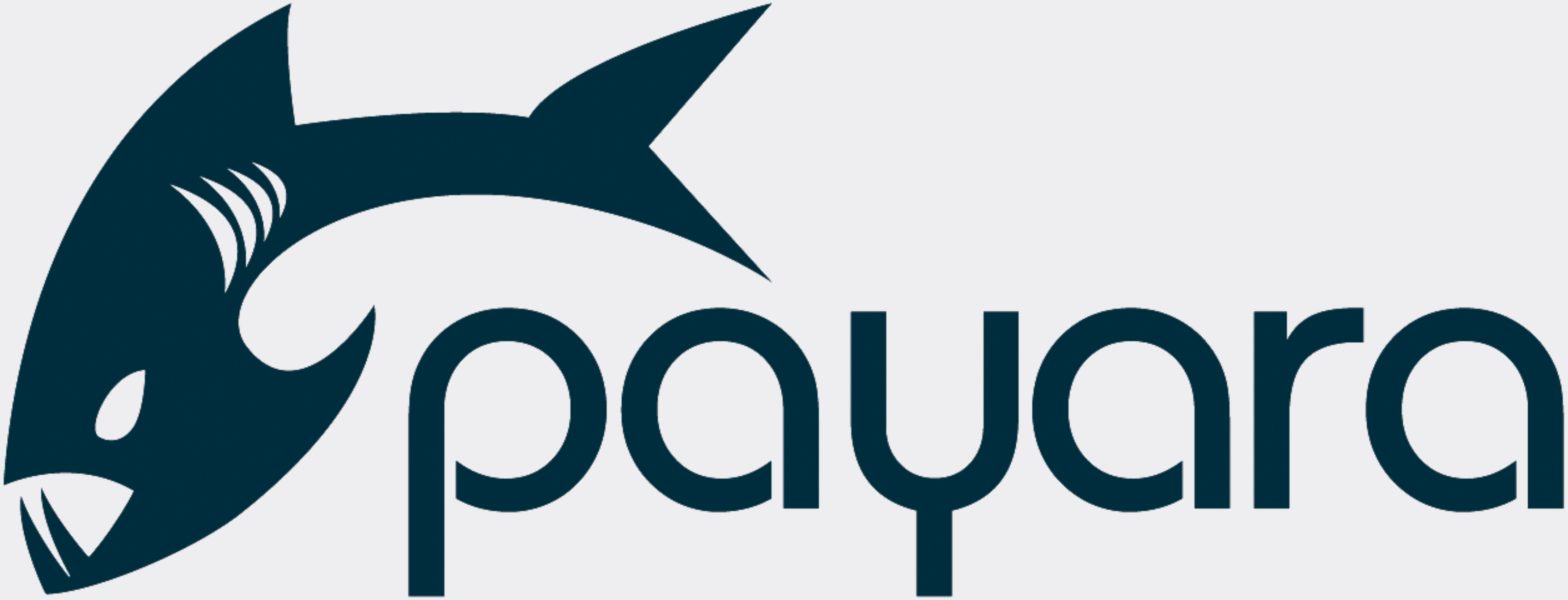
Java EE Microservice platforms - which is best?

Mike Croft – Payara Support Engineer
@croft | @Payara_Fish

The Best is?



Get Serious

- What is a Microservice?
- Features of the platforms
- Experiences in Use

What is a Microservice?

In computing, **microservices** is a software architecture style in which complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs.

Wikipedia

What is a Microservice?

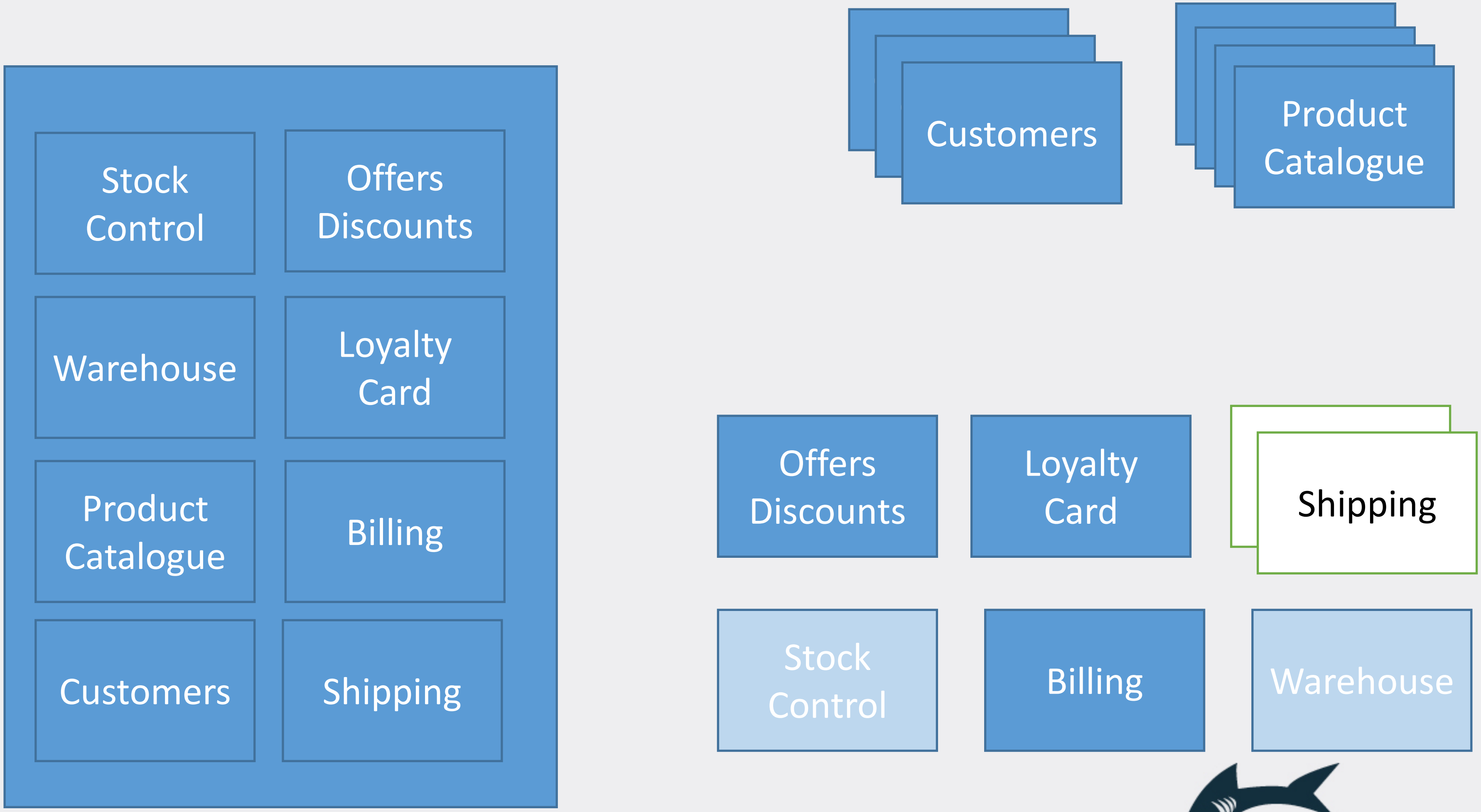
The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, ***automated deployment,*** *intelligence in the endpoints, and* ***decentralized control of languages and data****.*

Martin Fowler

Some Principles of MicroServices

- Componentization via Services
  - Independently replaceable and upgradable
- Organized around Business Capabilities
  - One team one capability across the stack
- Products not Projects
  - Team manages the whole lifecycle of the service(s)
- Independent Replacement and Upgradeability
- Infrastructure Automation
- Decentralized Data Management

Micro Service Architecture



FUD!

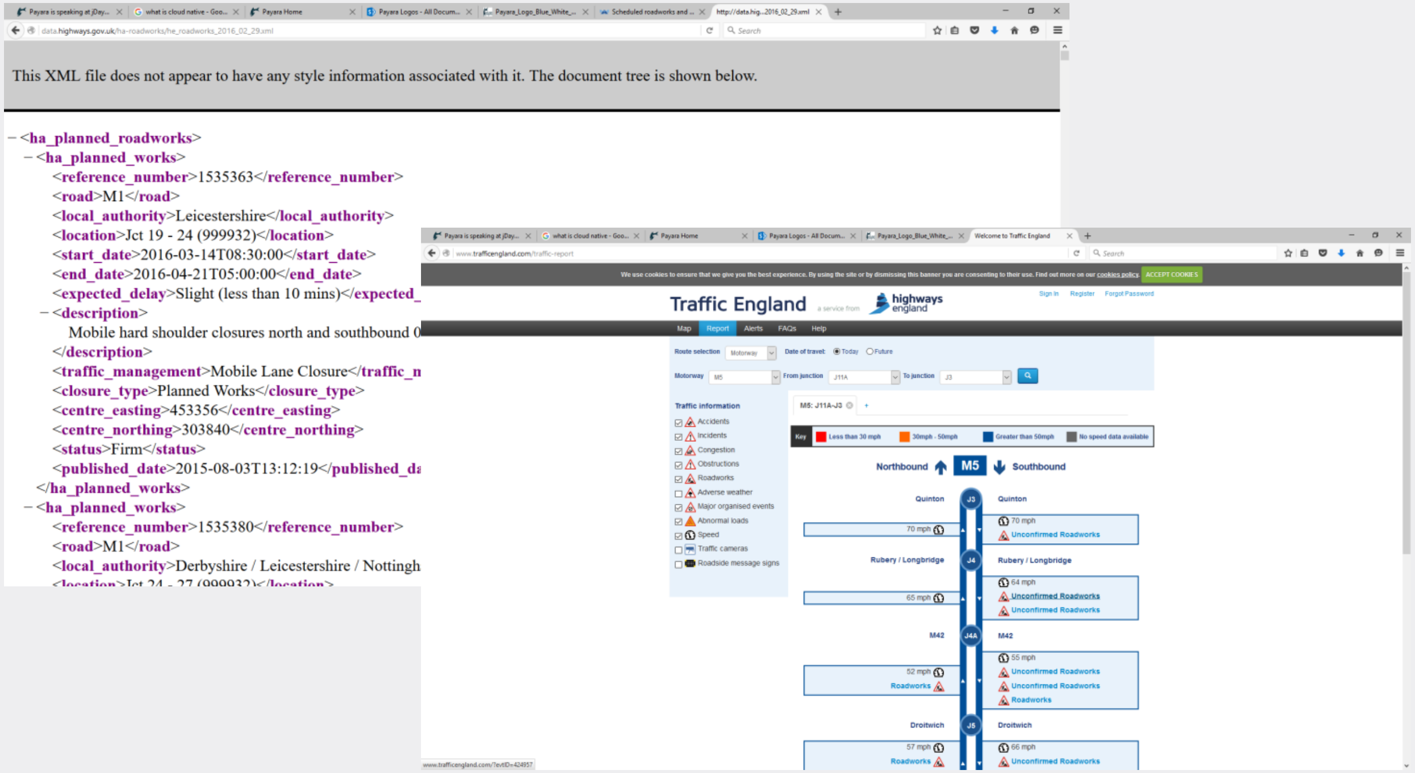
You can’t build microservice architectures with Java EE.

It is heavyweight and monolithic.

Why Java EE and not ######?

- Java EE is incredibly lightweight
- Rapid to develop
- Majority of alternative Java frameworks based on Java EE apis
- Concentrate on building business functionality
- Produces small skinny wars
- Versioned runtime – aids operations

Idea for Testing



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <ha_planned_roadworks>
  – <ha_planned_works>
      <reference_number>1535363</reference_number>
      <road>M1</road>
      <local_authority>Leicestershire</local_authority>
      <location>Jct 19 - 24 (999932)</location>
      <start_date>2016-03-14T08:30:00</start_date>
      <end_date>2016-04-21T05:00:00</end_date>
      <expected_delay>Slight (less than 10 mins)</expected_
    – <description>
        Mobile hard shoulder closures north and southbound 0
      </description>
      <traffic_management>Mobile Lane Closure</traffic_n
      <closure_type>Planned Works</closure_type>
      <centre_easting>453356</centre_easting>
      <centre_northing>303840</centre_northing>
      <status>Firm</status>
      <published_date>2015-08-03T13:12:19</published_da
    </ha_planned_works>
  – <ha_planned_works>
      <reference_number>1535380</reference_number>
      <road>M1</road>
      <local_authority>Derbyshire / Leicestershire / Nottingh
      <location>Jct 24 - 27 (999932)</location>
```

We use cookies to ensure that we give you the best experience. By using the site or by dismissing this banner you are consenting to their use. Find out more on our cookies policy. ACCEPT COOKIES

Traffic England a service from highways england

Sign In Register Forgot Password

Map Report Alerts FAQs Help

Route selection: Motorway — Date of travel: ● Today ○ Future

Motorway: M5 — From junction: J11A — To junction: J3

Traffic information

☑ Accidents
☑ Incidents
☑ Congestion
☑ Obstructions
☑ Roadworks
□ Adverse weather
☑ Major organised events
☑ Abnormal loads
☑ Speed
□ Traffic cameras
□ Roadside message signs

M5: J11A-J3

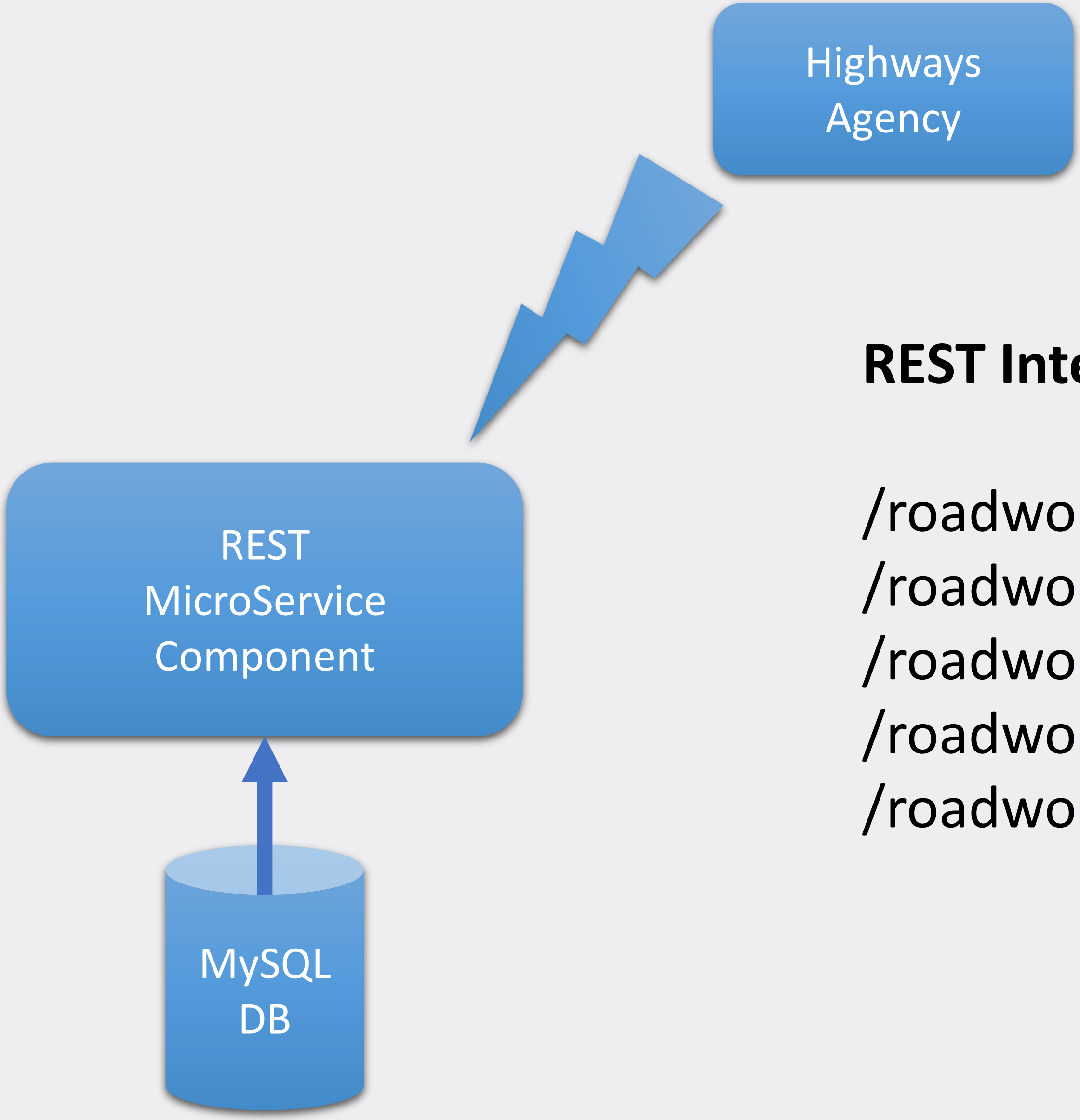
Key: 🟥 Less than 30 mph 🟧 30mph - 50mph 🟦 Greater than 50mph ⬛ No speed data available

Northbound M5 Southbound

Northbound	Junction	Southbound
Quinton	J3	Quinton
70 mph		70 mph Unconfirmed Roadworks
Rubery / Longbridge	J4	Rubery / Longbridge
65 mph		64 mph Unconfirmed Roadworks Unconfirmed Roadworks
M42	J4A	M42
52 mph Roadworks		55 mph Unconfirmed Roadworks Unconfirmed Roadworks Roadworks
Droitwich	J5	Droitwich
57 mph Roadworks		66 mph Unconfirmed Roadworks

Payara is speaking at jDay...
what is cloud native - Goo...
Payara Home
Payara Logos - All Docum...
Payara_Logo_Blue_White_...
Scheduled roadworks and ...
http://data.hig...2016_02_29.xml
Welcome to Traffic England
data.highways.gov.uk/ha-roadworks/he_roadworks_2016_02_29.xml
www.trafficengland.com/traffic-report
Search
www.trafficengland.com/?evtID=424957

http://data.highways.gov.uk/ha-roadworks/he_roadworks_2016_02_29.xml



REST Interface

/roadworks
/roadworks/now
/roadworks/{road}/now
/roadworks/{region}
/roadworks/{region}/now

Payara Micro

- Derived from GlassFish Open Source
- Key Design Goals
  - Elastic (auto clustering)
  - Container Friendly (no installation)
  - Small Footprint
  - Very Easy to Use (no maven messing)
  - Run wars on the command line

WildFly Swarm

- Derived from WildFly
- Flexible
  - Specify components with maven dependencies
- Generates Uber Jar
  - “BYOR”
- Integrates with other Red Hat projects
- Java EE based
- Integrates with Service Discovery

Spring Boot

- Encapsulates Spring
- Creates Standalone Spring Applications
- Embeds Runtime
  - Jetty, Tomcat, Undertow
- Opinionated Starter POMs
- Automatically Configures itself
- No XML
- Provides production ready capabilities

Code Walkthrough

Platforms Tested

Feature	Payara Micro	WildFly Swarm	Spring Boot
Composable	X	√	√
Requires Platform Knowledge	X	X	√
Needs Maven	X	√	√
File Size	60MB + 1MB war	100MB	26MB
Boot Time	4s	4s	4s
Memory (full GC)	34MB Meta 63MB	35MB Meta 83MB	44MB Meta 51MB
Embeddable API	√	√	√

payara

Payara Micro Thoughts

?

WildFly Swarm

- A little complexity in composing dependencies
  - Repeatedly failed for a while
  - Needed jaxrs-weld not jaxrs + weld!
  - After all that largest jar
- Small Footprint
- Fast Startup
- Small Memory Footprint

SpringBoot

- Very Simple to learn
- Marginally different from JavaEE
  - Slightly different REST annotations
- Small Footprint
- Fast Startup
- Smallest jar size but could grow depending on needs
- Slightly larger memory footprint
- Good composability
  - I didn’t have to figure it out

Conclusions

- No real advantage to assembling your container
  - …objectively speaking

- ALL Solutions
  - Lightweight
  - Small Footprint
  - Fast

- Choose what you are Familiar with
  - If you know Spring stick with Spring
  - If you know JavaEE, JavaEE does microservices.

- Java EE apps are cross container
  - Including datasource definition

Code

https://github.com/smillidge/JDays2016MicroServices

Questions

payara

New in 162…

- “Lite” Nodes
  - Hazelcast cluster members with NO local storage
- Payara Micro will cluster with Payara Server
- Updated Docker images
- Much more to be announced very soon