JUDCon

JBoss Users & Developers Conference

London:2011

Arquillian

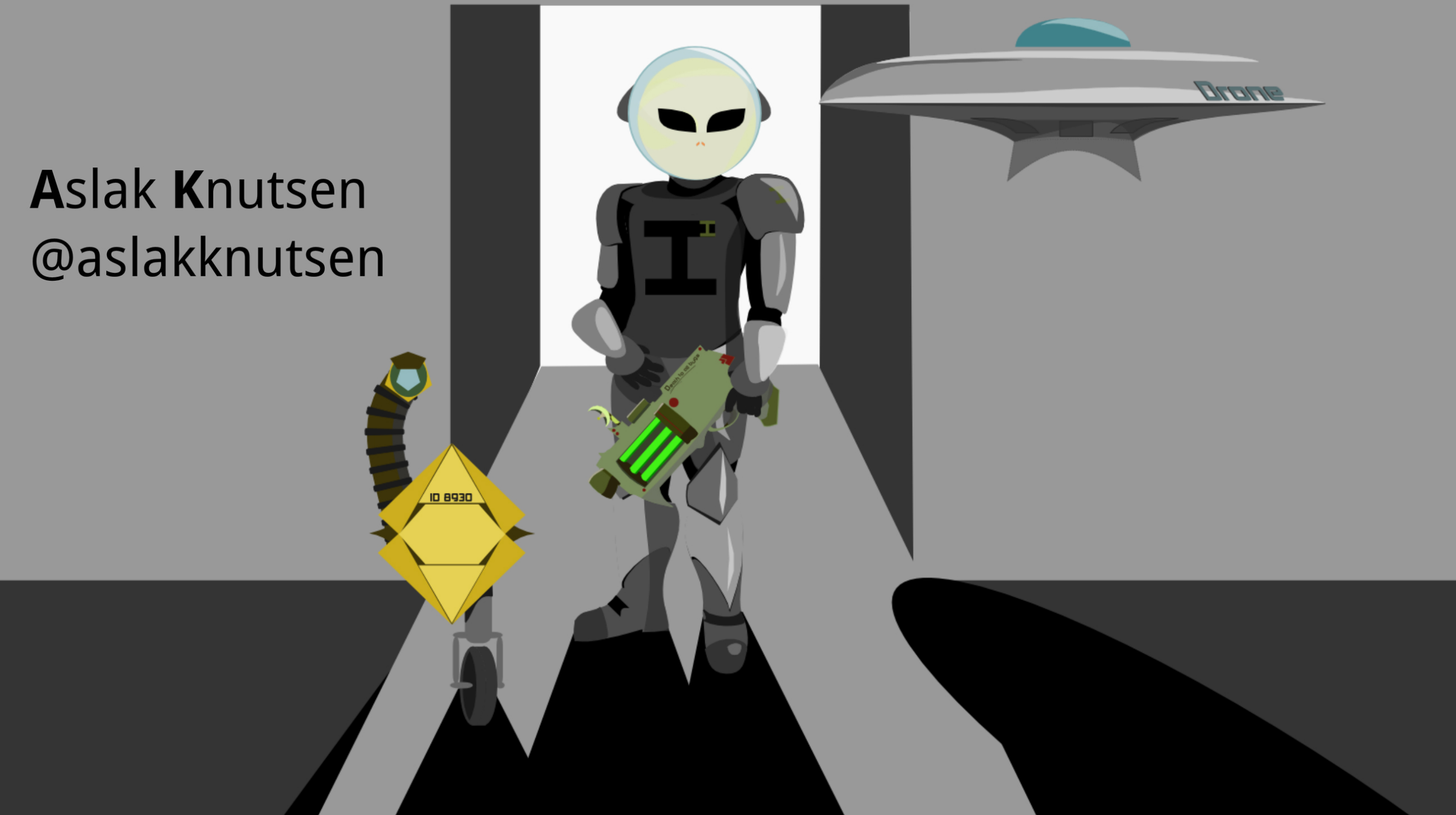
The Extendable Enterprise Test Platform

Arquillian

The Extendable Enterprise Test Platform


Arquillian
The Extendable Enterprise
Test Platform

Aslak **K**nutsen
@aslakknutsen

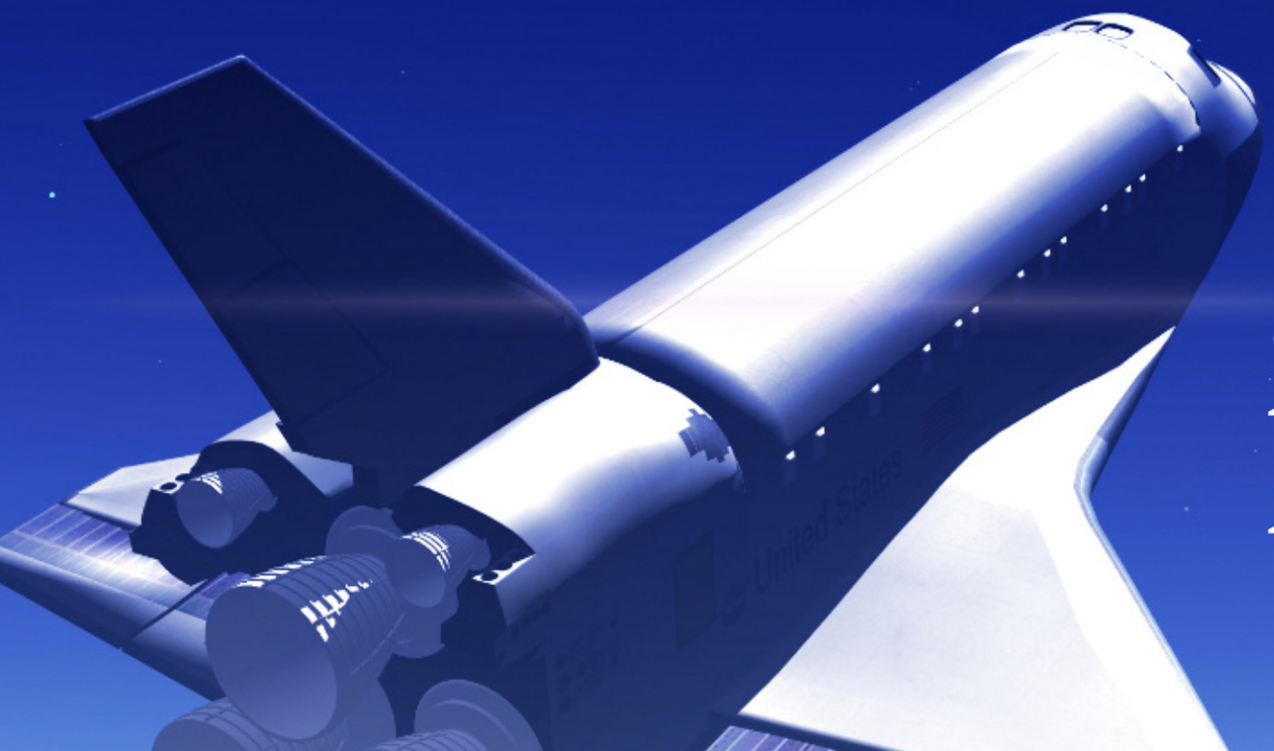


http://arquillian.org

Arquillian

A container-oriented testing framework

Bring your test to the runtime...



...instead of managing the runtime from your test.

Less (boilerpate) test code

As much or as little integration as you need

Looks like a unit test, but executes in a true environment

Same test can be run in multiple environments

It's a **learning** environment

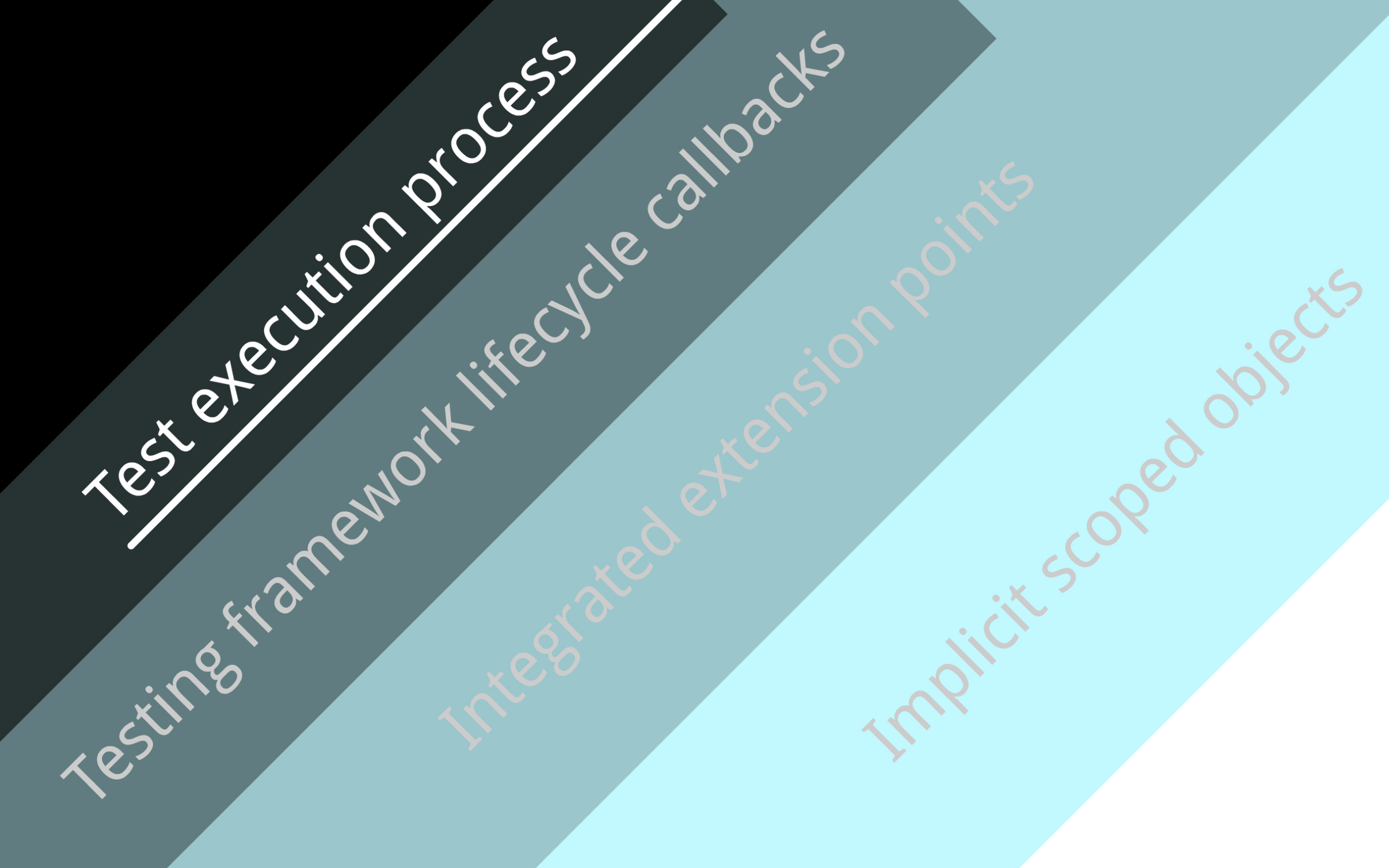
```
public class GreeterTestCase {









}
```

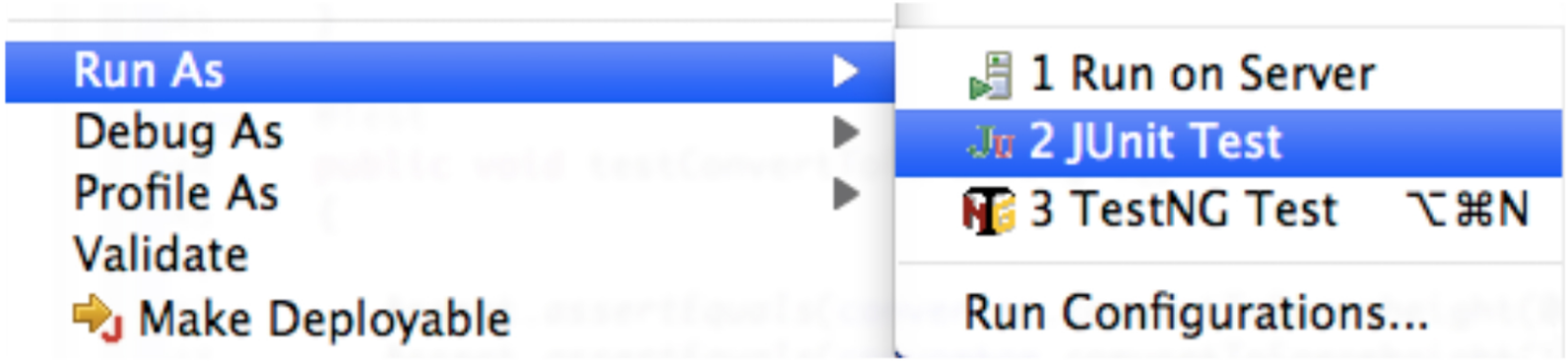
```
@RunWith(Arquillian.class)
public class GreeterTestCase {









}
```

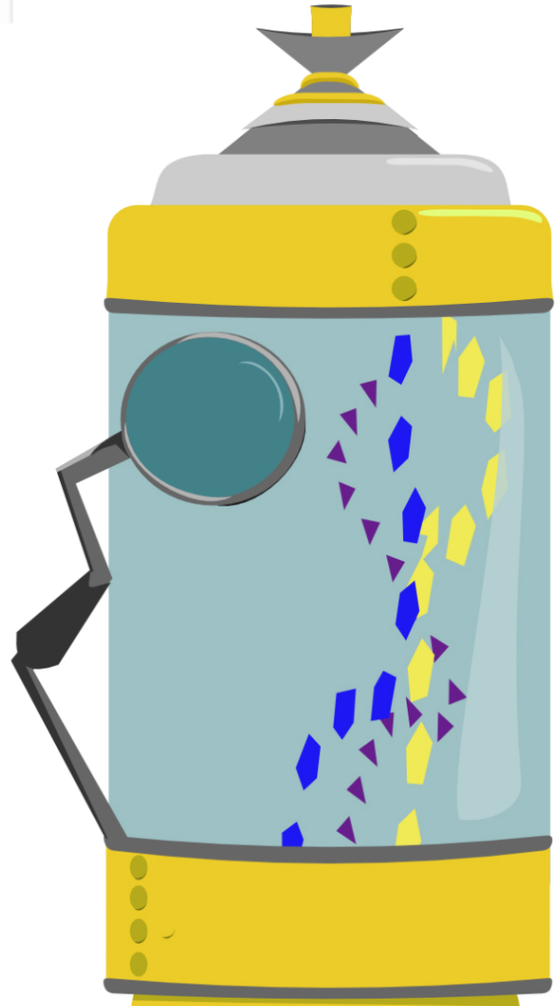
```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }









}
```

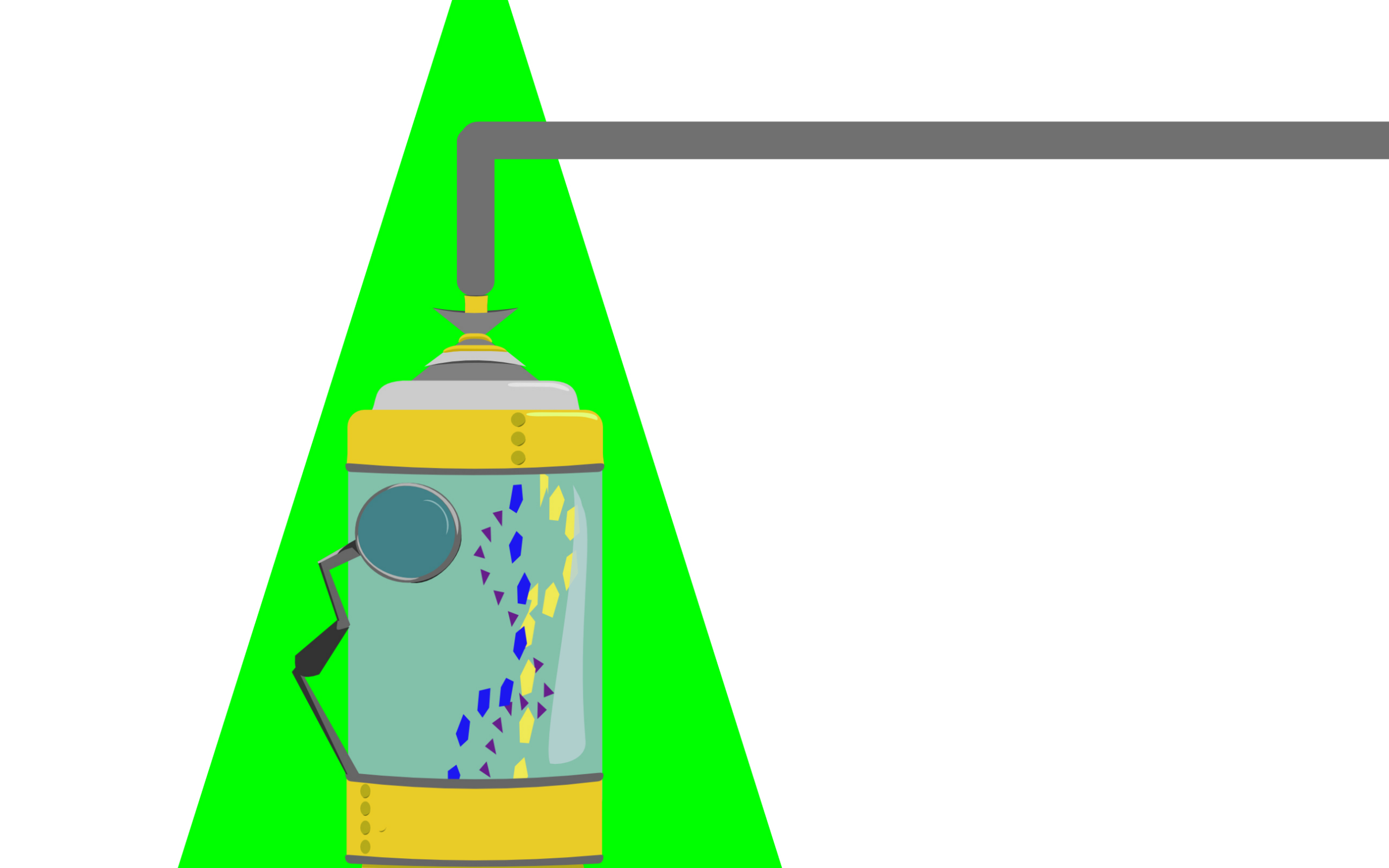
```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Inject Greeter greeter;




}
```

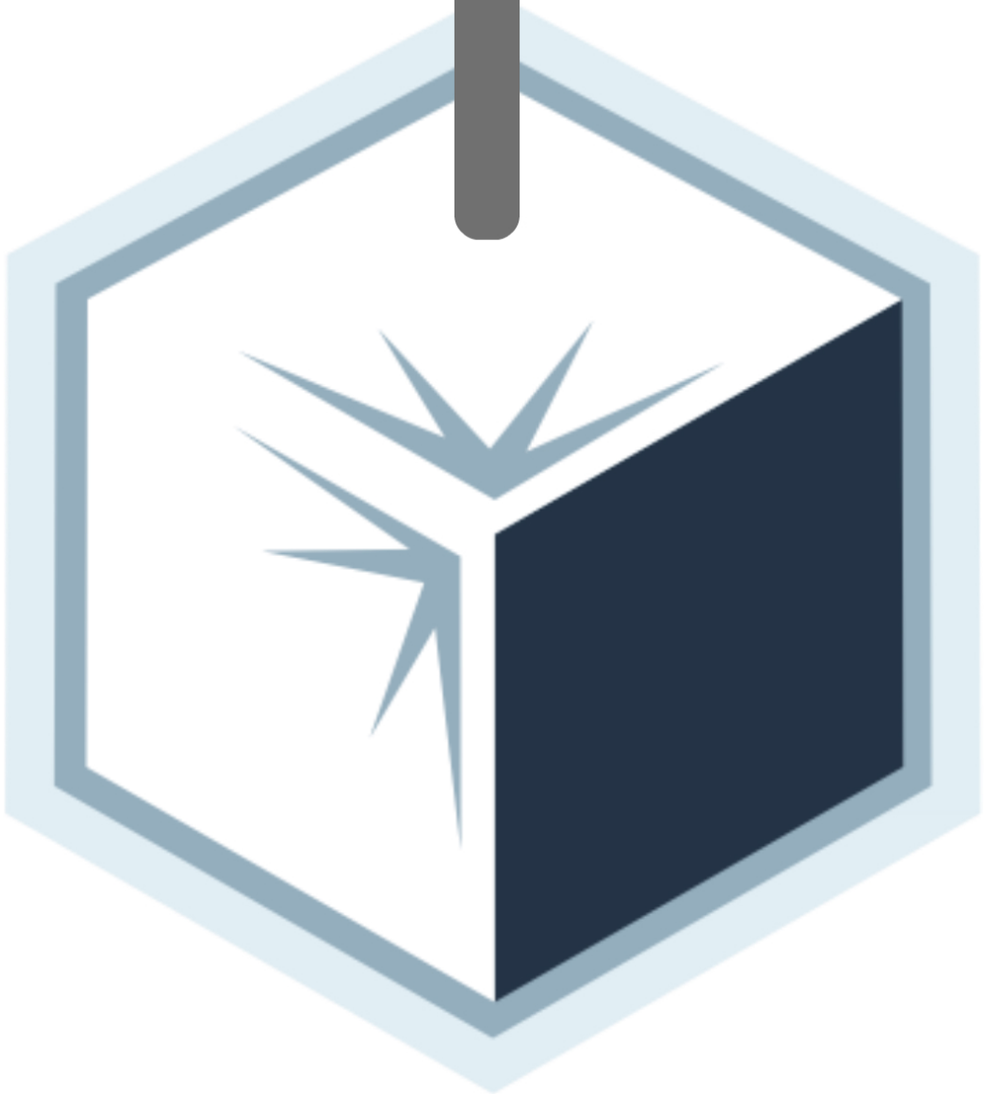
```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Inject Greeter greeter;

    @Test
    public void shouldBeAbleToInvokeBean() throws Exception {
        Assert.assertEquals("Hello, Earthlings", greeter.greet("Earthlings"));
    }
}
```

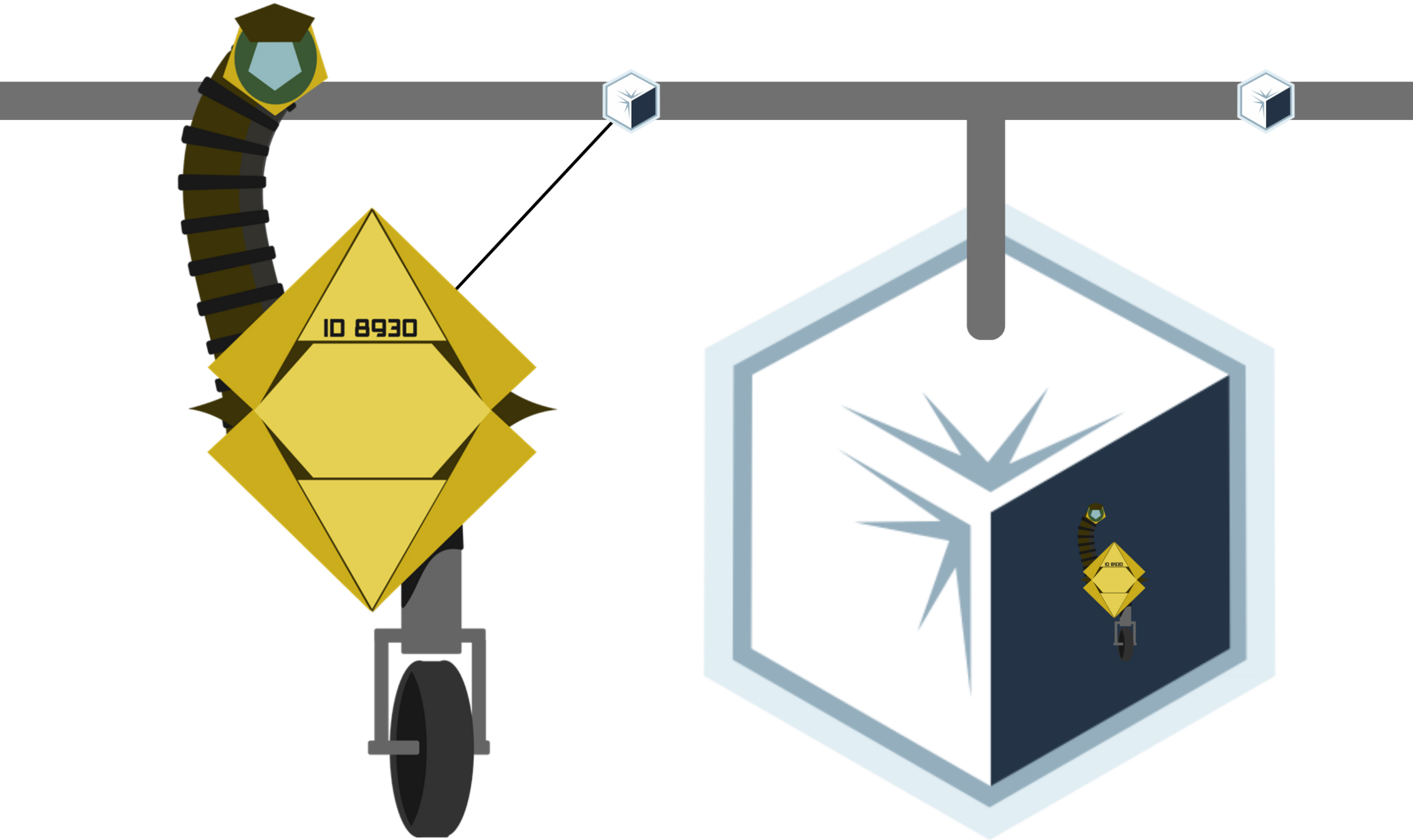

Test execution process
Testing framework lifecycle callbacks
Integrated extension points
Implicit scoped objects

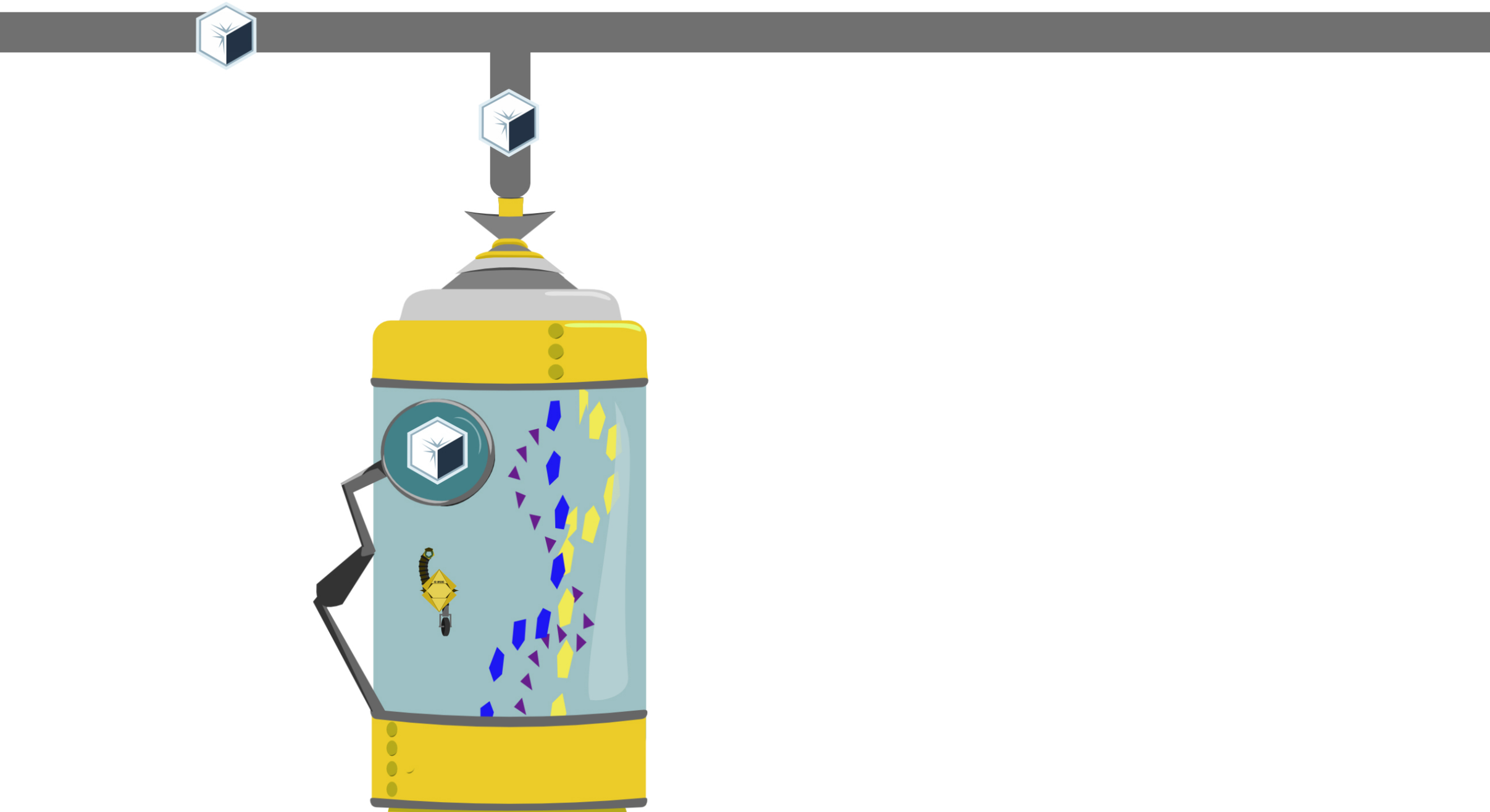

Run As
Debug As
Profile As
Validate
Make Deployable
1 Run on Server
2 JUnit Test
3 TestNG Test ⌥⌘N
Run Configurations...

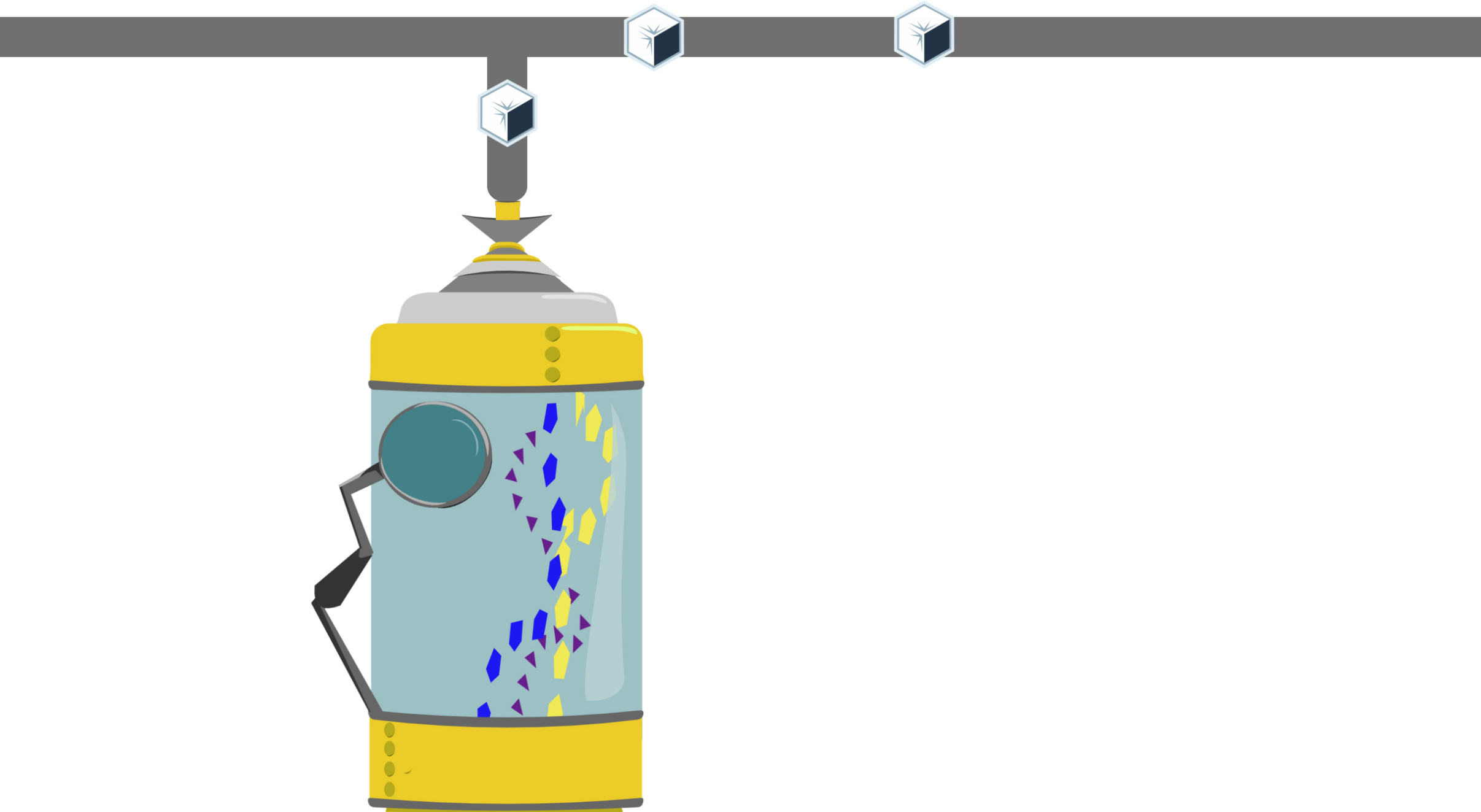


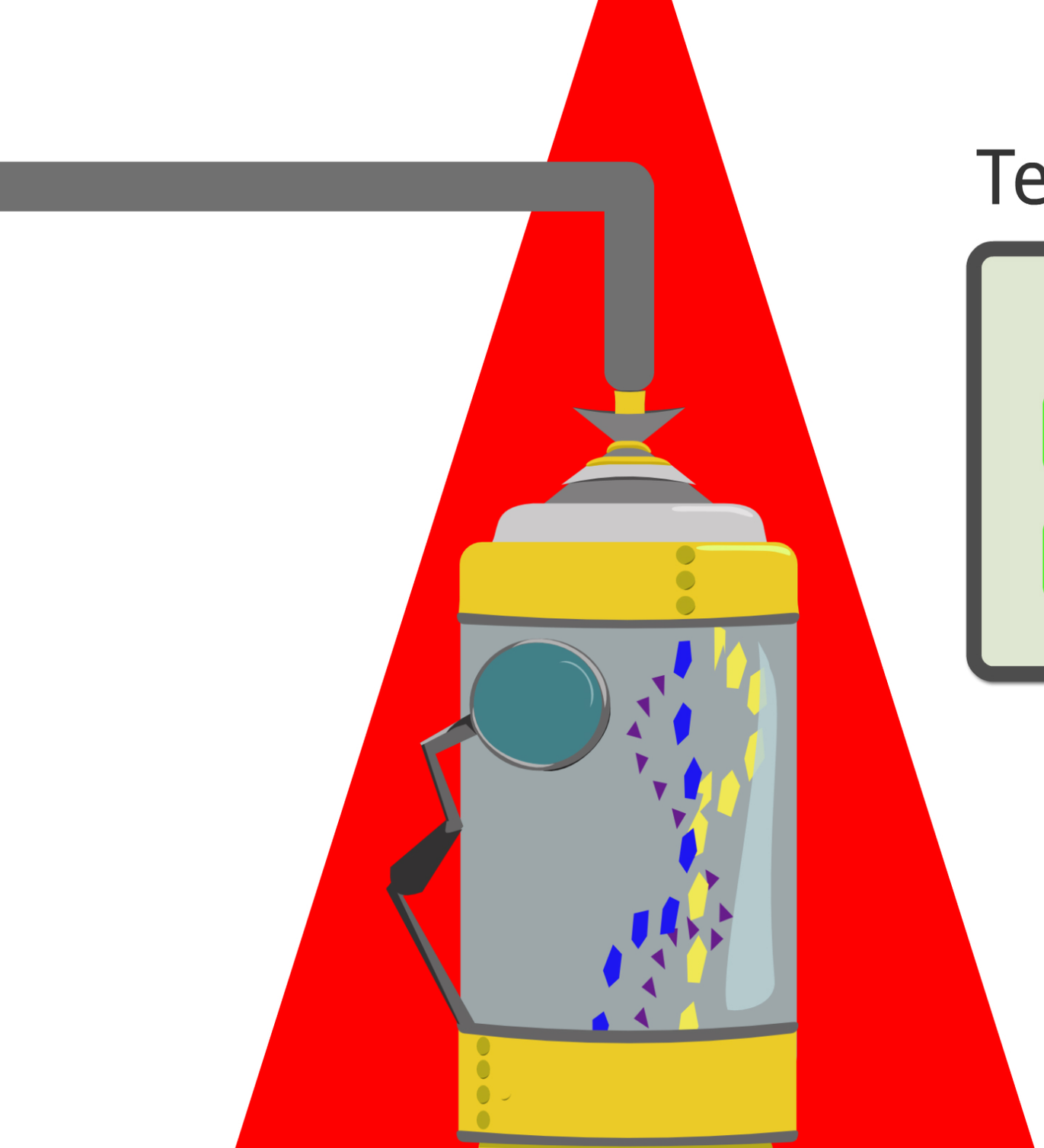





ID 8930







Test Results

Runs: 2/2 Failures: 0 Errors: 0

In Container Test

Test Framework

JUnit

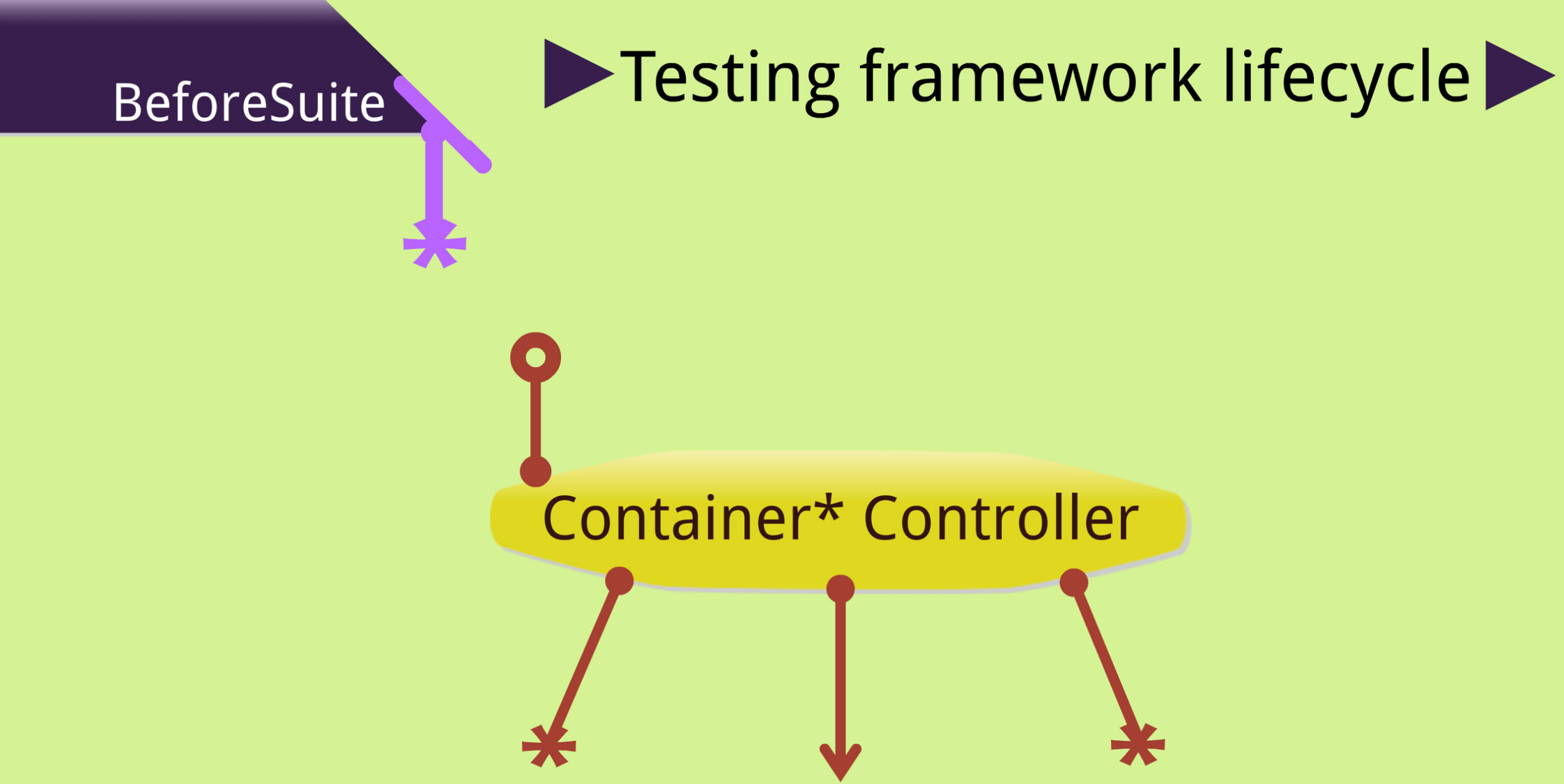
TestNG

AS 7

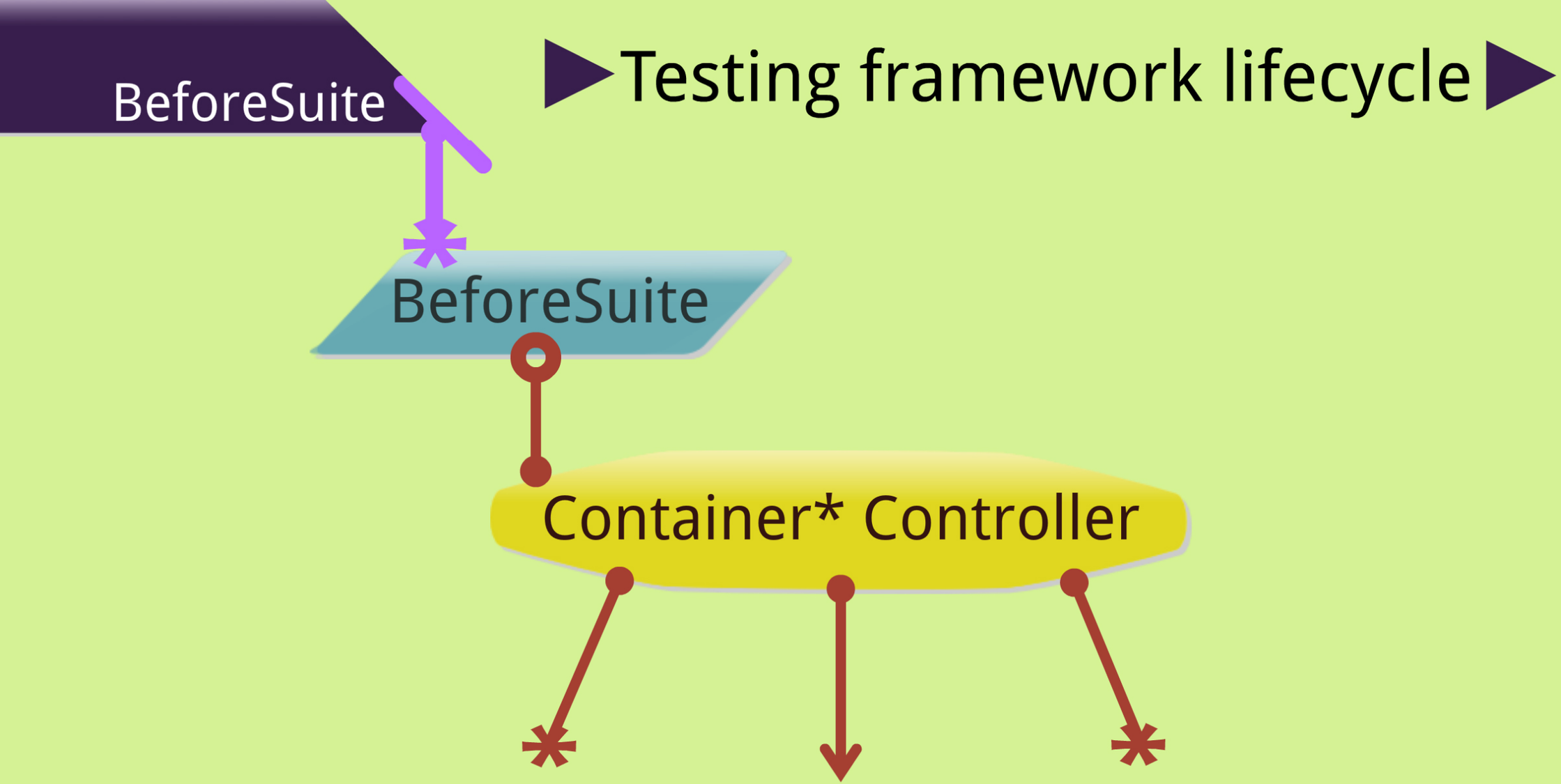
GlassFish

...

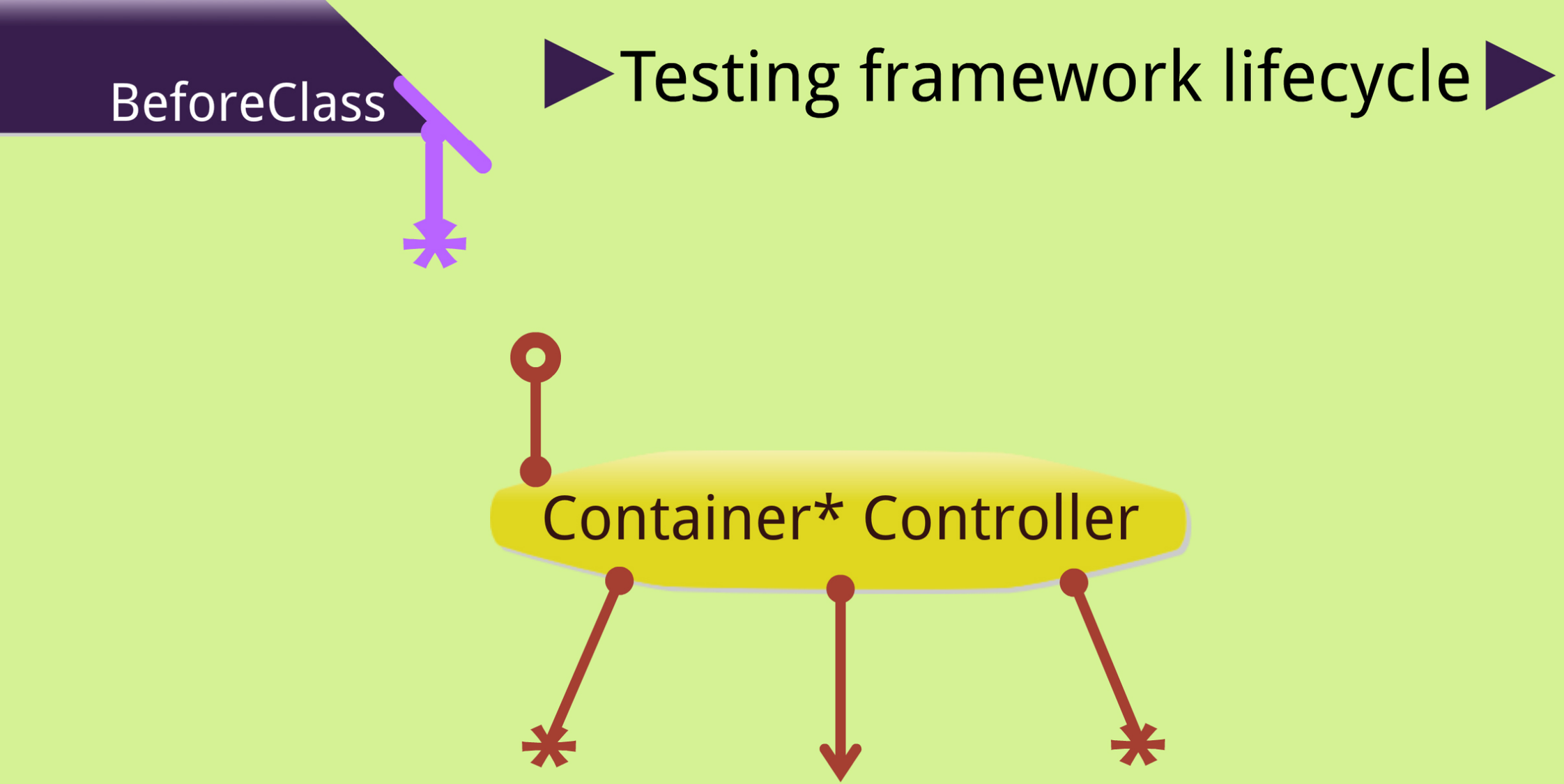
Testing framework lifecycle



Testing framework lifecycle

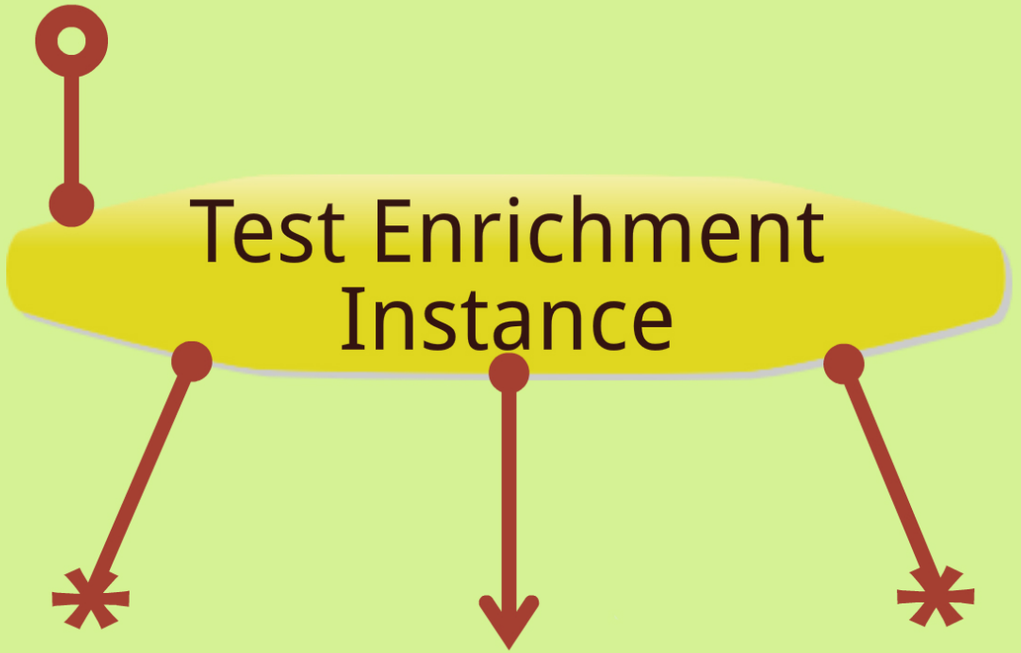


Testing framework lifecycle



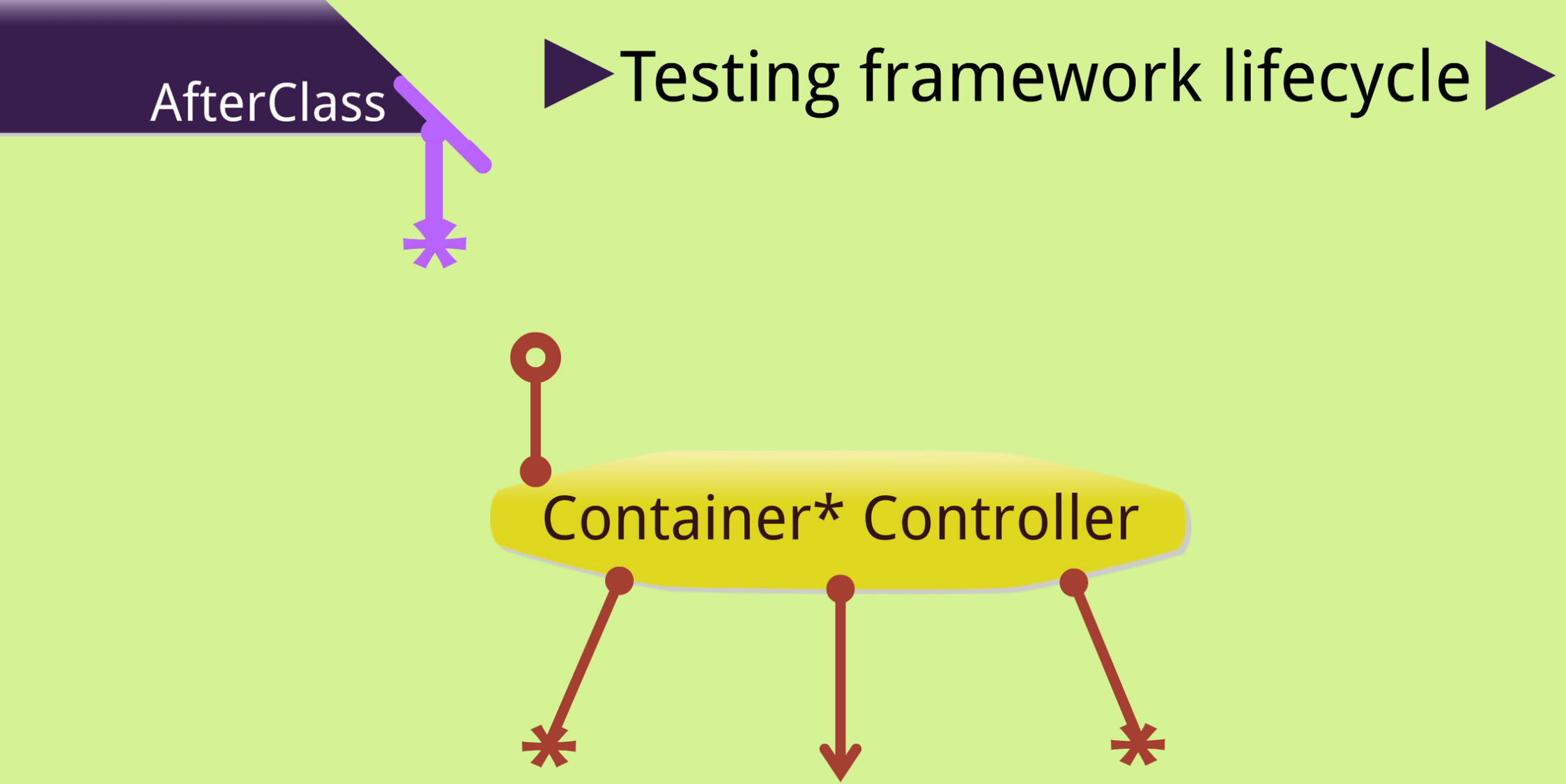
Testing framework lifecycle

Before

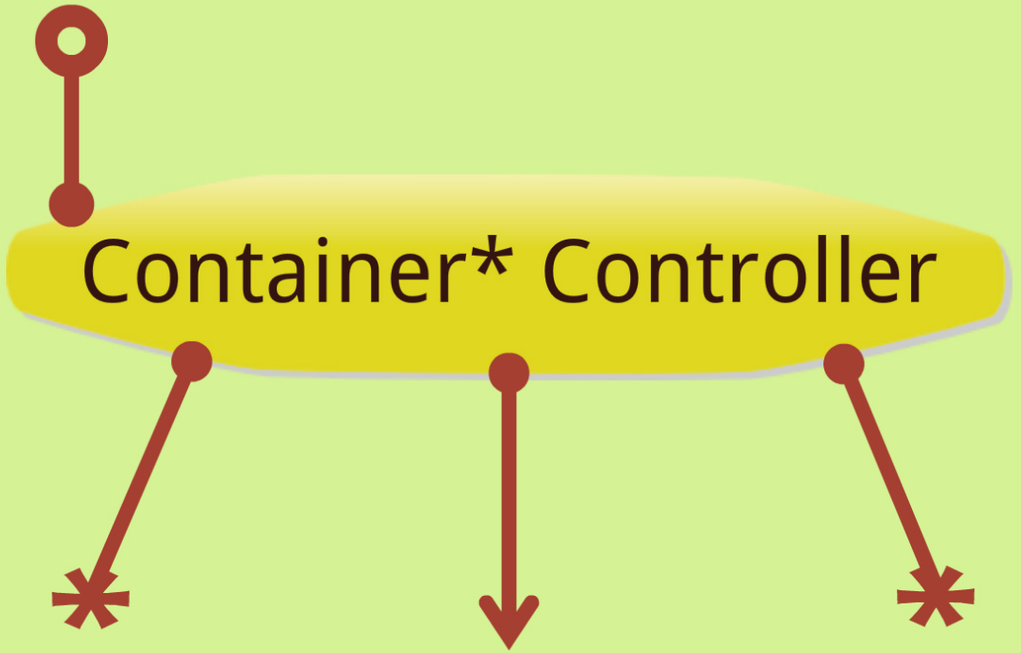


Testing framework lifecycle

Testing framework lifecycle



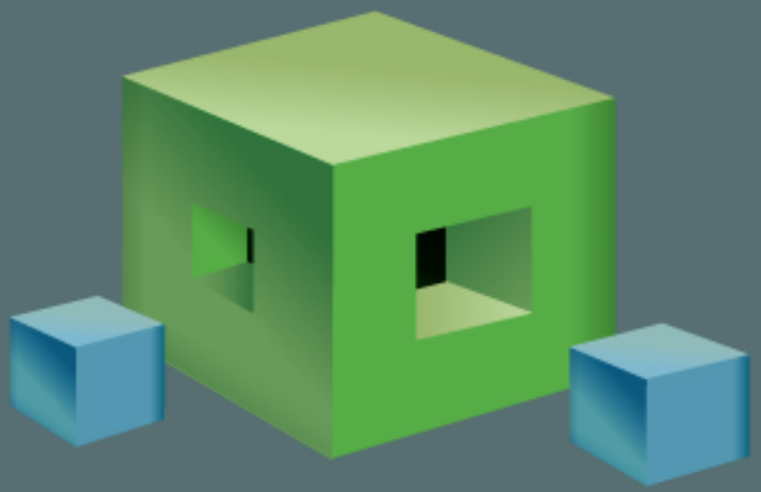
AfterSuite



Test Enricher

```
public class DataLayerTest {

    @Inject
    EntityManager em;

    @Inject
    UserTransaction tx;

    @Test
    public void should_...
```

Deployable Container

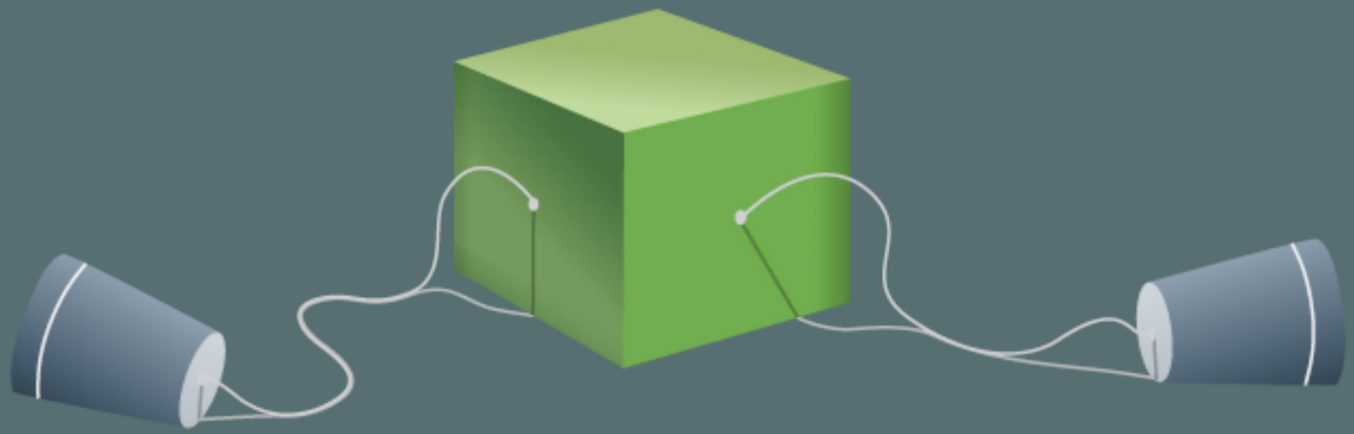


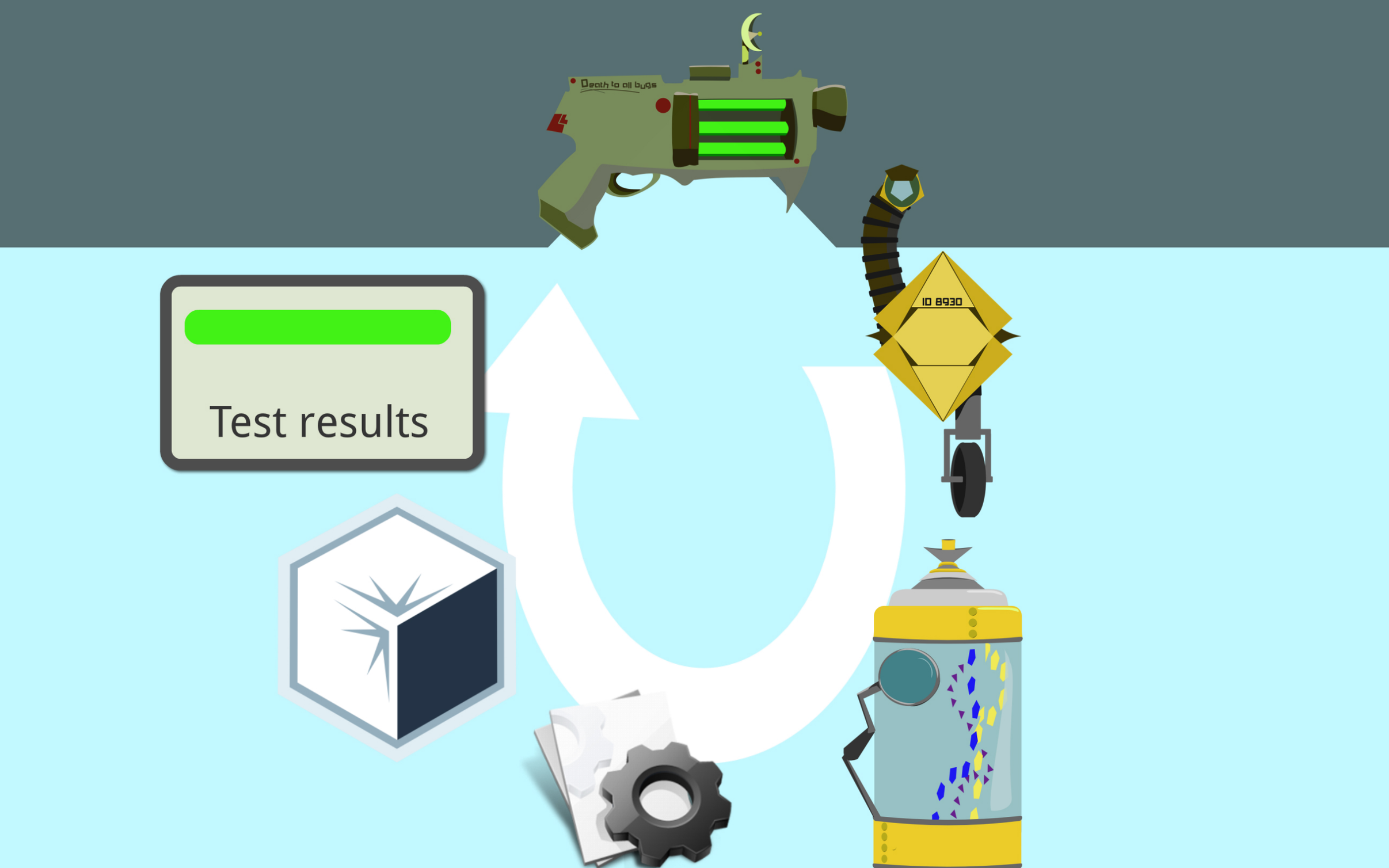
Auxiliary Archive Appender

Archive Processor

Deployment Generator

Protocol

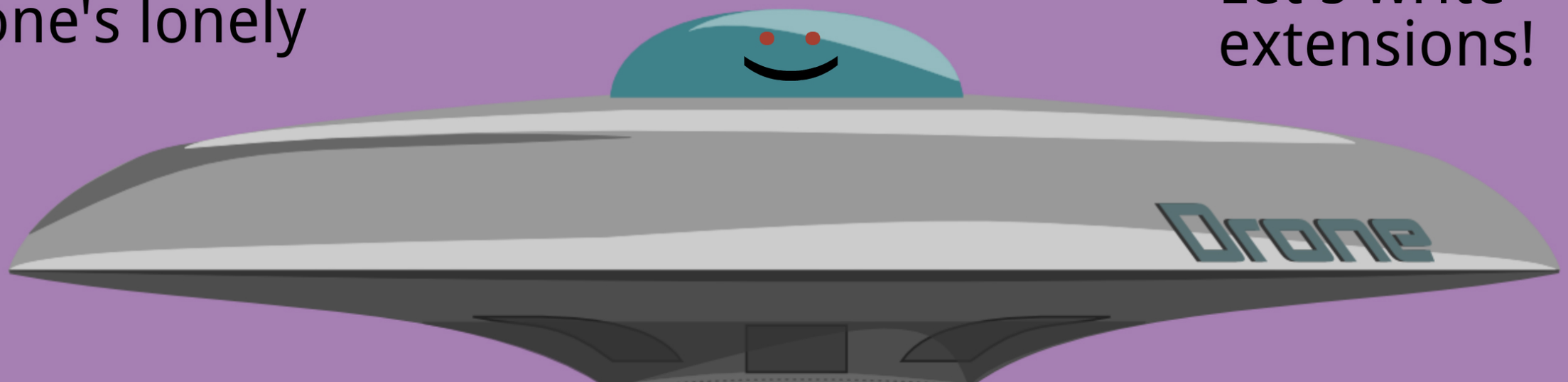



Death to all bugs
ID 8930
Test results

Extension case studies

Drone's lonely

Let's write extensions!



Ways to extend Arquillian

write container adapter

bind to life cycle event

append to deployment

manage secondary life cycle (such as Selenium Server)

modify or instrument deployment

enrich test

Thank you!

arquillian.org

github.com/arquillian