IX-F Database
IXP Database Project

- There are many IXP Databases...
Existing database issues

• No single trusted data source
• Poor quality & no or very little APIs / automation
• “We’ll only peer with you, if you’re in PeeringDB”
• Databases are updated manually
• We’re moving into a world of automation
• …IXPs hold accurate information but lack automation
Historical Problem

PeeringDB - Database for networks and data centres
IX-F Database

IX-F DB API server written using Python / Django, which can:

- get / create / update / delete IXP, organisation, IP addressing and network information
- all interaction is JSON
- all non-sensitive information will be publicly available
- Members of IXPAs will be able to create, update and delete IXPs from the databases.
- [https://db.ix-f.net/api/ixp](https://db.ix-f.net/api/ixp)
IX-F Database

• We have a proof of concept client to interact with this database in Python including unit test code at: https://github.com/euro-ix/ixf-client-py

• The PHP version of this with unit tests also available at: https://github.com/euro-ix/ixf-client-php
Where we are…

IXP Manager

IXPs

IXPs

IXPs

Euro-IX

Django/Python code

API

IXPAs

IX-F DB

API

PeeringDB

Telegeography
IX-F Database

Where are we..

• Designed for IXPs to export (push) data with minimal intervention
• Simple modifications can be used for IXP to IXPA or to PeeringDB
• At present Euro-IX is the only IXPA interface ready to talk with PeeringDB - we encourage **ALL** IXPs to use the Euro-IX website while others are being worked on.
• APIX started work, LAC-IX and AFIIX to come..
IX-F DB

IXP info

DC/Network info

PeeringDB

Af-IX
APIX
Euro-IX
LAC-IX

ISP
Network
Data Center

IXP
IXP
IXP
IXP

IXP
IXP
IXP
IXP

IXP
IXP
IXP
IXP
A real life example..

“Who am I not peering with at LONAP?”

• You have a script which load direct adjacencies into an array.

• You need a complete and canonical list of peers to compare differences
IX-F Database

- https://db.ix-f.net/api/ixp

Very very easy mate

{
    "ixp_info": {
        "status": "active",
        "updated": "2014-02-17T10:07:51Z",
        "name": "London Network Access Point",
        "created": "2011-08-16T13:26:26Z",
        "shortcut": "LONAP",
        "ixp_id": "IX-F#18"
    },
    "timestamp": "2015-09-16T08:17:31.116Z",
    "version": "2014110401",
    "member_list": [
        {
            "asnum": 20915,
            "name": "100 Percent"
        },
        {
            "url": "http://www.2connectbahrain.com/",
            "asnum": 51406,
            "name": "2Connect"
        },
        {
            "url": "http://www.34sp.com",
            "asnum": 41357,
            "name": "34SP.com Ltd"
        },
        {
            "url": "http://4d-dc.com/",
            "asnum": 31463,
            "name": "4D Data Centres"
        },
        {
            "url": "http://www.afilias.info",
            "asnum": 12041,
            "name": "Afilias"
        },
        {
            "url": "http://www.akamai.com",
            "asnum": 20940,
            "name": "Akamai Technologies"
        },
        {
            "url": "http://www.alentus.com",
            "asnum": 21321,
            "name": "Alentus UK Ltd"
        }
    ]
}
import urllib, json

url = "http://db.ix-f.net/api/ixp/18/member-list"
response = urllib.urlopen(url)

ixpdata = json.loads(response.read())

my_peers = [8916, 20940, 20915, 51406, 41357, 31463, 12041, 21321, 12536, 16509, 20712, 33920, 4]

for member in ixpdata["member_list"]:
    if member["asnum"] not in my_peers:
        print "Get some peering with " + str(member["asnum"]) + " (" + member["name"] + ")"
enigma:Desktop andy$
enigma:Desktop andy$ python ixp.py
Get some peering with 6871 (PlusNet)
Get some peering with 8689 (PowerGroup (Power Internet Ltd))
Get some peering with 8676 (PRT Systems)
Get some peering with 28792 (Public Internet Limited)
Get some peering with 31402 (Rank Interactive (Blue Square Limited))
Get some peering with 35662 (Redstation)
Get some peering with 5552 (Redstone Communications Ltd)
Get some peering with 5503 (RM Education Plc)
Get some peering with 51409 (Sectorsix)
Get some peering with 50056 (Advantage Interactive Ltd)
Get some peering with 29550 (Simply Transit Ltd.)
Get some peering with 48961 (Warwicknet Ltd)
Get some peering with 20738 (Webfusion)
Get some peering with 44444 (Websense Hosted R&D Ltd. (UK))
Get some peering with 49158 (Wifinity)
Get some peering with 13037 (Zen Internet)
enigma:Desktop andy$
Why not just use the IXPs own data?

- This gives you a single API for many IXPs
- Get the same format for all IXPs, it’s standard - wohoo!
- Data is fed from the IXP - IXPs have accurate data!
- Portable, supportable and scaleable!
In search of accurate information

- Give network operators the choice of getting accurate information from either IX-F or PeeringDB
- This data can be obtained using APIs
  - PeeringDB & Telegeography
- Increases use of automation
  - saves time, saves money, increases accuracy.
JSON Schema
Euro-IX IXP JSON Schema

• Contains both IXP data & IXP Participant data
  – ASN (member list), locations, switch, IXP info
• Open, consistent & an atomic design
• Currently 12 IXP independent implementation
• Open source implementation in IXP Manager
• Source available on github: https://github.com/euro-ix/json-schemas
Thank you!

Bijal Sanghani
bijal at euro-ix dot net
Twitter: @euroix