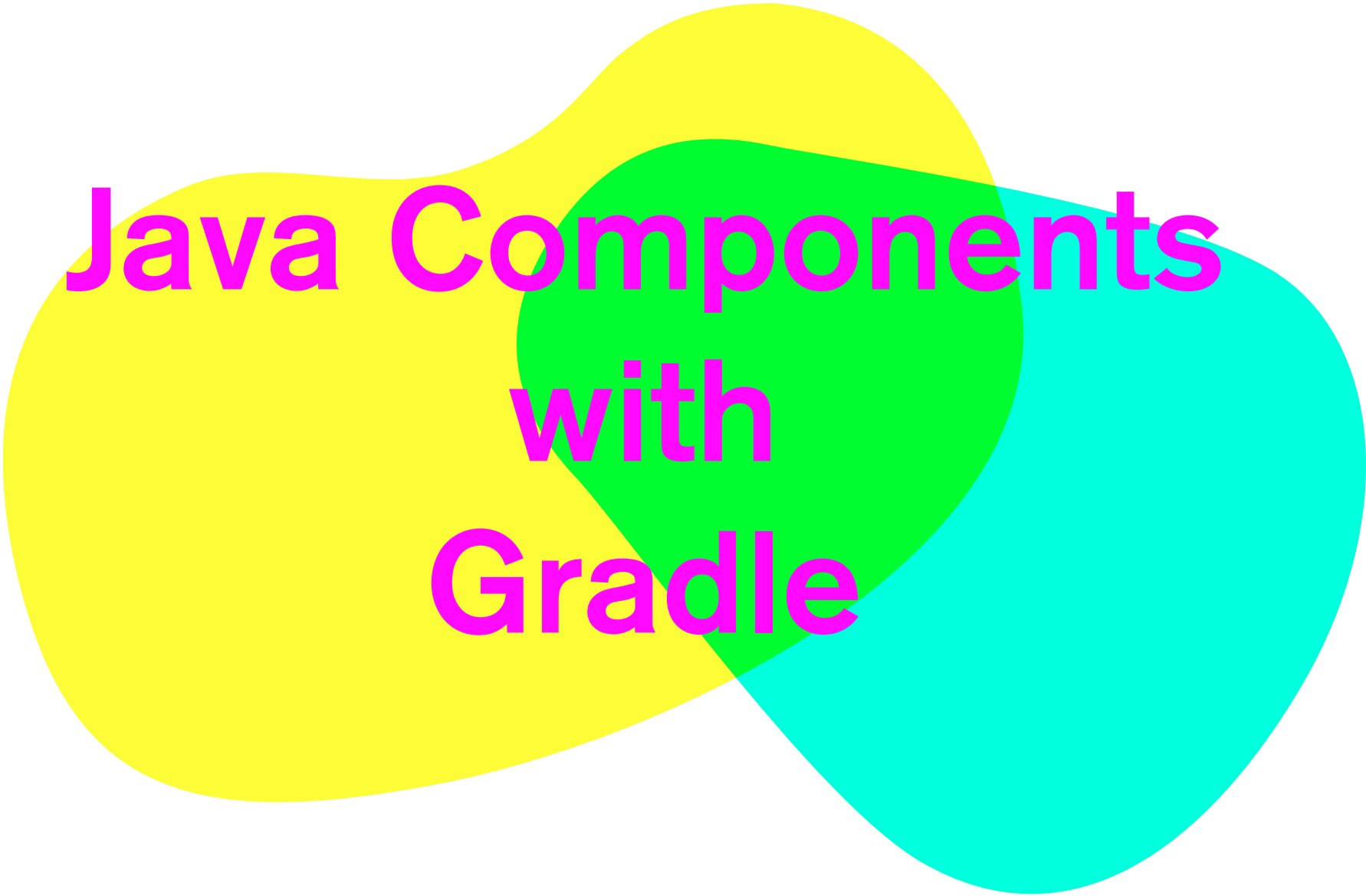




Java Components

Solving the Puzzle with Jigsaw and Gradle

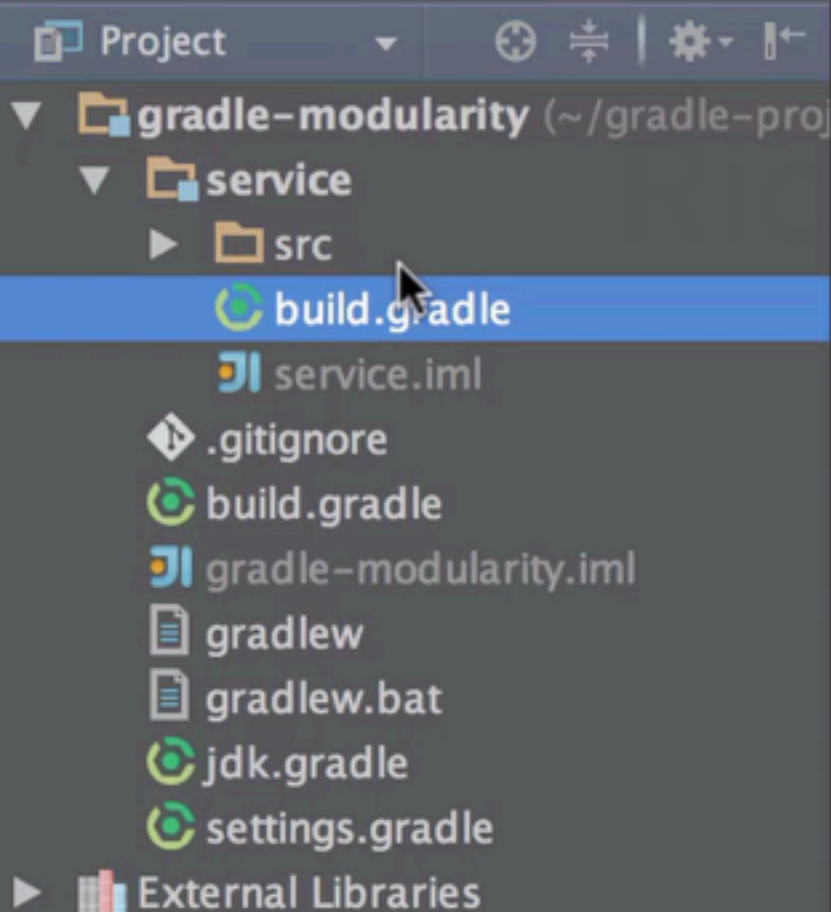
Hans Dockter
Gradle Founder & CEO
Twitter: [hans_d](#)
Email: hans@gradle.com

An abstract graphic consisting of three overlapping, organic shapes in yellow, green, and cyan. The text "Java Components with Gradle" is centered over these shapes in a bold, black, sans-serif font.

Java Components with Gradle

Rich Software Model

Component Based



No files are open

- Search Everywhere with **Double** ⏏
- Open a file by name with **⏏N**
- Open Recent Files with **⌘E**
- Open Navigation Bar with **⌘N**
- Drag and Drop file(s) here from Finder

Project

- gradle-modularity (~ / gradle-proj)
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
 - External Libraries

```
plugins {  
    id 'jvm-component'  
    id 'java-lang'  
}  
  
model {  
    components {  
        service(JvmLibrarySpec) {  
            sources {  
                java {  
                    source.srcDir 'src/service/java'  
                }  
            }  
        }  
        client(JvmLibrarySpec) {  
            sources {  
                java {  
                    source.srcDir 'src/client/java'  
                }  
            }  
        }  
    }  
}
```

Project

- gradle-modularity (~ /gradle-proje)
 - service
 - build
 - src
 - client
 - java
 - org.gradle.examp
 - Client**
 - service
 - java
 - org.gradle.examp
 - Service
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
- External Libraries

service x Client.java x

```
package org.gradle.example.client;

import org.gradle.example.service.Service;

public class Client {
    Service service;
}
```


- gradle-modularity (~ /gradle-proje)
 - service
 - build
 - src
 - client
 - java
 - org.gradle.examp
 - Client
 - service
 - java
 - org.gradle.examp
 - Service

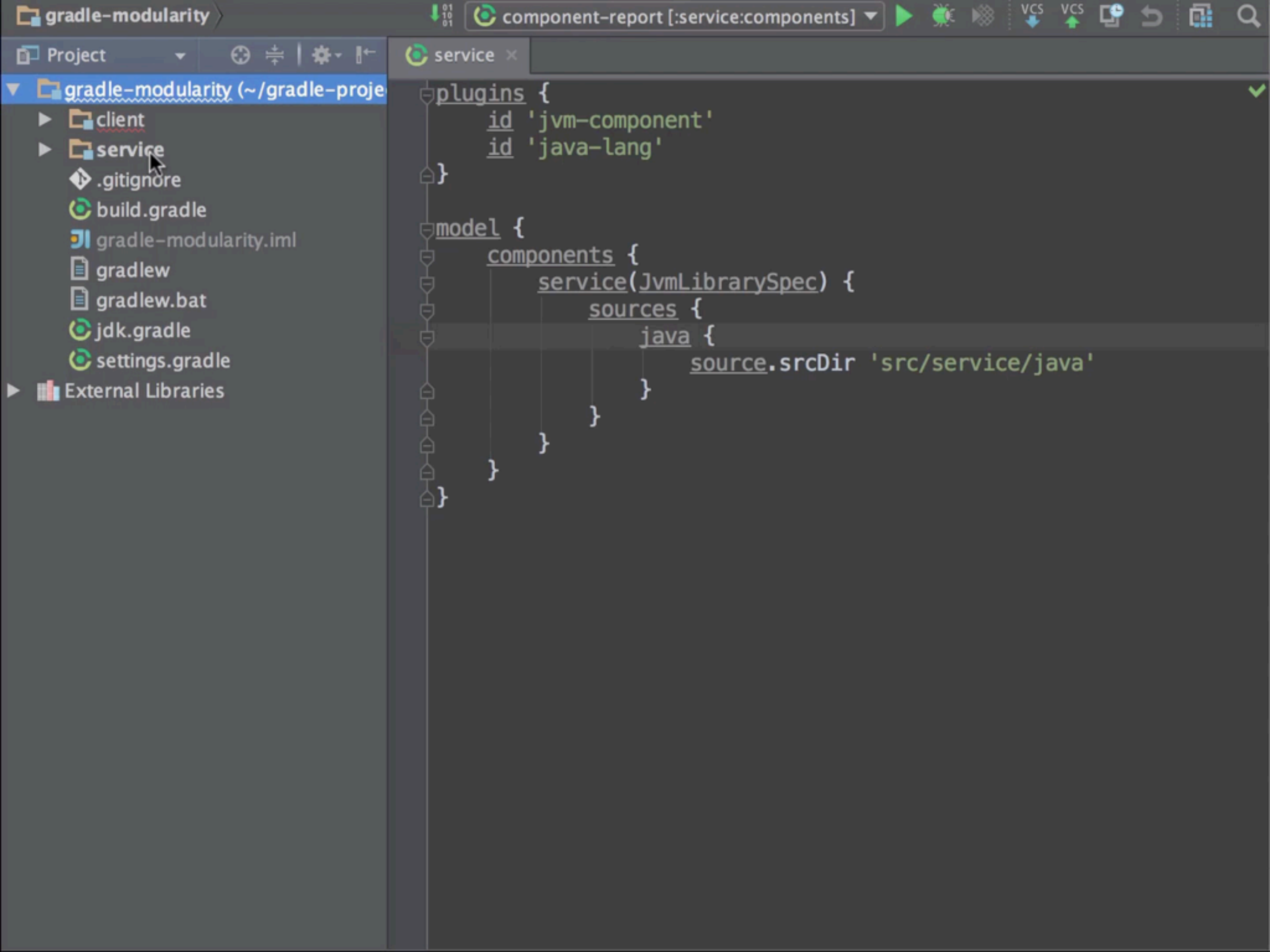
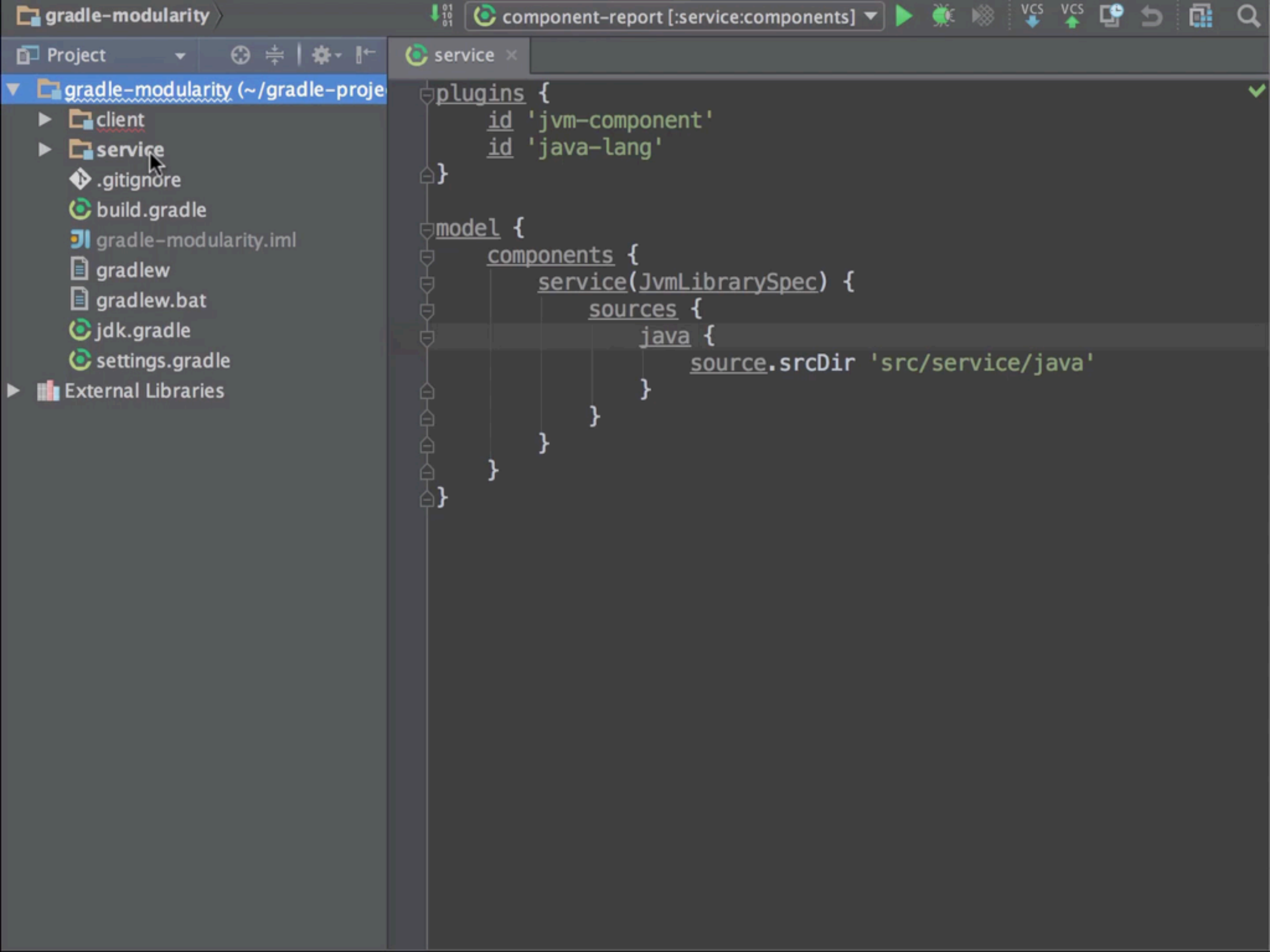
- build.gradle
- service.iml
- .gitignore
- build.gradle
- gradle-modularity.iml
- gradlew
- gradlew.bat
- jdk.gradle
- settings.gradle
- External Libraries

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'
                }
            }
        }
        client(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/client/java'
                }
            }
            dependencies {
                library 'service'
            }
        }
    }
}
    
```


Build Refactoring



Specify JDK

- gradle-modularity (~ / gradle-proje
 - client
 - src
 - build.gradle
 - client.iml
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
- External Libraries

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        client(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/client/java'
                }
                dependencies {
                    project ':service' library 'service'
                }
            }
            targetPlatform 'java8'
        }
    }
}
    
```

Multiple JDK's

Abstract colorful shapes in the background, including a large yellow circle, a green circle, a red circle, and a blue circle, all overlapping and partially obscured by a large, irregular, light blue shape on the right side.

Project

- gradle-modularity (~ / gradle-proje)
 - client
 - src
 - build.gradle
 - client.iml
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
- External Libraries

```
plugins {  
    id 'jvm-component'  
    id 'java-lang'  
}  
  
model {  
    components {  
        service(JvmLibrarySpec) {  
            sources {  
                java {  
                    source.srcDir 'src/service/java'  
                }  
            }  
            targetPlatform 'java7'  
            targetPlatform 'java8'  
        }  
    }  
}
```

Cannot resolve symbol 'targetPlatform'

JDK Version Conflicts

▼ gradle-modularity (~ /gradle-proje
 ▼ client
 ▶ build
 ▶ src

build.gradle

client.iml

▼ service

▶ build

▶ src

build.gradle

service.iml

.gitignore

build.gradle

gradle-modularity.iml

gradlew

gradlew.bat

jdk.gradle

settings.gradle

External Libraries

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        client(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/client/java'
                    dependencies {
                        project ':service' library 'service'
                    }
                }
            }
            targetPlatform 'java6'
            targetPlatform 'java7'
            targetPlatform 'java8'
        }
    }
}
    
```

Project

- gradle-modularity (~ /gradle-proje)
 - client
 - build
 - src
 - build.gradle
 - client.iml
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
- External Libraries

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'
                }
            }
            targetPlatform 'java6'
            targetPlatform 'java7'
            targetPlatform 'java8'
        }
    }
}
    
```

Cannot resolve symbol 'targetPlatform'

Project

gradle-modularity (~ / gradle-proje

client

build

src

build.gradle

client.iml

service

build

src

build.gradle

service.iml

.gitignore

build.gradle

gradle-modularity.iml

gradlew

gradlew.bat

jdk.gradle

settings.gradle

External Libraries

service

client

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        client(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/client/java'

                    dependencies {
                        project ':service' library 'service'
                    }
                }
            }

            targetPlatform 'java6'
            targetPlatform 'java7'
            targetPlatform 'java8'
            targetPlatform 'java9'
        }
    }
}

```

To API or not API

Project

- gradle-modularity (~ /gradle-proje)
 - client
 - build
 - src
 - build.gradle
 - client.iml
 - service
 - build
 - src
 - service
 - java
 - org.gradle.examp
- build.gradle
- service.iml
- .gitignore
- build.gradle
- gradle-modularity.iml
- gradlew
- gradlew.bat
- jdk.gradle
- settings.gradle
- External Libraries

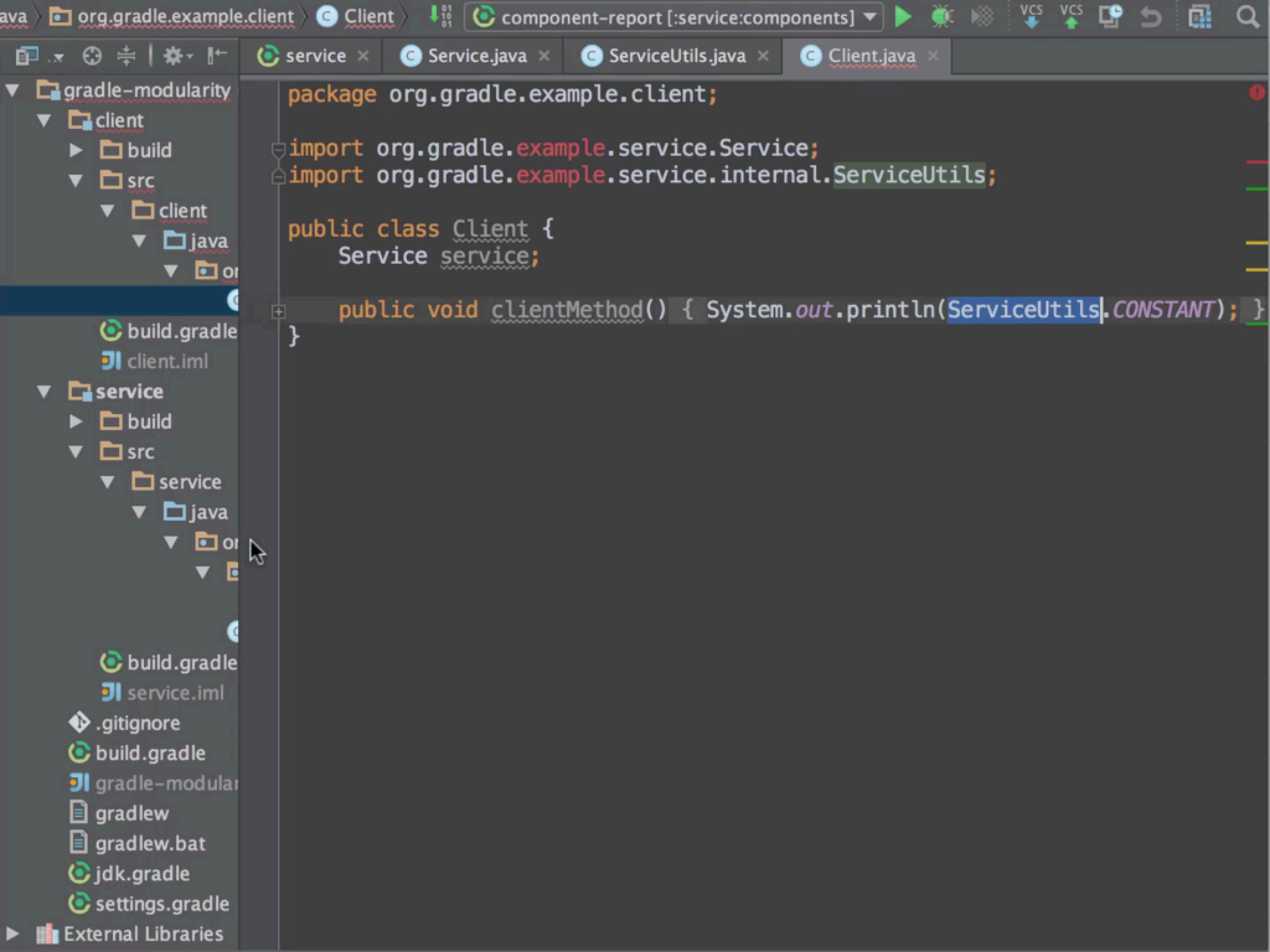
service

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'
                }
            }
            targetPlatform 'java6'
            targetPlatform 'java7'
            targetPlatform 'java8'
        }
    }
}
    
```


Gradle to the rescue



01 component-report [:service:components] 01

service × Service.java × ServiceUtils.java × Client.java ×

```
package org.gradle.example.client;

import org.gradle.example.service.Service;

public class Client {
    Service service;

    public void clientMethod() {
        System.out.println("E00");
    }
}
```

External Dependencies?

Project

- gradle-modularity (~ / gradle-projects)
 - client
 - build
 - src
 - build.gradle
 - client.iml
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
 - External Libraries

service

```

plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'
                }
                dependencies {
                    library 'org.eclipse.jetty:jetty-servlet'
                    library 'org.apache.httpcomponents:httpcore'
                }
            }
            api {
                exports 'org.gradle.example.service'
            }
            targetPlatform 'java6'
            targetPlatform 'java7'
            targetPlatform 'java8'
        }
    }
}
    
```

To API or not API

client > java > org.gradle.example.client

component-report [:service:components]

Project

- gradle-modularity (~ /gradle-projects)
 - client
 - build
 - src
 - client
 - java
 - org.gradle.example.
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle

External Libraries

```
plugins {  
    id 'jvm-component'  
    id 'java-lang'  
}  
  
model {  
    components {  
        service(JvmLibrarySpec) {  
            sources {  
                java {  
                    source.srcDir 'src/service/java'  
                }  
            }  
            dependencies {  
                library 'org.eclipse.jetty:jetty-servlet'  
                library 'org.apache.httpcomponents:httpcore'   
            }  
        }  
        api {  
            exports 'org.gradle.example.service'  
        }  
        targetPlatform 'java6'  
        targetPlatform 'java7'  
        targetPlatform 'java8'  
    }  
}
```


Project

- gradle-modularity (~ /gradle-projects)
 - client
 - build
 - src
 - client
 - java
 - org.gradle.example.client
 - Client
- build.gradle
- client.iml

- service
- build
- src
- build.gradle
- service.iml
- .gitignore
- build.gradle
- gradle-modularity.iml
- gradlew
- gradlew.bat
- jdk.gradle
- settings.gradle
- External Libraries

```

package org.gradle.example.client;

import org.gradle.example.service.Service;

public class Client {
    Service service;

    public void clientMethod() {
        System.out.println("FOO");
    }
}
    
```

API Dependencies

Abstract geometric shapes in various colors (cyan, green, yellow, dark green, and teal) overlapping each other in the bottom right corner of the slide.

Project

- gradle-modularity (~ /gradle-projects)
 - client
 - build
 - src
 - client
 - java
 - org.gradle.example.
 - Client
 - ClientServlet
 - service
 - build
 - src
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - settings.gradle
- External Libraries

```
package org.gradle.example.client;

import org.gradle.example.service.Service;

public class Client {
    Service service;

    public void clientMethod() {
        System.out.println("F00");
    }
}
```

▼ gradle-modularity (~ /gradle-projects)

- ▼ client
 - build
 - src
 - client
 - java
 - org.gradle.example.
 - Client
 - ClientServlet
- build.gradle
- client.iml

- ▼ service
- build
- src
 - service
 - java
 - org.gradle.example.
 - internal
 - Service
 - ServiceServlet
- build.gradle
- service.iml
- .gitignore
- build.gradle
- gradle-modularity.iml
- gradlew
- gradlew.bat
- jdk.gradle
- settings.gradle

```

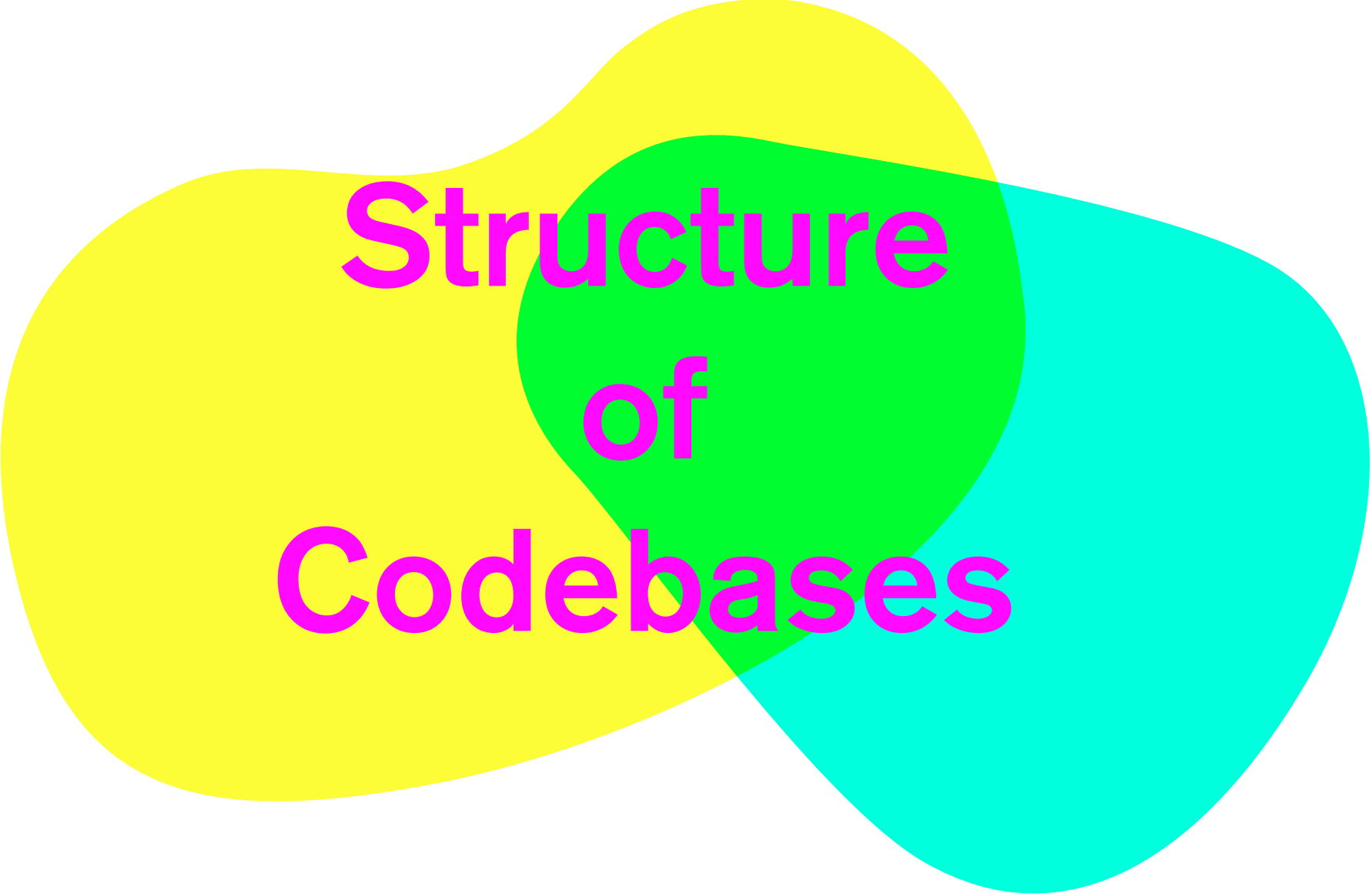
plugins {
    id 'jvm-component'
    id 'java-lang'
}

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'
                }
                dependencies {
                    library 'org.eclipse.jetty:jetty-servlet'
                    library 'org.apache.httpcomponents:httpcore'
                }
            }
        }
        api {
            exports 'org.gradle.example.service'
        }
        targetPlatform 'java6'
        targetPlatform 'java7'
        targetPlatform 'java8'
    }
}
    
```



Achievement?



A Venn diagram consisting of three overlapping circles. The top circle is yellow, the bottom-left circle is green, and the bottom-right circle is cyan. The text "Structure of Codebases" is written in a bold, black, sans-serif font across the center of the diagram, overlapping all three circles.

Structure of Codebases

Epic Problem

**Public API's
are
expensive**

Package Design

Tooling -> Reality

Tooling -> Enforce API

Tooling -> Enforce Thinking

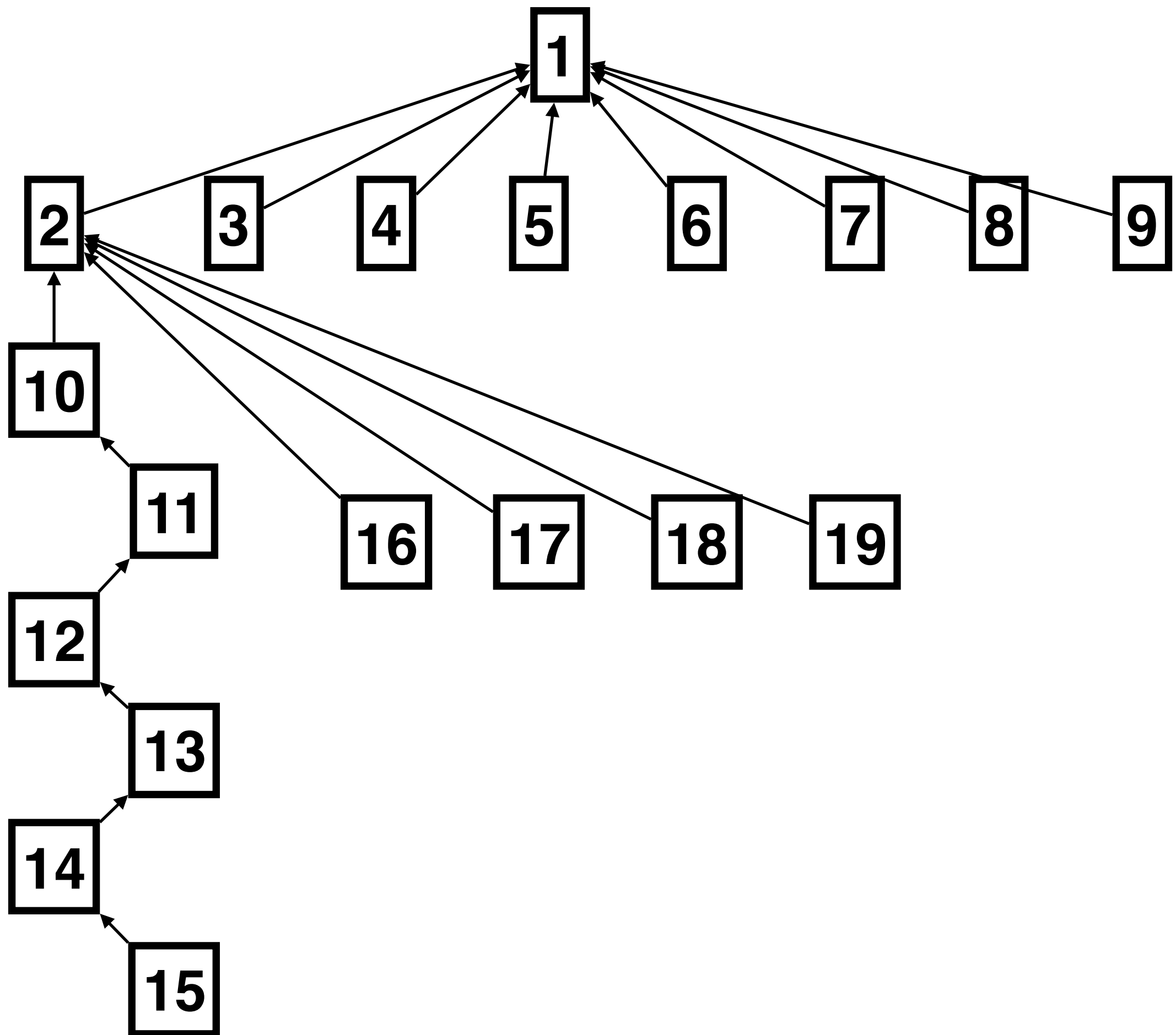
Tooling -> API Metrics



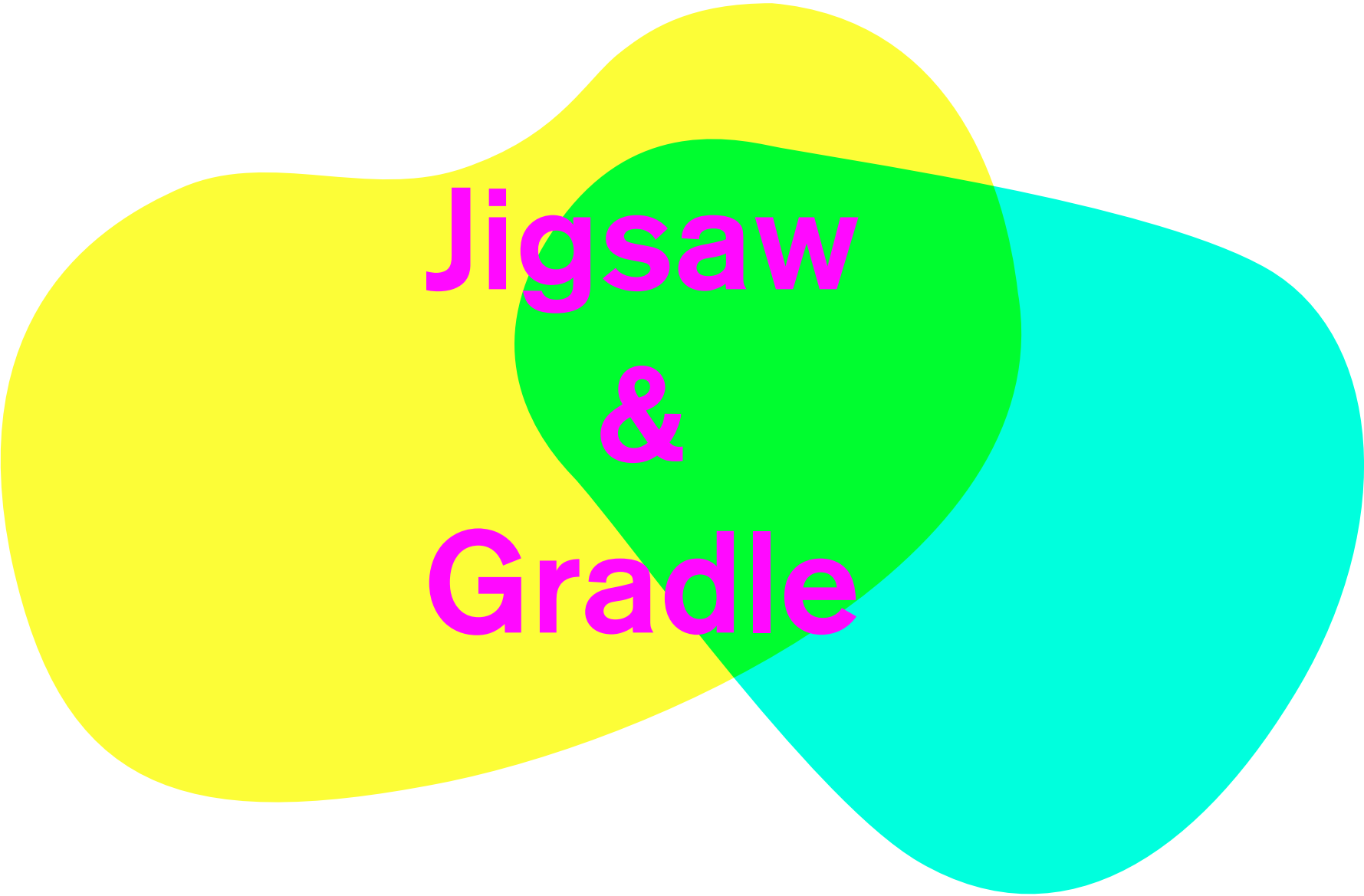
11.11.05

Security

Build Performance



```
hans:mediumNewJava$ █
```

A Venn diagram consisting of two overlapping circles. The left circle is yellow, and the right circle is cyan. The intersection of the two circles is shaded green. The text "Jigsaw & Gradle" is written in a bold, black, sans-serif font across the intersection.

**Jigsaw
&
Gradle**

How to get modular now



Lock it down with Gradle

- 1. Enforce the current API**
- 2. Improve API/package structure**
- 3. Enforce the improved API**
- 4. Go back to #2**

Educate!

Moving to Jigsaw

▼ **gradle-modularity** (~ /gradle-pro)

- ▼ client
 - ▼ src
 - ▼ client
 - ▼ java
 - ▼ org.gradle.examp
 - Client.java
 - ClientServlet.ja
- build.gradle
- client.iml
- exportToHTML
- ▼ service
 - ▼ src
 - ▼ service
 - ▼ java
 - ▼ org.gradle.examp
 - internal
 - Service
 - ServiceServ
 - build.gradle
 - service.iml
 - .gitignore
 - build.gradle
 - gradle-modularity.iml
 - gradlew
 - gradlew.bat
 - jdk.gradle
 - jigsaw.gradle
 - settings.gradle

```

    id 'jvm-component'
    id 'java-lang'
}
apply from: '../jigsaw.gradle'

model {
    components {
        service(JvmLibrarySpec) {
            sources {
                java {
                    source.srcDir 'src/service/java'

                    dependencies {
                        library 'org.eclipse.jetty:jetty-servlet:9'
                        library 'org.apache.httpcomponents:httpclient:4.5.13'
                    }
                }
            }
            api {
                exports 'org.gradle.example.service'
            }
            targetPlatform 'java7'
            targetPlatform 'java8'
            targetPlatform 'java9'
        }
    }
}

```

Start early ... so easy

Abstract colorful shapes in the background, including a large teal shape on the right, a yellow shape in the center, and various green and cyan shapes at the bottom.

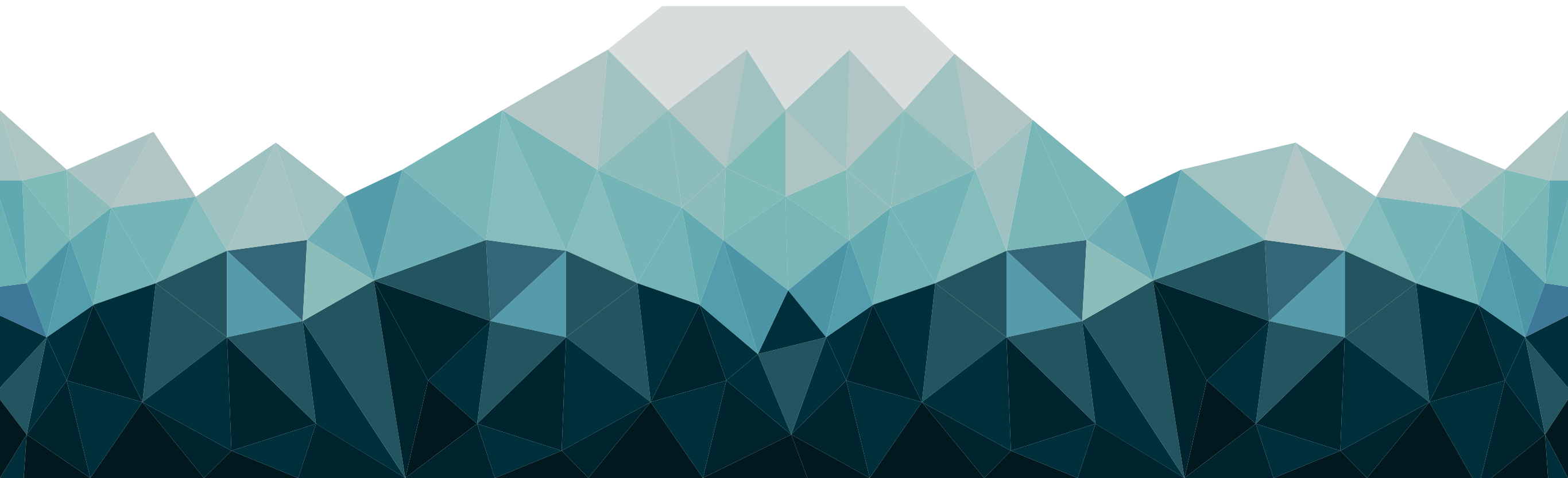
In a Java 9+ world



Jigsaw with Gradle

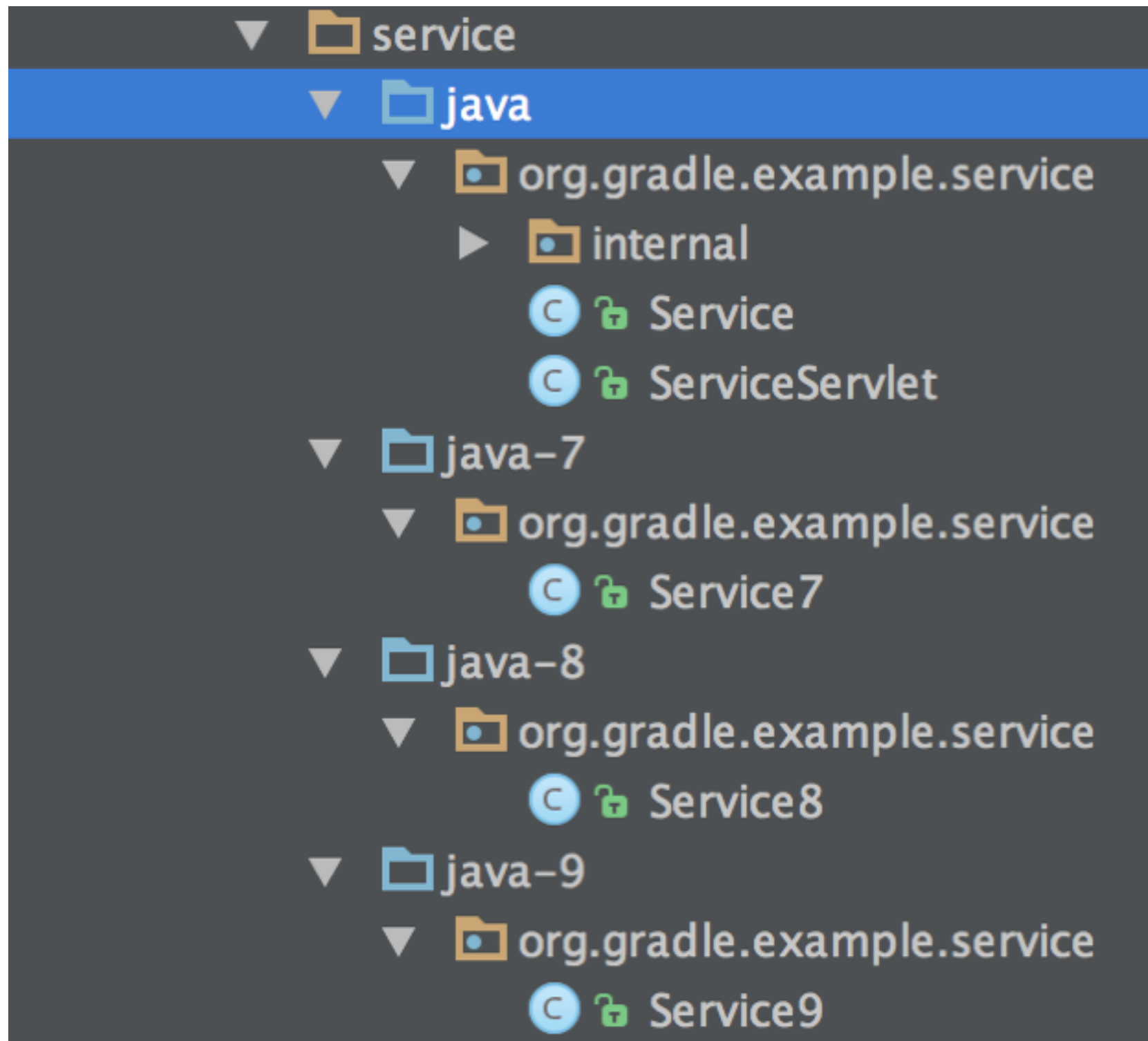
- **Module-info support not in core (yet)**
- **New Java Model is incubating**

Roadmap





Sources per JDK





Dependencies per JDK

```
sources {  
    java {  
        dependencies {  
            library 'org.eclipse.jetty:jetty-servlet:9.3.5'  
        }  
    }  
    java8 {  
        dependencies {  
            library 'org.apache.httpcomponents:httpclient:4.4.1'  
        }  
    }  
    java9 {  
        dependencies {  
            library 'org.apache.httpcomponents:httpclient:4.5.1'  
        }  
    }  
}
```




Multi-Release Jar

<http://openjdk.java.net/jeps/238>

```
jar root
  - A.class
  - B.class
  - C.class
  - D.class
  - META-INF
    - versions
      - 8
        - A.class
        - B.class
      - 9
        - A.class
```



Find out how we're building happiness.

Gradle.com is coming soon. Sign up for email updates.

Your email

Size of your team

Primary language

Your role

SUBMIT



■ ■ ■ Intermediate

Built by  gradle 

-  **Approx. 6 weeks**
Assumes 6hr/wk (work at your own pace)
-  **Join 2,735 Students**

Course Summary

This course explores how the Gradle build tool compiles and packages apps, and you'll learn to customize the build process. The first half of this course is for anyone interested in Gradle, build automation, and continuous delivery of software.

The latter half of the course reveals the magic that happens after you hit the "Run" button in Android Studio. You'll also explore advanced Android topics, learning to configure free vs paid app flavors, create and integrate Android libraries, test your app, and prepare your app for the Play Store.

Start Free Course

[Start free course](#)

 **Free**

You get

-  Instructor videos
-  Learn by doing exercises



gradle.org/trainings

Introductory & Intermediate

Introduction to Gradle

Gradle for Android

Gradle C/C++ Workshop

Advanced Gradle Fundamentals


Advanced

Extending Gradle

Mastering Dependencies and Multi-project Builds

Standardizing Enterprise Builds

Continuous Delivery with Gradle

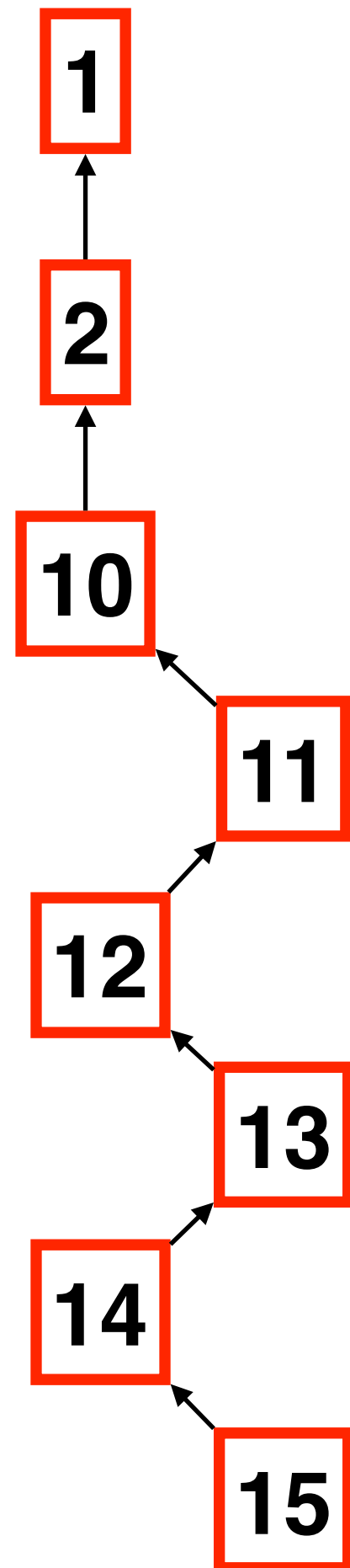
A decorative background consisting of two large, overlapping, organic shapes. The left shape is yellow and the right shape is cyan. They overlap in the center, creating a darker green area.

Come to our Booth #



Q & A

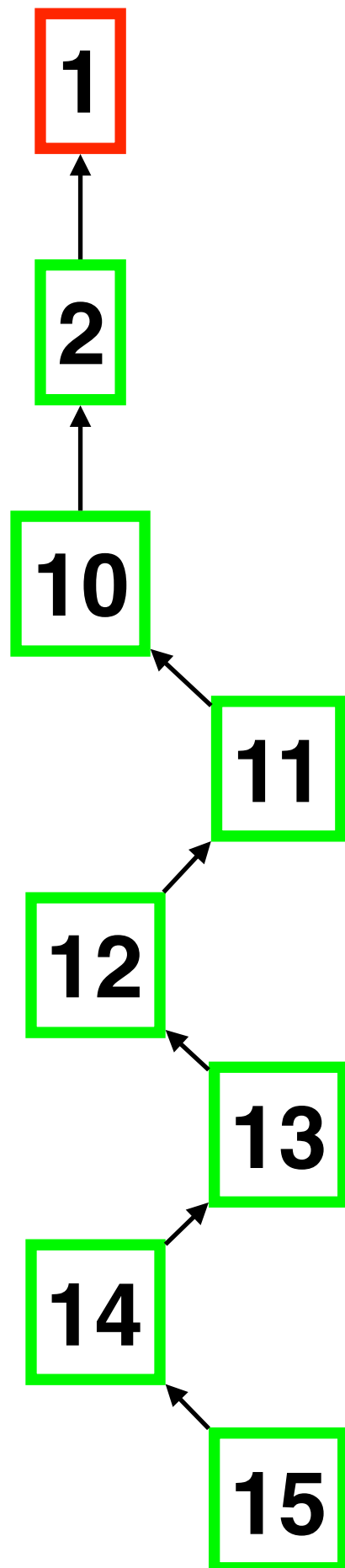
Compile Avoidance



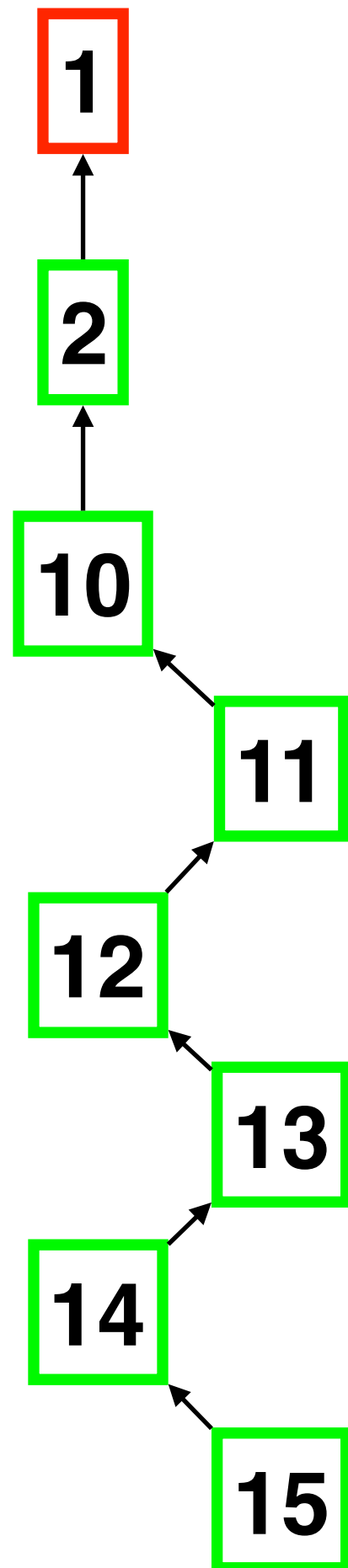
**Any change in the Java source code
of 1 that changes the byte code.**

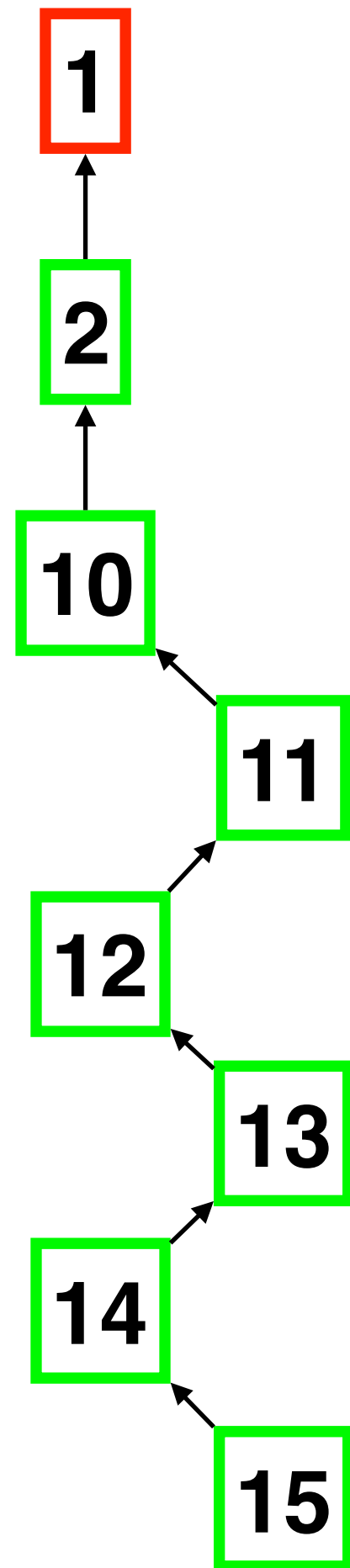
Any change of an external dependency of 1

Any change in an implementation class



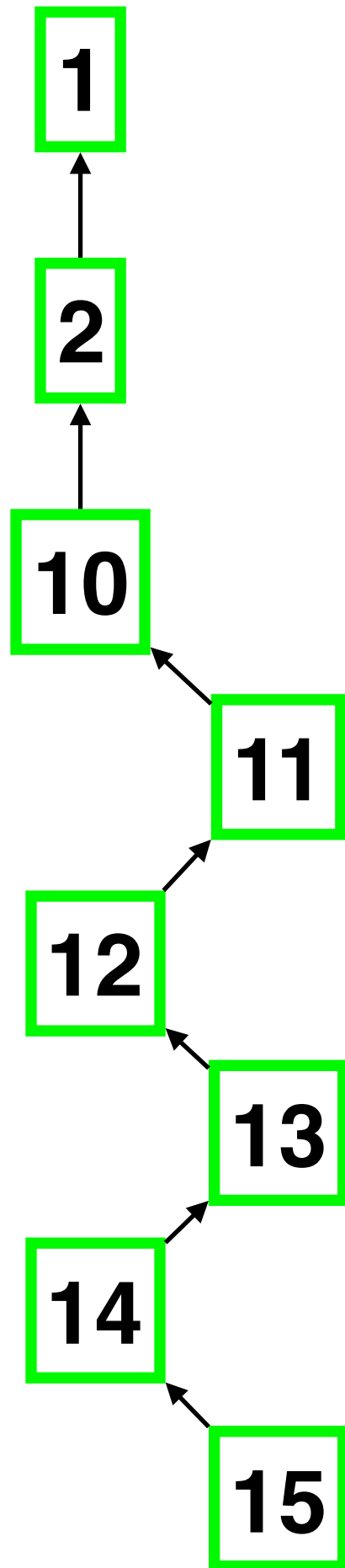
**Any change of an API class
that does not change the ABI**

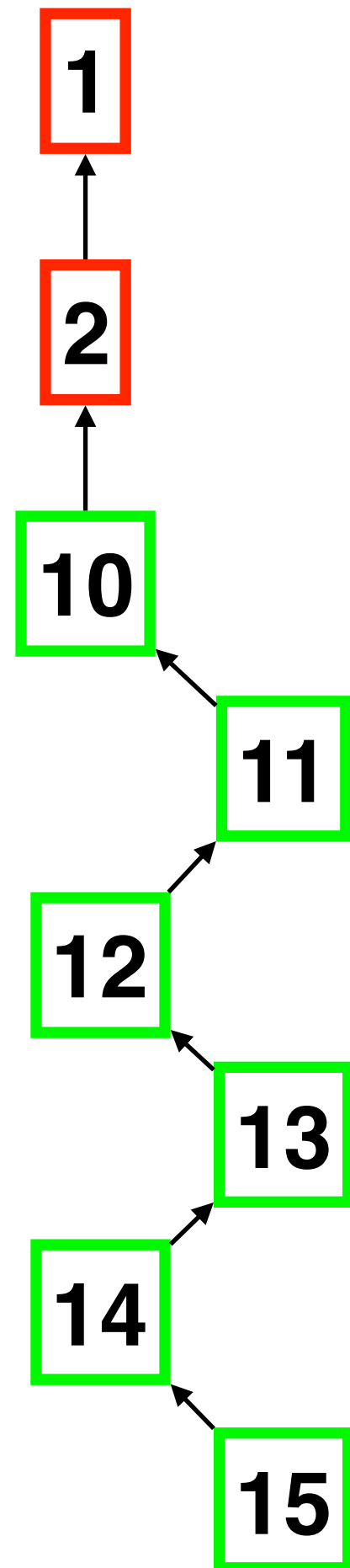




**Change of a version of a
non-exported dependency
that does modify its ABI.**

**Change of a version of a
non-exported dependency
that does not modify its ABI.
(not implemented yet)**

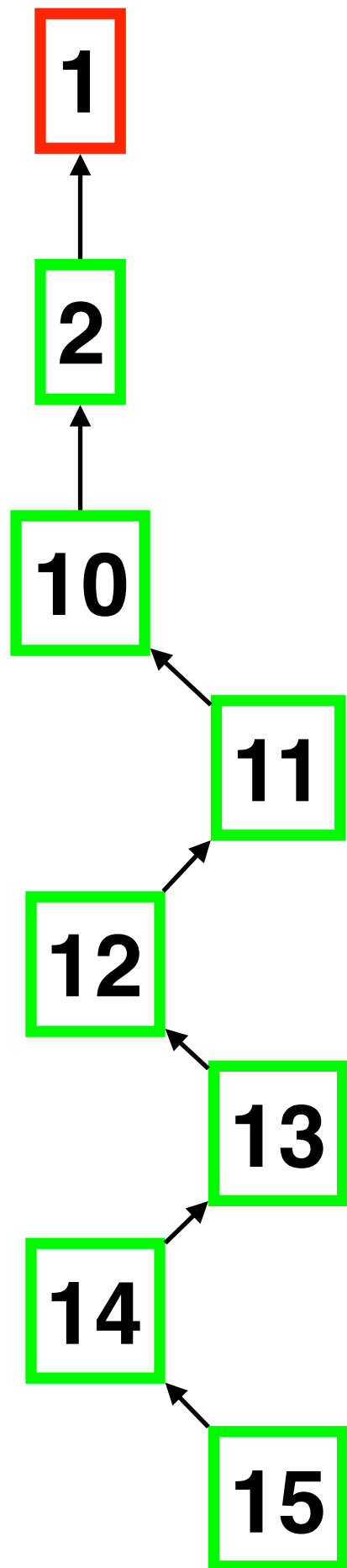


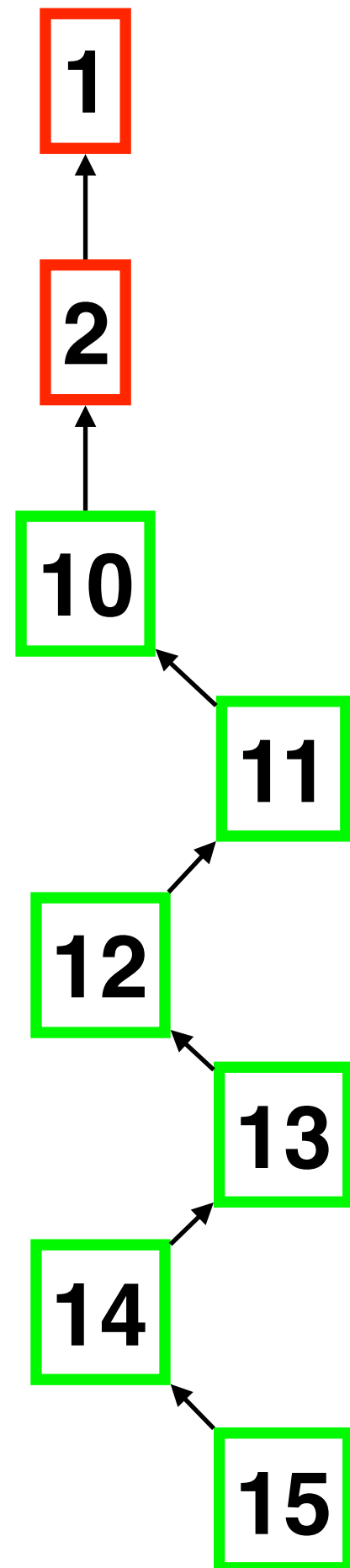


**Change of an API class
that do change the ABI
if 1 is not exported by 2**

**Change of a version of an
exported dependency
that does not modify its ABI.
(not implemented yet)**

Currently 2 get's recompiled





**Change of a version of an
exported dependency
that does modify its ABI.
(if 1 is not exported by 2)**

A Venn diagram consisting of three overlapping circles. The left circle is yellow, the right circle is blue, and the top circle is red. The word "Jigsaw" is written in a bold, black, sans-serif font across the intersection of the yellow and blue circles.

Jigsaw



Jigsaw makes the JDK even better

- **Better Security**
- **Custom Runtime**
- **Improved performance**



Performance API signature

- Any body of an API class (method, constructor)
- Adding a private method
- Order of methods change