Internals of the Adobe ColdFusion Runtime

Elliott Sprehn
elliott@teratech.com
http://enfinitystudios.thaposse.net/blog/

What *is* ColdFusion?

- A set of languages (CFML,CFScript)
- A compiler for the languages
  - Generates Java Bytecode
- A runtime environment
  - An interpreter for expressions

The Basic Process

- Parse and compile cfml/cfscript on first request
  - Java Classes (interesting packages):
    - coldfusion.runtime.*
    - coldfusion.tagext.*
- Cache resulting bytecode for next request
- Execute.

Variables

- coldfusion.runtime.Variable
  - Generic container, holds any value.
  - Not generally useful inside CF code.
  - Every named variable has an instance.
    - Automatically unwrapped by the runtime.

Code Sample:

```
<cfset myVar =
createObject(“java”,”coldfusion.runtime.Variable”).in
it(“myVar”)>
<cfif not myVar.isDefined()>
  ...
</cfif>
```

Types

- Types map to runtime classes
  - **Query:** coldfusion.runtime.QueryVector
  - **Struct:** java.util.Map: coldfusion.runtime.Struct
  - **Array:** java.util.List: coldfusion.runtime.Array
  - **List:** java.lang.String
  - **String:** java.lang.String
  - **Number:** java.lang.String, int, double
  - **Date:** java.util.Date
  - **Component:** coldfusion.runtime.TemplateProxy (more about this later)
  - **Java Object:** coldfusion.runtime.java.JavaProxy
- Casting Handled by coldfusion.runtime.Cast

Scopes

- Implemented by various classes
  - **Variables:** coldfusion.runtime.VariableScope
  - **Arguments:** coldfusion.runtime.ArgumentsCollection
  - **Client:** coldfusion.runtime.ClientScope
  - **Server:** coldfusion.runtime.ServerScope
  - **Application:** coldfusion.runtime.ApplicationScope
  - **Session:** coldfusion.runtime.MemorySessionScope
  - **Super:** coldfusion.runtime.SuperScope
  - **Local:** coldfusion.runtime.LocalScope
- Some scopes have extra behavior not normally exposed because of member access rules.

Code Sample:

```
application.getEventInvoker().getThisScope()
```

Functions

- Implemented by coldfusion.runtime.UDFMethod
- Compiled to static inner classes on pages.
- Requires
  - PageContext
  - LocalScope
  - ArgumentCollection
- LocalScope variables **are** actually stack local.
  - Still stored in a LocalScope collection.
  - Faster to access because they’re local.
  - Not discarded when function finished in conventional Java manner.

Java Objects

- Proxied by coldfusion.runtime.java.JavaProxy
  - Adds lots of overhead to invocation.
  - Uses Java reflection
    - **very** expensive
  - CF does some method caching to reduce the performance hit.
- **Real pain to invoke some method signatures.**
  - Use Java reflection yourself, avoid CF method lookup.

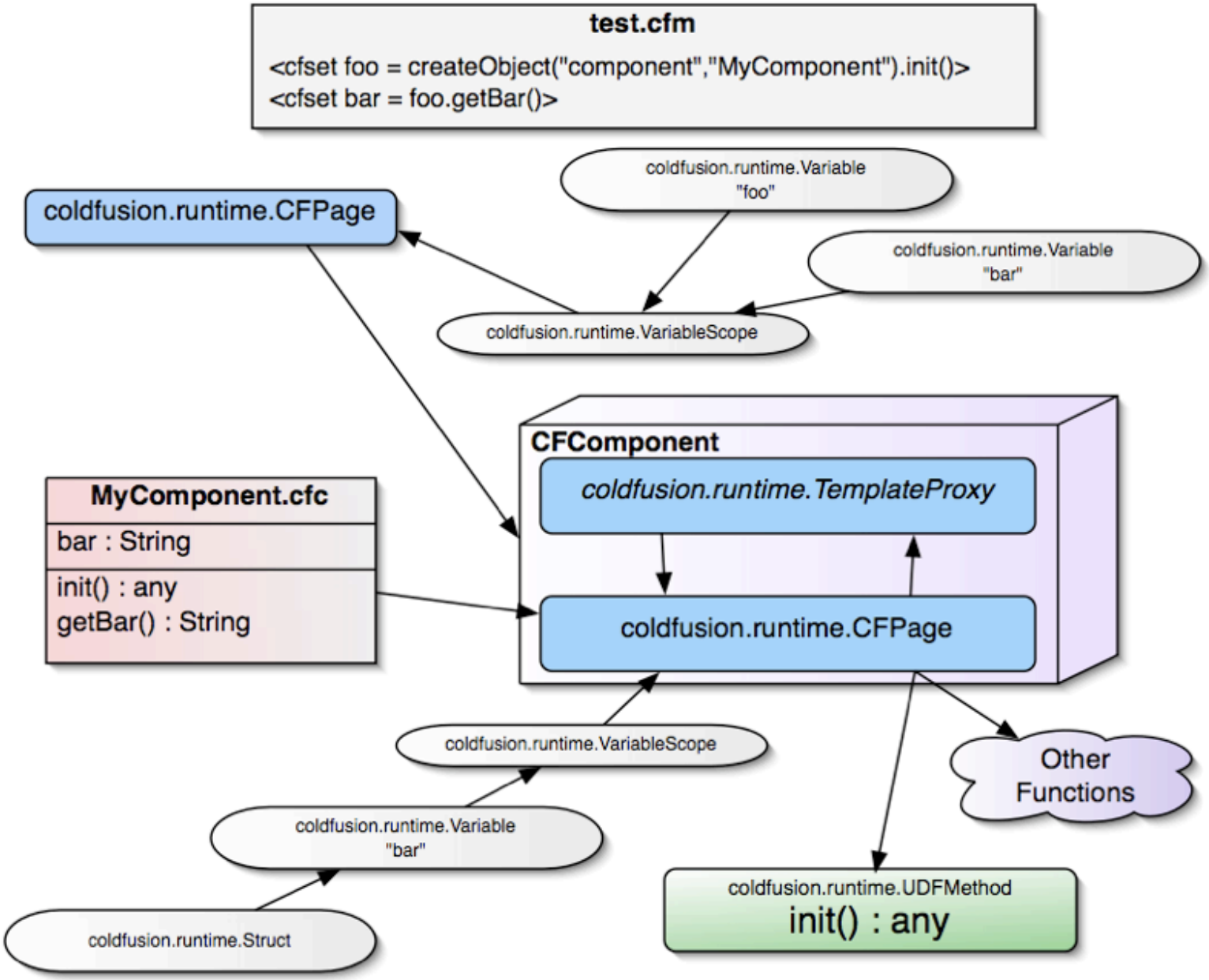
Tags

- Implemented as JSP Tags
  - Some are CF Code (ex. cfsavecontent)
- Map to a coldfusion.tagext.* class.
  - coldfusion.tagext.sql.QueryParamTag
  - coldfusion.tagext.lang.ThrowTag
  - coldfusion.tagext.io.DirectoryTag
- Tags implement javax.servlet.jsp.tagext.Tag
  - Execution
    1. setXXXX(value) (attributes)
    2. doStartTag()
    3. doEndTag()
    4. release()

Pages, Includes, Custom Tags, Components

- Everything is a coldfusion.runtime.CFPage
  - No real difference between a custom tag and a regular page.
  - Components are just wrapped up pages.
  - Everyone has their own PageContext.
  - Page is exposed though getPageContext().getPage()
    - Contains all your custom functions, variables, etc.
    - Not real useful without special code.
- Component creation is expensive.
  - Lots of work to create a Page.
  - Requires separate output buffer, PageContext, scopes, ...
  - Inheritance!

Pages, Includes, Custom Tags, Components (cont.)



Discovering how Adobe ColdFusion works...

- getMetaData(variable)
  - Returns the java.lang.Class of non-components
- Exceptions + Stack Traces
  - Expose the inner workings and classes for inspection.
  - Try passing null or non-CF type to create one.
- Google Search!
  - “coldfusion.runtime”
  - “coldfusion.tagext”
  - ...

Code Samples / Implementation Details

- Create and execute queries that use cfqueryparam from cfscript.
- Create a function like CF8 to get database info for a datasource.
- Add global mappings by appending to a structure like this.mappings in CF8.
- Call functions and component methods and inspect the local scope after their execution.
- Access the Application.cfc anywhere in an application.
- Allow passing arrays of values in the url scope instead of lists when there are duplicate keys.

Questions