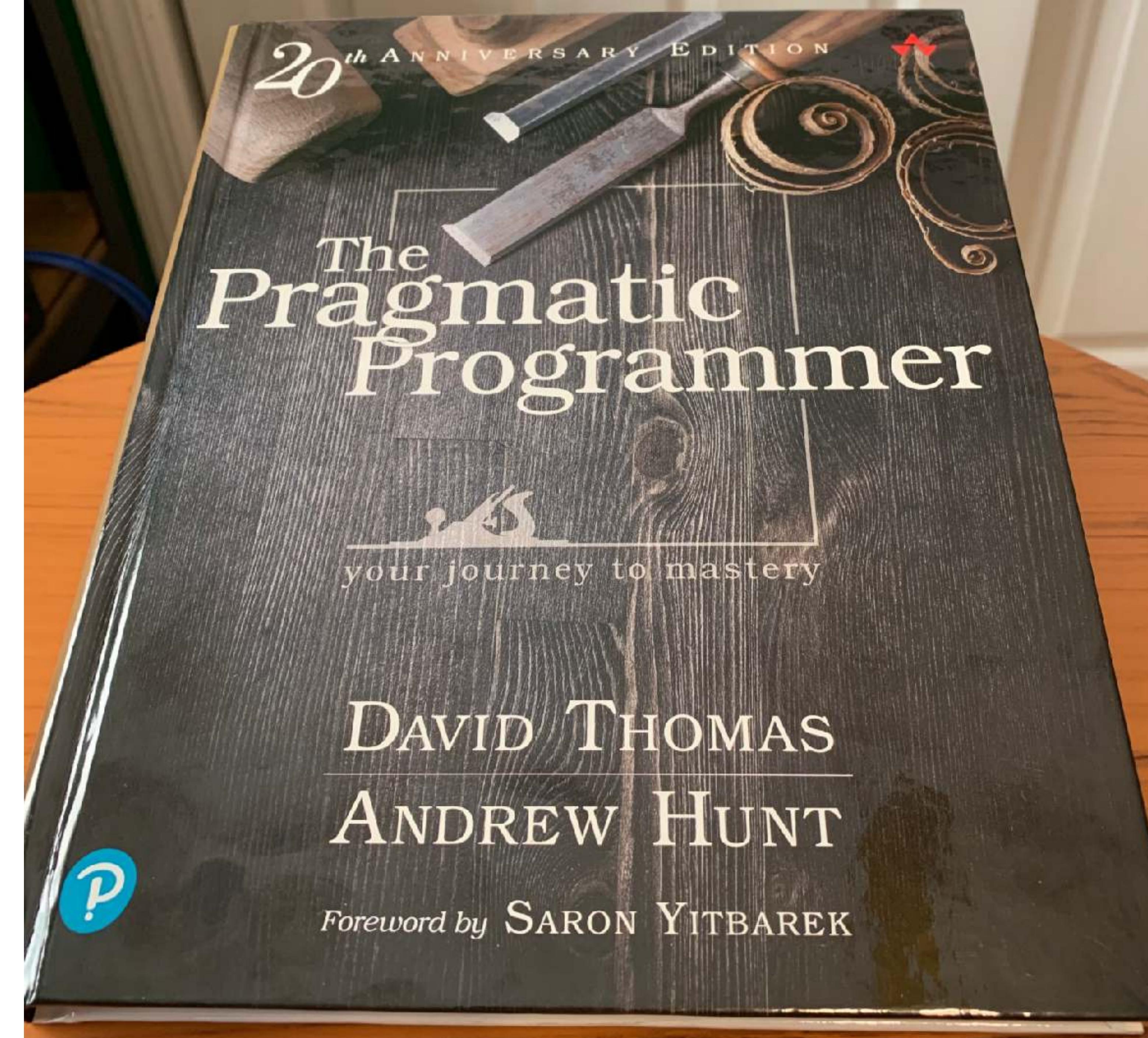


JVM Iceberg

We need to go deeper

Artur Skowroński

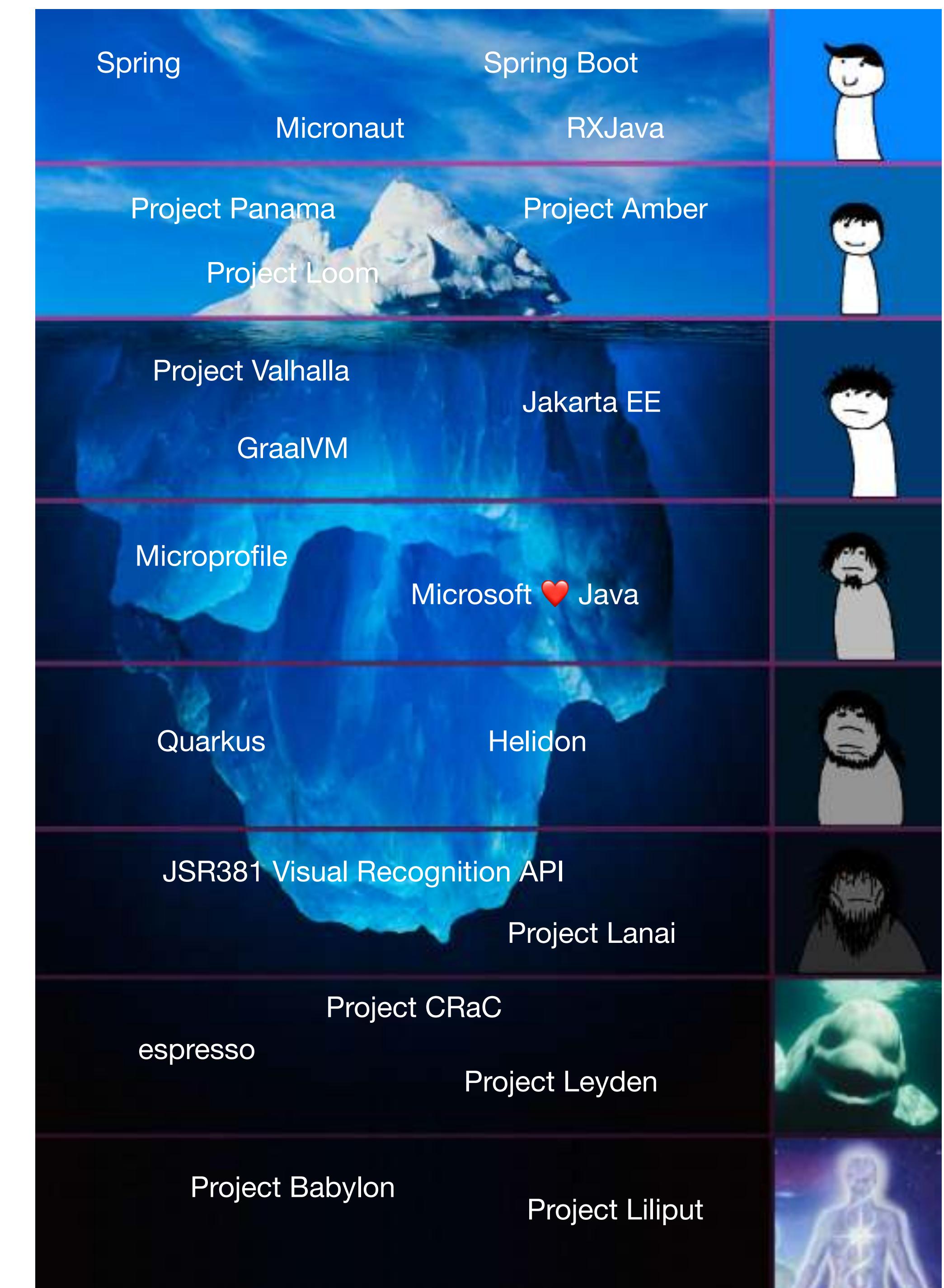






Black Hope Curse







140 Slides / ? Minutes

20 Topics





I CAN DO THIS ALL DAY



MEH.

Spring

Spring Boot

Micronaut

RXJava

Project Panama

Project Amber

Project Loom



Celem Projektu Amber jest zbadanie i inkubacja **mniejszych**, zorientowanych na **produktywność** funkcji języka Java.

Project Amber



286: Local-Variable Type Inference (var)

361: Switch Expressions

378: Text Blocks

427: Pattern Matching for switch (Third Preview)

395: Records

Project Amber





Project Amber



```
public class HelloWorld {  
    public static void main(String[ ] args) {  
        System.out.println("Hello World");  
    }  
  
    void main() {  
        println("Hello World");  
    }  
}
```



Project Amber



OpenJDK

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GAYEA Builds

Mailing lists
Wiki · IRC
Bylaws · Census
Legal

JEP Process
Source code
Mercurial
GitHub

Tools
Git
Jtreg harness

Groups
(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler

Conformance
Core Libraries
Governing Board
HotSpot

IDE Tooling & Support
Internationalization

JMX
Members
Networking
Porters
Quality
Security
Serviceability
Vulnerability
Web

Projects
(overview)
Amber
Annotations Pipeline
2.0
Audio Engine
Build Infrastructure
CRaC
Caciocavallo
Closures
Code Tools

JEP draft: Implicit Classes and Enhanced Main Methods (Preview)

Owner: Ron Pressler

Type: Feature

Scope: SE

Status: Draft

Component: specification/language

Effort: S

Created: 2023/02/13 13:58

Updated: 2023/02/15 17:50

Issue: 8302326

Summary

Evolve the Java language so that students can write their first programs without needing to understand language features designed for large programs. Far from using a separate dialect of Java, these first programs utilize streamlined declarations for single-class programs, and can be seamlessly expanded to incorporate more advanced features as needed. This is a preview language feature.

Goals

- Offer a smooth on-ramp to Java so that educators can introduce programming concepts in a gradual manner.
- Help students to write basic programs in a concise manner and grow the code gracefully as their skills grow.
- Reduce the ceremony of writing simple programs such as scripts and command-line utilities.

Non-Goals

- It is not a goal to introduce a separate "beginner's dialect" of Java or a new tool to run student programs. Indeed, that is an anti-goal to be avoided.
- Similarly, it is not a goal to introduce a separate beginners' toolchain; student programs should be compiled and run with all the same tools that compile and run any Java program.

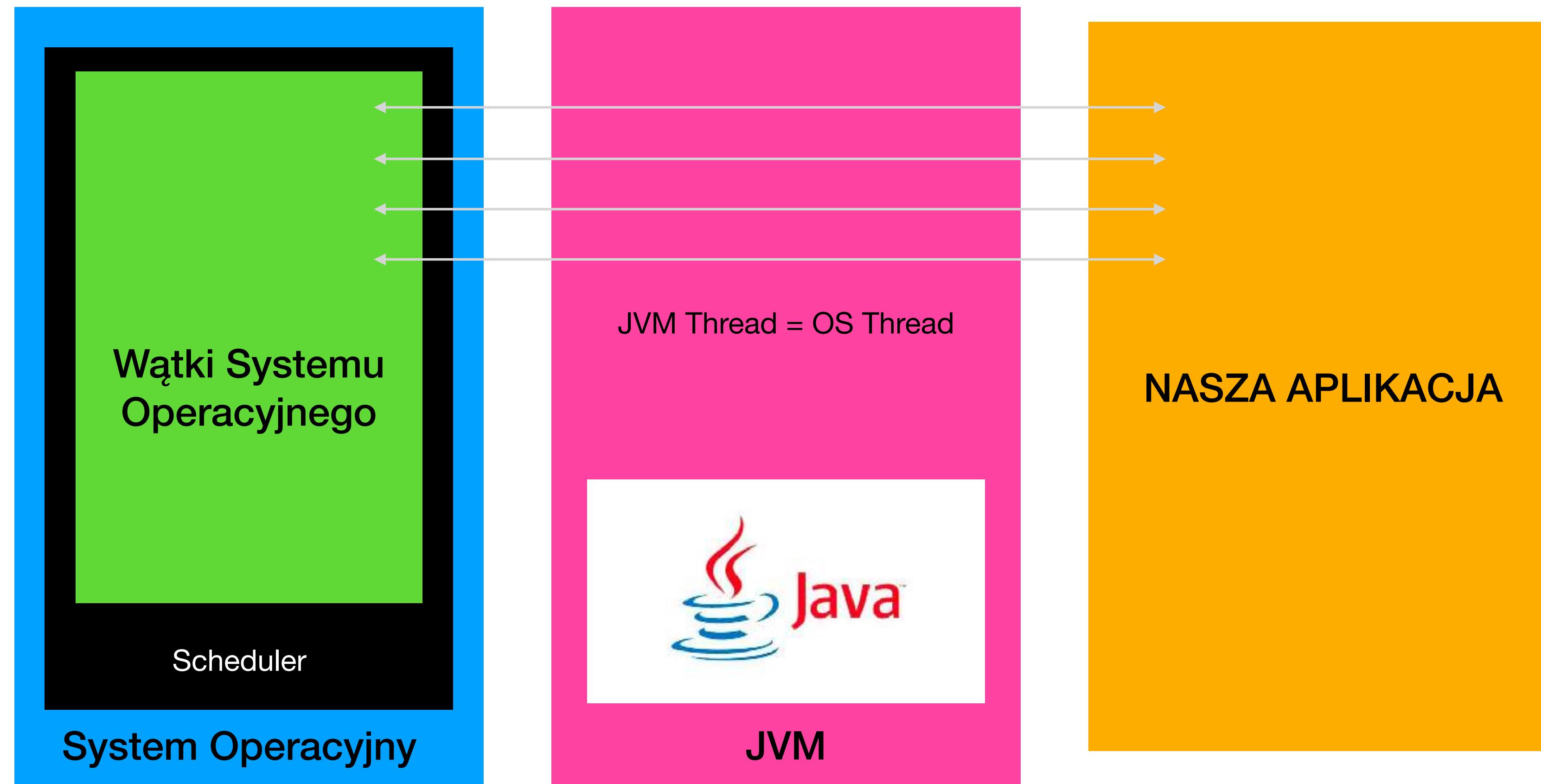
Project Amber



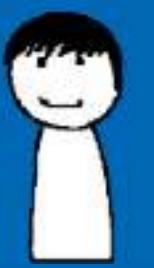
Celem **Project Loom** jest stworzenie
wysokowydajnego, lekkiego modelu
współbieżności w Javie.

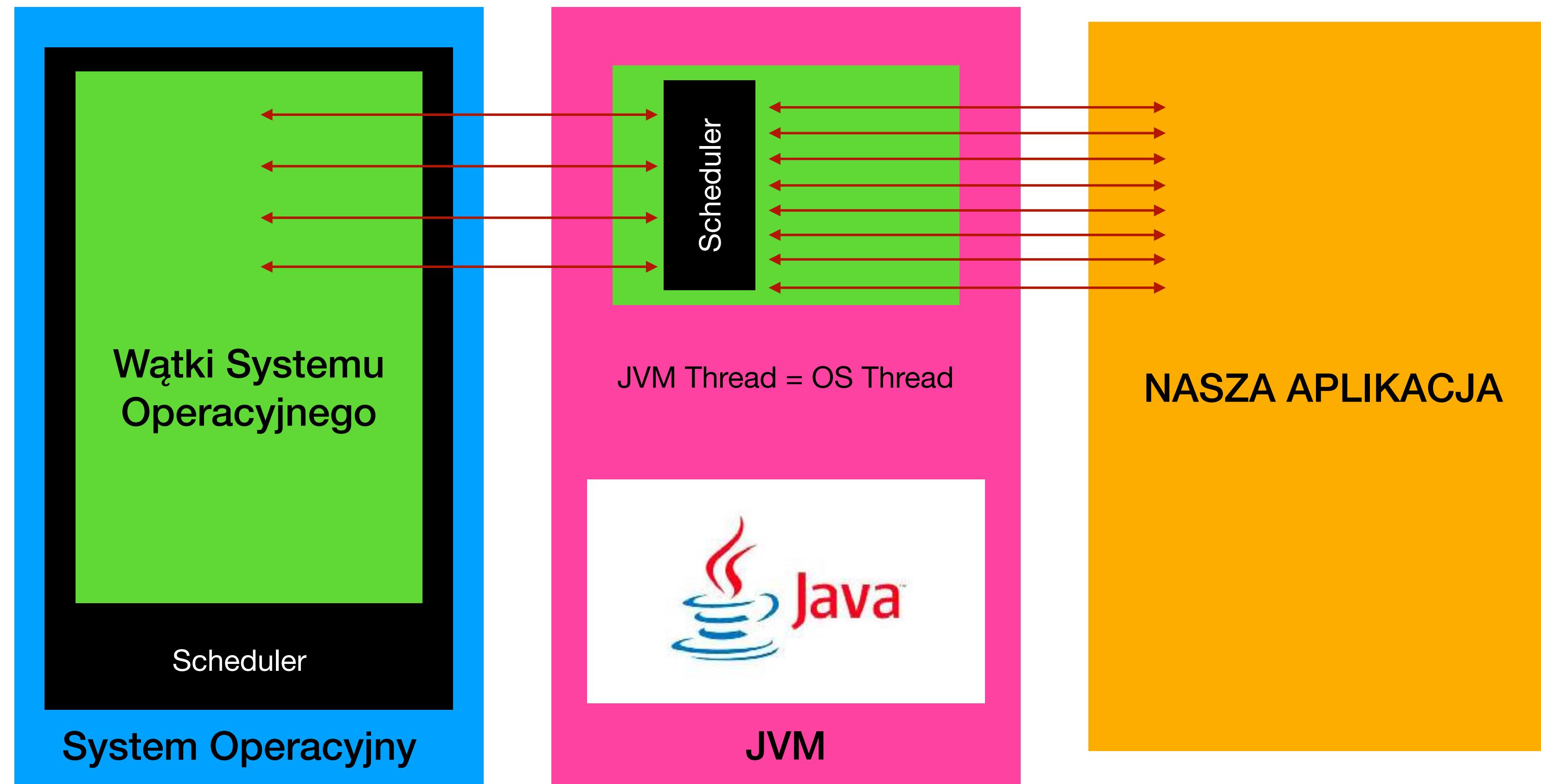
Project Loom



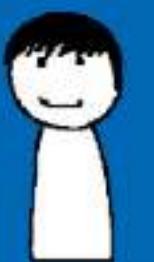


Project Loom





Project Loom





Project Loom



Structure Concurrency

Virtual Threads

Scope Locals

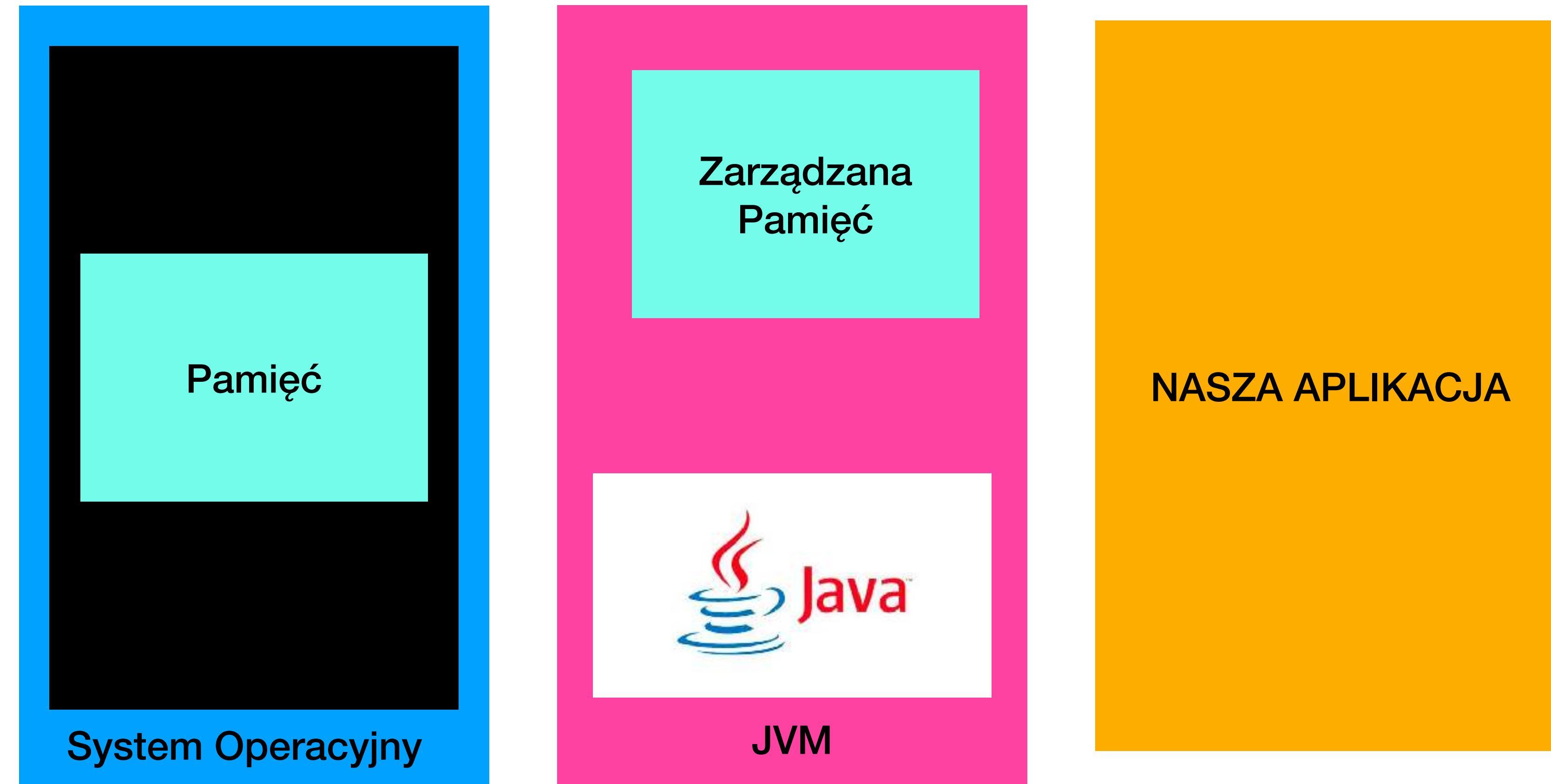
Project Loom



Celem **Project Panama** jest upraszczenie procesu łączenia programów w **Java** z komponentami **nie-Javowymi**

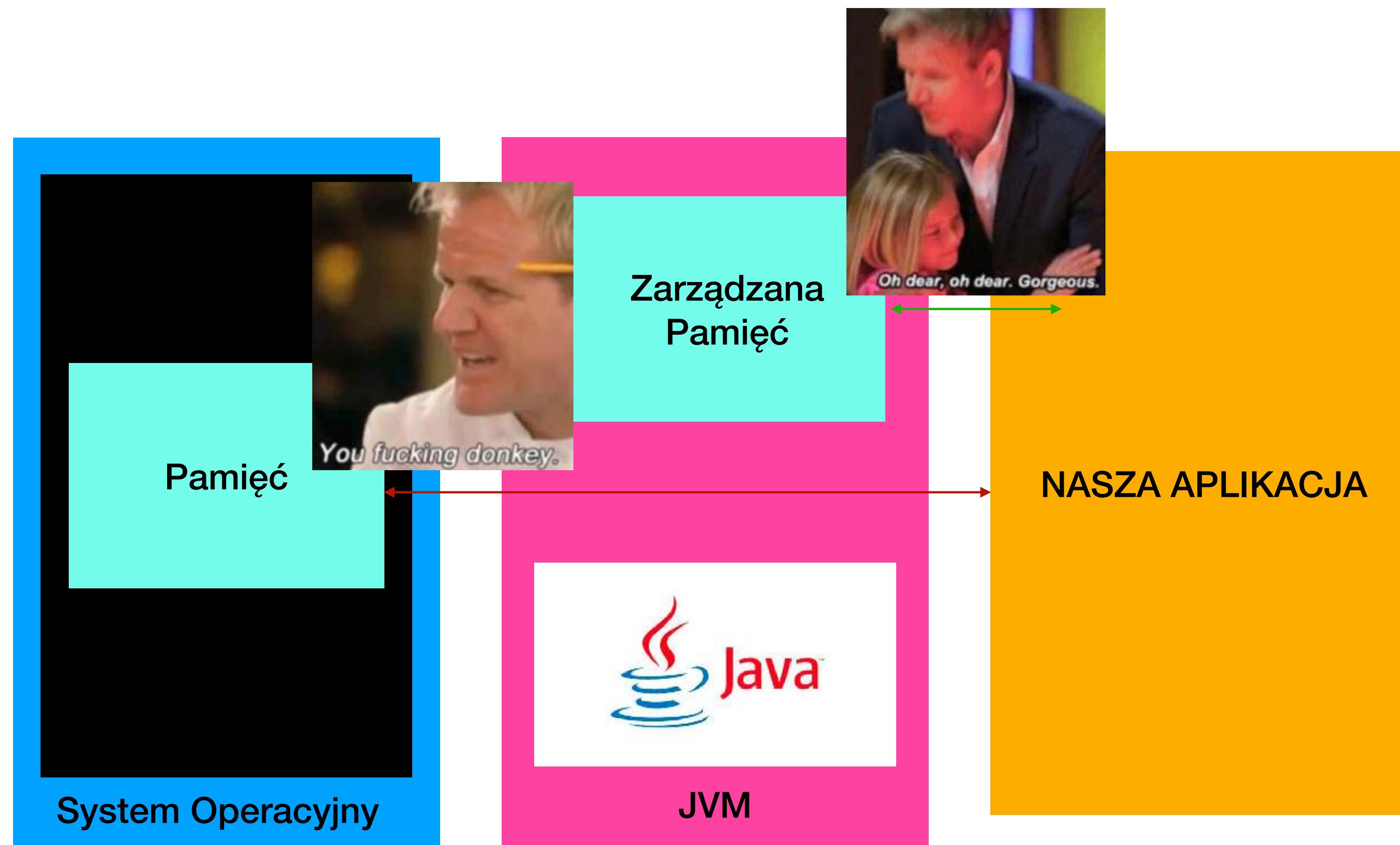
Project Panama



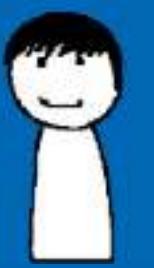


Project Panama





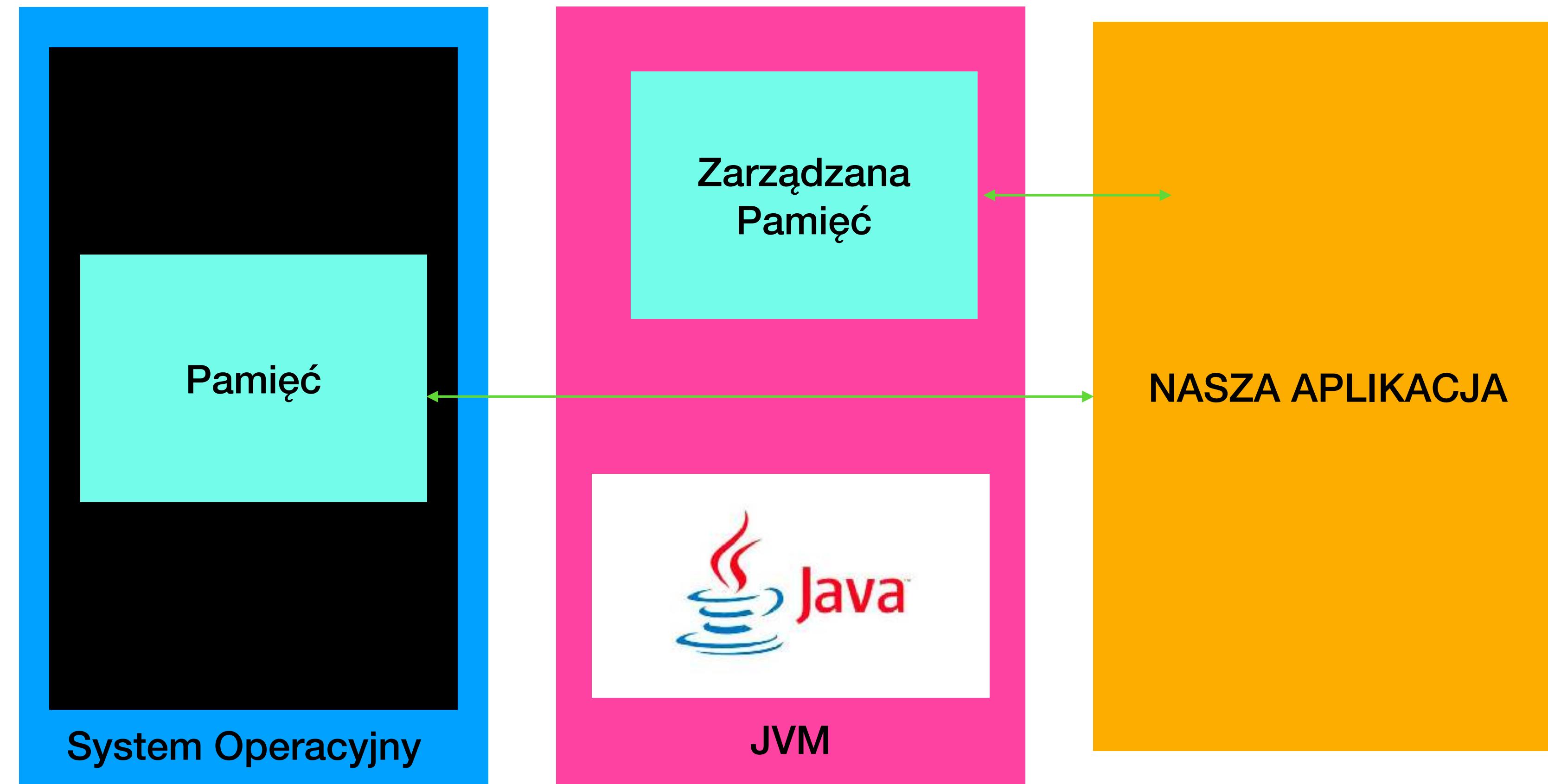
Project Panama



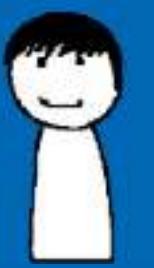


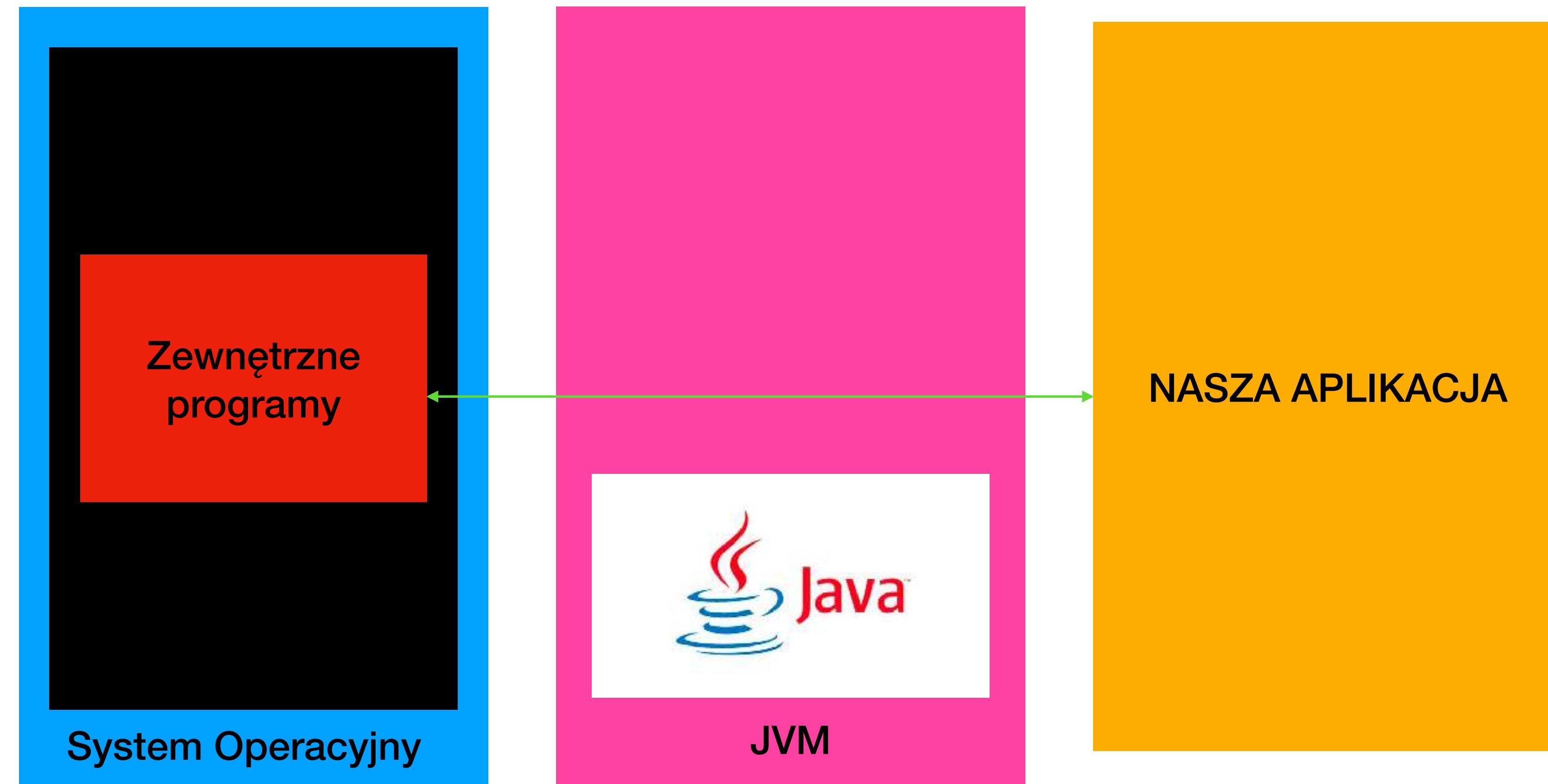
Project Panama



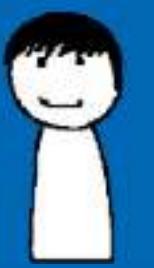


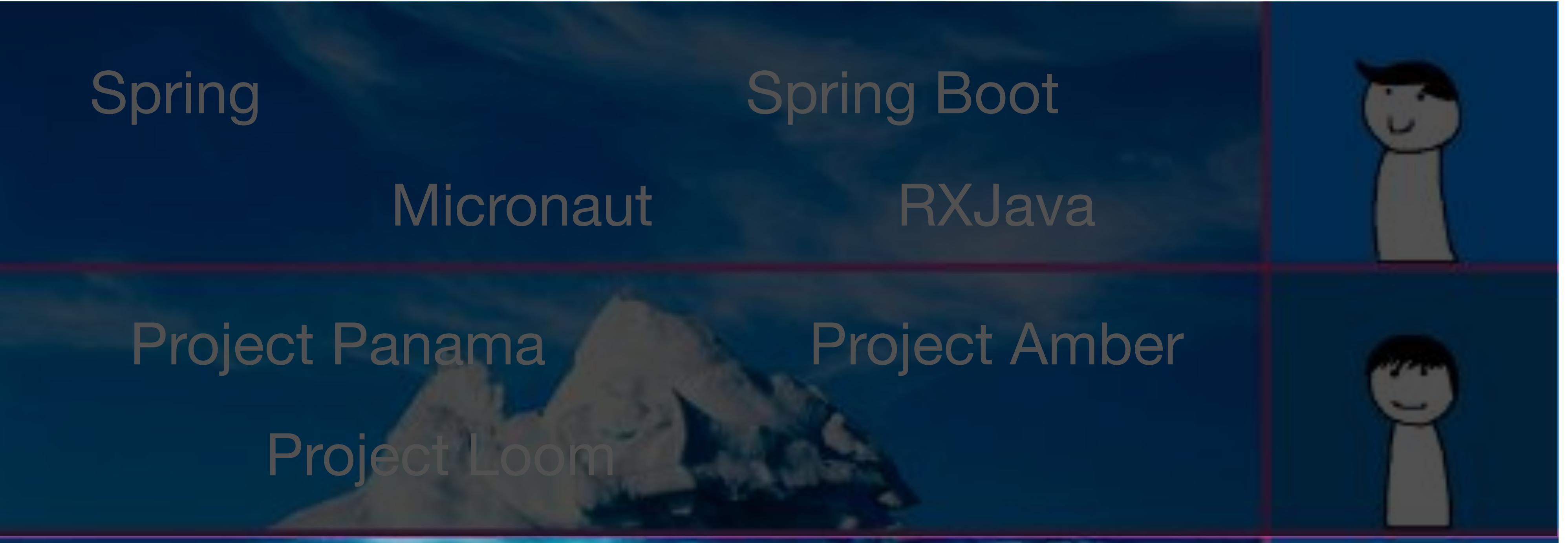
Project Panama





Project Panama





Spring

Micronaut

Project Panama

Project Loom

Project Valhalla

GraalVM

Spring Boot

RXJava

Project Amber

Jakarta EE



**GRANDPA SAID VALHALLA WILL BE
SO COOL**



I'M STILL WAITING

memegenerator.net

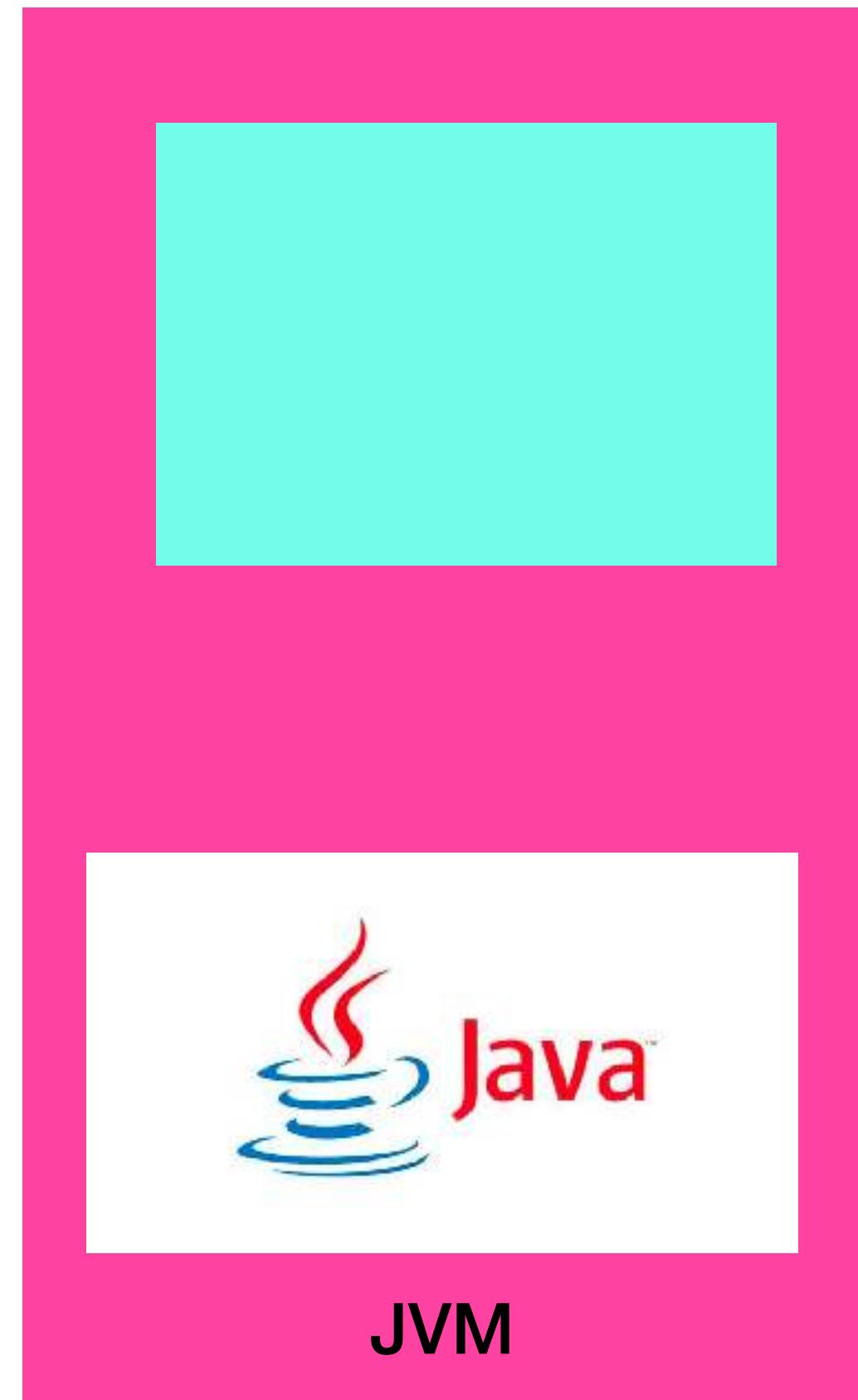
Project Valhalla





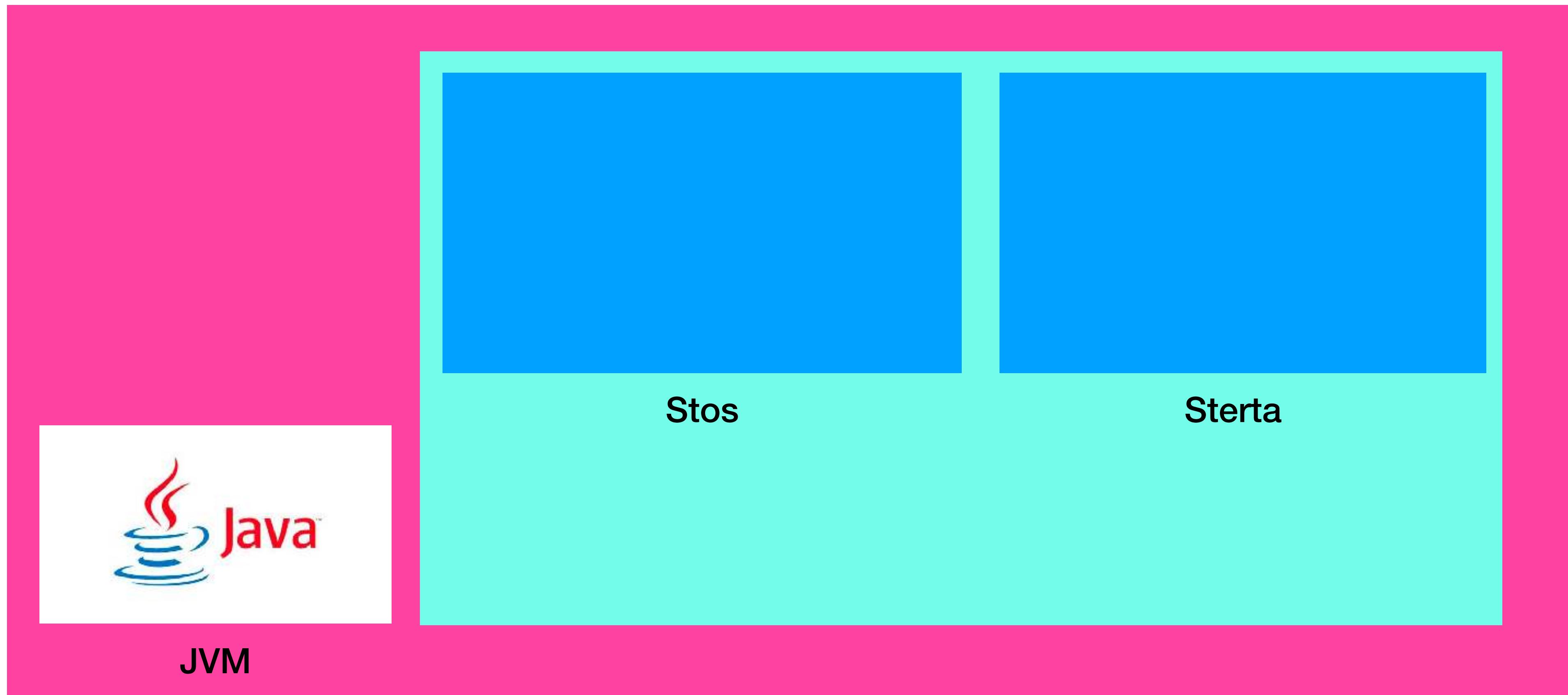
Project Valhalla





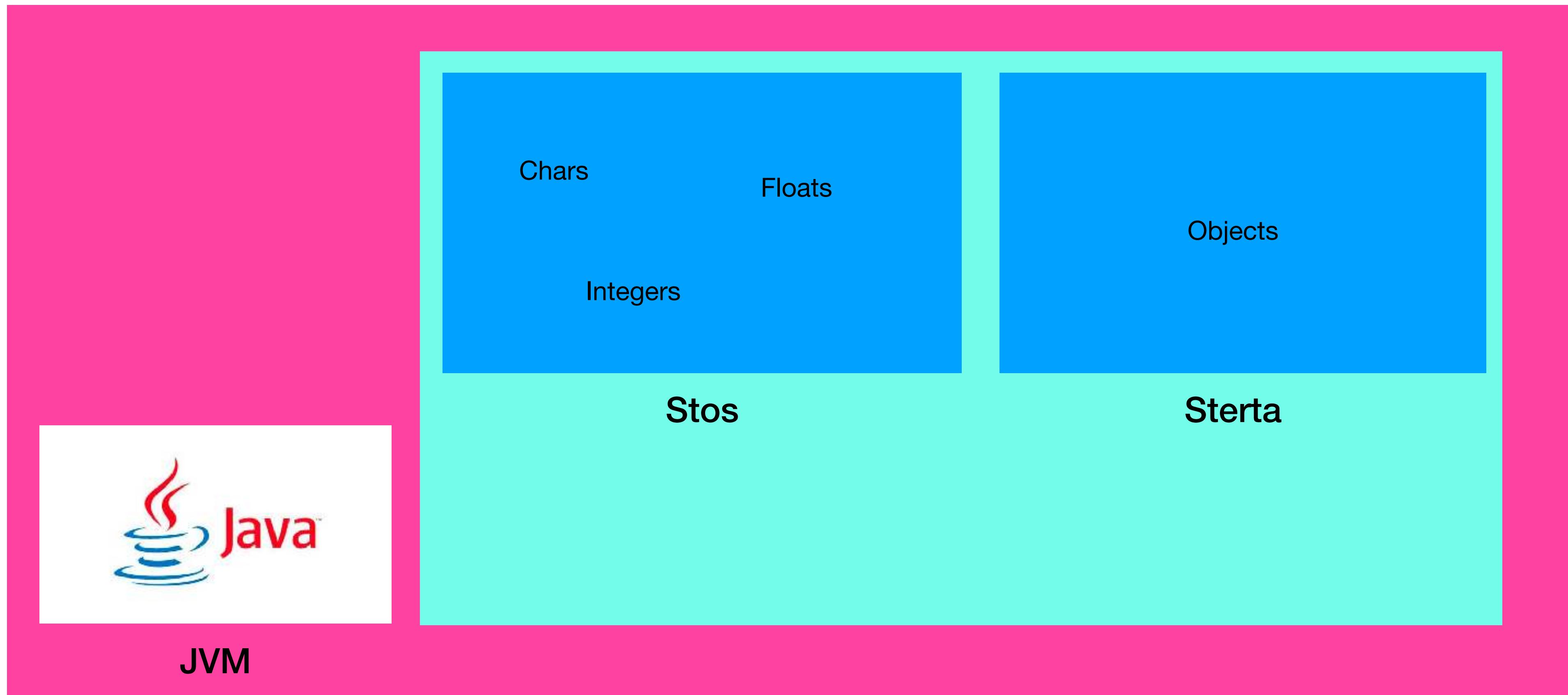
Project Valhalla





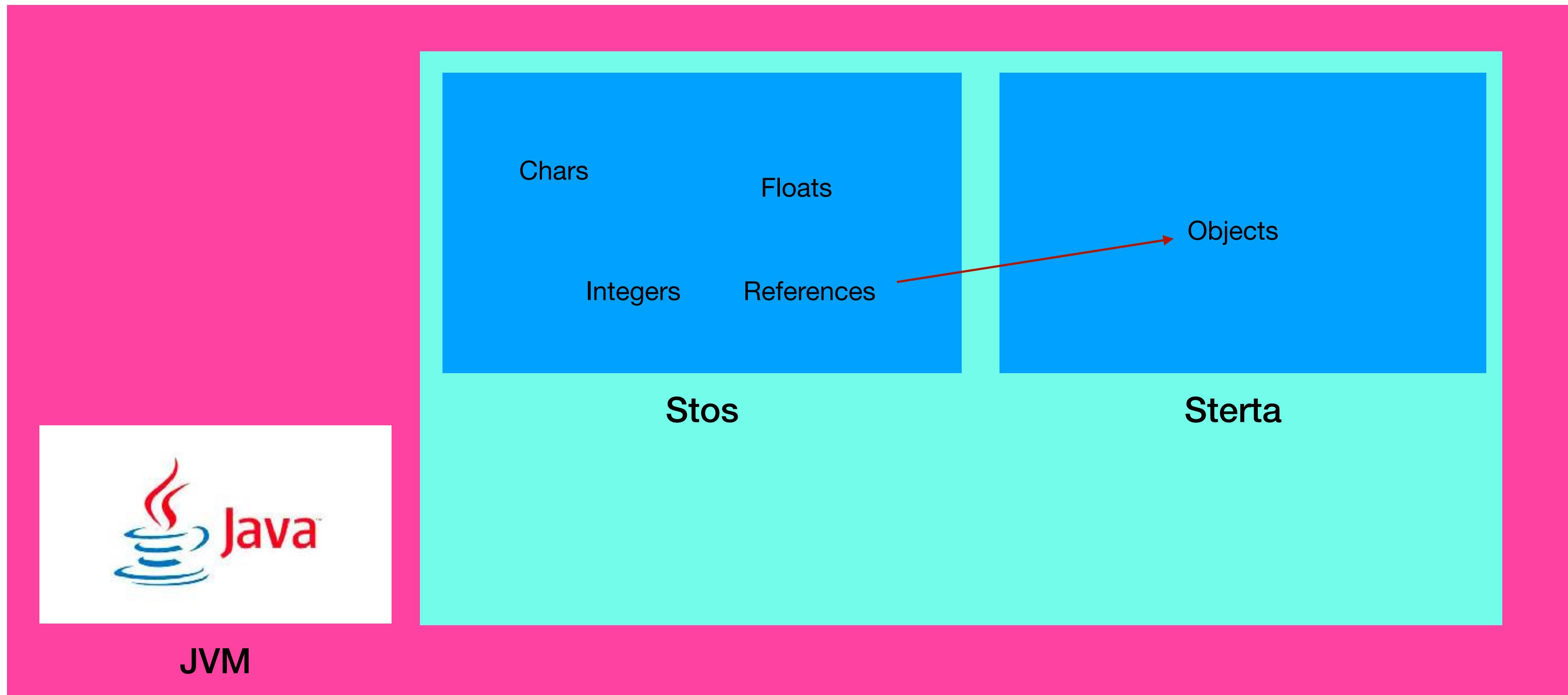
Project Valhalla





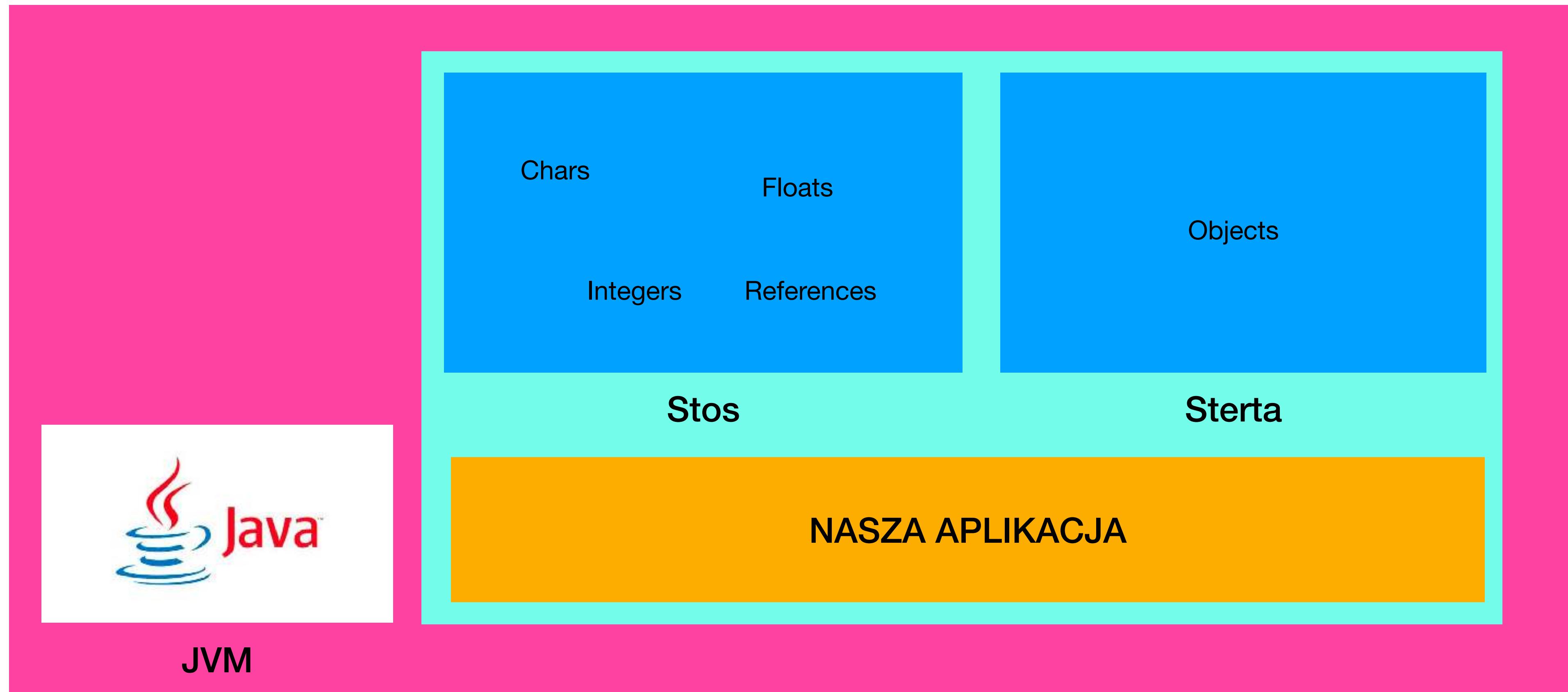
Project Valhalla





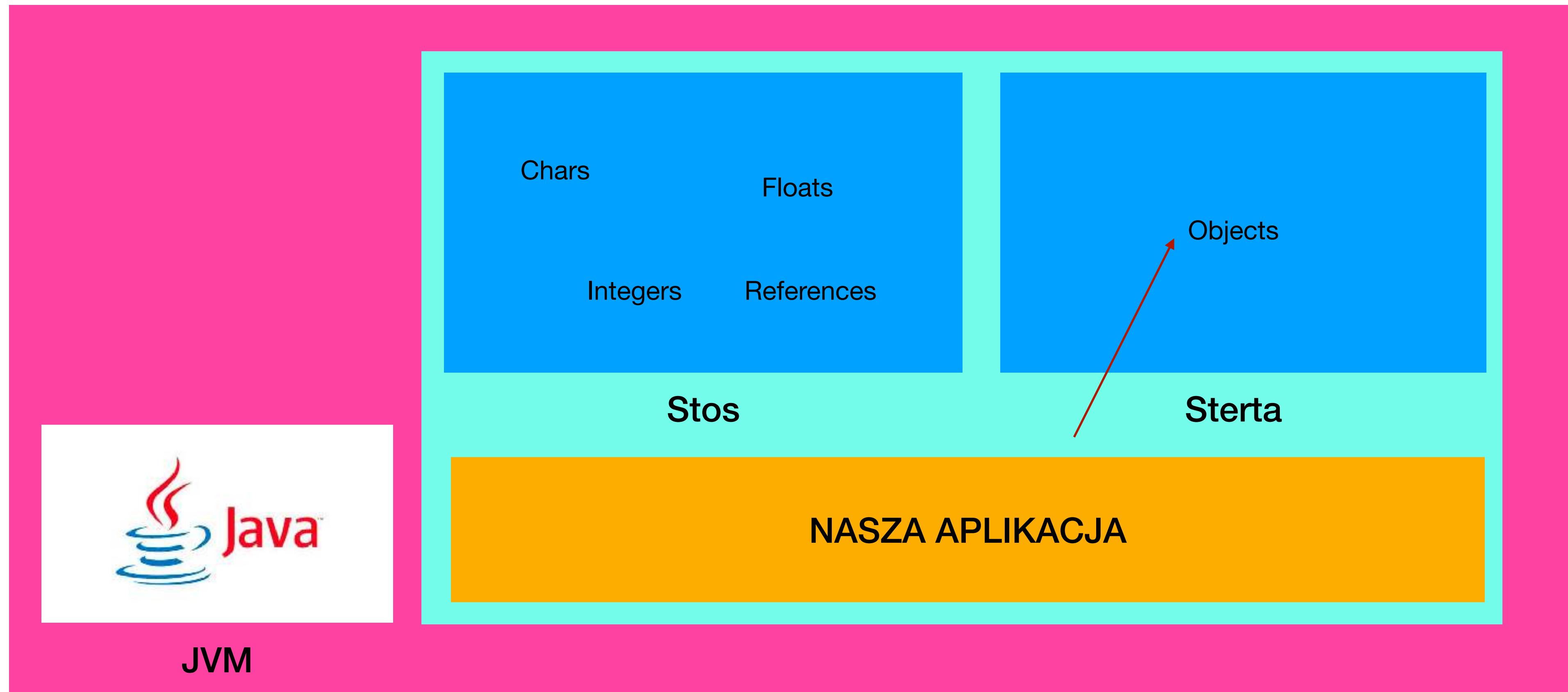
Project Valhalla





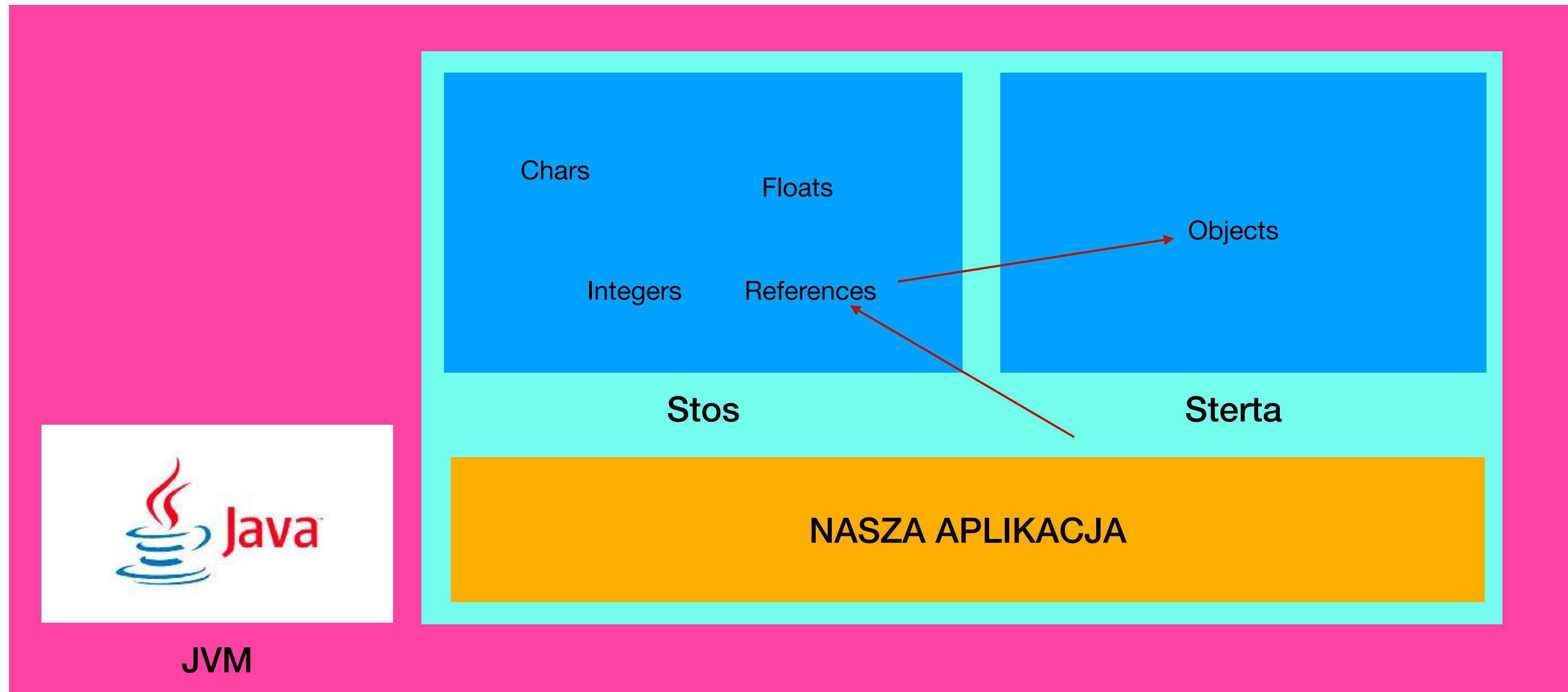
Project Valhalla





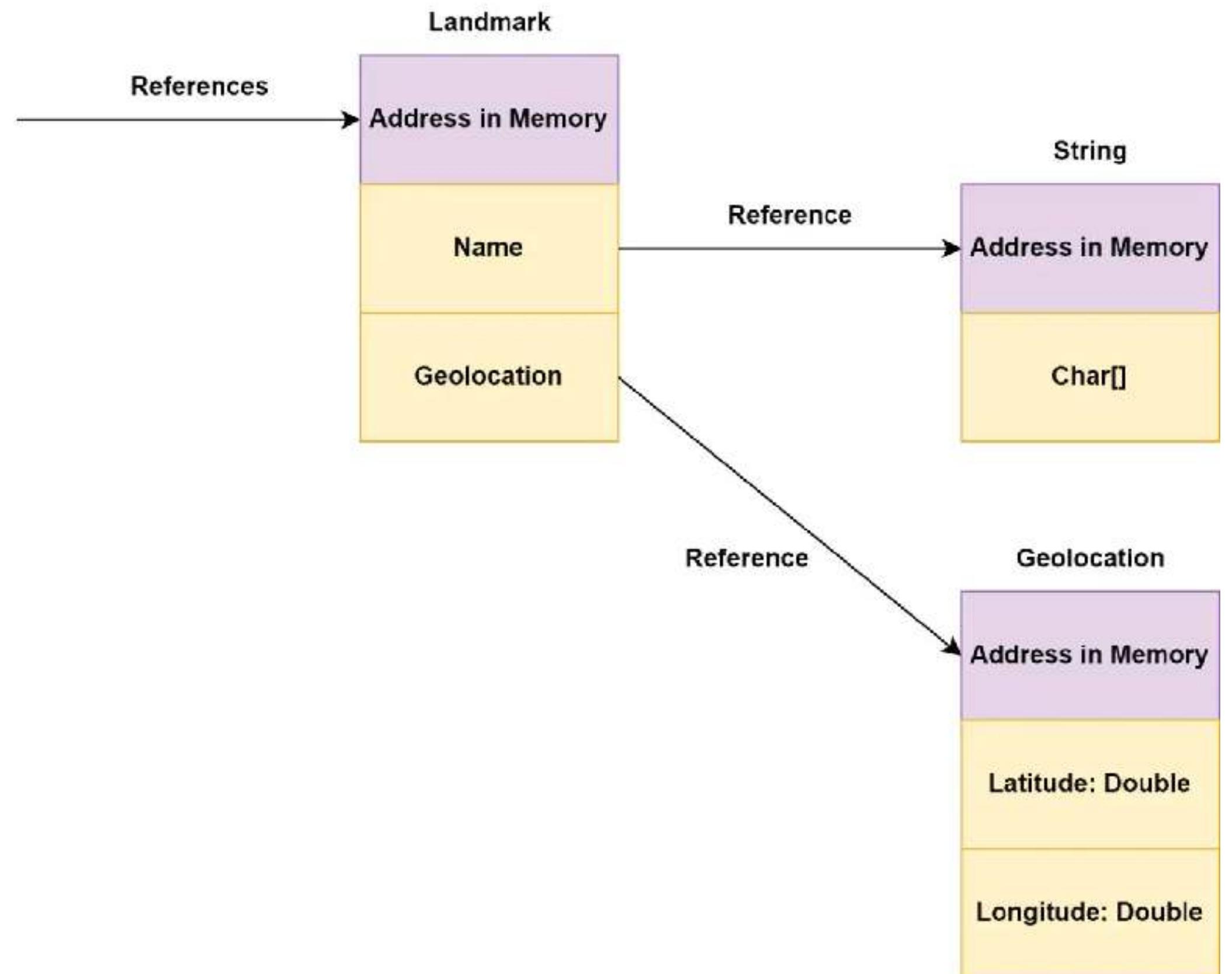
Project Valhalla





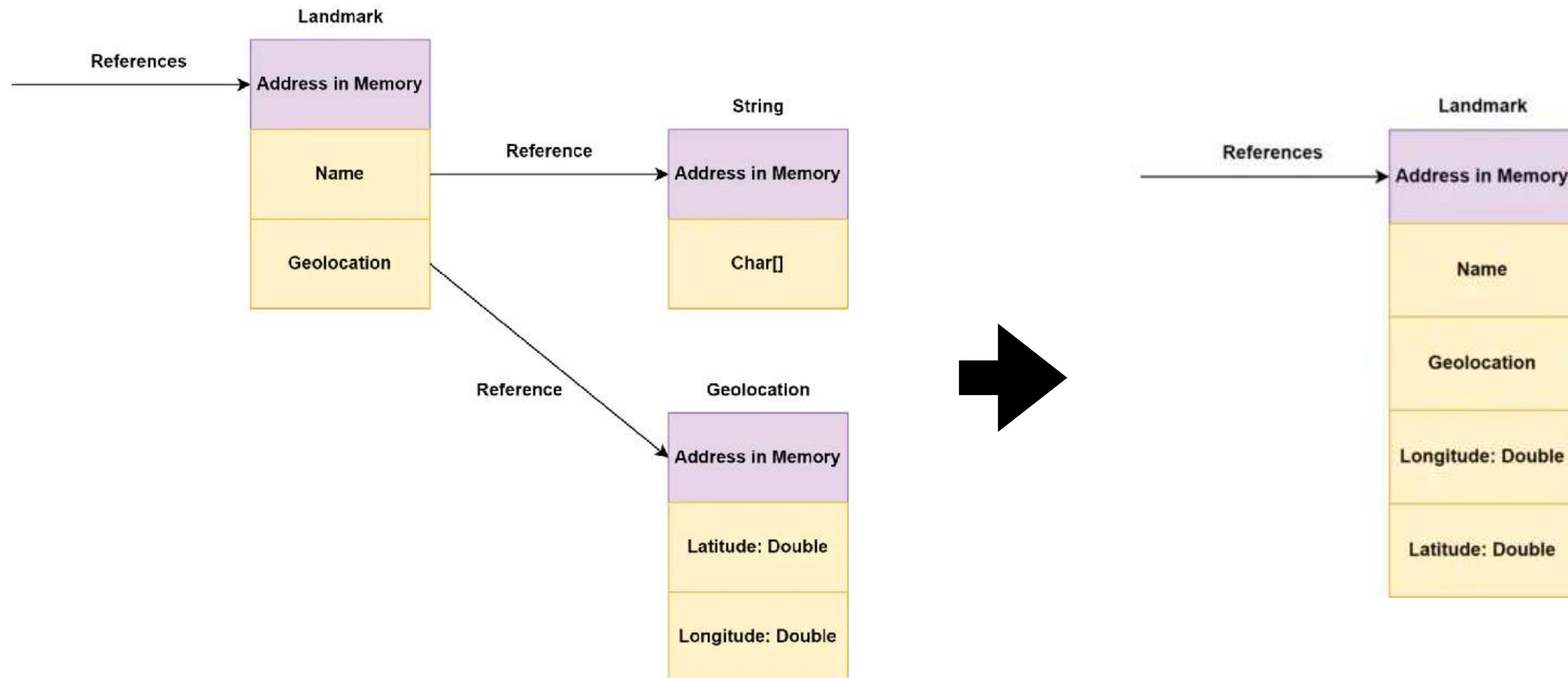
Project Valhalla





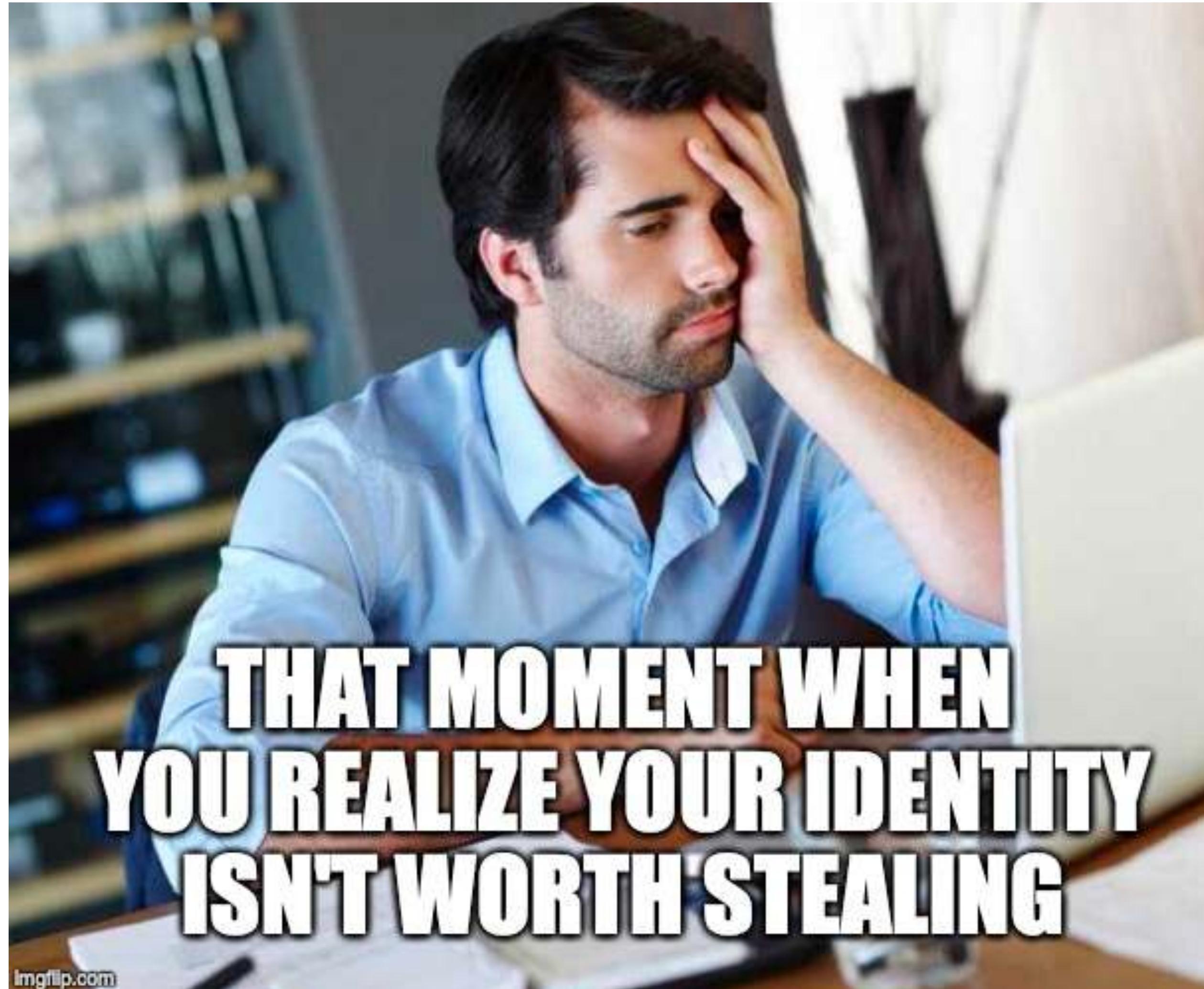
Project Valhalla





Project Valhalla



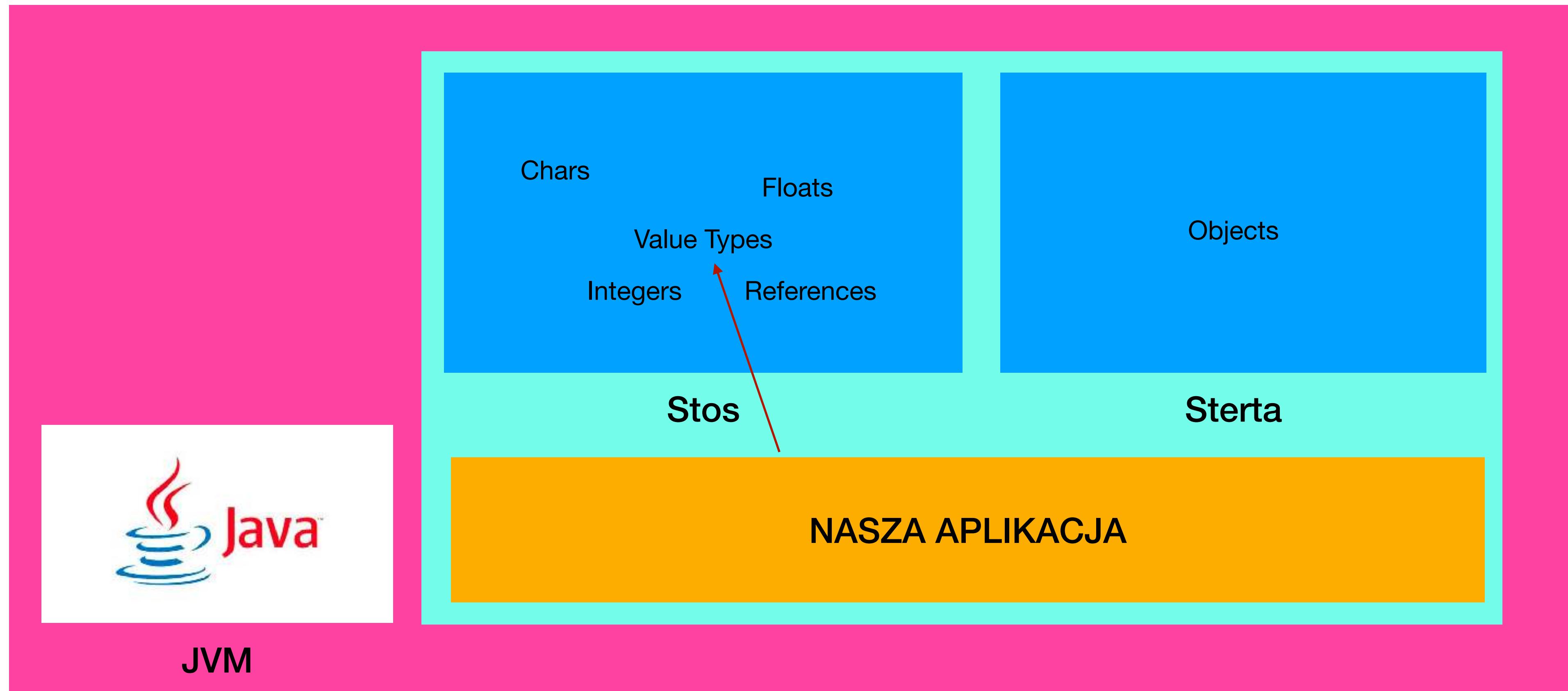


**THAT MOMENT WHEN
YOU REALIZE YOUR IDENTITY
ISN'T WORTH STEALING**

imgflip.com

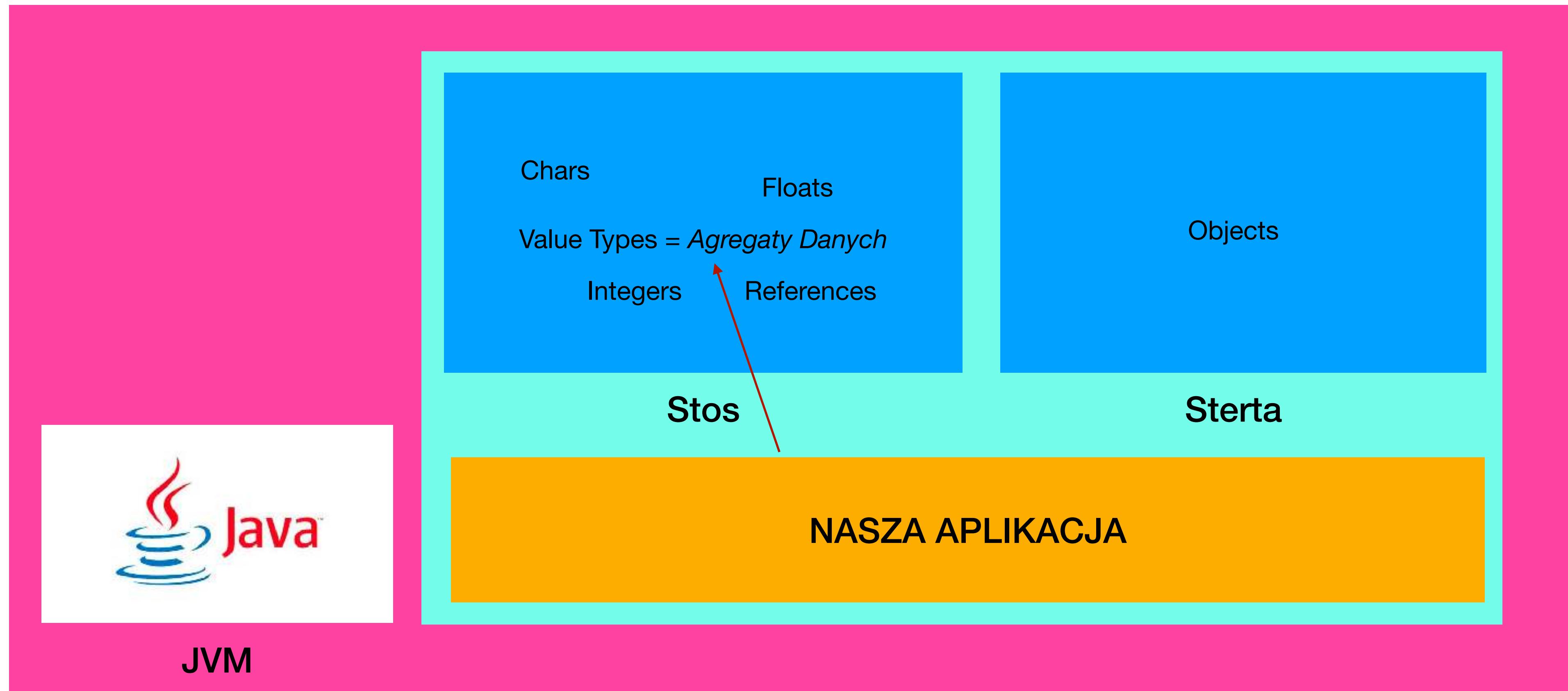
Project Valhalla





Project Valhalla





Project Valhalla





SOCIETY IF

VALHALLA EVER DELIVER

imgflip.com

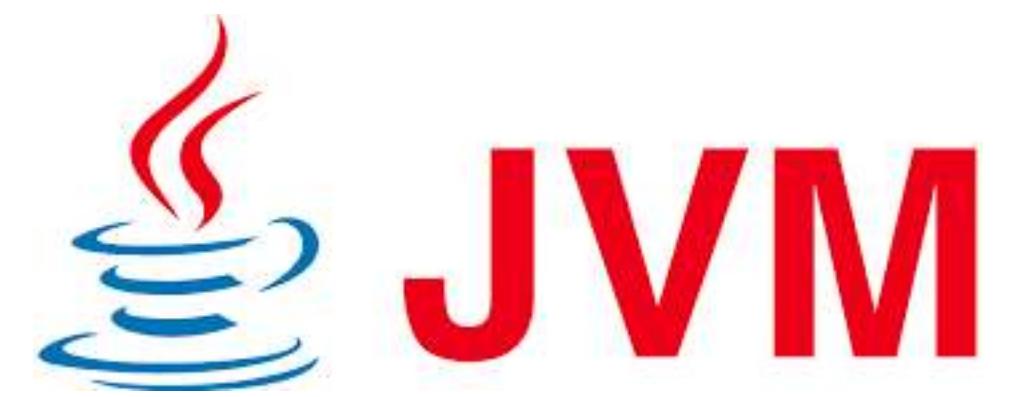
Project Valhalla





GraalVM





GraalVM

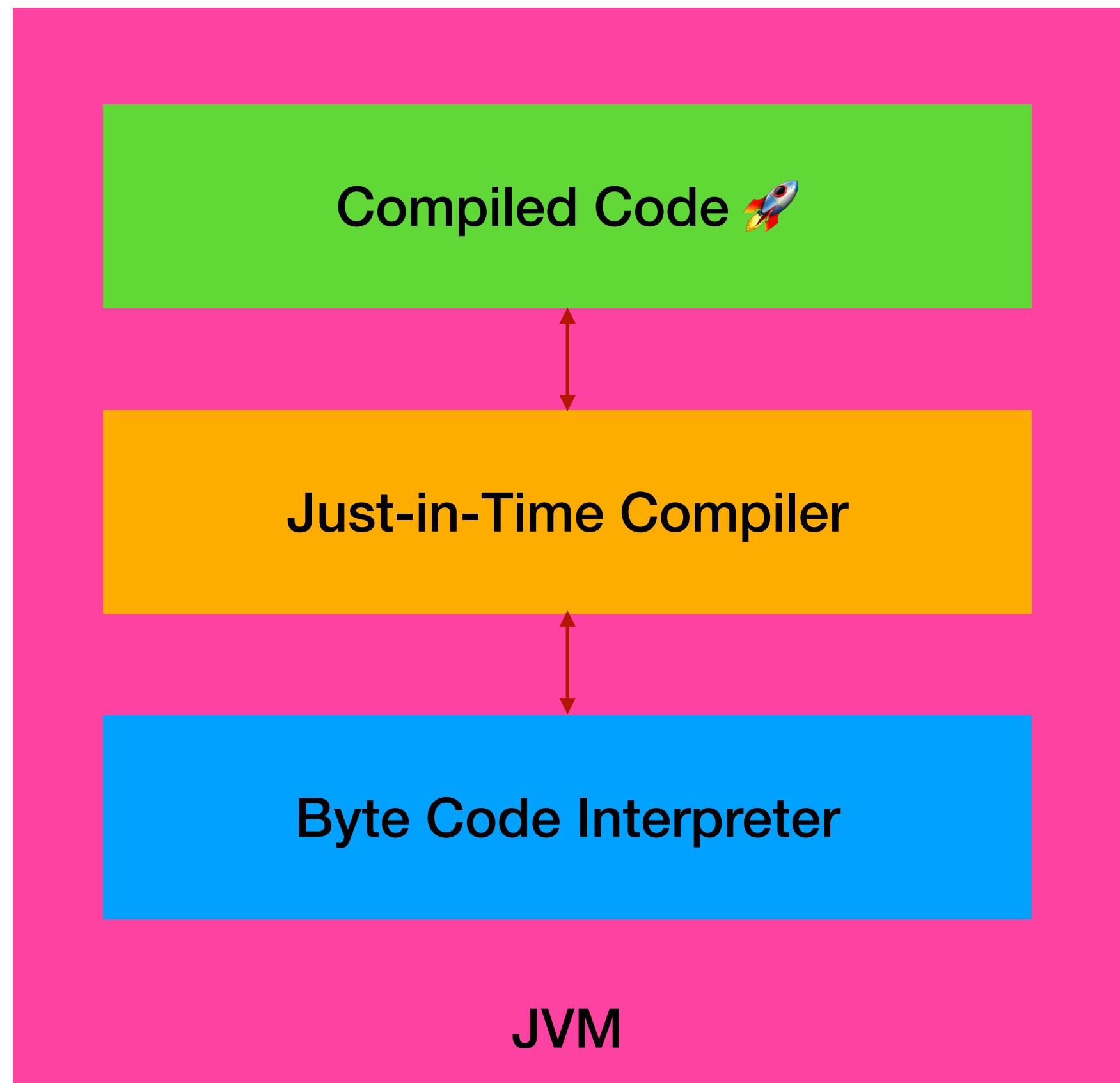


GraalVM™



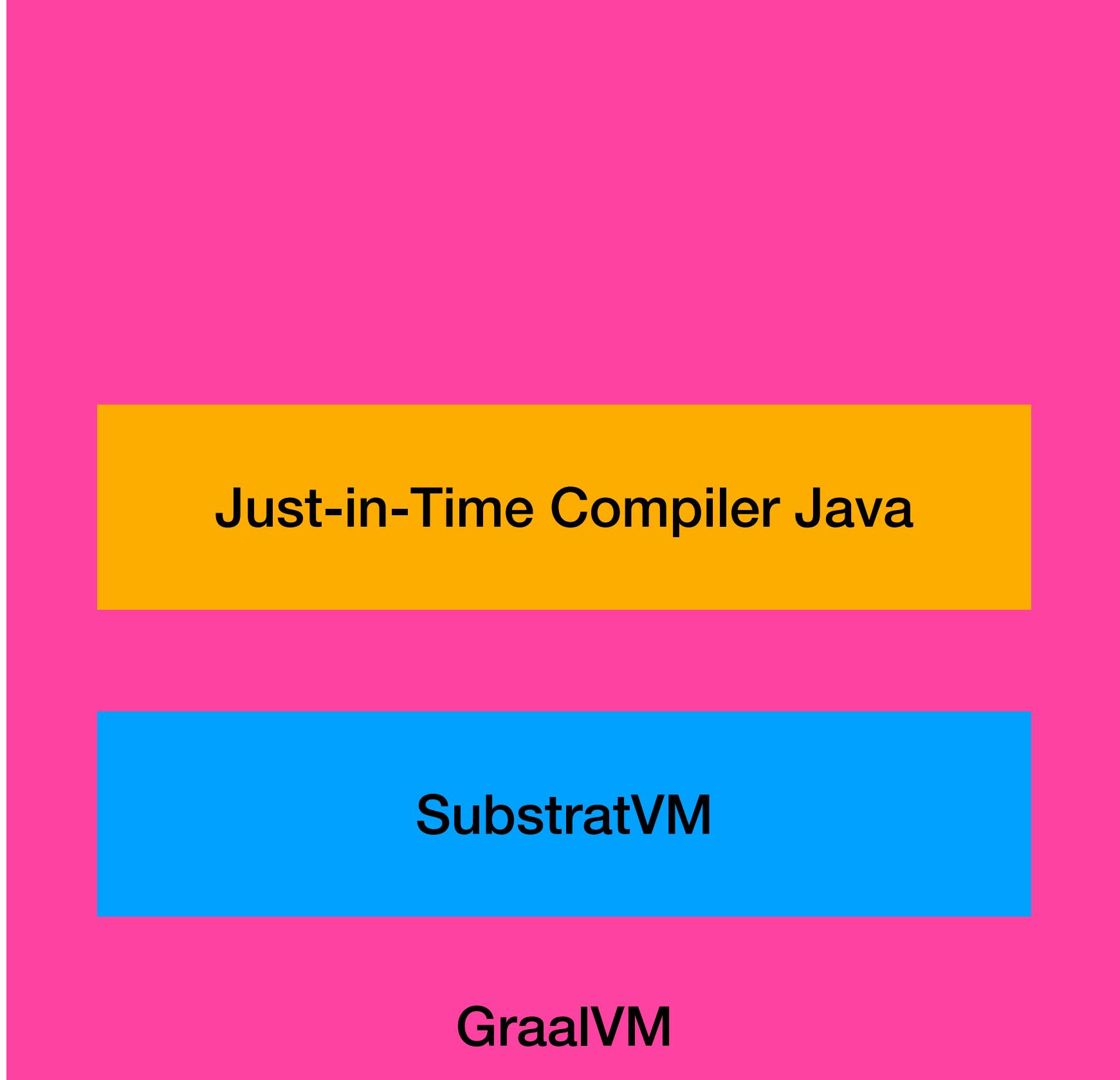
GraalVM





GraalVM





Just-in-Time Compiler Java

SubstratVM

GraalVM

GraalVM



Just-in-Time Compiler Python

GraalVM
Just-in-Time Compiler Java

Just-in-Time Compiler Ruby

SubstratVM

GraalVM

GraalVM



Just-in-Time Compiler Python

GraalVM
Just-in-Time Compiler Java

Just-in-Time Compiler Ruby

SubstratVM

GraalVM

GraalVM





GraalVM



AND THEN WE TOLD THEM



WRITE ONCE RUN ANYWHERE

memegenerator.net

GraalVM





gifbin.com

GraalVM





GraalVM



GraalVM
Just-in-Time Compiler Java



GraalVM





GraalVM

[Follow @graalvm](#)

19.5K followers



Oct 18 • 1 tweets • 1 min read

[Bookmark](#)

[Save as PDF](#)

[+ My Authors](#)

Excited about [@graalvm](#) JIT and Native Image becoming part of OpenJDK! #JavaOne

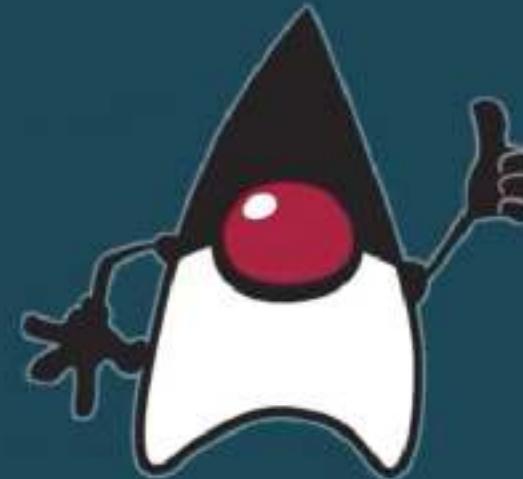
ANNOUNCEMENT

Oracle contributing
GraalVM CE Java code
to OpenJDK

Ongoing GraalVM CE Design and Development will move to OpenJDK Community

Going forward, GraalVM will use same development methodology and processes as used for Java

GraalVM will align with the Oracle Java release and licensing models



Additional details to follow.

GraalVM





GraalVM



Call for Discussion: New Project: Galahad

Douglas Simon doug.simon@oracle.com

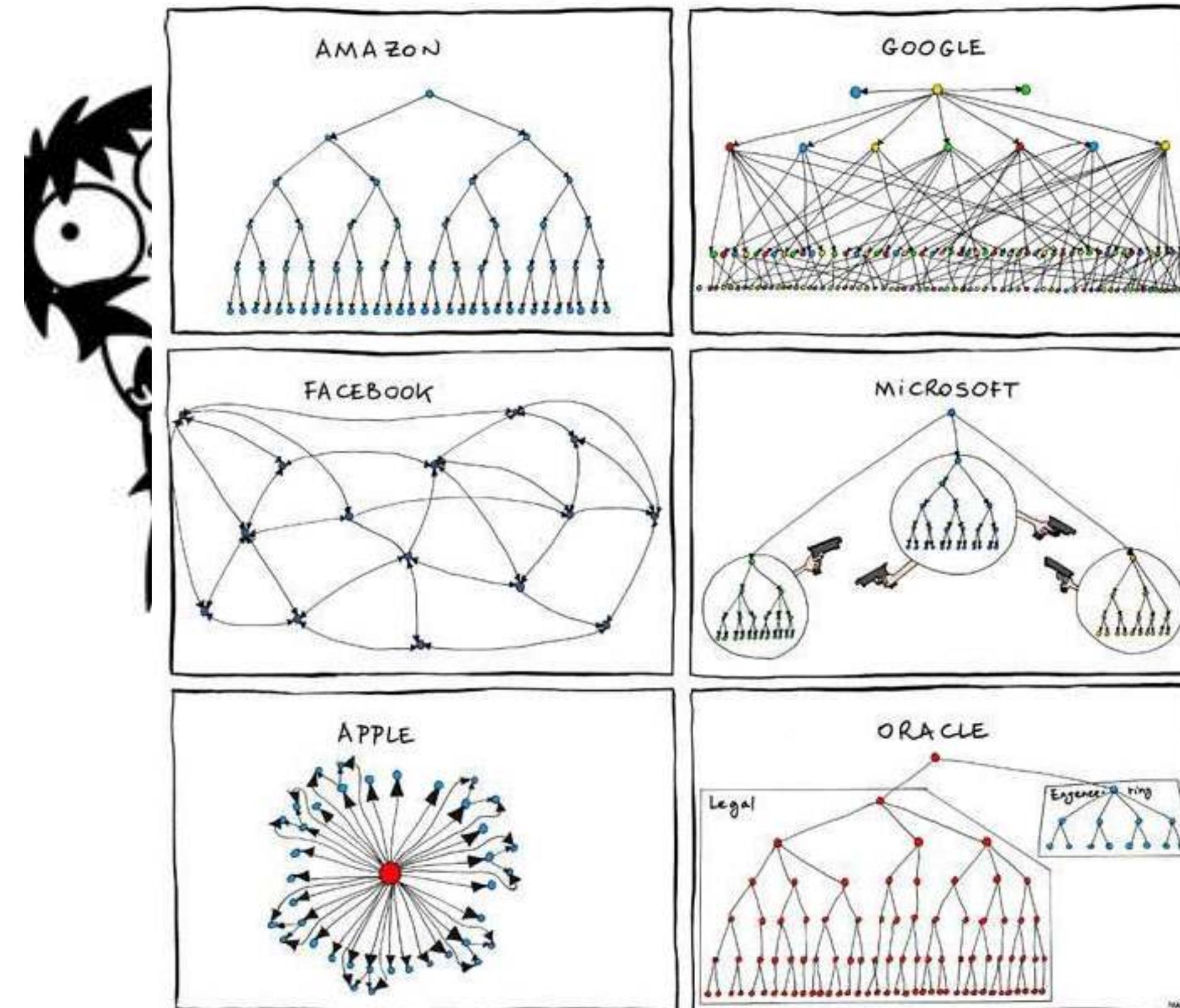
Tue Dec 13 09:24:12 UTC 2022

- Previous message (by thread): [Uses directive not visible in annotation processor when provider class not existing \(yet\)](#)
- Next message (by thread): [Call for Discussion: New Project: Galahad](#)
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

I hereby invite discussion of a new Project, Galahad*, whose primary goal is to contribute Java-related GraalVM technologies to the OpenJDK Community and prepare them for possible incubation in a JDK main-line release.

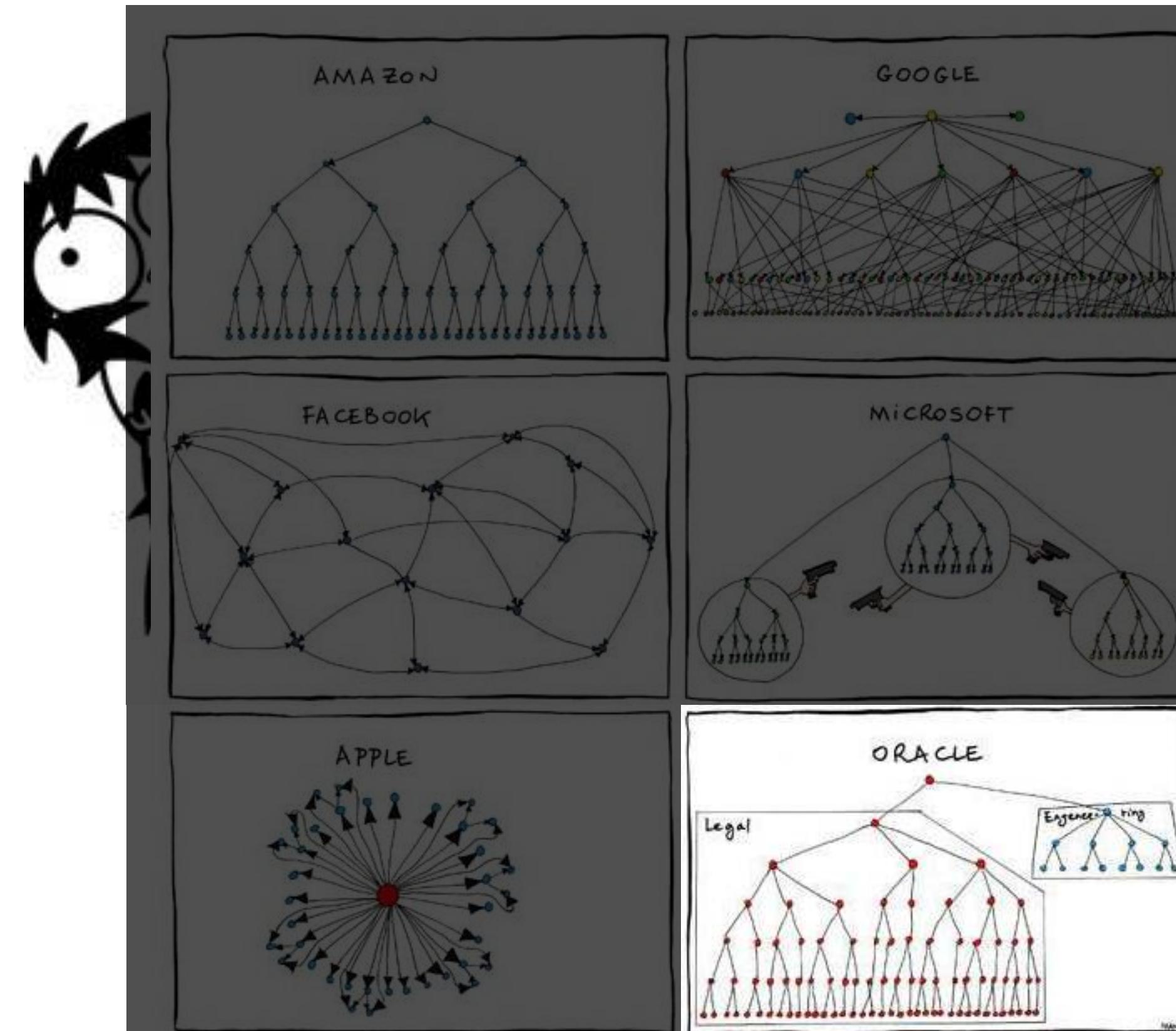
The initial focus will be on contributing the latest version of the GraalVM just-in-time (JIT) compiler and integrating it as an alternative to the existing JIT compiler of the HotSpot VM. Later steps will bring in the necessary ahead-of-time (AOT) compilation technology to make this new JIT compiler written in Java available instantly on JVM start and avoid any interference with application heap usage and execution profiling. We also intend to contribute portions of the Native Image technology as a general AOT compilation technology for Java applications. Galahad will pay close attention to Leyden and track the Leyden specification as it evolves.





Jakarta EE





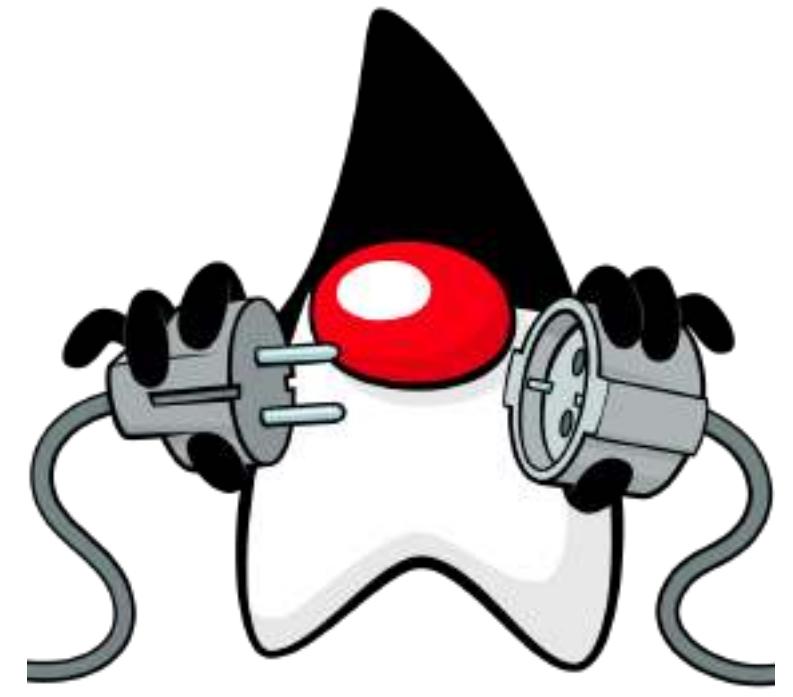
Jakarta EE





Jakarta EE

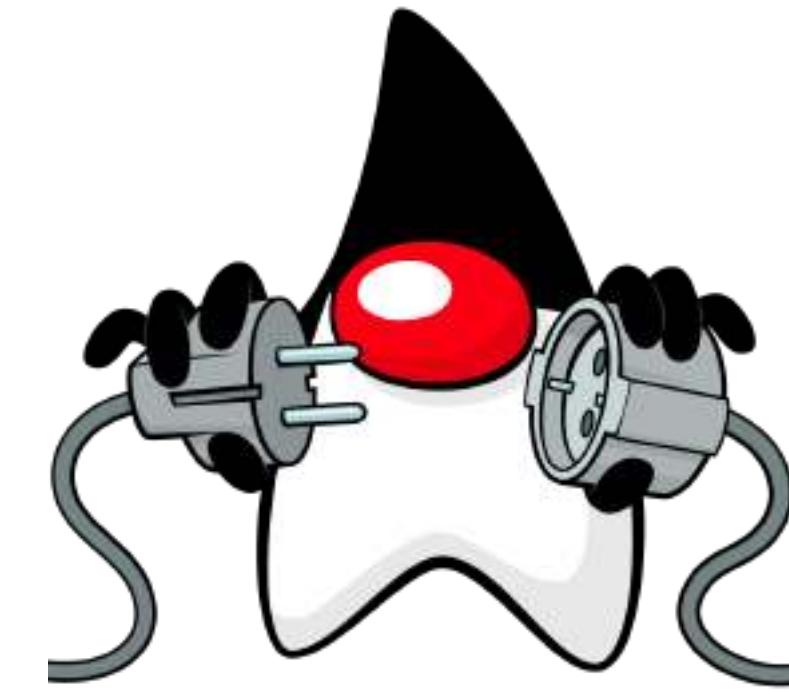




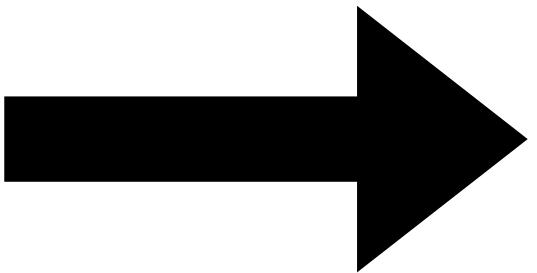
▪▪▪ AdoptOpenJDK

Jakarta EE





 AdoptOpenJDK



 ECLIPSE®
FOUNDATION

 ADOPTIUM

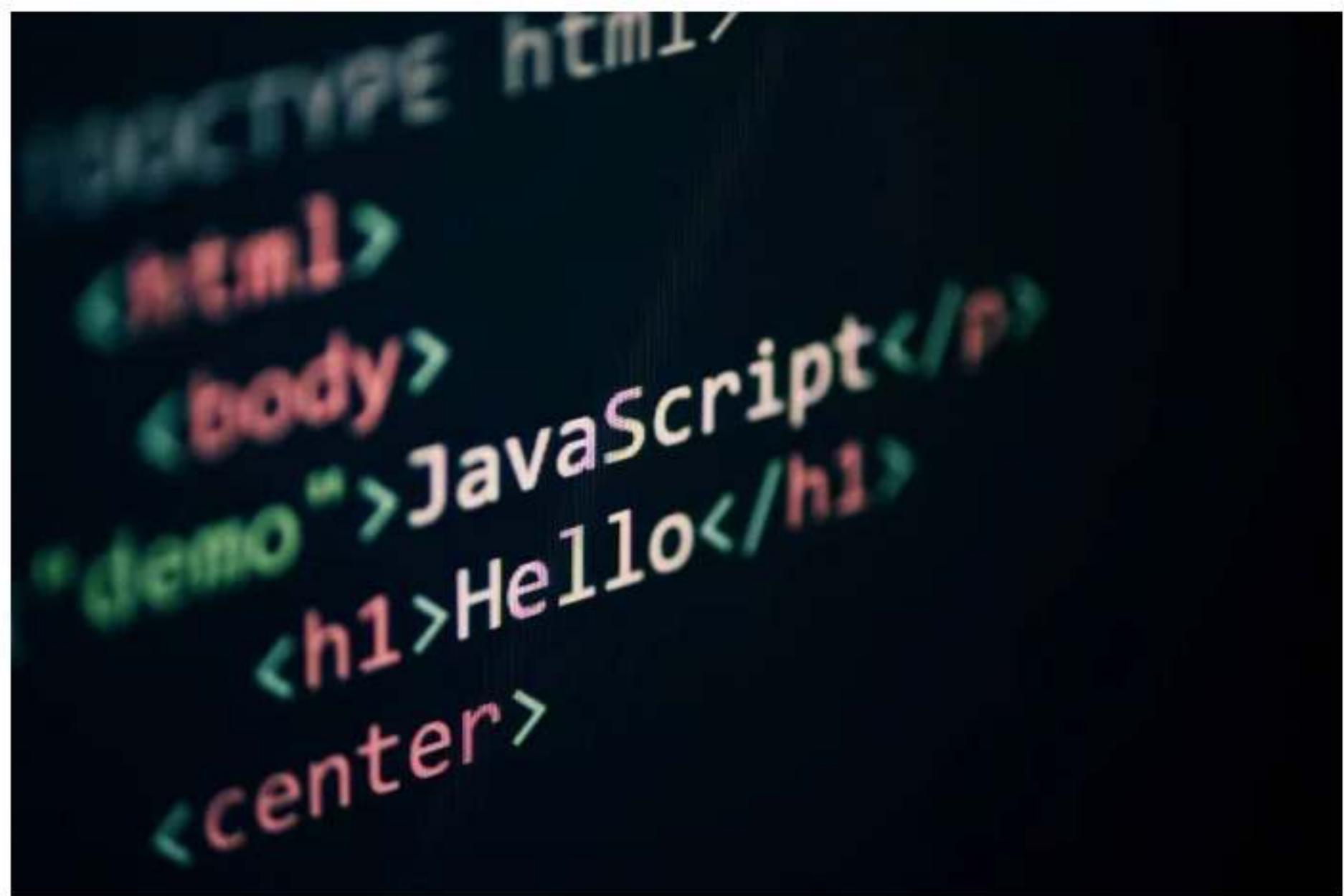
 TEMURIN

Jakarta EE



Node.js creator Ryan Dahl urges Oracle to release JavaScript trademark

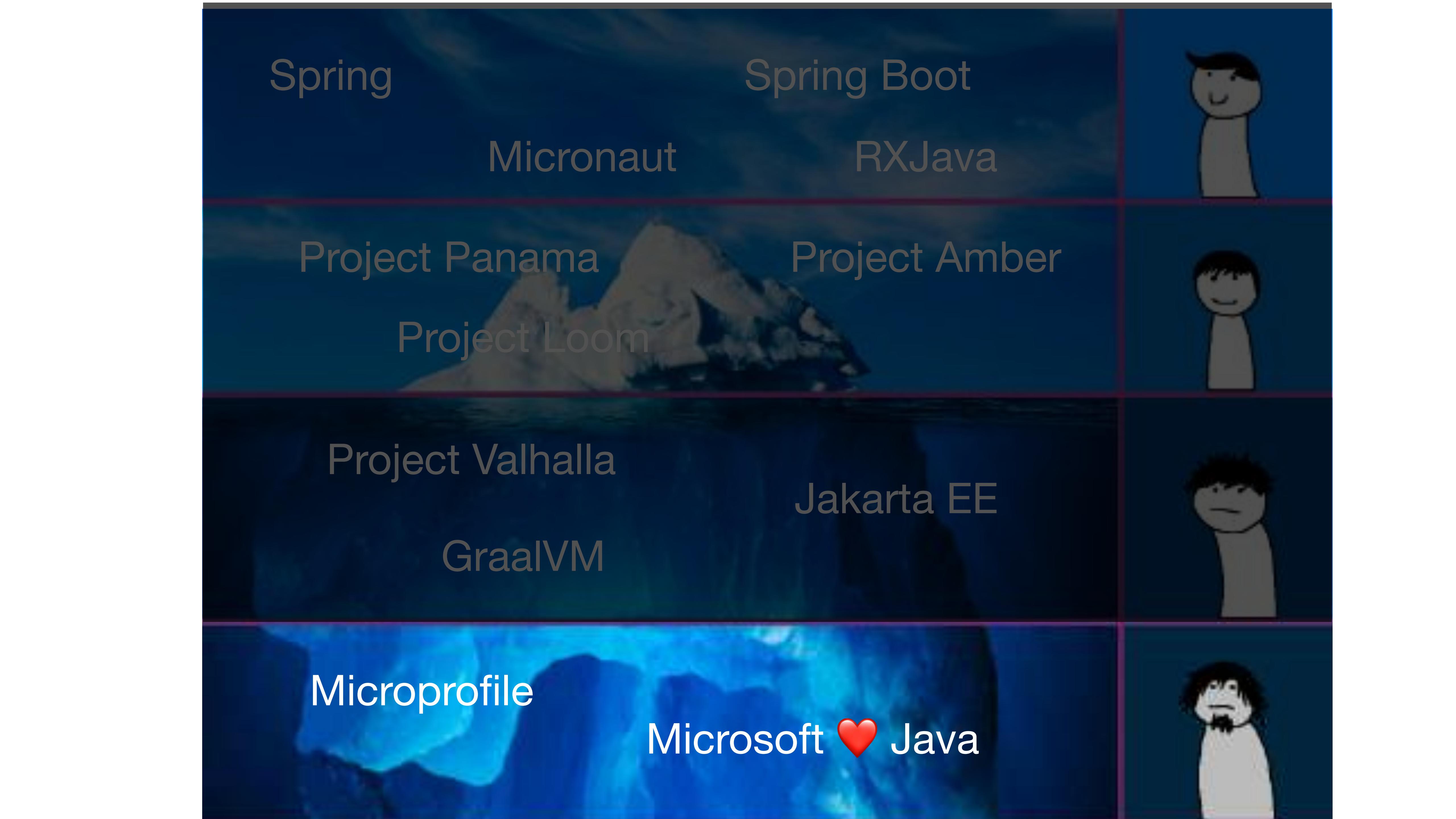
By [Tim Anderson](#) - September 5, 2022



The creator of Node.js and Deno, Ryan Dahl, has penned an open letter to Oracle imploring the company to release the JavaScript trademark into the public domain.

Jakarta EE





Spring

Spring Boot

Micronaut

RXJava

Project Panama

Project Amber

Project Loom

Project Valhalla

Jakarta EE

GraalVM

Microprofile

Microsoft ❤️ Java





Java EE™ (1999)

Microprofile





Full Profile



Microprofile





Full Profile



Web Profile (2009)



Microprofile





Full Profile

Microprofile

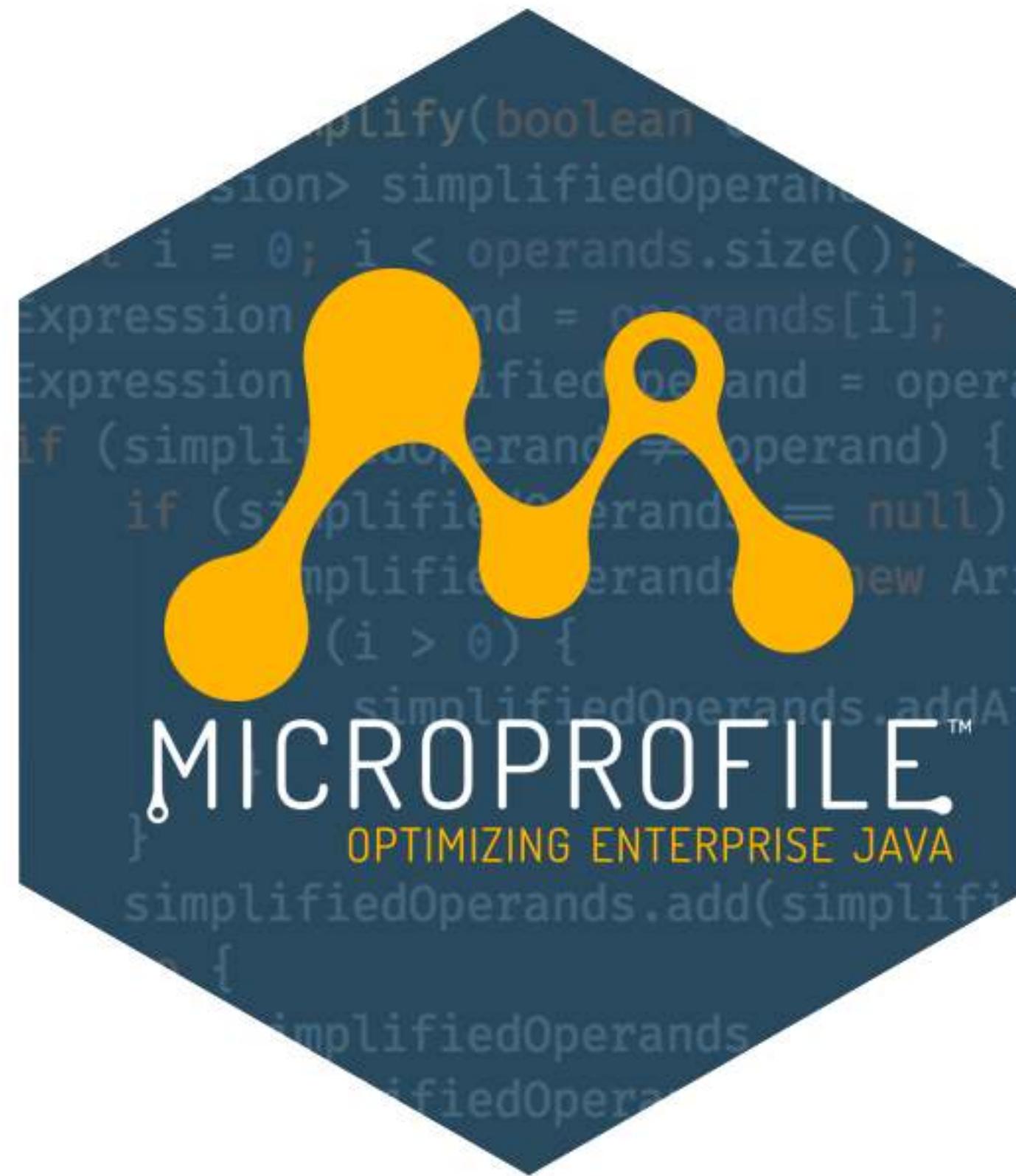




Web Profile

Microprofile





Microprofile

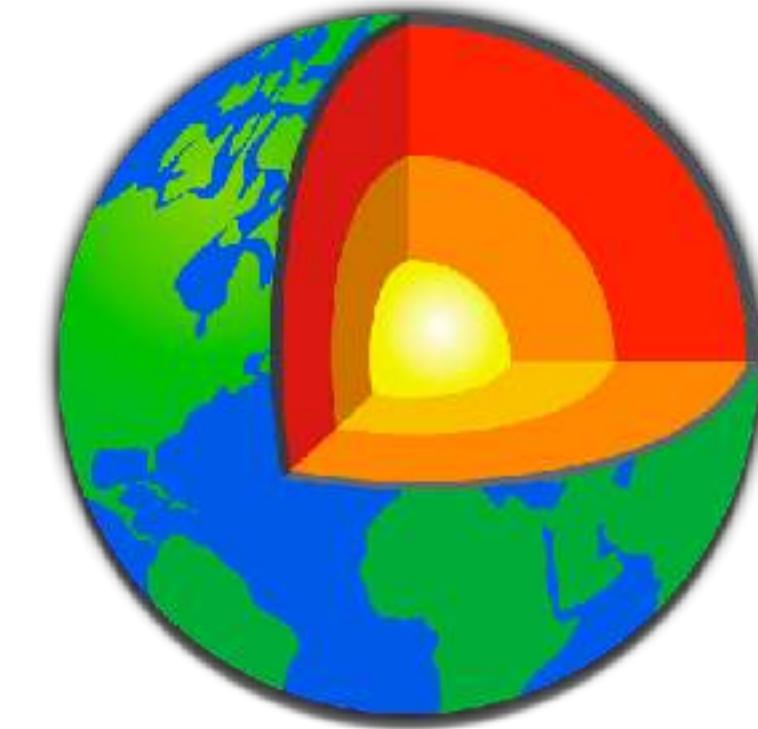




Full Profile



Web Profile



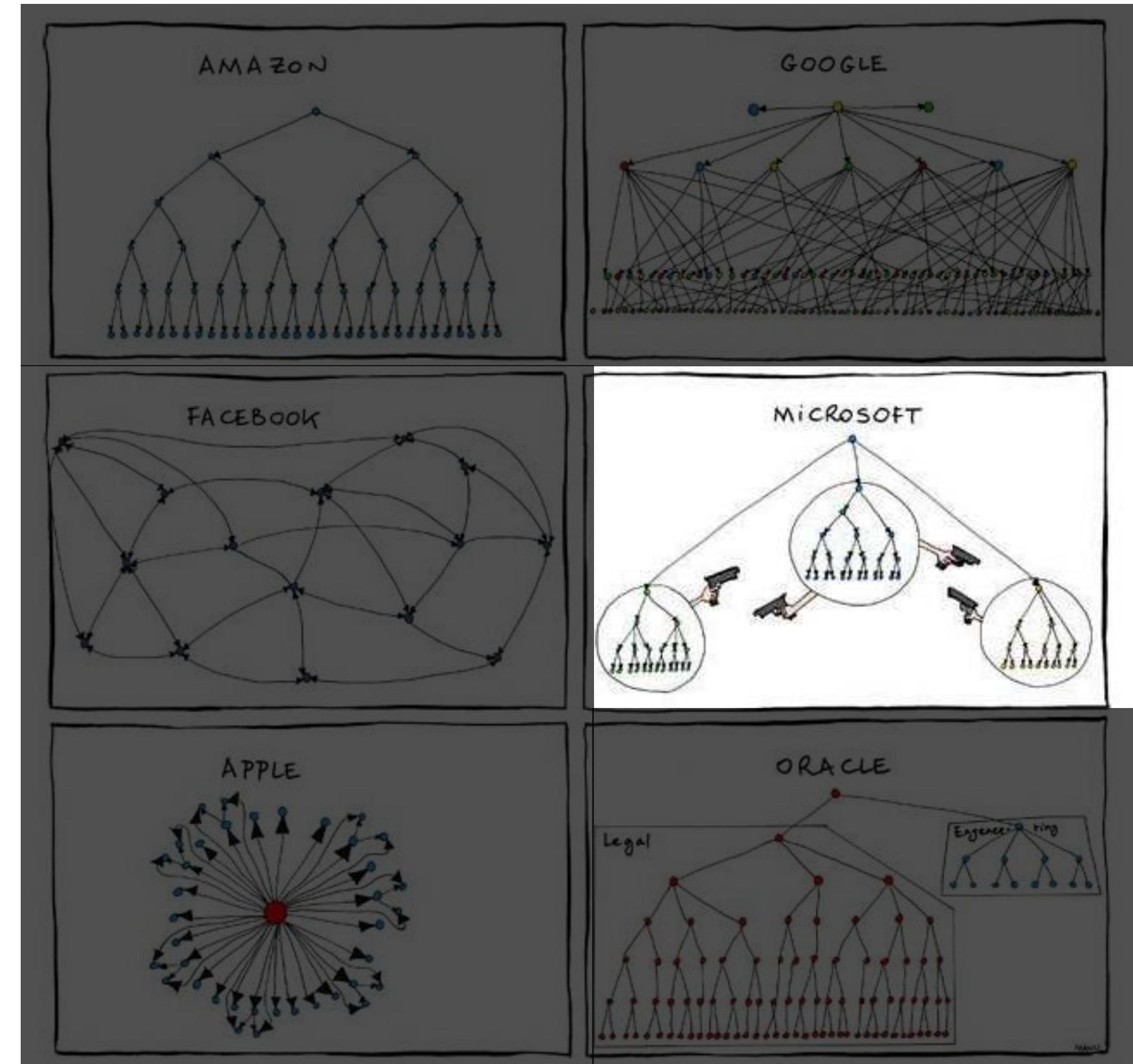
Core Profile



JAKARTA® EE

Microprofile





Microsoft ❤️ Java





OUR MEMBERS

Home / Membership / Members

Strategic Member



Enterprise Member



INFOIS 中创中间件®

Adoptium® Working Group Members

Strategic Members

Strategic Members are organizations that view Adoptium working group managed technology as critical to their organization's future, and are investing significant resources to sustain and define the core activities that are the responsibility of the working group.



Projects

Working Group

Implementations

Resources

Blog

Contributors

Join The Discussion

MicroProfile Starter

See The Code

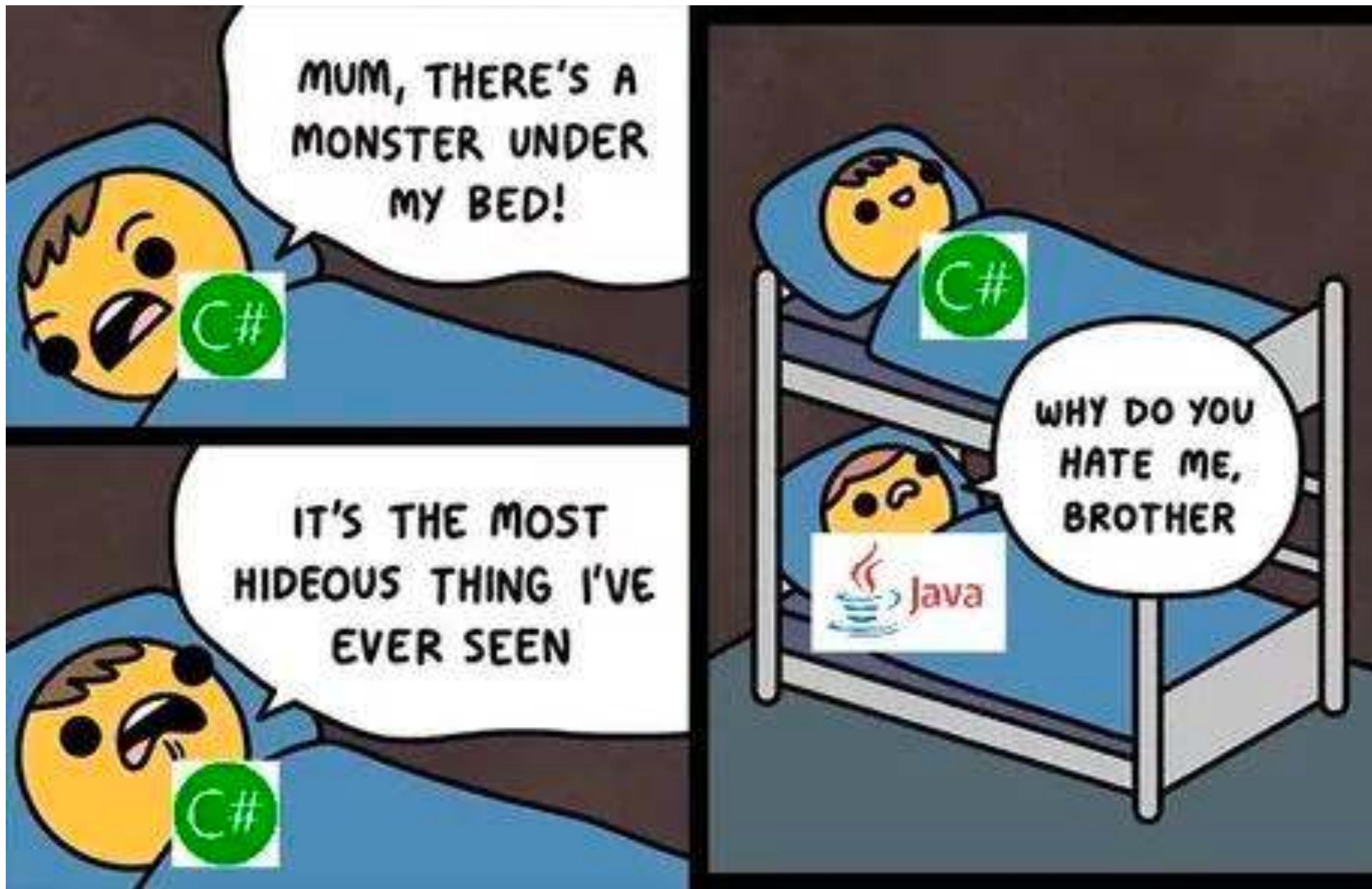


ORACLE®



Microsoft ❤️ Java





Microsoft ❤ Java



16-ea+10-macOS-aarch64

Pre-release

Compare ▾



lewurm released this 12 Nov 2020

· 31 commits to master since this release

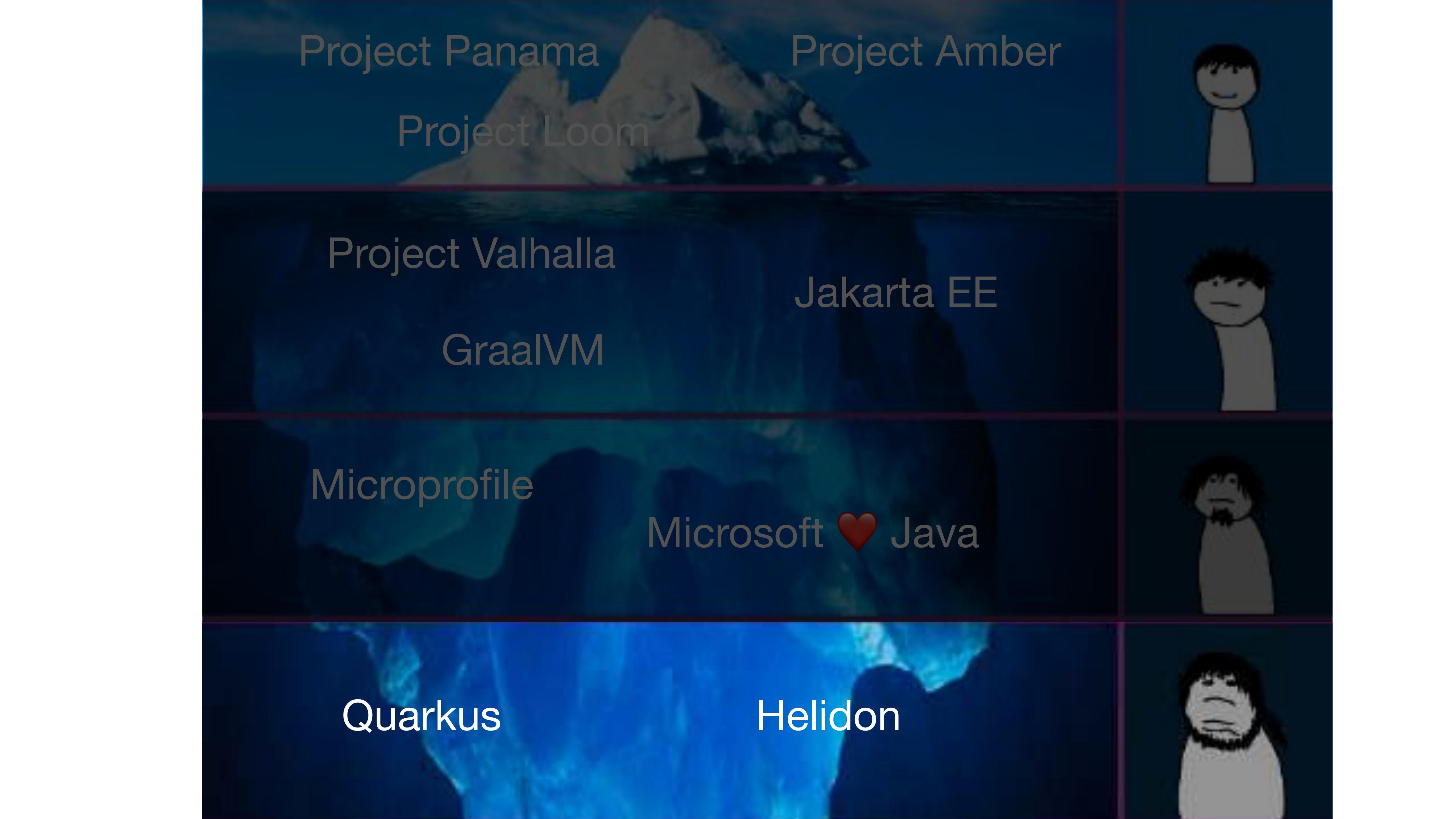
↳ 16-ea+10-ma...

-o- 2f8fc04 ✓

This release is an early access build of our macOS+AArch64 port (also known as Apple Silicon), based on [openjdk/jdk@ 68da63d](#)

Together with Azul Systems we are working on [JEP 391](#) to upstream relevant patches to the OpenJDK project. The WIP branch can be found here:
<https://github.com/openjdk/aarch64-port/tree/jdk-macos>





Project Panama

Project Loom

Project Valhalla

GraalVM

Microprofile

Microsoft ❤️ Java

Quarkus

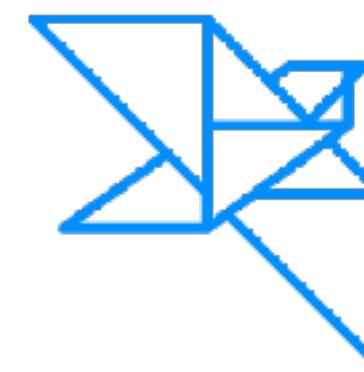
Helidon

Project Amber





QUARKUS



helidon.io



ORACLE

Quarkus/Helidon





GraalVM™

Quarkus/Helidon





Quarkus/Helidon



Project Valhalla

Jakarta EE

GraalVM

Microprofile

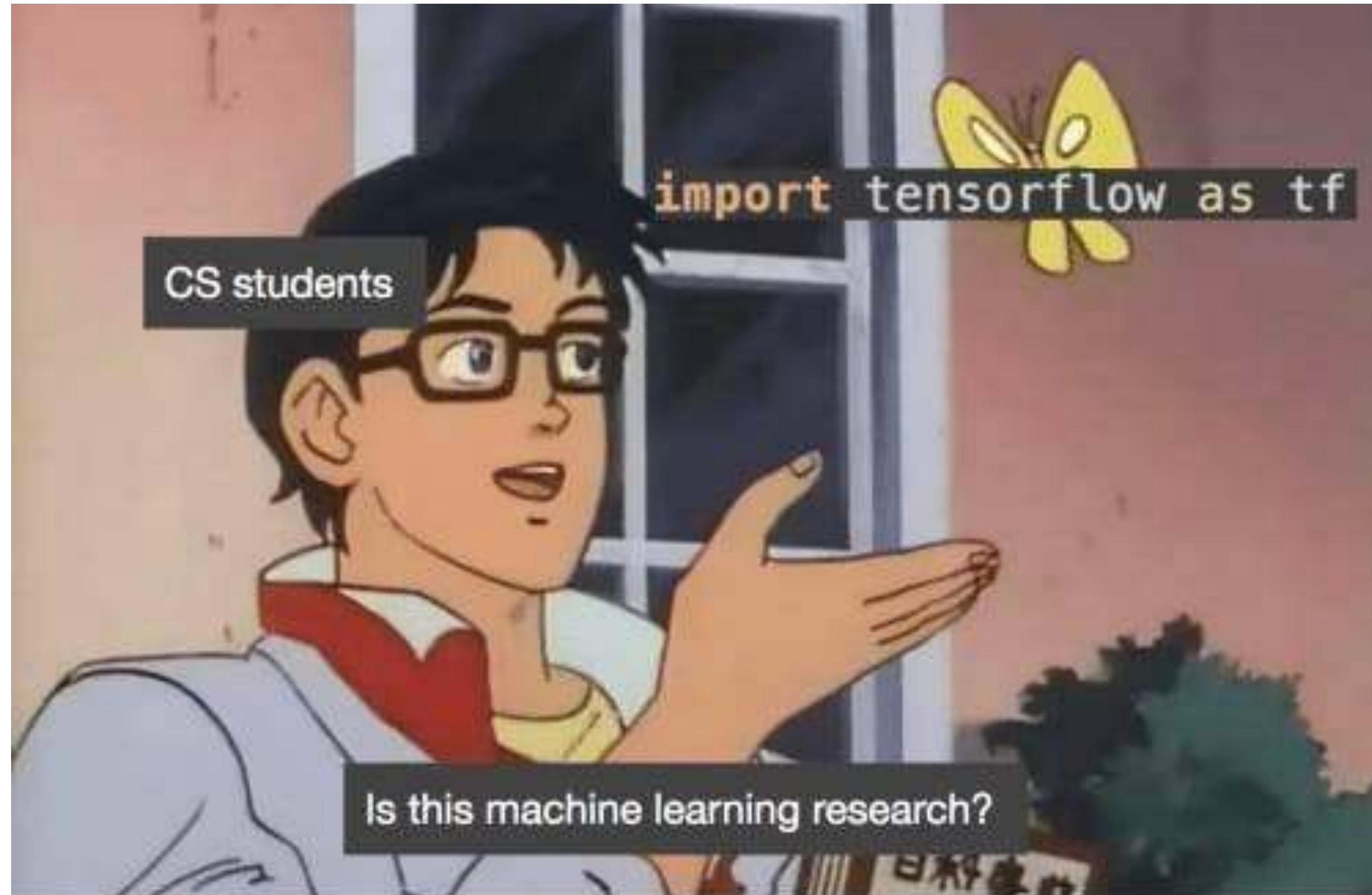
Microsoft ❤️ Java

Quarkus

Helidon

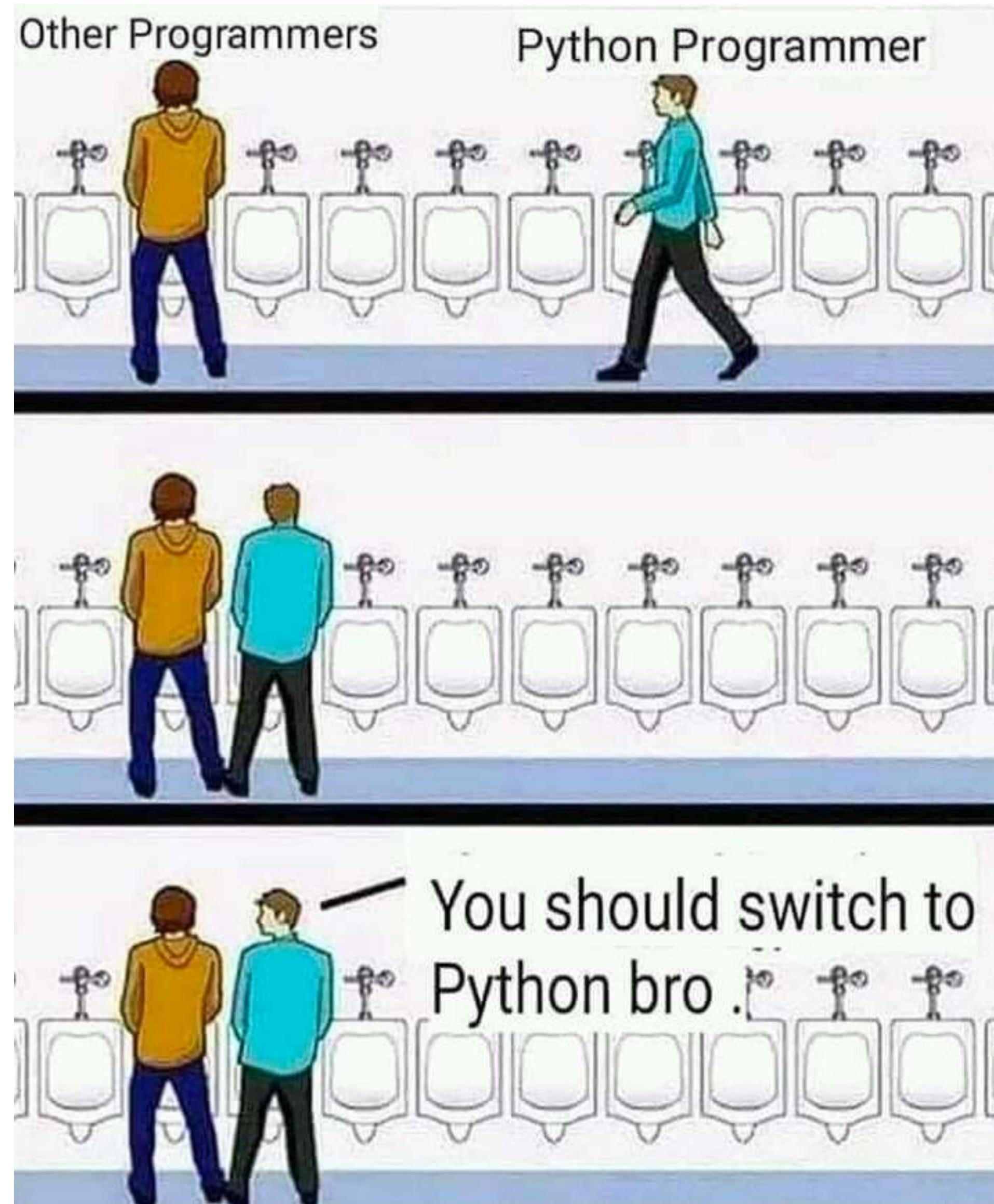
JSR381 Visual Recognition API

Project Lanai

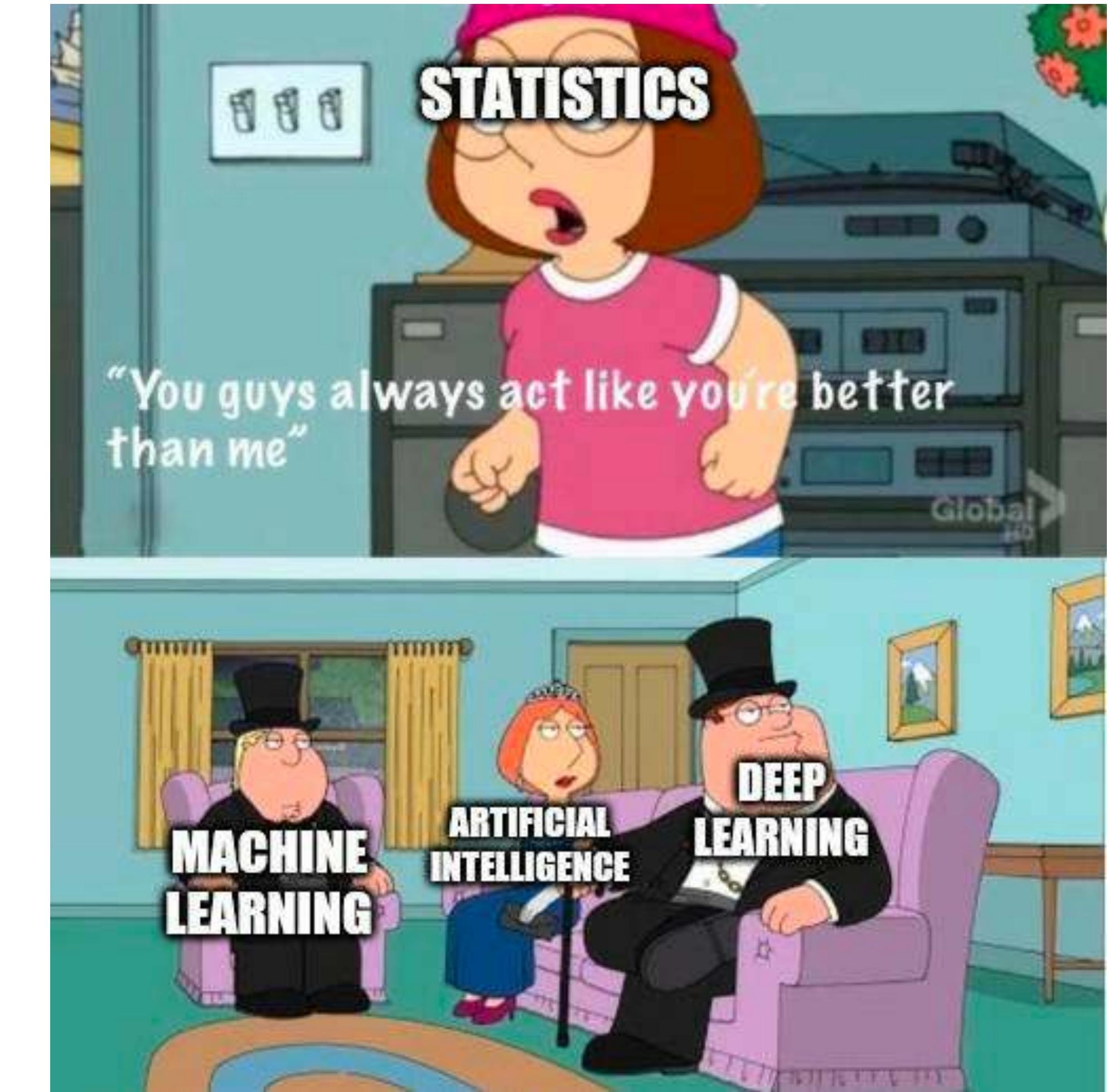
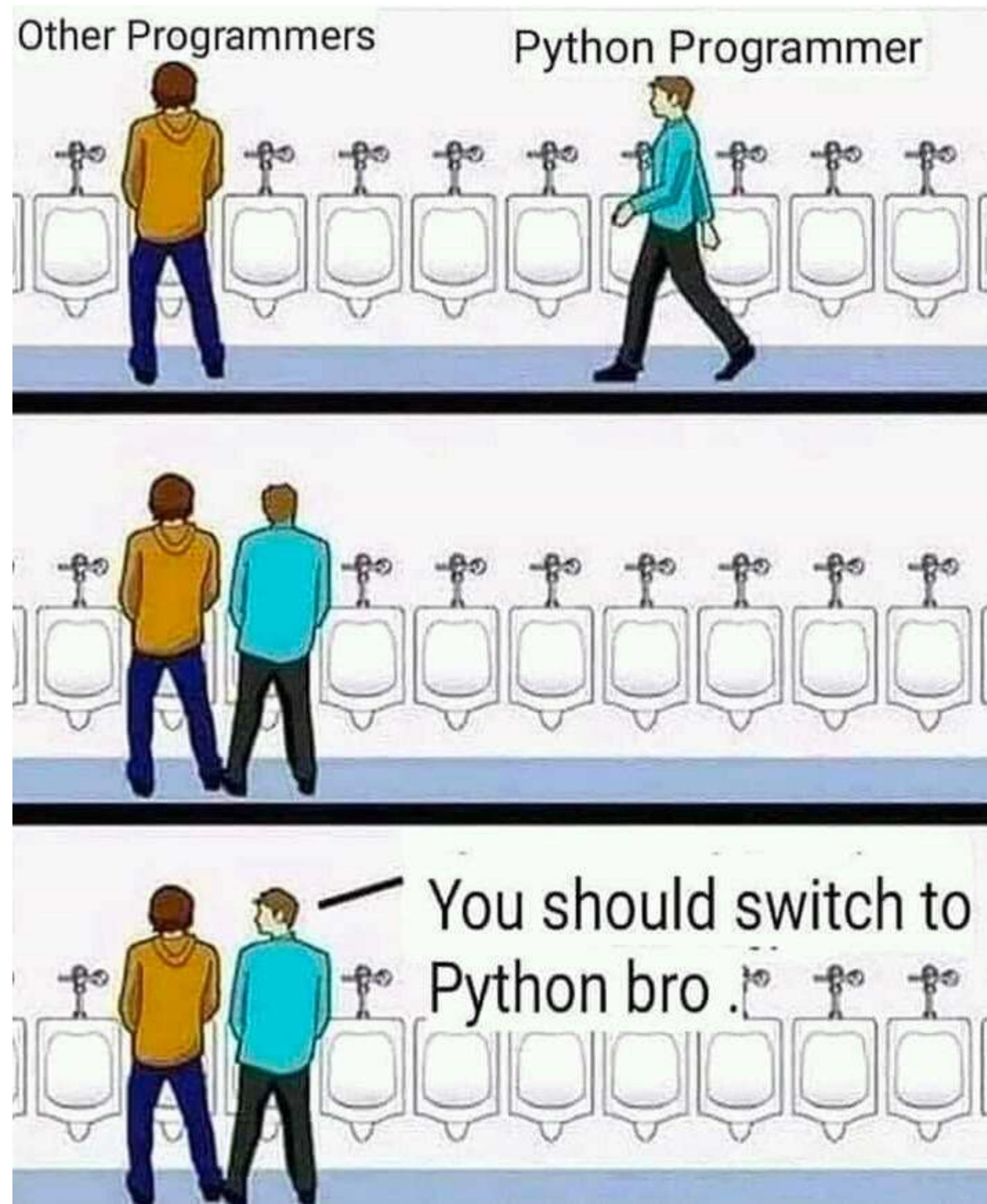


Programmers Nowadays

JSR381 Visual Recognition API



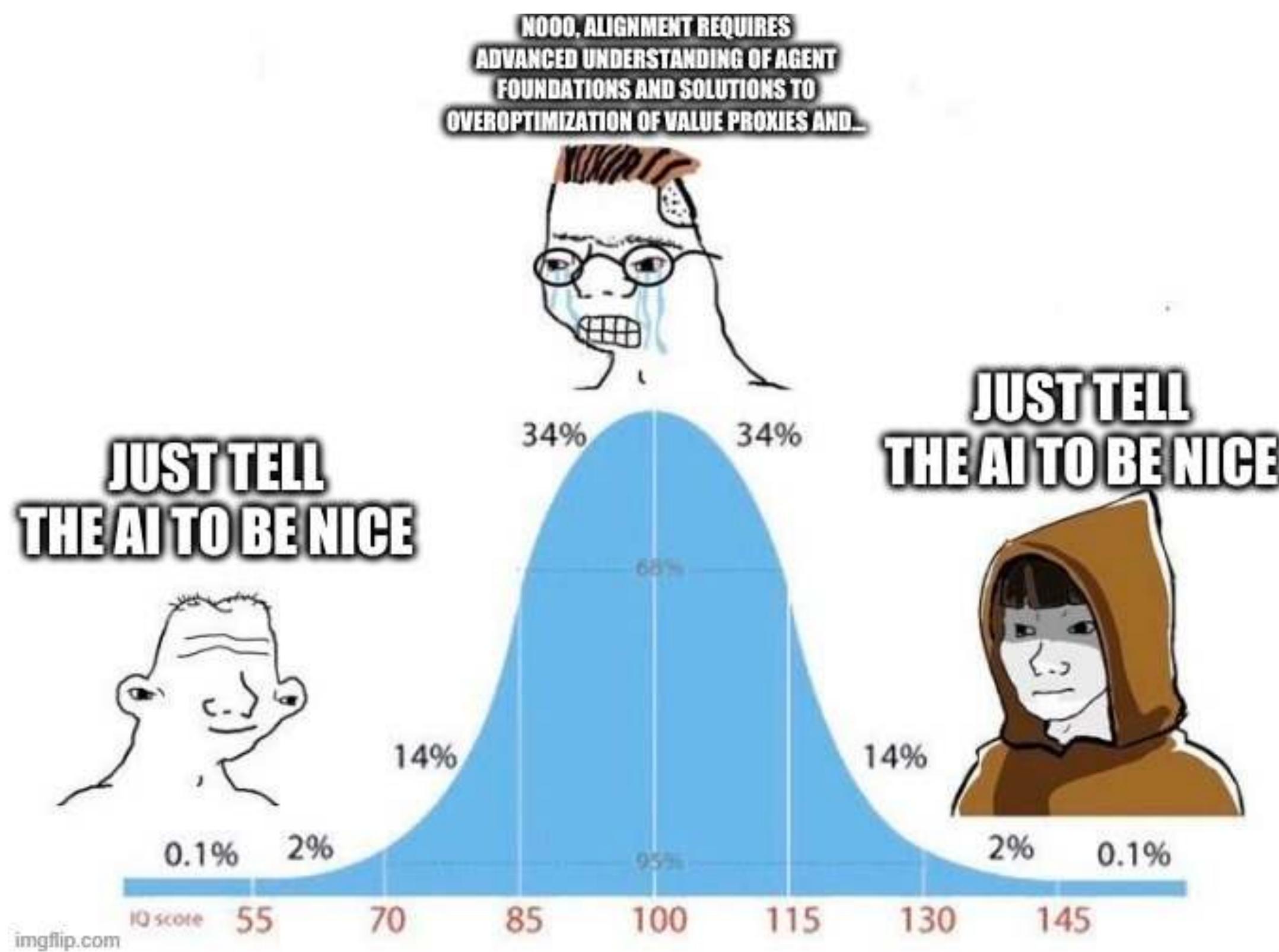
JSR381 Visual Recognition API



JSR381 Visual Recognition API

JSR381 "VisRec" to standardowy interfejs
API dla podstawowego **uczenia
maszynowego (ML)**, **klasyfikacji
obrazów** i **rozpoznawania obiektów.**

JSR381 Visual Recognition API



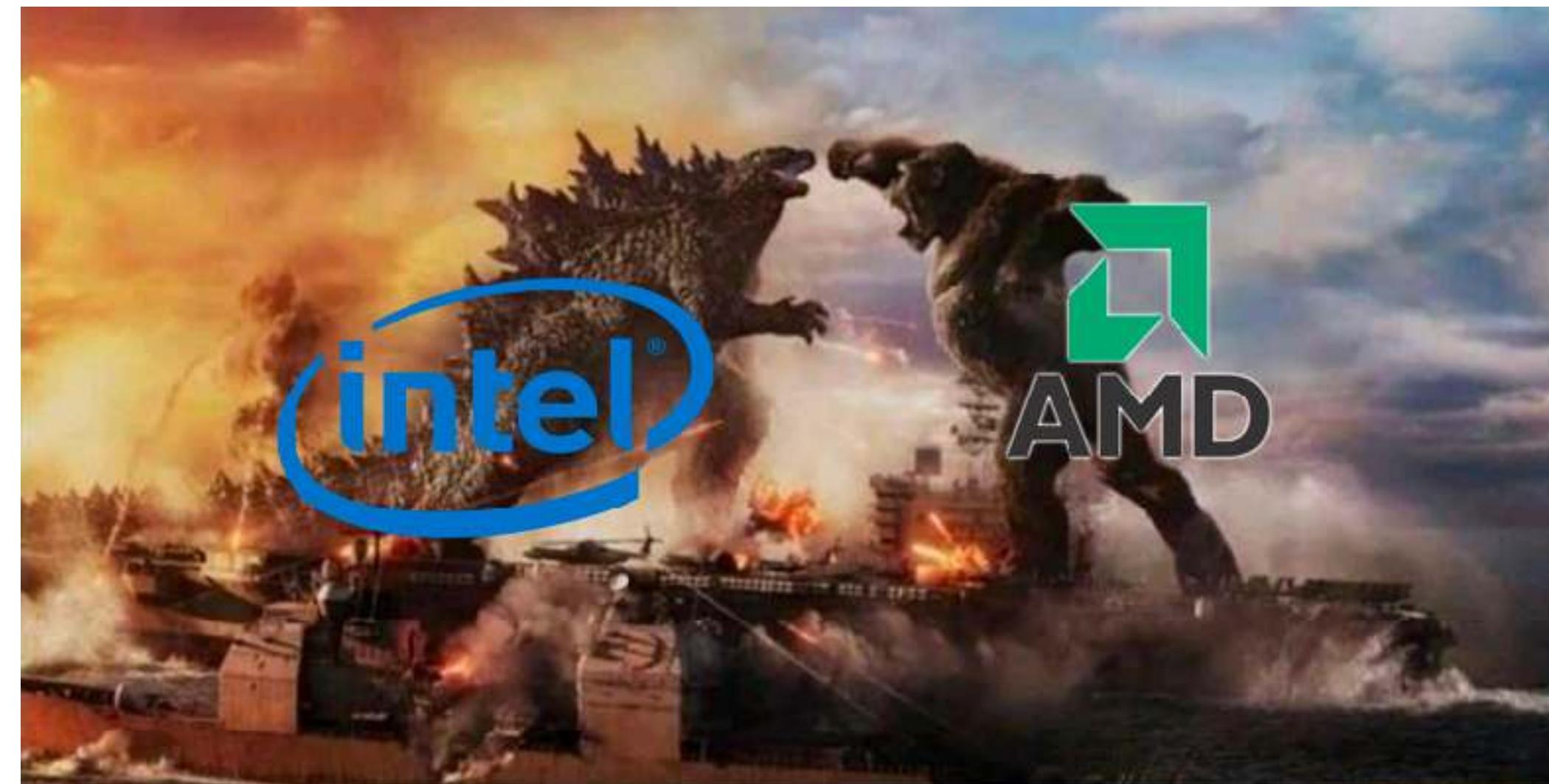
JSR381 Visual Recognition API

**CHCIELIBYŚCIE ABY KTOŚ
KIEDYŚ NA WAS TAK PATRZYŁ**

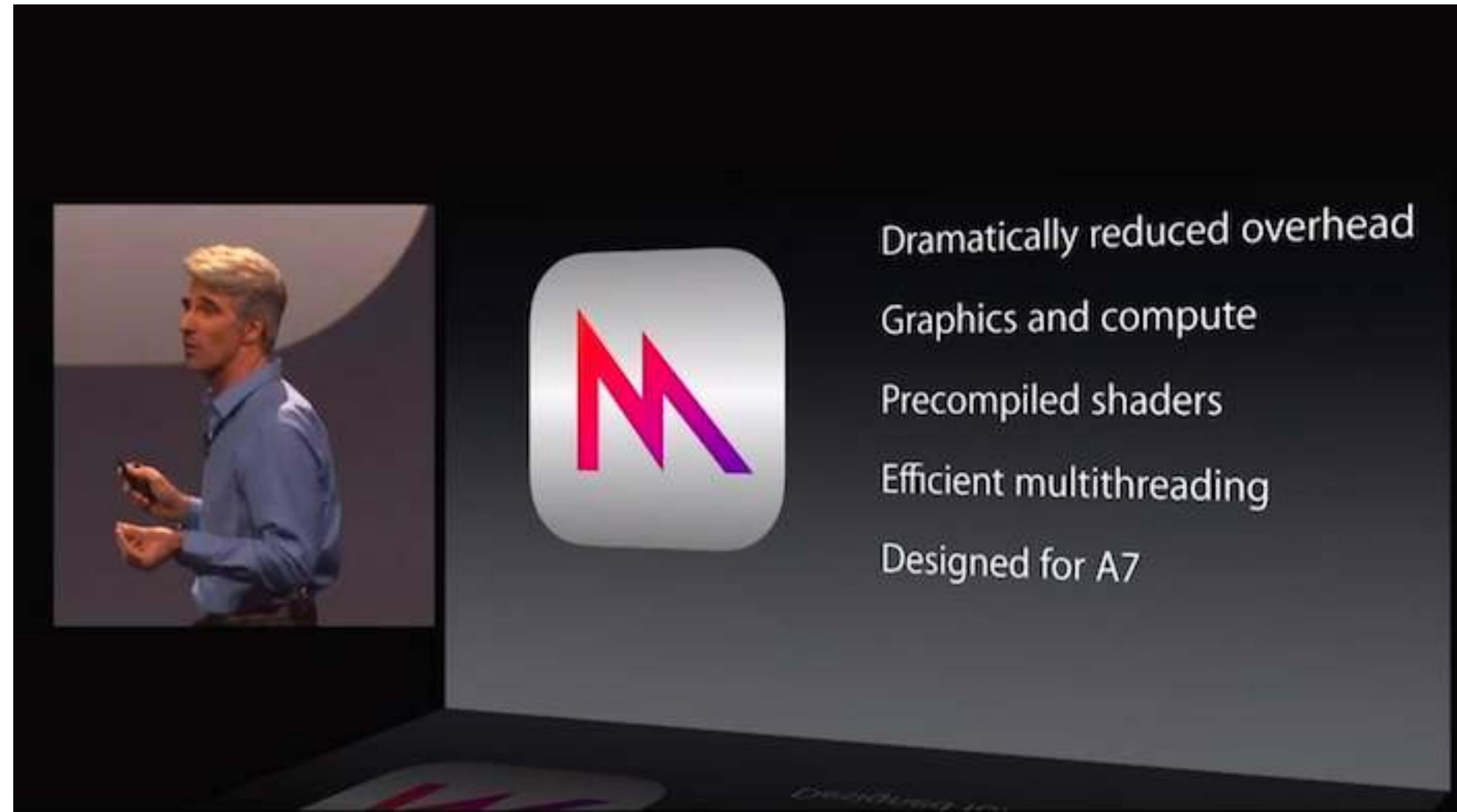
JAK CRAIG FEDERIGHI NA NOWEGO MACBOOKA

imgflip.com

Project Lanai



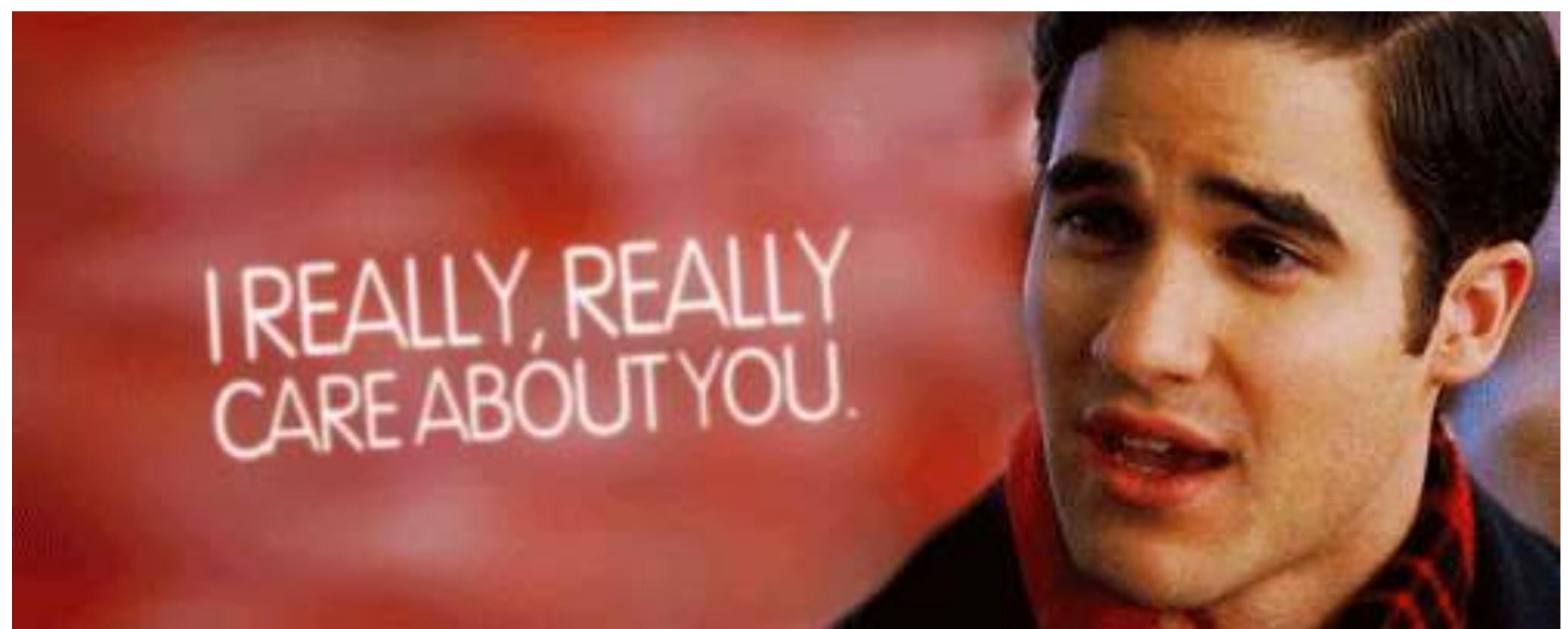
Project Lanai



Project Lanai



Project Lanai



Project Lanai

Starcraft II Editor on Apple Macbook M1

StarCraft II Technical Support

Вадим 1 post

Jul 13

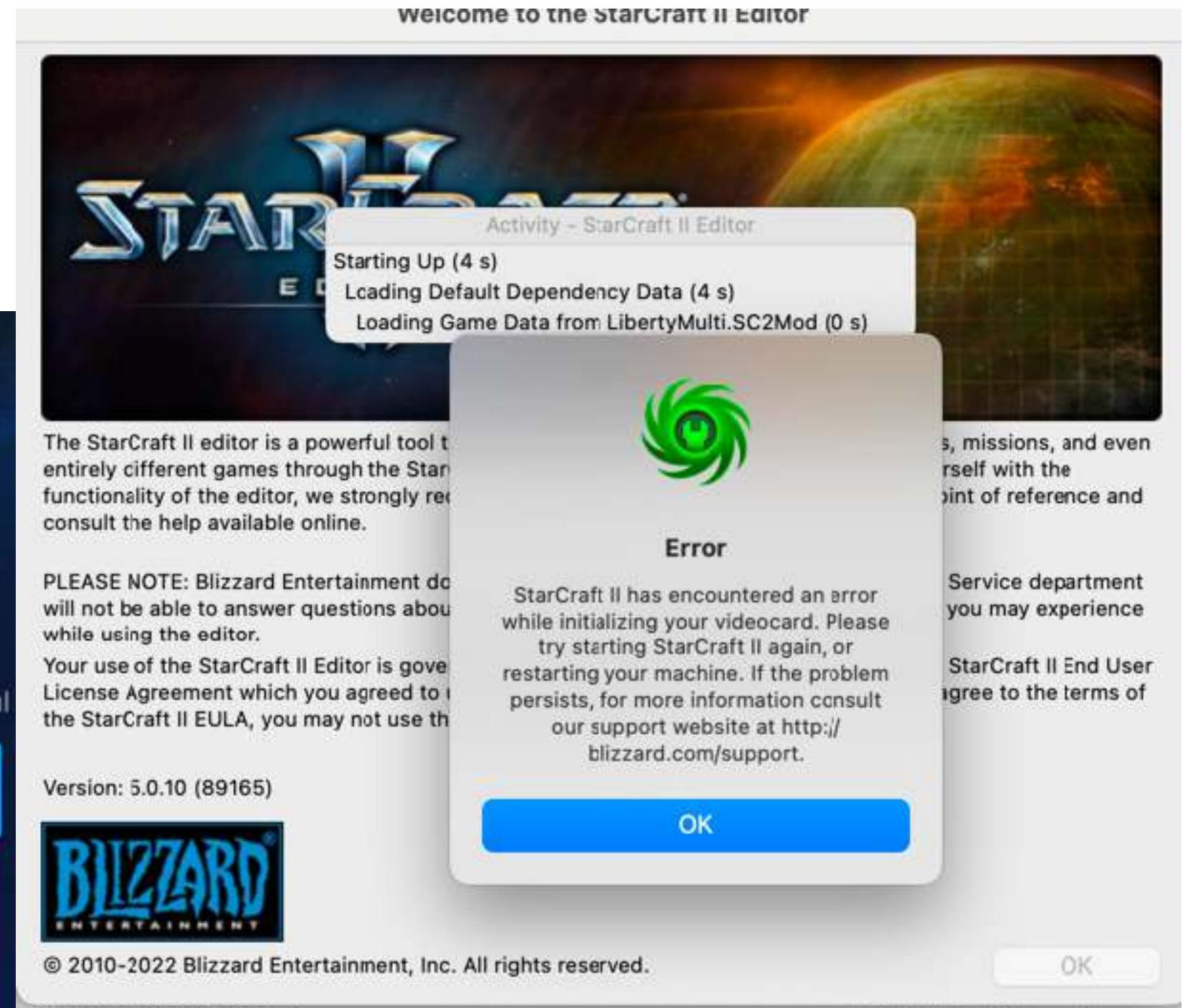
Starcraft II on Apple Macbook Air M1 work perfectly, but! I can't run the Starcraft II Editor to play third-party maps, I see an error " StarCraft II has encountered an error while initializing your videocard. Please try starting StarCraft II again, or restarting your machine." Why? Some maps I can play from Starcraft itself, but downloaded from the Internet must be run through the Editor, which does not work. What do I do?

Jul 13

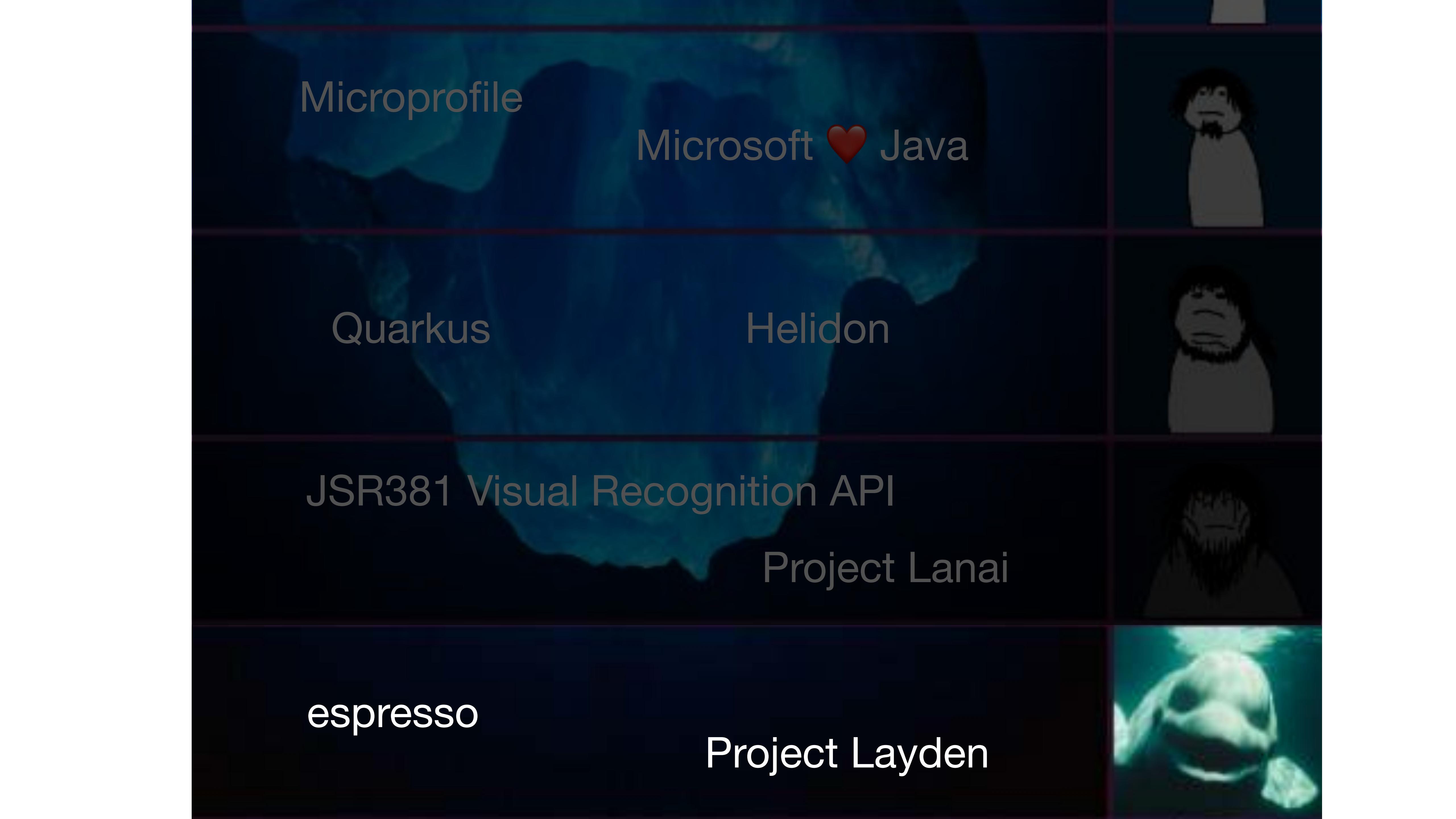
Jul 13

Aug 1

created last reply 5 replies 306 views 3 users 1 link



Project Lanai



Microprofile

Microsoft ❤️ Java

Quarkus

Helidon

JSR381 Visual Recognition API

Project Lanai

espresso

Project Layden

Celem **projektu Leyden** jest poprawa czasu uruchamiania, czasu do osiągnięcia szczytowej wydajności programów napisanych Java.

Project Leyden



Selectively Shifting and Constraining Computation

Mark Reinhold

2022/10/13

The goal of Project Leyden is to improve the startup time, time to peak performance, and footprint of Java programs. In this note we propose to work toward that goal by extending the Java programming model with features for *selectively shifting and constraining computation* by means of *condensing code*. We also propose an initial research and development roadmap.

We can often improve a program's startup time, warmup time, and footprint by *shifting* some of its computation temporally, either forward to a point later in run time (*e.g.*, via lazy initialization) or backward to a point earlier than run time (*e.g.*, via ahead-of-time compilation). We can further improve performance by *constraining* some of the computation related to Java's dynamic features (*e.g.*, class loading, class redefinition, and reflection), which enables better code analysis and thus even more optimization.



NASZA APLIKACJA

Skompilowana
Ahead-of-Time

Kubernetes

Project Leyden

Kompilacja Ahead-of-Time

Compile time

Application Class Data Sharing

Class Data Sharing Generation Time

?

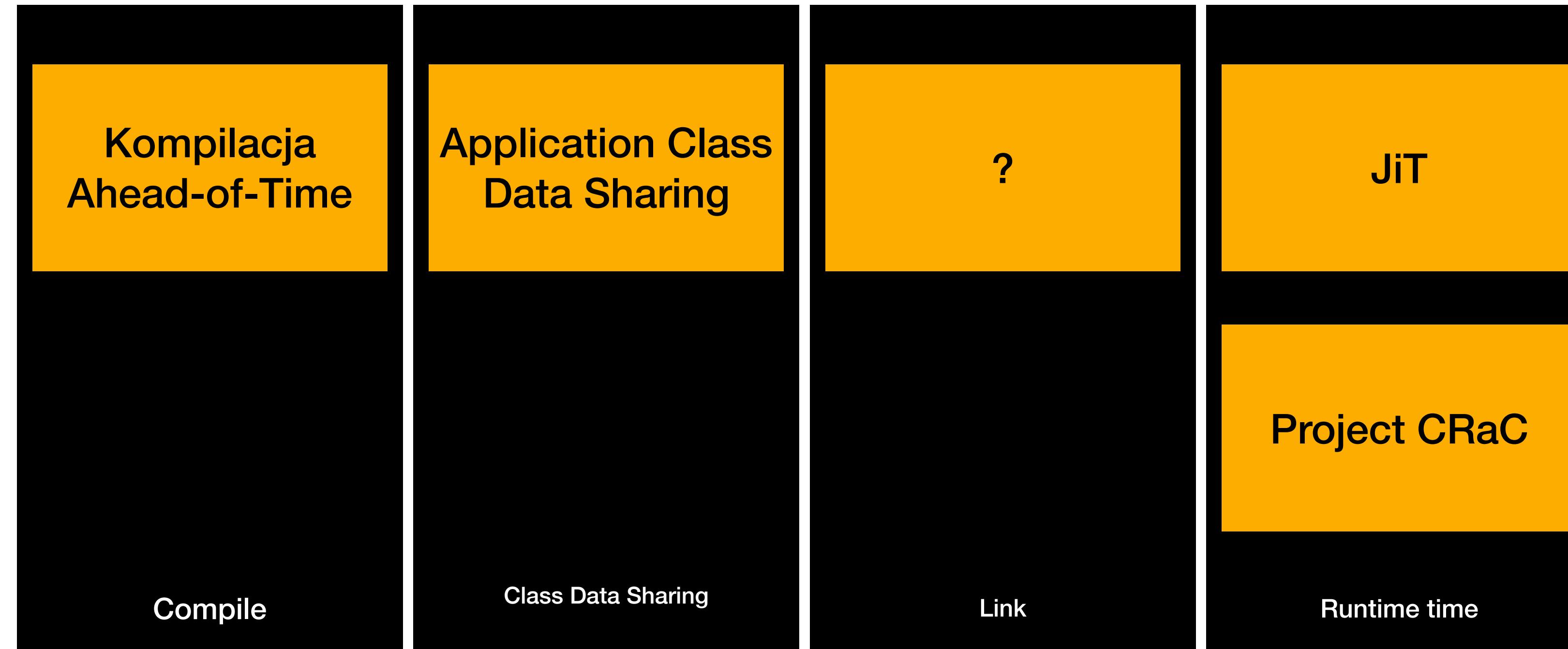
Link

JiT

Project CRaC

Runtime time

Project Leyden



Project Leyden

Kod

Kompilacja
Ahead-of-Time

Compile

Application Class
Data Sharing

Class Data Sharing

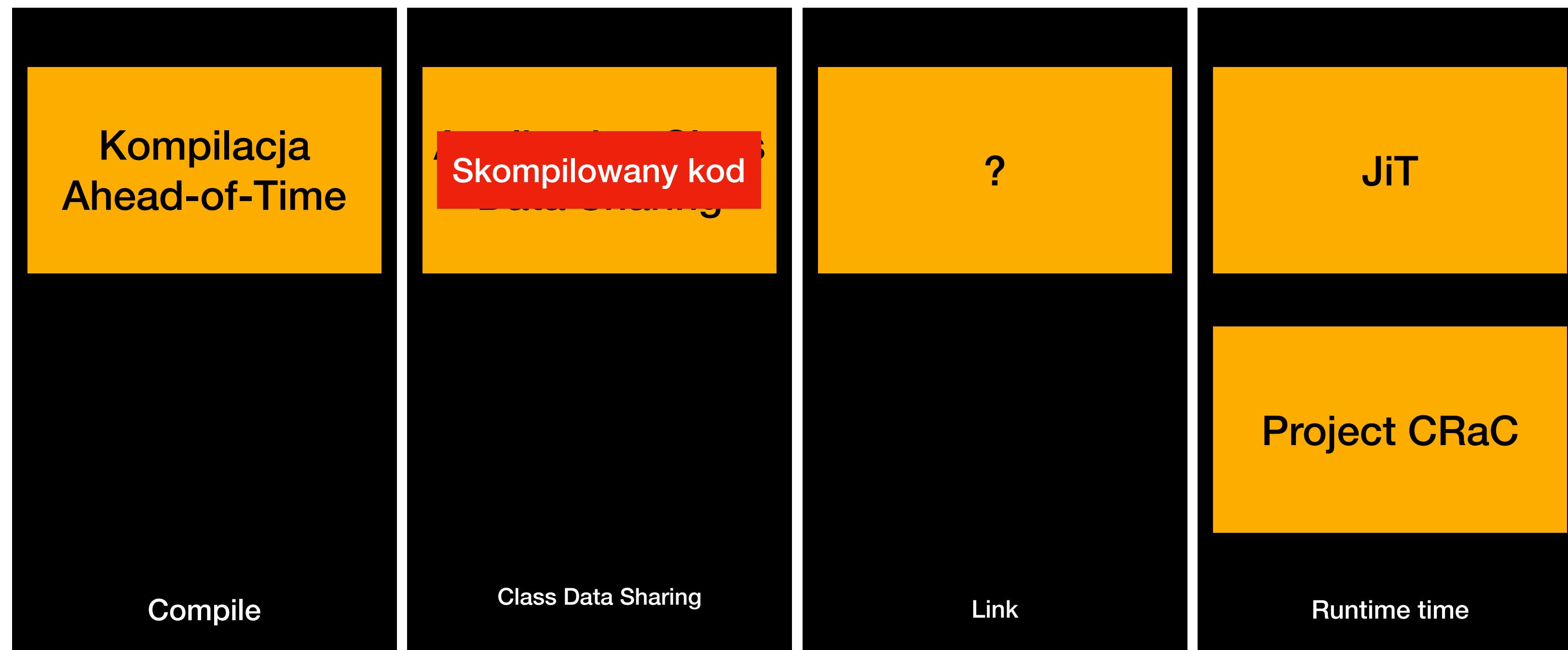
?

JiT

Project CRaC

Runtime time

Project Leyden



Project Leyden

Kod

Kompilacja
Ahead-of-Time

Compile

Class Data Sharing

Link

JiT

Runtime time

Project Leyden

Kompilacja
Ahead-of-Time

Compile

Interpretowany Bajtkod

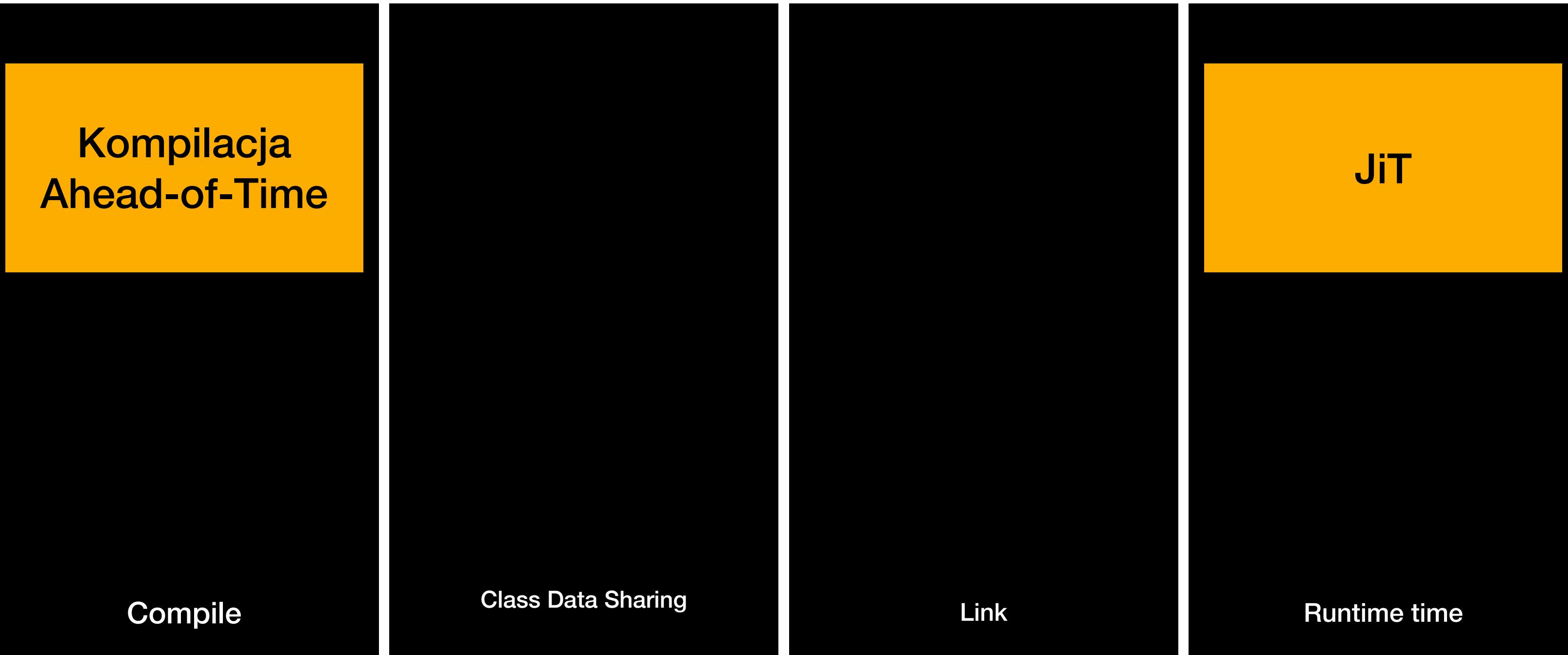
Class Data Sharing

Link

JiT

Runtime time

Project Leyden



Project Leyden



Kompilacja Ahead-of-Time

Application Class Data Sharing

Project CRaC

JiT

Condensers

Project Leyden



Kompilacja Ahead-of-Time



Application Class Data Sharing

Project CRaC

JiT

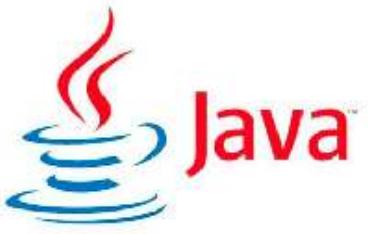
Condensers

Project Leyden

GraalVM Runtime (SubstratVM)

espresso





GraalVM JIT Compiler

GraalVM Runtime (SubstrateVM)

espresso



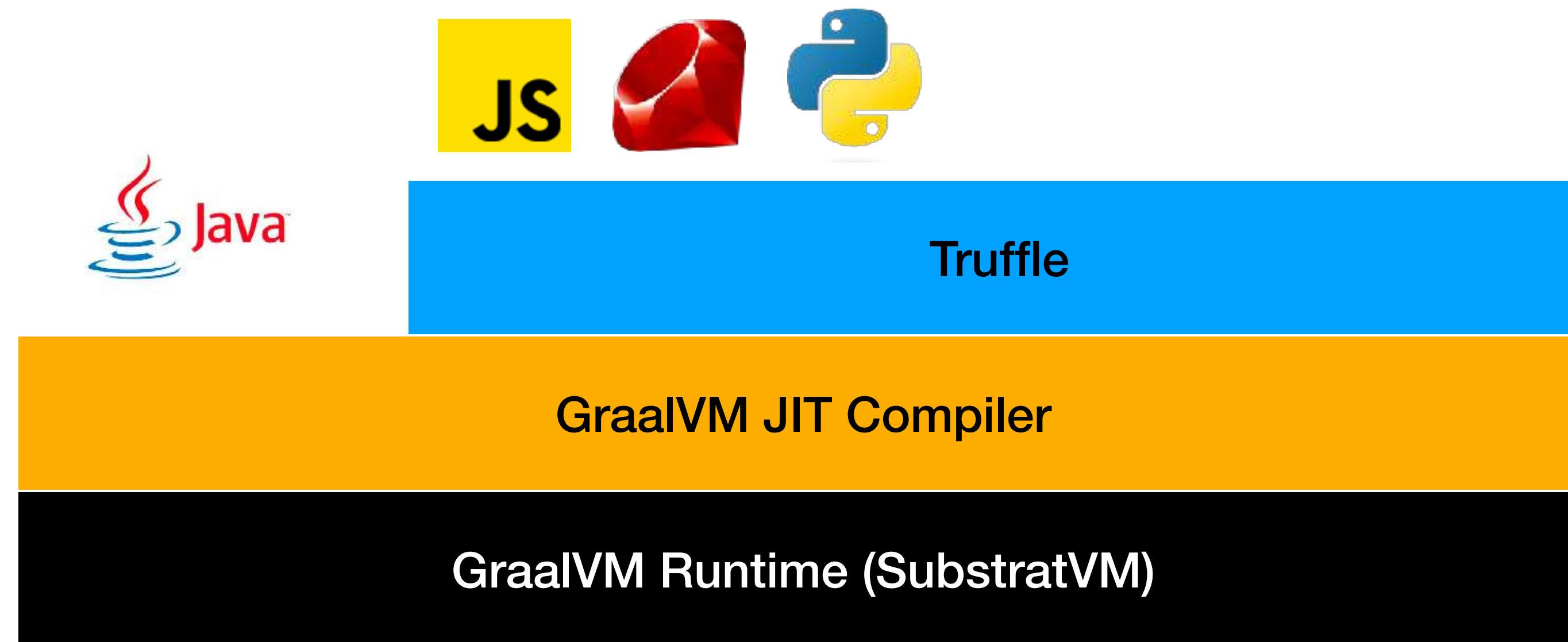


Truffle

GraalVM JIT Compiler

GraalVM Runtime (SubstrateVM)

espresso



espresso

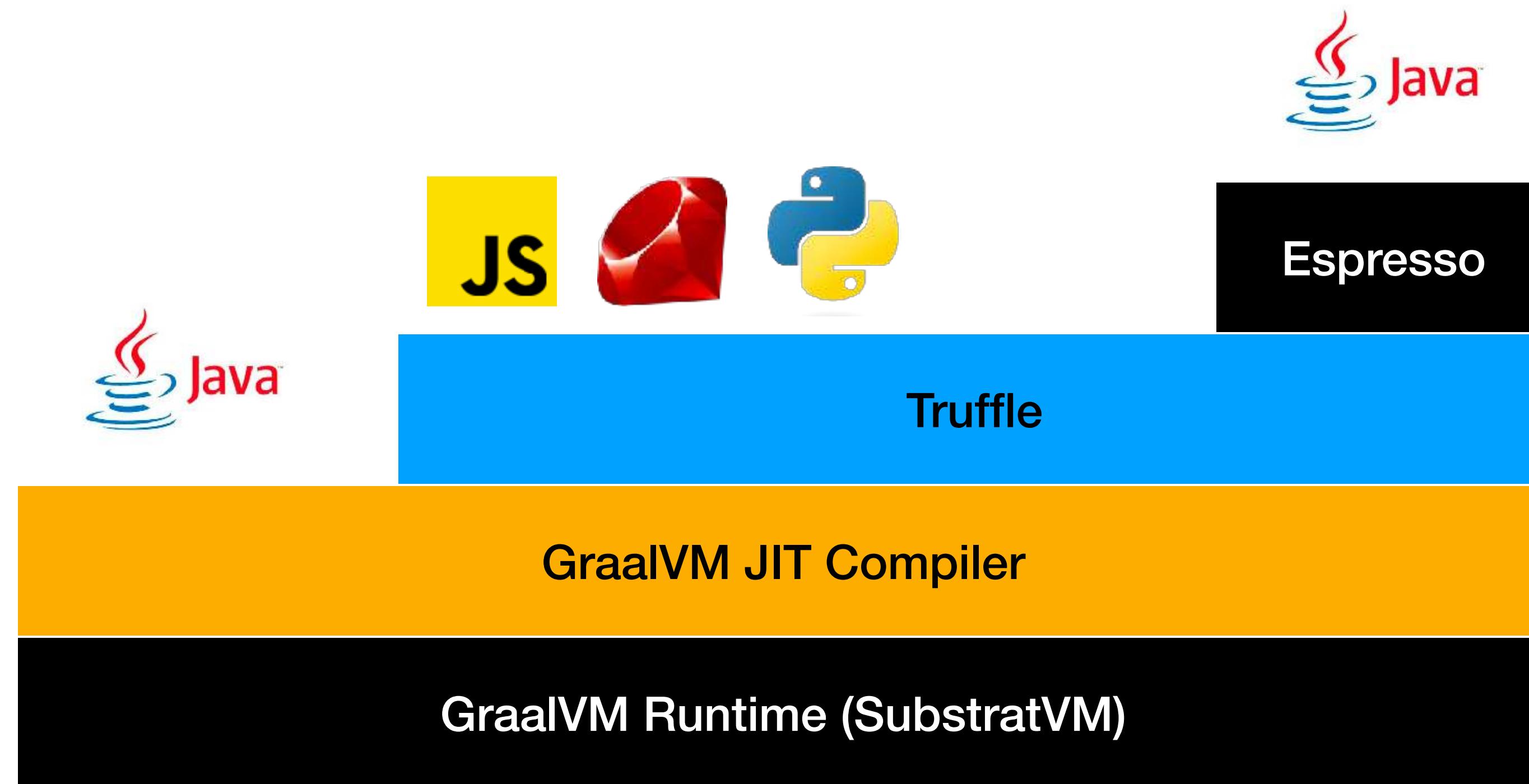


NASZA APLIKACJA

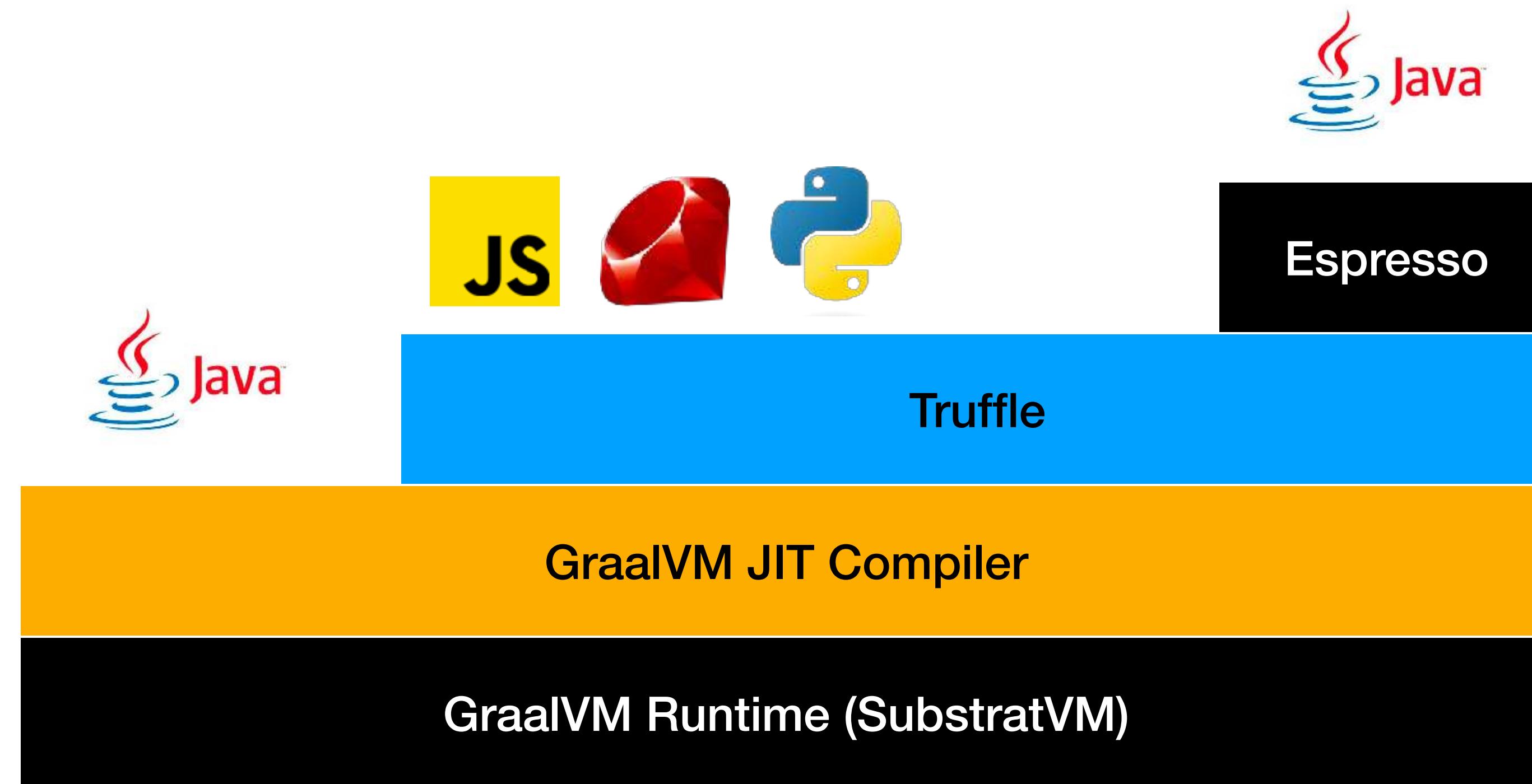


espresso





espresso



espresso

Espresso

NASZA APLIKACJA



espresso





Espresso

NASZA APLIKACJA



espresso





espresso

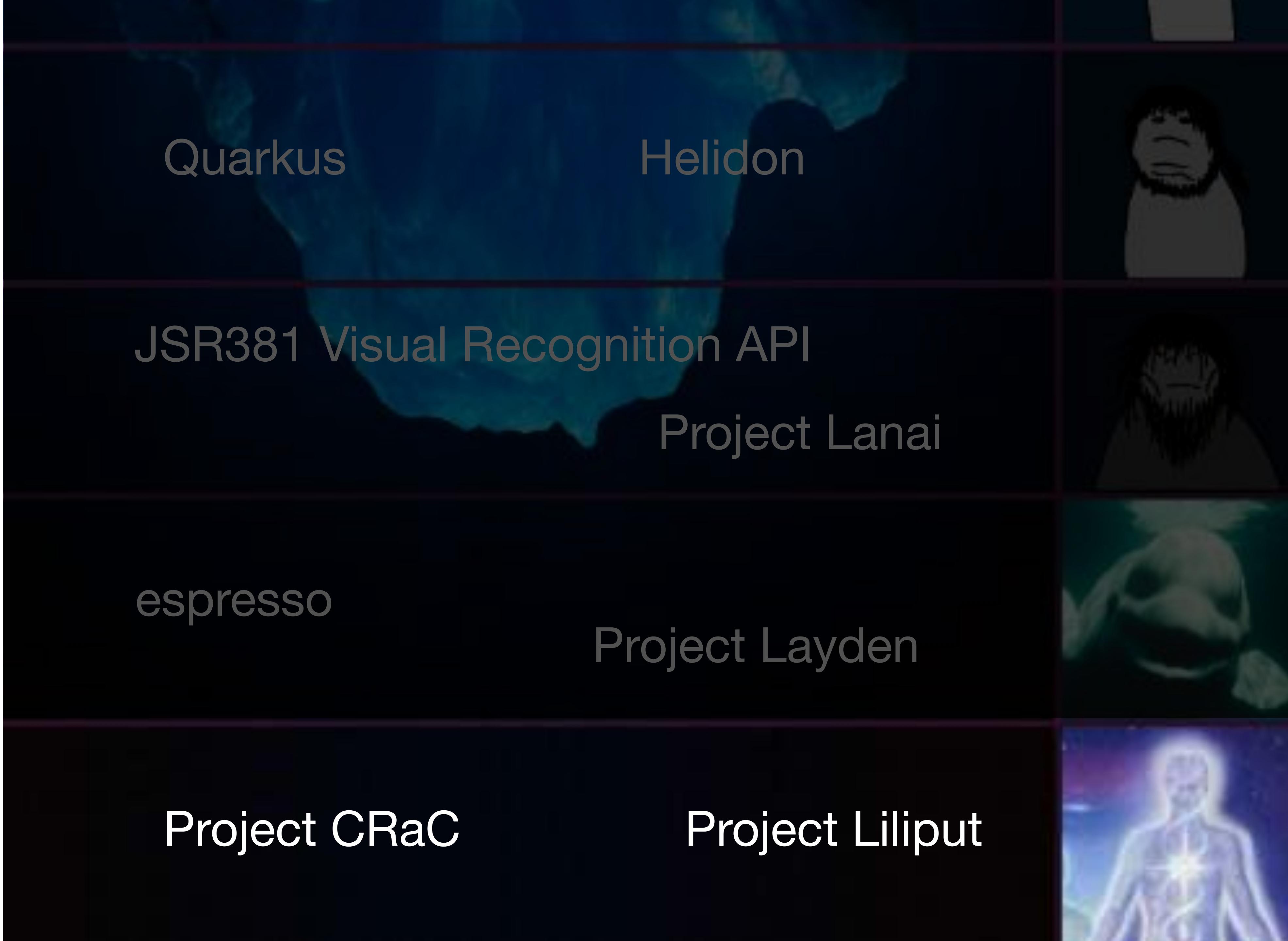




That's the evilest thing I can imagine.

espresso





Quarkus

Helidon

JSR381 Visual Recognition API

Project Lanai

espresso

Project Layden

Project CRaC

Project Liliput



Obiekt Java



Obiekt Java



Obiekt Java



Obiekt Java

Project Liliput



128 Bits

128 Bits

128 Bits

128 Bits

Obiekt Java

Obiekt Java

Obiekt Java

Obiekt Java

Project Liliput



**64 Bity dla
Garbage Collectora / Locka / etc**

**64 Bity dla
Wskaźnika Klasy**



Project Liliput

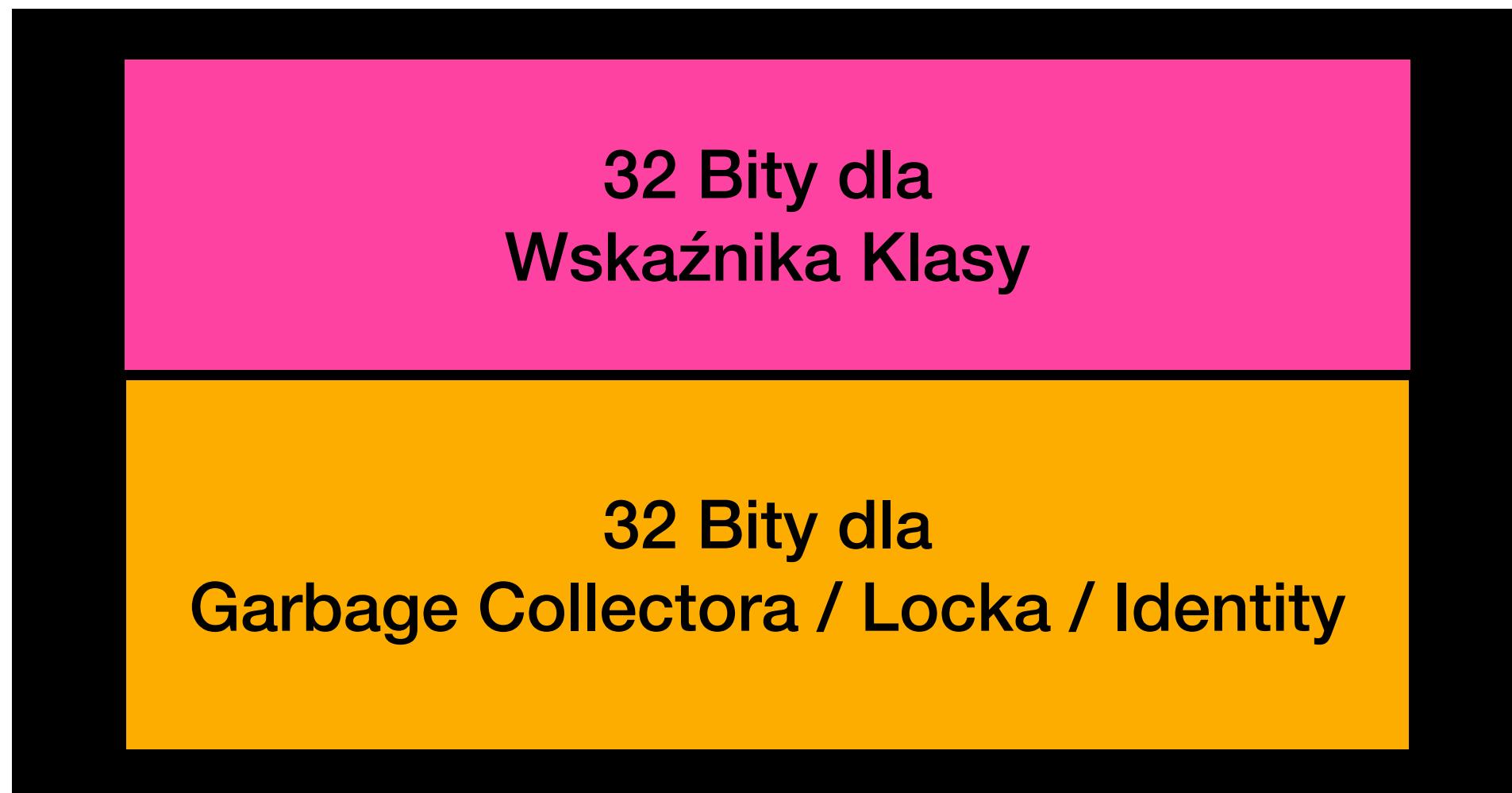


**64 Bity dla
Garbage Collectora / Locka / Identity**

**32 Bity dla
Wskaźnika Klasy**

Project Liliput





**When you delete a block
of code that you thought
was useless**



Project Liliput



Nullness markers to enable flattening

Dan Smith daniel.smith@oracle.com

Tue Feb 7 01:26:42 UTC 2023

- Previous message (by thread): [Valhalla super class/interfaces and bridges](#)
- Next message (by thread): [Nullness markers to enable flattening](#)
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

A quick review:

The Value Objects feature (see <https://openjdk.org/jeps/8277163>) captures the Valhalla project's central idea: that objects don't have to have identity, and if programmers opt out of identity, JVMs can provide optimizations comparable to primitive performance.

However, one important implementation technique is not supported by that JEP: maximally flattened heap storage. ("Maximally flattened" as in "just the bits necessary to encode an instance".) This is because flattened fields and arrays store an object's field values directly, and so 1) need to be initialized "at birth" to a non-null class instance, 2) may not store null, and 3) may be updated non-atomically. These are semantics that need to be surfaced in the language model.

We've tackled (3) by allowing value classes to be declared non-atomic (syntax/limitations subject to bikeshedding), and then claiming by fiat that fields/arrays of such classes are tearing risks. Races are rare enough that this doesn't really call for a use-site opt-in, and we don't necessarily need any deeper explanation for how new objects derived from random combinations of old objects can be created by a read operation. That's just how it works. <shrug>

We also allow value classes to declare that they support an all-zeros default instance (again, subject to bikeshedding). You could imagine similarly claiming that fields/arrays of these classes are null-hostile, as a side effect of how their storage works. But this is an idiosyncrasy that is going to affect a lot more programmers, and "that's just how it works" is pretty unsatisfactory. Sometimes programs count on being able to use 'null' in their computation. We need something in the language model to let programs opt in/out of nulls at the use site, and thus opt out/in of maximally flattenable heap storage.

We've long discussed "reference type" vs. "value type" as the language concept that captures this distinction. But where we once had a long list of differences between references and values, most of those have gone away. Notably, it's *not* useful for performance intuitions to imagine that references are pointers and values are inline. Value objects get inlined when the JVM want to do so. Reference-ness is not relevant.

Really, for most programmers, nullness is all that distinguishes a "reference type" from a "value type".

Meanwhile, expressing nullness is not a problem unique to Valhalla. Whether a variable is meant to store nulls is probably the most important property of most programs that isn't expressible in the language. Workarounds include informal javadoc specifications, type annotations (as explored by JSPECIFY), lots of 'Objects.requireNonNull' calls, and blanket "if you pass in a null, you might get an NPE" policies.

In Amber, pattern matching has its own problems with nullness: there are a lot of ad hoc rules to distinguish between "is this a non-null instance of class Foo?" vs. "is this null *or* an instance of class Foo?", because there's no good way to express those two queries as explicitly different.

To address these problems, we've been exploring nullness markers as an alternative to '.val' and '.ref'. The goal is a general-purpose feature that lets programmers express intent about nulls, and that is preserved at runtime sufficiently for JVMs to observe that "not null" + "value class" + "non-atomic (or compact) class" --> "maximally flattenable storage". There are no "value types", and there is no direct control over flattenable.

(A lot of these ideas build on what JSPECIFY has done, so appreciation to them for the good work and useful documentation.)

Some key ideas:





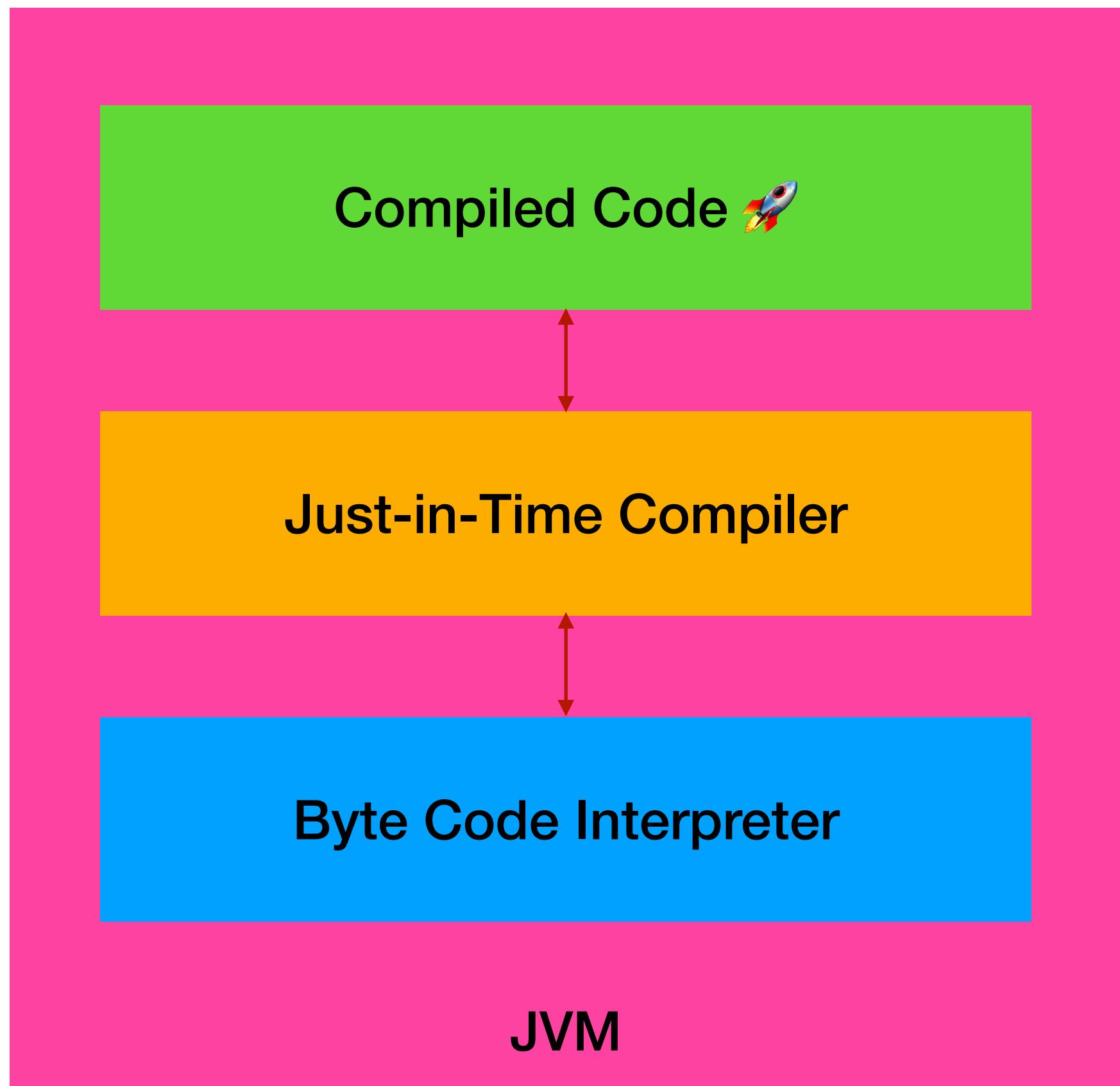
Project CRaC





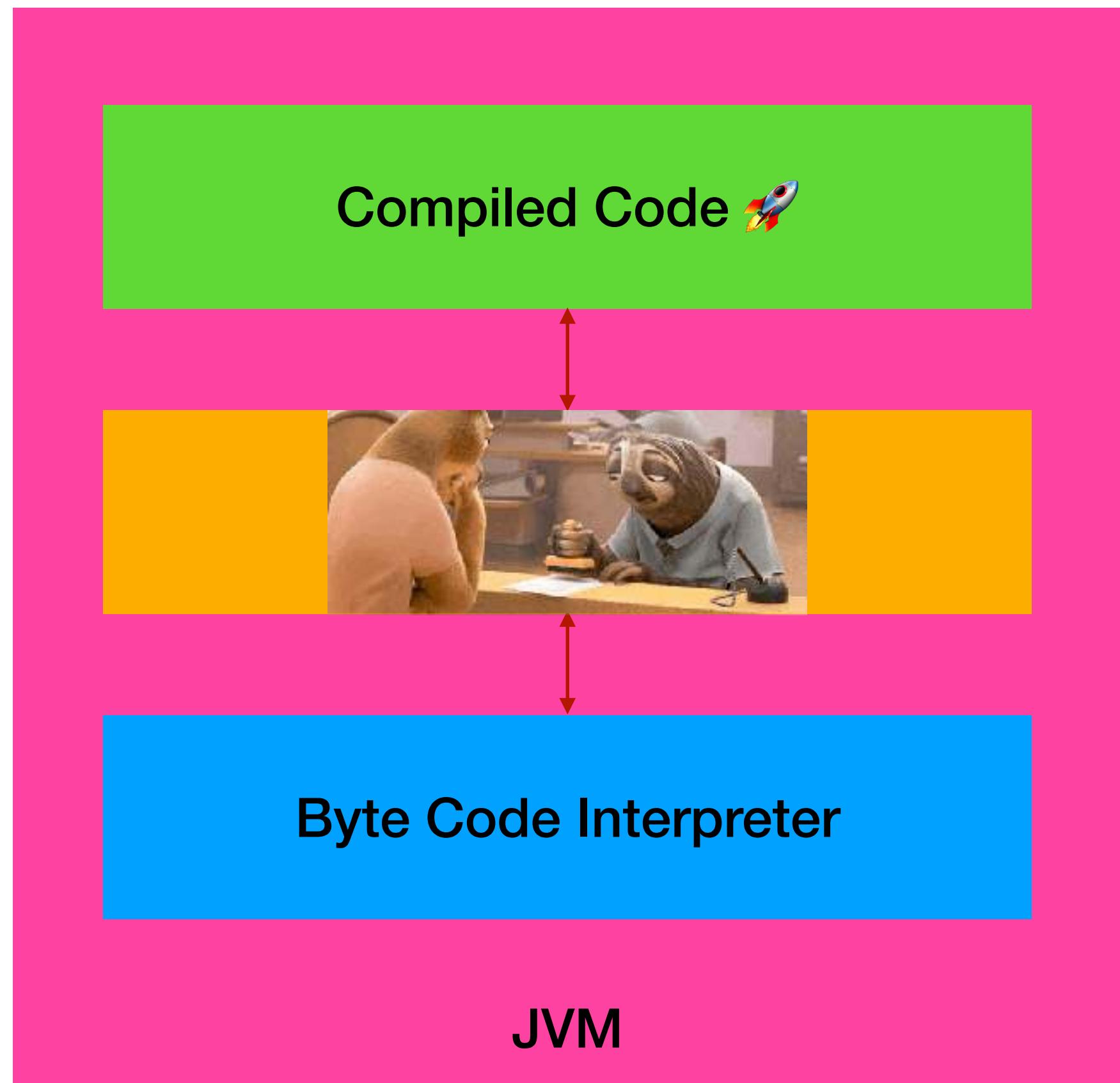
Project Coordinated Restore at Checkpoint





Project CRaC





Project CRaC





**NOOO... YOU CANNOT
SAVE STATE, YOU ARE
BREAKING ORIGINAL EXPERIENCE**

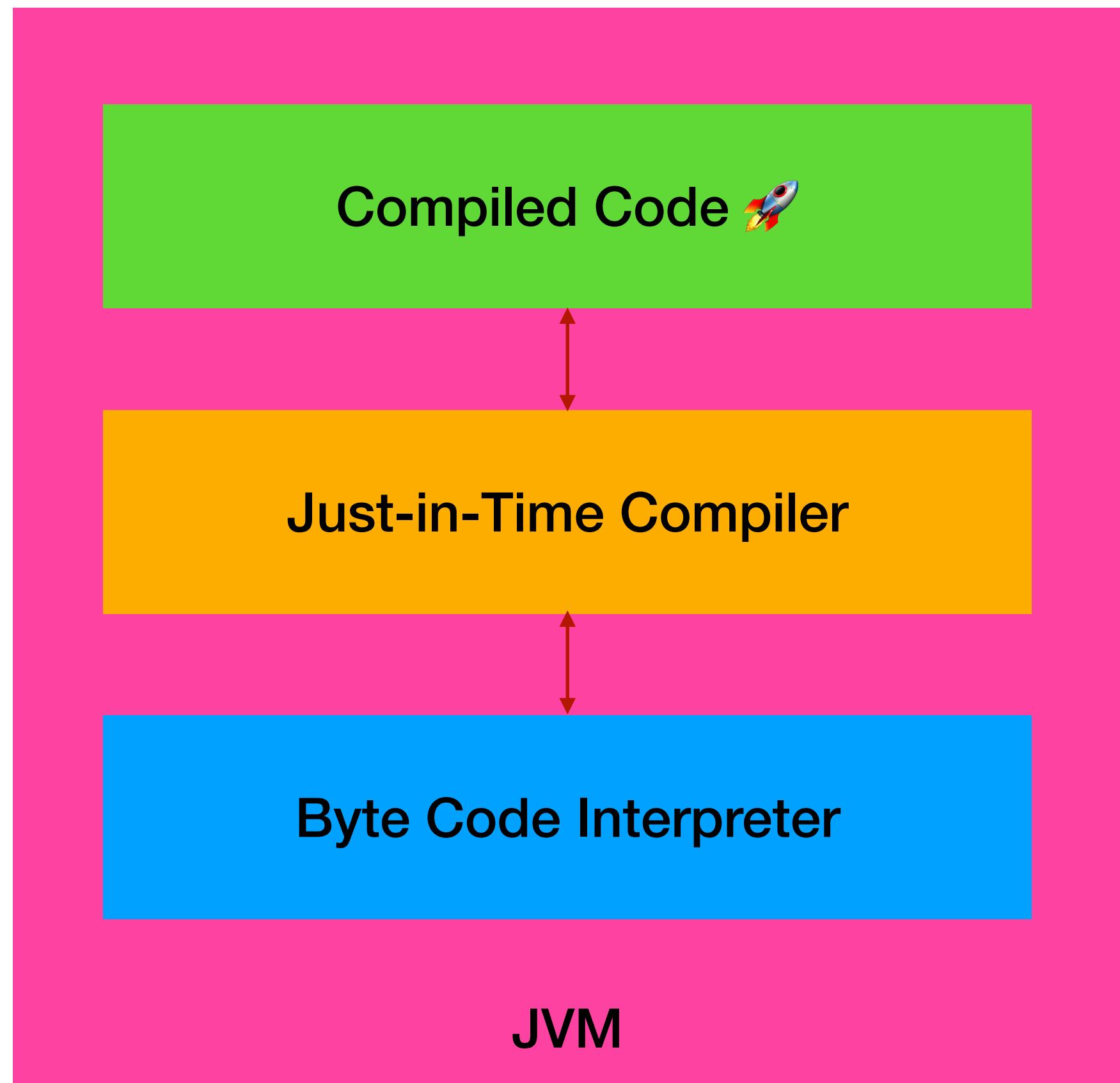
imgflip.com



**HAHA, EMULATOR
GOES BRRRRRR**

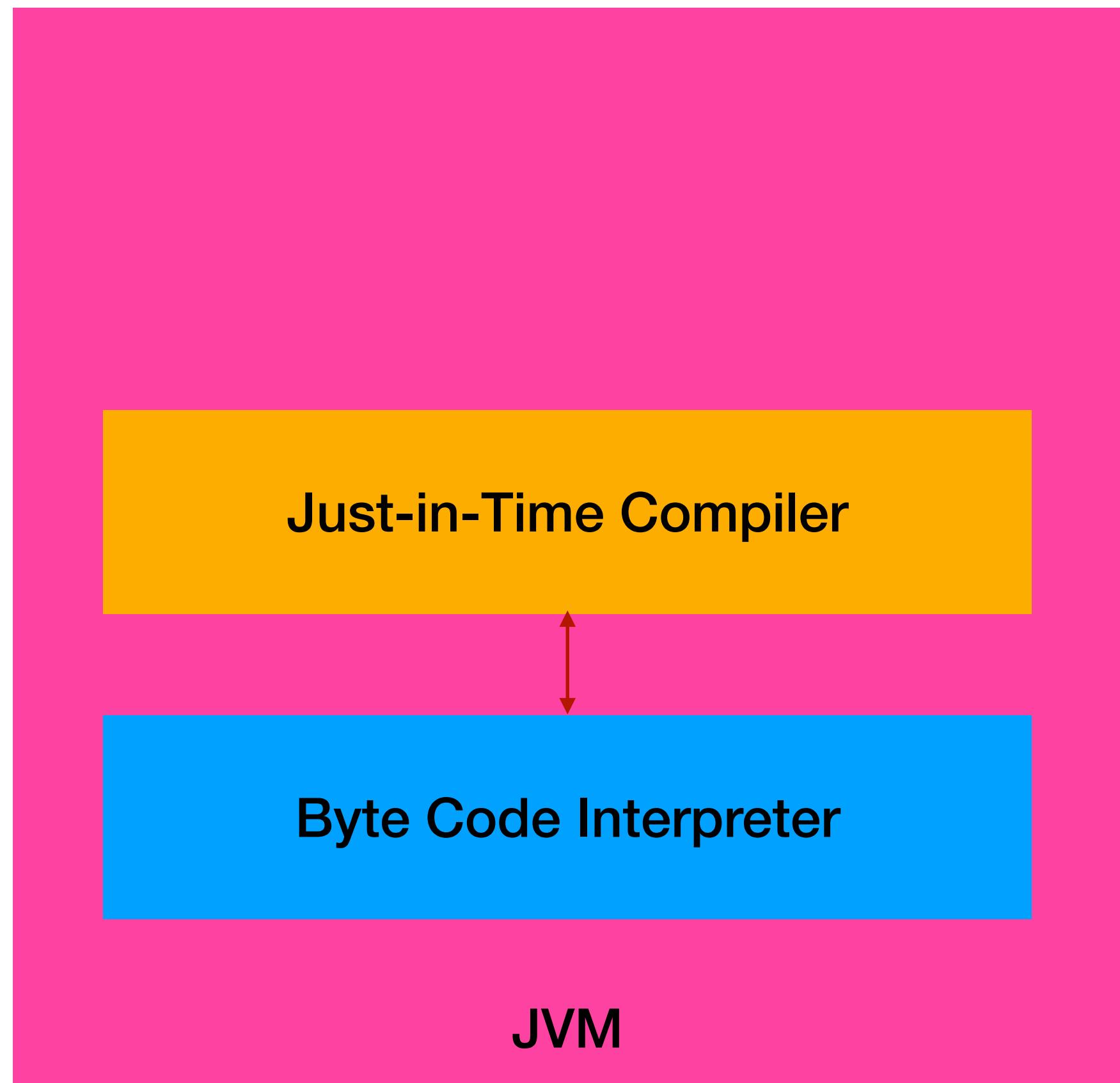
Project CRaC





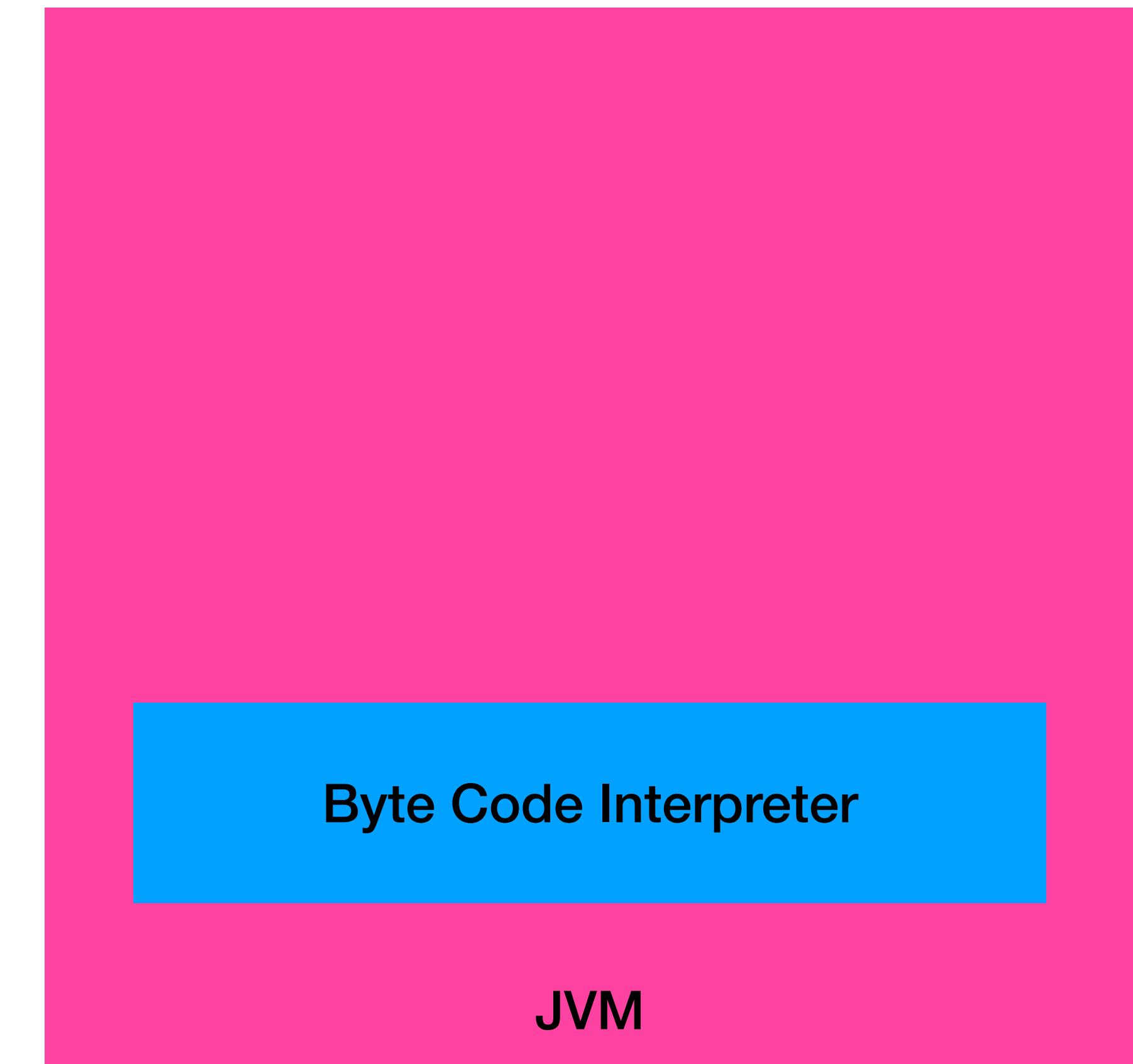
Project CRaC





Project CRaC





Compiled Code 🚀

Project CRaC



Compiled Code 🚀

Byte Code Interpreter

JVM

Project CRaC



NEW

NEW

AWS Lambda SnapStart

Drastically reduce Lambda function cold start latency at no extra cost

GENERALLY AVAILABLE TODAY

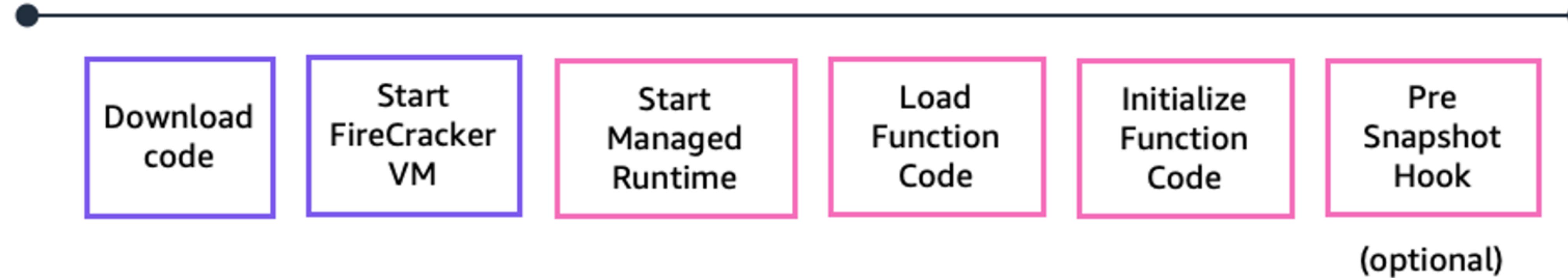


NEW

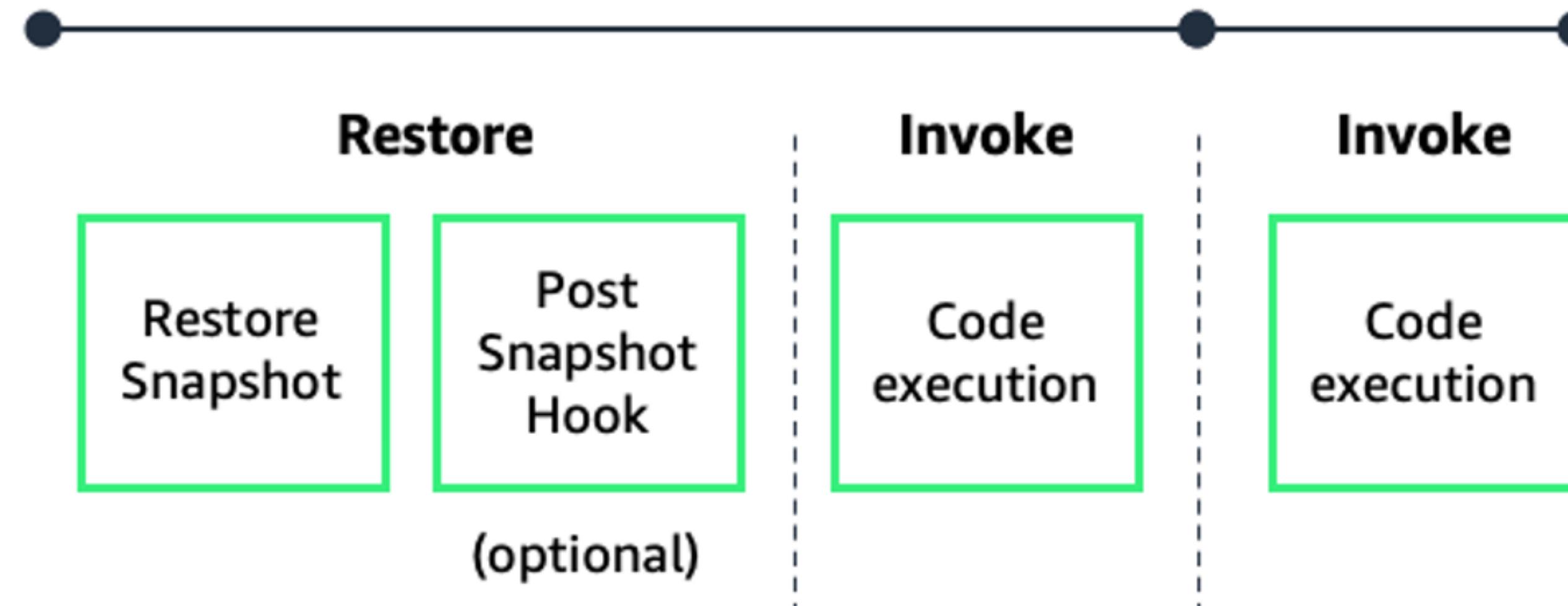
Project CRaC



Deployment



Invocation



Project CRaC



?

Project CRaC



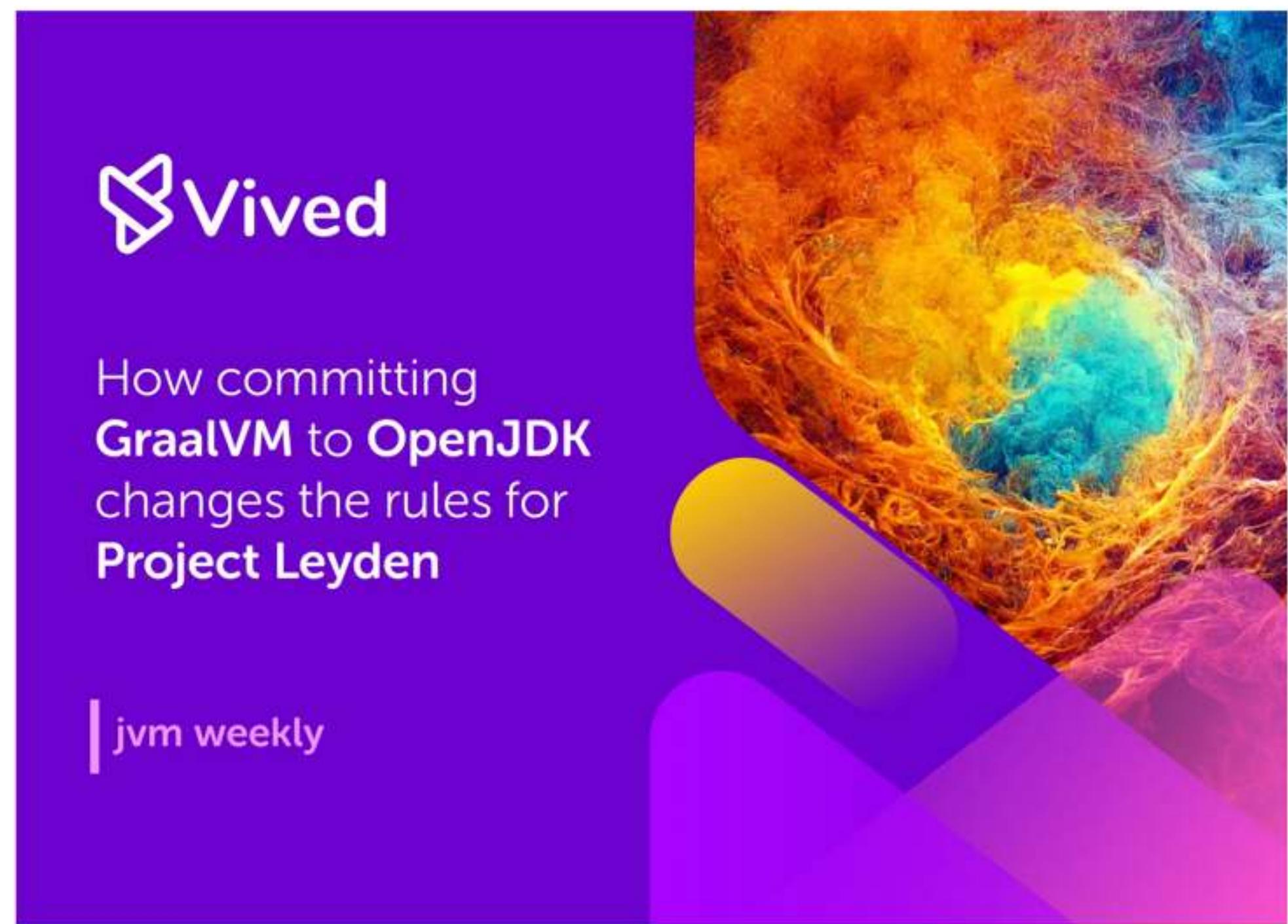


How contributing GraalVM to OpenJDK changes the rules for Project Leyden - JVM Weekly #19

Instead of the standard division, I decided to prepare one coherent article, in which I will try to guide you through everything that is happening in Project Leyden and GraalVM.

 Artur Skowronski
21 hr ago

...    





Thank you



@ArturSkowronski



jvm weekly