Key Considerations in using Oracle or Non-Oracle Java

Description

What are the Key Considerations in using Oracle or Non-Oracle Java is becoming an important question that needs answers. This is because Oracle Java license changes have made Java costs quite steep. The increase in Oracle Java costs are driving users to consider other options.
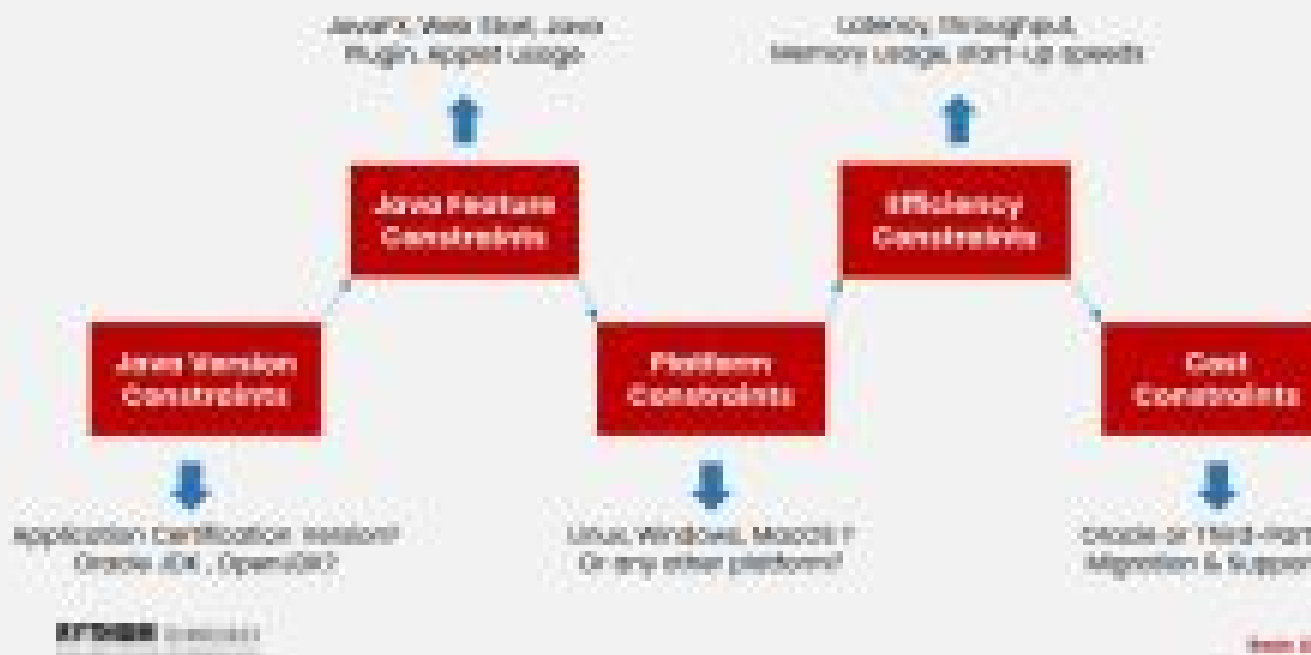
We have been working with many customers on Java license analysis. During these engagements much of the discussion veers round this very same question: What are the Key Considerations in using Oracle or Non-Oracle Java?

Key Considerations in using Oracle or Non-Oracle Java

While there are quite a few, we have shortlisted the following constraints that need to be looked into before deciding to choose any non-Oracle Java:

- Java Version Constraints
- Non-Oracle Java Feature Availability Constraints
- Platform Constraints
- Efficiency Constraints
- Cost Constraints

Java Version Constraints:

- What is the version on which your application is certified?
  Is it certified on Oracle JDK and OpenJDK as well?

- Is the application available on the recent and latest LTS versions?
  i.e., JDK 17 and JDK21(currently JDK21 is the latest)

Note: The applications not being certified on OpenJDK does not necessarily force you to use Oracle Java. We have customers who are:

- ○ Pushing vendors to test and certify applications on non-Oracle Java (i.e., versions developed on OpenJDK)
  ○ There are vendors of Java who are willing to make the application work on their versions of Java.

Non-Oracle Feature Constraints

These are issues specifically connected to:

- Java FX
- Java Web Start
- Java Plug-in
- Applet usage

Java FX, Web Start and Plug-ins

- - Of the above, Oracle has moved JavaFX, Web Start & the plug-ins to open-source projects starting with JDK11. The concern is for applications prior to this version requiring the use of any of the above features.
    - Interestingly, most application vendors who used these features have moved to test and certify the applications on the latest versions of OpenJDK. Hence these are not very concerning today.

The need for Applets is a legacy feature.

- - The 'applet' element was deprecated in HTML 4.01 and rendered entirely obsolete by HTML5.
    - All web browsers have dropped support for Java applets.

So, these are not critical issues today, but still important considerations to look into in enterprise applications.

Platform Constraints

All Java distributions are available on

- Linux
- Windows
- MacOS

In case you have applications on any other platform, it is important to check for the available of specific distributions of Java.

Efficiency Constraints

This is an interesting constraint because many of the non-Oracle Java distributions started off to provide Java builds with better performance than Oracle JDK.

For example:

- Azul has a low latency & high-throughput garbage collector with reduced memory usage.
- IBM's Sumeru uses reduced memory and starts-up faster.
- Amazon's Coretto has enhancements that improves performance and stability

- Apart from the above, there is also **GraalVM**:
    - GraalVM is open-source high performance JDK platform.
    - GraalVM runs on HotSpotJVM
    - GraalVM can also run other languages like JavaScript and Python
    - Oracle, to provide better performance, is today including an Oracle JDK implementation of GraalVM.

Cost Constraints

Cost considerations should be looked at in two dimensions:

- Direct Costs
- Indirect Costs

Direct Costs

The direct costs are easy to calculate because these are the price difference between:

- Oracle's Java proposals
- Java proposals from third party Java Vendors
- Free Java from Third Party Vendors
- Free Java from Oracle
- Free Java from OpenJDK

Indirect Costs

The indirect costs are more difficult to calculate because they are dependent on:

- Migration Costs from Oracle to non-Oracle JDKs
- Support considerations of Oracle and non-Oracle Java distributions.

What should customers do?

Before deciding to deciding on key considerations of deciding between Oracle and Non-Oracle Java, customers should

- Conduct an analysis that gives you complete visibility of Java deployments normalized by version, Oracle entitlements and third-party entitlements.
- Conduct a cost-benefit analysis of migrating away from Oracle.
- Decide on how to manage the accidental installation of Oracle's commercial JDK versions.

Java License Analyzer

We have also developed a tool called the Java License Analyzer that marries most of the above components of our service. You may see the features here

Java License Analyzer, How much should you pay for Java

Image not found or type unknown

Java License Analyzer, How much should you pay for Java

You might want to read more about our CEO Sheshagiri Anegondi (Sheshu). He is amongst the foremost Oracle License Experts globally. Contact him or our other Oracle license experts to know how much to pay for Java.

While Java license changes are the hottest Oracle Licensing issues just now, we also do work on Oracle ULAs, Audit Defense and Vendor Management.

Read this if you want to know the 12 steps for ULA Certification. Read here

About Rythium Technologies

Rythium Technologies is a leading provider of software license management (LMS) services. We help our clients reduce their software costs, manage their IT assets, and achieve compliance. Our Oracle LMS services include: Oracle ULA services, Java License Advisory, Oracle license audit defense, Oracle contracts management and Software Procurement consulting. *We have worked on more than 200 Oracle license management projects in the past 7 years.*

Author
adminlicens