

THE STATE OF JAVA IN THE ENTERPRISE



In recent months, it has been revealed that Java is once again undergoing some significant changes.

In addition to a vote that changed the name of enterprise Java, currently Java EE, to Jakarta EE -- the brand is having its governance handed over to The Eclipse Foundation. For those used to coding robust, complex enterprise applications using Java, tinkering with the product and breathing new life into an already overcrowded [DevOps](#) marketplace may sound like a worst case scenario.

But, Java may not be defunct yet as the goal of the changeover is to spearhead new innovations for Java developers to move forward in a cloud and container-driven enterprise world.

Here's a look at the future of Java for the enterprise and what it may mean for you as a developer.

Java Today: A Quick Review

As you already know, Java is a computing platform that relies on object-oriented programming or OOP. When it was first productized for the internet by Sun Microsystems in 1995, Java offered a welcomed relief from the headache of other programming languages like C++ in designing stand-alone applications.

Indeed, Java presented a lot of firsts to developers in the 90s who were happy to have a cross-platform language to churn out production code. It was initially designed to have as few implementation dependencies as possible, which led to the term "write once, run anywhere" (WORA). This means that compiled Java code can run on all platforms with no need for recompiling

the code. In addition, Java-based applications became known for their speed and scalability.

But by 2010, in spite of a growing number of enterprises using Java EE to write web applications, Forrester was [calling it a "dead end"](#) for enterprise businesses, recommending that they find new technologies to overcome the complexity and inflexibility of Java applications.

And while Java is still used for some web applications today, it is true that more and more enterprise organizations have made the transition to approaches that are agile and which allow for faster delivery of applications in the cloud.

Furthermore, container management platforms and [PaaS](#) have become the order of the day.

A Look at Enterprise Development Trends

Enterprise app development in today's competitive business environment is practically a given. And frequency of deployment is more critical than ever. In fact, you may already know that the highest-performing IT organizations deploy more applications more frequently, such as multiple deployments per day versus between once a month and once every six months for lower performing IT organizations.

If businesses aren't writing software for their own use, they are bringing programs to a mobile marketplace for their consumer-users.

In many cases, enterprise businesses find they are doing both. As such, the move from traditional views of IT to a culture centralized around DevOps and continuous delivery is afoot.

In turn, this paradigm shift has paved the way for new technology and enterprise development trends. These 2018 development trends pose challenges that must be answered for any future development of Jakarta EE:

Focus on Security

With governance changing hands to The Eclipse Foundation, how will a new Java-based platform ensure security? This is the questions burning on the minds of many developers as multi-cloud environments and integrations give way to security challenges for enterprise businesses.

In order to be competitive, Jakarta EE will need to be innovative with a focus on keeping applications safe from cyber threats.

Progressive Web Apps (PWAs)

As PWAs, or websites that operate with all the online and offline functionality of a mobile app, take hold of the enterprise world how will Java web apps compete?

Current Java instances are much bulkier than the lightweight platforms used to code in the cloud. But Jakarta hopes to overcome one of its biggest drawbacks by focusing on cloud native development.

Development Passed on to End-User

More and more, development is shifting to the end-user to create a completely customized enterprise app experience -- this is native to container, PaaS and other web-based development

platforms.

Critics of Java may be wondering if Jakarta can meet this requirement?

Traditional instances of Java require deep knowledge of coding languages and technical ability, but the trend is for applications to be easier and easier to customize for a dynamic marketplace. Staying ahead of the competition means planning for end-user development.

Room to Grow

As innovations like blockchain and the Internet of Things begin to impact the everyday lives of end-users, any new Java development platform should be flexible with room to grow.

It remains to be seen if Java can make a come back by meeting the demands of a changing enterprise market, but with The Eclipse Foundation behind the wheel we should expect to see new innovation.

Open Source

In recent years, open source technologies have gained widespread adoption in a number of areas, including virtualization, supercomputing, operating systems, databases, web servers and more. And there's a continued clear demand for open source as a critical part of the developer landscape today. The Eclipse Foundation has made it clear it intends to place a significant focus on leveraging the open source community for Jakarta EE.

The Eclipse Foundation and the Future of Java in the Enterprise

All indications suggest the Eclipse Foundation is focused on taking Java into the cloud to compete with containers and PaaS.

The reason that Oracle released governance of Java to the Eclipse Foundation was strategic. The goal was to take Java, which is a present day platform central to many enterprise operations, and keep it relevant by allowing the foundation to focus on modern innovation.

The Eclipse Foundation is expected to dive deep into MicroProfile, a move to make Jakarta EE more lightweight and therefore cloud-ready.

The vision is to create a cloud platform where Java developers don't have to know snippets of thirty different languages to get one program made. Instead, they can code in a familiar language like Jakarta EE, with a software partner who has the Java experience to deliver services in the cloud. But critics wonder if adding a new platform will mean they now have to know thirty-one languages, instead of thirty.

In a decree release by the Eclipse [EE4J Project Management Committee](#), it is expected that open-source communities will embrace these changes, offering technologies that can be adopted into the platform such as the following:

- Docker
- NoSQL; and
- Kubernetes

This gives developers the option to use standardized technology to create web apps and databases

in a Jakarta EE platform.

Jakarta EE

As mentioned above, Java EE became Jakarta EE in 2018 under the leadership of The Eclipse Foundation. To date, they have already relicensed Java projects from Sun and the Foundation is now in the development phase of Jakarta EE, working on more than 30 projects using Foundation resources, and preparing them to ship all at once. The goal is to build a [compatible implementation](#) that surpasses the current tech compatibility kits (TCKs) for Java EE 8 CTS.

In addition to 39 new projects to ship and 110 repositories to transfer, more than half of which is [reportedly complete](#), Jakarta EE faces a number of challenges before reaching the open market. These include checking licenses and making sure all bug-fixes are implemented before it reaches end-users.

As you can see, there's still a ways to go before developers have a tangible cloud-native Java version for enterprise, but The Eclipse Foundation seems confident their contributions will enhance the already vibrant development community of programming in the cloud.

Final Thoughts: Say Goodbye to Java as We Know It

From its early inception in the 1990s, to the advent of internet applications and OOP, Java has been a leader in enterprise coding technology. And even with some experts calling it an artifact, it still hasn't been ousted as a central figure in the tapestry of modern-day application architecture.

This is because Java provides a cross-platform standard for developers to create rich and robust applications for their internal and external customers to use in the day to day.

But given the shift to the fast moving pace of continuous delivery, it's no surprise that Java has had a hard time keeping up.

However, things may be about to change, as The Eclipse Foundation enters the marketplace with a peek behind the curtain of Jakarta EE. Using Jakarta EE, it's likely developers comfortable with Java will be able to do many of the same functions as they can with container platforms and PaaS. The Eclipse Foundation themselves said that [Jakarta EE's mission](#) is "more frequent releases, lowered barriers to participation, and putting the community back into the platform."

This certainly sounds like a solid plan but only time will tell how successful yet another (confusing) name change will be for Java. Indeed, as it stands now, there is no great consensus on what impact this could have on the enterprise development community as app developers prepare for blockchain, IoT and other innovations.