Kubernetes CronJobs

Does Anyone *Actually* Use This [In Production]?

Kevin Yang, Lyft

Kevin Yang

Software Engineer, Lyft

@yangkeIO

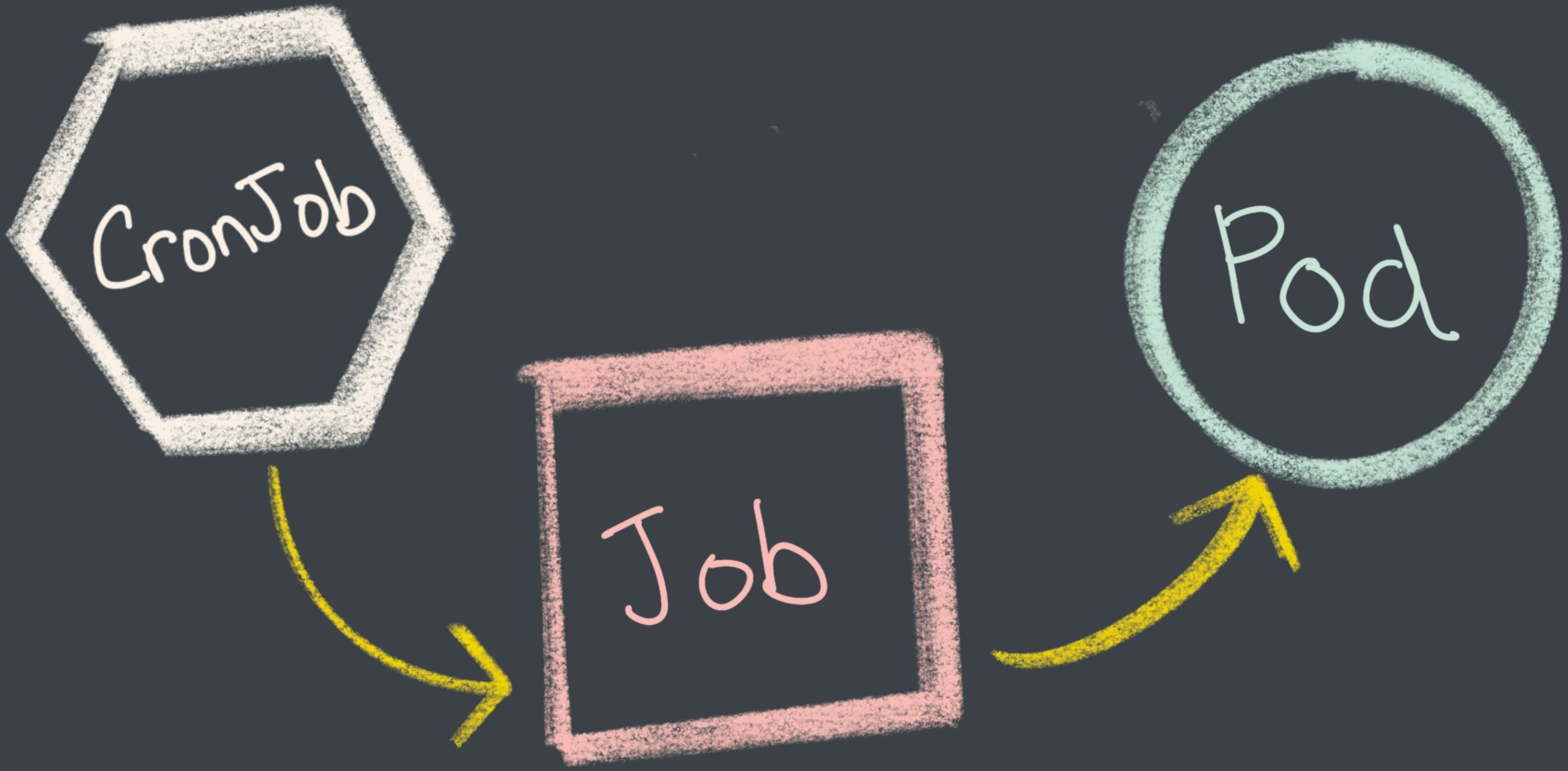
kyang@lyft.com

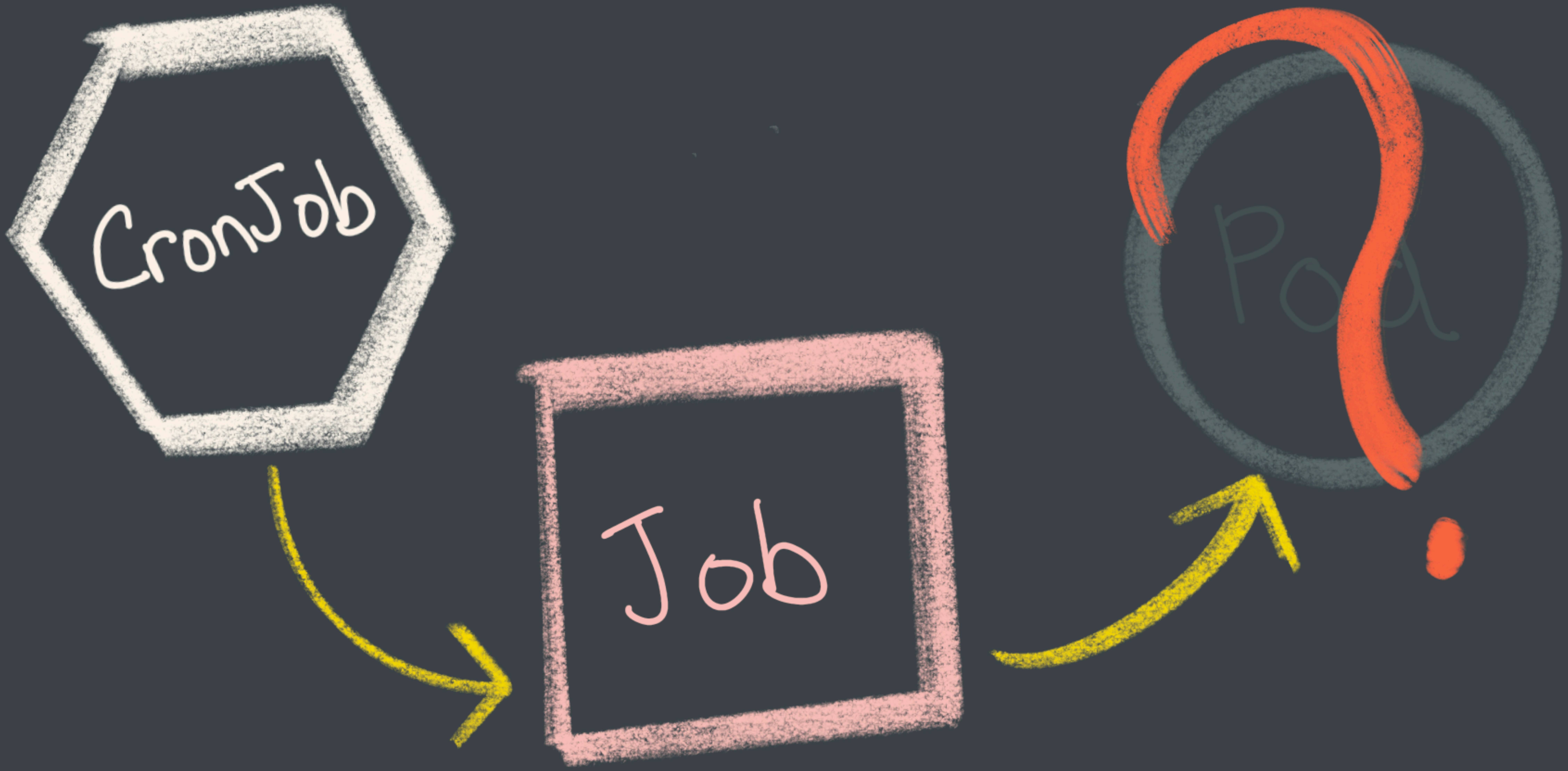
Motivation

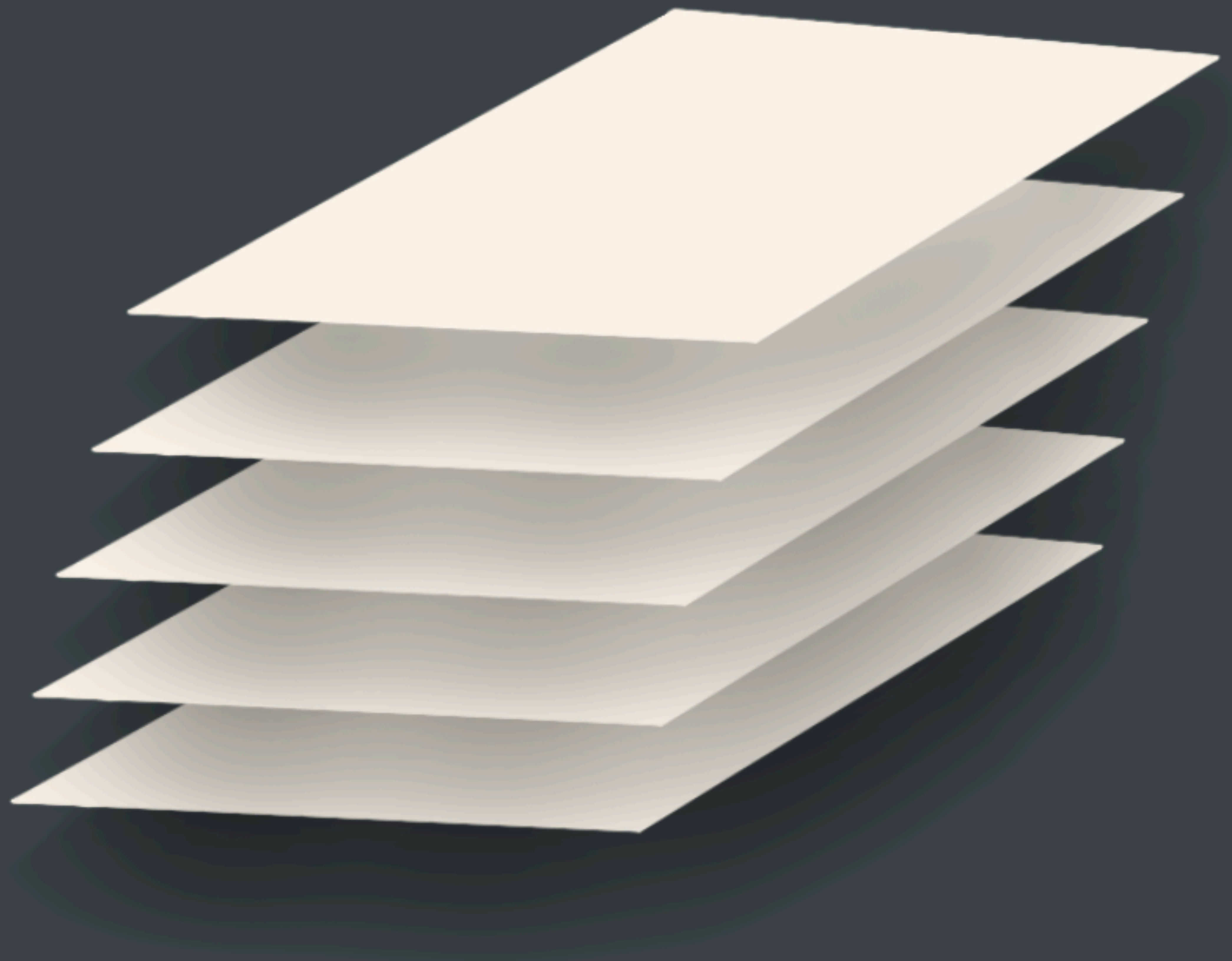
What is this talk?

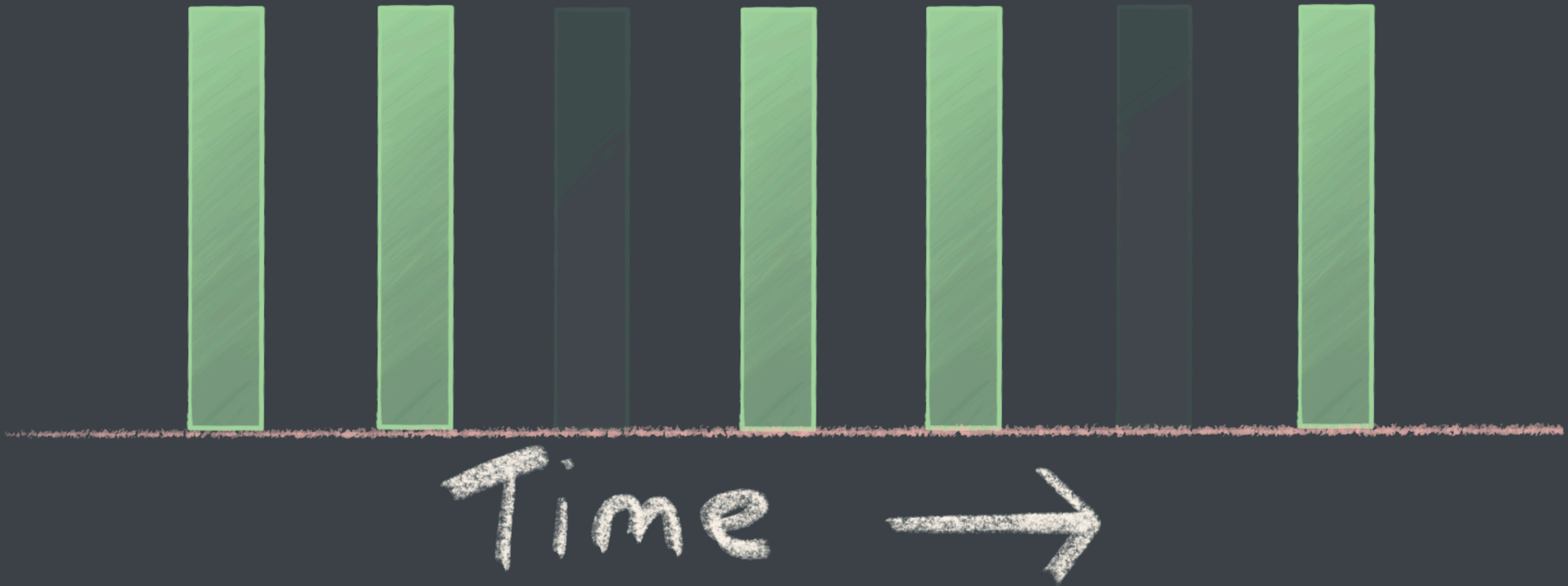
Why should you care?

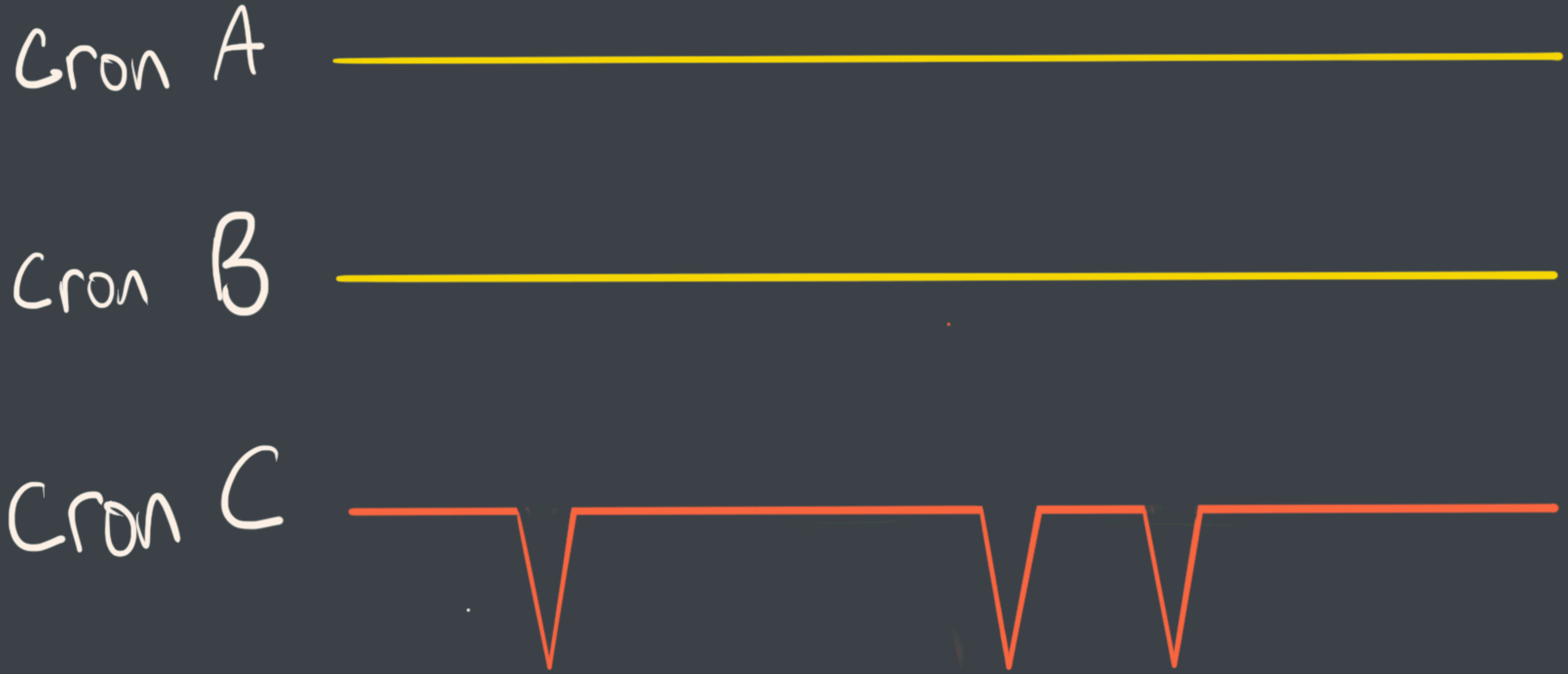
Story Time

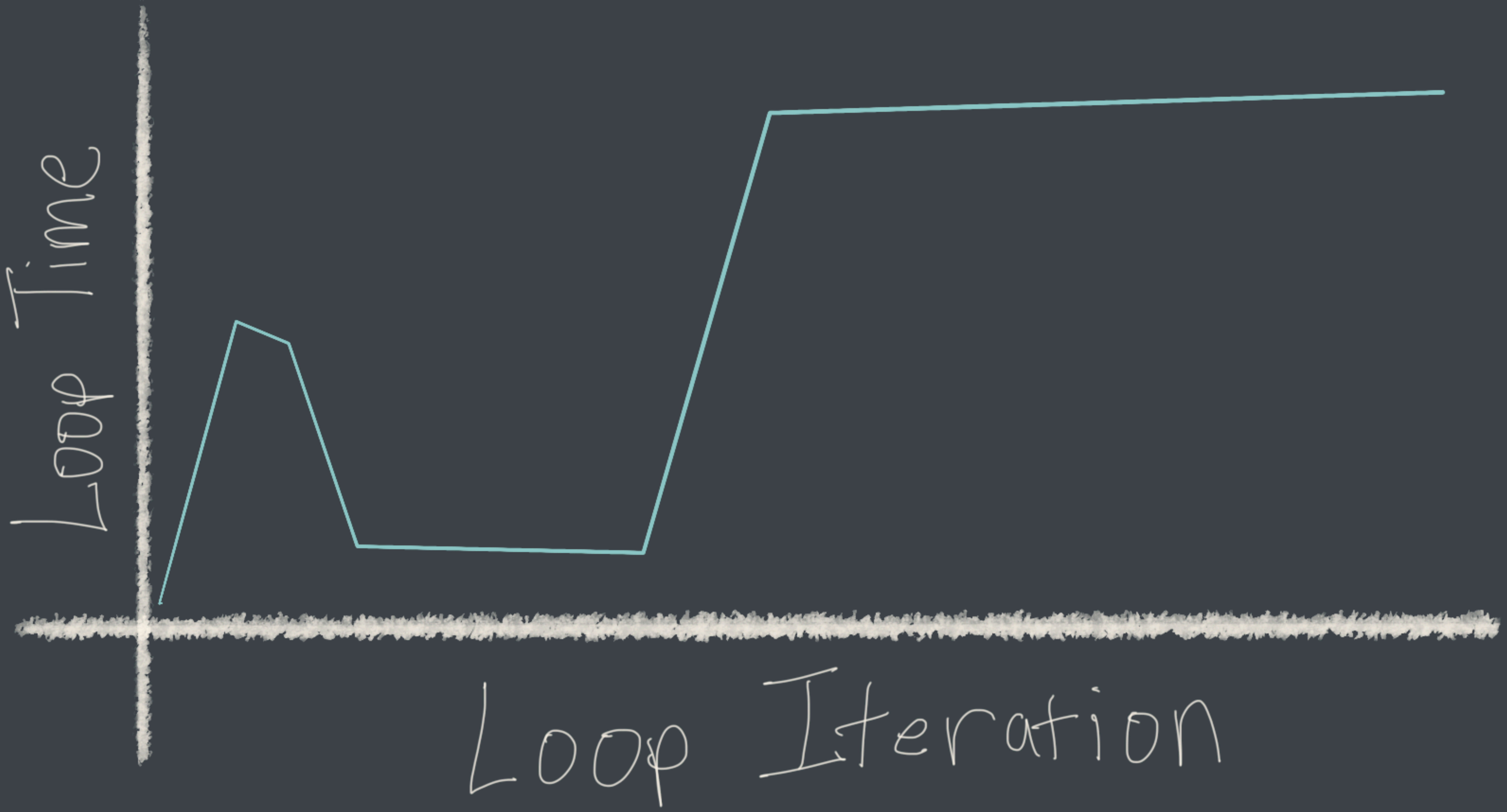

CronJob
Job
Pod

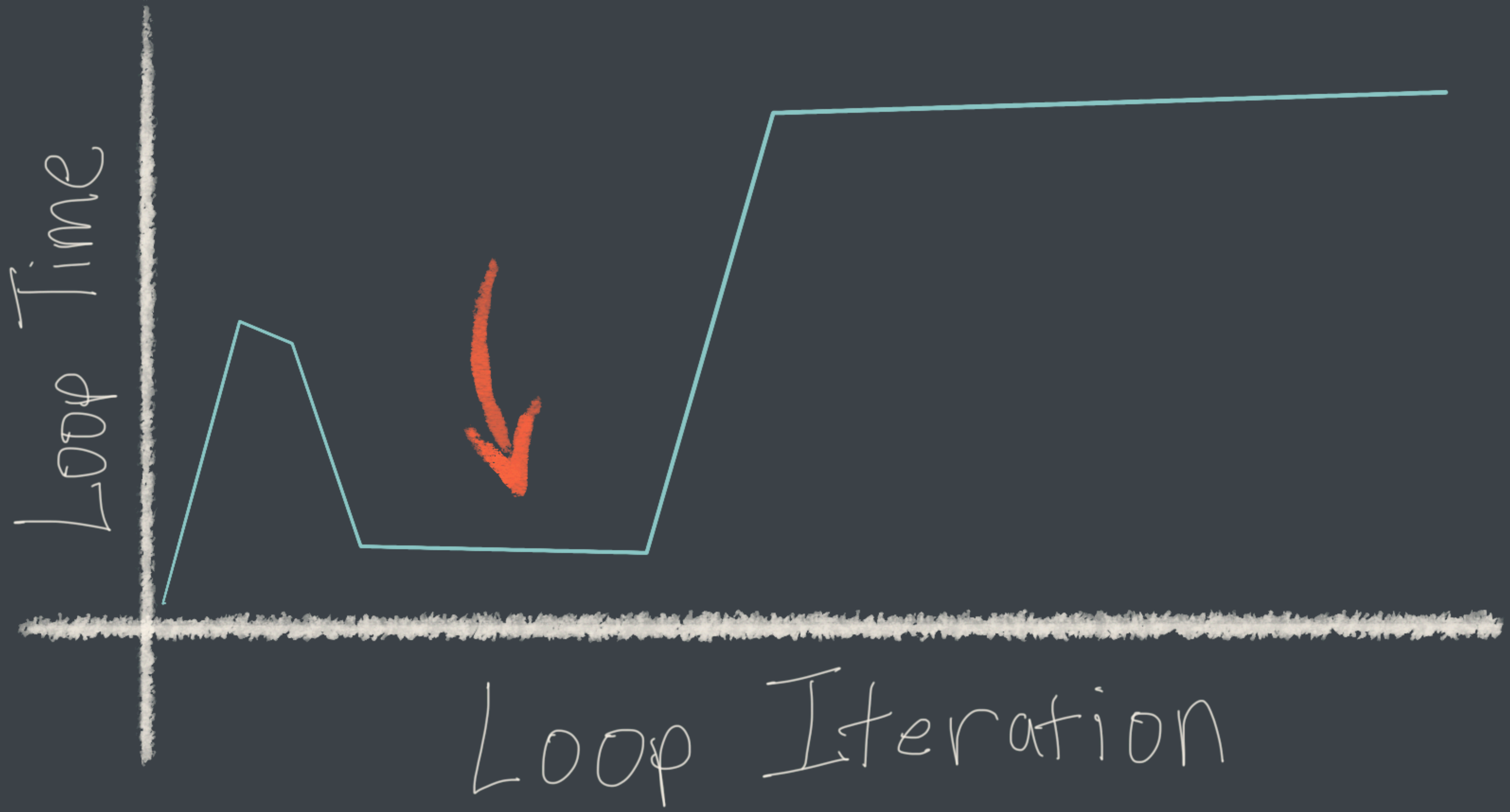

CronJob
Job
Pod

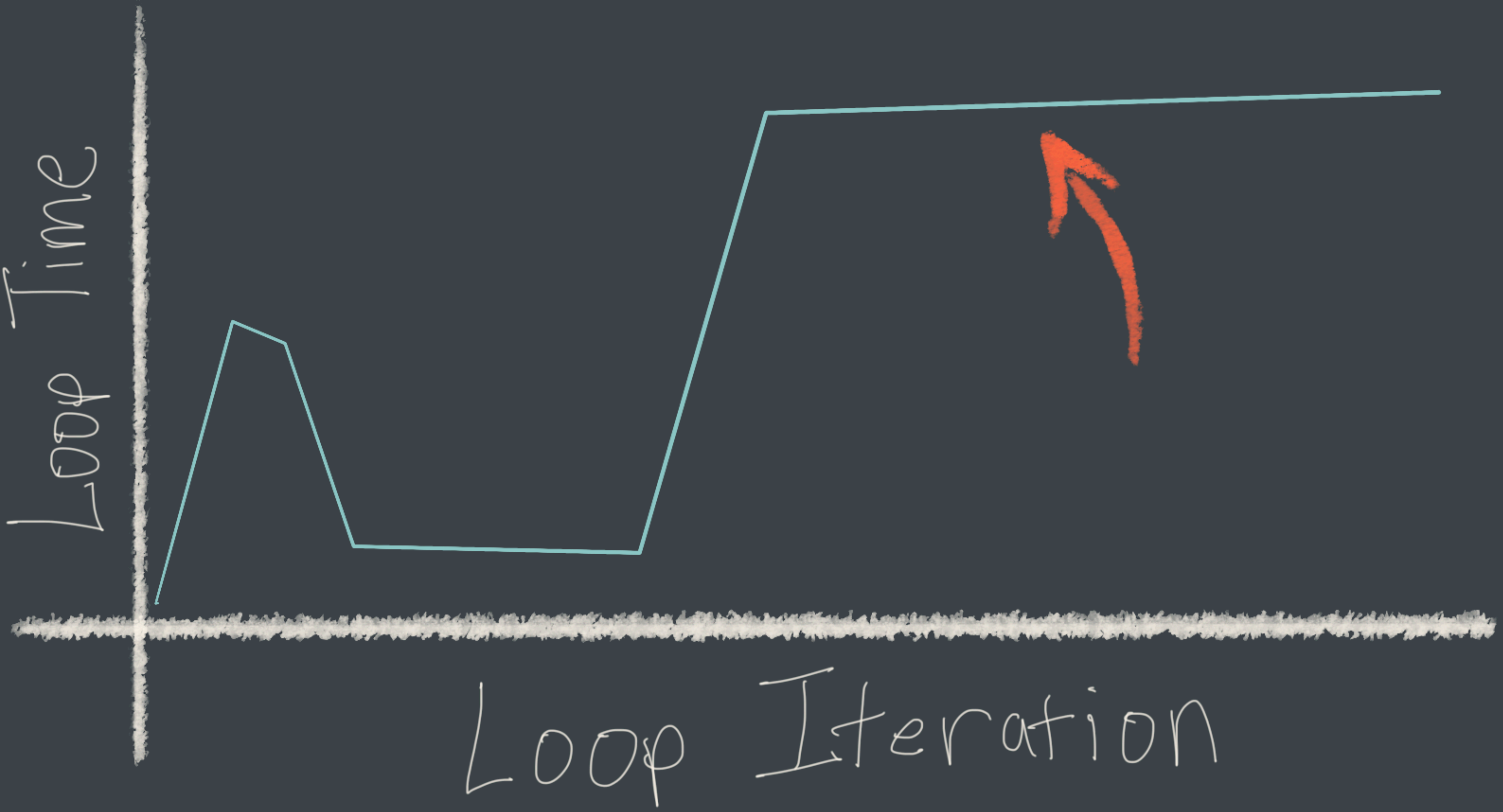


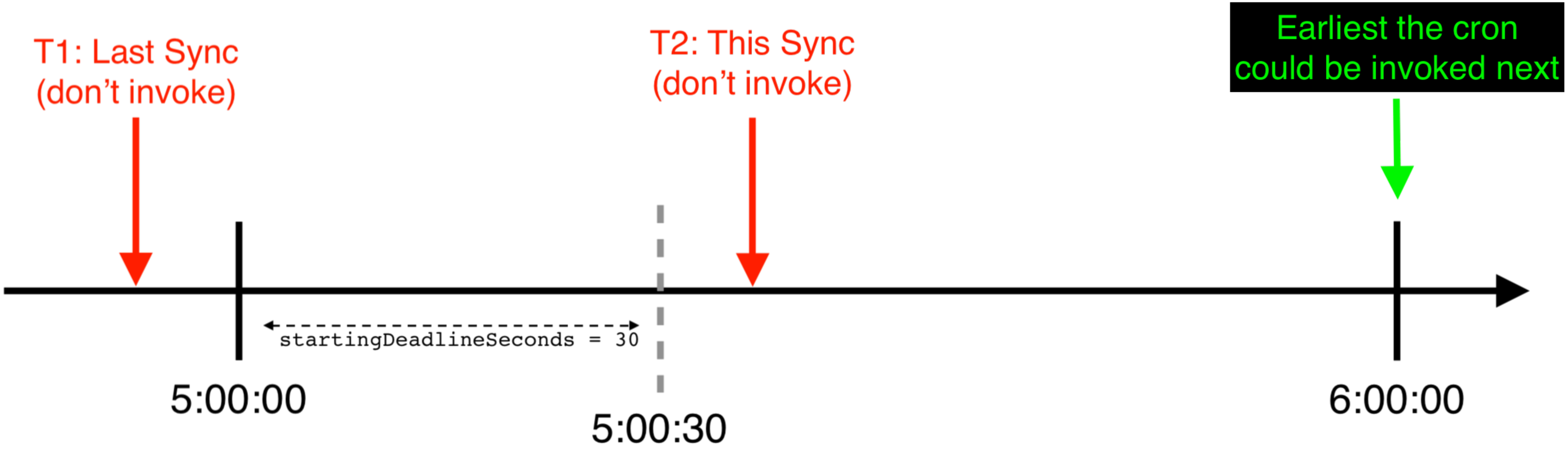

Time →


Cron A
Cron B
Cron C


Loop Time
Loop Iteration


Loop Time
Loop Iteration

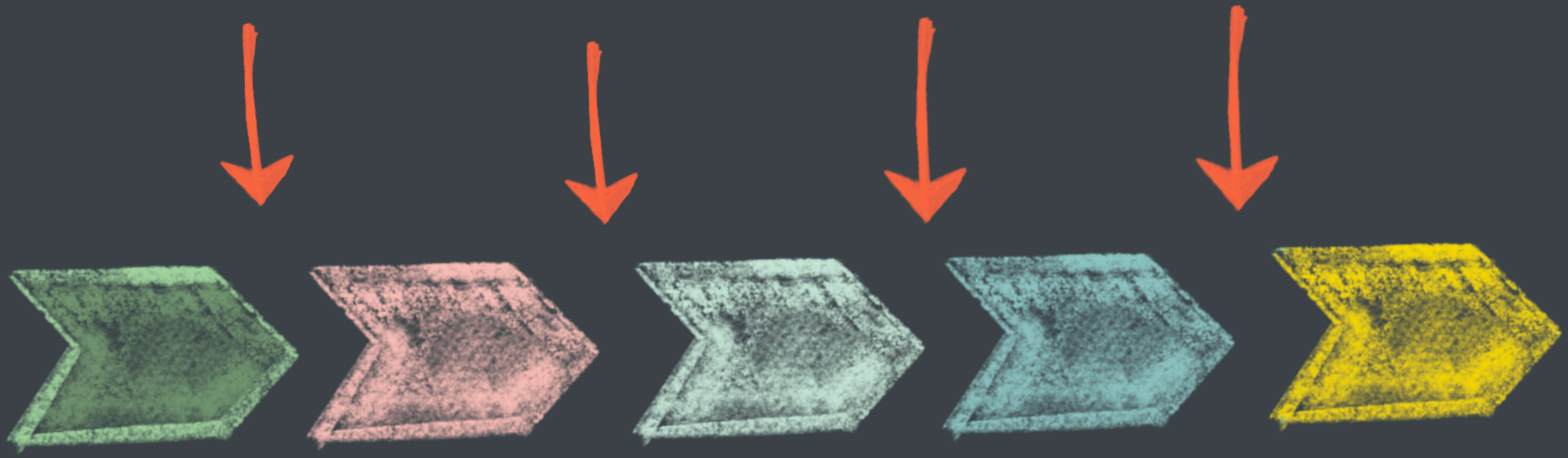

Loop Time
Loop Iteration

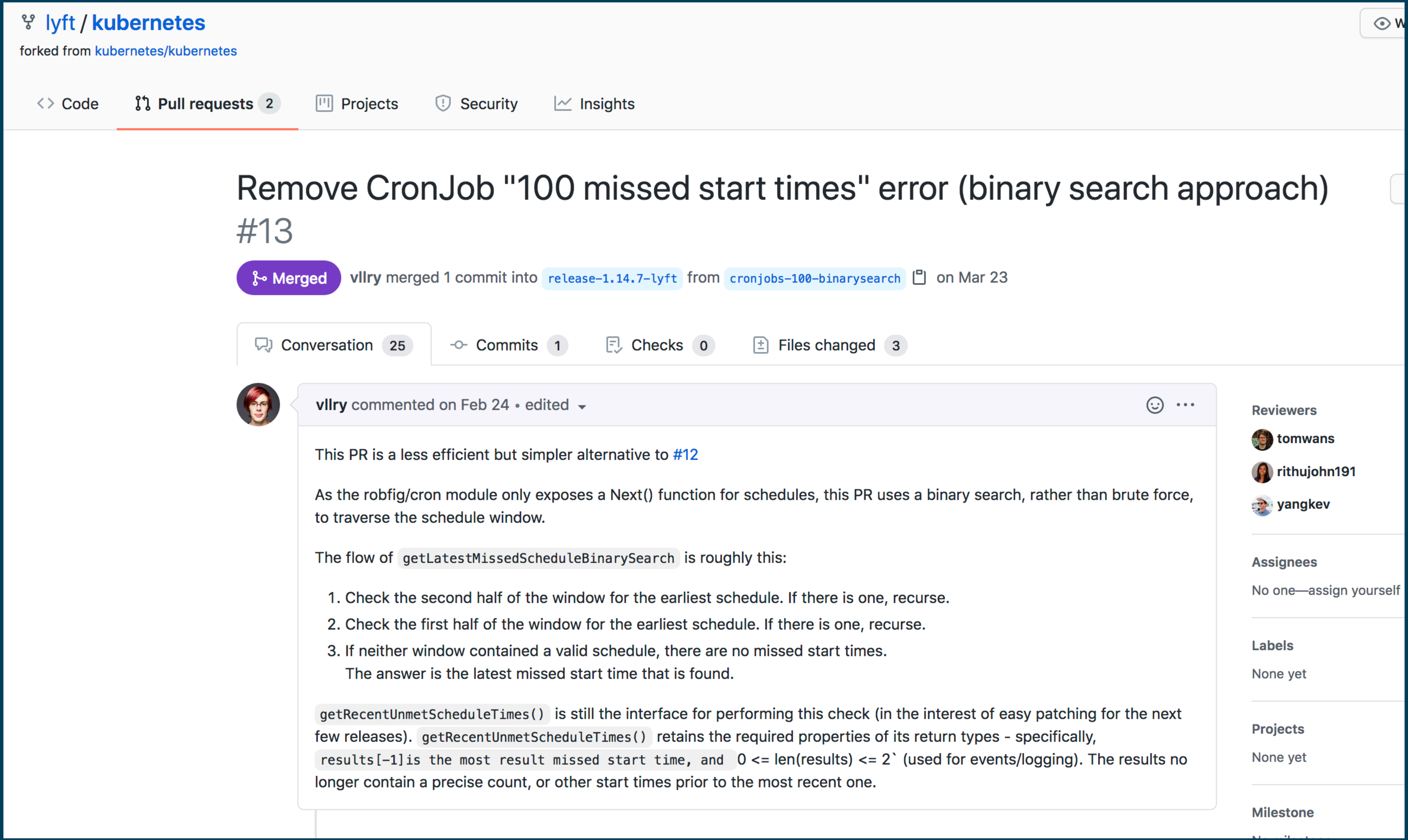

T1: Last Sync
(don't invoke)
T2: This Sync
(don't invoke)
Earliest the cron
could be invoked next
startingDeadlineSeconds = 30
5:00:00
5:00:30
6:00:00

What did these experiences tell us?

- Can fail in many ways:
  - `TooManyMissedStarts`
  - API client rate-limiting
  - …
- CronJobs are difficult to **monitor**, **understand**, and **debug**

What we did





Remove CronJob "100 missed start times" error (binary search approach) #13

Merged vllry merged 1 commit into `release-1.14.7-lyft` from `cronjobs-100-binarysearch` on Mar 23

vllry commented on Feb 24 • edited

This PR is a less efficient but simpler alternative to #12

As the robfig/cron module only exposes a Next() function for schedules, this PR uses a binary search, rather than brute force, to traverse the schedule window.

The flow of `getLatestMissedScheduleBinarySearch` is roughly this:

1. Check the second half of the window for the earliest schedule. If there is one, recurse.
2. Check the first half of the window for the earliest schedule. If there is one, recurse.
3. If neither window contained a valid schedule, there are no missed start times.
   The answer is the latest missed start time that is found.

`getRecentUnmetScheduleTimes()` is still the interface for performing this check (in the interest of easy patching for the next few releases). `getRecentUnmetScheduleTimes()` retains the required properties of its return types - specifically, `results[-1]is the most result missed start time, and` 0 <= len(results) <= 2` (used for events/logging). The results no longer contain a precise count, or other start times prior to the most recent one.

Reviewers

tomwans

rithujohn191

yangkev

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

Ok, so what does this mean for me?

What do your users need?

Alerts

- Cron did not run on schedule
- Cron ran but failed
- Cron ran but took longer than usual

What do your users need?

Failure recovery

- Ability to re-run a failed CronJob

On Evaluating Cron Solutions

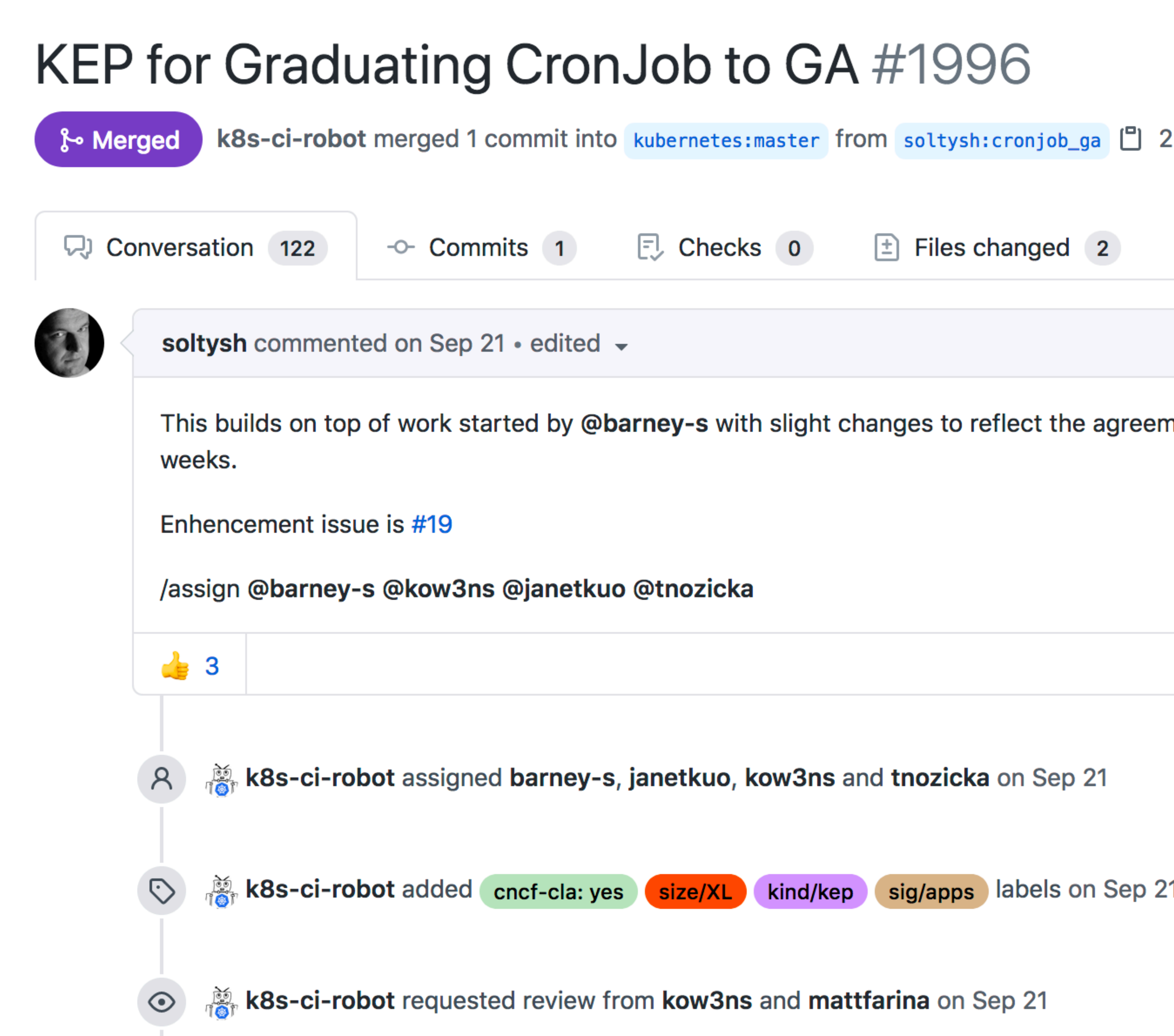
What *should* the User Experience be?

What *should* the Operator Experience be?

To Conclude…

Not all hope is lost

KEP to graduate CronJob to GA



KEP for Graduating CronJob to GA #1996

Merged k8s-ci-robot merged 1 commit into `kubernetes:master` from `soltysh:cronjob_ga` 2

soltysh commented on Sep 21 • edited

This builds on top of work started by @barney-s with slight changes to reflect the agreem
weeks.

Enhencement issue is #19

/assign @barney-s @kow3ns @janetkuo @tnozicka

👍 3

k8s-ci-robot assigned barney-s, janetkuo, kow3ns and tnozicka on Sep 21

k8s-ci-robot added cncf-cla: yes size/XL kind/kep sig/apps labels on Sep 21

k8s-ci-robot requested review from kow3ns and mattfarina on Sep 21

Kubernetes is *no silver bullet*

Thank You!

lyft.com/careers