

Microsoft.Testing.Platform

Jakub Jareš





01

Microsoft.Testing.Platform (MTP)

The image features a dark blue background with decorative geometric shapes in the corners. In the top right corner, there is a large, stylized 'L' shape composed of two overlapping rectangles, one in a lighter blue and the other in a darker blue. In the bottom left corner, there is a smaller, similar 'L' shape, also composed of two overlapping rectangles in the same color scheme.

**Lightweight, modern alternative to VSTest.
Refreshing how tests are run in .NET.**

Microsoft.Testing.Platform (MTP)


Introducing MSTest Runner



Jan 24, 2024

25 9

Introducing the MSTest Runner – CLI, Visual Studio, & More

 Amaury, Marco, Jakub


MSTest runner is a new, light-weight and portable runner for MSTest tests available in the .NET CLI, Visual Studio, and more!



Feb 10, 2025

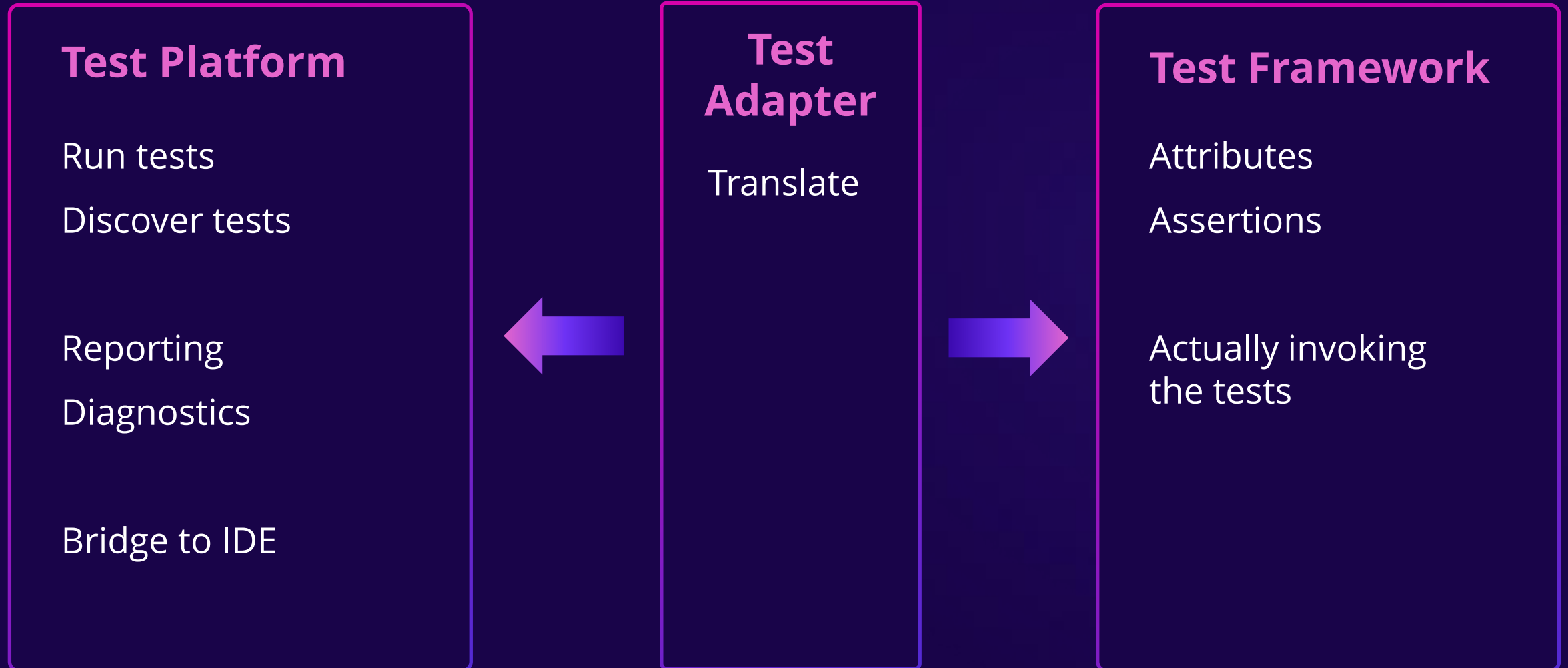
12 8

Microsoft.Testing.Platform: Now Supported by All Major .NET Test Frameworks

 Amaury Levé

All major .NET testing frameworks are now supporting Microsoft.Testing.Platform. Whether you are using Expecto, MSTest, NUnit, TUnit, or xUnit.net, you can now leverage the...

Basic concepts



ğăğêşřăçe Şğîlîng Têşş

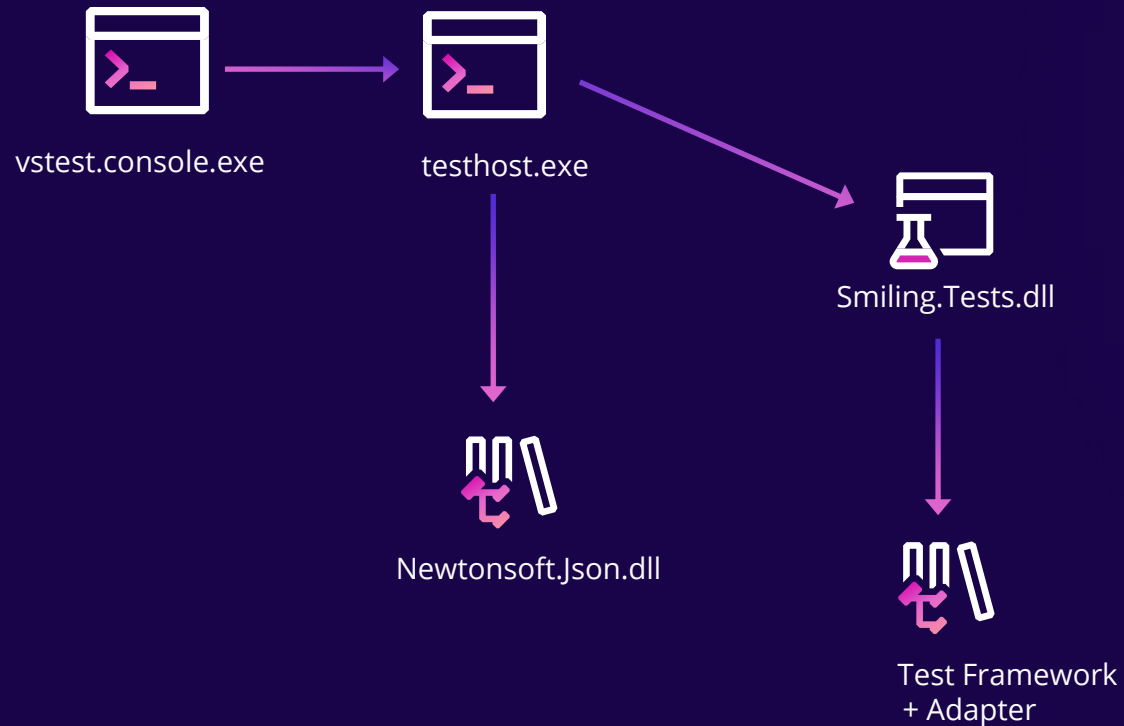
TêşşClăşş
řučlîç çlăşş ÉñokîTêşş

TêşşÑêşhòđ
řučlîç wôîđ ŞğîlêşNộşGsộxηş

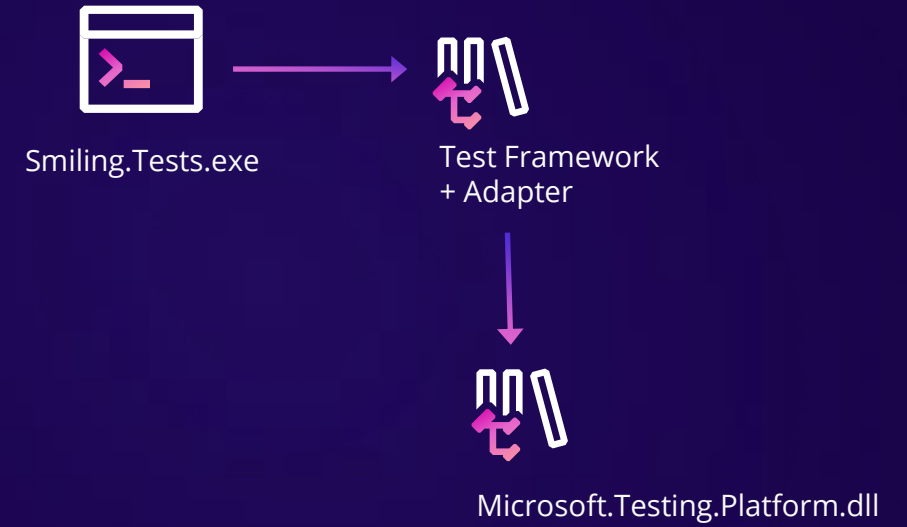
Aşşêşş AsêÉřụắl 😊 😞

Architectures

VSTest



MTP





02

Running tests with Microsoft.Testing.Platform

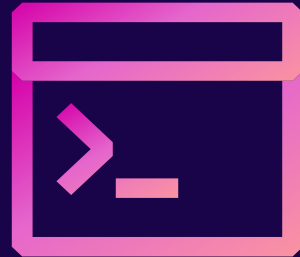


Demo: Running our first test



Demo summary

Microsoft.Testing.Platform test projects are built as executables.



HelloWorldTests.exe

03

Extensions and hot reload



Demo: Using extensions



Demo summary

We have cool extensions.

- Hot reload
- Retry
- Code coverage
- Crash dump
- Hang dump
- Fakes
- Output with progress
- TRX

<https://aka.ms/testingplatform/extensions>

04

Running with xUnit and NUnit



Demo: Running with XUnit and NUnit

Demo summary

**XUnit and NUnit both support
Testing.Platform.**

- Both require recent versions of packages to be used.

05

new dotnet test



Demo:
Running with new dotnet test

Demo summary

Testing.Platform enabled projects are still usable in Visual Studio, and dotnet test.

- MSTest and NUnit are keeping backwards compatibility.
- You can enable Testing.Platform today, and reap the benefits, while keeping your workflow.



05

**Why consider using the
new platform?**



Deterministic

Future proof

Performant

Portable

Extensible

Secure

Deterministic

Running the same tests, with the same configuration produces the same results on your machine and on build server.



My tests are crashing, because they pick up wrong version of TestAdapter dll from NuGet cache.



Static extension registration.

No reflection lookup, or folder scanning.

No custom assembly resolver.

Portable

Tests are portable, and detached from their surrounding environment. Moving tests from one computer to another is easy.



My test are failing on server, but not locally. I cannot debug the problem.



No folder scanning of VisualStudio or NuGet cache.

No sharing of adapters between projects.

Future proof

Test projects are not special, they can be manipulated and handled as any other dotnet application.

New form factors and architectures should require minimal effort to adopt.



Running with NativeAOT is not possible with `dotnet test`.

`dotnet publish` does not work with tests.



Whole platform compatible with NativeAOT.

MSTest added NativeAOT support in preview.

Test projects are simple .exe executables what you can do with console, you can do with tests.

Extensible

The platform is built around extensibility.

Framework authors, or service providers should have easy time extending the platform.



My tests are flaky, how do I retry them?

VSTest relies on Newtonsoft.Json, how do I test Newtonsoft.Json?



We added extension for retrying tests, and other extensions.

The core platform has no dependencies. You can test any dll.

Performant

Overhead of testing platform is minimal. All computing power and memory is spent running your tests.



Test running is slow, vstest.console takes too much memory.



New platform reduced number of started process and serialization.

Teams see 30% increase in speed just by migrating to Testing.Platform.

Secure

Whole platform is shipped as nuget package, with minimal dependencies.



Dependencies of vstest.console are marked as vulnerable in our code compliance checks.



Whole platform is shipped as nugets, we have minimal dependencies.

Security fixes can be shipped next day.

06

Summary

**We would love your feedback,
ideas, improvements, and
issues:**

github.com/microsoft/testfx