Barry O'Donovan, INEX

APIX 19, Daejeon, South Korea

It seems that many APIX members know IXP Manager in general but do not know the benefits and how to integrate it into their operations.

— an INEX project for over 10 years
— full stack management system for IXPs
— teaches and implements best practice

The probability of a DFZ leak is equal between the smallest rural ISPs and the largest experienced operators

— Nick Hilliard, INEX CTO

IX Requirements:

Security
Consistency
Reliability

IXP Manager History in a Nutshell

— Pre-2007: Simple CRUD interface
— 2007: Zend Framework, Smarty, Doctrine
— 2010: V2 open sourced (INEX committee approval)
— 2012: V3 released (better documentation, evangelism)
— 2017: V4 released (framework transition to Laravel)
— 2019: V4.9.0 released on Jan 8th
— 2019: V5 to be released

Development Effort (1/2)

— continue to develop to solve our own problems
  — our problems are usually your problems!
— project management and oversight
— code quality reviews
— first user / eat our own dog food

Development Effort (2/2)

Yann Robin, full-time developer, thanks to:

It seems that many APIX members know IXP Manager in general but do not know the benefits and how to integrate it into their operations.

DEMO

- Login
- Dashboard
- Adding / viewing members

Top Level Objects

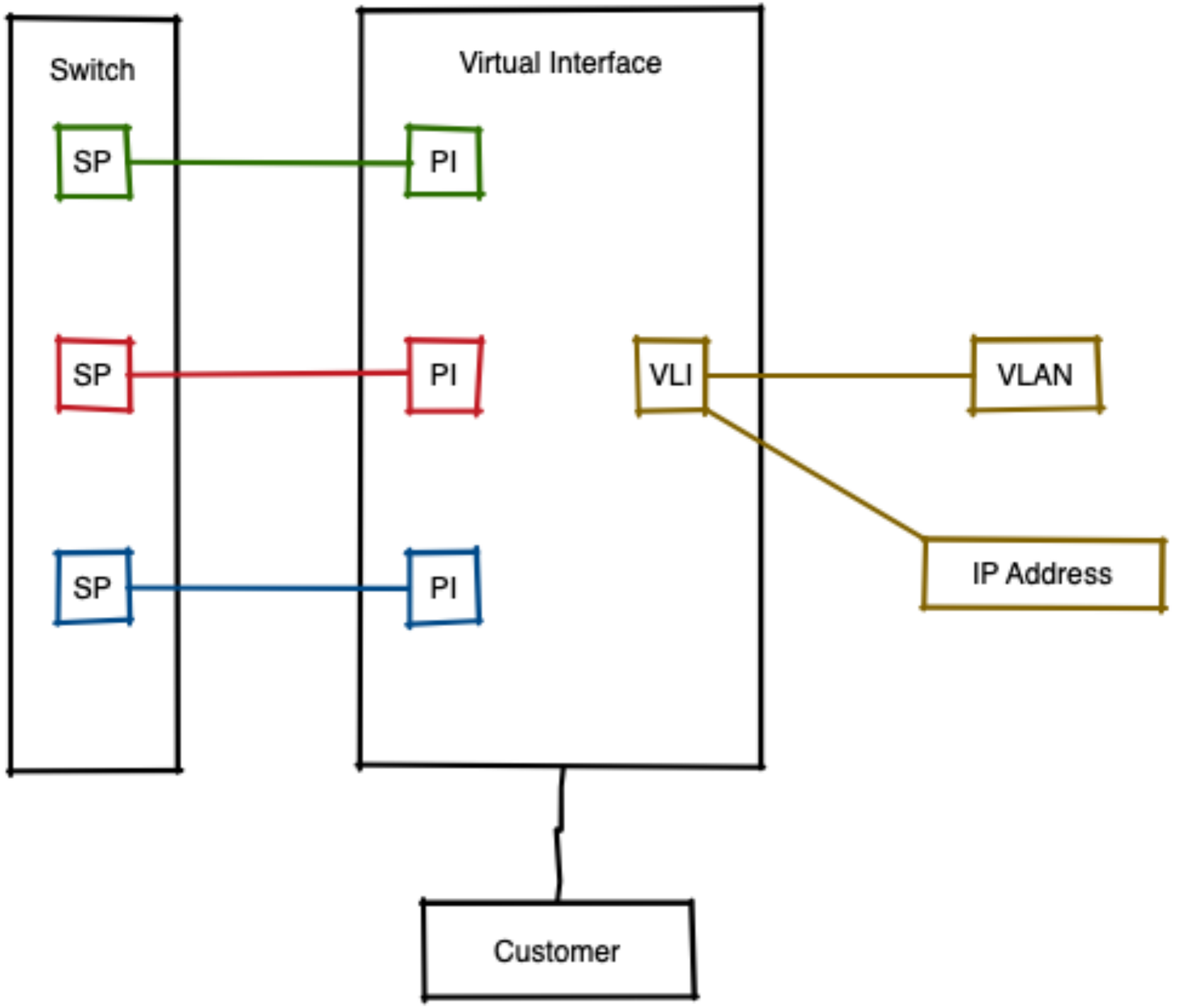
— Facilities (locations / data centres)
— Racks (cabinets) [∈ facility]
— Infrastructures (think of these as an IXP)
— VLANs [∈ infrastructure]
— Switches [∈ infrastructure, ∈ rack]
— Switch Ports [∈ switch]

DEMO

— Top Level Objects

Interfaces in IXP Manager

- Flexible schema that has survived 20 years
- Member port represented by a: virtual interface (VI)
  - a VI has one or more physical interfaces (PI)
    - each PI has a switch port (SP)
  - a VI has one or more VLAN interfaces (VLI)
    - each VLI has a parent VLAN
    - optionally has an IPv4 and/or IPv6 Address
    - options such as route server participation


Switch
Virtual Interface
SP
PI
SP
PI
VLI
VLAN
SP
PI
IP Address
Customer

DEMO

— Interfaces

Auto-Provisioning

When a interface is added to IXP Manager, you (can) get:

— Route Collector BGP session auto-provisioned
— Route Server BGP sessions auto-provisioned
— MRTG / graphing auto-provisioned
— Peer to peer graphs auto-provisioned
— Smokeping target for member’s interface
— Nagios monitoring of member’s interface
— AS112 BGP session
— ARPA DNS for IXP assigned address
— RIR AS-SET / ASN objects

Grapher

— Backend agnostic (MRTG, RRD, OpenTSDB, Carbon)
— Renderer agnostic (just needs JSON data and a PNG)
— Appropriate dynamic backend selection
— New implementations: just extend abstract class

Grapher Backends (1/2)

- Dummy (testing, development, demos)
- Mrtg (Log/RRD) - physical interfaces
  - Overall Aggregate (all IXs / Infrastructures)
  - Per infrastructure
  - Per switch (peering ports + core ports)
  - inter-switch (core ports)
  - member aggregate, LAGs and individual ports
- Mrtg graphs: Bits, packets, errors, discards, broadcast

Grapher Backends (2/2)

- Sflow
  - peer to peer traffic
  - per VLAN
  - per protocol for each of the above
- Smokeping
  - latency to member IPs
- Weathermap

Router Configuration

Router Configuration

```
Subject: [inex-tech] Route server system now in beta
From: Nick Hilliard
Date: Fri Nov 23 12:20:17 GMT 2007

Following the announcement at the last INEX members meeting that we were
looking into running a route server system, we are now pleased to announce
that we now have a route server system which is in stable beta.
As a brief summary, the route server system offers the following advantages:
 - dramatically reduces the number of BGP sessions required to peer with
   other INEX members
 - strict route filtering on inbound announcements means that only prefixes
   registered at RIPE by exchange members will be visible
 - dual-hosted system offers high reliability
 - community based filtering allows route server users to control which
   INEX members their prefixes are sent to
```

strict route filtering on inbound announcements means that only prefixes registered at RIPE by exchange members will be visible

— Nick in 2007 stating INEX's "from day one" policy

Router Configuration - Why Trust IXP Manager?

- Automated, templated and secure from day one
- Nick Hilliard is a co-author of the route server RFCs:
  - RFC7947: BGP Route Server
  - RFC7948: BGP Route Server Operations
- This knowledge and experience part of IXP Manager
- V5 includes better looking glass integration
- Supports route collectors, servers and AS112 services.

Router Configuration - Features (1/2)

— Secure by design / out of the box
— IANA special purpose ranges / bogons filtered
— Next hop hijacking detection and filtering
— Maximum / minimum AS path lengths
— Filters known transit networks (v5)

Router Configuration - Features (2/2)

- Origin ASN filtering based on IRRDB entries (AS sets)
- Prefix filtering based on IRRDB entries (`route[6]:`)
  - Option since v4.8.0 to allow more specifics
- Max prefix limits
- Standard community filters supported
  - including large communities - RFC8092
- MD5 shared secrets

BREAKING NEWS

RPKI Support in v5

Find out more at APRICOT

Routing Security 2, Wednesday, Room 202, 11:30

Configuring Route Servers

Configuring Route Servers

1. Get the required information
2. Add router to IXP Manager
3. Enable route server clients for your members
4. Enable cron job for IRRDB database updates
5. Pull the route server configuration via API

Configuring Route Servers

Pull the route server configuration via API and start:

```
curl -X GET
    -H "X-IXP-Manager-API-Key: my-api-key"
    https://ixp.example.com/api/v4/router/gen-config/rs1-lan1-ipv4
    > /etc/bird/rs1-lan1-ipv4.conf

bird -s /var/run/rs1-lan1-ipv4.ctl –c /etc/bird/rs1-lan1-ipv4.conf
```

Sample intelligent scripts available for this.

Configuring Route Servers

Templates can be skinned to make your own changes:

```
ls -1 resources/views/api/v4/router/server/bird2

community-filter.foil.php
community-filtering-definitions.foil.php
filter-transit-networks.foil.php
footer.foil.php
header.foil.php
neighbor-template.foil.php
neighbors.foil.php
rpki.foil.php
standard.foil.php                          <-- main template file
```

IRRDB Filtering

IRRDB Filtering in IXP Manager

Based on `route[6]:` objects, e.g.:

```
$ whois 192.0.2.1
...
route:    192.0.2.0/24
descr:    Packet Loss Ltd
origin:   AS65501
mnt-by:   JOE-MNT
source: RIPE
...
```

```
$ whois 2001:db8::1
...
route6:   2001:db8::/32
descr:    Packet Loss Ltd
origin:   AS65501
mnt-by:   JOE-MNT
source: RIPE
...
```

IRRDB Filtering in IXP Manager

- Local database storage of members' IRRDB entries (via bgpq3)
- Flexible configuration of IRRDB source database(s) on a per member basis
- Support for both AS sets and just ASNs
- Updating IRRDB database is an asynchronous operation to generating route server configuration

IRRDB Filtering in IXP Manager

Very efficient when using the appropriate data structures (php-ds)

```
$ ./artisan irrdb:update-prefix-db hurricane -vv
Hurricane Electric:
    [IPv4: 703624 total; 0 stale; 0 new; DB updated]
    [IPv6:  77796 total; 0 stale; 0 new; DB updated]
Time for net/database/processing: 6.142705/10.669578/1.262988 (secs)
```

Bird's Eye

Secure API Micro Service for Bird

— Winning project from the RIPE73 IXP Tools Hackathon (Madrid, 2016)

— Integrated with IXP Manager for a built-in looking glass

Bird's Eye

Secure API Micro Service for Bird

Demonstration:

- BGP summary
- Routes received / exported
- Internal tagging for filtered routes

https://www.inex.ie/ixp/lg/rs1-cork-ipv4

Bird's Eye

IXP Manager's Frontend on Bird's Eye

— Provides visibility of 6 x AS112 / 6 x route collectors / 12 x route servers

— Does not expose Bird’s Eye to the public

— Bird status and ‘show bgp summary’ also available as JSON for monitoring

IXP Manager & Routers - Future Work

- Support OpenBGPd (and GoBGP?)
  - Support means feature parity with Bird
- ~~RPKI support~~ (complete, V5)
- UI controls to block/prepend advertised/received prefixes

Patch Panel Management

aka Cross Connects Management

- INEX has ~300 xconnects over 8 PoPs
  - mostly inbound, mostly at member's expense
- used to manage these in Dokuwiki
  - doesn't scale (duh!)
- devised IXP Manager schema to capture most conceivable cases

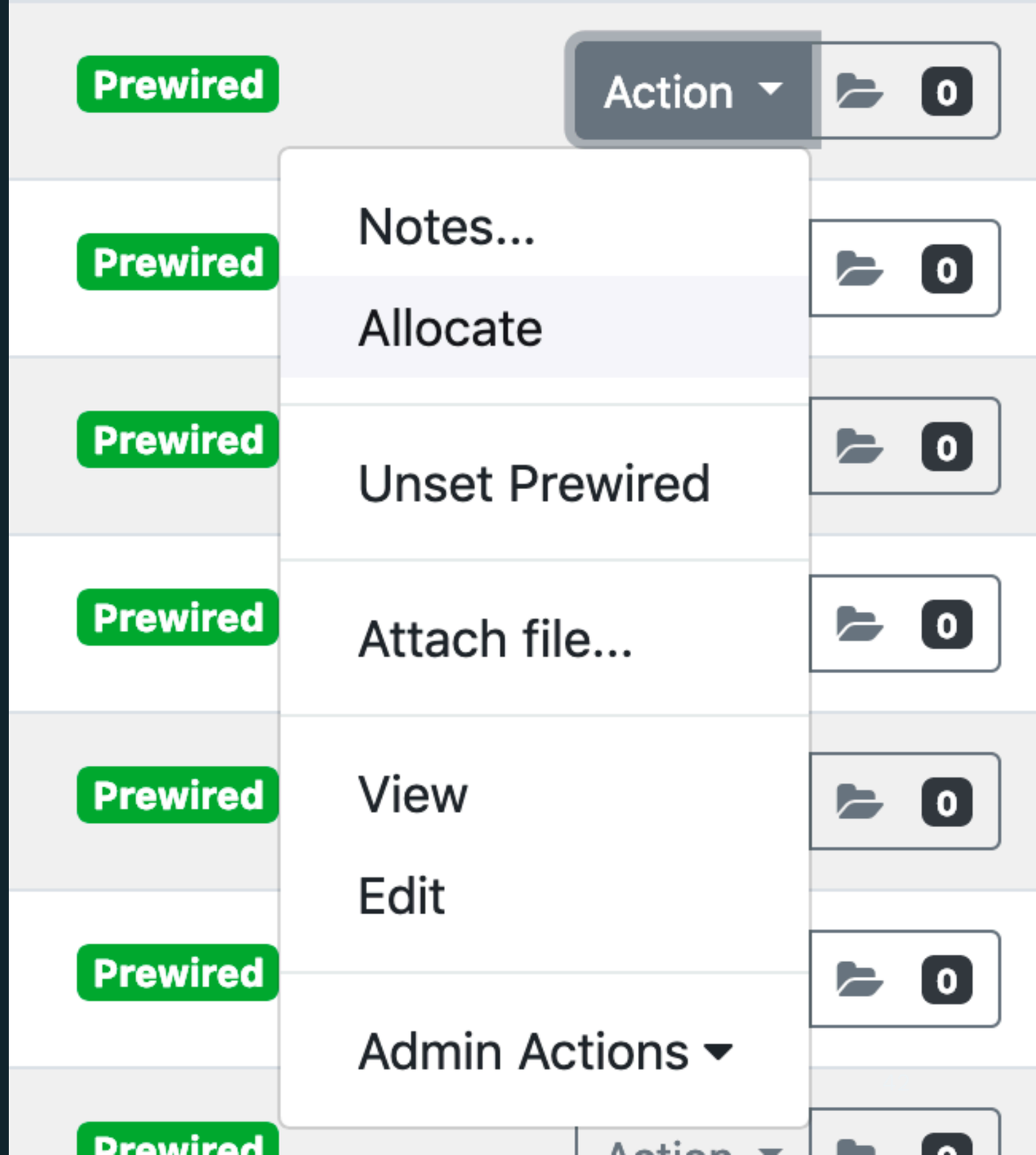
Patch Panel Management - Data Capture

— patch panel ID (ours and facility's)
— utp / smf / mmf - rj47 / sc / lc / mu / etc
— facility's x-connect reference
— installation date, customer and switch port
— description (for non-standard cases)
— reserved / prewired ports
— markdown enabled public/private notes and attachments
— LoA generation
— full history

Patch Panels - Life Cycle Management (1/2)

Context aware options for lifecycle events:

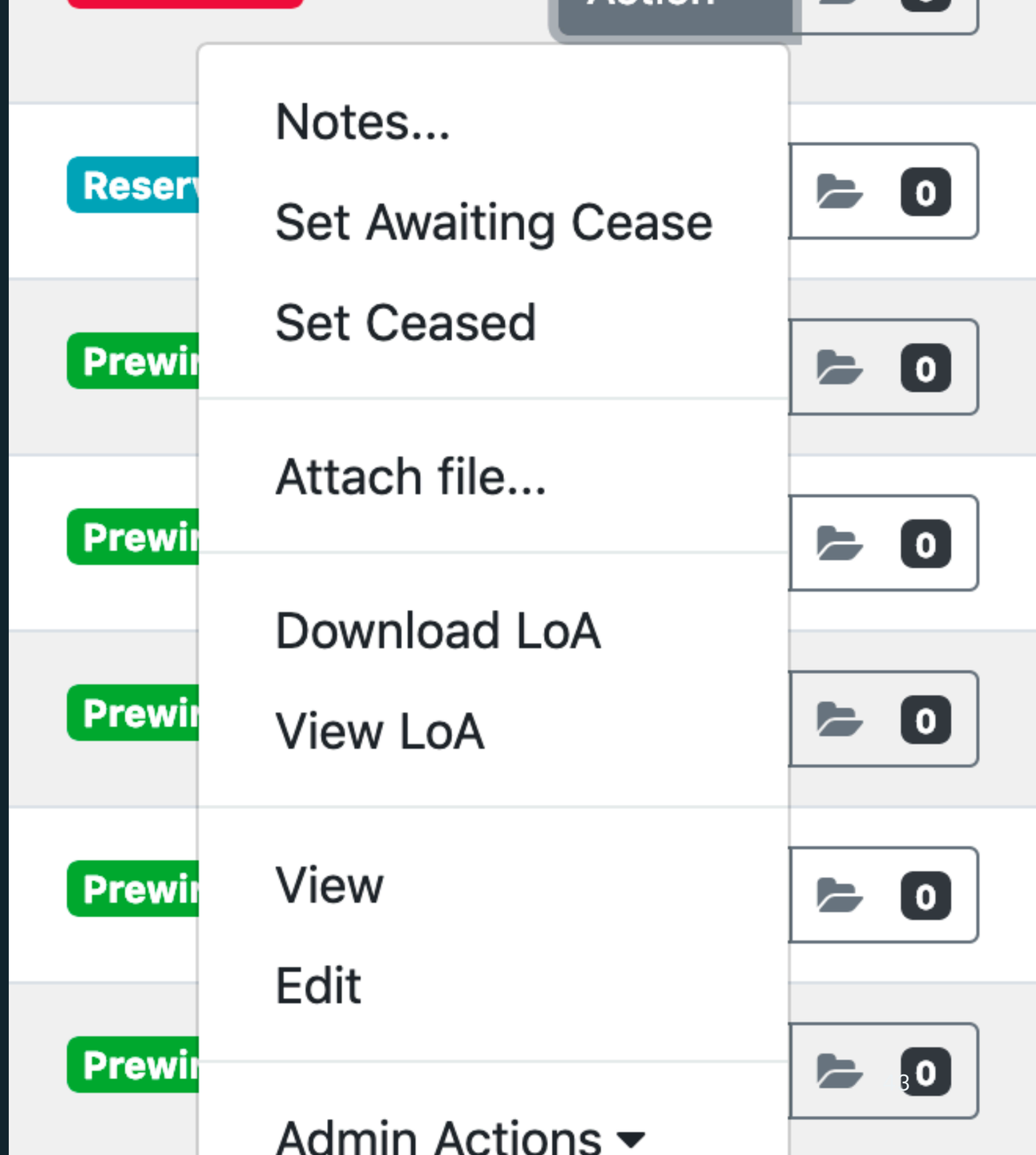
1. Allocate -> Awaiting Cross Connect
2. Connected
3. Cease Request
4. Ceased (and archived)



Patch Panels - Life Cycle Management (2/2)

Context aware options for lifecycle events:

1. Allocate -> Awaiting Cross Connect
2. Connected
3. Cease Request
4. Ceased (and archived)



Automation with IXP Manager

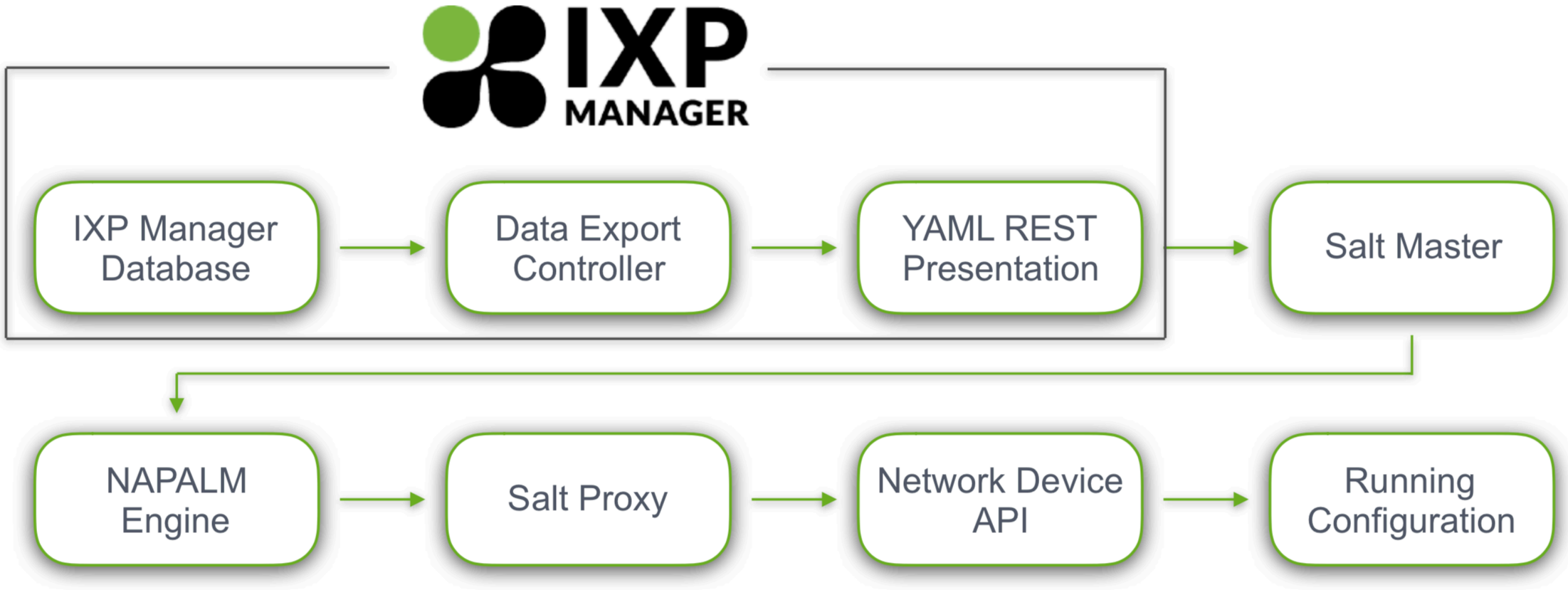
- Always existed - route server configuration
- Reticent about automating network configuration
  - Cost / return ratio wasn’t right
  - Concerns about configuration deployment
  - Poor tool support for interfacing with switches
- Started to change 2016/17
  - Openflow / Yang / Vendor API
- https://github.com/inex/ixp-manager-provisioning

Automation with IXP Manager

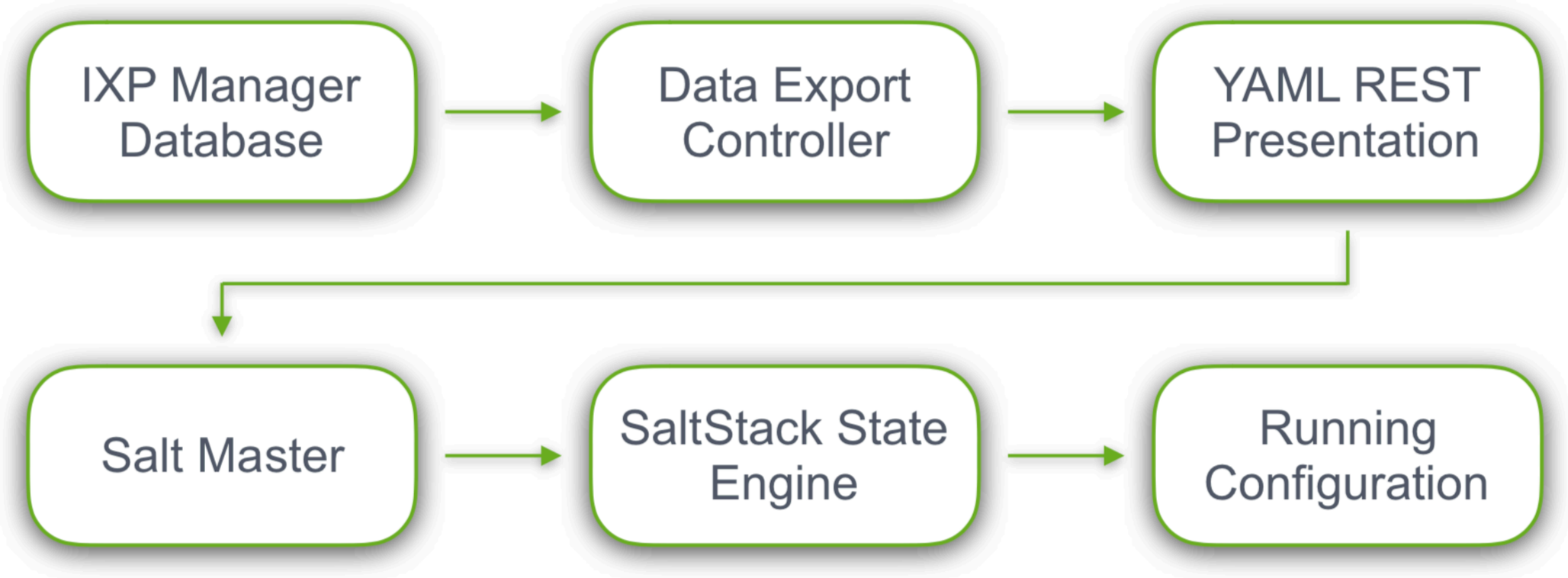
We chose a practical approach:

- YANG: only well supported on tiny number of NOSs
- Openflow: too low level, not loved by chipset manufacturers
- Decided to use NAPALM
  - Integrates with vendor APIs
  - Integrates with Ansible and SaltStack
  - Long term support is likely to be good

Data Flow - Traditional NOS



Data Flow – Cumulus Linux



IXP Manager Date Presentation

- API version 4 exports YAML via REST calls
- Exported data roughly breaks down as:
  - Vlans
  - Layer 2 interface information
  - Layer 3 interface information
  - Information required for routed core (bgp + vxlan)

Sample YAML

```yaml
layer2interfaces:
  - name: swp2
    type: edge
    description: "Packetloss Services Ltd"
    dot1q: yes
    shutdown: yes
    autoneg: yes
    speed: 10000
    lagindex: 1
    lagmaster: no
    fastlacp: yes
    virtualinterfaceid: 334
    vlans:
      - number: 12
        macaddress:
          - "54:1e:56:35:77:d0"
```

Idempotent Atomic Session-Based Configuration Merge

```
{% if bgp.local_as|default(false) %}
no router bgp {{ bgp.local_as }}
router bgp {{ bgp.local_as }}
    no bgp default ipv4-unicast
    bgp always-compare-med
    [...]
{% endif %}

{% for iface in pillar.get('layer2interfaces') %}
default interface {{ iface.name }}
interface {{ iface.name }}
    load-interval 30
    [...]
{% endfor %}
```

That's all folks...

- Travel support
  - Thanks to Facebook, Euro-IX, IX-F and ISOC
  - INEX
- https://www.ixpmanager.org/
- barry.odonovan@inex.ie - @barryo79
- https://www.ixpmanager.org/support