

JUNIT 5 EXTENSIONS



MARC PHILIPP

*Software Engineer @ LogMeIn
in Karlsruhe, Germany*

JUnit Maintainer since 2012

Twitter: @marcphilipp

Web: marcphilipp.de





JUNIT 5 IS RELEASED!

Release date: *September 10, 2017*



Old Test

New Test

Another Test

JUNIT 5

VINTAGE

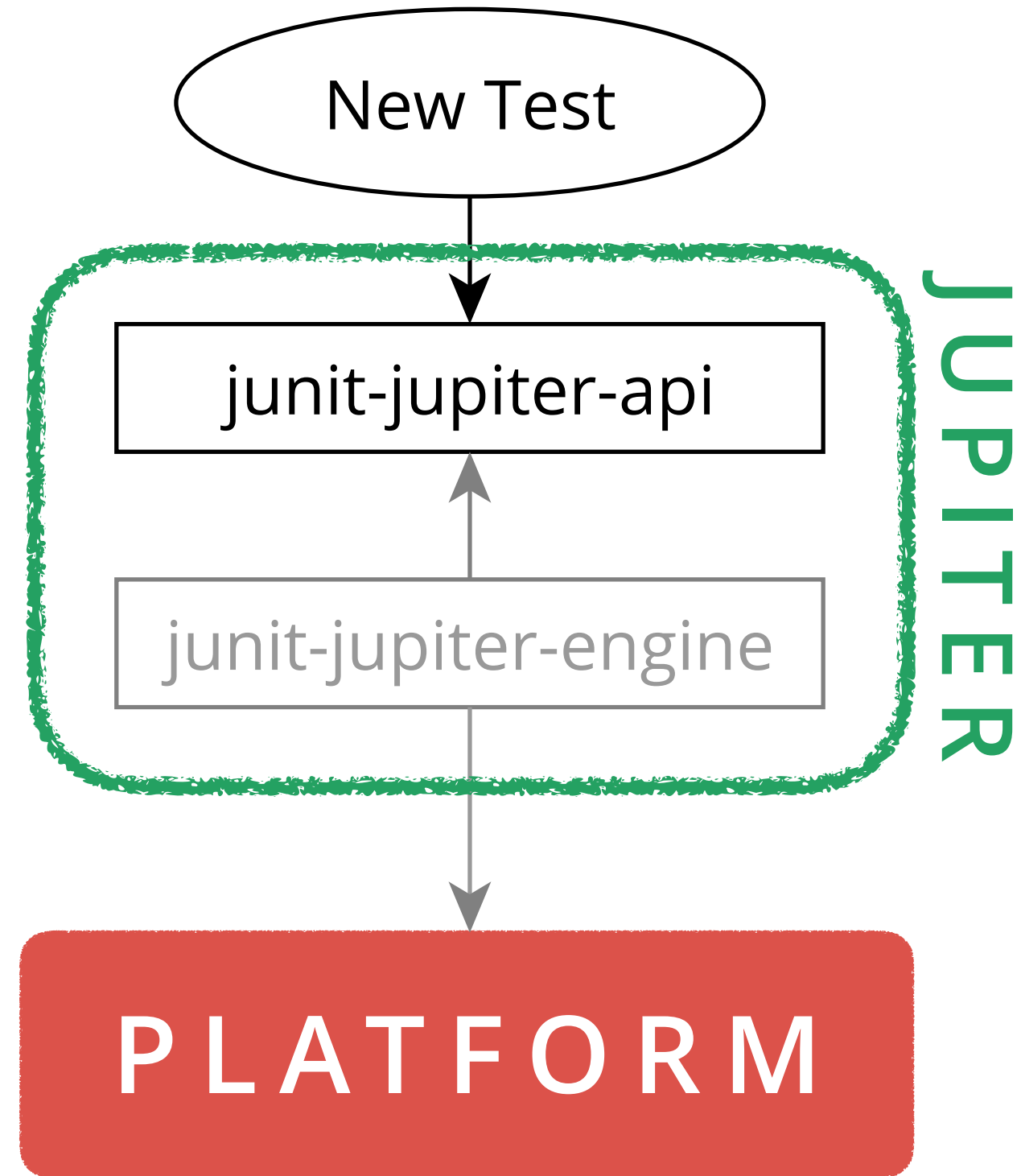
JUPITER

THIRD
PARTY

PLATFORM

IDEs/Build Tools

JUNIT JUPITER



JUNIT JUPITER API

- programming model for test authors
- extension model for extension authors



PROGRAMMING MODEL

```
import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class SimpleTest {
    @Test
    @DisplayName("1 + 1 = 2")
    @ExtendWith(MyCustomExtension.class)
    void onePlusOneEqualsTwo() {
        assertEquals(2, 1 + 1);
    }
}
```

EXTENSION MODEL

```
import org.junit.jupiter.api.extension.*;

class MyCustomExtension
    implements BeforeEachCallback, AfterEachCallback {
    @Override
    public void beforeEach(ExtensionContext context) {
        // setup
    }
    @Override
    public void afterEach(ExtensionContext context) {
        // teardown
    }
}
```


EXTENSION REGISTRATION

- *Declarative*: Annotate your test classes or methods with `@ExtendWith` to register extensions
 - May be used as a meta-annotation
- *Global*: Opt-in support for registration via `ServiceLoader`

CONDITIONAL TEST EXECUTION



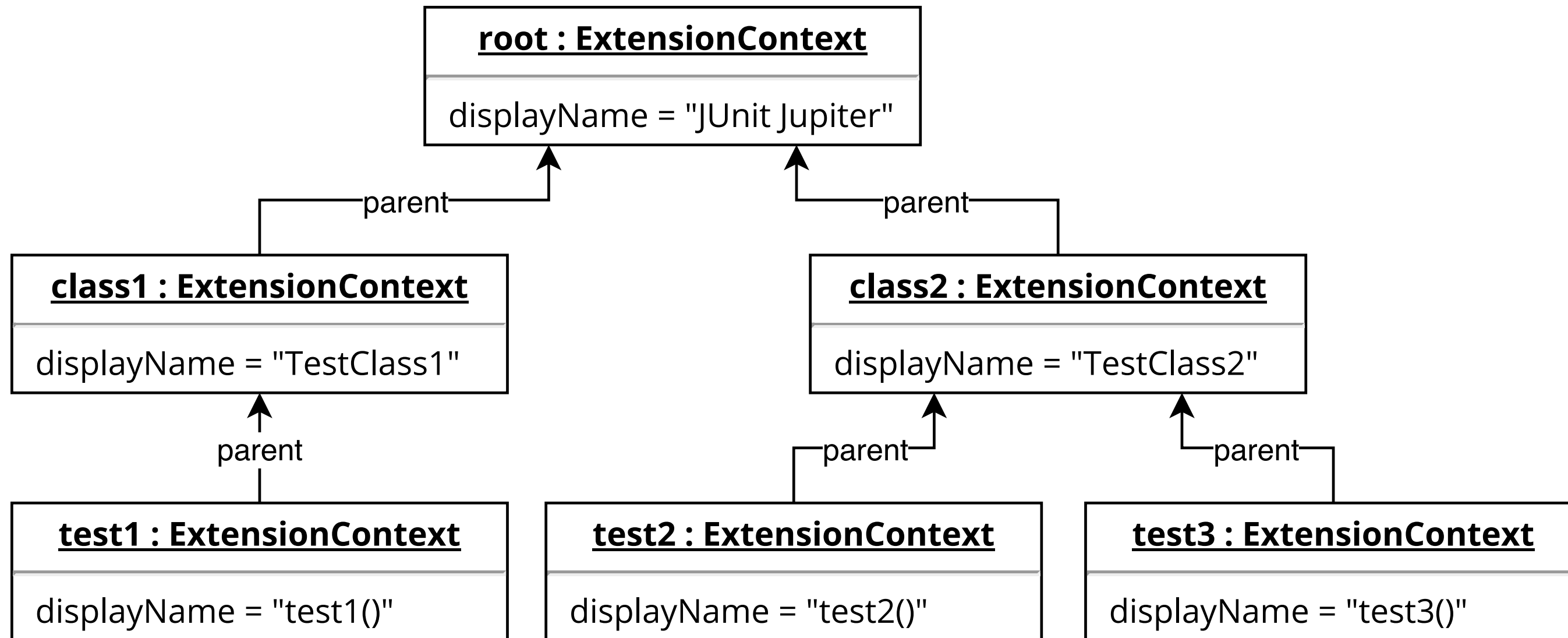
DEMO



EXTENSION CONTEXT

- Represents the current node in the test tree, e.g. test method or class
- Provides access to meta information about such a node, e.g. display name, method, class

EXTENSION CONTEXT



LESSONS LEARNED

- How to implement custom logic to determine whether a test class/method should be skipped?
- How to use *global* extension registration?
- How to deactivate a condition without changing the code using system properties?



REUSABLE TEST SETUP & TEARDOWN



DEMO



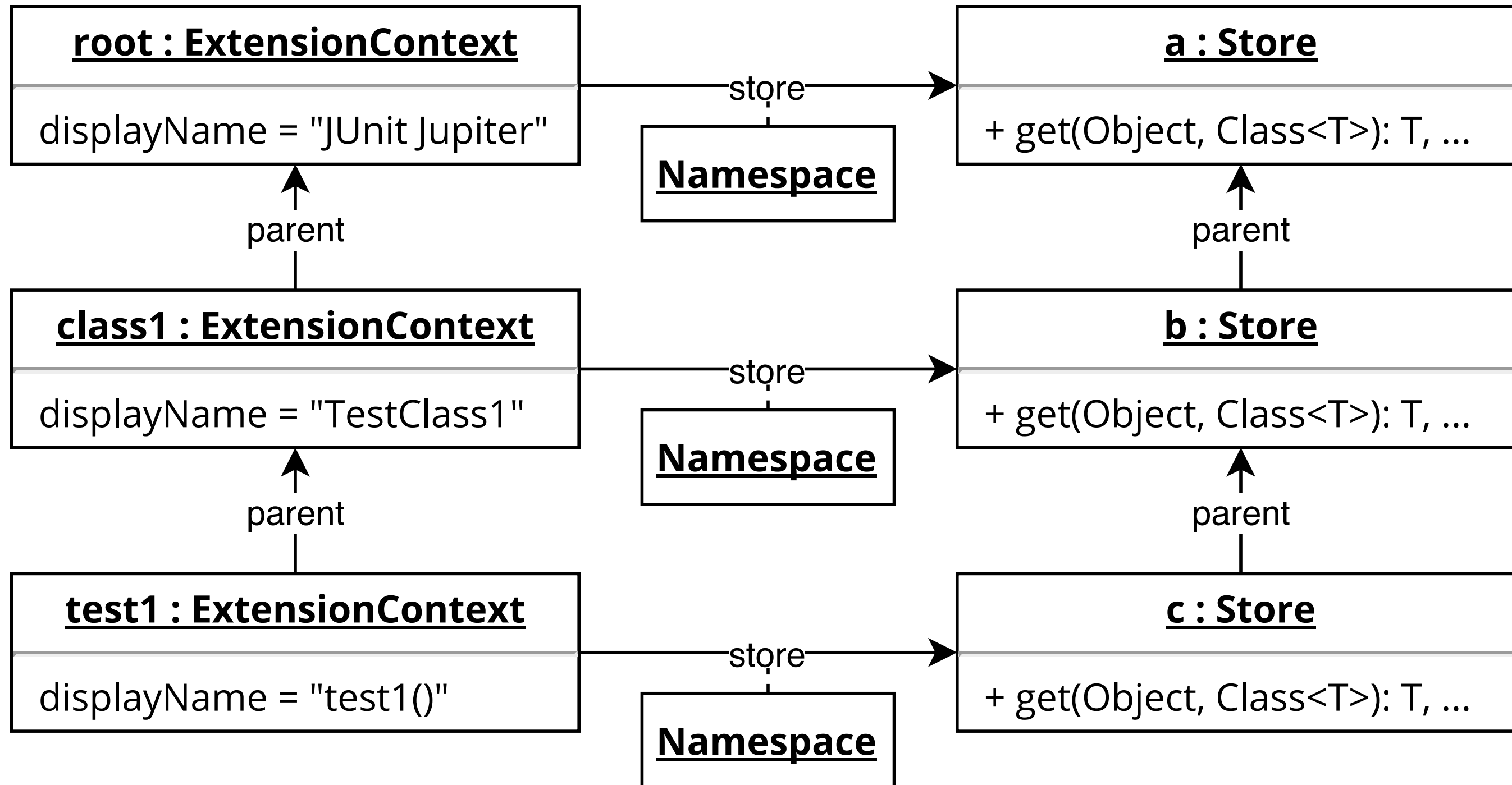
WHY THE STORE ABSTRACTION?

- Methods within an extension need to save and retrieve data, e.g. to cleanup in the end
- `Extensions` are instantiated once and called for multiple tests
- Using instance variables would be error-prone

STORE

- `Map`-like interface for extensions to save and retrieve data, e.g. `store.put(key, value)`
- Accessed via a `Namespace` : Enables sharing data across extensions, but makes it a deliberate decision (e.g. `Namespace.GLOBAL`)
- Reading from a `Store` follows the hierarchy upwards, if a key is not found

STORE



LESSONS LEARNED

- Using the `Store` class for extension state
- Using Lifecycle Callbacks to enable reuse of common setup/teardown code
- Implementing multiple `Extension` interfaces in a single extension



RESOLVING TEST PARAMETERS



DEMO



LESSONS LEARNED

- How to resolve test method parameters in an `Extension` ?
- You can also inject parameters into test class constructors and `@BeforeEach` / `@AfterEach` methods

PROVIDING ARGUMENTS FOR PARAMETERIZED TESTS



DEMO



LESSONS LEARNED

- How to use Parameterized Tests?
- Writing a custom `ArgumentsProvider` that loads data from a JSON file, ...

FROM TESTS TO TEST TEMPLATES



DEMO



LESSONS LEARNED

- How to execute a test multiple times with different contexts?
- How to implement a `TestInvocationContextProvider`

SUPPORT CLASSES

Package `org.junit.platform.commons.support` contains:

- `AnnotationSupport` to scan for custom annotations, including meta-annotations
- `ReflectionSupport` for classpath scanning, finding methods, invoking them etc.

SUMMARY



JUNIT JUPITER IS EXTENSIBLE

- A lot of extension points to choose from
- The JUnit team will add more in future releases
- Combining multiple extension points in one extension is very powerful!



GETTING STARTED

- User Guide:
<http://junit.org/junit5/docs/current/user-guide/>
- Sample projects for Gradle and Maven:
<https://github.com/junit-team/junit5-samples>
- Javadoc:
<http://junit.org/junit5/docs/current/api/>



WANTED: FEEDBACK!

- StackOverflow:
<http://stackoverflow.com/questions/tagged/junit5>
- Code & Issues:
<https://github.com/junit-team/junit5/>
- Twitter:
<https://twitter.com/junitteam>



THANK YOU!

Example code:

<https://github.com/marcphilipp/junit5-extensions-demo>

