Observability for Software Teams

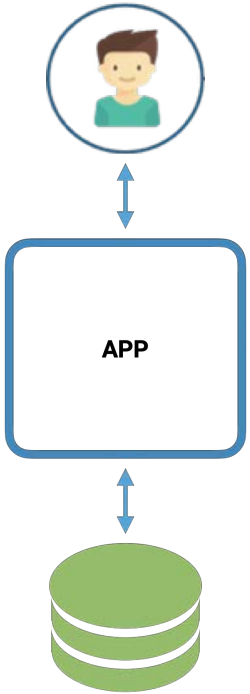
Shelby Spees
Developer Advocate @ Honeycomb.io

Illustrations by @emilywithcurls!

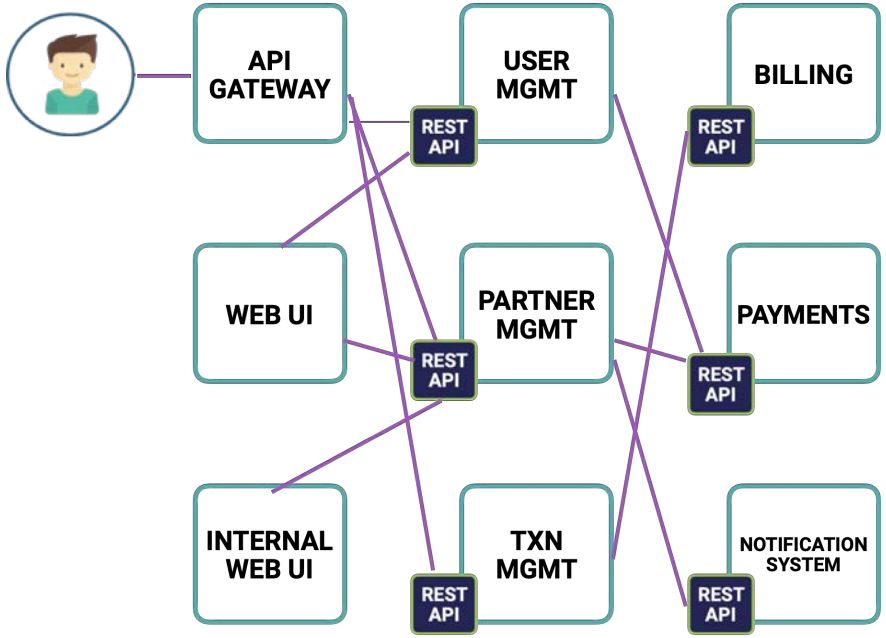
honeycomb.io

Production is increasingly complex

© 2021 Hound Technology, Inc. All Rights Reserved.



THEN



NOW

© 2021 Hound Technology, Inc. All Rights Reserved.

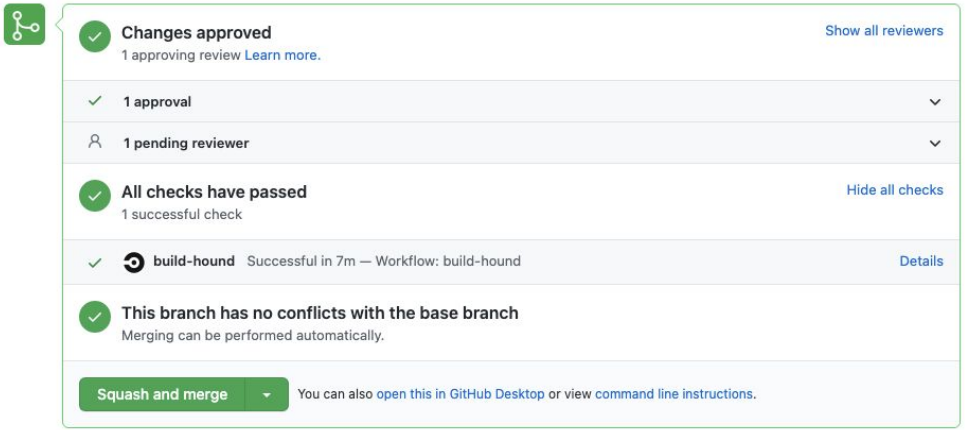
Software teams

DevOps is more than “Dev” and “Ops”:

- application engineers
- platform engineers
- infrastructure engineers
- SREs
- SDETs
- support engineers
- and more!

© 2021 Hound Technology, Inc. All Rights Reserved.

We've come a long way



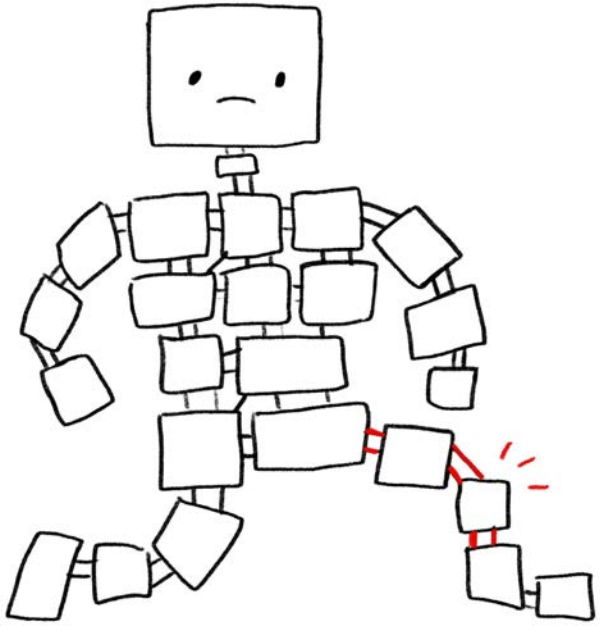

```
# and private subnet for that list index,
# then create an EIP and attach a nat_gateway for each one.  and an aws route
# table should be created for each private subnet, and add the correct nat_gw

resource "aws_subnet" "private" {
  vpc_id            = aws_vpc.mod.id
  cidr_block        = var.private_ranges[count.index]
  availability_zone = var.azs[count.index]
  count             = length(var.private_ranges)
  tags = {
    Name = "${var.env}_private_${count.index}"
  }
}

resource "aws_subnet" "public" {
  vpc_id            = aws_vpc.mod.id
  cidr_block        = var.public_ranges[count.index]
  availability_zone = var.azs[count.index]
  count             = length(var.public_ranges)
  tags = {
    Name = "${var.env}_public_${count.index}"
  }
  map_public_ip_on_launch = true
}

# refactor to take all the route {} sections out of routing tables,
# and turn them into associated aws_route resources
# so we can add vpc peering routes from specific environments.
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.mod.id
  tags = {
    Name = "${var.env}_public_subnet_route_table"
```

© 2021 Hound Technology, Inc. All Rights Reserved.

But we still encounter problems

emergent failure modes
small failures cascading together to degrade or take down a system

see also: [how.complexsystems.fail]



© 2021 Hound Technology, Inc. All Rights Reserved.

Getting paged can be scary

© 2021 Hound Technology, Inc. All Rights Reserved.

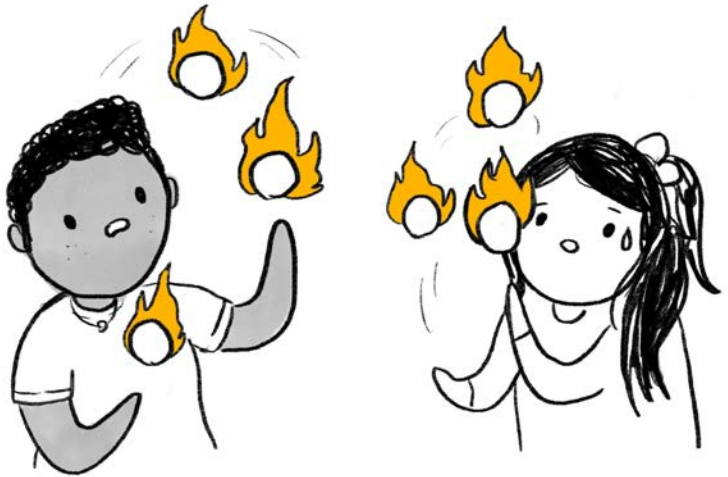
Our tools don’t answer our questions

© 2021 Hound Technology, Inc. All Rights Reserved.

We don’t make forward progress

42%

of developer time is spent on dealing with bad code and tech debt



Source: [The Developer Coefficient, Stripe, 2018]

© 2021 Hound Technology, Inc. All Rights Reserved.

It doesn’t have to be this way

We have the technology!

How can we better interact with production?

Our applications have all the answers at runtime!

We want to

- capture those answers
- make that data available to interrogate

Big companies have done this

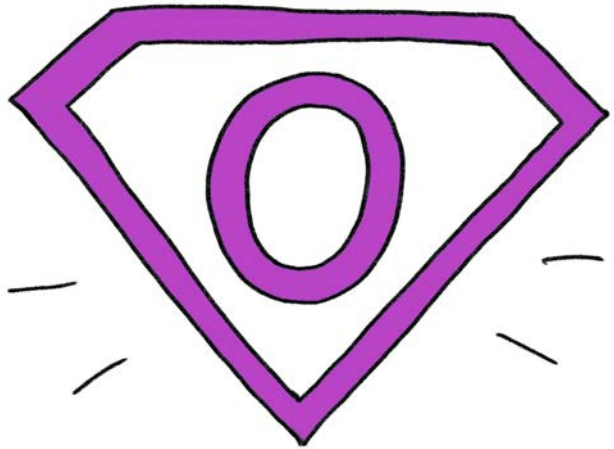
- at Facebook: Scuba
- at Google: Dapper

© 2021 Hound Technology, Inc. All Rights Reserved.

We need observability!

What is observability?

in software, the ability to understand and explain any state a system can get into, no matter how novel or bizarre, without deploying new code

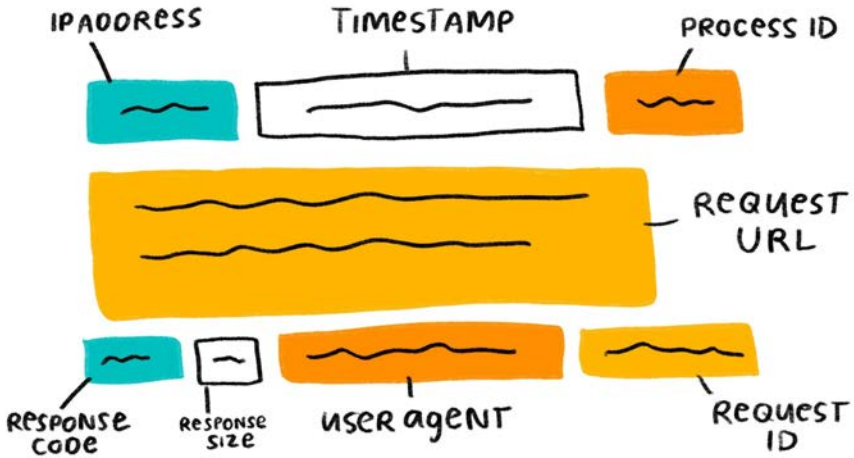


© 2021 Hound Technology, Inc. All Rights Reserved.

How to gain observability?

Observability requires

- capturing telemetry data with lots of runtime context
- interacting with that telemetry in near-real time



© 2021 Hound Technology, Inc. All Rights Reserved.

Instrumentation generates telemetry

instrumentation: code or tooling that captures data about the state of your running system

telemetry: data generated by your system that documents its state

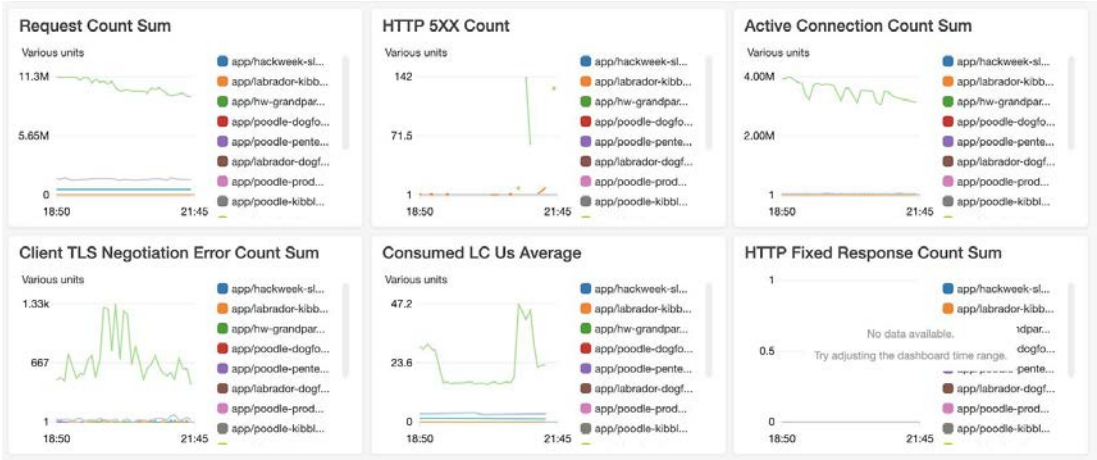

```
{
  "timestamp": "2018-03-20T00:47:25.339Z",
  "app.interesting_thing": "banana",
  "duration_ms": 772.446625,
  "handler.name": "main.hello",
  "handler.pattern": "/hello/",
  "handler.type": "http.HandlerFunc",
  "meta.beeline_version": "0.2.0",
  "meta.local_hostname": "cobbler.local",
  "meta.span_type": "root",
  "meta.type": "http_request",
  "name": "main.hello",
  "request.content_length": 0,
  "request.header.user_agent": "curl/7.54.0",
  "request.host": "localhost:8080",
  "request.http_version": "HTTP/1.1",
  "request.method": "GET",
  "request.path": "/hello/",
  "request.remote_addr": "127.0.0.1:60379",
  "response.status_code": 200,
  "service_name": "sample app",
  "trace.span_id": "9e4fe697-3ea9-48c9-b673-72d7ddf118a6",
  "trace.trace_id": "b64c89a9-7671-4732-bef1-9ef75ab831f6"
}
```


© 2021 Hound Technology, Inc. All Rights Reserved.

You're already collecting data

You've probably instrumented your system to generate telemetry like

- metrics
- logs
- traces



That's great!

© 2021 Hound Technology, Inc. All Rights Reserved.

But traditional approaches aren't enough

© 2021 Hound Technology, Inc. All Rights Reserved.

Observability is for hard-to-debug problems

Distributed Systems
small change causing downstream effects?

Poor Performance
what part of the app is worth optimizing?

Subset of Traffic
only some users are complaining?

© 2021 Hound Technology, Inc. All Rights Reserved.

Metrics

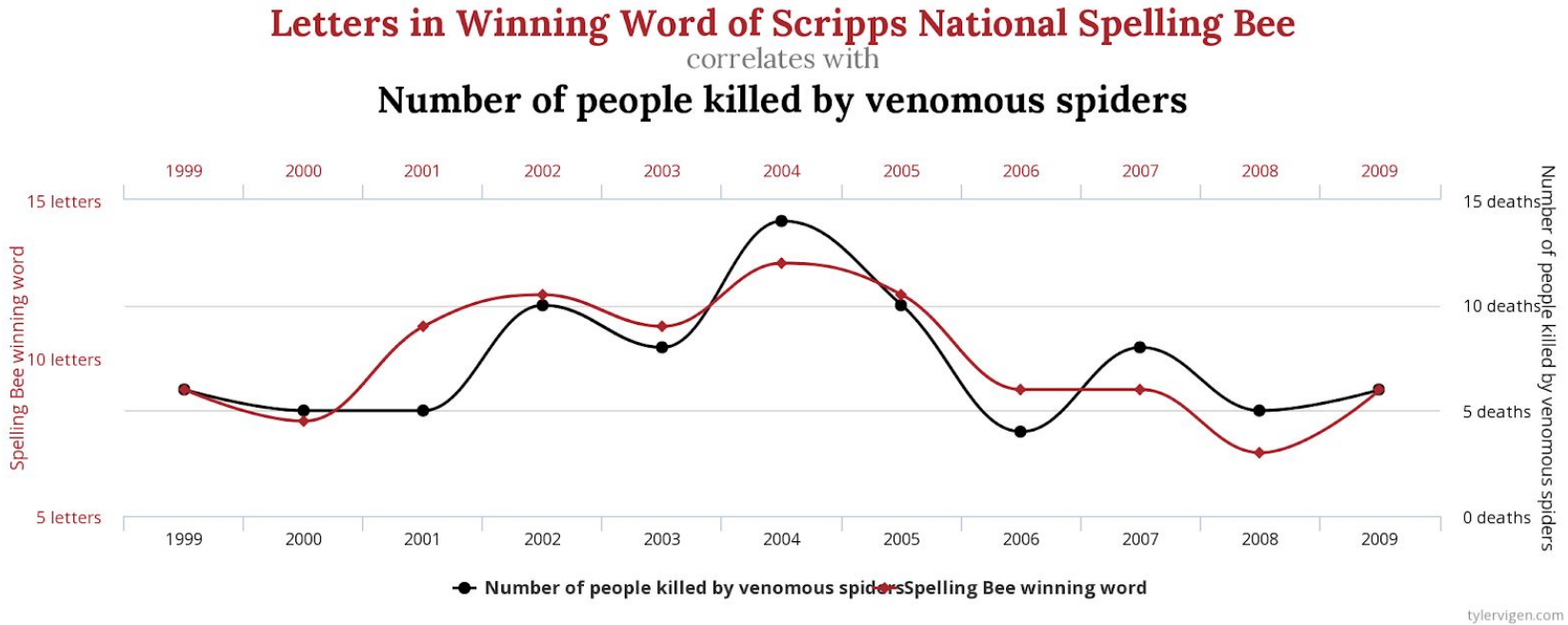
great when thinking about infrastructure!

but they:

- only answer known questions
- don’t capture application context
- don’t support high-cardinality use cases

© 2021 Hound Technology, Inc. All Rights Reserved.

Metrics have little context



https://www.tylervigen.com/spurious-correlations

© 2021 Hound Technology, Inc. All Rights Reserved.

Flat logs

- capturing output from the code itself
- no standard log formatting across libraries or services
- requires string parsing--expensive at scale

Centralized logging services are expensive and slow to query

- great for compliance, not for debugging

© 2021 Hound Technology, Inc. All Rights Reserved.

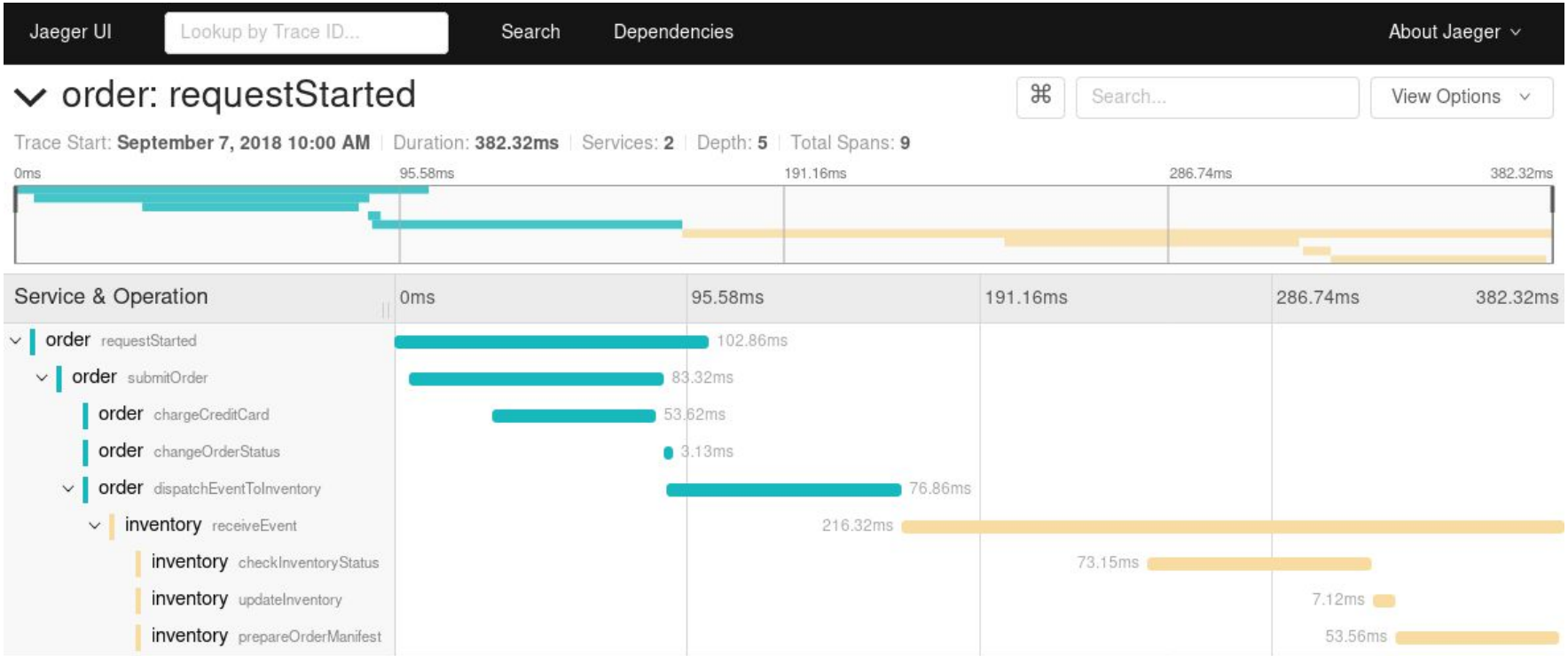
Distributed traces

Visualization of the stack trace across your distributed system

Traces are formed by a directed graph made up of objects called *spans*

- each span has a **start time** and a **duration**, and points to its **parent** span

© 2021 Hound Technology, Inc. All Rights Reserved.

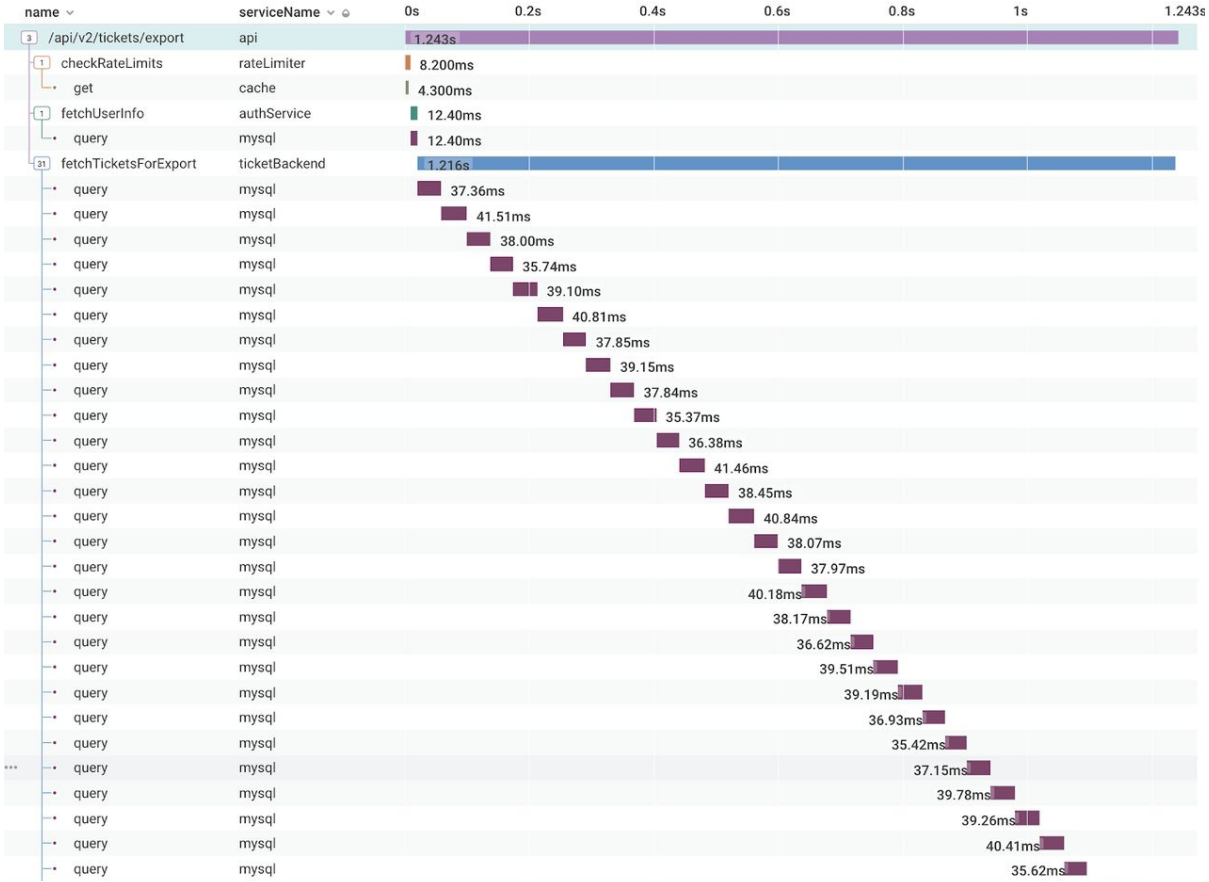


distributed trace visualization in Jaeger UI

© 2021 Hound Technology, Inc. All Rights Reserved.

1.243s API call

What made it slow?



© 2021 Hound Technology, Inc. All Rights Reserved.

Structured events!

What are structured events?

structured logs + benchmarking + (optionally) tracing

```
{
  "Timestamp": "2018-07-03T04:57:12.517022Z",
  "duration_ms": 619.703,
  "request.method": "GET",
  "request.path": "/",
  "request.user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
  "response.status_code": 200,
  "trace.span_id": "3eada0ce-934b-4ffd-bb72-2c9f57b02bf1",
  "trace.parent_id": "aa23-3becd17d726-607d2150c-5397-4e8e",
  "trace.trace_id": "07d2150c-5397-4e8e-aa23-3becd17d7266",
  ...
}
```

© 2021 Hound Technology, Inc. All Rights Reserved.

Keep all your runtime context together

No predefined index or schema
add fields as needed

Key-value pairs
less expensive to parse and query

© 2021 Hound Technology, Inc. All Rights Reserved.

Flat logs vs. structured events

Flat logs get written eagerly

You can’t keep track of state changes, even within a specific context

© 2021 Hound Technology, Inc. All Rights Reserved.

```java
@PostMapping("flip")
public ResponseEntity flipImage(@RequestParam("image") MultipartFile file,
                                @RequestParam(value = "vertical") Boolean vertical,
                                @RequestParam(value = "horizontal") Boolean horizontal) {

    if (file.getContentType() != null) {
        LOGGER.warn("Wrong content type uploaded: {}", file.getContentType());
        return new ResponseEntity<>("Wrong content type uploaded: " + file.getContentType());
    }

    LOGGER.info("Receiving {} image to flip.", file.getContentType());
    byte[] flippedImage = imageService.flip(file, vertical, horizontal);

    if (flippedImage == null) {
        return new ResponseEntity<>("Failed to flip image", HttpStatus.INTERNAL_SERVER_ERROR);
    }

    LOGGER.info("Successfully flipped image id: {}", file.getId());
    return new ResponseEntity<>(flippedImage, headers, HttpStatus.OK);
}
```

write as you go

write as you go

write as you go

© 2021 Hound Technology, Inc. All Rights Reserved.

Flat logs vs. structured events

Structured events store all the data within your context, from beginning to end

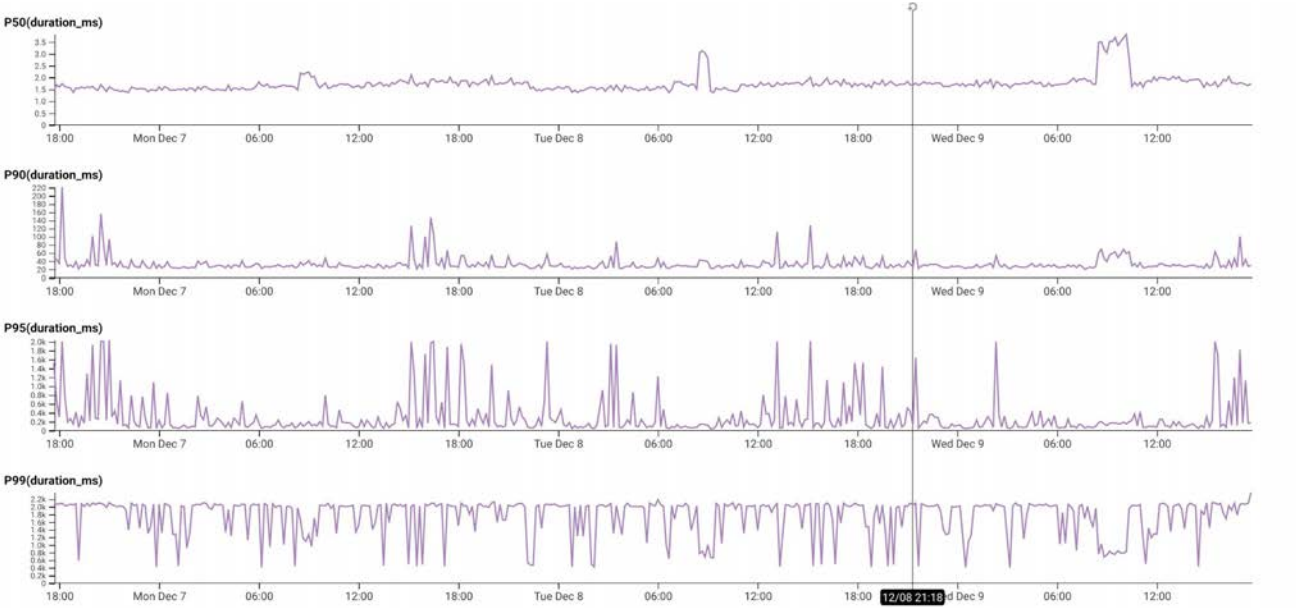
© 2021 Hound Technology, Inc. All Rights Reserved.

```
@PostMapping("flip")
public ResponseEntity flipImage(...) {
    EVENT.addField("content.type", file.getContentType());
    EVENT.addField("action", "flip");
    EVENT.addField("image_id", file.getId());
    EVENT.addField("flip_vertical", vertical);
    EVENT.addField("flip_horizontal", horizontal);
...

    LOGGER.info("Receiving {} image to flip.", file.getContentType());
    byte[] flippedImage = imageService.flip(file, vertical, horizontal);

...

    LOGGER.info("Successfully flipped image id: {}", file.getId());
    EVENT.addField("action.success", "true");
    return new ResponseEntity<>(flippedImage, headers, HttpStatus.OK);
}
```

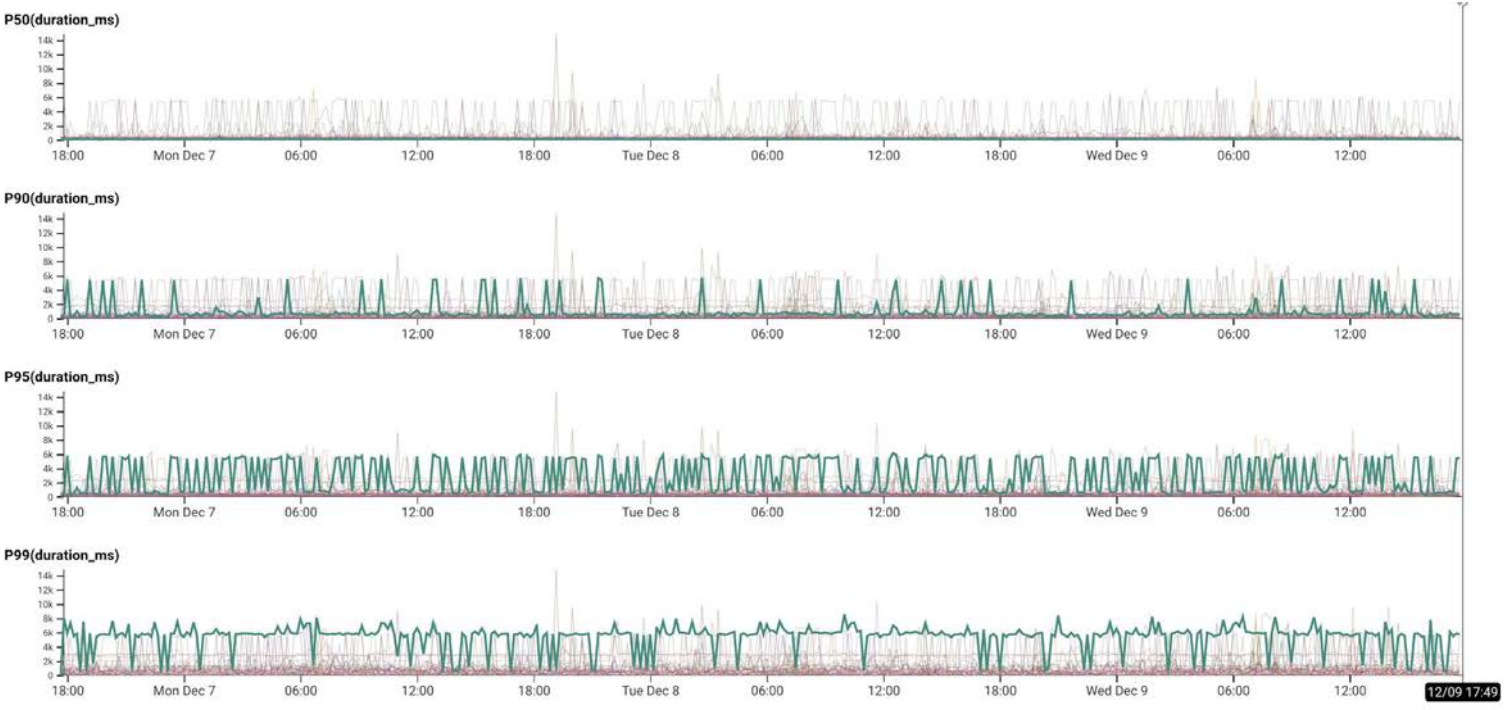
write at the end

© 2021 Hound Technology, Inc. All Rights Reserved.

From events: time-series graphs



© 2021 Hound Technology, Inc. All Rights Reserved.



request.controller	P50(duration_ms)	P90(duration_ms)	P95(duration_ms)	P99(duration_ms)
blazer/queries	143.52602	1,027.18551	1,190.16712	8,749.75702
admin/feedback_messages	94.7299	6,612.339	6,663.38747	8,680.03934
api/v0/followers	5,374.92179	5,559.53726	5,733.41747	6,443.01729
dashboards	93.98542	591.42713	5,317.80995	5,986.49859
followings	39.64306	5,560.11238	5,737.98967	5,904.64685
admin/buffer_updates	2,026.63906	3,428.42137	4,196.07214	4,944.13369
github_repos	451.35437	1,565.51192	2,416.51922	4,484.70042
admin/users	174.62718	347.75028	742.64141	3,654.00178
admin/articles	85.29517	2,149.73763	2,856.83406	3,395.87078
omniauth_callbacks	626.52449	2,644.6029	2,802.78852	3,165.218

From events: error logging

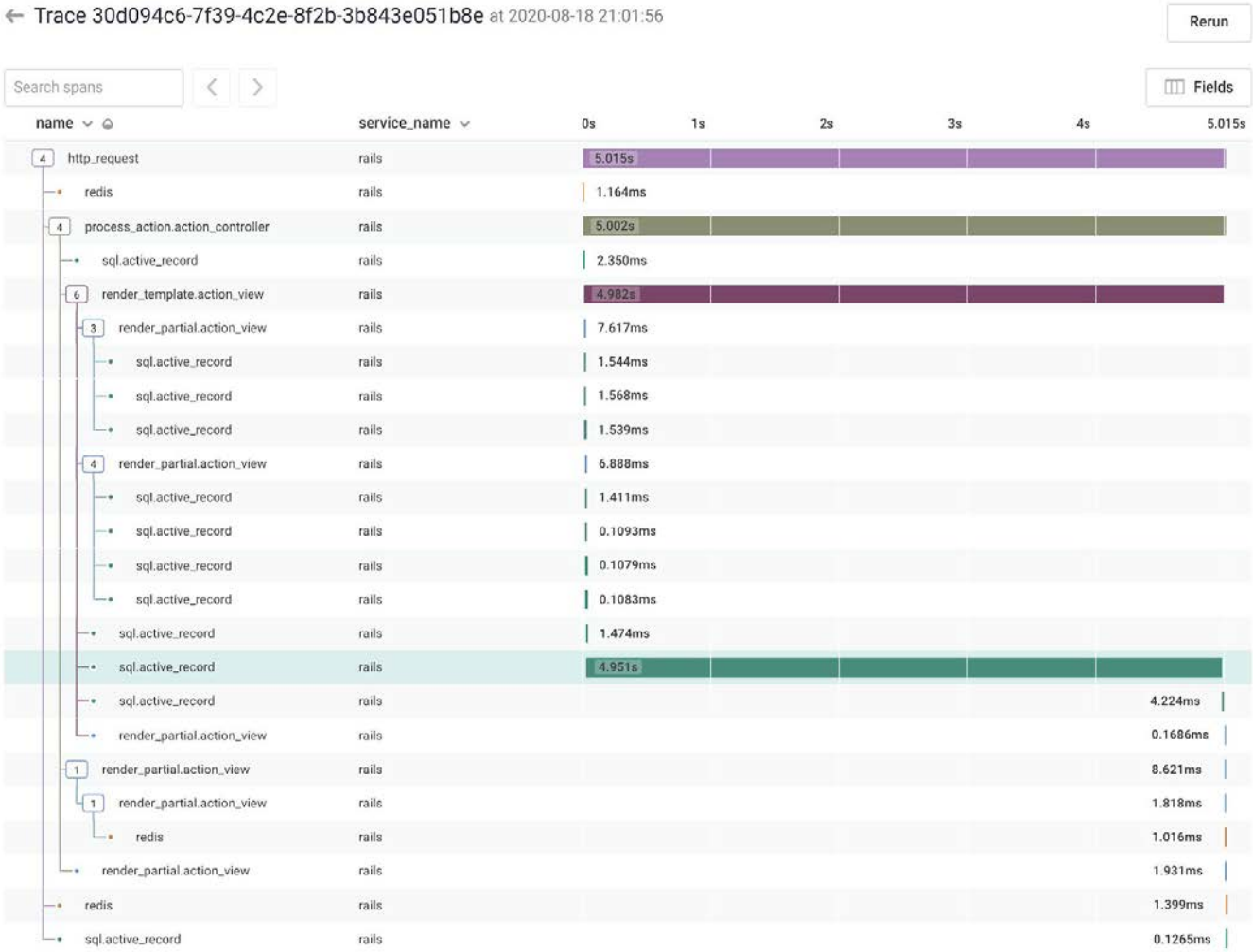
response.status_code	app.error		COUNT
500	something went wrong!		107
500	We couldn't fetch the completed query results! Please reload in a bit.		20
500	rpc error: code = Canceled desc = context canceled		17
500	context canceled		10
500	context canceled /home/circleci/project/types/dataset.go:1297 /home/circleci/project/types/dataset.go:1260 /home/circleci/project/cmd/poodle/handlers/queries.go:498 /home/circleci/project/cmd/poodle/handlers/datasets.go:105 /home/circleci/project/cmd/poodle/handlers/feature_flags.go:96 /home/circleci/project/cmd/poodle/handlers/teams.go:193 /home/circleci/project/cmd/poodle/handlers/datasets.go:83 /home/circleci/project/cmd/poodle/handlers/datasets.go:118 /home/circleci/project/cmd/poodle/handlers/queries.go:468 /home/circleci/project/cmd/poodle/handlers/handlers.go:89 /home/circleci/go/pkg/mod/github.com/gorilla/csrf@v1.5.2-0.20181210000252-d1620373958b/csrf.go:267 /home/circleci/project/cmd/poodle/handlers/util.go:300 /usr/local/go/src/net/http/server.go:2041 /home/circleci/project/cmd/poodle/handlers/util.go:627 /usr/local/go/src/net/http/server.go:2041 /home/circleci/project/cmd/poodle/handlers/util.go:531 /usr/local/go/src/net/http/server.go:2041 /home/circleci/project/cmd/poodle/handlers/util.go:455	(*Dataset).FindSchemaWithoutEditors (*Dataset).FindSchema (*QueriesHandler).Samples.func2 (*Handler).withDataset.func1 (*Handler).withFeatureFlags (*Handler).withTeam (*Handler).withDataset (*Handler).withDatasetMaybePublic (*QueriesHandler).Samples AppHandler.ServeHTTP (*csrf).ServeHTTP LogRequest.func1 HandlerFunc.ServeHTTP AttachRequestID.func1 HandlerFunc.ServeHTTP AttachBuildInfo.func1.1 HandlerFunc.ServeHTTP AddUserToTrace.func1.1	4

© 2021 Hound Technology, Inc. All Rights Reserved.

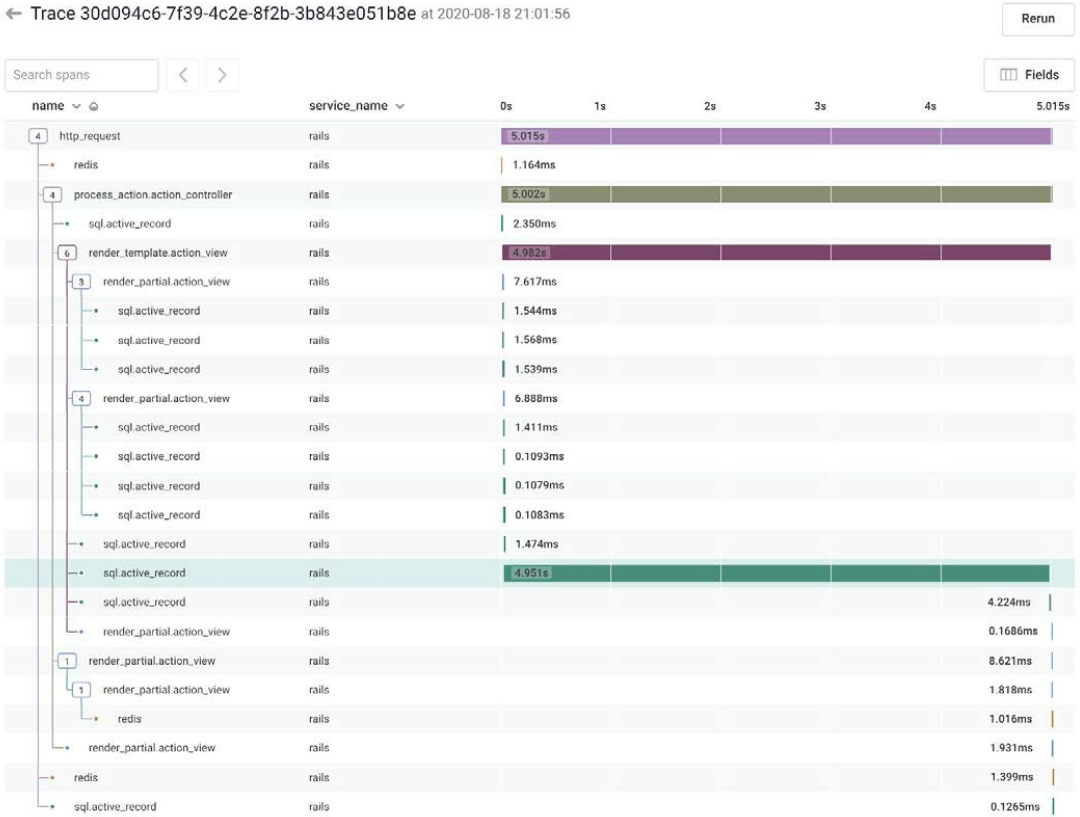
From events: tracing

If your events have start time, duration, parent: then you can generate a trace!

© 2021 Hound Technology, Inc. All Rights Reserved.



sql.active_record: 4.951s



Fields

Filter fields and values in span

name
sql.active_record

service_name
rails

sql.active_record.connection
#<ActiveRecord::ConnectionAdapters::PostgreSQLAdapter:0x00007f39eb11c868>

sql.active_record.connection_id
1147120

sql.active_record.name
Follow Load

sql.active_record.sql
SELECT "follows".* FROM "follows" WHERE "follows"."follower_id" = [redacted] AND "follows"."follower_type" = 'User' AND "follows"."blocked" = FALSE AND "follows"."followable_type" = 'User' ORDER BY "follows"."created_at" DESC LIMIT 80

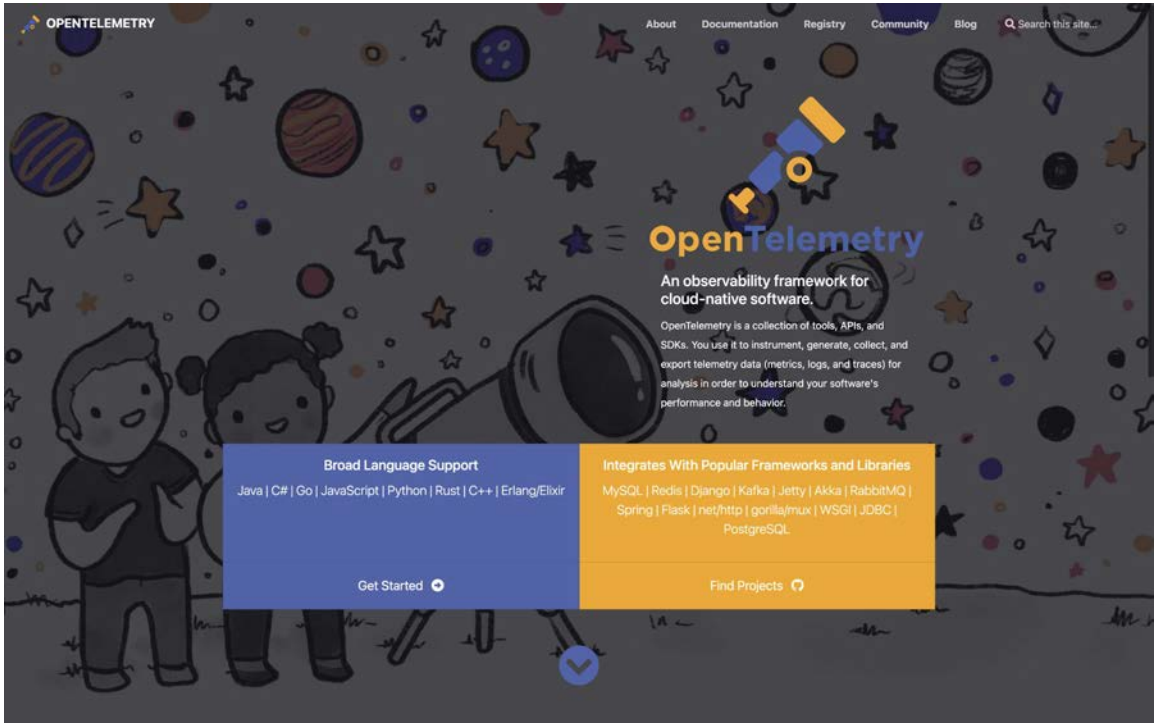
© 2021 Hound Technology, Inc. All Rights Reserved.

Instrument your code

Instrument with OpenTelemetry!

Vendor-neutral framework with auto-instrumentation

Learn more: **OpenTelemetry.io**



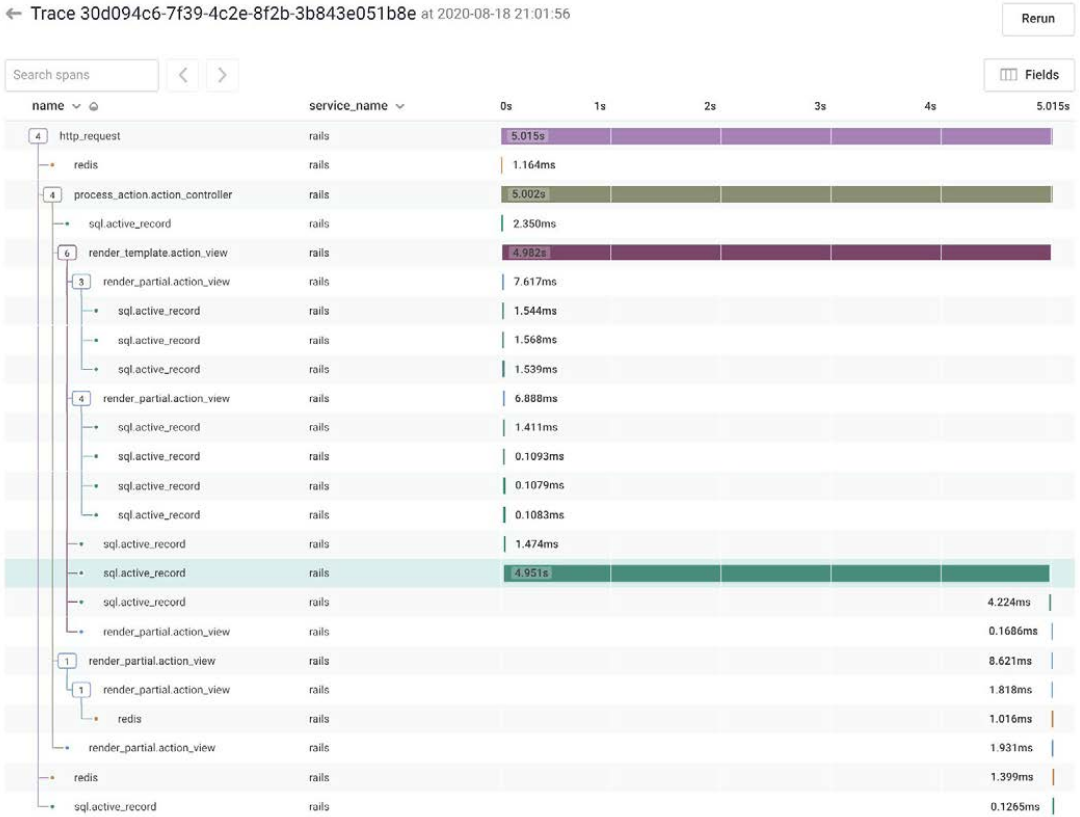
© 2021 Hound Technology, Inc. All Rights Reserved.

Start with auto-instrumentation

Rich context and distributed tracing out of the box:

- HTTP headers
- gRPC calls
- SQL queries

© 2021 Hound Technology, Inc. All Rights Reserved.



Fields

Filter fields and values in span

name
sql.active_record

service_name
rails

sql.active_record.connection
#
<ActiveRecord::ConnectionAdapters::PostgreSQLAdapter:0x00007f39eb11c868>

sql.active_record.connection_id
1147120

sql.active_record.name
Follow Load

sql.active_record.sql
SELECT "follows".* FROM "follows" WHERE "follows"."follower_id" = [redacted] AND "follows"."follower_type" = 'User' AND "follows"."blocked" = FALSE AND "follows"."followable_type" = 'User' ORDER BY "follows"."created_at" DESC LIMIT 80

© 2021 Hound Technology, Inc. All Rights Reserved.

Custom instrumentation

What's important for the service that I'm providing?

- lib/ code consumed by your request endpoints
- 3rd-party API calls

© 2021 Hound Technology, Inc. All Rights Reserved.

```ruby
def get_all_tweets(user)
  all = []
  options = { count: 200, include_rts: true }
  loop do
    tweets = client.user_timeline(user, options)
    return all if tweets.empty?
    all += tweets
    options[:max_id] = tweets.last.id - 1
  end
end
```

© 2021 Hound Technology, Inc. All Rights Reserved.

```ruby
def get_all_tweets(user)
  Instrumentation.start_span(name: 'get_all_tweets') do
    Instrumentation.add_field_to_trace('user', user)
    all = []
    options = { count: 200, include_rts: true }
    loop do
      Instrumentation.start_span(name: 'get_batch') do
        Instrumentation.add_field('options', options)
        tweets = client.user_timeline(user, options)
        return all if tweets.empty?
        all += tweets
        options[:max_id] = tweets.last.id - 1
      end
    end
  end
end
```

© 2021 Hound Technology, Inc. All Rights Reserved.

How does this benefit teams?

Better data → better conversations

Shared source of truth.

Improved knowledge transfer.

Shared ownership of production.

© 2021 Hound Technology, Inc. All Rights Reserved.

Observability-driven development

Before making a change, ask:
How will I know this code is working as intended in prod?

Observe the current state:
What am I trying to change?

Then implement + instrument:
Look, it's working!

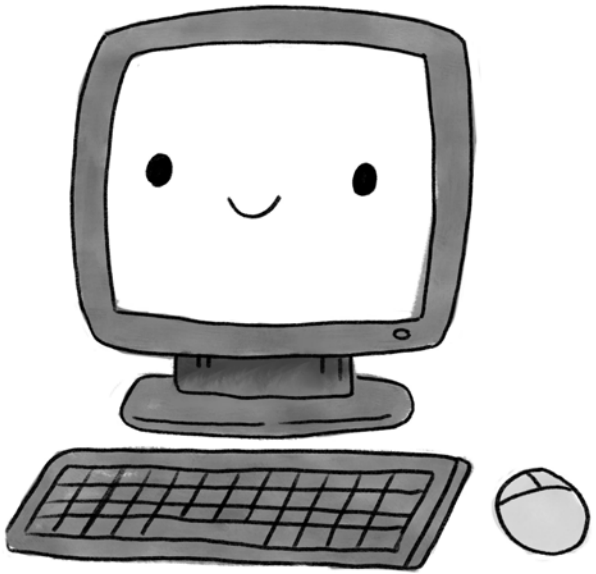
© 2021 Hound Technology, Inc. All Rights Reserved.

One tool for dev and prod

Fast feedback loop in dev

Strong debugging skills in prod

Read more:
The Future of Developer Careers
go.hny.co/dev-careers



© 2021 Hound Technology, Inc. All Rights Reserved.

Level up your team

Better data leads to better conversations

Custom instrumentation enables self-serve querying

Knowledge transfer creates production ownership

© 2021 Hound Technology, Inc. All Rights Reserved.

Where to start?

Start where you're at

Take inventory of the tools and data your team has access to

- How often are people actually using them?
- How often are you actually able to answer questions?
- How much of the time do you rely on experience (and guessing) vs. data?

© 2021 Hound Technology, Inc. All Rights Reserved.

Instrument your code

Structured events!

- How to capture them?
- Look into OpenTelemetry
- Is there auto-instrumentation for your tech stack?

© 2021 Hound Technology, Inc. All Rights Reserved.

Start small and grow

Start with one app or one chunk of code in your critical path

- Critical code helps you learn the fastest

Try out some tools

- Some vendors offer a free tier or trial
- Send from dev or deploy a canary

© 2021 Hound Technology, Inc. All Rights Reserved.

Reach out!

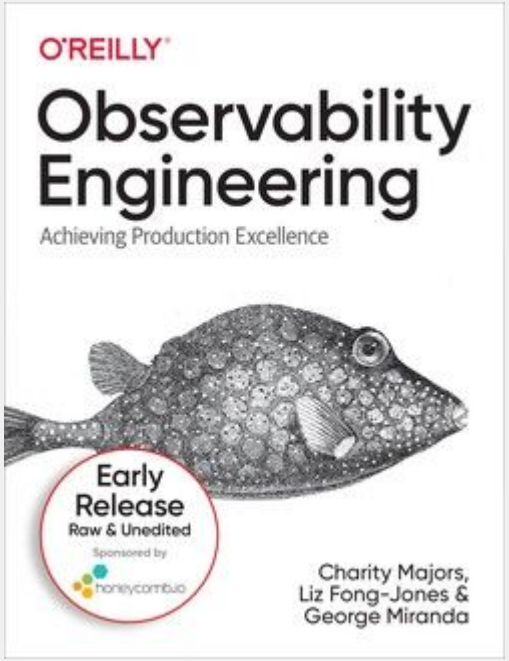
Get these slides: **hny.co/shelby**

Twitter: **@shelbyspees**

Visit our Booth!

Get O’Reilly pre-release chapters!

Get Honeycomb swag!



Questions?