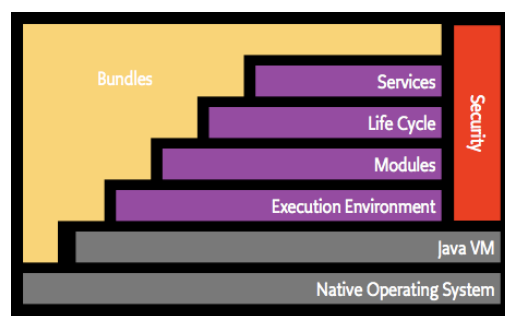


## OSGi : Building Modular Java System

The hardest thing when dealing a big software system is complexity. So we must decompose the problem into the small module. We can take a lot of advantage from modularity. Abstraction and reusable is the most important thing can be achieved from modularity.

OSGi enables us to build application in dynamic and modular ways. We compose the application from components. Each of these components communicates via services. The component hides its implementation from the outside. This is not a new paradigm, but OSGi is the first technology that gains a success which based on the component in solving the real world problem.



**OSGi architecture**

Component in OSGi is called bundle. A bundle is actually no more than JAR file with the standard manifest file. The difference is everything from a previous JAR file is completely accessible from all other JARs. In OSGi this changed a lot. OSGi hides everything in that JAR unless explicitly exported. A bundle that wants to use another JAR must explicitly import the parts it needs. So we must handle dependency explicitly. And we can use same JAR file with different version in the same VM. That resolve jar hell problem.

Each bundle communicates with each other via service layer. This idea is quite identical with SOA approach.

Bundles are deployed on the bundle runtime environment (OSGi framework). Bundles run in the same VM. The framework uses the explicit imports and exports to wire up the bundles so they do not have to concern themselves with class loading.

OSGi technology also widely implemented in the Open Source community, as demonstrated by the Apache Felix and Derby projects, the Eclipse Callisto, Equinox and Corona projects, OSCAR, Knopflerfish, and others.

The real world example of an application which is build on the top of OSGi is Eclipse. Almost everything in Eclipse is a plug-in.

OSGi improves the whole process of developing software and change the way you think about how to build the application. The monolithic application is converted to plug-in base system. So we can easily extend the application when it's required.