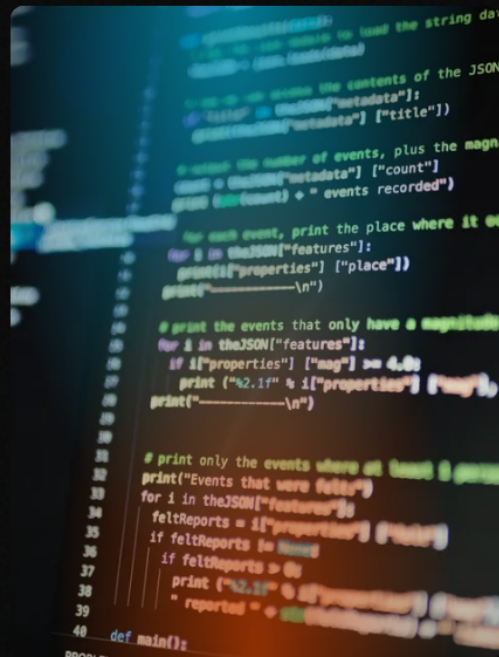


Parcelpath API

DESIGN REVIEW · 04



Rate limiting the public API

PLATFORM TEAM · PARCELPATH.EXAMPLE
(FICTIONAL)



Made with MagicSlides



THE PROBLEM

Three customers sent 61% of all API traffic last month. When one of them ran a backfill on 9 September, p99 latency for everyone else went from 180 ms to 2.4 seconds.

We have no per-key limits today. The only protection is the load balancer, which drops requests at random once it is full.

61%

TRAFFIC FROM 3 KEYS

2.4 s

P99 DURING BACKFILL

0

PER-KEY LIMITS TODAY



Made with MagicSlides

THE PROPOSAL

A token bucket per API key, stored in Redis, checked at the gateway before any request reaches a service. Every response says how much budget is left.



One counter per key, refilled every second.



Made with MagicSlides

Fair by default, **generous on purpose.**

Bucket size

600 requests per key, so normal bursts never hit the limit.

Refill rate

20 requests per second on the standard plan, 100 on enterprise.

429 response

Includes a Retry-After header and the RateLimit-Remaining count.



Made with MagicSlides



Rollout plan

01

Week 1: shadow mode

Count and log would-be 429s without blocking anything.

02

Week 2: talk to heavy users

Email the keys that would have been limited, with their own numbers.

03

Week 3: enforce

Turn on for standard keys first, enterprise a week later.

<1%

KEYS EXPECTED TO HIT LIMIT

0.4 ms

ADDED GATEWAY LATENCY

200 ms

Made with MagicSlides

Approve shadow mode **this Monday**.

Risk

Shadow mode blocks nothing, so the downside is one Redis call per request.

Owner

Platform team, with support briefed before week 2 emails go out.

Rollback

A single feature flag turns enrollment off in under a minute.

