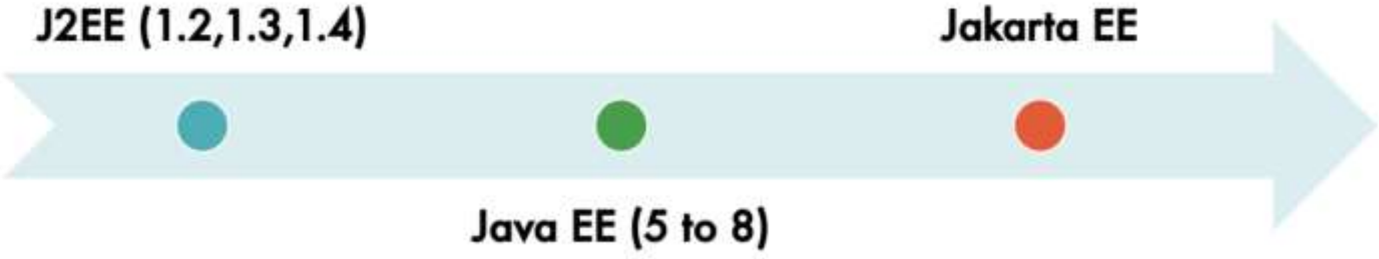
EVOLUTION OF JAKARTA EE

COMAPRING WITH J2EE AND JAVA EE



Jakarta EE is a platform for developing enterprise-level Java applications. It was originally developed under the name Java Enterprise Edition (Java EE) and was later transitioned to the Eclipse Foundation, where it was renamed to Jakarta EE due to trademark issues.

Here is a brief evolution of Jakarta EE:

1. **Java 2 Enterprise Edition (J2EE):** This was the first version of the enterprise Java platform that was released in December 1999. It provided a set of specifications and APIs for building enterprise-level applications.
2. **Java EE 5:** This version was released in 2006 and introduced several new features, including the Java Persistence API, web services support, and simplified EJB development.
3. **Java EE 6:** This version was released in 2009 and introduced several new features, including web fragments, a new API for managing dependency injection, and support for JavaServer Faces 2.0.
4. **Java EE 7:** This version was released in 2013 and introduced several new features, including WebSocket support, JSON parsing, batch processing, and CDI enhancements.
5. **Java EE 8:** This version was released in 2017 and introduced several new features, including support for HTTP/2, a new security API, and updated JSON binding API.
6. **Jakarta EE:** In 2017, the Eclipse Foundation took over the development of Java EE from Oracle and renamed it to Jakarta EE due to trademark issues.
7. **Jakarta EE 8** was released in 2019, which is essentially Java EE 8 but with a new name and under the governance of the Eclipse Foundation.
8. **Jakarta EE 9** was released in 2020, which introduced a new namespace for the APIs and specifications and removed all references to the old "javax" package namespace, which was associated with Java EE.

a) IMPORTANT SPECIFICATION OF JAKARTA EE:

1. Jakarta Server Pages (JSP)

2. Jakarta Standard Tag Library (JSTL)
3. Jakarta Enterprise Beans (EJB)
4. Jakarta RESTful Web Services (JAX-RS)
5. Jakarta Bean Validation
6. Jakarta Contexts and Dependency Injection (CDI)
7. Jakarta Persistence (JPA)

b) SUPPORTED BY SPRING 6 AND SPRING BOOT 3

1. That's why we use jakarta. packages (instead of javax.)