RateLimit Headers

Communicate service status

HTTPAPI-WG @ IETF-113

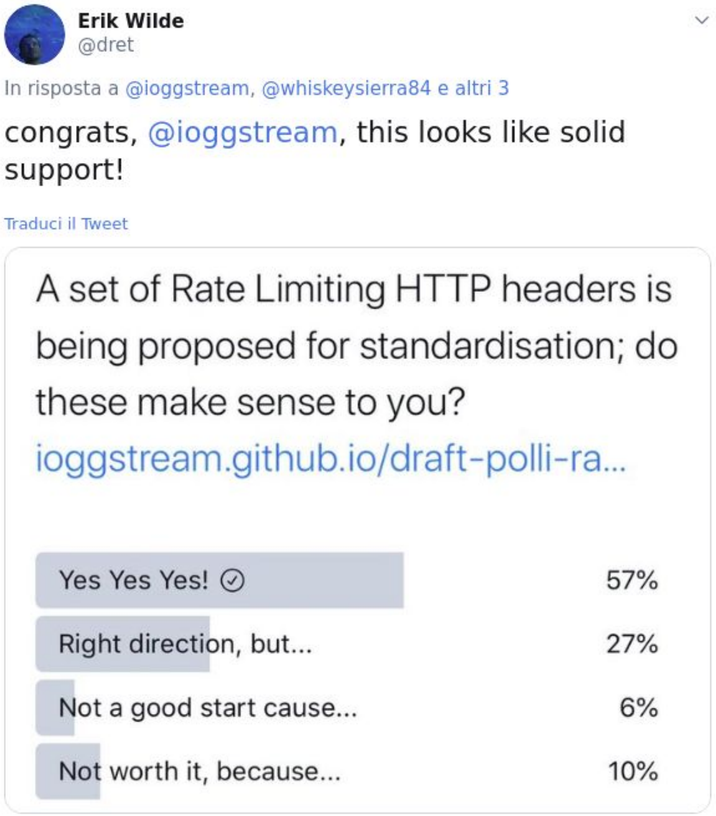
ietf-httpapi-ratelimit-headers

[see the specifications]

RateLimit Structured Fields - Goals

- communicate service limits, so clients can stop before being throttled out
- align all the *already existing* ratelimit headers and stop headers' proliferation
- express multiple RateLimit policies

Who wants it & Implementers



Erik Wilde
@dret

In risposta a @ioggstream, @whiskeysierra84 e altri 3

congrats, @ioggstream, this looks like solid support!

Traduci il Tweet

A set of Rate Limiting HTTP headers is being proposed for standardisation; do these make sense to you? ioggstream.github.io/draft-polli-ra...

Yes Yes Yes! ✓ 57%

Right direction, but... 27%

Not a good start cause... 6%

Not worth it, because... 10%

Configurable in:

- Red Hat 3scale
- Kong
- Envoy
- Azure API Gateway

Supported by:

- Italy
- The Netherlands

STOP headers proliferation

X-RateLimit-UserLimit: 1231513

X-RateLimit-UserRemaining

X-Rate-Limit-Limit: name=rate-limit-1,1000

x-custom-retry-after-ms

x-ratelimit-minute: 100

x-rate-limit-hour: 1000

X-RateLimit-Remaining-month

X-RateLimit-Retry-After: 11529485261

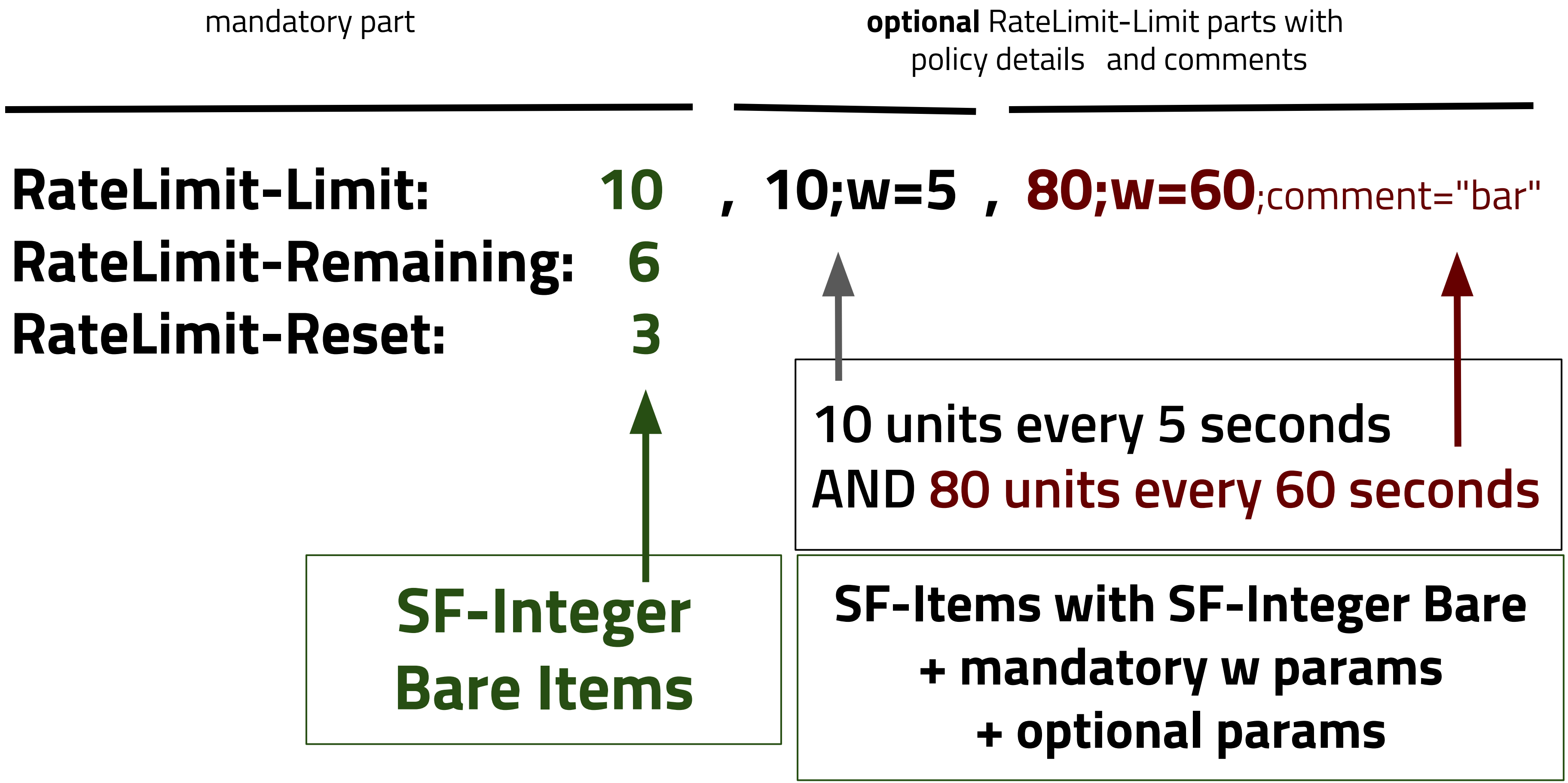
X-Rate-Limit-Reset: Wed, 21 Oct 2015 07:28:00 GMT

BANNED

RateLimit-Limit:	**SF-List**	**#quota-units**
RateLimit-Remaining:	**SF-Integer**	**#quota-units**
RateLimit-Reset:	**SF-Integer**	**#delta-seconds**

... and many more!

Example with multiple quotas



mandatory part

optional RateLimit-Limit parts with policy details and comments

```
RateLimit-Limit:        10 , 10;w=5 , 80;w=60;comment="bar"
RateLimit-Remaining:     6
RateLimit-Reset:         3
```

10 units every 5 seconds
AND 80 units every 60 seconds

SF-Integer Bare Items

SF-Items with SF-Integer Bare + mandatory w params + optional params

Technical choices

- #60 support **only delta-seconds** (no ntp skew & adjustment issues) like Retry-After
- #35 Use Structured-Headers

- flexible semantics to express dynamic policies, sliding windows and concurrency limits
- don't mention infrastructural concepts like connections

https://github.com/ietf-wg-httpapi/ratelimit-headers/pulls?q=is%3Apr+is%3Aclosed

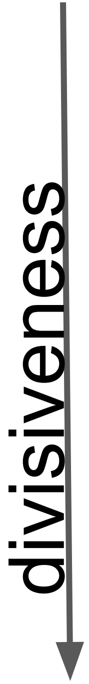
Changes from -02

- #35 Use Structured-Headers (may need editorial rework)
- #80 Field dependencies
    - RateLimit-Limit, Ratelimit-Reset: REQUIRED
    - RateLimit-Remaining: RECOMMENDED
- #83 Throttling scope is delegated to parameters, that can be further registered in a IANA table

https://github.com/ietf-wg-httpapi/ratelimit-headers/pulls?q=is%3Apr+is%3Aclosed
https://www.ietf.org/rfcdiff?url2=draft-ietf-httpapi-ratelimit-headers-03

Open Issues Needing Input before WGLC

- #79 separate quota policies from expiring limit (editors are supportive)
- #41 Upper bound for RateLimit-Reset? (feedback welcome)
- #65 Field names (editors do not support changing field names due to adoption concerns)



divisiveness

https://github.com/ietf-wg-httpapi/ratelimit-headers/issues

#79 separate quota policies

Now

RateLimit-Limit: **SF-List**

List[0]: **SF-Integer** #expiring limit

List[1:]: **SF-Item** #quota-policy

TO BE

RateLimit-Limit: **SF-Integer** expiring-limit

RateLimit-Policy: **SF-List**

List[]: SF-Item #quota-policy

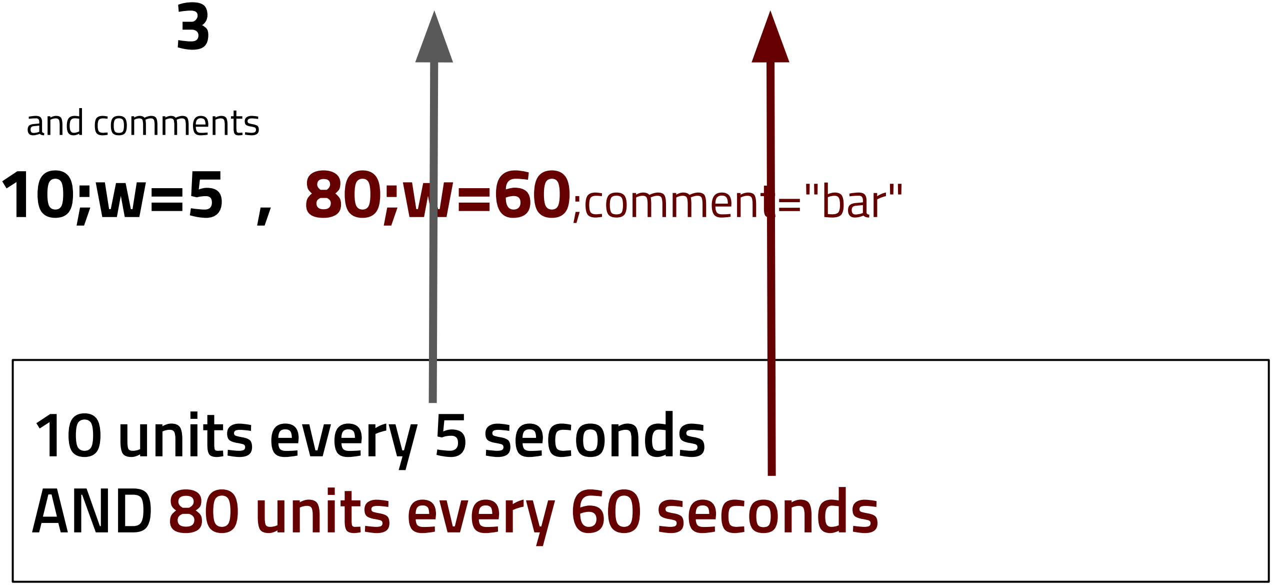
- easier to parse
- avoid confusion between the Expiring Limit and Quota Policy
- all list items have the same structure

#79 Separate Policy Field

RateLimit-Limit: 10
RateLimit-Remaining: 6
RateLimit-Reset: 3

optional parts with policy details and comments

RateLimit-Policy 10;w=5 , 80;w=60;comment="bar"



10 units every 5 seconds
AND 80 units every 60 seconds

FAQ

Q: Are we inventing a new service management model?

A: No. We just standardize headers semantic for the many who *already* use this pattern.

Q: Why don't use timestamps for RateLimit-Reset?

A: Timestamps *require* NTP on both sides. NTP in the real world is hard (skew, adjust, IoT, ...). We like Retry-After too ;)

Thanks!

Roberto Polli - robipolli@gmail.com

Alex Martinez - amr@redhat.com

Backup slides