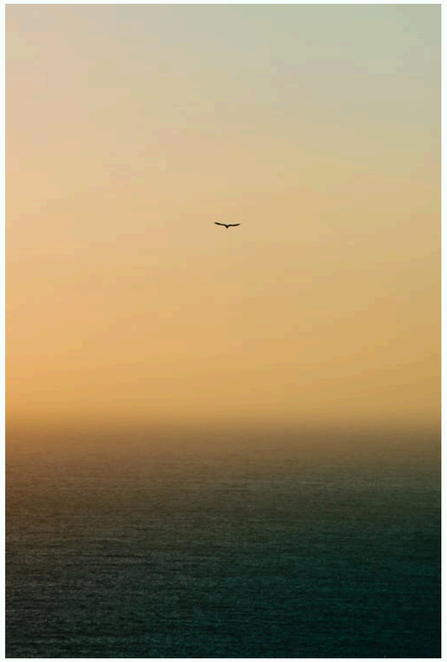
resource catalog

CAIDA's resource catalog

catalog.caida.org

problem: Internet infrastructure research

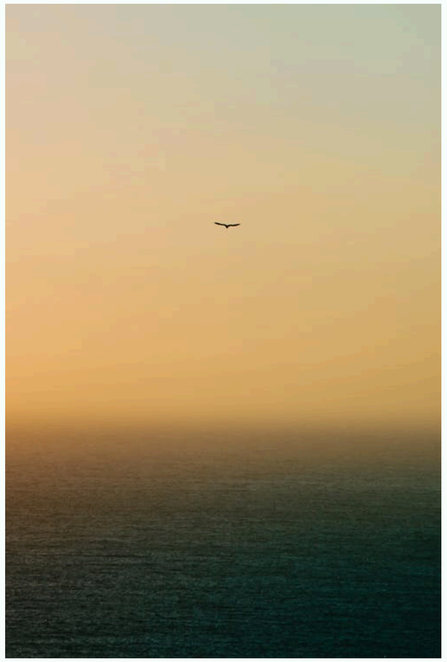


relevant data sets intractable for researchers to create and maintain

existing data sets have high bar to effectively use (and share)

consequence: stunted progress and low reproducibility in field

problem: CAIDA-specific version



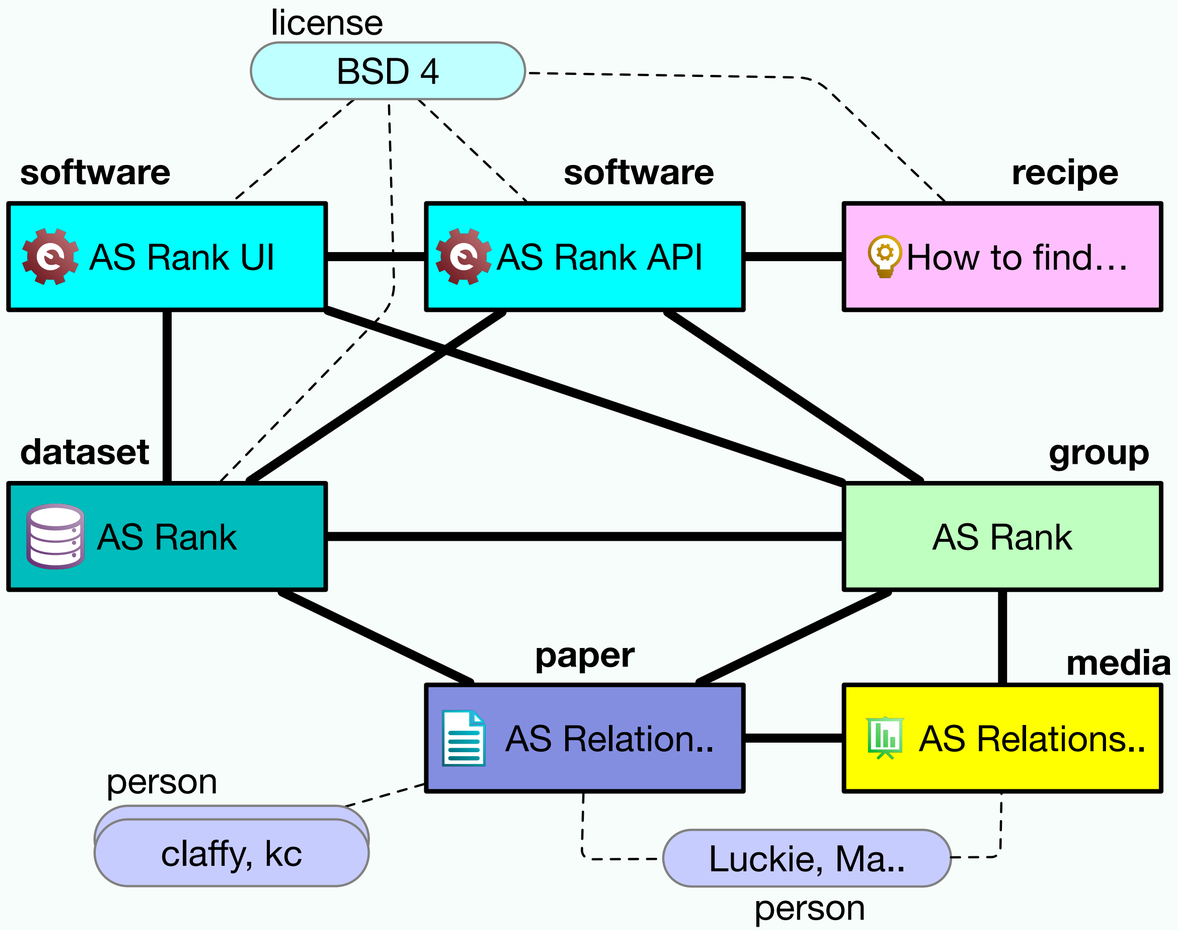
10-20 year longitudinal data sets about Internet infrastructure

researchers and students still often “reinvent wheel” when using it

consequence: stunted progress and low reproducibility in field

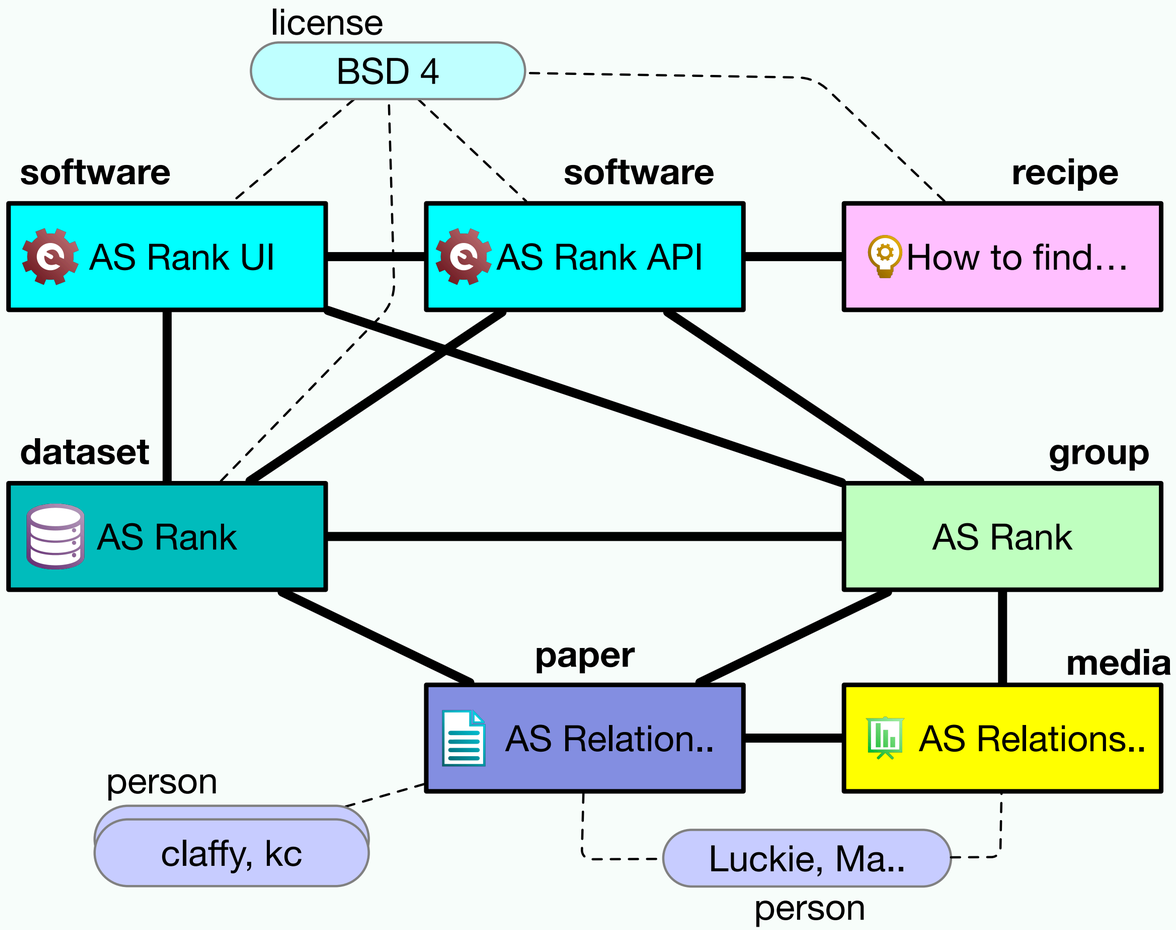
our solution

rich context
graph



what is a rich context graph?

a graph that shows the relationship between data and other things



what are our goals?

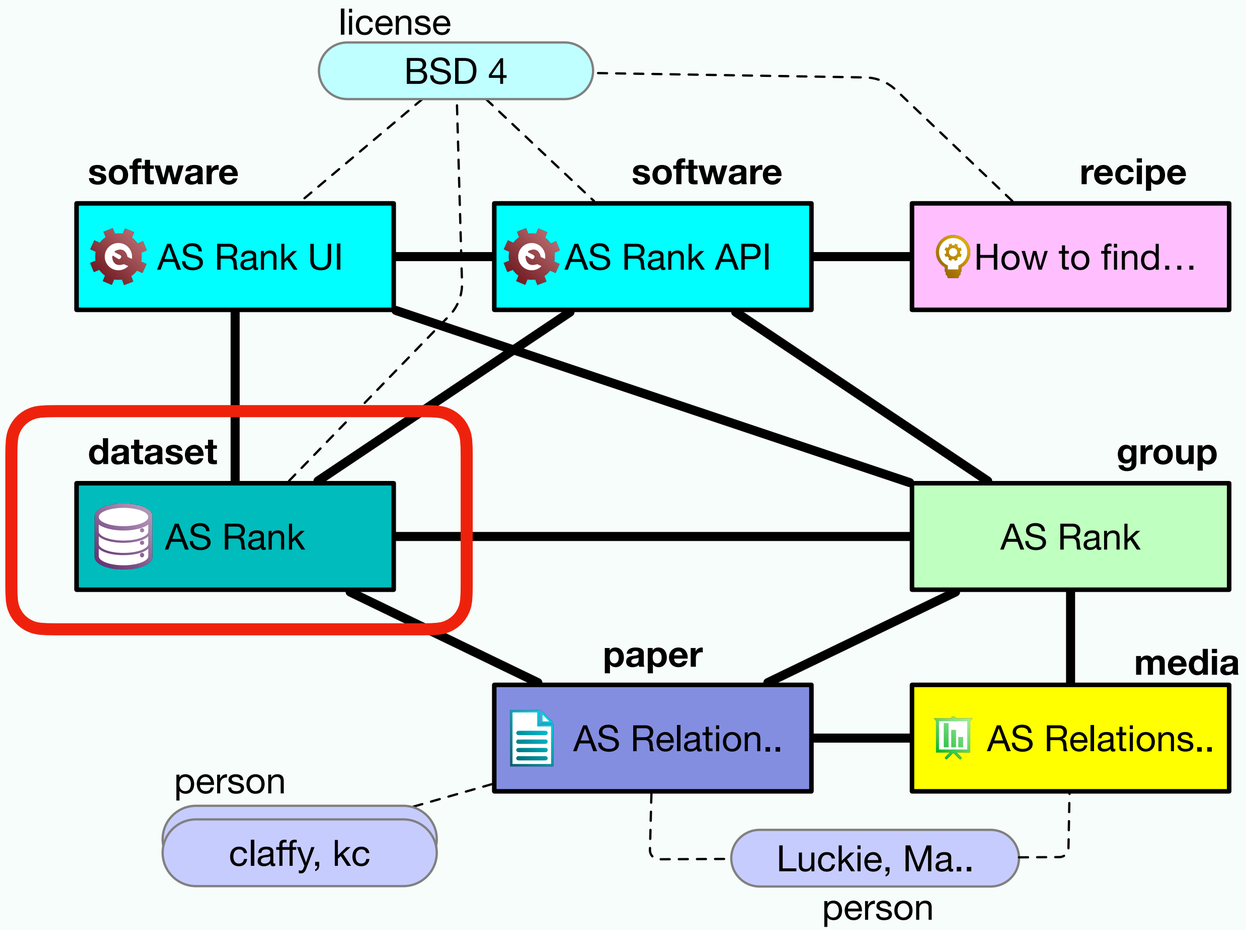
discovery	help to find data
understanding	help to understand data
inspiration	help to find new uses for data

how does a graph do that?

discovery	ability to search by word, tag, related objects
understanding	link data to sample code and explanations
inspiration	links to existing works, helps inspire new works

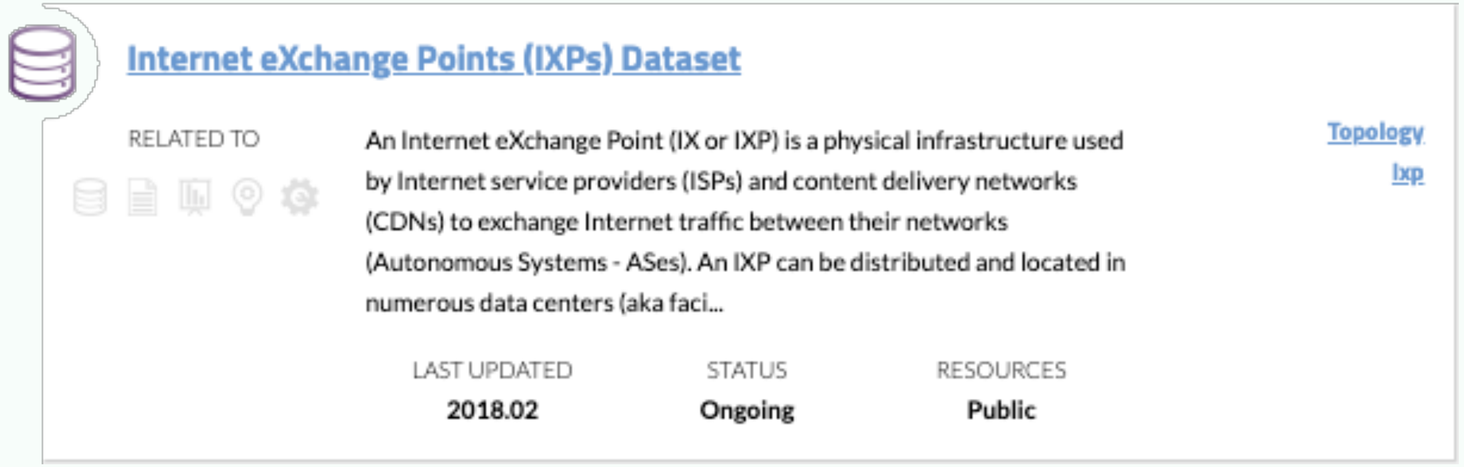
what is in our graph?

- dataset
- paper
- media
- software
- recipe



what counts as a dataset?

- dataset
- paper
- media
- software
- recipe



- collection of related information
- single file, database, or collection of files

what counts as a paper?

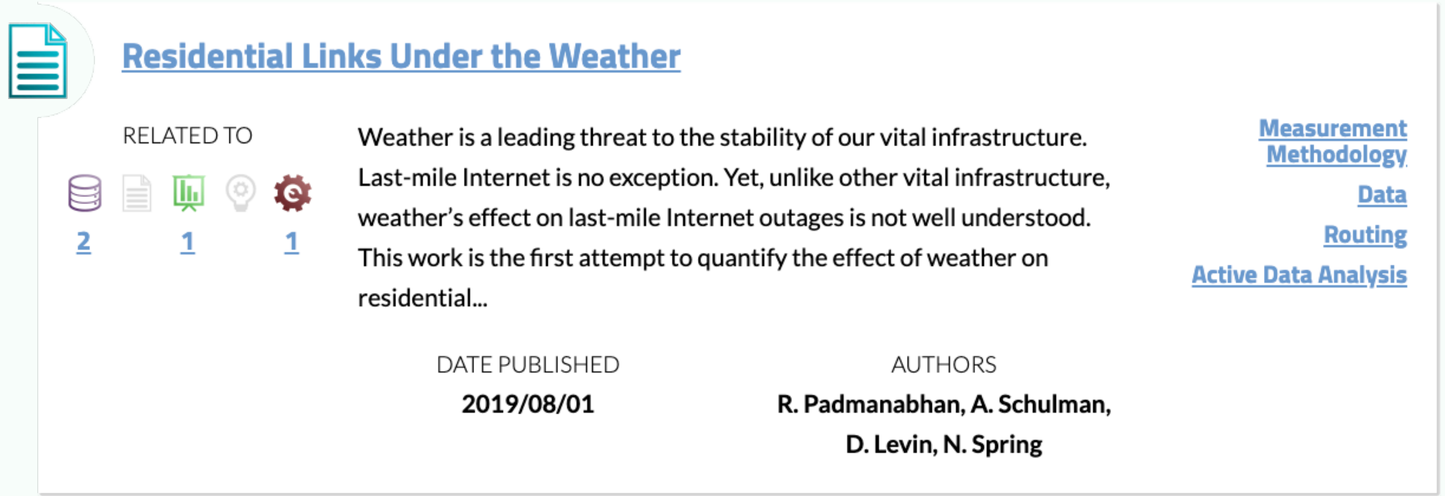
dataset

paper

media

software

recipe



- papers, research reports, articles
- it doesn't need to be published
  - just linked to an object

what counts as media?

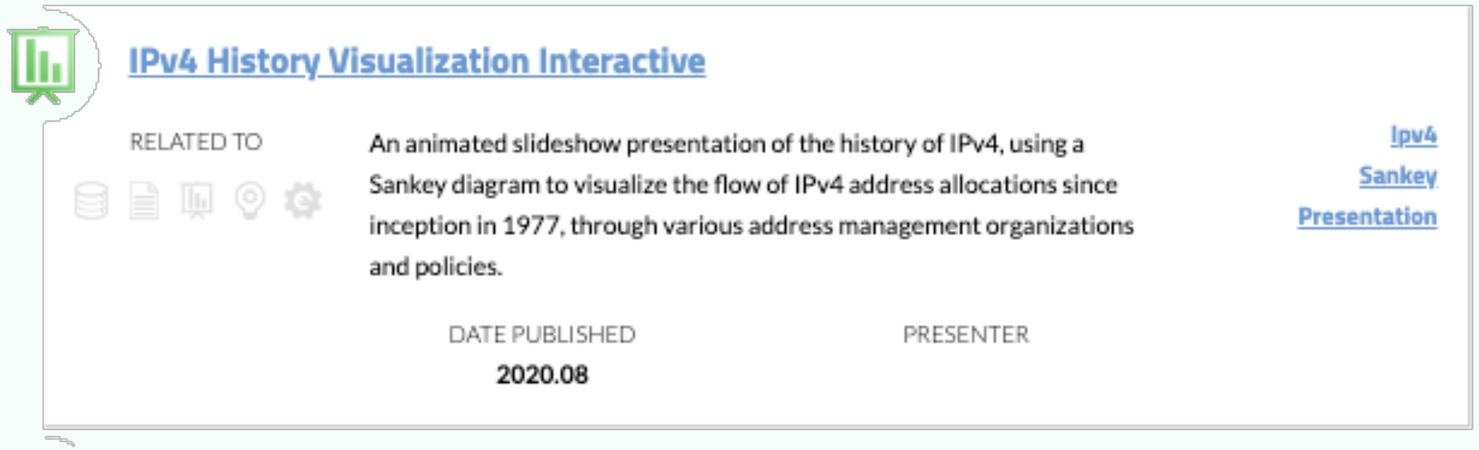
dataset

paper

media

software

recipe



- presentations, videos, visualizations

what counts as software?

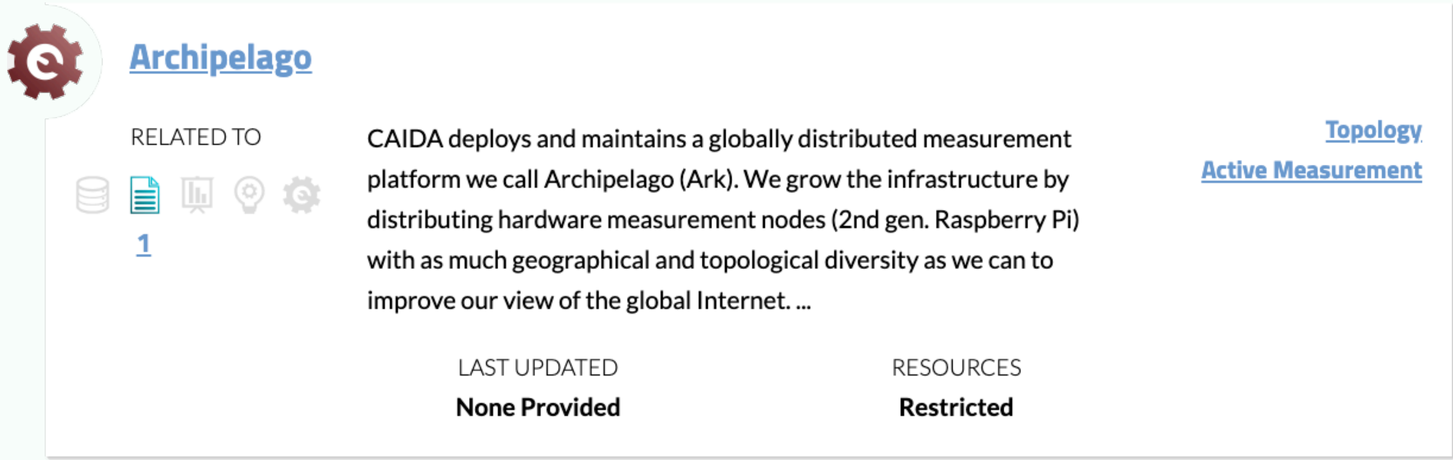
dataset

paper

media

software

recipe



- scripts, executable, web UIs, APIs
- creates, shares, or processes data

what counts a recipe?

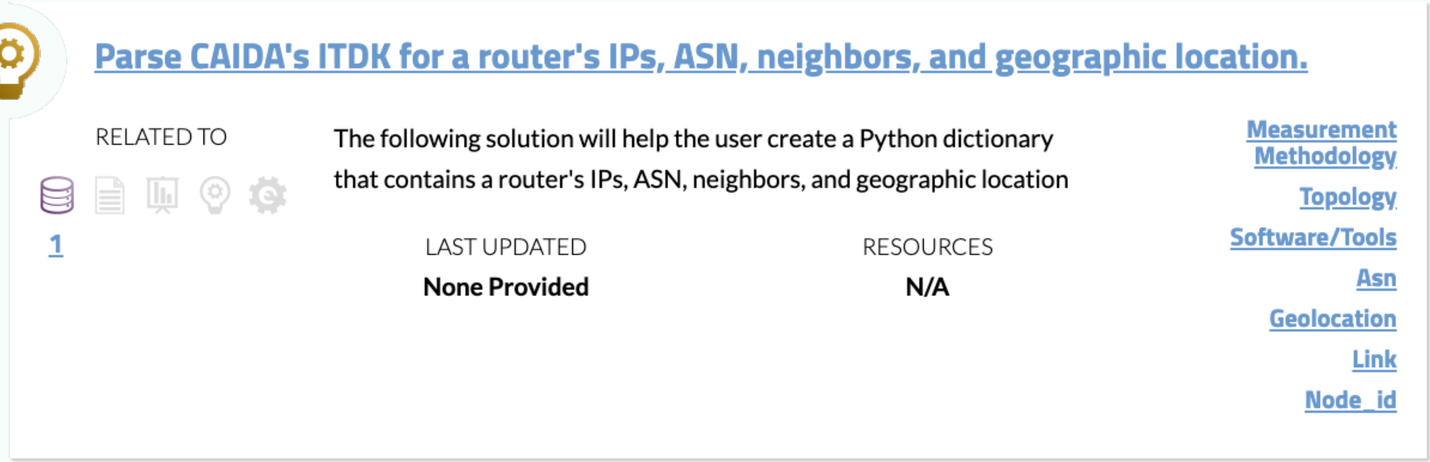
dataset

paper

media

software

recipe



- overview, methodologies, code snippets
- help to quickly understand how to use the data

recipe with code snippet

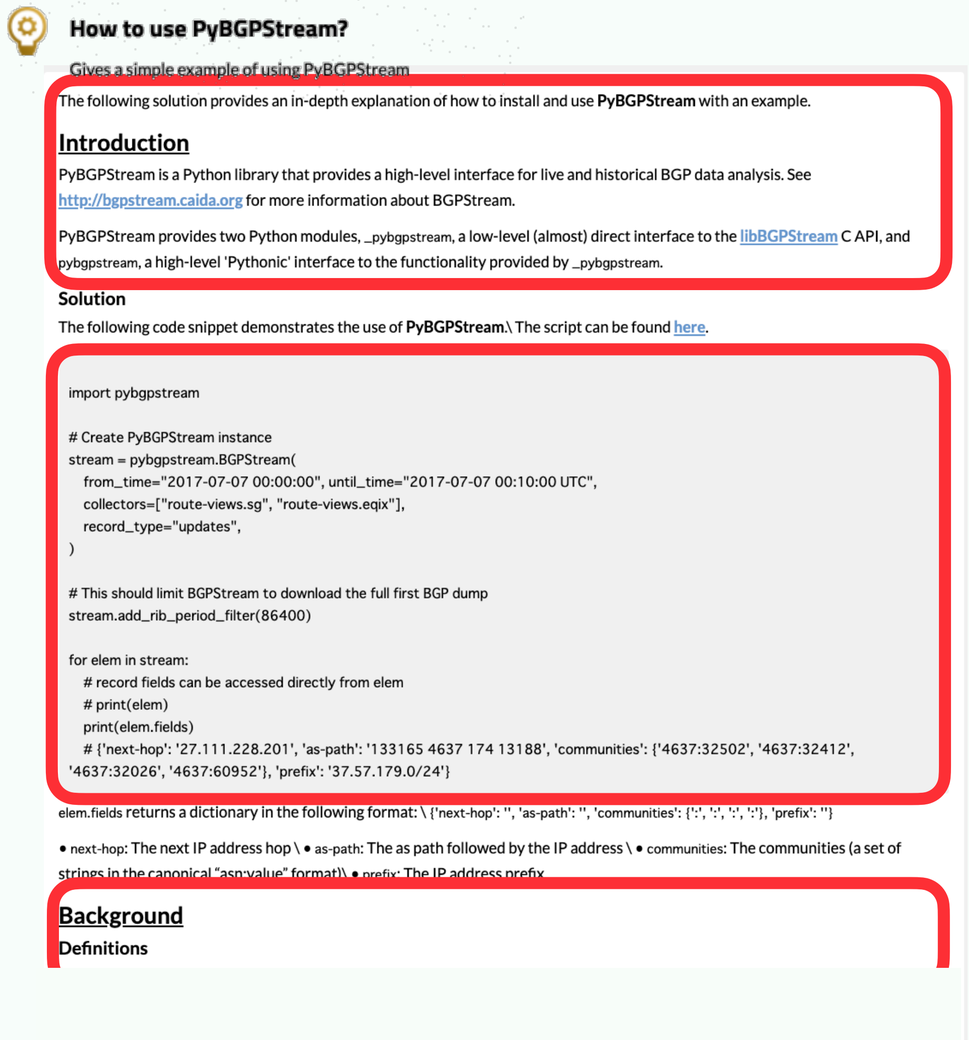
dataset

paper

media

software

recipe



How to use PyBGPStream?

Gives a simple example of using PyBGPStream

The following solution provides an in-depth explanation of how to install and use **PyBGPStream** with an example.

Introduction

PyBGPStream is a Python library that provides a high-level interface for live and historical BGP data analysis. See http://bgpstream.caida.org for more information about BGPStream.

PyBGPStream provides two Python modules, _pybgpstream, a low-level (almost) direct interface to the libBGPStream C API, and pybgpstream, a high-level 'Pythonic' interface to the functionality provided by _pybgpstream.

Solution

The following code snippet demonstrates the use of **PyBGPStream**.\ The script can be found here.

```
import pybgpstream

# Create PyBGPStream instance
stream = pybgpstream.BGPStream(
    from_time="2017-07-07 00:00:00", until_time="2017-07-07 00:10:00 UTC",
    collectors=["route-views.sg", "route-views.eqix"],
    record_type="updates",
)

# This should limit BGPStream to download the full first BGP dump
stream.add_rib_period_filter(86400)

for elem in stream:
    # record fields can be accessed directly from elem
    # print(elem)
    print(elem.fields)
    # {'next-hop': '27.111.228.201', 'as-path': '133165 4637 174 13188', 'communities': {'4637:32502', '4637:32412',
'4637:32026', '4637:60952'}, 'prefix': '37.57.179.0/24'}
```

elem.fields returns a dictionary in the following format: \ {'next-hop': '', 'as-path': '', 'communities': {':', ':', ':', ':'}, 'prefix': ''}

• next-hop: The next IP address hop \ • as-path: The as path followed by the IP address \ • communities: The communities (a set of strings in the canonical "asn:value" format)\ • prefix: The IP address prefix

Background

Definitions

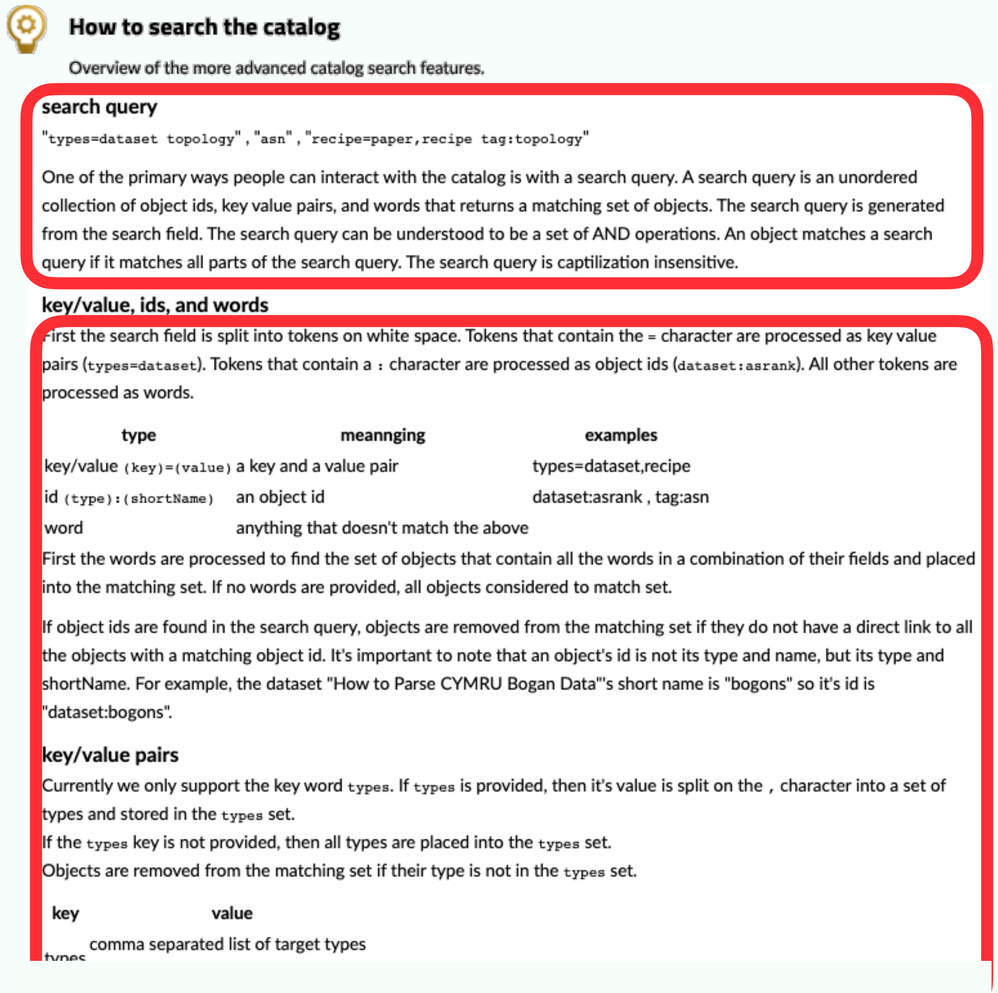
introduction

code snippet

background/caveats

overview recipe

- dataset
- paper
- media
- software
- recipe



How to search the catalog

Overview of the more advanced catalog search features.

search query

```
"types=dataset topology","asn","recipe=paper,recipe tag:topology"
```

One of the primary ways people can interact with the catalog is with a search query. A search query is an unordered collection of object ids, key value pairs, and words that returns a matching set of objects. The search query is generated from the search field. The search query can be understood to be a set of AND operations. An object matches a search query if it matches all parts of the search query. The search query is captilization insensitive.

key/value, ids, and words

First the search field is split into tokens on white space. Tokens that contain the = character are processed as key value pairs (`types=dataset`). Tokens that contain a `:` character are processed as object ids (`dataset:asrank`). All other tokens are processed as words.

type	meannging	examples
key/value `(key)=(value)`	a key and a value pair	types=dataset,recipe
id `(type):(shortName)`	an object id	dataset:asrank , tag:asn
word	anything that doesn't match the above	

First the words are processed to find the set of objects that contain all the words in a combination of their fields and placed into the matching set. If no words are provided, all objects considered to match set.

If object ids are found in the search query, objects are removed from the matching set if they do not have a direct link to all the objects with a matching object id. It's important to note that an object's id is not its type and name, but its type and shortName. For example, the dataset "How to Parse CYMRU Bogan Data"'s short name is "bogons" so it's id is "dataset:bogons".

key/value pairs

Currently we only support the key word `types`. If `types` is provided, then it's value is split on the `,` character into a set of types and stored in the `types` set.
If the `types` key is not provided, then all types are placed into the `types` set.
Objects are removed from the matching set if their type is not in the `types` set.

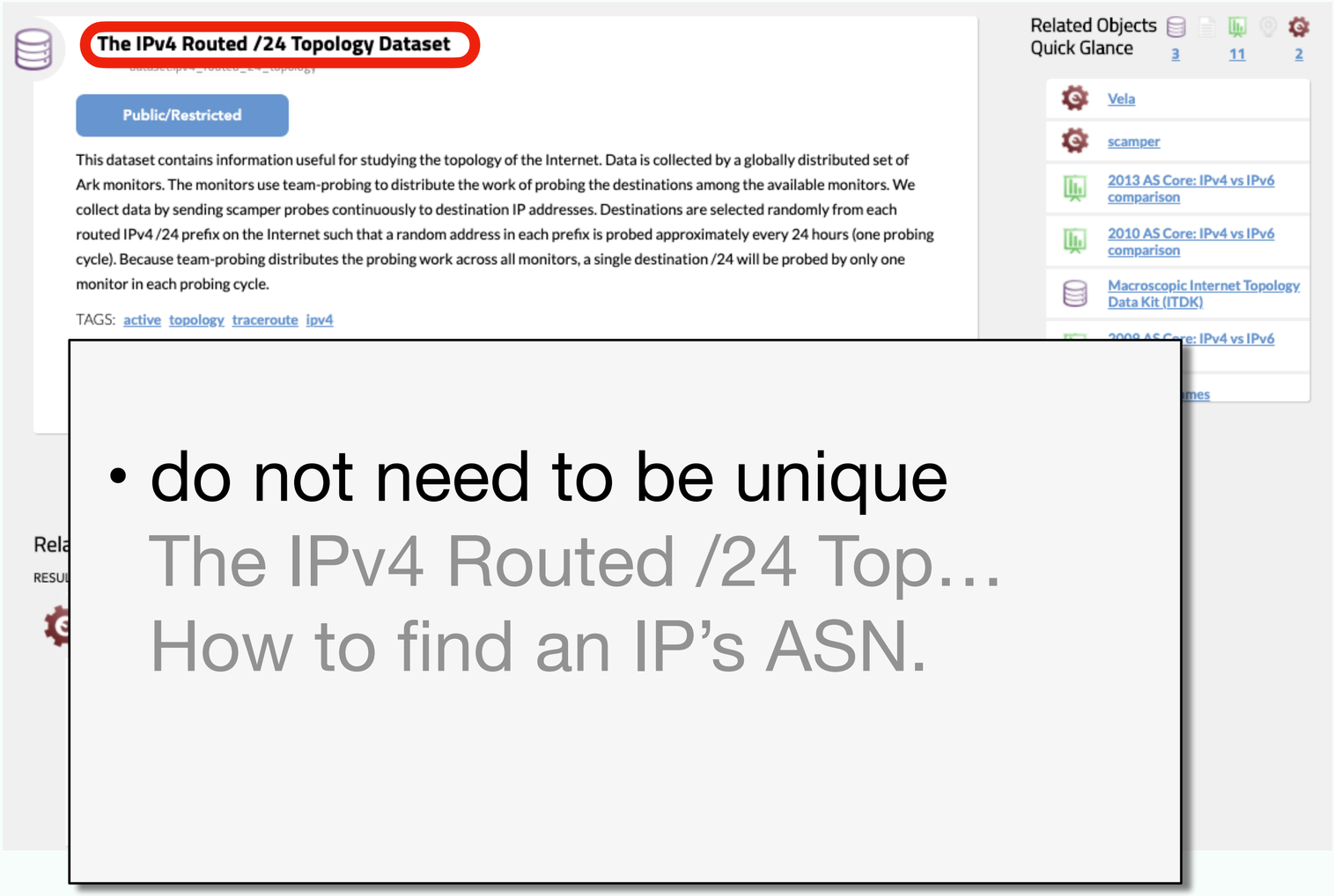
key	value
types	comma separated list of target types

introduction

breakdown

which fields are common to all objects?

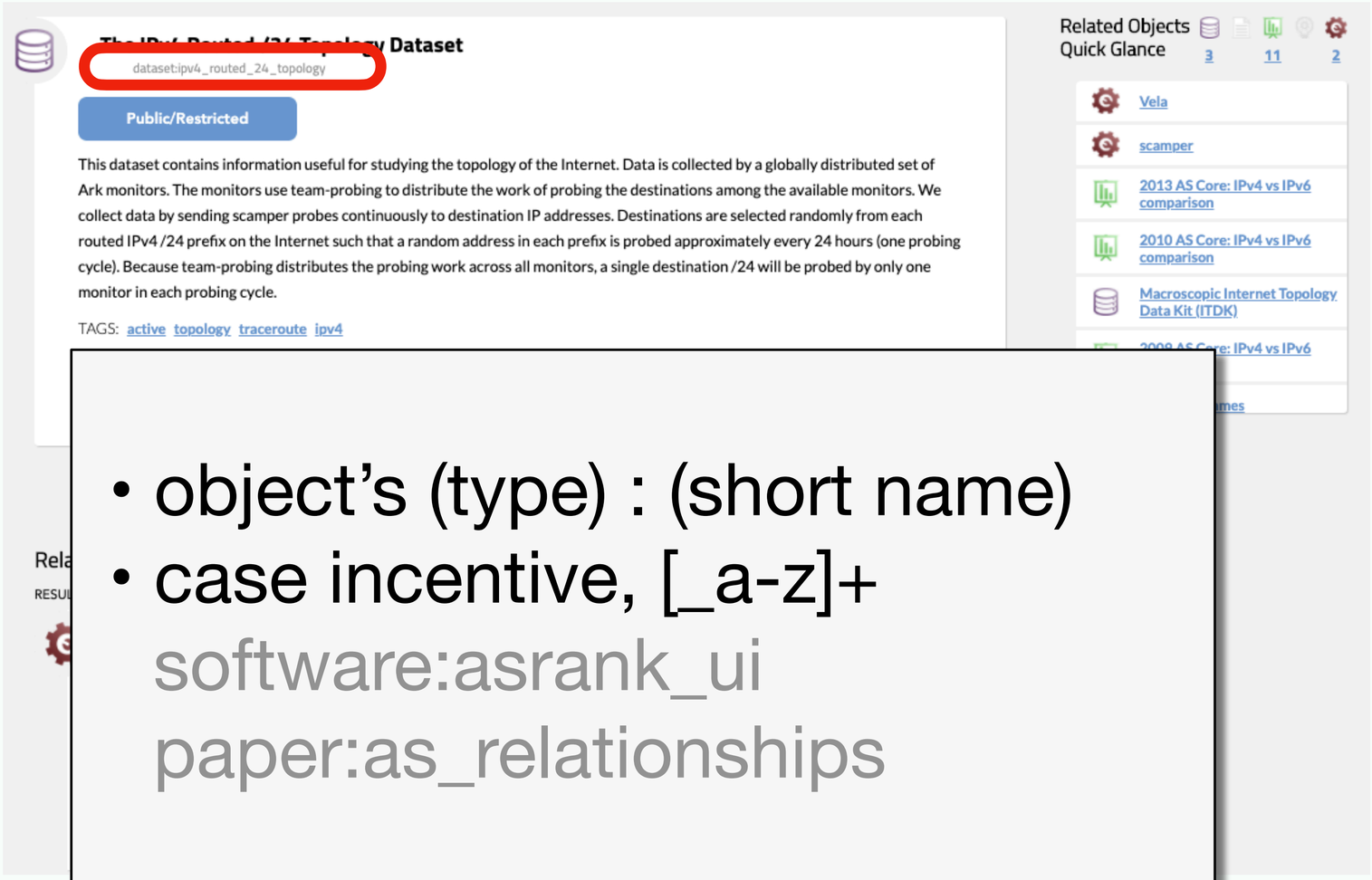
- name
- id
- resources
- description
- tags
- links



- do not need to be unique
  The IPv4 Routed /24 Top…
  How to find an IP's ASN.

which fields are common to all objects?

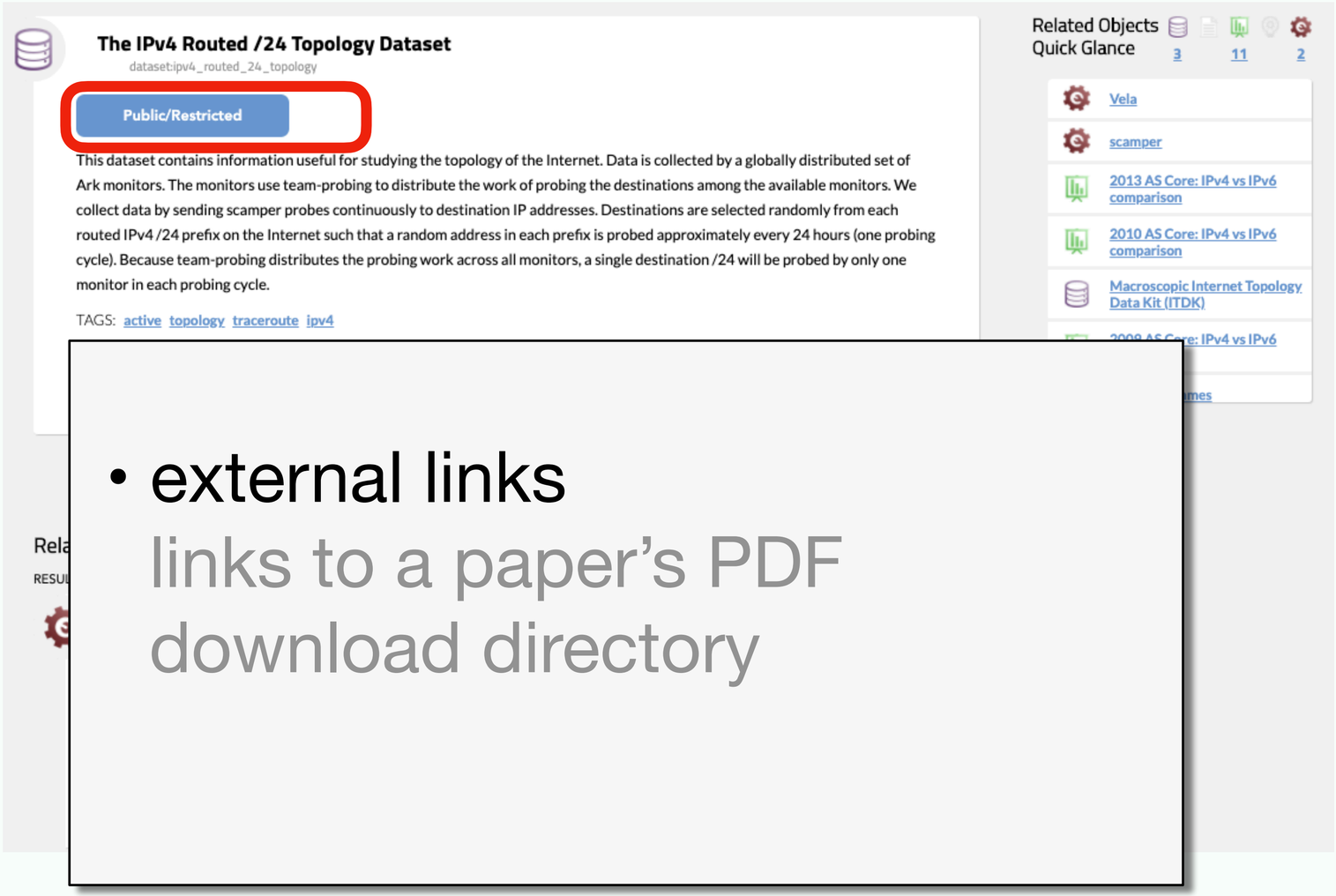
- name
- id
- resources
- description
- tags
- links



- object's (type) : (short name)
- case incentive, [_a-z]+
  software:asrank_ui
  paper:as_relationships

which fields are common to all objects?

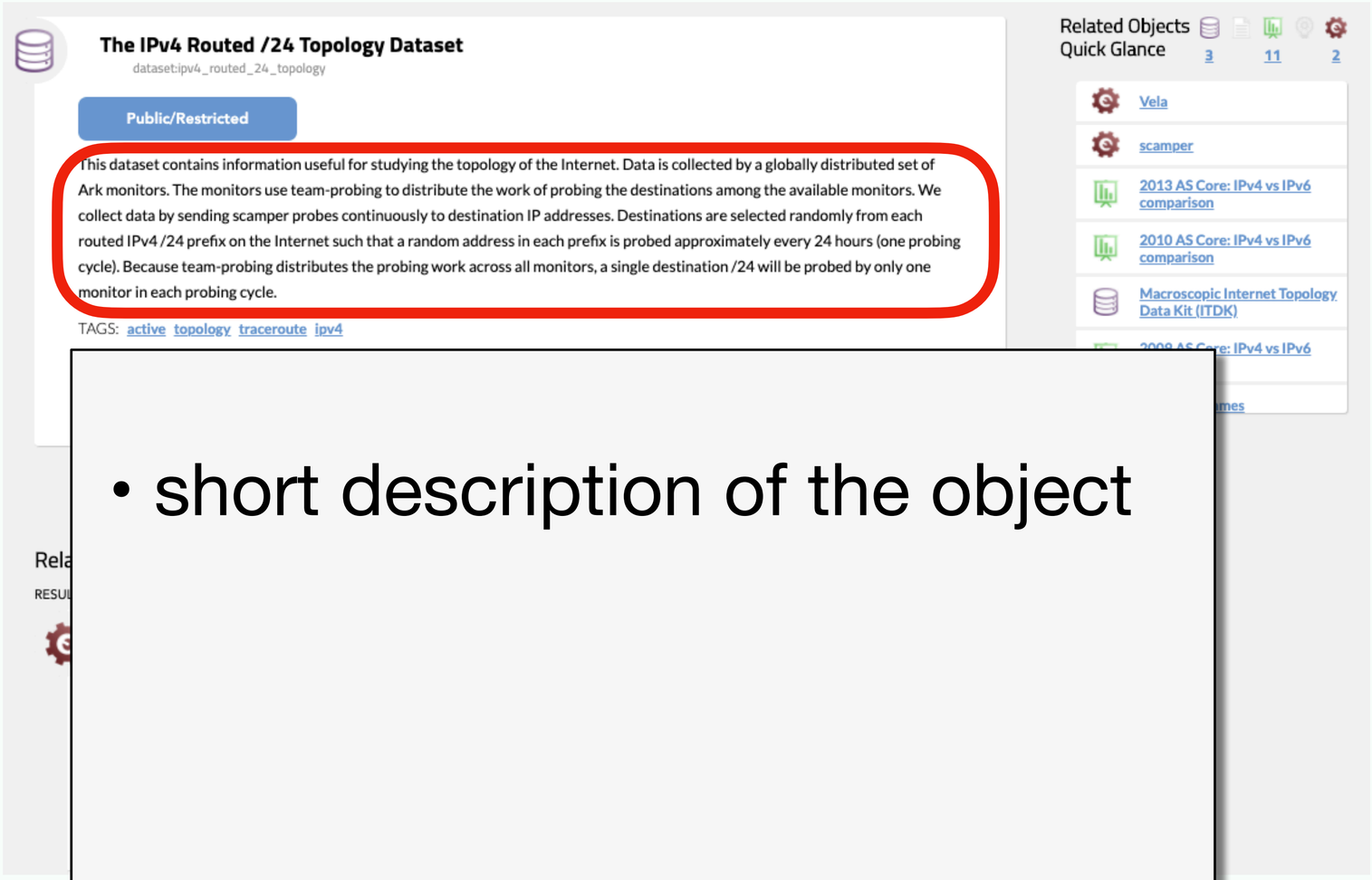
- name
- id
- resources
- description
- tags
- links



- external links
  links to a paper's PDF
  download directory

which fields are common to all objects?

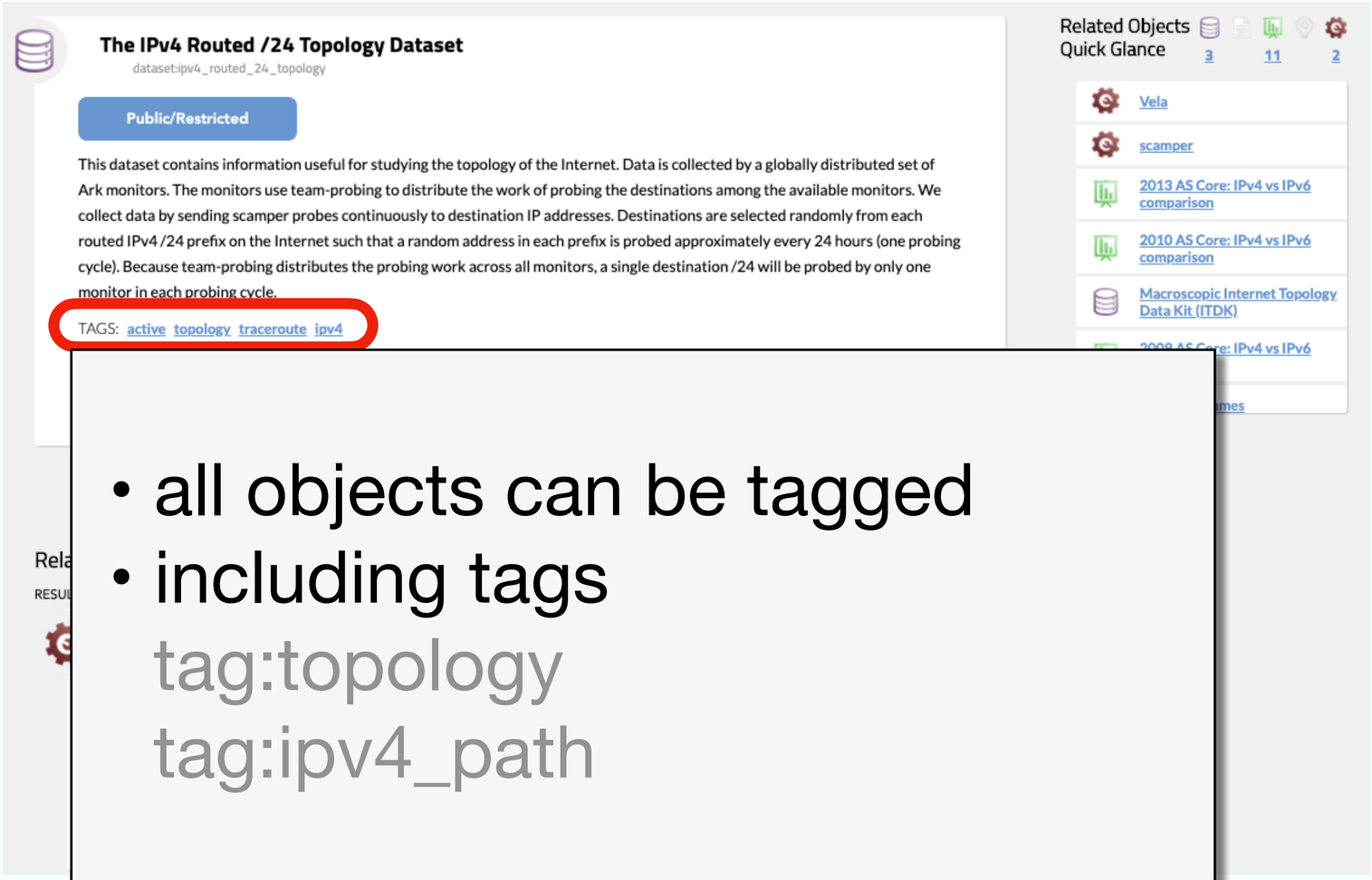
- name
- id
- resources
- description
- tags
- links



- short description of the object

which fields are common to all objects?

- name
- id
- resources
- description
- tags
- links

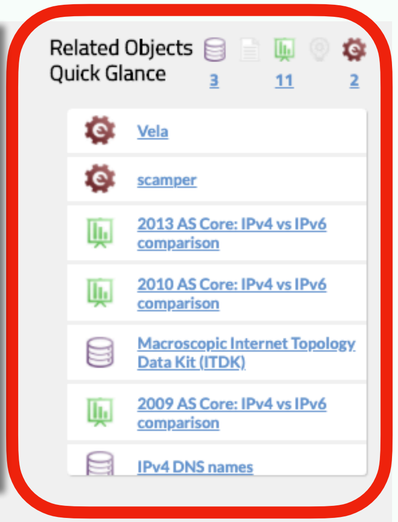


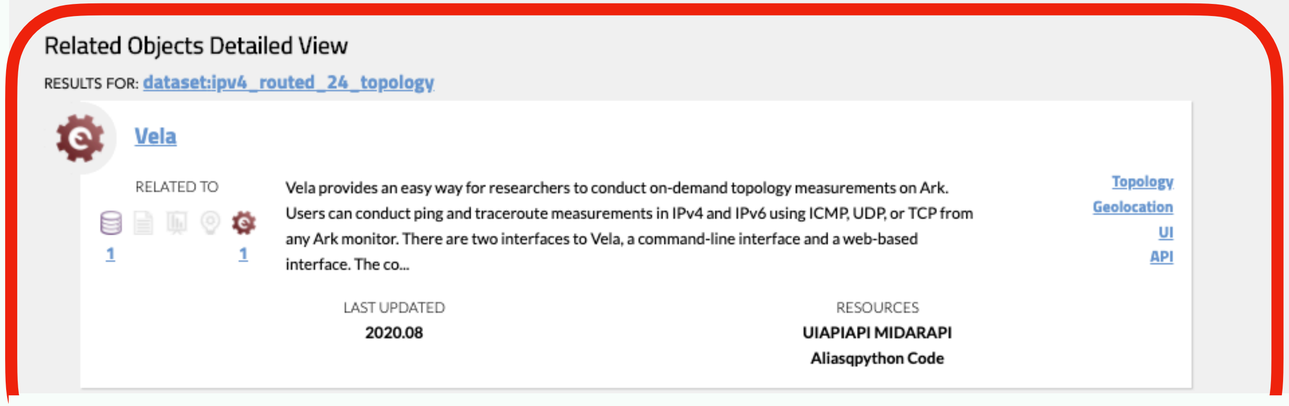
- all objects can be tagged
- including tags
  tag:topology
  tag:ipv4_path

which fields are common to all objects?

- name
- id
- resources
- description
- tags
- links

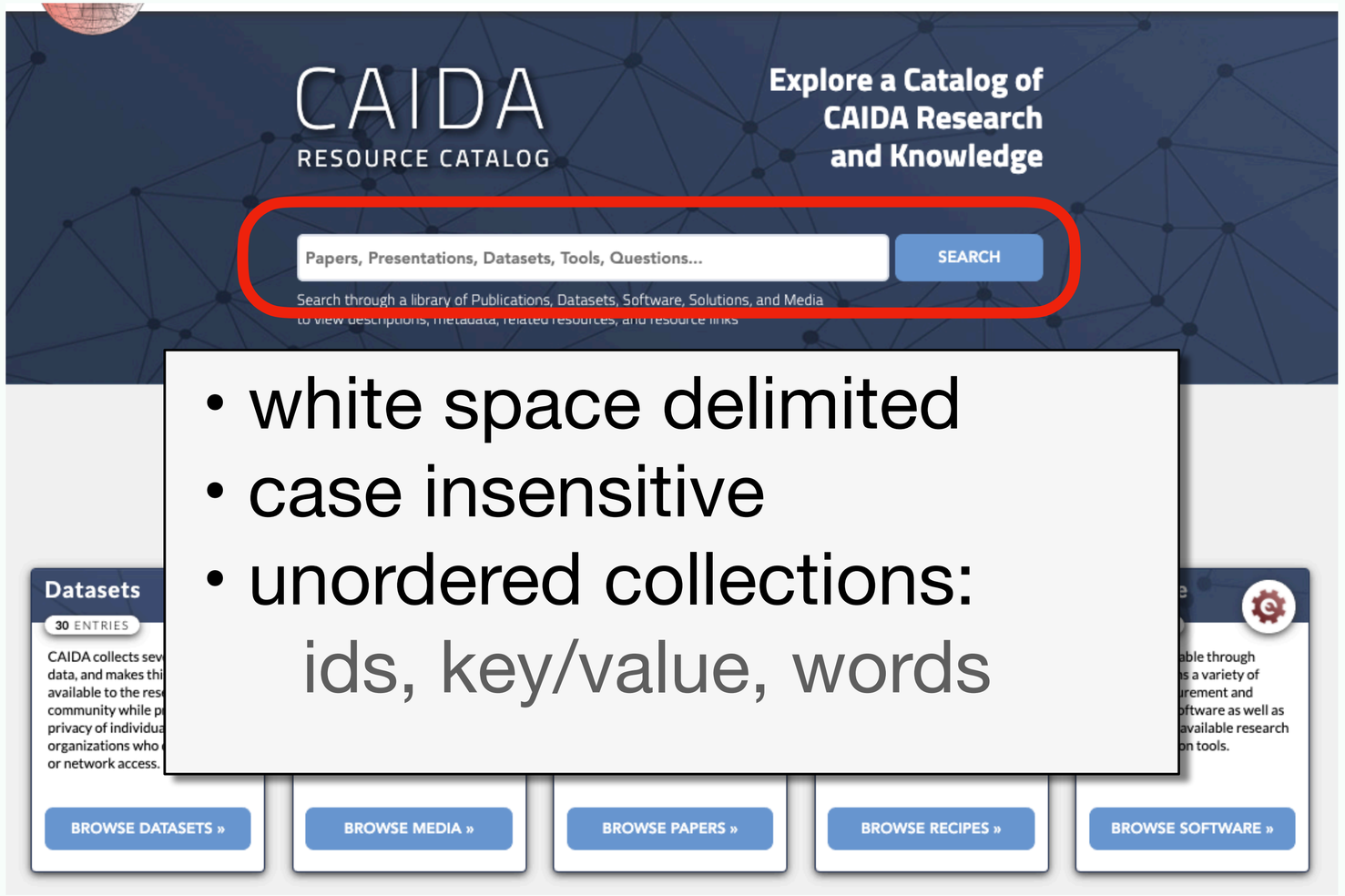
- summary of related objects
- more detailed related objects





where should you start?

search field



- white space delimited
- case insensitive
- unordered collections:
  ids, key/value, words

what are ids, key/values, words?

ids

(type):(value) extracts all objects linked to id

- filters on topology of the context graph

  tag:bgp recipe:how_to_parse_bgp

key/value

types=(types) filters by object type

- comma-separated list of object types

  types=dataset types=recipe,dataset

words

all remaining tokens

- words in object fields

  asn bgp huffaker ucsd ip security encrypted

how you can help

use

we built it, please use it

catalog.caida.org

share

with your friends and colleagues



contribute

your recipes, datasets, and papers

github.com/CAIDA/catalog-data