RIPE NCC
RIPE NETWORK COORDINATION CENTRE

RIPE RIS

Routing Information System for Shared Routing Intelligence

Ties de Kock | RIPE NCC | 4 June 2026

The RIPE Routing Information Service

What is RIS?

- RIS is a route data collection platform
- Collecting BGP data since 1999
- Collaborative with community

The RIPE Routing Information Service

What is RIS?

- RIS is a route data collection platform
- Collecting BGP data since 1999
  - We *permanently* store the data (BGP tables + BGP updates)
  - Long-term archive is *publicly and freely available*
- Collaborative with community
- Used by many tools and researchers
  - RIPE NCC: RIPEstat, riswhois (IP to origin AS + prefix), ris-live, …
  - Others: Academic research, open source tools, commercial tools, …

What is RIPE RIS?

System overview

- A world-wide network of route collectors
- *Many* BGP peers that have BGP sessions with those collectors
- Standard components
  - Linux servers
  - ExaBGP
  - Kafka message broker
- + Custom data processing and plumbing

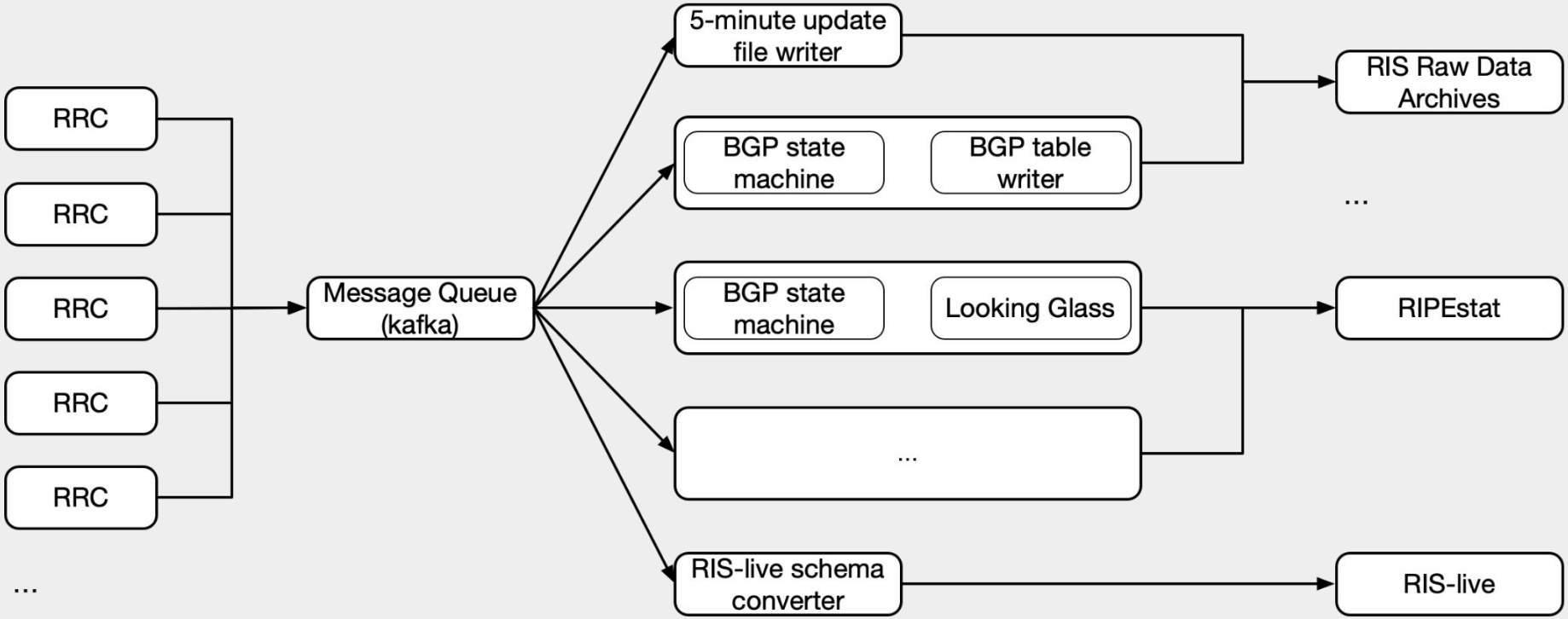
What is RIPE RIS?

System overview

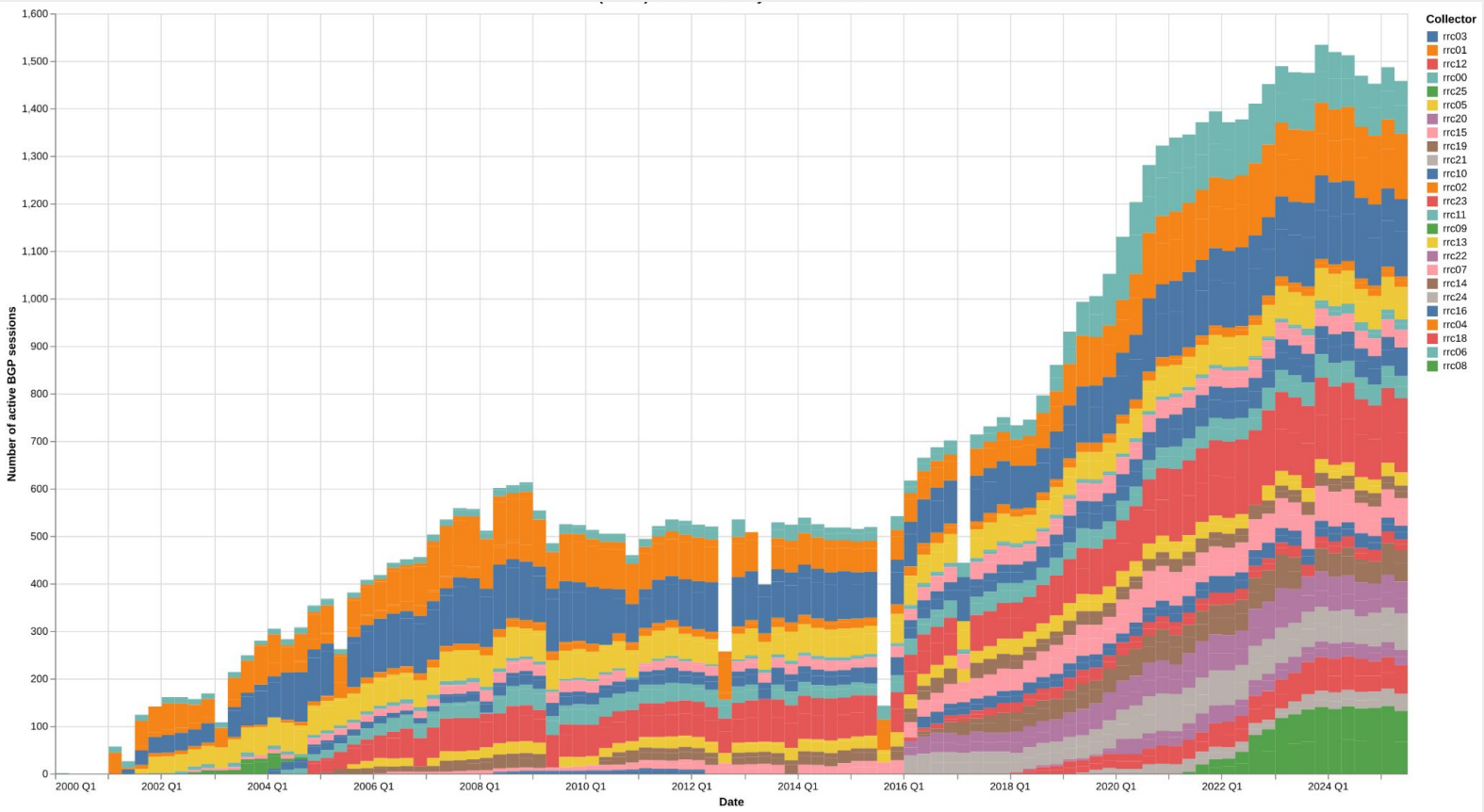
- A world-wide network of route collectors
- *Many* BGP peers that have BGP sessions with those collectors
- Standard components
  - Linux servers
  - ExaBGP
  - Kafka message broker
- + Custom data processing and plumbing

- *Many* BGP peers: A lot of routes and updates.
- 21 May 2026:
  - ~366M IPv4 routes, ~84M IPv6 routes
  - 17.5K bgp announcements/withdraws per second on average,
  - 46k A/W per second peak one-minute average (across five days)

RIS Architecture



RIS Over Time: Number of Active BGP Sessions



RIS Data Over Time

Remember: We store all data permanently

- Data size continuously increases
- Selecting relevant data is hard
  - For researchers: data volume is too big
  - For us as well: peering is selecting data for long-term observation

RIS Data Over Time

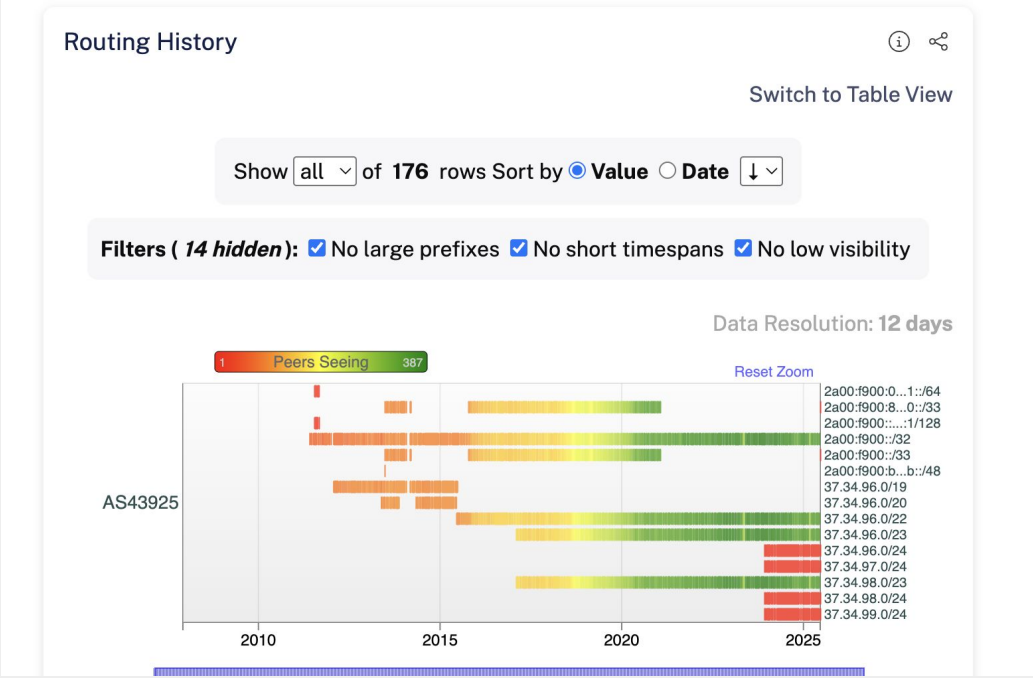
Remember: We store all data permanently

- Data size continuously increases
- Selecting relevant data is hard
  - For researchers: data volume is too big
  - For us as well: peering is selecting data for long-term observation
- We now have a selective peering policy
  - We will update it soon
  - Part data driven
  - Important part: validation
- We *really* want to learn about the most relevant interconnection in a region. Please talk to us!

How is RIS Data Used?

By us:

- In our tools
- In our internet analysis



How is RIS Data Used?

By us:

- In our tools
- In our internet analysis

```
❯ whois -h riswhois.ripe.net 37.34.96.0/24
% This is RIPE NCC's Routing Information Service
% Whois gateway to collected BGP Routing Tables, version2.0
% IPv4 or IPv6 address to origin prefix match
%
% For more information visit http://www.ripe.net/ris/riswhois.html
%
% Connected to backend ris-whois16.ripe.net

route:        37.34.96.0/22
origin:       AS43925
descr:        MOLDCELL_AS MOLDCELL S.A., MD
lastupd-frst: 2025-06-03 10:25Z  5.57.80.210@rrc01
lastupd-last: 2025-06-18 23:58Z  103.87.125.0@rrc25
seen-at:      rrc00,rrc01,rrc03,rrc04,rrc05,rrc06,rrc07,rrc10,rrc11,rrc12,rrc13,rrc14,rrc15,rrc16,rrc18,rrc19
,rrc20,rrc21,rrc22,rrc23,rrc24,rrc25,rrc26
num-rispeers: 373
source:       RISWHOIS

route:        37.34.96.0/23
origin:       AS43925
descr:        MOLDCELL_AS MOLDCELL S.A., MD
lastupd-frst: 2025-06-03 10:25Z  5.57.80.210@rrc01
lastupd-last: 2025-06-18 23:58Z  103.87.125.0@rrc25
seen-at:      rrc00,rrc01,rrc03,rrc04,rrc05,rrc06,rrc07,rrc10,rrc11,rrc12,rrc13,rrc14,rrc15,rrc16,rrc18,rrc19
,rrc20,rrc21,rrc22,rrc23,rrc24,rrc25,rrc26
num-rispeers: 373
source:       RISWHOIS

route:        37.34.96.0/24
origin:       AS43925
descr:        MOLDCELL_AS MOLDCELL S.A., MD
```

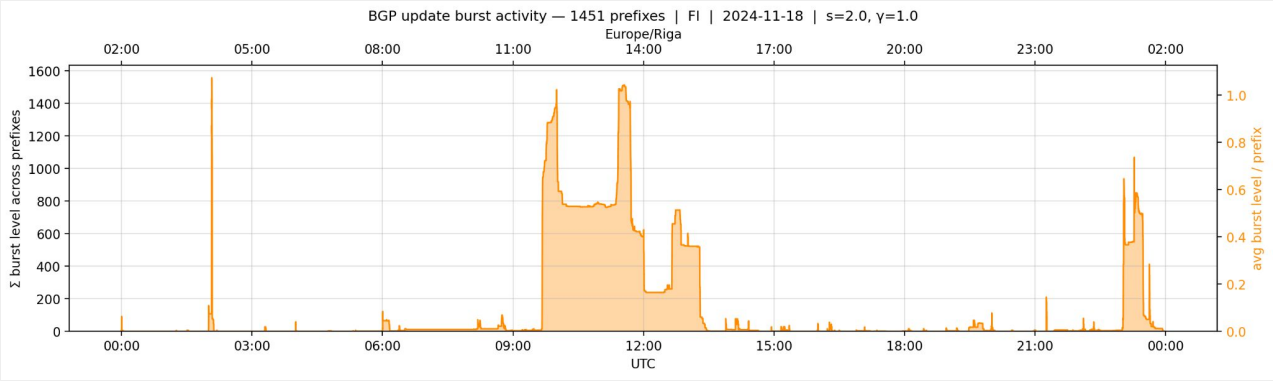
How is RIS Data Used?

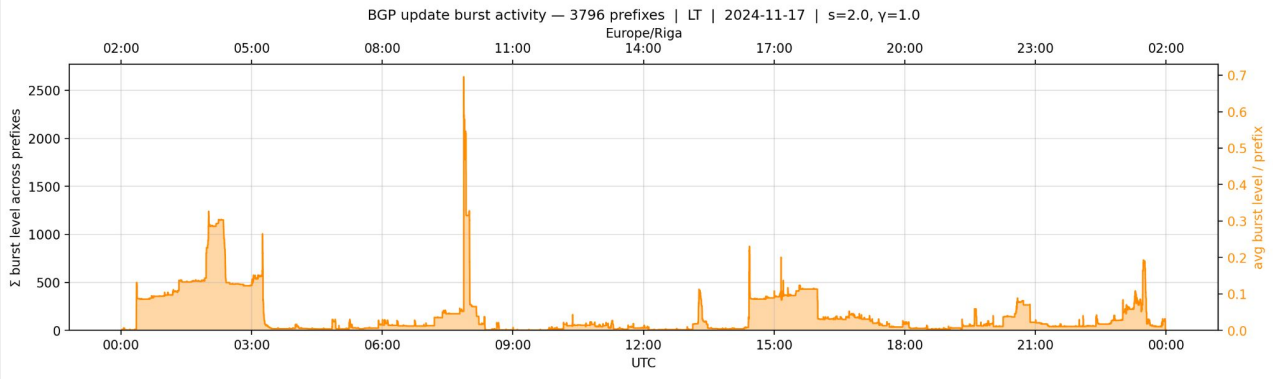
By us: routing analysis

- In our tools
- In our internet analysis

This example:

- Cable damage caused a spike in BGP updates for affected prefix
- Hard to find the needle in the haystack!

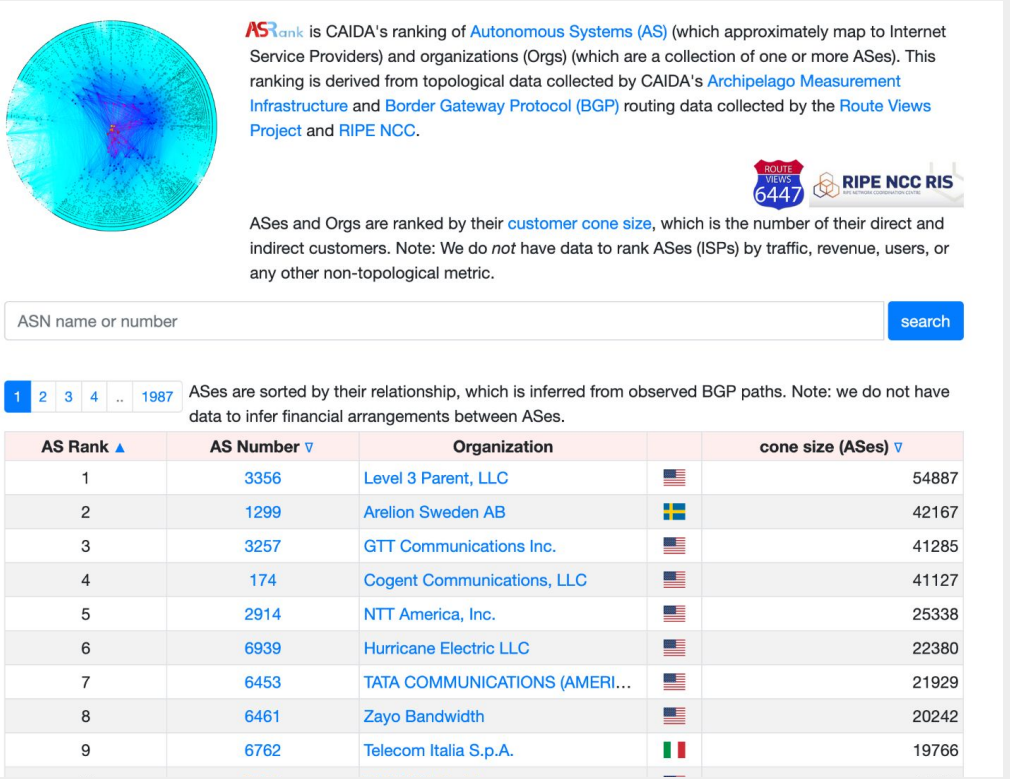




How is RIS Data Used?

By others:

- Often used in research
  - Example: CAIDA ASRank
- In open source software
- In other products



ASRank is CAIDA's ranking of Autonomous Systems (AS) (which approximately map to Internet Service Providers) and organizations (Orgs) (which are a collection of one or more ASes). This ranking is derived from topological data collected by CAIDA's Archipelago Measurement Infrastructure and Border Gateway Protocol (BGP) routing data collected by the Route Views Project and RIPE NCC.

ASes and Orgs are ranked by their customer cone size, which is the number of their direct and indirect customers. Note: We do *not* have data to rank ASes (ISPs) by traffic, revenue, users, or any other non-topological metric.

ASN name or number | search

1 2 3 4 .. 1987

ASes are sorted by their relationship, which is inferred from observed BGP paths. Note: we do not have data to infer financial arrangements between ASes.

AS Rank ▲	AS Number ▽	Organization		cone size (ASes) ▽
1	3356	Level 3 Parent, LLC		54887
2	1299	Arelion Sweden AB		42167
3	3257	GTT Communications Inc.		41285
4	174	Cogent Communications, LLC		41127
5	2914	NTT America, Inc.		25338
6	6939	Hurricane Electric LLC		22380
7	6453	TATA COMMUNICATIONS (AMERI...		21929
8	6461	Zayo Bandwidth		20242
9	6762	Telecom Italia S.p.A.		19766

How is RIS Data Used?

By others: Routing Analysis

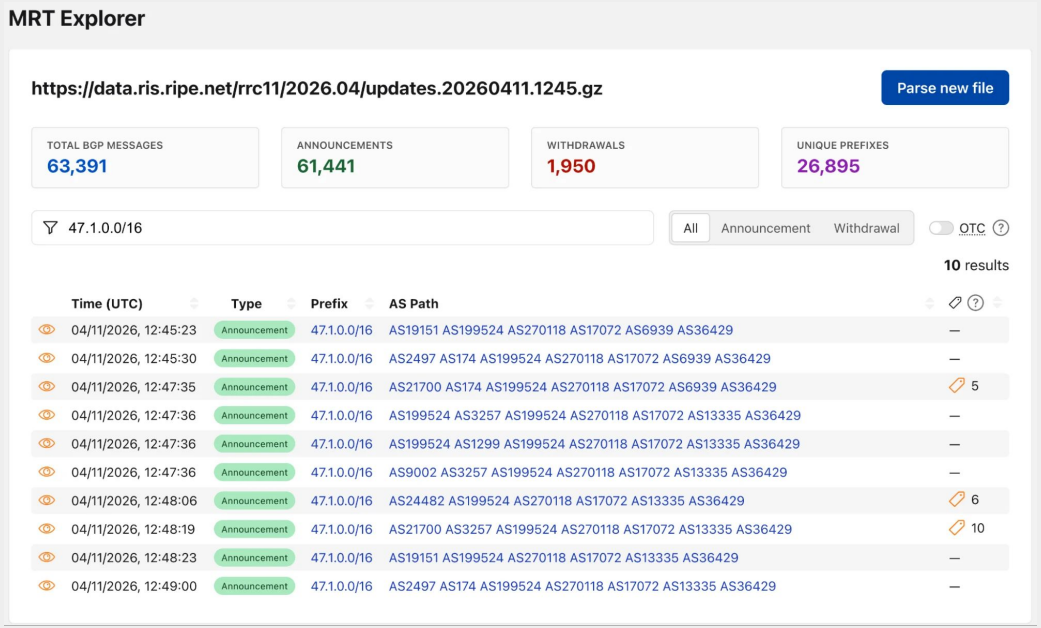
- Cloudflare blog: Enforcing the First AS in BGP AS_PATHs
- Blog shows multiple ways to query RIS data
- `monocle` - search for event
- MRT explorer - view raw data

```
→  ~ monocle search --start-ts 2026-04-13T00:20:00Z --end-ts 2026-04-13T00:23:59Z --prefix 90.98.0.0/15 --collector rrc26 --json
{
  "aggr_asn": null,
  "aggr_ip": null,
  "as_path": "48237 1299 199524 270118 17072 41128",
  "atomic": false,
  "collector": "rrc26",
  "communities": null,
  "local_pref": 0,
  "med": 0,
  "next_hop": "185.1.8.3",
  "origin": "IGP",
  "peer_asn": 48237,
  "peer_ip": "185.1.8.3",
  "prefix": "90.98.0.0/15",
  "timestamp": 1776039612.0,
  "type": "ANNOUNCE"
}
```

How is RIS Data Used?

By others: Routing Analysis

- Cloudflare blog: [Enforcing the First AS in BGP AS_PATHs]
- Blog shows multiple ways to query RIS data
- `monocle` - search for event
- MRT explorer - view raw data



How to use RIS Data Yourself?

Example: RIPEstat Looking Glass API

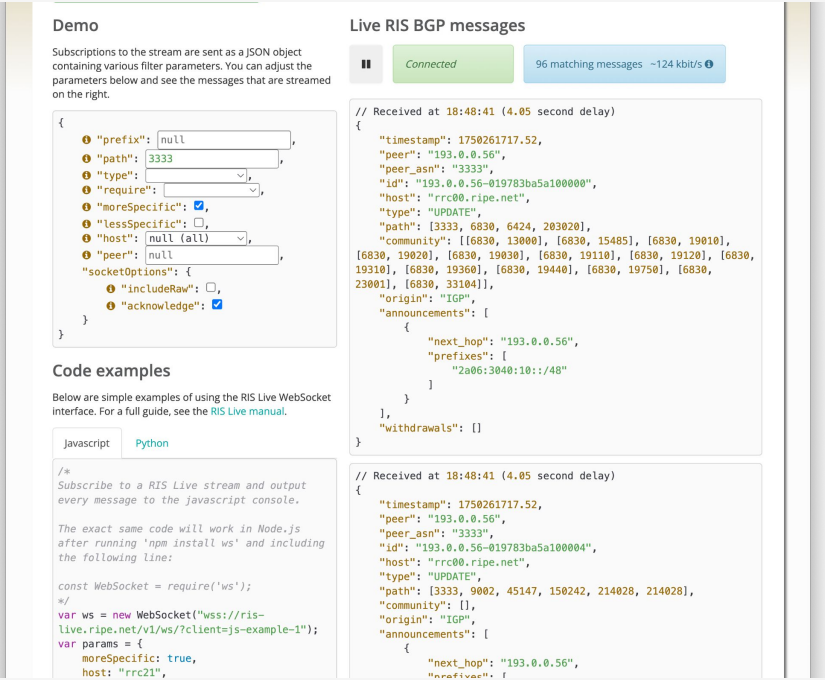
- Use it like any REST API
- Many ways to process this data

```
❯ curl -s "https://stat.ripe.net/data/looking-glass/data.json?resource=193.0.14.0/24"
 | jq '.data.rrcs[]'
{
  "rrc": "RRC00",
  "location": "Amsterdam, Netherlands",
  "peers": [
    {
      "asn_origin": "25152",
      "as_path": "34854 6939 25152",
      "community": "34854:1000",
      "largeCommunity": "",
      "extendedCommunity": "",
      "last_updated": "2025-06-18T10:55:20",
      "prefix": "193.0.14.0/24",
      "peer": "2.56.11.1",
      "origin": "IGP",
      "next_hop": "2.56.11.1",
      "latest_time": "2025-06-18T21:44:55"
    },
    {
      "asn_origin": "25152",
      "as_path": "59919 25152",
      "community": "25152:1028 59919:65001 64512:11 64512:21 64512:31",
      "largeCommunity": "25152:1028:61968",
      "extendedCommunity": "64512:11",
```

How to use RIS Data Yourself?

Example: RIS-live

- https://ris-live.ripe.net/
- A websocket with BGP update messages in JSON format
- Most suitable for getting a *filtered stream* of BGP updates.



"Easy" to use from your own code.

Also used in open source projects and commercial projects.

Conclusion

You *are* probably using RIS data

- RIS is one of the key tools for capturing the development of the Internet
- You *are* probably using RIS data indirectly
- You *can* use this data yourself
  - This can be complicated - in depth bonus?

Questions & Comments

tdekock@ripe.net,
ris@ripe.net, ris-peering@ripe.net

Working with MRT Data: Import into a Database

Workflow: parse mrt to CSV, import into database, query.

```
SELECT
    prefix,
    splitByChar(' ', as_path)[-2] AS first_upstream,
    count(*) AS cnt
FROM rib
WHERE prefix = '193.0.14.0/24'
GROUP BY ALL
ORDER BY cnt DESC
LIMIT 5

Query id: 02917ea2-4239-45cb-b21a-df41a04f9410

    ┌─prefix──────┬─first_upstream─┬─cnt─┐
1.  │ 193.0.14.0/24 │ 6939           │  70 │
2.  │ 193.0.14.0/24 │ 513            │  20 │
3.  │ 193.0.14.0/24 │ 20612          │  20 │
4.  │ 193.0.14.0/24 │ 28186          │  15 │
5.  │ 193.0.14.0/24 │ 1103           │  13 │
    └─────────────┴────────────────┴─────┘

5 rows in set. Elapsed: 0.019 sec. Processed 4.64 million rows, 157.65 MB (250.59 million rows/s., 8.52 GB/s.)
Peak memory usage: 249.23 KiB.
```

~/tmp

```
~/tmp
❯ wget https://data.ris.ripe.net/rrc00/2025.09/bview.20250905.0800.gz
--2025-09-05 12:07:50--  https://data.ris.ripe.net/rrc00/2025.09/bview.20250905.0800.gz
Resolving data.ris.ripe.net (data.ris.ripe.net)... 2001:67c:2e8:25::c100:b18, 193.0.11.24
Connecting to data.ris.ripe.net (data.ris.ripe.net)|2001:67c:2e8:25::c100:b18|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 406560243 (388M) [application/octet-stream]
Saving to: ‘bview.20250905.0800.gz’

bview.20250905.0800.gz        100%[=============================================>] 387.73M  3.65MB/s    in 54s

2025-09-05 12:08:44 (7.18 MB/s) - ‘bview.20250905.0800.gz’ saved [406560243/406560243]


~/tmp 54s
❯ bgpreader -m -d singlefile -o rib-file=bview.20250905.0800.gz | grep '193.0.14.0/24' | head -n 100 | fold -w 120 -s
WARN: No time window specified, defaulting to all available data
TABLE_DUMP2|1757059200|B|80.77.16.114|34549|193.0.14.0/24|34549 25152|IGP|80.77.16.114|0|0|34549:200 34549:10000|NAG||
TABLE_DUMP2|1757059200|B|165.16.221.66|37721|193.0.14.0/24|37721 25152|IGP|165.16.221.66|0|0|37721:2000 37721:6002
37721:6003 37721:6004 37721:11000 37721:11100 37721:11101|NAG||
TABLE_DUMP2|1757059200|B|185.210.224.254|49432|193.0.14.0/24|49432 25152|IGP|185.210.224.254|0|1000||NAG||
TABLE_DUMP2|1757059200|B|193.33.94.251|58057|193.0.14.0/24|58057 1836 25152|IGP|193.33.94.251|0|0|1836:20000 1836:110
1836:3000 1836:3020 58057:65010 25152:1016|NAG||
TABLE_DUMP2|1757059200|B|202.150.221.37|38001|193.0.14.0/24|38001 136168 137955 25152|IGP|202.150.221.37|0|0||NAG||
TABLE_DUMP2|1757059200|B|5.255.90.109|202365|193.0.14.0/24|202365 43727 9198 25152|IGP|5.255.90.109|0|0|43727:0
43727:500|NAG||
TABLE_DUMP2|1757059200|B|45.61.0.85|22652|193.0.14.0/24|22652 6939 25152|IGP|45.61.0.85|0|0||NAG||
TABLE_DUMP2|1757059200|B|49.12.70.222|44393|193.0.14.0/24|44393 394256 25152|IGP|49.12.70.222|0|0|25152:1148|NAG||
TABLE_DUMP2|1757059200|B|89.234.186.6|204092|193.0.14.0/24|204092 6939 25152|IGP|89.234.186.6|0|150|64496:100
64496:2150|NAG||
TABLE_DUMP2|1757059200|B|165.254.255.2|15562|193.0.14.0/24|15562 2914 12859 25152|IGP|165.254.255.2|0|0|2914:410
2914:1206 2914:2203 2914:3200 12859:4000 25152:1|NAG||
TABLE_DUMP2|1757059200|B|193.163.86.231|34800|193.0.14.0/24|34800 58057 1103 25152|IGP|193.163.86.231|0|0|2603:665
2603:667 11537:40 11537:888 20965:3000 20965:5000 25152:1|NAG||
TABLE_DUMP2|1757059200|B|45.12.55.0|208972|193.0.14.0/24|208972 6939 25152|IGP|45.12.55.0|0|0||NAG||
TABLE_DUMP2|1757059200|B|102.217.156.3|328977|193.0.14.0/24|328977 25152|IGP|102.217.156.3|0|0|25152:1356|NAG||
```

```python
import bgpkit
from collections import Counter

views = bgpkit.Broker().query(ts_start="2025-09-05T08:00:00", ts_end="2025-09-05T08:00:00", data_type="rib", project="riperis")

upstreams = Counter()
for file_metadata in views:          # b.gtld-servers.net has address 192.33.14.30
    parser = bgpkit.Parser(url=file_metadata.url, filters={"prefix": "192.33.14.0/24"})
    for route in parser:
        as_path = (route.as_path or "").split()
        if len(as_path) > 1:
            upstreams[as_path[-2]] += 1

print(f"[{elapsed()}] Results:")
for upstream, cnt in upstreams.most_common(10):
    print(f"  AS{upstream:<8} {cnt:>4}")
```

```
❯ uv run python main.py
[   0.4s] Reading 23 dump files...
[2133.8s] Results:
  AS7342       77
  AS396746     60
  AS396761     46
  AS396707     29
  AS396688     17
  AS396652     17
  AS396675     14
  AS396748     13
  AS396686     12
  AS396658     12
```