Java SE Platform Update

CREATE
THE
FUTURE

Mark Reinhold (@mreinhold)
Chief Architect, Java Platform Group
Oracle

JCP EC Meeting
2015/01/14

ORACLE

Java SE 8 (JSR 337): Key Features

- Annotations on Java Types (JSR 308)
- Date & Time API (JSR 310)
- Lambda Expressions & Streams (JSR 335)
- Compact Profiles

ORACLE

Java SE 8: Additional Features

- Language
  - Access to Parameter Names at Runtime (JSR 269 MR, 337)
  - Add Javadoc to javax.tools (JSR 199 MR)
  - Annotations on Java Types (JSR 308)
  - Generalized Target-Type Inference (JSR 335)
  - Lambda Expressions & Virtual Extension Methods (JSR 269 MR, 335)
  - Repeating Annotations (JSR 269 MR, 337)
- Core Libraries
  - Base64 Encoding & Decoding
  - Bulk Data Operations for Collections (JSR 335)
  - Concurrency Updates
  - Date & Time API (JSR 310)
  - Enhance Core Libraries with Lambda (JSR 335)
  - JDBC 4.2 (JSR 114 MR, 221 MR)
  - Parallel Array Sorting
  - Statically-Linked JNI Libraries
- I18n
  - BCP 47 Locale Matching
  - Unicode 6.2
- Networking
  - HTTP URL Permissions
- Security
  - Configurable Secure Random-Number Generation
  - Enhance the Certificate Revocation-Checking API
  - Limited doPrivileged
  - NSA Suite B Cryptographic Algorithms
  - TLS Server Name Indication (SNI) Extension
- XML
  - Restrict Fetching of External XML Resources (JSR 206 MR)
- Platform
  - Compact Profiles (JSR 3 MR, 160 MR, 337)
  - Prepare for Modularization (JSR 173 MR, 206 MR, 337)

Java SE 8: Reception

Java SE 8: Reception

The First Word on Tech
INFOWORLD TECH WATCH

MARCH 18, 2014

Java 8 officially arrives at last

Oracle's long-anticipated update includes addition of lambda expressions and shift to functional programming language

By Paul Krill | InfoWorld Follow @pjkrill

eclipse

GETTING STARTED MEMBERS PROJECTS MORE

HOME / ABOUT US / ECLIPSE FOUNDATION ANNOUNCES JAVA 8 SUPPORT

About Us
- Foundation
- Governance
- Legal Resources
- Contact Us

Eclipse Foundation Announces Java 8 Support

Ottawa, Canada – April 2, 2014 - The Eclipse top-level proj
support for Java™ 8 for Eclipse Kepler SR2 (Eclipse 4.3.2).
install the Java 8 support please visit our Java™ 8 support

ZEROTURNAROUND

ZeroTurnarour
Product Updates, Com
Fun

JRebel 5.6.0 Released – Java 8, Weld 2 and much more!

June 30, 2014 Adam Koblentz No comments

JRebel 5.6.0 is released and available for download!

The biggest new feature for JRebel 5.6.0 is official sup
supports reloading on all of the new language features
default methods, and streams. Be sure to get your fee
features. And by the way, you might want to check ou
get you started with Java 8: Java 8 Revealed and Java
also find some Practical Advice on Migrating to Java 8.

Forbes

COVER STORY
The World's Richest Doctor

America's Most Expensive Homes For Sale Right Now

Six Trends That W
Consumer Behavi
Year

Log in | Sign up

ForbesBrandVoice Connecting marketers to the Forbes audience. What is this?

TECH 3/26/2014 @ 12:00PM | 29,838 views

8 Reasons To Love Java 8

Caroline Kvitka , Oracle

+ Comment Now

OracleVoice
FOLLOW

f developers, is
ost significant
ava is thriving
l users.

DZone

devops LINKS REFCARD

ZONES AGILE BIG DATA CLOUD DEVOPS ECLIPSE INTEGRATION IOT JAVA MO

Alec Noller
Bio @DZone

Java 8 Released

03.19.2014 8280 views

Java 8 has been released! Surely you've heard quite a bit about it al
widely anticipated and available for early access for a while, but no
can get everything you need from Oracle:

- Installation Instructions
- Release Notes
- What's New in JDK 8
- Known Issues for JDK 8
- JDK ReadMe
- JRE ReadMe

There are a few pretty big changes to check out in Java 8 - Lambdas, in particular, are getting a lot of attention. Luckily, we've had quite a

Dr.Dobb's
THE WORLD OF SOFTWARE DEVELOPMENT

Sections ▼ Channels ▼

JVM LANGUAGES

Tweet Permalink

JetBrains IntelliJ IDEA 13.1 Includes Support For Java 8

By Adrian Bridgwater, March 20, 2014

ComputerSweden

Publicerad 2014-03-17 05:31

Massiv pepp för Java 8

ORACLE

Java SE 8: Adoption

- GitHub project builds on Java SE 8 (Travis CI)
  - August 2014: 2,380
  - January 2015: 4,982
- Many prominent projects will require SE 8 in near-future releases
  - Scala 2.12, Lucene 5.0, Hippo CMS 7.10, OpenNMS 1.14, Weld 3.0, …
- Typesafe survey (September 2014): Adoption faster than expected
  - A fifth of SE 8 adopters are already running it in production
  - A third are planning to deploy it shortly

Java SE 8: Maintenance Review (closes 2015/02/09) (*details*)

- Language & VM specification corrections
  - Update the class and interface initialization algorithm
  - Clarify overload resolution with respect to lambda expressions
- Debugging: Support static and default methods in JDI, JDWP, and JDB
  - Not finished in time for SE 8 Final Release
- Deprecation, in preparation for modularization
  - Extension Mechanism
  - Endorsed-Standards Override Mechanism

Java SE 9 (JSR TBD)

- Key features
  - Java Platform Module System (JSR 376)
  - Java SE Platform Modularization (JEPs 200, 220)
- Additional potential features (so far)
  - Small language cleanups (JEPs 211, 213)
  - Enhanced volatiles (JEP 193)
  - Java Read-Eval-Print Loop (REPL) (JEP 222)
  - Datagram Transport Layer Security (JEP 219)
  - HTTP 2 Client API (JEP 110)

Java Platform Module System (JSR 376)

"Approachable, yet scalable"

- Key features:
  - Reliable configuration
  - Secure encapsulation
- … which enable:
  - Escape from "JAR hell"
  - A modular, scalable platform
    - Both Java SE and, eventually, Java EE
  - Improved platform security and integrity
  - Improved performance

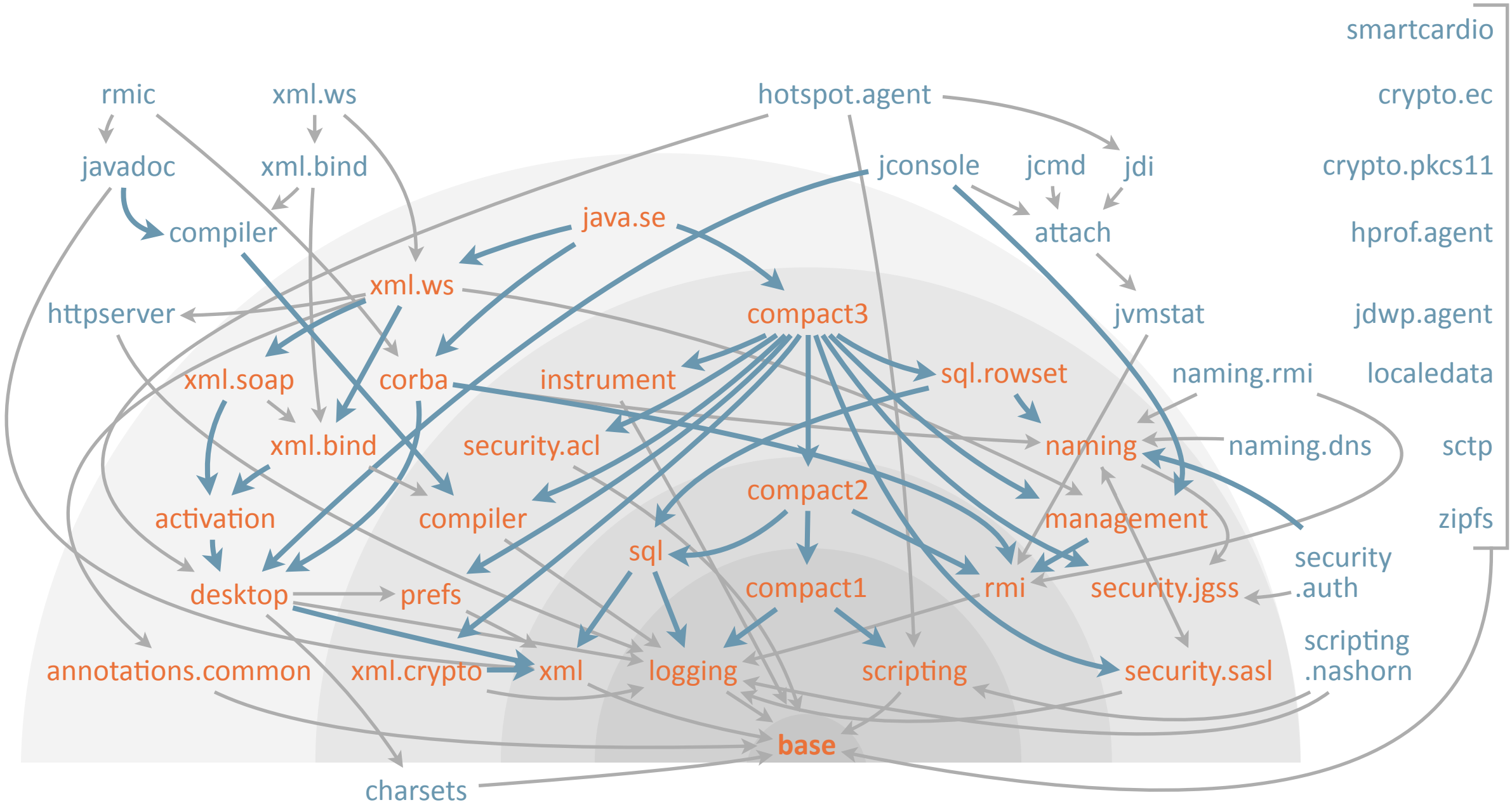
ORACLE

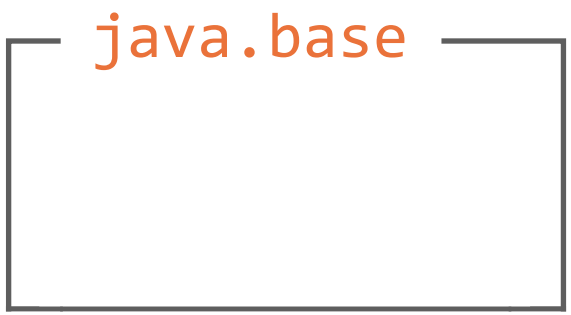
Java Platform Module System: Technology Overview

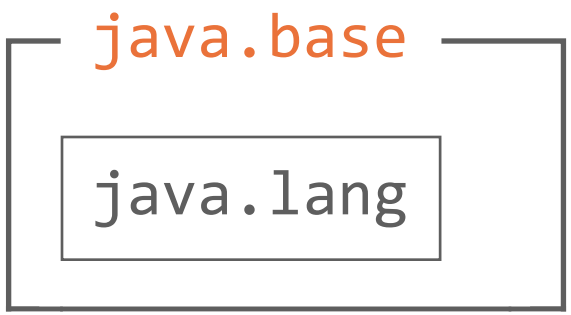
- *Module* = A named, self-describing collection of code and data
  - Declares which modules are *required* in order to compile and run it
  - Declares which of its API packages are *exported* for use by other modules, and which are not
  - Declares which service interfaces it *uses*, and which services it *provides*
- Run-time system responsibilities
  - Locate all required modules, given an initial module *(resolution)*
  - Ensure that modules do not interfere with each other
    - *E.g.*, in case two modules contain packages of the same name
  - Load code on behalf of the compiler and VM
  - Provide a means for other module systems (*e.g.*, OSGi) to locate and resolve modules
- Language/VM responsibilities
  - Prevent code from accessing types in non-exported packages

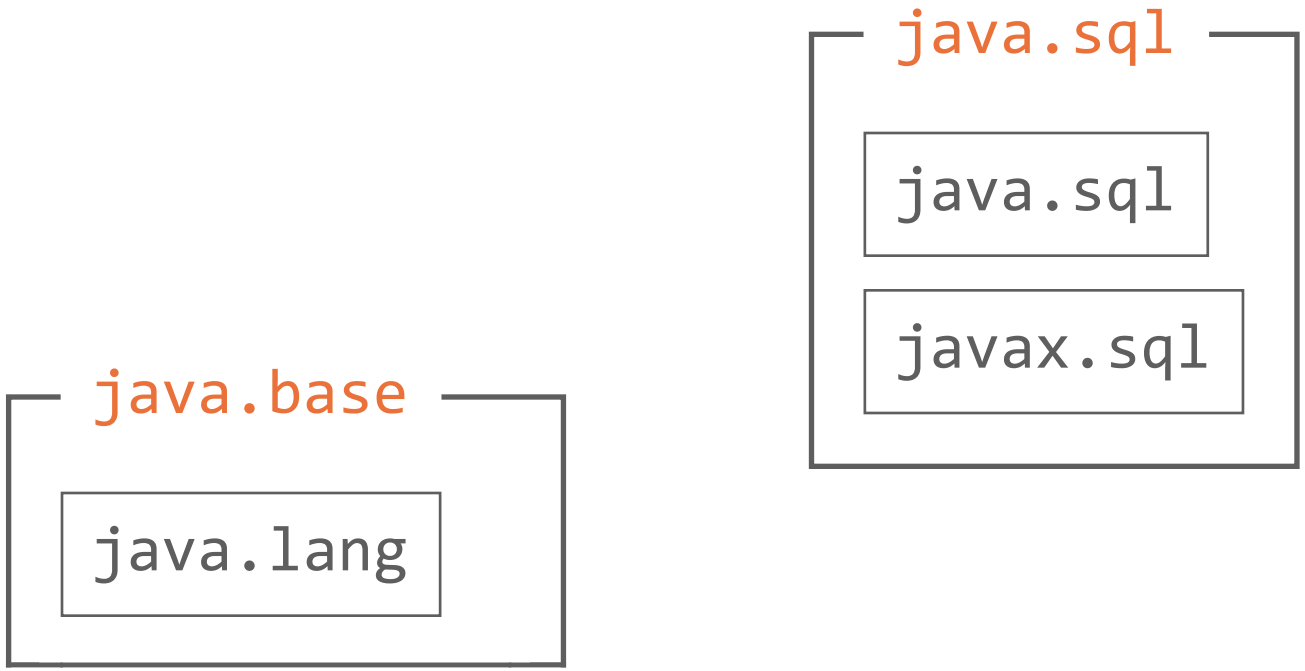
Java SE Platform Modularization (JEPs 200, 220)

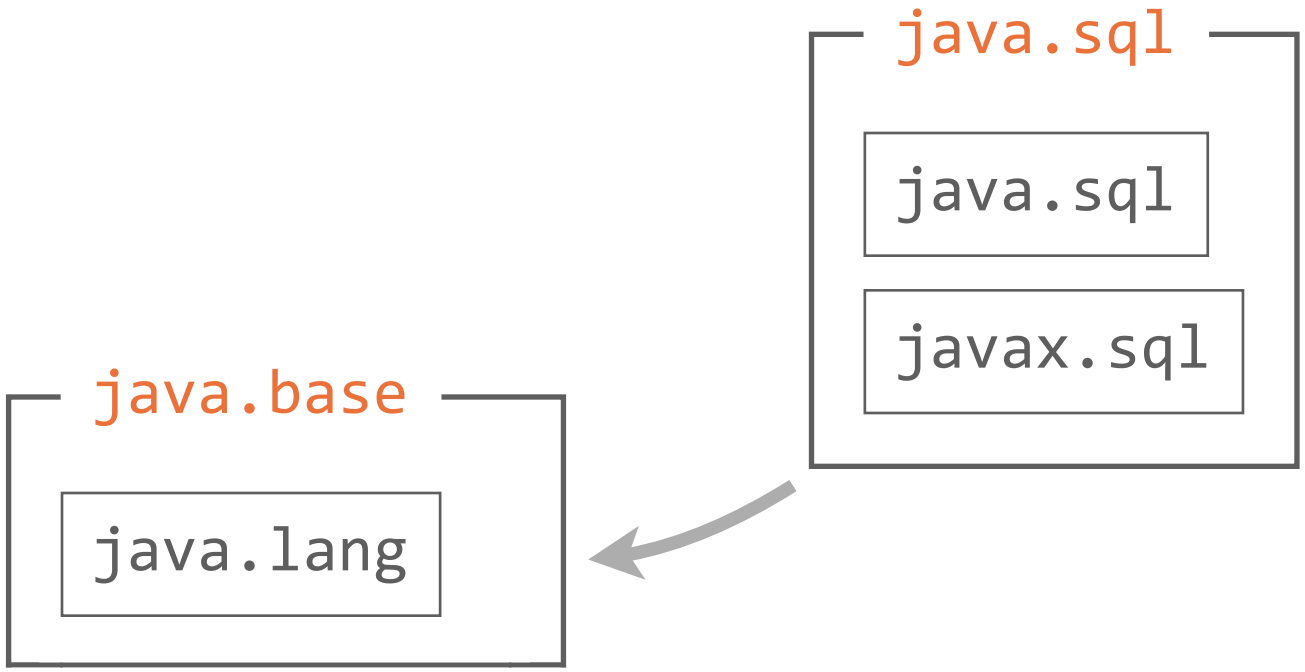
- Goals
  - Divide the Java SE Platform into a set of medium-grained modules
    - More flexible than current Compact Profiles
    - Not so fine-grained as to be difficult to use or costly to implement
  - Clearly distinguish between standard SE *vs*. implementation-specific modules
- Constraints
  - Compatibility
    - An SE 8 application that uses only standard, non-deprecated SE 8 APIs must continue to work, *unchanged*, on SE 9
  - Performance
    - Static footprint, startup time, memory footprint, and run-time performance of typical implementations must not degrade
      - Except that footprint may increase to accommodate new features


smartcardio
crypto.ec
crypto.pkcs11
hprof.agent
jdwp.agent
localedata
sctp
zipfs
security
.auth
scripting
.nashorn
rmic
xml.ws
hotspot.agent
javadoc
xml.bind
jconsole
jcmd
jdi
compiler
java.se
attach
xml.ws
jvmstat
httpserver
compact3
xml.soap
corba
instrument
sql.rowset
naming.rmi
xml.bind
security.acl
naming
naming.dns
compact2
activation
compiler
management
sql
desktop
prefs
compact1
rmi
security.jgss
annotations.common
xml.crypto
xml
logging
scripting
security.sasl
base
charsets

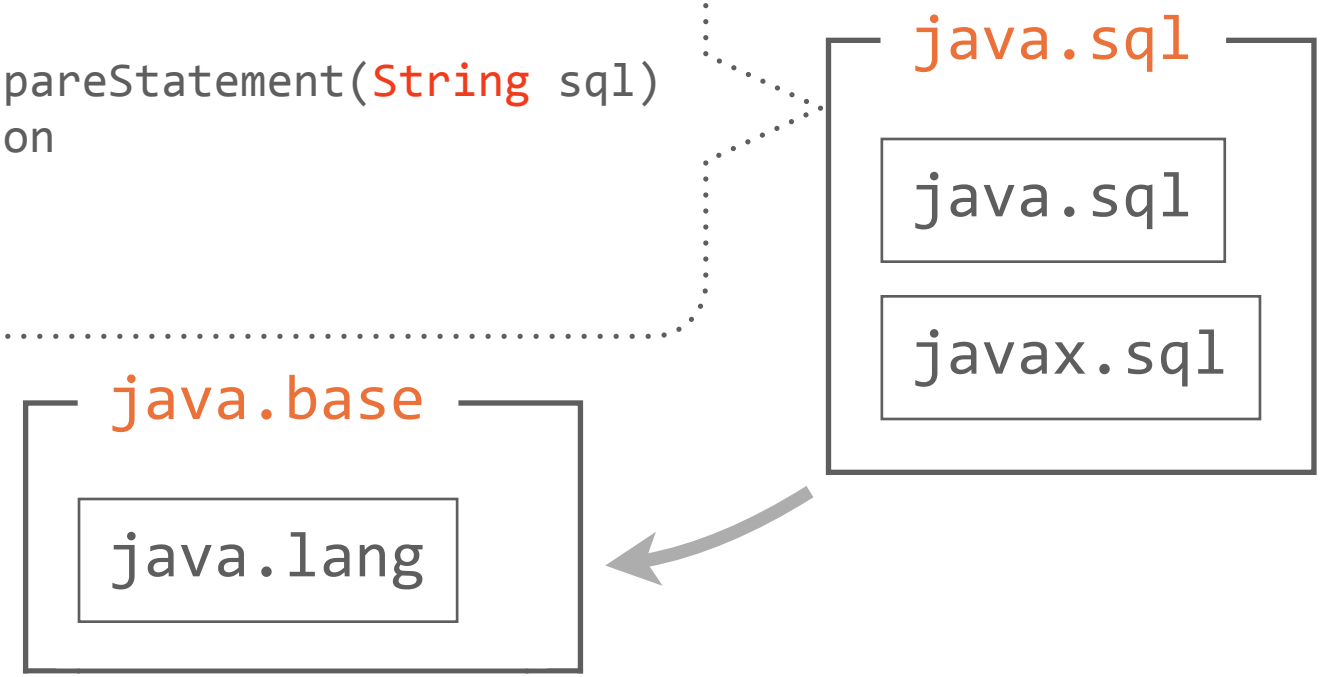




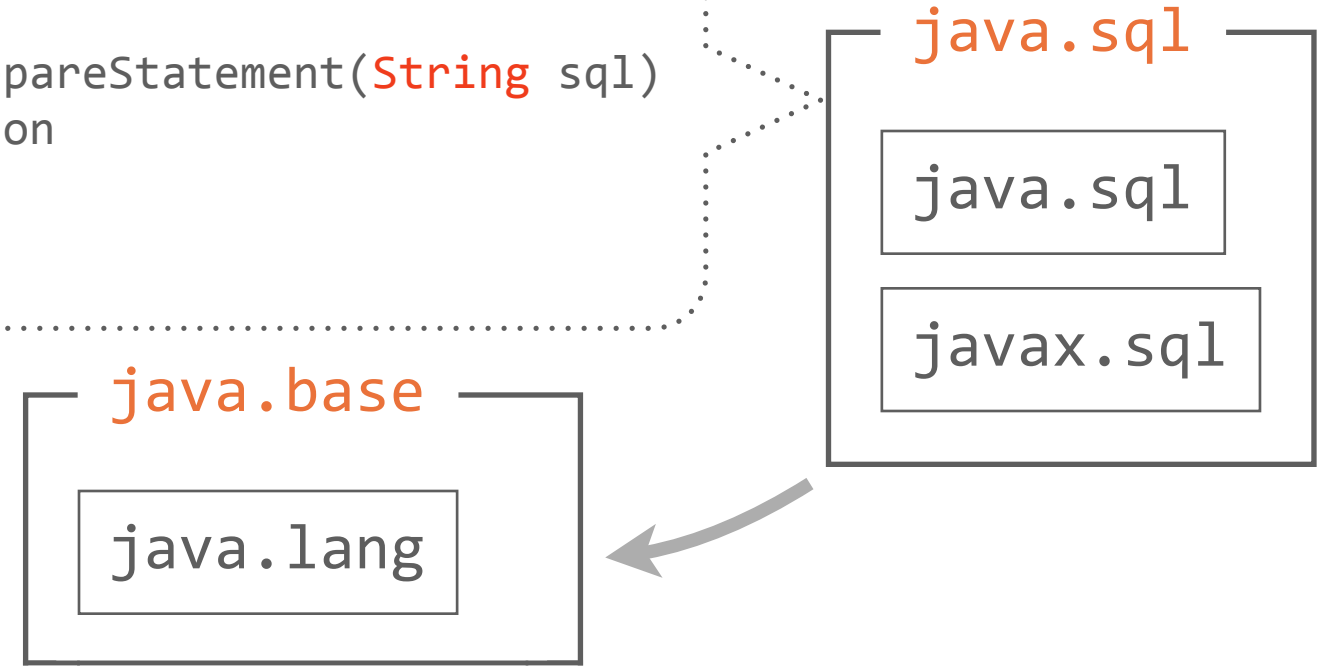





```
package java.sql;

public interface Connection {

    PreparedStatement prepareStatement(String sql)
        throws SQLException
    {
        ...
```

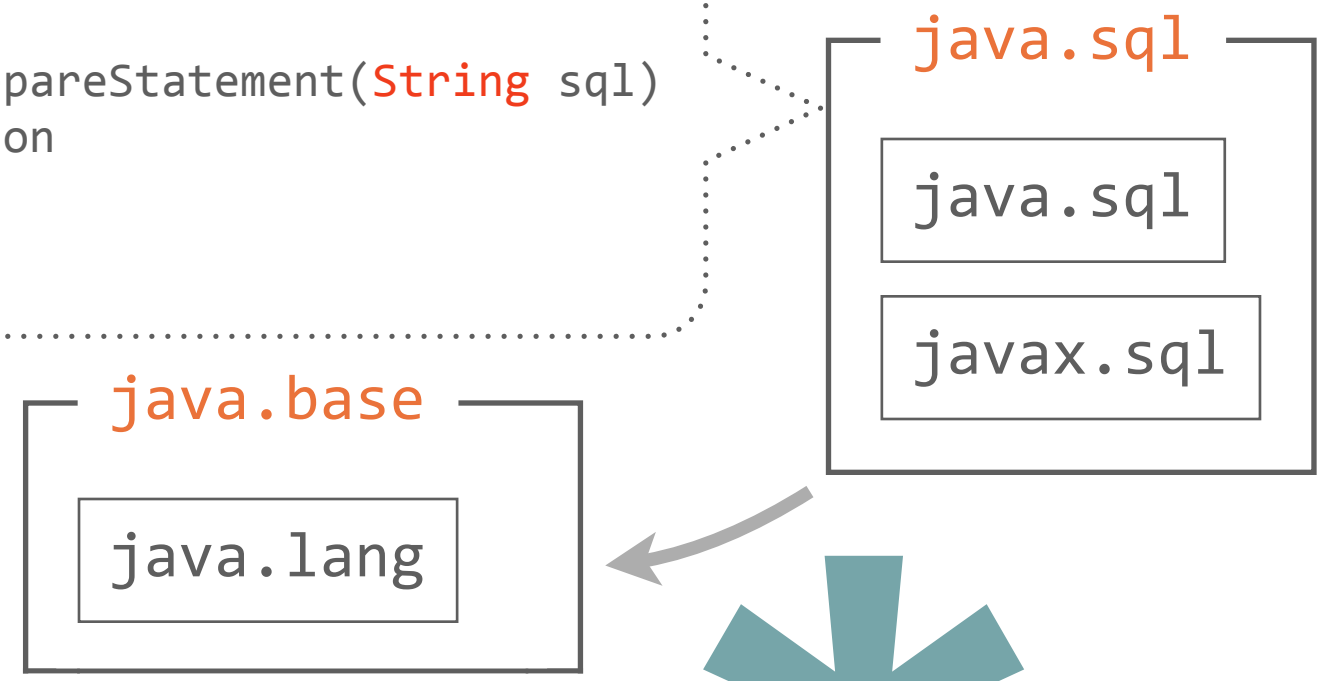



```
package java.sql;

public interface Connection {

    PreparedStatement prepareStatement(String sql)
        throws SQLException
    {
        ...
```



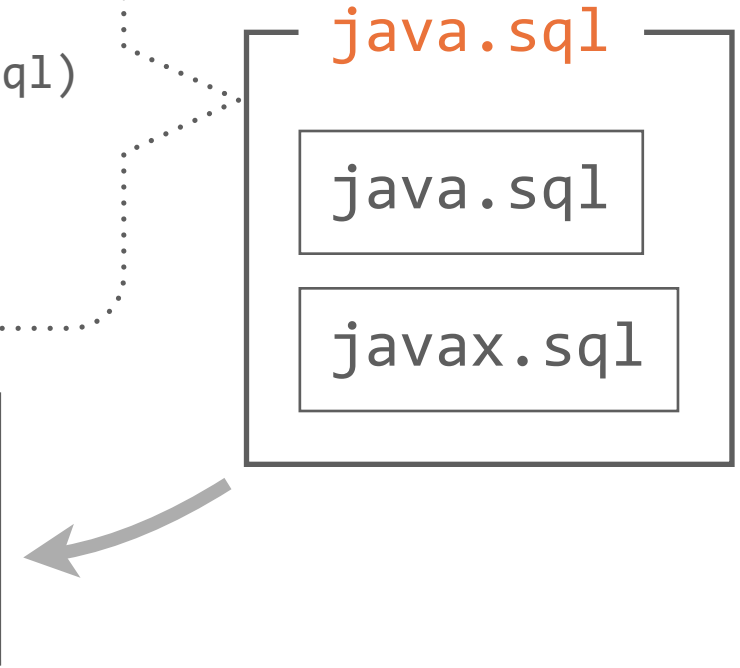
java.lang not exported by java.base


```
package java.sql;

public interface Connection {

    PreparedStatement prepareStatement(String sql)
        throws SQLException
    {
        ...
```



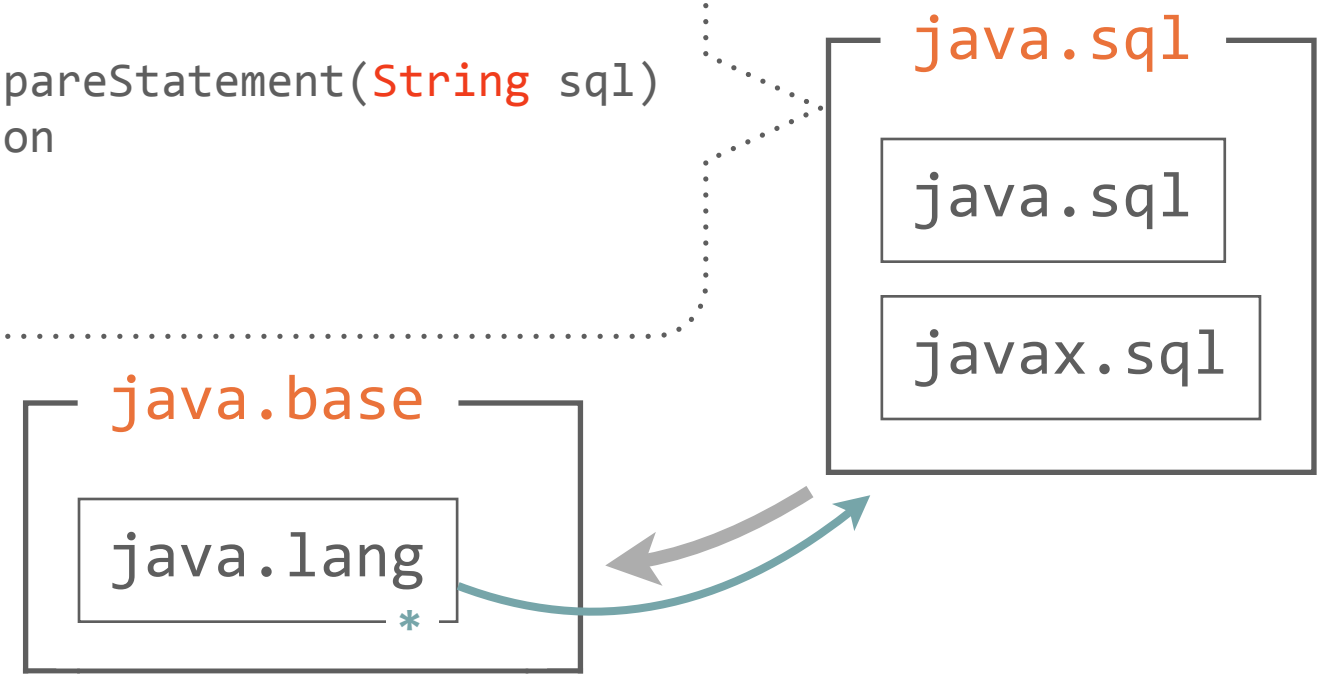
java.lang not exported by java.base


```
package java.sql;

public interface Connection {

    PreparedStatement prepareStatement(String sql)
        throws SQLException
    {
        ...
```

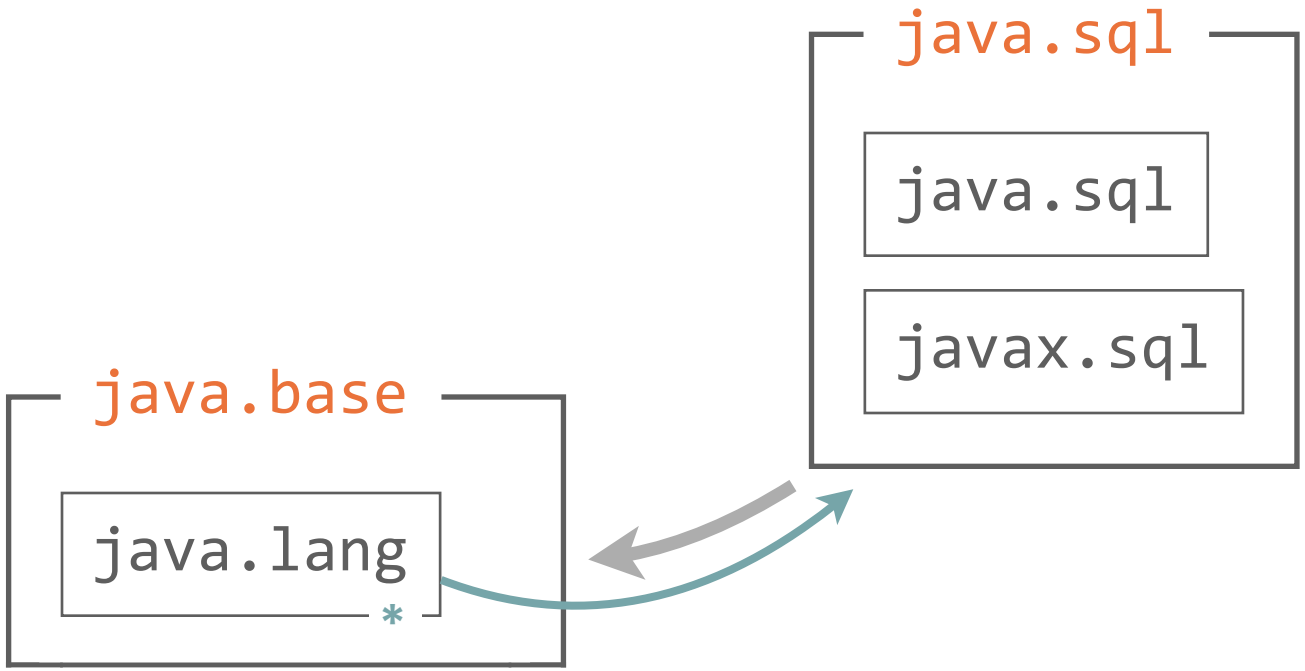


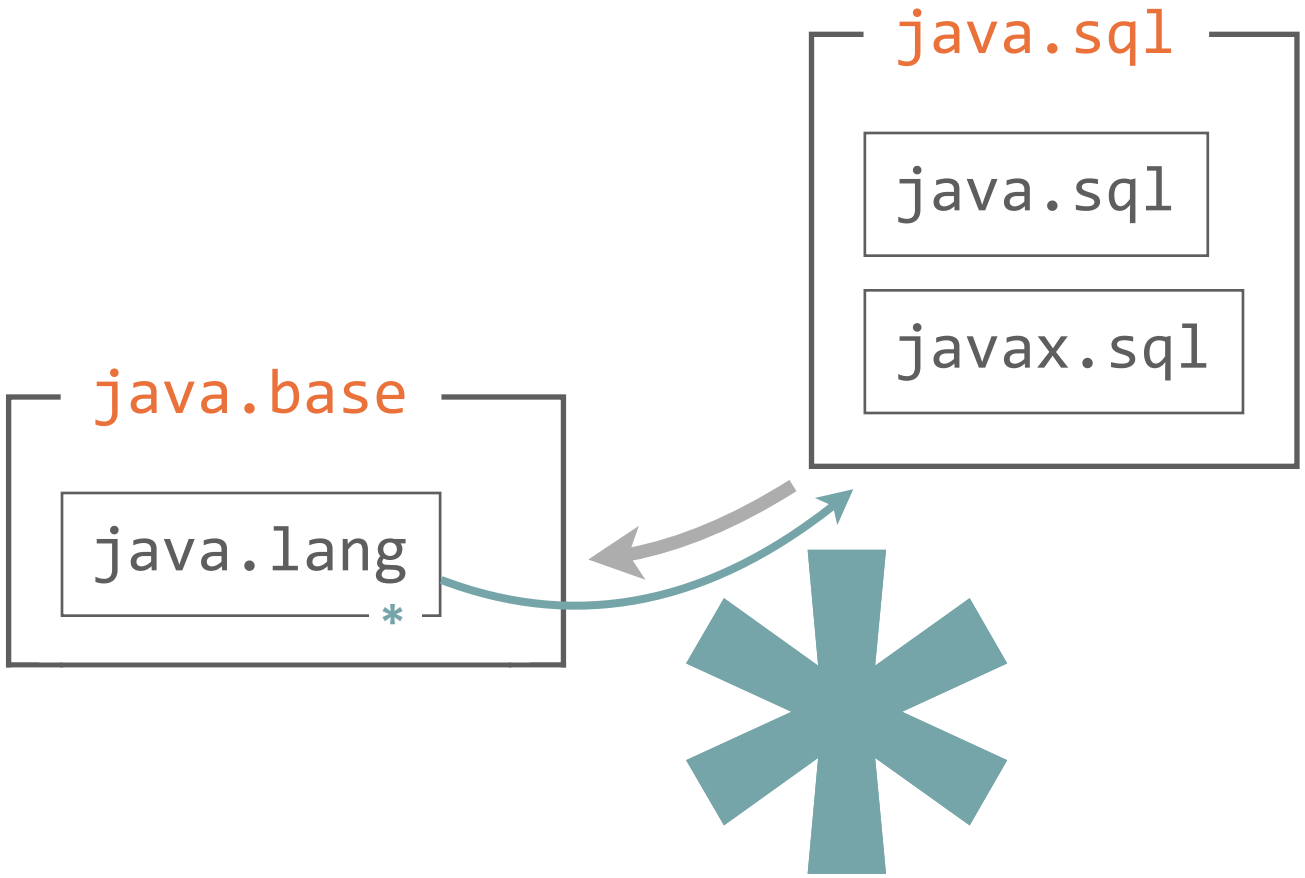
java.lang not exported by java.base

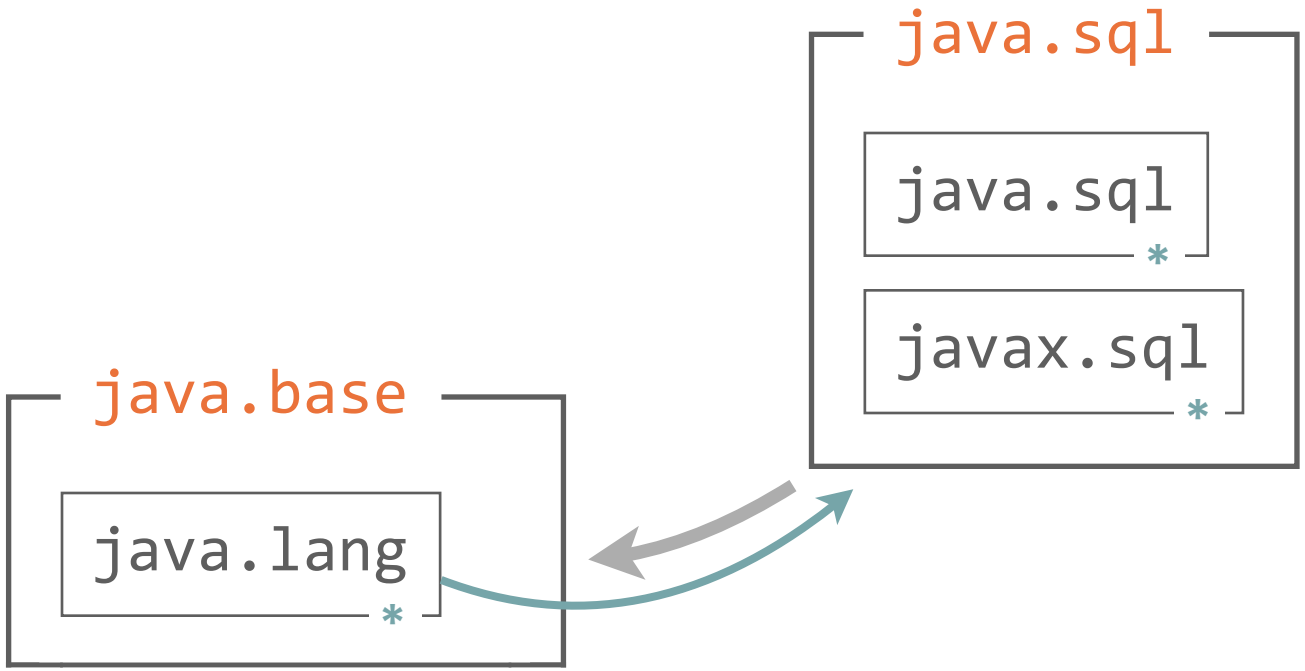

```
package java.sql;

public interface Connection {

    PreparedStatement prepareStatement(String sql)
        throws SQLException
    {
        ...
```



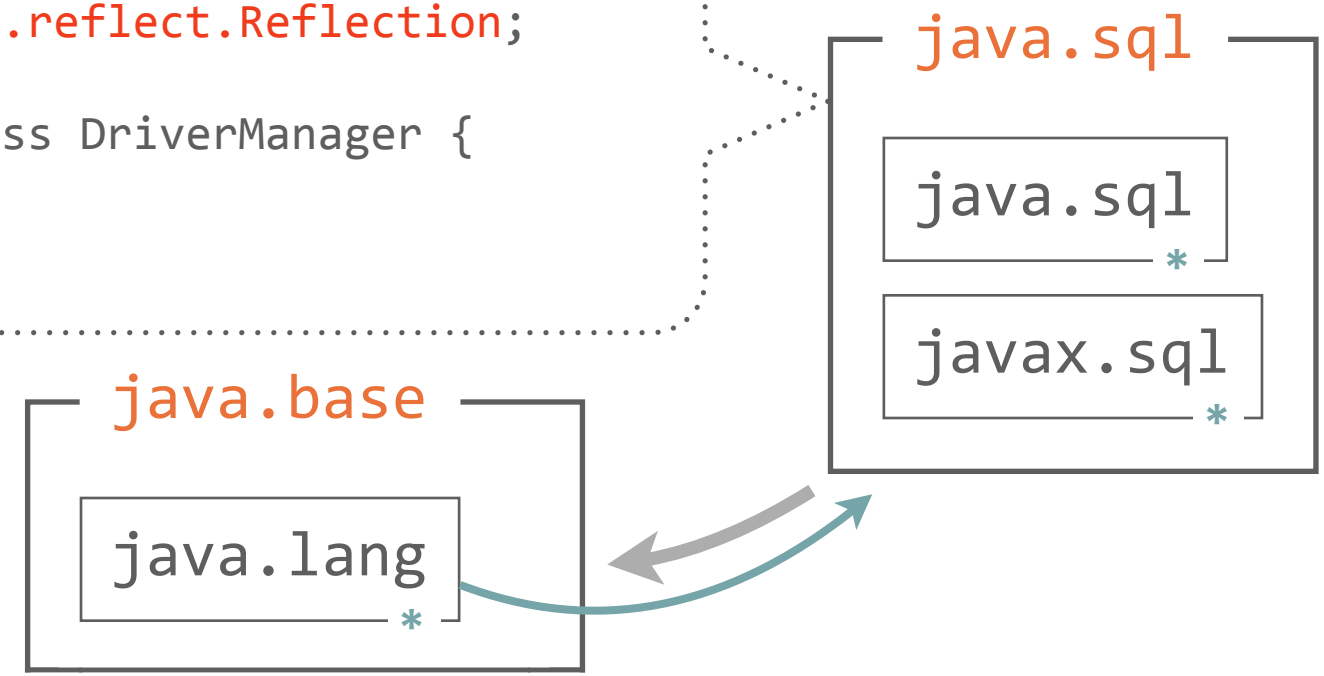
java.lang not exported by java.base



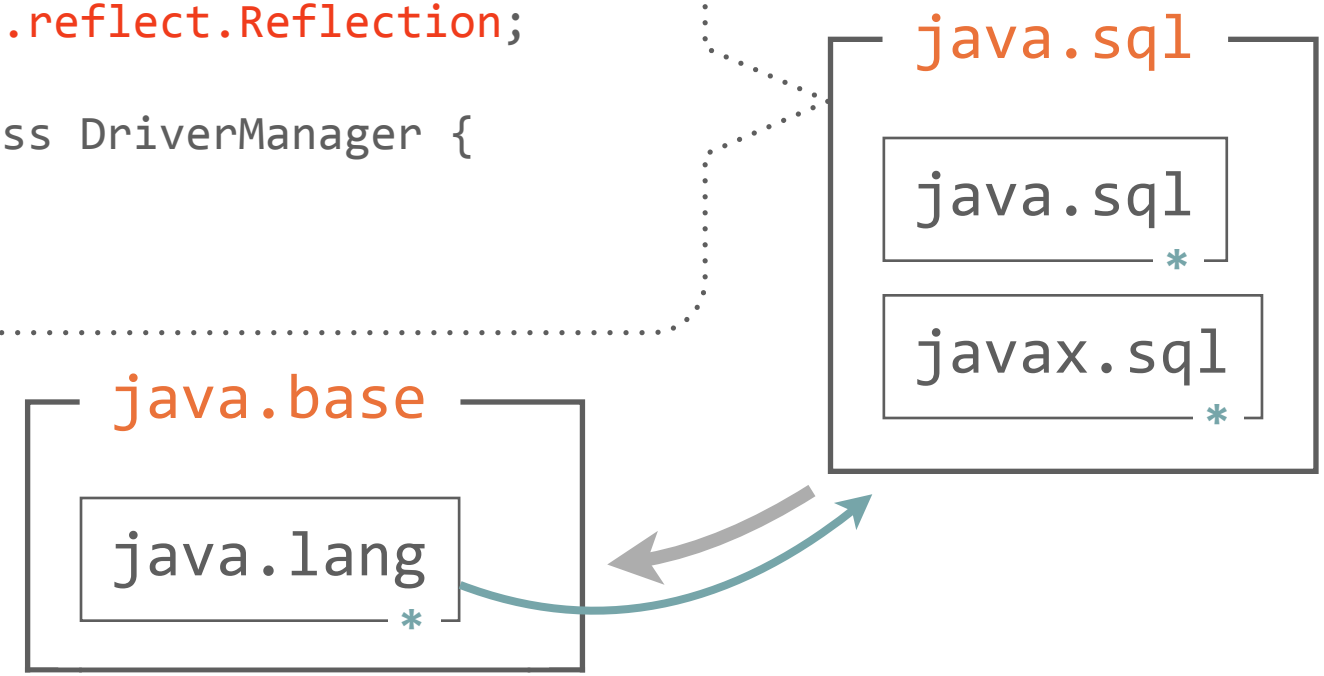





```
package java.sql;

import sun.reflect.CallerSensitive;
import sun.reflect.Reflection;

public class DriverManager {

    ...
```

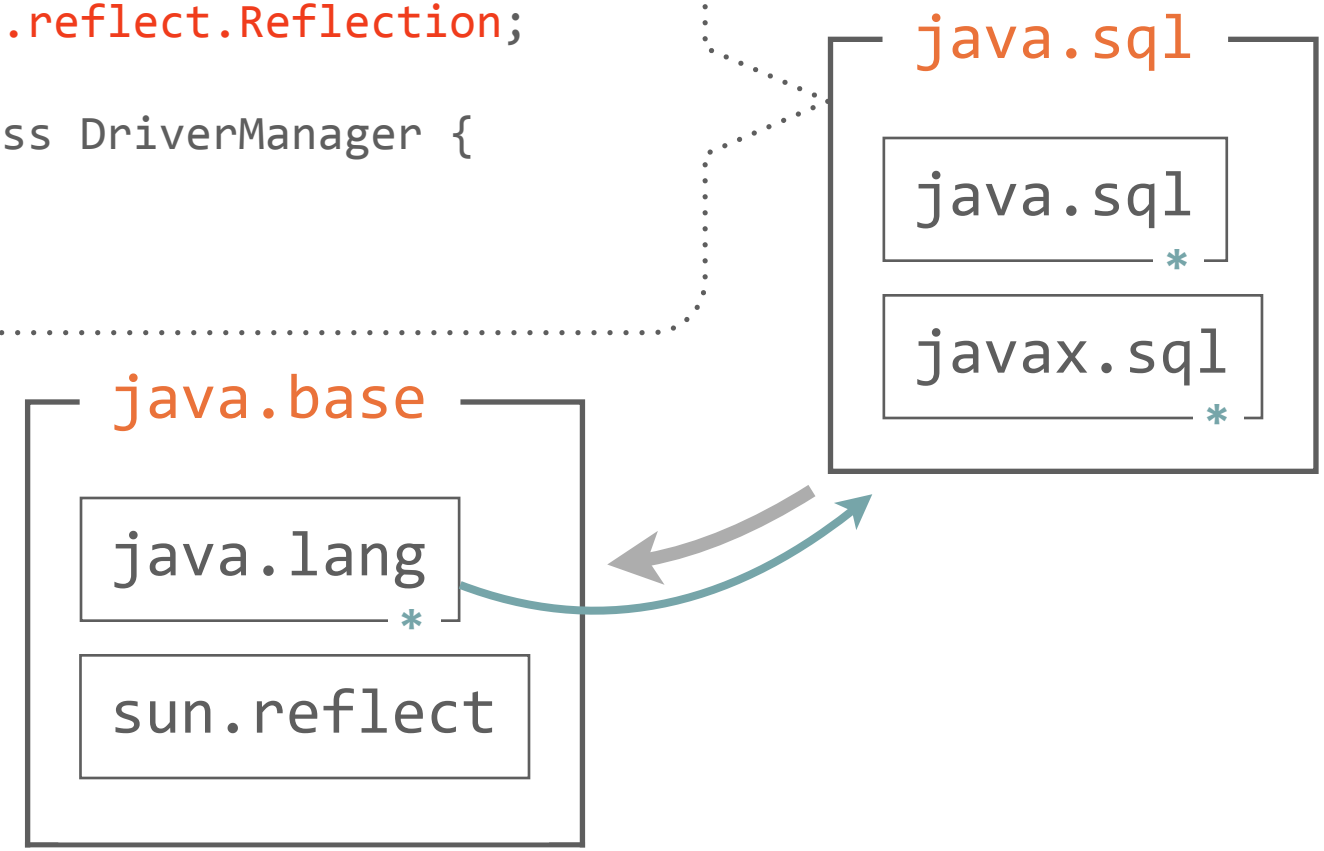



```
package java.sql;

import sun.reflect.CallerSensitive;
import sun.reflect.Reflection;

public class DriverManager {

    ...
```



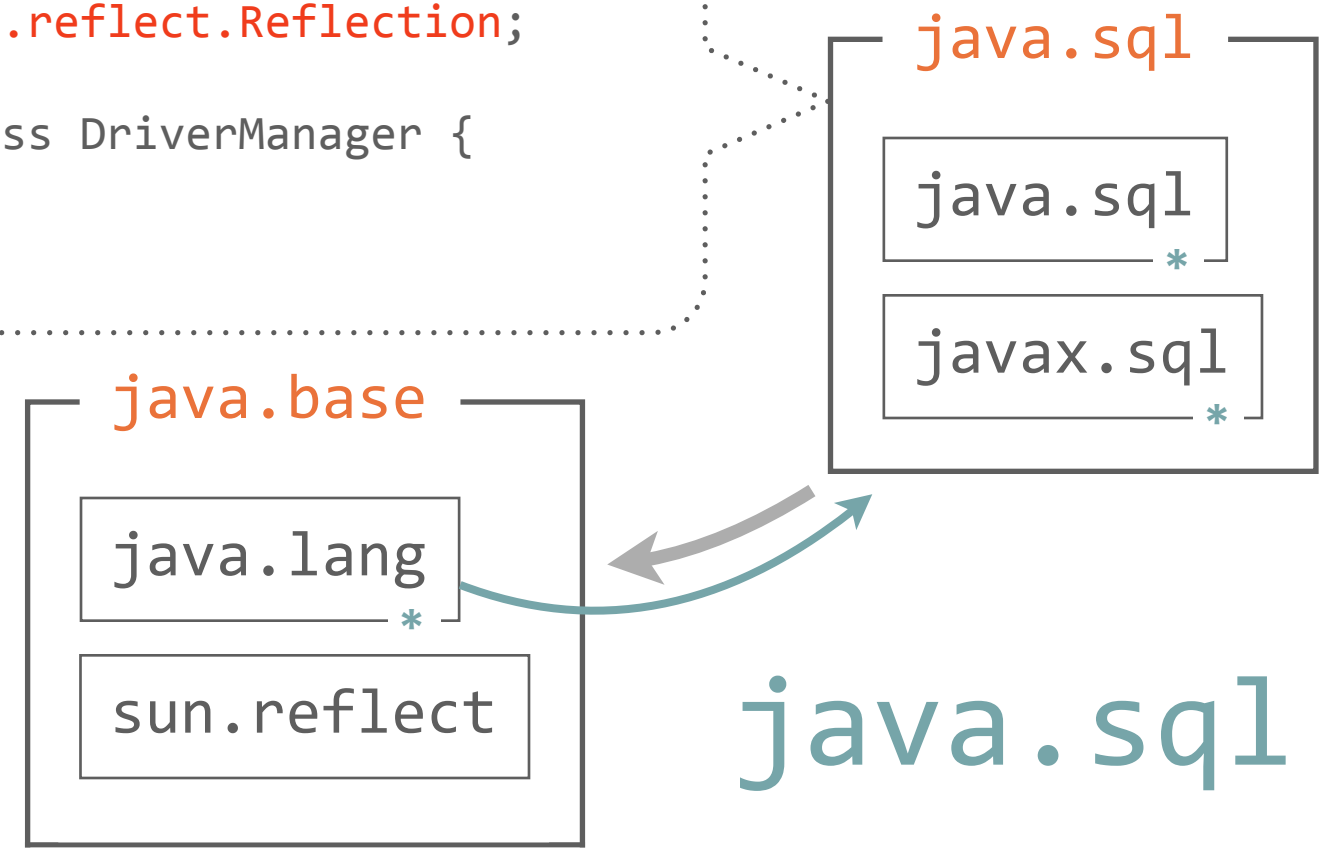
sun.reflect not exported by java.base

```
package java.sql;

import sun.reflect.CallerSensitive;
import sun.reflect.Reflection;

public class DriverManager {

    ...
```



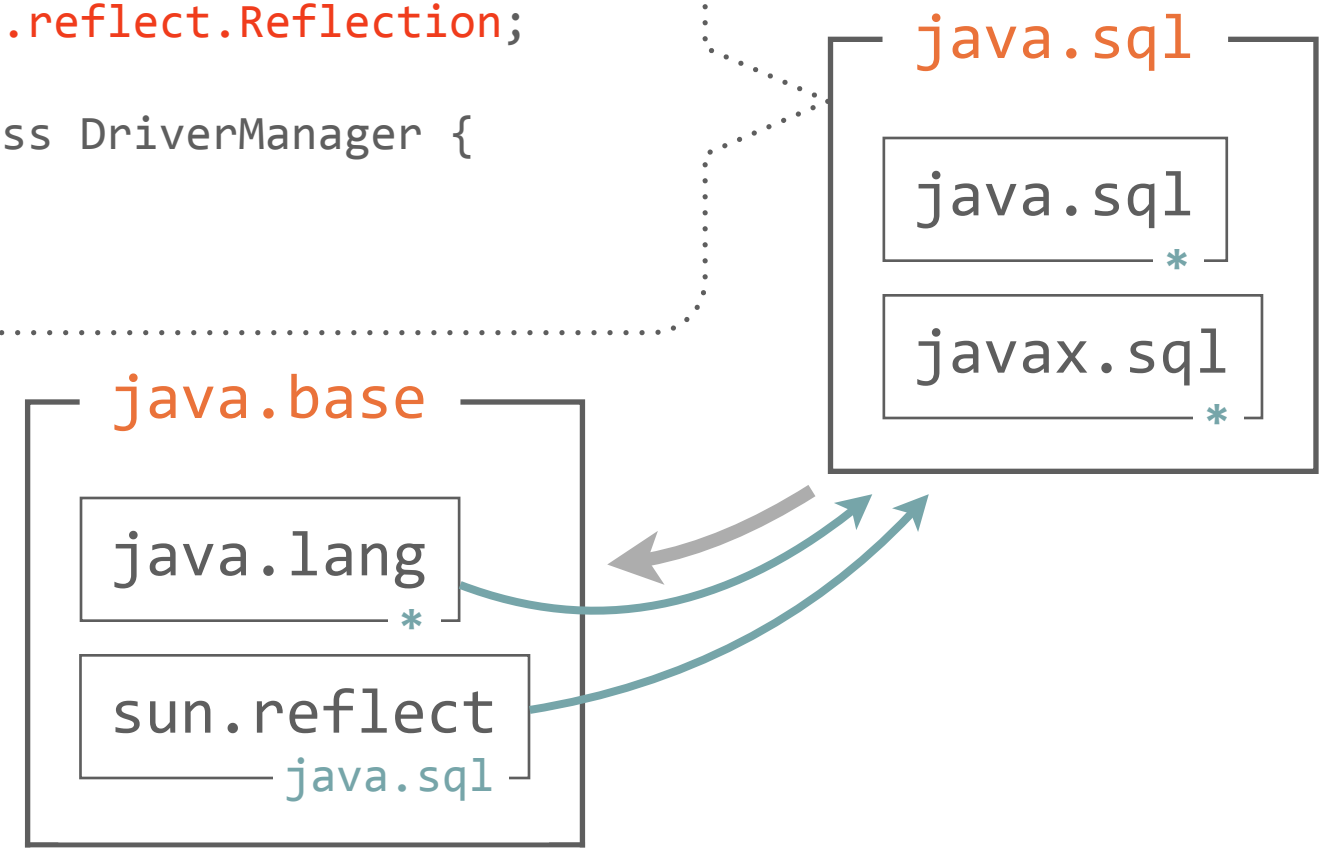
sun.reflect not exported by java.base

```
package java.sql;

import sun.reflect.CallerSensitive;
import sun.reflect.Reflection;

public class DriverManager {

    ...
```

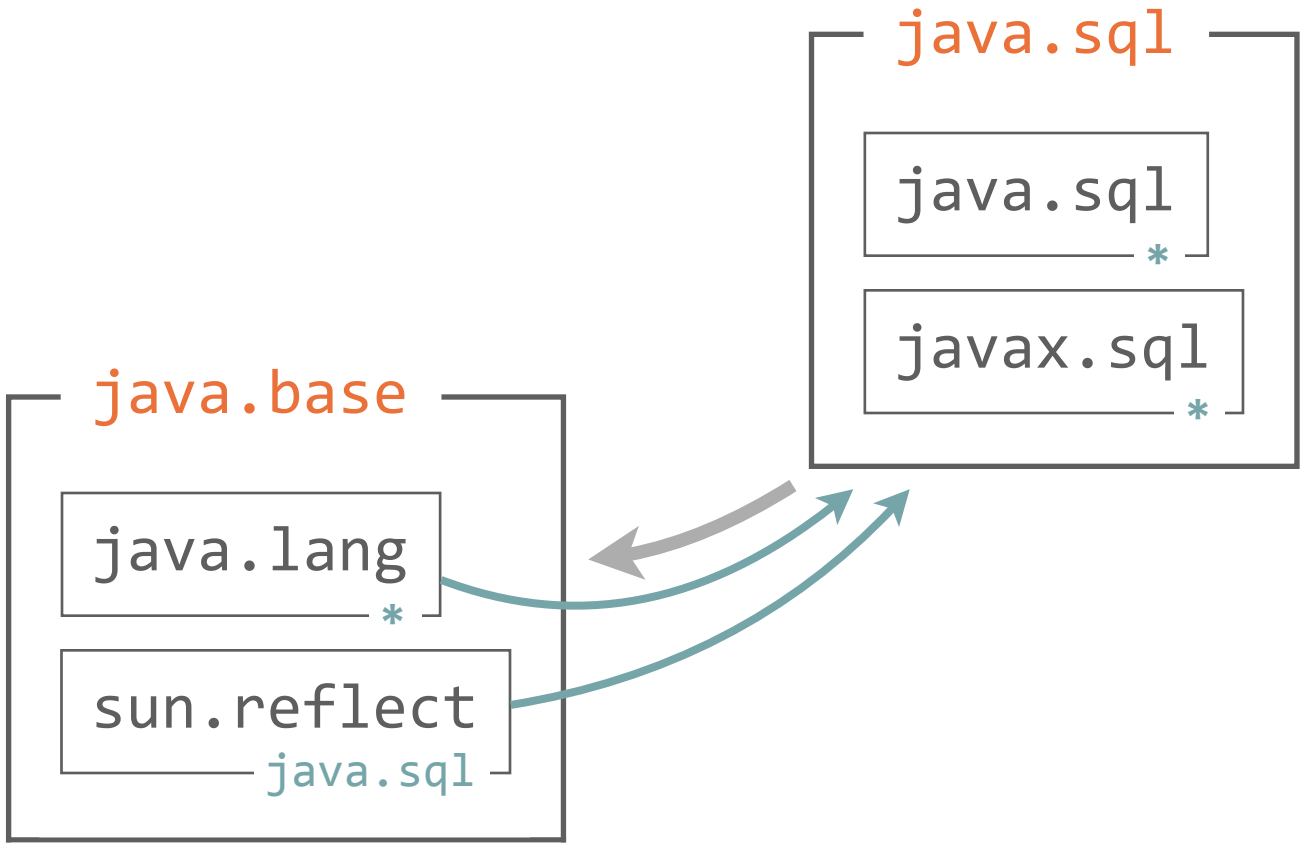


sun.reflect not exported by java.base

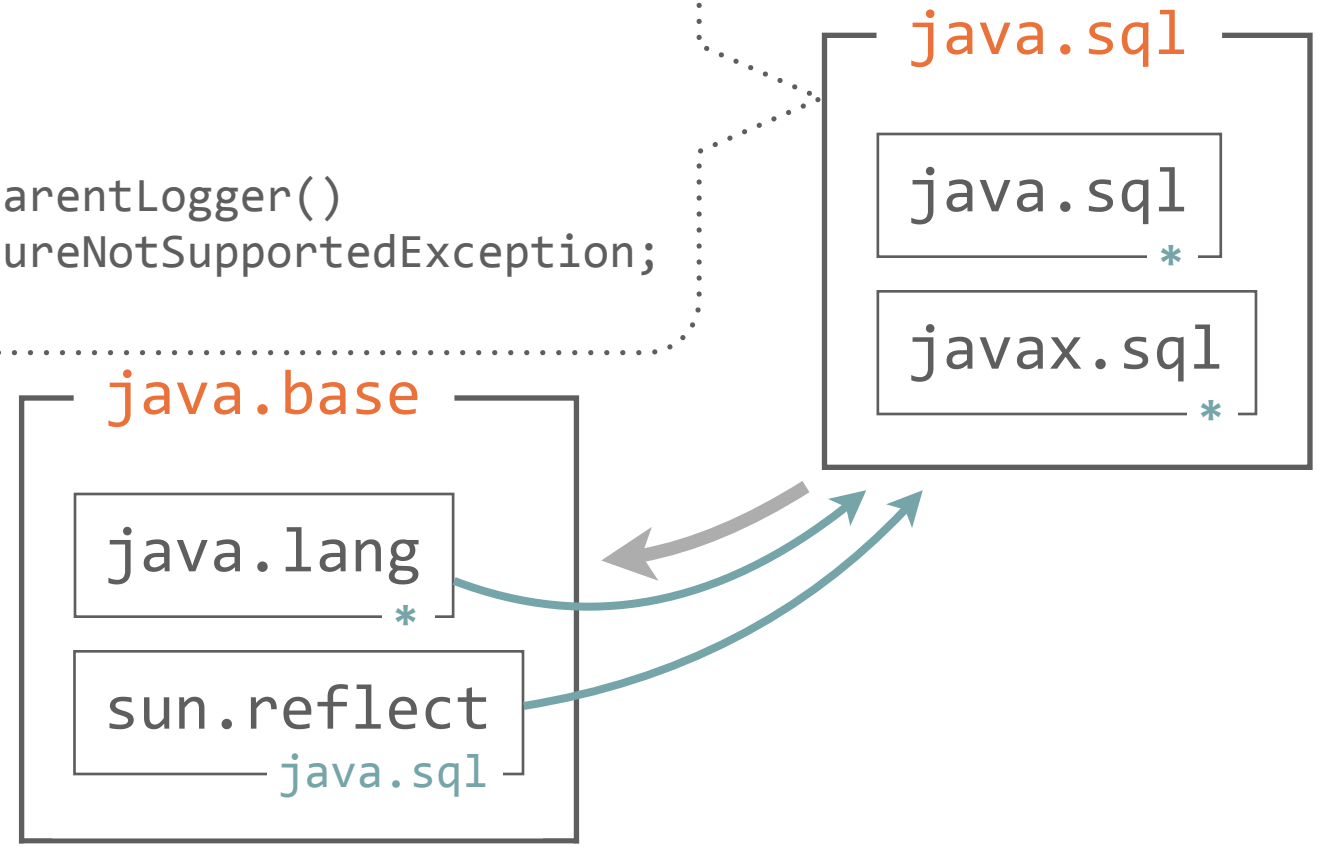

```
package java.sql;

import sun.reflect.CallerSensitive;
import sun.reflect.Reflection;

public class DriverManager {

    ...
```

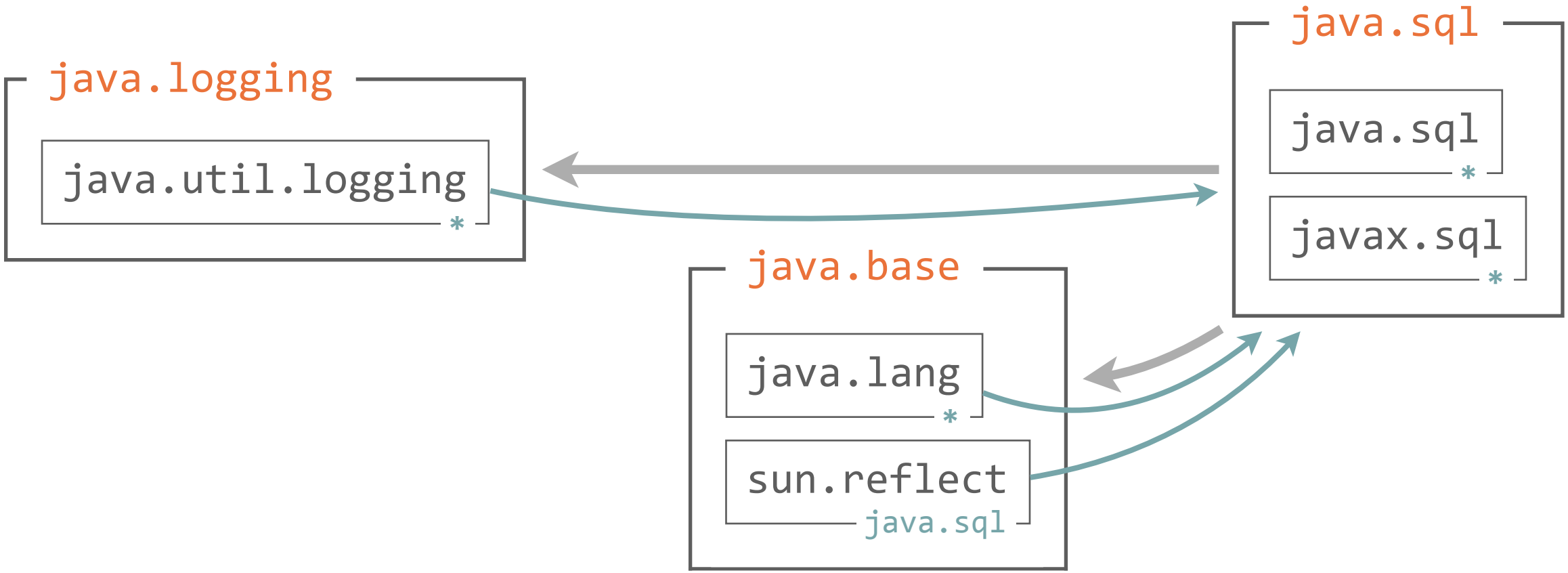


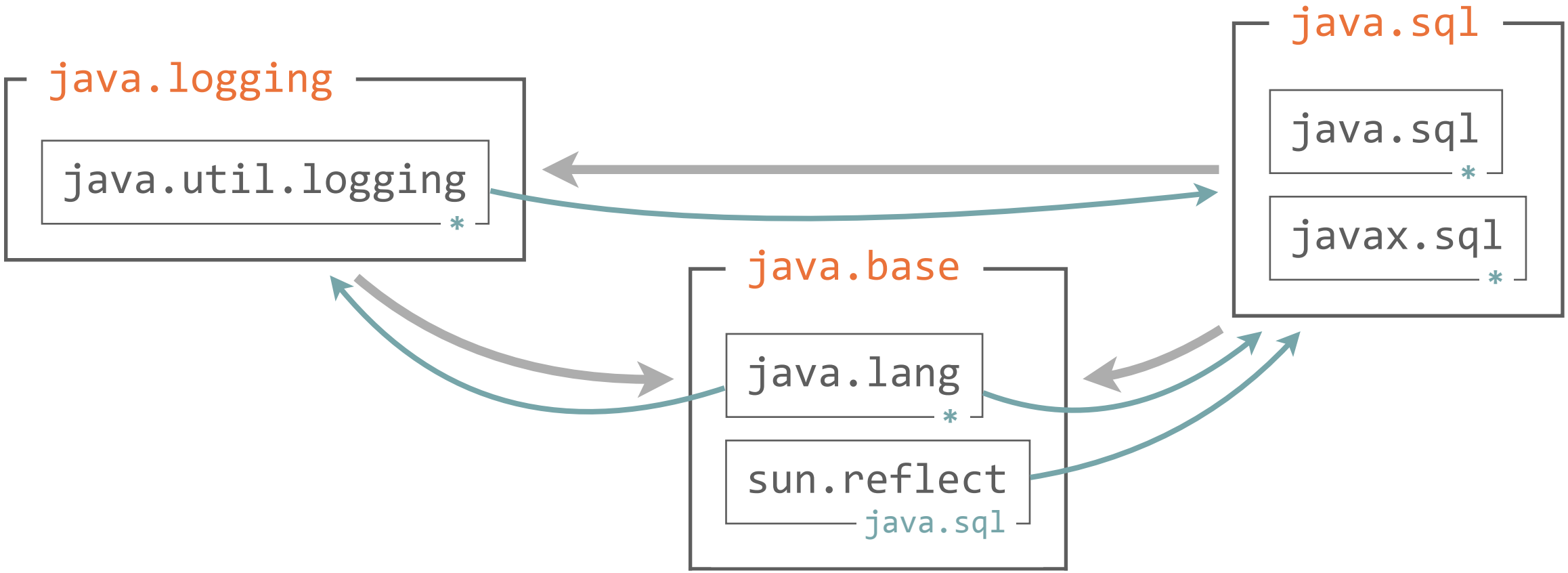
sun.reflect not exported by java.base

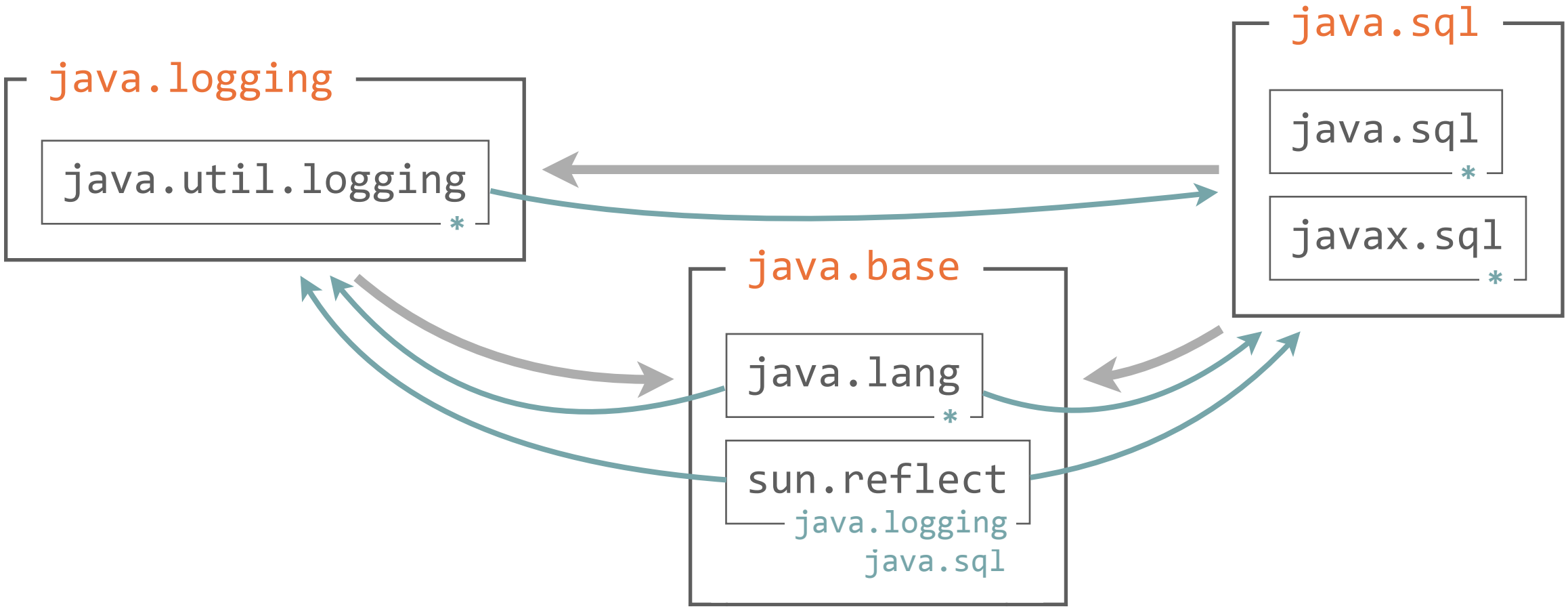


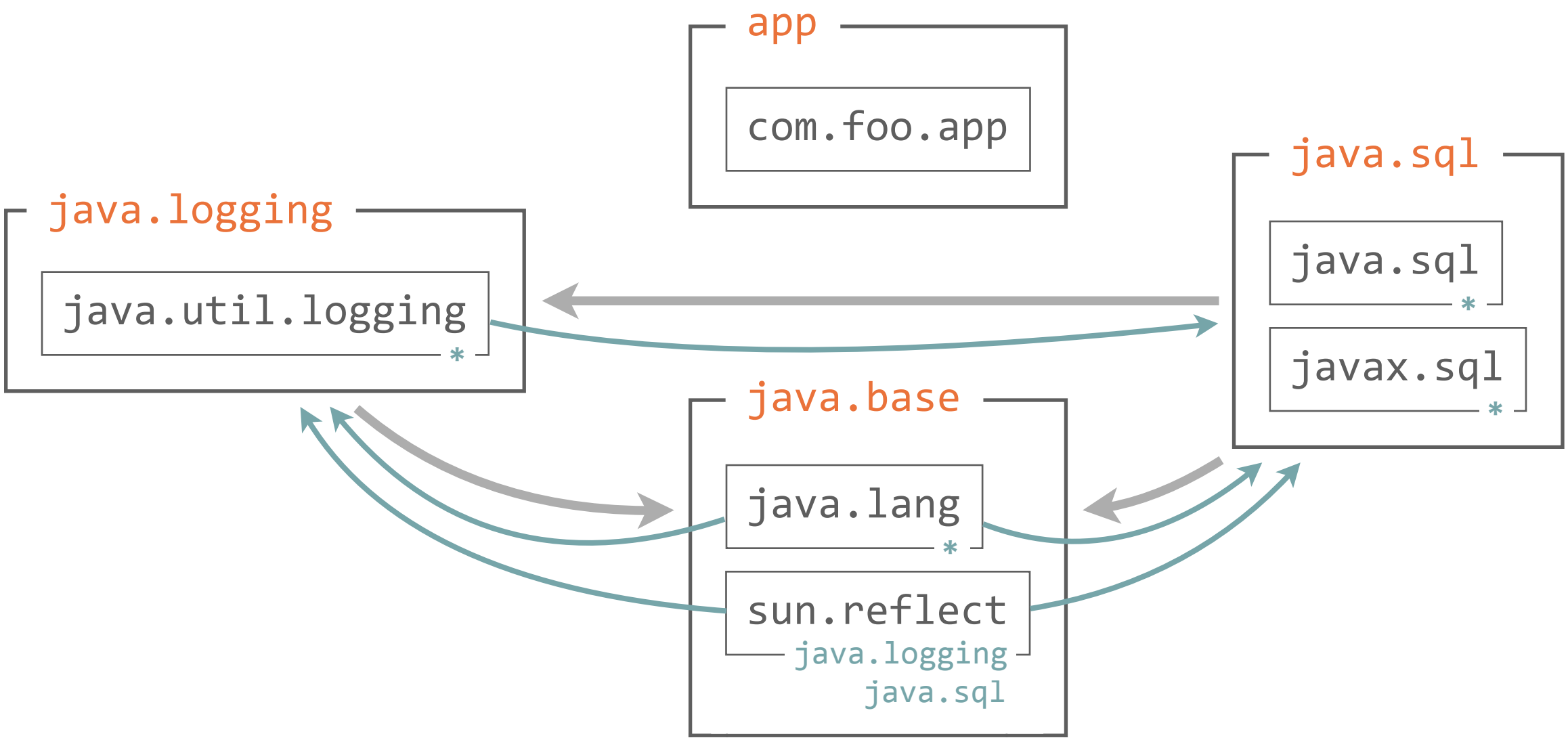

```
package java.sql;

import java.util.logging.Logger;

public class Driver {

    public Logger getParentLogger()
        throws SQLFeatureNotSupportedException;
```

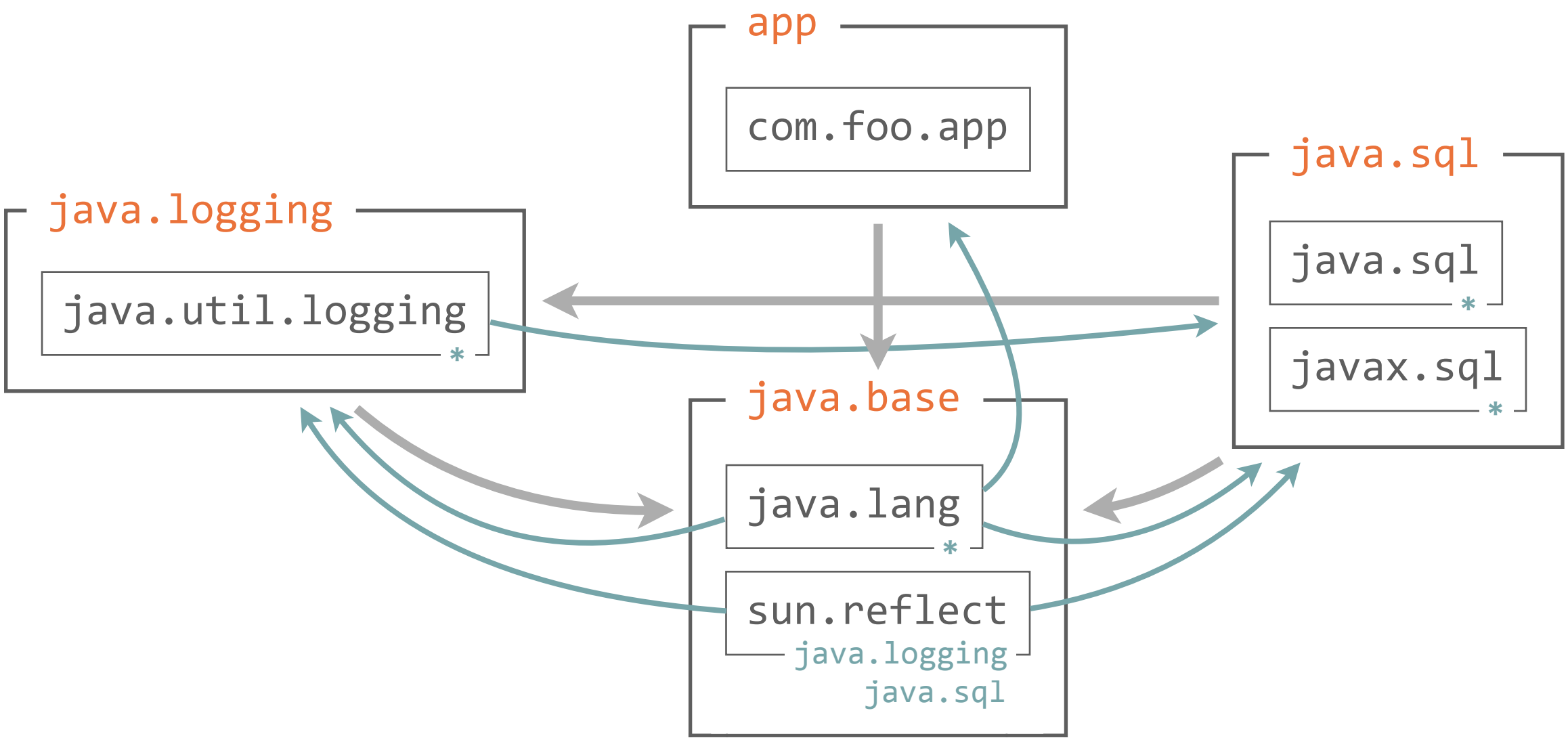


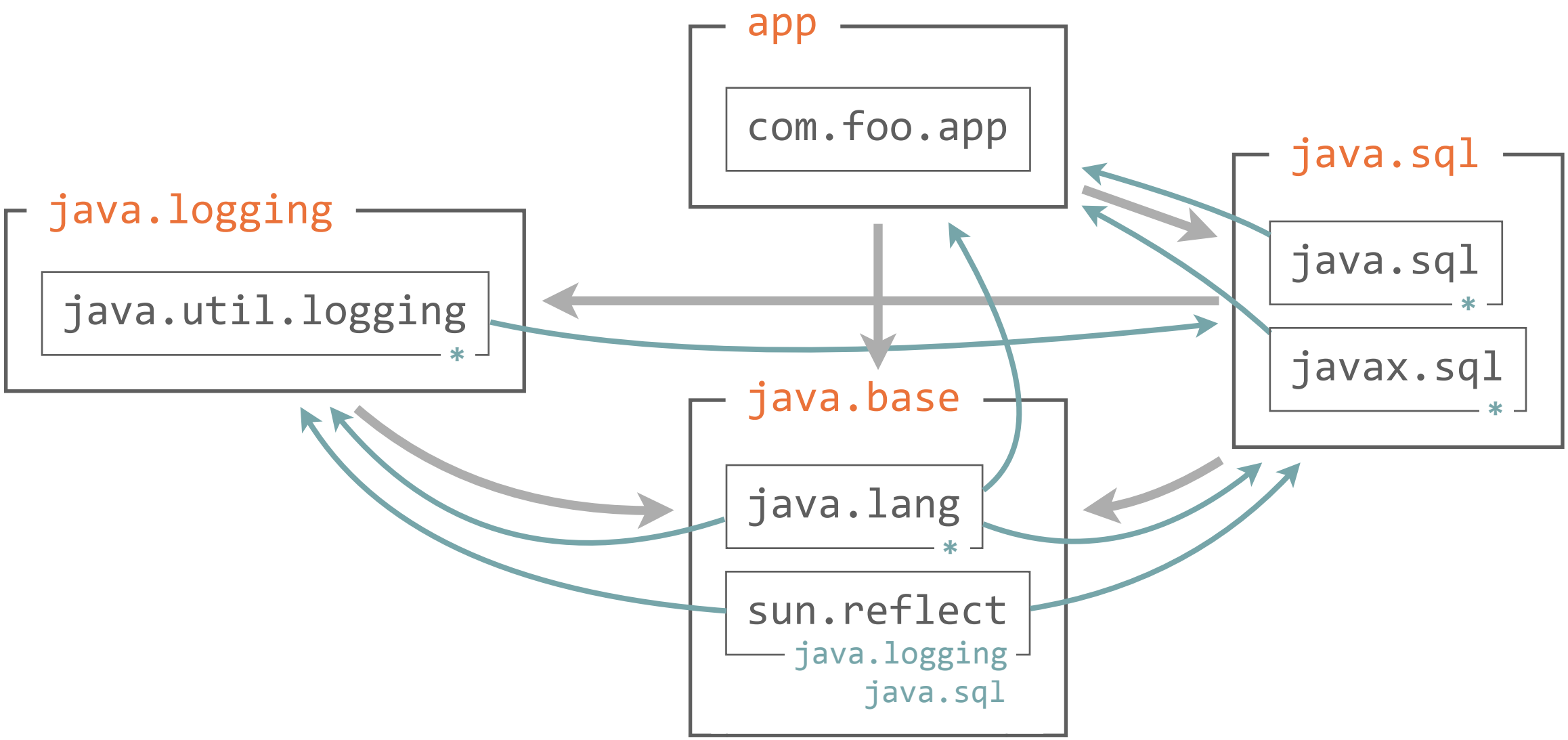


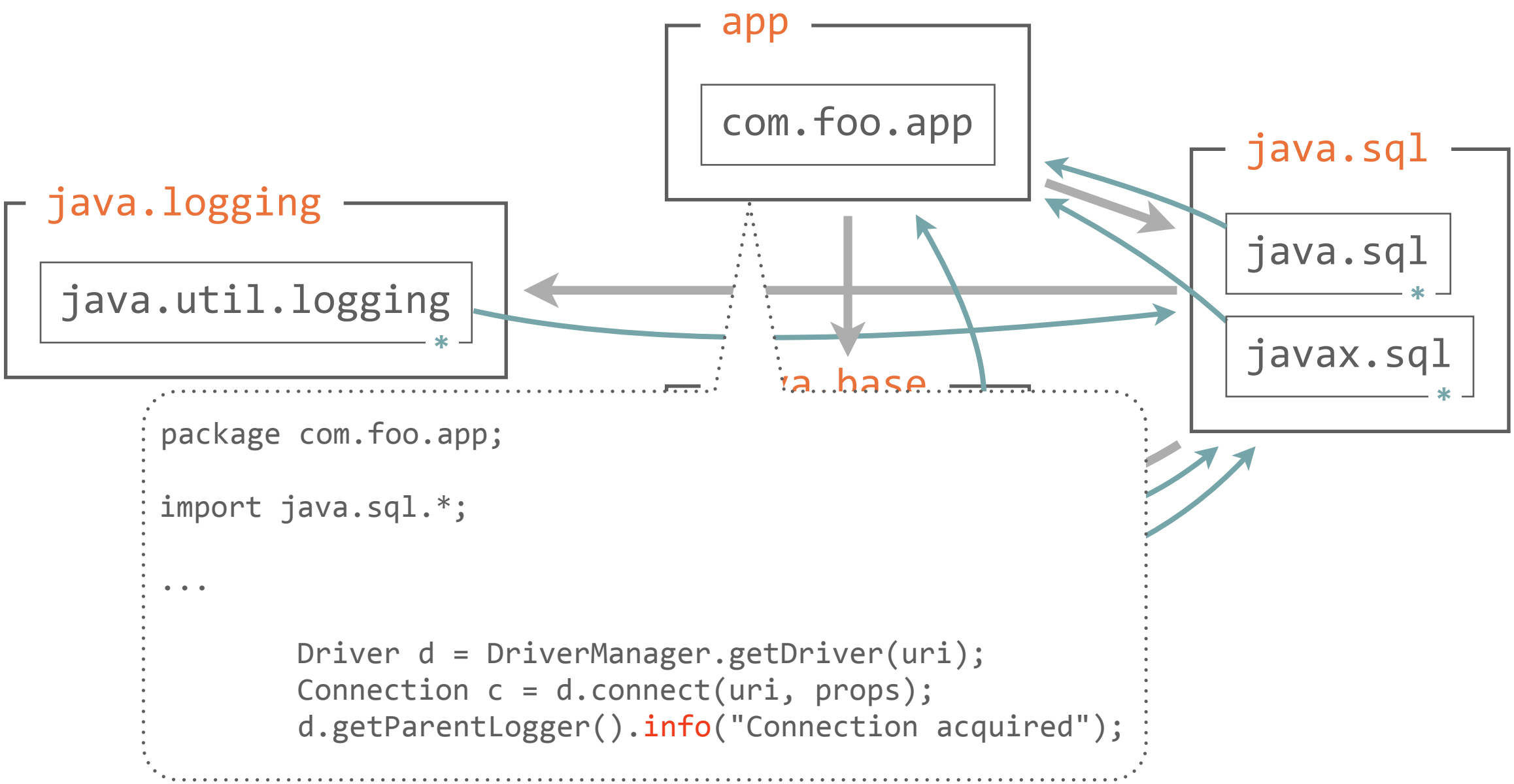


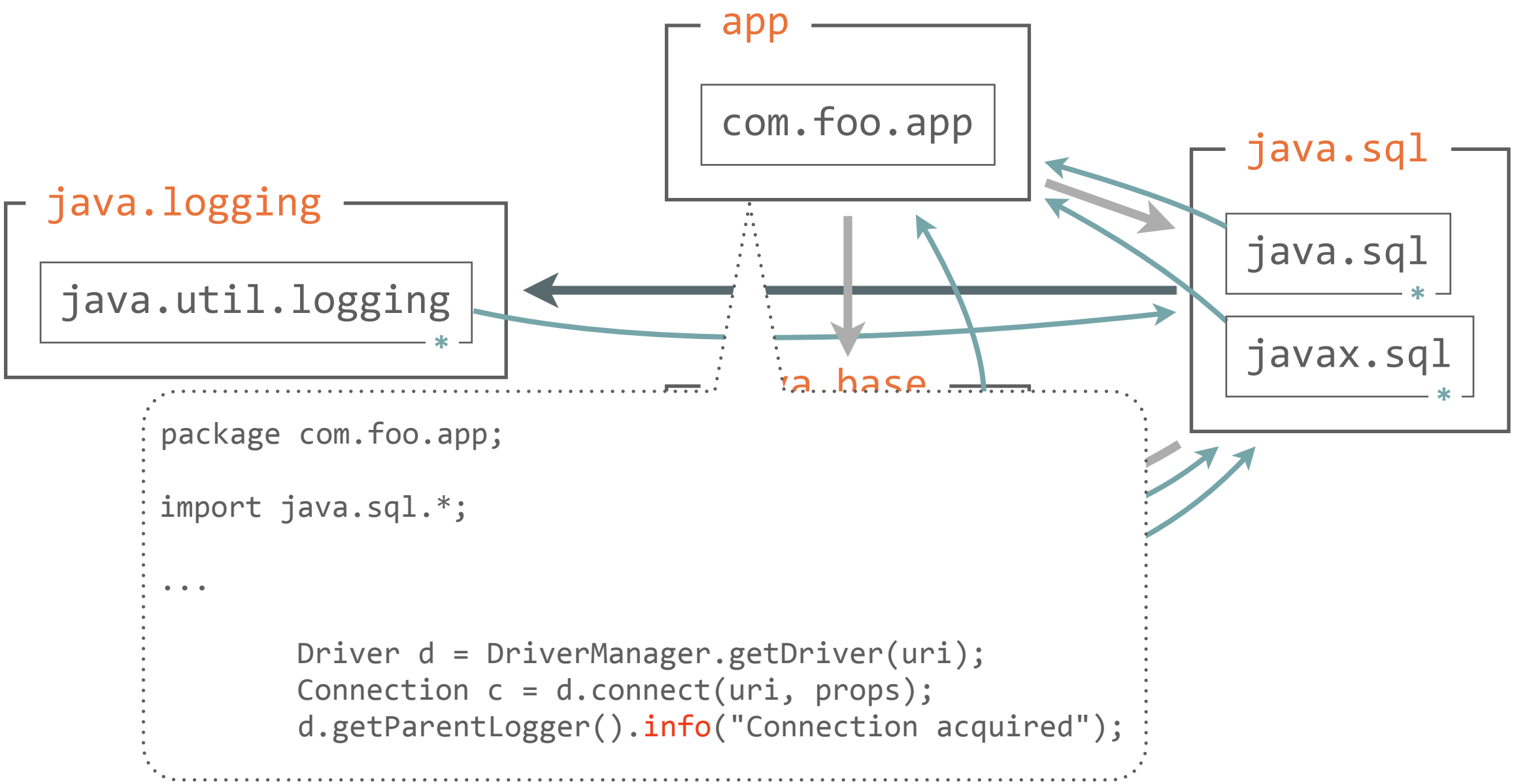


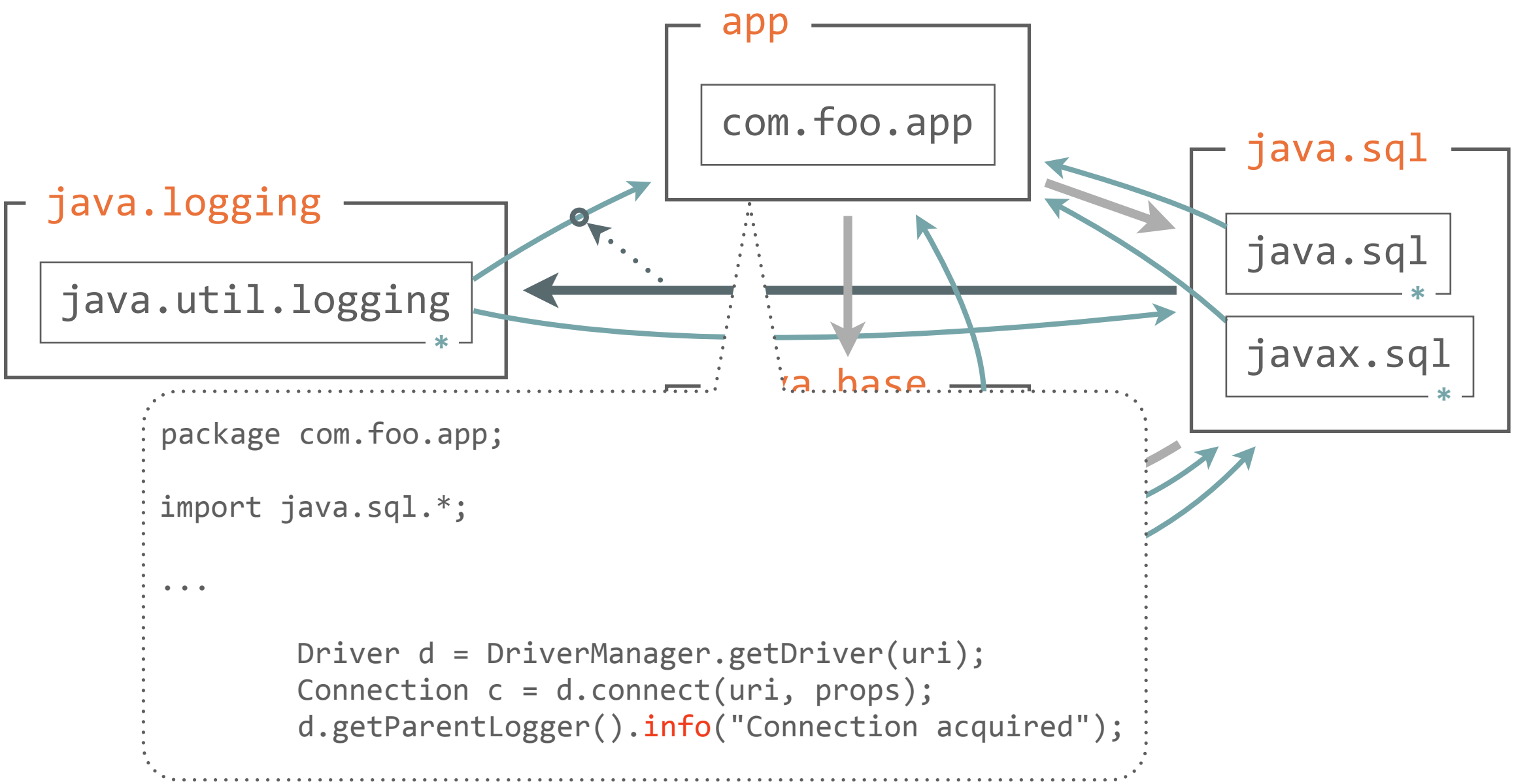


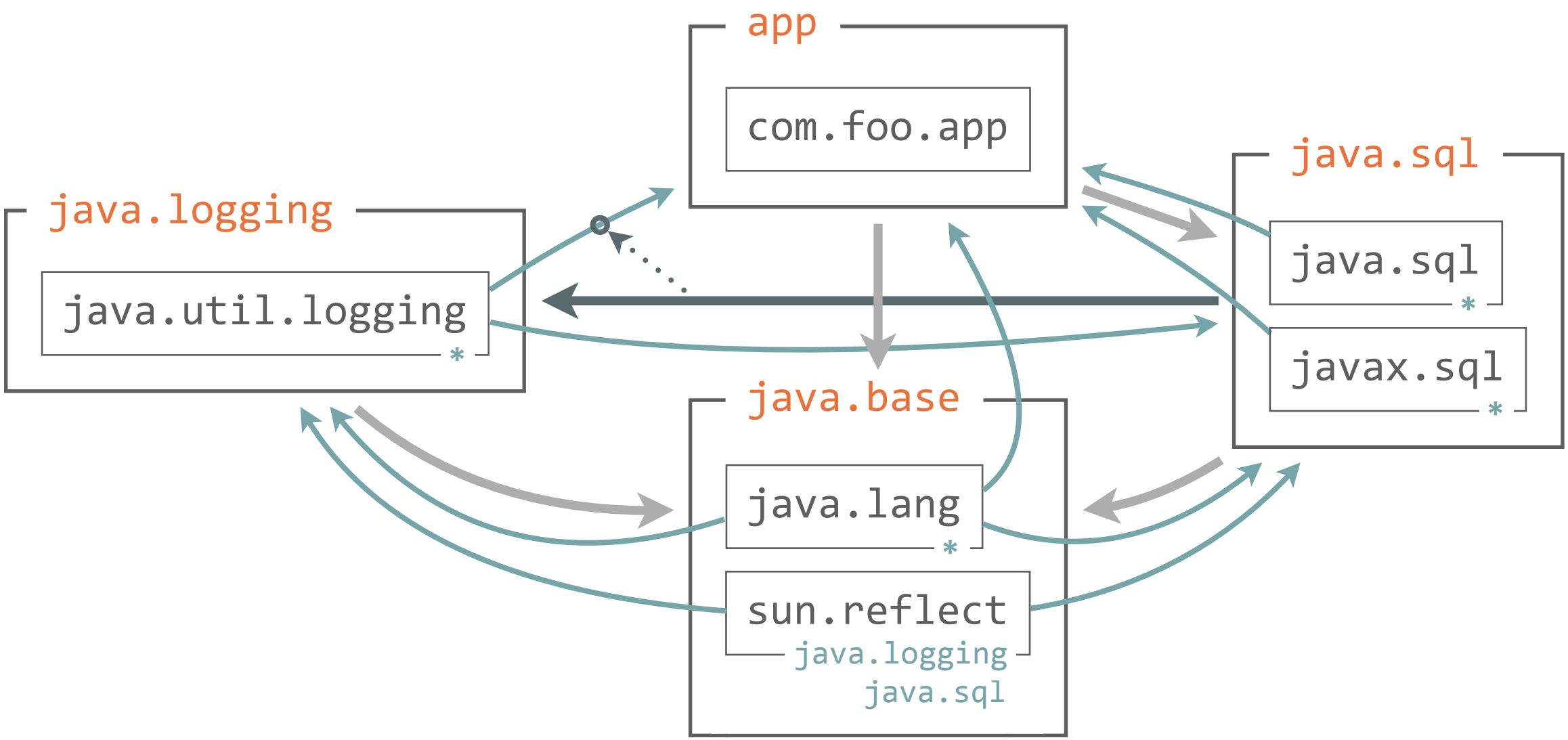


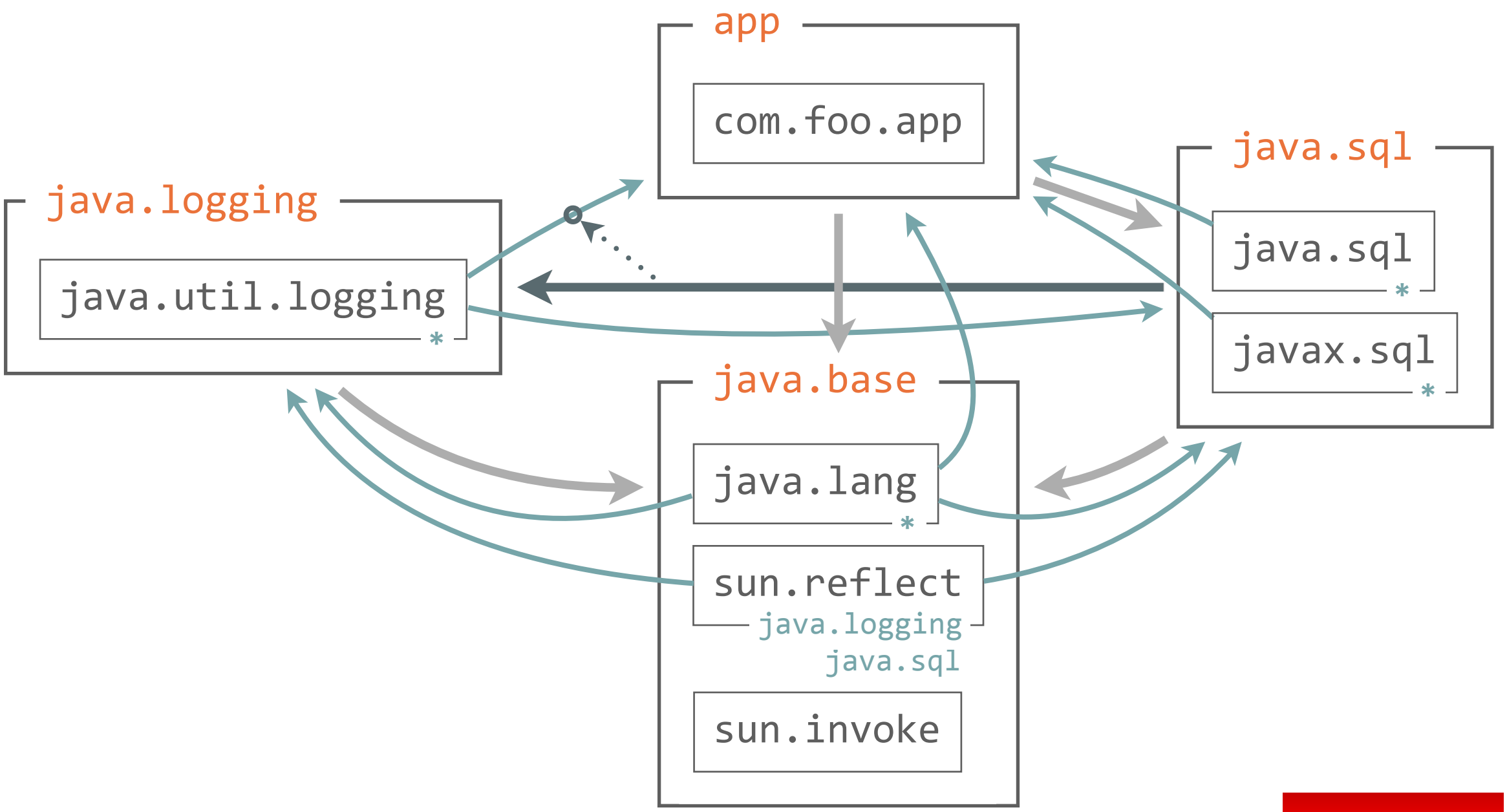


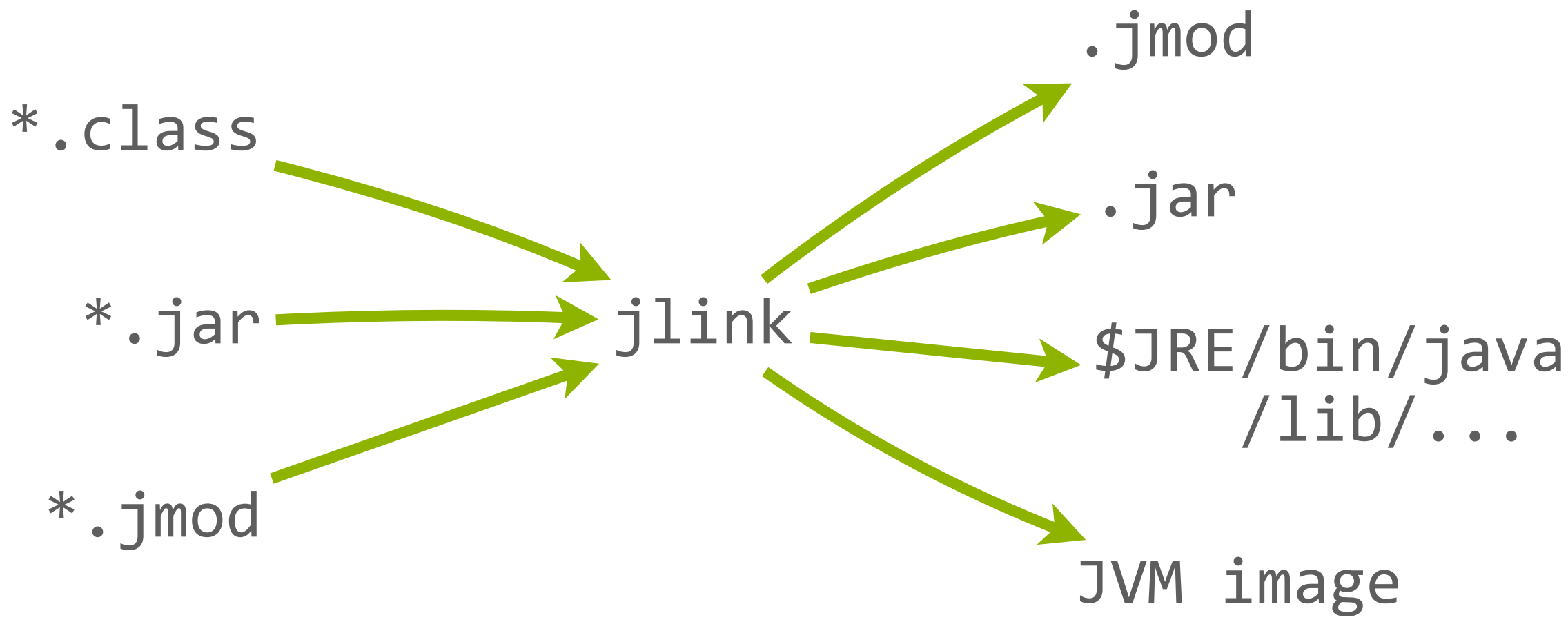












The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Java SE Platform Update

CREATE
THE
FUTURE

Mark Reinhold (@mreinhold)
Chief Architect, Java Platform Group
Oracle

JCP EC Meeting
2015/01/14

ORACLE