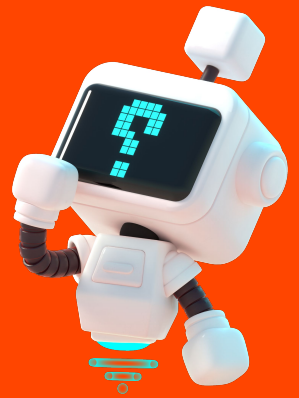
reddit



How We Protect Cat Memes from DDoS

Spencer Koch
Principal Security Engineer

Pratik Lotia
Senior Security Engineer

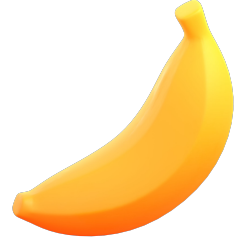
reddit

Context Time

What You'll Get From This Talk

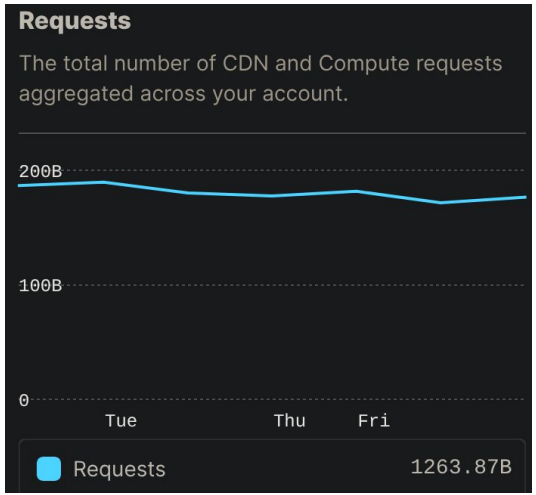
- How Reddit thinks about ratelimiting architecture
- Signals and their importance
- Where to do ratelimiting
- Other resiliency strategies
- Lessons learned from 6 years of DDoS fights

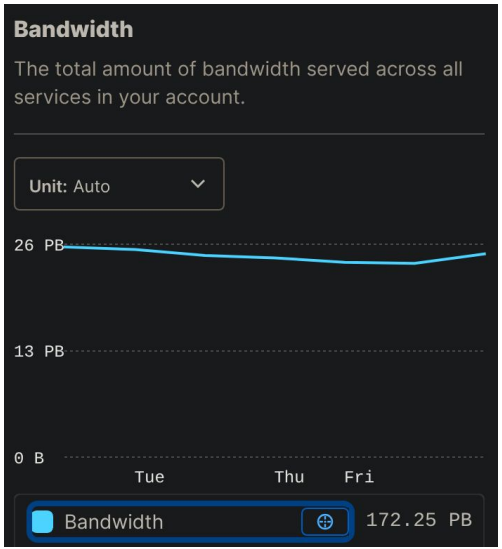
Reddit's Kind of a Big Deal...



Per week: 1.3 Trillion requests, 175 Petabytes (83% cached)

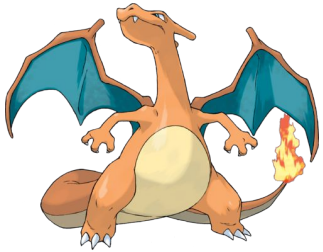
Service	RPS
GraphQL	275k
Ads Pixel	100k+
Server Side Rendering	150k
Media	1200k





Where the DDoS Crew Came From...

2019 - evolved from a couple of OG engineers duct taping things together



2021 - our first Transport team, getting more serious about defenses

2025 - now both Transport and Traffic teams; dedicated engineers focused on DDoS mitigations and response

reddit

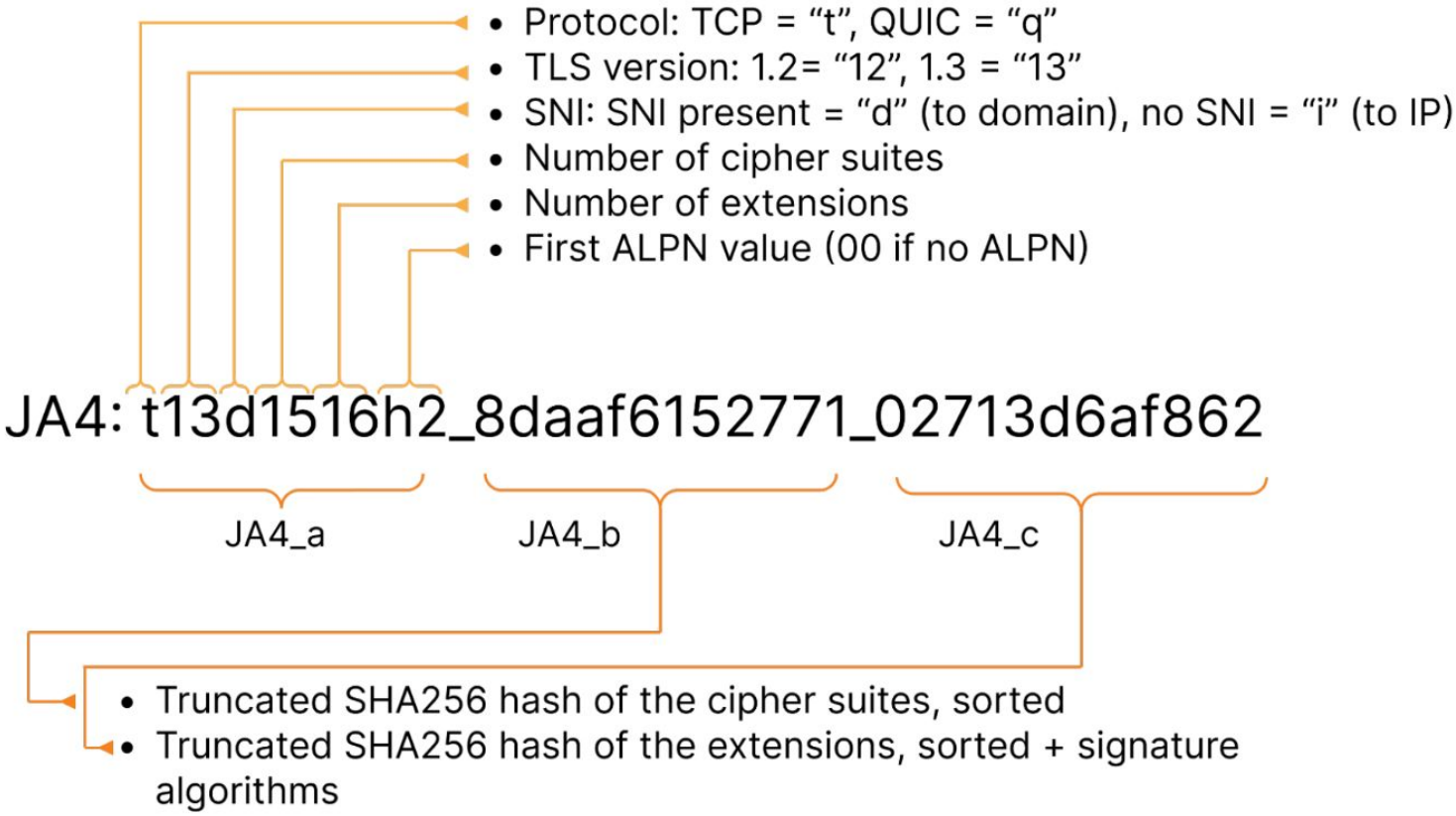
Signals

Signals - Foundational

- **Foundational**:
  - GeoIP+
  - User Agent
  - Path, Query Params
- **TLS Fingerprints**:
  - JA3 / JA4 Calculations
  - GREASE - incompatibility vs randomness
  - Protocol sprawl

https://browserleaks.com/tls - for more info
https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967/ - JA3 details
https://github.com/FoxIO-LLC/ja4/blob/main/technical_details/JA4.md - JA4 details

JA4: TLS Client Fingerprint



https://blog.cloudflare.com/ja4-signals/

Signals - Intermediate

- **Request Header Fingerprints**
  - Order, casing, and presence of headers
  - Must be done at receiving proxy before munging
  - Fastly - *`fastly_info.oh_fingerprint`*
  - Cloudflare - *`request.cf?.requestHeaderNames`*

https://privacycheck.sec.lrz.de/passive/fp_hs/fp_header_signature.php - RHS info

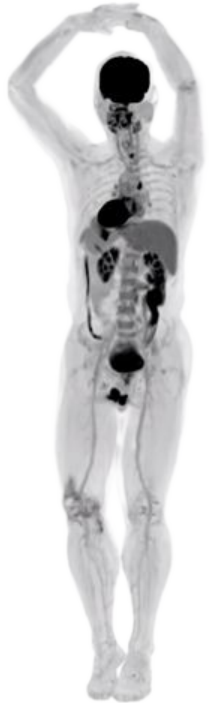
Signals - Intermediate

```
Request Example (Chrome)
========================
    Host: reddit.local
    X-Forwarded-For: 192.168.56.1
    Connection: close
    Content-Length: 295
    Accept: application/json, text/javascript, */*; q=0.01
    Origin: https://reddit.local
    X-Requested-With: XMLHttpRequest
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Ge
    Content-Type: application/x-www-form-urlencoded; charset=UTF-8
    Referer: https://reddit.local/new/
    Accept-Encoding: gzip, deflate
    Accept-Language: en-US,en;q=0.8
    Cookie: loid=ajOuPPBsarNtdho6iK; loidcreated=2016-03-03T05%3A09%3A08.565Z; reddit_session=53%2
    DNT: 1
    X-Forwarded-Proto: https
```

```
request_header_signature': 'Am:GX:K6:Ac:AT:M9:Gj:A6:Af:Az:AQ:AR:Ag:Kv:CB
```

Signals - Advanced

- **Behavioral Fingerprinting**
  - Network request timings
  - Analytical events fired or missing
  - Cookies, JWT presence and validation
  - Login flow or “expected” order of operations
  - CAPTCHA or “someone else’s model”



reddit

Where to Ratelimit

Where to Ratelimit?

- Leave Layer 3 / 4 attacks to internet / cloud providers
- Difference between Edge and Application layer
  - **Edge**:
    - Cheaper, but lacks context of the app
    - User session, login state, limited business logic
  - **Application**:
    - More sophisticated logic and enriched data
    - Expensive as your compute does the work

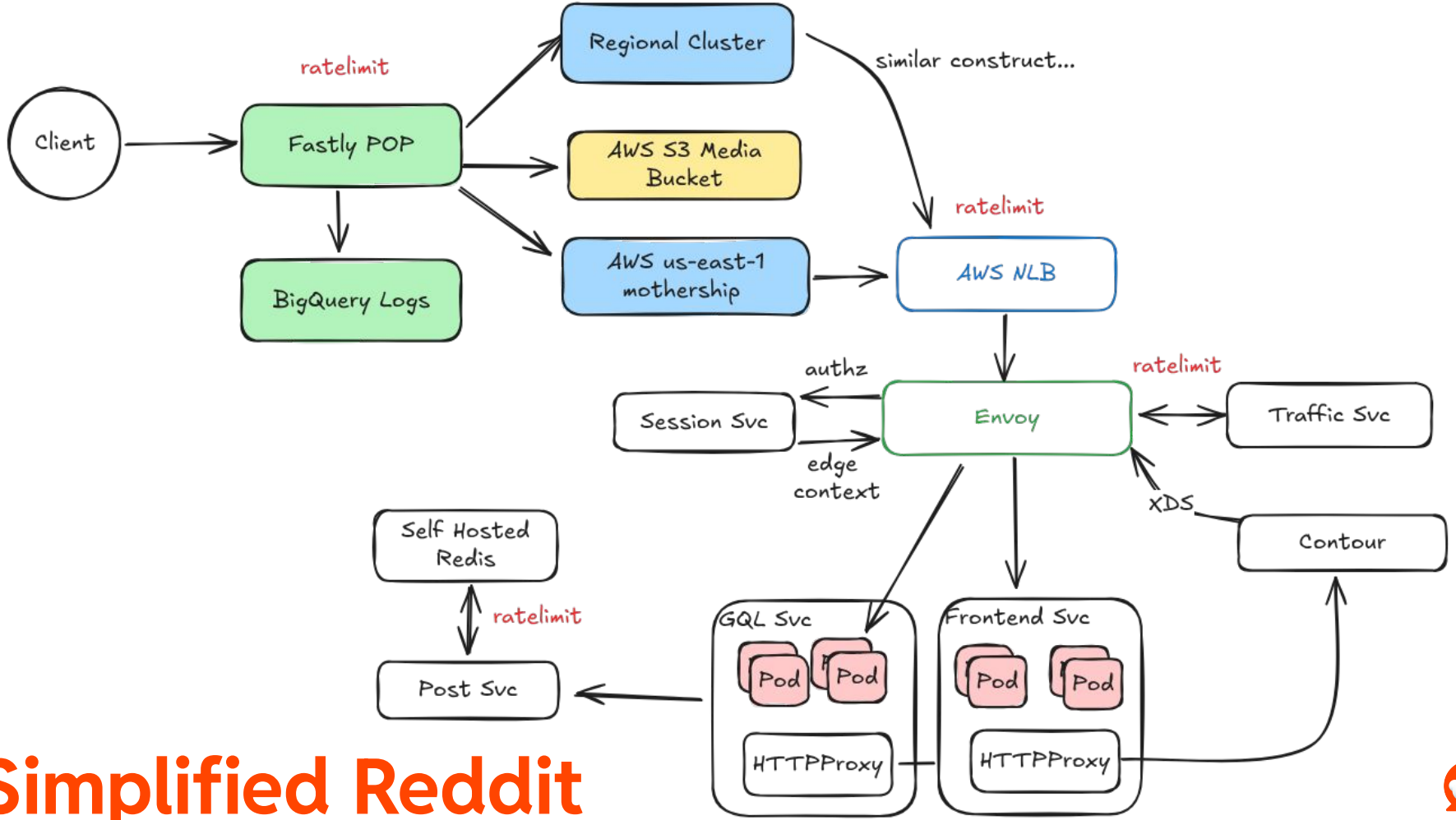
Where to Ratelimit?

- Both need a KV store
  - Do you need more control over behavior?
  - Difficulty of flushing / unsetting?
  - Programmatic interactions?
- How to handle eventual consistency?
  - Consistent routing based on geo / session / stickiness
  - Circular hash ring algorithms

https://dev.to/mochafreddo/nginx-generic-hash-load-balancing-a-comprehensive-guide-3bca - nginx hashing primer

Where to Ratelimit?

- Routing architecture principles:
  - “Reasonable” domain carve outs: media vs app split, chat vs main site, modmail
  - Cache where you can
  - Isolate workloads with specific purpose
- Gotta use both!



Simplified Reddit

reddit

Edge Ratelimiting

Edge Ratelimiting

- **Fastly's Edge Rate Limiting**
  - Two KV stores:
    - `penaltybox`: a TTL for naughty elements
    - `ratecounter`: counter for observed hits
  - You determine what element to track (IP, TLS, etc.)
  - Edge dictionaries to control configurables (request rate, exclusions)

https://www.fastly.com/documentation/guides/concepts/rate-limiting - Fastly Edge Rate Limiter (ERL)

```
sub ip_rate_limit_run {
  declare local var.ip_ratelimit_exceeded BOOL;
  declare local var.window INTEGER;
  declare local var.window_limit INTEGER;
  declare local var.increment INTEGER;
  declare local var.key STRING;

  # Defaults (for comparison) with the TLS rate limiter.
  set var.key = req.http.CDN-Client-IP;
  set var.increment = 1;
  set var.window = 10;
  set var.window_limit = 15;

  set var.ip_ratelimit_exceeded = ratelimit.check_rate(
    var.key,
    ip_counter_10s,
    var.increment,     # Increment.
    var.window,        # Sample window (in seconds)
    var.window_limit,  # Limit in RPS
    ip_pbox,
    2m);
```

Example Fastly Construction

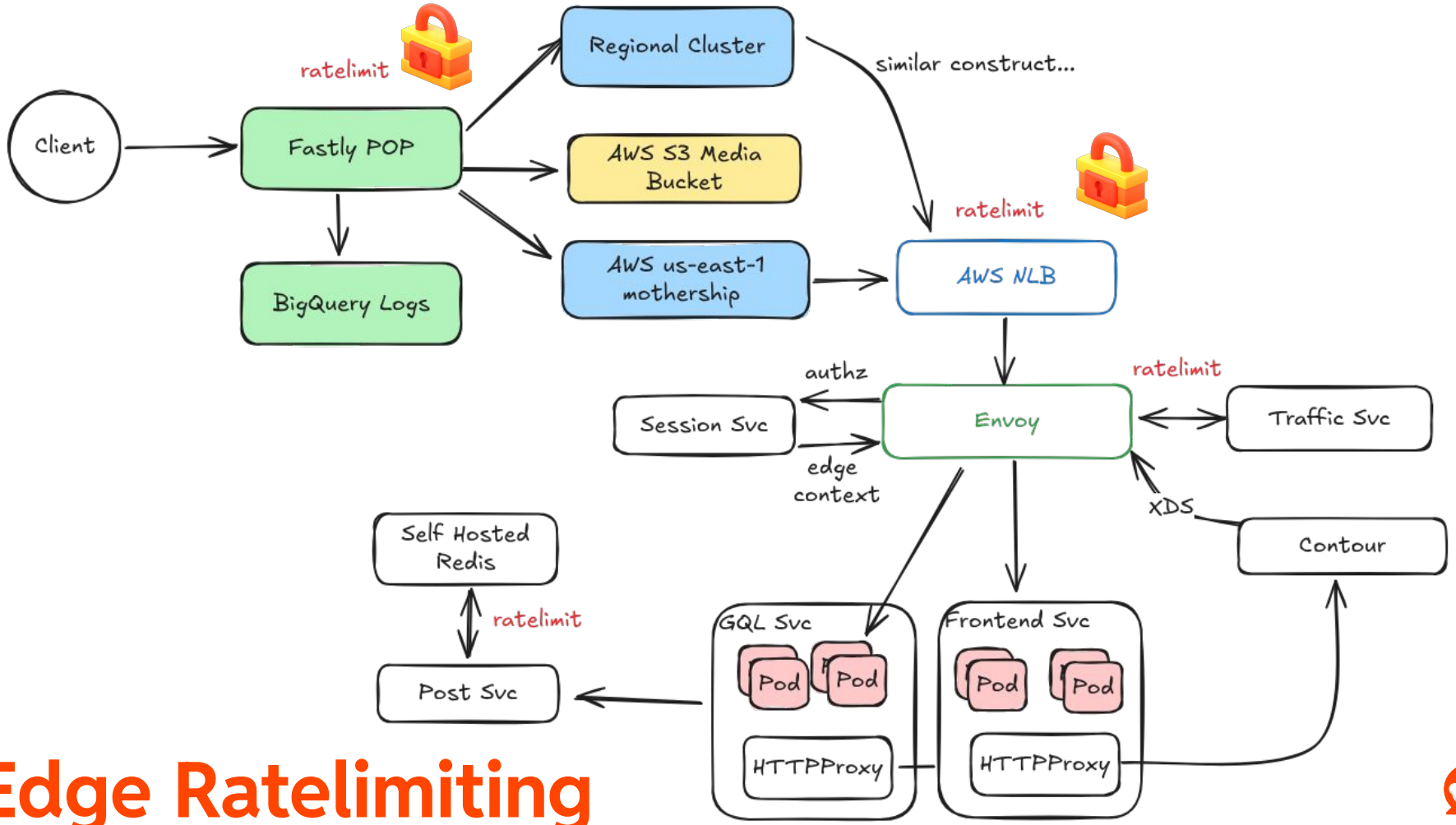
Edge Ratelimiting

- **Cloudflare**
  - WAF ratelimit rules - precanned and Enterprise only
  - Workers - use a KV and do it yourself
  - Don’t forget your logging
    - `X-reddit-synthetic-code` & `x-reddit-synthetic-reason`
    - 429 is extremely terse signal to engineers

https://developers.cloudflare.com/waf/rate-limiting-rules/ - WAF rules

Edge Ratelimiting

- **Series of band-pass filters** with decreasing rate values: IPs→TLS→pools→recaptcha
- **Weaknesses at Edge**:
  - CGNAT, shared IP blocks, TLS by user agent
  - Requires tuning - approved crawlers, third parties
  - Country+ASN reputation scoring or crowd-level heuristics? Depends on your business model
  - Collateral damage will happen! Revert fast



Edge Ratelimiting

reddit

Application Ratelimiting

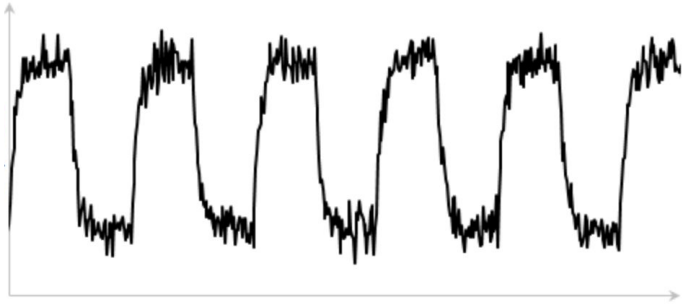
Application Ratelimiting

- FIRST: **Common library for ratelimiting**
  - Pick an algorithm and provide guidance to your devs!
  - Sliding window algorithm with Redis TTL
  - Redis - common, cheap, make it easy for devs to use
  - “Allowance” and “interval” selection:

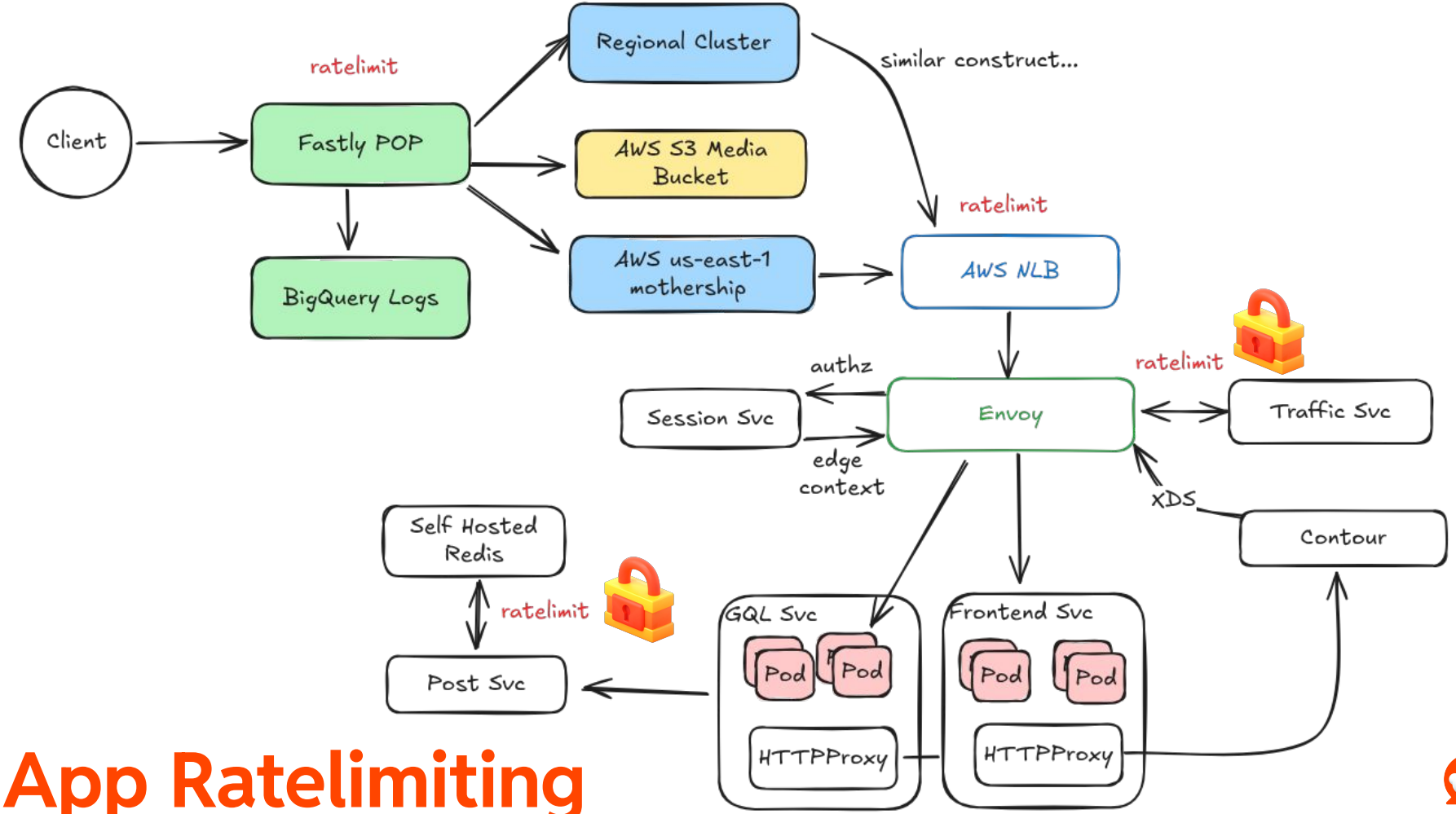
(how fast could a human do it) x (some reasonable multiplier)

https://github.com/reddit/baseplate.py/blob/develop/baseplate/lib/ratelimit/ratelimit.py - Reddit’s implementation

Application Ratelimiting



- Best for:
  - “Per user, per endpoint” logic
  - OAuth ratelimits per client ID
- API route handling (front) vs datastore execution (back)?
- Square wave signals for abuse detection
- World wide Reddit and global state:
  - Login / UGC endpoints - slower calls to a global state
  - Mutating endpoints - POP / regional state



App Ratelimiting

reddit

Resiliency Techniques

Resiliency Techniques

- **Get Observability!**
  - Customized web logs with request metadata for modeling / triage

```
sub generate_bq_log {
    set resp.http.json_generate_bq_json = "{ "
    if(std.strlen(req.request) > 0, "%22" + "req_request" + "%22:%22" + json.escape(req.request) + "%22", "")
    if(std.strlen(req.service_id) > 0, ",%22" + "req_service_id" + "%22:%22" + json.escape(req.service_id) + "%22", "")
    if(std.strlen(req.backend.name) > 0, ",%22" + "req_backend" + "%22:%22" + json.escape(req.backend.name) + "%22", "")
    if(std.strlen(client.ip) > 0, ",%22" + "client_ip" + "%22:%22" + json.escape(client.ip) + "%22", "")
    ",%22" + "client_asn" + "%22:" + client.as.number
    ",%22" + "req_restarts" + "%22:" + req.restarts
    if(std.strlen(server.datacenter) > 0, ",%22" + "server_datacenter" + "%22:%22" + json.escape(server.datacenter) + "%22", "")
    if(std.strlen(fastly_info.state) > 0, ",%22" + "fastly_info_state" + "%22:%22" + json.escape(fastly_info.state) + "%22", "")
    if(std.strlen(resp.response) > 0, ",%22" + "resp_response" + "%22:%22" + json.escape(resp.response) + "%22", "")
    ",%22" + "resp_status" + "%22:" + resp.status
    ",%22" + "time_elapsed_msec" + "%22:" + time.elapsed.msec
    ",%22" + "time_end_msec" + "%22:" + time.end.msec
    ",%22" + "req_bytes_read" + "%22:" + req.bytes_read
    ",%22" + "resp_bytes_written" + "%22:" + resp.bytes_written
    ",%22" + "resp_body_bytes_written" + "%22:" + resp.body_bytes_written
    if(std.strlen(req.http.Host) > 0, ",%22" + "req_http_host" + "%22:%22" + json.escape(req.http.Host) + "%22", "")
    if(std.strlen(req.url) > 0, ",%22" + "req_url" + "%22:%22" + json.escape(req.url) + "%22", "")
    if(resp.http.bq-logging-skip-field:req_url_path,"",if(std.strlen(req.url.path) > 0, ",%22" + "req_url_path" + "%22:%22" + json.escape(req.url.path) + "%22", ""))
    if(resp.http.bq-logging-skip-field:req_url_qs,"",if(std.strlen(req.url.qs) > 0, ",%22" + "req_url_qs" + "%22:%22" + json.escape(req.url.qs) + "%22", ""))
    if(std.strlen(req.xid) > 0, ",%22" + "req_xid" + "%22:%22" + json.escape(req.xid) + "%22", "")
    if(std.strlen(req.method) > 0, ",%22" + "req_method" + "%22:%22" + json.escape(req.method) + "%22", "")
    if(std.strlen(substr(req.digest, 0, 16)) > 0, ",%22" + "req_cache_digest" + "%22:%22" + json.escape(substr(req.digest, 0, 16)) + "%22", "")
    if(std.strlen(req.http.CDN-Client-IP) > 0, ",%22" + "req_http_cdn_client_ip" + "%22:%22" + json.escape(req.http.CDN-Client-IP) + "%22", "")
    if(req.http.CDN-Client-IP && req.http.CDN-Client-IP ~ "^([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)[0-9]{1,3}|^([a-f0-9:]+:)([a-f0-9]*:){4}",",%22" + "req_http_cdn_client_
    if(std.strlen(req.http.CDN-Org-Name) > 0, ",%22" + "req_http_cdn_org_name" + "%22:%22" + json.escape(req.http.CDN-Org-Name) + "%22", "")
    if(std.strlen(req.http.CDN-Proxy-Description) > 0, ",%22" + "req_http_cdn_proxy_description" + "%22:%22" + json.escape(req.http.CDN-Proxy-Description) + "%22", "")
    if(std.strlen(req.http.CDN-Proxy-Type) > 0, ",%22" + "req_http_cdn_proxy_type" + "%22:%22" + json.escape(req.http.CDN-Proxy-Type) + "%22", "")
    if(std.strlen(req.http.X-reddit-synthetic-code) > 0, ",%22" + "req_http_x_reddit_synthetic_code" + "%22:%22" + json.escape(req.http.X-reddit-synthetic-code) + "%22",
    if(std.strlen(req.http.X-reddit-synthetic-reason) > 0, ",%22" + "req_http_x_reddit_synthetic_reason" + "%22:%22" + json.escape(req.http.X-reddit-synthetic-reason) + "
    if(std.strlen(req.http.X-reddit-pool-reason) > 0, ",%22" + "req_http_x_reddit_pool_reason" + "%22:%22" + json.escape(req.http.X-reddit-pool-reason) + "%22", "")
    if(std.strlen(req.http.X-reddit-block-reason) > 0, ",%22" + "req_http_x_reddit_block_reason" + "%22:%22" + json.escape(req.http.X-reddit-block-reason) + "%22", "")
    if(req.http.X-Reddit-Compression,",%22" + "req_http_x_reddit_compression" + "%22:" + json.escape(if(std.atoi(req.http.X-Reddit-Compression) != 1,"-1", "1")),"")
```

Resiliency Techniques

- **Get Observability!**
  - Customized web logs with request metadata for modeling / triage
  - Unique UUIDs on each request
  - Invest in tooling: Fastly terraform module for logging, `tportctl`

Resiliency Techniques

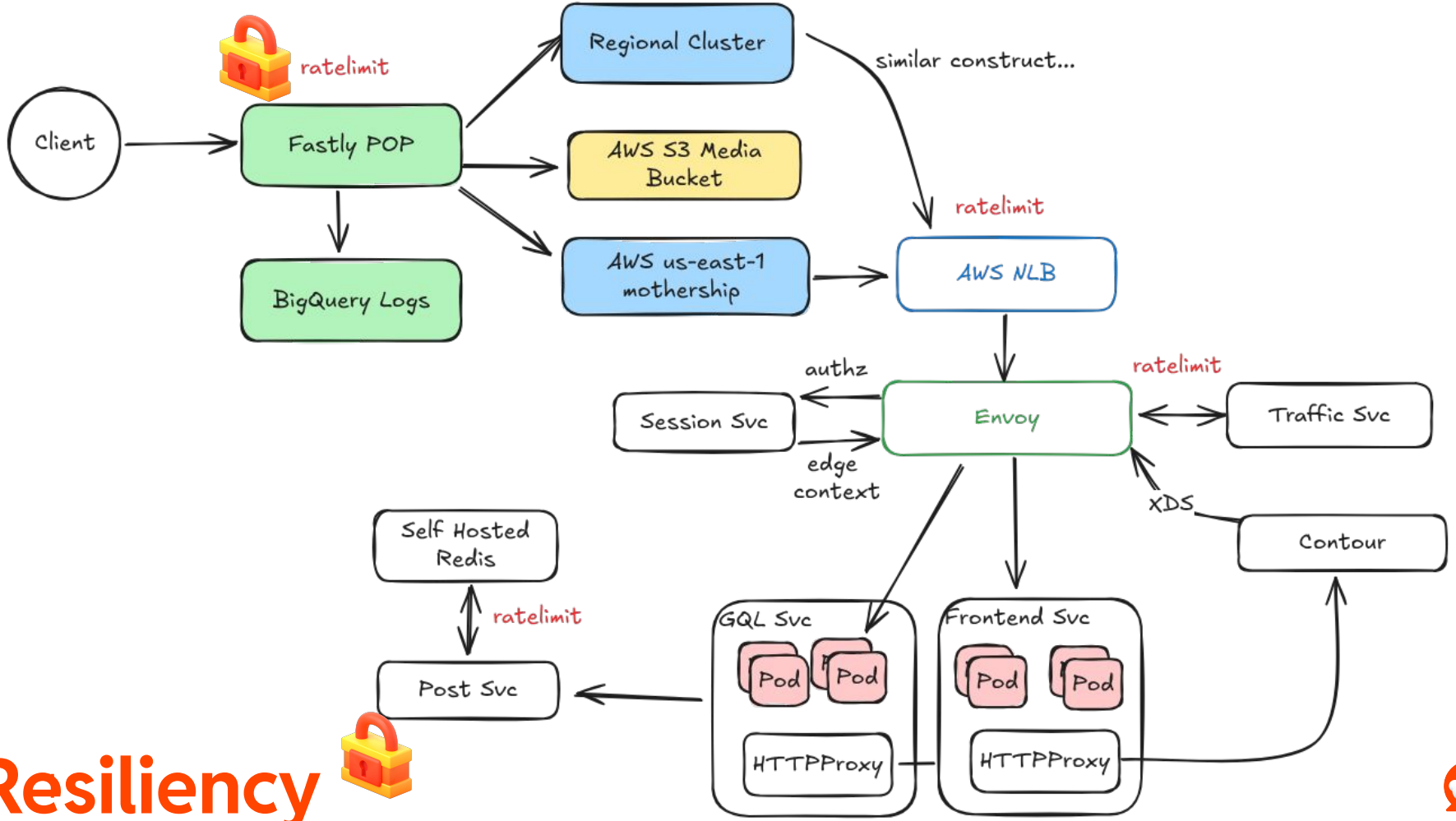
- Drop impossible requests (bad verbs, paths, synthetics to limit backend load)
- Session material validation at edge (JWT vs cookie)
- Resource pools
  - “Slowlane” and “woahlane” on trusted signal
  - Isolate degradation of service by request type / shape

Resiliency Techniques

- Make attacks more expensive for attackers:
  - **Tarpitting** - increase response time, requires attacker to have more concurrent port/request processing which might limit their size of attack
  - **Response Bloat** - best attacks in early Reddit were simple GET requests that cost nothing. Make them eat their network bandwidth at scale.

Resiliency Techniques

- **Cookie Trampolining**
  - For brand new connections, add a cookie to a redirect response and see if client follows it
  - Expected cookie not returned that should have had a request already?
  - Good for blocking dumb scrapers / clients


ratelimit
Client
Fastly POP
Regional Cluster
similar construct...
AWS S3 Media Bucket
AWS us-east-1 mothership
BigQuery Logs
ratelimit
AWS NLB
authz
Session Svc
edge context
Envoy
ratelimit
Traffic Svc
XDS
Contour
Self Hosted Redis
ratelimit
Post Svc
GQL Svc
Pod
Pod
HTTPProxy
Frontend Svc
Pod
Pod
HTTPProxy

Resiliency

Lessons Learned

- Want it done right, do/staff it yourself
- You don’t need a WAF
- Expect cat and mouse
- Start with largest needles
- Want more insight? Buy votes

reddit

WAFS?!

WhAt AbOuT WAFS?!

- **Not a great fit** for Reddit - too many false positives, too many edge cases
  - HTML or SWE topics would trigger
- **Impacts performance, latency, and cost at our RPS scale**
- Better do it ourselves since we know our traffic patterns
- 🌶️ - Trade WAF for better appsec hygiene

reddit

Thanks

Made Possible with Contributions By:

Shadi Altarsha

Justin Ely

Clayton Gonsalves

Jason Harvey

Brian Landers

Udgeetha Mallampalli

Sotiris Nanopoulos

Sean Rees

Chris Schomaker

Hiram Silvey