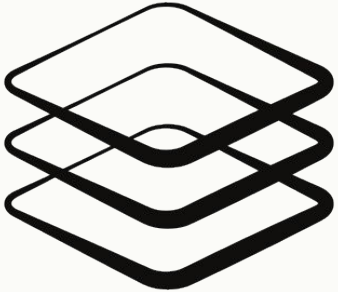


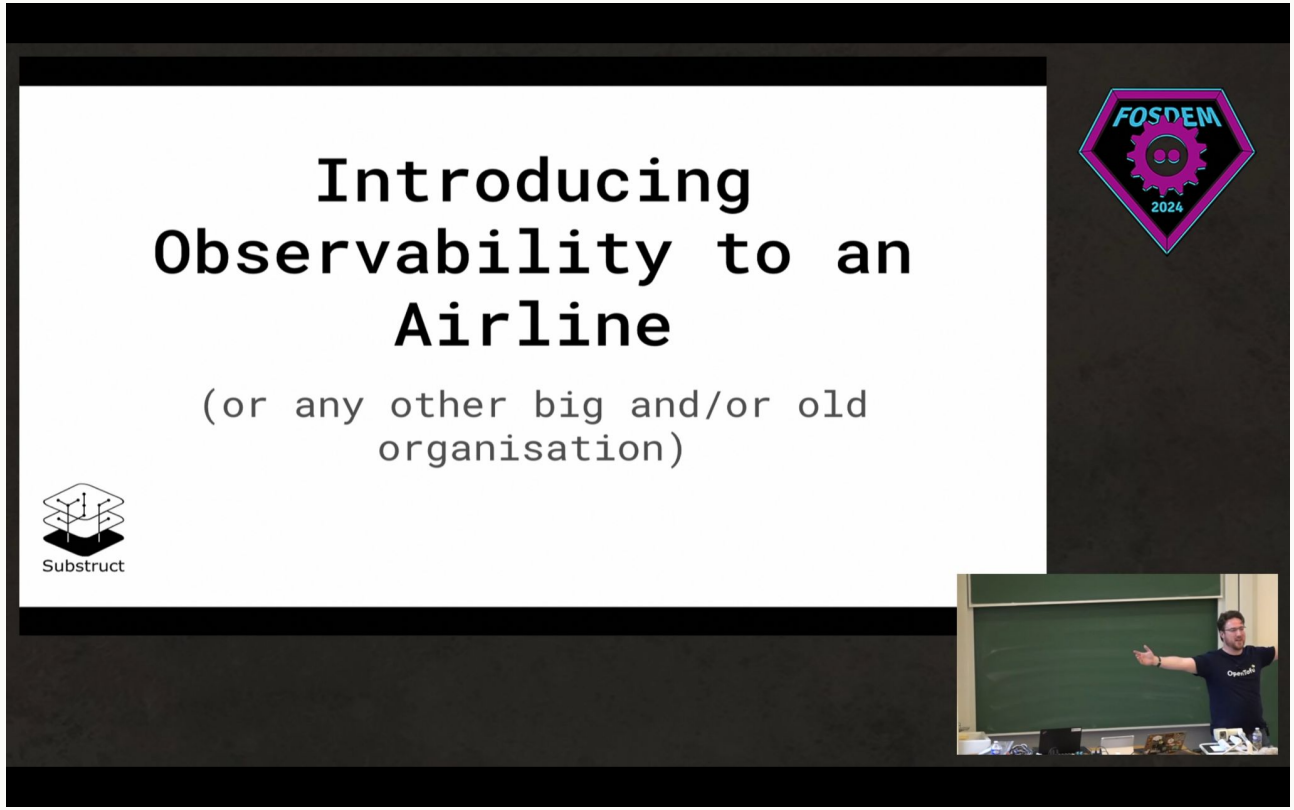
The Performance Impact of Auto-Instrumentation

James Belchamber

Who am I

- 20 years in the field
  - Computer Repair
  - IT Support
  - Linux SysAdmin
  - “DevOps Engineer”
  - Platform Engineer
- Started a two-person IT consultancy
  - Dev 🤝 Ops
  - Quickly found a niche in the observability space
- Working on an observability transformation since 2023

Who am I


Introducing Observability to an Airline
(or any other big and/or old organisation)
Substruct
FOSDEM
2024

Who am I

What on earth are you doing, James?

- Building an “Observability Platform”
  - ADOT Collectors > LGTP stack
  - Vended Dashboards
  - Legacy data sources
  - Everything as-code 😎
- Dramatically increasing Monitoring & Observability
  - Node/Windows Exporter on all instances
  - RED/USE-based Automatic Dashboards
  - Auto-Instrumenting ALL THE THINGS

Auto-what?

- Auto-Instrumentation!
- Free Traces from your existing code!
- Just attach it to your applications in:
  - Java/JavaScript
  - Python
  - PHP
  - .NET
  - Go?!?
- And they start leaking tasty telemetry.

But WAIT!

You can’t just start automatically adding code to all our services!

Well.. code does cost cycles, James.

Part 1: Basic Testing

Java Auto Instrumentation

Getting started

Setup

1. Download opentelemetry-javaagent.jar from Releases of the `opentelemetry-java-instrumentation` repository and place the JAR in your preferred directory. The JAR file contains the agent and instrumentation libraries.
2. Add `-javaagent:path/to/opentelemetry-javaagent.jar` and other config to your JVM startup arguments and launch your app:
   - Directly on the startup command:

```
gent:path/to/opentelemetry-javaagent.jar -Dotel.service.na
```

   - Via the `JAVA_TOOL_OPTIONS` and other environment variables:

```
export JAVA_TOOL_OPTIONS="-javaagent:path/to/opentelem
export OTEL_SERVICE_NAME="your-service-name"
java -jar myapp.jar
```

Hello World Java

```
public class DemoApplication {
    public static void main(String[] args) {
      SpringApplication.run(DemoApplication.class, args);
    }
    @GetMapping("/hello")
    public String hello() {
      return String.format("Hello");
    }
}
```

Hello World Java - Sleep 1s

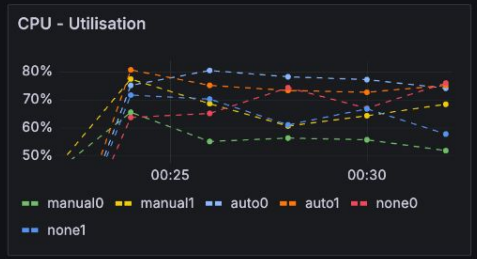
```
import http from 'k6/http';
import { sleep } from 'k6';
export const options = {
  vus: 10,
  duration: '10m',
};
export default function() {
  http.get('http://<super-secret-ip-address>:8080/hello');
  sleep(1);
}
```

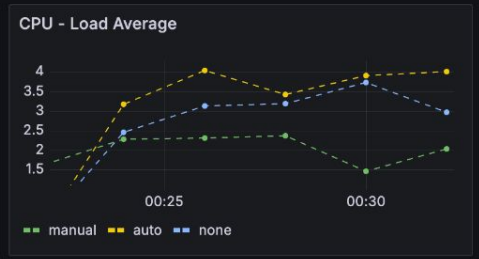
Hello World Java - Sleep 1s - RED

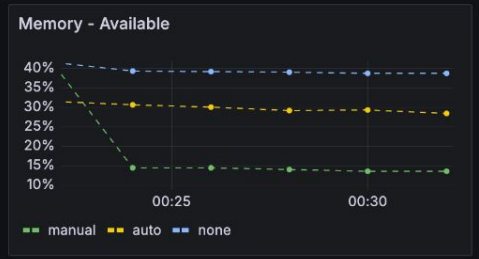
Hello World Java – Sleep 1s – USE

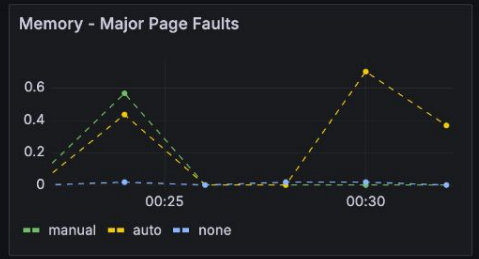
Hello World Java - No Sleep - RED

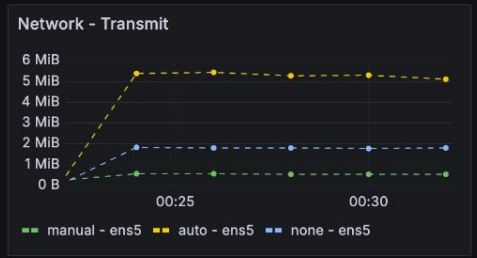
Hello World Java - No Sleep - USE

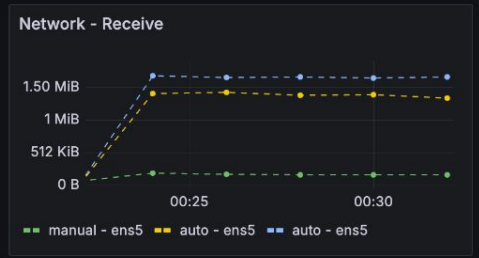


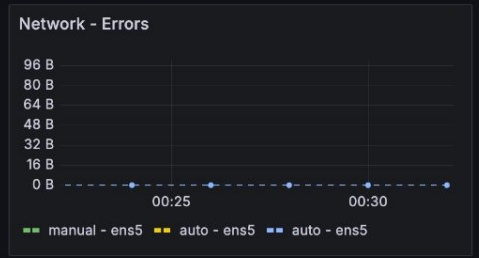


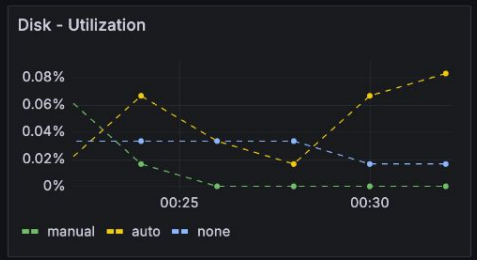


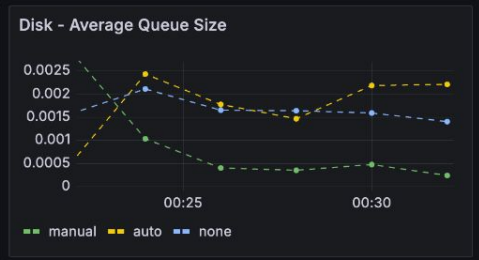












Go Auto Instrumentation

OpenTelemetry

Docs Ecosystem Status Community Blog English Search this site...

Docs

- What is OpenTelemetry?
- Getting Started
- Concepts
- Demo
- Language APIs & SDKs
- Zero-code Instrumentation
  - Go
  - .NET
  - PHP
  - Python
  - Java
  - JavaScript
- Collector
- Kubernetes
- FaaS
- Migration
- Specs
- Security
- Contributing

Docs / Zero-code Instrumentation / Go

Go zero-code instrumentation

Zero-code instrumentation for Go provides a way to instrument any Go application and capture telemetry data from many popular libraries and frameworks without any code changes.

This project is currently work in progress and you can visit the opentelemetry-go-instrumentation repository to learn more.

Feedback

Was this page helpful?

Yes No

Last modified June 20, 2024: add a base page for go zero code instrumentation (#4718) (8a977193)

Hello World Go

```
func main() {
    router := gin.Default()
    router.GET("/hello", helloWorld)

    router.Run()
}
func helloWorld(c *gin.Context) {
    c.IndentedJSON(http.StatusOK, "Hello world!")
}
```

Hello World Go - Sleep 1s - RED

Hello World Go - Sleep 1s - USE

Hello World Go - No Sleep - RED

Hello World Go – No Sleep – USE

Auto-Instrumentation does use resources

Lesson 1

Manual Instrumentation also uses resources

Lesson 2

Manual Instrumentation performance depends on the implementation

Lesson 3

Hello World Java - No Sleep - RED - 2nd

Hello World Go - No Sleep - USE - 2nd

Auto-Instrumentation performance is far more consistent

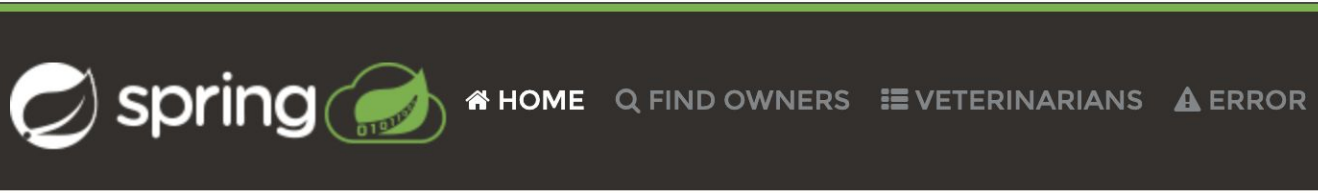
Lesson 4

Testing other applications says nothing about YOUR application

Lesson 5

Part 2: Real Applications

PetClinic

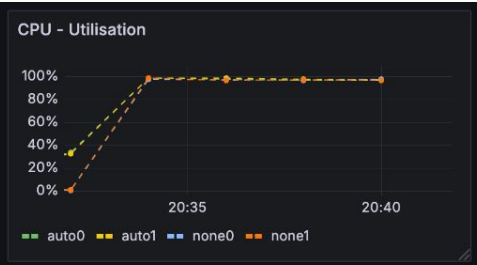


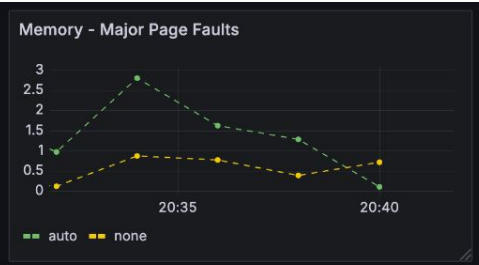
Welcome

spring by VMware Tanzu

PetClinic - 20VUs - RED

PetClinic – 20VUs – USE





PetClinic - 1VU - RED

PetClinic – 1VU – USE

An application that is doing something significant is probably not impacted by auto-instrumentation

Lesson 6

```
httpd
```

httpd - 50VUs - RED

httpd - 50VUs - USE

Word
Pres
s

My WordPress website 0 New Howdy, admin

Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu

Screen Options Help

Dashboard

Site Health Status

Good

Your site's health is looking good, but there are still some things you can do to improve its performance and security.

Take a look at the **4 items** on the Site Health screen.

At a Glance

1 Post
1 Page
1 Comment

WordPress 6.4.3 running Twenty Twenty-Four theme.

Activity

Recently Published

Today, 10:11 am Hello world!

Recent Comments

From A WordPress Commenter on Hello world!
Hi, this is a comment. To get started with moderating, editing, and deleting comments, please visit the Comments screen in...

All (1) | Mine (0) | Pending (0) | Approved (1) | Spam (0) | Trash (0)

Quick Draft

Title

Content

What's on your mind?

Save Draft

WordPress Events and News

Attend an upcoming event near **London**. Select location

WordPress 6.5 Brighton Launch Party
Meetup • Brighton, United Kingdom
Tuesday, Mar 26, 2024
8:00 pm GMT+1

Online WordPress Portsmouth Meetup - My Favourite Contact Form
Meetup • Online
Wednesday, Mar 27, 2024
8:00 pm GMT+1

Secure Your Site: Join Cambridge WordPress Backups & Security Meetup Apr8th 7pm
Meetup • Online
Monday, Apr 8, 2024
8:00 pm GMT+2

WordPress 6.5 Release Candidate 3

WP Briefing: Episode 75: WordCamp Asia 2024 Unwrapped

Gutenberg Times: Dev hours on Block Hooks, building better patterns, customizing your store, a new book and more — Weekend Edition 288

WordPress - 20VUs - RED

What are you doing WordPress

Test

Sample Page

Sample Page

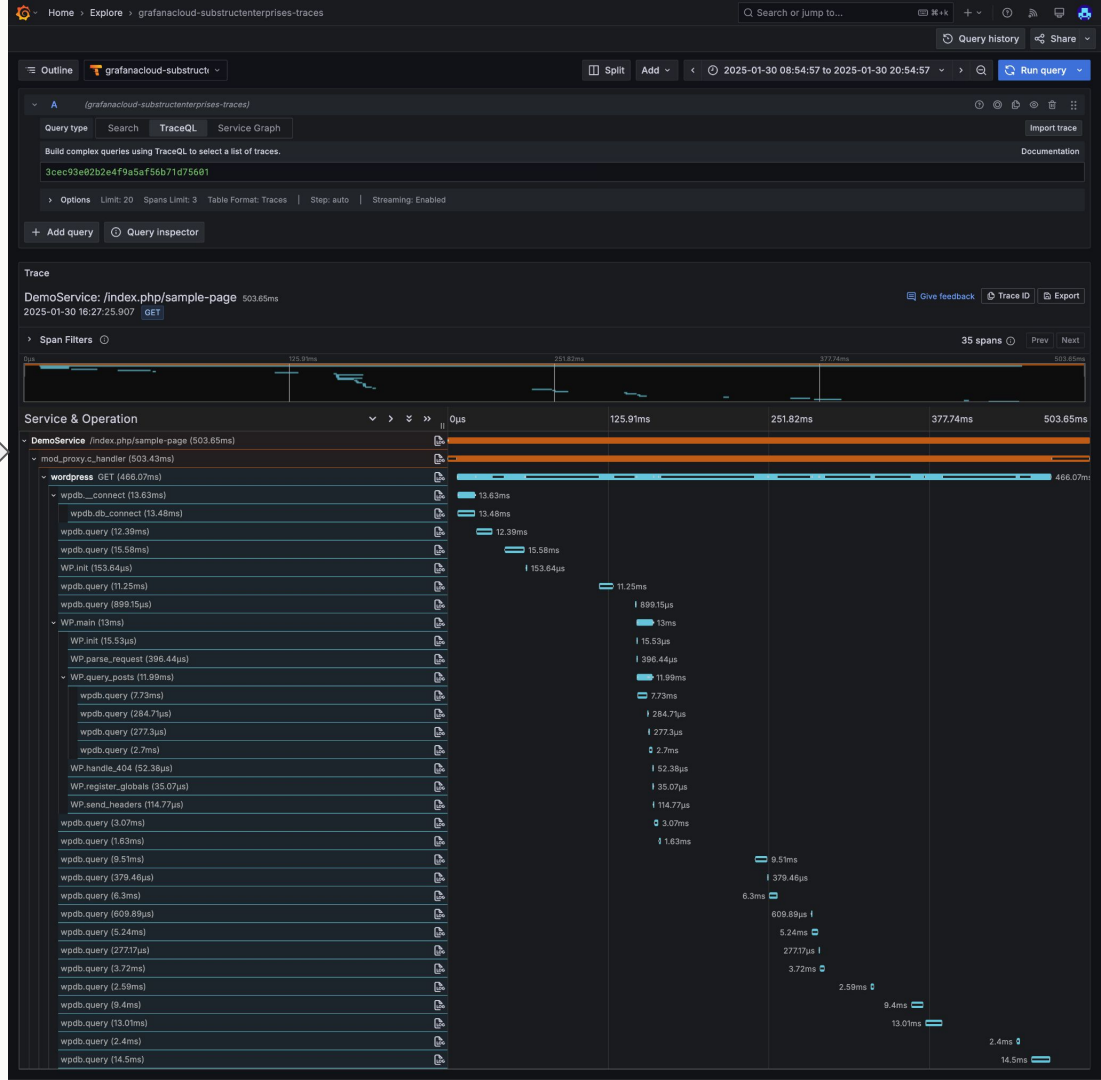
This is an example page. It's different from a blog post because it will stay in one place and will show up in your site navigation (in most themes). Most people start with an About page that introduces them to potential site visitors. It might say something like this:

> Hi there! I'm a bike messenger by day, aspiring actor by night, and this is my website. I live in Los Angeles, have a great dog named Jack, and I like piña coladas. (And gettin' caught in the rain.)

...or something like this:

> The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the public ever since. Located in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for the Gotham community.

As a new WordPress user, you should go to your dashboard to delete this page and create new pages for your content. Have fun!



WordPress – 20VUs – USE

Performance Testing is incredibly important!

Lesson 7

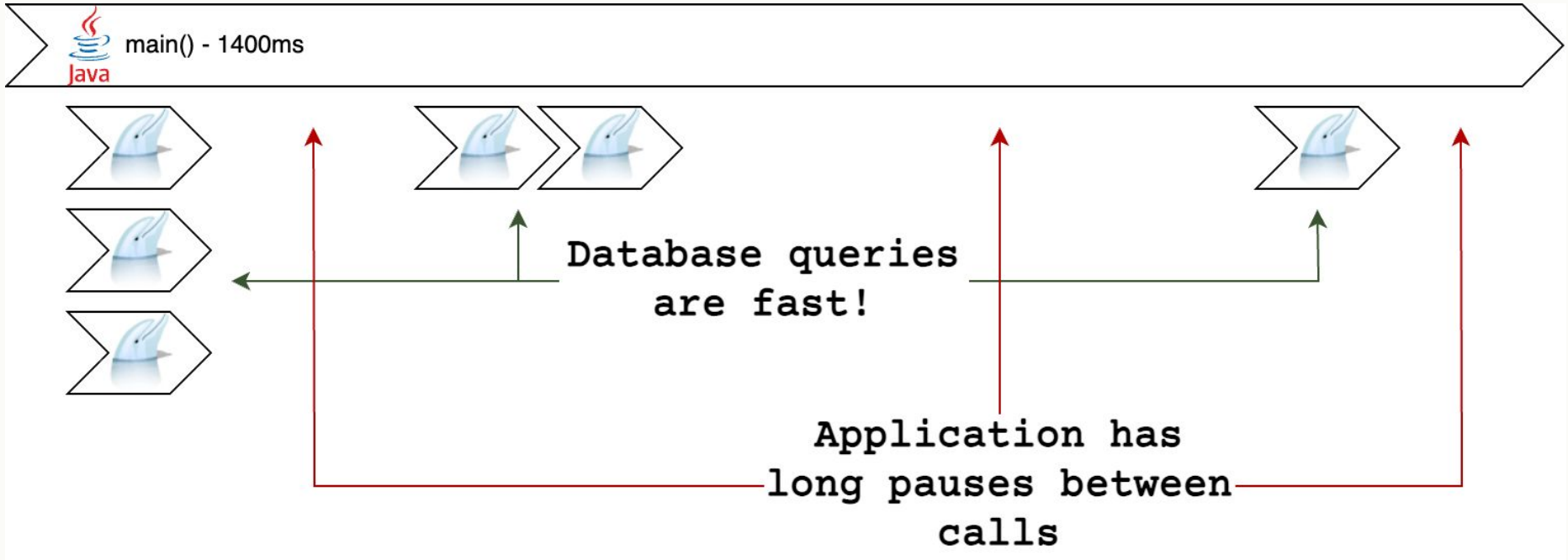
Was it the little mouse,
the last to get in,
who was lightest of all?

Could it be him?

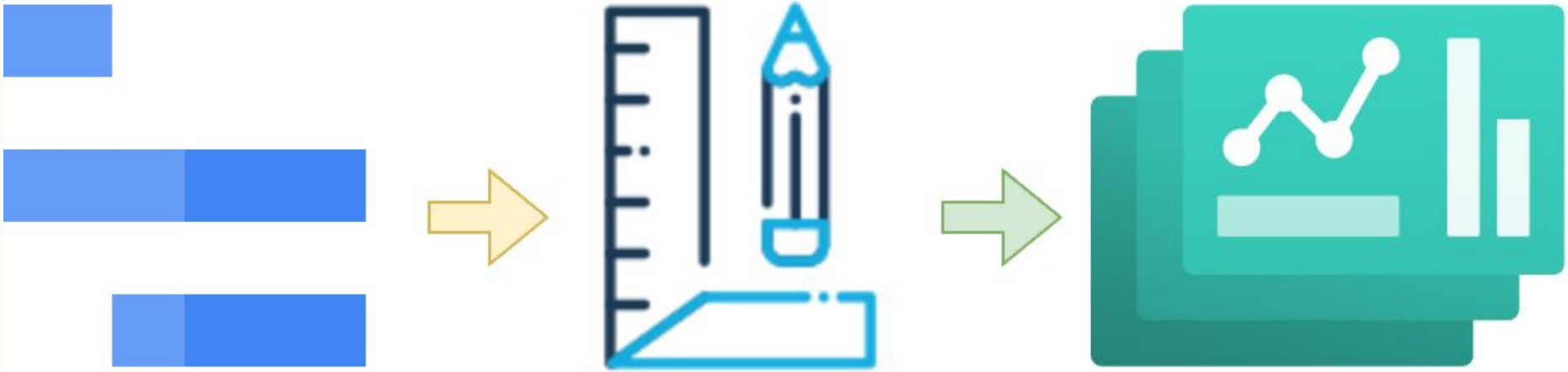
You DO know who sank the boat.

Part 3: But why?

Individual Traces



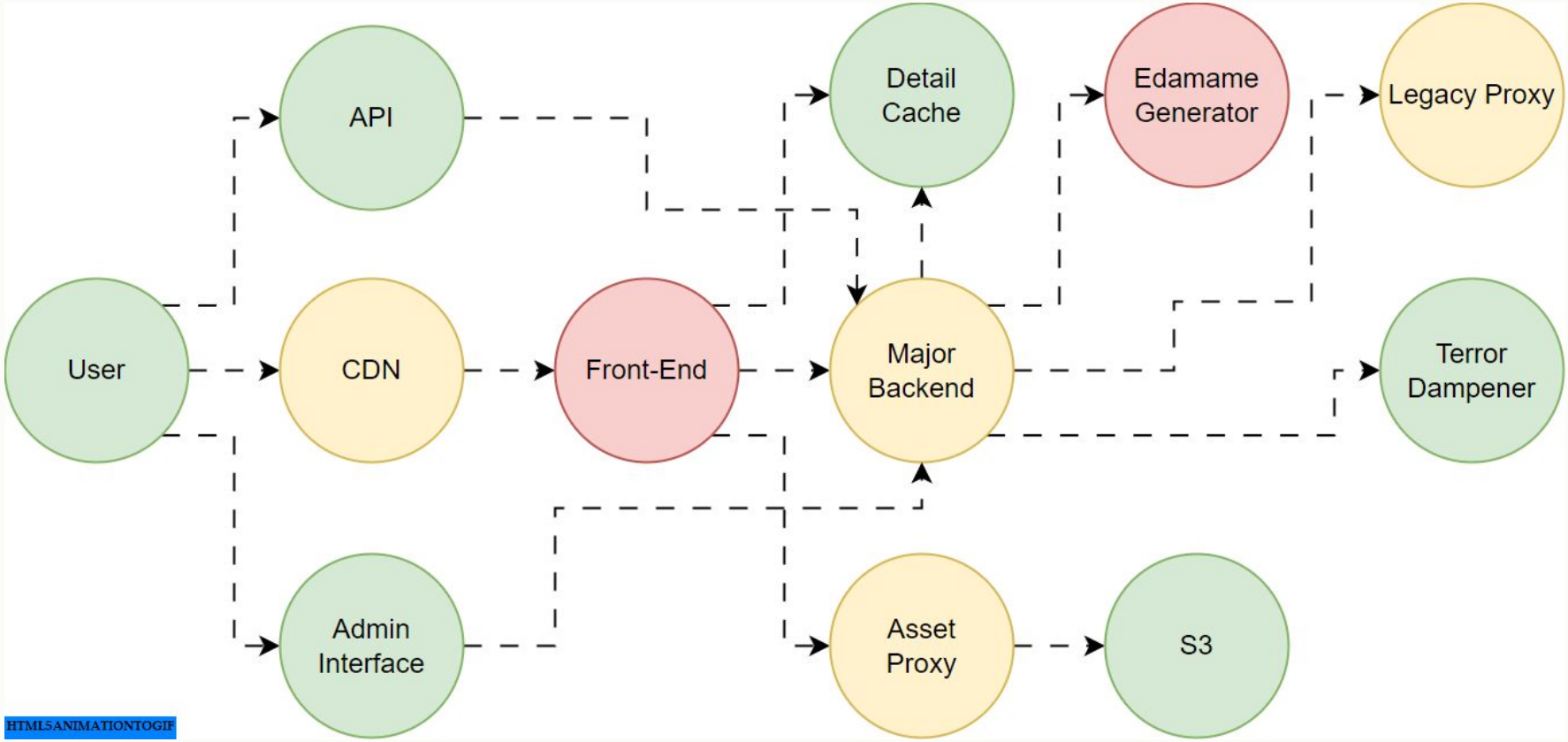
Automatic Metrics

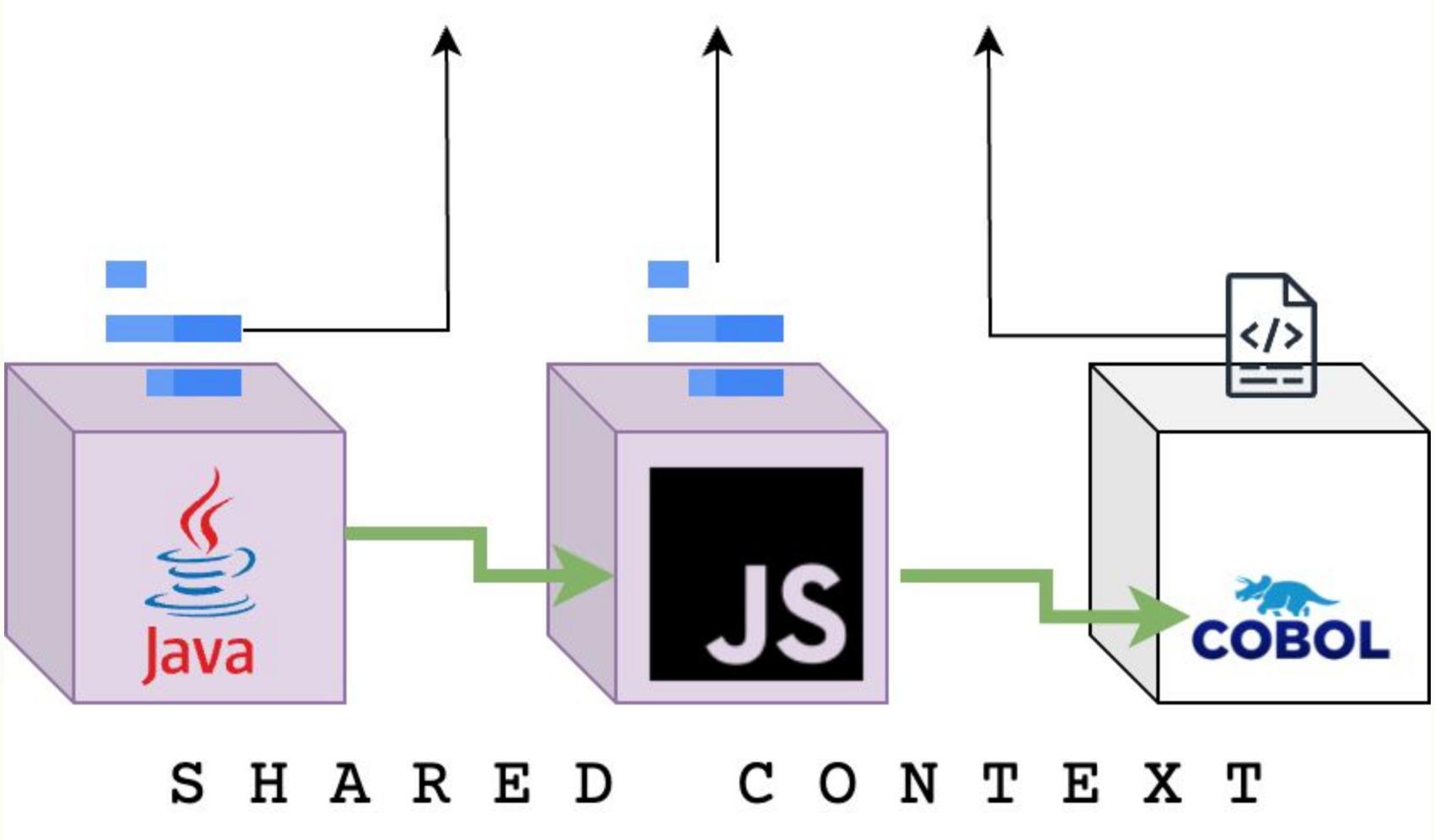


Auto-Generate Metrics from Traces

Auto-Generate Dashboards with Metrics

Service Graphs




Java
JS
COBOL
SHARED CONTEXT

Most people should probably be implementing auto-instrumentation

Lesson 8

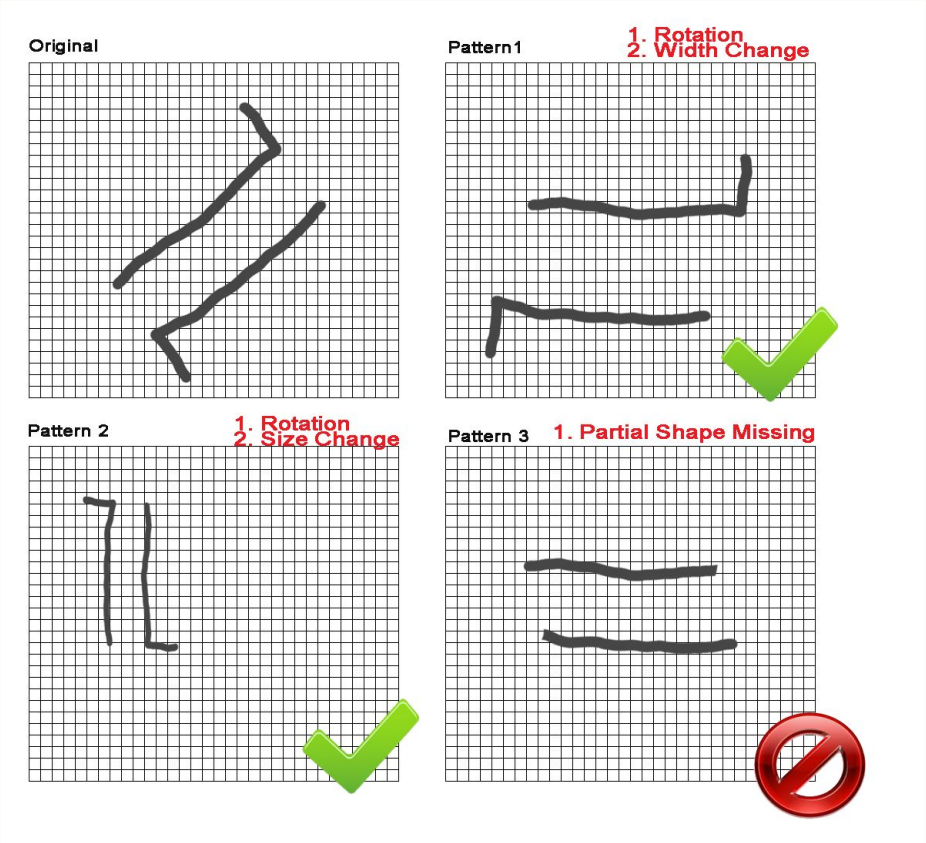
Sidenote: Manual Instrumentation

Instrumentation = modifying ALL YOUR CODE

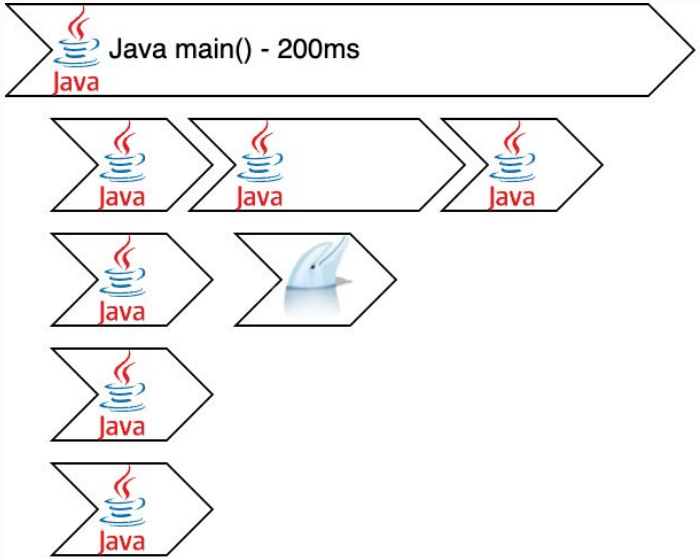
```
from opentelemetry import trace
from opentelemetry.trace import Status,
StatusCode

tracer =
trace.get_tracer("sample-python-handler")

def parent_work():
    with
tracer.start_as_current_span("parent") as
parent_span:
    print("Calling Children..")
    parent_span.set_attribute("foo",
"bar")
    child_work("bob")
    child_work("dick")
    child_work("harry")
```

```
def child_work(child_name):
    with
tracer.start_as_current_span("child") as
child_span:
        try:
            print(child_name + " where
are you?")
            print("Here I am!")
        except:
            child_span.set_status(
                Status(StatusCode.ERROR)
            )
```

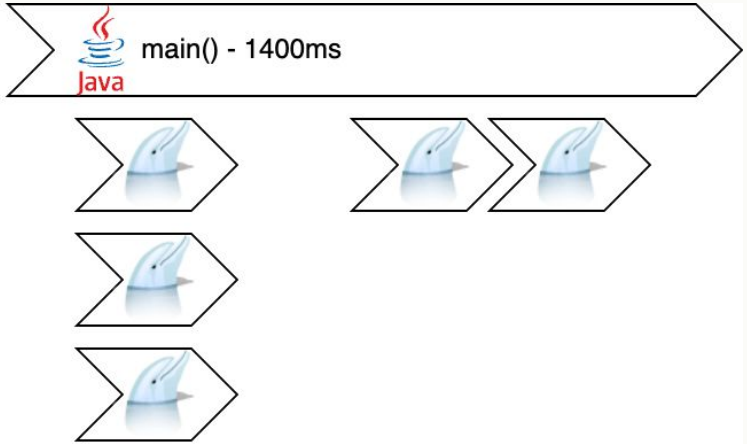
Pattern matching



Only instruments frameworks and libraries



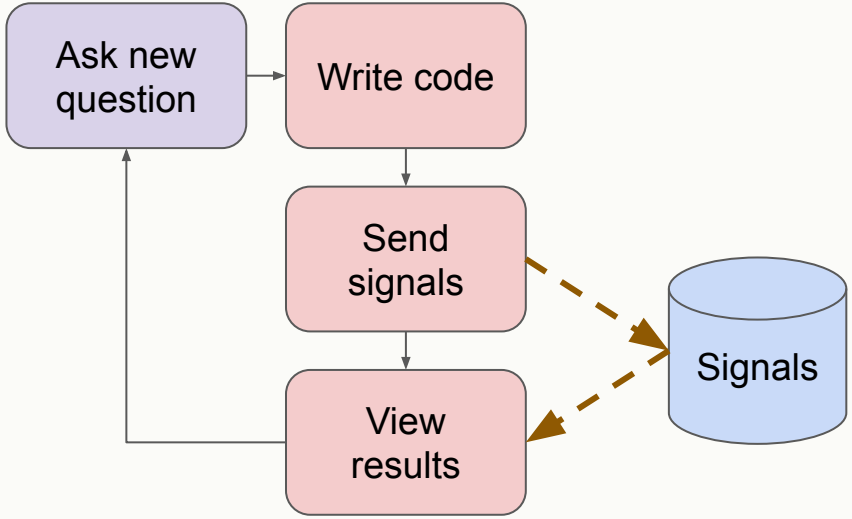
- Mostly frameworks
- Lots of insight into application
- Lots of spans



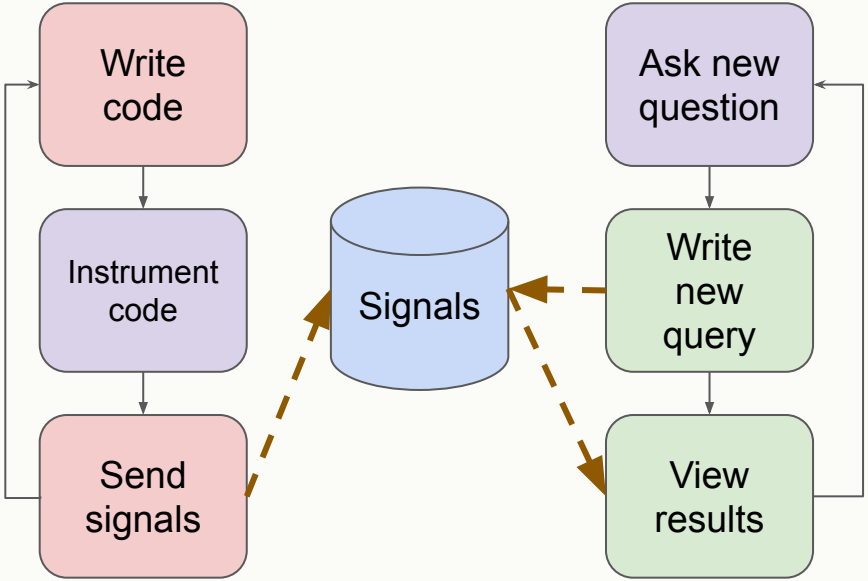
- Mostly custom code
- Little insight into application
- Only database calls are instrumented

Separation of Tasks

Monitoring



Observability



Part 4: Miscellaneous

AWS Lambda

- The ADOT Lambda Layer includes a collector
  - This dramatically increases execution time of your functions
- We have had good experiences with external collectors
  - Use upstream OpenTelemetry layers instead
  - These split the collector and language layers - use the latter
- The layer is fine when execution time isn’t important

(We didn’t actually do any public performance testing here to demonstrate - “trust me bro” or test it yourself)

Beyla (eBPF) - PetClinic (Java) 1VU - RED

Beyla (eBPF) – PetClinic (Java) 1VU – USE

What did we learn?

- Auto-Instrumentation does use resources
- Manual Instrumentation also uses resources
- Manual Instrumentation performance depends on the implementation
- Auto-Instrumentation performance is far more consistent
- Testing other applications says nothing about YOUR application
- An application that is doing something significant is probably not impacted by auto-instrumentation
- Performance Testing is incredibly important!
- Most people should probably be implementing auto-instrumentation

James Belchamber

Substruct

✉ james@substruct.co.uk

🌐 https://www.substruct.co.uk/

Personal

✉ james@belchamber.com

🌐 https://james.belchamber.com/

Any community-led projects out there that want to start instrumenting?

Talk to us :)

Any questions?