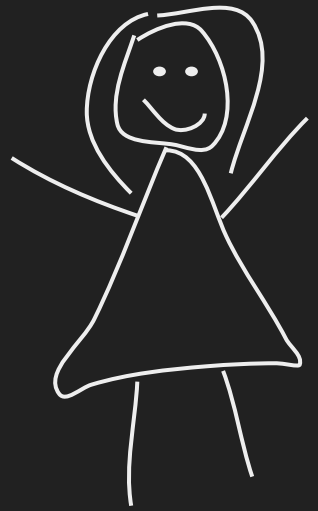
UX Design and Education for Effective Monitoring Tools

Hi!👋



- I'm Amy!
- Observability engineer!
- Also on the plenary panel tomorrow at 4pm!
- amynguyen.net
  @amyngyn

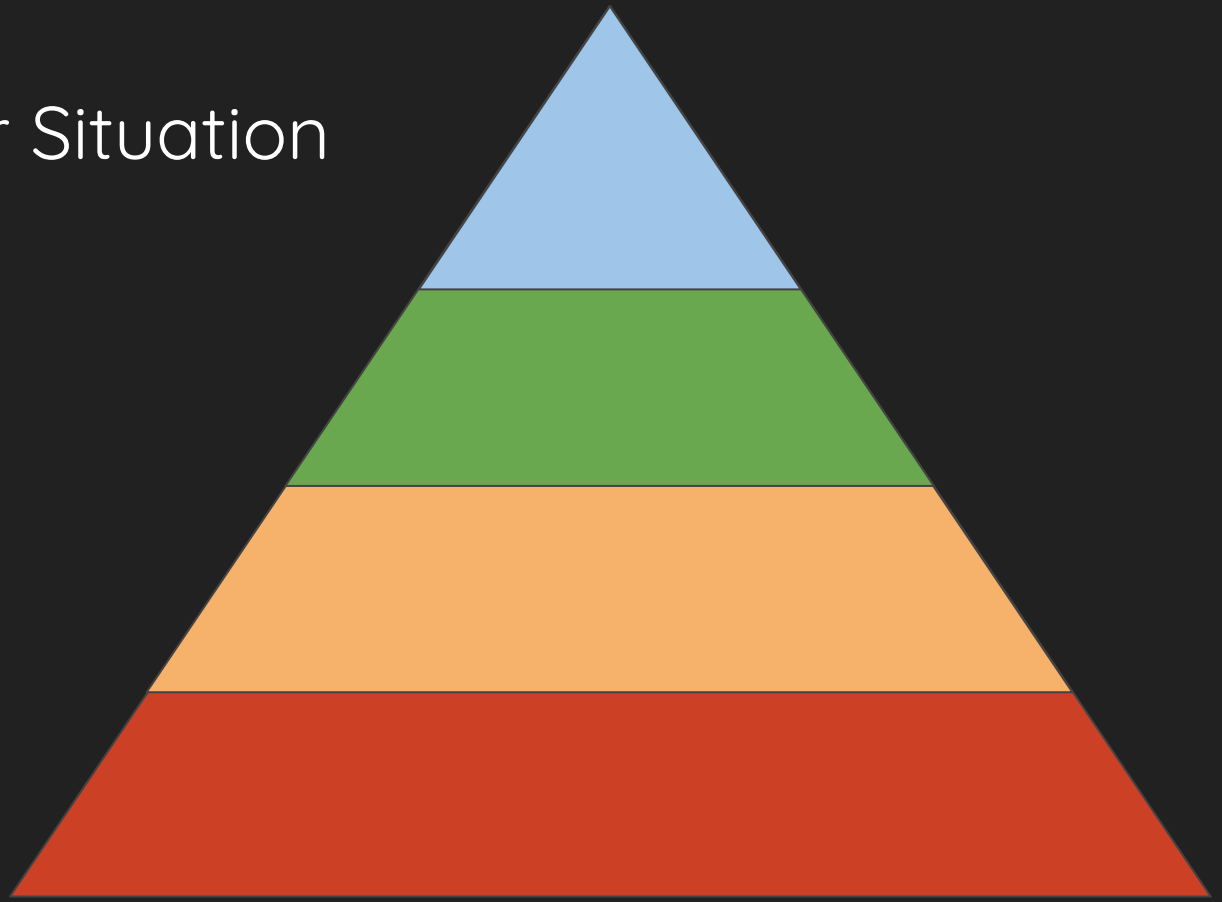
Agenda

- Motivation
- Sharing what you know
- Designing what your users want
- Recap

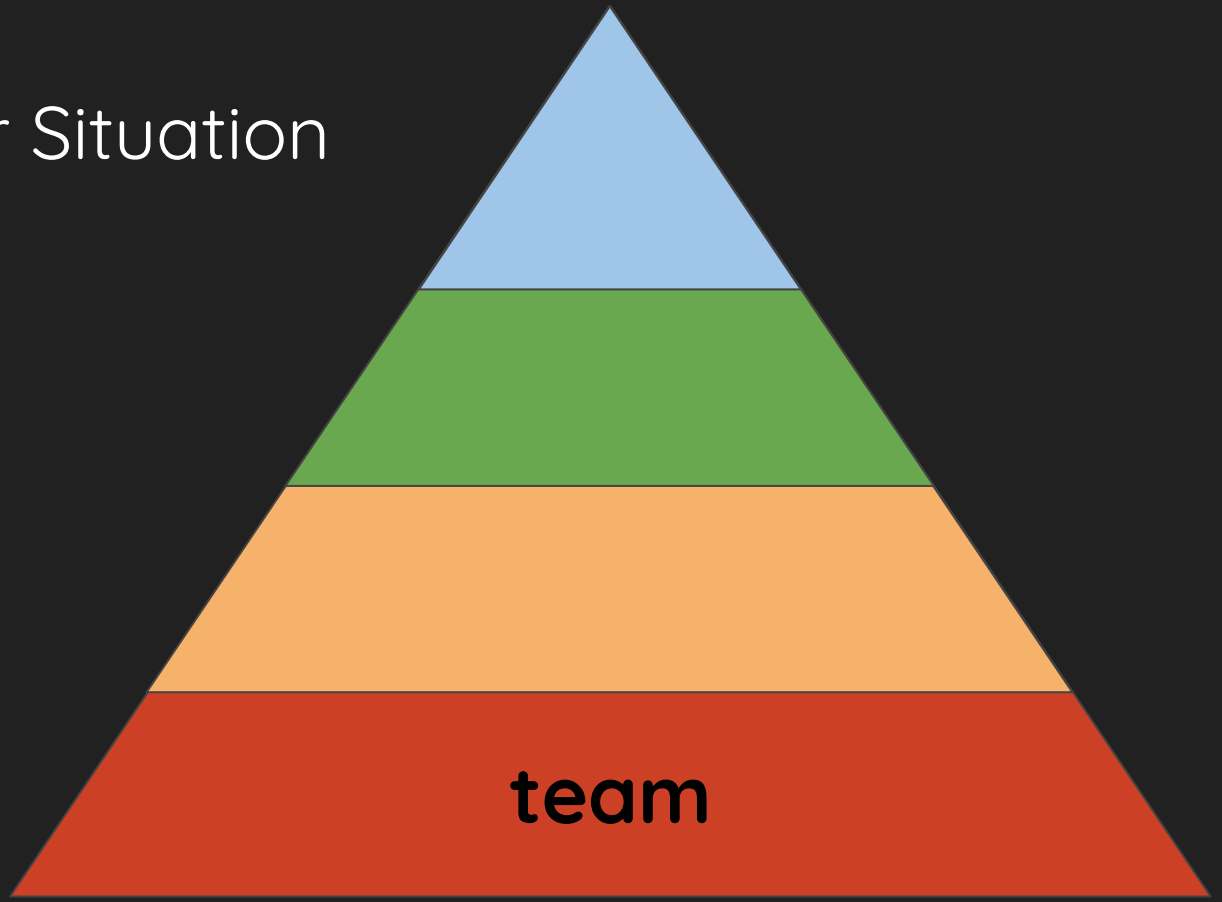
Agenda

- **Motivation**
- Sharing what you know
- Designing what your users want
- Recap

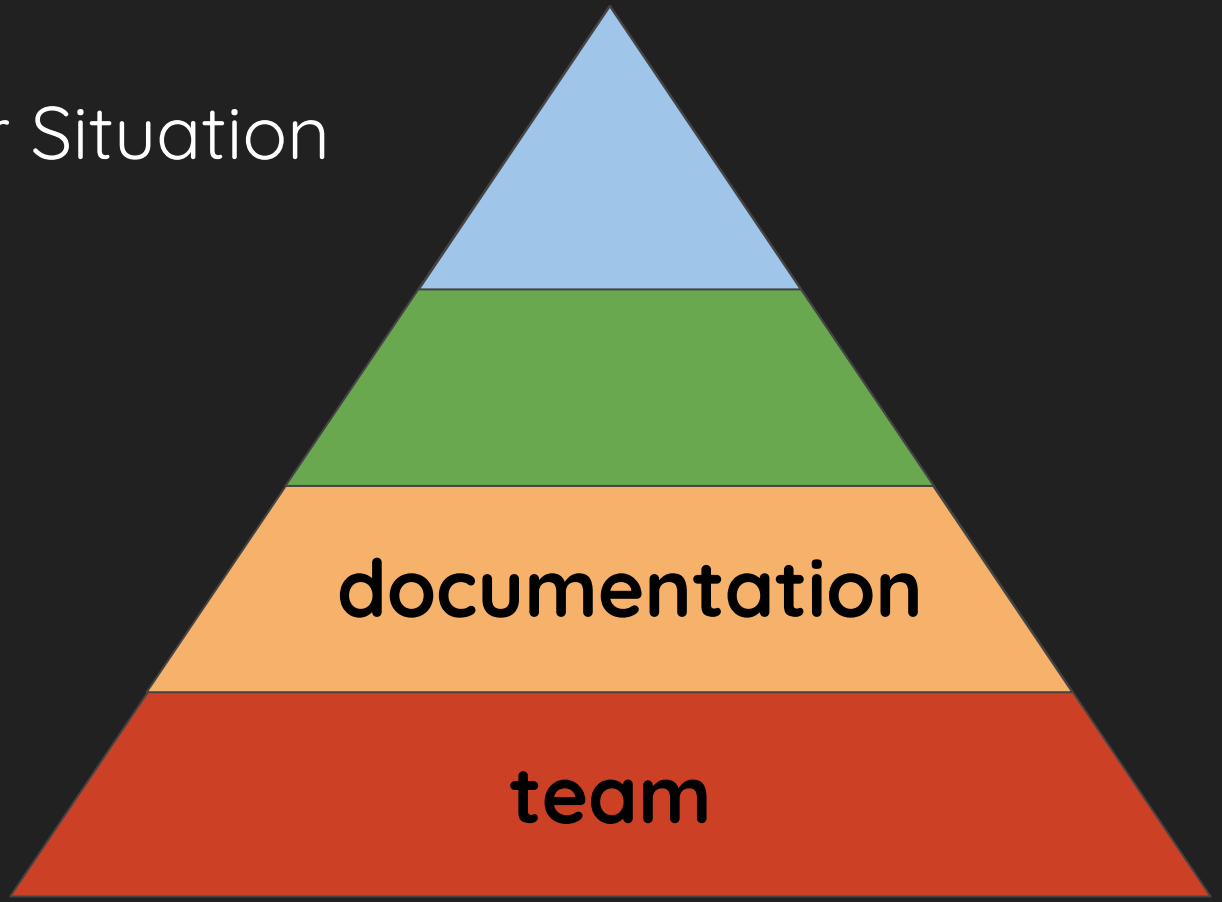
UX and Your Situation



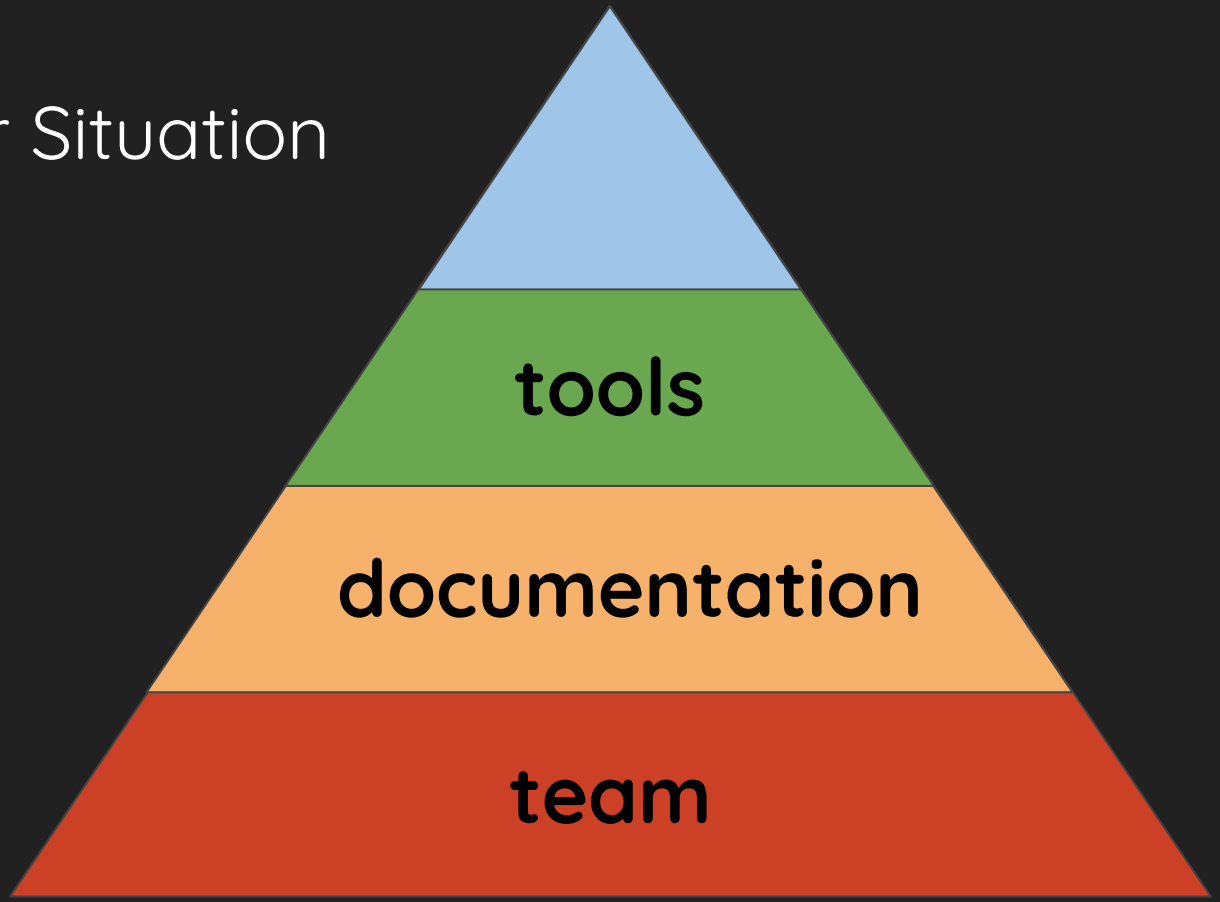
UX and Your Situation



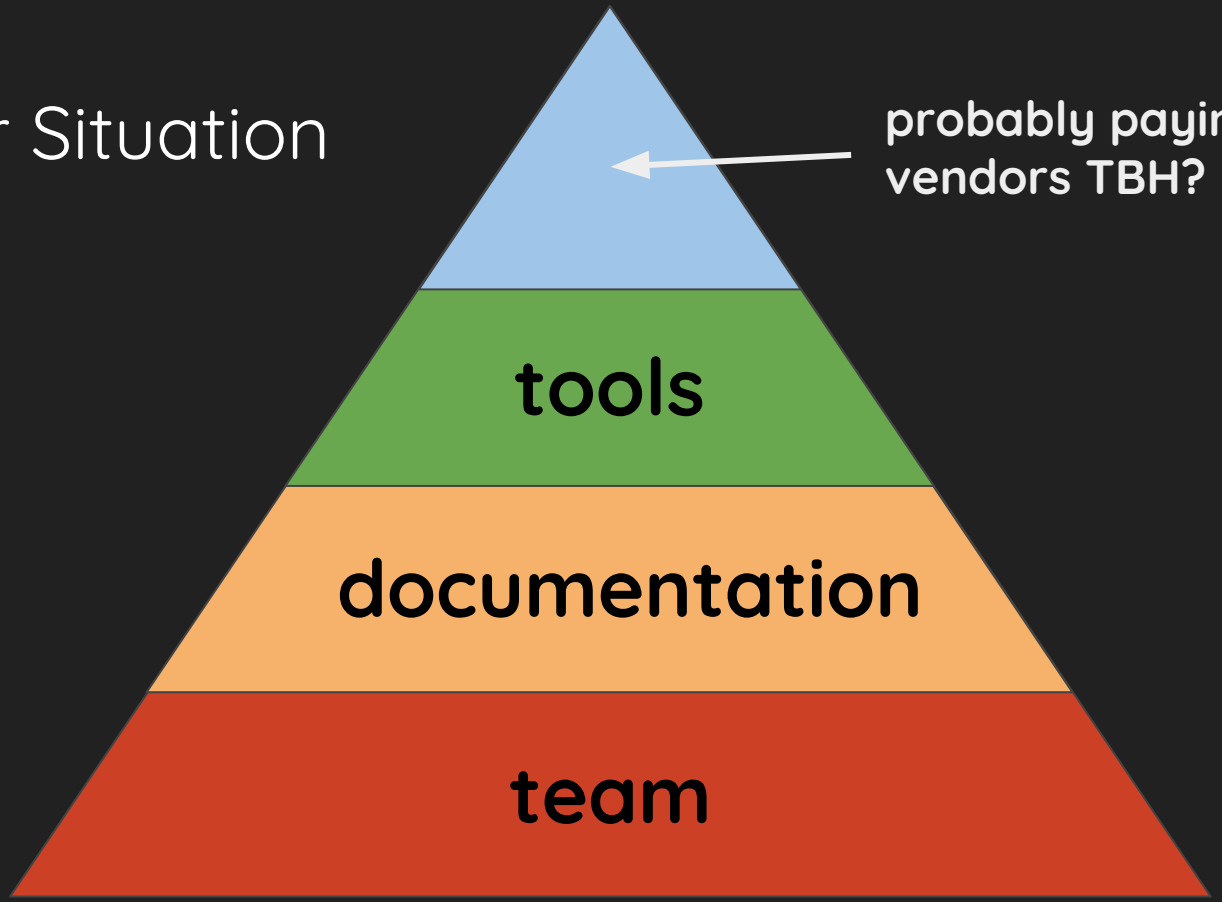
UX and Your Situation



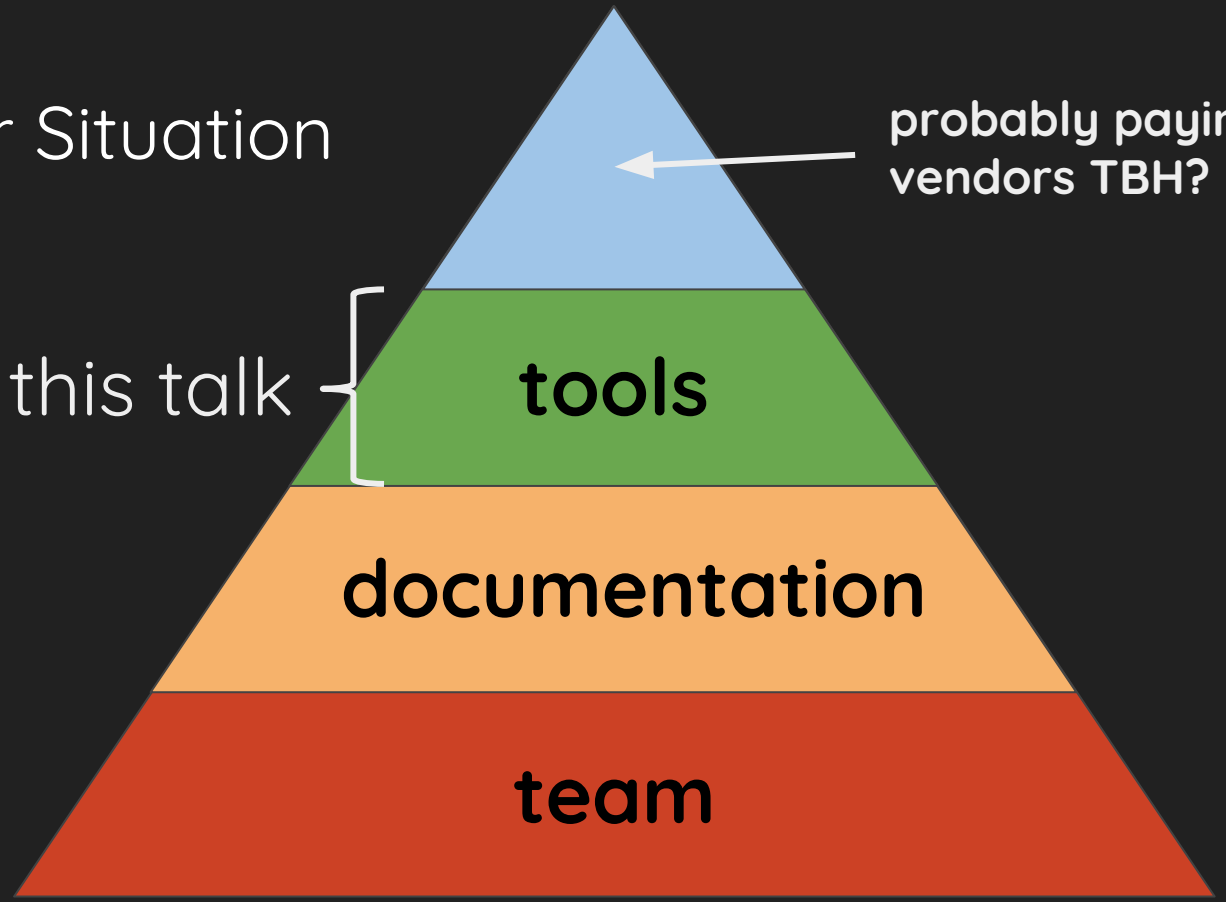
UX and Your Situation



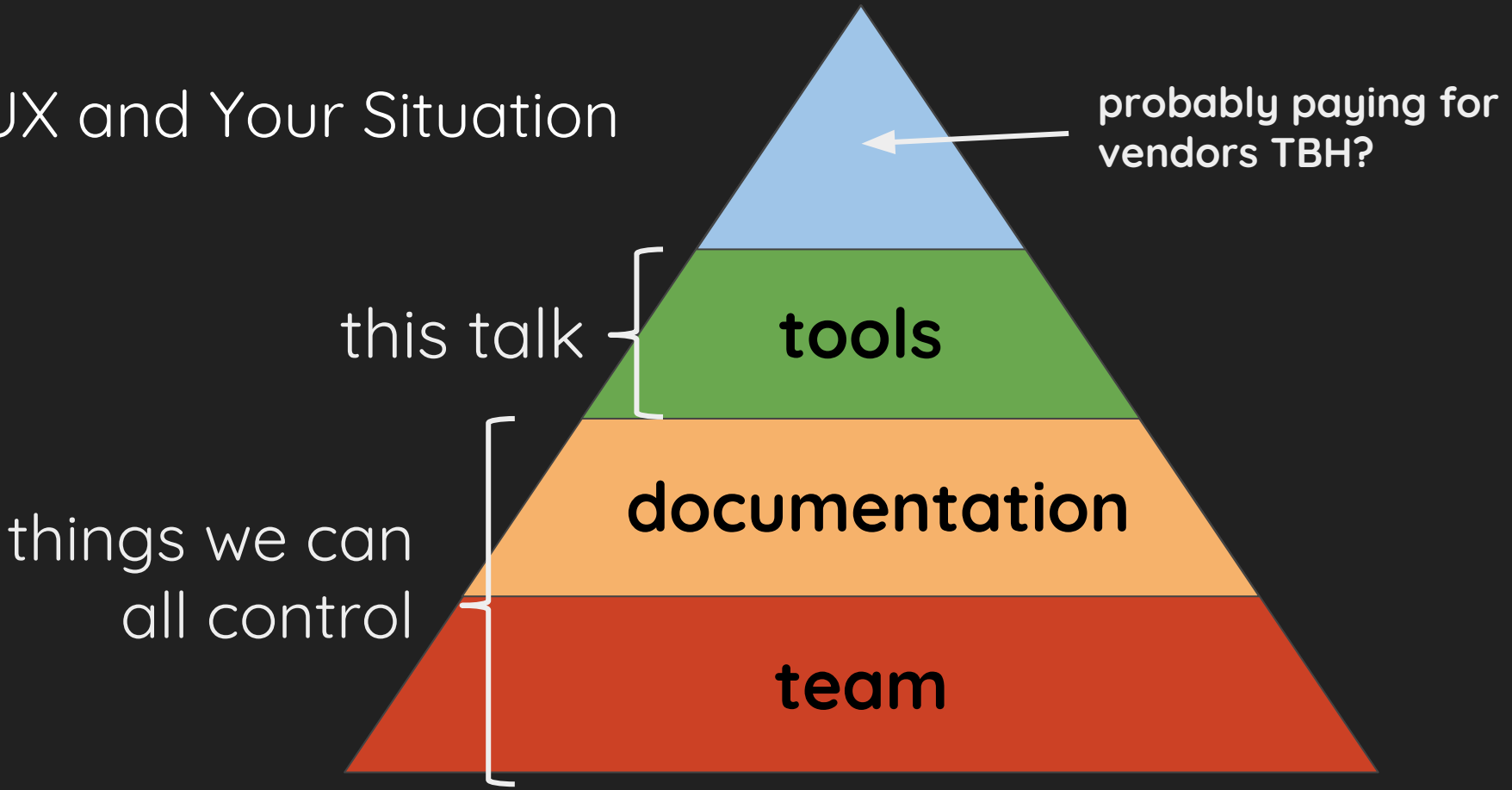
UX and Your Situation



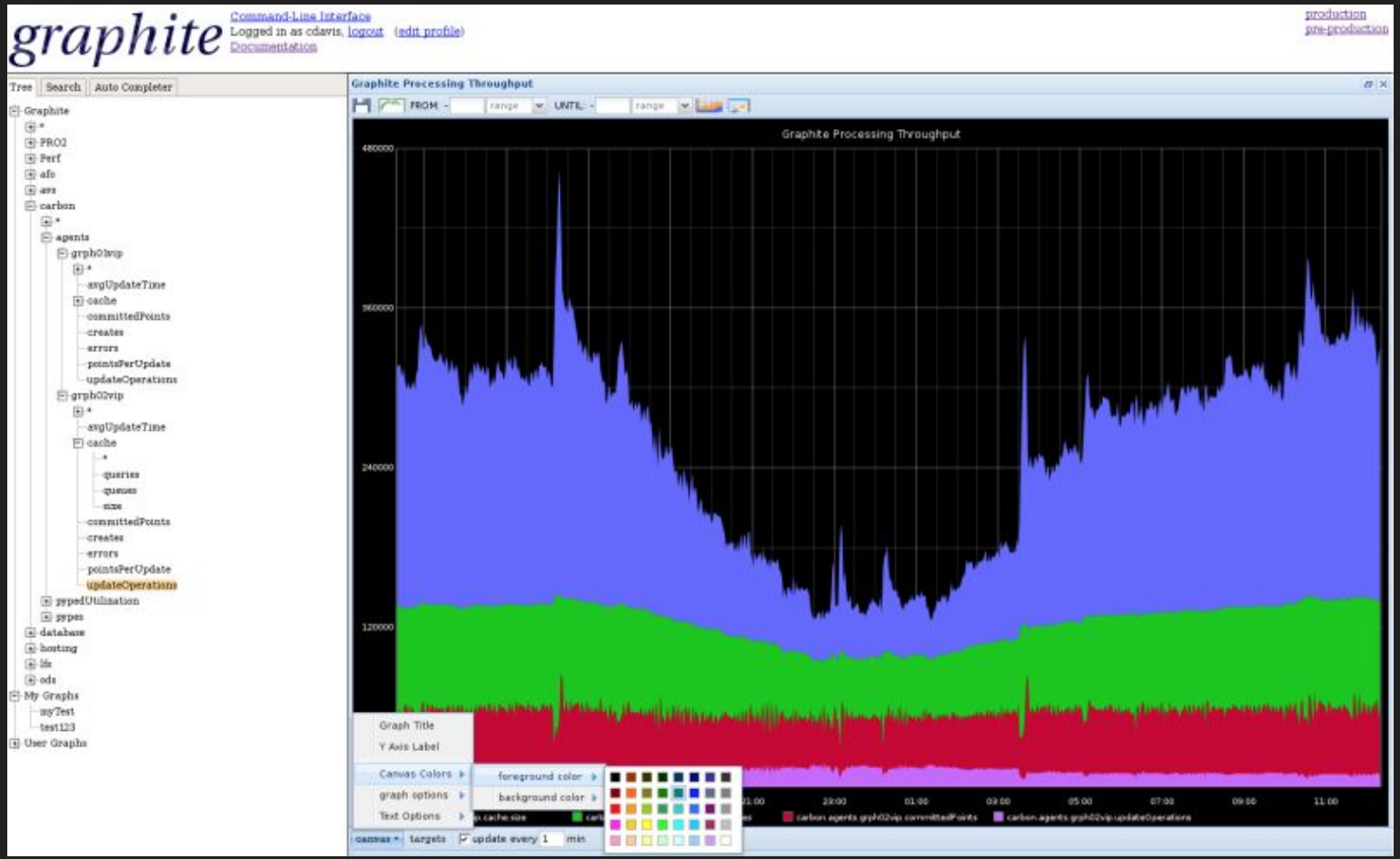
UX and Your Situation

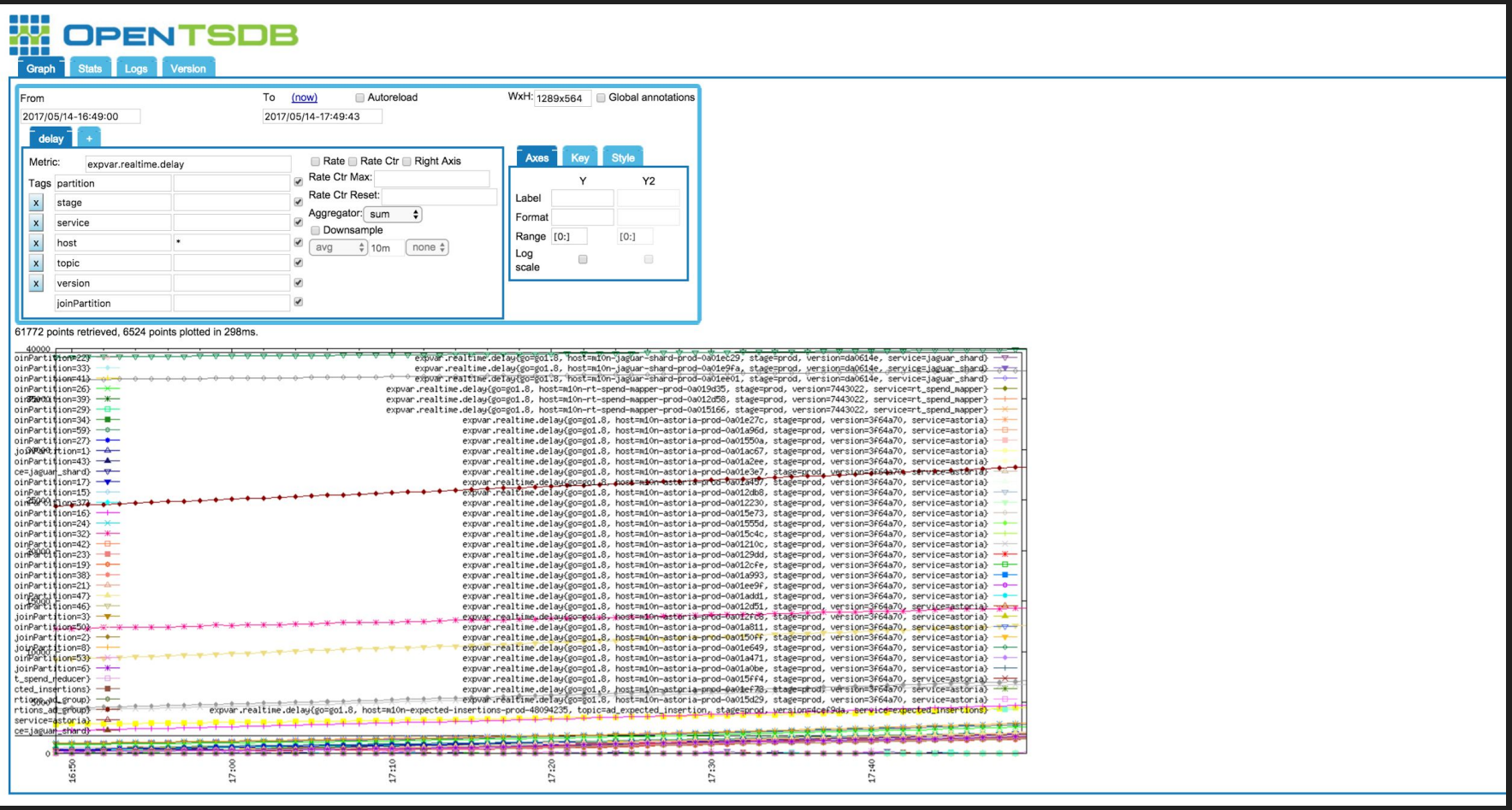


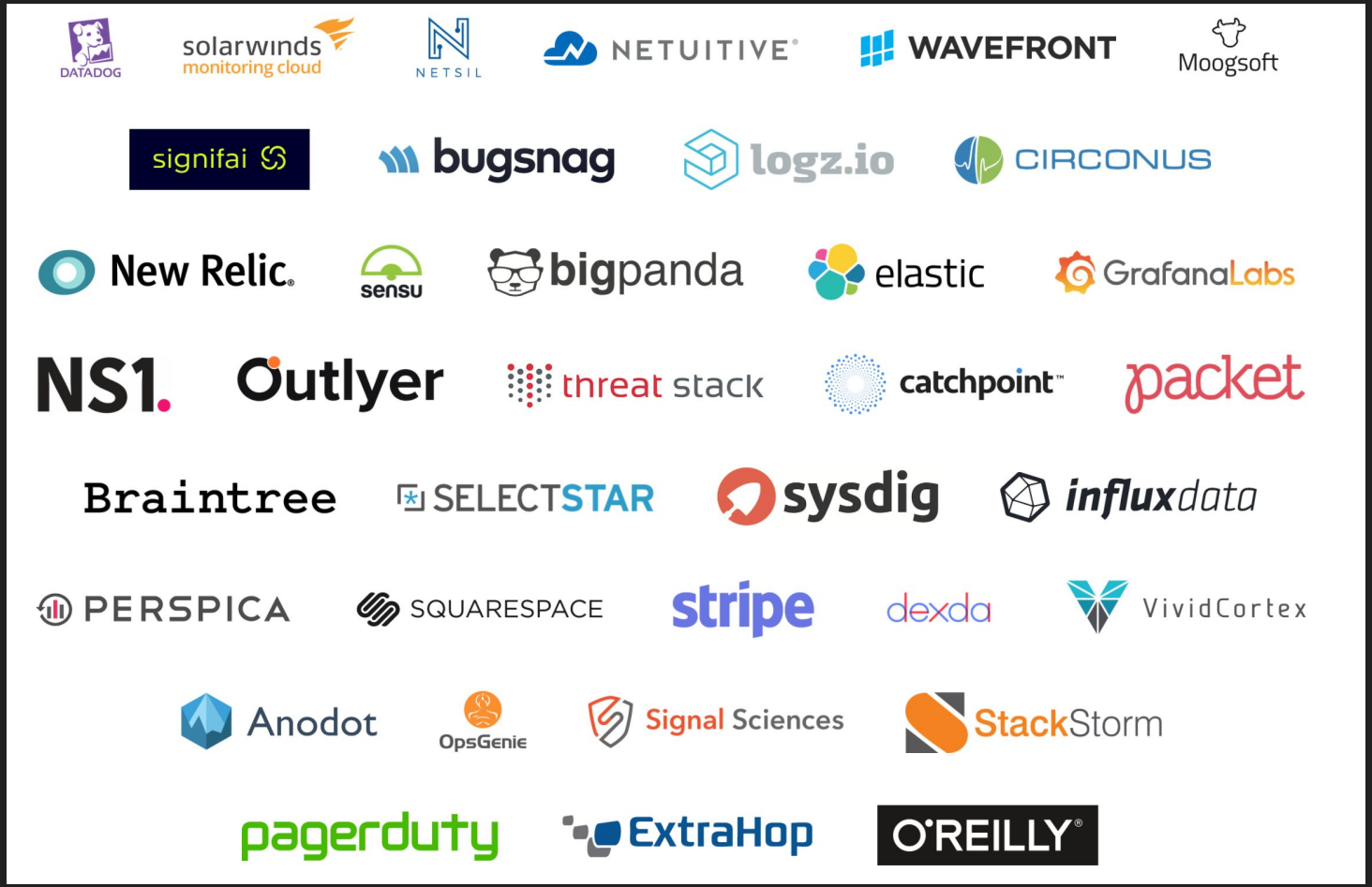
UX and Your Situation



Why should we care about user experience?


graphite
Command-Line Interface
Logged in as cdavis, logout (edit profile)
Documentation
production
pre-production
Tree
Search
Auto Completer
Graphite
carbon
agents
grph0lvip
grph02vip
cache
committedPoints
creates
errors
pointsPerUpdate
updateOperations
queries
queues
size
My Graphs
myTest
test123
User Graphs
Graphite Processing Throughput
FROM
UNTIL
480000
360000
240000
120000
Graph Title
Y Axis Label
Canvas Colors
graph options
Text Options
foreground color
background color
carbon.agents.grph02vip.committedPoints
carbon.agents.grph02vip.updateOperations
canvas
targets
update every 1
min


OpenTSDB
Graph
Stats
Logs
Version
From
2017/05/14-16:49:00
To
(now)
Autoreload
WxH:
1289x564
Global annotations
2017/05/14-17:49:43
delay
+
Metric:
expvar.realtime.delay
Rate
Rate Ctr
Right Axis
Tags
partition
stage
service
host
*
topic
version
joinPartition
Rate Ctr Max:
Rate Ctr Reset:
Aggregator:
sum
Downsample
avg
10m
none
Axes
Key
Style
Y
Y2
Label
Format
Range
[0:]
[0:]
Log scale
61772 points retrieved, 6524 points plotted in 298ms.
40000
30000
20000
10000
0
16:50
17:00
17:10
17:20
17:30
17:40


DATADOG
solarwinds
monitoring cloud
NETSIL
NETUITIVE
WAVEFRONT
Moogsoft
signifai
bugsnag
logz.io
CIRCONUS
New Relic
sensu
bigpanda
elastic
GrafanaLabs
NS1.
Outlyer
threat stack
catchpoint
packet
Braintree
SELECTSTAR
sysdig
influxdata
PERSPICA
SQUARESPACE
stripe
dexda
VividCortex
Anodot
OpsGenie
Signal Sciences
StackStorm
pagerduty
ExtraHop
O'REILLY

Why should we care about user experience?

Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data

Why should we care about user experience?

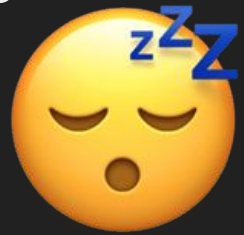
- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data
- **Developer velocity:** help people reach conclusions faster, help your company move faster

Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data
- **Developer velocity:** help people reach conclusions faster, help your company move faster
- **Data democracy:** you don't know what questions people want to answer with their own data

Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting timeseries data
- **Developer velocity:** help people reach conclusions faster, help your company move faster
- **Data democracy:** you don't know what questions people want to answer with their own data



Why should we care about user experience?

Because we can help engineers work *correctly*, *quickly*, and *independently*.

"The fastest way to become a 10x engineer is to help 10 other engineers do their jobs better.

- Wayne Gretzky"

- Michael Scott

Agenda

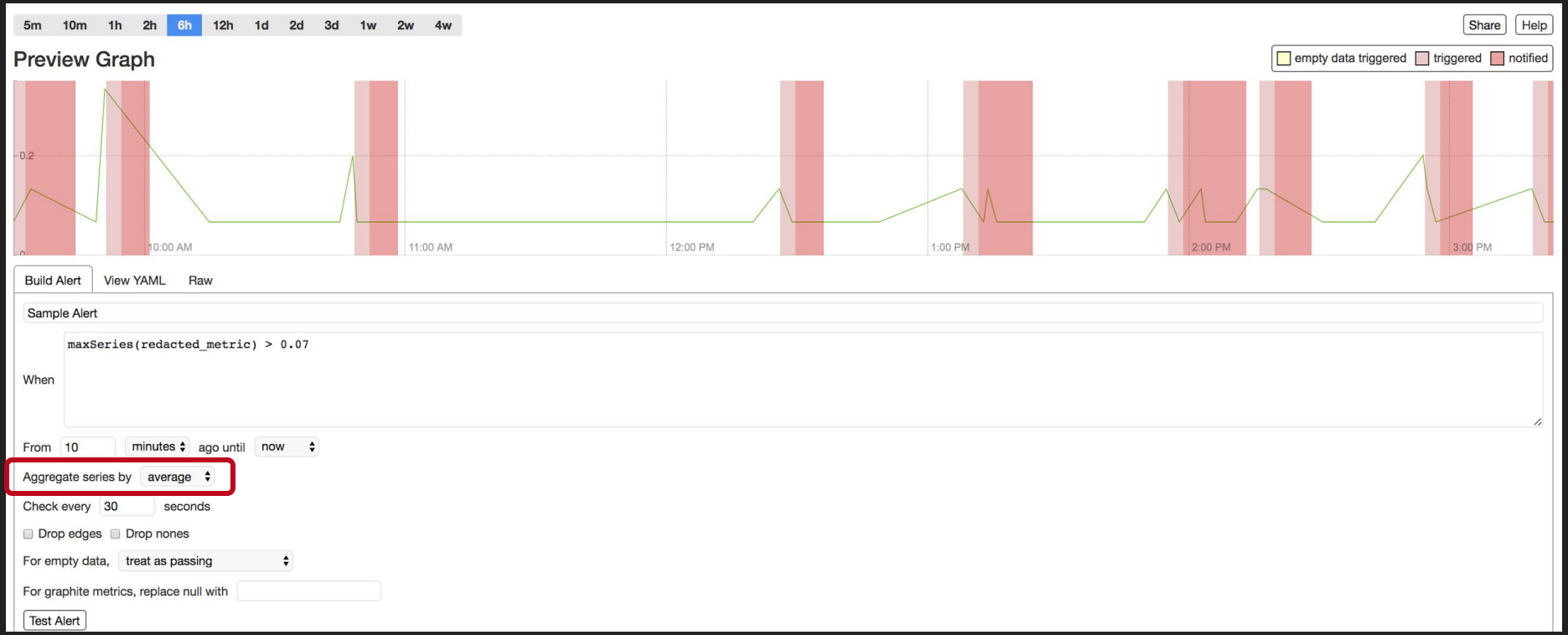
- Motivation
- **Sharing what you know**
- Designing what your users want
- Recap

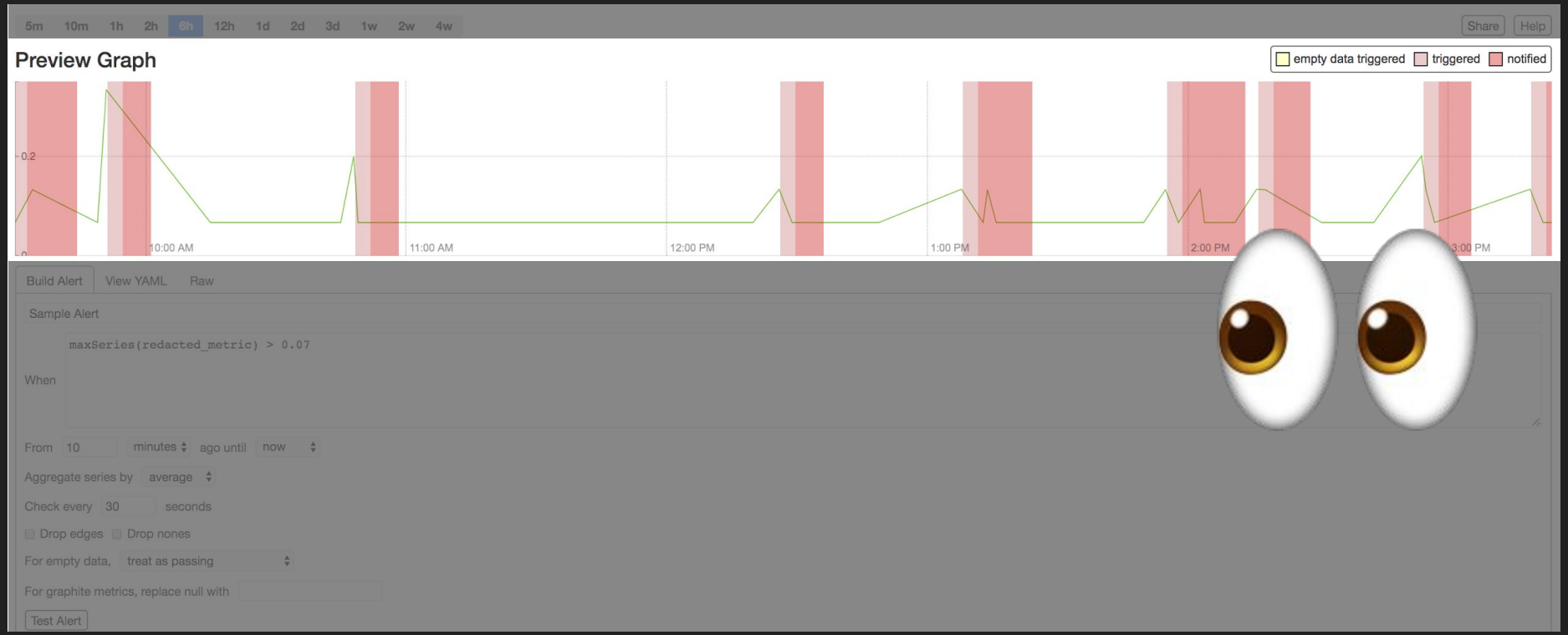
Sharing what you know

Sharing what you know

1. **Education vs intuition:** Don't overload people with too much information.

stats.example.metric.errors > 5

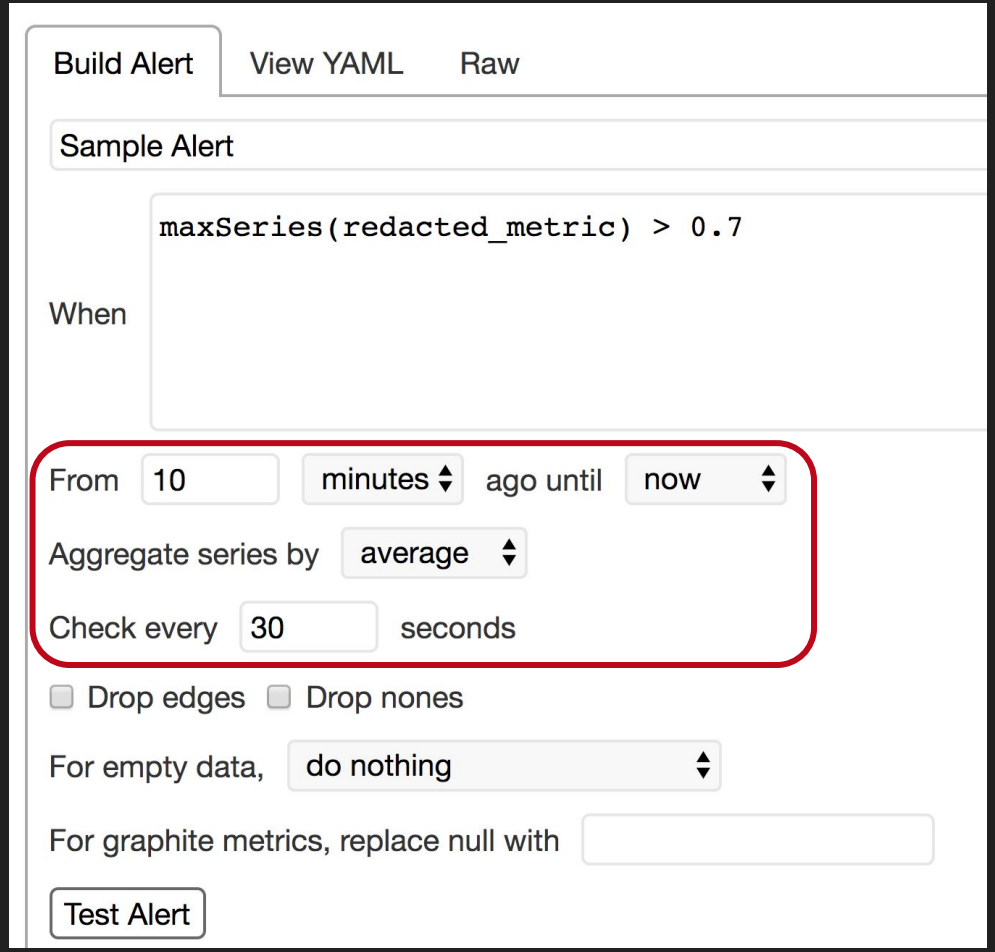

5m 10m 1h 2h 6h 12h 1d 2d 3d 1w 2w 4w
Share
Help
Preview Graph
empty data triggered
triggered
notified
0.2
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
Build Alert
View YAML
Raw
Sample Alert
When
maxSeries(redacted_metric) > 0.07
From 10 minutes ago until now
Aggregate series by average
Check every 30 seconds
Drop edges
Drop nones
For empty data, treat as passing
For graphite metrics, replace null with
Test Alert


5m 10m 1h 2h 6h 12h 1d 2d 3d 1w 2w 4w
Share Help
Preview Graph
empty data triggered triggered notified
0.2
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
Build Alert View YAML Raw
Sample Alert
maxSeries(redacted_metric) > 0.07
When
From 10 minutes ago until now
Aggregate series by average
Check every 30 seconds
Drop edges Drop nones
For empty data, treat as passing
For graphite metrics, replace null with
Test Alert

Sharing what you know

1. **Education vs intuition:** Don't overload people with too much information.

2. **Best practices:** Use your expertise to determine the most helpful default behavior.

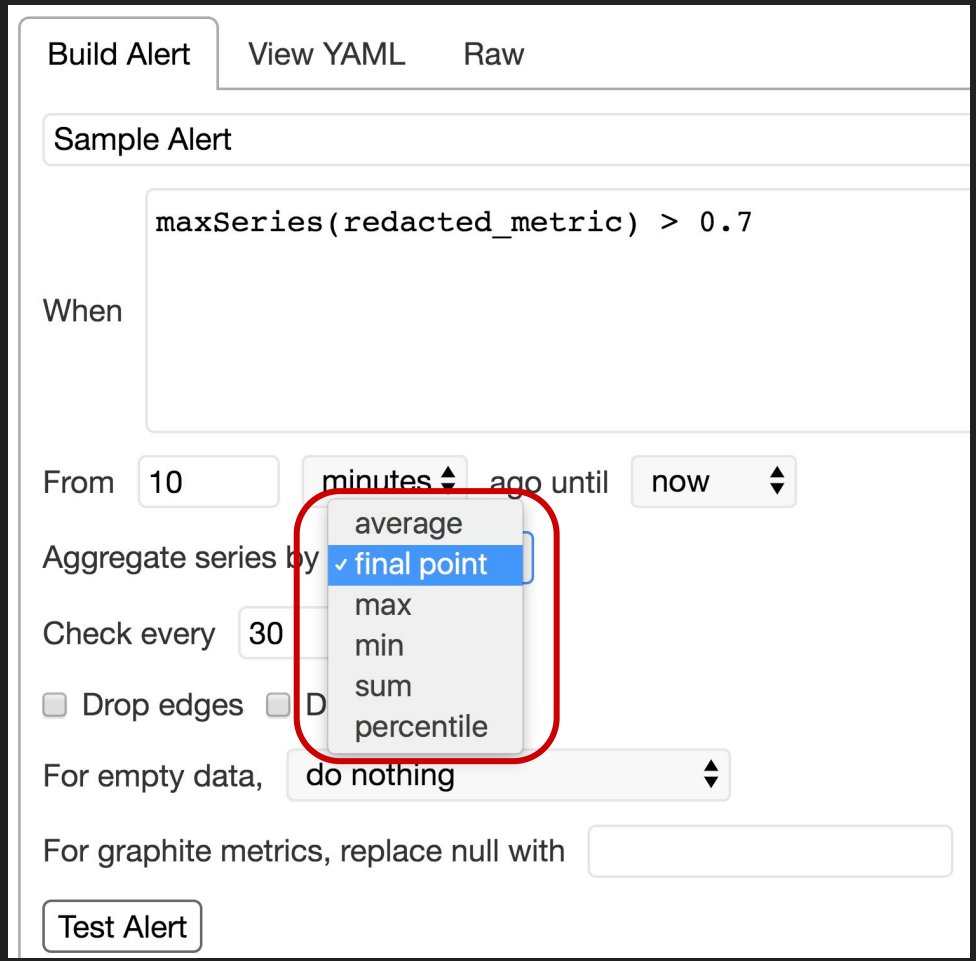

Build Alert
View YAML
Raw
Sample Alert
When
maxSeries(redacted_metric) > 0.7
From
10
minutes
ago until
now
Aggregate series by
average
Check every
30
seconds
Drop edges
Drop nones
For empty data,
do nothing
For graphite metrics, replace null with
Test Alert

Sharing what you know

1. **Education vs intuition:** Don't overload people with too much information.

2. **Best practices:** Use your expertise to determine the most helpful default behavior.

3. **Potential pitfalls:** Make it hard to do the wrong thing.


Build Alert
View YAML
Raw
Sample Alert
When
maxSeries(redacted_metric) > 0.7
From
10
minutes
ago until
now
Aggregate series by
average
final point
max
min
sum
percentile
Check every
30
Drop edges
For empty data,
do nothing
For graphite metrics, replace null with
Test Alert

Build Alert | View YAML | Raw

Sample Alert

When `maxSeries(redacted_metric) > 0.7`

From 10 minutes ago until now

Aggregate series by

- average
- ✓ final point
- max
- min
- sum
- percentile

Check every 30

Drop edges D

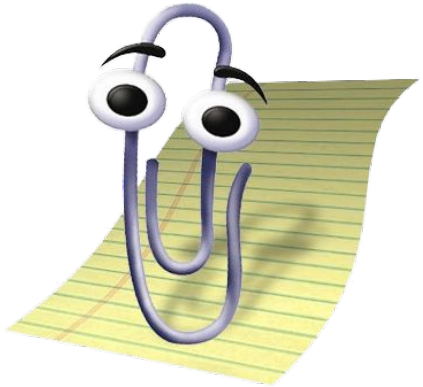
For empty data, do nothing

For graphite metrics, replace null with

Test Alert

It looks like you're trying to alert on the most recent data.

Are you sure you want to do that?



Agenda

- Motivation
- Sharing what you know
- **Designing what your users want**
- Recap

Designing what your users want

Designing what your users want

1. **Performance:** Do whatever it takes to make it fast.

Performance: Cool Ideas!

Performance: Cool Ideas!

- Backend

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges

Performance: Cool Ideas!

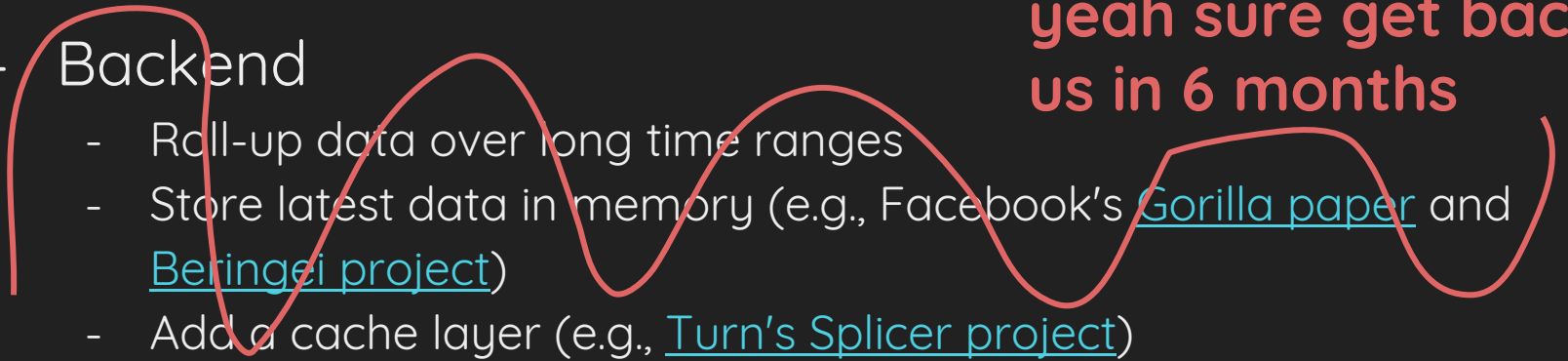
- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)



yeah sure get back to us in 6 months

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)
- Frontend

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)
- Frontend
  - Don't reload existing data if the user changes the time window

Performance: Cool Ideas!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)
- Frontend
  - Don't reload existing data if the user changes the time window
  - Prevent the user from requesting the data incessantly

Performance: Cool Ideas!

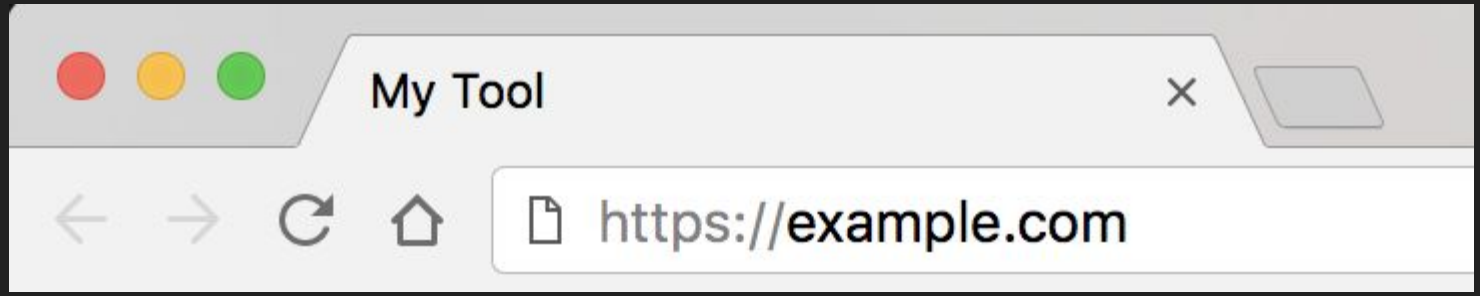
- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook's Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn's Splicer project)
- Frontend
  - Don't reload existing data if the user changes the time window
  - Prevent the user from requesting the data incessantly
  - Lazy-load graphs on a dashboard

Designing what your users want

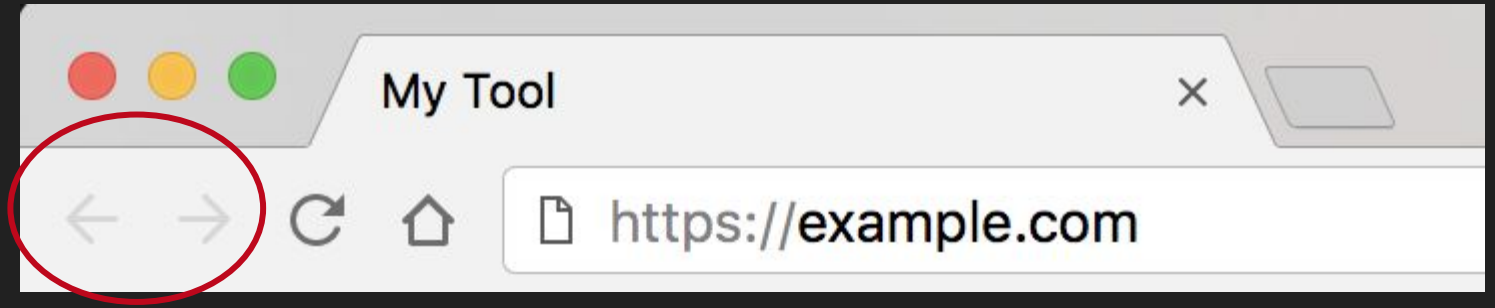
1. **Performance:** Do whatever it takes to make it fast.

2. **Exploration:** Make it easy to try things without fear.

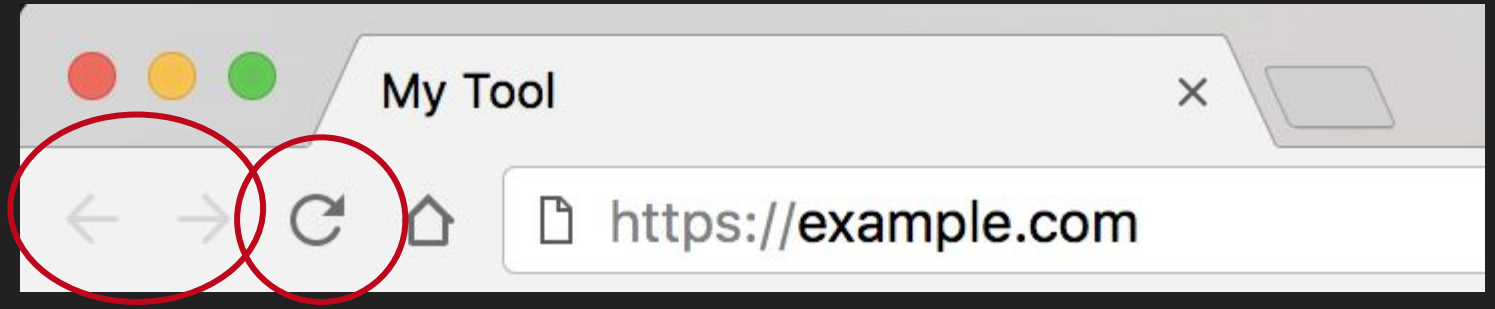
don't mess with this



don't mess with this

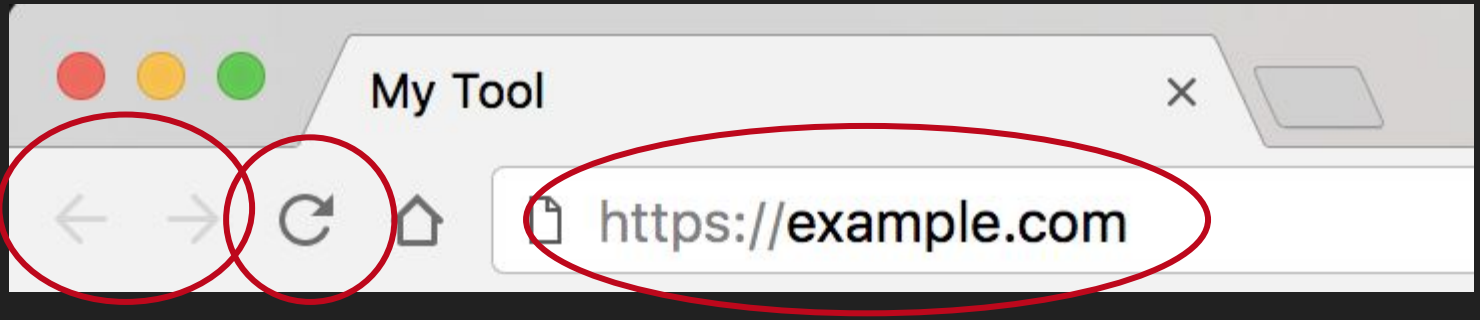


don't mess with this



or this

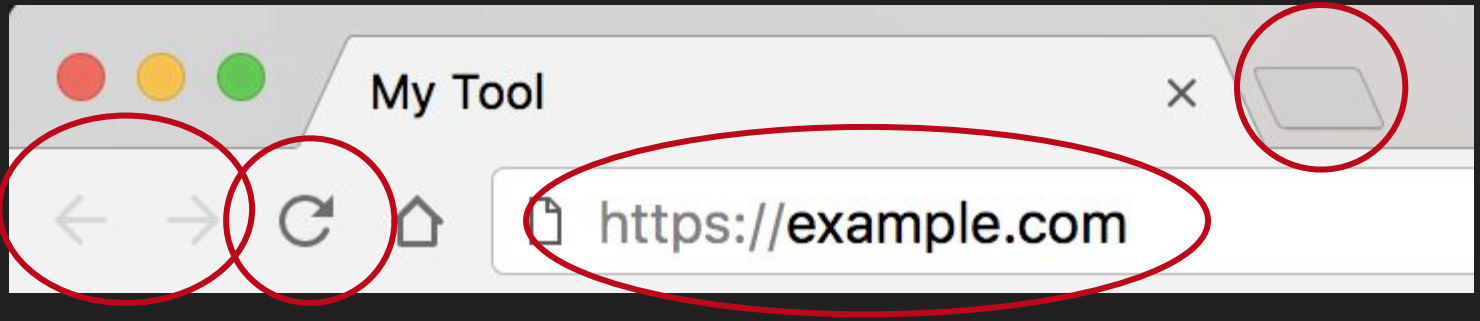
don't mess with this



or this

don't change it

don't mess with this



or this

don't change it

It doesn't deserve to be treated this way
it's just trying to help

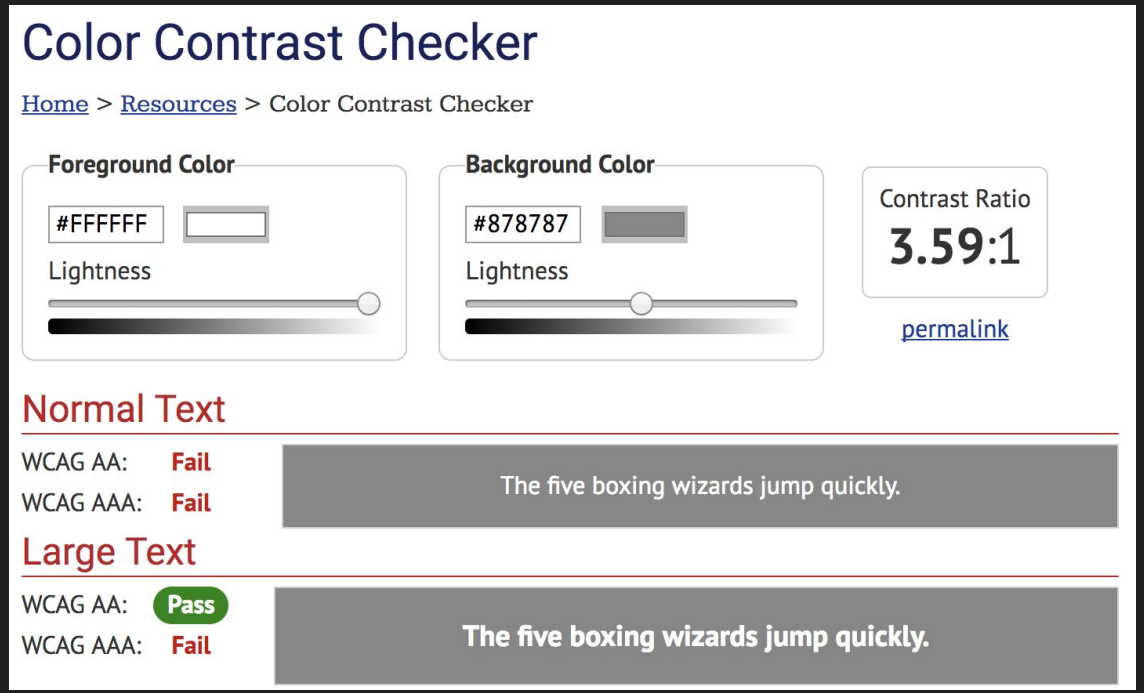
Accessibility is for everyone

Accessibility is for everyone

- Color contrast

Accessibility is for everyone

- Color contrast



Accessibility is for everyone

- Color contrast
- Customizable color schemes

Accessibility is for everyone

- Color contrast
- Customizable color schemes
- Navigation bar that isn't entirely controlled by JavaScript

Accessibility is for everyone

- Color contrast
- Customizable color schemes
- Navigation bar that isn't entirely controlled by JavaScript
- Keyboard navigation

Accessibility is for everyone

- Color contrast
- Customizable color schemes
- Navigation bar that isn't entirely controlled by JavaScript
- Keyboard navigation
- Obvious focus on what needs attention

Exploration through previews

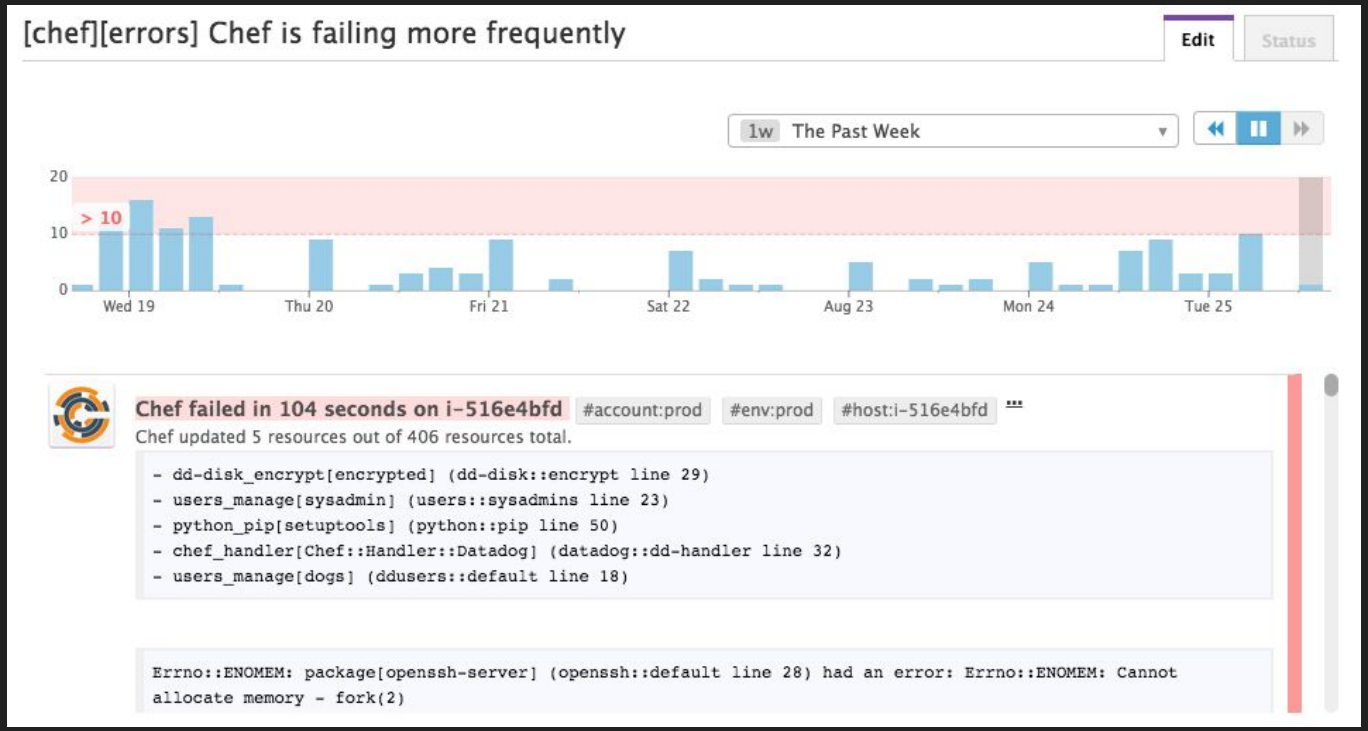
Exploration through previews: `terraform plan`

```
+ aws_dynamodb_table.my-dynamo-table
    arn:                          "<computed>"
    attribute.#:                  "1"
    attribute.4228504427.name:    "id"
    attribute.4228504427.type:    "S"
    hash_key:                     "id"
    name:                         "my-dynamo-table"
    read_capacity:                "2"
    stream_arn:                   "<computed>"
    stream_enabled:               "<computed>"
    stream_view_type:             "<computed>"
    write_capacity:               "2"

~ aws_elb.elb_my_test
    health_check.0.interval:                  "0" => "30"
    health_check.0.target:                    "HTTPS:3443/health" => "HTTPS:443/health"
    health_check.0.timeout:                   "5" => "3"
    health_check.0.unhealthy_threshold:       "10" => "2"
    listener.3438714722.instance_port:        "3443" => "0"
    listener.3438714722.instance_protocol:    "https" => ""
    listener.3438714722.lb_port:              "443" => "0"
    listener.3438714722.lb_protocol:          "https" => ""
    listener.3438714722.ssl_certificate_id:   "arn:aws:iam::46461245429:server-certificate/.example.com" => ""
    listener.3238714211.instance_port:        "" => "443"
    listener.3238714211.instance_protocol:    "" => "https"
    listener.3238714211.lb_port:              "" => "443"
    listener.3238714211.lb_protocol:          "" => "https"
    listener.3238714211.ssl_certificate_id:   "" => "arn:aws:iam::46461245429:server-certificate/.example.com"

~ aws_iam_policy.user-my-test
    policy:   }
                },
                {
                  "Effect": "Allow",
                  "Resource": [
            -       "arn:aws:s3:::my-test-development",
            -       "arn:aws:s3:::my-test-development/*"
            +       "arn:aws:s3:::my-test-development"
                  ],
                  "Action": [
                    "s3:*"
                  ],
                  "Condition": {
```

Exploration through previews: alert previews



Exploration recap

1. Don't break basic browser features.

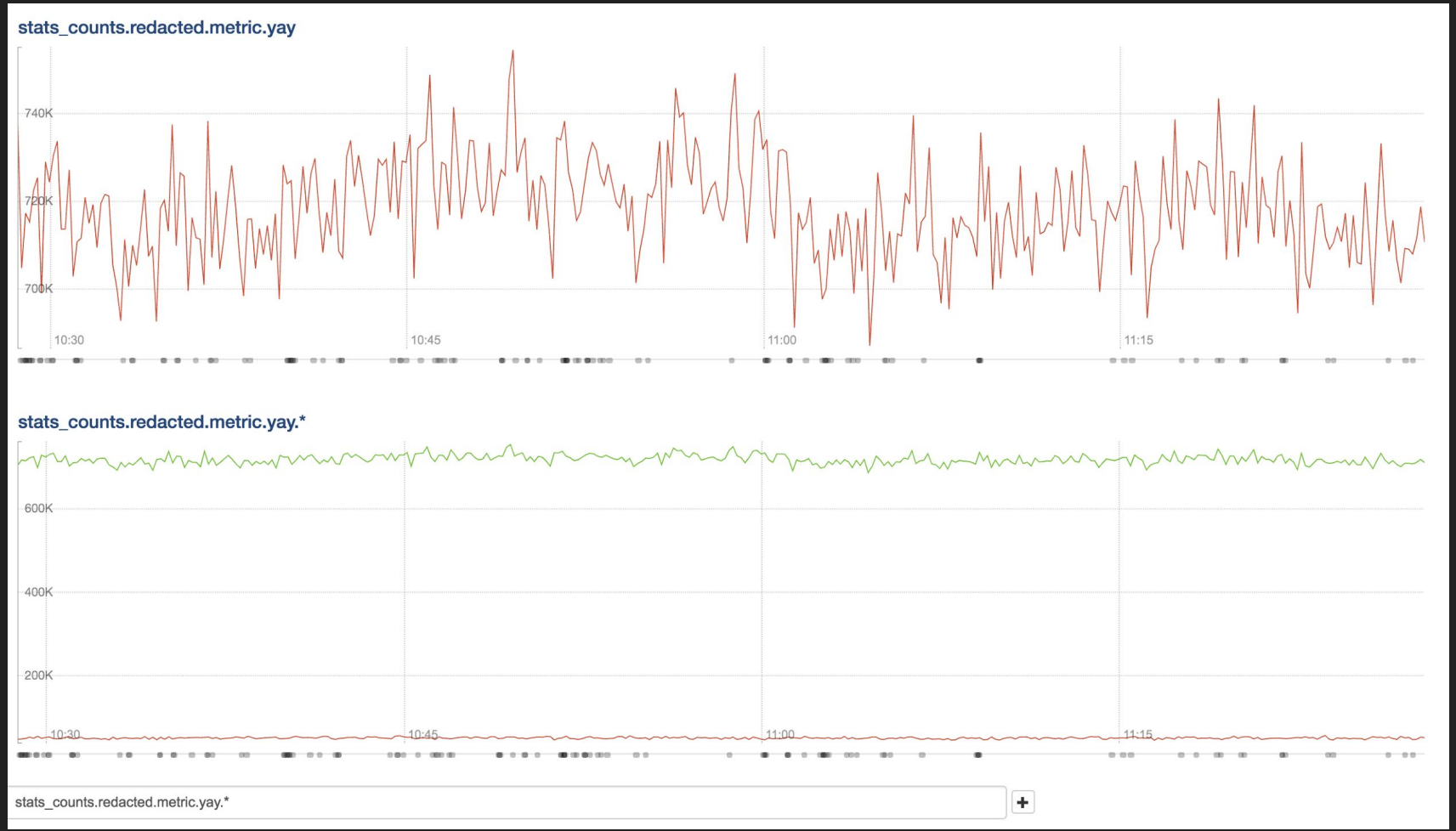
2. Design for accessibility.

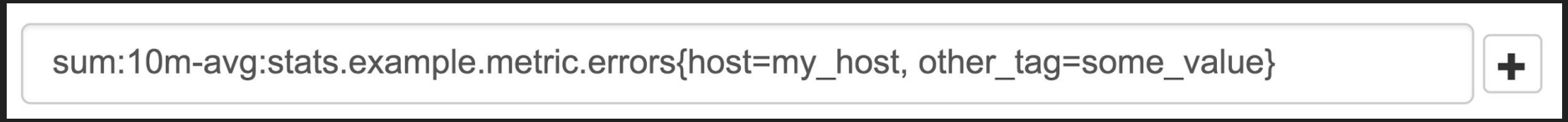
3. Make it easy to try things without making permanent changes.

Designing what your users want

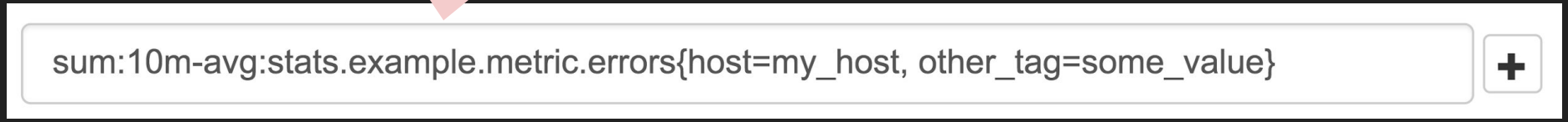
1. **Performance:** Do whatever it takes to make it fast.
2. **Exploration:** Make it easy to try things without fear.
3. **Simplicity:** Make it easy to figure out what to do.


Am I so out of touch?

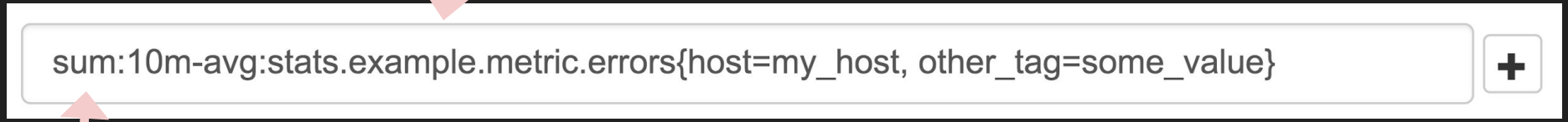

stats_counts.redacted.metric.yay
740K
720K
700K
10:30
10:45
11:00
11:15
stats_counts.redacted.metric.yay.*
600K
400K
200K
10:30
10:45
11:00
11:15
stats_counts.redacted.metric.yay.*


sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
+

manually type in your metric if
you know the name somehow

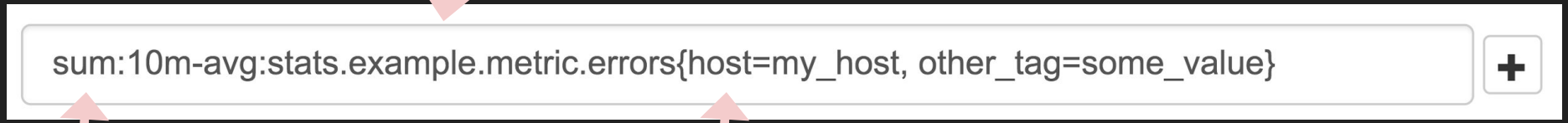


manually type in your metric if you know the name somehow



aggregator

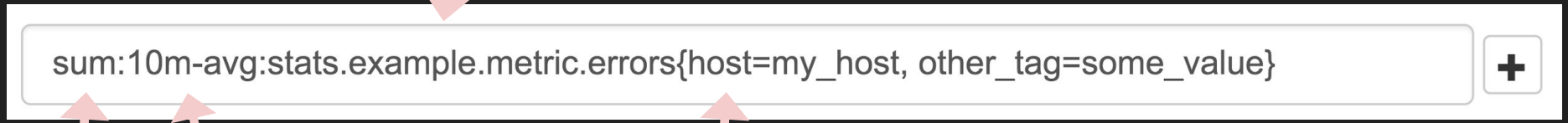
manually type in your metric if you know the name somehow



aggregator

hope you remember what tags are available for this metric lol

manually type in your metric if
you know the name somehow



aggregator

secret bonus:
you can downsample?!

hope you remember what tags are
available for this metric lol

manually type in your metric if
you know the name somehow

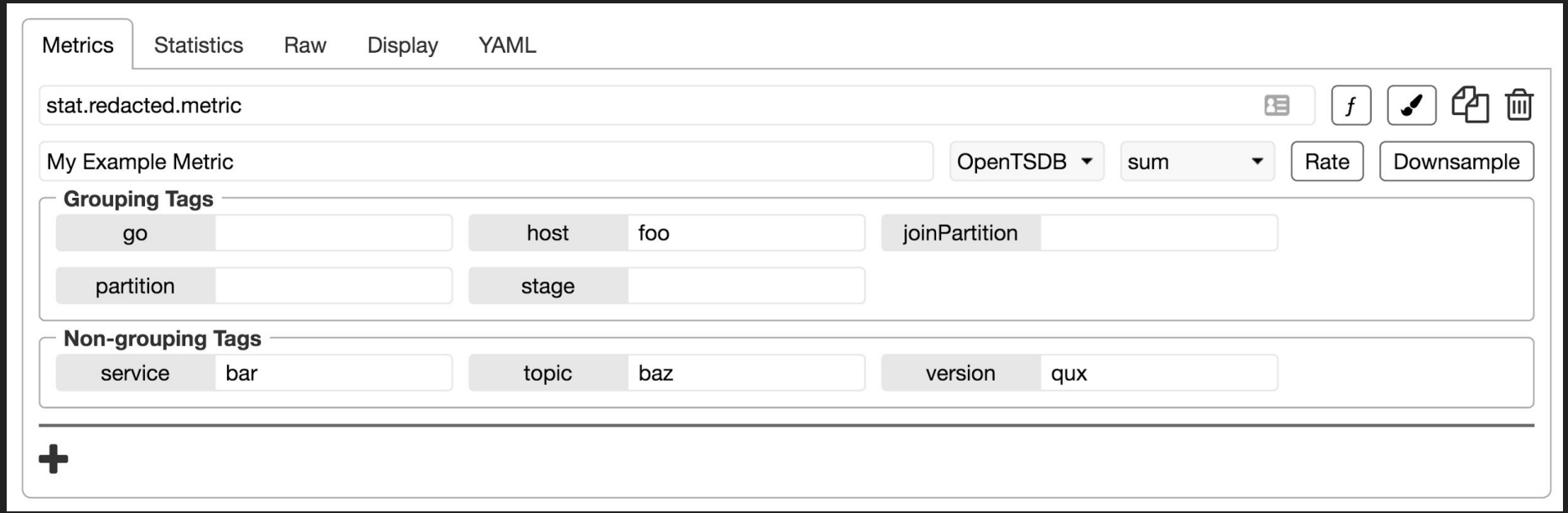
sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}

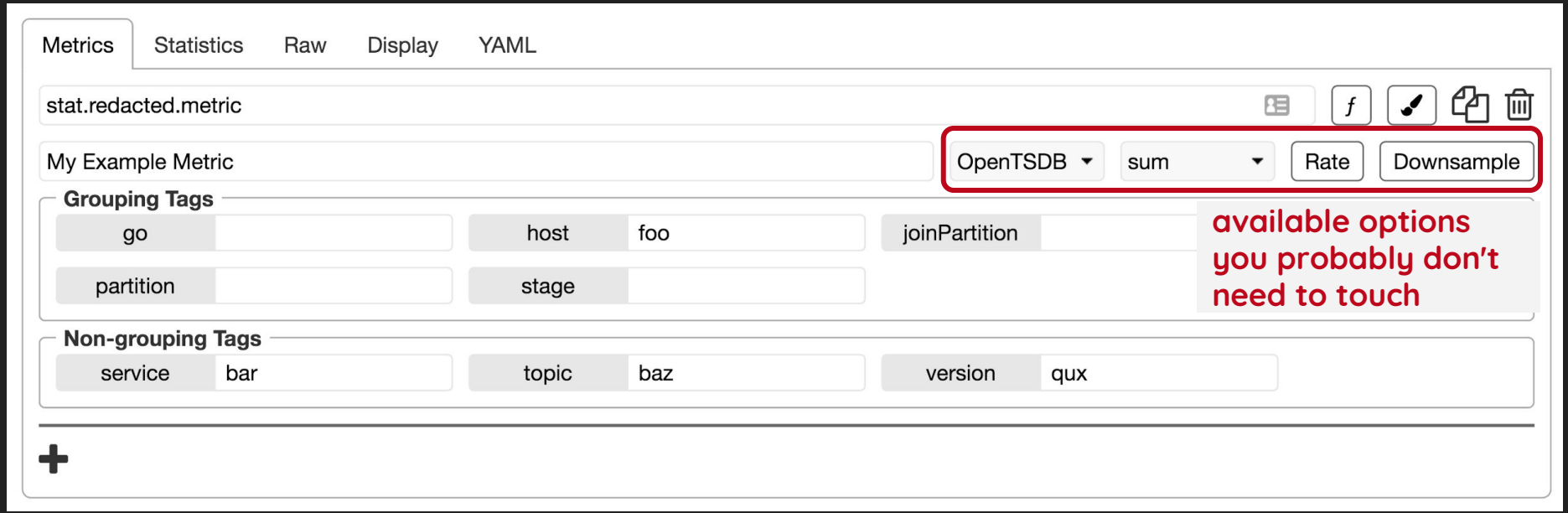
aggregator

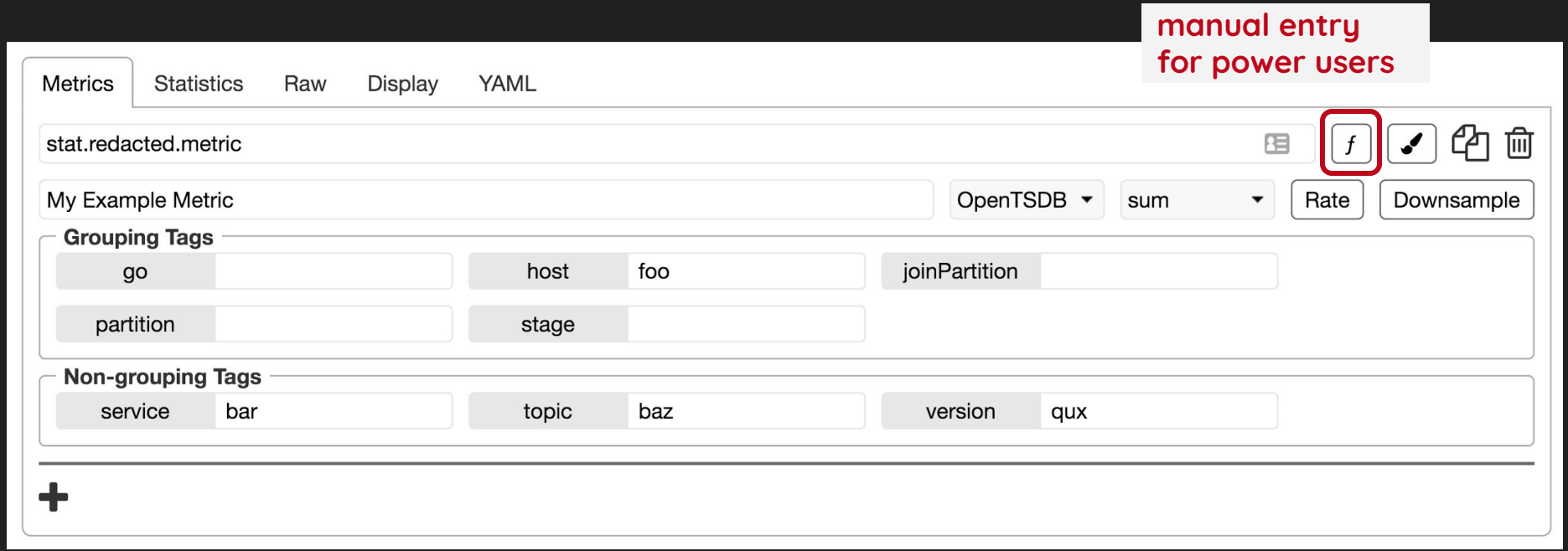
secret bonus:
you can downsample?!

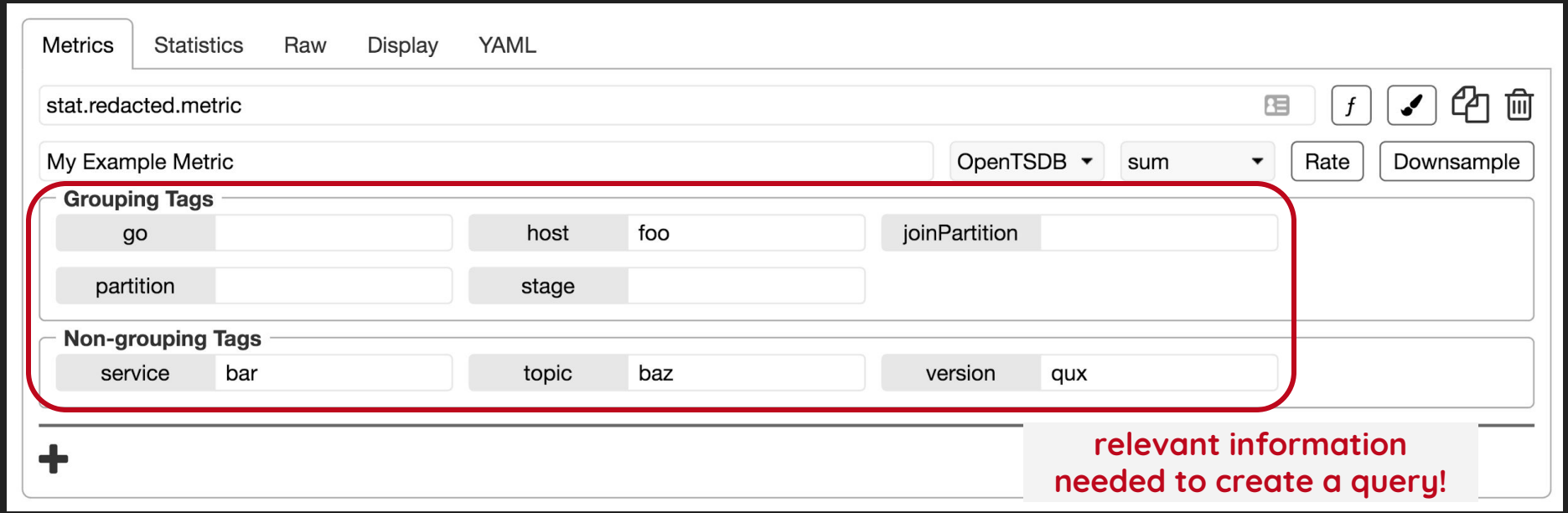
hope you remember what tags are
available for this metric lol




Metrics
Statistics
Raw
Display
YAML
stat.redacted.metric
My Example Metric
OpenTSDB
sum
Rate
Downsample
Grouping Tags
go
host
foo
joinPartition
partition
stage
Non-grouping Tags
service
bar
topic
baz
version
qux


Metrics
Statistics
Raw
Display
YAML
stat.redacted.metric
My Example Metric
OpenTSDB
sum
Rate
Downsample
Grouping Tags
go
host
foo
joinPartition
partition
stage
available options you probably don't need to touch
Non-grouping Tags
service
bar
topic
baz
version
qux


manual entry
for power users
Metrics
Statistics
Raw
Display
YAML
stat.redacted.metric
My Example Metric
OpenTSDB
sum
Rate
Downsample
Grouping Tags
go
host
foo
joinPartition
partition
stage
Non-grouping Tags
service
bar
topic
baz
version
qux


Metrics
Statistics
Raw
Display
YAML
stat.redacted.metric
My Example Metric
OpenTSDB
sum
Rate
Downsample
Grouping Tags
go
host
foo
joinPartition
partition
stage
Non-grouping Tags
service
bar
topic
baz
version
qux
relevant information needed to create a query!

Agenda

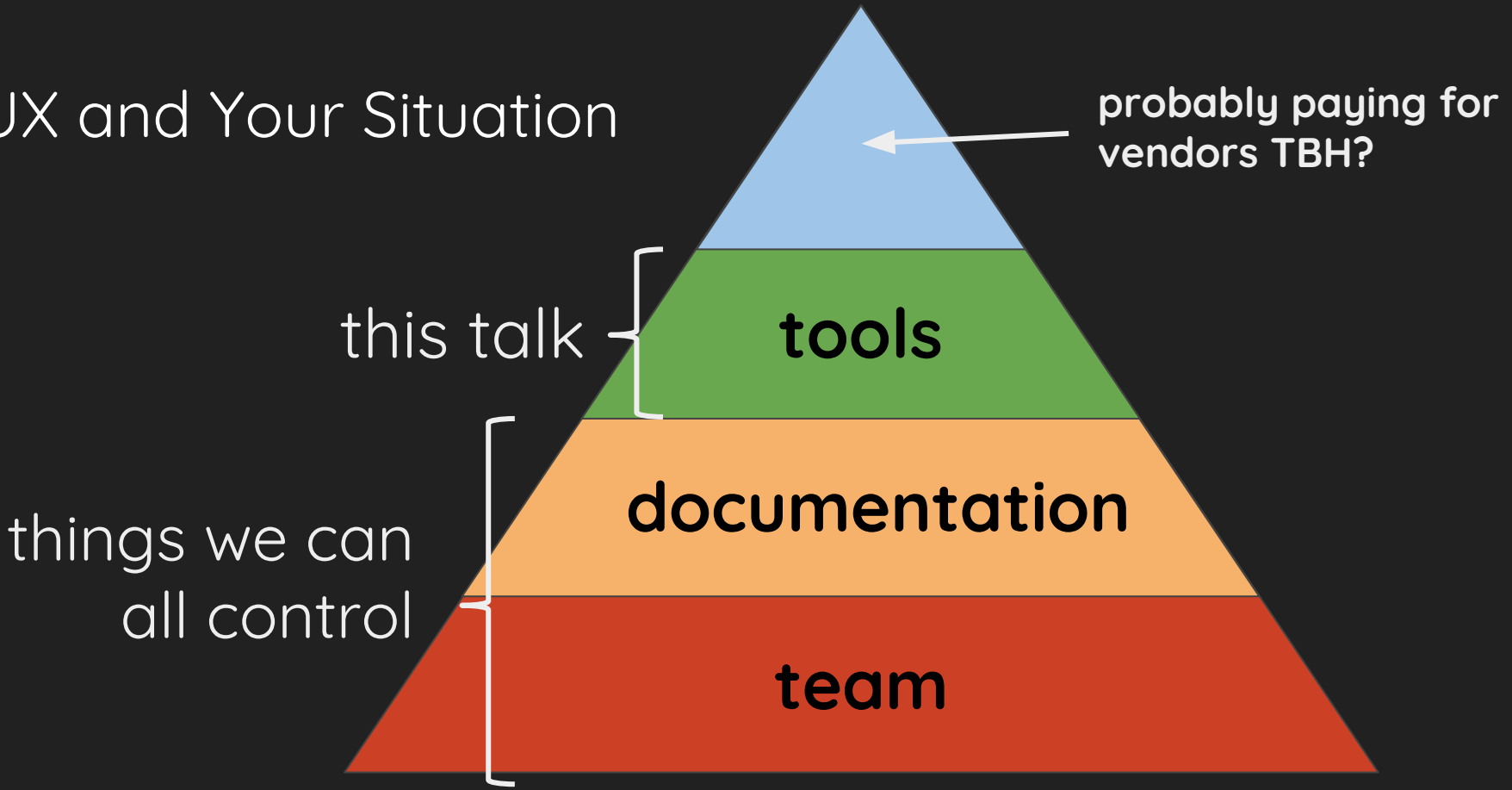
- Motivation
- Sharing what you know
- Designing what your users want
- **Recap**

"The fastest way to become a 10x engineer is to help 10 other engineers do their jobs better.

- Wayne Gretzky"

- Michael Scott

UX and Your Situation



Sharing what you know

1. **Education vs intuition:** Don't overload people with too much information.

2. **Best practices:** Use your expertise to determine the most helpful default behavior.

3. **Potential pitfalls:** Make it hard to do the wrong thing.

Designing what your users want

1. **Performance:** Do whatever it takes to make it fast.

2. **Exploration:** Make it easy to try things without fear.

3. **Simplicity:** Make it easy to figure out what to do.

Thanks! 

(also check out the plenary panel! 4pm tomorrow!)