组件化和插件化的最大区别（应该也是唯一区别）就是组件化在运行时不具备动态添加和修改组件的功能，但是插件化是可以的。

单独调试（发布）、组件交互、UI跳转、集成调试、代码边界和生命周期等六个方面

组件化：

代码解耦，组件单独运行，组件与组件之间，主项目与组件间的数据传递，UI跳转，组件的生命周期，集成调试，代码隔离

组件的单独调试：

需要把apply plugin: 'com.android.library'切换成apply plugin: 'com.android.application'就可以，但是我们还需要修改一下AndroidManifest文件，因为一个单独调试需要有一个入口的actiivity。

　　我们可以设置一个变量isRunAlone，标记当前是否需要单独调试，根据isRunAlone的取值，使用不同的gradle插件和AndroidManifest文件，甚至可以添加Application等Java文件，以便可以做一下初始化的操作。

　　为了避免不同组件之间资源名重复，在每个组件的build.gradle中增加resourcePrefix "xxx\_"，从而固定每个组件的资源前缀。下面是读书组件的build.gradle的示例：

if(isRunAlone.toBoolean()){

apply plugin: 'com.android.application'

}else{

apply plugin: 'com.android.library'

}

.....

resourcePrefix "readerbook\_"

sourceSets {

main {

if (isRunAlone.toBoolean()) {

manifest.srcFile 'src/main/runalone/AndroidManifest.xml'

java.srcDirs = ['src/main/java','src/main/runalone/java']

res.srcDirs = ['src/main/res','src/main/runalone/res']

} else {

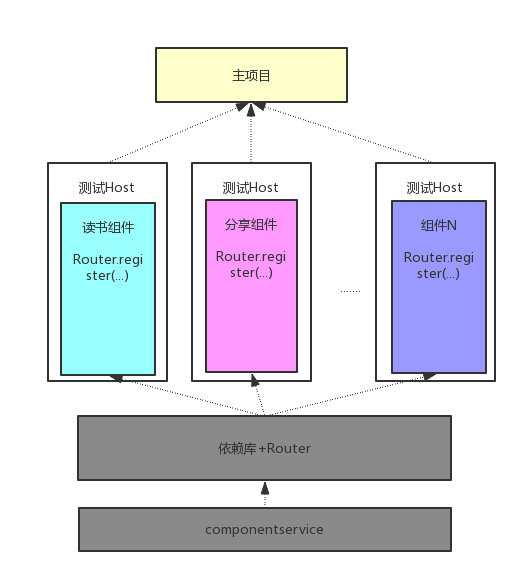
manifest.srcFile 'src/main/AndroidManifest.xml'

}

}

}

组件之间的数据传输



2-4 组件之间的UI跳转