DNS-STATS:
DNS Traffic capture and visualisation developments

Jim Hague jim@sinodun.com
https://sinodun.com
@SinodunCom

DNS-STATS: Traffic capture and visualisation for IMRS

- **What is DNS-STATS?** (dns-stats.org)
  - Organisation for open source DNS traffic tool development
  - Sinodun contracted to do development for ICANN DNS Engineering Team (who manage IMRS)
- **What is IMRS?** ICANN Managed Root Server (L-root)

DNS-STATS: Traffic capture and visualisation for IMRS

- **What is DNS-STATS?** (dns-stats.org)
  - Organisation for open source DNS traffic tool development
  - Sinodun contracted to do development for ICANN DNS Engineering Team (who manage IMRS)
- **What is IMRS?** ICANN Managed Root Server (L-root)

IMRS is ~280 servers
Managed as ~170 (location based) "Instances"

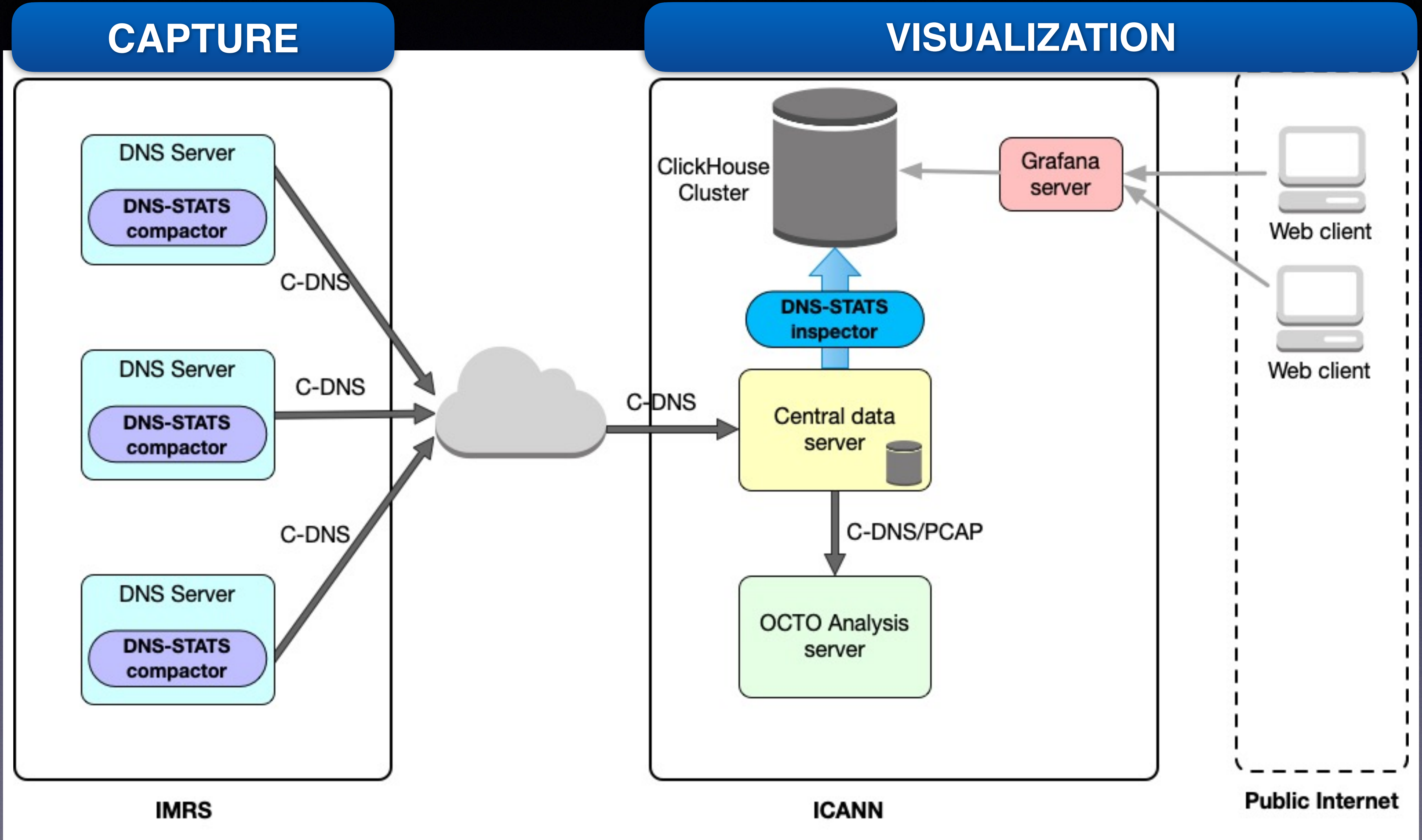
Total traffic is **~17 billion queries per day**

IMRS data collection - Background

- **Historically** used a combination DSC XML + Hedgehog (+PCAP)
- **Now** migrated to DNS-STATS: C-DNS & (ClickHouse/Grafana solution)

- **Traffic capture:** “Compressed-DNS” (RFC8618, C-DNS):
  Published in 2019: a CBOR based DNS specific file format for traffic capture
  - Capture pairs query/responses and indexes common data
  - Why use it? **Much** smaller than PCAP with most of same info
- **Data visualisation:** Real time import raw C-DNS data into ClickHouse & display in Grafana. (Aggregation of data allows for faster displays.)

Architectural Overview



DNS-STATS Status &Updates

- **IMRS fully migrated** to new system over the last few years
- **C-DNS capture software open sourced in** 2017 (IDS presentation)
  - v1.0 based on the RFC format released in 2020
    - (Biggest change was making all fields optional)

- Two recent updates:
  - **Capture**: Add ability to capture from DNSTAP
  - **Visualisation**: Open sourced the visualisation component

C-DNS motivation:
Target limited use case

IMRS is (mainly) hosted servers in challenging environments

- Data collection on **same hardware** as nameserver
- Minimise server resources conflict: **1 RU server**
- Collected data **stored on same hardware**
- **Upload** will use the same interface as DNS traffic

C-DNS File sizes

Format	PCAP	C-DNS
File size (Mb)	660	75
Compressed with ‘xz -9’ (Mb)	49	18
User time for compression (s)	161	39

C-DNS File sizes

Format	PCAP	C-DNS
File size (Mb)	660	75
Compressed with ‘xz -9’ (Mb)	49	18
User time for compression (s)	161	39

***COMPRESSED SIZE**: C-DNS is 30-40% size of PCAP*

C-DNS File sizes

Format	PCAP	C-DNS
File size (Mb)	660	75
Compressed with ‘xz -9’ (Mb)	49	18
User time for compression (s)	161	39

***COMPRESSED SIZE**: C-DNS is 30-40% size of PCAP*

***COMPRESSION CPU**: C-DNS uses ~25% of PCAP*

DNS-STATS Implementation Status:

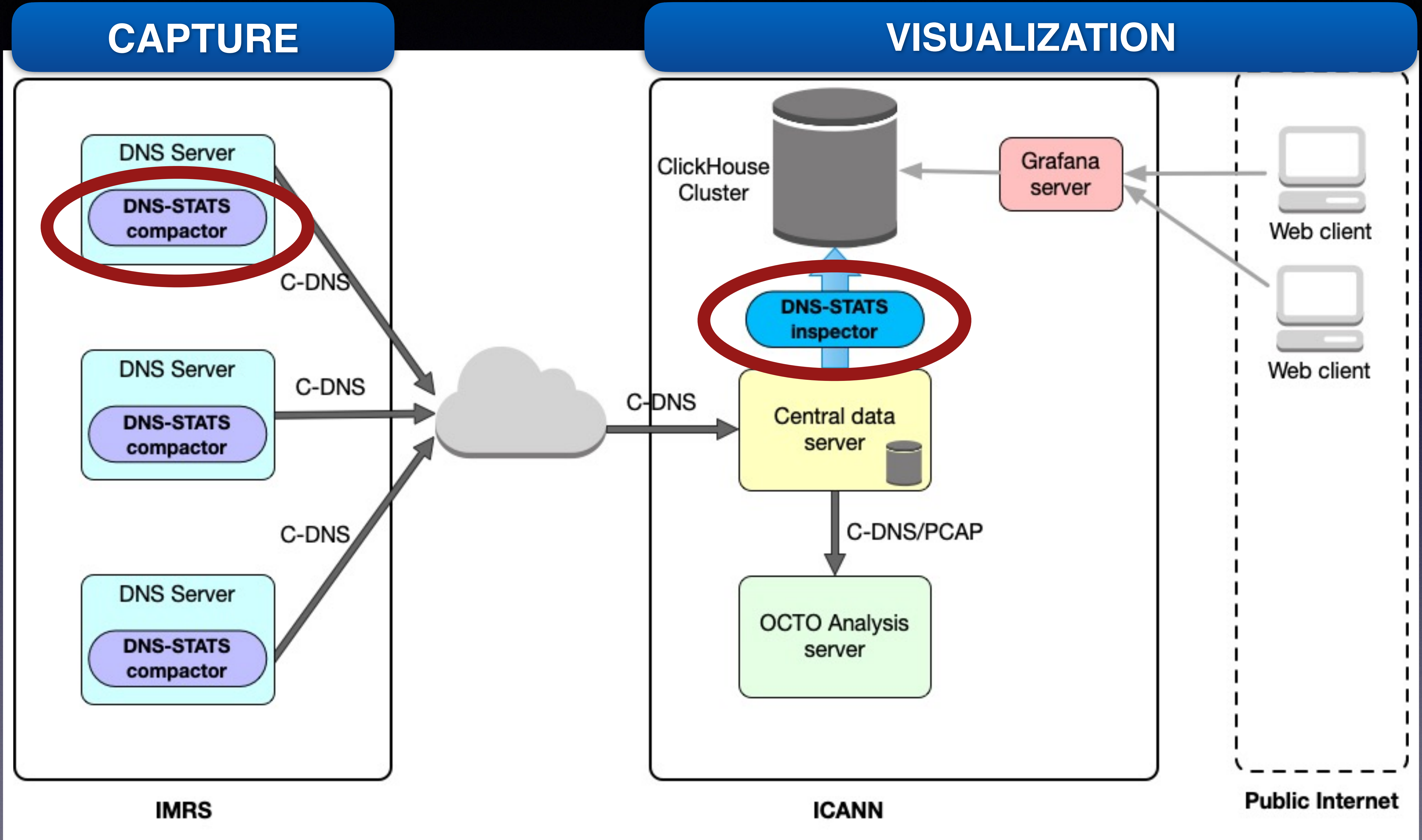
compactor

DNS-STATS Implementation Status: *compactor*

- dns-stats github: https://github.com/dns-stats/compactor

- Software actually has two components:
  - **compactor**: Captures & compresses traffic in C-DNS format from:
    1. Network interface or PCAP files
    2. (NEW) DNSTAP socket or file
  - **inspector**: Reads C-DNS and has 2 output formats:
    1. Templated text output (for import to database)
    2. PCAP (Lossy reconstruction)

Architectural Overview



DNSTAP support in *compactor*

- What is the **use case**?
  - C-DNS originated when “DNS Privacy” was in infancy
  - Original use case and implementation was for **authoritative traffic capture on the wire**
  - But for recursives using DoT/DoH, **capture on wire not possible**

- DNSTAP is **implemented IN nameserver** software
  - It reports processed DNS queries
  - Implementations exist for BIND, Unbound, NSD, Knot Res+Auth

DNSTAP background

- **DNSTAP** (https://dnstap.info/)
  - Introduced 2013 by Farsight Security
  - Google Protocol Buffers binary format
  - Several implementations but **not standardised**
- Implementations:
  - Nameserver connects to socket provided by listening app
  - Nameserver reports DNS binary packet contents
    (+ selected meta-data):
    - Wrapped in Frame Streams transport

DNSTAP support in *compactor*

- **Challenges**:
  - **Lack of standard definition** leads to slight differences in implementations (e.g which fields are supplied)
  - **Frame Streams** is not part of spec and largely undocumented
  - **Meta-data is different** (c.f. network interface collection)
    - DNSTAP **will not** include packet stats, (probably) server IP, …
    - DNSTAP **will** report query bailiwick, message type, “from cache”

- **compactor** implementation fully tested with **Unbound** (limited testing: **BIND** and **Knot Res)**

DNS-STATS
Implementation Status:

visualizer

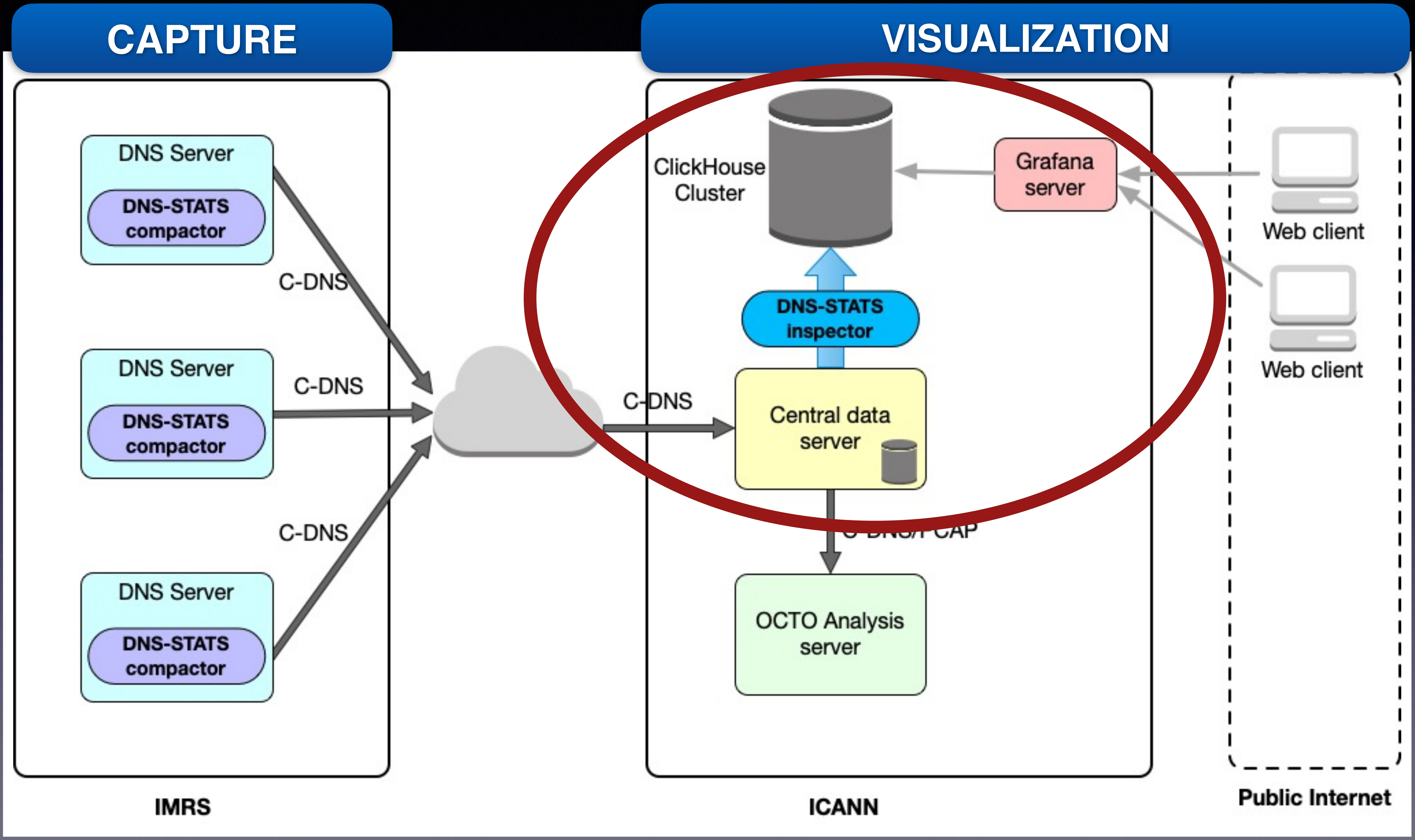
DNS-STATS Implementation Status: *visualizer*

- dns-stats github: https://github.com/dns-stats/visualizer

- A *FRAMEWORK* for displaying C-DNS data which combines…
  - **inspector**
  - File processing queue management (Gearman)
  - DB schemas for Postgres and ClickHouse
  - Management tools
  - Basic Grafana dashboards

- **Packages are provided** - they install a basic but complete system
  - NOTE: A customised version is used for IMRS

Architectural Overview



DNS-STATS: *visualizer*

- **“Build your own *visualizer*”** - Extension & customization is straightforward
- **Graphs**:
  - Add dashboards/graphs of your choosing
- **Data**:
  - Plot raw C-DNS or custom aggregation periods (e.g 1s or 1 week!)
  - Meta-data from C-DNS imports (packet counts, malformed DNS, etc.)
- **Front ends**:
  - Customise different Grafana sites (e.g. public, private)
- **Extras**:
  - Additional data written to ClickHouse e.g. *visualiser* system monitoring

Deployment example:

IMRS use of DNS-STATS software

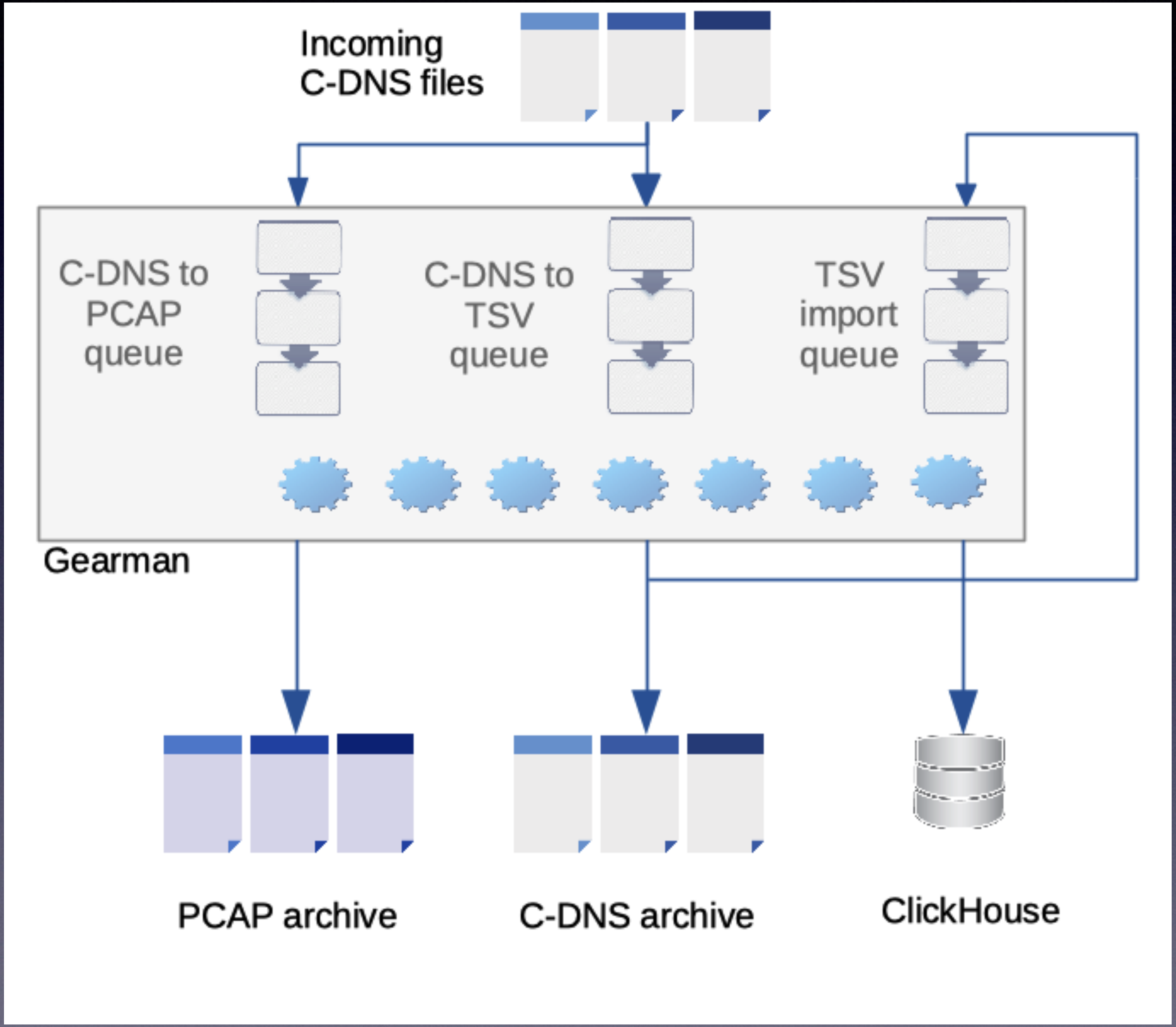
https://stats.dns.icann.org

IMRS *compactor* Deployment

- **compactor** constrained to **1 CPU** on the DNS server
  - Collects all data in query+response (specified in C-DNS)
  - Writes **xz** compressed files to local storage
  - Output file rotated every **5 minutes (configurable)**
  - Handles query rates of up to 80 kqps
    (depending on core and compression level)

- Periodically files uploaded to central collection server

inspector processing



- Uploaded C-DNS files queued for processing using **Gearman** job server and suite of Python programs
- Separate queues for:
  - **Convert C-DNS to TSV** (Tab-Separated-Value) files
  - **Import TSV** into ClickHouse database
  - (Optional) **C-DNS to PCAP** e.g. for anonymised DITL

visualiser: ClickHouse

- **ClickHouse** is an open source time series SQL column database with Grafana plugin (other plugins are available!)
- Used by various other DNS projects (CloudFlare, NIC Chile)

- C-DNS schema:
  - **Main table**: holds raw C-DNS data - per q/r pair data
  - **Aggregation tables:** Does 'ON INSERT' **aggregation** of data into separate 1s and 5min tables
    - Aggregation is simple SQL MATERIALIZED VIEW with specialised storage engine (more here)

ClickHouse cluster

- 6 server cluster
- Import process handles **~17 billion records** per day (~200 kqps)
- Disc usage **1Tb per ~39 billion records** (2+ days of raw data)
- **Management tools** provide option to retain configurable amount of each type of data (raw **vs** 1s **vs** 5m)
- Serves multiple **Grafana** front ends and can be used for ad-hoc queries for data analysis

ClickHouse numbers

- Sample query speed: **count all AAAA queries in a week**
  - Raw data is 200 kqps i.e. a packet every ~5 micro sec
  - Table sizes are for full set of DSC like aggregations

Data Type	Query Speed (s)	Rows processed	Data size (1 week)
Raw	22	123 billion	4 Tb
1 sec agg	1.6	760 million	~1 Tb
5 min agg	0.13	3 million	~0.1 Tb

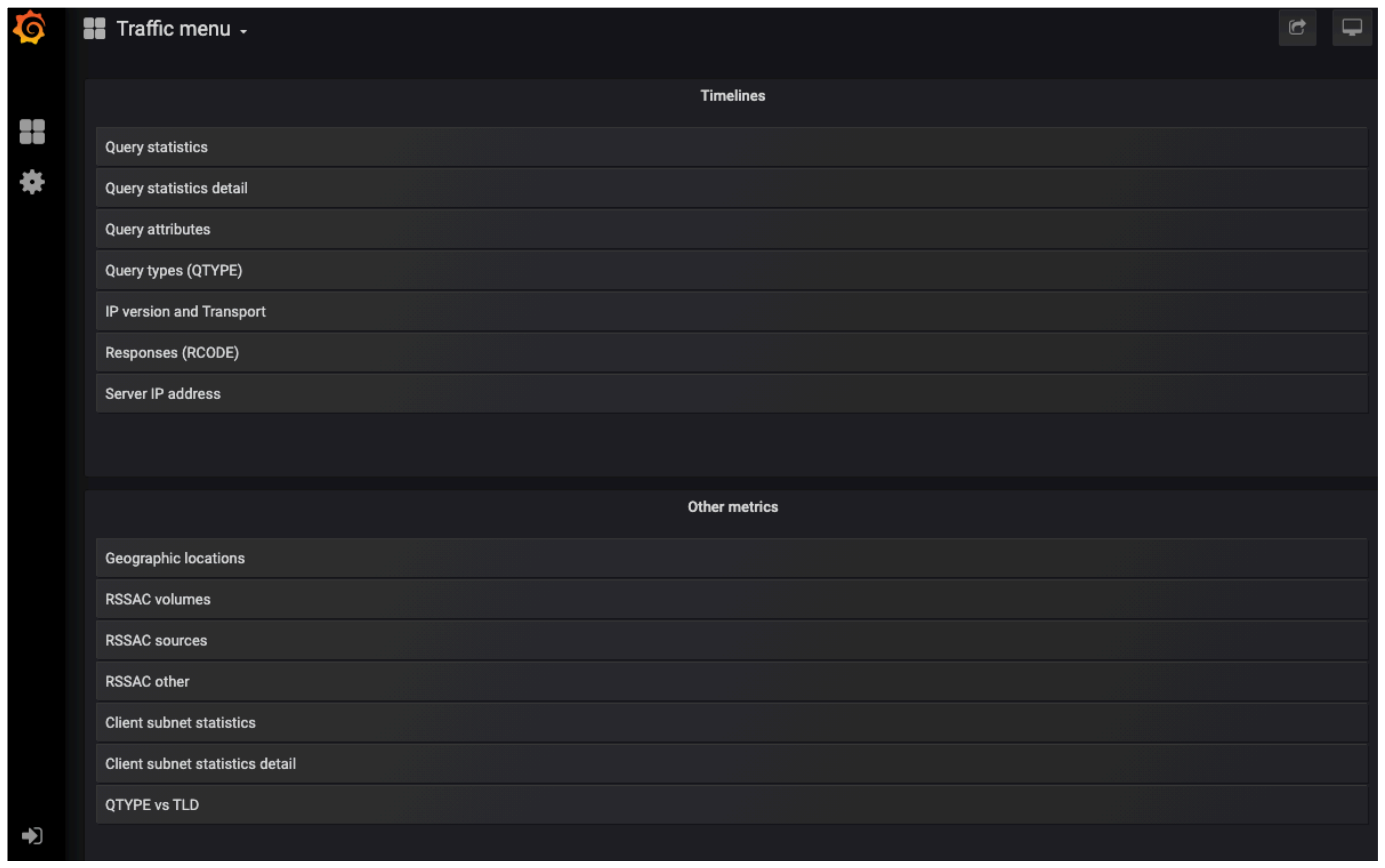
- Orders of magnitude reductions in query time and storage

visualizer: Grafana

- **Web-based visualisation platform** with various plot types:
  - Time series
  - Bar chart (using Sinodun modified plugin based on Plotly)
  - Map (using standard plugin)
  - Other plugins: ClickHouse data access, Image rendering

- ICANN public Grafana interface https://stats.dns.icann.org
  - Reproduces the various DSC like plots
  - Exposes the 5 minute data with max time window
- Additional data available via customised Grafana to ICANN staff

Grafana dashboard



Traffic menu

Timelines

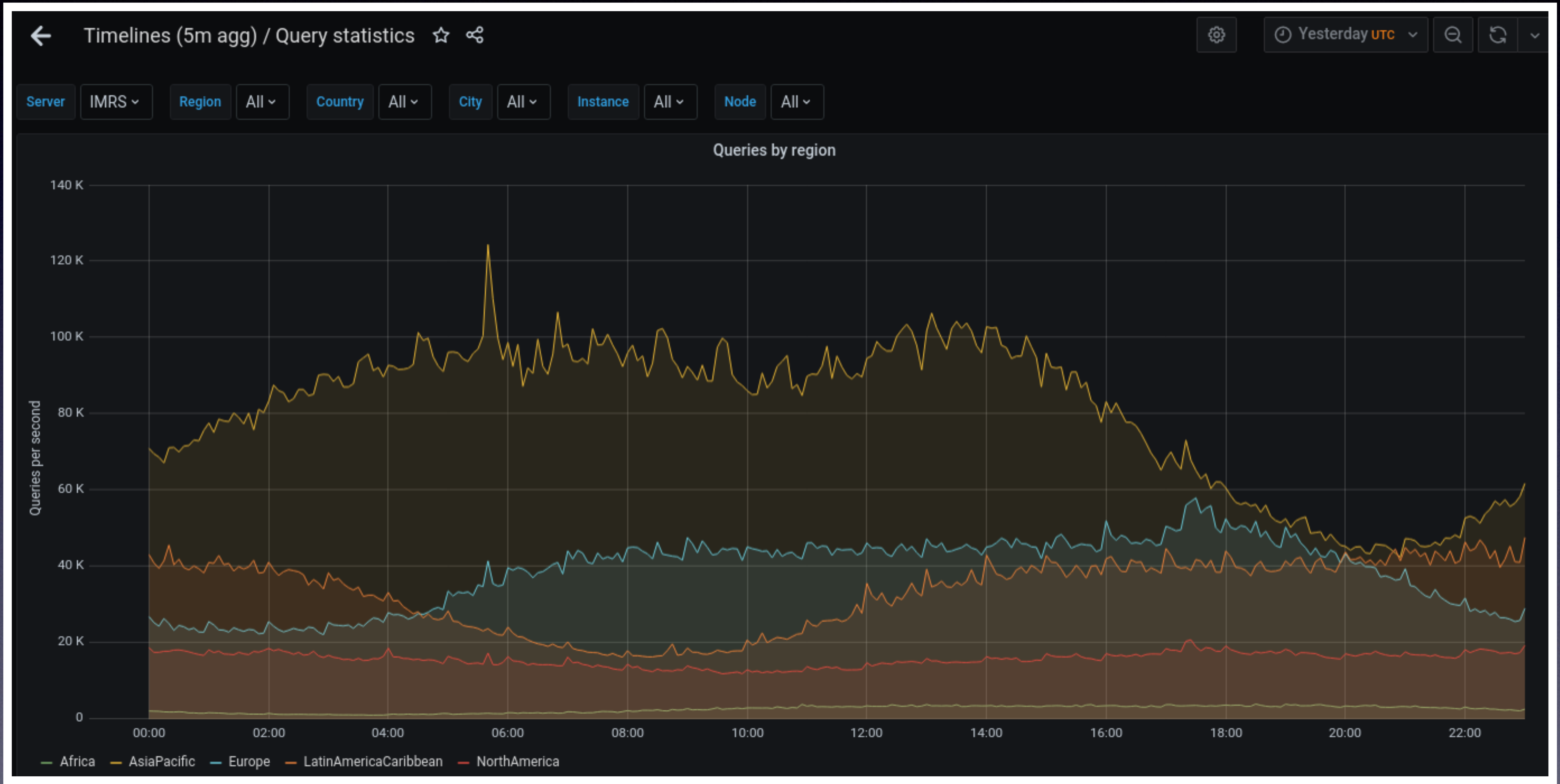
- Query statistics
- Query statistics detail
- Query attributes
- Query types (QTYPE)
- IP version and Transport
- Responses (RCODE)
- Server IP address

Other metrics

- Geographic locations
- RSSAC volumes
- RSSAC sources
- RSSAC other
- Client subnet statistics
- Client subnet statistics detail
- QTYPE vs TLD

Timeseries graph

Query Statistics



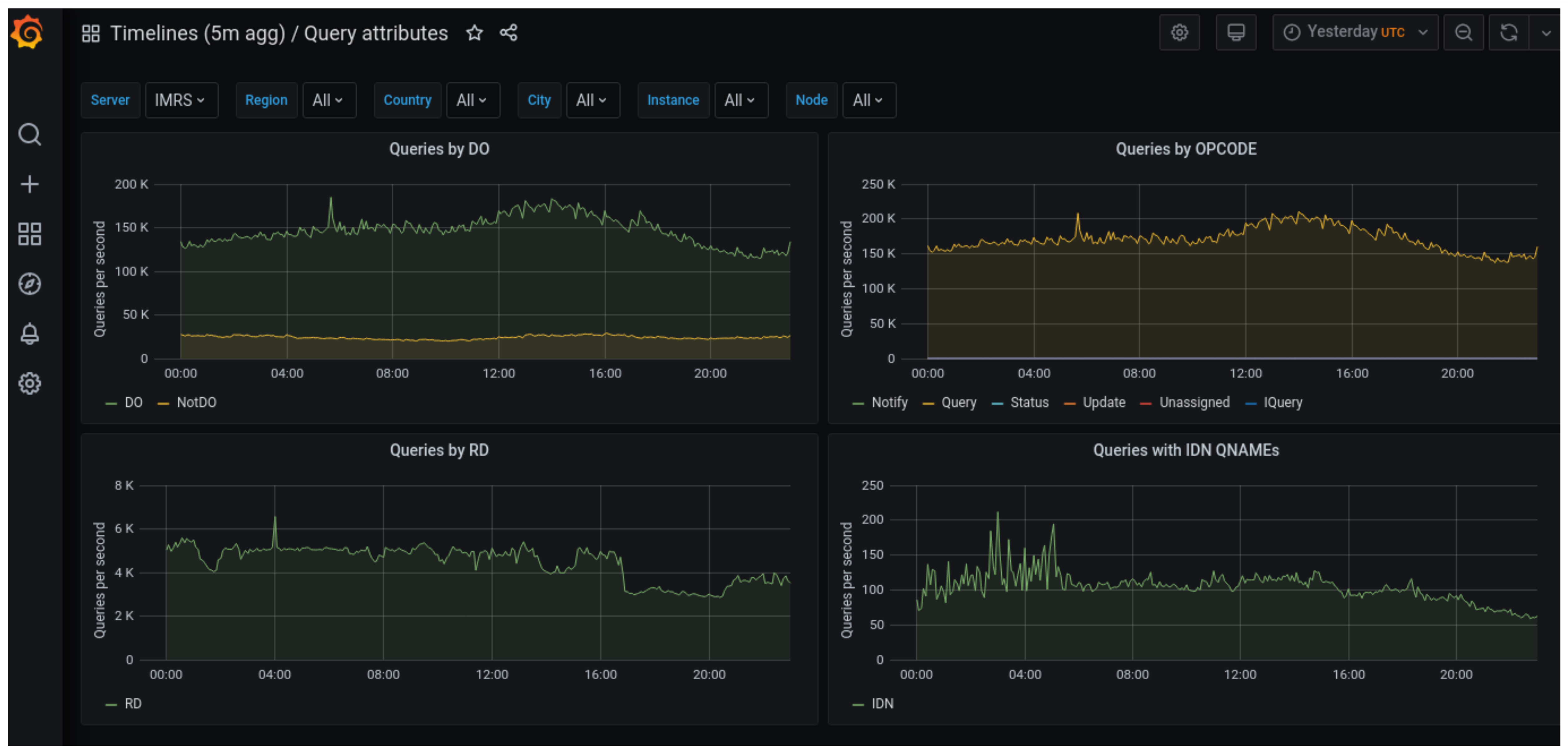
Timelines (5m agg) / Query statistics

Server: IMRS · Region: All · Country: All · City: All · Instance: All · Node: All

Yesterday UTC

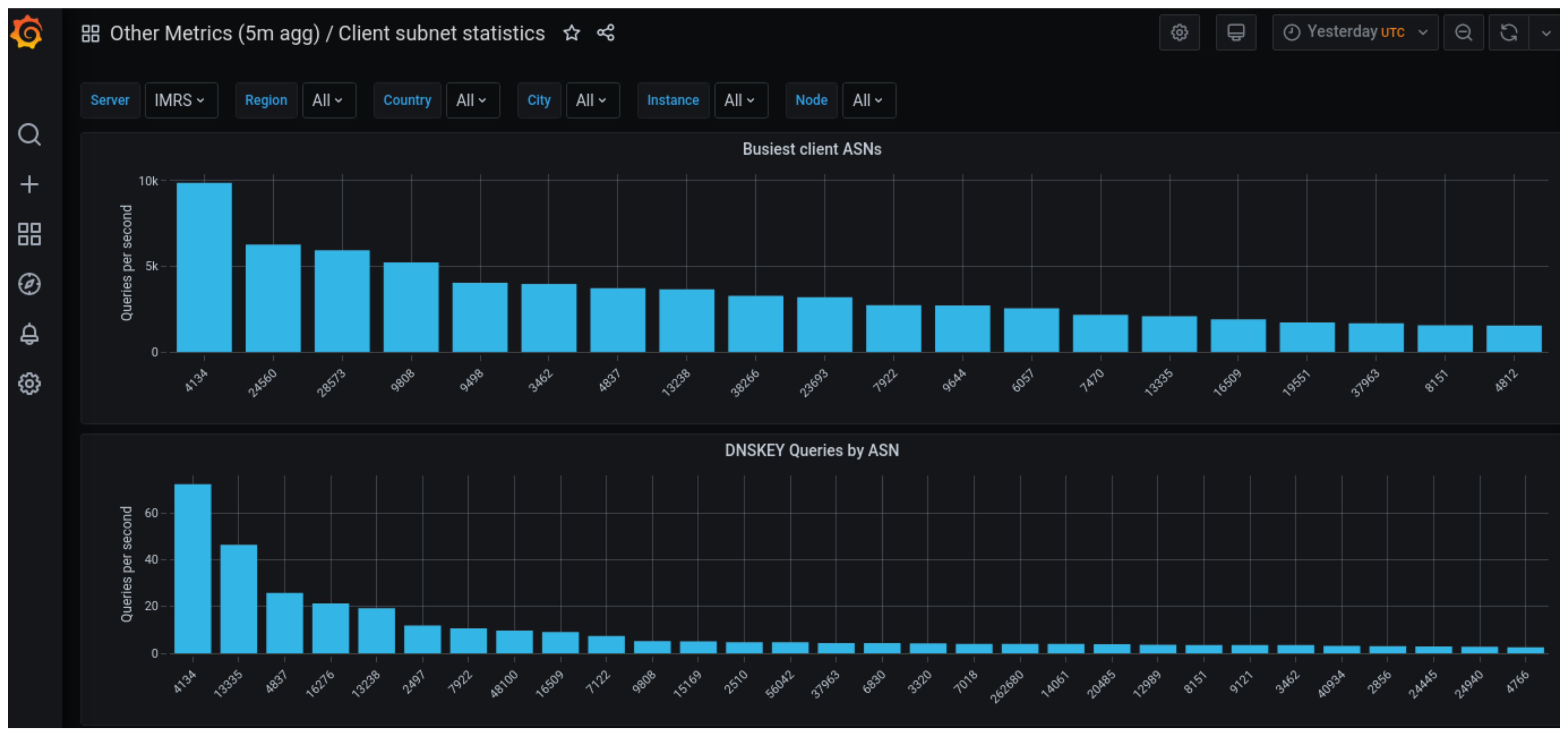
Timeseries graphs

Query Attributes



Simple bar chart

Client subnet statistics



Other Metrics (5m agg) / Client subnet statistics

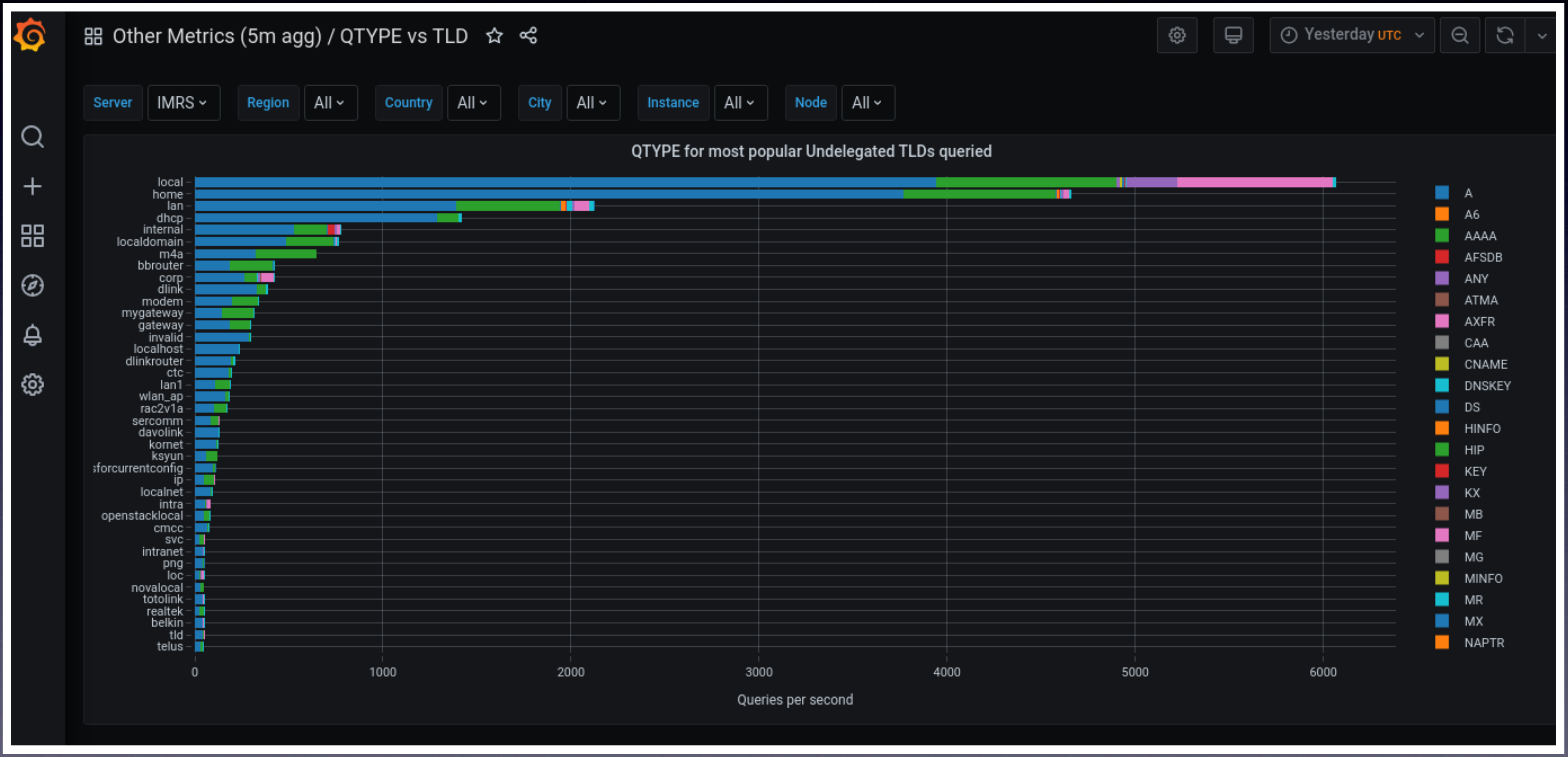
Yesterday UTC

Server: IMRS | Region: All | Country: All | City: All | Instance: All | Node: All

inspector template output modifiers provide geo location and ASN lookup with MaxMind GeoLite data

More complex bar chart

QTYPE vs TLD



Other Metrics (5m agg) / QTYPE vs TLD

Yesterday UTC

Server: IMRS

Region: All

Country: All

City: All

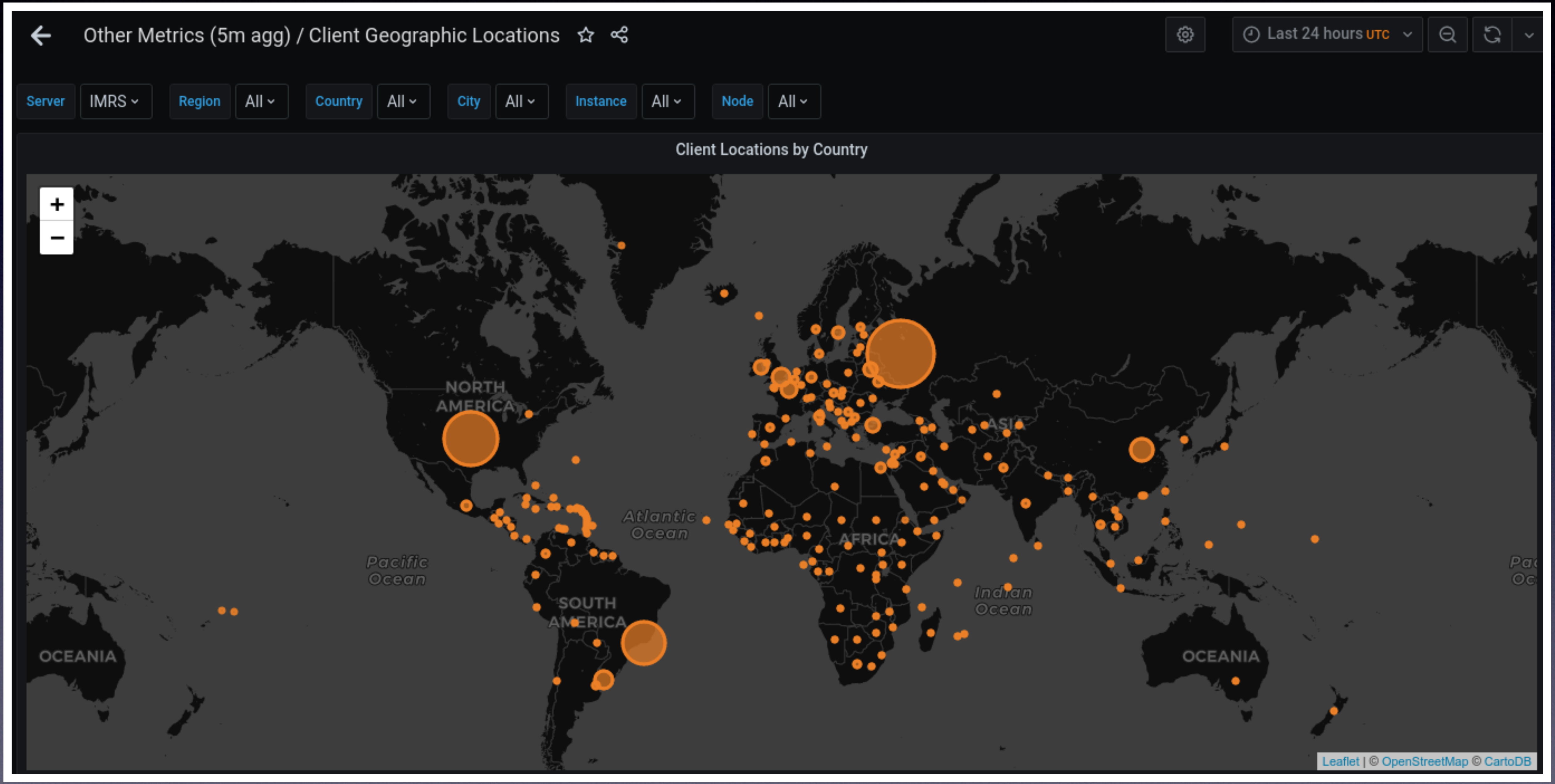
Instance: All

Node: All

Using Sinodun modified plugin based on Plotly

Map based graph

Client geographic locations



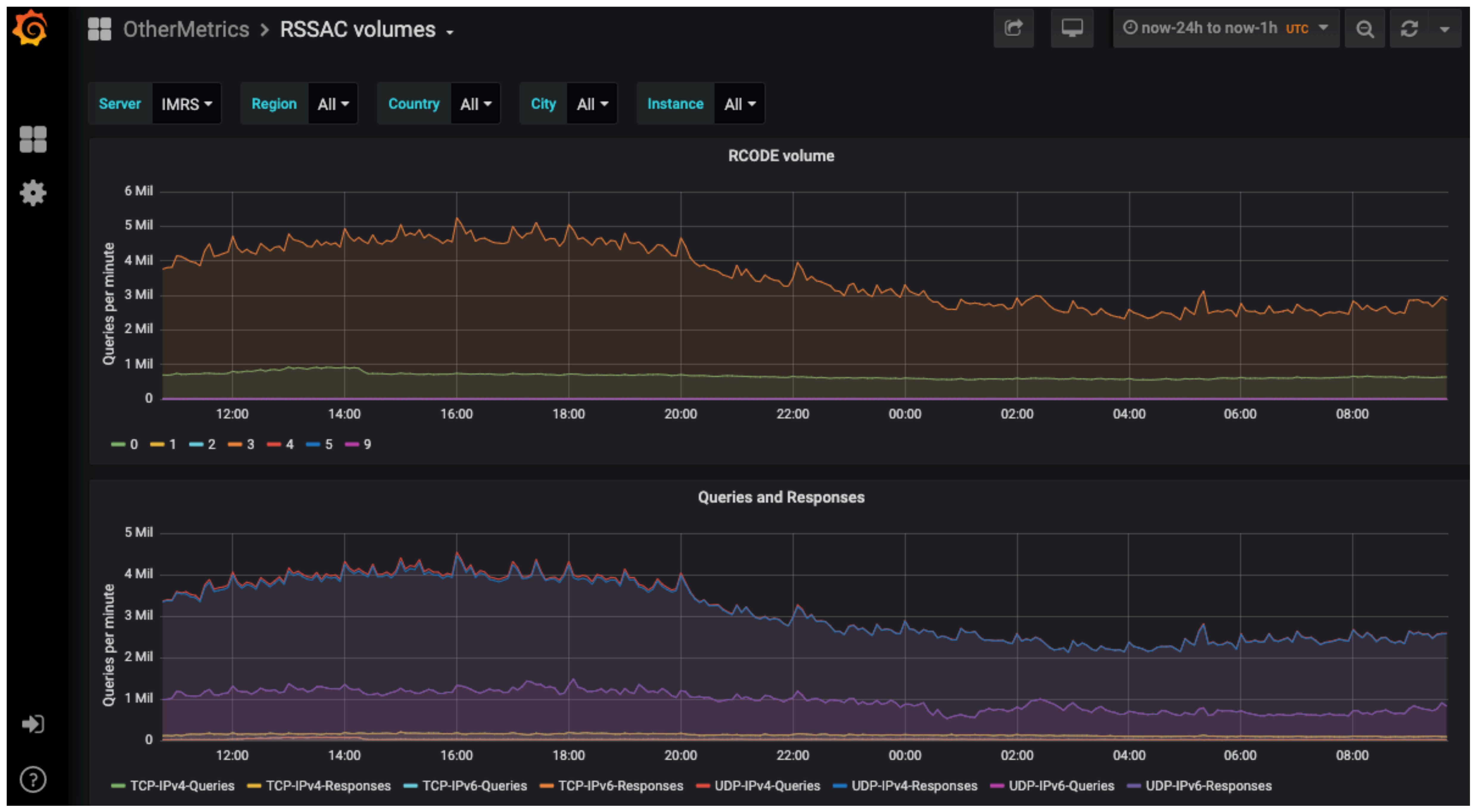
Other Metrics (5m agg) / Client Geographic Locations

Last 24 hours UTC

Server IMRS · Region All · Country All · City All · Instance All · Node All

Client Locations by Country

RSSAC graphs



RSSAC reports generated by management tools

Summary

- DNS-STATS: C-DNS, ClickHouse and Grafana provide nice package for traffic capture and visualisation
- C-DNS now supports DNSTAP input
- ClickHouse aggregations allows for flexibility to choose trade-offs between storage and performance
- Grafana can reproduce DSC like graphs with the right plugins!
- More at dns-stats.org

Thank you!

Any questions?

Offline questions to either

- SaNE (noc@dns.icann.org) or
- Sinodun (info@sinodun.com)