A concept for and an implementation of the Bean Scripting Framework for Rexx

by Peter Kalender

What is Rexx?

- Scripting language

- Very readable

- Designed as macro language

=> Easy to add new functions
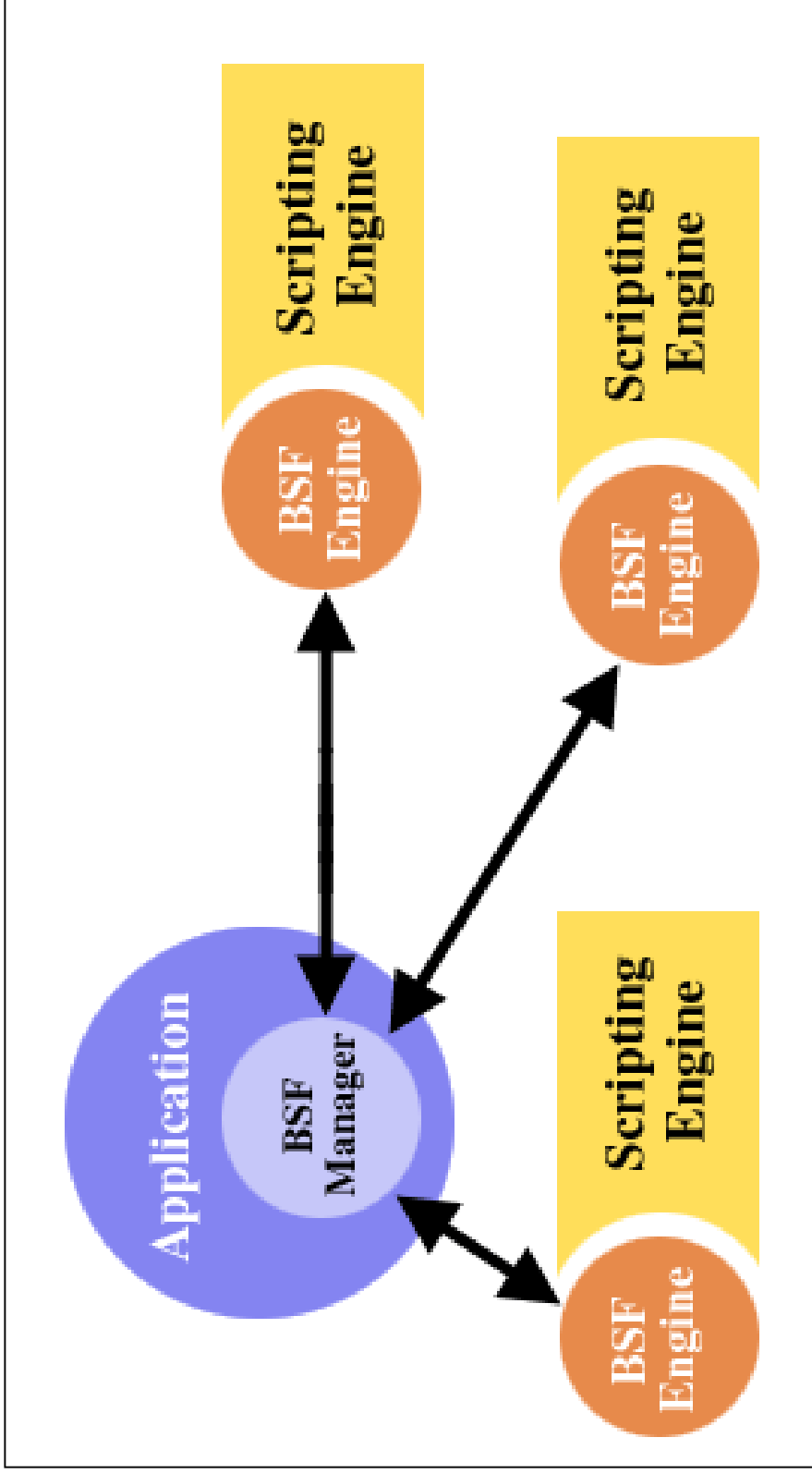
=> Easy access from other languages

What are Java Beans?

- Components

- Expose features to tools

- Used for visual programming

- Communicate through events

What is the Bean Scripting Framework (BSF)?

- Use scripts in Java

- Control Java Objects with scripts

Scripting Engine

Scripting Engine

BSF Engine

BSF Engine

Application
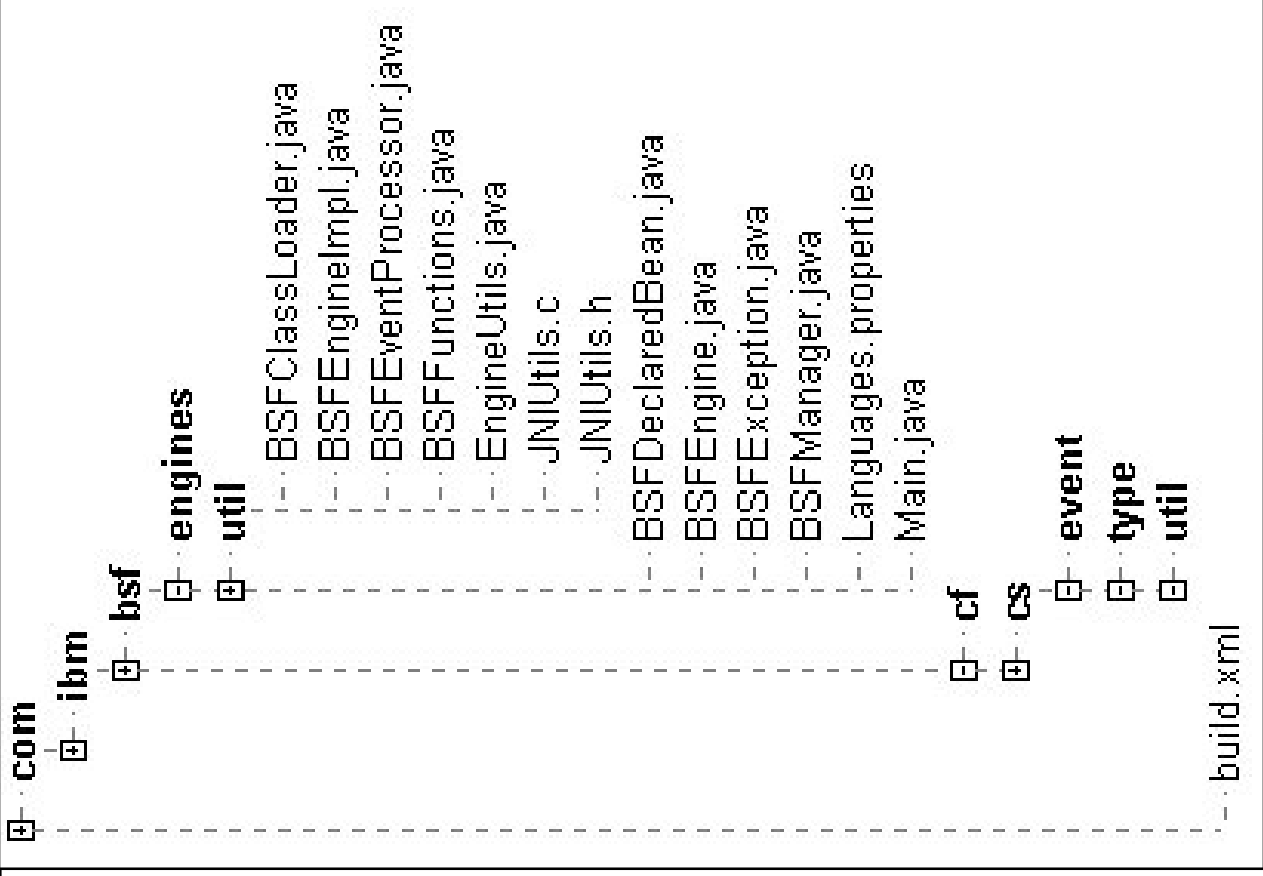
BSF Manager

Scripting Engine

BSF Engine

The BSF architecture

Scripting languages supported by BSF 2.1

- BML

- Jacl

- JavaScript

- Jpython

- NetRexx

- TCL

- XSL

- On Win32: VBScript, Jscript and PerlScript

```
com
  ibm
    bsf
      engines
      util
            BSFClassLoader.java
            BSFEngineImpl.java
            BSFEventProcessor.java
            BSFFunctions.java
            EngineUtils.java
            JNIUtils.c
            JNIUtils.h
            BSFDeclaredBean.java
            BSFEngine.java
            BSFException.java
            BSFManager.java
            Languages.properties
            Main.java
    cf
    cs
      event
      type
      util
build.xml
```

The main BSF files

The BSFManager

- Central class of the BSF

- Administers scripting engines
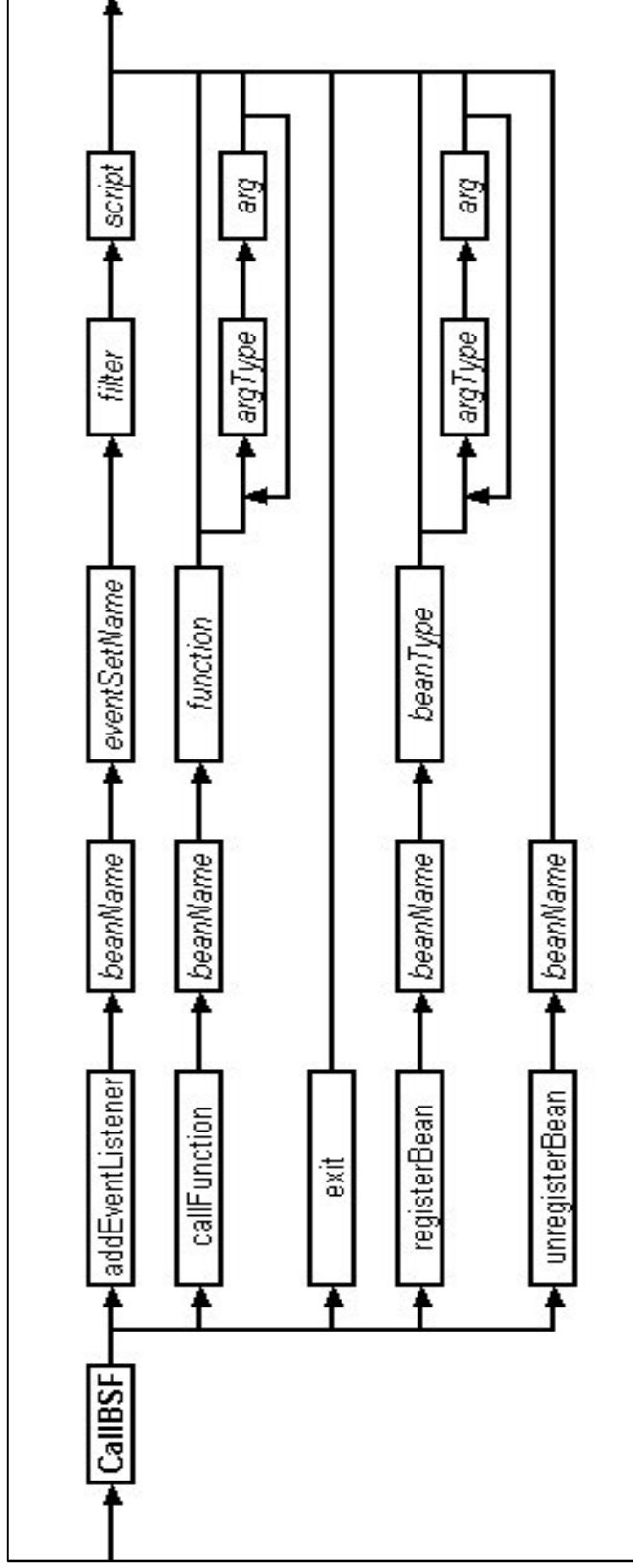
- Administers beans

- Handles scripts

The **BSFEngine**

- Interface for scripting engines

- declare & undeclare Beans

- Run scripts

- Basic implementation => BSFEngineImpl

Beans in Rexx?

- No objects in Rexx

=> Beans stored in BSFManager

=> Function to manipulate beans

CallBSF

addEventListener — beanName — eventSetName — filter — script

callFunction — beanName — function — argType — arg

exit

registerBean — beanName — beanType — argType — arg

unregisterBean — beanName

The CallBSF function

11

Connecting Java and Rexx
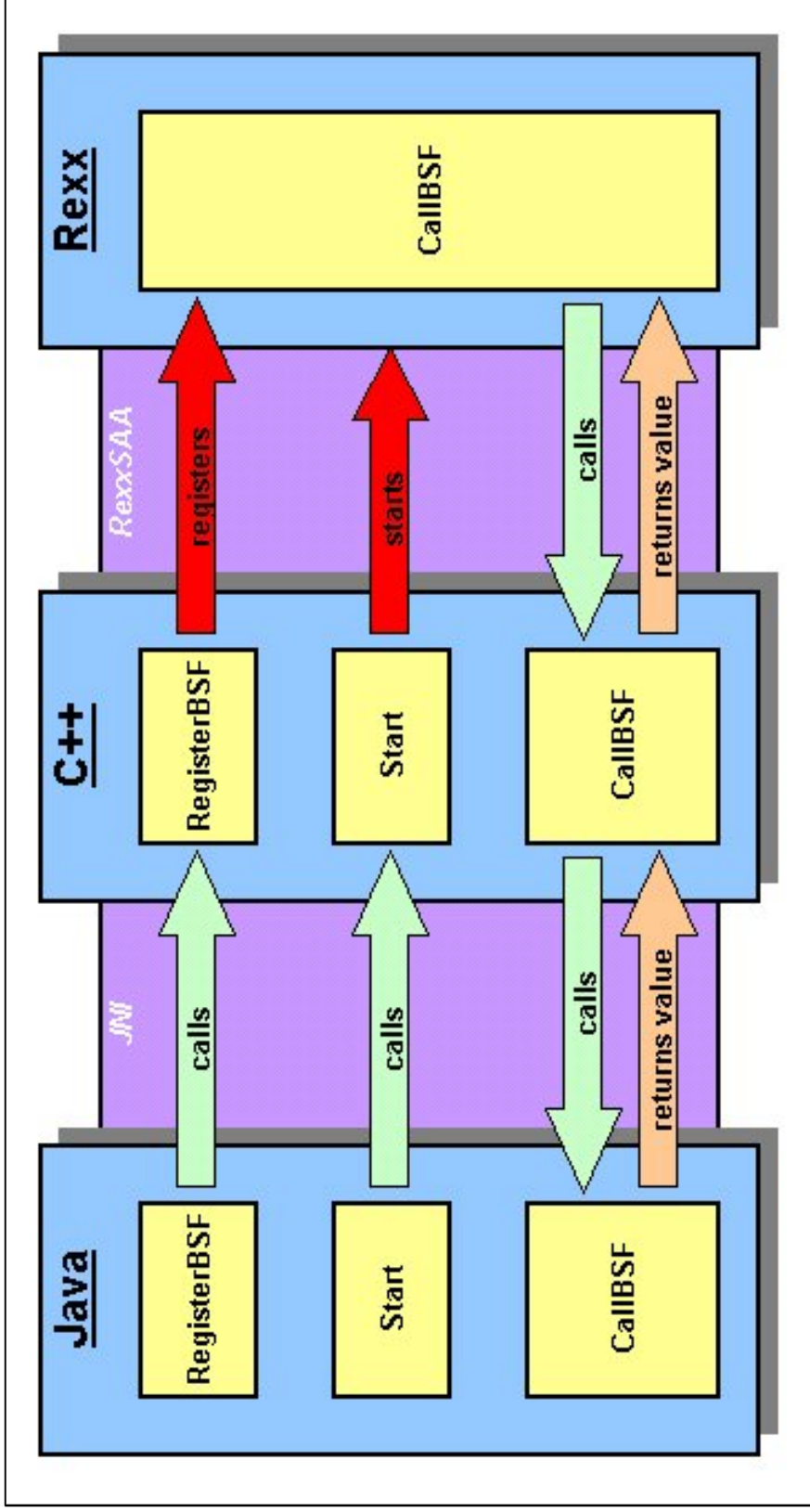
- Rexx SAA API uses C header

=> JNI

The Rexx SAA API

- Subcommand handlers

- External function handlers

- Interpreting

- Variable interface

- System exits

The Java Native Interface (JNI)

- Native methods
- Callbacks
- Type mapping
- Invoking the Java Virtual Machine

Connecting Java and Rexx

```
JNIEXPORT jlong JNICALL
Java_com_ibm_bsf_engines_rexx_JavaRexxSAA_RegisterBSF(JNIEnv *env,
    jobject obj)
{
LONG ReturnValue;
ReturnValue = RexxRegisterFunctionDll("CallBSF",
    "JaReSAA.dll",
    "CallBSF");
    return ReturnValue;
}
```

```
public Object eval (String source, int lineNo, int columnNo,
                    Object oscript) throws BSFException {

    String script = oscript.toString ();

    if(source.equals("<event-script>")){

        script = script+"\nReturn;\n"+mainScript;

    } else {

        mainScript = script;

    }

    Rexx.Start(script);

    return null;

}
```

Bibliography

Rexx

http://www.RexxLA.org

http://wwwi.wu–wien.ac.at/Studium/LVA–Unterlagen/poolv/folien/

Rexx history

http://www2.hursley.ibm.com/rexx/rexxhist.htm

http://www2.hursley.ibm.com/rexx/faq4.htm

Regina Rexx interpreter

http://www.lightlink.com/hessling/Regina/

Bibliography

BSF

http://oss.software.ibm.com/developerworks/opensource/bsf/

http://www.javaworld.com/javaworld/jw–03–2000/jw–03–beans.html

Java

http://java.sun.com/

JNI

http://java.sun.com/docs/books/tutorial/

http://developer.java.sun.com/developer/onlineTraining/Programming/JDCBook/index.html