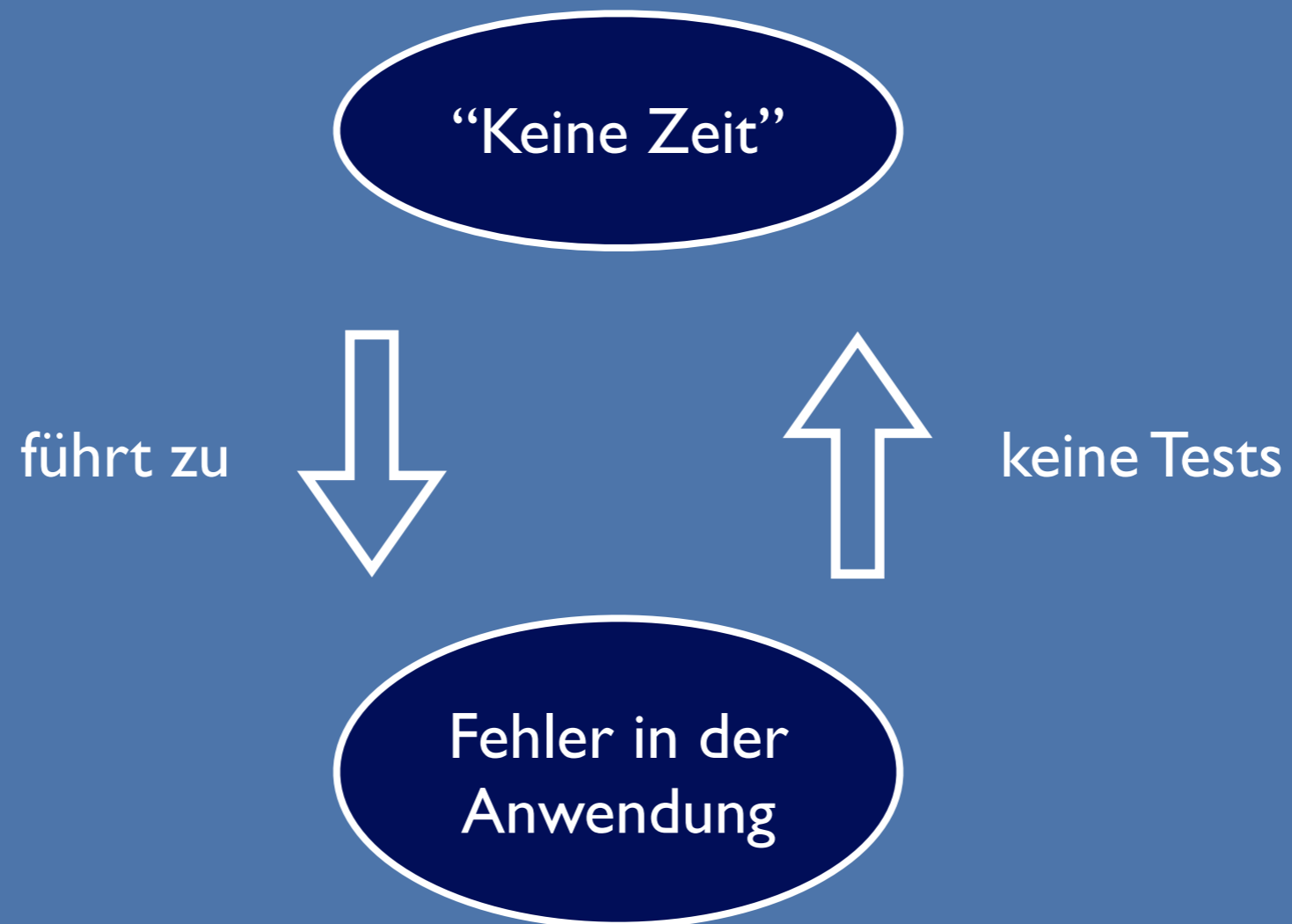


JUnit

"It compiles, therefore
it must work"

Teufelskreis



Sinn

Sinn von JUnit:

Testen von Einheiten (= Klassen)

nicht jedoch:

Testen der Applikation

Liebling “println”

```
public class MySum {  
    public static int addTwoInt(int a, int b) {  
        return a+b;  
    }  
    public static void main(String[] args) {  
  
        if (addTwoInt(1, 2) == 3)  
            System.out.println("addTwoInt arbeitet korrekt");  
  
    }  
}
```

Erster Testfall

```
public class MySum {  
    public static int addTwoInt(int a, int b) { return a+b; }  
}
```

```
import junit.framework.*;
```

```
public class MySumTest extends TestCase {
```

```
    public void testAddTwoInt() {  
        Assert.assertEquals(MySum.addTwoInt(1, 2), 3);  
    }
```

```
    public static void main(String[] args) {
```

```
        junit.textui.TestRunner.run(MySumTest.class);
```

```
    }  
}
```

Test fehlgeschlagen - I.

```
import junit.framework.*;

public class AssertFails extends TestCase {
    public void testPassed() {
        Assert.assertEquals(1, 1);
    }
    public void testFails() {
        Assert.assertEquals(1, 2);
    }
}
```

Test fehlgeschlagen - 2.

```
$ java junit.textui.TestRunner AssertFails
```

```
..F
```

```
Time: 0.013
```

```
There was 1 failure:
```

```
1)
```

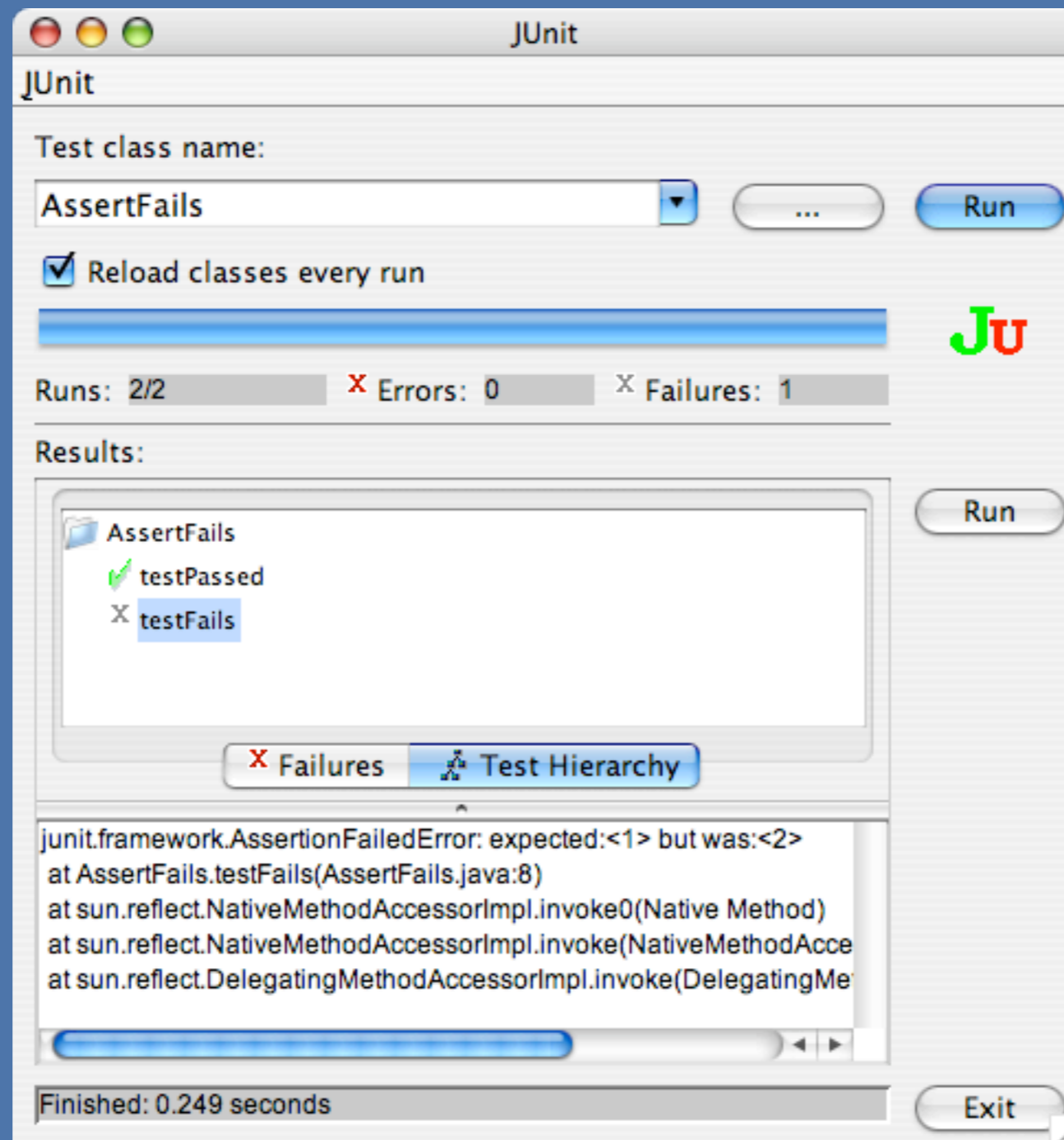
```
testFails(AssertFails)junit.framework.AssertionFailedError:  
expected:<1> but was:<2>
```

```
:
```

```
FAILURES!!!
```

```
Tests run: 2, Failures: 1, Errors: 0
```

Test fehlgeschlagen - 3.

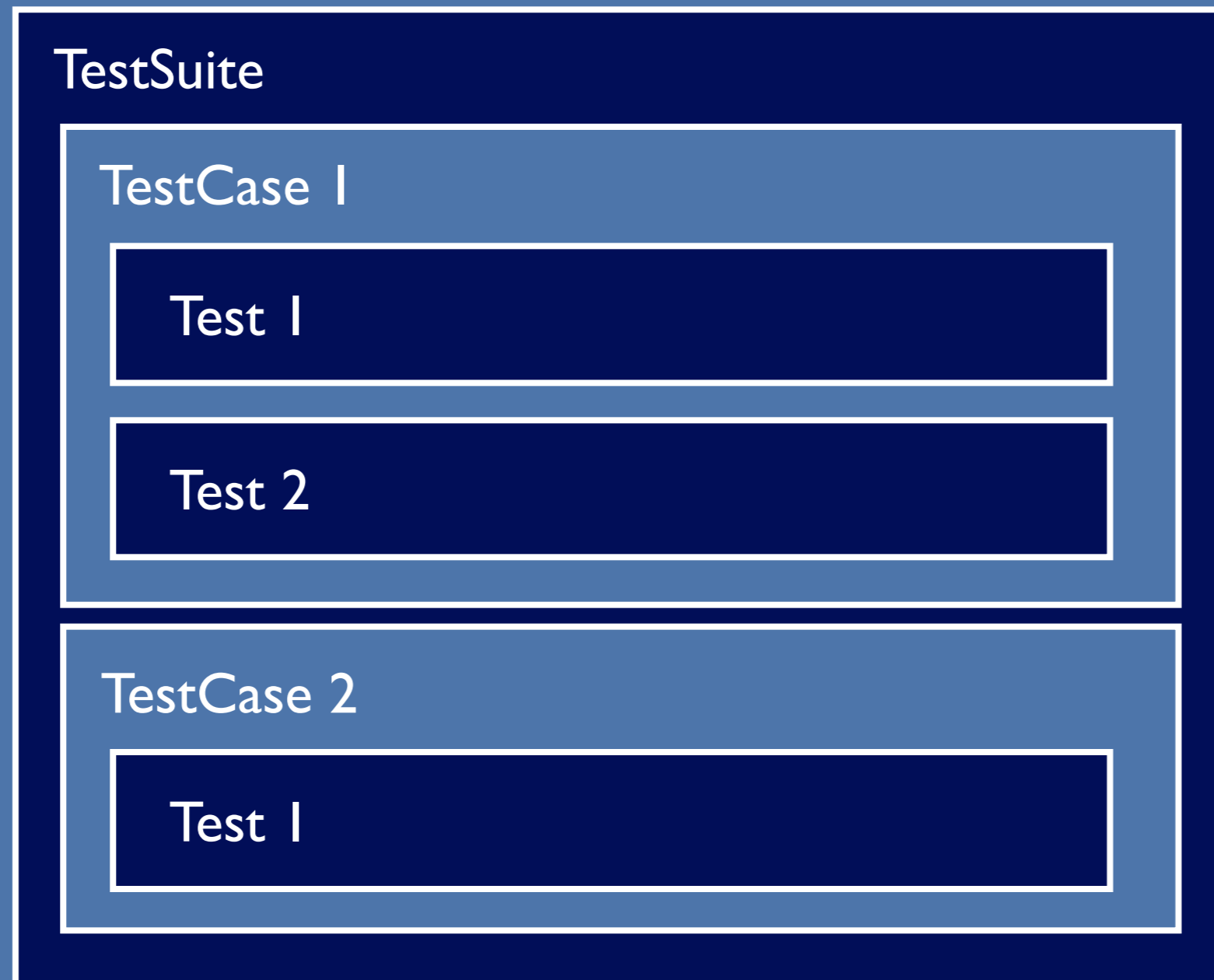


Exceptions

```
import junit.framework.*;

public class ExceptionTest extends TestCase {
    public void testPassed() {
        try {
            throw new Exception("Exception einer Methode");
        } catch (Exception e) {
            Assert.fail("Eine Exception ist aufgetreten");
        }
    }
}
```

Schalenmodell



Eine Suite

```
public class AnotherTest extends TestCase {  
    public void testThree() {}  
}
```

```
public class ThreeTests extends TestCase {  
    public ThreeTests(String name) { super(name); }  
    public void testZero() {}  
    public void testOne() {}  
    public void testTwo() {}  
    public static Test suite() {  
        TestSuite suite = new TestSuite();  
        suite.addTest(new ThreeTests("testOne"));  
        suite.addTest(new ThreeTests("two") {  
            public void runTest() { testTwo(); }  
        });  
        suite.addTestSuite(AnotherTest.class);  
        return suite;  
    }  
}
```

Fixtures - I.

- setUp = vor jedem Test
- tearDown = nach jedem Test

Fixtures - 2.

```
import junit.framework.*;
import java.util.ArrayList;

public class Fixtures extends TestCase {
    private ArrayList list;

    public void setUp() {
        list = new ArrayList();
    }

    public void testWithList() {
        Assert.assertNotNull(list);
    }
}
```

Die Asserts

- assertEquals
- assertFalse / True
- assertNotNull / Null
- assertNotSame
- fail

JUnit & Eclipse

The screenshot displays the Eclipse IDE interface during a JUnit test run. On the left, the 'JUnit (AssertFails)' window shows the test results: 'Runs: 2/2', 'Errors: 0', and 'Failures: 1'. Below this, the 'Failures' tab is active, showing a list of failed tests: 'testFails - AssertFails'. The 'Failure Trace' section below the list shows the following stack trace:

```
junit.framework.AssertionFailedError: expected:<1> but was:<2>  
at AssertFails.testFails(AssertFails.java:8)  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

On the right, the 'AssertFails.java' source code is visible. The code is as follows:

```
1 import junit.framework.*;  
2  
3 public class AssertFails extends TestCase {  
4     public void testPassed() {  
5         Assert.assertEquals(1, 1);  
6     }  
7     public void testFails() {  
8         Assert.assertEquals(1, 2);  
9     }  
10 }  
11
```

Line 8, which contains the failing assertion `Assert.assertEquals(1, 2);`, is highlighted in blue.

JUnit & Ant

```
project default="runtests"
```

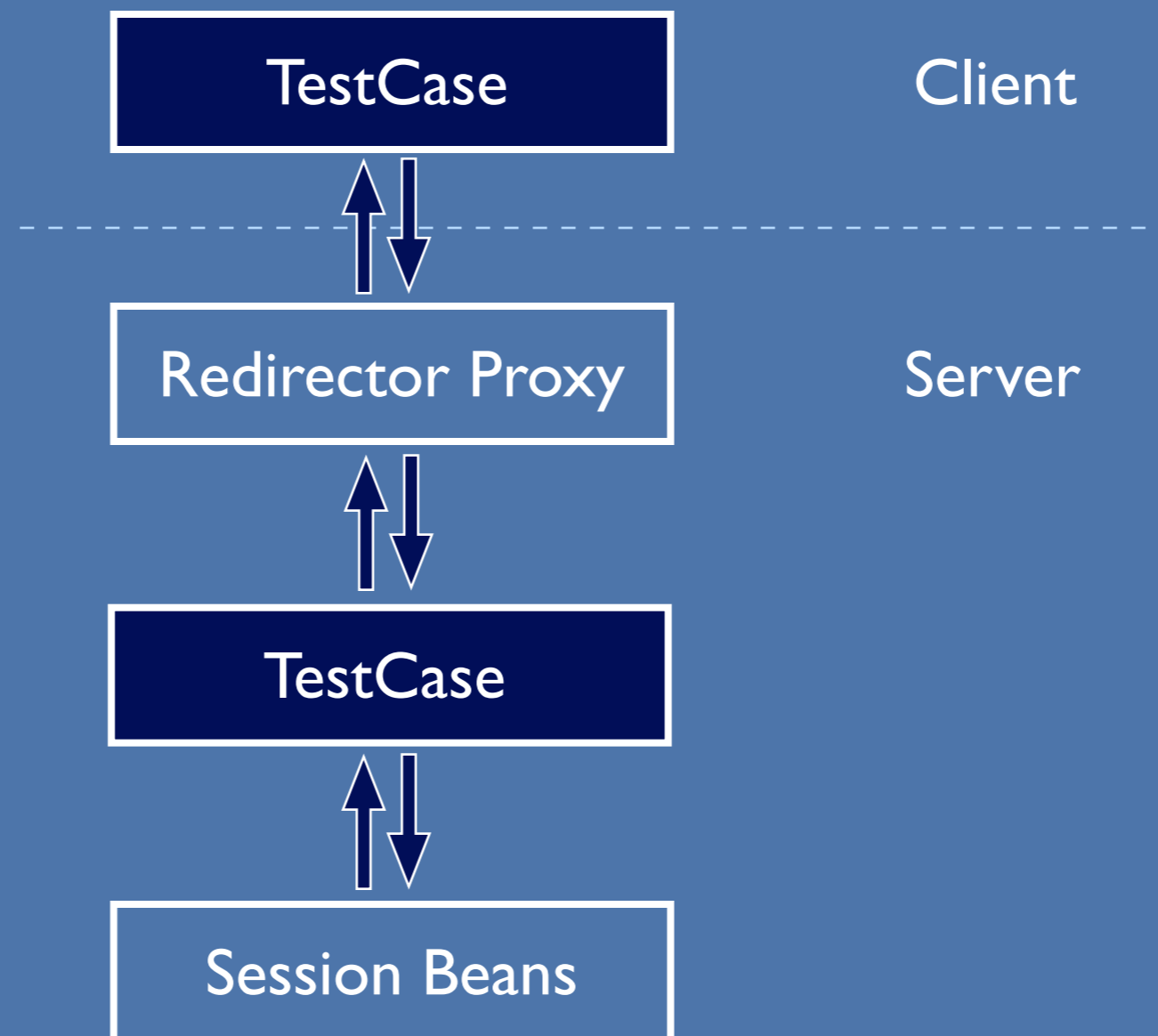
```
target "compile"
```

```
target "compiletests" depends="compile"
```

```
target "runtests" depends="compiletests"
```

```
<target name="runtests" depends="compiletests">  
  <java fork="yes" classname="junit.textui.TestRunner"  
    taskname="junit" failonerror="true">  
    <arg value="MySum"/>  
  </java>  
</target>
```

Cactus



Quellen

- <http://www.junit.org>
- <http://ant.apache.org/manual>
- <http://jakarta.apache.org/cactus>