事件驱动架构的基本思想是：系统对外部的行为表现可以通过它对事件的处理来实现。构件不直接调用一个过程，而是触发或广播一个或多个事件。系统中的其它构件中的过程在一个或多个事件中注册，当一个事件被触发，系统自动调用在这个事件中注册的所有过程，这样，一个事件的触发就导致了另一模块中的过程的调用。构件是一些模块，这些模块既可以是一些过程，又可以是一些事件的集合。过程可以用通用的方式调用，也可以在系统事件中注册一些过程，当发生这些事件时，过程被调用。

隐式调用的思想是，不直接调用一个过程，而是发布或广播一个或多个事件。隐式调用为软件重用提供了强大的支持。当需要将一个构件加入现存系统中时，只需将它注册到系统的事件中。为改进系统带来了方便。当用一个构件代替另一个构件时，不会影响到其它构件的接口。