ORACLE®

Adventures in JSR-292 or How To Be A Duck Without Really Trying

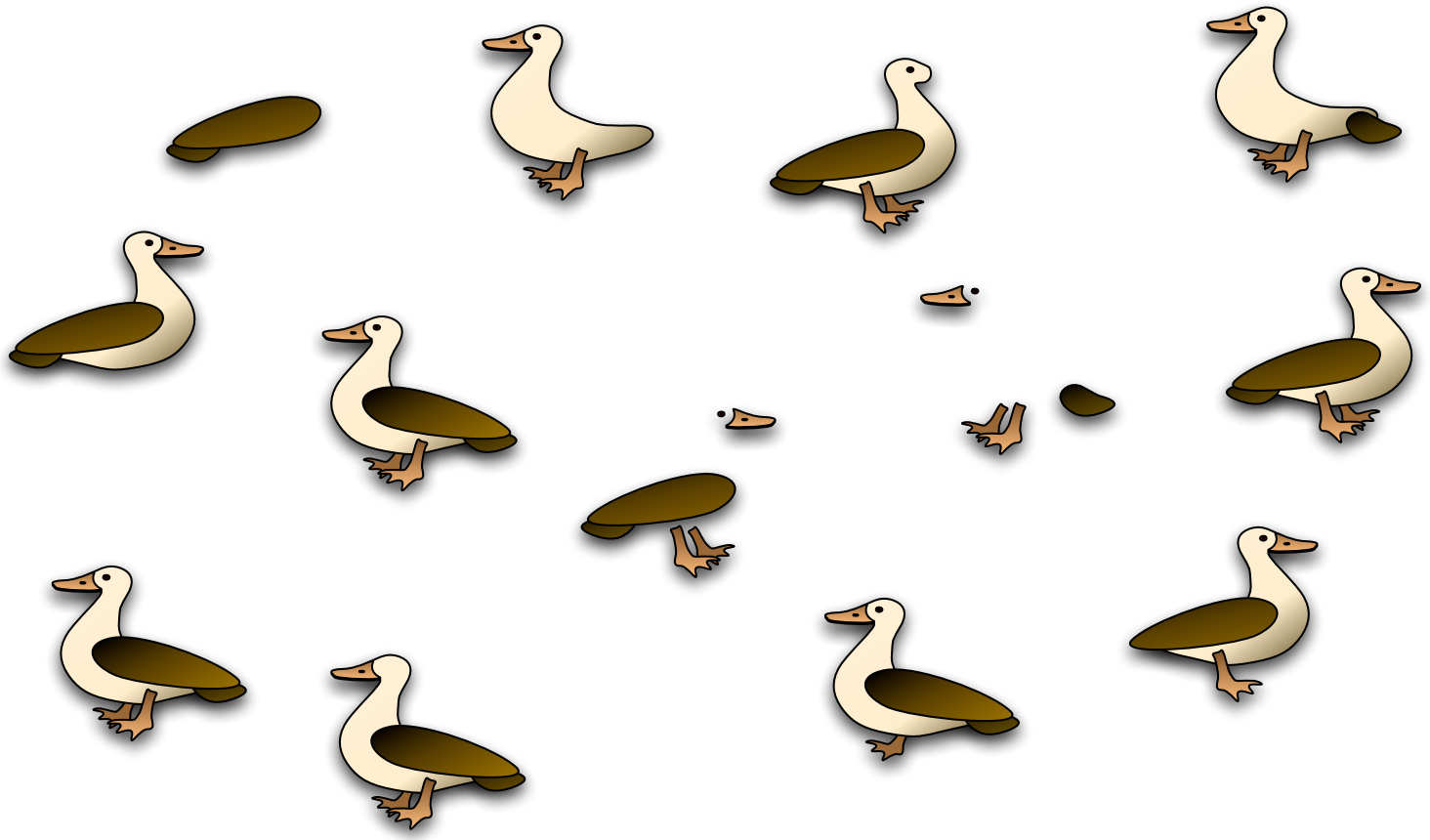
Jim Laskey
Multi-language Lead
Java Language and Tools Group

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Introduction
- Nashorn
- JSR-292 Usage In Nashorn
- Avoiding Polymorphism At CallSites
- Q & A

INTRODUCKTION



The Importance Of Being Multilingual.

4GL
Ada
Algol/Simula
APL
Assembler
Basic/VisualBasic
BLISS
C/C++/Objective-C
C#
COBOL
Eiffel
Flex
Forth
Fortran/Fortran5
FoxPro
HTML/CSS
HyperTalk
Java
JavaScript
LabVIEW
Lisp/Dylan/Scheme
Lua
M (Mumps)
MEL (Maya)
Modula/Oberon
Occam
Pascal/ObjectPascal
Perl
PHP
Pilot/Tudor
PL/1
Postscript
Prograph
Prolog/MicroProlog
Python
QuartzComposer
Ruby
Self
Serius
Smalltalk
Snobol
SQL
Sympl
TorqueScript
Watfor/Watbol
XML
YACC/Bison/Antler

NASHORN

NASHORN Perfect Storm

- JavaFX Script
- HTML5/JavaScript
- Oracle acquisition
- Rhino
- John Rose taunt

NASHORN Goals

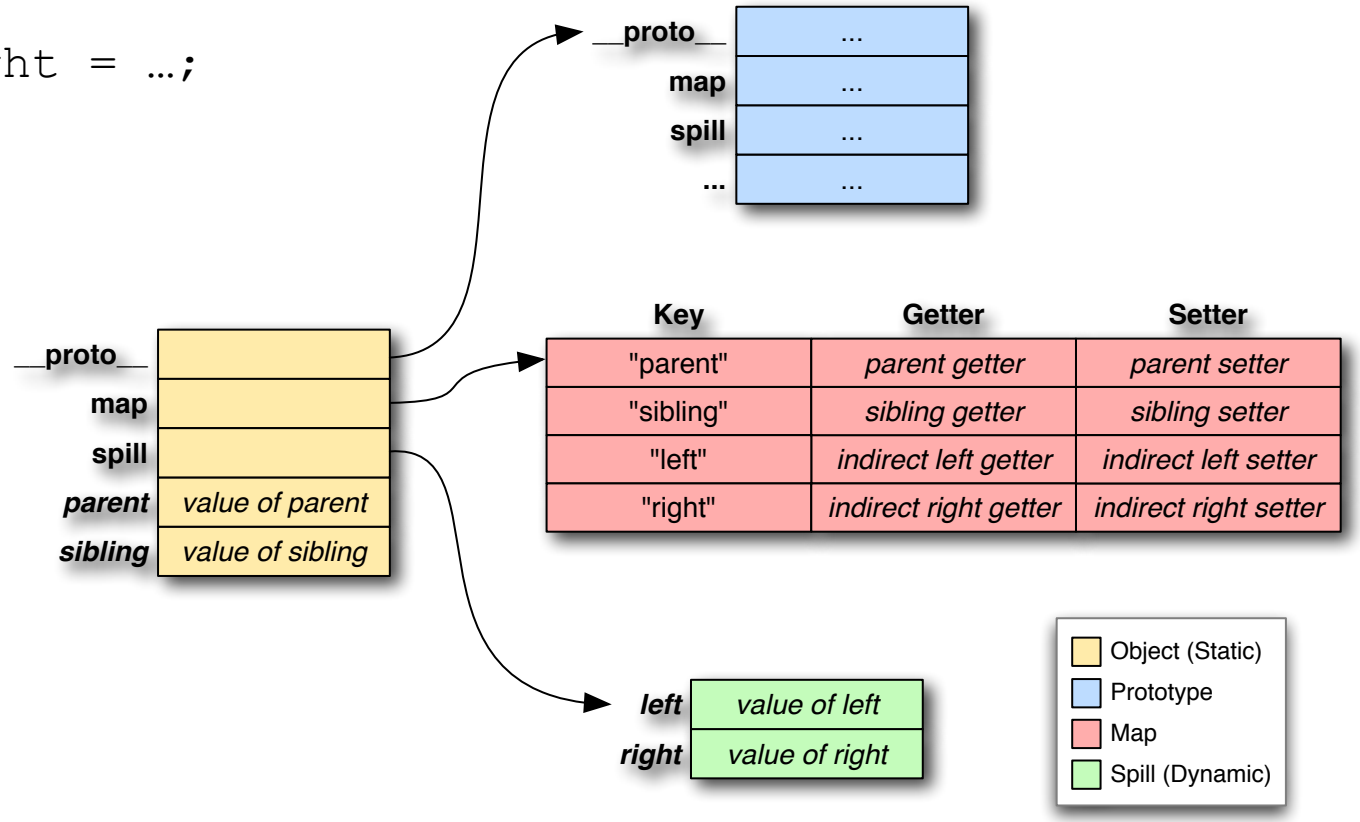
- Make scripting accessible to Java developers
  - Thin API, low overhead, Java objects, collections, Java Beans
- Based on ECMAScript-262
- Example of JSR-292 usage
- Promote the JVM relevance as a multi language platform
- Customizable to specialized needs

NASHORN Tentative Timetable

- Introduction at the JVM Language Summit in summer 2011
- Open source is TBD
- Release with JDK 8 – probably beta in the GA and fully supported in a later release

NASHORN Internals (Simplification)

```
var node = { parent: …, sibling: … };
node.__proto__ = …;
…
node.left = …;
…
node.right = …;
```



NASHORN Objects Objects Everywhere

- Basic objects
  - arrays, date, functions, regexp, ...
- Globals
  - global dynamic object
- Prototypes
  - delegate
- Scopes
  - chained objects
  - not always necessary (eval, arguments, nested functions)

NASHORN JSR-292 Usage In Nashorn

JSR-292 Handling Java void Methods

```
if (mh.type().returnType() == void.class) {
  mh = MethodHandles.filterReturnValue
       (mh, undefinedFilter);
}
…
static final MethodHandle undefinedFilter =
Linker.getMethodHandle
       (NativeJavaObject.class, "undefinedFilter");
…
public static Object undefinedFilter() {
  return ScriptRuntime.UNDEFINED;
}
```

JSR-292 JavaScript Closures

```
function gen() {
    var list = [];

    for (var i = 0; i < 5; i++) {
        list.push(function() { return i; });
    }

    i = "fish";

    return list;
}

var funcs = gen();

for (f in funcs) print(funcs[f]());
```

```
fish
fish
fish
fish
fish
```

JSR-292 JavaScript Closures

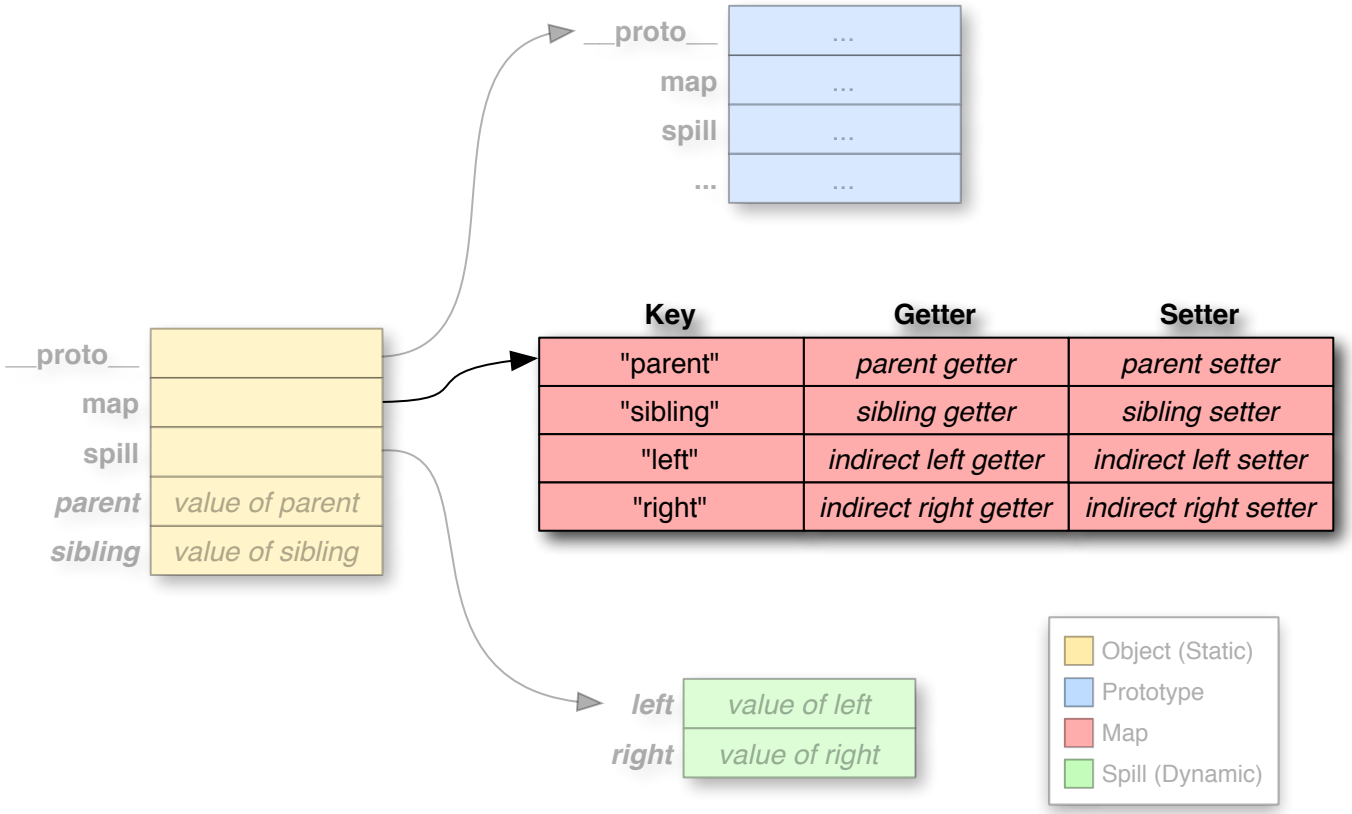
```
static Object gen(Object receiver, ScriptObject scope, …) {
  scope = new Scope(scope, scope$map, …);
  ScriptArray list = newScriptArray(0);

  …

  MethodHandle
    mh =linker.getMethod(scriptClass, "f$1", f$1$mt);
  mh = MethodHandles.insertArguments(mh, 1, scope);
  ScriptFunction function = new Function(mh);
  list.push(function);

  …

  return list;
}
```

JSR-292 Receivers

```
var x = node.left;
```

- InvokeDynamic to “left” with a GET bootstrap
- “left” could be a member of the object, or, a member of the object’s prototype

Binding **Receivers**



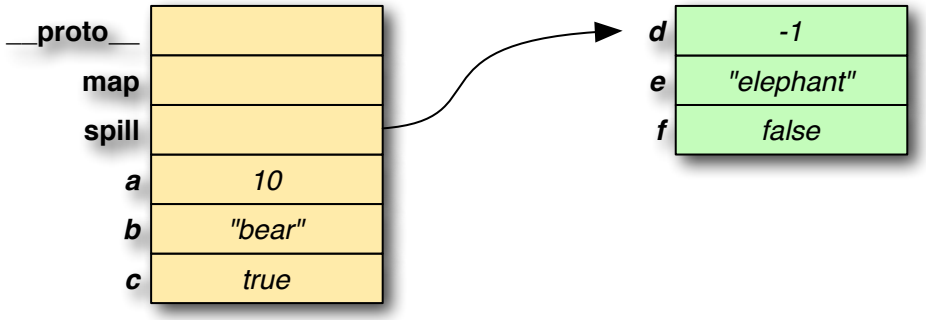
JSR-292 Receivers

- If found in the object's map, bind the getter to the call site as-is
- Otherwise, bind the getter's receiver arg to the prototype and bind the resulting mh to the call site

```
FindResult find = object.findProperty("left");
MethodHandle getter = find.getProperty().getGetter();
if (find.getDepth() == 0) {
  getter = getter.bindTo(find.getPrototype());
  getter = MethodHandles.dropArguments(getter, 0,
                    callSite.type().parameterType(0));
}
callSite.setMethod(getter, …);
```

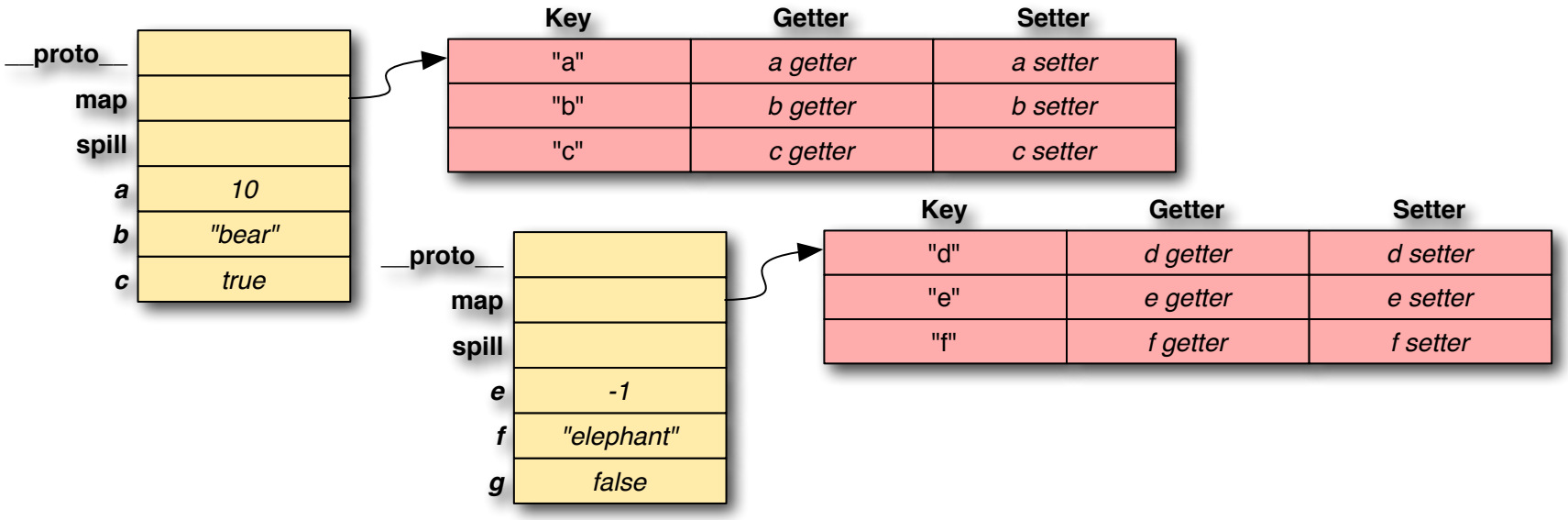
JSR-292 Direct References in Dynamic Situations

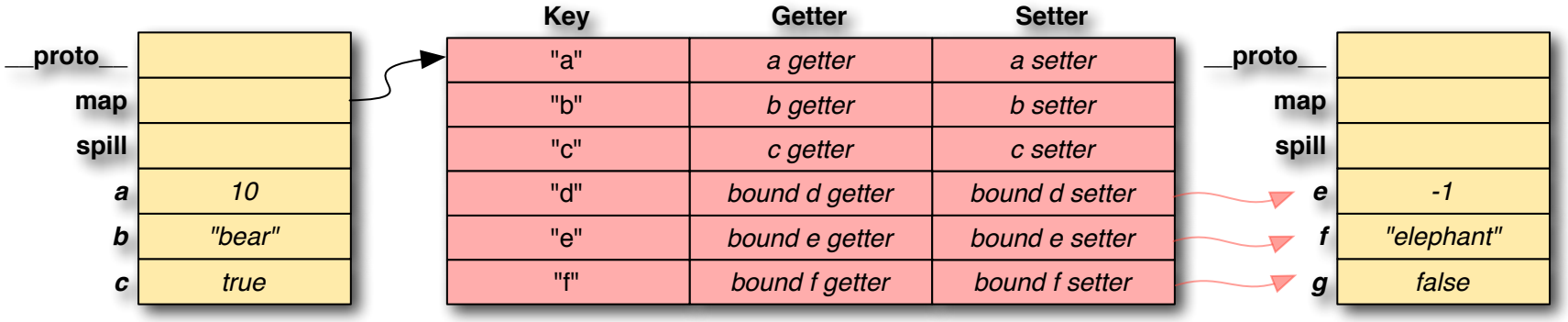
```
a = 10; b = "bear"; c = true;
eval("d = -1; e = "elephant"; f = false;");
```



- Global is a singleton
  - Getters/setters can be bound to the global
  - Can be bound to anything
- Map merging

JSR-292 Direct References in Dynamic Situations





POLYMORPHISM

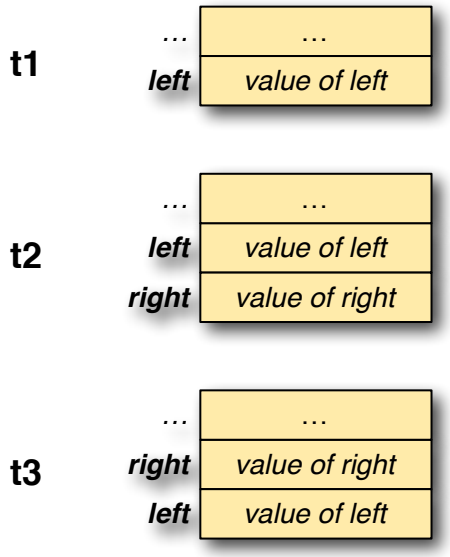
Avoiding Polymorphism At CallSites

POLYMORPHISM The Problem

```
var node = { parent: …, sibling: … };
node.__proto__ = …;
…
node.left = …;
…
node.right = …;
…
f(node.left);
```

OR

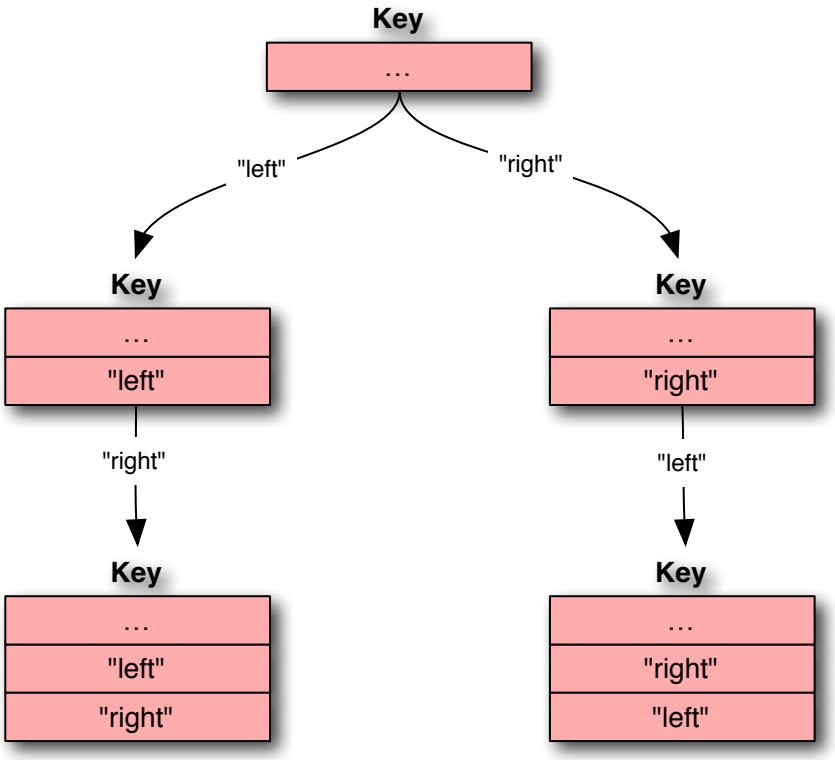
```
…
node.right = …;
…
node.left = …;
```




```
if (node isa t3)
   return t3.left
else if (node isa t2)
   return t2.left
else if (node isa t1)
   return t1.left
else
   lookup
```

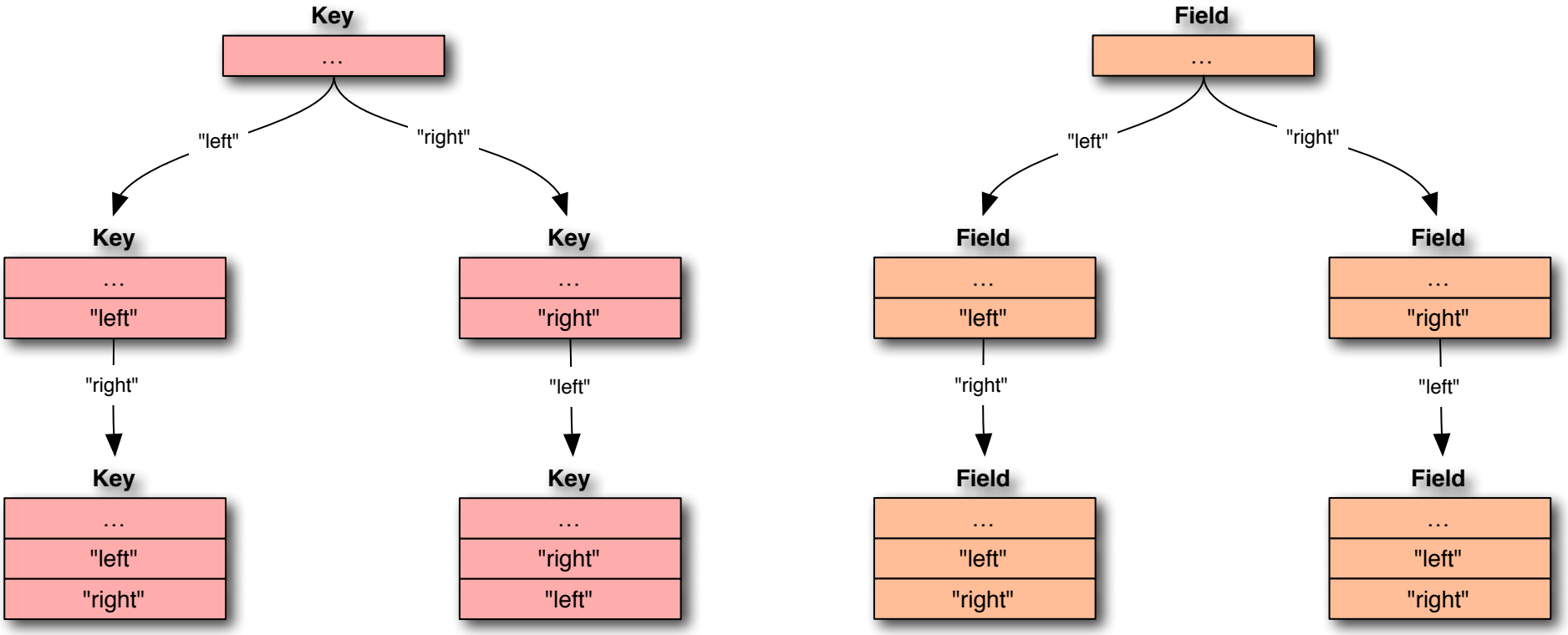
POLYMORPHISM By Map

- Guard by testing map
  - Immutable and interned to prevent proliferation



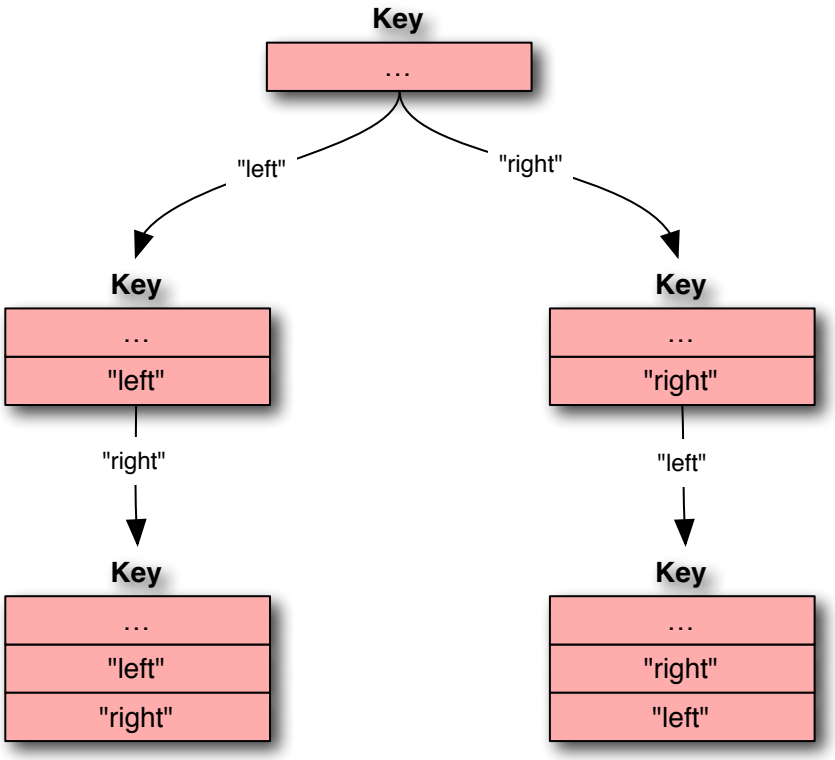
POLYMORPHISM By Organization

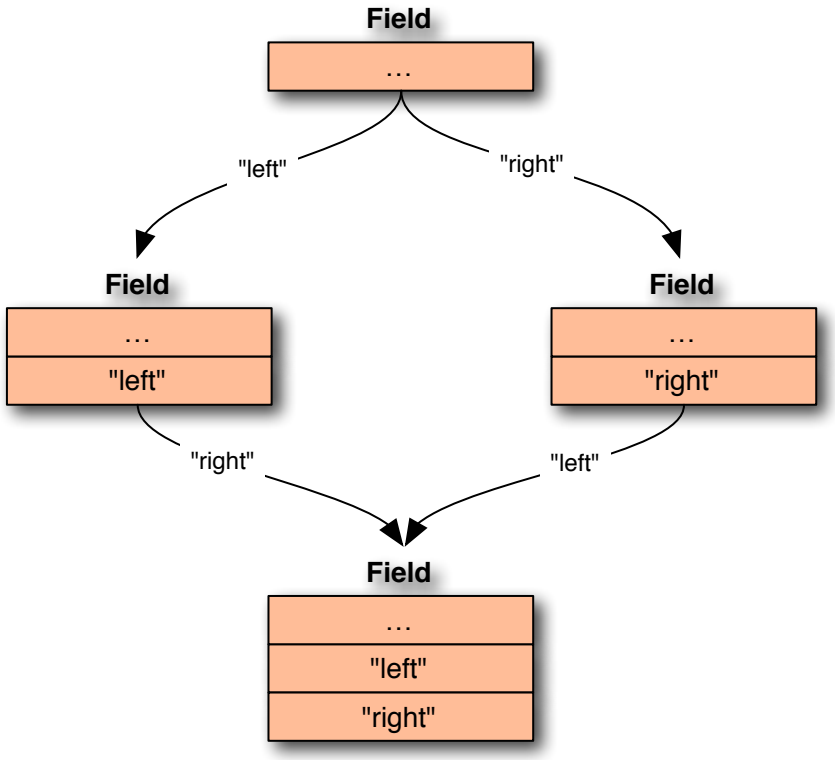
- Guard by testing organization
  - Reduce the number of cases by sorting fields



POLYMORPHISM By Organization

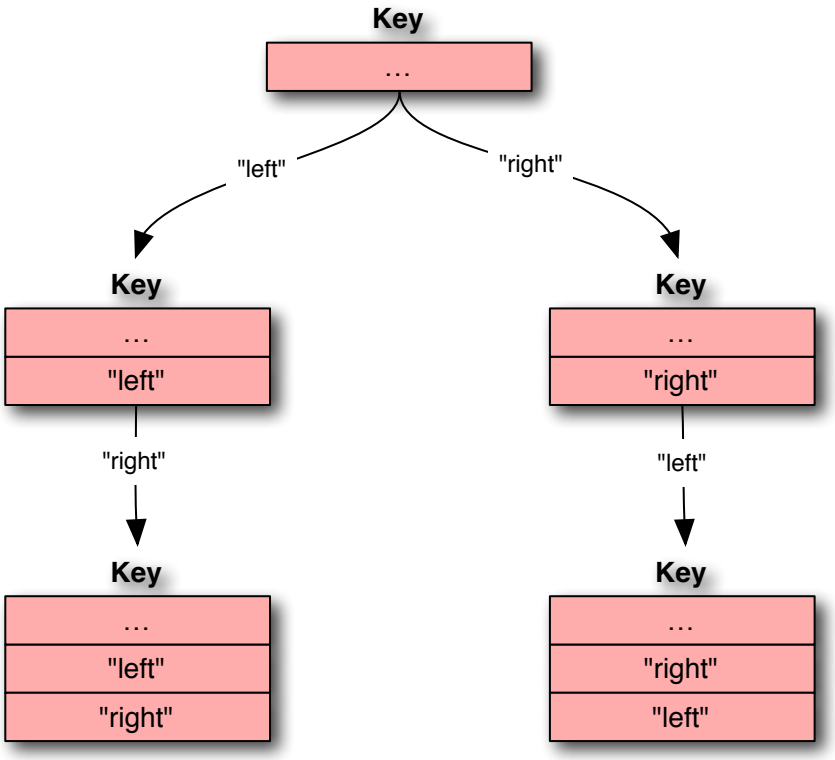
- Guard by testing organization
  - Reduce the number of cases by sorting fields

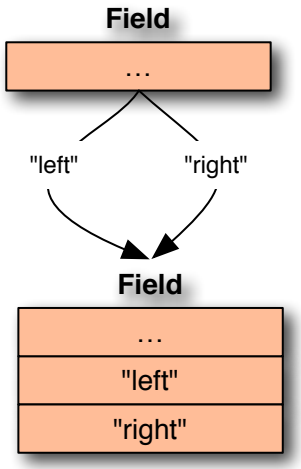




POLYMORPHISM By Predicted Organization

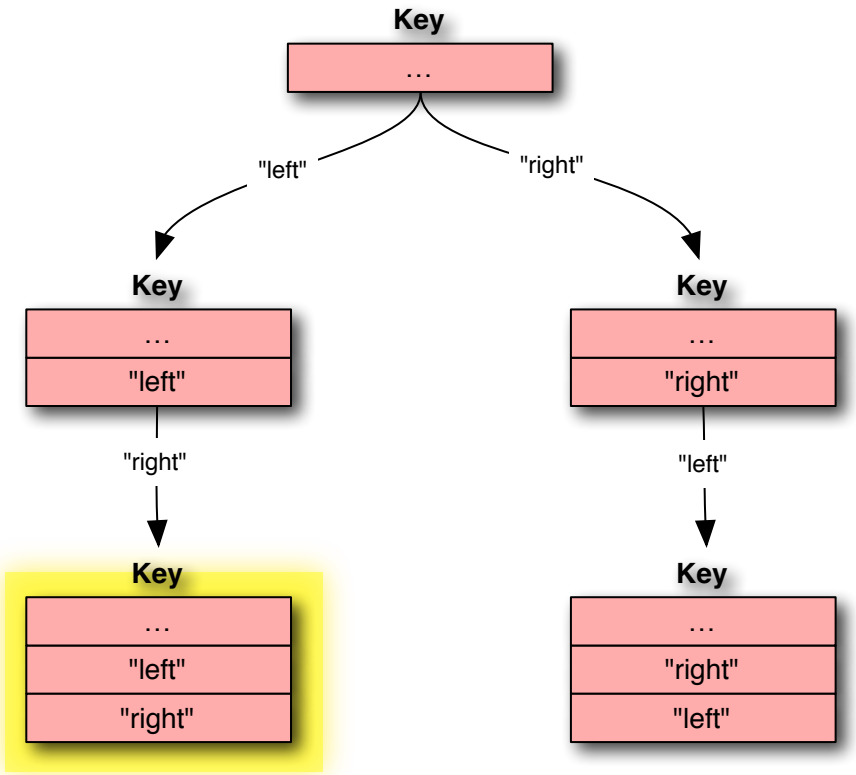
- Assume history will repeat itself

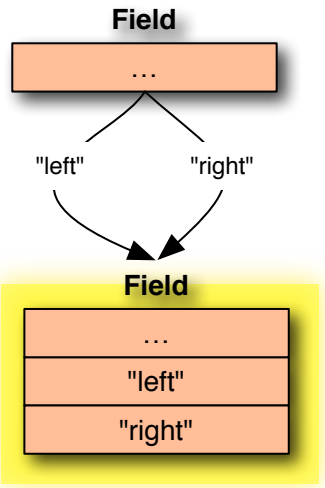




POLYMORPHISM First To

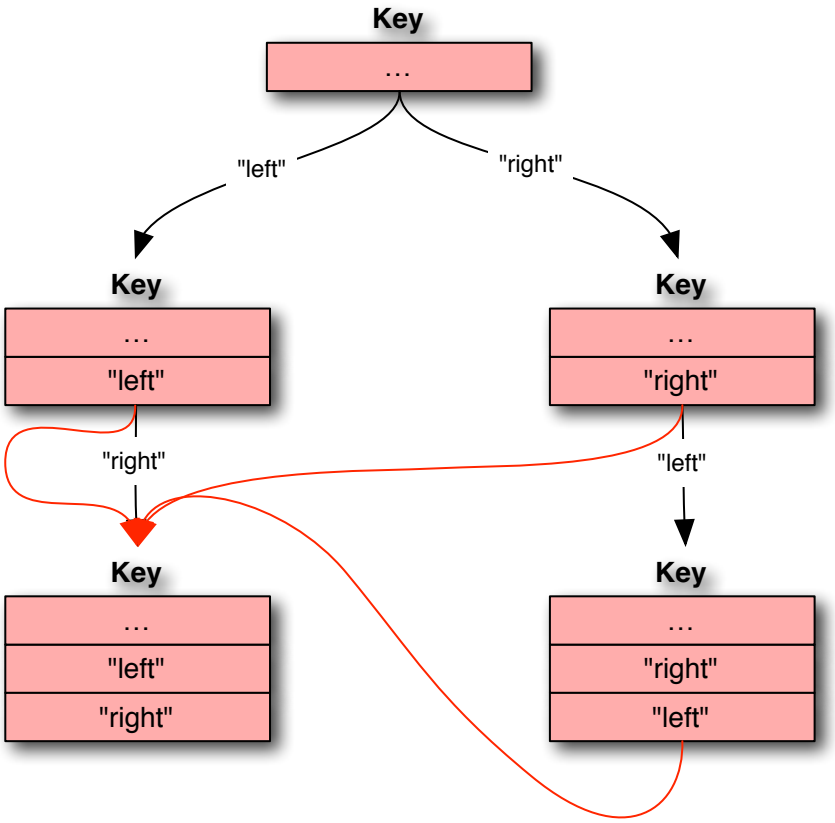
- No additional structure required.



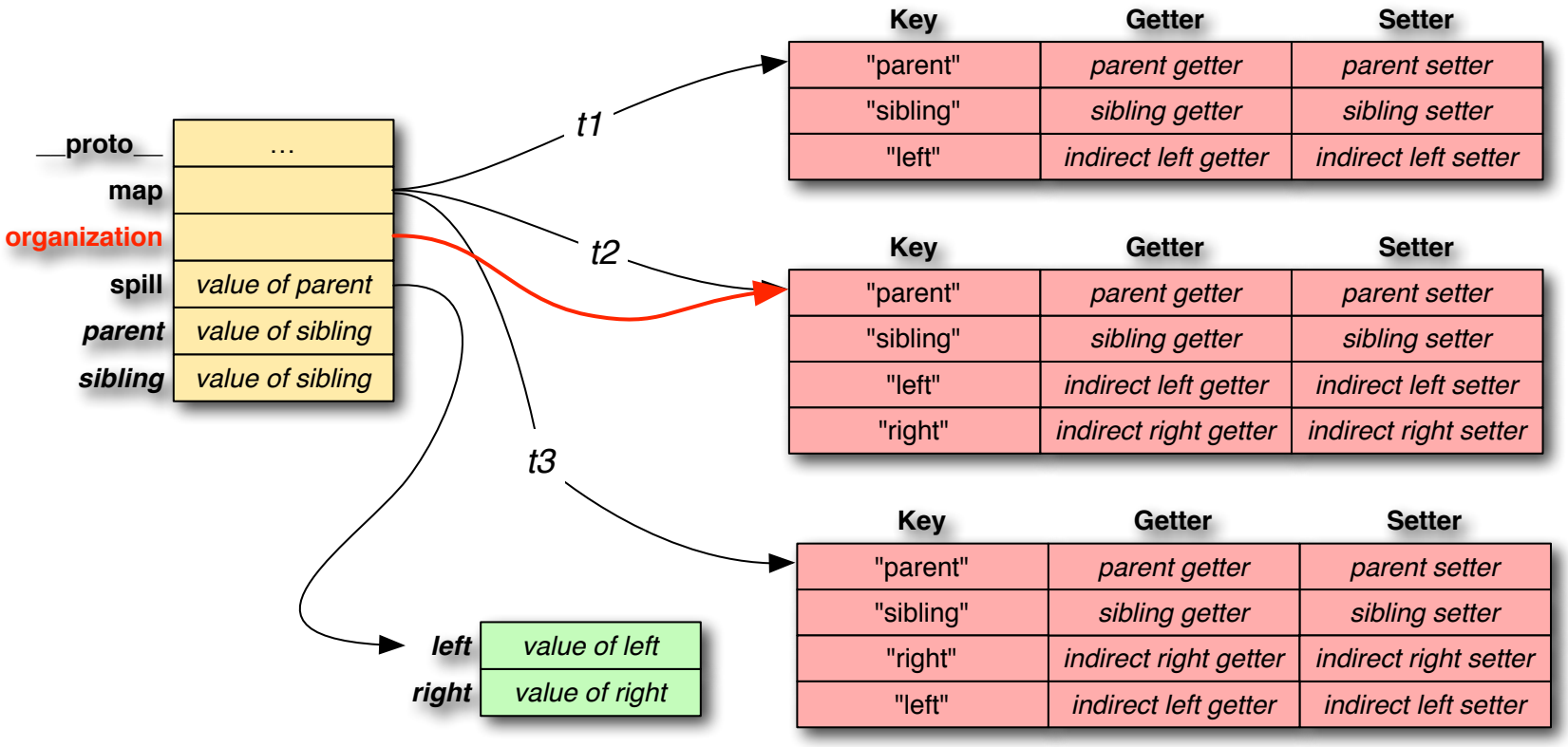


POLYMORPHISM First To

- No additional structure required.



POLYMORPHISM Only The Map Changes



POLYMORPHISM Cost/Benefit

- The spill is dynamic anyway, no net loss
  - Allow for new properties,
  - Implies copying and slop
  - Quanta, cache lines, yada, yada, yada
  - Less copying
- Assuming that organization patterns will reoccur, yield of less thrashing at CallSites

POLYMORPHISM

NASHORN Contacts

Jim Laskey
james.laskey@oracle.com

mlvm-dev@openjdk.java.net

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Q&A