How bgproutes.io supports latest technologies such as RPKI ROV, ASPA, and BMP

Current sponsors:

Thomas Holterbach
University of Strasbourg

Joint work with:
Thomas Alfroy
Cristel Pelsser

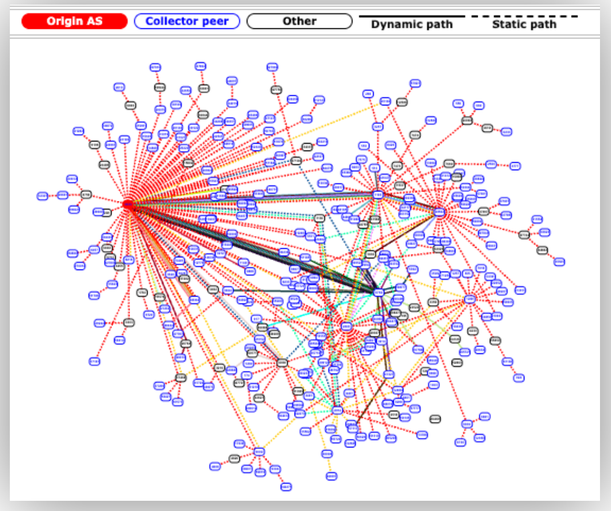
bgproutes.io

BGP data collected from real BGP routers is essential for understanding and monitoring Internet routing

arrangements between ASes.

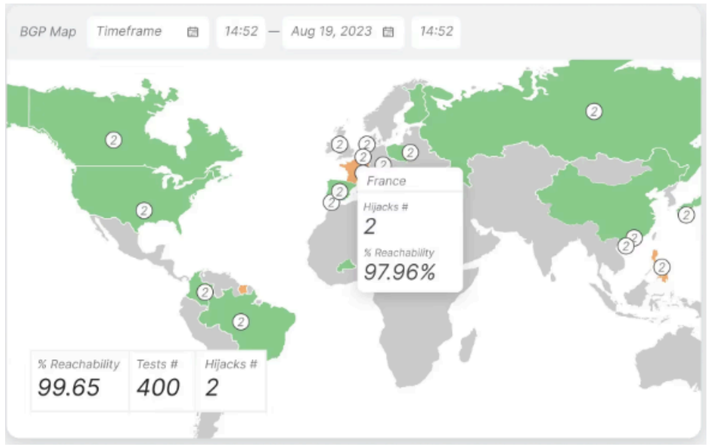
AS Rank ▲	AS Number ▽	Organization		cone size (ASes) ▽
1	3356	Level 3 Parent, LLC		53982
2	1299	Arelion Sweden AB		41919
3	174	Cogent Communications		37559
4	3257	GTT Communications Inc.		36334
5	2914	NTT America, Inc.		26976
6	6939	Hurricane Electric LLC		21697
7	6762	Telecom Italia S.p.A.		20271
8	6453	TATA COMMUNICATIONS (AMERICA) INC		19334
9	6461	Zayo Bandwidth		18116
10	3491	PCCW Global, Inc.		11530

CAIDA's AS rank



BGPlay

RIPEStat



Catchpoint's BGP map

The three longstanding platforms that collect, archive, and publicly share BGP data are RIPE RIS, RouteViews and PCH

- RIS and RouteViews collect and store data from ~1500 BGP routers (*a.k.a.* "*vantage points*")

bgproutes.io

Our new next-gen BGP route collection platform

https://bgproutes.io

You can contribute data

bgproutes.io

Our new next-gen BGP route collection platform

We launched it
1.5 years ago

https://bgproutes.io

You can contribute data

BGP data collection platforms typically pursue two main objectives

High coverage	Perpetual data retention

BGP data collection platforms typically pursue two main objectives **that are not easily compatible**

Priority	
High coverage	Perpetual data retention

BGP data collection platforms typically pursue two main objectives that are not easily compatible

	Priority	
	High coverage	Perpetual data retention
RIPE RIS RouteViews	Lower priority	High priority 

BGP data collection platforms typically pursue two main objectives that are not easily compatible

	Priority	
	High coverage	Perpetual data retention
RIPE RIS RouteViews	Lower priority	High priority 
bgproutes.io	High priority 	Lower priority

bgproutes.io

Our new next-gen BGP route collection platform

- New strategies for high coverage (e.g., BMP)

https://bgproutes.io

You can contribute data

bgproutes.io relies on three ingredients to increase its coverage

Ingredient #1

Ingredient #2

Ingredient #3

bgproutes.io relies on three ingredients to increase its coverage

Ingredient #1 | Automated peering establishment

Ingredient #2 |

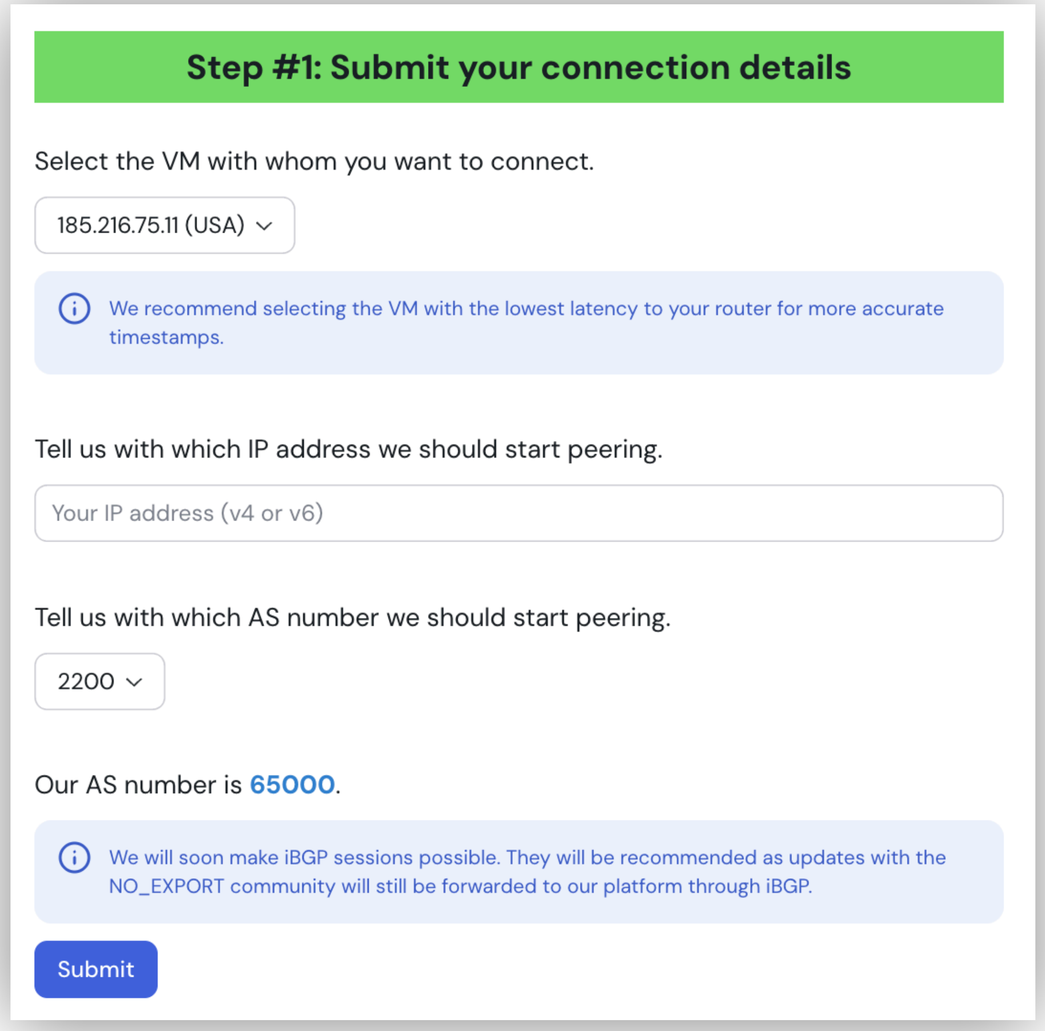
Ingredient #3 |

bgproutes.io simplifies, automates
and opens BGP data contribution to every ASN

bgproutes.io simplifies, automates and opens BGP data contribution to every ASN

Network operators can authenticate using PeeringDB

Network operators just have to fill a form to start peering with bgproutes.io



bgproutes.io/contribute

Today, **bgproutes.io** collects BGP data from ~100 networks

Today, **bgproutes.io** collects BGP data from ~100 networks

This is useful, but far from sufficient for accurate monitoring

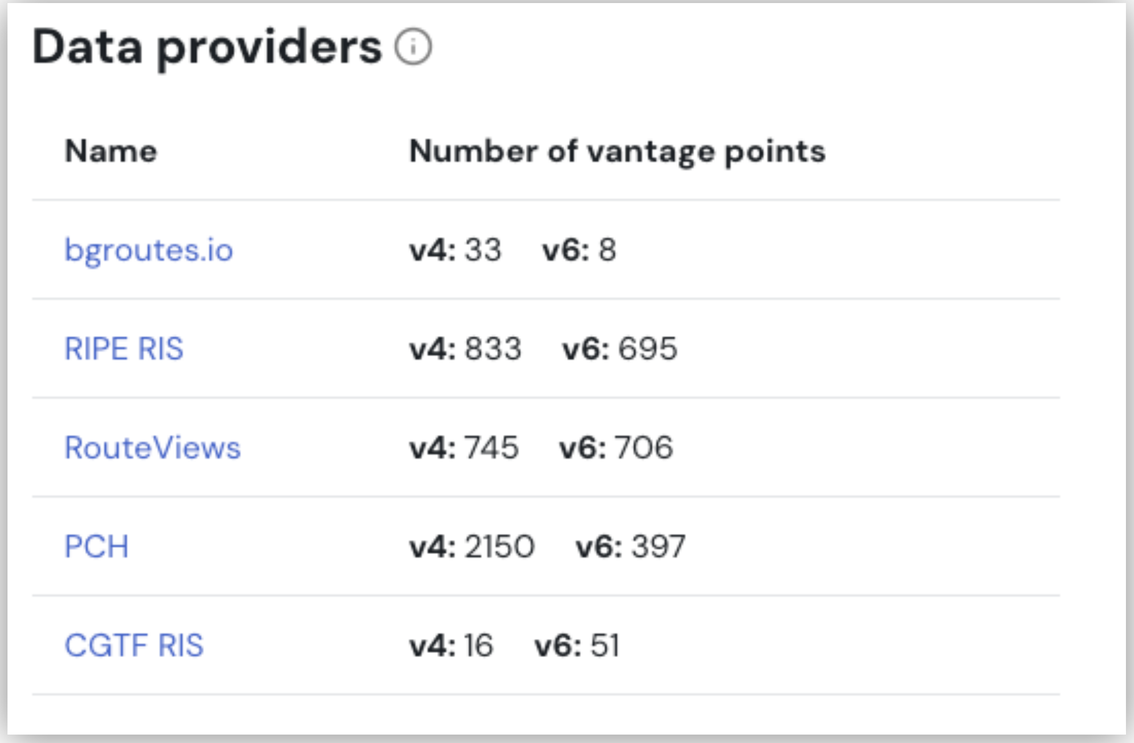
bgproutes.io relies on three ingredients to increase its coverage

Ingredient #1 | Automated peering establishment

Ingredient #2 | Data aggregation

Ingredient #3 |

bgproutes.io increases coverage by centralising all the data collected by RIPE RIS, RouteViews, PCH, CGTF into a single database

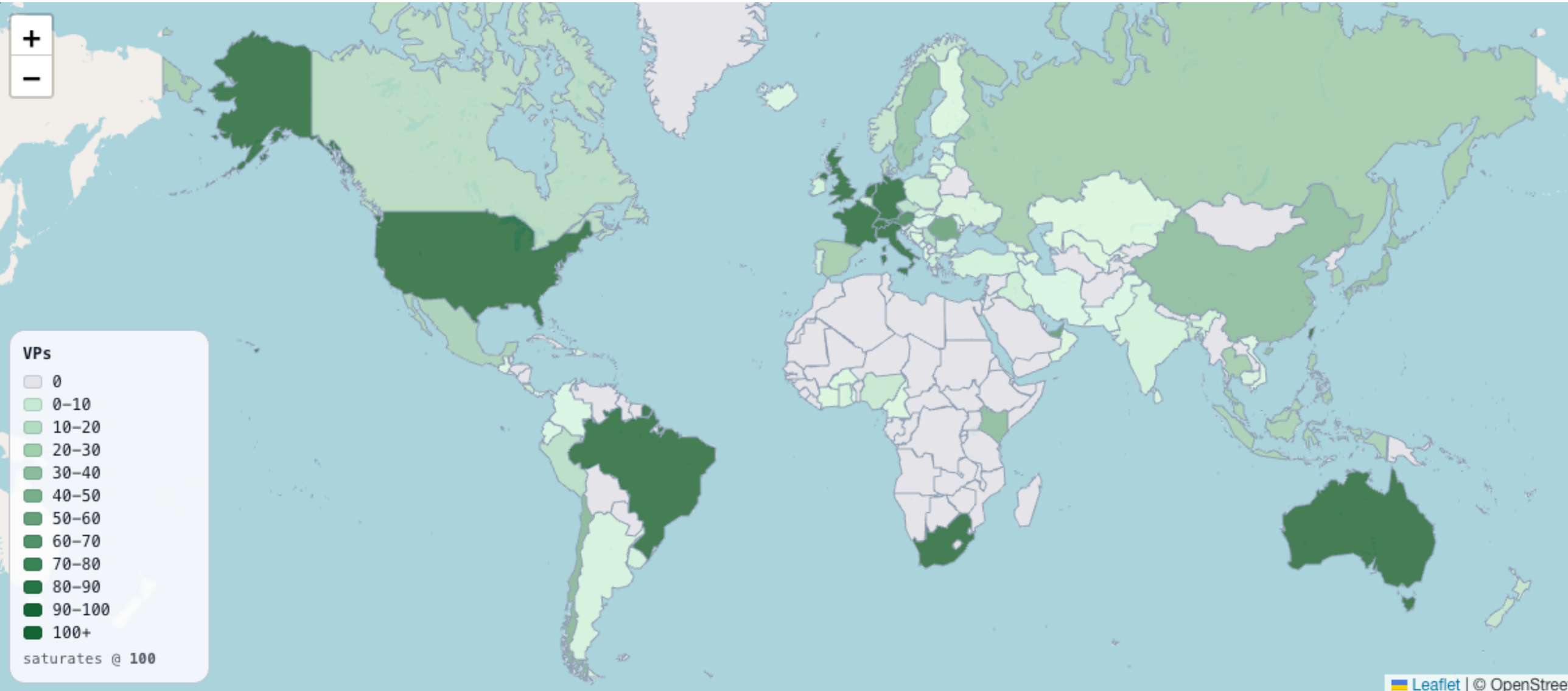


Data providers

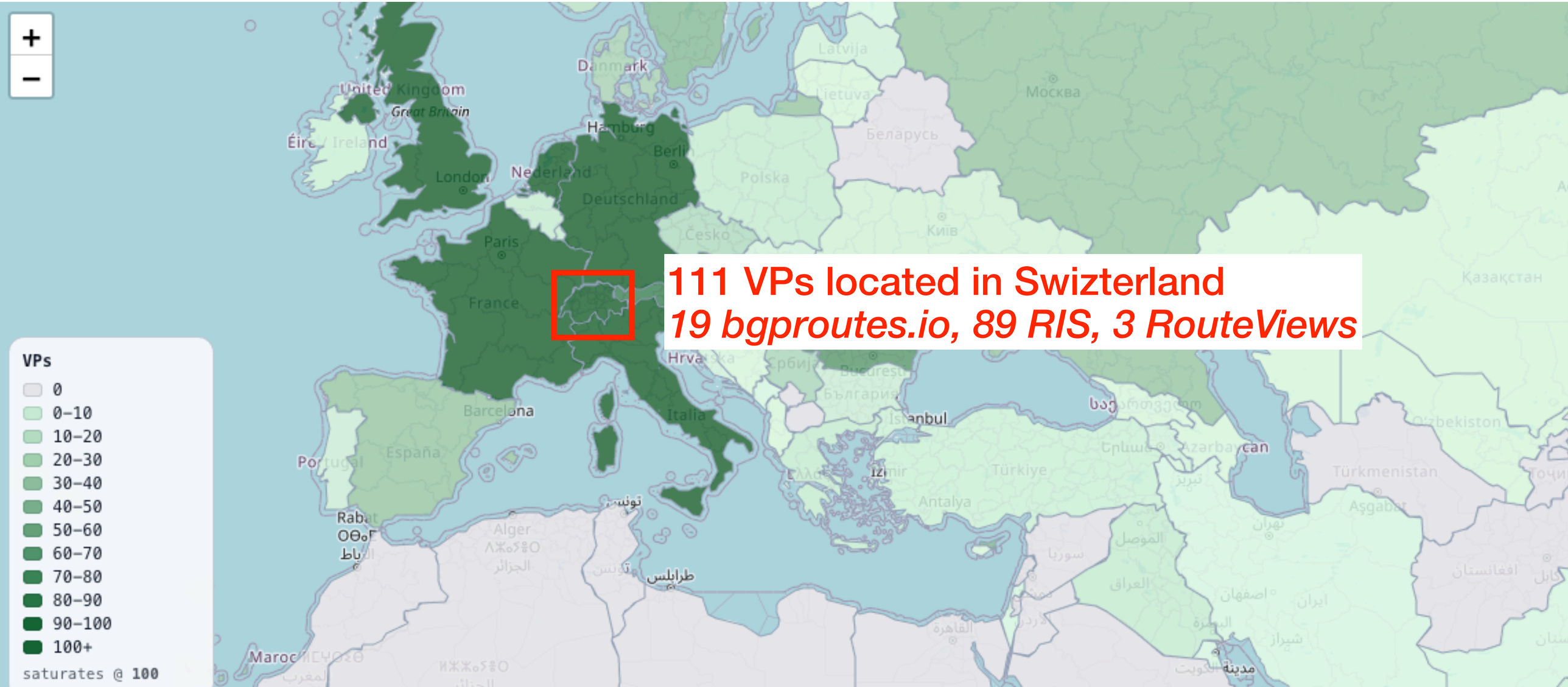
Name	Number of vantage points
bgroutes.io	**v4:** 33 **v6:** 8
RIPE RIS	**v4:** 833 **v6:** 695
RouteViews	**v4:** 745 **v6:** 706
PCH	**v4:** 2150 **v6:** 397
CGTF RIS	**v4:** 16 **v6:** 51

https://vp-explorer.bgproutes.io

bgproutes.io already stores data
for more than 5000 vantage points



bgproutes.io already stores data for more than 5000 vantage points



bgproutes.io relies on three ingredients to increase its coverage

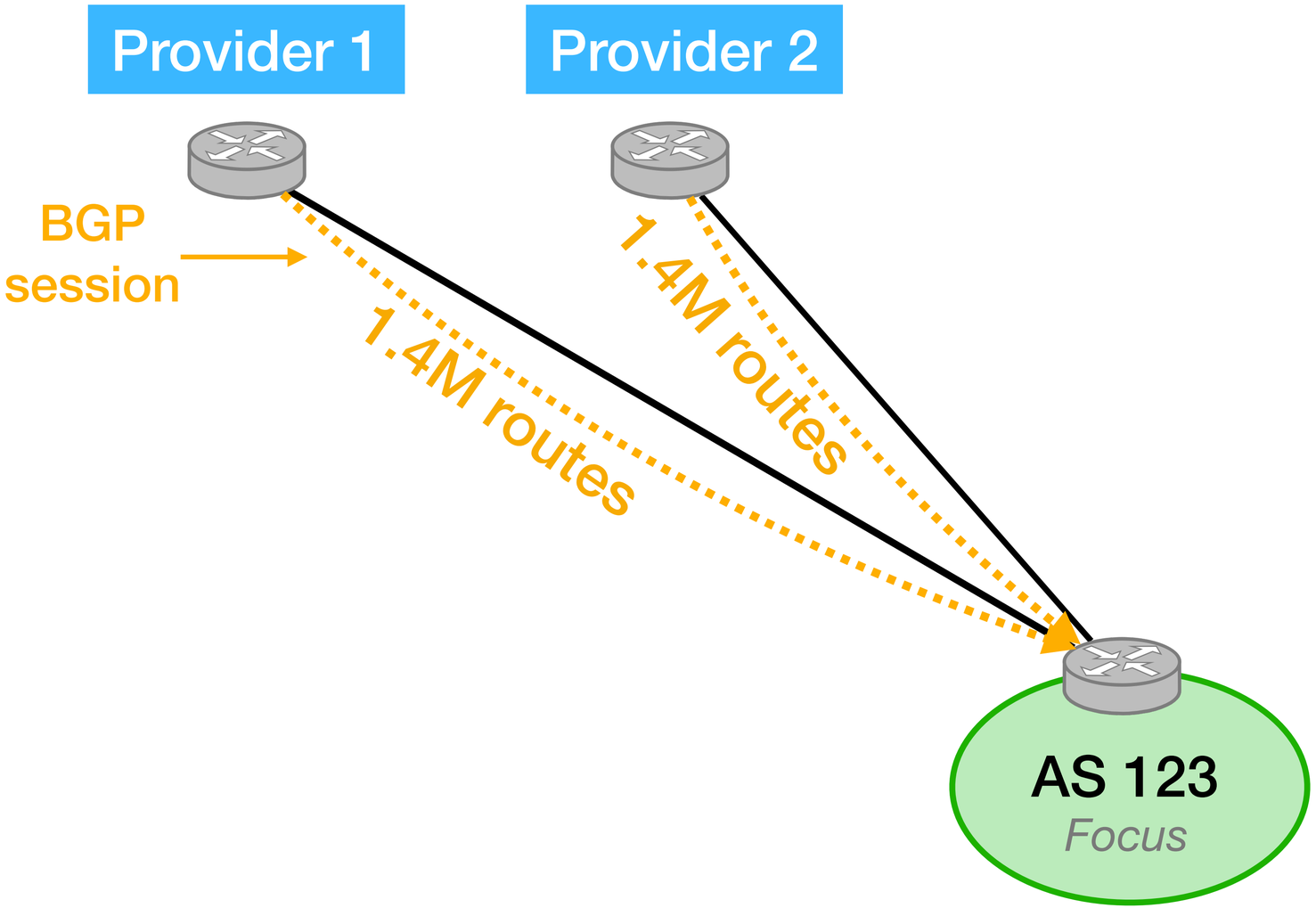
Ingredient #1 — Automated peering establishment

Ingredient #2 — Data aggregation

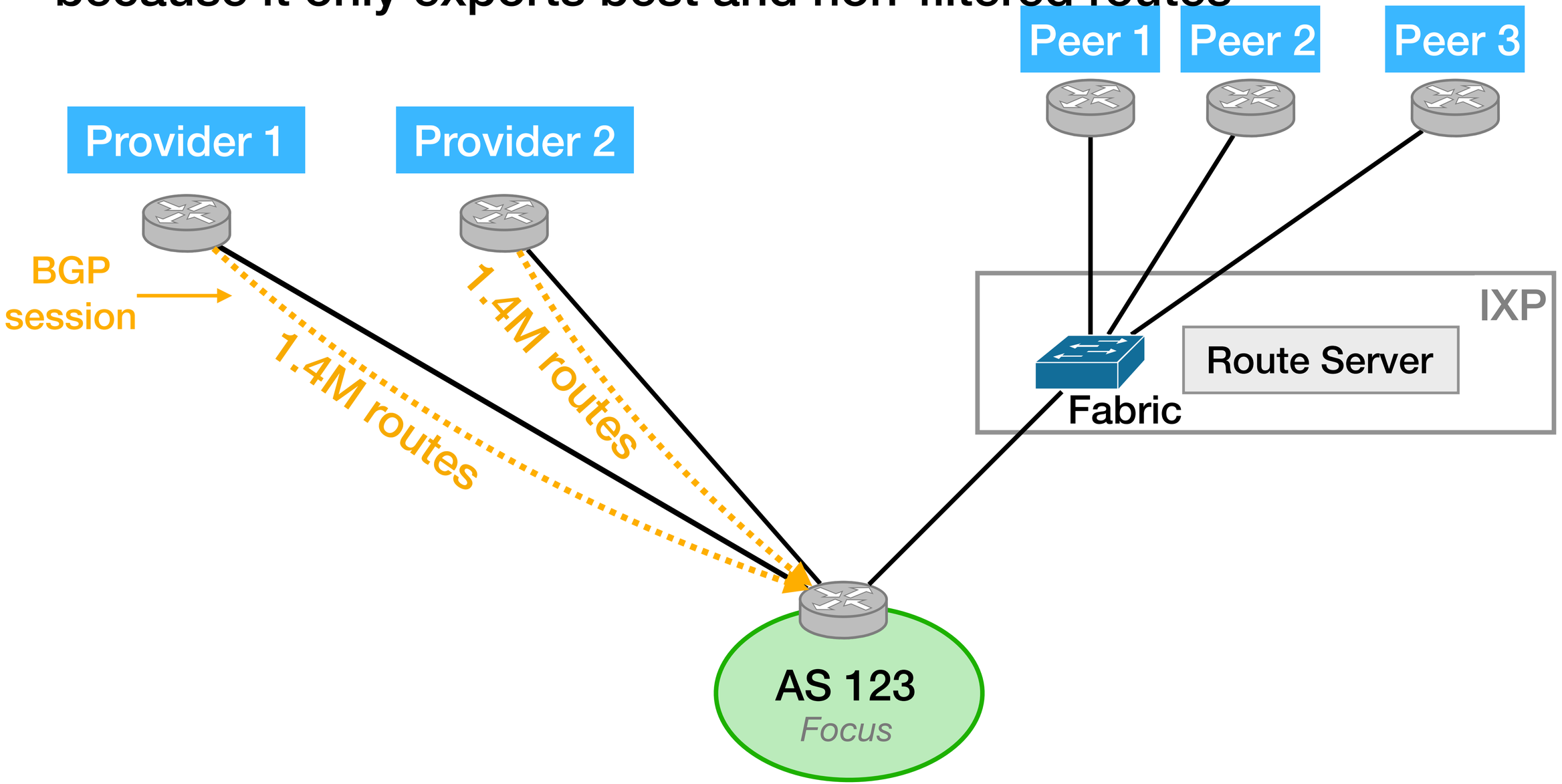
Ingredient #3 — **BMP**

BGP wastes useful information when used for monitoring because it only exports best and non-filtered routes

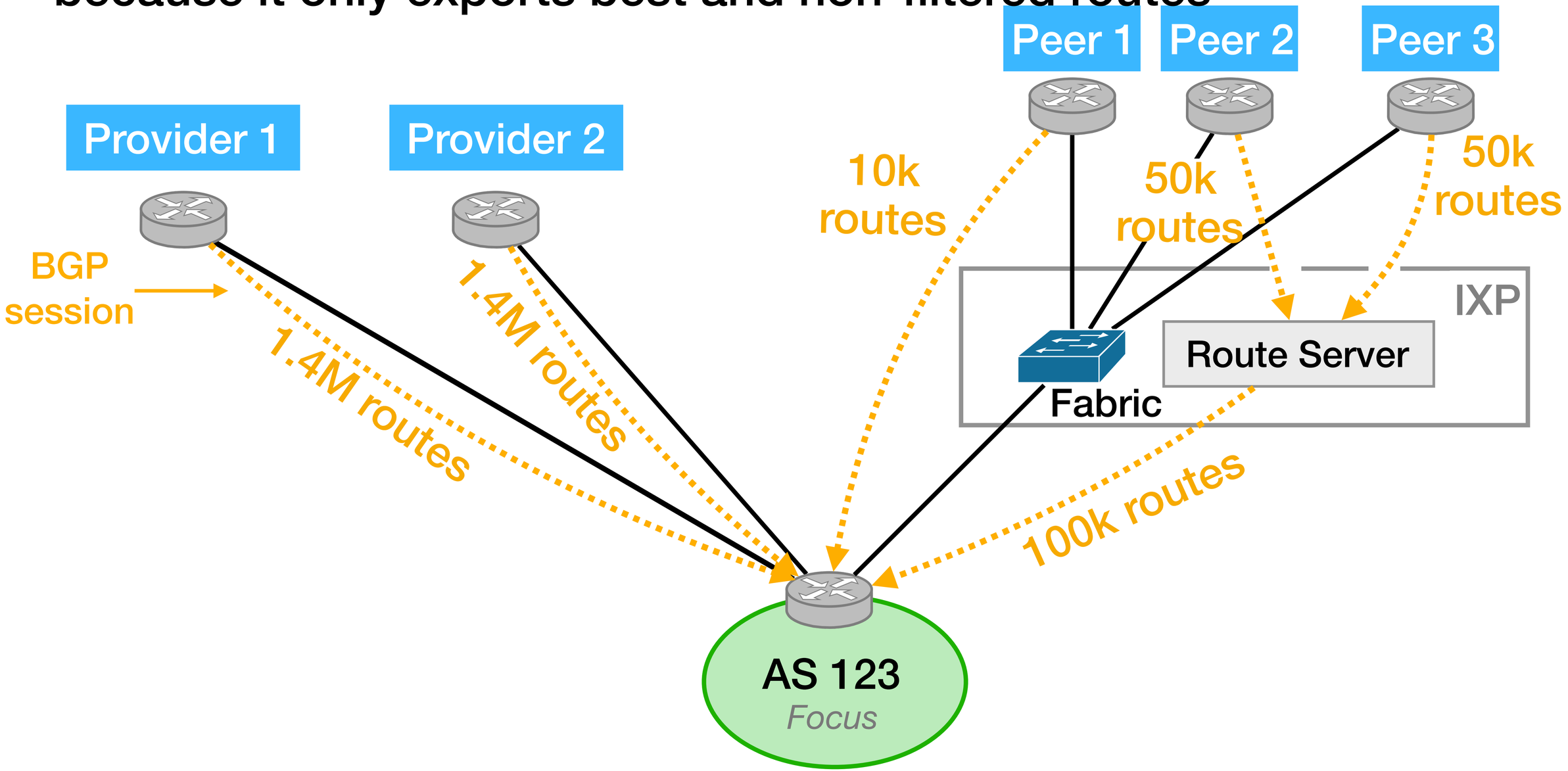
BGP wastes useful information when used for monitoring because it only exports best and non-filtered routes



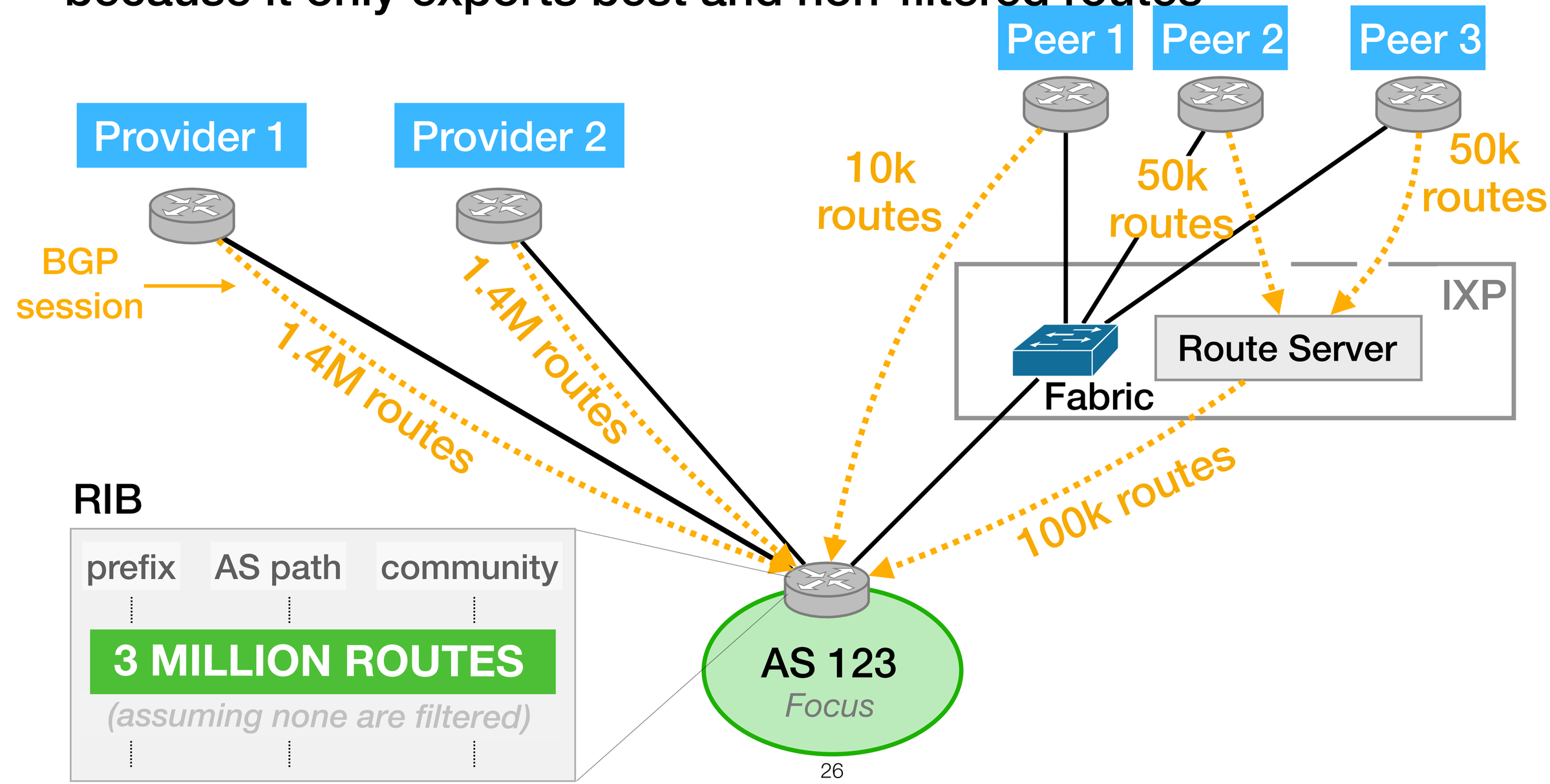
BGP wastes useful information when used for monitoring because it only exports best and non-filtered routes



BGP wastes useful information when used for monitoring because it only exports best and non-filtered routes

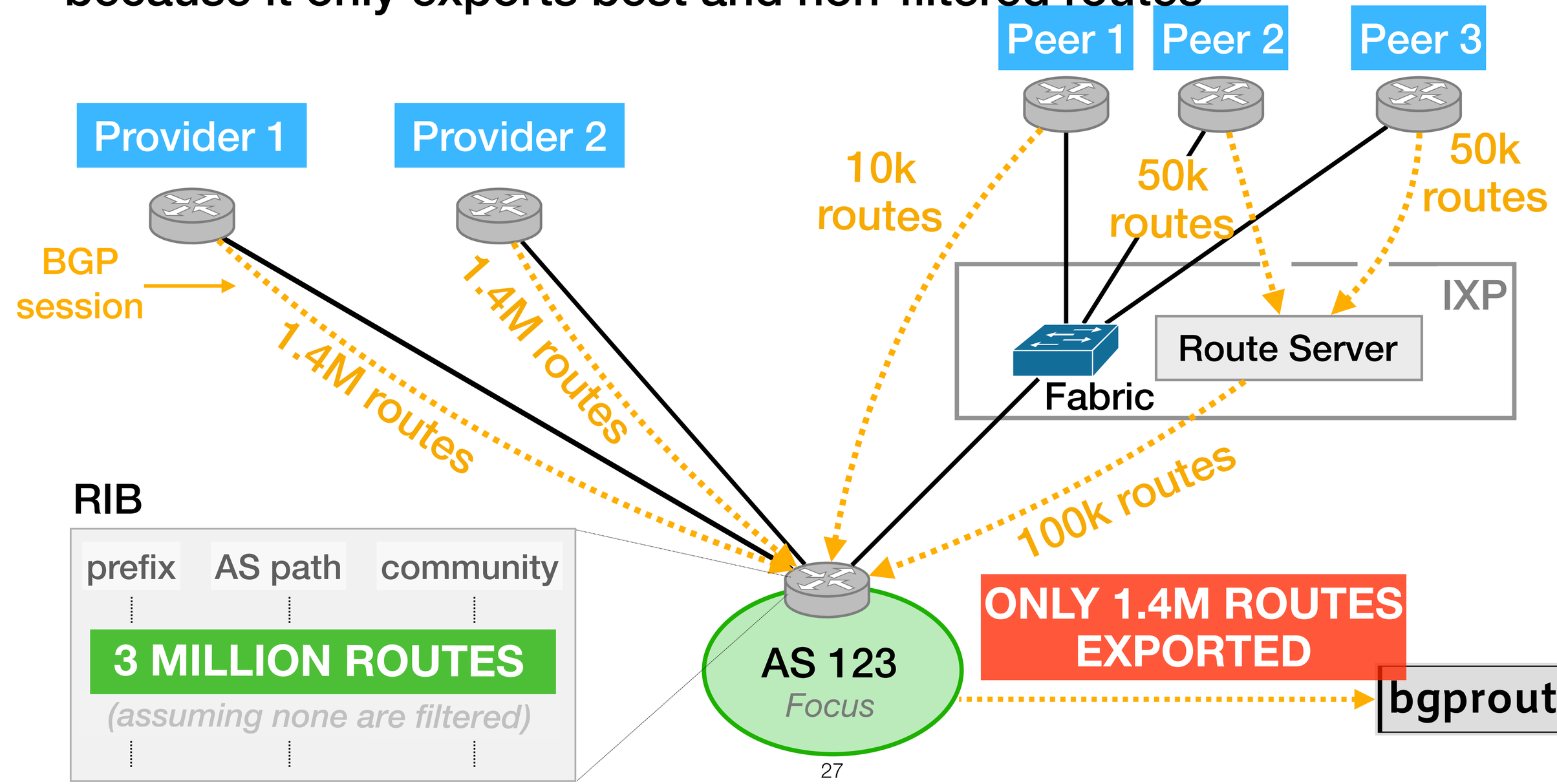


BGP wastes useful information when used for monitoring because it only exports best and non-filtered routes



BGP wastes useful information when used for monitoring

because it only exports best and non-filtered routes



The iceberg analogy works well here

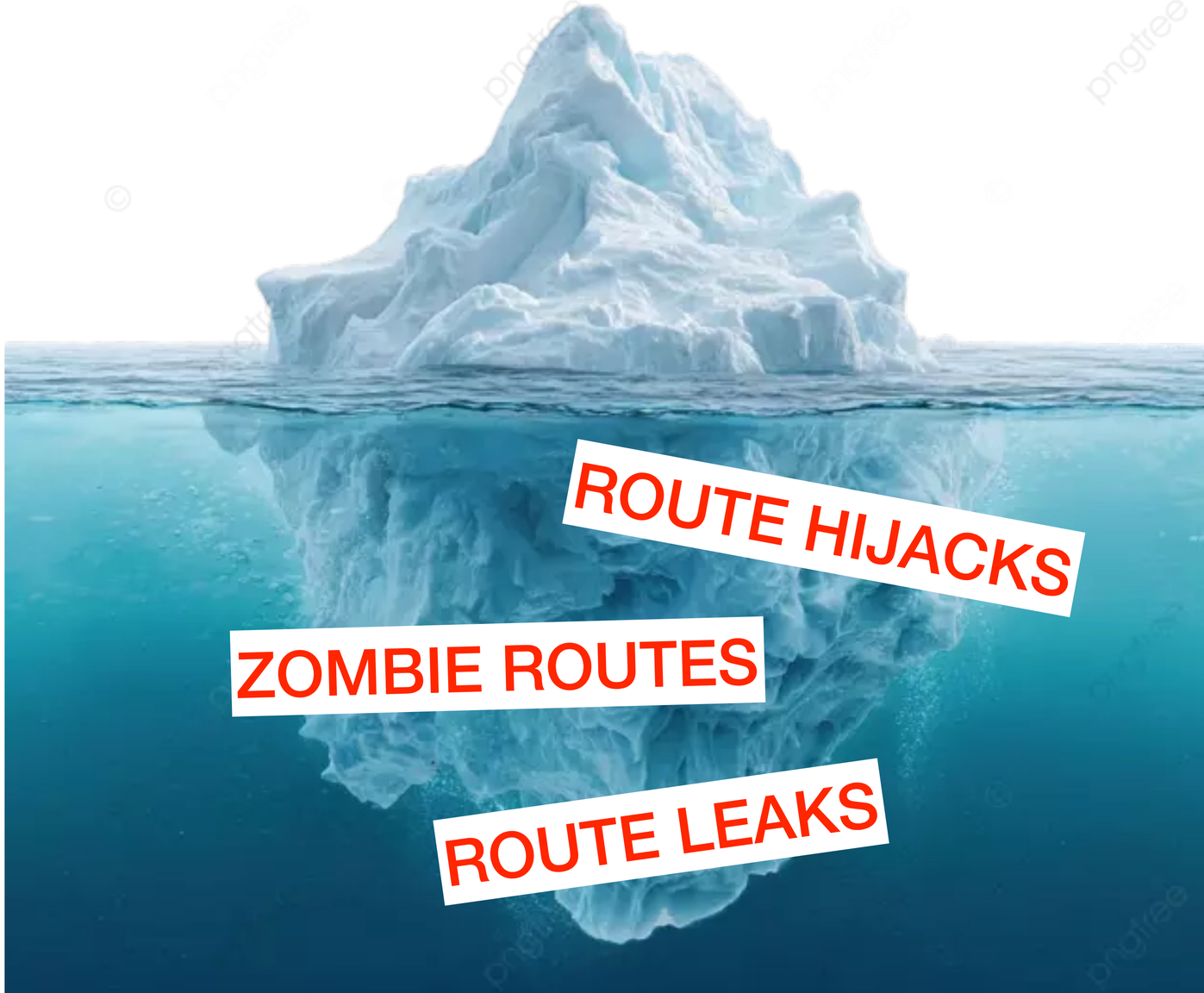
Routes collected from BGP sessions are the top of the iceberg

Most remain invisible

The iceberg analogy works well here

Routes collected from BGP sessions are the top of the iceberg

Most remain invisible



bgproutes.io can gather BGP route updates via the BGP Monitoring Protocol (BMP) — *and it is a game changer!*

- BMP exports all route updates received from every peer
  That means, backup routes are exported!

bgproutes.io can gather BGP route updates via the BGP Monitoring Protocol (BMP) — *and it is a game changer!*

- BMP exports all route updates received from every peer
  That means, backup routes are exported!

- BMP exports route updates at all stages of the BGP processing pipeline
  Including the pre-filter stage — which is not the case with BGP Add Path

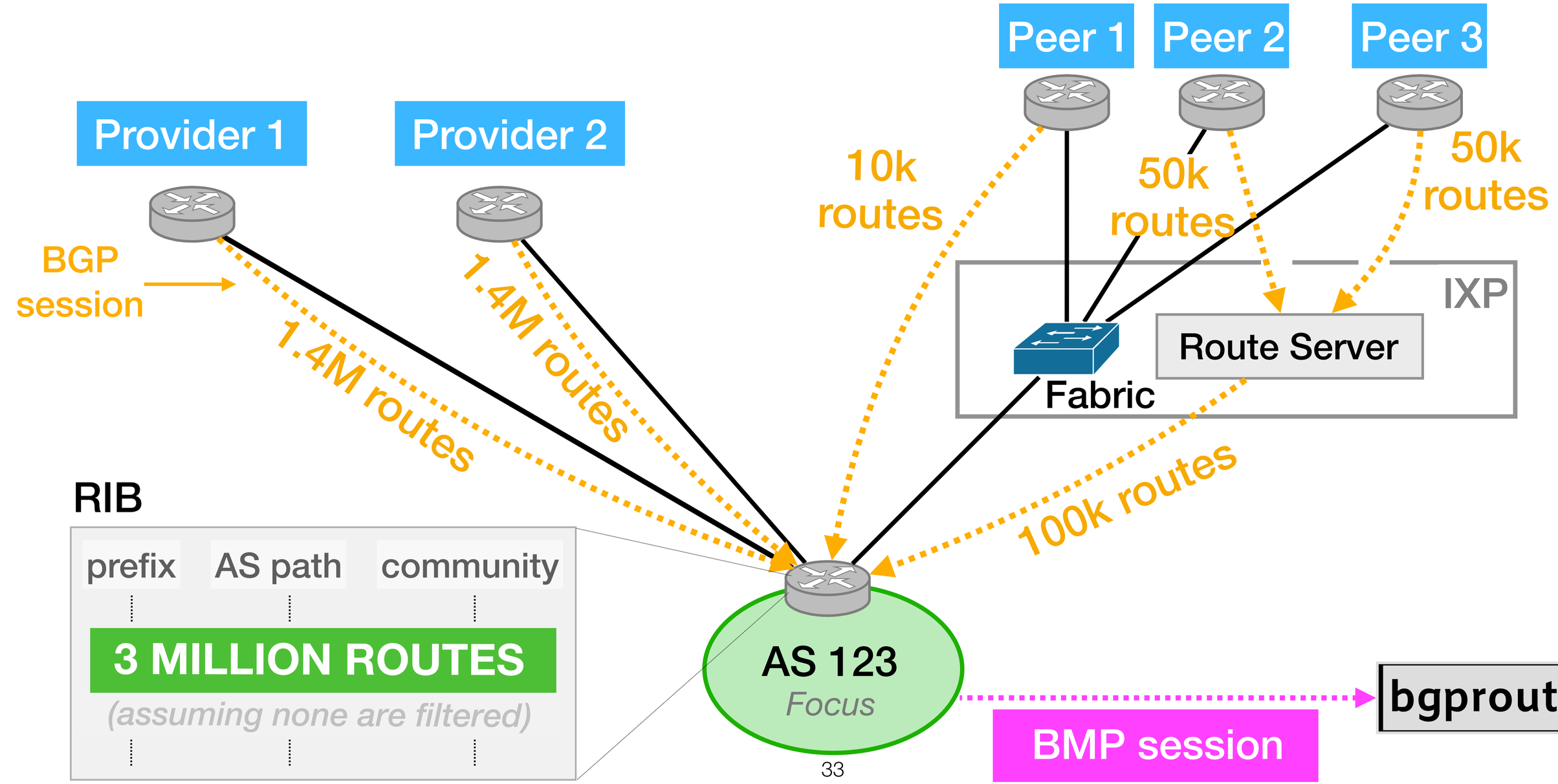
bgproutes.io can gather BGP route updates via the BGP Monitoring Protocol (BMP) — *and it is a game changer!*

- BMP exports all route updates received from every peer
  That means, backup routes are exported!

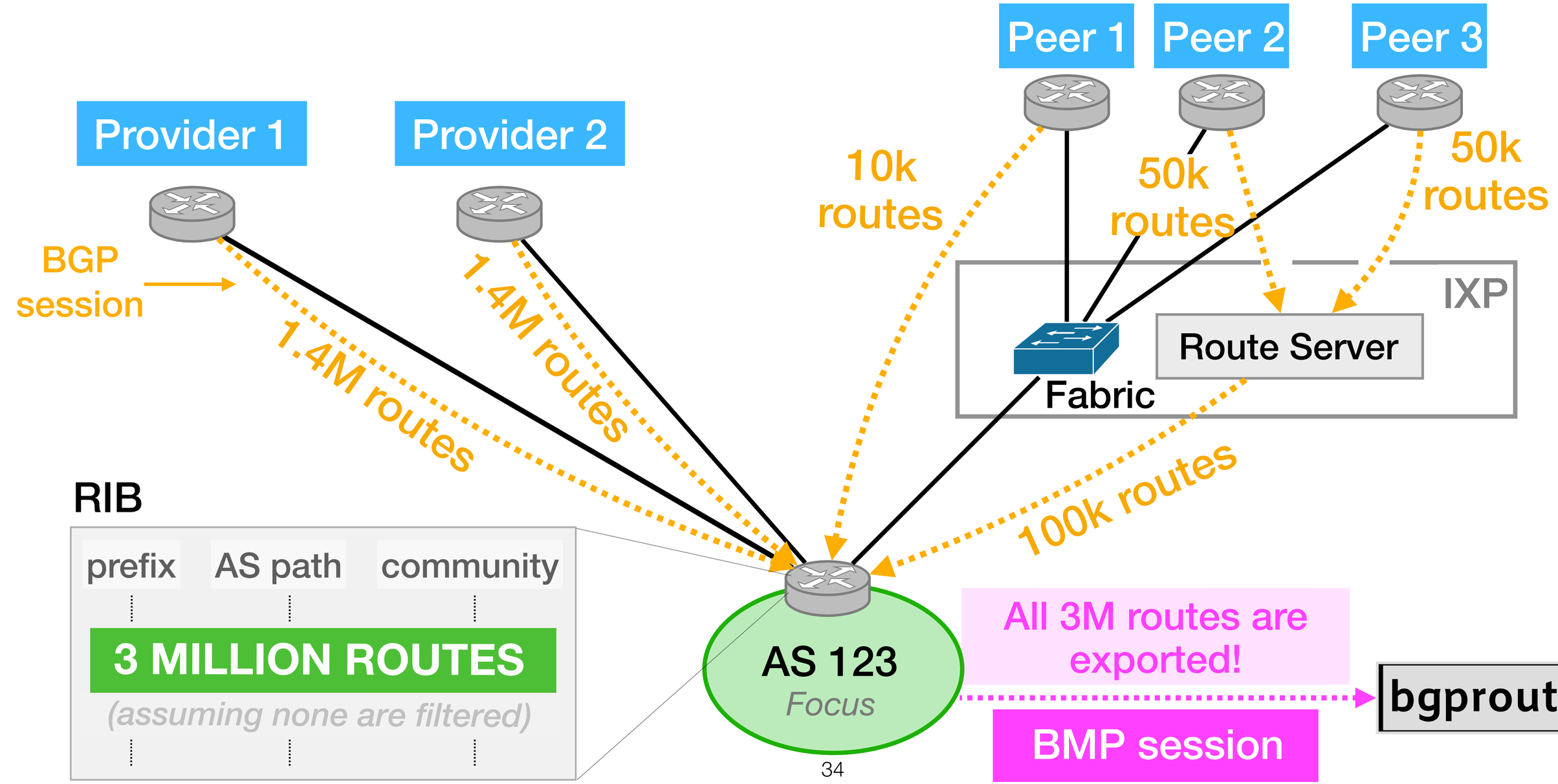
- BMP exports route updates at all stages of the BGP processing pipeline
  Including the pre-filter stage — which is not the case with BGP Add Path

→ BMP is also simple to configure

With BMP: no more useful information wasted



With BMP: no more useful information wasted



bgproutes.io has ~39 BMP sessions with 19 networks and the previous example translates well in practice

bgproutes.io has ~39 BMP sessions with 19 networks
and the previous example translates well in practice

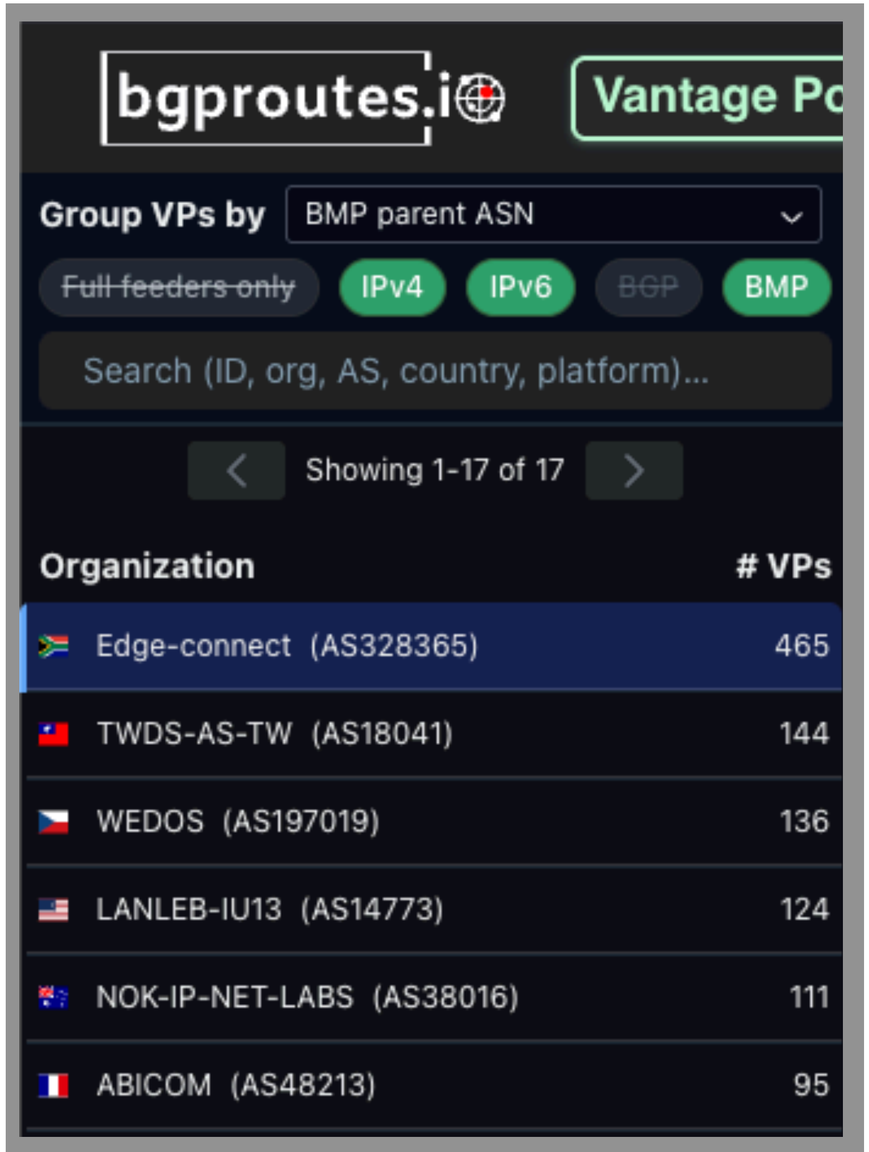
Example: **Edge-connect** *(AS328365)* from South Africa 
exports all routes it receives from 465 peers!

bgproutes.io has ~39 BMP sessions with 19 networks and the previous example translates well in practice

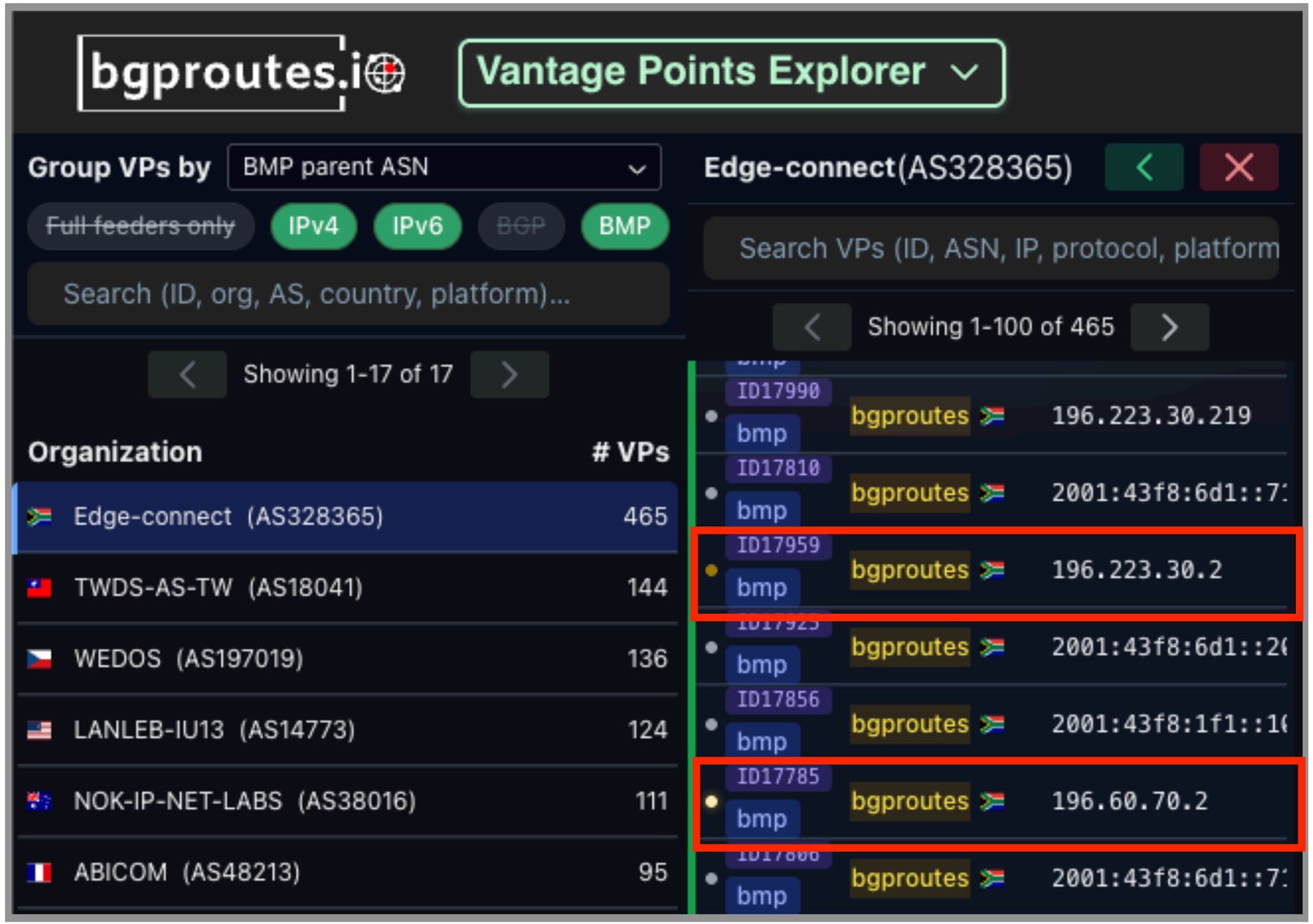
Example: **Edge-connect** *(AS328365)* from South Africa 🇿🇦
exports all routes it receives from 465 peers!

- One IPv6 full feed from Hurricane Electric
- Few feeds from IXP route servers (e.g. NAPAfrica, INX-ZA)
- Hundreds of feeds from bilateral sessions through IXPs

Several BMP sessions export data for more than 100 peers!
They are visible in our dashboard “Vantage Points Explorer”



Several BMP sessions exports data for more than 100 peers!
They are visible in our dashboard “Vantage Points Explorer”

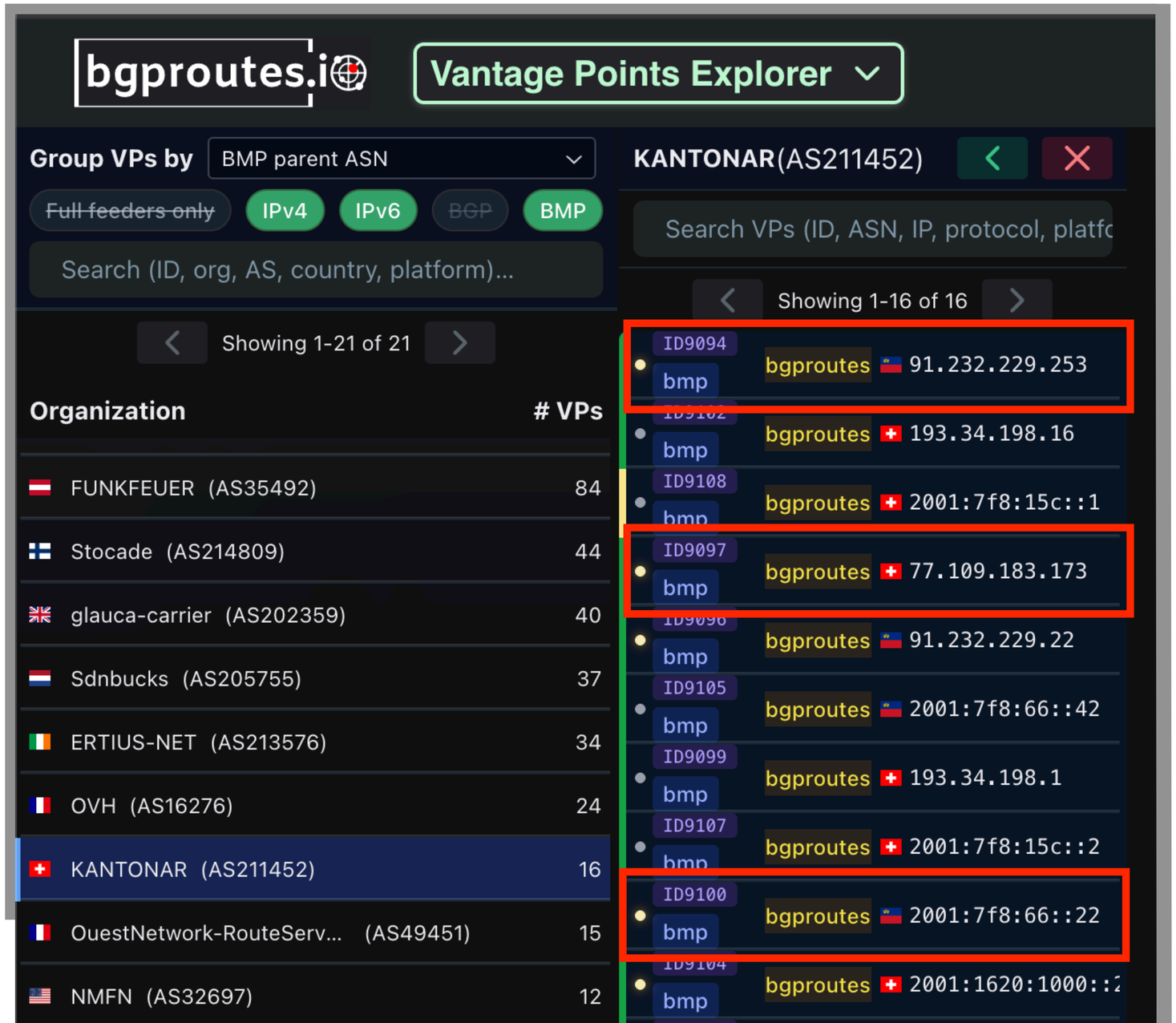


INX-ZA route server

NAPAFRICA route server

Canton of Appenzell Ausserrhoden is already connected with BMP!

Canton of Appenzell Ausserrhoden is already connected with BMP!



Rheintal IX

Init7

SwissIX

bgproutes.io

Our new next-gen BGP route collection platform

- New strategies for high coverage (e.g., BMP)
- Support the latest security standards (ROV, ASPA)

https://bgproutes.io

You can contribute data

bgproutes.io stores each collected route along with its RPKI ROV and ASPA status inferred at the time the route is received

bgproutes.io stores each collected route along with its RPKI ROV and ASPA status inferred at the time the route is received

Route Origin Validation: Verify that the origin of an AS path is legitimate
Possible outcome: Valid, Invalid, Unknown

bgproutes.io stores each collected route along with its RPKI ROV and ASPA status inferred at the time the route is received

Route Origin Validation: Verify that the origin of an AS path is legitimate
Possible outcome: Valid, Invalid, Unknown

ASPA-based Validation: Verify that there is no valley-free violation in the AS path
Possible outcome:

Upstream Valid or Upstream Invalid or Upstream Unknown	And	Downstream Valid or Downstream Invalid or Downstream Unknown

bgproutes.io

Our new next-gen BGP route collection platform

- New strategies for high coverage (e.g., BMP)
- Support the latest security standards (ROV, ASPA)
- New API and dashboards

https://bgproutes.io

You can contribute data

Collecting more data is useful

But only if users can effectively process it

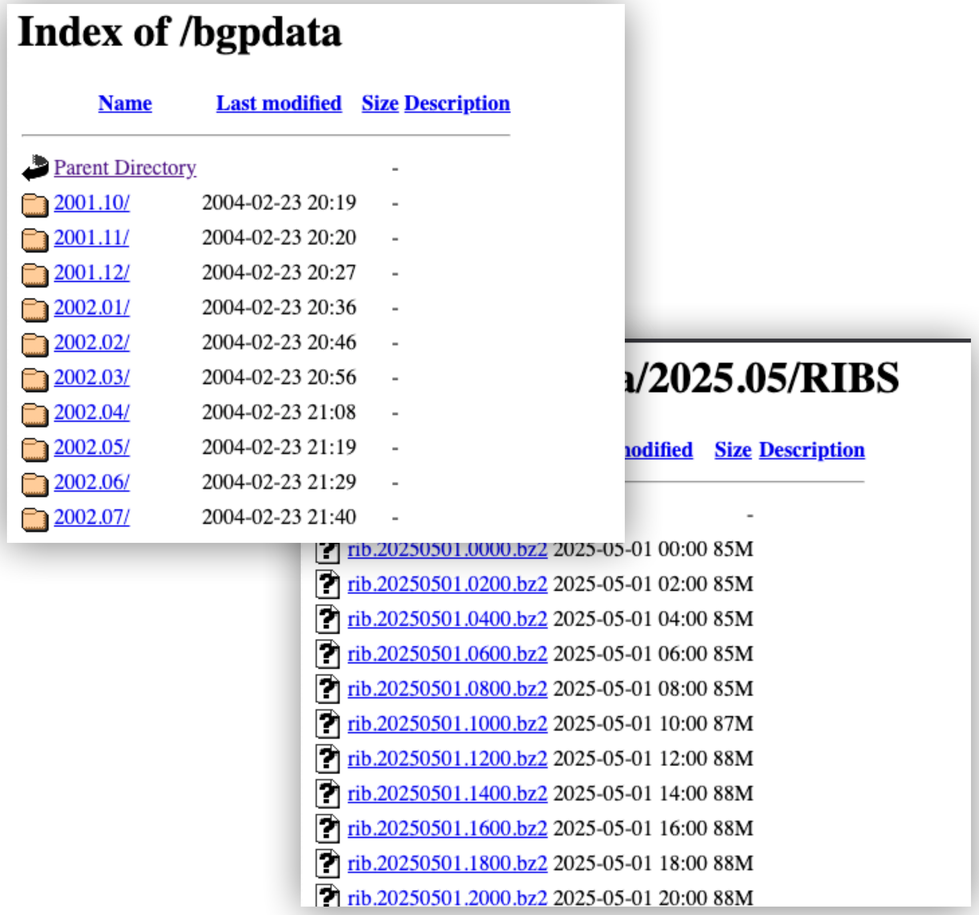
Collecting more data is useful

But only if users can effectively process it

Current MRT archives consist of 1000+ compressed files for updates and RIB dumps

BGP data is intermingled across all these files

➡ MRT is hard to process



Typical MRT archive

bgproutes.io comes with a simple and fast API that provides high-granularity access to BGP data

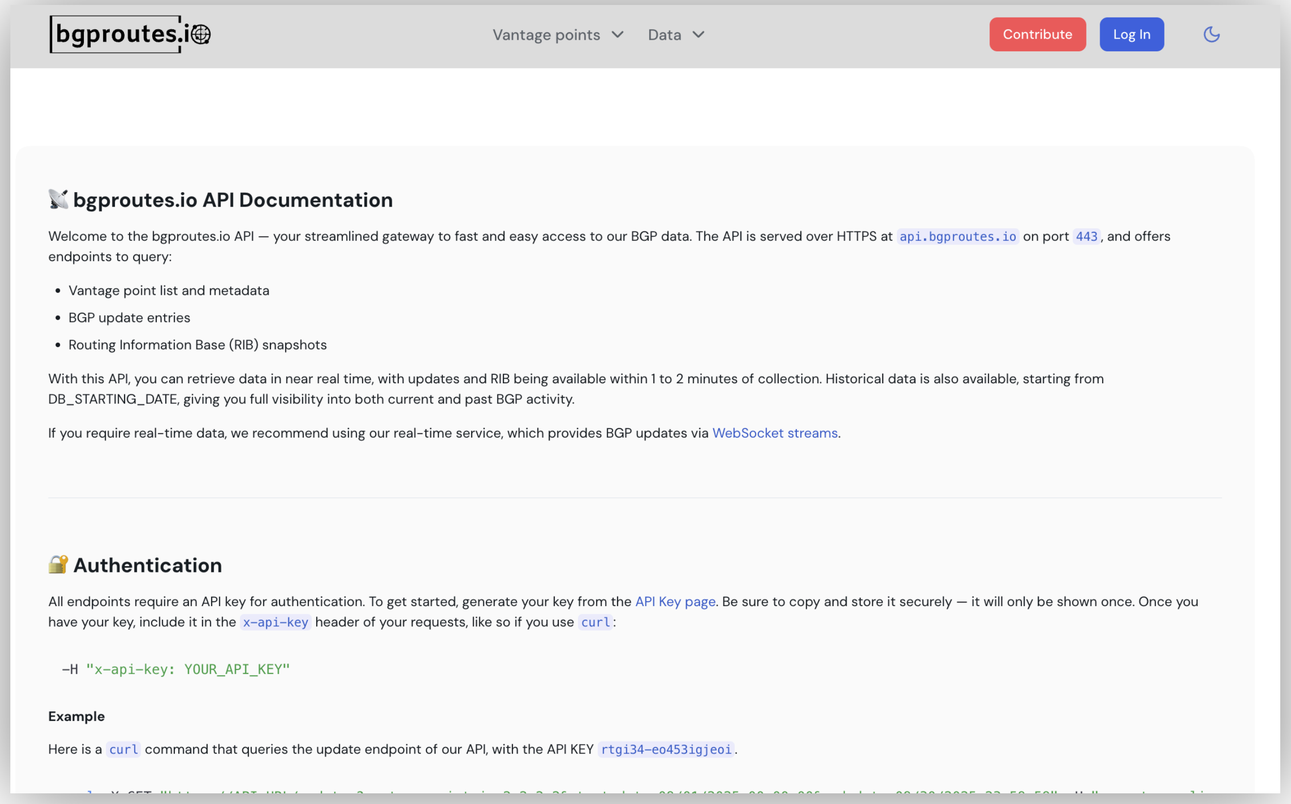
The API offers four endpoints:

vantage_points

updates

rib

topology



bgproutes.io/data_api

Imagine a network operator trying to retrieve all BGP updates for which their ASN appears in the AS path on April 28, 2026

```
# Let’s use our Python client to retrieve the data
from pybgproutesapi import vantage_points, updates
```

Imagine a network operator trying to retrieve all BGP updates for which their ASN appears in the AS path on April 28, 2026

```
# Let’s use our Python client to retrieve the data
from pybgproutesapi import vantage_points, updates

# Let’s retrieve the vantage points to use
vps = vantage_points(source=[‘bgproutes.io', ‘ris’])
```

Imagine a network operator trying to retrieve all BGP updates for which their ASN appears in the AS path on April 28, 2026

```
# Let’s use our Python client to retrieve the data
from pybgproutesapi import vantage_points, updates

# Let’s retrieve the vantage points to use
vps = vantage_points(source=[‘bgproutes.io', ‘ris’])

# We retrieve updates and print them for every VP
for vp in vps:
    vp_upd = updates(
        vp,
        start_date=“2026-02-10T00:00:00”,
        end_date=“2026-02-11T00:00:00”,
        aspath_regexp=“(^|  )3(  |$)“)

    print (vp_upd)
```

Imagine a network operator trying to retrieve all BGP updates for which their ASN appears in the AS path on April 28, 2026

```
# Let’s use our Python client to retrieve the data
from pybgproutesapi import vantage_points, updat

# Let’s retrieve the vantage points to use
vps = vantage_points(source=[‘bgproutes.io', ‘ris’])

# We retrieve updates and print them for every VP
for vp in vps:
    vp_upd = updates(
        vp,
        start_date=“2026-02-10T00:00:00”,
        end_date=“2026-02-11T00:00:00”,
        aspath_regexp=“(^|  )3(  |$)“)

    print (vp_upd)
```


Example of an output

Time	Prefix	ASPath
T1	1/24	1-3-7
T2	2/24	3-9-4
T3	3/24	2-4-3
T4	4/24	3-4-5

Other attributes omitted for space

Each returned update includes the RPKI ROV and ASPA status

```
from pybgproutesapi import vantage_points, updates
vps = vantage_points(source=['bgproutes.io', 'ris'])

# We retrieve updates and print them for every VP

for vp in vps:
    vp_upd = updates(
        vp,
        start_date="2026-02-10T00:00:00",
        end_date="2026-02-11T00:00:00",
        aspath_regexp="(^|  )3(  |$)")


    print (vp_upd)
```

Example of an output

Time	Prefix	ASPath	ROV status	ASPA status
T1	1/24	1-3-7	Valid	Up-valid Down-invalid
T2	2/24	3-9-4	Unknown	Up-valid Down-valid
T3	3/24	2-4-3	Invalid	Up-invalid Down-invalid
T4	4/24	3-4-5	Valid	Up-valid Down-Unknown

Other attributes omitted for space

Users can filter BGP route updates based on their ROV or ASPA status

Users can filter BGP route updates based on their ROV or ASPA status

```
from pybgproutesapi import vantage_points, updates
vps = vantage_points(source=['bgproutes.io', 'ris'])

# We retrieve updates and print them for every VP
for vp in vps:
    vp_upd = updates(
        vp,
        start_date="2026-02-10T00:00:00",
        end_date="2026-02-11T00:00:00",
        aspath_regexp="^3 | 3 | 3$"
        rov_status=["valid", "unknown"],
        aspa_status=['down-invalid','down-unknown'])

    print (vp_upd)
```

Example of an output

Users can filter BGP route updates based on their ROV or ASPA status

```
from pybgproutesapi import vantage_points, updates
vps = vantage_points(source=['bgproutes.io', 'ris'])

# We retrieve updates and print them for every VP
for vp in vps:
    vp_upd = updates(
        vp,
        start_date="2026-02-10T00:00:00",
        end_date="2026-02-11T00:00:00",
        aspath_regexp="^3 | 3 | 3$"
        rov_status=["valid", "unknown"],
        aspa_status=['down-invalid','down-unknown'])

    print (vp_upd)
```

Example of an output

Time	Prefix	ASPath	ROV status	ASPA status
T1	**1/24**	**1-3-7**	**Valid**	**Up-valid Down-invalid**
T2	2/24	3-9-4	Unknown	Up-valid Down-valid
T3	3/24	2-4-3	Invalid	Up-invalid Down-invalid
T4	**4/24**	**3-4-5**	**Valid**	**Up-valid Down-Unknown**

Other attributes omitted for space

Users can filter BGP route updates based on their ROV or ASPA status

… ***and easily compute statistics!***

Users can filter BGP route updates based on their ROV or ASPA status

… ***and easily compute statistics!***

Example: Let’s analyze ASPA for downstream AS paths
using all collected updates within a 2-hour timeframe

Users can filter BGP route updates based on their ROV or ASPA status … ***and easily compute statistics!***

Example: Let's analyse ASPA for downstream AS paths
using all collected updates within a 2-hour timeframe

99.3% are Unknown

There are fewer than 1000 ASPA objects so far

Statistics computed in February 2026

Users can filter BGP route updates based on their ROV or ASPA status … ***and easily compute statistics!***

Example: Let's analyse ASPA for downstream AS paths
using all collected updates within a 2-hour timeframe

99.3% are Unknown

There are fewer than 1000 ASPA objects so far

0.04% are Invalid

Potential route leaks!

Statistics computed in February 2026

Users can filter BGP route updates based on their ROV or ASPA status … ***and easily compute statistics!***

Example: Let's analyse ASPA for downstream AS paths
using all collected updates within a 2-hour timeframe

99.3% are Unknown

There are fewer than 1000 ASPA objects so far

0.04% are Invalid

Potential route leaks!

0.7% are Valid

Most involve short AS paths (avg length: 2.6)

Statistics computed in February 2026

bgproutes.io provides four dashboards, and more will come soon

AS Explorer	Prefix Explorer	Vantage Points Explorer	AS Rank
General info about ASes, prefixes, IXPs	“Giant” looking glass	List all VPs and provide statistics	Rank ASes based on various metrics

bgproutes.io

https://bgproutes.io

You can contribute data

Current sponsors:

Follow us on LinkedIn, Discord,
or reach out at contact@bgproutes.io