

ORACLE AI World

Everything you need to know about AI and the Java Platform

Donald Smith

Vice President, Product Management

Java Platform Group

October 15, 2025



Wednesday, October 15, 2025

Java Sessions



Session ID	Title	Time	Speaker(s)	Room Location
BOF2993	This isn't Your Grandparent's Java	1:00 p.m. – 1:45 p.m.	Sharat Chander, Senior Director, Head of Java Customer & Community Engagement, Oracle	Expo 202, Level 1
PAN2990	Industry Perspectives of AI Innovation with Java	2:15 p.m. – 3:00 p.m.	Chad Arimura, VP DevRel and Education, Oracle	Titian 2205, Level 2
			Ali Mukadam, Senior Principal Product Manager, Oracle	
			Dr. Christopher Leach, Director of Security Strategy & Regulatory, Quantum Knight	
LRN2995	Modernize Your Enterprise with Java	3:30 p.m. – 4:15 p.m.	Ikroop Dhillon, Director, Developer Relations, NVIDIA	Marco Polo 807, Level 1
			Donald Smith, , Vice President, Product Management, Java Platform Group, Oracle	
LRN3057	AI and Security: The Present and Future of the Java Platform	4:45 p.m. – 5:30 p.m.	Bernard Traversat, Vice President, Software Development, Java Platform, Oracle	Marco Polo 802, Level 1

Thursday, October 16, 2025

Java Sessions



Session ID	Title	Time	Speaker(s)	Room Location
LRN2994	Java in the World of Oracle Cloud Infrastructure	10:15 a.m. – 11:00 a.m.	Bernard Traversat, Vice President, Software Development, Java Platform, Oracle	Marco Polo 807, Level 1
LRN3058	Elevate Your Emerging AI Solutions with Java	11:30 a.m. – 12:15 p.m.	Chad Arimura, VP DevRel and Education, Oracle	Marco Polo 802, Level 1

AI World

QR Code for all Java Sessions



Let's continue the conversation!

Stop by our Java Pod:
Expo Hall Level 2
In the Oracle booth (OCI area)
Directly across from "Oracle TV"



Ask to see our Java and AI demo



JAVAONE '26



Conference Pass

Tuesday, March 17 – Thursday, March 19



Oracle Conference Center
350 Oracle Parkway
Redwood Shores, CA

javaone.com

**“Java *is* everywhere AI
needs to be”**

Build Better Agents in Java Than Python: Embabel vs Pydantic AI

In my last post, I showed how a CrewAI example could be rewritten in Java using Embabel to be more robust, concise,...

Aug 21 🖱️ 256 💬 2



You Can Build Better AI Agents in Java Than Python

Too many people assume that Gen AI means Python. Not so.

Aug 18 🖱️ 744 💬 8



Rod Johnson

7K followers

Founder at Embabel. Creator of Spring, Cofounder/CEO at SpringSource (acq by VMW)



* <https://redmonk.com/jgovernor/2016/10/12/when-web-companies-grow-up-they-turn-into-java-shops/>

**Oracle and Java are well positioned in
AI services, tools and platforms**

**Investments in the Java Platform will
benefit Oracle customers for decades
to come**



Java is the #1 Language for AI Developer Productivity

VDC 2025 Survey
n=532 survey of professional enterprise developers and hobbyists

Agenda

What I mean by “Java”

What I mean by “AI”

Interesting uses of Java and AI today

Java Platform Innovation Projects:

Babylon – Java and GPU

Panama – Java and Native

Valhalla – Java and Value Objects



Java Platform

- Java Development Kit (JDK)
- Java Runtime Environment (JRE)
- Java Virtual Machine (JVM)
- Java Language
- Java APIs
- Java SE Specification
- Related Oracle Java SE Products



Developer productivity *Program performance*

In the face of constantly-evolving
programming paradigms,
application areas,
deployment styles,
and hardware



Continued renaissance



Java continues to be an **important choice** for enterprises and developers

The Java technology **innovation pipeline** has never been richer

Java 25 available on September 16

#1

Language in overall
enterprise organizational use

#1

Language for
cloud-native deployments

73B

Active JVMs

51B

Cloud-based JVMs

Choose what to work on

JDK expectations for compatibility, stability, security, and performance are extreme



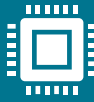
Evolving use-cases



Improvements in hardware



New programming paradigms



New trends or hardware architectures



Endeavor to understand problems



Look to identify synergies



Avoid attempting to shoehorn solutions



Avoid the temptation to shortcut

THOUGHTFUL EVOLUTION

“Tip and Tail”

Conservatism

Compatibility
Don't alienate users

Innovation

Adapting to change
Fixing mistakes





We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**

AI Terminology

AI Compute

- Training, Machine learning, Model creation

- Computationally intense, large data sets

AI Integration

- Making Inferences

- Connecting inferences to business logic and business data

- Connecting to AI Services and Platforms

- Running and *trusting* AI agents

AI Applications, AI Tooling

- Ready made, business-facing apps

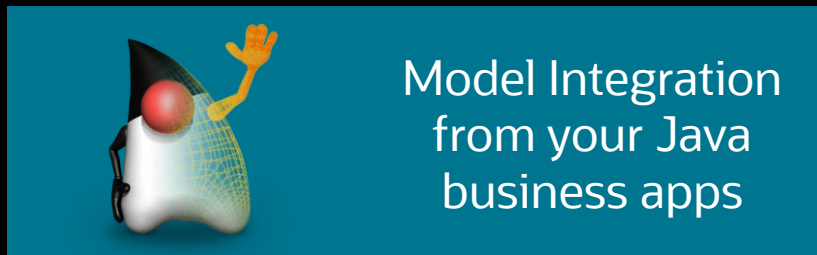
- Developer productivity tooling (code + build + support) and related AI features

Java and AI Today

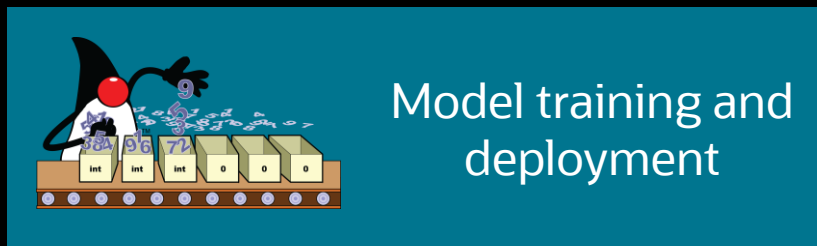
Oracle Java and AI: Three-pronged strategy



Now with GenAI tools and services like **Oracle Code Assist, Cursor + Java VS Code Extension by Oracle**



Now with tools like **Langchain4j, Spring AI, OCI SDK for Java**, and Projects **Amber, ZGC, Panama**



Next up with Projects **Panama, Valhalla, Babylon** (Demonstrated by **HAT**)

Lots of progress in the Java + AI ecosystem

- **Langchain4j** hit 1.0 GA release, introducing Virtual Threads, model expansion, agentic mode, enhanced reasoning support, multimodality, and more
- **Spring AI** hit 1.0 GA release with model expansion, MCP integration, tool calling, and more
- **Embabel**: Agentic Framework launched in May: Goal-oriented action planning, seamless LLM integration, and more. "Natively Agentic"

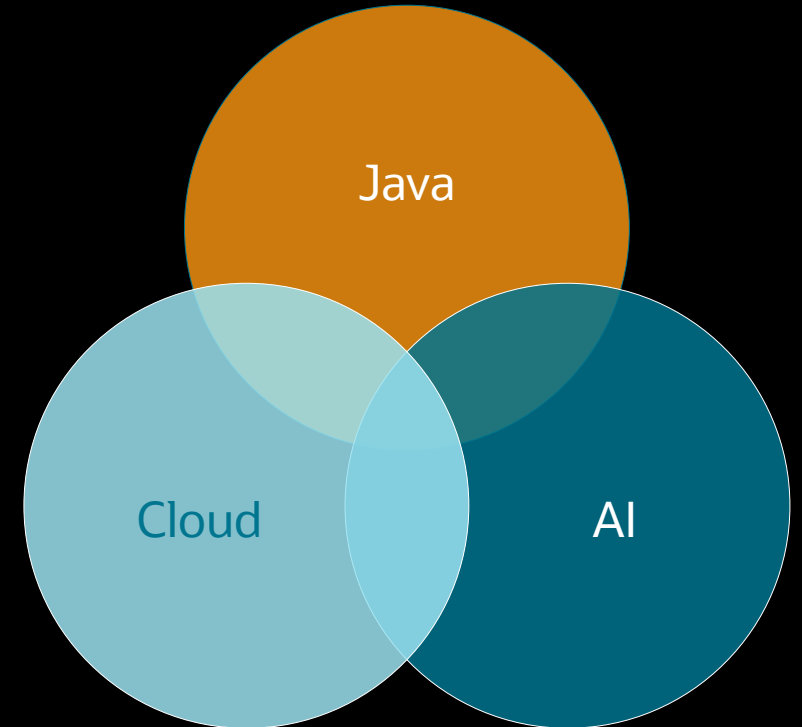




Helidon Framework

Modern, open-source Java Framework for micro services

- **Java**
 - Simplicity and performance without sacrificing functionality
 - Early adoption and support of new Java features, JDK releases
 - Blazing fast, virtual threads from the socket up
- **Developer & cloud-native friendly**
 - Write, understand, debug, maintain clean code
 - Developer choices: programming and API models
 - Container, Kubernetes ready
- **Loaded for AI**
 - Scalable MCP Server based virtual threads
 - LangChain4j support
 - Enhanced support for Oracle AI models, Oracle DB 23ai embedding Store ++
 - Oracle Cloud Generative AI Service ++



AI Integration

Several new AI projects in Java

May also wrap a Runtime and run local but inference computation isn't within Java

Many SDKs as REST calls which perform LLM inference as a service

Very well served already by Java's Strengths

- Strong typing, good abstractions, good core libraries, memory safety, performance, observability, security, cloud support, web servers, extensive third-party libraries/tools, development speed, developer base, stability and predictability



JDK 25: AI-related JEPs



- **Primitive Types in Patterns, instanceof, and switch 3rd Preview [507]**
 - Integrating business logic with primitive types from AI inference now easier
- **Module Import Declarations [511]**
 - More easily integrate business logic with AI inference, libraries and/or service calls
- **Vector API 10th Incubator [508]**
 - Often used in AI inference and compute scenarios
- **Structured Concurrency 5th Preview**
 - AI Development often involved running multiple tasks in parallel
- **Scoped Values [506]**
 - Enable sharing of immutable data within and across threads with lower space and time costs vs thread-local variables

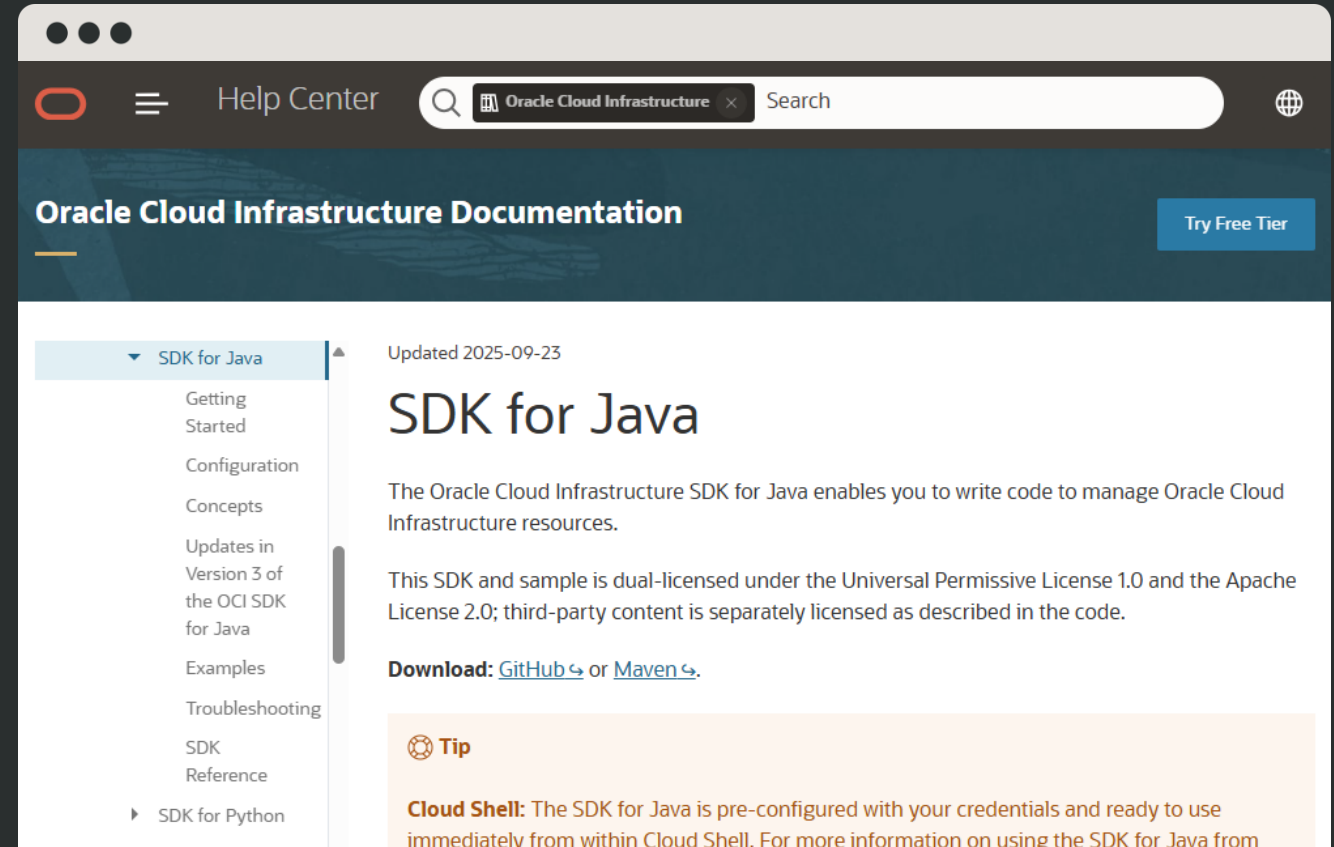
OCI SDK for Java

Over 120 Services Supported!

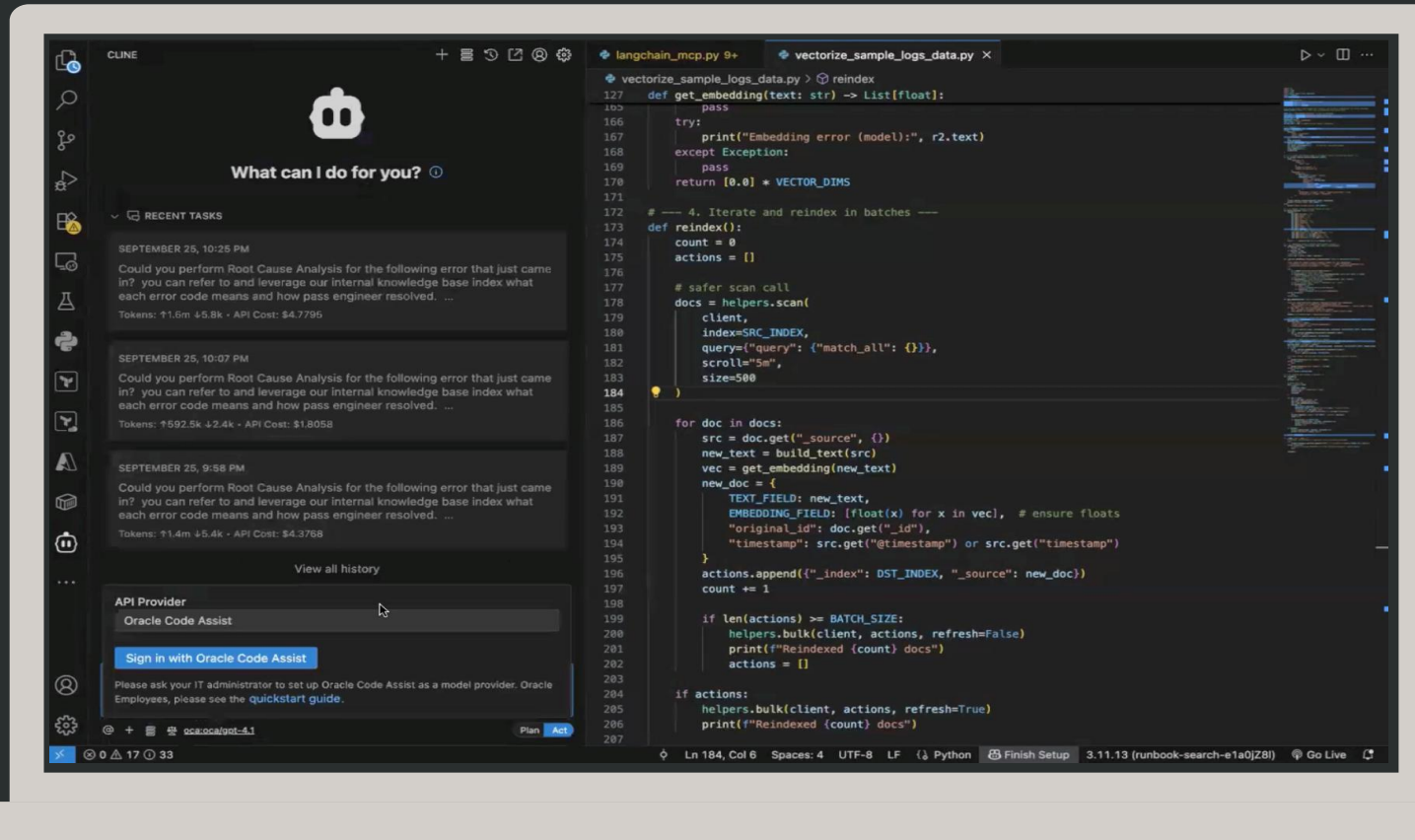
AI Data Platform, AI Language, AI Speech. AI Vision, Generative AI Agent, Generative AI Inference

UPL and Apache Licensed

Support for Java 8, 11, 17, 21 – and soon 25!



Oracle Code Assist: Accelerating developer productivity



10 hours
saved per week for devs

60%
faster code migrations

10 %
production release
code used OCA

Java is Innovating on all fronts – integration, tooling and compute



Features we are working on to empower Java developers to build better AI solutions



We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**



We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**

Project Valhalla – Exec Summary

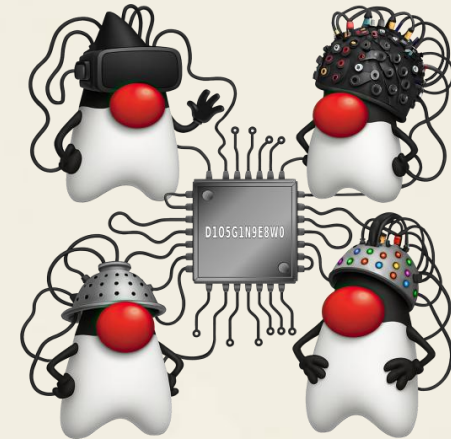
Biggest refactor of Java language and runtime ever undertaken

Key Motivators:

- Flatter and denser memory layouts

- Unify the type system rift between primitives and objects

- Grow the language to allow new numeric types



Project Valhalla – Exec Summary

Biggest refactor of Java language and runtime ever undertaken

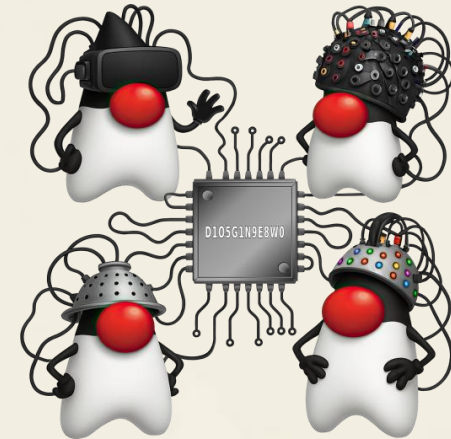
Key Motivators:

- Flatter and denser memory layouts

- Unify the type system rift between primitives and objects

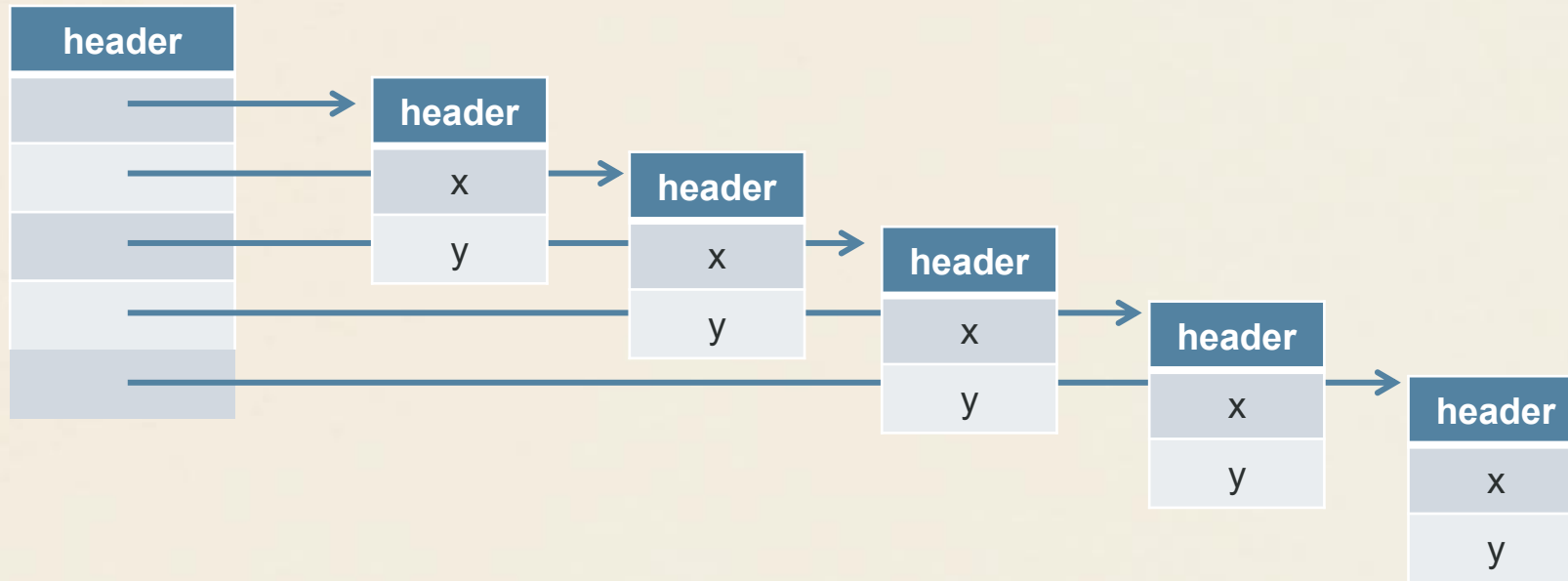
- Grow the language to allow new numeric types

I contend the benefits to AI
Compute here are self evident....



Project Valhalla – Flatter and denser memory layout

For an array of XY point objects, we get this layout:



Over the past 30 years, relative cost of memory fetches vs arithmetic operations has increased by 100-1000x, pointer-heavy layouts beg for expensive cache misses.

Project Valhalla – Flatter and denser memory layout

We would often prefer this layout...

header
x
y
x
y
x
y
x
y

Project Valhalla – Unify the type system

Rift between primitives and objects (int vs Integer) is “Java’s original sin”

- Necessary in 1995 for numeric performance, but challenging ever since

- Puts developers to a bad choice – clear, clean, abstract, safe OO code, or...

- Error-prone, unmaintainable primitives to get performance

Valhalla offers the best of both worlds – “Codes like a class, works like an int”

Challenging problem to address:

- Classes are user-defined, primitives built in

- Polymorphism, Identity, nullability, initialization, layout, atomicity

Project Valhalla – Growing the language

How would we add new numeric types such as half-floats, complex, fixed-point decimals, or unsigned integers, so they feel “built in”?

Wrong way -> add more primitives, bytecodes, conversion rules and API surface

Right way -> as libraries (!)

Focus on making Java extensible via libraries allowing classes to opt into primitive-like semantics, layout, performance, convertibility, operators

Project Valhalla – Examples

```
// Records delivered in JDK 16 – all members final, auto constructors  
// Provided automatically: accessors, toString, equals, hashCode  
record Rectangle(float length, float width) { }  
record Point(int x, int y) { }
```

```
// Records currently in Valhalla EA release at jdk.java.net  
value record Rectangle(float length, float width) { }  
value record Point(int x, int y) { }  
// ‘value’ indicates this type has no need for identity features  
// This gives the JVM freedom to encode at runtime in ways to optimize memory,  
// locality and GC efficiency.
```

Project Valhalla – What's the potential?

Benchmarks comparing operations on arrays of objects vs arrays of equivalent value objects:

500x500 Complex matrix multiplication: throughput 6x, 1000x less allocation

Summing array of int pairs 3.5x – 12x depending on factors such as:

Mutability, identity, loops vs streams, etc

Gains come from increased locality, reduced allocation, scalarization

Not to mention the benefits of being able to easily create and use new numeric types!

Call To Action

jdk.java.net

Production and Early-Access OpenJDK Builds, from Oracle

Ready for use: JDK 25, JavaFX 25, JMC 9.1.1

Early access: JDK 26, JavaFX 26, JavaFX Direct3D 12, Jextract, Leyden, Loom, & Valhalla



jdk.java.net
inside.java



We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**

Interconnecting the JVM and native code

Native interop used to be frowned upon – “Pure Java” was the goal

Legacy “Java Native Interface” (JNI) has challenges:

- Native-first programming model, brittle, expensive to maintain

- Passing data is cumbersome and inefficient

While there are many great Java libraries, there are also many important native libraries

- Off-CPU computing (Cuda, OpenCL)

- Machine Learning (Blis, ONNX, Tensorflow)

- Graphics processing (OpenGL, Vulkan, DirectX)

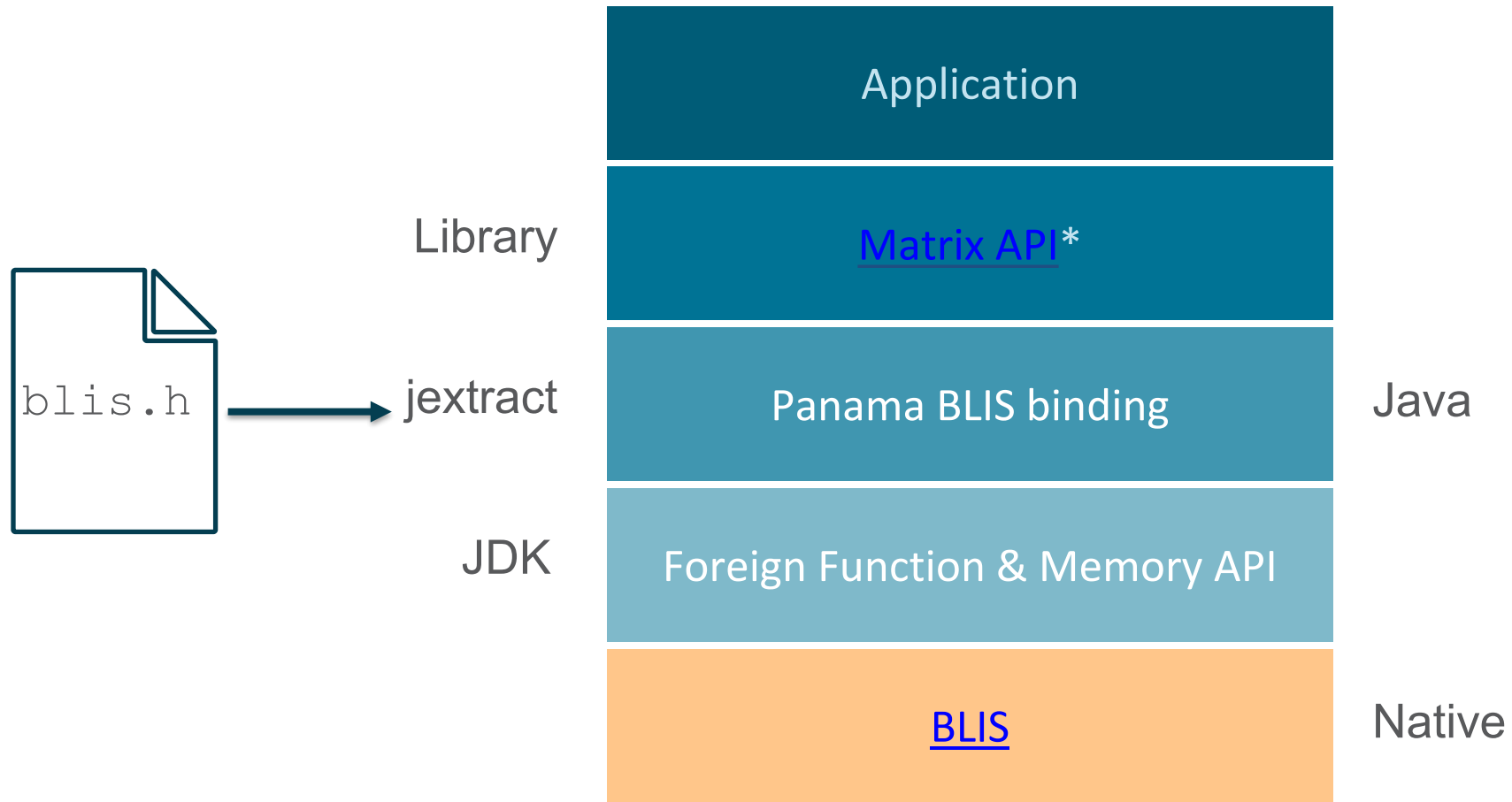
- Others (CURL, fuse, OpenSSL)

Enter Project Panama

JEP 454: Foreign Function & Memory API

<i>Owner</i>	Maurizio Cimadamore
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	22
<i>Component</i>	core-libs / java.lang.foreign
<i>Discussion</i>	panama dash dev at openjdk dot org
<i>Relates to</i>	JEP 442: Foreign Function & Memory API (Third Preview) JEP 472: Prepare to Restrict the Use of JNI
<i>Reviewed by</i>	Alex Buckley, Jorn Vernee
<i>Endorsed by</i>	Alan Bateman
<i>Created</i>	2023/06/22 09:36
<i>Updated</i>	2024/01/29 21:28
<i>Issue</i>	8310626

Matrix API using native BLIS



* See also JavaOne 2022 presentation [“BLISful Linear Algebra with Project Panama”](#)

Java using bindings

```
int dt;  
long m, n, k;  
long rs, cs;
```

```
MemorySegment a = obj_t.allocate(arena);  
MemorySegment b = obj_t.allocate(arena);  
MemorySegment c = obj_t.allocate(arena);  
MemorySegment alpha;  
MemorySegment beta;
```

```
// Create some matrix operands to work with.  
dt = BLIS_DOUBLE();  
m = 4; n = 5; k = 3; rs = 0; cs = 0;  
bli_obj_create(dt, m, k, rs, cs, a);  
bli_obj_create(dt, k, n, rs, cs, b);  
bli_obj_create(dt, m, n, rs, cs, c);
```

Java using Matrix API

```
long m, n, k;
```

```
// Create some matrix operands to work with.
```

```
m = 4; n = 5; k = 3;  
DoubleMatrix a = newDoubleMatrix(arena, m, k);  
DoubleMatrix b = newDoubleMatrix(arena, k, n);  
DoubleMatrix c = newDoubleMatrix(arena, m, n);
```

Project Panama - Interconnecting JVM and native code

Accessing memory

Heap Segments (eg, Java array), Native Segments (eg, malloc/mmap)

Size, Lifetime, Confinement (optional)

Arena-based memory management – clients define custom arenas based on need

Global, Automatic, Confined (single-thread), Shared

Strong safety guarantee

Accessing foreign functions

Native linker knows *calling conventions* on the platform where JVM is running

Provides both downcall method handles and upcall stubs to Java

Handles memory layouts for signatures, memory segments, arenas and lifetime

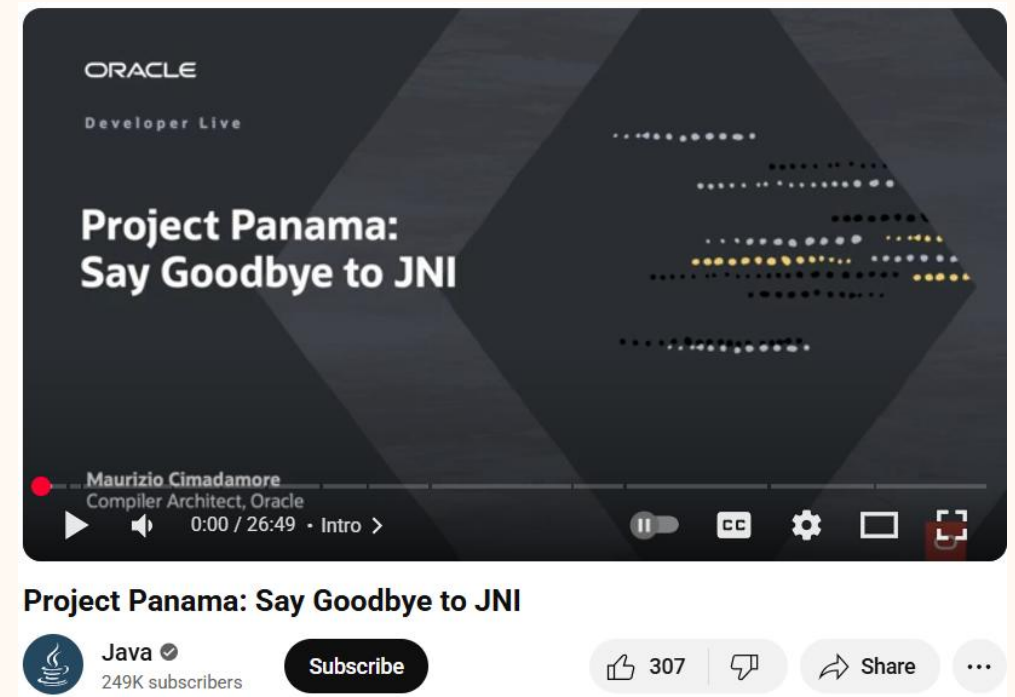
Project Panama – More Information

Search terms:

Panama JVMLS | YouTube Panama Java

Sites:

openjdk.org/projects/panama





We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**

Project Babylon – Exec Summary

“Babylon’s primary goal is to extend the reach of Java to foreign programming models such as SQL, differentiable programming, ***machine learning models, and GPUs***. Babylon will achieve this with an enhancement to reflective programming in Java, called code reflection.”

Developers should not have to:

- Embed snippets of non-Java code

- Write tedious data structures to represent their program

- Use non-standard APIs to access or analyze their program

Project Babylon – GPU Code Example

```
// Developers should write ordinary Java code
static double f(double x, double y) {
    return x * (-Math.sin(x * y) + y) * 4.0d;
}

// Java Developers should not have to write this code, the compiler should do that
var fModel = func("f", methodType(double.class, double.class, double.class))
    .body(entry -> {
        var x = entry.parameters().get(0);
        var y = entry.parameters().get(1);
        var r = entry.op(mul(
            entry.op(mul(
                x,
                entry.op(add(
                    entry.op(neg(
                        entry.op(call(MATH_SIN,
                            entry.op(mul(
                                x,
                                y)))))),
                    y))))), entry.op(constant(DOUBLE, 4.0))));
        entry.op(_return(r));
    });
```

Project Babylon – GPU Code Example

Java Method to differentiate:

```
@CodeReflection
static double f(double x, double y) {
    return x * (-Math.sin(x * y) + y) * 4.0d;
}
```

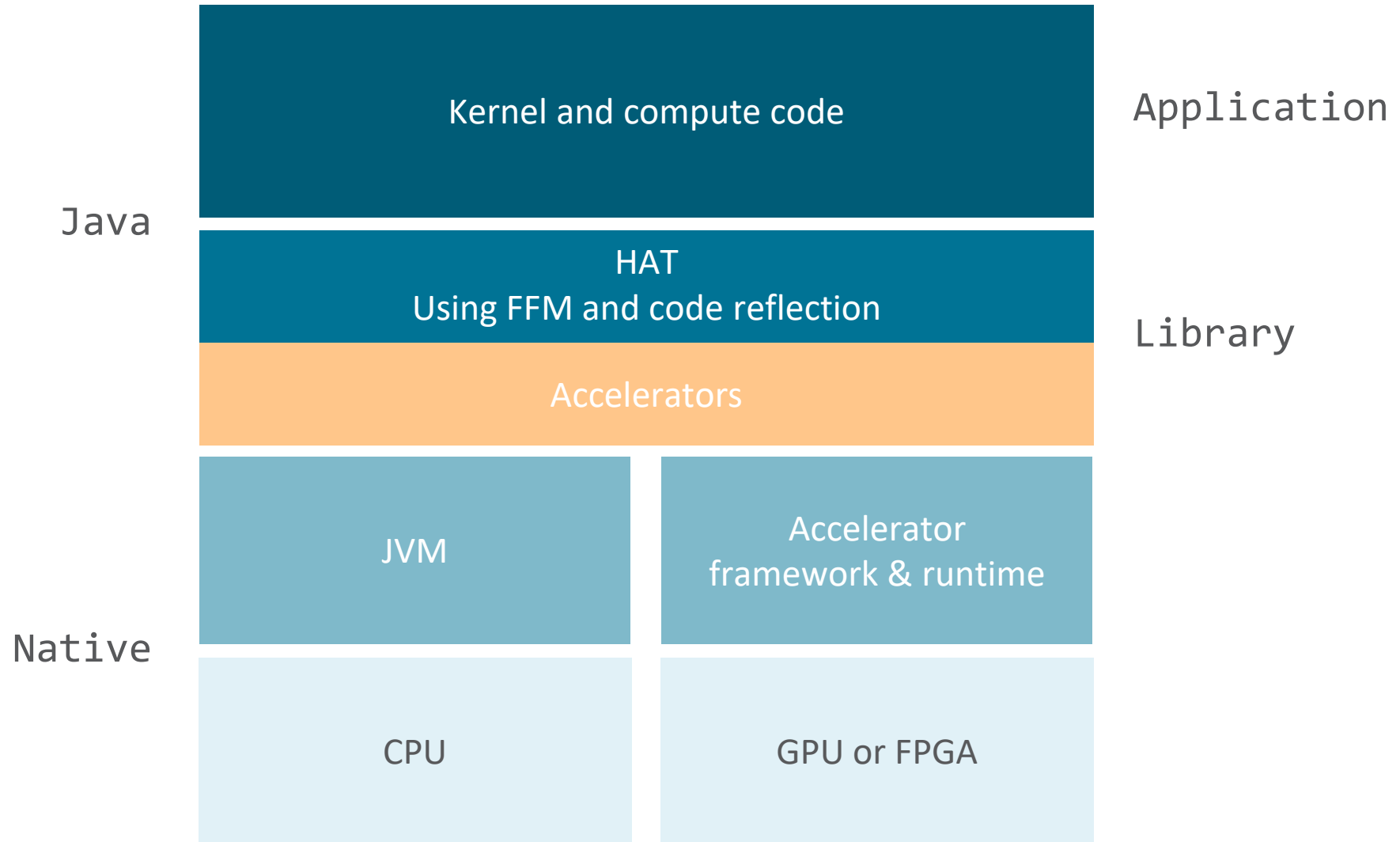
Serialized Code Model:

```
func @"f" (%0 : double, %1 : double)double -> {
    %2 : Var<double> = var %0 @"x";
    %3 : Var<double> = var %1 @"y";
    %4 : double = var.load %2;
    %5 : double = var.load %2;
    %6 : double = var.load %3;
    %7 : double = mul %5 %6;
    %8 : double = call %7 @"java.lang.Math::sin(double)double";
    %9 : double = neg %8;
    %10 : double = var.load %3;
    %11 : double = add %9 %10;
    %12 : double = mul %4 %11;
    %13 : double = constant @"4.0";
    %14 : double = mul %12 %13;
    return %14;
};
```

Element-wise operations and HAT

- Use code reflection to generate Java GPU kernels
 - Which are then translated by HAT, again using code reflection, to foreign GPU kernels
 - Optimal execution on the CPU may also be possible via HAT, unifying kernel representation and execution
- Abstract matrix allocation
 - Data may be allocated on the CPU (host) or GPU, sharing the data model
- Abstract matrix computation
 - Similarly to that of FFM's confined arenas and HAT's computation model
 - Execute close to where data resides, the host (CPU) or GPU, copy when necessary

HAT architecture



HAT accelerators

Java CPU Accelerator	CUDA Accelerator	HIP/ROCm Accelerator	OpenCL/SPIRV Accelerator
JVM	CUDA framework & runtime	HIP/ROCm framework & runtime	OpenCL/SPIRV framework & runtime
AMD Zen5	NVIDIA Blackwell B200	AMD Instinct MI300X accelerator	Intel Arc



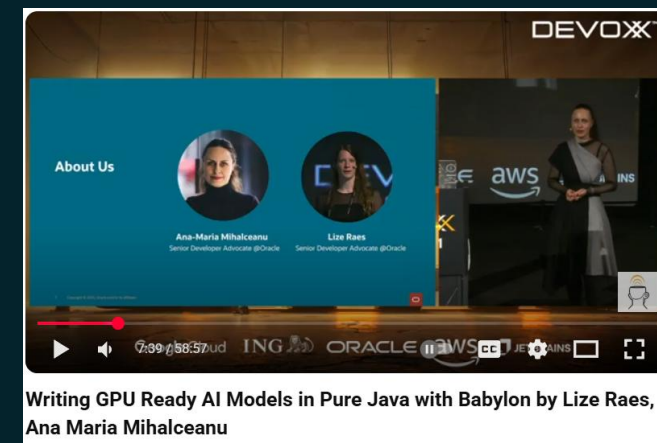
Call To Action: More High-quality technical and community information from the source



Dev.java



Inside.java



youtube.com/java





We continuously evolve Java to meet your **future app development** needs

Java has **30 years** of experience evolving with the latest tech trends

Data-centric World

Amber

Continuously **improve developer productivity** through evolution of the Java language.

ZGC

Create a **scalable low-latency** garbage collector capable of handling terabyte heaps.

Cloud-powered World

Loom

Massively scale lightweight threads, **making concurrency simple again.**

Leyden

Improve the start-up time and time to peak performance of cloud applications.

AI-driven World

Babylon

Extend the reach of Java including to machine learning models and GPUs.

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to **improve productivity and performance.**



Thank you

Name or org

Additional information

**Your feedback is
important.**

**Scan this QR Code or use
the Mobile App to share
your thoughts on this
session.**



ORACLE