



TRANSITION
TECHNOLOGIES

JUnit 5

Zbyszko NATKAŃSKI

Agenda

- **Why JUnit5 was introduced**
- **Who is responsible for JUnit 5**
- **Current stable version, IDE & Build tools support**
- **Briefly about JUnit 5 key features:**
 - **Modularity**
 - **Launchers and test engines**
 - **Architecture**
- **Demo & Live coding**
- **Summary**
- **Q&A Session**

Hands on

- **Testing**
- **Unit testing**
- **Junit 4**

Why JUnit 5 was introduced

- JUnit 4 was released more than 10 years ago
- Testing expectations and requirements have grown
- Modularity, everything in one file jar (*junit-4.12.jar*)
- Test discovery and execution (*tightly coupled*)
- Extensibility
- Java 8 and 9 have appeared

JUnit 5

JUnit 5 and crowdfunding campaign

- Ran from July 2015 to October 2015
- Raised about 54,000 euro from 474 companies
- 4 companies donated 6 weeks of development time
- Lead by Pivotal (Spring developers)



Shortly about JUnit 5

- **Working title:** JUnit Lambda
- **Version 5.1.0 was released on February 2018 (but 5.0.1 is stable!)**
- **JUnit 5 key features are:**
 - **Modularity**
 - **Extensibility**
 - **Modern (uses Java 8 features)**
 - ***Forward and backward compatibility***
 - Supports running JUnit 3.8, JUnit 4 and JUnit 5
- **Runs on Java 8 and 9**
 - Working fine with Java 10(?) or whatever the name will be
- **IDE support**
 - **IntelliJ IDEA:** 2016.2+
 - **Eclipse:** Oxygen (4.7.1a)
 - **Netbeans** ☹
- **Build tools**
 - **Maven**
 - **Gradle**

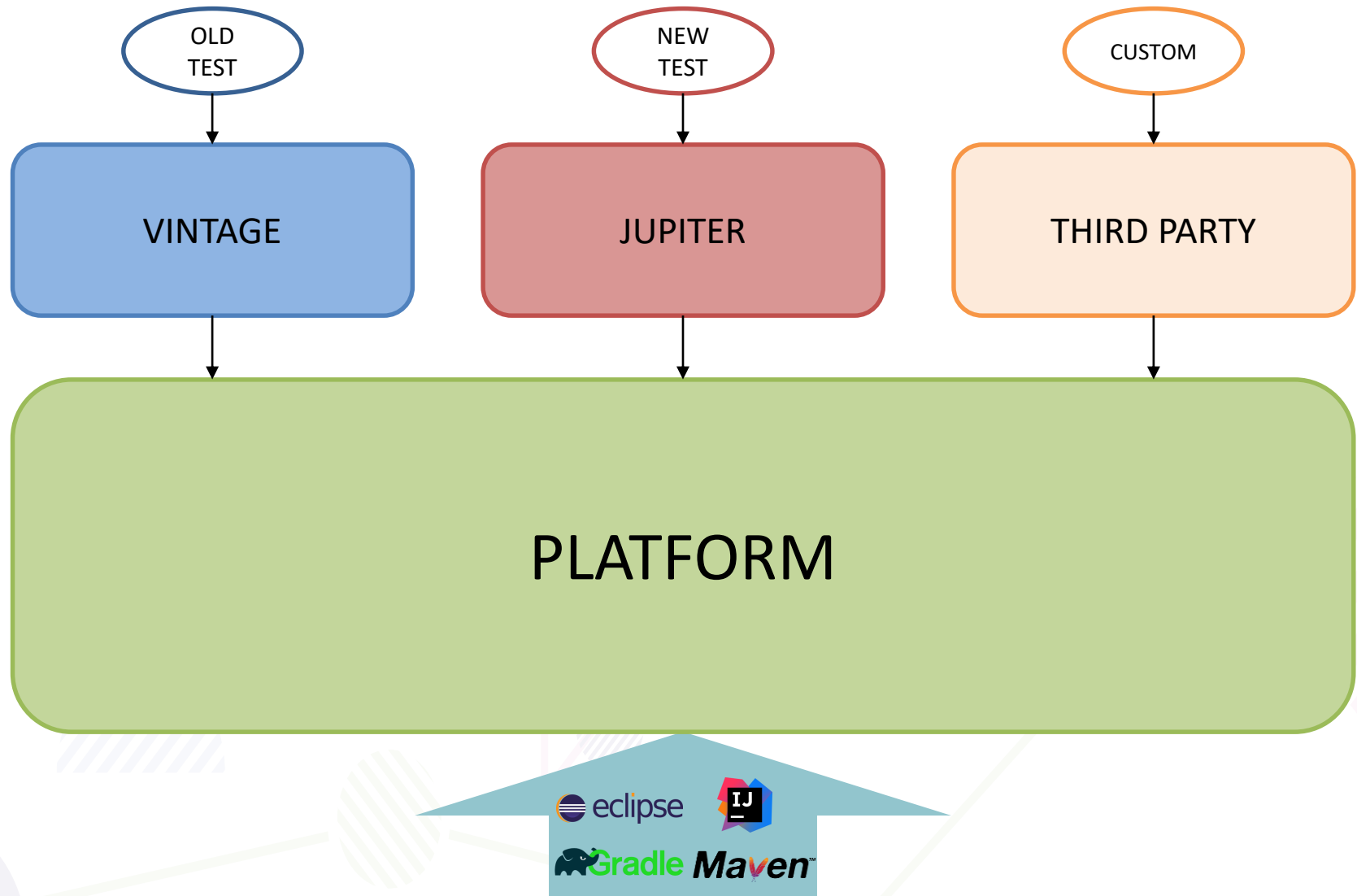
JUnit 5 Modularity

- JUnit 5 is not a library any more...
- Now it is a **framework!**
- **JUNIT 5 = PLATFORM + JUPITER + VINTAGE**
- **JUnit Platform 1.0.0**
 - Foundation for launching testing frameworks on the JVM
 - Launcher and TestEngine APIs
 - *ConsoleLauncher*, Gradle plugin, Maven Surefire provider
- **JUnit Jupiter 5.0.0**
 - New programming model and extension API for JUnit 5
- **JUnit Vintage 4.12.0**
 - TestEngines for running JUnit 3 and 4

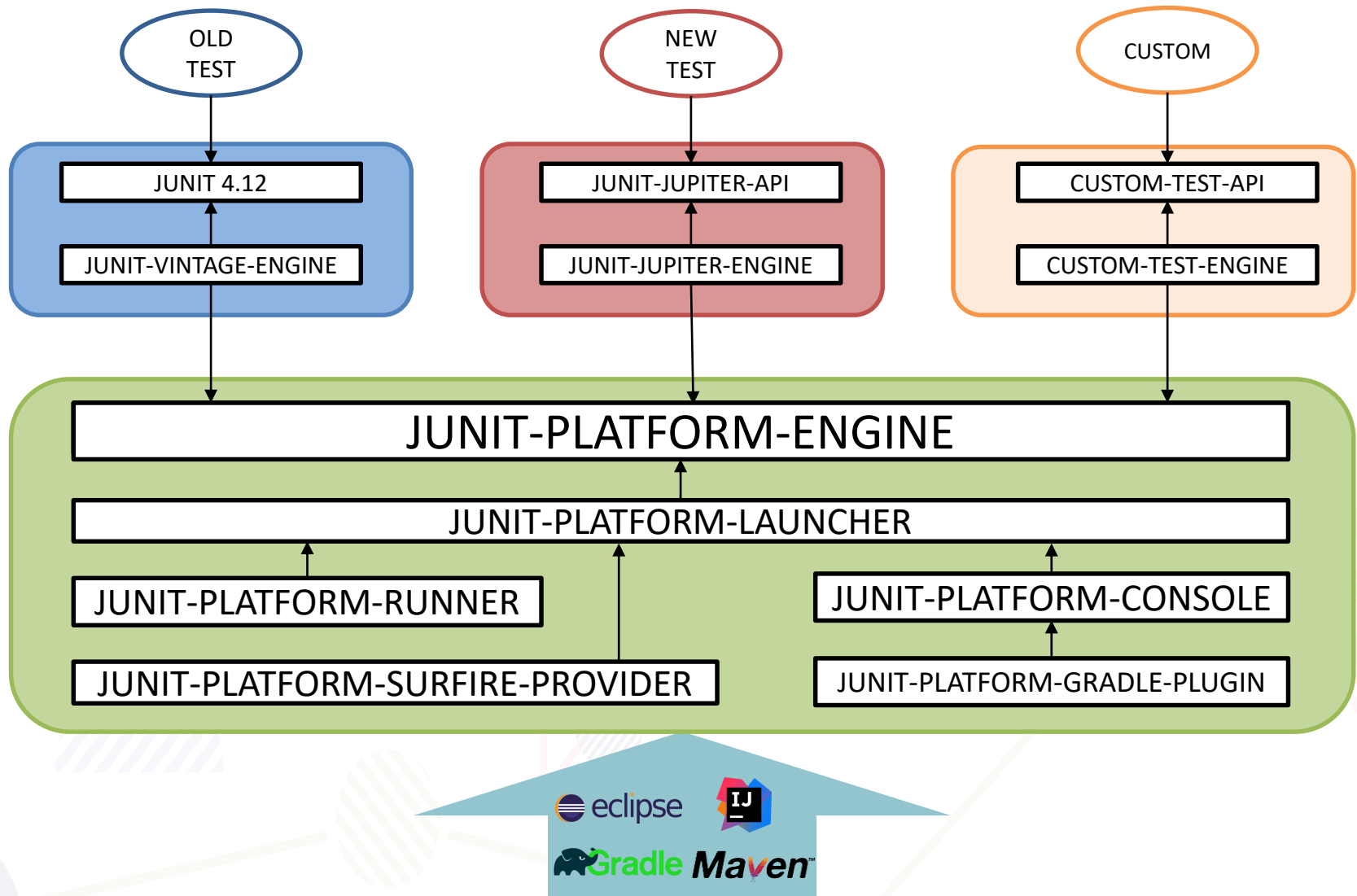
JUnit 5 Launchers and test engines

- **Launchers**
 - Used by IDEs and build tools to launch the framework
 - ConsoleLauncher for command line support
- **Test Engines**
 - Used to discover tests
 - Define the way how the test will be executed
 - JupiterTestEngine
 - VintageTestEngine
 - *Write your own implementation*

JUnit 5 Architecture



JUnit 5 architecture in details



Let's start JUnit 5

- **JUnit 5 is open source:**
 - Github repository: <https://github.com/junit-team/junit5/>
- **User guide:**
 - <https://junit.org/junit5/docs/current/user-guide/>
- **JUnit 5 samples repository:**
 - <https://github.com/junit-team/junit5-samples>
- **The repository hosts a collection of sample projects based on JUnit Jupiter and JUnit Vintage:**
 - For Gradle, check out the *junit5-gradle-consumer* project
 - For Maven, check out the *junit5-maven-consumer* project
- **Javadoc:**
 - <https://junit.org/junit5/docs/current/api>

Jupiter API

- **New annotations API**
 - @BeforeAll
 - @BeforeEach
 - @Test
 - @AfterEach
 - @AfterAll
- **New assertions**
 - assertThrows(...)
 - assertAll(...)
 - assertTimeout(...)
- **Assumptions**
- **@DisplayName**
- **@Nested**
- **Dynamic tests**
 - @TestFactory
- **On/Off**
 - @Disabled
 - @EnableOnJRE
 - @Disabled
 - Custom...
- **@RepeatedTest**
- **Test method parameters**
 - TestInfo
- **@ParametrizedTest**
 - @ValueSource
 - @CsvSource
 - @MethodSource
 - @CsvFileSource
 - @EnumSource
 - @ArgumentSource

Extensions API

- **Declaratively via @ExtendWith**
- **Test instance post processing**
 - TestInstancePostProcessor
- **Conditional test execution**
 - ExecutionCondition
- **Parameters resolution**
 - ParameterResolver
- **Callbacks**
 - BeforeAllCallback
 - BeforeEachCallback
 - BeforeTestExecutionCallback
 - AfterTestExecutionCallback
 - AfterEachCallback
 - AfterAllCallback
- **Exception handling**
 - TestExcecutionExceptionHandler

Summary

- Easy in use and understandable
- Many key features have finally appeared
- Some features are experimental and could be deprecated in future
- A little bit slower than JUnit 4, but they are working on that
- Extensions empower!

- Code samples from presentation itself
 - <https://github.com/znpSC/junit5-playground>
- Questions?
- Thank you!