

# Java 8 ne va plus recevoir de mises à jour et correctifs de sécurité à partir de septembre

à moins de passer à un support commercial

**Mise à jour le 04/02/2018 :** Oracle ne va pas cesser de livrer des mises à jour publiques pour JDK 8 avant janvier 2019

*De manière exceptionnelle, JDK 8 ne sera pas concerné par les nouvelles mesures annoncées par Oracle en ce qui concerne le support du JDK compte tenu du nouveau cycle de publication adopté. Le journaliste Paul Krill, spécialiste du monde Java, a en effet fait remarquer le 1<sup>er</sup> février qu'Oracle a prolongé les mises à jour publiques du JDK 8 jusqu'en 2019, et c'est ce qu'on constate d'ailleurs en se rendant sur le site officiel d'Oracle : JDK 8 va continuer à recevoir des mises à jour publiques au moins jusqu'en janvier 2019.*

*Par rapport à la date initialement annoncée de septembre 2018, ça fait au moins quatre mois de plus, mais cela devrait permettre au JDK 8 de recevoir des mises à jour trimestrielles supplémentaires gratuites en octobre 2018 et janvier 2019. Il est également important de préciser qu'Oracle pourrait continuer à proposer des mises à jour publiques pour JDK 8 bien au-delà de janvier 2019. Oracle va peut-être suivre l'utilisation de cette version du JDK avant de se décider.*

Source : [Oracle](#)

**31/01/2018 :** L'année dernière, Oracle [a annoncé](#) un certain nombre de changements en ce qui concerne le développement de Java SE. Expliquant la nécessité d'accélérer le cycle de publication du JDK pour rester compétitif, Oracle a préféré passer à une nouvelle version de Java avec de nouvelles fonctionnalités tous les six mois, en mars et en septembre. Et ces versions de fonctionnalités (feature release) seront suivies par des mises à jour (update release) chaque trimestre, comme c'est le cas maintenant en janvier, avril, juillet et octobre. C'est le nouveau cycle de publication qui a été adopté après la sortie de Java 9 en septembre dernier.

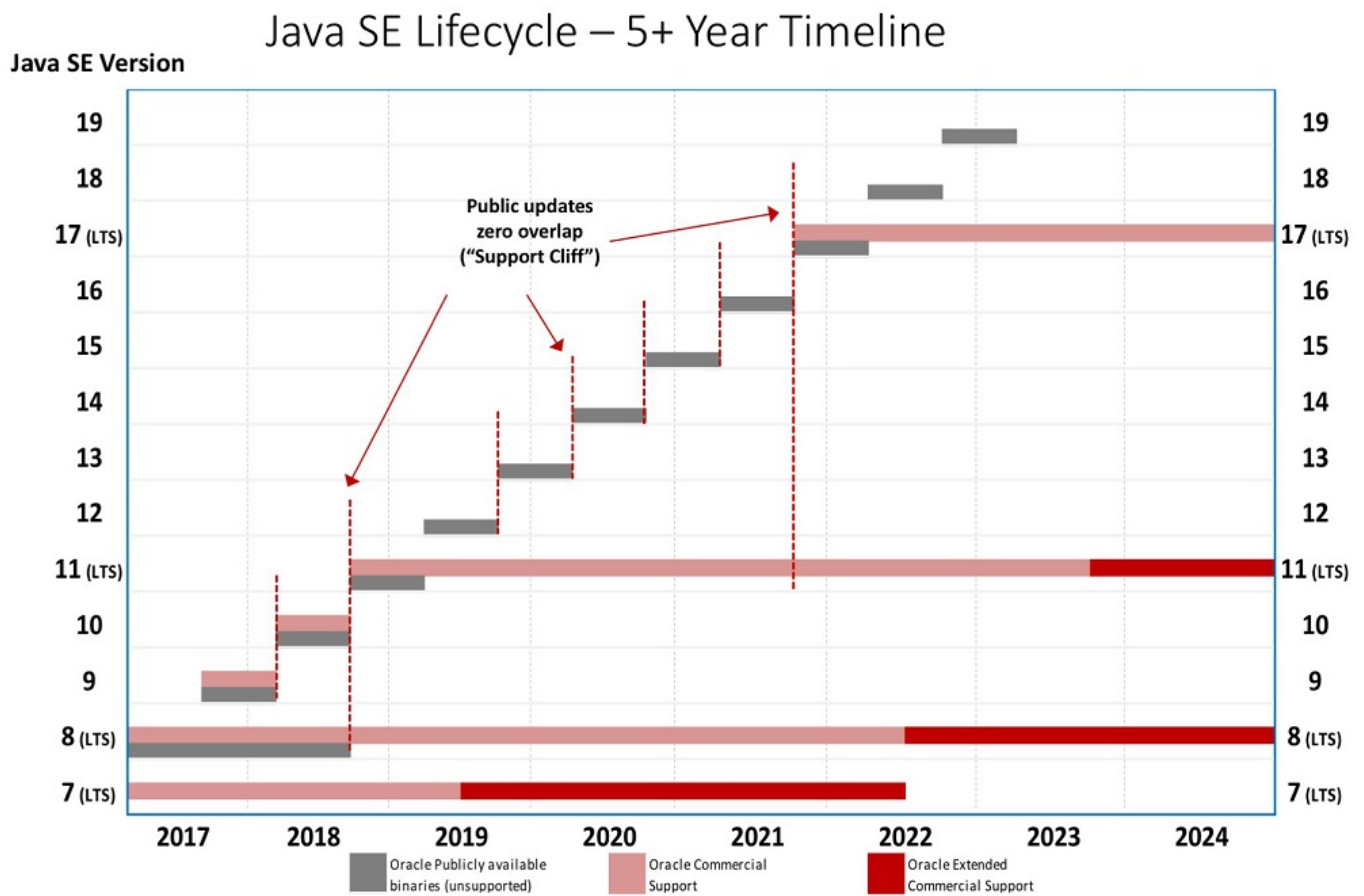
C'est probablement une très bonne nouvelle, étant donné que cela signifie que les fonctionnalités seront intégrées dès que possible et que les développeurs auront accès à de nouvelles fonctionnalités beaucoup plus vite qu'auparavant. Toutefois, l'adoption de ce nouveau rythme de publication a conduit Oracle à prendre d'autres décisions qui pourraient ne pas réjouir autant les développeurs et les entreprises : en septembre, lorsque le JDK 11 (version 18.09) sera publié, il n'y aura plus de mises à jour publiques, y compris les patches de sécurité, pour le JDK 8. Si l'on se fie aux [chiffres de Semaphore CI](#), Java 8 était encore utilisé fin 2017 dans plus de 80 % des projets commerciaux. Il y a donc beaucoup d'utilisateurs qui seront affectés par cette décision.

Le fait est que désormais quand une nouvelle version de Java sera disponible, la version précédente va immédiatement cesser de recevoir des mises à jour publiques. Pour mieux comprendre ce que cela veut dire, rappelons un peu comment se sont passées les mises à jour jusqu'ici. Lorsque le JDK 6 a été publié, les mises à jour du JDK 5 ont continué à être publiées pendant près de trois ans (deux ans et onze mois pour être précis). Avec le JDK 7, le JDK 6 a continué à avoir des mises à jour publiques pendant un an et neuf mois. Le JDK 7 a ensuite eu treize mois de mises à jour publiques après la sortie du JDK 8.

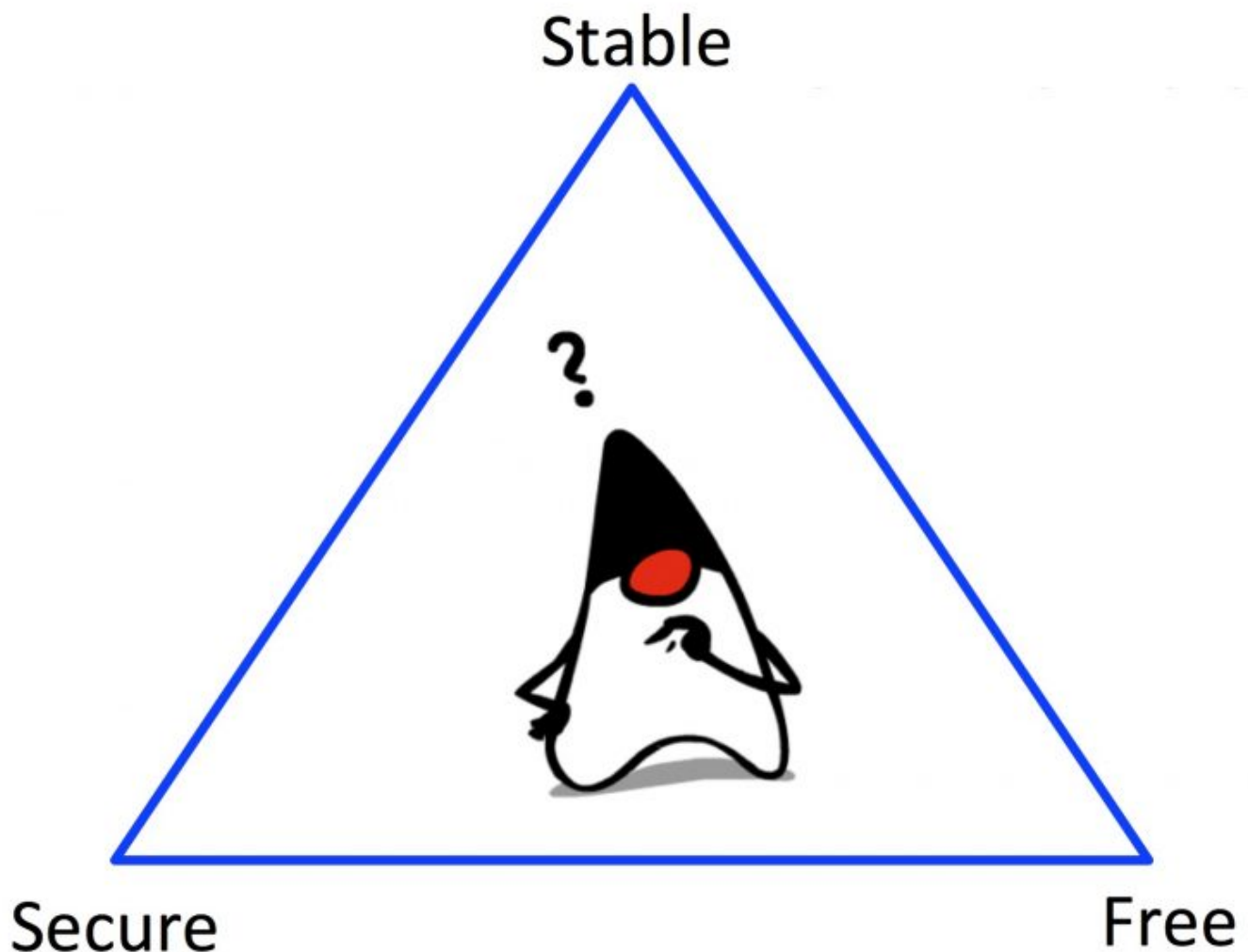
Ce que faisait Oracle après la sortie d'une nouvelle version du JDK c'était d'accorder une période de grâce, qui dure généralement plus de 12 mois. Pendant cette période de grâce, la version précédente du JDK continue à recevoir des mises à jour publiques, non seulement pour permettre une transition en douceur, mais encore pour permettre à la nouvelle version d'être suffisamment testée en production afin que tous ses petits problèmes soient corrigés ; et c'est ce qu'attendent la plupart des utilisateurs avant de migrer. Il faut en noter que dans la pratique, de nombreuses équipes de développement ne veulent pas prendre le risque de migrer immédiatement après la sortie d'une nouvelle version. Seules les équipes qui ont vraiment besoin d'une nouvelle fonctionnalité migrent immédiatement. Quant à celles qui ont tendance à toujours migrer vers les nouvelles versions, elles attendent quand même quelques versions mineures avant de se lancer, parce que les versions x.0.0 sont susceptibles d'avoir certains problèmes.

Mais avec le nouveau rythme de publication du JDK, Oracle estime que l'effort d'ingénierie pour rendre les mises à jour disponibles pour de nombreuses versions antérieures serait insoutenable. C'est pour cela qu'à la sortie d'une nouvelle version de fonctionnalités (feature release), la version précédente ne va plus recevoir de mises à jour. Pour atténuer le problème, Oracle a toutefois décidé de passer à un modèle LTS (support à long terme). Des versions spécifiques du JDK seront des versions LTS, ce qui signifie qu'elles auront des mises à jour pendant trois ans (jusqu'à la prochaine version LTS). Toutes les autres versions

intermédiaires (les versions de fonctionnalités) ne seront mises à jour que pendant six mois (jusqu'à la prochaine version de fonctionnalités). Et pour synchroniser le nouveau système, le JDK 8 a été classé comme une version LTS, ce qui veut dire qu'après la sortie du JDK 11 (LTS) en septembre, le JDK ne va plus recevoir de mises à jour publiques.



Comme l'explique Simon Ritter, ingénieur chez le fournisseur de JVM et runtime pour les applications Java Azul Systems, cela veut dire que Java sera toujours stable, sécurisé et gratuit, mais vous devez maintenant choisir deux de ces trois qualités.



« Supposons que vous êtes plus préoccupé par la sécurité (comme vous devriez l'être), mais que vous ne voulez pas dépenser d'argent. Pour ce faire, vous devrez passer aux nouvelles versions du JDK dès leur sortie. En l'absence de chevauchement de mises à jour (pour les versions Feature ou LTS), vous devez effectuer cette opération pour pouvoir installer tous les correctifs de sécurité. Malheureusement, sans chevauchement, vous devez passer à une version JDK qui n'a pas eu le temps d'être testée dans des environnements de production réels, de sorte que vous perdez la stabilité que vous aviez », dit-il.

« Alternativement, vous pouvez favoriser la stabilité et le coût nul. Encore une fois, c'est possible. Tout ce que vous faites est de continuer à utiliser la version précédente LTS après la sortie d'une nouvelle version. Vous continuez à avoir la stabilité à laquelle vous êtes habitué, vous ne payez rien, mais vous n'obtiendrez pas de correctifs de sécurité, car ils ne seront plus disponibles pour cette version. » Et enfin, poursuit-il, « si vous voulez la stabilité et la sécurité, ça ne va pas être gratuit. Vous pouvez continuer à utiliser la version précédente de LTS, mais l'accès aux correctifs ne sera disponible que via un contrat de support commercial. » Azul Systems a pour sa part expliqué comment il compte offrir à ses clients un support étendu de différentes versions du JDK.