Asia Pacific Network Information Centre

APNIC

APNIC
Internet Routing Registry

Kuala Lumpur, Malaysia,

February 2004

APNIC 17 Open Policy Meeting

Introduction

- Presenters
  - Andy Linton
    Citylink, New Zealand
  - Miwa Fujii
    Training Officer

<training@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
  - A chance for practical exercise
- NOT to:
  - Teach basic routing
  - Explain Internet resource policy and procedures
  - Provide advise on network configuration

Assumptions

- The audience
  - Knowledgeable about BGP routing
  - Familiar with basic APNIC database operations
  - Curious about Internet Routing Registry usage (IRR)
  - But not yet familiar with Routing Policy Specification Language (RPSL) and IRR

Schedule and house keeping matter

1. IRR overview
2. RPSL
3. Using RPSL in practice
4. Case studies
5. Practical usage of IRR

House keeping matter

Evaluation Form

Asia Pacific Network Information Centre

APNIC

Internet Routing Registry

Overview

Overview

- APNIC database recap
- What is IRR?
- Why use an IRR?
- APNIC database and the IRR
- Using the Routing Registry
  - Overview of IRRToolSet
- 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

Attributes	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

Please see “APNIC Whois Database queries” card for more details in your folder.

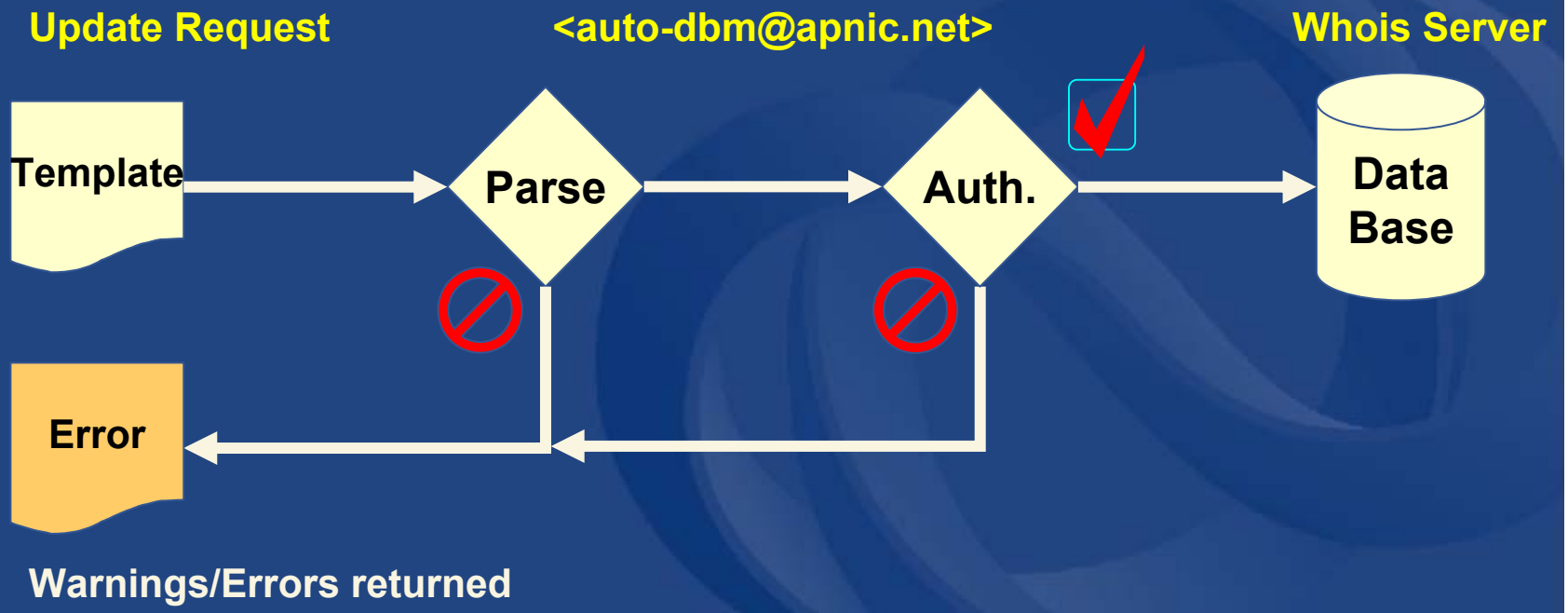
Database update process

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

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

Database update process

- Email requests to <auto-dbm@apnic.net>
- Each request contains an object template



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
  - ~~<none>~~
    - Deprecated
  - Crypt-PW
    - Crypt (Unix) password encryption
    - Use web page to create your maintainer
  - PGP – GNUPG
    - Strong authentication
    - Requires PGP keys
  - MD5
    - Available

New policy as of Dec-03

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
  - The 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
  - ASes 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

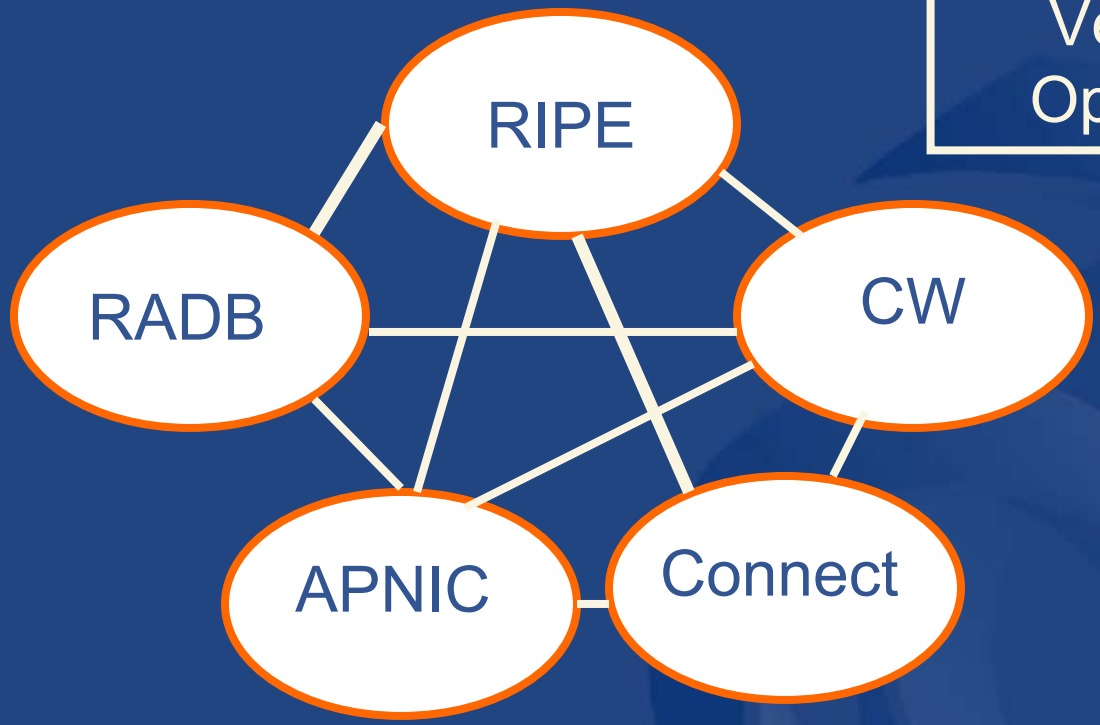
Routing registry objects

- 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
  - Established in 1995
- 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

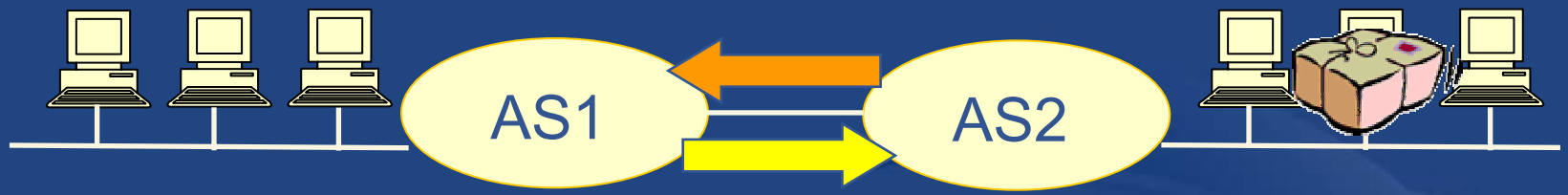
What is a Routing Registry?



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

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

Representation of routing policy



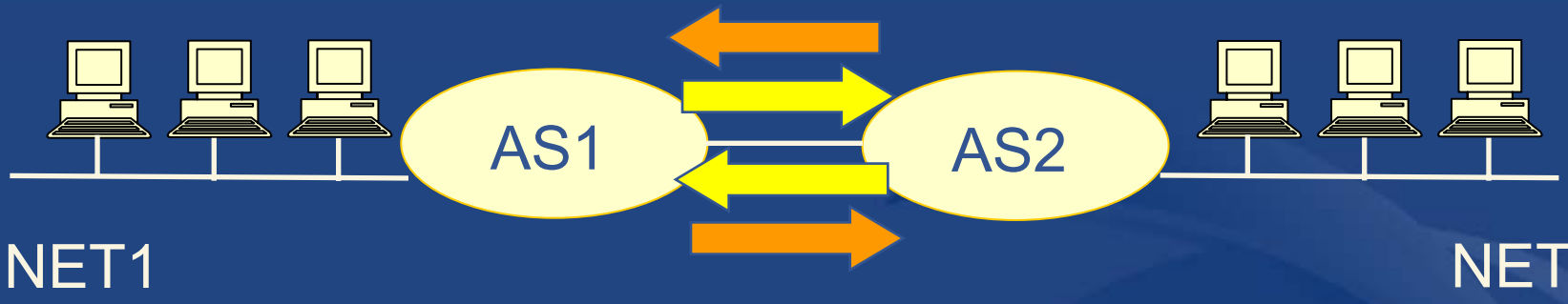
In order for traffic to flow from NET2 to NET1 between AS1 and AS2:

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, upsteam
  - 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

Network planning

- Network planning
  - Simulation
    - Changes in polices can be simulated first by changing the registry but not the routers
      - To understand effects of policy changes to the existing networks
      - To make better network planning
      - To make it easier to adjust policies to maximise the performance of the network
  - Route filtering
    - Peering networks
    - A provider and its customer

Router configuration and network troubleshooting

- Router configuration
  - By using IRRToolSet
    - ftp.ripe.net/tools/IRRToolSet
    - Abstract 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

Asia Pacific Network Information Centre

APNIC

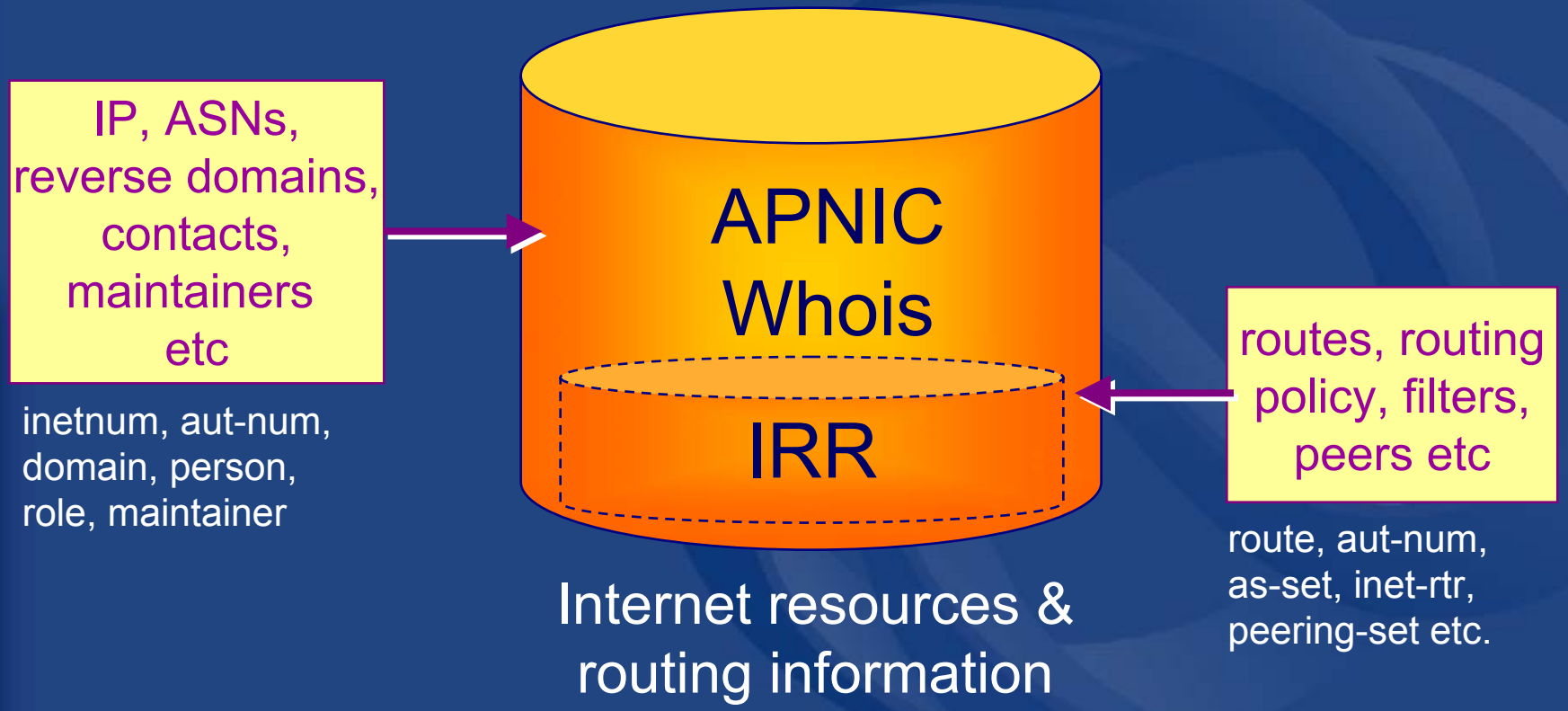
APNIC database and the IRR

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



RPSL

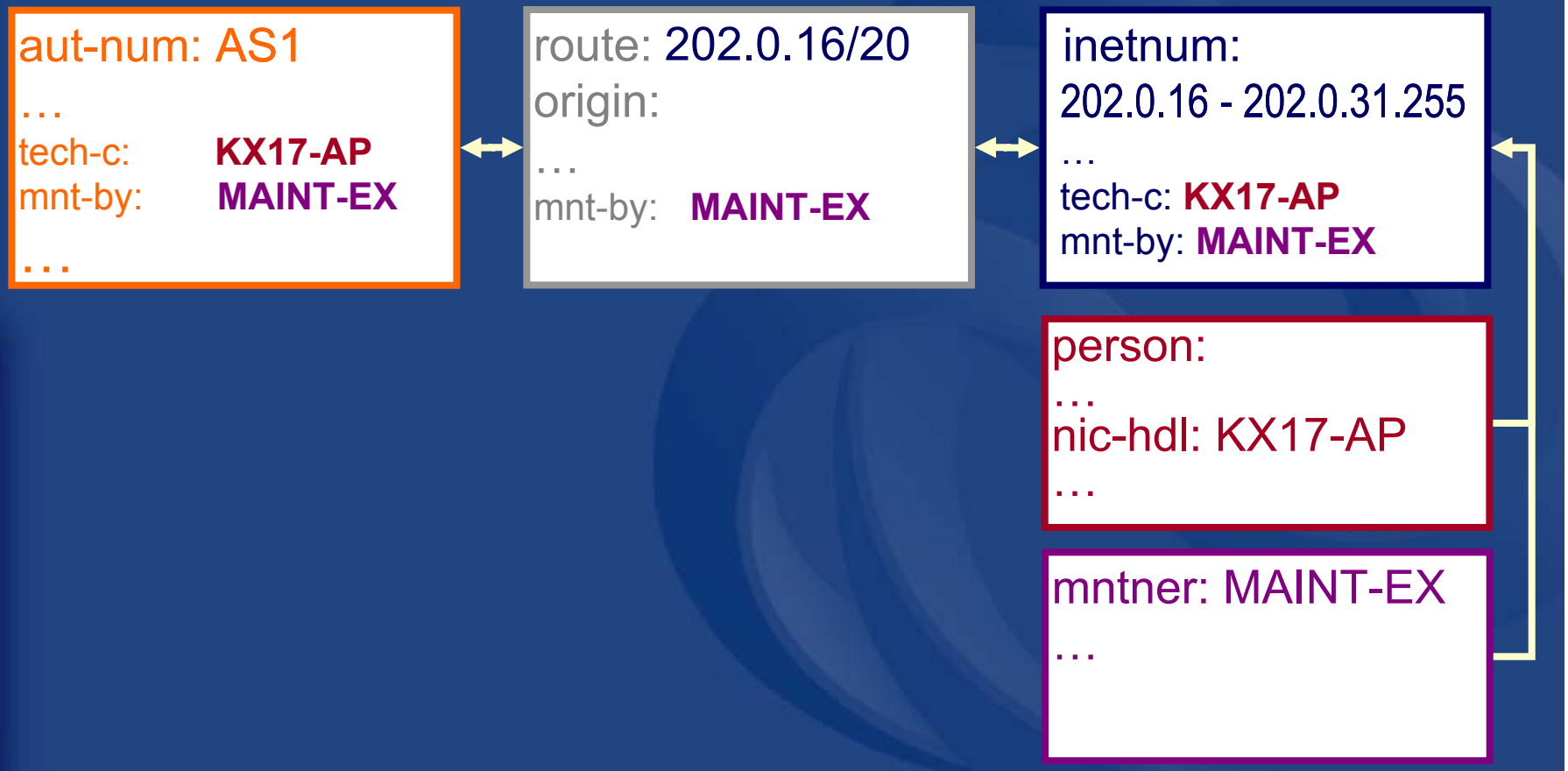
- Routing Policy Specification Language
  - Object oriented language
    - Based on RIPE-181
  - Structured whois objects
- Higher level of abstraction than access lists
- Describes things interesting to routing policy:
  - Routes, AS Numbers …
  - Relationships between BGP peers
  - Management responsibility
- Relevant RFCs
  - Routing Policy Specification Language
  - Routing Policy System Security
  - Using RPSL in Practice

IRR objects

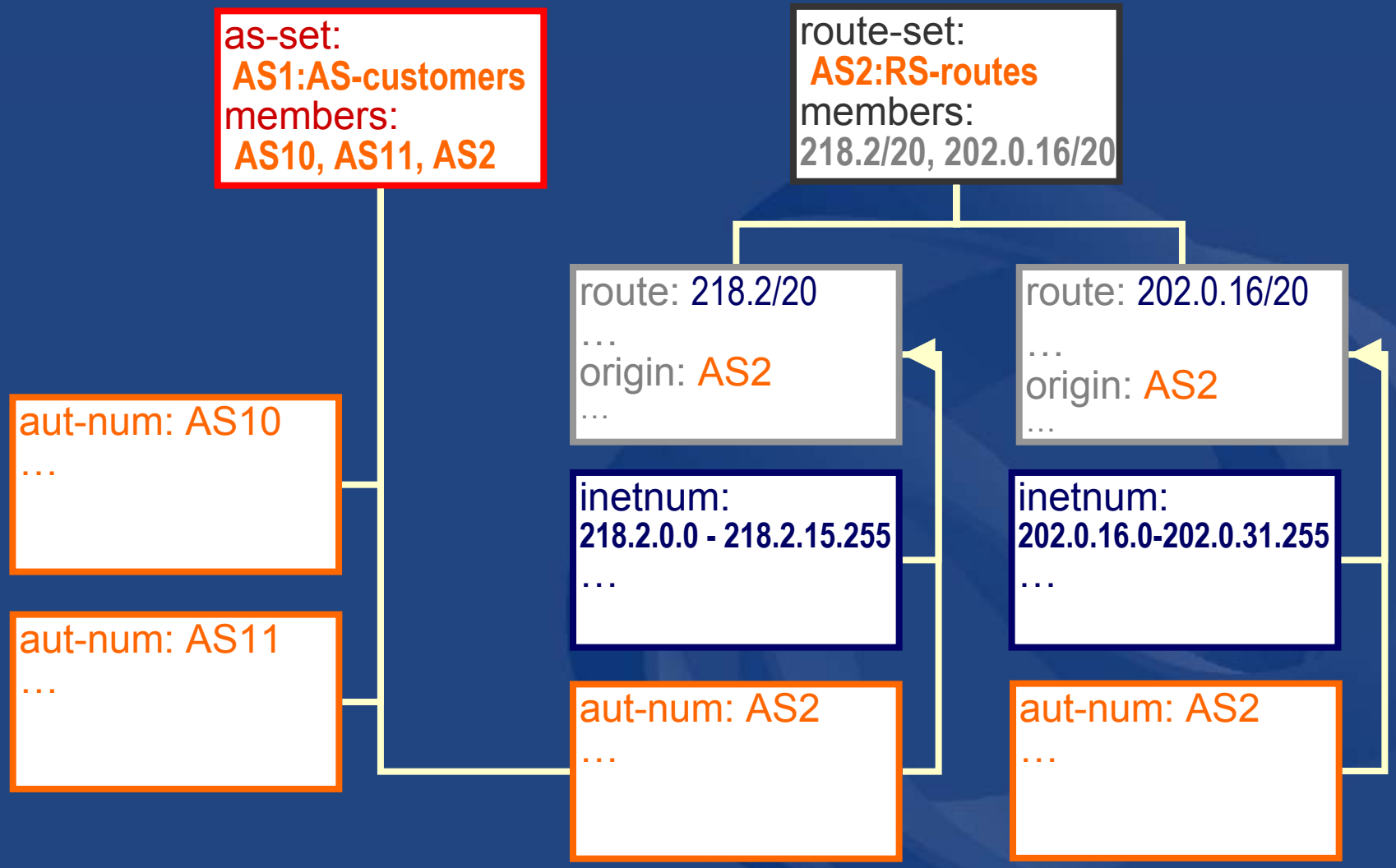
- route
  - Specifies interAS routes
- aut-num
  - Represents an AS. Used to describe external routing policy
- inet-rtr
  - Represents a router
- peering-set
  - Defines a set of peerings
- route-set
  - Defines a set of routes
- as-set
  - Defines a set of **aut-num** objects
- rtr-set
  - Defines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



Inter-related IRR objects



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>`

In:

`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
```

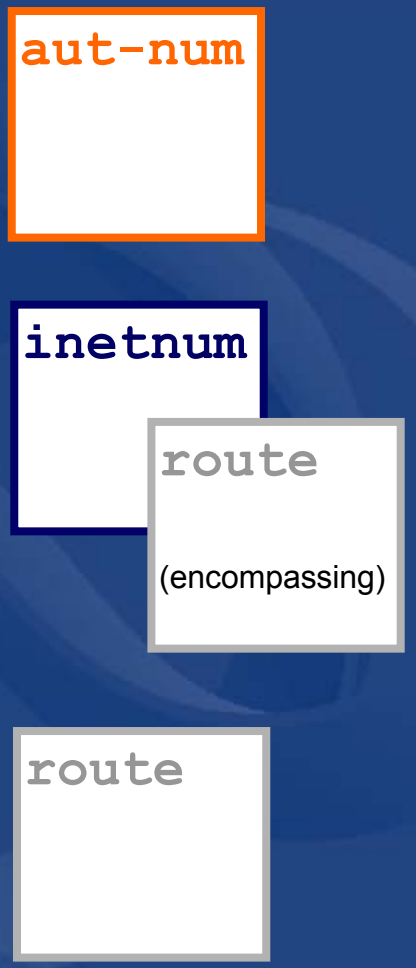
This object can only be modified by APNIC

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

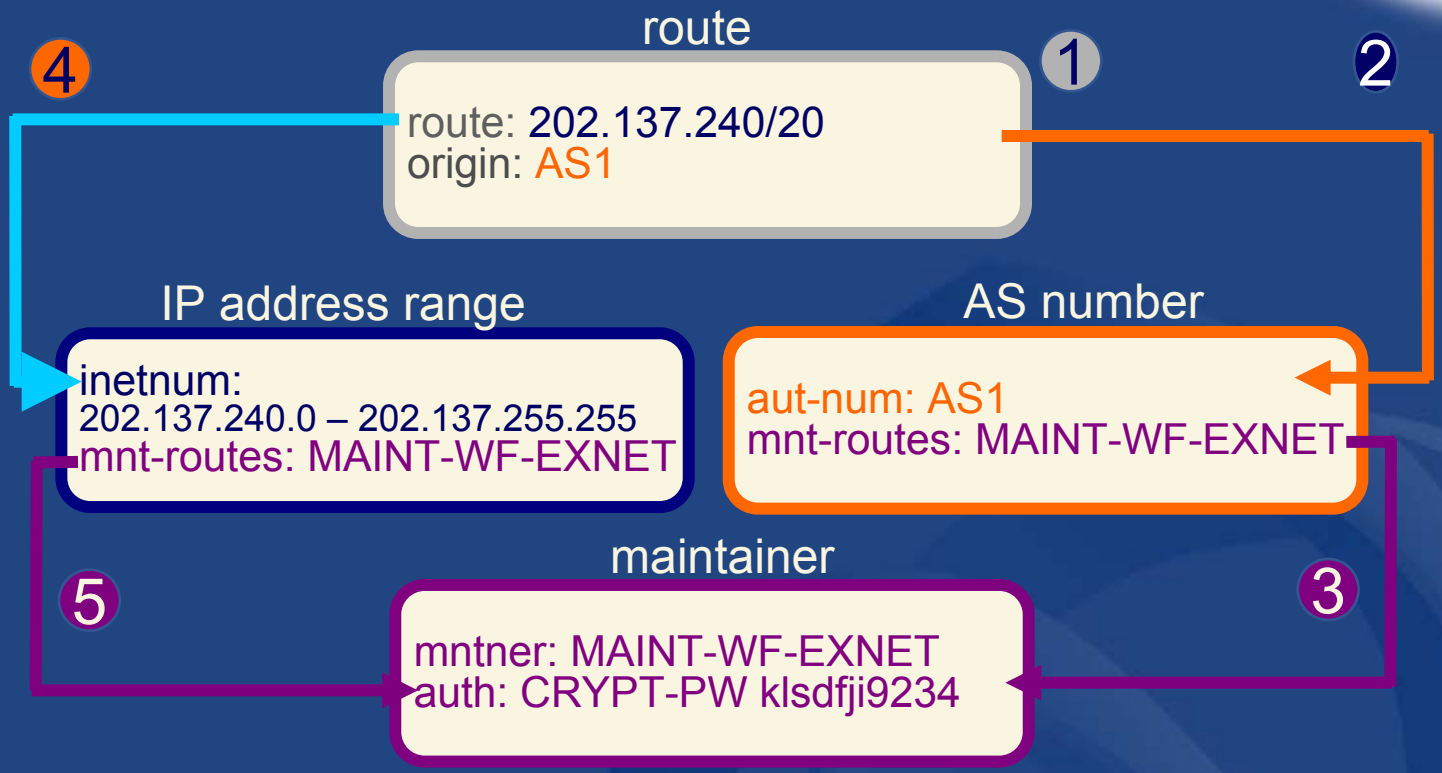
Creation of route objects matching/within this range has to pass the authentication of MAINT-SPARKYNET-WF

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.

Using the Routing Registry

Overview of the IRRToolSet

IRRToolSet

- Set of tools developed for using the Internet Routing Registry
  - Started as RAToolSet

- Now maintained by RIPE NCC:
  - http://www.ripe.net/db/irrtoolset/
  - Download: ftp://ftp.ripe.net/tools/IRRToolSet/
    - Installation needs: lex, yacc and C++ compiler

Use of RPSL - RtConfig

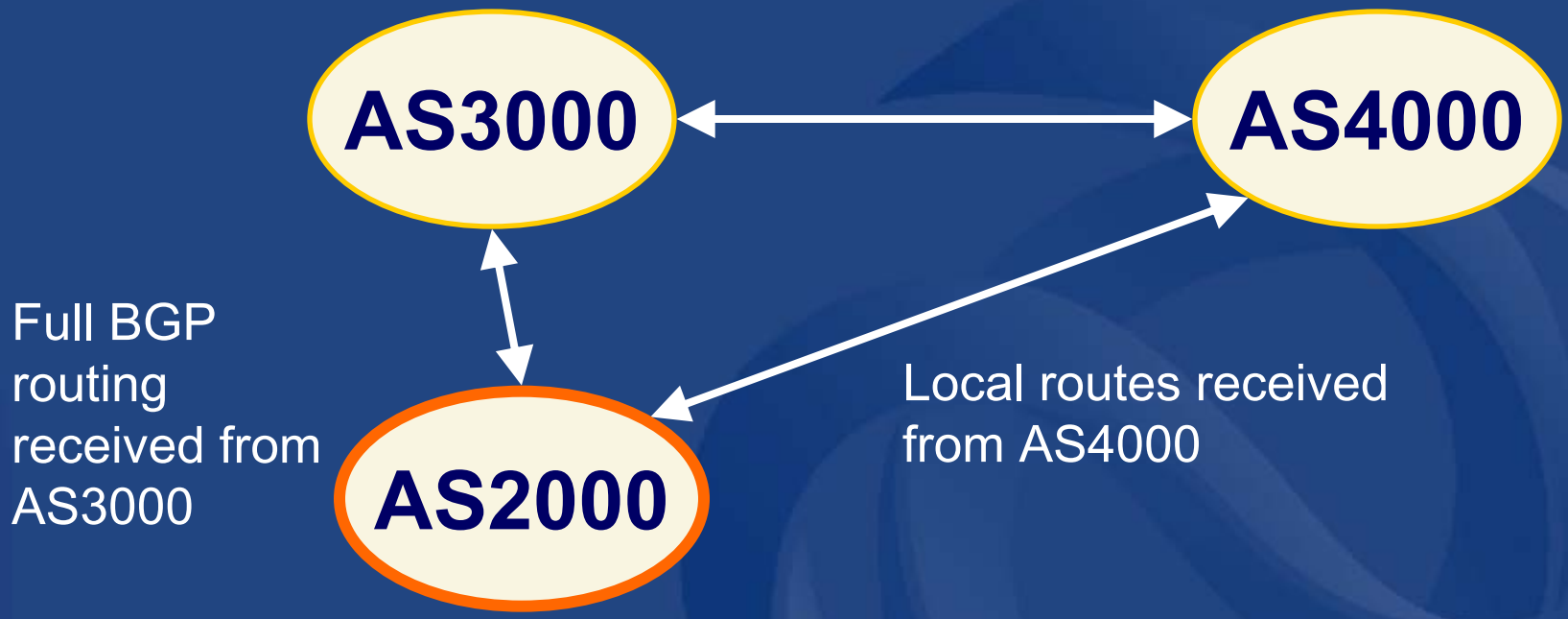
- RtConfig v4
  - part of IRRToolSet
- Reads policy from IRR (aut-num, route & -set objects) and generates router configuration
  - vendor specific:
    - Cisco, Bay's BCC, Juniper's Junos and Gated/RSd
  - Creates route-map and AS path filters
  - Can also create ingress / egress filters
    - (documentation says Cisco only)

Why use IRR and RtConfig?

- Benefits of RtConfig
  - Avoid filter errors (typos)
  - Expertise encoded in the tools that generate the policy rather than engineer configuring peering session
  - Filters consistent with documented policy
    - (need to get policy correct though)
  - Engineers don't need to understand filter rules
    - it just works :-)

Using RtConfig - Case scenario

Not fully multi-homing



10.20.0.0/24 *(range received from upstream)*

10.187.65.0/24 *(portable address range)*

Using RtConfig – IRR objects

```
aut-num: AS2000
import:  from AS3000 accept ANY
export:  to AS3000 announce AS2000
import:  from AS4000 accept AS4000
export:  to AS4000 announce AS2000
[…]
```

full BPG routing

local routes

```
route: 10.20.0.0/24
origin: AS2000
[…]
```

```
route: 10.187.65.0/24
origin: AS2000
[…]
```

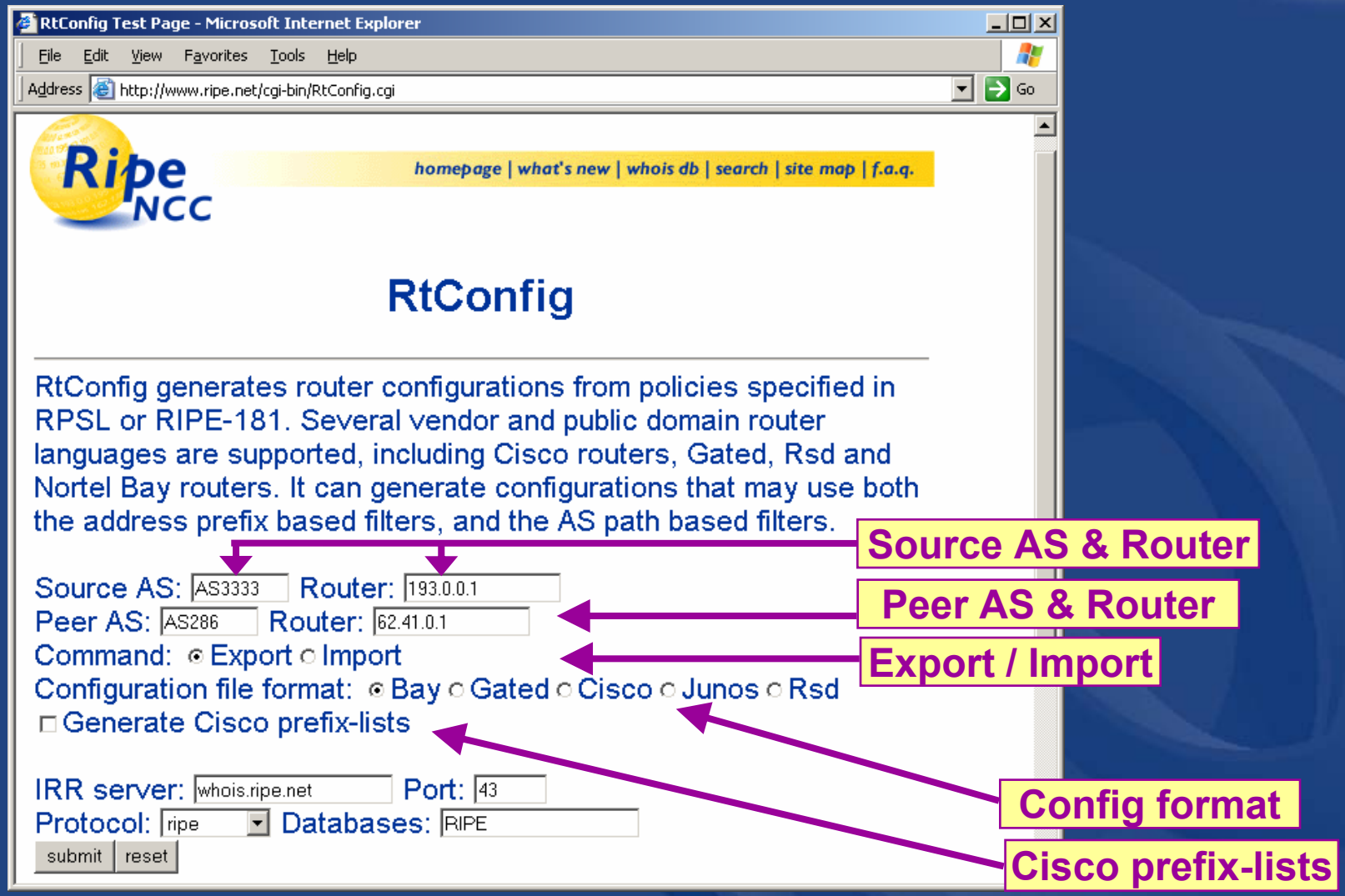
RtConfig commands

```
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.4 AS4000 10.4.192.2
!
```

RtConfig output (import)

```
route-map AS3000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-IMPORT in
!
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-IMPORT in
```

RtConfig – web prototype



http://www.ripe.net/cgi-bin/RtConfig.cgi

RtConfig - web prototype

- Let's try the same command we tried in the previous slide:

@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2

- You can select different configuration format depends on a vendor
  - Select "Cicso"
  - Select "Bay"
  - Select "Junos"

The rest of the IRRToolSet

- peval
  - (Lightweight) policy evaluation tool
- prtraceroute
  - Prints the route packets take - including policy information (as registered in RR)
- aoe (aut-num object editor)
  - Displays the aut-num object for the specified AS
- roe
  - Creates the “route” object (based on BGP dump and routes in aut-num objects)

The rest of the IRRToolSet

- prpath
  - enumerates possible paths between two ASes
- CIDRAdvisor
  - suggests safe aggregates per AS
- *rpslcheck*
  - syntax checks objects for IRR

Benefit of using IRR

Using the Routing Registry

Costs

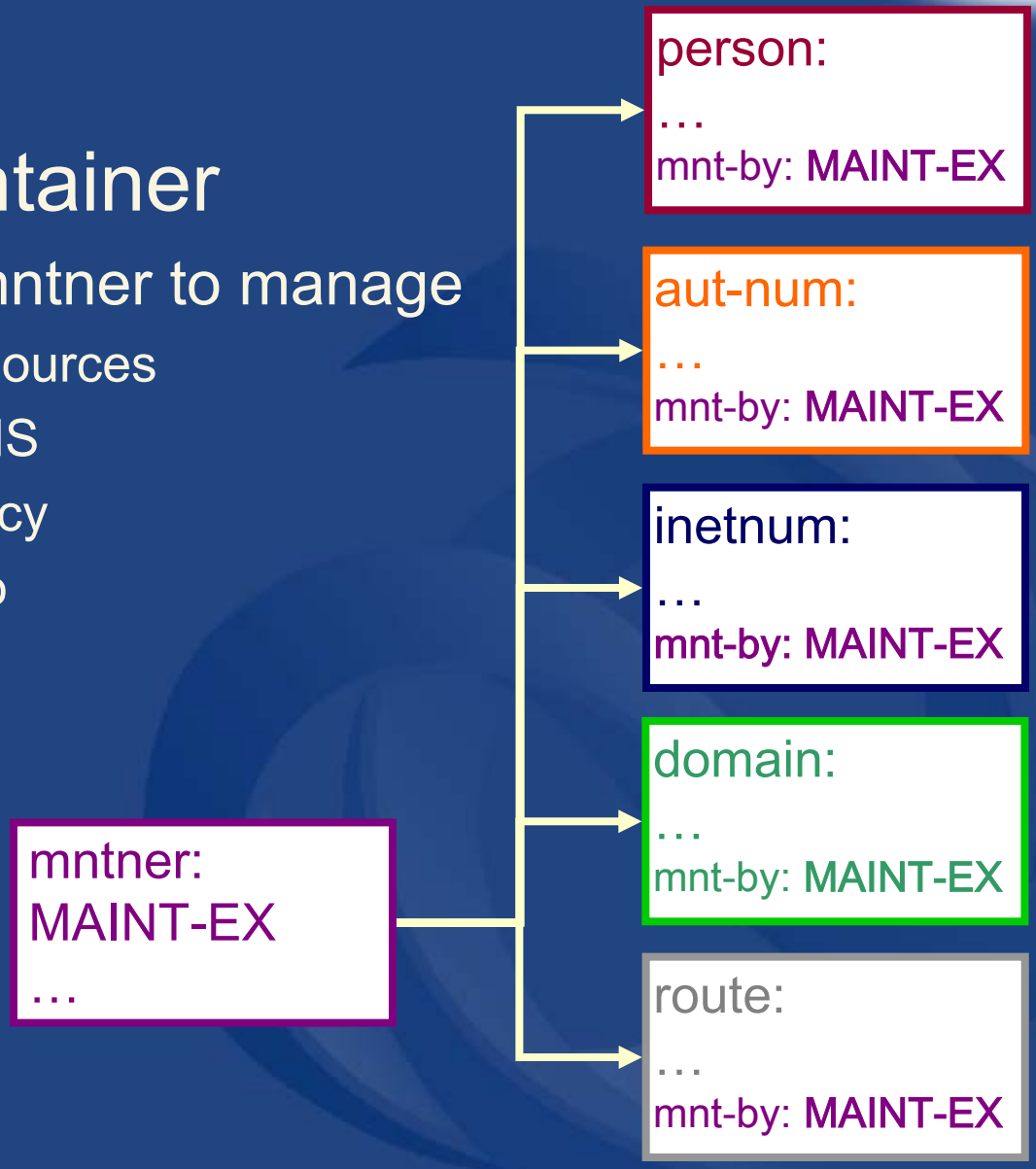
- Requires some initial planning
- Takes some time to define & register policy
- Need to maintain data in RR

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
...
```

APNIC RR service scope

- Routing Queries
  - Regular whois clients
  - APNIC whois web interface
  - Special purpose programs such as IRRToolSet
    - ftp://ftp.ripe.net/tools/IRRToolSet
- Routing Registration and Maintenance
  - Similar to registration of Internet resources

APNIC RR service scope

- Support
  - APNIC Helpdesk support

<helpdesk@apnic.net>

- Training
    - IRR workshop under development

- Mirroring
  - APNIC mirrors IRRs within Asia Pacific and major IRRs outside of the region.

Summary

- APNIC RR integrated in APNIC Whois DB
  - whois.apnic.net
  - <auto-dbm@apnic.net>
- IRR benefits
  - Facilitates network troubleshooting
  - Generation of router configuration
  - Provides global view of routing
- APNIC RR benefits
  - Single maintainer (& person obj) for all objects
  - APNIC asserts resources for a registered route
  - Part of the APNIC member service!

Questions ?

RPSL

Objects, syntax and semantics

Overview

- Review of some of RR objects
- Useful queries
- Address prefix range operator
- AS-path regular expression
- Action specification
- Seven rp-attributes
- 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# x	optional, multi-value

Route object

- Origin attribute
  - Specifies the AS that originates the route
  - The corresponding aut-num object should be registered in the database
- Members-of
  - identifies a set object that this object wants to be a member of.
  - must start with "rs-”
- Mnt-routes
  - references a maintainer object which is used in determining authorisation for the creation of route objects

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

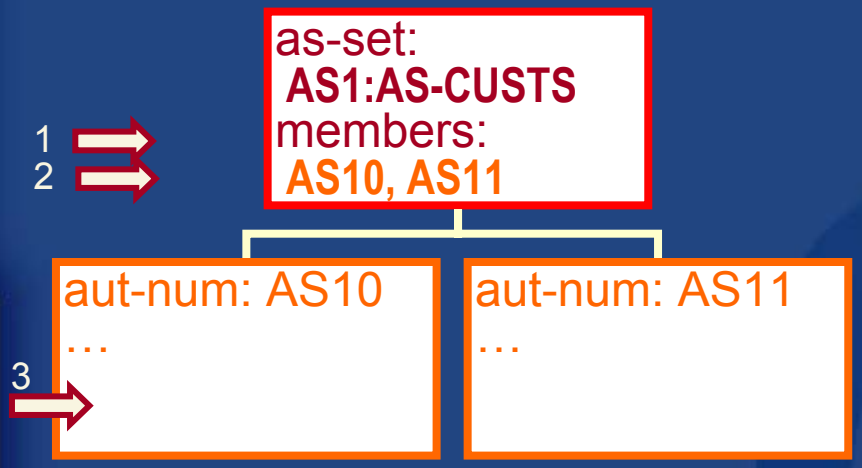
- Route-set attribute starts with “rs-”

‘Set-’ objects and their members

- Two ways of referencing members

members

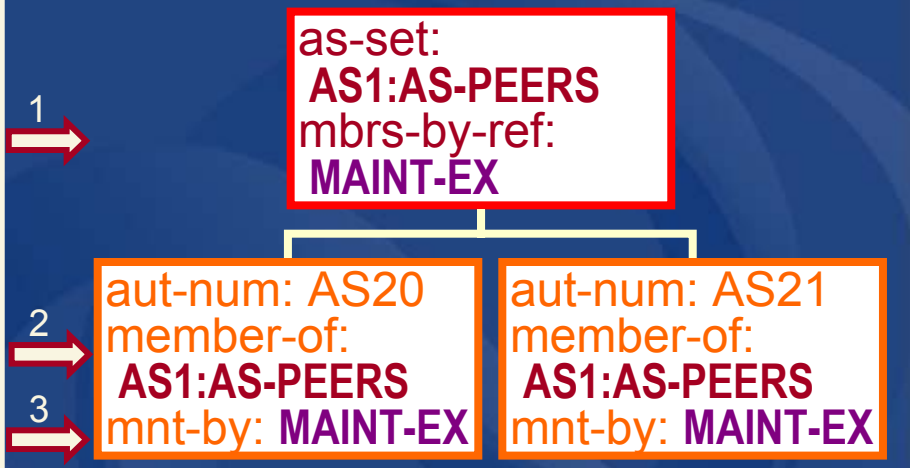
- members specified in the ‘set-’ object



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



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: 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: 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: E.g. 30.0.0.0/8^16 contains all the more specifics of 30.0.0.0/8 which are length of l6 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: 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 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

Action specification

- Policy action in RPSL
  - Set or modify route attributes
    - assigning a preference to a route
    - adding a BGP community to the BGP community path attribute
    - setting the Multi-Exist-Discriminator (MED) attribute
    - instructing routers to perform special operations
      - Route flap damping

Action specification (cont.)

- Routing policy attributes (rp-attributes)
  - Specified in the RPSL dictionary
  - Each action in RPSL is terminated by ";"
  - Possible to form composite policy actions
  - Actions are executed left to right

  Sample:

```
aut-num:  AS1
import:   from AS2
          action pref = 10; med = 0;
          community.append (10250, 3561:10);
          accept { 128.9.0.0/16 }
```

Seven rp-attributes

pref	To assign local preference to the routes accepted
med	To assign a value to the Multi-Exit-Discriminator BGP attribute
dpa	To assign a value to the DPA BGP attribute
aspath	To prepend a value to the AS_PATH BGP attribute
community	To assign a value to or to check the value of the community BGP attribute
next-hop	To assign next hop routers to static routes
cost	To assign a cost to static routes

Syntax of policy actions and filters

- pref
  - Can be assigned a positive integer
  - Smaller values represent higher preference
    - Note: Larger values represent higher preference in BGP local pref attribute
  - Sample:
    - pref = 10;
    - pref = 80;

Syntax of policy actions and filters (cont.)

- Med
  - BGP multi exit discriminator
  - Can be assigned either:
    - a positive integer
      - To set med to a number
    - or the word “igp_cost”
      - To set med to the IGP metric
  - Sample
    - med = 10;
    - med = igp_cost;

Syntax of policy actions and filters (cont.)

- dpa
  - BGP destination preference attribute
  - Can be assigned a positive integer
  - Sample:
    - dpa = 100;

Syntax of policy actions and filters (cont.)

- Community
  - BGP community attribute
  - Can be assigned either:
    - A 4 byte (32 bits) integer
      - Can be specified using two 2 byte integers separated by “:”
        - » First 2 bytes (16 bits) can represent ASN
        - » Last 2 bytes (16 bits) can represent a semantics of its choice
    - Or keywords
      - Internet, no_export, no_advertise
  - Sample
    - community .={100};
    - community .={NO_EXPORT};
    - community .={3561:10};

Syntax of policy actions and filters (cont.)

- community (cont.)
  - .append
    - Append community value
      - E.g. community.append(1:1)
  - .delete
    - Delete community value
      - E.g. community.delete(1:1)
  - .contains
    - Add filter
      - E.g. community.contains(1:1)
    - Shortcut to .contains
      - E.g. community(1:1)

Questions ?

Asia Pacific Network Information Centre

APNIC

APNIC
Internet Routing Registry

Using RPSL in practice

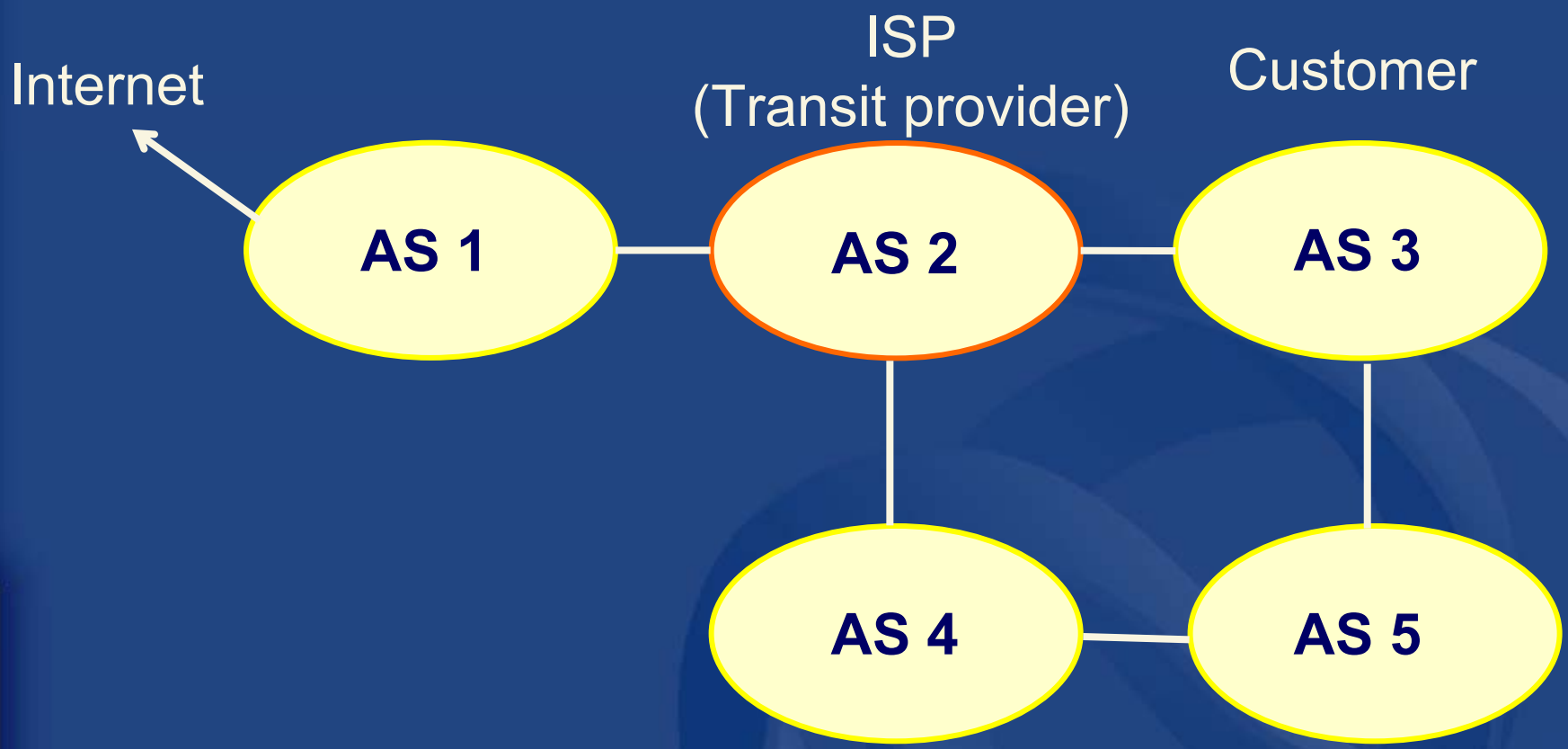
Overview

- Review examples of routing policies expression
  - Peering policies
  - Filtering policies
  - Backup connection
  - Multihoming policies

RPSL - review

- Purpose of RPSL
  - Allows specification of your routing configuration in the public IRR
    - Allows you to check “Consistency” of policies and announcements
  - Gives opportunities to consider the policies and configuration of others

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:      SN36-AP
tech-c:       MF53-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

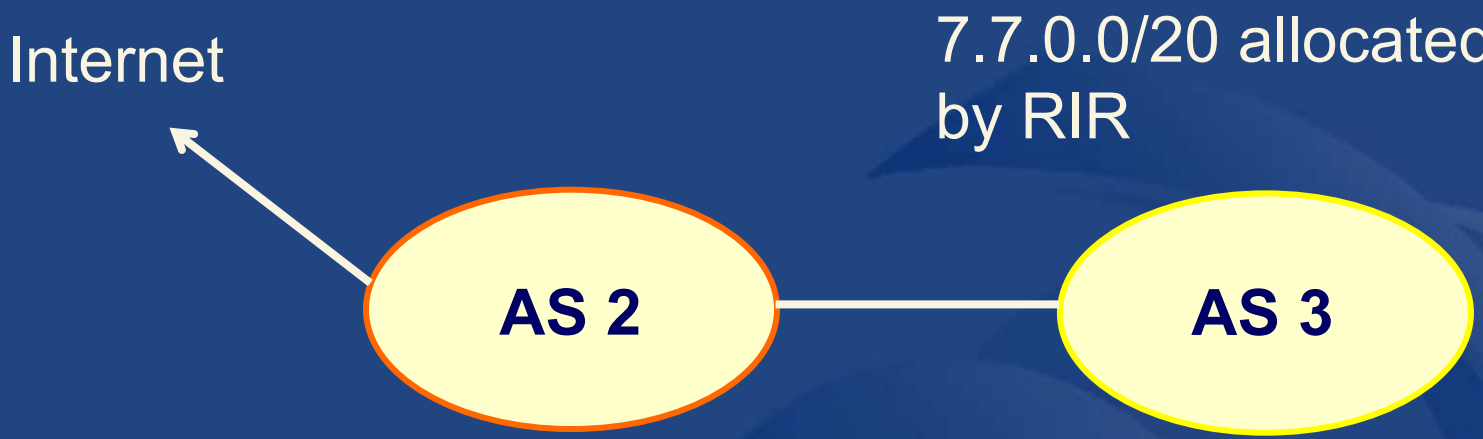
aut-num:	AS2
import:	from AS1 accept ANY
import:	from AS2:AS-CUSTOMERS accept <^AS2:AS-CUSTOMERS+$>
export:	to AS2:AS-CUSTOMERS announce ANY
export:	to AS1 announce AS2 AS2:AS-CUSTOMERS

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 on 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

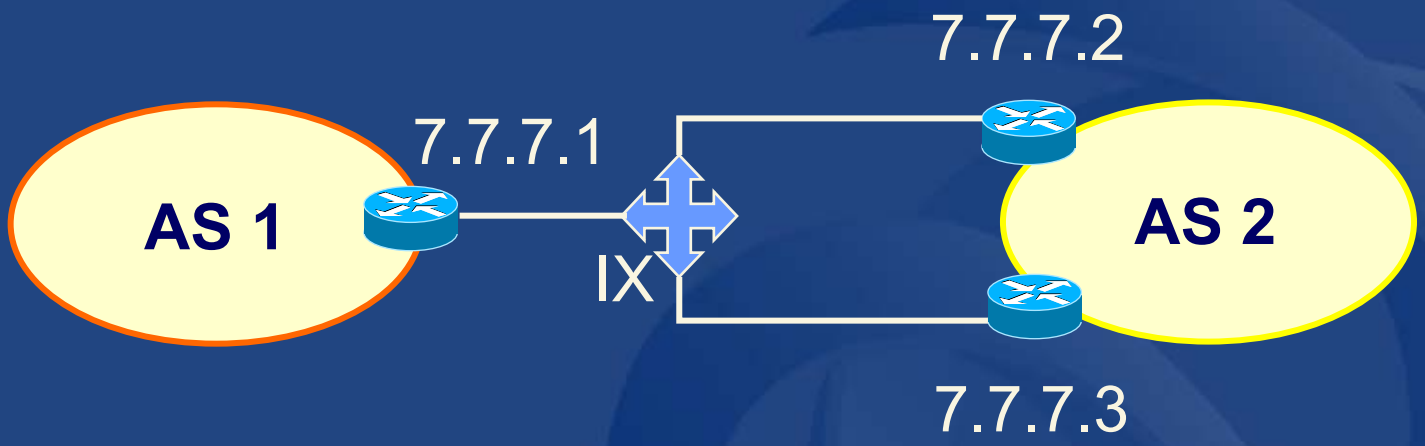
Filter configuration using route-set – AS2

```
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.)

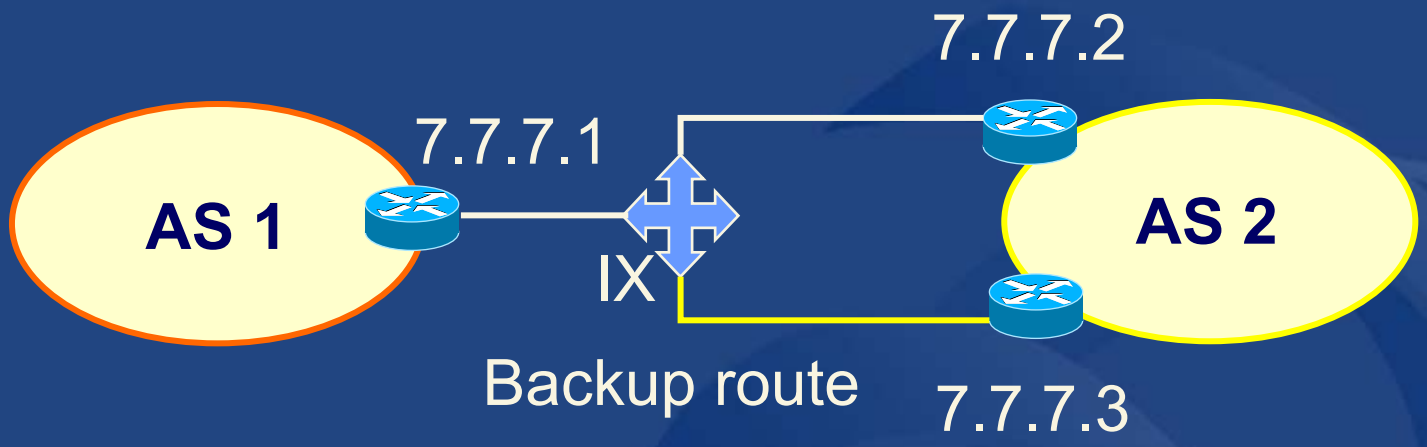
```
aut-num:     AS1
import:      from AS2 at 7.7.7.1 accept <^AS2+$>
```

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

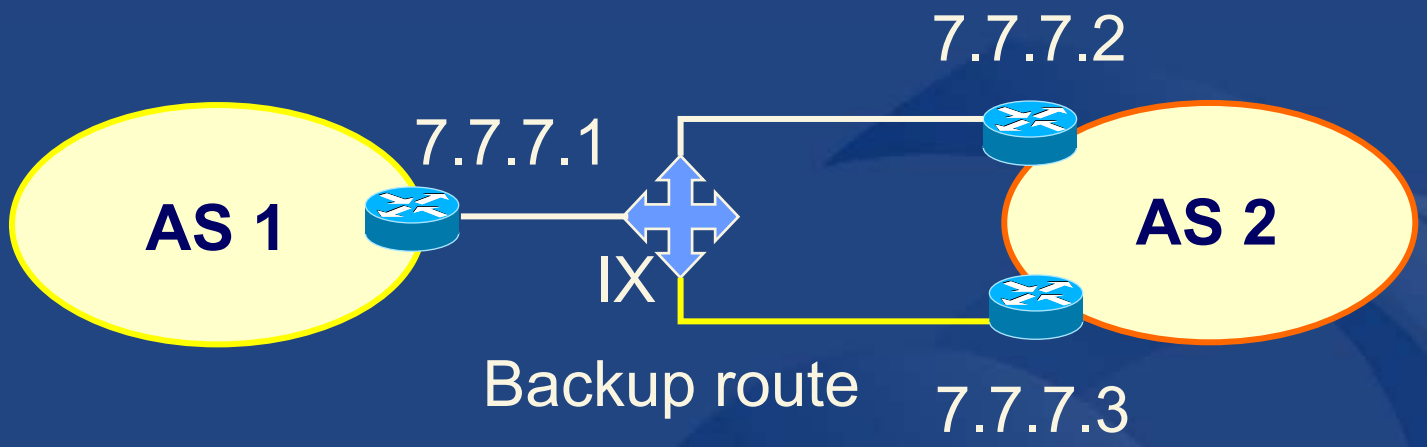
Describing simple backup connections: AS1 (cont.)

```
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

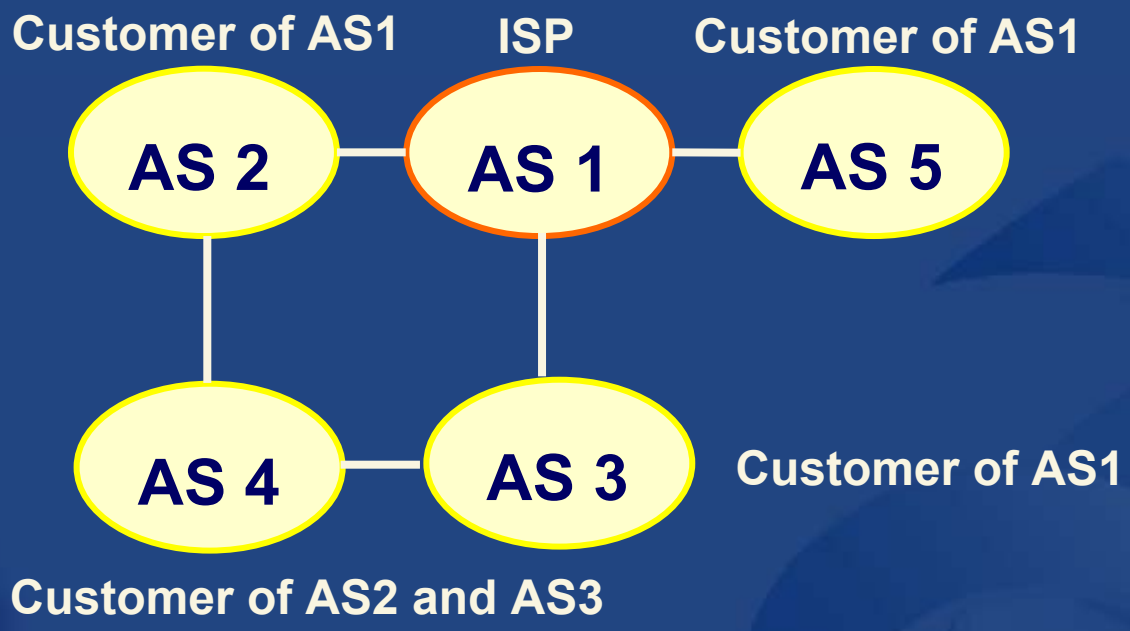
Describing simple backup connections: AS2 (cont.)

```
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

Multihome routing policy



AS1's base policy

- Only accepts routes from customers that are originated by the customer
- or by the customer's customers

Multihome routing policies (cont.)

```
aut-num:  AS1
import:   from AS2 accept (AS2 or AS4) AND
          <^AS2+AS4*$>
import:   from AS3 accept (AS3 or AS4) AND
          <^AS3+AS4*$>
import:   from AS5 accept AS5 AND <^AS5+$>
```

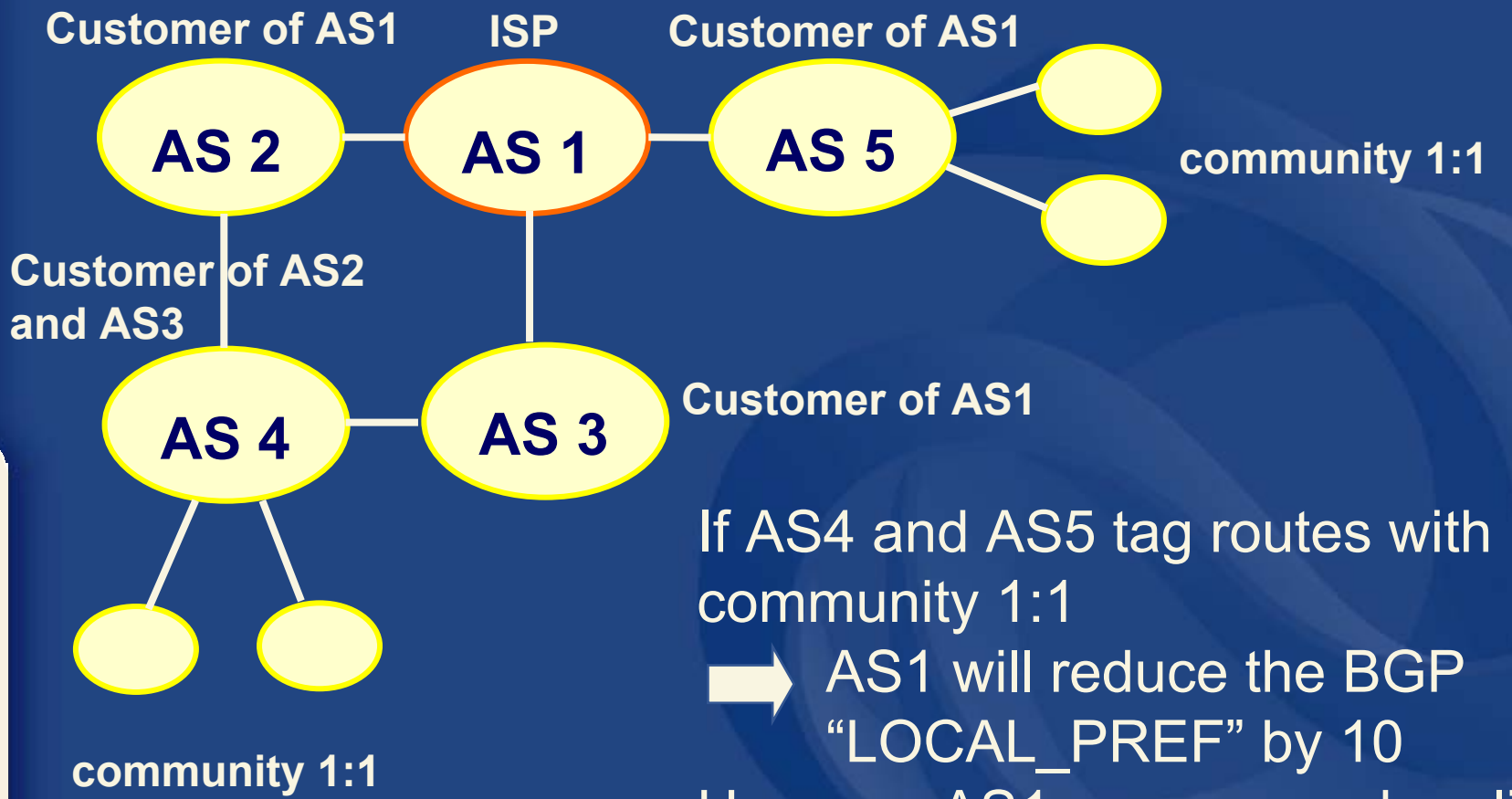
Review - BGP community attribute

- Use of the BGP community attribute to provide support for:
  - Load balancing
  - Backup connections
- Basic premise of RFC1997:
  - To allow a customer to configure the BGP “LOCAL_PREF” on a provider’s router

Review - BGP community attribute (cont.)

- Method to group destinations into communities and apply routing decisions
  - Community = a group of destinations (i.e. prefixes) that share some common attribute
  - Optional transitive attribute of variable length
- BGP community attribute
  - Format = aa:nn
    - aa = ASN
    - nn = 1 to 65,536

Multihome routing policies using the RPSL community attribute



If AS4 and AS5 tag routes with community 1:1

➡ AS1 will reduce the BGP "LOCAL_PREF" by 10

How can AS1 express such policy in RPSL?

Multihome routing policies using the RPSL community attribute (cont.)

```
aut-num: AS1
import:   from AS2 action pref=10;
          accept (AS2 or AS4) AND <^AS2+AS4*$>  AND community(1:1)
import:   from AS2 action pref=0;
          accept (AS2 or AS4) AND <^AS2+AS4*$>

import:   from AS3 action pref=10;
          accept (AS3 or AS4) AND <^AS3+AS4*$> AND community(1:1)
import:   from AS3 action pref=0;
          accept (AS3 or AS4) AND <^AS3+AS4*$>

import:   from AS5 action pref=10;
          accept AS5 AND <^AS5+$> AND community(1:1)
import:   from AS5 accept pref=0;
          accept AS5 AND <^AS5+$>
```

Questions ?

If AS4 tags community 1:1 to the announcement to AS2 but not to AS3, what AS1 will do?

RtConfig

Case studies

Overview

- IRRToolSet common options
- RtConfig commands
- Alias and template file
- Case studies

IRRToolSet common options

-T [whois_query | whois_response | input | all]	Trace the argument. Useful for debugging.
-version	Print the version number and quit
-h <host-name>	Establish a whois connection to host <host-name>.
-p <port-no>	Establish an whois connection to port <port-no>. The default is 43.
-protocol <protocol>	Use the <protocol> to connect to the IRR server. (irrd, ripe and ripe_perl)
-s <source-list>	Define object source
-prompt <new-prompt>	Change the prompt to <new-prompt>. The default is “RtConfig”

IRRToolSet common options (cont.)

-cisco_empty_lists	Only affects Cisco configurations. It forces interpreting ANY/NOT ANY prefix filters as universal/empty set of prefixes, and produces access lists for them. By default only warning is issued.
-cisco_use_prefix_lists	Only affects Cisco configurations. RtConfig output will be generated with prefix-lists.
-supress_martian	Only affects Cisco configurations. All access lists generated will deny the martian routes.

RtConfig commands

- @RtConfig import <ASN-1> <rtr-1> <ASN-2> <rtr-2>
  - Generate import filters where <rtr-1> in <ASN-1> is importing routes from <rtr-2> in <ASN-2>

  i.g. @RtConfig import AS1 10.20.0.3 AS2 10.3.15.2

- @RtConfig export <ASN-1> <rtr-1> <ASN-2> <rtr-2>
  - Generate export filters where <rtr-1> in <ASN-1> is exporting routes to <rtr-2> in <ASN-2>

  i.g. @RtConfig export AS1 10.20.0.3 AS2 10.3.15.2

RtConfig command (cont.)

- @RtConfig static2bgp <ASN-1> <rtr-1>
  - Generate the import policies of <ASN-1> where “protocol STATIC” or “protocol STATIC into BGP4” is used

  i.g. @RtConfig static2bgp AS1 10.3.0.1

RtConfig Cisco specific commands

- @RtConfig set cisco_map_name = <map-name>
  - Use <map-name> as the name for the route maps generated.
  - %d in <map-name> ➡ replaced by the peer's ASN.
  - Second %d ➡ replaced by an integer incremented No. of new map file creation.
  - The default cisco_map_name is "MyMap_%d_%d".

  i.g. @RtConfig set cisco_map_name = "AS%d-IMPORT"

- @RtConfig set cisco_max_preference = <no>
  - <no> is an integer defaulting to 1000.
  - Instruct RtConfig to start using preferences from <no> and counting down from there.

RtConfig Junos specific commands

- @RtConfig set junos_policy_name = <policy-name>
  - Instruct RtConfig to use <policy-name> as the name for the policy statement generated.
  - %d in <policy-name> ➡ replaced by the peer's ASN.
  - Second %d ➡ replaced by an integer incremented No. of new policy file creation.
  - The default junos_policy_name is "policy_%d_%d".

RtConfig BCC specific commands

- @RtConfig set bcc_max_preference =<no>
  - <no> is an integer defaulting to 1000
  - Instruct RtConfig to start using preferences from <no> (most preferred) and counting down from there

RtConfig vendor specific commands

- More commands available
  - RtConfig man page
    - http://www.ripe.net/ripencc/pub-services/db/irrtoolset/documentation/RtConfig.html

Create an alias

- Create bashrc alias

  # alias rt='RtConfig -h testwhois.apnic.net -p 43 -s RRTEST -protocol ripe'

Template file usage

- RtConfig can be fed from the “template file”
  - Easy way to type in all the commands (per router) once
  - Listing of commands that translate IRR policy into router configuration
  - Lines NOT starting with “@RtConfig” will be printed out as-is
    - Allows to specify additional specific router-config commands
    - Comments (to Cisco configuration) start with “!”

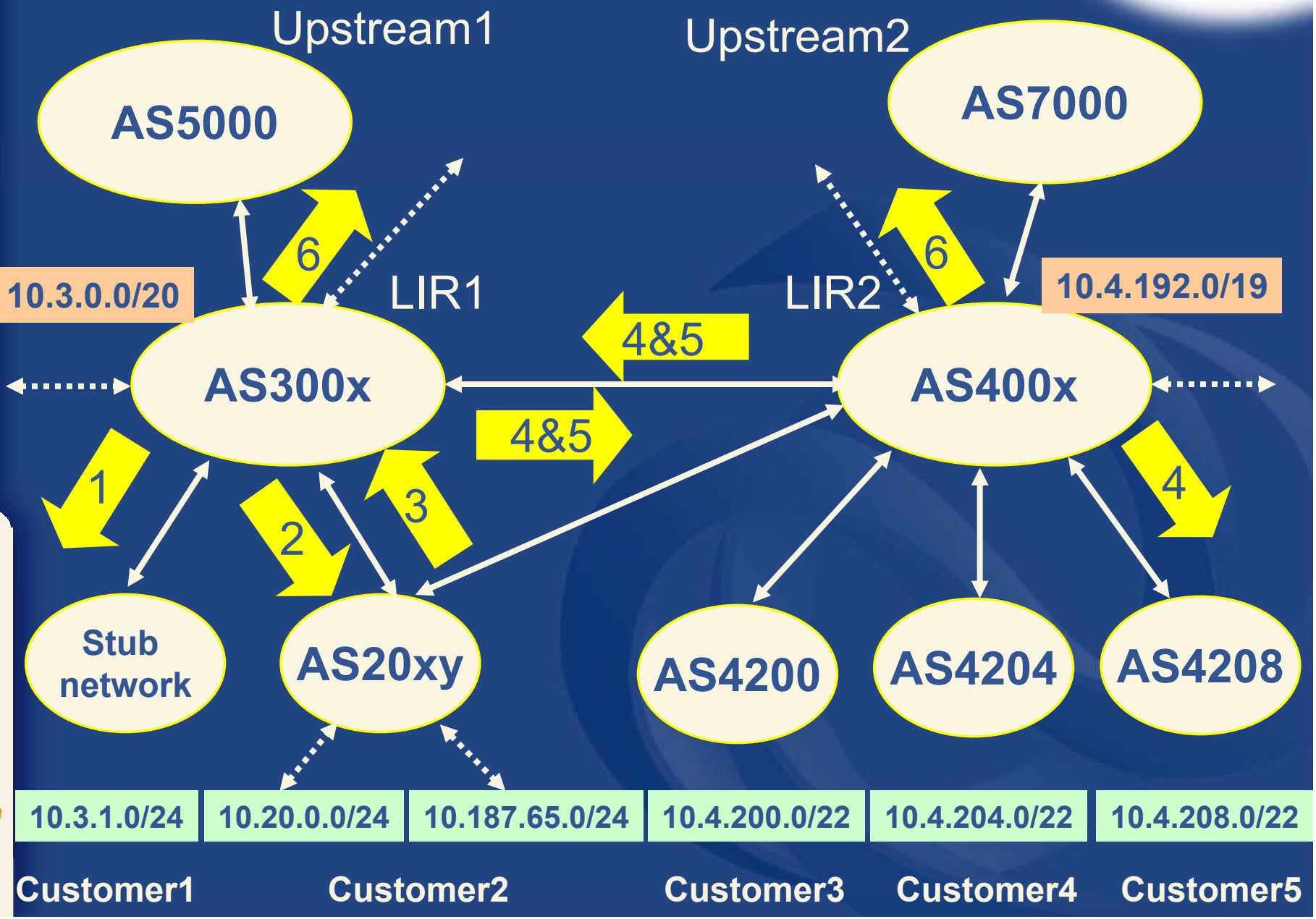
Template file usage (cont.)

- Create a template file as a txt file
  - e.g. case1-rt-file.txt

```
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
```

- Use the following command line (alias is handy)
  - rt < case1-rt-file.txt > output-case1.txt
- RtConfig output can be stored into the specified file name

Experimental Setup: AS Relations



Case Studies Overview

- AS3000 has a number of routing requirements:
  - Static routes injected at customer edge
  - BGP peering with customers
  - BGP peering with peers
  - BGP peering with upstream providers
- Learning objectives
  - how to express routing policies via RPSL
  - how to extract information from IRR by using RtConfig

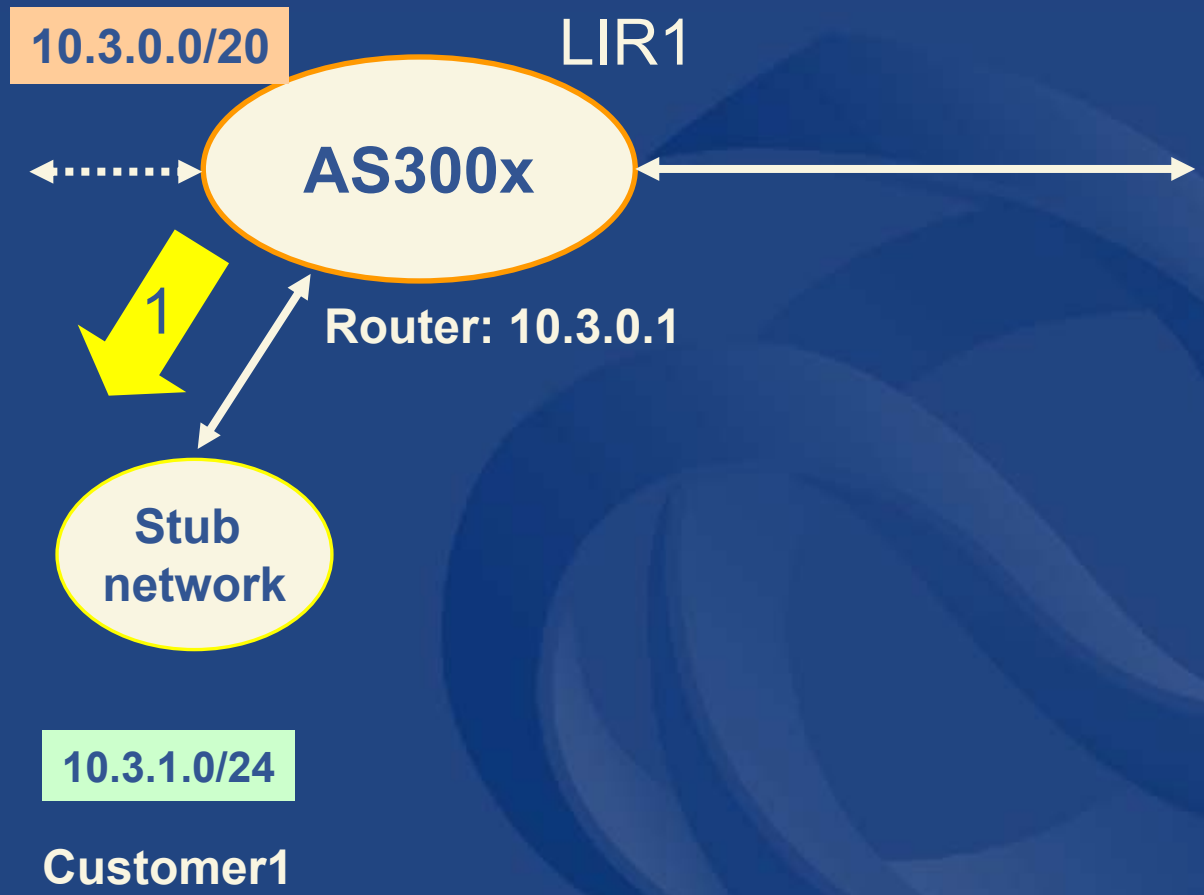
Case study overview (cont.)

- Case 1: Provider inserting static routes
- Case 2: Provider setup for various BGP customers
- Case 3: Multi-homed customer set-up
- Case 4: Peering set-up
- Case 5: Multiple connections to the same neighbour
- Case 6: Using communities
- Case 7: Various filtering

Case study overview (cont.)

- AS “families” used in the examples
  - AS300x = {AS3000, AS3001, AS3007, …}
  - Similar “position”, policies of different complexity
    - Enables keeping track of changes throughout the course
- Neighbour = customer, peer, upstream…

Case 1: Static route importation into BGP



Case 1: Static Route Importation into BGP

- Use policy to filter static routes into BGP
  - Allows for martian/bogon filtering
  - AS path pre-pending, setting the value of "pref", tagging routes with special communities, etc.
- Simplest option:

```
aut-num:    AS3000
import:     protocol STATIC into BGP4
            from  AS3000 accept {10.3.1.0/24}
```

Case 1: RtConfig command

RtConfig> @RtConfig static2bgp AS3000 10.3.0.1

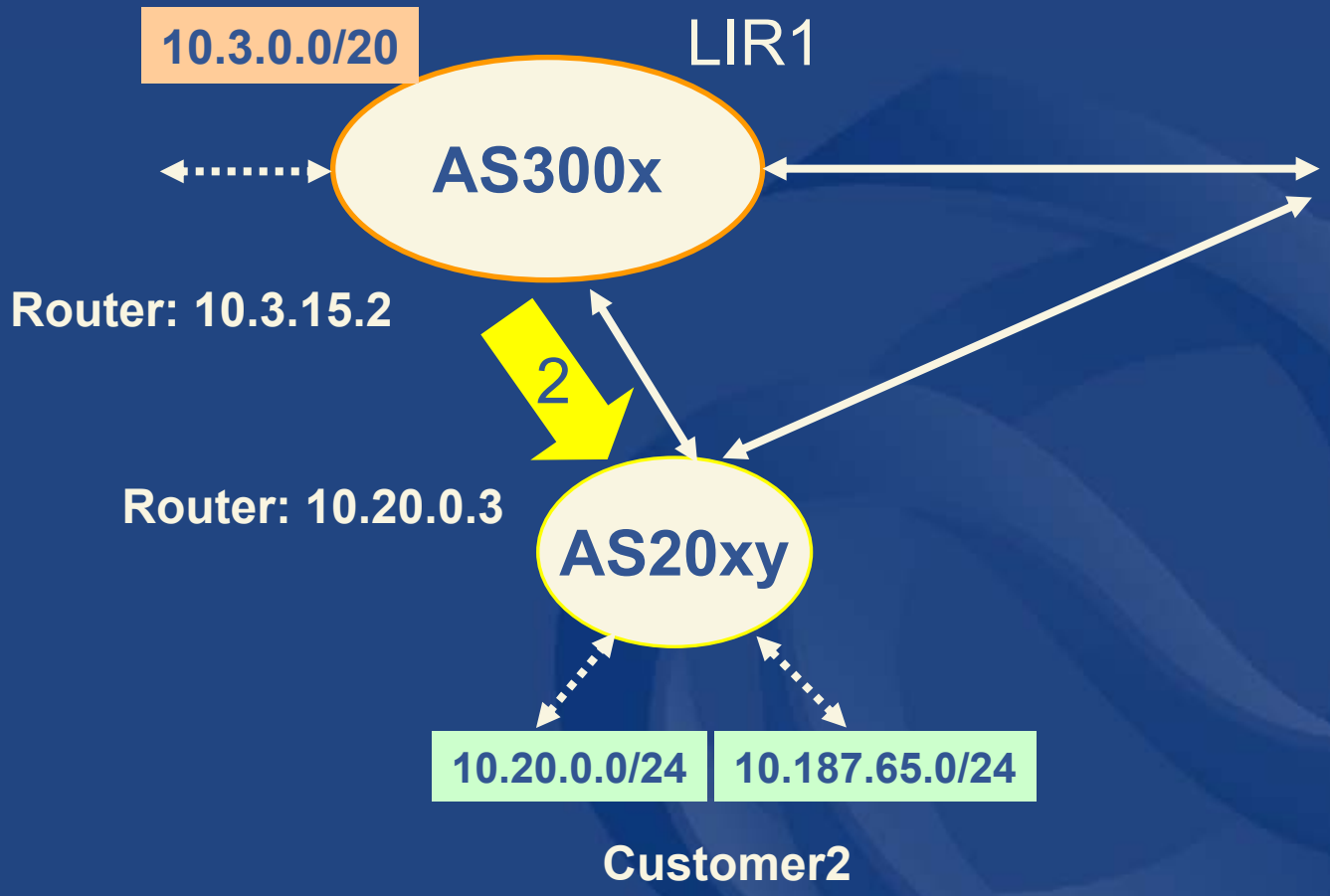
```
!
no ip prefix-list pl100
ip prefix-list pl100 permit 10.3.1.0/24
Ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map MyMap_3000_1
!
route-map MyMap_3000_1 permit 1
 match ip address prefix-list pl100
!
router bgp 3000
Redistribute static route-map My_Map_3000_1
```

Default value

Case 1: Summary

- Static route import
  - aut-num
    - import: protocol STATIC into BGP4 from <ASN> accept <filter>
  - RtConfig command
    - Static2bgp <ASN> <rtr-IP>

Case 2: Provider set-up for BGP customers



Case 2: Review customer aut-num object

```
aut-num:    AS2000
import:     from AS3000 accept ANY
export:     to AS3000 announce AS2000
[…]
```

Case 2: Review provider aut-num object

```
aut-num:    AS3000
import:     from AS2000 accept AS2000
export:     to AS2000 announce ANY
[…]
```

- The simplest policy is strict customer/provider relationship
  - Customer sends it routes to provider
  - Customer accepts everything the provider sends

Case 2: Review route objects

- whois –i origin AS3000

```
route:   10.3.0.0/20
descr:   Route for AS3000
origin:  AS3000
[…]
```

- whois –i origin AS2000

```
route:   10.20.0.0/24
descr:   Route for AS2000
origin:  AS2000
[…]
```

```
route:  10.187.65.0/24
descr:  Route for AS2000
origin:  AS2000
[…]
```

Case 2: RtConfig template file for AS3000

```
# cat case-2.1-rt-file.txt
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS3000 10.3.15.2 AS2000 10.20.0.3
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS3000 10.3.15.2 AS2000 10.20.0.3
```

"%d" gets replaced with the peer-ASN

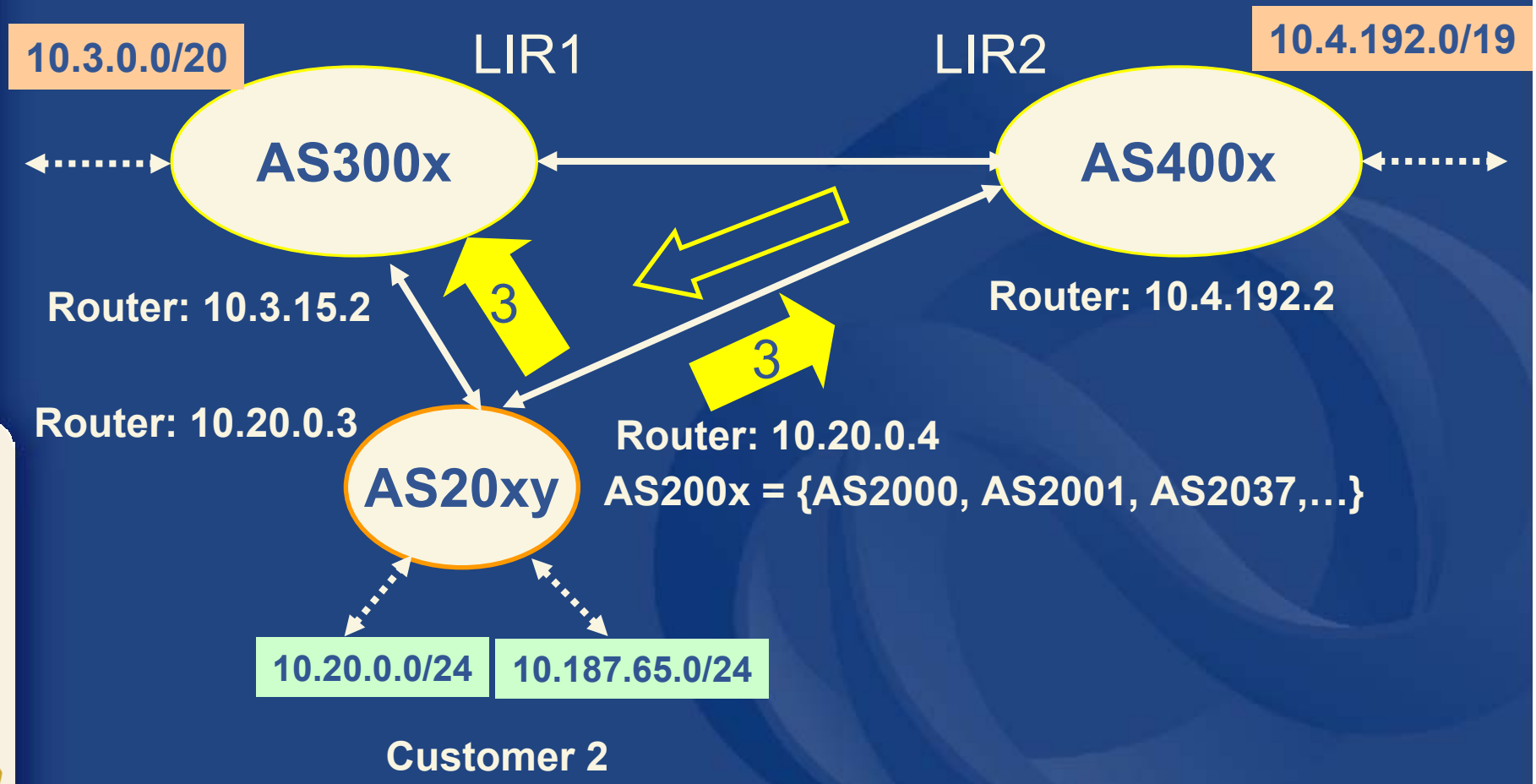
Case 2: RtConfig output

```
no ip prefix-list pl100
ip prefix-list pl100 permit 10.20.0.0/24
ip prefix-list pl100 permit 10.187.65.0/24
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS2000-IMPORT
!
route-map AS2000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 3000
neighbour 10.20.0.3 route-map AS2000-IMPORT in
!
no route-map AS2000-EXPORT
!
route-map AS2000-EXPORT permit 1
!
route bgp 3000
neighbour 10.20.0.3 route-map AS2000-EXPORT out
```

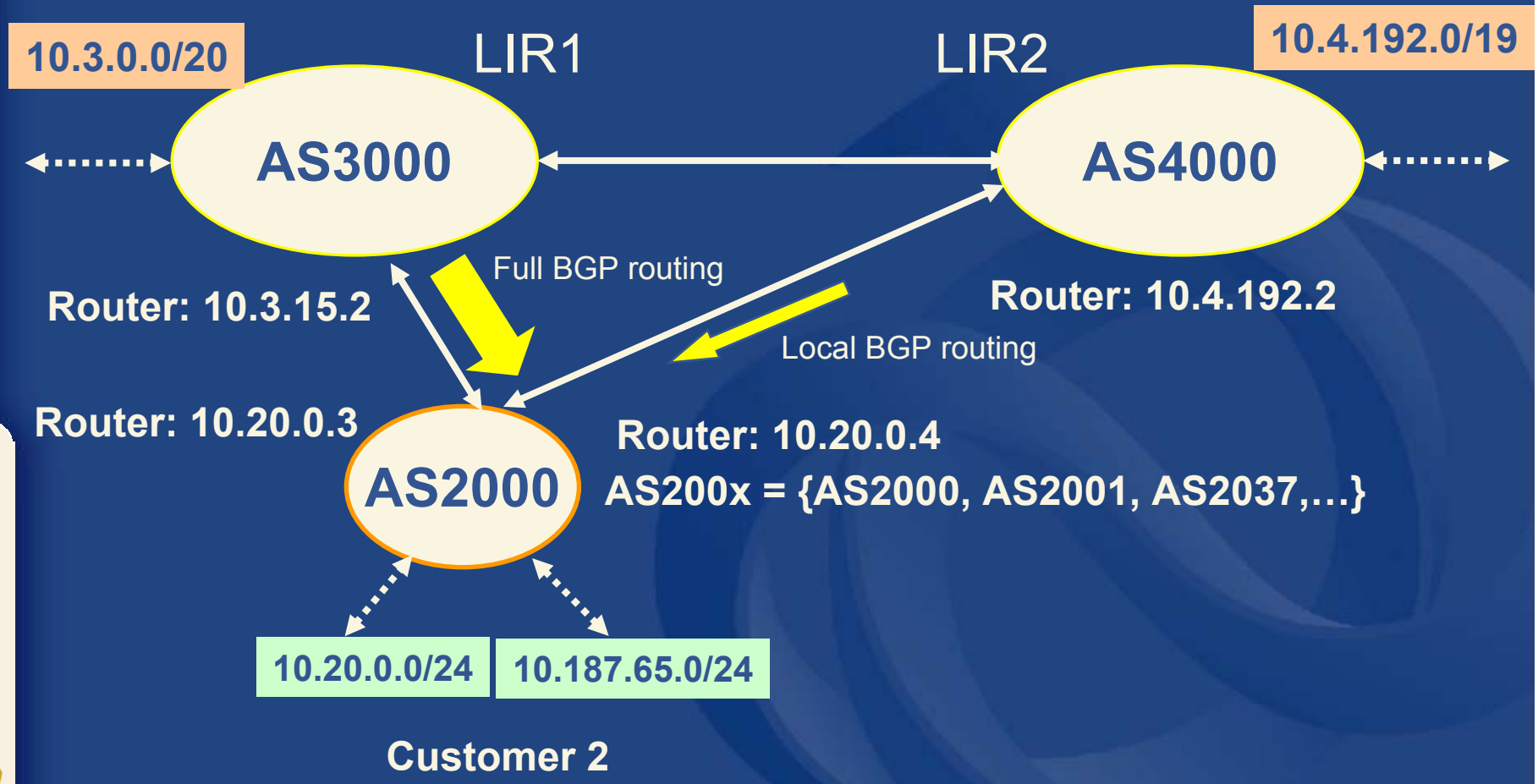
Case 2: Summary

- aut-num contains policy specifications
  - In import and export attributes
- RtConfig commands per neighbour
  - set cisco_map_name = “AS%d-IMPORT”
  - import <ASN1> <rtr1> <ASN2> <rtr2>
  - set cisco_map_name = “AS%d-EXPORT”
  - Export <ASN1> <rtr1> <ASN2> <rtr2>
- Use RtConfig template file for efficiency

Case 3: Multihomed customer setup



Case 3.1: "Partial" multihoming



Case 3.1: "Partial" multihoming, review customer's IRR objects

```
aut-num:    AS2000
import:     from AS3000 accept ANY
export:     to AS3000 announce AS2000
import:     from AS4000 accept AS4000
export:     to AS4000 announce AS2000
[…]
```

```
route:   10.20.0.0/24
origin:  AS2000
[…]
```

```
route:   10.187.65.0/24
origin:  AS2000
[…]
```

Case 3.1: RtConfig template file for AS2000

```
# cat case-3.1-rt-file.txt
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.4 AS4000 10.4.192.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2000 10.20.0.4 AS4000 10.4.192.2
```

Case 3.1: RtConfig output

```
!
no ip prefix-list pl100
ip prefix-list pl100 permit 0.0.0.0/0 le 32
!
no route-map AS3000-IMPORT
!
route-map AS3000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-IMPORT in
```

Case 3.1: RtConfig output (cont.)

```
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-IMPORT in
```

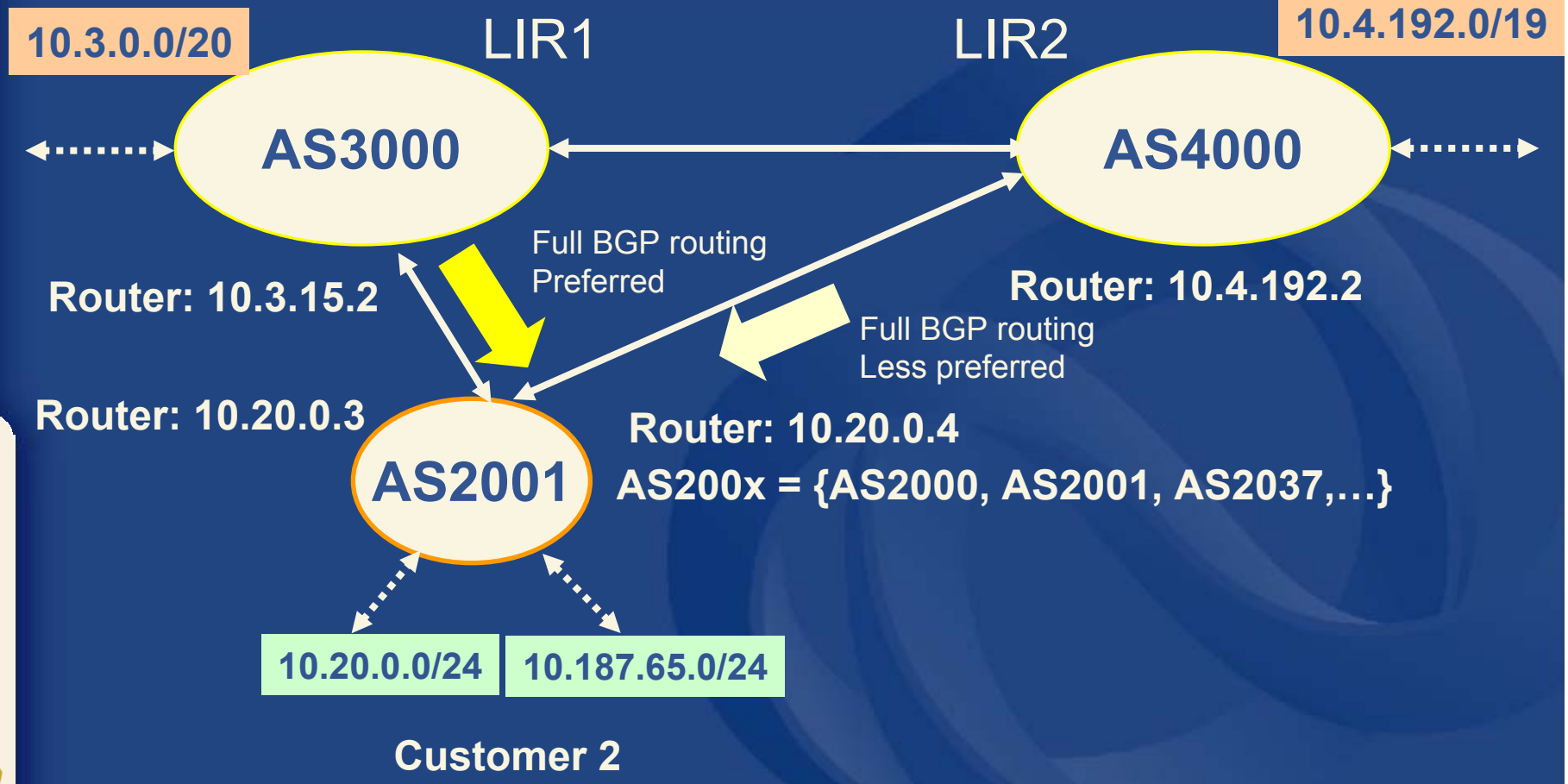
Case 3.1: RtConfig output (cont.)

```
!
no ip prefix-list pl102
ip prefix-list pl102 permit 10.20.0.0/24
ip prefix-list pl102 permit 10.187.65.0/24
ip prefix-list pl102 deny 0.0.0.0/0 le 32
!
no route-map AS3000-EXPORT
!
route-map AS3000-EXPORT permit 1
 match ip address prefix-list pl102
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-EXPORT out
```

Case 3.1: RtConfig output (cont.)

```
!
no route-map AS4000-EXPORT
!
route-map AS4000-EXPORT permit 1
 match ip address prefix-list pl102
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-EXPORT out
```

Case 3.2: “Full” multihoming



Case 3.2: "Full" multihoming, review customer's IRR objects

```
aut-num:    AS2001
import:     from AS3000 action pref=50; accept ANY
export:     to AS3000 announce AS2001
import:     from AS4000 action pref=100; accept ANY
export:     to AS4000 announce AS2001
[…]
```

```
route:   10.20.0.0/24
origin:  AS2001
[…]
```

```
route:   10.187.65.0/24
origin:  AS2001
[…]
```

- Lower the "pref", the more preferred the route
- Controlling outbound traffic

Case 3.2: RtConfig template file for AS2001

```
!
! Setting max pref to 100
@RtConfig set cisco_max_preference = 100
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2001 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2001 10.20.0.4 AS4000 10.4.192.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2001 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2001 10.20.0.4 AS4000 10.4.192.2
```

Note: See slide #7 “RtConfig Cisco specific commands”

Case 3.2: RtConfig output

```
!
! Setting max pref to 100
!
!
no ip prefix-list pl100
ip prefix-list pl100 permit 0.0.0.0/0 le 32
!
no route-map AS3000-IMPORT
!
route-map AS3000-IMPORT permit 1
match ip address prefix-list pl100
set local-preference 50
!
router bgp 2001
neighbor 10.3.15.2 route-map AS3000-IMPORT in
!
```

Case 3.2: RtConfig output (cont.)

```
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
match ip address prefix-list pl100
set local-preference 0
!
router bgp 2001
neighbor 10.4.192.2 route-map AS4000-IMPORT in
!
```

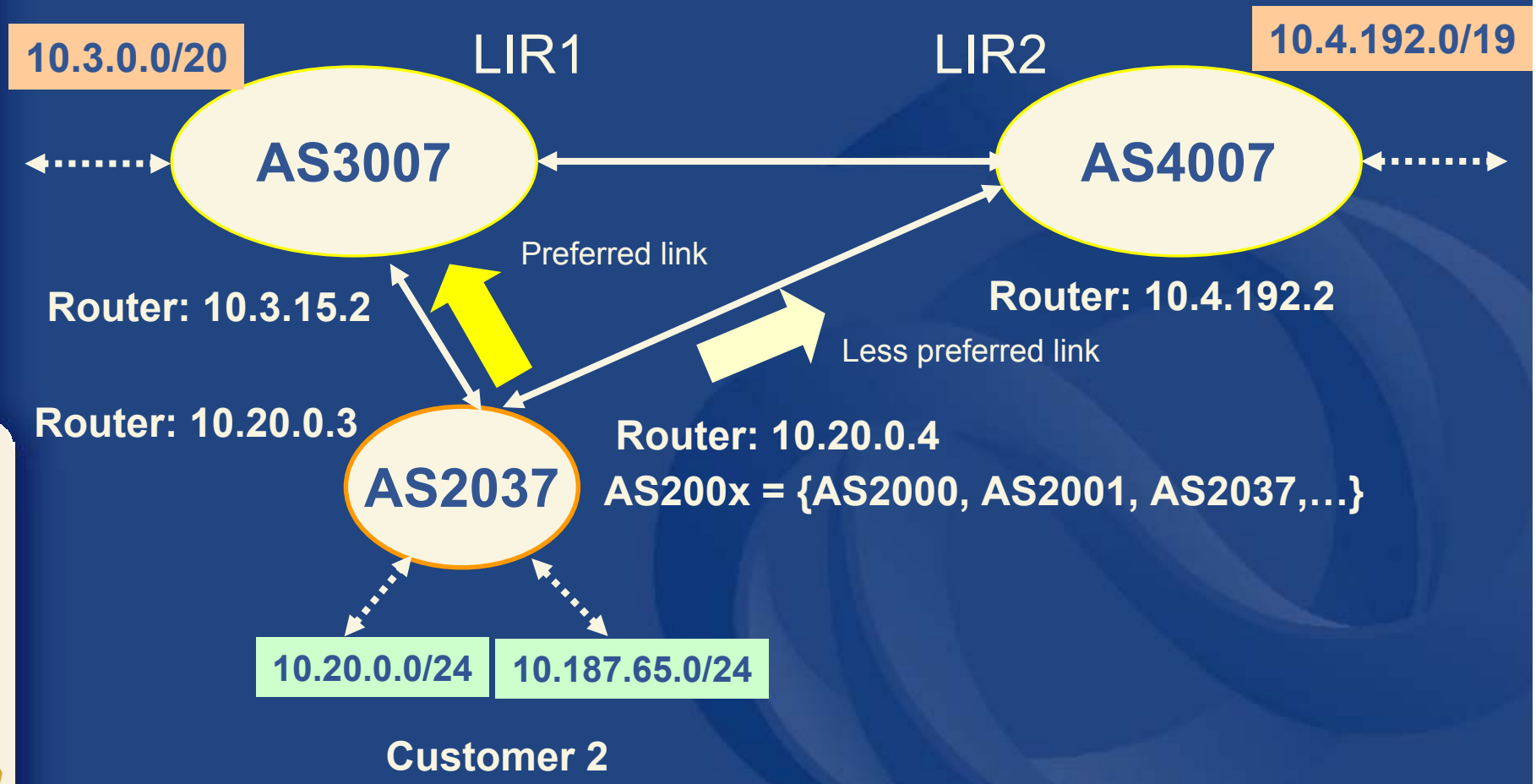
Case 3.2: RtConfig output (cont.)

```
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.20.0.0/24
ip prefix-list pl101 permit 10.187.65.0/24
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS3000-EXPORT
!
route-map AS3000-EXPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2001
neighbor 10.3.15.2 route-map AS3000-EXPORT out
!
```

Case 3.2: RtConfig output (cont.)

```
!
no route-map AS4000-EXPORT
!
route-map AS4000-EXPORT permit 1
match ip address prefix-list pl101
!
router bgp 2001
neighbor 10.4.192.2 route-map AS4000-EXPORT out
```

Case 3.3: Controlling inbound traffic



Case 3.3: Controlling inbound traffic

- Prepending your ASN on the less preferred link, in the “export” attribute
  - E.g. AS2037 connected to AS3007 and AS4007

```
aut-num:   AS2037
Remarks:   More preferred link to A3007;
export:    to AS4007
           action aspath.prepend (AS2037,
           AS2037); announce AS2037
export:    to AS3007 announce as2037
```

Case 3.3: RtConfig template file for AS2037

```
!
! The Export Policy - THIS IS the only IMPORTANT bit! –
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2037 10.20.0.3 AS3007 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2037 10.20.0.4 AS4007 10.4.192.2
```

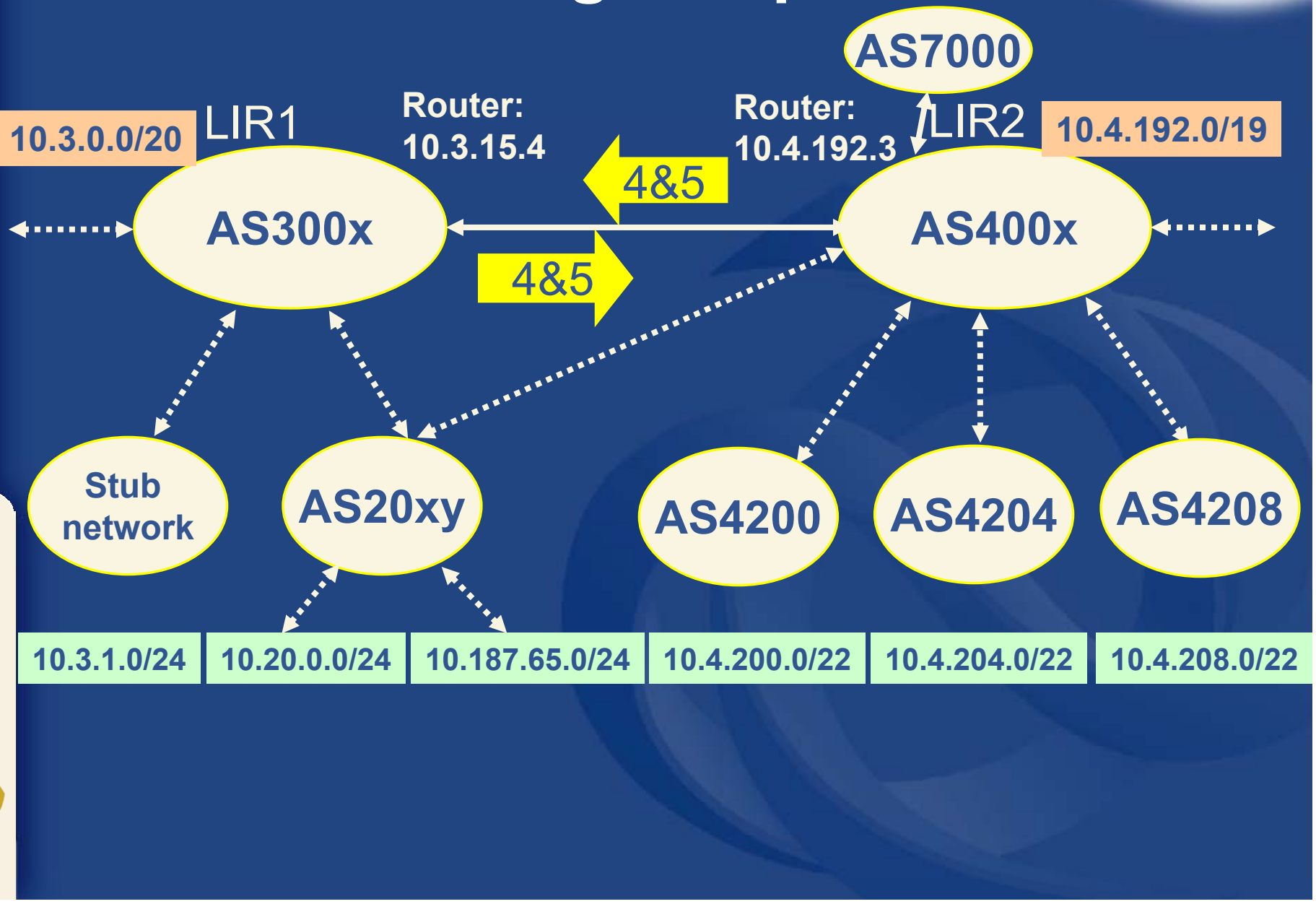
Case 3.3: RtConfig output

```
!
! The Export Policy - THIS IS the only IMPORTANT bit! –
!
!
no route-map AS3007-EXPORT
!
route-map AS3007-EXPORT permit 1
!
router bgp 2037
neighbor 10.3.15.2 route-map AS3007-EXPORT out
!
!
no route-map AS4007-EXPORT
!
route-map AS4007-EXPORT permit 1
set as-path prepend 2037 2037
!
router bgp 2037
neighbor 10.4.192.2 route-map AS4007-EXPORT out
```

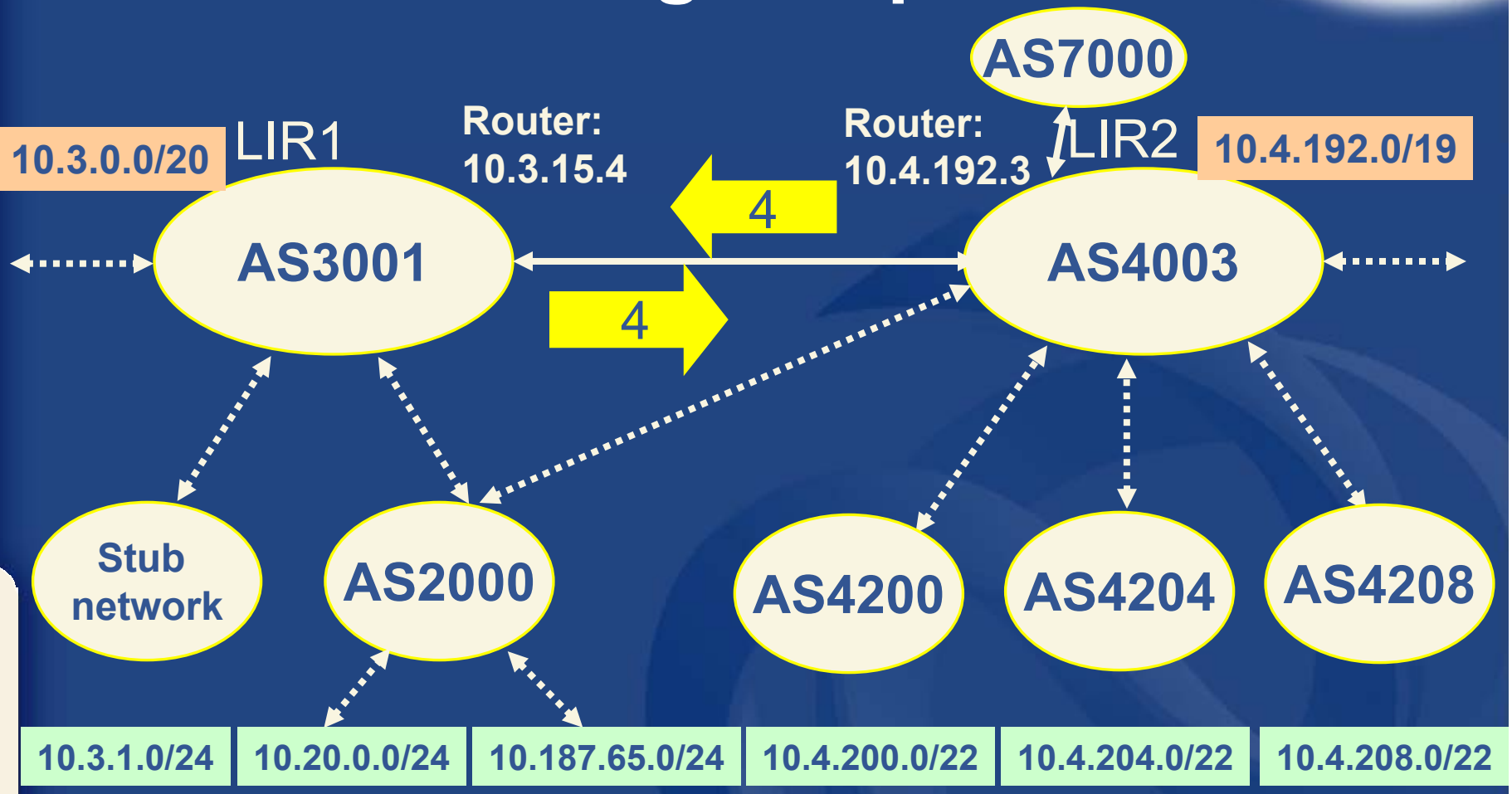
Case 3: Summary

- Policy expressions to achieve/control multihoming
  - Outbound traffic
    - Set the value of local preference
      - **"action pref=NN" in the "import" attribute of aut-num object**
  - Inbound traffic
    - Modify as-path length
      - **"action aspath.prepend (ASN)" in the "export" attribute of aut-num object**
- RtConfig template file has to contain a set of commands for each neighbour

Case 4: Peering setup



Case 4: Peering setup



Case 4: BGP with peers – AS4003 view, review db objects

```
aut-num:    AS4003
import:     from AS3001 accept AS3001 AS2000
Import:     from AS4003:AS-CUSTOMERS accept PeerAS
export:     to AS4003:AS-CUSTOMERS announce ANY
export:     to AS4003:AS-PEERS announce AS4003
            AS2000 AS4003:AS-CUSTOMERS
[…]
```

```
as-set:     AS4003:AS-CUSTOMERS
descr:      AS4003 Customers
members:    AS4200, AS4204, AS4208
```

```
as-set:     AS4003:AS-PEERS
descr:      AS4003 Peers
members:    AS3001,AS7000
```

Case 4: BGP with peers – AS3001 view, review db objects

aut-num:	AS3001
import:	from AS4003 accept <^AS4003+AS4003:AS-customers*$>
export:	to AS4003 announce AS3001 AS2000

- Since the <filter> is expressed using regular expression, AS-path list is created instead of the prefix list
  - See slide 169 "Case 4: RtConfig command"

Case 4: BGP with peers

- Peering policy does not need to be symmetrical
  - Compare slide 166 and 167
    - E.g. AS4003 is announcing AS2000 back to AS3001, but AS3001 is not accepting it
- An implicit logical OR exists between multiple ASN and/or multiple as-sets in the <filter>
  - = summary of address range announced from those ASNs

Case 4: RtConfig command

@RtConfig import AS3001 10.3.15.4 AS4003 10.4.192.3

```
!
no ip as-path access-list 1
ip as-path access-list 1 permit
    ^(_4003)+(_(4200|4204|4208))*$
```

Case 4: Summary

- Peers can announce to each other
  - only own ASN, and/or
  - direct customers, and/or
  - multihomed customers
- Peers can accept some of those announcements
  - but set different “pref” value on them
  - or modify other route parameters
  - if this is used to achieve full customer multihoming, it requires lot of manual co-operation between parties

Case 4: Summary (cont.)

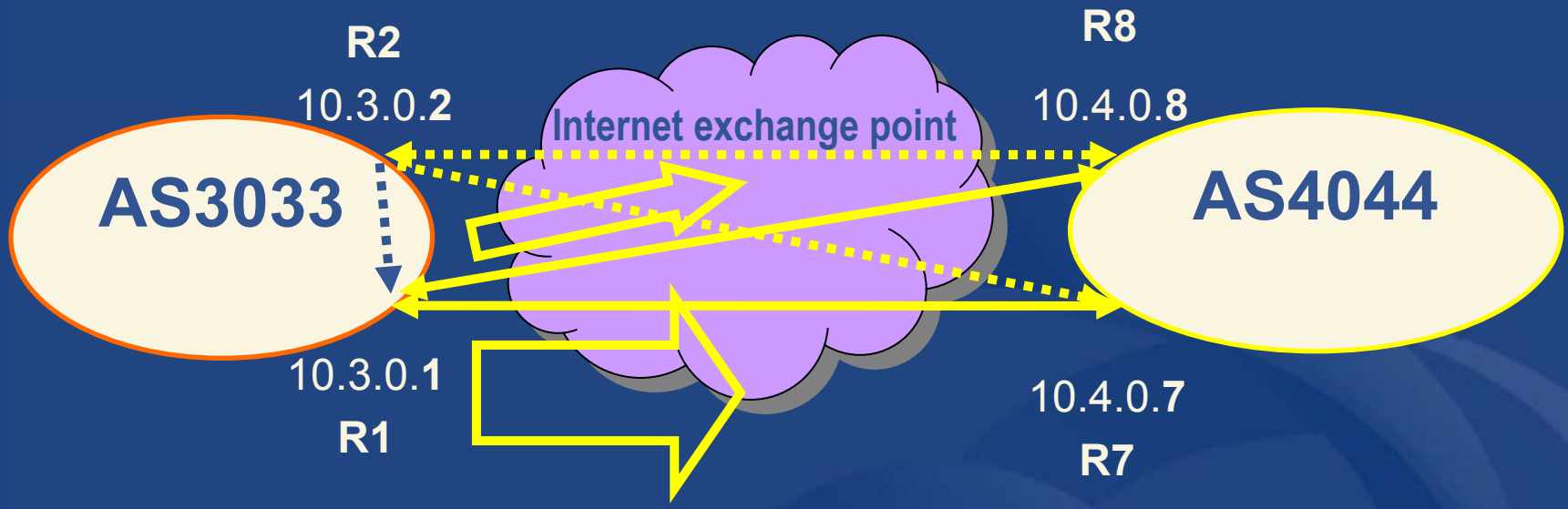
- If you wish to use AS-path filters, the filter rules in aut-num should use regular expressions.
  - Slide 166: Case 4: BGP with peers – AS3001 view, review db objects

Case 5: Multiple links to the same neighbour

- Two (or more) routers from one AS connected to one (or more) routers in the neighbouring AS
  - Either over IXP, or directly
  - Providing link redundancy
  - Can be full mesh, or any combination
- Aut-num: both link-end router IPs must be specified
  - Multiple import/export lines per peer (one pair per rtr IP)

  Import: from ANS <remote-rtr-IP> at <your-rtr-IP> <action> accept <filter>
- Separate RtConfig file is needed for each router

Case 5.1: Controlling outbound traffic



- By using local preference in import attribute, AS3033 can influence the choice between two links that AS3033 has on R1 to AS4044
  - Link R1-R7 is preferred
    - ="pref" has to be lower

Case 5.1: AS3033 aut-num object for Router 1

Aut-num: AS3033
Import: from AS4044 10.4.0.7 at 10.3.0.1 action pref=10;
accept AS4044
Import: from AS4044 10.4.0.8 at 10.3.0.1 action pref=20;
accept AS4044

Case 5.1: RtConfig command for R1

- One RtConfig file is needed for each router

```
@RtConfig set cisco_map_name = "AS%d-IMPORT-%d"
@RtConfig import AS3033 10.3.0.1 AS4044 10.4.0.7
!
@RtConfig set cisco_map_name = "AS%d-IMPORT-%d"
@RtConfig import AS3033 10.3.0.1 AS4044 10.4.0.8
```

- Note: additional formatting of the “map name”
  - First %d replaced by peer’s ASN
  - Second %d incremented
  - See slide 122 “RtConfig Cisco specific commands”

Case 5.1: RtConfig output for R1 setup (import)

Incremented number

To differentiate “map name”

```
no ip prefix-list pl100
ip prefix-list pl100 permit 10.4.192.0/19
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS4044-IMPORT-1
!
route-map AS4044-IMPORT-1 permit 1
match ip address prefix-list pl100
set local-preference 90
!
router bgp 3033
neighbor 10.4.0.7 route-map AS4044-IMPORT-1 in
!
!
no route-map AS4044-IMPORT-2
!
route-map AS4044-IMPORT-2 permit 1
match ip address prefix-list pl100
set local-preference 80
!
router bgp 3033
neighbor 10.4.0.8 route-map AS4044-IMPORT-2 in
```

Case 5.1: AS3033 aut-num object for Router 2

- If R2 also has two links to AS4044, and R1-R2 do iBGP
  - E.g. if R2 should also use R1 to reach AS4044
    - Preferences on R2 need ti be much “higher”

Aut-num: AS3033
Import: from AS4044 10.4.0.7 at 10.3.0.2 action pref=72; accept AS4044
Import: from AS4044 10.4.0.8 at 10.3.0.2 action pref=82; accept AS4044

Aut-num: AS3033
Import: from AS4044 10.4.0.7 at 10.3.0.1 action pref=10; accept AS4044
Import: from AS4044 10.4.0.8 at 10.3.0.1 action pref=20; accept AS4044

Case 5.1: RtConfig command for R2

- One RtConfig file is needed for each router

```
! Router 2!!!
!
! Setting max pref value to 100
@RtConfig set cisco_max_preference = 100
!
@RtConfig set cisco_map_name = "AS%d-IMPORT-%d"
@RtConfig import AS3033 10.3.0.2 AS4044 10.4.0.7
!
@RtConfig set cisco_map_name = "AS%d-IMPORT-%d"
@RtConfig import AS3033 10.3.0.2 AS4044 10.4.0.8
```

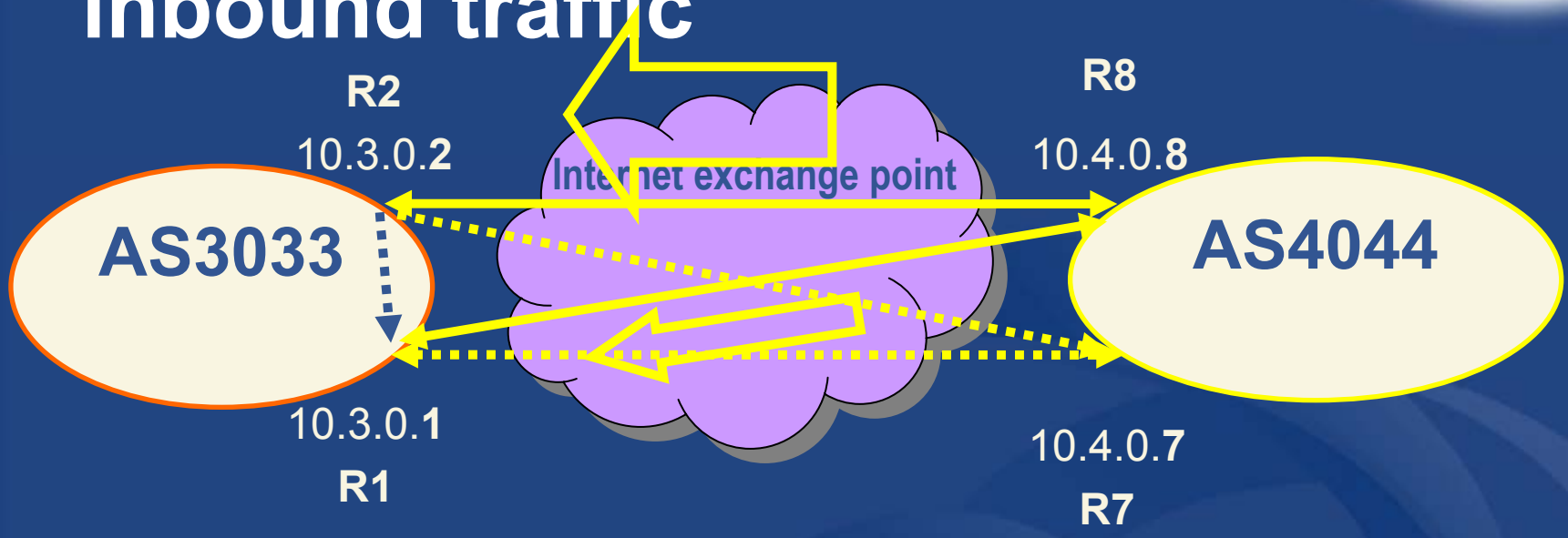
Case 5.1: RtConfig output for R2 setup (import)

```
no ip prefix-list pl100
ip prefix-list pl100 permit 10.4.192.0/19
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS4044-IMPORT-1
!
route-map AS4044-IMPORT-1 permit 1
match ip address prefix-list pl100
set local-preference 28
!
router bgp 3033
neighbor 10.4.0.7 route-map AS4044-IMPORT-1 in
!
!
no route-map AS4044-IMPORT-2
!
route-map AS4044-IMPORT-2 permit 1
match ip address prefix-list pl100
set local-preference 18
!
router bgp 3033
neighbor 10.4.0.8 route-map AS4044-IMPORT-2 in
```

Case 5.1: R2 configuration variation

- E.g. if R2 should send traffic to R7 directly,
  - It is also possible to specify appropriate pref values.

Case 5.2: Controlling inbound traffic



- By setting the value of MED on export attribute, the preferred entry point into your AS can be controlled.
  - If two routers in your AS are connected to the same router in the neighbouring AS
  - E.g. AS3033 can influence the choice between two links that R8 (AS4044) has to the R1 & R2 (AS3033)

Case 5.2: AS3033 aut-num object

```
Aut-num: AS3033
export: to AS4044 at 10.3.0.1 action med=2000; announce
        AS3033
export: to AS4044 at 10.3.0.2 action med=1000; announce
        AS3033
```

- Bigger MED = less preferred
- Smaller MED = more preferred

Case 5.2: Controlling inbound traffic

- Since the local preferences has precedence over MED,
  - AS4044 must agree to honour AS3033's MED values
  - and should not set AS4044's own local-pref values.
  - Instead of MED, possible to use as-path prepend on less preferred link.
- Used for load sharing / balancing, redundancy etc.
  - E.g. action med = igp_cost (see "RPSL: Syntax and semantics, slide #12)
- One RtConfig template file is needed for each router

Case 5.2: RtConfig command for R1

- One RtConfig file is needed for each router

```
! Router1
!
@RtConfig set cisco_map_name = "AS%d-EXPORT-%d"
@RtConfig export AS3033 10.3.0.1 AS4044 10.4.0.8
!
! Example 2 - if there is a full mesh (second link to the second router, etc)
!
@RtConfig set cisco_map_name = "AS%d-EXPORT-%d"
@RtConfig export AS3033 10.3.0.1 AS4044 10.4.0.7
!
```

Case 5.2 RtConfig output for R1

```
no ip prefix-list pl100
ip prefix-list pl100 permit 10.3.0.0/20
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS4044-EXPORT-1
!
route-map AS4044-EXPORT-1 permit 1
match ip address prefix-list pl100
set metric 1000
!
router bgp 3033
neighbor 10.4.0.8 route-map AS4044-EXPORT-1 out
!
no route-map AS4044-EXPORT-2
!
route-map AS4044-EXPORT-2 permit 1
match ip address prefix-list pl100
set metric 2000
!
router bgp 3033
neighbor 10.4.0.7 route-map AS4044-EXPORT-2 out !
```

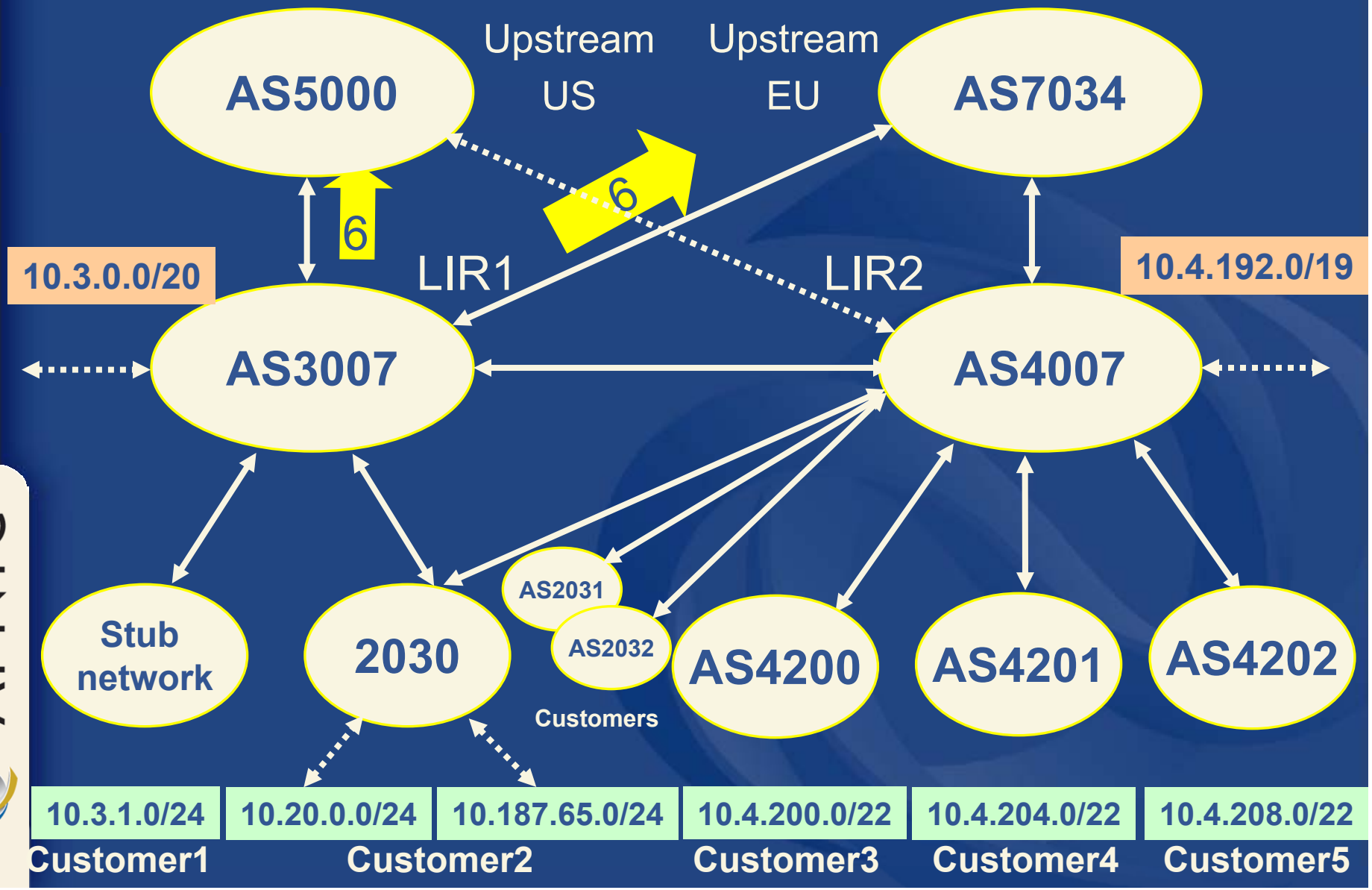
Case 5: Summary

- For each of the multiple connections:
  - A pair of import/export attributes' values in ASN object are required
    - import attributes control outbound traffic, using pref
    - export attributes control inbound traffic, using MED /aspath prepend
  - Separate RtConfig template file for each router is required
    - a pair of import / export commands per each router-IP of each peer

Case 5: Summary (cont.)

- Router-map names need to be unique:
  - E.g. set cisco_map_name = “as%d-EXPORT-%d”
    - First %d replaced by peer’s ANS, second incremented

Case 6: Towards upstreams



Case 6: Using communities

- To allow customers semi-automatically control their own policy:
  - ISP
    - publishes multiple values of user-settable communities
    - filters incoming route announcements to match them
    - and modifies route parameters accordingly
  - Customer
    - marks their announcements with wanted communities
      - Possible to specify multiple values/belong to several categories
- See “Using RPSL in practice” slide #22 and 23
- RFC-1997: BGP Communities Attribute
- RFC-1998: An application of the BGP Community Attribute in Multi-home Routing

Case 6: Example communities – AS3007

3007:20	Multihomed customers, preferred route
3007:30	Multihomed customers, backup route (pref=30, localpref=70) (etc)
3007:440	Only local traffic, community set to “no-export”
3007:112	Pre-pend 2 times to peers
3007:222	Pre-pend 2 times to US upstreams
3007:331	Pre-pend 1 time to EU upstreams
3007:332	Pre-pend 2 times to EU upstreams

The same community definitions for AS4007 too

Example ASN:

AS5400 (BT Ignite European Backbone)

AS3356 (Level3 Communications)

Case 6: Import policy of AS3007

```
# Multihomed customers, backup route
# match community 3007:30, pref=30, localpref=970
import:    from AS3007:AS-BGP-CUSTOMERS action
           pref=30 ;
              accept community.contains (3007:30)
              AND AS3007:AS-BGP-CUSTOMERS;

# Announce only to customers (not to peers)
import:    from AS3007:AS-BGP-CUSTOMERS
              action community = {no_export};
              accept community.contains (3007:440)
              AND AS3007:AS-BGP-CUSTOMERS;
```

Filtering on the community value set by customer

Setting community

There is an implicit logical OR when combining filter rules in aut-num.

So, an explicit "AND" has to be used!

Case 6: Export policy of AS2030

```
export:   to AS4007 action community =
          {4007:20}; announce AS2030
export:    to AS3007 action community =
          {3007:30}; announce AS2030
```

- Customer can specify their route preference by using community value specified by ISP.
  - Set on export attributes

Case 6: Export policy of AS3007

```
# Prepend own 2 times when announcing to peers
# (preferred route is through another ISP)
export:   to AS3007:AS-PEERS
          action aspath.prepend (AS3007,AS3007);
          announce
          community.contains (3007:112)
          AND AS3007:AS-BGP-CUSTOMERS
```

Filtering on the community value

Set by customer

- Note for Cisco configuration
  - Have to manually/explicitly add to the router config file/RtConfig template file:
    - send community

Case 6: Customer setup

- Controlling inbound and outbound traffic using “pref” value and communities
- E.g. AS2030’s scenario for load sharing
  - Provider and its customers through their link
  - US traffic through AS3007
  - EU from the AS4007

Case 6: AS2301 aut-num object

```
import:  from AS3007 accept AS3007:as-customers
         AS3007:as-upstreams
export:  to AS3007
         action community = {3007:20, 3007:331};
         announce AS2031
# prepend AS3007 once to EU peers
```

- Possible to specify multiple values belong to several categories

Case 6: Summary

- Provider:
  - specify communities that your customers can use to influence inbound traffic
  - publish the value in the IRR
  - filter on import/export routes as appropriate
  - set on export routes to other neighbours
- Customer
  - use values specified by ISP
  - set on export routes

Case 7: Various filtering

- Case 7.1: “Martians” filtering
  - Packets with source addresses not routable
    - E.g. packet with private IP addresses
  - RtConfig has built in list of martians
    - can be added automatically to filters by use of command line option
      - RtConfig –supress_martian < output-file-name
    - Based on Bill Manning’s internet draft
      - draft-manning-dsua-04.txt

Case 7.1: “Martians” filtering

```
# rt –supress_martians
@RtConfig set cisco_max_preference = 100
!
@RtConfig set cisco_map_name = "AS%d-IMPORT“
@RtConfig import AS2030 10.20.0.3 AS3007
    10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT“
@RtConfig import AS2030 10.20.0.4 AS4007
    10.4.192.2
```

Case 7.1 RtConfig output

```
no ip prefix-list pl101
ip prefix-list pl101 deny   0.0.0.0/0 ge 32
ip prefix-list pl101 deny   127.0.0.0/8 le 32
ip prefix-list pl101 deny   10.0.0.0/8 le 32
ip prefix-list pl101 deny   172.16.0.0/12 le 32
ip prefix-list pl101 deny   192.168.0.0/16 le 32
ip prefix-list pl101 deny   192.0.2.0/24 le 32
ip prefix-list pl101 deny   128.0.0.0/16 le 32
ip prefix-list pl101 deny   191.255.0.0/16 le 32
ip prefix-list pl101 deny   192.0.0.0/24 le 32
ip prefix-list pl101 deny   223.255.255.0/24 le 32
ip prefix-list pl101 deny   224.0.0.0/3 le 32
ip prefix-list pl101 deny   169.254.0.0/16 le 32
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4007-IMPORT
!
route-map AS4007-IMPORT permit 1
match ip address prefix-list pl101
set local-preference 10
!
router bgp 2030
neighbor 10.4.192.2 route-map AS4007-IMPORT in
 neighbor 10.3.15.2 route-map MyMap_3000_1 in
```

Case 7: Various filtering

- Case 7.2: “Bogon” filtering
  - Informal name for an IP packet that claims to be from an unallocated address space by IANA
  - prefixes that should never appear in the Internet routing table,
  - and obviously should not appear as the source address in any packets you receive

Case 7.2: “Bogon” filtering

aut-num:	AS3004
descr:	AS300x family optimised to use route-set
	and to filter out bogons
Import:	protocol STATIC into BGP4
	from AS3004 action pref = 10;
	accept AS3004:RS-STATIC AND NOT fltr-bogons

Case 7.2: “Bogon Filtering” (cont.)

```
! Setting the max preference to 100
@RtConfig set cisco_max_preference = 100
!
! Setting the map name
@RtConfig set cisco_map_name = "AS%d-STATIC-
    import"
!
@RtConfig static2bgp AS3004 10.3.0.1
```

Output is…

case-1.4-rt-file.txt

Case 7.2: "Bogon Filtering" (cont.)

```
! Setting the max preference to 100
!
! Setting the map name
!
!
no ip prefix-list pl100
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS3004-STATIC-import
!
route-map AS3004-STATIC-import permit 1
 match ip address prefix-list pl100
 set local-preference 90
!
router bgp 3004
 redistribute static route-map AS3004-STATIC-import
```

- Gives empty prefix list – Why?

Case 7.2: “Bogon Filtering” (cont.)

```
route-set:   AS3005:RS-STATIC
descr:       AS300x (family) Static routes
             but for this example - REAL address ranges!!
members:  80.4.0.0/24, 195.168.0.0/22, AS3004:RS-STATIC
```

Case 7.2: “Bogon Filtering” (cont.)

```
! Setting the max preference to 100
@RtConfig set cisco_max_preference = 100
!
! Setting the map name
@RtConfig set cisco_map_name = "AS%d-STATIC-import"
!
@RtConfig static2bgp AS3005 10.3.0.1
!
```

output

```
no ip prefix-list pl100
ip prefix-list pl100 permit 80.4.0.0/24
ip prefix-list pl100 permit 195.168.0.0/22
ip prefix-list pl100 deny 0.0.0.0/0 le 32
```

case-1.5-rt-file.txt

Case 7.2: Example ‘filter-set” objects

filter-set: fltr-bogons
filter: fltr-unallocated OR fltr-martian
descr: All bogon IPv4 prefixes.
remarks: For the complete set of bogons, please see:
fltr-unallocated - unallocated prefixes.
fltr-martian - special use and reserved prefixes.
http://www.cymru.com/Documents/bogon-list.html
admin-c: Rob Thomas RT624
source: RADB

Case 7.2: Example "filter-set" object

```
filter-set: fltr-unallocated
filter:     {1.0.0.0/8^+, 2.0.0.0/8^+, 5.0.0.0/8^+, 7.0.0.0/8^+, 23.0.0.0/8^+, 27.0.0.0/8^+,
        31.0.0.0/8^+, 36.0.0.0/8^+, 37.0.0.0/8^+, 39.0.0.0/8^+, 41.0.0.0/8^+, 42.0.0.0/8^+,
        49.0.0.0/8^+, 50.0.0.0/8^+, 58.0.0.0/8^+, 59.0.0.0/8^+, 70.0.0.0/8^+, 71.0.0.0/8^+,
        72.0.0.0/8^+, 73.0.0.0/8^+, 74.0.0.0/8^+, 75.0.0.0/8^+, 76.0.0.0/8^+, 77.0.0.0/8^+,
        78.0.0.0/8^+, 79.0.0.0/8^+, 83.0.0.0/8^+, 84.0.0.0/8^+, 85.0.0.0/8^+, 86.0.0.0/8^+,
        87.0.0.0/8^+, 88.0.0.0/8^+, 89.0.0.0/8^+, 90.0.0.0/8^+, 91.0.0.0/8^+, 92.0.0.0/8^+,
        93.0.0.0/8^+, 94.0.0.0/8^+, 95.0.0.0/8^+, 96.0.0.0/8^+, 97.0.0.0/8^+, 98.0.0.0/8^+,
        99.0.0.0/8^+, 100.0.0.0/8^+, 101.0.0.0/8^+, 102.0.0.0/8^+, 103.0.0.0/8^+, 104.0.0.0/8^+,
        105.0.0.0/8^+, 106.0.0.0/8^+, 107.0.0.0/8^+, 108.0.0.0/8^+, 109.0.0.0/8^+,
        110.0.0.0/8^+, 111.0.0.0/8^+, 112.0.0.0/8^+, 113.0.0.0/8^+, 114.0.0.0/8^+,
        115.0.0.0/8^+, 116.0.0.0/8^+, 117.0.0.0/8^+, 118.0.0.0/8^+, 119.0.0.0/8^+,
        120.0.0.0/8^+, 121.0.0.0/8^+, 122.0.0.0/8^+, 123.0.0.0/8^+, 124.0.0.0/8^+,
        125.0.0.0/8^+, 126.0.0.0/8^+, 173.0.0.0/8^+, 174.0.0.0/8^+, 175.0.0.0/8^+,
        176.0.0.0/8^+, 177.0.0.0/8^+, 178.0.0.0/8^+, 179.0.0.0/8^+, 180.0.0.0/8^+,
        181.0.0.0/8^+, 182.0.0.0/8^+, 183.0.0.0/8^+, 184.0.0.0/8^+, 185.0.0.0/8^+,
        186.0.0.0/8^+, 187.0.0.0/8^+, 189.0.0.0/8^+, 190.0.0.0/8^+, 197.0.0.0/8^+,
        223.0.0.0/8^+}
descr:      Unallocated (by IANA) IPv4 prefixes.
remarks:    For the complete set of bogons, please see:
            fltr-martian - special use and reserved prefixes.
            fltr-bogons - fltr-unallocated + fltr-martian.
            http://www.cymru.com/Documents/bogon-list.html
changed:    radb@cymru.com 20030606
source:     RADB
```

Secure BGP Template by Rob Thomas

http://www.cymru.com/Documents/secure-bgp-template.html

Case 7: Summary

- RtConfig option to filter “martians”
  - “-supress_martian”
- Filter-set object can be used to maintain filter routers easier
- Always add “AND NOT fltr-bogons” to your import (and even export) line filter rules
  - “Bogon” filter-set maintained by Rob Thomas
  - “fltr-bogons” object is in RADB

Questions ?

Practical usage of the IRR

Overview

- How to put into daily practice all the things learned by now
- Introduction of actual IRR and RtConfig usage by CityLink in NZ
- The rest of the IRRToolSet

Usage: Potential practical problems

- Policy can easily get very complex and result in even more complex router configuration
- Line limit on cisco AS path filters
  - Need to be careful when using as-sets

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

Usage: adding a new neighbour

- Your neighbour needs to:
  - Obtain and register an ASN
  - Create route objects for the new AS
- Automating the process:
  - Add the new AS to (one of) your as-set object(s)
  - Add a set of commands to your master RtConfig template file
  - Run again your scripts/program
    - E.g. Use Make to rebuild RtConfig template file(s)

Usage: simulating policy change

- To avoid the impact of the policy change
  - Simulation before publishing your aut-num is available
    1. Copy the aut-num object into a txt file
    2. Modify the aut-num and save it in the file
    3. Run RtConfig with the flag “-f”
       - E.g. “rt –f my_new+asn.txt < rt-template > new_route_config”
    4. Compare new router config output with the old
       - Or check if the result describes desired behaviour

The rest of the IRRToolSet

Prtraceroute

- Prints the route packets take – including policy information (as registered in IRR)
  - Requires root privileges and access to RR
- Used as diagnostics tool
- Reports in 3 parts:
  - [ASN} inaddr-name (IP) time
  - Traversed ASNs
  - If the hop was within AS, external, preferred (1) or backup (2)

Aoe (Aut-num object editor)

- Displays the aut-num object for the specified AS
- Given a BGP dump from a router inside the AS
  - aoe parses the AS_Path attributes
  - determines the peer ASes
    - By taking the first AS number in the AS_PATH
  - takes the import policies for each peer AS
    - by taking the last AS number in the AS_PATHs that start with the peer's AS number
- Command line options:
  - aoe –h <host> -p 43 –s <source> -protocol ripe <ASN>

Roe (Route object editor)

- GUI lists the routes & dependencies, can add / delete specified routes
- Displays and compares routes registered:
  - by an AS in the IRR (RR)
  - in a BGP routing table
    - NotRtd (not routed) and NotReg (not registered)
- Creates the “route” object for you, based on:
  - BGP dump (local to your ASN)
  - policy in aut-num objects of your peers

Peval

- (Lightweight) policy evaluation tool
- Transforms policy expressions in the matching set of routes (e.g. expands AS numbers)
  - may require connection to RR server
- Handy to compose and check your RPSL filter before putting in into RR server
  - Can be used to write router configuration generators
- Web interface:
  - http://www.ripe.net/cgi-bin/peval.cgi

IRRToolSet: The rest

- prpath – enumerates possible paths between two AS-s, as registered in RR
- CIDRAdvisor – suggests safe aggregates per AS
  - Practical usage:
    - http://www.cidr-report.org/
- rpslcheck – syntax checks objects for IRR
  - RIPE DB has additional restrictions (e.g. hierarchical authorisation)

IRRToolSet: Conclusions

- The quality of data provided by tools strongly depends on the data you have in the RR!
  - Crucial to maintain RR objects up-to-date
- Tools work with both RIPE and IRRd based RRs
- Using the tools will help you to benefit from registering your data in RR, to achieve:
  - Automating access-list generation
  - Avoiding mistakes
  - Improving configuration/operation process
- RPSLng
  - Language extension to include IPv6 and multicast

Questions ?

Thank you!

Reference and acknowledgements

Reference

- RPSL - RFC 2622
  - ftp.rfc-editor.org/in-notes/rfc2662.txt
- Using RPSL in Practice - RFC 2650
  - ftp.rfc-editor.org/in-notes/rfc2650.txt
- RAToolSet
  - ftp://ftp.isi.edu/ra/RAToolSet
- BGP community attribute
  - ftp://ftp.rfc-editor.org/in-notes/rfc1997.txt
- An Application of the BGP Community Attribute in Multi-home Routing
  - ftp://ftp.rfc-editor.org/in-notes/rfc1998.txt
- RADB
  - http://www.merit.edu/radb

Reference (cont.)

- RIPE NCC IRR training material
  - http://www.ripe.net/training/rr/material/rr-june-3.pdf
- Examples used during the training course (created by RIPE NCC)
  - http://www.ripe.net/training/rr/material/examples/

Extra: Further references

- Introduction to RPSL, Ambrose Maagee (NANOG 25, June 2002)
  - http://www.nanog.org/mtg-0206/ppt/ambrose/index.htm
- RPSL Tutorial, Andy Linton (RIPE 43 meeting, Rhodes, September 2002)
  - http://www.ripe.net/ripe/meetings/archive/ripe-43/tutorials/rpsl-tut-ripe43/
- RPSL 101, Mark Prior (January 2001)
  - http://ncne.nlanr.net/training/techs/2001/0128/presentations/200101-prior1/
- BGP Configuration from the IRR, Cengiz Alaettinoglu, Packet Design Inc.
  - http://www.isi.edu/ra/rps/training/tutorial
- Internet Routing Registry, Craig Labovitz (Internet2, December 1999)
  - http://www.ncne.nlanr.net/training/techs/1999/991205/Talks/labovitz_991205_IRR_tutorial/

Extra: Initialise Cisco list parameters

$ RtConfig -cisco_use_prefix_lists -cisco_empty_lists

>RtConfig

{ @RtConfig set cisco_map_first_no = 10
@RtConfig set cisco_map_increment_by = 10
@RtConfig set cisco_prefix_acl_no = 130
@RtConfig set cisco_aspath_acl_no = 130
@RtConfig set cisco_pktfilter_acl_no = 130
@RtConfig set cisco_community_acl_no = 30
@RtConfig set cisco_max_preference = 100 }

Extra: Juniper - “Martians” Filter Access List

```
$ RtConfig -protocol ripe -config junos
RtConfig> @RtConfig access_list filter AS4000

    policy-statement prefix-list-100 {
        term prefixes {
            from {
                route-filter 10.4.192.0/19 exact
   accept;
            }
        }
        term catch-rest {
            then reject;
        }
    }
```

Extra: BGP Path Selection Algorithm

- Ignore the path if there is no route to the next hop
- Prefer the route with the highest weight
  - Local to the router; Cisco specific
- Prefer the route with the highest local preference
  - AS-wide
- Prefer locally originated route
- Prefer the route with the shortest AS path
- Prefer the route with the lowest origin type/code
  - BGP < EGP < incomplete
- Prefer the route with the lowest MED
  - Default: only if the paths are from the same AS
    - Can change with “bgp-always-compare-med”
- Prefer eBGP over iBGP
- Prefer lowest IGP metric to next hop
- Prefer the route with the lowest router ID (if multipath not enabled)
  - Lowest neighbour address

Extra: BGP Route Attributes Summary

- AS-path (mandatory, transitive)
- Next hop (mandatory, non-transitive)
- Local preference (optional, non-transitive)
- Multi-exit Discriminator (mandatory, non-transitive)
- Community (optional, transitive)
  - Well-known: no-export, no-advertise, local-AS

Extra: Registering Your RR in the IRR

http://www.irr.net/docs/list.html

This list of Routing Registries was designed for the Internet Community. It enables users to coordinate their Routing Registry efforts by providing mirroring and contact information.

If you wish to join this list, please fill out this form and send it to <db-admin@radb.net>.

Registry Name (Source):
IP address or DNS name:
Ftp site:
Databases Mirrored:
Mirror Port and Info:
Whois Location:
Type of DB (RPSL?):
Type of Primary Data:
Contact Info:
NOC Info:

Have the RADB be a
mirror for this database? (Y/N)

Extra: RPSLng

- Language extension to include IPv6 and multicast
- Prototype server available:
  - rpslng.ripe.net, port 53001 (RIPE DB snapshot)
  - Updates: rpslng-auto@ripe.net
  - Implementing old draft specs: draft-damas-rpslng-00.txt
- New draft: http://www.radb.net/rpslng.html
- More info: http://www.ripe.net/db/rpslng
- Mailing list:
  - Archives: http://www.ripe.net/ripe/mail-archives/rpslng/
  - Info: http://www.ripe.net/mailman/listinfo/rpslng

Acknowledgments

- Andy Linton
  - Wellington Internet Exchange, NZ
- Terry Manderson
  - Senior System Administrator, APNIC
- Arno Meulenkamp
  - Technical Trainer/Advisor, RIPE NCC
- Katie Petrusha
  - IRRToolSet Project coordinator, RIPE NCC