



ORACLE

ORACLE



# New Opportunities for Developers with GraalVM

Alina Yurenko

GraalVM Developer Advocate

Oracle Labs

October 08, 2019

## Safe harbor statement

---

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# Agenda

---

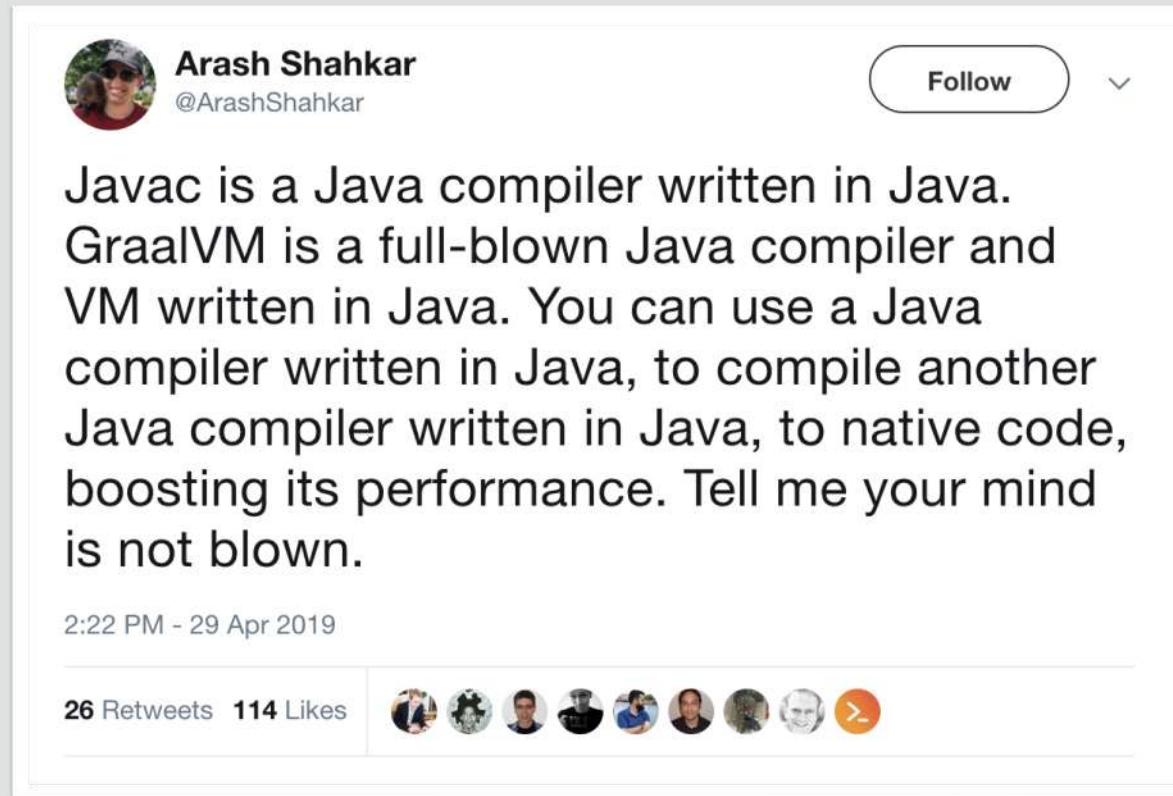
- 1 What's under the hood
- 2 Performance Optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next

# Agenda

---

- 1 What's under the hood
- 2 Performance Optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next

## GraalVM magic in one tweet



A screenshot of a tweet from Arash Shahkar (@ArashShahkar) dated 2:22 PM on April 29, 2019. The tweet text reads: "Javac is a Java compiler written in Java. GraalVM is a full-blown Java compiler and VM written in Java. You can use a Java compiler written in Java, to compile another Java compiler written in Java, to native code, boosting its performance. Tell me your mind is not blown." The tweet has 26 retweets and 114 likes. The interface includes a profile picture, name, handle, a 'Follow' button, and a list of user avatars who interacted with the tweet.

**Arash Shahkar**  
@ArashShahkar

Follow

Javac is a Java compiler written in Java. GraalVM is a full-blown Java compiler and VM written in Java. You can use a Java compiler written in Java, to compile another Java compiler written in Java, to native code, boosting its performance. Tell me your mind is not blown.

2:22 PM - 29 Apr 2019

26 Retweets 114 Likes

# Why GraalVM

---



## High Performance

Optimize application performance with GraalVM compiler



## Fast Startup

Compile your application AOT and start instantly



## Polyglot

Mix & match languages with seamless interop



## Open Source

See what's inside, track features progress, contribute



# GraalVM™





# Production-ready! 🎉

---

📌 Pinned Tweet



**GraalVM** @graalvm · May 9

First production release - we are stoked to introduce GraalVM 19.0! 🚀🏆

Here's the announcement: [medium.com/graalvm/announ....](https://medium.com/graalvm/announ...)

Check out the release notes: [graalvm.org/docs/release-n...](https://graalvm.org/docs/release-n...) and get the binaries:

💬 14

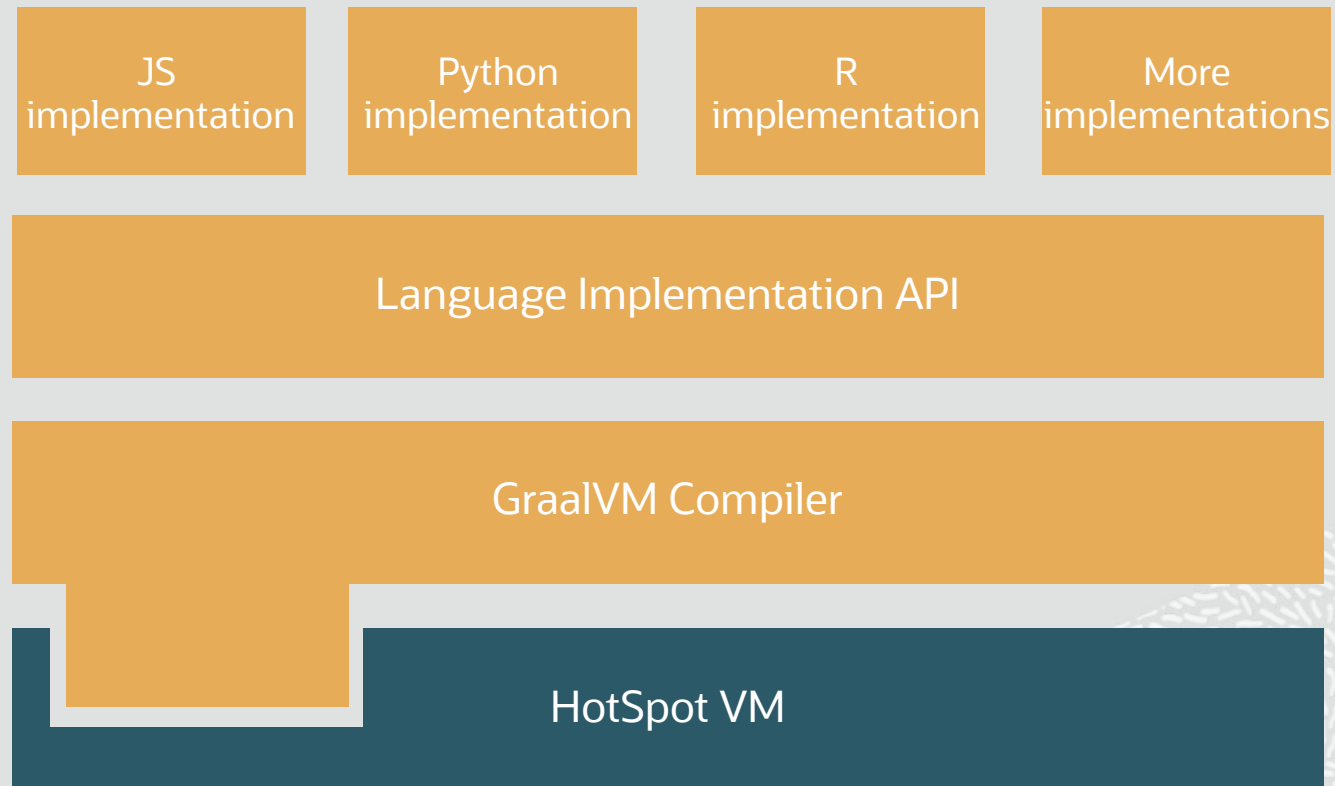
↻ 506

❤️ 822

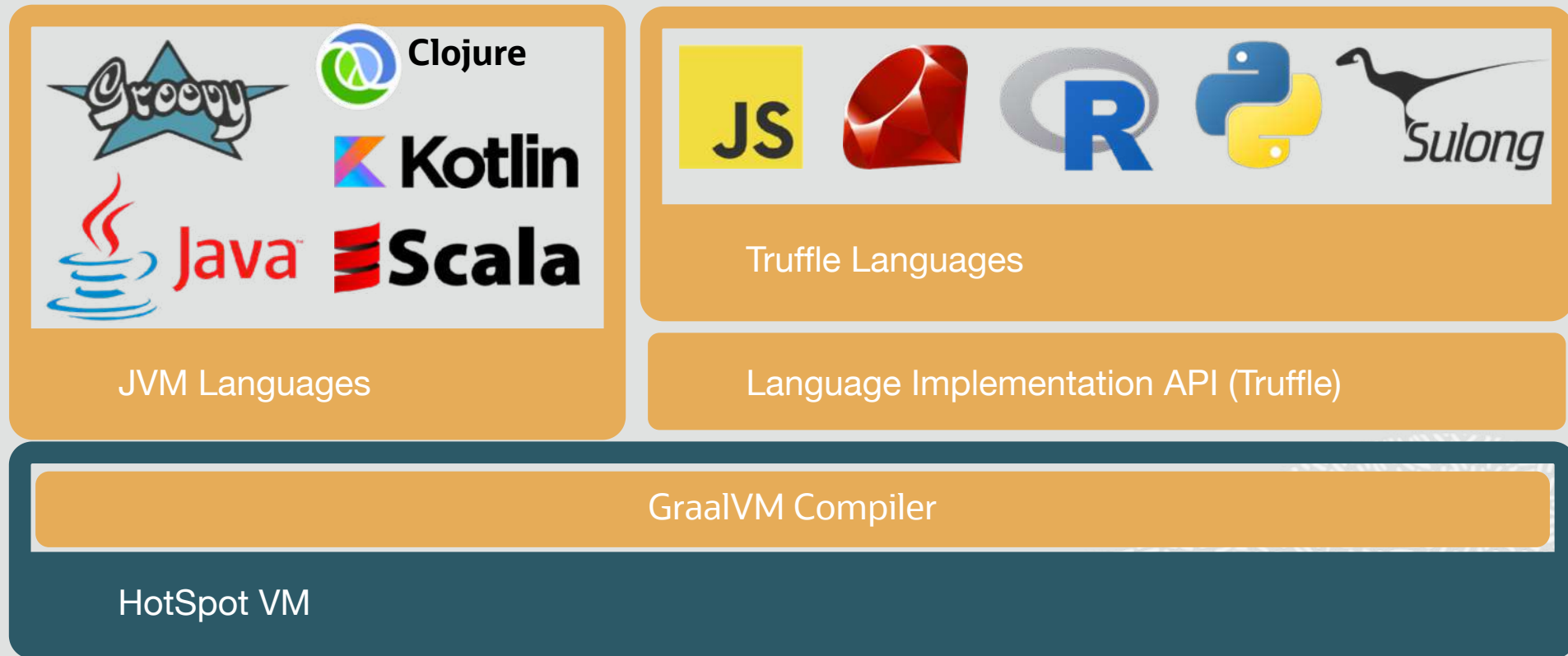


# Architecture

---

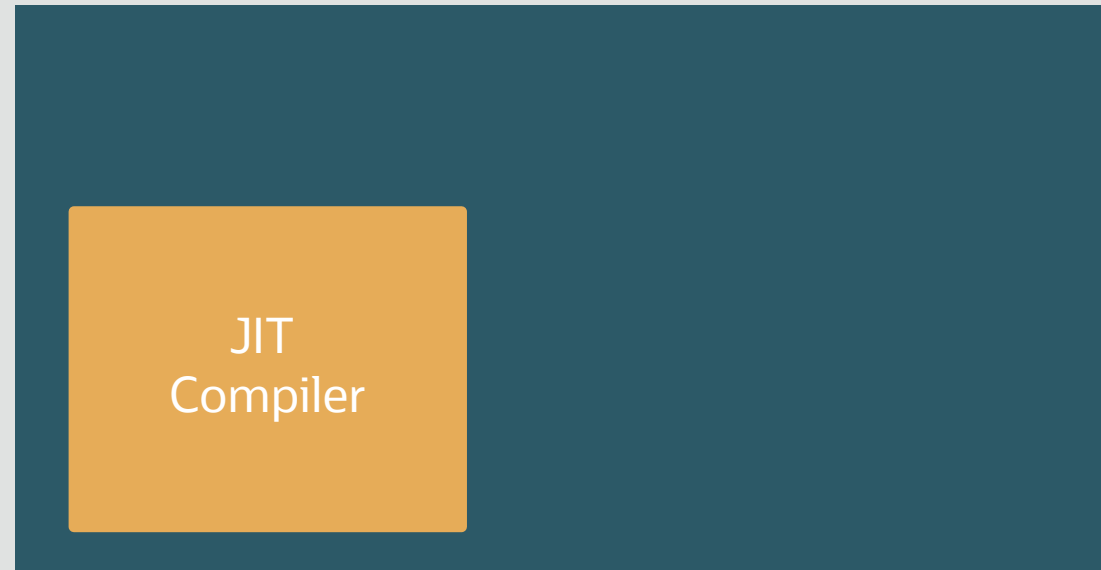
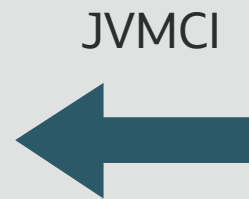


# Architecture



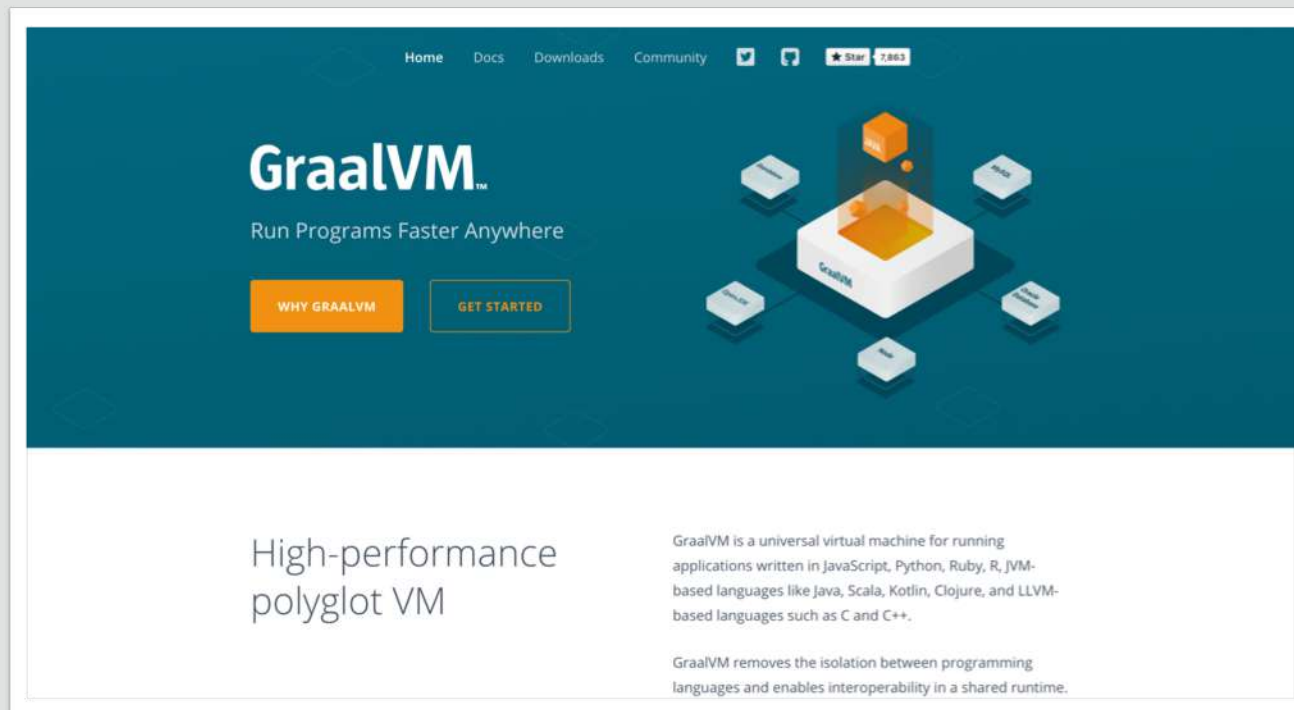
# Compiler <> VM Interaction

HotSpot VM



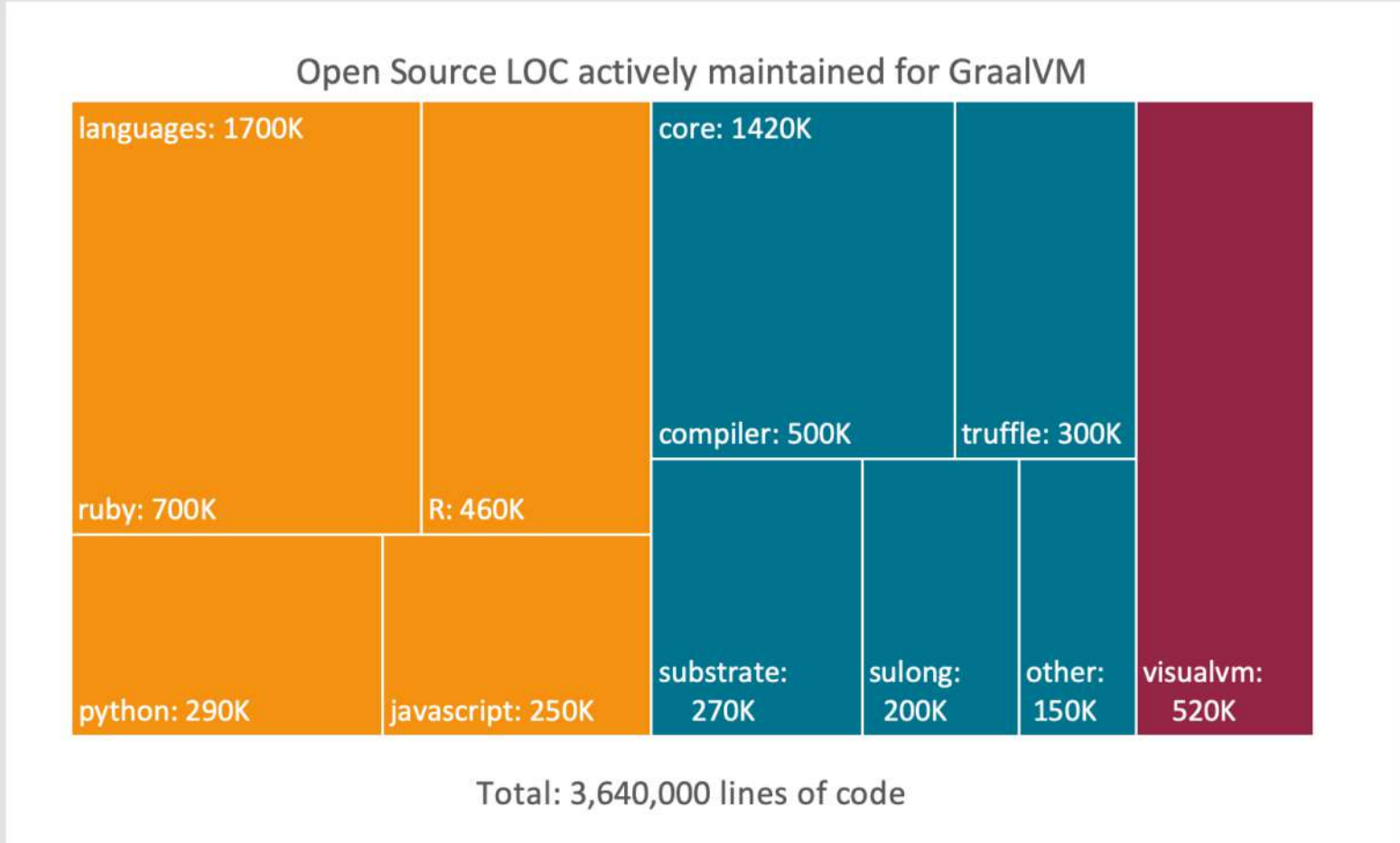
# Get Started

---



- Downloads
  - Documentation
  - Community
- support

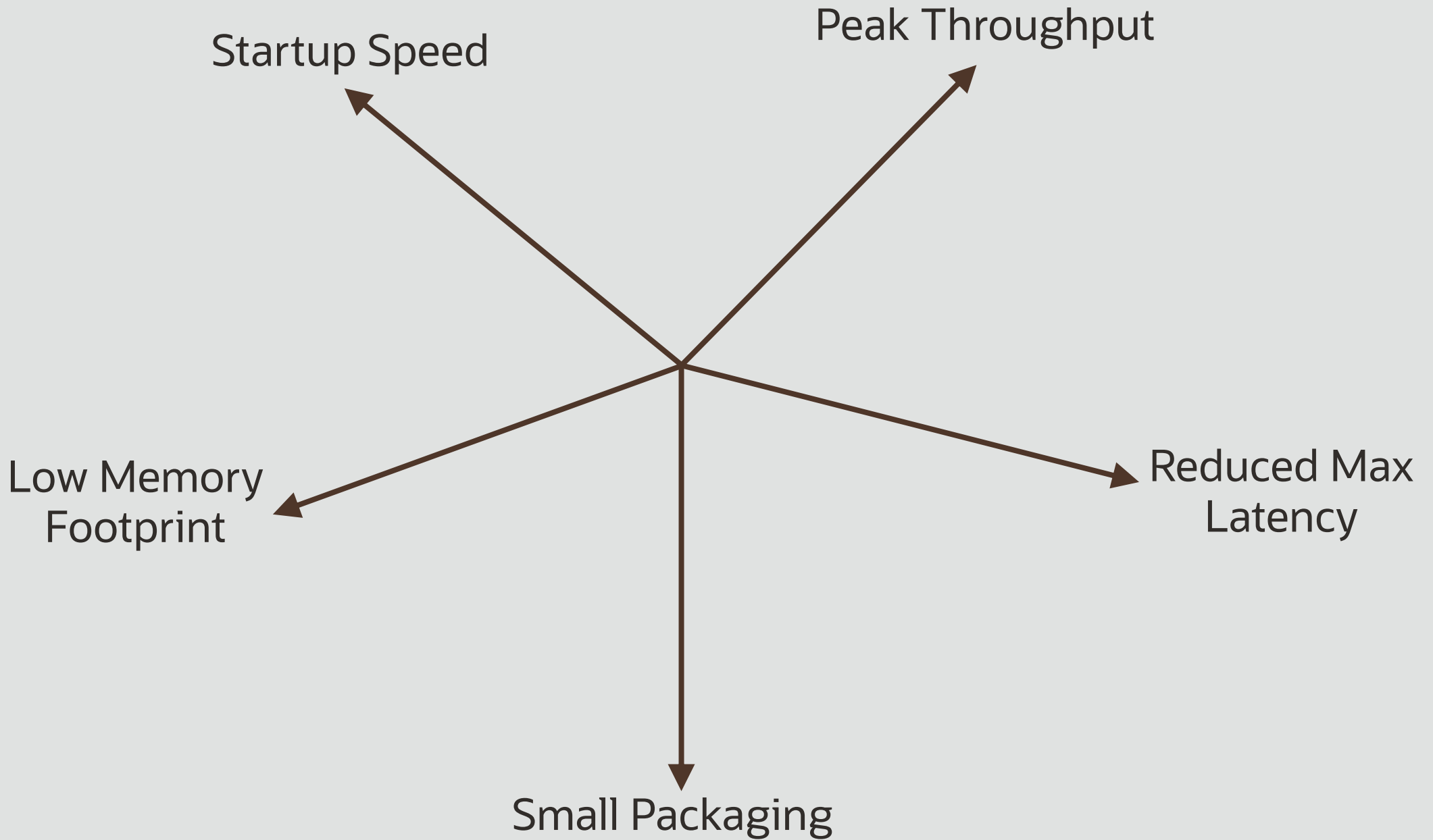
# GraalVM Open Source



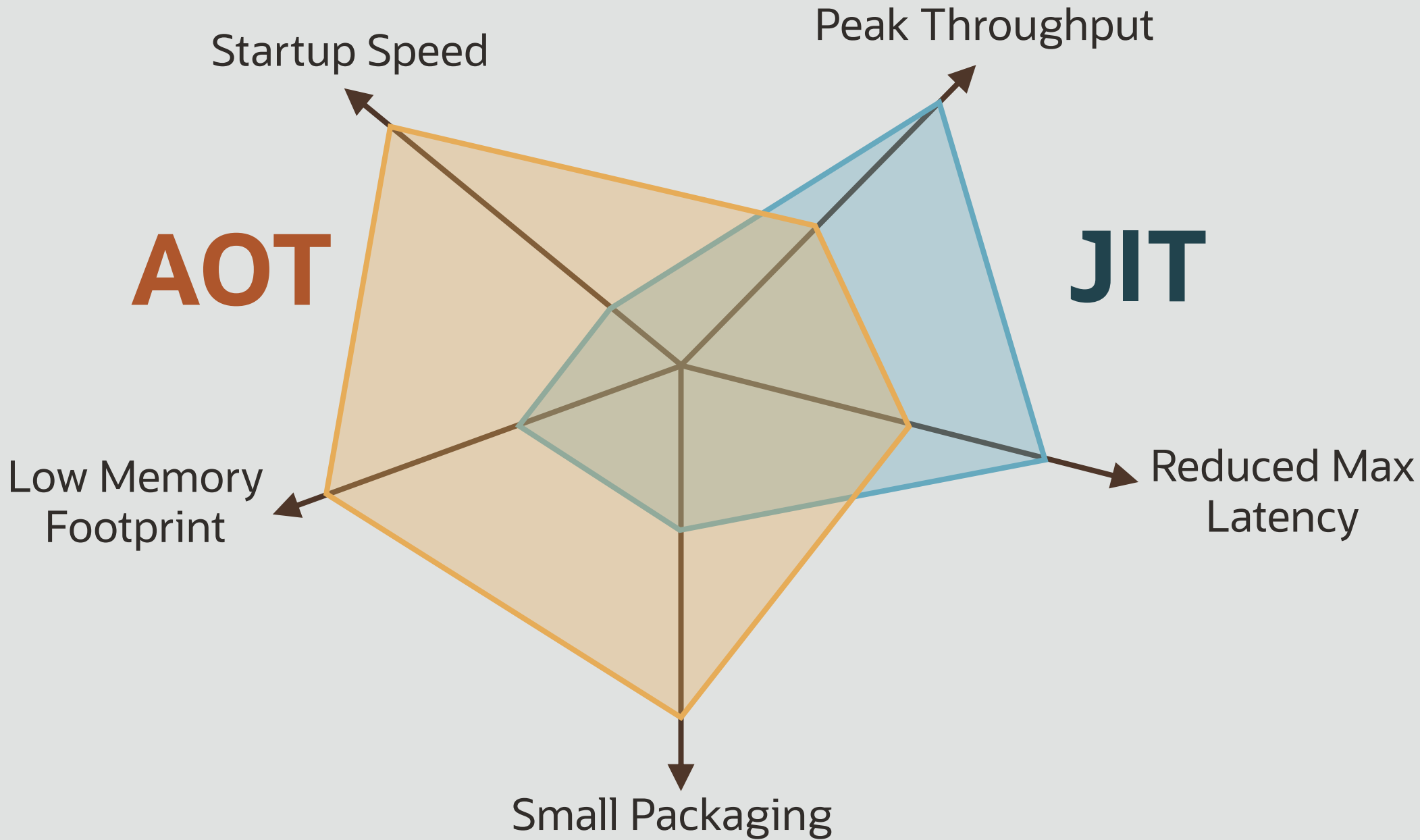
# Agenda

---

- 1 What's under the hood
- 2 Performance optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next



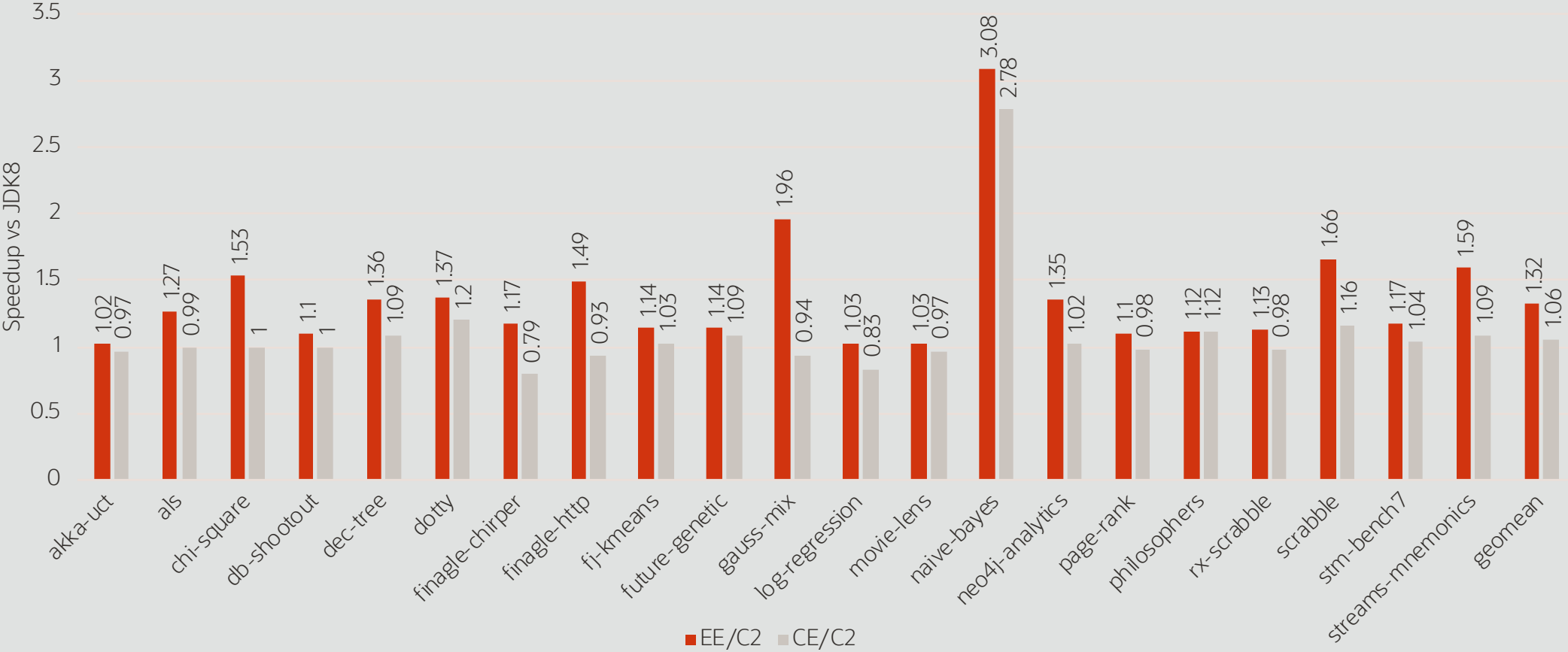




# Demo time



# GraalVM JIT Performance: Renaissance.dev



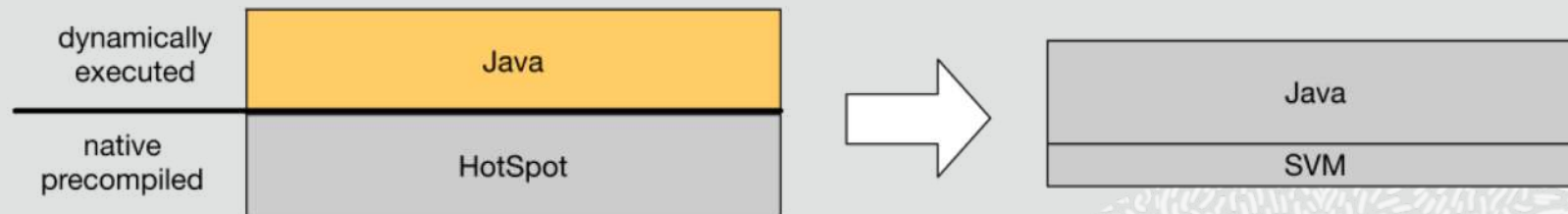
# Agenda

---

- 1 What's under the hood
- 2 Performance Optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next

# GraalVM Native Images

- Instant startup;
- Low memory footprint;
- Works with memory management;
- AOT-compiled using the GraalVM compiler.



## Native Image: Details

**Input:**  
All application classes,  
libraries, and VM

Application

Libraries

JDK

Substrate VM

Points-to Analysis

Run Initializations

Heap Snapshotting

Iterative analysis until  
fixed point is reached

Ahead-of-Time  
Compilation

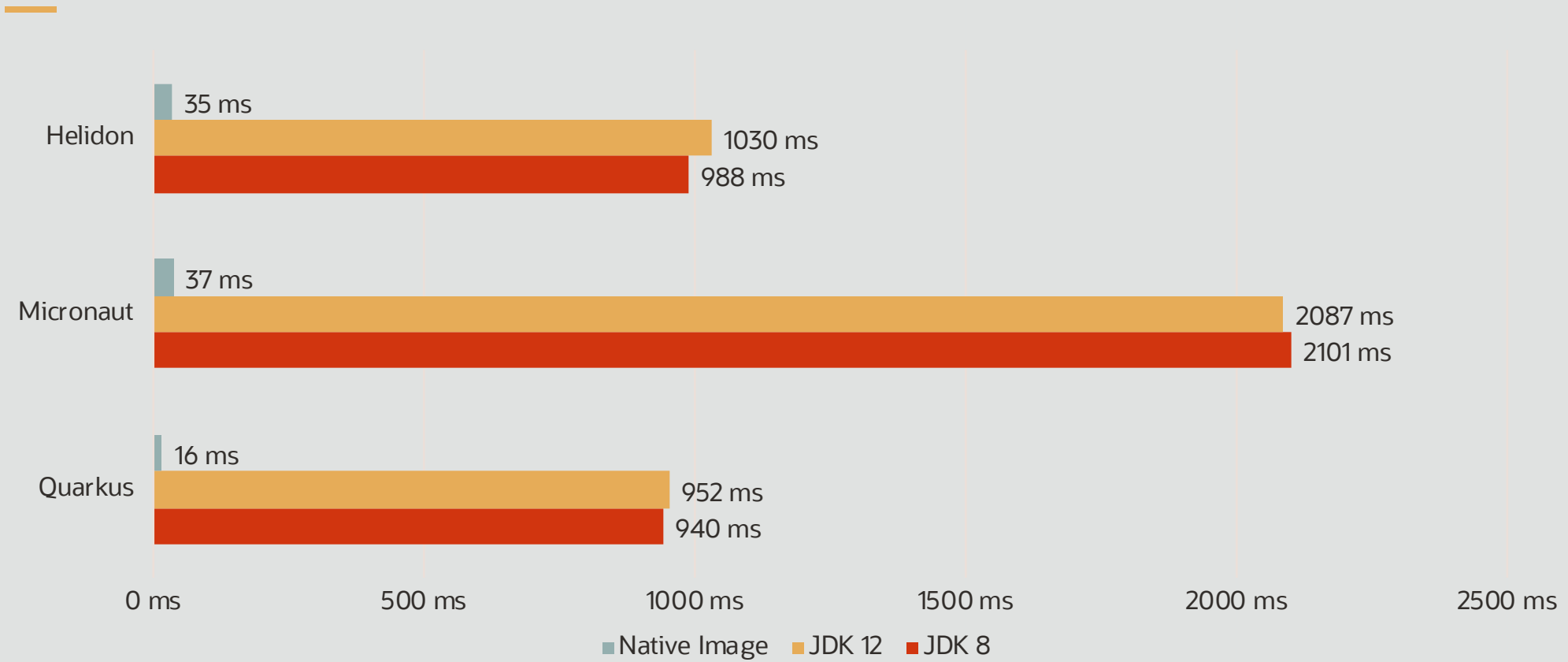
Image Heap  
Writing

**Output:**  
A native executable

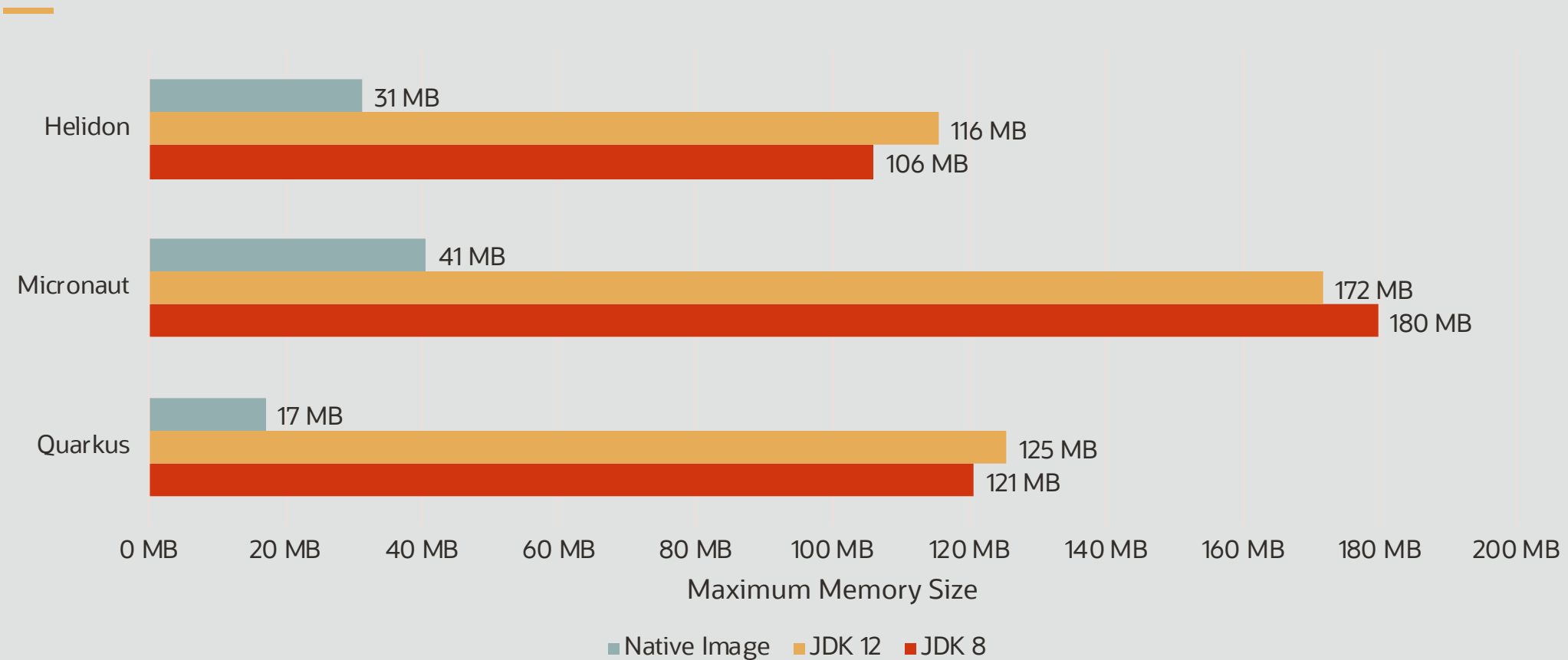
Code in  
Text Section

Image Heap in  
Data Section

# Microservice Frameworks: Startup Time



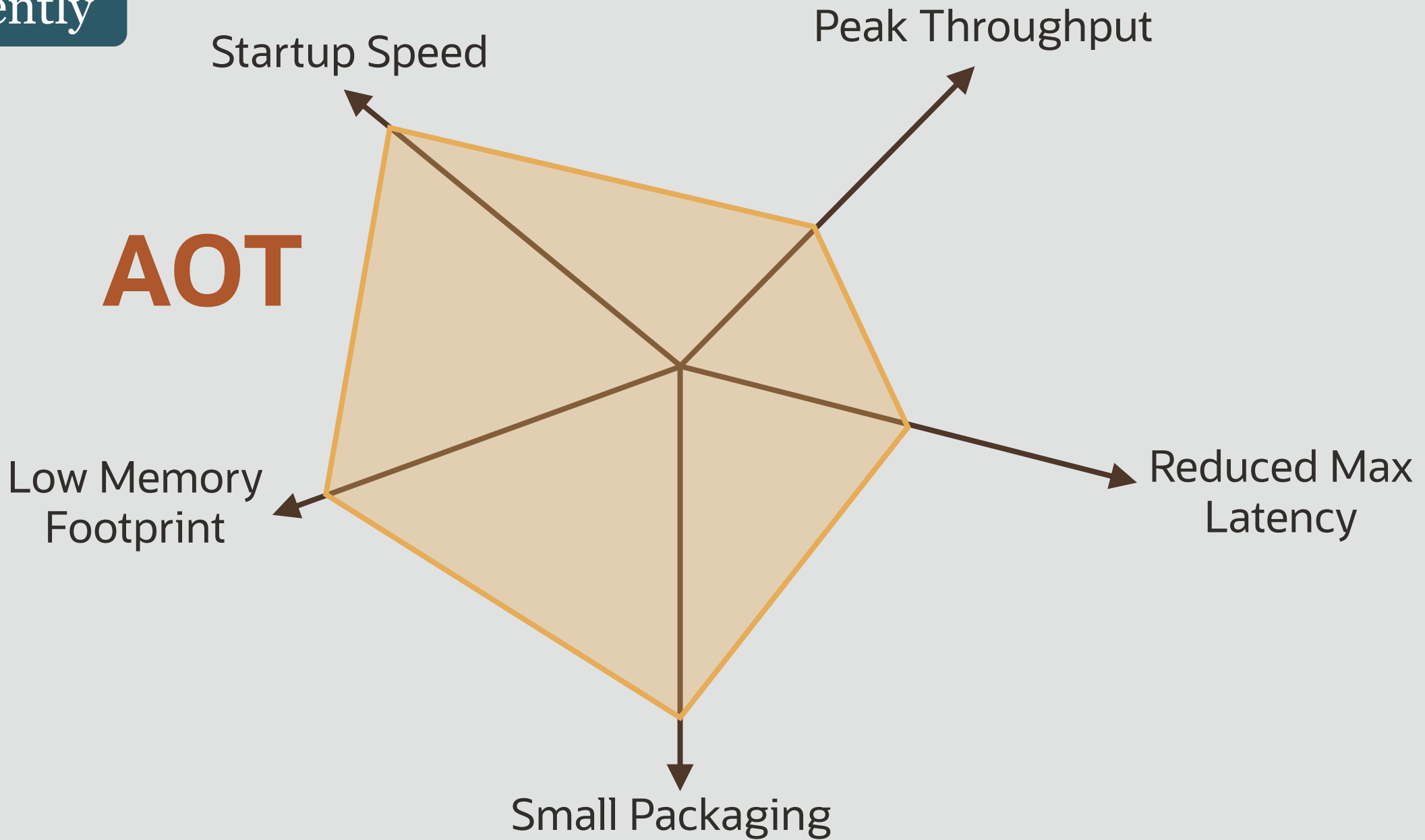
# Microservice Frameworks: Memory Usage





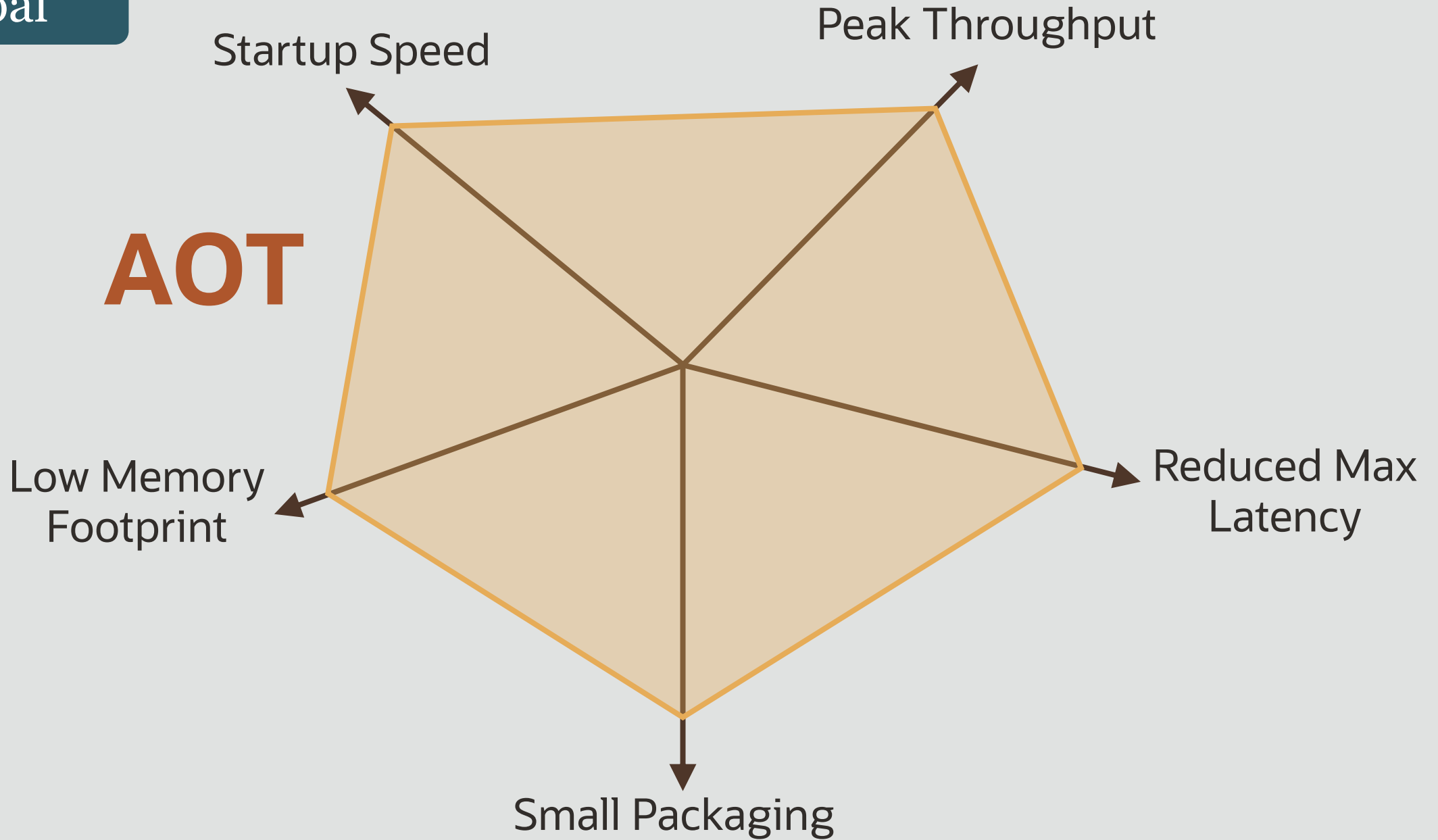
Currently

**AOT**



# Goal

# AOT



## Native Image: Profile-Guided Optimizations (PGO)

---

The GraalVM compiler is built ground-up with profiles in mind

Collecting profiles is essential for performance of native images

PGO requires running relevant workloads before building an image

```
$ java -Dgraal.PGOInstrument=myclass.iprof MyClass
```

```
$ native-image --pgo=myclass.iprof MyClass
```

```
$ ./myclass
```

# Simplifying the Native Image Configuration

---

## Introducing the Tracing Agent: Simplifying GraalVM Native Image Configuration



Christian Wimmer [Follow](#)

Jun 5 · 6 min read

*tl;dr: The tracing agent records behavior of a Java application running, for example, on GraalVM or any other compatible JVM, to provide the GraalVM Native Image Generator with configuration files for reflection, JNI, resource, and proxy usage. Enable it using `java -agentlib:native-image-agent=...`*

## Continue Learning About GraalVM Native Images

---

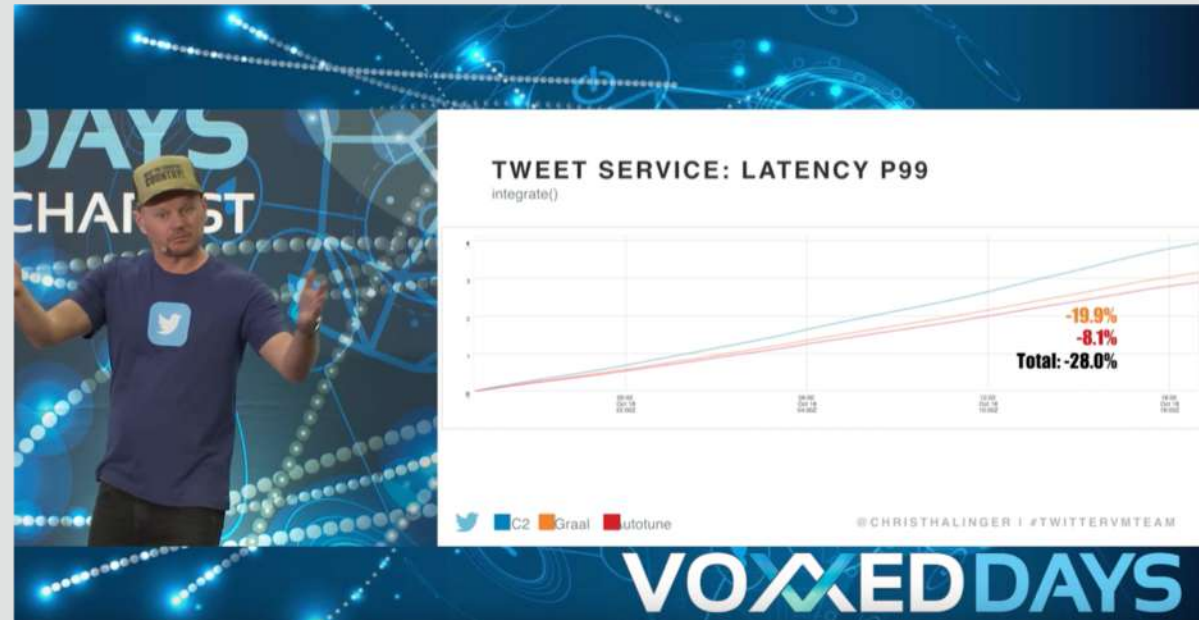
- Reference manual: [graalvm.org/docs/reference-manual/aot-compilation/](https://graalvm.org/docs/reference-manual/aot-compilation/)
- Improving performance of GraalVM native images with PGO:  
<https://medium.com/graalvm/improving-performance-of-graalvm-native-images-with-profile-guided-optimizations-9c431a834edb>
- GraalVM Native Images: The Best Startup Solution for Your Applications:  
<https://www.youtube.com/watch?v=z0jedLjcWjl>

# Agenda

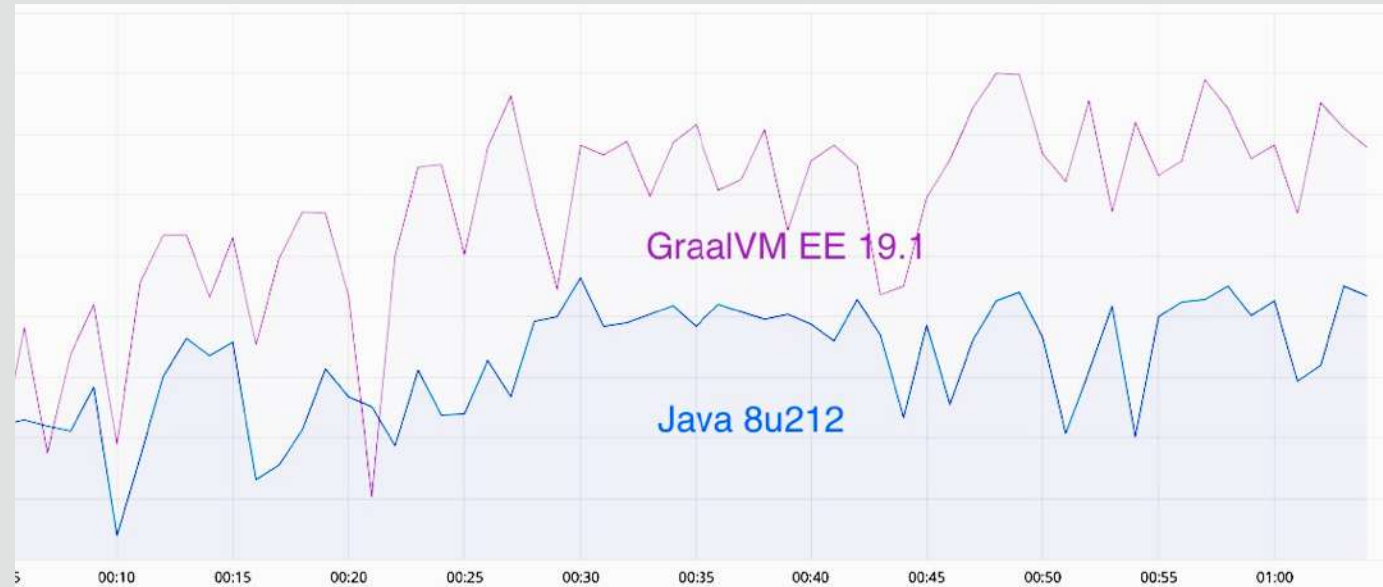
---

- 1 What's under the hood
- 2 Performance optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next

Twitter uses GraalVM compiler in production to run their Scala microservices



- Peak performance: +10%
- Garbage collection time: -25%
- Seamless migration



**ORACLE<sup>®</sup>**  
Cloud Infrastructure



The rich ecosystem of CUDA-X libraries is now available for GraalVM applications.

GPU kernels can be directly launched from GraalVM languages such as R, JavaScript, Scala and other JVM-based languages.



# Agenda

---

- 1 What's under the hood
- 2 Performance optimization
- 3 Fast startup for Java programs
- 4 Industry use cases
- 5 What's next for GraalVM

# What's next for GraalVM

---

- JDK-11 based builds;
- ARM64 and Windows support;
- Low-latency, high-throughput, and parallel GC for native images;
- Work with the community to support important libraries;
- New languages and platforms;
- Your choice – contribute!

## What's next for you

---

- Download:  
[graalvm.org/downloads](https://graalvm.org/downloads)
- Follow updates:  
[@GraalVM](https://twitter.com/GraalVM) / [#GraalVM](https://twitter.com/GraalVM)
- If you need help:
  - [gitter.im/graalvm](https://github.com/graalvm/graalvm)
  - [graalvm-users](https://twitter.com/graalvm-users)  
[@oss.oracle.com](https://oss.oracle.com)

# Thank you!

---

**Alina Yurenko / @alina\_yurenko**

GraalVM Developer Advocate  
Oracle Labs

