Java.next() et Jigsaw

ParisJUG, le 13 mars 2012

Alexis Moussine-Pouchkine

Programmer's SELECTION

Beginning Java EE 6

Antonio Goncalves 著
日本オラクル株式会社 監訳
株式会社プロシステムエルオーシー 訳

GlassFish 3で始める
エンタープライズJava

サンプルWebアプリに触れながら、
次世代の開発標準Java EE 6を体系的に習得する

SE SHOEISHA

Beginning Java EE 6 Platform with GlassFish 3:
From Novice to Professional, Second Edition

Java.next() et Jigsaw

ParisJUG, le 13 mars 2012

Alexis Moussine-Pouchkine

A long long time ago…

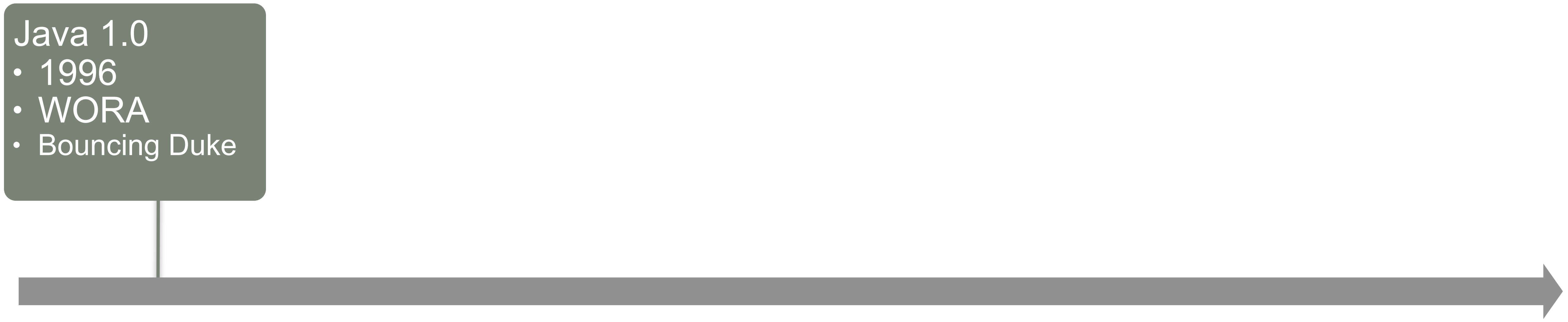
Java 1.0
- 1996
- WORA
- Bouncing Duke

Java 1.0

- 1996
- WORA
- Bouncing Duke

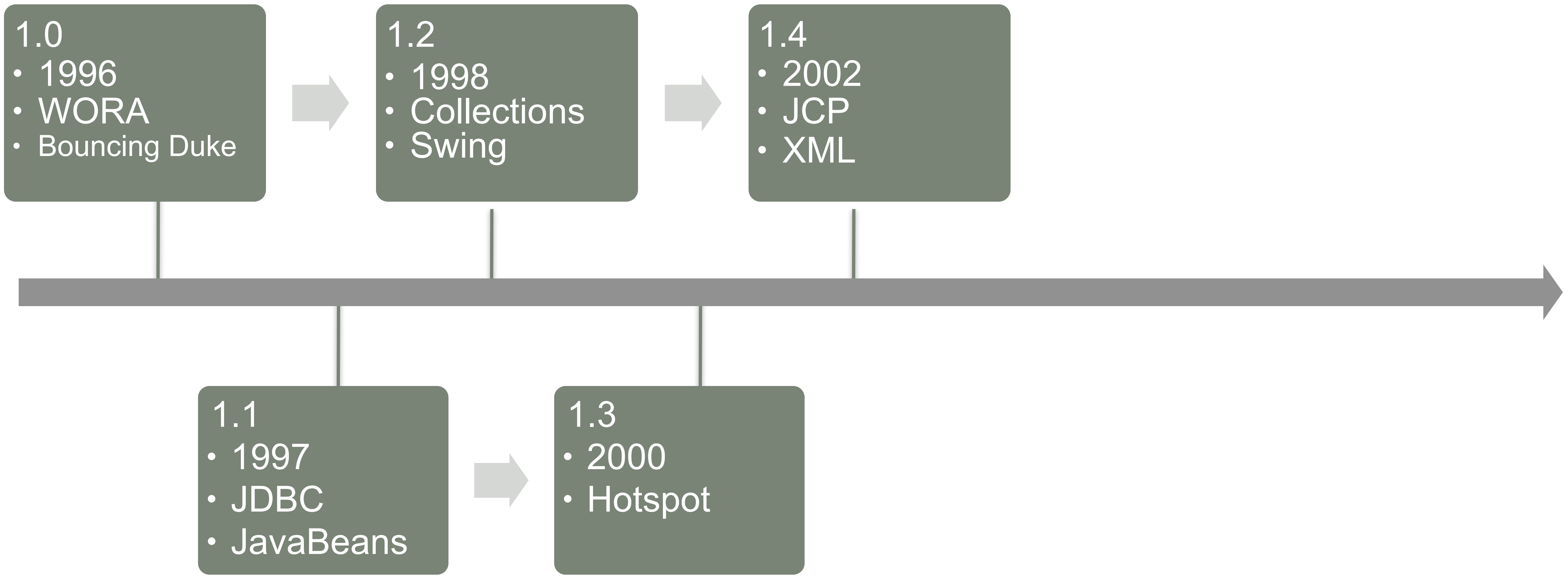
Java 1.0

- 1996
- WORA
- Bouncing Duke



Java 1.0

- 1996
- WORA
- Bouncing Duke


1.0
• 1996
• WORA
• Bouncing Duke
1.1
• 1997
• JDBC
• JavaBeans
1.2
• 1998
• Collections
• Swing
1.3
• 2000
• Hotspot
1.4
• 2002
• JCP
• XML

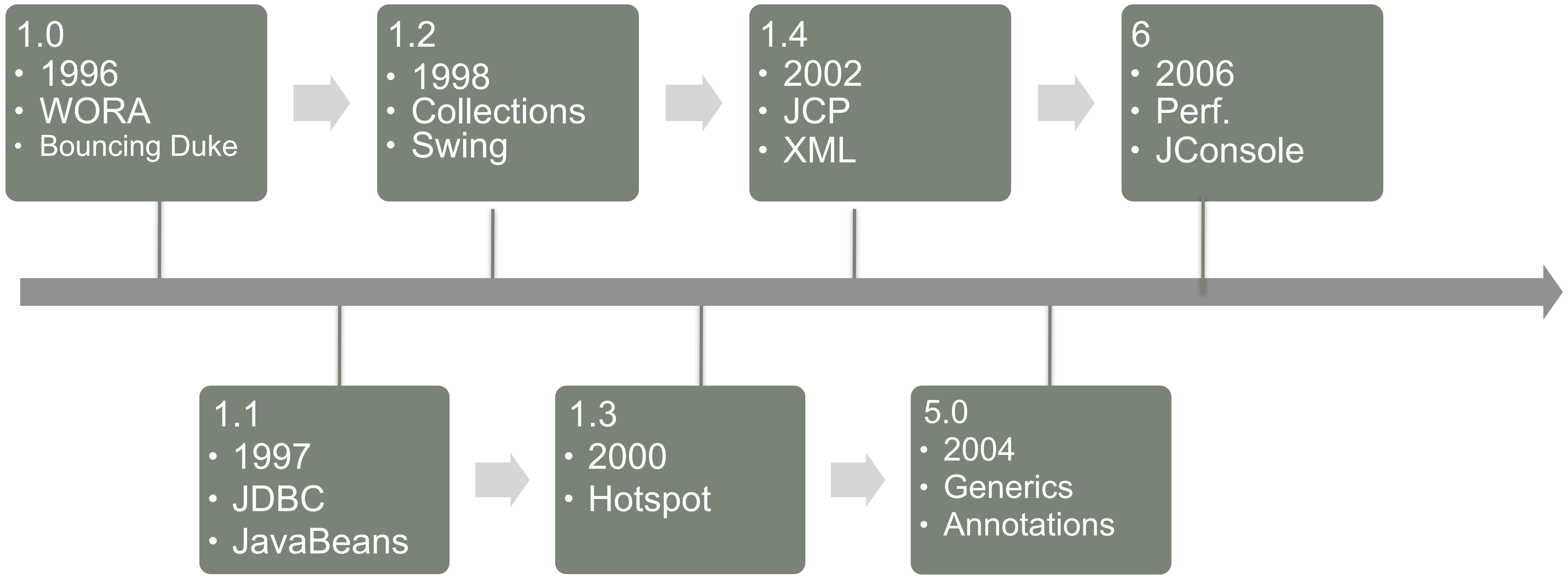
1.0
- 1996
- WORA
- Bouncing Duke

1.1
- 1997
- JDBC
- JavaBeans

1.2
- 1998
- Collections
- Swing

1.3
- 2000
- Hotspot

1.4
- 2002
- JCP
- XML


1.0
• 1996
• WORA
• Bouncing Duke
1.1
• 1997
• JDBC
• JavaBeans
1.2
• 1998
• Collections
• Swing
1.3
• 2000
• Hotspot
1.4
• 2002
• JCP
• XML
5.0
• 2004
• Generics
• Annotations
6
• 2006
• Perf.
• JConsole

1.0
- 1996
- WORA
- Bouncing Duke

1.1
- 1997
- JDBC
- JavaBeans

1.2
- 1998
- Collections
- Swing

1.3
- 2000
- Hotspot

1.4
- 2002
- JCP
- XML

5.0
- 2004
- Generics
- Annotations

6
- 2006
- Perf.
- JConsole

Java 7
- 2011
- Project Coin

1.0
- 1996
- WORA
- Bouncing Duke

1.1
- 1997
- JDBC
- JavaBeans

1.2
- 1998
- Collections
- Swing

1.3
- 2000
- Hotspot

1.4
- 2002
- JCP
- XML

5.0
- 2004
- Generics
- Annotations

6
- 2006
- Perf.
- JConsole

Java 7
- 2011
- Project Coin

Java SE 7 Release Contents

JSR-336: Java SE 7 Release Contents

- Java Language
  - Project Coin (JSR-334)
- Class Libraries
  - NIO2 (JSR-203)
  - Fork-Join framework, ParallelArray (JSR-166y)
- Java Virtual Machine
  - InvokeDynamic bytecode (JSR-292)
- Miscellaneous enhancements

Project Coin

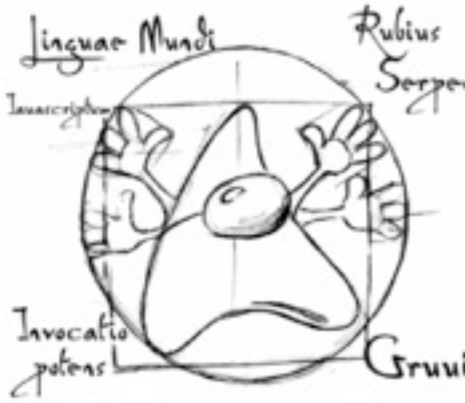
- First language changes since 2004 (Java 5)
  - Strings in switch statements
  - Multi-catch and precise re-throw
  - Try-with-resources (AutoCloseable)
  - Diamond operator
  - Simplified Varargs
  - Binary literals and underscores in numeric literals

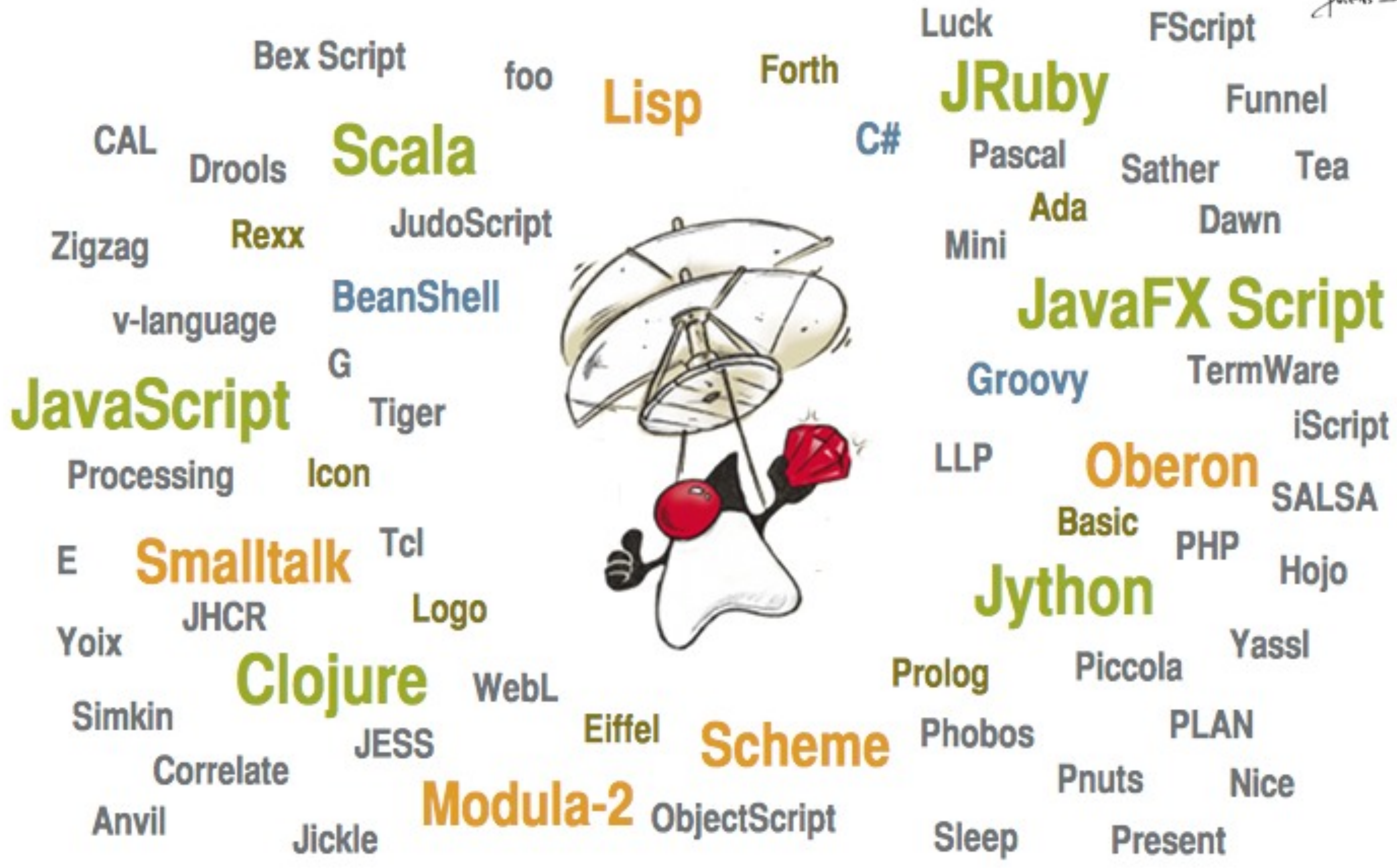
http://www.flickr.com/photos/chefranden/908539119/

New I/O 2 (NIO2) Libraries
JSR 203

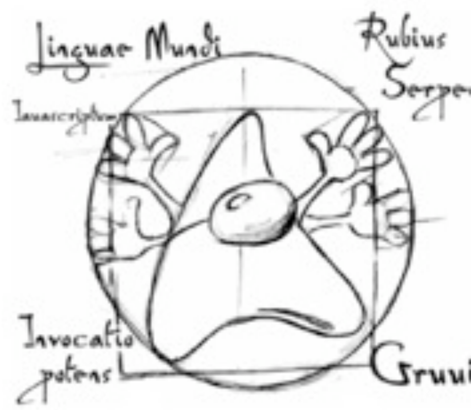
- Designed to be extensible
- Access to file metadata
- Utility methods (e.g. copy(), move(), ...)
- Higher abstraction (java.nio.file.Path)
- Consistent exception handling

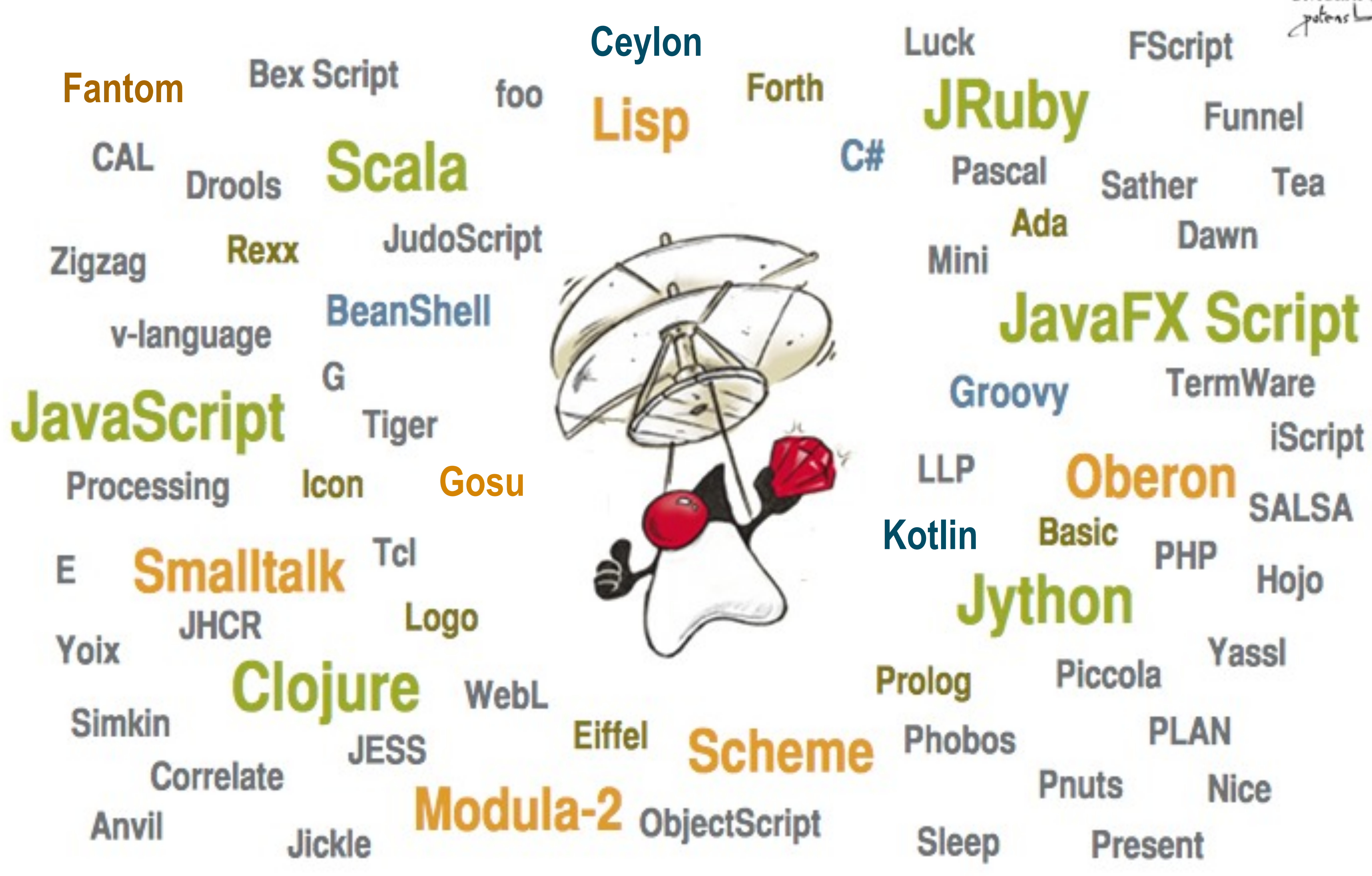
invokedynamic





invokedynamic

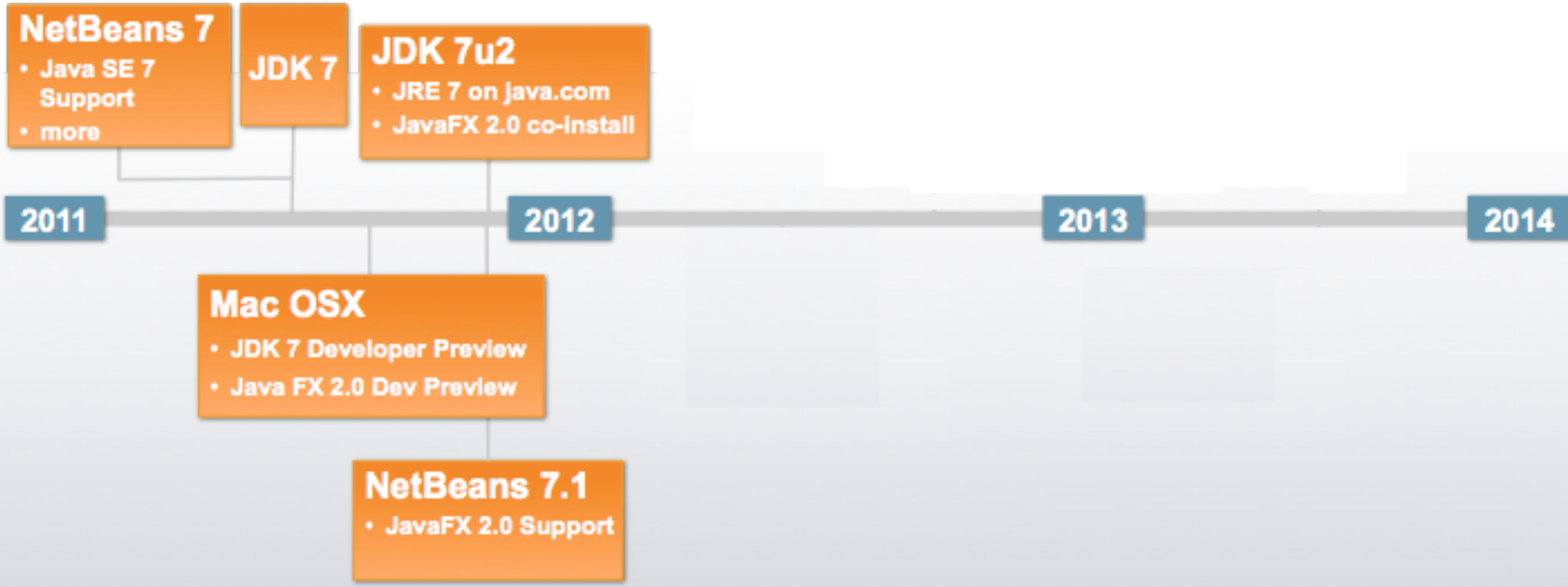




- JDBC 4.1, RowSet 1.1
- Security: Elliptic curve cryptography, TLS 1.2
- Unicode 6
- JAXP 1.4.4, JAX-WS 2.2, JAXB 2.2
- Swing: Nimbus L&F, JXLayer, HW accelerations
- ClassLoader architecture changes
- close() for URLClassLoader
- Javadoc support for CSS
- New Objects class
- ...

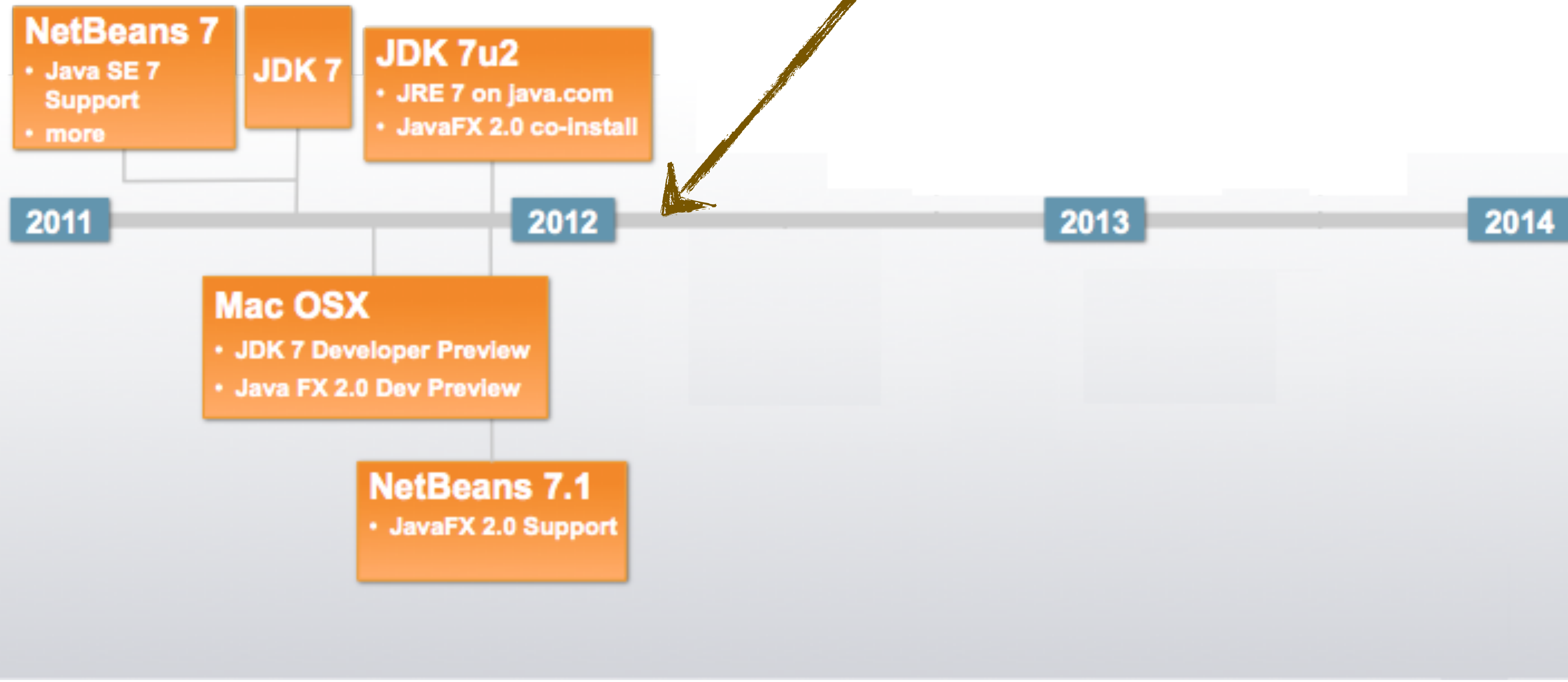
JavaOne 2011

Java Roadmap

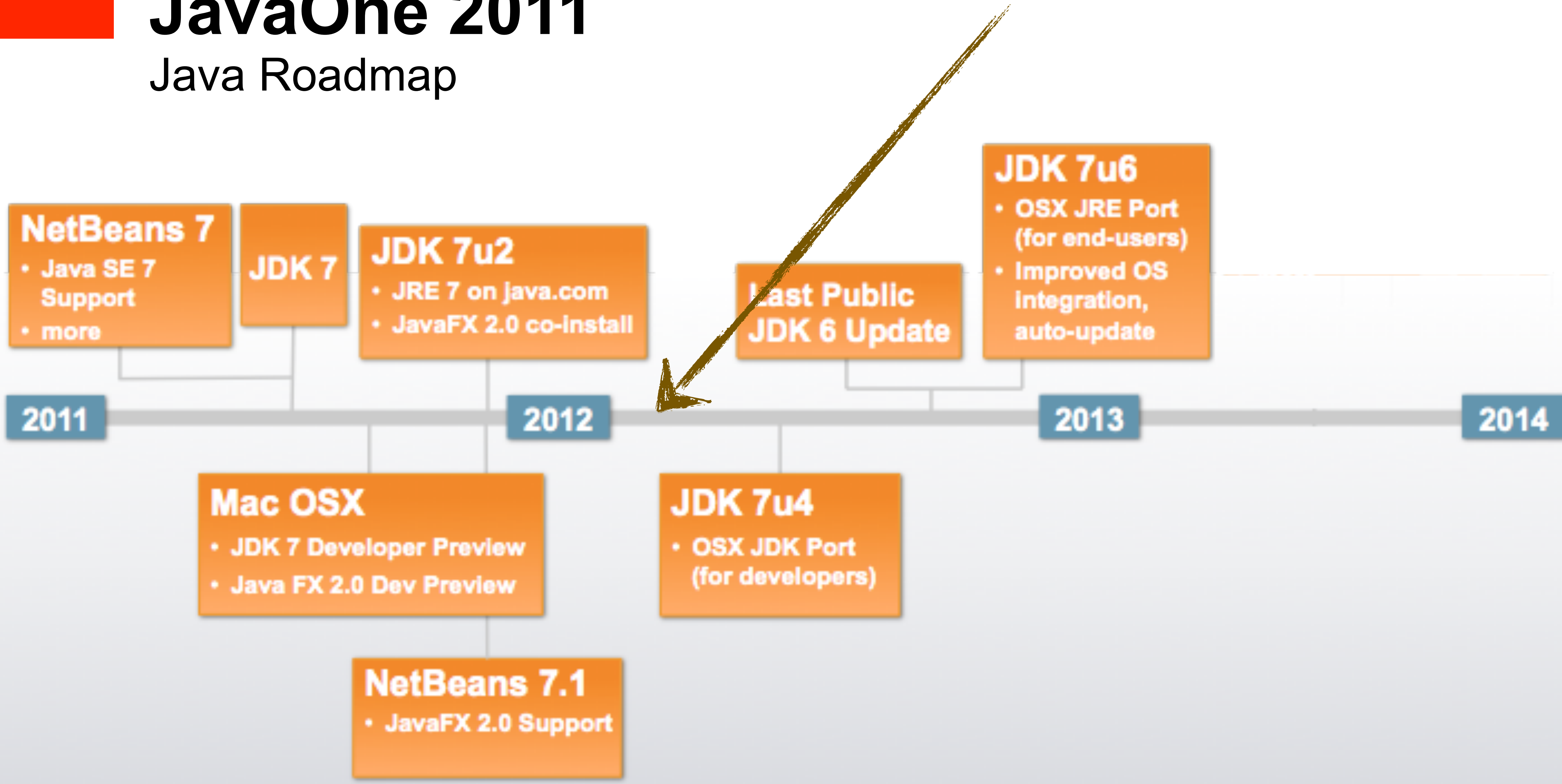


JavaOne 2011

Java Roadmap



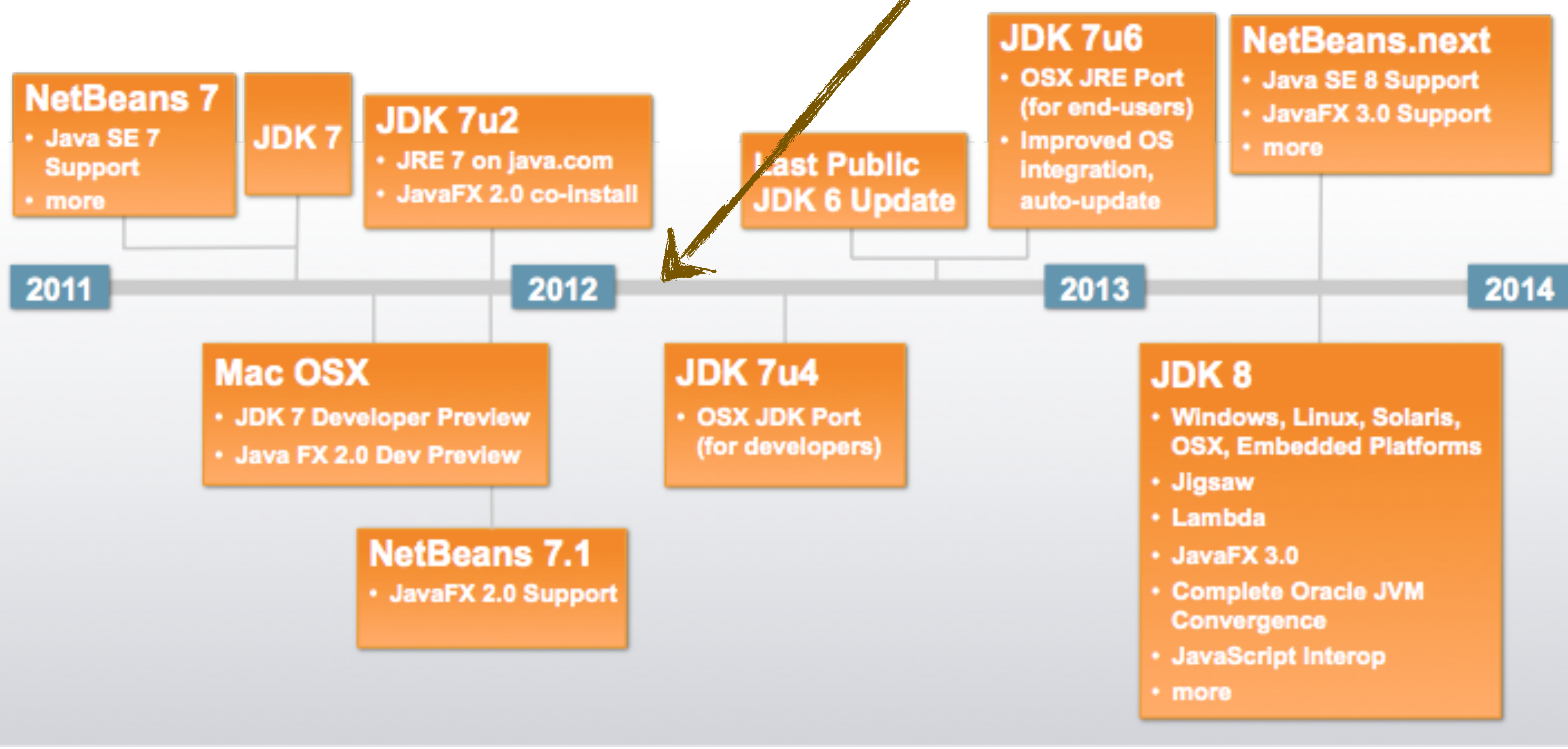
JavaOne 2011
Java Roadmap



ORACLE

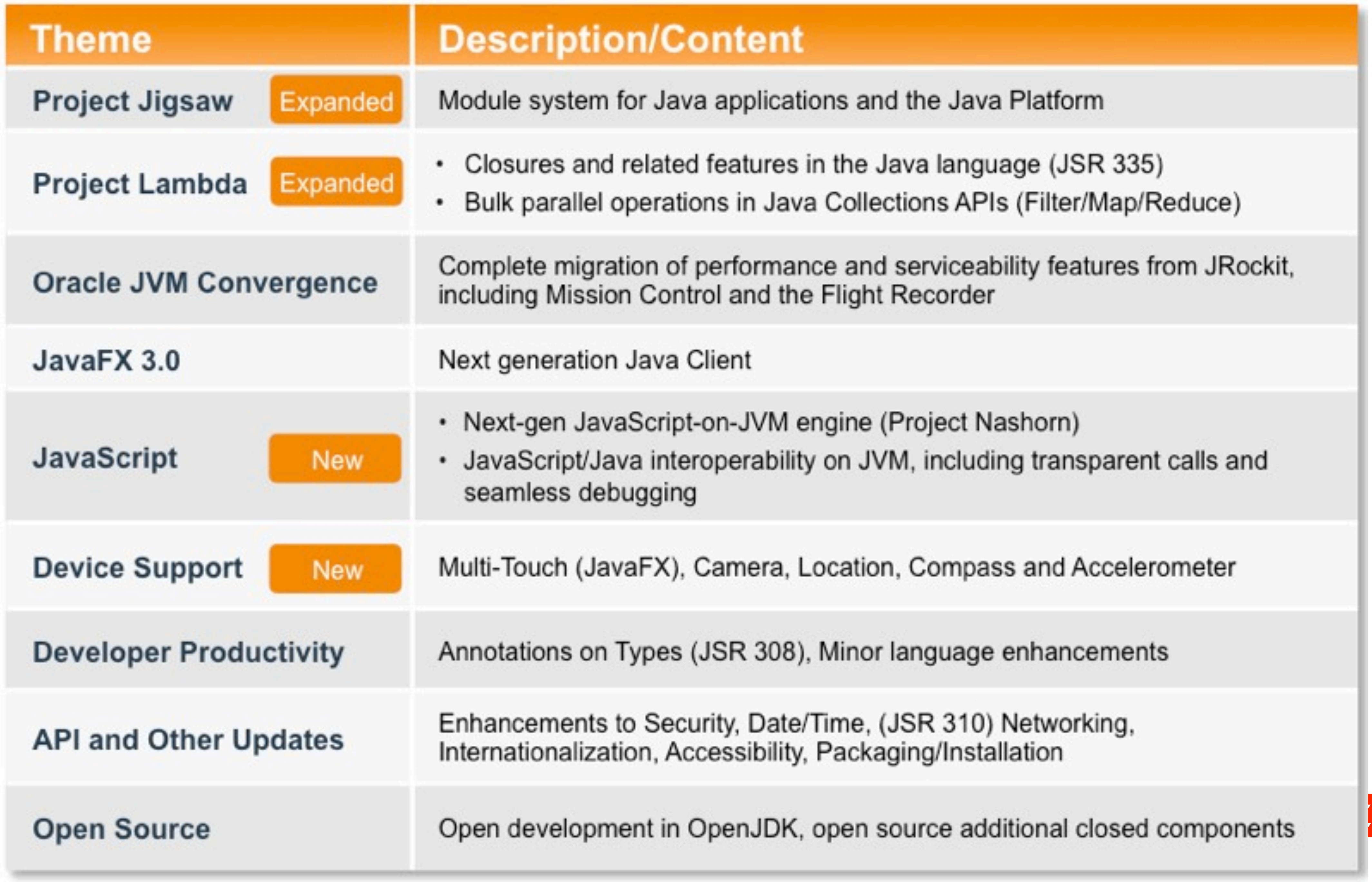
JavaOne 2011

Java Roadmap



JDK 8 – Summer 2013

- Strong feedback - 2 years needed between JDK releases
- Release date revised to summer 2013 (from late 2012)



Theme		Description/Content
Project Jigsaw	Expanded	Module system for Java applications and the Java Platform
Project Lambda	Expanded	• Closures and related features in the Java language (JSR 335) • Bulk parallel operations in Java Collections APIs (Filter/Map/Reduce)
Oracle JVM Convergence		Complete migration of performance and serviceability features from JRockit, including Mission Control and the Flight Recorder
JavaFX 3.0		Next generation Java Client
JavaScript	New	• Next-gen JavaScript-on-JVM engine (Project Nashorn) • JavaScript/Java interoperability on JVM, including transparent calls and seamless debugging
Device Support	New	Multi-Touch (JavaFX), Camera, Location, Compass and Accelerometer
Developer Productivity		Annotations on Types (JSR 308), Minor language enhancements
API and Other Updates		Enhancements to Security, Date/Time, (JSR 310) Networking, Internationalization, Accessibility, Packaging/Installation
Open Source		Open development in OpenJDK, open source additional closed components

OpenJDK FAQ
Installing
Contributing
Sponsoring
Developers' Guide
Mailing lists
Bylaws
Census
Legal
GB Election
JEP Process

OpenJDK.org

OpenJDK

- **Projects**
  - Jigsaw, Lambda, Multi-language VM, OpenJFX, Mac OS X Port
  - etc...
- **JEPs** (JDK Enhancement Proposals)
  - JEP 122: Remove the Permanent Generation
  - JEP 126: Lambda Expressions and Virtual Extension Methods
  - JEP 135: Base64 Encoding and Decoding
  - JEP 147: Reduce Class Metadata Footprint
  - JEP 148: Small VM
  - JEP 150: Date and Time API, JSR 310
  - etc...

JSR 277 - Java Modules

~~JSR 277 - Java Modules~~
OSGi

~~JSR 277 - Java Modules~~

~~OSGi~~

JSR 294 - Super Packages

~~JSR 277 - Java Modules~~

~~OSGi~~

~~JSR 294 - Super Packages~~

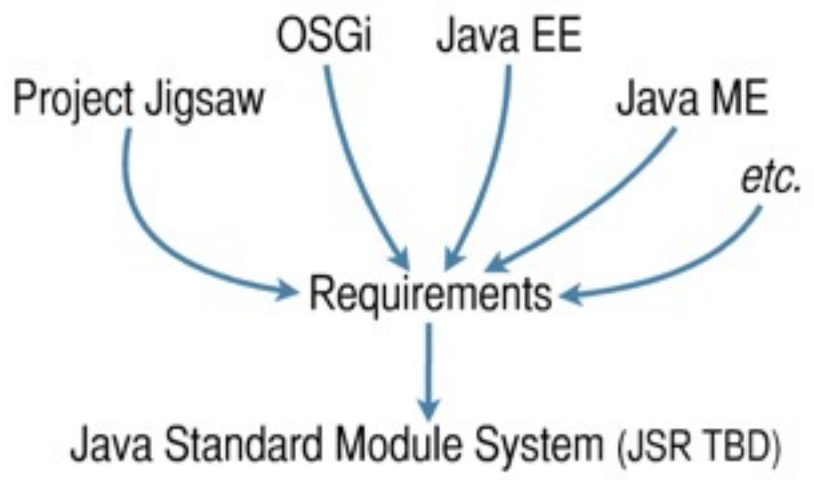
Jigsaw

Java™

Problem Space

- Application construction, packaging and publication
  - "JAR hell"
- Platform Scalability
  - Down to small devices
- Performance
  - Download time
  - Startup time

Jigsaw in one slide



- Goals
  - Simple static module system
  - Usable by every Java developer
  - Powerful enough to modularize the JDK itself
  - Increase performance
- Consequences
  - The end of the classpath
  - The end of rt.jar
- Project at OpenJDK
  - Prototype available

Modularity

Modularity = Grouping
+ Dependence
+ Versioning
+ Encapsulation
+ Optional modules
+ Module aliases

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

com.foo.Main

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

com.foo.Main

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

com.foo.Main

com.foo.Other

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

```
com.foo.Main
com.foo.Other
```

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

```
// com/foo/ui/Shell.java

package com.foo.ui;
import java.io.*;
import java.util.*;
public class Shell {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

```
com.foo.Main
com.foo.Other
com.foo.ui.Shell
```

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

```
// com/foo/ui/Shell.java

package com.foo.ui;
import java.io.*;
import java.util.*;
public class Shell {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```

```
com.foo.Main
com.foo.Other
com.foo.ui.Shell
```

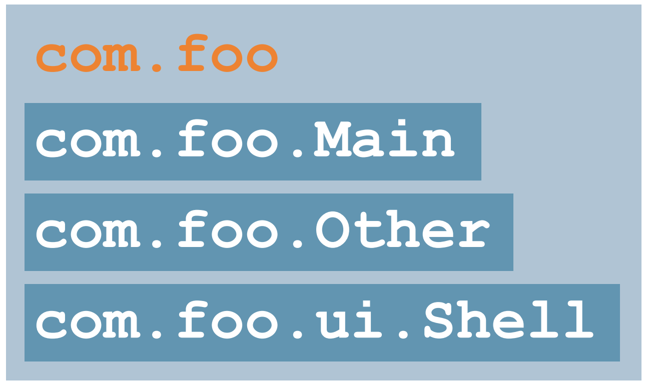
```
// module-info.java
module com.foo { }
```

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

```
// com/foo/ui/Shell.java

package com.foo.ui;
import java.io.*;
import java.util.*;
public class Shell {
    ...
}
```

Grouping

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
    ...
}
```



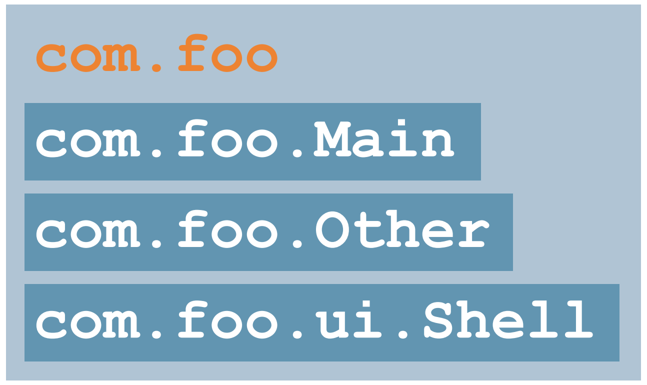
```
// module-info.java
module com.foo { }
```

```
// com/foo/Other.java

package com.foo;
import java.io.*;
import java.util.*;
public class Other {
    ...
}
```

```
// com/foo/ui/Shell.java

package com.foo.ui;
import java.io.*;
import java.util.*;
public class Shell {
    ...
}
```

Entry Point

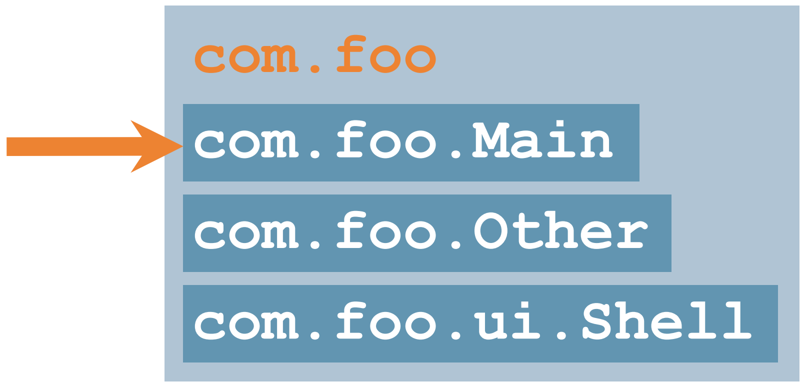
```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
  public static void main (...)
}
```



```
// module-info.java
module com.foo {
  class com.foo.Main;
}
```

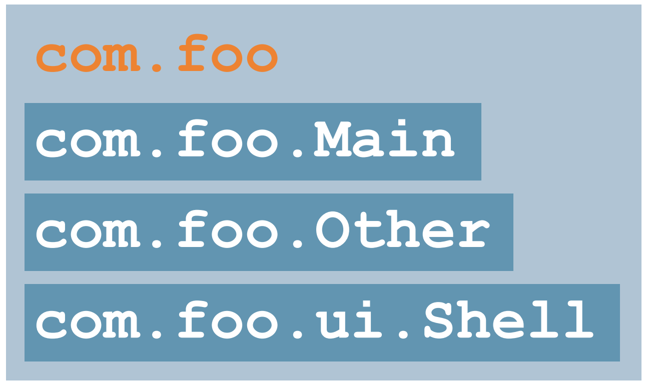
Entry Point

```
// com/foo/Main.java

package com.foo;

import java.io.*;
import java.util.*;

public class Main {
  public static void main (...)
}
```



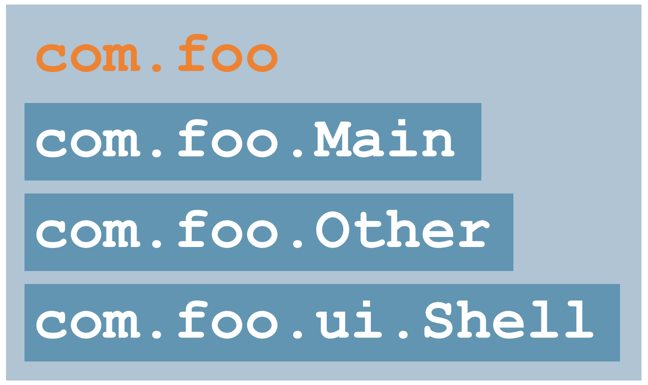
```
// module-info.java
module com.foo {
  class com.foo.Main;
}
```

External Libraries



```
// com/foo/ui/Shell.java
package com.foo.ui;
import java.io.*;
import java.util.*;

import org.bar.lib.*;
import edu.baz.util.*;


public class Shell {
    ...
}
```

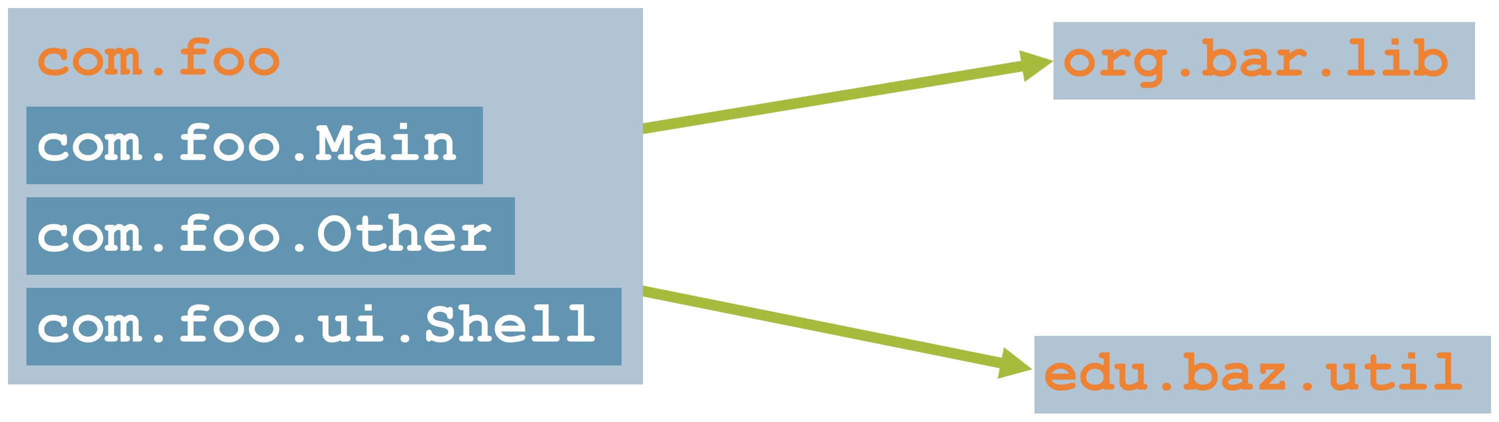
External Libraries



```
// com/foo/ui/Shell.java
package com.foo.ui;
import java.io.*;
import java.util.*;

import org.bar.lib.*;
import edu.baz.util.*;

public class Shell {
    ...
}
```

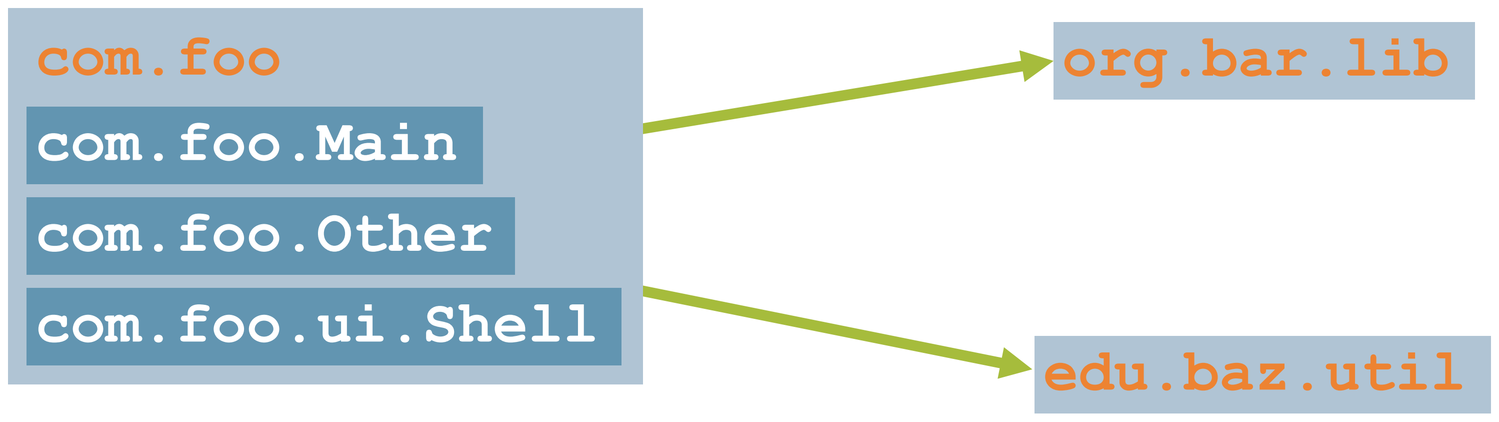
```
$ java -cp app.jar:bar.jar:baz.jar
```

External Libraries



```
// com/foo/ui/Shell.java
package com.foo.ui;
import java.io.*;
import java.util.*;

import org.bar.lib.*;
import edu.baz.util.*;

public class Shell {
    ...
}
```

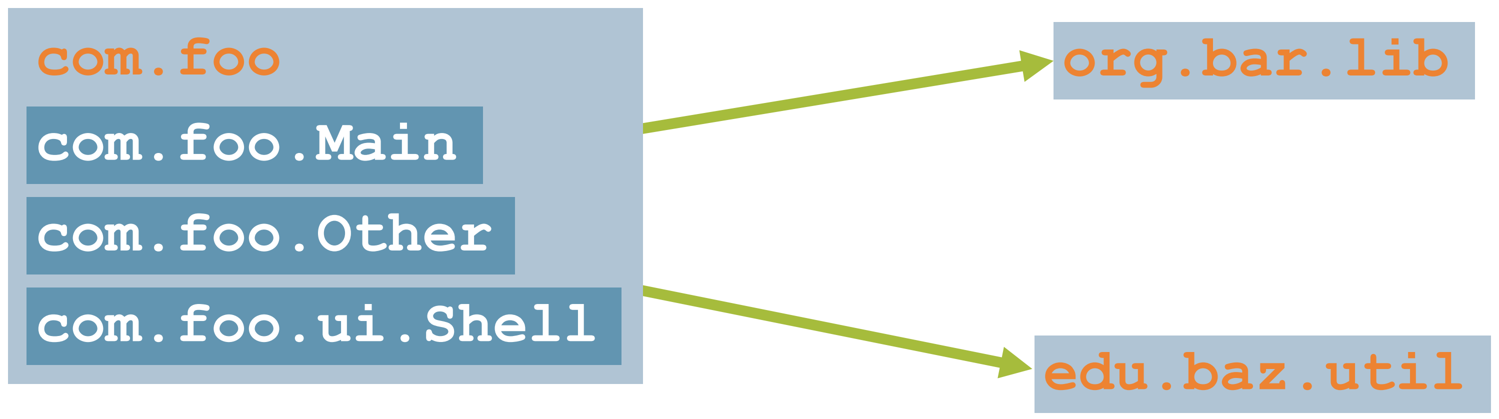
External Libraries



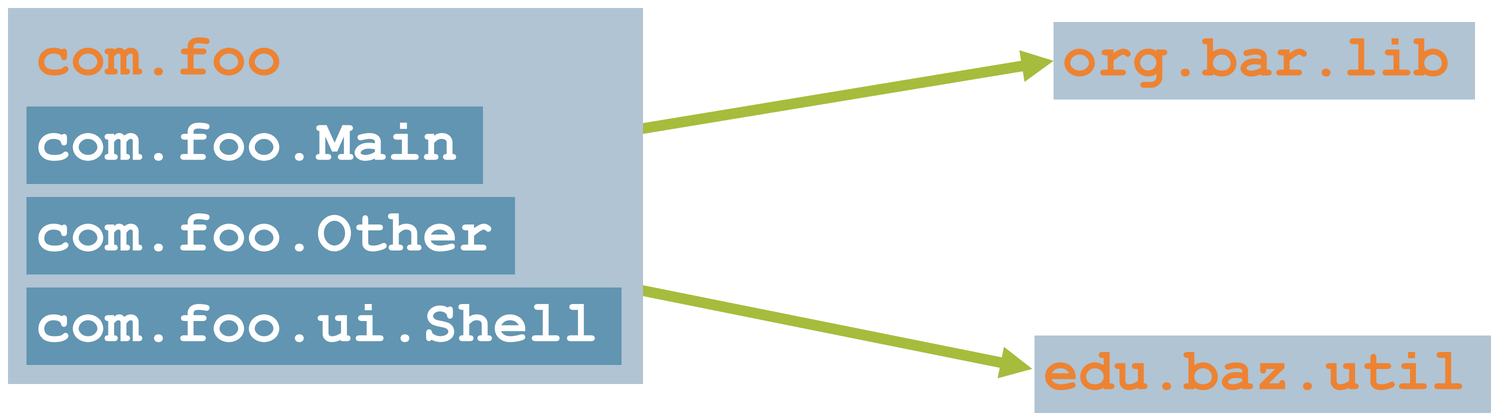
```
// com/foo/ui/Shell.java
package com.foo.ui;
import java.io.*;
import java.util.*;

import org.bar.lib.*;
import edu.baz.util.*;


public class Shell {
    ...
}
```

```
// module-info.java
module com.foo {
  class com.foo.Main;
  requires org.bar.lib;
  requires edu.baz.util;
}
```

Versioning


com.foo
com.foo.Main
com.foo.Other
com.foo.ui.Shell
org.bar.lib
edu.baz.util

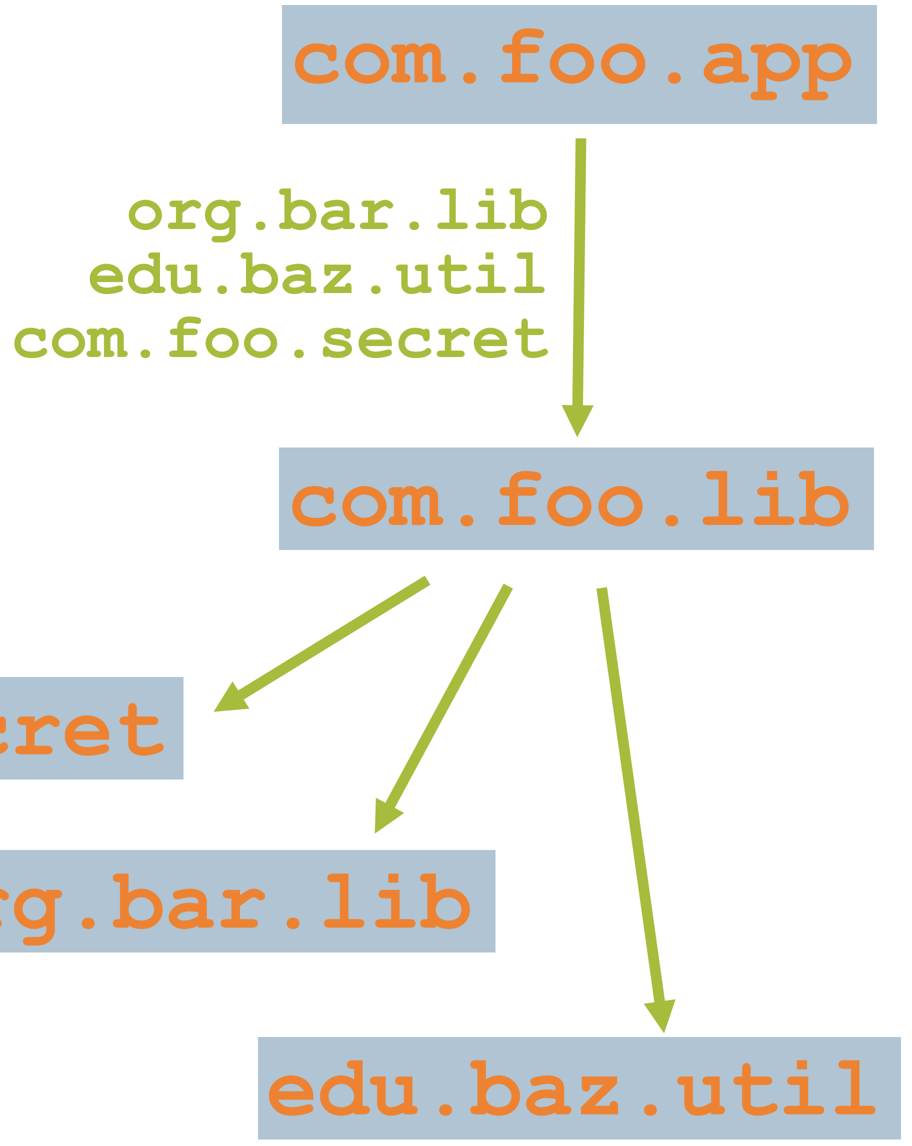
Versioning

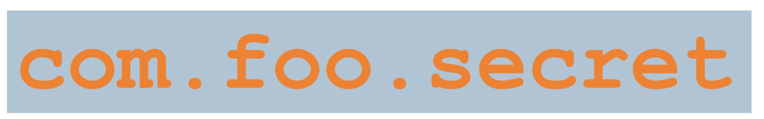


```
// module-info.java
module com.foo @ 1.0.0 {
  class com.foo.Main;
  requires org.bar.lib @ 2.1-alpha;
  requires edu.baz.util @ 5.2_11;
}
```

Encapsulation

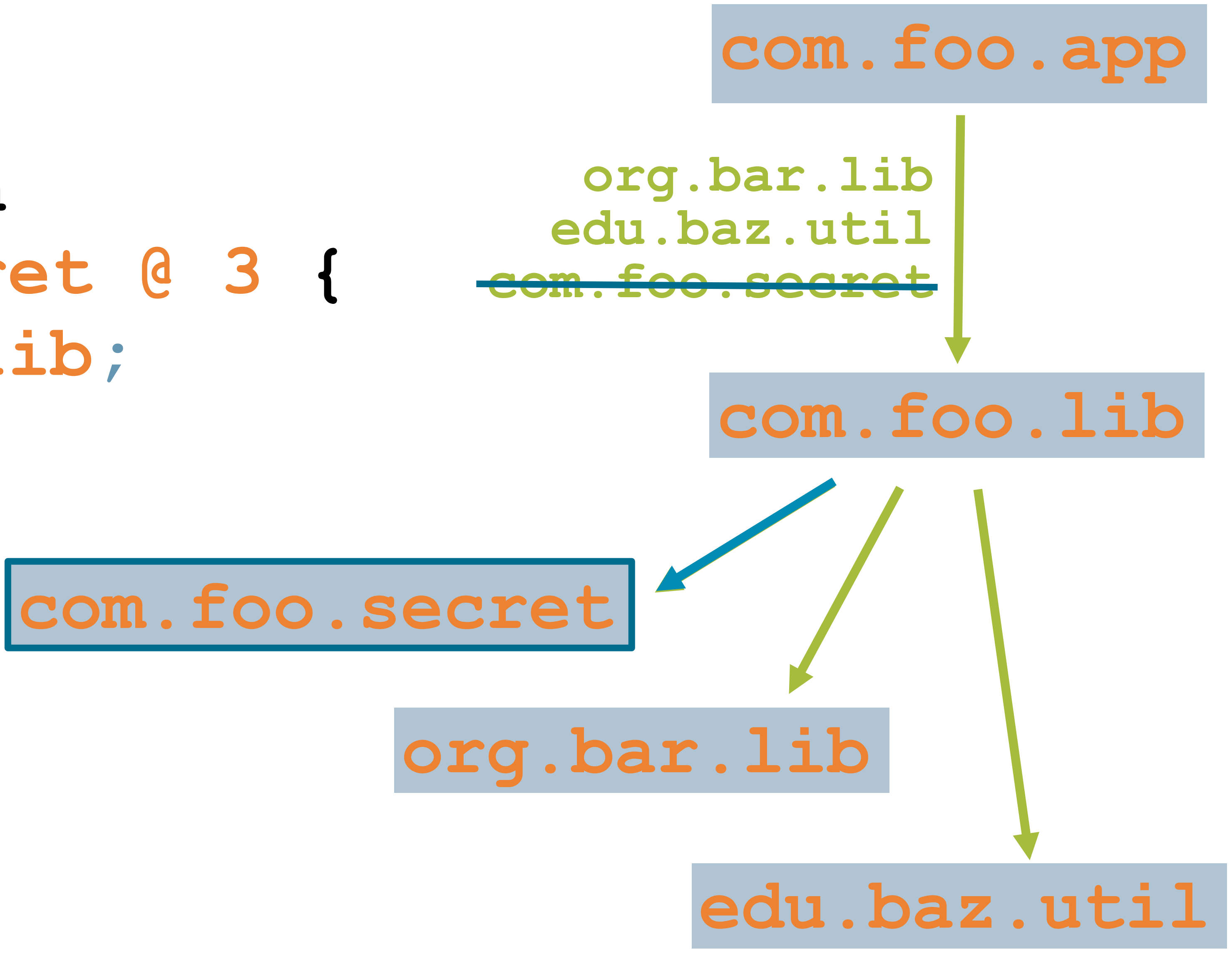
```
// module-info.java
module com.foo.secret @ 3 { }
```



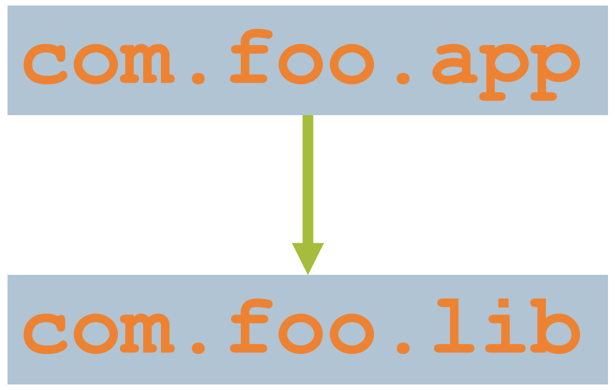


Encapsulation

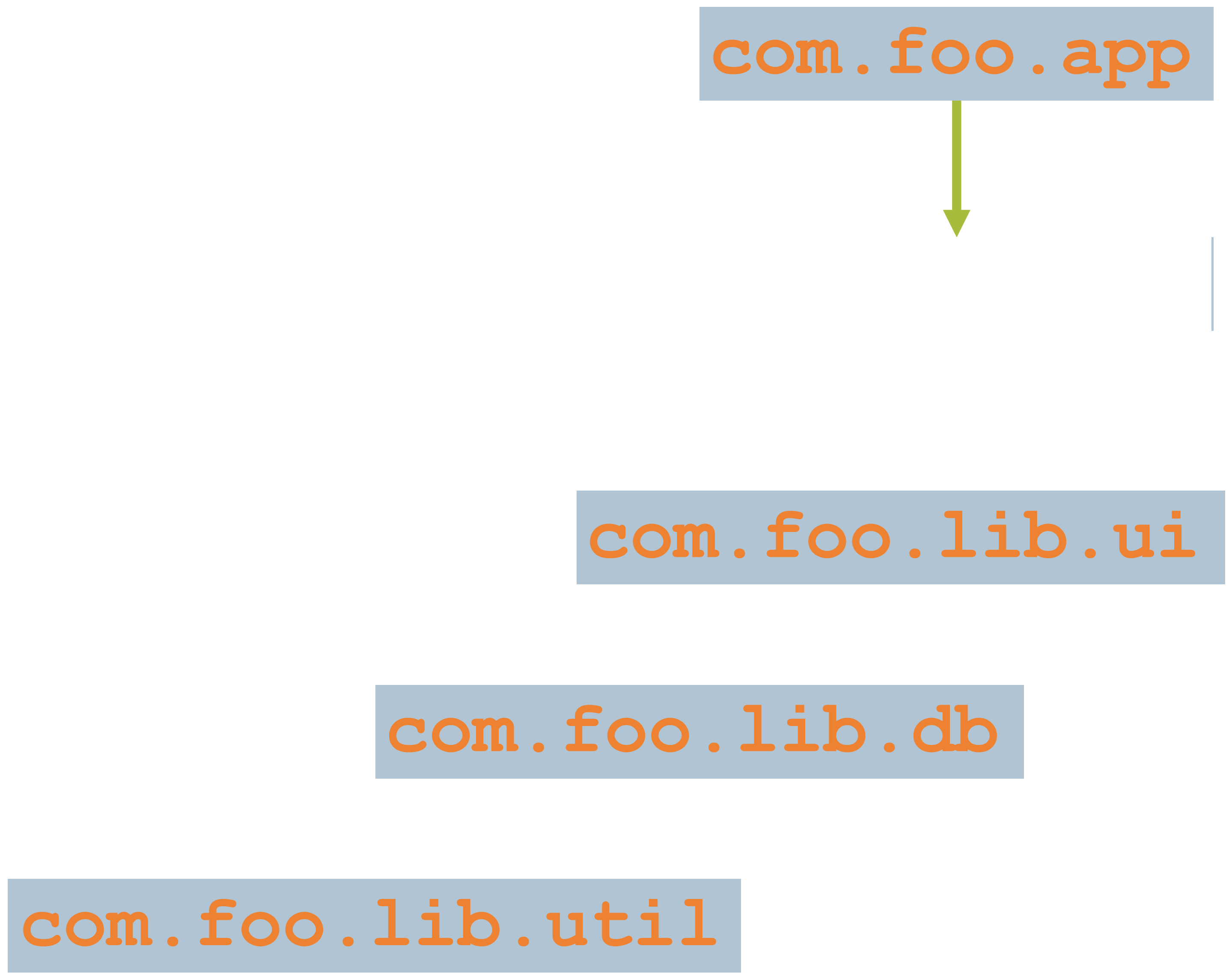
```
// module-info.java
module com.foo.secret @ 3 {
   permits com.foo.lib;
}
```



Splitting

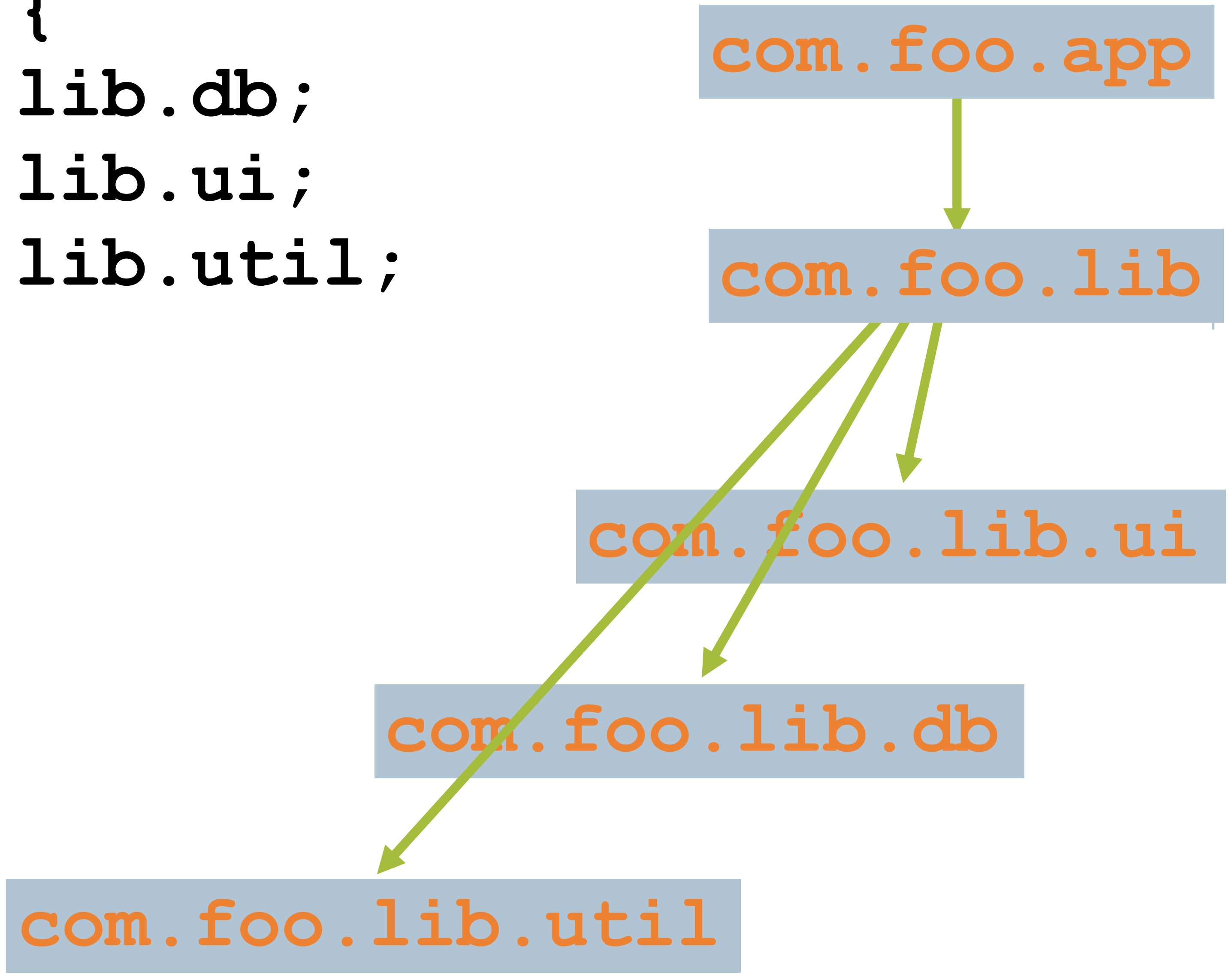


Splitting



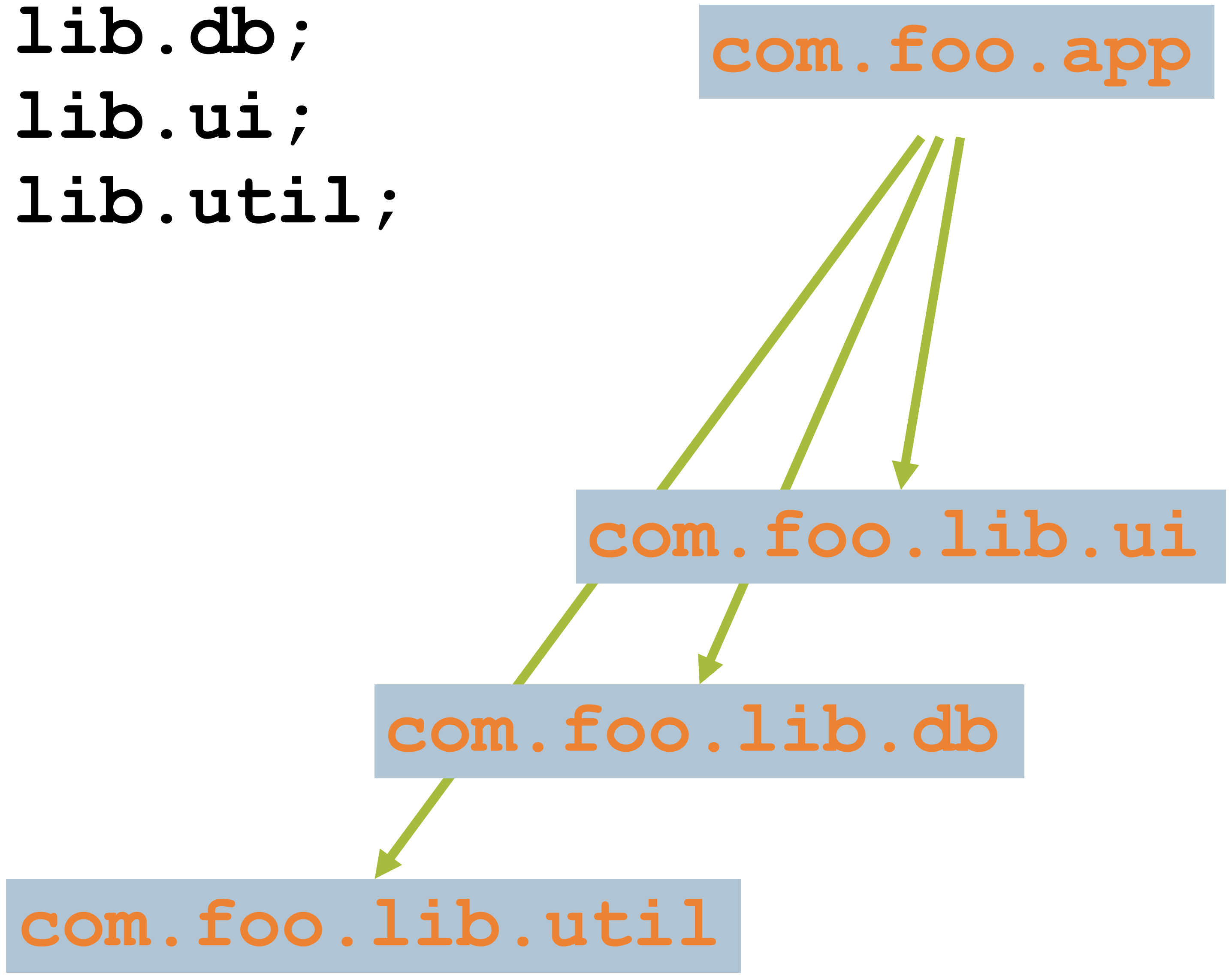
Splitting

```
module com.foo.lib {
   requires com.foo.lib.db;
   requires com.foo.lib.ui;
   requires com.foo.lib.util;
}
```



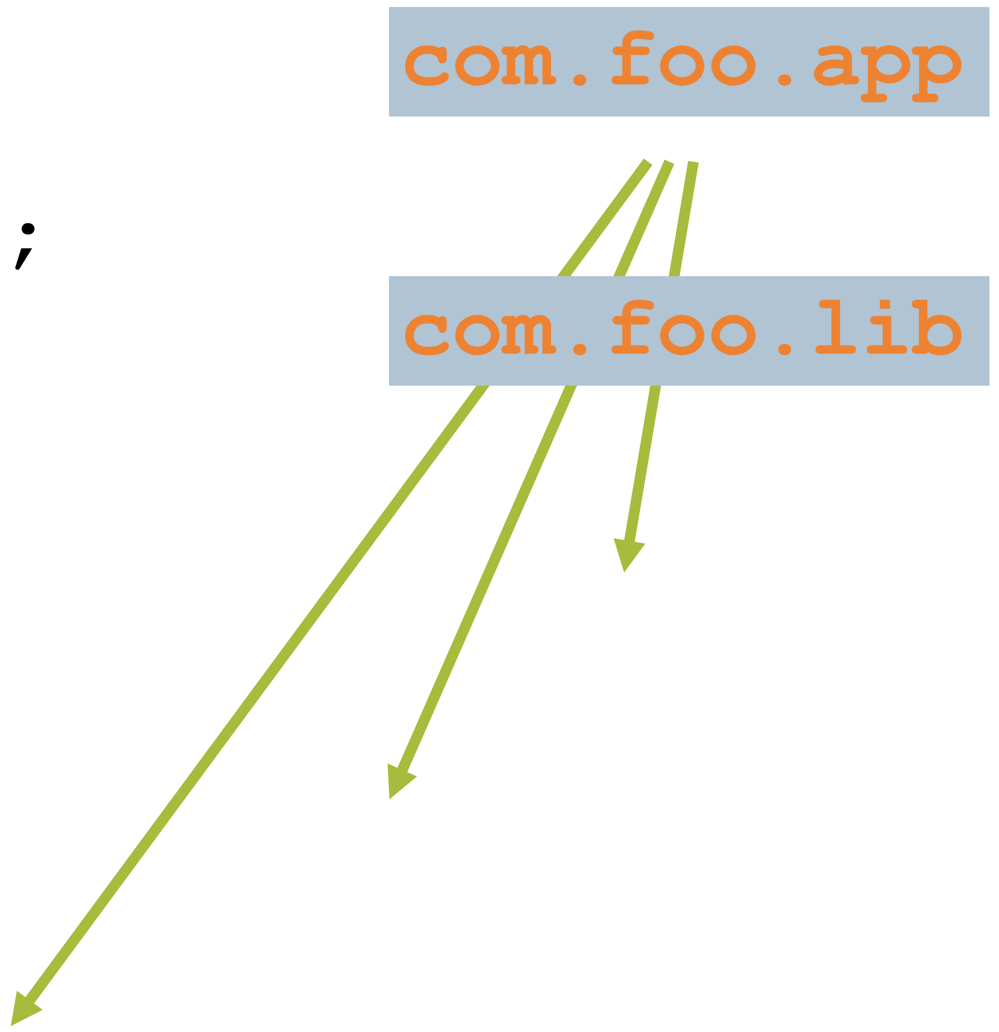
Aggregation

```
module com.foo.app {
   requires com.foo.lib.db;
   requires com.foo.lib.ui;
   requires com.foo.lib.util;
}
```



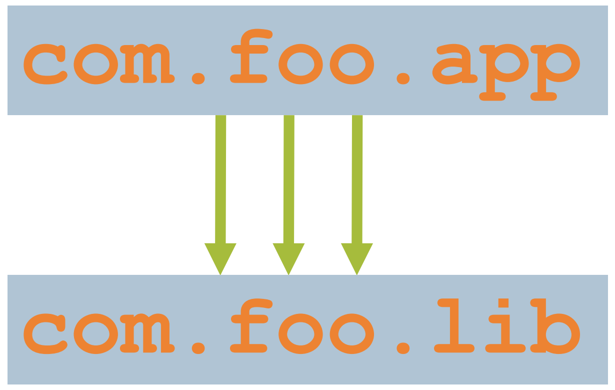
Aggregation

```
module com.foo.app {
   requires com.foo.lib.db;
   requires com.foo.lib.ui;
   requires com.foo.lib.util;
}
```



Aggregation

```
module com.foo.app {
  requires com.foo.lib.db;
  requires com.foo.lib.ui;
  requires com.foo.lib.util;
}
```



```
// Module aliases
module com.foo.lib {
  provides com.foo.lib.db;
  provides com.foo.lib.ui;
  provides com.foo.lib.util;
}
```

Packaging

Packaging

= Module files
+ Libraries
+ Repositories
+ Native packages

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$
```

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$ ls mods
com.foo.app/
com.foo.extra/
com.foo.lib/
$
```

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$ ls mods
com.foo.app/
com.foo.extra/
com.foo.lib/
$ ls -R mods/com.foo.app
mods/com.foo.app/com/foo/app/Main.class
mods/com.foo.app/com/foo/app/Other.class
mods/com.foo.app/com/foo/ui/Shell.class
mods/com.foo.app/module-info.class
$
```

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$ ls mods
com.foo.app/
com.foo.extra/
com.foo.lib/
$
```

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$ ls mods
com.foo.app/
com.foo.extra/
com.foo.lib/
$ jpkg -modulepath mods jmod \
  com.foo.app com.foo.extra com.foo.lib
$
```

Module Files

```
$ javac -modulepath mods src/com.foo.app/...
$ ls mods
com.foo.app/
com.foo.extra/
com.foo.lib/
$ jpkg -modulepath mods jmod \
  com.foo.app com.foo.extra com.foo.lib
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$
```

Libraries

```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$
```

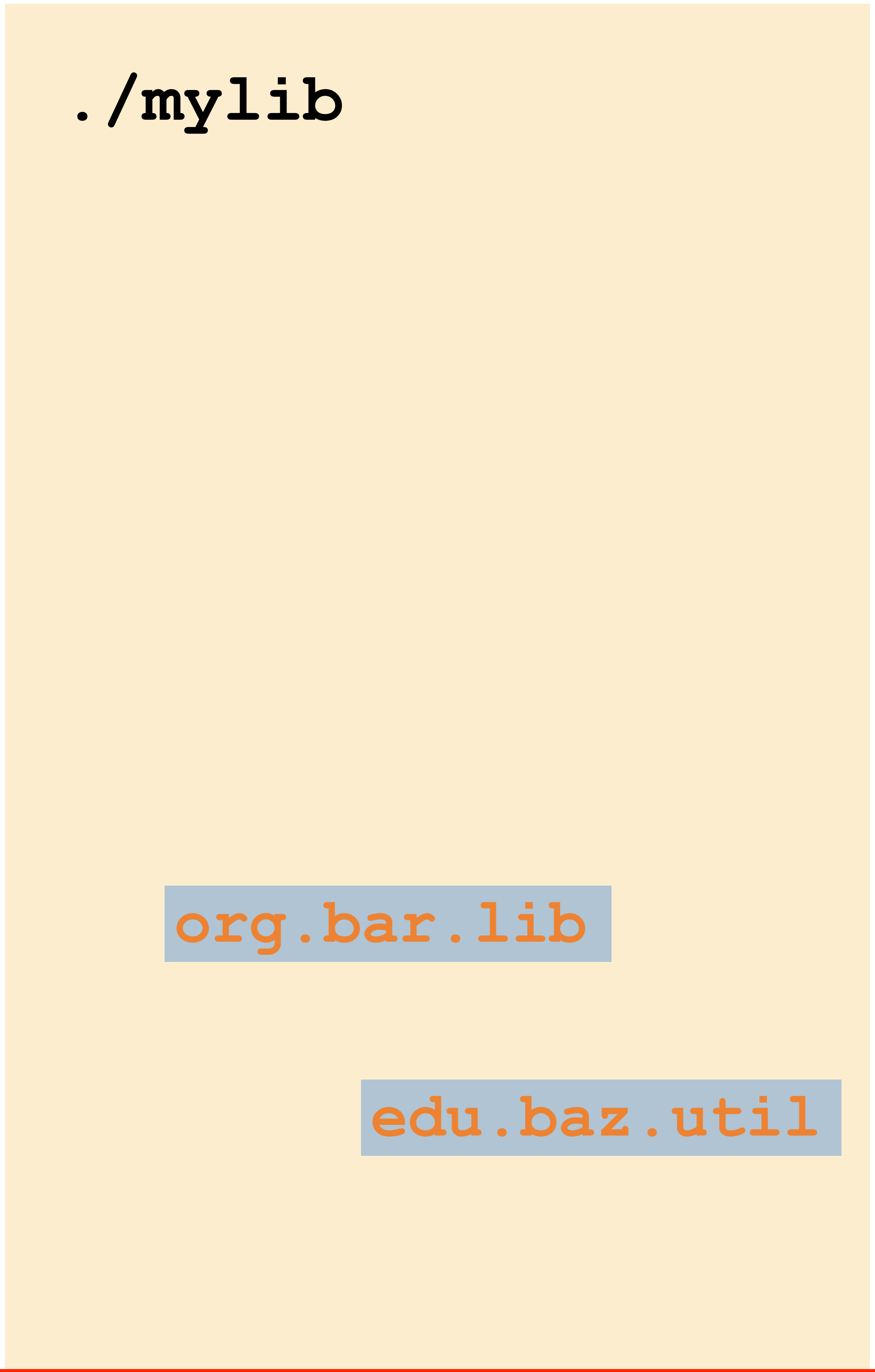
Libraries

```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$ jmod -L mylib create
$
```

```
./mylib
```

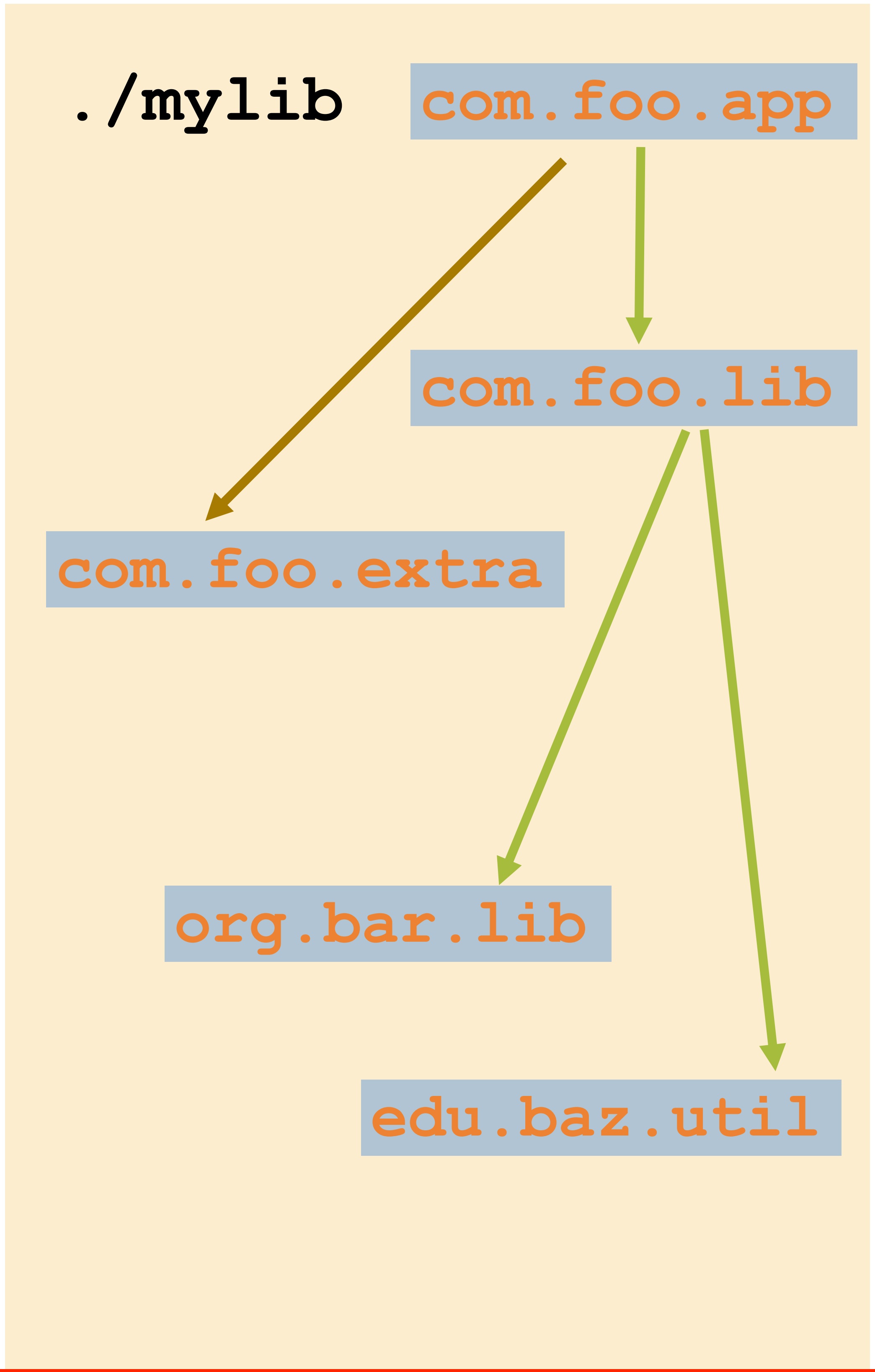
Libraries

```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$ jmod -L mylib create
$ jmod -L mylib install \
  $EXT/edu.baz.util@*.jmod \
  $EXT/org.bar.lib@*.jmod
$
```



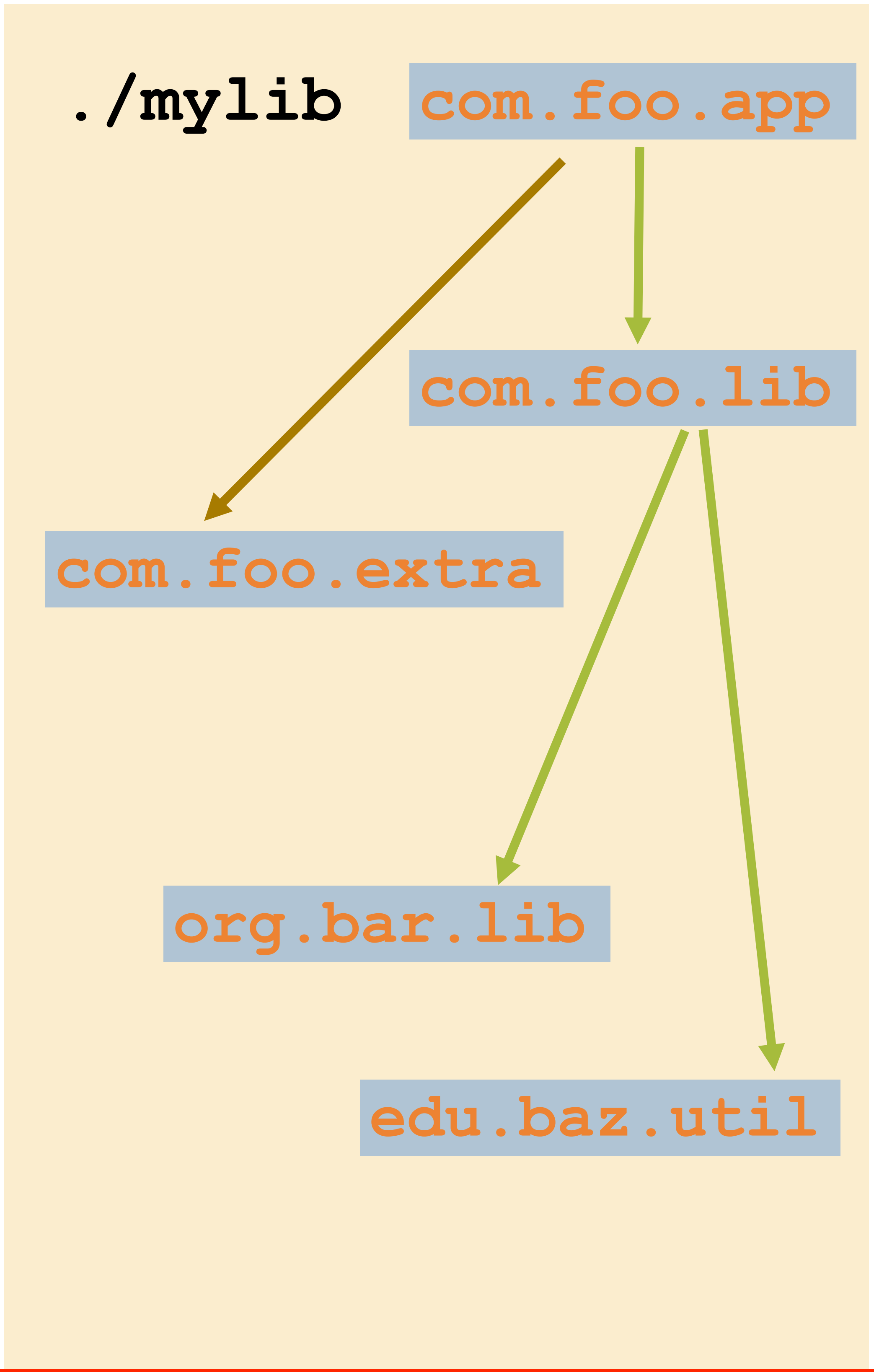
Libraries

```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$ jmod -L mylib create
$ jmod -L mylib install \
  $EXT/edu.baz.util@*.jmod \
  $EXT/org.bar.lib@*.jmod
$ jmod -L mylib install *.jmod
$
```



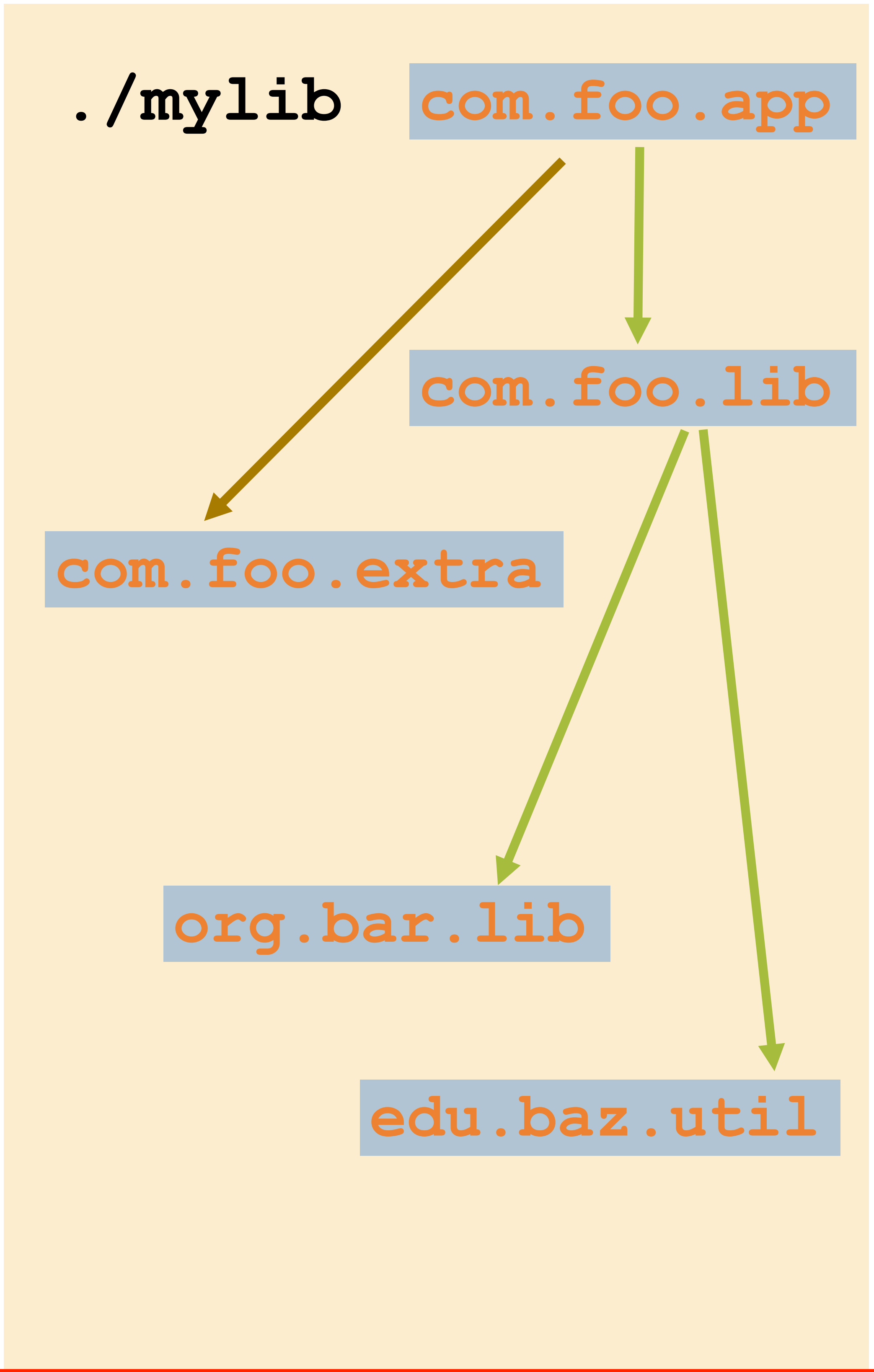
Libraries

```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$ jmod -L mylib create
$ jmod -L mylib install \
  $EXT/edu.baz.util@*.jmod \
  $EXT/org.bar.lib@*.jmod
$ jmod -L mylib install *.jmod
$ jmod -L mylib ls
com.foo.app @ 1.0.0
com.foo.extra @ 0.9a
com.foo.lib @ 1.0.2
edu.baz.util @ 5.2_11
org.bar.lib @ 2.1-alpha
$
```

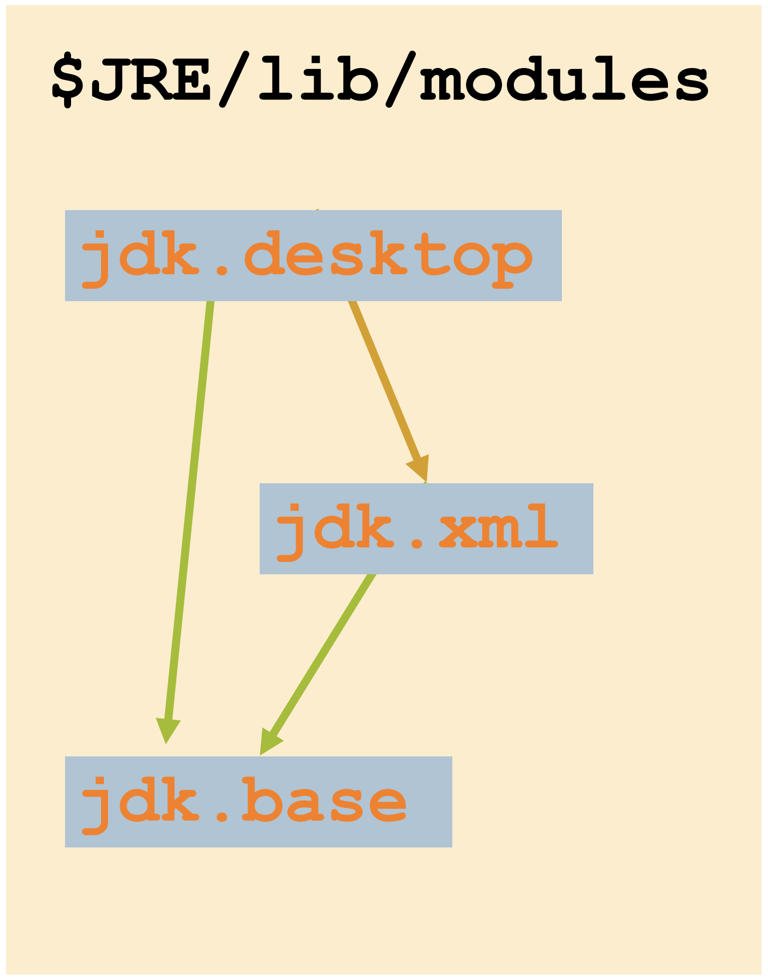


Libraries

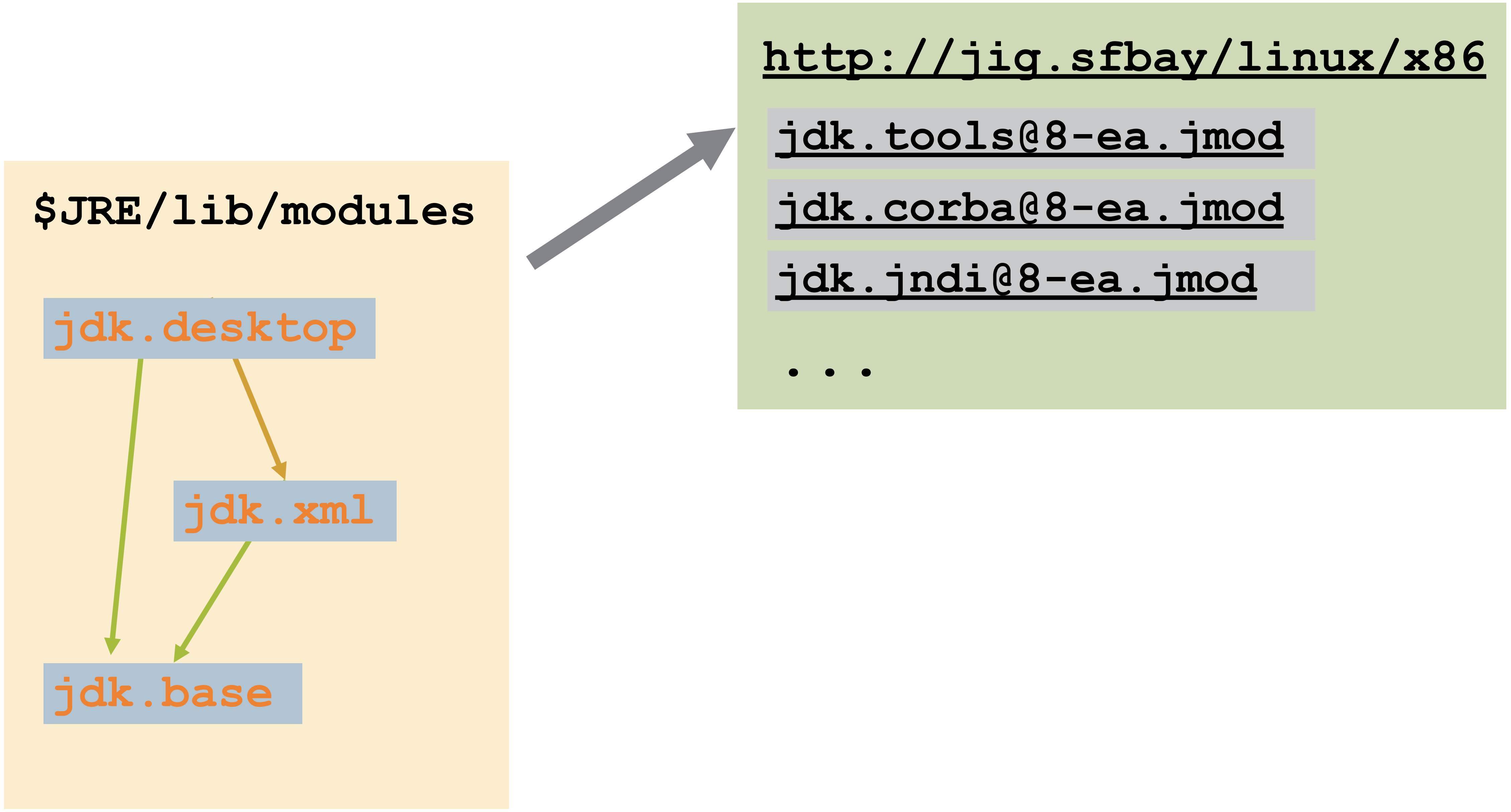
```
$ ls *.jmod
com.foo.app@1.0.0.jmod
com.foo.extra@0.9a.jmod
com.foo.lib@1.0.2.jmod
$ jmod -L mylib create
$ jmod -L mylib install \
  $EXT/edu.baz.util@*.jmod \
  $EXT/org.bar.lib@*.jmod
$ jmod -L mylib install *.jmod
$ jmod -L mylib ls
com.foo.app @ 1.0.0
com.foo.extra @ 0.9a
com.foo.lib @ 1.0.2
edu.baz.util @ 5.2_11
org.bar.lib @ 2.1-alpha
$ java -L mylib -m com.foo.app
Welcome to Foo, v1.0.0 ...
```



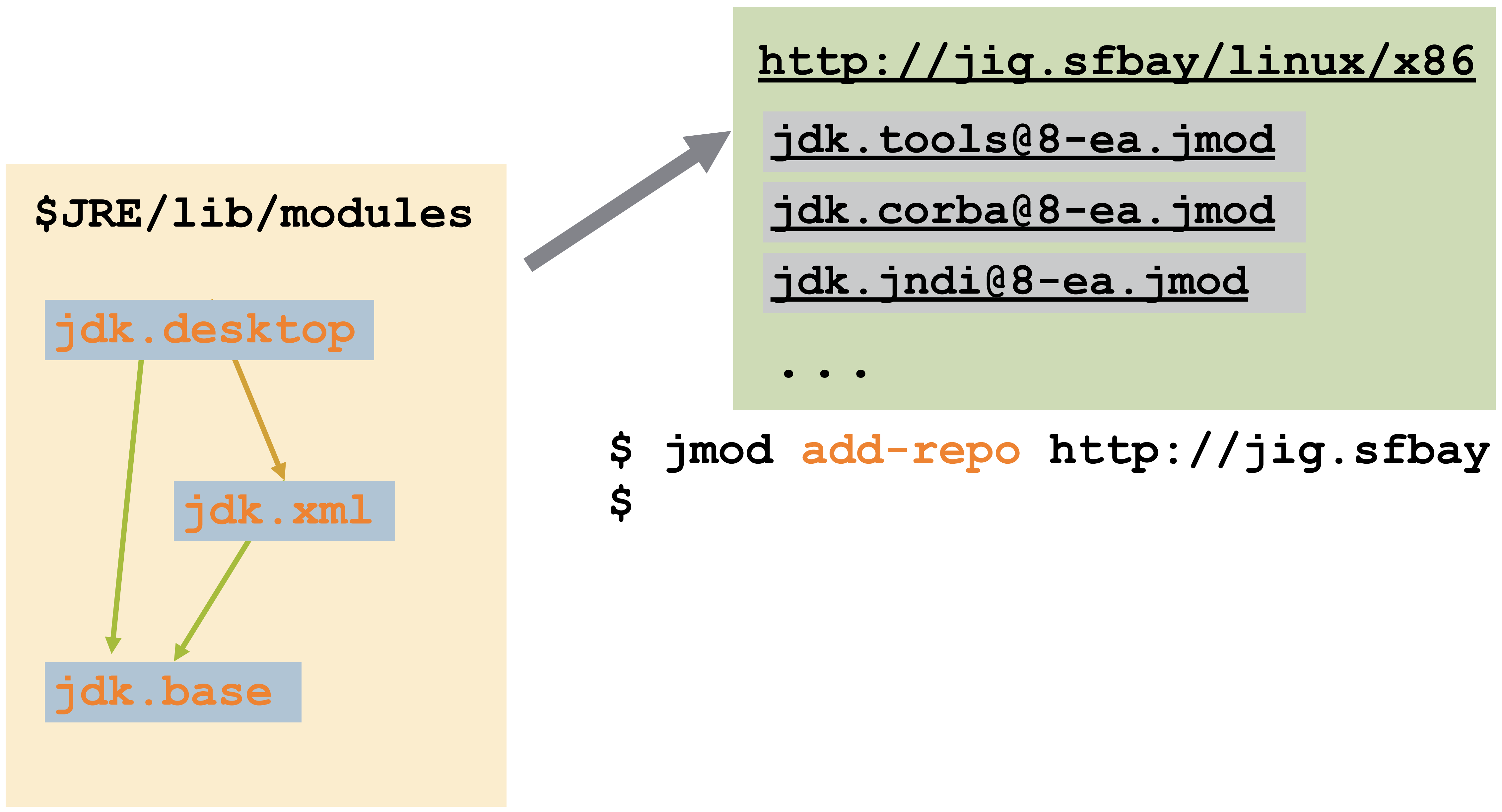
Repositories



Repositories

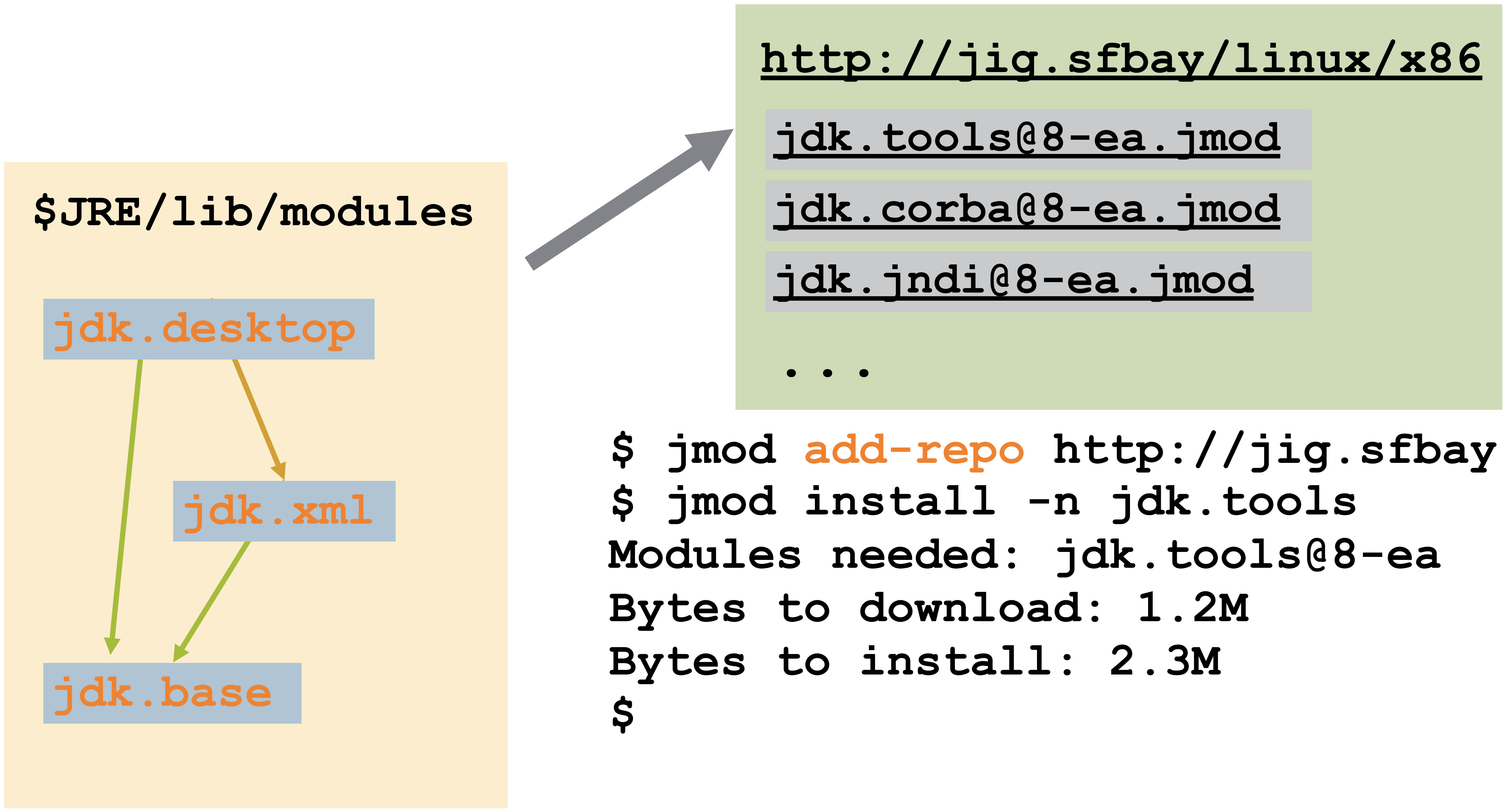


Repositories



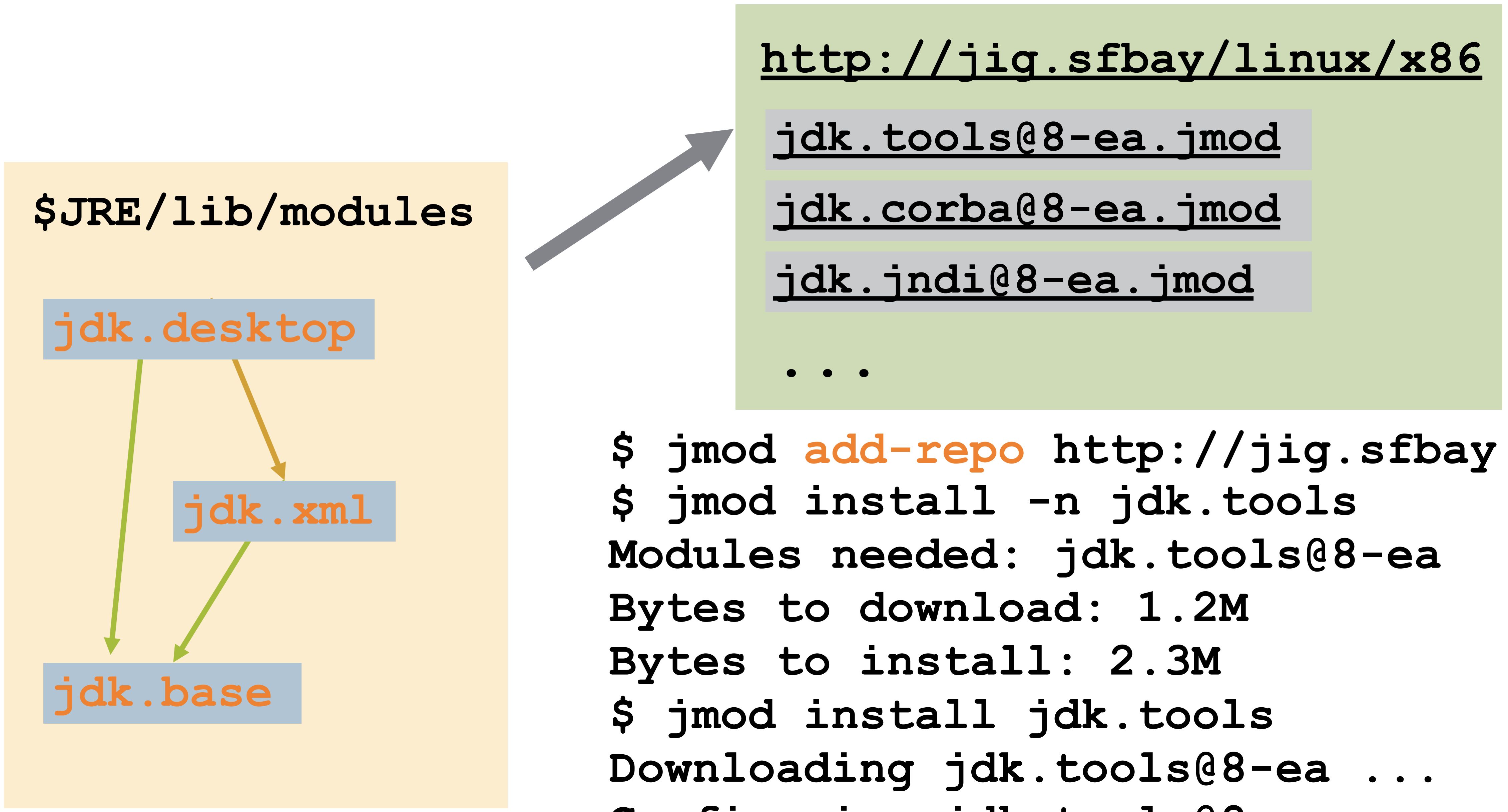
```
$ jmod add-repo http://jig.sfbay
$
```

Repositories



```
$ jmod add-repo http://jig.sfbay
$ jmod install -n jdk.tools
Modules needed: jdk.tools@8-ea
Bytes to download: 1.2M
Bytes to install: 2.3M
$
```

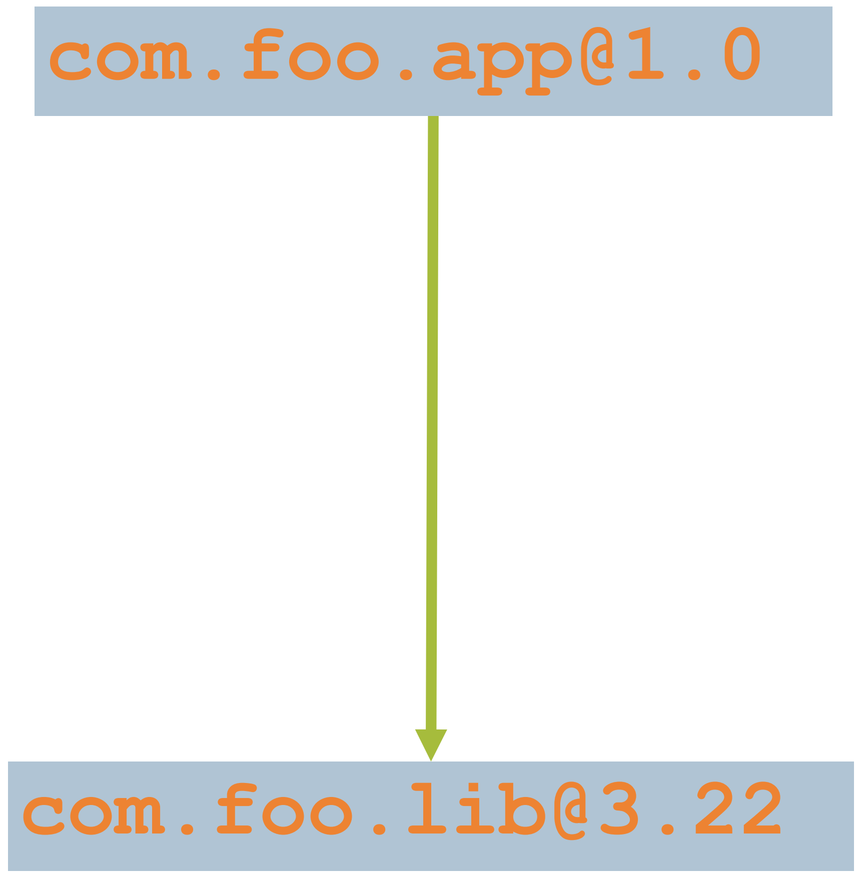
Repositories



```
$ jmod add-repo http://jig.sfbay
$ jmod install -n jdk.tools
Modules needed: jdk.tools@8-ea
Bytes to download: 1.2M
Bytes to install: 2.3M
$ jmod install jdk.tools
Downloading jdk.tools@8-ea ...
Configuring jdk.tools@8-ea ...
$
```

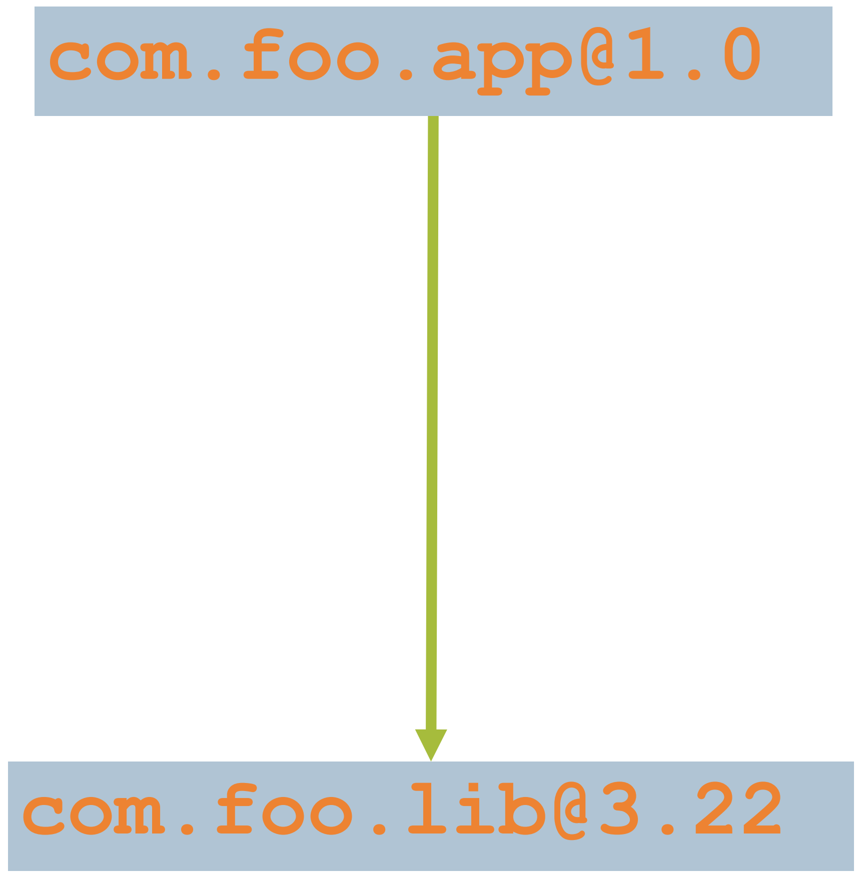
Native Packages

```
module com.foo.app @ 1.0 {
  requires com.foo.lib >= 3.0;
}
```



```
module com.foo.lib @ 3.22 { }
```

Native Packages



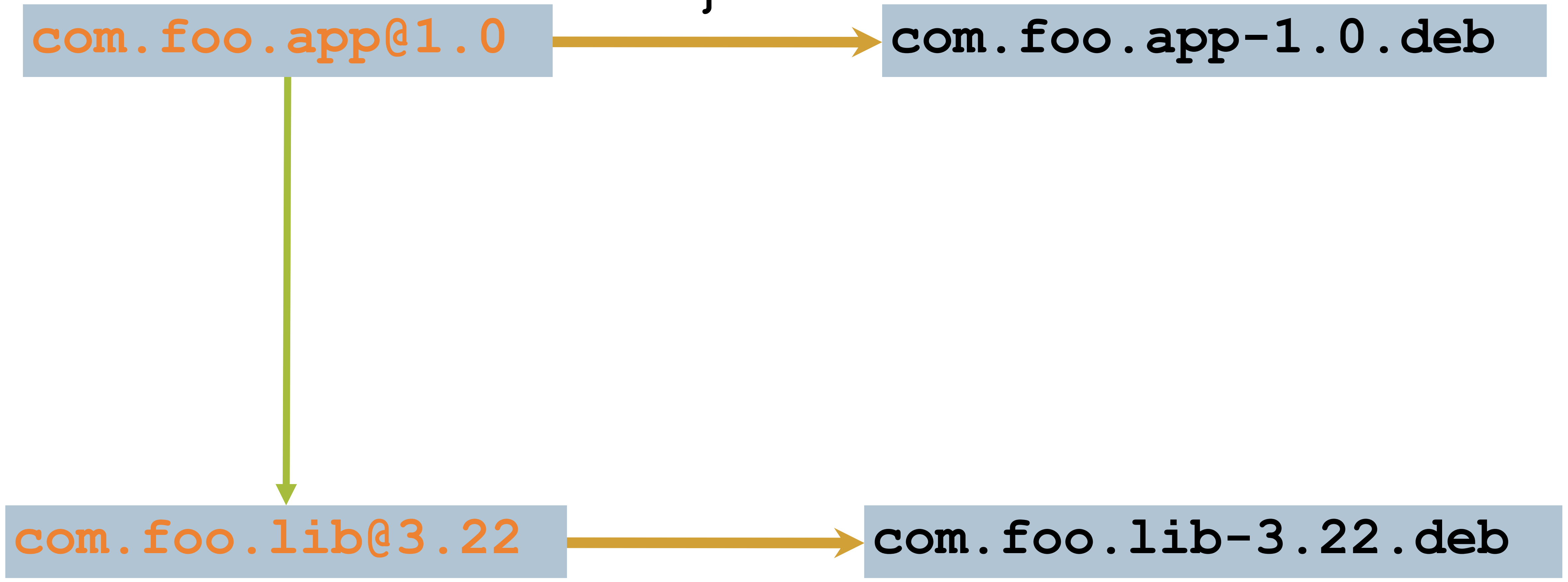
```
module com.foo.app @ 1.0 {
  requires com.foo.lib >= 3.0;
}
```

```
module com.foo.lib @ 3.22 { }
```

```
$ jpkg -m mods deb com.foo.app com.foo.lib
```

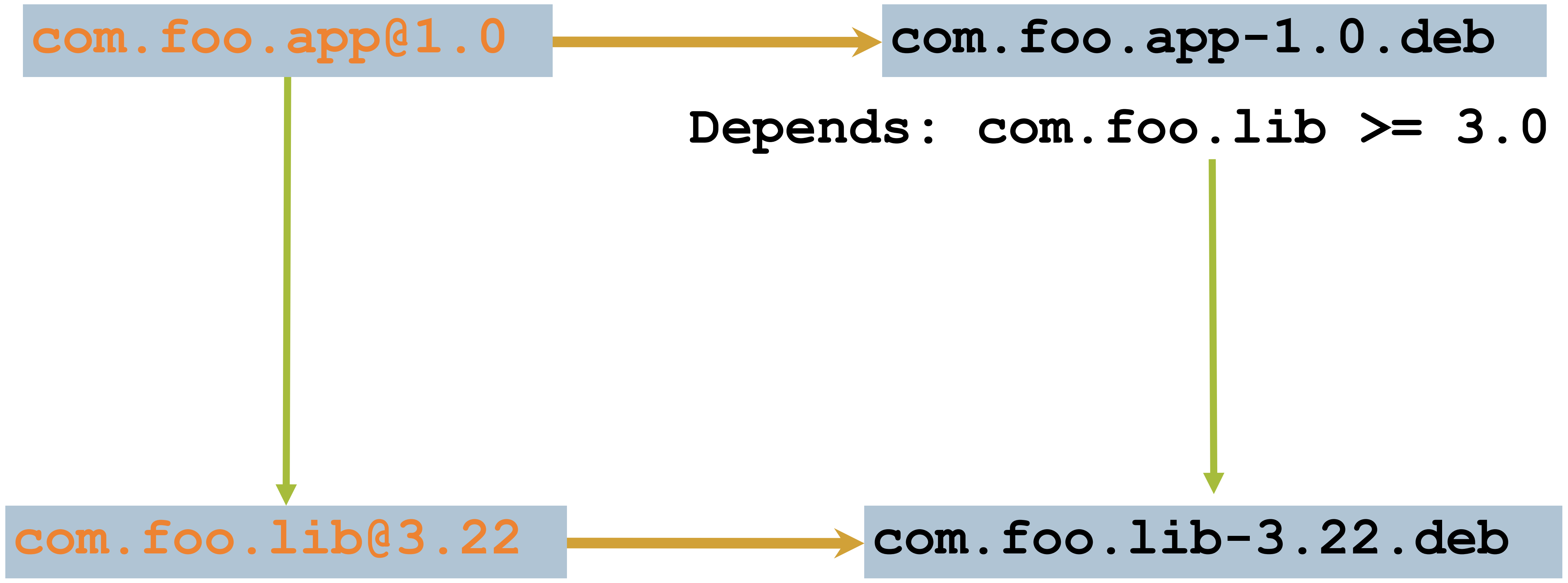
Native Packages

```
module com.foo.app @ 1.0 {
  requires com.foo.lib >= 3.0;
}
```



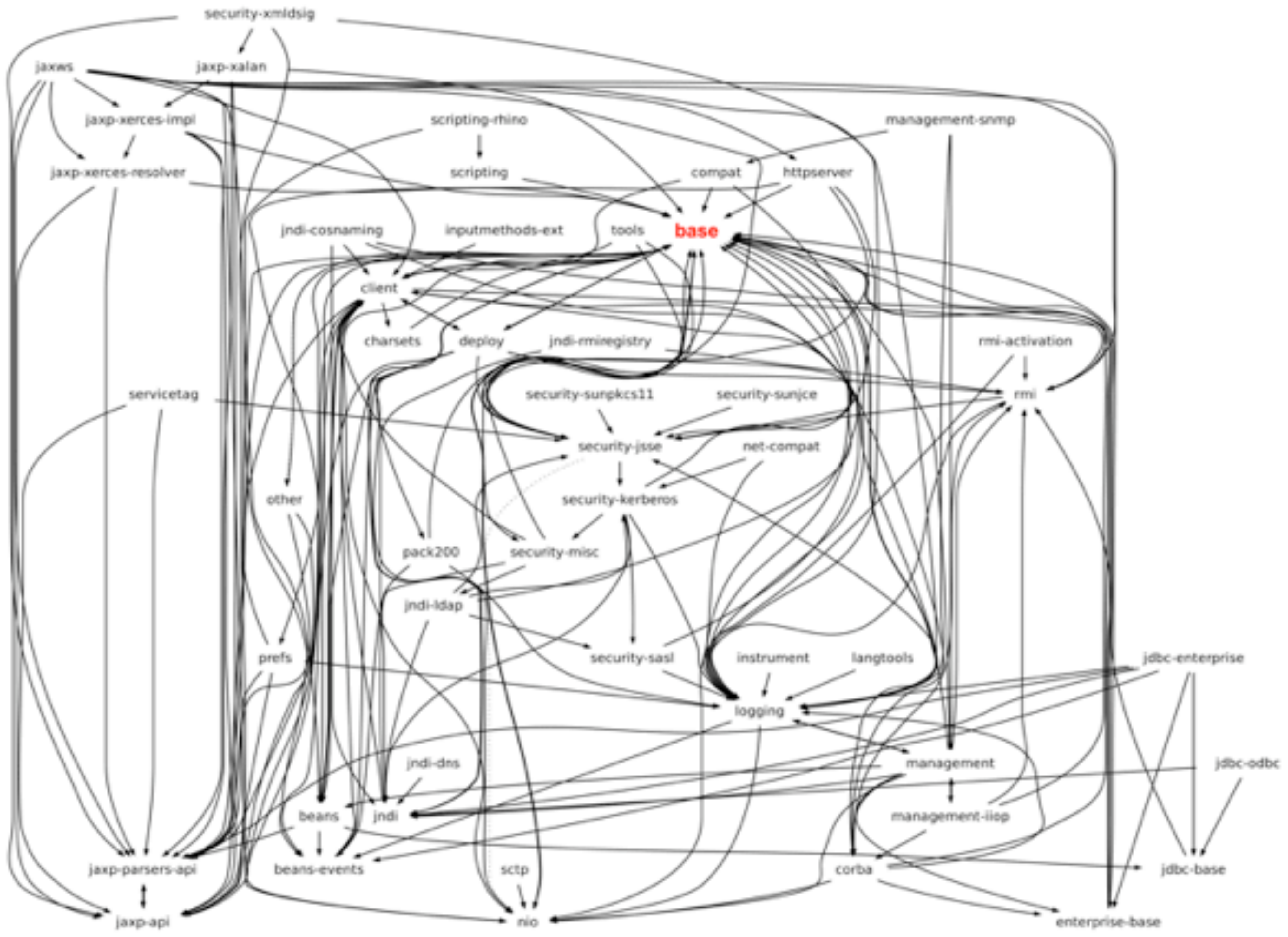
```
$ jpkg -m mods deb com.foo.app com.foo.lib
```

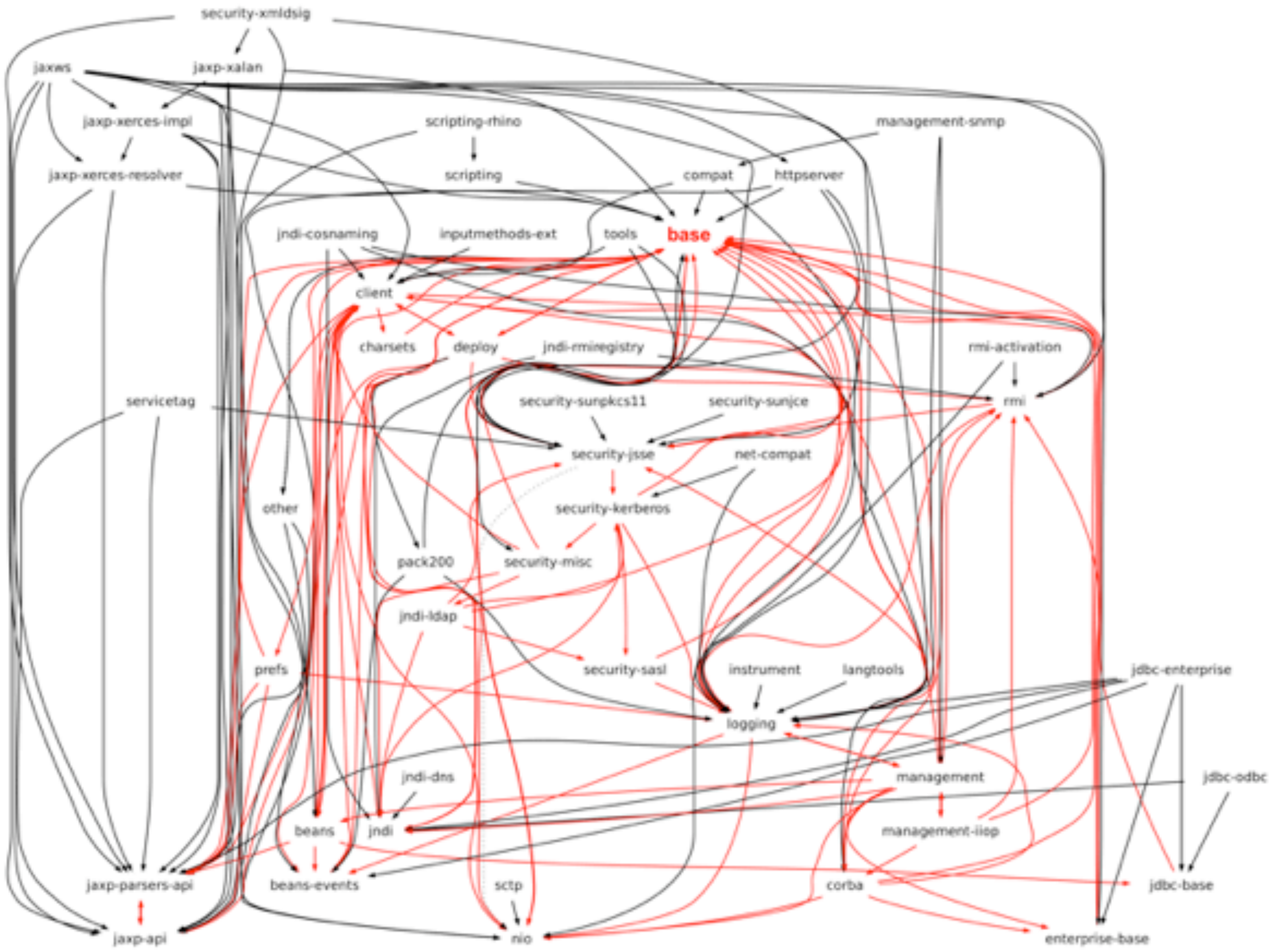
Native Packages

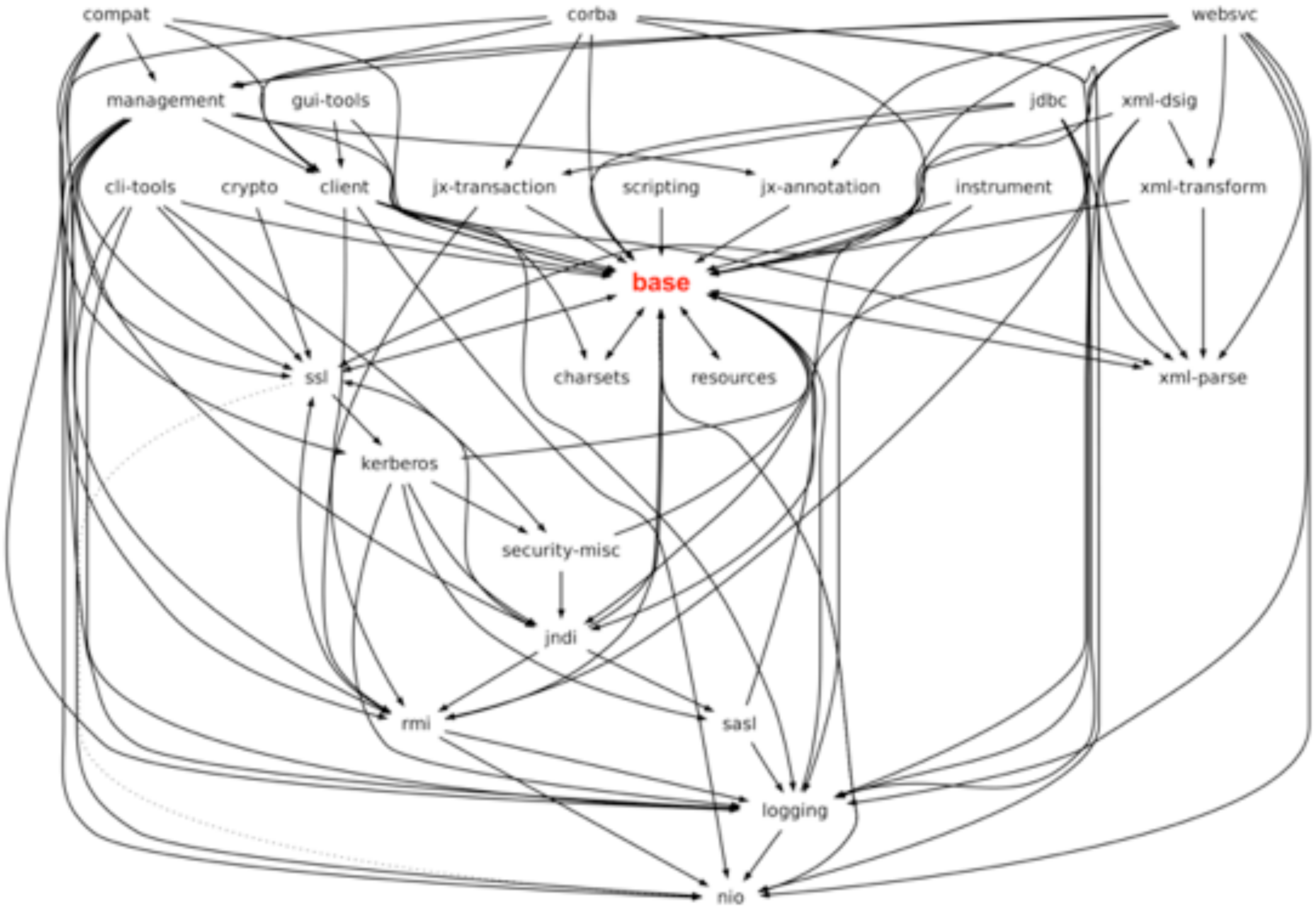


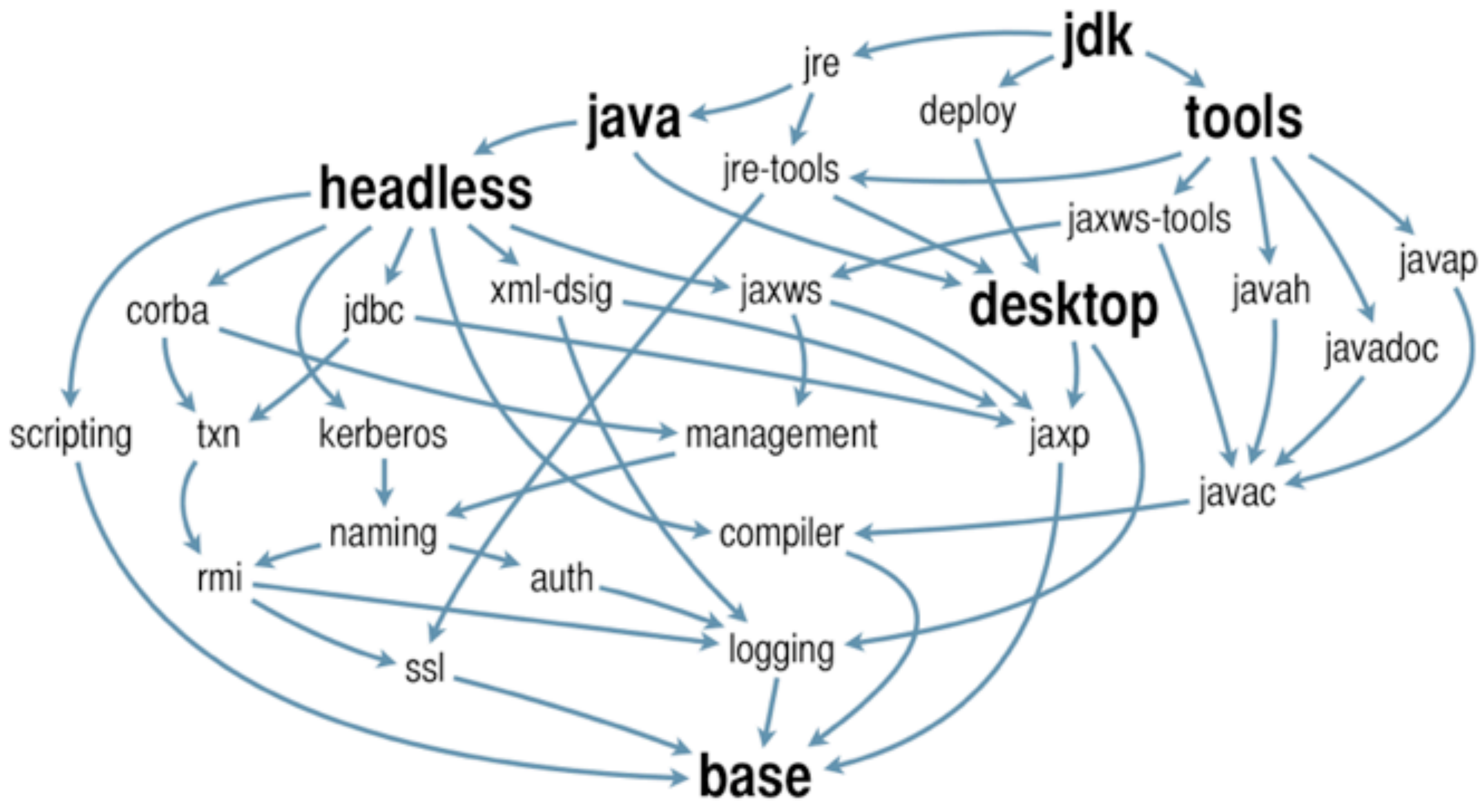
```
$ jpkg -m mods deb com.foo.app com.foo.lib
```

Slicing the JDK


security-xmldsig
jaxws
jaxp-xalan
jaxp-xerces-impl
scripting-rhino
management-snmp
jaxp-xerces-resolver
scripting
compat
httpserver
jndi-cosnaming
inputmethods-ext
tools
base
client
charsets
deploy
jndi-rmiregistry
rmi-activation
servicetag
security-sunpkcs11
security-sunjce
rmi
security-jsse
net-compat
other
security-kerberos
pack200
security-misc
jndi-ldap
prefs
security-sasl
instrument
langtools
jdbc-enterprise
logging
jndi-dns
management
jdbc-odbc
beans
jndi
management-iiop
jaxp-parsers-api
beans-events
sctp
corba
jdbc-base
jaxp-api
nio
enterprise-base


security-xmldsig
jaxws
jaxp-xalan
jaxp-xerces-impl
scripting-rhino
management-snmp
jaxp-xerces-resolver
scripting
compat
httpserver
jndi-cosnaming
inputmethods-ext
tools
base
client
charsets
deploy
jndi-rmiregistry
rmi-activation
servicetag
security-sunpkcs11
security-sunjce
rmi
security-jsse
net-compat
other
security-kerberos
pack200
security-misc
jndi-ldap
prefs
security-sasl
instrument
langtools
jdbc-enterprise
logging
jndi-dns
management
jdbc-odbc
beans
jndi
management-iiop
jaxp-parsers-api
beans-events
sctp
corba
jdbc-base
jaxp-api
nio
enterprise-base


compat
corba
websvc
management
gui-tools
jdbc
xml-dsig
cli-tools
crypto
client
jx-transaction
scripting
jx-annotation
instrument
xml-transform
base
ssl
charsets
resources
xml-parse
kerberos
security-misc
jndi
rmi
sasl
logging
nio

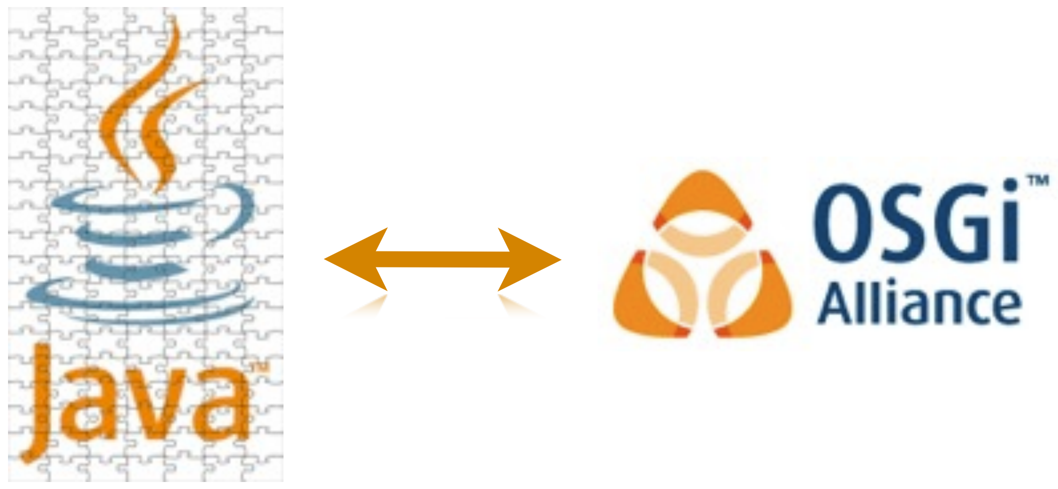

jdk
jre
deploy
tools
java
jre-tools
headless
jaxws-tools
javap
javah
desktop
xml-dsig
jaxws
corba
jdbc
javadoc
scripting
txn
kerberos
management
jaxp
javac
naming
compiler
rmi
auth
logging
ssl
base

Project Penrose

- Jigsaw / OSGi interop
  - Level 0 : toleration
  - Level 1 : interop of module info metadata
  - Level 2 : OSGi implementation exploit of Jigsaw modularity
  - Level 3+ : Full interop

Project Penrose

- Jigsaw / OSGi interop
  - Level 0 : toleration
  - Level 1 : interop of module info metadata
  - Level 2 : OSGi implementation exploit of Jigsaw modularity
  - Level 3+ : Full interop



Compatibility

- Existing applications will run unmodified
  - One can still use the classpath
  - But don't rely on rt.jar, tools.jar, non-standard classes, or internal JDK structure
- JAR files will not disappear overnight
  - Modular JAR files:
  - `% jar ufI foo-2.0.jar foo@2.0`
  - `% jmod install foo-2.0.jar`
- Maven integration in prototype
  - Take POM's as input to populate jigsaw library
  - Numerous challenges with varying pom.xml quality...

Resources

- Java 8 - http://openjdk.java.net.projects/jdk8

- http://oracle.com/javase
- http://openjdk.org

- http://openjdk.java.net/projects/jigsaw/
- http://openjdk.java.net/projects/jigsaw/doc/quickstart.html
- http://julien.ponge.info/notes/building-openjdk8-with-jigsaw/

Alexis Moussine-Pouchkine
@alexismp
http://alexismp.wordpress.com