RED HAT
OPEN SOURCE DAY
Europe, Middle East & Africa

redhat.

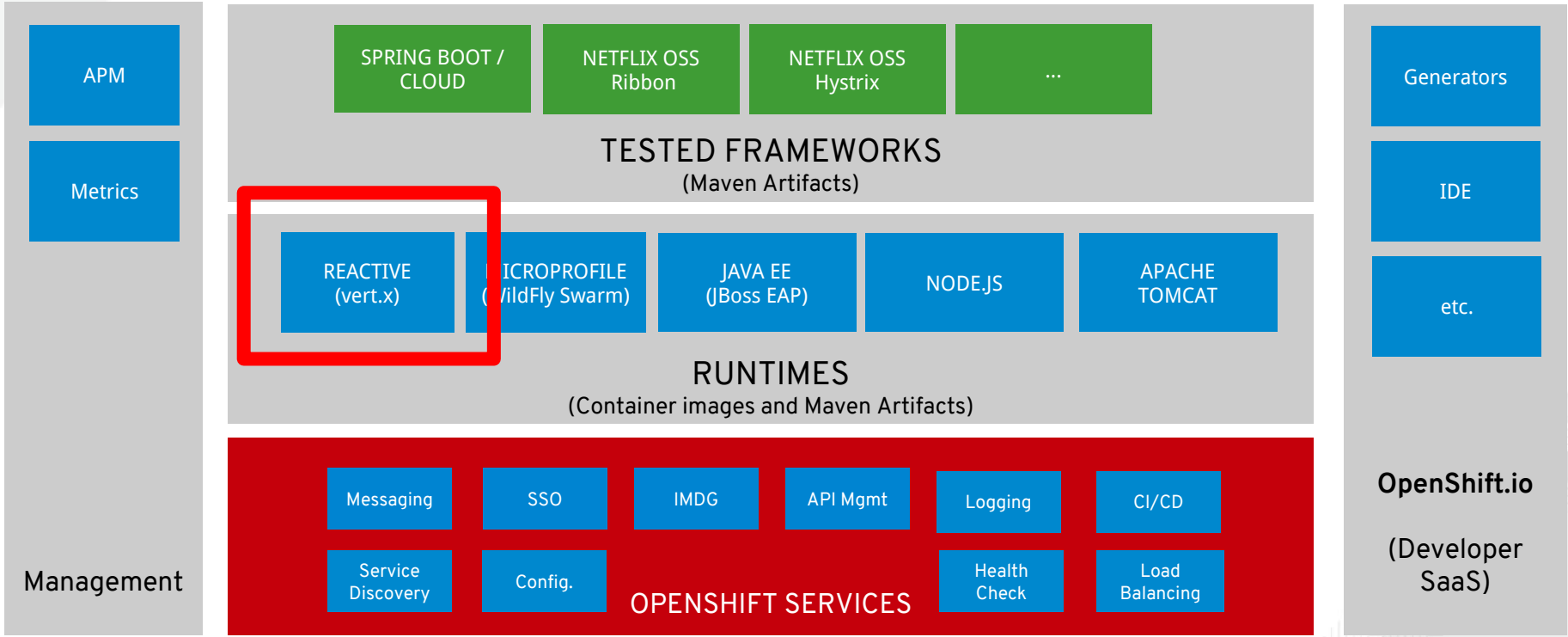
VERTX.IO: REACTIVE MICROSERVICES

EAP and beyond: Red Hat Open Application Runtimes

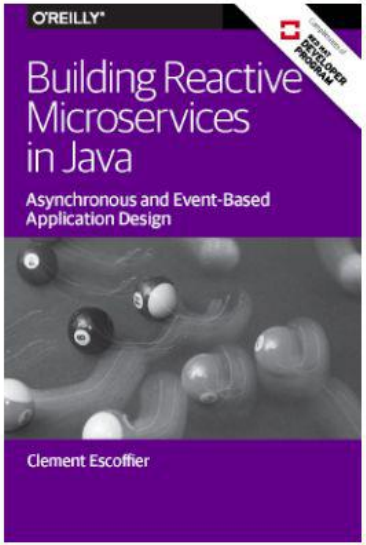
Ugo Landini

Solution Architect

RHOAR: OpenShift Application Runtimes



VERT.X



Building Reactive Microservices in Java

Investigating distributed microservices? Want to get rid of your monolithic enterprise applications or not create new ones? Reactive design can help. Author and Red Hatter, Clement Escoffier, explains why and how Eclipse Vert.x is a good choice to build effective microservices systems.

In this O'Reilly book learn how:

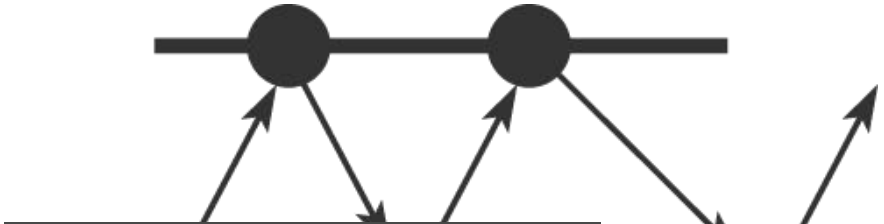
- Explore the elements of reactive microservices and learn how Vert.x works.
- Build and consume a single microservice to understand how messaging improves its reactiveness.
- Create an entire microservices system, using stability and resilience patterns to manage failures.
- Use the OpenShift container platform to deploy and manage microservices in a virtual or cloud environment.

Sign in or join now (it's free) to download the full book.

Free from : https://developers.redhat.com/

VERT.X

- **Reactive Microservices Toolkit** for the JVM
- Polyglot - **Java, JavaScript**, jRuby, Python, Groovy, Scala, Kotlin
- **Distributed Event Bus** for lightweight messaging
- Event Driven **Non-Blocking I/O**
- Ideal for high concurrency, low latency applications / services
- 2014 JAX Innovation Awards Winner



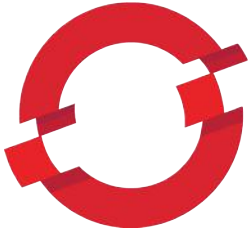
```
import io.vertx.core.AbstractVerticle;
public class Server extends AbstractVerticle {
  public void start() {
    vertx.createHttpServer().requestHandler(req -> {
      req.response()
        .putHeader("content-type", "text/plain")
        .end("Hello from Vert.x!");
    }).listen(8080);
  }
}
```

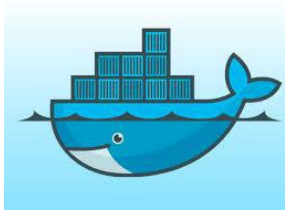
How to select the runtimes

Selection Consideration	Project Type				Framework Pref		Learning Effort			Deployment Pkg		
Runtimes	Cloud Native (new)	Cloud Enable (existing)			Java EE	Non-Java EE	No	Little	Invest	Thin	Fat	Hollow
		Lift & Shift	Connect & Enhance	Refactor & Rewrite								
EAP	+	+	+	+	+		+			+		
Swarm	+		+	+	+	+	+	+		+	+	+
Vert.x	+		+	+		+			+	+	+	
Node.js	+		+	+		+			+	+		
Tomcat	+	Spring Boot	+	+		+	+			+	Spring Boot	

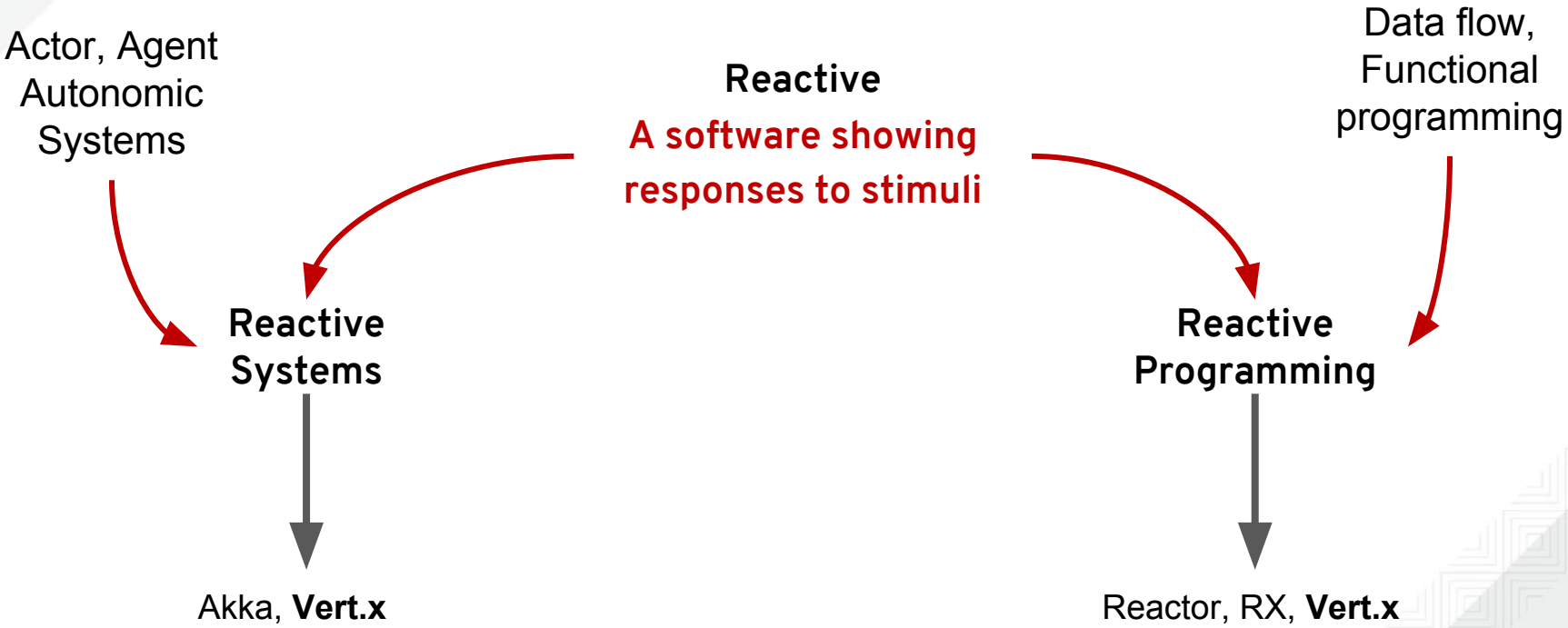
Why Reactive?

Apps In The Past	Apps Today
Single/Few Machines	Clusters of Machines
Single/Few Core Machines	Multicore Machines
Expensive RAM	Cheap RAM
Expensive Disk	Cheap Disk
Slow N/W	Fast N/W
Few Concurrent Users	Many Concurrent Users
Small Data Sets	Large Data Sets
Latency in secs	Latency in ms





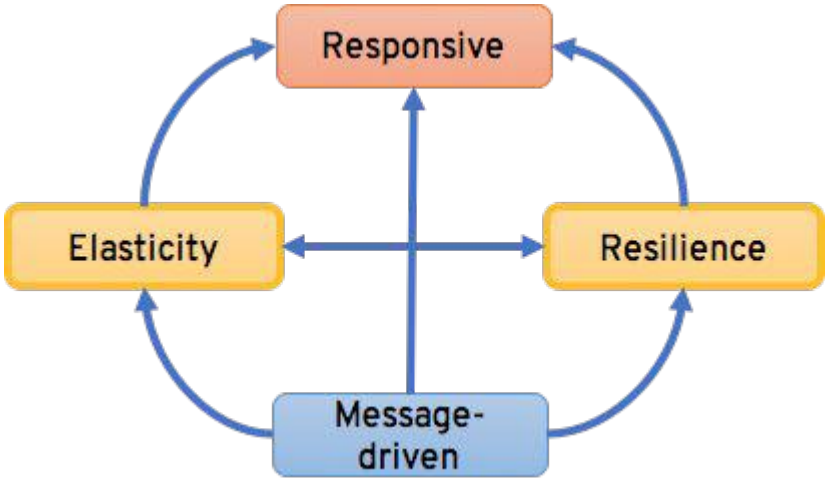
The 2 faces of Reactive



Reactive Manifesto

http://www.reactivemanifesto.org/

Reactive Systems are an architecture style focusing on **responsiveness**

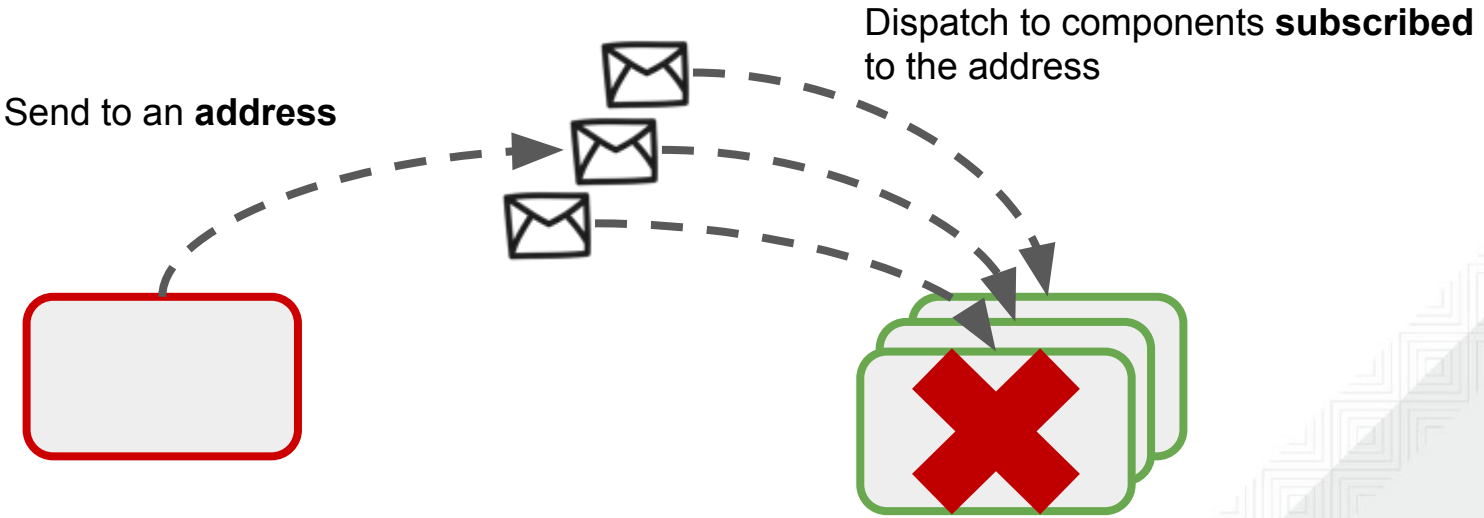


Asynchronous message passing => Elasticity

Components interacts using **messages**

Messages allows **elasticity**

Resilience is not only about failures, it's also about **self-healing**



Pragmatic Reactive systems

And that's what Vert.x offers to you

Development model => Embrace **asynchronous**

Simplified concurrency => **Event-loop**, not thread-based

I/O

- **Non-blocking I/O**, if you can't isolate
- HTTP, TCP, RPC => Virtual address
- Messaging

Asynchronous development models

Async programming

- **Exists since the early days of computing**
- Better usage of hardware resource, avoid blocking threads

Approaches

- **Callbacks**
- **Future / Promise (single value, many read, single write)**
- **Data streams - Reactive Programming**
- Data flow variables (cell)
- *Continuation*
- *Co-Routines*

Reactive Architecture / Software

Application to software

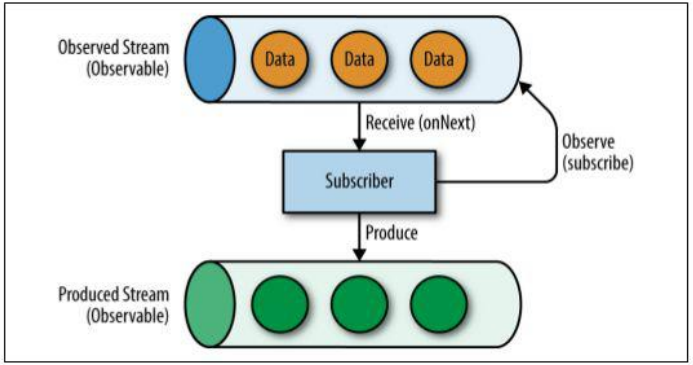
A software showing responses to stimuli

- Events, Messages, Requests, Failures, Measures, Availability…
- The end of the flow of control ?

Is it new?

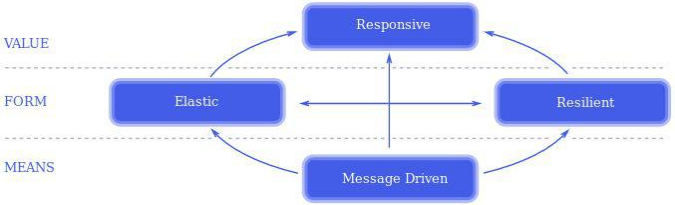
- Actors, Object-oriented programming…
- IOT, Streaming platform, complex event processing, event sourcing…

Building Reactive Systems



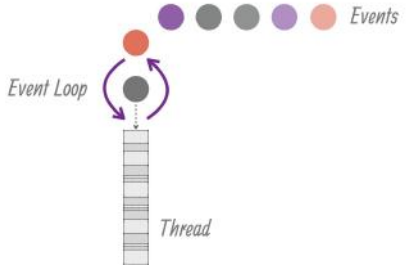
- Message Driven
- Microservices
- Tooling (OCP, Kubernetes)

Reactive Systems

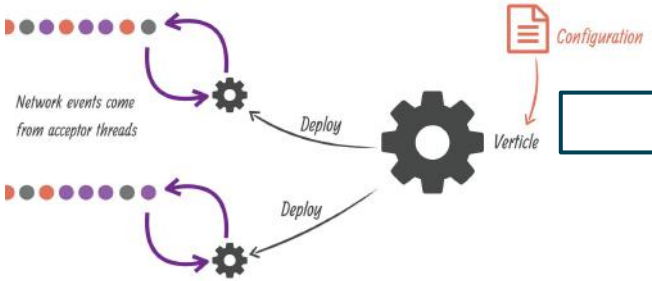


https://www.reactivemanifesto.org/

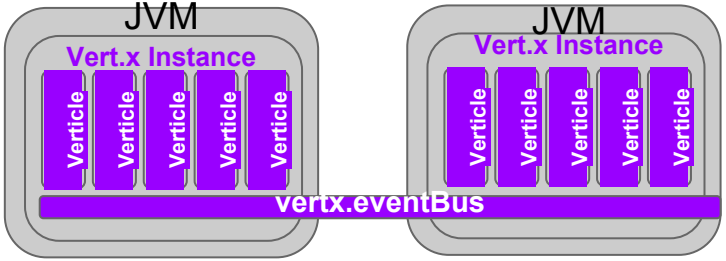
Vert.x

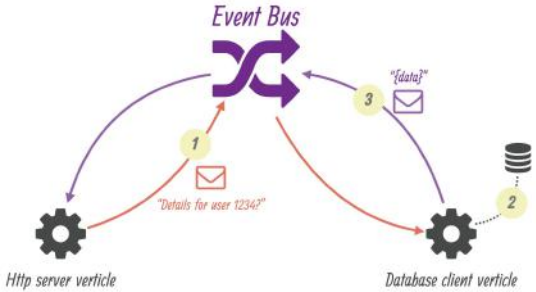


Event Loop



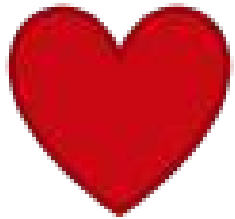
Multi-Reactor





All you need is (reactive) love

Reactive Systems + Reactive Programming =



QUESTIONS?

RED HAT
OPEN SOURCE DAY

Europe, Middle East & Africa