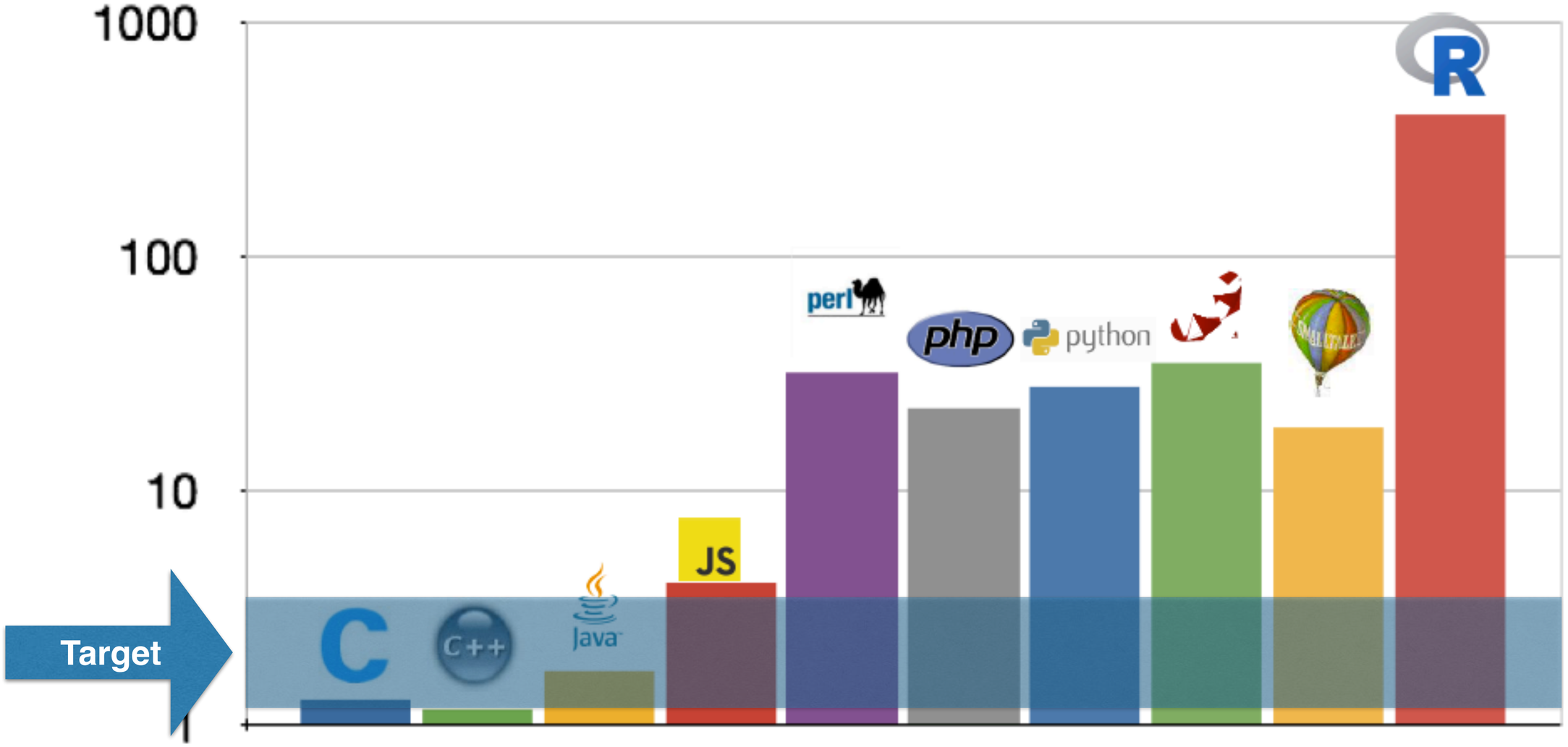
The quest to the language Graal: one JVM to rule them all

Elder Moraes

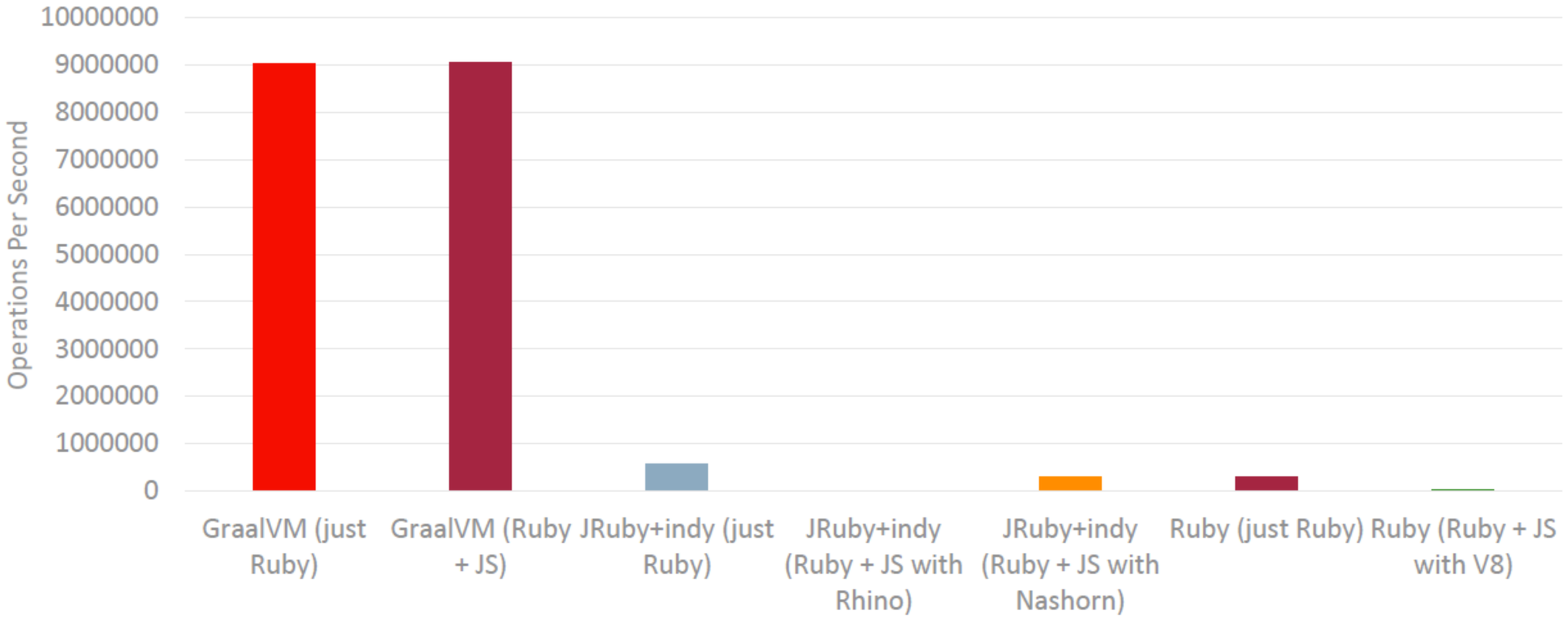
	Usually	Graal VM
PERFORMANCE	Only for languages with high Investments	High performance for all languages
INTEROPERABILITY	High cost for serialization	Zero overhead
TOOLING	Each language use its own tools (debugging, profiling, etc)	Shared tools for all languages

Performance

Languages Performance Benchmark



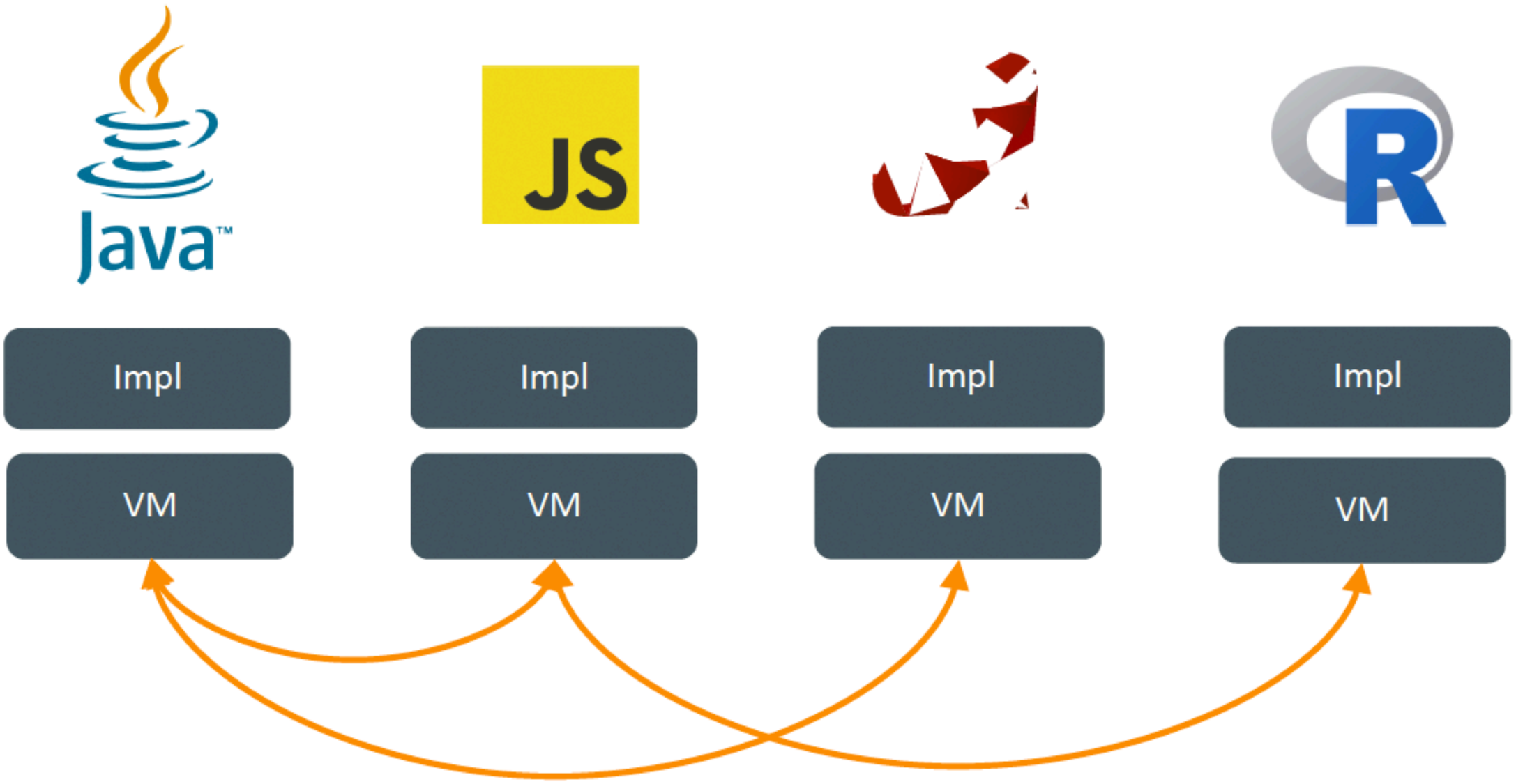
Math function in different environments

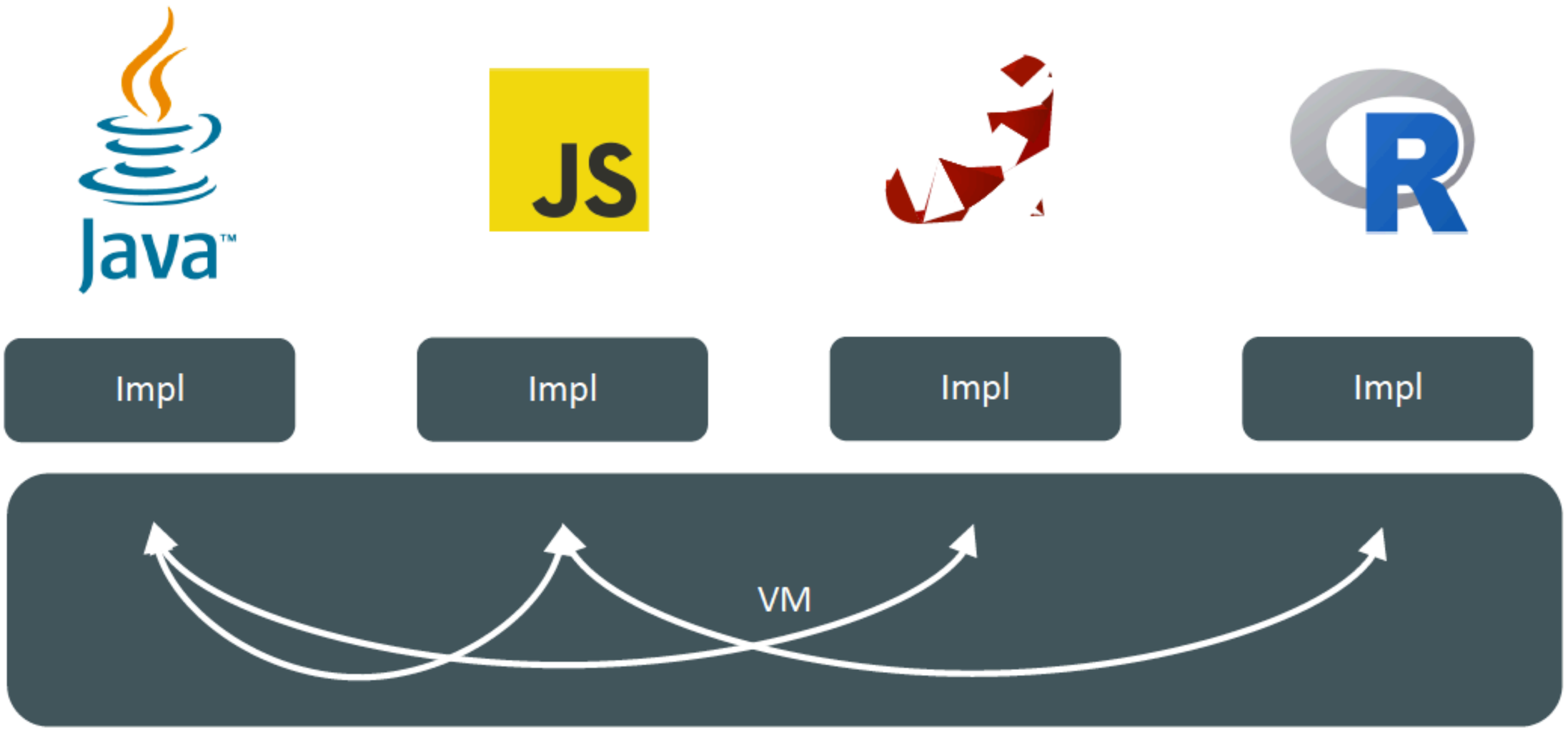


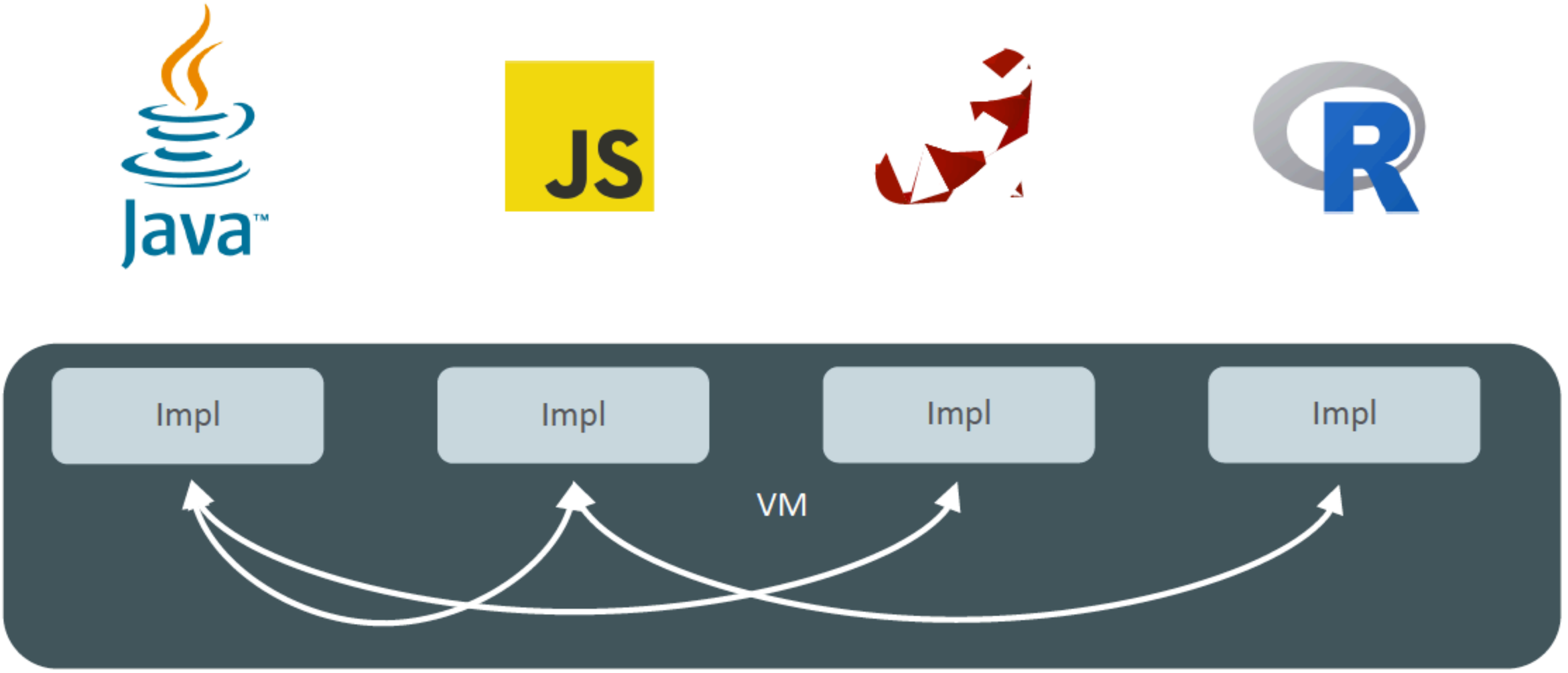
The secret: Graal compiler

- JIT (Just in Time) compiler
- AOT (Ahead of Time) compiler
- Written in Java - easier and faster to evolve (even more optimisation options available)
- Years and years of research and work

Interoperability

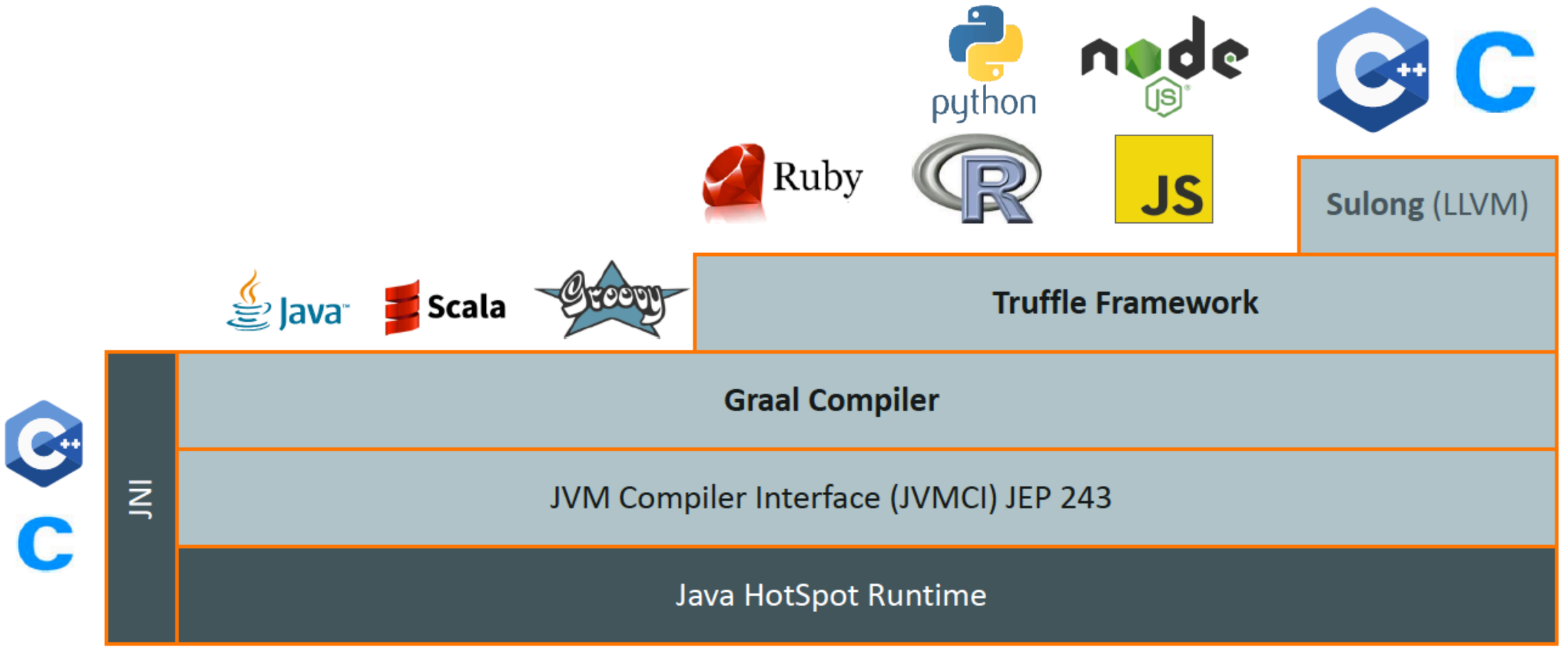

Java™
JS
R
Impl
Impl
Impl
Impl
VM
VM
VM
VM


Java™
JS
R
Impl
Impl
Impl
Impl
VM


Java™
JS
R
Impl
Impl
Impl
Impl
VM

Truffle: a language implementation framework

- Make possible to implement language interpreters
- Truffle uses GraalVM on your behalf to gives you a JIT compiler for your language
- So... today GraalVM is a JIT compiler for Java, JavaScript, Ruby, R and Python


python
node
C++
C
Ruby
R
JS
Sulong (LLVM)
Java
Scala
Groovy
Truffle Framework
C++
C
JNI
Graal Compiler
JVM Compiler Interface (JVMCI) JEP 243
Java HotSpot Runtime

Truffle has an API for polyglot code

```
1  color_rgb = Polyglot.eval('ruby', `
2    require 'color'
3    Color::RGB
4  `);
5
6  app.get('/css/:name', function (req, res) {
7    color = color_rgb.by_name(req.params.name).html()
8    res.send('<h1 style="color: ' + color + '" >' + color + '</h1>');
9  });
```

IMPORTANT

- The point is not to mix code from different languages, but:
  - Allows libs interchange among languages
  - Select the best (or preferred) language for your task

Tooling

Tooling

- If you are a Java developer, probably you are used to high quality tools
- If you are not… Truffle comes to the rescue
- Whatever tool you make using Truffle’s tool API, you’ll only write it once

Governance

Graal VM Advisory Board

Advisory Board Members

- Aleksei Voitylov, BellSoft. Works on bringing musl libc support to GraalVM and enhancing GraalVM on ARM platforms.
- Brian Clozel, Broadcom. Works on Spring Framework, Spring GraphQL, and Spring Boot.
- Bruno Caballero, Microdoc. Works on GraalVM integrations in the embedded space.
- Johan Vos, Gluon. Works on the JavaFX and mobile/embedded platform support for GraalVM Native Image.
- Kevin Menard, Shopify. Contributes to TruffleRuby – GraalVM Ruby implementation.
- Max Rydahl Andersen, Red Hat. Develops Quarkus – a Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best-of-breed Java libraries and standards.
- Michael Simons, Neo4j. Works on the Neo4j integration with GraalVM to support polyglot dynamic languages for user-defined procedures.
- Paul Hohensee, Amazon. Interested in GraalVM Community Edition, GraalVM Native Image, and AWS Lambda on GraalVM.
- Sandra Ahlgrimm, Microsoft. Interested in GraalVM Native Image, particularly for running Java in the cloud.
- San-Hong Li, Alibaba. Contributes to the project and share their experience with the community.
- Thomas Wuerthinger, Oracle. Vice President of GraalVM Development at Oracle.

Source: https://www.graalvm.org/community/advisory-board/

Thank you