

The Problem with **PRE-AGGREGATED METRICS**

AND A GLORIOUS ALTERNATIVE FUTURE

The Problem ^S with **PRE-AGGREGATED METRICS**

THE
~~AND A~~ GLORIOUS ALTERNATIVE FUTURE



Parse Analytics

`requests.appserver50.reads.count`

(time)

↓

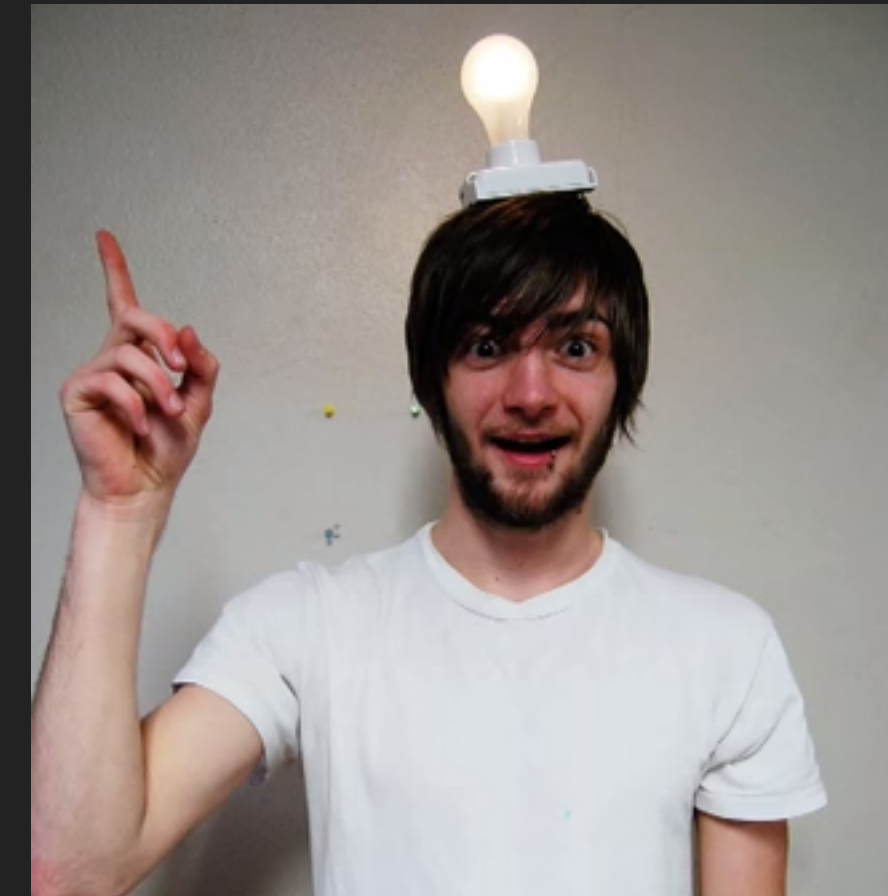
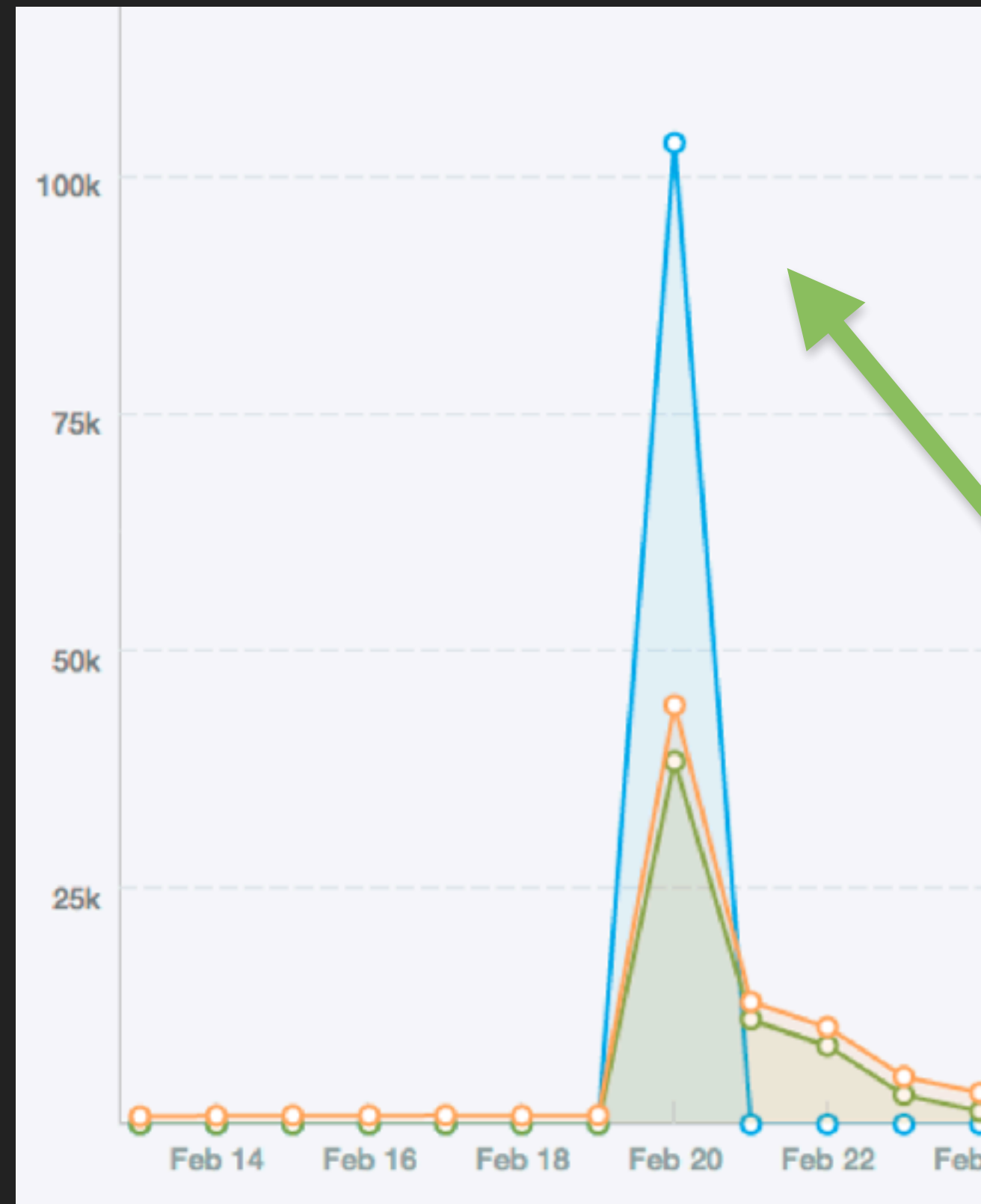
+1

+1

+1 +1 +1

+1 +1 +1 +1 +1 +1 +1 +1 +1 +1 +1

...



(time) —————→

requests.appserver50.reads.count

+++++

requests.appserver50.errors.count

+ + + + +

requests.appserver50.rejected.count

+ ++ + + +

requests.appserver50.latency.avg

+++++

requests.ceo.latency.avg

+ ++ +++ +

requests.x86.64.memory.kb.count

+++++

requests.us-east-1.errors.count

+ + ++

requests.mozilla-53-4.count

++ + +

The Problem with **PRE-AGGREGATED METRICS**

If I send a push notification, how many app-opens does my app get?

Have people been posting more photos with my app lately?

Has my app been getting more popular?

PART 1: THE PROBLEM WITH "PRE"

- MyApp.PushSent
- MyApp.PushSent.iOS
- MyApp.AppOpened
- MyApp.AppOpened.iOS
- MyApp.APIRequest
- MyApp.APIRequest.Write
- MyApp.APIRequest.iOS

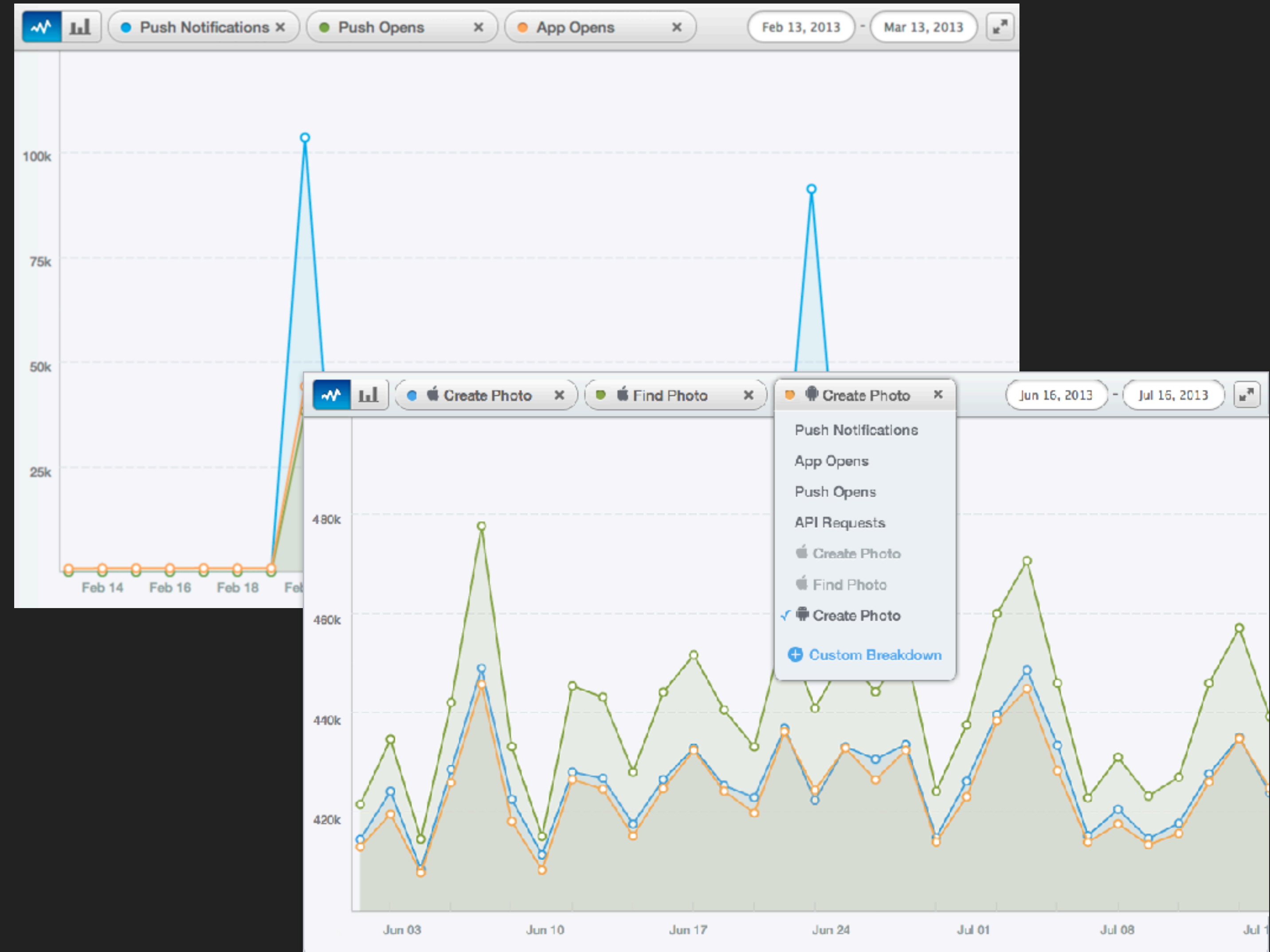
If I send a push notification...

... how many app-opens does my app get?

... have people been posting more lately?

PART 1: THE PROBLEM WITH "PRE"

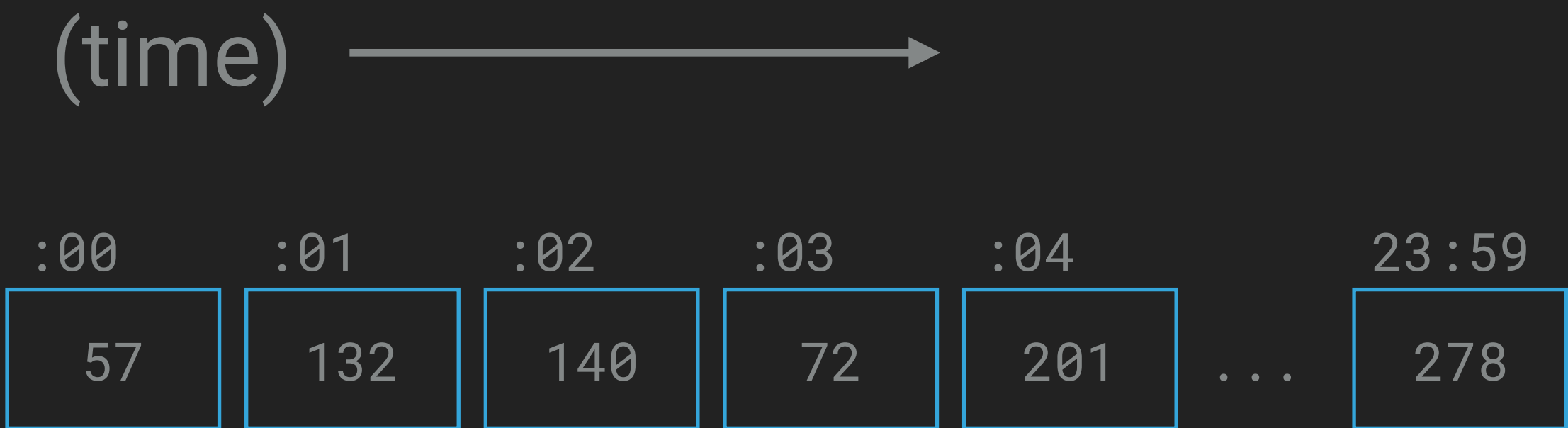
- MyApp.PushSent
- MyApp.PushSent.iOS
- MyApp.AppOpened
- MyApp.AppOpened.iOS
- MyApp.APIRequest
- MyApp.APIRequest.Write
- MyApp.APIRequest.iOS



PRE-aggregation = WRITE-TIME aggregation

PART 1: THE PROBLEM WITH "PRE"

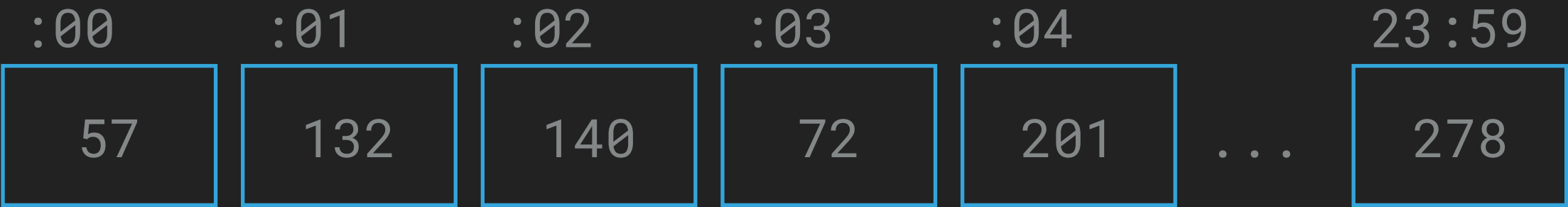
MyApp.APIRequest



PART 1: THE PROBLEM WITH "PRE"

(time) →

MyApp.APIRequest



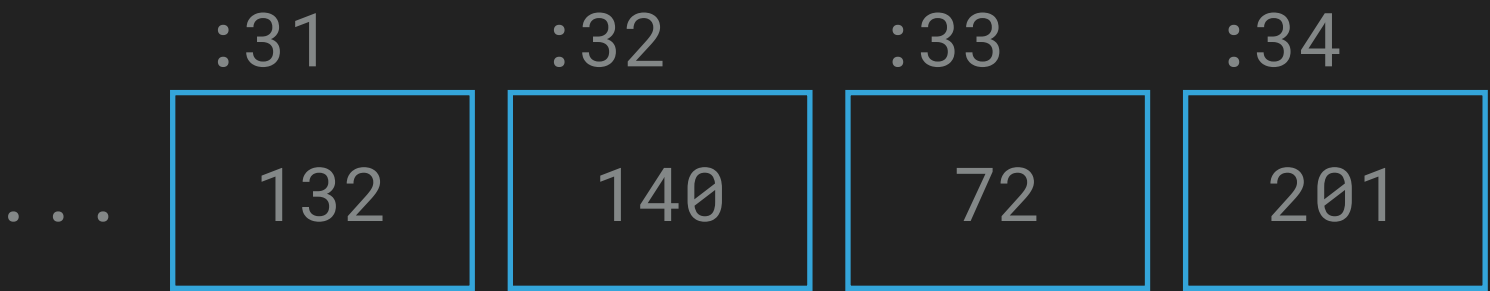
MyApp.APIRequest:Write

?

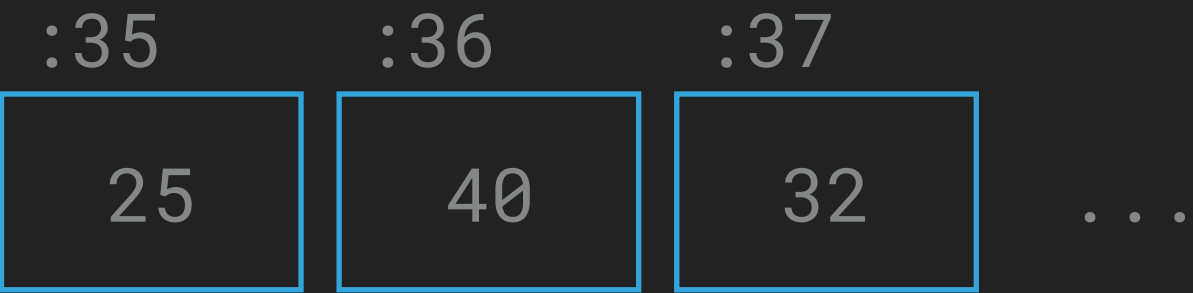
PART 1: THE PROBLEM WITH "PRE"

(time) →

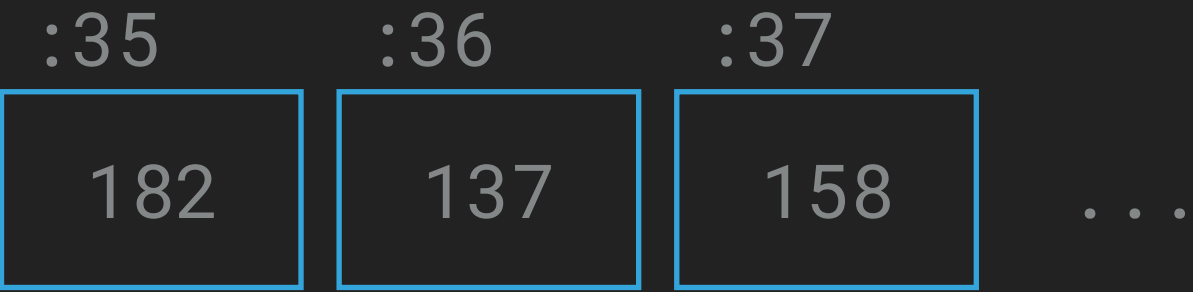
MyApp.APIRequest.Write



MyApp.APIRequest.Insert



MyApp.APIRequest.Update



PART 1: THE PROBLEM WITH "PRE"

(time) →

MyApp.APIRequest.Write

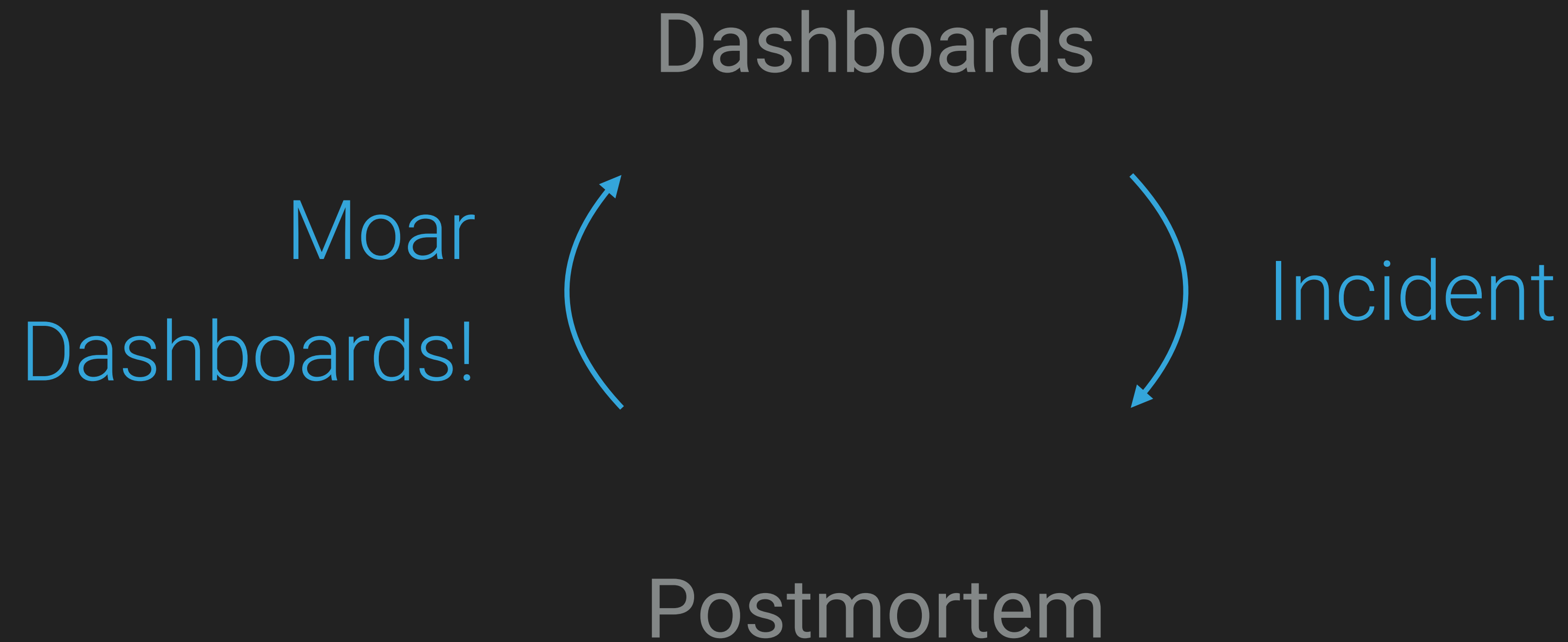
...	:31	:32	:33	:34	:35	:36	:37
	132	140	72	201	178	152	123

MyApp.APIRequest.iOS

...	:31	:32	:33	:34	:32	:33	:34
	279	290	182	317	273	201	280

MyApp.APIRequest.Write.iOS

	:35	:36	:37	...
	122	137	118	



The Problem with **PRE-AGGREGATED METRICS**

PART 1: THE PROBLEM WITH "PRE"

```
{ "endpoint":      "/purchase/complete",  
  "status":        201,  
  "client_platform": "iOS",  
  "request_dur_ms": 32.153,  
  "txn_dollars":    172.83,  
  "build_id":       1325,  
  "hostname":       "appserver50" }
```



```
requests.count++  
requests.appserver50.count++  
requests.ios.count++  
requests.success.count++  
requests.latency.avg += 32.153
```

The Problem with **PRE-AGGREGATED METRICS**

PART 2: THE PROBLEM WITH "AGGREGATED"

an API Request:

```
{  
  "hostname": "appserver50",  
  "method": "POST",  
  "sdk": "ios",  
  "status_code": 201,  
  "request_dur_ms": 32.153,  
  "payload_bytes": 172,  
}
```

PART 2: THE PROBLEM WITH "AGGREGATED"

API Request: processed a Write from an iOS device for MyApp

APIRequest.MyApp +1

APIRequest.MyApp.Write +1

APIRequest.MyApp.iOS +1

APIRequest.MyApp.Write.iOS +1

PART 2: THE PROBLEM WITH "AGGREGATED"

API Request: processed a ~~Write~~^{Insert} from an iOS device for MyApp

APIRequest.MyApp	+1
APIRequest.MyApp. Write ^{Insert}	+1
APIRequest.MyApp.iOS	+1
APIRequest.MyApp. Write ^{Insert} .iOS	+1
APIRequest.MyApp.Update	+0
APIRequest.MyApp.Update.iOS	+0

PART 2: THE PROBLEM WITH "AGGREGATED"

API Request: processed a Insert from an iOS device for MyApp with our version 2.0 SDK

APIRequest.MyApp	+1
APIRequest.MyApp.Insert	+1
APIRequest.MyApp.iOS	+1
APIRequest.MyApp.Insert.iOS	+1
APIRequest.MyApp.SDKv2	+1
APIRequest.MyApp.Insert.SDKv2	+1
APIRequest.MyApp.iOS.SDKv2	+1

APIRequest.MyApp.Update.iOS.SDKv2 +1

`MyApp.CustomEvent.purchases.user962359.sess293486`

(̂ ◦ ◻ ◦) ̂ ̂ ̂ ̂ ̂)

PART 2: THE PROBLEM WITH "AGGREGATED"

MyApp.CustomEvent.purchases.user648988.sess848949

MyApp.CustomEvent.purchases.user629475.sess272738

MyApp.CustomEvent.purchases.user552404.sess265317

MyApp.CustomEvent.purchases.user253012.sess160889

MyApp.CustomEvent.purchases.user962359.sess293486

MyApp.CustomEvent.purchases.user881923.sess171031

MyApp.CustomEvent.purchases.user572034.sess471889

MyApp.CustomEvent.purchases.user867596.sess321446

MyApp.CustomEvent.purchases.user079640.sess708259

WAYS TO EXPLODE YOUR PRE-AGGREGATED METRICS STORAGE

WAYS TO EXPLODE YOUR PRE-AGGREGATED METRICS STORAGE

- Wanting to break down by some new attribute
 - (e.g. major SDK versions)

WAYS TO EXPLODE YOUR PRE-AGGREGATED METRICS STORAGE

- Wanting to break down by some new attribute
 - (e.g. major SDK versions)
- Tracking high-cardinality values
 - (e.g. userID, app version, user agent, referrer)

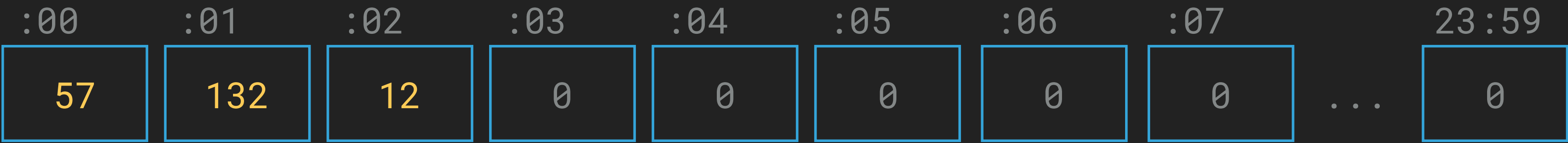
WAYS TO EXPLODE YOUR PRE-AGGREGATED METRICS STORAGE

- Wanting to break down by some new attribute
 - (e.g. major SDK versions)
- Tracking high-cardinality values
 - (e.g. userID, app version, user agent, referrer)

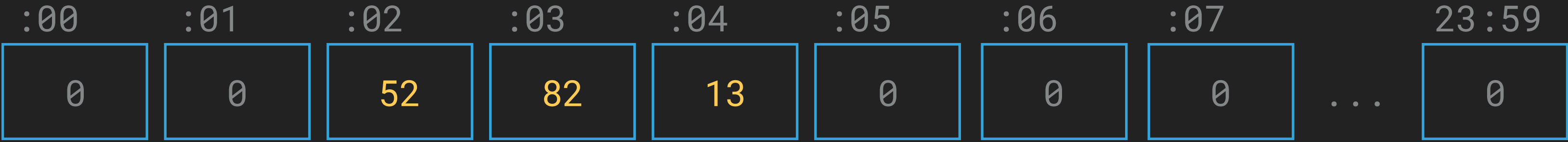
PART 2: THE PROBLEM WITH "AGGREGATED"

(time) →

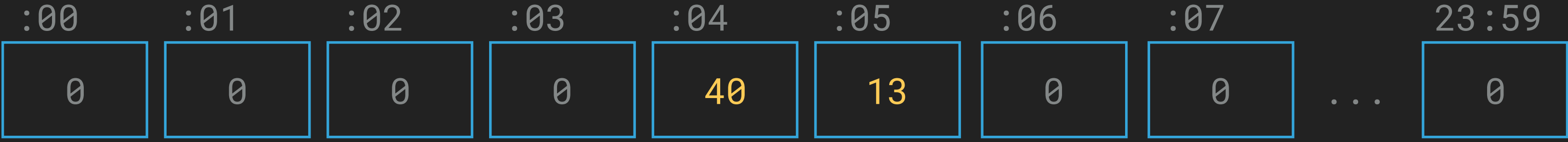
MyApp.build151



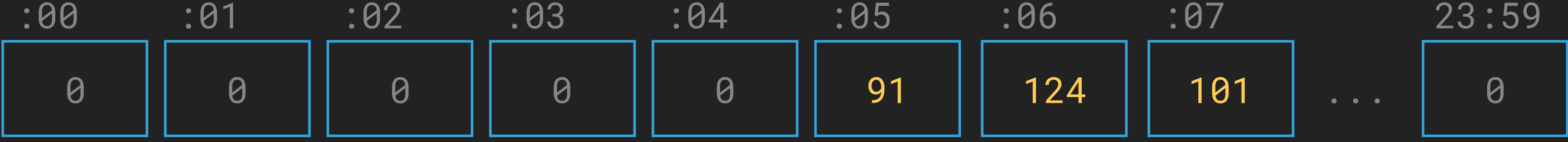
MyApp.build152



MyApp.build154



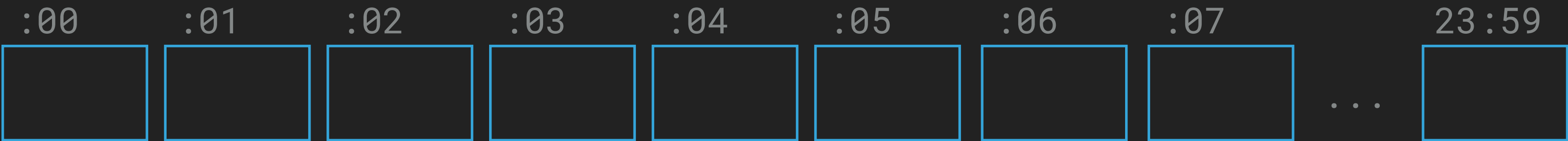
MyApp.build157



PART 2: THE PROBLEM WITH "AGGREGATED"

(time) →

MyApp.0xc42000fc0



MyApp.0xc420079af



MyApp.0x0012ff74c



MyApp.0x471cb03e7



The Problem with **PRE-AGGREGATED METRICS**

The Problem with **PRE-AGGREGATED METRICS**

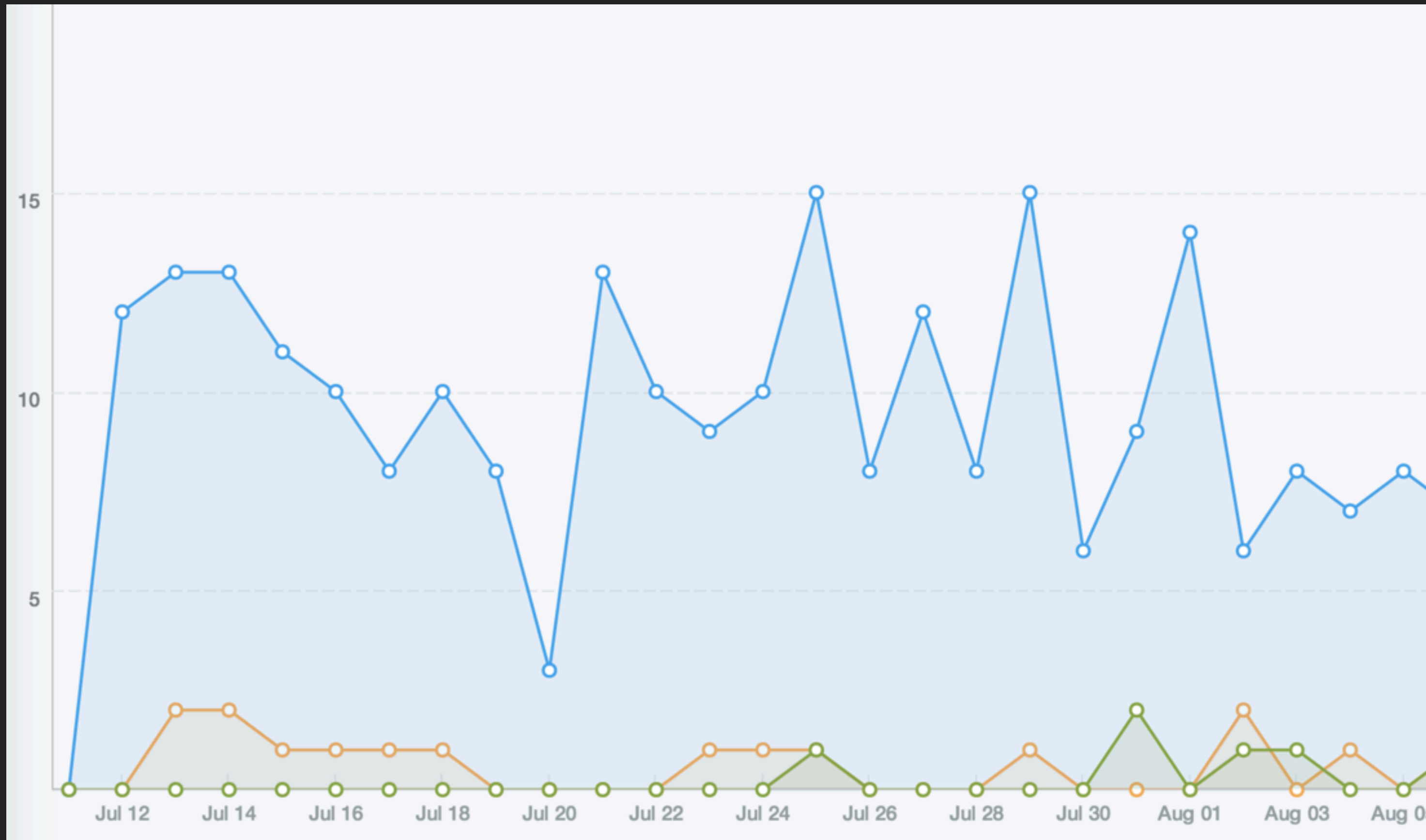
METRIC:

A single measurement, usually tracked over a period of time.

PART 3: THE PROBLEM WITH "METRICS"

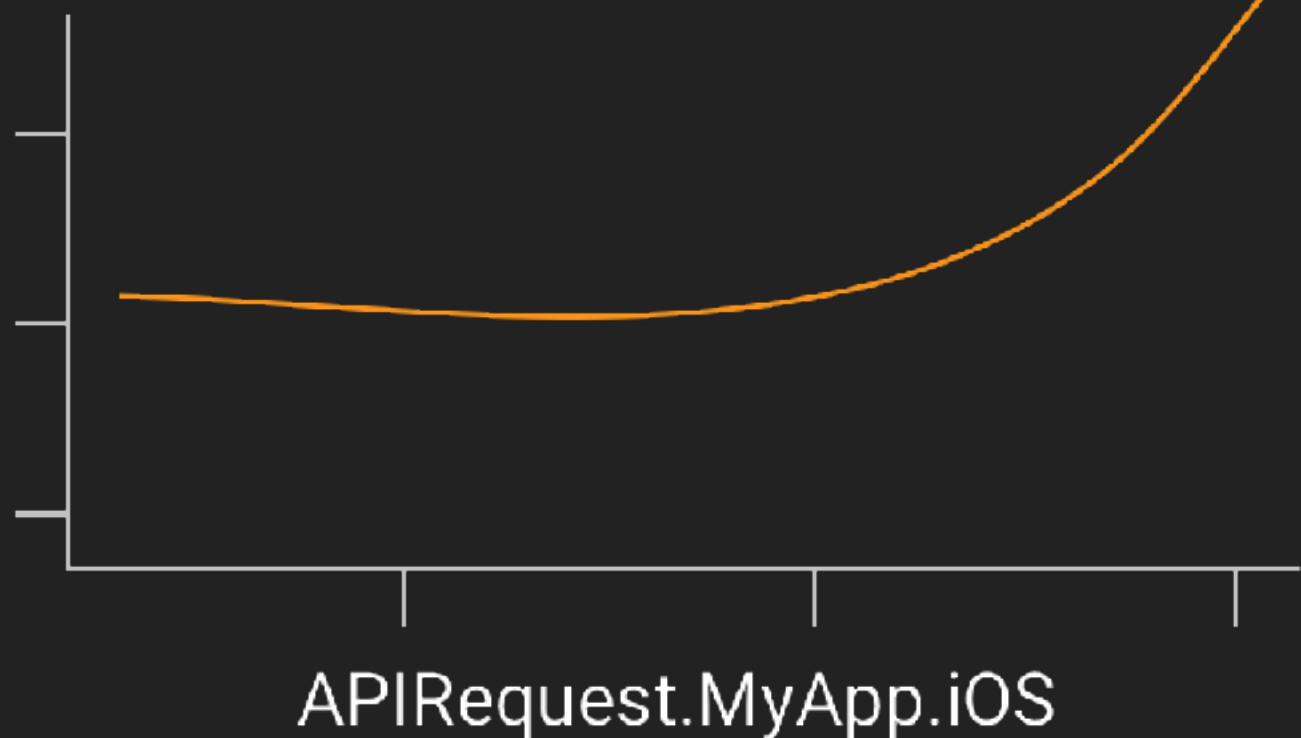
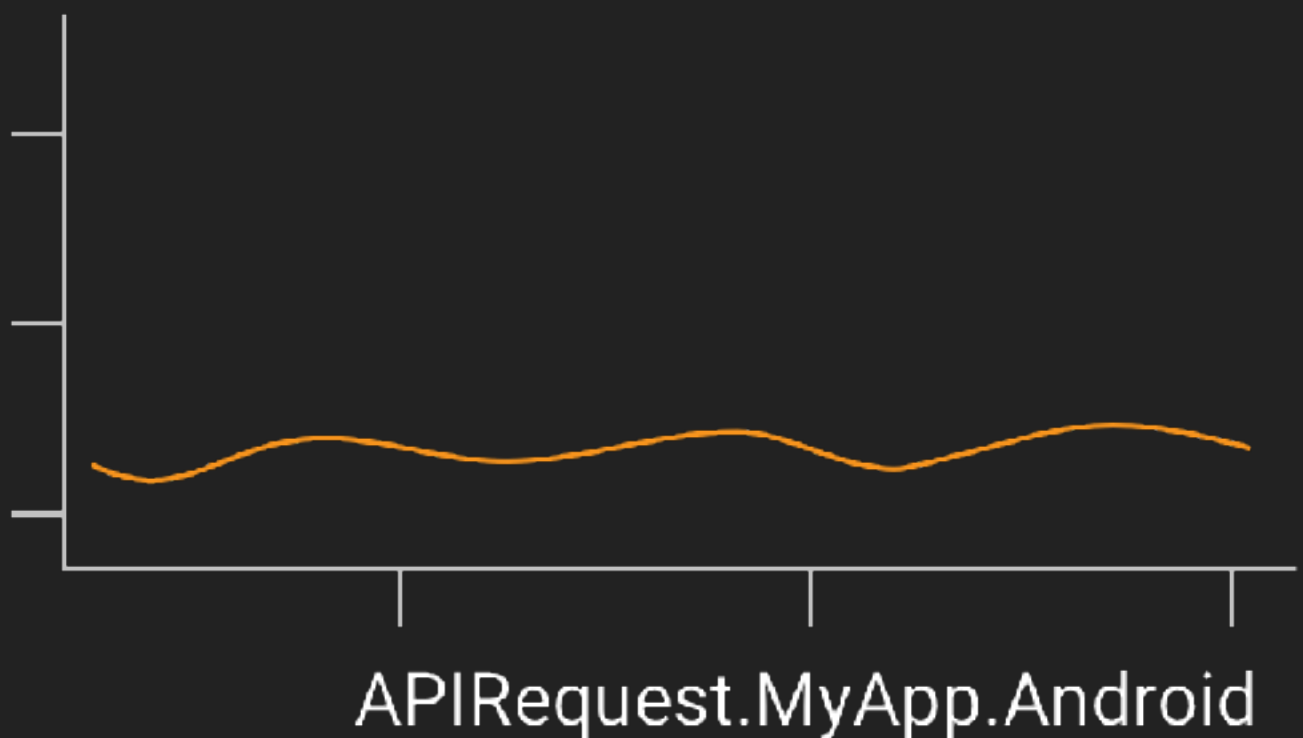
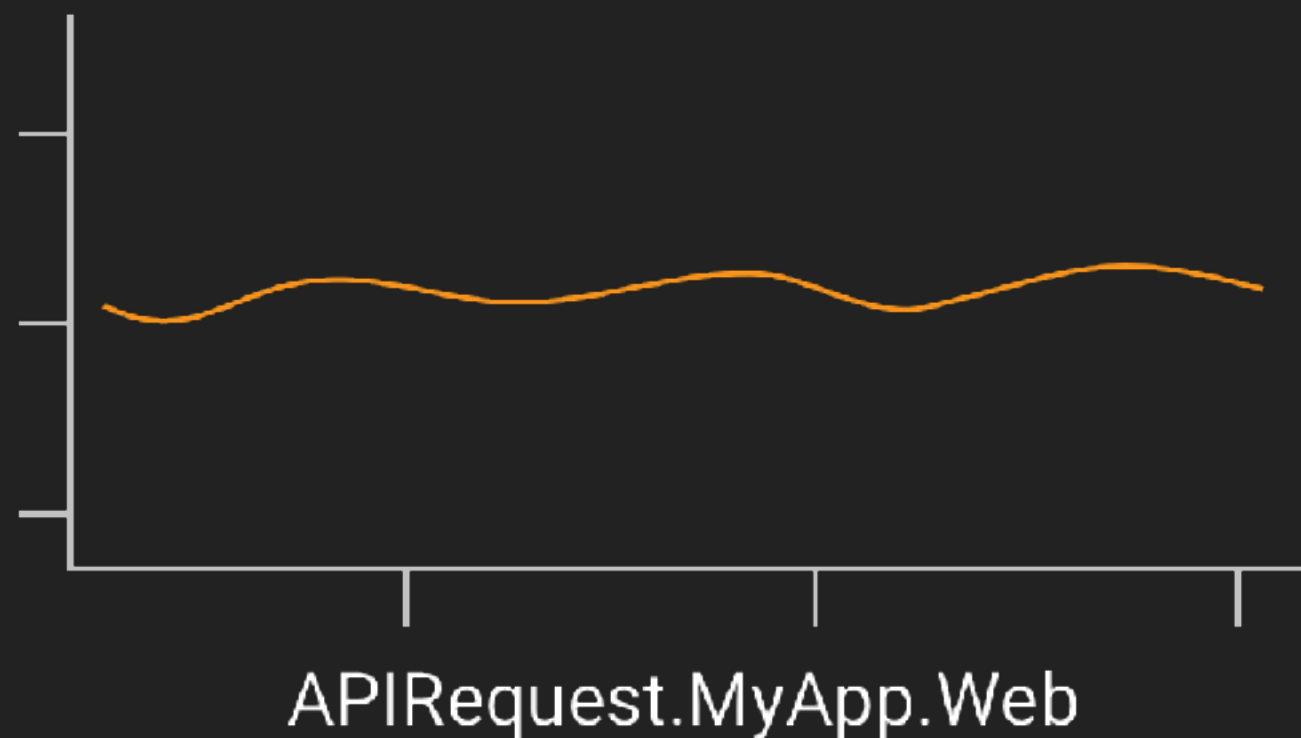
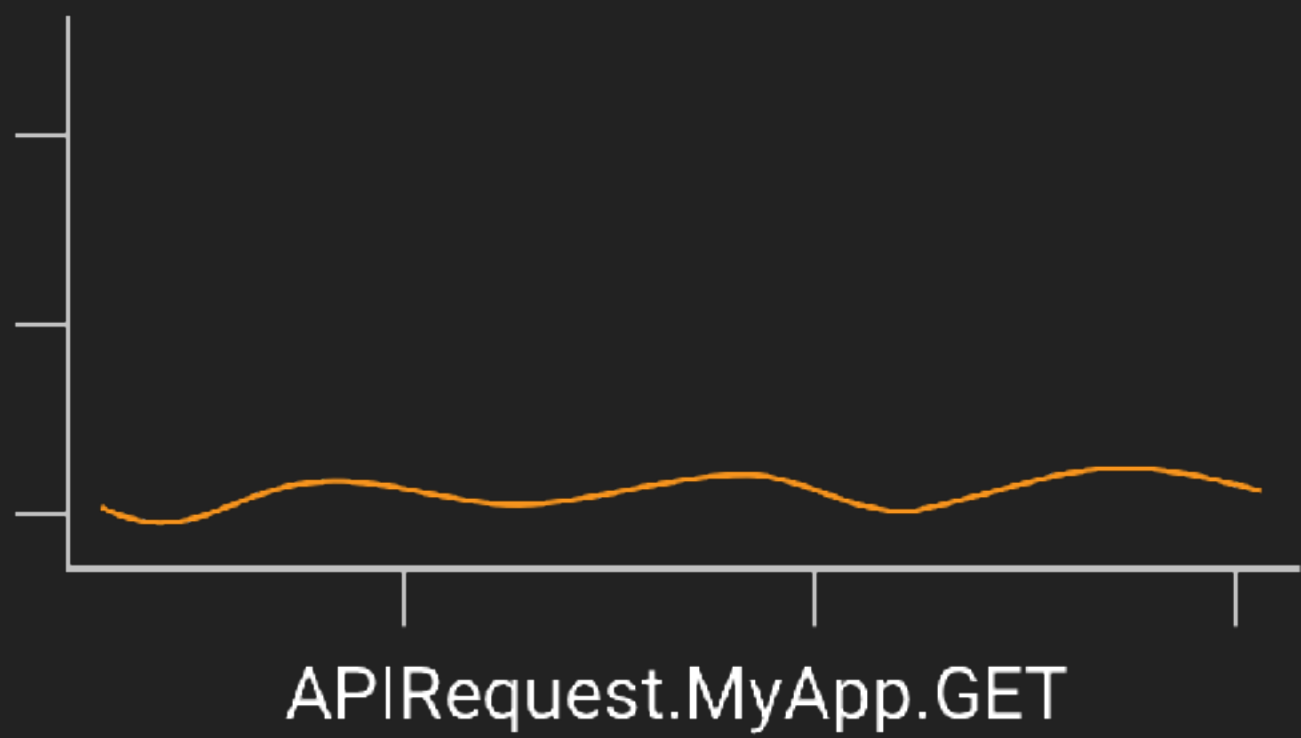
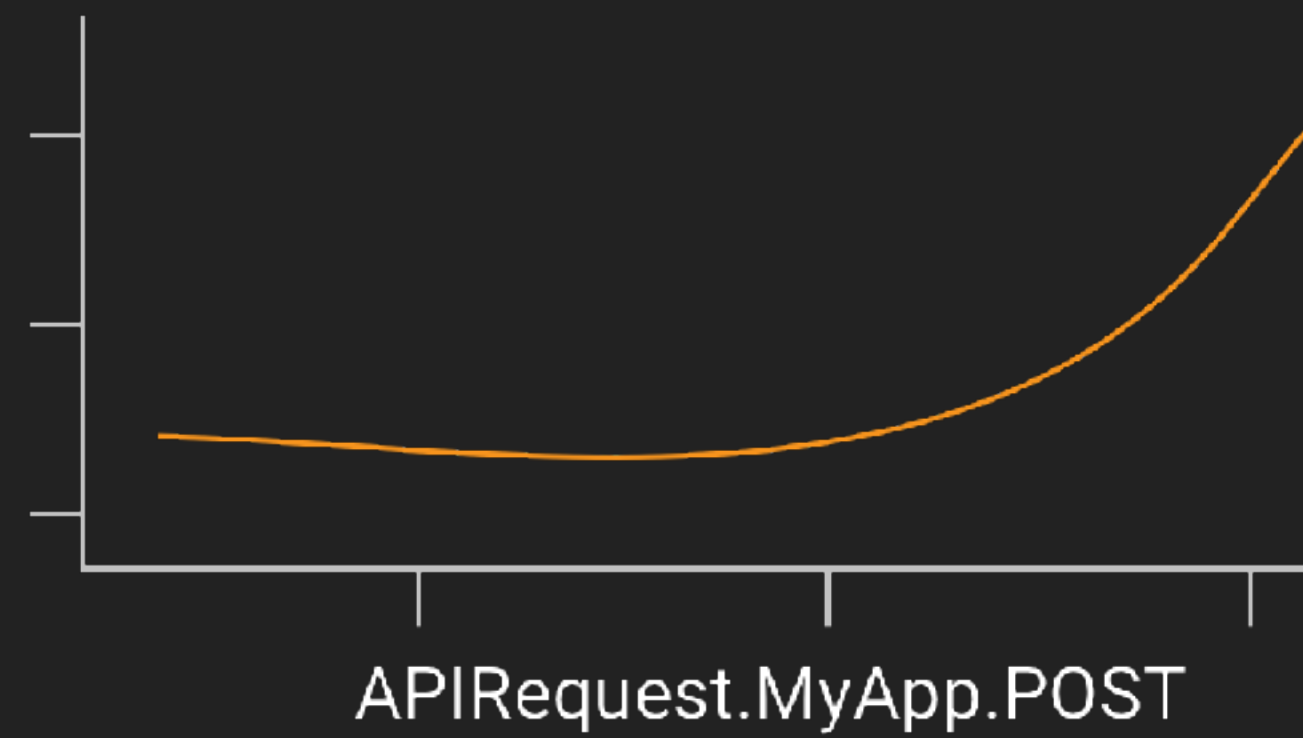
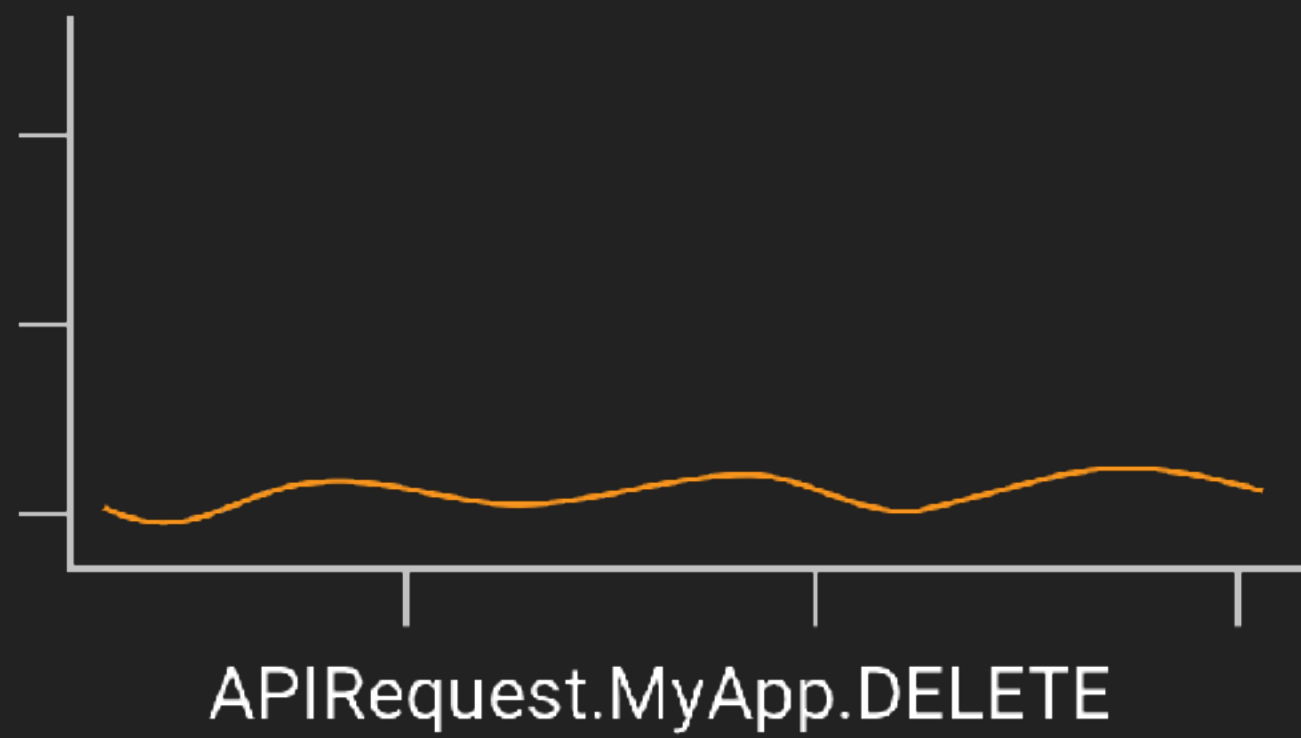


PART 3: THE PROBLEM WITH "METRICS"

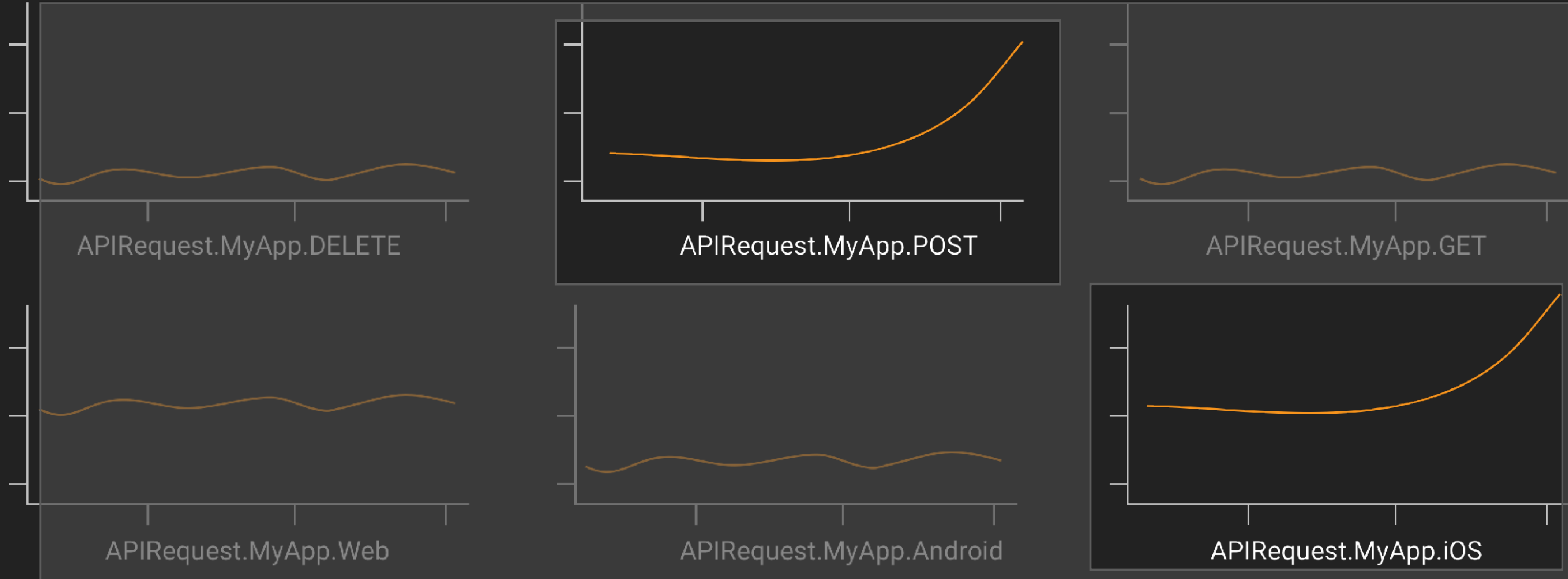


"Hey, so, uh, what caused this dip in my app's POST requests?"

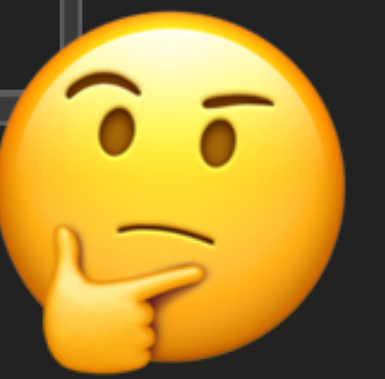
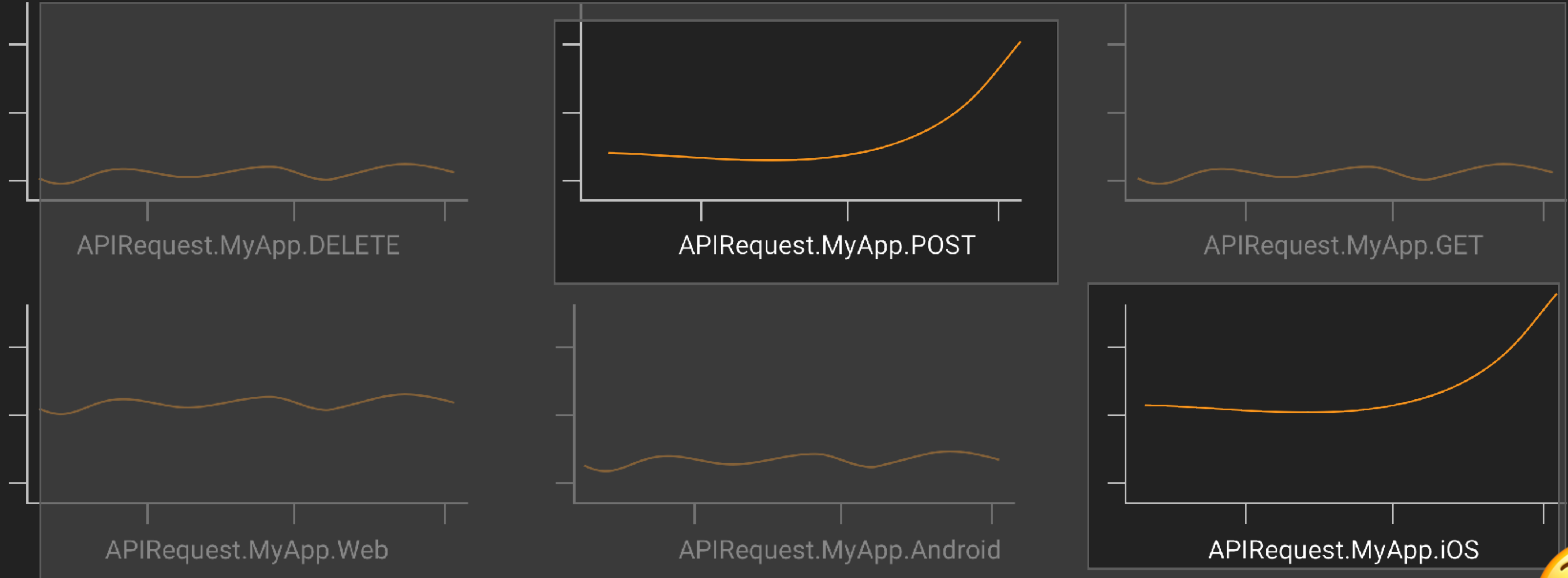
PART 3: THE PROBLEM WITH "METRICS"



PART 3: THE PROBLEM WITH "METRICS"



PART 3: THE PROBLEM WITH "METRICS"



"KNOW THY DATA"

... EXCEPT YOU'LL WANT TO ADD NEW ATTRIBUTES WHEN YOU REALIZE NEW THINGS ARE IMPORTANT AND YOU'LL WANT TO LEAVE ROOM FOR CHANGES BETWEEN APP VERSIONS AND POSSIBLY ALSO WHEN OUR SDK INTERACTS POORLY WITH YOUR APP AND SOMETIMES INDIVIDUAL DEVICES JUST DO BAD THINGS AND...

The Problem with **PRE-AGGREGATED**

(Part 1)

→  **METRICS**

The Problem with **PRE-AGGREGATED**

(Part 2)

→  **METRICS**

Parse + facebook

Scuba: Diving into Data at Facebook

Lior Abraham^{*}
Vinayak Borkar
Daniel Merl
Subbu Subramanian

John Allen
Bhuwan Chopra
Josh Metzler
Janet L. Wiener
Facebook, Inc. Menlo Park, CA

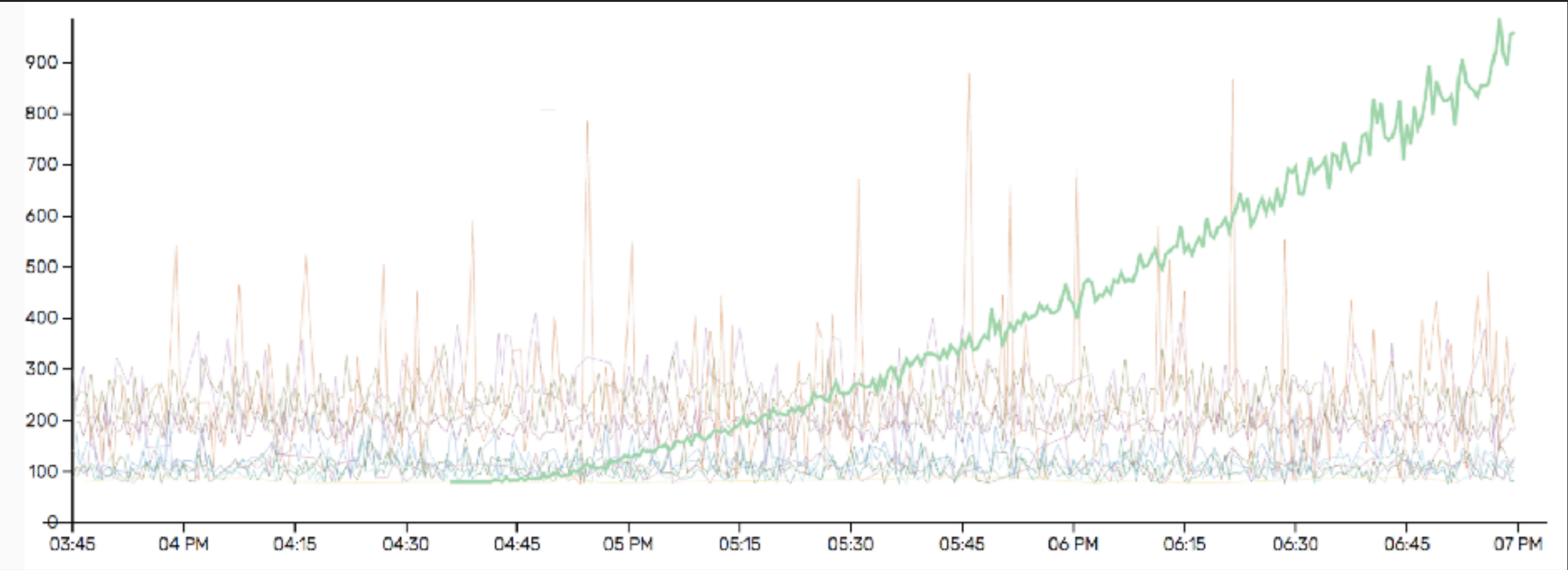
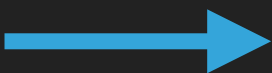
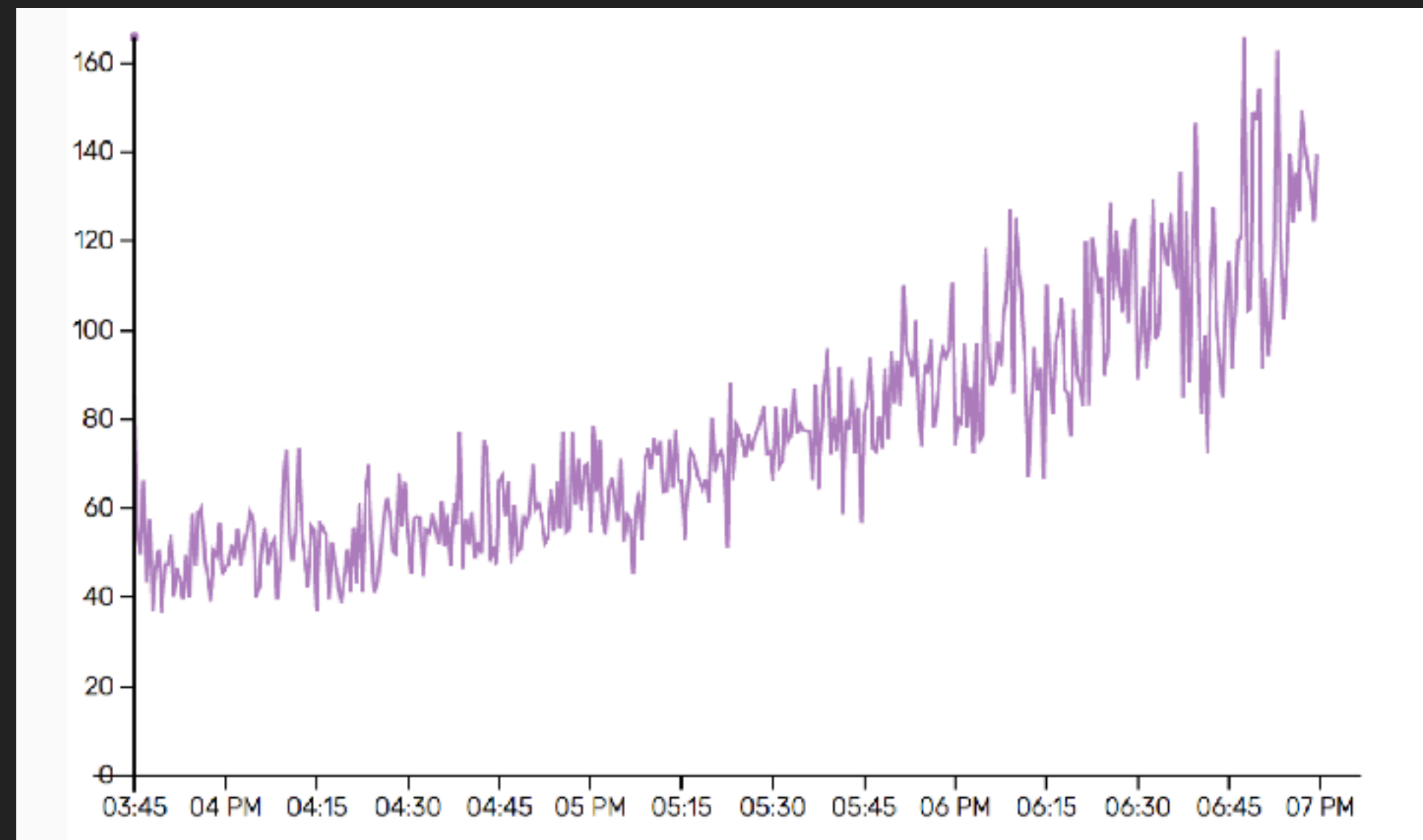
Oleksandr Barykin
Ciprian Gerea
David Reiss
Okay Zed

ABSTRACT

Facebook takes performance monitoring seriously. Performance issues can impact over one billion users so we track thousands of

Originally, we relied on pre-aggregated graphs and a carefully managed, hand-coded, set of scripts over a MySQL database of performance data. By 2011, that solution became too rigid and slow.

PART 3: THE PROBLEM WITH "METRICS"

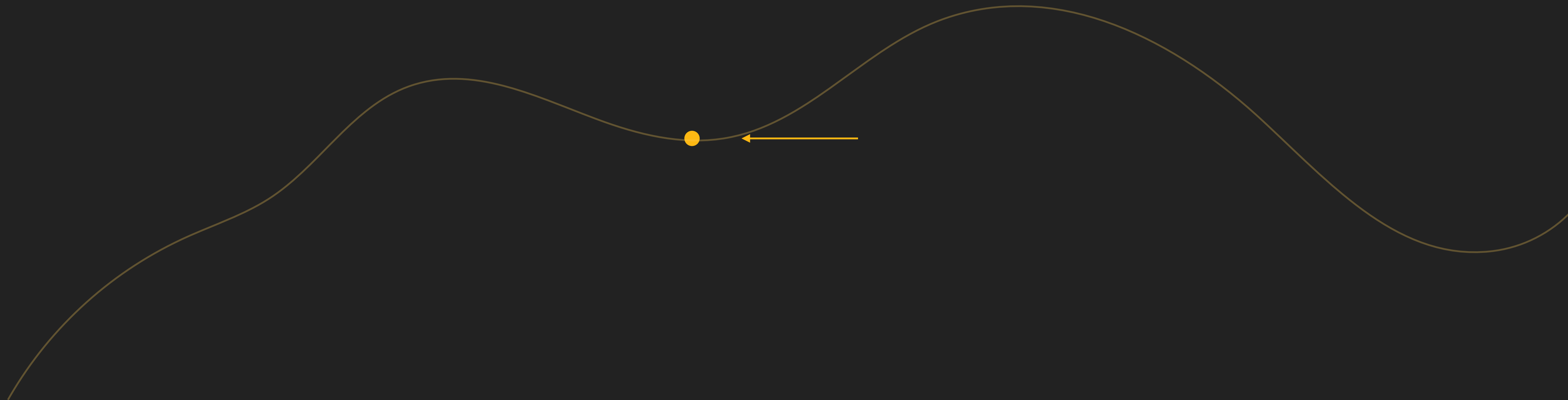


PART 3: THE PROBLEM WITH "METRICS"





The overall error rate for my API is 0.012!



PART 3: THE PROBLEM WITH "METRICS"



PART 3: THE PROBLEM WITH "METRICS"



PART 3: THE PROBLEM WITH "METRICS"

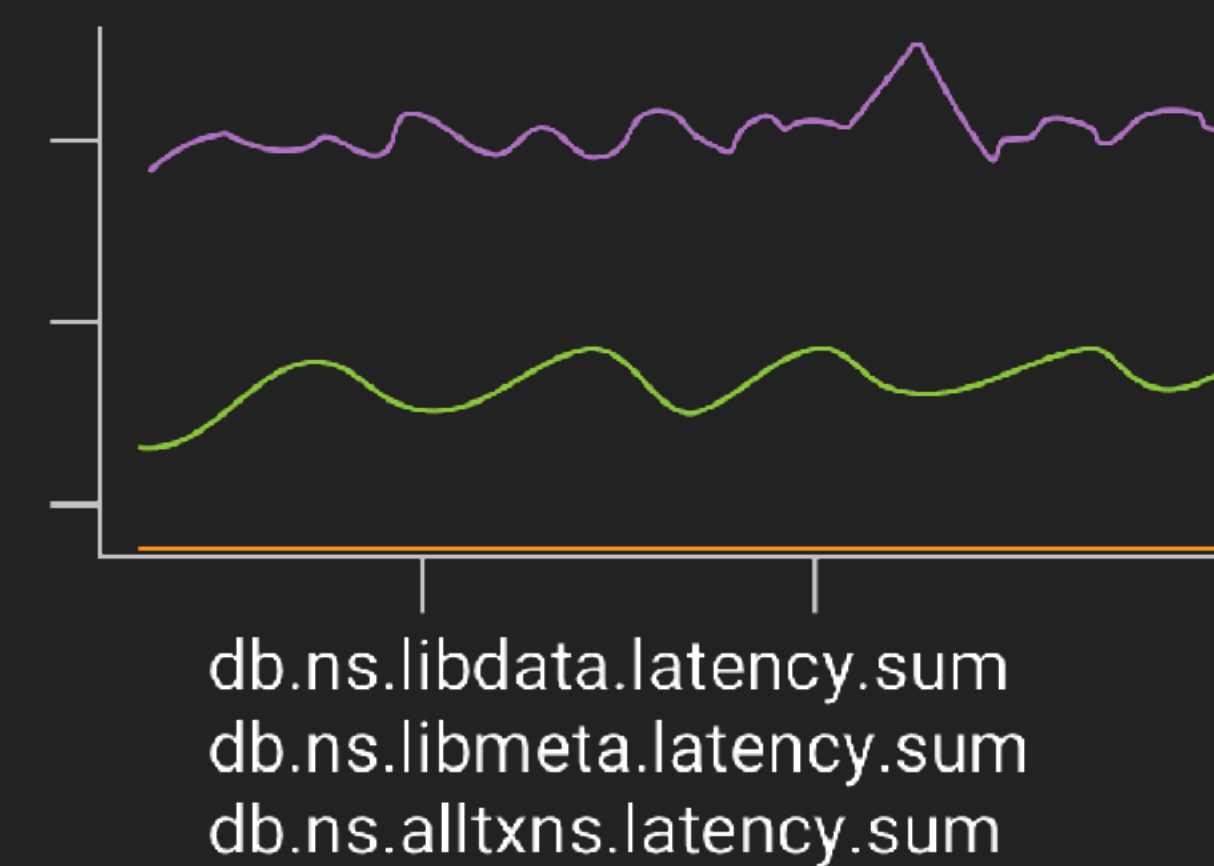
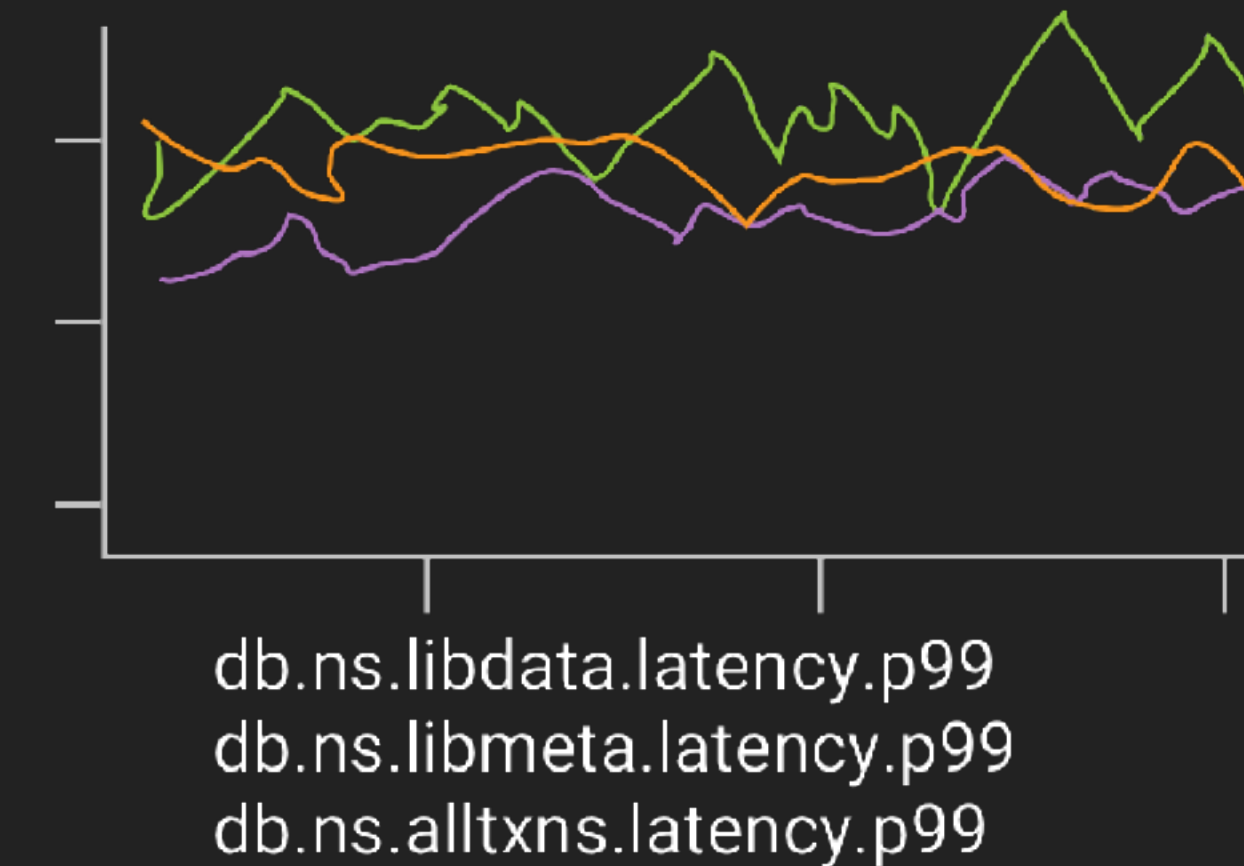
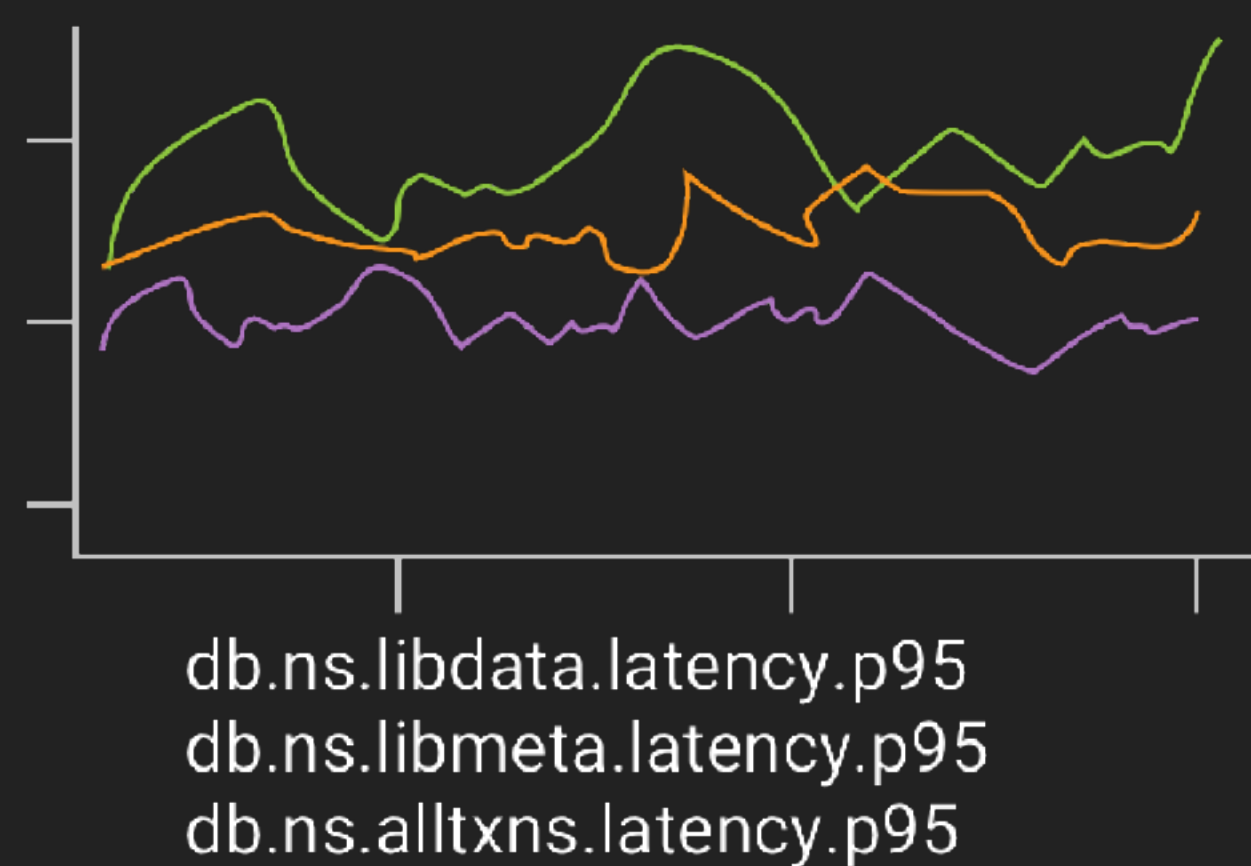
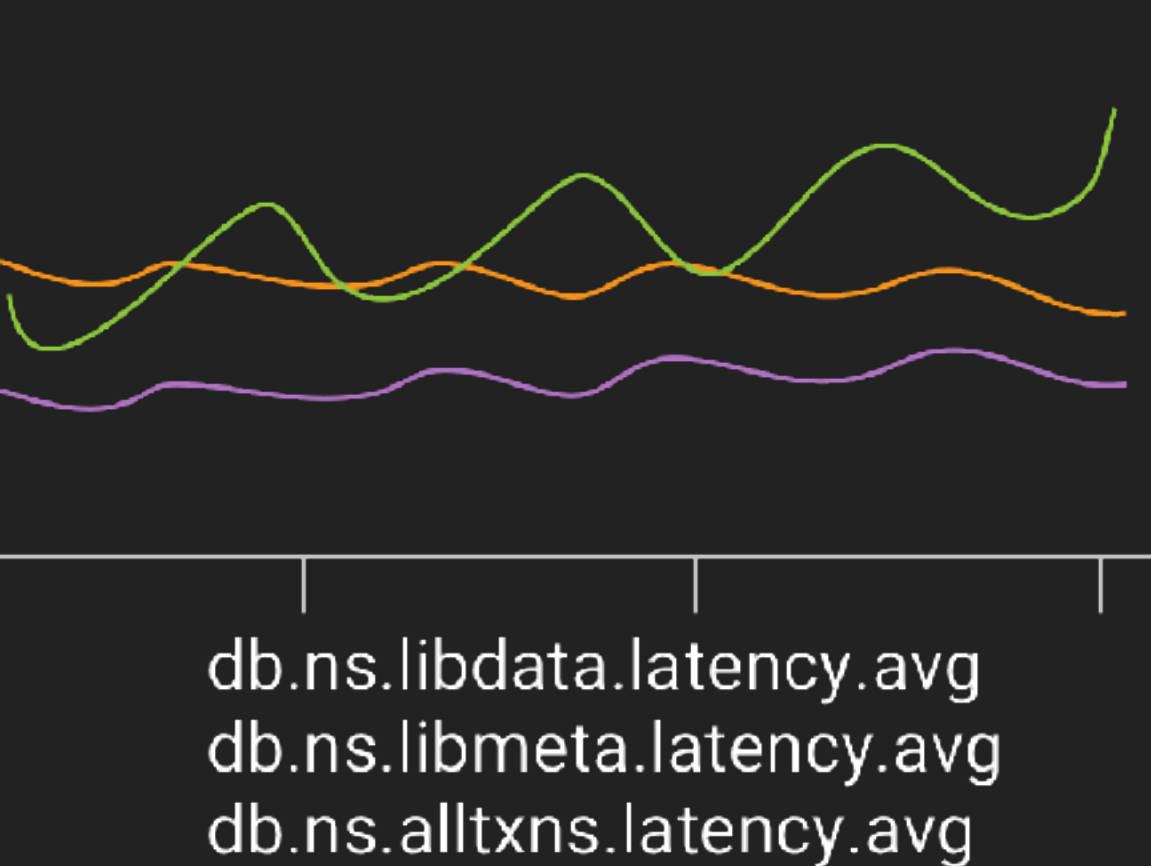
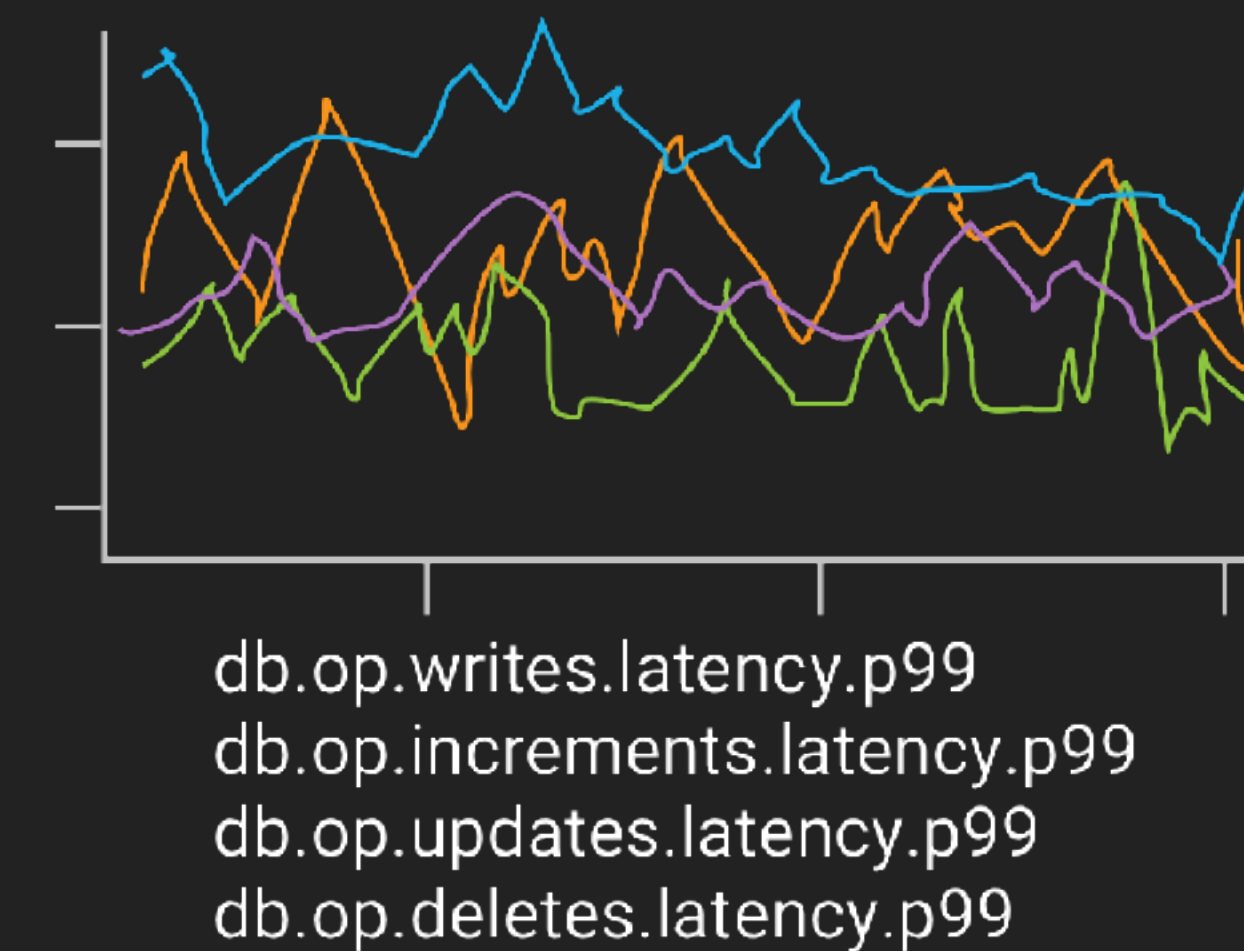
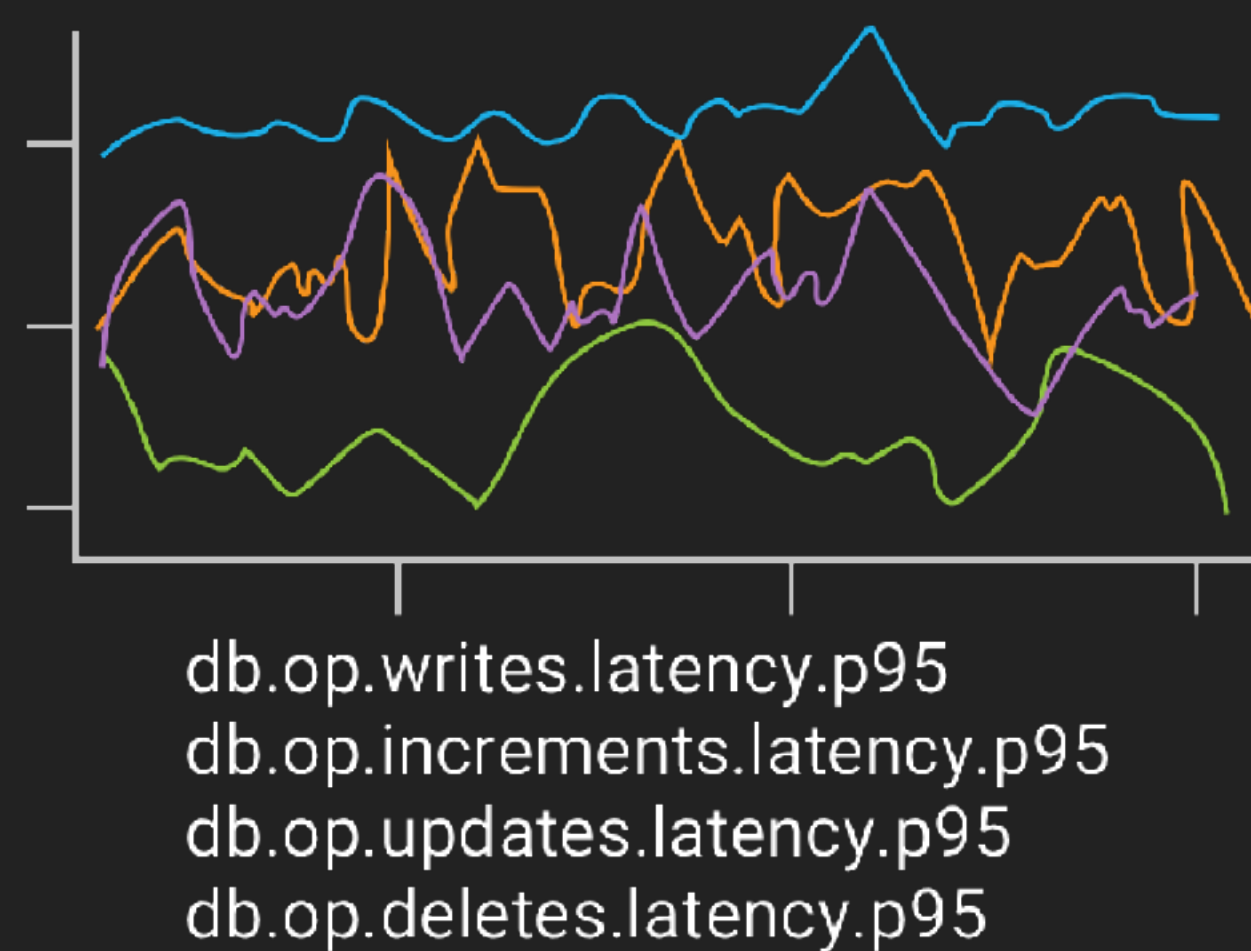
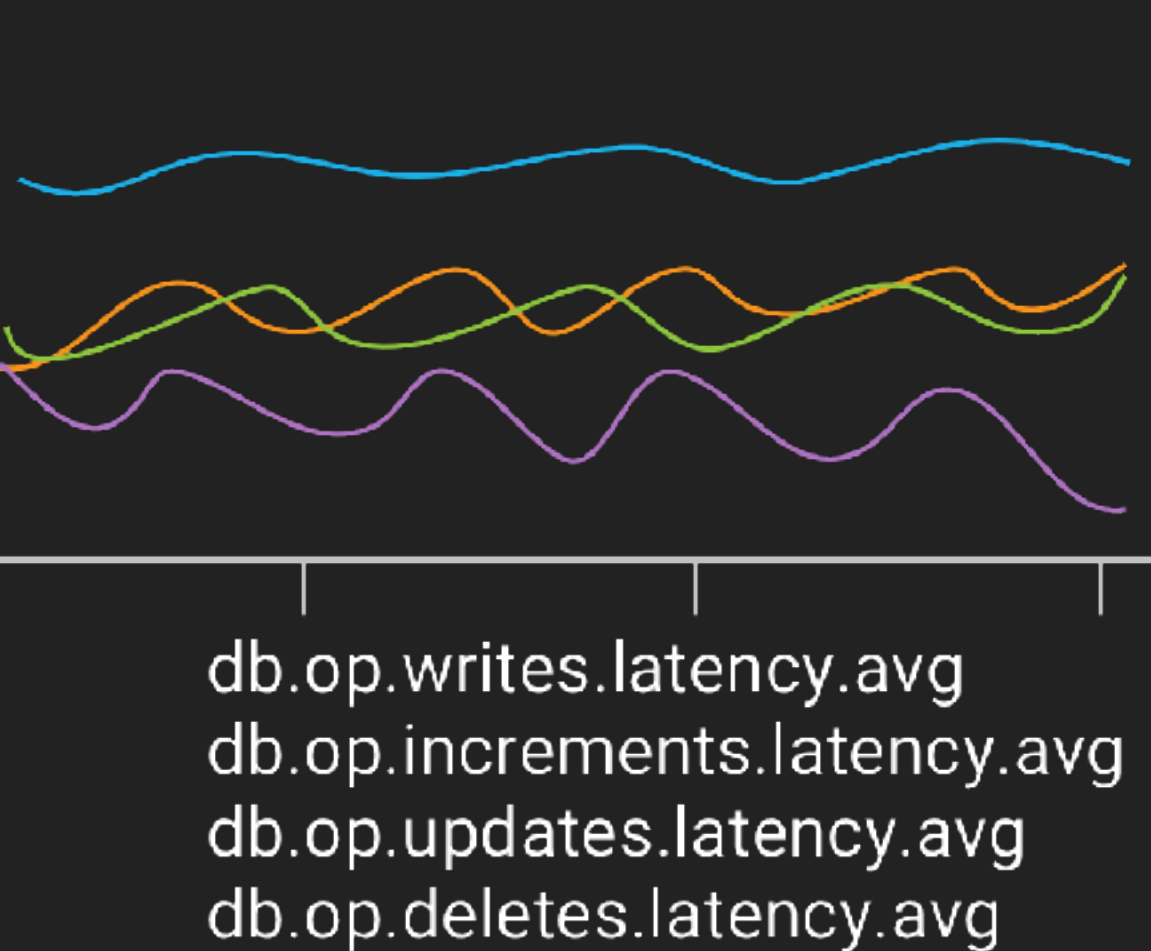
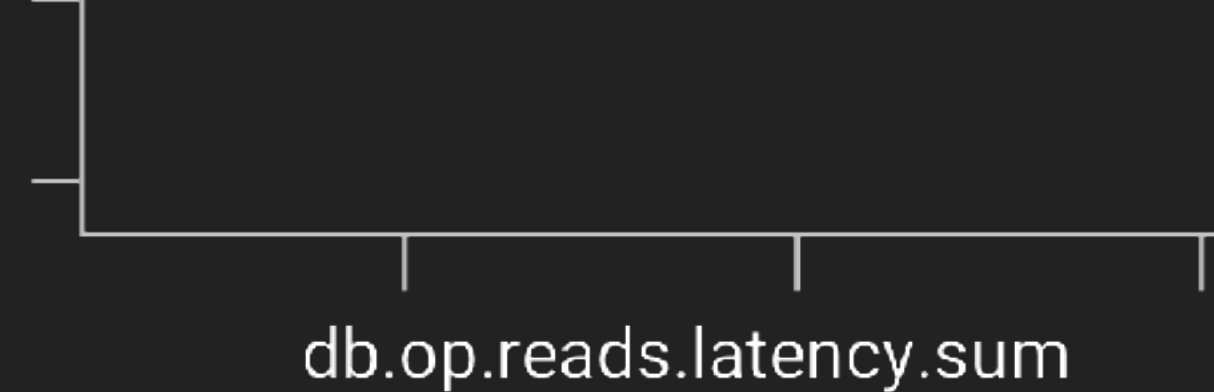
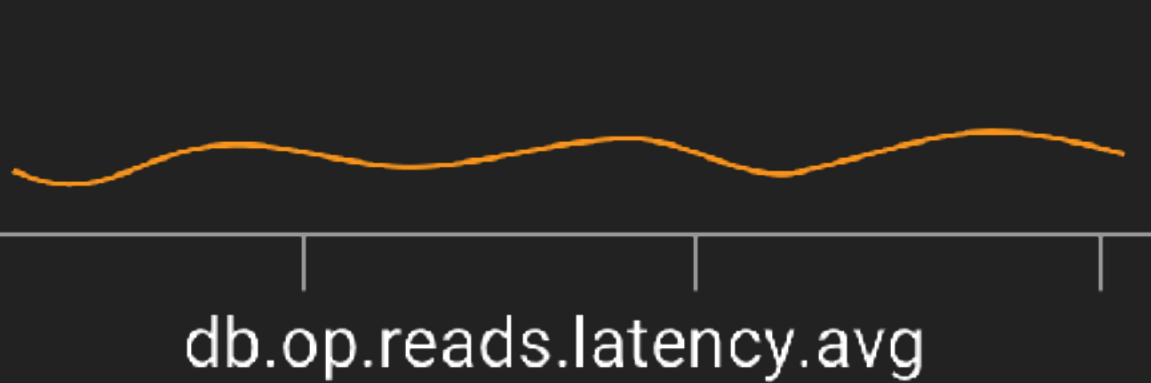


THE FUTURE?

OBSERVABILITY



(THE OBLIGATORY DISCLAIMER)



OBSERVABILITY AND NEXT-GEN TOOLS

OBSERVABILITY AND NEXT-GEN TOOLS

- Event-based systems

OBSERVABILITY AND NEXT-GEN TOOLS

- Event-based systems
- Approximation algorithms

OBSERVABILITY AND NEXT-GEN TOOLS

- Event-based systems
- Approximation algorithms
- Intelligent sampling techniques

OBSERVABILITY AND NEXT-GEN TOOLS

- Event-based systems
- Approximation algorithms
- Intelligent sampling techniques

OBSERVABILITY AND NEXT-GEN TOOLS

- Event-based systems
- Approximation algorithms
- Intelligent sampling techniques

How Events Fix The Problem with **PRE-AGGREGATED METRICS**

How Events Fix The Problem with **PRE-AGGREGATED METRICS**

How Events Fix The Problem with **PRE-AGGREGATED METRICS**

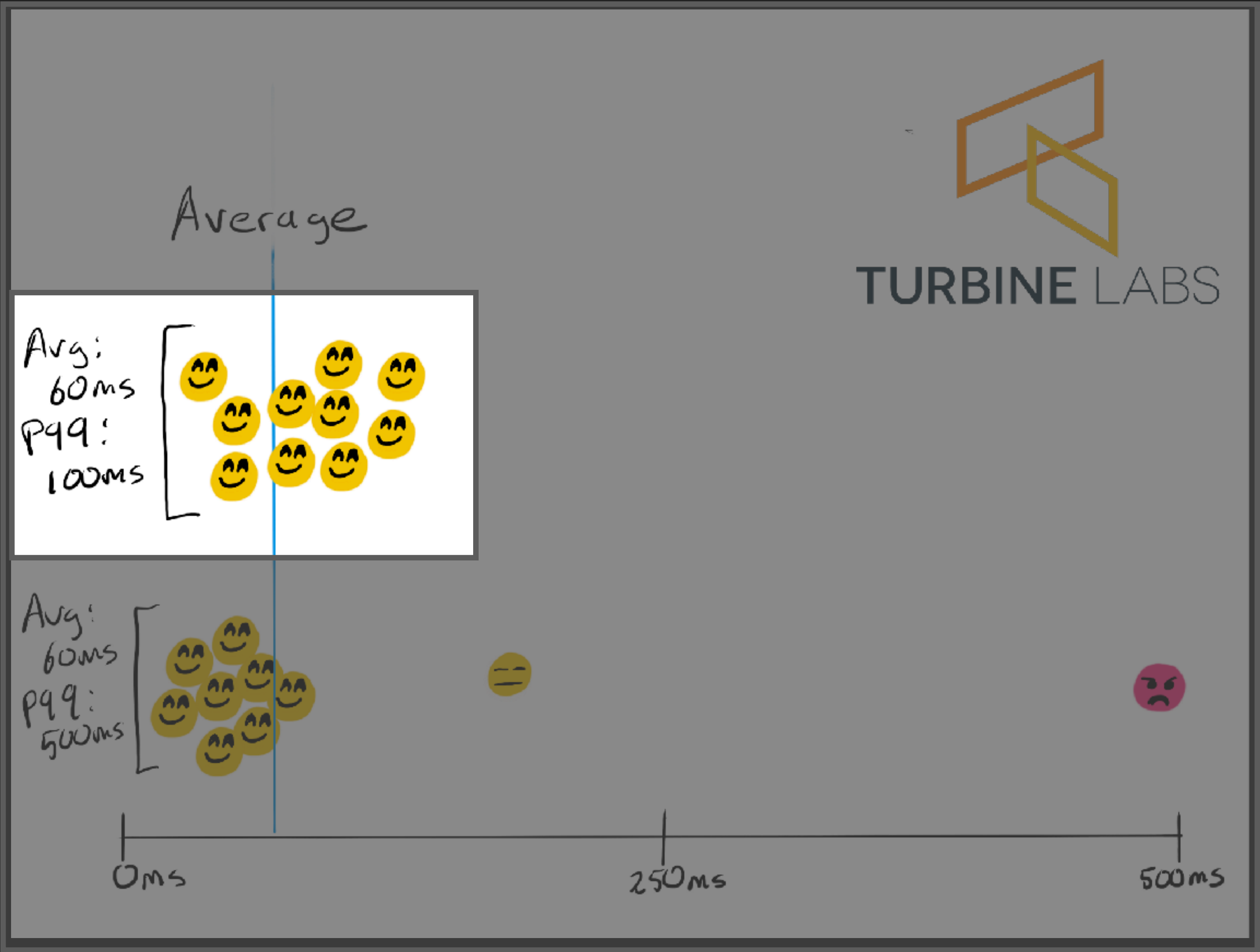
How Events Fix The Problem with **PRE-AGGREGATED METRICS**

How Events Fix The Problem with **PRE-AGGREGATED METRICS** (sort of)

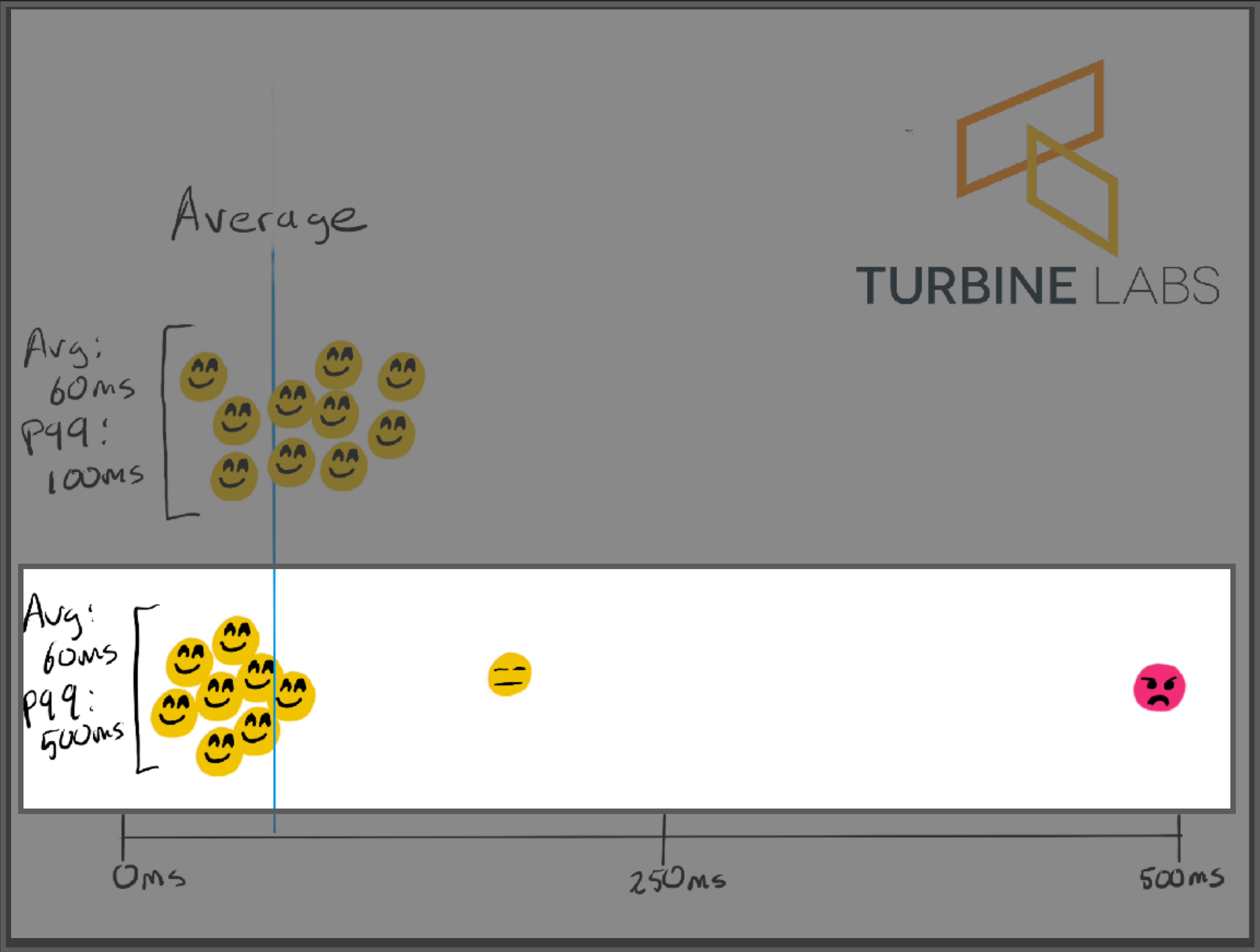
How Events Fix The Problem with **PRE-AGGREGATED METRICS**

How Events Fix The Problem with **PRE-AGGREGATED METRICS**

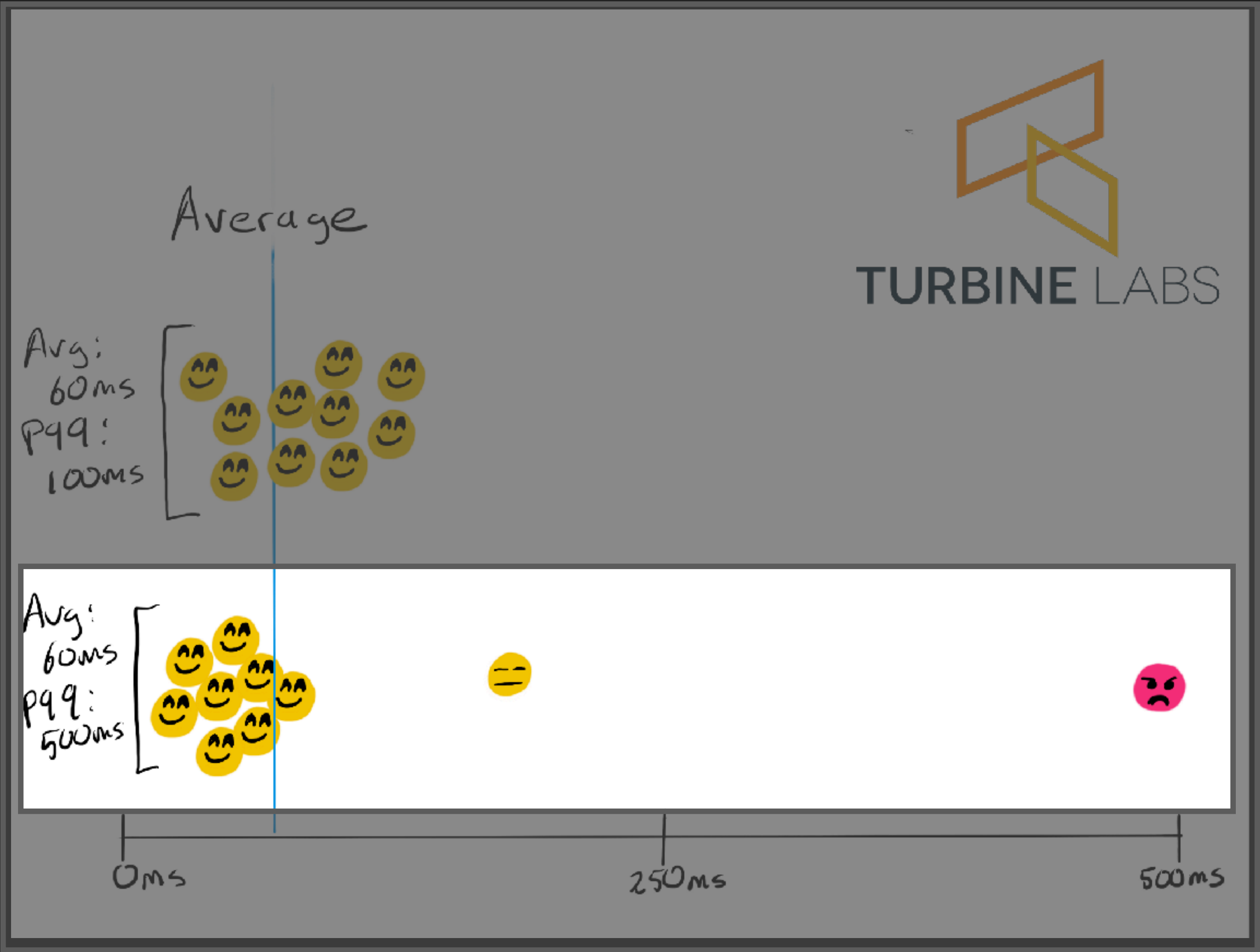
THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



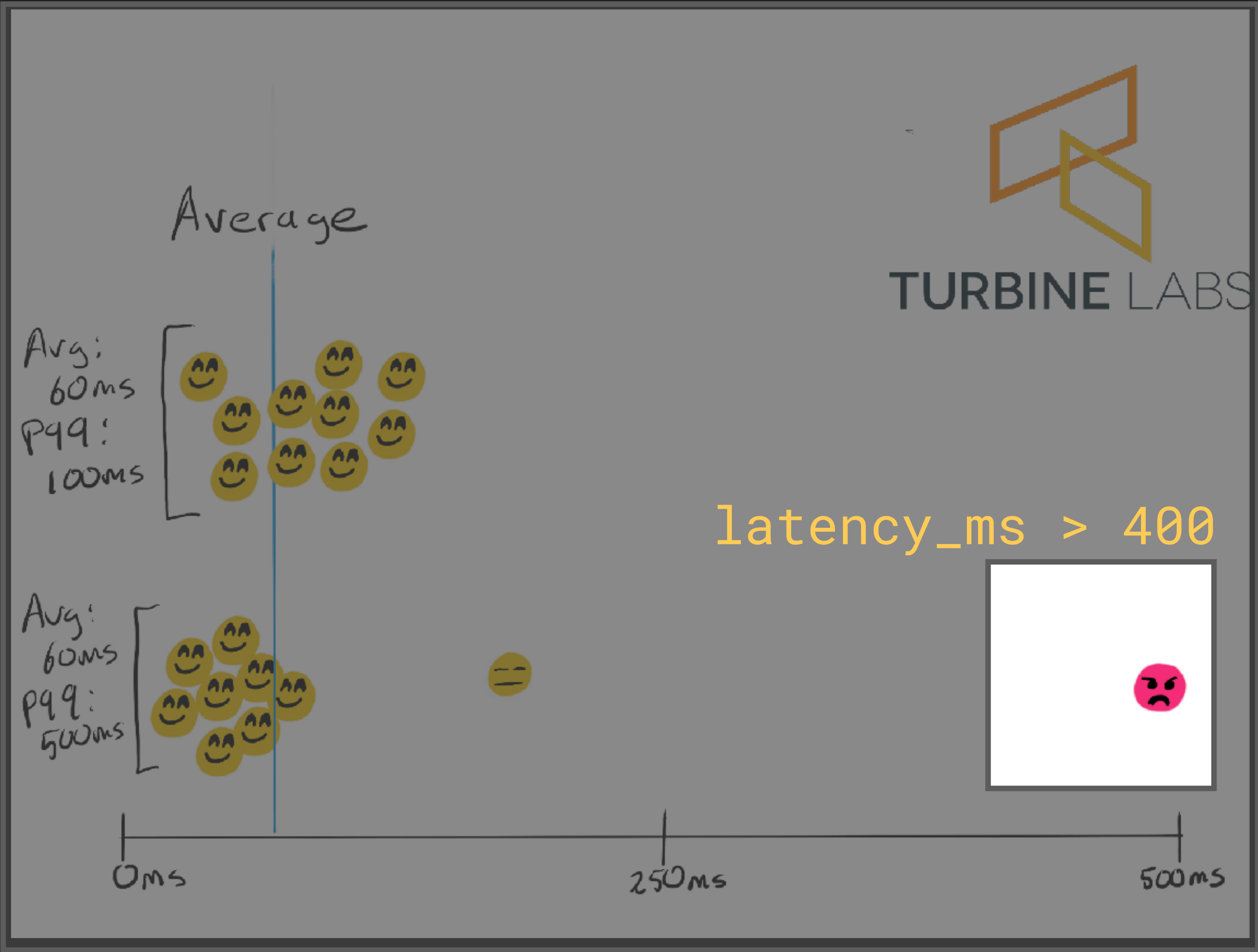
THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



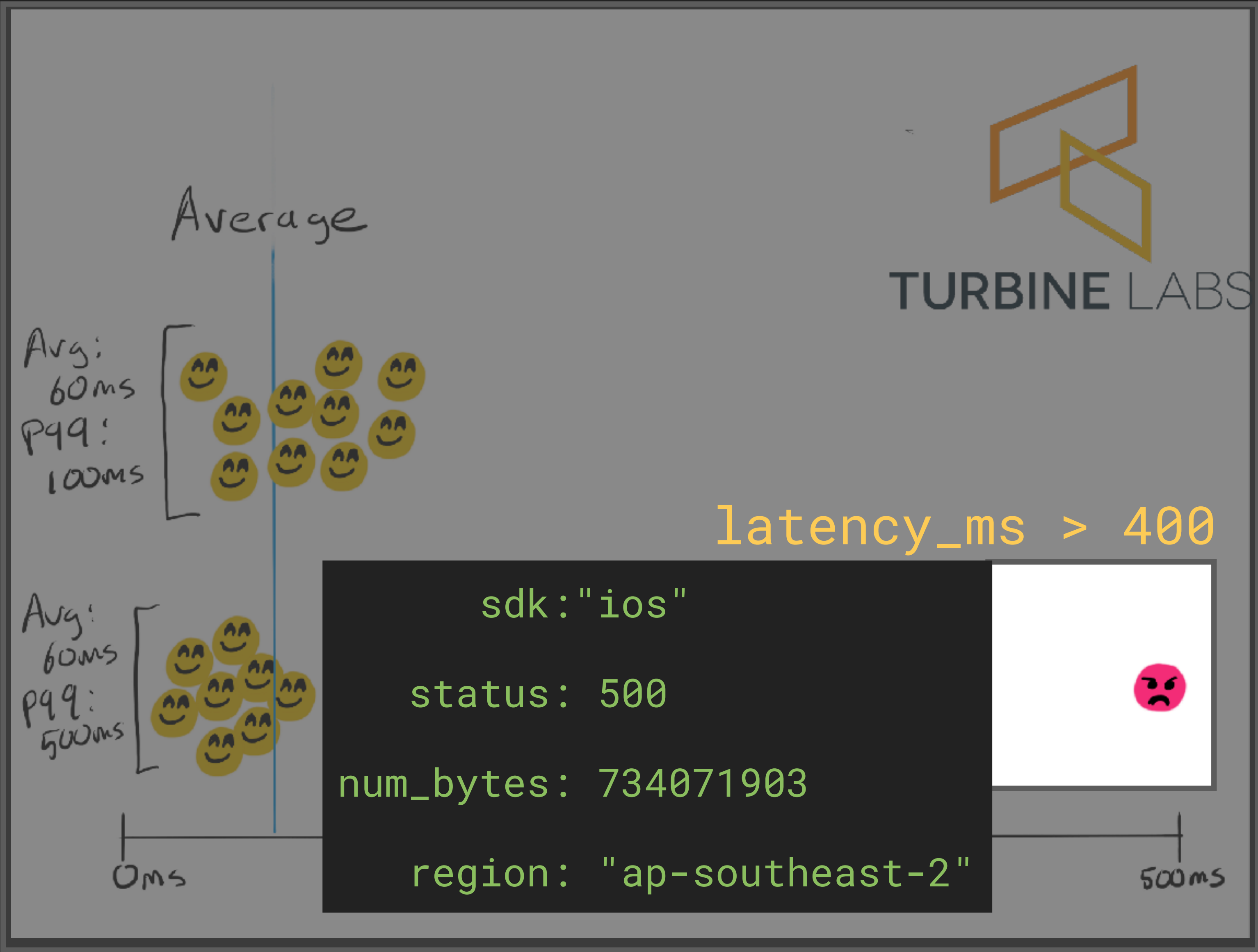
THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



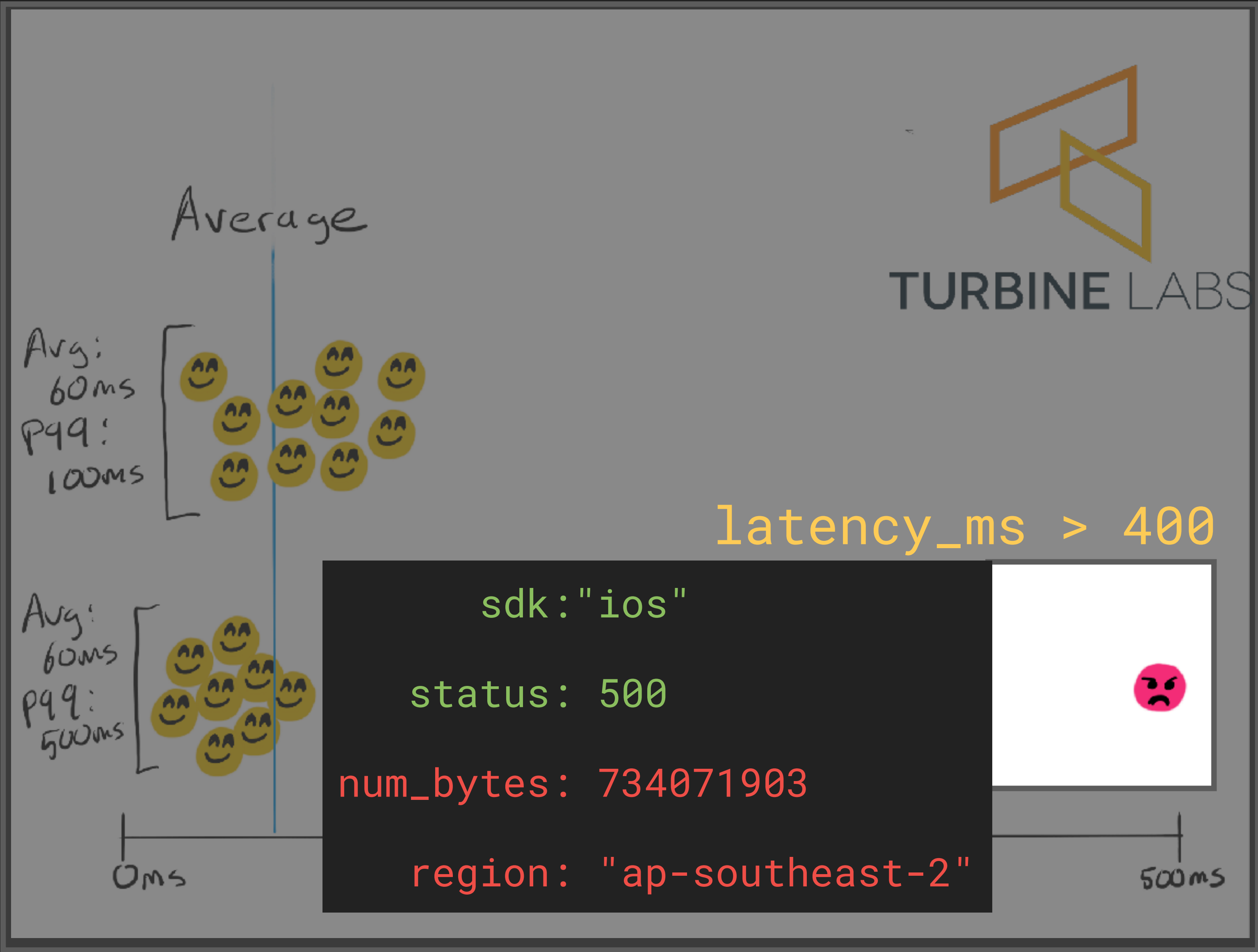
THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



THE FUTURE: HOW EVENTS FIX THE PROBLEM WITH "METRICS"



The Problem with **PRE-AGGREGATED METRICS** AND WHY EVENT-BASED SYSTEMS ARE THE FUTURE

The Problem with **PRE-AGGREGATED METRICS** AND WHY EVENT-BASED SYSTEMS ARE THE FUTURE

The Problem with **PRE-AGGREGATED METRICS** AND WHY EVENT-BASED SYSTEMS ARE THE FUTURE

The Problem with **PRE-AGGREGATED METRICS** AND WHY EVENT-BASED SYSTEMS ARE THE FUTURE

THANKS!
