



IcedTea Java Plugin

Deepak Bhole

Senior Software Engineer

Overview

■ IcedTeaPlugin

- History
- Design
- Future
- Demos



Plugin History

- **Project started by Thomas Fitzsimmons in September 2007**
- **Prototype working code in by February 2008**
- **Taken over by Deepak Bhole in August 2008, full time work on plugin started**
- **First stable release was publicly available on October 27th 2008 in IcedTea 1.3.1**
- **First 64-bit plugin**



Plugin Design

- **Overview**
- **Performance**
- **Security**
- **Other notes**

Design - Overview

- **Design Goals**
- **Features and stats**
- **Code components**
- **Full compatibility with GCJ Web Plugin**
- **Provide full liveconnect and signed applet support**
- **Basically, have any site with applets work “out of the box” with Fedora**

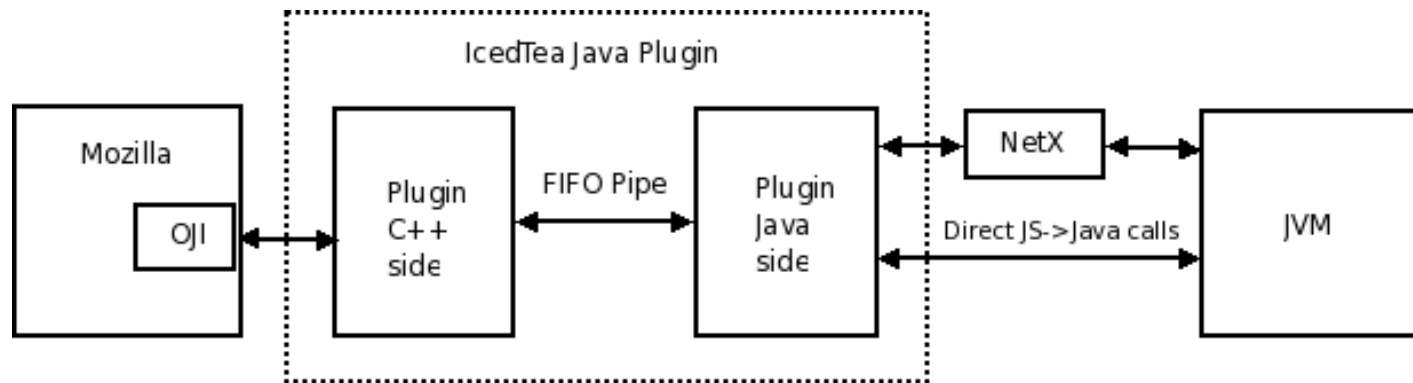
Design - Overview

- Design Goals
 - **Features and stats**
 - Code components
- ~5500 lines of C++ code
 - ~5700 lines of Java code
 - Single JVM running at any given time
 - Applet drawing and security delegated to NetX

Design - Overview

- Design Goals
- Features and stats
- Code components
- Split into “C++ side” and “Java side”
- C++ side handles all communication with the browser
- Java side handles all communication with the JVM instance

Code components



OJI interface into Mozilla

C++ side talks to OJI on one side, Java on the other

C++/Java communication over a FIFO pipe

Java side talks to NetX and the JVM

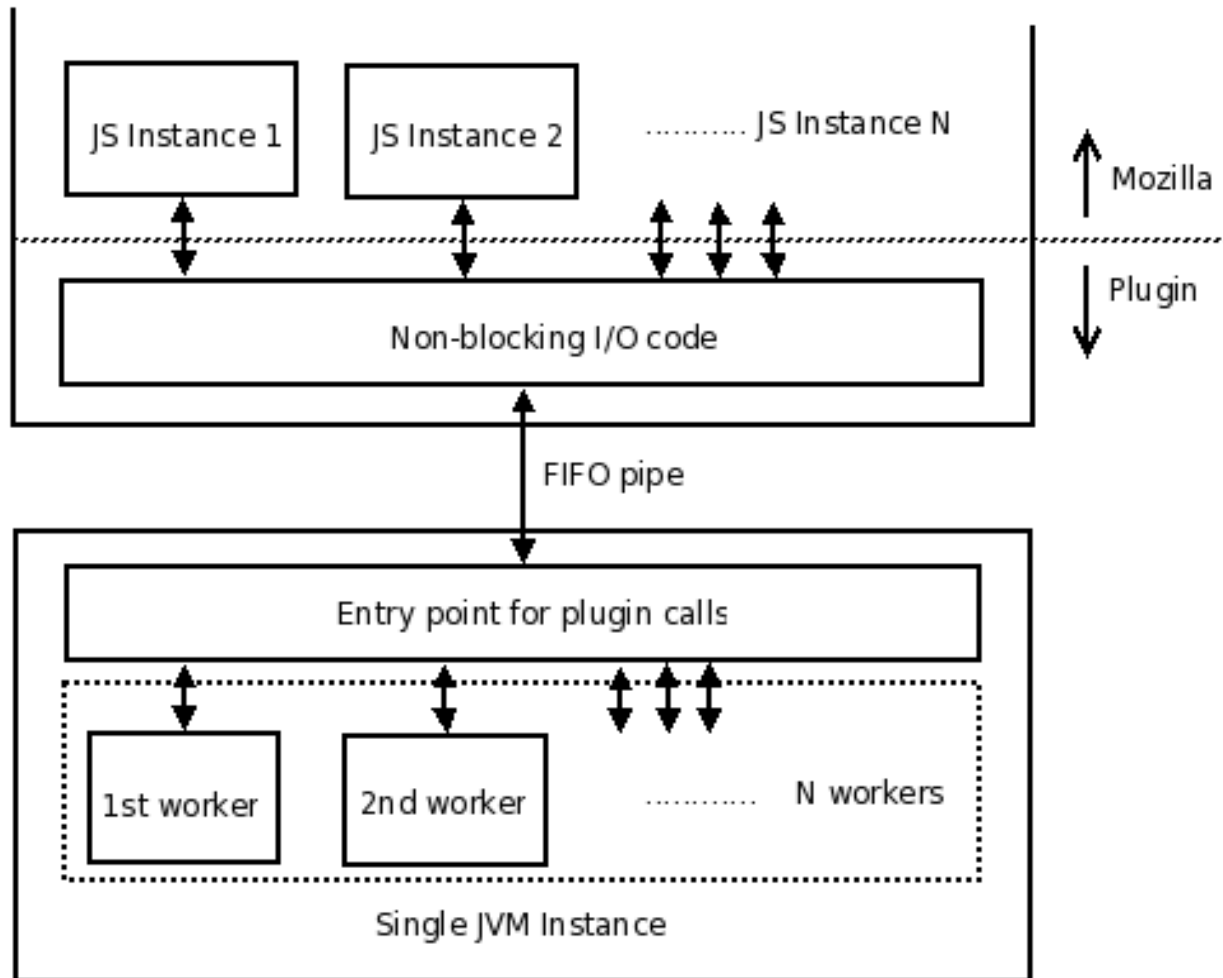
Design - Performance

- **Overview**
- Performance oriented design
- Limitations
- **On par with the Sun plugin**
- **Single JVM means subsequent applet loads are fast**
- **Multiple applet operations can continue in parallel despite single JVM instance**

Design - Performance

- Overview
- Performance oriented design
- Limitations
- Each side of the plugin (C++/Java) is multi-threaded
- C++ side threading is per JS instance, handled by Mozilla for the most part
- Java side threads are generic worker threads that can be scaled as needed

Design - Performance



Design - Performance

- Overview
- Performance oriented design
- Limitations
- FIFO pipe means that events need to be polled
- Great deal of initial table building is a significant overhead for that first “click”

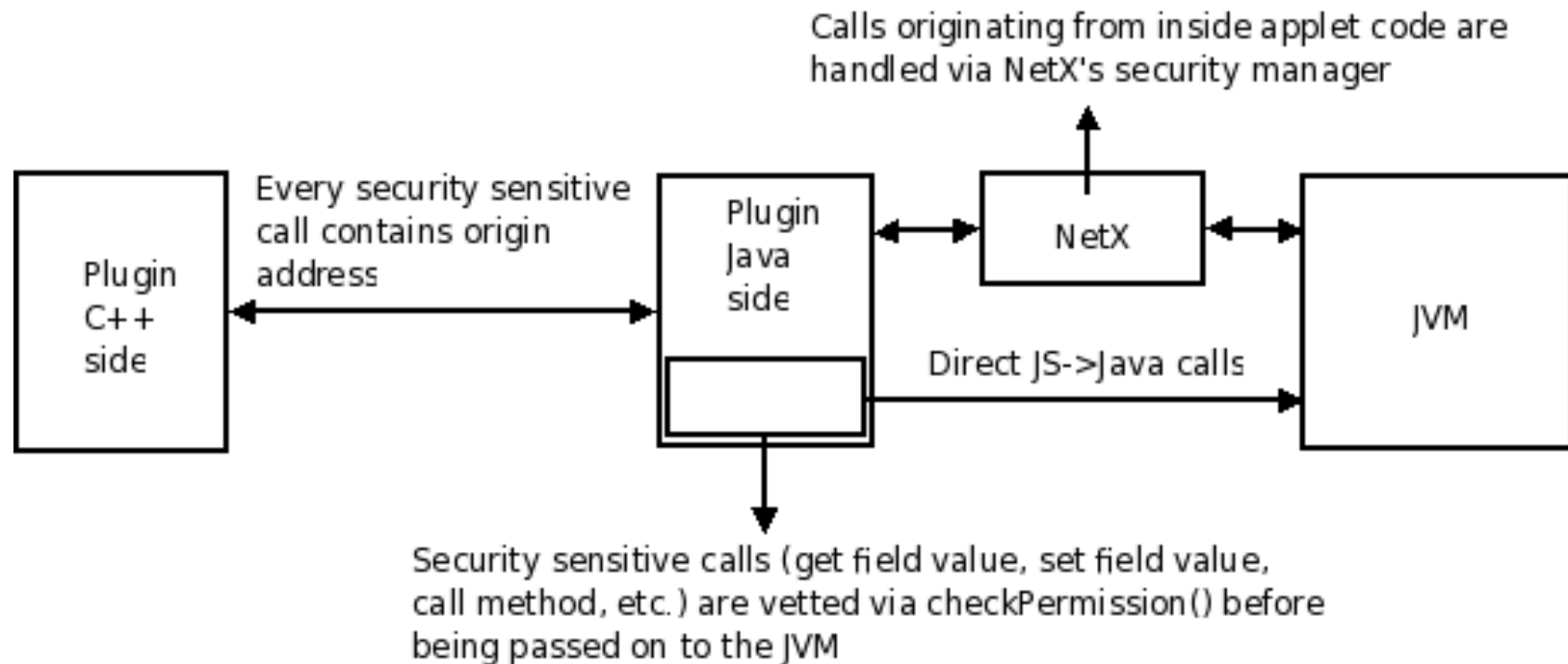
Design – Security

- **Concerns**
 - Security oriented design
 - NetX security
 - Plugin Security
 - Limitations
- **Two main vectors of attack**
 - Applet
 - Java script -> Java
 - **Single JVM instance opens possibilities of DoS**

Design – Security

- Concerns
 - **Security oriented design**
 - NetX security
 - Plugin Security
 - Limitations
- **All security sensitive external code runs with a SecurityManager**
 - **Applet security is delegated to NetX**
 - **Liveconnect JS->Java call security is handled by the plugin**
 - **Exceptions cannot bring down the VM**

Design – Security



Design – Security

- Concerns
- Security oriented design
- **NetX security**
- Plugin Security
- Limitations
- **Custom security manager (JNLPSecurityManager)**
- **JNLPSecurityManager not only sandboxes applets, but also handles cases where user prompt is more appropriate than denial**
- **User is prompted for socket connect permissions with untrusted applets**

Design – Security

- Concerns
 - Security oriented design
 - NetX security
 - **Plugin Security**
 - Limitations
- **All security sensitive calls (GetField, SetField, CallMethod, etc.) are run through a security manager**
 - **Execution of such calls is within a dynamically generated AccessControlContext**
 - **Both, UniversalBrowserRead and UniversalJavaPermission permissions are honoured**

Design – Security

- Concerns
 - Security oriented design
 - NetX security
 - Plugin Security
 - **Limitations**
- **Attempts to tie up the JVM cannot be prevented ... but this will be true for any plugin**

Design – Other notes

- **Out of process JVM means that a VM crash cannot bring down a browser**
- **JVM instance is monitored, and should it go down for some reason, another VM is brought up upon request**
- **Objects on the VM side are automatically unreferenced when Mozilla is done with them, allowing the GC to clear up the memory**
- **Debugging is very easy – single variable (ICEDTEAPLUGIN_DEBUG) controls debug output, and allows the Java side code to be remotely debugged**



Future

- **Create a Testsuite**
- **Memory auditing**
- **Performance improvements**
- **FX support**
- **Win32 testing**

Demos

- **DoS prevention**
- **Security demo**
- **Liveconnect in action**
 - JS->Java
 - Java->JS
- **External sites (JMol, Yahoo!, Facebook)**



Thanks

- **Thomas Fitzsimmons**
- **Sun**
- **All IcedTea contributors**
- **FOSDEM organizers**



Q/A

Questions?

Deepak Bhole
dbhole@redhat.com