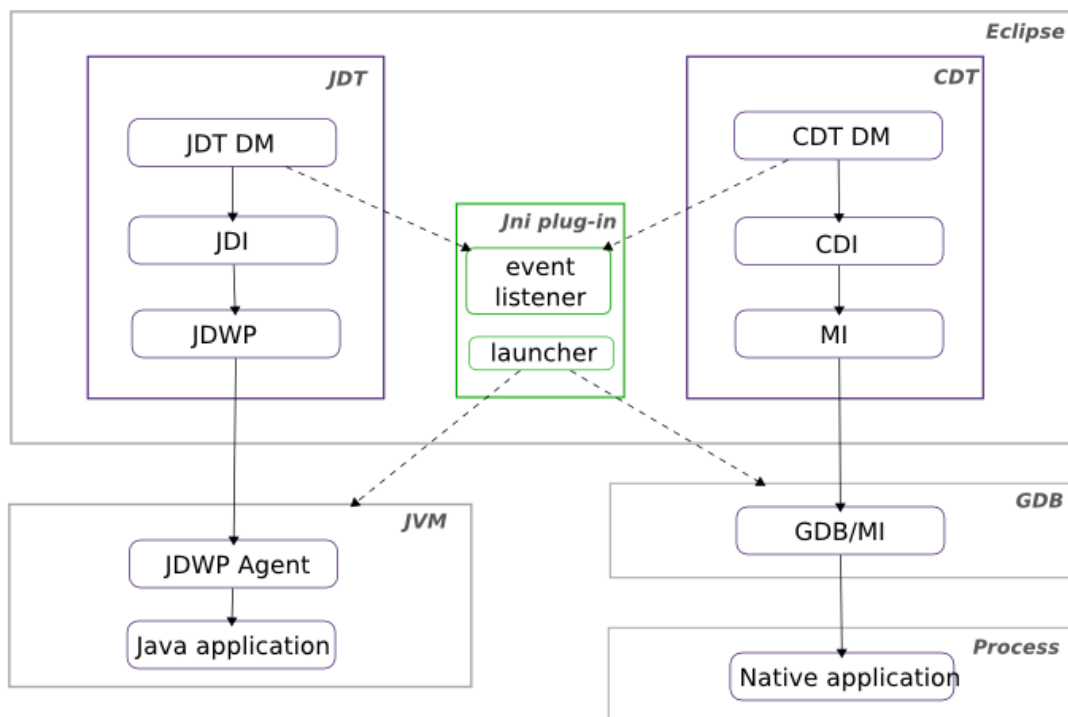


## General infos



- project started for google summer of code targeted Eclipse 3.3
- project based on JDT and CDT
- project goal : step back and forth between JNI/Java code
- main issue : no architecture in Eclipse for mixed-mode debugging

## Technical infos : followed approach



- using a custom launcher
- handling step actions with “invisible” breakpoints (like gdb)
- listening to debug events to

# Custom debug launcher



## launcher steps

- set breakpoints on native functions in C/C++ code
- launch the jvm
- attach C/C++ debugger to the launched jvm process
- resume the jvm
- restart the debug event listener

# Debug event listener



## resume events

- handle step into, step over and step return events
- save for each thread kind (C and Java) the last event kind

## suspend events

- handle step end and breakpoint events
- resume, step return or do nothing  
(depending on the last resume event)

## Limitations & other approach



- can not see both targets at each time
- there are some threads problems
- the good works depends on the JVM JNI implementation
  
- Intel's proposal : new debug interface for mixed-mode debugging
- generic debug model with multi-language support
- need a JVM which supports the new interface
- prototype for Eclipse 3.2 with patched Apache Harmony JVM