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.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Java in Serverless Land

Matthew Gilliard
Twitter: @MaximumGilliard

Fn Project // Container Native Group

javaBin Oslo
Aug 23rd 2018

JavaYourNext (Cloud)

Java Day ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Fn Project is a **Serverless platform**. But what *is* that?

- Create applications without thinking about Infrastructure.
- Unit of deployment is a **Function**
- Hence: **Functions-as-a-Service** or **FaaS**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

…and what is **a Function**?

- Short-lived
- Ephemeral / Event-driven
- Stateless

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Java in Serverless Land

JavaYourNext (Cloud)

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Java is…

- the **No. 1** programming language
- **No. 3** in Serverless

… why?

Source: serverless.com

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

- Is it actually possible to use Java on Serverless?
- Is FaaS more than just a “scripting language” for the web?
- Can Java development habits be used for FaaS?
- Can the JVM effectively run short-lived apps?

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

- Is it actually possible to use Java on Serverless? **Yes!**
- Is FaaS more than just a “scripting language” for the web? **Yes!**
- Can Java development habits be used for FaaS? **Yes!**
- Can the JVM effectively run short-lived apps? **Yes!**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

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

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

What does ”Grown-up” Java development look like?

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

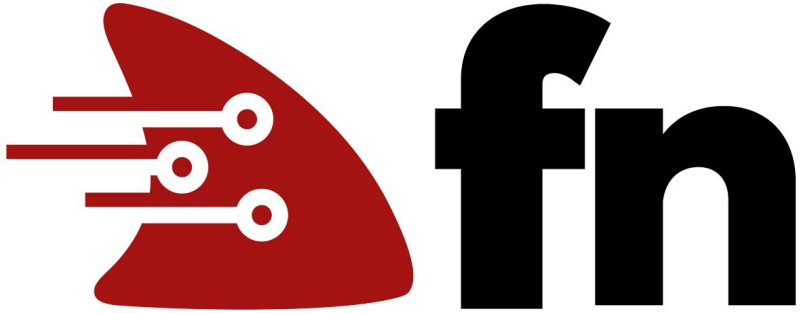
Fn Project

JavaYourNext

(Cloud)

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

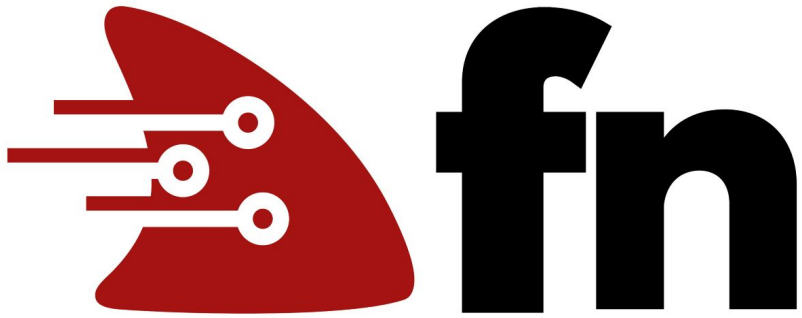
Fn Project – an Open-Source FaaS platform



http://fnproject.io

- Fully **open** (Apache 2.0) with commercial backing from Oracle
- Run **anywhere** - Datacenter / laptop / rPi
- Functions are **containers**
- Fn ♥ Java

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |



Demo

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Fn Flow

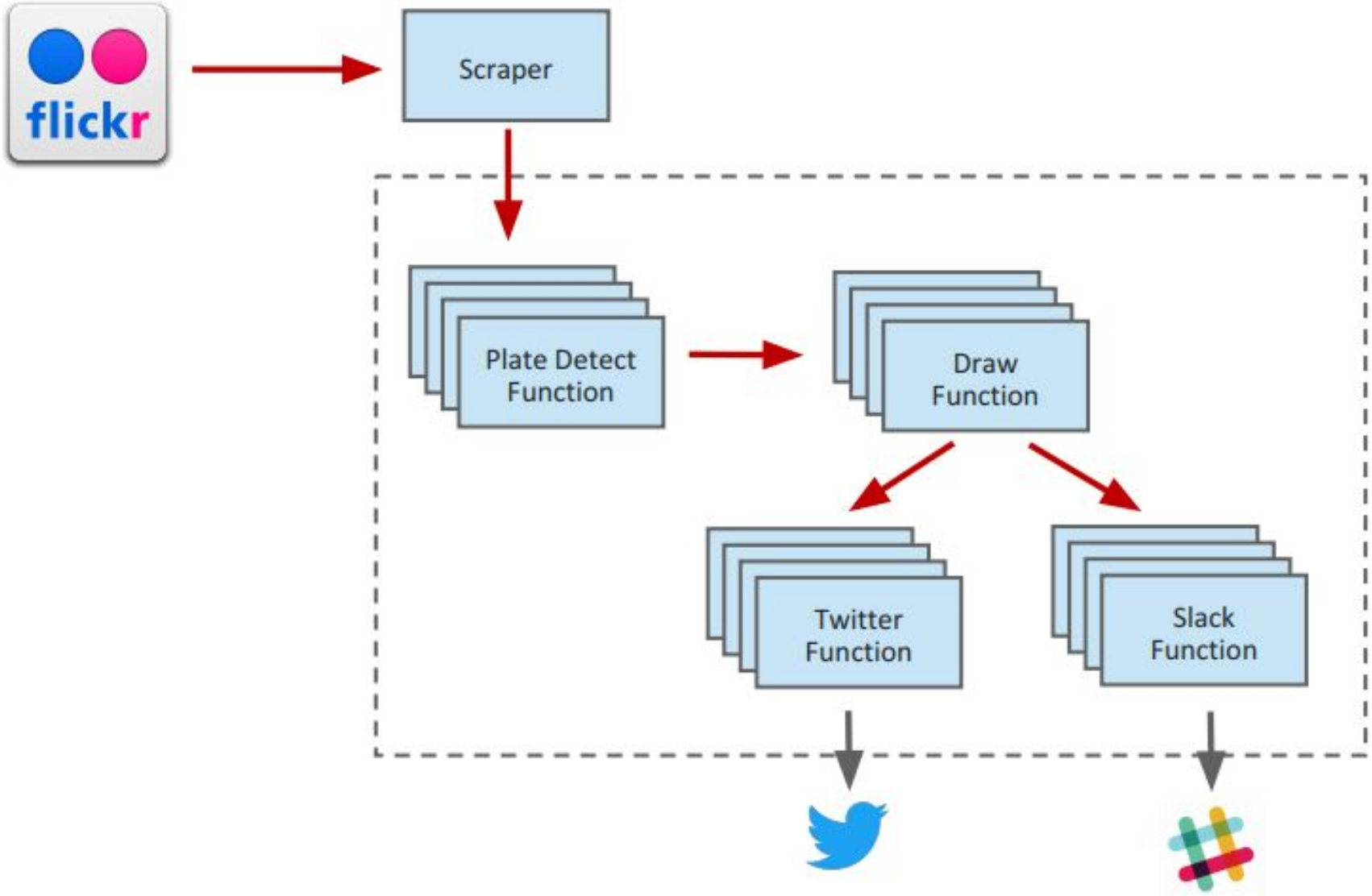
JavaYourNext (Cloud)

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

What is Fn Flow?

- Fn Flow lets you build **reliable** and **scalable** applications out of functions
- Fn Flow supports **rich concurrency** options
- Fn Flow lets **regular functions** define workflows
- Fn Flow functions are written **purely as code**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Fn Flow Under the Hood

- Inspired by the **CompletionStage** API from JDK 8:

```
public class CompletableFuture<T>
implements Future<T>, CompletionStage<T>
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

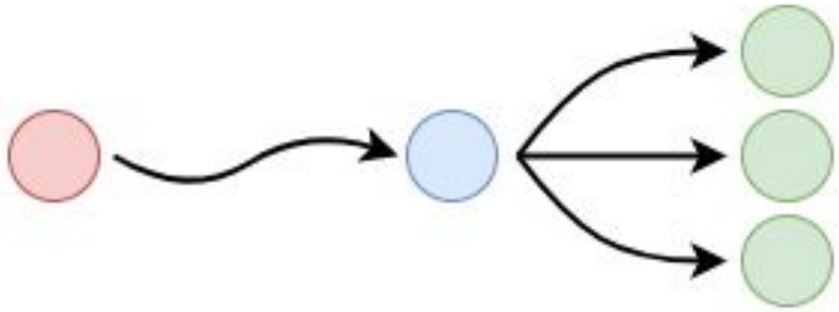
```
CompletableFuture<Image> imageRequest = makeHttpImageRequest(imgName);
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

```
CompletableFuture<Image> imageRequest = makeHttpImageRequest(imgName);

CompletableFuture<FaceDetectionResult> facesResult =
        imageRequest.thenApply( image -> detectFaces(image) );
```

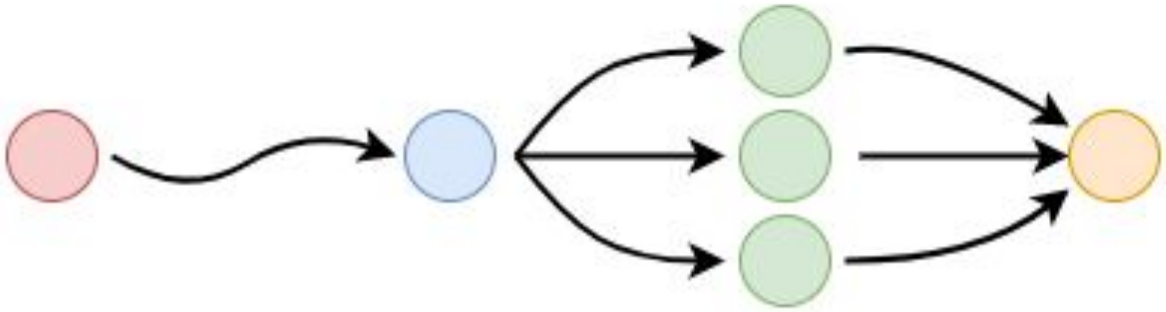
Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

```
CompletableFuture<Image> imageRequest = makeHttpImageRequest(imgName);

CompletableFuture<FaceDetectionResult> facesResult =
        imageRequest.thenApply( image -> detectFaces(image) );

res1 = facesResult.thenApply( moodDetectionMethodSimple );
res2 = facesResult.thenApply( moodDetectionMethodComplex );
res3 = facesResult.thenApply( moodDetectionMethodExperimental );
```



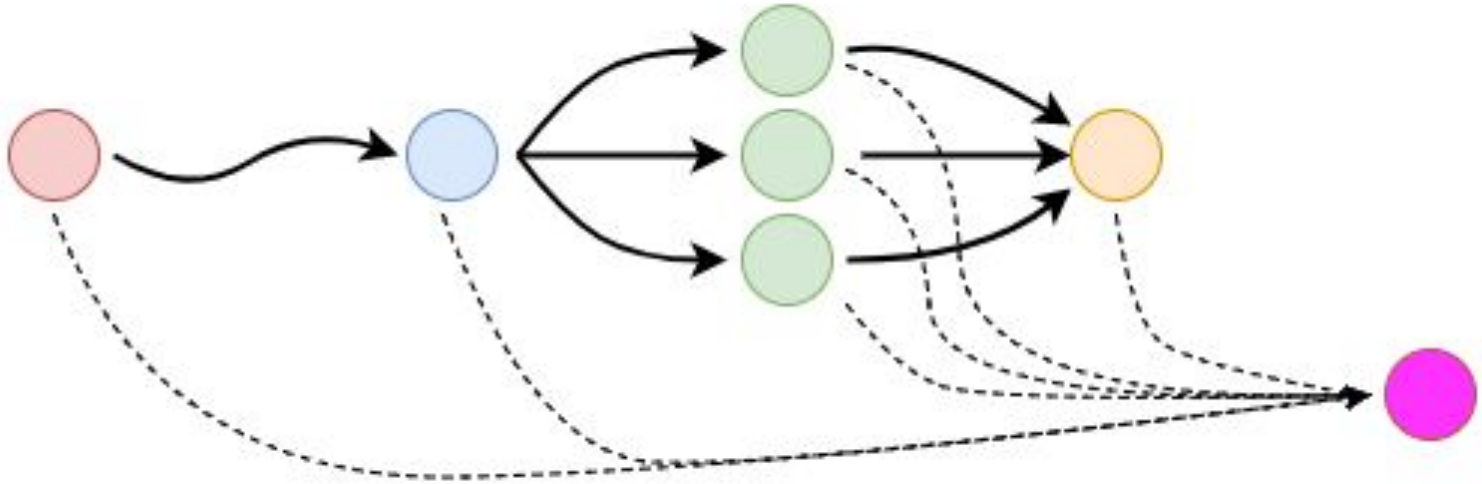
Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

```
CompletableFuture<Image> imageRequest = makeHttpImageRequest(imgName);

CompletableFuture<FaceDetectionResult> facesResult =
        imageRequest.thenApply( image -> detectFaces(image) );

res1 = facesResult.thenApply( moodDetectionMethodSimple );
res2 = facesResult.thenApply( moodDetectionMethodComplex );
res3 = facesResult.thenApply( moodDetectionMethodExperimental );

finish = CompletableFuture.allOf(res1, res2, res3).thenRun(
    registerDetectionFinished(imgName);
);
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

```
CompletableFuture<Image> imageRequest = makeHttpImageRequest(imgName);

CompletableFuture<FaceDetectionResult> facesResult =
        imageRequest.thenApply( image -> detectFaces(image) );

res1 = facesResult.thenApply( moodDetectionMethodSimple );
res2 = facesResult.thenApply( moodDetectionMethodComplex );
res3 = facesResult.thenApply( moodDetectionMethodExperimental );

finish = CompletableFuture.allOf(res1, res2, res3).thenRun(
    registerDetectionFinished(imgName);
);

finish.exceptionally( ex -> System.out.println("Oh no!") )
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

CompletionStage – What's going on?

A workflow of **asynchronous actions**

is constructed **in memory**

and completed asynchronously using the **ForkJoinPool**.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Fn Flow – What's going on?

A workflow of **asynchronous functions**

is constructed **in the Flow Server**

and completed asynchronously using the **FaaS**.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

CompletionStage API

A workflow of **asynchronous actions**

is constructed **in memory**

and completed asynchronously using the **ForkJoinPool**.

Fn Flow

A workflow of **asynchronous functions**

is constructed **in the Flow Server**

and completed asynchronously using the **FaaS**.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Effective JVM for containers

JavaYourNext

(Cloud)

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively

Remember: Fn functions are just **containers**

We want:

- **Fast** Startup
- **Small** Images
- **Respect** the environment

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: **Share memory** between JVMs

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: **Share memory** between JVMs

Shared image layers => Shared memory:

- **libc**, **libjava.so** & friends can be shared between JVMs by using a common base image

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: Move **startup** activities to **build time**, and cache the results

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: Move **startup** activities to **build time**, and cache the results

- **CDS** – pre-compute classloading for core classes

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: Move **startup** activities to **build time**, and cache the results

- **CDS** – pre-compute classloading for core classes
- **AppCDS** – same, but for application classes

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Fast Startup

Technique: Move **startup** activities to **build time**, and cache the results

- **CDS** – pre-compute classloading for core classes
- **AppCDS** – same, but for application classes
- **AOT** compilation – compile bytecode to native in advance

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Small Container images

Technique: Shrink images to enable faster **download** and **startup**

- OS components
- JDK
- Application Code
- JVM

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Small Container images

Technique: Shrink images to enable faster **download** and **startup**

- OS components **Project Portola** – JVM build for Alpine Linux
- JDK
- Application Code
- JVM

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Small Container images

Technique: Shrink images to enable faster **download** and **startup**

- OS components
- JDK
- Application Code
- JVM

Project Portola – JVM build for Alpine Linux

jlink – Removes parts of the JDK you don't need

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Small Container images

Technique: Shrink images to enable faster **download** and **startup**

- OS components **Project Portola** – JVM build for Alpine Linux
- JDK **jlink** – Removes parts of the JDK you don't need
- Application Code **Your code** - Use fewer and smaller dependencies
- JVM

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Small Container images

Technique: Shrink images to enable faster **download** and **startup**

- OS components **Project Portola** – JVM build for Alpine Linux
- JDK **jlink** – Removes parts of the JDK you don't need
- Application Code **Your code** - Use fewer and smaller dependencies
- JVM **SubstrateVM** – New lightweight VM from Oracle Labs

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Respecting the environment

Technique: Ensure that the JVMs **Ergonomics** chooses appropriate settings:

- Memory
  - Heap size
  - GC region sizes
  - JIT code cache sizes
- CPU
  - ThreadPool sizes
  - Runtime.availableProcessors()

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Using the JVM effectively – Respecting the environment

Technique: Ensure that the JVMs **Ergonomics** chooses appropriate settings:

Starting with JDK8 and improving through 9, 10 and 11 – the JDK supports this **automatically**.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

1. Program using **plain** Java
2. Strong acceptance of **Open-Source**
3. Huge number of available **tools** and **libraries**
4. **Complex** applications
5. Use the JVM **effectively**

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Thank You for Listening!

Check out Fn Project: **http://fnproject.io**

I've been Matthew Gilliard

Twitter: @MaximumGilliard

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |