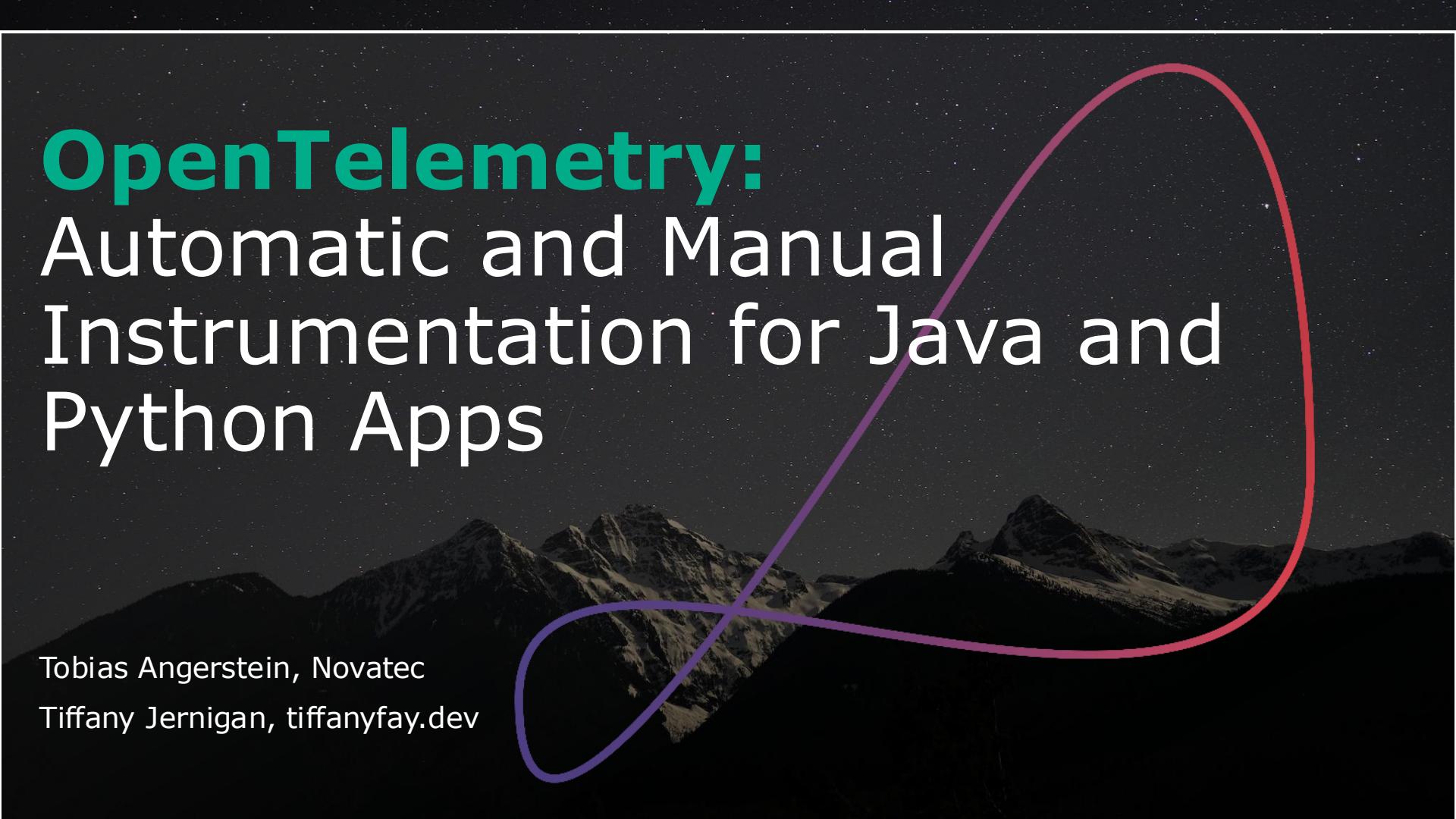


OpenTelemetry: Automatic and Manual Instrumentation for Java and Python Apps



Tobias Angerstein, Novatec

Tiffany Jernigan, tiffanyfay.dev

Who we are



- **Tobias Angerstein**
- Senior Observability Consultant
- Novatec Consulting
- angerstone (LI)



- **Tiffany Jernigan**
- Developer Advocate, CNCF Ambassador
- tiffanyfay.dev 
- tiffanyfayj (LI/X)



Credits



Jan-Niklas Tille



Jens Plüddemann



Matthias Haeussler

Agenda

- OpenTelemetry Overview
- Demo: OpenTelemetry in Action
- Demo: Automatic Instrumentation
- Demo: Manual Instrumentation



Want to follow along?

The screenshot shows the course landing page for 'Getting Started with OpenTelemetry (LFS148)'. The page has a blue header with the course title and a large image of a smartphone displaying the course interface. Below the phone image, there's a 'Enroll Today' button. To the right, there's a sidebar with course details like 'Who is it For?' and a QR code.

Getting Started with OpenTelemetry (LFS148)

Learn to use OpenTelemetry to build and manage unified observability, skills increasingly important to IT developers and engineers career growth.

Who is it For?

This course is designed for software developers, DevOps engineers, site reliability engineers (SREs), and full-stack or backend developers looking to

\$0
Enroll Today

My Portal Below Enrolling

Who is it For?

This course is designed for software developers, DevOps engineers, site reliability engineers (SREs), and full-stack or backend developers looking to

Online, Self Paced
8-10 Hours of Course Material
Discussion Forum
Unlimited Access to Online Course
Digital Badge

QR Code:

The screenshot shows a GitHub repository page for 'lfs148x/LFS148-code'. The repository is public and contains several recent commits from 'JenSeReal' and 'Jens Plüddemann'. The commits are related to updating package names and .github/workflows. The repository also includes .gitignore and .nordr.xml files. On the right side, there's a sidebar with various repository management options.

lfs148x / LFS148-code

Code Issues Pull requests Actions Projects Security Insights

LFS148-code Public

main Go to file + Code About

JenSeReal and Jens Plüddemann Update package names last week

.devcontainer Update package names last week

.github/workflows Update package names last week

.idx

.vscode

assets

exercises

.gitignore

.nordr.xml

Public code for lfs148x

Readme View license Activity Custom properties

stars watching fork repository issues published

What is OpenTelemetry (in a nutshell)?

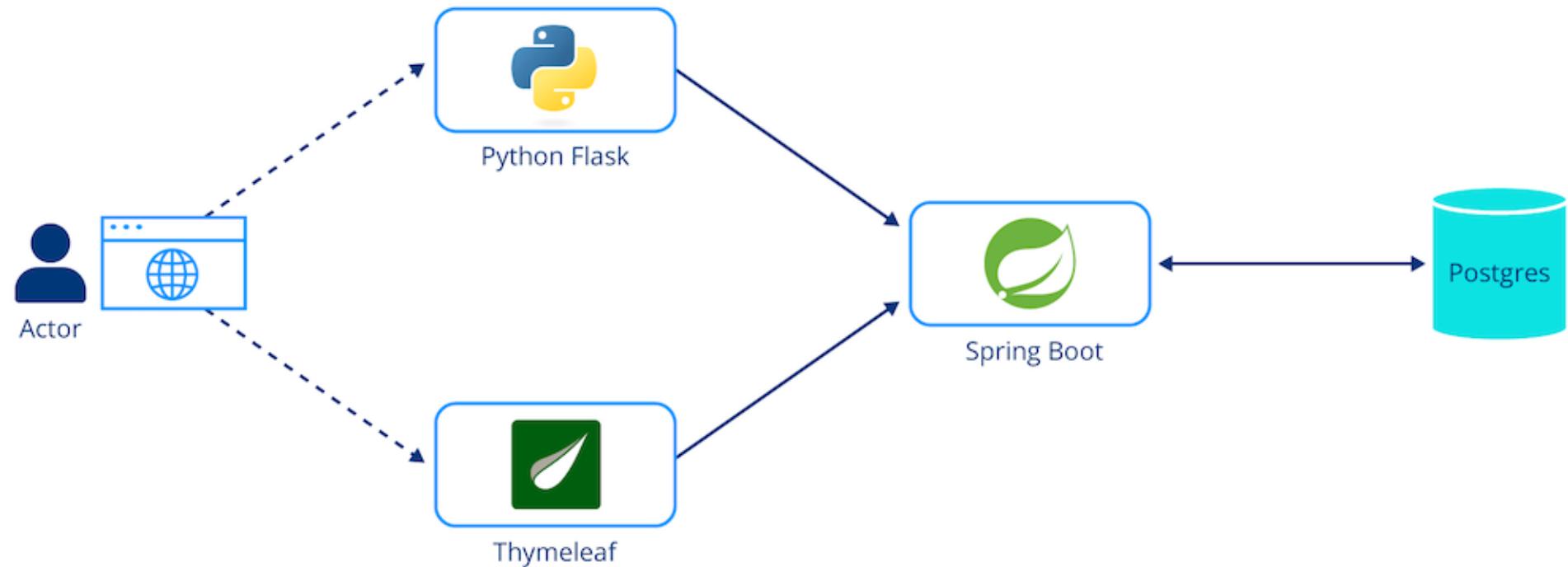
- The main goals of OpenTelemetry (OTel) are:
- **Unified telemetry:** Combines tracing, logging, and metrics into a single framework enabling correlation of all data and establishing an open standard for telemetry data.
- **Vendor-neutrality:** Integration with different backends for processing the data.
- **Cross-platform:** Supports various languages (Java, Python, Go, etc.) and frameworks, making it versatile for different development environments.

What OpenTelemetry is NOT

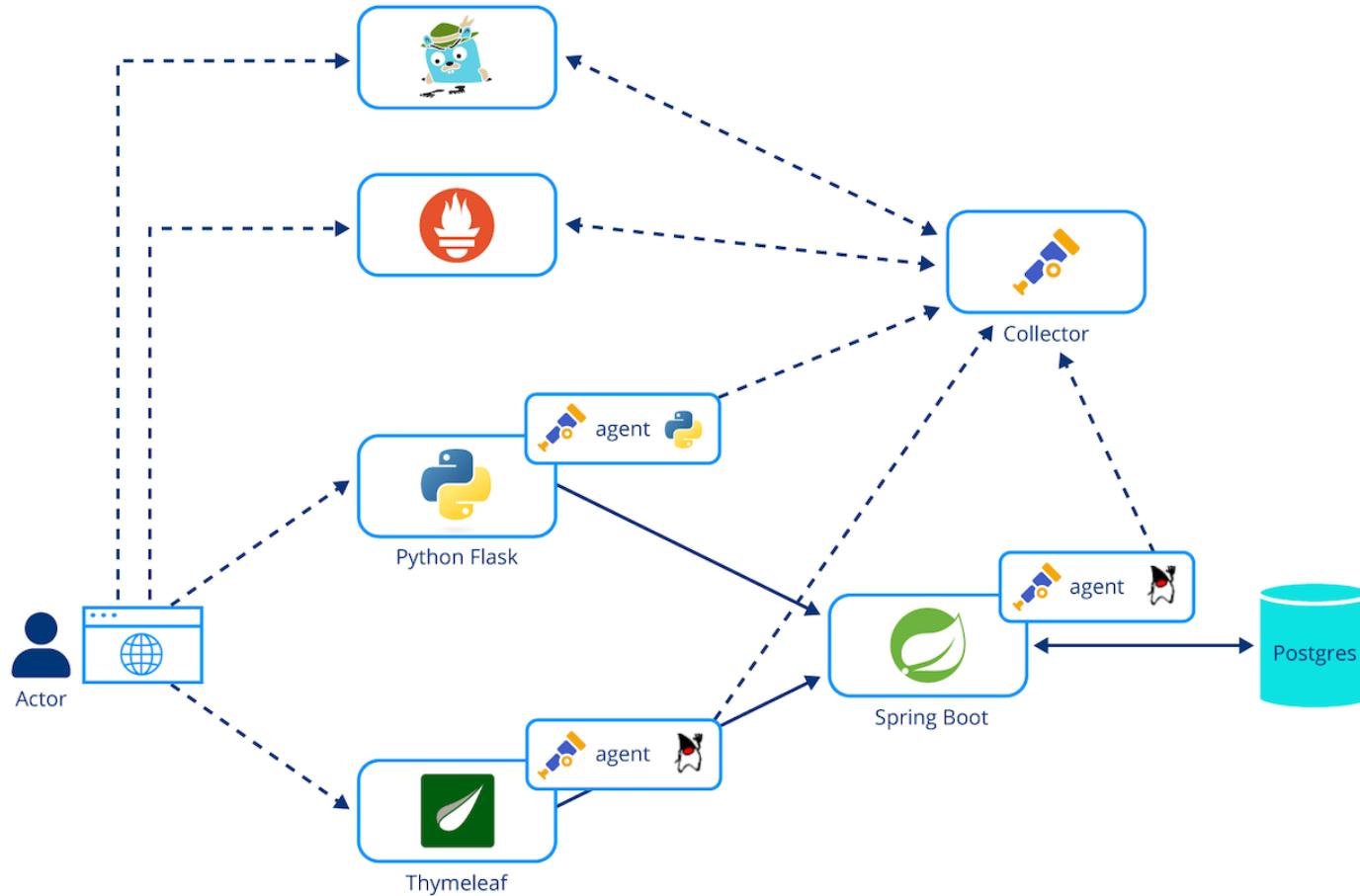
- Not an All-in-One Monitoring or Observability Tool
- Not a Data Storage or Dashboarding Solution
- Not a Pre-configured Monitoring Tool
- Not a Performance Optimizer

OpenTelemetry in Action

Application Architecture

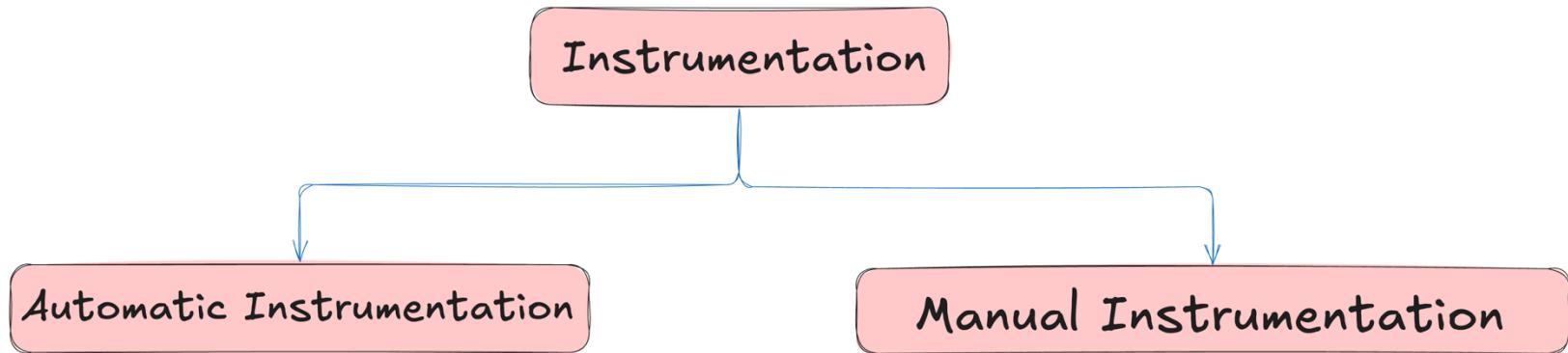


Application Architecture Instrumented



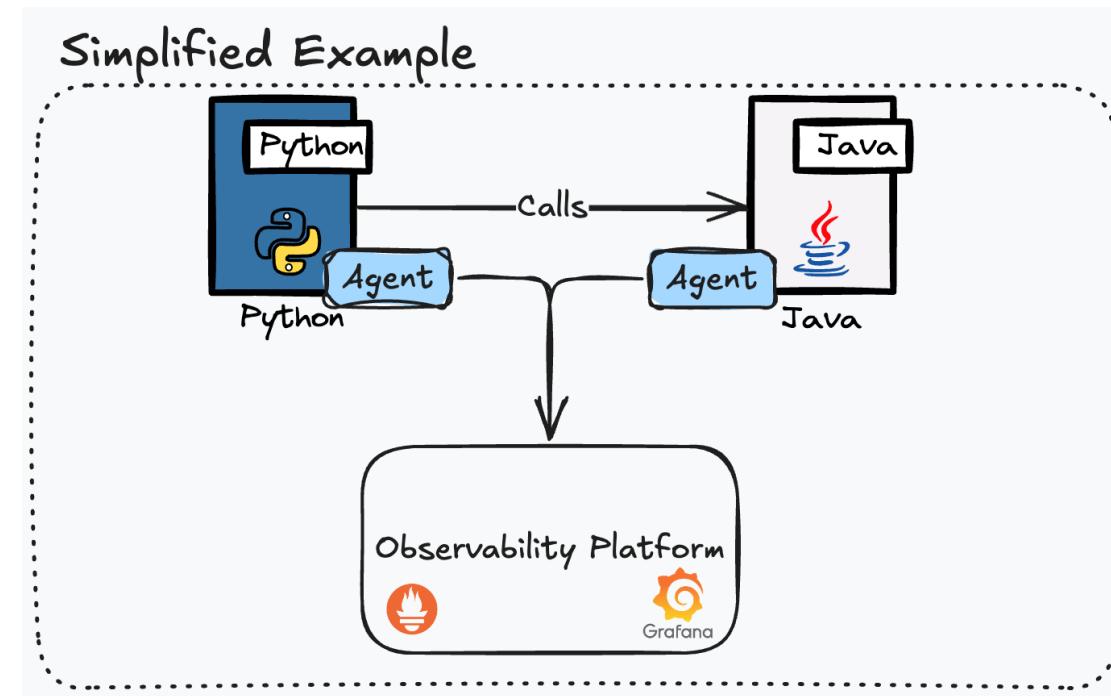
Let's get started

Instrumentation Approaches



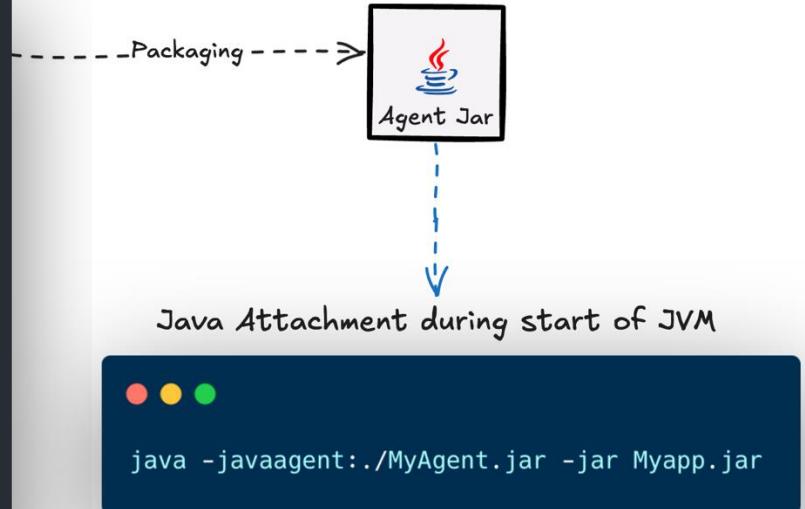
Automatic: Zero Code Instrumentation

Zero Code Instrumentation



Zero Code Instrumentation

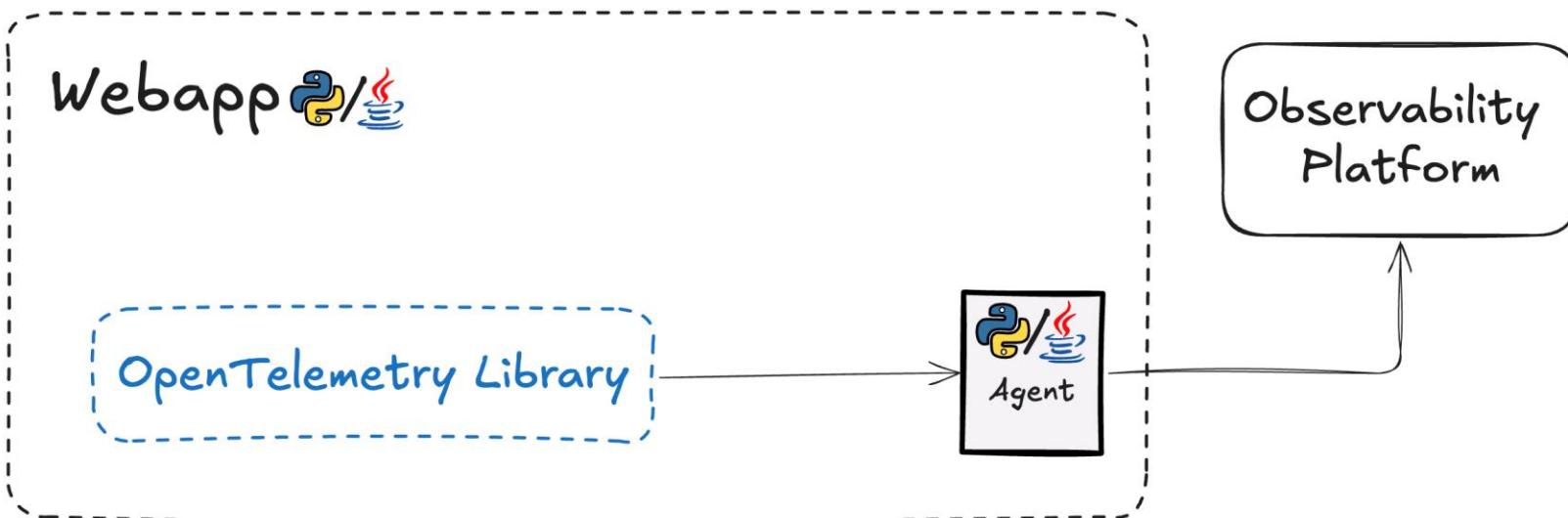
```
● ● ●  
public class MyAgent {  
    public static void premain(String agentArgs, Instrumentation inst) {  
        inst.addTransformer(new MyTransformer());  
    }  
}  
  
public class MyTransformer implements ClassFileTransformer {  
    @Override  
    public byte[] transform(  
        //  
    ) {  
        // Perform bytecode transformation  
        // ...  
  
        // Return the modified bytecode  
        return modifiedBytecode;  
    }  
    seed, []  
}
```



Let's jump into VS Code

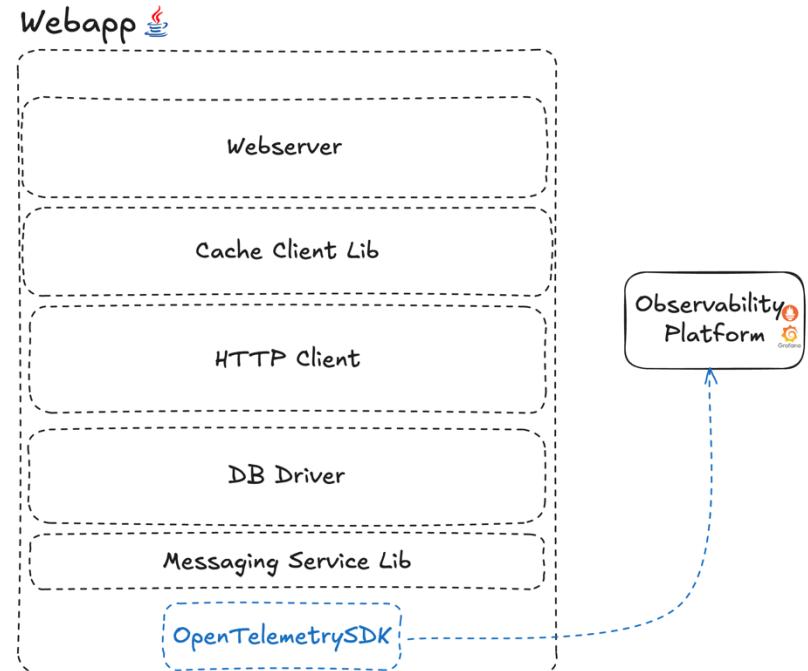
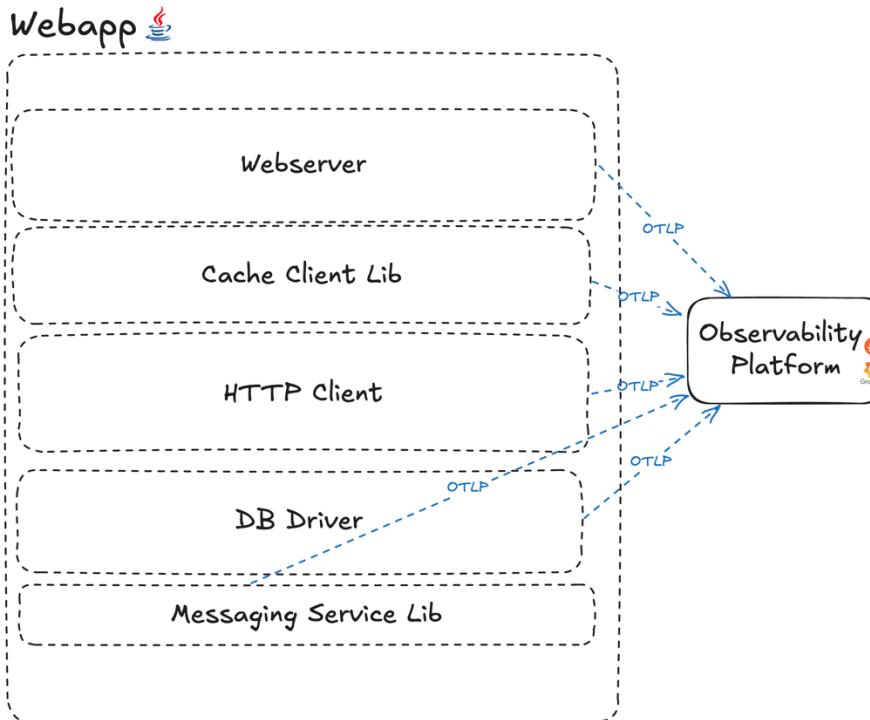
Automatic: Code-based/Libraries

Code-based Instrumentation

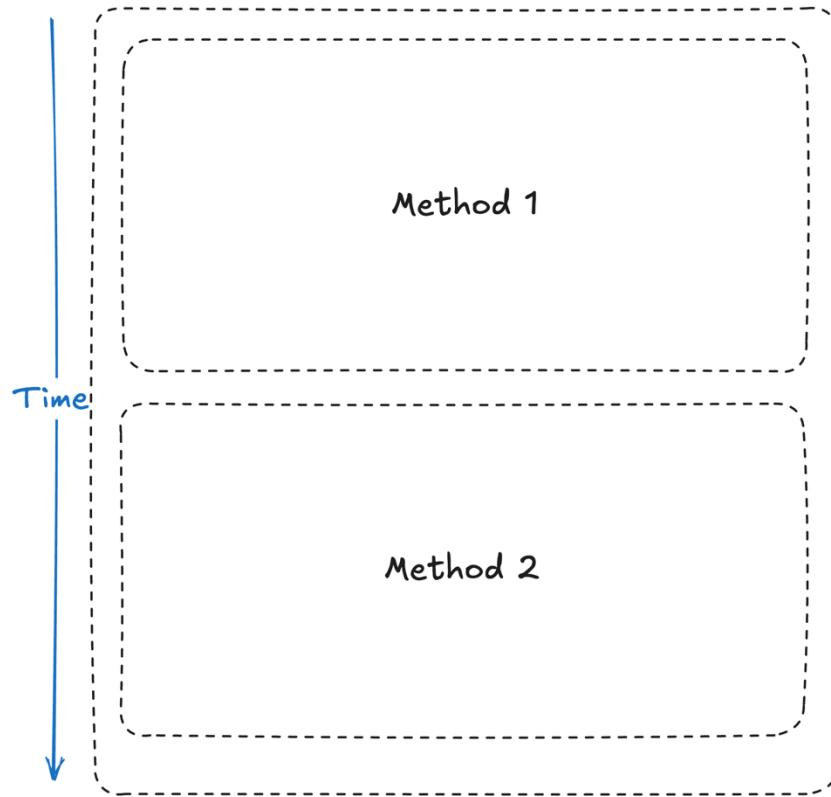


Manual Instrumentation

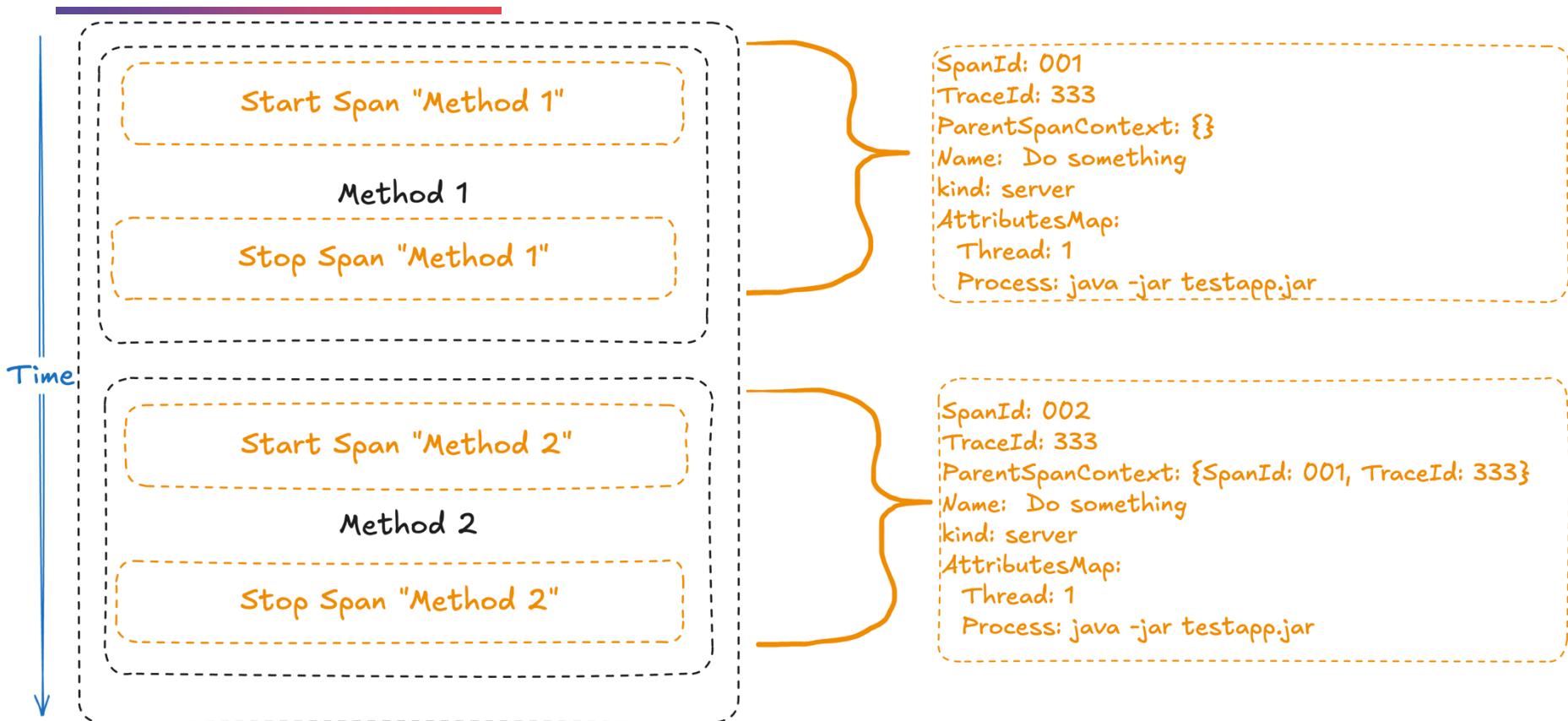
Manual Instrumentation



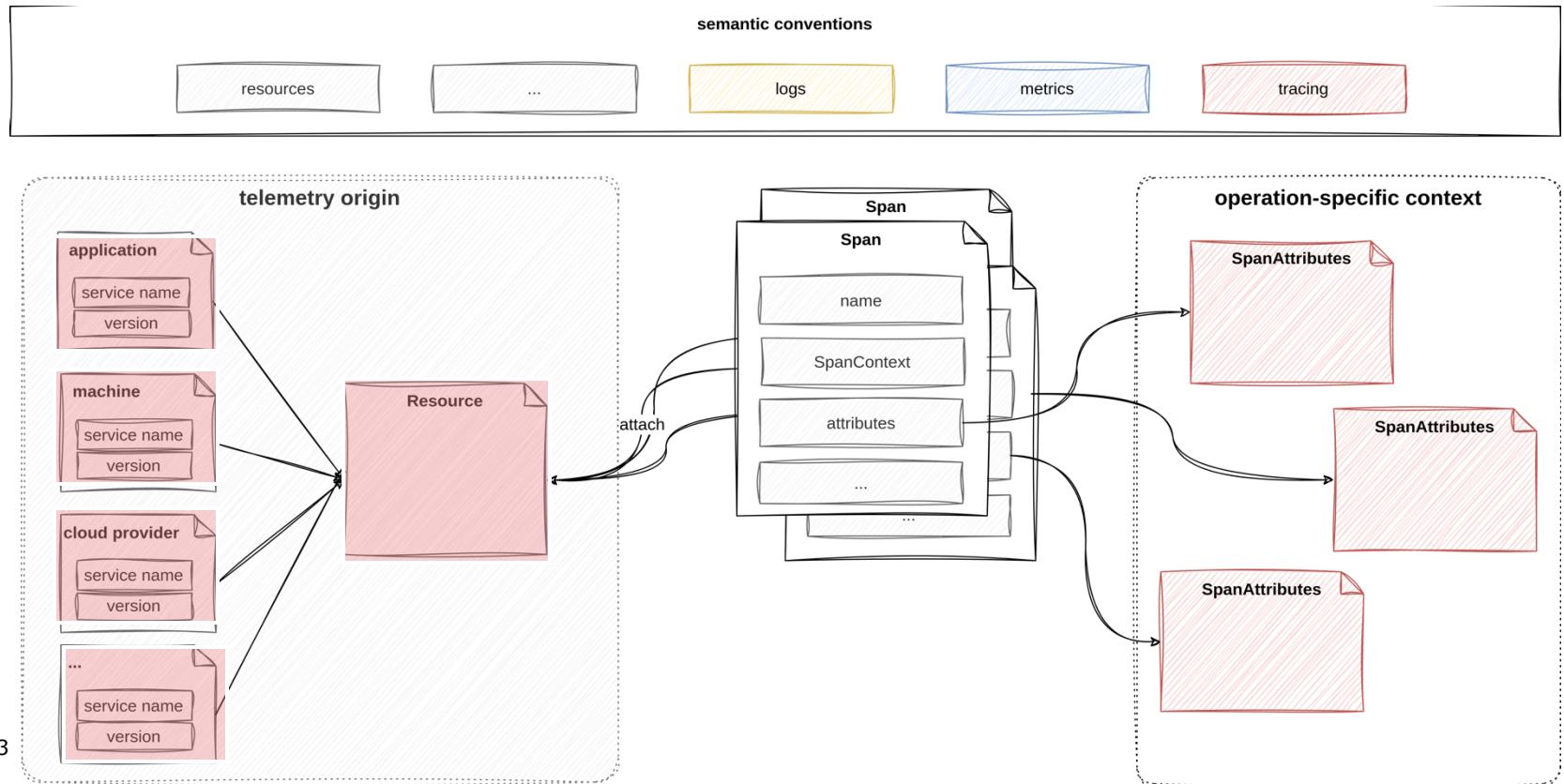
Manual Instrumentation (Tracing)



Manual Instrumentation (Tracing)



Manual Instrumentation: Context Enrichment



Manual Instrumentation Java Example

```
● ● ●

@PostMapping( "/todos/{todo}" )
    String addTodo(HttpServletRequest request, HttpServletResponse response, @PathVariable
String todo) {

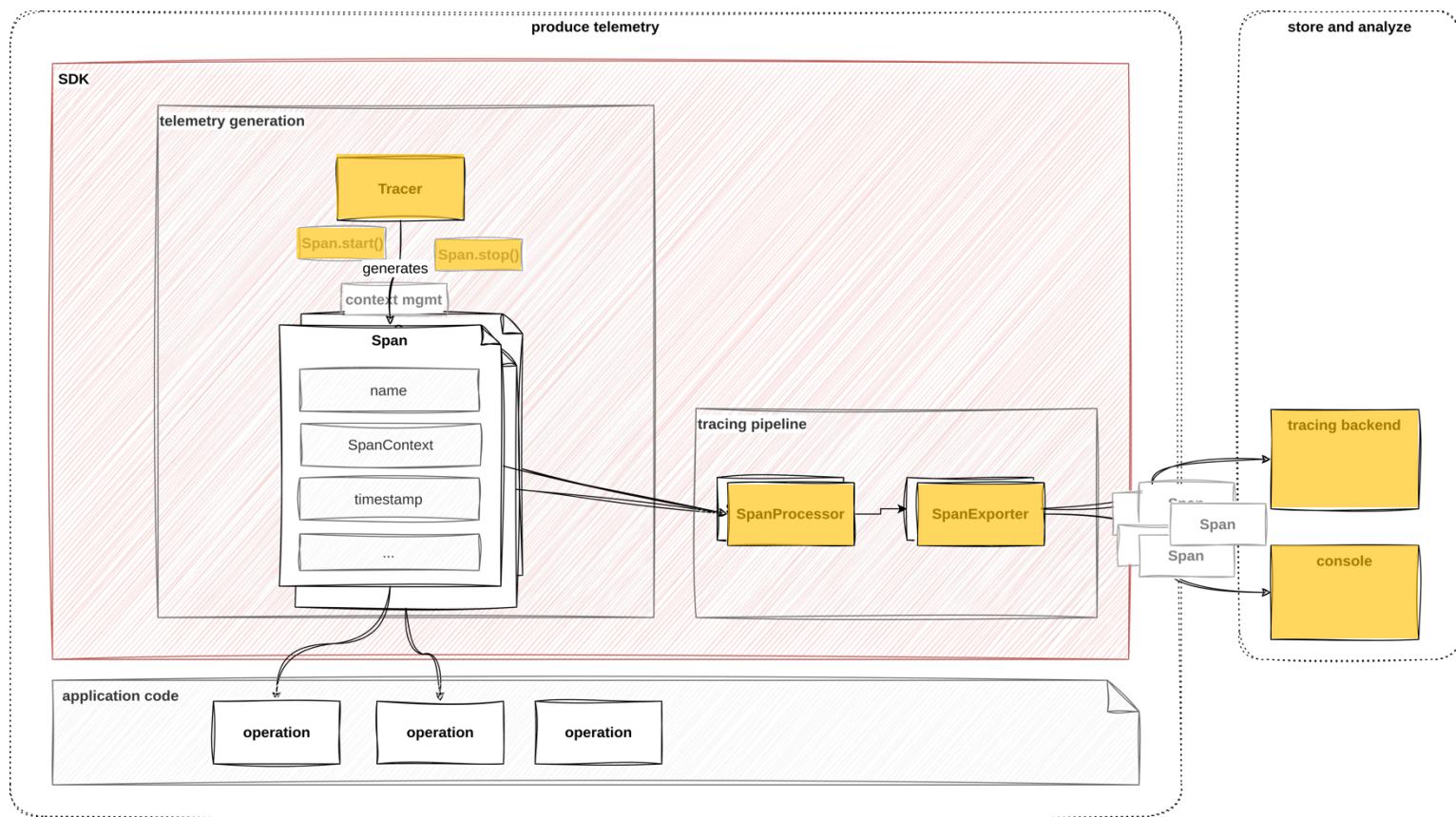
    logger.info("POST /todos/ " + todo.toString());

    Span span = tracer.spanBuilder("addTodo").setSpanKind(SpanKind.SERVER).startSpan();

    span.setAttribute("http.method", request.getMethod());
    span.setAttribute("http.url", request.getRequestURL().toString());
    span.setAttribute("client.address", request.getRemoteAddr());
    span.setAttribute("user.agent", request.getHeader("User-Agent"));
    this.someInternalMethod(todo);
    span.end();

    logger.info("Span.toString(): " + span.toString());
    return todo;
}
```

Manual Instrumentation Pipeline



Let's jump into VS Code

Start the tutorial on your own

<https://training.linuxfoundation.org/training/getting-started-with-opentelemetry-lfs148/>

The screenshot shows the Linux Foundation Education website with the following details:

- Course Title:** Getting Started with OpenTelemetry (LFS148)
- Description:** Learn to use OpenTelemetry to build and manage unified observability, skills increasingly important to IT developers and engineers career growth.
- Who Is It For:** This course is designed for software developers, DevOps engineers, site reliability engineers (SREs), and full-stack or backend developers looking to understand the basics of OpenTelemetry, including how to instrument code for traces, metrics, and logs; and use manual and automatic instrumentation, and
- What You'll Learn:** Understand the basics of OpenTelemetry, including how to instrument code for traces, metrics, and logs; and use manual and automatic instrumentation, and
- What It Prepares You For:** You'll be prepared to enhance application observability using OpenTelemetry, implement instrumentation, and
- Cost:** \$0
- Includes:**
 - Online, Self Paced
 - 8-10 Hours of Course Material
 - Discussion Forum
 - Unlimited Access to Online Materials
 - Digital Badge

A large QR code is located in the bottom right corner of the page.

Running in Codespaces

<https://github.com/lftraining/LFS148-code>

The screenshot shows the GitHub repository page for 'LFS148-code'. At the top, there's a banner asking for feedback on GitHub Codespaces. Below it, the repository details show 2 branches and 0 tags. The main area lists files like '.devcontainer', '.github/workflows', '.idx', '.vscode', 'assets', 'exercises', '.gitignore', and '.gitpod.yml'. A prominent feature is the 'Codespaces' sidebar, which displays a list of workspaces in the cloud. One workspace, 'refactored couscous', is marked as 'Active' and 'last week'. Another workspace, 'tobiangerstein', is listed as 'last month'. A note at the bottom of the sidebar states: 'Codespace usage for this repository is paid for by tobiangerstein.' On the right side of the repository page, there's a 'Watch' button with 6 notifications.



Get in contact with us



- **Tobias Angerstein**
- Senior Observability Consultant
- Novatec Consulting
- angerstone (LI)



- **Tiffany Jernigan**
- Developer Advocate, CNCF Ambassador
- tiffanyfay.dev 
- [tiffanyfayj \(LI/X\)](https://tiffanyfayj.com)



Feedback

