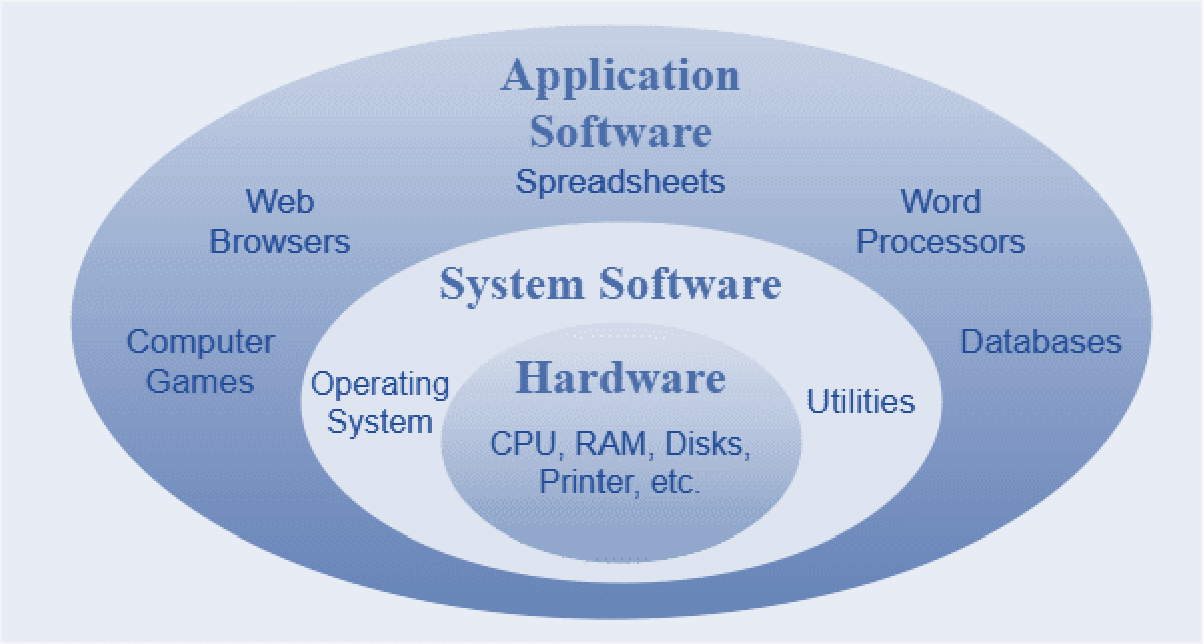
3-Palform and Platform Dependency

Java is renowned for its platform independence, often summarized by the term WORA (Write Once, Run Anywhere). Besides this feature, Java supports object-oriented programming, is an architecturally neutral language, is portable, and is robust, among many other advantages. Before diving deeper into these features, it is crucial to understand what platform and platform dependency mean. Once we grasp these concepts, we can appreciate how Java achieves platform independence.

What is a platform?

A platform is a combination of both software and hardware systems:

- **Software:** This typically refers to the operating system (OS), such as Linux, Windows, or macOS.
- **Hardware:** This includes components like processors. For example, Intel and AMD Ryzen are common, while Apple uses ARM architecture-based microprocessor chips (Apple Silicon).

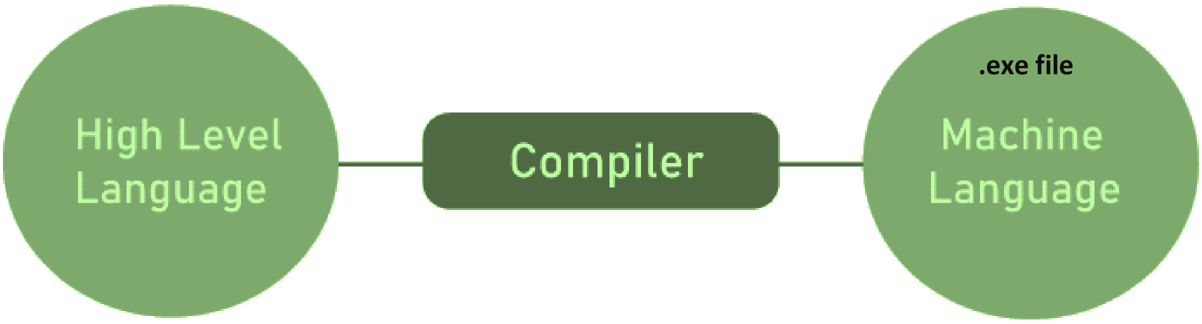


As developers, we primarily focus on the software aspects of platforms.

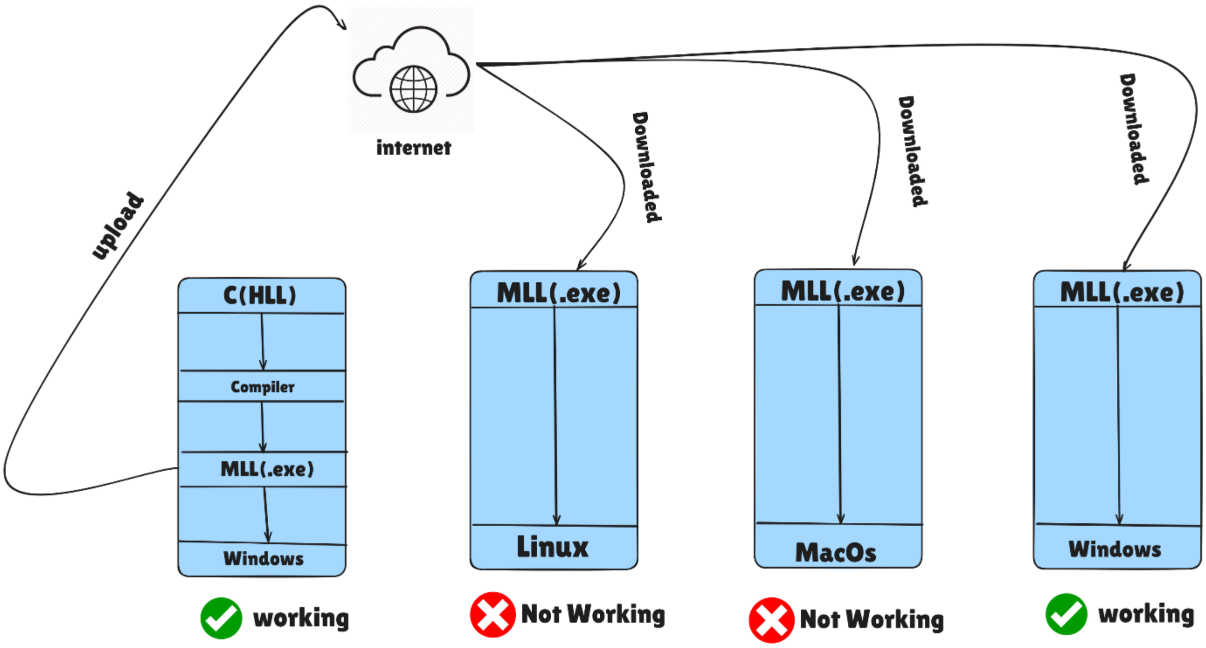
What is platform dependency?

To understand platform dependency, let's consider an example:

Suppose we have a system running Windows OS, and we build an application in the C language (a High-Level Language, or HLL). For this application to execute, it must be converted to machine-level language (MLL) that the processor can understand. This conversion is done by a compiler, resulting in an executable file with a .exe extension.



Now, if we try to run this .exe file on systems with different operating systems, like Linux or macOS, it won't work. The executable was created specifically for Windows OS, making it platform-dependent.



How Java Achieves Platform Independence

In contrast to languages like C, Java claims to be platform-independent. But how does Java achieve this?

Both Java and C use HLL for programming, converting it to MLL for execution. However, the converted MLL is typically platform-dependent. Java overcomes this with a unique approach, which we will explore in the next chapter when we delve into the internal architecture of Java and its workings.

Stay curious and keep learning.