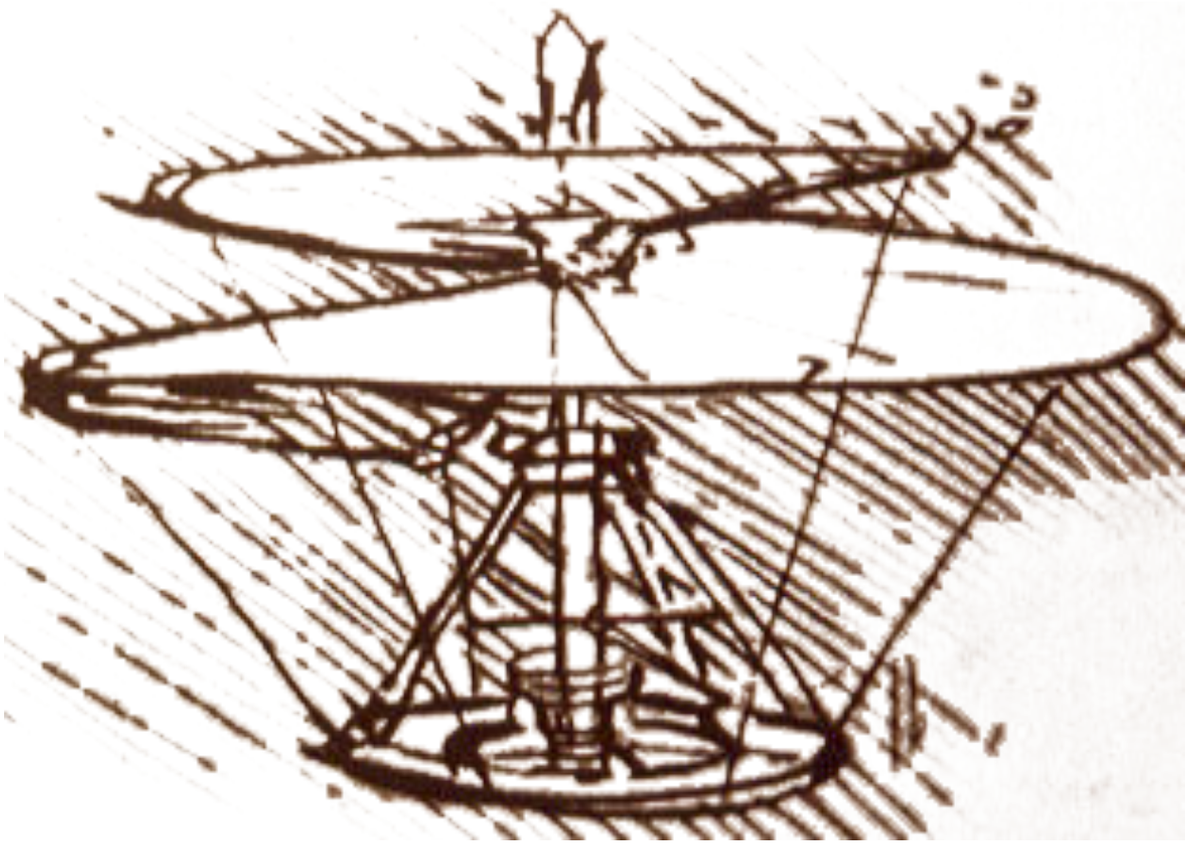
ThoughtWorks®

Adopting the JVM



Ola Bini

computational metalinguist

ola.bini@gmail.com

http://olabini.com/blog

Your host

From Sweden to Chicago through ThoughtWorks

Language geek at ThoughtWorks

JRuby core developer, Ioke and Seph creator

Member of the JSR 292 EG

Demographics

Smorgasbord

Whys and wherefores

How to get started

Polyglot patterns

Interoperability

JVM tweaks and tools

Bytecode

JSR 292

Politics

Why the JVM?

- Fantastic platform
  - JIT
  - Memory model and GC
  - Threading
  - Libraries
  - Tools
- Runs mostly anywhere
- Interoperability between languages
- Many interesting languages already available

Why not Java?

- More ceremony than essence
  - Verbosity
- Slow turnaround
  - Compile time cycle
- Abstraction level
- Expressivity
- Rate of change
- Useful libraries in other languages

Sapir-Whorf

onsdag, 2010 november 03

Programming languages

Iverson - “Notation as a tool of thought”

Paul Graham - The Blub paradox

Ruby - Matz says one inspiration was novel Babel-17

Steve Yegge - The difference between recursion and iteration

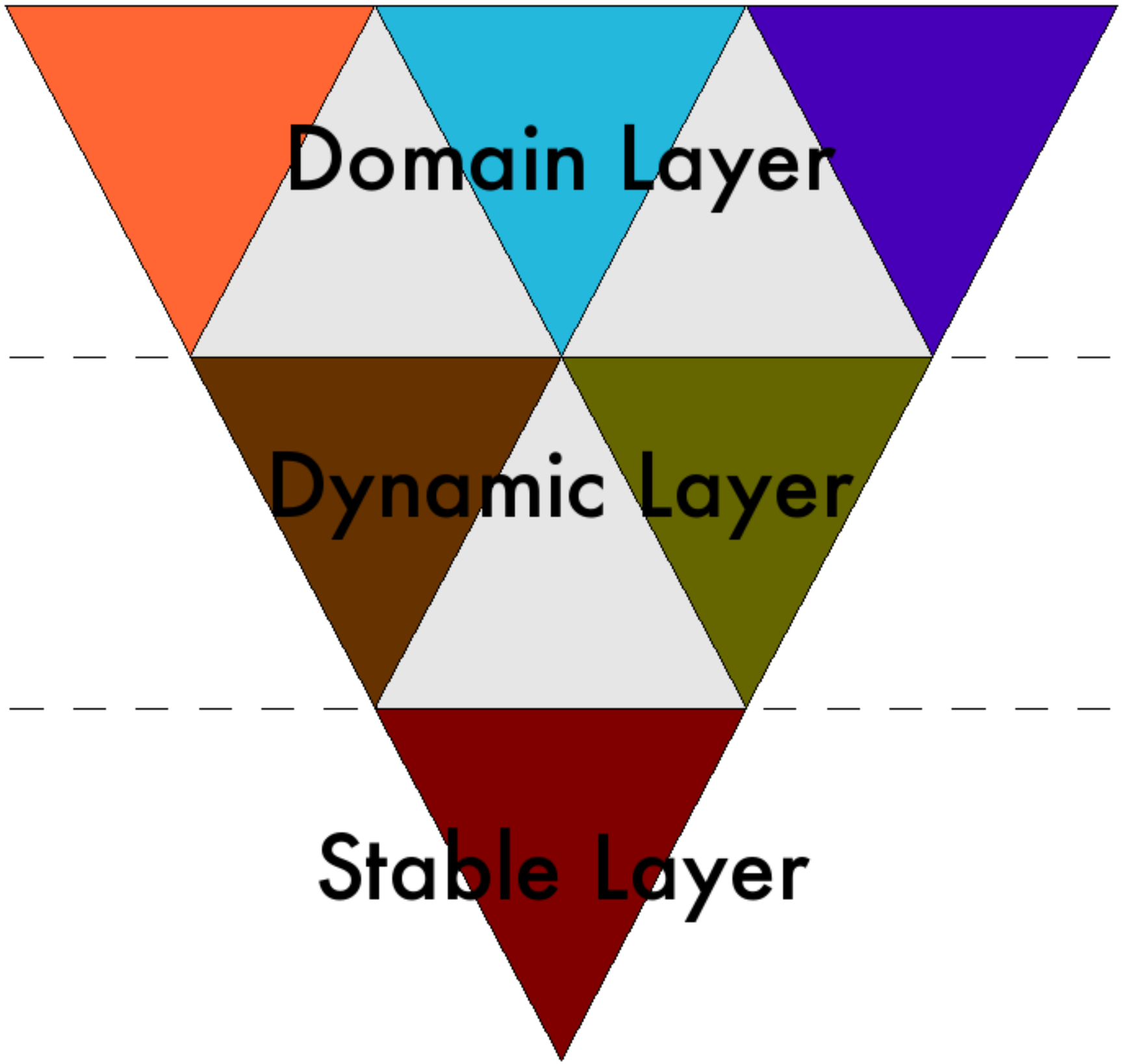
Where do you start?

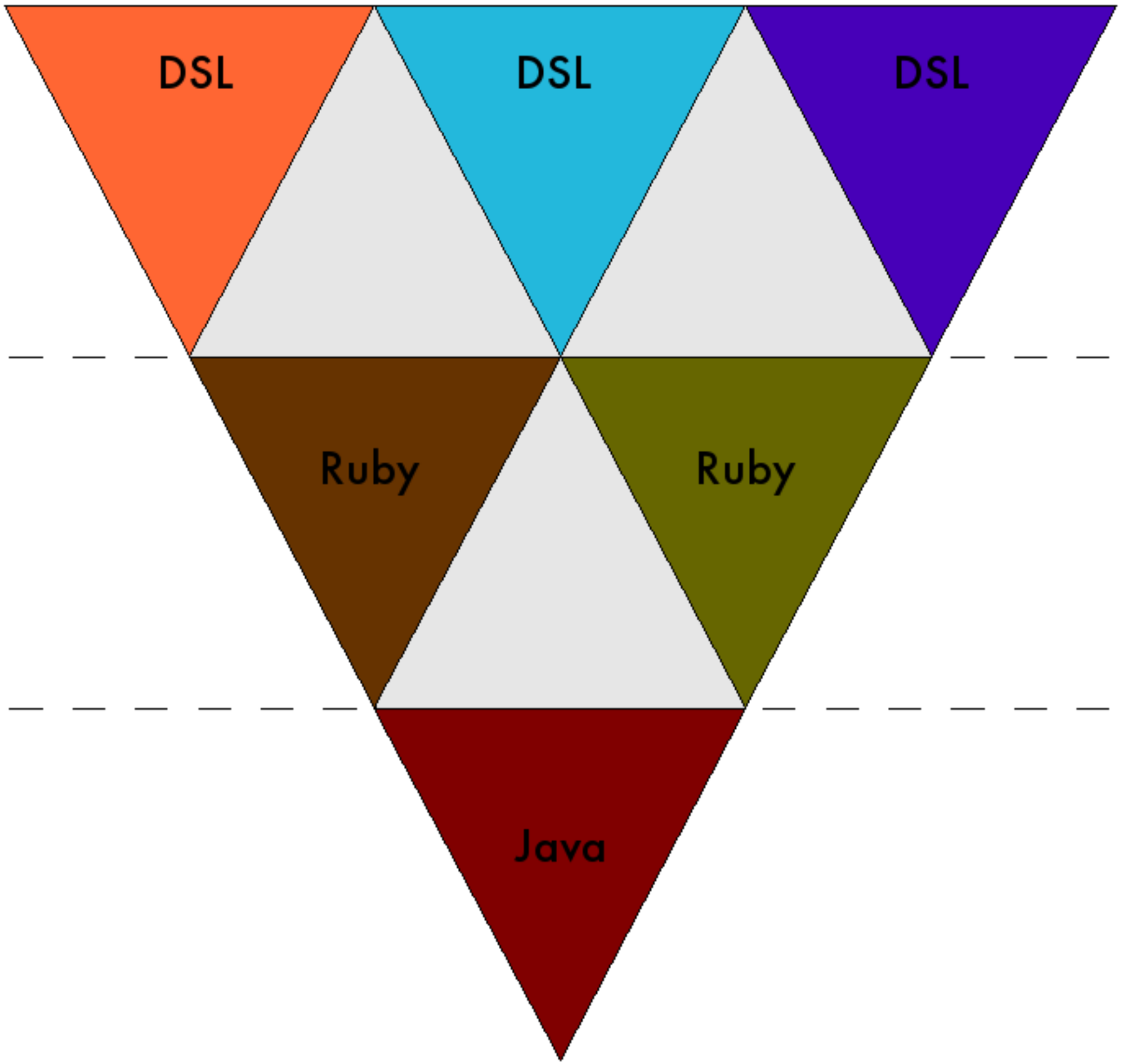
Adopting a new language

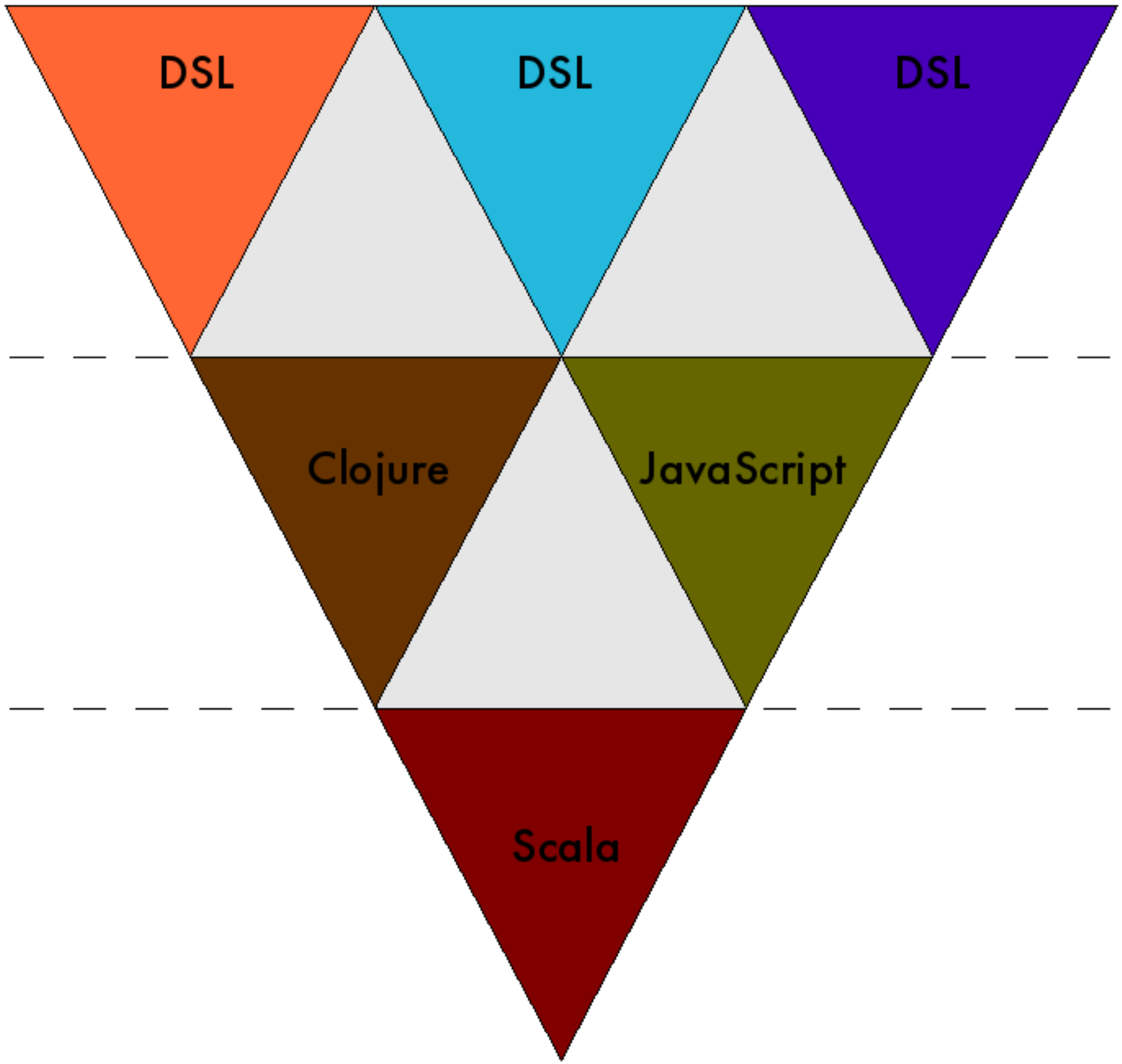
- Choose with care
  - Be strategic about language use
- Productivity bump
  - New syntax and new libraries
  - But also new ways of thinking
- Leverage should increase quickly
- Not that different from learning a new framework

Polyglot Patterns

Architecture


Domain Layer
Dynamic Layer
Stable Layer


DSL
DSL
DSL
Ruby
Ruby
Java


DSL
DSL
DSL
Clojure
JavaScript
Scala

Polyglot Testing

- Scala, Clojure, Ruby, Groovy, etc
  - All of these can test Java code very well
- Quicker turnaround
- More manageable tests
- Use libraries from test language (RSpec, Erjang QuickCheck)
- Approaches
  - Exploratory testing - REPL
  - Unit and functional testing
  - Acceptance testing - Cucumber

```
import org.thoughtworks.PrimeFinder

describe PrimeFinder do
  it "finds the next prime number" do
    pf = PrimeFinder.from(10)
    pf.next.should == 11

    pf = PrimeFinder.from(25)
    pf.next.should == 29
  end
end
```

```
import(org:thoughtworks:PrimeFinder)

forAll(natural n,
  pf = PrimeFinder from(n)
  pf next should be odd
)
```

Build Scripting

Is Maven or Ant the best way to build your system?

BuildR

GAnt

Polyglot Maven

General build scripts

```
includeTargets << gant.targets.Clean
cleanPattern << [ '**/*~' ,  '**/*.bak' ]
cleanDirectory << 'build'

target ( stuff : 'A target to do some stuff.' ) {
  println ( 'Stuff' )
  depends ( clean )
  echo ( message : 'A default message from Ant.' )
  otherStuff ( )
}

target ( otherStuff : 'A target to do some other stuff' ) {
  println ( 'OtherStuff' )
  echo ( message : 'Another message from Ant.' )
  clean ( )
}

setDefaultTarget ( stuff )
```

Alien Libraries

- X library consumed by Java
  - Depends on how the language exposes Java functionality
- Java library consumed by X
  - Using Xs Java Integration features

Sometimes complicates build process

```
(.. System (getProperties) (get "os.name"))
```

Service Injection

- Using DI to compose languages
- Spring has ScriptFactory
  - Allows implementation of any interface in most languages

```java
package com.thoughtworks;

public interface Animal {
    void run();
}
```

```ruby
class Fox
  def run
    puts "Running away!"
  end
end

Fox.new
```

```
<?xml version="1.0" encoding="utf-8"?>

<beans>
  <bean
class="org.springframework.scripting.support.ScriptFactoryPostProcessor" />

  <bean id="fox"
        class="org.springframework.scripting.jruby.JRubyScriptFactory">
    <constructor-arg value="file:fox.rb" />
    <constructor-arg value="com.thoughtworks.Animal" />
  </bean>
</beans>
```

End User Scripting

Make it possible to customize behaviour

Save/Load scripts

Expose a model to the scripts

How much of the application can be implemented like this?

Versioning

Java as the C of the JVM

- Use another language as the main language
- Drop down to Java
  - Performance
  - Features that top level doesn't have
  - To bind things together

MOPping

BSF

```
JLabel mylabel = new JLabel();
BSFManager.registerScriptingEngine("ruby",
                                   "org.jruby.javasupport.bsf.JRubyEngine",
                                   new String[] { "rb" });

BSFManager manager = new BSFManager();
manager.declareBean("label", mylabel, JFrame.class);
manager.exec("ruby", "(java)", 1, 1, "$label.setText(\"This is a test.\")");
```

JSR 223

```
ScriptEngineManager m = new ScriptEngineManager();
ScriptEngine rubyEngine = m.getEngineByName("jruby");
ScriptContext context = rubyEngine.getContext();

context.setAttribute("label", new Integer(4),
                     ScriptContext.ENGINE_SCOPE);

try {
    rubyEngine.eval("puts 2 + $label", context);
} catch(ScriptException e) {
    e.printStackTrace();
}
```

Dynamic MOP

JVM Tweaking

The basic flags

-server, -client

-Xmx, -Xms, -Xss

-Xverify:none

-XX:PermSize=512m

-XX:MaxPermSize=512m

What's happening with GC?

-XX:+PrintGC

-XX:+PrintGCTimeStamps

-XX:+PrintGCDetails

What's happening with JITting?

-XX:+PrintCompilation

-XX:+PrintInlining (debug build only)

-XX:+PrintOptoAssembly (debug build only)

-XX:+PrintAssembly (debug build only)

-XX:+UnlockDiagnosticVMOptions

-XX:+PrintNMethods

-XX:+PrintSignatureHandlers

Other diagnostics

-XX:+HeapDumpOnOutOfMemoryError

-XX:+TraceClassLoading

-XX:+TraceClassUnloading

-XX:+TraceClassLoadingPreorder

-XX:+TraceClassResolution

Tuning

-XX:+AggressiveOpts

-XX:+UseCompressedOops (for x64 versions)

-XX:+DoEscapeAnalysis

-XX:CompileThreshold=n (default 10000)

-XX:NewRatio=5

-XX:ParallelGCThreads=2

More information here:

http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html

JVM Tools

JMX

JConsole

VisualVM

Eclipse Memory Analyzer

JVM Bytecode

ASM

BiteScript

JSR 292

Method handles

- A pointer to a Java method
- As fast to invoke as calling the method directly
- No introspection available - it's an opaque stateless object
- Combinator based library around them
- Capability based
  - Security check at lookup, not at invocation
  - If you have the object you can call it

Method handle combinators

MH catchException(MH target, Class exType, MH handler)

MH collectArguments(MH target, MethodType newType)

MH convertArguments(MH target, MethodType newType)

MH dropArguments(MH target, int pos, Class... valueTypes)

MH filterArguments(MH target, MH... filters)

MH foldArguments(MH target, MH combiner)

MH guardWithTest(MH test, MH target, MH fallback)

MH insertArguments(MH target, int pos, Object... values)

MH spreadArguments(MH target, MethodType newType)

```
MethodType type = MethodType.methodType(void.class, String.class);
MethodHandle handle =
    MethodHandles.lookup().findVirtual(PrintStream.class, "print", type);
handle.invokeExact(System.out, "Hello method handles\n");
```

Neat trick

```
public static void sayClassName() {
    String name = MethodHandles.lookup().lookupClass().getName();
    System.err.println(name);
}
```

Invoke Dynamic

New byte code

For each call site - calls a bootstrap method

Bootstrap method defined per class (in general)

Returns a CallSite or MethodHandle that should be used

Usually a guardWithTest MethodHandle is expected

CallSites can be invalidated according to language specific conditions

InvokeDynamic interface that can be used to call any method with

```
@BootstrapMethod(value=DynTest.class, name="bootstrap")
public final class DynTest {
    private static CallSite bootstrap(Class<?> theClass,
                                      String theName, MethodType theType) {
        MethodHandle mh = MethodHandles.lookup().
            findStatic(DynTest.class, "generic", theType);
        CallSite     cs = new CallSite(mh);

        cs.setTarget(mh);
        return cs;
    }

    public static String generic(String theArgument) {
        return theArgument + ": brought to you by DynTest.generic()";
    }

    public static void main(String[] theArgs) throws Exception {
        System.out.println((String)InvokeDynamic.bar("bar invoked dynamically"));
    }
}
```

Java Politics

The JCP and JSRs

Apache Harmony

Java 7 delayed

Oracle sues Google

IBM joins OpenJDK

Doug Lea

Apple JDK deprecated

Java 7

Forking?

Oracle as Java steward

Questions?

Ola Bini

ThoughtWorks®

http://olabini.com
obini@thoughtworks.com
@olabini