JavaOne™

ORACLE®

FastR

R on the JVM with the FastR Runtime

Christian Humer
Senior Member of Technical Staff
VM Research Group,
Oracle Labs Switzerland
October 26, 2015

JavaOne
ORACLE

Safe Harbor Statement

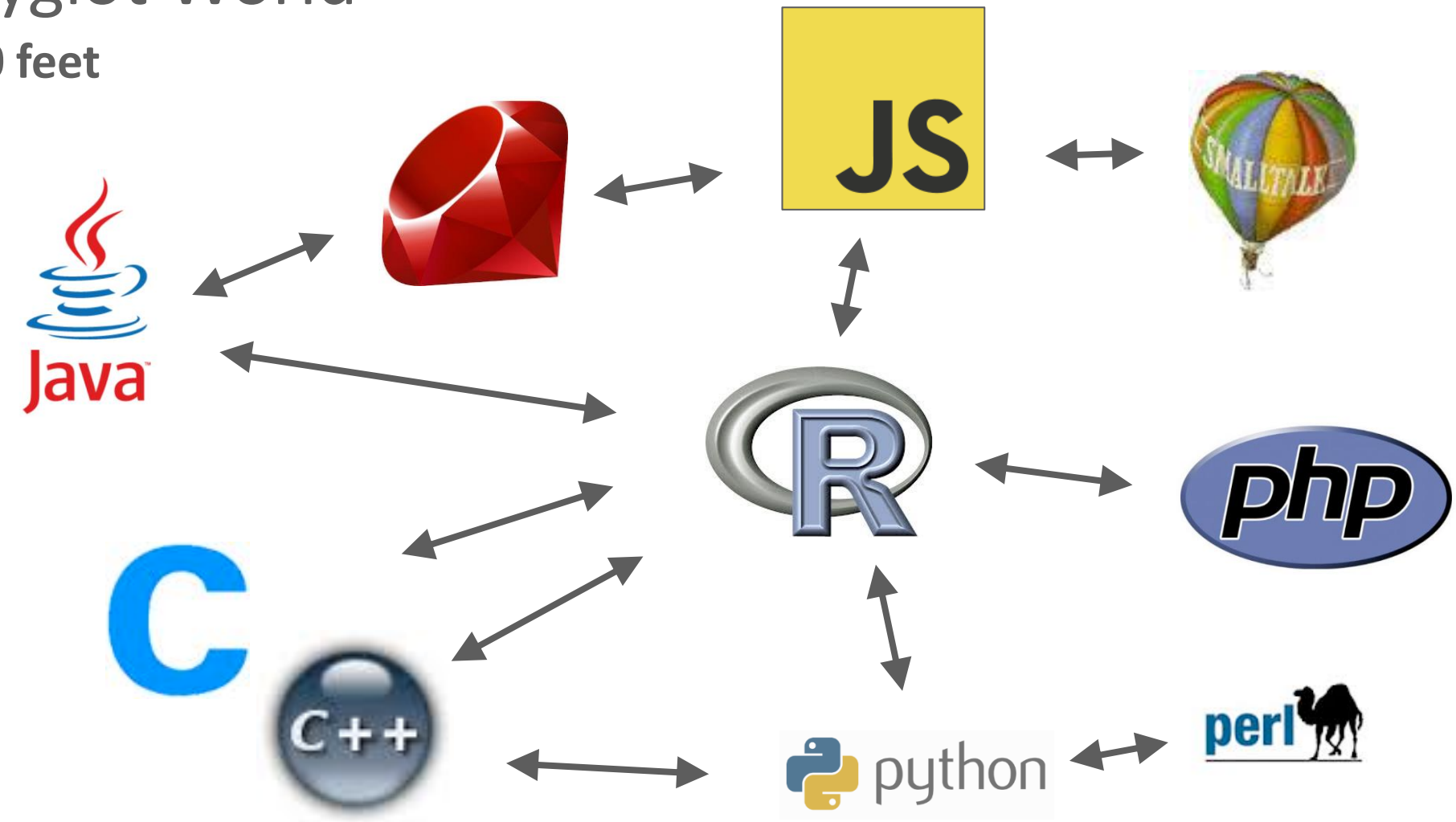
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

„The Polyglot World“

From 30,000 feet

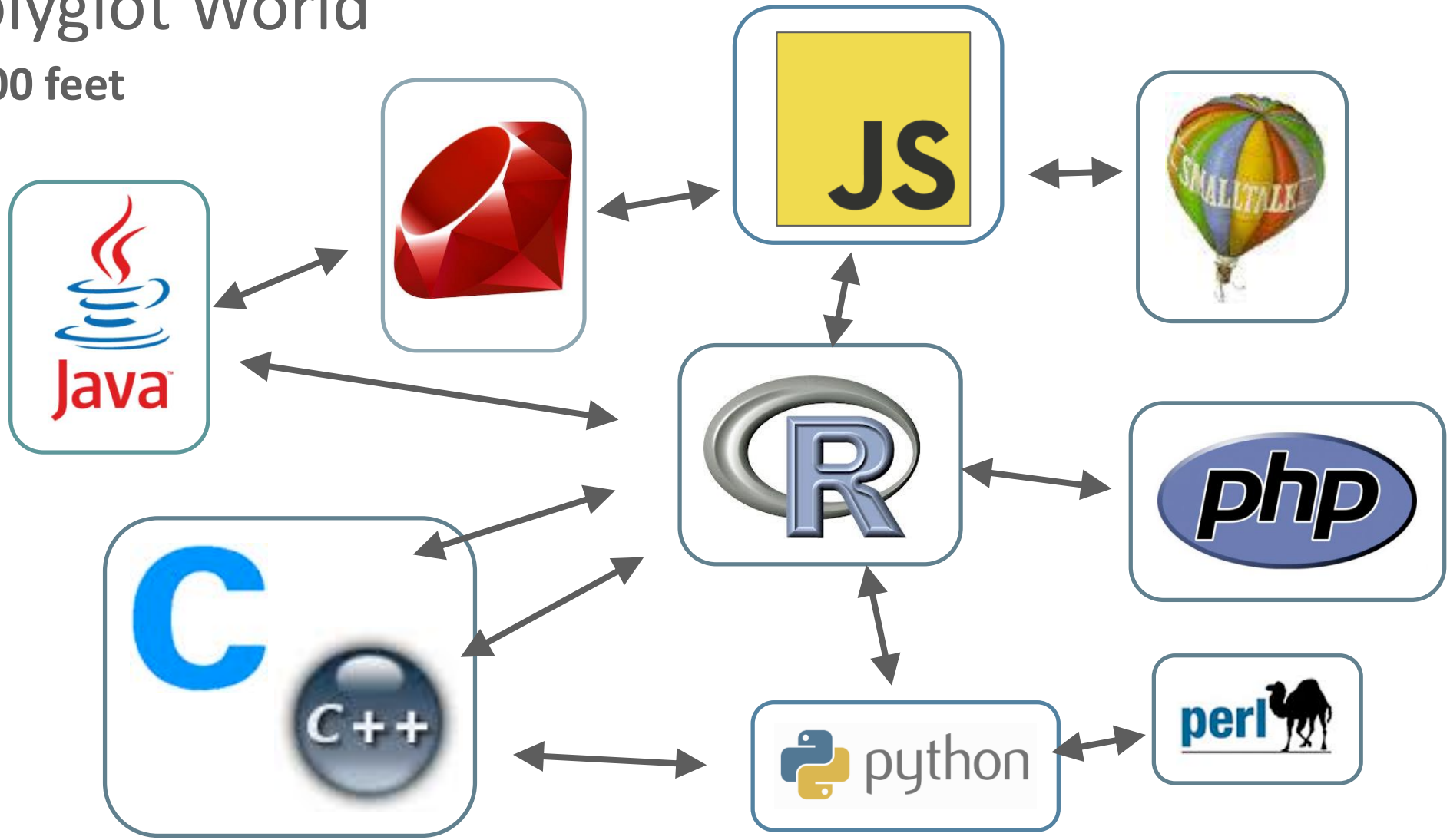
„The Polyglot World“

From 20,000 feet



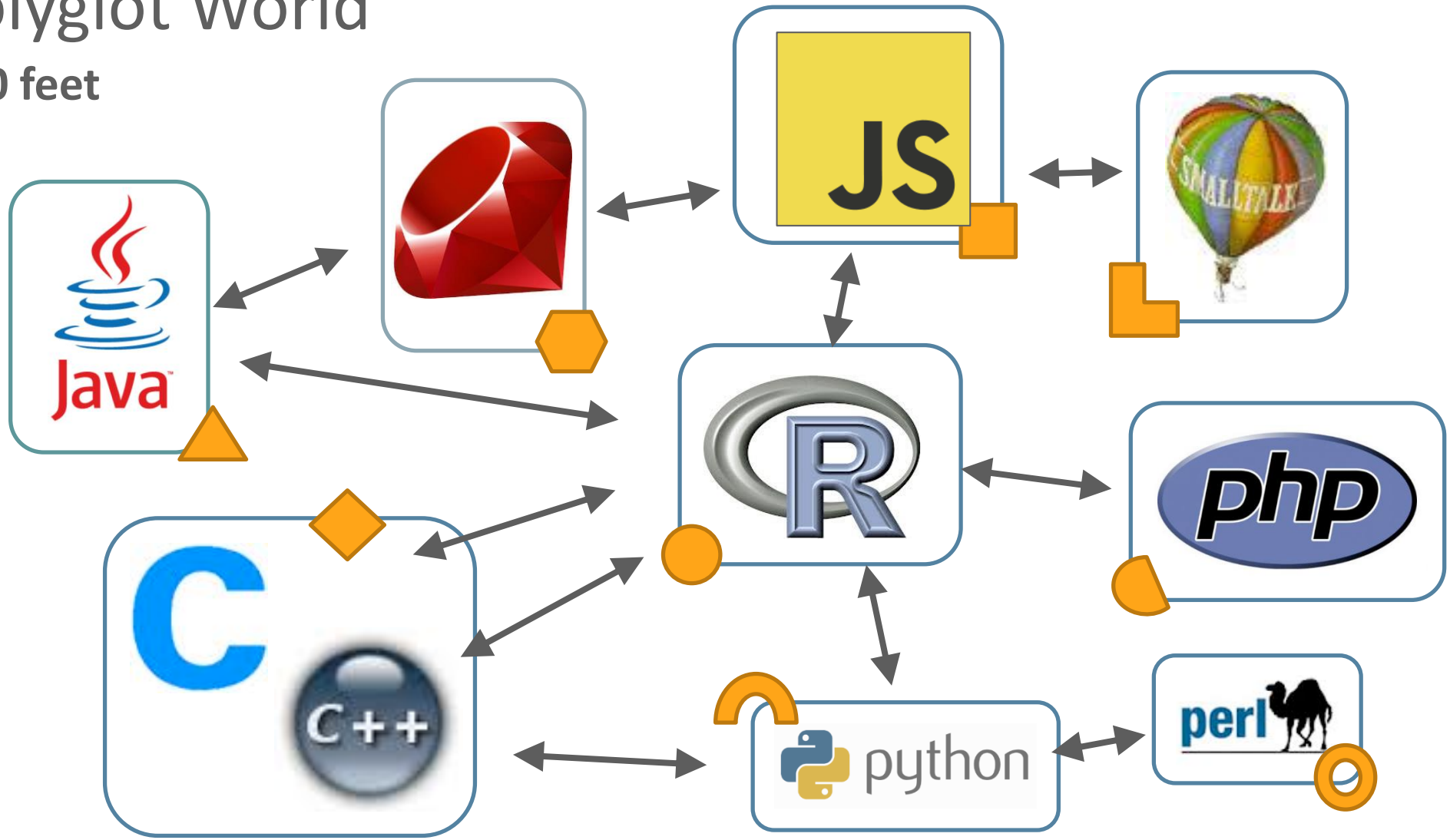
„The Polyglot World“

From 10,000 feet



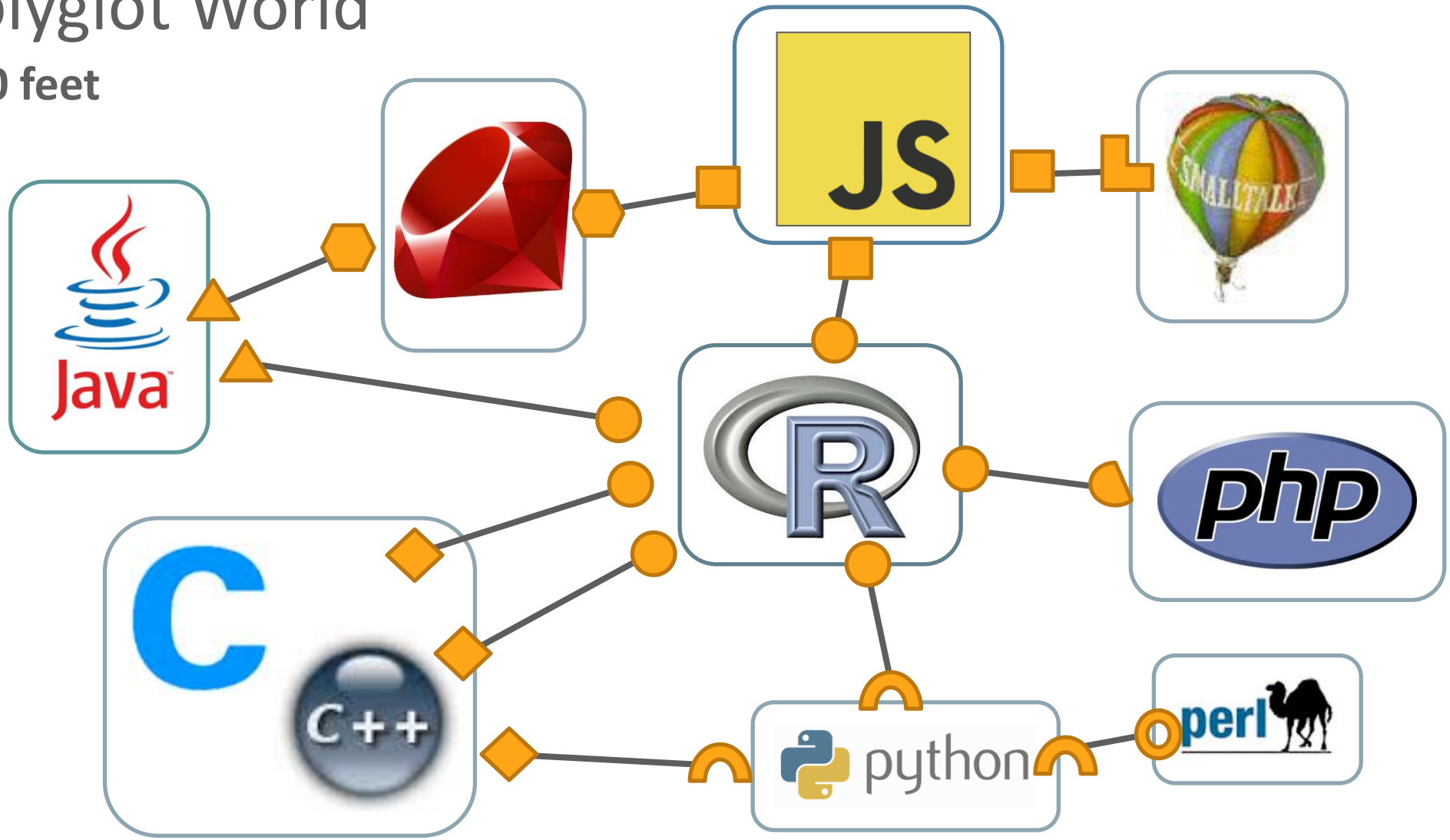
„The Polyglot World“

From 1,000 feet



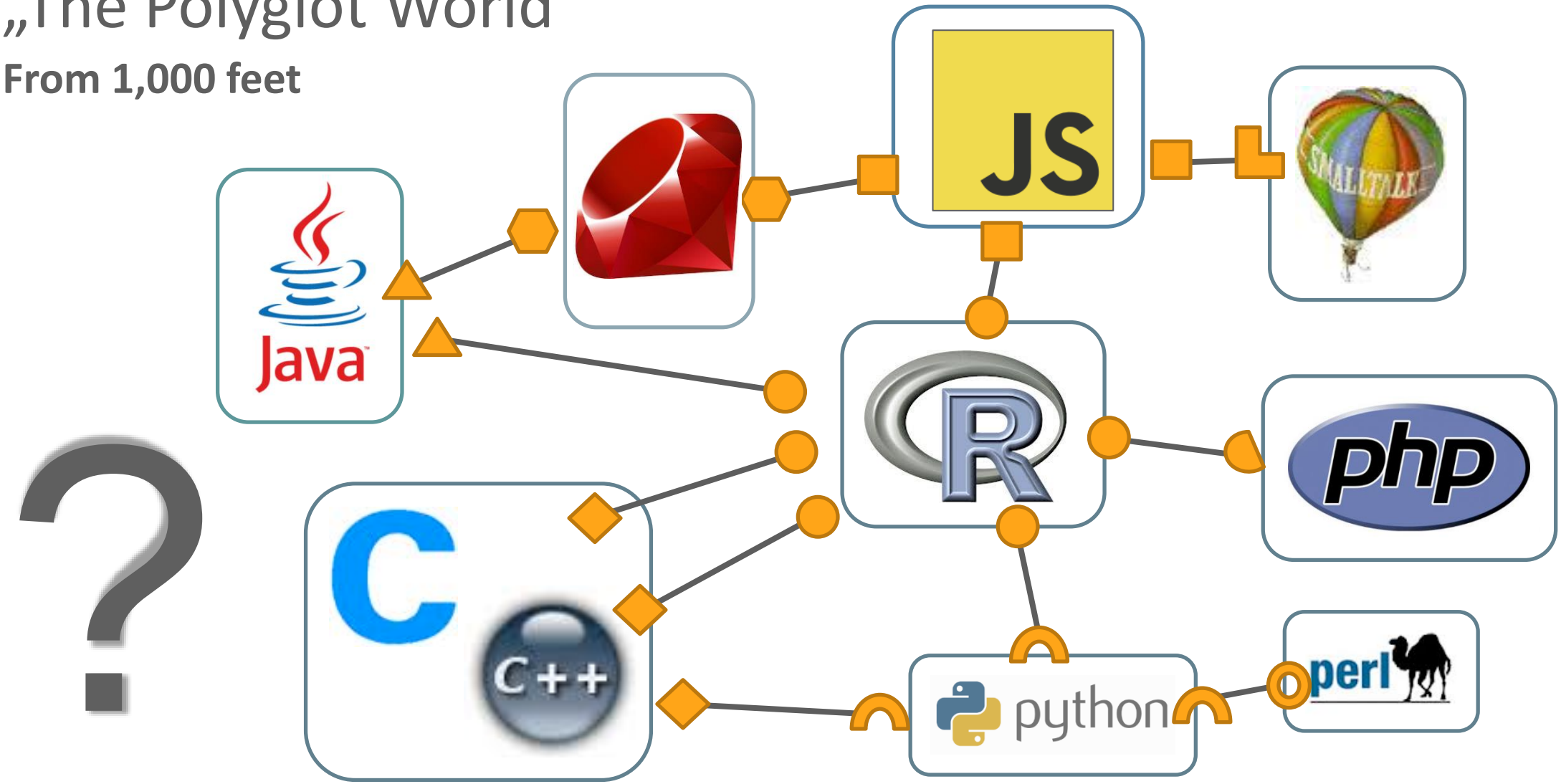
„The Polyglot World“

From 1,000 feet

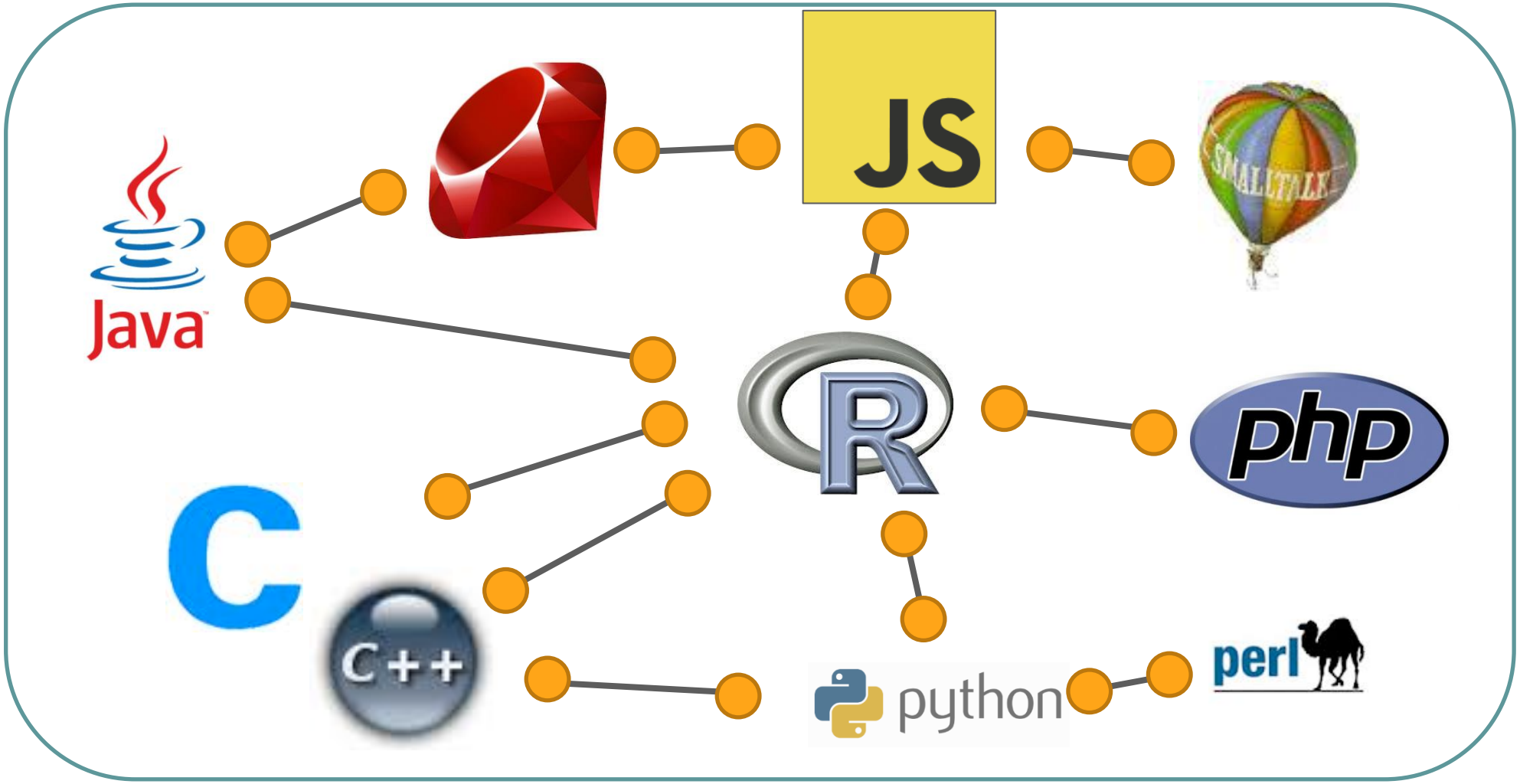


„The Polyglot World“

From 1,000 feet



„The Polyglot World“



The R Language for Statistical Computing

An important citizen of the polyglot world.

The R Language

- Powerful vector operations

```
v <- 3
[1] 3

v <- 1:4
[1] 1 2 3 4

v <- matrix(1:4, nrow=2, ncol=2)

v <- 1:8; dim(v) <- c(2,2,2)

v <- cos(v) * v / log(v)
```

The R Language

- Powerful vector operations

```
v[3]

v[3:6]

v[c(1,3,8)]

v[c(-3, -2)]

v[v > 4]

v[1, 2, ..]
```

The R Language

- Dynamic lookup for every operation (even "**+**", "**[**", etc.)

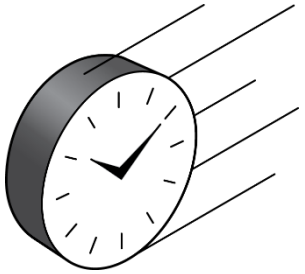
```
> `+` <- function(a, b) {a - b}
> 1 + 1
[1] 0

> `(` <- function(a) {a + 1}
> ((1))
[1] 3
```

Why R?

- Ease of use
  - Readily available
  - High level of abstraction
  - Lots of builtin functionality
  - Data types for statistics
  - Ease access to data in files
  - General-purpose language

- Interactive, agile system
  - Data exploration and manipulation
  - Graphics abilities
  - Integration with data representation technologies (web servers, typesetting systems, …)



- Large ecosystem
  - Large corpus of third-party packages (CRAN: 6000, bioconductor: 1000, …)
  - Covers many areas of data science: statistics, machine learning, bioinformatics, health sciences, geospatial, …

The R Language – code puzzle

```
a <- 0;
foo <- function(b) {
    a <<- 42
    b
}
bar <- function(b) {
    a <<- 0
    b + a
}
bar(foo(0))
=> [1] 42
```

- Lazy evaluation of arguments (“promises”) + side effects!

What's wrong with R?

- Centered around GNUR
  - Complex and asymmetrical language features
  - Lack of specification
  - Lack of tests for language
  - Tricky licensing situation causes lack of diversity
- Poor performance (except for vector operations)
  - Encourages expressing simple operations in complex ways (convoluted matrix operations, etc.)
  - Encourages unnecessary porting of R code to C/C++/Fortran/... for performance reasons
  - Single-threaded execution (high overhead for parallel execution)
- Monolithic, standalone runtime
  - Integration with other systems either limits functionality or incurs communication overhead
  - No tight integration with RDBMS (because of security and licensing)

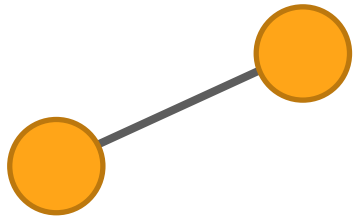
The FastR project

- An Oracle Labs Open Source R implementation based on the Graal technology stack.

  https://bitbucket.org/allr/fastr

Compatible with GNUR

High Performance „FastR“



A member of the polyglot world

Compatibility Status

- Lack of specification
  - Extensive test suite: 10k tests (80% pass)
    - Approx. 4k auto-generated (“testr” project, Roman Tsegelskyi, Purdue)
  - Expected test results not derived from specification, but from GNU-R execution
- Extensive set of builtins
  - Significant subset implemented (as required by benchmarks, test applications, etc.)
- Complex and “disruptive” language features
  - Promises, deparsing, error handling, visibility, …
  - Close to complete support - shows viability of the approach

Performance – Removing Abstractions

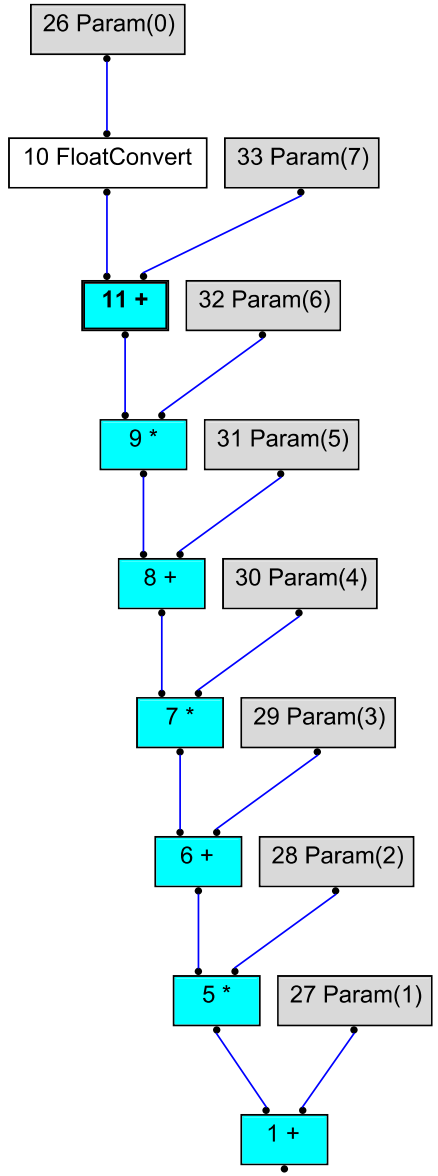
```
function(a)
  (((a+7)*3+5)*2+1)*8)+6
```

- Usually: arithmetic operations, but:
  - Do „+“ and „*“ resolve to the arithmetic builtins? (completely dynamic lookup)
  - Class of „a“?
  - Do „+.class“ and „*.class“ exist?
  - Type of „a“?
  - Any method can be added/removed/changed at the point where „a“ is evaluated (lazy promise)
  - Intermediate results needed?
- VM needs to remove all these abstractions

Performance – Removing Abstractions

```
function(a)
  ((((a+7)*3+5)*2+1)*8)+6
```

A long way



Performance Expectations

- No speedup on code calling C libraries:

```
qra <- qr(a, tol = 1e-7)
c <- qr.coef(qra, b)
```

- 2-5x speedup on vector code:

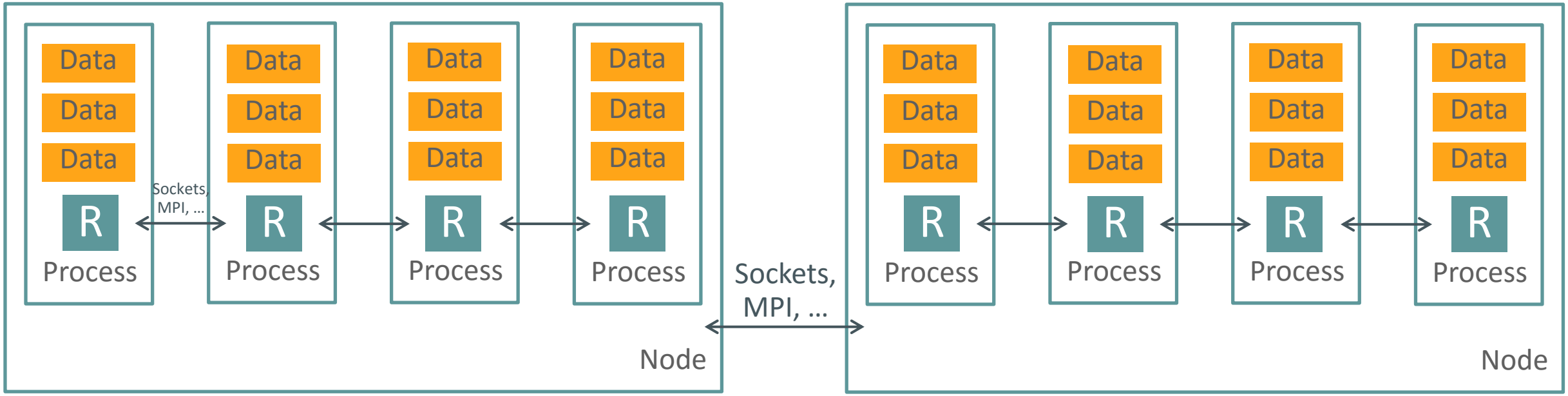
```
a <- floor(runif(10000)*1000)
b <- (phi^a - (-phi)^(-a))/sqrt(5)
```

- 50x speedup on loops and scalar code:

```
for (j in 1:input_size) {
  for (k in 1:input_size) {
    b[k,j] <- abs(j - k) + 1
  }
}
```

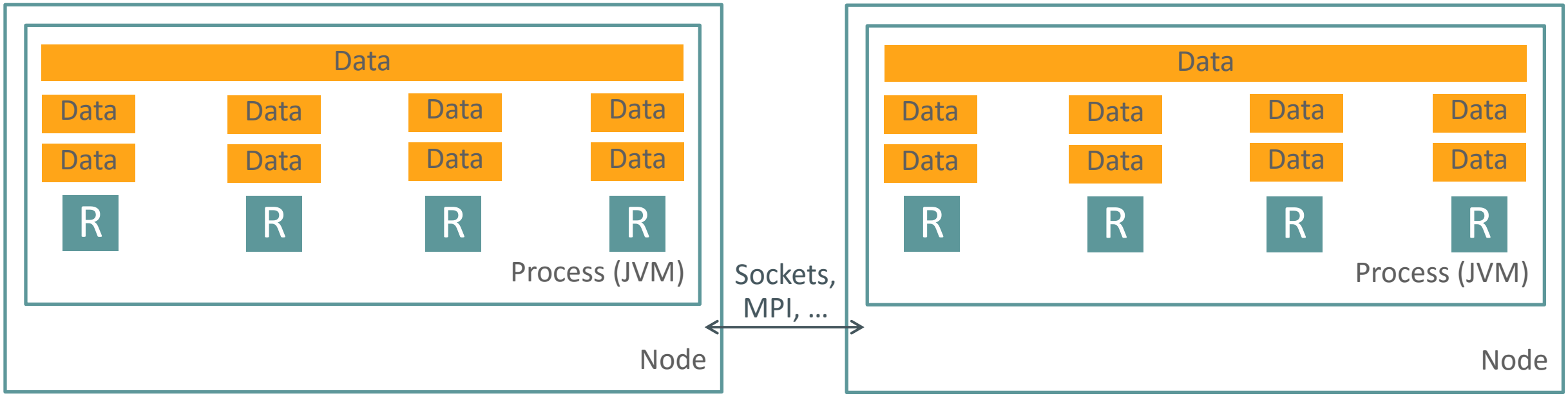
Performance - Parallelization

“Traditional” R-style parallelism



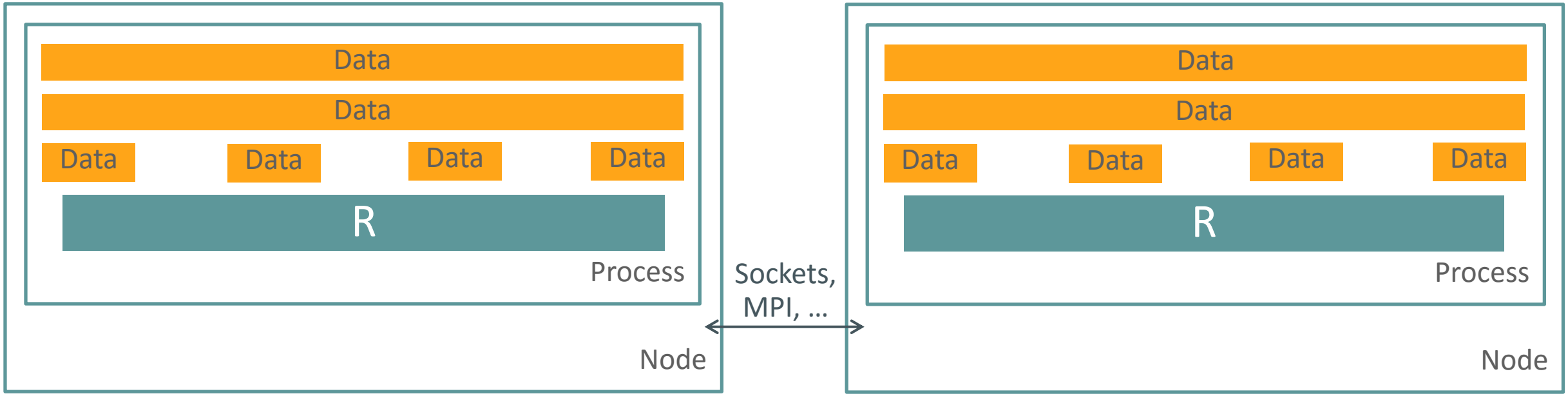
Performance - Parallelization

Simulated R-style parallelism

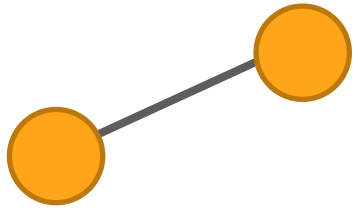


Performance - Parallelization

Custom parallelization APIs, automatic parallelization

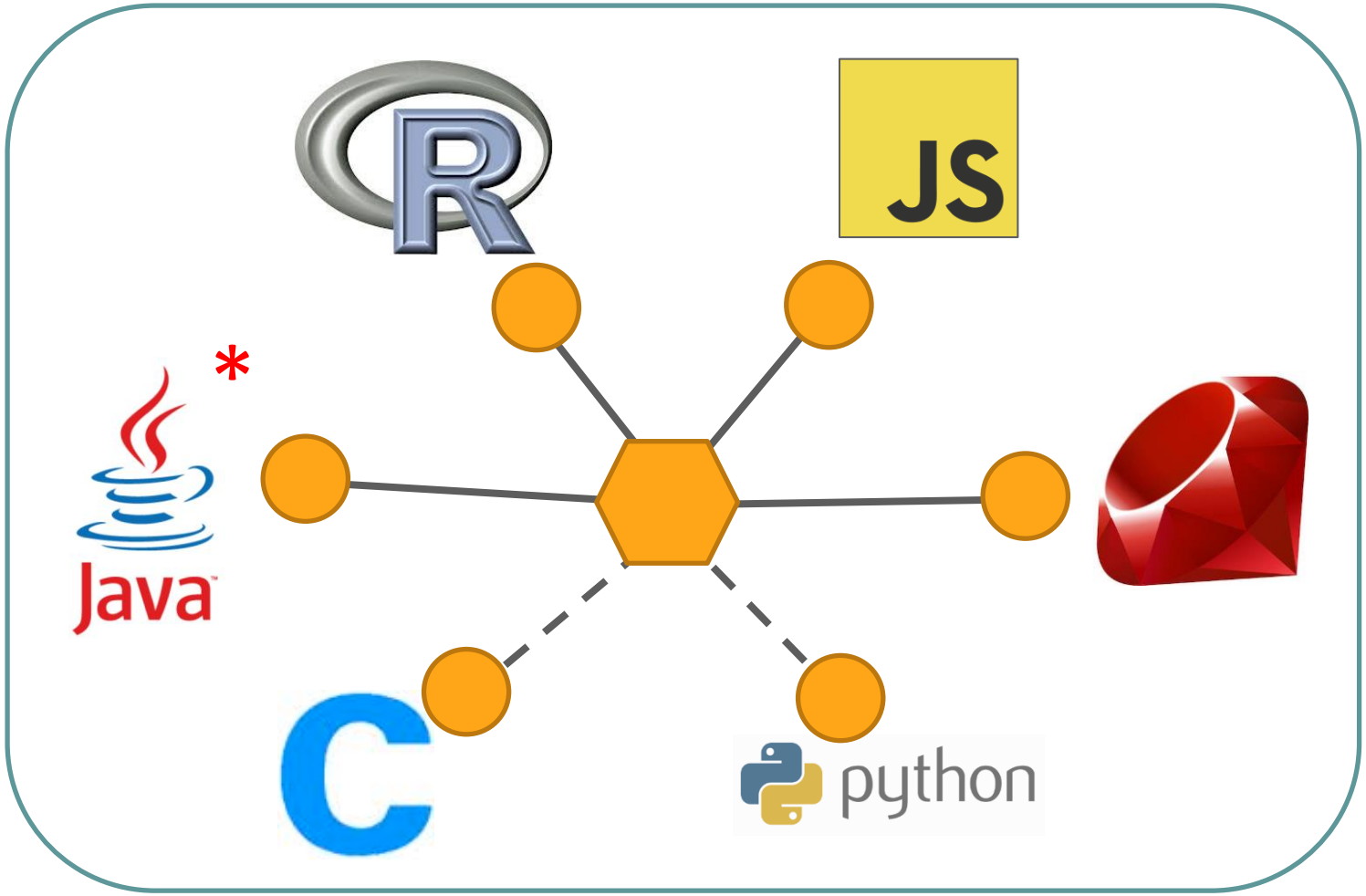


„The Polyglot World“

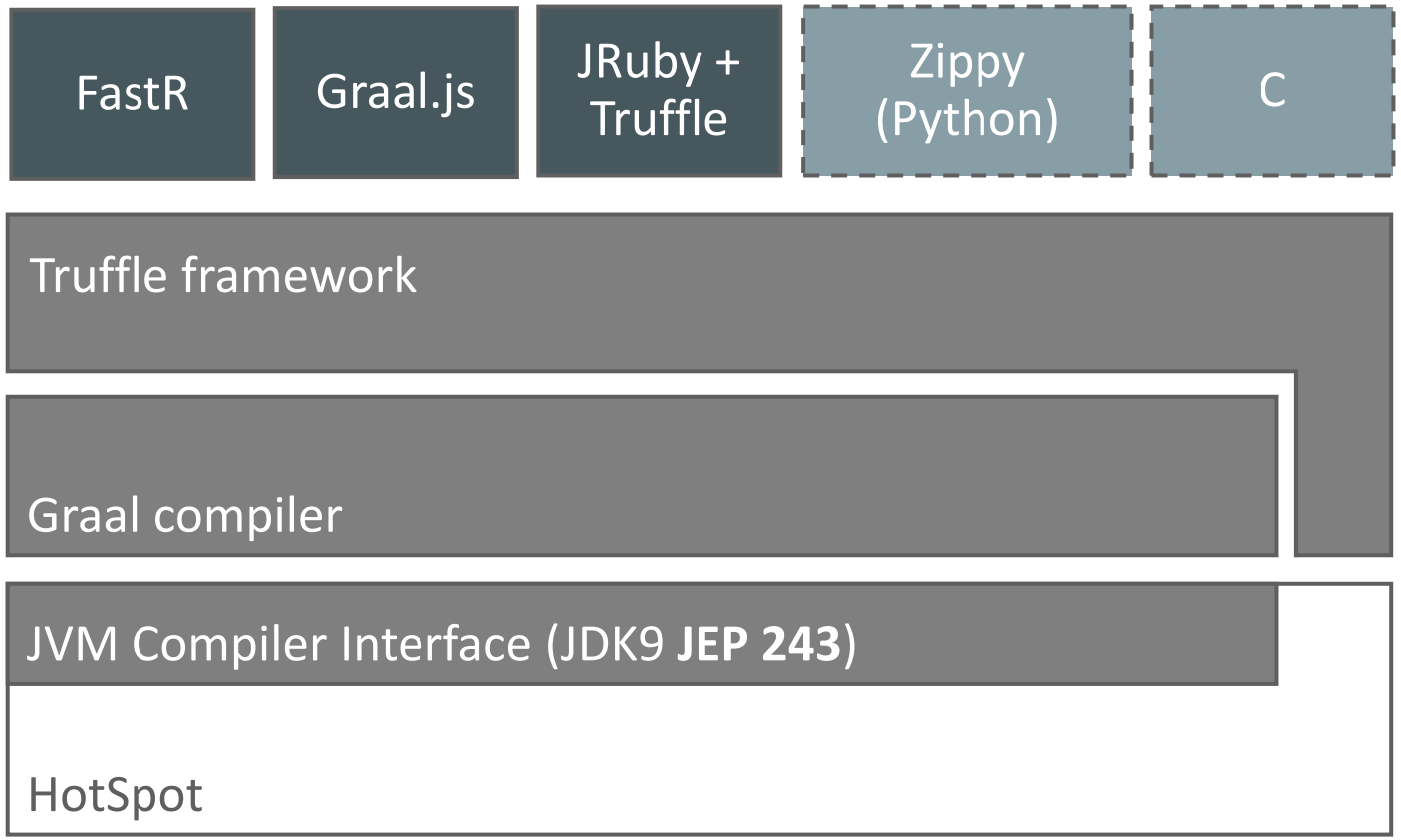


?

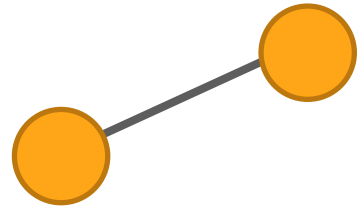
„The Polyglot Graal World“



The Polyglot Graal Technology Stack



FastR – as a member of the polyglot **Graal** world



- FastR is implemented as Graal language
  => AST Interpreter instead of dynamic bytecode generation.
- Graal enables Java-like performance for dynamic languages
- Optimizations across language boundaries
- Reuses customizable Truffle components (e.g. Object model)
- Tools built for Graal ecoystem can be used for all languages.

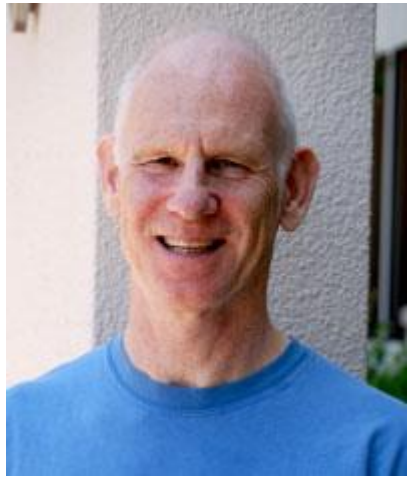
GraalVM - Early Access

- **Download it on Oracle Technology network!**
  - **http://goo.gl/Cptlpf**
  - Or goto OTN and search for “graalvm”
- JDK 8 with JVMCI + Graal + JavaScript + Ruby + FastR
- Disclaimer: FastR and Language Interopability are still experimental

Demo

The FastR Team

Lukas Stader



Mick Jordan

Adam Welc

Christian Humer

Acknowledgements

Oracle Labs
Danilo Ansaloni
Stefan Anzinger
Daniele Bonetta
Matthias Brantner
Laurent Daynès
Gilles Duboscq
Michael Haupt
Christian Humer
Mick Jordan
Peter Kessler
Hyunjin Lee
David Leibs
Kevin Menard
Tom Rodriguez
Roland Schatz
Chris Seaton
Doug Simon
Lukas Stadler

Michael Van De Vanter

Oracle Labs (continued)
Adam Welc
Till Westmann
Christian Wimmer
Christian Wirth
Paul Wögerer
Mario Wolczko
Andreas Wöß
Thomas Würthinger

Oracle Labs Interns
Shams Imam
Stephen Kell
Gero Leinemann
Julian Lettner
Gregor Richards
Robert Seilbeck
Rifat Shariyar

Oracle Labs Alumni
Erik Eckstein

Christos Kotselidis

JKU Linz
Prof. Hanspeter Mössenböck
Benoit Daloze
Josef Eisl
Thomas Feichtinger
Matthias Grimmer
Christian Häub
Josef Haider
Christian Hube
David Leopoltsederr
Manuel Rigger
Stefan Rumzucker
Bernhard Urban

University of Edinburgh
Christophe Dubach
Juan José Fumero Alfonso Ranjeet Singh
Toomas Remmelg

LaBRI

Floréal Morandat

University of California, Irvine
Prof. Michael Franz
Codrut Stancu
Gulfem Savrun Yeniceri
Wei Zhang

Purdue University
Prof. Jan Vitek
Tomas Kalibera
Romand Tsegelskyi
Prahlad Joshi
Petr Maj Lei Zhao

T. U. Dortmund
Prof. Peter Marwedel
Helena Kotthaus
Ingo Korb

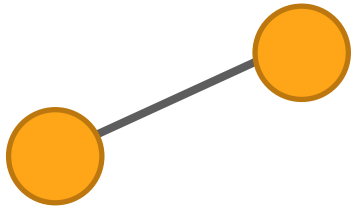
University of California, Davis
Prof. Duncan Temple Lang

Nicholas Ulle

More on FastR and Graal

- Visit us at the Geek Bar for more information
- Chris Seaton´s talk on Ruby on top of Graal.
  - Same Room Wednesday 15:00 – 16:00

- Download GraalVM @ OTN
  - http://goo.gl/Cptlpf
- We are interested in user stories/cases!
- Contact: christian.humer@oracle.com



JavaOne™

ORACLE®