

# Securing Internet Routing: RPSL

SANOG 28

01 - 09 August, 2016

Mumbai, India

**APNIC**

Issue Date: 14/01/2016

Revision: 1.0



# Fakrul Alam

## Senior Training Officer

Fakrul is responsible for the development and delivery of technical training to the APNIC community and works closely with network operating members in the Asia Pacific region. His specialist training areas include Routing & Switching, Network Architecture, Network Security & Management and Network Forensics.

Prior to joining APNIC, Fakrul worked for several organizations which includes IXP, ISP, Financial Institutes. He has strong knowledge of, and operational experience in building and deploying scalable, reliable network infrastructure.

**email : [fakrul@apnic.net](mailto:fakrul@apnic.net)**



# Bei (Jessica) Wei

Training Officer

After graduating from China's Huazhong University of Science and Technology in 2007 with a degree in electronics engineering, Bei (whose nickname is Jessica) joined Huawei as a network training officer.

Over the next six years, she provided Huawei technical training on LAN/WAN systems, broadband access, IP core and IP mobile backhaul networks as well as working on technical training course design and the development of IP training materials. At the Huawei training center in China she provided technical training to engineers and administrators from more than 15 nations including Vietnam, Papua New Guinea, Thailand, Pakistan and Bangladesh.



**Email: [jwei@apnic.net](mailto:jwei@apnic.net)**

# Target Audience

- Knowledge of Internet Routing(specially BGP)
- Fair idea on Routing Policy
- Familiar with any IRR Database
- No need to know Cryptography
- Basic knowledge of PKI(Public Key Infrastructure)

# Agenda

- BGP 101
- Routing Policy
- RPSL
  - Configuration & Hands on Lab
- RPKI
  - Configuration & Hands on Lab

# AS Path



# AS Path

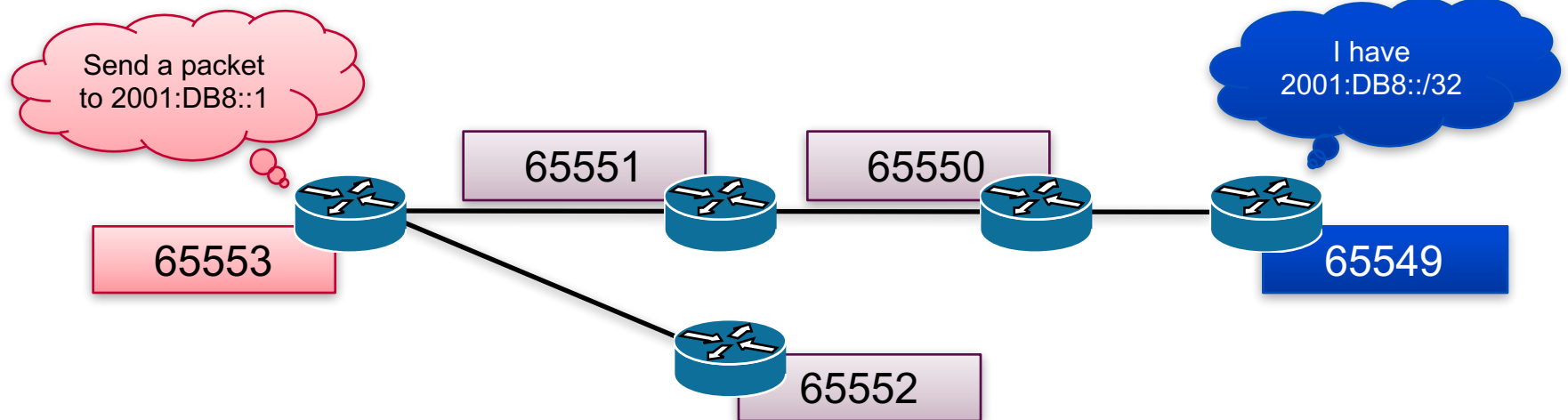
2001:DB8::/32

65551

65550

65549

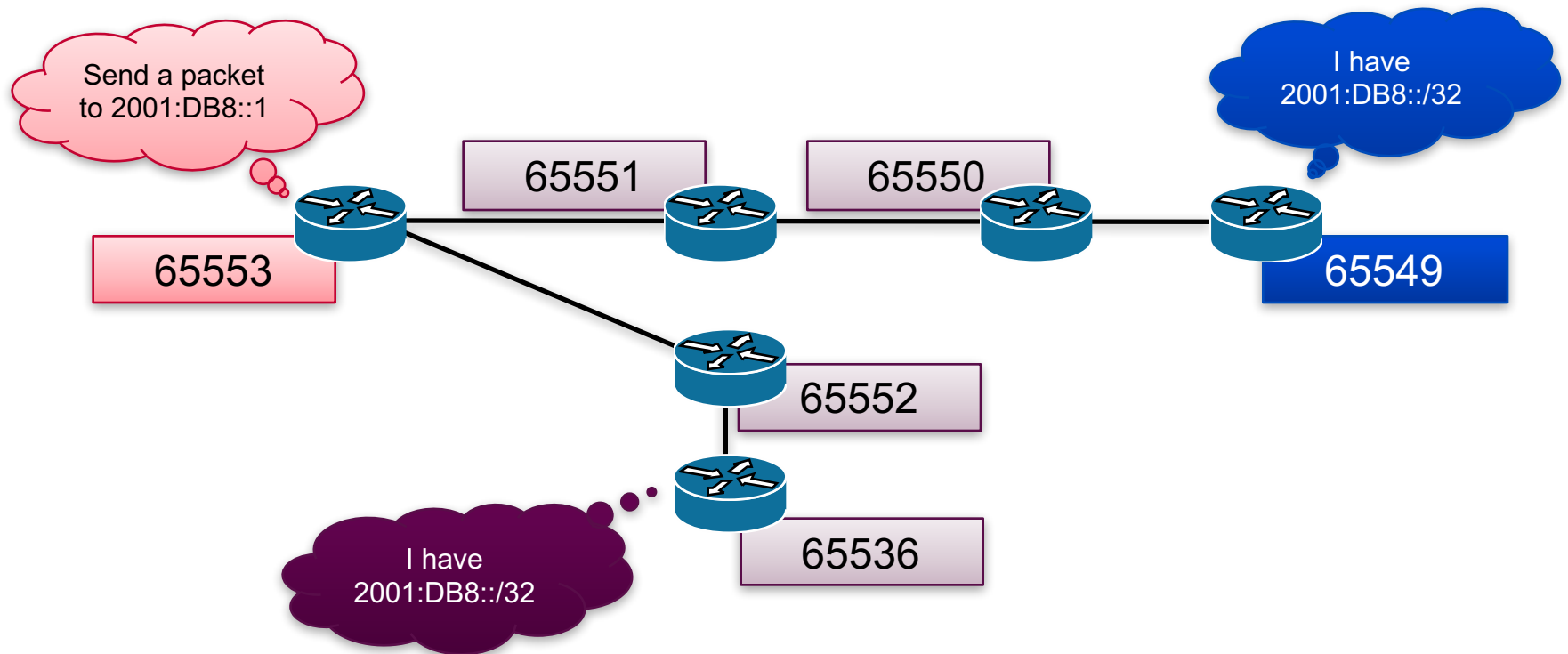
i



# AS Path

2001:DB8::/32	65551	65550	65549	i
---------------	-------	-------	-------	---

2001:DB8::/32	65552	65536	i
---------------	-------	-------	---

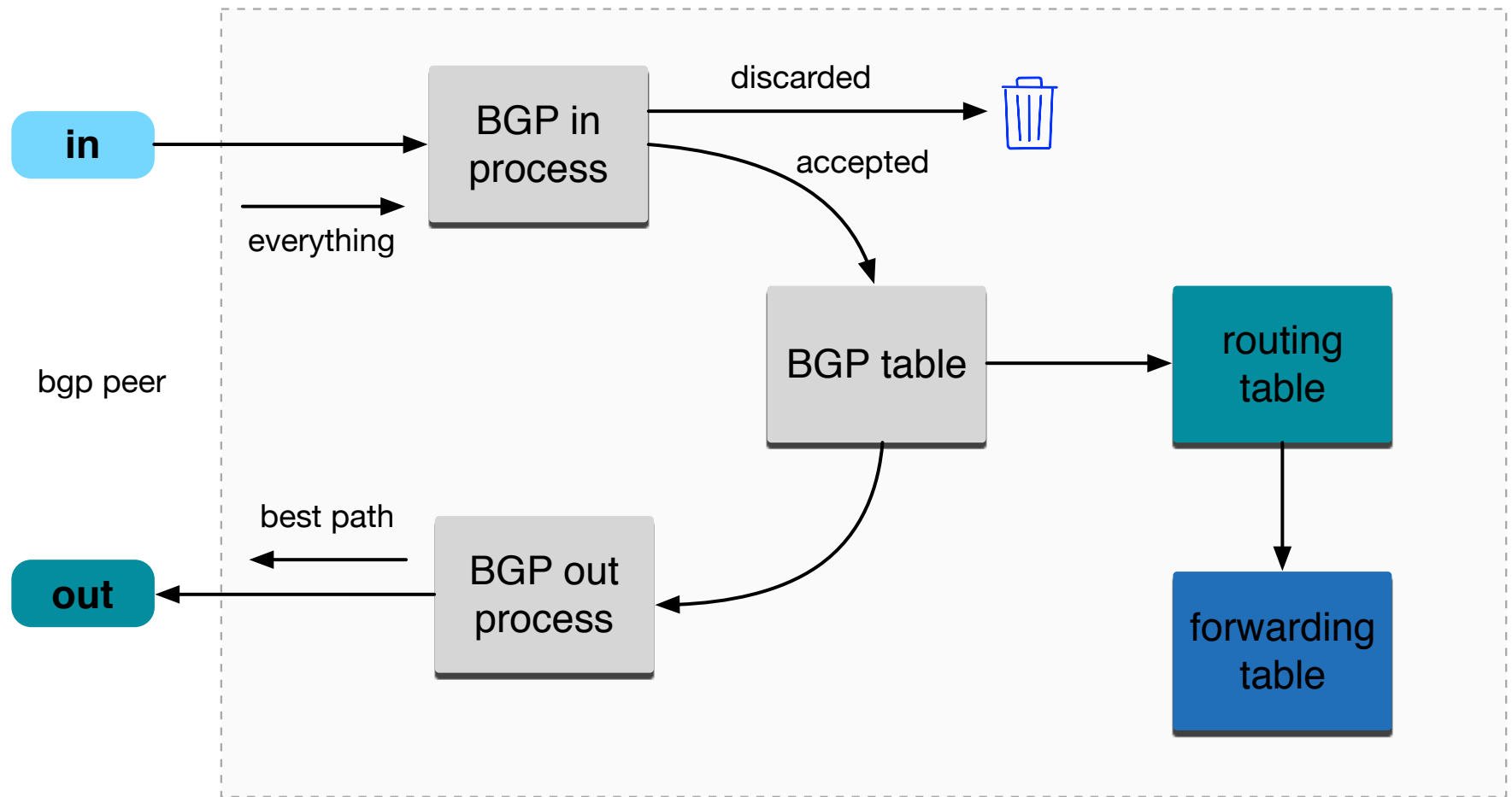




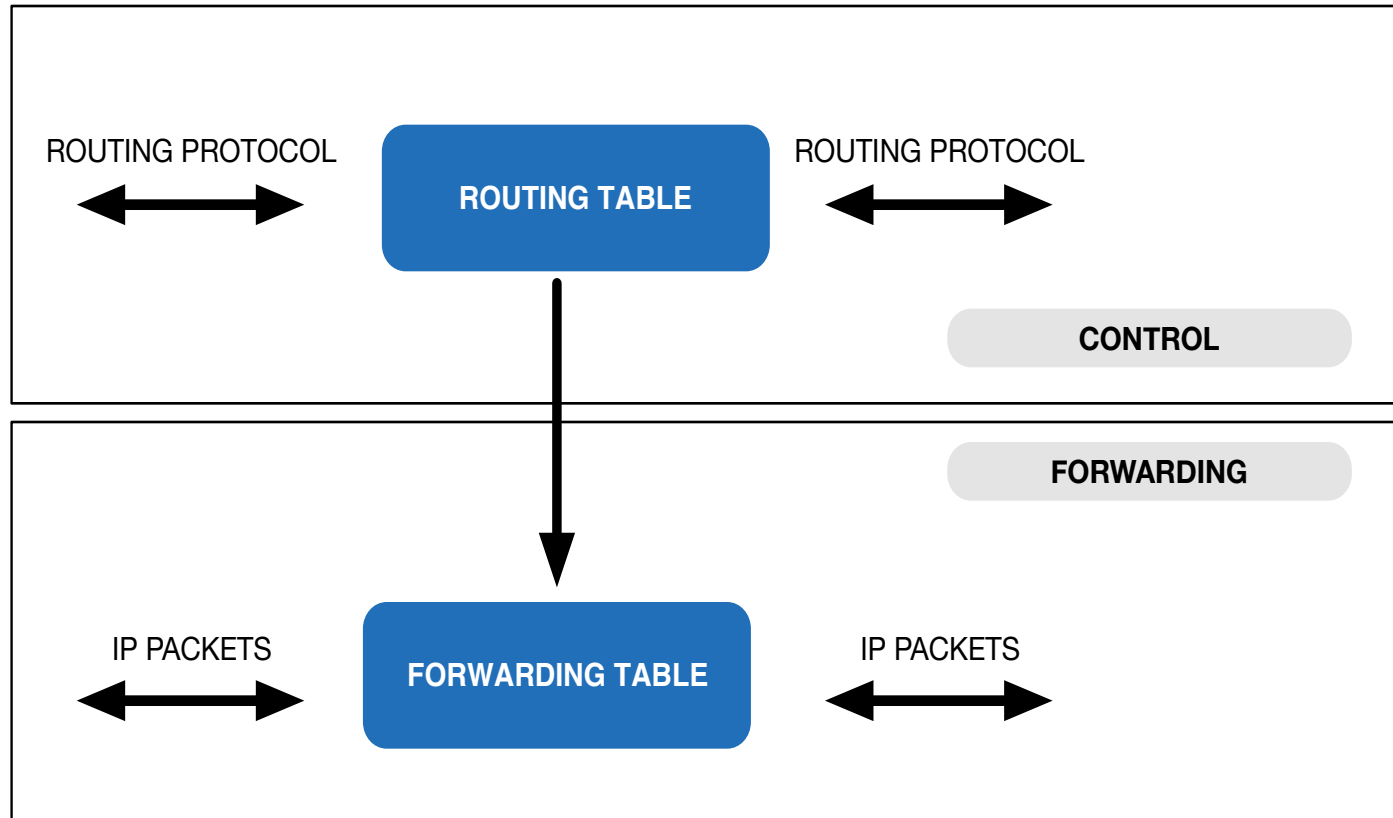
# BGP Best Path Calculation

- Drop if own AS in AS-Path
- Prefer path with highest Weight
- Highest Local Preference
- Shortest AS-Path
- Lowest MED
- Path with shortest next hop metric (minimum IGP cost)
- Oldest received path
- Path from lowest neighbour address

# Constructing the Forwarding Table



# Control Plane and Forwarding Plane



# Routing Incidents Types

- Incidents
  - Misconfiguration
  - Malicious
  - Targeted Traffic Misdirection
- For theory of positivity lets call all these as Mis-Origination
- Traffic Hijacking or Prefix Hijacking assumes Negative intent

# Historical Incident

- April 1997: The "AS 7007 incident" UU/Sprint for 2 days
- February 24, 2008: Pakistan's attempt to block YouTube access within their country takes down YouTube entirely.[6]
- November 11, 2008: The Brazilian ISP CTBC - Companhia de Telecomunicações do Brasil Central leaked their internal table into the global BGP table.
- April 8, 2010: China Telecom originated 37,000 prefixes not belonging to them in 15 minutes, causing massive outage of services globally.
- source : [http://en.wikipedia.org/wiki/IP\\_hijacking](http://en.wikipedia.org/wiki/IP_hijacking)

# Securing Internet Routing

To Secure Internet Routing; we need to check:

**A network should only originate his own prefix**

1. How do we verify?
2. How do we avoid false advertisement?

**A transit network should filter customer prefix**

1. Check customer prefix and ASN delegation
2. Transitive trust

# Secure Internet Routing

Secure Internet Routing

```
graph TD; A[Secure Internet Routing] --- B[Secure Inter-Domain Routing (SIDR) Working Group's model]; A --- C[Routing Policy System (RPS) Working Group's model]
```

Secure Inter-Domain Routing  
(SIDR) Working Group's model

Routing Policy System (RPS)  
Working Group's model

# RPSL & IRR



# Routing Policy

- Public description of the relationship between external BGP peers
- Can also describe internal BGP peer relationship
- Usually registered at an IRR (Internet Routing Registry) such as RADB or APNIC

# Routing Policy

- Who are my BGP peers
- What routes are
  - Originated by a peer
  - Imported from each peer
  - Exported to each peer
  - Preferred when multiple routes exist
- What to do if no route exists

# Why Define a Routing Policy

- Documentation
- Provides routing security
  - Can peer originate the route?
  - Can peer act as transit for the route?
- Allows automatic generation of router configurations
- Provides a debugging aid
  - Compare policy versus reality

# What is RPSL

- Routing Policy Specification Language
- RPSL is object oriented
  - These objects are registered in the Internet Routing Registry (IRR)
  - route, autonomous system, router, contact and set objects
- RIPE-81 was the first language deployed in the Internet for specifying routing policies
  - It was later replaced by RIPE-181
  - RPSL is a replacement for the RIPE-181 or RFC-1786
  - RPSL addresses RIPE-181's limitations

# What is RPSL

- Describes things interesting to routing policy
  - Prefixes
  - AS Numbers
  - Relationships between BGP peers
  - Management responsibility
- For more about RPSL
  - RFC-1786: RIPE-181
  - RFC-2622: Routing Policy Specification Language
  - RFC-2650: Using RPSL in Practice
  - RFC-2726: PGP Authentication for RIPE Database Updates
  - RFC-2725: Routing Policy System Security
  - RFC-2769: Routing Policy System Replication
  - RFC-4012: Routing Policy System Replication next generation

# RPSL Objects

- RPSL objects are similar to RIPE-181 objects
- Objects
  - set of attributes
- Attributes
  - mandatory or optional
  - values: single, list, multiple
- Class “key”
  - set of attributes
  - usually one attribute has the same name as the object’s class
  - uniquely identify each object
- Class “key” = primary key
  - must be specified first

# RPSL Attributes

- Case insensitive
- Value of an attribute has a type
  - <object-name>
  - <as-number>
  - <ipv4-address>
  - <ipv6-address>
  - <address-prefix>
  - etc
- Complete list of attributes and types in RFC 2622
  - <https://www.rfc-editor.org/rfc/rfc2622.txt>

# RPSL Objects Example

**Attribute Name**                      **Attribute Value**

role: ←                      APNIC Training ←

address:                      6 Cordelia Street

address:                      South Brisbane

address:                      QLD 4101

country:                      AU

phone:                      +61 7 3858 3100

fax-no:                      +61 7 3858 3199 ← **Comments**

e-mail:                      training@apnic.net

admin-c:                      NR97-AP

tech-c:                      NR97-AP

nic-hdl:                      AT480-AP

mnt-by:                      MAINT-AU-APNICTRAINING

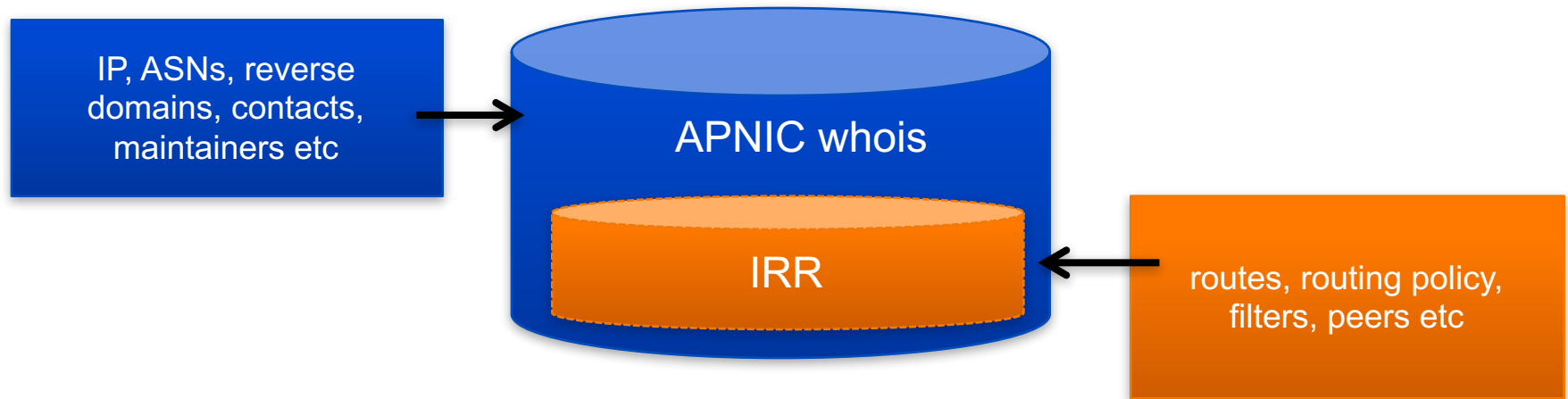
changed:                      hm-changed@apnic.net 20080424

source:                      APNIC



# Integration of whois & IRR

- Integrated APNIC whois database & Internet Routing Registry



Internet Resources & Routing Information

# APNIC Database Objects and Routing Registry

OBJECT	PURPOSE
person	Technical or administrative contacts responsible for an object
role	Technical or administrative contacts represented by a role, performed by one or more people
inetnum	Allocation or assignment of IPv4 address space
inet6num	Allocation or assignment of IPv6 address space
aut-num	Registered holder of an AS number and corresponding routing policy
domain	in-addr.arpa (IPv4) or ip6.arpa (IPv6) reverse DNS delegations
route / route6	Single IPv4/IPv6 route injected into the Internet routing mesh
mntner	Authorized agent to make changes to an object
irt	Dedicated abuse handling team

# person / role Object

- The Person object register contact information

person:	[mandatory]	[single]	[lookup key]
address:	[mandatory]	[multiple]	[ ]
country:	[mandatory]	[single]	[ ]
phone:	[mandatory]	[multiple]	[ ]
fax-no:	[optional]	[multiple]	[ ]
e-mail:	[mandatory]	[multiple]	[lookup key]
nic-hdl:	[mandatory]	[single]	[primary/look-up key]
remarks:	[optional]	[multiple]	[ ]
notify:	[optional]	[multiple]	[inverse key]
abuse-mailbox:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# person / role Object

person: Fakrul Alam  
address: 6 Cordelia Street  
address: South Brisbane  
address: QLD 4101  
country: AU  
phone: +61738583100  
e-mail: fakrul@apnic.net  
**nic-hdl: FA129-AP**  
mnt-by: MAINT-AU-APNICTRAINING  
changed: fakrul@apnic.net 20151217  
source: APNIC

# intenum / inetnum6 Object

- Contains details of an allocation or assignment of IPv4/IPv6 address space

inet6num:	[mandatory]	[single]	[primary/lookup key]
netname:	[mandatory]	[single]	[lookup key]
descr:	[mandatory]	[multiple]	[ ]
country:	[mandatory]	[multiple]	[ ]
geoloc:	[optional]	[single]	[ ]
language:	[optional]	[multiple]	[ ]
admin-c:	[mandatory]	[multiple]	[inverse key]
tech-c:	[mandatory]	[multiple]	[inverse key]
status:	[mandatory]	[single]	[ ]
remarks:	[optional]	[multiple]	[ ]
notify:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
mnt-routes:	[optional]	[multiple]	[inverse key]
mnt-irt:	[mandatory]	[single]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# intenum / inetnum6 Object

```
inet6num:      2406:6400::/32
netname:       APNIC-TRAININGIPv6-Lab-AP
descr:        APNIC TRAINING Lab
country:       AU
admin-c:       AT480-AP
tech-c:        AT480-AP
mnt-by:        APNIC-HM
mnt-lower:     MAINT-AU-APNICTRAINING
mnt-routes:    MAINT-AU-APNICTRAINING
status:        ALLOCATED PORTABLE
remarks:       -+-+-+
remarks:       To report network abuse, please contact the IRT
remarks:       For troubleshooting, please contact tech-c and admin-c
remarks:       For assistance, please contact the APNIC Helpdesk
remarks:       -+-+-+
source:        APNIC
mnt-irt:       IRT-APNICTRAINING-AU
changed:       hm-changed@apnic.net 20100216
changed:       hm-changed@apnic.net 20100818
```

# mntner Object

- Maintainer objects used for authentication
  - Multiple auth / mnt-by / mntner-s are OR-ed

mntner:	[mandatory]	[single]	[primary/lookup key]
descr:	[mandatory]	[multiple]	[ ]
country:	[optional]	[single]	[ ]
admin-c:	[mandatory]	[multiple]	[inverse key]
tech-c:	[optional]	[multiple]	[inverse key]
upd-to:	[mandatory]	[multiple]	[inverse key]
mnt-nfy:	[optional]	[multiple]	[inverse key]
auth:	[mandatory]	[multiple]	[inverse key]
remarks:	[optional]	[multiple]	[ ]
notify:	[optional]	[multiple]	[inverse key]
abuse-mailbox:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
referral-by:	[mandatory]	[single]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# mntner Object Example

```
mntner:          MAINT-AU-APNICTRAINING  
descr:           APNIC Training  
country:         AU  
admin-c:         NR97-AP  
tech-c:          NR97-AP  
auth:            # Filtered  
mnt-by:          MAINT-AU-APNICTRAINING  
upd-to:          nurul@apnic.net  
referral-by:     APNIC-HM  
changed:         hm-changed@apnic.net 20131129  
source:          APNIC
```



# Hierarchical Authorization

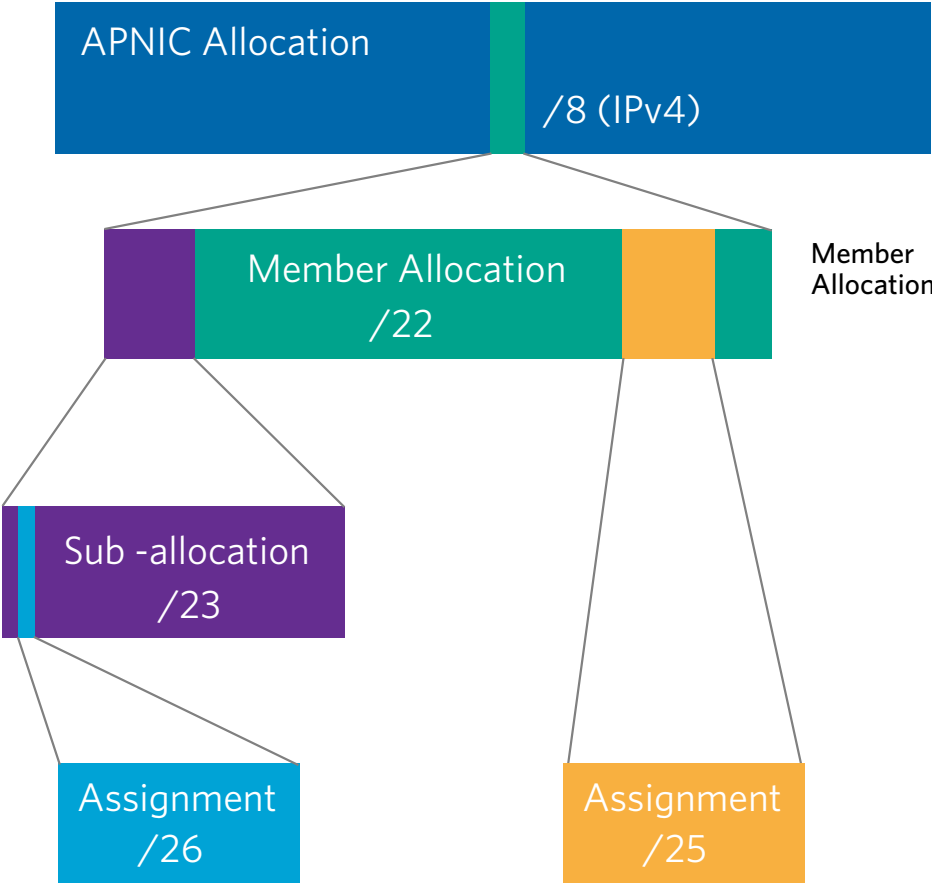
- 'mnt-by' attribute
  - Refers to mntner object
  - Can be used to protect any object
  - Changes to protected object must satisfy authentication rules of 'mntner' object
- 'mnt-lower' attribute
  - Also refers to mntner object
  - Hierarchical authorization for inetnumm inetnum6 & domain objects
  - The creation of child objects must satisfy this mntner
  - Protects against unauthorized updates to an allocated range - highly recommended!
- 'mnt-routers' attribute
  - Can be used to control the creation of 'route' objects associated with the address range specified by the inetnum and inet6num objects

# Maintainer Hierarchy Diagram

**Allocated to APNIC:**  
mnt-by can only be changed by IANA

**Allocated to Member:**  
mnt-by can only be changed by APNIC

**Sub-allocated to Customer:**  
mnt-by can only be changed by Member



# Authorisation Mechanism

```
fakrul@www:~$ whois -h whois.apnic.net 2406:6400::/32
```

```
% Information related to '2406:6400::/32'
```

```
inet6num:      2406:6400::/32
netname:       APNIC-TRAININGIPv6-Lab-AP
descr:        APNIC TRAINING Lab
descr:        LEVEL 1, 33 PARK RD
country:      AU
admin-c:      AT480-AP
tech-c:       AT480-AP
mnt-by:       APNIC-HM
mnt-lower:    MAINT-AU-APNICTRAINING
mnt-routes:   MAINT-AU-APNICTRAINING
status:      ALLOCATED PORTABLE
```

1. This object can only be modified by **APNIC-HM**
2. Creation of more specific objects within this range has to pass the authentication of **MAINT-AU-APNICTRAINING**
3. Creation of route objects matching/within this range has to pass the authentication of **MAINT-AU-APNICTRAINING**

# route/route6 Object

- Use CIDR length format
- Specifies origin AS for a route.
- Use both route and origin fields as the primary key

route:	[mandatory]	[single]	[primary/lookup key]
descr:	[mandatory]	[multiple]	[ ]
country:	[optional]	[single]	[ ]
origin:	[mandatory]	[single]	[primary/inverse key]
holes:	[optional]	[multiple]	[ ]
member-of:	[optional]	[multiple]	[inverse key]
inject:	[optional]	[multiple]	[ ]
aggr-mtd:	[optional]	[single]	[ ]
aggr-bndry:	[optional]	[single]	[ ]
export-comps:	[optional]	[single]	[ ]
components:	[optional]	[single]	[ ]
remarks:	[optional]	[multiple]	[ ]
notify:	[optional]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
mnt-routes:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# route/route6 Example

```
route6:          2406:6400::/32
descr:           APNIC Training Lab parent block
country:         AU
origin:          AS17821
notify:          training@apnic.net
mnt-by:          MAINT-AU-APNICTRAINING
changed:         hm-changed@apnic.net 20100818
source:          APNIC
```

# aut-num Object

- Defines routing policy for an AS
- Uses import/mp-import: and export/mp-export: attributes to specify policy
- These define the incoming and outgoing routing announcement relationships
- Can reference other registry objects such as
  - as-sets / route-sets / filter-sets

# aut-num Object

aut-num:	[mandatory]	[single]	[primary/lookup key]
as-name:	[mandatory]	[single]	[ ]
descr:	[mandatory]	[multiple]	[ ]
country:	[mandatory]	[single]	[ ]
member-of:	[optional]	[multiple]	[inverse key]
import:	[optional]	[multiple]	[ ]
export:	[optional]	[multiple]	[ ]
default:	[optional]	[multiple]	[ ]
remarks:	[optional]	[multiple]	[ ]
admin-c:	[mandatory]	[multiple]	[inverse key]
tech-c:	[mandatory]	[multiple]	[inverse key]
notify:	[optional]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
mnt-routes:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
mnt-irt:	[mandatory]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# aut-num Object Example

```
aut-num:          AS17821  
as-name:          APNIC-TRAINING-Lab-AS-AP  
descr:           Two-byte AS number for APNIC Training  
import:          from as4608 accept ANY  
export:          to AS4608 announce AS17821  
admin-c:         AT480-AP  
tech-c:          AT480-AP  
mnt-by:          MAINT-AU-APNICTRAINING  
mnt-routes:      MAINT-AU-APNICTRAINING  
mnt-irt:         IRT-APNICTRAINING-AU  
changed:         hm-changed@apnic.net 20110701  
source:          APNIC
```



# as-set Object

- Collect together Autonomous Systems with shared properties
- Can be used in policy in place of AS
- RPSL has hierarchical names, can reference other as-set's
  - Non-Hierarchical : AS-
  - Hierarchical: <origin-as-number>: AS-CUSTOMERS  
<origin-as-number>: AS-PEERS

# as-set Object

as-set:	[mandatory]	[single]	[primary/lookup key]
descr:	[mandatory]	[multiple]	[ ]
country:	[optional]	[single]	[ ]
members:	[optional]	[multiple]	[ ]
mbrs-by-ref:	[optional]	[multiple]	[inverse key]
remarks:	[optional]	[multiple]	[ ]
tech-c:	[mandatory]	[multiple]	[inverse key]
admin-c:	[mandatory]	[multiple]	[inverse key]
notify:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

# as-set Object Example

```
as-set:           AS-APNICTRAINING
descr:           AS-SET for APNIC Training
tech-c:          AT480-AP
admin-c:         AT480-AP
mnt-by:          MAINT-AU-APNICTRAINING
changed:         fakrul@apnic.net 20151215
members:         AS17821
source:          APNIC
```

# route-set Object

- Defines a set of routes prefixes
- Name must begin with prefix “RS-” or in the format
  - ASNUM:RS-<ORGANIZATION>
- Can reference other route-sets, AS's or as-set's
  - In this case, the route-set will include all route object prefixes which have an origin which matches the AS numbers

# route-set Object

route-set:	[mandatory]	[single]	[primary/lookup key]
descr:	[mandatory]	[multiple]	[ ]
members:	[optional]	[multiple]	[ ]
mp-members:	[optional]	[multiple]	[ ]
mbrs-by-ref:	[optional]	[multiple]	[inverse key]
remarks:	[optional]	[multiple]	[ ]
tech-c:	[mandatory]	[multiple]	[inverse key]
admin-c:	[mandatory]	[multiple]	[inverse key]
notify:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

source : <https://www.rfc-editor.org/rfc/rfc2622.txt>

# route-set Object Example

```
route-set:      RS-APNICTRAINING  
descr:          Routes announced by APNIC Training  
tech-c:         AT480-AP  
admin-c:        AT480-AP  
mnt-by:         MAINT-AU-APNICTRAINING  
changed:        fakrul@apnic.net 20151215  
mp-members:     2406:6400::/32, AS17821  
source:         APNIC
```

# filter-set Object

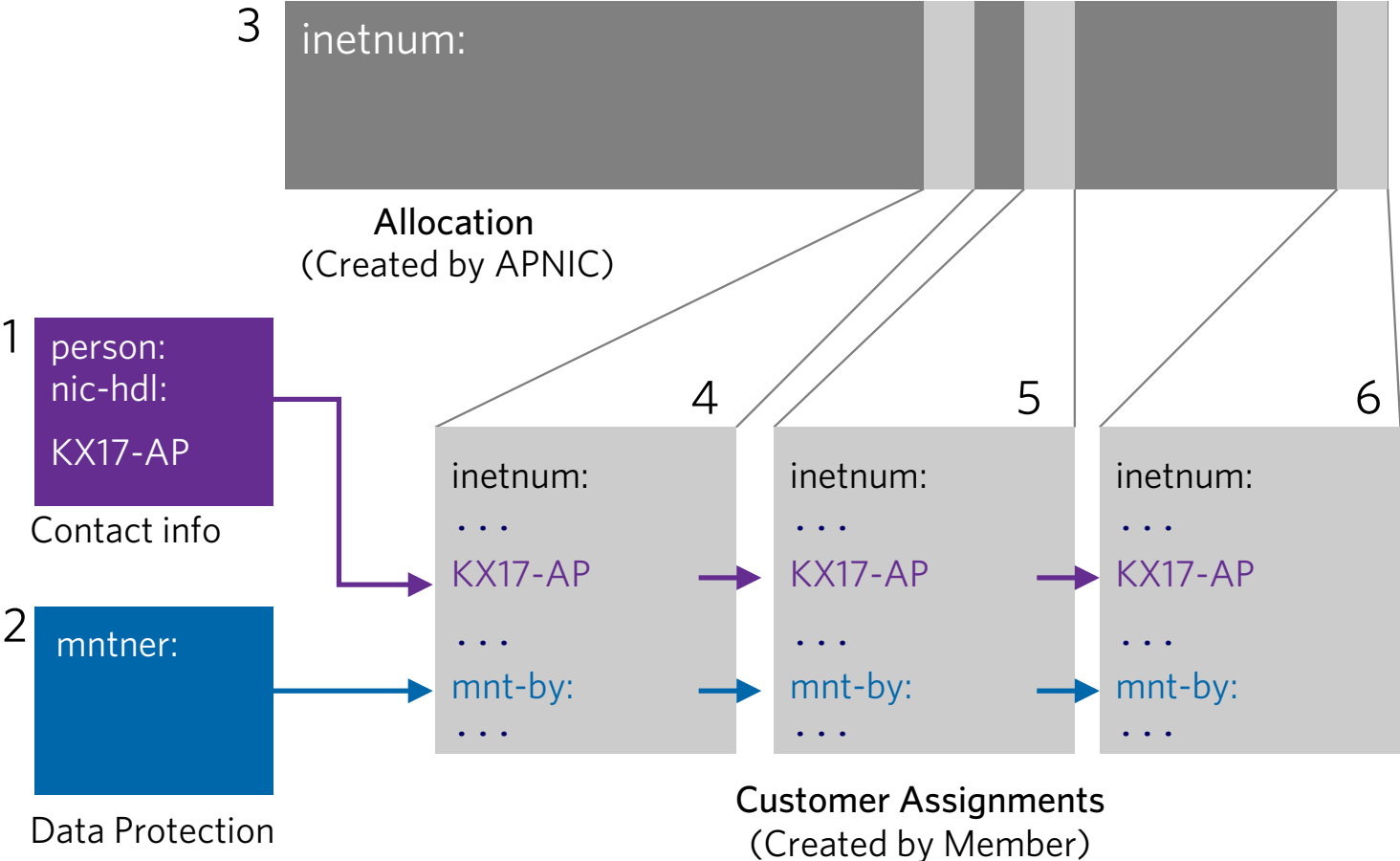
- Defines a set of routes that are matched by a filter expression
- Similar in concept to route-set's
- Name must begin with prefix “fltr-”

# filter-set Object Example

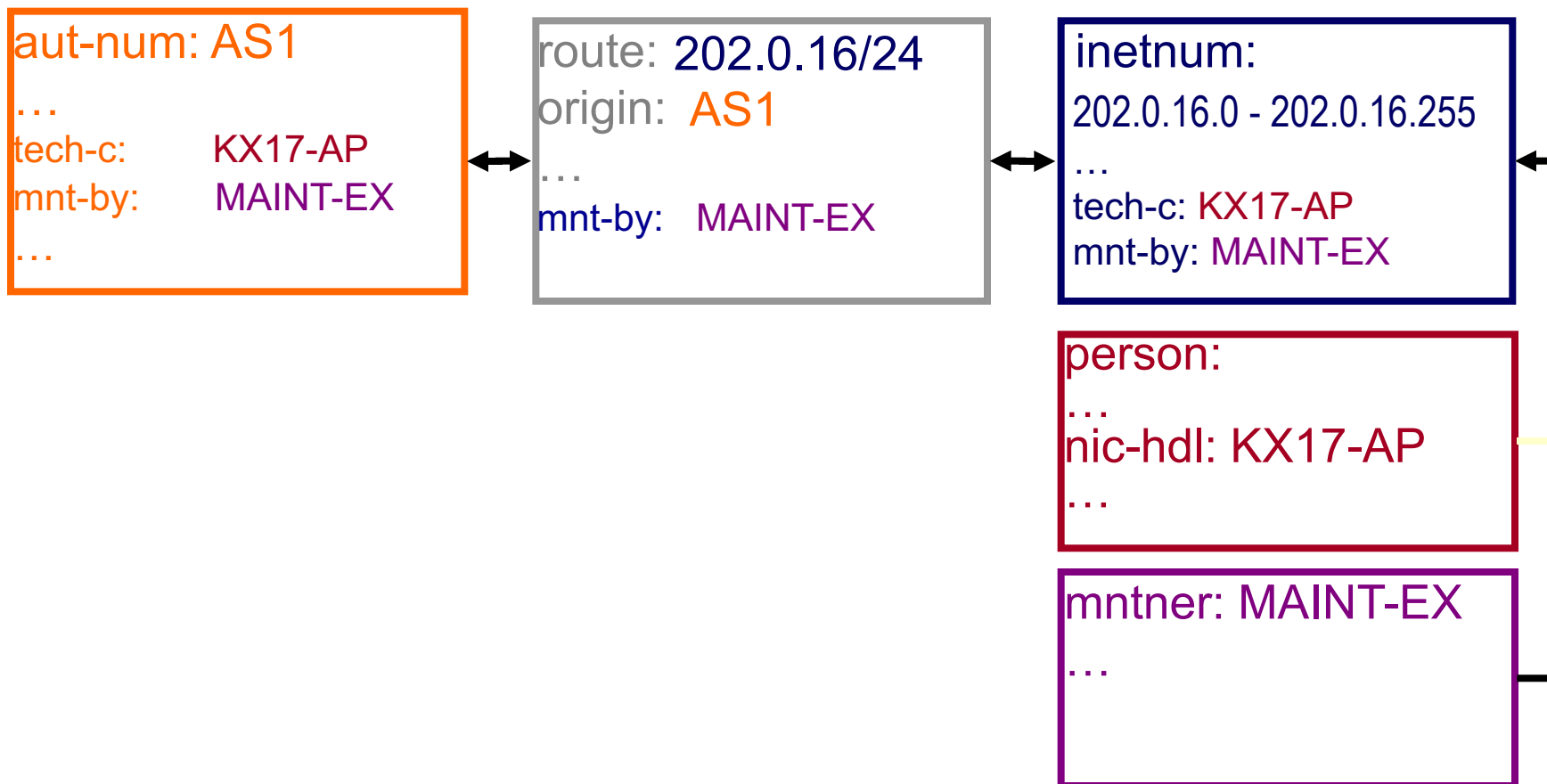
```
filter-set:      fltr-martian-v6
descr:          Current IPv6 MARTIANS
tech-c:         FA129-AP
admin-c:        FA129-AP
mnt-by:         MAINT-AU-APNICTRAINING
changed:        fakrul@apnic.net 20151221
mp-filter:      {
0000::/8^+,      # loopback, unspecified, v4-mapped
0064:ff9b::/96^+, # IPv4-IPv6 Translat. [RFC6052]
0100::/8^+,      # reserved for Discard-Only Address Block [RFC6666]
0200::/7^+,      # Reserved by IETF [RFC4048]
0400::/6^+,      # Reserved by IETF [RFC4291]
0800::/5^+,      # Reserved by IETF [RFC4291]
c000::/3^+,      # Reserved by IETF [RFC4291]
e000::/4^+,      # Reserved by IETF [RFC4291]
f000::/5^+,      # Reserved by IETF [RFC4291]
f800::/6^+,      # Reserved by IETF [RFC4291]
fc00::/7^+,      # Unique Local Unicast [RFC4193]
fe80::/10^+,     # Link Local Unicast [RFC4291]
fec0::/10^+,     # Reserved by IETF [RFC3879]
ff00::/8^+      # Multicast [RFC4291]
}
remarks:        fltr-martian-v6 from RIPE-NCC
remarks:        this object is manually maintained.
source:         APNIC
```



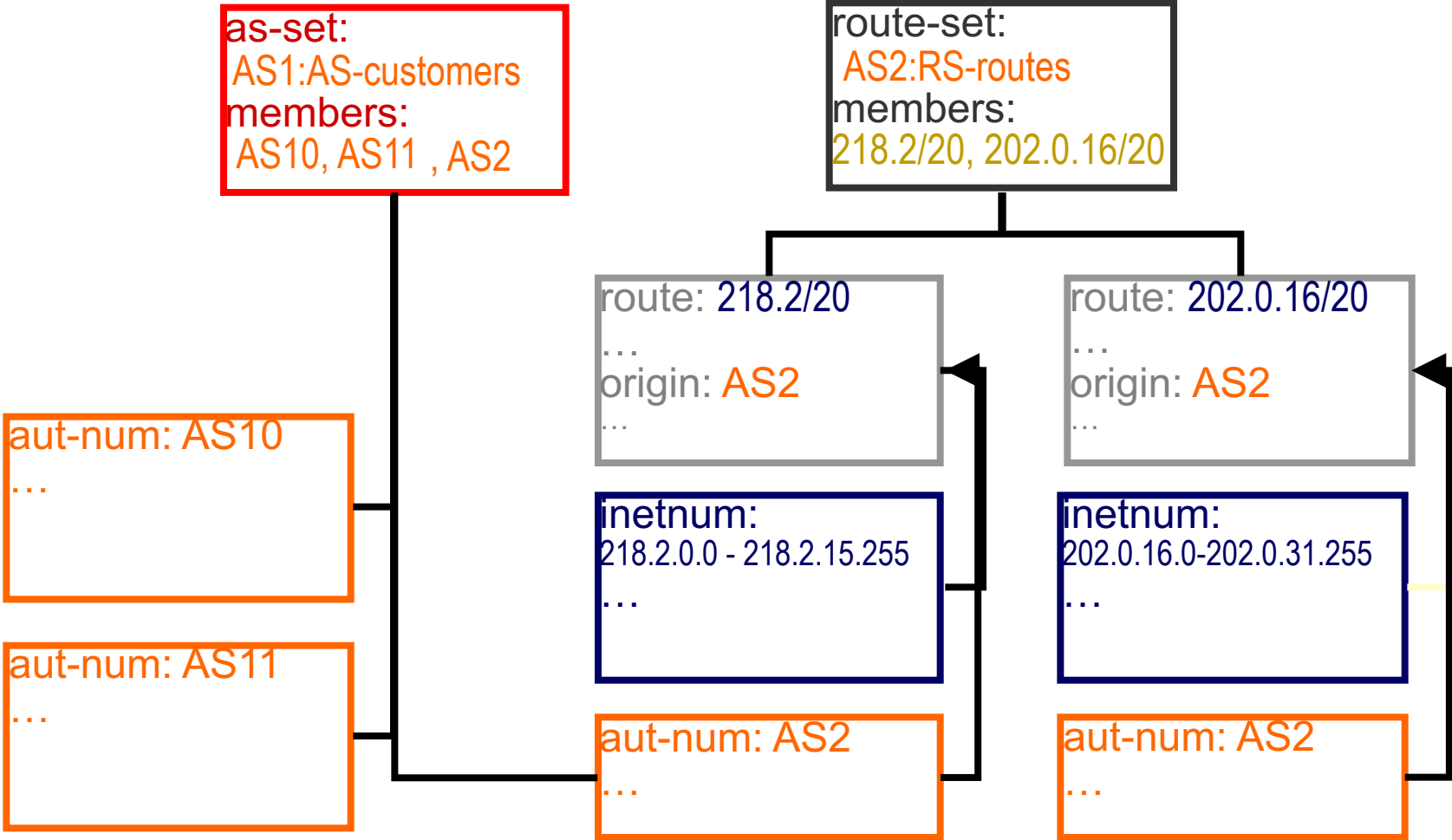
# Relation between objects



# Inter-related IRR Objects



# Inter-related IRR Objects



# RPSL Objects & Routing Policy

# The Internet Routing Registry (IRR)

- Number of public databases that contain routing policy information which mirror each other:
  - APNIC, RIPE, RADB, JPIRR, Level3
  - <http://www.irr.net/>
- 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
- List of Routing Registry
  - <http://www.irr.net/docs/list.html>

# The Internet Routing Registry (IRR)

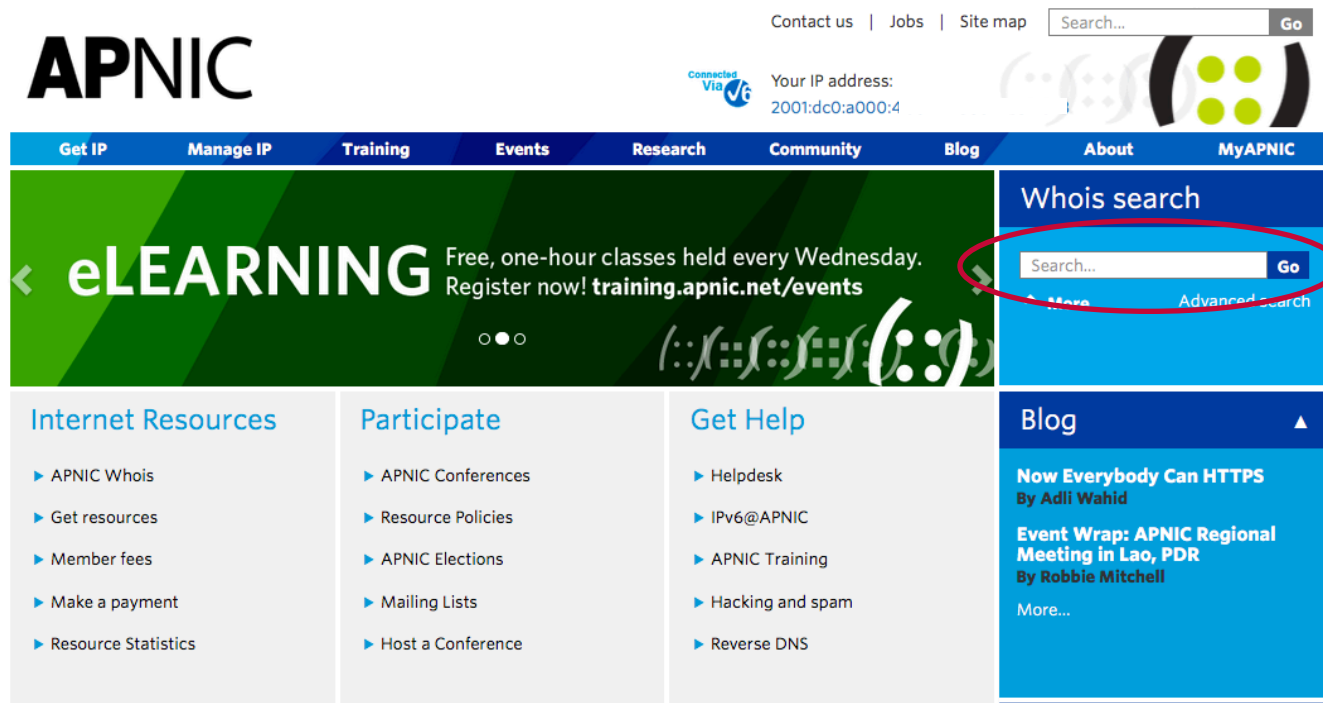
- IRRs are used in at least three distinct ways
  - To publish your own routing intentions
  - To construct and maintain routing filters and router configurations
  - Diagnostic and information service for more general network management

# IRR Objects Query

- `whois` query from cli

```
whois -h whois.apnic.net 2406:6400::/32
```

- You can search from APNIC website also



The screenshot shows the APNIC website homepage. At the top, there is a navigation bar with links for 'Get IP', 'Manage IP', 'Training', 'Events', 'Research', 'Community', 'Blog', 'About', and 'MyAPNIC'. A search bar is located in the top right corner. Below the navigation bar, there is a large banner for 'eLEARNING' with the text 'Free, one-hour classes held every Wednesday. Register now! [training.apnic.net/events](https://training.apnic.net/events)'. To the right of the banner, there is a 'Whois search' section with a search input field and a 'Go' button, which is circled in red. Below the banner, there are three columns of links: 'Internet Resources' (including APNIC Whois, Get resources, Member fees, Make a payment, and Resource Statistics), 'Participate' (including APNIC Conferences, Resource Policies, APNIC Elections, Mailing Lists, and Host a Conference), and 'Get Help' (including Helpdesk, IPv6@APNIC, APNIC Training, Hacking and spam, and Reverse DNS). On the right side, there is a 'Blog' section with a dropdown arrow and two article titles: 'Now Everybody Can HTTPS By Adli Wahid' and 'Event Wrap: APNIC Regional Meeting in Lao, PDR By Robbie Mitchell'.

# IRR Objects Query Flags

- IRR supports a number of flag option
  - ! RADB Query Flags
  - - RIPE/BIRD Query Flags
- `-i` flags for inverse query

```
whois -h whois.apnic.net -i mnt-by MAINT-AU-  
APNICTRAINING
```

[All the objects with a matching **mnt-by** attribute]

```
whois -h whois.apnic.net -i origin as17821
```

[**route** and **route6** objects with a matching **origin** attribute]

- `-q` flag for Informational queries

```
whois -h whois.apnic.net -q sources
```

[list of sources]



# IRR Objects Query Flags

- -K flags for primary keys of an object are returned

```
whois -h whois.apnic.net -K 2406:6400::/32
```

- IRRd (IRR Daemon) supports service side set expansions (as-set and route-set)

```
whois -h whois.radb.net '!iAS-APNICTRAINING'
```

[returns members of AS-APNICTRAINING as-set object]

- For details please check
  - [https://www.apnic.net/apnic-info/whois\\_search/using-whois/searching/query-options](https://www.apnic.net/apnic-info/whois_search/using-whois/searching/query-options)
  - <http://www.radb.net/support/query2.php>

# RPSL Implementation : How to Begin

- Need to identify which IRR to use
  - May want to run your own for control
- Need to decide what degree of filtering is desired
  - Prefix filters
  - AS path filters
  - Both
- Register a maintainer object at chosen IRR
  - Usually a “manual” process and could be multi-stage if PGP key authentication required

# RPSL Implementation : Checklist

1. Define your routing policy
2. Creating the objects in IRR
3. Use automated tools to generate the configuration

# Objects Involved

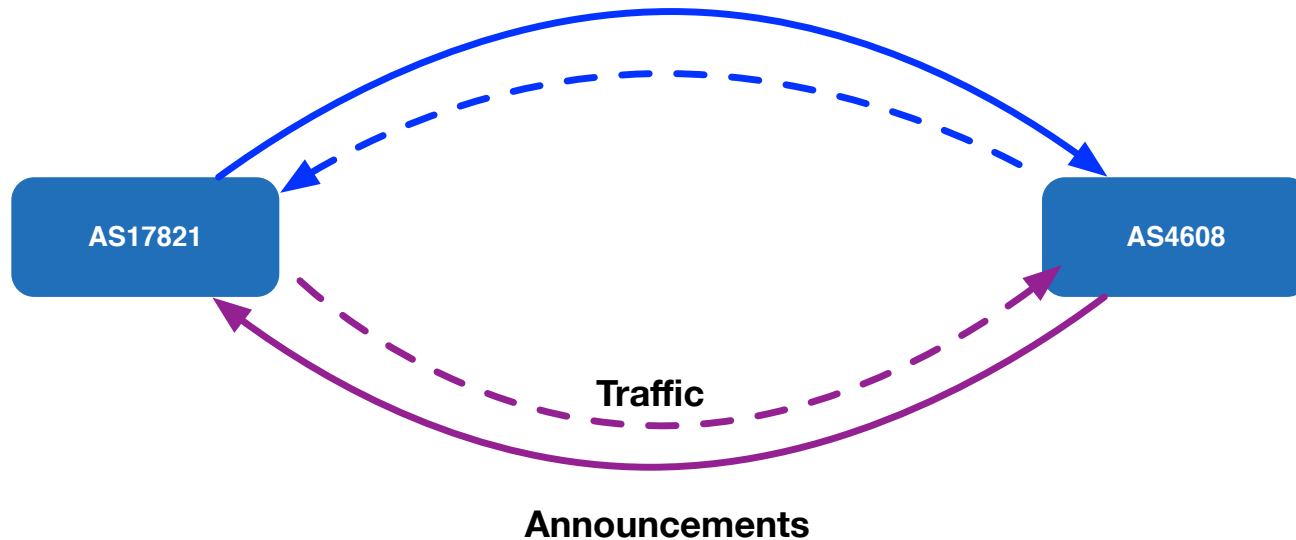
Objects	Functions
route or route6 object	Connects a prefix to an origin AS
aut-num object	Registration record of an AS Number Contains the routing policy
sets	Objects can be grouped in sets, i.e. as-set, route-set
keywords	“ANY” matches every route

# Import and Export Attributes

- You can document your routing policy in your aut-num object in the APNIC Database:
  - Import lines describe what routes you accept from a neighbor and what you do with them
  - Export lines describe which routes you announce to your neighbor

```
aut-num: AS17821
as-name: APNIC-TRAINING-Lab-AS-AP
descr: Two-byte AS number for APNIC Training Lab
country: AU
import: from AS45192 action pref=200; accept ANY
import: from AS4608 action pref=100; accept ANY
export: to AS45192 announce AS17821
export: to AS4608 announce AS17821
default: to AS45192 action pref=50; networks ANY
admin-c: AT480-AP
tech-c: AT480-AP
mnt-by: MAINT-AU-APNICTRAINING
mnt-routes: MAINT-AU-APNICTRAINING
changed: hm-changed@apnic.net 20080424
changed: hm-changed@apnic.net 20100818
changed: hm-changed@apnic.net 20100819
mnt-irt: IRT-APNICTRAINING-AU
changed: hm-changed@apnic.net 20110701
source: APNIC
```

# Route Announcements vs Traffic Direction

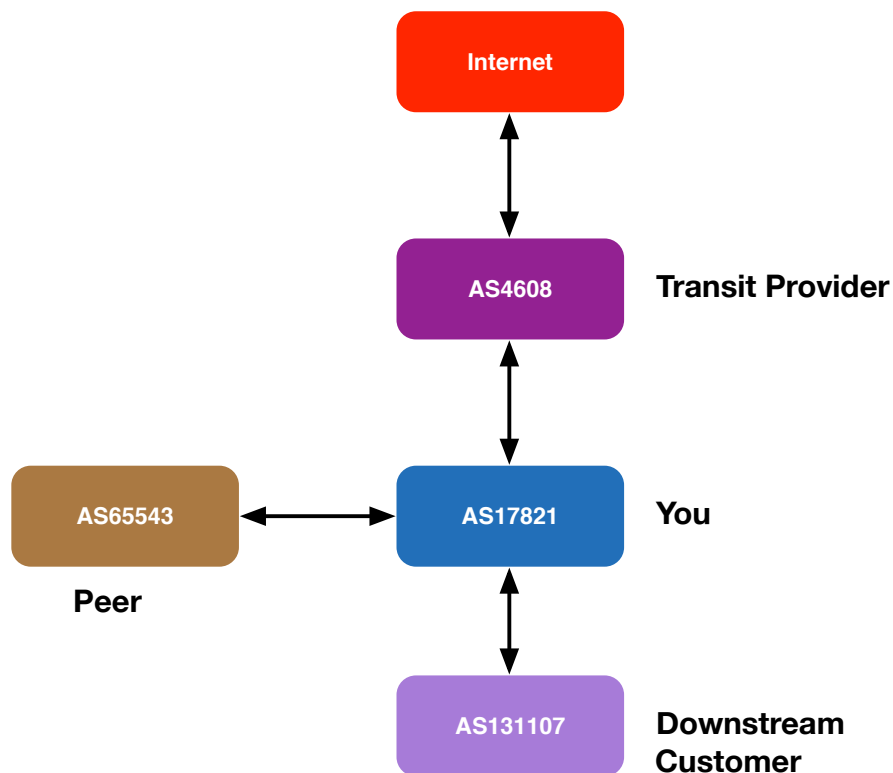


- AS17821 accepting all prefixes from AS4608 so that outbound traffic goes towards AS4608. It also makes localpref to 100
- AS17821 announcing prefixes (originating in AS17821) to AS4608, so that the incoming traffic for AS17821 can flow away from the AS4608

```
aut-num: AS17821
```

```
import: from AS4608 action pref=100; accept ANY  
export: to AS4608 announce AS17821
```

# Routing Policy Scenarios



**aut-num: AS17821**

**import:** from AS4608 accept ANY  
**export:** to AS4608 announce AS17821 AS131107

**import:** from AS131107 accept AS131107  
**export:** to AS131107 announce ANY

**import:** from AS65543 accept AS65543  
**export:** to AS65543 announce AS17821 AS131107

# Building an aut-num Object

- RPSL is older than IPv6, the defaults are IPv4
- IPv6 was added later using a different syntax
  - You have to specify that it's IPv6

```
mp-import: afi ipv6.unicast from AS131107 accept AS131107
mp-export: afi ipv6.unicast to AS131107 announce ANY
```

- More information in RFC 4012 RPSLng



# Filter List : Regular Expression

AS17821	AS 17821
AS17821*	0 or more occurrences of AS17821
AS17821+	1 or more occurrences of AS17821
AS17821?	0 or 1 occurrence of AS17821
&	Beginning of Path
\$	End of Path
\	Escape a regular expression character
_	Beginning, end, white-space, brace
AS17821 AS45192	AS17821 or AS45192
AS17821AS45192	AS17821 followed by AS45192
()	Brackets to contain expression
[]	Brackets to contain numbers

Enclose the expression in “<” and “>”

# Address Prefix Range Operator

Operator	Meanings
$\wedge_-$	Exclusive more specifics of the address prefix: E.g. 128.9.0.0/16 $\wedge_-$ contains all more specifics of 128.9.0.0/16 excluding 128.9.0.0/16
$\wedge_+$	Inclusive more specific of the address prefix: E.g. 5.0.0.0/8 $\wedge_+$ contains all more specifics of 5.0.0.0/8 including 5.0.0.0/8
$\wedge_n$	$n$ = integer, stands for all the length “ $n$ ” specifics of the address prefix: E.g. 30.0.0.0/8 $\wedge_{16}$ contains all the more specifics of 30.0.0.0/8 which are length of 16 such as 30.9.0.0/16
$\wedge_{n-m}$	$m$ = integer, stands for all the length “ $n$ ” to length “ $m$ ” specifics of the address prefix: E.g. 30.0.0.0/8 $\wedge_{24-32}$ 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

# RPSL: localpref / prepend

- Controlling the traffic flow:
  - for outbound traffic set the value of local-pref
    - “action pref=NN” in the “import” lines of aut-num object
    - the lower the “pref”, the more preferred the route
  - for inbound traffic, modify as-path length
    - “action aspath.prepend(ASN)” in the “export” lines
    - Longer the as-path, less preferred the route

Note: the direction of traffic is reverse from accepting / announcing routes

# RPSL: localpref/prepend Example

## Local preference:

```
mp-import:      afi ipv6.unicast from AS65001
2406:6400:10::2 at 2406:6400:10::1 action
community.append(17821:65001); pref=200; accept
<^AS65001+$> AND RS-APNICTRAINING:AS65001
```

Default value is 1000. Setting pref value to 200 mean downgrade the pref value by 200. Local pref will be 800.

## Prepend:

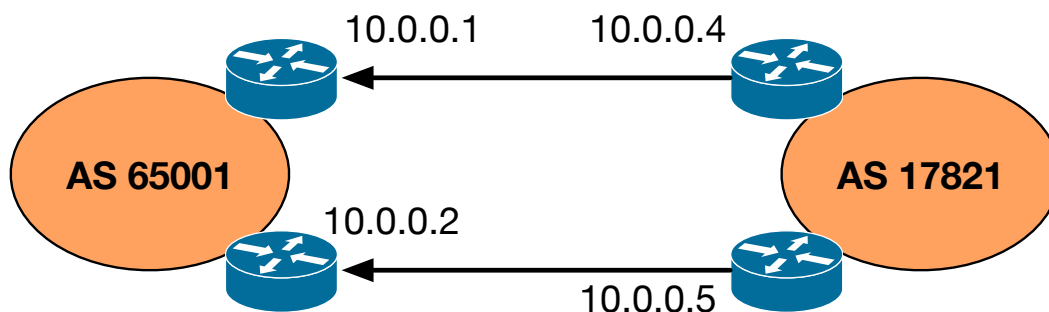
```
mp-export:      afi ipv6.unicast to AS65001 2406:6400:10::2
at 2406:6400:10::1 action aspath.prepend (AS17821,AS17821);
announce ANY AND NOT FLTR-MARTIAN-V6
```

# RPSL: Multiple Links / MED

- By setting the value of MED on export lines, the preferred entry point into your AS can be controlled
- The neighbour must agree to honour your MED values
  - Instead of MED, it is possible to use as-path prepend on less preferred link

# RPSL: MED Example

```
export: to AS17821 10.0.0.4 at 10.0.0.1 action med=1000;  
announce AS65001  
export: to AS17821 10.0.0.5 at 10.0.0.2 action med=2000;  
announce AS65001
```



# RPSL: BGP Communities

- Elegant solution for implementing policies
- Optional tags
  - Can go through many peers
- Can be used for advanced filtering
- Enables customers to control their own routing policy
  - Publish your communities, and what you do with them
  - Filter incoming announcements accordingly

# RPSL: BGP Communities Example

```
mp-import:      afi ipv6.unicast from AS65001
2406:6400:10::2 at 2406:6400:10::1 action
community.append(17821:65001); pref=200; accept
<^AS65001+$> AND RS-APNICTRAINING:AS65001
```



# RPSL Tools

- IRRToolkit (written in C++)
  - <https://github.com/irrtoolset/irrtoolset/>
- Rpsltool (perl, using Template::Toolkit)
  - <http://www.linux.it/~md/software>
- IRR Power Tools (PHP)
  - <http://sourceforge.net/projects/irrpt/>
- BGPQ3 (C)
  - <http://snar.spb.ru/prog/bgpq3/>
- Filtergen (Level 3)
  - Online tool using whois protocol
  - `whois -h filtergen.level3.net RIPE::ASxxxx`

# RPSL Tools

Tool	Advantages	Disadvantages
IRRToolSet	<ul style="list-style-type: none"> <li>• Full RPSL support</li> <li>• RPSLNg support</li> <li>• 32-bit ASN support</li> <li>• Full BGP config generation</li> </ul>	<ul style="list-style-type: none"> <li>• No AS-Set query support</li> <li>• Manual peering configuration on the fly</li> <li>• Difficult to understand</li> </ul>
IRR Power Tools	<ul style="list-style-type: none"> <li>• Route aggregation</li> <li>• AS-SET queries</li> </ul>	<ul style="list-style-type: none"> <li>• No RPSLNg support</li> <li>• No 32-bit ASN support</li> </ul>
BGPq3	<ul style="list-style-type: none"> <li>• RPSL support</li> <li>• RPSLNg support</li> <li>• 32-bit ASN</li> <li>• AS-SET queries</li> <li>• Easy to use</li> </ul>	<ul style="list-style-type: none"> <li>• Only partial BGP configuration. Can't extract policy from IRR</li> </ul>
RPSLtool	<ul style="list-style-type: none"> <li>• 32-bit ASN</li> <li>• AS-SET queries</li> </ul>	<ul style="list-style-type: none"> <li>• No RPSLNg support</li> </ul>
Net::IRR	<ul style="list-style-type: none"> <li>• RPSL and RPSLNg support</li> </ul>	<ul style="list-style-type: none"> <li>• Outdated</li> <li>• Doesn't support community attribute from RPSL data</li> <li>• No AS-SET queries</li> </ul>
Netconfigs	<ul style="list-style-type: none"> <li>• Provides peering analysis</li> <li>• Can generate full configuration based on peering relationship</li> </ul>	<ul style="list-style-type: none"> <li>• Doesn't support RPSLNg</li> <li>• No command line query</li> <li>• Vendor dependent (CISCO)</li> </ul>

Source : Research project on "Automated configuration of BGP on edge routers" by University of Amsterdam; August 14, 2015

# Use of RPSL

- Use RtConfig to generate filters based on information stored in our routing registry
  - Avoid filter errors (typos)
  - Filters consistent with documented policy (need to get policy correct though)
  - Engineers don't need to understand filter rules (it just works :-)
- Some providers have own tools.

# Using RPSL to Configure Routers

- Need to define “policy” for filtering
  - Inbound from customers & peers
  - Outbound to customers & peers
- Need to be aware of shortcomings in router configuration and/or configuration generator
  - Command line length (on cisco this is 512 bytes)
  - Complexity of rules

# Filtering Philosophy

- Inbound
  - Filter customer by prefix and AS path
  - Filter peer by AS path only but don't accept host routes
  - Filter providers for prefixes longer than a /24
  - Don't accept martians from anyone
- Outbound
  - Filter by BGP community, which indicates the class of the prefix (customer, peer, etc)

# Martians

- RtConfig has built in list of martians that can be added automatically to filters by use of command line option
- `-supress_martian` is Deprecated
- Properly maintained martian and bogon lists are visible in both the RIPE and Merit whois servers
- You can use following filter-set from APNIC whois
  - `fltr-martian-v4 / fltr-martian-v6`

# IRRToolSet : Installation

- Dependency (Debian / Ubuntu)

```
# apt-get install build-essential libtool subversion bison  
flex libreadline-dev autoconf automake
```

- Installation

```
# wget ftp://ftp.isc.org/isc/IRRToolSet/IRRToolSet-  
5.0.1/irrtoolset-5.0.1.tar.gz  
# tar -zxvf irrtoolset-5.0.1.tar.gz  
# cd irrtoolset-5.0.1  
# ./configure  
# make  
# make install
```

For details : <https://github.com/irrtoolset/irrtoolset/blob/master/README.md>

# RtConfig Command Line Options

- Defaults to using RADB
  - -h whois.ra.net / whois.radb.net
  - -p 43
  - Default protocol irrd
- For other RIR use protocol bird
  - -protocol bird/ripe
- Defaults to “cisco” style output
  - -config cisco / -config junos
- -s <list of IRR sources>
  - -s APNIC,RADB,RIPE



# RtConfig Syntax

- import / export pair for each link; syntax

```
@RtConfig [import/export] <yourASN> <yourRouterIP>  
<neighbourASN> <neighbourRouterIP>
```

- Takes other command also

```
@RtConfig configureRouter <inet-rtr-name>  
@RtConfig static2bgp <ASN-1> <rtr-1>  
@RtConfig access_list filter <filter>
```

- And many more. But best thing to look

```
man rtconfig
```

# IRRToolSet Cisco Example

```
bash-3.2$ rtconfig -protocol bird -config cisco -h whois.radb.net
```

```
rtconfig> @RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
!  
no ipv6 access-list ipv6-500  
ipv6 access-list ipv6-500 permit 2406:6400:8000::/48 any  
ipv6 access-list ipv6-500 deny any any  
!  
no ip as-path access-list 500  
ip as-path access-list 500 permit ^(_65001)+$
```

**<output truncated>**

```
router bgp 17821  
!  
neighbor 2406:6400:10::2 remote-as 65001  
address-family ipv4  
no neighbor 2406:6400:10::2 activate  
address-family ipv6 unicast  
neighbor 2406:6400:10::2 activate  
neighbor 2406:6400:10::2 route-map AS65001-IN in  
exit
```

# IRRToolSet JunOS Example

```
bash-3.2$ rtconfig -protocol bird -config junos -h whois.radb.net
```

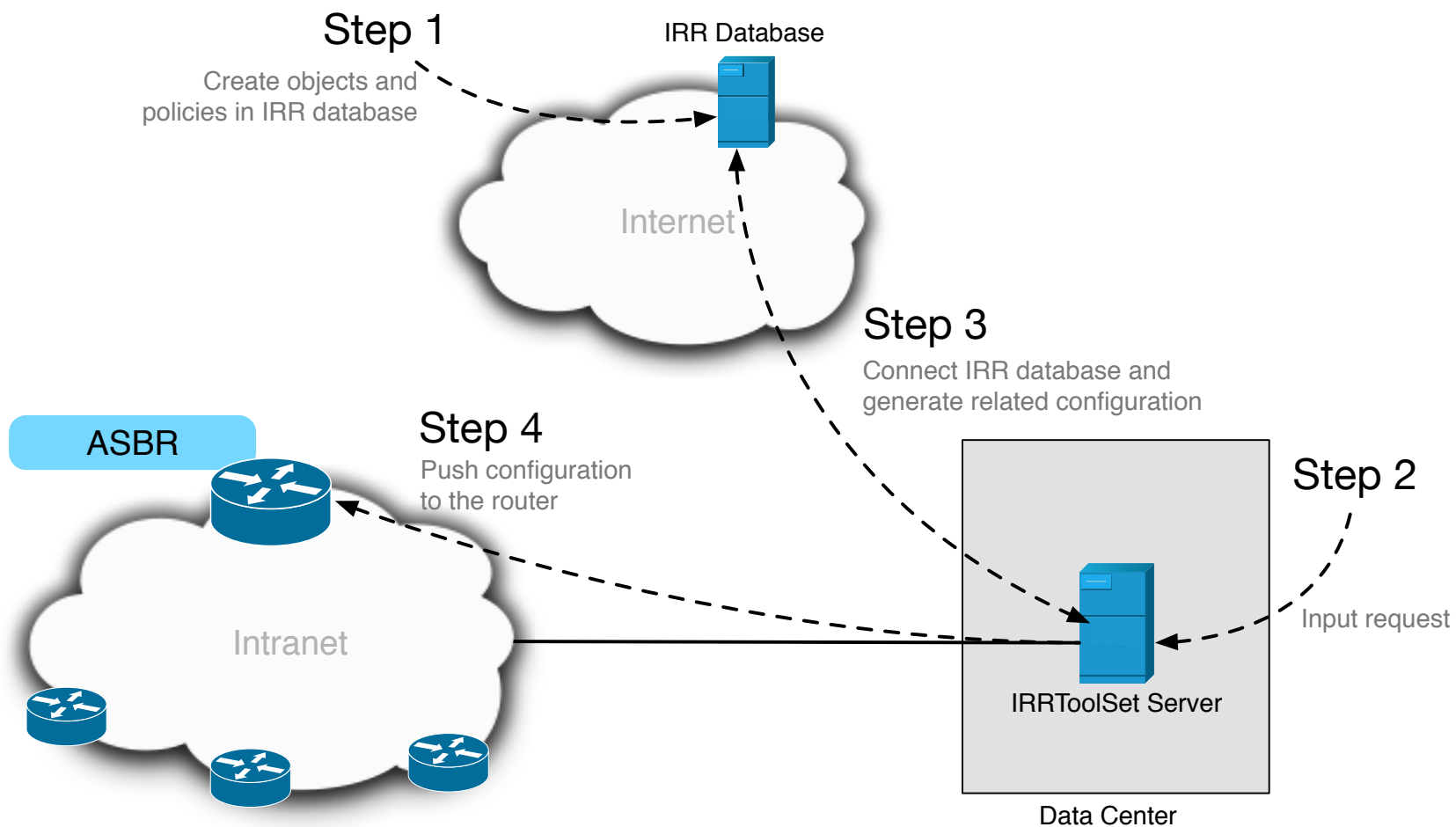
```
rtconfig> @RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
policy-options {
  community community-1 members [17821:65001];
  as-path as-path-1 "( 65001)+";
```

**<output truncated>**

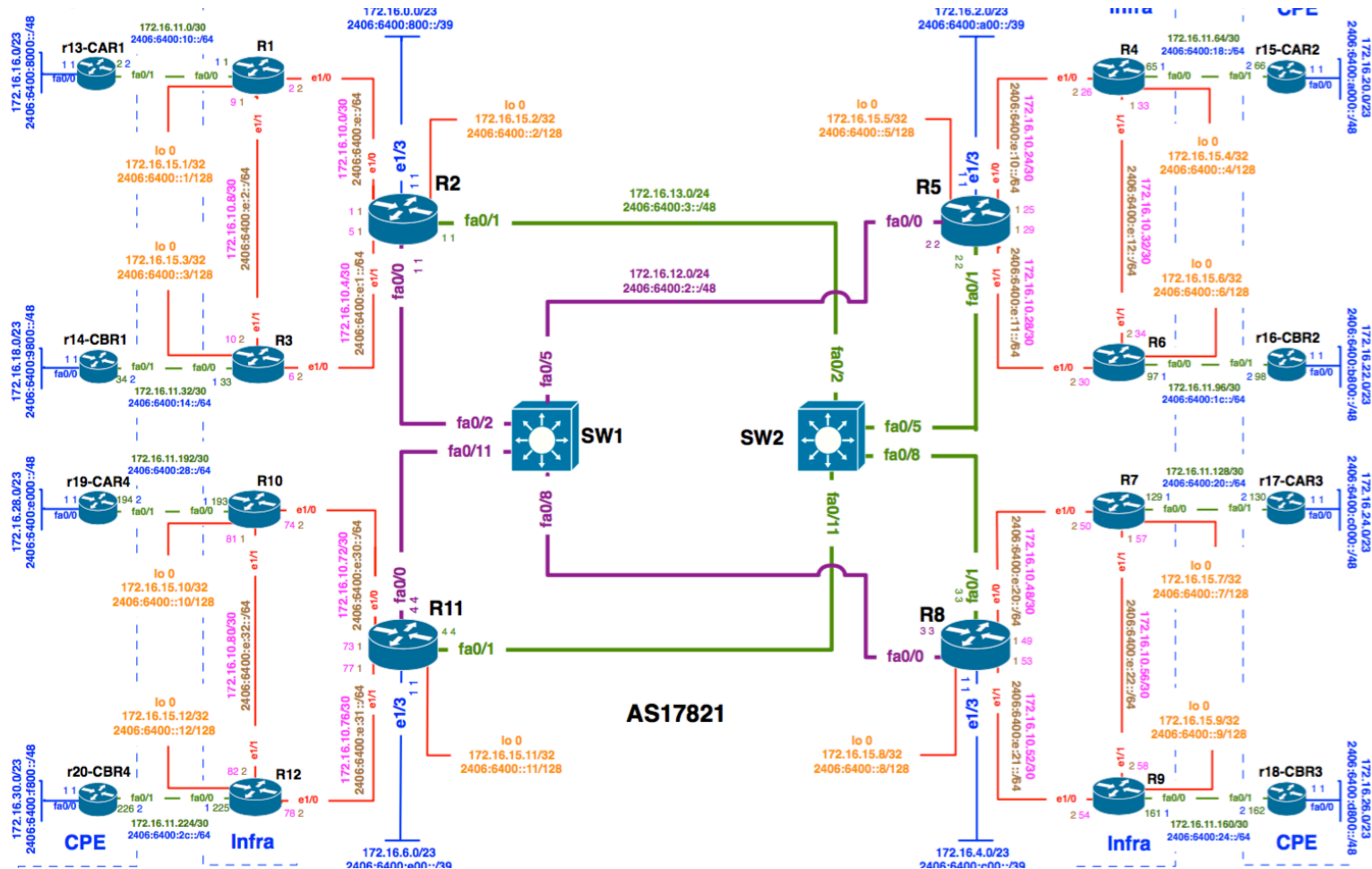
```
protocols {
  bgp {
    group peer-2406:6400:10::2 {
      type external;
      peer-as 65001;
      neighbor 2406:6400:10::2 {
        import policy_65001_1 ;
        family inet6 {
          unicast;
        }
      }
    }
  }
}
```

# RPSL in practice : LAB

# RtConfig: The Big Picture

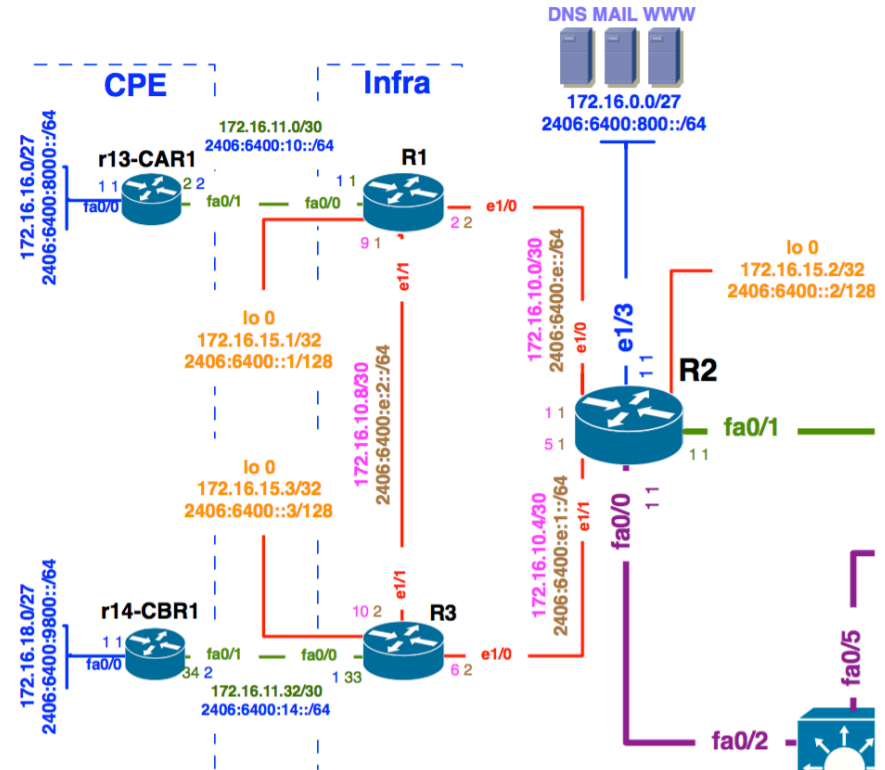


# Topology



# Topology : Region 1

- RPSL Object
  - aut-num : AS17821
  - mnt-by: MAINT-AU-APNICTRAINING
  - route-set: RS-APNICTRAINING
  - fltr-set: FLTR-MARTIAN-V6



# IRRToolSet : RPSL Object

```
# whois -h whois.apnic.net as17821
```

```
mp-import:      afi ipv6.unicast from AS65001 2406:6400:10::2 at  
2406:6400:10::1 action community.append(17821:65001); pref=200;  
accept <^AS65001+$> AND RS-APNICTRAINING:AS65001
```

```
mp-export:      afi ipv6.unicast to AS65001 2406:6400:10::2 at  
2406:6400:10::1 announce ANY AND NOT FLTR-MARTIAN-V6
```



# RtConfig Configuration Template (provision.cfg) – Provision Customer

```
@RtConfig set cisco_map_first_no = 10
@RtConfig set cisco_map_increment_by = 10
@RtConfig set cisco_prefix_acl_no = 100
@RtConfig set cisco_aspath_acl_no = 100
@RtConfig set cisco_pktfilter_acl_no = 100
@RtConfig set cisco_community_acl_no = 10
@RtConfig set cisco_max_preference = 500
!
ip bgp-community new-format
ipv6 unicast-routing
!
! AS65001 CONFIGURATION
@RtConfig set cisco_access_list_no = 500
@RtConfig set cisco_map_name = "AS65001-IMPORT"
@RtConfig import AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
@RtConfig set cisco_access_list_no = 501
@RtConfig set cisco_map_name = "AS65001-EXPORT"
@RtConfig export AS17821 2406:6400:10::1 AS65001 2406:6400:10::2
!
end
```

# IRRToolSet : RtConfig Output File

- Now generate the router configuration file

```
rtconfig -protocol bird -cisco_use_prefix_lists -config  
cisco -h whois.radb.net < provision.cfg >  
/private/tftpboot/router_config.cfg
```

- You will get output of full configuration
- Configuration will be saved in /private/tftpboot

# RtConfig Configuration Template (change.cfg) – Update Customer

- Filter customer based on
  - Prefix List
  - AS-PATH access list
- For that we use
  - AS-SET

# Upload Configuration

- Various ways to upload configuration:
  - SNMP Write
  - NETCONF XML Based
  - Automated Script using expect

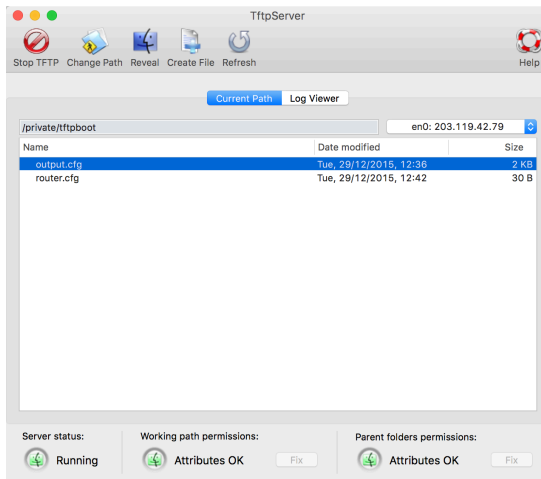
# Upload Configuration : SNMP

- Enable SNMP:

```
access-list 99 permit 10.10.0.0 0.0.255.255
snmp-server community APNIC rw 99
snmp-server ifindex persist
```

– Recommended to use SNMPv3.

- Run TFTP server



# Upload Configuration : SNMP

```
#Set copy method:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.1.2.116 i 1
#Set sourcefile to network file:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.1.3.116 i 1
#Set destination to running-config:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.1.4.116 i 4
#Set TFTP server ip:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.1.5.116 a {ip-address-tftp-server}
#Set desination filename:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.9a6.1.1.1.1.6.116 s router_config.cfg
#Start tftp upload via via OID ccCopyEntryRowStatus:
snmpset -v 2c -c {community-string} {device-ip-address}
1.3.6.1.4.1.9.9.96.1.1.1.1.14.116 i 1
```

Note: The integer highlighted in **red** is a random integer and you can choose any integer between 1 and 255. Keep in mind to use the same integer for the whole upload procedure! See the integer as a session.

# Getting the Complete Picture

- Automation relies on the IRR being complete
  - Not all resources are registered in an IRR
  - Not all information is correct
- Small mistakes can have a big impact
  - Check your output before using it
- Be prepared to make manual overrides
  - Help others by documenting your policy

# RPSL in Summary

1. Define Routing Policy

2. Create IRR Object/Objects

3. Run RtConfig to generate config

4. Push config to router/routers



# Challenges for the Routing Registries

- Lots of Routing Registries
- Accuracy and completeness
- Not every Routing Registry is linked directly to an Internet Registry
  - Offline verification of the resource holder is needed
- Different authorization methods
- Mirrors are not always up to date

**Thanks**