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 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

Moving Java forward

30 years of innovation

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

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

Java 23 planned for Sept 17

#1
Language for today's Technology Trends

#1
Overall Enterprise/IT Organizational Use

63B
Active JVMs

41B
Cloud-based JVMs

* VDC 2023 study

Choosing 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"

Conservation
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

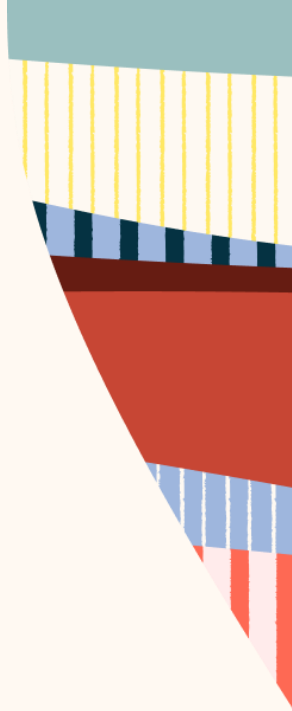
AI Compute

- Training, Machine learning, Model creation
- Computationally intense, large data sets

AI Integration

- Making Inferences
- Connecting inferences to business logic and business data
- Connecting AI Compute processes

Java and AI Today



AI Integration

Several new AI projects in Java

LangChain4J, SpringAI, Semantic Kernel Java, etc

REST calls which perform LLM inference

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

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

AI Integration

Several new AI projects in Java

LangChain4J, SpringAI, Semantic Kernel Java, etc

REST calls which perform LLM inference

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

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

"Elevate your AI Solutions with Java" – LRN2932 – 3:30pm, Marco Polo 805

Oracle Cloud Infrastructure SDK for Java

Enables you to easily write code to manage OCI Resources

Support for OCI AI Related Services:

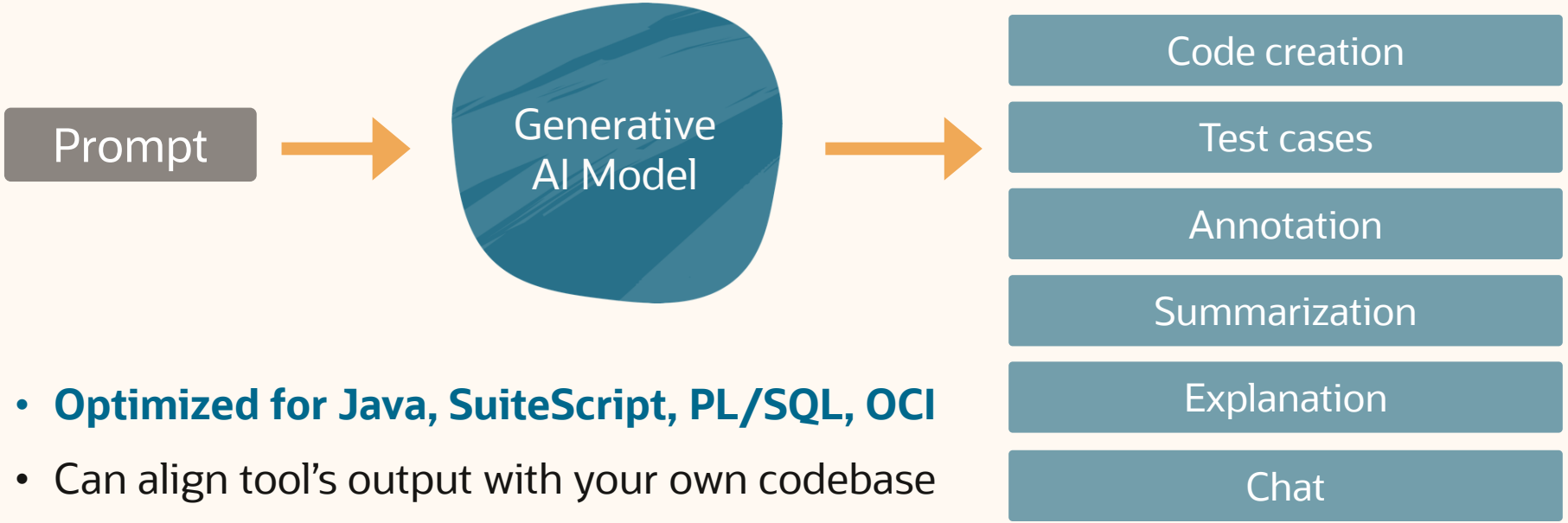
- AI Anomaly Detection
- AI Language
- AI Speech
- AI Vision
- Generative AI
- Generative Inference

Oracle Code Assist

Build applications and customizations faster with AI

1. Boost developer velocity
2. Enhance code consistency
3. Optimized for Java, SuiteScript, PL/SQL, OCI

Powered by AI models running on OCI



- **Optimized for Java, SuiteScript, PL/SQL, OCI**
- Can align tool's output with your own codebase
- Respect for privacy: customer data is not shared

What is Code Assist and what will it do?

- **Plugin** for JetBrains IntelliJ IDEA and Microsoft Visual Studio Code
- Powered by large language models (LLMs) running on **Oracle Cloud Infrastructure (OCI)**
- Provides **context-specific** suggestions and explanations to help developers stay in flow

What's new with Oracle Code Assist?

- Announcing Oracle Code Assist beta

  Now Available!

  Optimized for Java initially, but available for C, C++, Go, JavaScript, PL/SQL, Python, Ruby, Rust

- Announcing Oracle Code Assist offering for NetSuite customers

  Will help facilitate development when building customizations using SuiteScript

  To be available in 2025

AI Compute

Mostly native libraries for GPU, FPGA

- CUDA, C++, etc.

AI Compute (currently?) tries to spend as little time as possible in platform languages

Runtimes like from ONNX (Open NN eXchange) provide flexible interface into platforms

- Machine learning Java libraries like Tribuo can integrate with ONNX

Platforms must provide

- Easy connection to native libraries, functions and data
- Access to business logic, data and services
- Provide more direct native compute innovations

Java is innovating on both fronts

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

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 – 10 years (!) of R&D

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 – 10 years (!) of R&D

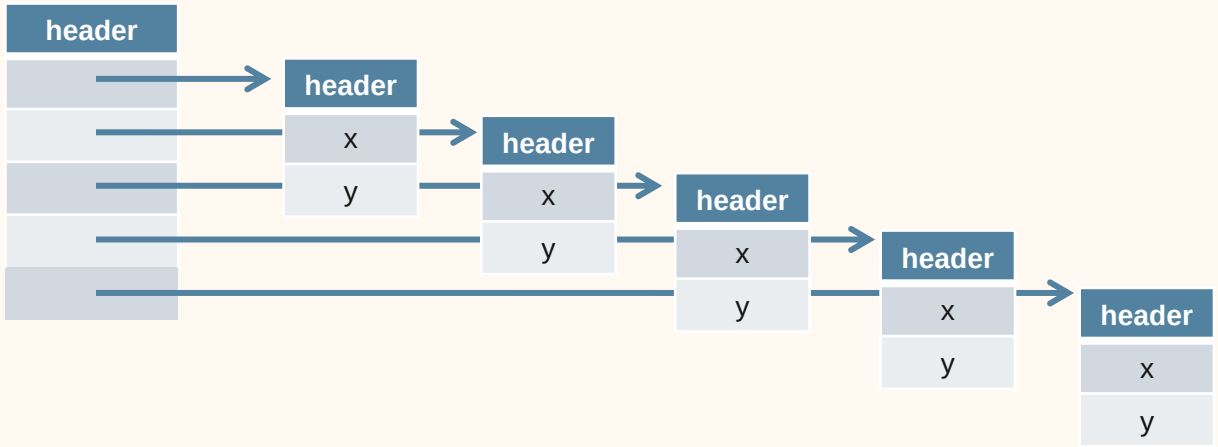
Key Motivators

- 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

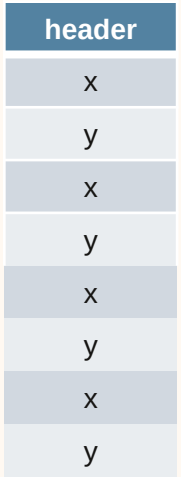
For an array of XY point objects, we get this layout:



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…



Project Valhalla – Unify the type system

Rift between primitives and objects is “Java’s original sin”

- Necessary in 1995 for numeric performance, but dogged us ever since Integer vs int
- 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 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?

Earlier prototypes (2018-2022) offer great performance results

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 22, JavaFX 22, JMC 9

Early access: JDK 24, JDK 23, JavaFX 24, JavaFX 23, Jextract, Leyden, Loom, & Valhalla

Valhalla - Where Are We? #JVMLS

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

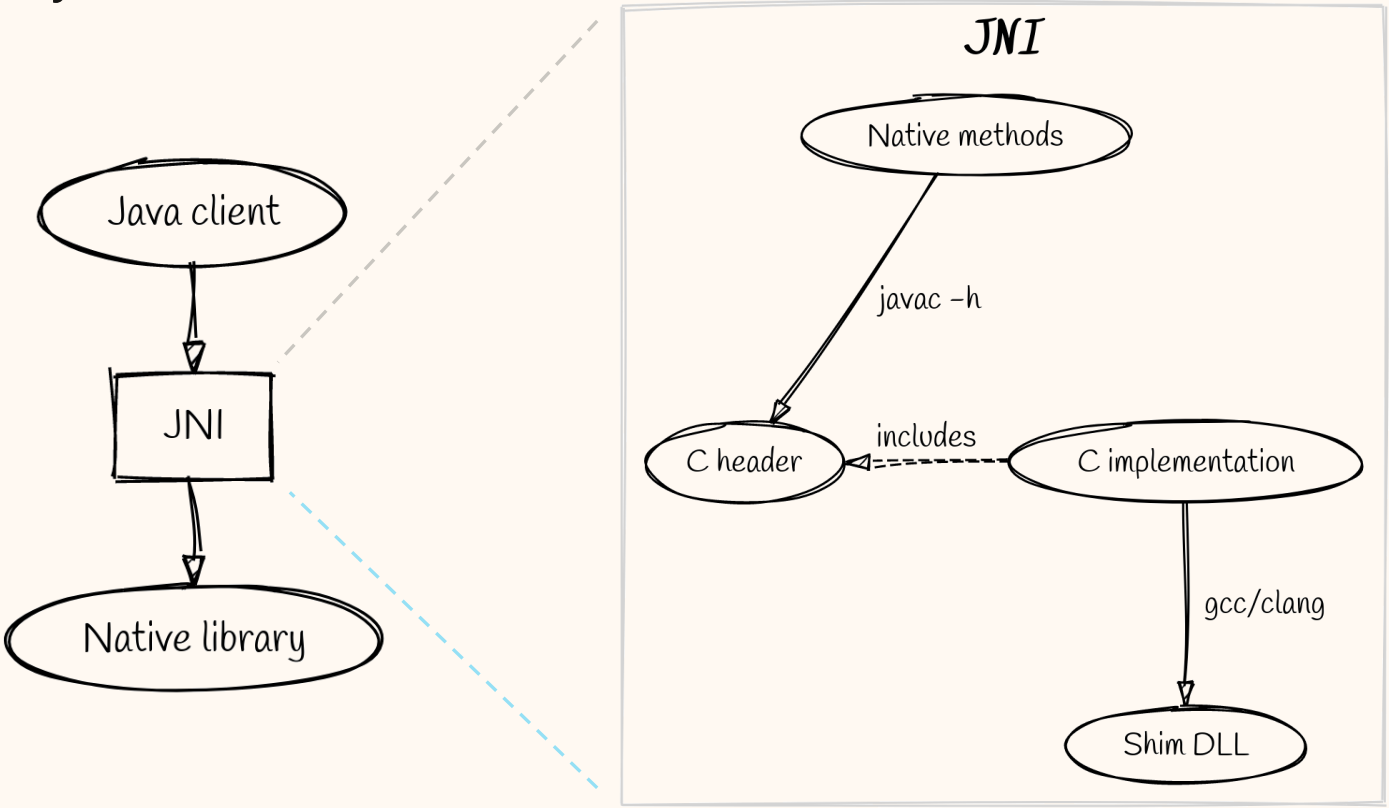
- Off-CPU computing (Cuda, OpenCL)
- Machine Learning (Blis, ONNX, Tensorflow)
- Graphics processing (OpenGL, Vulkan, DirectX)
- Others (CRIU, fuse, OpenSSL)

Enter Project Panama

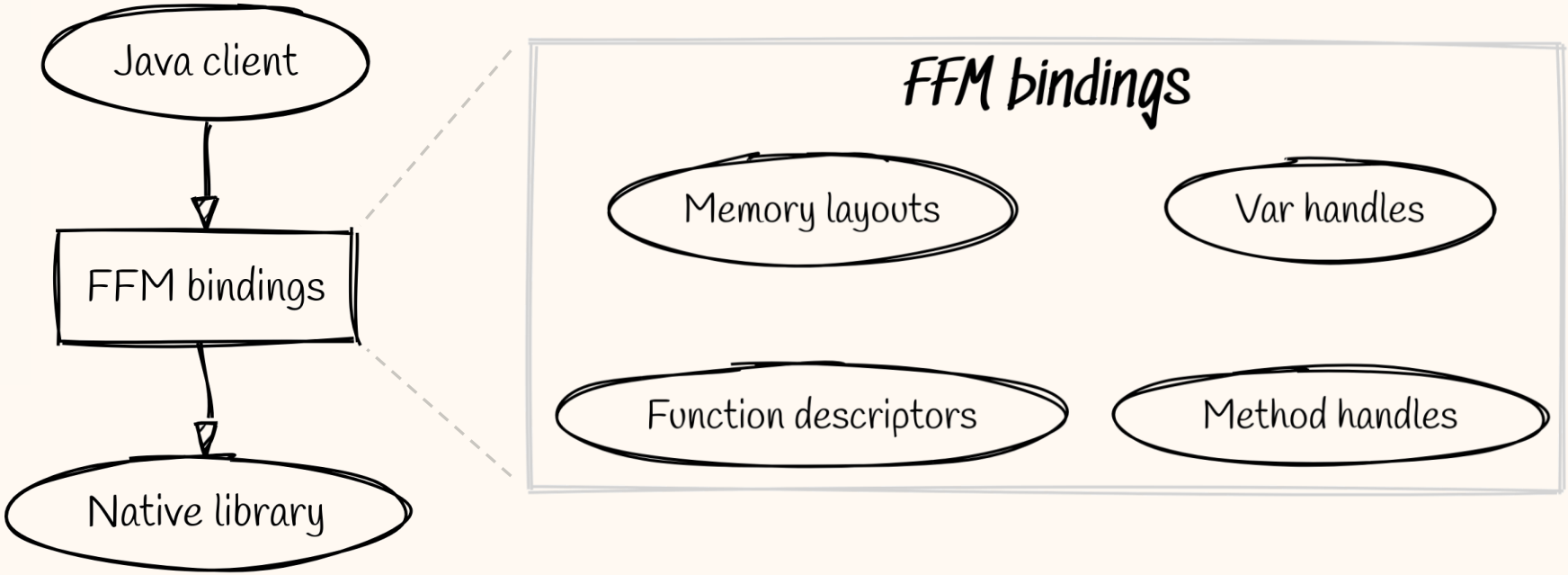
JEP 454: Foreign Function & Memory API

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

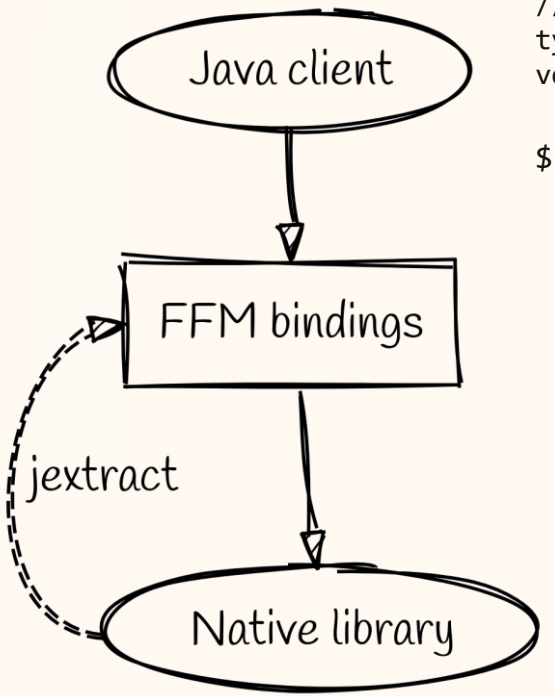
Legacy JNI workflow



FFM API workflow



Enter jextract




```
// stdlib.h
typedef int (*__compar_fn_t) (const void *, const void *);
void qsort (void *__base, size_t __nmemb, size_t __size, __compar_fn_t __compar);

$ jextract --target-package org.stdlib /usr/include/stdlib.h
```

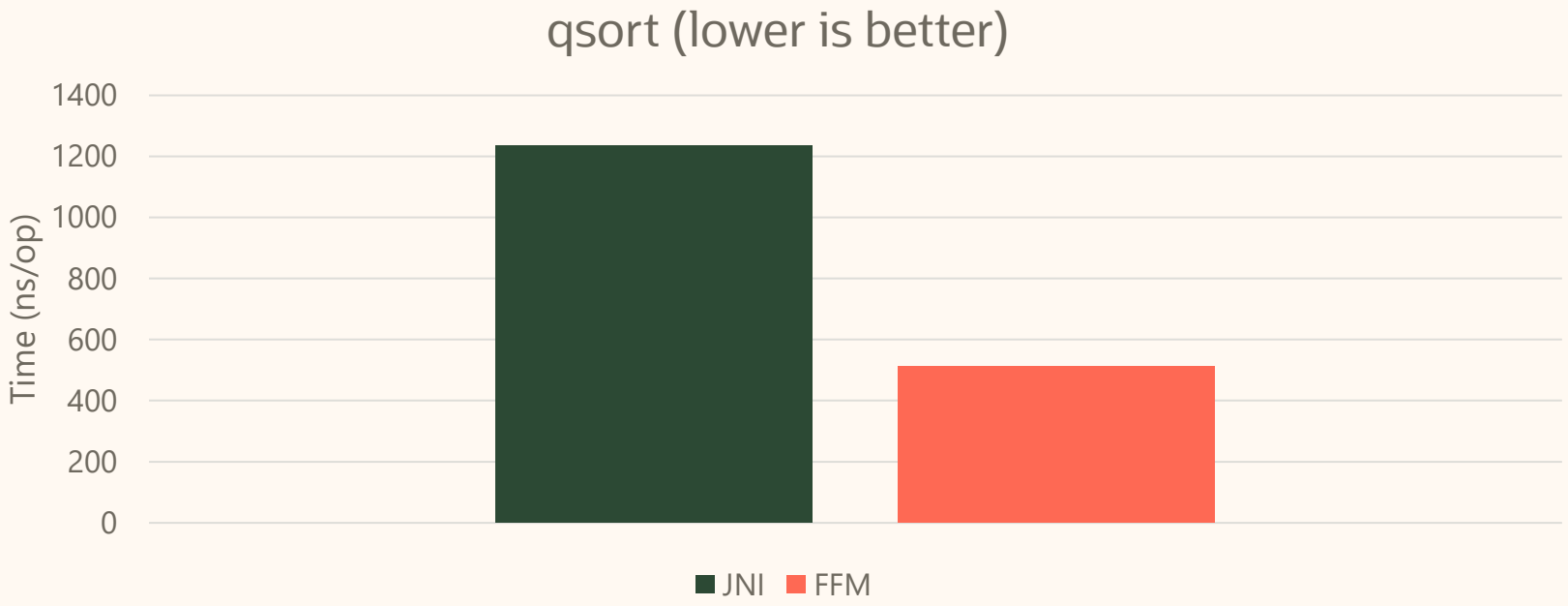
```
import static org.stdlib.stdlib_h.*;
…

try (Arena offHeap = Arena.ofConfined()) {
    MemorySegment array = offHeap.allocateFrom(
            C_INT, 0, 9, 3, 4, 6, 5, 1, 8, 2, 7);

    var compareFunc = __compar_fn_t.allocate((a1, a2) ->
            Integer.compare(a1.get(C_INT, 0), a2.get(C_INT, 0)), offHeap);
    qsort(array, 10L, 4L, comparFunc);

    int[] sorted = array.toArray(JAVA_INT);
// [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
}
```

Performance



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

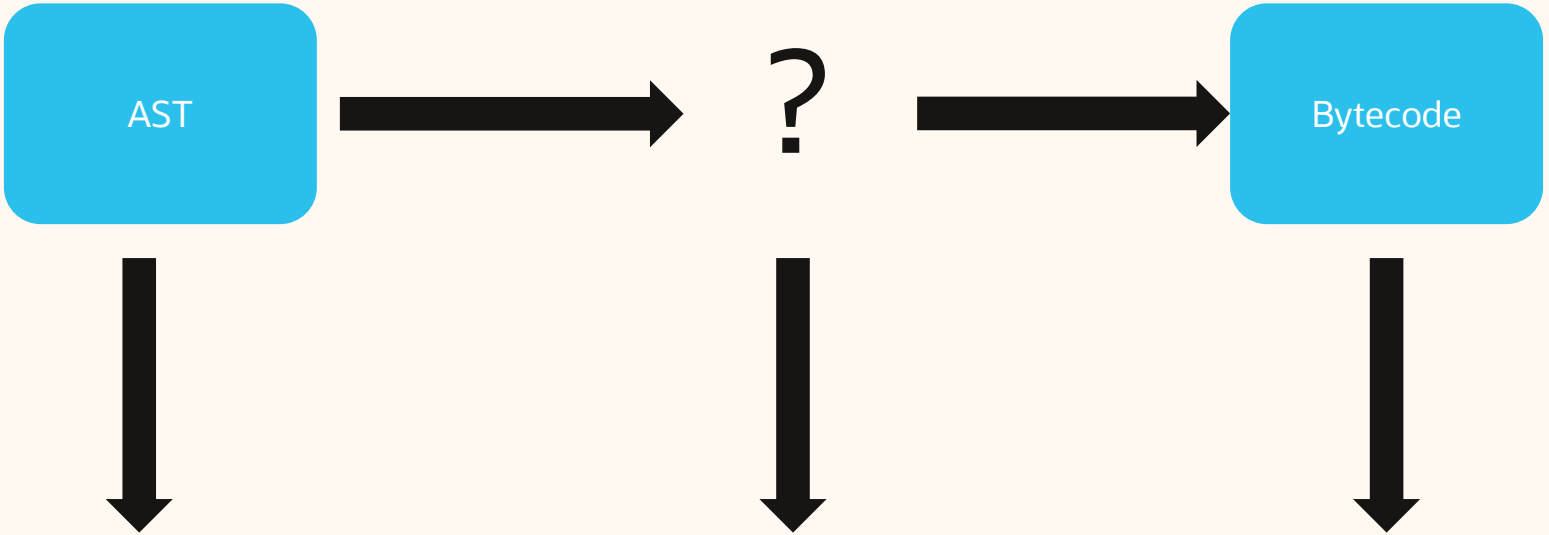
Project Babylon – GPU Code Example

```
// 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;
};
```

Project Babylon – Spectrum of possibilities



Java compiler and annotation processors

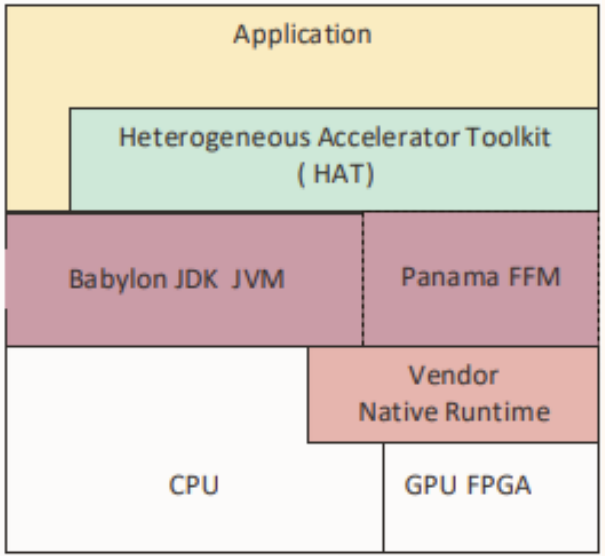
Code model analyzers and transformers

Java Virtual Machine

Heterogenous Accelerator Toolkit (HAT)

Leverage programming model from Babylon

Leverage FFM API from 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

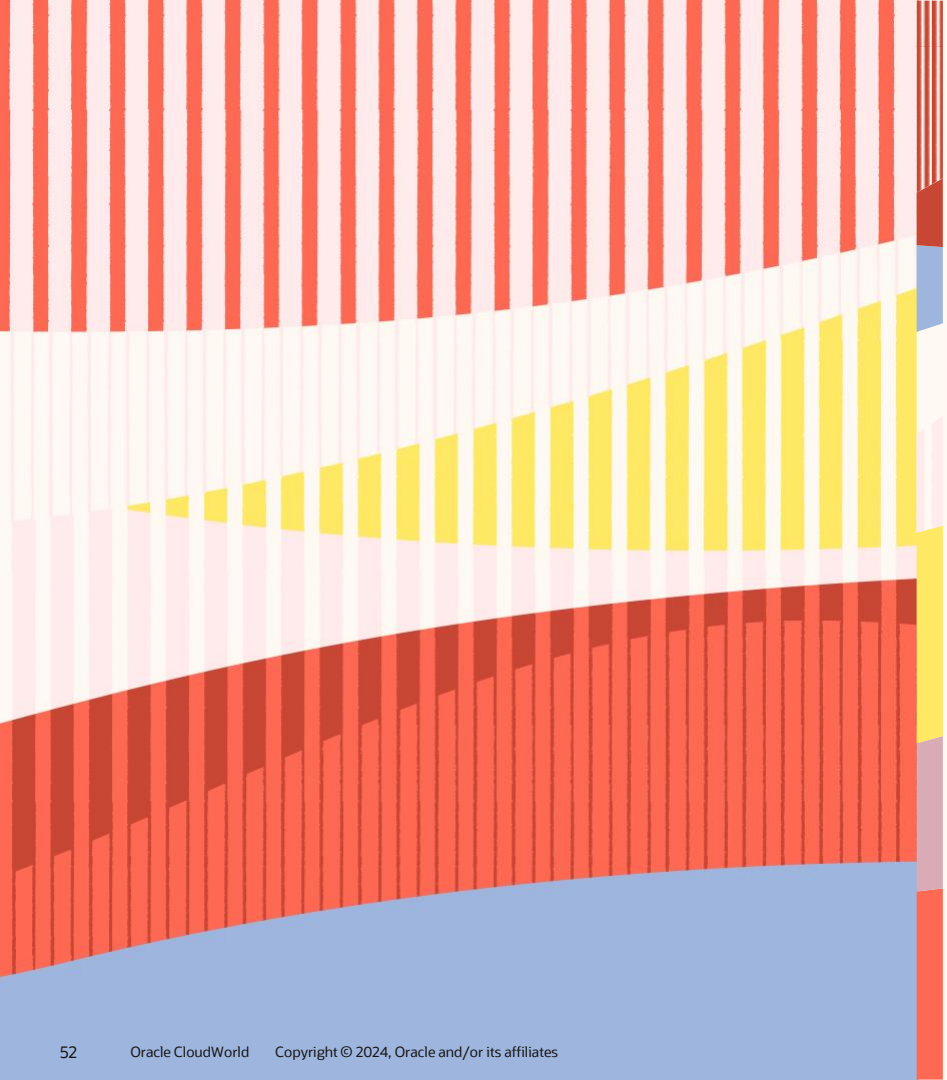
Extend the reach of Java including to machine learning models and GPUs

Panama

Safe and efficient interoperation with native libraries and native Java.

Valhalla

Unify primitives and classes to improve productivity and performance.



ORACLE
CloudWorld

Thank You

Everything you need to know about Artificial Intelligence and the Java Platform

Please complete attendee survey!

