RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Learning to use RIPE NCC's Tools for Network Operators

Tutorial

9 October 2019 - LACNIC32 / LACNOG 2019 - Panama

Before starting...

- Have you filled out the attendance form?
  - You're helping us a lot into getting to know our attendees!

Form

Troubleshooting

How to find information on what happened

Monitoring

Create measurements to detect changes

Integration with your tools

Make use of the created measurements to generate alarms

Troubleshooting

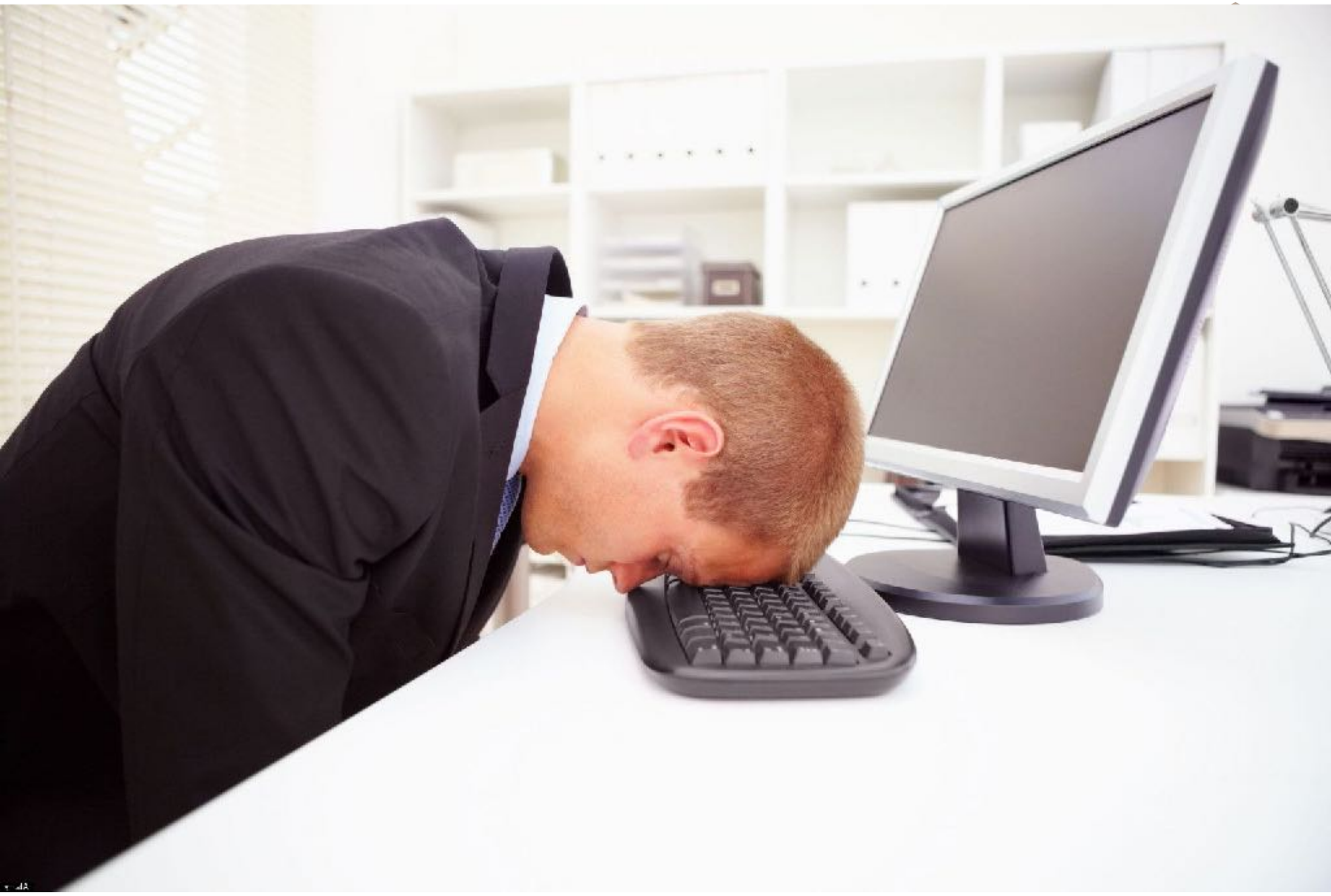
Section 1

What happened?

Exercise 1

- You work in a NOC of a big company with:
  - ASN **4725**
  - Prefix: **202.33.0.0/16**
- On the **23/09/2019**, right after 8:30 AM there are 100 emails complaining about connectivity problems to your web site in the IP **202.33.14.5**

- The **DNS** admin checked that the service is **OK**
- The **Web** admin checked that the service is also **OK**



- What can you do?

- Go to https://stat.ripe.net and try to find out

- Any change in the announcement of your prefix?
- Any other ASN announcing (part of) your addresses?

- Can you find information about the “attacker”?

Conclusions

Another AS hijacked part of my address space

- Different actions to be taken:
  1. Contact the AS announcing my addresses
  2. Announce same/more specifics too
  3. Move the service to another “safe” IP
  4. Create Monitoring measurements

RIPEstat

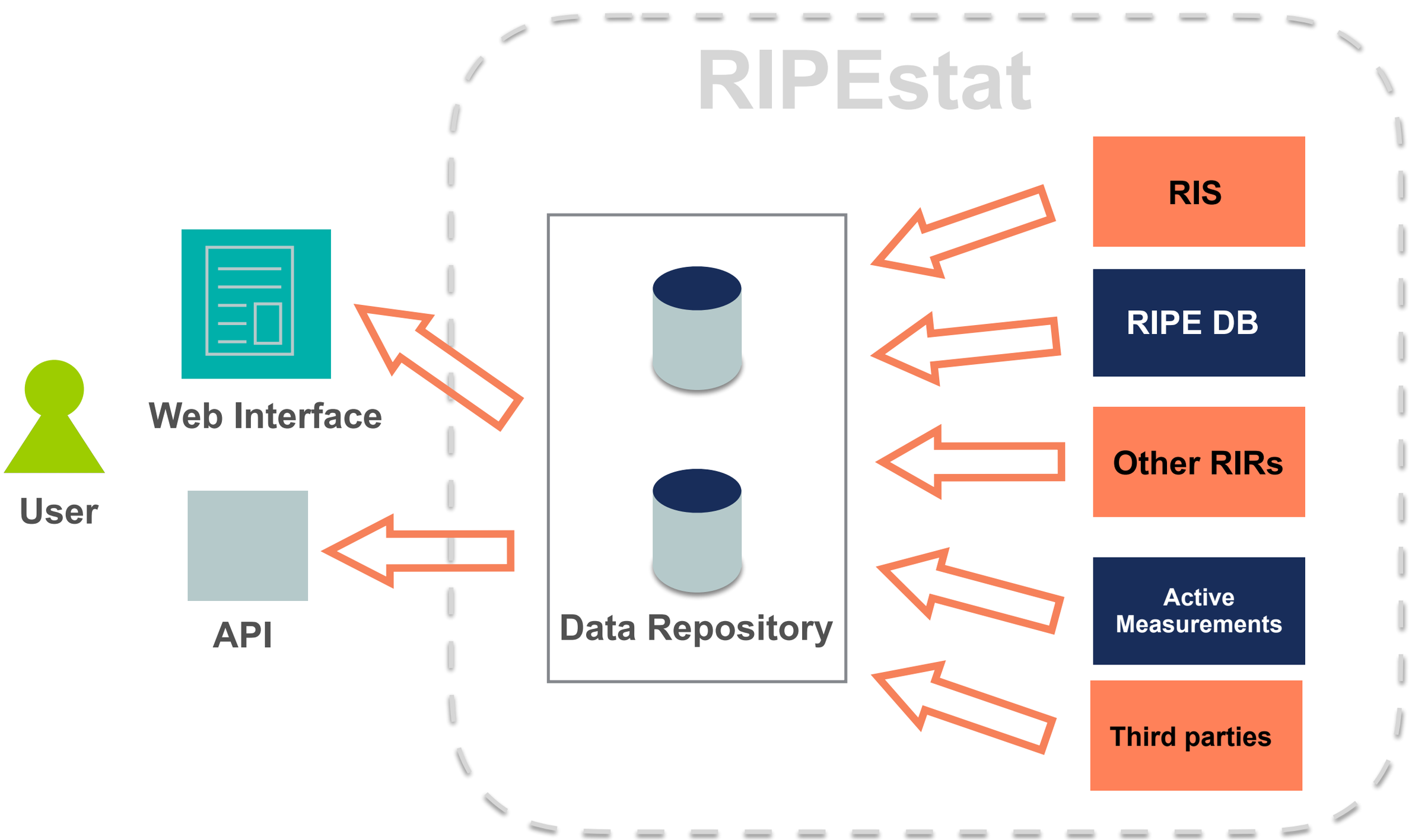
Section 1

What is RIPEstat?

One interface for Internet data and statistics

“One-stop shop”

RIPE NCC
RIPEstat


RIPEstat
RIS
RIPE DB
Other RIRs
Active Measurements
Third parties
Web Interface
API
User
Data Repository

What data? What sources?

- RIPE Database
- Other RIR data
- BGP routing data (RIS)
- Active measurements (RIPE Atlas, DNSMON)
- Geolocation (third party)
- Blacklist data (third party)
- More…

RIS - Routing Information Service

- RIPE NCC collecting BGP information since 1999
  - Raw data: ris.ripe.net

- 20+ route collectors
  900+ peers

- RIS data visualised in:
  - RIPEstat
  - RIS Live (ris-live.ripe.net)

Why use RIPEstat?

Historical Information

Info about your/other's network

Easy to share

Easy to embed

Several interfaces: Web and API

Query Types

- IPv4 and IPv6 address/prefix
- ASN
- Hostname
- Country code

RIPEstat Interfaces

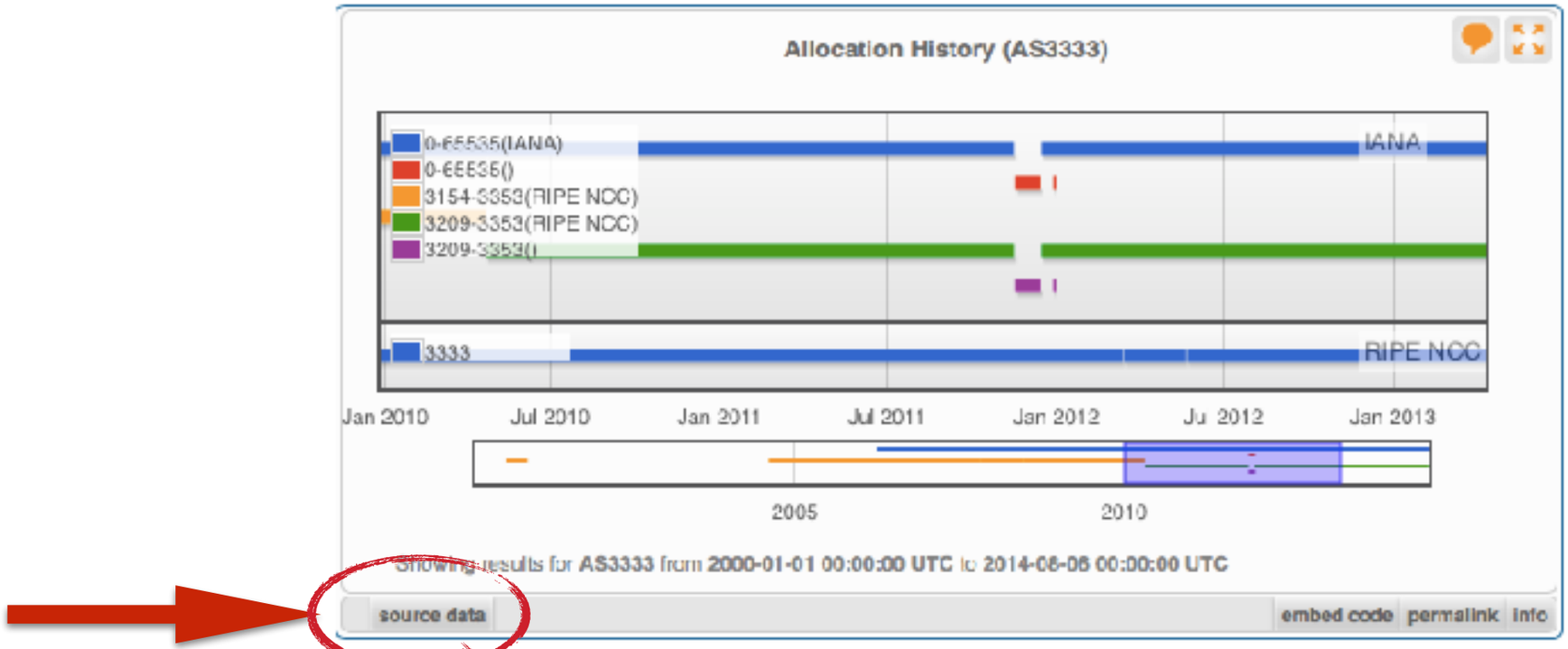
- Web interface

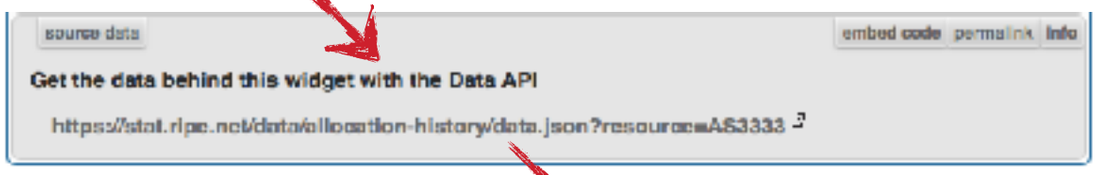
  https://stat.ripe.net

- RIPEstat widget API

- RIPEstat data API

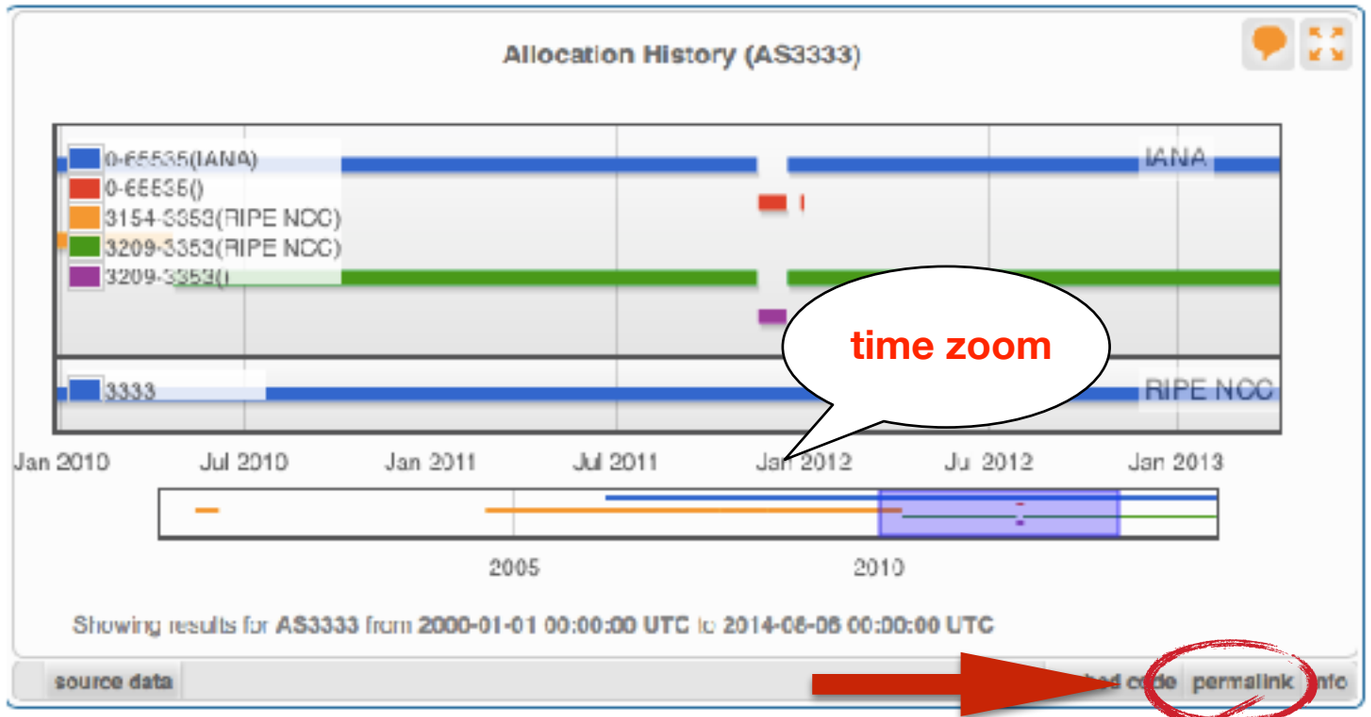
Get the data behind the widget!





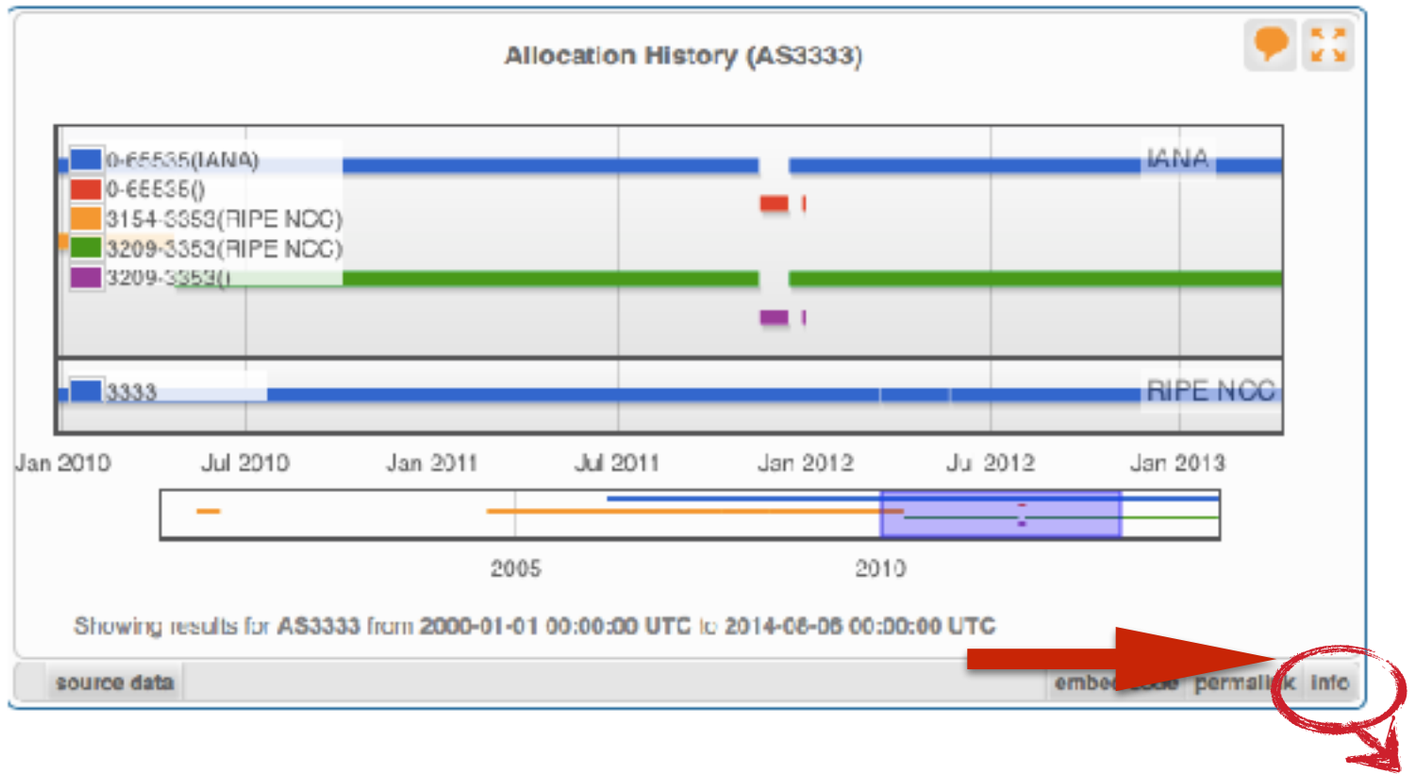

```
{
    "cached": true,
    "data": {
        "query_endtime": "2014-08-06T00:00:00",
        "query_starttime": "2000-01-01T00:00:00",
        "resource": "3333",
        "results": {
            "IANA": [
                {
                    'resource': '0-65535',
                    'status': "IANA",
                    'timelines': [
                        {
                            'endtime': "2007-10-11T00:00:00',
                            'starttime': '2007-10-11T00:00:00'
                        },
                        {
                            'endtime': "2008-11-03T00:00:00',
                            'starttime': '2007-10-27T00:00:00'
                        },
```

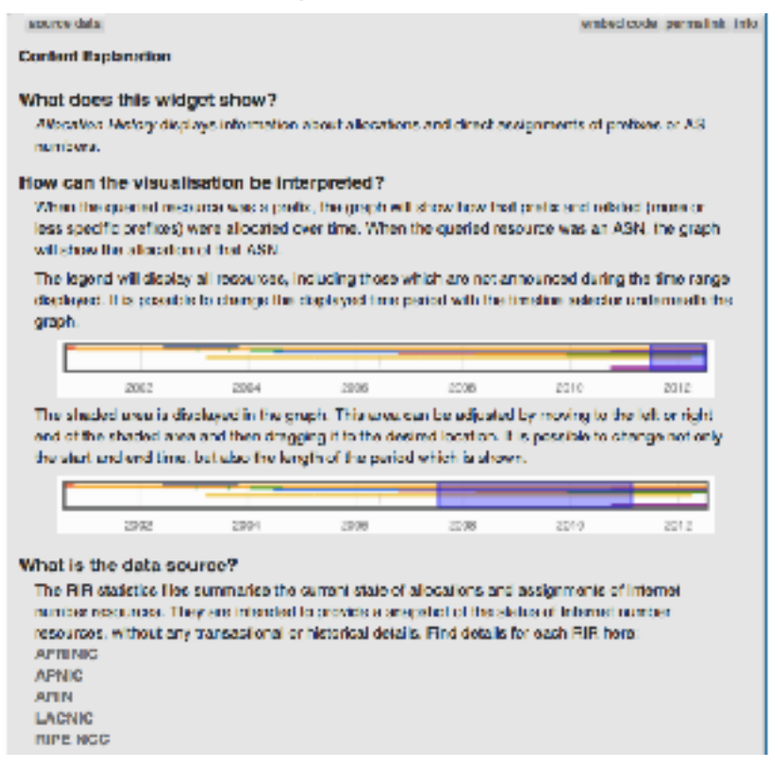
Shareable results URL



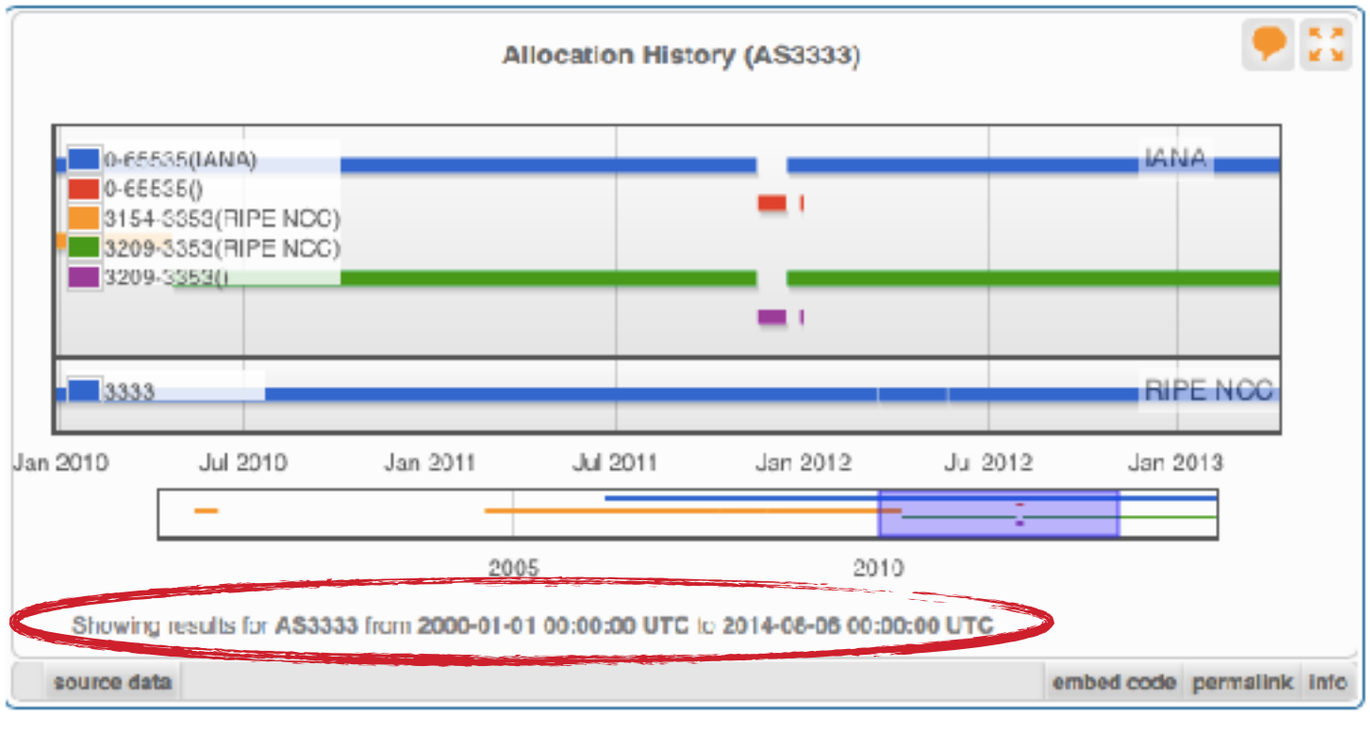
- Immutable shareable URL for each result!
- URL includes:
  - **widget** + queried **resource**
  - for some widgets: **settings**, **zoom**, **time period**

Where’s the data from?





Freshness and timescale of the data



- Timestamp and time period
- Different widgets = different update frequency
- Adjustable usually
  - Limits: different maximum granularities

Widgets List

https://stat.ripe.net/widget/list

RIPEstat Widgets

This is a complete list of all of the widgets that RIPEstat offers. Each of these widgets can be accessed using the links below.

When you view a widget you can also get code for **embedding** it in your own pages. The full procedure for embedding and configuring widgets is described in the Widget API Documentation.

Show 25 entries Search:

Title (show slug)	Example	Prefix	IP address	ASN	Hostname	Country code
Abuse Contact Finder		✓	✓	✓		
Address Space Hierarchy		✓	✓			
Address Space Usage		✓	✓			
Allocation History		✓	✓	✓		
Announced Prefixes				✓		
Announced Prefixes (Inrdb)				✓		
Announced Prefixes (Ursa)				✓		
AS Overview				✓		
AS Path Length				✓		
AS Routing Consistency				✓		
ASN Neighbours				✓		
ASN Neighbours History				✓		

BGPlay

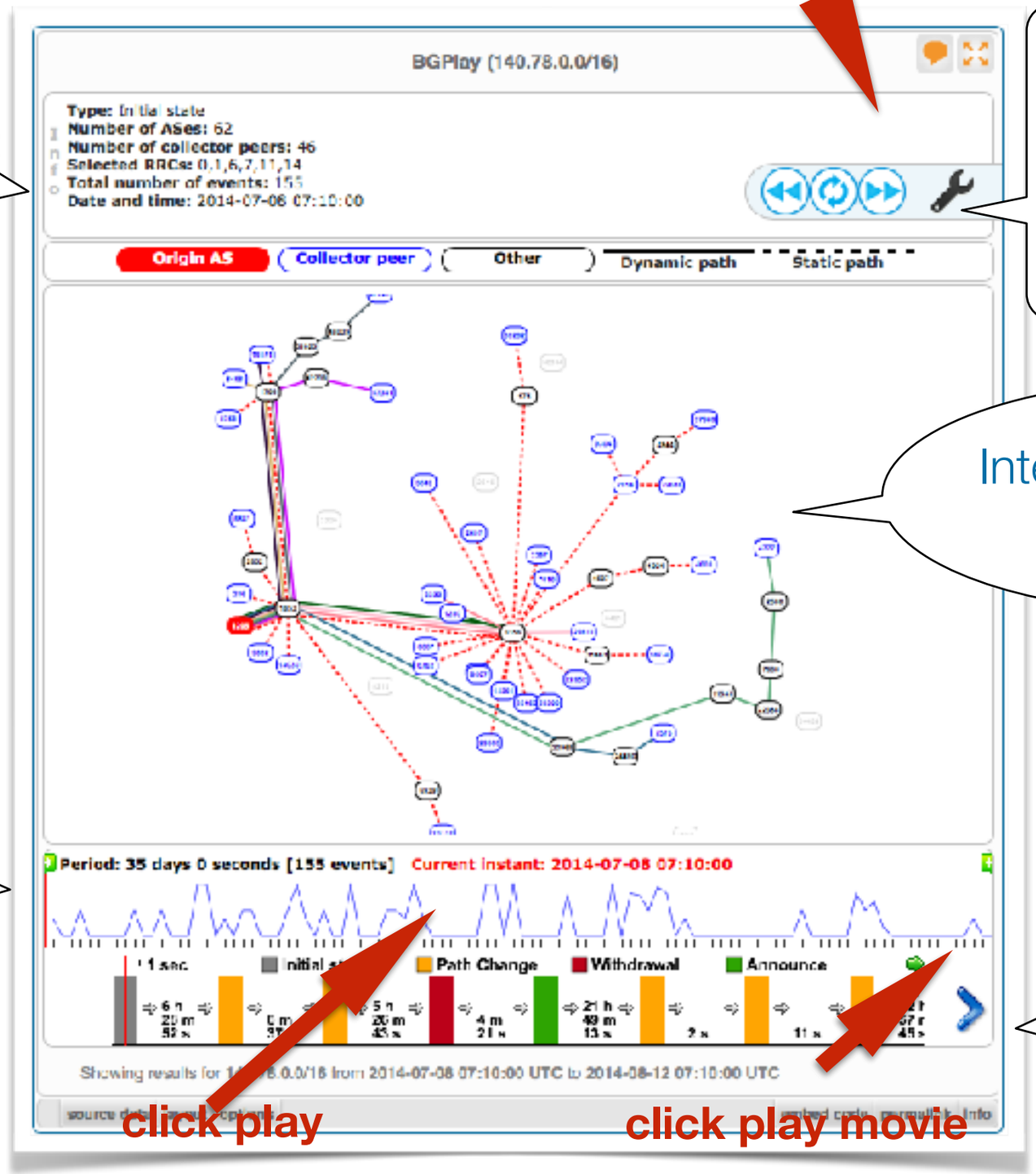
- See how your network is routed
  - Announcements
  - Withdrawals
  - Path changes
- Shows routing history
  - Animated graphic
  - Highly interactive

https://stat.ripe.net/widget/bgplay

BGPlay

BGP event, ASN or ASN path details

click play



Control panel:

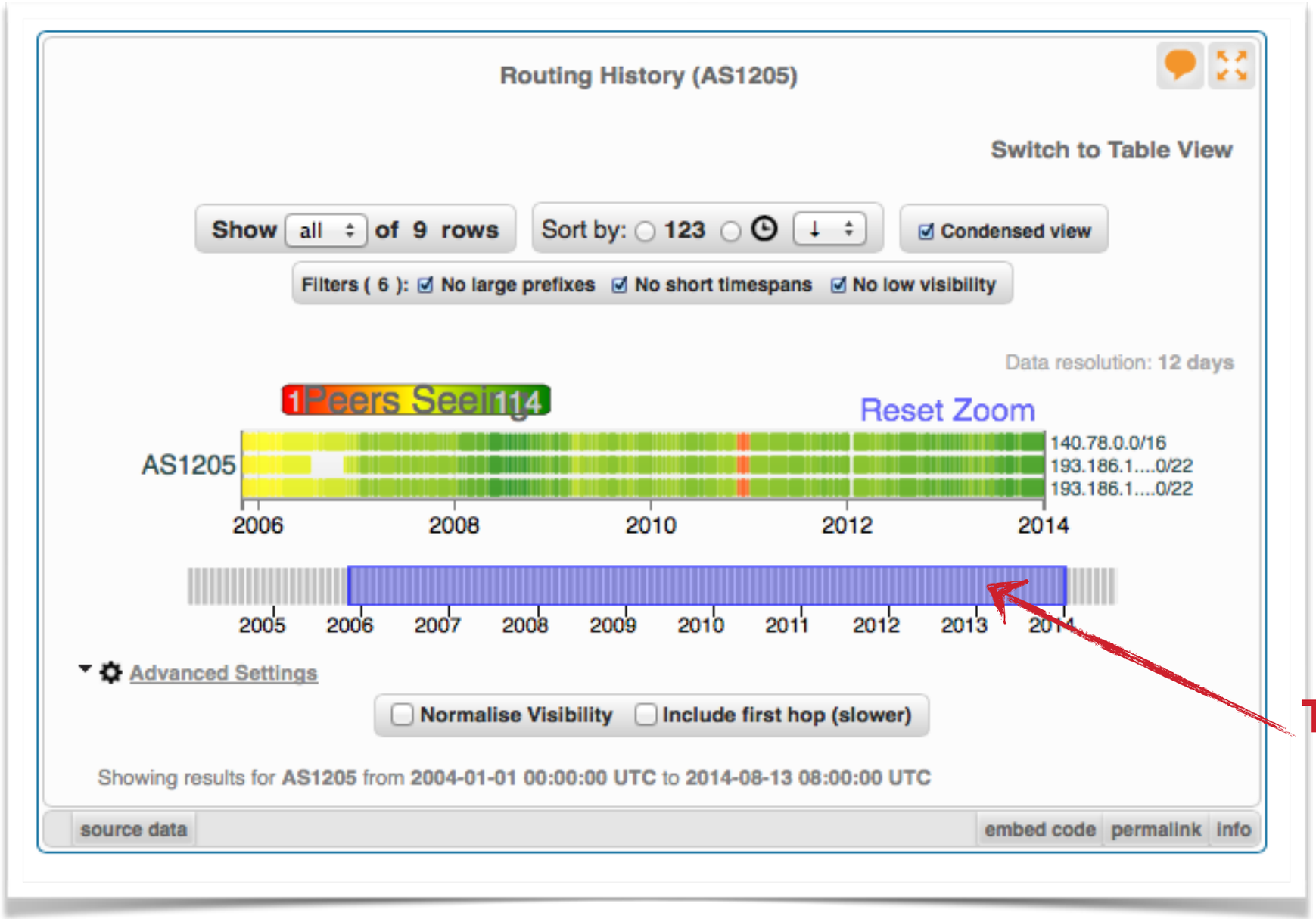
- Covered time period
- RRC selection

Interactive animated graph

Control timeline

Detailed timeline with events

History of Prefixes Announced by ASN



Time scale selection

Monitoring

Section 2

How to detect problems

Exercise 2

After the incident, **you move the service to another IP**: **202.33.33.53**

You decide to use RIPE Atlas to monitor the service to detect problems:

- Check delays from 5-10 places in Japan (ping)
- Check paths from 5-10 places in Japan (traceroute)

You need…

- **RIPE NCC Access Account** (access.ripe.net)
  - Create one if you don't have it yet

- **Credits**: Every measurement has a cost in credits
  - Why? Fairness and avoid overload
  - How to earn credits?
  - Hosting a probe / anchor, Being an RIPE NCC member (LIR), Being RIPE Atlas sponsor, Transfer, **Voucher**

LACNIC 32 / LACNOG 2019 voucher

- Go to RIPE Atlas (atlas.ripe.net)
  - **Login** using your RIPE access account
  - My Atlas (left menu)
  - Credits
  - Redeem voucher

The voucher! :-) **LACNIC32PANAMA**

Exercise

- Go to https://atlas.ripe.net
- **Login** with your access account
- “Measurements, Maps and Tools”
- “Measurements” ->

- New Measurements:
  - Run for **5 days**, 5-10 probes in **Japan**, target **202.33.33.53**
    1. Create a **ping** measurement
    2. Create **traceroute** measurement

Conclusions

You have two measurements running towards your monitored service

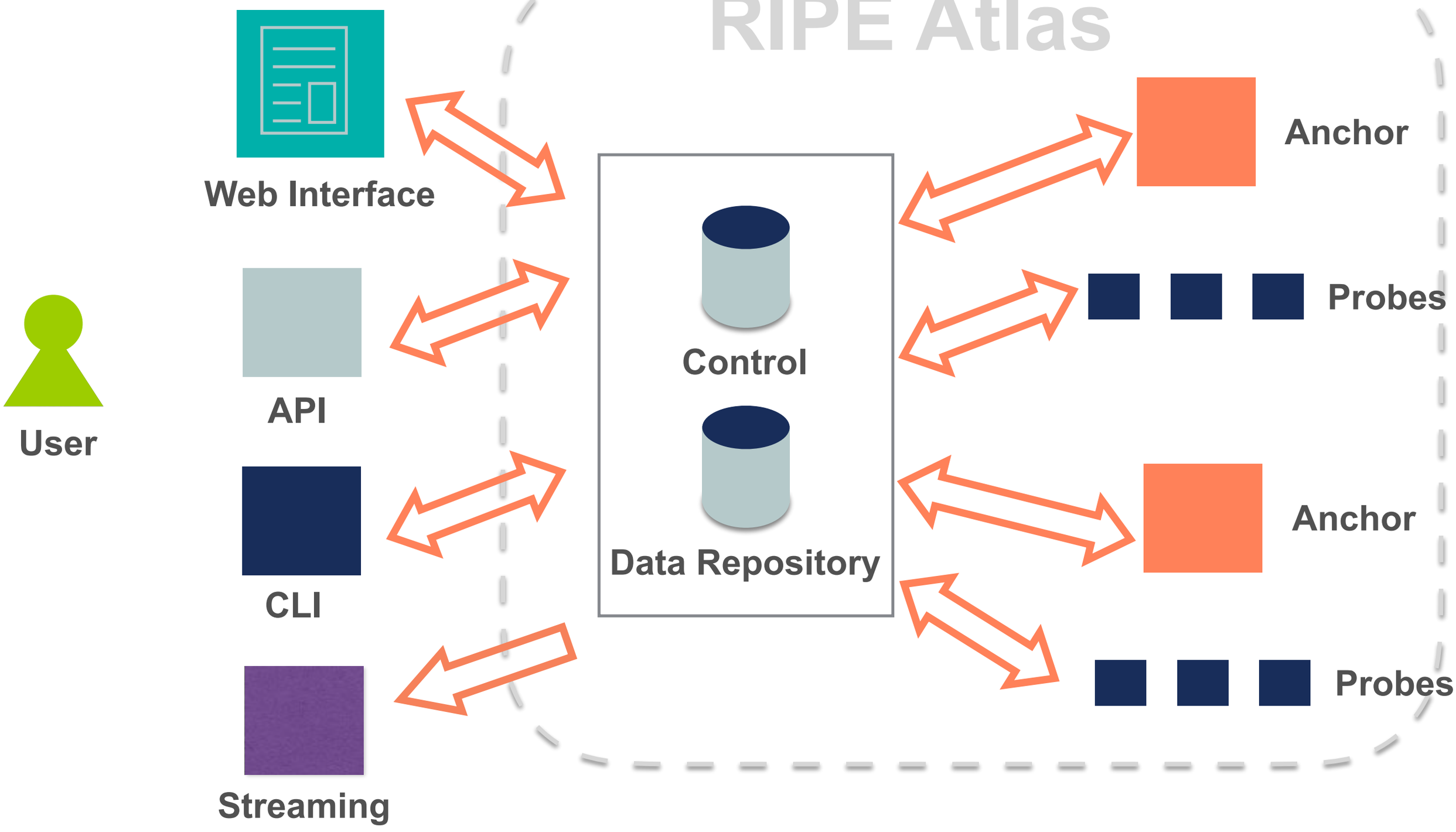
RIPE Atlas

Section 2

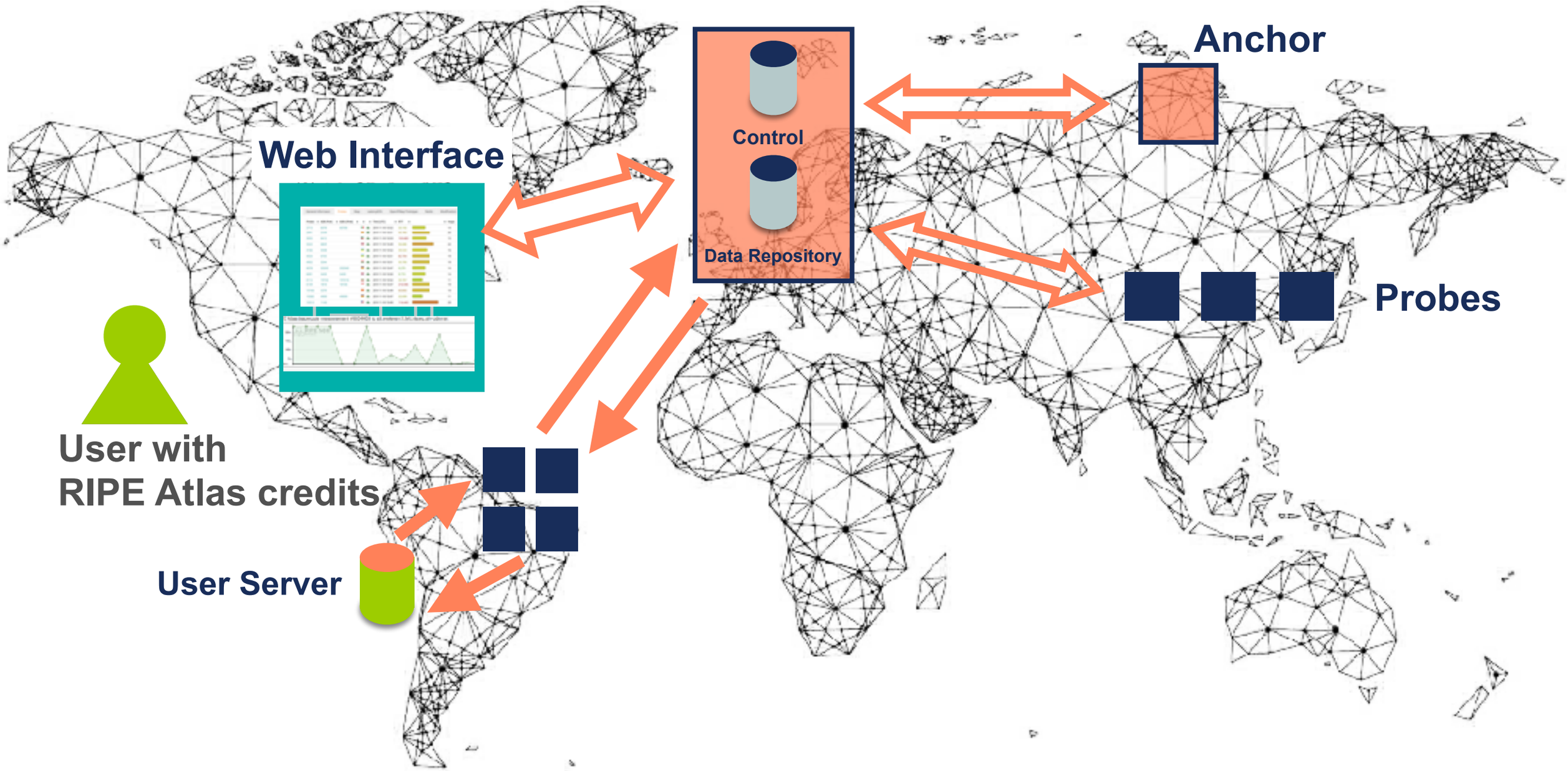
RIPE Atlas Goals

- Internet wide measurement system
  - Internet infrastructure, not all applications
- Real time & historical info
- Outbound and inbound (active) measurements
- Collaborative effort
- Open and free
- IPv4 and IPv6 capable

RIPE Atlas Overview (1)



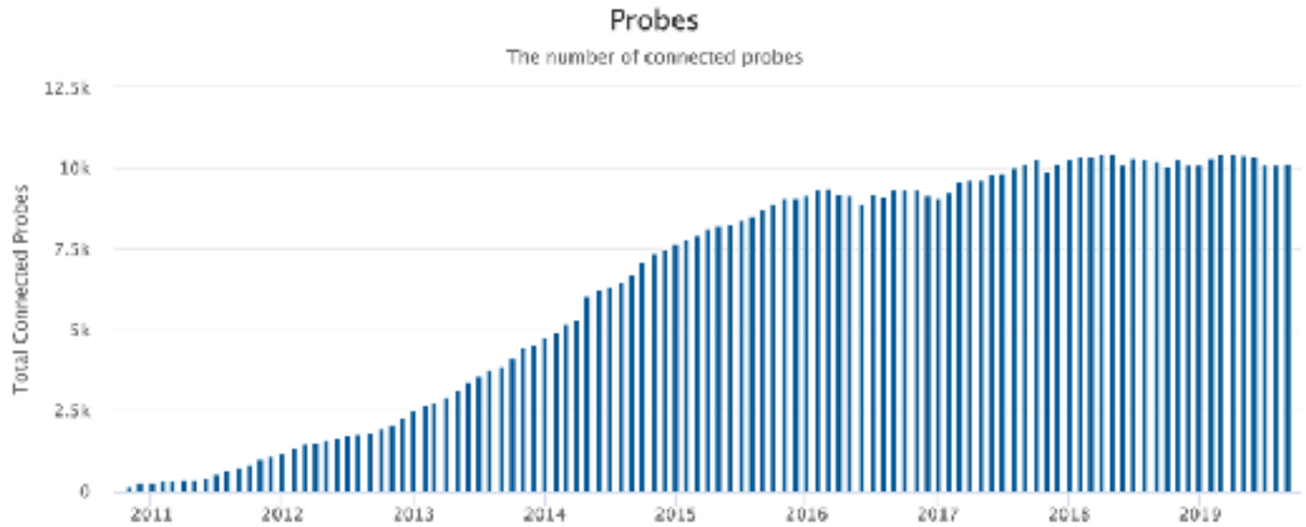
RIPE Atlas Overview (2)



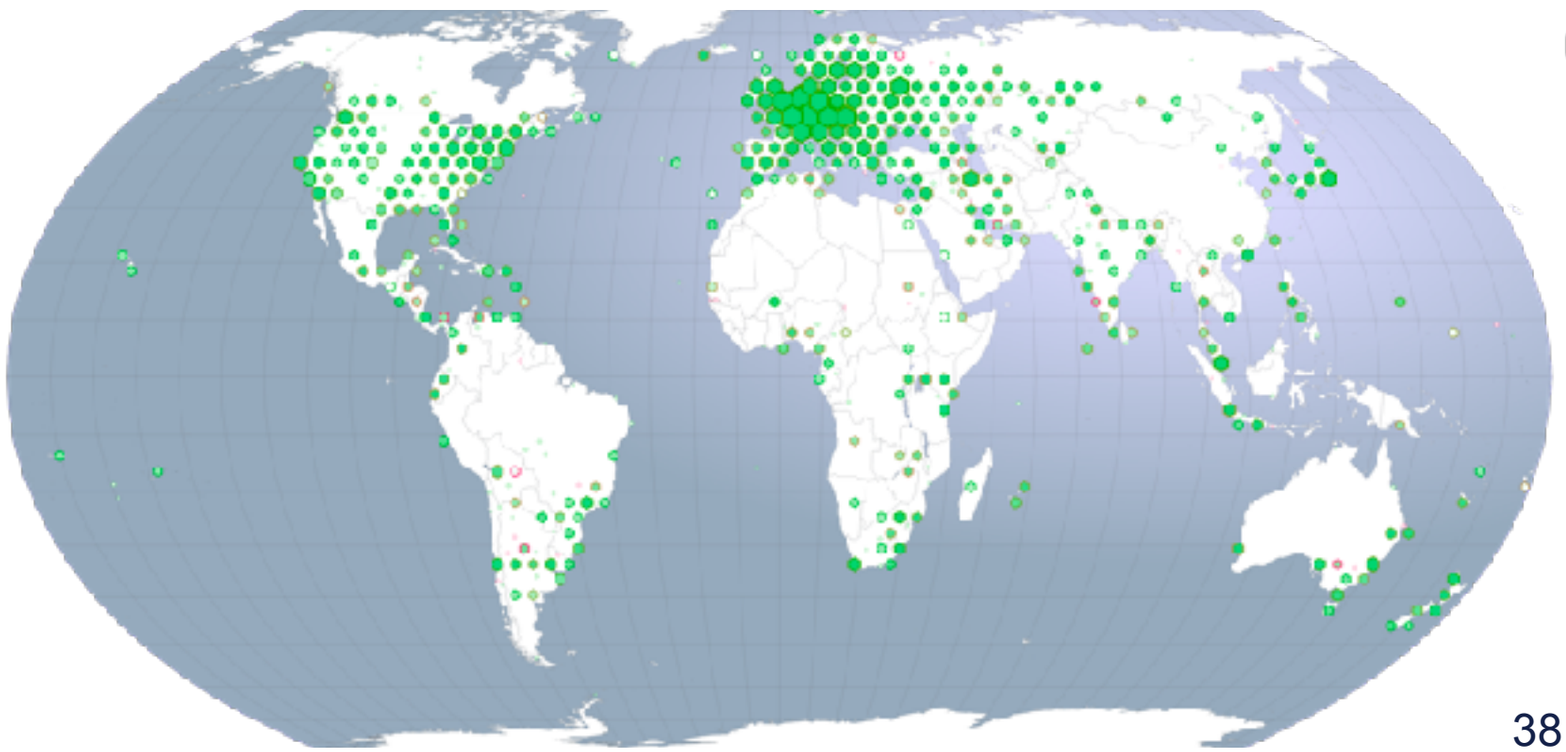
RIPE Atlas is made of…

Probes

- 10200+

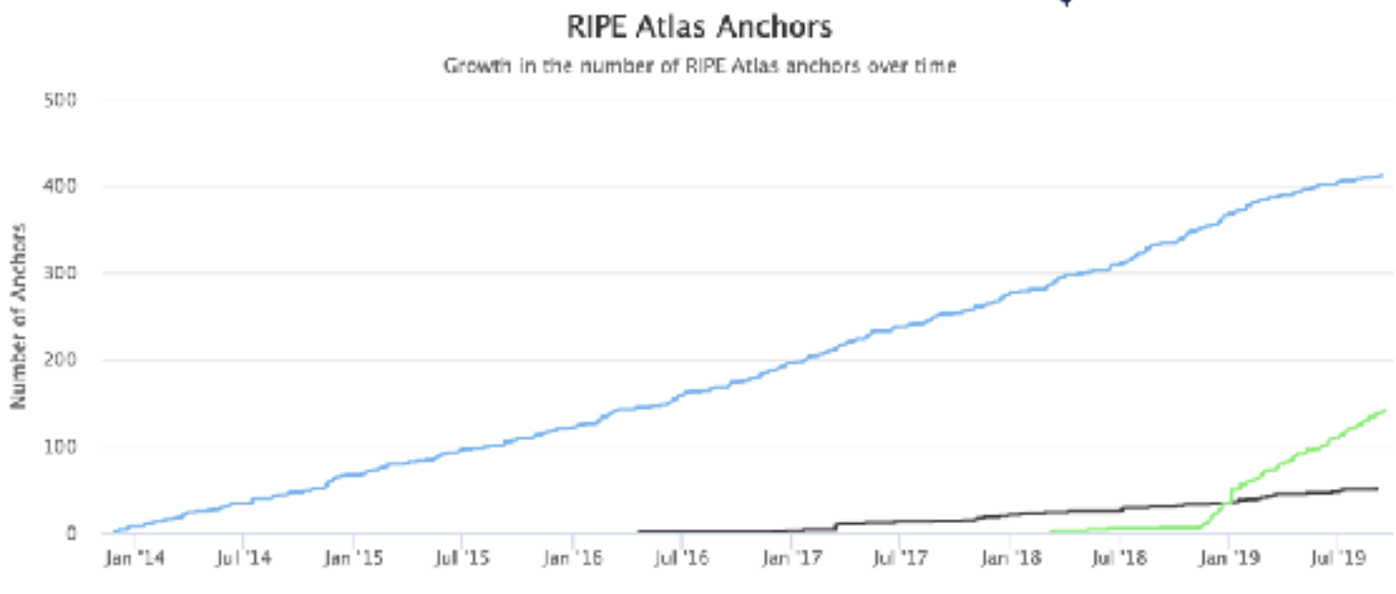


- Around the world

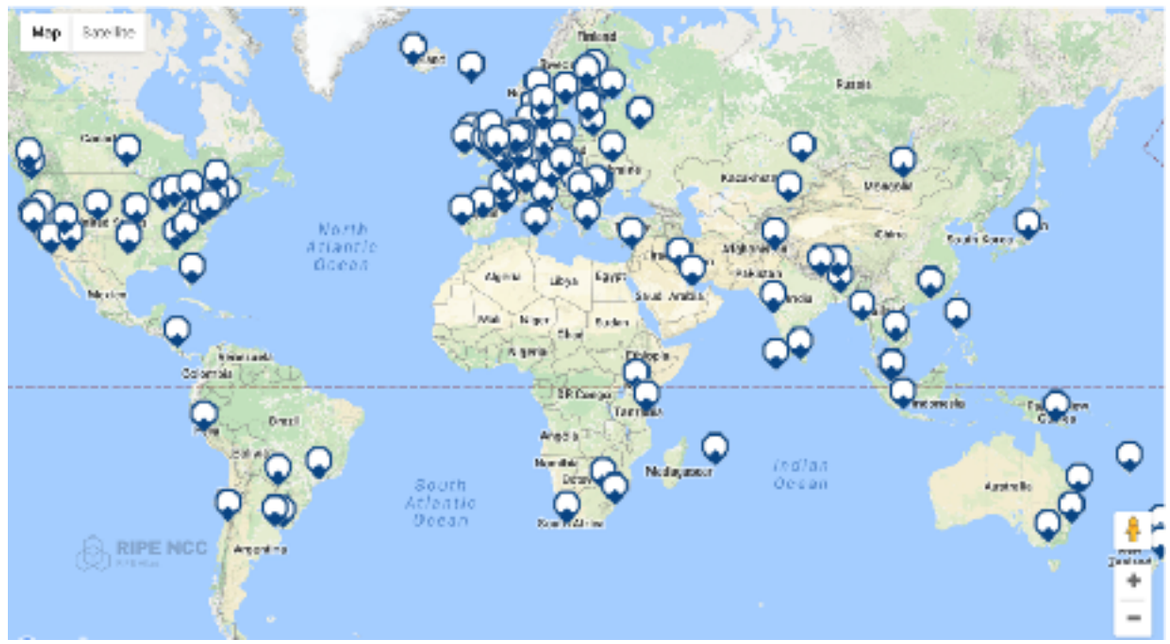


RIPE Atlas is made of…

Anchors

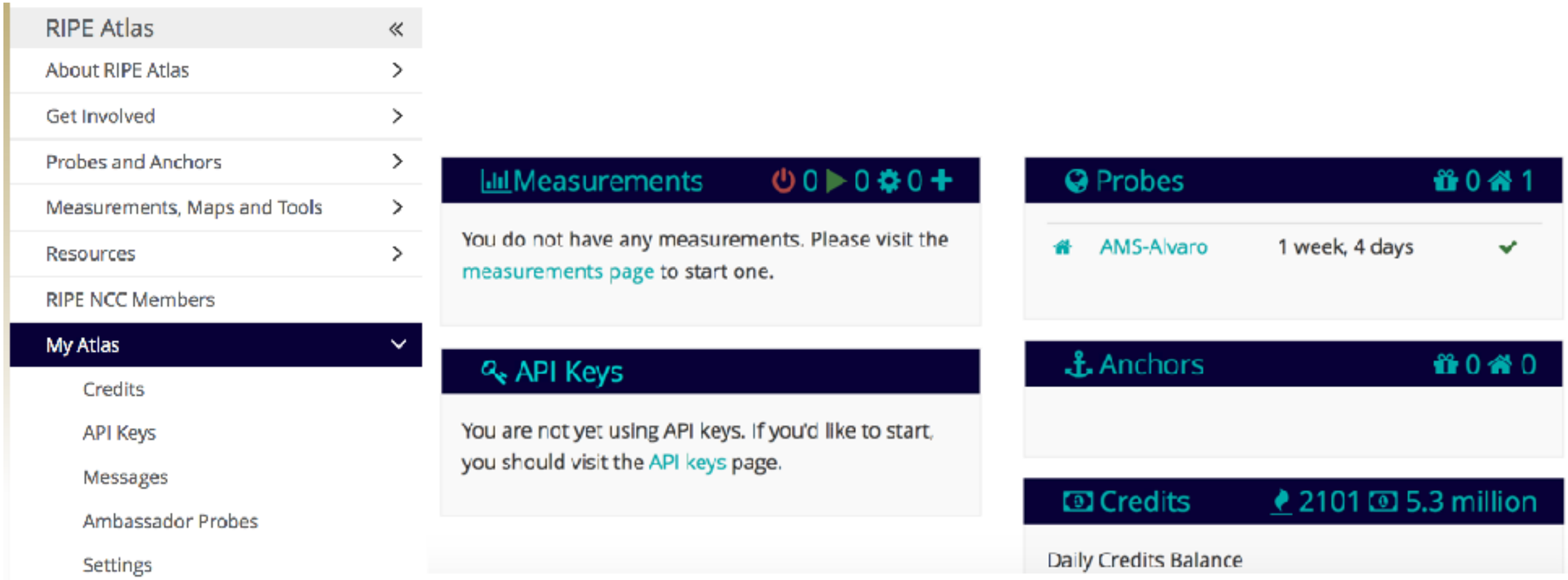


- 550+ (22 LAC)
- Around the world



RIPE Atlas is made of…

User interfaces: **Web interface / API / CLI / Streaming**



RIPE Atlas is made of…

Measurements

Measurements currently running

	Built-in	User-defined			
		Total UDM	Anchoring	DNSMON	Other
Ping	41	6829	2138	0	4691
Traceroute	45	6156	2143	887	3126
DNS	158	6026	1	3548	2477
SSL/TLS Certificate	4	374	0	0	374
NTP	0	102	0	0	102
HTTP	4	2201	2139	0	62
WiFi	0	14	0	0	14

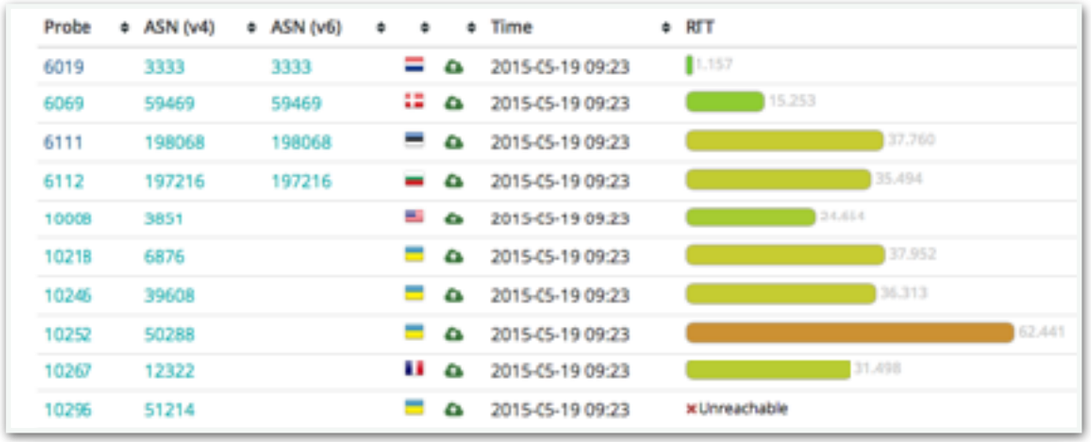
RIPE Atlas measurements

- **Built-in** global measurements towards root nameservers
  - Visualised as Internet traffic maps
- **Built-in** regional measurements towards “anchors”
- **Users** can run customised measurements
  - ping, traceroute, DNS, SSL/TLS, NTP and HTTP*

* Only towards anchors

Available visualisations: ping

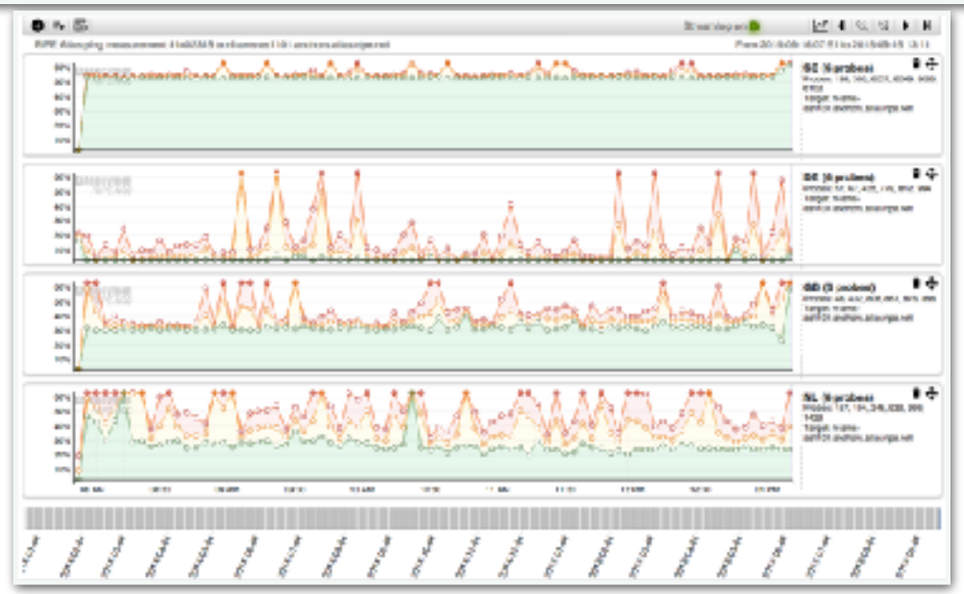
- List of probes: sortable by RTT



Probe	ASN (v4)	ASN (v6)	Time	RTT
6019	3333	3333	2015-05-19 09:23	1.157
6069	59469	59469	2015-05-19 09:23	15.253
6111	198068	198068	2015-05-19 09:23	37.760
6112	197216	197216	2015-05-19 09:23	35.494
10008	3851		2015-05-19 09:23	24.414
10218	6876		2015-05-19 09:23	37.952
10246	39608		2015-05-19 09:23	36.313
10252	50288		2015-05-19 09:23	62.441
10267	12322		2015-05-19 09:23	31.498
10296	51214		2015-05-19 09:23	✖ Unreachable

- Map: colour-coded by RTT

- LatencyMON: compare multiple latency trends



Available visualisations: traceroute

- TraceMON: network topology, latency and nodes information

- RIPE IPmap: hops geolocation on map

Security Aspects

- Probes:
  - Hardware trust material (regular server address, keys)
  - No open ports; initiate connection; NAT is okay
  - Don’t listen to local traffic
  - No passive measurements
  - Automatic FW updates
- Measurements triggered by “command servers”
  - Inverse ssh tunnels
- Source code published

Ethical Considerations

- No passive measurements (no user traffic)
- Set of measurements is limited
- HTTP measurements only to Anchors
- All data is open and available to anyone
- Barrier to entry is low/cheap
- Open APIs
- Open source code on GitHub

Integration

Section 3

Preparing the lab

- Python, pip and the ripe-atlas CLI
- curl and jq commands
- How?
  - Native installations
  - Virtualbox
  - Docker

Command-line Interface (CLI) Toolset

RIPE Atlas CLI

- Open source
  - RIPE NCC led community contribution
- Documentation
  - https://ripe-atlas-tools.readthedocs.org/
- Source:
  - https://github.com/RIPE-NCC/ripe-atlas-tools/

Install RIPE Atlas tools

- Requirements for this section
  - ripe-atlas-tools running

    `ripe-atlas` command

- Native options
  - OSX

    ```
    sudo easy_install pip
    sudo pip install ripe-atlas-tools
    ```

  - Linux

    Available from many package repositories
    ...or same as in OSX

Install RIPE Atlas tools

- Windows native (*experimental*)
  - Windows 7
    https://github.com/chrisamin/ripe-atlas-tools-win32/releases/download/0.1/RipeAtlasToolsSetup.exe
  - Windows 8 and up
    https://github.com/chrisamin/ripe-atlas-tools-win32/releases/download/v0.1.1/RipeAtlasToolsSetup.exe

Install RIPE Atlas tools

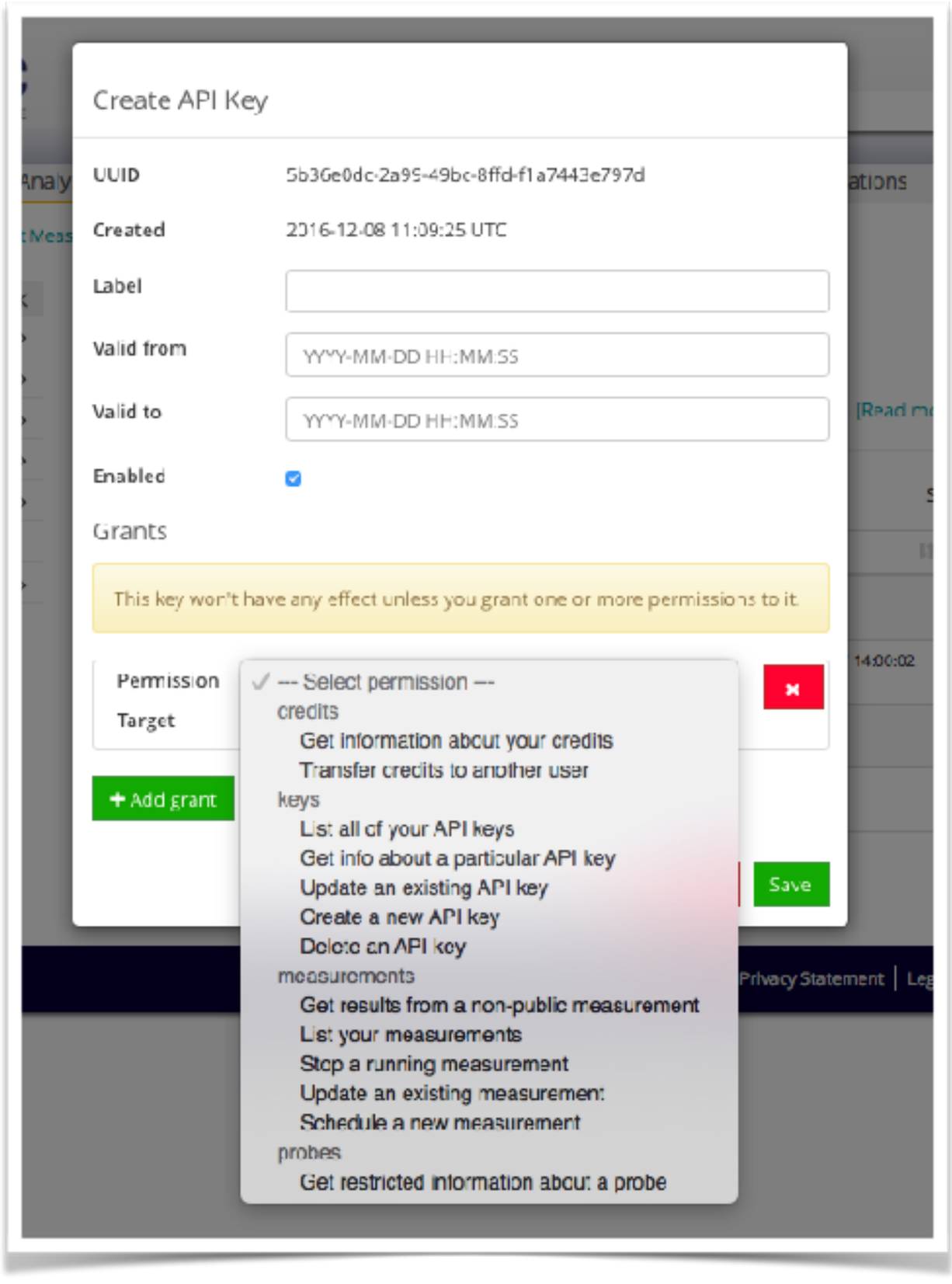
- Feel like trying Docker?
  - **`docker build -t workshop-mnt github.com/aguformoso/workshop-mnt`**
  - **`docker run -ti workshop-mnt bash`**
    - Opens a shell into your container. Don't log out from it.
- Virtualbox installation?
  - user: **participante**
  - password: **participante**

Create API Key

- Go to MyAtlas
- Click on “Create an API Key”
- Choose “permission”: “schedule new measurement”
- Careful! Time is UTC!
- Give it a label

API key creation dialogue box

- My Atlas > Create an API Key



Configure RIPE Atlas CLI

- Create an API Key
  - https://atlas.ripe.net/keys/
- Configure your CLI
  - `ripe-atlas configure --editor`
  - `ripe-atlas configure --set authorisation.create=<MY_API_KEY>`

The `--help` flag

- Navigate through the available options

```
$ ripe-atlas --help

Usage: ripe-atlas <command> [arguments]

Commands:
   alias                Manage measurements' and probes' aliases
   configure            Adjust or initialize configuration options
   go                   Visit the web page for a specific measurement
   measure              Create a measurement and wait for the results
   measurement-info     Return the meta data for one measurement
   measurement-search   Fetch and print measurements fulfilling
specified criteria based on given filters
   probe-info           Return the meta data for one probe
```

The `--help` flag

- Get sub-command-specific information

```
$ ripe-atlas measure --help

Usage: ripe-atlas measure <type> [arguments]

Types:
   dns
   http
   ntp
   ping
   sslcert
   traceroute

For extended options for a specific measurement type, try ripe-
atlas measure <type> –help.
```

Fetch an existing measurement

- Fetch the measurement created previously
  - `ripe-atlas report <MEASUREMENT_ID>`
  - `ripe-atlas report 1001`

Search for probes

- Search for useful probes

```
-ripe-atlas probe-search --asn 3333

-ripe-atlas probe-search --country pa

-ripe-atlas probe-search --country pa --
 status 1
```

- IPv6? Use the help menu
- Use probe tags! https://atlas.ripe.net/docs/probe-tags/

Search for probes

- Show specific fields
  - `ripe-atlas probe-search --asn 3333 --field asn_v6 --field country --field description --field status`

- Check the `--ids-only` flag

Create a measurement

- `ripe-atlas measure ping`

  `--target wikipedia.org`

  `--interval 60` (default one-off)

- `ripe-atlas measure traceroute`

  `--target wikipedia.org`

  `--from-area`

  `--from-prefix`

  `--from-country`

  `--traceroute-show-asns`

Better Reporting

- Your measurement might involve several world-wide distributed probes
  - Difficult to debug
  - Report on specific probes

```
ripe-atlas report --probes 1,3 1001
```

  - Or group by country

```
ripe-atlas report --aggregate-by country
1001
```

  - Bonus points: discover what measurement 1001 is by using the CLI only

Status Checks

Status Checks

- Documentation (Manual)
  - atlas.ripe.net/docs/api/v2/manual/measurements/status-checks.html
- Requirements for this section
  - **`curl`** command
  - **`jq`** command

```
curl <URL> | jq <filters>
```

Status Checks

- Metrics
  - **Availability** (packet loss)
  - **Response time** (latency)
- Status Check endpoint
  - atlas.ripe.net/api/v2/measurements/<MEASUREMENT_ID>/status-check
  - Use the <MEASUREMENT_ID> created previously

Status Checks

- Check the HTTP response headers
- This allows to HEAD the service

```
$ curl -v "https://atlas.ripe.net/api/v2/measurements/23018851/
status-check"

HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 03 Oct 2019 07:32:19 GMT
Content-Type: application/json
Content-Length: 133
Connection: keep-alive
X-RIPE-Atlas-Global-Alert: 0
Allow: GET, HEAD, OPTIONS
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15768000
```

HTTP response header based on the alert

Status Checks

- Checking the response body

```
curl "https://atlas.ripe.net/api/v2/measurements/23018851/status-
check" | jq

{
  "global_alert": false,
  "total_alerts": 0,
  "probes": {
    "6615": {
      "alert": false,
      "last": 0.245,
      "last_packet_loss": 0,
      "source": "Probes: 6615"
    }
  }
}
```

Global alert

Per-probe alerts

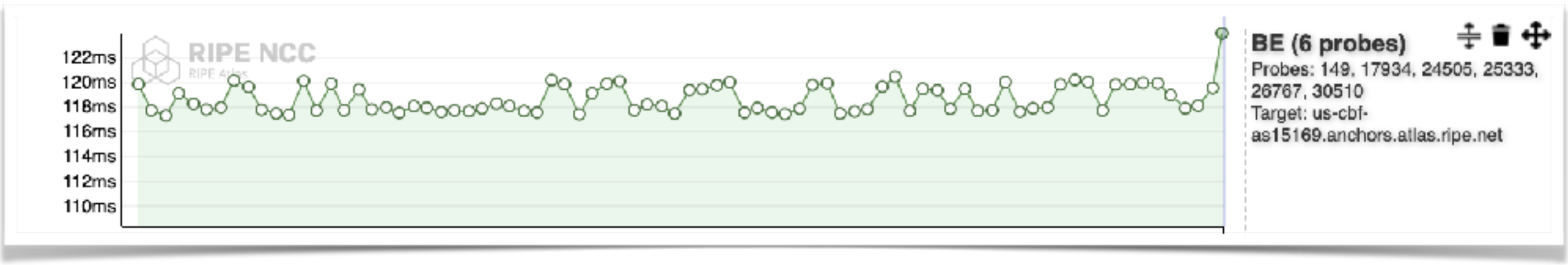
Status Checks: Your service

- <MEASUREMENT_ID>
  - atlas.ripe.net/api/v2/measurements/**<MEASUREMENT_ID>**/status-check

- Any connectivity loss?
- Any anomalies in RTT?
- Any alerts?
- Global alert?

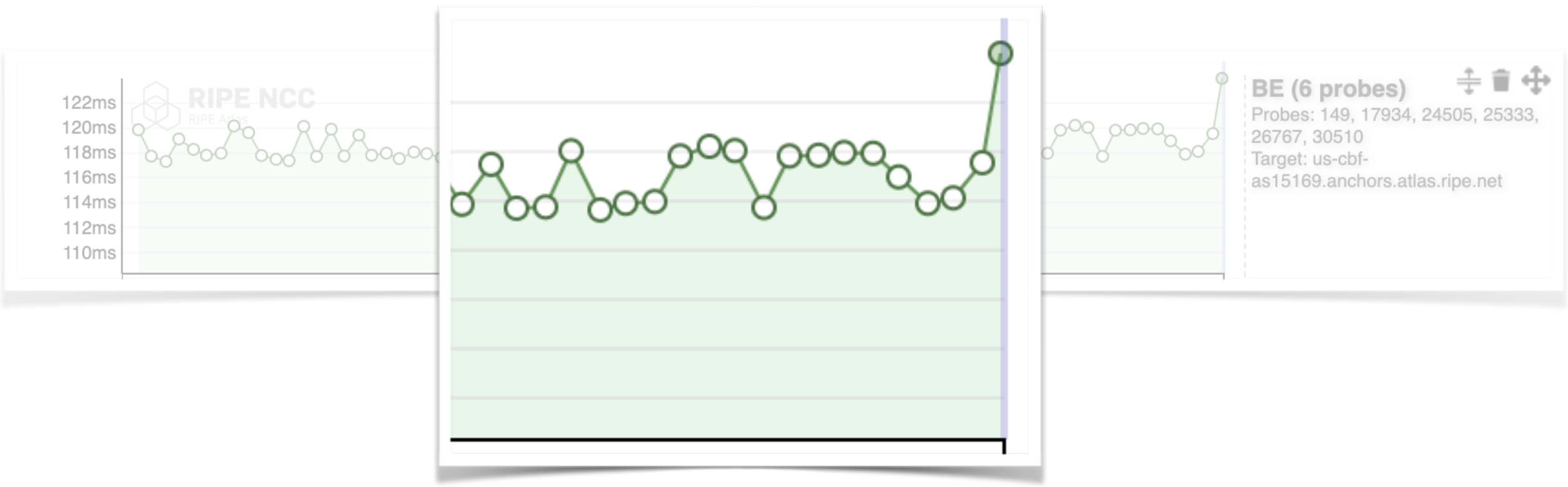
Status Checks: Fine-tuning

- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`



Status Checks: Fine-tuning

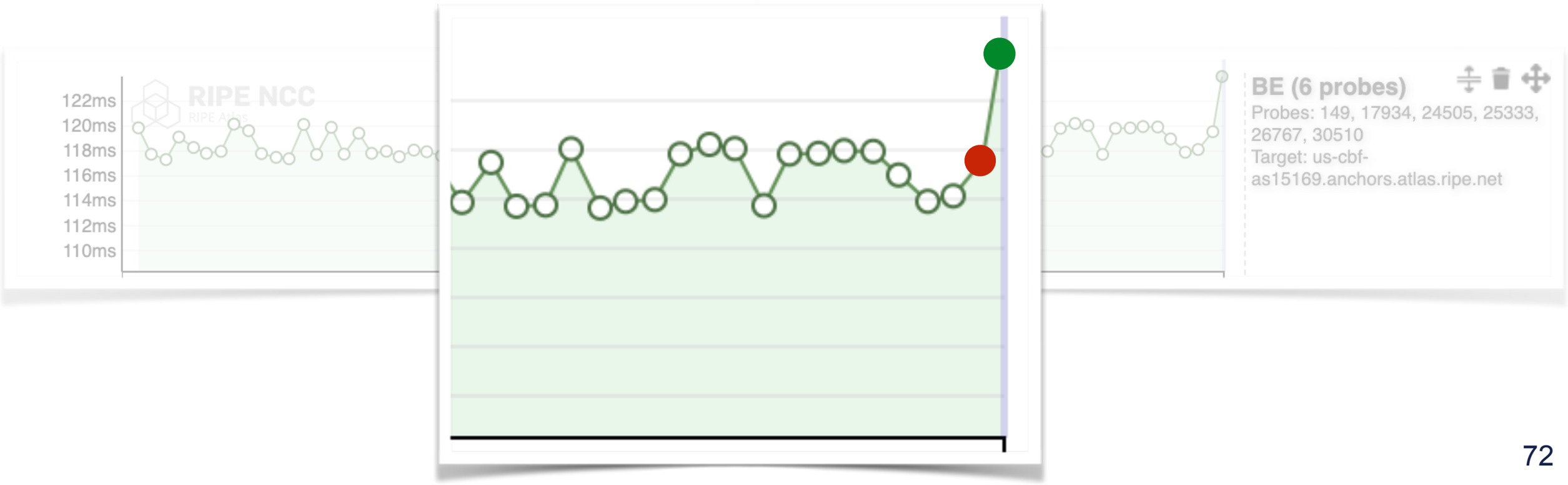
- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`



Status Checks: Fine-tuning

- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`

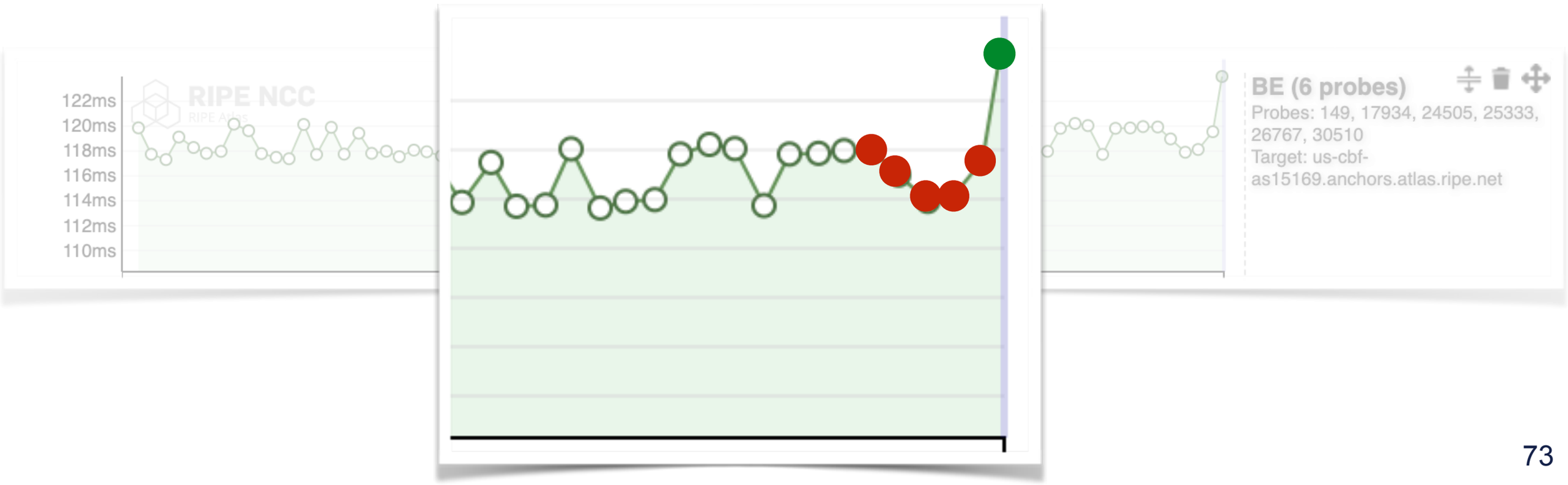
lookback=1



Status Checks: Fine-tuning

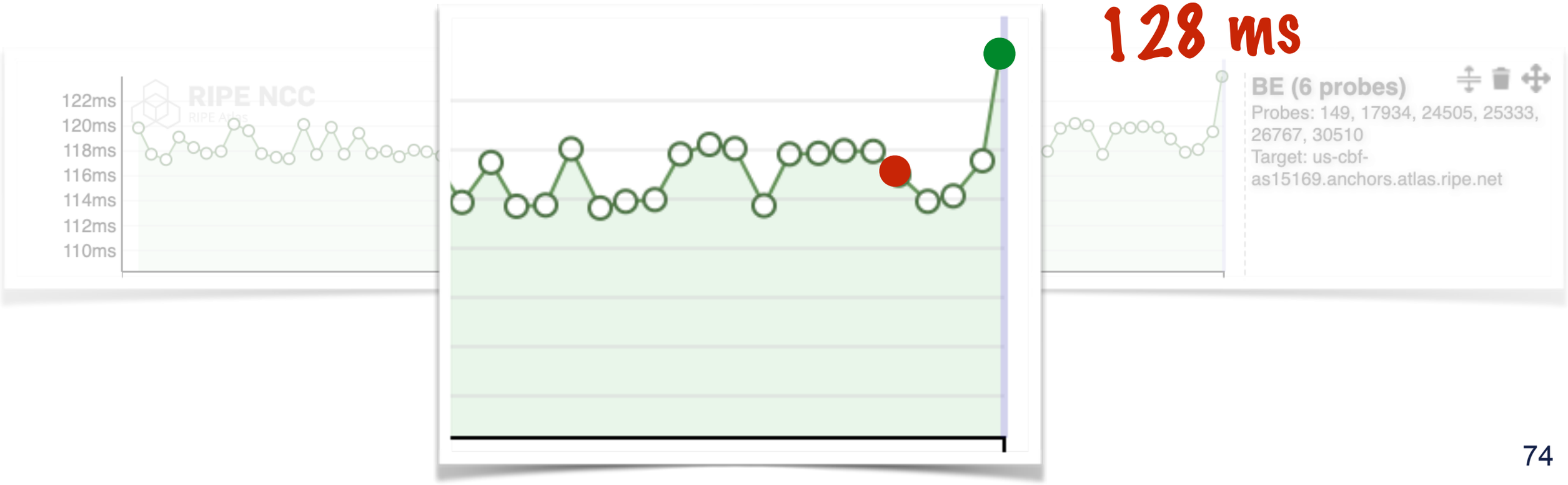
- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`

lookback=5



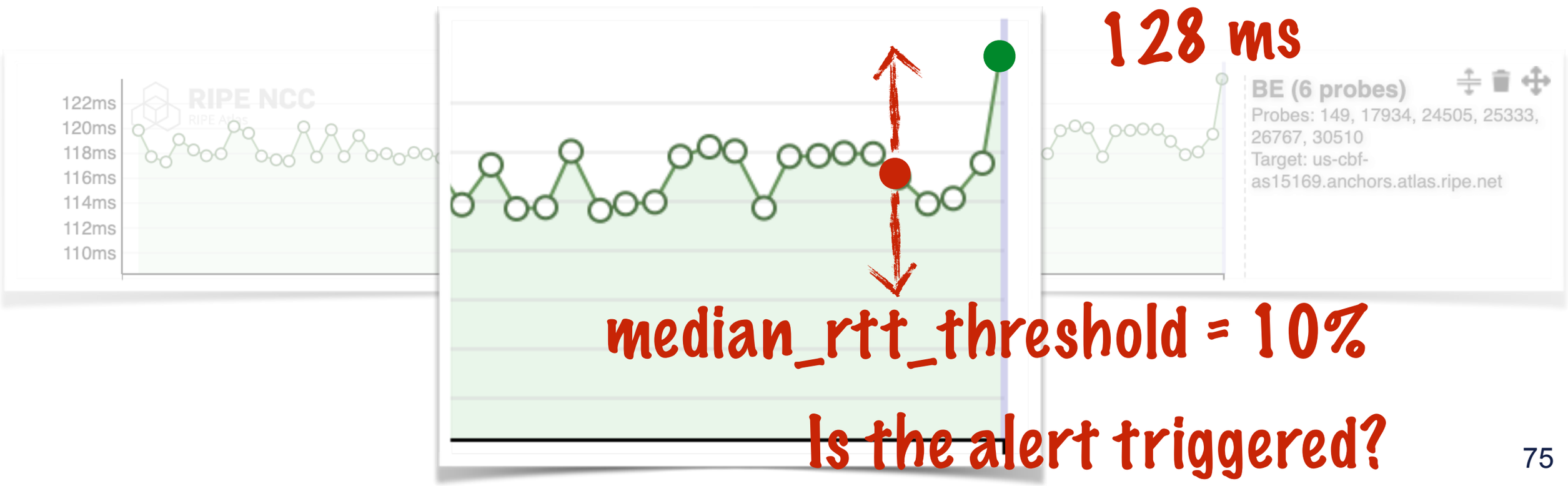
Status Checks: Fine-tuning

- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`



Status Checks: Fine-tuning

- Alerts defaults need tuning?
  - **`lookback`**
  - **`median_rtt_threshold`**
  - `max_packet_loss`
  - `permitted_total_alerts`



Status Checks: URL parameters

- lookback
- median_rtt_threshold

- atlas.ripe.net/api/v2/measurements/23018851/status-check
?**lookback**=10
&**median_rtt_threshold**=10%

Status Checks: URL parameters

- max_packet_loss
- permitted_total_alerts

- atlas.ripe.net/api/v2/measurements/23018851/status-check
?**max_packet_loss**=10
& **permitted_total_alerts** =10

Status Checks integration

- **`curl`** and **`jq`** commands (and optionally **`mail`**)

```
curl "https://atlas.ripe.net/api/v2/measurements/22937039/status-
check" |

jq 'select(.global_alert==true)' |

mail –s “My own Atlas integration" aformoso@ripe.net <<< "Something
happened with measurement 22937039!"
```

Integration with Icinga

- Examples on GitHub
- Monitoring Integration example:

```
define service {
    use generic-service
    host_name myhostname
    service_description Test_Atlas
    check_command check_http!-I atlas.ripe.net  -r
'global_alert":false' --ssl=1 -u /api/v2/measurements/1040425/
status-check/?permitted_total_alerts=1
}
```

Streaming API

RIPE Atlas Streaming API

- Receive measurement results as soon as they are sent by the probes
  - Publish + subscribe through web sockets
- There are three types of data:
  - **Measurement results**
  - Probe connection status events
  - Measurements metadata

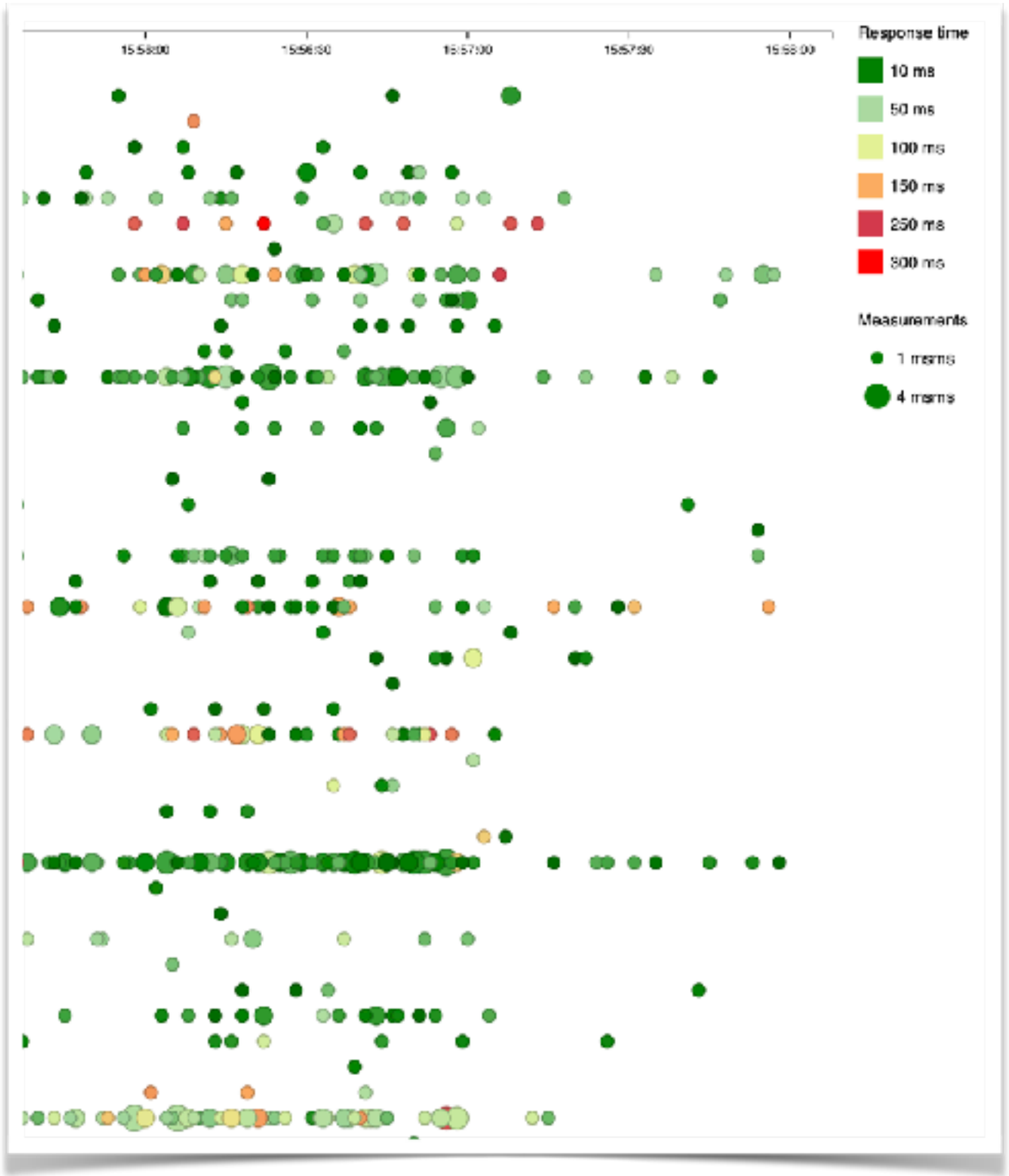
Streaming from the CLI

- `ripe-atlas stream --help`

```
ripe-atlas stream --traceroute-show-asns 5001
```

RIPE Atlas Streaming API

- Powerful visualisations
  - DNS root instances http://sg-pub.ripe.net/demo-area/atlas-stream/dns-instances.html
- Filtering and reusing measurement results
- Documentation:
  - https://atlas.ripe.net/docs/result-streaming/



RIPE Atlas Streaming API

- Requirements for this section
  - Web browser with *Developer Console*
  - In Safari

    Safari > Preferences > Advanced > *Show Develop menu in menu bar*
- Chrome or Firefox needs no reconfiguration

Steps

1. http://atlas.ripe.net/webinar/streaming01.html
2. Open the development console
3. Wait for results to arrive
4. Save the HTML file locally and edit the code
5. Open the edited html file in a browser and view results

Editing the HTML code

```
// Subscribe to results coming from
// all the probes involved in the measurement 19230504
socket.emit("atlas_subscribe",
  {
    stream_type: "result",

    type: "ping",

    greaterThan: {
      // out of 3 packets (or more),
      // choose the minimum (min)
      min: 100
    }

    destinationPrefix: "193/8",
    passThroughPrefix: "193/8" // only for traceroute
  }
);
```

EX1: Monitoring server reachability

- Back to our initial problem
- Are your pings still running?
- Edit streaming01.html
  - Use the **`msm`** parameter
  - Choose acceptable latency threshold, through the **`greaterThan`** parameter on **`min`** value
  - Notice and react when you start receiving samples

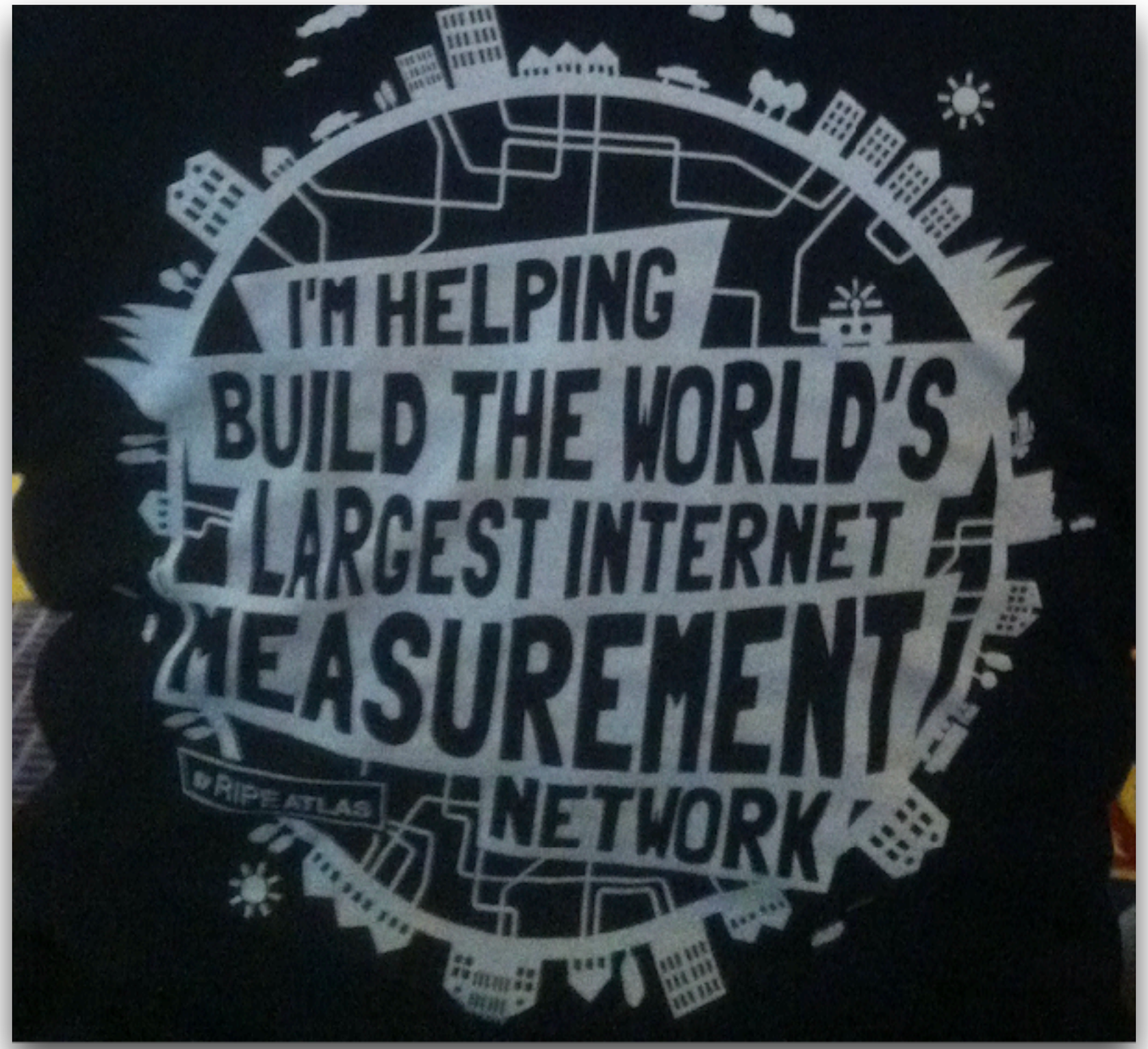
EX2: Monitoring server reachability

- Same as previous exercise, but you didn't schedule a measurement in advance
  - You don't have a measurement ID
- You want to get all the measurements reaching 167.250.112.0/24
- You can filter by **`type:"http"`**
  - Bonus point for hosting an Anchor!

```
destinationPrefix: '167.250.112.0/24'
```

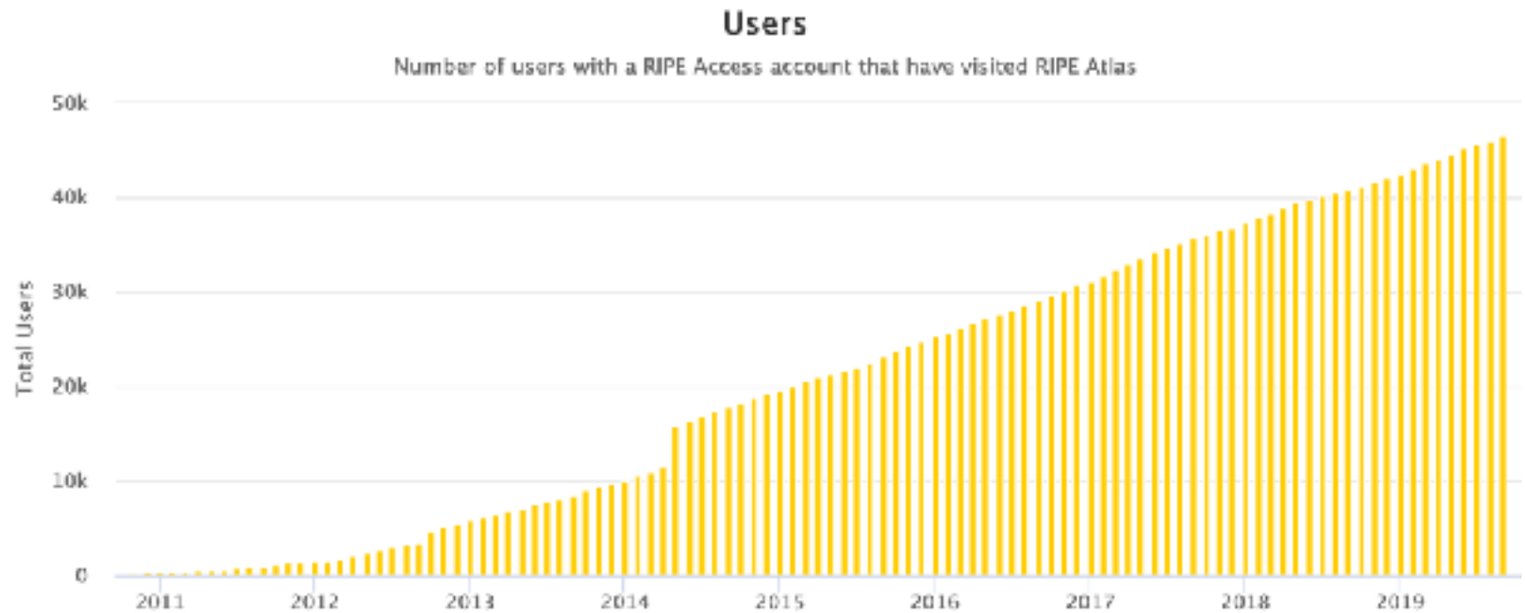
Wrapping up

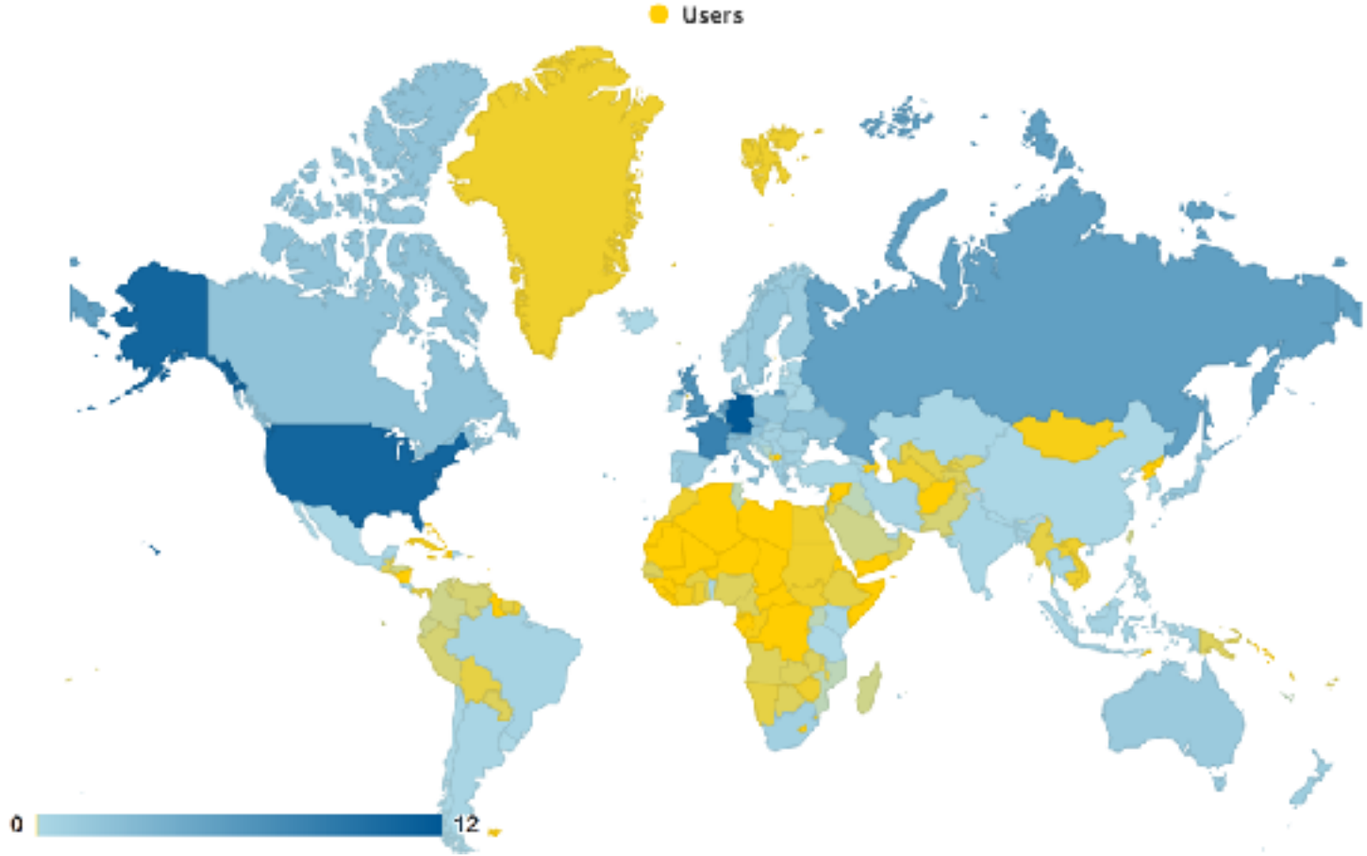
- Command Line Interface
  - Search for probes
  - Create measurements! *(stop those you don't need any longer)*
  - Use filters!
- Status Checks
  - Fine-tune your service monitors
  - Custom integrations through a simple URL
- Streaming API
  - From the CLI
  - From the browser


I'M HELPING
BUILD THE WORLD'S
LARGEST INTERNET
MEASUREMENT
NETWORK
RIPE ATLAS

RIPE Atlas community

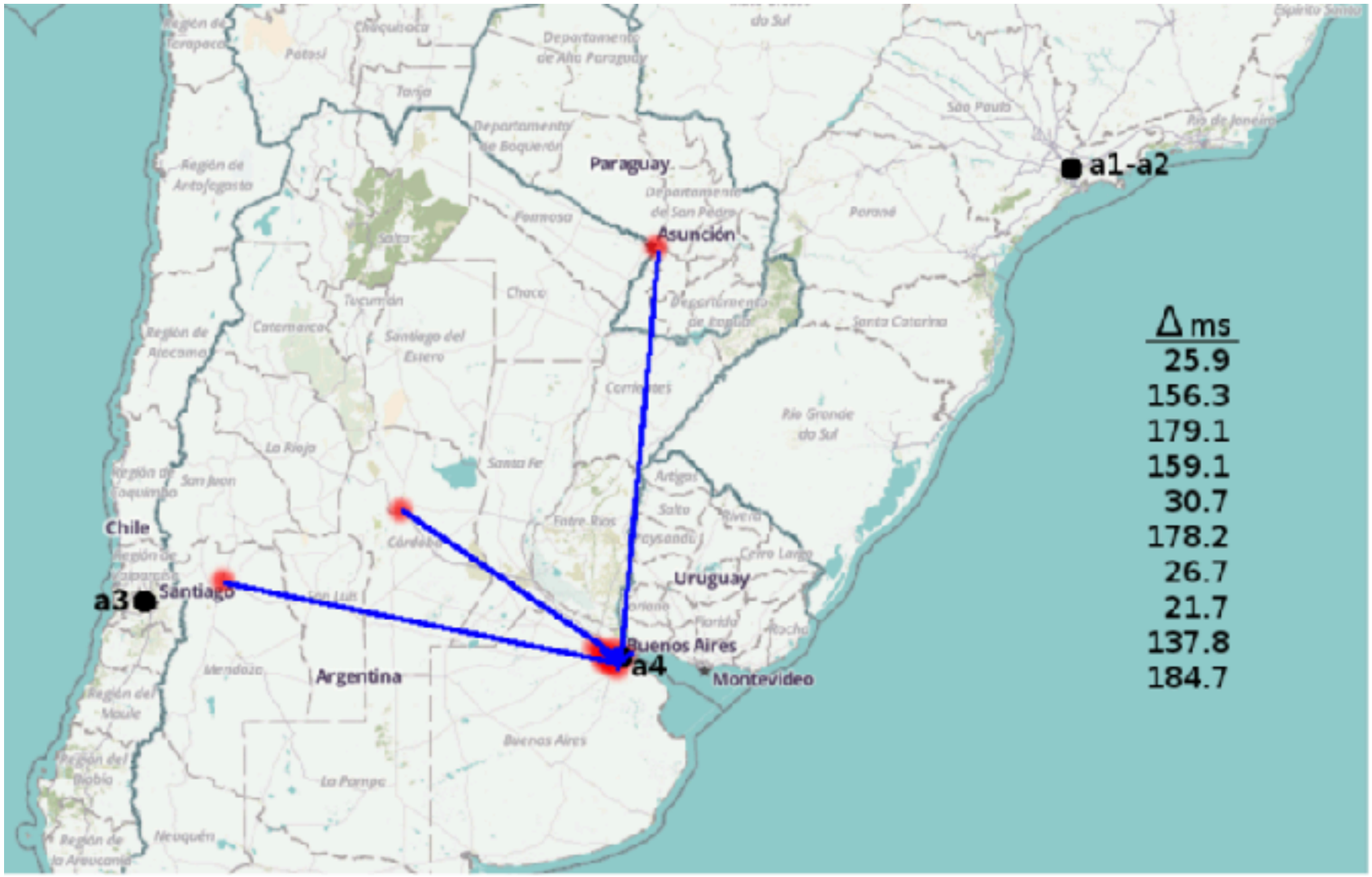
- Users
- Hosts
  - Probes
  - Anchors
- Sponsors
- Ambassadors
- Developers
- Network operators
- Researchers





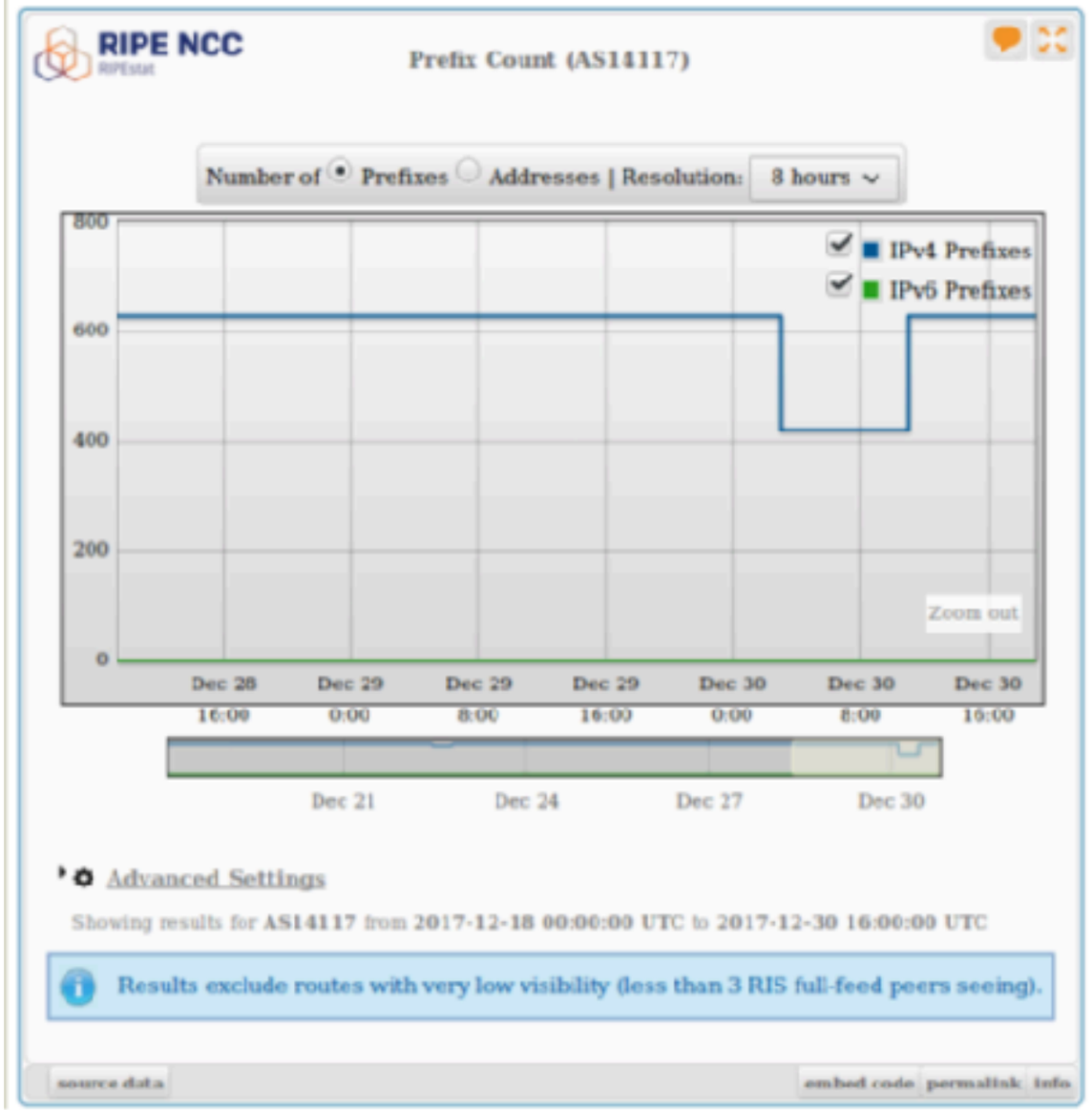
What RIPE Atlas users say (part 1)

- Visualisation of a New Node in LACTLD Anycast Service (Hugo Salgado):
  - https://labs.ripe.net/Members/hugo_salgado/visualisation-of-a-new-node-in-lactld-anycast-service



What RIPE Atlas users say (part 2)

- Using RIPEstat to Analyse Cable Cuts in Chile:
  - https://labs.ripe.net/Members/hugo_salgado/using-ripestat-to-analyse-cable-cuts-in-chile

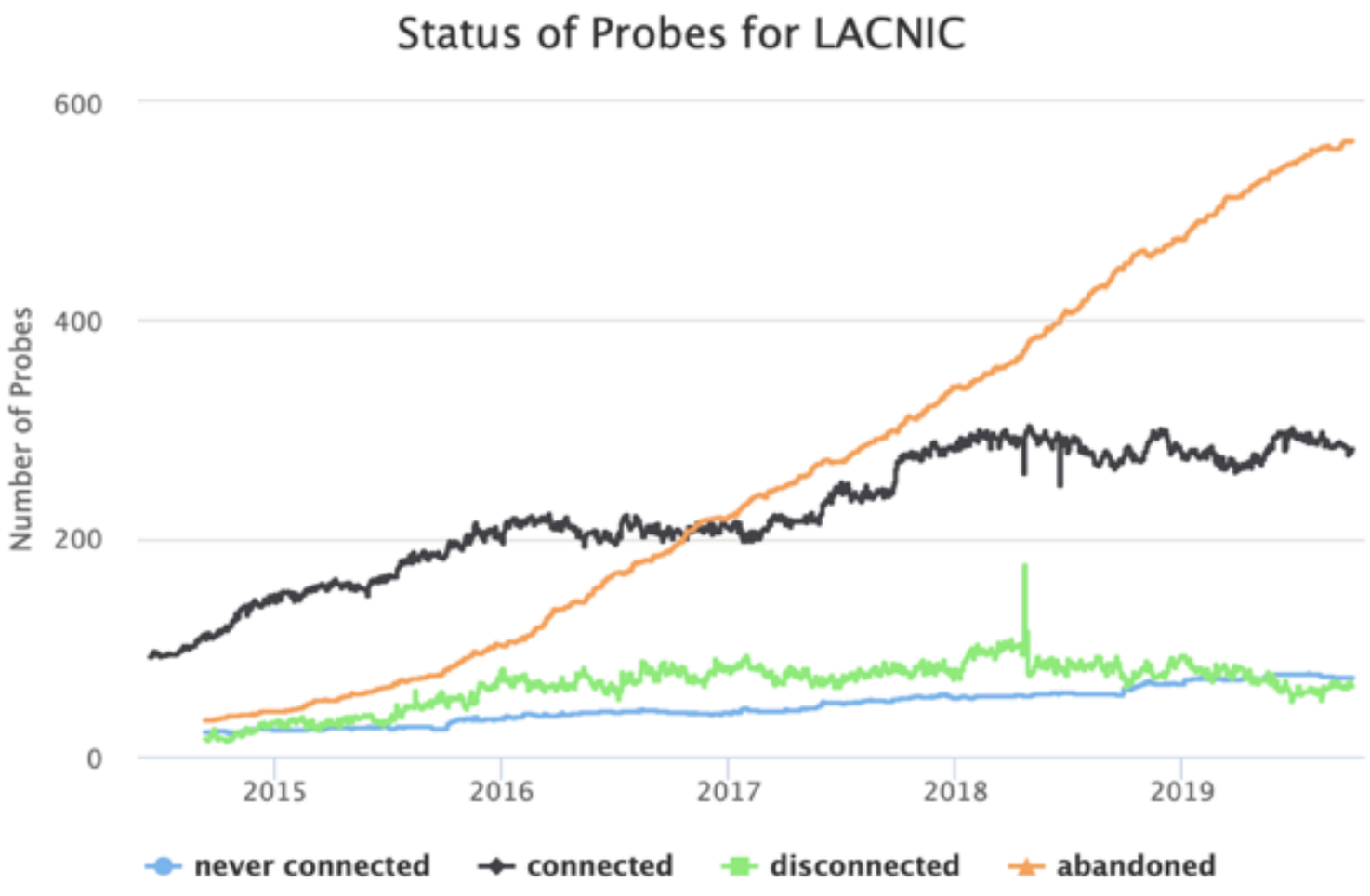


Hosting a probe

- Create a RIPE NCC Access account
- Go to https://atlas.ripe.net/apply
- Get a probe from a RIPE Atlas ambassador at a conference
- Available soon the software probes:
  - More news during RIPE 79 Meeting in Rotterdam

Probes in LAC

- 282 connected probes (total worldwide is 10.000+)
- Connect your abandoned probe!
  - https://atlas.ripe.net/docs/troubleshoot-probe-issues/



Hosting an Anchor

- Measurement targets that function as powerful RIPE Atlas probes
- Regional **baseline**
- RIPE NCC Anchoring measurements: ping, traceroute, **HTTP**
- +500 probes targeting each anchor
- 10x **more credits** than a probe

Anchors in LAC

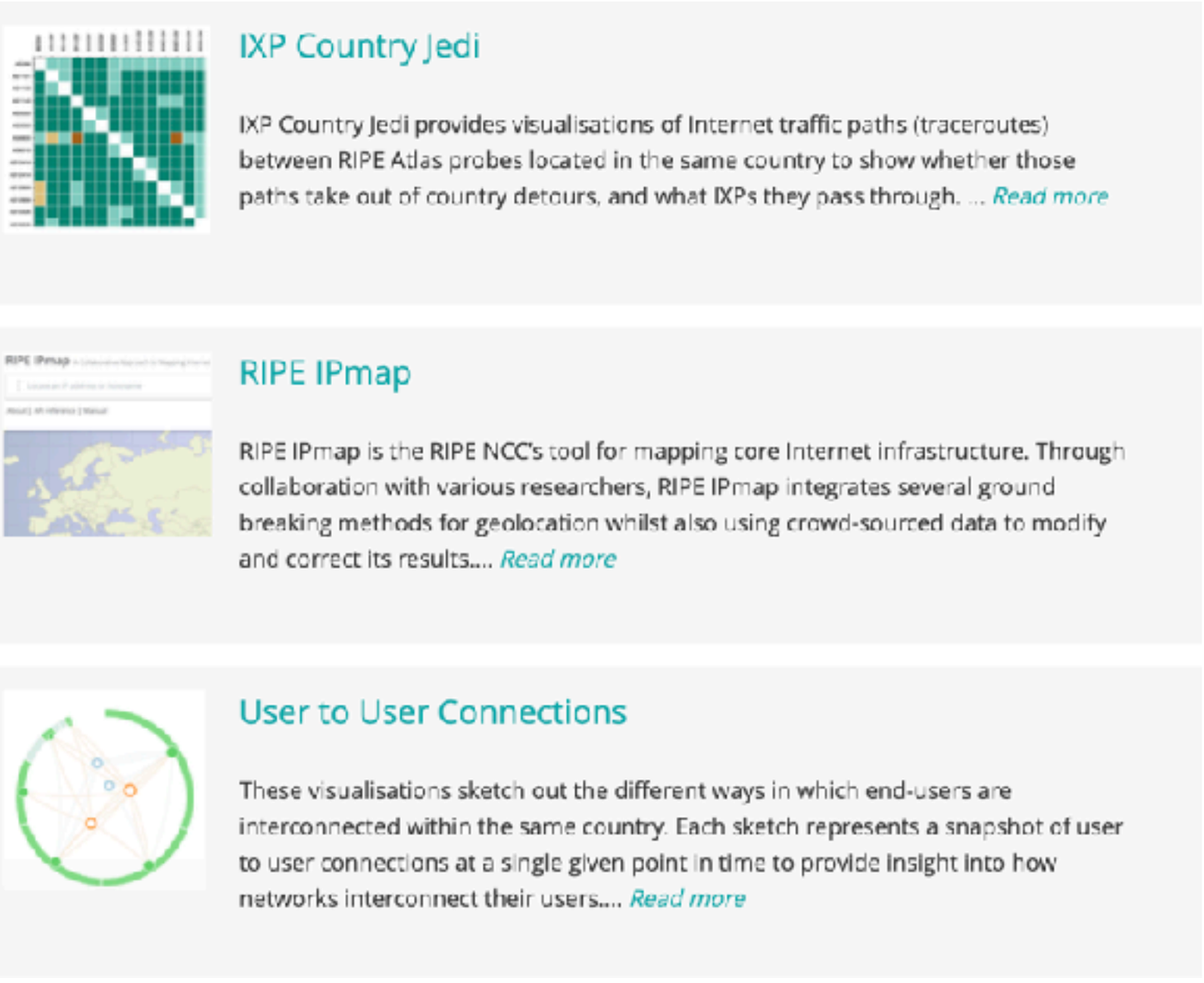
- 22 **anchors (5** are **VMs)** thanks to:
  - Hosting organisations
  - Sponsoring from **LACNIC**

Apply Now!

- We need more anchors in LAC; more coverage, better results!
- Now also VM anchors available
- What is required to host an anchor:
  - https://atlas.ripe.net/get-involved/become-an-anchor-host/
- Apply!
  - https://atlas.ripe.net/anchors/apply/

New: RIPE Labs Tools

- A list of prototype tools and visualisations we are currently working on
  - https://labs.ripe.net/playground



Contacting RIPE Atlas

- https://atlas.ripe.net
- Users mailing list: ripe-atlas@ripe.net
- Articles & updates on RIPE Labs: https://labs.ripe.net/atlas
- Questions and bugs: atlas@ripe.net
- Twitter: @RIPE_Atlas and #RIPEAtlas

References

- atlas.ripe.net
- RIPE Atlas en Latino America y Caribe
- Get involved
- About RIPE Atlas anchors
- VM anchors requirements and installation instructions
- VM anchors in production - RIPE Labs

The End!

Край

Y Diwedd

Fí

Finis

النهاية

Соңы

վերջ

Liðugt

Ende

Finvezh

Кінець

Konec

Kraj

Ënn

Fund

پایان

Kpaj

Lõpp

Beigas

Vége

Son

An Críoch

הסוף

Endir

Fine

Sfârşit

Fin

Τέλος

Einde

Конец

Slut

Slutt

დასასრული

Pabaiga

Tmiem

Koniec

Fim

Amaia

Loppu