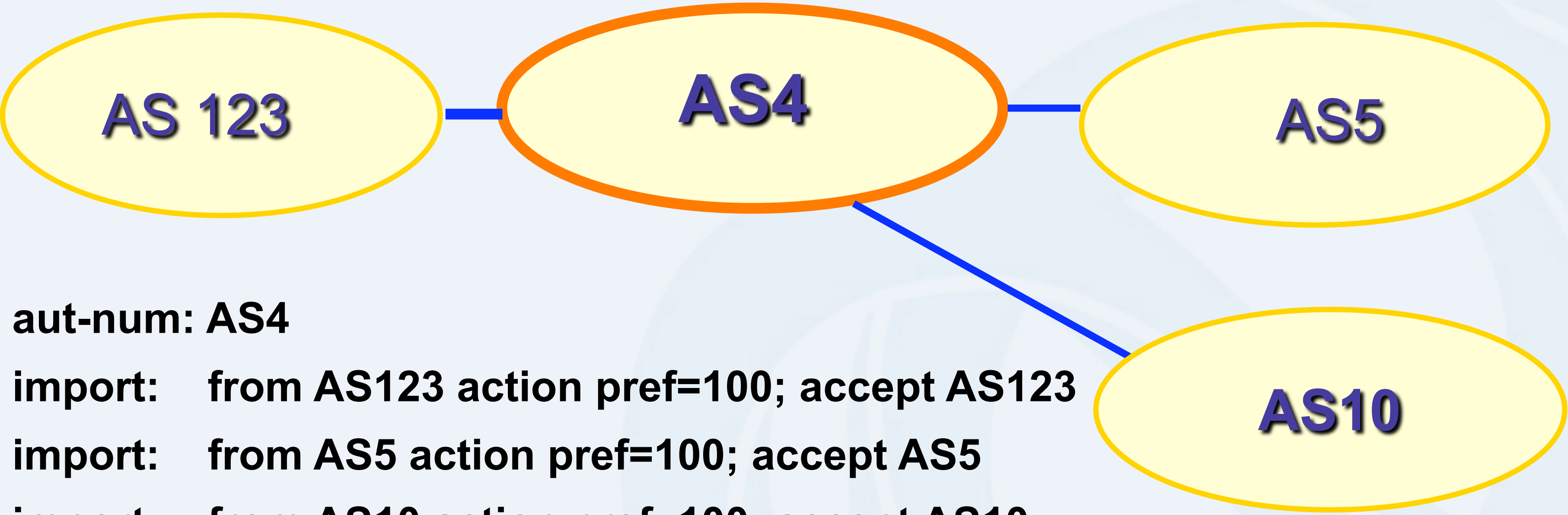
# Representation of routing policy



More complex example

- AS4 gives transit to AS5, AS10
- AS4 gives local routes to AS123

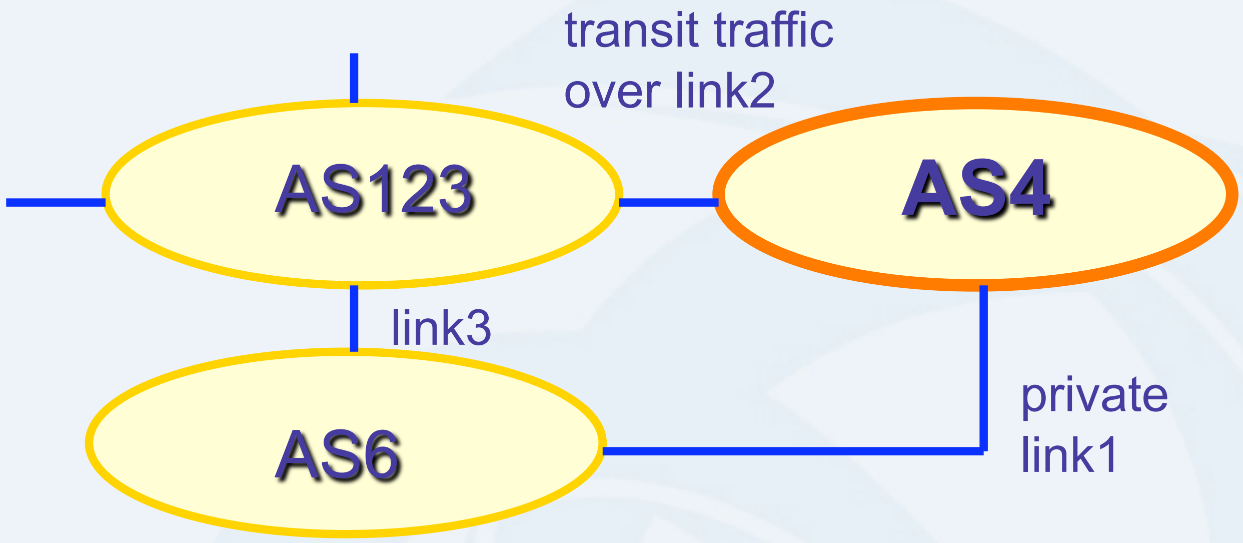
# Representation of routing policy



```
aut-num: AS4
import:    from AS123 action pref=100; accept AS123
import:    from AS5 action pref=100; accept AS5
import:    from AS10 action pref=100; accept AS10
export:    to AS123  announce  AS4
export:    to AS5 announce AS4 AS10
export:    to AS10 announce AS4 AS5
```

*Not a path*

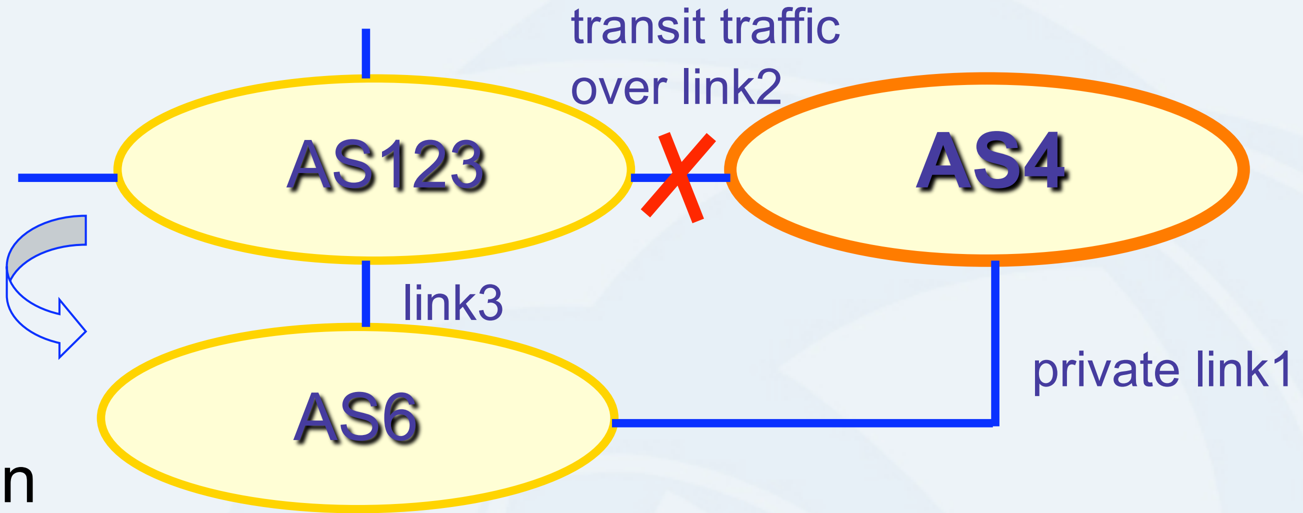
# Representation of routing policy



More complex example

- AS4 and AS6 private link1
- AS4 and AS123 main transit link2
- backup all traffic over link1 and link3 in event of link2 failure

# Representation of routing policy



AS representation

**aut-num: AS4**

**import: from AS123 action pref=100; accept ANY**

*full routing received*

**import: from AS6 action pref=50; accept AS6**

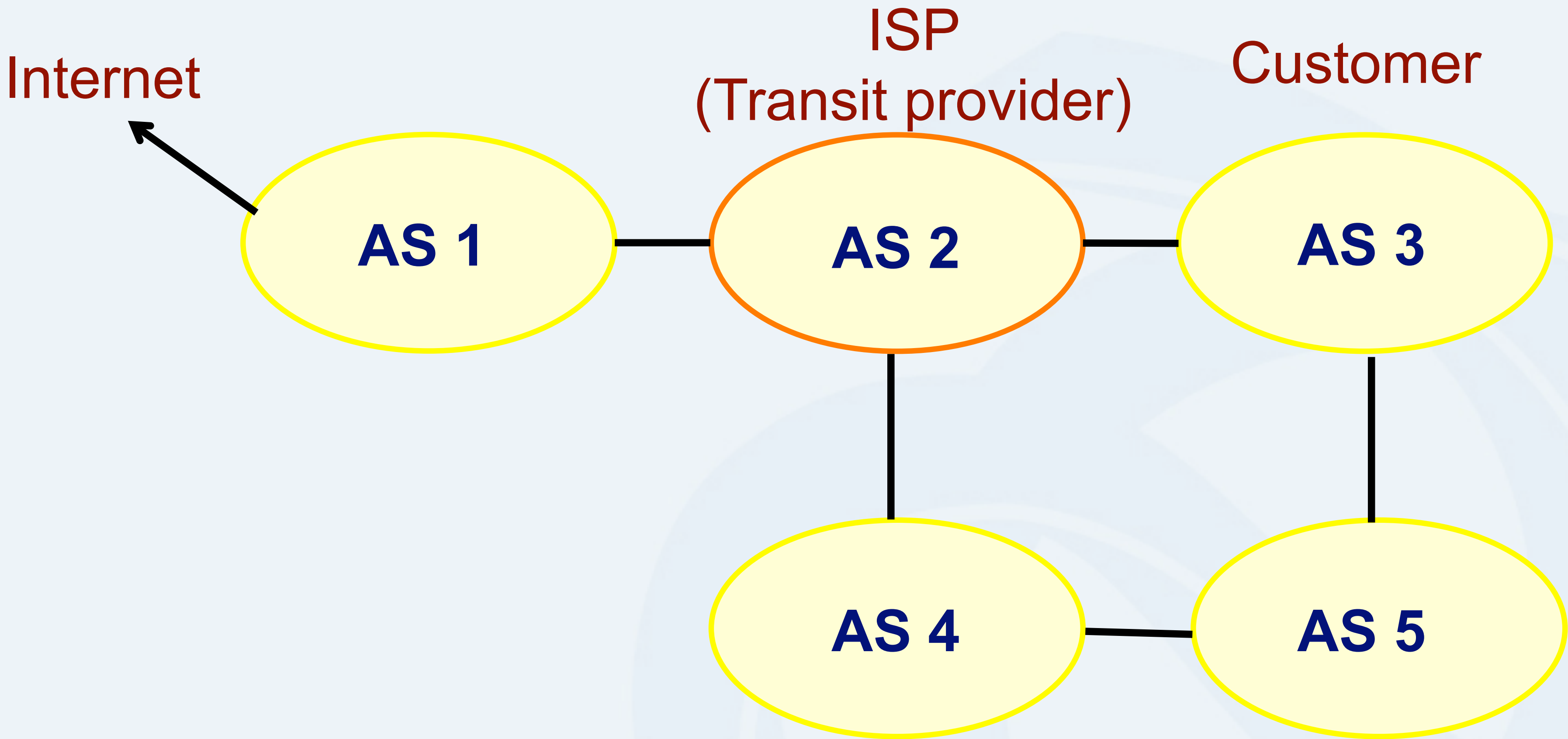
**import: from AS6 action pref=200; accept ANY**

**export: to AS6 announce AS4**

**export: to AS123 announce AS4**

*higher cost for backup route*

# Common peering policies



- Peering policies of an AS
  - Registered in an aut-num object

# Common peering policies

- Policy for AS3 in the AS2 aut-num object

```
aut-num:     AS2
as-name:     SAMPLE-NET
dsescr:      Sample AS
import:      from AS1 accept ANY
import:      from AS3 accept <^AS3+$>
export:      to AS3 announce ANY
export:      to AS1 announce AS2 AS3
admin-c:     CW89-AP
tech-c:      CW89-AP
mtn-by:      MAINT-SAMPLE-AP
changed:     sample@sample.net
```

# ISP customer – transit provider policies

- Policy for AS3 and AS4 in the AS2 aut-num object

```
aut-num:      AS2
import:       from AS1 accept ANY
import:       from AS3 accept <^AS3+$>
import:       from AS4 accept <^AS4+$>
export:       to AS3 announce ANY
export:       to AS4 announce ANY
export:       to AS1 announce AS2 AS3 AS4
```

# AS-set object

- Describe the customers of AS2

```
as-set:        AS2:AS-CUSTOMERS
members:       AS3 AS4
changed:       sample@sample.net
source:        APNIC
```

# Aut-num object referring as-set object

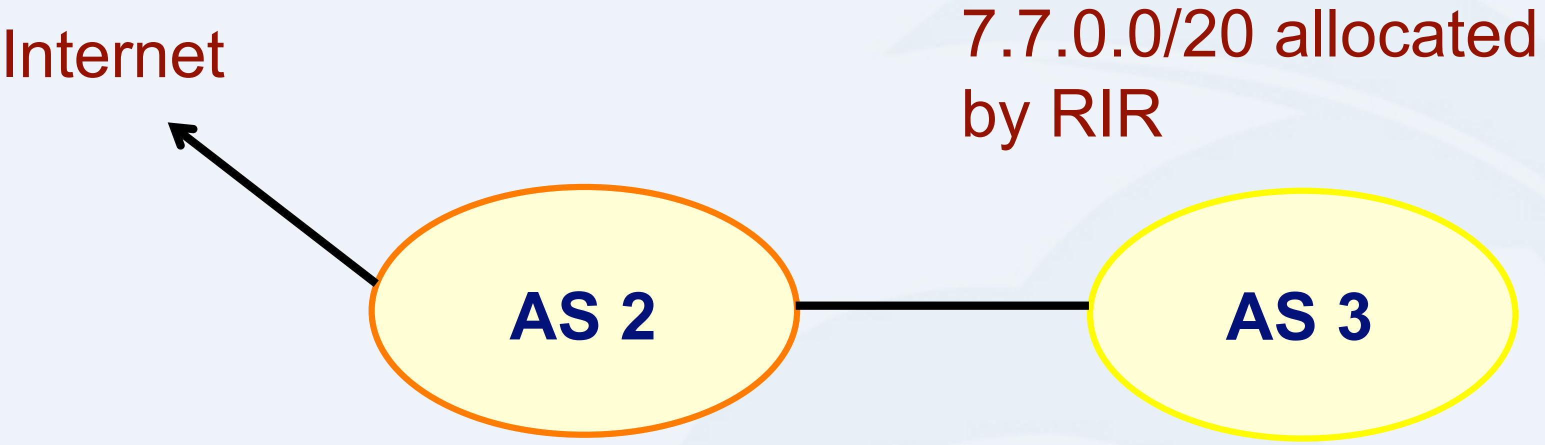
```
aut-num:        AS2
import:         from AS1 accept ANY
import:         from AS2:AS-CUSTOMERS accept
                <^AS2:AS-CUSTOMERS+$>
export:         to AS2:AS-CUSTOMERS announce ANY
export:         to AS1 announce AS2 AS2:AS-
                CUSTOMERS
```

```
aut-num:        AS1
import:         from AS2 accept <^AS2+AS2:AS-
               CUSTOMERS+$>
export:         ………
```

# Express filtering policy

- To limit the routes one accepts from a peer
  - To prevent the improper use of unassigned address space
  - To prevent malicious use of another organisation’s address space

# Filtering policy



AS3 wants to announce part or all of 7.7.0.0/20 the global Internet.

AS2 wants to be certain that it only accepts announcements from AS3 for address space that has been properly allocated to AS3.

# Aut-num object with filtering policy

```
aut-num:        AS2
import:         from AS3 accept { 7.7.0.0/20^20-24 }
…….
```

For an ISP with a growing or changing customer base, this mechanism will not scale well.

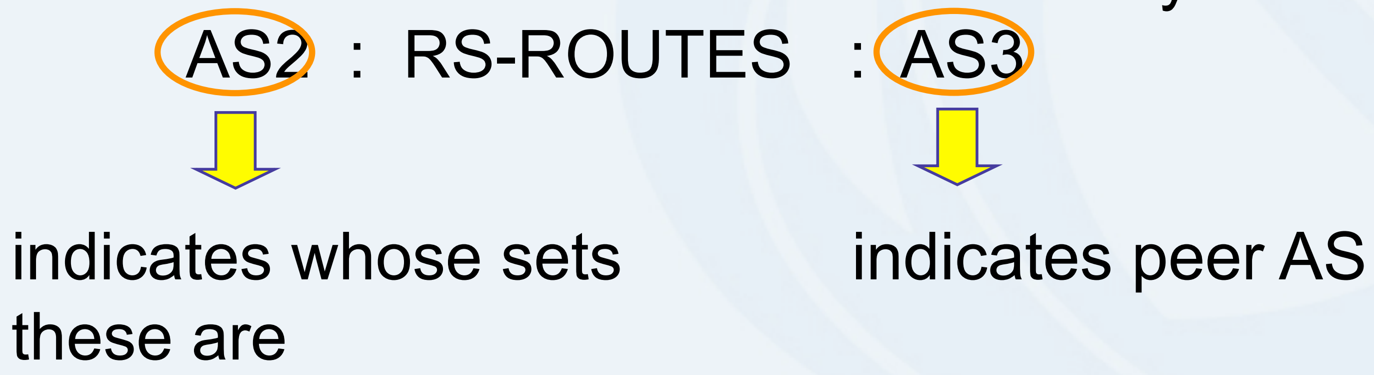
Route-set object can be used.

# Route-set

```
route-set:    AS2:RS-ROUTES:AS3
members:      7.7.0.0/20^20-24
changed:      sample@sample.net
source:       APNIC
```

Specifies the set of routes that will be accepted from a given customer

Set names are constructed hierarchically:



AS2 : RS-ROUTES : AS3

indicates whose sets these are

indicates peer AS

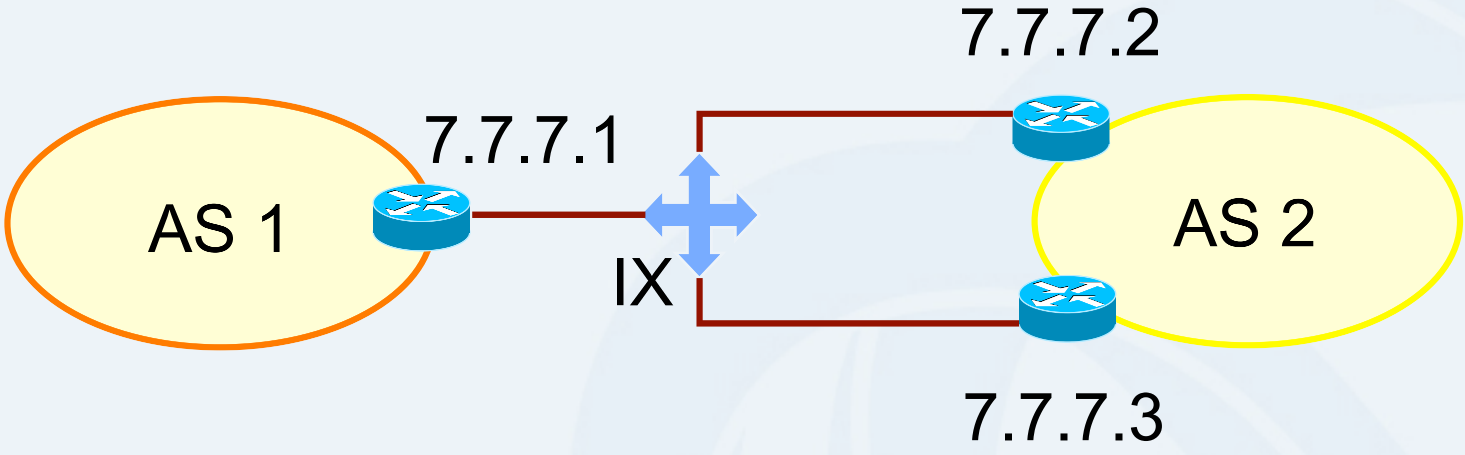
# Filter configuration using route-set – AS2

```
import:    from AS1 accept ANY
import:    from AS3 accept AS2:RS-ROUTES:AS3
import:    from AS4 accept AS2:RS-ROUTES:AS4
export:    to AS2:AS-CUSTOMERS announce ANY
export:    to AS1 announce AS2 AS2:AS-CUSTOMERS
```

RPSL allows the peer’s AS number to be replaced by the keyword PeerAS

```
import:  from AS2:AS-CUSTOMERS accept
         AS2:RS-ROUTES:PeerAS
```

# Including interfaces in peering definitions: AS1



How to define AS1’s routing policy by specifying its boundary router?

# Including interfaces in peering definitions: AS1 (cont.)

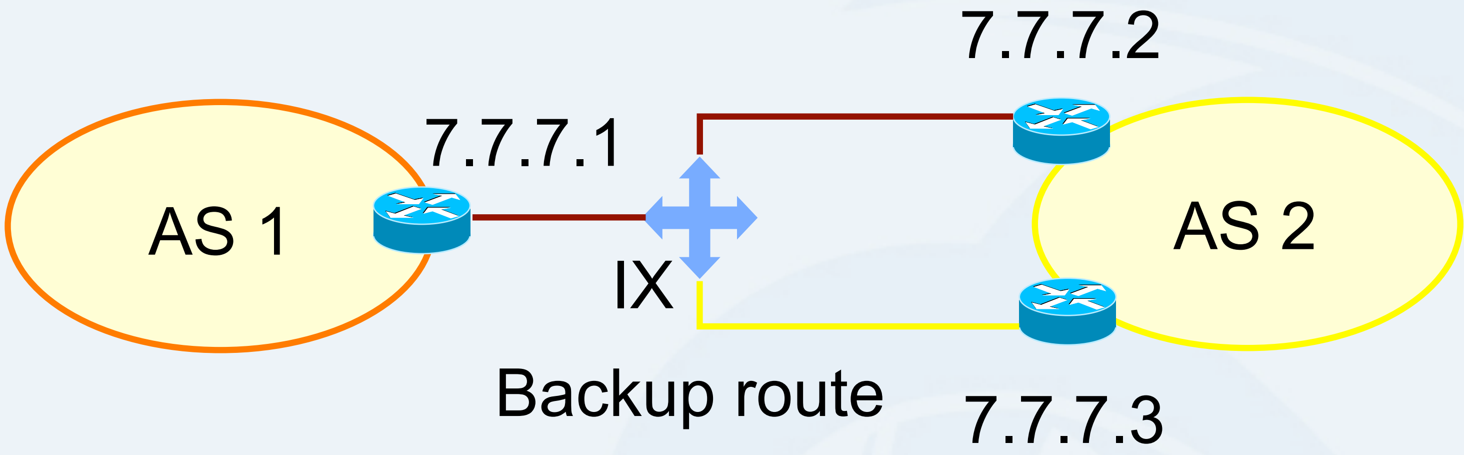
```
aut-num:    AS1
import:     from AS2 at 7.7.7.1 accept <^AS2+$>
```

AS1 may want to choose to accept:

- only those announcements from router 7.7.7.2
- discard those announcements from router 7.7.7.3

```
aut-num: AS1
import:   from AS2 7.7.7.2 at 7.7.7.1 accept <^AS2+$>
```

# Describing simple backup connections: AS1



How to define AS1’s routing policy of its backup route?

⇨ Use preference

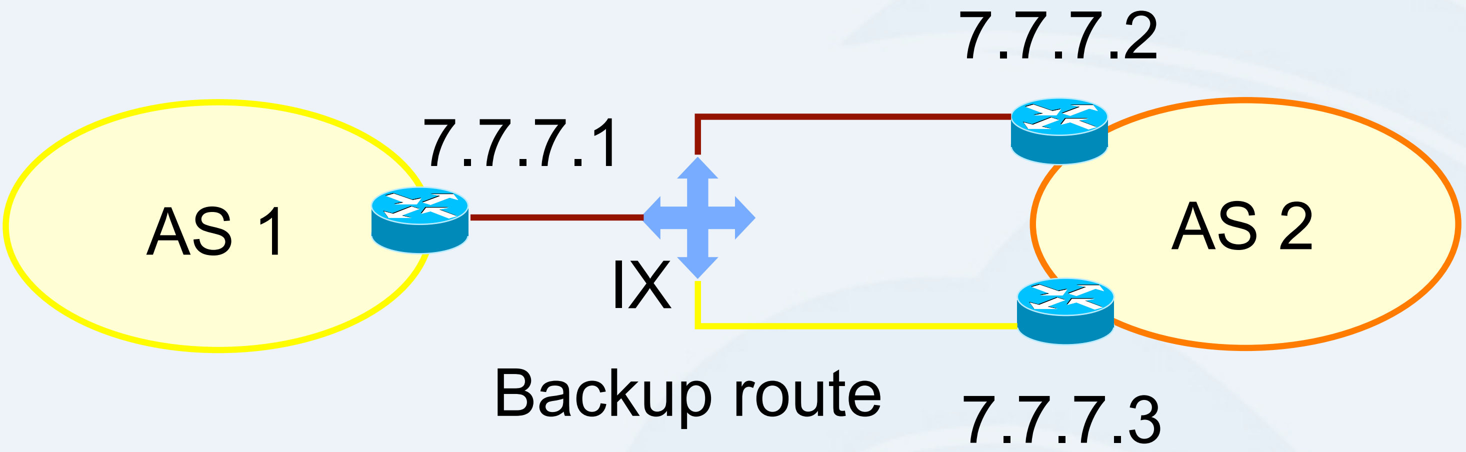
# Describing simple backup connections: AS1 (cont.)

```
aut-num:   AS1
import:    from AS2 7.7.7.2 at 7.7.7.1 action pref=10;
           from AS2 7.7.7.3 at 7.7.7.1 action pref=20;
           accept <^AS2+$>
```

Use of pref

- pref is opposite to local-pref
- Smaller values are preferred over larger values

# Describing simple backup connections: AS2



How to define AS2's routing policy of AS1's backup route?

⇨ multi exit discriminator metric (med) can be used

# Describing simple backup connections: AS2 (cont.)

```
aut-num:   AS2
export:    to AS1 7.7.7.1 at 7.7.7.2 action med=10;
           to AS1 7.7.7.1 at 7.7.7.3 action med=20;
           announce <^AS2+$>
```

Use of med

- Suitable for load balancing including backups

# Summary

# What we discussed

- APNIC Whois database recap
- What is IRR and Why use it
- How to use the Routing Registry
- Benefit of using IRR
- Using RPSL in practice

# Usage: preliminary work for your AS

- Enter in the APNIC RR
  - Or in your own RR database
- Create person and mntner objects
- Describe policy in your aut-num object
- Identify IP prefixes associated with your AS
  - Create route objects in the database
  - Create route-set objects
- Crete various as-set objects, to group different categories of neighbours
- Create RtConfig template files
- Run RtConfig periodically to produce (parts of) router configuration file

# Questions ?

Thank you! ☺