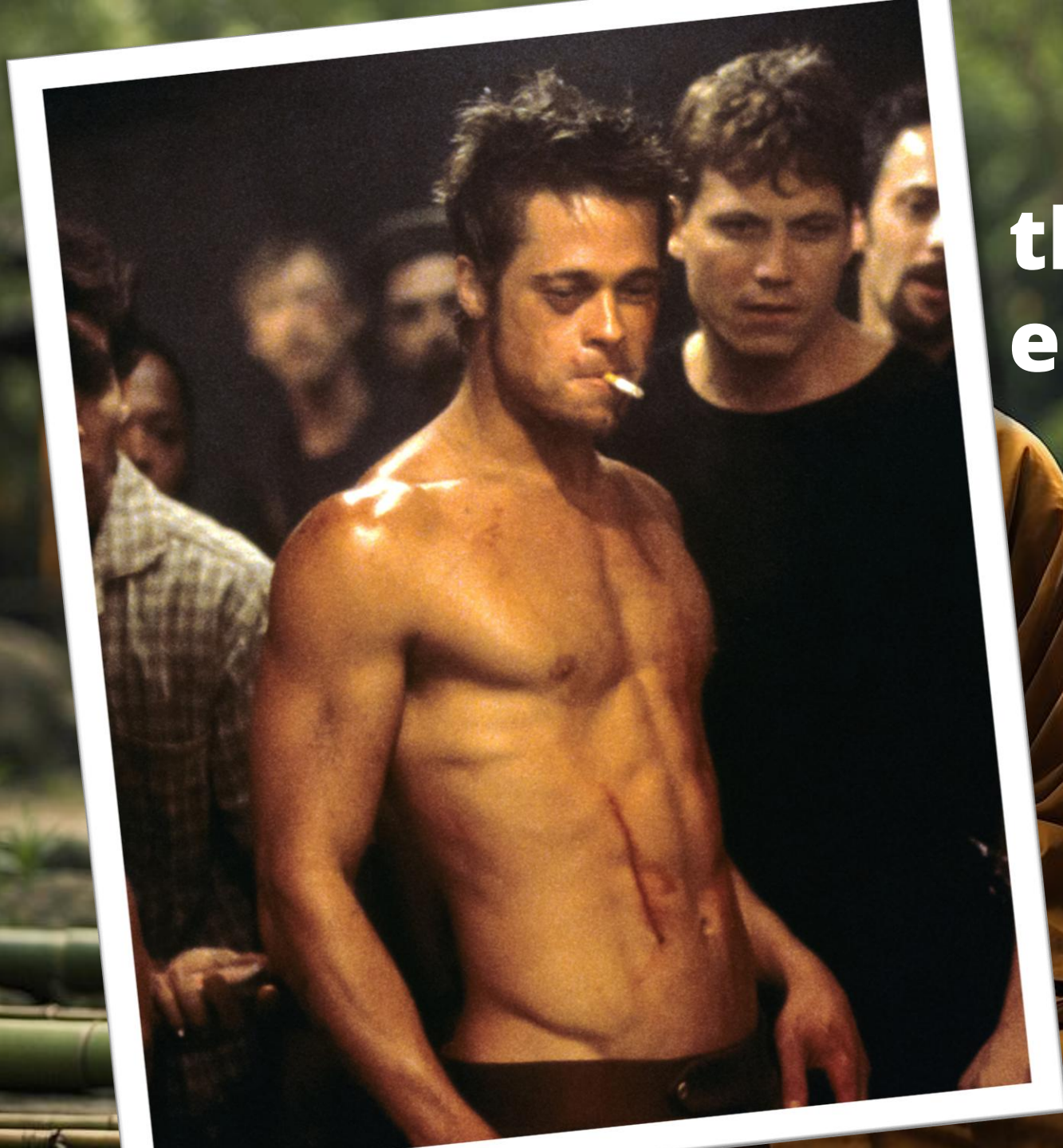




Java Modernization

**Breathe New Life into Your Codebase
Without Breaking the Bank**



**the things you own
end up owning you**

Technical Debt and/or Skipped Updates







Modernization ≠ Reinvention



Java™
Champions



IBM Champion



Oracle ACE
Pro



JAKARTA™ EE
AMBASSADORS

Richard Fichtner

20+ years full-stack mostly Java

XDEV Software

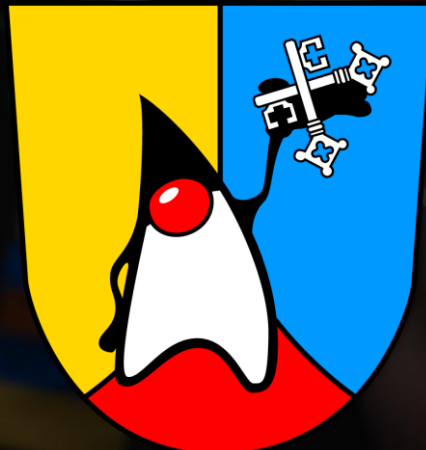
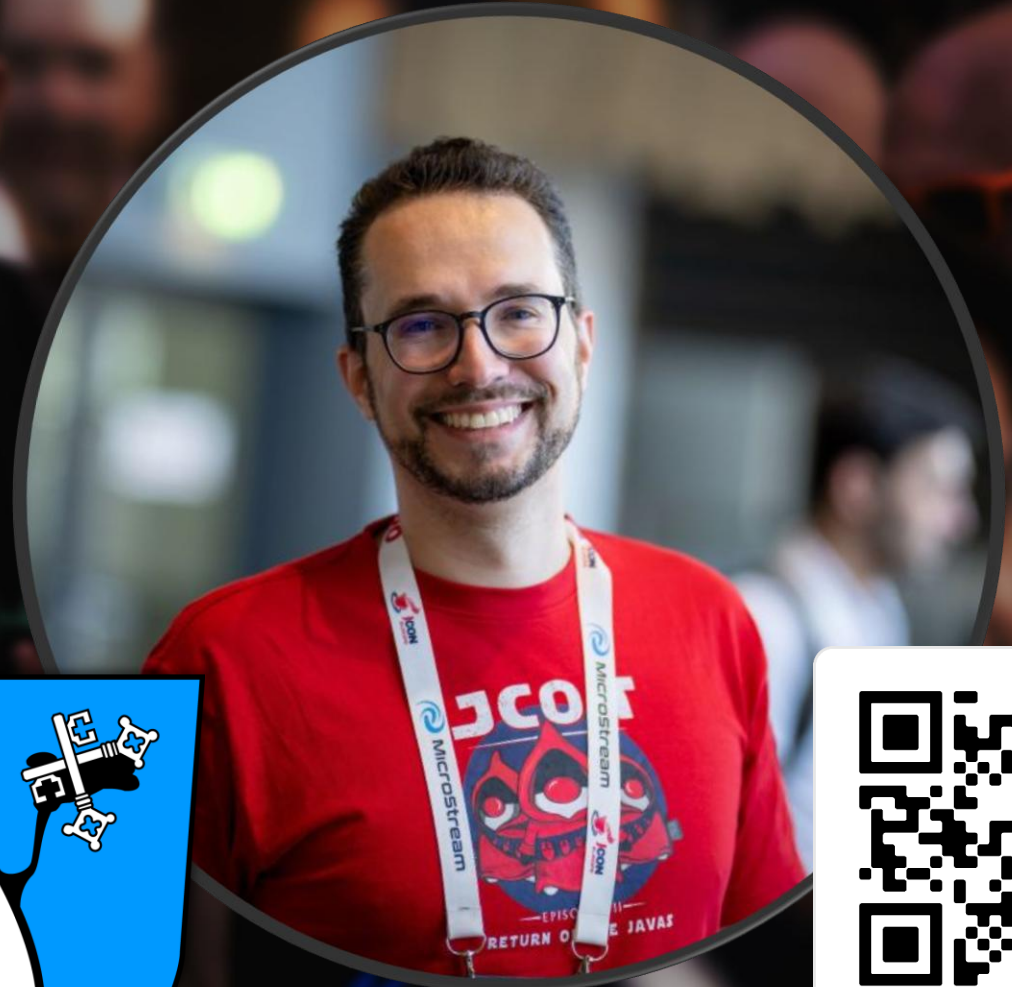
<https://xdev.software>

Projects

JUG Oberpfalz

JCON Co-Organisator

Digital Craftsmanship N.Opf.



scan me



Java™

we make your
Java projects fly



AIRBUS

Red Bull



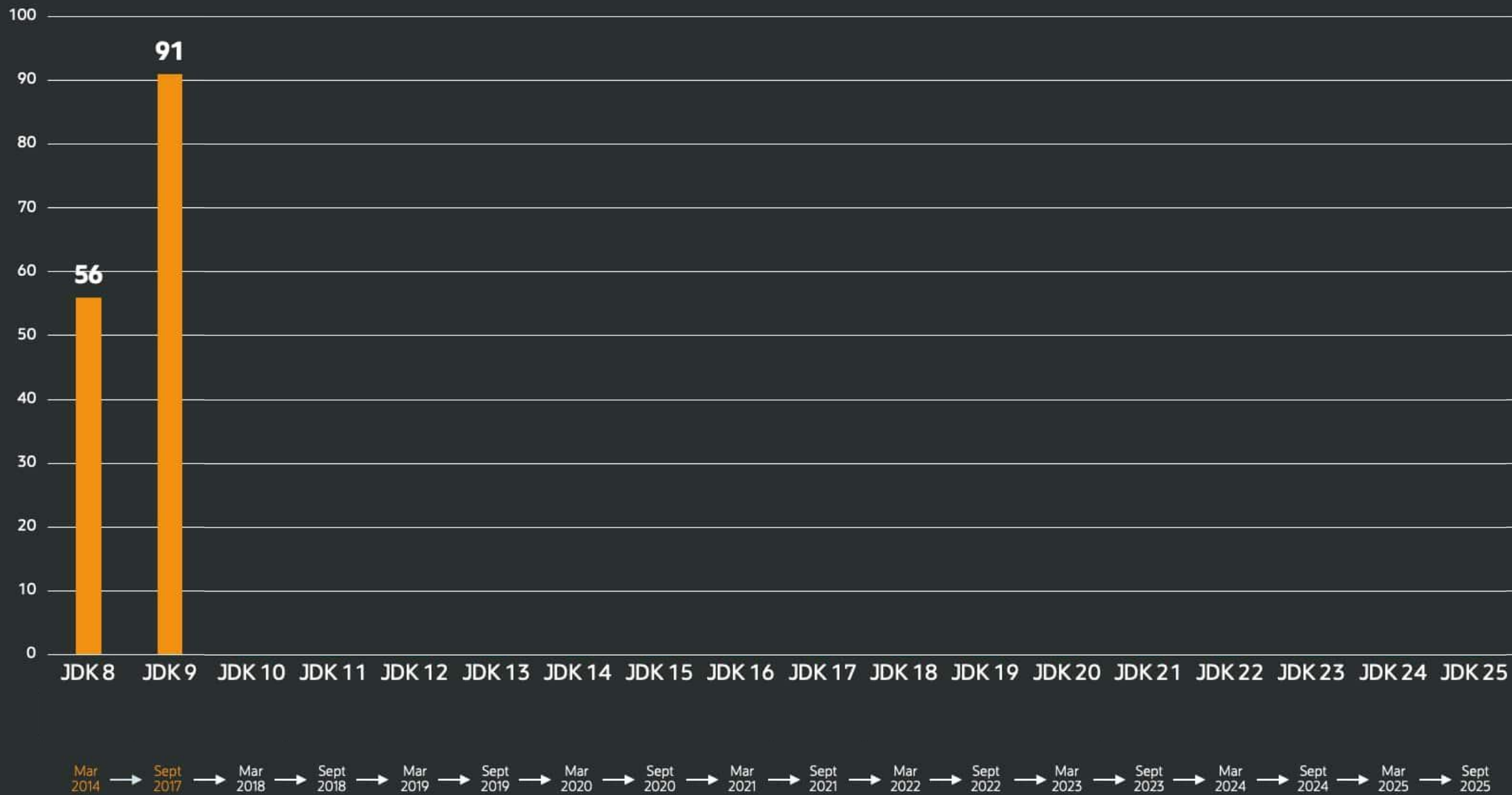
OKI

Jägermeister

EMPORIO  ARMANI

The evolution of Java







<https://javaalmanac.io>



Don't touch this

No code changes required *

azul



OpenJDK



GraalVM



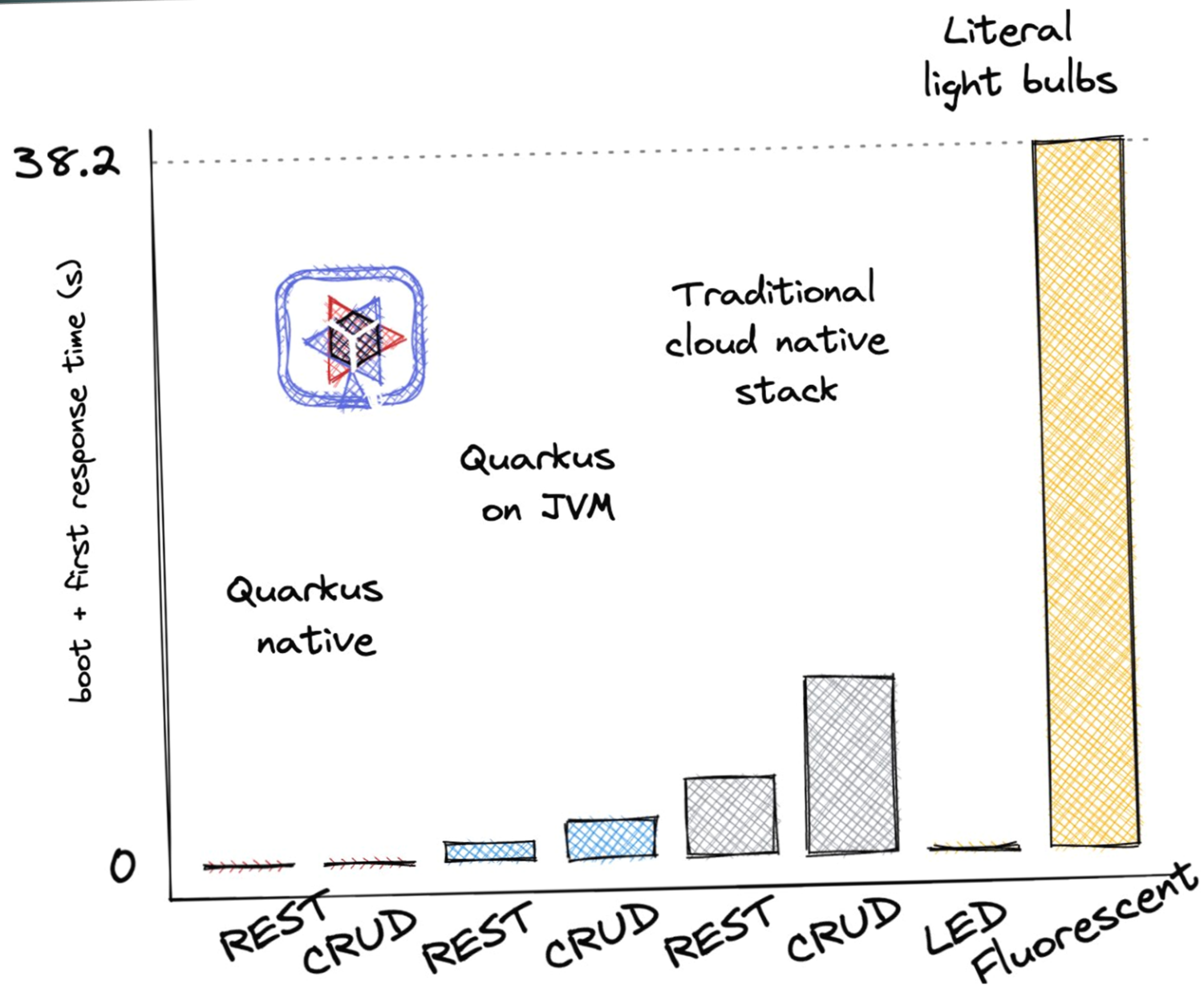


<https://whichjdk.com>



Java Startup

<https://openjdk.org/projects/leyden/>



Solution for Start-up: GraalVM - Native Image

- Small Footprint

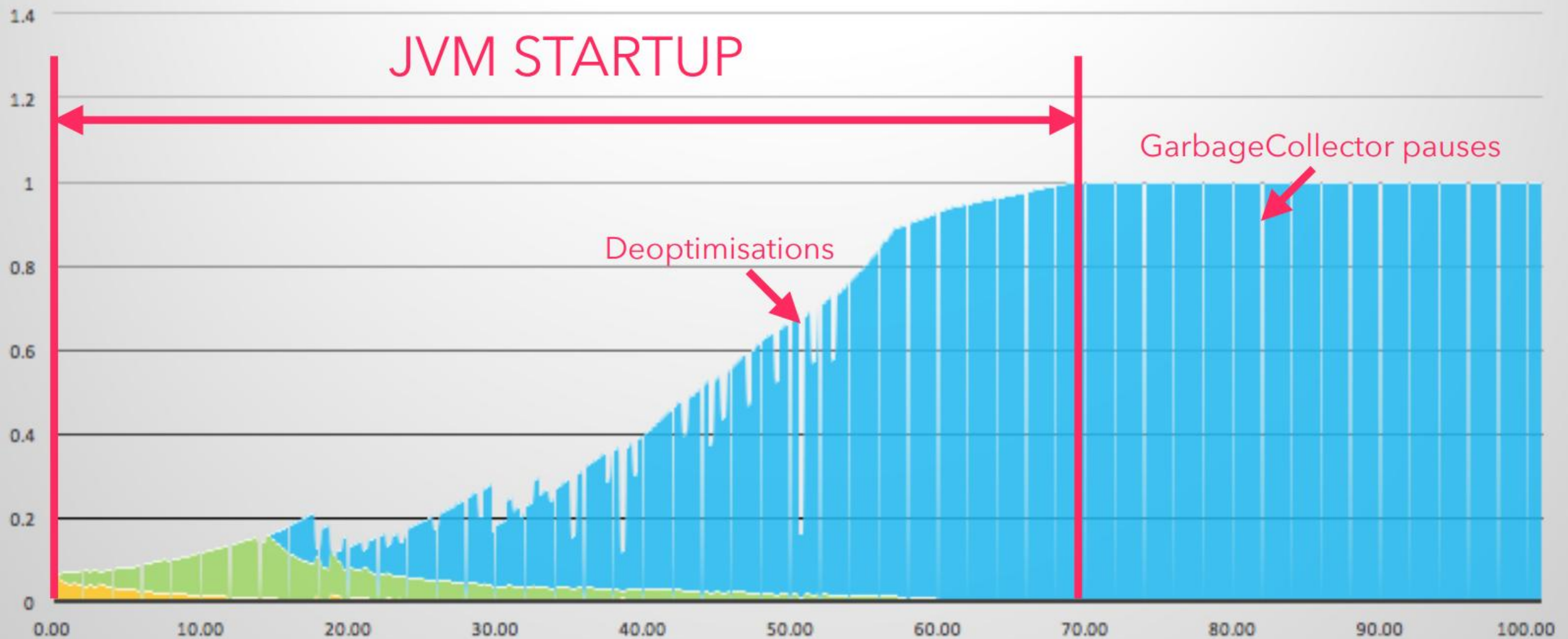
Ahead-of-time compiled applications use only a fraction of the resources required by the JVM, which means they cost less to run and improve utilization.

- Fast Startup

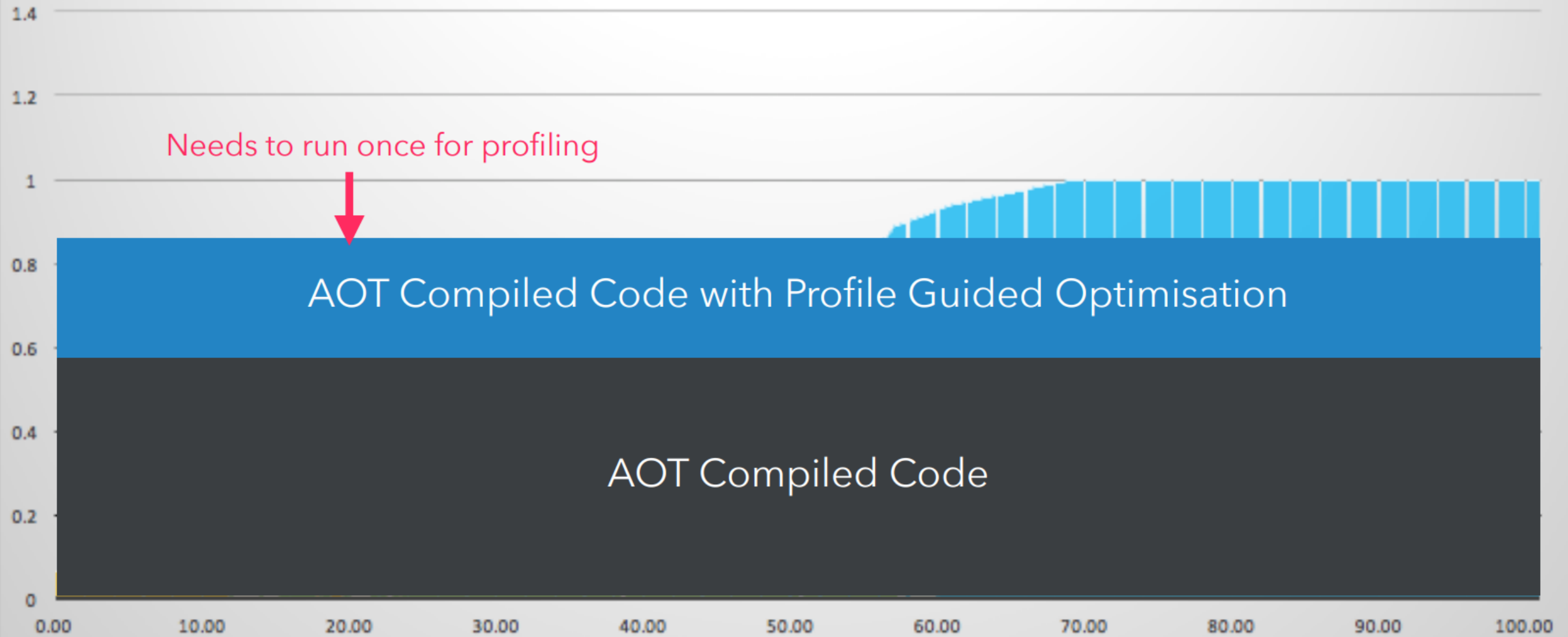
Ahead-of-time compiled applications start in milliseconds and deliver peak performance with no warmup.



Speed



Speed

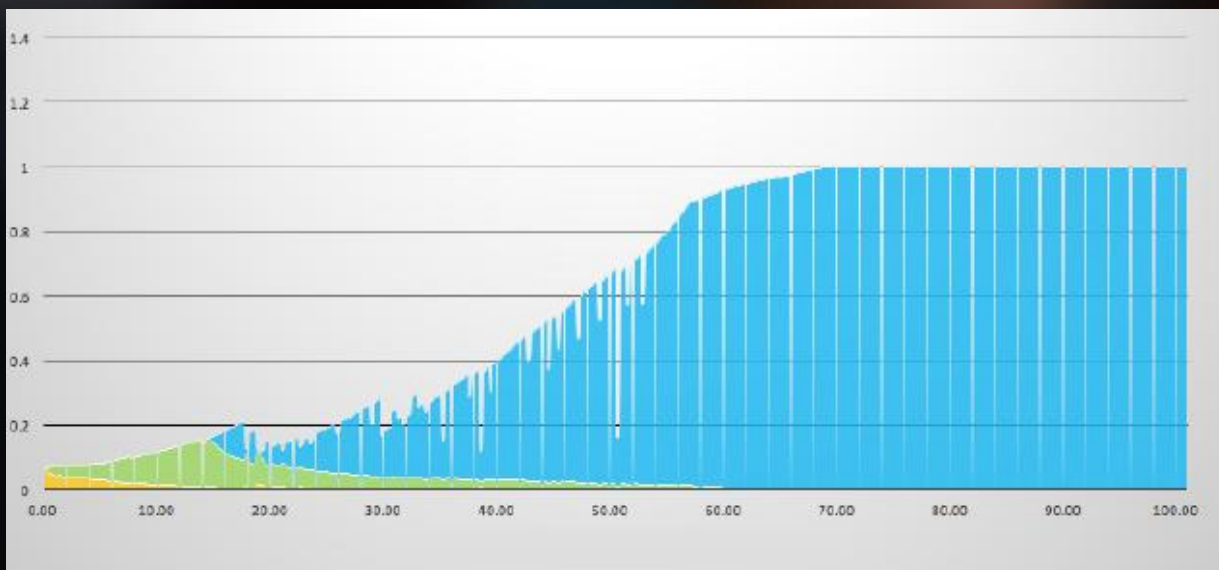


Solution for Start-up: CRIU

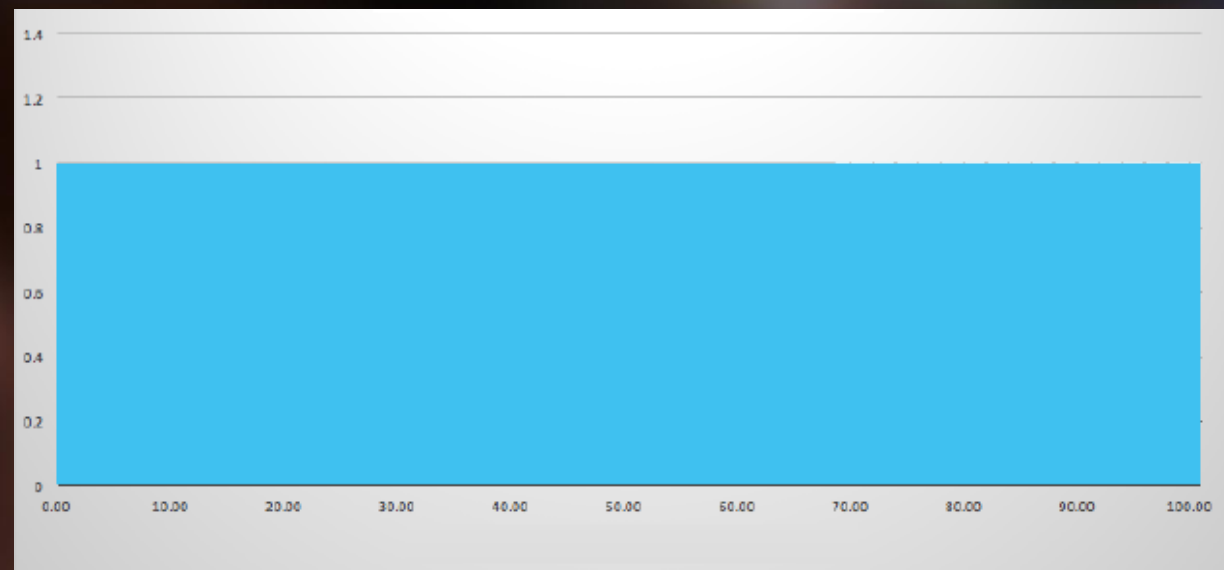
- Real (warm) JVM with all the features
 - Reflection, Monitoring, JIT, ...
- Linux only snapshot / restore
- CRaC
(Coordinated Restore at Checkpoint)
<https://openjdk.org/projects/crac/>



First run



Second run



Java Garbage Collection

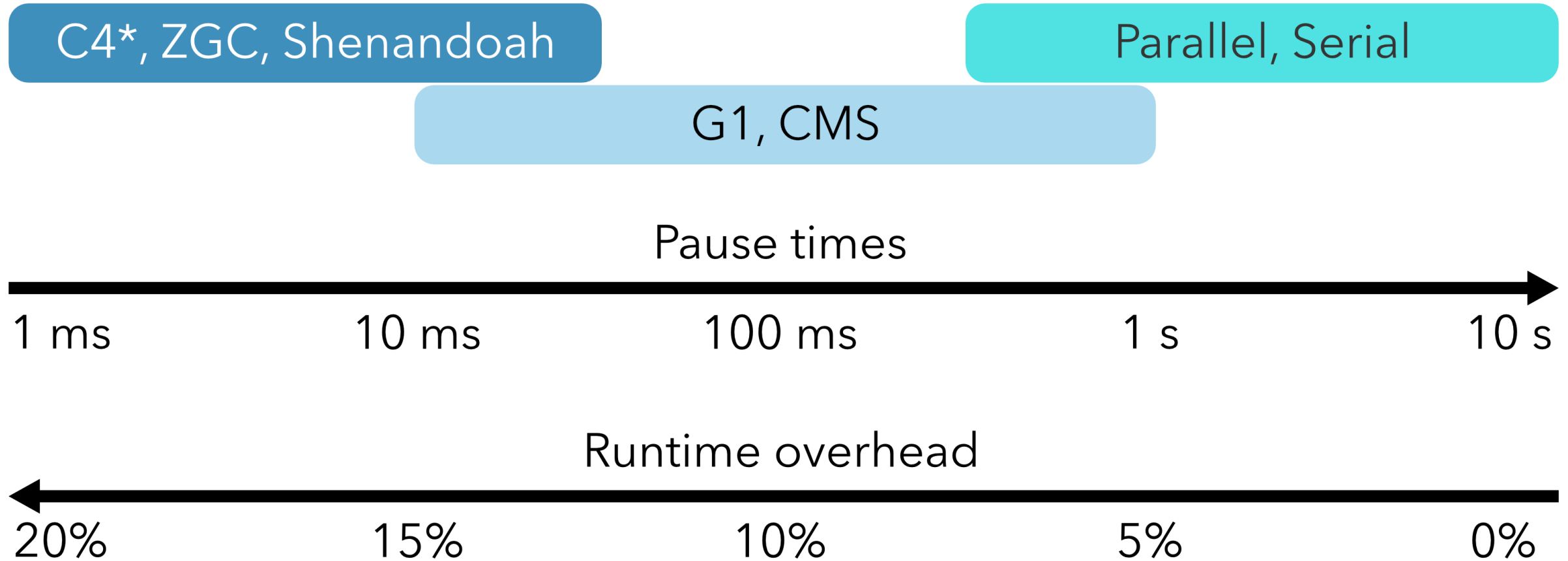
Special Thanks to Gerrit Grunwald

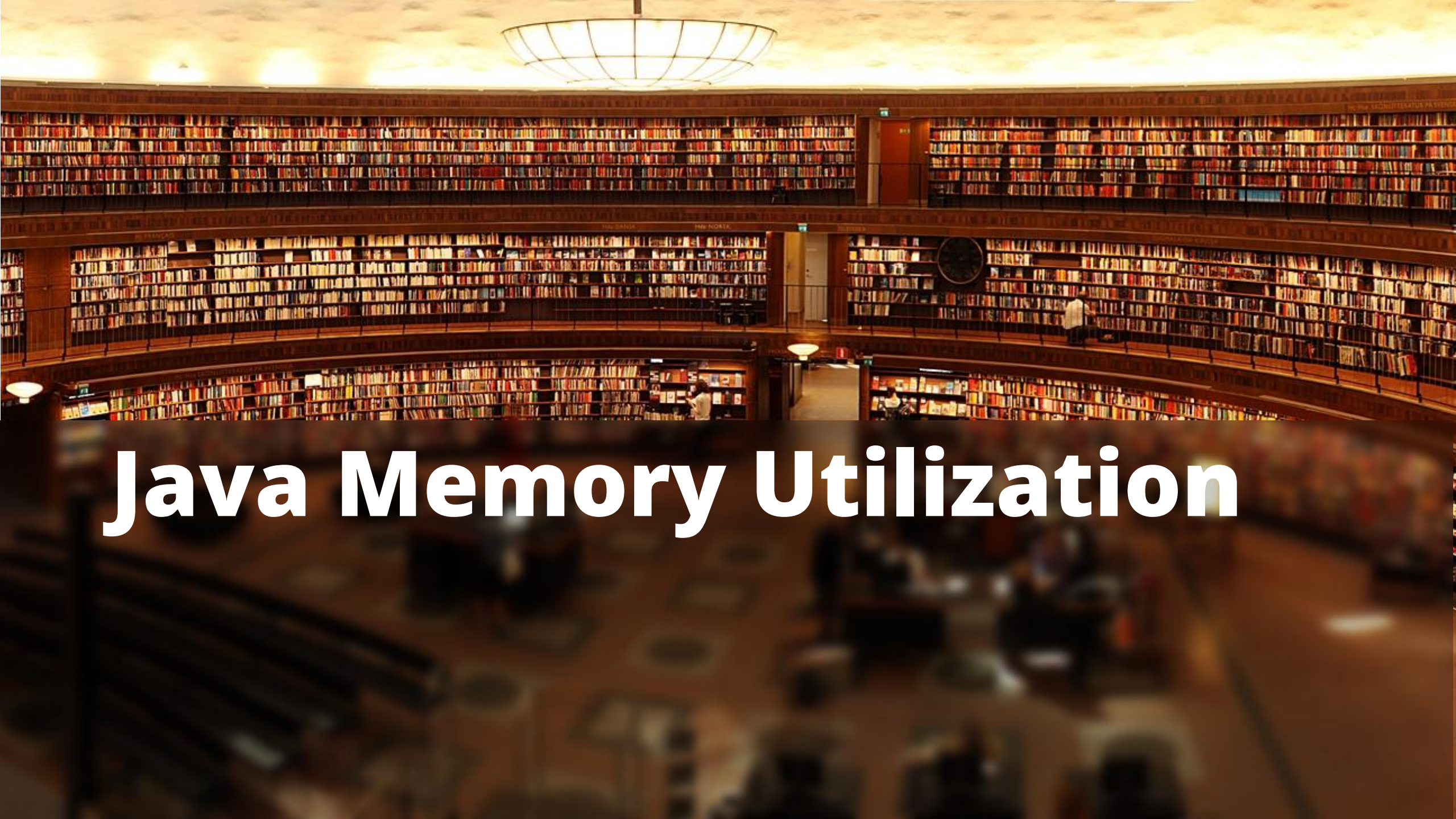


Garbage Collectors Comparison

	Serial GC 	Parallel GC 	CMS GC 	G1 	Epsilon 	Shenandoah 	ZGC 	C4 
Availability	ALL JDK's	ALL JDK's	JDK 1.4-13	JDK 7u4+	JDK 11+	JDK 11.0.9+ / 24+	JDK15 / 21+	Azul Prime
Parallel	NO	YES	YES	YES	-	YES	YES	YES
Concurrent	NO	NO	PARTIALLY	PARTIALLY	-	FULLY	FULLY	FULLY
Generational	YES	YES	YES	YES	-	NO / YES	NO / YES	YES
Heap Size	SMALL - MEDIUM	MEDIUM - LARGE	MEDIUM - LARGE	MEDIUM - LARGE	-	LARGE	VERY LARGE	VERY LARGE
Pause Times	LONGER	MODERATE	MODERATE	SHORT - MEDIUM	-	VERY SHORT (<10ms)	VERY SHORT (<1ms)	VERY SHORT (<1ms)
Throughput	LOW	HIGH	MODERATE	HIGH	-	VERY HIGH	VERY HIGH	VERY HIGH
Latency	HIGHER	LOWER	MODERATE	LOWER	-	VERY LOW	VERY LOW	VERY LOW
Performance	LOWER	HIGHER	MODERATE	HIGHER	VERY HIGH	VERY HIGH	VERY HIGH	VERY HIGH
CPU Overhead	LOW	LOWER	MODERATE	MODERATE	VERY LOW	LOW - MODERATE	MODERATE - HIGH	MODERATE - HIGH
Tail latency	HIGH	HIGH	HIGH	HIGH	-	MODERATE	LOW	LOW

Garbage Collectors Comparison

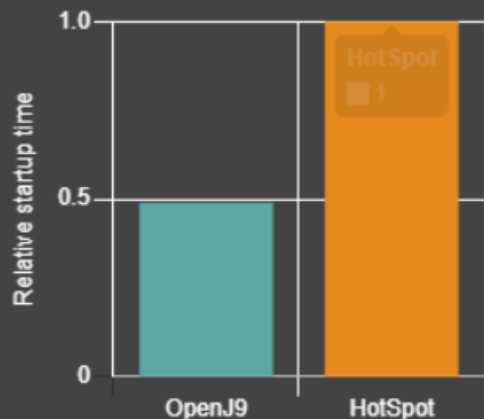




Java Memory Utilization

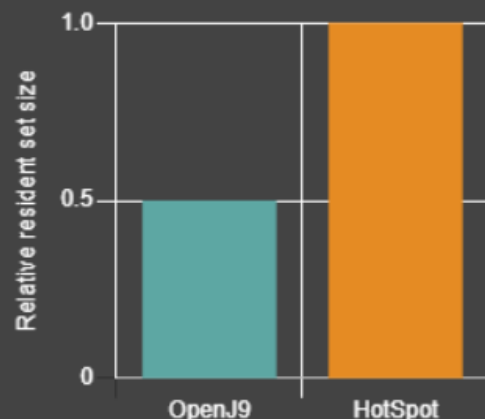
OpenJ9

51% faster startup time



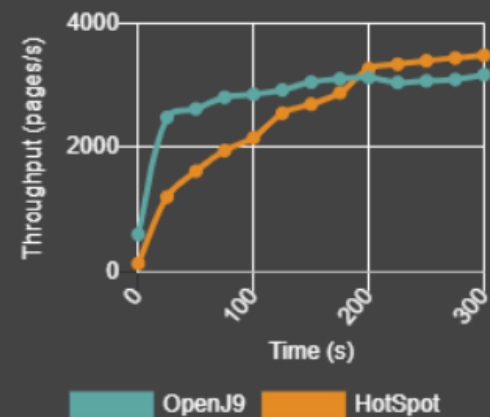
By using shared classes cache and AOT technology, OpenJ9 starts in roughly half the time it takes HotSpot.

50% smaller footprint after startup



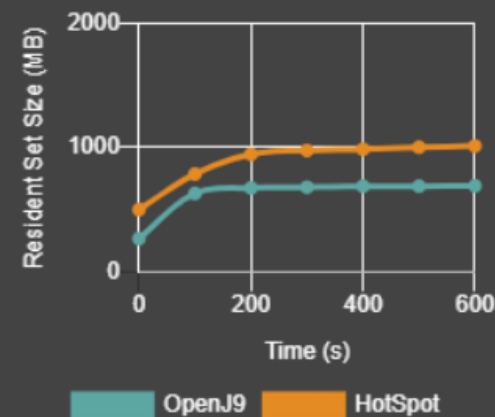
After startup, the OpenJ9 footprint is half the size of HotSpot, which makes it ideal for cloud workloads.

Faster ramp-up time in the cloud



OpenJ9 reaches peak throughput much faster than HotSpot making it especially suitable for running short-lived applications.

33% smaller footprint during load



Consistent with the footprint results after startup, the OpenJ9 footprint remains much smaller than HotSpot when load is applied.



Let's mix things up

Code changes required



Virtual Threads

Java 21

JEP 444

<https://openjdk.org/jeps/444>

**Virtual Threads are best suited for
code that involves blocking**
logging, file I/O, accessing databases, network calls

That would cover nearly all Java business apps.

Benefits of Virtual Threads

Keep using the Imperative Programming Model

Easier to write code

Easier to understand code

Easier to debug code

Separate technical concerns from business logic

No more need for external APIs (for this aspect)

Compatible with existing Thread API

```
public static void main(String[] args) throws Exception{

    var executorService = Executors.newFixedThreadPool(100);

    for (var i = 0; i < 1_000; i++) {
        var index = i;
        executorService.submit(() -> doSomeIO(index));
    }

    executorService.shutdown();
    executorService.awaitTermination(10, TimeUnit.SECONDS);
}
```

Important Features

Records

JEP 359

Sealed Classes

JEP 409

Stream Gatherers

JEP 473 (2nd Preview)

Text Blocks

JEP 378

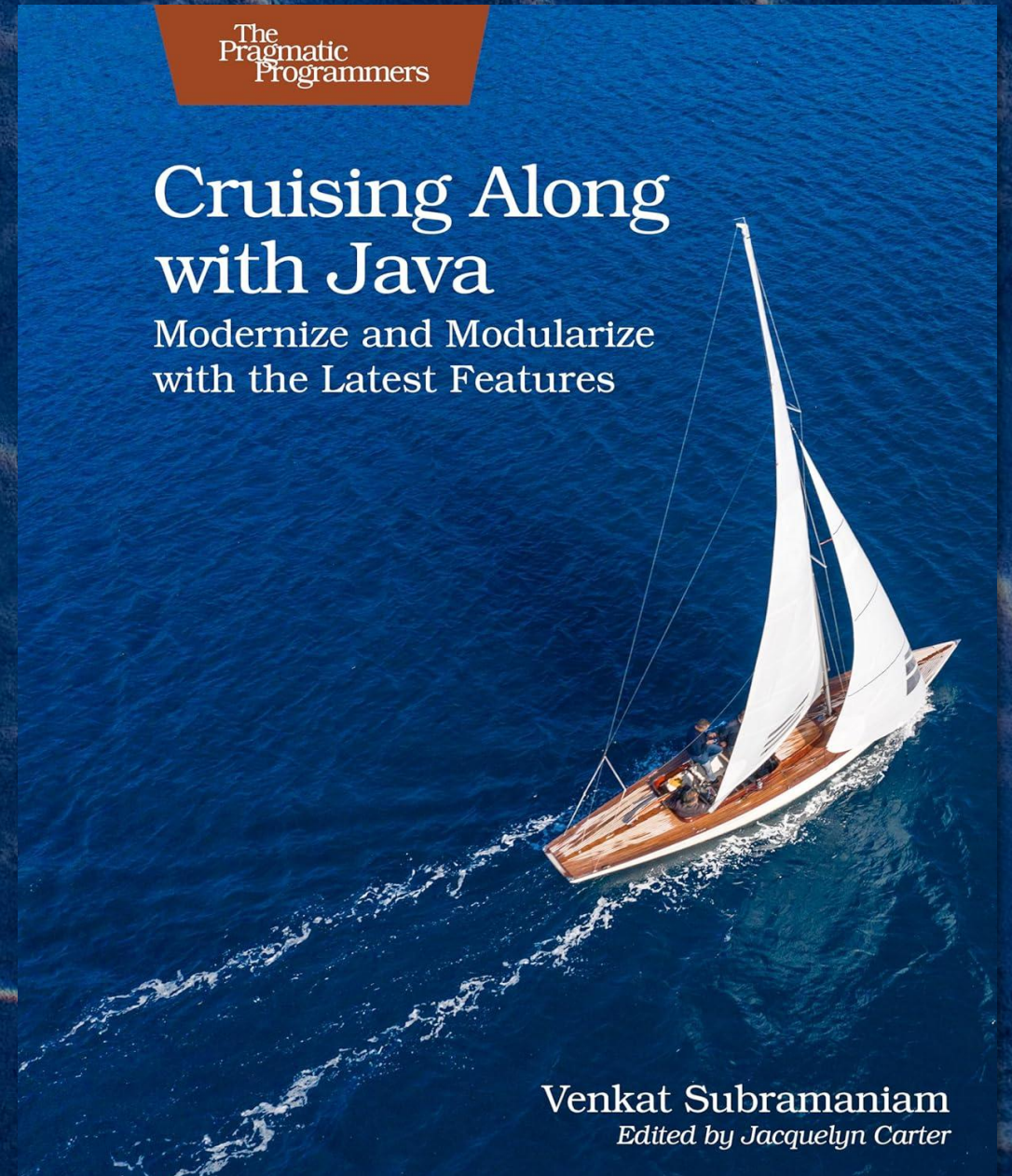
Pattern Matching

JEP 394, 433

Switch Expressions

JEP 361

Book Recommendation





**People First.
Technology Second.**

JAVAPRO



Software (Stack) Modernization

To developers, this can feel like:

- **A dismissal** of everything they've built and learned.
- **Anxiety** over new tools, with no space to fail safely.
- **Imposter syndrome**, even for your most experienced team members.

And worst of all? It makes your best developers feel **replaceable**, not valued.



**No system starts out as a
legacy system — it
becomes one by surviving.**



**You don't burn down your
house just because the
kitchen is outdated**



Upgrade the Development Experience First

Improve the DX (developer experience):

- Faster builds (e.g., better Maven setups)
- Local Docker/Container dev environments
- Clear README.md, or mvn goals for common tasks
- Have a buddy



Stay up-to-date



Go to conferences (you are in the right place)

Read blogs like <https://myfear.substack.com/>

Check out <https://inside.java/> and <https://dev.java/>

Go to Meetups / JUGs



JUGs around the world

<https://dev.java/community/jugs/>



JUGs im DACH Raum



Mitglieder des iJUG

- | | |
|----------------------------------|---------------------------------|
| 01 BED-Con e.V. | 23 JUG Karlsruhe |
| 02 Clojure User Group Düsseldorf | 24 JUG Köln |
| 03 DOAG e.V. | 25 Kotlin User Group Düsseldorf |
| 04 EuregJUG Maas-Rhine | 26 JUG Mainz |
| 05 JUG Augsburg | 27 JUG Mannheim |
| 06 JUG Berlin-Brandenburg | 28 JUG München |
| 07 JUG Bremen | 29 JUG Münster |
| 08 JUG Bielefeld | 30 JUG Oberland |
| 09 JUG Bonn | 31 JUG Ostfalen |
| 10 JUG Darmstadt | 32 JUG Paderborn |
| 11 JUG Deutschland e.V. | 33 JUG Saxony |
| 12 JUG Dortmund | 34 JUG Stuttgart e.V. |
| 13 JUG Düsseldorf rheinjug | 35 JUG Switzerland |
| 14 JUG Erlangen-Nürnberg | 36 JSUG |
| 15 JUG Freiburg | 37 Lightweight JUG München |
| 16 JUG Goldstadt | 38 SUG Deutschland e.V. |
| 17 JUG Görlitz | 39 JUG Thüringen |
| 18 JUG Hannover | 40 JUG Saarland |
| 19 JUG Hessen | 41 JUG Duisburg |
| 20 JUG HH | 42 JUG Frankfurt |
| 21 JUG Ingolstadt e.V. | 43 JUG Oberpfalz |
| 22 JUG Kaiserslautern | |





OpenRewrite

 Moderne®

JUnit 4 to 5 migration:

“assertThrows” call added to the test’s method body

No longer define an expected exception on the ‘@Test’ annotation

```
@Test(expected = NoSuchBeanDefinitionException.class)
```

```
@Test
```

Auto-refactoring relocates the original contents into a lambda argument in an ‘assertThrows’ call and formats per existing code style

```
public void testLocalSchedulerEnabled() {  
    assertFalse(context.getEnvironment().containsProperty("kubernetes_service_host"));  
    assertFalse(CloudPlatform.CLOUD_FOUNDRY.isActive(context.getEnvironment()));  
    context.getBean(Scheduler.class);  
    assertThrows(NoSuchBeanDefinitionException.class, () -> {  
        assertFalse(context.getEnvironment().containsProperty("kubernetes_service_host"));  
        assertFalse(CloudPlatform.CLOUD_FOUNDRY.isActive(context.getEnvironment()));  
        context.getBean(Scheduler.class);  
    });  
}
```

Key concepts defined

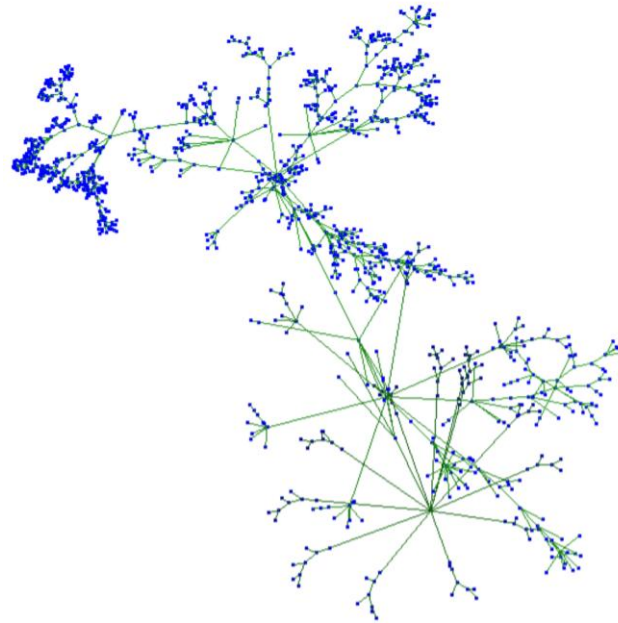
Visitor Pattern	Recipes	Lossless Semantic Tree (LST)
A traversal strategy used by recipes to systematically visit nodes in the lossless semantic tree, enabling targeted analysis and modification based on node type and context.	A deterministic program that defines <i>what</i> to find and <i>how</i> to change it in code, using language-aware search and transformation logic to automate research, upgrades, remediations, and refactorings.	A structured, language-specific representation of code that preserves both syntax and semantics without losing original formatting or comments — enabling precise analysis and transformation at scale.

How does the LST compare to other code representations?

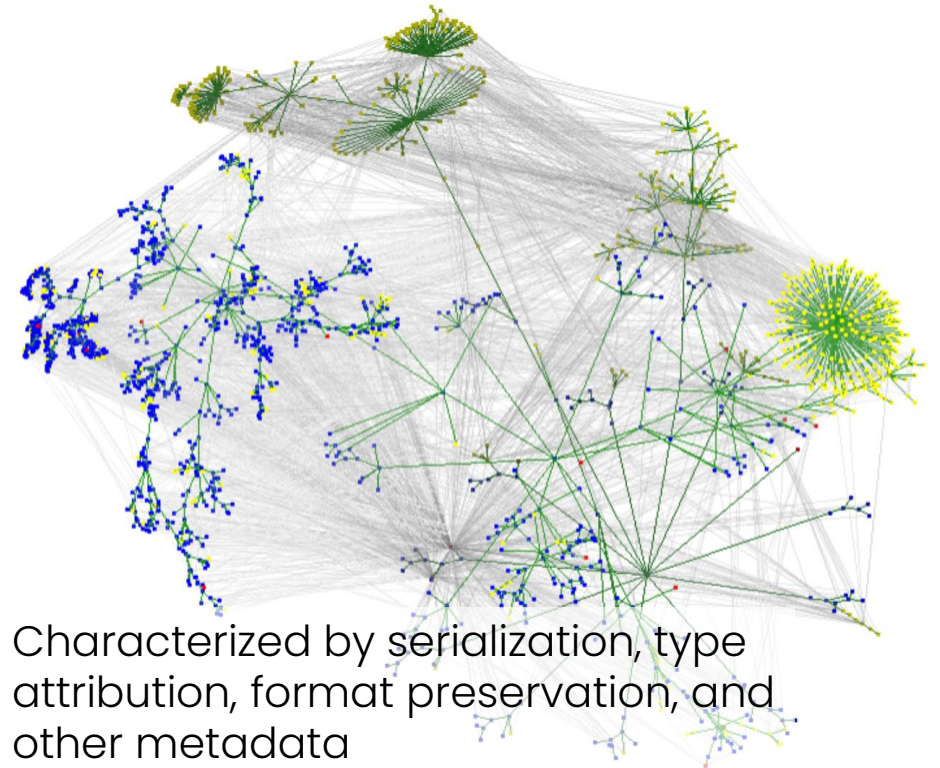
Text-based search

```
void m() {  
    if(c > 2) {  
        x = c;  
    }  
    while(c < 10) {  
        x += p();  
        c++;  
    }  
}
```

Abstract Syntax Tree (AST)



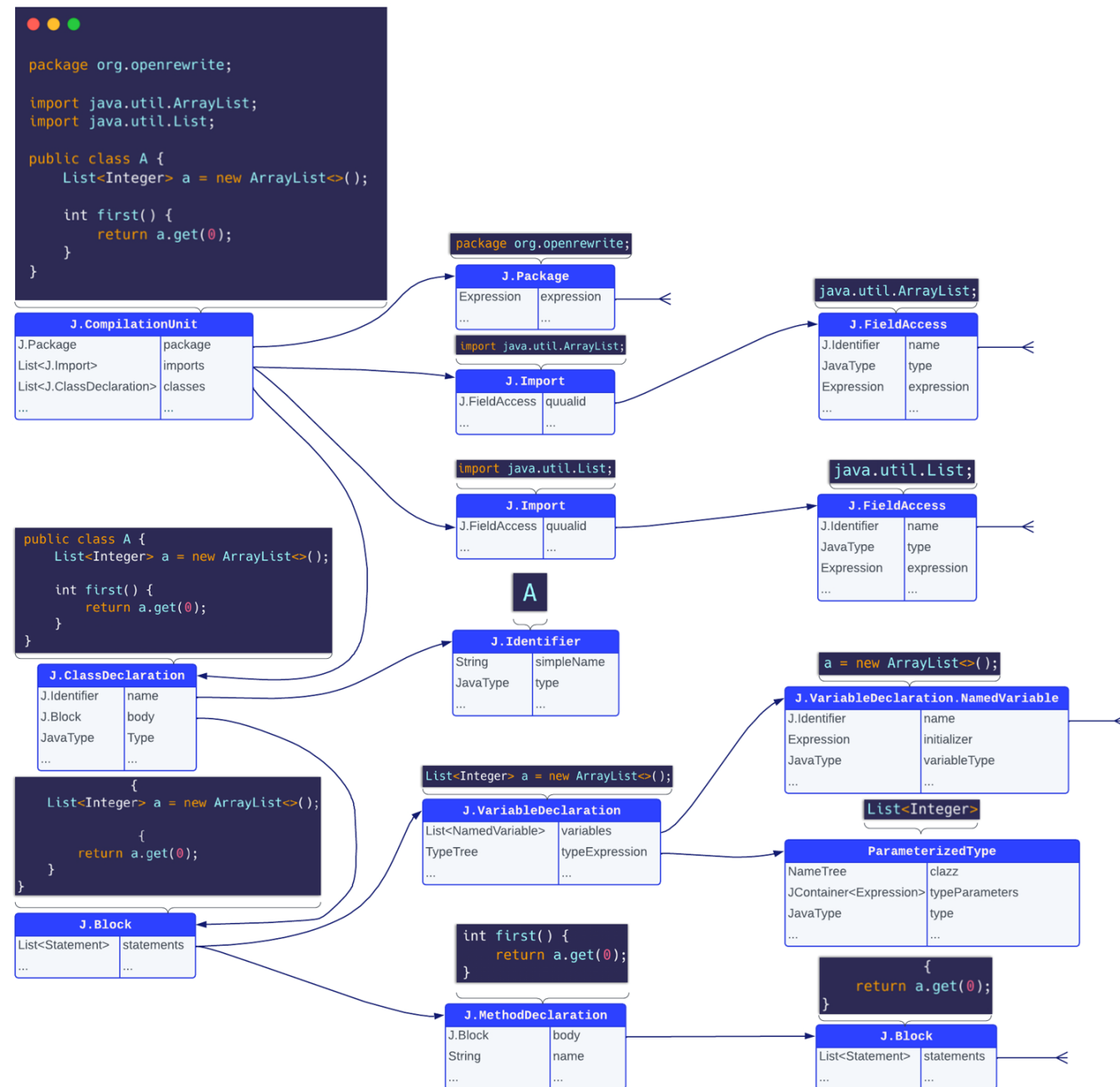
Lossless Semantic Tree (LST)



Characterized by serialization, type attribution, format preservation, and other metadata

+ another 600,000 nodes that didn't fit in this graphic

LST represents the depth of your code



```
package org.openrewrite.test;

import java.util.ArrayList;
import java.util.List;

public class A {
    List<Integer> a = new ArrayList<>();

    int first() {
        return a.get(0);
    }
}
```

J.CompilationUnit	
J.Package	package
List<J.Import>	imports
List<J.ClassDeclaration>	classes
...	...

```
public class A {
    List<Integer> a = new ArrayList<>();

    int first() {
        return a.get(0);
    }
}
```

J.ClassDeclaration	
J.Identifier	name
J.Block	body
JavaType	Type

```
package org.openrewrite;
```

J.Package	
Expression	expression
...	...

```
import java.util.ArrayList;
```

J.Import	
J.FieldAccess	qualified
...	...

```
import java.util.List;
```

J.Import	
J.FieldAccess	qualified
...	...

A

J.Identifier	
String	simpleName
JavaType	type
...	...

```
java.util.ArrayList;
```

J.FieldAccess	
J.Identifier	name
JavaType	type
Expression	expression
...	...

```
java.util.List;
```

J.FieldAccess	
J.Identifier	name
JavaType	type
Expression	expression
...	...

```
a = new ArrayList<>();
```

J.VariableDeclaration.NamedVariable	
J.Identifier	name
Expression	initializer

Automatically fix Checkstyle violations



Migrate to Java 17

Before

```
package org.openrewrite.example;

import java.math.BigDecimal;

public class Example {
    Boolean bool = new Boolean(true);
    Byte b = new Byte("1");
    Character c = new Character('c');
    Double d = new Double(1.0);
    Float f = new Float(1.1f);
    Long l = new Long(1);
    Short sh = new Short("12");
    short s3 = 3;
    Short sh3 = new Short(s3);
    Integer i = new Integer(1);

    void divide() {
        BigDecimal bd = BigDecimal.valueOf(10);
        BigDecimal bd2 = BigDecimal.valueOf(2);
        bd.divide(bd2, BigDecimal.ROUND_DOWN);
        bd.divide(bd2, 1);
        bd.divide(bd2, 1, BigDecimal.ROUND_CEILING);
        bd.divide(bd2, 1, 1);
        bd.setScale(2, 1);
    }
}
```

After

```
package org.openrewrite.example;

import java.math.BigDecimal;

public class Example {
    Boolean bool = Boolean.valueOf(true);
    Byte b = Byte.valueOf("1");
    Character c = Character.valueOf('c');
    Double d = Double.valueOf(1.0);
    Float f = Float.valueOf(1.1f);
    Long l = Long.valueOf(1);
    Short sh = Short.valueOf("12");
    short s3 = 3;
    Short sh3 = Short.valueOf(s3);
    Integer i = Integer.valueOf(1);

    void divide() {
        BigDecimal bd = BigDecimal.valueOf(10);
        BigDecimal bd2 = BigDecimal.valueOf(2);
        bd.divide(bd2, RoundingMode.DOWN);
        bd.divide(bd2, RoundingMode.DOWN);
        bd.divide(bd2, 1, RoundingMode.CEILING);
        bd.divide(bd2, 1, RoundingMode.DOWN);
        bd.setScale(2, RoundingMode.DOWN);
    }
}
```



Migrate to Java 25

```
public static void main(String[] args) {  
    System.out.println("Running main method");  
}
```

```
// Main methods can now be instance methods  
void main() {  
    IO.println("Running main method");  
}
```

```
public void checkSecurity() {  
    Policy.setPolicy(null);  
  
    SecurityManager sm = System.getSecurityManager();  
    sm.checkRead("someFile.txt");  
}
```

```
public void checkSecurity() {  
    // SecurityManager is obsolete and unsupported in Java 25  
    // Setting to `null` makes the dead code explicit and future-proof  
    SecurityManager sm = null;  
}
```



JUnit



Framework Migrations

Migrate to JUnit 5

Migrate to Spring Boot 3.x

Migrate to Spring Boot 2 from Spring Boot 1

Migrate to Quarkus 2

Migrate to Micronaut 4

Migrate to Micronaut 3

Migrate to SLF4J

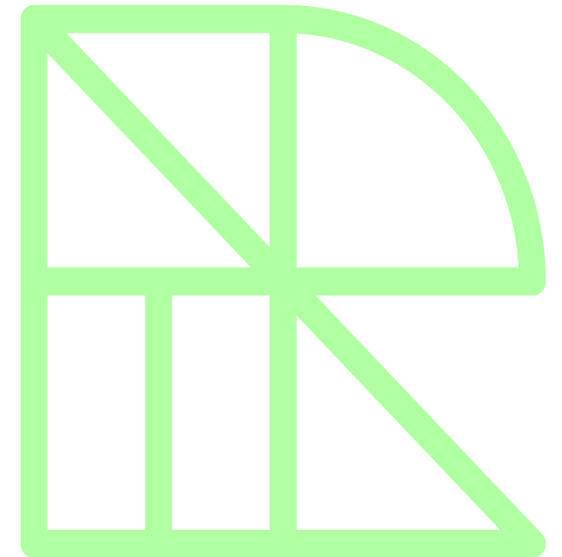
Use SLF4J Parameterized Logging

Automating Maven dependency management

Migrate from Hamcrest to AssertJ

Migrate from JUnit Asserts to AssertJ

Migrate to Jakarta EE 10.0





on



m
Maven
OpenRewriter

rewrite

AI



Java modernization with IBM Bob

<https://www.ibm.com/products/ai-coding-agent/java-modernization>

GitHub Copilot app modernization for Java

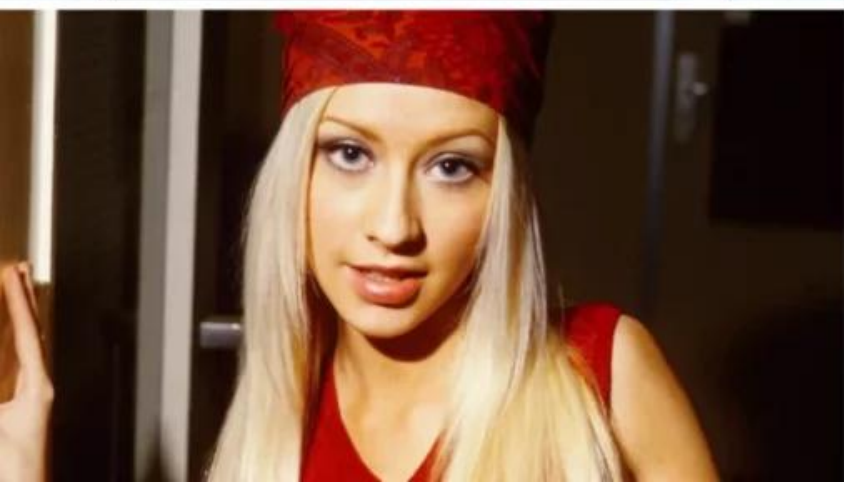
<https://docs.github.com/en/copilot/tutorials/modernize-java-applications>

AI + Recipe authoring

AI coding agents (e.g. Claude Code) can be used to generate OpenRewrite recipes and tests, given the right prompts.

But remember:

- Humans still need to guide, verify, and refine
- Recipe fundamentals are important so you can instruct AI effectively and spot mistakes
- Handwritten recipes may fade, but understanding them won't



Swing to Browser App

The screenshot displays a web application for managing space travel bookings. The interface is divided into three main sections: a sidebar, a main content area, and a detailed booking panel.

Sidebar: Contains navigation links for **Bookings**, **Support**, and **Admin**.

Main Content Area: Features a **Bookings** section with a filter menu (Upcoming, Past, Customers, Invoices) and a list of upcoming bookings. A dropdown menu is open, showing options: **New Booking**, **New Customer**, and **New Invoice**.

Upcoming Bookings:

- IRST-2789** (Galactic Pioneer) - 147 Earth days - **Nora Hansen** (231396647-IRST) - Utility Deck, Luxury Suite 6B - 3 items, up to 3kg
- IRST-969282898** (Celestial Harmony) - 274 Earth days - **Aarav Singh** (314823984-IRST) - Habitation Deck, Luxury Suite 7A - 1 item, up to 10kg
- IRST-827564967** (Galactic Pioneer) - **Heliosshade Realm**

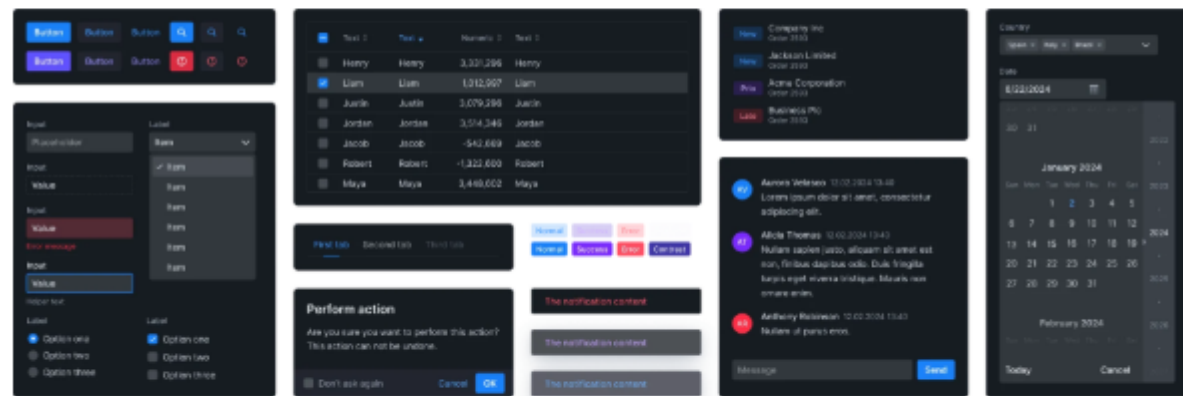
Booking Details Panel (Right): Displays details for booking **IRST-289163384** (Confirmed).

Flight Details:

- Flight:** ISV-2789
- Departure:** Earth, Friday, November 15, 2024
- Arrival:** Kepler-22b, Friday, April 11, 2025
- Duration:** 147 Earth days
- Vessel:** Galactic Pioneer

Modern web apps. 100% Java.

Vaadin Flow is an open-source full-stack Java UI framework and component set that lets you build data-rich, business web applications without writing HTML or JavaScript. Integrates seamlessly with your existing codebase on [your Java server](#) and [favorite IDE](#).



[Latest: 24.8.5](#) | [All releases](#) →

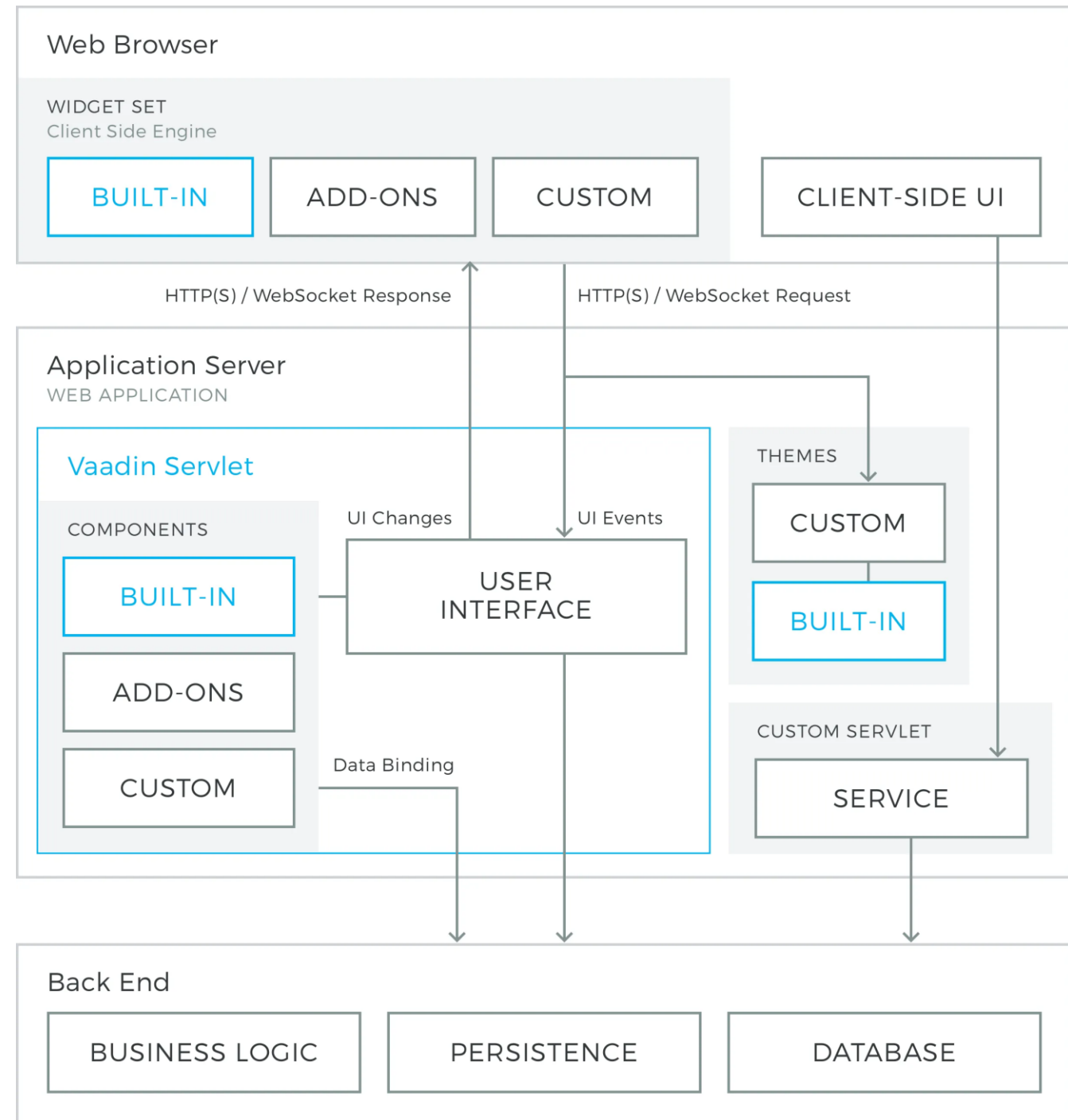


Web standards

The web is a ubiquitous user interface that works on every device. As a continually developed standard, it provides a stable, evergreen foundation for building upon.



Source: <https://vaadin.com/>



```

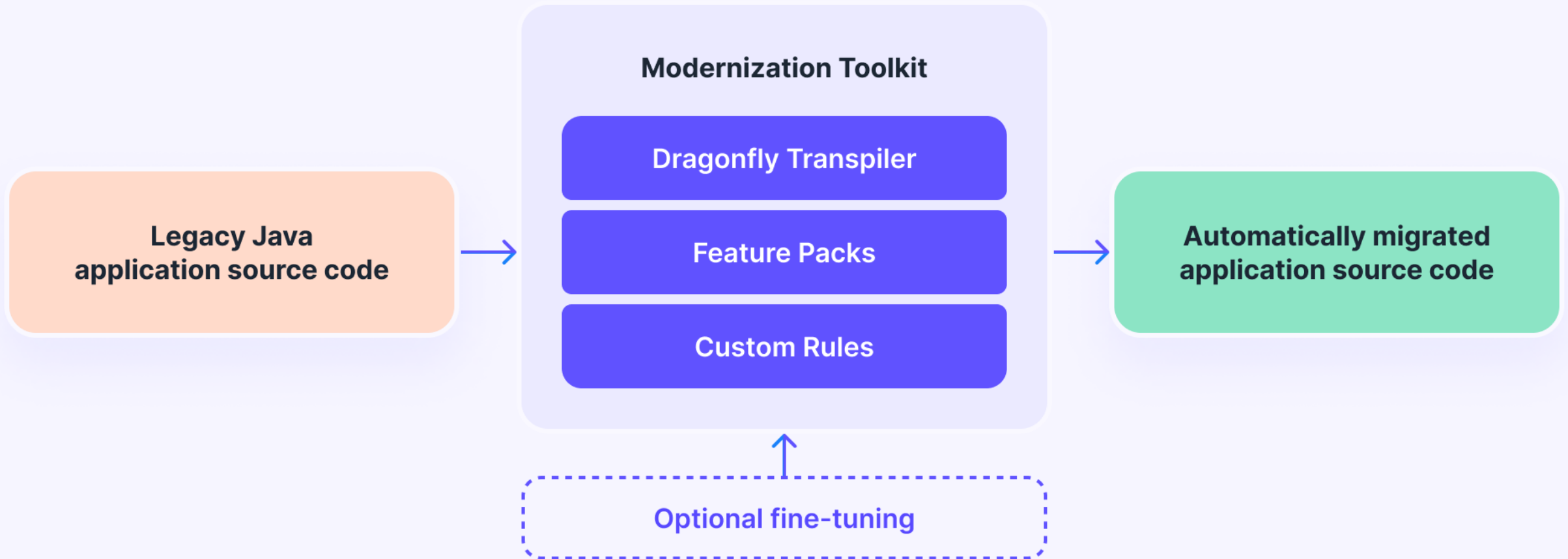
@Route public VoyageView() {

    String[] locations = new String[]{"Bayonne", "Berlin", "Bern", "Bilbao", "Birmingham", "Bremen", "Brussels"};
    ListBox<String> departingCity = new ListBox<String>();
    departingCity.setItems(locations);
    ListBox<String> arrivingCity = new ListBox<String>();
    arrivingCity.setItems(locations);

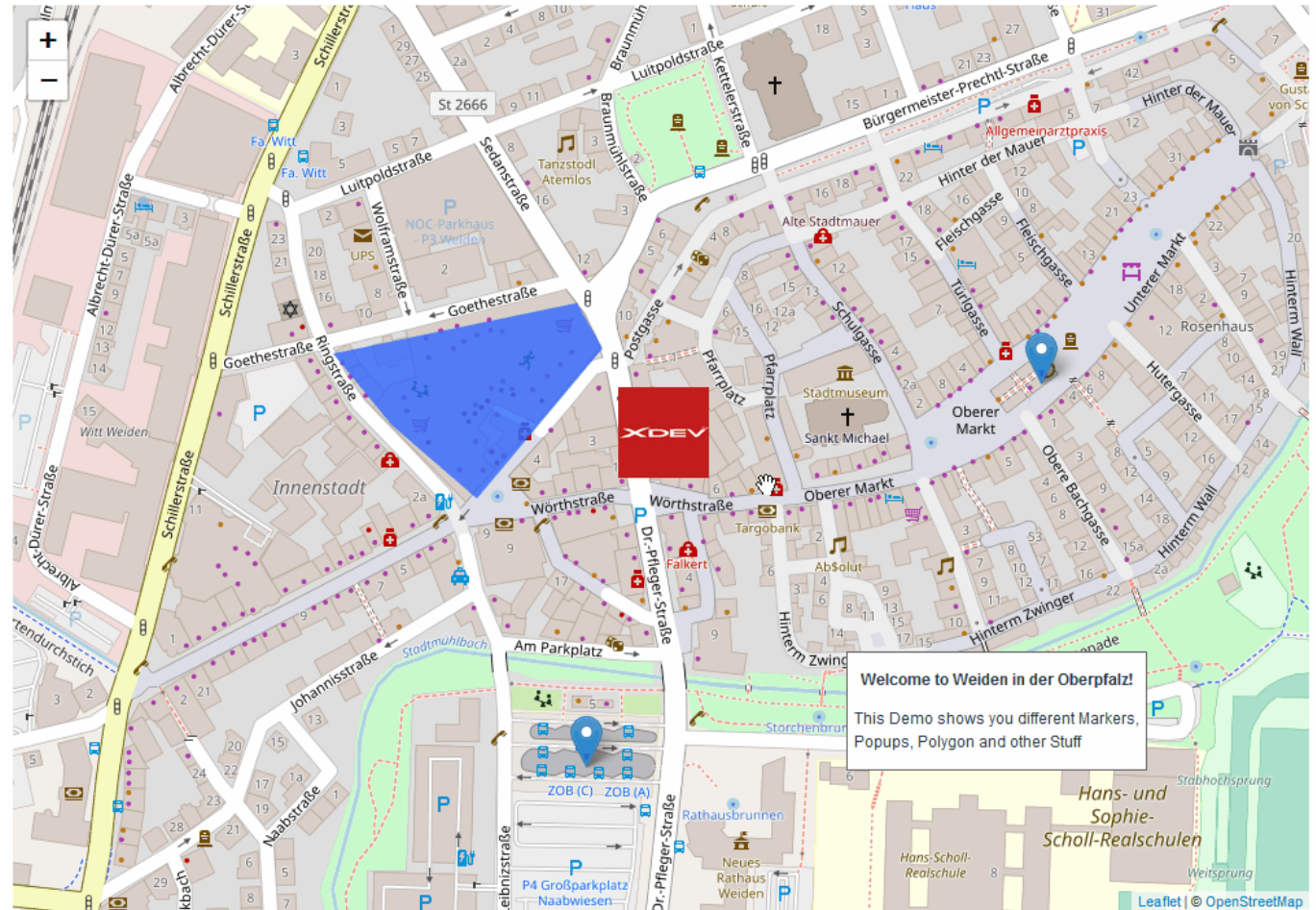
    this.setLayout(new BorderLayout());
    FButton searchButton = new FButton("Search");
    FButton cancelButton = new FButton("Cancel");
    Panel citySelection = new Panel(new GridLayout(0,2));
    citySelection.add(departingCity);
    citySelection.add(arrivingCity);
    Panel buttonBar = new Panel();
    buttonBar.setLayout(new BoxLayout(buttonBar, BoxLayout.VERTICAL));
    buttonBar.add(searchButton);
    buttonBar.add(cancelButton);
    this.add(buttonBar, BorderLayout.WEST);
    this.add(citySelection, BorderLayout.CENTER);
    departingCity.addValueChangeListener(new ValueChangeEvent<String>(){
        @Override
        public void valueChanged(ValueChangeEvent<String>e) {
            if (arrivingCity.getValue() == null) return;
            if (arrivingCity.getValue().equals(departingCity.getValue())) {
                arrivingCity.setValue(null);
            }
        }
    });
}

```

Automatically migrate your legacy Java projects



LeafletMap for Vaadin



Where do XDEV employees go for lunch break?

Other components by XDEV

Home

Cars

Audi

NumberField

123.45



TextField

Some text

TextField with empty value



▼ 26.10.20 - 01.11.20

Period



Week



Start

26.10.2020



End

1.11.2020



Extras for Spring Security

- Full Spring Security control before Vaadin handles requests
- Creates Vaadin sessions only when needed
- CSRF request whitelisting
- Built-in Content Security Policy



**PMD X**

Provides code analysis and highlighting with PMD - an extensible cross-language static code analyzer. The plugin is optimized for

143 downloads

Free

**OpenRewriter X**

Provides support for executing OpenRewrite recipes. Allows running OpenRewrite recipes with Maven and Gradle on project modules

5,314 downloads

Free

**Save Actions X**

Fork of the popular but now deprecated Save Actions plugin by XDEV Software. Supports configurable, Eclipse like, save

281,740 downloads

Free

music-maven-plugin

```
<configuration>
  <sources>
    <musicSource>
      <uri>https://incompetech.com/music/royalty-free/mp3-royaltyfree/Corncob.mp3</uri>
    </musicSource>
    <musicSource>
      <!-- Relative to project directory -->
      <file>my_cool_music.mp3</file>
    </musicSource>
    <musicSource>
      <classpath>/default/Andrew_Codeman_-_03_-_Mussels_short_version.ogg</classpath>
    </musicSource>
  </sources>
  <shuffle>true</shuffle>
</configuration>
```



**Don't reinvent
the wheel**



Is there a library that does
exactly what you need?
If there is already maintained
and secure code, why
shouldn't you use it?

XDEV Commit Cards

Write tests



Whenever it is possible, write
tests for your methods.
Even the most mundane tests
can help at some point.

Q & A

Richard Fichtner

 [RichardFichtner](#)



picture credits

Some pictures in this presentation originate from pixabay.

This is a human-readable summary of the [Pixabay License \(read the full text\)](#).

You can use all images and videos published on Pixabay for free (except as set out below). You may use them for commercial and non-commercial purposes, in altered and unaltered form. You don't need to ask permission from or provide credit to the image author or Pixabay, although it is appreciated when possible.