Enseñando trucos de 20 años a Kotlin

Víctor Orozco

30 de mayo de 2019

@tuxtor

NABENIK

Kotlin

–

Kotlin

- Tipado estático y compilado
- Java inter-op
- OO + FP
- Null safety
- Extensions
- Operator overloading
- Data classes
- One line methods

NABENIK

Kotlin - Datos interesantes

- Effective Java - Inmutabilidad, builder, singleton, override, final by default, variance by generics
- Elvis - Groovy
- Inferencia de tipos - Scala
- Inmutabilidad - Scala
- Declaración de variables - Scala
- Manejo de Null - Groovy
- Closures y funciones - Groovy
- Google

Java - Muriendo desde 1995

- Sistemas legados (IBM)
- Retrocompatibilidad
- Release cadence (6 meses)
- Innovación constante en el ecosistema (Spring Boot, Micronaut, MicroProfile, GraalVM)
- Raw performance (Beam, Spark, Hadoop)
- Tooling - IDE, Maven, Drivers RDBMS
- JVM - (Twitter, Alibaba)
- OpenJDK

Java™

NABENIK

Java - Muriendo desde 1995

Bruno Borges @brunoborges · May 27

Stop trying to fit front-end development approach to back-end development.

Great back-end systems become legacy systems as in: they keep running for years with minor changes.

Great front-end systems become legacy systems as in: they haven't changed to adapt to user's needs.

NABENIK

Java - Muriendo desde 1995

TheServerSide
Your Enterprise Java Community

Java management ▼ | All Subtopics | Search TechTa

Transcipt of James' TSSJS 2011 Discussion about Java and the JVM

"At the core of the Java ecosystem is the JVM. Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less.

"What I really care about is the Java Virtual Machine as a concept, because that is the thing that ties it all together; it's the thing that makes Java the language possible; it's the thing that makes things work on all kinds of different platforms; and it makes all kinds of languages able to coexist.

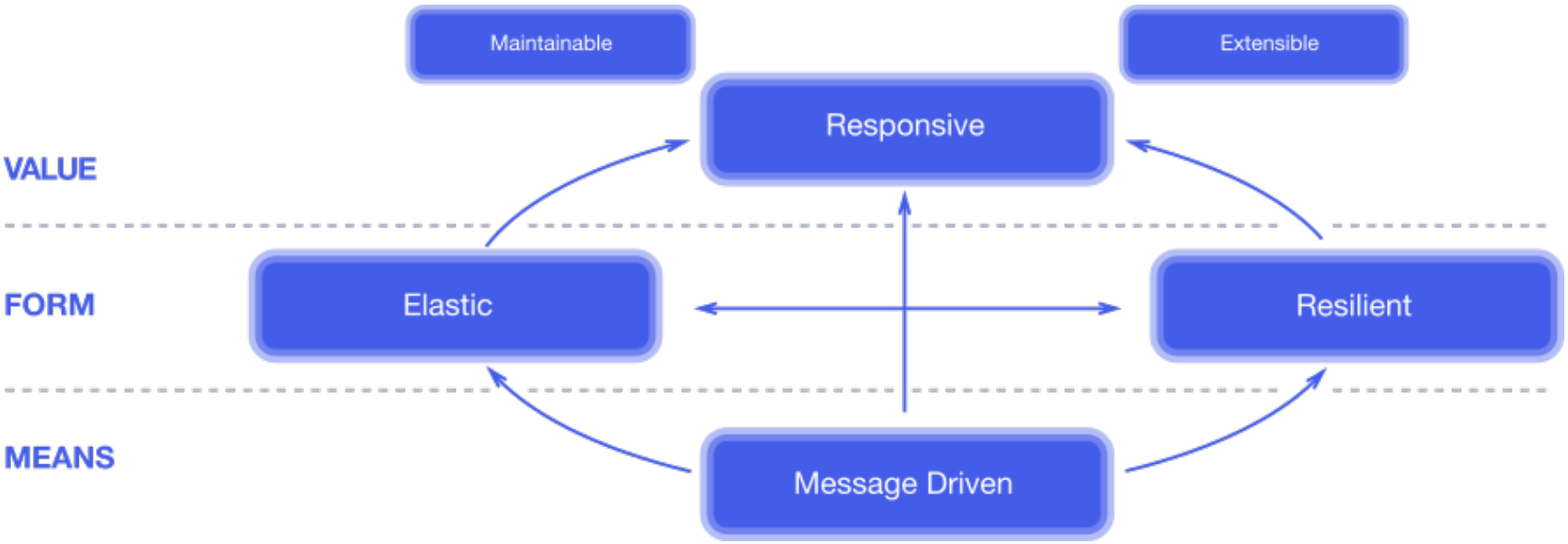
> Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less.
>
> James Gosling

NABENIK

¿Microservicios?

Reactive applications

Aplicaciones reactivas



Microservicios son una (de muchas) herramienta para creación de aplicaciones reactivas

Microservicios

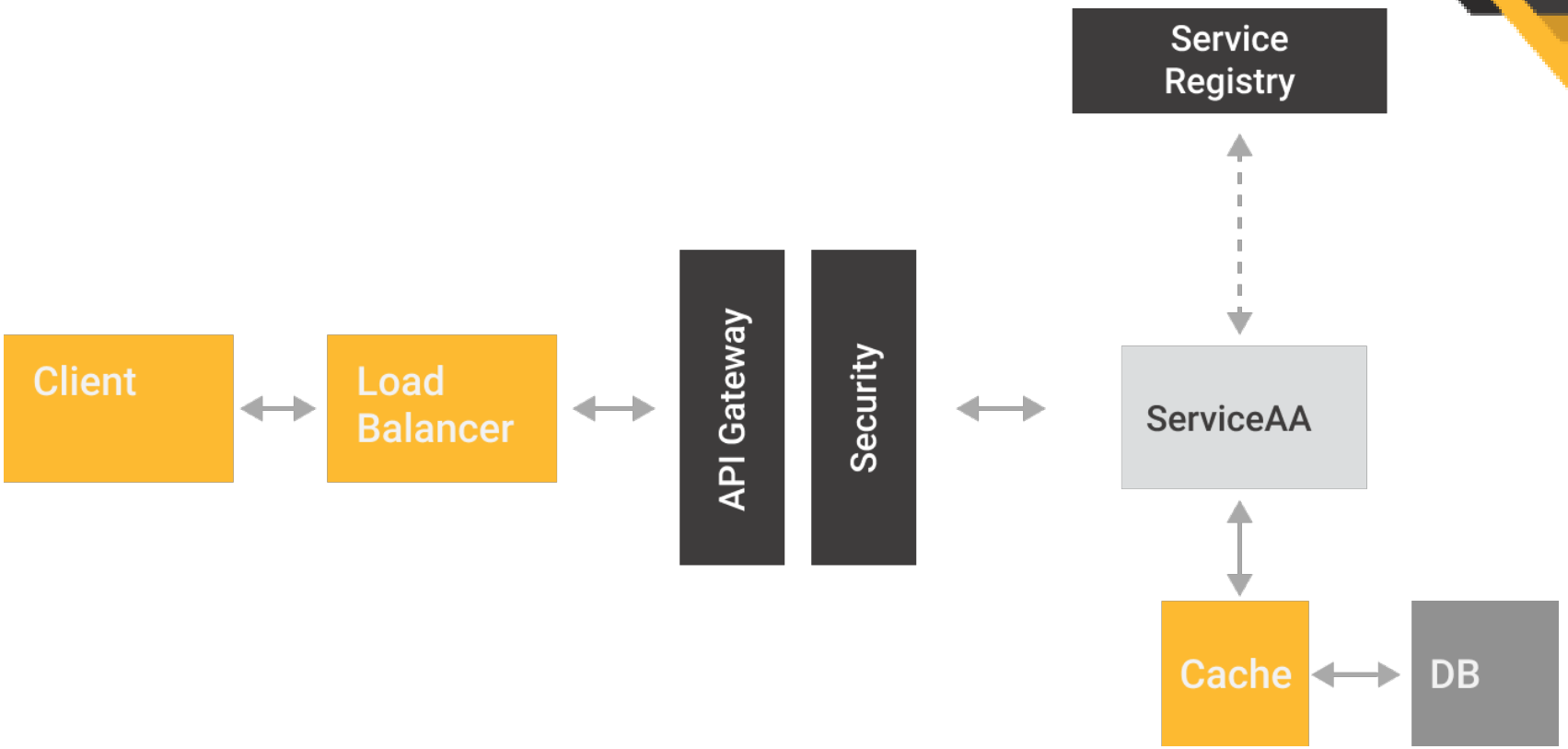


Figura 1: Microservicios

NABENIK

Microservicios - Java

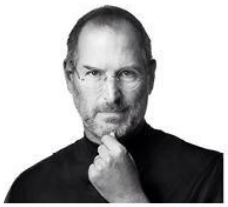
- DIY - Jooby, Javalin, Micronaut, Spark, Vert.x, Helidon SE
- Enterprise - Spring Boot, Microprofile (implementaciones)

NABENIK

Microservicios - Kotlin

- DIY - Jooby, Javalin, Micronaut, Spark, Vert.x, Helidon SE, **Ktor**
- Enterprise - Spring Boot, Microprofile (implementaciones)

NABENIK

	DEVELOPERS	DESIGNERS	PROJECT MANAGERS	QA	SYSADMINS
SEEN BY DEVELOPERS					
SEEN BY DESIGNERS					
SEEN BY PROJECT MANAGERS					
SEEN BY QA					
SEEN BY SYSADMINS					

Jakarta EE 8

NABENIK

Jakarta EE 8

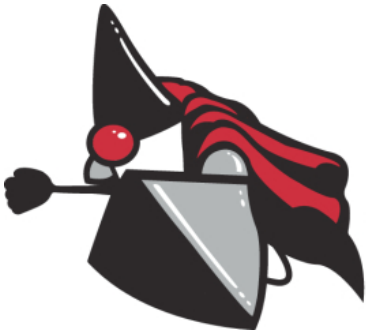
Java EE 8

Batch	Dependency Injection	JACC	JAXR	JSTL	Management
Bean Validation	Deployment	JASPIC	JMS	JTA	Servlet
CDI	EJB	JAX-RPC	JSF	JPA	Web Services
Common Annotations	EL	JAX-RS	JSON-P	JavaMail	Web Services Metadata
Concurrency EE	Interceptors	JAX-WS	JSP	Managed Beans	WebSocket
Connector	JSP Debugging	JAXB			
JSON-B	Security				

NABENIK

Jakarta EE 8 - Comunidad Java EE

JAKARTA EE



JAVA EE GUARDIANS

MICROPROFILE™

OPTIMIZING ENTERPRISE JAVA

NABENIK

Eclipse MicroProfile

Eclipse MicroProfile

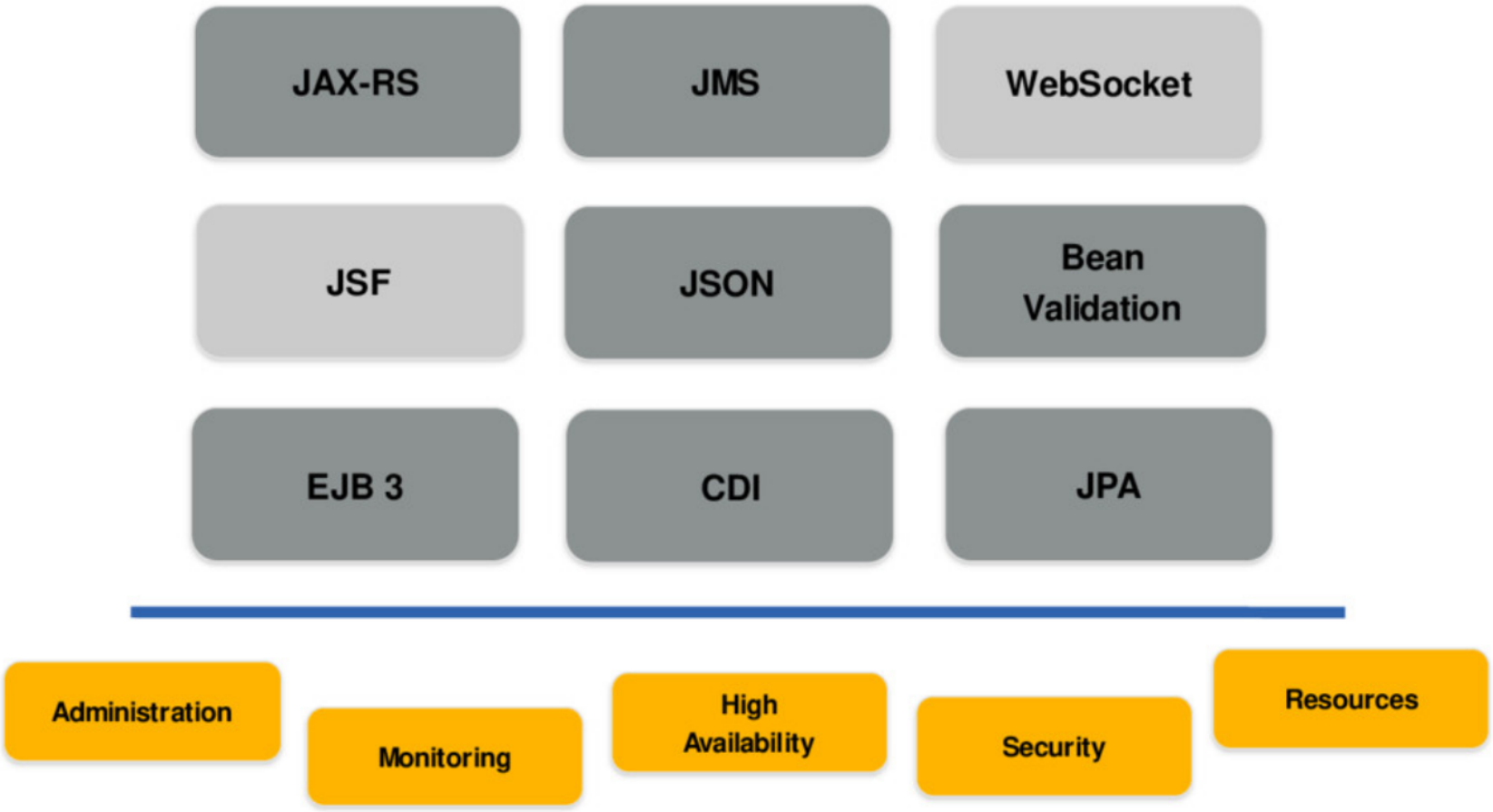
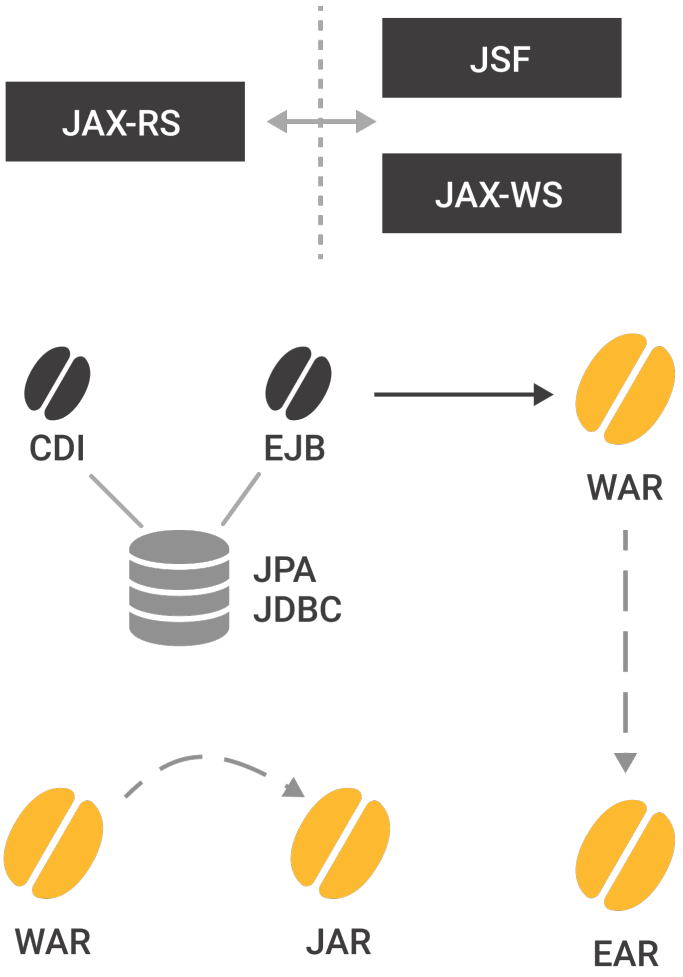


Figura 2: Credito: Reza Rahman

NABENIK

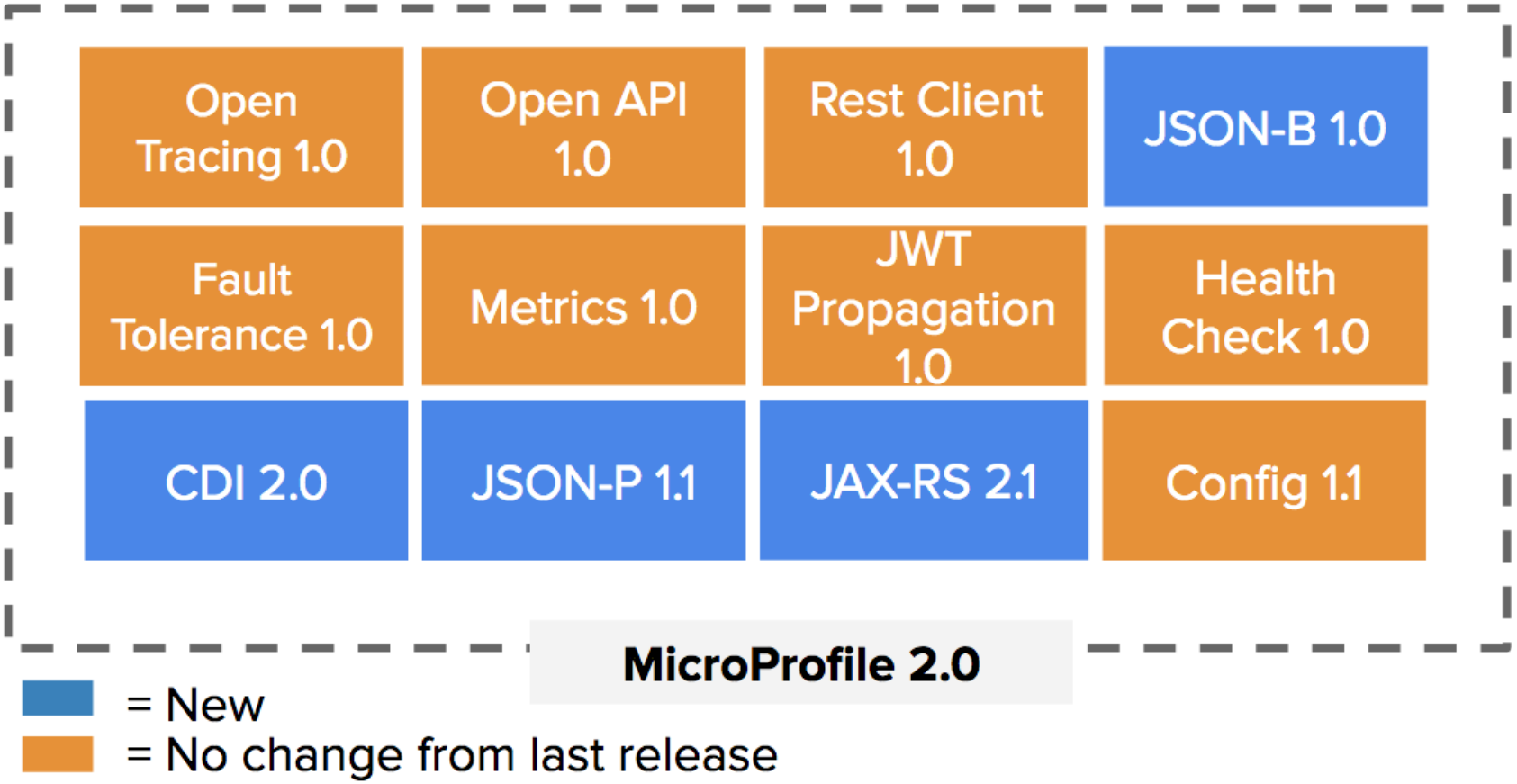
Eclipse MicroProfile



BV + JSON-P JSON-B + Security

NABENIK

Eclipse MicroProfile



NABENIK

Eclipse MicroProfile - Implementaciones

Bibliotecas

- SmallRye (Red Hat)
- Hammock
- Apache Geronimo
- Fujitsu Launcher

JEAS - Fat Jar, Uber Jar

- Dropwizard
- KumuluzEE
- Helidon (Oracle)
- Open Liberty (IBM)
- Apache Meecrowave
- Thorntail (Red Hat)
- Quarkus (Red Hat)
- Payara Micro

NABENIK

Eclipse MicroProfile - Implementaciones

Micro server - Thin War

- Payara Micro
- TomEE JAX-RS

Full server

- Payara Application Server
- JBoss Application Server / Wildfly Application Server
- WebSphere Liberty (IBM)

https://wiki.eclipse.org/MicroProfile/Implementation

NABENIK

Eclipse MicroProfile + Kotlin + Maven

Eclipse MicroProfile en Payara 5

```
<dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <type>pom</type>
        <version>2.0.1</version>
        <scope>provided</scope>
</dependency>
```

NABENIK

Kotlin en Maven - Dependencias

```
<dependency>
        <groupId>org.jetbrains.kotlin</groupId>
        <artifactId>kotlin-stdlib-jdk8</artifactId>
        <version>${kotlin.version}</version>
</dependency>
```

NABENIK

Kotlin en Maven - maven-compiler-plugin

```
<execution>
        <id>default-compile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>default-testCompile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>java-compile</id>
        <phase>compile</phase>
        <goals> <goal>compile</goal> </goals>
</execution>
<execution>
        <id>java-test-compile</id>
        <phase>test-compile</phase>
        <goals> <goal>testCompile</goal> </goals>
</execution>
```

Kotlin en Maven - kotlin-maven-plugin

```
<compilerPlugins>
<plugin>all-open</plugin>
</compilerPlugins>
...
<option>all-open:annotation=javax.ws.rs.Path</option>
<option>all-open:annotation=javax.enterprise.context.RequestScoped</option>
<option>all-open:annotation=javax.enterprise.context.SessionScoped</option>
<option>all-open:annotation=javax.enterprise.context.ApplicationScoped</option>
<option>all-open:annotation=javax.enterprise.context.Dependent</option>
<option>all-open:annotation=javax.ejb.Singleton</option>
<option>all-open:annotation=javax.ejb.Stateful</option>
<option>all-open:annotation=javax.ejb.Stateless</option>
```

NABENIK

Demo

NABENIK

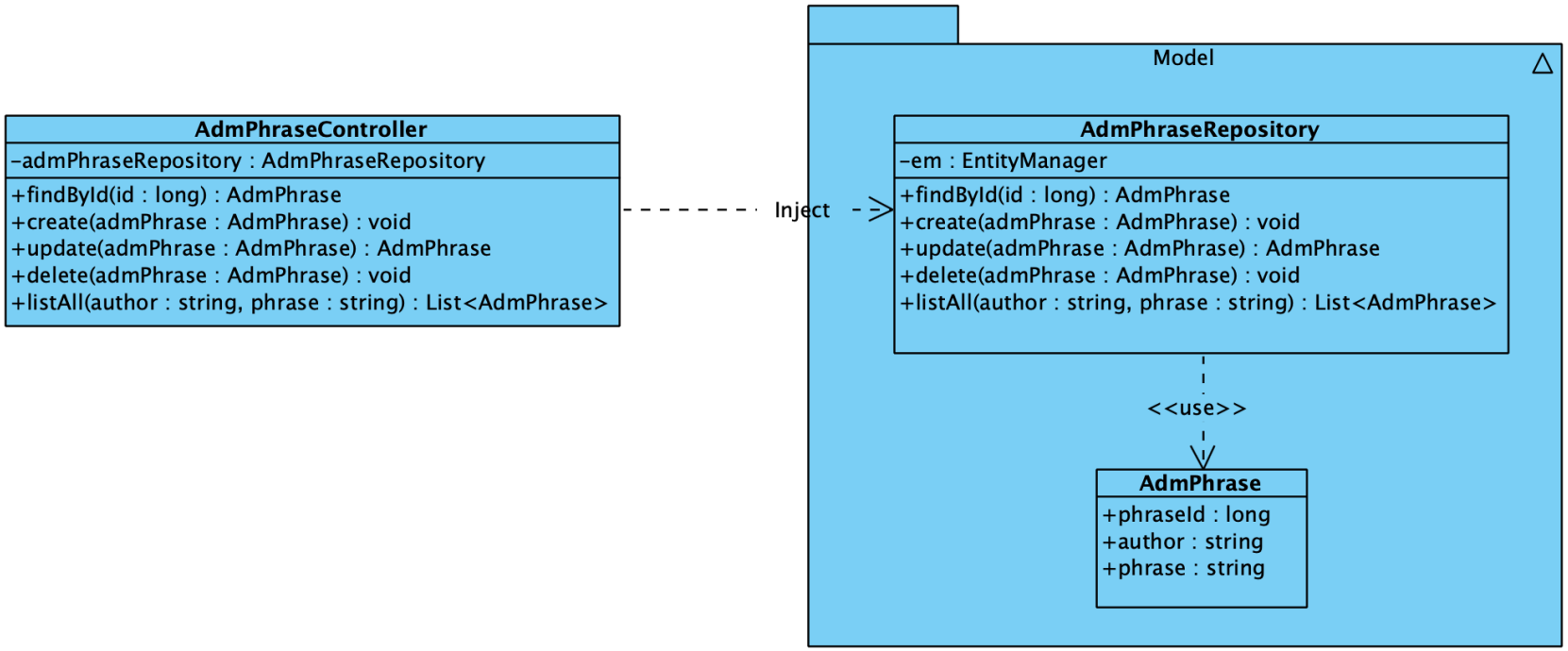
Kotlin + Jakarta EE + MicroProfile - Demo

- Kotlin 1.3
- Bibliotecas externas - SL4J, Flyway, PostgreSQL
- Jakarta EE 8 - EJB, JPA
- MicroProfile - CDI, JAX-RS, MicroProfile config
- Testing - Arquillian, JUnit, Payara Embedded

https://dzone.com/articles/
the-state-of-kotlin-for-jakarta-eemicroprofile-tra
https://github.com/tuxtor/integrum-ee

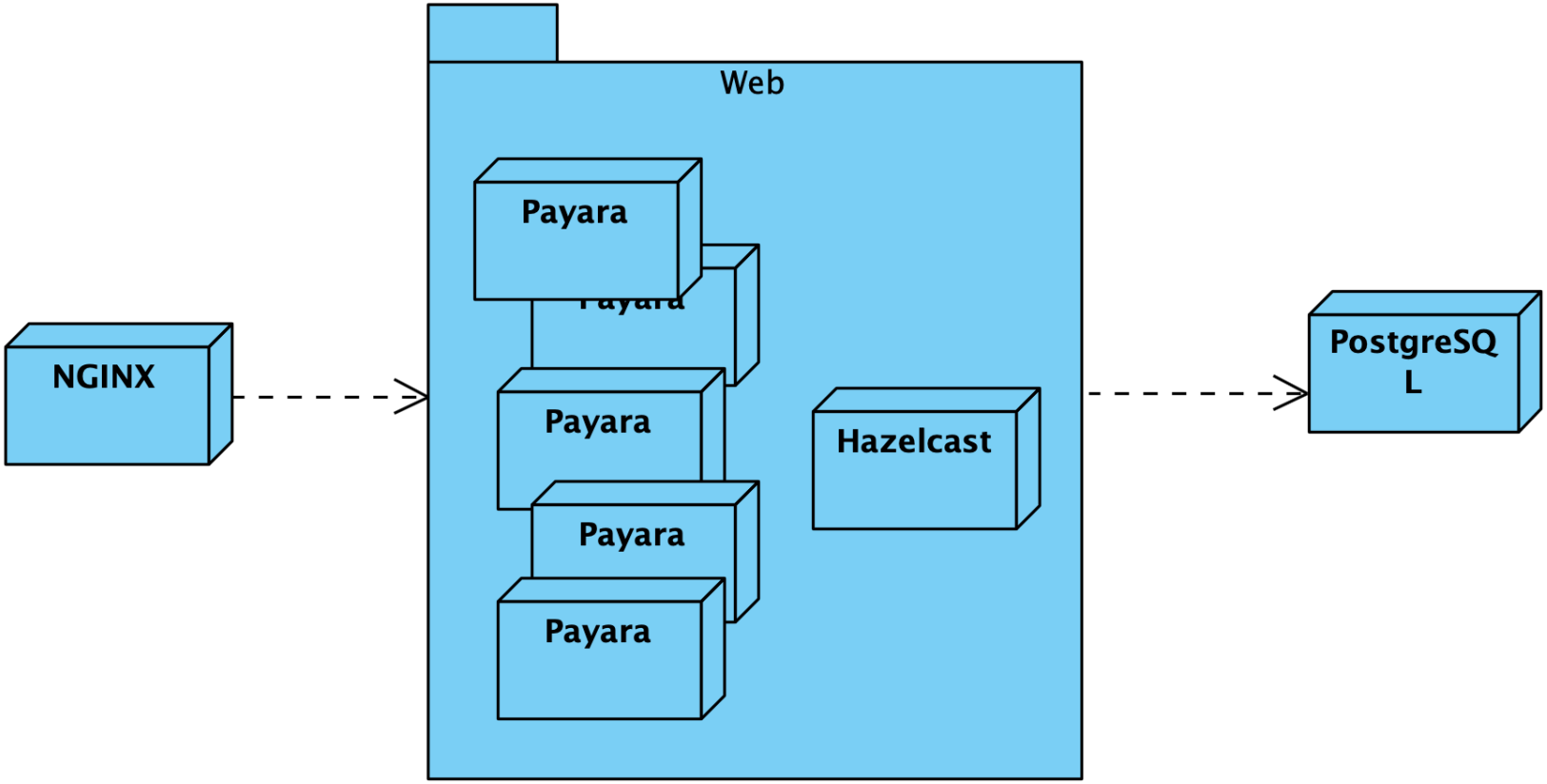
NABENIK

Kotlin + Jakarta EE + MicroProfile - Demo



NABENIK

Kotlin + Jakarta EE + MicroProfile - Demo



NABENIK

Kotlin - Entidad JPA

```
@Entity
@Table(name = "adm_phrase")
@TableGenerator(...)
data class AdmPhrase(
        @Id
        @GeneratedValue(strategy = GenerationType.TABLE,
                generator = "admPhraseIdGenerator")
        @Column(name = "phrase_id")
        var phraseId:Long? = null,
        var author:String = "",
        var phrase:String = ""
)
```

Data Clases, Nullable Types

NABENIK

Kotlin - Repositorio CDI

```
@RequestScoped
class AdmPhraseRepository {

        @Inject
        private lateinit var em:EntityManager

        ...

}
```

Lateinit (nullable type)

NABENIK

Kotlin - Repositorio CDI

```
fun create(admPhrase:AdmPhrase) = em.persist(admPhrase)

fun update(admPhrase:AdmPhrase) = em.merge(admPhrase)

fun findById(phraseId: Long) =
em.find(AdmPhrase::class.java, phraseId)

fun delete(admPhrase: AdmPhrase) = em.remove(admPhrase)
. . .
```

Single expression functions (One line methods)

NABENIK

Kotlin - Repositorio CDI

```
fun listAll(author: String, phrase: String):
        List<AdmPhrase> {

        val query = """SELECT␣p␣FROM␣AdmPhrase␣p
␣␣␣␣␣␣␣␣where␣p.author␣LIKE␣:author
␣␣␣␣␣␣␣␣and␣p.phrase␣LIKE␣:phrase
␣␣␣␣␣␣␣␣"""

        return em.createQuery(query, AdmPhrase::class.java)
                .setParameter("author", "%$author%")
                .setParameter("phrase", "%$phrase%")
                .resultList
}
```

Multiline String, mutable declaration

Kotlin - Controlador JAX-RS

```
@Path("/phrases")
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
class AdmPhraseController{

        @Inject
        private lateinit var admPhraseRepository: AdmPhraseReposit

        @Inject
        private lateinit var logger: Logger
        ...

}
```

NABENIK

Kotlin - Controlador JAX-RS

```
@GET
fun findAll(
@QueryParam("author") @DefaultValue("%") author: String,
@QueryParam("phrase") @DefaultValue("%") phrase: String) =
        admPhraseRepository.listAll(author, phrase)

@GET
@Path("/{id:[0-9][0-9]*}")
fun findById(@PathParam("id") id:Long) =
        admPhraseRepository.findById(id)

@PUT
fun create(phrase: AdmPhrase): Response {
        admPhraseRepository.create(phrase)
        return Response.ok().build()
}
```

NABENIK

Kotlin - Controlador JAX-RS

```
@POST
@Path("/{id:[0-9][0-9]*}")
fun update(@PathParam("id") id: Long?, phrase: AdmPhrase)
        :Response {
        if(id != phrase.phraseId)
                return Response.status(Response.Status.NOT_FOUND).

        val updatedEntity = admPhraseRepository.update(phrase)
        return Response.ok(updatedEntity).build()
}

@DELETE
@Path("/{id:[0-9][0-9]*}")
fun delete(@PathParam("id") id: Long): Response {
        val updatedEntity = admPhraseRepository.findById(id) ?:
        return Response.status(Response.Status.NOT_FOUND).build()
        admPhraseRepository.delete(updatedEntity)
        return Response.ok().build()
}
```

Elvis operator as expression

12 factores cloud native (Heroku)

Microprofile

- Config
- Backing service
- Disposability

Cloud

- Codebase (Git-Flow)
- Dependencies (Maven)
- Build, Release, Run
- Processes (Pipelines)
- Port binding
- Concurrency (Docker - k8s)
- Dev / Prod parity
- Logs
- Admin process

NABENIK

Oracle Cloud

```
<groupId>io.fabric8</groupId>
<artifactId>docker-maven-plugin</artifactId>
<version>0.30.0</version>
...
<image>
        <name>iad.ocir.io/tuxtor/microprofile/integrum-ee</name>
        <build>
                <dockerFile>${project.basedir}/Dockerfile</dockerF
        </build>
</image>
```

NABENIK

Oracle Cloud

Registry

Create Repository

tuxtor

- microprofile (Public)
- microprofile/hello-ee
- microprofile/hello-escalable
- microprofile/home-ee
- microprofile/integrum-ee
  - 1
  - latest
- microprofile/jvmservice
- microprofile/omdb-demo
- microprofile/payara-demo (Public)

microprofile/integrum-ee

User: ...42db3y5hmh4zajq Show Copy

Created: a month ago

Access: Private

Size: 138.19 MB

Last Push: 39 minutes ago

Readme

No readme has been created yet for this repository.

NABENIK

Oracle Cloud

MENU

ORACLE
Cloud Infrastructure

Compute » Instances » Instance Details

RUNNING

instance-20181206-0243

Create Custom Image | Start | Stop | Reboot | Terminate | Apply Tag(s)

Instance Information | Tags

Instance Information

Availability Domain: hgWe:US-ASHBURN-AD-1

Fault Domain: FAULT-DOMAIN-2

Region: iad

Shape: VM.Standard2.1

Virtual Cloud Network: vcn20181206044411

Maintenance Reboot: -

Primary VNIC Information

NABENIK

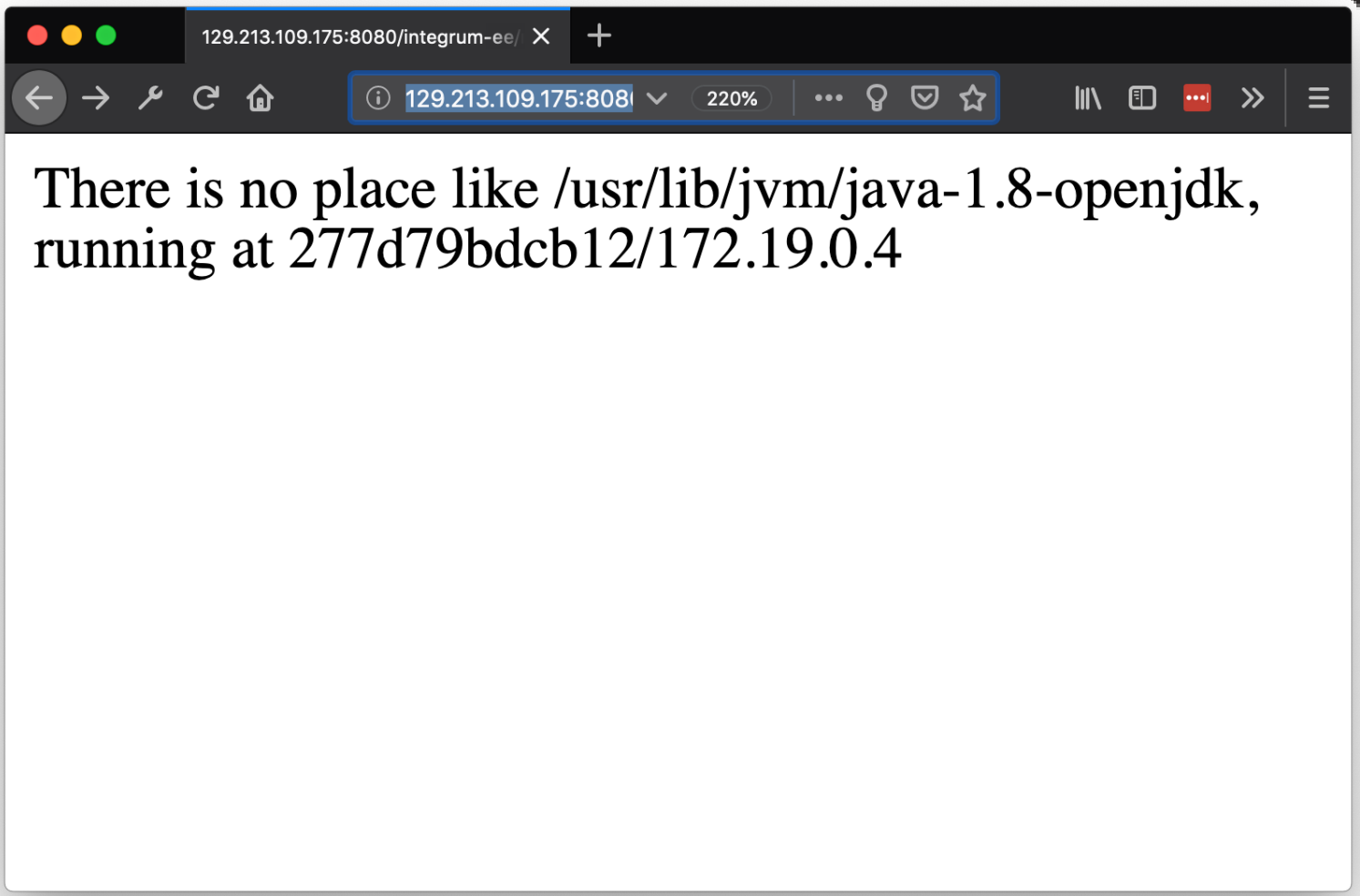
Oracle Cloud

1. bash

```
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:build
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- docker-maven-plugin:0.30.0:build (default-cli) @ integrum-ee ---
[INFO] Building tar: /Users/tuxtor/GitHub/integrum-ee/target/docker/iad.ocir.io/tuxtor/microprofile/integrum-ee/t
mp/docker-build.tar
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Created docker-build.tar in 145 milliseconds
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Built image sha256:26156
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Removed old image sha256:a2361
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.765 s
[INFO] Finished at: 2019-05-30T16:39:15-06:00
[INFO] Final Memory: 17M/239M
[INFO] ------------------------------------------------------------------------
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:push
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- docker-maven-plugin:0.30.0:push (default-cli) @ integrum-ee ---
[INFO] DOCKER> The push refers to repository [iad.ocir.io/tuxtor/microprofile/integrum-ee]
67b8beb0da59: Pushed
cfa8d5eeea6b: Layer already exists
aae73696d349: Layer already exists
714e1dd7c2e4: Layer already exists
344fb4b275b7: Layer already exists
bcf2f368fe23: Layer already exists
[INFO] DOCKER> latest: digest: sha256:066a2a86fca1ac8b2918603e22a2178e2d2eaff8c0cbb5f861f831a51984eb77 size: 1578
[INFO] DOCKER> Pushed iad.ocir.io/tuxtor/microprofile/integrum-ee in 26 seconds
```

NABENIK

Oracle Cloud



NABENIK

Kotlin

Ventajas

- Código más conciso
- Soporte real Java inter-op
- Aprovechar a personal Android para backend
- Un lenguaje para dominar todo

Desventajas

- IntelliJ IDEA Ultimate (monolitos)
- Requiere mejores programadores (más convenciones)
- Tiempo de compilación
- No es una buena idea utilizar corutinas en entornos con managed threads

Víctor Orozco

- vorozco@nabenik.com
- @tuxtor
- http://www.nabenik.com

NABENIK