Scaling up: How we made millions of domains happier

Tom Arnfeld, DNS Engineer
Pavel Odintsov, DNS Engineer

Cloudflare DNS Services

CLOUDFLARE

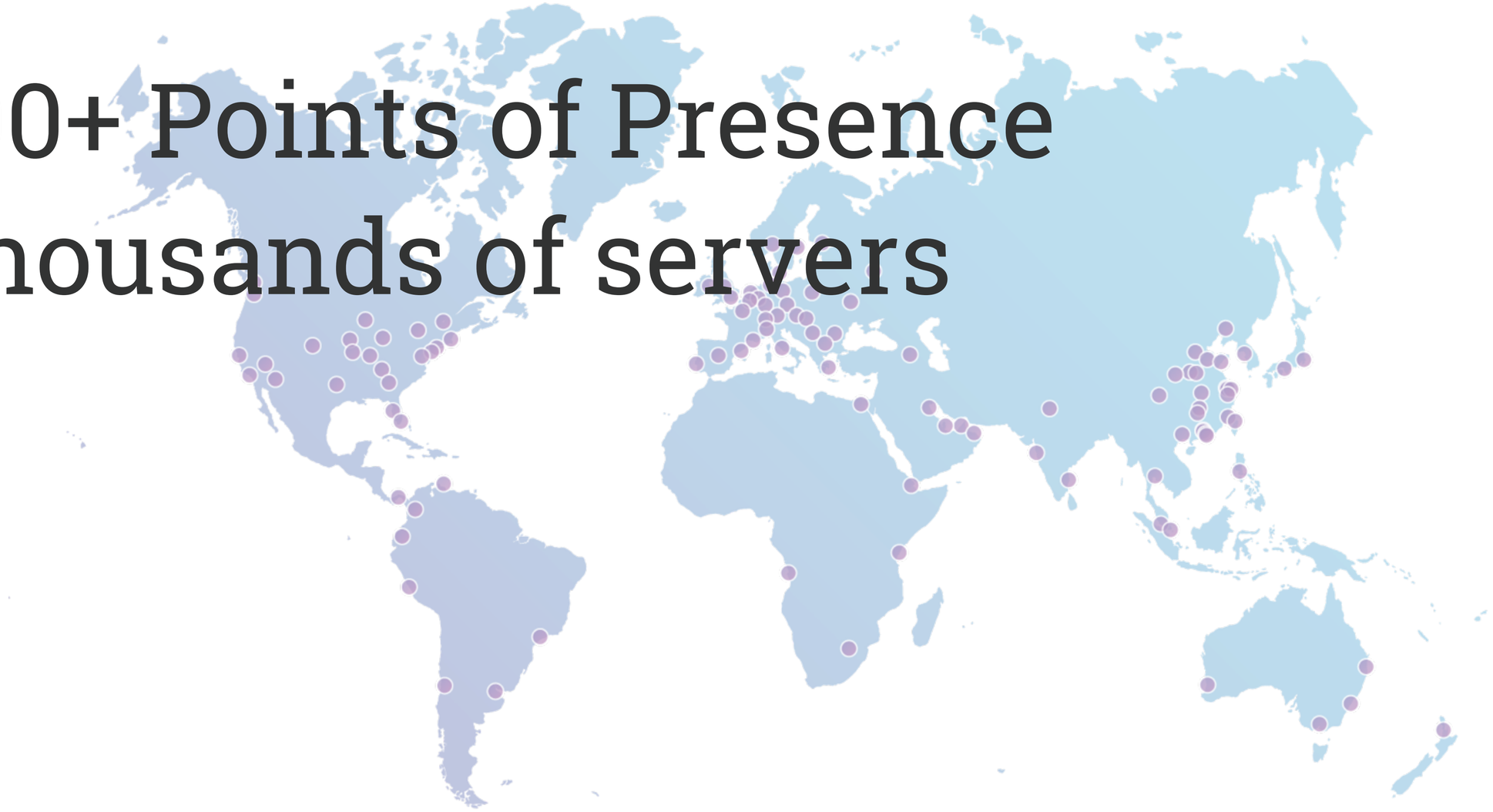
- Authoritative DNS
  - IPv4 and IPv6 support
  - Low latency propagation
  - DNSSEC on the fly signing
  - CNAME flattening
  - Load balancing
- DNS Firewall (former VDNS), DNS proxy with bundled security
- Rich Analytics

Cloudflare DNS Scale

CLOUDFLARE

100+ Points of Presence

Thousands of servers



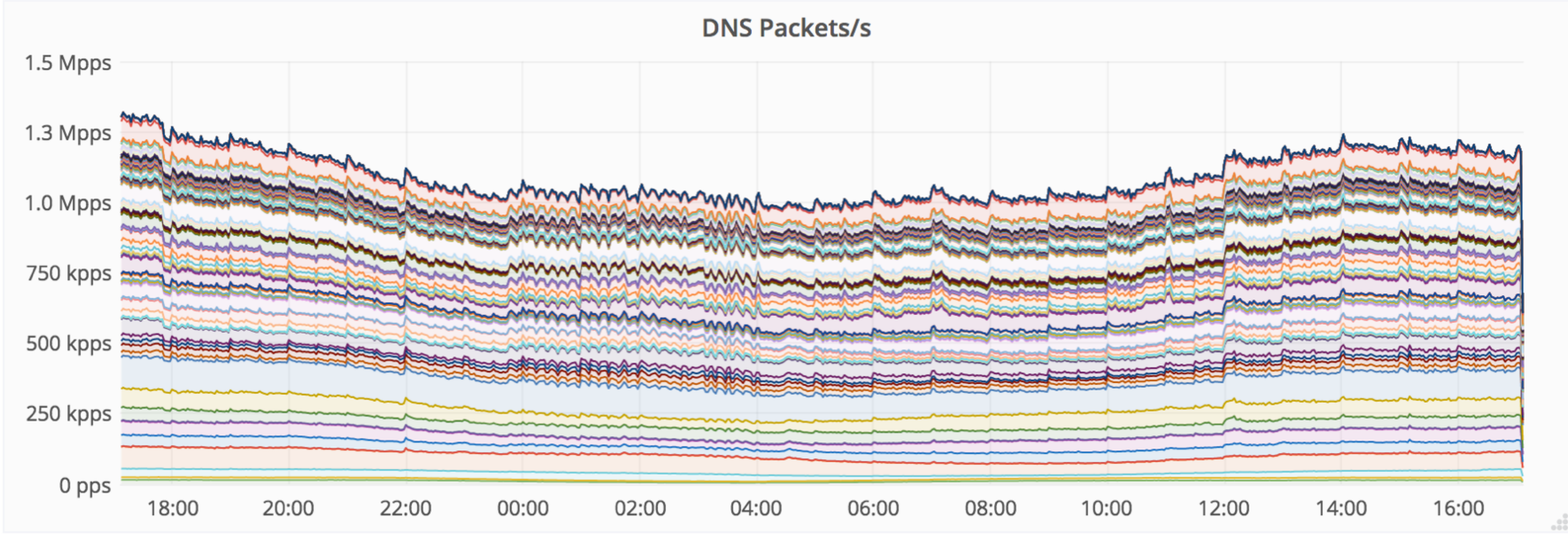
Cloudflare DNS Scale

5.5m+ DNS Zones

Cloudflare DNS Scale

CLOUDFLARE

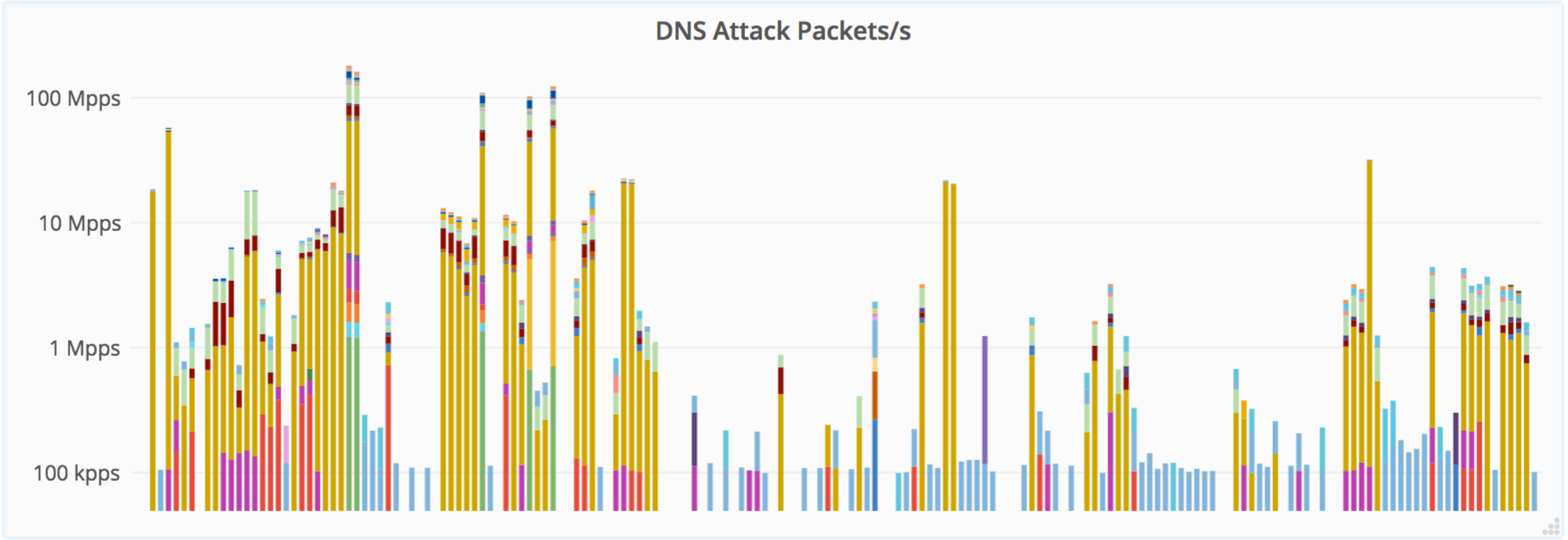
~1M QPS Throughput



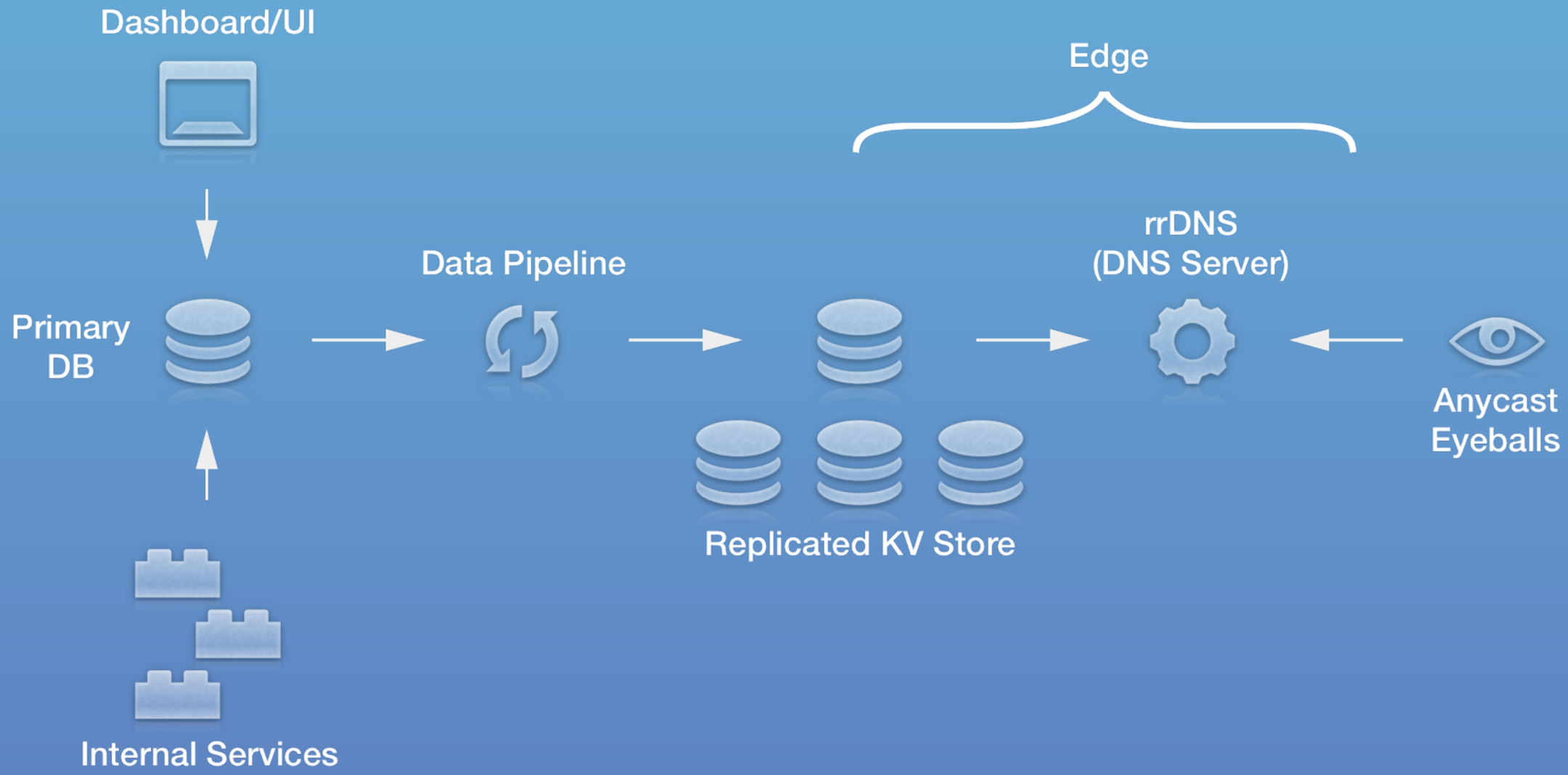
Cloudflare DNS Scale

CLOUDFLARE

O(100M) QPS **Attack** Throughput



Cloudflare DNS Architecture



rrDNS

CLOUDFLARE

What is rrDNS?

CLOUDFLARE

- Written in Golang
- Powered by Miek Gieben's library
- Modular design for different query paths
- Detailed statistics and logging for queries and responses
- Uses a custom globally replicated key-value store

Issues with rrDNS v1

CLOUDFLARE

- Poor architecture (no strict modularity)
- Bloated and slow storage encoding (protobuf)
- Everything stored as separate RR’s
- Poor performance
- Unpredictable behaviour in certain complex cases
- Hard to maintain

Goals for rrDNS v2 *business*

CLOUDFLARE

- Scale to serve >= 100 million zones
- Better query performance
- Easier development of new features / products
- No single switch when migrating to v2

Goals for rrDNS v2 *technical*

CLOUDFLARE

- More scalable data encoding
- Improve correctness against DNS RFC's
- Optimise based on experience and numbers
- Predictable and documented business logic
- Support for arbitrary RR types (only require changes in our UI)
- Considerably more maintainable, testable and well documented code

rrDNS v2 *technical decisions*

CLOUDFLARE

- Completely reworked design
- Totally new, optimised storage and encoding
- MessagePack (binary json encoding) as new storage encoding
- Faster paths for common query types (A, AAAA, NS)
- Predictable behaviour with >95% test coverage

rrDNS v2 *achievements*

CLOUDFLARE

- Increased max throughput by 3-4x
- Improved correctness against DNS Clear RFCs
- Mixed-mode DNS for a smooth, safe migration
- We love this code

rrDNS v2 *RFC correctness*

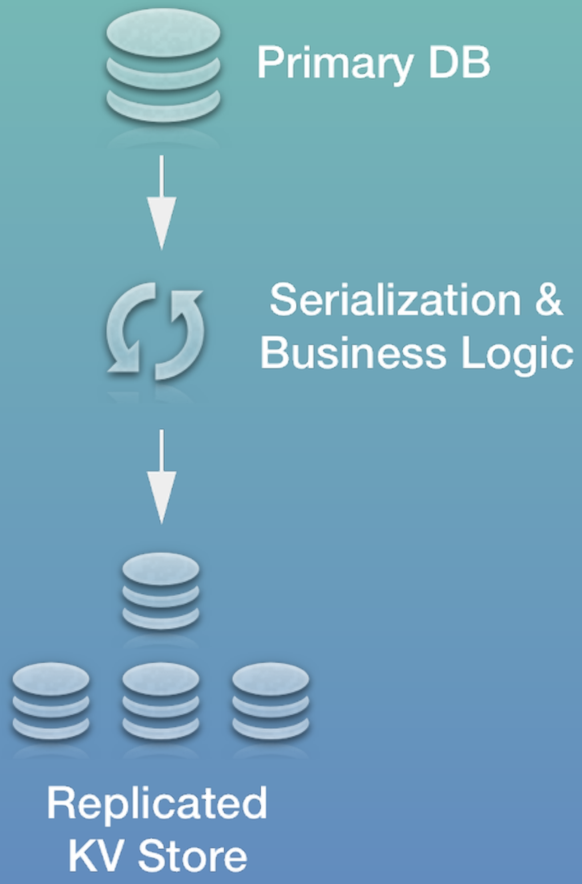
CLOUDFLARE

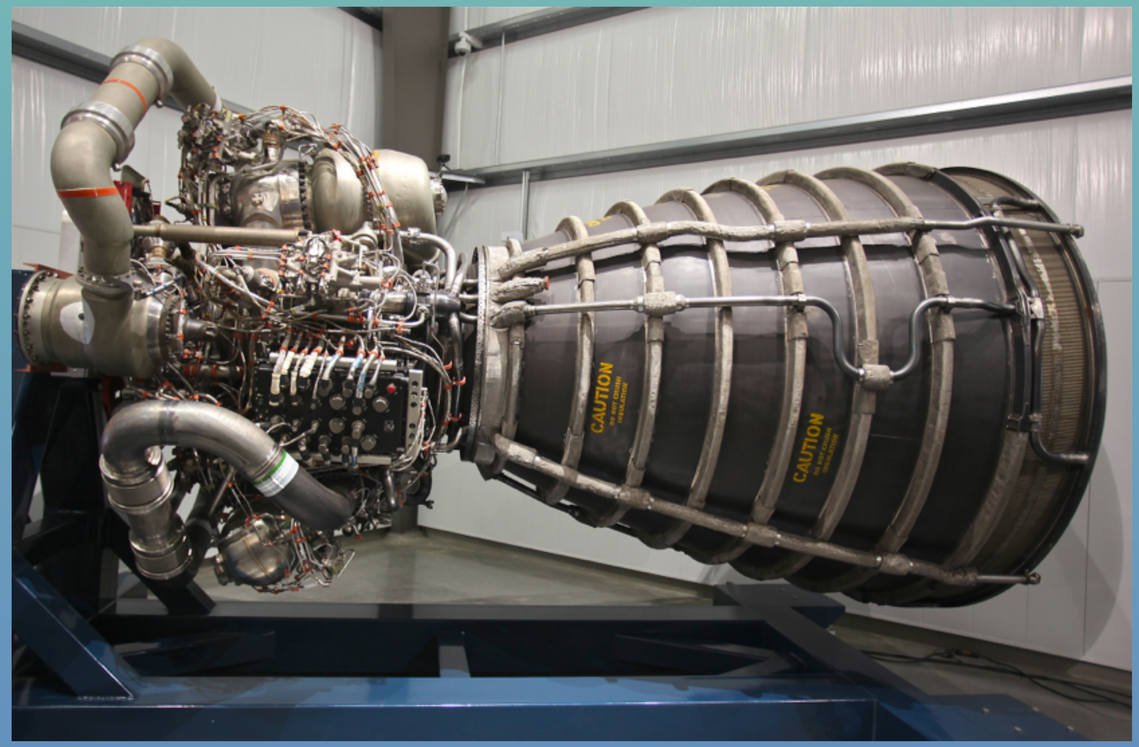
- Fixed DS query for unsigned delegation, DVE-2017-0008
  https://github.com/DNS-OARC/dns-violations/blob/master/2017/DVE-2017-0008.md
- Do not return authority flag for delegations
- Do not return useless glue records

DNS Data Pipeline

CLOUDFLARE

What is our Data Pipeline?





Data Pipeline *v1*

CLOUDFLARE

- Extremely complicated and ancient design
- Written in PHP/C/C++/bash
- No documentation or test coverage
- Very poor performance, especially for larger zones
- Impossible to offer guaranteed processing time
  - Falls over with large bursts of changes

Data Pipeline *v2*

CLOUDFLARE

- More efficient encoding/storage
- Drastically improve performance
- Guarantee low propagation time for changes
    - Even for zones with millions of records
- Predictable behaviour
- Handle bursts of changes gracefully

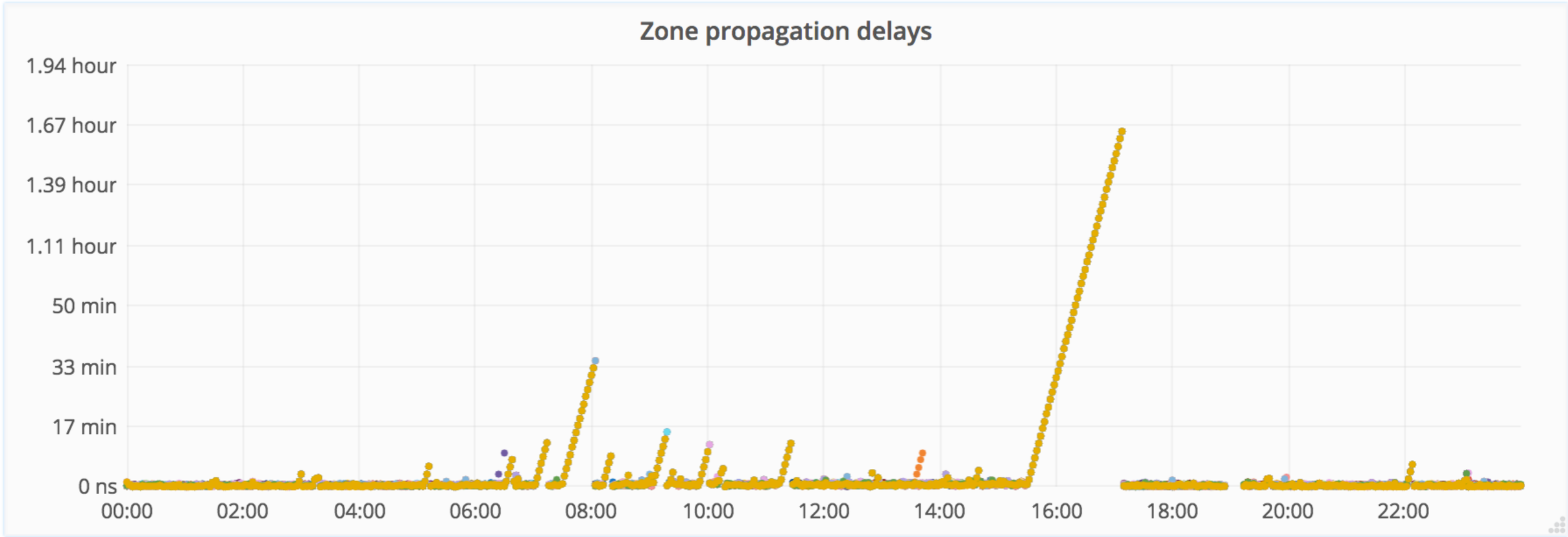
Data Pipeline *v2 - what did we get?*

CLOUDFLARE

- Written in pure Go
- Scales linearly with number of CPU cores
- ~95% test coverage
- Extremely well documented
- Lots of new debugging tools for SREs/Support/Operations
- Guaranteed propagation time
- 10x performance increase

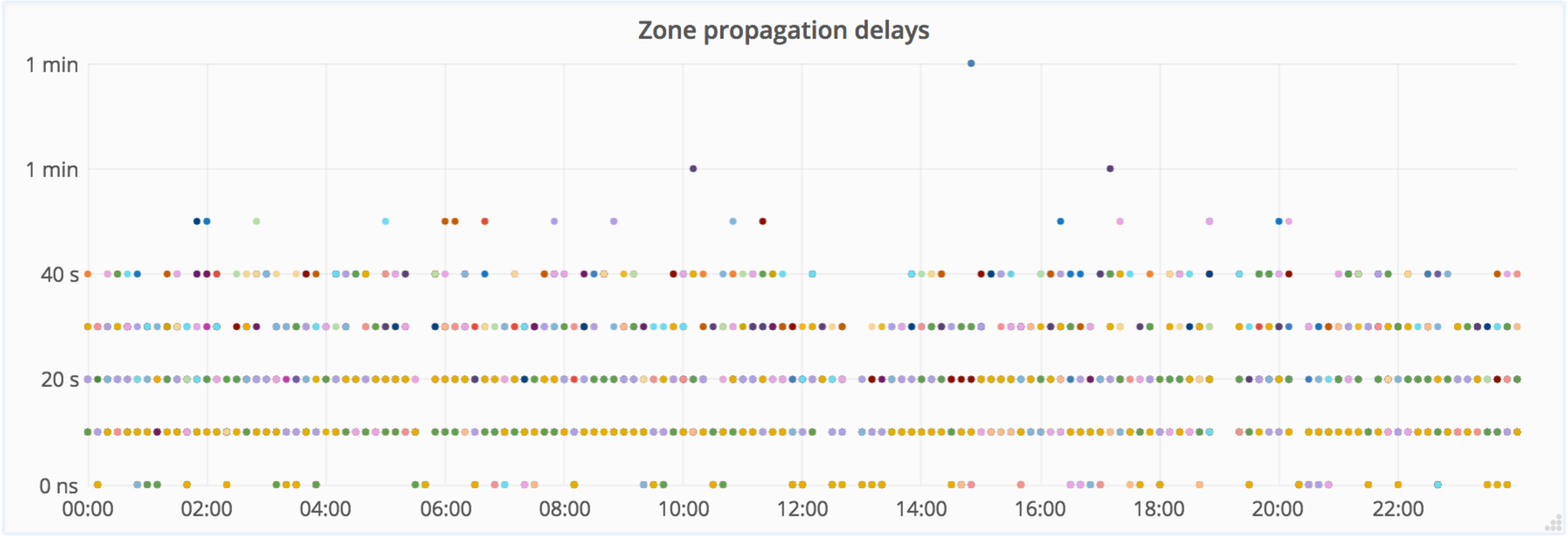
Monitoring Propagation *before*

CLOUDFLARE



Monitoring Propagation *after*

CLOUDFLARE



CLOUDFLARE

Migrating v1 → v2

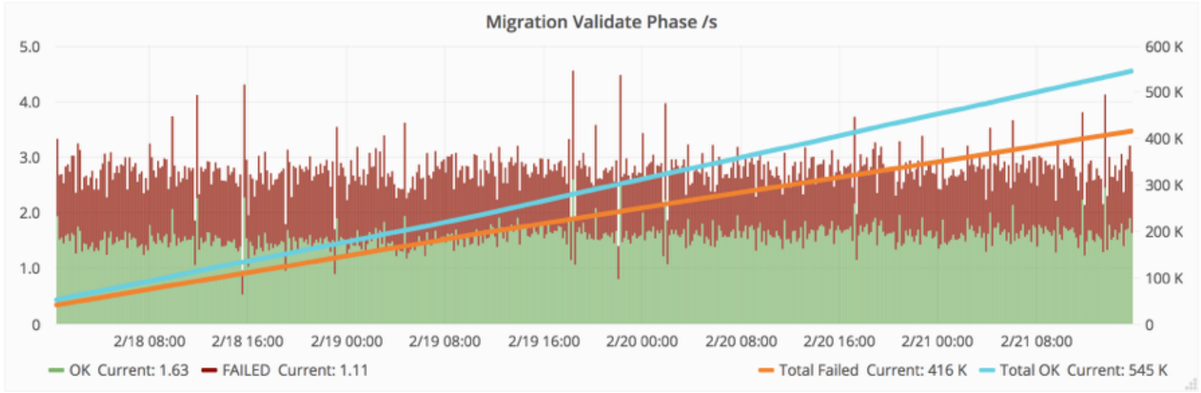
How do we know it works?

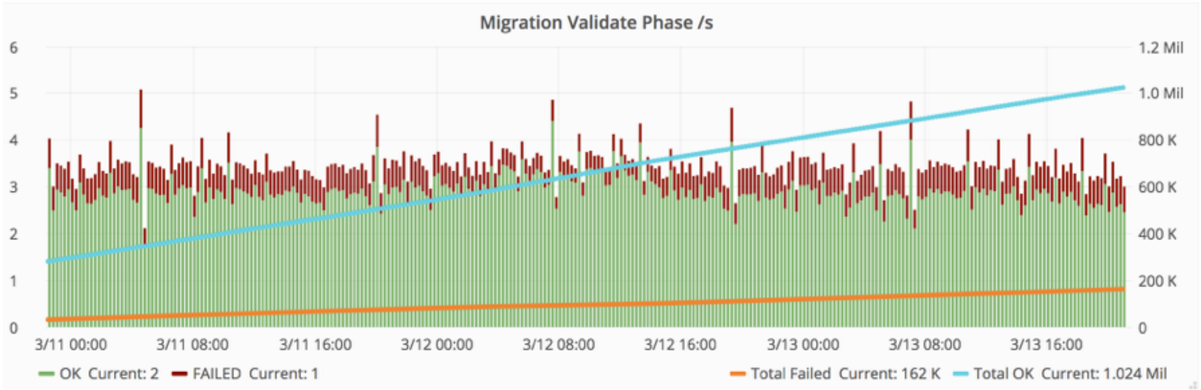
CLOUDFLARE

- **DNS mirror** Feed a sample of production queries to both old and new and compare responses
- **Full comparison** Iterate over all zones and records to compare answers from v1 and v2, ensuring a “safe” migration
- **Tests** Cover every new discovered case with a test, *religiously*
- **Smart Comparisons** Wrote “smart comparators” to filter acceptable differences, because of bad behaviour in v1

Migration Process

- Tools to perform validation and migration of each zone automatically
- Treated like a prod. service meaning metrics, logging etc.
- Many iterations fixing bugs to improve the validation pass rate





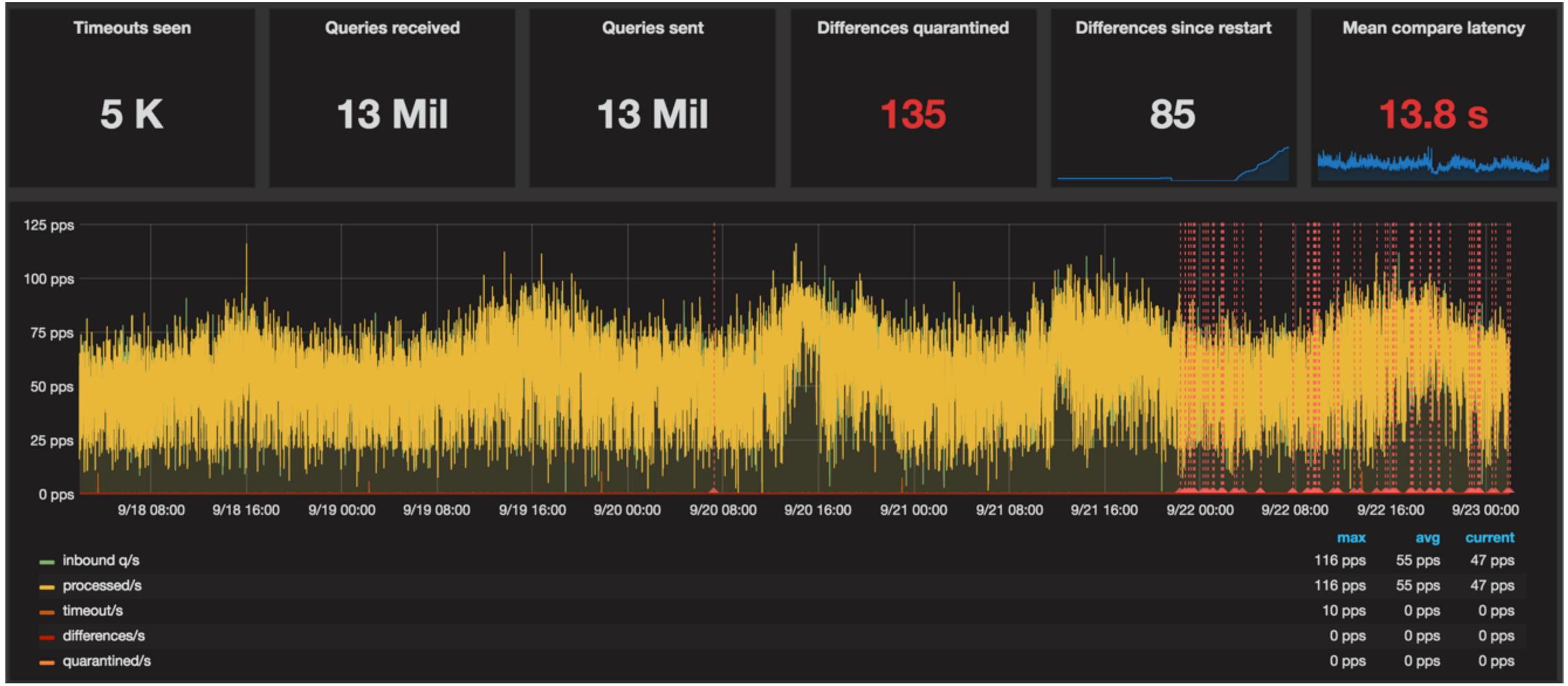
DNS Mirror

CLOUDFLARE

- Samples real DNS traffic from entire Cloudflare network
- Compares responses between multiple instances of rrDNS
- Tools for debugging detected differences
- Part of every rrDNS release to minimise regressions

DNS Mirror

CLOUDFLARE



Migrating to v2

CLOUDFLARE

- Total consumed time (development) **~1 year**
- Time consumed by migration **< 3 months**
- Number of new test cases **183**
- Number of Pull Requests **250+**
- Number of rrDNS releases **~50 (about one per week)**
- Total number of working hours consumed by project **~3 years**
- Percent of zones with noticeable impact **0.005%**

Thank you.

Questions?

DNS is hard!

https://blog.cloudflare.com/how-we-made-our-dns-stack-3x-faster/