Abhi tak hamne 'Global Rate Limiting' dekh rahe the matlab all over globe kahi se jisne req di, uspe rate limiting kar rahe the

↙

lekin humlog Rate limiting lagate h user level pe ya fir, IP level pe.

↙

User ya IP level banane ke liye dikkat ye h ki, mujhe har ek userid ya IP Address ke corresponding mujhe bucket banana padega.

Note:- Hum Rate limiter API based bhi bana sakte h, matlab ek application mai let say 100 API, to mai chahoe to haar ek ke liye alag Rate limiter laga sakta hu

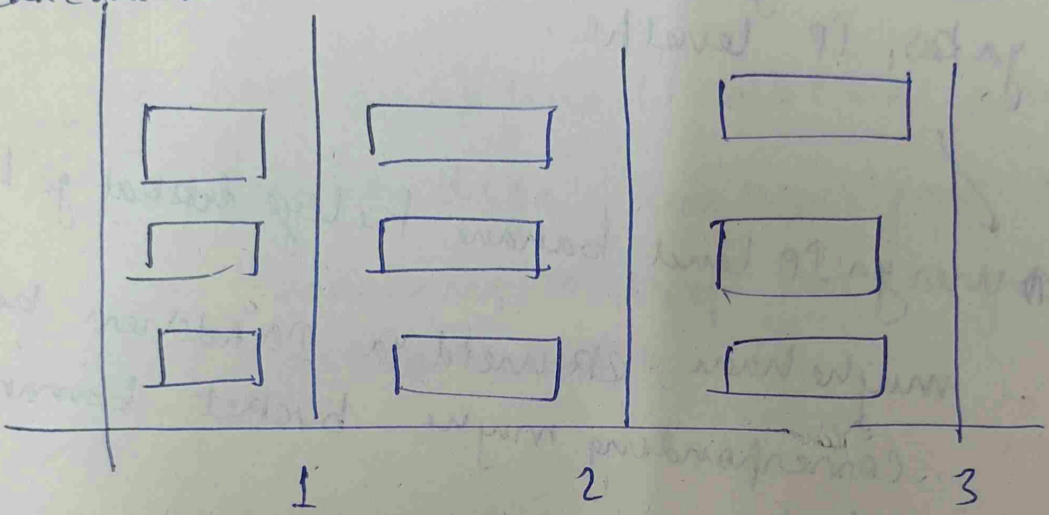
→Reason:- ~~saare API~~ kuch API ka task small ho sakta, jaldi process karle aur mai uska rate big rakhna type.

Fixed Window Counter

① In a fixed interval (say a sec or a minute) only a specific no. of req can come.

→ Isse bhi aap global ya user ya IP level pe implement kar sakte ho

Ideal case me



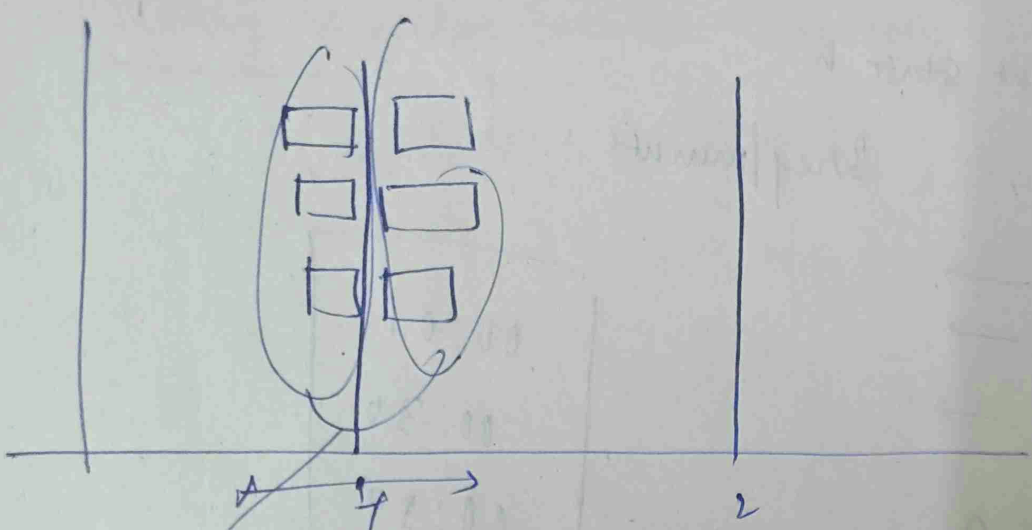
Fixed Counter = ③ → ek particular fixed time interval mai 3 se jayda reqni aa sakte.

~~Cons~~:- ~~Pros: Server pe load n~~

Cons:-

If burst traffic comes at edges of window it may lead to server crash, high latency.

(matlab PTO



Ye teeno alag alag interval mai h

lekin in short interval mai kaafi saare req aagaye lekin fir bhi accept karna padega.

Sliding window log Algo

- → Best Algorithm for Rate limiting
- — Strict
- → Slow & memory consuming

Algorithm:

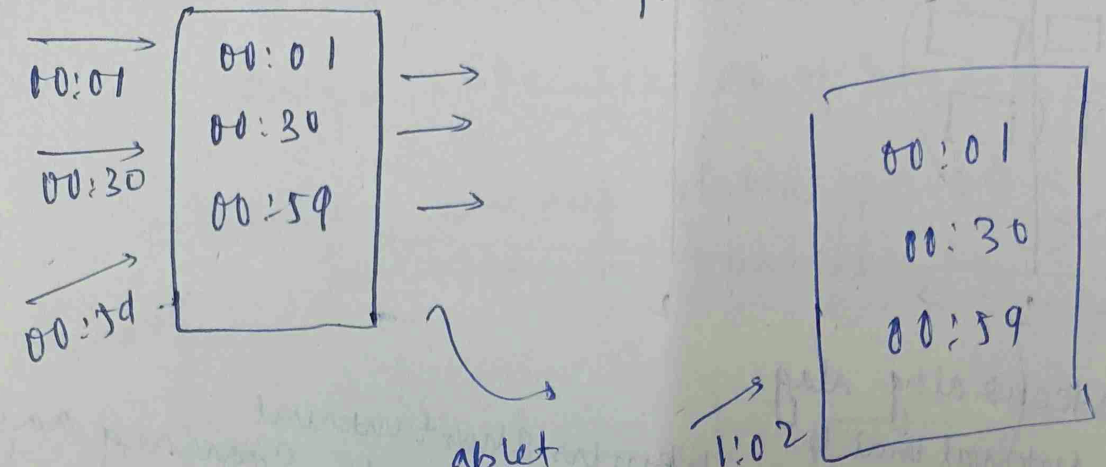
(i) It will store each req in a log file.

(ii) Whenever a req comes, it will first remove all the outdated req from the file.

(iii) Then it will log the new req and check, if the counter limit reached, drop the req else send it to server.

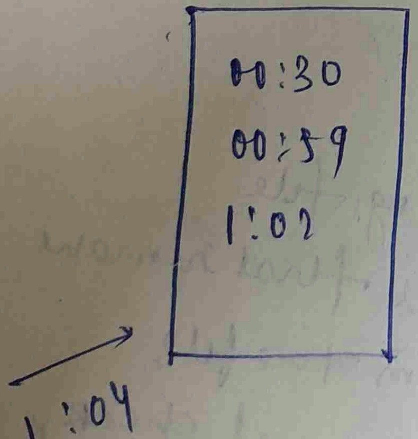
lets say ye current state h.

3req/minute.



ablet Say ek req aayi at 1:02

ye dekhega ki
1:02 - 00:01 > 1m.
to 00:01 hatega.
aur 1:02 aaega.



ab 1:04 - 0:30 < 1 min to
ye req discard ho jayega.
ya memory queue mai
daal dega.

Is process mai memory jayda lagta h
as saari req ka log store karke
rakhna parta h.

Sliding Window Counter Algorithm

1:40:00

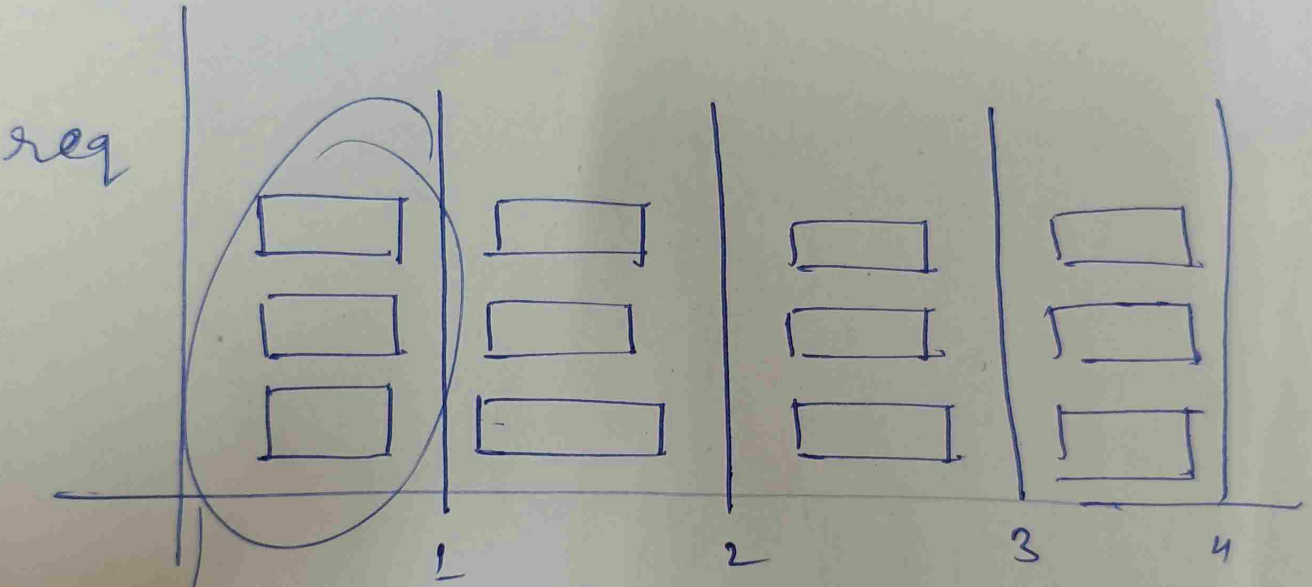
(will do at end)

Create our own Rate limiter

→ Aapko khud ka Rate limiter banane ke liye kin cheezo ke jarurat padne wale h?

→ Which algo to implement?

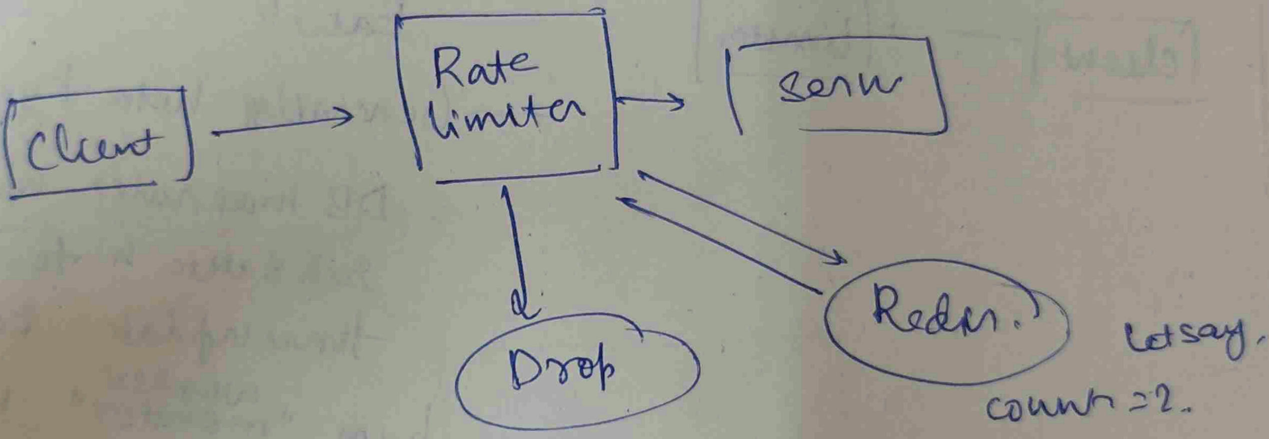
→ let's say for simplicity → Fixed window counter.



Assume kar rahe initially ki uniformly aa raha req aur server pe load ni pad raha.

Expected Counter = 3
Current Count = [0 to 3] } to in values ko kaha store karenge
→ Cache or Redis kyuki fast chahiye

Rate limiting ko kaha implement karo?
→ Mai kar raha Middle war.

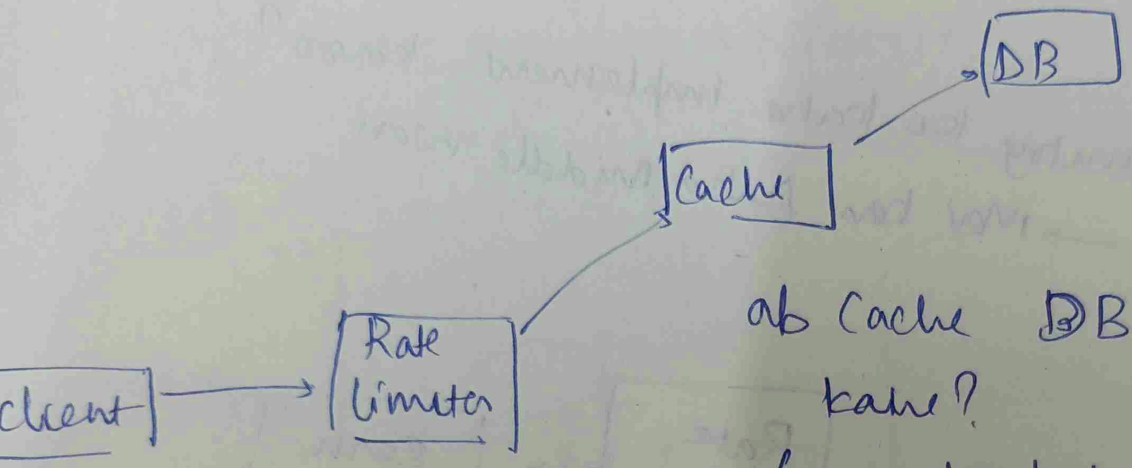


Jabhi req aayega Ratelimiter to
wo Redis se puchega ki me allow
karu ya ni,

aur kuch time baad sab reset

Is Rate limiter ke kuch rules honge → Jo mai DB mai Store karunga

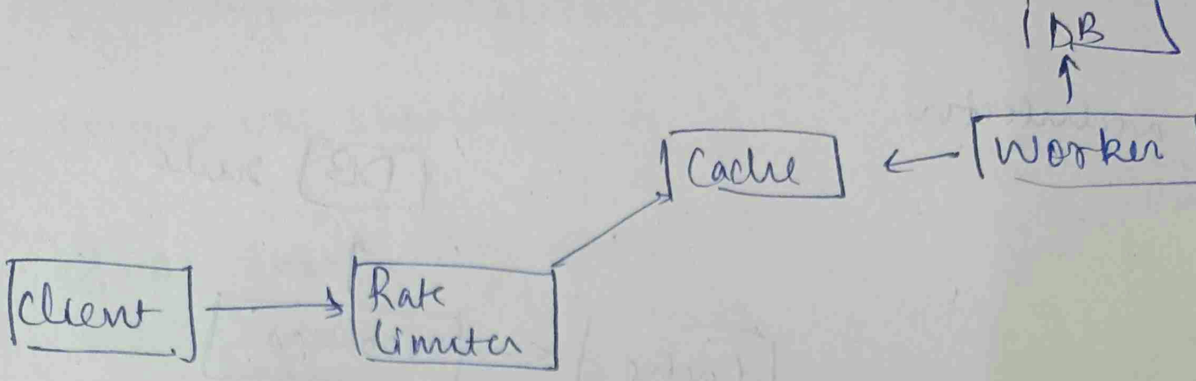
lekin DB se access kaafi slow ho jayega, isliye Cache bhi use karenge.



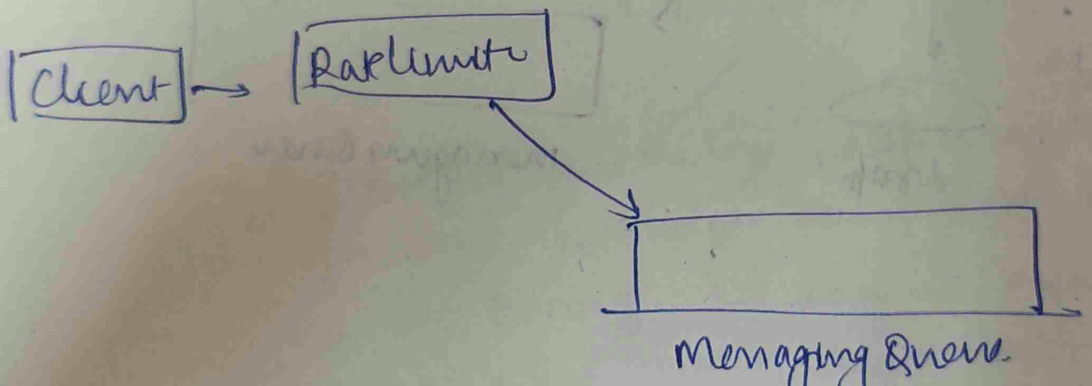
ab Cache DB se data liga kahe?

→ Generally hota kya h ki DB mai rules update hote reh sakte h' to usse ~~time~~ update karne ke liye hum "~~monitor~~ worker" install karenge

→ ye bas ek code type h jo haar ek particular interval pe DB se latest data fetch karke Cache mai daal dega



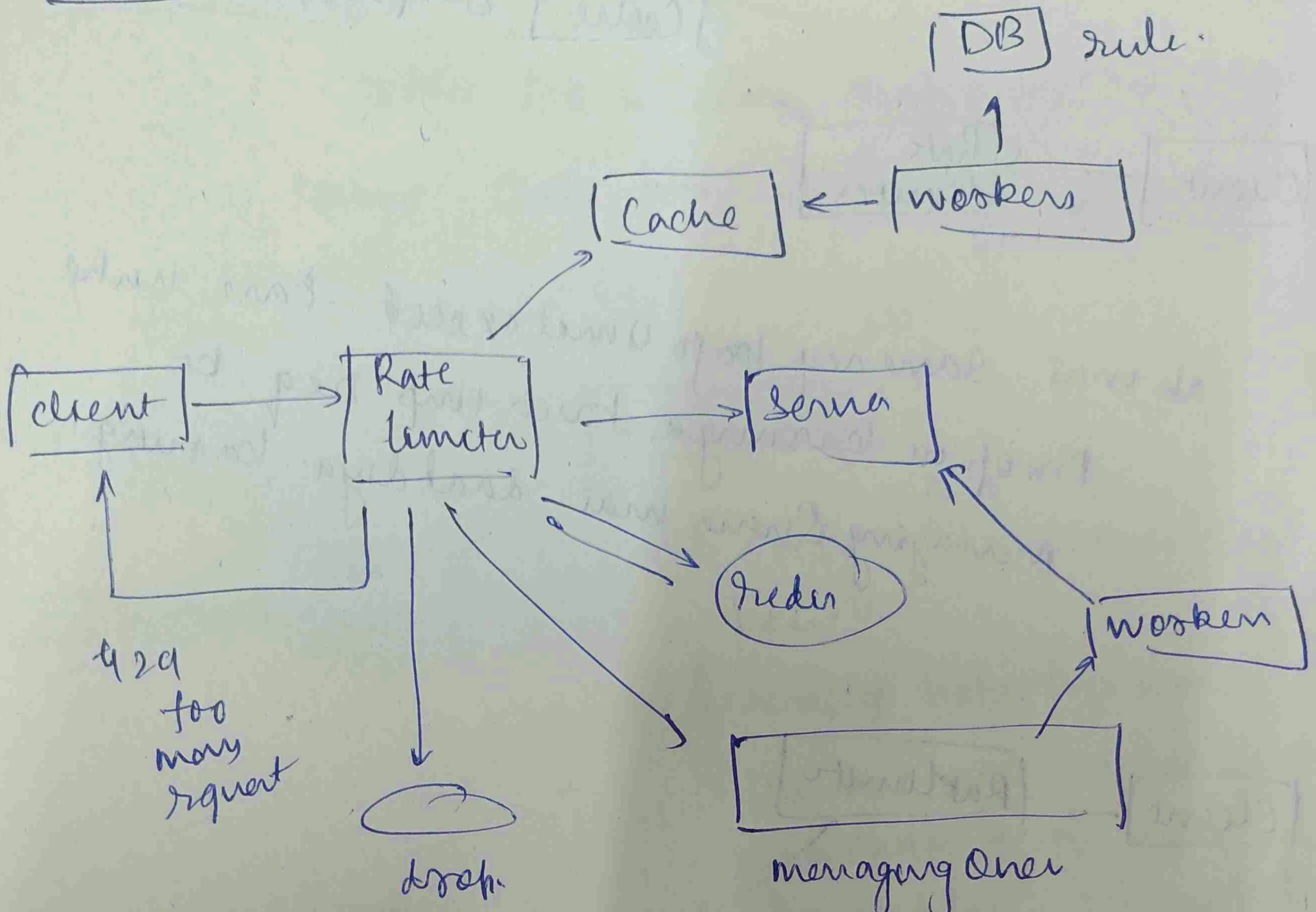
ab mai saare req 100 jo limit exceed kare unhe. Drop ni karunga, kuch imp req ko. Managing Queue mai daal diya karung



Ab messaging Queue se req server pe chale jayeg by "workers"

"workers" dekhenge ki server free h. aur messaging Queue mai req pending to wo usse utha ke server ko de dega.

whole architectur


DB
rule.
Cache
workers
client
Rate limiter
Server
redis
worker
429 too many request
drop.
messaging Queue