

Creating Modules



Sander Mak

@sander_mak www.branchandbound.net





The Modular JDK



The Modular JDK

Java Module System Concepts



The Modular JDK

Java Module System Concepts

Introducing EasyText



The Modular JDK

Java Module System Concepts

Introducing EasyText

Build our first module!



The Modular JDK

Java Module System Concepts

Introducing EasyText

Build our first module!



The Modular JDK

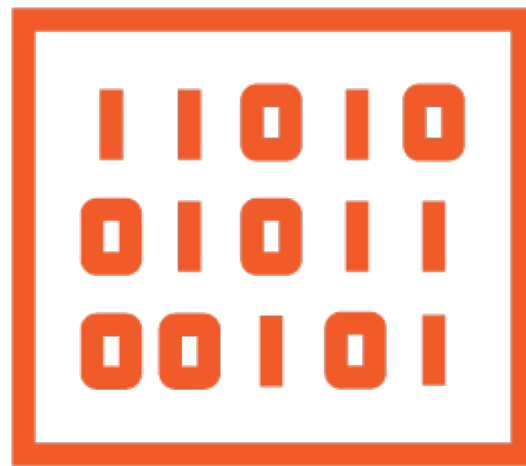
Java Module System Concepts

Introducing EasyText

Build our first module!

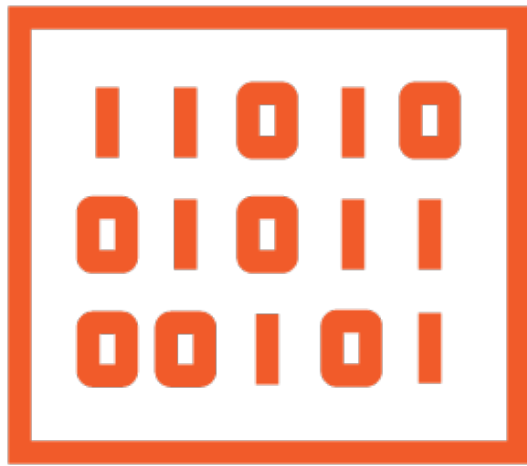
Remember: Three Tenets of Modularity

Remember: Three Tenets of Modularity



Strong Encapsulation

Remember: Three Tenets of Modularity

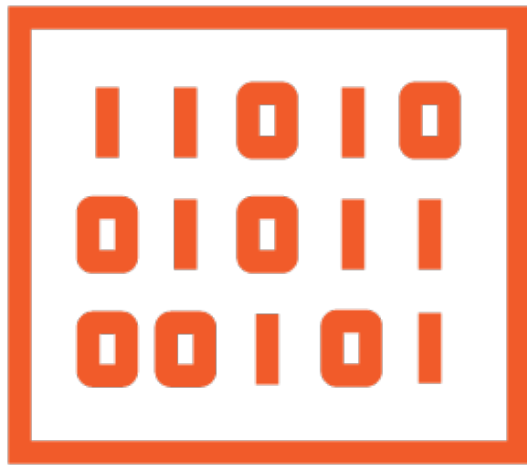


Strong Encapsulation



Well-defined Interfaces

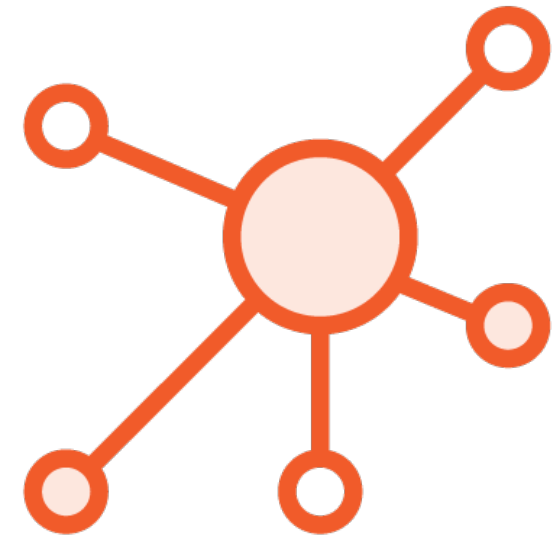
Remember: Three Tenets of Modularity



Strong Encapsulation

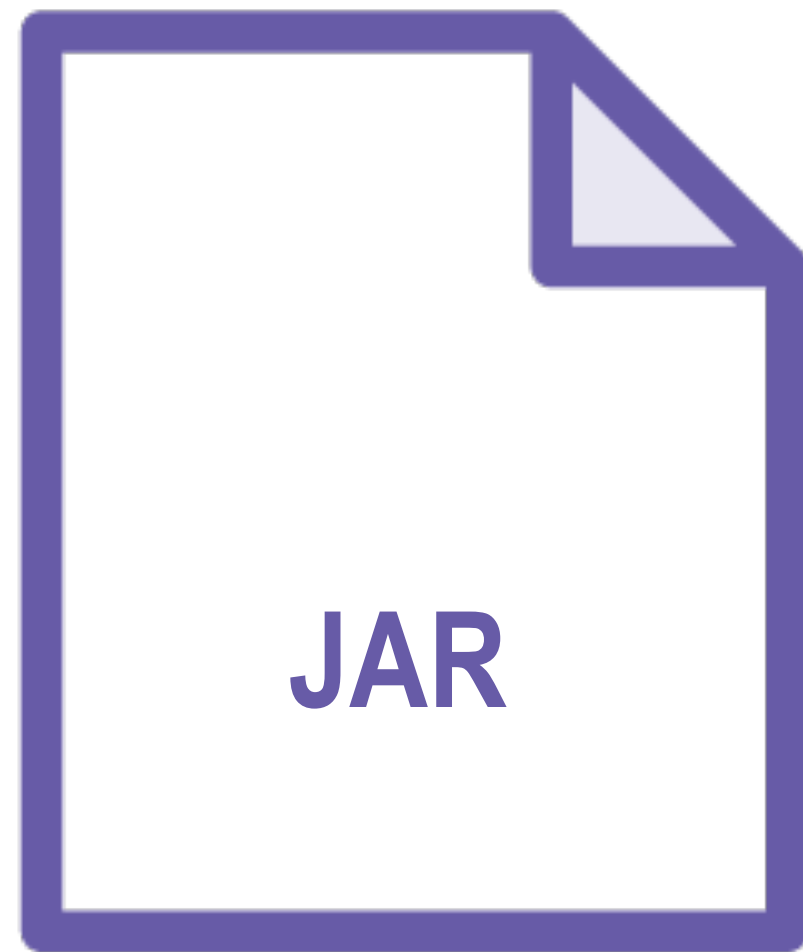


Well-defined Interfaces



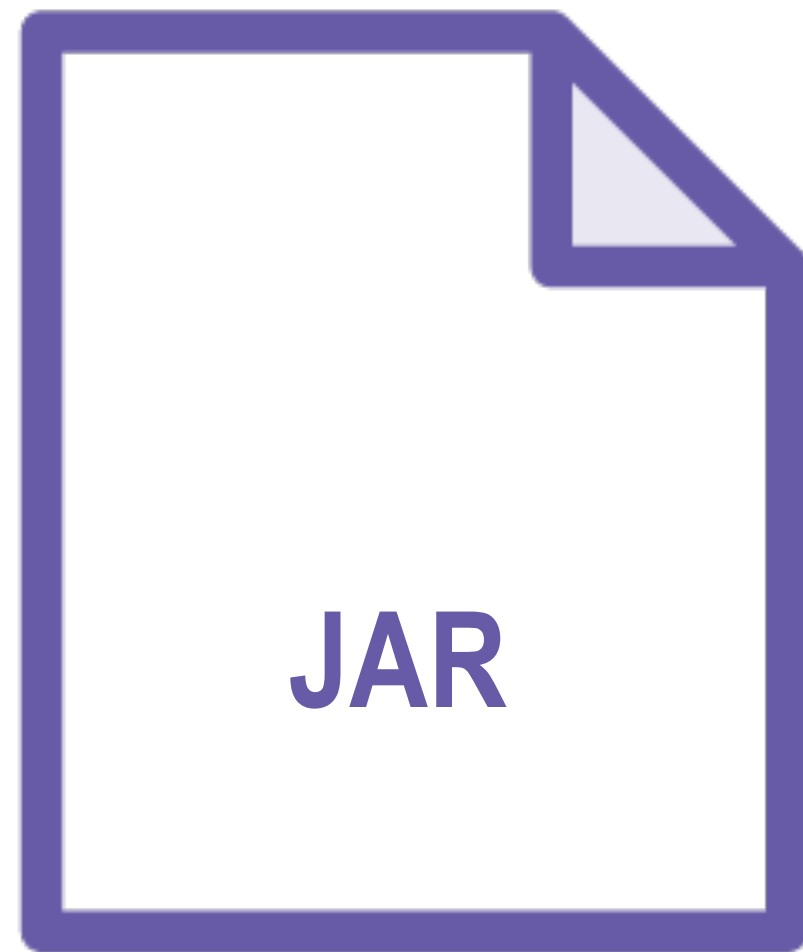
Explicit Dependencies

Before the Modular JDK



rt.jar

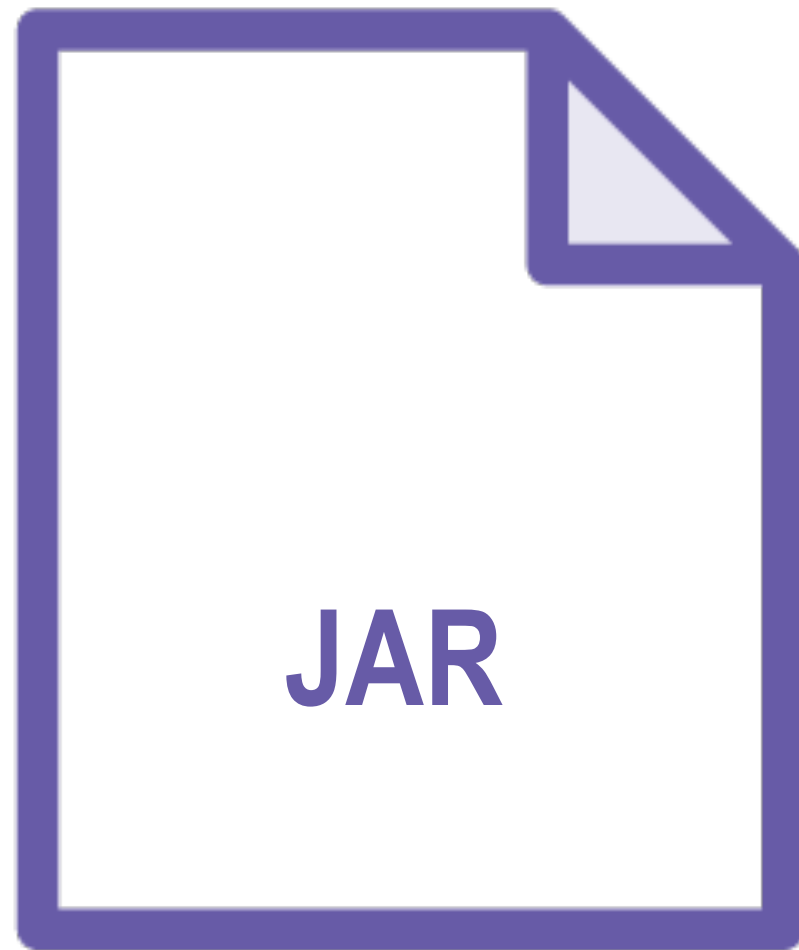
Before the Modular JDK



rt.jar

One huge library

Before the Modular JDK

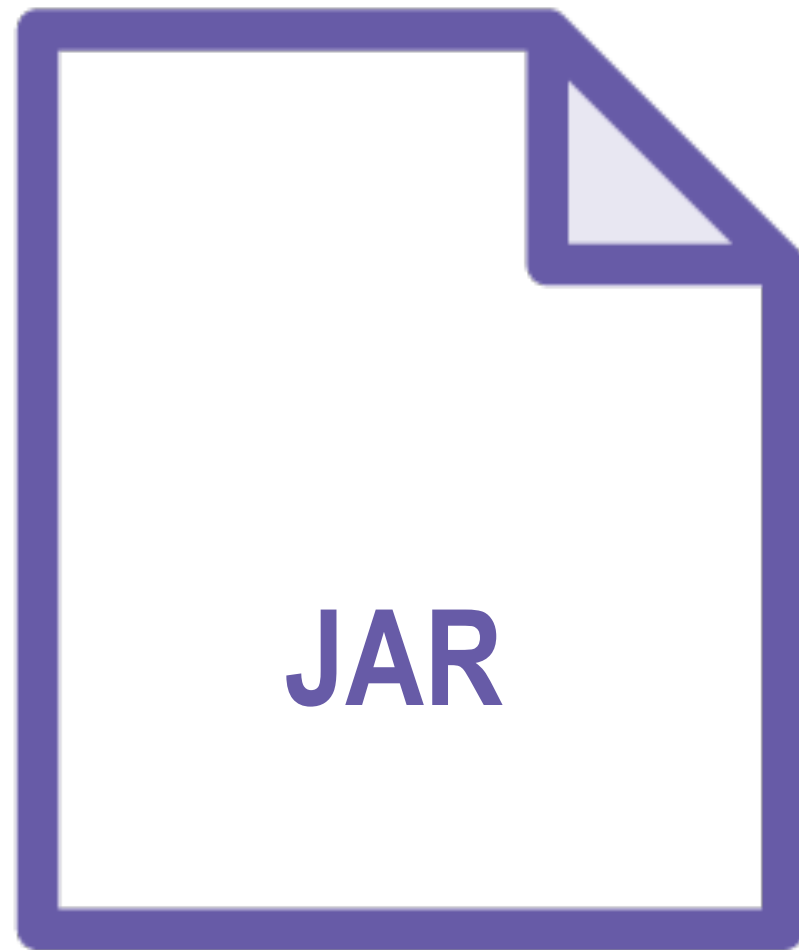


rt.jar

One huge library

Many entangled classes

Before the Modular JDK



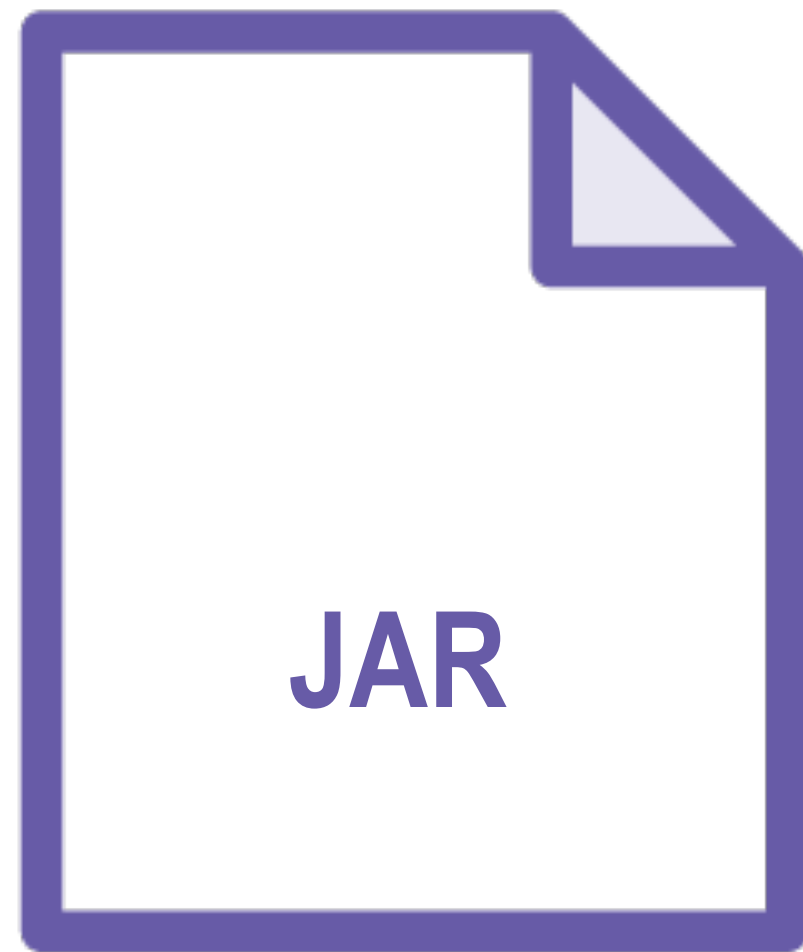
rt.jar

One huge library

Many entangled classes

Hard to evolve

Before the Modular JDK



rt.jar

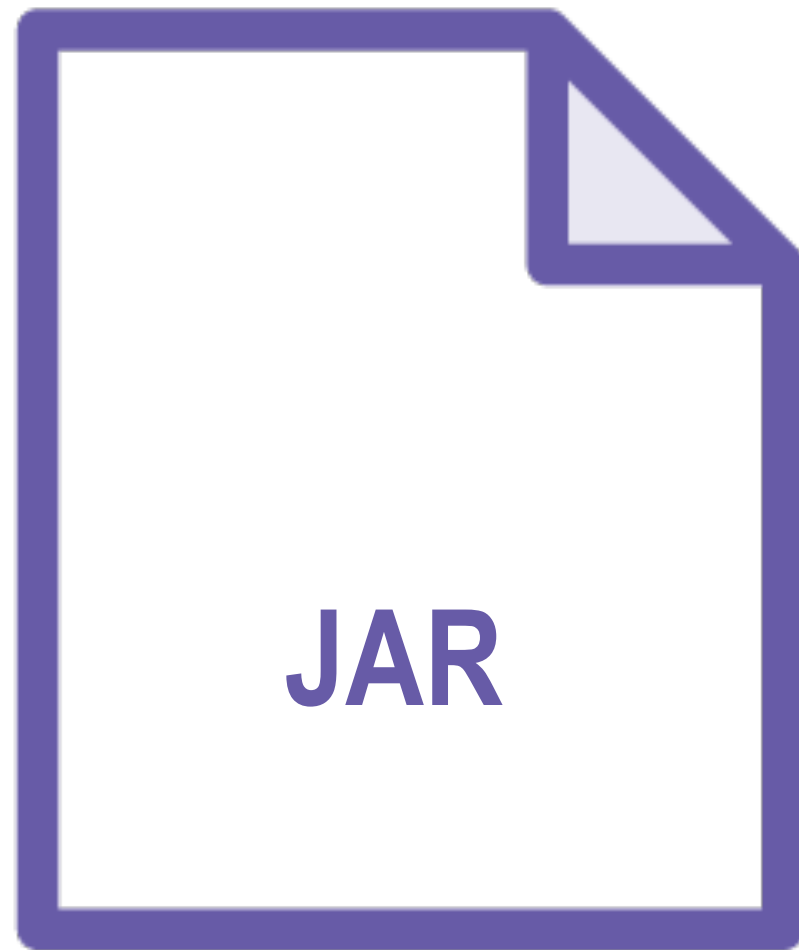
One huge library

Many entangled classes

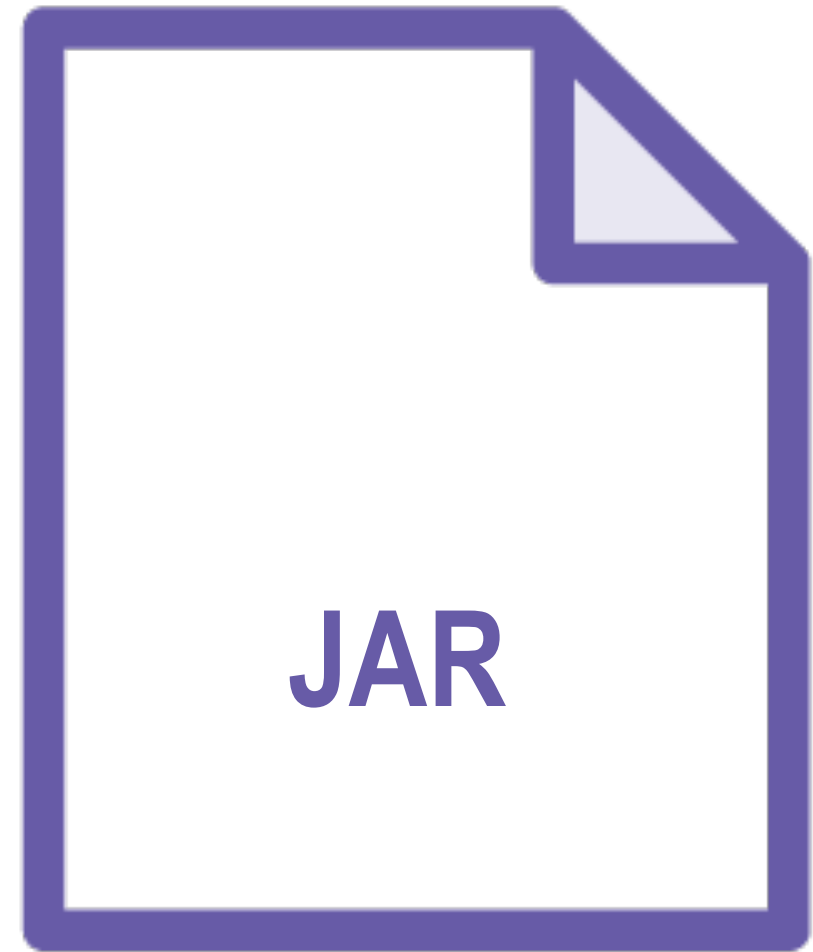
Hard to evolve

Restricted by backwards compatibility

Before the Modular JDK

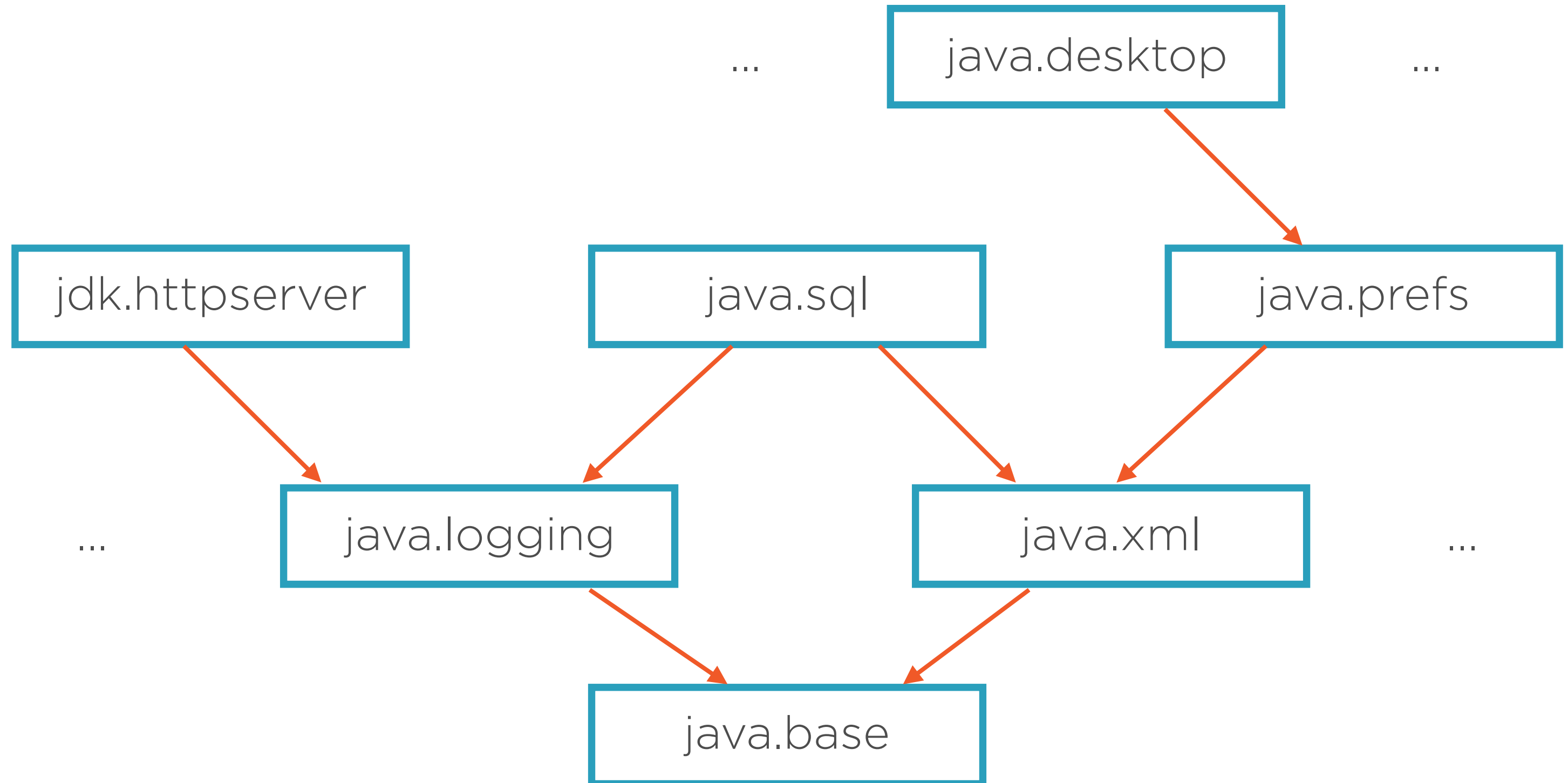


rt.jar



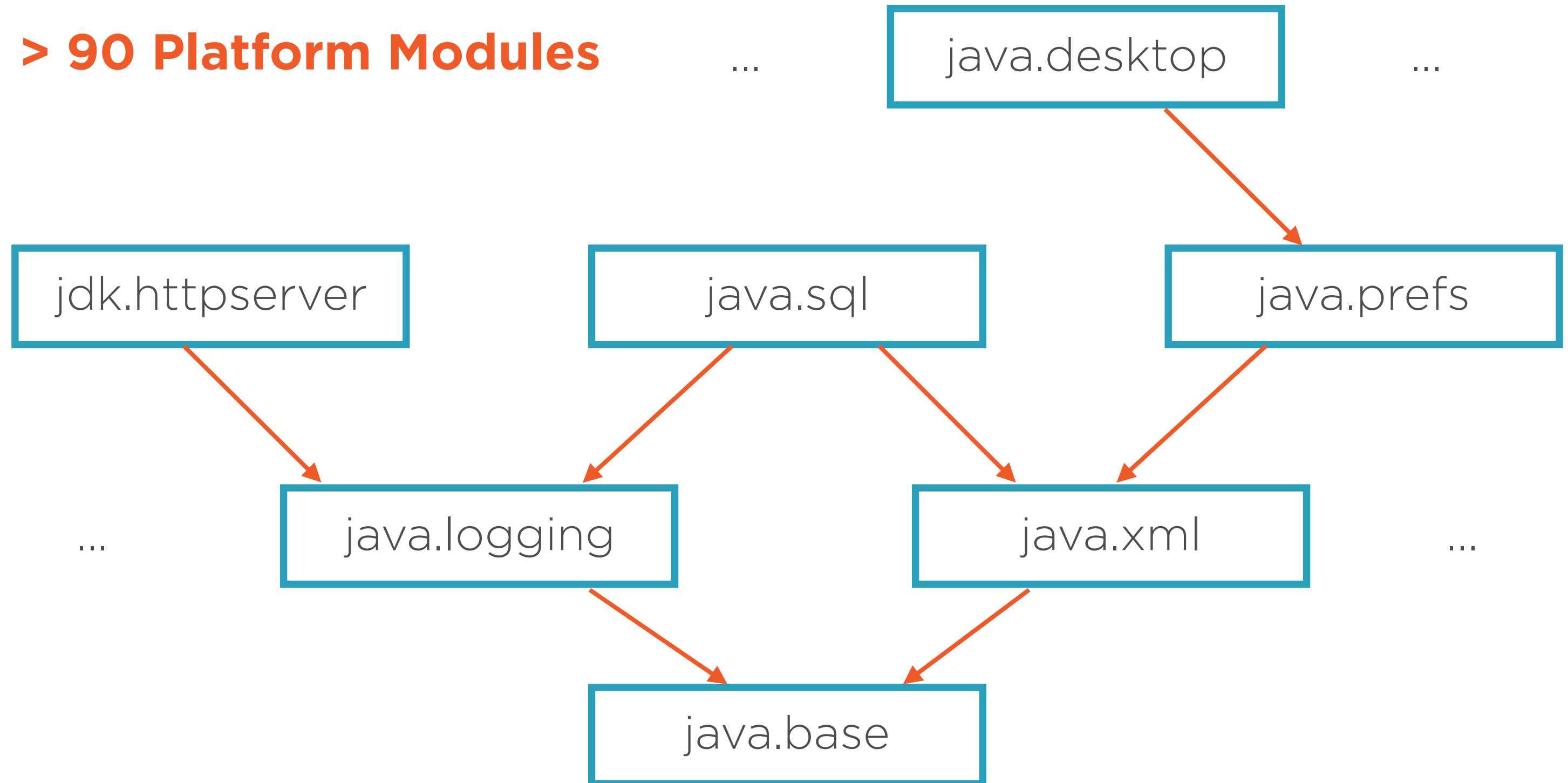
Your application?

The Modular JDK: Explicit Dependencies



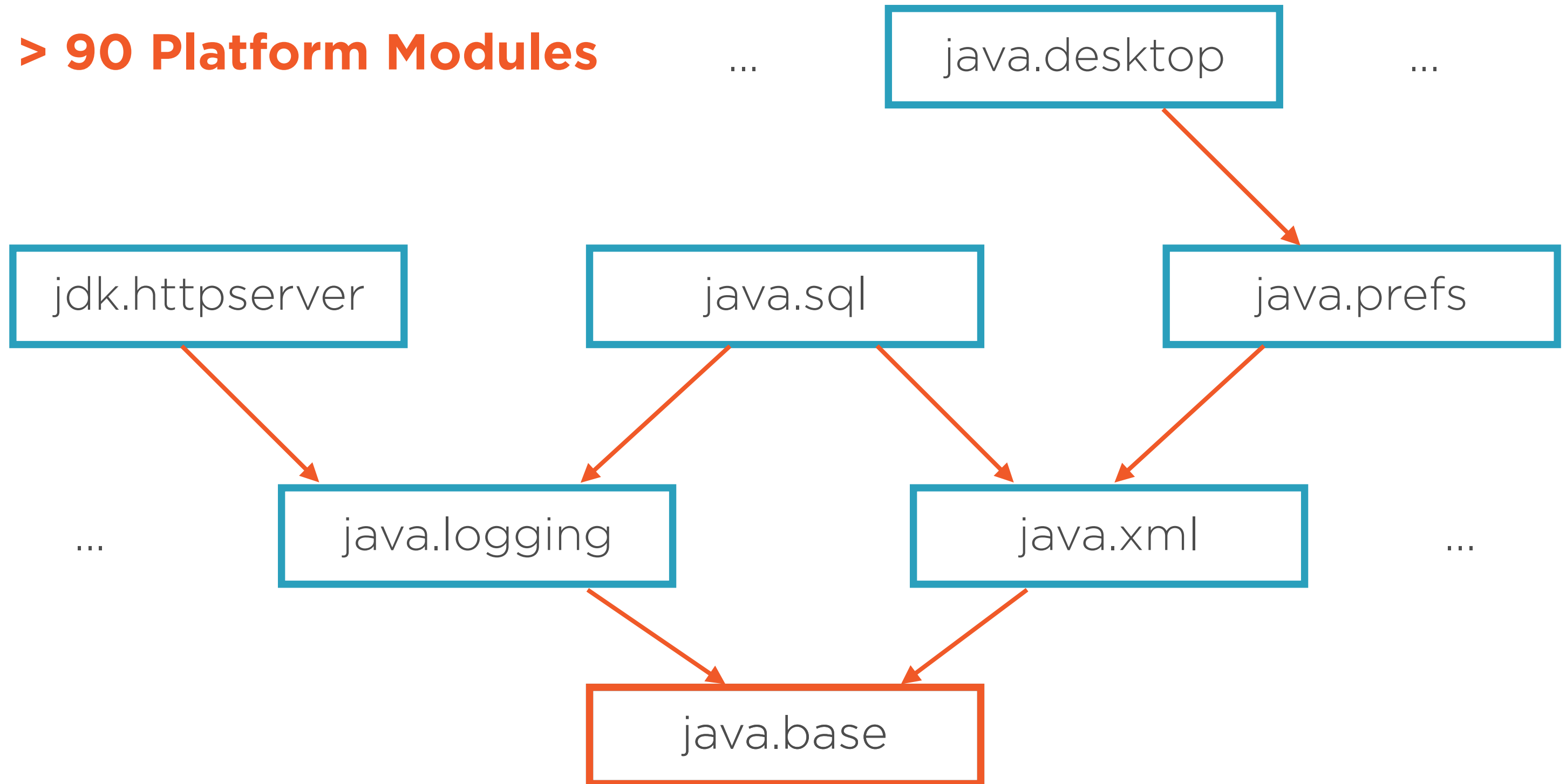
The Modular JDK: Explicit Dependencies

> **90 Platform Modules**



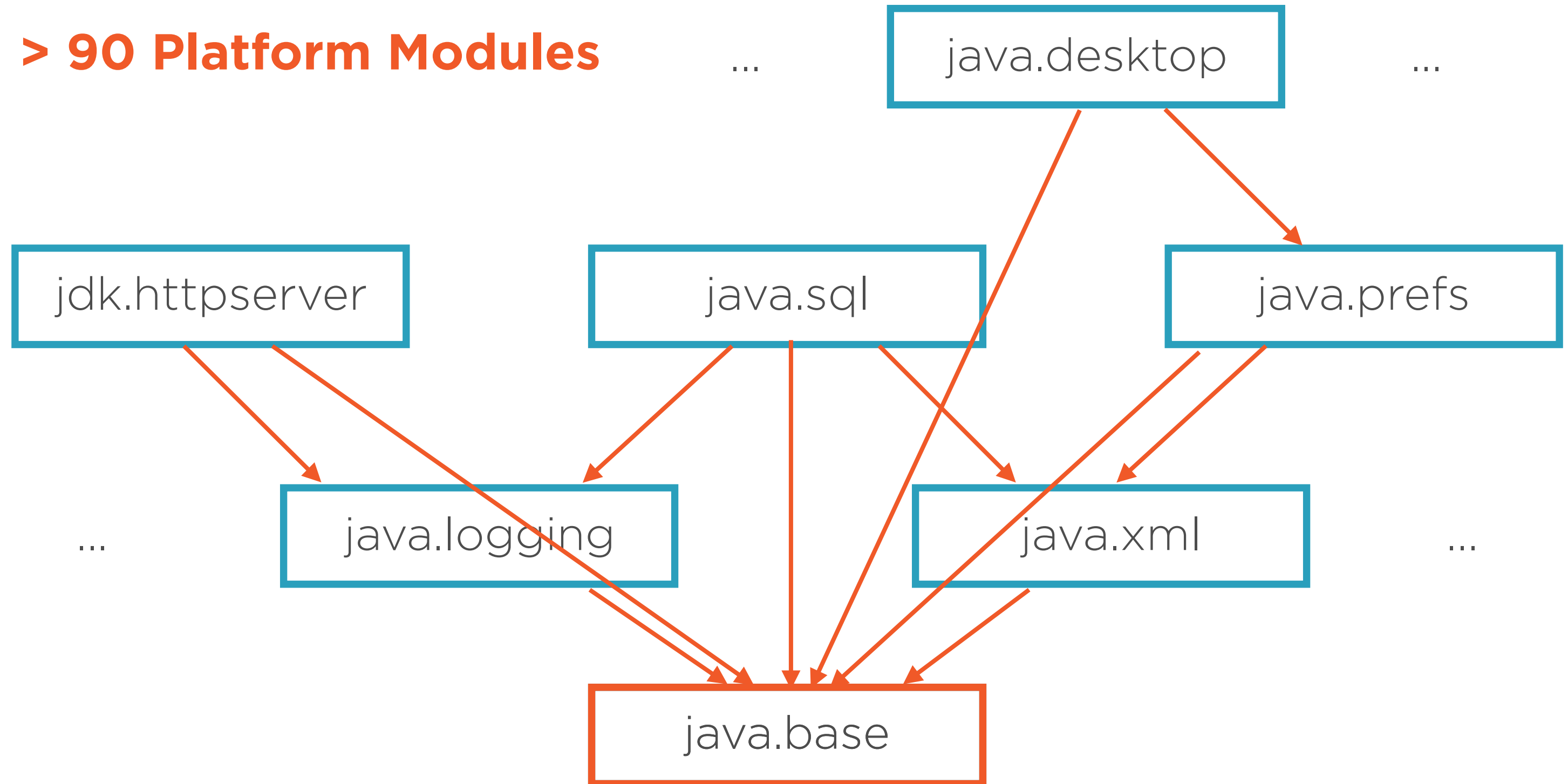
The Modular JDK: Explicit Dependencies

> **90 Platform Modules**



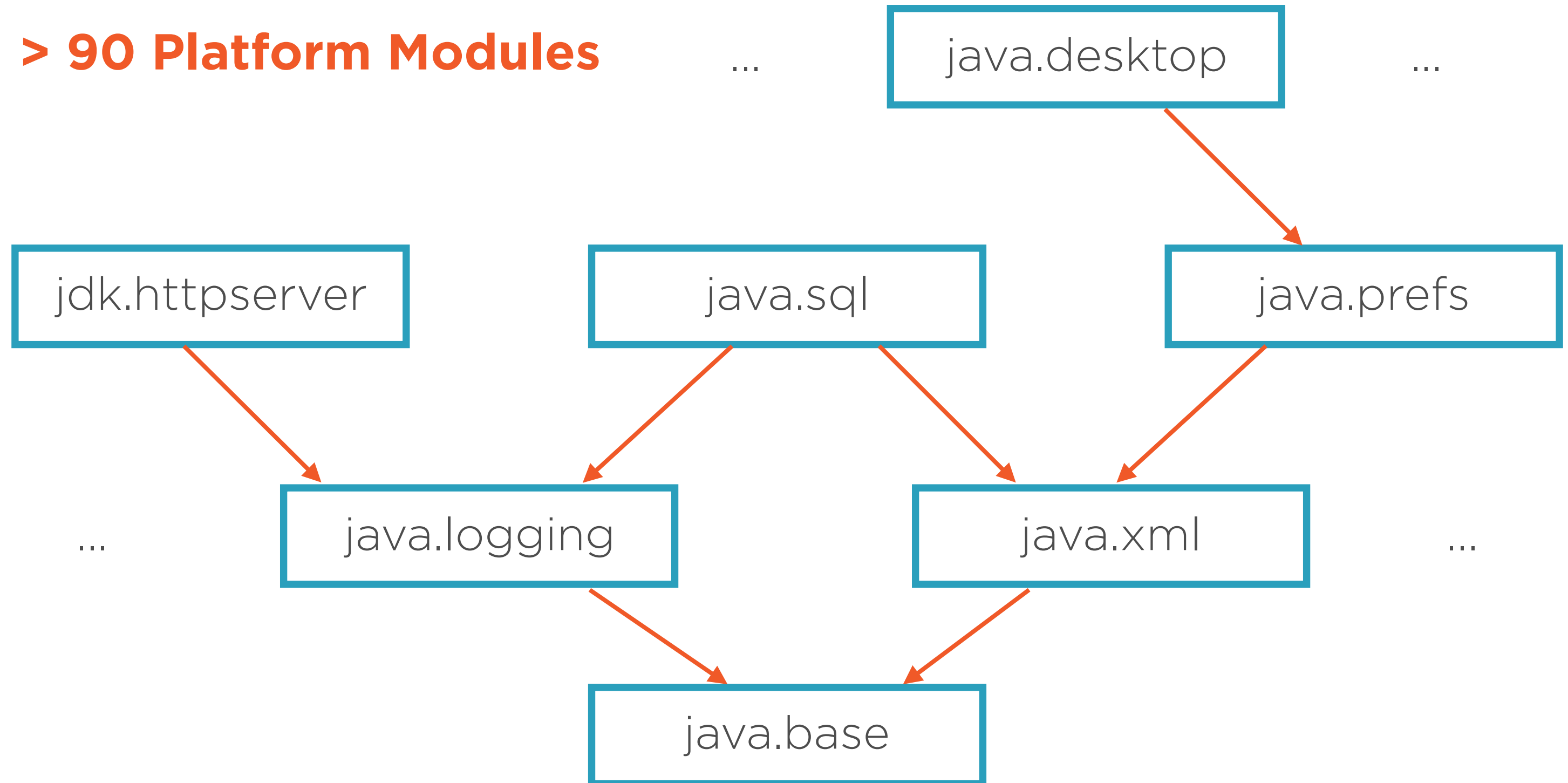
The Modular JDK: Explicit Dependencies

> 90 Platform Modules



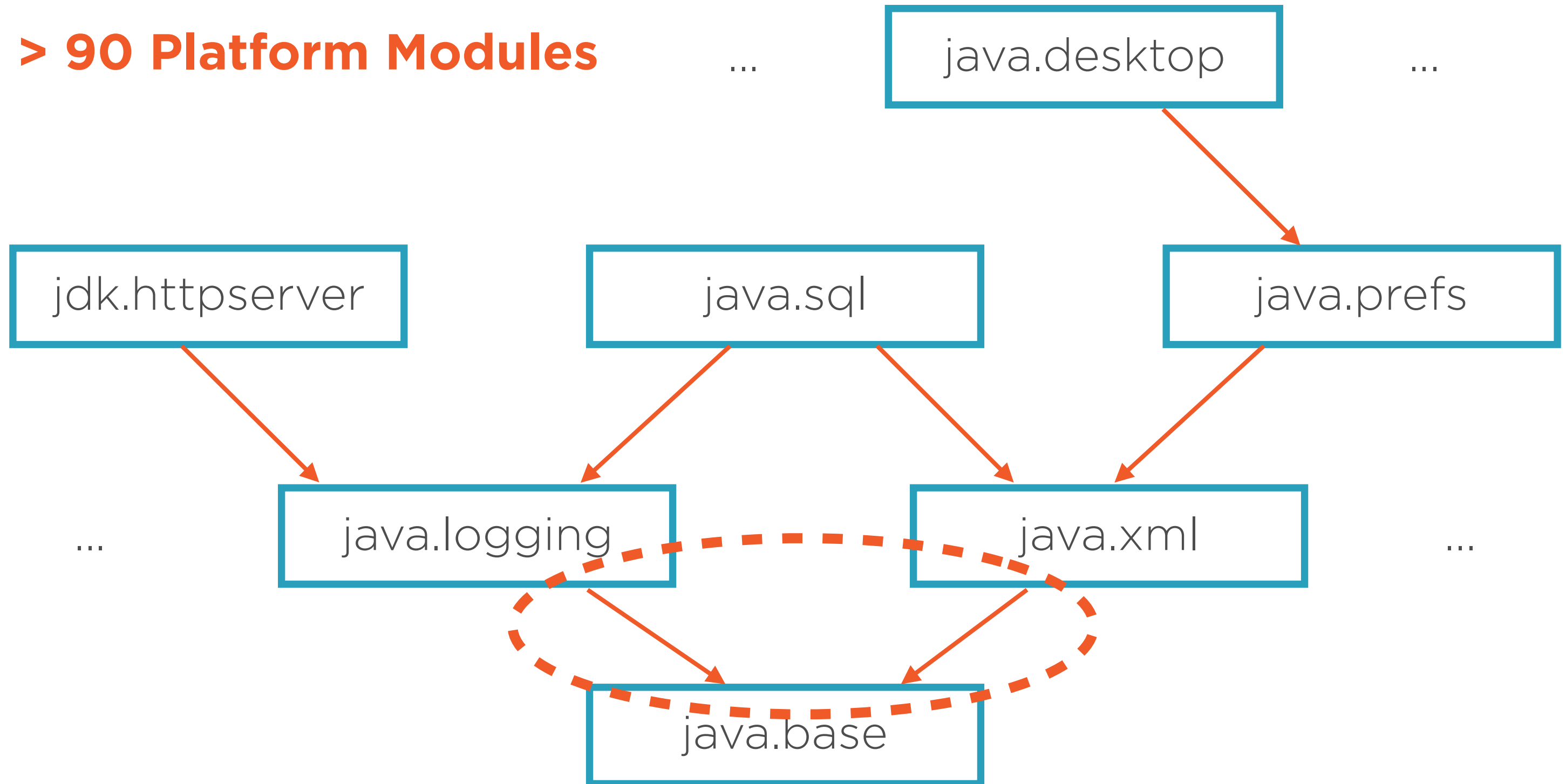
The Modular JDK: Explicit Dependencies

> **90 Platform Modules**



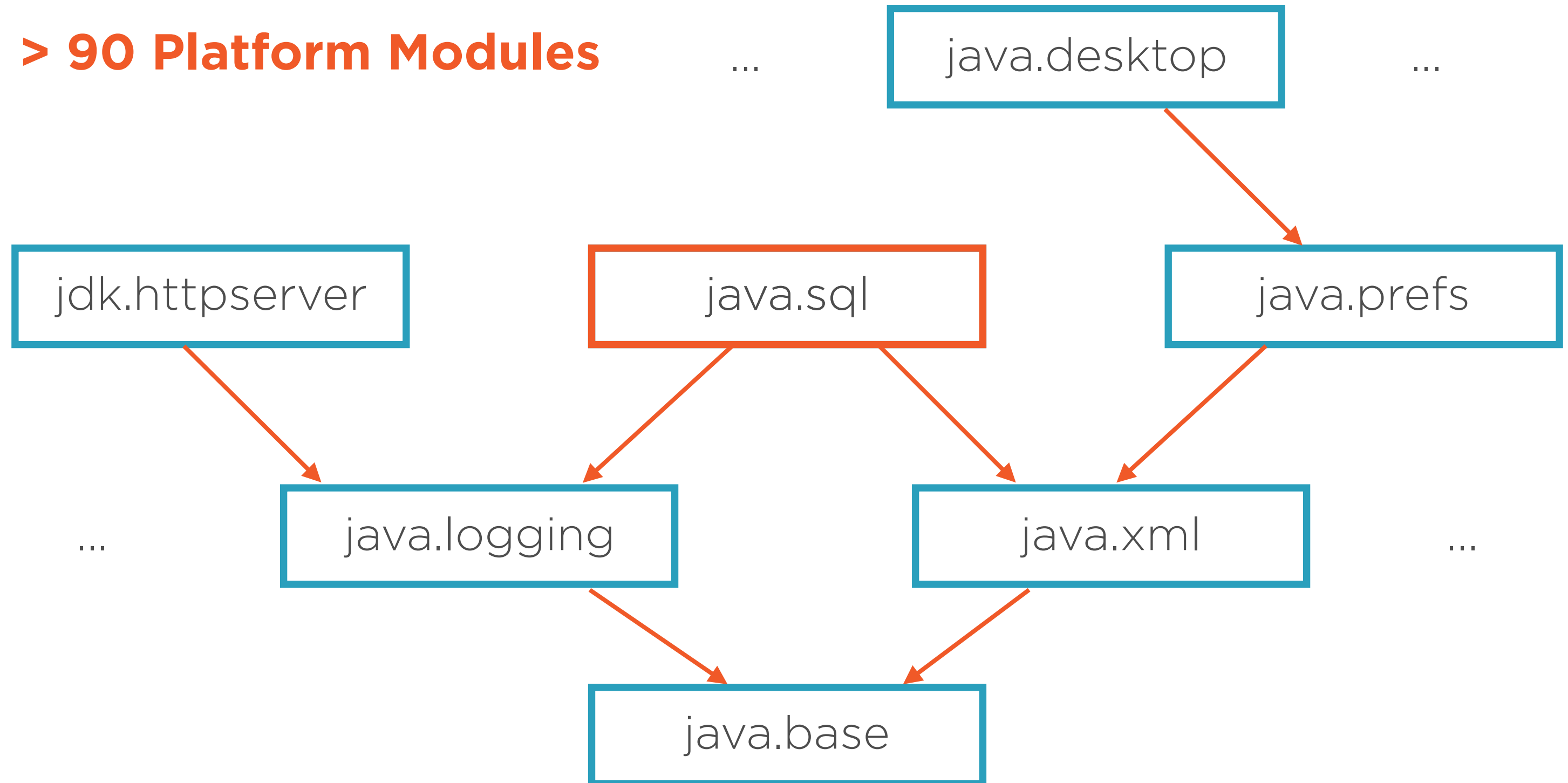
The Modular JDK: Explicit Dependencies

> **90 Platform Modules**



The Modular JDK: Explicit Dependencies

> **90 Platform Modules**



The Modular JDK: Encapsulation



The Modular JDK: Encapsulation



Module Descriptors

java.base	
java.lang java.util java.io ...	
sun.util jdk.internal ...	

module-info.java

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

Module Descriptors

java.base	
java.lang java.util java.io ...	
sun.util jdk.internal ...	

module-info.java

Module Descriptors

```
module java.base {
```

```
    exports java.lang;  
    exports java.util;  
    exports java.io;  
    // and more
```

```
}
```

java.base

java.lang
java.util
java.io
...

sun.util
jdk.internal
...



module-info.java

```
module java.base {
```

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

```
    // and more
```

```
}
```

Module Descriptors

java.base

java.lang
java.util
java.io

...

sun.util
jdk.internal

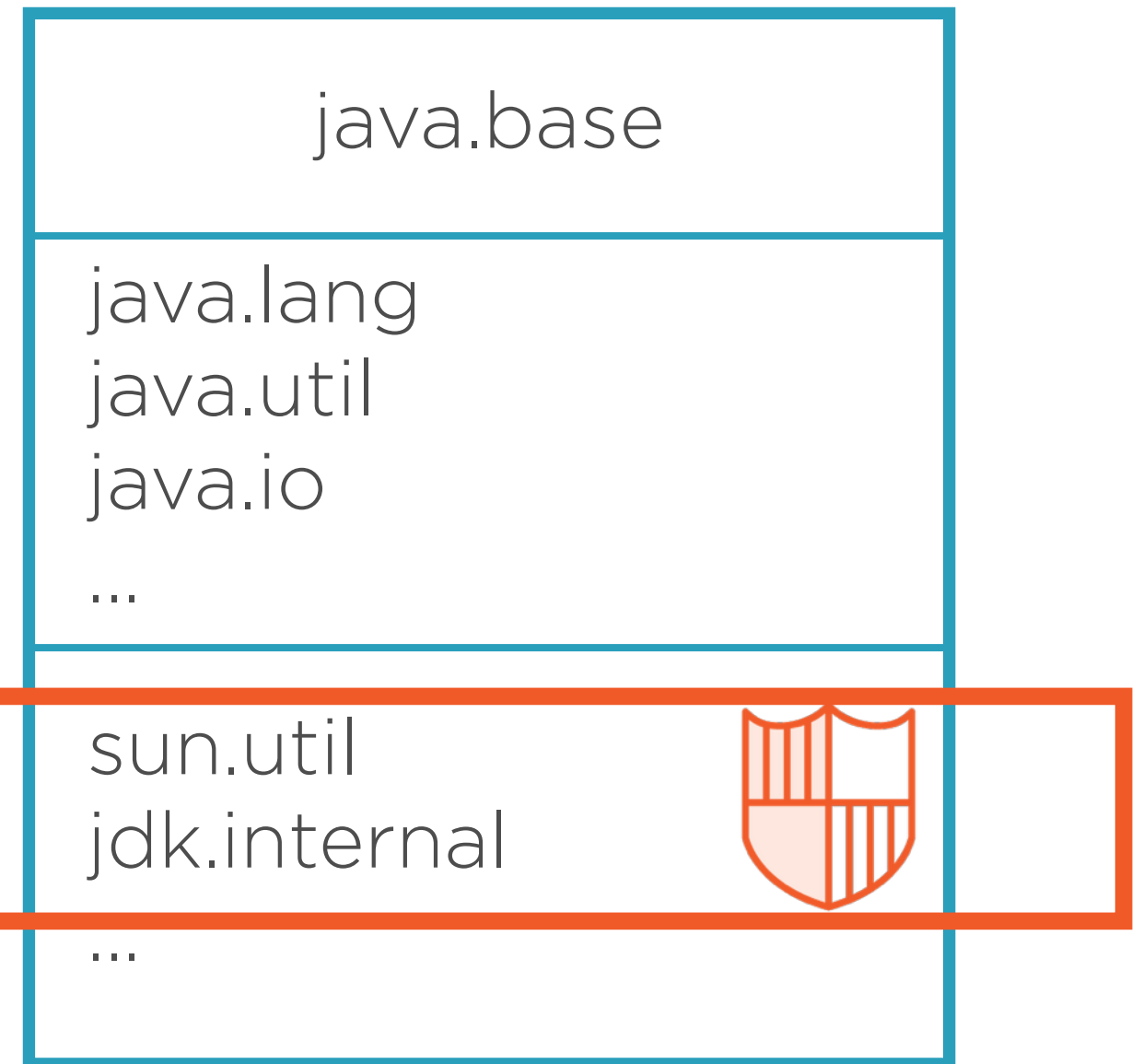
...



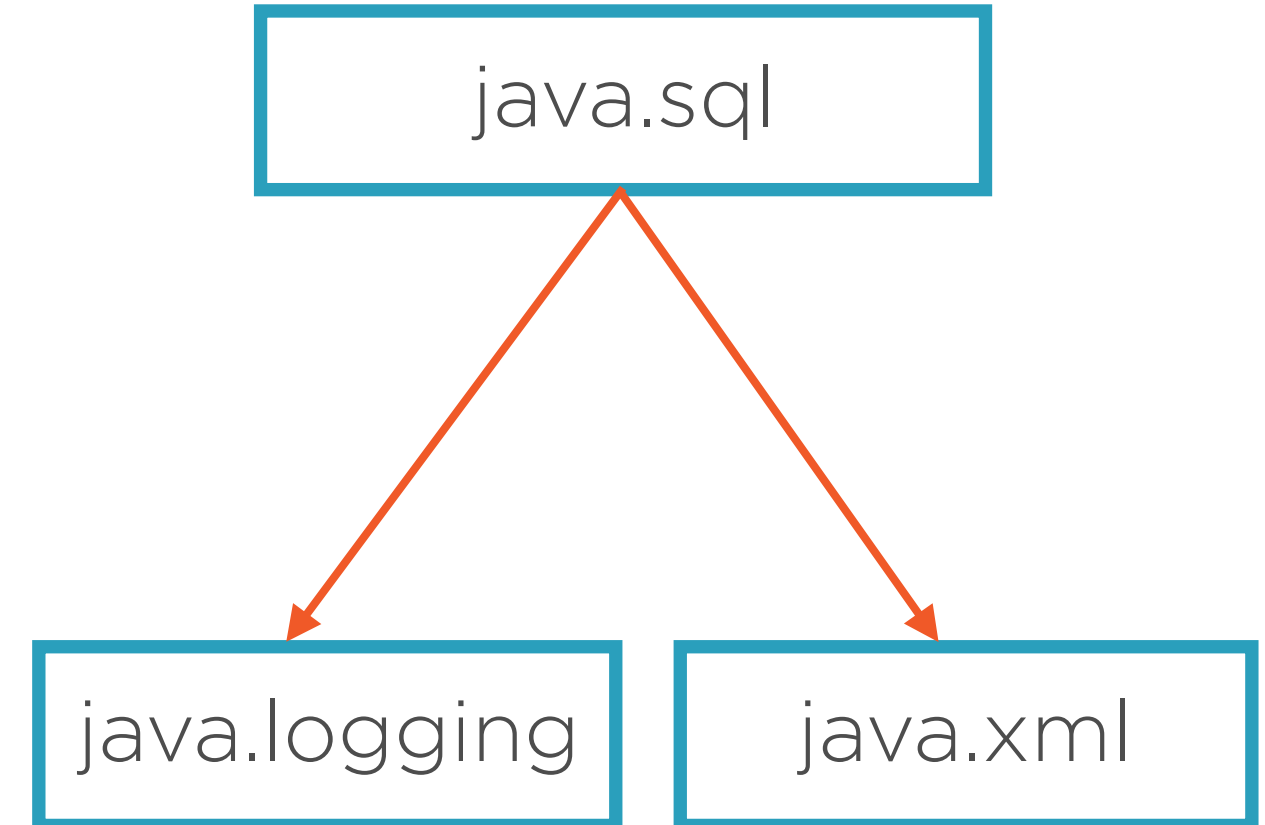
module-info.java

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

Module Descriptors



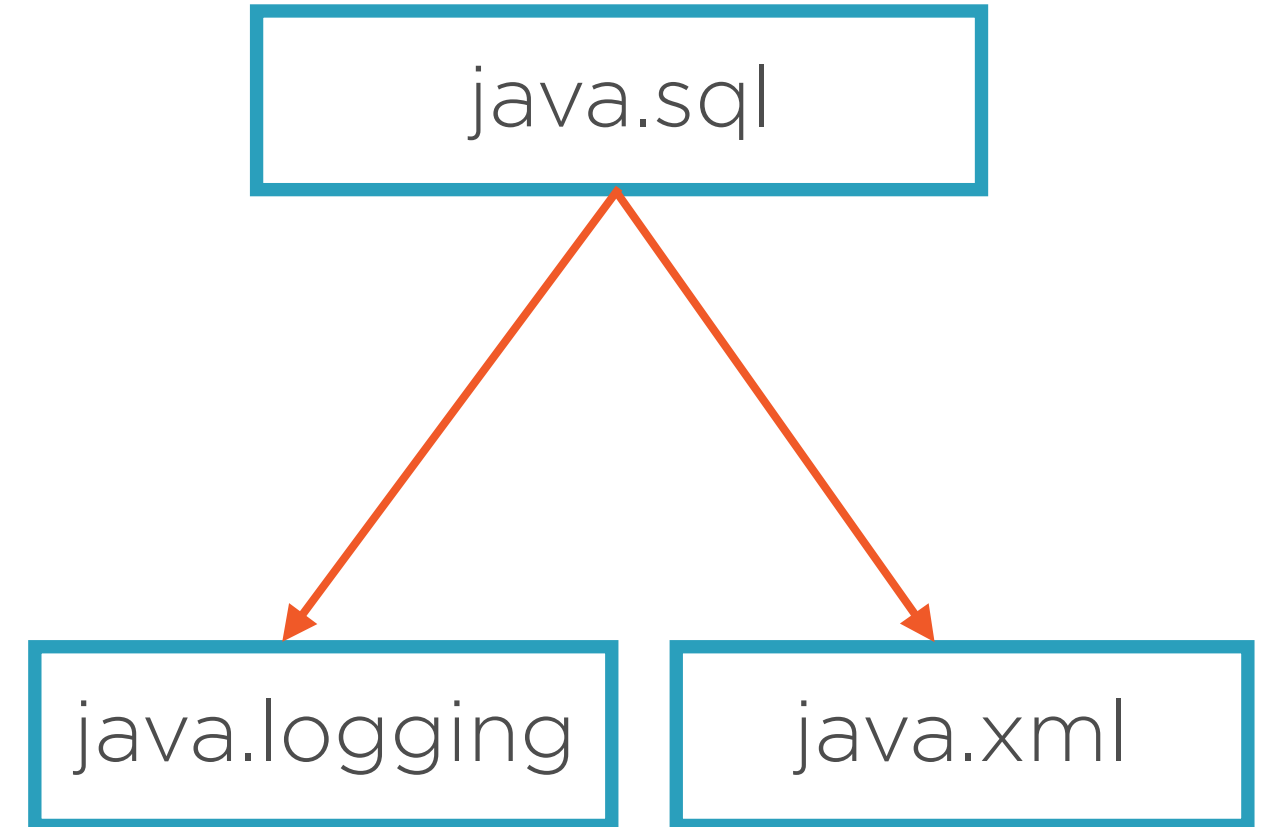
Module Descriptors



module-info.java

```
module java.sql {  
  
    exports java.sql;  
    exports javax.sql;  
    exports  
        javax.transaction.xa;  
  
    requires java.logging;  
    requires java.xml;  
  
}
```

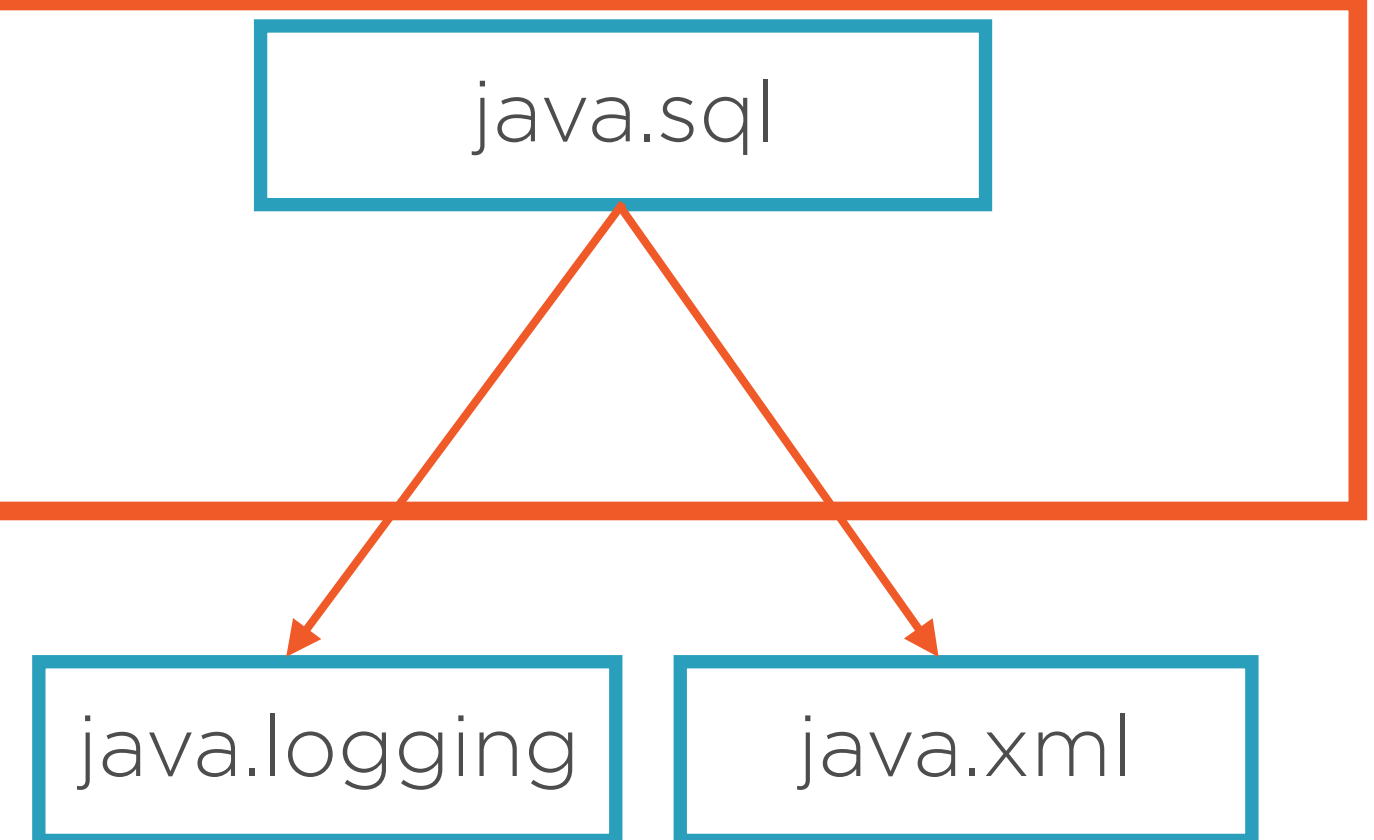
Module Descriptors



module-info.java

```
module java.sql {  
  
    exports java.sql;  
    exports javax.sql;  
    exports  
        javax.transaction.xa;  
  
    requires java.logging;  
    requires java.xml;  
  
}
```

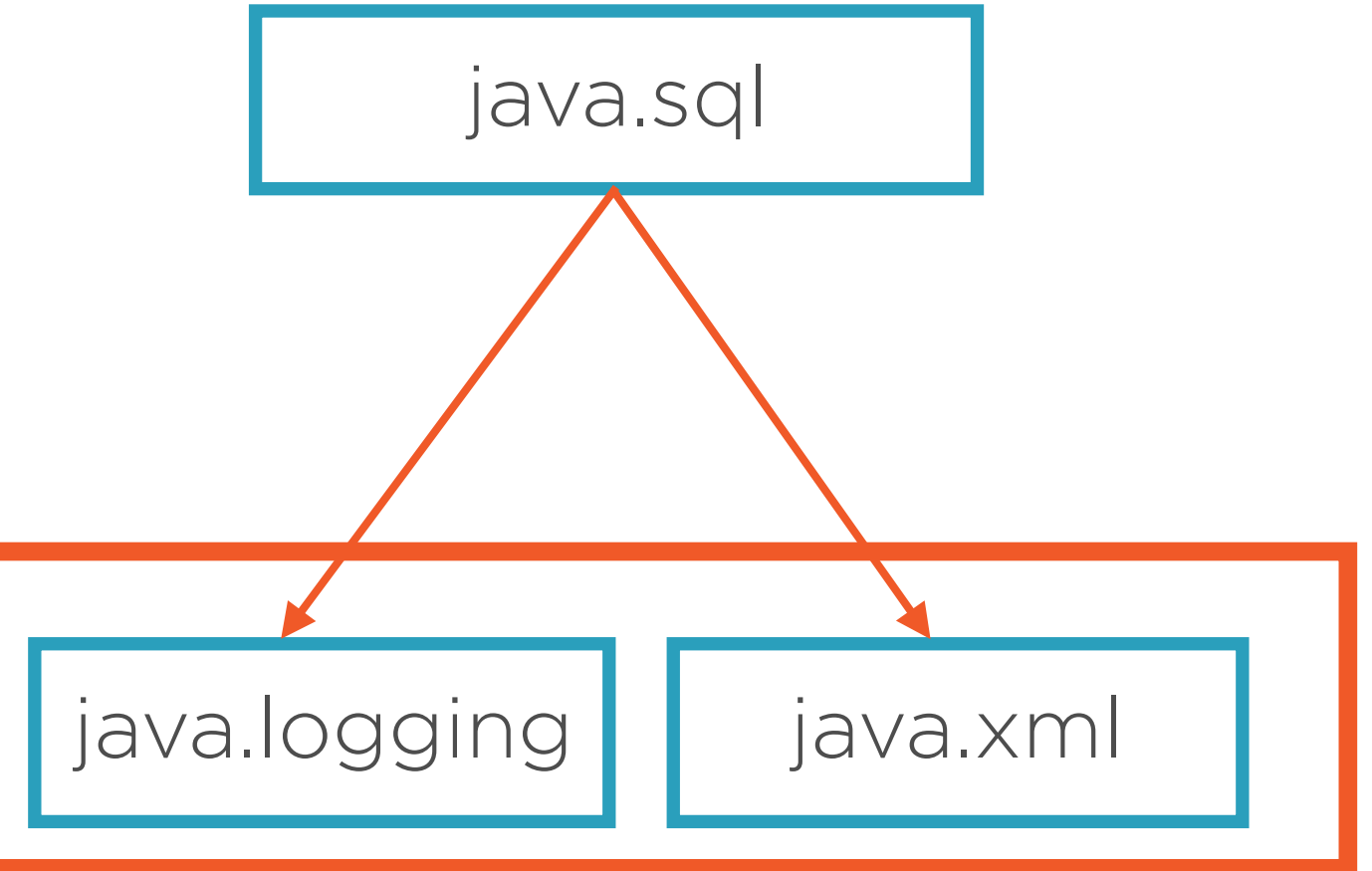
Module Descriptors



module-info.java

```
module java.sql {  
  
    exports java.sql;  
    exports javax.sql;  
    exports  
        javax.transaction.xa;  
  
    requires java.logging;  
    requires java.xml;  
  
}
```

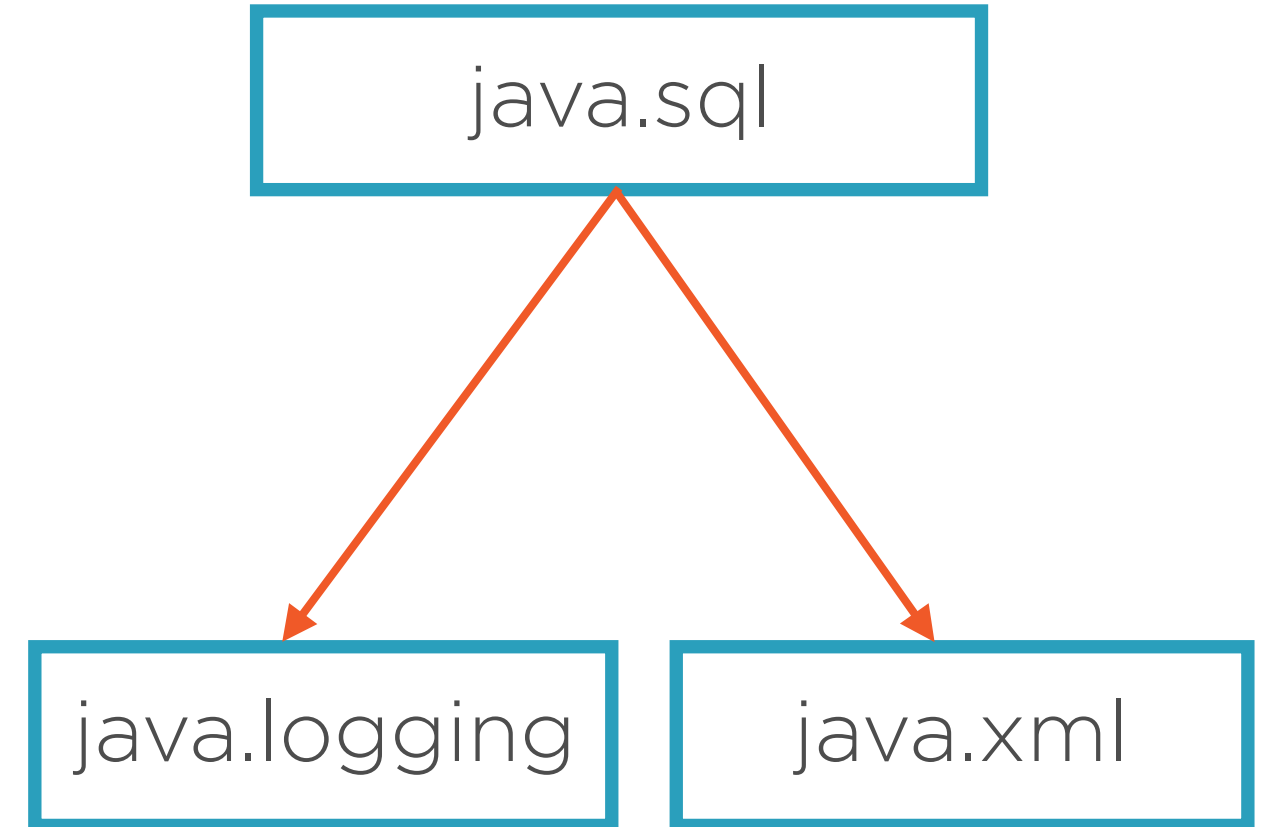
Module Descriptors



module-info.java

```
module java.sql {  
  
    exports java.sql;  
    exports javax.sql;  
    exports  
        javax.transaction.xa;  
  
    requires java.logging;  
    requires java.xml;  
  
}
```

Module Descriptors



Example Application: EasyText



Example Application: EasyText



Analyze text complexity

Example Application: EasyText



Analyze text complexity

Read file => output metric

Example Application: EasyText



Analyze text complexity

Read file => output metric

Multiple algorithms, e.g. Flesch-Kincaid, Coleman

Example Application: EasyText



Start simple:

easytext

Example Application: EasyText



Start simple:

easytext

Later:

- Separate analysis from main app
- Multiple front-ends
- Multiple analyses

Example Application: EasyText



easytext

Summary

Modular JDK

- Explicit dependencies
- Encapsulation

Module Descriptors

Compiling and running a module