

Working with Modules



Sander Mak

@sander_mak www.branchandbound.net



Adding a second module

Module resolution

Readability & implied readability

**Creating a GUI module using
platform modules**

Remember: Creating Modules

```
module mymodule {  
    requires somelibrary;  
    exports mymodule.api;  
}
```

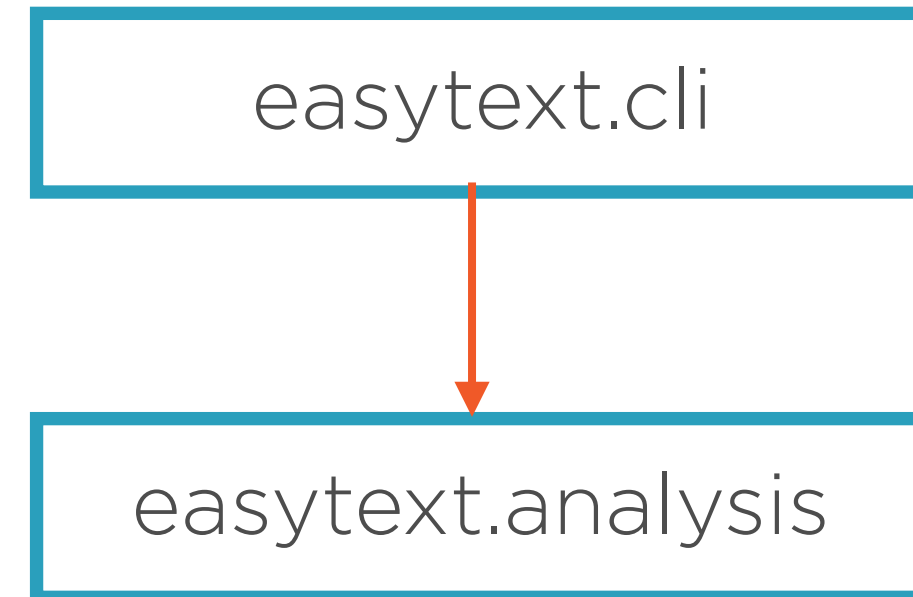
Remember: Creating Modules

```
module mymodule {  
    requires somelibrary;  
    exports mymodule.api;  
}
```

```
javac --module-source-path src ...  
java --module-path out -m easytext/..
```

Demo

Adding a second module



Module Resolution

Start with a *root* module:

easytext.cli

Resolved modules

Module Resolution

Resolved modules

easytext.cli

Module Resolution

Find its dependencies:

Resolved modules

easytext.cli

Module Resolution

Find its dependencies:

easytext.analysis

java.base

Resolved modules

easytext.cli

Module Resolution

Resolved modules

easytext.cli

easytext.analysis

java.base

Module Resolution

Find their dependencies: **no dependencies, done!**

Resolved modules

easytext.cli

easytext.analysis

java.base

Module Resolution

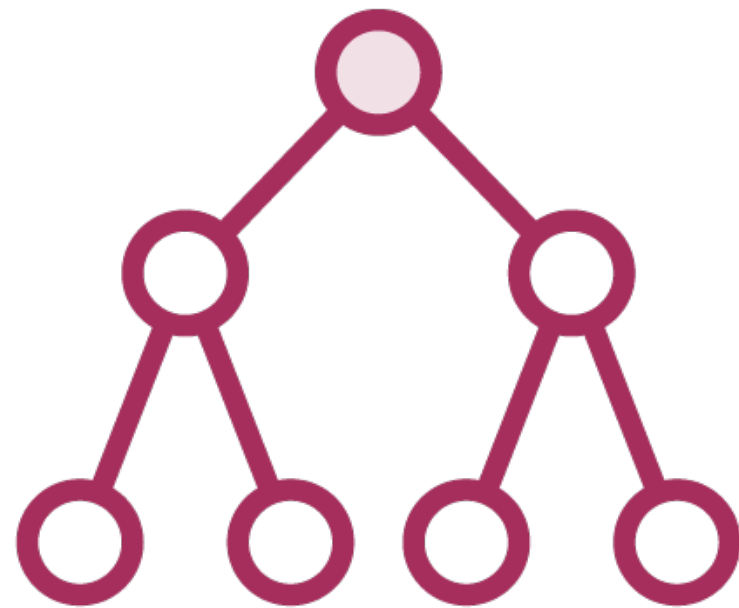
Resolved modules

easytext.cli

easytext.analysis

java.base

Module Resolution



Checks module path for correctness

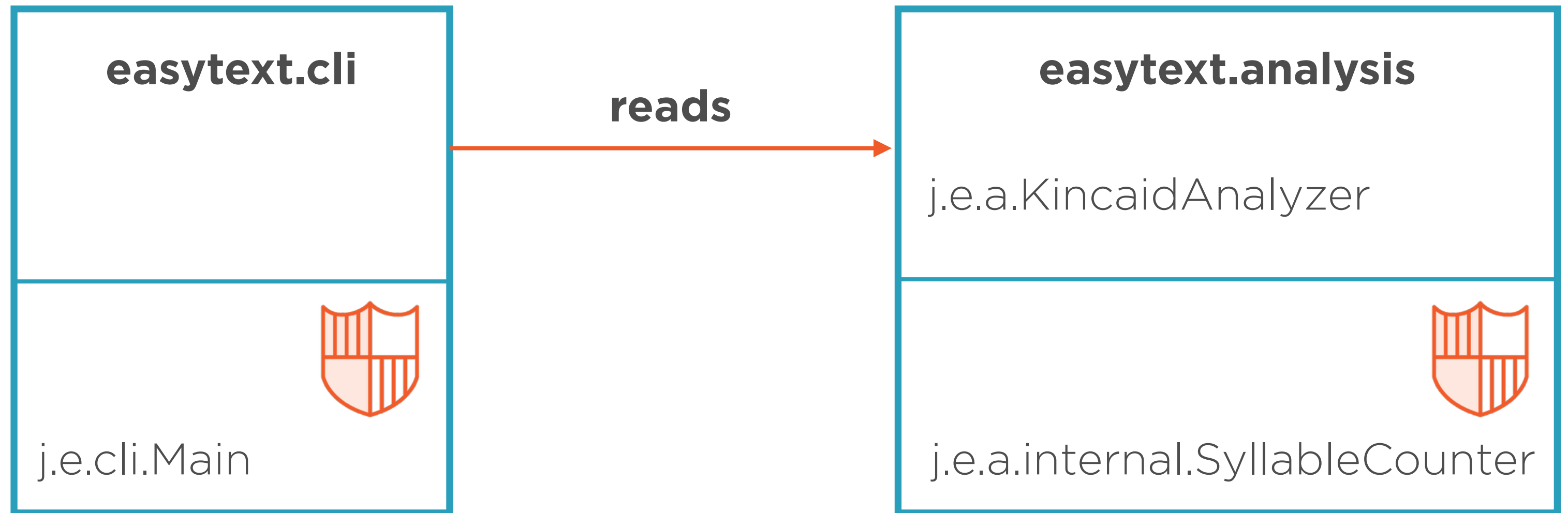
Compile & run-time

For application and platform modules

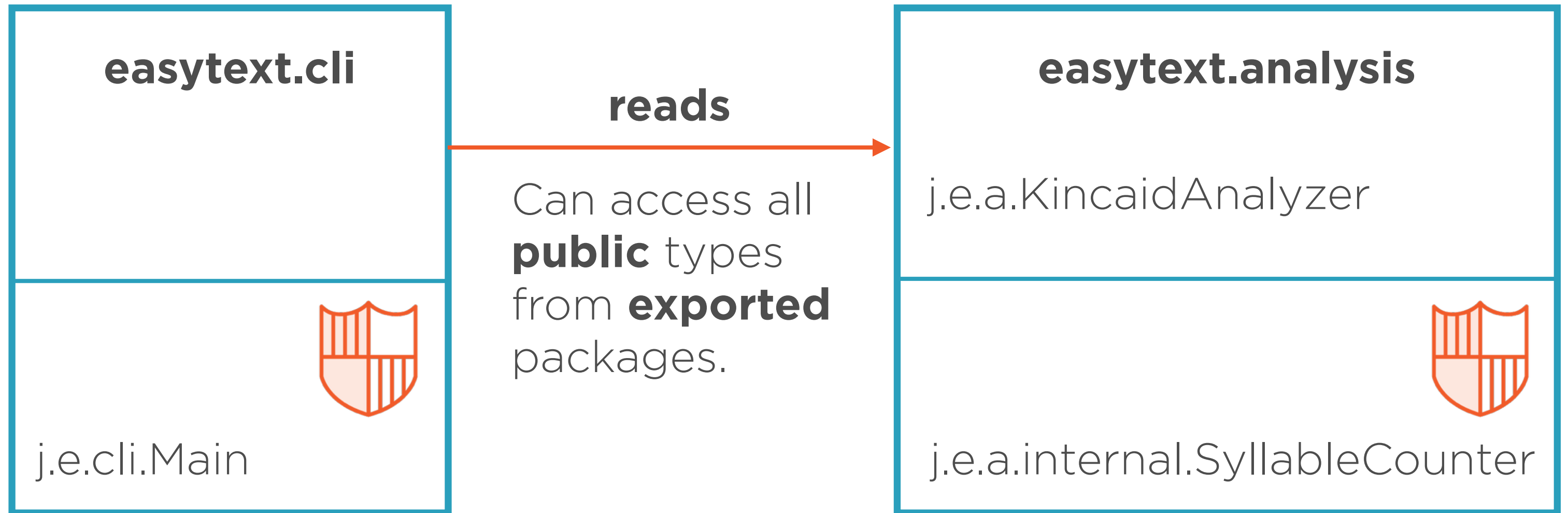
Rejects multiple versions of module

Multiple root modules possible

Readability



Readability



Wait, public no longer
means public?

Readability Is Not Enough



**Do you read
the module?**

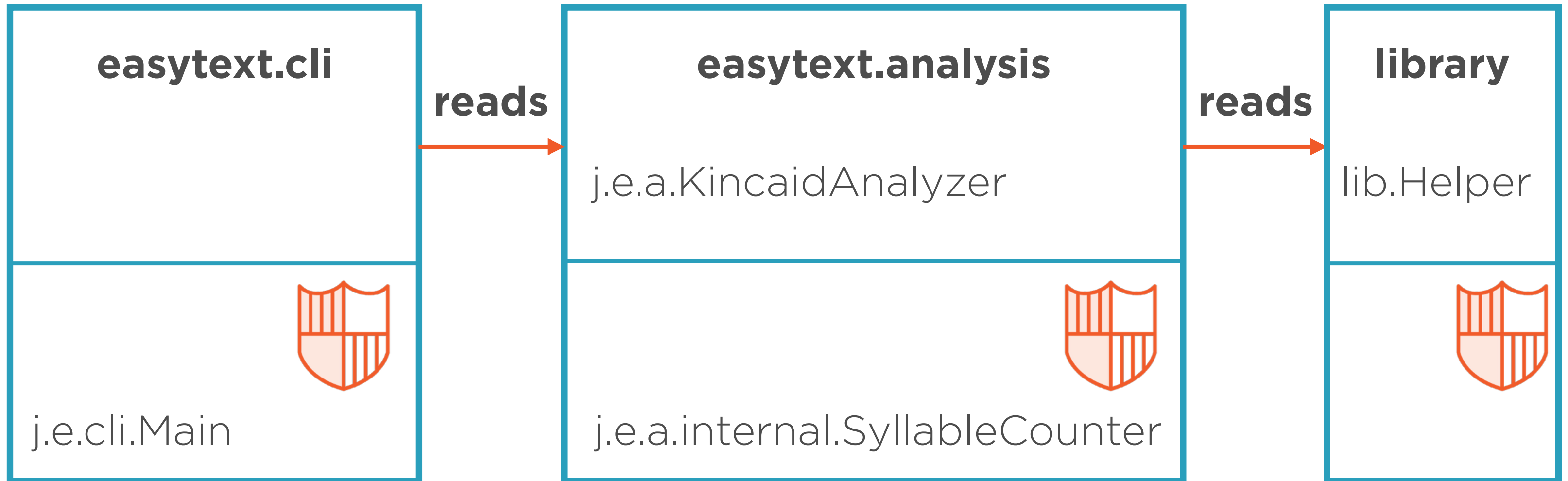


**Is the package
exported?**

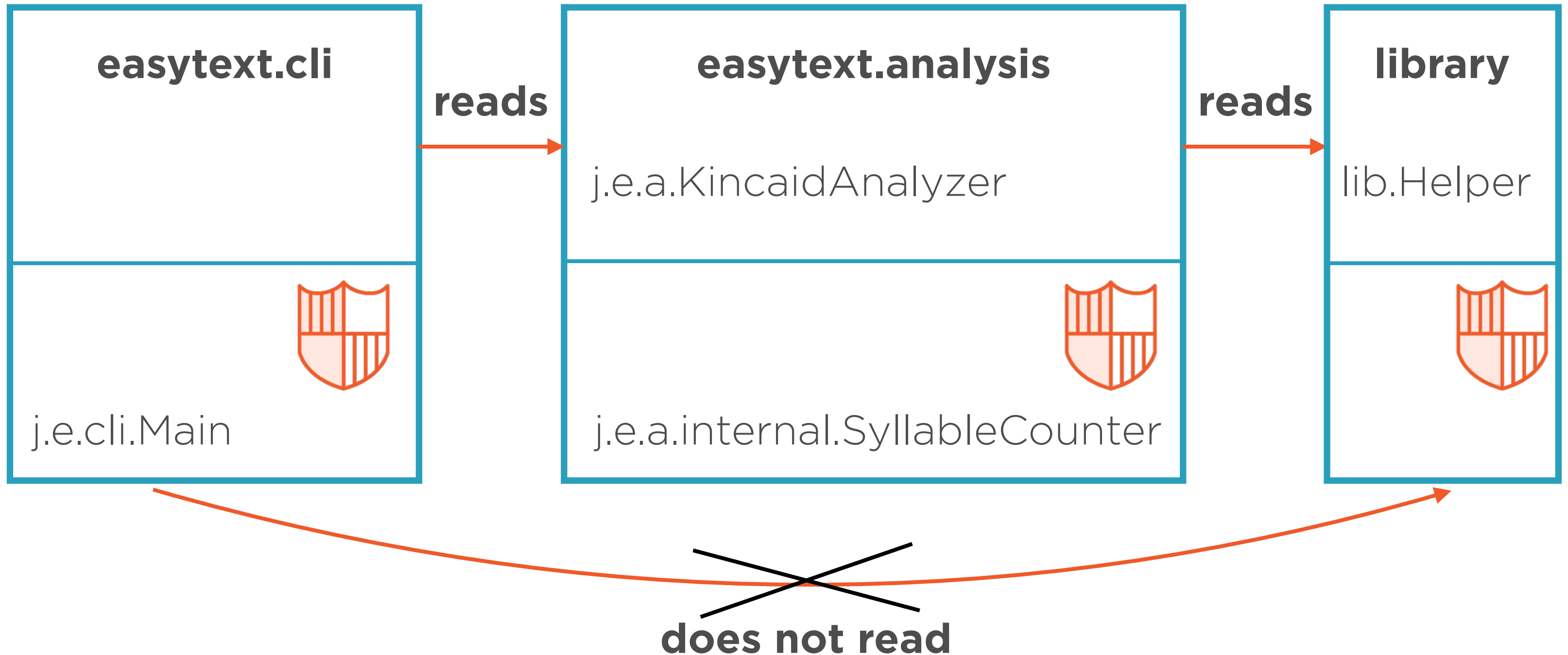


**Is the type
public?**

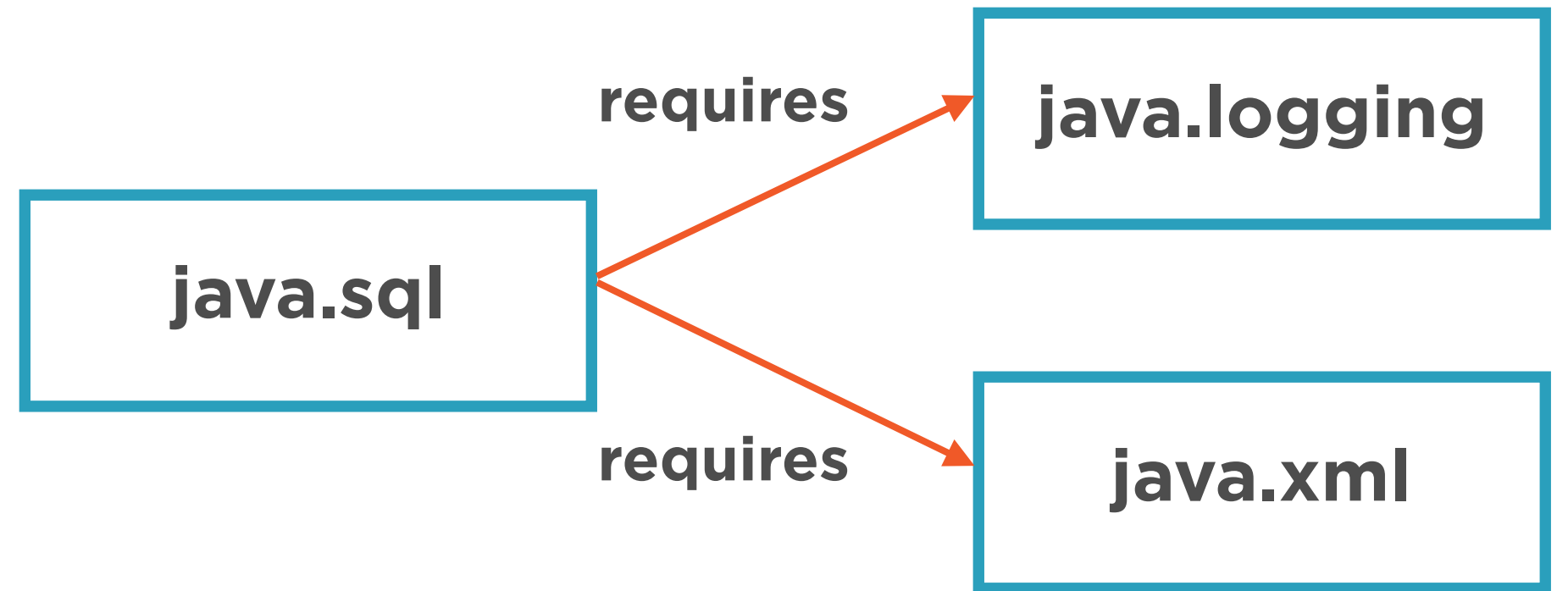
Readability Is Not Transitive



Readability Is Not Transitive



Implied Readability

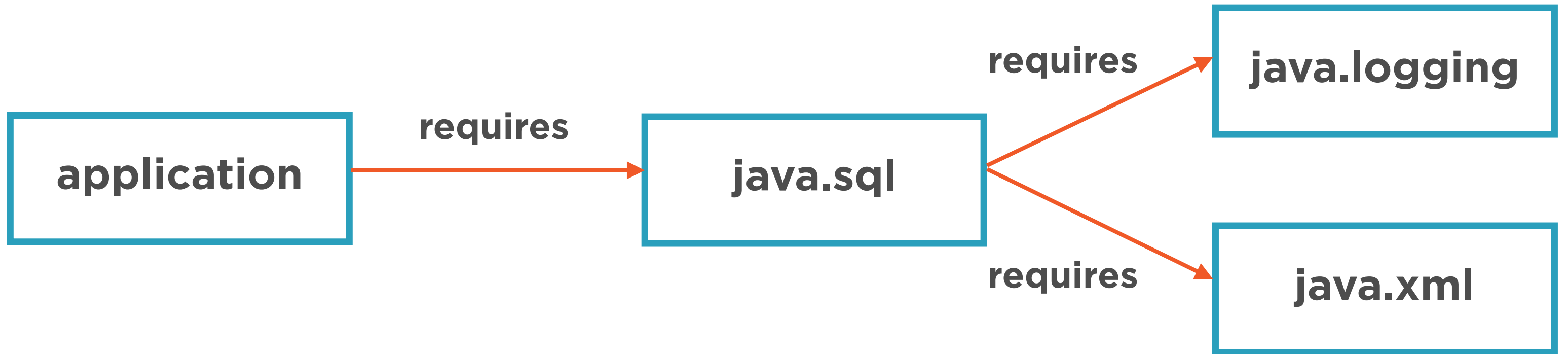


```
package java.sql;
```

```
import java.util.logging.Logger;
```

```
public interface Driver {  
    public Logger getParentLogger();  
}
```

Implied Readability

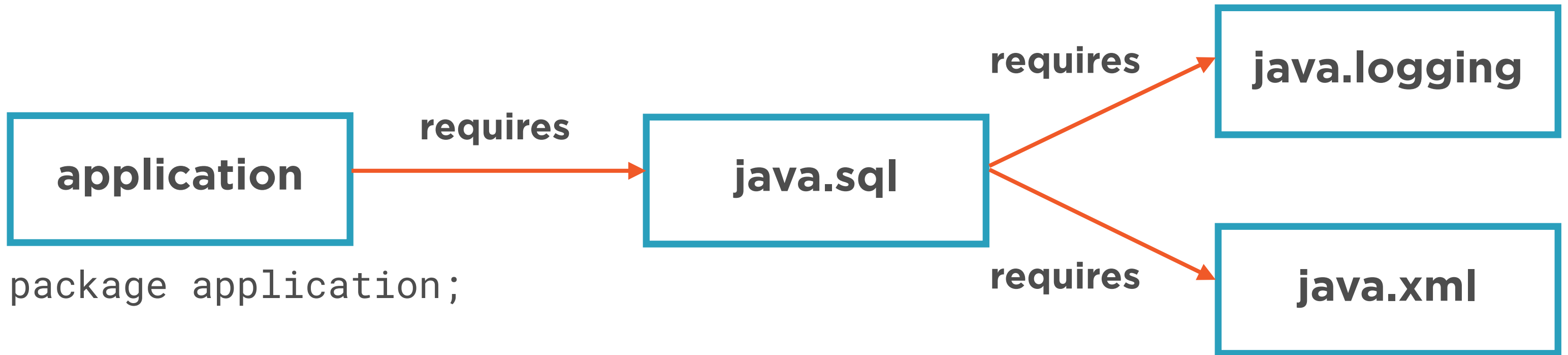


```
package java.sql;
```

```
import java.util.logging.Logger;
```

```
public interface Driver {  
    public Logger getParentLogger();  
}
```

Implied Readability

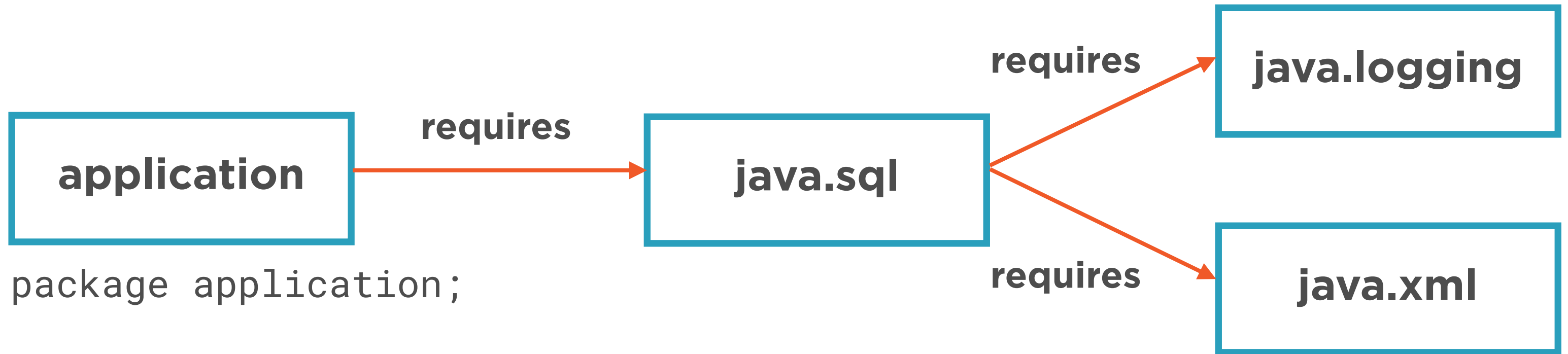


```
package application;
```

```
import java.sql.Driver;
```

```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```

Implied Readability



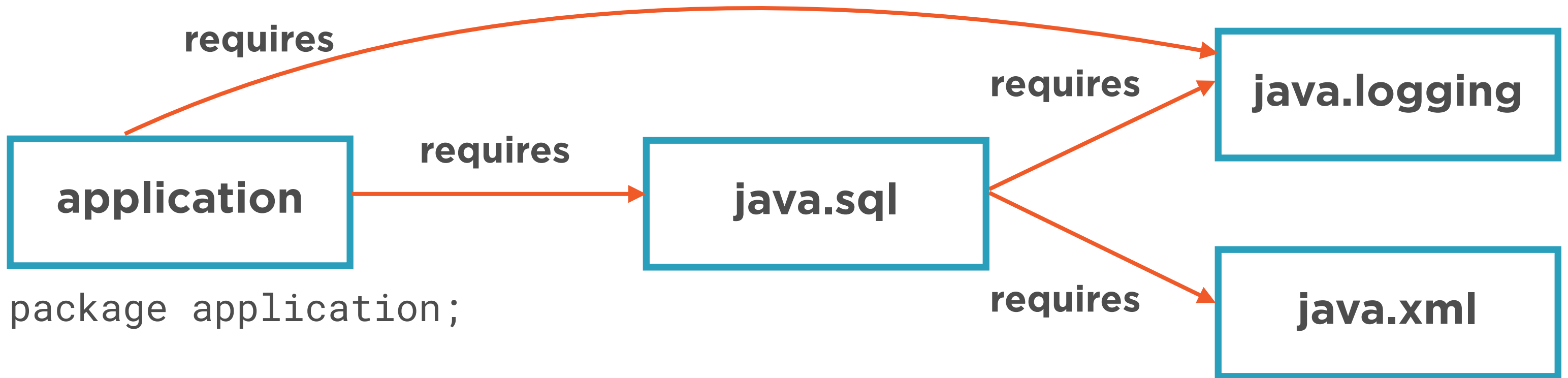
```
package application;
```

```
import java.sql.Driver;
```

```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```

**error: warning() in Logger is defined
in an inaccessible class or interface**

Implied Readability

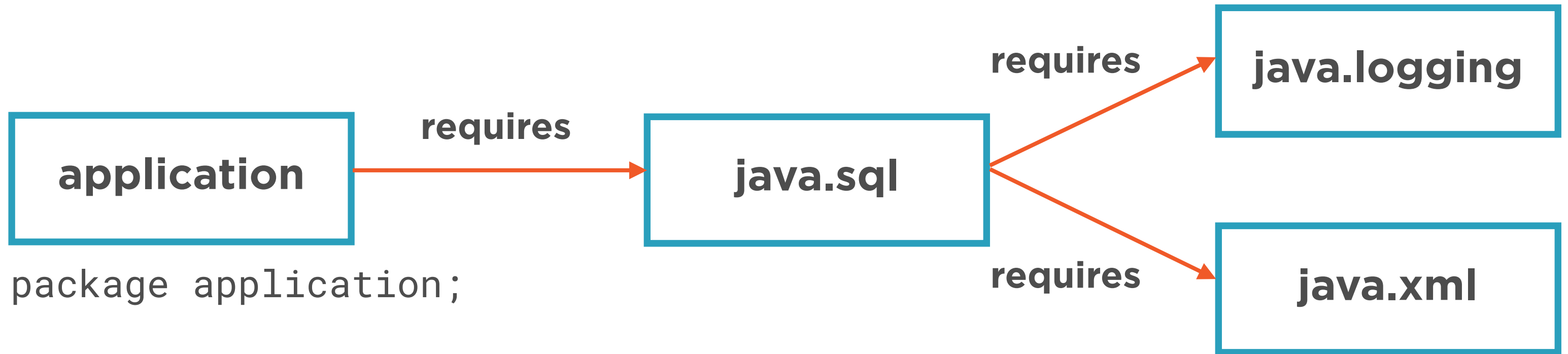


```
package application;
```

```
import java.sql.Driver;
```

```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```


Implied Readability

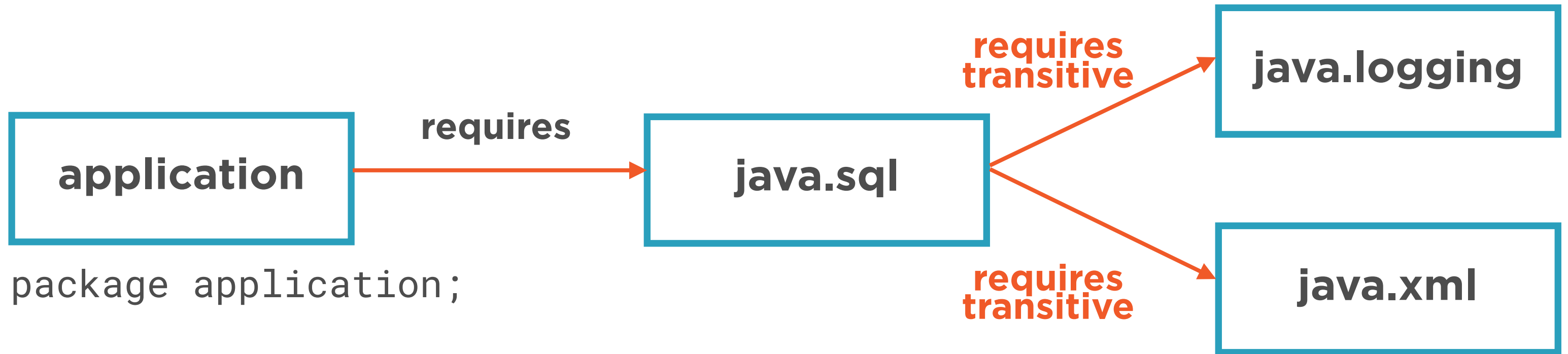


```
package application;
```

```
import java.sql.Driver;
```

```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```

Implied Readability

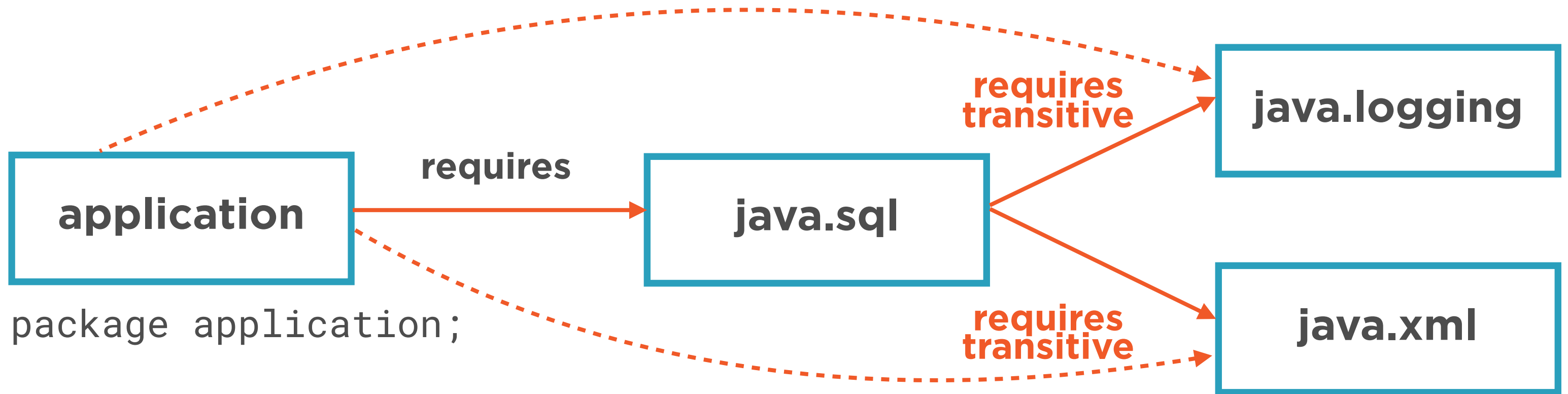


```
package application;
```

```
import java.sql.Driver;
```

```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```

Implied Readability

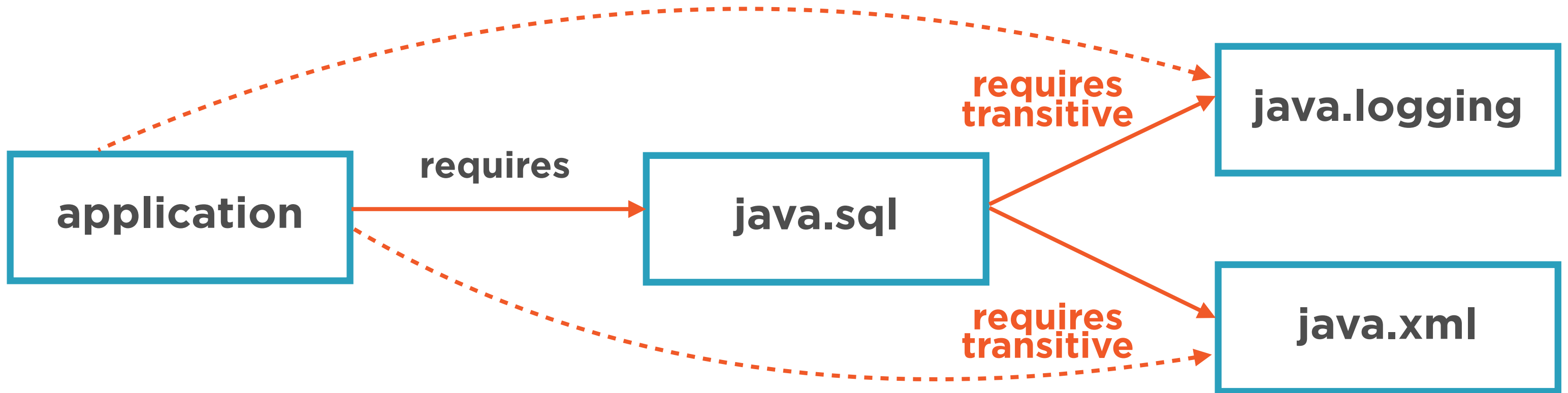


```
package application;
```

```
import java.sql.Driver;
```

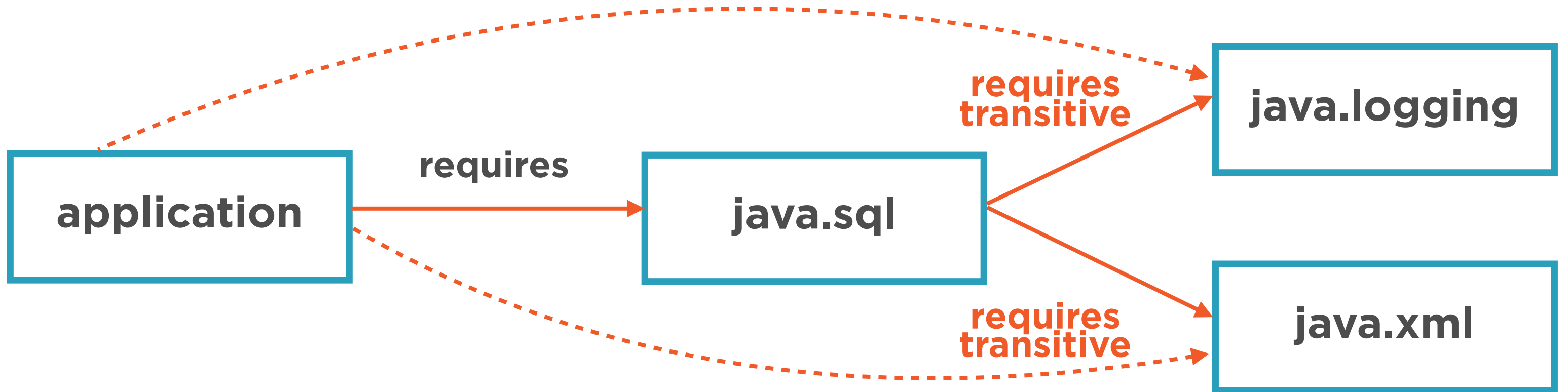
```
public class App {  
    void start(Driver driver) {  
        driver.getParentLogger()  
            .warning("!")  
    }  
}
```

Implied Readability



```
module application {  
  requires java.sql;  
}
```

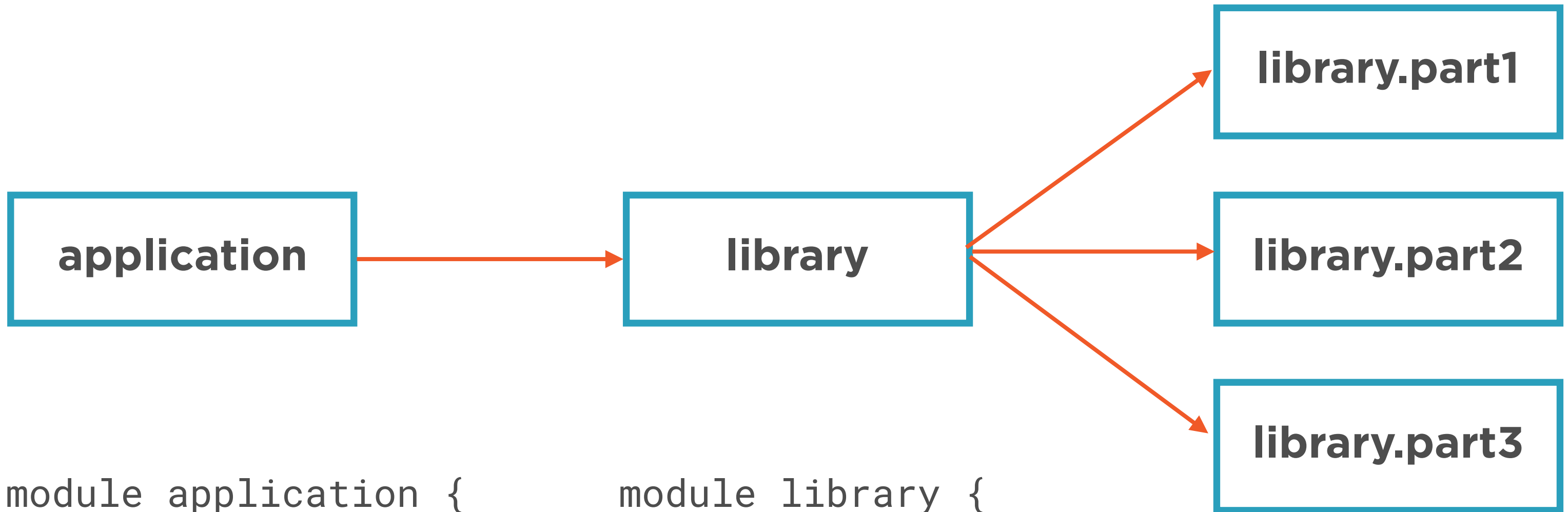
Implied Readability



```
module application {  
  requires java.sql;  
}
```

```
module java.sql {  
  requires transitive java.logging;  
  requires transitive java.xml;  
  exports java.sql;  
  ..  
}
```

Aggregator Modules



```
module application {  
  requires library;  
}
```

```
module library {  
  requires transitive library.part1;  
  requires transitive library.part2;  
  requires transitive library.part3;  
}
```

Platform Modules

Core Java SE

java.base

java.desktop

java.sql

java.se

Enterprise

java.corba

java.xml.ws

java.transaction

JavaFX

javafx.base

javafx.web

javafx.controls

JDK

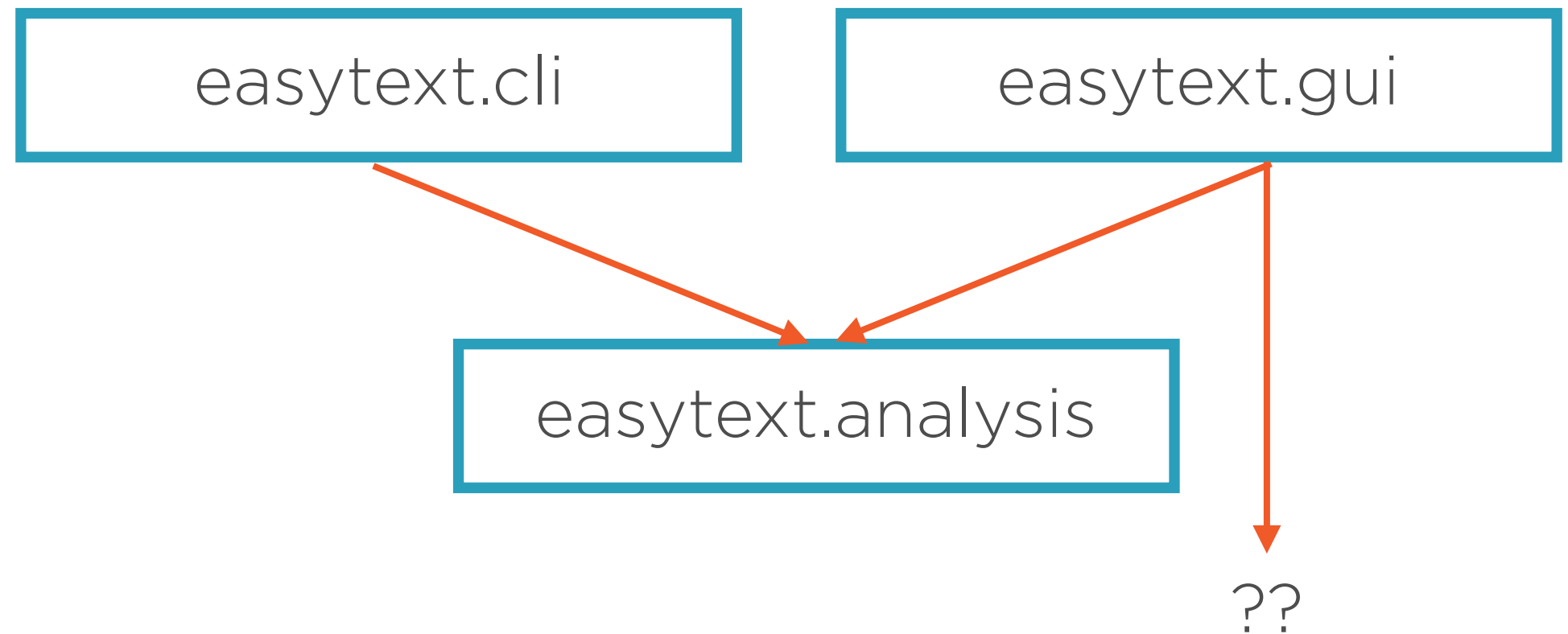
jdk.compiler

jdk.httpserver

jdk.jartool

Demo

Adding a GUI module



Summary

easytext.cli



easytext.analysis

Summary

Resolved modules

easytext.cli

easytext.analysis

java.base

Summary



**Do you read
the module?**



**Is the package
exported?**



**Is the type
public?**

Summary

