Modern Java EE Microservices with WildFly Swarm

Ken Finnigan

Who is Ken Finnigan?

- Project lead for WildFly Swarm
- Contributor to Eclipse MicroProfile
- Author of several books

redhat®

Java EE

Skills

Knowledge

Experience

Microservices

Change Velocity

Bounded Context

- Each service does **one thing** well,with well-defined bounds
- If it used to be a library, it’s probably a service now

Java EE

and

Microservices ???

What we're here for!

Demo

```
<plugin>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>wildfly-swarm-plugin</artifactId>
  <version>${version.wildfly-swarm}</version>
  <executions>
    <execution>
    <id>package</id>
      <goals>
        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

```
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>jaxrs</artifactId>
</plugin>
```

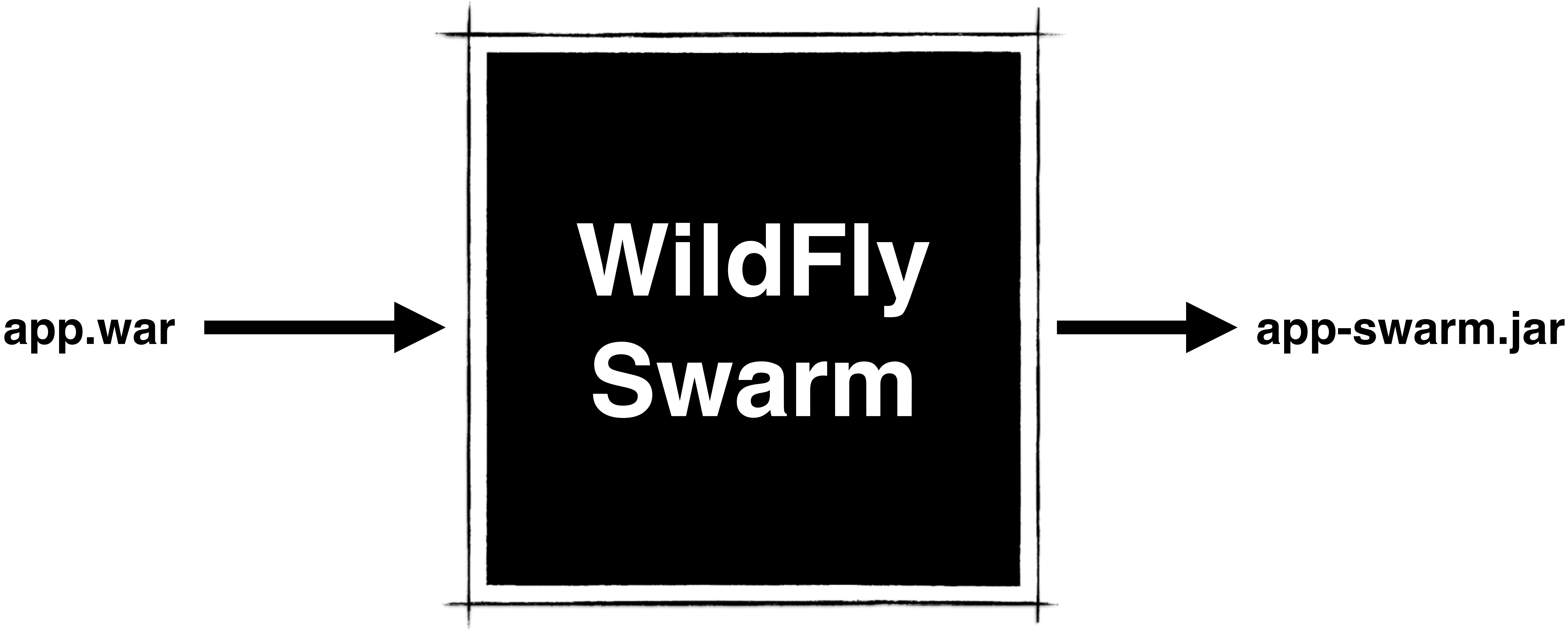
WildFly SWARM

It’s like WildFly

just swarmier

Just enough App Server!

WildFly Swarm allows you to wrap just the bits of a normal JavaEE app-server you need, ***around your app***.



app.war

WildFly Swarm

app-swarm.jar

- Servlet
- JAX-RS
- JMS
- CDI
- Batch
- JPA
- JAAS

- Transactions
- Naming
- Bean Validation
- Resource Adapters
- Java FX
- JMX
- More…

Auto Detection

```
<plugin>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>wildfly-swarm-plugin</artifactId>
  <version>${version.wildfly-swarm}</version>
  <executions>
    <execution>
    <id>package</id>
      <goals>
        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

WildFly Swarm figures out what bits you *need*

Demo

YAML

project-defaults.yaml

```
swarm:
  datasources:
    data-sources:
     MyDS:
        driver-name: h2
        connection-url:
jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE
        user-name: sa
        password: sa
```

What about "microservicey stuff" ?!

Keycloak

```
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>keycloak</artifactId>
</plugin>
```

define in project-defaults.yml

```
swarm:
  keycloak:
    secure-deployments:
      time.war:
        realm: service
        bearer-only: true
        auth-server-url: http://localhost:9090/auth
        ssl-required: external
        resource: time-service
        enable-cors: true
  deployment:
    time.war:
      web:
        security-constraints:
          - url-pattern: /*
            roles: [ time-access ]
```

```
swarm:
  keycloak:
    secure-deployments:
      time.war:
        realm: service
        bearer-only: true
        auth-server-url: http://localhost:9090/auth
        ssl-required: external
        resource: time-service
        enable-cors: true
  deployment:
    time.war:
      web:
        security-constraints:
          - url-pattern: /*
            roles: [ time-access ]
```

```
swarm:
  keycloak:
    secure-deployments:
      time.war:
        realm: service
        bearer-only: true
        auth-server-url: http://localhost:9090/auth
        ssl-required: external
        resource: time-service
        enable-cors: true
  deployment:
    time.war:
      web:
        security-constraints:
          - url-pattern: /*
            roles: [ time-access ]
```

Discovery

```
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>topology-*</artifactId>
</plugin>
```

Handles registration and discovery of services for NetFlix Ribbon support.

Demo

MicroProfile

WildFly SWARM

Eclipse MicroProfile

- 1.0 released September 2016
- 1.1 planned for Q2 2017
    - Config API
    - Health Check
    - JWT Security Token Interoperability

Bottom line..

Java EE

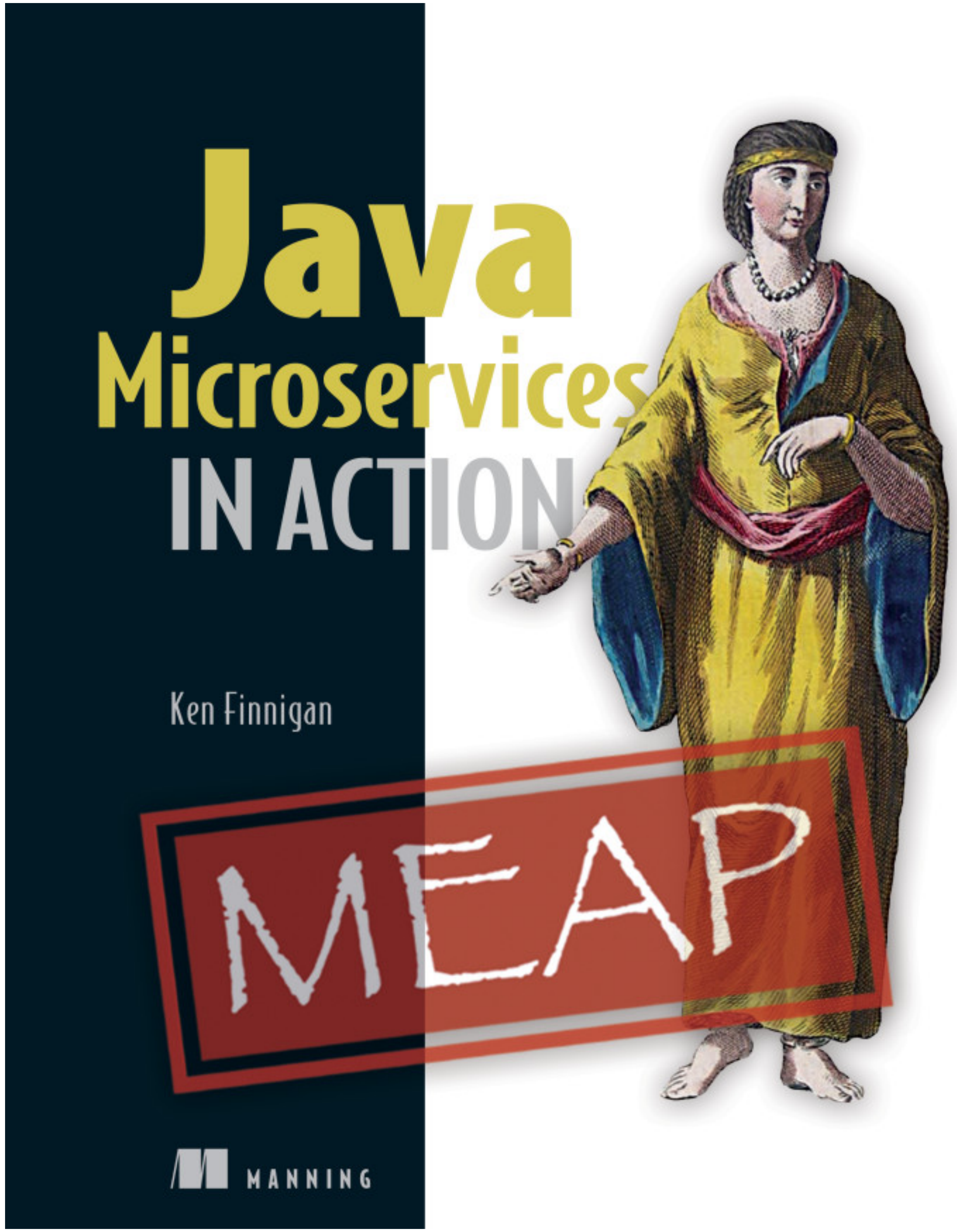
is an

awesomely fantastic

way to write microservices.

Currently writing

www.manning.com/books/java-microservices-in-action



Resources

http://github.com/wildfly-swarm/wildfly-swarm/

http://github.com/wildfly-swarm/wildfly-swarm-examples/

http://wildfly-swarm.io/

#wildfly-swarm on irc.freenode.net

issues.jboss.org/browse/SWARM

https://groups.google.com/forum/#!forum/wildfly-swarm

Thanks!