Hello!

I am Riteek Srivastav

I work as lead software engineer at Gojek, Data Engineering team.

A/B Testing platform at scale

Agenda

- What is A/B testing?
- Objective
- Litmus
- Old Architecture
- Current scale
- New Architecture

What is A/B Testing?

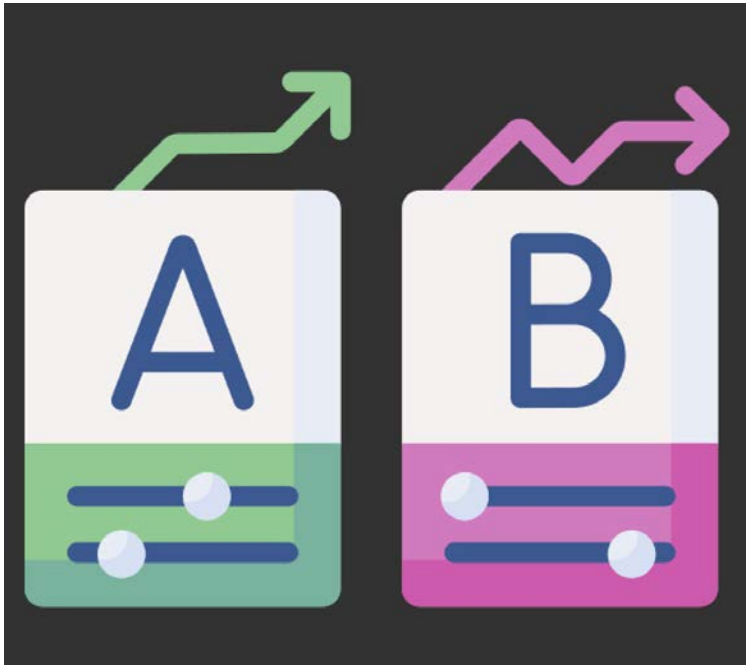


Image credit: flaticon.com

If you double the number of experiments per year, you are going to double your INVENTIVENESS

– *Jeff Bezos, Founder of Amazon*

Litmus

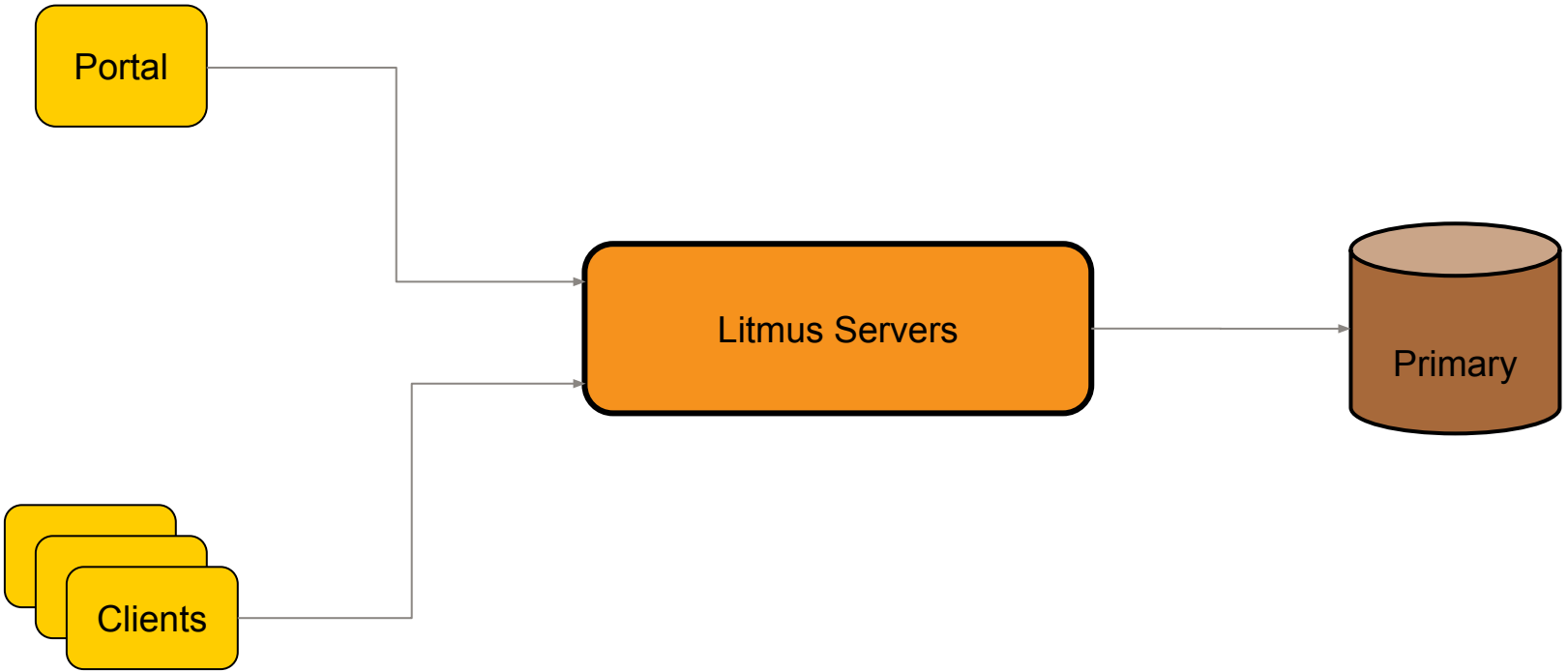
- Experimentation platform of Gojek
- It Supports
  - A/B testing [Experiment]
  - Staggered rollout [Release]
- Supports targeting rules, e.g.
  - `(app-version newer-than "3.12")`
  - `(platform in ["Android","iOS"])`
  - `(os-version not-in ["Android,4.4.2","iOS,10.3.3"])`
  - `(lat-long within-any-of [[-6.176189, 106.827056, 750], [-6.265930,106.783779,288]])`

Objective

Build and improve the system with the below non-functional requirement

- Low latency
- Highly available
- High throughput
- Weak Consistency

Initial Architecture



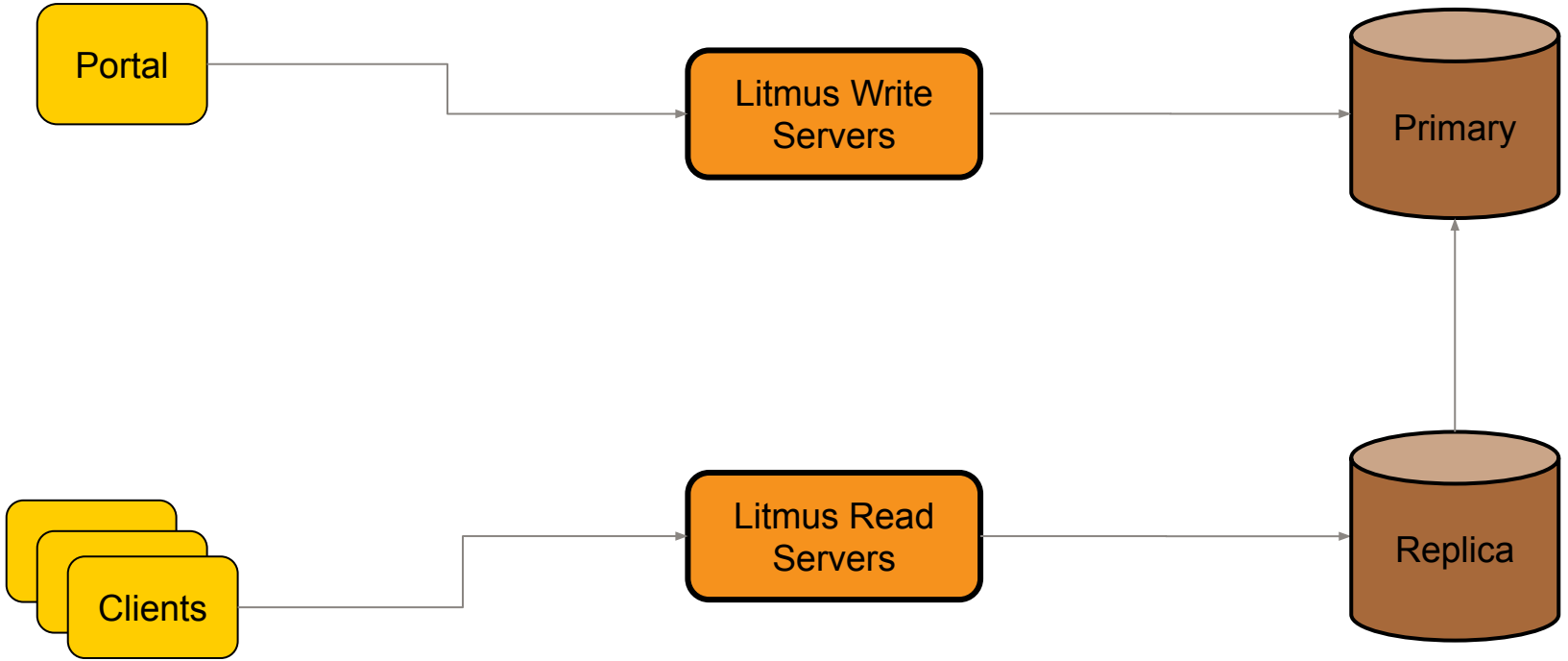
Limitations

👉 Single cluster handling the request of read (portal) and write (clients)

👉 Primary Database being used for read and write

👉 Single deployment

Separate read write cluster



Limitations

Pros

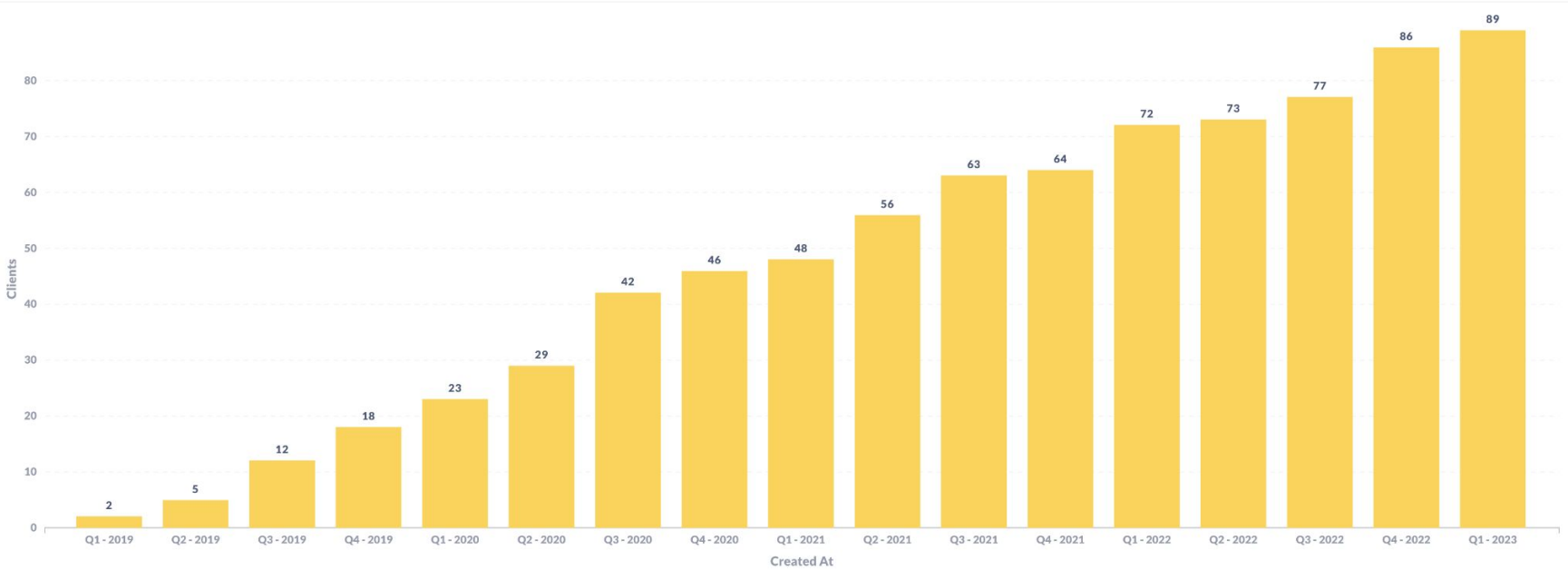
- Read and write cluster can scale independently
- Decoupled deployment
- Master/Slave setup being used

Cons

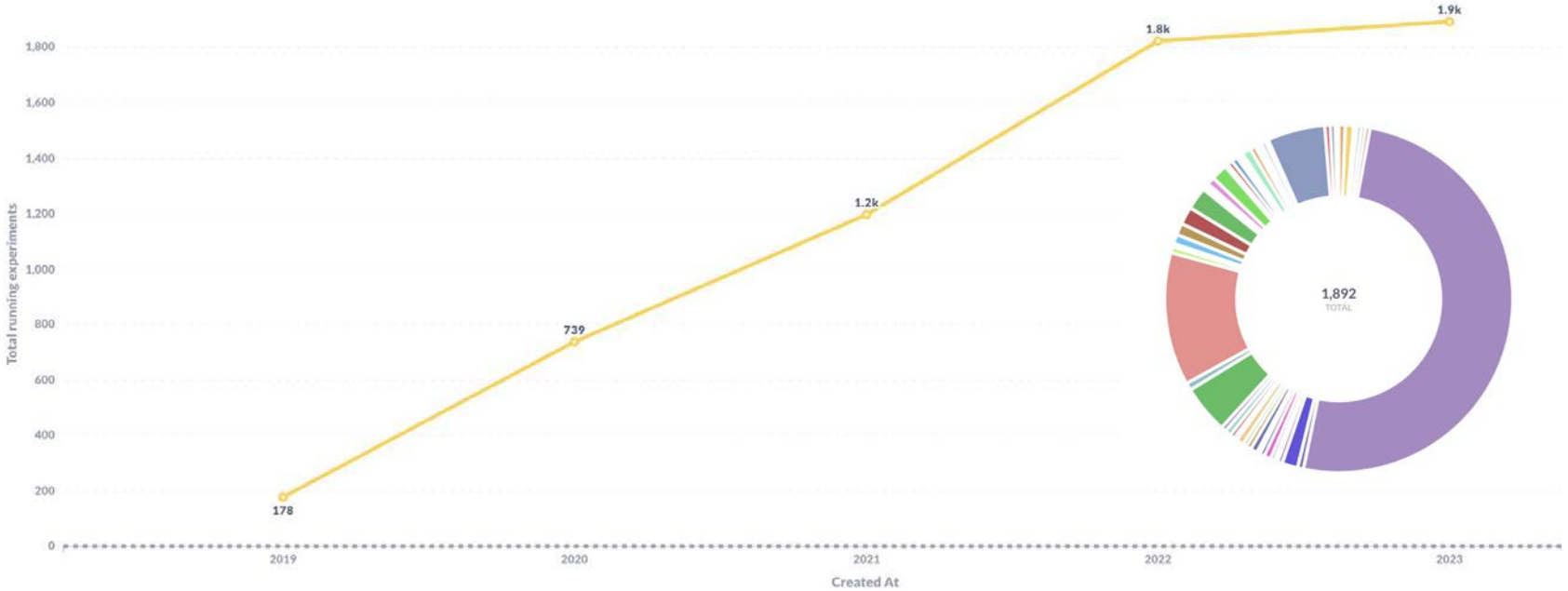
- All the clients are sharing the resources
- Not horizontally scalable

Scale over time

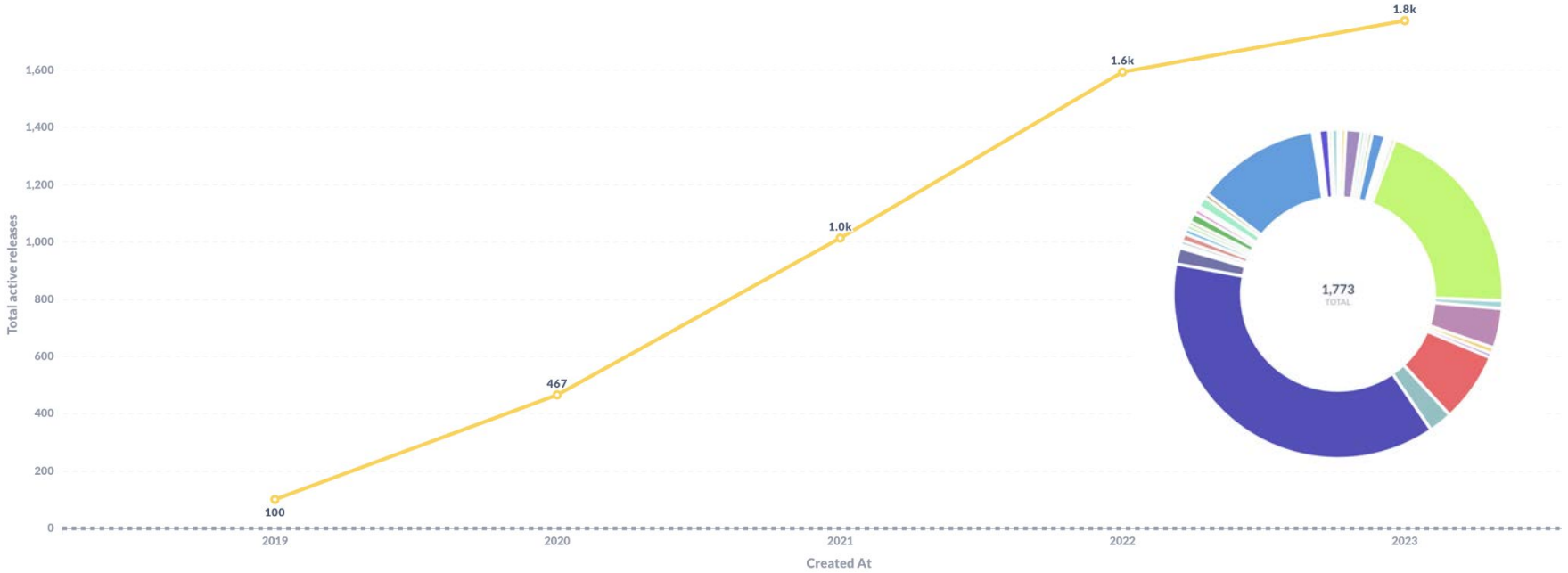
Total number clients over time



Total number of active experiments at a time



Total number of active releases at a time



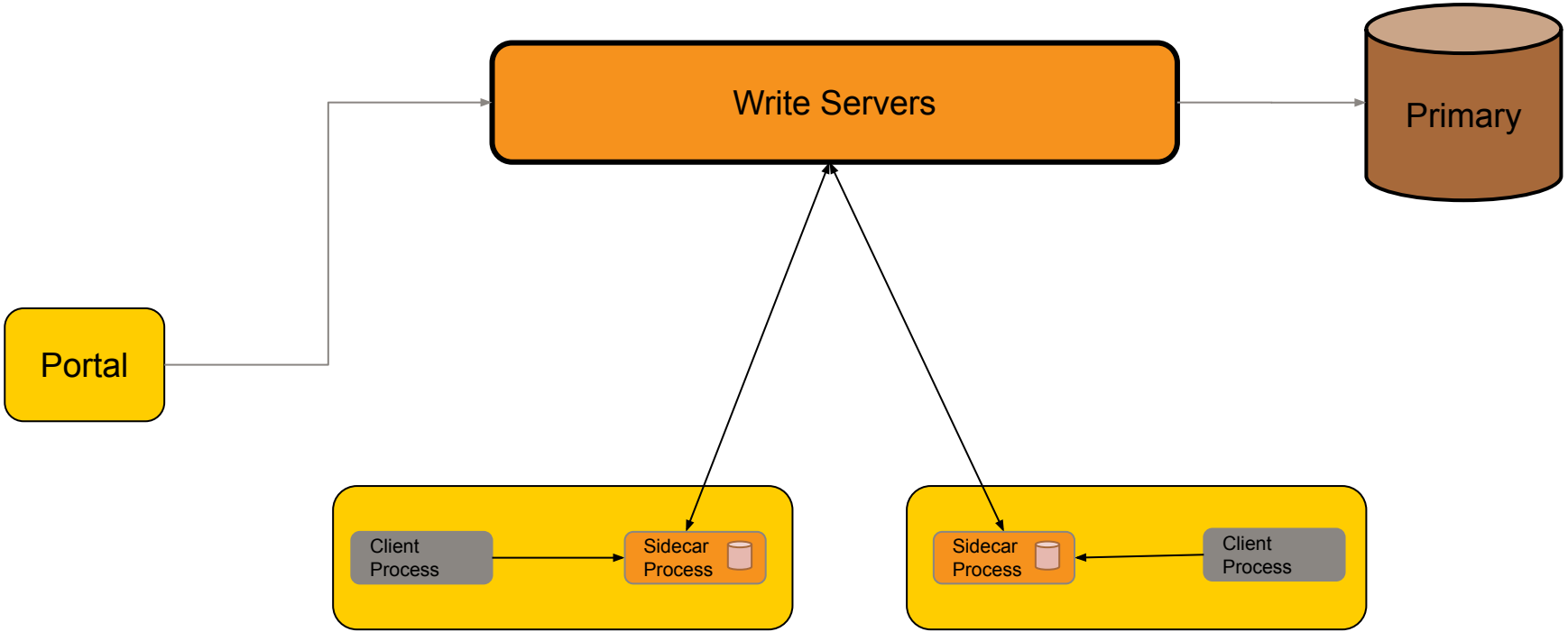
- Every year Litmus is receiving a new scale.

Year End	Throughput
2019	~100K
2020	~500K
2021	~1.1 M
2022	~2M

- Last two architectures were able to handle ~500K throughput within SLA.

- Looking at the adoption and growth of experimentation per year, we adopted sidecar pattern to optimize the latency

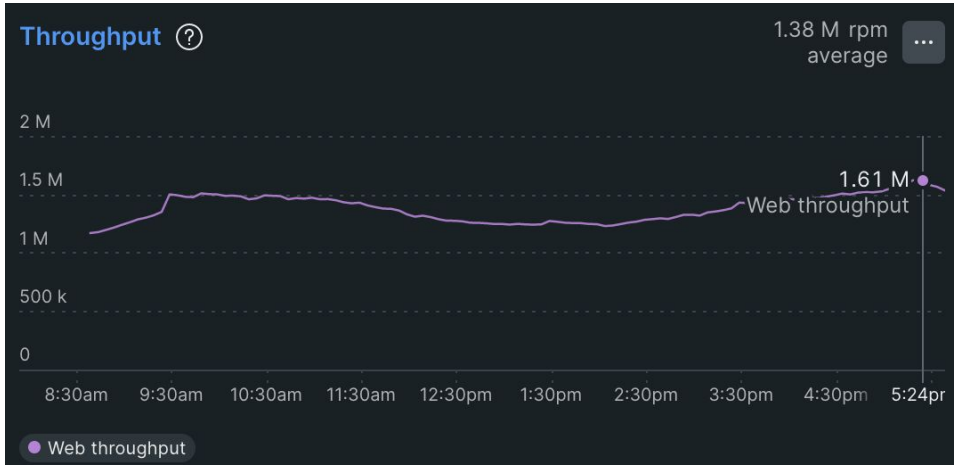
Sidecar Pattern



Key points for sidecar pattern

- Given sidecar is distributed, it is designed with Availability and Consistency (PACELC).
- Horizontally scalable.
- Resources are distributed, response time of a client will not be affected by experiments of other clients.
- We have central monitoring of all the sidecars on Newrelic and grafana.

Throughput/Response time



Thanks!

Any questions ?

You can find me at

- https://twitter.com/riteeksrivastav
- https://www.linkedin.com/in/riteek-srivastav/
- https://medium.com/@riteeksrivastava