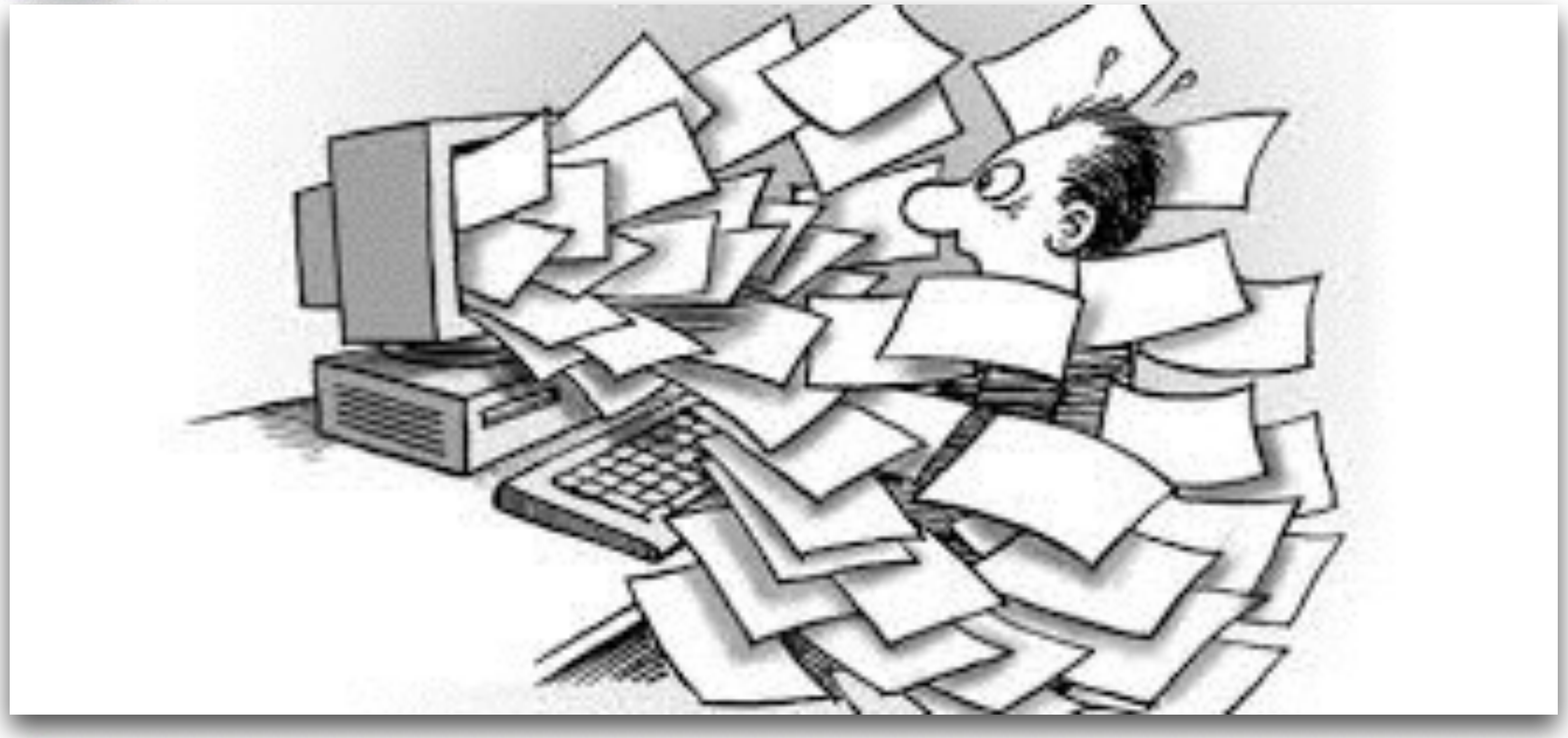
Taming the Floods of Maintenance Notifications

Maintenance Schema

Barry O'Donovan
@ComePeerWithMe / @barryo79
27th Euro-IX Forum, Berlin, Germany

The Problem



The Problem

- Every network needs service affecting maintenance
- Notifications need to be sent **and received** for service affecting maintenance
- Typically, these are done by email by ISPs / IXPs presently.

The Problem

- Every network needs service affecting maintenance
- Notifications need to be sent **and received** for service affecting maintenance
- Typically, these are done by email by ISPs / IXPs presently.
- This presents a number of problems:
  - email overload
  - missed notifications
    - changed aliases, staff rotation, staff promotion, inappropriate distribution lists
  - manually intensive
  - despair / couldn’t be bothered keeping up!
- A solution?
  - Publish maintenance details via an agreed JSON schema

Rationale - Provider View

- All networks announce service affecting maintenance
- Announcement / customer mailing lists are always out of date / inaccurate
- Notifications all have a common subset of data:
  - start and end time
  - description of services affected
  - contact details of on call support team
  - short after action report
- These are all typically unformatted emails
- This sucks and does not allow for automation

Rationale - Customer View

- Even medium sized networks receive a flood of maintenance notifications
  - Quiet often, from old services
  - Quiet often, from services that never made it past a sales enquiry!
- They usually **do not** receive the ones that actually matter
- Exceptionally difficult to find and understand upstream networks' procedures for:
  - announcing maintenance windows
  - publishing planned maintenance windows
  - understanding who receives announcements for maintenance windows
- Managing / sharing / scheduling work around floods of maintenance notifications is time consuming
- Cost efficiency and proper management requires automation

The Problem - In Numbers

INEX generated ~100 maintenance related notifications in the last 12 months

The Problem - In Numbers

INEX generated ~100 maintenance related notifications in the last 12 months

Average of 2 / week

Doesn’t sound like much, right?

The Problem - In Numbers

INEX generated ~100 maintenance related notifications in the last 12 months

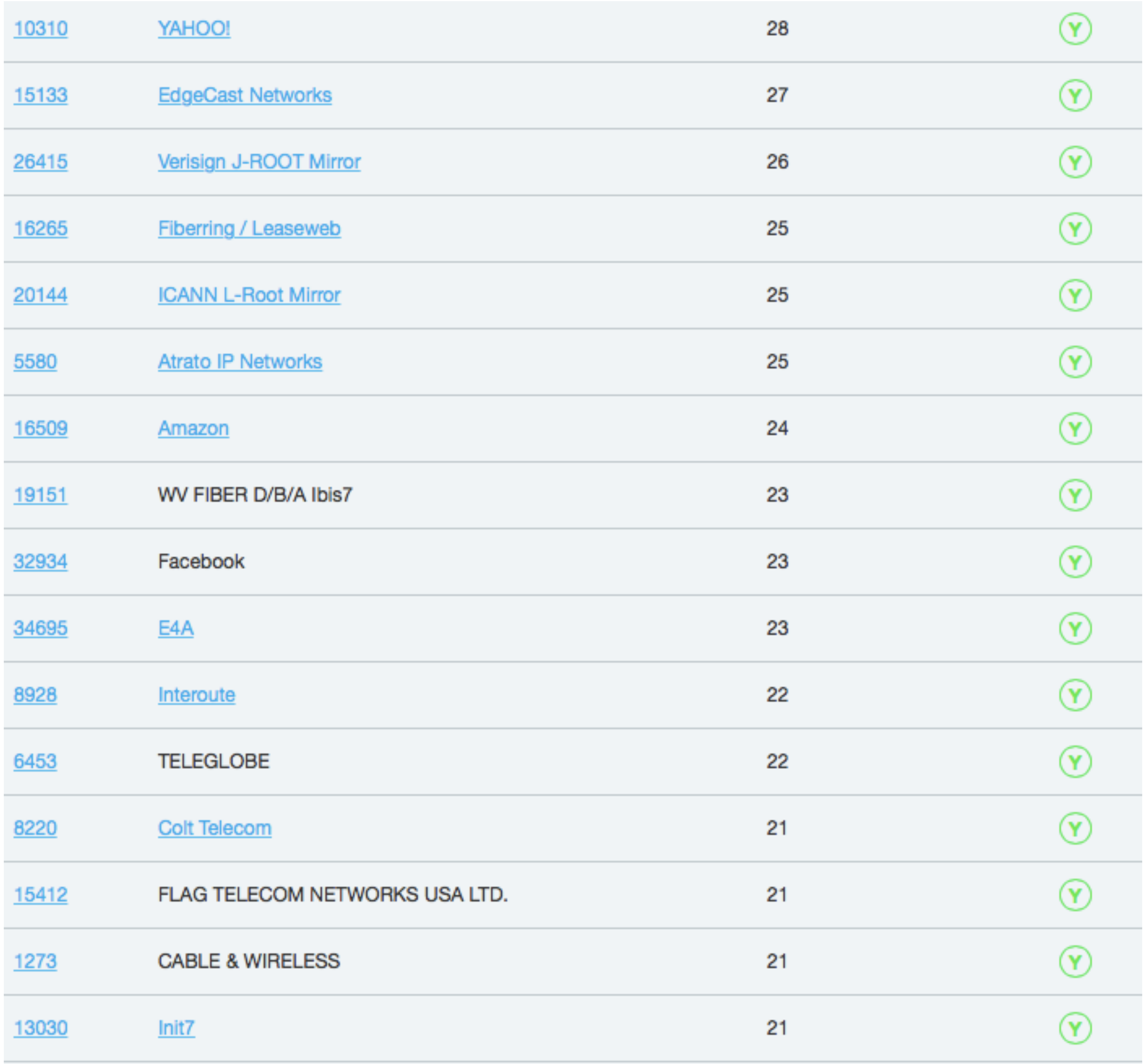
“small-ish” IXP on the western edge of Europe

~80 members, 6 PoPs, 2 LANs, 100Gbps peeks

Stats
Search
Recent
Common

ASN Database

AS#	COMPANY	PRESENT AT	IPV6
20940	Akamai Technologies	73	Y
6939	Hurricane Electric	71	Y
15169	Google	67	Y
3856	Packet Clearing House	58	Y
42	PCH/Woodynet DNS Anycast	53	Y
8075	Microsoft	46	Y
22822	Limelight Networks Inc.	43	Y



10310	YAHOO!	28	Y
15133	EdgeCast Networks	27	Y
26415	Verisign J-ROOT Mirror	26	Y
16265	Fiberring / Leaseweb	25	Y
20144	ICANN L-Root Mirror	25	Y
5580	Atrato IP Networks	25	Y
16509	Amazon	24	Y
19151	WV FIBER D/B/A Ibis7	23	Y
32934	Facebook	23	Y
34695	E4A	23	Y
8928	Interoute	22	Y
6453	TELEGLOBE	22	Y
8220	Colt Telecom	21	Y
15412	FLAG TELECOM NETWORKS USA LTD.	21	Y
1273	CABLE & WIRELESS	21	Y
13030	Init7	21	Y

The Solution?

PeeringDB

euro-IX

Growing trend towards automation
Growing acceptance of the need for agreed schemas

The History?

Discussions on solving this problem has come up many times.

Yet, no solution*.

*) that I know of at least….

We’re Not Promising the Ultimate Solution

This is not a *fait accompli* proposal.

Opening a discussion
(with skin in the game)

EPF -> Euro-IX -> RIPE

We're Not Promising the Ultimate Solution, But…

We have a good track record

IXP Manager, JSON Export Schema

https://github.com/inex/IXP-Manager / https://github.com/euro-ix/json-schemas

IXPs can generally move fast to implement when community demand exists

Maintenance Schema

- Agreed industry standard format with minimal required information
- Uses the KISS principal: **Keep It Simple, Stupid**
  - Goal is to get to a better place than where we are
  - Goal is not to encapsulate every imaginable maintenance type / situation
- Basic minimal required information will include:
  - Schema version
  - Organisation details
  - Support contact details
  - Maintenance window(s) - start / stop time, summary, description, after action report
- Optional (but recommended) to include live updates

Schema Example Definition

```
{
    "version":   "0.1",
    "timestamp": "2015-01-28T00:00:00Z",
    "source":    "https://www.example.com/maintenance.json",

    "organisation":    {},
    "support_details": {},
    "windows": [
        {}, …
    ]
}
```

Schema Example Definition

```
"organisation": {
    "name": "INEX",
    "url": "https://www.inex.ie/",
    "type": "ixp", (or "isp")
    "peeringDB_ix_id": 48, (or "peeringDB_net_id")
    "asn": 2128
},
```

Schema Example Definition

```
"support_details": {
        "url": "https://www.inex.ie/support/",
        "email": "operations@inex.ie",
        "phone": [ "35315313339" ],
        "notes": "Support is available…"
},
```

Schema Example Definition

```
"windows": [
  {
      “id”: “uuid",
      "type": "planned",
      "planned_time_start": "2015-09-01T00:00:00Z",
      "planned_time_stop":  "2015-09-01T02:00:00Z",
      "summary": "Software upgrade & reboot of swi1-tcy1-1",
      "details": "INEX is scheduling a ...",
      "action_required": "no"
  },
  { … }, { … },
]
```

Schema Example Definition

```
"windows": [
  {
      “id": "uuid",
      "type": "planned",
      "planned_time_start": "2015-09-01T00:00:00Z",
      "planned_time_stop":  "2015-09-01T02:00:00Z",
      "summary": “Software upgrade & reboot of swi1-tcy1-1",
      "details": "INEX is scheduling a ...",
      "action_required": "no"
  },
  { … }, { … },
]
```

Schema Example - Type

- **planned**: scheduled in advance as per *normal maintenance* procedures

- **unplanned**: typically a **live outage** - this is happening right now, take planned_time_stop as an *estimated time for restore*

- **emergency**: while it is planned, it is also outside of normal maintenance procedures and will happen soon. Usually <24 hours.

- **noncritical**: informational purposes only, not service affecting. Typical examples include:
  - non-production maintenance such as management connectivity
  - email / phone support issues

Schema Example Definition

```
"windows": [
  {
      "id": "uuid",
      "type": “planned”,
      "planned_time_start": "2015-09-01T00:00:00Z",
      "planned_time_stop":  "2015-09-01T02:00:00Z",
      "summary": “Software upgrade & reboot of swi1-tcy1-1",
      "details": "INEX is scheduling a ...",
      "action_required": “NO"
  },
  { … }, { … },
]
```

Schema Example - Summary / Details

- Summary: equivalent to an email subject. Brevity is appreciated!

- Details: equivalent to an email body. Put your detail here including, in the case of an IXP, full list of members affected for example.

- Details should support Markdown syntax
  - plain text formatting syntax
  - no loss of detail if not converted to HTML
  - no tags to strip / no processing required
  - if used, can add greater clarity and detail

Schema Example Definition

```
"windows": [
  {   "id": "uuid",
      "type": "planned",
      "planned_time_start": "2015-09-01T00:00:00Z",
      "planned_time_stop":  "2015-09-01T02:00:00Z",
      "summary": “Software upgrade & reboot of swi1-tcy1-1",
      "details": "INEX is scheduling a ...",
      "action_required": "yes",
      “action": "Tear down BGP sessions in advance."
  },
  { … }, { … },
]
```

Schema Example - Action Required

- **yes** - member / customer action required
- **no** - no action required. e.g. IXP / ISP will bring down BGP sessions in advance
- **recommended** - at the customers' discretion - for example, an IXP with switches incapable of L3 IPv6 ACLs cannot force IPv6 sessions down but IPv6 amounts to <<1% of overall traffic. *recommended* may be more appropriate than *yes*.

- For *yes* and *recommended*, the details of the advised / required action should be included in a `"action":` key/value pair. This is optional for *yes*.

Schema Example Definition

```
"windows": [
  {
      “id”: “uuid",
      "type": "planned",
      "planned_time_start": "2015-09-01T00:00:00Z",
      "planned_time_stop":  "2015-09-01T02:00:00Z",
      "summary": "Software upgrade & reboot of swi1-tcy1-1",
      "details": "INEX is scheduling a ...",
      "action_required": "no"
  },
  { … }, { … },
]
```

Advanced Schema Example - Affected IPs

```
"windows": [ {…
    "addresses_affected": [
        "193.242.111.8",
        "2001:7f8:18::8"
    ],
}, … ]
```

- Ideally suited to IXPs
  - could have circuit_id / service_id / asns / etc also
- Identify IXP peers who will be affected by a maintenance window. Unaffected members can then ignore / silence peer down alerts with these (for example).
- ISPs could use with authenticated API requests

Schema Example - Updates

- Realtime updates on start / completion, issues, rescheduling, etc can be very useful:

```
"windows": [ {
    …,
    "updates": [ {
      "id": 10,
      "timestamp": “2015-01-28T00:10:00Z",
      “type”: “started,completed,update,restarted,rescheduled,cancelled”,
      “rescheduled_id”: “uuid”,
      “notes”: “This maintenance window is now…”
    }, … ],
 }, …
]
```

Schema Example - Updates

- Typical `updates` sequence:

1. update: This maintenance window will begin as planned in 1 hour
2. start: This maintenance window begins now
3. update: Due to unforeseen complications, this maintenance window is expected to run over by 30mins
4. completed: This maintenance window is now complete. After action / summary will be issued in approx. 1 hour.
5. update: Maintenance Report: tonight's maintenance window proceeded as scheduled but ran later than planned due to…

Been here before. So friggin’ bored right now.

ERR: TOO_MUCH_EFFORT

Networks / IXPs will never implement this

<insert many valid reasons>

ERR: TOO_MUCH_EFFOT

Networks / IXPs will never implement this

<insert many valid reasons>

We promised skin in the game…

Promised skin in the game…

https://www.maintenancemanager.org/

A **working** proof of concept.

A statement of intent.

Promised skin in the game…

Implemented:

- Register / Authentication
- Add network(s) *(first come, first served)*
- Add a maint window & updates
- JSON export

Promised skin in the game...

Planned:

- ical export - others? rss?
- Network verification
- Multiple users to networks
- Directory, Subscriptions, Emails

Practical Uses

I just want to add an ICAL feed to my own calendar

JSON Maintenance Schema => ICAL - easy!

We use RSS feeds internally

JSON Maintenance Schema => RSS - easy!

You know, I'm pretty happy with emails

JSON Maintenance Schema => Email - easy!

Practical Uses

Add intelligence:

Just show me the windows that affect me!

/peerip/1.2.3.4
/asn/64511
/customerid/4685

Thank you!

https://www.maintenancemanager.org/

Barry O'Donovan
barry.odonovan@inex.ie
@ComePeerWithMe / @barryo79

27th Euro-IX Forum, Berlin, Germany