Introduction to Micronaut

Ultra-Lightweight Microservices for the JVM

— by Graeme Rocher

OCI | WE ARE SOFTWARE ENGINEERS.

About Me - Graeme Rocher

- Creator of Grails (http://grails.org)
- Creator of Micronaut (http://micronaut.io)
- Author "The Definitive Guide to Grails"
- Former SpringSource -> VMware -> Pivotal
- Senior Engineer at Object Computing (http://objectcomputing.com)
- Just Received 2018 Oracle Groundbreaker award - Thanks!

OCI | WE ARE SOFTWARE ENGINEERS.

Agenda

— How we got here
— Microservice Challenges
— Microservice Framework Lanscape
— Micronaut Demos

OCI | WE ARE SOFTWARE ENGINEERS.

Then and Now

— Since 2008, a lot has changed
— 10 Years is a long time in technology
— Everybody was building Monoliths
— No Angular, No React, No Docker, No Microservices

So We Try to Adapt

— Let's try adapt existing legacy technologies for Microservices

— Technologies like Spring, Jakarta EE etc were never optimized for low memory footprint Microservices

What to do, What to do?

Shall we:

1. Try and convince people that something never designed for Microservices is still ok?
2. Go back to the drawing board

The Goal

— Create a New Framework designed from the ground-up for Microservices and Serverless Computing
— Blazing fast startup time
— Low Memory Footprint
— As Small As Possible JAR Sizes
— Zero Dependency
— 12 Factor - https://12factor.net

The Analysis

To meet this goal we performed an analysis of Spring and Grails and the challenges to using them to develop Microservice applications

What Spring and Jakarta EE Do

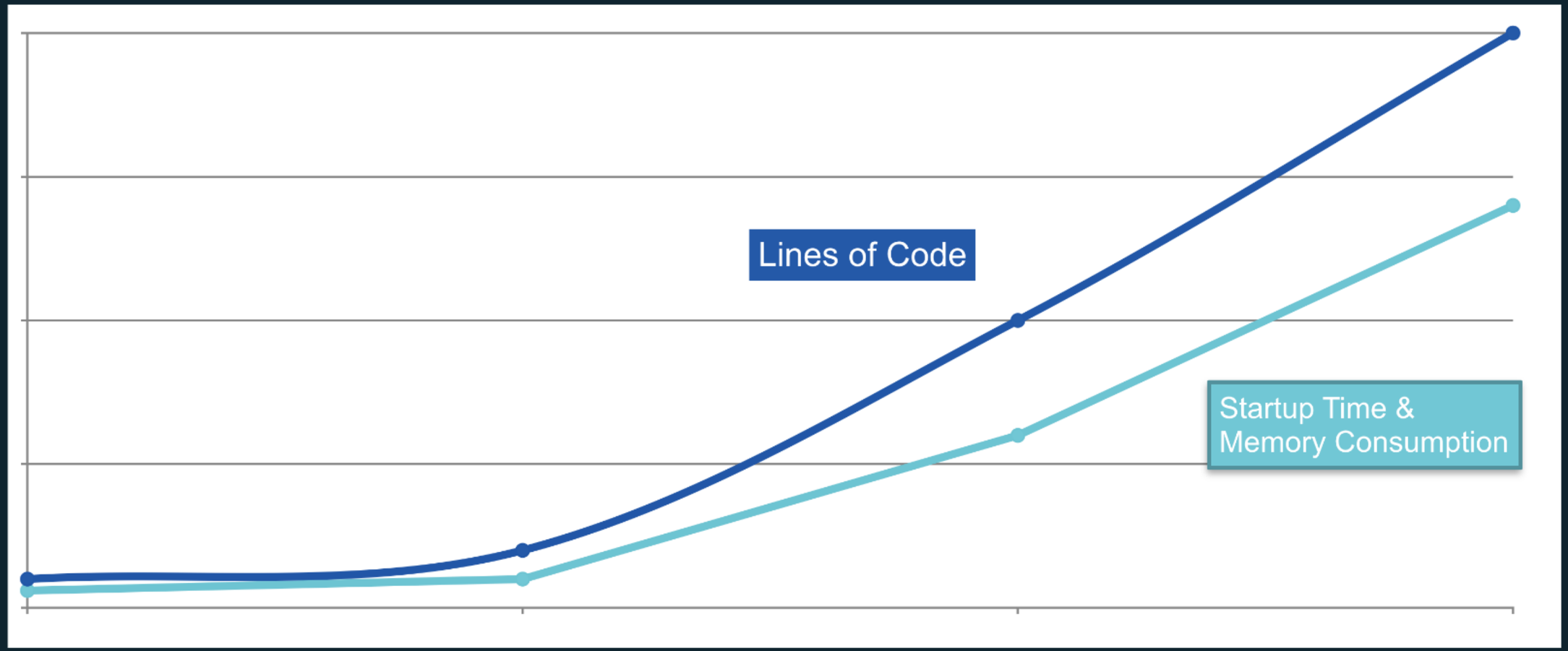
Spring is an amazing technical achievement and does so many things, but does them at Runtime.

— Reads the byte code of every bean it finds

— Synthesizes new annotations for each annotation on each bean method, constructor, field etc. to support Annotation metadata

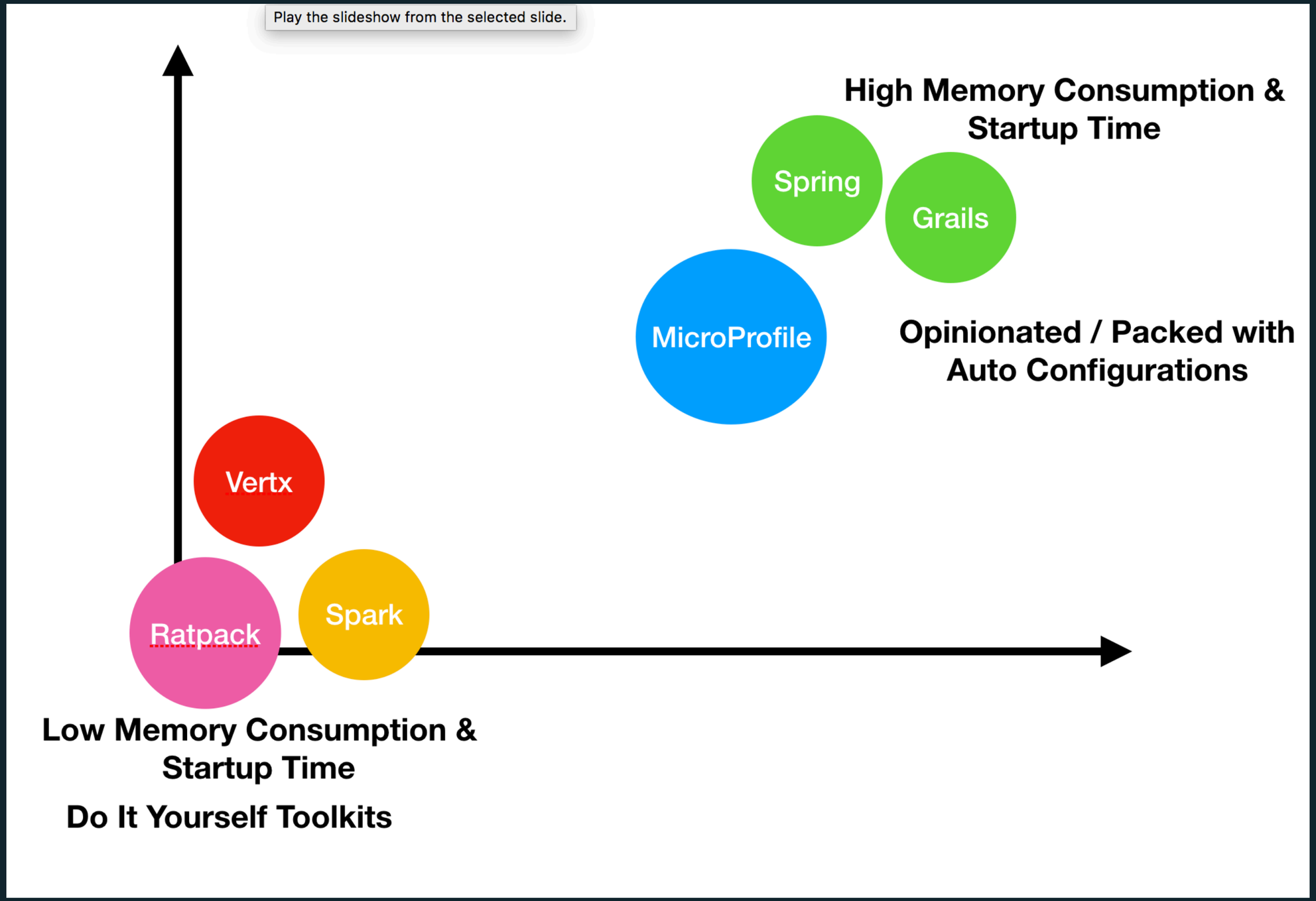
— Builds Reflective Metadata for each bean for every method, constructor, field etc.

So What's the Problem?



OCI | WE ARE SOFTWARE ENGINEERS.

Play the slideshow from the selected slide.



The Micro Reality

- Frameworks based on reflection and annotations become fat
- But we love the programming model and productivity so we live with it
- So … why should we be more efficient?

Imagine if Kubernetes or Docker had been written in Spring or Jakarta EE instead of Go?

Already Solved by Ahead of Time (AOT) Compilation

- The Android Community already solved the problem
- Ahead of Time Compilation used extensively
- Google Dagger 2.x
  - Compile Time Dependency Injector
  - Reflection Free
  - Limited in Scope to just DI

MICRONAUT

Introducing Micronaut

- Designed from the ground up with Microservices in mind
- Ultra-light weight and Reactive - Based on Netty
- Uses Ahead of Time Compilation
- HTTP Client & Server
- Support for Java, Kotlin and Groovy

MICRONAUT

OCI | WE ARE SOFTWARE ENGINEERS.

DEMO

— Hello Micronaut

OCI | WE ARE SOFTWARE ENGINEERS.

MICRONAUT

Hello Micronaut

```
@Controller
class HelloController {
    @Get("/hello/{name}")
    String hello(String name) { return "Hello " + name; }
}
@Client("/") // Client Generated at Compile Time
interface HelloClient {
  @Get("/hello/{name}")
  String hello(String name);
}
```

OCI | WE ARE SOFTWARE ENGINEERS.

MICRONAUT

How Small?

— Smallest Micronaut Hello World JAR is 10MB when written Java or 12MB in Groovy

— Can be run with as little as 10mb Max Heap with Kotlin or Java (22mb for Groovy)

— Startup time around a second for Kotlin or Java (a little more for Groovy)

— All Dependency Injection, AOP and Proxy generation happens at compile time

What Micronaut Computes at Compile Time

— All Dependency & Configuration Injection
— Annotation Metadata (Meta annotations)
— AOP Proxies
— Essentially all framework infrastructure
— ie. What Spring/CDI do at runtime

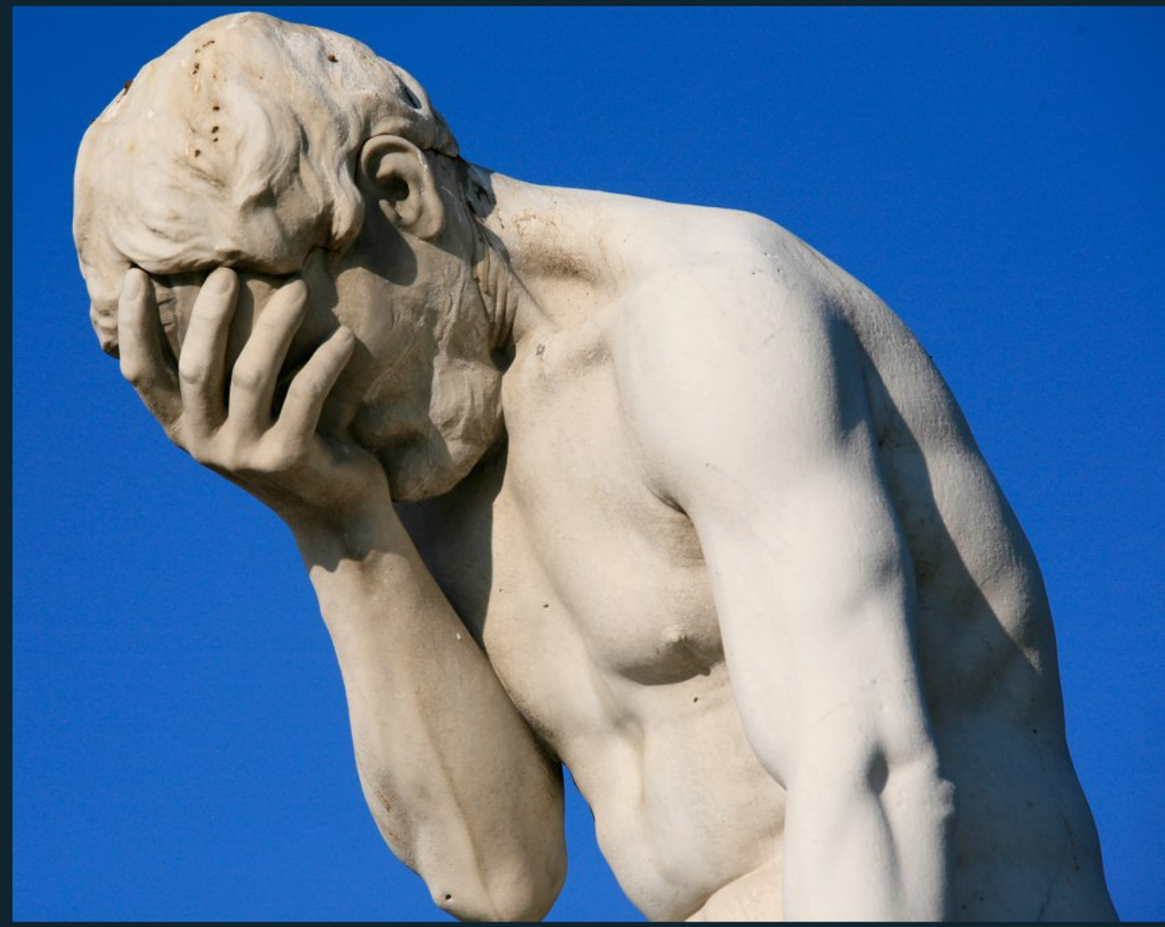
OCI | WE ARE SOFTWARE ENGINEERS.

MICRONAUT

Not Another Framework!?

— If all we had achieved was another HTTP server Micronaut wouldn't be very interesting

— So what else does it do?



Natively Cloud Native

— Service Discovery - Consul, Eureka, Route 53 and Kubernetes

— Configuration Sharing - Consul Supported and Amazon ParameterStore

— Client Side Load Balancing - Integrated or Netflix Ribbon Supported

— Support for Serverless Computing; AWS Lambda, OpenFaas, Fn Supported; Azure coming

OCI | WE ARE SOFTWARE ENGINEERS.

MICRONAUT

DEMO

— Micronaut Pet Store

OCI | WE ARE SOFTWARE ENGINEERS.

MICRONAUT

Serverless Computing

OCI | WE ARE SOFTWARE ENGINEERS.

— Write Functions and Run them locally or as regular server applications

— Deploy Functions to AWS Lambda - after warm-up functions execute in milliseconds

```
@Field @Inject Twitter twitter

@CompileStatic
URL updateStatus(Message status) {
    Status s = twitter.updateStatus(status.text)
    String url = "https://twitter.com/$s.user.screenName/status/${s.id}"
    return new URL(url)
}
```

GraalVM

- New Polyglot VM from Oracle
- Runs JS, Java, Ruby, R etc.
- Ability to turn Java code native
- https://www.graalvm.org

GraalVM Native

- Works well when:
  - Little or no runtime reflection is used
  - Limited or no dynamic classloading
  - You plan ahead
  - You use third party libraries selectively

DEMO

— Micronaut + GraalVM

Micronaut 1.0 Out Now

— Compile Time DI & AOP
— HTTP Client & Server
— Service Discovery
— Distributed Tracing
— Serverless Functions
— Data Access: SQL, MongoDB, Redis, Cassandra etc.

MICRONAUT

OCI | WE ARE SOFTWARE ENGINEERS.

Micronaut 1.0 on SDKman!

— The Micronaut CLI now available
via SDKman!

```
$ curl -s "https://get.sdkman.io" | bash
$ source "$HOME/.sdkman/bin/sdkman-init.sh"
$ sdk install micronaut
$ mn create-app hello-world
```

MICRONAUT

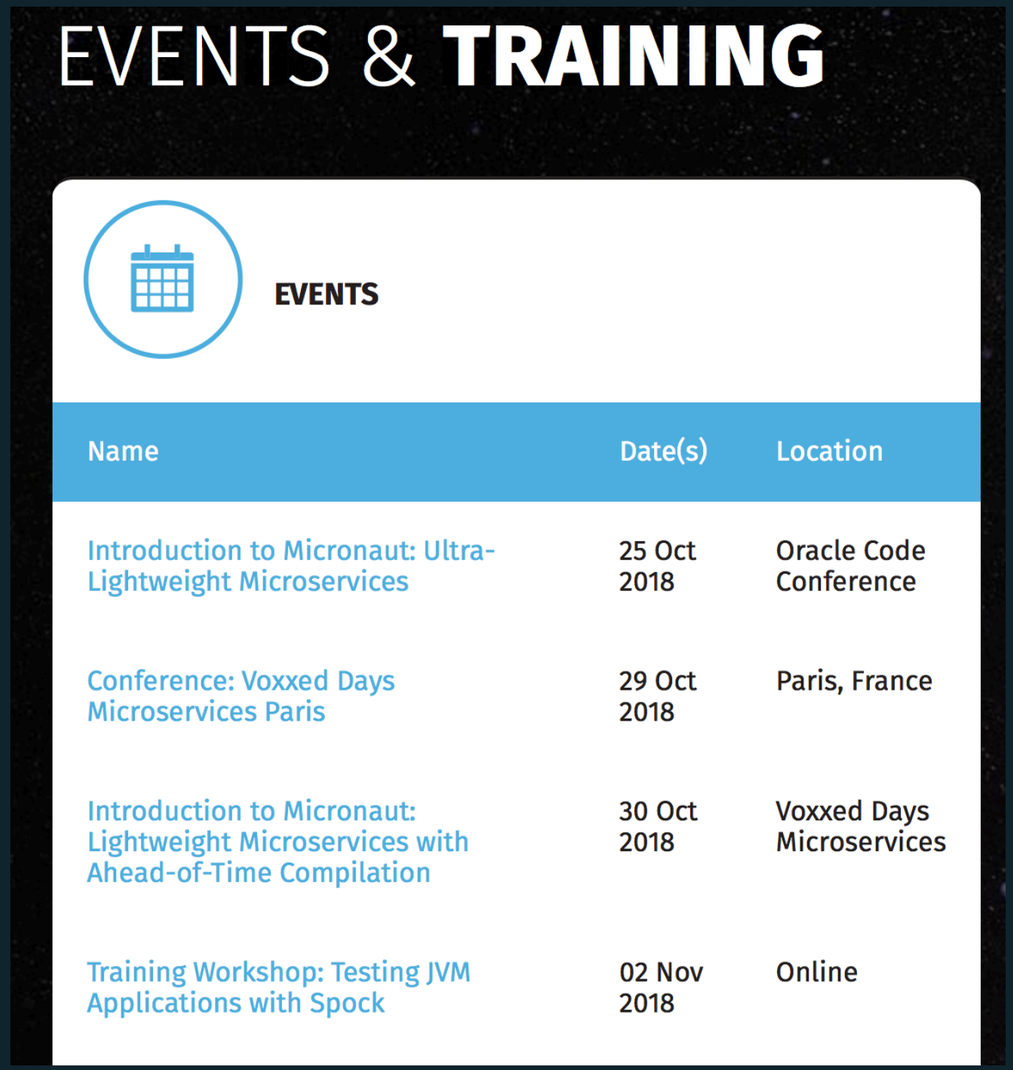
OCI | WE ARE SOFTWARE ENGINEERS.

Micronaut Resources

- Gitter Community: https://gitter.im/micronautfw
- User Guide: http://micronaut.io/documentation.html
- Micronaut Guides: http://guides.micronaut.io
- FAQ: http://micronaut.io/faq.html
- Github: https://github.com/micronaut-projects/micronaut-core
- Examples: https://github.com/micronaut-projects/micronaut-examples

Micronaut Events

- Loads of upcoming Events
- Checkout - http://micronaut.io/events.html
- Fancy a trip to Paris?
  - https://voxxeddays.com



Summary

- Micronaut aims to provide the same wow factor for Microservices that Grails did for Monoliths
- Built by the people that made Grails, leveraging over 10 years experience in framework development
- Uses Ahead of Time Compilation to support low-memory footprint
- Micronaut 1.0 is available now

Q & A

OCI | WE ARE SOFTWARE ENGINEERS.