

JAVA 9, 10, 11

<http://bit.ly/java91011-codemash-2020>

<https://github.com/cjudd/java-91011-workshop/>

Christopher M. Judd

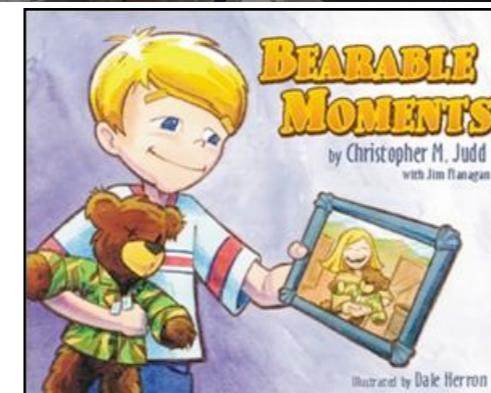
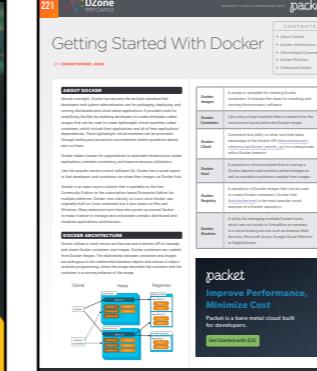
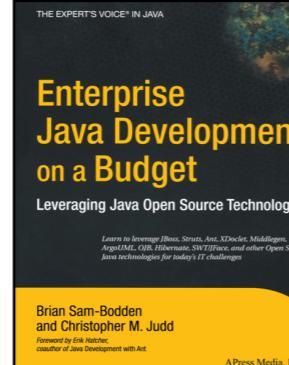
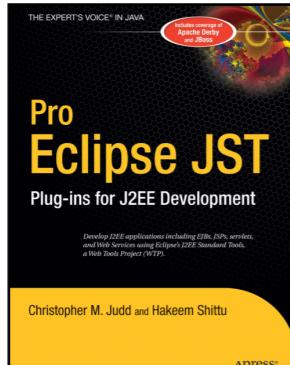
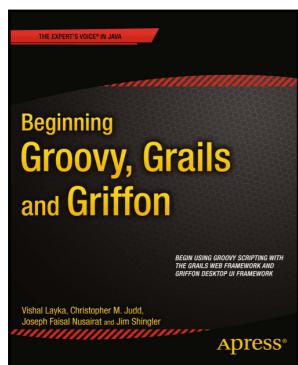


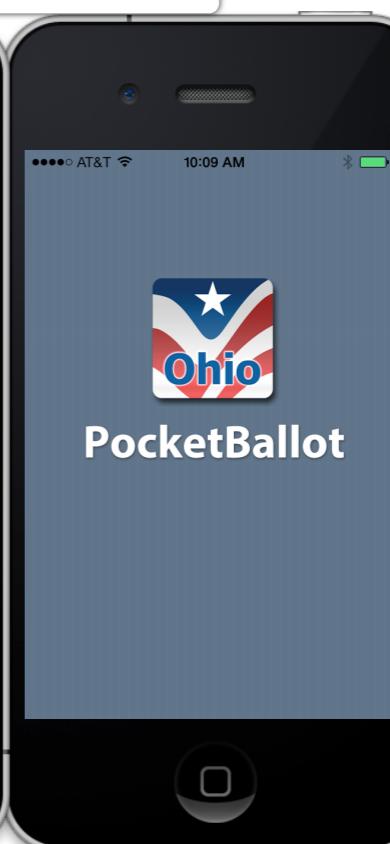
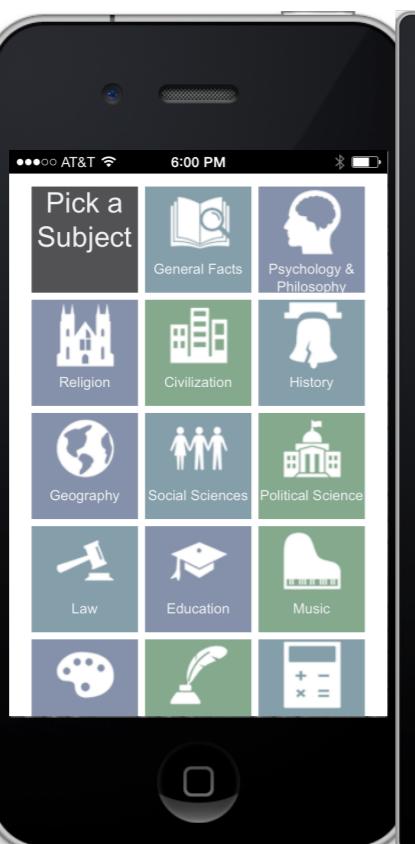
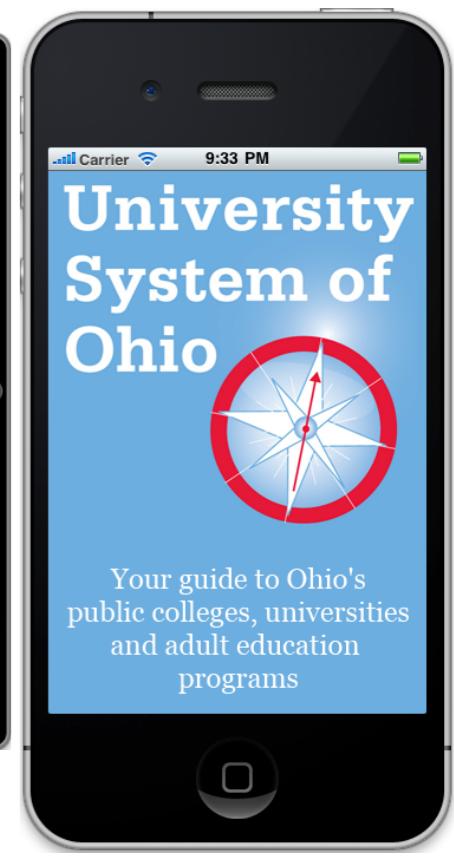
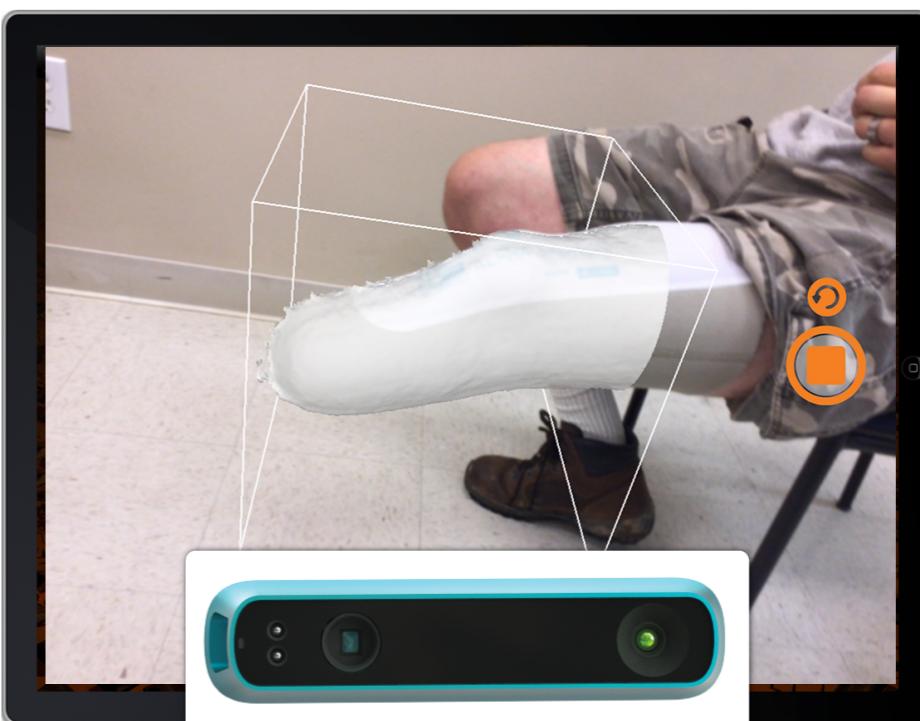
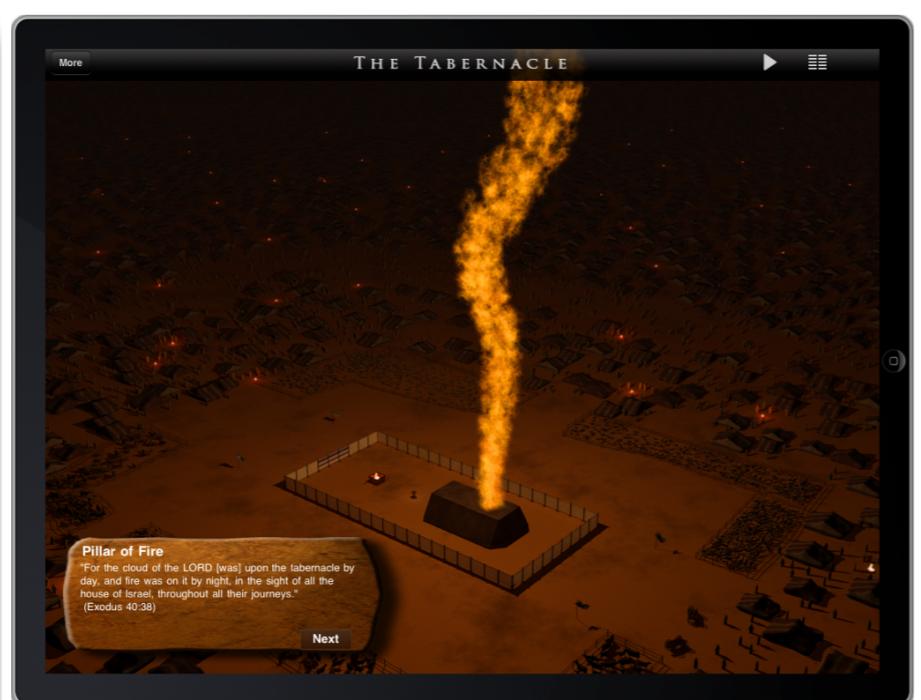
Christopher M. Judd

CTO and Partner at



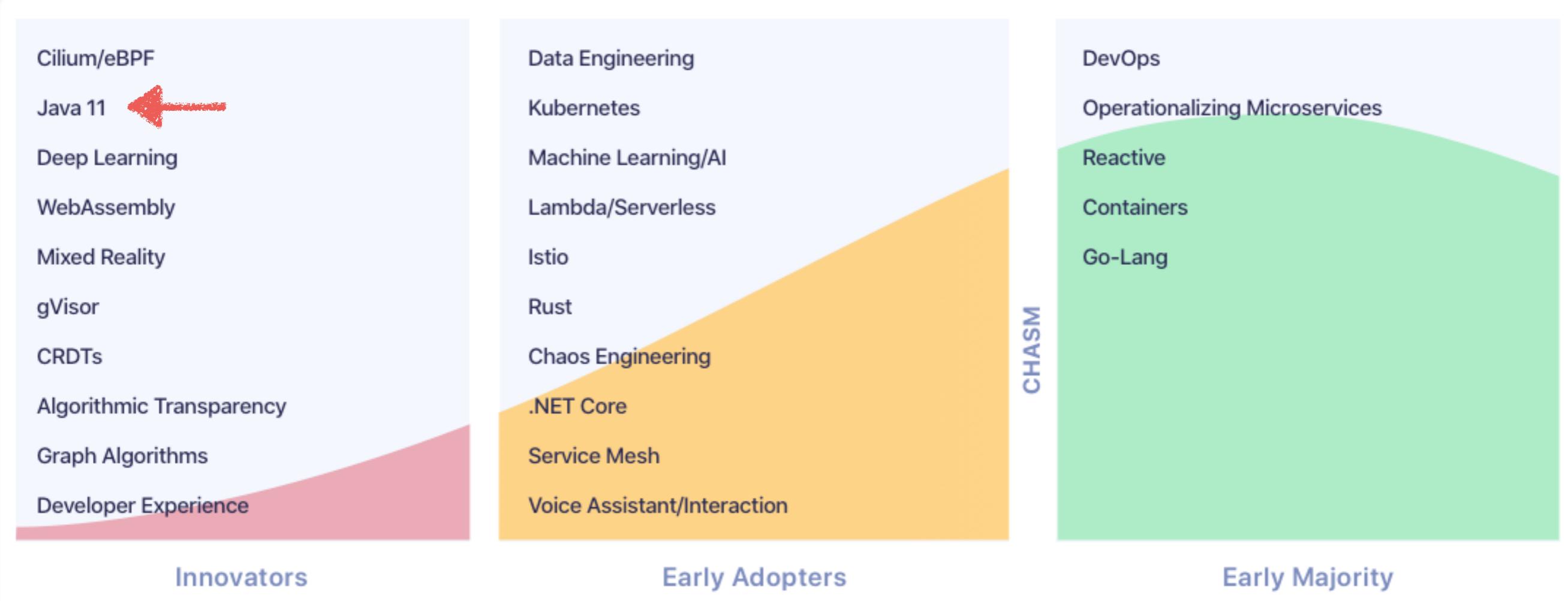
Central Ohio Java Users Group leader





Bleeding-edge for the Enterprise

Bring trends from innovator and early adopter companies home to your team





< Java 11

Proprietary Bits
Monolith
Long Unpredictable Release

> Java 11

Completely Open Source
Modular
Regularly Scheduled Trains
New Long Term Support Model
Merged Hotspot & JRocket
Removal of Stuff

JDK 11 Releases Not Secure | jdk.java.net/11/

jdk.java.net **JDK 11 Releases**

GA Releases
JDK 12

Early-Access
Releases
JDK 13

Jpackage
OpenJFX
Panama
Valhalla
JMC

Reference
Implementations
Java SE 12
Java SE 11
Java SE 10
Java SE 9
Java SE 8
Java SE 7

Feedback
Report a bug

Archive

The JDK 11 Early Access Program has concluded. Please visit Java SE Downloads for production ready builds.

Older releases, **which do not include the most up to date security vulnerability fixes and are no longer recommended for use in production**, remain available in the OpenJDK Archive.

© 2019 Oracle Corporation and/or its affiliates
Terms of Use · Privacy · Trademarks

ORACLE

Archived OpenJDK GA Release X +

Not Secure | jdk.java.net/archive/

jdk.java.net **Archived OpenJDK General-Availability Releases**

GA Releases
[JDK 12](#)
[Early-Access Releases](#)
[JDK 13](#)
[Jpackage](#)
[OpenJFX](#)
[Panama](#)
[Valhalla](#)
[JMC](#)
Reference
Implementations
[Java SE 12](#)
[Java SE 11](#)
[Java SE 10](#)
[Java SE 9](#)
[Java SE 8](#)
[Java SE 7](#)
Feedback
[Report a bug](#)
Archive

This page is an archive of previously released builds of the JDK licensed under the GNU General Public License, version 2, with Classpath Exception.

WARNING: These older versions of the JDK are provided to help developers debug issues in older systems. **They are not updated with the latest security patches and are not recommended for use in production.**

Releases

12 GA

Windows	64-bit	zip (sha256) 188M
Mac	64-bit	tar.gz (sha256) 182M
Linux	64-bit	tar.gz (sha256) 189M

11.0.2

Windows	64-bit	zip (sha256) 179 MB
Mac	64-bit	tar.gz (sha256) 174 MB
Linux	64-bit	tar.gz (sha256) 179 MB

11.0.1

Windows	64-bit	zip (sha256) 179 MB
Mac	64-bit	tar.gz (sha256) 174 MB
Linux	64-bit	tar.gz (sha256) 179 MB

11 GA

Windows	64-bit	zip (sha256) 179 MB
Mac	64-bit	tar.gz (sha256) 175 MB
Linux	64-bit	tar.gz (sha256) 179 MB

```
$ java --version
openjdk 11.0.3 2019-04-16
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.3+7)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.3+7, mixed mode)
```

Archived OpenJDK GA Release X +

Not Secure | jdk.java.net/archive/ 

jdk.java.net

GA Releases
JDK 12

Early-Access
Releases
JDK 13

Jpackage
OpenJFX
Panama
Valhalla
JMC

Reference
Implementations
Java SE 12
Java SE 11
Java SE 10
Java SE 9
Java SE 8
Java SE 7

Feedback
Report a bug

Archive

Archived OpenJDK GA Availability Releases

This page is an archive of the Java Platform, Standard Edition (Java SE) releases of the JDK licensed under the GNU General Public License, Version 2 (GPLv2).
WARNING: These releases are provided for historical reference only. Issues in older releases are not tracked by the Java team, and developers debug security patches and fix critical bugs.

Releases

Version	Platform	Architecture	File Type	Size
12 GA	Windows			
	Mac			
	Linux	64-bit	zip (sha256)	179 MB
11.0.2	Windows			
	Mac			
	Linux	64-bit	tar.gz (sha256)	179 MB
11 GA	Windows			
	Mac			
	Linux	64-bit	zip (sha256)	179 MB
	Mac	64-bit	tar.gz (sha256)	179 MB
	Linux	64-bit	tar.gz (sha256)	179 MB

JDK DISTRIBUTIONS

JDK Enhancement Proposal (JEP)

JEP 2: JEP Template

Owner Mark Reinhold

Type Process

Scope JDK

Status Active

Discussion discuss at openjdk dot java dot net

Created 2011/06/23 20:00

Updated 2018/03/30 15:47

Issue 8046186

Summary

// REQUIRED -- Provide a short summary of the proposal, at most one or two sentences.

Goals

// What are the goals of this proposal?

Non-Goals

// Describe any goals you wish to identify specifically as being out of scope for this proposal.

Success Metrics

Motivation

// Why should this work be done?

Description

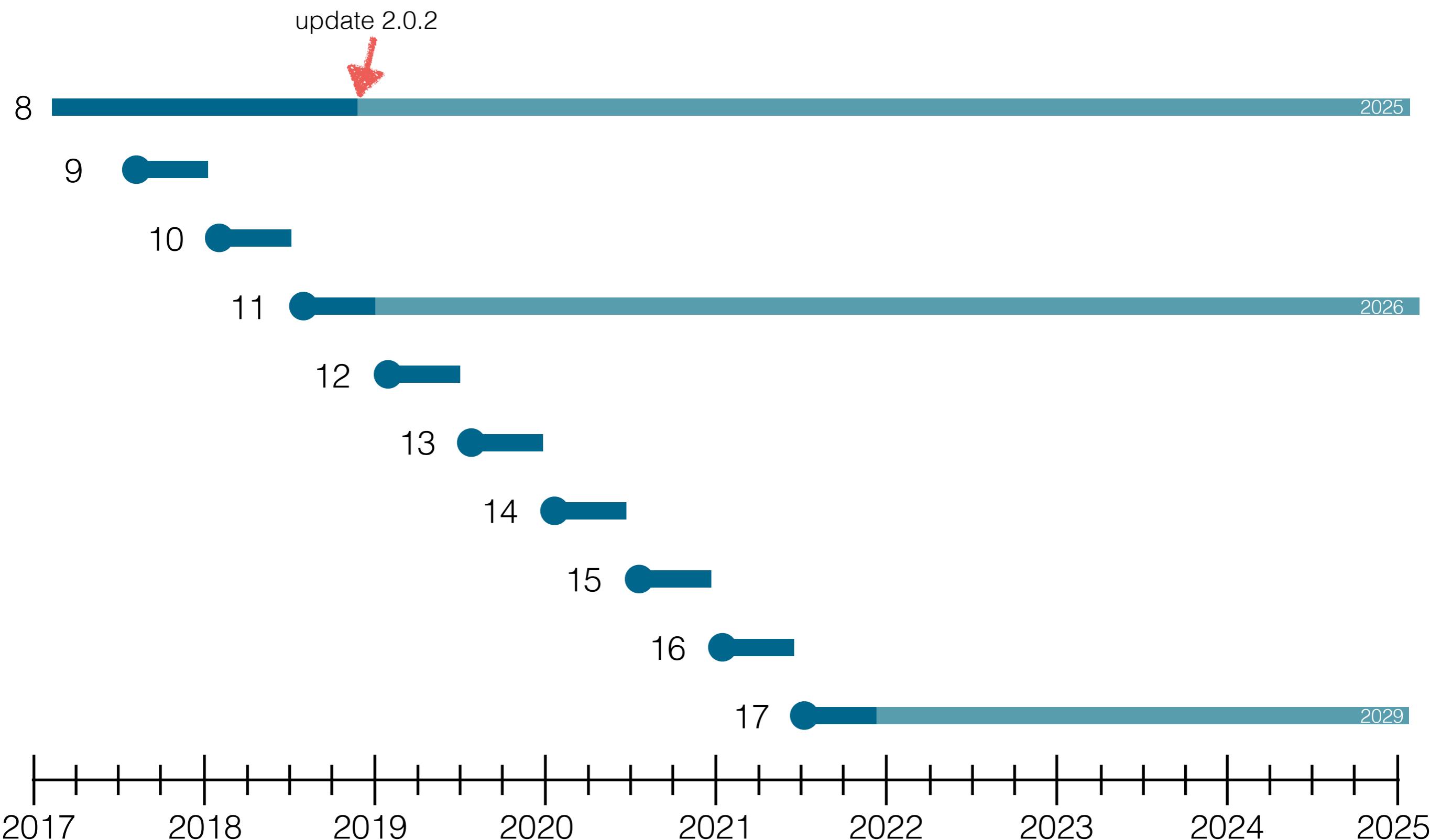
// REQUIRED -- Describe the enhancement in detail.

Alternatives

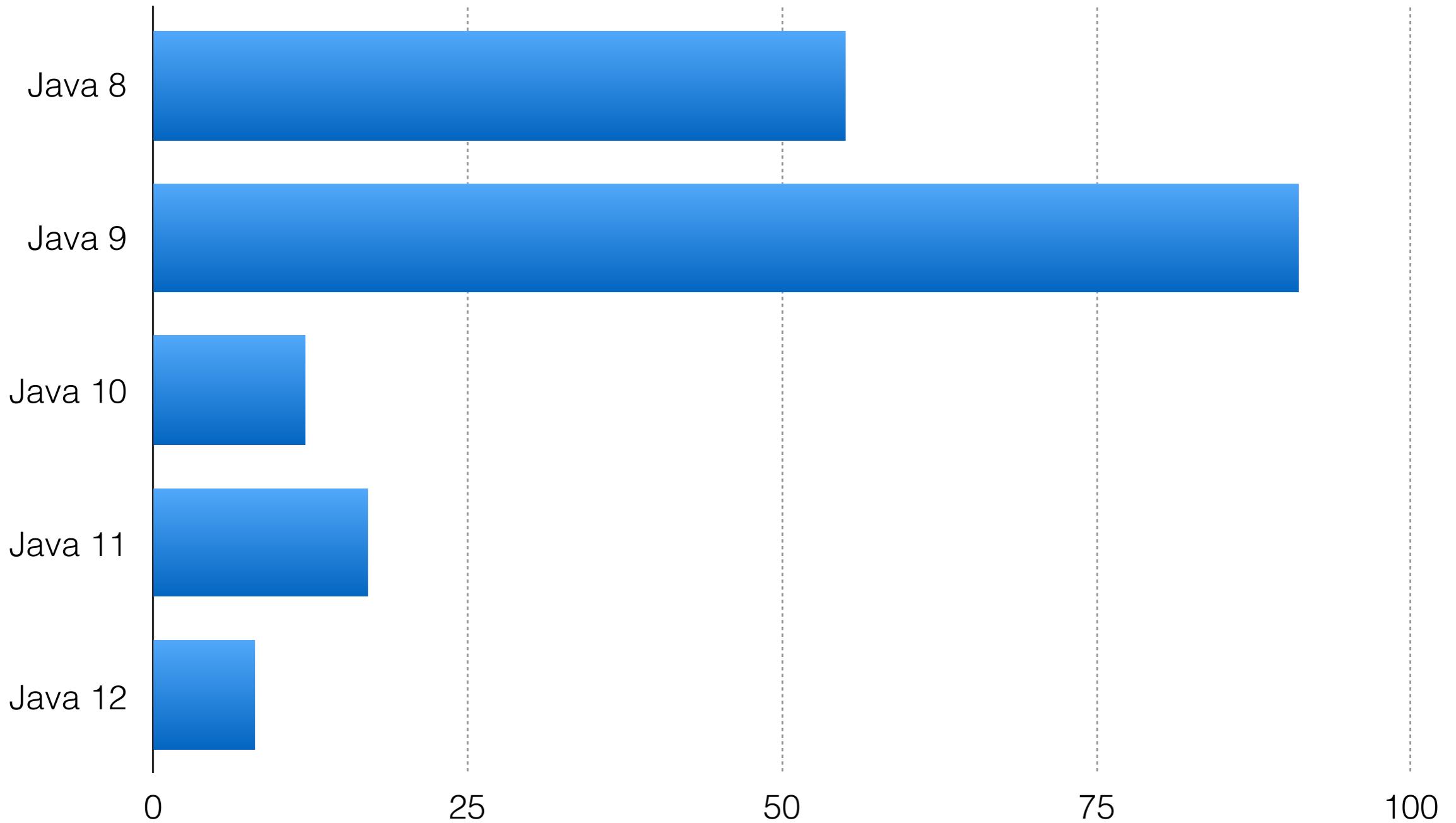
Testing

Risks and Assumptions

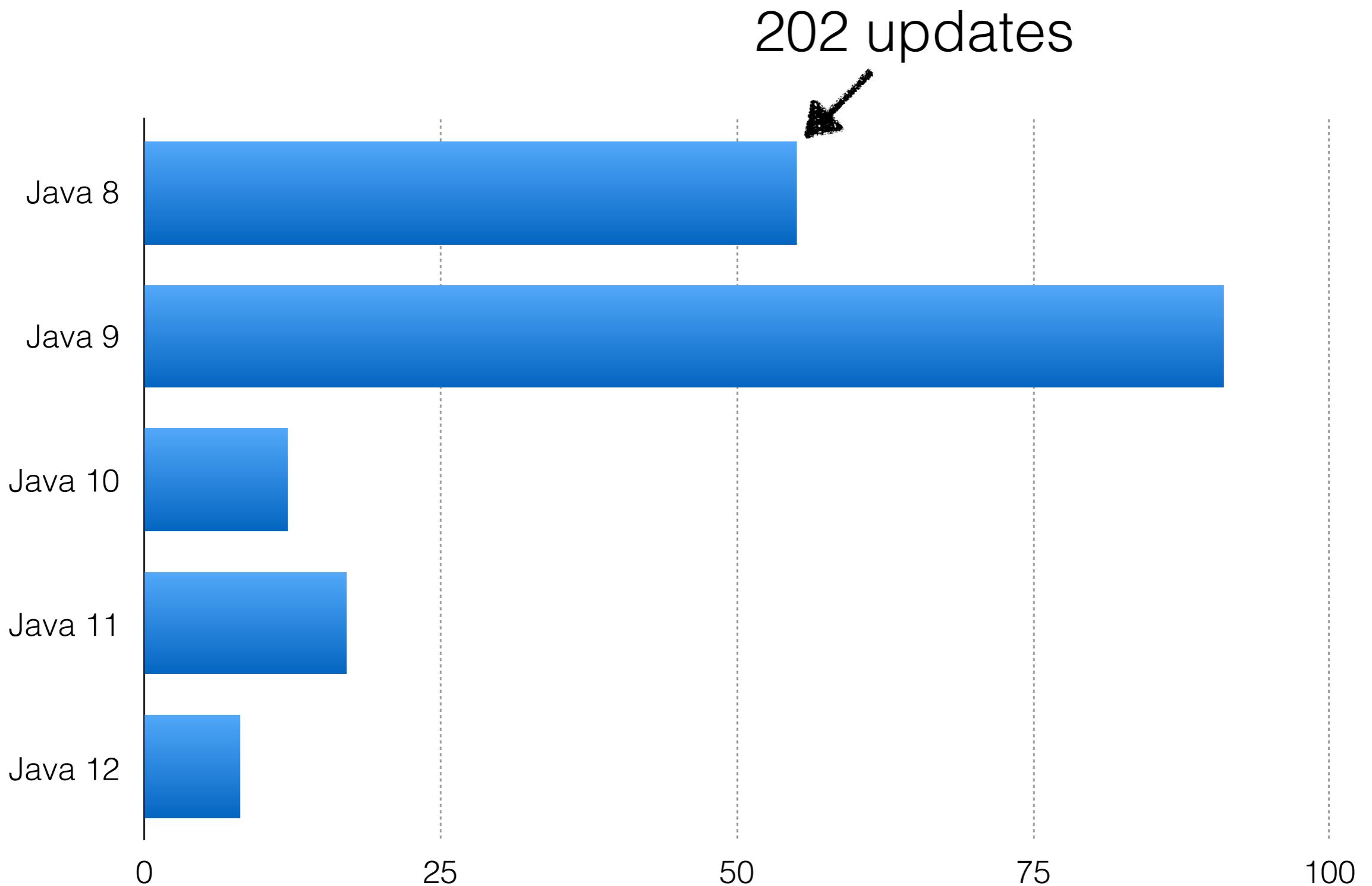
Dependencies



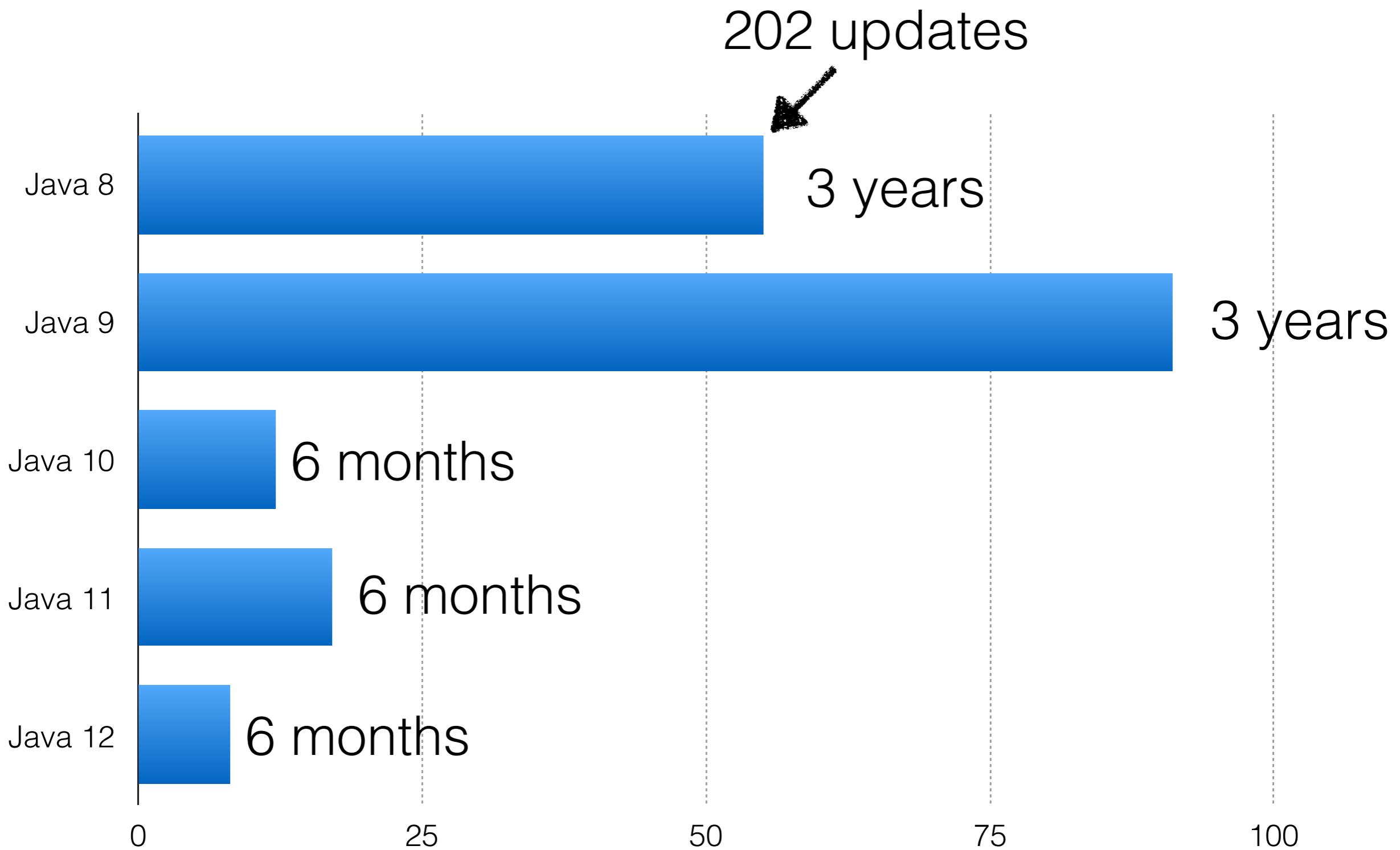
JEPs

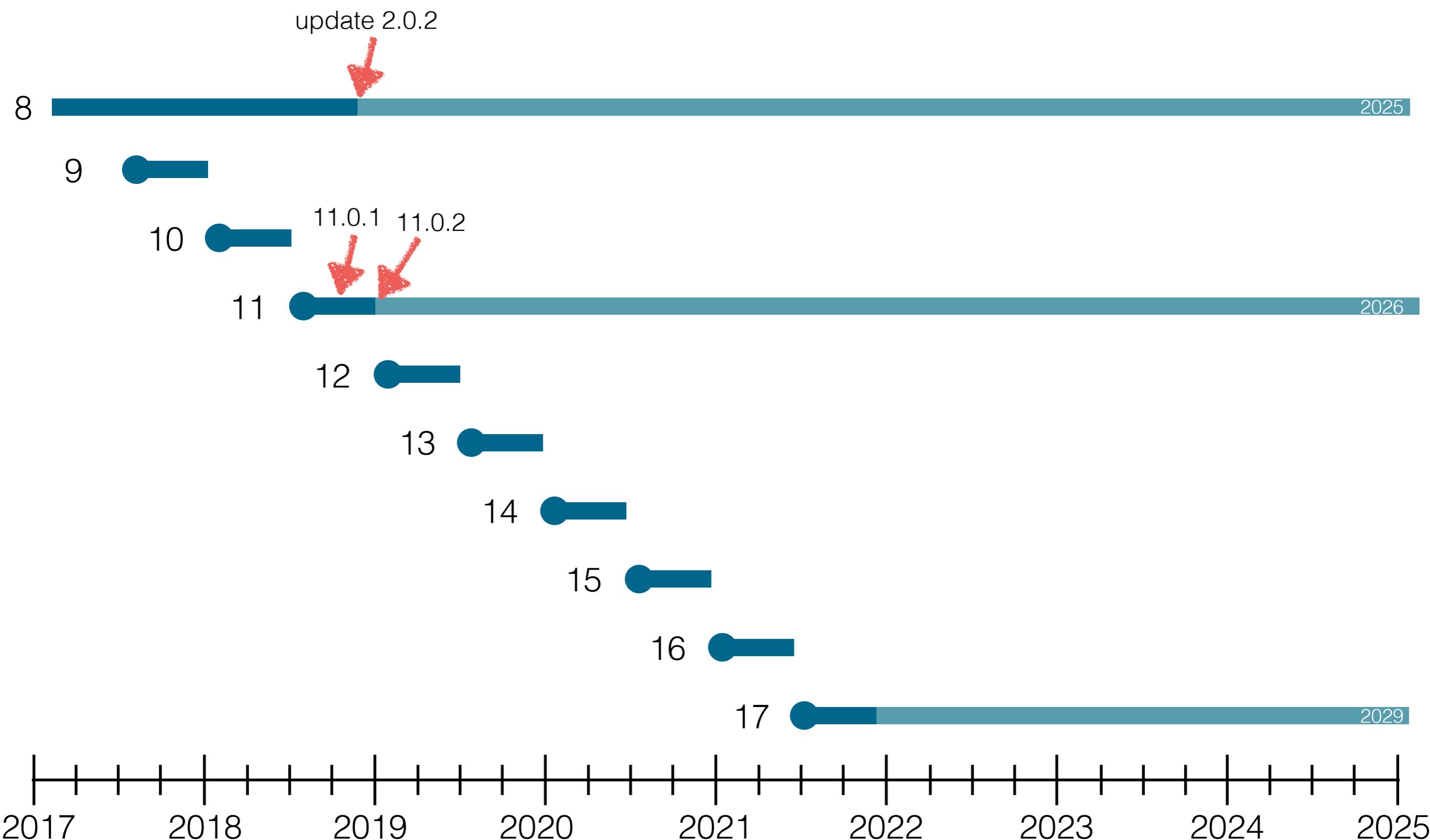


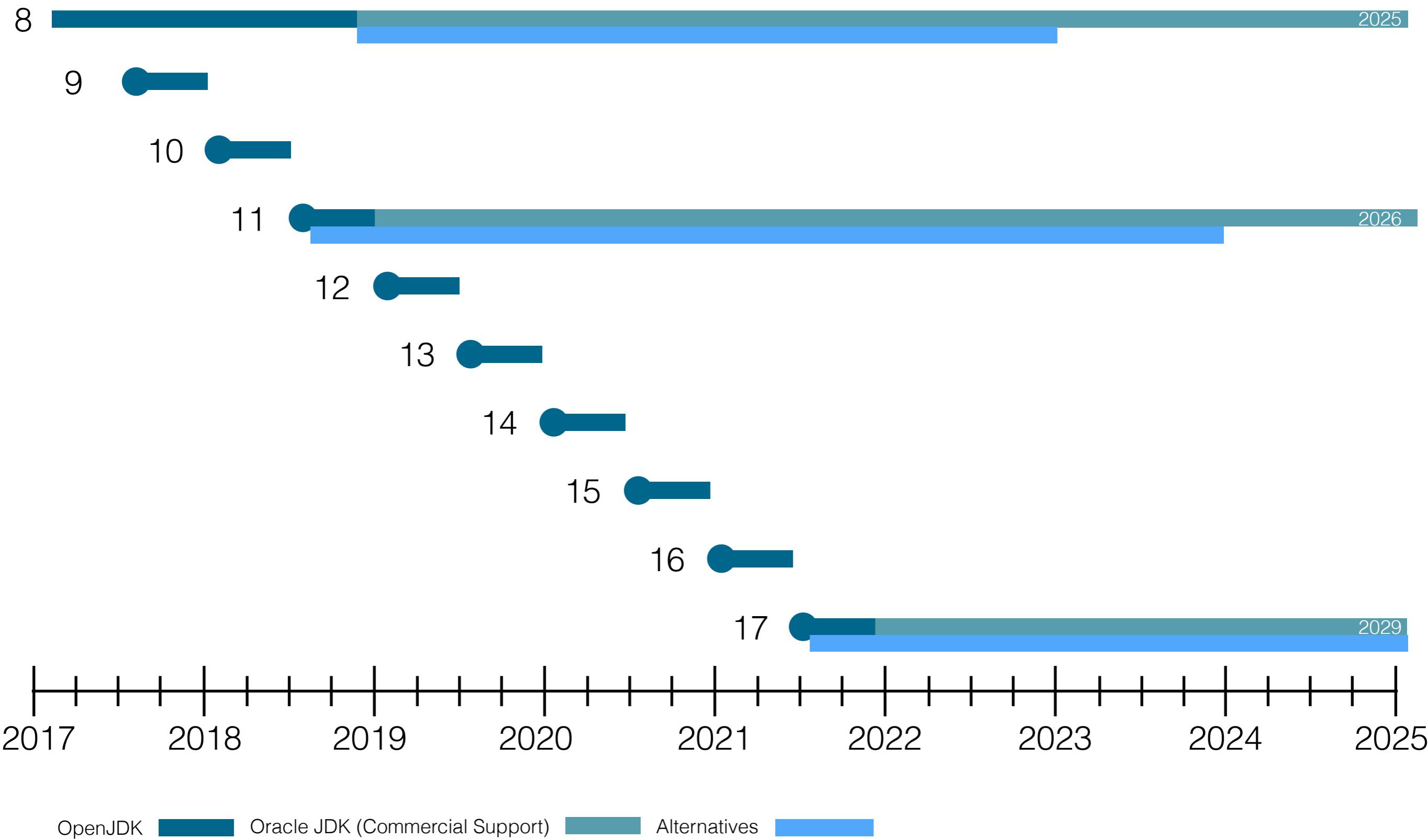
JEPs

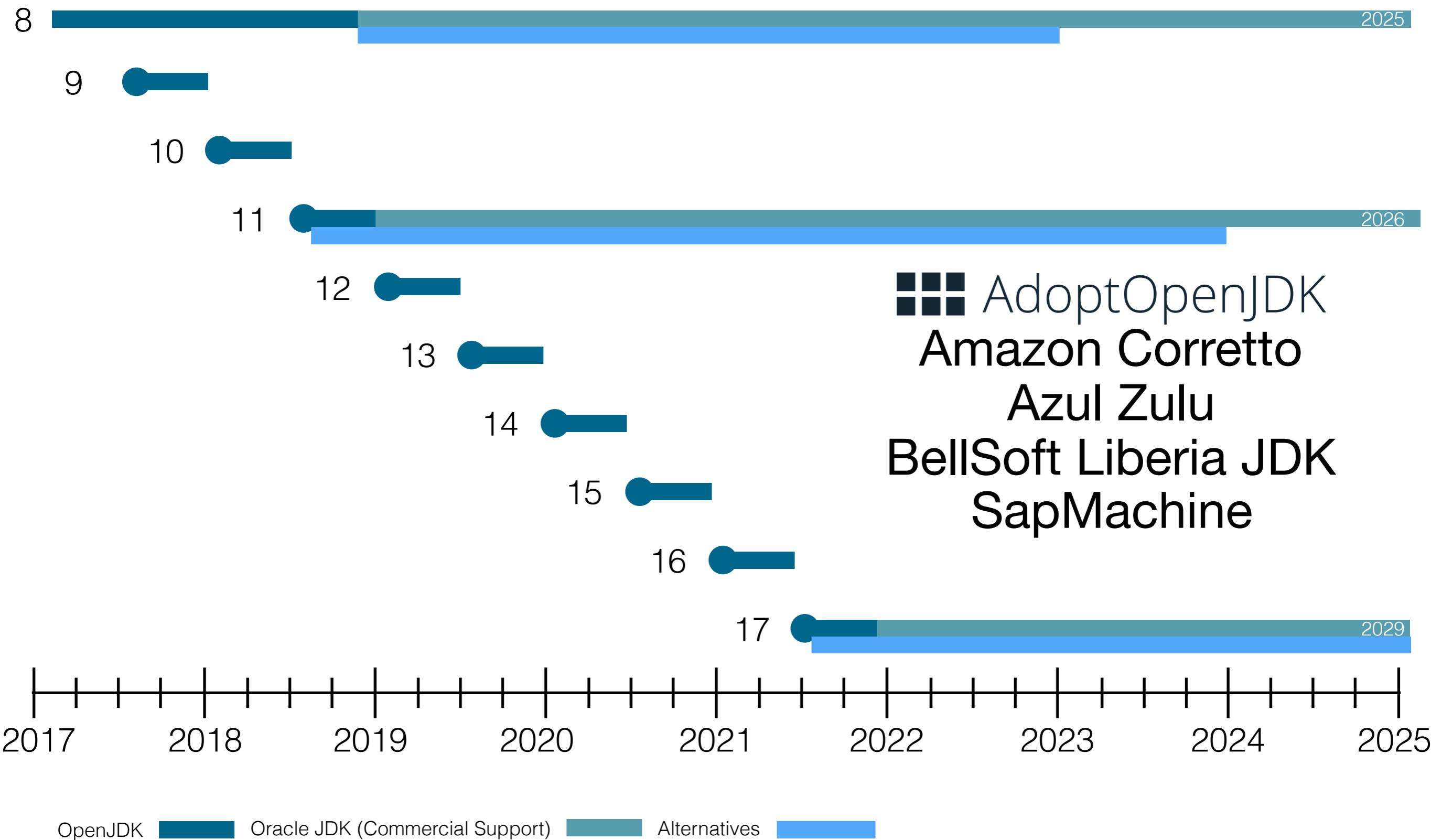


JEPs

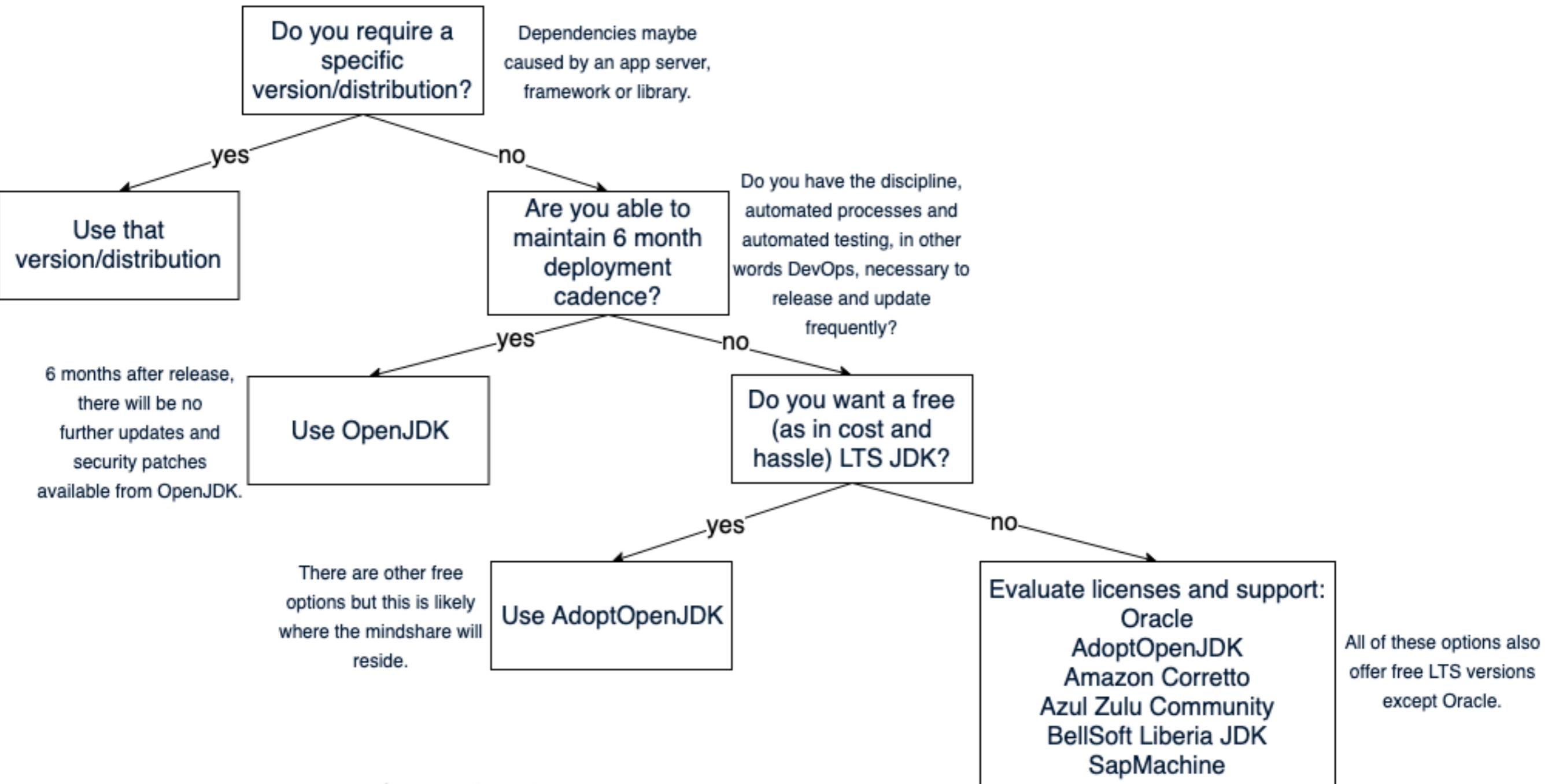








Java Version/Distribution Decision Tree



OpenJDK - <http://jdk.java.net/>

Oracle - <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

AdoptOpenJDK - <https://adoptopenjdk.net/>

Amazon Corretto - <https://aws.amazon.com/corretto/>

Azul Zulu Community - https://www_azul_com/downloads/zulu-community/

BellSoft Liberia JDK - <https://bell-sw.com/>

SapMachine - <https://sap.github.io/SapMachine/>

Java is Still Free 2.0.3



Java Champions

Follow

Mar 3 · 24 min read

This is a repeat of (version 2.0.3) of the [Google Doc](#) put together by the Java Champions community of independent Java leaders and experts.

- Older version 1.0.0—17th of Sept 2018—If you’re curious where we were ~6 months ago.
- Portuguese translation by @leomrlima—TBD
- Spanish translation by @aalmiray et al—TBD

... .

With the changes to Oracle JDK distribution and support, there has been considerable uncertainty over the rights to use Oracle JDK vs Oracle OpenJDK builds vs OpenJDK builds from other providers. There are various ways to get free updates (**including security**), and (new and existing) paid support models available from various vendors to consider. This document has a Shorter Version and a much Longer Version section with all of the detail.



Prebuilt OpenJDK Binaries for Free!

Java™ is the world's leading programming language and platform. AdoptOpenJDK uses infrastructure, build and test scripts to produce prebuilt binaries from OpenJDK™ class libraries and a choice of either the OpenJDK HotSpot or Eclipse OpenJ9 VM. All AdoptOpenJDK binaries and scripts are open source licensed and available for free.

Download for macOS x64

1. Choose a Version

- OpenJDK 8 (LTS)
- OpenJDK 11 (LTS)
- OpenJDK 13 (Latest)

2. Choose a JVM

[Help Me Choose](#)

- HotSpot
- OpenJ9

 [Latest release](#)

jdk-11.0.5+10

[Other platforms](#) [Release Archive & Nightly Builds](#)

AdoptOpenJDK now also distributes OpenJDK upstream builds! (Built by Red Hat)



```
docker pull openjdk:11
docker run -it --rm openjdk:11 bash
docker run -it --rm openjdk:11 jshell
```

```
docker pull adoptopenjdk/openjdk11
docker run -it --rm adoptopenjdk/openjdk11 bash
docker run -it --rm adoptopenjdk/openjdk11 jshell
```

```
docker pull amazoncorretto:11
docker run -it --rm amazoncorretto:11 bash
docker run -it --rm amazoncorretto:11 jshell
```

```
docker pull azul/zulu-openjdk:11
docker run -it --rm azul/zulu-openjdk:11 bash
docker run -it --rm azul/zulu-openjdk:11 jshell
```

JEP 223: NEW VERSION- STRING SCHEME JEP 322: TIME-BASED RELEASE VERSIONING

Release: 9 & 10
Component: core-libs / java.lang

Define a version-string scheme that easily distinguishes major, minor, and security-update releases, and apply it to the JDK.

```
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

```
openjdk version "11.0.3" 2019-04-16
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.3+7)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.3+7, mixed mode)
```

\$FEATURE.\$INTERIM.\$UPDATE.\$PATCH+\$BUILD

[Manage topics](#)

23 commits

1 branch

0 releases

1 contributor

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find File](#)[Clone or download ▾](#)

 cjudd	Fixed mistakes in jep224.	Latest commit d5b3749 5 hours ago
 jep200	Finished jlinking part of jep200 lab.	8 hours ago
 jep222	Added JEP222 JShell	3 days ago
 jep224	Fixed mistakes in jep224.	5 hours ago
 jep286	Added jep286 lab and solutions.	yesterday
 jep330	Added a simple HelloWorld.	9 days ago
 .gitignore	Ignore common intellij and Visual Studio Code files.	2 months ago
 README.md	Added jigsaw to menu.	5 hours ago
 SETUP.md	Added setup instructions.	21 hours ago

[README.md](#)

Java 9, 10, 11 Workshop

Java 9, 10, 11 Workshop for preparing Java developers to migrate from Java 8 (LTS) to Java 11 (LTS)

Labs

1. [Setup](#)
2. [JEP 222: JShell Lab](#)
3. [JEP 330: Launch Single-File Source-Code Programs Lab](#)
4. [JEP 224: HTML5 & JEP 225: JavaDoc Search](#)
5. [JEP 200: Jigsaw](#)
6. [JEP 286: Local-Variable Type Inference & JEP 323: Local-Variable Syntax for Lambda Parameters](#)

LAB

Setup

TOOLS

JEP 222: JSHELL

Release: 9
Component: tools/jshell

Provide an interactive tool to evaluate declarations, statements, and expressions of the Java programming language, together with an API so that other applications can leverage this functionality.

```
$ jshell
| Welcome to JShell -- Version 11.0.3
| For an introduction type: /help intro

jshell>
```

```
jshell> /help
Type a Java language expression, statement, or declaration.
Or type one of the following commands:
/list [<name or id>|-all|-start]
    list the source you have typed
/edit <name or id>
    edit a source entry
/drop <name or id>
    delete a source entry
/save [-all|-history|-start] <file>
    Save snippet source to a file
/open <file>
    open a file as source input
/vars [<name or id>|-all|-start]
    list the declared variables and their values
/methods [<name or id>|-all|-start]
    list the declared methods and their signatures
/types [<name or id>|-all|-start]
    list the type declarations
/imports
    list the imported items
/exit [<integer-expression-snippet>]
    exit the jshell tool
/env [-class-path <path>] [-module-path <path>] [-add-modules <modules>] ...
    view or change the evaluation context
/reset [-class-path <path>] [-module-path <path>] [-add-modules <modules>]...
    reset the jshell tool
/reload [-restore] [-quiet] [-class-path <path>] [-module-path <path>]...
    reset and replay relevant history -- current or previous (-restore)
/history [-all]
    history of what you have typed
/help [<command>|<subject>]
    get information about using the jshell tool
/set editor|start|feedback|mode|prompt|truncation|format ...
    set configuration information
/? [<command>|<subject>]
    get information about using the jshell tool
/!
    rerun last snippet -- see /help rerun
/<id>
    rerun snippets by ID or ID range -- see /help rerun
/-<n>
    rerun n-th previous snippet -- see /help rerun

jshell>
```

```
jshell> var i = 1;  
i ==> 1
```

```
jshell> i + 3  
$2 ==> 4
```

```
jshell> /list
```

```
1 : var i = 1;  
2 : i + 3
```

```
jshell> var i = 1;  
i ==> 1  
  
jshell> i + 3  
$2 ==> 4  
  
jshell> /list  
  
 1 : var i = 1;  
 2 : i + 3  
  
jshell> /edit
```

The screenshot shows a Mac OS X window titled "JShell Edit Pad". The window has three panes:

- Top pane (Code):** Contains the JavaScript code:

```
var i = 1;  
i + 3;
```
- Middle pane (History):** Shows the command history:

```
jshell> var  
i ==> 1  
  
jshell> i +  
$2 ==> 4  
  
jshell> /lis  
1 : var i _,  
2 : i + 3  
  
jshell> /edit
```
- Bottom pane (Buttons):** Contains three buttons: "Cancel", "Accept", and "Exit".

auto complete tab key

```
jshell> System.  
Logger  
console()  
getLogger()  
identityHashCode()  
loadLibrary()  
setErr()  
setSecurityManager()  
  
jshell> System.out.  
append()          checkError()    close()  
notify()         notifyAll()     print()  
  
jshell> System.out.print  
print()      printf()      println()  
  
jshell> System.out.print
```

```
jshell> public class Person {  
...>     private int age;  
...>  
...>     public Person setAge(int age) {  
...>         this.age = age;  
...>         return this;  
...>     }  
...>  
...>     public int getAge() {  
...>         return age;  
...>     }  
...>  
...> }  
| created class Person  
  
jshell> /type  
|     class Person  
  
jshell> var chris = new Person().setAge(47)  
chris ==> Person@735b5592  
  
jshell> chris.getAge()  
$1 ==> 47  
  
jshell> /var  
|     Person chris = Person@735b5592  
|     int $1 = 47
```

```
jshell> /exit  
| Goodbye
```

LAB

JEP 222: JShell

JEP 330: LAUNCH SINGLE-FILE SOURCE- CODE PROGRAMS

Release: 11
Component: tools/javac

Enhance the java launcher to run a program supplied as a single file of Java source code, including usage from within a script by means of "shebang" files and related techniques.

HelloWorld.java

```
public class HelloWorld {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World");  
    }  
}
```

```
$ javac HelloWorld.java  
$ java HelloWorld  
Hello World
```

HelloWorld.java

```
public class HelloWorld {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World");  
    }  
}
```

```
$ javac HelloWorld.java  
$ java HelloWorld  
Hello World
```

HelloWorld.java

```
public class HelloWorld {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World");  
    }  
}
```

```
$ javac HelloWorld.java  
$ java HelloWorld  
Hello World
```

```
$ java HelloWorld.java  
Hello World
```

no .java extension



HelloWorld

```
#!/usr/bin/java --source 11 ← shebang
```

```
public class HelloWorld {  
  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World");  
    }  
  
}
```

```
$ chmod 744 HelloWorld  
$ ./HelloWorld  
Hello World
```

LAB

JEP 330: Launch Single-File Source Code Programs

JEP 224: HTML5

JAVADOC

Release: 9
Component: tools/javadoc

Enhance the javadoc tool to generate HTML5 markup.

JEP 225: JAVADOC SEARCH

Release: 9

Component: tools/javadoc

Add a search box to API documentation generated by the standard doclet that can be used to search for program elements and tagged words and phrases within the documentation. The search box appears in the header of all pages generated by the standard doclet.

[All Classes](#) [All Profiles](#)**Packages**

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im
java.awt.im.spi
java.awt.image
java.awt.image.renderable
java.awt.print

All Classes

AbstractAction
AbstractAnnotationValueVisitor
AbstractAnnotationValueVisitor
AbstractAnnotationValueVisitor
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractChronology
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeCol
AbstractDocument.Content
All Classes | All Profiles | Index | Help

[OVERVIEW](#)[PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)Java™ Platform
Standard Ed. 8[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)

Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: [Description](#)

Profiles

- compact1
- compact2
- compact3

Packages

Package

[java.applet](#)

Description

Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.

[java.awt](#)

Contains all of the classes for creating user interfaces and for painting graphics and images.

[java.awt.color](#)

Provides classes for color spaces.

Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

All Modules Java SE JDK Other Modules

Module	Description
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
java.datatransfer	Defines the API for transferring data between and within applications.
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.
java.instrument	Defines services that allow agents to instrument programs running on the JVM.
java.logging	Defines the Java Logging API.

ALL CLASSES

SEARCH: X

Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose modules whose names start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available on the Java Platform. These APIs are in modules whose names start with `jdk`.

All Modules Java SE JDK Other Modules

Module	Description
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, a
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.
<code>java.logging</code>	Defines the Java Logging API.

Types

`java.util.Base64`

`java.util.Base64.Decoder`

`java.util.Base64.Encoder`

Members

`javax.xml.crypto.dsig.Transform.BASE64`

`java.util.Base64.Decoder.decode(byte[], byte[])`

`java.util.Base64.Decoder.decode(byte[])`

`java.util.Base64.Decoder.decode(ByteBuffer)`

`java.util.Base64.Decoder.decode(String)`

`java.util.Base64.Encoder.encode(byte[], byte[])`

`java.util.Base64.Encoder.encode(byte[])`

`java.util.Base64.Encoder.encode(ByteBuffer)`

`java.util.Base64.Encoder.encodeToString(byte[])`

`java.util.Base64.getDecoder()`

`java.util.Base64.getEncoder()`

`java.util.Base64.getMimeDecoder()`

`java.util.Base64.getMimeEncoder()`

`java.util.Base64.getMimeEncoder(int, byte[])`

`java.util.Base64.getUrlDecoder()`

`java.util.Base64.getUrlEncoder()`

`java.util.Base64.Encoder.withoutPadding()`

`java.util.Base64.Decoder.wrap(InputStream)`

`java.util.Base64.Encoder.wrap(OutputStream)`

```
/**  
 * Business {@index domain} class representing a person in the application.  
 */  
public class Person {  
  
}
```

LAB

JEP 224: HTML5

JEP 225: JavaDoc Search

JAVA PLATFORM MODULE SYSTEM

Java Platform Modular System

JPMS

JSR 376

Project Jigsaw

- JEP 200: The Modular JDK
- JEP 201: Modular Source Code
- JEP 220: Modular Run-Time Images
- JEP 261: Module System
- JEP 260: Encapsulate Most Internal APIs
- JEP 282: jlink: The Java Linker

JDK



rt.jar

- monolithic jar
- entangled classes
- hard to evolve
- constrained by backwards compatibility
- entire JDK has to be installed

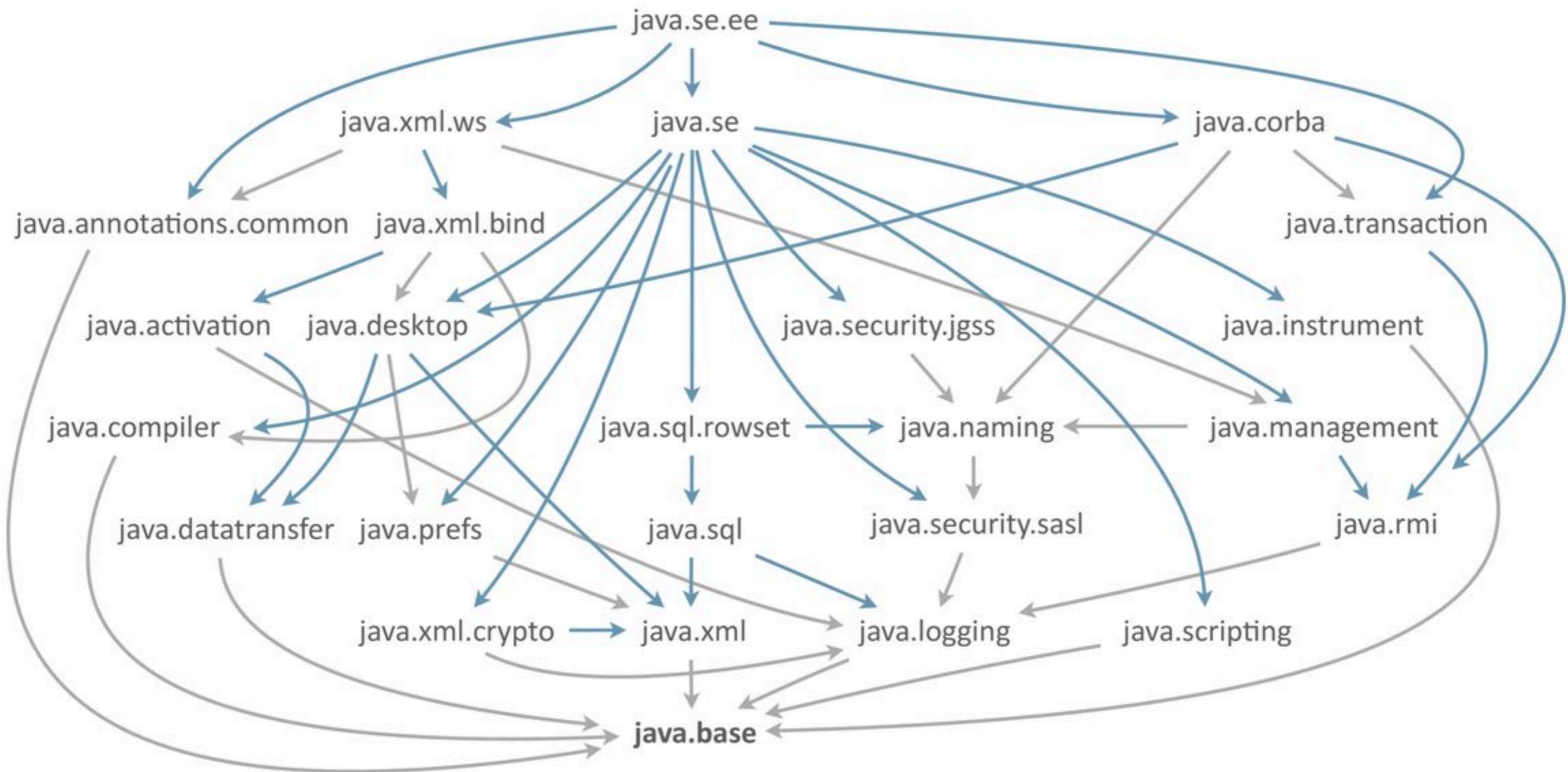
JAR Hell

- multiple versions of same jar in classpath
- all classes had access to all other classes
 - including internal classes like sun.*
- ClassNotFoundException failures at runtime

Design Principles

- loose coupling between components
- clear contracts and dependencies between components
- hidden implementation using strong encapsulation

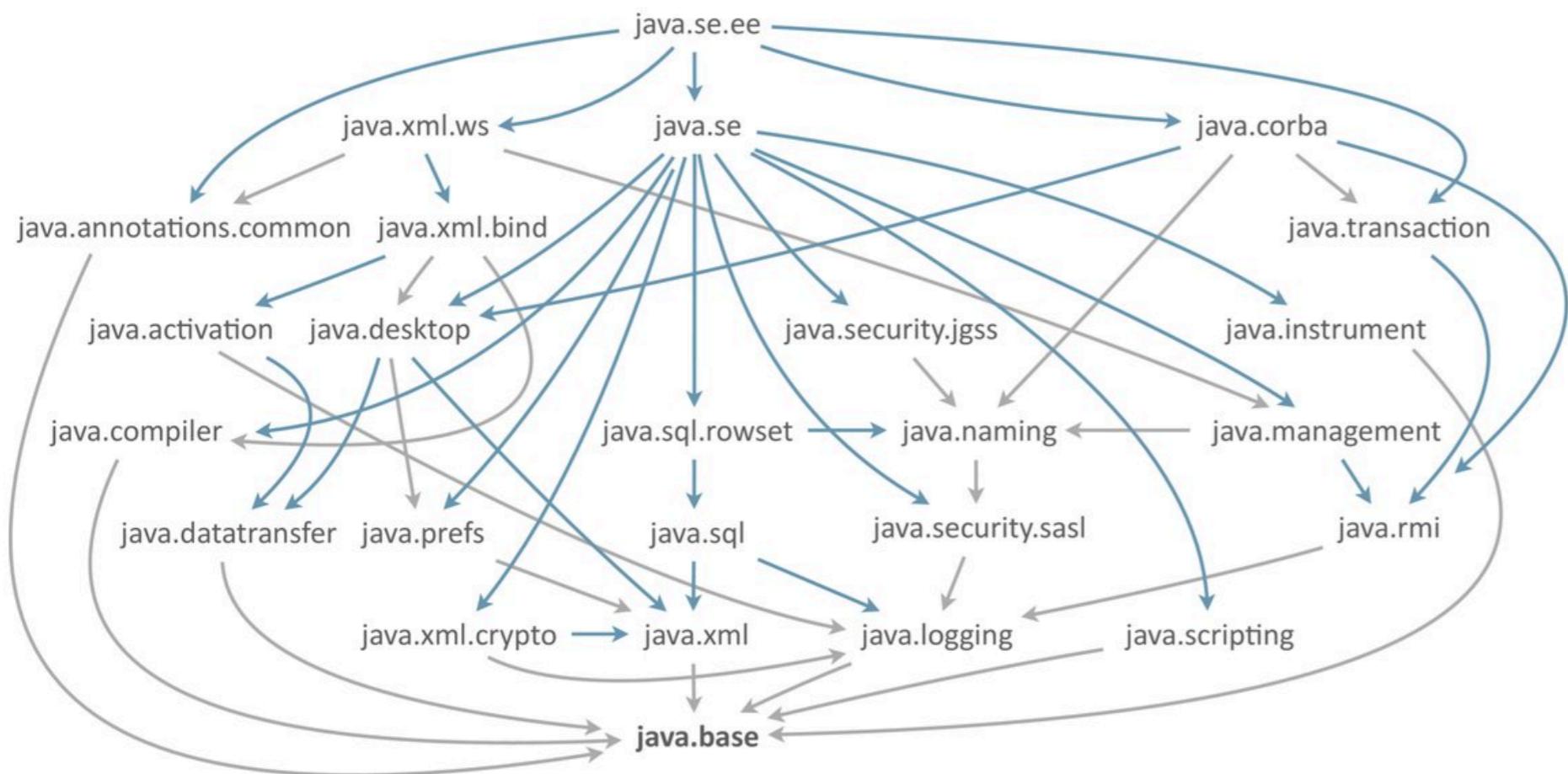
Explicit Dependencies



JDK 9: 75 modules

JDK 11: 70 modules

```
java --list-modules
```



java.base@11.0.3
java.compiler@11.0.3
java.datatransfer@11.0.3
java.desktop@11.0.3
java.instrument@11.0.3
java.logging@11.0.3
java.management@11.0.3
java.management.rmi@11.0.3
java.naming@11.0.3
java.prefs@11.0.3
java.rmi@11.0.3
java.scripting@11.0.3
java.se@11.0.3
java.security.jgss@11.0.3
java.security.sasl@11.0.3
java.smartcardio@11.0.3
java.sql@11.0.3
java.sql.rowset@11.0.3
java.transaction.xa@11.0.3
java.xml@11.0.3
java.xml.crypto@11.0.3
jdk.accessibility@11.0.3
jdk.aot@11.0.3
jdk.attach@11.0.3
jdk.charsets@11.0.3
jdk.compiler@11.0.3
jdk.crypto.cryptoki@11.0.3
jdk.crypto.ec@11.0.3
jdk.dynalink@11.0.3
jdk.editpad@11.0.3
jdk.hotspot.agent@11.0.3
jdk.httpserver@11.0.3
jdk.internal.ed@11.0.3
jdk.internal.jvmstat@11.0.3
jdk.internal.le@11.0.3
jdk.internal.opt@11.0.3
jdk.internal.vm.ci@11.0.3
jdk.internal.vm.compiler@11.0.3
jdk.internal.vm.compiler.management@11.0.3
jdk.jartool@11.0.3
jdk.javadoc@11.0.3
jdk.jcmd@11.0.3
jdk.jconsole@11.0.3
jdk.jdeps@11.0.3
jdk.jdi@11.0.3
jdk.jdwp.agent@11.0.3
jdk.jfr@11.0.3
jdk.jlink@11.0.3
jdk.jshell@11.0.3
jdk.jsobject@11.0.3
jdk.jstard@11.0.3
jdk.localedata@11.0.3
jdk.management@11.0.3
jdk.management.agent@11.0.3
jdk.management.jfr@11.0.3
jdk.naming.dns@11.0.3
jdk.naming.rmi@11.0.3
jdk.net@11.0.3
jdk.pack@11.0.3
jdk.rmic@11.0.3
jdk.scripting.nashorn@11.0.3
jdk.scripting.nashorn.shell@11.0.3
jdk.sctp@11.0.3
jdk.security.auth@11.0.3
jdk.security.jgss@11.0.3
jdk.unsupported@11.0.3
jdk.unsupported.desktop@11.0.3
jdk.xml.dom@11.0.3
jdk.zipfs@11.0.3

JDK 9: 75 modules

JDK 11: 70 modules

Encapsulation



Module Descriptors

module-info.java

```
module java.base {  
  
    exports java.lang;  
    exports java.util;  
    exports java.io;  
  
    // many more  
}
```



Module Descriptors

module-info.java

```
module java.base {
```

```
    exports java.lang;  
    exports java.util;  
    exports java.io;
```

```
    // many more
```

```
}
```

java.base

java.lang
java.util
java.io

...

sun.util
jdk.util
jdk.internal

...

```

module java.management.rmi {

    requires java.naming;

    requires transitive java.management;
    requires transitive java.rmi;

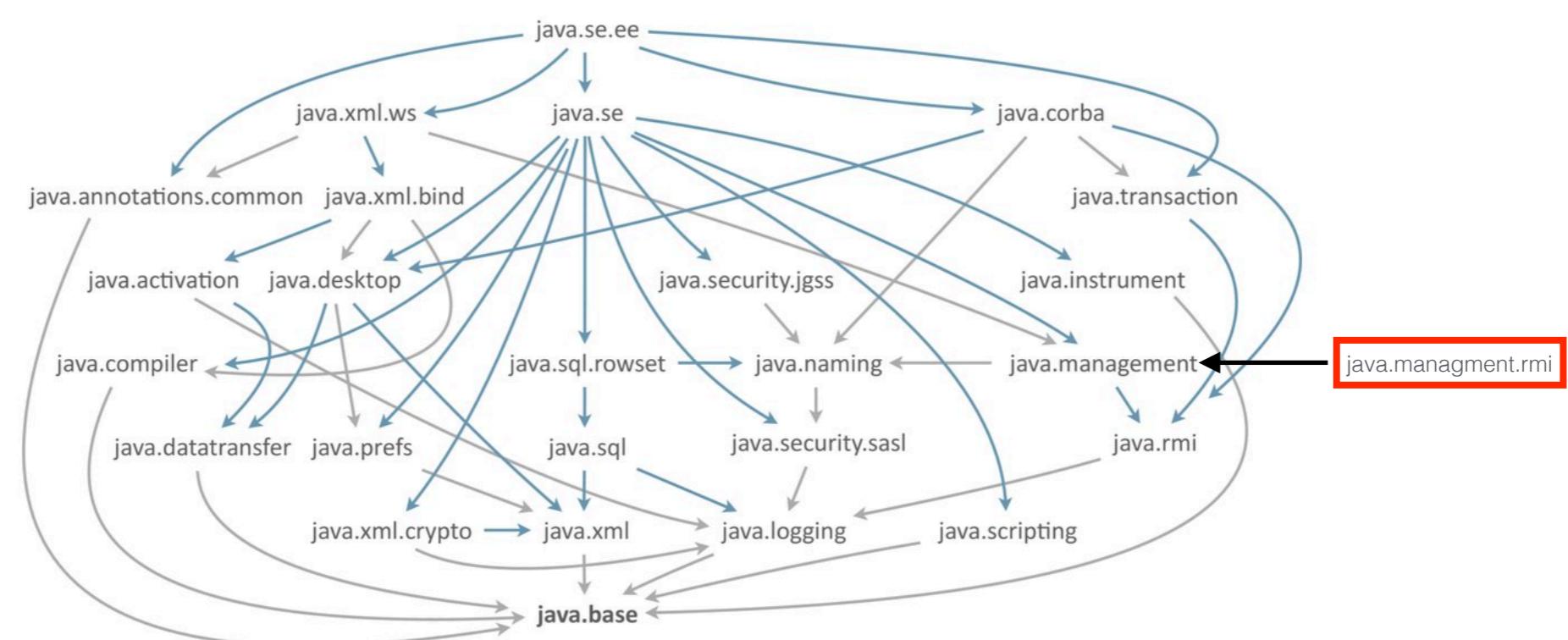
    exports javax.management.remote.rmi;

    exports com.sun.jmx.remote.protocol.rmi to java.management;

    exports com.sun.jmx.remote.internal.rmi to jdk.management.agent;

    // The java.management.rmi module provides implementations
    // of the JMXConnectorProvider and JMXConnectorServerProvider
    // services supporting the RMI protocol.
    provides javax.management.remote.JMXConnectorProvider with
        com.sun.jmx.remote.protocol.rmi.ClientProvider;
    provides javax.management.remote.JMXConnectorServiceProvider with
        com.sun.jmx.remote.protocol.rmi.ServerProvider;
}

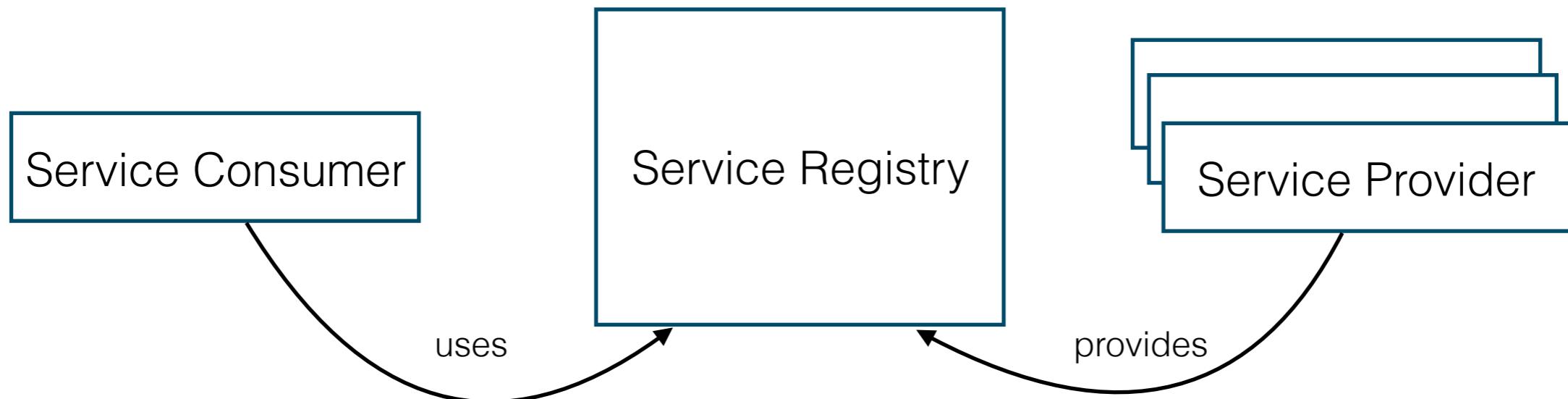
```



```

module java.management.rmi {
    requires java.naming;
    requires transitive java.management;
    requires transitive java.rmi;
    exports javax.management.remote.rmi;
    exports com.sun.jmx.remote.protocol.rmi to java.management;
    exports com.sun.jmx.remote.internal.rmi to jdk.management.agent;
    // The java.management.rmi module provides implementations
    // of the JMXConnectorProvider and JMXConnectorServerProvider
    // services supporting the RMI protocol.
    provides javax.management.remote.JMXConnectorProvider with
        com.sun.jmx.remote.protocol.rmi.ClientProvider;
    provides javax.management.remote.JMXConnectorServiceProvider with
        com.sun.jmx.remote.protocol.rmi.ServerProvider;
}

```



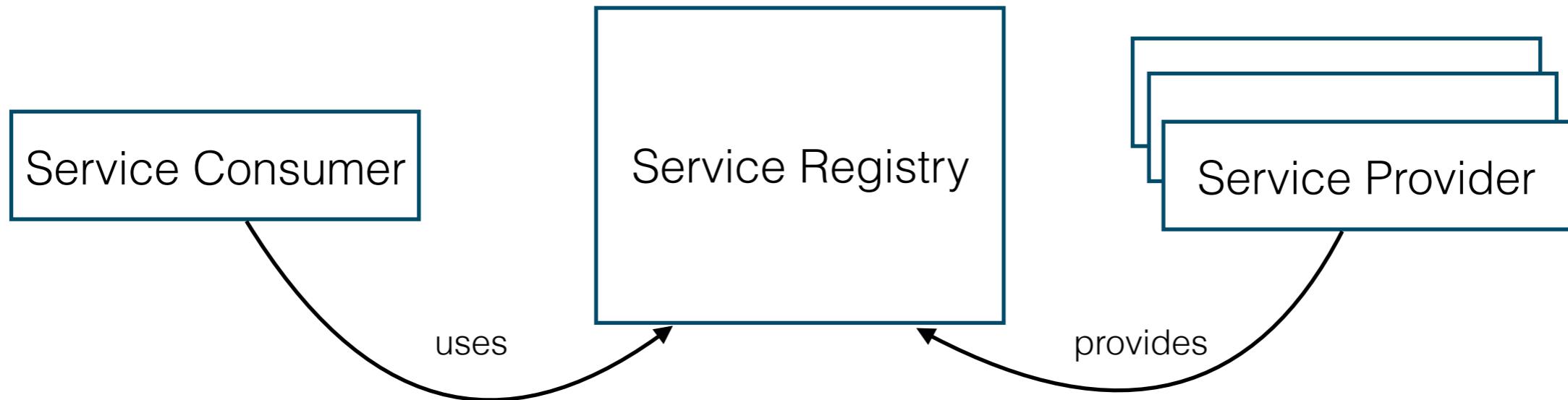
`javax.management.remote.JMXConnectorProvider`

`com.sun.jmx.remote.protocol.rmi.ClientProvider`

```

module java.management.rmi {
    requires java.naming;
    requires transitive java.management;
    requires transitive java.rmi;
    exports javax.management.remote.rmi;
    exports com.sun.jmx.remote.protocol.rmi to java.management;
    exports com.sun.jmx.remote.internal.rmi to jdk.management.agent;
    // The java.management.rmi module provides implementations
    // of the JMXConnectorProvider and JMXConnectorServerProvider
    // services supporting the RMI protocol.
    provides javax.management.remote.JMXConnectorProvider with
        com.sun.jmx.remote.protocol.rmi.ClientProvider;
    provides javax.management.remote.JMXConnectorServiceProvider with
        com.sun.jmx.remote.protocol.rmi.ServerProvider;
}

```



`javax.management.remote.JMXConnectorProvider` `com.sun.jmx.remote.protocol.rmi.ClientProvider`

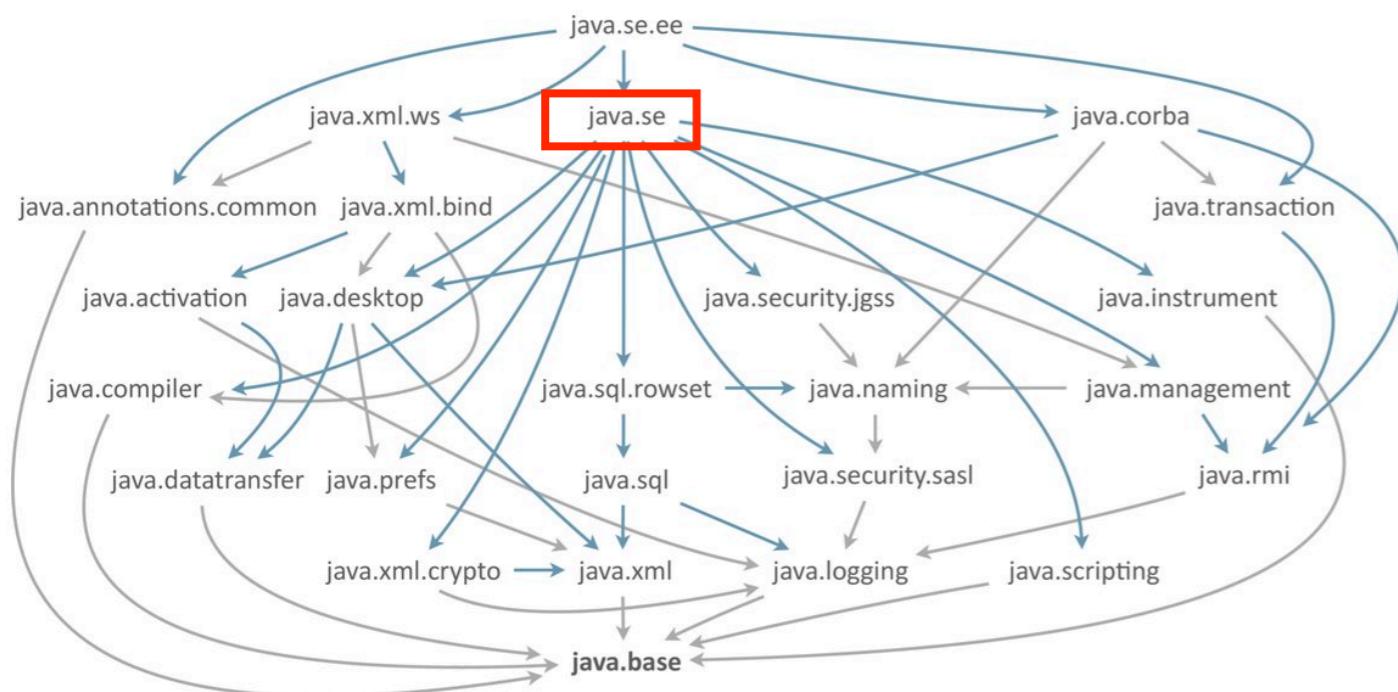
```

ServiceLoader<JMXConnectorProvider> loader = ServiceLoader.load(JMXConnectorProvider.class);
for (var provider : loader) {
    System.out.println(provider.getClass().getName());
}

```

`com.sun.jmx.remote.protocol.rmi.ClientProvider`

module-info.java



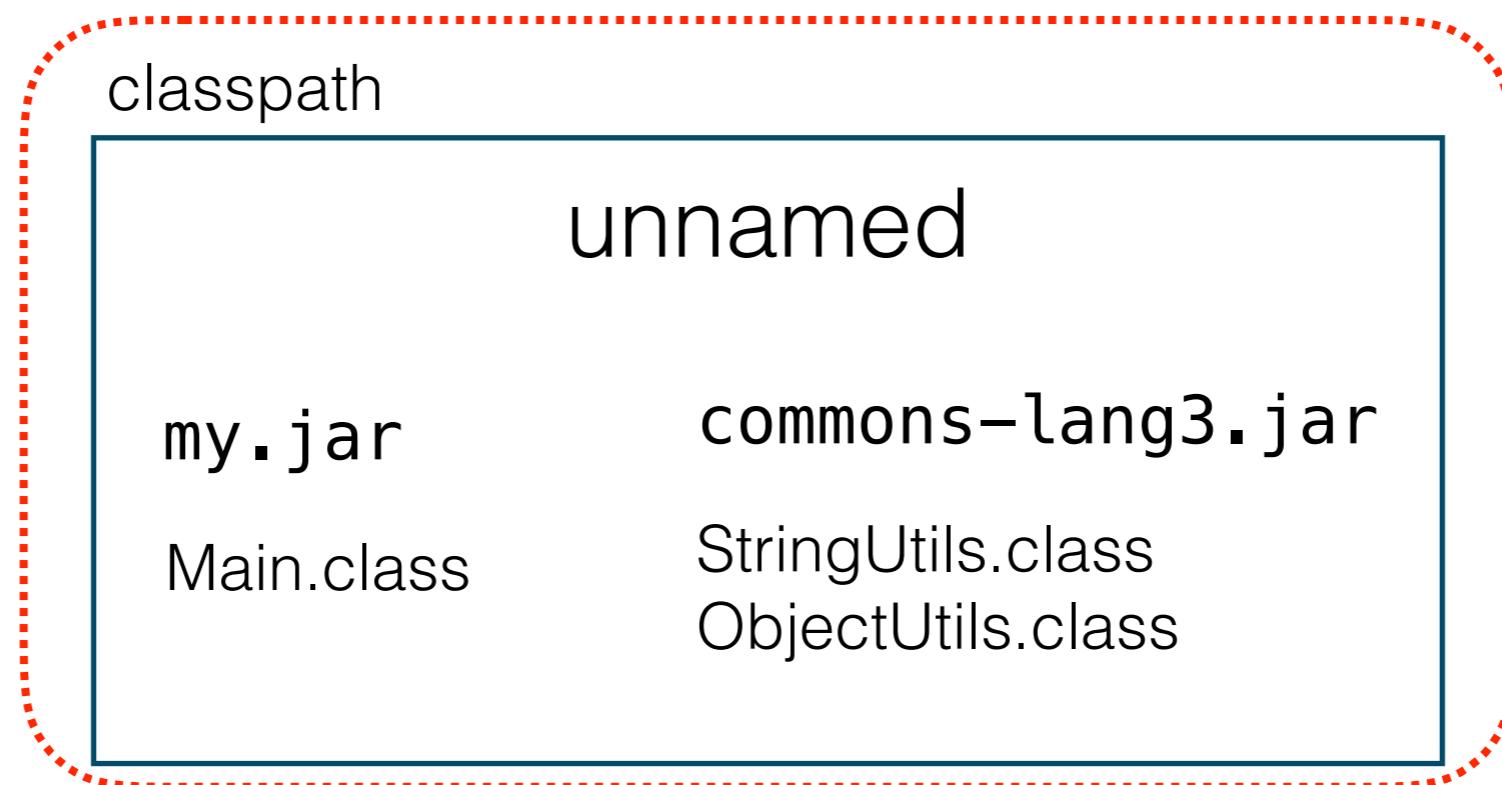
```
module java.se {  
    requires transitive java.compiler;  
    requires transitive java.datatransfer;  
    requires transitive java.desktop;  
    requires transitive java.instrument;  
    requires transitive java.logging;  
    requires transitive java.management;  
    requires transitive java.management.rmi;  
    requires transitive java.naming;  
    requires transitive java.net.http;  
    requires transitive java.prefs;  
    requires transitive java.rmi;  
    requires transitive java.scripting;  
    requires transitive java.security.jgss;  
    requires transitive java.security.sasl;  
    requires transitive java.sql;  
    requires transitive java.sql.rowset;  
    requires transitive java.transaction.xa;  
    requires transitive java.xml;  
    requires transitive java.xml.crypto;  
}
```

Do you have to change everything to modules?

NO

```
javac -cp $CLASSPATH ...  
java -cp $CLASSPATH ...
```

```
javac -cp $CLASSPATH ...
java -cp $CLASSPATH ...
```



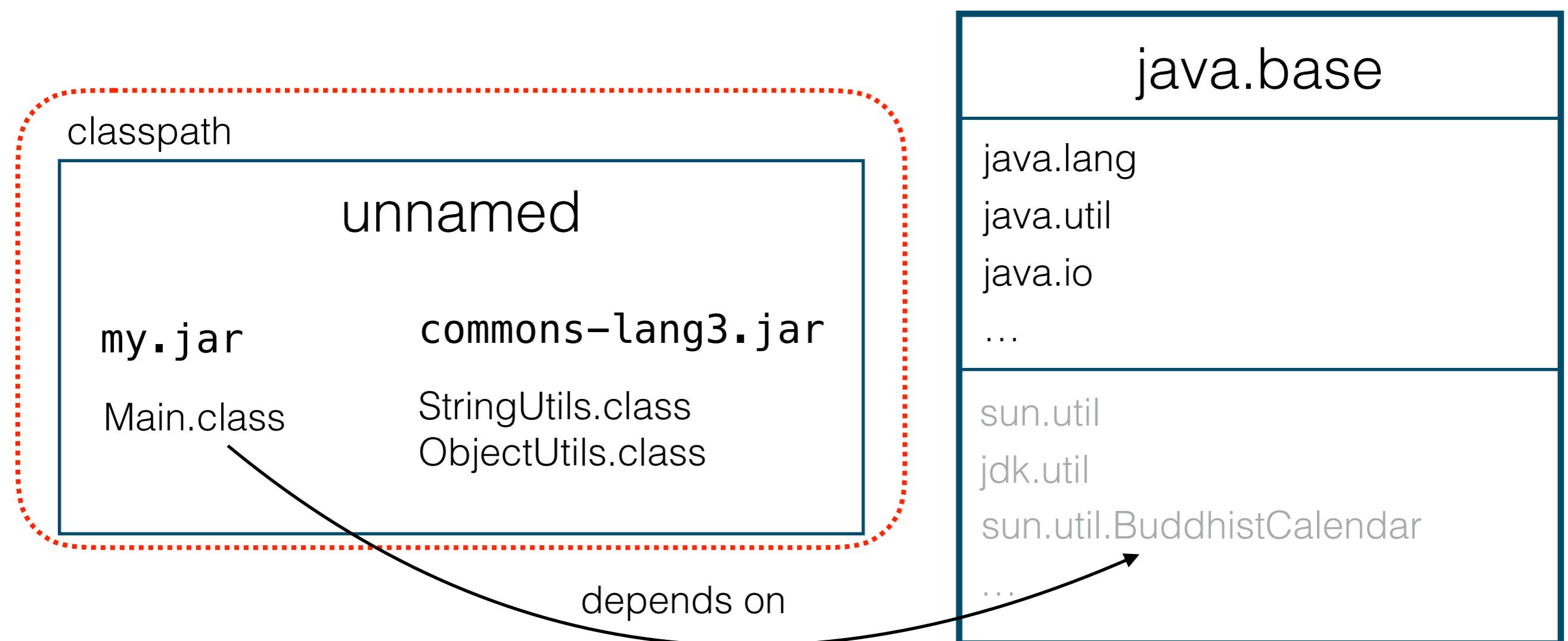
```
import sun.util.BuddhistCalendar;  
  
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        System.out.println(BuddhistCalendar.class.getName());  
    }  
}
```

```
javac -d target/classes src/main/java/net/javajudd/jep286/Main.java
```

```
src/main/java/net/javajudd/jep286/Main.java:3: error: package sun.util is not visible  
import sun.util.BuddhistCalendar;
```

^

```
(package sun.util is declared in module java.base, which does not export it)  
1 error
```



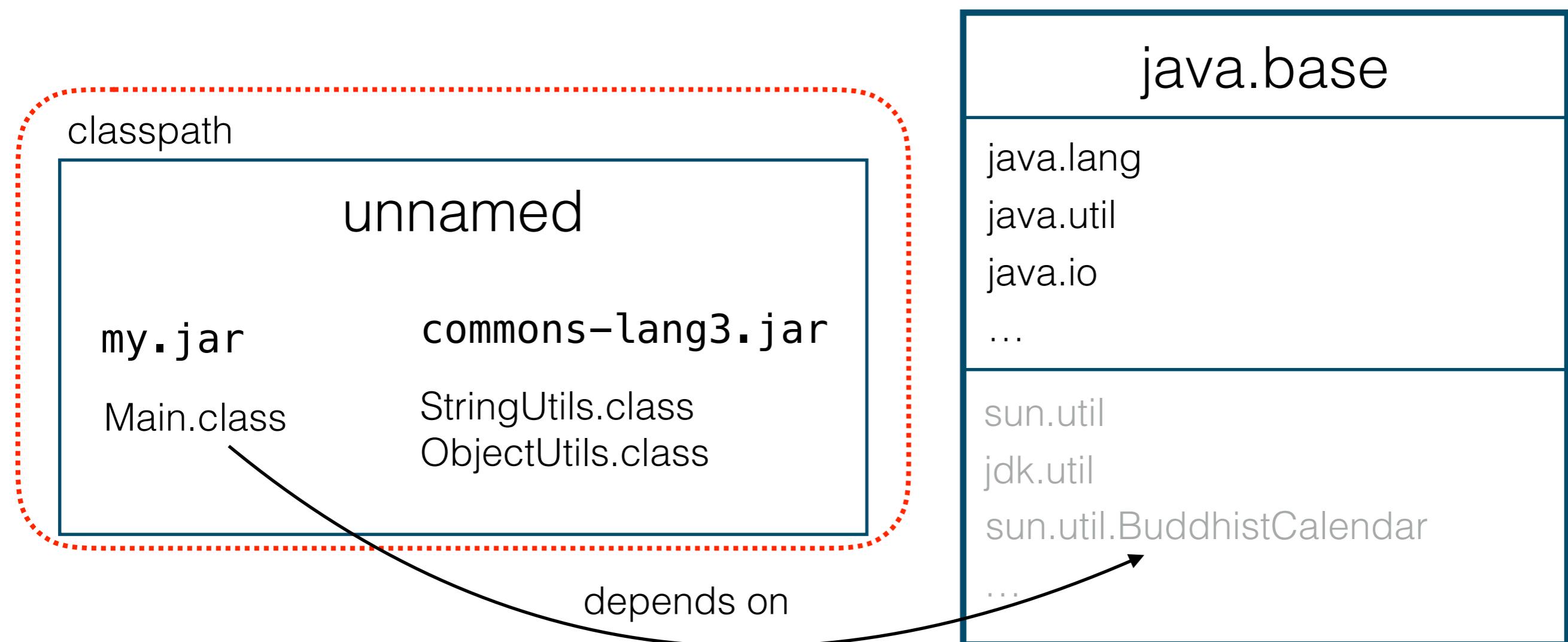
```
import sun.util.BuddhistCalendar;  
  
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        System.out.println(BuddhistCalendar.class.getName());  
    }  
}
```

jdeps target/classes

classes → java.base
classes → not found
net.javajudd.jep286

→ sun.util

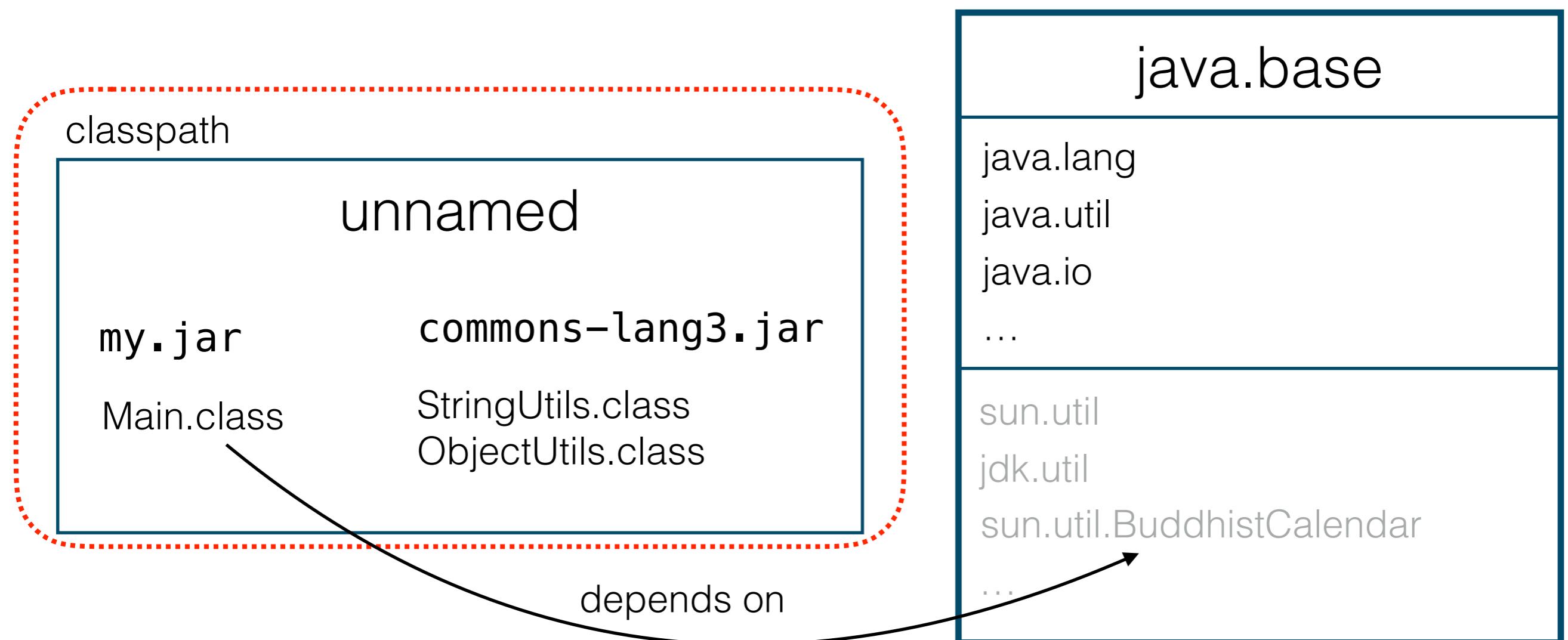
JDK internal API (java.base)

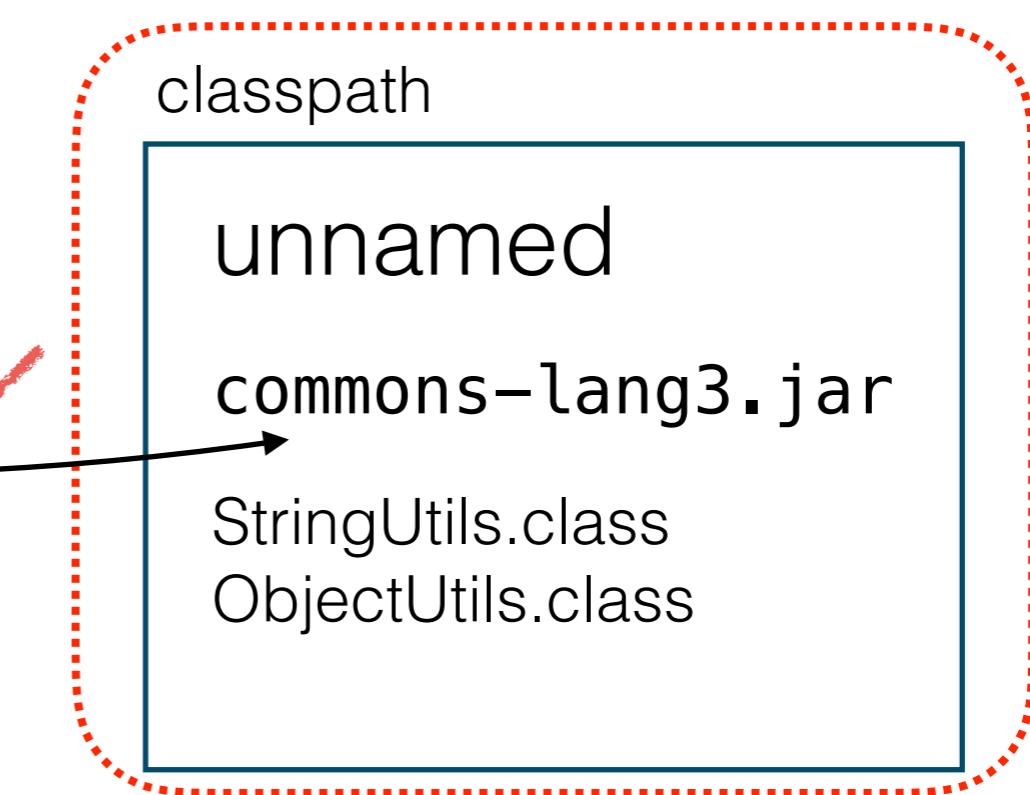
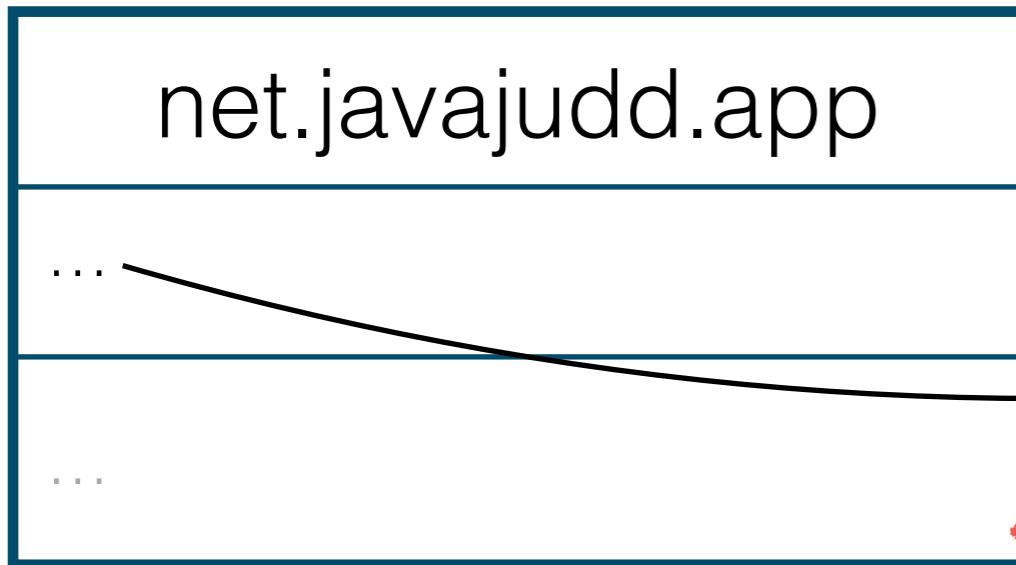


```
import sun.util.BuddhistCalendar;  
  
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        System.out.println(BuddhistCalendar.class.getName());  
    }  
}
```

javac --add-exports java.base/sun.util=ALL-UNNAMED -d target/classes src/main/java/net/javajudd/jcp286/Main.java

java --add-exports java.base/sun.util=ALL-UNNAMED -cp target/classes net.javajudd.jcp286.Main





Automatic Module

```
javac -p lib/<name>.jar
```

```
java -p lib:out -m <module>/<class>
```

net.javajudd.app

...

...

<jar name or Automatic-Module-Name>

...

...

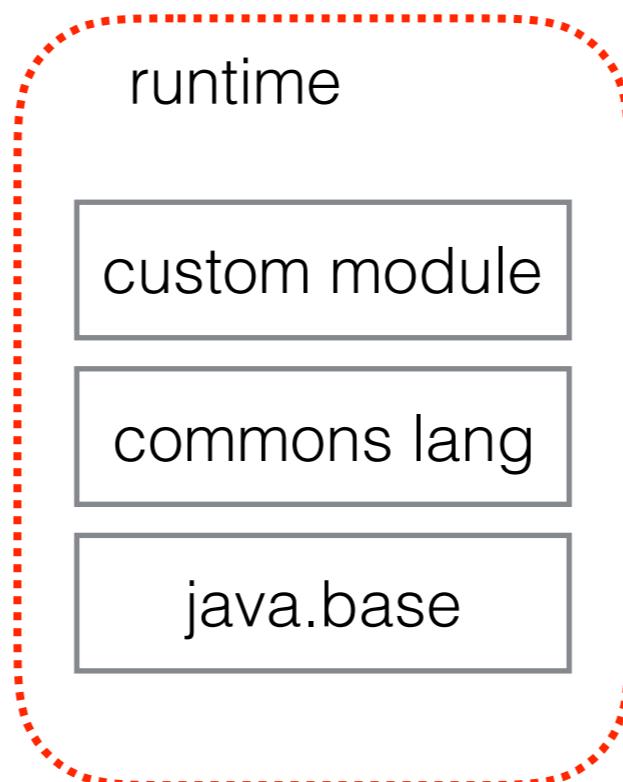
Automatic Module

MANIFEST.MF

```
Manifest-Version: 1.0
Created-By: Apache Maven Bundle Plugin
Built-By: chtompki
Build-Jdk: 11.0.2
Specification-Title: Apache Commons Lang
Specification-Version: 3.9
Specification-Vendor: The Apache Software Foundation
Implementation-Title: Apache Commons Lang
Implementation-Version: 3.9
Implementation-Vendor-Id: org.apache.commons
Implementation-Vendor: The Apache Software Foundation
Implementation-URL: http://commons.apache.org/proper/commons-lang/
Automatic-Module-Name: org.apache.commons.lang3
Bnd-LastModified: 1554946229157
Bundle-Description: Apache Commons Lang, a package of Java utility classes for the
classes that are in java.lang's hierarchy, or are considered to be so standard as
to justify existence in java.lang.
Bundle-DocURL: http://commons.apache.org/proper/commons-lang/
Bundle-License: https://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-ManifestVersion: 2
Bundle-Name: Apache Commons Lang
Bundle-SymbolicName: org.apache.commons.lang3
Bundle-Vendor: The Apache Software Foundation
Bundle-Version: 3.9.0
Export-Package:
    org.apache.commons.lang3;version="3.9",org.apache.commons.lang3.arch;version="3.9"
    ,org.apache.commons.lang3.builder;version="3.9",org.apache.commons.lang3.concurrent;version="3.9"
    ,org.apache.commons.lang3.event;version="3.9",org.apache.commons.lang3.exception;version="3.9"
    ,org.apache.commons.lang3.math;version="3.9",org.apache.commons.lang3.mutable;version="3.9"
    ,org.apache.commons.lang3.reflect;version="3.9",org.apache.commons.lang3.text;version="3.9"
    ,org.apache.commons.lang3.text.translate;version="3.9",org.apache.commons.lang3.text.translate;version="3.9"
    ,org.apache.commons.lang3.time;version="3.9",org.apache.commons.lang3.tuple;version="3.9"
Include-Resource: META-INF/NOTICE.txt=NOTICE.txt,META-INF/LICENSE.txt=LICENSE.txt
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-4.1.0.201810181252
```

modules → jlink → custom runtime

- your modules
- jdk modules
- jdk runtime



```
jlink -p <module path>:$JAVA_HOME/jmods  
--add-modules <root module>  
--strip-debug  
--compress=2  
--output <out directory>
```

Doesn't work with Automatic Modules

```
jdeps --generate-module-info <out dir> lib/commons-lang3-3.9.jar
```

&
generate new jar

LAB
JEP 200

LANGUAGE

JEP 286: LOCAL-VARIABLE TYPE INFERENCE

Release: 10
Component: tools

Enhance the Java Language to extend type inference to declarations of local variables with initializers.

var

```
@RequestMapping({"/"})
public ModelAndView index() {
    ModelAndView mav = new ModelAndView("index");
    List<Person> allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}
```

I will not repeat myself
I will not repeat myself



DON'T REPEAT YOURSELF

Repetition is the root of all software evil.

```
@RequestMapping({"/"})
public ModelAndView index() {
    ModelAndView mav = new ModelAndView("index");
    List<Person> allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}
```

```
@RequestMapping({"","/"})
public ModelAndView index() {
    ModelAndView mav = new ModelAndView("index");
    List<Person> allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}
```

```
@RequestMapping({"" ,"/"})
public ModelAndView index() {
    var mav = new ModelAndView("index");
    var allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}
```

```

@RequestMapping({"/"})
public ModelAndView index() {
    ModelAndView mav = new ModelAndView("index");
    List<Person> allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}

/*
 * Decompiled with CFR 0.145.
 *
 * Could not load the following classes:
 * org.springframework.beans.factory.annotation.Autowired
 * org.springframework.stereotype.Controller
 * org.springframework.web.bind.annotation.RequestMapping
 * org.springframework.web.servlet.ModelAndView
 */
package net.javajudd.jep224.controllers;

import java.util.List;
import net.javajudd.jep224.domain.Person;
import net.javajudd.jep224.services.PeopleService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping(value={"/people"})
public class PeopleController {
    @Autowired
    PeopleService peopleService;

    @RequestMapping(value={"/", "/"})
    public ModelAndView index() {
        ModelAndView mav = new ModelAndView("index");
        List<Person> allPeople = this.peopleService.findAllPeople();
        mav.addObject("people", allPeople);
        return mav;
    }
}

```

LocalVariableTable:

Start	Length	Slot	Name	Signature
0	30	0	this	Lnet/javajudd/jep224/controllers/PeopleController;
10	20	1	mav	Lorg/springframework/web/servlet/ModelAndView;
20	10	2	allPeople	Ljava/util/List;

```

@RequestMapping({"/"})
public ModelAndView index() {
    var mav = new ModelAndView("index");
    var allPeople = peopleService.findAllPeople();
    mav.addObject("people", allPeople);
    return mav;
}

/*
 * Decompiled with CFR 0.145.
 *
 * Could not load the following classes:
 * org.springframework.beans.factory.annotation.Autowired
 * org.springframework.stereotype.Controller
 * org.springframework.web.bind.annotation.RequestMapping
 * org.springframework.web.servlet.ModelAndView
 */
package net.javajudd.jep224.controllers;

import java.util.List;
import net.javajudd.jep224.domain.Person;
import net.javajudd.jep224.services.PeopleService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping(value={"/people"})
public class PeopleController {
    @Autowired
    PeopleService peopleService;

    @RequestMapping(value={"/", "/"})
    public ModelAndView index() {
        ModelAndView mav = new ModelAndView("index");
        List<Person> allPeople = this.peopleService.findAllPeople();
        mav.addObject("people", allPeople);
        return mav;
    }
}

```

LocalVariableTable:

Start	Length	Slot	Name	Signature
0	30	0	this	Lnet/javajudd/jep224/controllers/PeopleController;
10	20	1	mav	Lorg/springframework/web/servlet/ModelAndView;
20	10	2	allPeople	Ljava/util/List;

```
List.of(1, 2.0, "3");
```

What is the return type?

```
List<? extends Serializable> list = List.of(1, 2.0, "3");
```

or

```
var list = List.of(1, 2.0, "3");
```

```
public class Person {  
  
    var id;  
    var firstName;  
    var lastName;  
    var age;  
  
    public var getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(var firstName) {  
        this.firstName = firstName;  
    }  
}
```

can not declare a field, return type or parameter

```
var s;  
s = "Hello World";
```

must declare and initialize

```
var s = "Hello World";  
s = 100;
```

strongly typed

```
var ints = {0, 1, 2};  
var appendSpace = a -> a + " ";  
var compareString = String::compareTo;
```

poly expressions

Can a variable be named var?

```
var var = "Hello World";  
assertThat(var).isEqualTo("Hello World");
```

yes, var is not a keyword

JEP 323: LOCAL-VARIABLE SYNTAX FOR LAMBDA PARAMETERS

Release: 11
Component: tools

Allow `var` to be used when declaring the formal parameters of implicitly typed lambda expressions.

Java 11 add var to Lambda parameters

```
list.stream()
    .map((var s) -> s.toLowerCase())
    .collect(Collectors.toList());
```

Java 11 add var to Lambda parameters

```
list.stream()
    .map((var s) -> s.toLowerCase())
    .collect(Collectors.toList());
```

Lambda's already act like var so why add vars to Lambda?

```
list.stream()
    .map(s -> s.toLowerCase())
    .collect(Collectors.toList());
```

Java 11 add var to Lambda parameters

```
list.stream()  
    .map((var s) -> s.toLowerCase())  
    .collect(Collectors.toList());
```

Lambda's already act like var so why add vars to Lambda?

```
list.stream()  
    .map(s -> s.toLowerCase())  
    .collect(Collectors.toList());
```

```
list.stream()  
    .map(@NotNull var s) -> s.toLowerCase()  
    .collect(Collectors.toList());
```



LAB

JEP 286: Local-Variable Type
Inference

JEP 323: Local-Variable Syntax for
Lambda Parameters

CORE LIBRARIES

JEP 321: HTTP CLIENT

Release: 11

Component: core-libs/java.net

Define a new HTTP client API that implements HTTP/2 and WebSocket, and can replace the legacy `HttpURLConnection` API.

```
import java.net.*;
import java.net.http.*;
import java.net.http.HttpClient.*;
import java.net.http.HttpResponse.*;
import java.time.*;

public class jcurl {

    public static void main(String[] args) throws Exception {

        HttpClient client = HttpClient.newBuilder()
            .version(Version.HTTP_1_1)
            .followRedirects(Redirect.NORMAL)
            .connectTimeout(Duration.ofSeconds(20))
            .build();

        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create(args[0]))
            .timeout(Duration.ofMinutes(1))
            .build();

        HttpResponse<String> response = client.send(request, BodyHandlers.ofString());
        System.out.println(response.statusCode());
        System.out.println(response.body());
    }
}
```

LAB

JEP 321: HttpClient

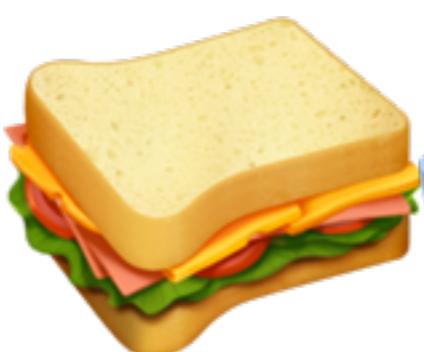
JEP 327: UNICODE 10

Release: 11

Component: core-libs/java.lang

Upgrade existing platform APIs to support version 10.0 of the Unicode Standard.





<https://emojipedia.org/unicode-10.0/>





฿

LAB

JEP 327: Unicode

<code>String.repeat(int)</code>	<code>ProcessHandle.pid()</code>
<code>String.lines()</code>	<code>Reader.transferTo()</code>
<code>String.strip()</code>	<code>Reader.nullReader()</code>
<code>String.stripLeading()</code>	
<code>String.stripTrailing()</code>	
<code>String.isBlank()</code>	<code>Writer.nullWriter()</code>
<code>List.of()</code>	<code>InputStream.nullInputStream()</code>
<code>List.copyOf()</code>	<code>InputStream.readAllBytes()</code>
<code>Collection.toArray()</code>	<code>InputStream.readNBytes()</code>
	<code>InputStream.transferTo()</code>
<code>Map.of()</code>	<code>OutputStream.nullOutputStream()</code>
<code>Map.ofEntries()</code>	
<code>Map.Entry.entry()</code>	<code>ByteArrayOutputStream.writeBytes()</code>
<code>Map.copyOf()</code>	
<code>Optional.isEmpty()</code>	<code>Files.readString()</code>
<code>Optional.ifPresentOrElse()</code>	<code>Files.writeString()</code>
<code>Optional.or()</code>	<code>Path.of()</code>
<code>Optional.stream()</code>	
<code>Optional.orElseThrow()</code>	<code>TimeUnit.toChronoUnit()</code>
<code>Predicate.not()</code>	<code>TimeUnit.of()</code>
	<code>TimeUnit.convert()</code>

SECURITY

JEP 332: TRANSPORT LAYER SECURITY (TLS)

1.3

Release: 11

Component: security-libs/java.net.ssl

Implement version 1.3 of the Transport Layer Security (TLS) Protocol RFC 8446.

Java 9

```
main, WRITE: TLSv1.2 Handshake, length = 40
[Raw write]: length = 45
0000: 16 03 03 00 28 00 00 00 00 00 00 00 C5 62 F9 .....(.....b.
0010: C6 4B 80 EE 26 C5 0D 1A 79 47 5F DA CD 3B DD 10 .K..&...yG_...;...
0020: D7 B5 27 64 DE 12 57 6D 35 35 00 1A EB ..'d..Wm55...
[Raw read]: length = 5
0000: 14 03 03 00 01 .....
[Raw read]: length = 1
0000: 01 .
main, READ: TLSv1.2 Change Cipher Spec, length = 1
```

Java 11

```
javax.net.ssl|DEBUG|0D|HttpClient-1-Worker-0|2019-07-08 21:12:46.516 EDT|ServerHello.java:  
884|Consuming ServerHello handshake message (  
"ServerHello": {  
    "server version" : "TLSv1.2",  
    "random" : "C8 62 CE 2D 4B E7 B9 A4 F5 C9 D5 17 89 62 DD 26 B9 D0 81 7D 20 43  
AD 47 93 8D 71 27 27 6B C3 07",  
    "session id" : "25 88 4F 27 6D F4 D5 59 8F 18 40 4C 61 D0 91 3E 4B F7 9B 52 1E 30  
3E 42 19 1A C0 AF 65 AF 47 FA",  
    "cipher suite" : "TLS_AES_128_GCM_SHA256(0x1301)",  
    "compression methods" : "00",  
    "extensions" : [  
        "key_share (51)": {  
            "server_share": {  
                "named group": secp256r1  
                "key_exchange": {  
                    0000: 04 25 B5 23 3D 04 58 76 FD CF C7 E1 69 24 55 30 %.#=.Xv....i$U0  
                    0010: 79 AD 9E E9 F0 79 CD 65 96 FD FA 29 E7 D5 54 6A y....y.e....)Tj  
                    0020: EB D0 27 7B C7 94 B2 2E 31 33 9B A6 D8 A6 00 5A ...'.13....Z  
                    0030: C6 A0 A0 90 B4 33 1F 2A AE 2D 41 D8 74 00 86 DF .....3.*.-A.t...  
                    0040: E8  
                }  
            },  
        },  
        "supported_versions (43)": {  
            "selected version": [TLSv1.3]  
        }  
    ]  
}  
}  
javax.net.ssl|DEBUG|0D|HttpClient-1-Worker-0|2019-07-08 21:12:46.516 EDT|SSLExtensions.java:  
188|Consumed extension: supported_versions  
javax.net.ssl|DEBUG|0D|HttpClient-1-Worker-0|2019-07-08 21:12:46.517 EDT|ServerHello.java:  
980|Negotiated protocol version: TLSv1.3
```

JEP 329: CHACHA20 AND POLY1305 CRYPTOGRAPHIC ALGORITHMS

Release: 11

Component: security-libs/javax.crypto

Implement the ChaCha20 and ChaCha20-Poly1305 ciphers as specified in RFC 7539. ChaCha20 is a relatively new stream cipher that can replace the older, insecure RC4 stream cipher.

<https://www.flickr.com/photos/krypto/373341548/>

<https://openjdk.java.net/jeps/329>

JEP 324: KEY AGREEMENT WITH CURVE25519 AND CURVE448



Release: 11

Component: security-libs / javax.crypto

Implement key agreement using Curve25519 and Curve448 as described in RFC 7748.

RUNTIME

JEP 328: FLIGHT RECORDER

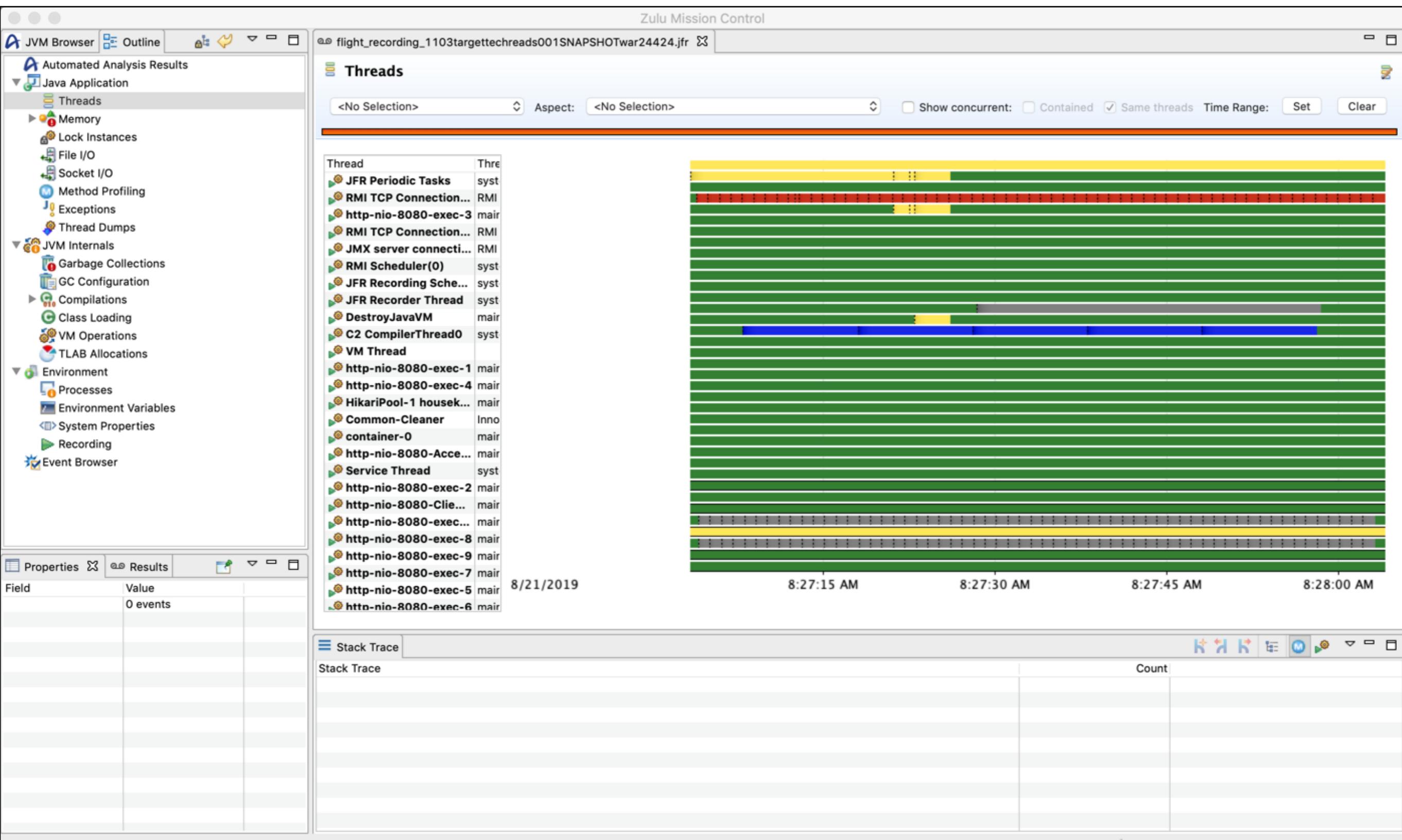
Release: 11
Component: hotspot/jfr

Provide a low-overhead data collection framework for troubleshooting Java applications and the HotSpot JVM.

```
java -XX:StartFlightRecording
```

Or

```
jcmd <pid> JFR.start  
jcmd <pid> JFR.dump filename=recording.jfr  
jcmd <pid> JFR.stop
```

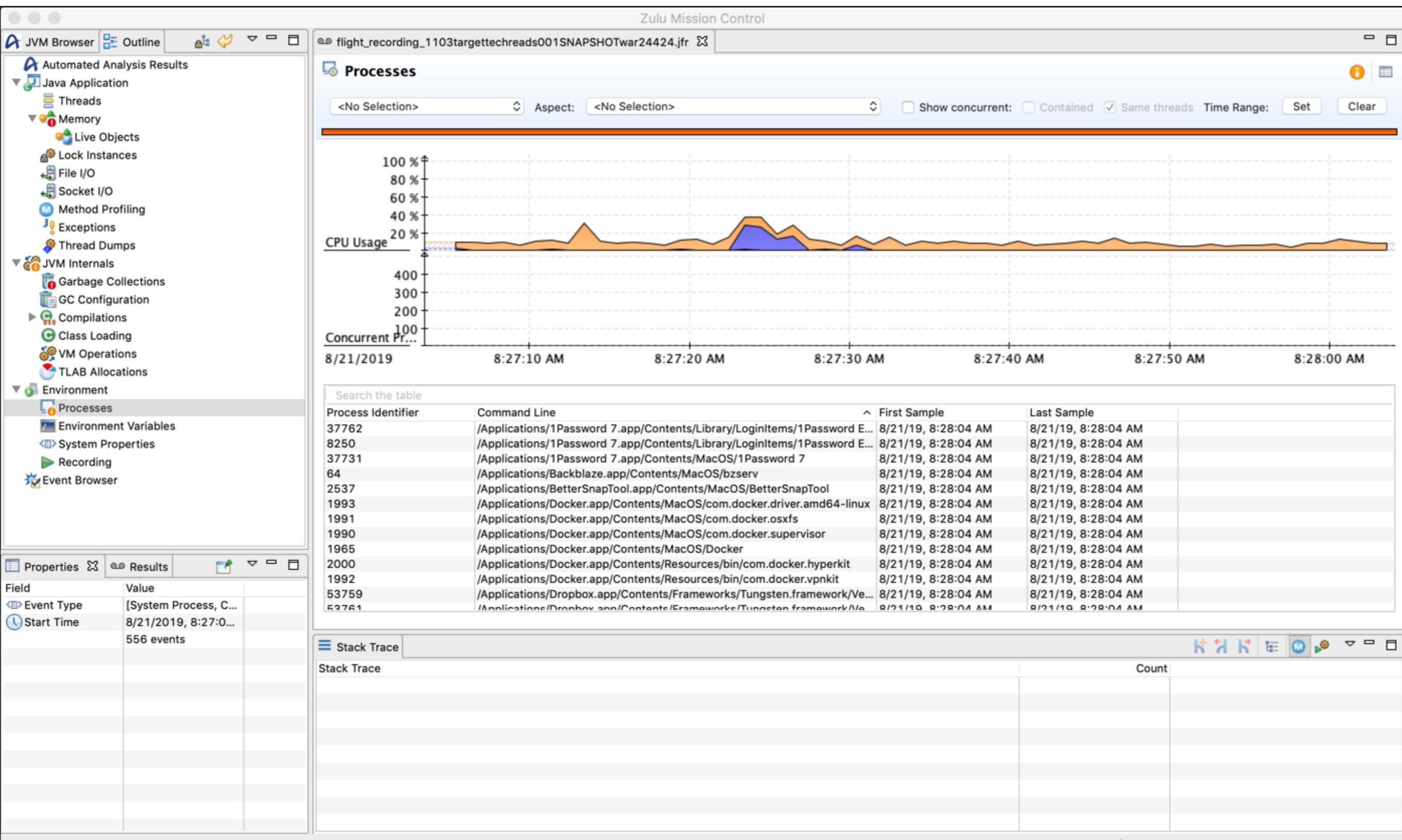


The screenshot shows the Zulu Mission Control interface with the following details:

- Toolbar:** Includes icons for JVM Browser, Outline, Lock Instances, Thread Dump, and Help.
- Left Sidebar (JVM Browser):**
 - Automated Analysis Results
 - Java Application
 - Threads
 - Memory
 - Live Objects
 - Lock Instances
 - File I/O
 - Socket I/O
 - Method Profiling
 - Exceptions
 - Thread Dumps
 - JVM Internals
 - Garbage Collections
 - GC Configuration
 - Compilations
 - Class Loading
 - VM Operations
 - TLAB Allocations
 - Environment
 - Processes
 - Environment Variables
 - System Properties
 - Recording
 - Event Browser
- Central Area:**
 - Memory:** A table with columns: Class, Max Live Count, Max Live Size, Live Size Increase, Total Allocation. The table is currently empty.
 - Allocation:** A graph showing memory usage over time. The Y-axis is labeled "Allocation" and ranges from 64 B to 768 B. The X-axis shows dates and times: 8/21/2019, 8:27:15 AM, 8:27:30 AM, 8:27:45 AM, 8:28:00 AM. A purple line represents the used heap, which starts at approximately 192 MiB and drops to 64 MiB around 8:27:30 AM. A legend on the right lists metrics: Garbage Collection (red), Total Allocation (blue), Used Heap (purple), Heap Space : Committed Size (green), Heap Space : Reserved Size (orange), Used Size (brown), Total Size (dark blue), and Live Size (yellow).
 - Memory Usage:** A table showing memory statistics for the heap summary. It includes fields like Event Type, Start Time, GC Identifier, When, Heap Space : St..., Heap Space : C..., Heap Space : C..., Heap Space : Re..., Heap Space : Re..., and Heap Used. The "Heap Used" row indicates 27.7 MiB – 195 MiB across 4 events.
 - Stack Trace:** A table titled "Stack Trace" with columns: Stack Trace and Count. The table is currently empty.
- Bottom Bar:** Includes icons for Home, Help, Recent, New, Open, Save, Print, and Exit.

The screenshot shows the Zulu Mission Control interface with the following details:

- Left Sidebar (JVM Browser):** Includes sections for Automated Analysis Results, Java Application (Threads, Memory, Live Objects, Lock Instances, File I/O, Socket I/O), Method Profiling, Exceptions, Thread Dumps, JVM Internals (Garbage Collections, GC Configuration, Compilations, Class Loading, VM Operations, TLAB Allocations), Environment (Processes, Environment Variables, System Properties, Recording, Event Browser), and a Properties panel.
- Central Area (Socket I/O):** A table showing I/O statistics for a connection to 10.199.28.240 (port 65,128). The connection was active for 59.887 seconds, with 63 reads and 63 writes, totaling 63 bytes read and 63 bytes written.
- Timeline View:** A bar chart titled "Socket Read" showing the timing of 63 read operations between 8:27:10 AM and 8:28:00 AM. The y-axis ranges from 512 x 2^-10 B to 3 B.
- Stack Trace:** A table listing the stack traces for the 63 read events. The most frequent method is `int java.net.SocketInputStream.read(byte[], int, int, int)`, which appears 63 times.



IMPROVED DOCKER/ CONTAINER SUPPORT

Release: 10

- adhere to container memory limits
- adhere to container available cpu
- adhere to container cpu limits



LAB

Docker

JEP 181: NEST-BASED ACCESS CONTROL

Release: 11

Component: hotspot/runtime

Nests allow nested classes that are part of the same enclosing class but compiled to different class files, to access each other's private members without the need for compilers to insert synthetic generated accessibility-broadening bridge methods. This is a Java class bytecode level change.



```
public class Outer {  
    private static int x = 5;  
  
    public static class Nested {  
        public static void doSomething () {  
            System.out.println(x);  
        }  
    }  
}
```

JEP 248: G1 Default GC

JEP 318: Epsilon: No-Op GC

JEP 333: ZGC: Scalable Low-Latency GC



JEP 248: MAKE G1 DEFAULT GC

Release: 9

Component: hotspot/gc

Make G1 the default garbage collector on 32- and 64-bit server configurations.

JEP 318: No-OP GC

Release: 11
Component: hotspot/gc

Develop a GC that handles memory allocation but does not implement any actual memory reclamation mechanism. Once the available Java heap is exhausted, the JVM will shut down.

Why would you want a Garbage
Collector that doesn't collect Garbage?

*Why would you want a Garbage
Collector that doesn't collect Garbage?*

- performance testing
- memory pressure testing
- extremely short lived jobs
- last-drop latency improvements
- last-drop throughput improvements

`-XX:+UnlockExperimentalVMOptions -XX:+UseEpsilonGC`

JEP 333: ZGC: SCALABLE LOW-LATENCY GC

Release: 11
Component: hotspot/gc

ZGC is a scalable low-latency garbage collector.

- GC pause times should not exceed 10ms
- Handle heaps ranging from relatively small (a few hundreds of megs) to very large (many terabytes) in size
- No more than 15% application throughput reduction compared to using G1
- Initially supported by Linux/x64 platforms

`-XX:+UnlockExperimentalVMOptions -XX:+UseZGC`

Removed



Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations



Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations

Maven™

com.sun.xml.ws : jaxws-ri
com.sun.xml.bind : jaxb-ri
wsgen and wsimport: com.sun.xml.ws : jaxws-tools
schemagen and xjc: com.sun.xml.bind : jaxb-jxc
com.sun.xml.bind : jaxb-xjc
javax.xml.ws : jaxws-api
javax.xml.bind : jaxb-api
javax.activation : javax.activation-api
javax.annotation : javax.annotation-api



Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations



- JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions

Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations
- JEP 313: Native-Header Gen Tool
 - javah
- JEP 240: JVM TI hprof Agent
- JEP 241: jhat Tool



- JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions

Removed

- JEP 320: Java EE
- JAX-WS (Java API for XML-Based Web Services)
- JAXB (Java Architecture for XML Binding)
- JAF (JavaBean Activation Framework)
- Common Annotations



- JEP 313: Native-Header Gen Tool
- javah
- JEP 240: JVM TI hprof Agent
- JEP 241: jhat Tool
- JEP 298: Demos and Samples
- JEP 214: GC Combinations

- JEP 320: CORBA Modules
- RMI-IIOP
- Java IDL
- JTA (Java Transaction API)*
- J2EE Activity Service for Extended Transactions

Removed

- JEP 320: Java EE
- JAX-WS (Java API for XML-Based Web Services)
- JAXB (Java Architecture for XML Binding)
- JAF (JavaBean Activation Framework)
- Common Annotations
- JEP 313: Native-Header Gen Tool
- javah
- JEP 240: JVM TI hprof Agent
- JEP 241: jhat Tool
- JEP 298: Demos and Samples
- JEP 214: GC Combinations



- JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions

```
DefNew + CMS          : -XX:-UseParNewGC -XX:+UseConcMarkSweepGC
ParNew + SerialOld   : -XX:+UseParNewGC
ParNew + iCMS         : -Xincgc
ParNew + iCMS         : -XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC
DefNew + iCMS         : -XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC -XX:-UseParNewGC
CMS foreground        : -XX:+UseCMSCompactAtFullCollection
CMS foreground        : -XX:+CMSFullGCsBeforeCompaction
CMS foreground        : -XX:+UseCMSCollectionPassing
```

Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations
 - JEP 313: Native-Header Gen Tool
 - javah
 - JEP 240: JVM TI hprof Agent
 - JEP 241: jhat Tool
 - JEP 298: Demos and Samples
 - JEP 214: GC Combinations
 - JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions
 - JEP 231: Launch-Time JRE Version Selection
- 

Removed

- JEP 320: Java EE
- JAX-WS (Java API for XML-Based Web Services)
- JAXB (Java Architecture for XML Binding)
- JAF (JavaBean Activation Framework)
- Common Annotations

- SNMP Agent
- Java Applet Plug-in & Web Start
- policytool

- JEP 320: CORBA Modules
- RMI-IIOP
- Java IDL
- JTA (Java Transaction API)*
- J2EE Activity Service for Extended Transactions



- JEP 313: Native-Header Gen Tool
- javah

- JEP 240: JVM TI hprof Agent

- JEP 241: jhat Tool

- JEP 298: Demos and Samples

- JEP 214: GC Combinations

- JEP 231: Launch-Time JRE Version Selection

Removed

- JEP 320: Java EE
 - JAX-WS (Java API for XML-Based Web Services)
 - JAXB (Java Architecture for XML Binding)
 - JAF (JavaBean Activation Framework)
 - Common Annotations
 - SNMP Agent
 - Java Applet Plug-in & Web Start
 - policytool
 - Thread's destroy() & stop(Throwable)
 - System & Runtime's runFinalizersOnExit()
 - JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions
- 
- JEP 313: Native-Header Gen Tool
 - javah
 - JEP 240: JVM TI hprof Agent
 - JEP 241: jhat Tool
 - JEP 298: Demos and Samples
 - JEP 214: GC Combinations
 - JEP 231: Launch-Time JRE Version Selection

Removed

- JEP 320: Java EE
- JAX-WS (Java API for XML-Based Web Services)
- JAXB (Java Architecture for XML Binding)
- JAF (JavaBean Activation Framework)
- Common Annotations
- SNMP Agent
- Java Applet Plug-in & Web Start
- policytool
- Thread's destroy() & stop(Throwable)
- System & Runtime's runFinalizersOnExit()
- JEP 320: CORBA Modules
 - RMI-IIOP
 - Java IDL
 - JTA (Java Transaction API)*
 - J2EE Activity Service for Extended Transactions
- JEP 313: Native-Header Gen Tool
 - javah
- JEP 240: JVM TI hprof Agent
- JEP 241: jhat Tool
- JEP 298: Demos and Samples
- JEP 214: GC Combinations
- JEP 231: Launch-Time JRE Version Selection
 - Windows XP support



Removed

JavaFX
<https://openjfx.io/>

VisualVM
<https://visualvm.github.io/>



Java Mission Control
<https://jdk.java.net/jmc/>

<https://www.azul.com/products/zulu-mission-control/>

Migrating from Oracle JDK

The following table illustrates the proprietary components that can be found in Oracle JDK 8 and the alternative technologies that are either planned or available to replace them. Click on the links to learn more about each component and any steps that might be necessary to adopt them.

Oracle JDK 8 proprietary component	Alternative component	OpenJDK 8	OpenJDK 11
Java Web Start	IcedTea-Web	✓	✗
JavaFX	OpenJFX	✗	✓ (coming soon)
T2K font rendering engine	Freetype	✓	✓
Monotype Lucida fonts	Relicensed Lucida fonts	✗ (coming soon)	✗ (coming soon)
Ductus 2D renderer	Pisces/Marlin	✓ (Pisces)	✓ (Marlin)
Kodac Color Matching System (KCMS) library	LCMS	✓	✓
SNMP	Use JMX (or SNMP4J)	✓ (not bundled)	✓ (not bundled)
Sound drivers	Use Windows sound drivers	✓ (not bundled)	✓ (not bundled)
Java Flight Recorder (JFR)	Java Flight Recorder	✗ (coming soon)	✓
Java Mission Control (JMC)	Use JDK Mission Control	✗ (coming soon)	✗ (coming soon)

Deprecated



Deprecated

- JEP 289: Applet API
 - java.applet.AppletStub
 - java.applet.Applet
 - java.applet.AudioClip
 - java.applet.AppletContext
 - javax.swing.JApplet



Deprecated

- JEP 289: Applet API
 - java.applet.AppletStub
 - java.applet.Applet
 - java.applet.AudioClip
 - java.applet.AppletContext
 - javax.swing.JApplet



- JEP 335: Nashorn JavaScript Engine
 - jdk.scripting.nashorn
 - jdk.scripting.nashorn.shell
 - jjs

Deprecated

- JEP 289: Applet API
 - java.applet.AppletStub
 - java.applet.Applet
 - java.applet.AudioClip
 - java.applet.AppletContext
 - javax.swing.JApplet

- JEP 291: Concurrent Mark Sweep (CMS) GC
 - -XX:+UseConcMarkSweepGC



- JEP 335: Nashorn JavaScript Engine
 - jdk.scripting.nashorn
 - jdk.scripting.nashorn.shell
 - jjs

Deprecated

- JEP 289: Applet API
 - java.applet.AppletStub
 - java.applet.Applet
 - java.applet.AudioClip
 - java.applet.AppletContext
 - javax.swing.JApplet

- JEP 291: Concurrent Mark Sweep (CMS) GC
 - -XX:+UseConcMarkSweepGC



- JEP 335: Nashorn JavaScript Engine
 - jdk.scripting.nashorn
 - jdk.scripting.nashorn.shell
 - jjs

- JEP 336: Pack200 Tools and API
 - pack200
 - unpack200

Practical Suggestions

- Don't plan on just upgrading the JDK and running app
- Analyze dependencies with jdeps
 - Encapsulated and moved
- Side by Side CI
- Upgrade Tools & libraries
 - CI, Build, IDE
 - If using Spring Boot, first upgrade to newest version
 - Replace Cobertura with JaCoCo
 - Update Java byte code enhancement libraries
 - ASM
 - bytebuddy
 - javaassist
 - cglib
- Change variables that start with _
- No more rt.jar and tools.jar
- Many command-line options deprecated or remove

[WARNING]

```
java.lang.reflect.InvocationTargetException
  at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.base/java.lang.reflect.Method.invoke(Method.java:566)
  at org.springframework.boot.maven.AbstractRunMojo$LaunchRunner.run(AbstractRunMojo.java:527)
  at java.base/java.lang.Thread.run(Thread.java:834)
```

Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in class path resource [org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaAutoConfiguration.class]: Invocation of init method failed; nested exception is java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException

```
  at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1628)
  at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:555)
  at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:483)
  at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:306)
  at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
  at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:302)
  at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:197)
  at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:1078)
  at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:857)
  at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:543)
  at org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.refresh(EmbeddedWebApplicationContext.java:122)
  at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:693)
  at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:360)
  at org.springframework.boot.SpringApplication.run(SpringApplication.java:303)
  at org.springframework.boot.SpringApplication.run(SpringApplication.java:1118)
  at org.springframework.boot.SpringApplication.run(SpringApplication.java:1107)
  at com.manifestcorp.techreads.TechreadsApplication.main(TechreadsApplication.java:10)
  ... 6 more
```

Caused by: java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException

```
  at org.hibernate.boot.spi.XmlMappingBinderAccess.<init>(XmlMappingBinderAccess.java:43)
  at org.hibernate.boot.MetadataSources.<init>(MetadataSources.java:87)
  at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.<init>(EntityManagerFactoryBuilderImpl.java:179)
  at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.<init>(EntityManagerFactoryBuilderImpl.java:149)
  at org.springframework.orm.jpa.vendor.SpringHibernateJpaPersistenceProvider.createContainerEntityManagerFactory(SpringHibernateJpaPersistenceProvider.java:54)
  at org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean.createNativeEntityManagerFactory(LocalContainerEntityManagerFactoryBean.java:353)
  at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.buildNativeEntityManagerFactory(AbstractEntityManagerFactoryBean.java:370)
  at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.afterPropertiesSet(AbstractEntityManagerFactoryBean.java:359)
  at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:1687)
  at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1624)
  ... 22 more
```

Caused by: java.lang.ClassNotFoundException: javax.xml.bind.JAXBException

```
  at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:471)
  at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:588)
  at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
  ... 32 more
```

[INFO] -----

[INFO] BUILD FAILURE

[INFO] -----

[INFO] Total time: 5.281 s

[INFO] Finished at: 2019-08-21T08:05:42-04:00

[INFO] Final Memory: 41M/154M

[INFO] -----

[ERROR] Failed to execute goal org.springframework.boot:spring-boot-maven-plugin:1.5.4.RELEASE:run (default-cli) on project techreads: An exception occurred while running. null: InvocationTargetException: Error creating bean with name 'entityManagerFactory' defined in class path resource [org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaAutoConfiguration.class]: Invocation of init method failed; nested exception is java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException: javax.xml.bind.JAXBException -->

[Help 1]

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException



The Home of the Java Almanac

Collection of information about the history of Java primarily from a technical point of view.

- [Versions: 1.0 - 1.1 - 1.2 - 1.3 - 1.4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15](#)
- [References](#)

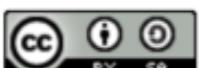
API Evolution

from	to	to	to	to	to	to	to
Java 8	Java 9	Java 10	Java 11	Java 12	Java 13	Java 14	Java 15
Java 9		Java 10	Java 11	Java 12	Java 13	Java 14	Java 15
Java 10			Java 11	Java 12	Java 13	Java 14	Java 15
Java 11				Java 12	Java 13	Java 14	Java 15
Java 12					Java 13	Java 14	Java 15
Java 13						Java 14	Java 15
Java 14							Java 15

The reports are based on simple comparison of all public and protected APIs of the `java.*` and `javax.*` packages of the respective JDKs. They do not make statements about [compatibility](#).

More detailed API comparisons based on the [japicmp](#) tool can be found [here](#).

Author & License



This content is provided by Java Champion Marc R. Hoffmann under the [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.



New APIs in Java 11

Comparing **Java 11** (build 11.0.5.hs-adpt) with **Java 8** (build 8.0.232.hs-adpt).

Element	Modification
java.awt.desktop	added
java.awt.dnd	
DragSourceContext	
• DragSourceContext(DragGestureEvent, Cursor, Image, int)	added
• DragSourceContext(DragSourceContextPeer, int)	removed
DragSource	
• createDragSourceContext(DragGestureEvent, Cursor, Image, int)	added
• createDragSourceContext(DragSourceContextPeer, int)	removed
DropTargetContext	
• addNotify(DropTargetContextPeer)	removed
• removeNotify()	removed
DropTarget	
• addNotify()	added
• addNotify(ComponentPeer)	removed
• removeNotify()	added
• removeNotify(ComponentPeer)	removed
java.awt.event	
FocusEvent	
• FocusEvent(Component, int, boolean, Component, long)	added
• getCause()	added
FocusEvent.Cause	added
java.awt.font	
TextLayout	
• equals(Object)	removed
• hashCode()	removed
NumericShaper.Range	
◦ MYANMAR_TAI_LAING	added
◦ SINHALA	added
java.awt.geom	
Path2D.Double	
• trimToSize()	added
Path2D.Float	
• trimToSize()	added
Path2D	
• trimToSize()	added
java.awt.image	
AbstractMultiResolutionImage	added
BaseMultiResolutionImage	added
ComponentColorModel	
• hashCode()	added
IndexColorModel	
• equals(Object)	added
• hashCode()	added
PackedColorModel	

LAB

Update Project

Christopher M. Judd



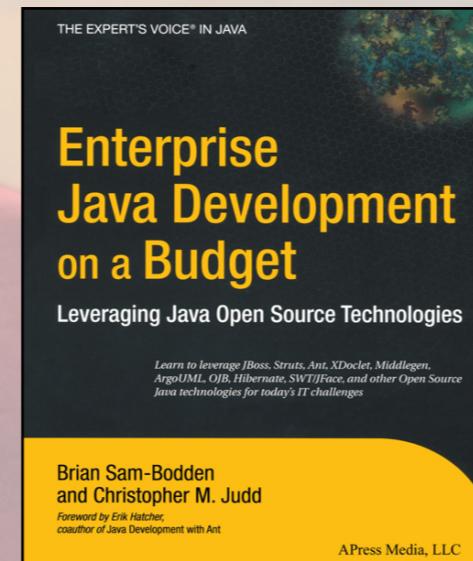
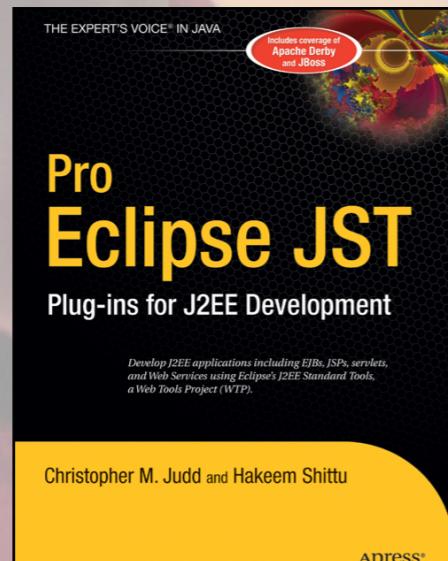
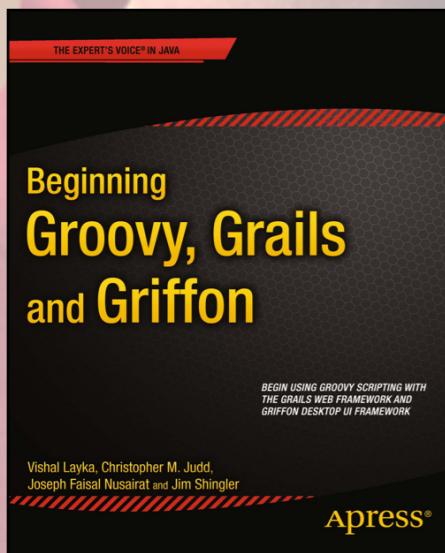
CTO and Partner

email: javajudd@gmail.com

web: www.juddsolutions.com

blog: juddsolutions.blogspot.com

twitter: [javajudd](https://twitter.com/javajudd)



A screenshot of a DZone Refcardz page titled 'Getting Started With Docker'. The page has a header with the DZone logo and 'packet'. It includes sections like 'ABOUT DOCKER', 'DOCKER ARCHITECTURE', and 'Clients', 'Hosts', 'Regions'. There are also links to 'packet' and 'Improve Performance, Minimize Cost'.

