

ONEK8s

ONEJAVA

ALL CLOUDS



OMG!

The Best Java Ever!

Erin Schnabel
Maker of Ridiculous Things at Red Hat
@ebullientworks

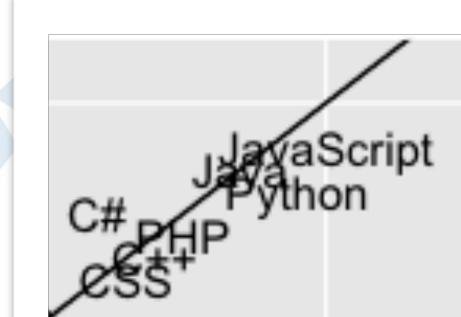
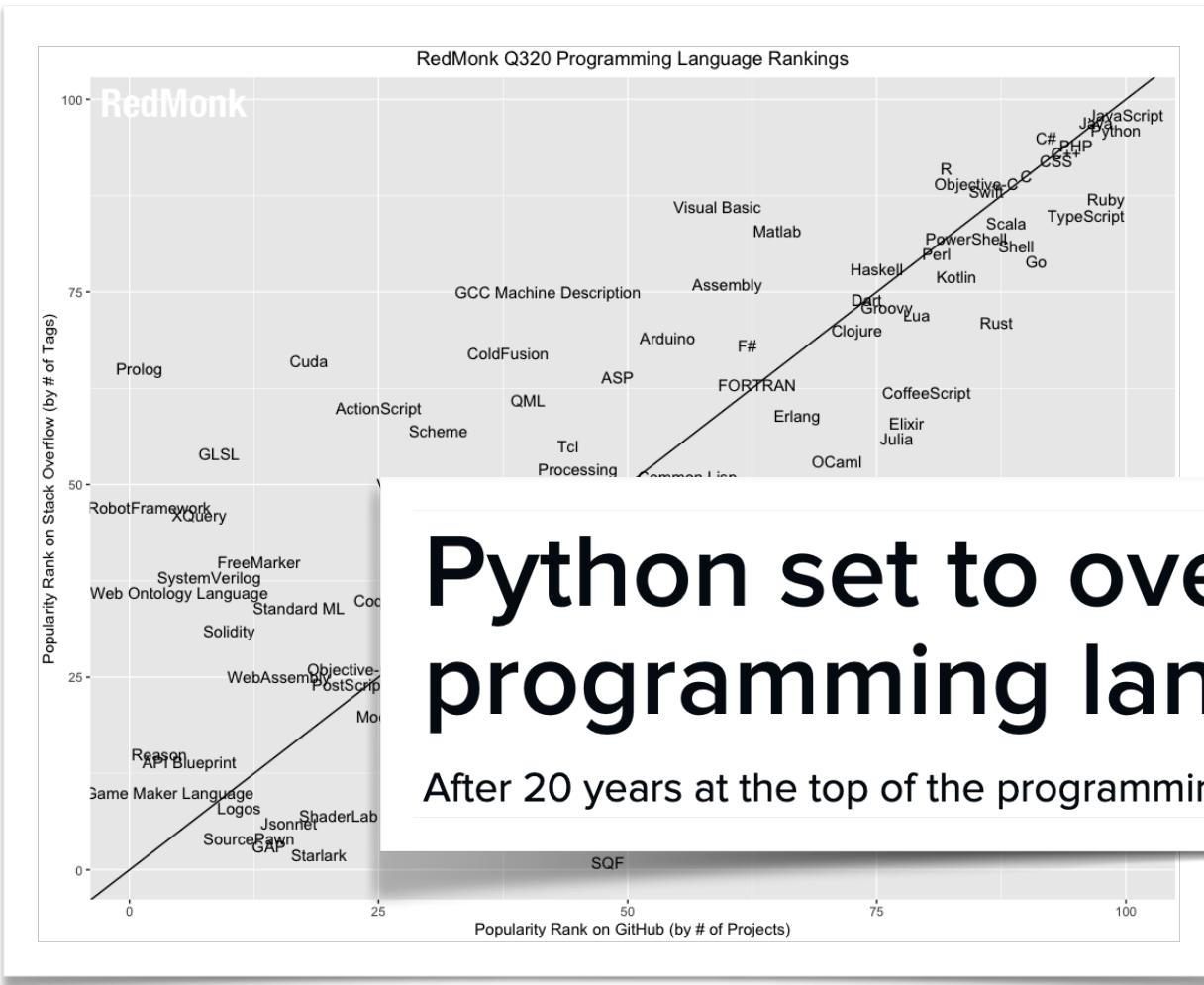
www.J4K.io

Cats, Dogs, and Java

Java is fun right now.
Let me show the ways...

Also: Generated Titles^[1].
May try again...

The Truth Is: You Are Not The Only Person Concerned About Java



Python set to overtake Java in latest programming language rankings

After 20 years at the top of the programming-language tree, Java looks set to be overtaken by Python.

ZDNet: Programming language popularity [2]
The RedMonk Programming Language Rankings: June 2020 [3]

The Death Of Java And How To Avoid It

Lambdas

"It was pretty clear at the time that Java was falling behind, and this was our big chance to prove to the world that it was possible for Java to remain relevant and fun to program in; that we could continue to teach this old dog some fancy new tricks"

--Brian Goetz [\[4\]](#)

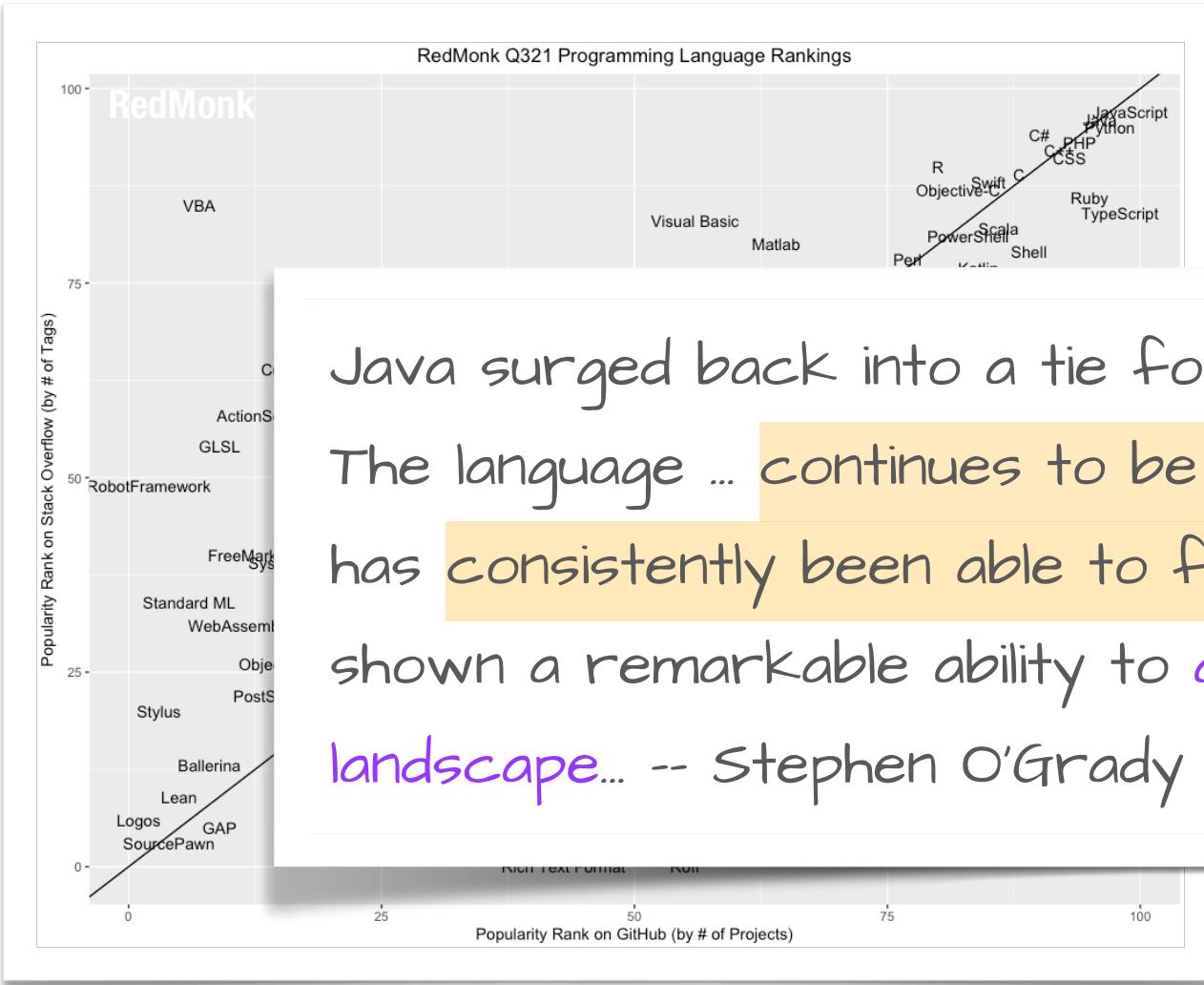
How To Turn Java Into Success



"... Java programming language need[s] an ongoing series of innovative, must-have, and 'delightful' features that make the language even more user friendly and cloud capable."

-- Report by Omdia as Java turned 25 [\[5.6\]](#)

Proof That Java Really Works



Java surged back into a tie for second place with Python. ...

The language ... continues to be a workhorse, and importantly ..

has consistently been able to find new work to do. ... it's

shown a remarkable ability to adapt to a rapidly changing
landscape... -- Stephen O'Grady [7]

Why Java Succeeds

Focus: Make Java more *extensible*

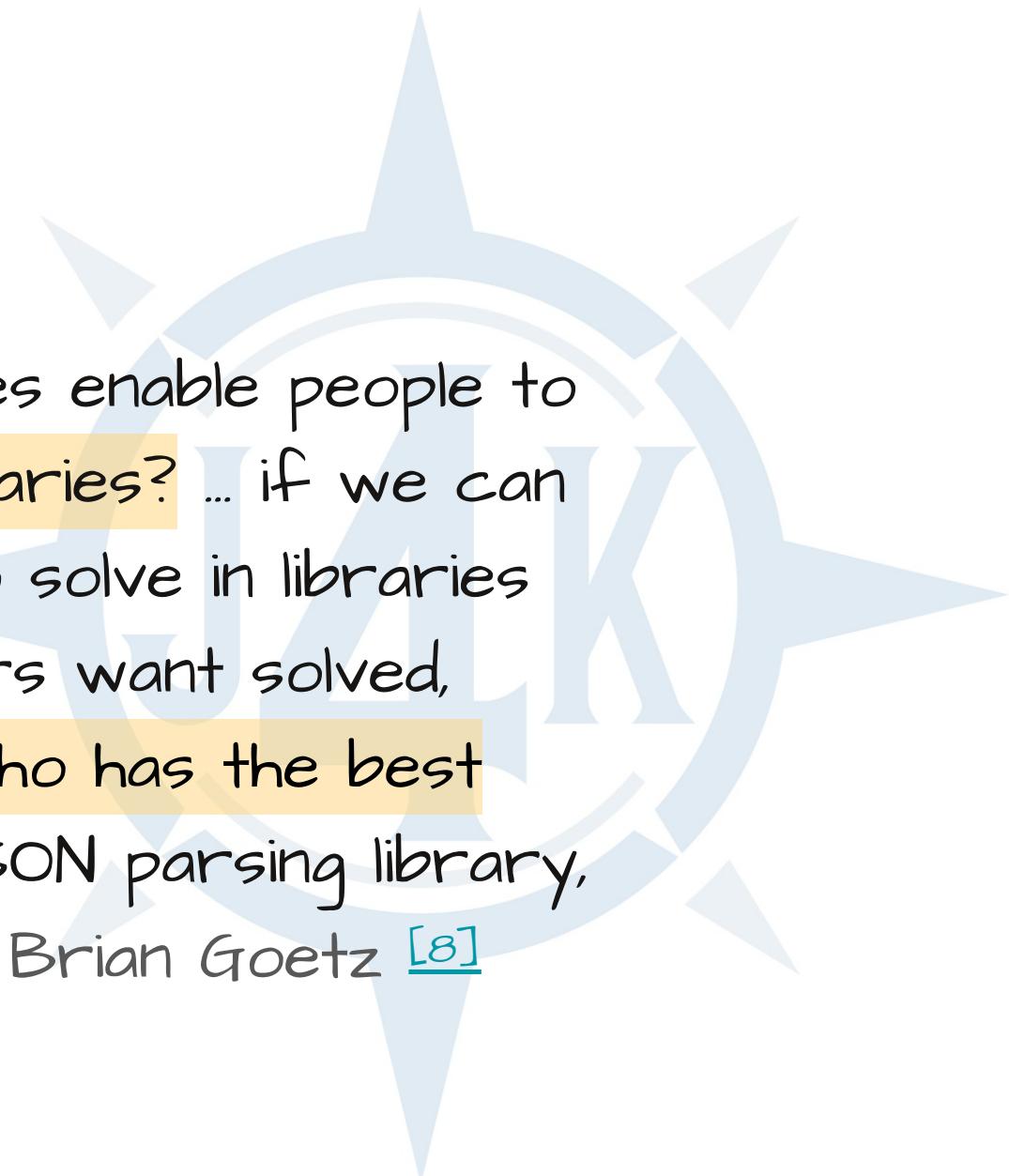
"The principle of doing, in[side] the platform,
what just absolutely cannot be done outside it"

-- Mark Reinhold [\[8\]](#)

"I think the thing that has kept us going ...
is the relentless commitment to compatibility."

-- Brian Goetz [\[4\]](#)

The Best Way to Java



How can language features enable people to write more powerful libraries? ... if we can enable the ecosystem to solve in libraries the problem developers want solved, and to compete for who has the best TensorFlow library or JSON parsing library, then everyone wins." -- Brian Goetz [\[8\]](#)

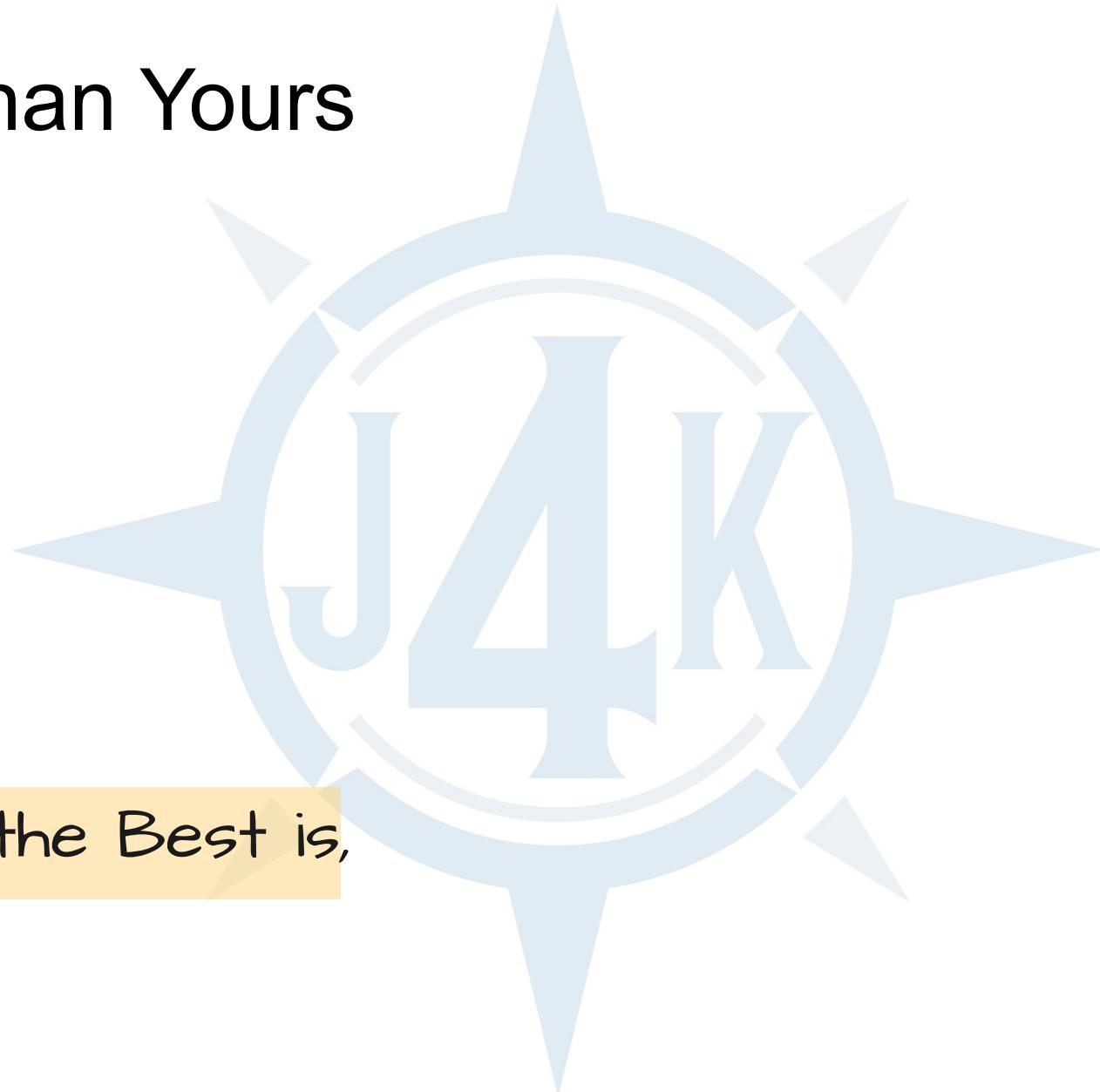
Why My Java Is Better Than Yours

It was all a set-up!

We LOVE to compare!

Benchmarks are our jam!

But do we really know what the Best is,
or how to measure it?



JAVA Smackdown!

Performance of Native Images

Matt Raible @mraible · Sep 16
Unknown

I updated my native Java framework comparison to use Micronaut 3.0.1, Quarkus 2.2.2, and Spring Boot 2.5.4.

Quarkus still wins the "startup time" metric, but Spring Boot wins when comparing memory used.

👉 developer.okta.com/blog/2020/09/16/native-image-comparison

#micronaut #quarkus #springboot

Table 2. Native Java memory used in me

Framework	Executor
Micronaut	app
Quarkus	runner
Spring Boot	demo

Used with permission [\[9,10\]](#)
Thanks, Matt! ❤️

Matt Raible @mraible · Sep 16
Unknown

Doh! I was not measuring memory usage fairly. I changed the numbers to use a macOS binary (`./mvnw package -Pnative`) when calculating memory usage for Spring Boot.

Now, Quarkus wins startup and memory metrics!

Thanks for pointing this out [@maxandersen](#).

No More Mistakes With Java [Benchmarking]

We all want to look good...

- Make sure you're comparing apples to apples
 - Docker view != Platform view
 - Lazy vs. Eager initialization
- Show your work!
 - Be transparent about your methods[\[12\]](#)



There are so many choices for how and what... Your why is interesting! Share it!

I Don't Want To Spend This Much ~~Time~~ On Java. How About You?

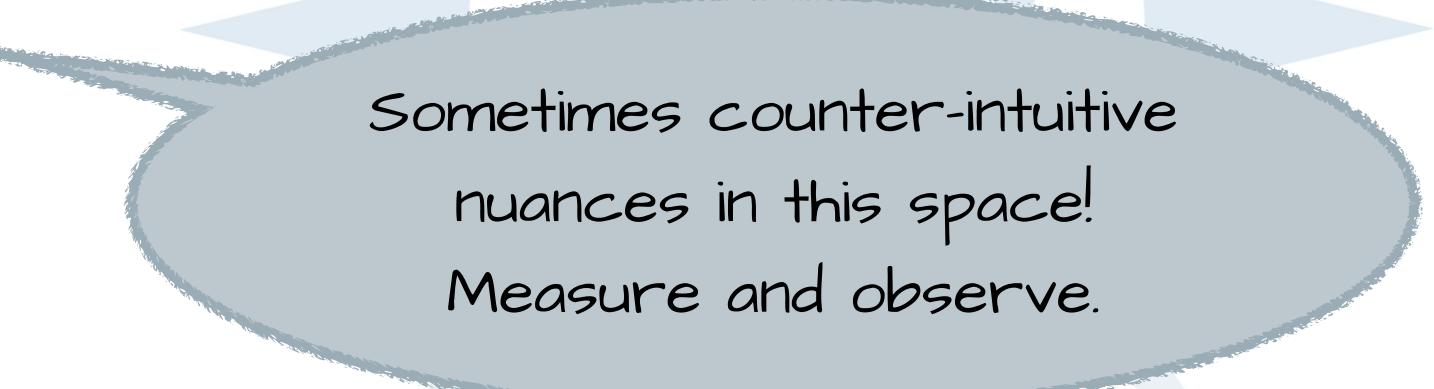
A different lens for thinking about performance:

How much does it cost?

Process longevity

Density

Memory/Resource allocation



Sometimes counter-intuitive
nuances in this space!
Measure and observe.

Boost Your Java With These Tips

JVM or Native mode?

- **Fast startup**
 - Images: Caching/platform optimizations
- **Resource utilization**
 - Density / Throughput: horizontal vs. vertical scale
 - GC algorithms: long-run/big heap vs. short-run/small heap



Reactive Java The Right(?) Way

Looming on the horizon...

Project Loom will kill reactive programming

-- Brian Goetz [\[15\]](#)

"Reactive programming sort of breaks everything^[*]..."

Project Loom ... solve[s] the same problem in a way that's harmonious with the design of the Java platform and its tools."

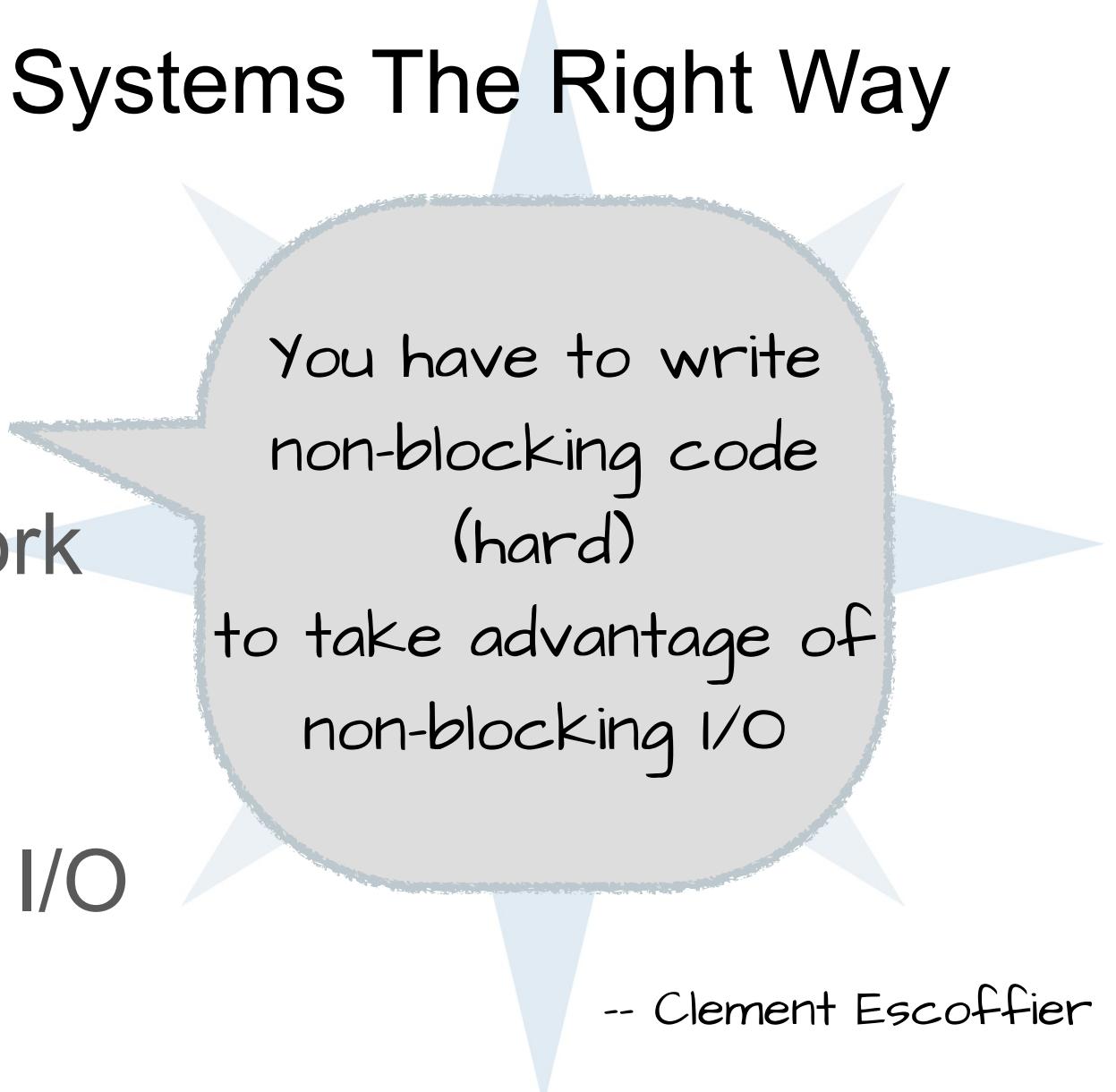
-- Ron Pressler [\[8\]](#)

* -- myriad places that assume context is tied to the state of a thread

~~Reactive Java~~ Distributed Systems The Right Way

LAYERS:

1. Your (messy) code
2. Your favorite Framework
3. Netty (most likely)
4. OS-level non-blocking I/O



You have to write
non-blocking code
(hard)
to take advantage of
non-blocking I/O

-- Clement Escoffier

Don't Get Too Excited. You May Not Be Done With Reactive Java

"My guess is that when Loom arrives, many developers will prefer getting better throughput by using virtual threads, ... It's also possible that ... libraries might ... find ways to ... reduce that mismatch ..."

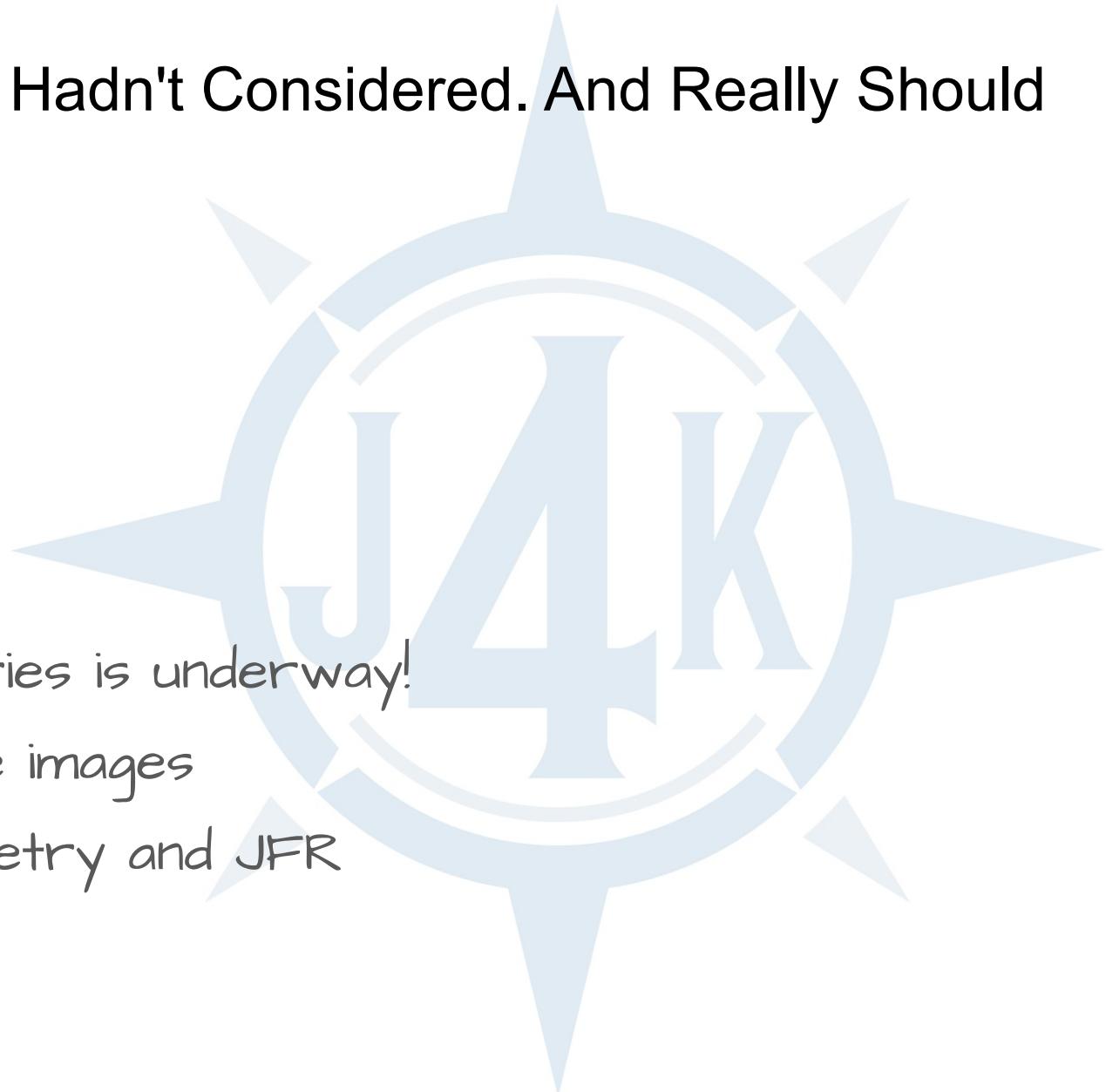
-- Ron Pressler [\[8\]](#)

"Loom will have an impact on reactive programming. Reactive programming will be used where it excels instead of being a solution for every problem. Loom will help developers follow reactive principles at the architecture level."

-- Clement Escoffier [\[13\]](#)

The Stuff About JAVA You Probably Hadn't Considered. And Really Should

- **Java Image creation**
 - Buildah, Docker, Jib, Podman
 - Fabric8, Cloud Native Buildpacks
- **OpenTelemetry**
 - Integration with well-known libraries is underway!
 - Instrumentation agents and native images
 - intersection between OpenTelemetry and JFR



Are You Actually Doing Enough Java?

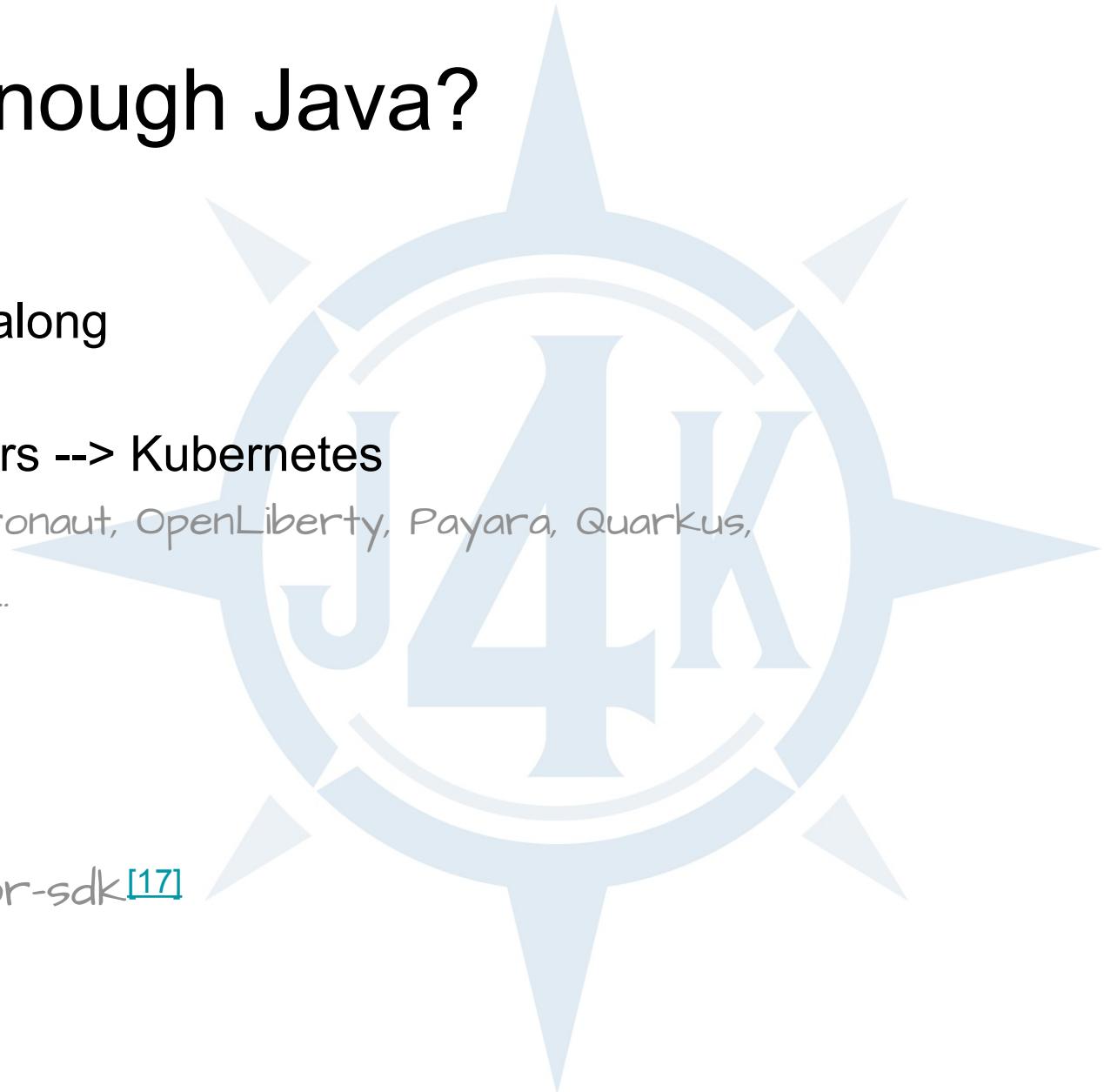
- UI Desktop application
- Traditional app servers ... still chugging along
- Self-contained Applications --> Containers --> Kubernetes

Dropwizard, Helidon, Javalin, KumuluzEE, Micronaut, OpenLiberty, Payara, Quarkus,
Spring/Spring Boot, TomEE, Wildfly, Vert.x, ...

- Native images:
 - Command-Line applications: picocli

Kubernetes Operators: [java-operator-sdk](#)^[17]

Service Mesh Filters?!



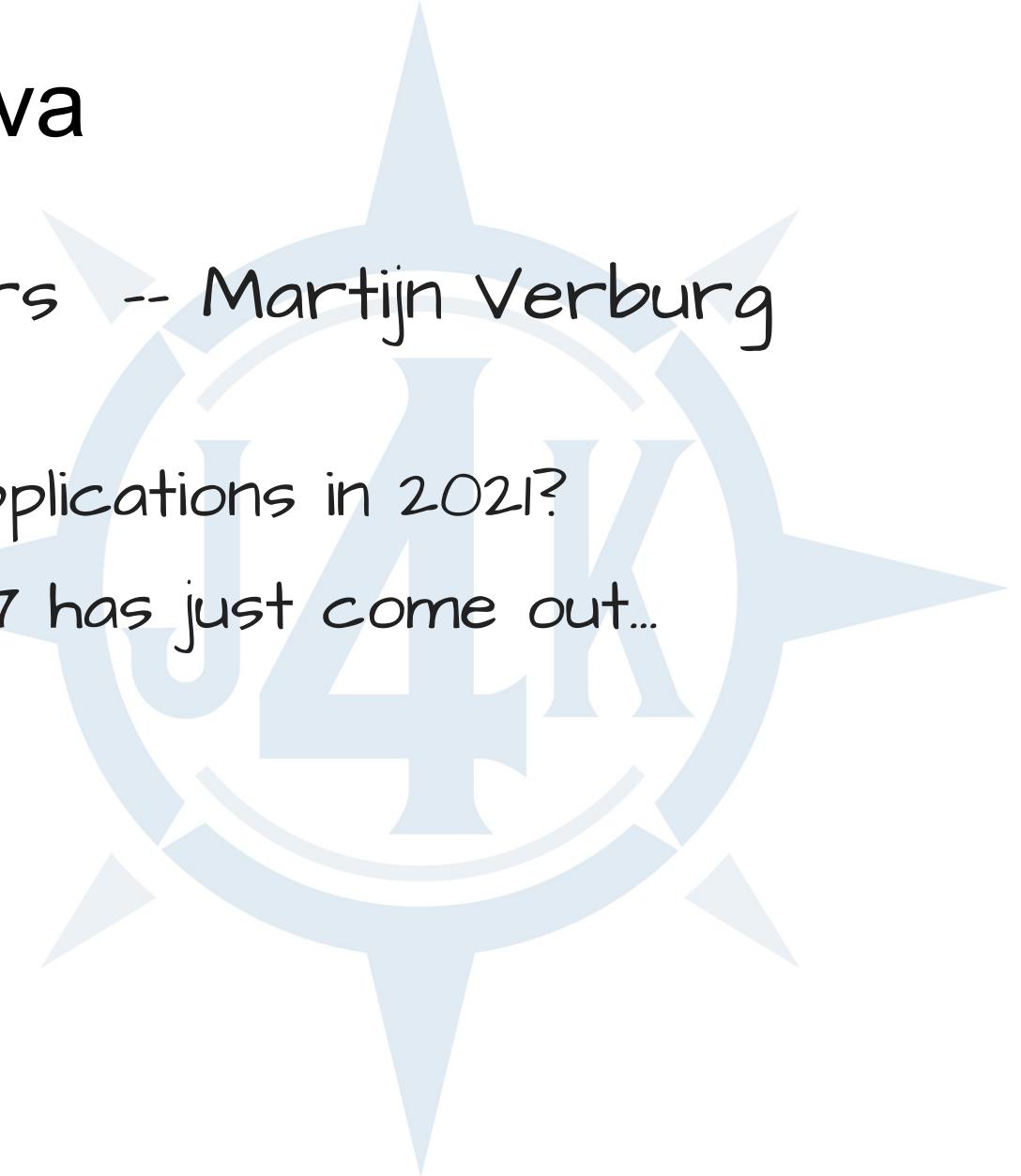
10 Ways To Reinvent Your Java

Modern Java for Cloud Developers -- Martijn Verburg

Java for Cloud Native applications in 2021?

You better believe it! Java 17 has just come out...

I love the enthusiasm.
I feel the same way!



Java: One Question You Don't Want To Ask Anymore

Which build tool

But seriously, try [JBANG!](#)



ONEK8s

ONEJAVA

ALL CLOUDS



Thank you



@j4kio



<https://www.linkedin.com/company/j4k/>



<https://www.facebook.com/j4kio>



www.J4K.io

References

1. Title Generator: <https://www.title-generator.com/>
2. <https://www.zdnet.com/article/python-set-to-overtake-java-in-latest-programming-language-rankings/>
3. <https://redmonk.com/sogrady/2020/07/27/language-rankings-6-20/>
4. <https://freecontent.manning.com/interview-with-brian-goetz/>
5. <https://www.zdnet.com/article/oracles-java-15-new-features-aim-to-keep-millions-away-from-languages-like-rust-kotlin/>
6. <https://www.oracle.com/a/ocom/docs/corporate/analystrelations/omdia-java-turns-25.pdf>
7. <https://redmonk.com/sogrady/2021/08/05/language-rankings-6-21/>
8. <https://blogs.oracle.com/javamagazine/post/what-are-they-buildingand-why-6-questions-for-the-top-java-architects>
9. <https://twitter.com/mraible/status/1438510437885091851?s=20>
10. <https://twitter.com/mraible/status/1438579388233224193?s=20>
11. <https://www.jbang.dev/>
12. <https://quarkus.io/guides/performance-measure>
13. Sadly, I don't have a quotable source for this, Clement said it in an email. But I thought it should be shared!
14. <https://mail.openjdk.java.net/pipermail/discuss/2021-September/005953.html>: GraalVM Capabilities & Leyden
15. <https://youtu.be/9si7gK94gLo?t=1163> -- Project Loom will kill reactive programming
16. <https://github.com/open-telemetry/opentelemetry-java>
17. <https://github.com/java-operator-sdk>