PeeringDB

# PeeringDB Workshop

How is PeeringDB organised? // Track 1

arnold@peeringdb.com

# Agenda

- Please always use the tutorial DB at https://tutorial.peeringdb.com

- What is PeeringDB?
- Organisation
  - History
  - Association
  - Committees

# What is PeeringDB?

**Mission statement:** “PeeringDB, a nonprofit member-based organization, facilitates the exchange of user maintained interconnection related information, primarily for Peering Coordinators and Internet Exchange, Facility, and Network Operators.”

# Why should I have a record in PeeringDB?

- As a network a PeeringDB record makes it easy for other networks to find you, and helps you to establish peering / interconnection
- As a colocation provider a PeeringDB record creates visibility, and helps you to attract additional networks and Ixes
- As an IX a PeeringDB record provides information about your participants, and colocations where your service is available
- Provides a user friendly GUI and a powerful API for automasation

# Governance and Membership

- PeeringDB is a United States 501(c)(6) volunteer organization that is 100% funded by sponsorships
- Healthy organization, building financial reserves and executing the long term strategic plan
- Membership rules
  - A corporation, limited liability company, partnership or other legal business entity may be a Member of the Corporation
  - Membership is determined by having both an active PeeringDB.com account and an individual representative or role subscription to the PeeringDB Governance mailing list
  - 344 addresses subscribed to the Governance mailing list (as of April 16, 2019)
  - Governance list is at http://lists.peeringdb.com/cgi-bin/mailman/listinfo/pdb-gov
  - More information available at http://gov.peeringdb.com/

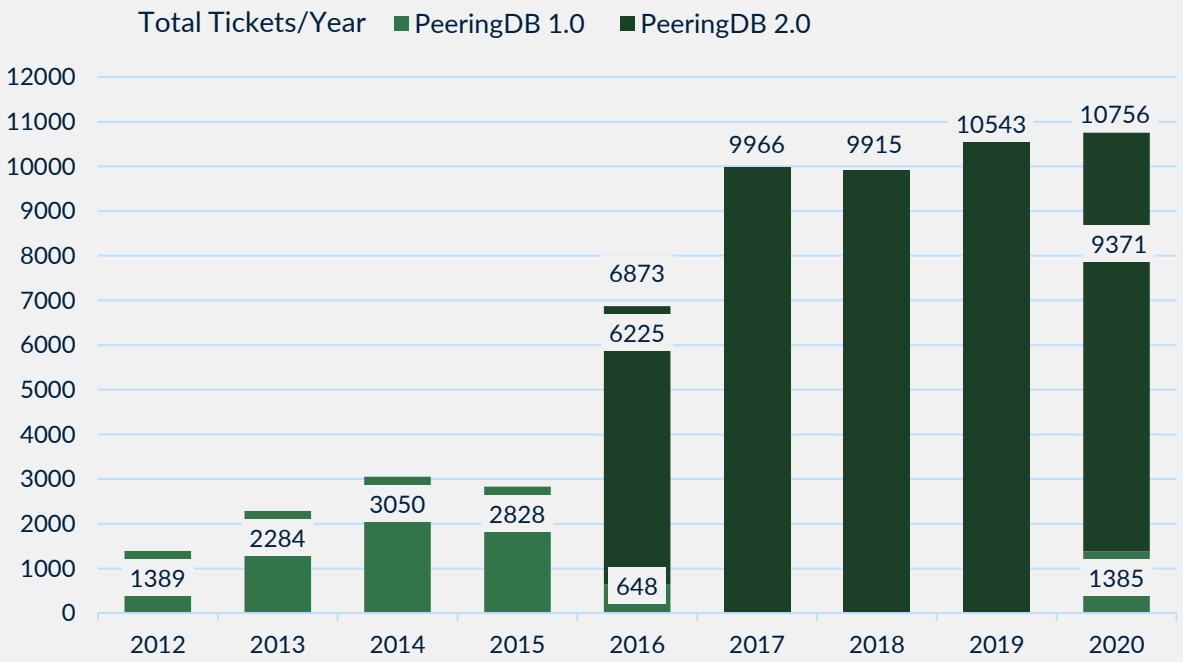
# Governance

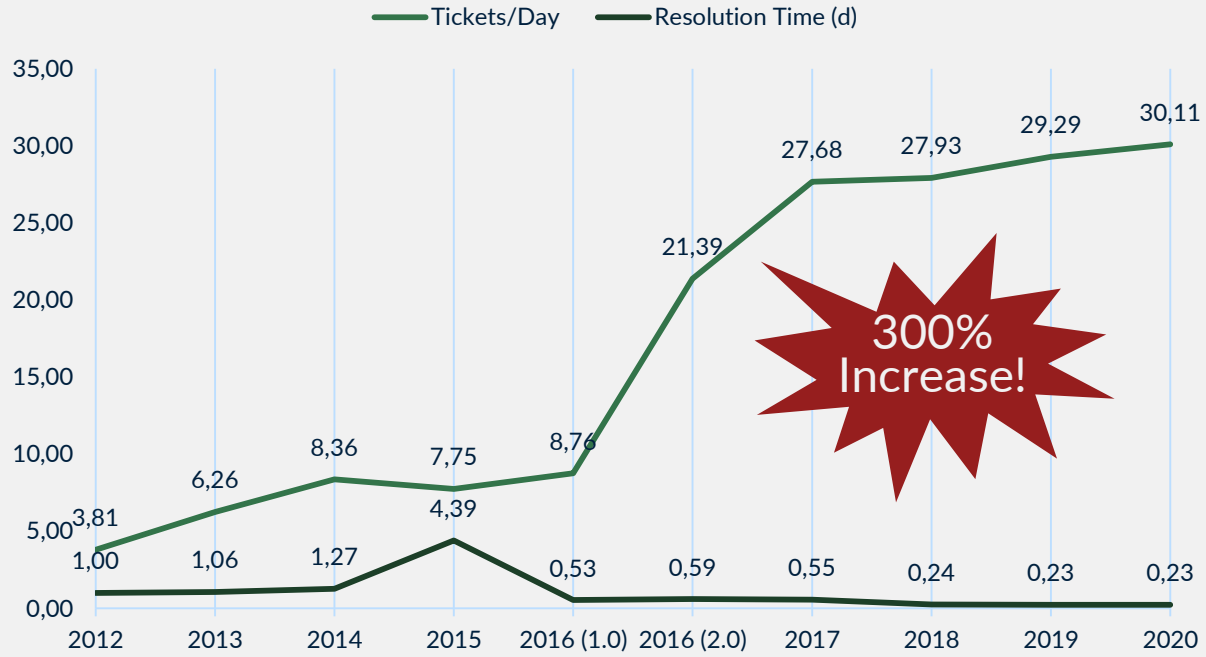
- The Members
  - Any corporation, limited liability company, partnership or other legal business entity may be a Member
  - One (virtual / online) member meeting per year
- The Board
  - Sets strategic directions and overlooks financial issues
  - Half of the board is elected every year
- The Committees
  - Responsible for the day to day work
  - Admin Committee
  - Operations Committee
  - Outreach Committe
  - Product Committe

# Committees

| Admin Committee | Operations Committee | Outreach Committee | Product Committee |
|---|---|---|---|
| • Manage administration of user accounts and PeeringDB records<br>• Answer support tickets<br>• Cleansing and completion of PeeringDB records | • Manage PeeringDB infrastructure | • Manage marketing and social media<br>• Develop and maintain presentations, workshops and webinars<br>• Coordinate presentations and attendance at events | • Manage roadmap and development priorities<br>• Ask for input from the community on desired features<br>• Write SoWs to solicit bids to complete requested features |
| Leads: Stefan Funke (Chair)<br><br>Contact:admincom@lists.peeringdb.com | Leads: Job Snijders (Chair) and Aaron Hughes (Vice Chair)<br>Contact: pdb-ops@lists.peeringdb.com | Leads: Greg Hankins (Chair) and Bijal Sanghani (Vice Chair)<br>Contact: outreachcom@lists.peeringdb.com | Leads: Stephen McManus (Chair) and<br>Matt Griswold (Vice Chair)<br>Product Manager: Filiz Yilmaz<br>Contact: productcom@lists.peeringdb.com |

# Support Ticket Statistics

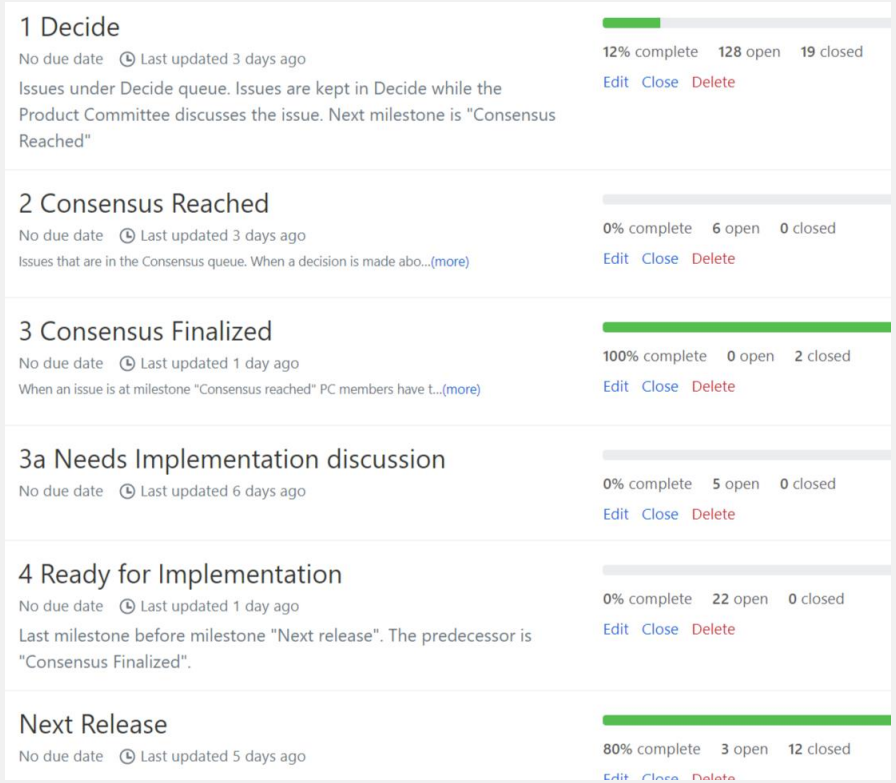



- Admin Committee volunteers are based around the world in a variety of time zones with diverse language skills
- Goal is to resolve support tickets within 24 hours

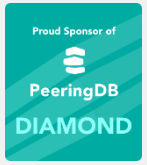
# Product Development Workflow

- All issues tracked using GitHub at https://github.com/peeringdb/peeringdb/issues
  - Anyone can open a feature requests or file a bug report
  - Open and transparent process for product development
  - Workflow is at http://docs.peeringdb.com/workflow/
- Product Committee issue process
  - Evaluate and prioritize the requests
  - Request a quote for development costs
  - Request budget from the board
  - Manage implementation and scheduling
- Your input is needed on features!



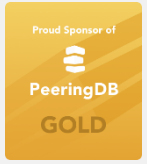
# Become a PeeringDB Sponsor!

- Diamond Sponsorship - $25,000 / year




  - Limited to 2 sponsors
  - Very large logo on top line of Sponsors page with URL
  - Diamond Sponsor badge display on all records
  - Social media promotion
- Platinum Sponsorship - $10,000 / year


  - Large logo on second line of Sponsors page with URL
  - Platinum Sponsor badge display on all records
  - Social media promotion
- Gold Sponsorship - $5,000 / year




  - Medium logo on third line of Sponsors page
  - Gold Sponsor badge display on all records
  - Social media promotion
- Silver Sponsorship - $2,500 / year


  - Small logo on fourth line of Sponsors page
  - Silver Sponsor badge display on all records
  - Social media promotion

## Microsoft Diamond Sponsor

| | |
|---|---|
| Organization | Microsoft Corporation |
| Also Known As | 8068 8069 |
| Company Website | |
| Primary ASN | 8075 |

## DE-CIX Frankfurt Platinum Sponsor

| | |
|---|---|
| Organization | DE-CIX Management GmbH |
| Long Name | Deutscher Commercial Internet Exchange |
| City | Frankfurt |
| Country | DE |
| Continental Region | Europe |
| Media Type | Ethernet |

Proud Sponsor of **PeeringDB** Gold

Contact sponsorship@peeringdb.com for sponsorship info!

# Thank you to our sponsors!

Diamond Sponsor



Platinum Sponsors




Gold Sponsors







Silver Sponsors





















# PeeringDB Workshop

Learning the Ropes // Track 2

arnold@peeringdb.com

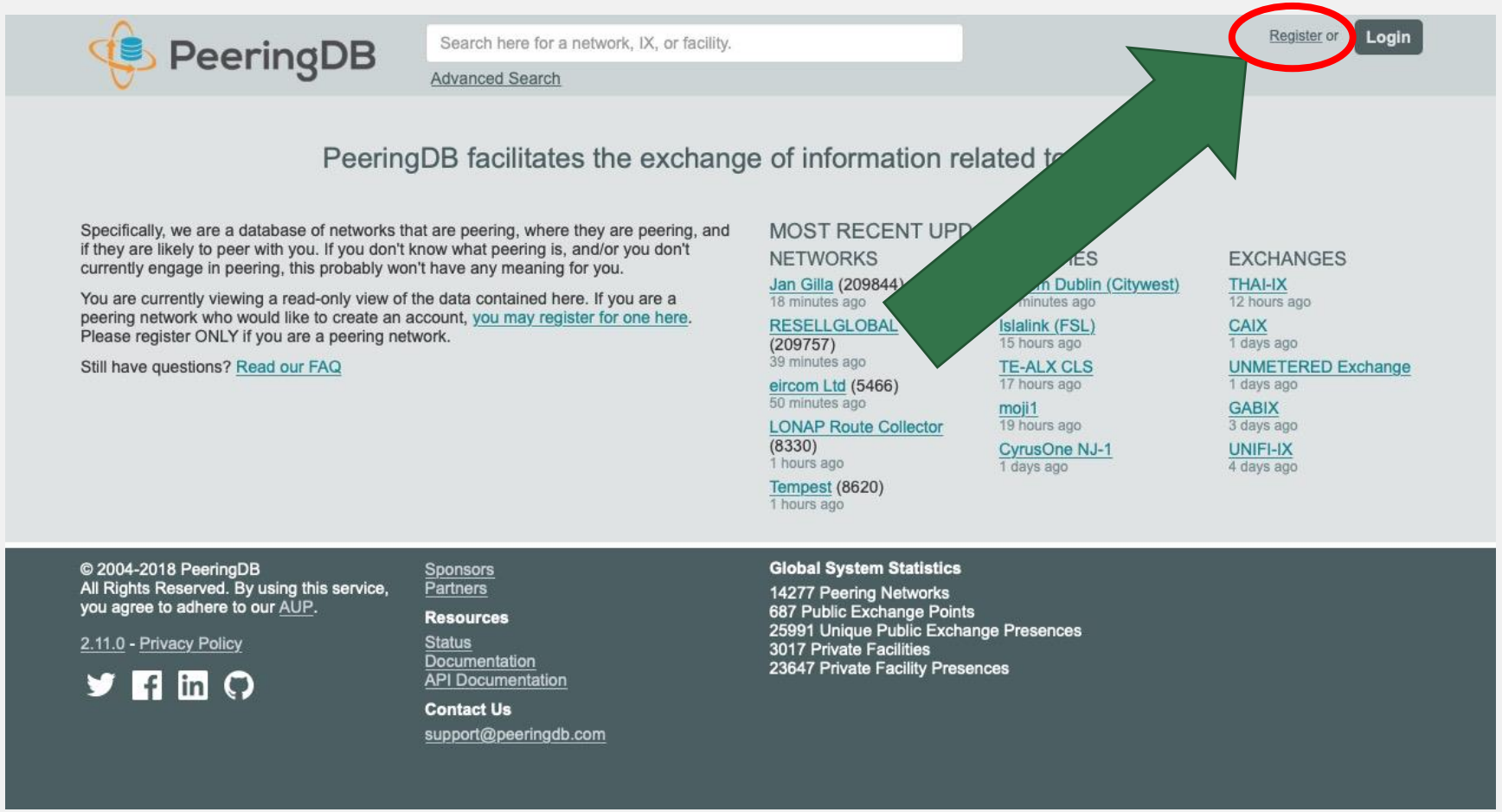
PeeringDB

# Agenda

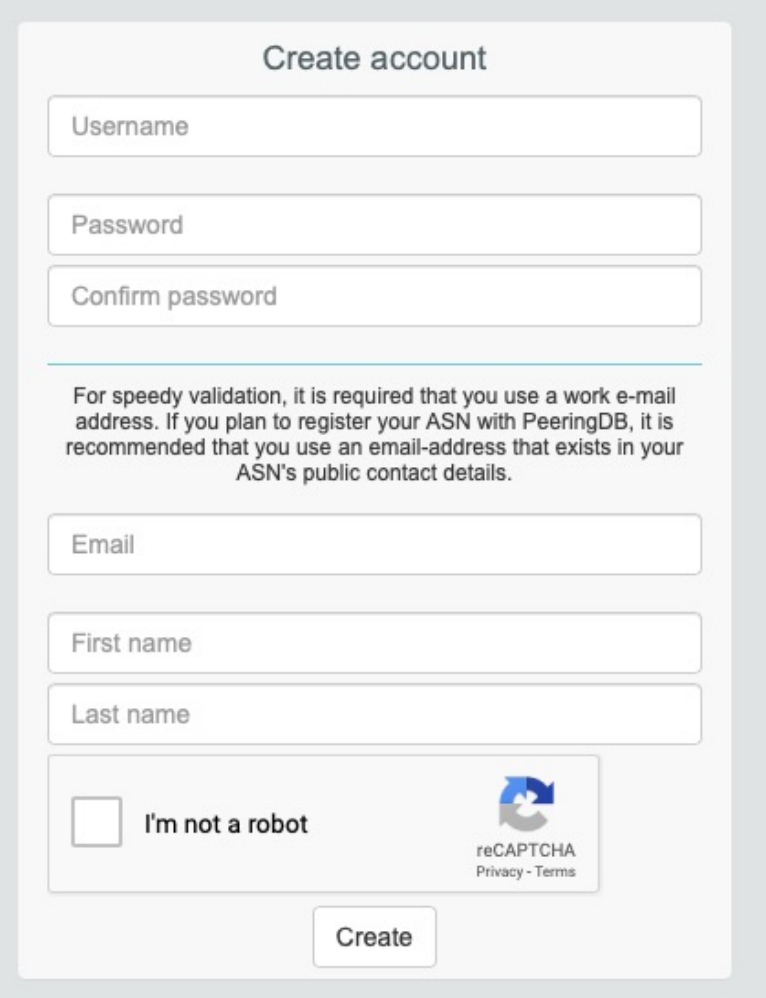
- Please always use the tutorial DB at https://tutorial.peeringdb.com

- Registering
  - Yourself
  - Your organization
- Adding information
  - About your organization
  - About your network / Autonomous System
  - Adding your peering policy
- Internet Exchanges and Facilities
  - Add where your network is present
- Retrieve information
  - Present what you have learned about another network

# Registering



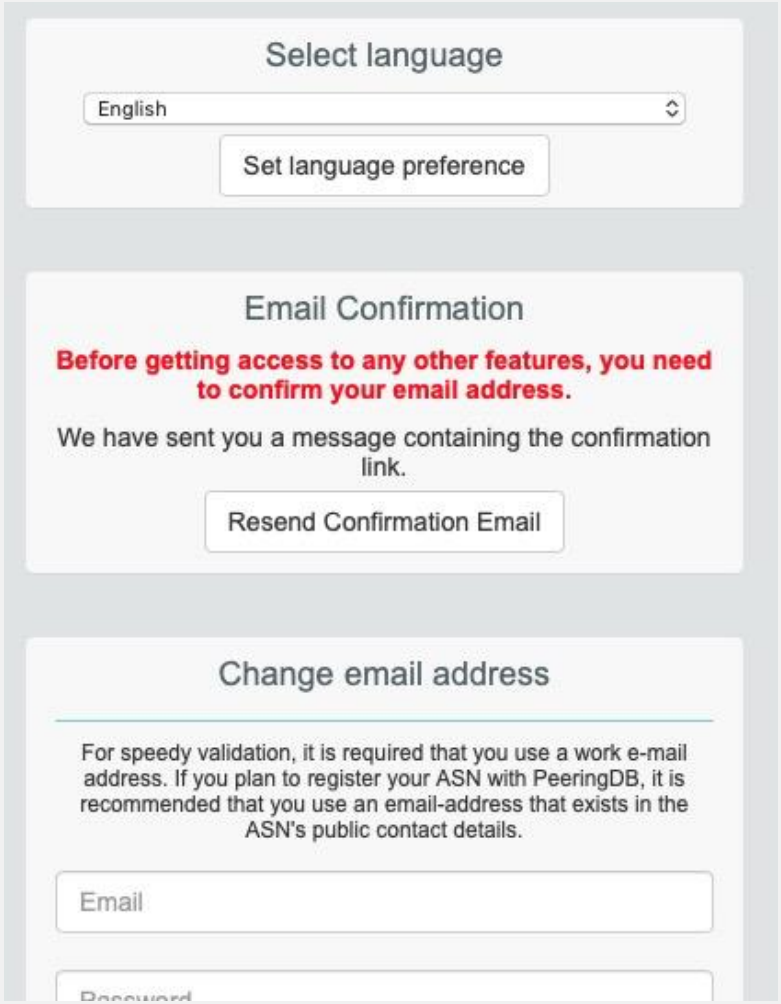
# Registering




- Choose an username
- Password must be at least 10 characters long

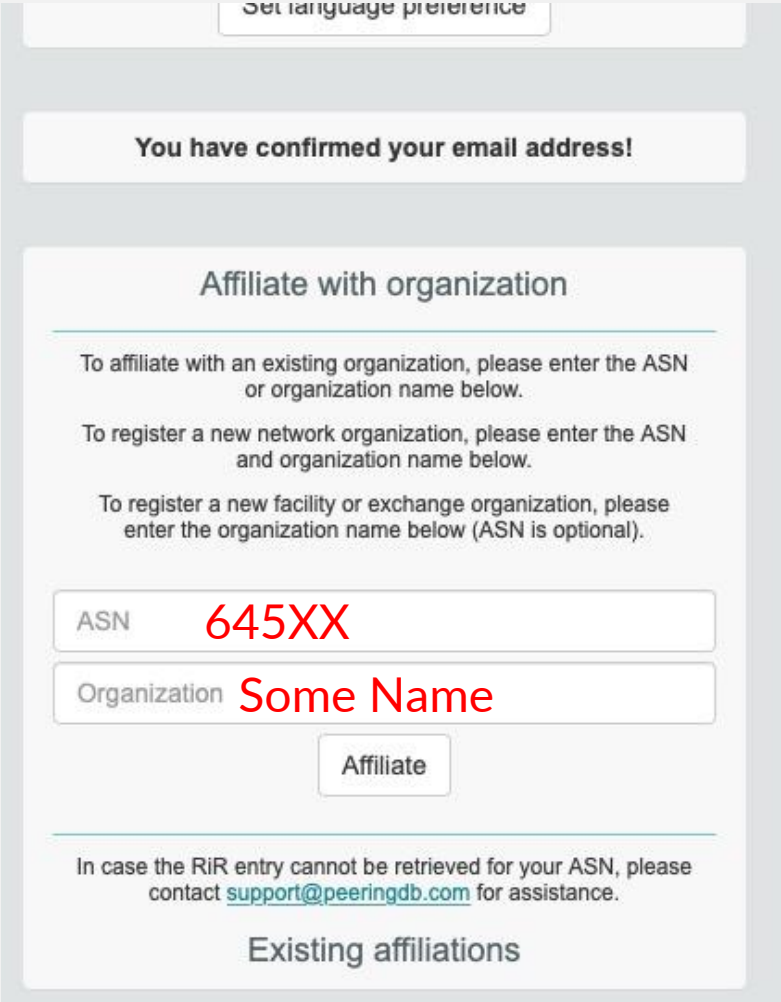
- Use a **real** work **e-mail address**
  - Ideally the one you registered the ASN with
- And put in your first and last name
- You receive a confirmation email
- Click on the link in the email

# Registering



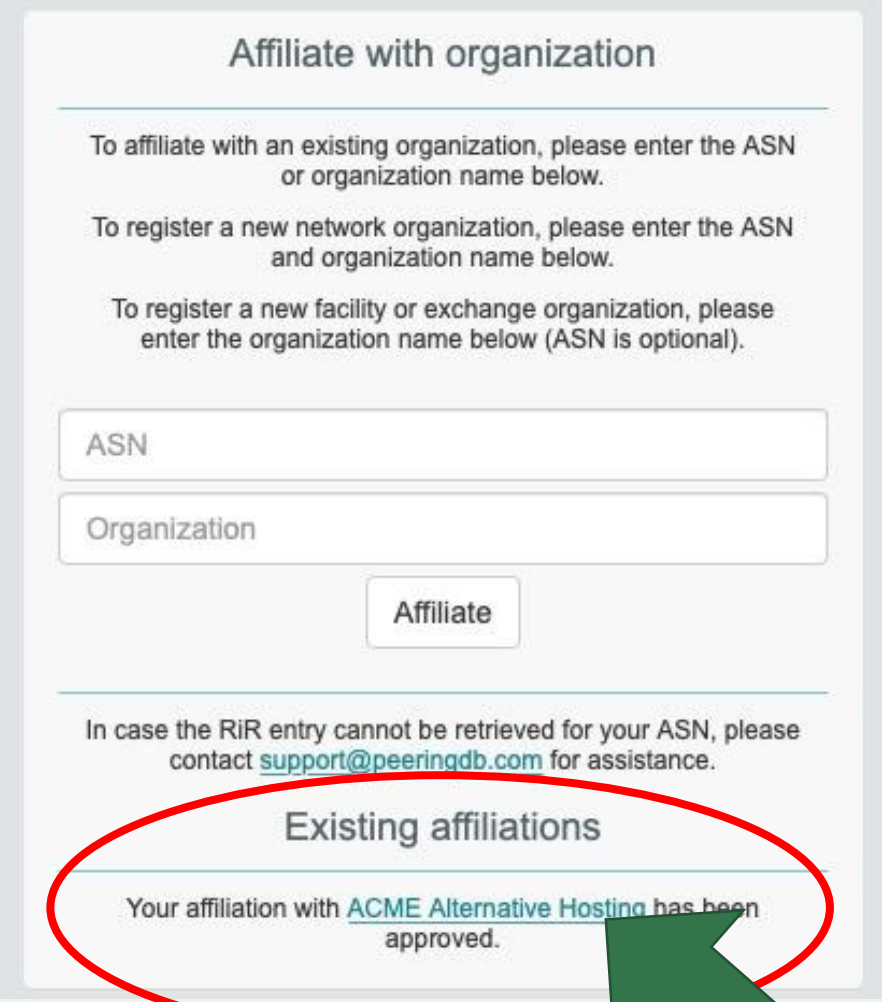

- You can choose a language
  - If your language is not available yet and you want to help – let us know!
  - Not all languages are fully translated
- To re-send the confirmation email, click the button
- Further options here:
  - Change email address
  - Change password
- You have to click the link in the email to continue!

# Registering




- Once your email is confirmed:
- Get affiliated with an organization
  - Your company
  - Which is already in PeeringDB
- Or is new to PeeringDB
  - Can be an ISP – enter your AS number!
  - Or a Datacenter
  - Or an Internet Exchange

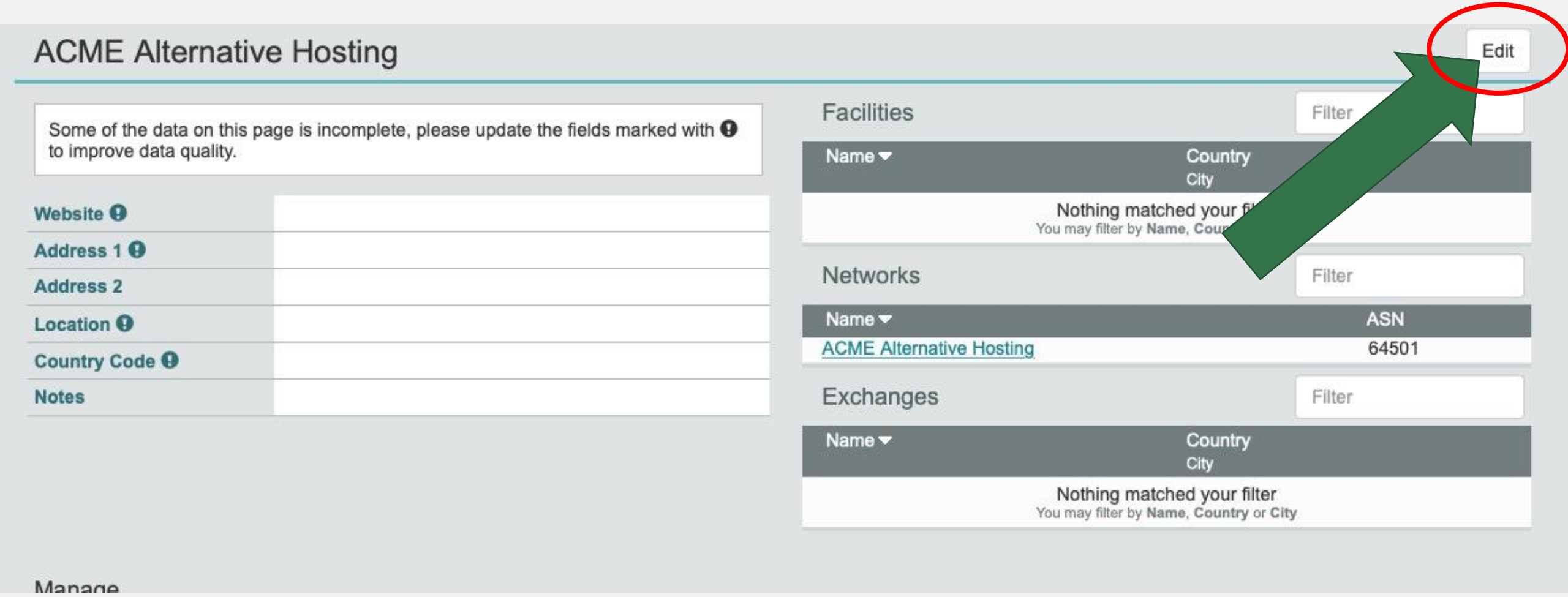
# Registering



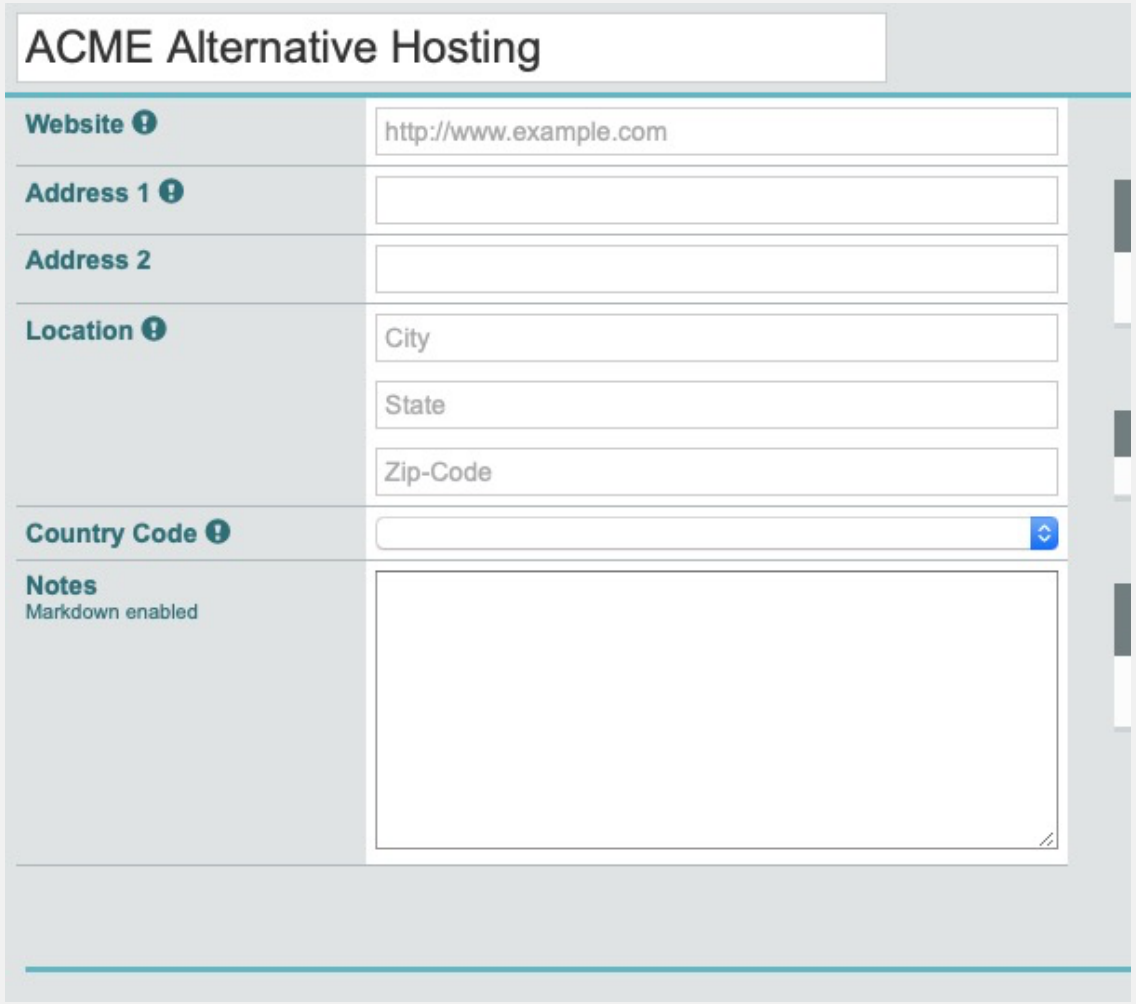

- Once approved, you can edit your organization
- Click on your organization to continue…

# Your Organization

- Enter information about your organization – click on edit

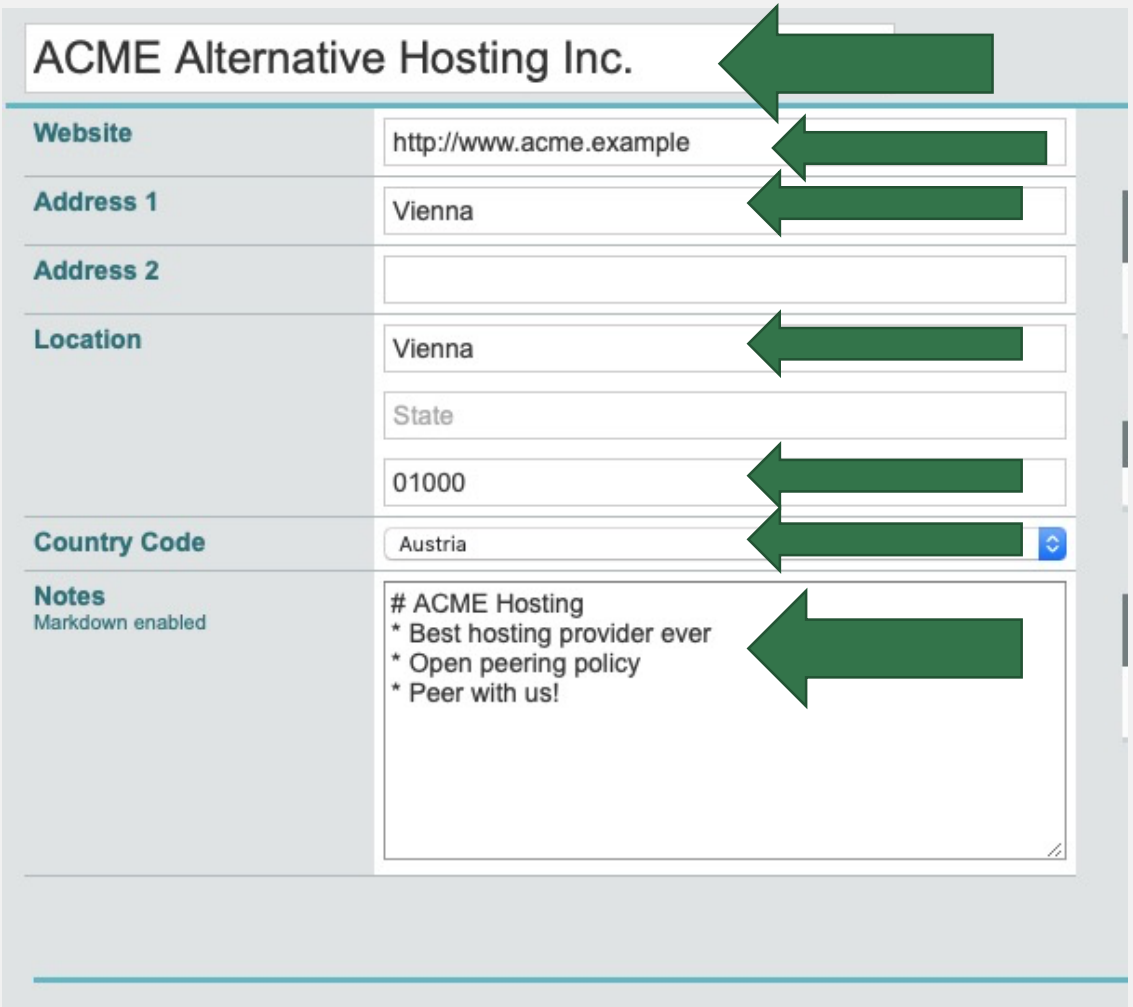


# Your Organization




- Lets focus on the left side of the screen
- Enter the required information (use either your Sheet or your real company information)
- Use the notes field to promote your company as a peer (if you want to)
  - You can use Markdown (see handout for URL)

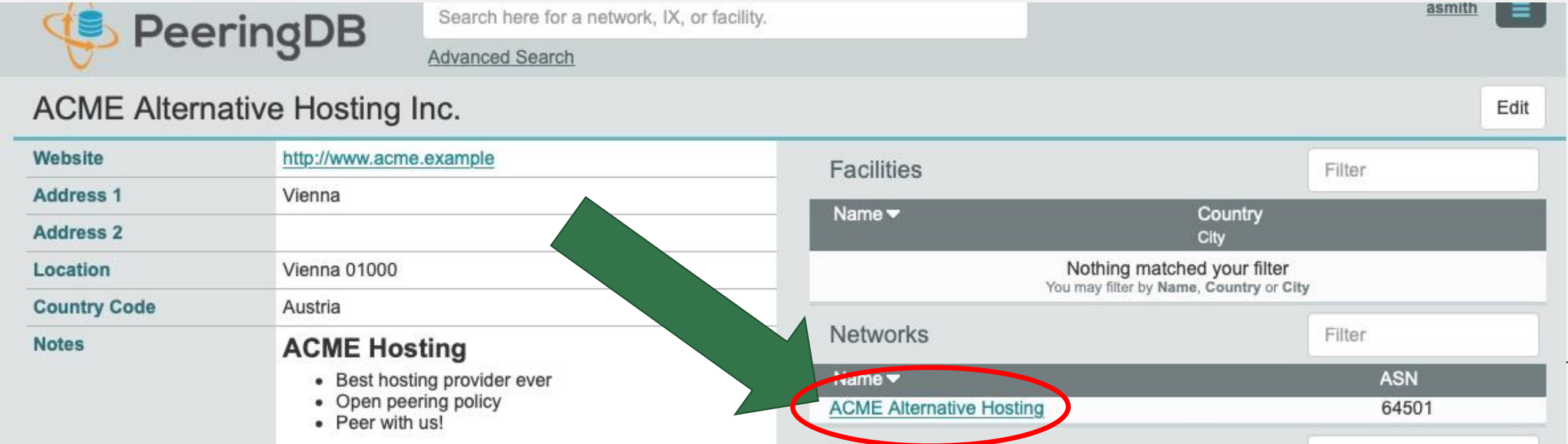
# Your Organization



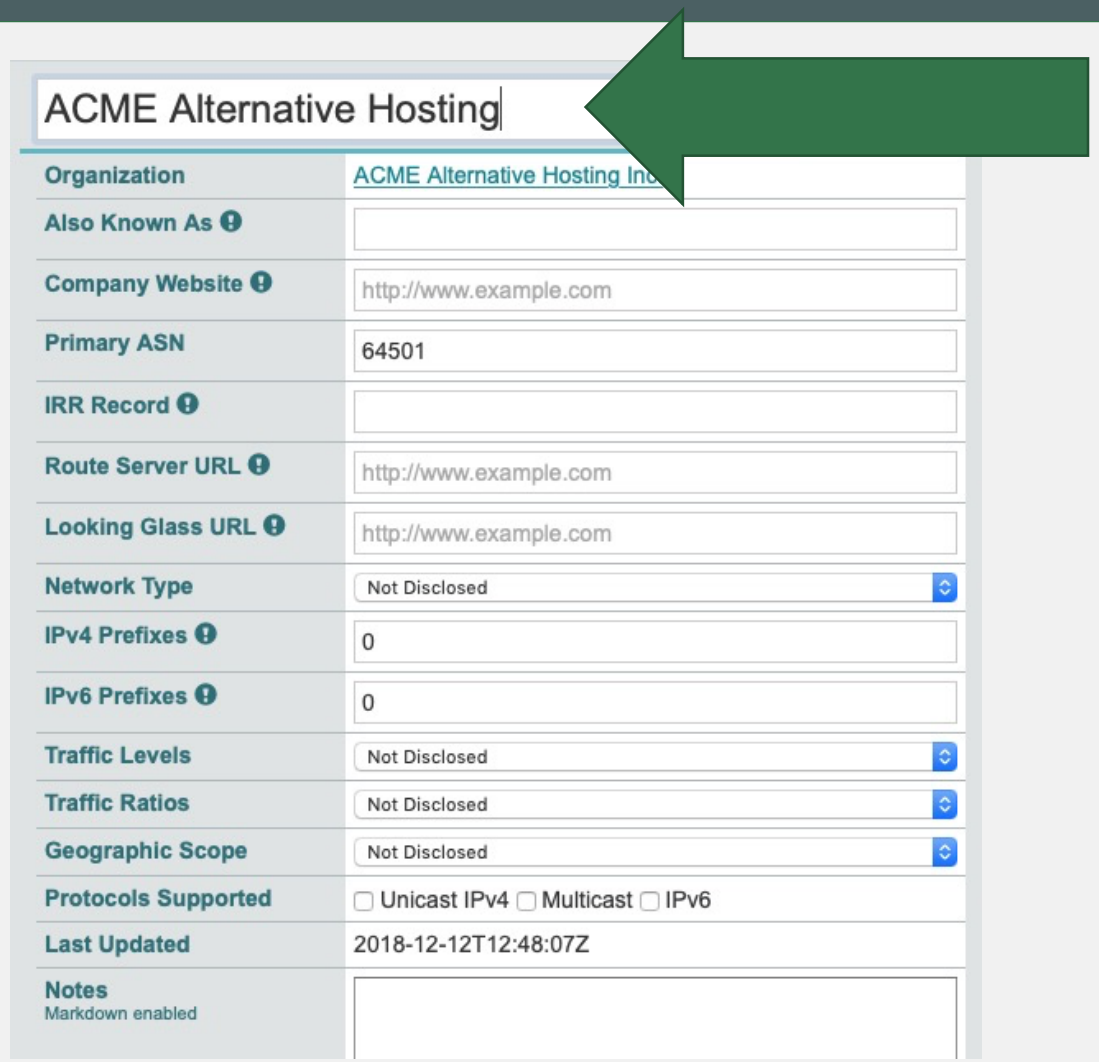

- You can also change your company name
- Enter some information and click “Save“

# Network Information

- Now it gets interesting
- With your basic company information now in, let‘s add information about your network
- Click on your network name – beside your AS number on the right side

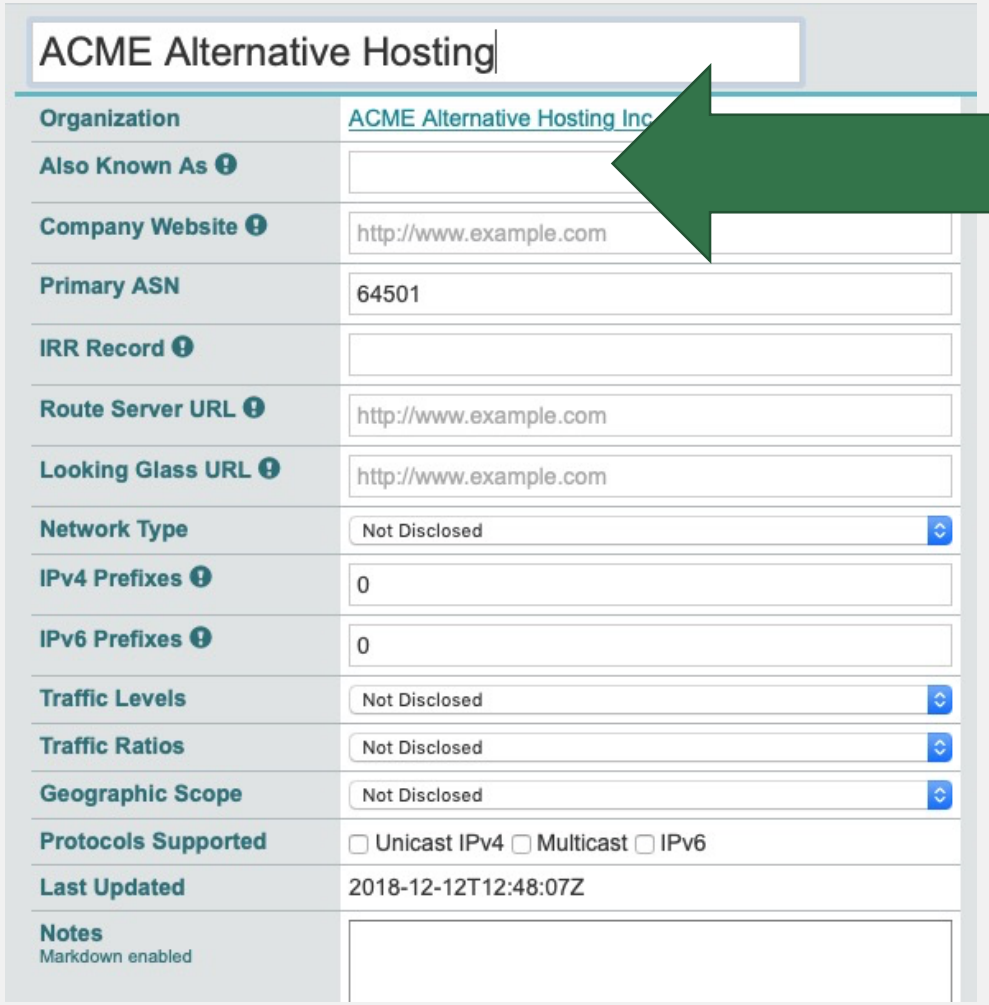


# Network Information



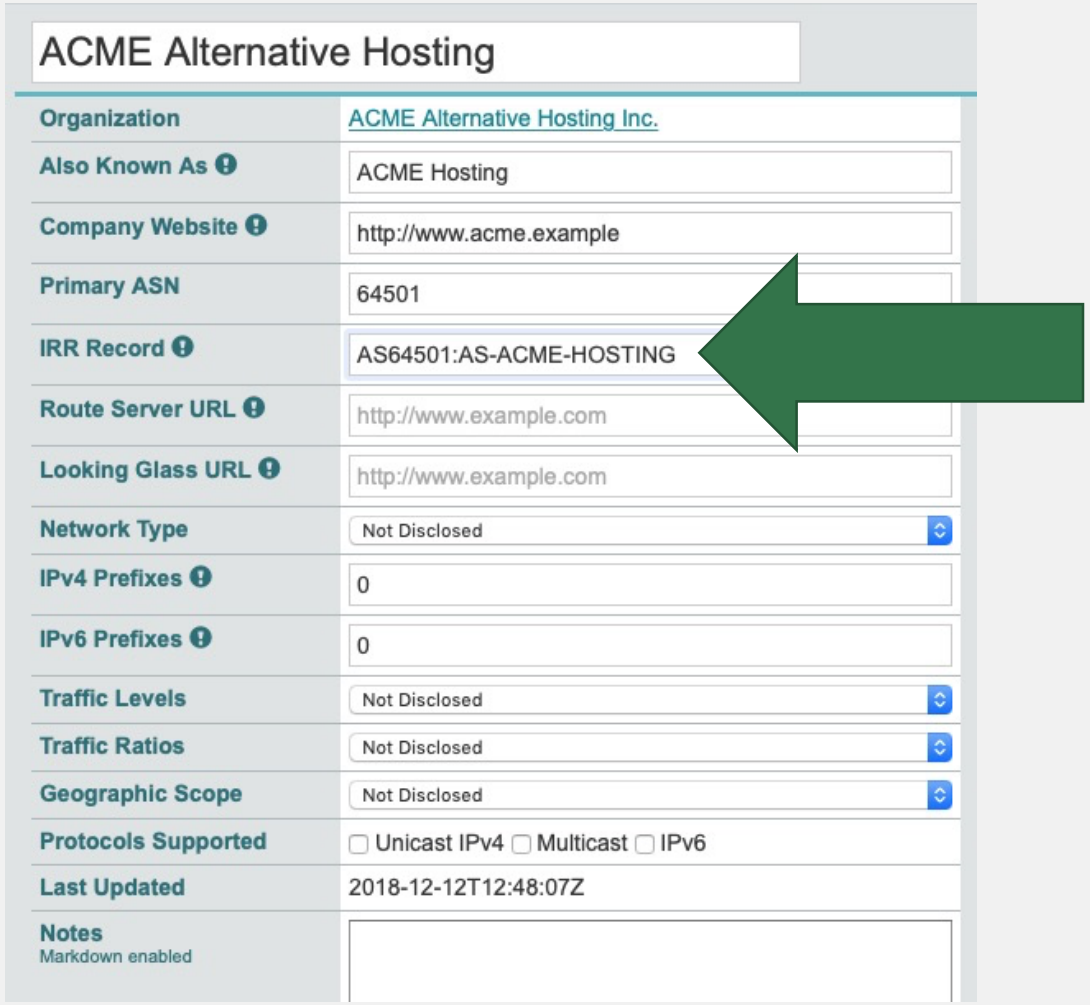

- Again, click on „Edit“ (top right)
- Here is a lot of information to enter
- Most is self-explanatory
- But some is not that obvious
  - Title here is your **network name**
  - Does not have to be the same as your company name
  - Some companies run more than one network
  - Or use a different name for their networking business

# Network Information



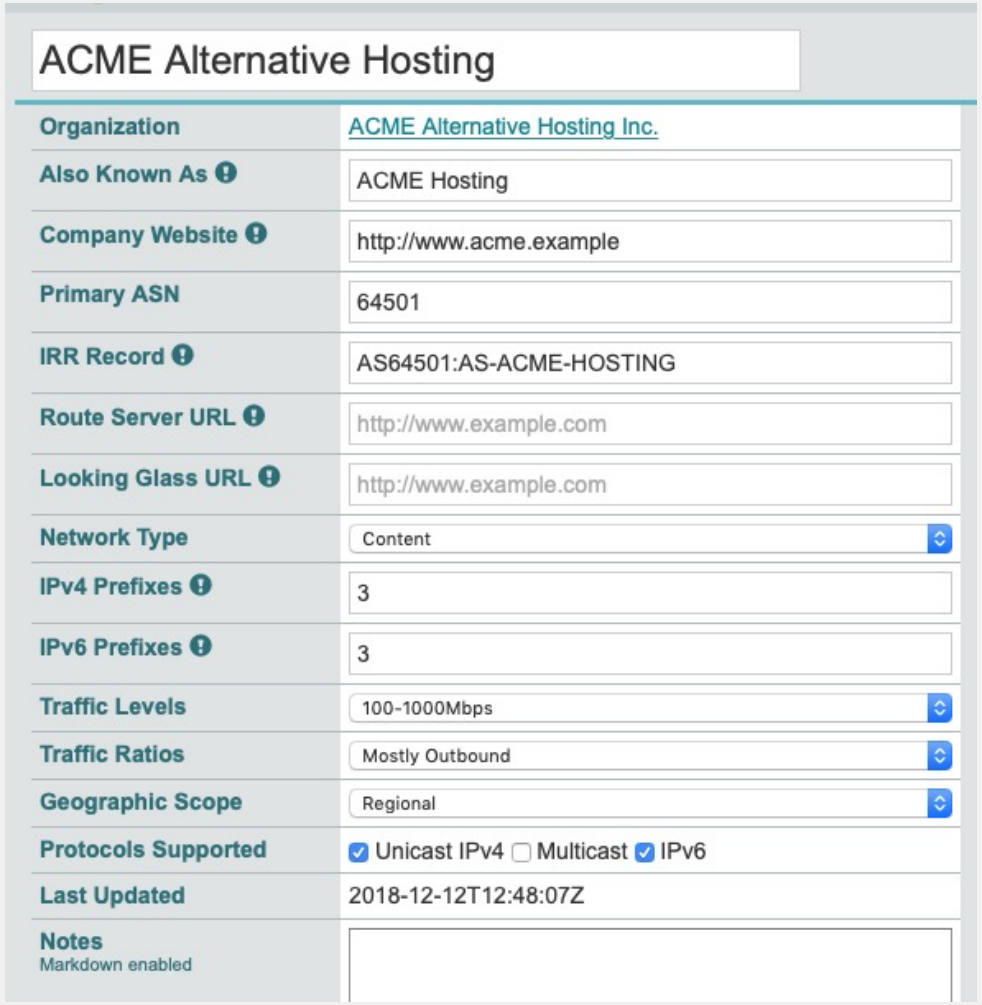

- Use this field for an alternative name
- Or an „old“ name if you changed names
- Or leave it empty
- Company website – remember this is PeeringDB
  - Put in the URL your peers should see
- AS number – your main one
  - If you have more, you can add a 2nd entry later
  - This field will probably be removed

# Network Information



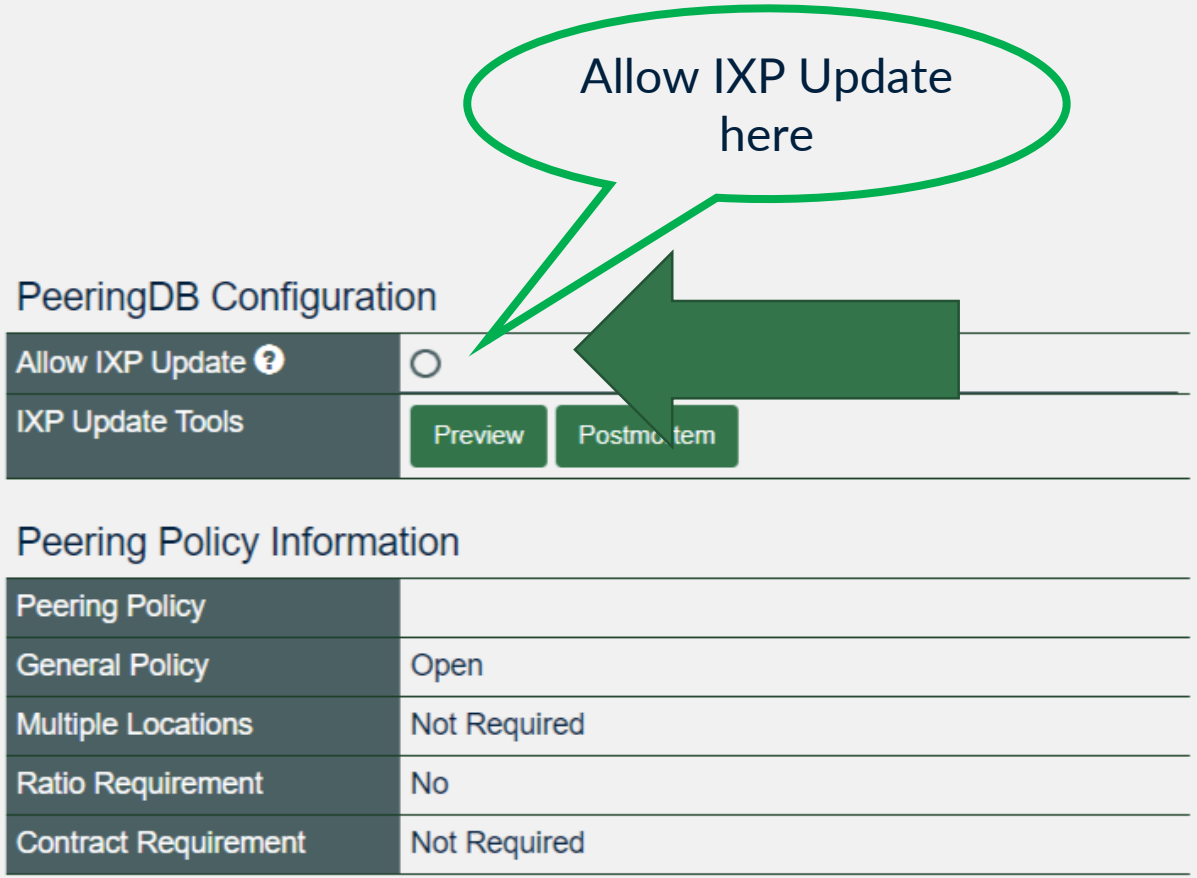

- IRR Record
  - Your AS-Macro (also called AS-Set)
  - You have registered in *IRR* database
  - Of your RIR (Regional Internet Registry)
- Route Server URL - if you have one
- Looking Glass URL

# Network Information



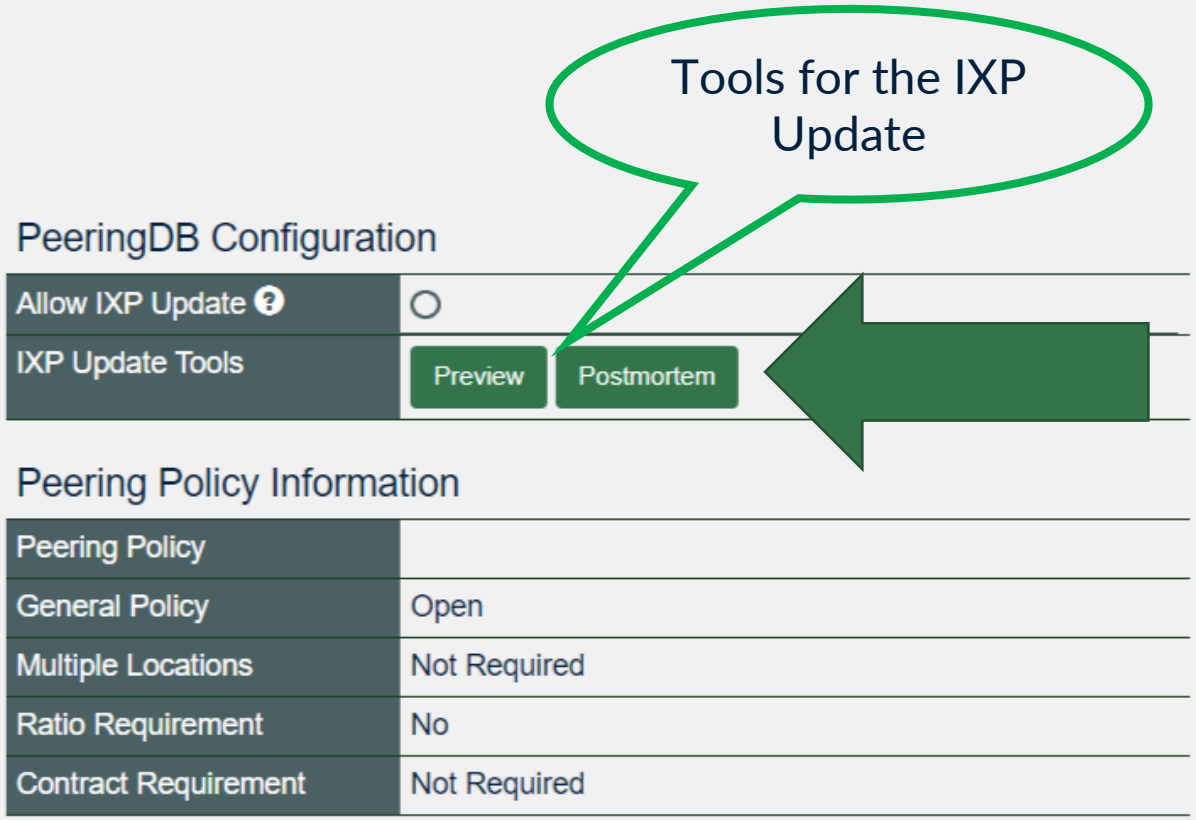

- Fill in the rest
  - Either according to your sheet
  - Or choose your real network
- In „Notes“ you can use Markdown
  - You can give your peers free text information
  - Like details about your peering policy
  - More about your peering policy further down below

# Peering Information




- Now we add information about Peering!
- Important: Allowing IXP Update helps maintaining DB accuracy
- So please allow if you trust your IXPs
- Currently the so-called IX-F importer is disabled

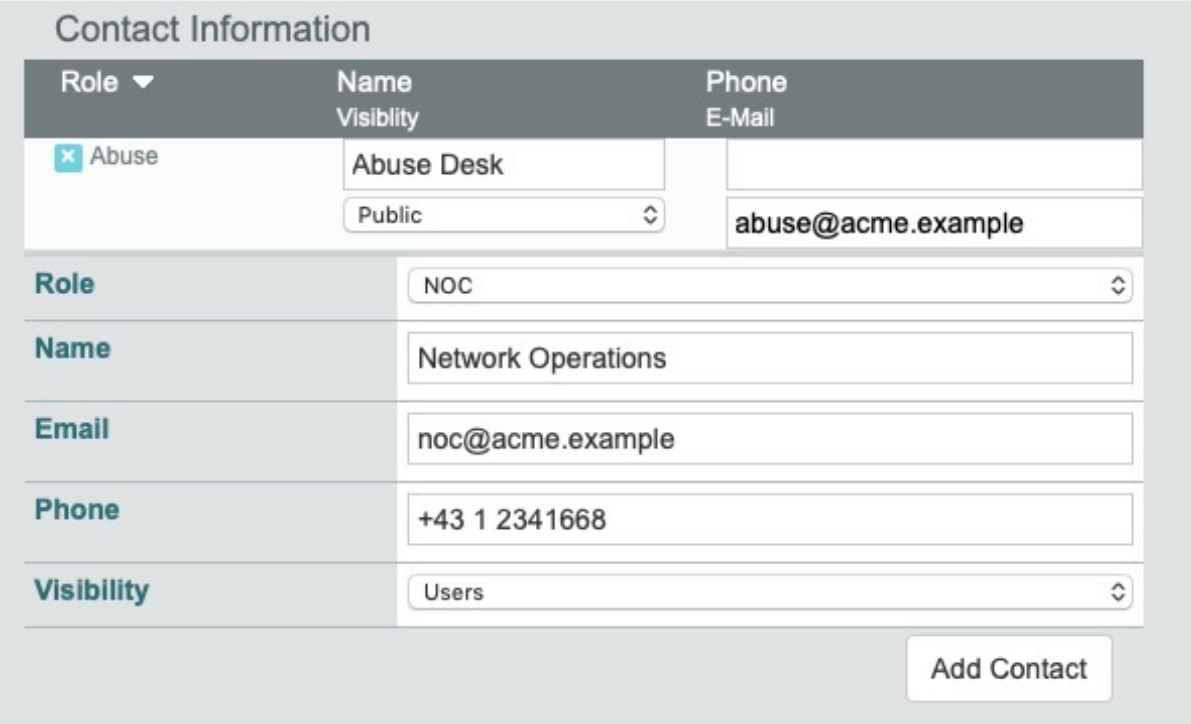
# Peering Information




- **Preview** lets you see what will happen wit the next import
- **Postmortem** shows what happened at the last import

- Use the dropdowns to publish your peering policy
- In case of selective/restrictive you may use the notes field above

# Contact Information

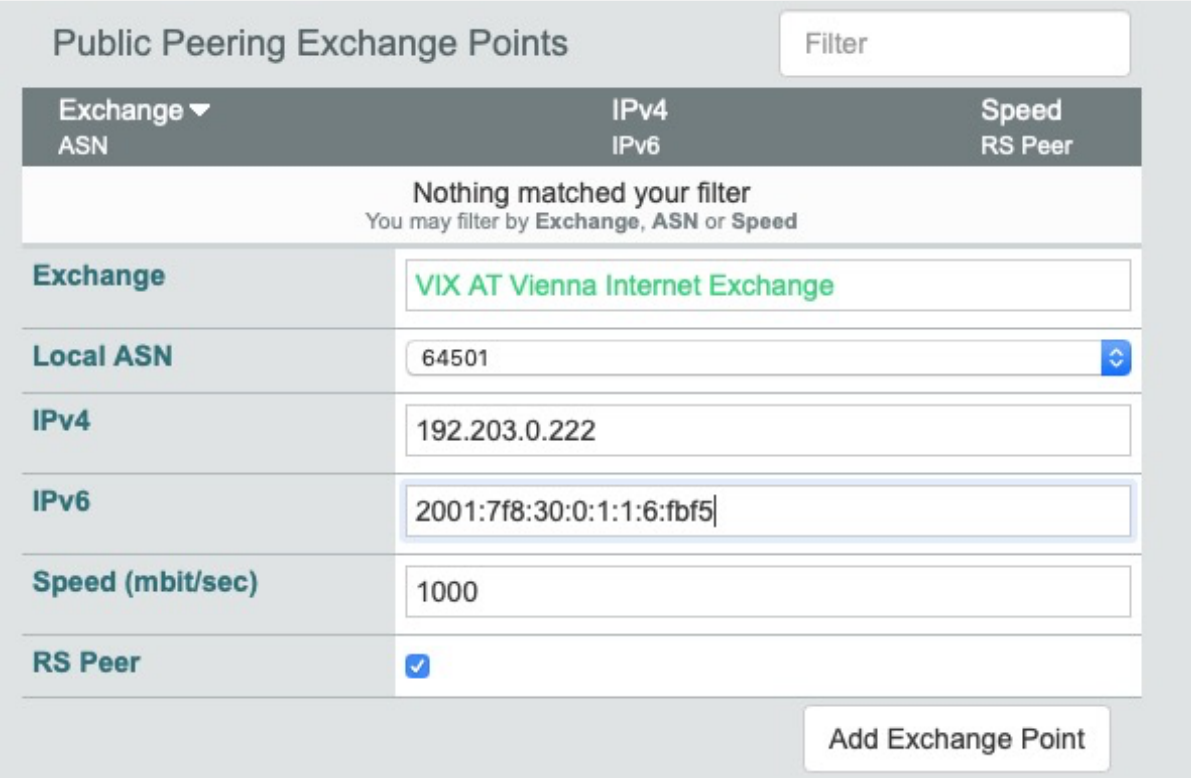



- **Very important!**
- To inform peers how to contact you
  - In a number of roles
- You can add as many as you need
  - „Add Contact“ to store and next
  - Public, for logged in users, or private
- You do not have to fill out all fields
- But please:
  **keep contact info up to date**
- Do not forget to „save“ when complete

# Review what you have entered

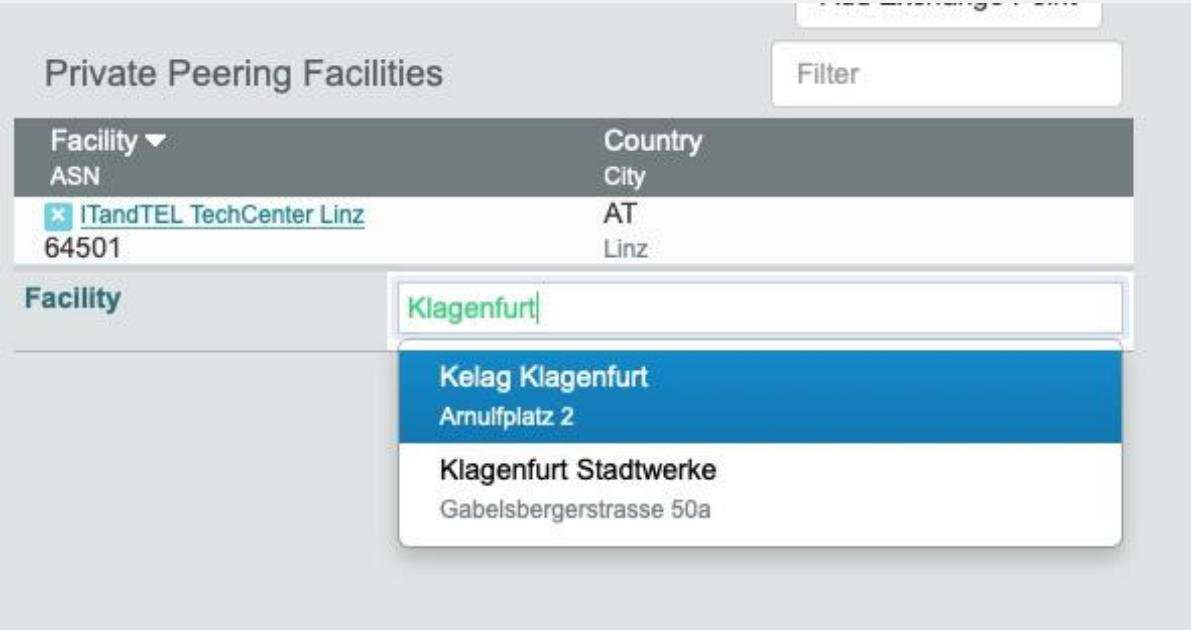
- You now have entered:
  - Information about your organization, like:
    - Address
    - Website
    - Free form text
  - Network information
    - Your AS number
    - Number of prefixes you announce,
    - Traffic info
  - Peering information, like peering policy
  - Contacts for your peers
- Please check if everything is correct

# Add peering at an IXP



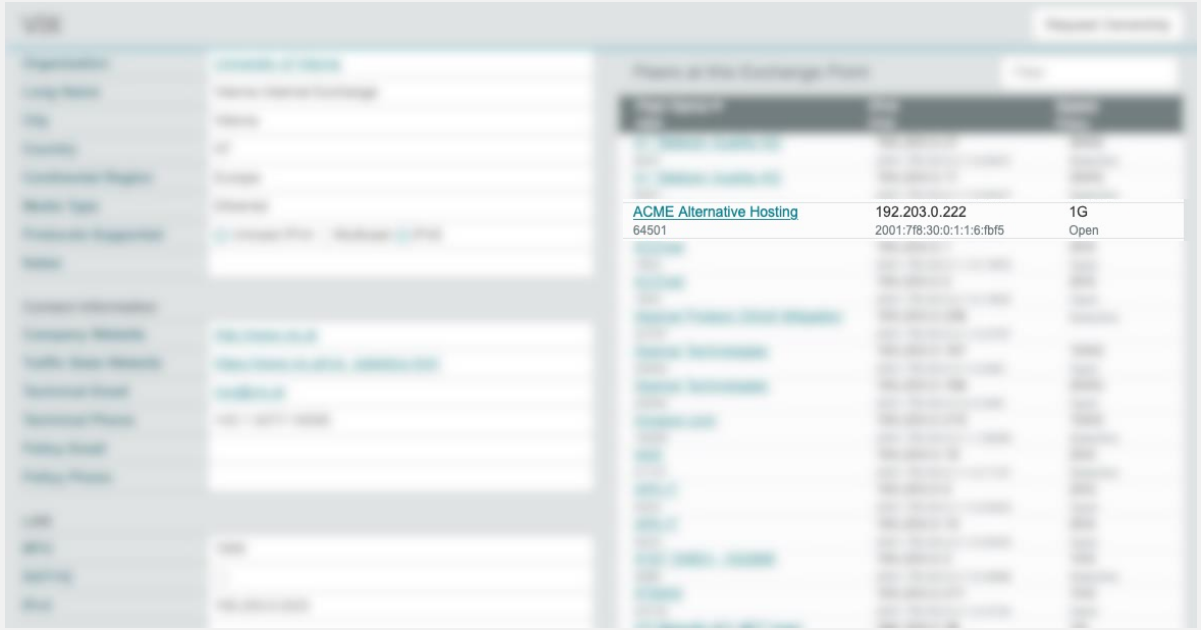

- Click on „Edit“ at the top right again
- Enter an IXP name in Exchange
  - And select the IXP you are connected to from the list
- Enter speed, IPv4 and IPv6
- Select “RS Peer“ if you are peering with the route server
- **Click on „Add Exchange Point“!**
- And then click on „save“.

# Presence at a datacenter




- Now again click on „edit“
- Scroll down and enter a city or datacenter name in „Facility“
- Select the facility you are in from the list and….
- …click on „Add Facility“
- When  you have added all facilites click on „Save“

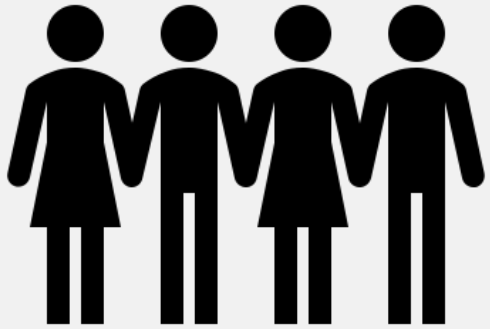
# Check what you have entered




- Click on the name of the IXP you entered
- Find your entry in the list
- Do the same for the facilities you are in

# Now lets see if others find you...

- Write your AS number on a sheet of paper
- All stand up and form a circle
- Give the person opposite to you your AS number
- and receive their AS number
- Use PeeringDB to learn about
  - Their organization
  - Their network



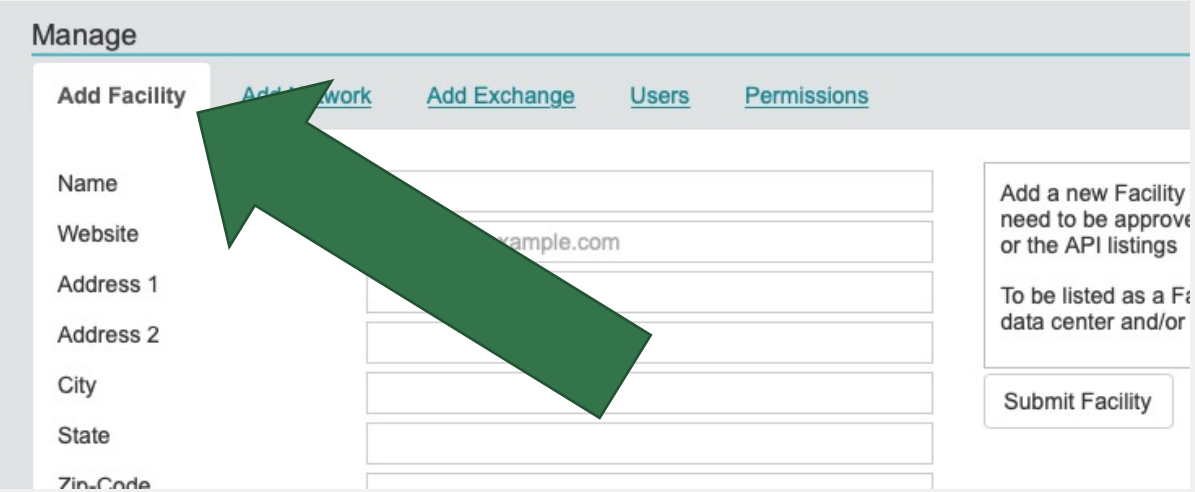
# What to find out

- You should answer the following questions:
  - What is the main business of their organization / network
  - What is their peering policy
  - What is the size of the networks in terms of
    - Traffic
    - Prefixes
  - Where are they present
    - IXPs
    - Facilities
  - Would you peer with them?
    - According to your own peering policy
    - Why? / Why not?

# Present what you have learned

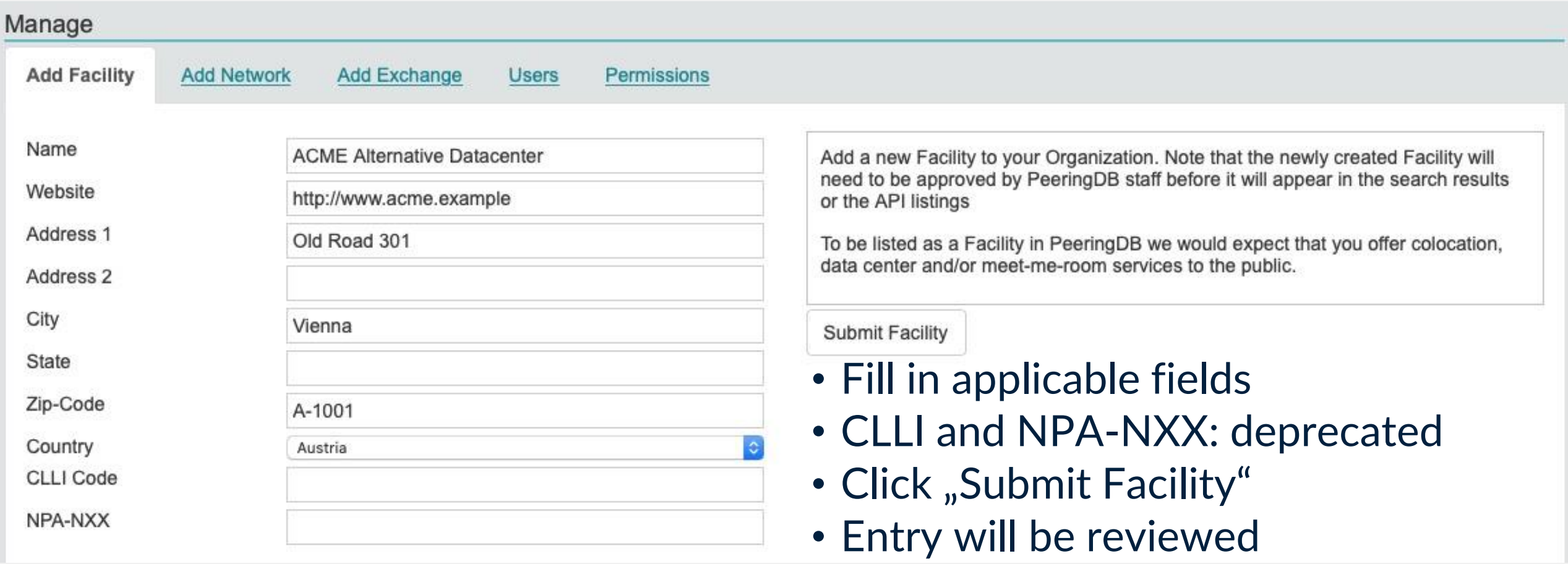
- Present to the group what you have learned
- About the organization and network you have received
- The person who has entered the information checks
  - If everything is correct
  - If anything important is missing

# Add your facility



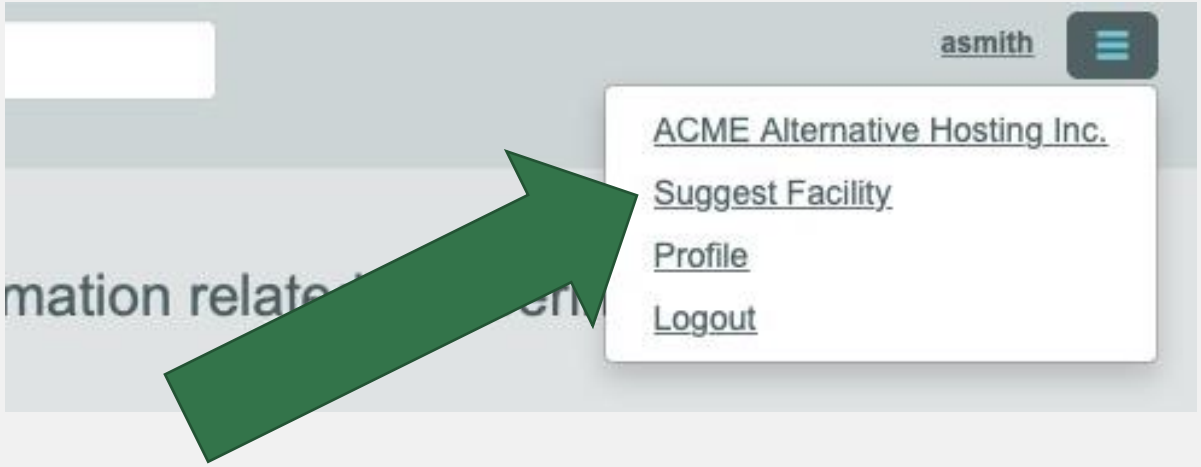

- If you run your **own** datacenter
- Why not add it to PeeringDB?
- Go to your organization page
- Scroll down to „Manage“
- Choose „Add Facility“

# Add your facility



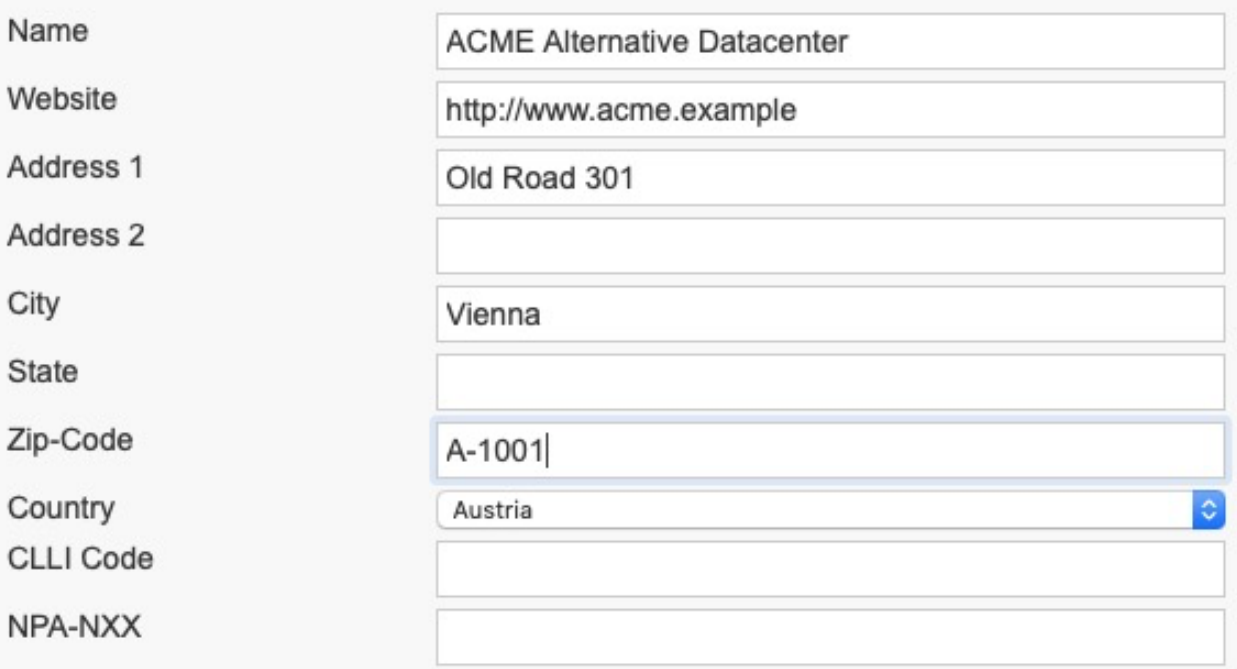

- Fill in applicable fields
- CLLI and NPA-NXX: deprecated
- Click „Submit Facility“
- Entry will be reviewed
- And added or declined

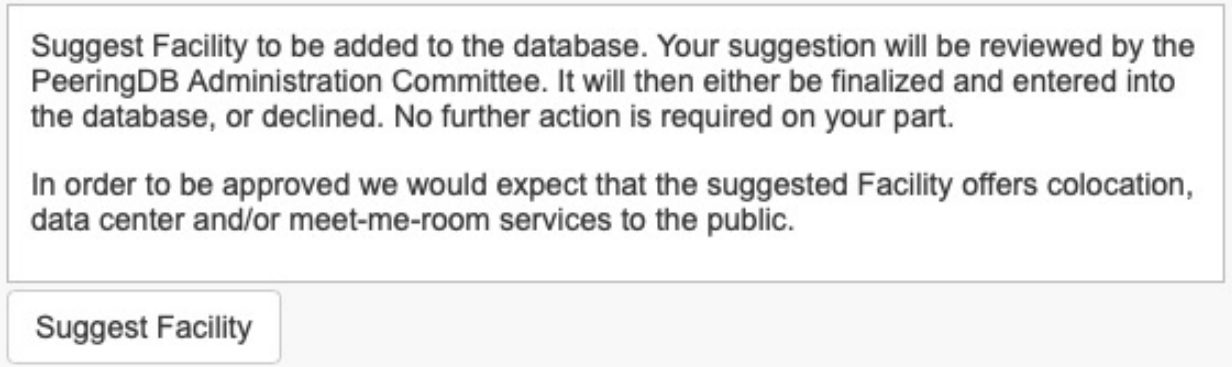
# Suggesting a facility




- For facilities **not your own**
- **But you are in or know about**
- Choose „Suggest Facility“
- Needs to be reviewed by PeeringDB staff

# Suggesting a facility



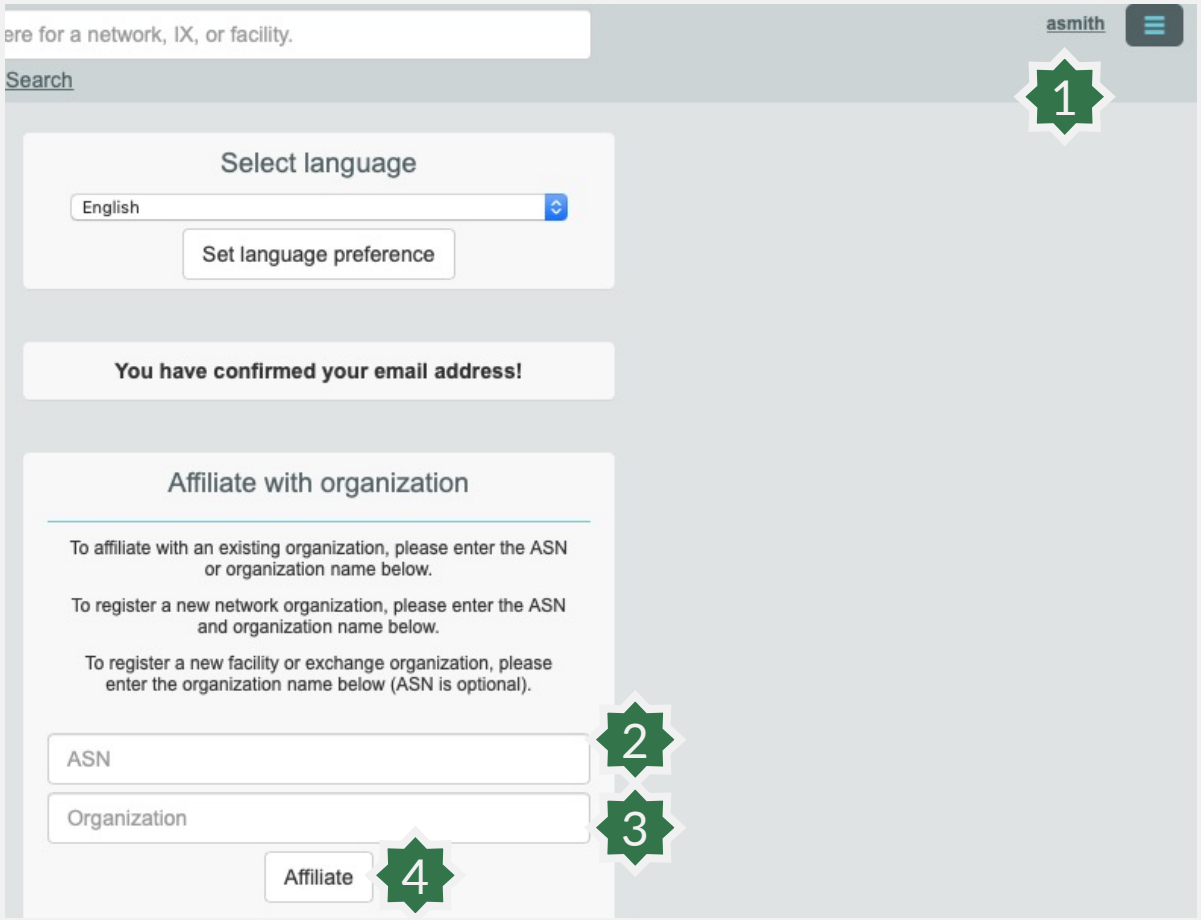




- Fill in applicable fields
- CLLI and NPA-NXX: deprecated
- Click „Suggest Facility“
- Entry will be reviewed
- And added or declined

# Adding users

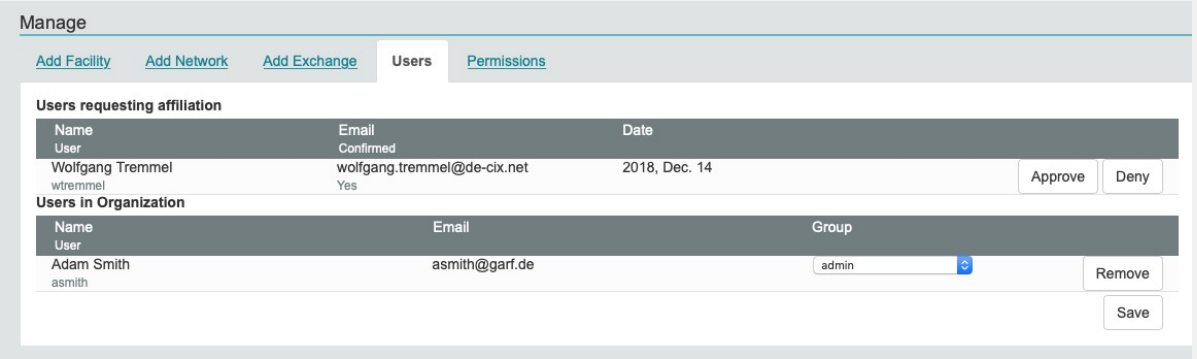
- You do not have to be the only person working with PeeringDB
- Other users from your organization may also register!
- Users can be „admin“ or „member“
  - The first user automatically will be an „admin“
- Administrators are allowed to edit all fields (of your organization)
- Members rights can be as restrictive or as open as you need them to be
- Let‘s try it out
- Request affiliation with the organization you just presented

# Affiliate with an organization



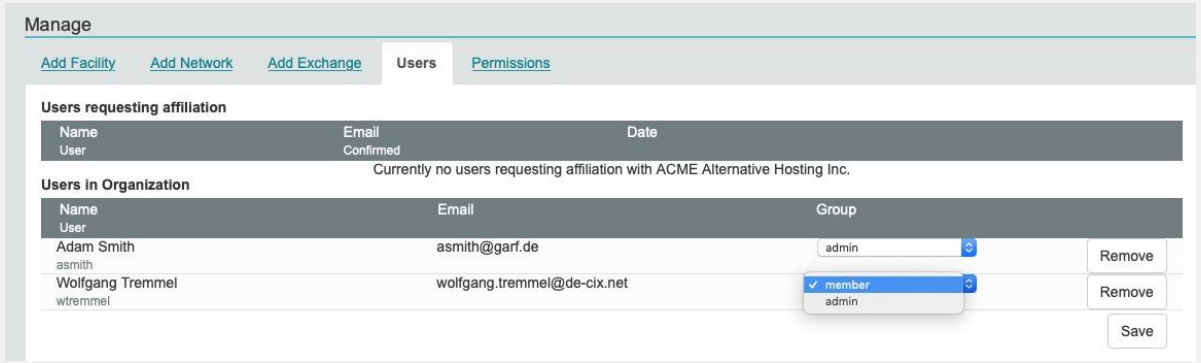

- Go to your profile page (1)
- Request affiliation
  - Either use the AS number (2)
  - Or name and select (3)
  - Click on „Affiliate“ (4)
- Admin of organization gets an email if there is one. Otherwise PeeringDB support
- Checks, and either approves or denies

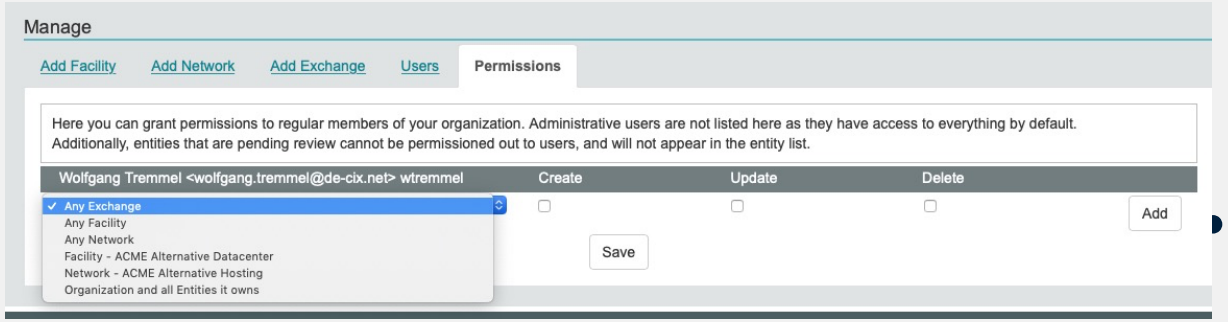
# Approve affiliation requests




- Go to your organization page
- Scroll down to the „manage“ section
- Click on the „Users“ tab
- Either approve or deny new users

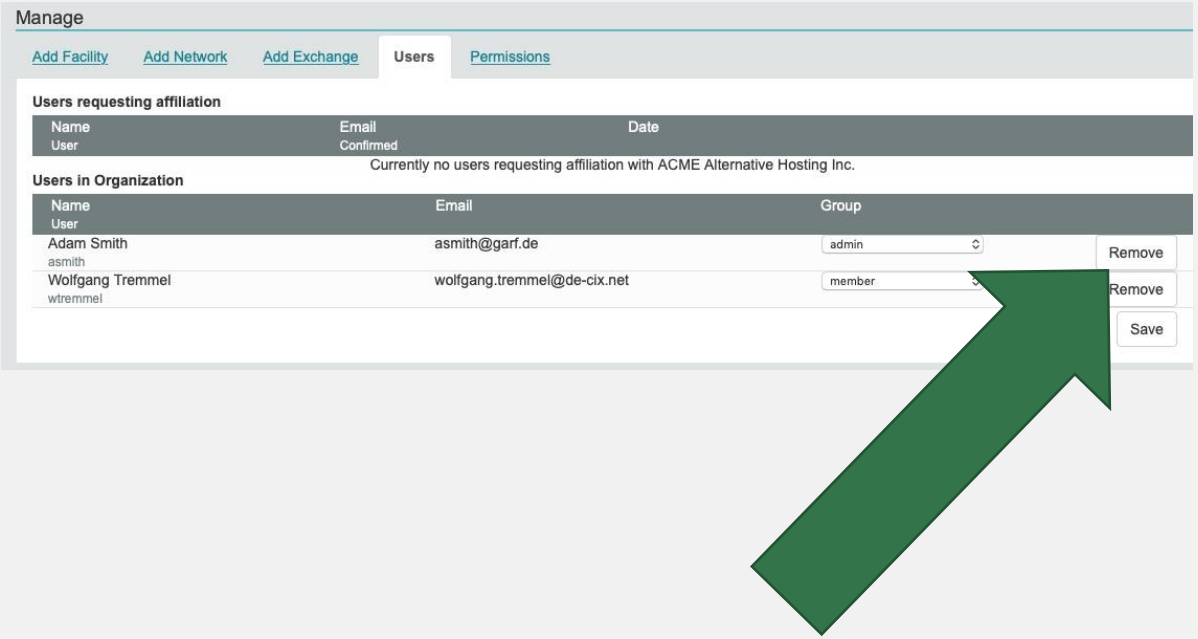
# User administration



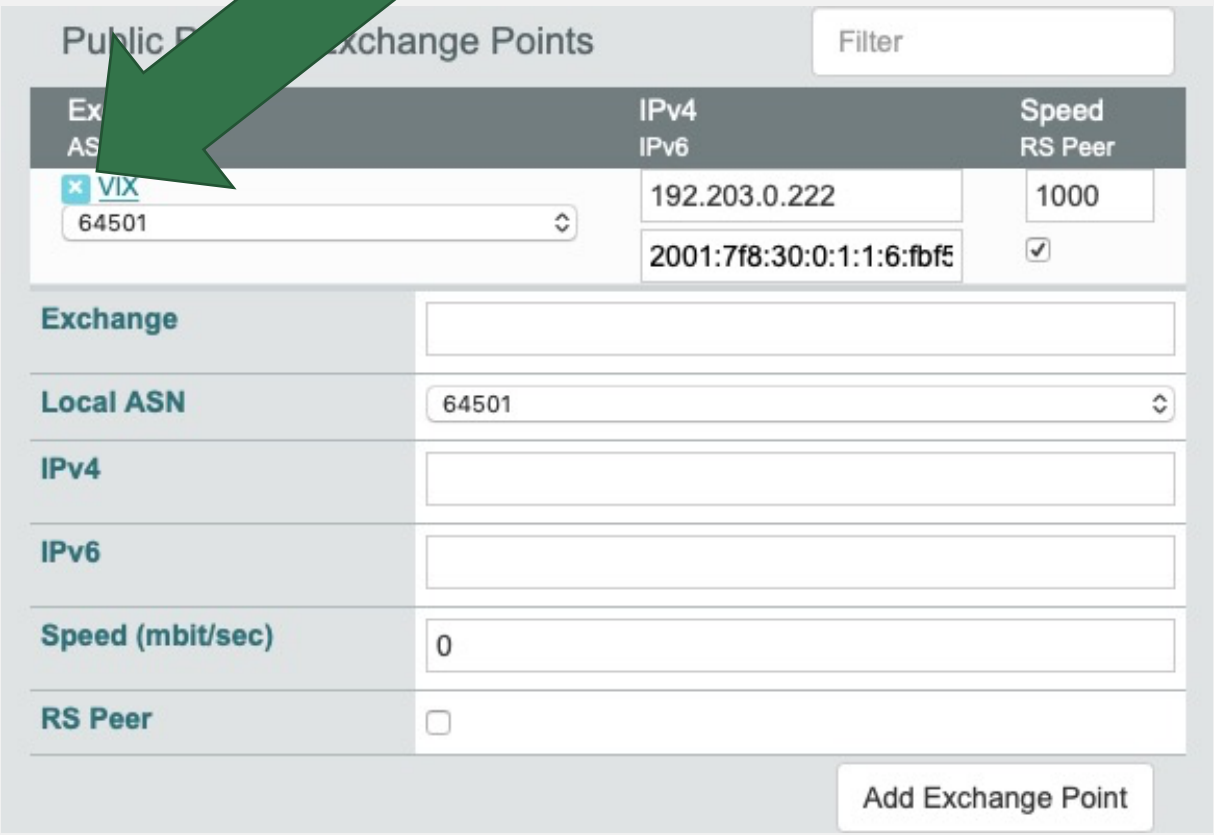

- Users can be admins or members
- Use the dropdown to change
- For members – you can add permissions
- Use the permissions tab to grant create, update and/or delete to selected of any entities
- Do not forget to „save“ your changes

# Removing Users




- Be sure you are logged in as an admin
- Go to your organization page
- Scroll down to the „manage“ section
- Click on the „Users“ tab
- Click on „Remove“ on the right side
- And do not forget to „save“
- The user is only deleted from your organization
- If you want to remove a user completely, email PeeringDB support

# More removing...

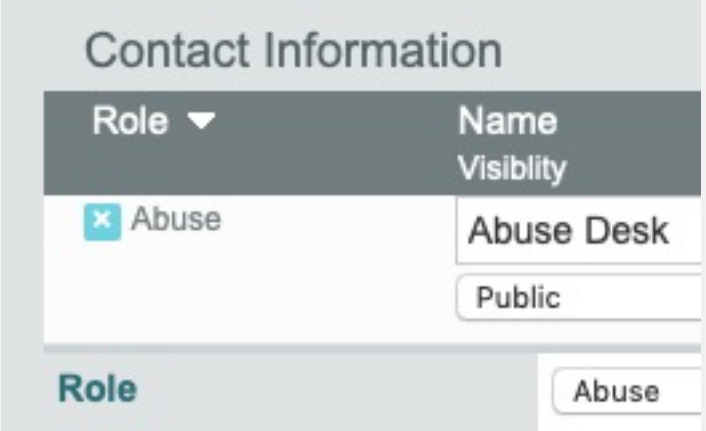



- If you leave an internet exchange:
  – please remove your peering IP addresses

- Go to your network page
- Click on „Edit“ (top right)
- Click on th beside the entry of the exchange
- Confirm and do not forget to „Save“

# More removing...

- Facilities, Contacts, all the same
- Click on „Edit“
- Click on the remove symbol at the entry
- Click on “Save“

# Removing – more information

- Objects are only *marked deleted, but stay in the DB*
- ***You cannot simply re-add them***
- Please contact support@peeringdb.com if you need help

PeeringDB

# The PeeringDB API

Workshop // Track 3

arnold@peeringdb.com

# Agenda

- Please always use the tutorial DB at https://tutorial.peeringdb.com

- Introduction
- jq
- JSON
- HTML Operations
- Record Types
  - Basic Records
  - Derived Records

# Introduction

- Why API (Application Programming Interface)?
  - The GUI is nice for human beings
  - Automation needs structured data
- Makes it easy to integrate PeeringDB in your environment

# jq

- Light-weight and flexible command-line processor
- awk, sed and grep equivalent to JSON data
- A jq program is a filter
  - Needs an input and produces anoutput
  - Maybe piped
  - Looks weird sometimes, like „add/length“ produces average of an array
  - Simplest filter is „.“ which is the Identity
    - Maybe used to pretty print JSON output
- See https://stedolan.github.io/jq/manual for an introduction
- Ex: curl -sG https://peeringdb.com/api/org --data-urlencode fields=id | jq -c '[.data[] | .id] | length'

# JSON

- Open standard file format
- Short for JavaScript Object Notation
- Filenames use the extension .json
- Language independent data format
- Basic data types
  - Number
  - String
  - Boolean
  - Array
  - Object
  - null

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

# Basics

- In general https://peeringdb.com/api/OBJ
  - OBJ is case insensitive
  - So called endpoint: /api/OBJ
- Output always fits in one object
  - Meta is optional
  - Data always an array

```
{
meta:
  {
  status:
  message:
  }
data:
  [
  {},
  {}
  ]
}
```

# Authentication

- Authentication via basic HTTP authorization
- Guest access does not need any authentication
- Examples
  - curl -sG https://username:password@peeringdb.com/api/poc
  - curl -u username:password https://peeringdb.com/api/poc
  - Put credentials in ~/.netrc
    - machine peeringdb.com login username password password
- Recap: only access to contact information may be restricted
  - Endpoint /api/poc
  - You need to be authenticated for view „user“

# Operations

- All HTML operations are supported
  - GET
    - Requests a representation of the specified resource
  - POST
    - Used to submit an entity to the specified resource
  - PUT
    - Replaces all current representations of the target resource with the request payload
  - DELETE
    - Deletes the specified resource

# GET

- GET
  - Multiple objects
    - Endpoint /api/OBJ
  - Single object
    - Endpoint /api/OBJ/id

# Optional URL parameters for GET

- limit
  - Integer value
  - Limits to n rows in the result set
- skip
  - Integer value
  - Skips n rows in the result set
- depth
  - Integer value
  - Nested sets will be loaded
  - See Nesting slide

# Optional URL parameters for GET

- fields
  - String value
  - comma separated list of field names
  - only matching fields will be returned in the data
- since
  - Integer value
  - Retrieve all objects updated since specified time
  - Unix timestamp in seconds
- *fieldname*
  - Integer or string value
  - Queries for fields with matching value

# Nested Data / Depth

- Of type OBJ_set
- Example: *net_set* will hold network objects
- Depth (for endpoint /api/OBJ)
  - 0: don't expand anything (default)
  - 1: expand all first level sets to ids
  - 2: expand all first level sets to objects
- Depth (for endpoint /api/OBJ/id)
  - 0: don’t exand anything
  - 1-4: expand all sets and related objects according to level of depth specified
  - 2 is default

# Nested Data / Depth

https://peeringdb.com/net/947?pretty

https://peeringdb.com/net/947?pretty&depth=0


```
{
  "meta": {},
  "data": [
    {
      "id": 947,
      "org_id": 1187,
      "name": "DE-CIX Frankfurt Route Servers",
      "aka": "DE-CIX",
      "website": "https://fra.de-cix.net",
      "asn": 6695,
      "looking_glass": "https://lg.de-cix.net",
      "route_server": "https://www.de-cix.net/en/locations/germany/frankfurt/routeserver-gu
      "irr_as_set": "AS-DECIX",
      "info_type": "Route Server",
      "info_prefixes4": 240000,
      "info_prefixes6": 50000,
      "info_traffic": "1 Tbps+",
      "info_ratio": "Balanced",
```



```
{
  "meta": {},
  "data": [
    {
      "id": 947,
      "org_id": 1187,
      "org": {
        "id": 1187,
        "name": "DE-CIX Management GmbH",
        "website": "https://de-cix.net",
        "notes": "",
        "net_set": [
          947,
          5547,
          6978,
          6979,
          8383,
          8703,
          8919,
          8920,
          9840,
          9841,
          10018,
          13190,
          13251,
          19331,
          20739
        ],
        "fac_set": [],
        "ix_set": [
          31,
          74,
          248,
          804,
          1131,
          1149,
          1150,
          1214,
          1249,
          1277,
          2531
        ],
        "address1": "Lindleystr. 12",
```

# Query modifiers

- numeric fields
  - __lt: less than
  - __lte: less than equal
  - __gt: greater than
  - __gte: greater than equal
  - __in: value inside set of values (comma separated)
- string fields
  - __contains: field value contains this value
  - __startswith: field value starts with this value
  - __in: value inside set of values (comma separated)

# POST

- Used to create an object
- Endpoint /api/OBJ
- Required parameters
  - Depending on OBJ
  - For *org* you need the *name*
  - For *fac*, *ix*, *net* you need the *org_id*
  - for *fac* you need the *name*
  - For *ix* you need the *name* and *prefix*
  - For *net* you need the *asn*

```
{
  "name": "Org-22106"
}
```

File 22106.json

- Example
  - curl -sn -X POST -H "Content-Type: application/json" -d @22106.json \
    https://tutorial.peeringdb.com/api/org

# PUT

- Used to edit object
- Endpoint /api/OBJ/id
- Updates data in OBJ/id
- You have to send all the data, not just the change
- Example
  - curl -sn -X PUT -H "Content-Type: application/json" -d @22106.json \
    https://tutorial.peeringdb.com/api/org/22114
- Operation of PUT is idempotent

```
{
        "name": "Org-22106",
        "address1": "23 Mulholland Drive",
        "city": "Los Angeles",
        "country": "US"
}
```

File 22106.json

# DELETE

- Used to delete objects
- Endpoint /api/OBJ/id
- Example
  - curl -sn -X DELETE -H "Content-Type: application/json“ \
    https://tutorial.peeringdb.com/api/org/22114

# Object Types

- Basic Objects
  - org, fac, ix, net, poc, as_set
- Derived Objects
  - ixlan, ixpfx, netixlan, netfac

# Basic Objects

- org
  - Root object for fac, ix, net
  - Holds information about organisation
- fac
  - Describes a facility / colocation record
  - More useful information are in derived records netfac
- ix
  - Describes an Internet Exchange
  - More useful information are in derived records ixlan, ixpfx and netixlan
- net
  - Describes a network / ASN
  - More useful information are in netfac and netixlan
- poc
  - Describes various role accounts (point of contact)
  - Currently only for net objects
- as_set
  - Array of all AS-SETs corresponding to a network / ASN
  - Only introduced recently

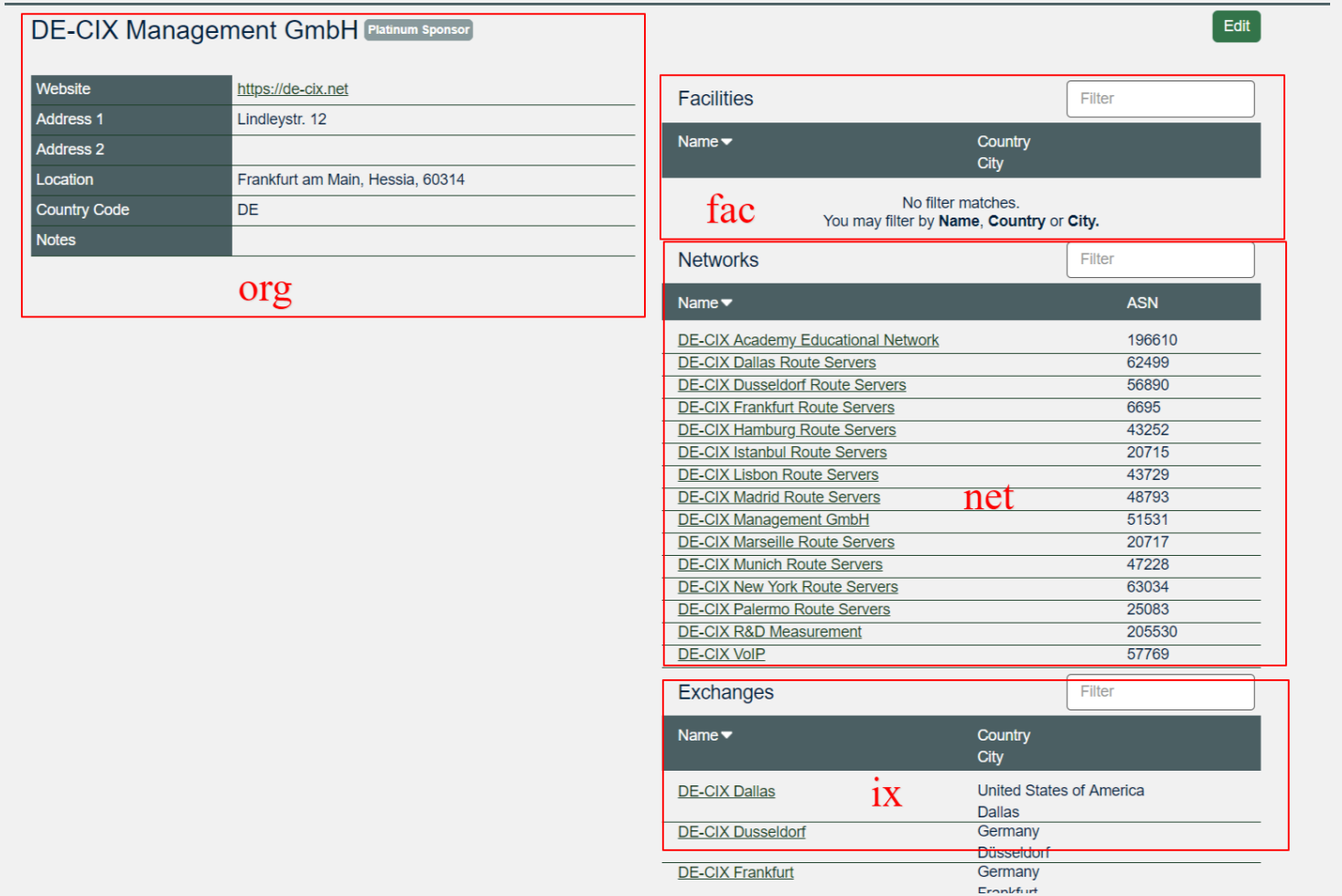
# Derived Objects

- ixlan
  - Describes the LAN of an IX
  - One IX may have multiple ixlan
  - May go away with PeeringDB 3.0
- ixpfx
  - Describes the IP range (IPv4 and IPv6) for an ixlan
  - One ixlan may have multiple ixpfx, both for IPv4 and IPv6
- netixlan
  - Describes the presence of a network at an IX
- netfac
  - Describes the presence of a network at a facility

# GUI to API // org

- https://peeringdb.com/org/1187

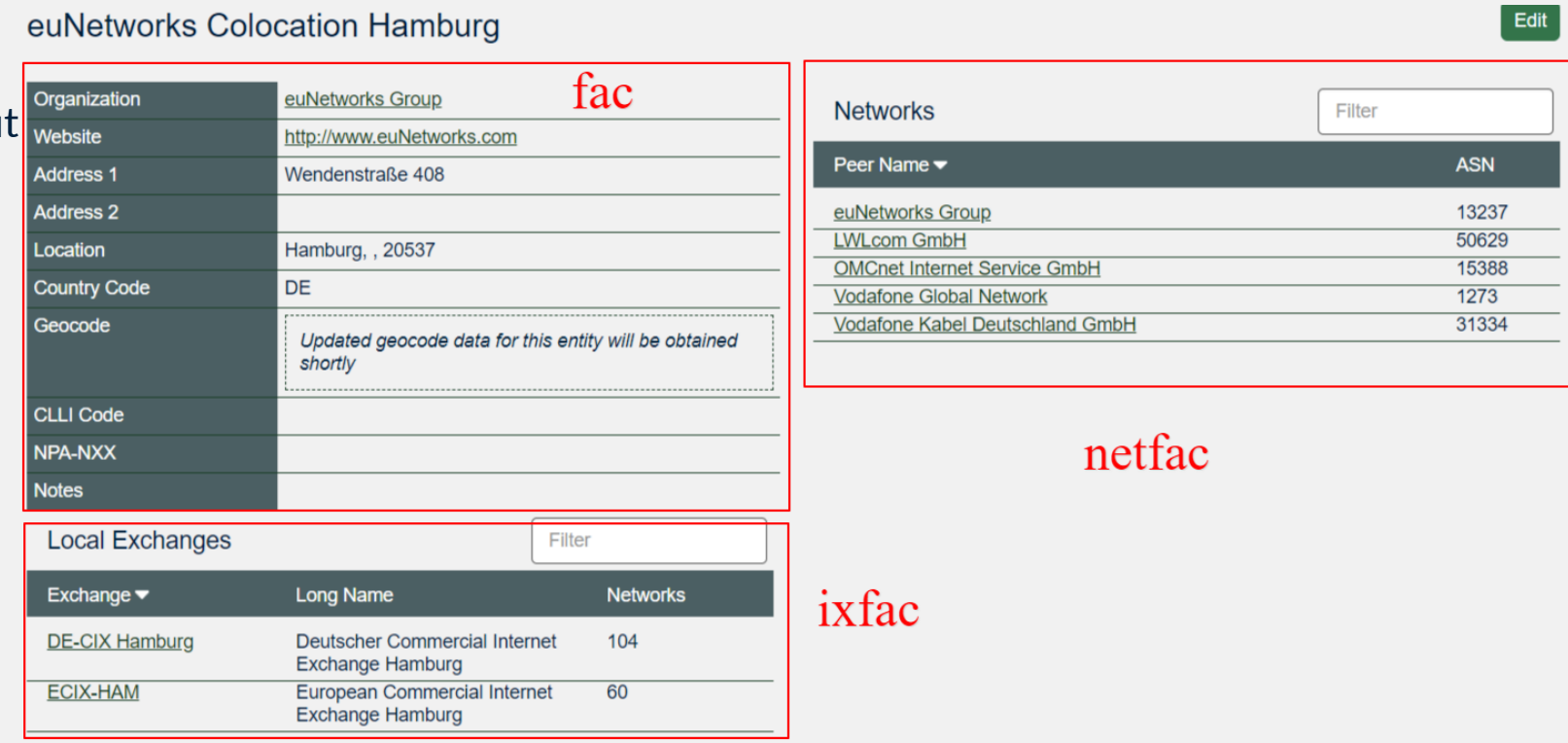
- Add pretty and depth for human friendly output
- https://peeringdb.com/api/org/1187
- https://peeringdb.com/api/fac?org_id=1187
- https://peeringdb.com/api/net?org_id=1187
- https://peeringdb.com/api/ix/org_id=1187



# GUI to API // fac

- https://peeringdb.com/fac/752

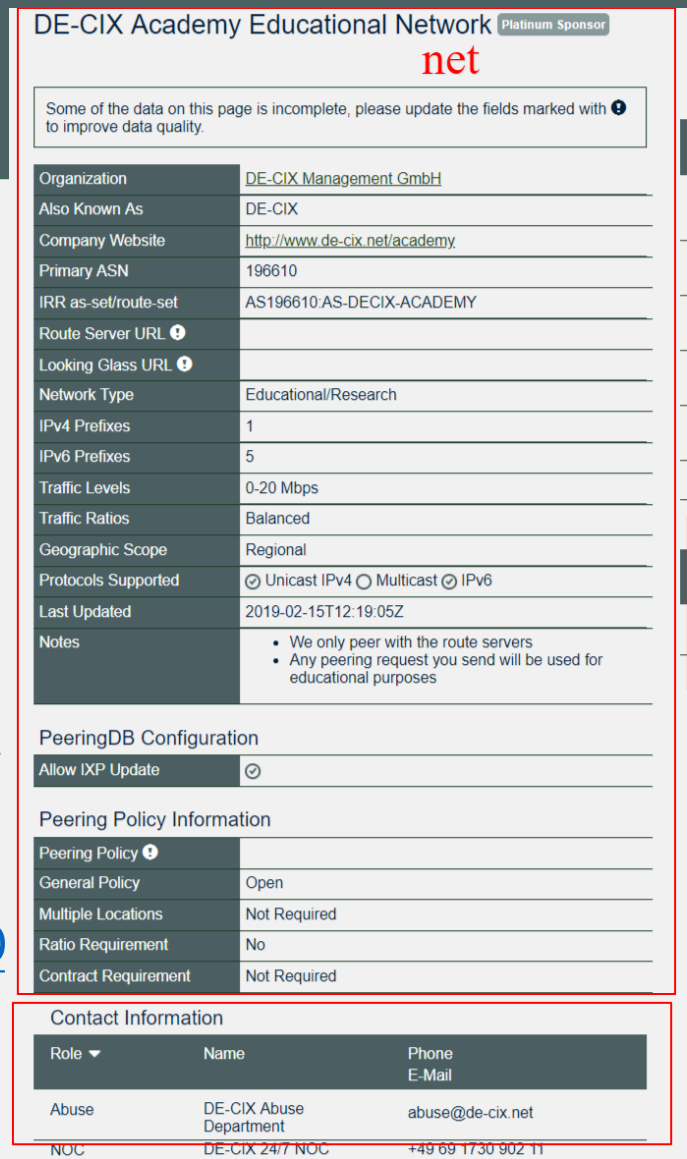
- Add pretty and depth for human friendly output
- https://peeringdb.com/api/fac/752
- https://peeringdb.com/api/ixfac?fac_id=752
- https://peeringdb.com/api/netfac?fac_id=752



# GUI to API // net

- https://peeringdb.com/net/13251

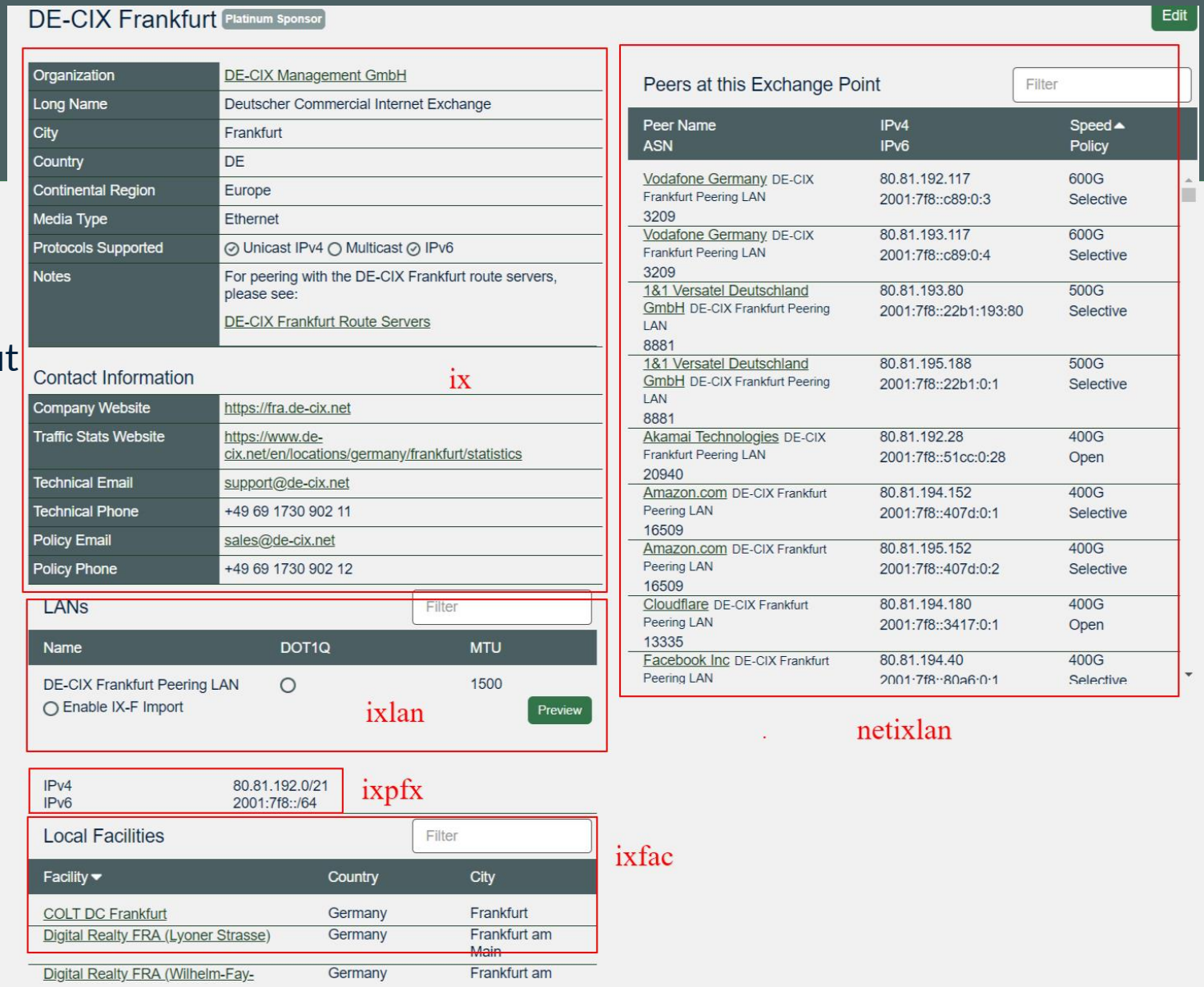
- Add pretty and depth for human friendly output
- https://peeringdb.com/api/net/13251
- https://peeringdb.com/api/poc?net_id=13251
- https://peeringdb.com/api/netixlan?net_id=31
- OR https://peeringdb.com/api/netixlan?asn=196610
- https://peeringdb.com/api/netfac?net_id=13251
- OR https://peeringdb.com/api/netfac?local_asn=196610



# GUI to API // ix

- https://peeringdb.com/ix/31

- Add pretty and depth for human friendly output
- https://peeringdb.com/api/ix/31
- https://peeringdb.com/api/ixlan?ix_id=31
- https://peeringdb.com/api/ixpfx?ixlan_id=31
- https://peeringdb.com/api/ixfac?ix_id=31
- https://peeringdb.com/api/netixlan?ix_id=31



# Basic records in detail // ix and org


```
"data": [
  {
    "id": 31,
    "org_id": 1187,
    "name": "DE-CIX Frankfurt",
    "name_long": "Deutscher Commercial Internet Exchange",
    "city": "Frankfurt",
    "country": "DE",
    "region_continent": "Europe",
    "media": "Ethernet",
    "notes": "For peering with the DE-CIX Frankfurt route servers, please see:\n\n[DE
    "proto_unicast": true,
    "proto_multicast": false,
    "proto_ipv6": true,
    "website": "https://fra.de-cix.net",
    "url_stats": "https://www.de-cix.net/en/locations/germany/frankfurt/statistics",
    "tech_email": "support@de-cix.net",
    "tech_phone": "+49 69 1730 902 11",
    "policy_email": "sales@de-cix.net",
    "policy_phone": "+49 69 1730 902 12",
    "net_count": 805,
    "created": "2010-07-29T00:00:00Z",
    "updated": "2018-06-19T11:53:46Z",
    "status": "ok"
  }
]
```



```
"data": [
  {
    "id": 1187,
    "name": "DE-CIX Management GmbH",
    "website": "https://de-cix.net",
    "notes": "",
    "address1": "Lindleystr. 12",
    "address2": "",
    "city": "Frankfurt am Main",
    "country": "DE",
    "state": "Hessia",
    "zipcode": "60314",
    "created": "2006-11-01T23:37:13Z",
    "updated": "2017-07-02T00:03:27Z",
    "status": "ok"
  }
]
```

# Basic records in detail // fac


```
"data": [
  {
    "id": 752,
    "org_id": 8540,
    "org_name": "euNetworks Group",
    "name": "euNetworks Colocation Hamburg",
    "website": "http://www.euNetworks.com",
    "clli": "",
    "rencode": "",
    "npanxx": "",
    "notes": "",
    "net_count": 5,
    "latitude": null,
    "longitude": null,
    "created": "2010-07-29T00:00:00Z",
    "updated": "2019-09-25T22:00:34Z",
    "status": "ok",
    "address1": "Wendenstra\u00dfe 408",
    "address2": "",
    "city": "Hamburg",
    "country": "DE",
    "state": "",
    "zipcode": "20537"
  }
]
```

# Basic records in detail // net and poc


```
"data": [
  {
    "id": 13251,
    "org_id": 1187,
    "name": "DE-CIX Academy Educational Network",
    "aka": "DE-CIX",
    "website": "http://www.de-cix.net/academy",
    "asn": 196610,
    "looking_glass": "",
    "route_server": "",
    "irr_as_set": "AS196610:AS-DECIX-ACADEMY",
    "info_type": "Educational/Research",
    "info_prefixes4": 1,
    "info_prefixes6": 5,
    "info_traffic": "0-20 Mbps",
    "info_ratio": "Balanced",
    "info_scope": "Regional",
    "info_unicast": true,
    "info_multicast": false,
    "info_ipv6": true,
    "notes": "* We only peer with the route servers\n* Any peering request you send will be used for educational purposes",
    "policy_url": "",
    "policy_general": "Open",
    "policy_locations": "Not Required",
    "policy_ratio": false,
    "policy_contracts": "Not Required",
    "created": "2017-04-20T19:44:59Z",
    "updated": "2019-02-15T12:19:05Z",
    "status": "ok"
  }
]
```

```
{
  "id": 25826,
  "net_id": 13251,
  "role": "Policy",
  "visible": "Users",
  "name": "Wolfgang Tremmel",
  "phone": "",
  "email": "academy@de-cix.net",
  "url": "",
  "created": "2018-07-24T09:26:21Z",
  "updated": "2018-07-24T09:26:21Z",
  "status": "ok"
},
```

# Basic records in detail // as_set


```
{
  "meta": {},
  "data": [
    {
      "196610": "AS196610:AS-DECIX-ACADEMY",
      "262150": "AR-EPEC2-LACNIC",
      "393223": "AS-CWICA",
      "32780": "AS-HSI",
      "196621": "AS196621:AS-CUSTOMERS",
      "327698": "AS-327698",
      "32787": "AS-PROLE",
      "327700": "AFRINIC",
      "32798": "RS-USCS-ALL",
      "5467": "AS-MIPT",
      "32806": "AS27822",
      "32808": "AS-UTBB",
      "42": "AS-PCH",
      "262189": "LACNIC",
      "46": "AS-RUTGERS",
      "262195": "AS-ITXAR1",
      "393269": "AS-DAILYMOTIONUS",
      "57": "AS-NLG-PARTICIPANTS",
      "327740": "ORG-TA38-AFRINIC",
      "62": "AS-C1",
      "393280": "AS393280 in Level3",
      "72": "AS-SLB",
      "327754": "AS-RMS-Powertronics",
      "327693": "AfriNIC::AS-ECHOSP/RS-ECHOSP",
      "81": "AS-NCREN@ARIN"
```


https://peeringdb.com/api/as-set


```
{
  "meta": {},
  "data": [
    {
      "42": "AS-PCH"
    }
  ]
}
```


https://peeringdb.com/api/as-set/42

# Derived records in detail // ixfac, ixlan and ixpfx


```
"data": [
  {
    "id": 41,
    "ix_id": 26,
    "fac_id": 63,
    "created": "2010-07-29T00:00:00Z",
    "updated": "2016-03-14T20:33:57Z",
    "status": "ok"
  }
]
```

```
{
  "id": 31,
  "ix_id": 31,
  "name": "DE-CIX Frankfurt Peering LAN",
  "descr": "",
  "mtu": 1500,
  "dot1q_support": false,
  "rs_asn": 0,
  "arp_sponge": null,
  "created": "2010-07-29T00:00:00Z",
  "updated": "2018-07-08T10:22:35Z",
  "status": "ok"
}
```

```
{
  "id": 312,
  "ixlan_id": 31,
  "protocol": "IPv6",
  "prefix": "2001:7f8::/64",
  "created": "2011-06-22T00:00:00Z",
  "updated": "2016-03-14T21:57:28Z",
  "status": "ok"
}
```

# Derived records // netfac and netixlan


```
{
  "id": 30451,
  "name": "Interxion Frankfurt (FRA1-13)",
  "city": "Frankfurt",
  "country": "DE",
  "net_id": 13251,
  "fac_id": 58,
  "local_asn": 196610,
  "created": "2018-07-24T09:25:24Z",
  "updated": "2018-07-24T09:25:24Z",
  "status": "ok"
}
```

```
{
  "id": 163,
  "net_id": 5
  "ix_id": 31
  "name": "DE-CIX Frankfurt: DE-CIX Frankfurt Peering LAN",
  "ixlan_id": 31,
  "notes": "",
  "speed": 20000,
  "asn": 3303,
  "ipaddr4": "80.81.193.183",
  "ipaddr6": "2001:7f8::ce7:0:2",
  "is_rs_peer": true,
  "created": "2010-07-29T00:00:00Z",
  "updated": "2019-01-18T11:19:59Z",
  "status": "ok"
},
```