



Building Koi Pond:

Simulating Millions of Slack Clients

Maude Lemaire (@qcmaude)



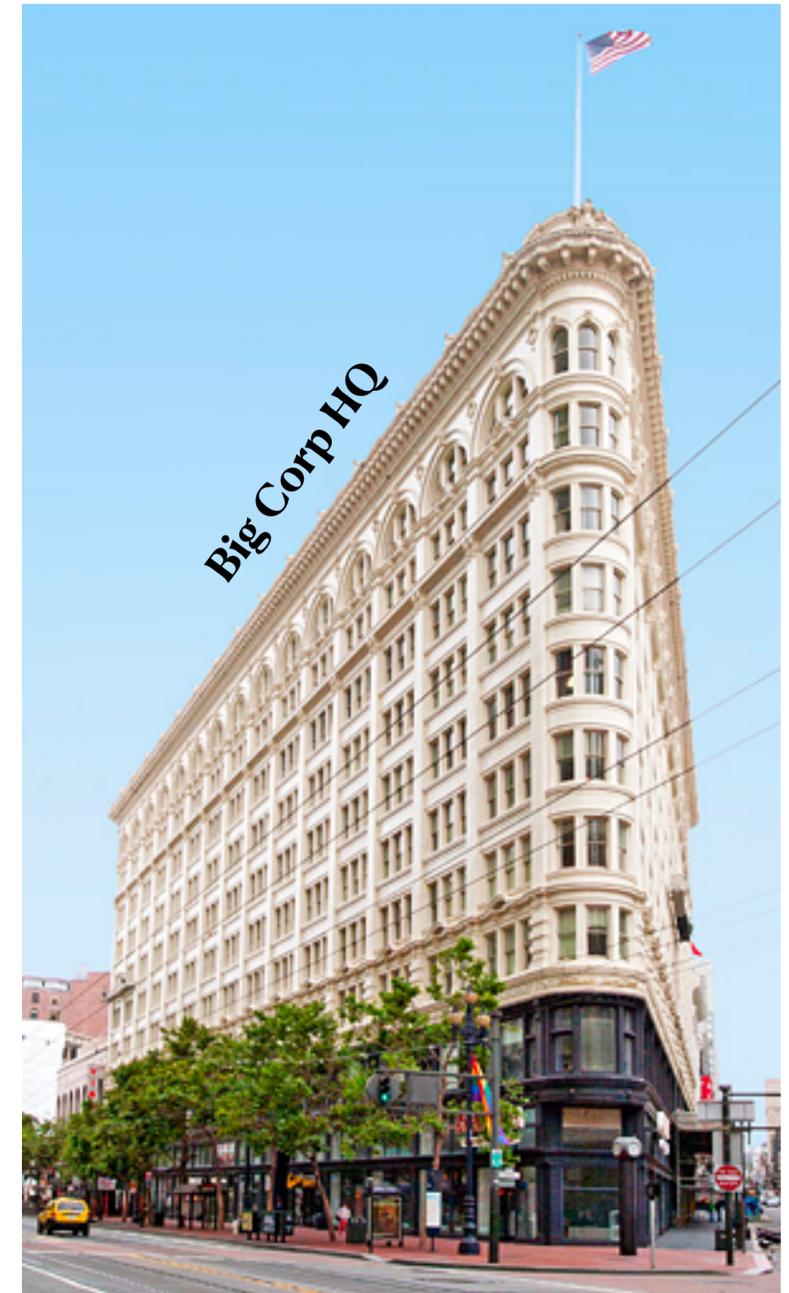
Building Koi Pond:

Simulating Millions of Slack Clients

Maude Lemaire (@qcmaude)

**Once upon a time ...
we had a *big* customer.**

We'll call them Big Corp.



Big Corp was already pushing the boundaries of our system.

Both of us were nervous to opt them into more features.



Servers fulfilling Big Corp requests

**But Big Corp really liked Slack,
and they wanted to use it *more*.**

So we needed to load test*.

***unfortunately, we didn't have any
tooling to do that.**

<https://slack.com/api/conversations.history?channel=C5JN6JH7V>
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>

Our first tool was simple. It blasted API requests to our servers.

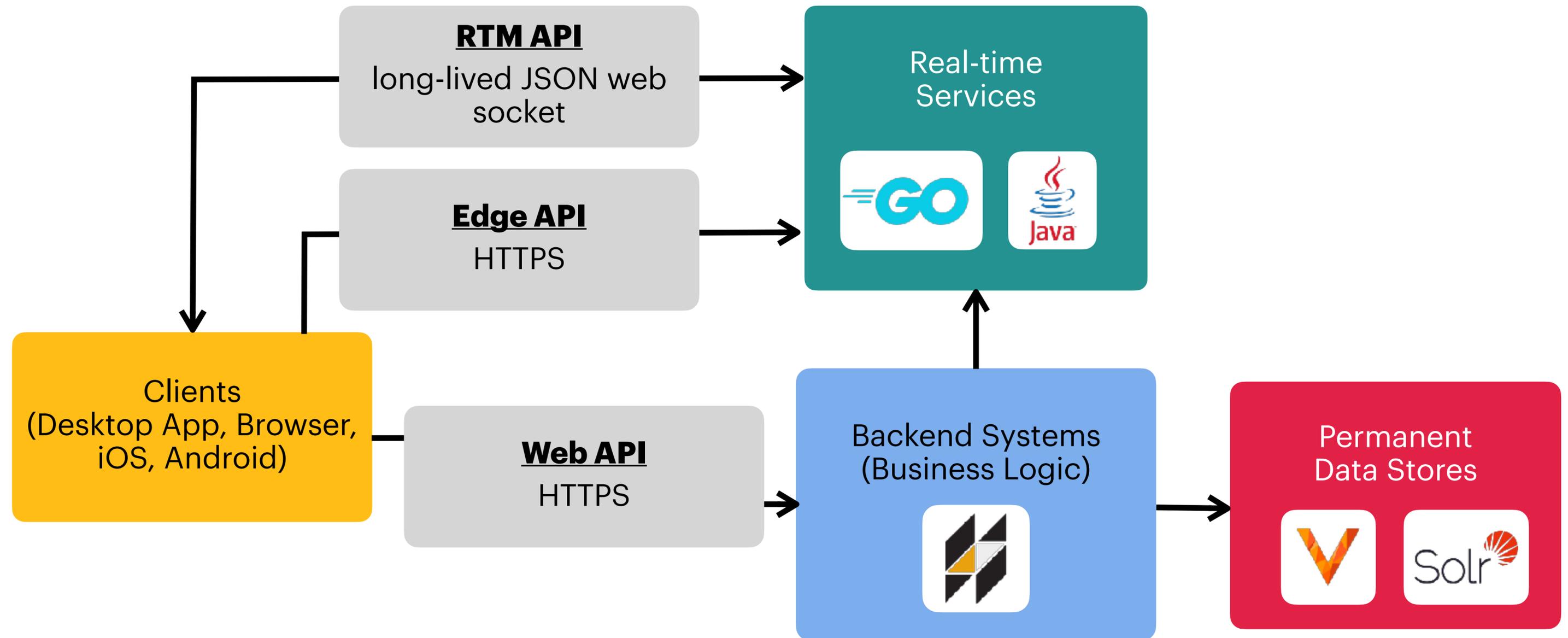
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>

Read more about API Blaster on our blog [here](#).

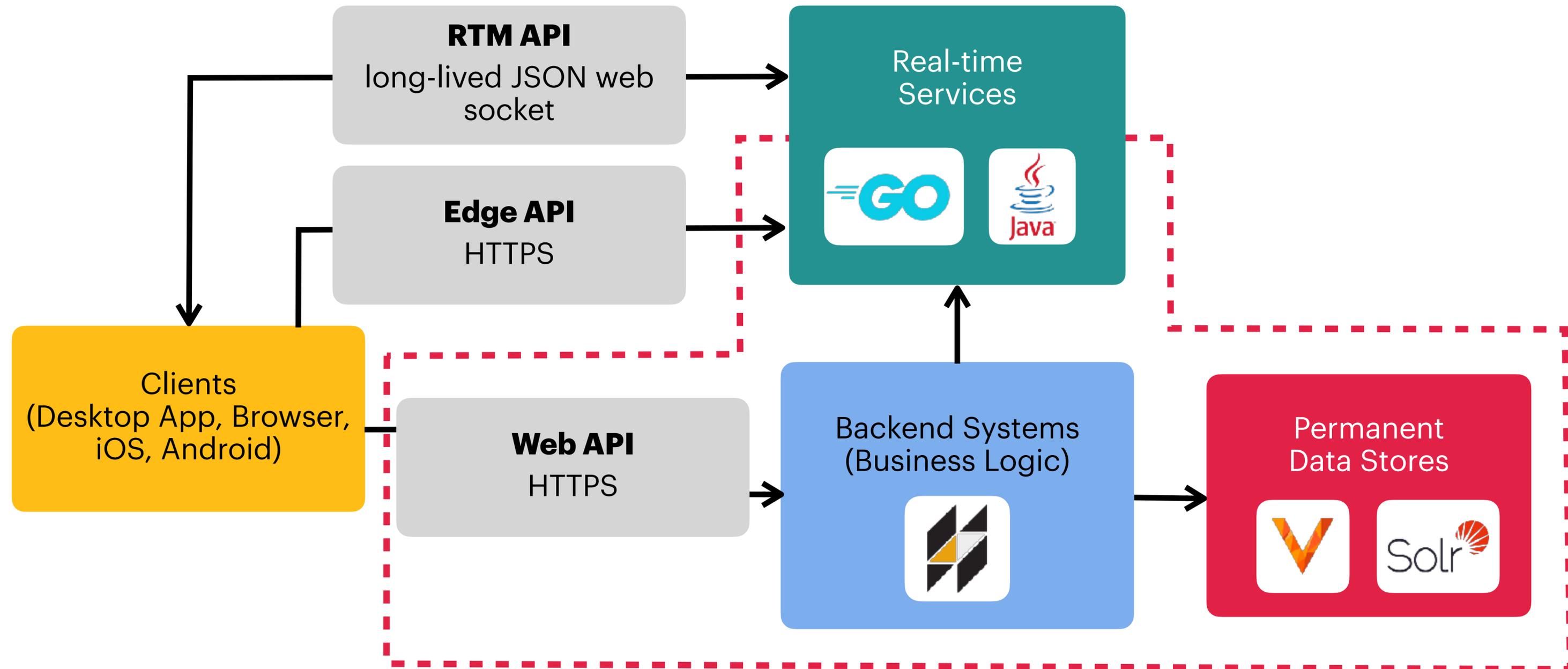
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>
<https://slack.com/api/conversations.history?channel=C5JN6JH7V>

Unfortunately, it didn't test a critical component of our infrastructure: the *websocket* stack.

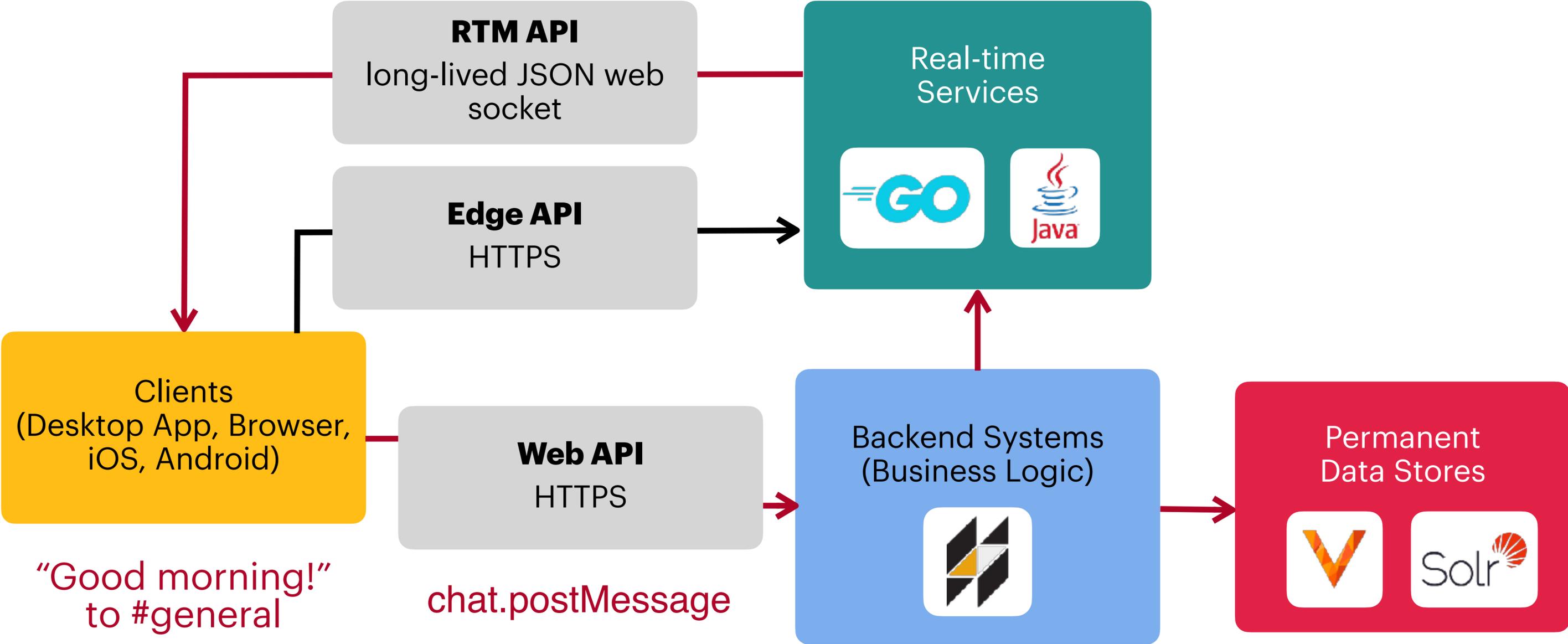
High-level overview of Slack's architecture



Surface area tested via API Blaster



Posting a message to Slack



**Even Bigger Corp.
came along.**



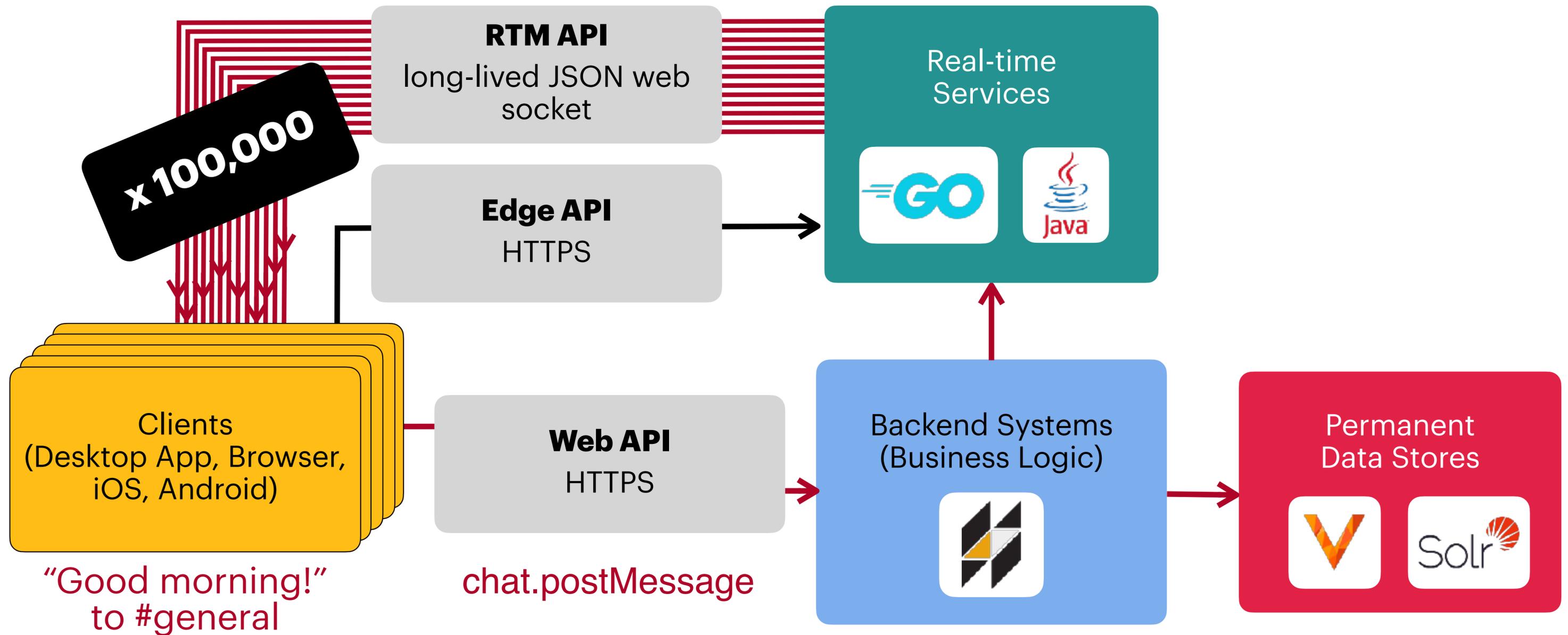
Like with Big Corp., Even Bigger Corp. also really liked Slack.

They wanted a channel to house all 300,000 of their users.

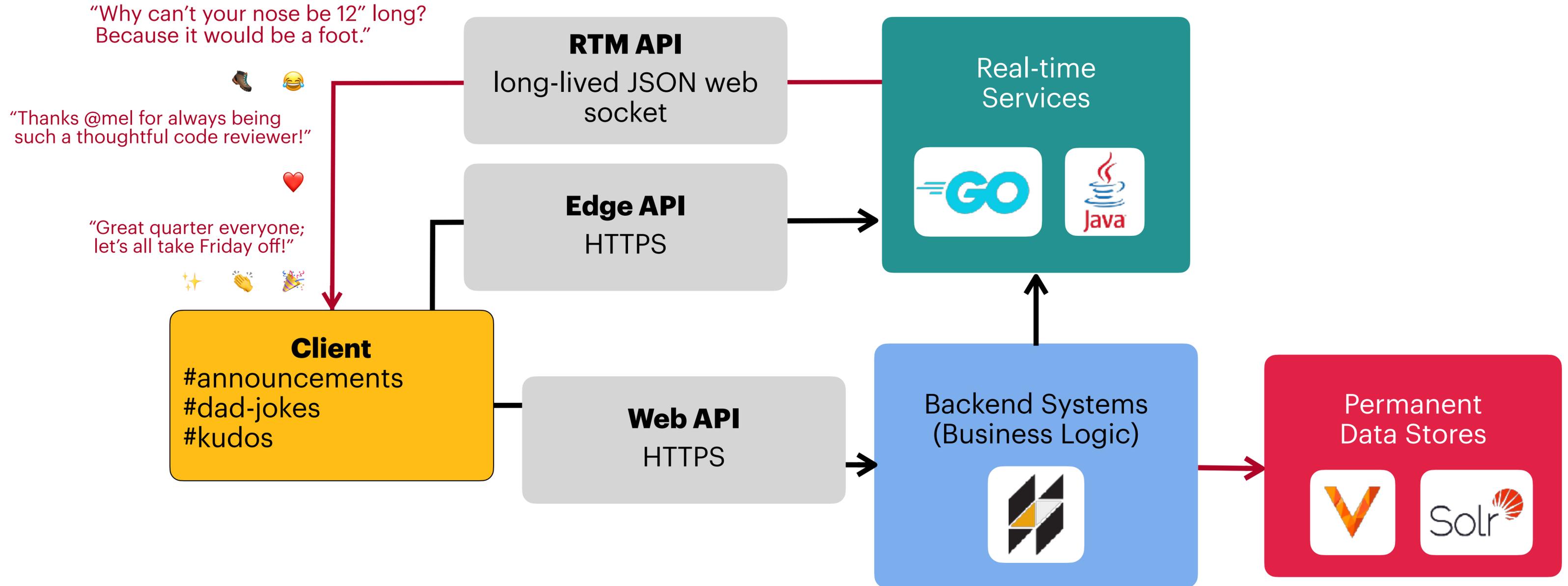
About 100,000 would likely be active at the same time.

Slack is susceptible to load-related performance problems in 3 key ways

1. Massive fan-out



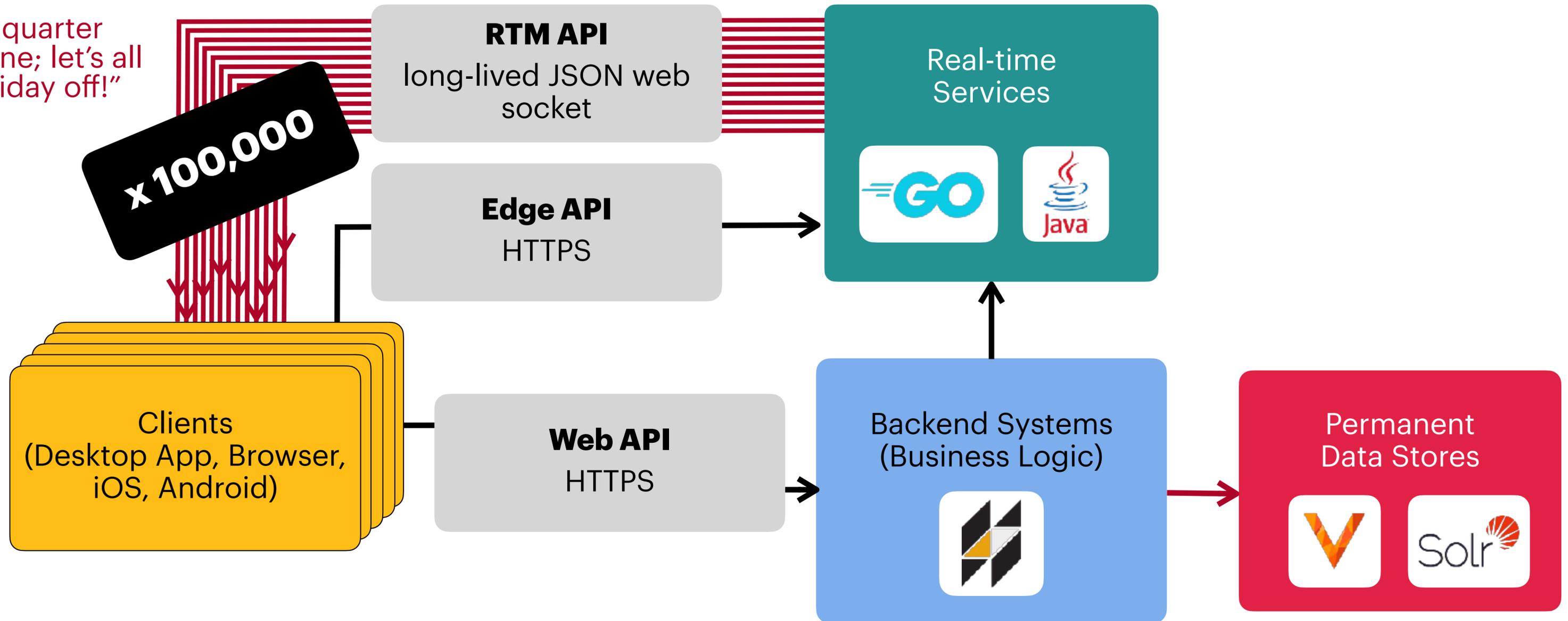
2. Event floods



3. Thundering herds

1

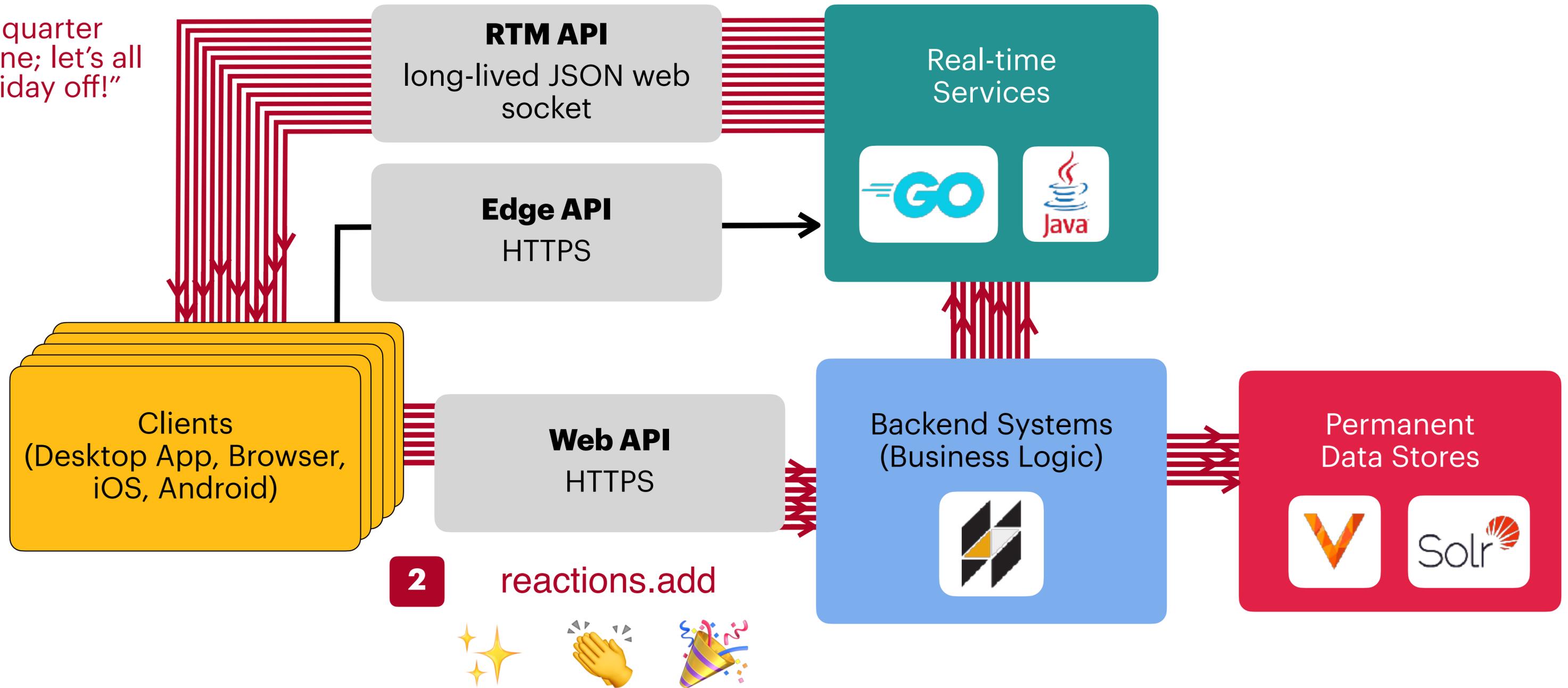
“Great quarter everyone; let’s all take Friday off!”



3. Thundering herds

1

“Great quarter everyone; let’s all take Friday off!”

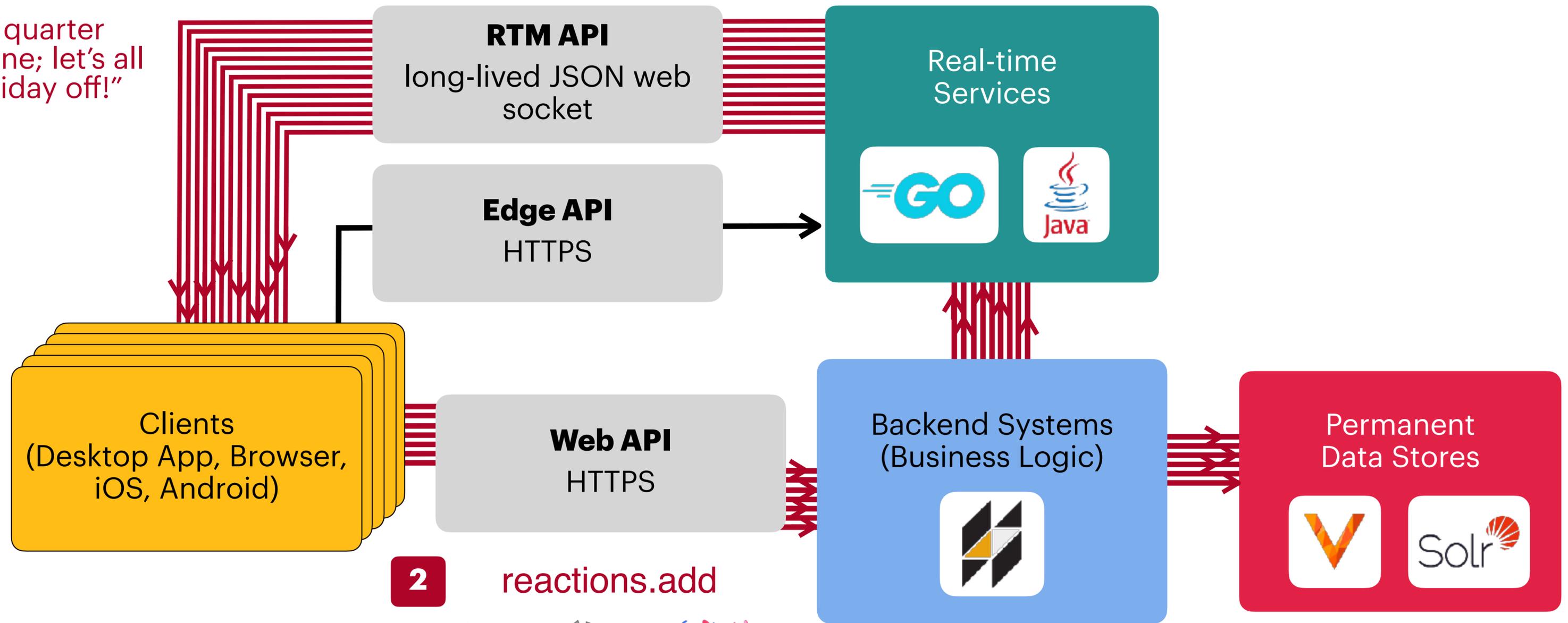


3. Thundering herds

1

“Great quarter everyone; let’s all take Friday off!”

3



2

reactions.add



3. Thundering herds

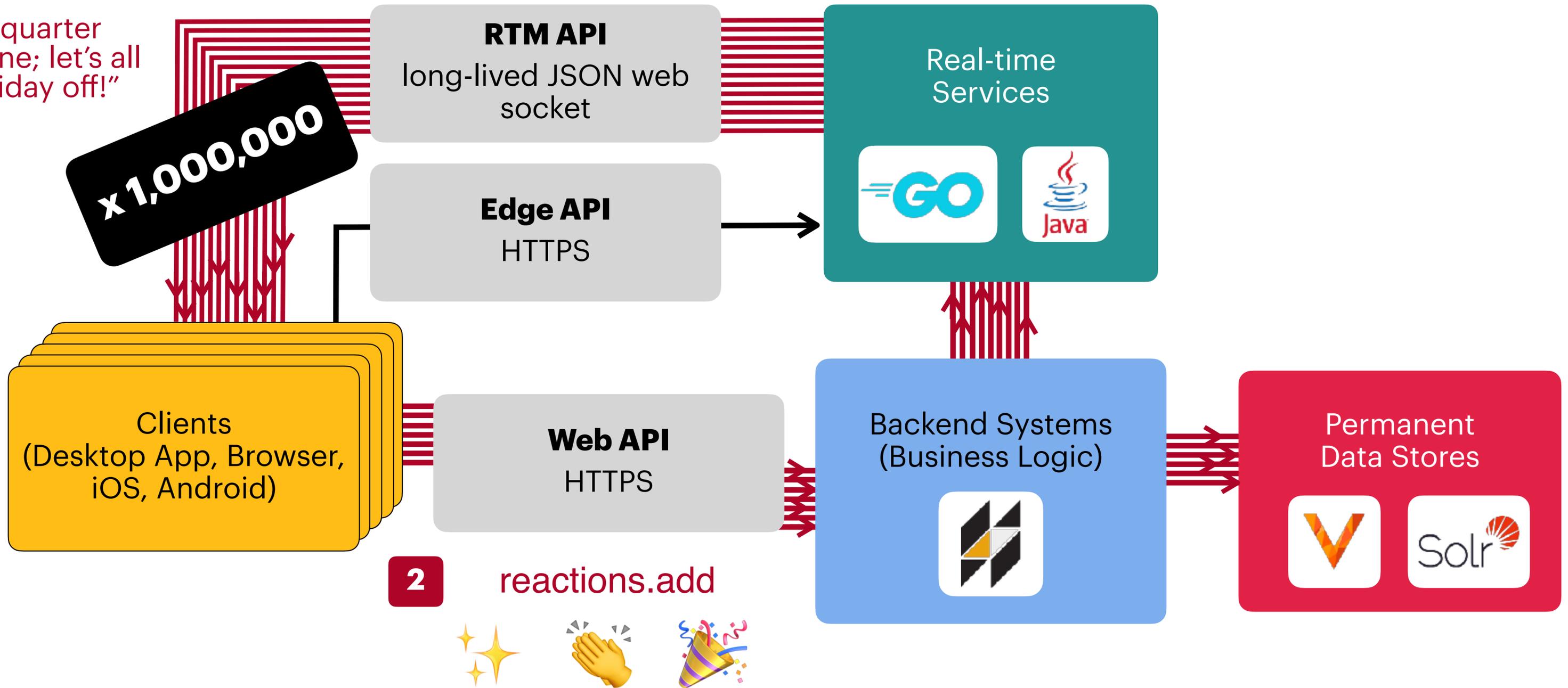
1

“Great quarter everyone; let’s all take Friday off!”

3



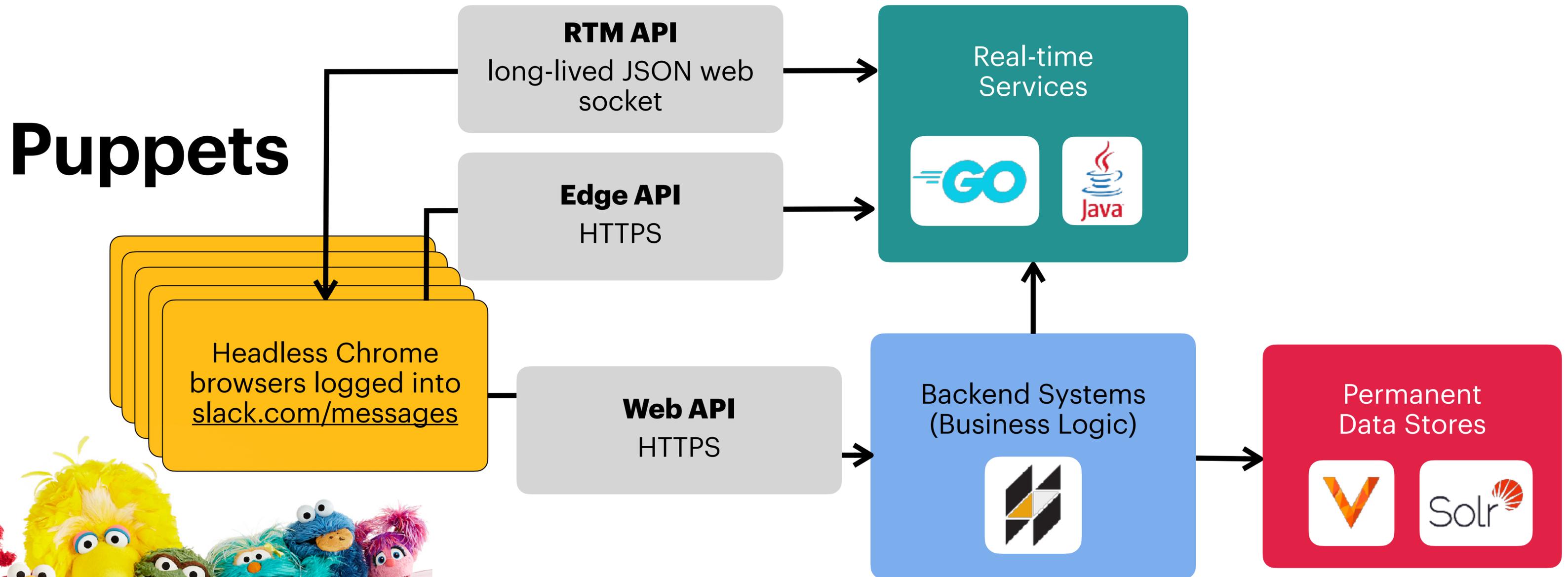
x1,000,000



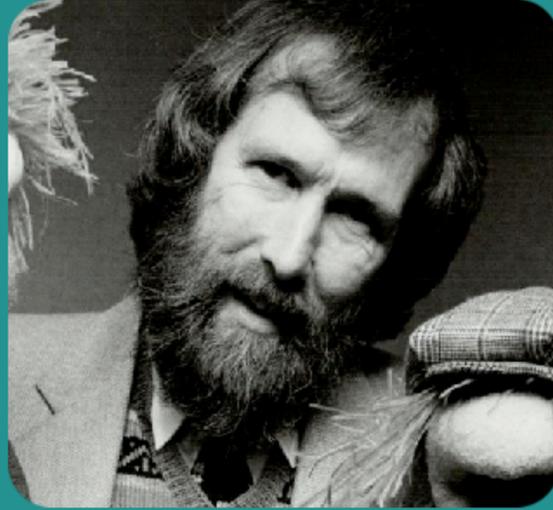
So we built Puppet Show.
***mid 2019**



Puppet Show within Slack's architecture



Puppeteer



**“Hey!
I’m DMing Oscar.
Have anything new for
me?”**

Puppet



Puppet



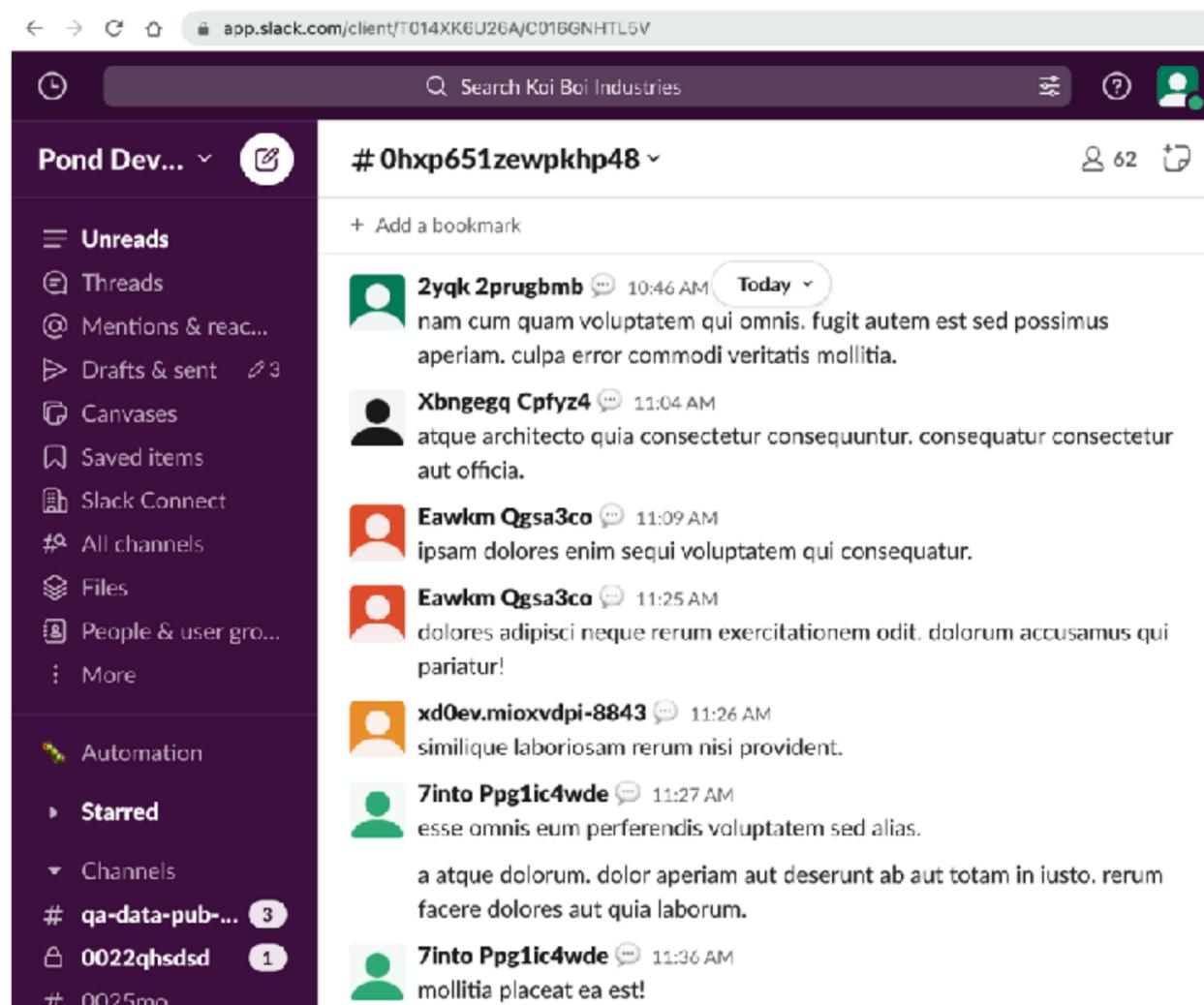
Puppet



Pros

High fidelity!

Flexible scripting!



Cons

Money-burning machine!

Wrangling thousands of Chrome instances!



**About a year later,
just a few weeks into the global pandemic,
we signed an ... even *bigger* customer.**

Support **500,000 users in the
same Slack instance.**

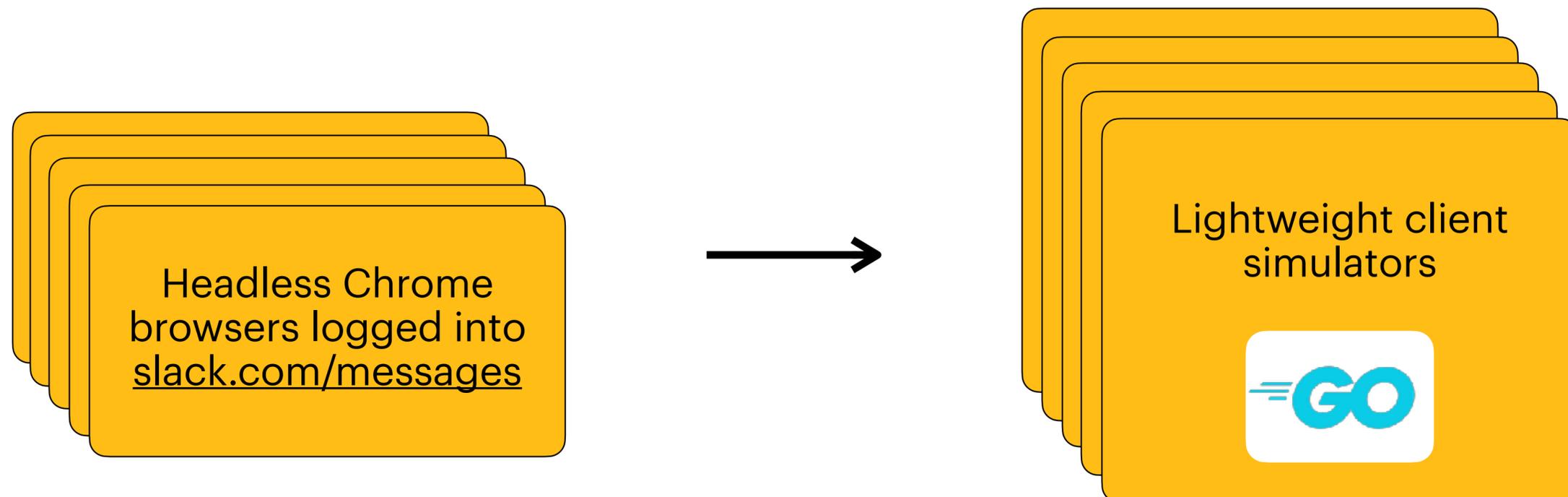
About a year later,
just a few weeks into the global pandemic,
we signed an ... even *bigger* customer.

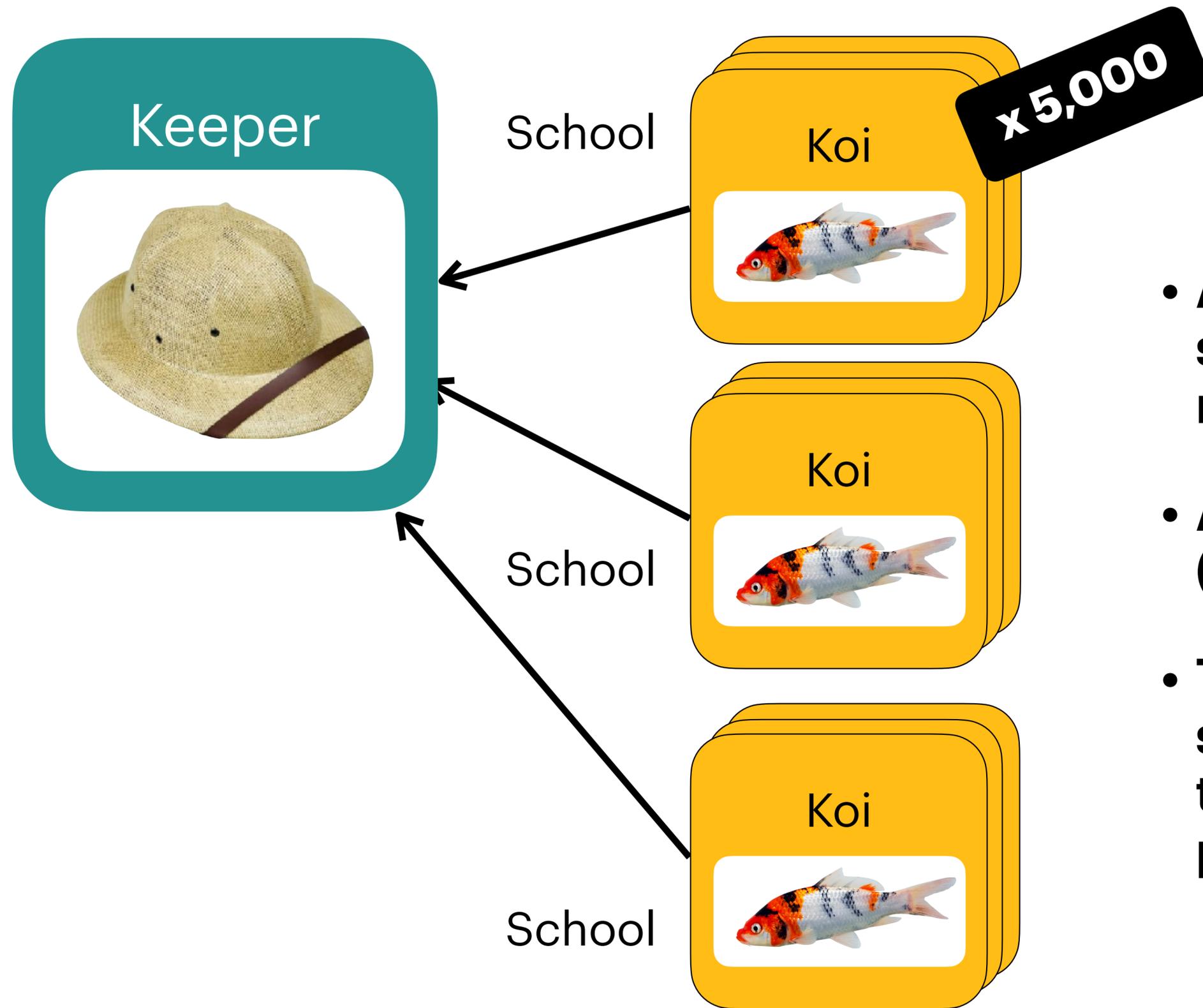
Support **500,000** users in the
same Slack instance

10 weeks



We designed and began building Koi Pond.

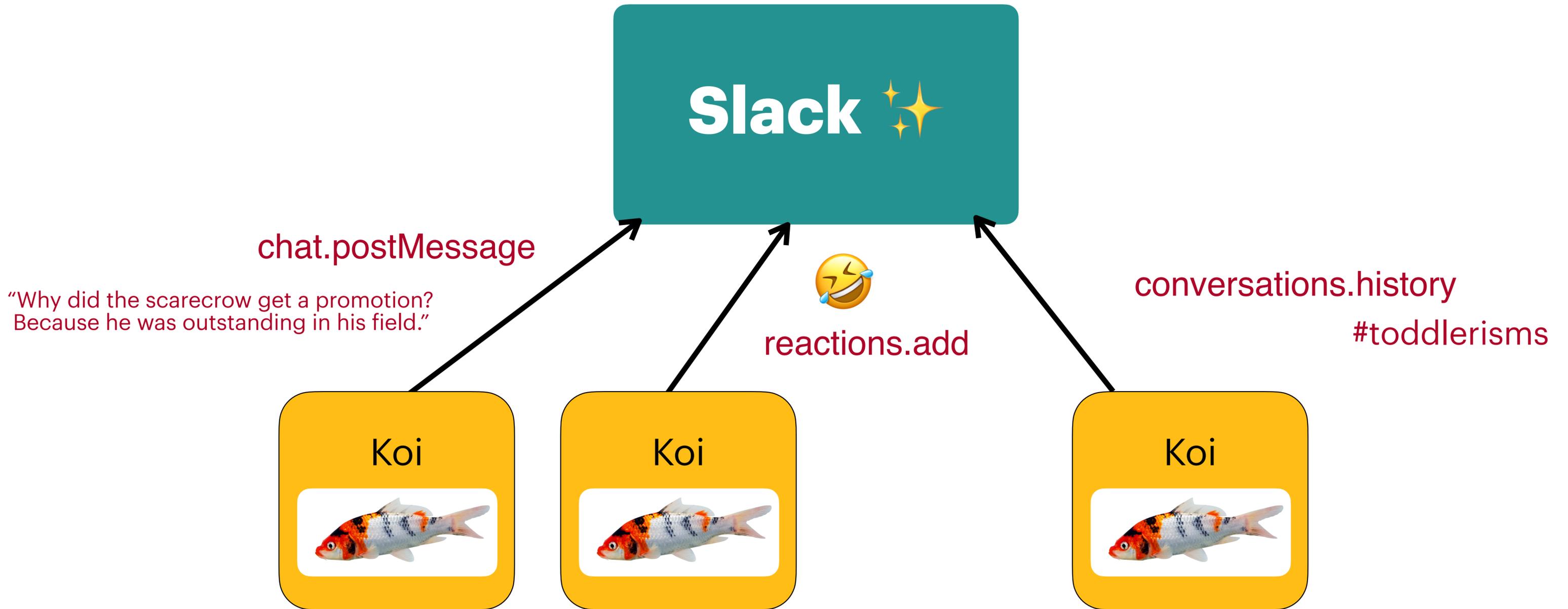




- A ***koi*** is a Slack client simulation (a single Go routine).
- A ***school*** is a collection of koi (a single Go program).
- The ***keeper*** manages the schools and keeps track of the overall load test state and parameters.

```
2
3 {
4   .. "behaviors" : .. {
5     .. "chat.postMessage" : .. {
6       .. "frequency" : .. 0.043
7     .. }
8   .. },
9   .. "sequences" : .. {
10    .. "chat.postMessage" : .. {
11      .. "doc" : .. "Sends a message to a random channel.",
12    > .. "steps" : .. [ ...
33    .. .. ]
34    .. .. }
35    .. .. }
36  }
37
```

Just *real* enough



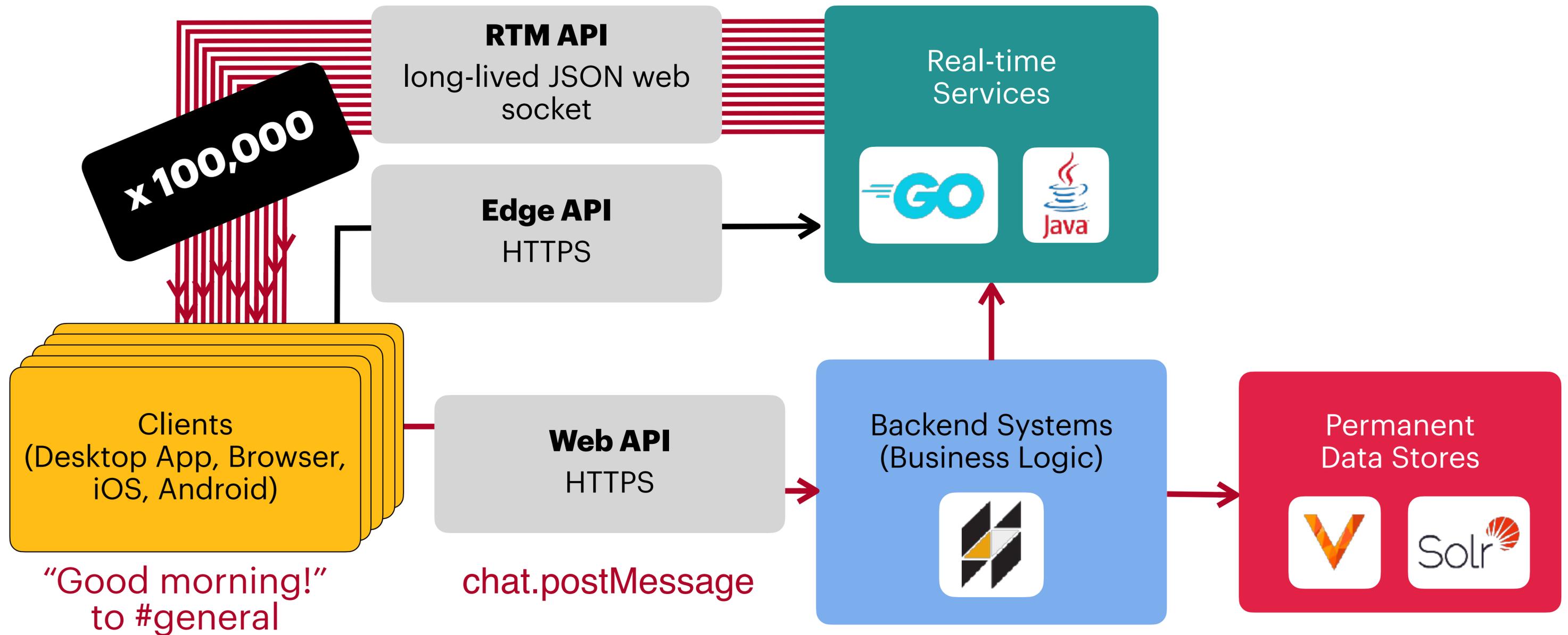
{...} default-config.json M ×

config > {...} default-config.json > ...

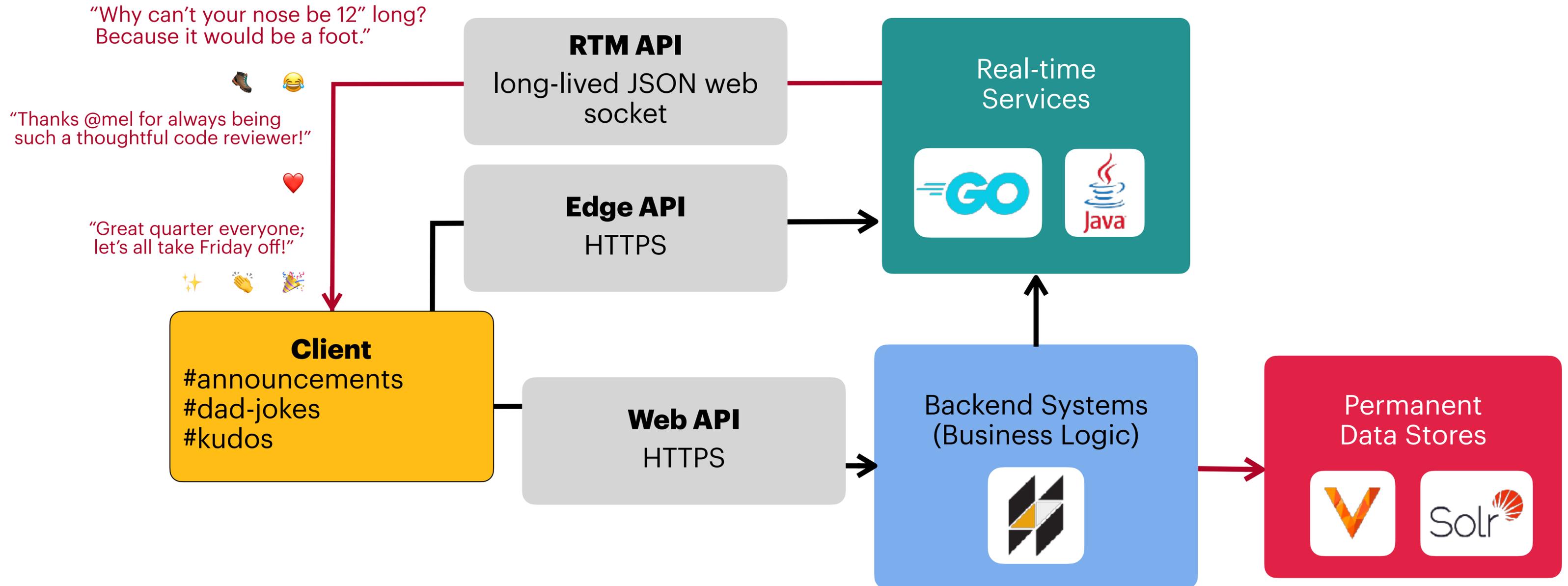
```
1 {
2 > .. "behaviors" : { ...
3497 .. },
3498 > .. "sequences" : { ...
4628 .. },
4629 > .. "weights" : { ...
4640 .. },
4641 // > .. "clients" : { ...
4645 .. }
4646 }
```

4646

1. Massive fan-out



2. Event floods



```
3
4  {
5  .. "formations": [
6  .. .. {
7  .. .. .. "name": "Populate announcement channel
with reactions",
8  .. .. .. "begin_within_secs": 30,
9  .. .. .. "percent": 1.0,
10 .. .. .. "sequence": {
11 .. .. .. .. "steps": [ ...
42 .. .. .. ]
43 .. .. .. }
44 .. .. }
45 .. ]
```



How about a live demo?



Koi Pond

Load testing with lightweight Slack client simulators

STATUS	NAME	SLACK ENV	OWNER	CONTINUOUS OR ONE-OFF	STARTED AT	ELAPSED TIME	WORKERS			
RUNNING	500k-continuous-koi	commercial_prod	gsanford	CONTINUOUS	Mar 9, 2023 2:33 PM 7 days ago	7 days	97	Pause Boot	Stop	Details

[New Load Test](#) [New Formation](#) [Show Default Configuration](#)

Useful Links

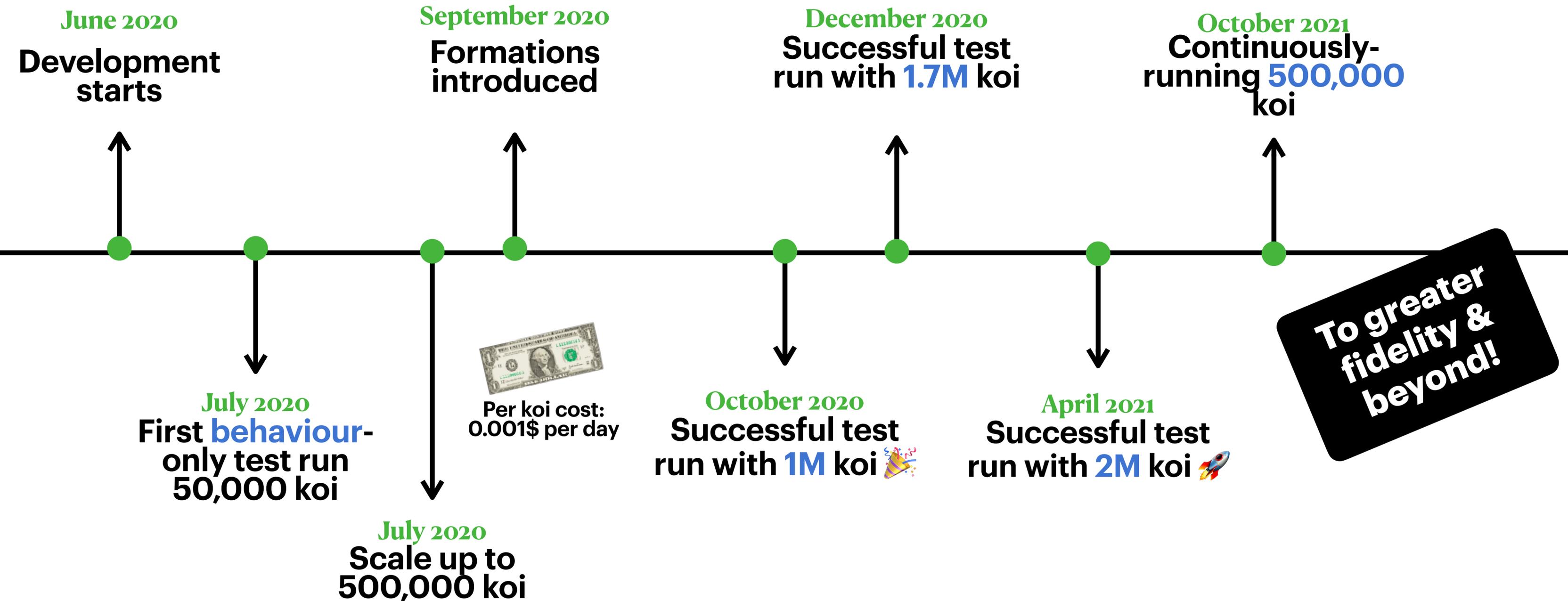
- [Main Grafana Dashboard](#)
 - [API traffic on `loadtest` nest](#)
 - [Koi Boi Industries Grafana Dashboard](#)
 - [Koi Boi Industries \(Mission Control\)](#)
 - [Emergency Stop](#)
 - [Deploy Health Dashboard](#)
-
- [Load test isolation architecture diagram](#)
 - [Build pipelines](#)



Were we successful?



Were we successful?



*timeline not to scale

Merci

You can find me most
places on the internet as
@qcmaude

