

# Modularity vs Java

Grzegorz Piwowarek

@pivovarit

What's a Module?  
Why do you think it's needed?

We're good at high-level encapsulation (Service-level)

We're good at low-level encapsulation (Class-level)

And what about everything inbetween?



rt.jar

All runtime Java libraries



What happens if a JAR is missing on classpath?

What if it's duplicated?

What if there are multiple versions of the same jar?

What are the visibility restrictions between jars?

# JAR HELL

- Lack of explicit dependencies
  - Transitive Dependencies
- Random Class Resolution (shadowing)
- Version Conflicts (NoClassDefFoundError)

# Modularity Visualized

(poorly)

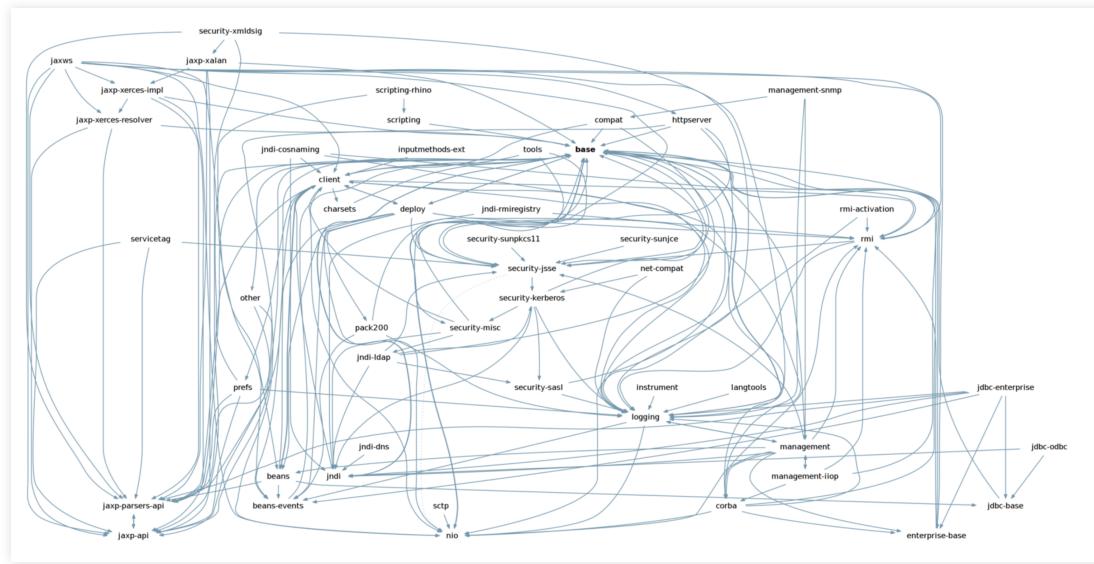


# Benefits of Modularization

- Strong Encapsulation (hidden implementation details)
  - Clear Interfaces/APIs
  - Explicit Dependencies
  - Divide and Conquer

Migrating beyond JDK 8 let's us benefit from modularity  
even if we don't use them explicitly

Why?



Looks good, right?  
Let's see how it feels like...

Thank You!

@pivovarit