Oracle Groundbreakers

Serverless Java Challenges and Triumphs

David Delabassée
@delabassee
Serverless Team - Oracle
April, 2019

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.

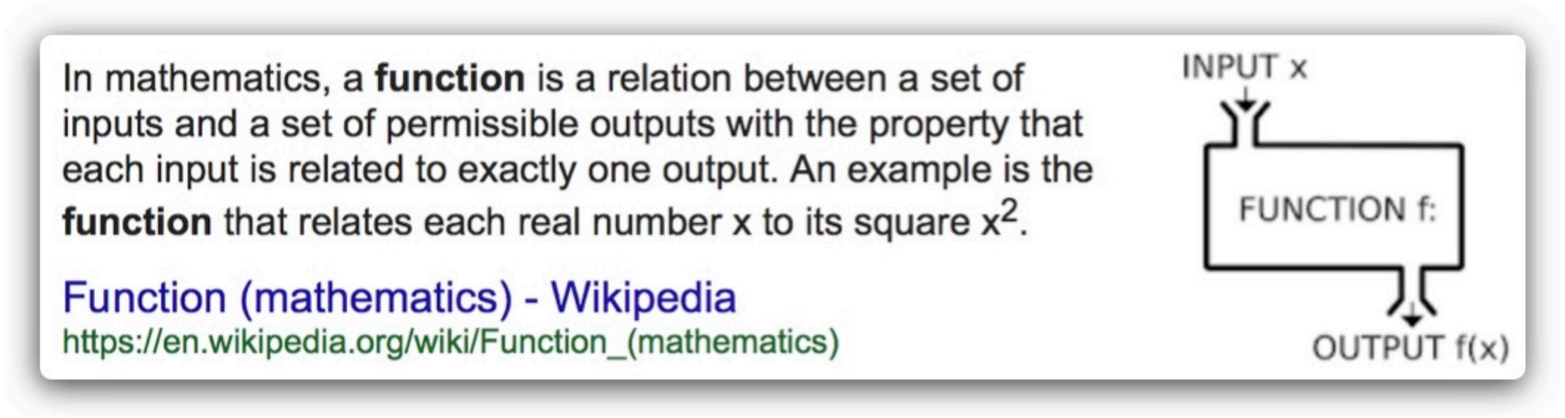
Oracle
Groundbreakers

Make, break, build.

ORACLE

Serverless

Function As a Service



In mathematics, a **function** is a relation between a set of inputs and a set of permissible outputs with the property that each input is related to exactly one output. An example is the **function** that relates each real number x to its square x^2.

Function (mathematics) - Wikipedia
https://en.wikipedia.org/wiki/Function_(mathematics)

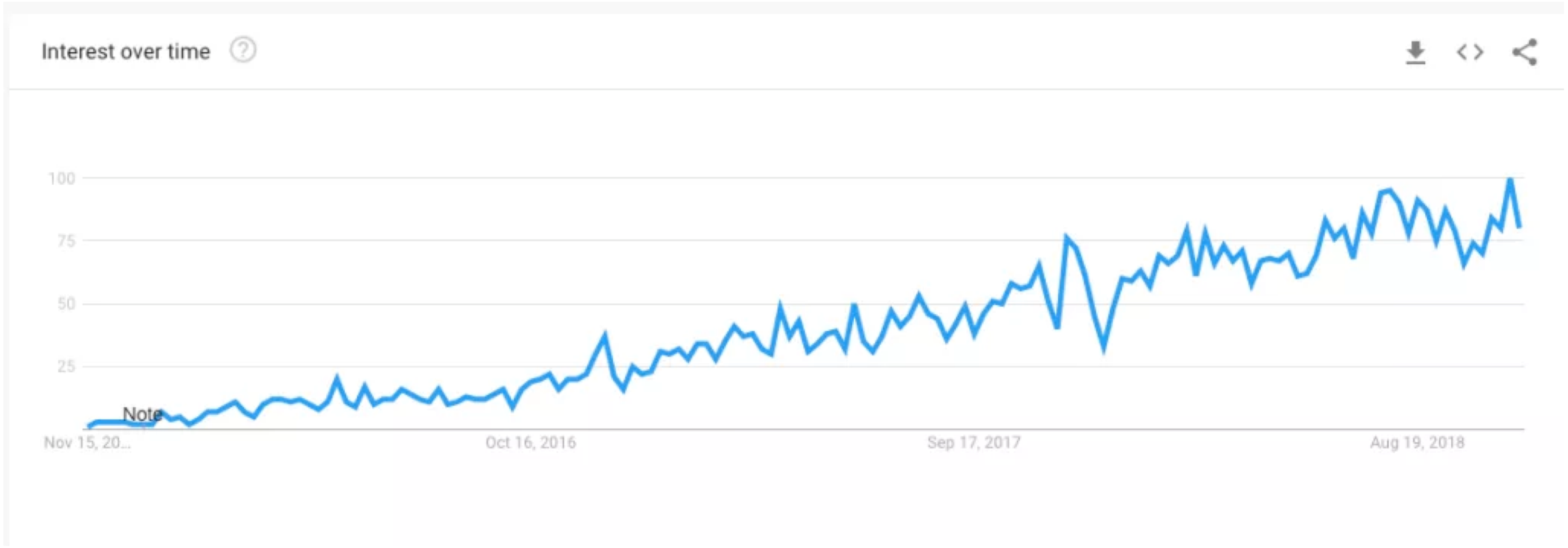
- **Function** Small bits of code with a well defined job

  Easy to understand and maintain

- **As a Service** The system takes care of provisioning, patching, scaling, ...

  Each function can scale independently

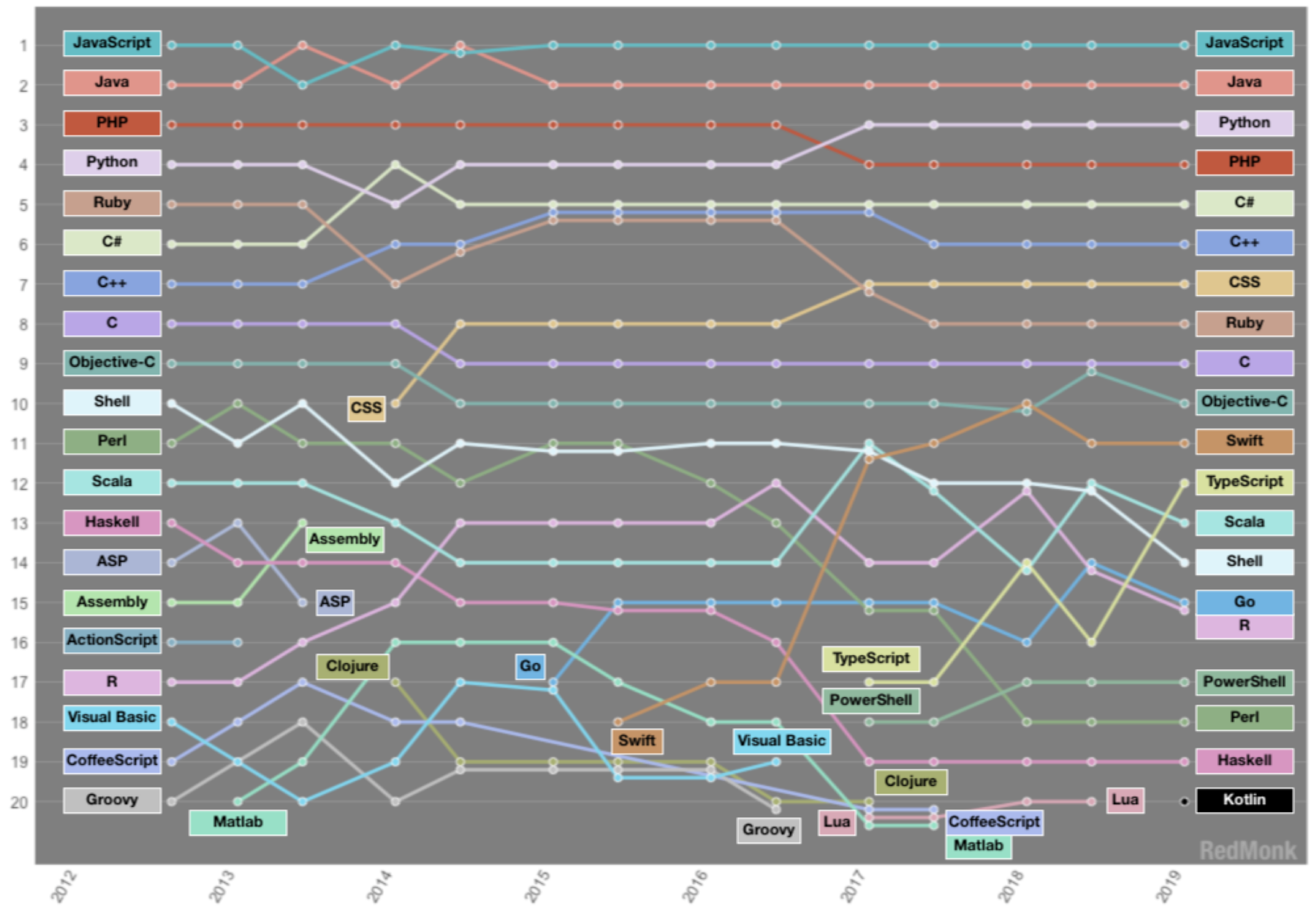
Interest in Serverless



Source: Google Trends

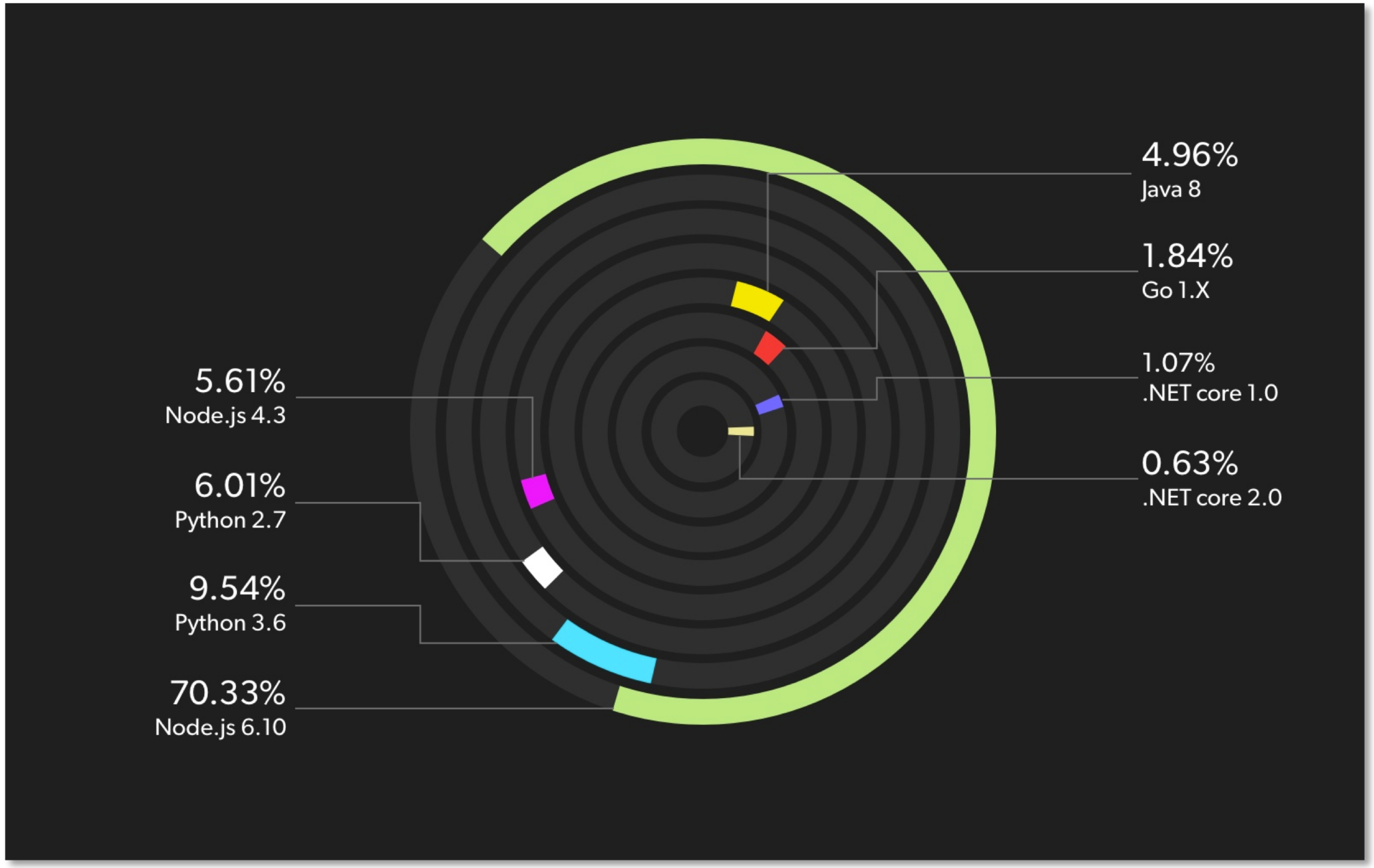
Serverless Java

RedMonk Top 20 Languages Over Time: Sept 2012 - Jan 2019



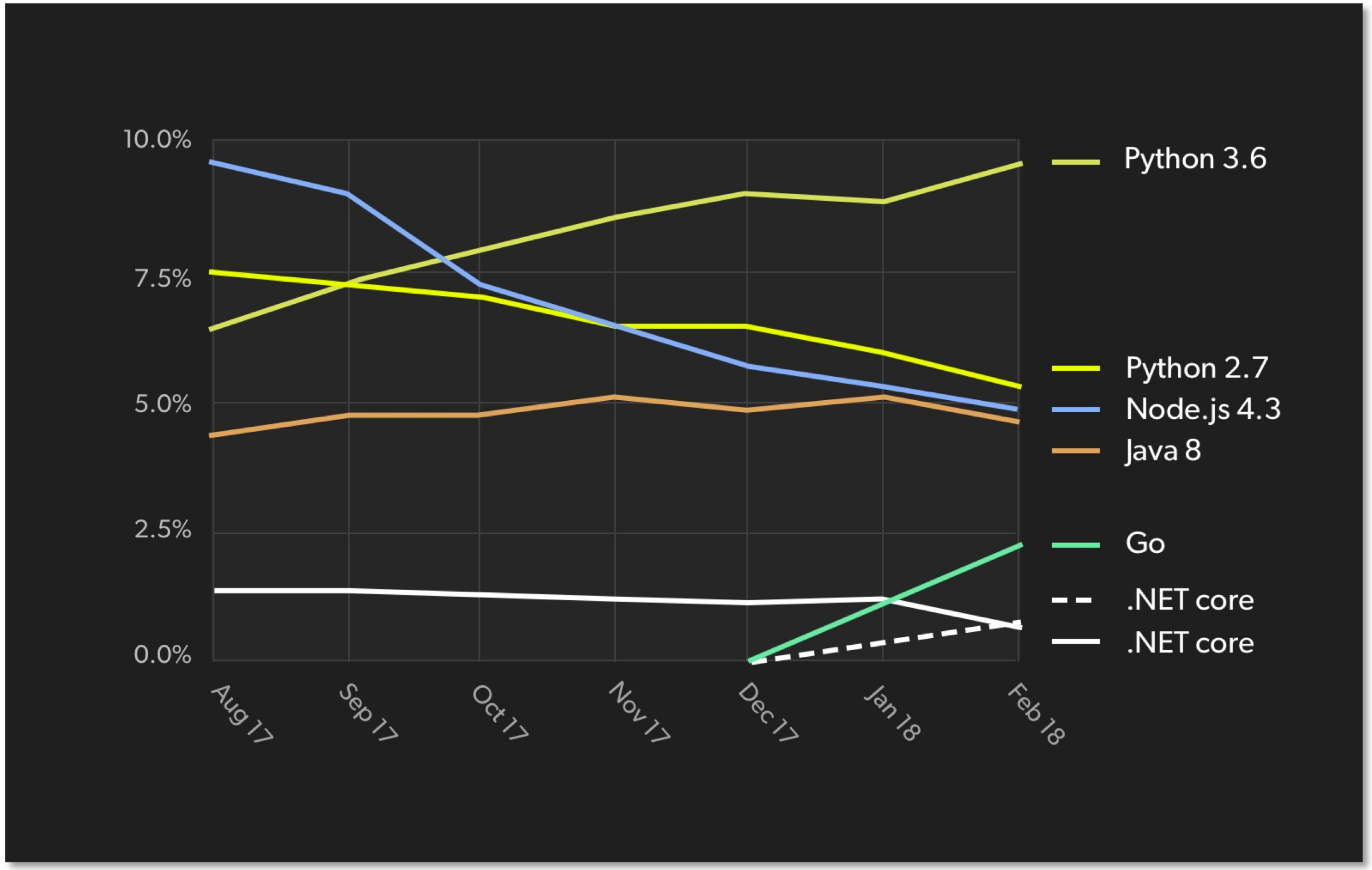
https://redmonk.com/rstephens/2019/03/20/redmonk-top-20-languages-over-time-january-2019/

4.96% of functions are Java 8!?



https://serverless.com/blog/serverless-by-the-numbers-2018-data-report/

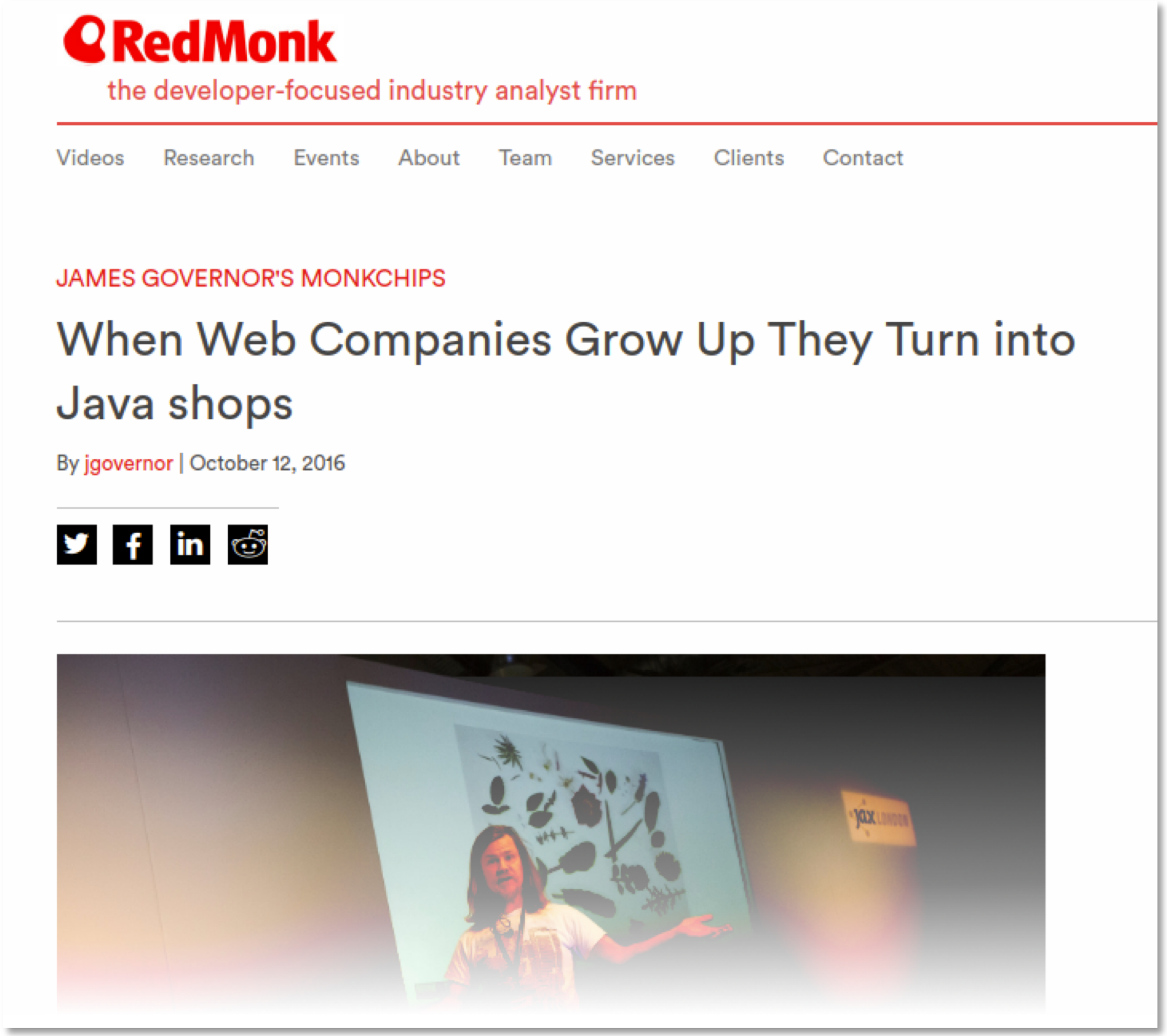
And the Trend isn't Great!



https://serverless.com/blog/serverless-by-the-numbers-2018-data-report/

Serverless Java Landscape

- AWS Lambda - Java 8 support (June 2015)
- Azure Function - Java 8 support (February 2019)
- Google Cloud Function - NA


RedMonk
the developer-focused industry analyst firm

Videos Research Events About Team Services Clients Contact

JAMES GOVERNOR'S MONKCHIPS

When Web Companies Grow Up They Turn into Java shops

By jgovernor | October 12, 2016

http://redmonk.com/jgovernor/2016/10/12/when-web-companies-grow-up-they-turn-into-java-shops

Serverless Java

- FaaS seen as a scripting platform for the web?
- Doesn't fit normal Java development patterns?
- JVM not suitable for short-lived “apps”?

Blueprints for Serverless Java

- “Plain old Java”
- Established toolchains
- Ability to build complex applications
- Low latency/high performance
- JVM ecosystem

fn

- Open-source, Container Native, Serverless Platform
- Apache v2 licence
- Run anywhere - Cloud / Datacenter / Laptop
- Fn ❤️ Java
- Functions are containers

https://github.com/fnproject/fn

Demo

Fn Java Function Development Kit

- Docker images
  - A build image for repeatable builds
  - An optimized runtime image
- JUnit test harness
- Maven support
- Input/output coercion
- Flow
- …

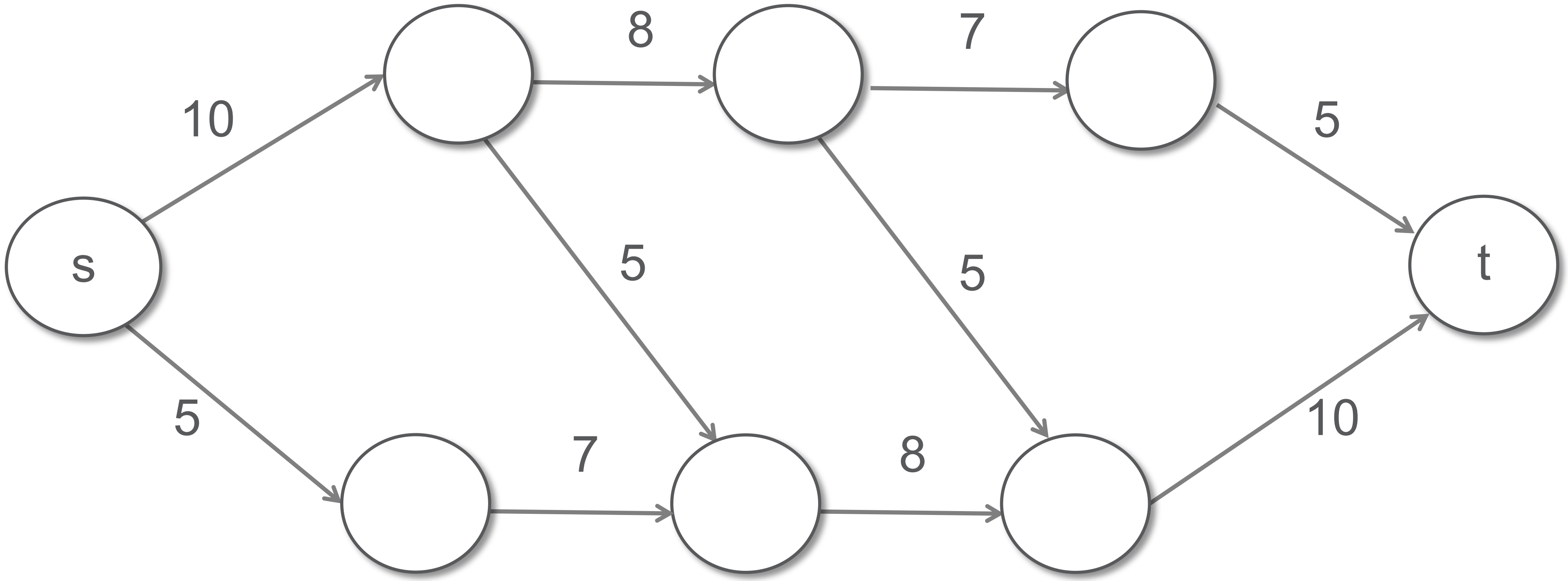
Blueprints for Serverless Java

- “Plain old Java” ✓
- Established toolchains ✓
- Ability to build complex applications
- Low latency/high performance
- JVM ecosystem

Fn Flow

Serverless

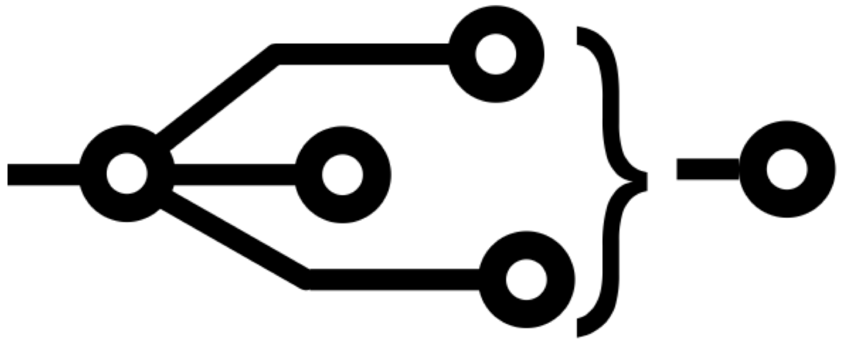
Permanent Storage Lives Elsewhere



Fn Flow

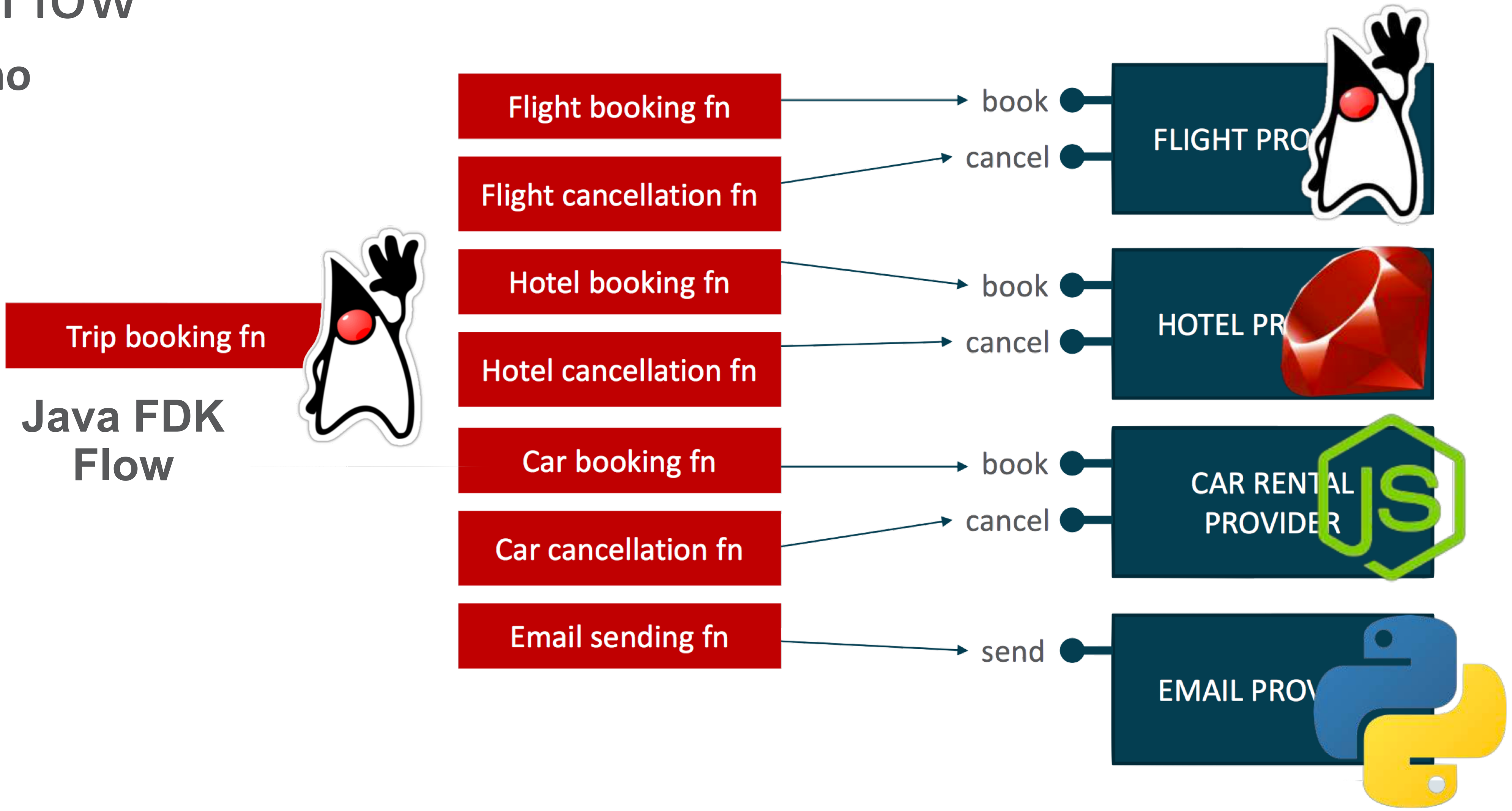
For long-running, reliable, scalable functions

- Build reliable and scalable **Faas applications**
- Provides **rich concurrency primitives** including fork-join, chaining, delays and error handling
- Java support based on Java 8 **CompletionStage** API



Fn Flow

Demo


Trip booking fn
Java FDK Flow
Flight booking fn
Flight cancellation fn
Hotel booking fn
Hotel cancellation fn
Car booking fn
Car cancellation fn
Email sending fn
book
cancel
book
cancel
book
cancel
send
FLIGHT PRO
HOTEL PR
CAR RENTAL PROVIDER
EMAIL PROV

Blueprints for Serverless Java

- “Plain old Java” ✓
- Wide choice of good tooling ✓
- Ability to build complex applications ✓
- Low latency/high performance
- Rich JVM ecosystem

Low latency/high performance

What do we want containerized JVMs to do?

- Respect resource constraints
- Start ***fast***
- Run in ***small(er)*** images

Respect Resource Constraints

Being Container Friendly

- JDK-8179498: `attach` in Linux should be relative to `/proc/pid/root` and namespace aware as `jcmd`, `jsack`, etc. fail to attach (resolved in JDK 10)
- JDK-8193710: `jcmd -l` and `jps` commands do not list Java processes running in Docker containers (resolved in JDK 11)
- And more... JDK-8203357: Container Metrics (resolved in JDK 11.0.1)

- JDK 11—JEP 318: *Epsilon*, i.e. No-Op, Garbage Collector (experimental)

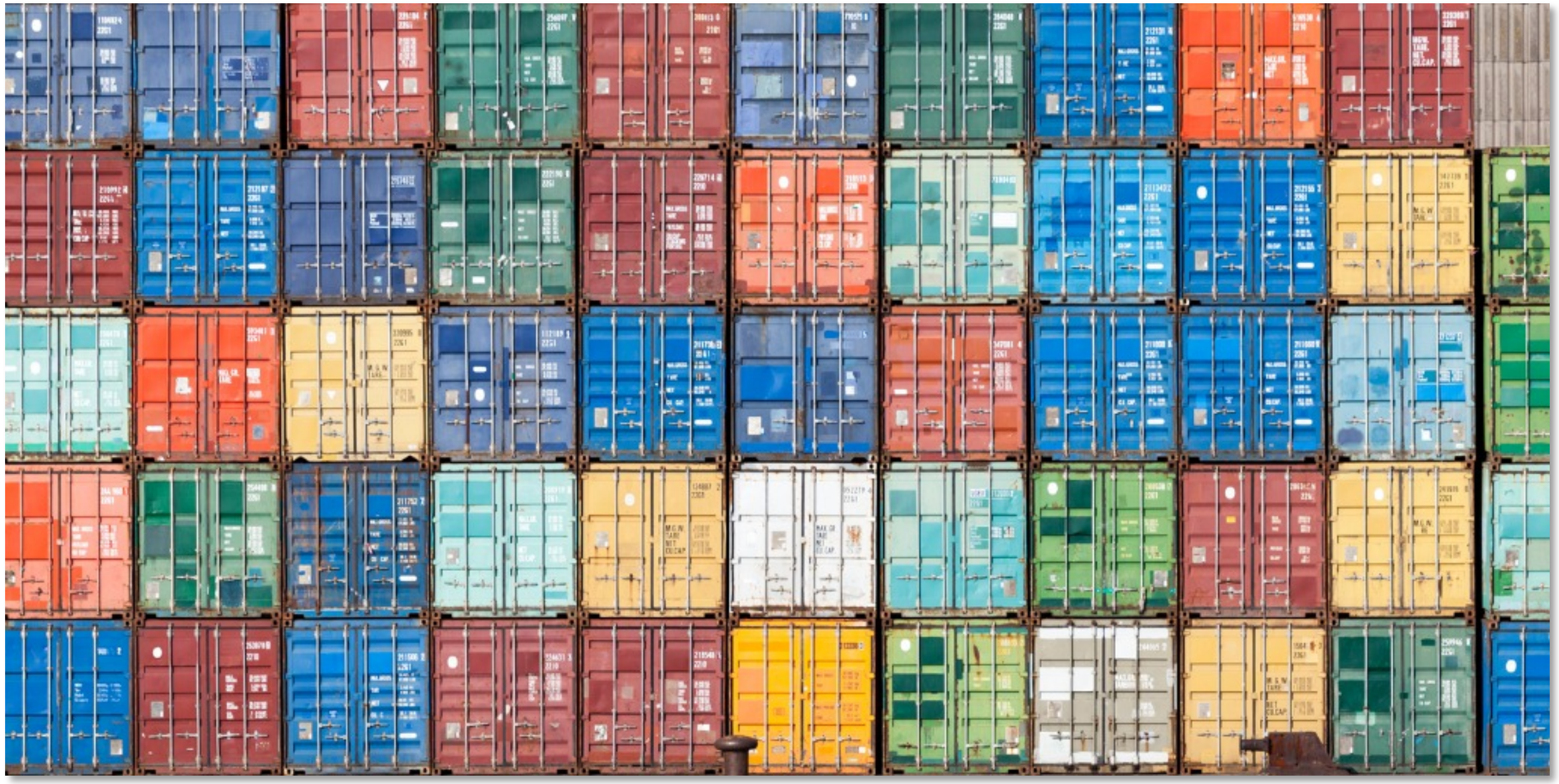
Start Fast

Moving Startup Costs to Build-Time

- Class Data Sharing
  - Avoid parsing JDK classes on start
- Application CDS
  - Avoid parsing App classes on start

Start Fast

Small(er) Container Images = Faster Start-up



Run Small(er) Images

Reduce layers size

- Java Function and its dependencies
- Java Runtime
- Operating System

Run Small(er) Images

Reduce layers size

- Project Portola
  - Run the JVM on musl
  - https://openjdk.java.net/projects/portola/
- musl
  - Lightweight, fast, simple, free, C standard library implementation
  - http://www.musl-libc.org
- Alpine
  - Security-oriented, lightweight Linux distro with 4MB base image
  - https://www.alpinelinux.org

Run Small(er) Images

Reduce Java Runtime layer size - jlink

	Modules	JLink flags	Mb		
JDK 12	Whole JDK!		318.7		
openjdk:11-jre-slim 11	(default)	NB: openjdk:12-jre-slim not yet available!	217.0		
JRE 12	all (explicit)	--add-module $(java --list-modules)	**168.3**	**100.0%**	
		+ --no-header-files --no-man-pages --strip-debug	143.0	85.0%	
		+ --compress=1	107.8	64.1%	
		+ --compress=2	83.7	49.7%	
Custom JRE 12	base, logging	--add-module $(jdeps --print-module-deps func.jar)	**47.4**	**28.2%**	**100.0%**
		+ --no-header-files --no-man-pages --strip-debug	41.6	24.7%	87.8%
		+ --compress=2	**32.0**	**19.0%**	**67.5%**

318 Mb → 168 Mb → 47 Mb → 32 Mb

OpenJDK (build 12-ea+29) - alpine:3.9 x86_64

Jlink/Alpine Demo

Run Small(er) Images

GraalVM

- GraalVM compiles Java source to a single native binary
- Tiny image sizes
- Low VM overhead

GraalVM™

GraalVM Demo

Blueprints for Serverless Java

- “Plain old Java” ✓
- Wide choice of good tooling ✓
- Ability to build complex applications ✓
- Low latency/high performance ✓
- Rich JVM ecosystem

Ongoing Ecosystem Evolution…

- Substrate VM
- Java release cadence
- Other JVM based languages
  - Kotlin, Groovy, etc.
- …
- Fn `init-image`

Blueprints for Serverless Java

- “Plain old Java” ✓
- Wide choice of good tooling ✓
- Ability to build complex applications ✓
- Low latency/high performance ✓
- Rich JVM ecosystem ✓

Serverless Java—Does it have a future? Absolutely!

fn

OpenJDK GraalVM™

Call to Action

https://github.com/fnproject/fn

Introducing Oracle Functions

Oracle Functions

Functions-as-a-Service

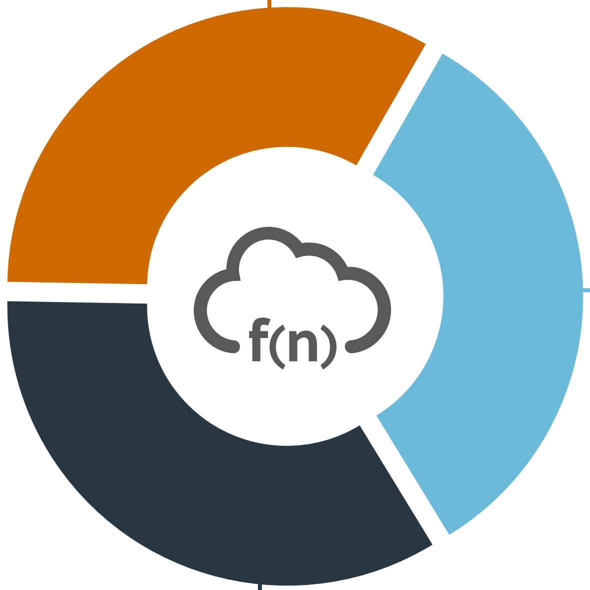
Oracle Cloud Integrated

Container Native

Open Source Engine

Multi-tenant

Secure



Pay Per Use
Pay for execution, not for idle time

Autonomous
Platform auto-scales functions
No servers to provision, manage

No Lock-in
Built on open-source Fn Project and Docker

Thank You!