ICANN DNS-STATS

Open Source tools for DNS Capture, analysis and monitoring

Mauricio Vergara Ereche

LACNOG 2018
Sep 2018

ICANN

Agenda

1. The need to monitor DNS internally
2. A bit of history: DSC
3. Introducing DNS-STATS
4. C-DNS Format and IETF
5. DNS-STATS Present and Future
6. Comments and Questions

The need to monitor and analyze DNS

Monitoring and Analyzing DNS servers

- What to monitor?
  - Internal Service status
    - Is the service available/responding/answering
    - How fast are we responding
    - What's the server capacity
    - More complex questions
      - Client characterization
      - Group/classify bulks of data
      - Grouping set of servers into different views.
      - Analyze traffic and search of patterns
  - External service status
    - Is the service available everywhere?
    - Are we giving the same answer consistently to every client?
    - Perception of service from the client side

Possible solutions (external monitoring)

- Using monitoring distributed services (à la “looking glass”)
  - RIPE: **DNSMON**
  - ThousandEyes: **DNS Monitoring**
  - Uptrends **Monitoring**
  - **DNSChecker** Propagation and Resolution tool
- DIY approach
  - RIPE **Atlas**
  - NLNOG **Ring**
  - Any cloud hosting service (build your own farm of monitors)
- **OUT OF SCOPE FOR THIS PRESENTATION**

Possible solutions (internal monitoring)

- Let's do some graphs!
  - `RRDtools` or similar approaches.
    - `Nagios, Icinga, MRTG, Cacti, Observium, Zabbix`
  - Let's do elastic graphs!
    - `Kibana, Grafana`
- What about the complex questions?
  - Analyze `syslog` and daemon `logs`
  - Command line tools
    - `dnstop, tcpdump, wireshark`
  - Collect traffic and then analyze
    - Capture: `pcap, dnscap, dnstap, dsc`
    - Analyze: `packetq` or your usual *swiss army knife* (`Perl, Python, awk`) with their own DNS libraries
- Build one solution for most of these requirements

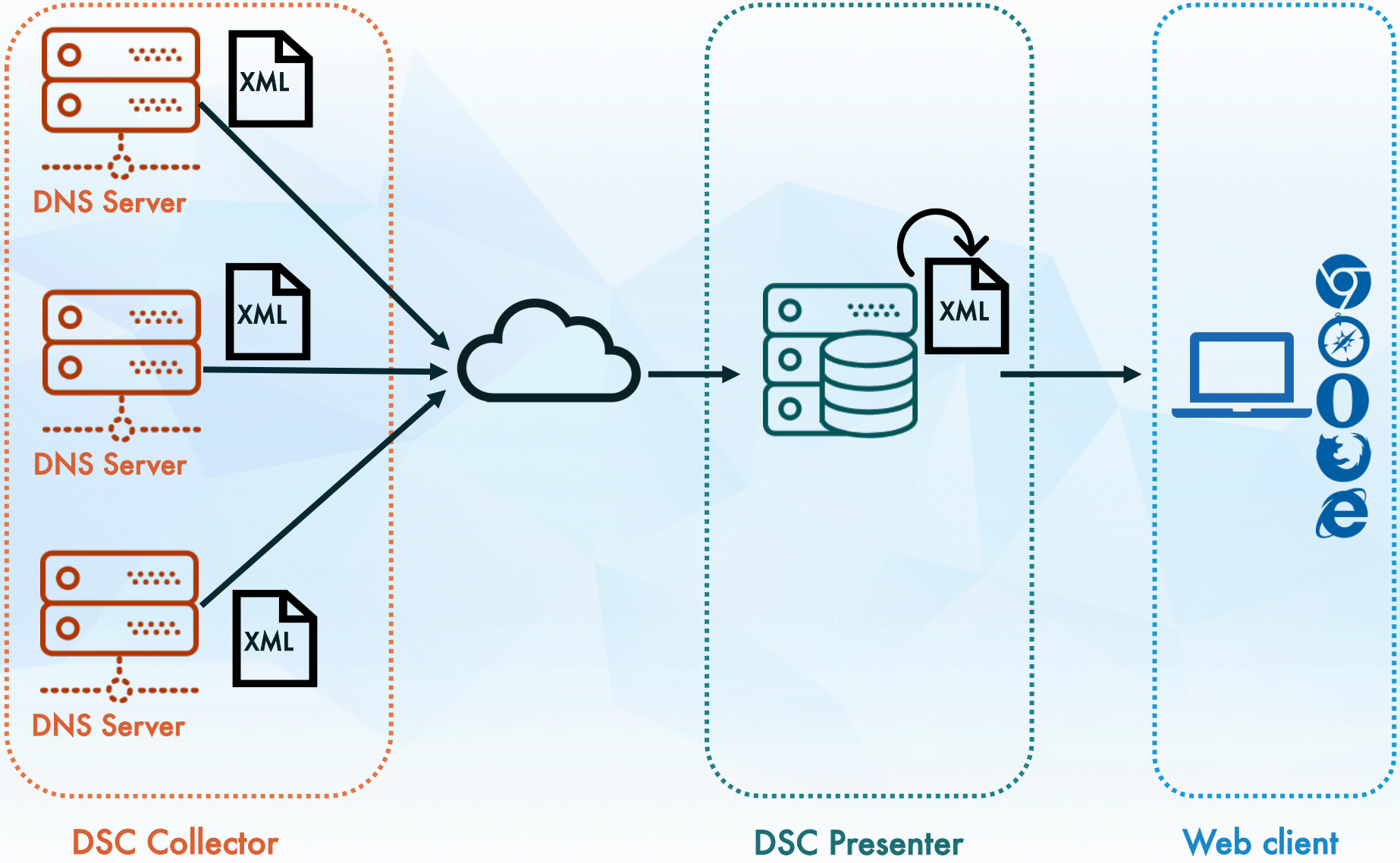
A bit of history: DSC

since 2006

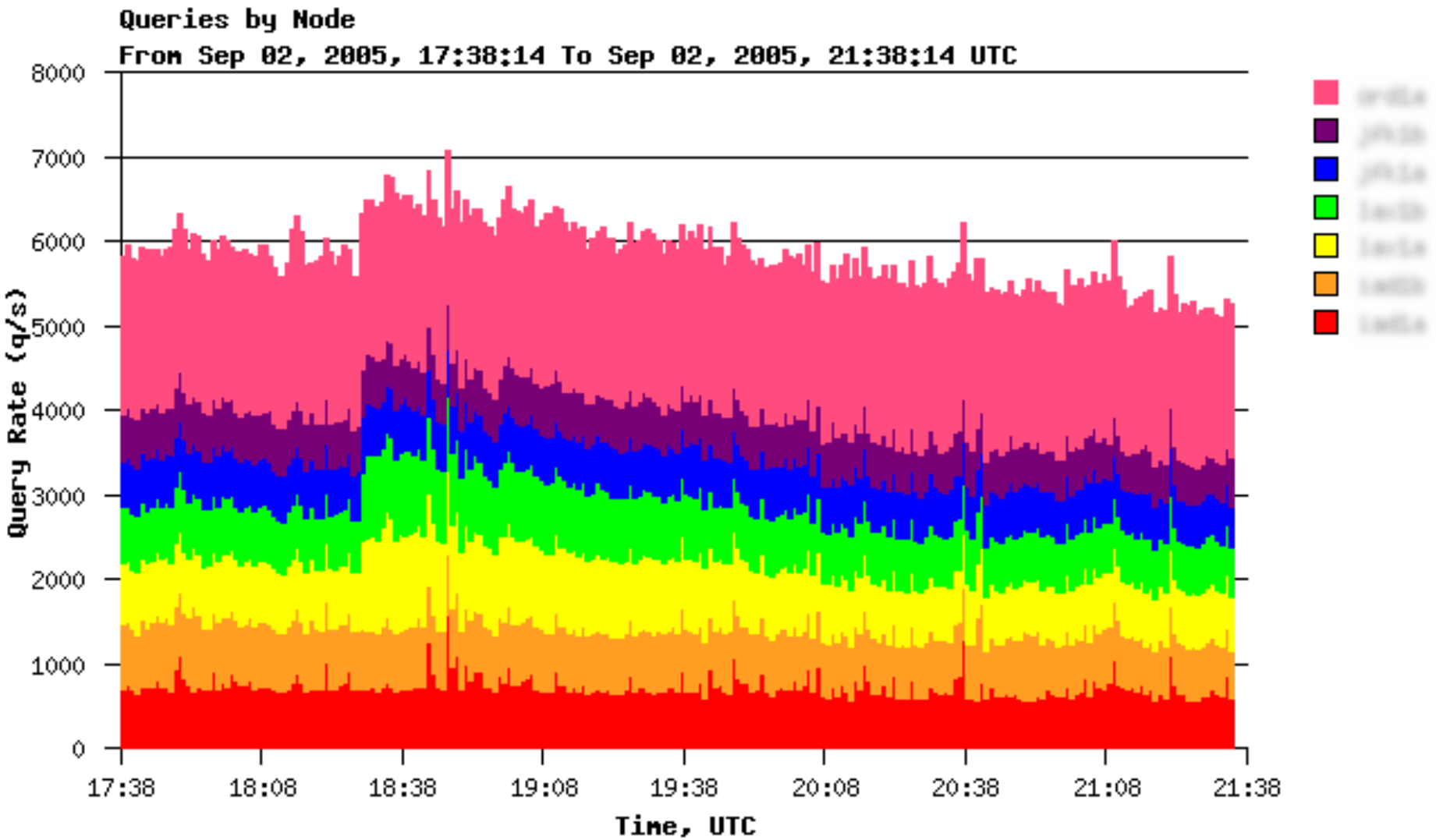
A bit of history: DSC

- DNS Statistics Collector (DSC)
  - Created by **The Measurement Factory** in 2005
  - Maintained by **DNS-OARC** since 2016
- 2 parts concept: *Collector* and *Presenter*
  - **DSC Collector**
    - Runs on every DNS server.
    - Captures raw traffic using `libpcap` (think `tcpdump`)
    - Extracts summary of DNS traffic from `PCAP` and creates `XML` every "X" amount of minutes.
    - Send `XML` data to *Presenter*
  - **DSC Presenter**
    - Receives `XMLs`, summarize every server using `Perl` and builds another `XML` (newer versions make a .DAT file, other newer patches includes DB integration)
    - Display graphs under web interface

A bit of history: DSC basic schema



A bit of history: DSC Presenter Screenshot



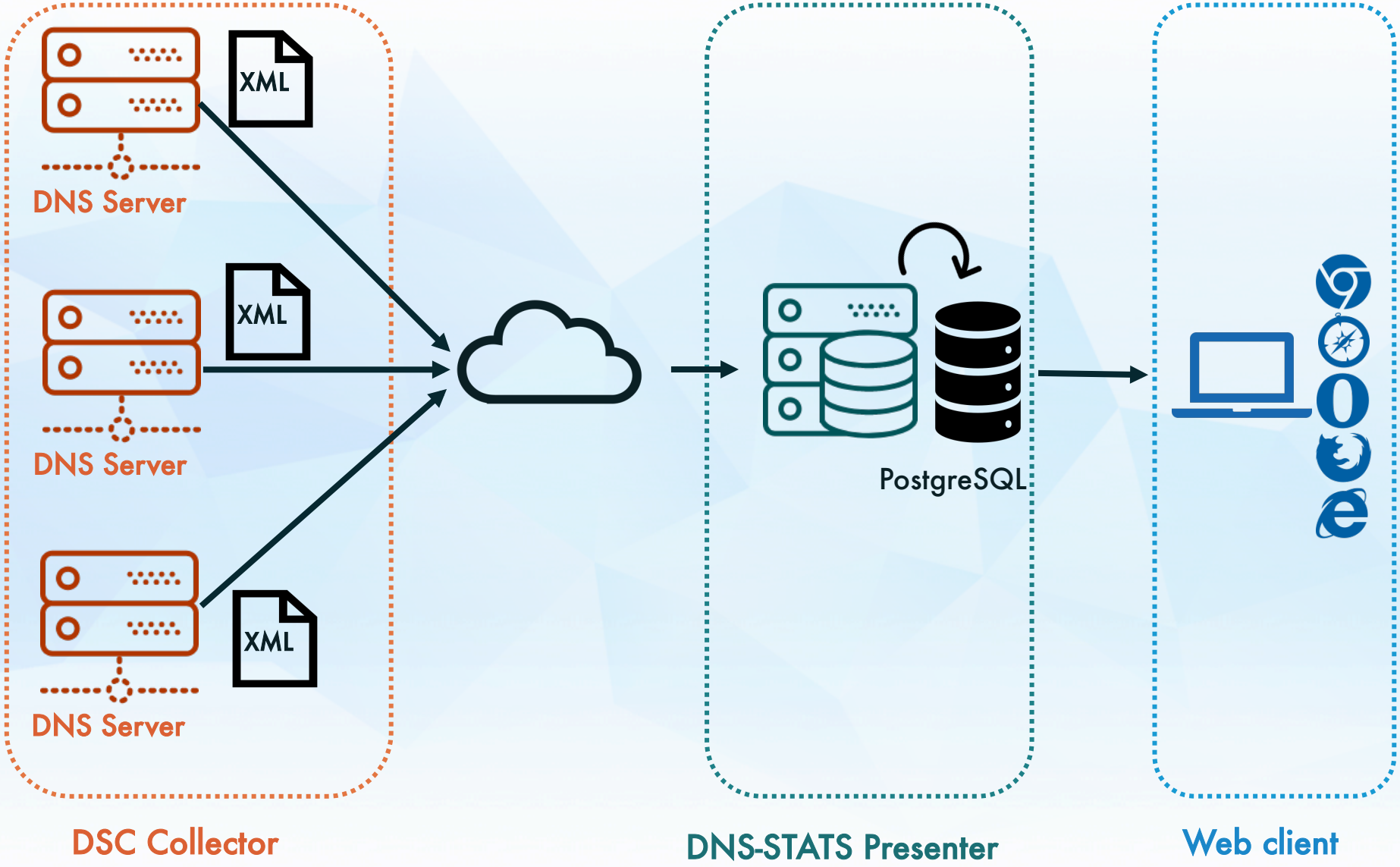
Introducing DNS-STATS

ICANN DNS Engineering approach

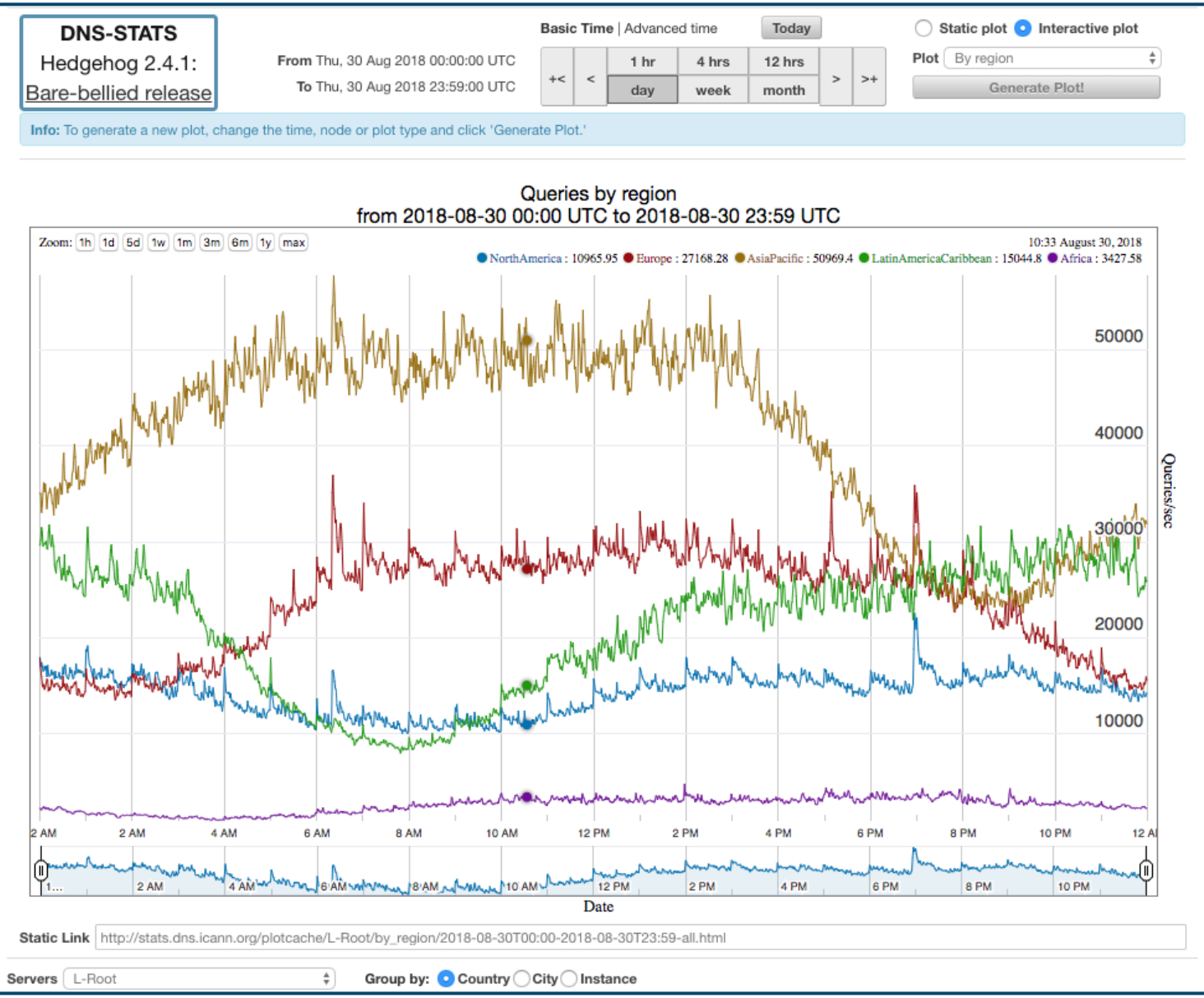
DNS-STATS Presenter (Hedgehog)

- DNS-STATS **Presenter** (Hedgehog) was originally designed to replace widely used DSC Presenter
  - Problems scaling Anycast cloud with 100+ nodes
    - Replace **file storage** on the DSC Presenter with Postgres DB
    - Several Interface improvements
- Development started from scratch in 2014
  - Open Source (Mozilla Public License v 2.0)
  - http://dns-stats.org
  - github.com/dns-stats/hedgehog
  - Developed for ICANN by Sinodun IT
  - Currently at version 2.4.1
- Live version used by ICANN DNS Engineering
  - http://stats.dns.icann.org

DNS-STATS Presenter (Hedgehog) basic schema



DNS-STATS Presenter (Hedgehog)



C-DNS

A new DNS Capture format

C-DNS: Compacted-DNS Format

- An efficient file format to collect DNS queries and responses
  - Uses Concise Binary Object Representation - `CBOR [RFC7049]`
  - It focuses on capturing and storing large packet capture files of DNS traffic
    - PCAP capture: `661.87MB (49.09MB xz-compressed)`
    - CBOR block: `67.98MB (17.94MB xz-compressed)`
- IETF draft introduced in 2016
  - https://datatracker.ietf.org/doc/draft-ietf-dnsop-dns-capture-format/00/
  - Current: `draft-ietf-dnsop-dns-capture-format-08`
  - Latest draft is under **WG Last Call**

DNS-STATS Compactor

- `DNS-STATS Compactor` was designed to capture `C-DNS`
  - Including more information:
    - `TCP` resets
    - `ICMP` Messages
  - With these new features, `DSC Collector` becomes redundant
- Development started in 2016
  - Open Source (Mozilla Public License v 2.0)
  - https://github.com/dns-stats/compactor
  - Developed for ICANN by Sinodun IT
  - Current version 0.11.1
- It became a set of tools for capturing and working with DNS server traffic in **`Compacted-DNS`** (`C-DNS`) files.

C-DNS and DNS-STATS Compactor suite

- DNS-STATS Compactor suite has 2 main programs:
  - **`compactor`**
    - Similar in usage to `tcpdump`
    - Reads traffic from 1+ interface(s) and outputs selected details into `C-DNS` and `PCAP` files (ignored, raw)
    - It can also be used to read pre-recorded `PCAP` files.
    - Can be configured to compress its output (`xz` or `gzip`)
  - **`inspector`**
    - Reconstructs traffic from `C-DNS` files generated by `compactor`
    - It outputs one or more `PCAP` files
- Reminder:
  - https://github.com/dns-stats

C-DNS and DNS-STATS `compactor CLI`

```
$ ls -l
-rw-rw-r-- 1 mave mave 5676956 Sep  3 18:40 input.pcap.xz

$ time unxz input.pcap.xz
real    0m0.658s
user    0m0.544s
sys     0m0.108s

$ ls -l
total 82032
-rw-rw-r-- 1 mave mave 83999423 Sep  3 18:40 input.pcap

$ time compactor -X on input.pcap -o new-cdns.cbor
real    0m6.206s
user    0m11.140s
sys     0m1.512s

$ ls -l
total 84912
-rw-rw-r-- 1 mave mave 83999423 Sep  3 18:40 input.pcap
-rw-rw-r-- 1 mave mave  2945948 Sep  3 18:48 new-cdns.cbor.xz
```

Decompressing a 5.5MB PCAP file took 0.6s to make a 84MB PCAP file.

compactor took 6s to convert the 84MB PCAP into a compressed 2.9MB CBOR (C-DNS) file

C-DNS and DNS-STATS `inspector CLI`

```
$ time unxz new-cdns.cbor.xz
real    0m0.365s
user    0m0.340s
sys     0m0.020s

$ ls -l
total 92820
-rw-rw-r-- 1 mave mave 83999423 Sep  3 18:40 input.pcap
-rw-rw-r-- 1 mave mave 11045941 Sep  3 18:48 new-cdns.cbor

$ time inspector new-cdns.cbor -o output.pcap
real    0m7.466s
user    0m7.208s
sys     0m0.236s

$ ls -l
total 174856
-rw-rw-r-- 1 mave mave 83999423 Sep  3 18:40 input.pcap
-rw-rw-r-- 1 mave mave 11045941 Sep  3 18:48 new-cdns.cbor
-rw-rw-r-- 1 mave mave 83999423 Sep  3 18:50 output.pcap
-rw-rw-r-- 1 mave mave     1131 Sep  3 18:50 output.pcap.info
```

Decompressing 2.9MB CBOR took 0.3s to make a 11MB C-DNS file.

Inspector took 7.5s to re-generate a 11MB C-DNS file into a 84MB PCAP file

DNS-STATS current status

and future

DNS-STATS Compactor latest versions

- Since 2018 (versions 0.11.0+):
  - Add pseudo-anonymization for the output of `inspector`.
  - Enable use with `libtins v4.0.`
  - CBOR - use definite-length items where possible.
  - Small packet receive optimization.
  - Improve detection of malformed `EDNS0`.

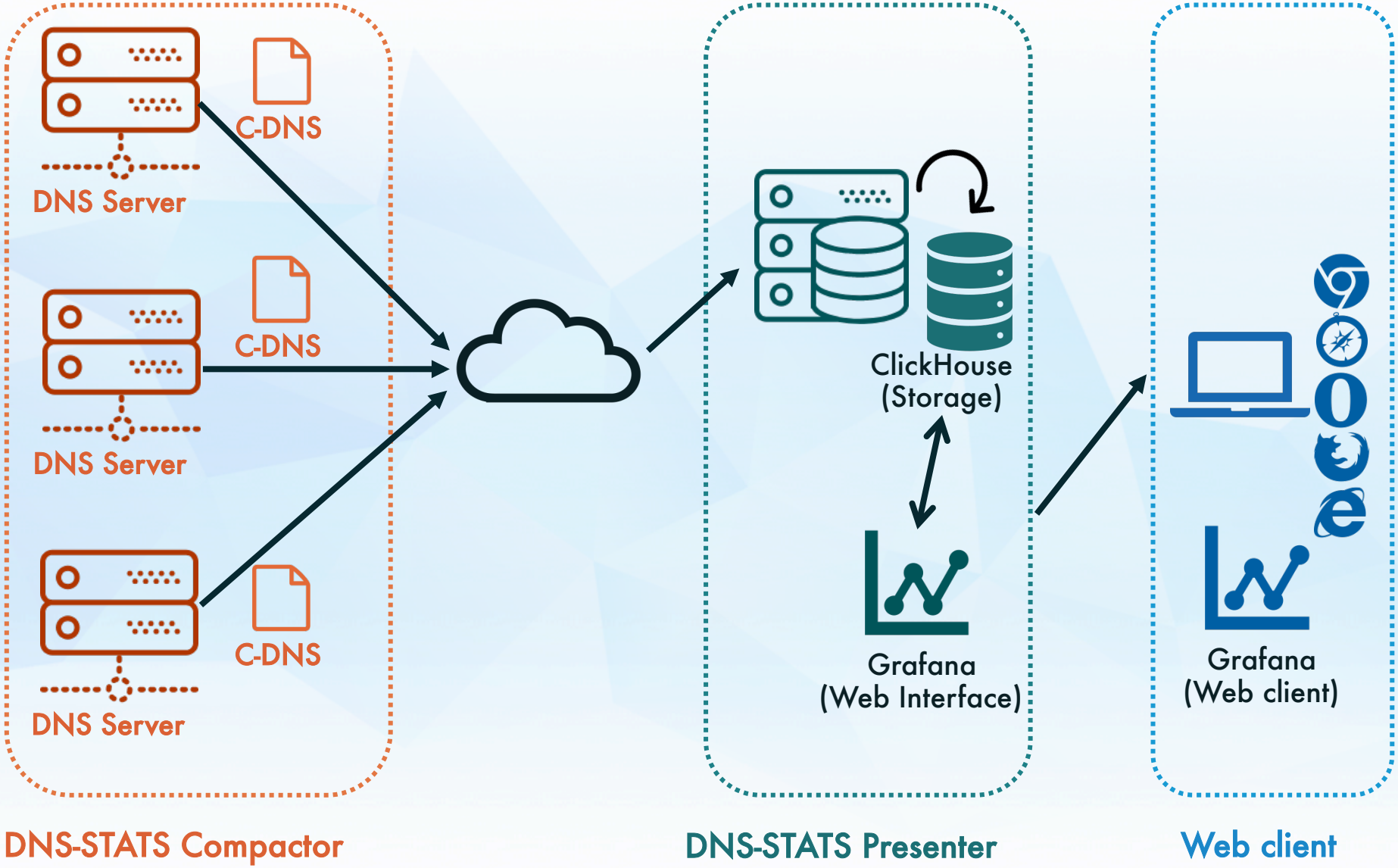
DNS-STATS Compactor - future

- More frequent **DITL** captures for the *ICANN Managed Root-Server* (IMRS)
  - Already in conversations with DNS-OARC org.
  - Use `C-DNS` format only (no-more `PCAP` files)
- Anonymization schema implementation based on DNS Operations mailing list conversations.
  - General Data Protection Restriction (GDPR) already in effect.
- Expected to change IETF status of C-DNS from DRAFT to RFC (current: Working Group Last Call)

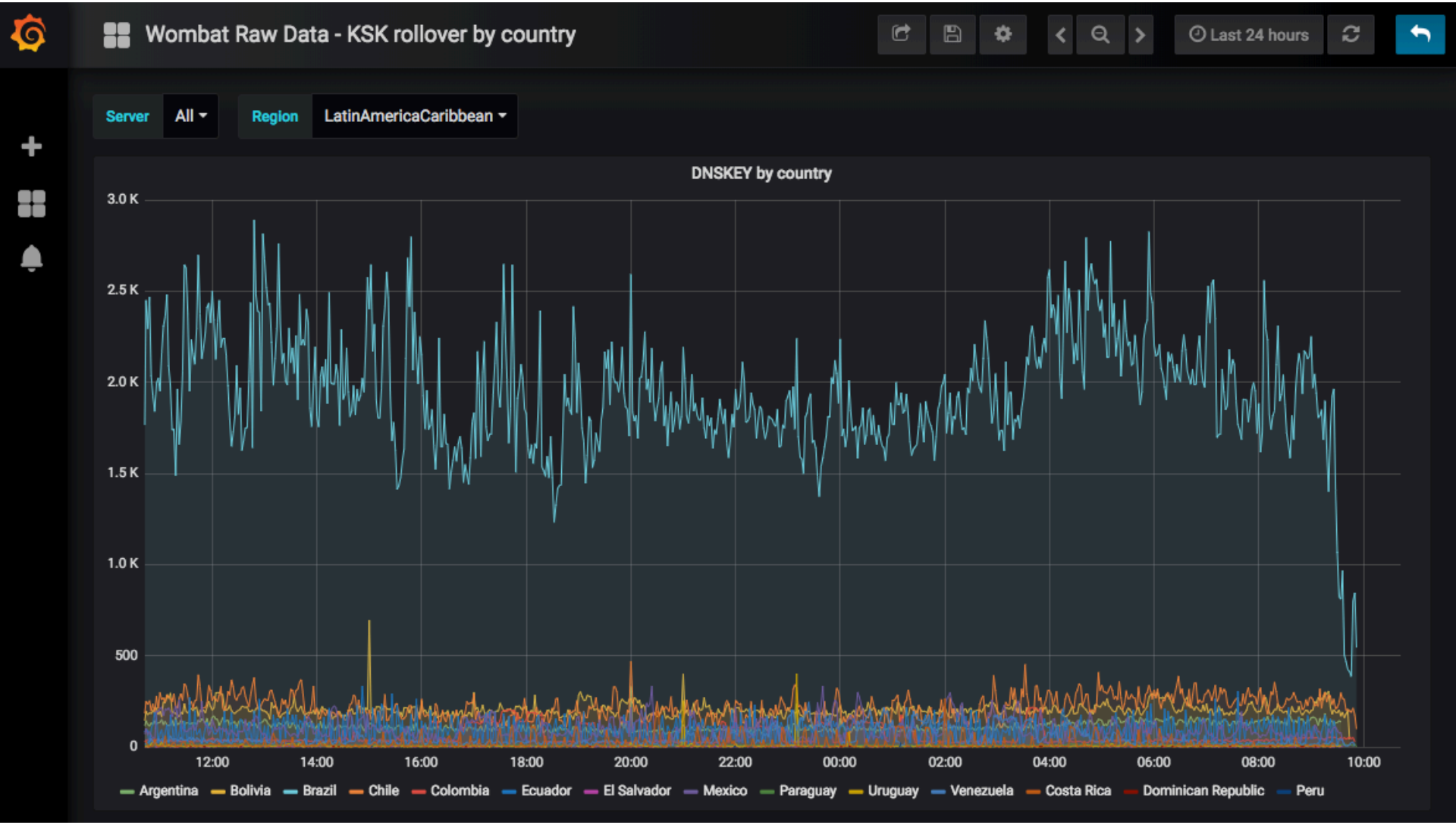
DNS-STATS Presenter - future

- Version 3 is currently in development
  - Current codename: **`Wombat`** (but it might change in the future)
- Reduce usage of `PostgreSQL` (only kept for meta-data) and makes a complete switch to a cluster of `ClickHouse.`
  - `SQL-alike` queries across all the data!
- Use `Grafana` web interface to plot the data in real time
  - Ability to create your own plots
  - Able to export PCAP files on request.
- Fed with `C-DNS` files instead of `DSC Collector XML` files.

DNS-STATS Compactor basic schema (future)



DNS-STATS Presenter: Wombat (future)



DNS-STATS Presenter: Wombat (future)

Graph | General | Metrics | Axes | Legend | Display | Alert | Time range

Data Source: Wombat | Query Inspector

A

```
SELECT t, groupArray((CountryName, ctryc)) as nodes
FROM
(
  SELECT t, CountryName, sum(c) as ctryc
  FROM
  (
    SELECT
      $timeSeries as t,
      NodeID,
      count() as c
    FROM $table
    WHERE QueryType = (SELECT value from wombat.iana_text WHERE registry_name = 'QTYPE' AND value_name = 'DNSKI
    GROUP BY t,NodeID
    ORDER BY t
  ) ANY INNER JOIN (SELECT server_name as ServerName, region_name as RegionName, country_name as CountryName, t
  WHERE c > 0 AND ServerName IN ($Server) AND RegionName in ($Region)
  GROUP BY t,CountryName
  ORDER BY t
)
GROUP BY t,CountryName
ORDER BY t,CountryName
```

Step: 2m | Resolution: 1/1 | Round: 0s

Format as: Time series | Show Help | Generated SQL | Reformat Query

B | Add Query

Summary

1. More alternatives to monitor DNS
2. DNS-STATS Open Source
3. DNS-STATS Compactor suite
4. C-DNS Format & IETF
5. DNS-STATS Presenter new version (Wombat)

Engage with ICANN – Thank You and Questions

ICANN

One World, One Internet

Visit us at **icann.org** and **dns.icann.org**

- @icann
- linkedin/company/icann
- facebook.com/icannorg
- slideshare/icannpresentations
- youtube.com/icannnews
- soundcloud/icann
- flickr.com/icann
- instagram.com/icannorg