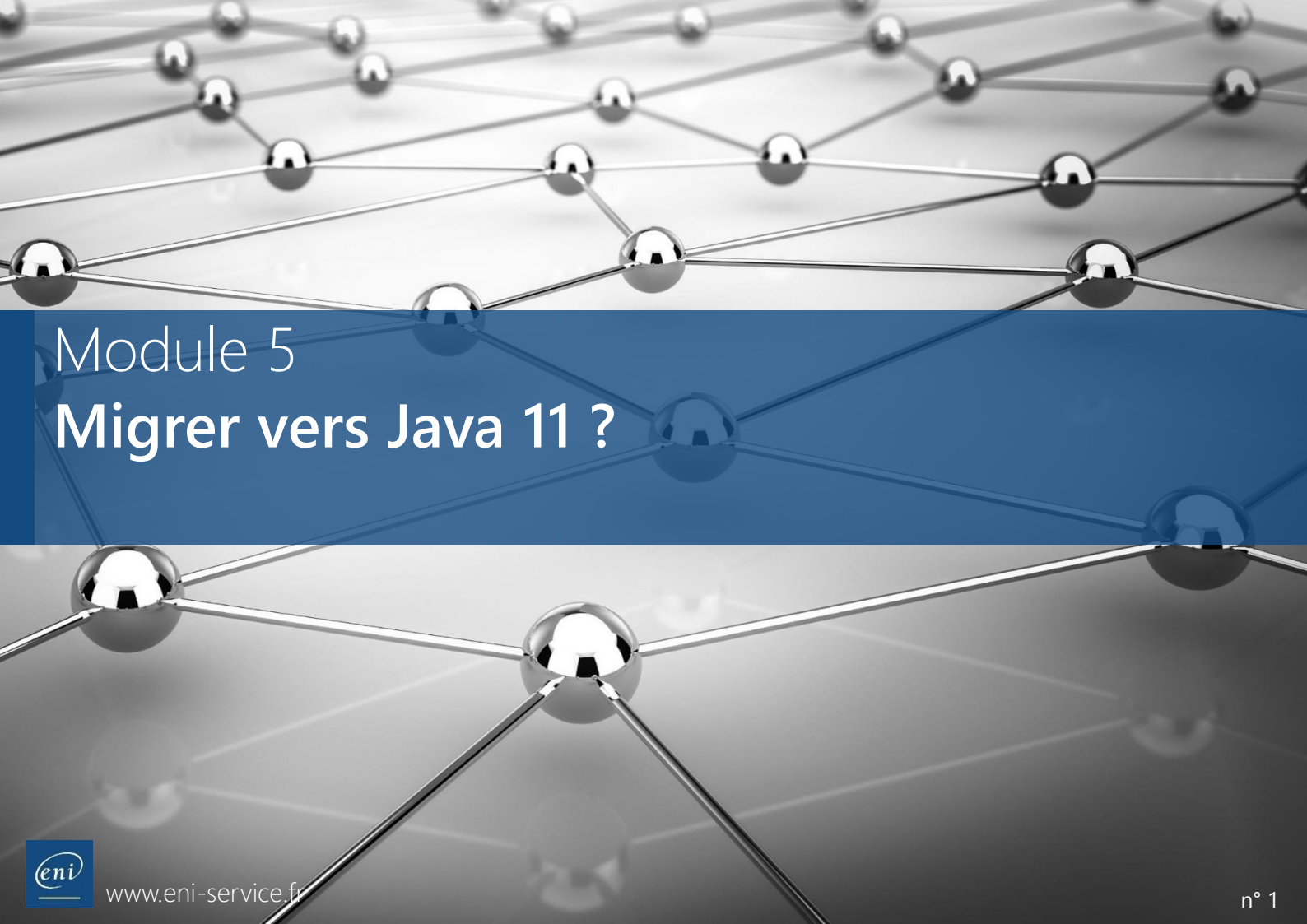

Module 5
Migrer vers Java 11 ?

Faut-il migrer vers Java 11 ?

- Au vu des dates annoncées pour le support de Java 8, c'est pertinent !

Version	Release date	End of Free Public Updates[5][6]	Extended Support Until
Java SE 8 (LTS)	March 2014	**January 2019 for Oracle (commercial)** December 2020 for Oracle (personal use) At least September 2023 for AdoptOpenJDK	March 2025
Java SE 9	September 2017	March 2018 for OpenJDK	N/A
Java SE 10	March 2018	September 2018 for OpenJDK	N/A
Java SE 11 (LTS)	September 2018	At least September 2022 for AdoptOpenJDK	September 2026
Java SE 12	March 2019	September 2019 for OpenJDK	N/A
Legend: Old version	Older version, still supported	**Latest version**	

- La migration permet de profiter des nouveautés apportées depuis Java 9
  - Nouvelles API
  - Améliorations du langage
  - Modularisation
- Et d'un runtime (JVM) plus performant

Comment migrer ?

- Etre à jour de ses versions !
  - Java evidemment ……..
  - IDE
    - Le IDE du marché : Eclipse, NetBeans, IntelliJ, sont prêts pour Java 11
      - Mais dans leurs versions récentes seulement !
  - Outils de construction
    - Maven, Gradle, ..
  - Librairies
    - Il est important de vérifier la compatibilité des librairies utilisées dans les projets et si besoin, les mettre à jour.
- Ajouter les dépendances manquantes
  - La modularisation du JDK depuis la version 9 implique des perturbations dans la localisation des classes
    - Ajouter les modules manquants au module path pour les projets modulaires
    - Pour les projets non modulaires, trouver la librairie externe à ajouter au CLASSPATH (dépôts Maven par exemple …)

Démarche

- 3 phases incrémentielles pour migrer entièrement vers Java 11:
  - Exécuter une application Java existante avec JDK 11
  - Compiler l'application avec Java 11
  - Modulariser (éventuellement) l'application pour qu'elle utilise le système de module de Java 9.

- Pour toutes ces étapes, il sera peut-être (probablement) nécessaire de mettre à jour les dépendances externes

Modulariser ?

- L'apport des modules avec Java 9 ne remet pas en cause la notion de CLASSPATH !
- Il n'est pas utile de modulariser les projets existants
  - A moins de voir bénéficier du système de fermeture de packages apporté par la modularisation
- Cependant, c'est pertinent pour les nouveaux projets
  - Au pire, ils sont prêts pour bénéficier des apports de la modularisation
  - Les livrables produits peuvent être plus petits
    - Y compris le JRE nécessaire
  - La vitesse de chargement de l'application peut être améliorée

Quelques ressources pertinentes …

- https://docs.oracle.com/en/java/javase/11/migrate/index.html
- https://winterbe.com/posts/2018/08/29/migrate-maven-projects-to-java-11-jigsaw/
- https://medium.com/criciumadev/its-time-migrating-to-java-11-5eb3868354f9
- https://blog.codefx.org/java/java-11-migration-guide/
- https://www.infogain.com/making-an-impact/java-8-to-11-a-migration-story/