CloudGen

TOPIC

.NET Interactive for your code and Azure

Marco Parenzan

#CodeGen2021

@cloudgen_verona

Marco Parenzan

- Solution Sales Specialist in Insight for Digital Innovation
- Azure MVP
- Community Lead 1nn0va // Pordenone
  - https://datasaturdays.com/events/datasaturday0001.html
  - 1nn0va After Hour
  - https://bit.ly/1nn0va-video
- Linkedin: https://www.linkedin.com/in/marcoparenzan/

Marco Parenzan

Marco Parenzan

marco_parenzan

marcoparenzan

marcoparenzan

CloudGen

TOPIC

.NET Interactive for your code and Azure

Marco Parenzan

• (Data) Science is all about notebooks

- Way of the Data Scientist
  - Sketching
  - Trial&Error
  - Crayons for whiteboards, Pencils & Paper for notebook
- Mathematica has a strong 30 years history in the field, with its product (Mathematica), language (Wolfram) and cloud (Wolfram Alpha)
- Mathematica has introduced the notion of notebooks
  - Annotations
  - Executable Code
- The evolution of REPL

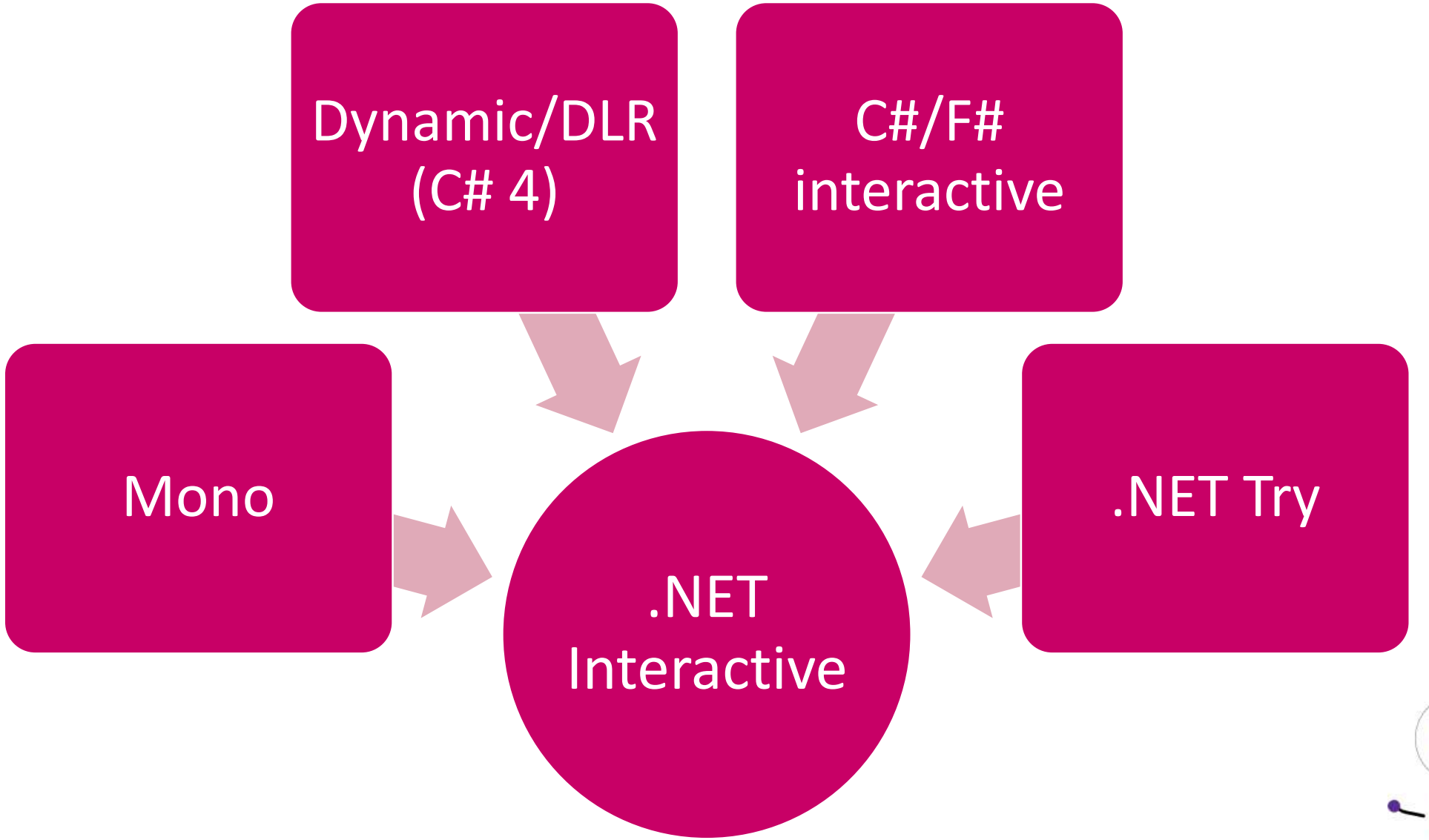
Jupyter

- Evolution and generalization of the seminal role of Mathematica
- In web standards way
  - Web (HTTP+Markdown)
  - Python adoption (ipynb)
- Written in Java
- Python has an interop bridge…not native (if ever important)

Using Notebooks

- On the web
  - Jupyter
  - Embedded in many platforms
    - DataBricks, Synapse, Azure Machine Learning
  - Binder
    - (no more https://notebooks.azure.com)
- And on premises ☺:
  - Anaconda
    - ...but you can still install by your own
  - Visual Studio Code
  - Azure Data Studio

Evolution of REPL in .NET World

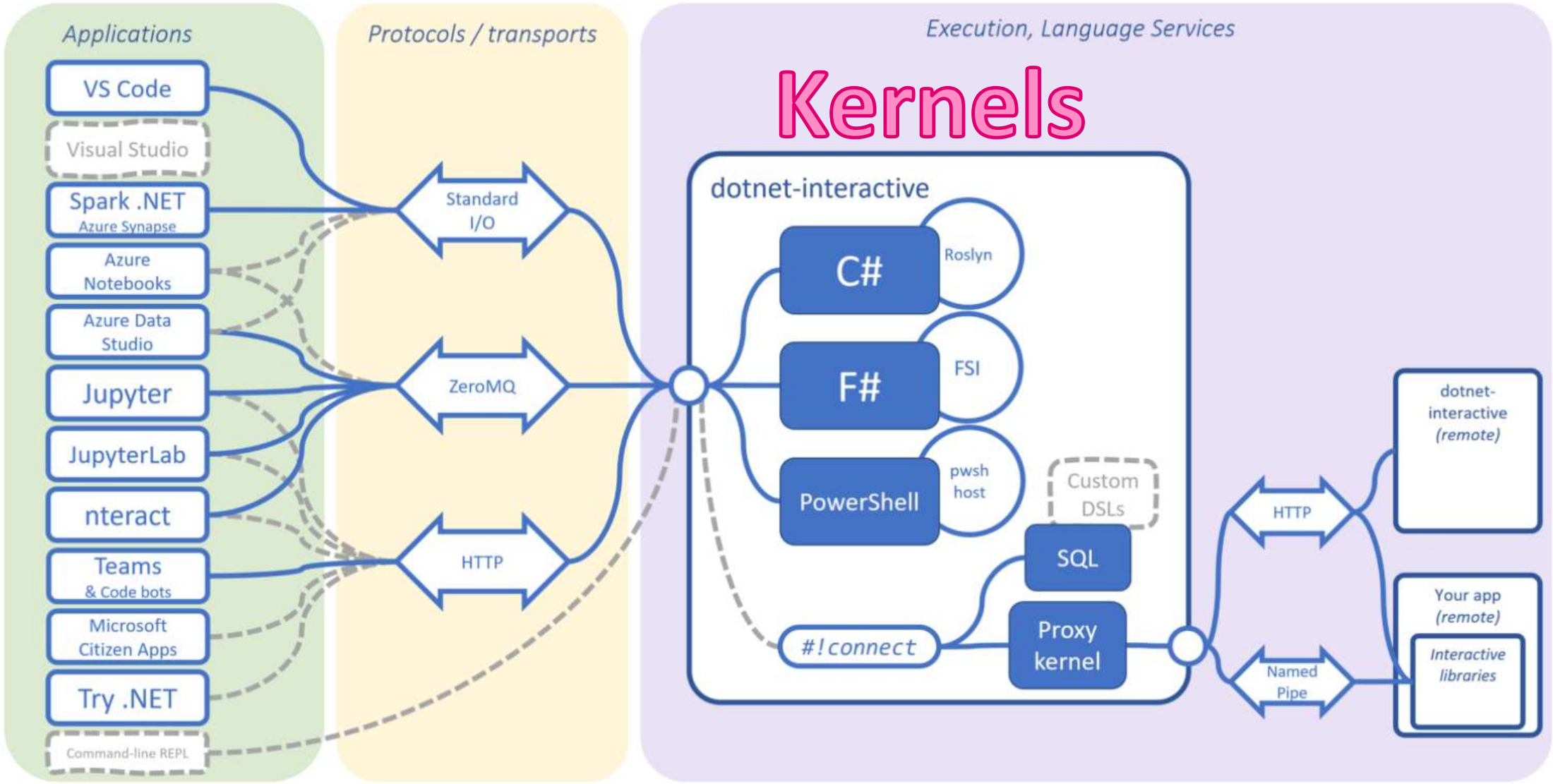


https://github.com/dotnet/interactive

.NET Interactive and Jupyter and Visual Studio Code

- .NET Interactive gives C# and F# kernels to Jupyter
- .NET Interactive gives all tools to create your hosting application independently from Jupyter
- In Visual Studio Code, you have two different notebooks (looking similar but developed in parallel by different teams)
  - .NET Interactive Notebook (by the .NET Interactive Team) that can run also Python
  - Jupyter Notebook (by the Azure Data Studio Team – probably) that can run also C# and F#
- There is a little confusion on that ☺
- .NET Interactive has a strong C#/F# Kernel…
  - …a less mature infrastructure (compared to Jupiter)

Kernel: The corner stone



How difficult is writing a kernel?


```
private async Task RunAsync(
    string code,
    CancellationToken cancellationToken = default,
    Func<Exception, bool> catchException = default)
{
    var currentDirectory = Directory.GetCurrentDirectory();
    if (_currentDirectory != currentDirectory)
    {
        _currentDirectory = currentDirectory;
        ScriptOptions = ScriptOptions.WithMetadataResolver(
            CachingMetadataResolver.Default.WithBaseDirectory(
                _currentDirectory));
    }

    if (ScriptState == null)
    {
        ScriptState = await CSharpScript.RunAsync(
                                        code,
                                        ScriptOptions,
                                        cancellationToken: cancellationToken)
                                    .UntilCancelled(cancellationToken);
    }
    else
    {
        ScriptState = await ScriptState.ContinueWithAsync(
                                        code,
                                        ScriptOptions,
                                        catchException: catchException,
                                        cancellationToken: cancellationToken)
                                    .UntilCancelled(cancellationToken);
    }

    if (ScriptState.Exception is null)
    {
        _workspace.UpdateWorkspace(ScriptState);
    }
}
```

Roslyn!

C# (just part of it)

```
using System.Threading.Tasks;
using Microsoft.DotNet.Interactive.Commands;

namespace Microsoft.DotNet.Interactive
{
    public class JavaScriptKernel :
        Kernel,
        IKernelCommandHandler<SubmitCode>
    {
        public const string DefaultKernelName = "javascript";

        public JavaScriptKernel() : base(DefaultKernelName)
        {
        }
        public Task HandleAsync(
            SubmitCode command,
            KernelInvocationContext context)
        {
            var scriptContent = new ScriptContent(command.Code);

            context.Display(scriptContent);

            return Task.CompletedTask;
        }
    }
}
```

Javascript (all of if ☺)

Agenda

Living with Notebooks

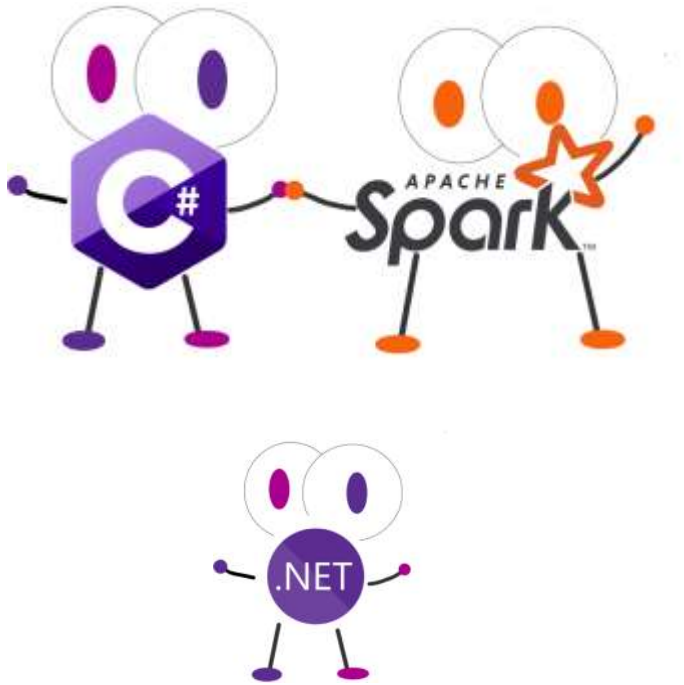
Data Science with Notebooks and .NET (and Spark)

Writing against Kernel

Living with Notebooks

- All .NET Libs and Packages
- Formatters for data
- Display info in HTML

Data Science with Notebooks and .NET (and Spark)



- .NET bindings (C# e F#) to Spark
  - Written on the Spark interop layer, designed to provide high performance bindings to multiple languages
- Re-use knowledge, skills, code you have as a .NET developer
  - Compliant with .NET Standard
- You can use .NET for Apache Spark anywhere you write .NET code
- Original project Moebius
  - https://github.com/microsoft/Mobius

Writing against Kernel

- Writing Extensions packages to embed in a nuget package format
  - custom kernels
  - Formatting
  - Magic commands
- Embedding Kernels in your custom applications

Conclusions

- Very interesting tool
- Practical for scripting and documenting
- Still in its infancy
- And .NET kernels in Data Science space has a huge work to do
  - not technical, evangelism!

Thank you

Any questions?

https://github.com/dotnet/interactive
https://github.com/marcoparenzan/DotNetInteractive


marcoparenzan


marco_parenzan


marcoparenzan