Observable JS Apps

@eanakashima

about://

Emily Nakashima
product engineering manager
@ honeycomb.io

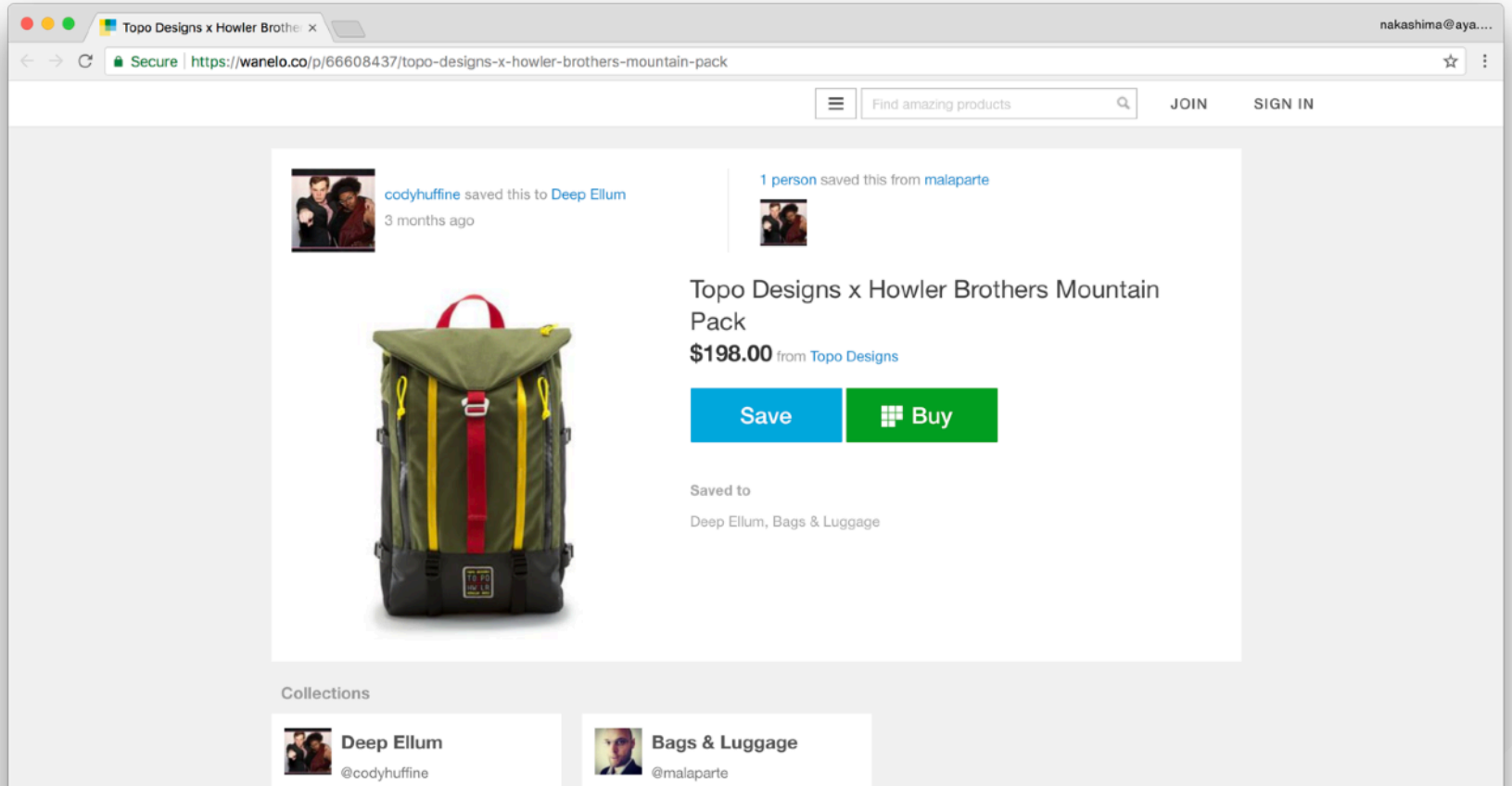

bugsnag


ModCloth

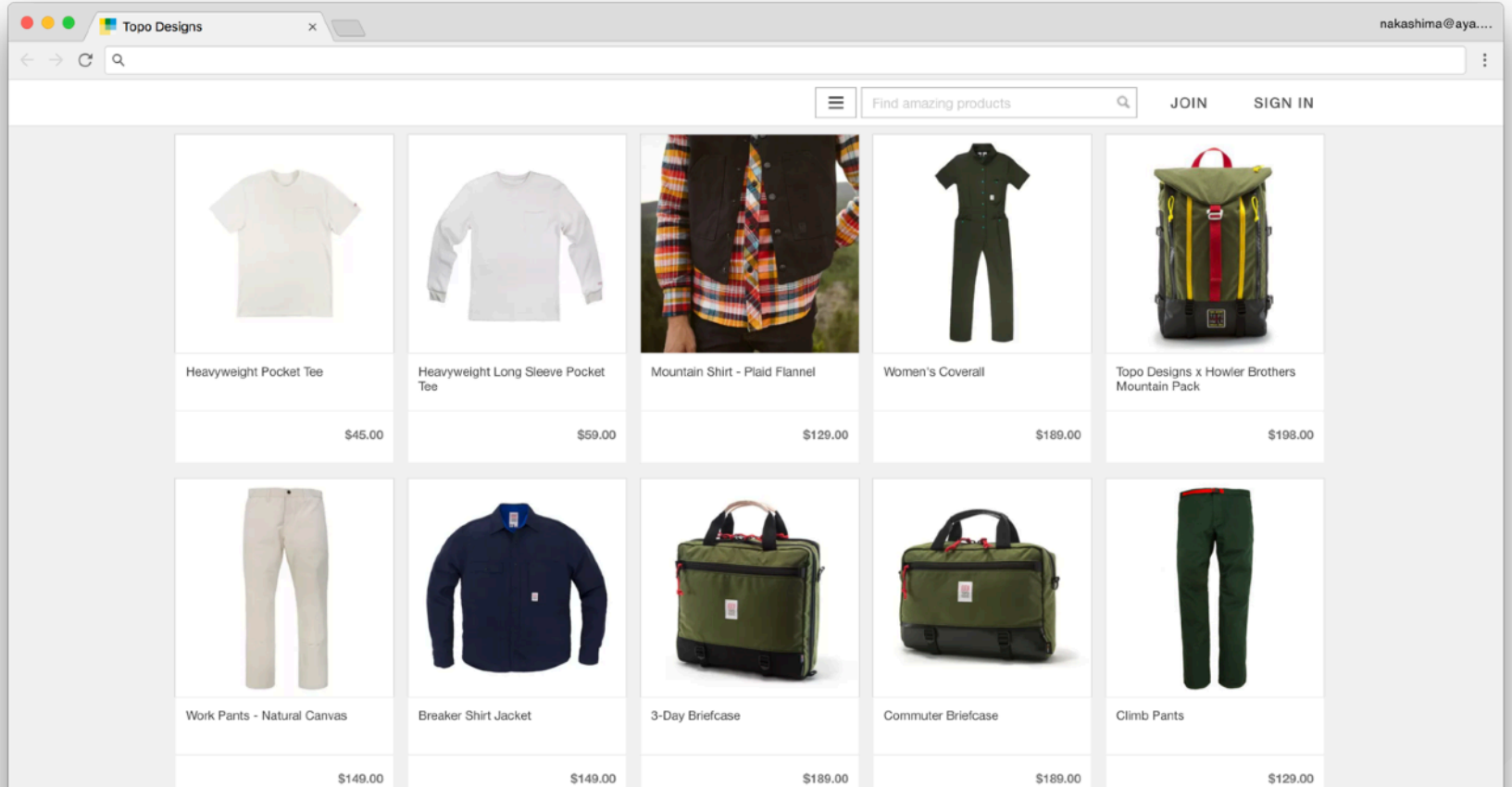

GitHub


charles
SCHWAB

The one where we DDOS'd ourselves



Epilogue


Topo Designs
nakashima@aya....
Find amazing products
JOIN
SIGN IN
Heavyweight Pocket Tee
$45.00
Heavyweight Long Sleeve Pocket Tee
$59.00
Mountain Shirt - Plaid Flannel
$129.00
Women's Coverall
$189.00
Topo Designs x Howler Brothers Mountain Pack
$198.00
Work Pants - Natural Canvas
$149.00
Breaker Shirt Jacket
$149.00
3-Day Briefcase
$189.00
Commuter Briefcase
$189.00
Climb Pants
$129.00

```
// if the page is long enough to scroll
if (document.body.clientHeight > window.innerHeight) {

    // add a scroll event listener
    document.addEventListener('scroll', function(e) {

        // if within 100px of the bottom of the page
        if (window.innerHeight + window.scrollY + 100 >
            document.body.clientHeight)
            fetchNextPage();
    });

// else fetch another page of results immediately
} else {
    fetchNextPage();
}
```

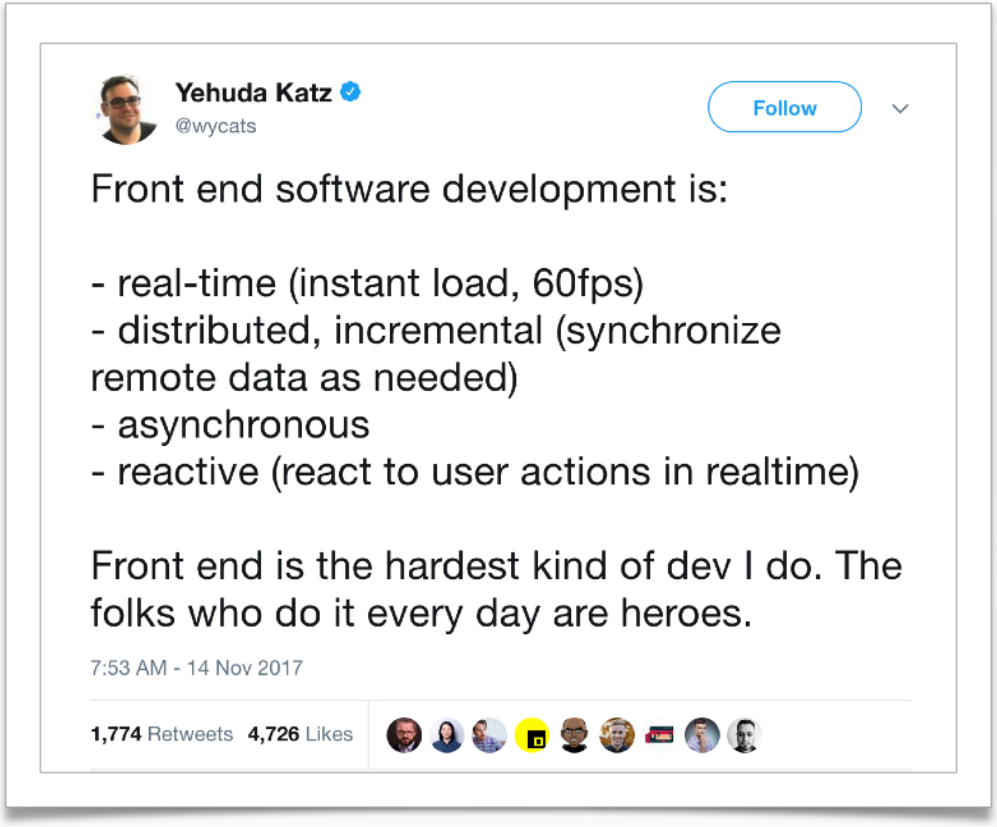
We had *lots* of production data

1. **Product Analytics**
2. **Metrics**
3. **Client-side metrics**
4. **Error monitoring**
5. **Logs**

What data (or graphs) would have helped?

- Total traffic count broken down into buckets by screen size.
- Average number of API requests triggered by a particular page type.

Frontend complexity is only increasing



Yehuda Katz
@wycats

Follow

Front end software development is:

- real-time (instant load, 60fps)
- distributed, incremental (synchronize remote data as needed)
- asynchronous
- reactive (react to user actions in realtime)

Front end is the hardest kind of dev I do. The folks who do it every day are heroes.

7:53 AM - 14 Nov 2017

1,774 Retweets **4,726** Likes

Observability

Instrument your code so that you can:

- ask any question, whether you anticipated it or not
- deeply understand the state of your system by observing its outputs

Backend Instrumentation

Honeycomb Architecture

labrador (static site)

basenji

Customer

ui.

data

api.

Give Me JPG!

basset (alerts)

poodle

shepherd

doodle

OK

retriever

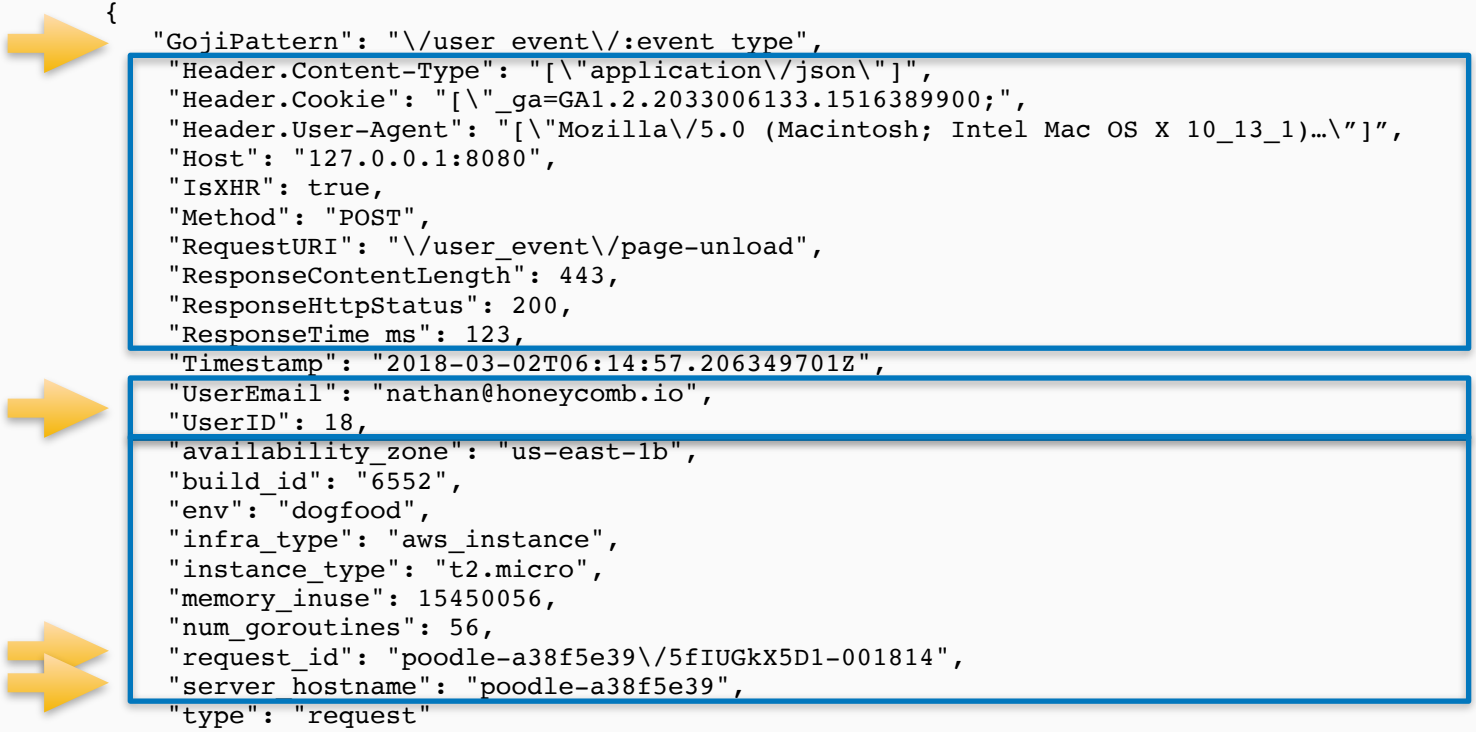
Storage Engine

kafka

Types of production data

1. **Metrics**
2. **Events:** sent to Honeycomb
    - Plus a few other tools that are “event-y”
        1. **Logs** (structured, no log aggregator)
        2. **Traces** (sometimes)
        3. **Error monitoring**
3. ~~**Product Analytics**~~

What's in an event?



```
{
  "GojiPattern": "\/user_event\/:event_type",
  "Header.Content-Type": "[\"application\/json\"]",
  "Header.Cookie": "[\"_ga=GA1.2.2033006133.1516389900;",
  "Header.User-Agent": "[\"Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_13_1)…\"]",
  "Host": "127.0.0.1:8080",
  "IsXHR": true,
  "Method": "POST",
  "RequestURI": "\/user_event\/page-unload",
  "ResponseContentLength": 443,
  "ResponseHttpStatus": 200,
  "ResponseTime_ms": 123,
  "Timestamp": "2018-03-02T06:14:57.206349701Z",
  "UserEmail": "nathan@honeycomb.io",
  "UserID": 18,
  "availability_zone": "us-east-1b",
  "build_id": "6552",
  "env": "dogfood",
  "infra_type": "aws_instance",
  "instance_type": "t2.micro",
  "memory_inuse": 15450056,
  "num_goroutines": 56,
  "request_id": "poodle-a38f5e39\/5fIUGkX5D1-001814",
  "server_hostname": "poodle-a38f5e39",
  "type": "request"
},
```

High cardinality

Fields that may have many unique values

Common examples:

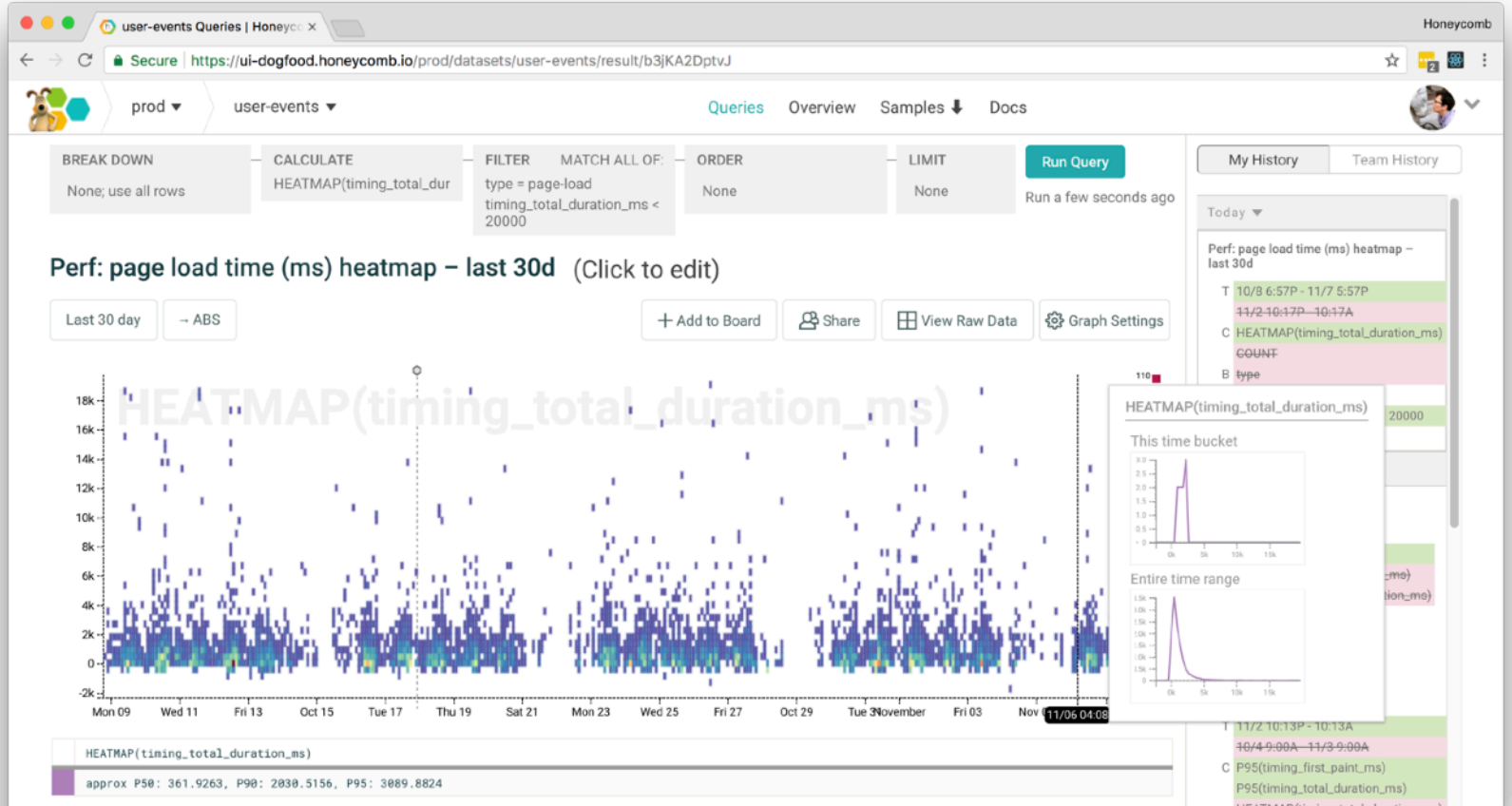
- email address
- username / user id / team id
- server hostname
- IP address
- user agent string
- build id
- request url
- feature flags / flag combinations

What's in an event?


```
{
   "GojiPattern": "\/user_event\/:event_type",
    "Header.Content-Type": "[\"application\/json\"]",
    "Header.Cookie": "[\"_ga=GA1.2.2033006133.1516389900;",
    "Header.User-Agent": "[\"Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_13_1)…\”]”,
    "Host": "127.0.0.1:8080",
    "IsXHR": true,
    "Method": "POST",
    "RequestURI": "\/user_event\/page-unload",
    "ResponseContentLength": 443,
    "ResponseHttpStatus": 200,
    "ResponseTime_ms": 123,
    "Timestamp": "2018-03-02T06:14:57.206349701Z",
    "UserEmail": "nathan@honeycomb.io",
    "UserID": 18,
    "availability_zone": "us-east-1b",
    "build_id": "6552",
    "env": "dogfood",
    "infra_type": "aws_instance",
    "instance_type": "t2.micro",
    "memory_inuse": 15450056,
    "num_goroutines": 56,
    "request_id": "poodle-a38f5e39\/5fIUGkX5D1-001814",
    "server_hostname": "poodle-a38f5e39",
    "type": "request"
},
```


Browser Instrumentation

Honeycomb Query Sandbox: React, SCSS, go templates, and lots of data



Instrumentation toolkit

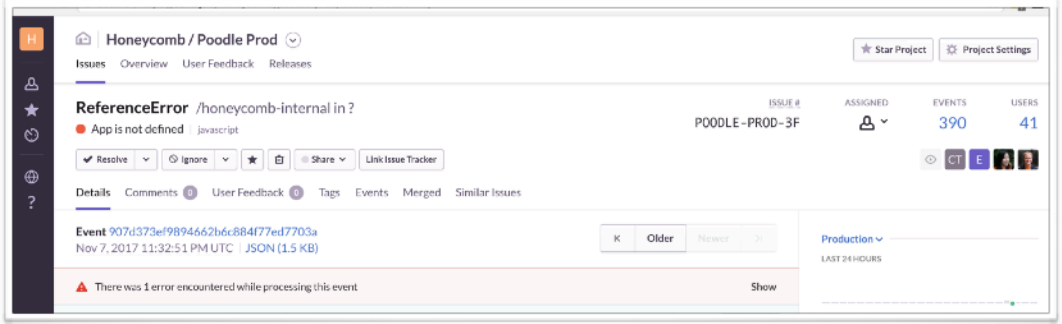
I. Performance

RAIL model

Loading metrics: page load time, resource load time, first paint.

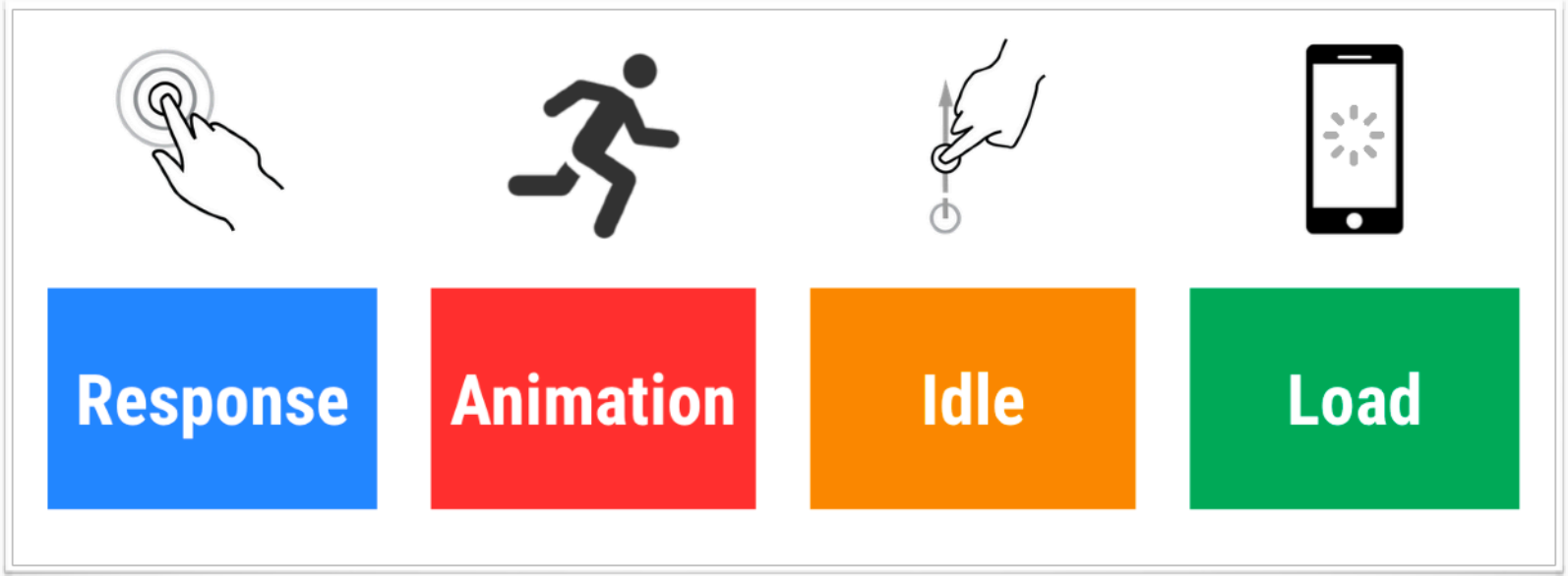
II. Errors

An event per client-side javascript error, with metadata like stack trace & event breadcrumb trail



RAIL performance model

https://developers.google.com/web/fundamentals/performance/rail



Instrumentation toolkit

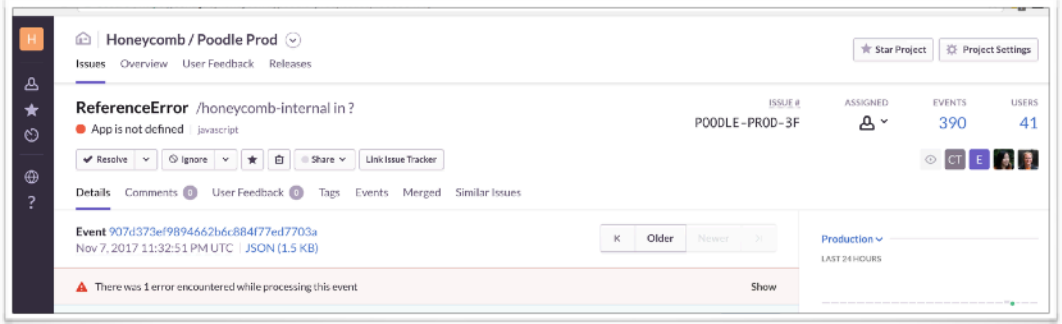
I. Performance

RAIL model

Loading metrics: page load time, resource load time, first paint.

II. Errors

An event per client-side javascript error, with metadata like stack trace & event breadcrumb trail



Instrumentation *code*

I. Performance

Write a custom thing

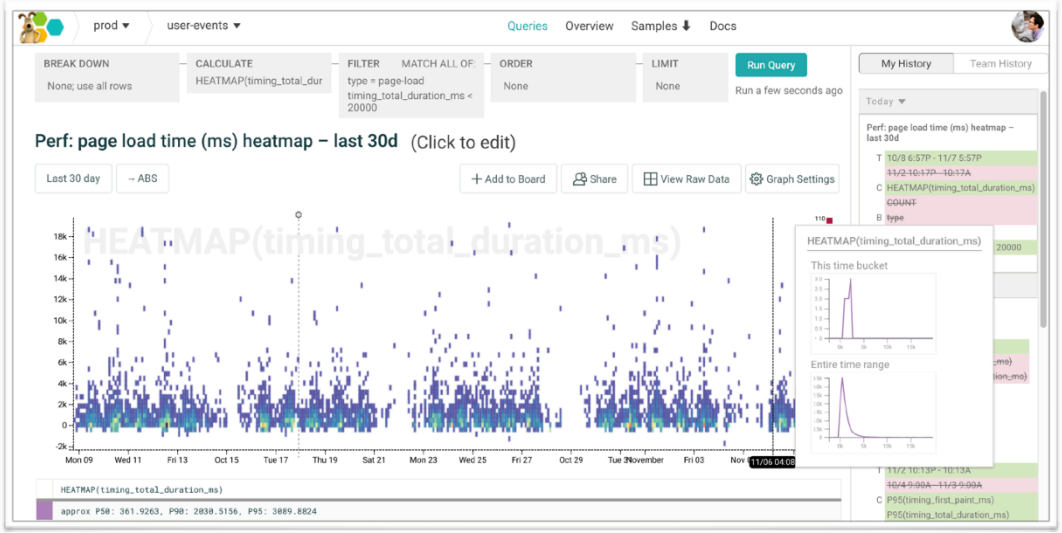
But actually, use Boomerang

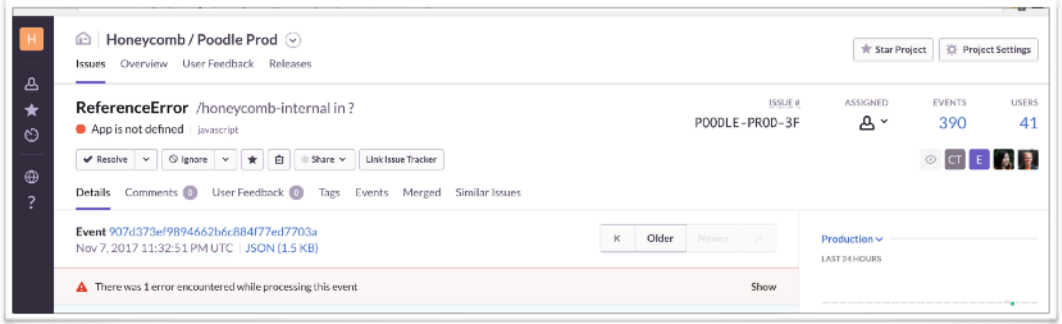
II. Errors

Sentry's Raven JS is o/s

So is Bugsnag's

… or write a custom thing (no)





When we fire events

1. **On page load**
2. **On SPA navigation**
3. **On significant user actions**
4. **On page unload**

Sample page load event

```
{
  // App-specific
  type: “page-navigation",
  page_type: “/:team/datasets/:dataset",
  user_id: 123,
  ab_groups:{ touch_ui: true, multi_team_chat: false }

  // Performance / Environment
  page_load_time_ms: 2145 // plus all navigation timing metrics
  resource_count: 21
  asset_version: "1.232.90"
  canary: false
  request_id: 123456,

  // Capabilities
  user_agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) SomeBrowser/123.45"
  window_height: 822,
  window_width: 1145,
  screen_height: 800,
  screen_width: 1245,
  feature_support_emoji: true,
  feature_support_service_worker: false,
}
```

More browser context

- Installed fonts
- Screen dimensions & color depth
- Browser language
- Online/offline status
- Page visibility (backgrounded?)
- Connection type
- Support for emerging browser APIs
- Geographical location (using a library*)

Sample SPA navigation event

```
{
  // Usage
  type: "react-router-navigation",
  page_type: "/:team/datasets/:dataset”,
  user_id: 123,
  ab_group_touch_ui: true,
  ab_group_multi_team_chat: false,
  request_id: 123456,

  // Performance / Regression
  api_request_duration_ms: 2145,
  api_response_parse_duration_ms 12,
  component_render_duration_ms: 42,
}
```

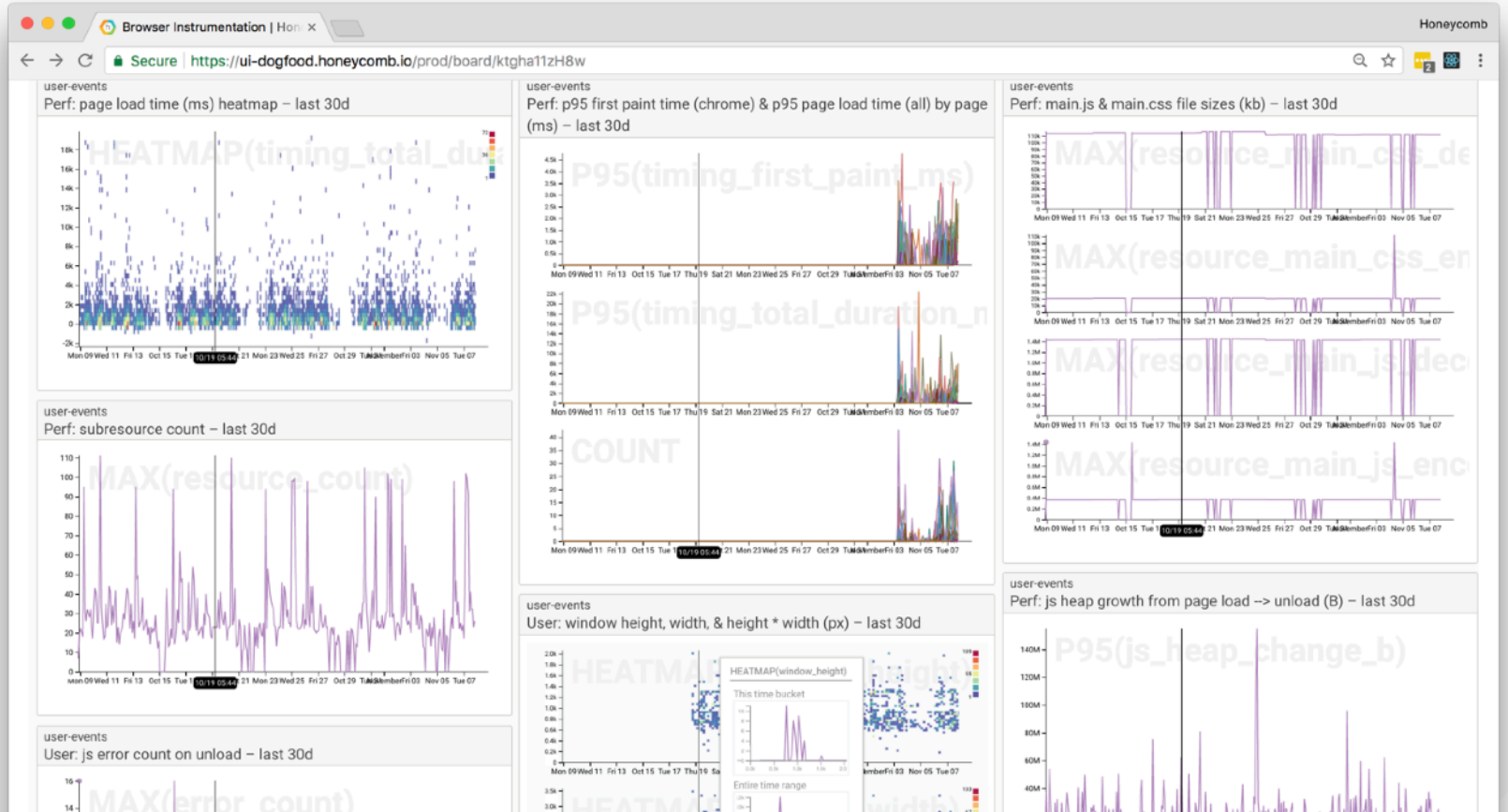
Sample user action event

```
{
  // Usage
  type: "user-derived-column-add",
  page_type: "/:team/datasets/:dataset",
  user_id: 123,
  ab_groups: { touch_ui: true, multi_team_chat: false, }
  request_id: 123456,

  feature_column_type: "number",
}
```

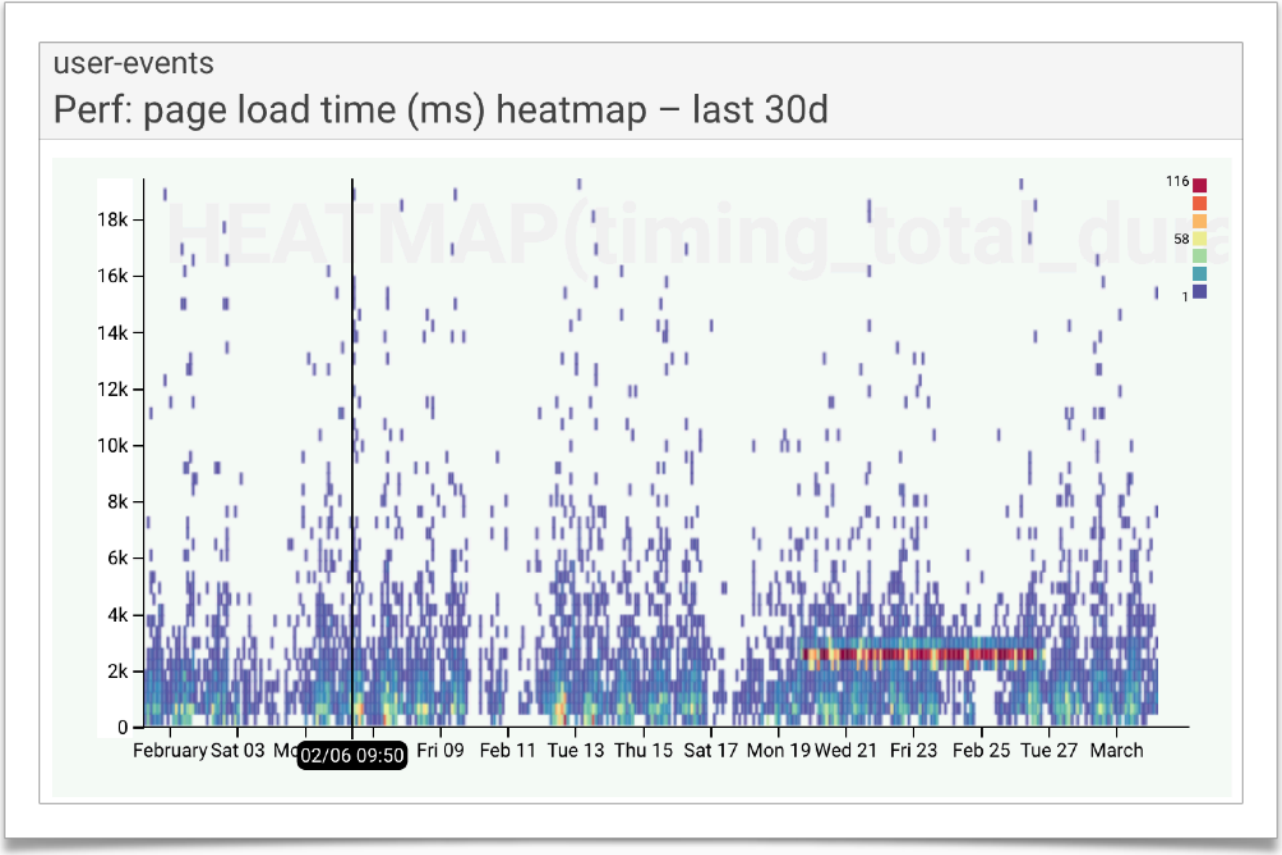
Sample page unload event

```
{
  // Usage
  type: "react-router-navigation",
  page_type: “/:team/datasets/:dataset",
  user_id: 123,
  ab_group_touch_ui: true,
  ab_group_multi_team_chat: false,

  // Performance / Regression
  request_id: 123456,
  js_error_count: 0,
  window_open_duration_s: 45003,

  // Memory info (Chrome) – also send this on load so we can compare heap size
  // and understand how much memory we're using as the user interacts with the page.
  js_heap_size_used_b: 123455,
  js_heap_change_b: 20000,
}
```

Honeycomb Query Sandbox: what we graph

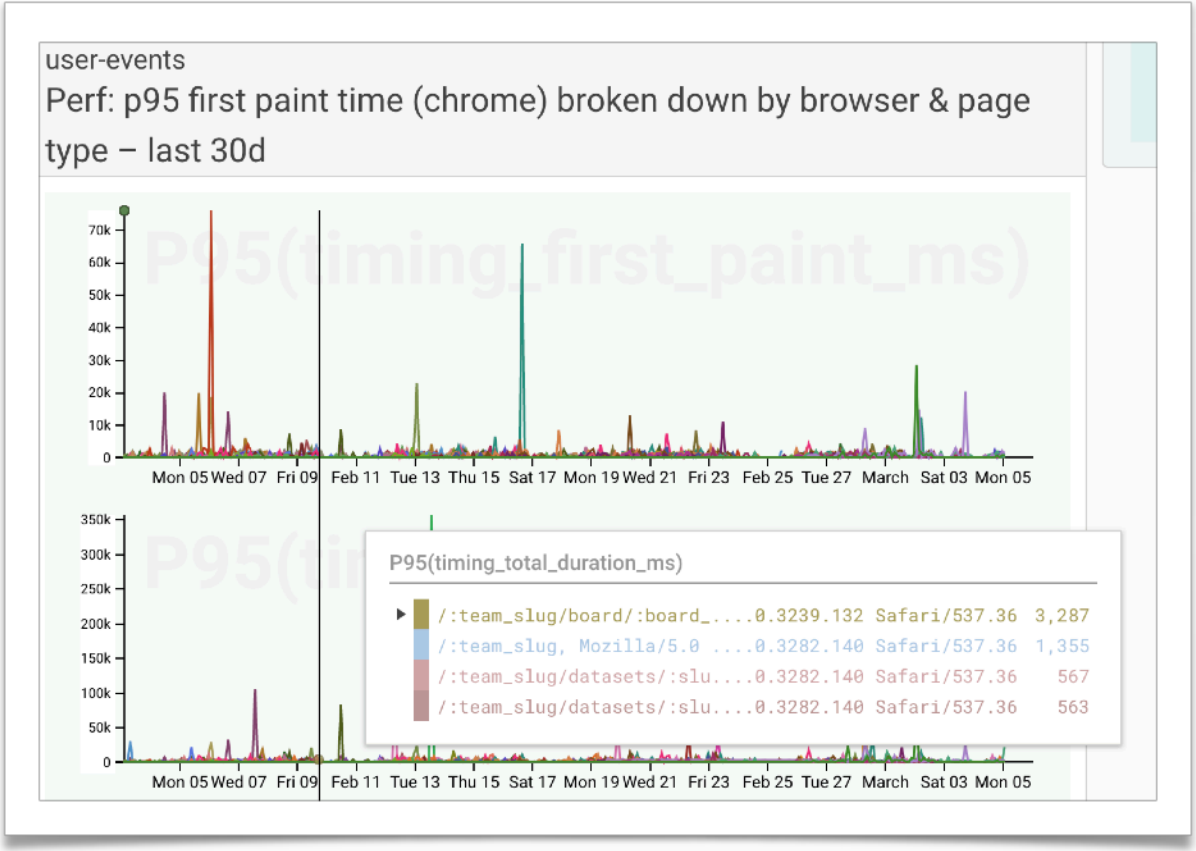


Honeycomb Query Sandbox: what we graph



Honeycomb Query Sandbox: what we graph

Honeycomb Query Sandbox: what we graph



Tool Choice

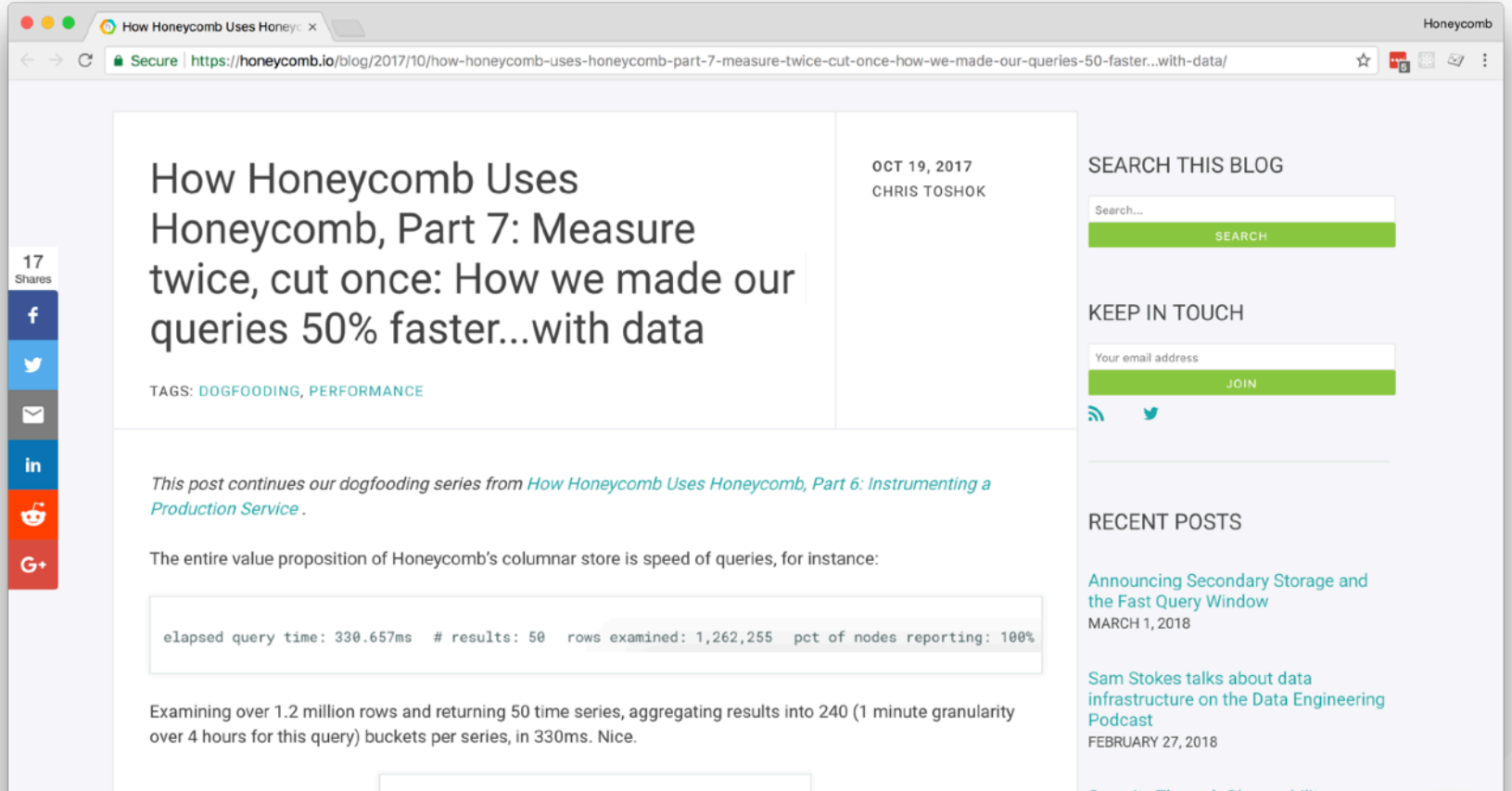
Tool choice: where to send events?

Places to send events (if you don't use Honeycomb)

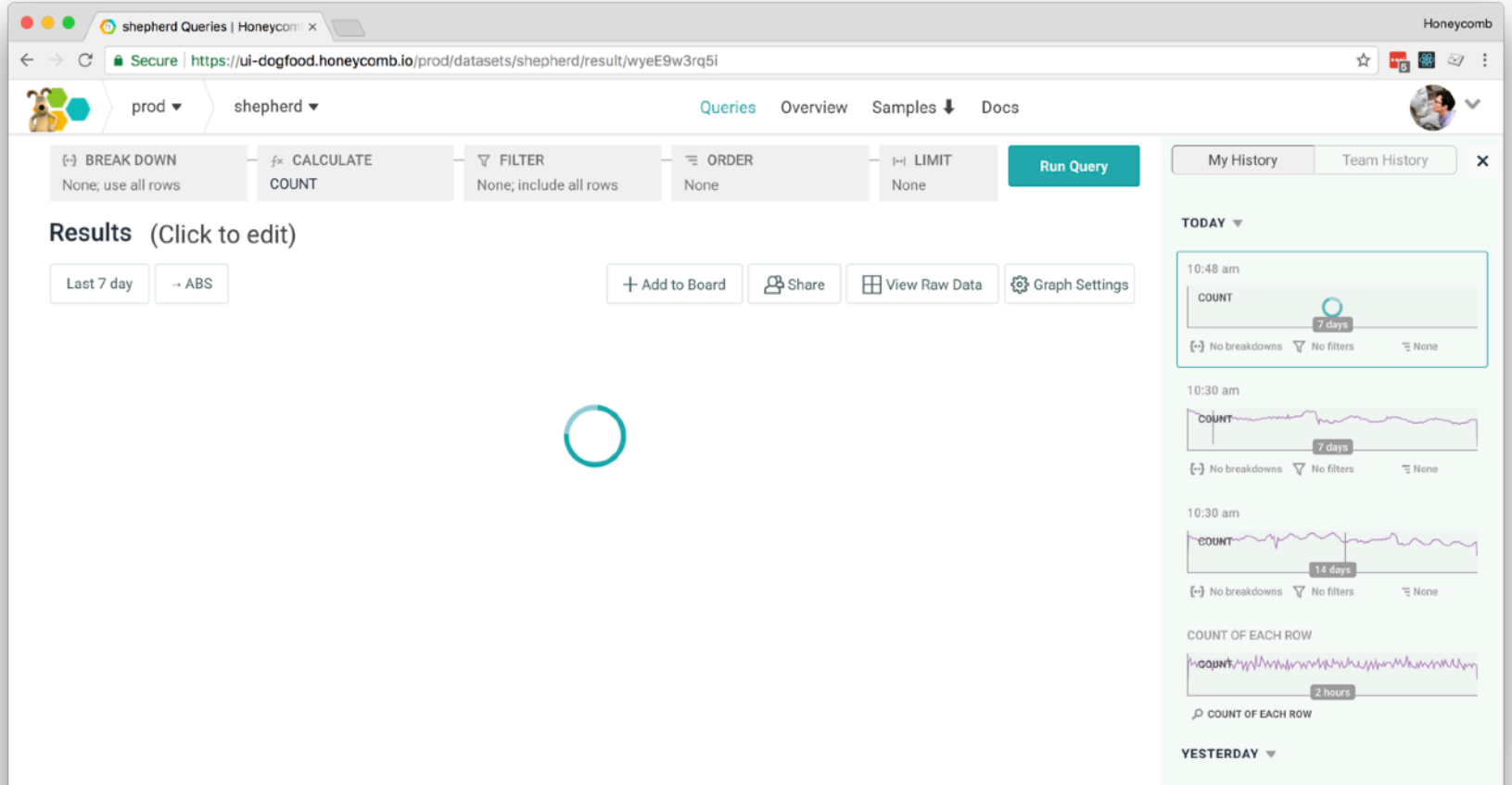
1. Log aggregator
2. Metrics tools with support for high cardinality labels/tags
3. Error monitoring tool (maybe)

Debugging performance

Understanding Normal



Understanding Normal



Understanding Normal

P10(query_and_persist_dur_ms)	P25(query_and_persist_dur_ms)	P50(query_and_persist_dur_ms)	P75(query_and_persist_dur_ms)	P90(query_and_persist_dur_ms)
108.475	198.75	411.4375	1,347	4,060.4

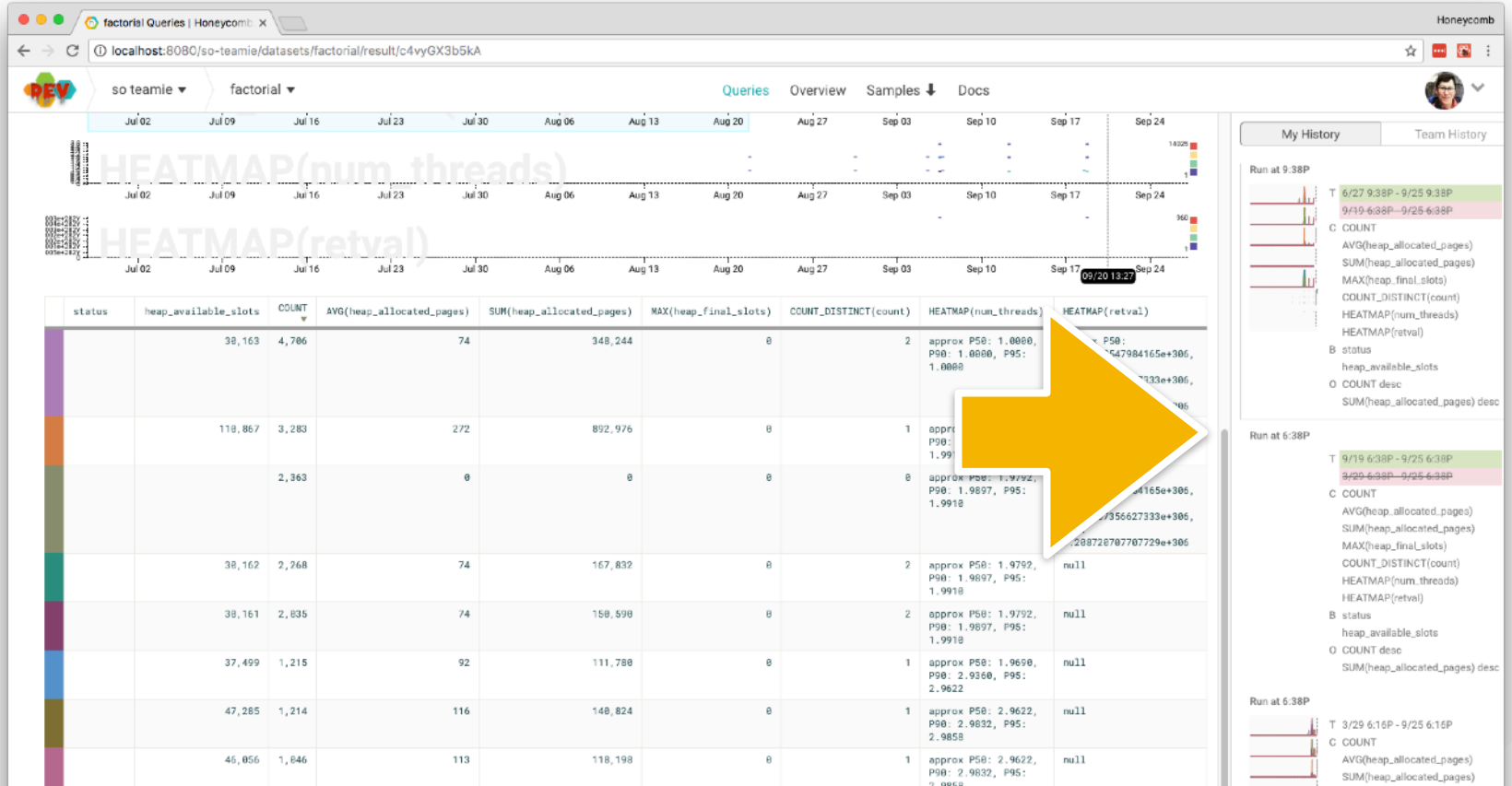
Understanding Normal

poll_iterations_required	COUNT ▼
1	4,840
2	155
3	109
4	55

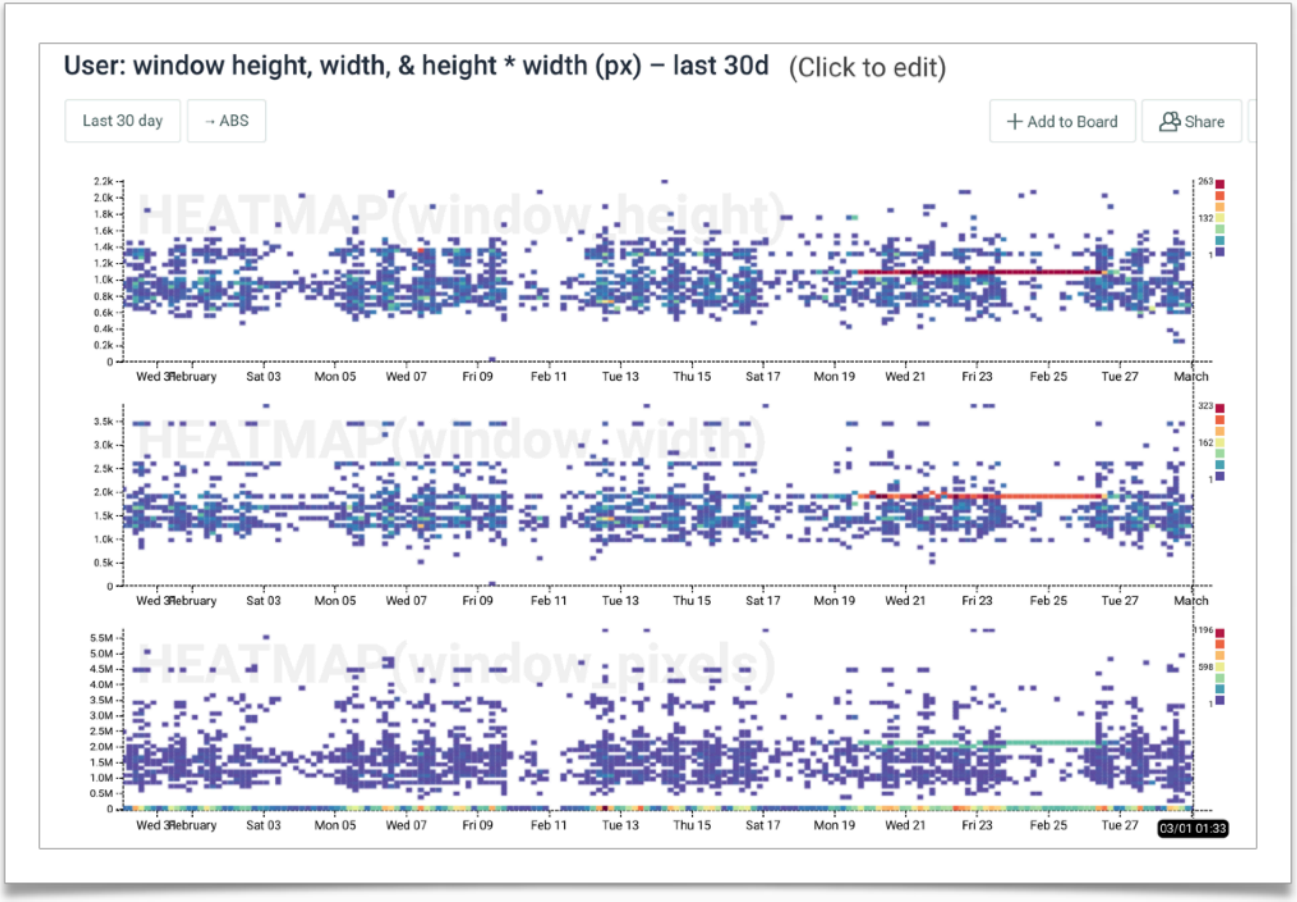
Looking ahead

Using observability to drive design

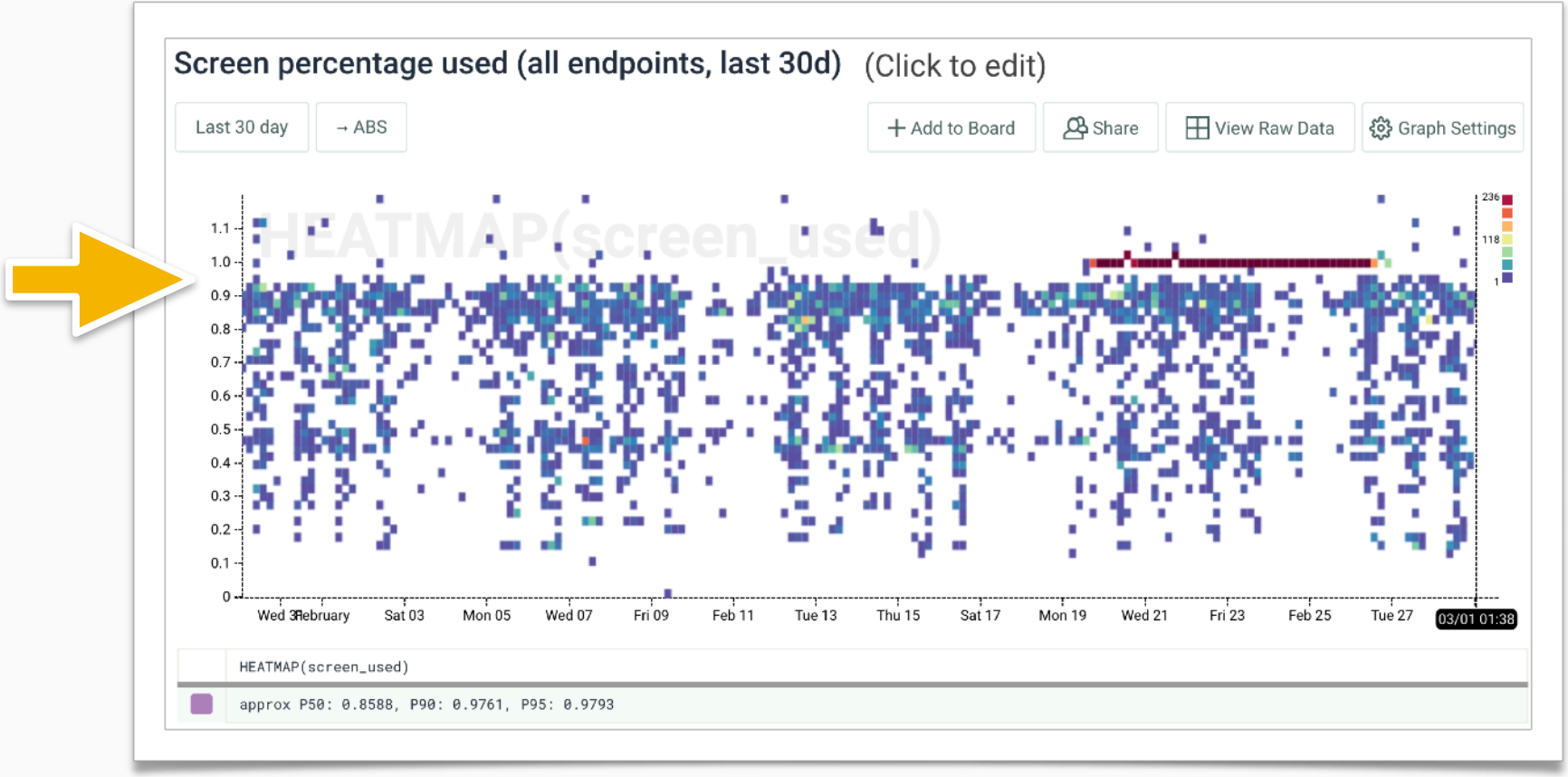
Using observability to drive design



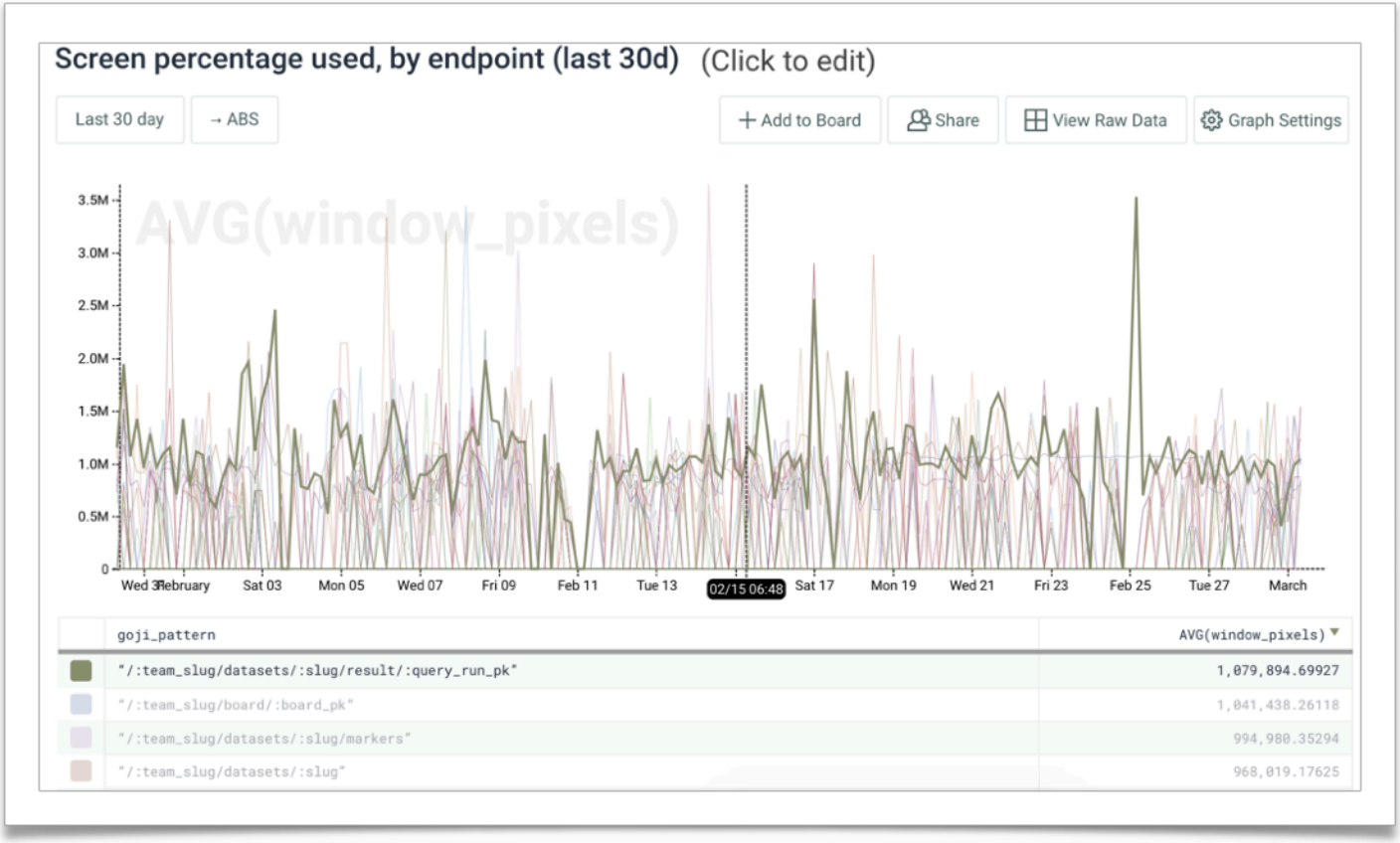
Using observability to drive design



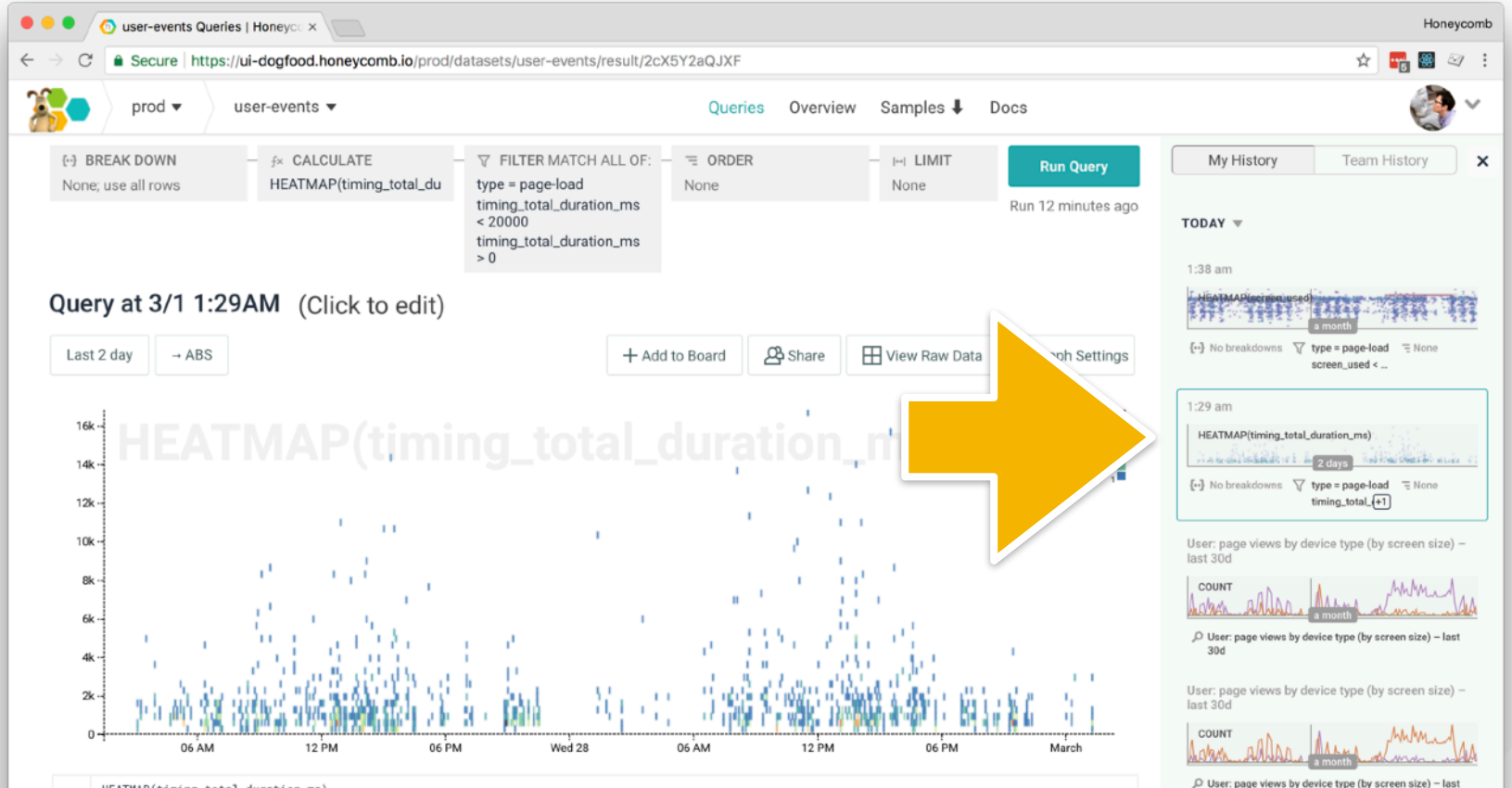
Using observability to drive design



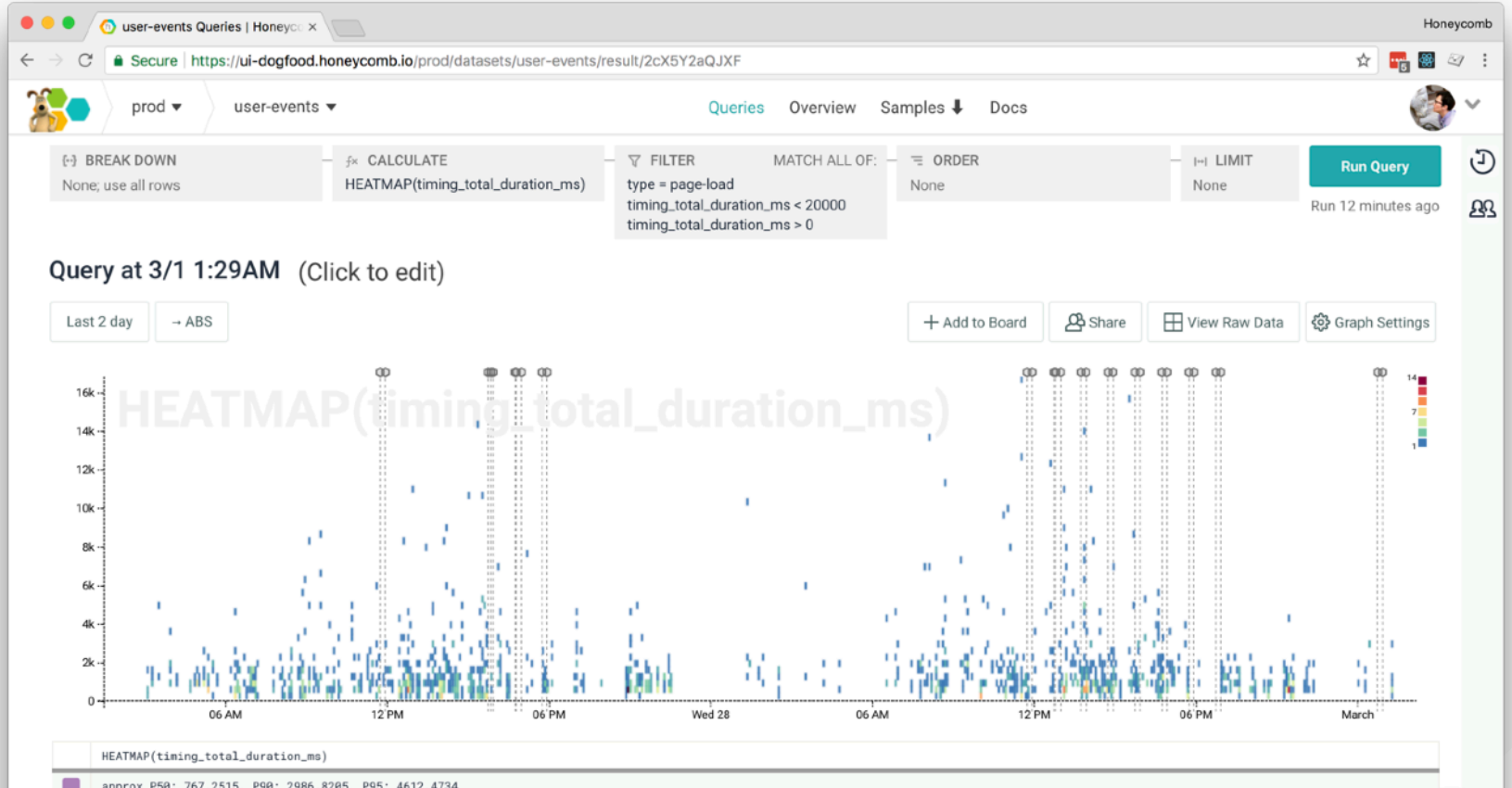
Using observability to drive design



Using observability to drive design



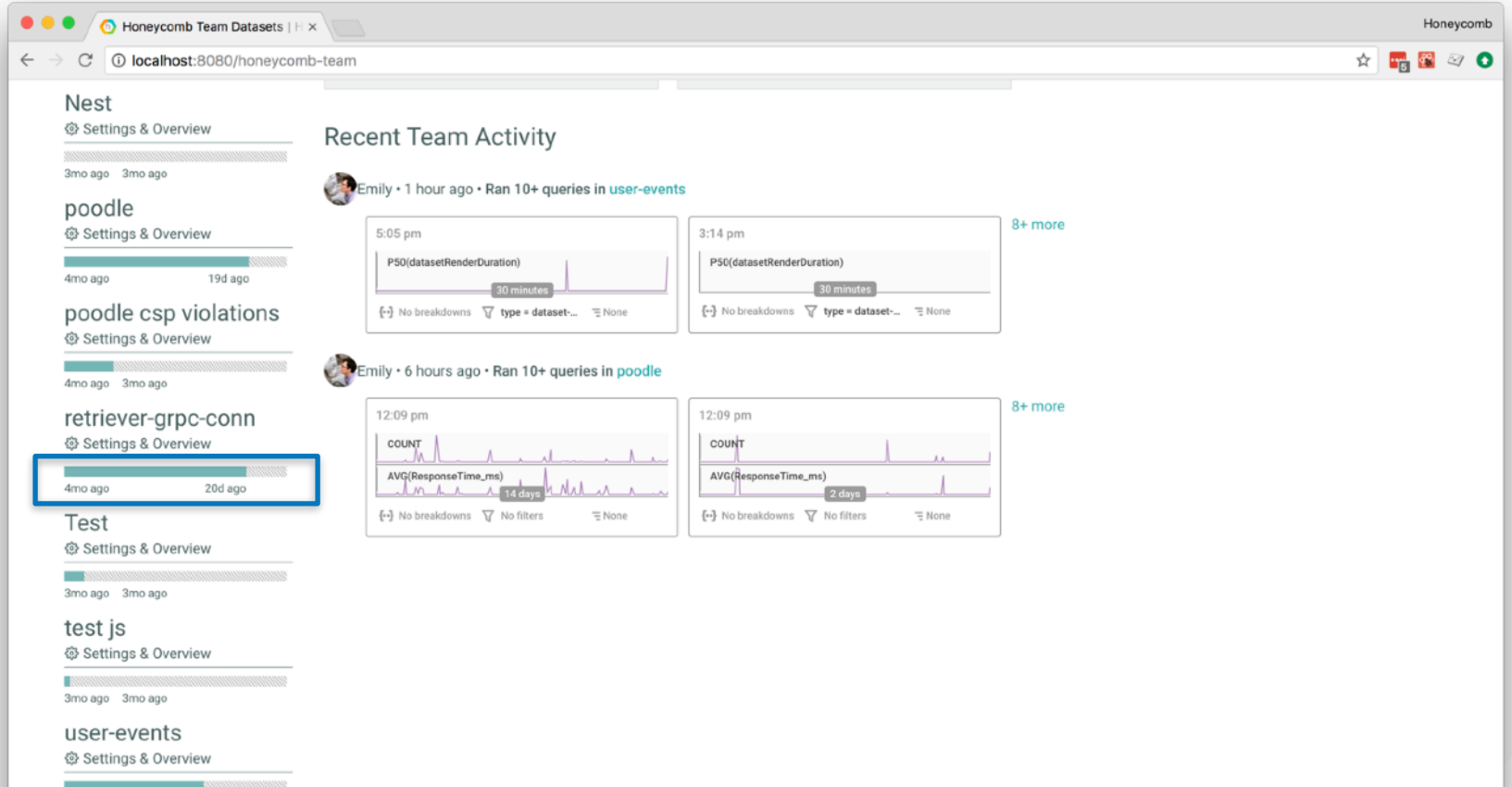
Using observability to drive design



Understanding Normal

Server-side rendering vs. client side rendering

Understanding normal



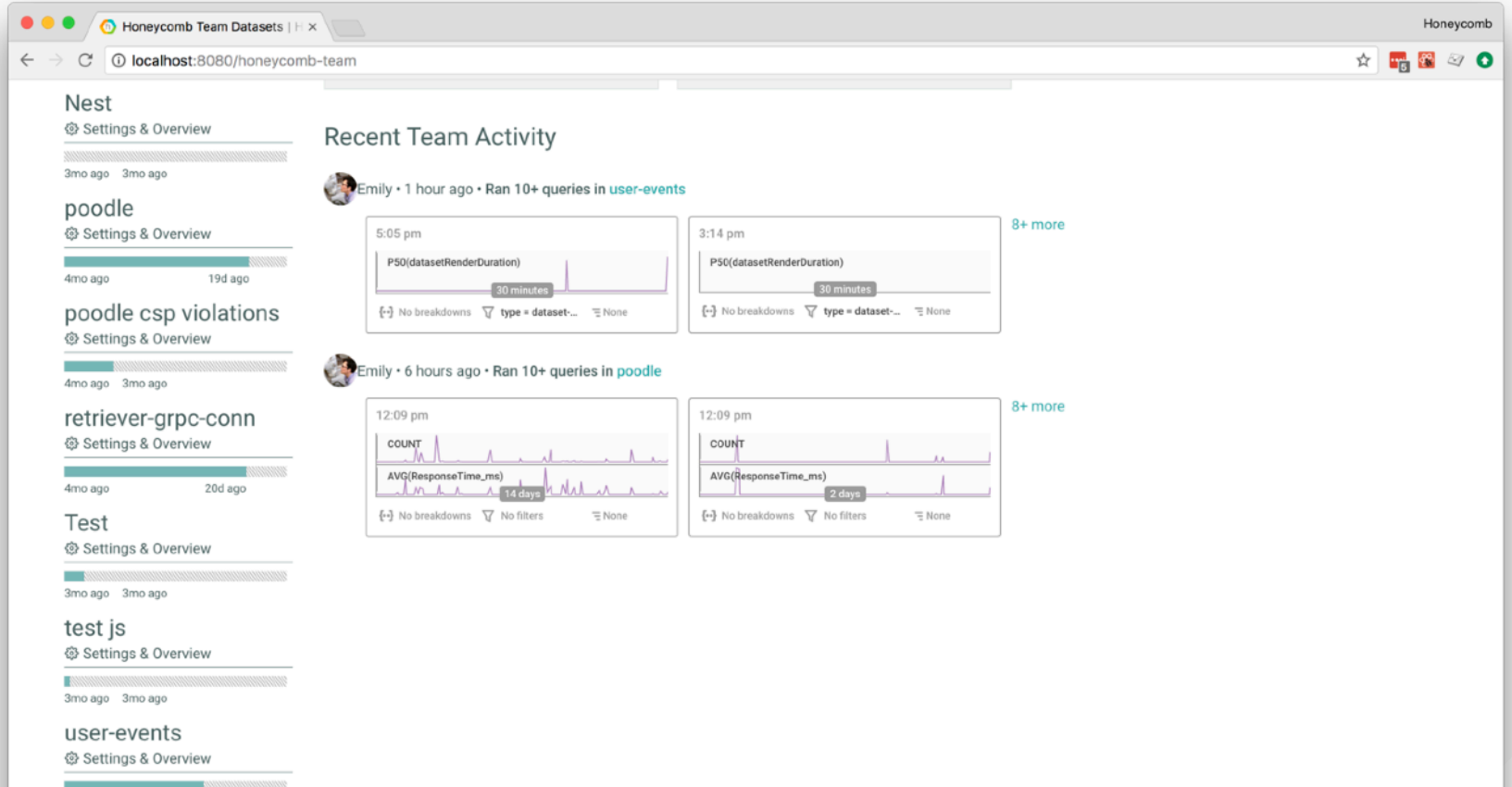
Page load event with server timings

```
{
  type: “page-navigation",
  page_type: “/:team/datasets/:dataset",

  // Performance / Environment
  page_load_time_ms: 2145 // plus all navigation timing metrics
  resource_count: 21
  asset_version: "1.232.90"
  canary: false

  // Already have this from window.performance navigation
  server_request_dur_ms: performance.timing.responseEnd - performance.timing.navigationStart,

  // New timings
  template_server_render_dur_ms: 12,
  time_to_component_rendered_ms: { dataset_usage_viz: 156 },

  request_id: 123456,
}
```

```
// capture time-to-component-rendered with react

class DatasetUsageViz extends React.Component {

    componentDidMount() {
        // updatePageLoadEventPayload will merge this payload with
        // our global event context so these fields appear on the
        // page-load or page-unload event.
        updatePageLoadEventPayload({
          time_to_component_rendered_ms: {
              dataset_usage_viz: window.performance.now()
           },
        });
    };

    // ...
}
```

```
// capture time-to-component-rendered with MutationObserver
// only run if there is feature support (older clients won't send data)
if (window.MutationObserver) {
    const container = document.guerySelector('dataset-container');

    const observer = new MutationObserver(function(mutations) {
      mutations.forEach(function(mutation) {

        if ( /* check if mutated dom is in expected "done" state */ ) {
        // updatePageLoadEventPayload will merge this payload with
        // our global event context so these fields appear on the
        // page-load or page-unload event.
        updatePageLoadEventPayload({
          time_to_component_rendered_ms: {
              dataset_usage_viz: window.performance.now()
           },
        });
      });
    });

    // Start observing! Pass in the target node as well as the observer config
    observer.observe(container, { childList: true });
}
```

Understanding normal



Privacy

Privacy: no new data, but new problems

1. Pipe events through your own infrastructure
2. Consider whether a field will truly be helpful for debugging
    - Is there a less-revealing field value that would be as helpful?
3. Check the data retention policies & practices of your SaaS products

High-performance instrumentation

Send events in a non-blocking way

High-performance browser instrumentation code:

1. Use the Beacon API to send events in a non-blocking way
2. Use `requestIdleCallback` or `setTimeout` to schedule slow calculations
3. Wrap rendering timers in a `requestAnimationFrame`

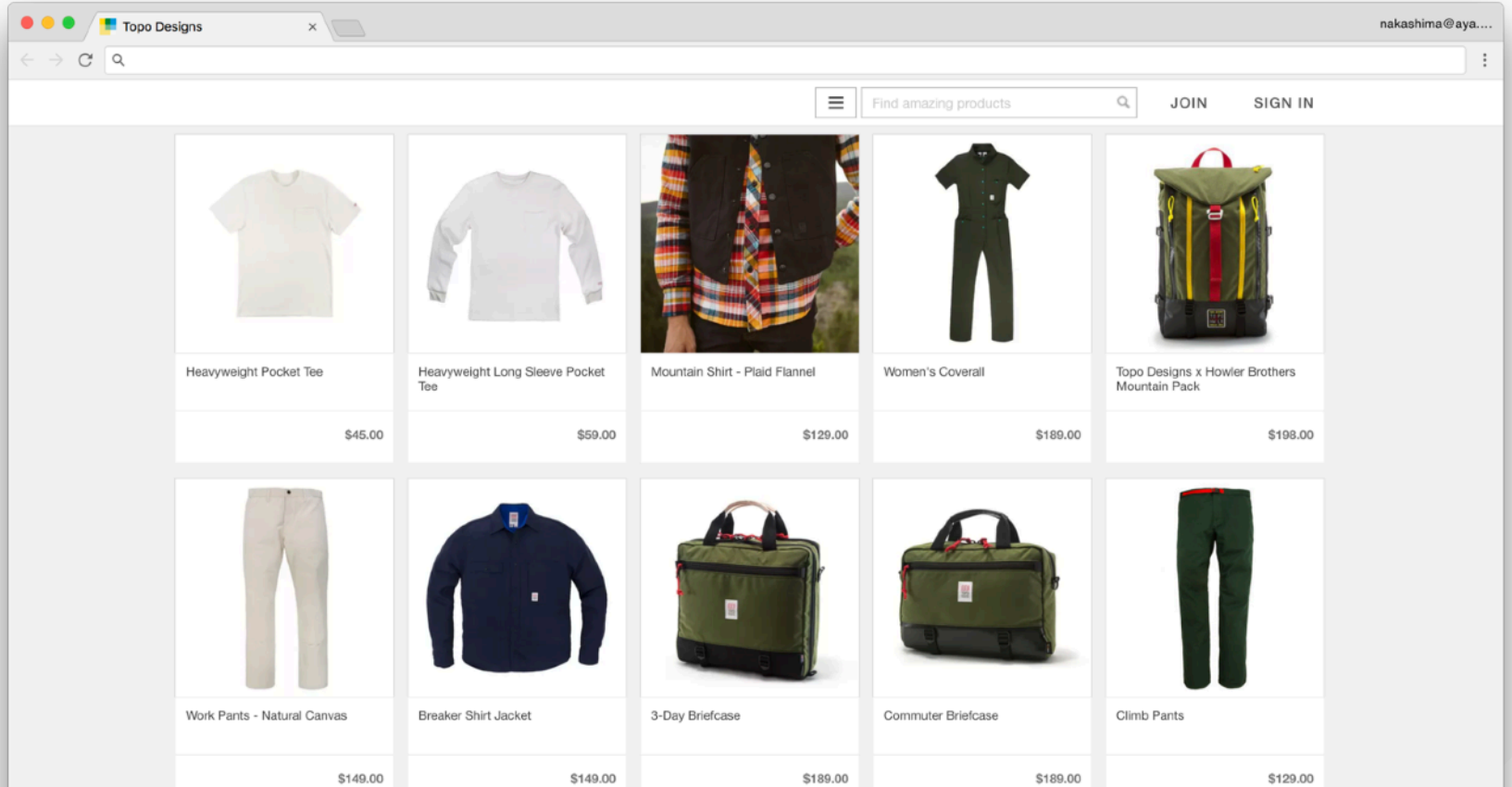
Sampling

Select & send representative events instead of sending *all* events.

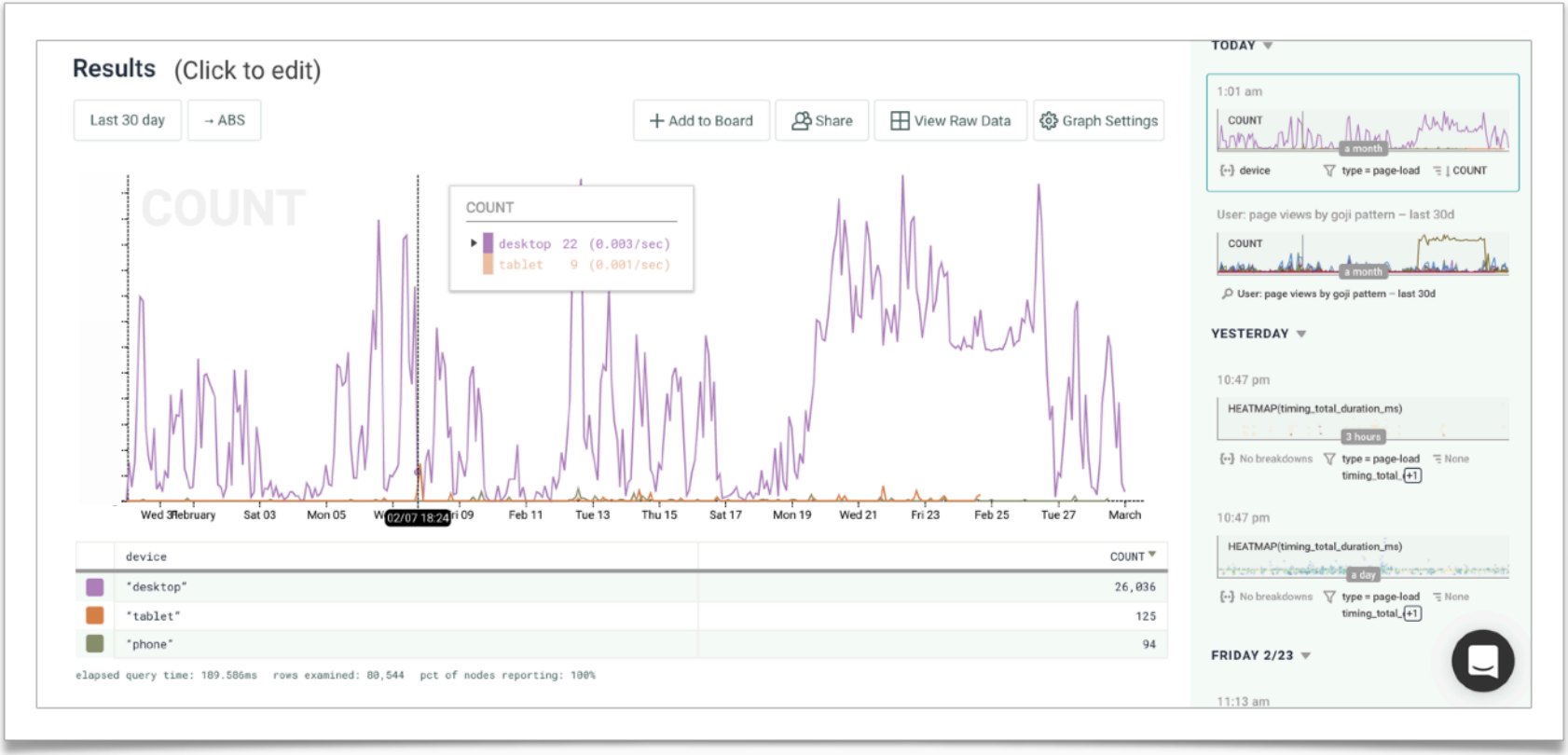
1. Watch Ben's talk from LISA17: Sample Your Traffic but Keep the Good Stuff
2. Can be either client-side or server-side
3. Use the dynsampler-js library or write your own code
4. Be sure to set the `sample_rate` field on your events

Epilogue

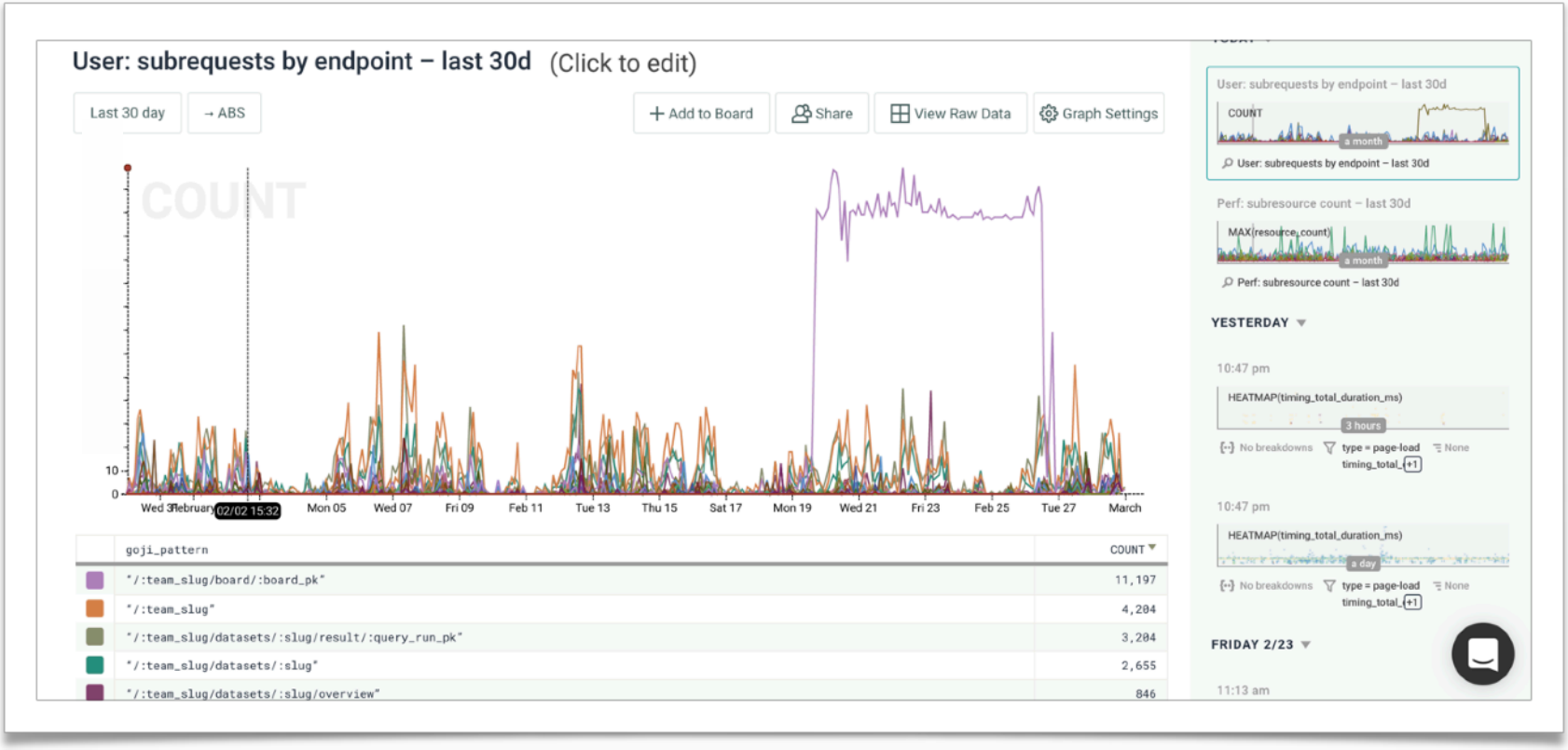
Epilogue


Topo Designs
nakashima@aya....
Find amazing products
JOIN
SIGN IN
Heavyweight Pocket Tee
$45.00
Heavyweight Long Sleeve Pocket Tee
$59.00
Mountain Shirt - Plaid Flannel
$129.00
Women's Coverall
$189.00
Topo Designs x Howler Brothers Mountain Pack
$198.00
Work Pants - Natural Canvas
$149.00
Breaker Shirt Jacket
$149.00
3-Day Briefcase
$189.00
Commuter Briefcase
$189.00
Climb Pants
$129.00

Epilogue


Results (Click to edit)
Last 30 day
→ ABS
+ Add to Board
Share
View Raw Data
Graph Settings
COUNT
COUNT
desktop 22 (0.003/sec)
tablet 9 (0.001/sec)
Wed 31 February
Sat 03
Mon 05
02/07 18:24
Feb 11
Tue 13
Thu 15
Sat 17
Mon 19
Wed 21
Fri 23
Feb 25
Tue 27
March
device
COUNT
"desktop" 26,036
"tablet" 125
"phone" 94
elapsed query time: 189.586ms rows examined: 80,544 pct of nodes reporting: 100%
TODAY
1:01 am
COUNT
a month
device
type = page-load
COUNT
User: page views by goji pattern – last 30d
COUNT
a month
User: page views by goji pattern – last 30d
YESTERDAY
10:47 pm
HEATMAP(timing_total_duration_ms)
3 hours
No breakdowns
type = page-load
timing_total_(+1)
None
10:47 pm
HEATMAP(timing_total_duration_ms)
a day
No breakdowns
type = page-load
timing_total_(+1)
None
FRIDAY 2/23
11:13 am

Epilogue


User: subrequests by endpoint – last 30d (Click to edit)
Last 30 day
→ ABS
+ Add to Board
Share
View Raw Data
Graph Settings
COUNT
10
0
Wed 31
February
02/02 15:32
Mon 05
Wed 07
Fri 09
Feb 11
Tue 13
Thu 15
Sat 17
Mon 19
Wed 21
Fri 23
Feb 25
Tue 27
March
goji_pattern
COUNT
"/:team_slug/board/:board_pk" 11,197
"/:team_slug" 4,284
"/:team_slug/datasets/:slug/result/:query_run_pk" 3,284
"/:team_slug/datasets/:slug" 2,655
"/:team_slug/datasets/:slug/overview" 846
User: subrequests by endpoint – last 30d
COUNT
a month
User: subrequests by endpoint – last 30d
Perf: subresource count – last 30d
MAX(resource_count)
a month
Perf: subresource count – last 30d
YESTERDAY
10:47 pm
HEATMAP(timing_total_duration_ms)
3 hours
No breakdowns
type = page-load
timing_total_[+1]
None
10:47 pm
HEATMAP(timing_total_duration_ms)
a day
No breakdowns
type = page-load
timing_total_[+1]
None
FRIDAY 2/23
11:13 am

Thank you!

@eanakashima

honeycomb.io