



RED HAT DEVELOPERS

Reactive Microservices with Eclipse Vert.x

@bursutter
vertx.io

<http://developers.redhat.com>
<http://bit.ly/reactivemsa>

Change History

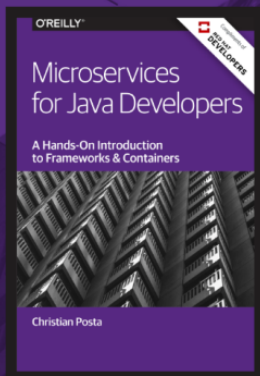
0.1 - Forked from bit.ly/helloworldmsa

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.

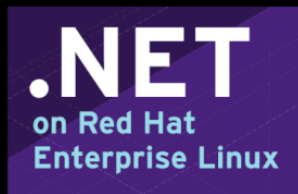
[DOWNLOAD NOW](#)

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



Didn't make it to DevNation? Watch Sessions OnDemand



Join Red Hat Developers and try OpenJDK



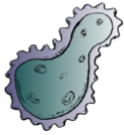
Getting started, collections, indexes, and dangers. Download now.



389 project | aeolus-configure | anaconda | attr | Audrey | Augeas | AuthHub | Autofs | Avahi | 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 | DPK | Dracut | Editline Library | EDS | EJB 3 | elfutils | Embedded Jopr | Evolution | Evolution-exchange | eXo JCR | ext3 | ext4 | fence-agents | fence-virt | Flannel | fontconfig | func | Gamin | Gateln Portal | gcc | gcc | gcj | gdb | GFS | GFS2 | glib | gss-proxy | GTK+ | gvfs | gzip | HTCondor | ID Utils | imagefactory | IPSec-Tools | iptraf-ng | Jandex | Java | JGroups | Jreadline | JRuby | JSFUnit | JUDDI | Katello | Kerberos | Kimchi | Koji | libguestfs | libibverbs | libminidump | libnotify | libqb | libstoragemgmt | libuser | libvirt | libxml2 | libxslt | 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 | nautilus | netfilter | NetworkManager | NFS-Ganesha | nmap | Open vSwitch | openais | OpenDaylight | OpenJDK | OpenSSH | OpenSSL | OPNFV | OProfile | oVirt | oVirt Engine | oVirt Node | OVMF | Oz pacemaker | PackageKit | PackageKit | pango | PAPI | Parady | patchutils | Performance Co-Pilot | PicketBox | PicketLink | piglit | pixman | polkit | Polkit Qt | Poppler | Portlet Bridge | PostgreSQL | PressGang | psmisc | Pulp | PulseAudio | punji | pyafs | 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-dwim | vdaagent | vdsim | vfs | Virt-clone | Virt-image | Virt-Install | Virt-manager | Virt-viewer | Virtual memory manager | Wallaby | Wayland | Wise | XFS | xinput | 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
*-aaS/Cloud
Reactive
Functional



Mythical
Beast

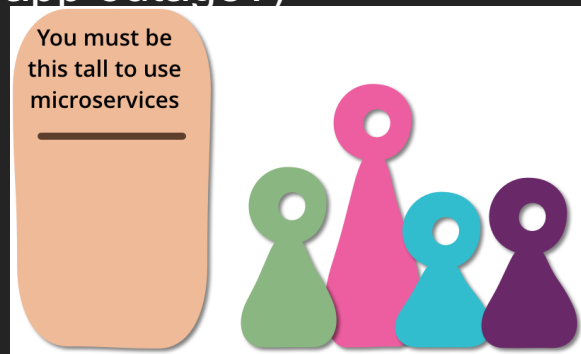
You Must Be This Tall

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

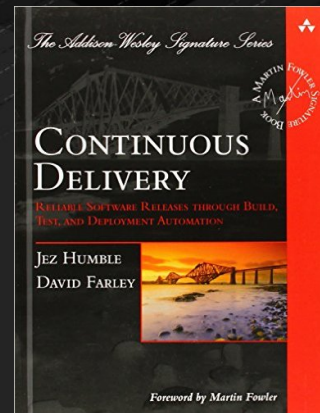


<http://martinfowler.com/bliki/MicroservicePrerequisites.html>

@burrsoter Deployment Pipeline

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 ALLSPAUGH

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



fabric8

fabric8.io

fabric8-console x 192.168.64.10:31776/workspaces/default/forge/command/project-new

fabric8 Teams default Create Project James

Create a new project

Project Type

 Django	 From 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 >

fabric8-console x | 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

Name of chat room to use for this project

Issue project name

Name of the issue tracker project

☐ Code review

Enable code review of all commits

Cancel Next >

James

192.168.64.10:31776/workspaces/default/projects/awesomcamel

fabric8

Teams

default

Apps

awesomcamel

Dashboard

Dashboard

Pipelines

Builds

Metrics

Settings

Source

Camel

Forge

Environments

Testing

Staging

Production

awesomcamel-1: 1.0.1

Build #1

d0232ea

Docs

awesomcamel-1: 1.0.1

Build #1

d0232ea

Docs

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 awesomcamel build 1
to Abort reply: fabric8 jenkins abort job awesomcamel 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

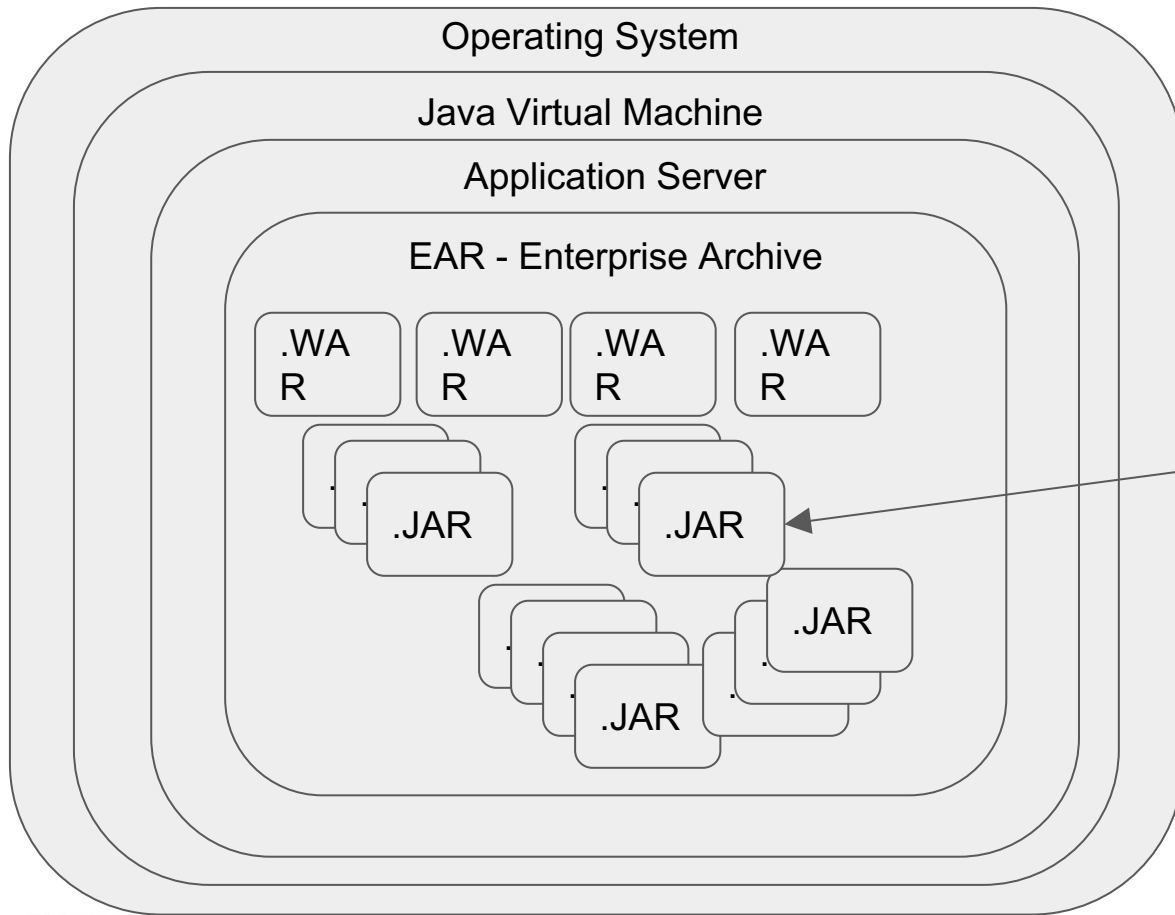
d0232ea

index on master: 2448d85 project-new --type=Integration --topLevelPackage=org.example --version=1.0.0-SNAPSHOT --
targetLocation=/tmp/fabric8-forge/user/gogsadmin --stack=None --named=awesomcamel

5f60107

@bursutter

RED HAT
DEVELOPERS





Programmers
(18)



Business
Analysts
(4)



Project
Managers
(2)



Quality
Assurance
(6)



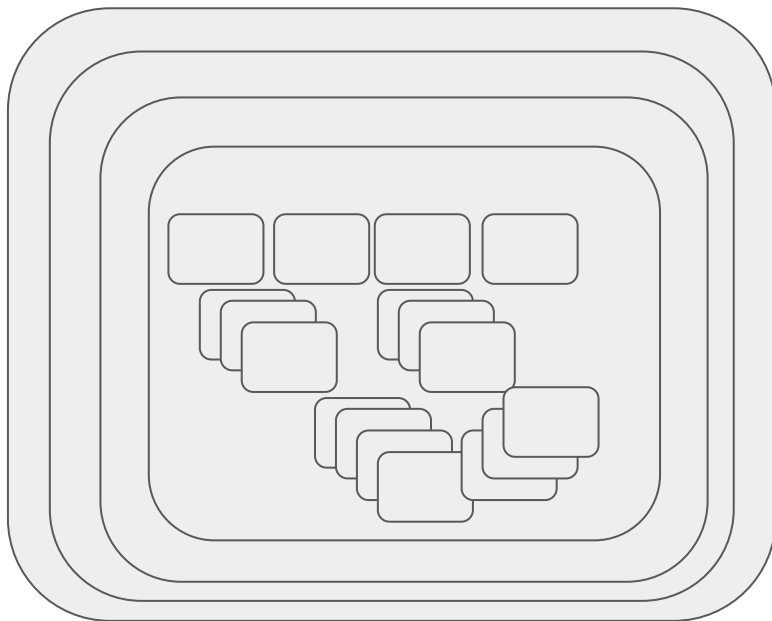
Compliance
(2)

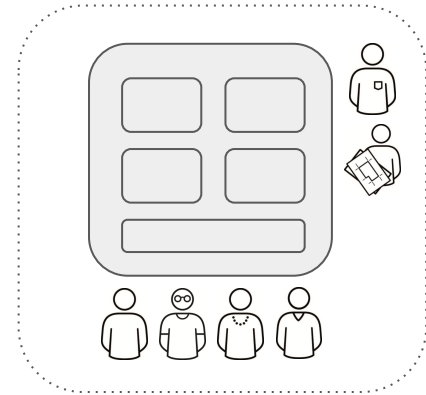
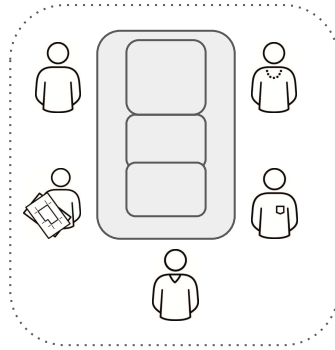
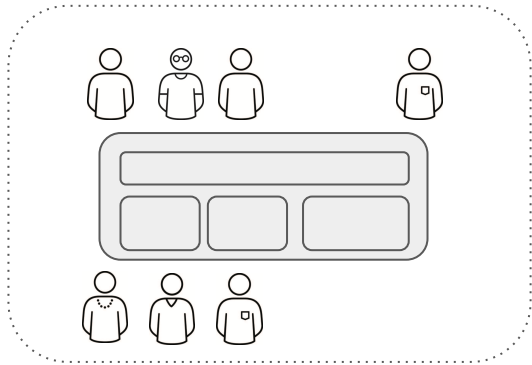


Operators
(6)

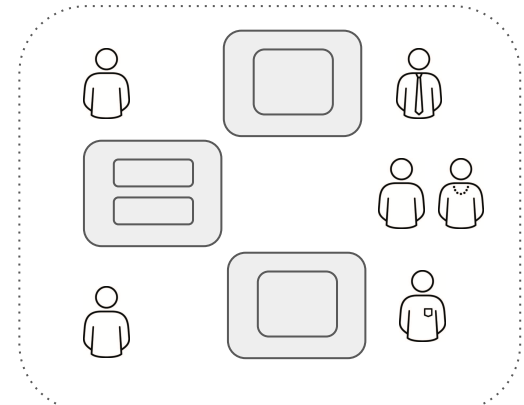
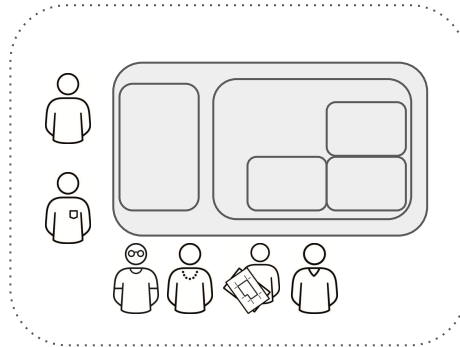
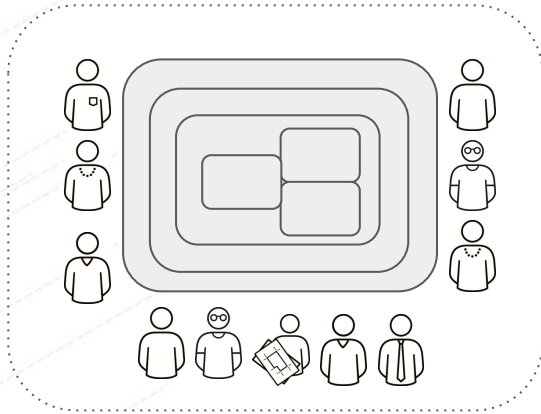


DBAs
(3)





2-Pizza Teams You Build It, You Own it



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

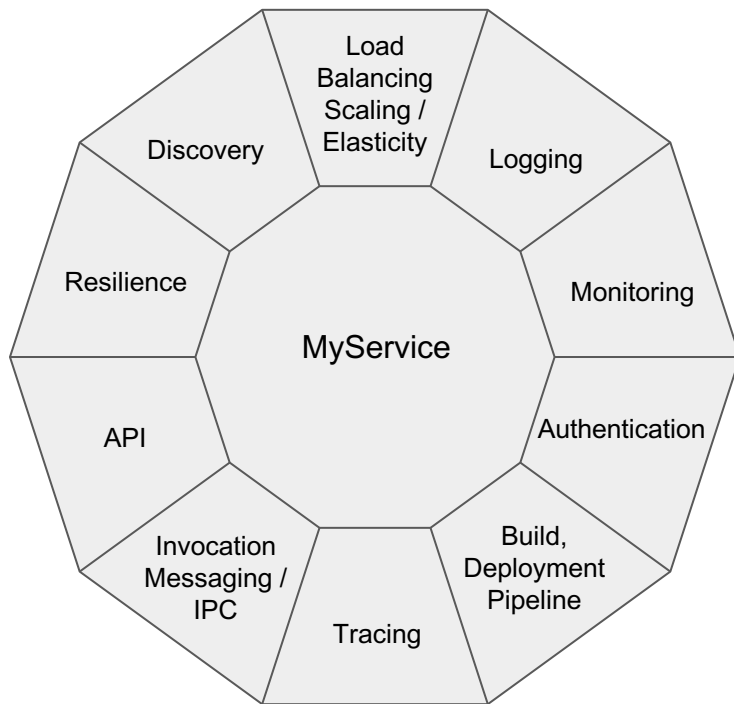
Decentralized **Data** Management

WTF?

Stay tuned - new book

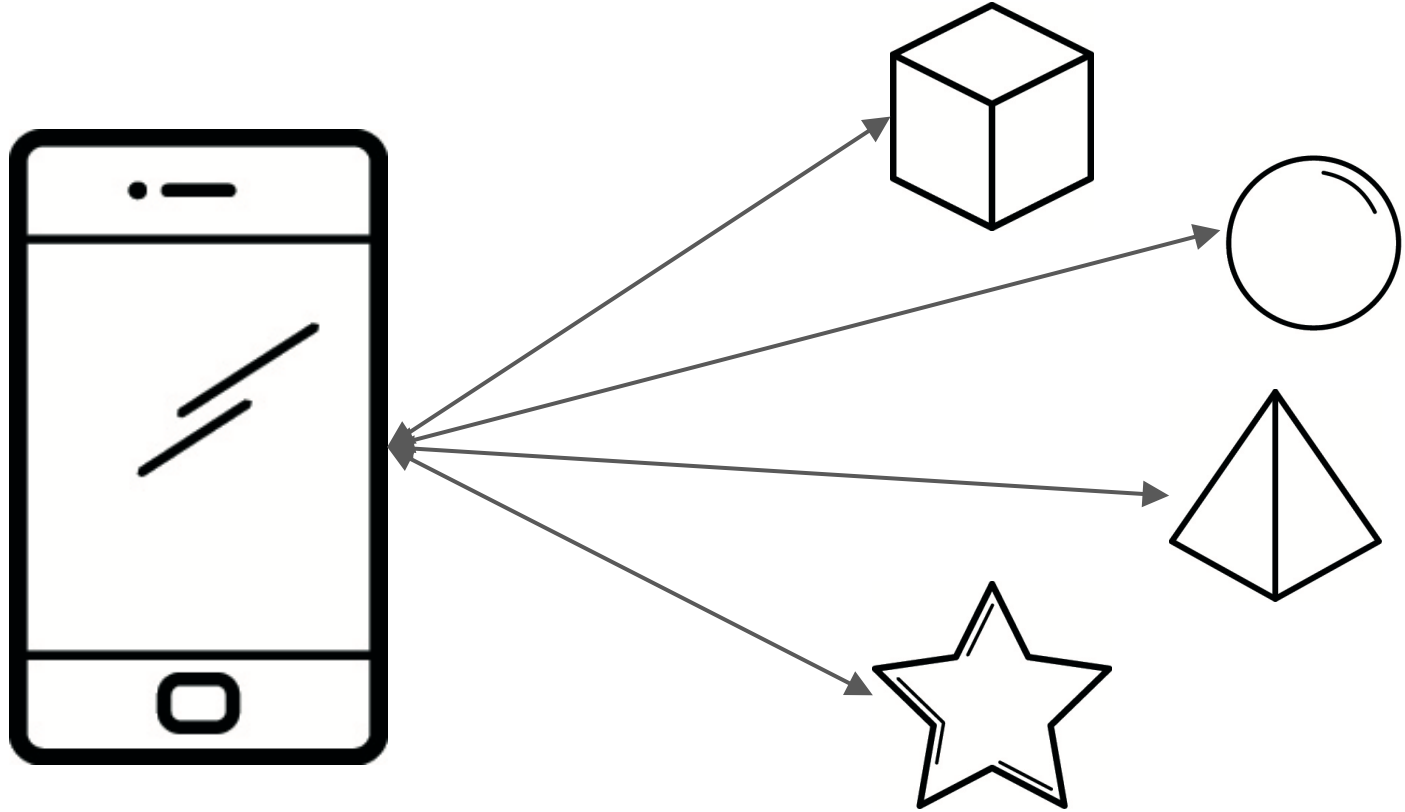
arriving at <https://developers.redhat.com> late Feb 2017

Microservice Concerns



Microservice Patterns

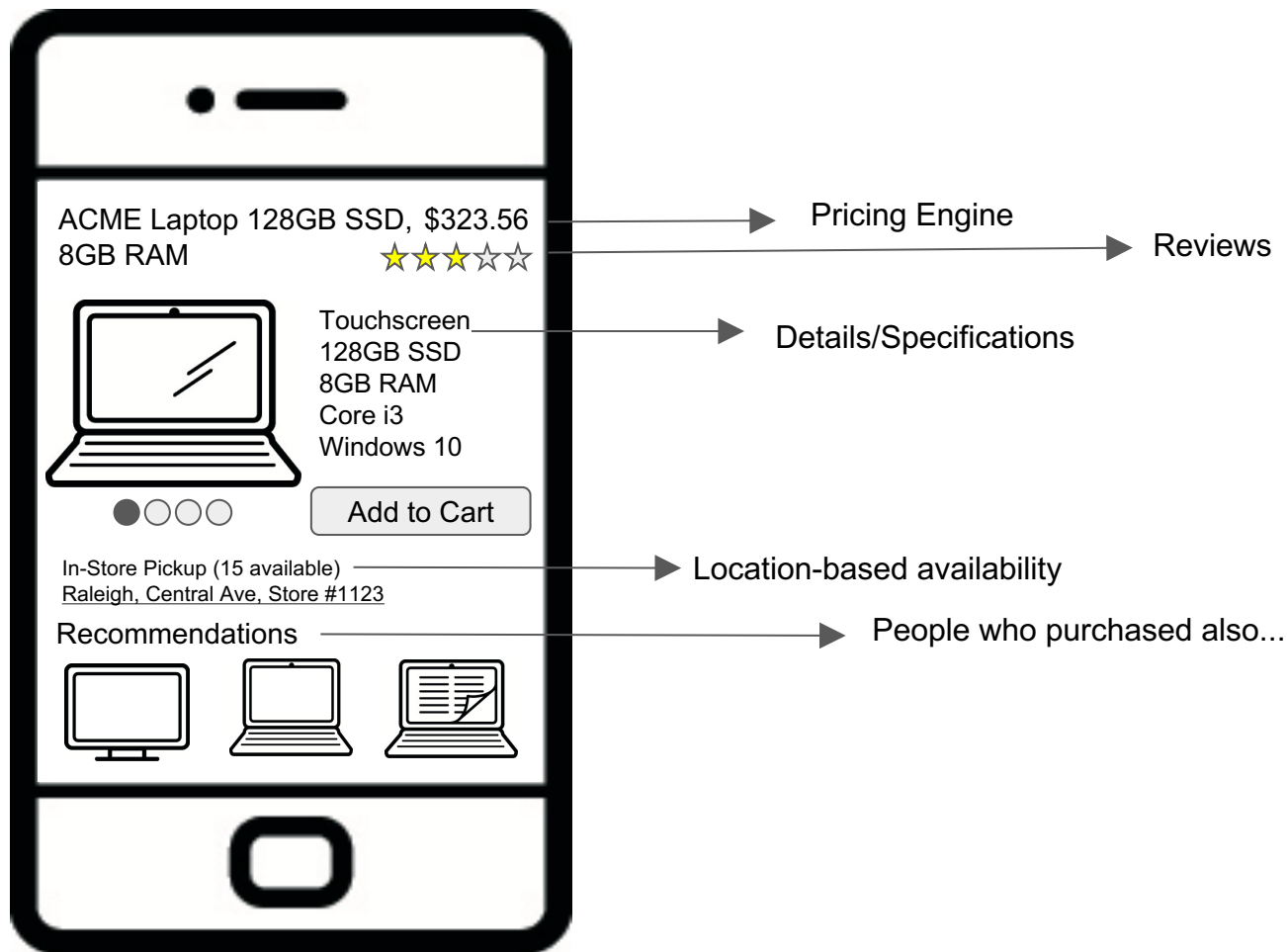
Browser



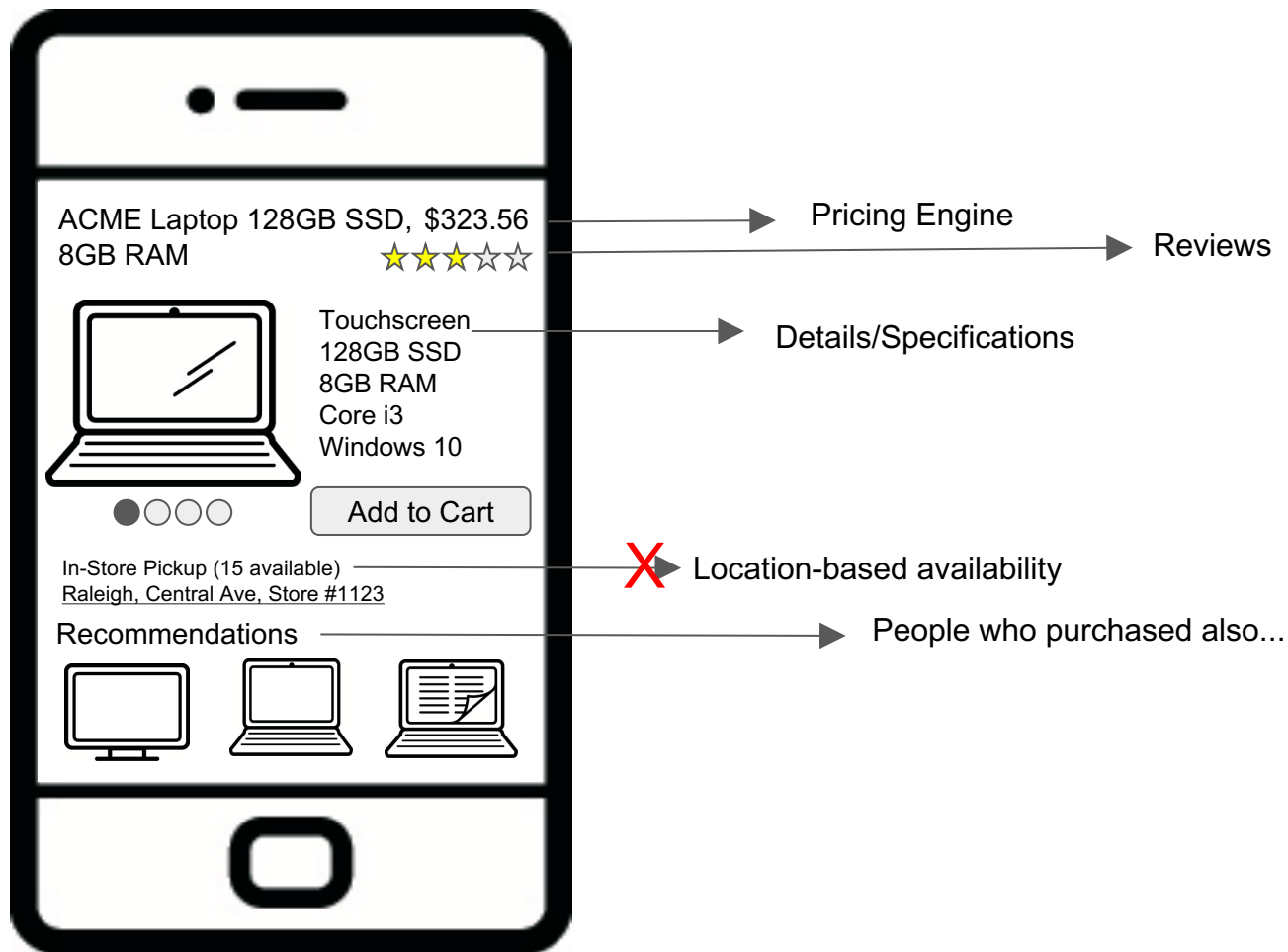
Example



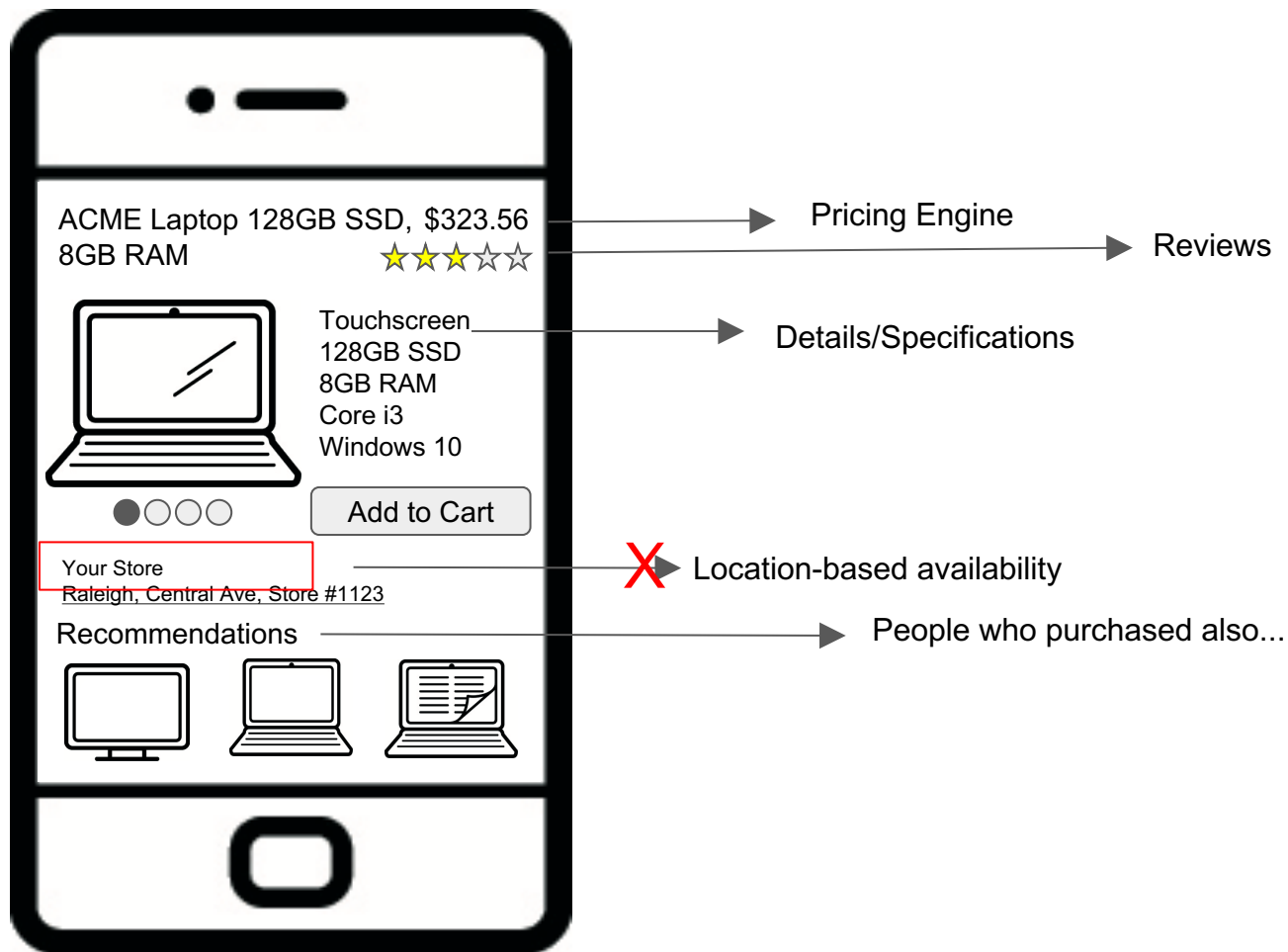
Example



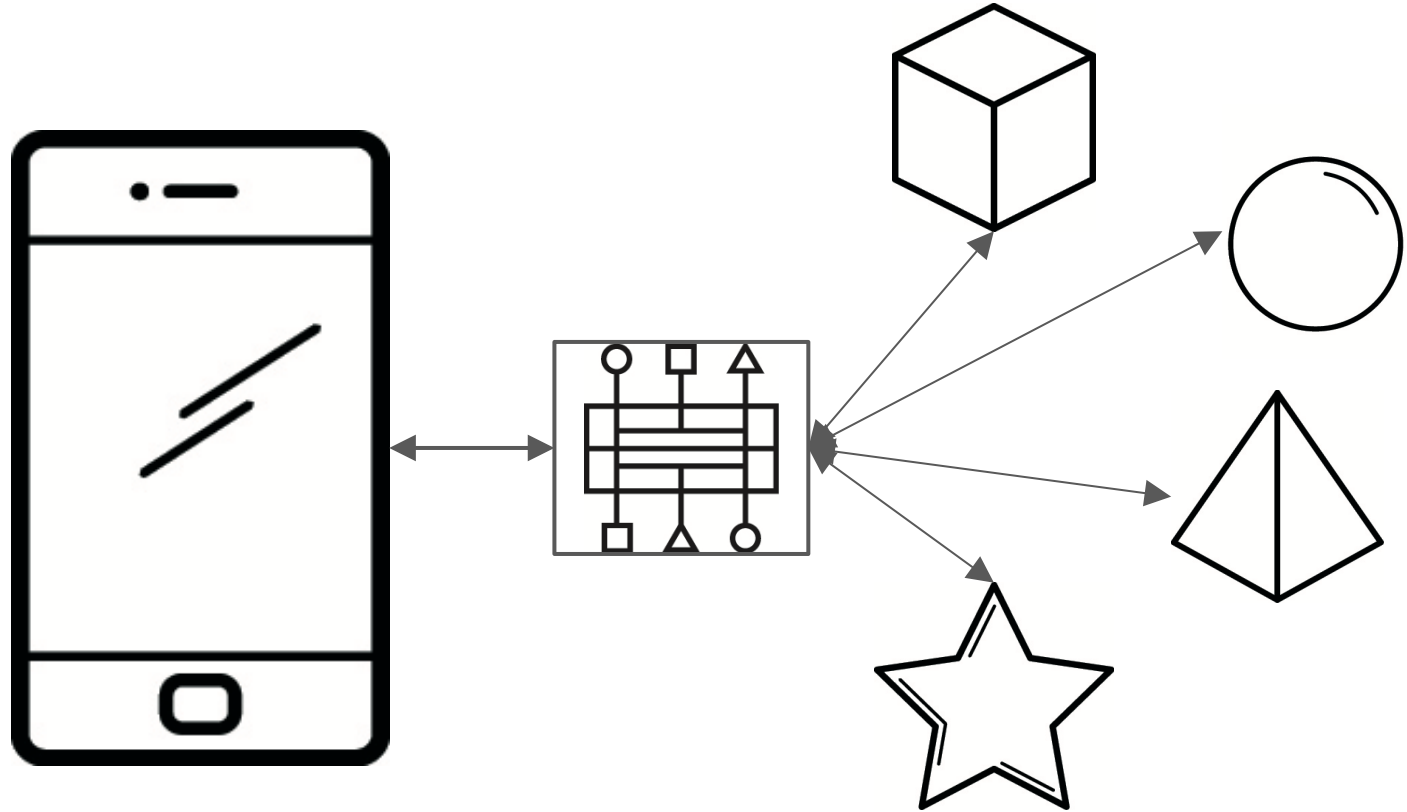
Example (Fail)



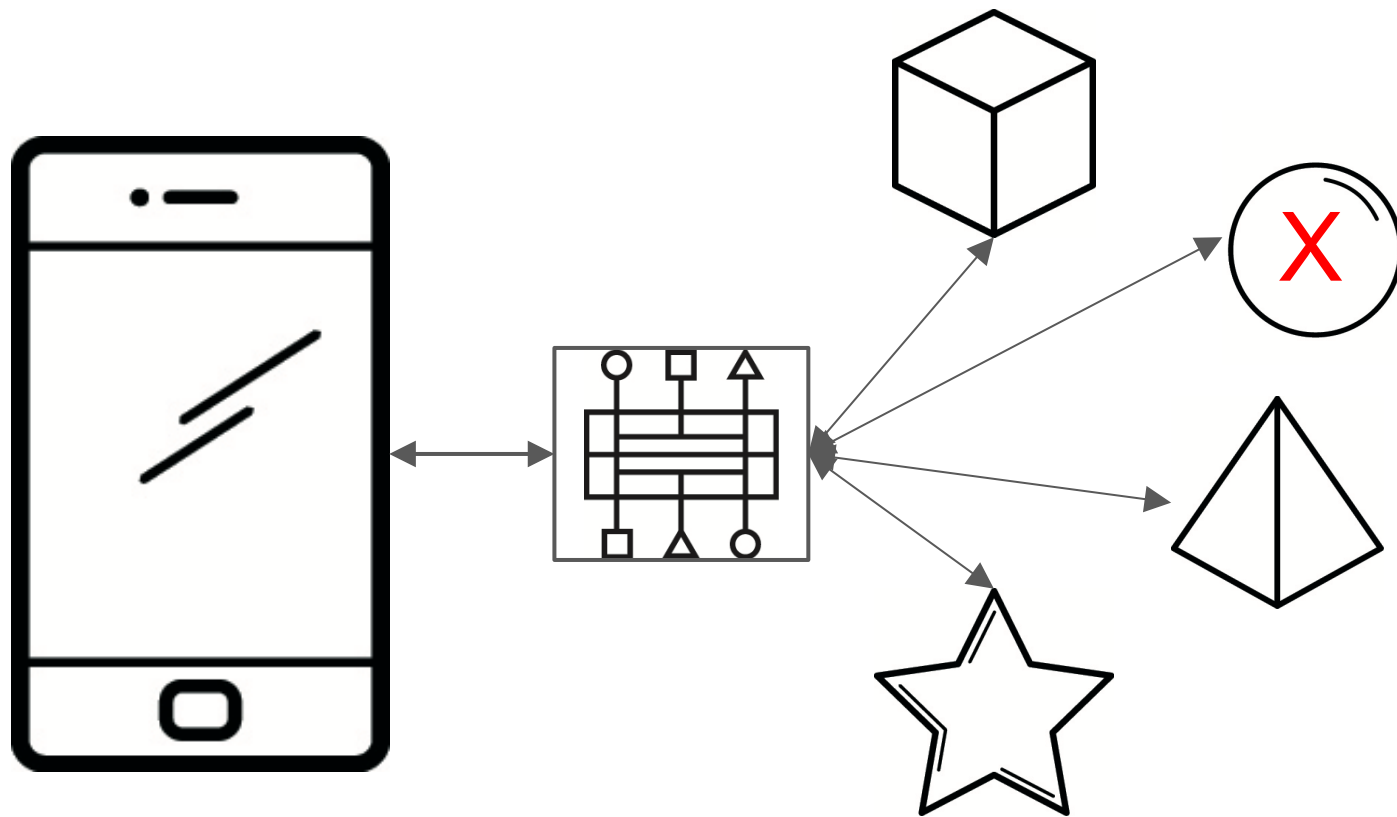
Example (Fallback)



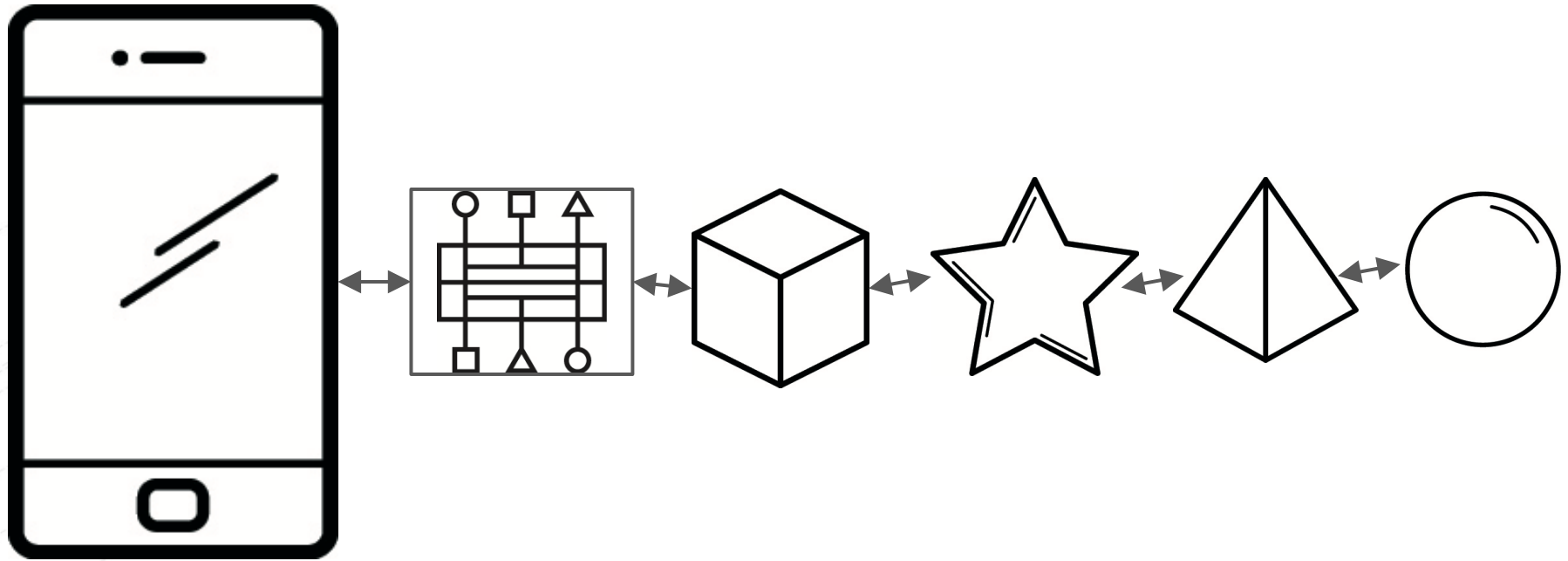
API “Gateway”



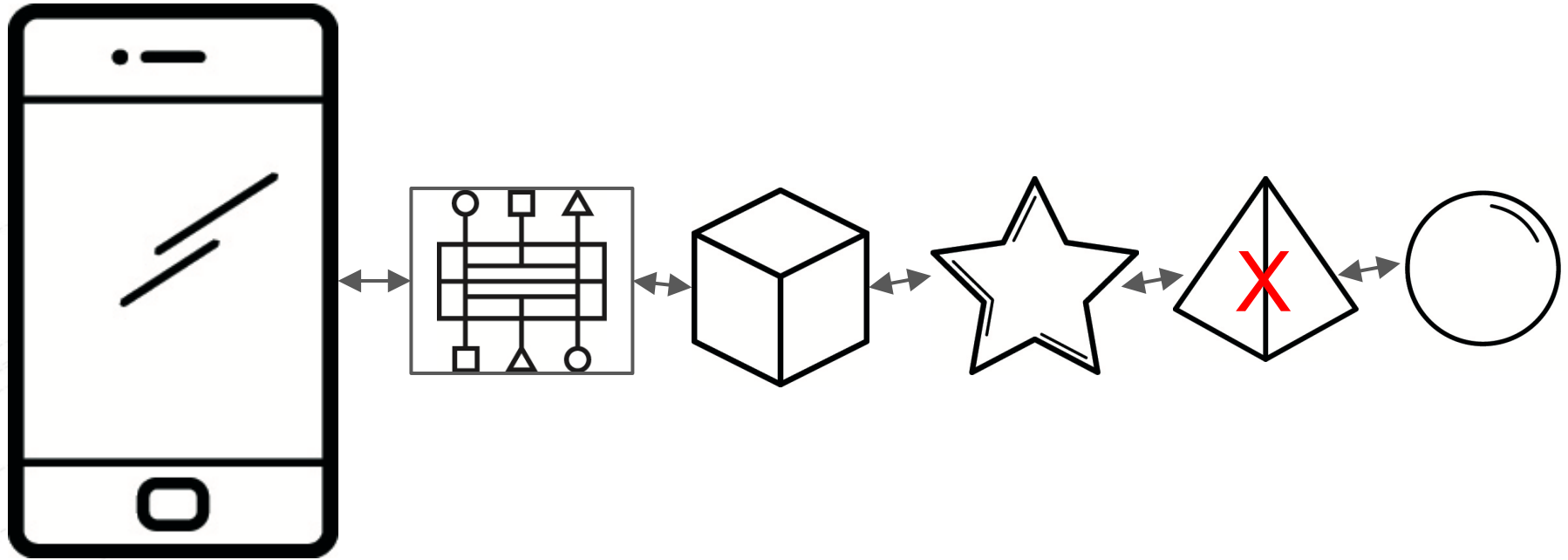
API “Gateway” (Fail)



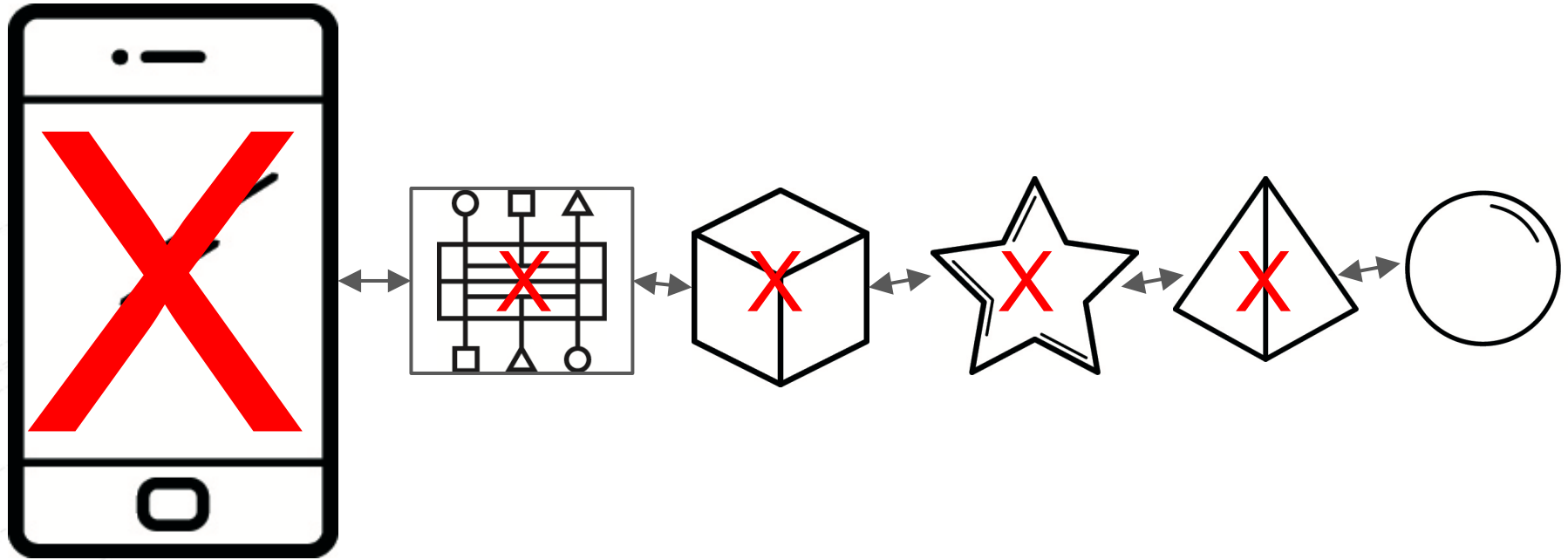
Chaining



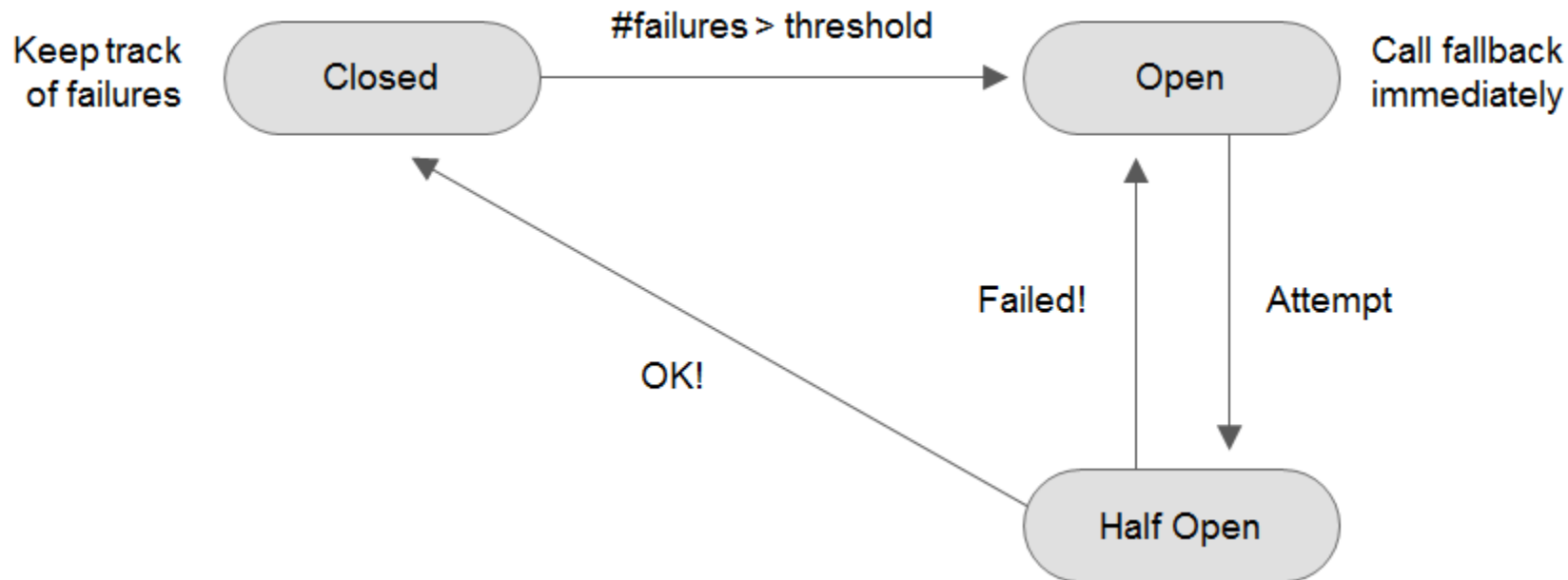
Chaining (Fail)



Chaining (Cascading Fail)



Circuit Breaker



Demo

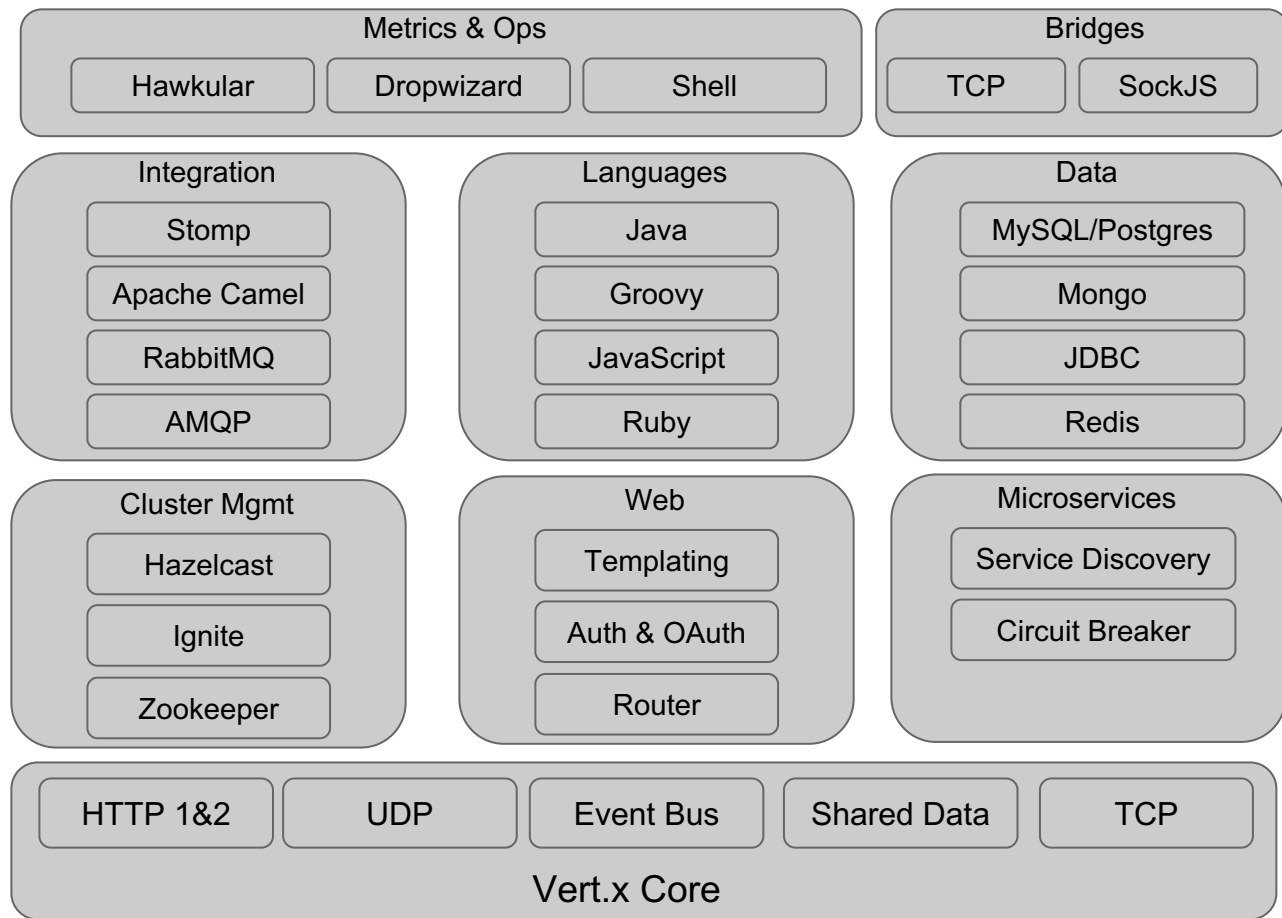
bit.ly/msa-instructions

<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

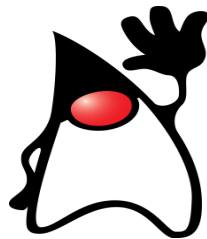
Low-latency microservices that interact using the EventBus

Built in elasticity and resiliency

Polyglot: Java, JavaScript, Groovy, Ruby, Ceylon

vertx.io

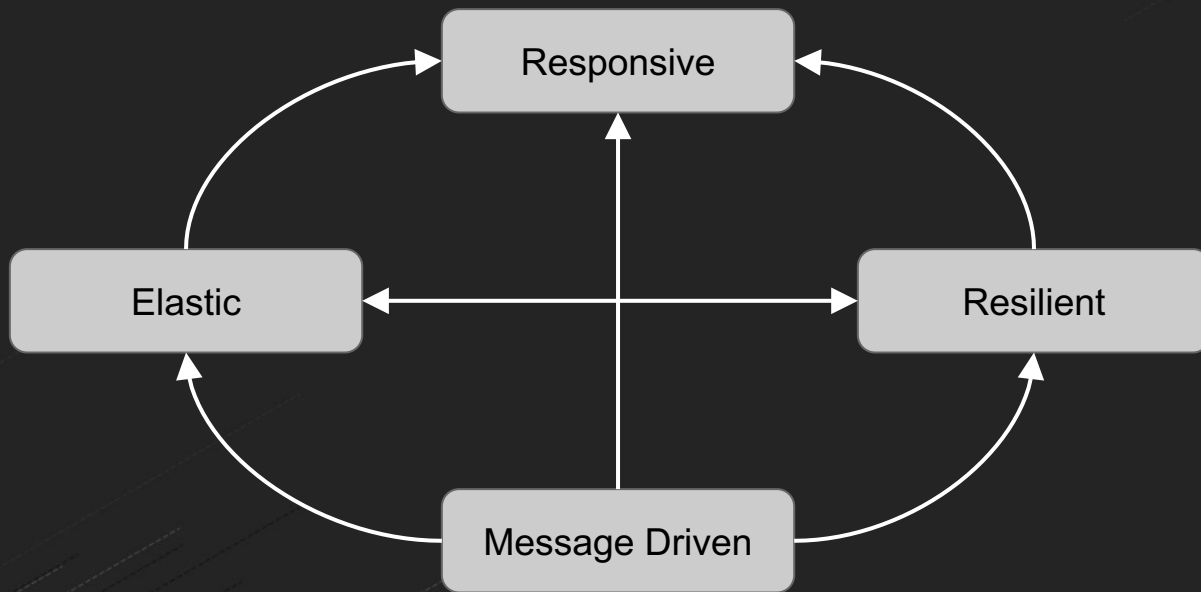
<http://vertx.io/blog/my-first-vert-x-3-application/>



Reactive

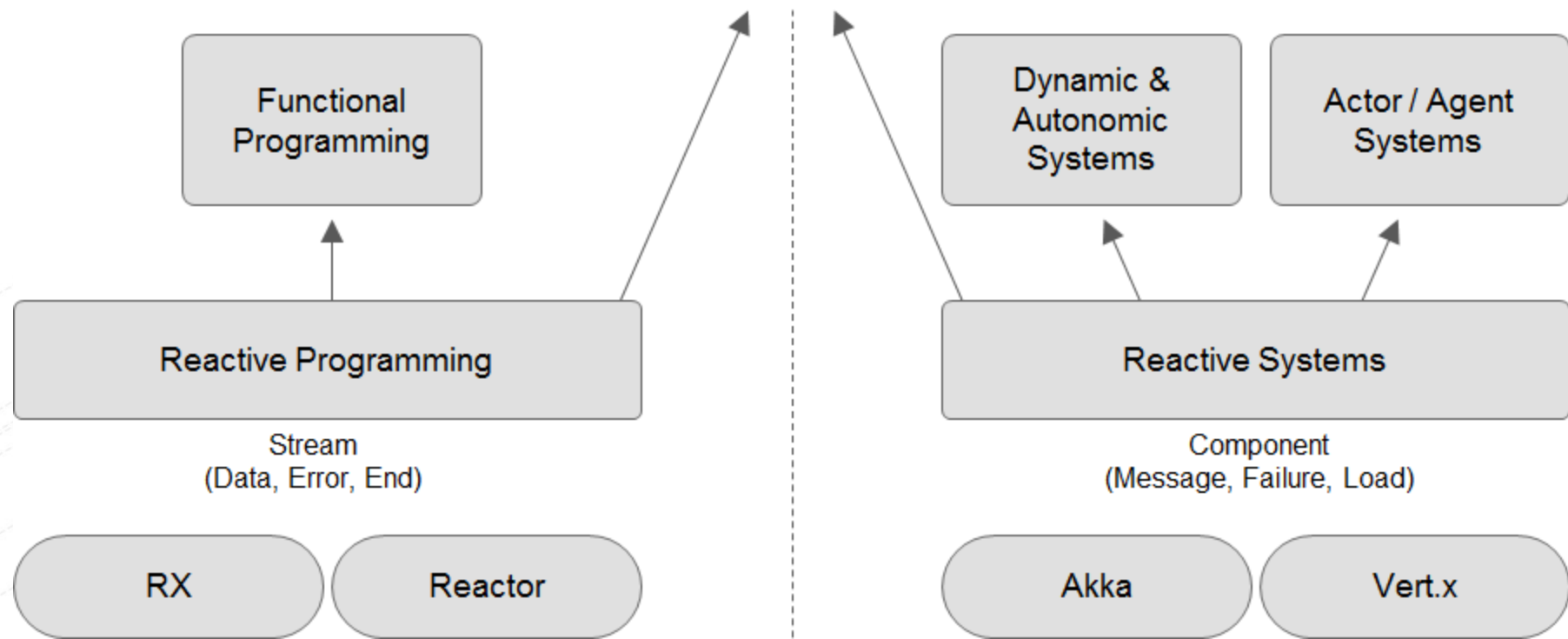
Reactive Manifesto

Reactive Systems (not Programming)



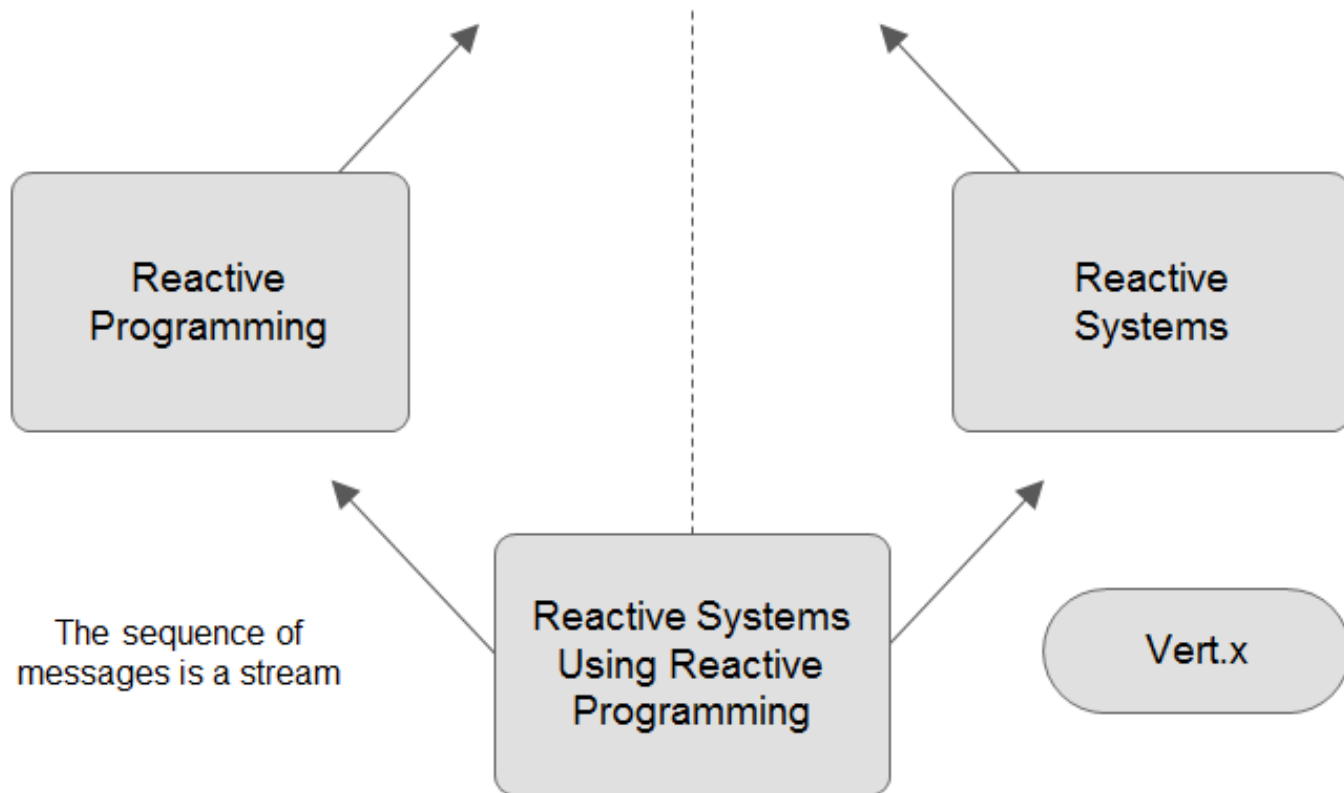
REACTIVE

"Showing a response to a stimulus"

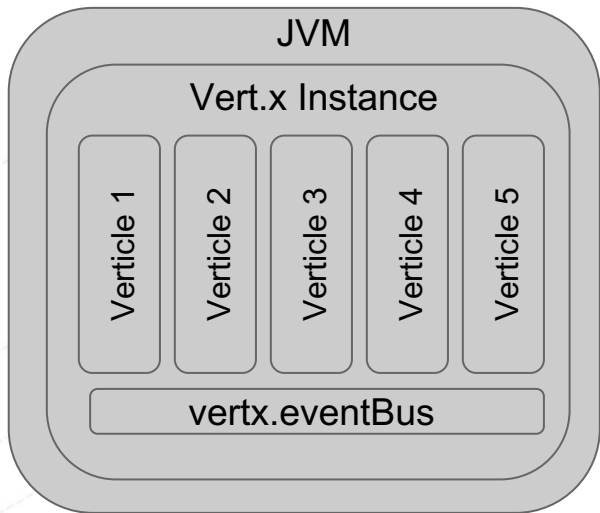


REACTIVE

"Showing a response to a stimulus"



Vert.x Verticle

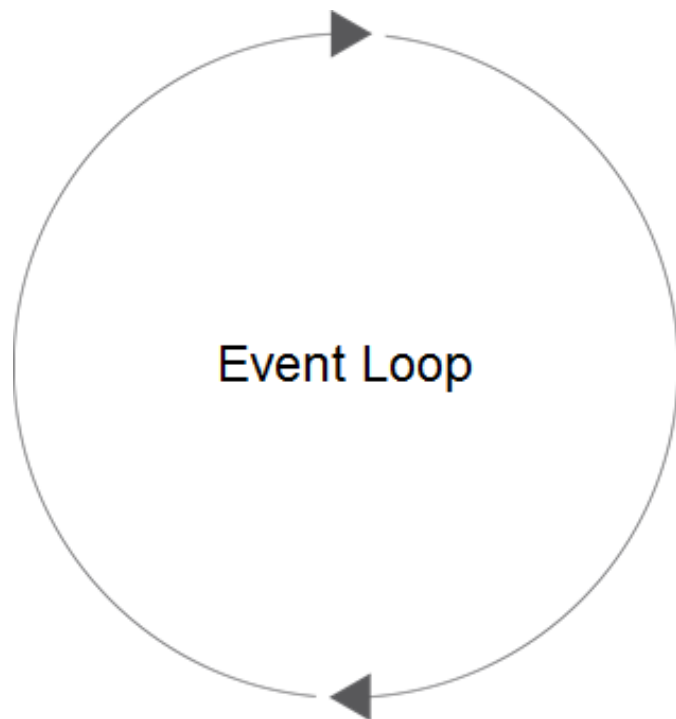


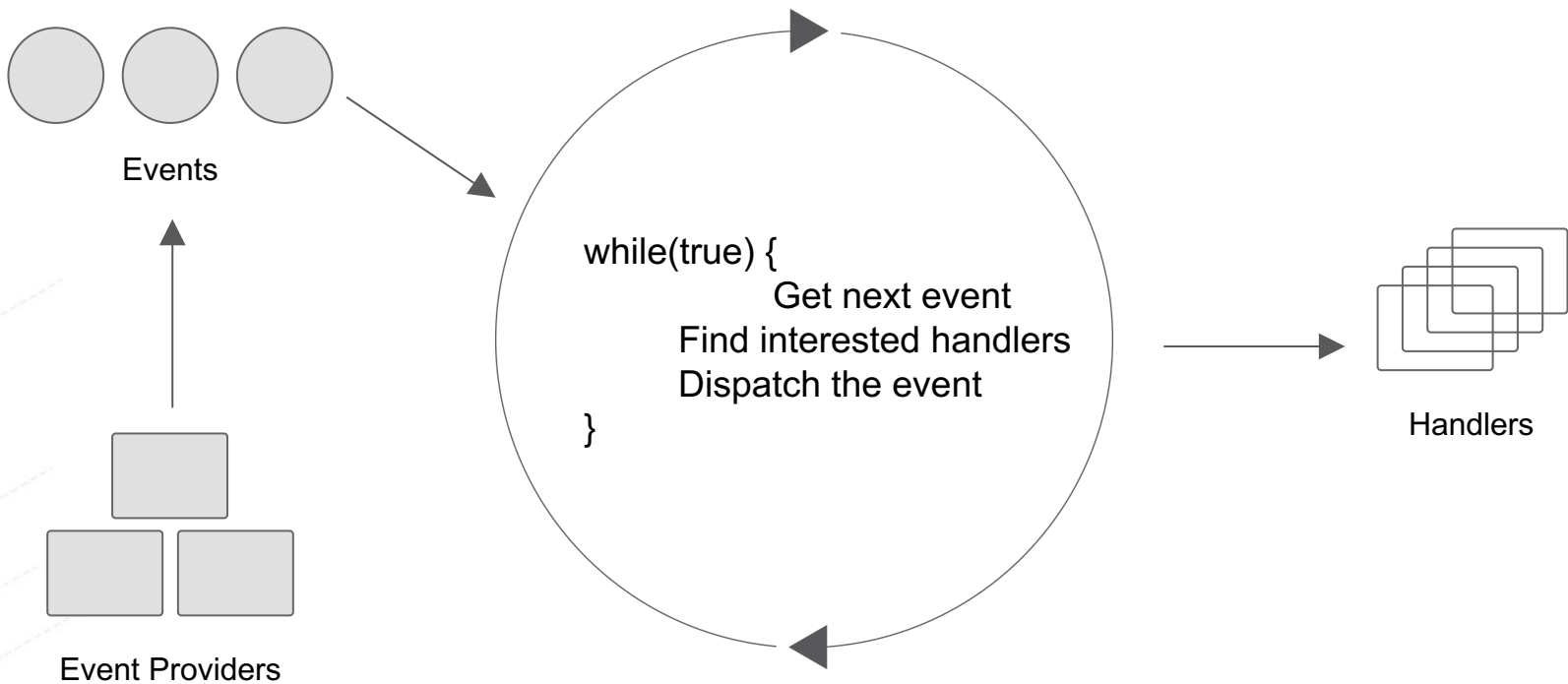
A Verticle is the programmable unit – a component

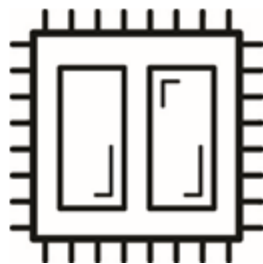
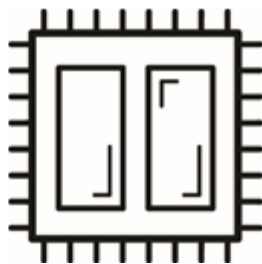
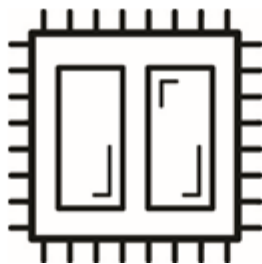
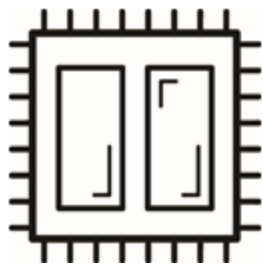
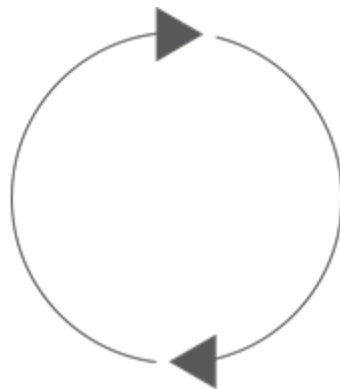
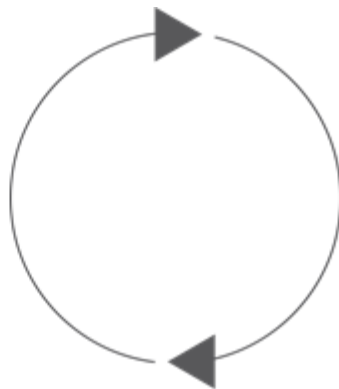
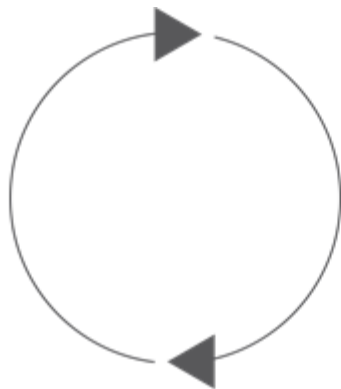
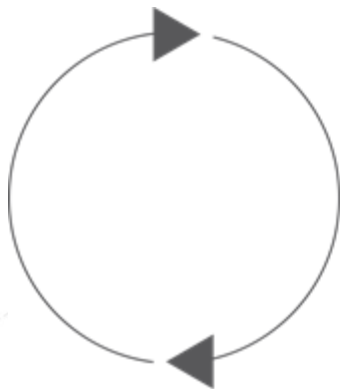
A Verticle is always executed on the same thread

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
```

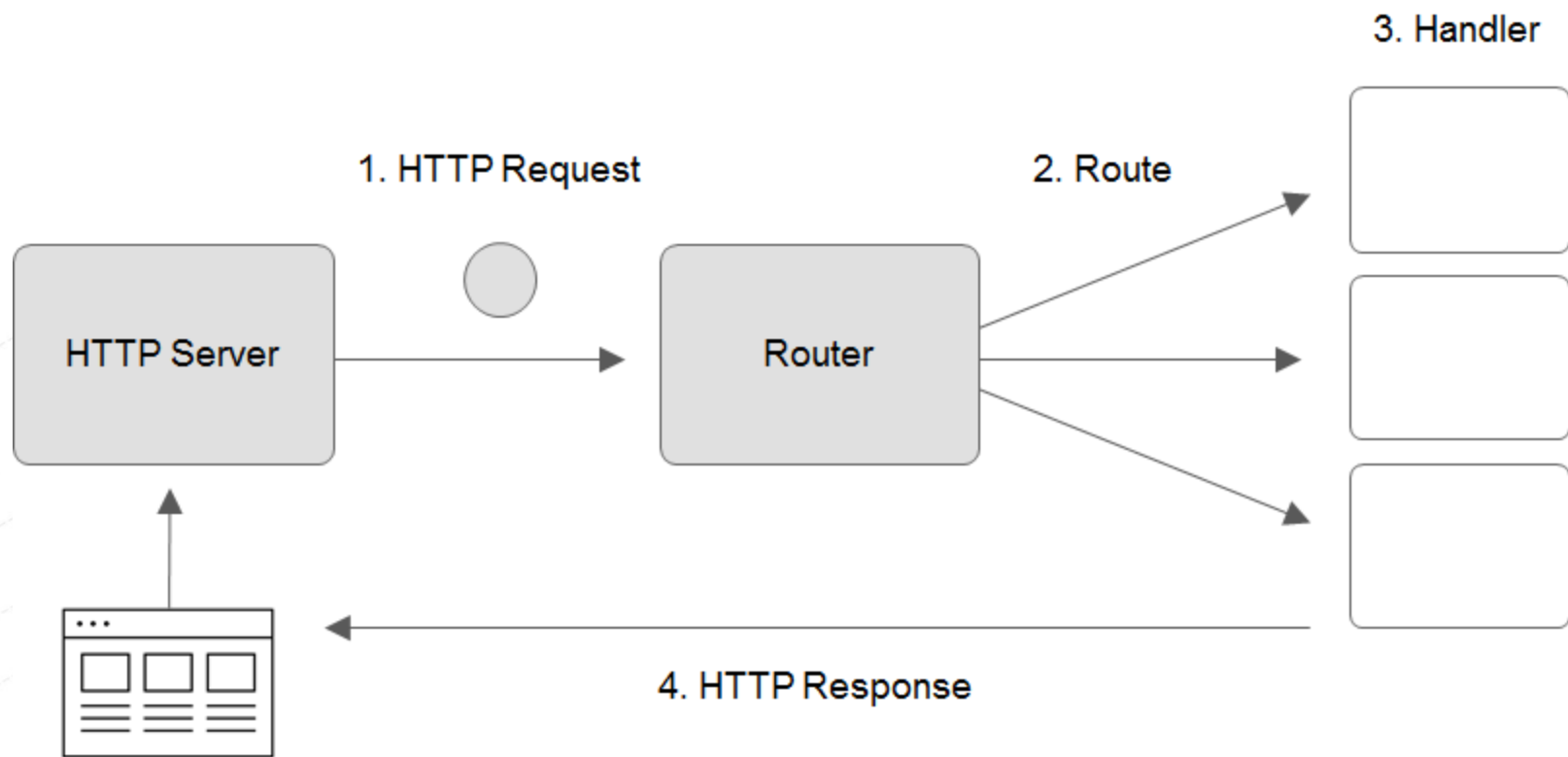
```
$ cd vertx-maven-starter
```

```
$ mvn package
```

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

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

Web Apps & REST

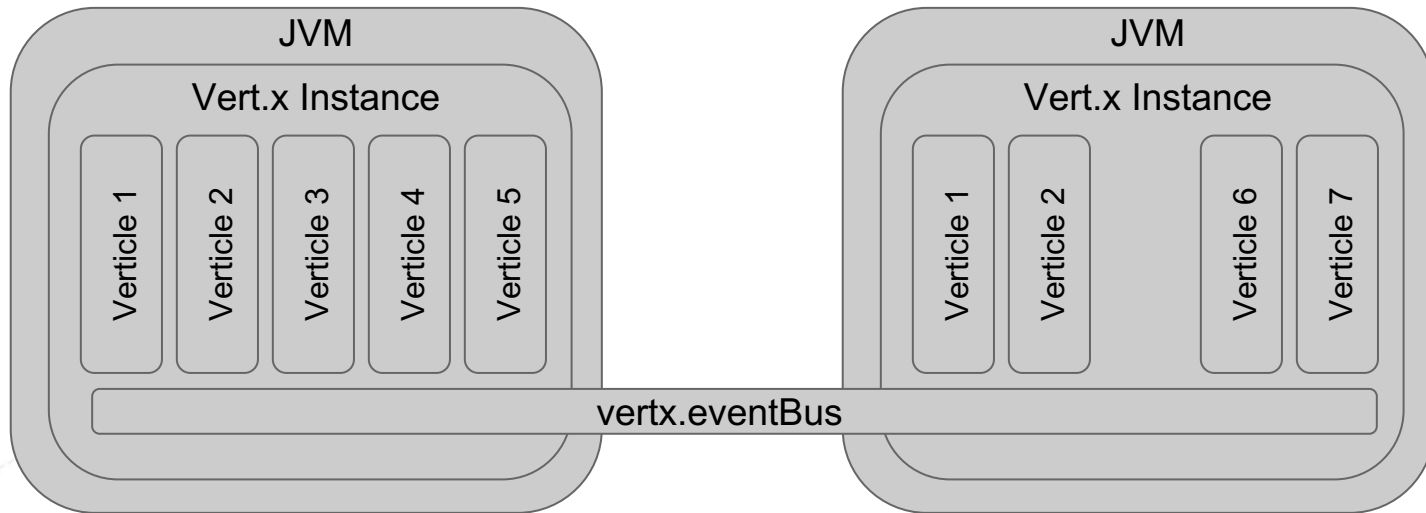


Vert.x HTTP Router

```
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

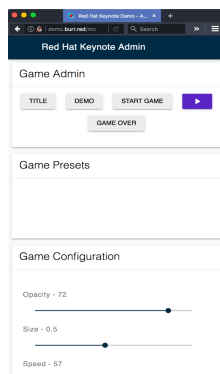
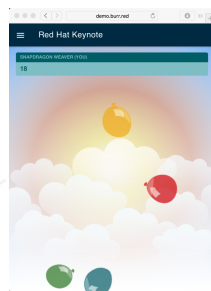
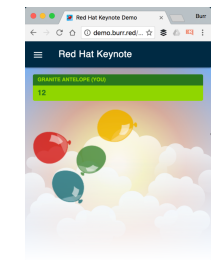
Demos

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

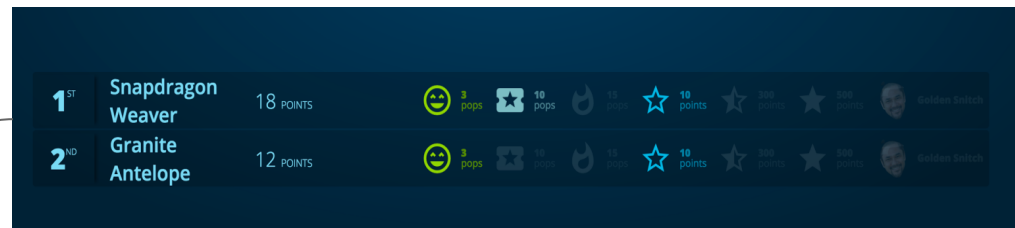
<https://github.com/cescoffier/vertx-microservices-examples>

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

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

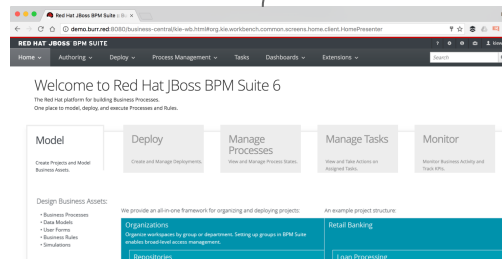


vertex-game-server



vertex-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/reactivemsa>

@burrsutter