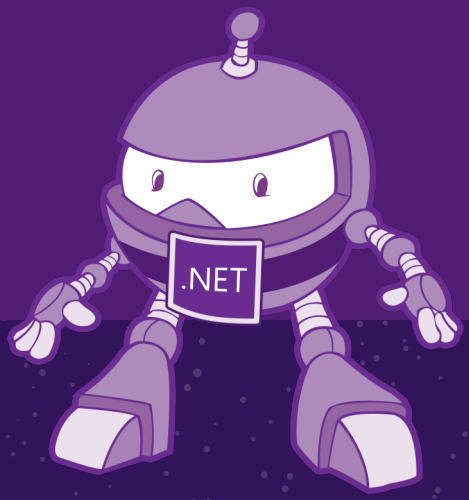
What You Need To Know About .NET Core 3.0 and Beyond

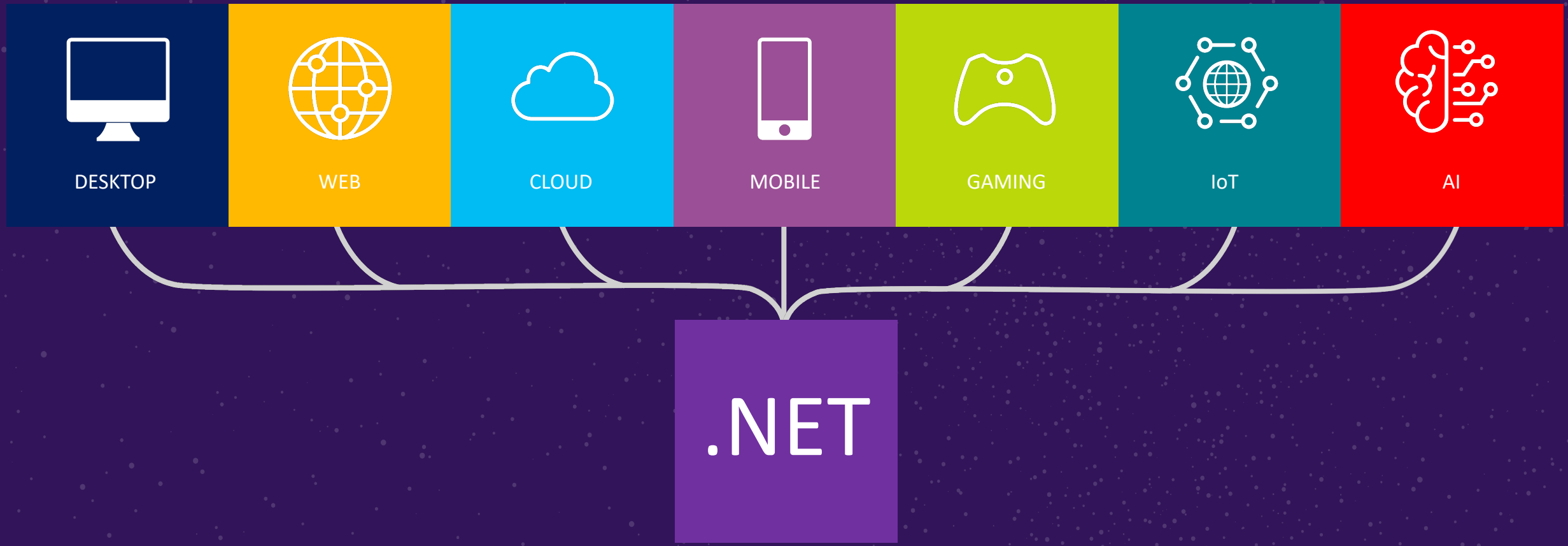
Jon Galloway | @jongalloway
Microsoft | .NET Foundation



What we'll cover

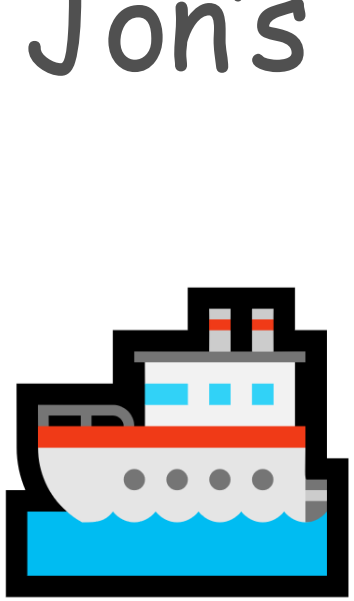
- .NET Core release overview
- Top Features in .NET Core 3.0
- When and how to update
- What's coming next

Your platform for building anything



.NET Release History

Jon's Version

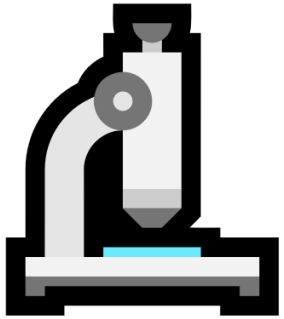


June 2016
.NET Core 1.0

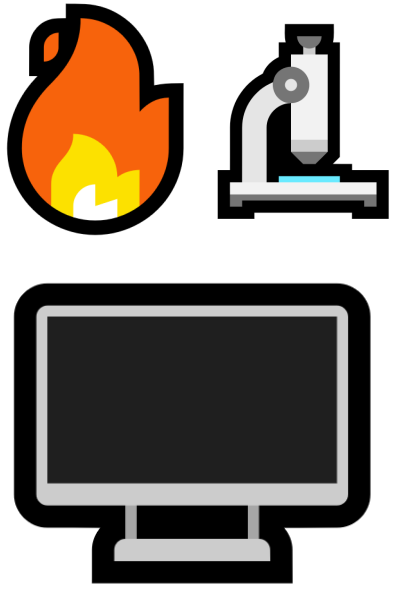
March 2017
.NET Core 1.1

Aug 2017
.NET Core 2.0

May 2018
.NET Core 2.1



Dec 2018
.NET Core 2.2



Sep 2019
.NET Core 3.0

Nov 2019
.NET Core 3.1

RELEASED

.NET Core 3.0

New C# 8.0 language features

WPF and Windows Forms support

Side-by-side support & self-contained EXEs

Microservice support with gRPC

Full-stack web development with C# and Razor

dot.net/get-core3

What’s New In C# 8

.NET

C# 8.0

Safe

Nullable and non-nullable reference types help you write safer code

Declare your intent more clearly

Modern

Async streams for modern workloads like cloud & IoT communication

Easily work with cloud scale datasets using indexes and ranges

Productive

Write less code using patterns

Protect data with readonly members

Improved using statements for resource management

Just a tiny summary!

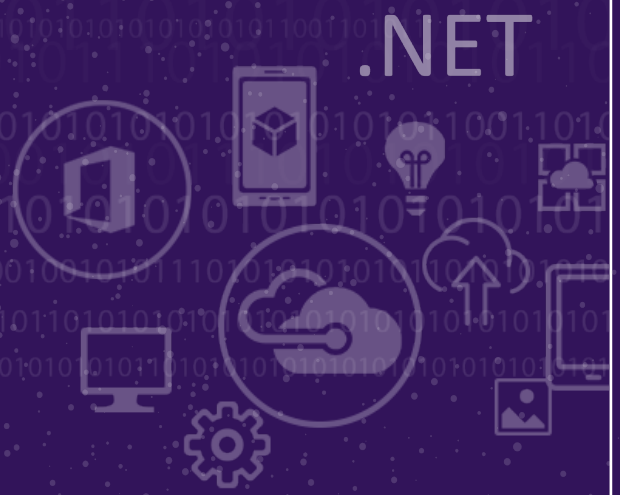
https://aka.ms/new-csharp

- Readonly members
- Default interface methods
- Pattern matching (Switch expressions, Property & Tuple patterns)
- Positional patterns
- Using declarations
- Static local functions
- Disposable ref structs
- Nullable reference types
- Asynchronous streams
- Indices and ranges
- Null-coalescing assignment
- Unmanaged constructed types
- Stackalloc in nested expressions
- Enhancement of interpolated verbatim strings

DEMO

C# 8.0

Nullable Reference Types

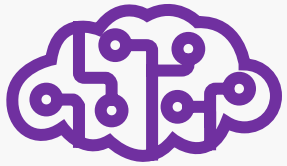


https://aka.ms/new-csharp

.NET Core 3.0 Developer Themes

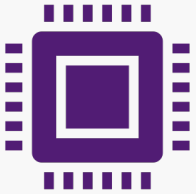
Windows Desktop Apps

Full-stack web development & Microservices



AI/ML

Big Data



IoT

Windows desktop apps

.NET

.NET Core 3.0 for Windows Desktop

Deployment Flexibility

Side-by-side deployment, self-contained EXEs

Install machine global or app local framework

Windows 10

Access modern Windows 10 APIs from WPF and WinForms

Use native Windows 10 controls via XAML islands

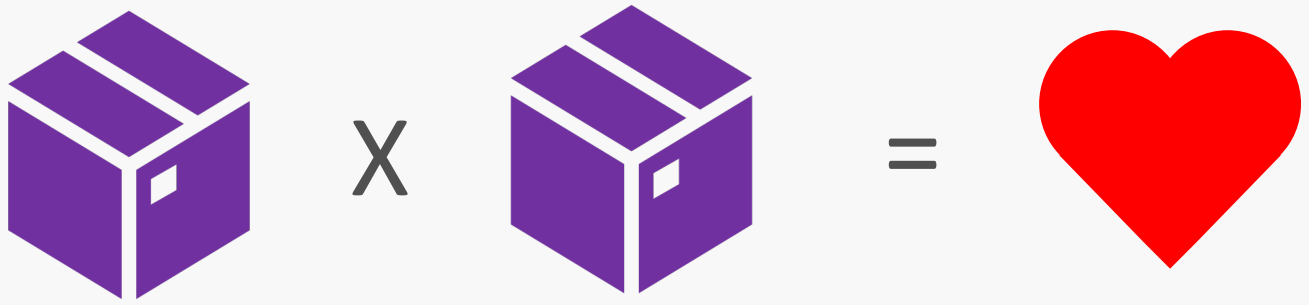
Open Source

WPF and WinForms projects also open source on GitHub

Take advantage of performance, runtime and API improvements happening in .NET Core

Why Windows Desktop on .NET Core?

- Deployment Flexibility
  - Side-by-side support
  - Machine global or app local framework
  - Self-contained EXEs
- Core runtime and API improvements
- Performance



DEMO

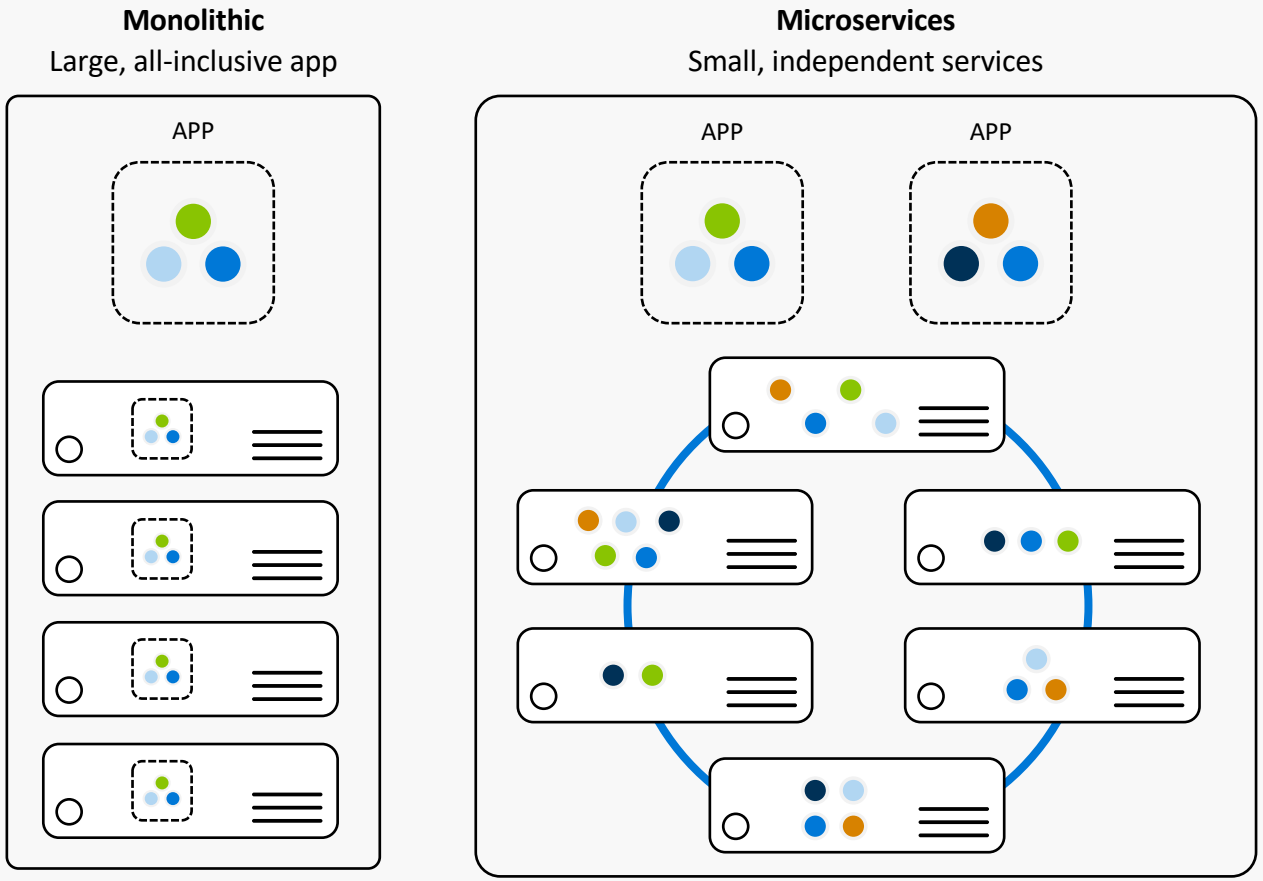
.NET Core 3.0 Windows Forms

Microservices

.NET

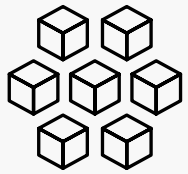
Microservices: for faster app development

- Independent deployments
- Improved scale and resource utilization per service
- Smaller, focused teams

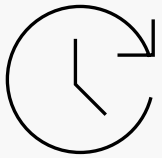


Azure Kubernetes Service (AKS)

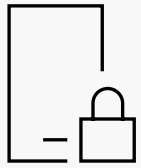
Ship faster, operate easily, and scale confidently with managed Kubernetes on Azure



Manage Kubernetes with ease



Accelerate containerized development



Build on an enterprise-grade, secure foundation

Run anything, anywhere

ASP.NET Core 3.0

gRPC

High performance contract-based RPC services with .NET

Works across many languages and platforms

Worker Service

Starting point for long running back processes like Windows Server or Linux daemon

Producing or consuming messages from a message queue

Web API's + Identity

Add security and authentication to Web API's

gRPC is…

- Popular open source RPC framework
  - Largest RPC mindshare
  - Cloud Native Computing Foundation project
  - **gRPC** stands for **g**RPC **R**emote **P**rocedure **C**alls
- Built with modern technologies
  - HTTP/2
  - Protocol Buffers
- Designed for modern apps
  - High performance
  - Platform independent

Protobuf (aka Protocol Buffers)

- IDL (interface definition language)
  - Describe once and generate interfaces for any language
- Service model
  - Service method and structure of the request and the response
- Wire format
  - Binary format for network transmission

```
5052 4920 2a20 4854 5450 2f32
0d0a 534d 0d0a 0d0a 0000 0004
0000 0000 0401 0000 0000 0000
```

Remote Procedure Calls vs HTTP APIs

Remote Procedure Calls

- Contract first (proto file)
- Contract is designed for humans
- Hides remoting complexity

HTTP APIs

- Content first (URLs, HTTP method, JSON)
- Content is designed for humans
- Emphasises HTTP

Performance
Developer productivity

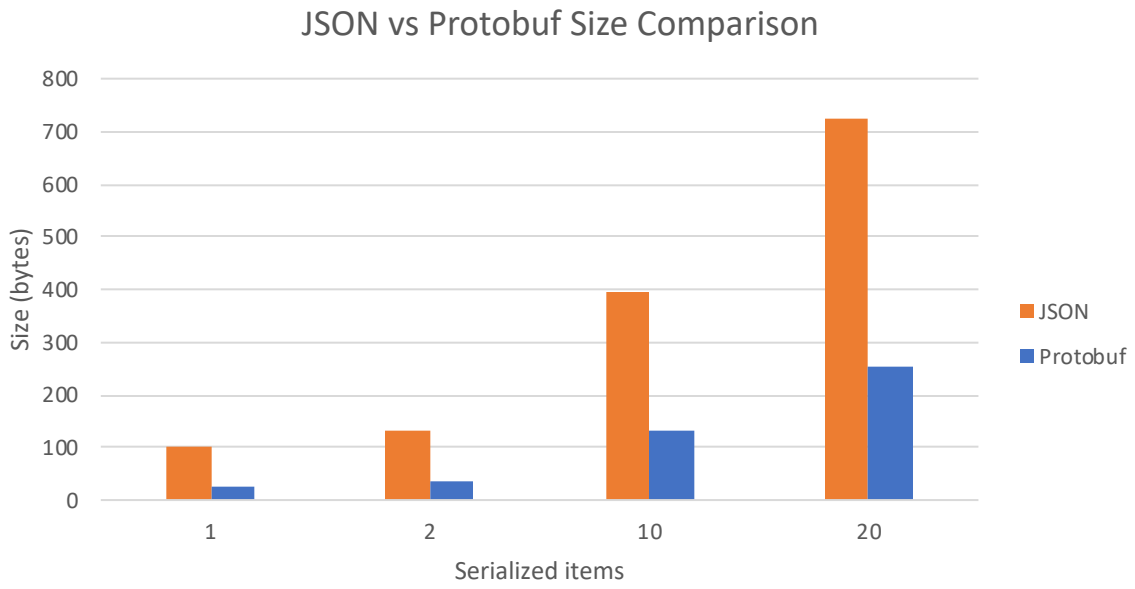
Widest audience
Ease of getting started

Key features - Performance

- Low network usage
  - HTTP/2 binary framing and header compression
  - Protobuf message serialization

```
{
  "query": "myQuery",
  "page_number": 42,
  "result_per_page": 100,
  "tickers": [
    {
      "name": "rPs",
      "value": 9.768923
    },
    {
      "name": "WEo",
      "value": 6.067048
    }
  ]
}
```

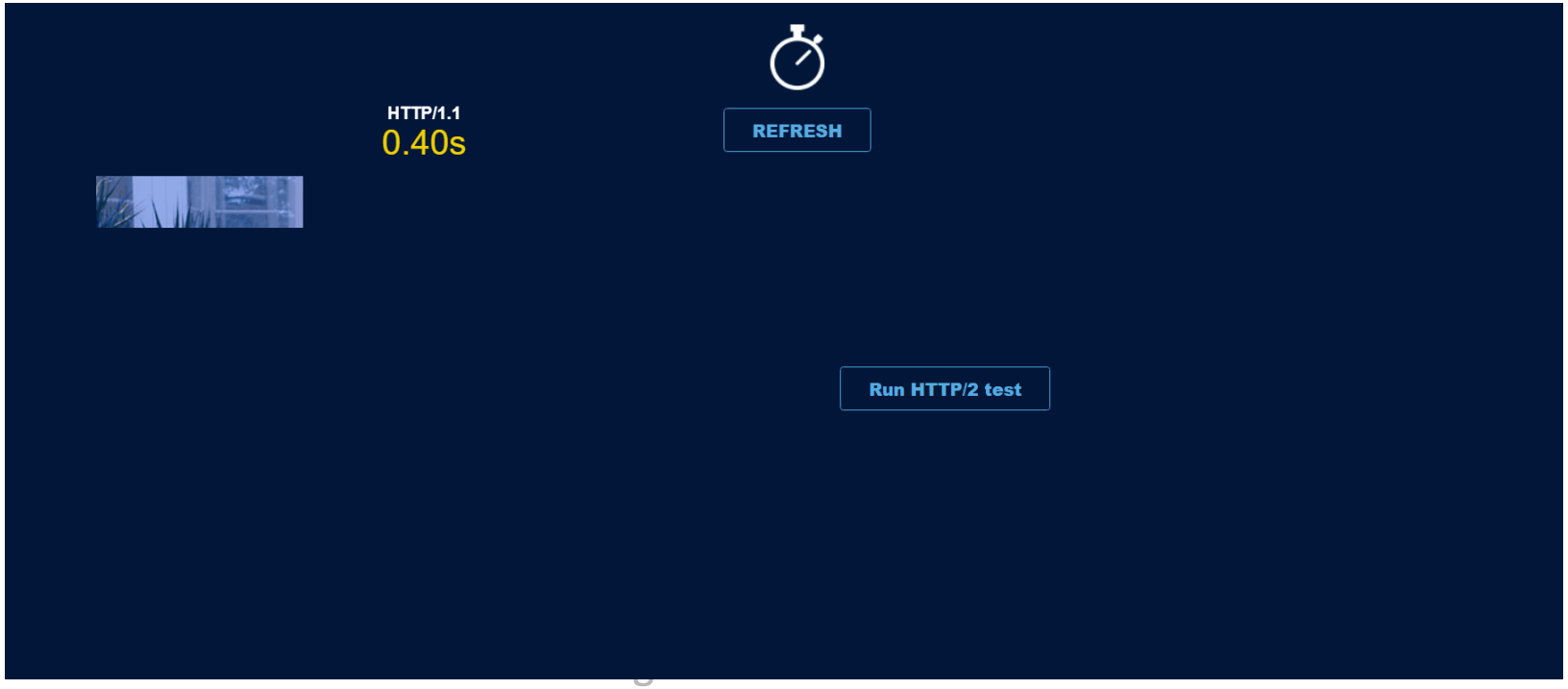
```
5052 4920 2a20 4854 5450 2f32
0d0a 534d 0d0a 0d0a 0000 0004
0000 0000 0401 0000 0000 0000
```



https://nilsmagnus.github.io/post/proto-json-sizes/

Key features - Performance

- HTTP/2 multiplexing
  - Multiple calls via a TCP connection
  - Avoid head-of-line blocking*



Key features - Code generation

- All gRPC libraries have first-class code generation support

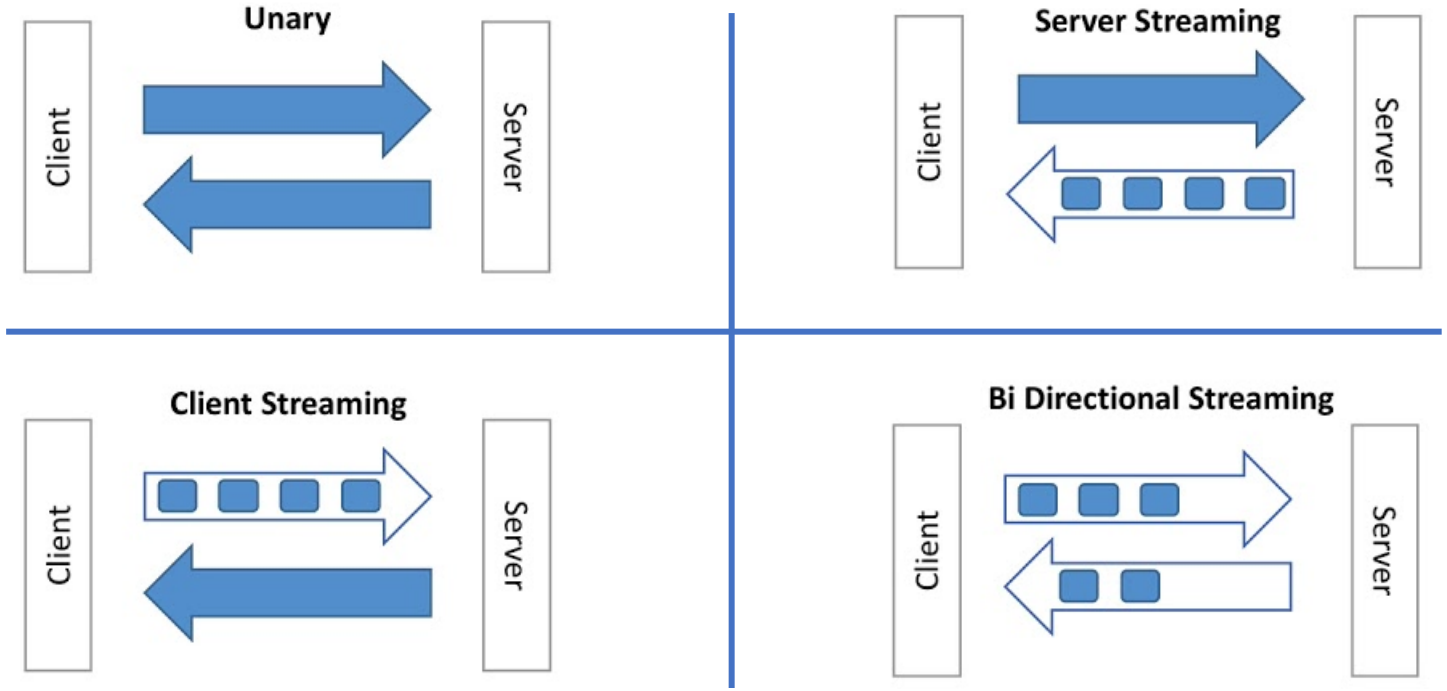
```
syntax = "proto3";

message SubscribeRequest {
  string topic = 1;
}

message Event {
  string details = 1;
}

service Topics {
  rpc Subscribe(SubscribeRequest)
    returns (stream Event);
}
```

```
public partial class TopicsClient : ClientBase<TopicsClient>
{
  public TopicsClient(ChannelBase channel) : base(channel)
  {
  }

  public virtual AsyncServerStreamingCall<Event> Subscribe(
    SubscribeRequest request,
    CallOptions options)
  {
    return CallInvoker.AsyncServerStreamingCall(__Subscribe, options, request);
  }
}
  </ItemGroup>

</Project>
```

Key features - Multiple languages

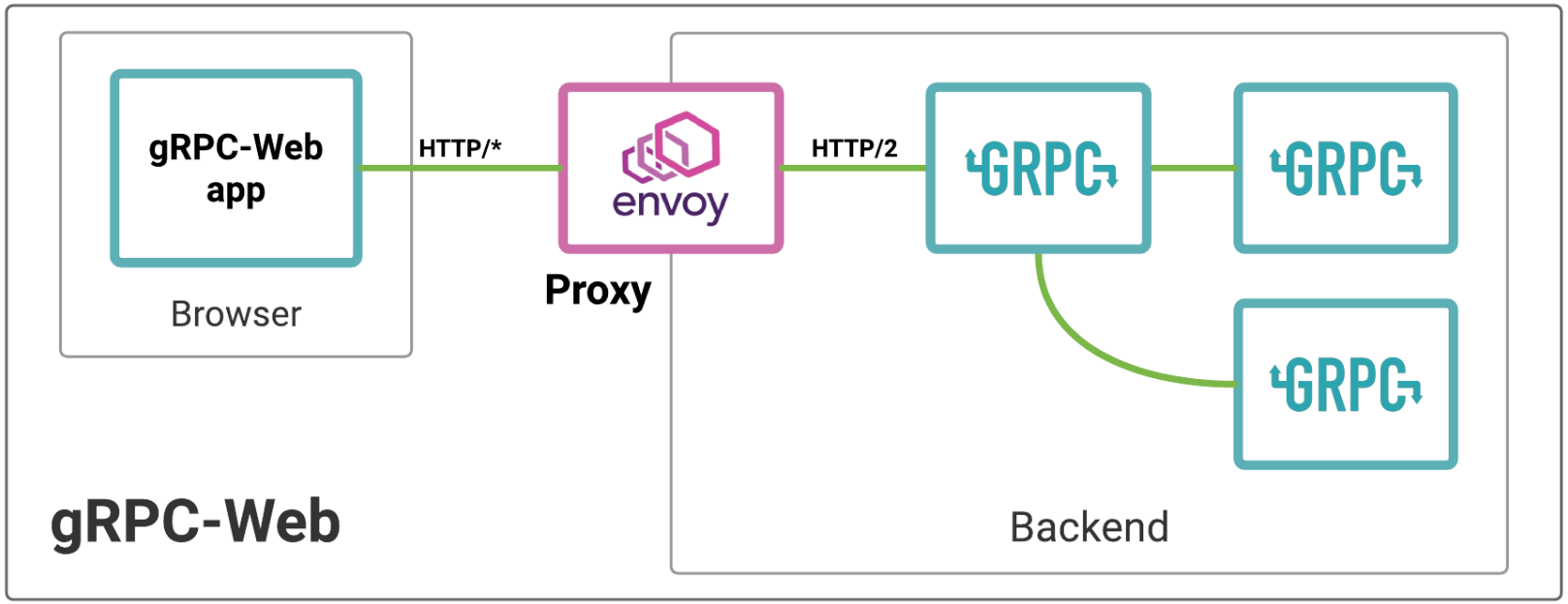
Key features - Streaming

- gRPC uses HTTP/2 to enable streaming



Disadvantages – Limited browser support

- Browsers have great HTTP/2 support ☺
- Browser JavaScript APIs haven't caught up ☹
- gRPC-web provides limited support for calling gRPC services

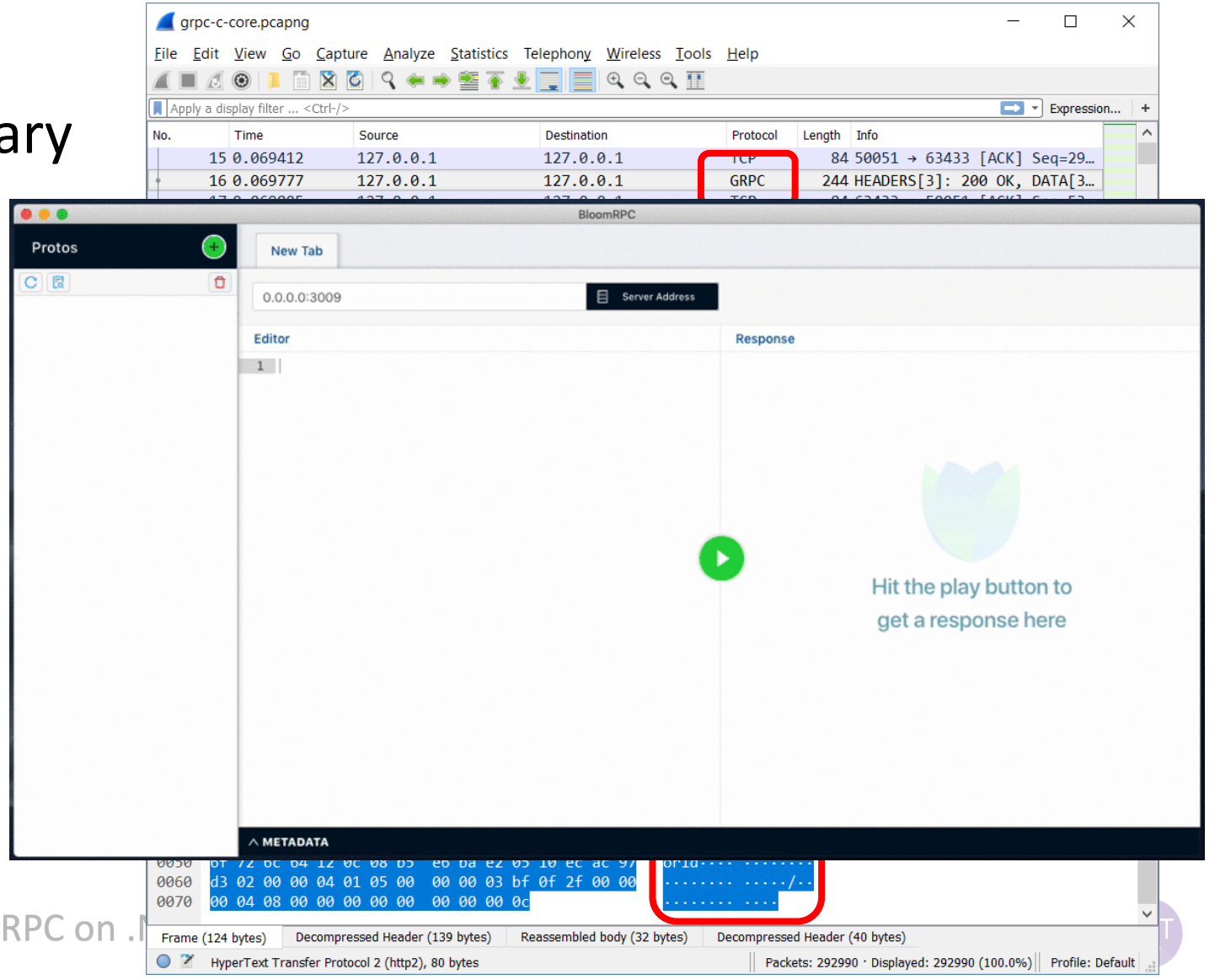


Disadvantages - Not human readable

- HTTP/2 and Protobuf are binary protocols
- Additional tools required to debug calls

BloomRPC

WIRESHARK



API / SPA Auth

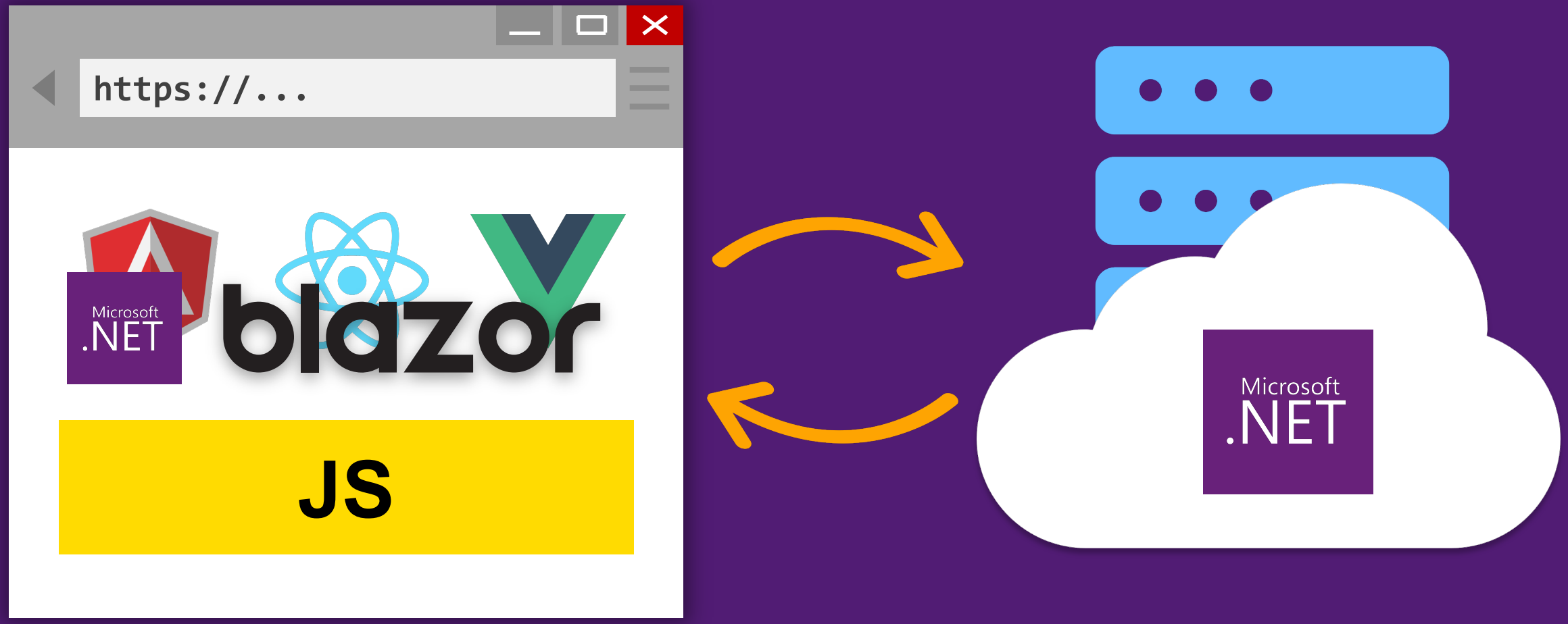
.NET

DEMO

.NET Core 3.0 Microservices

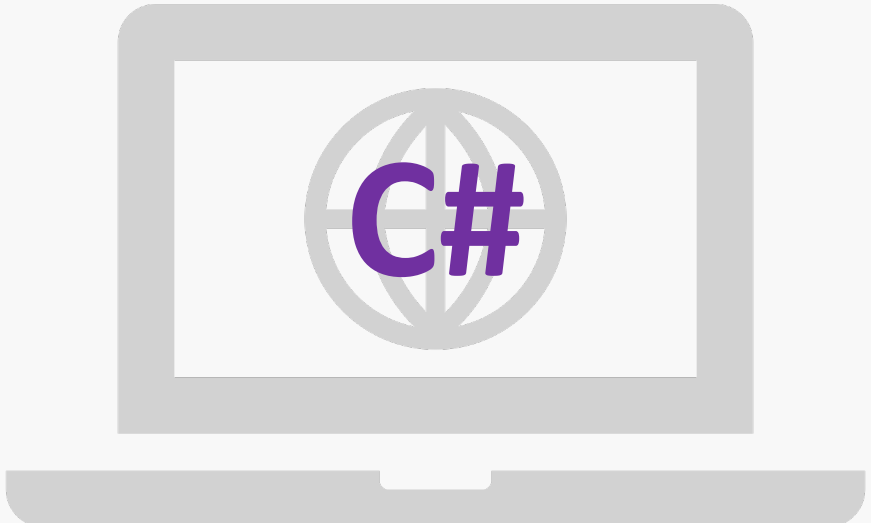
Web apps with Blazor

.NET


https://...
Microsoft
.NET
blazor
JS
Microsoft
.NET

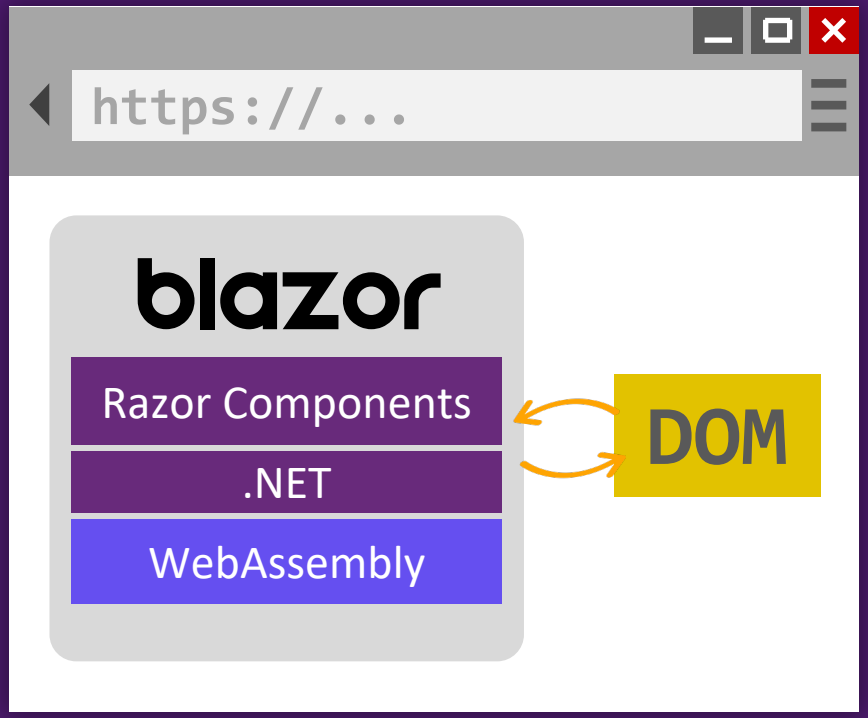
@Blazor

- Build client-side web UI with .NET instead of JavaScript
- Write reusable web UI components with C# and Razor
- Share .NET code with both the client and the server
- Call into JavaScript libraries & browser APIs as needed



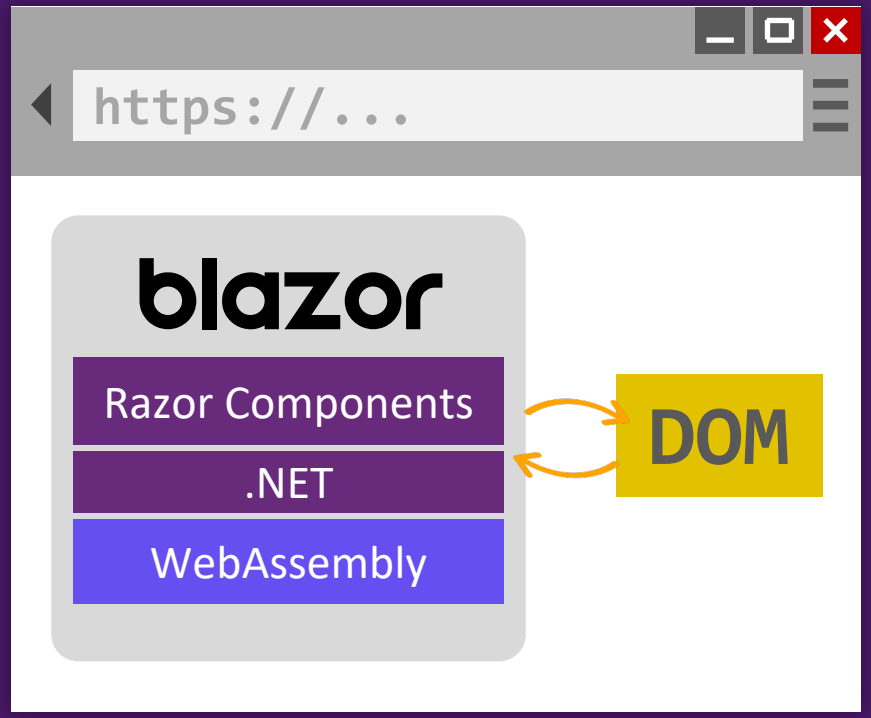

.NET
WA

How Blazor WebAssembly works



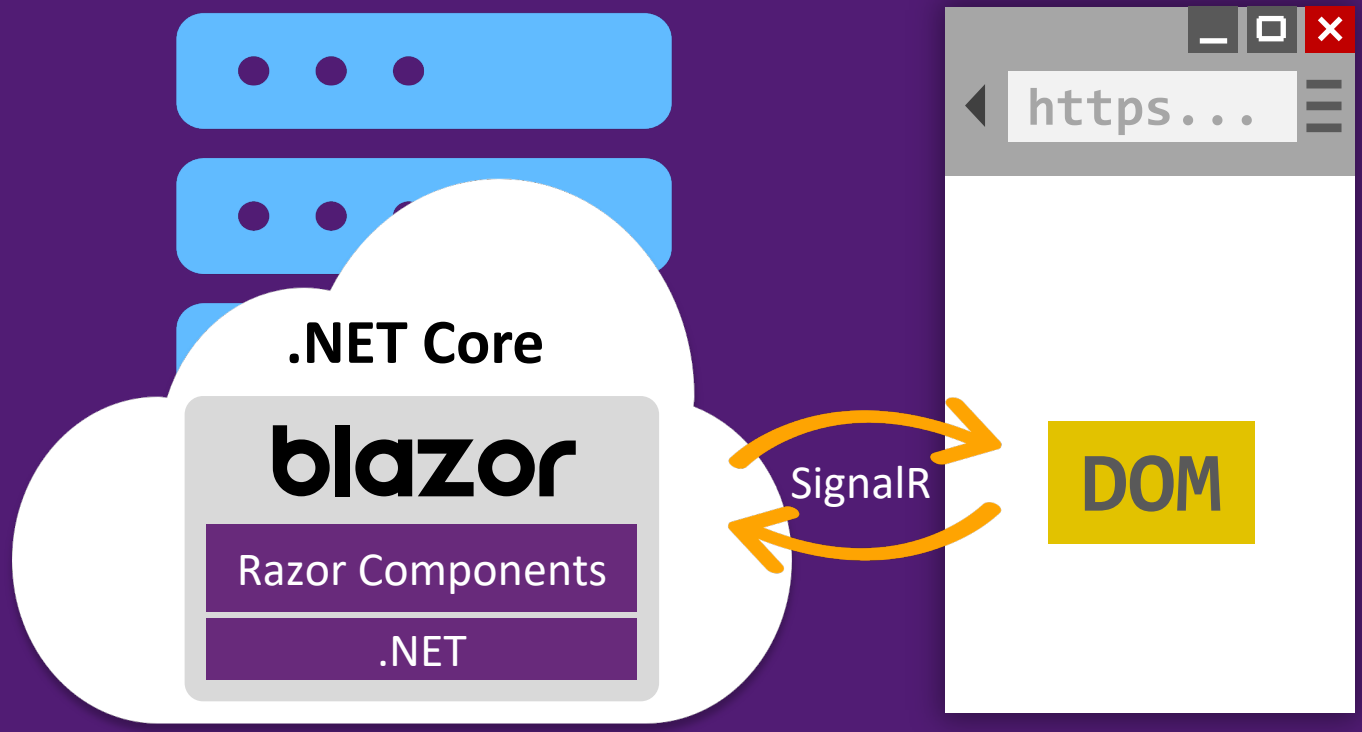
Blazor on client or server

Blazor WebAssembly



May 2020

Blazor Server



.NET Core 3.0

Blazor on client or server

Blazor WebAssembly

Pro:
- True SPA, full interactivity
- Utilize client resources
- Supports offline, static sites, PWA scenarios

Con:
- Larger download size
- Requires WebAssembly
- Still in preview

May 2020

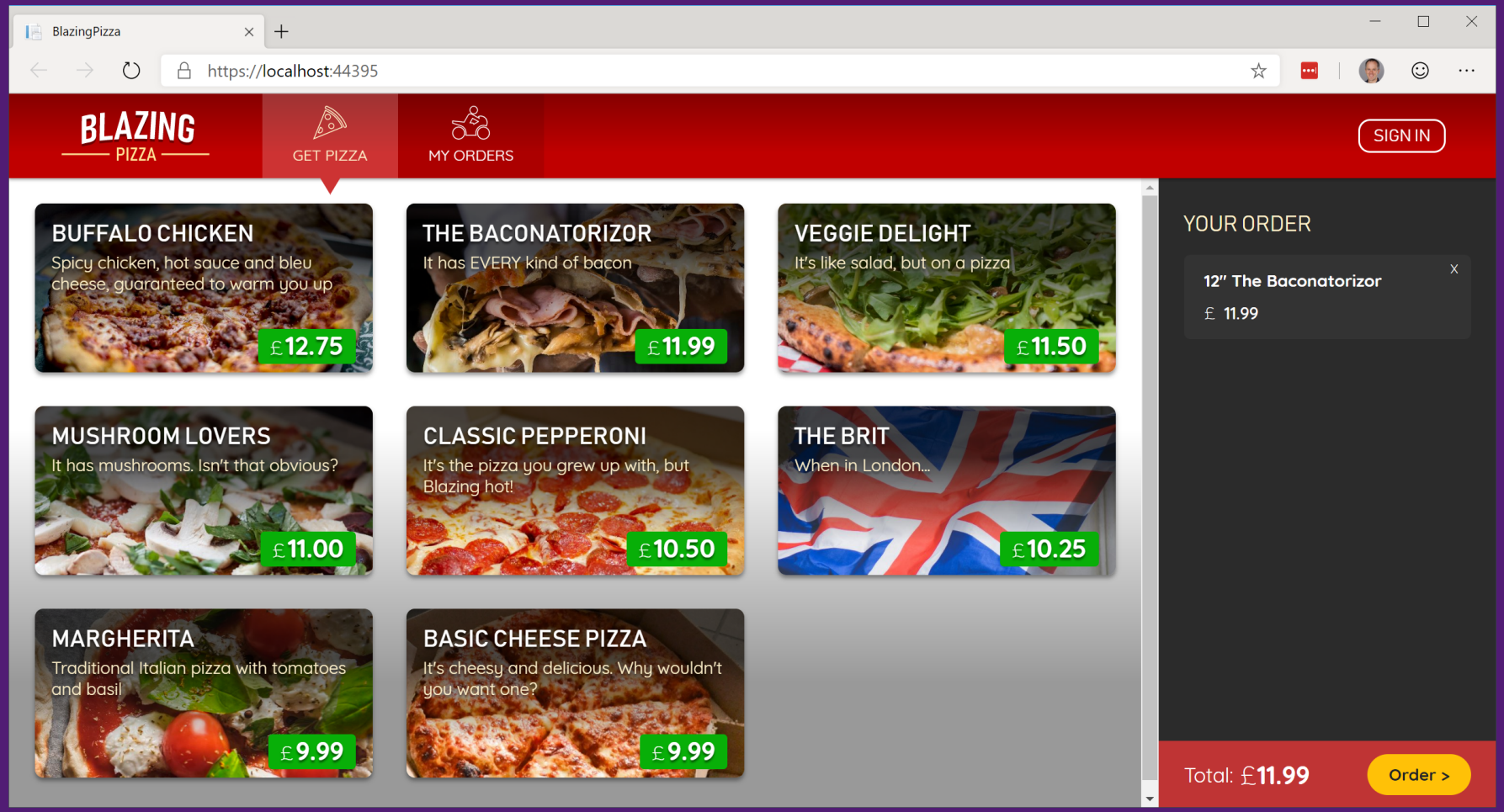
Blazor Server

Pro:
- Smaller download size, faster load time
- Running on fully featured .NET runtime
- Code never leaves the server
- Simplified architecture

Con:
- Latency
- No offline support
- Consumes more server resources

.NET Core 3.0

Build your own pizza store UI with Blazor



https://aka.ms/blazorworkshop

"Telerik UI for Blazor components have been built from the ground-up to ensure you experience shorter development cycles, quick iterations and cut time to market"

https://www.telerik.com/blazor-ui

"DevExpress UI for Blazor ships with 12 UI components (including a Data Grid, Pivot Grid, Charts and Scheduler) so you can design rich user experiences for both Blazor server-side and Blazor client-side platforms."

https://www.devexpress.com/blazor

"The Syncfusion ASP.NET Core Blazor Components library is the only suite that you will ever need to build an application, containing over 60 high-performance, lightweight, modular, and responsive UI controls in a single package."

https://www.syncfusion.com/blazor-components

The “Awesome Blazor” community

- https://aka.ms/awesomeblazor
- Free open-source components & JS interop libraries
- Lots of fun sample Blazor apps
- Articles, videos, blogs, and other learning materials
- Chat with the Blazor community on Gitter: https://gitter.im/aspnet/blazor
- Thank you, Simon and Chris!

Try Blazor today!

- Blazor: https://blazor.net
- Docs: https://blazor.net/docs
- .NET Core 3.0: https://dot.net/get-core3
- Visual Studio: https://visualstudio.com/
- Workshop: https://aka.ms/blazorworkshop
- Community: https://aka.ms/awesomeblazor

DEMO

Blazor

Blazor… on the desktop???

.NET

Machine Learning

.NET

ML.NET

Built for .NET developers

Create custom ML models using C# or F# without having to leave the .NET ecosystem

Custom ML made easy with AutoML

Visual Studio Model Builder and CLI make it super easy to build custom ML Models

Extended with TensorFlow & more

Leverage other popular ML frameworks (TensorFlow, ONNX, and more)

A few things you can do with ML.NET

Sentiment analysis

Analyze the sentiment of customer reviews using a binary classification algorithm.

Product recommendation

Recommend products based on purchase history using a matrix factorization algorithm.

Price prediction

Predict taxi fares based on distance traveled etc. using a regression algorithm.

Customer segmentation

Identify groups of customers with similar profiles using a clustering algorithm.

GitHub labeler

Suggest the GitHub label for new issues using a multi-class classification algorithm.

Fraud detection

Detect fraudulent credit card transactions using a binary classification algorithm.

Spam detection

Flag text messages as spam using a binary classification algorithm.

Image classification

Classify images (e.g. broccoli vs pizza) using a TensorFlow deep learning algorithm.

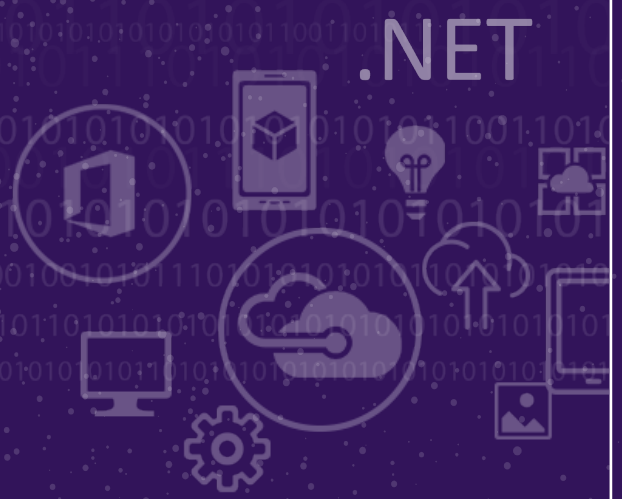
Sales forecasting

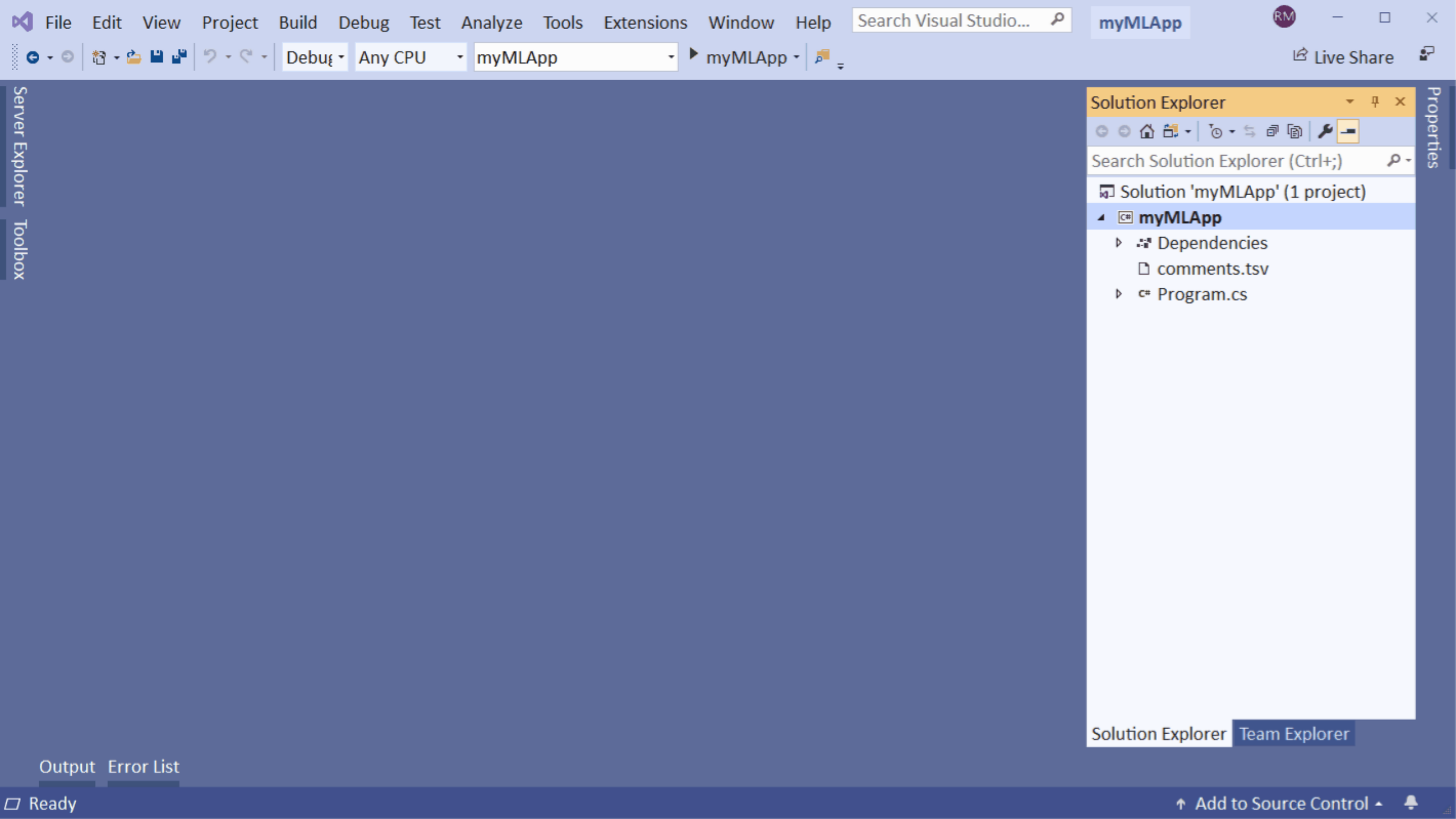
Forecast future sales for products using a regression algorithm.

dot.net/ml

DEMO

ML.NET




File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio...
myMLApp
RM
Debug
Any CPU
myMLApp
myMLApp
Live Share
Server Explorer
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'myMLApp' (1 project)
myMLApp
Dependencies
comments.tsv
Program.cs
Properties
Solution Explorer
Team Explorer
Output
Error List
Ready
Add to Source Control

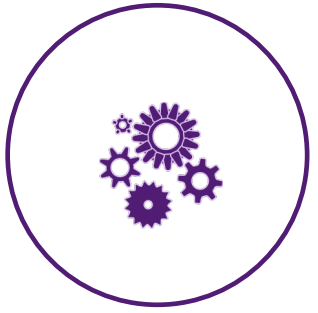
PREVIEW

.NET for Apache Spark

Available on Azure Databricks and Azure HDInsight

Spark SQL + DataFrames

Streaming & Interactive

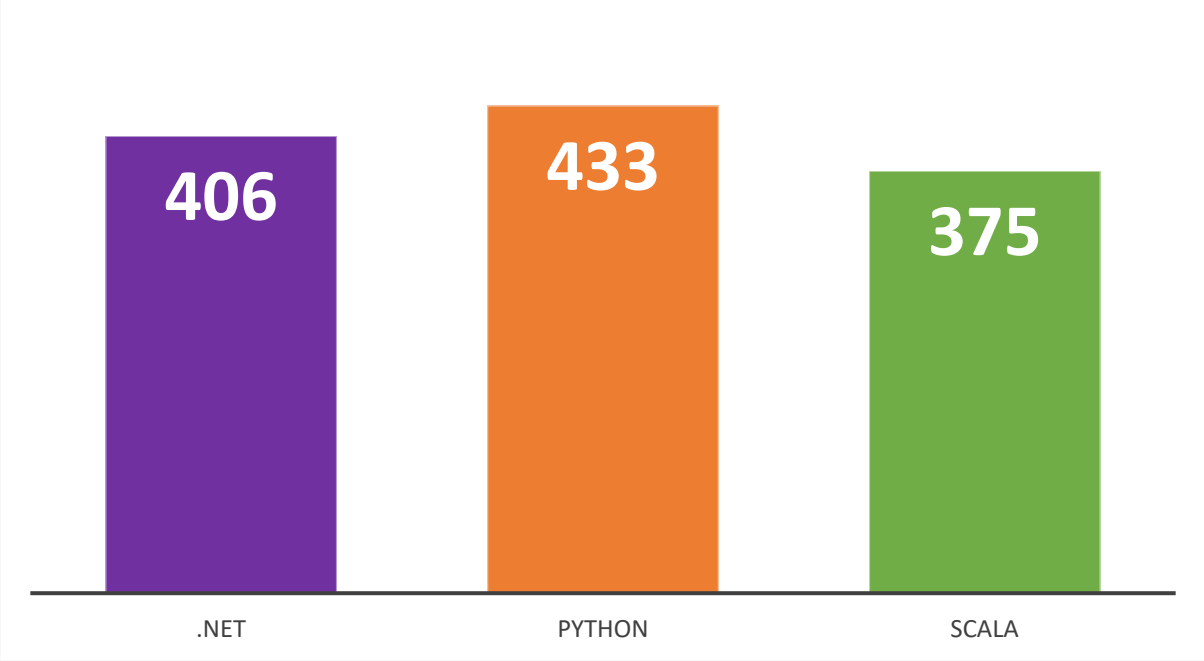


Machine Learning

Speed & Productivity

dot.net/spark

.NET for Apache Spark Performance



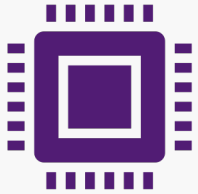
Total execution time (seconds) for all 22 queries in the TPC-H benchmark (lower is better).
Data sourced from an internal run of the TPC-H benchmark, using warm execution on Ubuntu 16.04.

.NET for Apache Spark is designed for high performance and performs better than python on the TPC-H benchmark tpc.org/tpch.

The TPC-H benchmark consists of a suite of business-oriented queries.

Learn more: dot.net/spark

.NET Core 3.0 IoT Support



Supports Raspberry Pi and other devices

You can now run .NET Core apps in small places, including ASP.NET

Read sensor data & write to displays

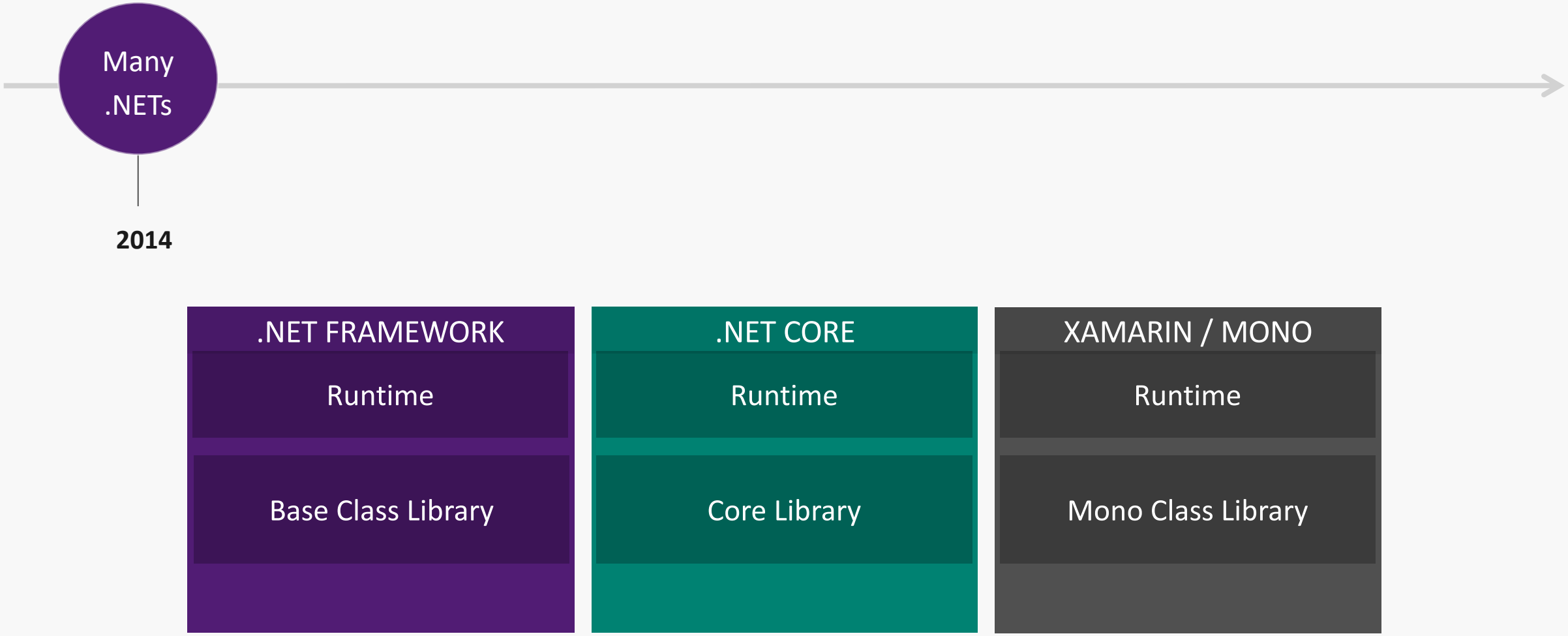
New APIs for GPIO pins that enable using millions of IoT peripherals

Works with containers

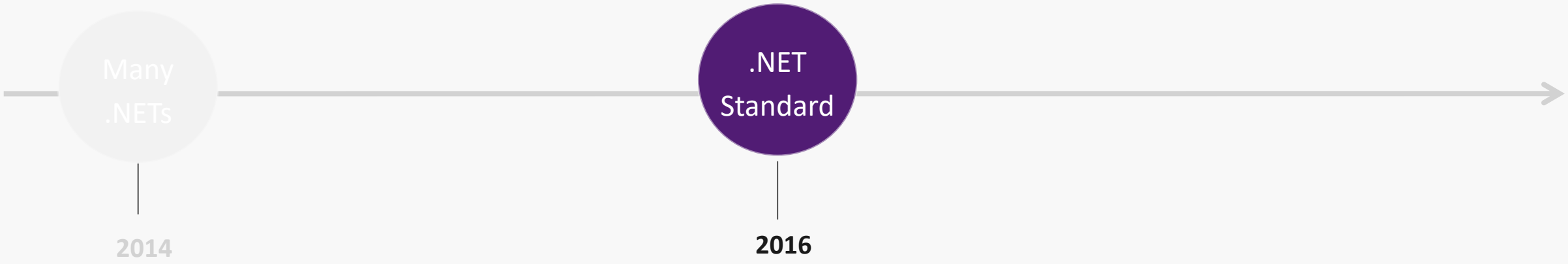
Deploy apps directly onto devices or with containers

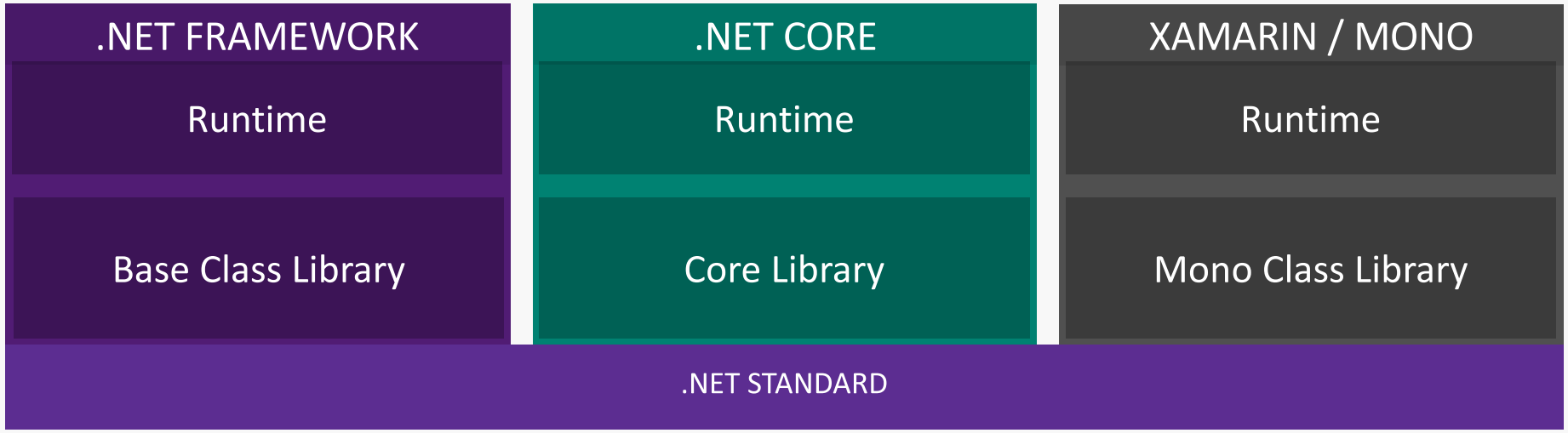
.NET 5

The .NET Roadmap

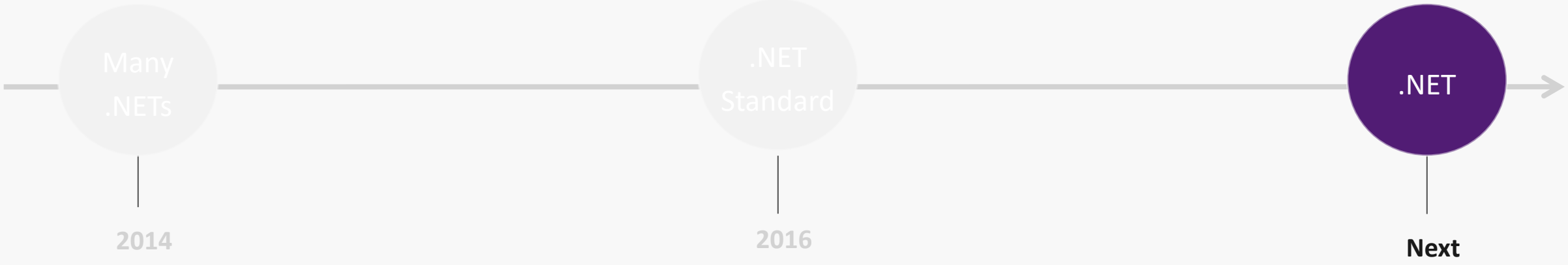


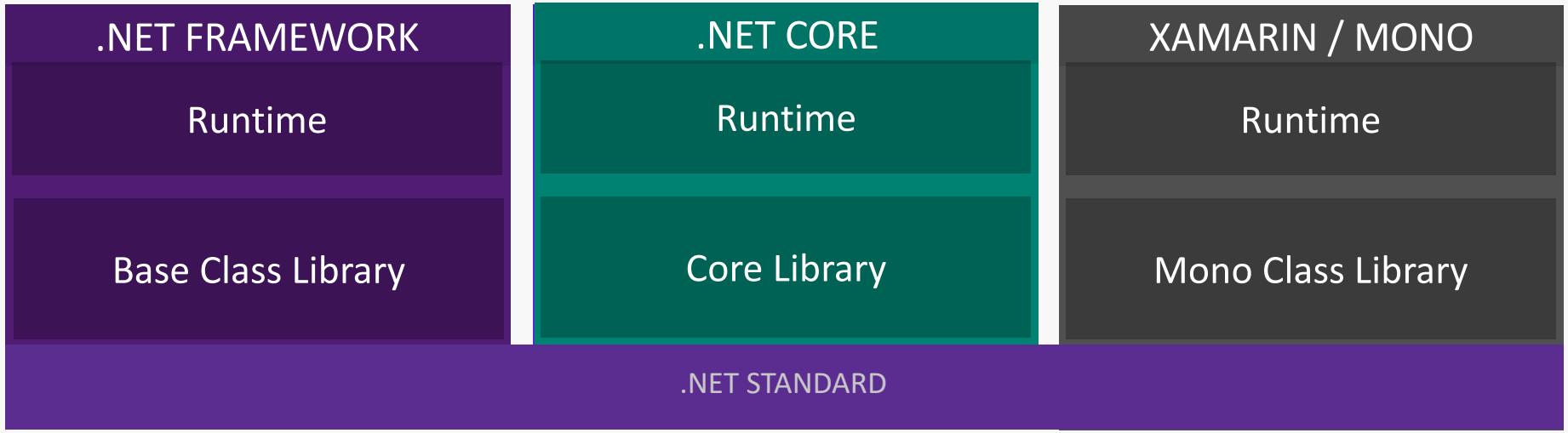
The .NET Roadmap





Introducing .NET 5





Introducing .NET 5

- Supports all .NET application types
- Evolution of .NET Core, adding the best of Mono into one unified platform
  - One BCL implementation, still adheres to .NET Standard
  - One toolchain (SDK style projects)
  - Both Just-in-Time (JIT) and Native models supported
  - Interop with Java and Swift

https://devblogs.microsoft.com/dotnet/introducing-net-5/

Rich Lander
@runfaster2000

I'm planning .NET Core 5.0. It's still very early, so is a great time to consider behavior (AKA "breaking") changes. I'm collecting a list of undesirable behaviors or APIs in .NET Core that we should consider changing in 5.0. What do you want changed?
#constantimprovement

7:58 PM · Nov 1, 2019 · Twitter Web App

Rich Lander
@runfaster2000

What changes should we make to the @dotnet CLI. Same question as I asked earlier but not about APIs but the “dotnet” command.

What is ***not*** in .NET 5?

- Web Forms, WCF Server and Windows Workflow remain on .NET Framework 4.8 **only**. There are no plans to port these.

- Recommendations
  - ASP.NET Blazor for ASP.NET Web Forms (we will provide a migration guide)
  - gRPC for WCF Server and Remoting (we will provide a migration guide)
  - Open Source Core Workflow for Windows Workflow (WF): https://github.com/UiPath/corewf

The Future of .NET Framework

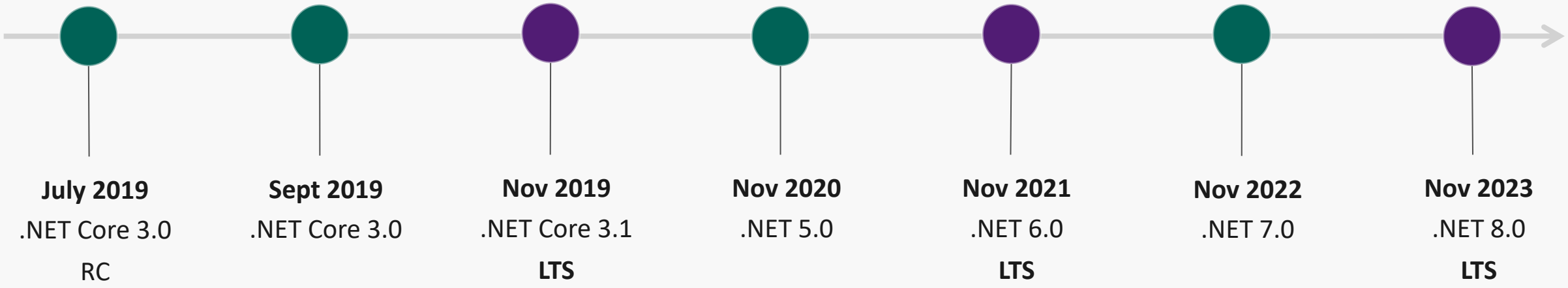
- .NET Framework 4.8 is the last major version of .NET Framework on Windows
- Support policy remains the same:
  - Will always be in Windows
  - Will be patched with Windows
  - Will be supported with Windows
- Keep existing applications on .NET Framework
- Recommend .NET Core for new applications

Some Update Recommendations

- New Development: .NET Core 3.1 LTS
- Windows Forms / WPF: Update to .NET Core 3
- Web Forms: Look at Blazor
- WCF: gRPC or CoreWCF
- .NET Framework is supported
- Use Migration Guides

.NET – A unified platform

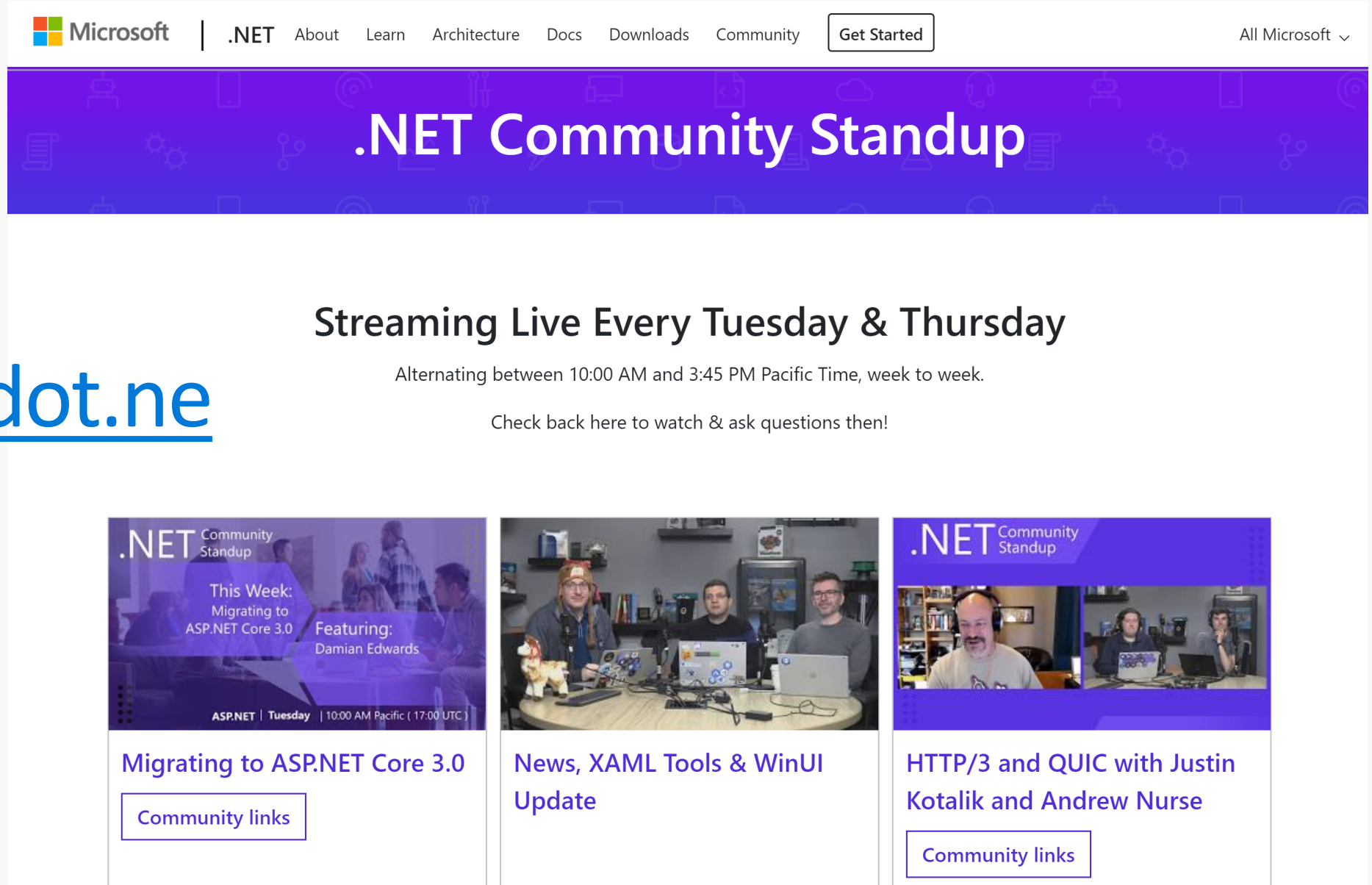
.NET Schedule



- .NET Core 3.0 released
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

.NET Community Standup

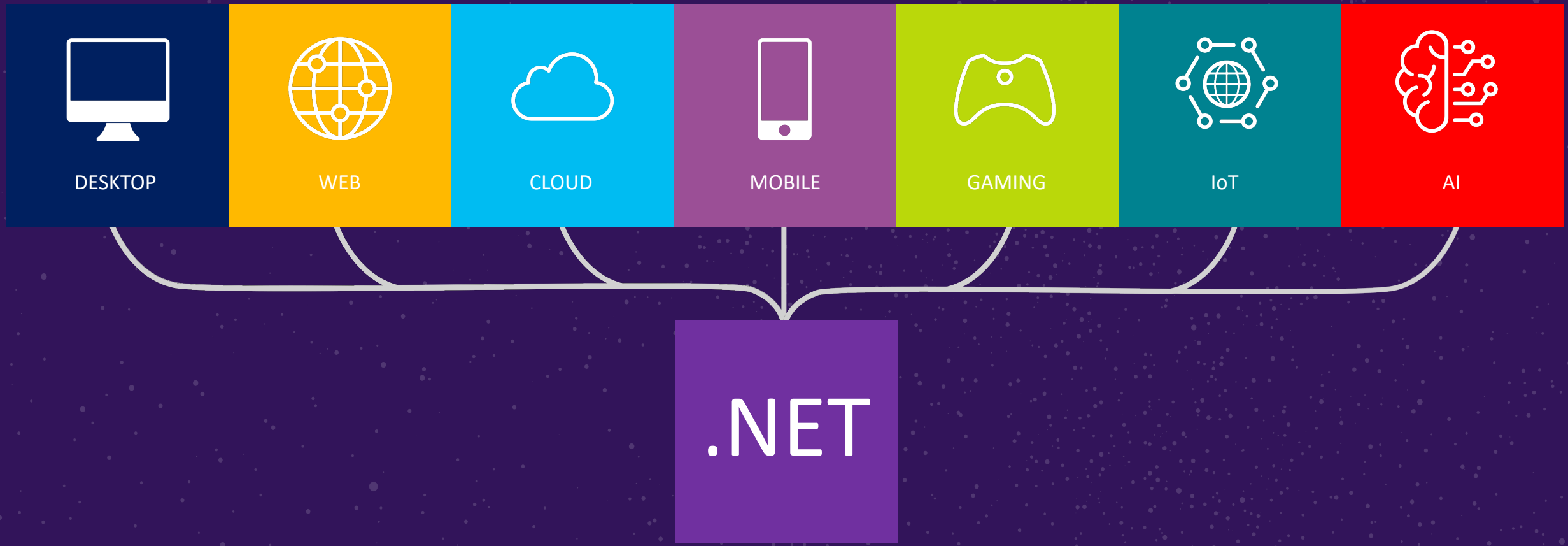
https://live.dot.net



Download .NET Core 3.0 Today!

dot.net/get-core3

visualstudio.com/downloads



Slides and Links

https://aka.ms/dotnext-2019-dotnetcore-3