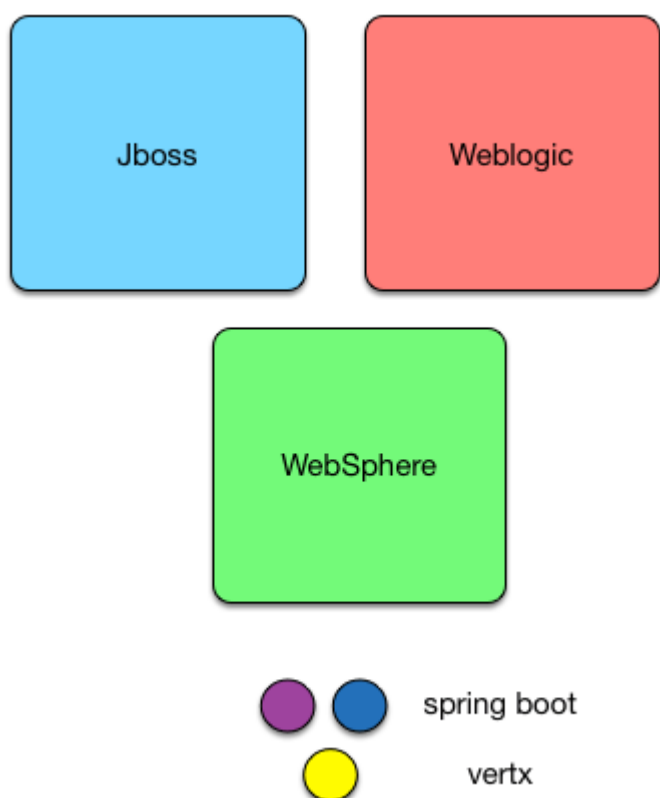


¿Es el fin de los servidores Java EE? .Esta es una pregunta que últimamente esta en el aire . La llegada de arquitecturas basadas en Microservicios y la necesidad de modularizar , esta haciendo que cuando miramos a nuestros servidores de aplicaciones nos entren dudas sobre su futuro. Soluciones como **Spring Boot** o frameworks como **VertX** están comenzando a hacerse un hueco en el mundo Java.

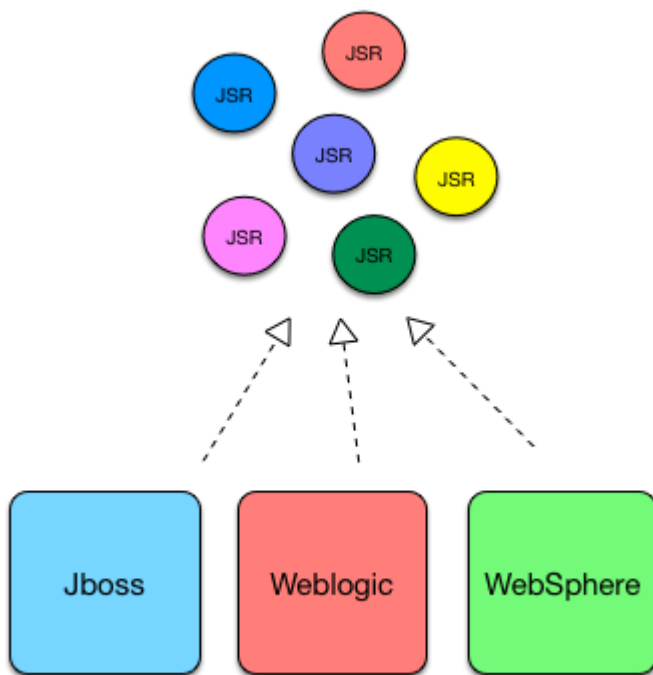
Servidores Java EE y Modularidad

Hoy por hoy cuando miramos a JBoss ,WebLogic y WebSphere nos damos cuenta de que son soluciones monolíticas en donde es todo o nada . Algo muy diferente a lo que los nuevos paradigmas proponen.



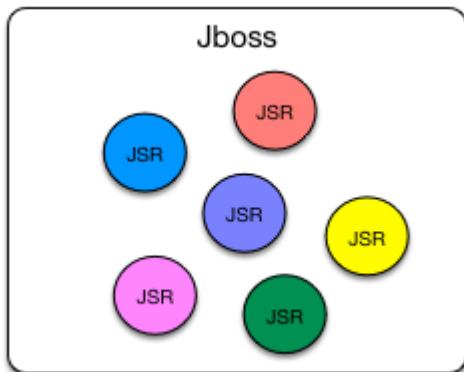
Es aquí donde Spring Boot o Vertx brillan. En un primer momento puede parecer que el futuro de los servidores esta en el aire. La pregunta más importante que nos tenemos que

hacer es : ¿ Son los servidores Java EE la parte más importante de Java EE? . La respuesta es NO , lo más importante de la plataforma Java EE **son sus especificaciones JSR** .Estas son las que construyen la plataforma y los servidores de aplicaciones son meros implementadores.



JSR y Modularidad

¿Son tan monolíticos los servidores Java EE como en un primer momento parecen?. La realidad es que no , ya que tienen que implementar cada una de las JSR.



Así pues no estamos ante una evolución del mercado en la que los servidores de aplicaciones no se puedan adaptar , sino más bien al contrario. Las JSR permitirán a los implementadores de los servidores modularizar su producto con cierta facilidad.

Servidores Java EE, evolución y Tiempo

Recordemos ademas que el mundo del lado del servidor se mueve más despacio que el del lado del cliente. Por ejemplo desde que Spring Framework apareció en el mercado hasta que los EJBs implementaron las mismas soluciones pasaron 4 , 5 años, hoy en día ambos tienen su hueco. Poco a poco los servidores de aplicaciones aumentarán su modularidad y se adaptarán a las necesidades del mercado, uno de los proyectos más interesantes es Wildfly Swarm

Otros artículos relacionados: [¿Tiene futuro JSF?](#) , [Java EE vs Spring Framework](#)