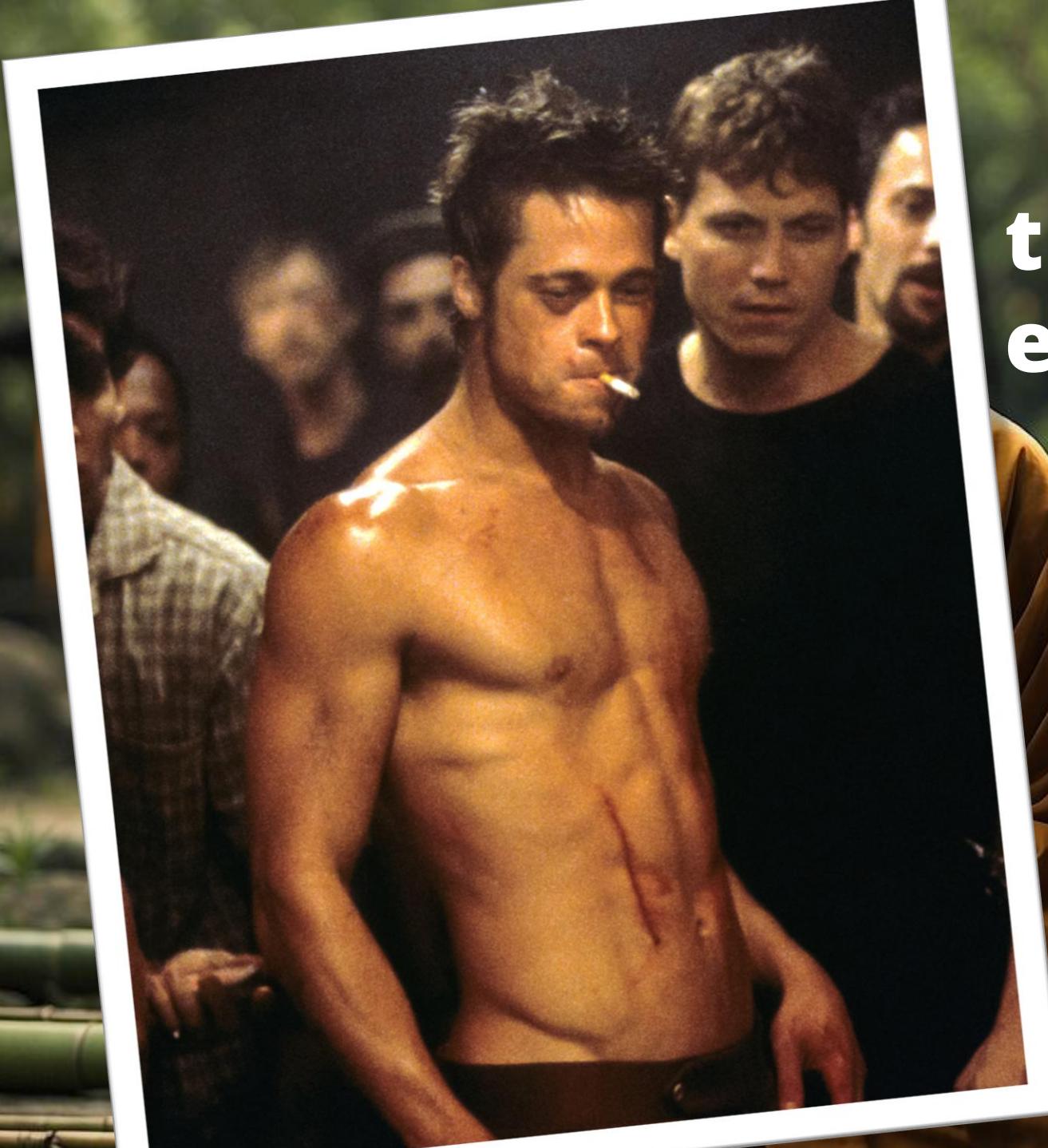


A wide-angle photograph of a massive, ancient tree, likely a live oak, with sprawling branches and hanging clusters of Spanish moss. The tree is set against a bright, clear blue sky. The foreground is slightly blurred, creating a sense of depth.

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

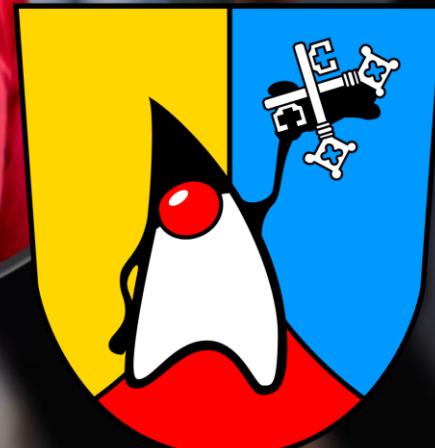
<https://xdev.software>

Projects

JUG Oberpfalz

JCON Co-Organisator

Digital Craftsmanship N.Opf.





Java™
Champions



IBM Champion



Oracle ACE
Pro



JAKARTA™ EE
AMBASSADORS

Rich

News

20+ years

XDEV Se

[https:](https://)

Project:

JUG O

JCON

Digital Craftsmanship N.Opf.



APR
20-23



International
Java Community Conference
JCON2026 www.jcon.one





we make your
Java projects fly



AIRBUS

Red Bull



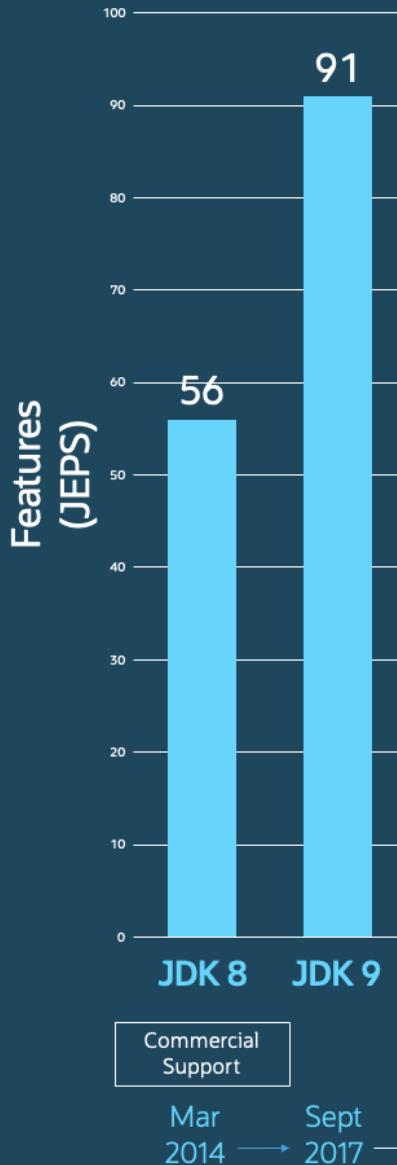
OKI

Jägermeister

EMPORIO  ARMANI

The evolution of Java





<https://javaalmanac.io>



Don't touch this

No code changes allowed !



IBM.



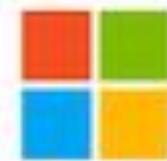
azul



OpenJDK



GraalVM.



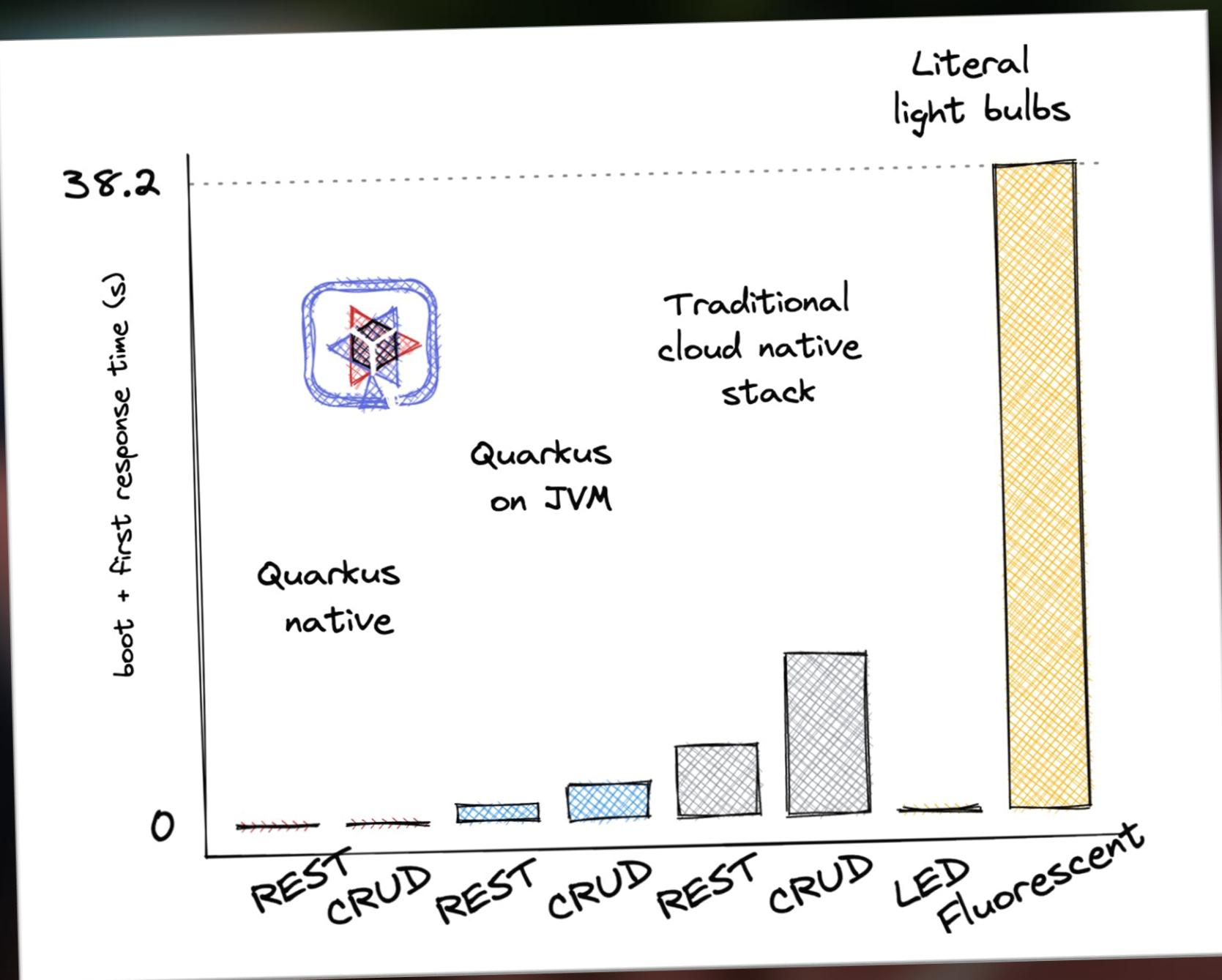
A blurred background image of a supermarket aisle filled with various packaged food items, including cereals, snacks, and drink boxes.

<https://whichjdk.com>

A close-up, low-angle shot of several male athletes in athletic gear, including blue tank tops and patterned shorts, in a starting position on a red running track. They are bent over with their hands on their knees, ready to start a race. The background is blurred, showing other people and trees.

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.

ORACLE **GraalVM**TM

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/>





Java Garbage Collection

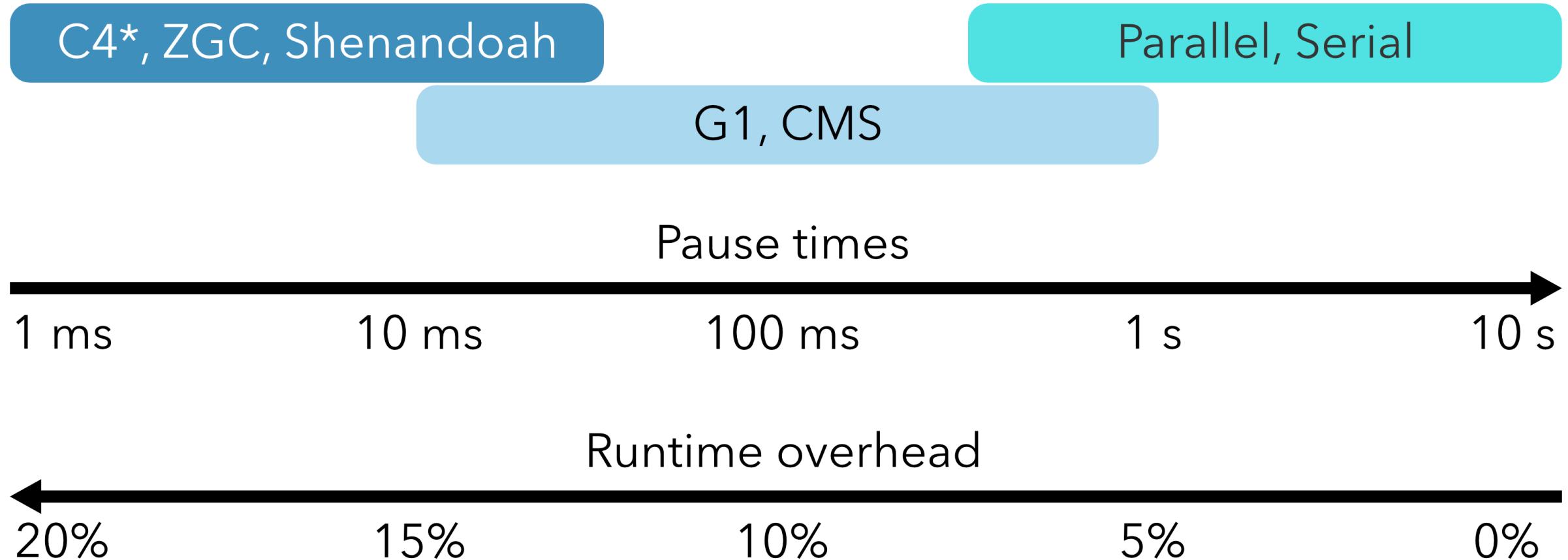
Special Thanks to Gerrit Grunwald



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

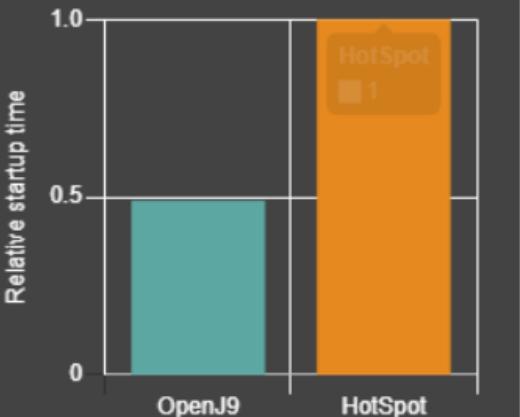




Java Memory Utilization

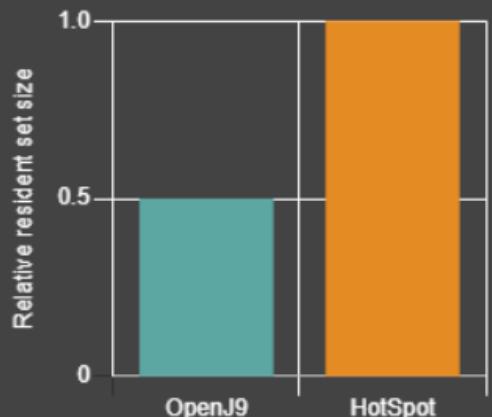
OpenJ9

51% faster startup time



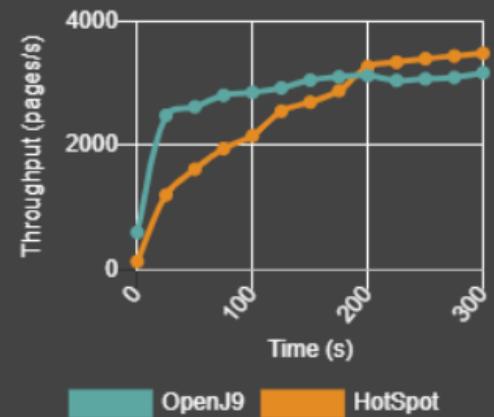
By using shared classes cache and AOT technology, OpenJ9 starts in roughly half the time it takes HotSpot.

50% smaller footprint after startup



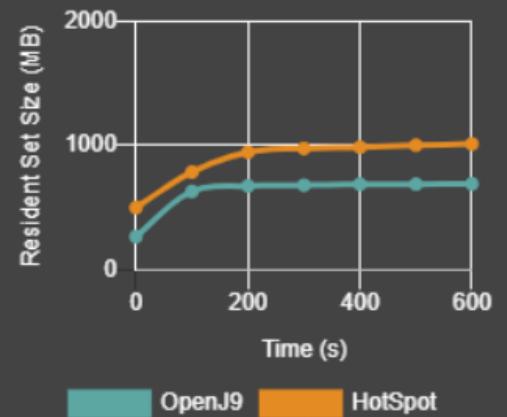
After startup, the OpenJ9 footprint is half the size of HotSpot, which makes it ideal for cloud workloads.

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 allowed



Virtual Threads

Java 21

JEP 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 (2 nd Preview)
Text Blocks	JEP 378
Pattern Matching	JEP 394, 433
Switch Expressions	JEP 361

Book Recommendation

The
Pragmatic
Programmers

Cruising Along with Java

Modernize and Modularize
with the Latest Features



Venkat Subramaniam
Edited by Jacquelyn Carter



**People First.
Technology Second.**

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

Stay up-to-date

Go to conferences



Go to Meetups / JUGs (you are in the right place)

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

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



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

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.

Recipes

A deterministic program that defines *what* to find and *how* to change it in code, using language-aware search and transformation logic to automate research, upgrades, remediations, and refactorings.

Lossless Semantic Tree (LST)

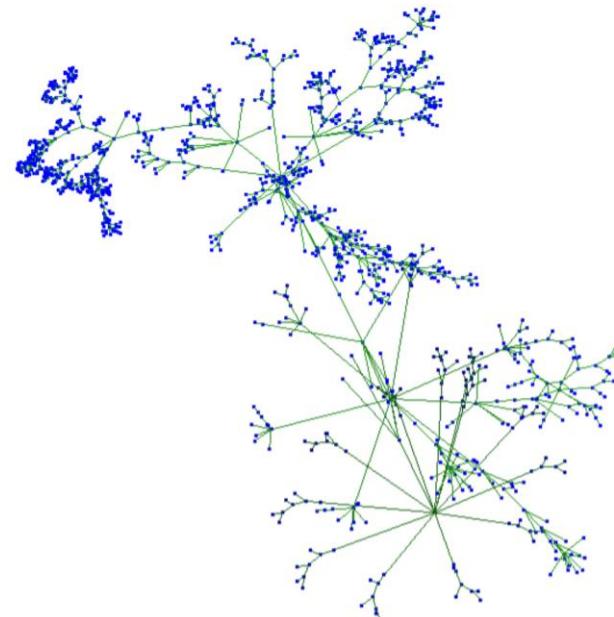
A structured, language-specific representation of code that preserves both syntax and semantics without losing original formatting or comments — enabling precise analysis and transformation at scale.

How does the LST compare to other code representations?

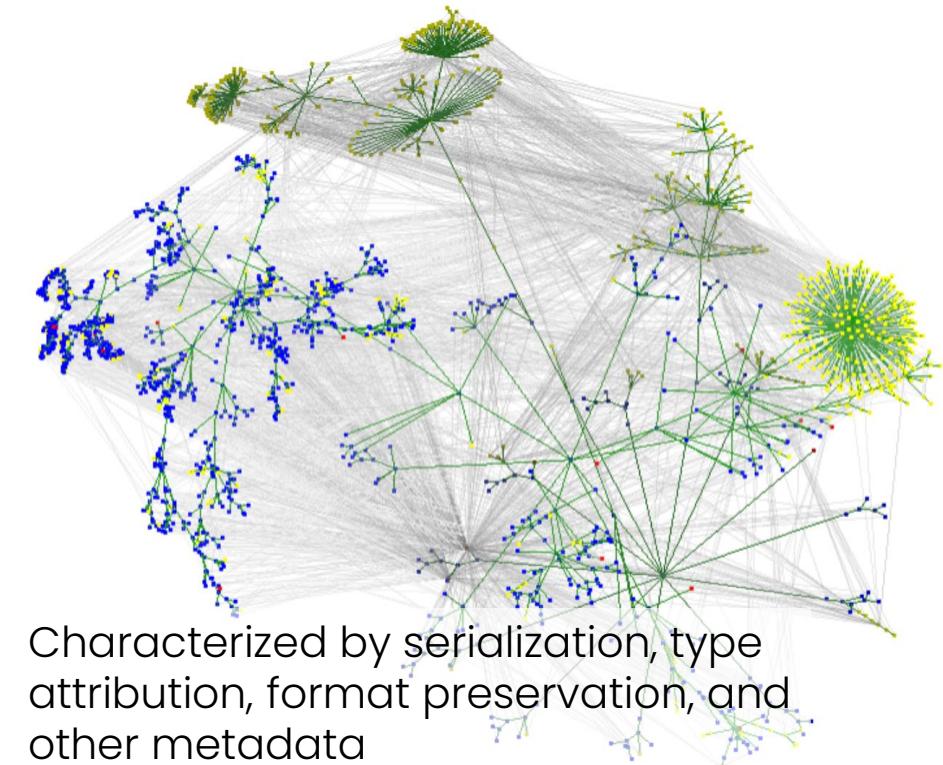
Text-based search

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

Abstract Syntax Tree (AST)



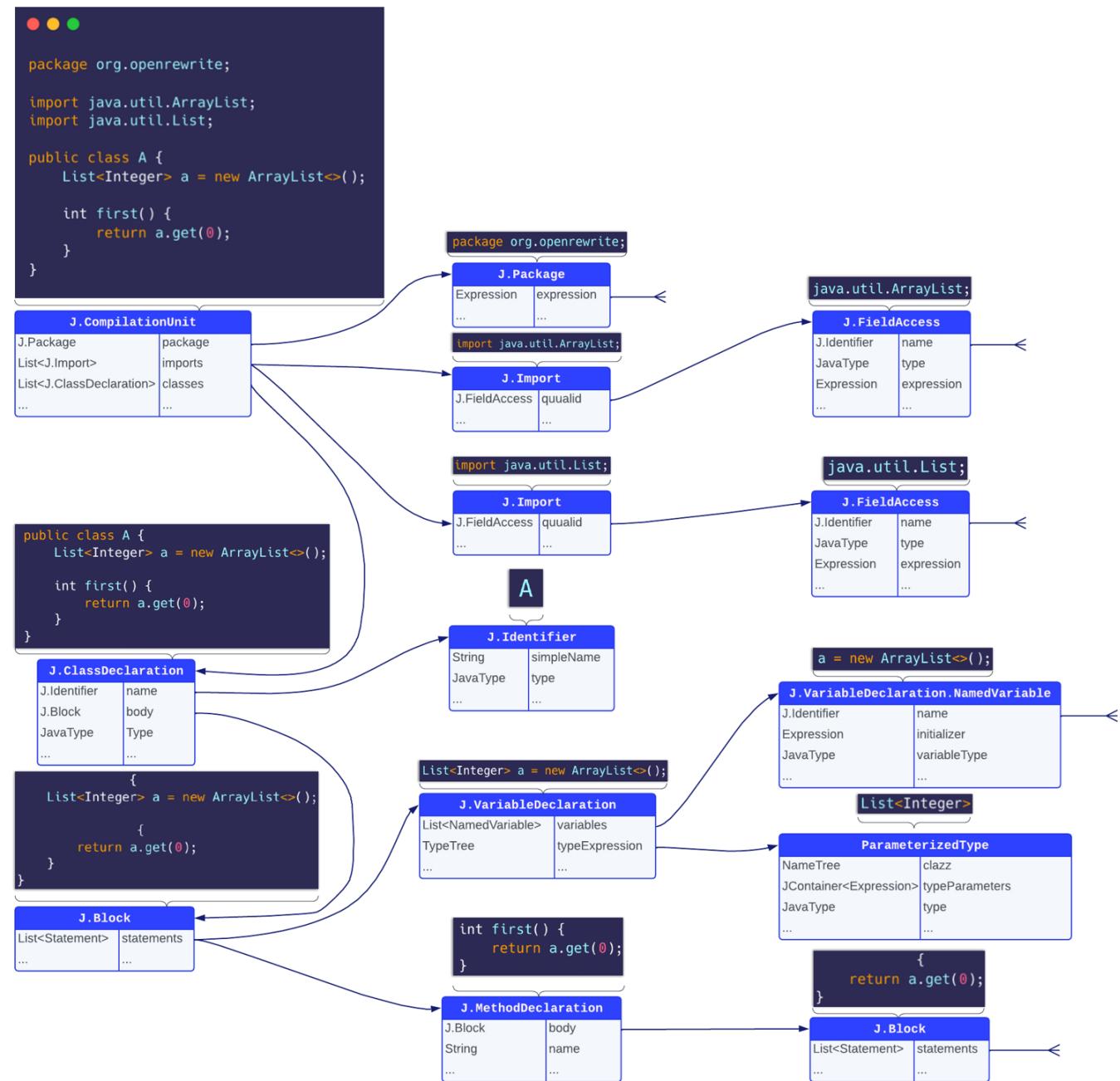
Lossless Semantic Tree (LST)



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

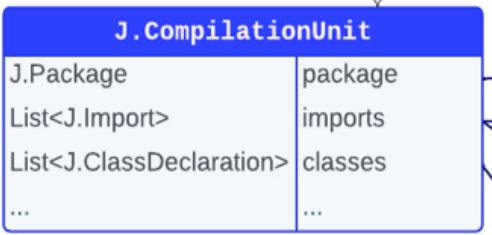


```
package org.openrewrite;

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

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

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



```
package org.openrewrite;
```

J.Package

Expression	expression
...	...

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

J.Import

J.FieldAccess	qualid
...	...

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

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

J.ClassDeclaration

J.Identifier	name
J.Block	body
JavaType	Type

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

J.FieldAccess

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

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

J.Import

J.FieldAccess	qualid
...	...

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

J.VariableDeclaration.NamedVariable

J.Identifier	name
Expression	initializer

J.Identifier

String	simpleName
JavaType	type
...	...

A



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;

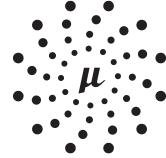
import java.math.BigDecimal;

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);
    }
}
```



QUARKUS



MICRONAUT™



Open Liberty

JUnit



Helidon MP

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

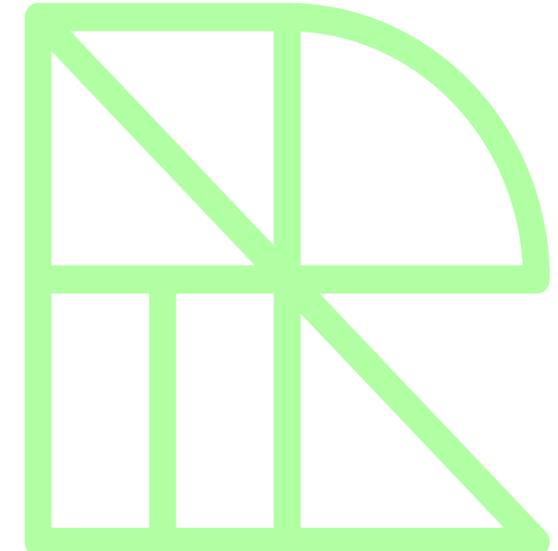
Use SLF4J Parameterized Logging

Automating Maven dependency management

Migrate from Hamcrest to AssertJ

Migrate from JUnit Asserts to AssertJ

Migrate to Jakarta EE 10.0





O



m Maven
OpenRewriter

AI

GitHub Copilot app modernization for Java

<https://github.blog/changelog/2025-05-19-github-copilot-app-modernization-for-java/>



IBM watsonx Code Assistant for Java

<https://www.ibm.com/products/watsonx-code-assistant/java-modernization>

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 booking application interface with the following elements:

- Left Sidebar:** Includes icons for Bookings (selected), Support, and Admin.
- Top Navigation:** Shows "Bookings" and a "+" button.
- Upcoming Tab:** Selected tab under "Upcoming". Other tabs include "Past" and "Customers".
- Booking Details:** A modal window for IRST-289163384, marked as "Confirmed".
 - Flight:** ISV-2789, Galactic Pioneer, 147 Earth days.
 - Passenger:** Nora Hansen, 231396647-IRST, Utility Deck, Luxury Suite 6B, 3 items, up to 3kg.
 - Departure:** Earth, Friday, November 15, 2024.
 - Arrival:** Kepler-22b, Friday, April 11, 2025.
 - Duration:** 147 Earth days.
 - Vessel:** Galactic Pioneer.
- Bookings List:** Shows three entries:
 - IRST-969282898:** ISV-2300, Celestial Harmony, Venus, Thu, Jul 25, 2024.
 - IRST-827564967:** ISV-2724, Galactic Pioneer, Helioshade Realm.



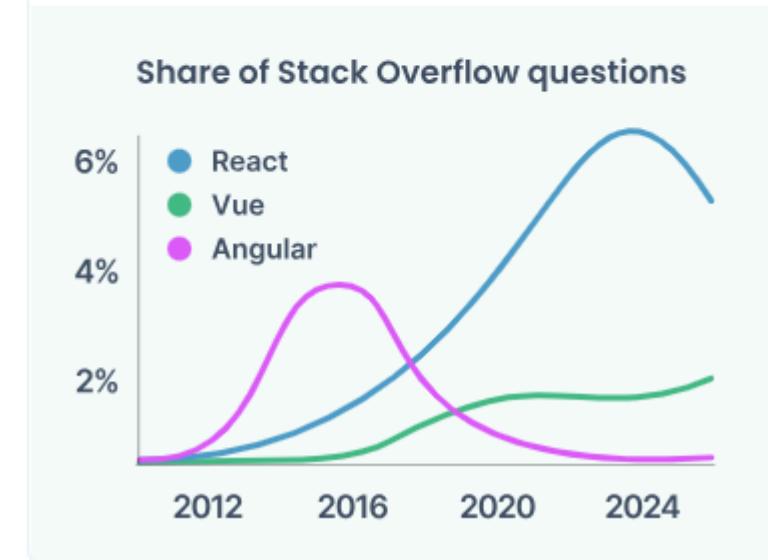
Web standards

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.



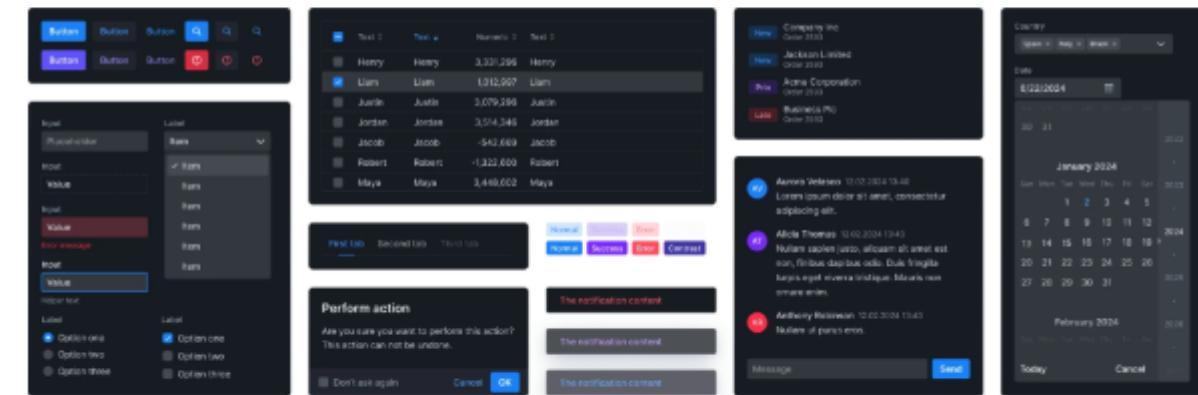
React ecosystem

React is the most popular frontend framework in the world.



Modern web apps. 100% Java.

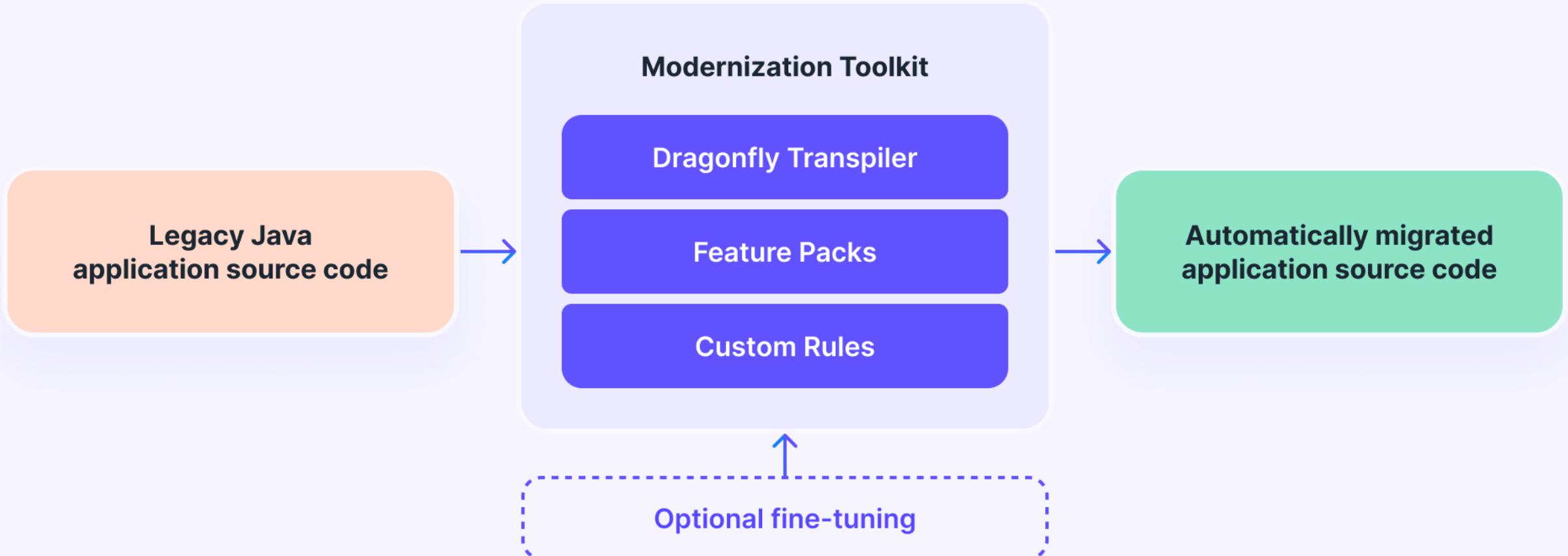
Vaadin Flow is an open-source full-stack Java UI framework and component set that lets you build data-rich, business web applications without writing HTML or JavaScript. Integrates seamlessly with your existing codebase on [your Java server](#) and [favorite IDE](#).



Latest: 24.8.5 | [All releases →](#)

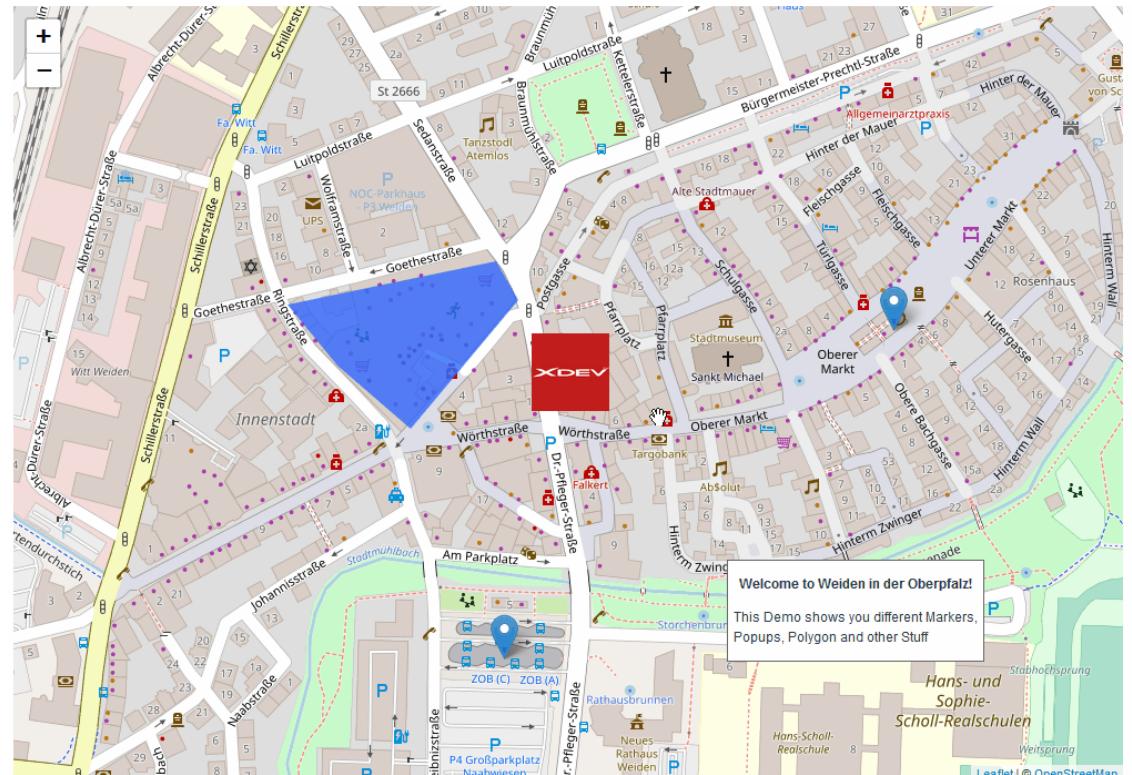
```
@Routepublic VoyageView() {  
  
    String[] locations = new String[]{"Bayonne", "Berlin", "Bern", "Bilbao", "Birmingham", "Bremen", "Brussels"};  
    ListBox <String> departingCity = new ListBox <String>();  
    departingCity. setItems (locations);  
    ListBox <String> arrivingCity = new ListBox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



XDEV Vaadin Tools and Components

- Extras for Spring Security / XDEV SSE (Vaadin Integration)
- Vaadin Components
- RapidClipse UI Builder
- Vaadin Migration



**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

...
something
happening



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

[in 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.