Java 9 et 10 en 60’ Chrono

Rappel sur les apports des version précédentes

Try with resources

```
try (BufferedReader br = new BufferedReader(new FileReader(path))) {
   return br.readLine();
}
```

Plutôt que

```
BufferedReader br = new BufferedReader(new FileReader(path));
try {
   return br.readLine();
} finally {
   if (br != null)
   br.close();
}
```

Lambda

```
list.forEach(System.out::println);
list.forEach(s -> System.out.println(s));
```

Plutôt que

```
for (int i = 0; i < list.size(); i++) {
  System.out.println(list.get(i));
}
```

Default methods

```
public interface Foo {
   public default void foo() {
System.out.println("Default
implementation of foo()");
   }
}
```

Java 9

Evolution du language

Méthode privée dans les interfaces

Objectif: permettre de mutualiser du code entre les méthodes statiques et par défaut

Java support in interfaces

	Java 7	Java 8	Java 9
constants			
abstract methods			
default methods			
public static methods			
private methods			
private static methods			

Try-with-ressources

* Possible avec une variable finale ou effectivement finale
* Plus besoin de définir une nouvelle variable

```
public void handle(InputStream is) throws
IOException {
   try (is) {
      // Traitement de l'input stream
   }
}
```

Compact Strings

* En Java 8 ou avant, les Strings sont encodés en UTF-16 sous forme de char[]
* En Java 9, il sont encodés en UTF-8 ou UTF-16 sont forme de byte[] et d’un bit d’encodage
* Pas de modification d’interface

Java 9

Evolution d’API

Collection factories

* Pour créer des collections non modifiables

```
Set<String> set =
   Set.of("code", "d'", "armor");
List<String> list =
   List.of("code", "d'", "armor");
Map<Integer, String> map =   Map.of(1,
"code",
        2, "d'",
        3, "armor");
```

Evolution de l'API Optional

* ifPresentOrElse() – Invoque le 1er argument si la valeur est non-nulle, sinon invoque le 2eme argument.
* or() – Si la valeur est présente, retourne un Optional décrivant la valeunr, Sinon, retourne l'Optional fourni par le supplier en argument
* stream() – Retourne un stream avec un élément si la valeur est non nulle ou un stream vide.

Evolution de l'API Stream

* takeWhile() et dropWhile() pour traiter un stream jusqu'à une certaine condition
* ofNullable() stream de 0 ou 1 élément selon que l'élément est null ou pas
* iterate() équivalent en fonctionnel à une boucle for

Process API

* Meilleur contrôle sur l'invocation d'un process externe
* Via l'API java.lang.ProcessHandle
* Pour récuperer le process courant: ProcessHandle.current()
* Pour récupérer des infos: processHandle.getInfo()
* Pour terminer: processHandle.destroy()

Evolution de @Deprecated

* Ajout de paramètres optionnels:
  * forRemoval
  * Since
* Le nouvel outil jdeprscan permet de scanner un jar pour vérifier l'utilisation d'une API dépréciée

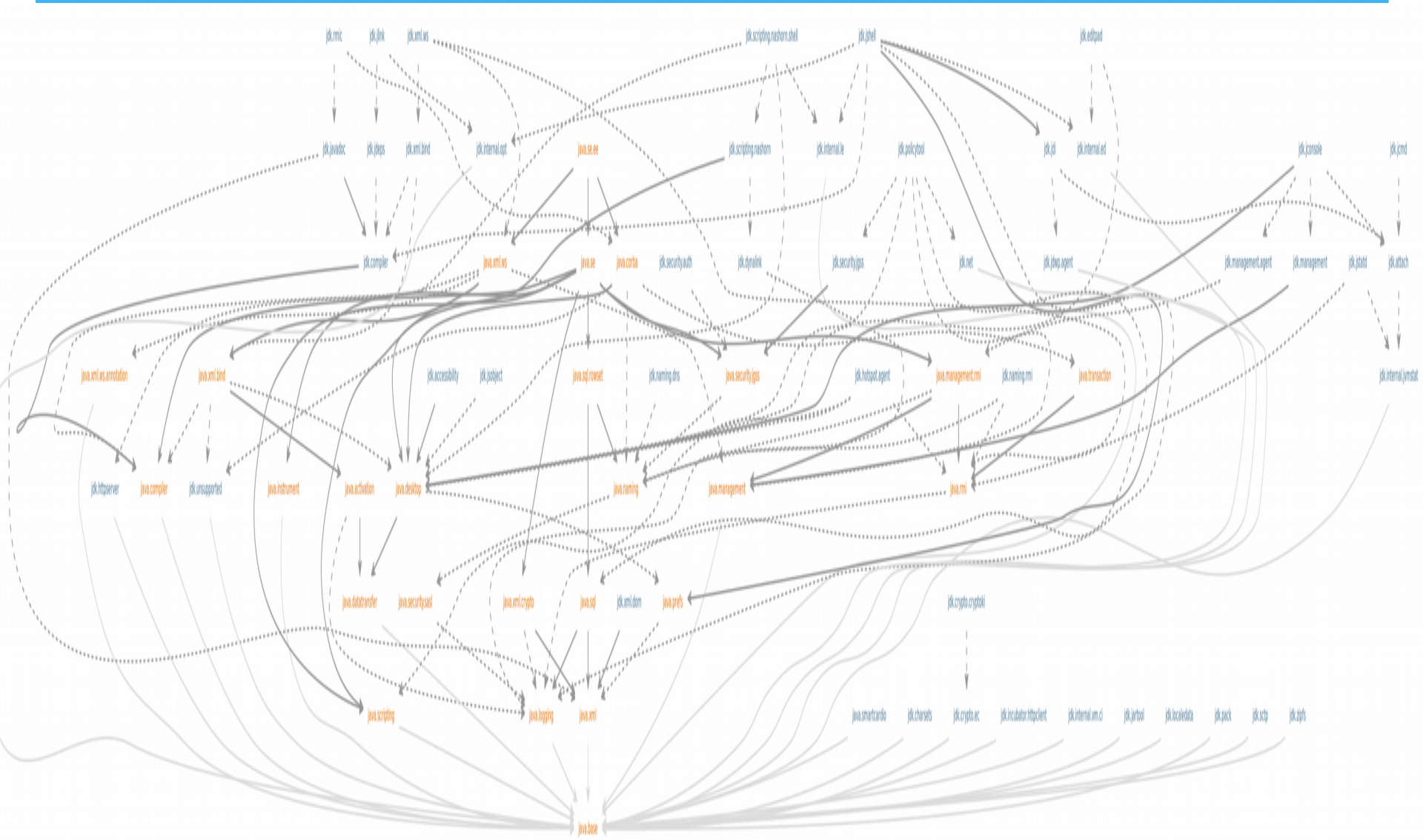
Java 9

Java Platform Module System
Jigsaw

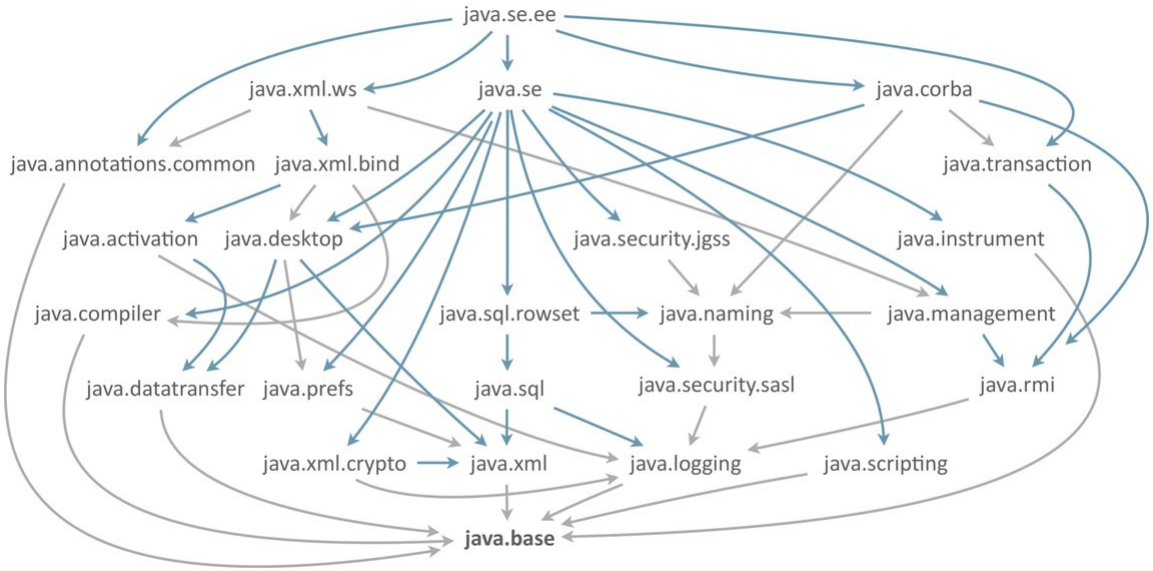
Introduction

* Principale nouveauté de Java 9
* Découpe le JDK en modules
* Les modules remplacent le classpath

Avant Jigsaw



Depuis Jigsaw



module-info.java

* Situé à la racine des sources
* Décrit les dépendances et les packages exportés par le module
* Ces dépendances sont vérifiés à la compilation et à l'execution
* Permet de s'assurer que seul la partie « api » d'un module est visible des autres modules
* Une classe publique n'est pas forcément visible des autres modules

Java 9

Autre

G1 GC

* Garbage first garbage collector
* GC par défaut depuis Java 9
* Le tas est découpé en régions de la même taille
* Chaque région est du type Eden, Survivor, Old ou Humongous
* Permet de réduire les temps de pause (Stop the world)

Jshell

* REPL (Read-Evaluate-Print-Loop)
* Permet de découvrir Java et les nouvelles API à travers d’une console

Javadoc

* En HTML 5
* Enfin une barre de recherche
* https://docs.oracle.com/javase/9/docs/api/index.html?java/util/Optional.html

Autre features

* Multi-release jar: Pouvoir fournir plusieurs version d'une classe en fonction de la version JRE
  * Utilisé par Apache Lucene 7. 3
* Jlink: Génération d'un JRE avec juste les modules nécessaires à l'application
* Ajustement de la mémoire pour docker
* Stack walking API: Pouvoir récupérer les appels de la pile sans générer une exception

Autre features

* Versioning
  * 1.8.0_121
  * 9.0.4 ($MAJOR.$MINOR.$SECURITY_PATCH)
  * Pas forcément plus clair, pourrait encore changer
* ‘_’ n’est plus un identifiant autorisé
* HTTP 2.0

Java 9

Contraintes de migration

Accès interdit aux API internes

* Warning suivant dans le console:
* **`WARNING: An illegal reflective access operation has occurred`**
* Le mieux est d’éviter les API internes
* Jdeps peut aider à trouver l’usage d’API interne
* Possibilité d’utiliser des flags pour accéder à certains packages

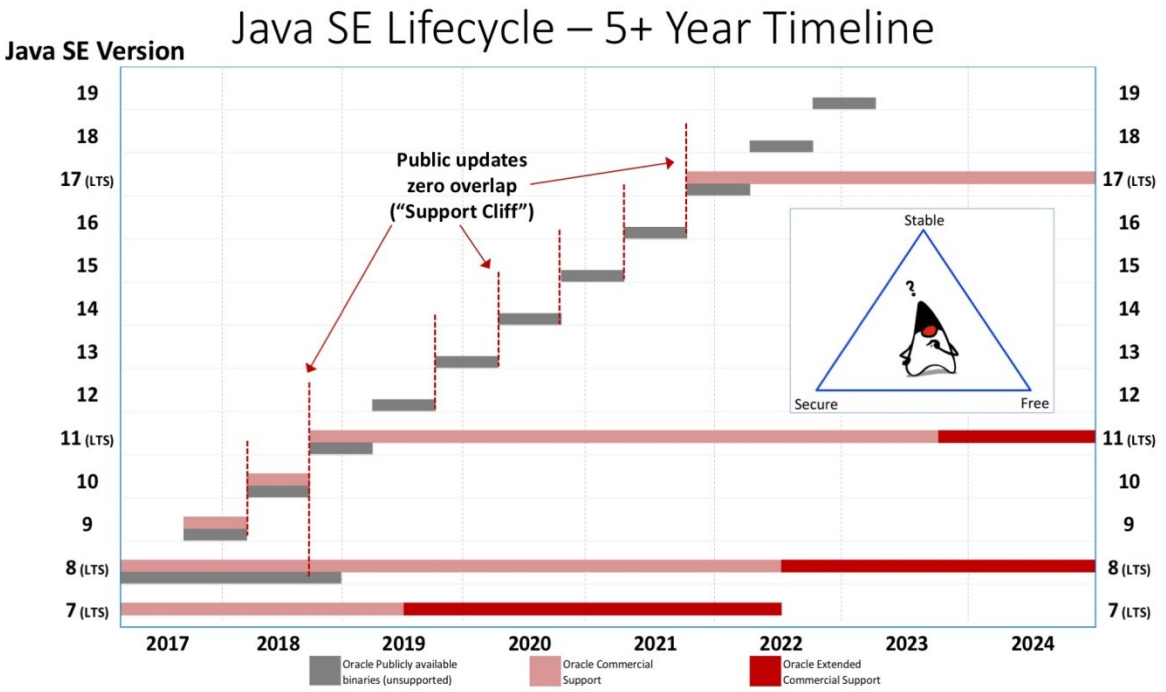
Accès aux API privées du JDK

* Les APIs internes du JDK ne sont plus accessibles
* Celles qui n'ont pas de remplacement sont dépréciés (et susceptible d'être supprimés dans les prochaines versions).
* `sun.misc.Unsafe` reste accessible pour l'instant car pas de remplaçant

Package défini dans plusieurs Jar

* Un package ne peut être défini que dans un seul module
* Ex `java.xml.ws.annotation` est défini dans plusieurs jar

Quand migrer vers Java 9 ?



Java 10

Java 10

* Release prévue le 20 mars 2018 (Release tous les 6 mois)
* Application CDS
* Changements internes:
  * Unified GC API

Inference de type

* `var total = x + y;`
* Utilisable dans les cas suivants:
  * variable locale initialisée
  * Indice de boucle for
  * Scope local d'une boucle conditionnelle
* var n'est pas un mot clé

Inference de type

* `var total = x + y;`
* Utilisable dans les cas suivants:
  * variable locale initialisée
  * Indice de boucle for
  * Scope local d’une boucle conditionnelle
* var n’est pas un mot clé

Merci !