

# Class Loading

# Table of Contents

Each JAIN SLEE Component has its own classloader (`ComponentClassLoader`) in the package named `org.mobicens.slee.container.component.deployment.classloading`. This classloader sees the component classes contained in the component jar (`URLClassLoaderDomain`) by declaring it as the parent classloader, and adding the classes seen by each component it refers. It does not see classes from a component that it does not depend on.

JAIN SLEE defines a class loading domain with the s required in the JAIN SLEE 1.1 container (for example, JAIN SLEE, and ). This domain (`JBoss Microcontainer ClassLoadingDomain`) imports all classes shared in the JBoss Application Server , and acts like the parent domain for all `URLClassLoaderDomains`, which means that a class imported by a SLEE classloading domain will always be used first.

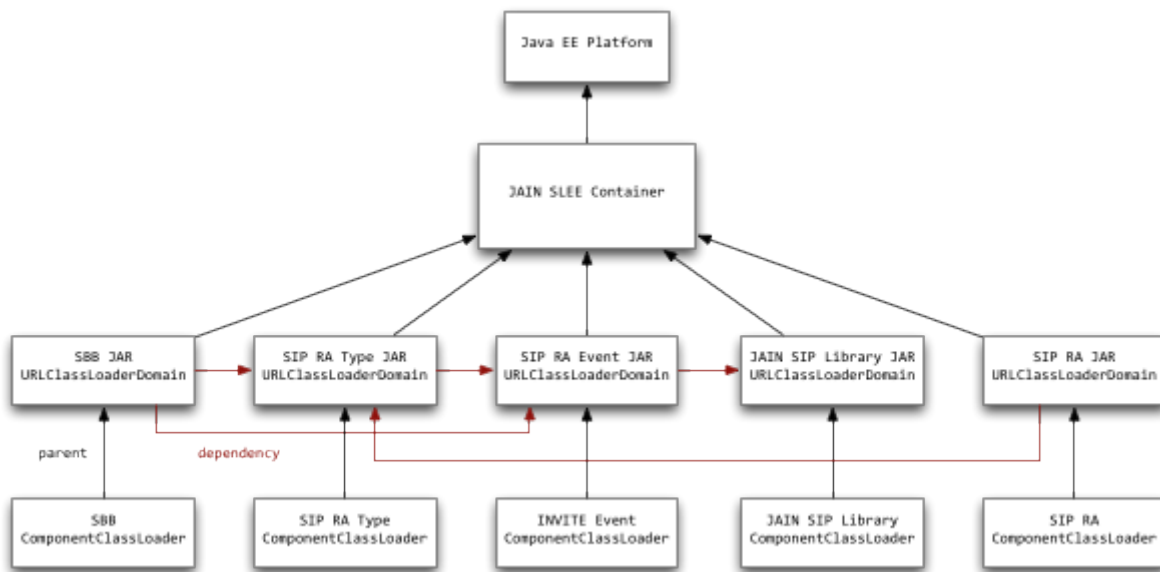


Figure 1. Classloading example in Restcomm JAIN SLEE

- The `SIP INVITE Event` component refers to the `JAIN SIP Library` in its XML descriptor, and its classloader domain depends on the classloader domain of the JAIN SIP Library.
- The `SIP RA Type` component refers to all Events in the SIP RA Event jar in its XML descriptor, and its classloader domain depends on the classloader domain of the SIP Event JAR and inherits its dependencies, including the JAIN SIP library classloading domain.
- The `SIP RA` component refers to the `SIP RA Type` component in its XML descriptor, and its classloader domain depends on the classloader domain of the SIP RA Type Component jar, and inherits its dependencies. This includes the SIP Event jar and JAIN SIP library classloading domains.
- The `SBB` component refers to the `SIP RA Type` component and `SIP INVITE Event` in its XML descriptor. Its classloader domain depends on the class loader domain of the SIP RA Type Component jar, and inherits its dependencies; the SIP Event jar and the JAIN SIP library classloading domains. It also depends on the classloader domain of the SIP Event jar.



JBoss Application Server does not see the classes of deployed JAIN SLEE components. This means that if it exports its classes for components that are complemented with Java EE components, the common classes must be deployed on JBoss Application Server , either directly or included in the Java EE component.