



RED HAT DEVELOPERS

Reactive Microservices with Eclipse Vert.x

@burrsutter

vertx.io

<http://developers.redhat.com>

<http://bit.ly/reactivems>

Change History

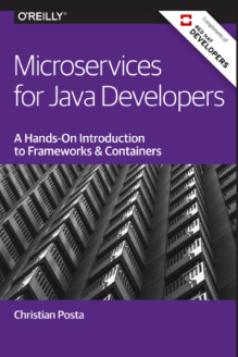
0.1 - Forked from bit.ly/helloworldmsa

1.0 - JavaOne 2016

2.0 - <http://javaday.org.ua/kyiv/>

2.1 - Added “Be This Tall” and 2 Gene Kim books slides

2.2 - Changed color scheme/art on several slides



MICROSERVICES FOR JAVA DEVELOPERS:

A hands-on introduction to frameworks and containers.

[DOWNLOAD NOW](#)

READ MORE ON MICROSERVICES

- ➡ Tear Down Data Silos with Microservices
- ➡ Different types of Microservices?
- ➡ Scalable Microservices through messaging



Join Red Hat Developers and try it now



Start using .NET on Linux today



Event Recap

Didn't make it to DevNation? Watch Sessions OnDemand



OpenJDK from Red Hat for Windows and you.

Join Red Hat Developers and try OpenJDK



MongoDB Shell Cheat Sheet

Getting started, collections, indexes, and dangers. Download now.



kubernetes



389 project | aeolus-configure | anaconda | attr | Audrey | Augeas | AuthHub | Autofs | Beaker | Boxes | Btrfs | CacheFS | Cairo | candlepin | certmonger | CIFS | Cluster 3 | cobbler | colord | Completely Fair Scheduler | Conductor Control Group Configuration Library | Corosync | CRASH | Crypto API | Cumin | Cygwin | D-Bus | Dashbuilder | Data Grids for the Java Platform | device-mapper | DeviceKit | DistributionUtilities | dlm | Dogtag | DPDK | Dracut | Editline Library EDS | EJB 3 | eutils | eutils | Embedded Jopr | Evolution | Evolution-exchange | eXo JCR | ext3 | ext4 | fence-agents | fence-virt | Flannel | fontconfig | func | Gamin | Gatein Portal | gcc | gcc | gci | gdb | GFS | GFS2 | glib | gss-proxy GTK+ | gvfs | gzip | HTCondor | ID Utils | imagefactory | IPsec-Tools | iptraf-ng | Jandex | Java | JGroups | Jheadline | JRuby | JSFUnit | jUDDI | Katello | Kerberos | Kimchi | Koji | libguestfs | libibverbs | libminidump | libnotify | libqb libstoragemgmt | libuser | libvirt | libxml2 | libxstl | Linux Audit | Linux Infiniband Project | Linux Kernel | Linux PAM | linux-nfs | Luci | LVM2 | LVM2 | lynx | ManageIQ | Matahari | Maven Integration for Eclipse | Maven Integration for Web Tools Platform | Mock | mod_cluster | moVirt | nautlius | netfilter | NetworkManager | NFS-Ganesha | nmap | Open vSwitch | opennais | OpenDaylight | OpenJDK | OpenSSH | OpenSSL | OPNFV | OProfile | oVirt | oVirt Engine | oVirt Node | OVMF | Oz pacemaker | PackageKit | PackageKit | pango | PAPI | Paradyn | patchutils | Performance Co-Pilot | PicketBox | PicketLink | piglit | pixman | polkit | Polkit Qt | Poppler | Portlet Bridge | PostgreSQL | PressGang psmisc | Pulp | PulseAudio | pungi | pynfs | QEMU | Red Hat Update Agent | RESTEasy | rgmanager | RHQ | rpm | rpmgrill | RSYSLOG | Samba | Samba | SETroubleshoot ShrinkWrap | slapi-ns | Smokestack | Spacewalk Spherical | Spice | Spice-gtk | Spice-protocol | Stilts | suds | SWI Prolog | SwitchYard | syslog-ng | System Security Services Daemon | systemd | SystemTap | The Foreman | Thin Crust | tin | UberFire | udev udisks | udisks | UPower | util-linux | Valgrind | vc-dwm | vdagent | vdsm | vfs | Virt-clone | Virt-image | Virt-Install | Virt-manager | Virt-viewer | Virtual memory manager | Wallaby | Wayland | Wise | XFS | xinut | XNIO | Xorg | XrandR | yum

Developer Evolution :-)

70s



COBOL
JCL
WFL

80s



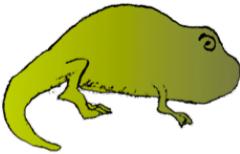
C/C++
4GLs
CASE
Netware
RDBMS
SQL
Unix

90s



HTTP/HTML
CGI
GET/POST
Cookies
Java
Servlet
EJB
Windows NT
Solaris/AIX
OOP

00s



MVC - Struts
DI - Spring
ORM - Hibernate
XML
WS-*
JSF
AJAX
Agile
Automated Testing
CI
SVN
Linux

10s



HTML5
Mobile
iOS/Android
Maven/Gradle
Git
Node.js
MongoDB/Redis
Hadoop
*-aaS/Cloud
Reactive
Functional



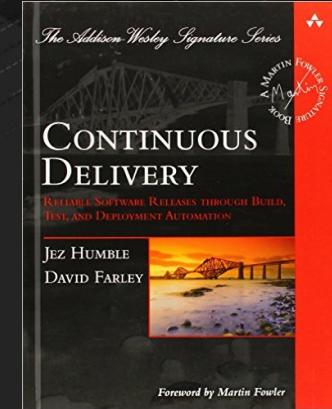
Mythical Beast

You Must Be This Tall

1. Self-Service, on-demand, elastic infrastructure as code
(how many days/weeks to provision a new VM?)
2. Dev vs Ops
(who is on the pager for production app outage?)
3. Automation
(phoenix vs snowflake?)
4. CI & CD



Jez Humble Continuous Integration



1. Software (trunk) is always deployable
2. Everyone is checking into trunk daily (at least), not feature branches
3. If the build breaks it is fixed in 10 minutes (all hands on deck)
4. A new engineer can be on-boarded in 1 day - with a production-like environment on the developer workstation
5. Deployment is a low-risk push button affair
6. NOT just about using Jenkins

The Phoenix Project

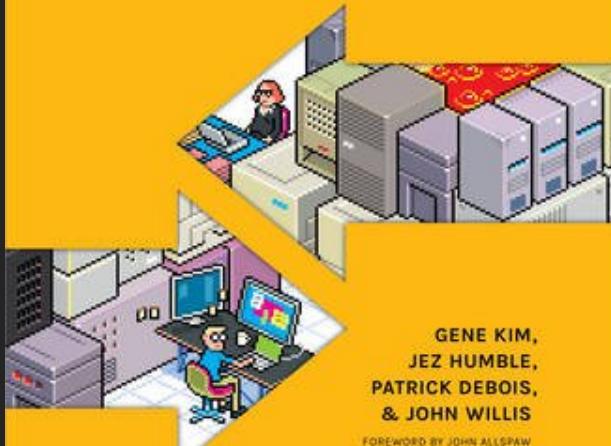
A Novel About IT, DevOps, and Helping Your Business Win

Gene Kim, Kevin Behr and George Spafford



The DevOps Handbook

HOW TO CREATE WORLD-CLASS
AGILITY, RELIABILITY, & SECURITY
IN TECHNOLOGY ORGANIZATIONS



GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS

FOREWORD BY JOHN ALLSPAW

TAKE THE DORA DEVOPS X-RAY ASSESSMENT AND SEE WHERE YOU STAND.



fabric8

fabric8.io

fabric8-console

192.168.64.10:31776/workspaces/default/forge/command/project-new

fabric8 Teams default Create Project James

Create a new project

Project Type

Django	Maven Archetype	Funktion	Generic	Go	Integration
java Enterprise Archive (EAR)	Java Library (JAR)	Java Web Application (WAR)	Microservice	NodeJS	Quickstart
Rails	Spring Boot	Swift	VERT.X	WildFly Swarm	

Choose the type of the project to create

Named

Top level package

Version

Cancel Next >

The screenshot shows the 'Create a new project' interface in the fabric8 console. At the top, there are tabs for 'Teams', 'default', and 'Create Project'. Below the tabs, the title 'Create a new project' is displayed. A section titled 'Project Type' contains a grid of 18 project icons. The 'Integration' icon (a camel) is highlighted with a blue border. Below the grid, there's a section titled 'Choose the type of the project to create' with three input fields: 'Named' (containing 'awesome'), 'Top level package' (containing 'org.example'), and 'Version' (containing '1.0.0-SNAPSHOT'). At the bottom right are 'Cancel' and 'Next >' buttons.

fabric8-console

192.168.64.10:31776/workspaces/default/projects/awesome/forge/command/devops-edit James

fabric8 Teams default Apps awesome Settings

Dashboard Pipelines Builds Metrics Settings Source Camel Forge

Core Secrets Pipeline Maven

Edit the DevOps configuration for this project

Pipeline

... BuildImage Stages canary image

... CanaryRelease Stages canary image integration test Environments Testing

... CanaryReleaseAndStage Stages canary image integration test Rolling upgrade Staging Environments Testing Staging

... CanaryReleaseStageAndApprovePro Stages canary image integration test Rolling upgrade Staging approve Rolling upgrade Production Environments Testing Staging Production

... Deploy Stages deploy

... Install Stages install

Copy pipeline to project
Should we copy the pipeline definition (in the Jenkinsfile) into the project source code

Chat room

Issue project name

Name of the issue tracker project

Code review
Enable code review of all commits

Cancel Next >

fabric8-console

192.168.64.10:31776/worksaces/default/projects/awesomecamel

fabric8 Teams default Apps awesomecamel Dashboard

Dashboard Pipelines Builds Metrics Settings Source Camel Forge

Environments

Testing Staging Production

awesomecamel-1: 1.0.1 Build #1 d0232ea Docs

awesomecamel-1: 1.0.1 Build #1 d0232ea Docs

Production

Active Pipelines [View All Pipelines >>](#)

Build #1 Approve | Started 6 seconds ago

Canary Release Integration Testing Rolling Upgrade Staging Approve

58 seconds 46 seconds 0 seconds 0 seconds

Logs [View Full Log](#)

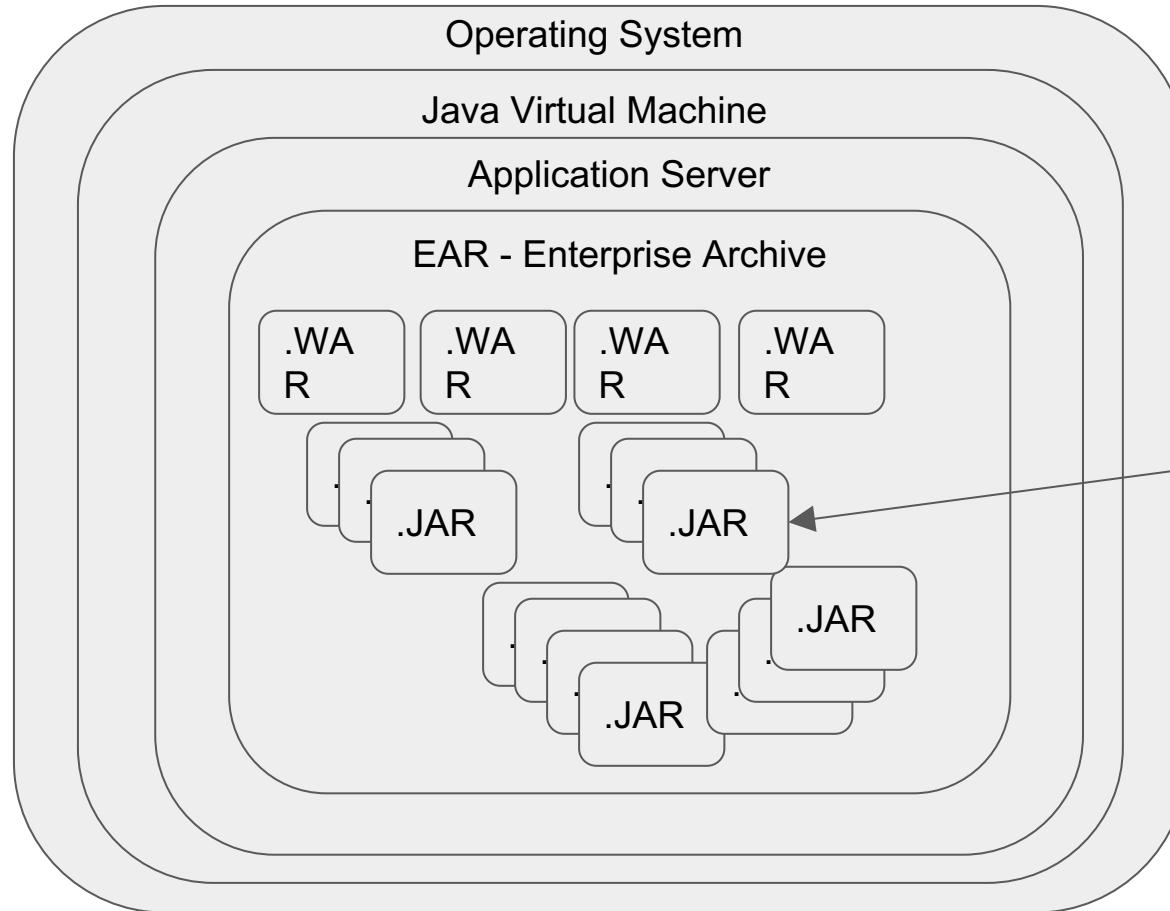
```
Would you like to promote version 1.0.1 to the Production namespace?  
to Proceed reply: fabric8 jenkins proceed job awesomecamel build 1  
to Abort reply: fabric8 jenkins abort job awesomecamel build 1  
No service hubot is running!!!  
No service found!  
[Pipeline] approveRequestedEvent  
Unable to connect to Elasticsearch service. Check Elasticsearch is running in the correct namespace  
[Pipeline] input  
null
```

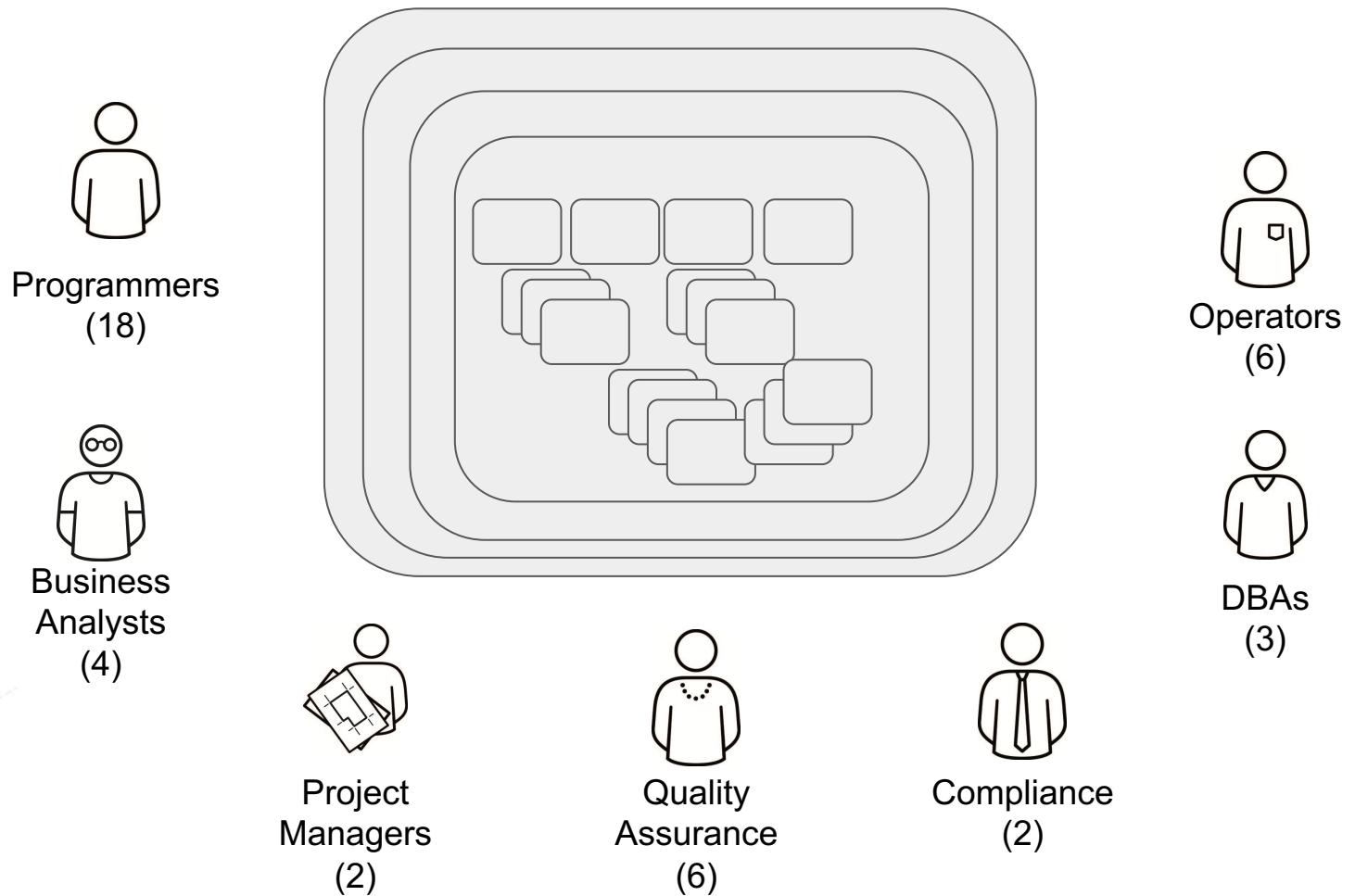
Proceed or Abort

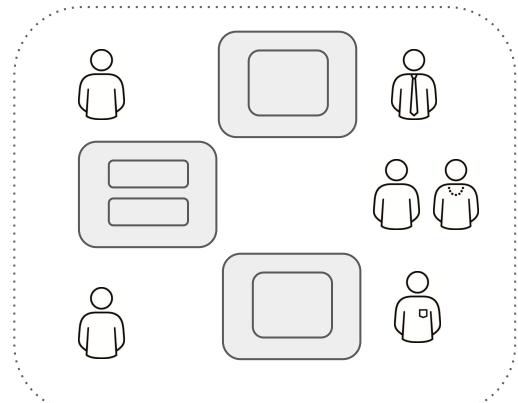
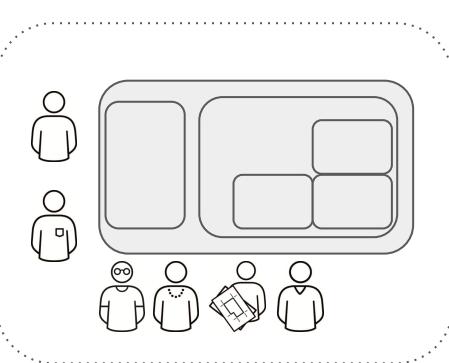
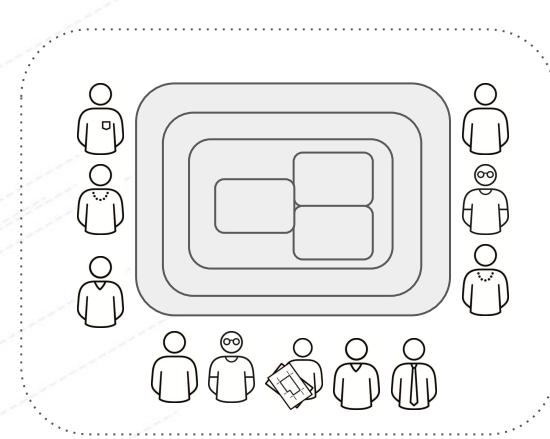
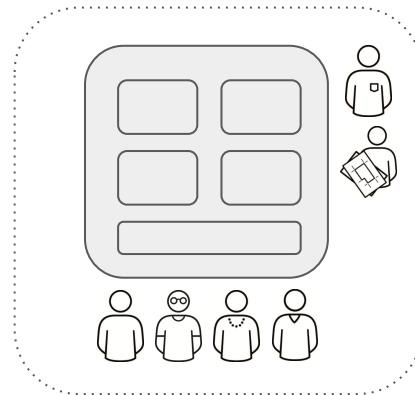
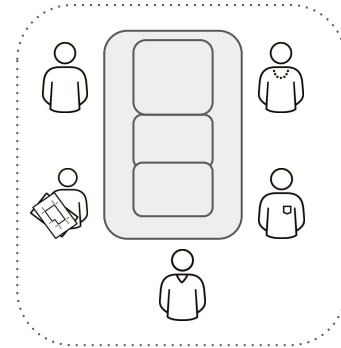
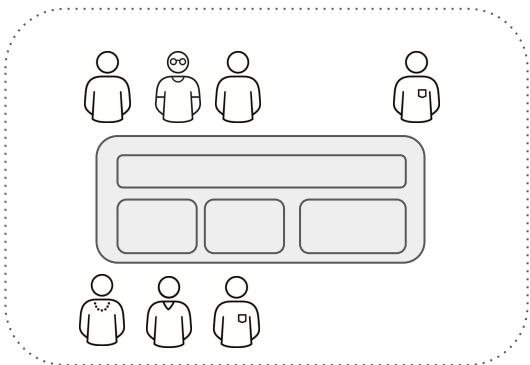
Commits [View All Commits >>](#)

devops-edit --copyPipelineToProject=true --pipeline=workflows/maven/CanaryReleaseStageAndApprovePromote/Jenkinsfile
gogsadmin committed d0232ea

index on master: 2448d85 project-new --type=integration --topLevelPackage=org.example --version=1.0.0-SNAPSHOT --
targetLocation=/tmp/fabric8-forge/user/nogosadmin--stack=None--named=awesomecamele 5f60107







Microservice Principles/Characteristics

1. Deployment **Independence**: updates to an individual microservice have no negative impact to any other component of the system. Optimized for **Replacement**.
2. Organized around **business** capabilities
3. **Products** not Projects
4. **API-Focused**
5. **Smart** endpoints and dumb pipes
6. Decentralized Governance
7. Decentralized Data Management

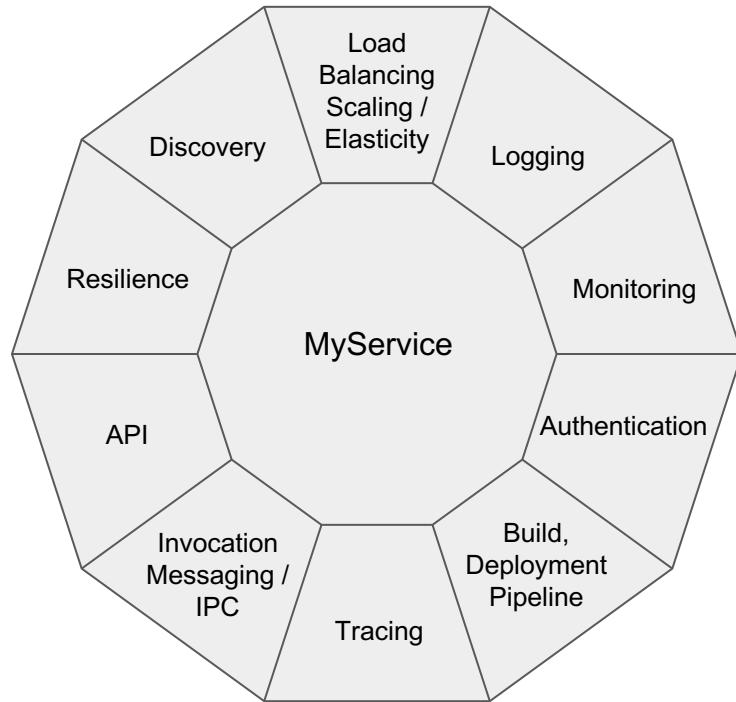
Decentralized **Data** Management

WTF?

Stay tuned - new book

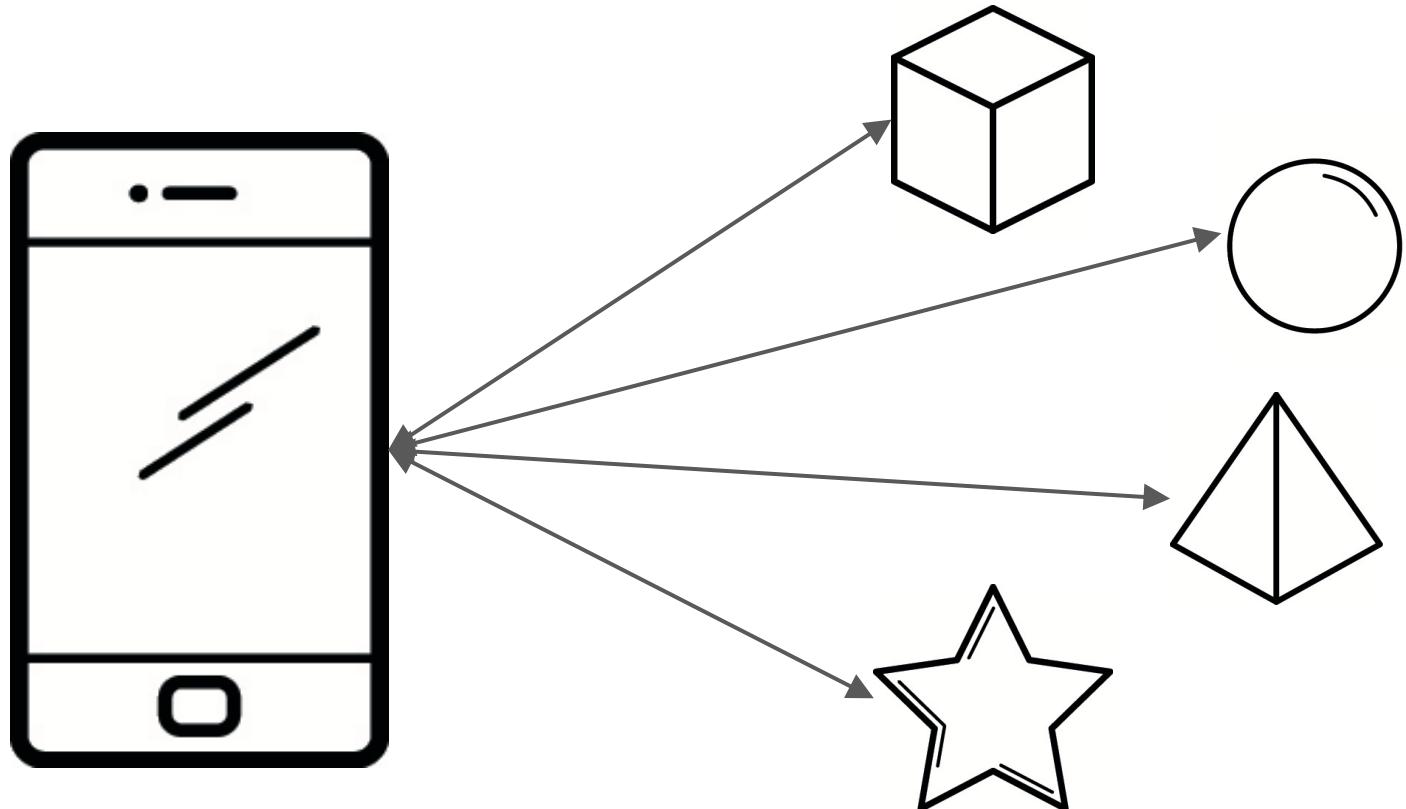
arriving at <https://developers.redhat.com> late Feb 2017

Microservice Concerns



Microservice Patterns

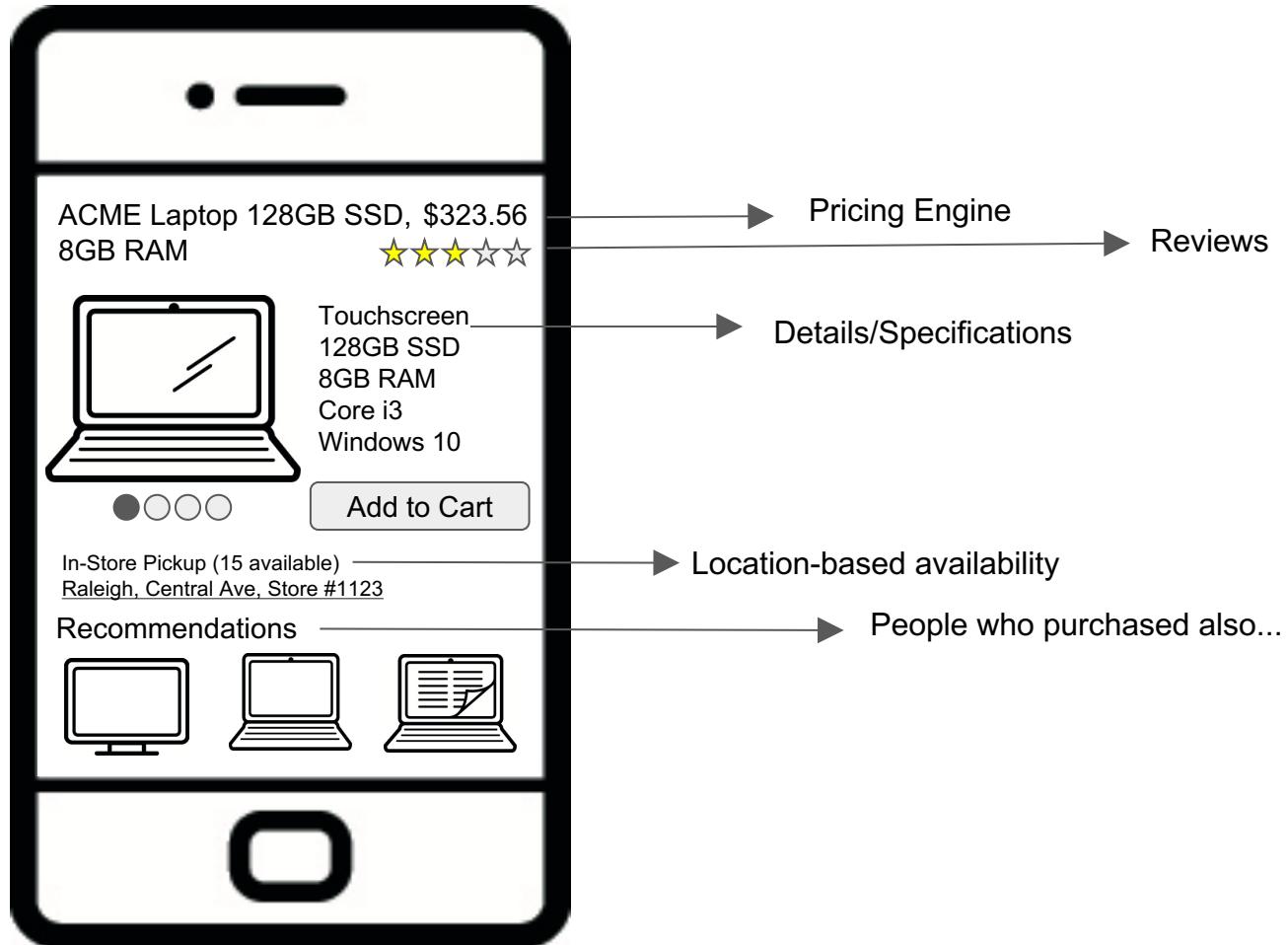
Browser



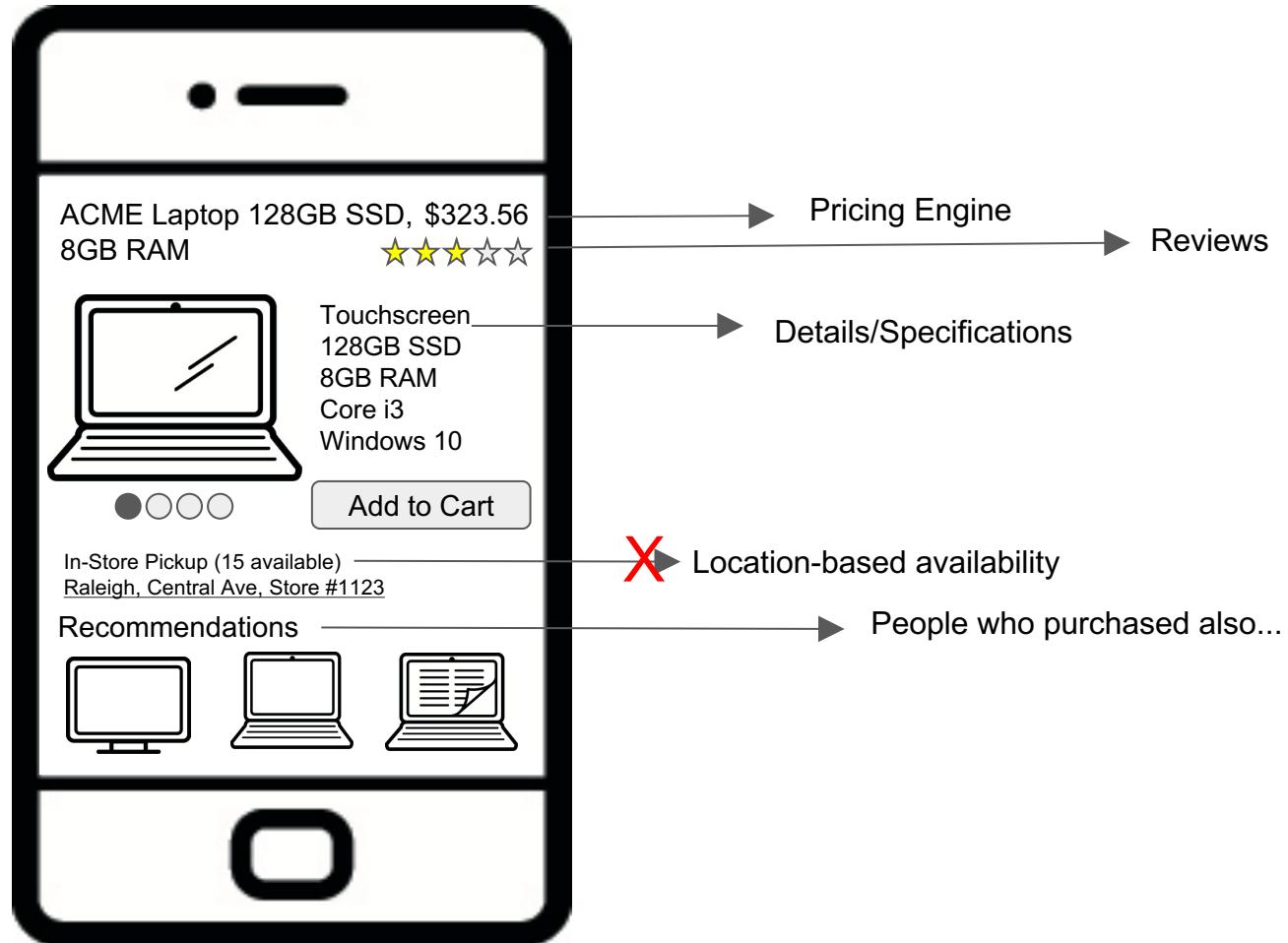
Example



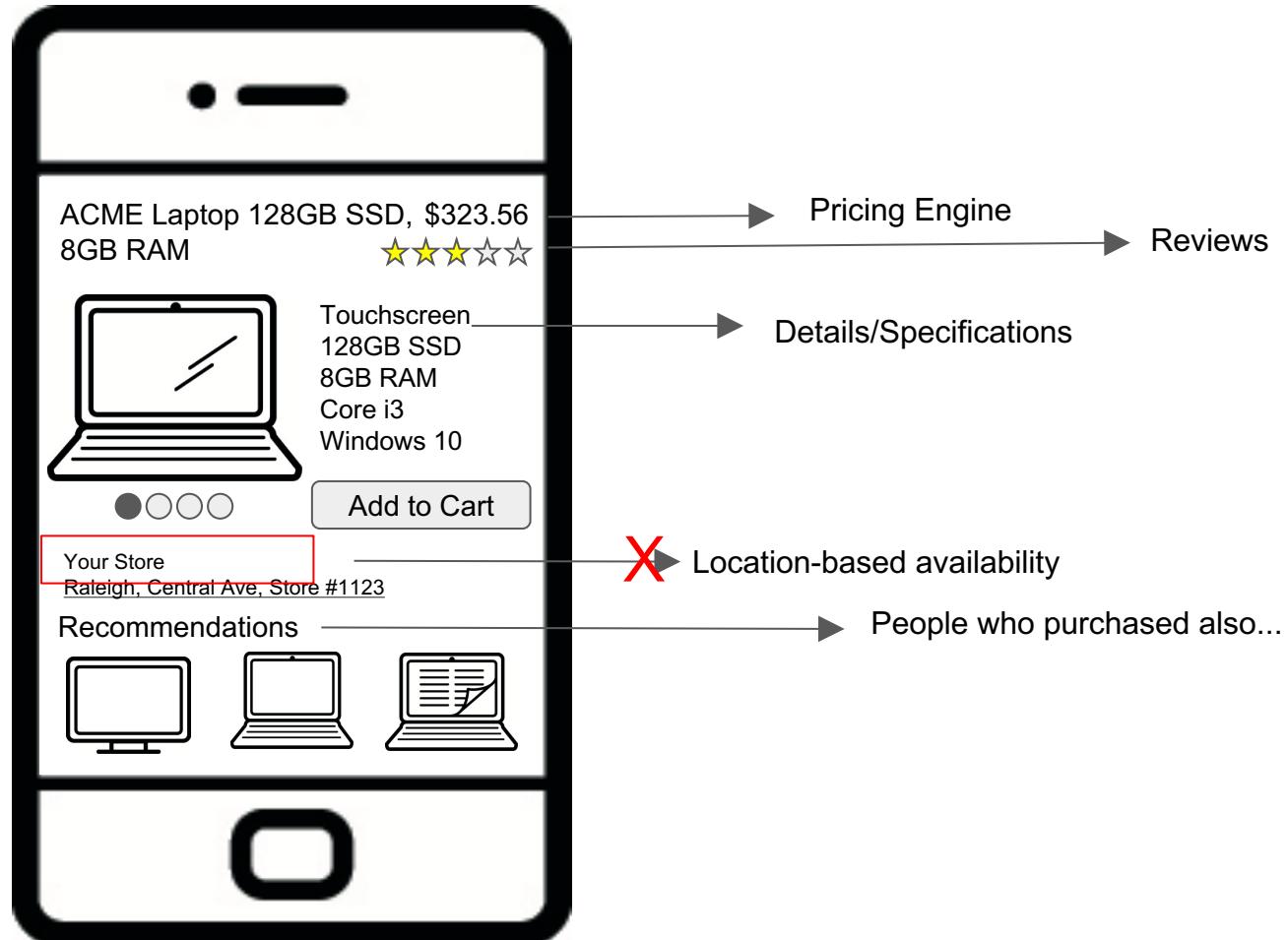
Example



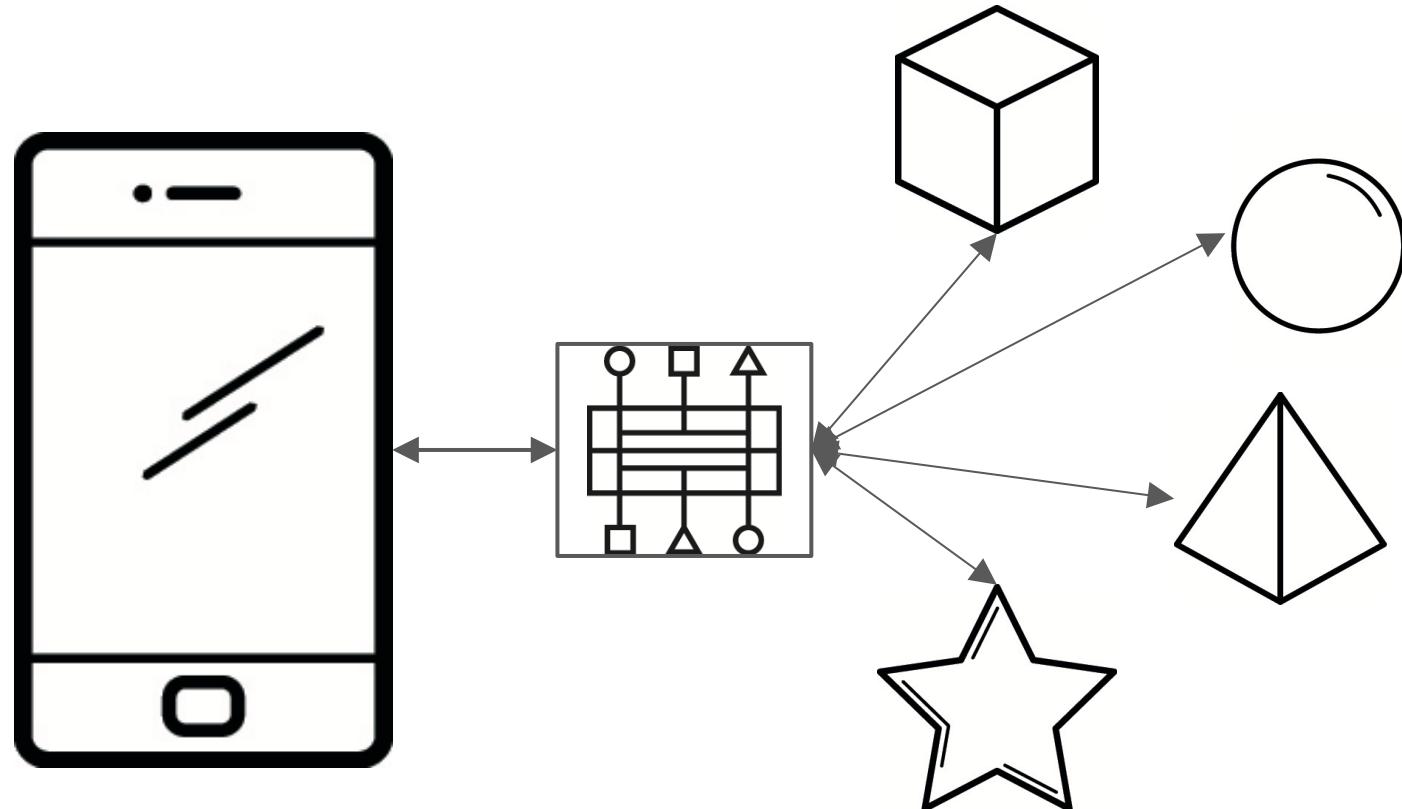
Example (Fail)



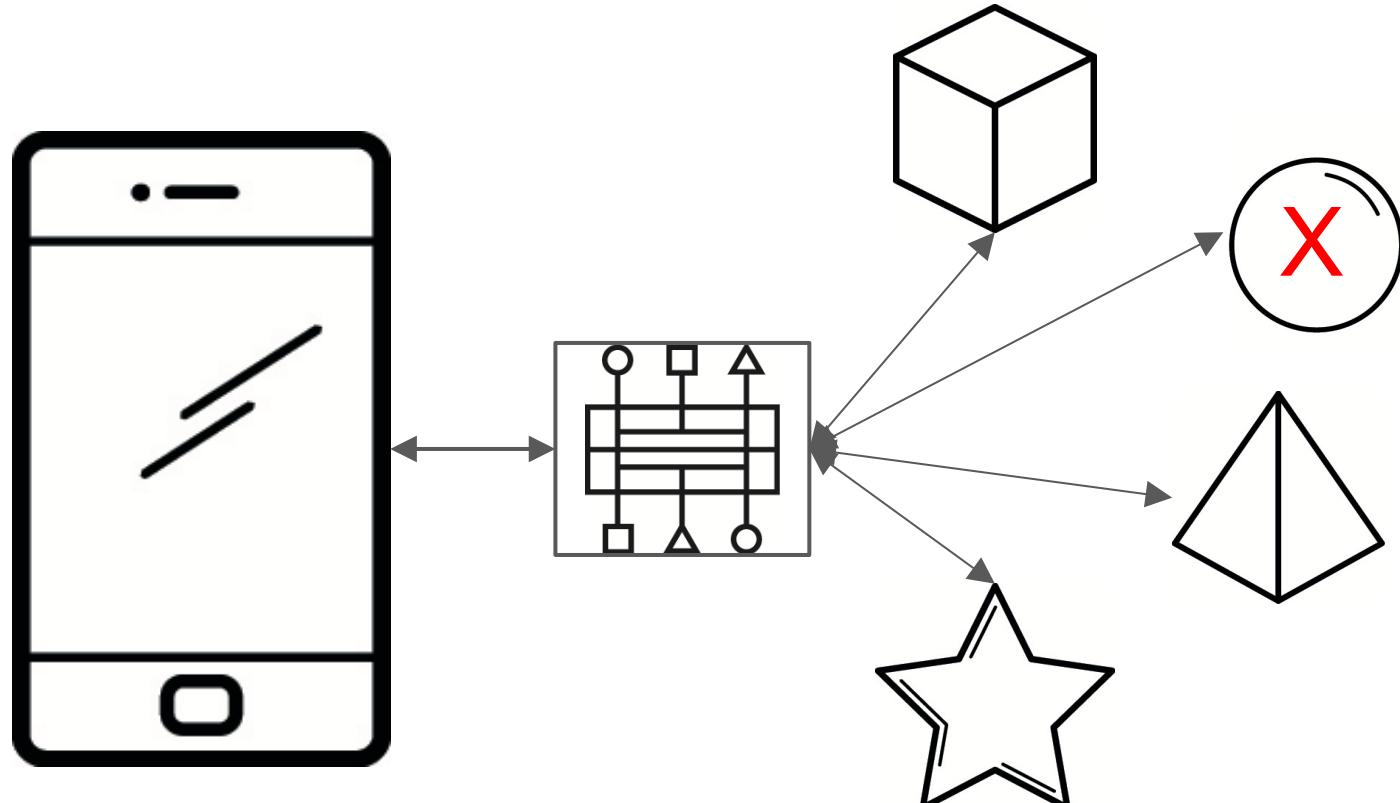
Example (Fallback)



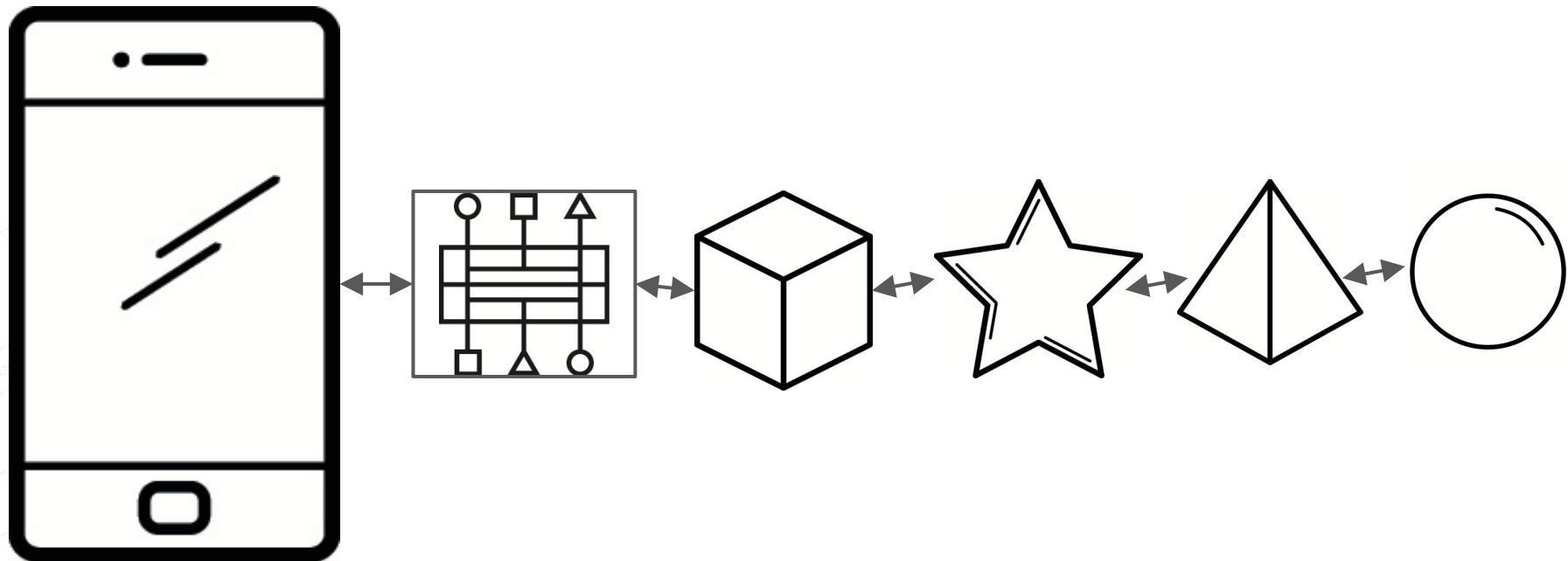
API “Gateway”



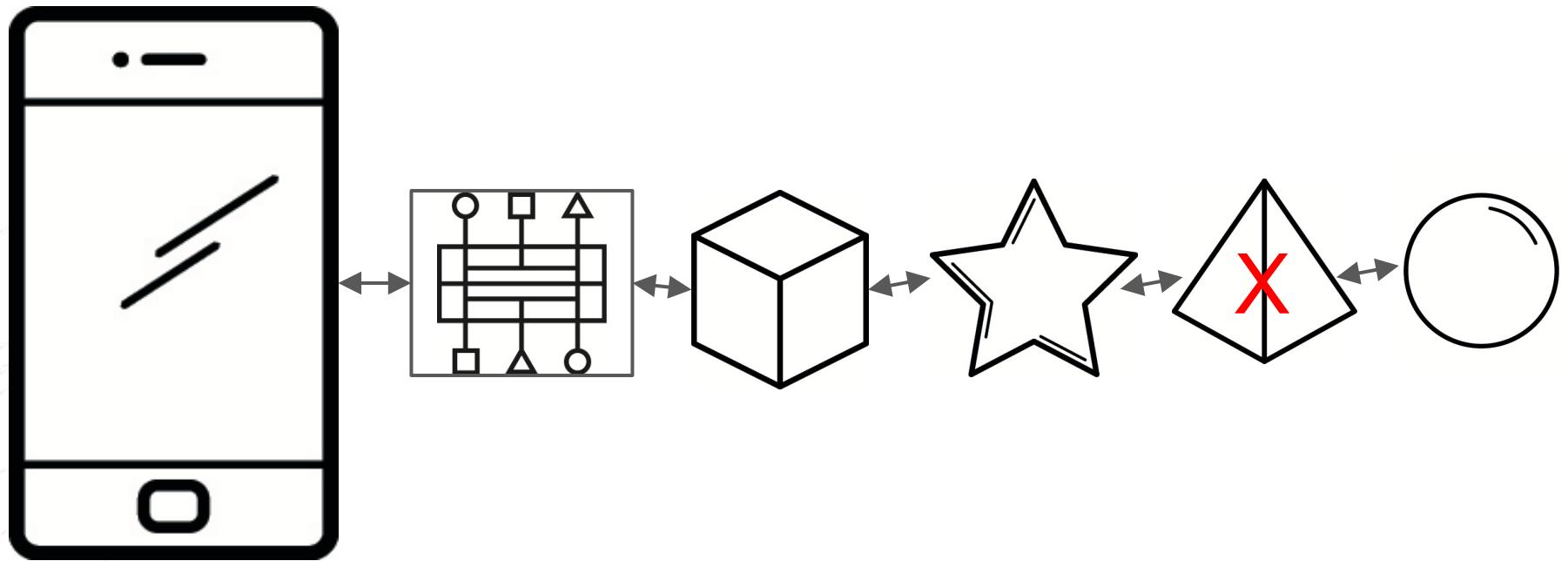
API “Gateway” (Fail)



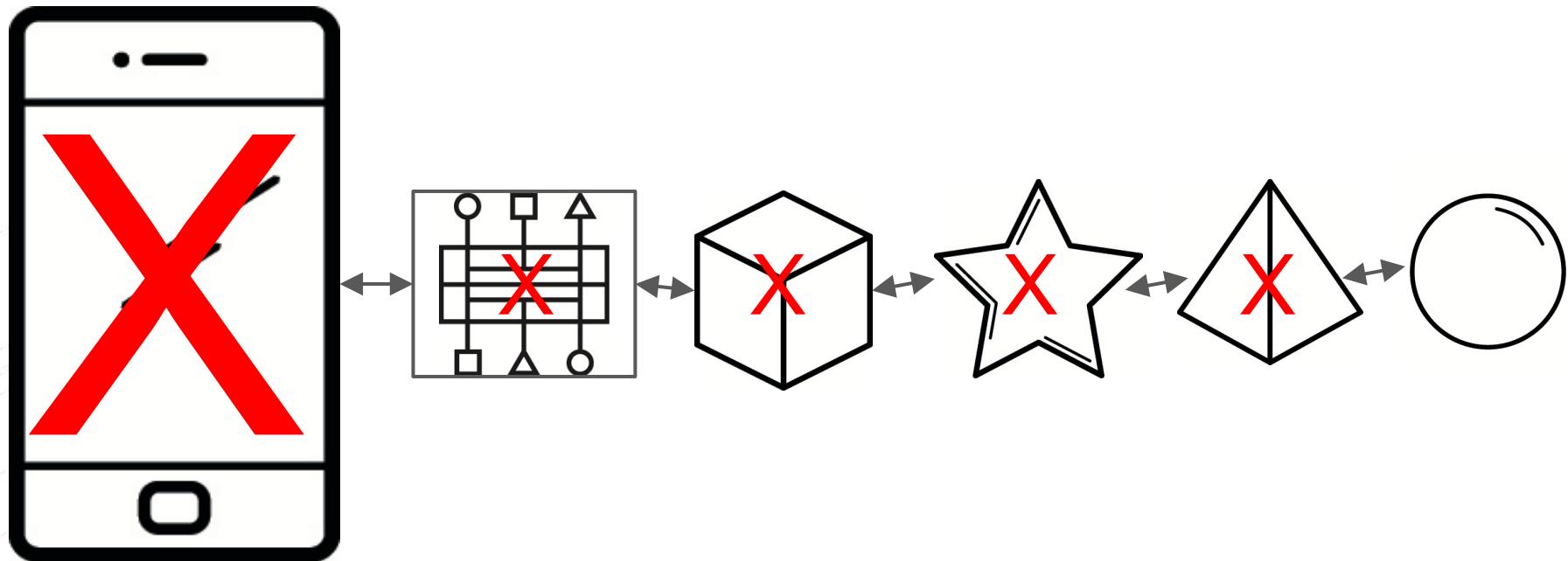
Chaining



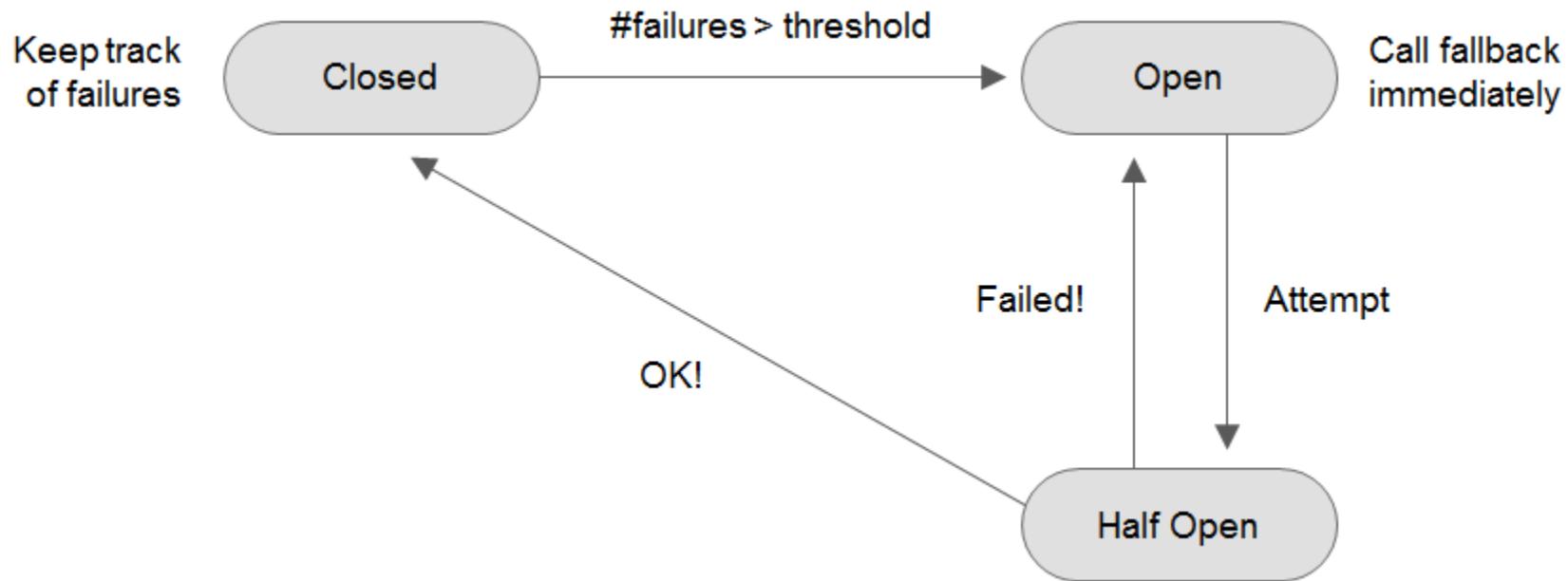
Chaining (Fail)



Chaining (Cascading Fail)



Circuit Breaker



Demo

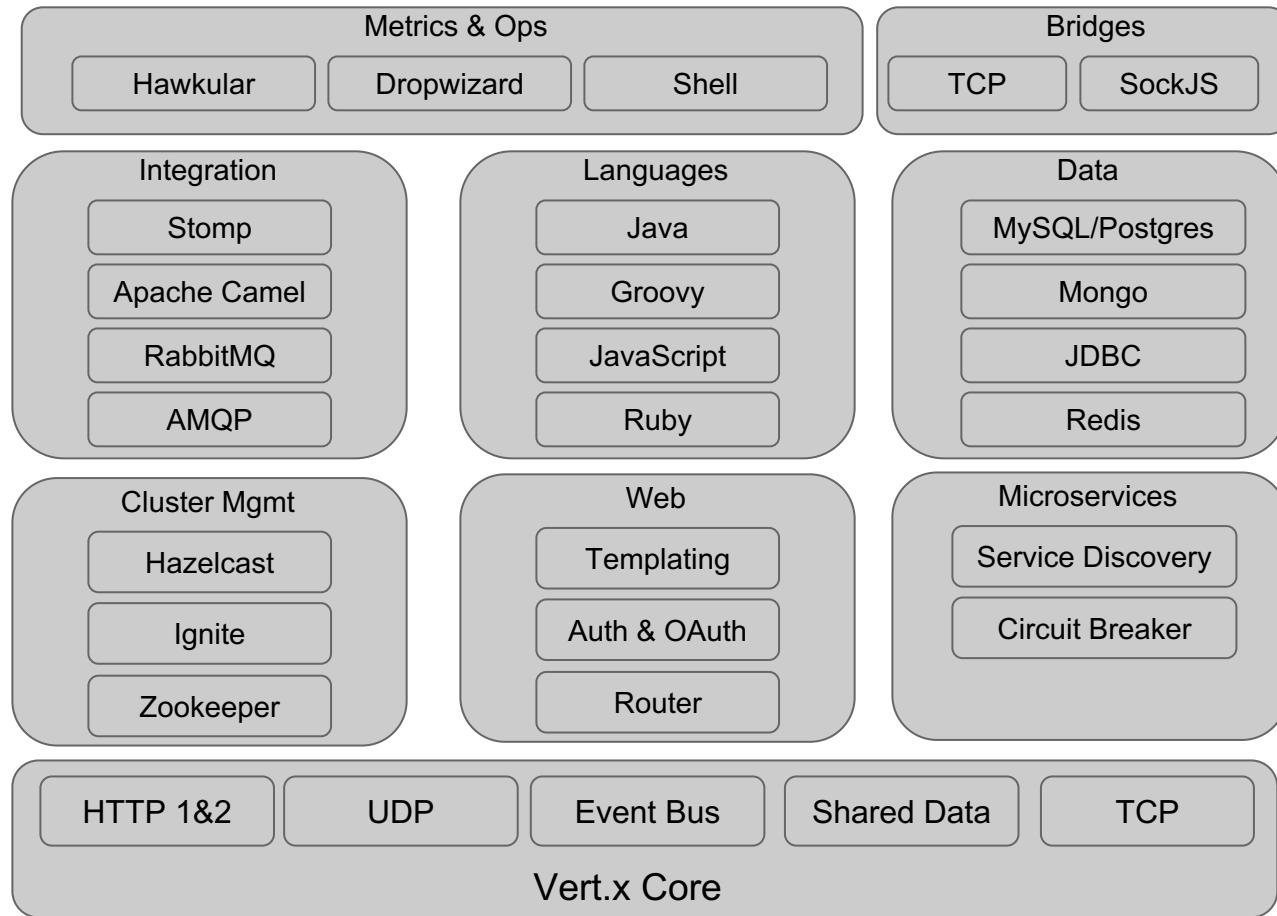
bit.ly/msa-instructions

<https://github.com/burrsutter/kube4docker>



Vertx.io—an Eclipse project

Vert.x is a toolkit
for building reactive
applications on the JVM.



Vert.x

(Un-)Opinionated) Toolkit for reactive, async applications

Every user has their own Vert.x way

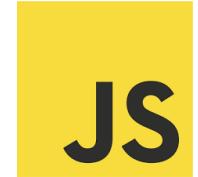
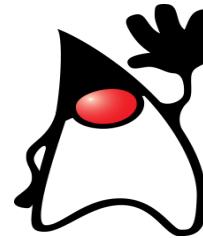
Low-latency microservices that interact using the EventBus

Built in elasticity and resiliency

Polyglot: Java, JavaScript, Groovy, Ruby, Ceylon

vertx.io

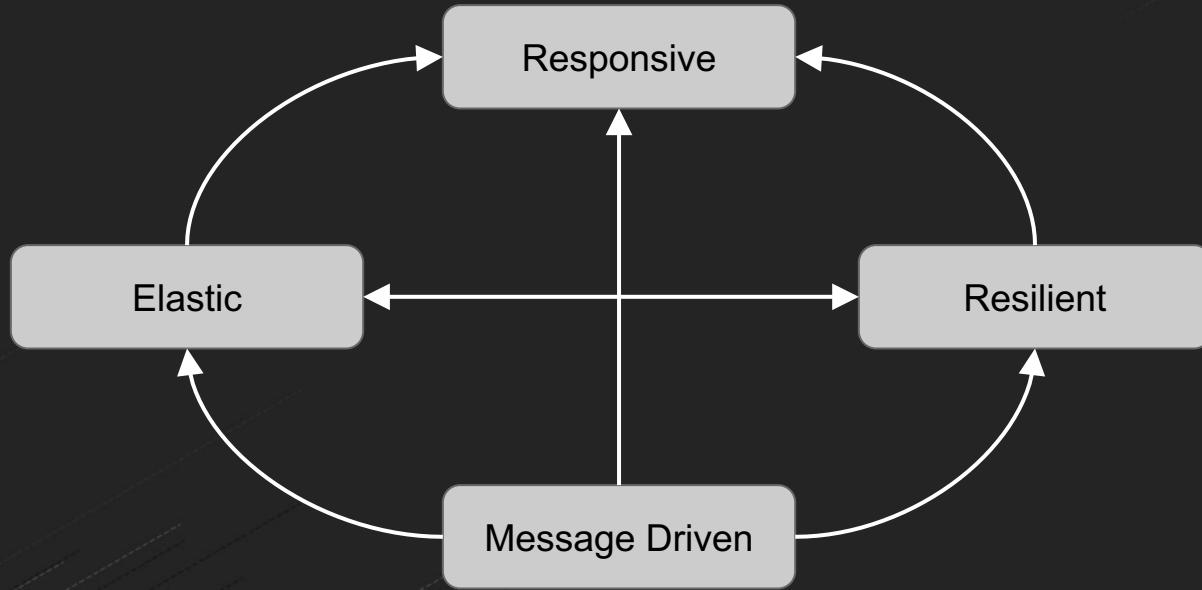
<http://vertx.io/blog/my-first-vert-x-3-application/>



Reactive

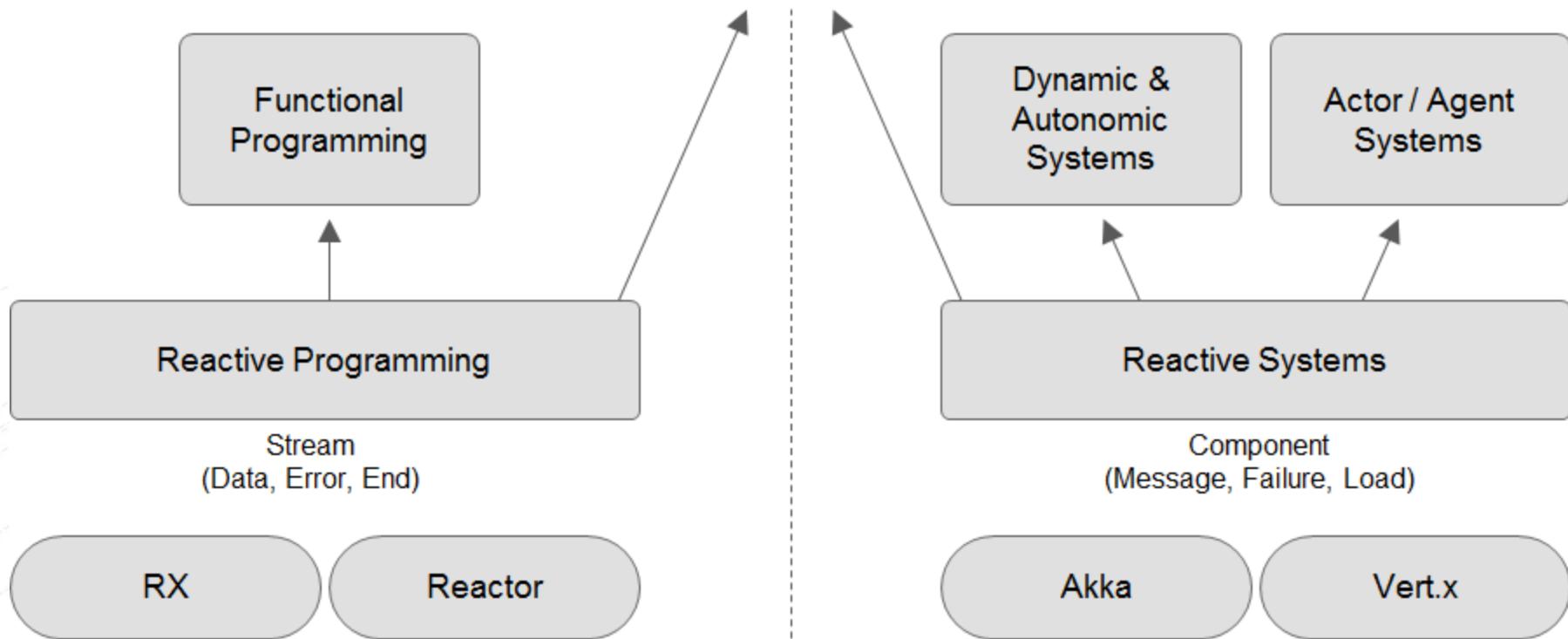
Reactive Manifesto

Reactive Systems (not Programming)



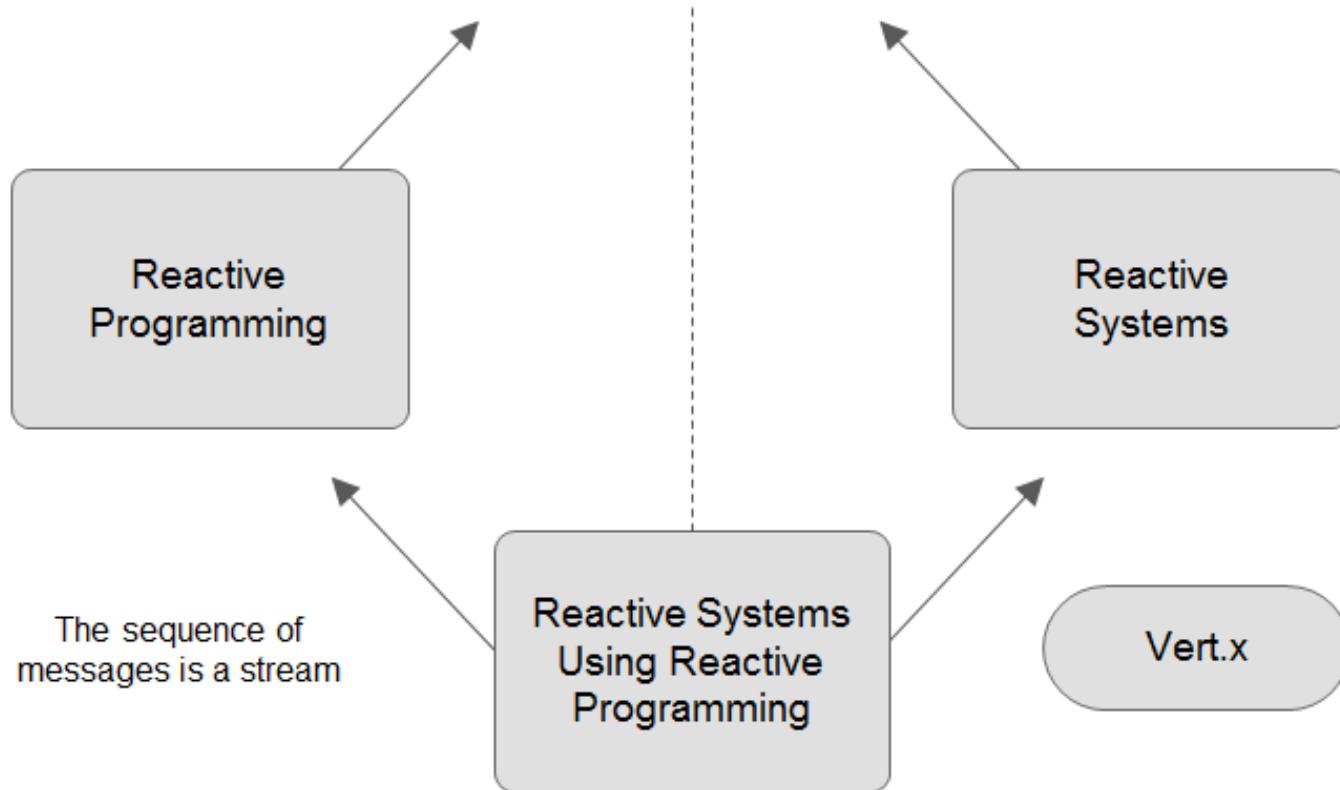
REACTIVE

"Showing a response to a stimulus"

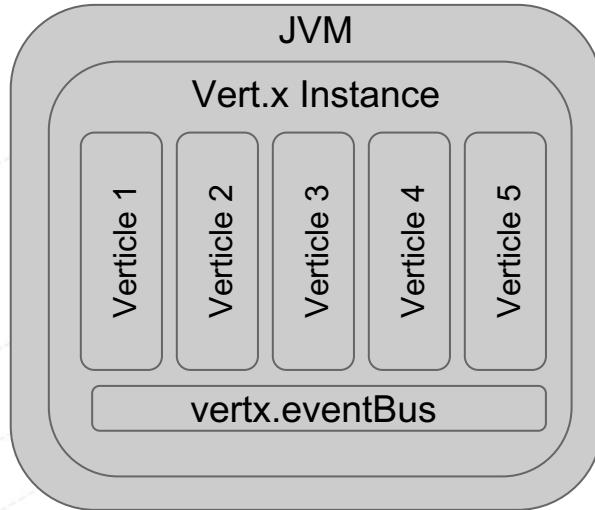


REACTIVE

"Showing a response to a stimulus"



Vert.x Verticle

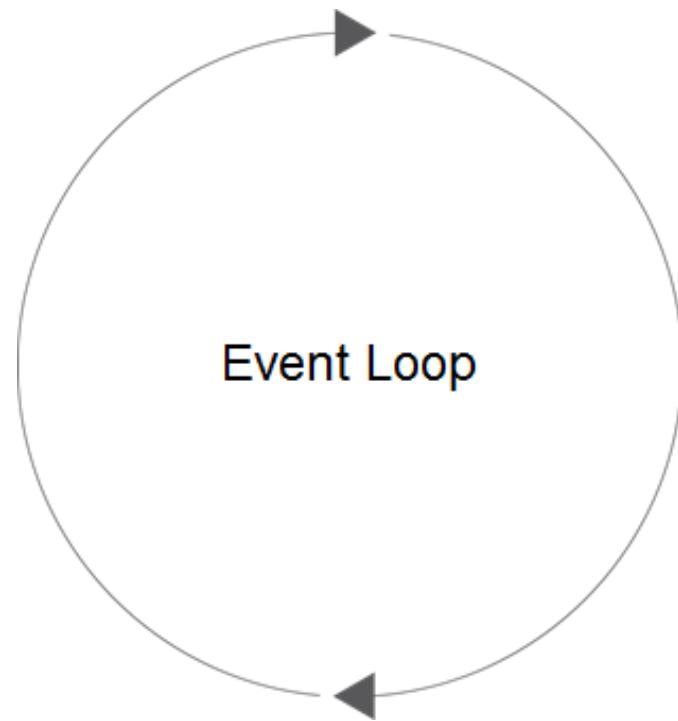


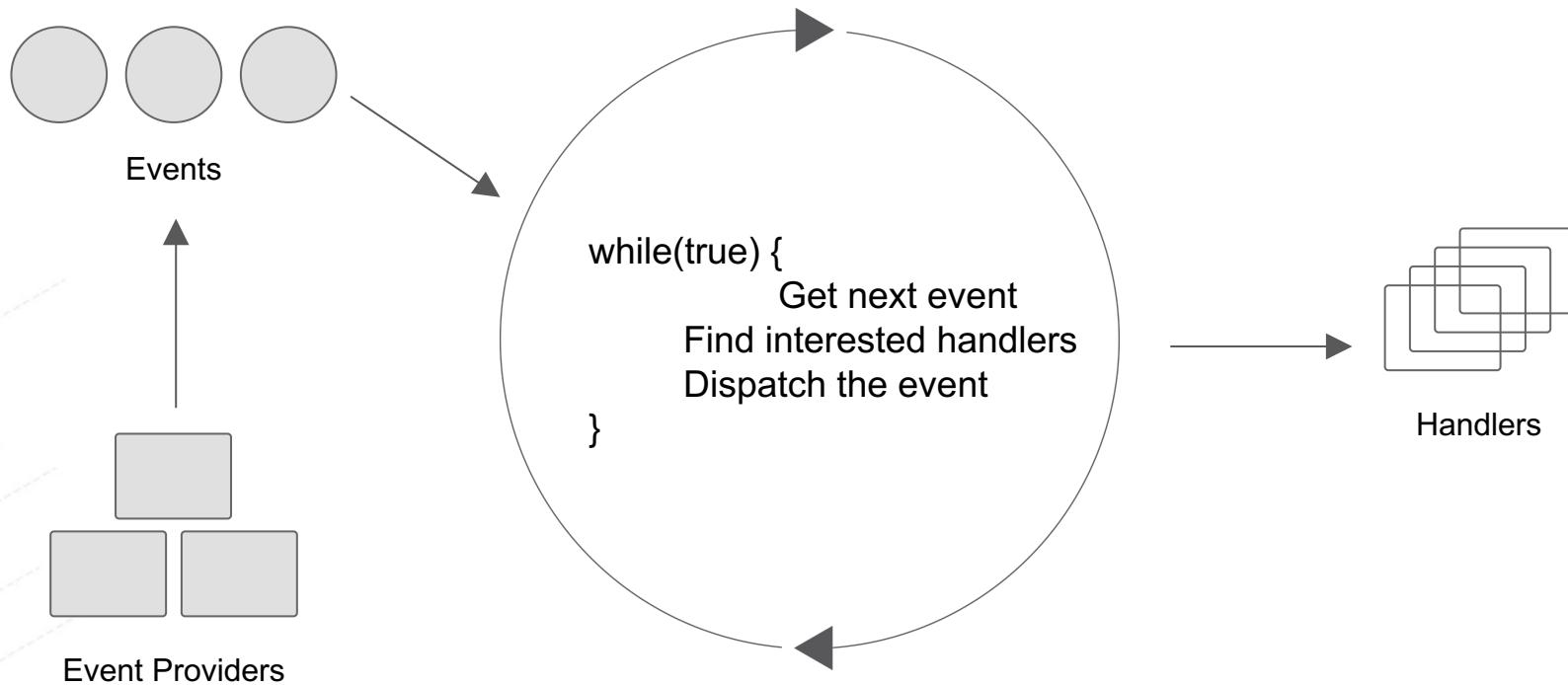
A Verticle is the programmable unit – a component

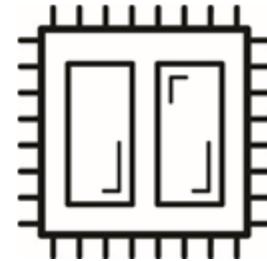
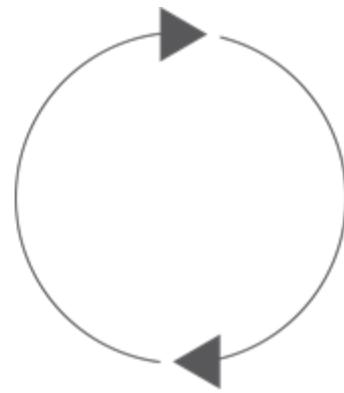
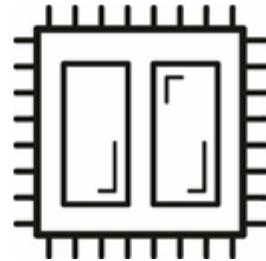
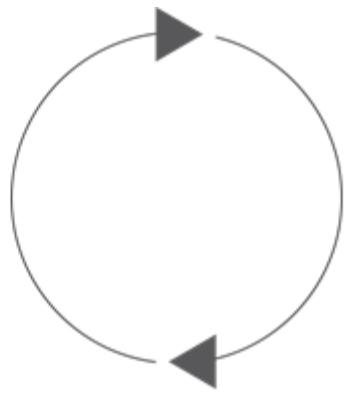
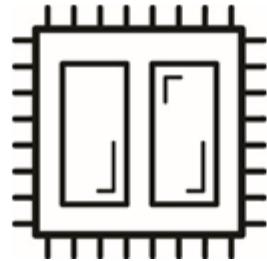
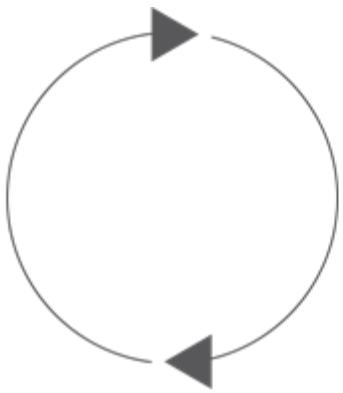
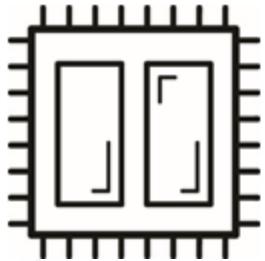
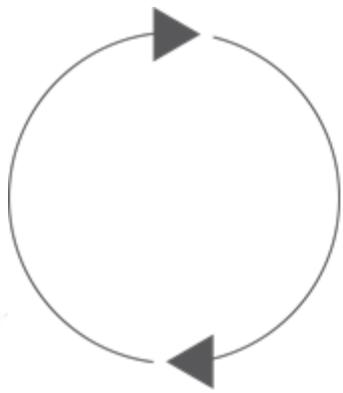
A Verticle is always executed on the same thread

A single thread may execute several verticles – event loop

A Verticle Instance normally starts 1 thread/event-loop per core







Vert.x HTTP Verticle

```
import io.vertx.core.AbstractVerticle;

public class MainVerticle extends AbstractVerticle {

    @Override
    public void start() {
        vertx.createHttpServer()
            .requestHandler(req ->
                req.response().end("Hello Vert.x v3!"))
            .listen(8080);
    }
}
```

Super-Fast Getting Started

```
$ git clone https://github.com/cescoffier/vertx-maven-starter
```

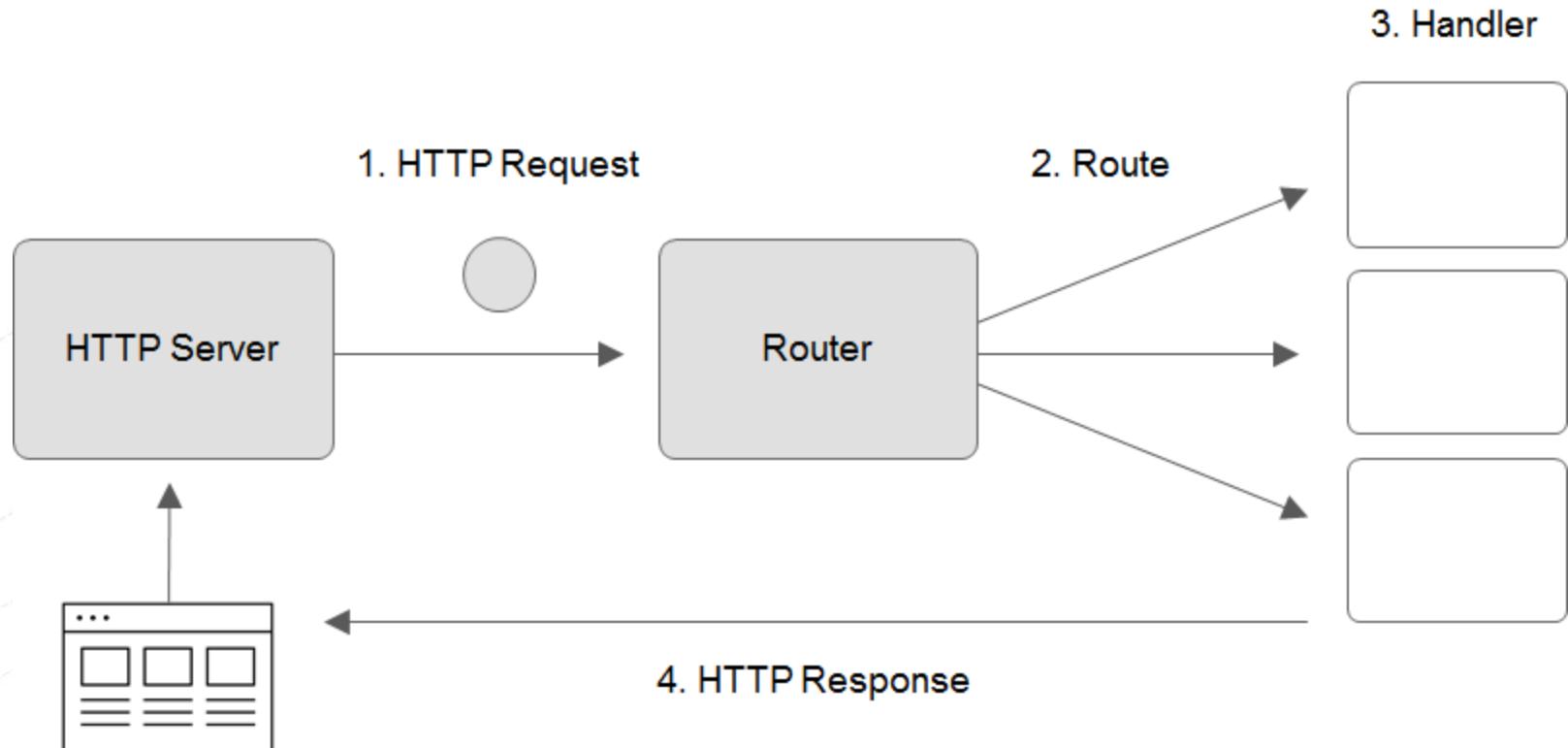
```
$ cd vertx-maven-starter
```

```
$ mvn package
```

```
$ java -jar target/vertx-start-project-1.0-SNAPSHOT-fat.jar
```

In your browser: <http://localhost:8080>

Web Apps & REST

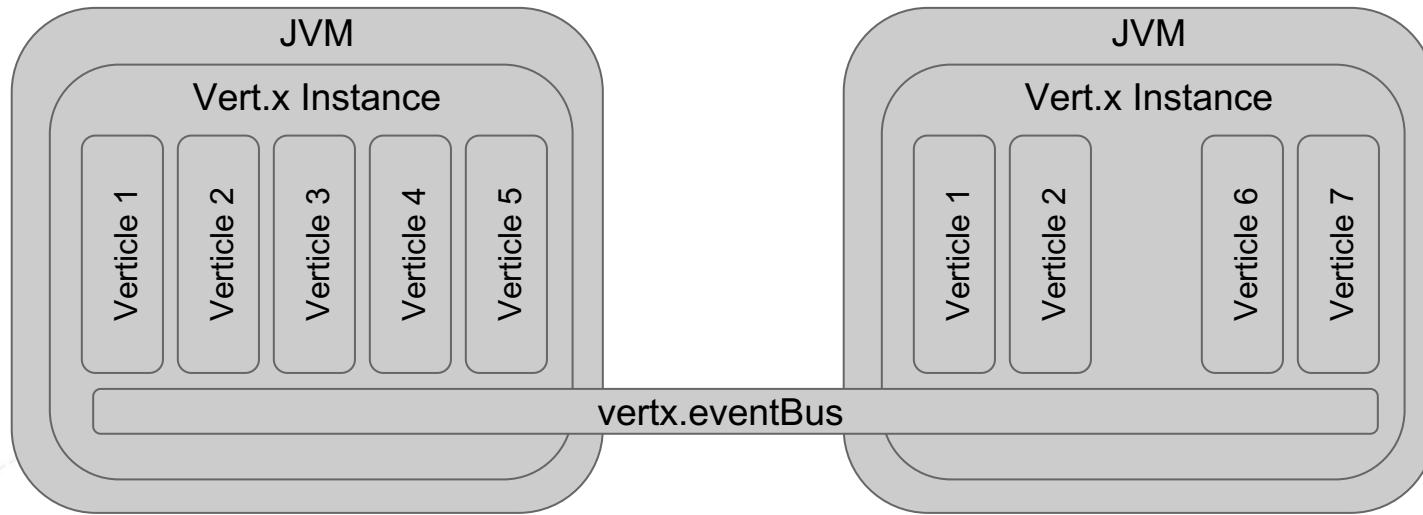


Vert.x HTTP Router

```
router.get("/api/whiskies").handler(this::getAll);
router.get("/api/whiskies/:id").handler(this::getOne);
router.post("/api/whiskies").handler(this::addOne);
router.delete("/api/whiskies/:id").handler(this::deleteOne);
router.put("/api/whiskies/:id").handler(this::updateOne);
// Serve static resources from the /assets directory
router.route("/assets/*").handler(StaticHandler.create("assets"));

vertx
    .createHttpServer()
    .requestHandler(router::accept)
    .listen(8080, result -> {
        if (result.succeeded()) {
            future.complete();
        } else {
            // could be something else is running on our port
            future.fail(result.cause());
        }
    });
});
```

Vert.x EventBus



Demo

https://github.com/burrsutter/vertx3_introduction

HTTP getNow() Circuit Breaker

```
CircuitBreakerOptions options = new  
CircuitBreakerOptions()  
.setMaxFailures(5)  
.setTimeout(5000)  
.setFallbackOnFailure(true);
```

```
CircuitBreaker breaker =  
CircuitBreaker.create("my-circuit-breaker", vertx,  
options)  
.openHandler(v -> {  
    System.out.println("Circuit opened");  
}).closeHandler(v -> {  
    System.out.println("Circuit closed");  
});
```

```
Future<String> result = breaker.executeWithFallback(future -> {  
    vertx.createHttpClient().getNow(8080, "localhost", "/",  
        response -> {  
            if (response.statusCode() != 200) {  
                future.fail("HTTP error");  
            } else {  
                response.exceptionHandler(future::fail).bodyHandler(buffer  
                    -> {  
                        future.complete(buffer.toString());  
                    });  
            }  
        });  
}, v -> {  
    // Executed when the circuit is opened  
    return "Hello (fallback)";  
});
```

https://github.com/burrsutter/vertx3_introduction/blob/master/7_httpinvoker/Client.java#L13

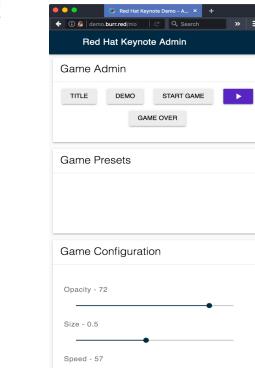
Demos

<https://github.com/redhat-reactive-msa>

<https://github.com/cescoffier/vertx-microservices-examples>

<http://escoffier.me/vertx-hol/>

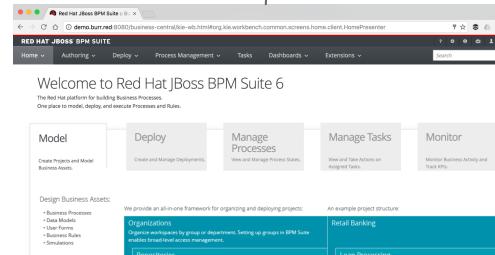
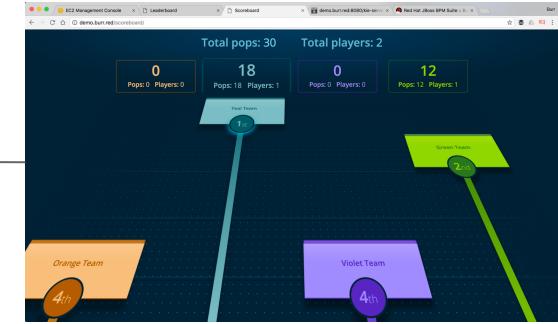
Recording: <https://youtu.be/ooA6FmTL4Dk?t=1739>



vertx-game-server

vertx-achievement-server

Java EE on JBoss with Drools+jBPM





RED HAT DEVELOPERS

Reactive Microservices with Eclipse Vert.x

@burrsutter

<http://developers.redhat.com>

<http://bit.ly/reactivems>