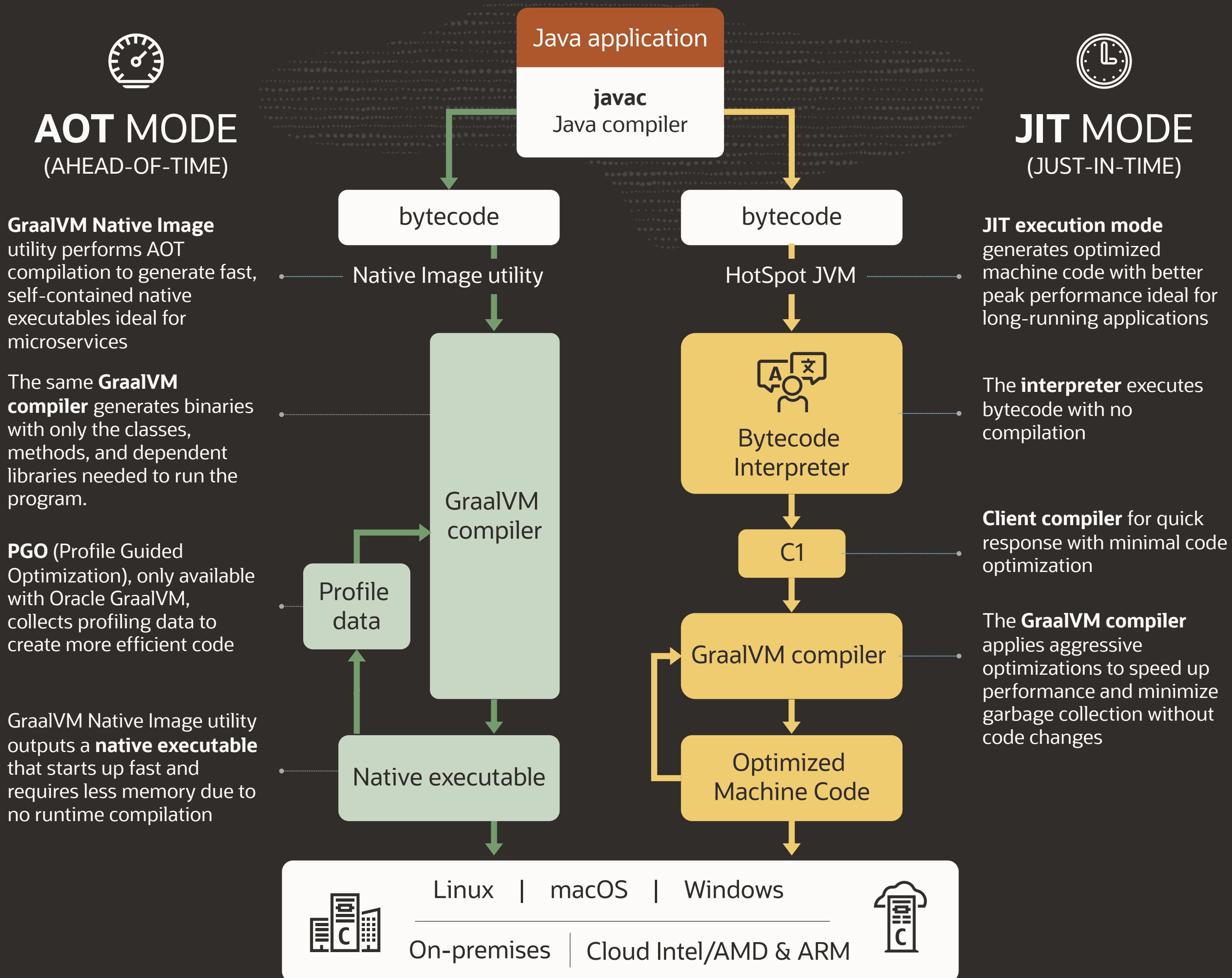Java™
ORACLE
+ ORACLE
GraalVM

Anatomy of Oracle GraalVM

Oracle GraalVM – a high-performance runtime that speeds up the performance of Java and JVM-based applications, lowers application latency, improves peak throughput by reducing garbage collection time, and comes with 24x7 Oracle support.
Supports JDK 17 and JDK 20

AOT MODE
(AHEAD-OF-TIME)

JIT MODE
(JUST-IN-TIME)

Java application

javac
Java compiler

bytecode

bytecode

GraalVM Native Image utility performs AOT compilation to generate fast, self-contained native executables ideal for microservices

Native Image utility

HotSpot JVM

JIT execution mode generates optimized machine code with better peak performance ideal for long-running applications

The same **GraalVM compiler** generates binaries with only the classes, methods, and dependent libraries needed to run the program.

GraalVM compiler

Bytecode Interpreter

The **interpreter** executes bytecode with no compilation

PGO (Profile Guided Optimization), only available with Oracle GraalVM, collects profiling data to create more efficient code

Profile data

C1

Client compiler for quick response with minimal code optimization

GraalVM compiler

The **GraalVM compiler** applies aggressive optimizations to speed up performance and minimize garbage collection without code changes

GraalVM Native Image utility outputs a **native executable** that starts up fast and requires less memory due to no runtime compilation

Native executable

Optimized Machine Code

Linux | macOS | Windows

On-premises | Cloud Intel/AMD & ARM

BENEFIT

| Up to **100x faster** helps scaling | Up to **5x smaller** memory easier to containerize | Up to **30% faster** throughput* | **24x7 support** for reliability | **Reduce cloud operations cost** or free up compute resources from faster performance and lower memory usage |

Learn more https://www.oracle.com/graalvm
Benchmark comparisons based on Oracle GraalVM release 22.0 vs JDK 17