# What's new* in .NET 8

Snapbean
Software AG

Ralph Bodmer
Manu Sidler

*new as of in September 2024

# .NET Plattform
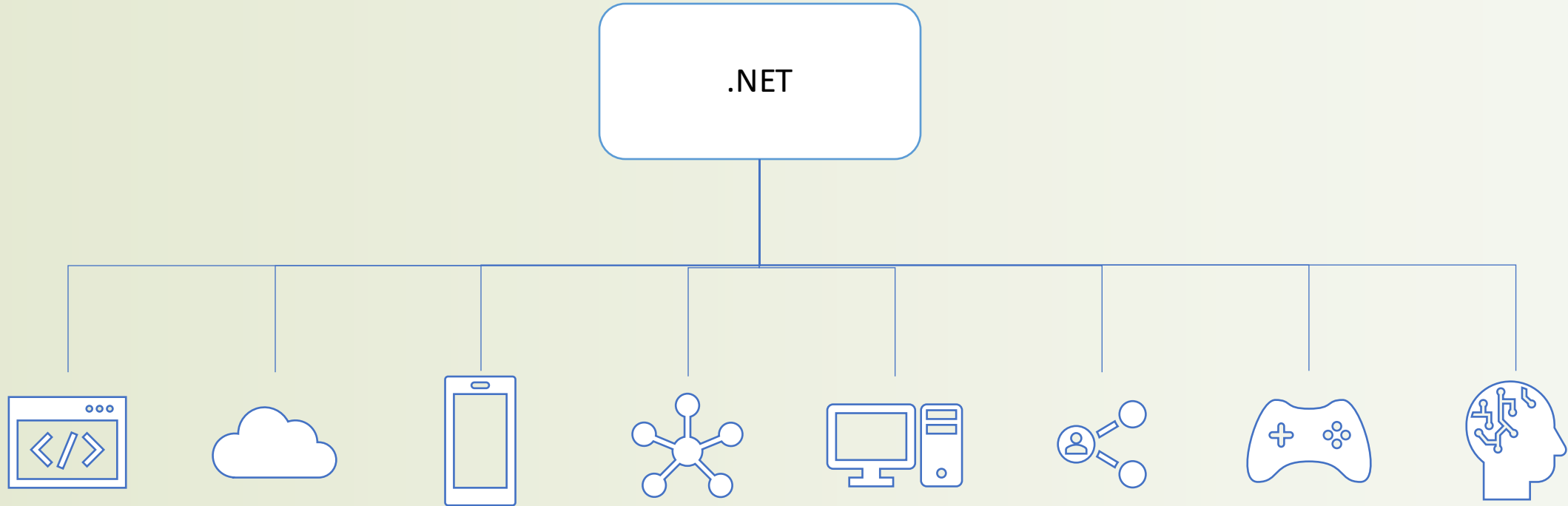## Rückblick – Pre .NET 5

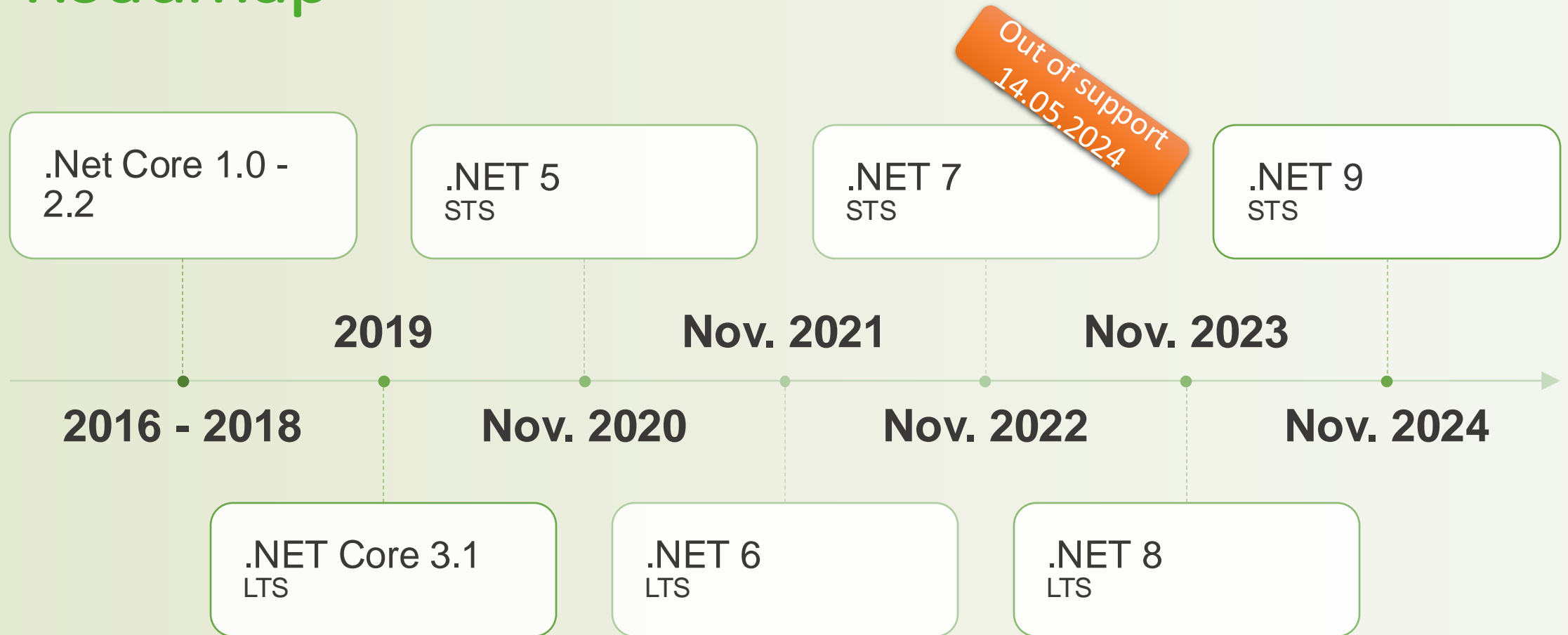# .NET Plattform
## Roadmap

Long Term Support = 3 Jahre
Short Term Support = 18 Monate

**Out of support 14.05.2024**

.Net Core 1.0 - 2.2

.NET 5
STS

.NET 7
STS

.NET 9
STS

**2019**

**Nov. 2021**

**Nov. 2023**

**2016 - 2018**

**Nov. 2020**

**Nov. 2022**

**Nov. 2024**

.NET Core 3.1
LTS

.NET 6
LTS

.NET 8
LTS

**Snapbean**
Software AG

# .NET 7
## Rückblick - Features

- **Performance**
1000 PRs mit Performance Verbesserungen

- **C# 11**
Raw string literals, required properties, list patterns

- **ASP.NET Core**
Rate-Limiting Middleware, Minimal API Verbesserungen

Snapbean
Software AG

# .NET 8
## Features

- C#12
  Primary Constructors, Collection Expressions, …

- ASP.NET Core
  Keyed Services, Short circuit, BackgroundServices, …

- .NET Aspire
  Cloud-ready Stack für verteilte Applikationen

# .NET 8
## Upgrade

```
--- a/Upgrade.csproj
+++ b/Upgrade.csproj
@@ -2,13 +2,13 @@

   <PropertyGroup>
     <OutputType>Exe</OutputType>
-    <TargetFramework>net7.0</TargetFramework>
+    <TargetFramework>net8.0</TargetFramework>^M
     <ImplicitUsings>enable</ImplicitUsings>
     <Nullable>enable</Nullable>
   </PropertyGroup>

   <ItemGroup>
-    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="7.0.8" />
+    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="8.0.4" />^M
   </ItemGroup>

 </Project>
( END )
```

Snapbean
Software AG

# .NET 8
## Tooling

- Visual Studio 2022 17.8+
  Http Files Support

  GitHub Copilot

Snapbean
Software AG

# .NET 8
# Tooling

- ## CLI
  Modernisierter Build Output



```
dotnet build
MSBuild version 17.8.3+195e7f5a3 for .NET
  Determining projects to restore...
  All projects are up-to-date for restore.
  02_RequiredProperties -> /Users/manuel/git/developer-day-2024/csharp/csharp11/02_RequiredPropert
ies/bin/Debug/net7.0/02_RequiredProperties.dll
  01_RawStringLiterals -> /Users/manuel/git/developer-day-2024/csharp/csharp11/01_RawStringLiteral
s/bin/Debug/net7.0/01_RawStringLiterals.dll
/Users/manuel/git/developer-day-2024/csharp/csharp11/03_ListPatterns/Program.cs(24,19): error CS01
03: The name 'resadfsult' does not exist in the current context [/Users/manuel/git/developer-day-2
024/csharp/csharp11/03_ListPatterns/03_ListPatterns.csproj]

Build FAILED.

/Users/manuel/git/developer-day-2024/csharp/csharp11/03_ListPatterns/Program.cs(24,19): error CS01
03: The name 'resadfsult' does not exist in the current context [/Users/manuel/git/developer-day-2
024/csharp/csharp11/03_ListPatterns/03_ListPatterns.csproj]
    0 Warning(s)
    1 Error(s)

Time Elapsed 00:00:00.78
```

```
dotnet build --tl
MSBuild version 17.8.3+195e7f5a3 for .NET
Restore complete (0.3s)
  02_RequiredProperties succeeded (0.1s) → 02_RequiredProperties/bin/Debug/net7.0/02_RequiredPrope
rties.dll
  01_RawStringLiterals succeeded (0.1s) → 01_RawStringLiterals/bin/Debug/net7.0/01_RawStringLitera
ls.dll
  03_ListPatterns failed with errors (0.6s)
    /Users/manuel/git/developer-day-2024/csharp/csharp11/03_ListPatterns/Program.cs(24,19): error
CS0103: The name 'resadfsult' does not exist in the current context

Build failed with errors in 1.0s
```

Snapbean
Software AG

# .NET 8
## Tooling

- ## CLI
  MSBuild Daten für Build Scripts nutzen

```
dotnet publish --getProperty:OutputType
Exe
```

```xml
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <LangVersion>12.0</LangVersion>
    <ImplicitUsings>true</ImplicitUsings>
  </PropertyGroup>
</Project>
```

# .NET 8
## Tooling

Demo Time !

Snapbean
Software AG

# C#
## 10

- Global usings

- Implicit usings

- Extended property patterns

```
var isFromStevenKing = book is { Author.Name: "Steven King" };

string bookDescription = book switch
{
    { ISBN: "1234" } or { ISBN: "4321" } => "Book has ISBN 1234 oder 4321",
    { Author.Name: "Steven King" } => "Book is from Steven King",
    _ => "No idea"
};
```

Snapbean
Software AG

# C#
## 11

- Raw string literals

- Required properties

- Extended list patterns

```
const string json = $$"""
    [
        {
            "firstName": "{{firstName}}",
            "lastName": "{{lastName}}"
        }
    ]
    """;

public class Person
{
    public required string FirstName { get; set; }
    public required string LastName { get; set; }
    public int Age { get; set; }
}

var numbers = new int[] { 1, 2, 4, 8, 16 };
numbers is [1, 2, 4, 8, 16]; // true
numbers is [_, _, _, 8]; // false
numbers is [1, ..]; // true
numbers is [1, var a, ..]; // true
```

Snapbean
Software AG

# C#
## 12

- Primary Constructors

- Collection Expressions

- Default lambda parameters

- Alias any type

- (Erweiterte Data Annotations)

[Language Feature Status](#)

Snapbean
Software AG

**C#**

Demo Time !

Snapbean
Software AG

# C#

- Neue C# Features in "alten" .NET Applikationen nutzen

- Erstellt PollyFills über Source Generators

- Einschränkungen: von Runtime abhängige Features

https://github.com/Sergio0694/PolySharp

Snapbean
Software AG

# C#

Demo Time !

# ASP.NET Core 8
## Architektur Übersicht

**ASP.NET Core Applications**

| ASP.NET Core MVC | ASP.NET Core API Minimal API | ASP.NET Core Razor Pages | ASP.NET Core Blazor |

**Miscellaneous**

| SignalR | gRPC |

**ASP.NET Core Platform**

| Model binding | Middleware | Dependency Injection |
| Logging | Razor | Kestrel |

Snapbean
Software AG

# ASP.NET Core 8
## Keyed Service Registration

- Neues Dependency Injection feature
  - Erweiterung des Service Descriptors mit einem Key

```csharp
// Regular registration
builder.Services.AddSingleton<ILoggingService, FileLogService>();
builder.Services.AddSingleton<ILoggingService, DatabaseLogService>();

// Keyed registration
builder.Services.AddKeyedSingleton<ILoggingService, FileLogService>("file");
builder.Services.AddKeyedSingleton<ILoggingService, DatabaseLogService>("db");
```

Snapbean
Software AG

# ASP.NET Core 8
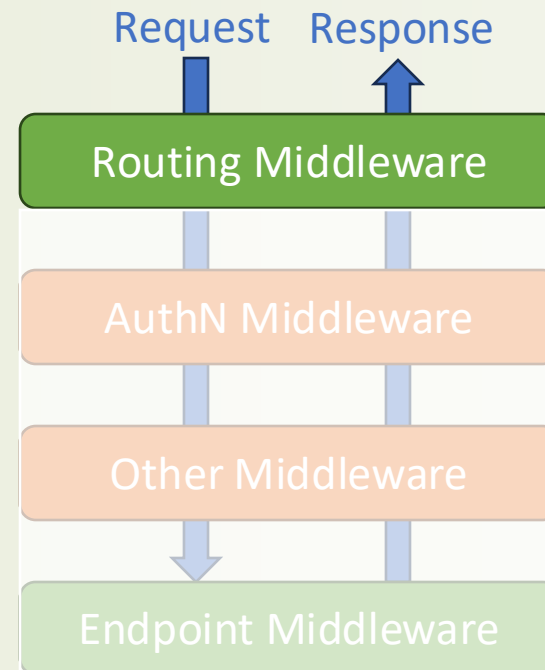## Keyed Service Registration

Demo Time!

Snapbean
Software AG
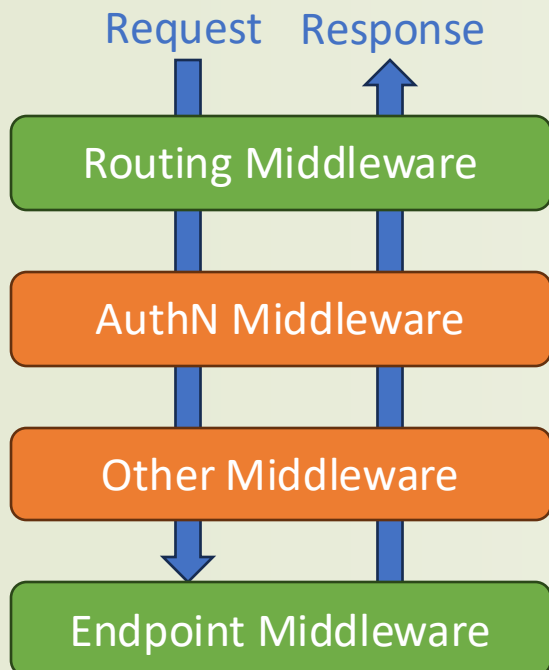
# ASP.NET Core 8
## Short-circuit routing

```
app.MapGet('/health', () ⇒ Results.Ok()).ShortCircuit()

app.MapShortCircuit(404, 'favicon.ico')
```

- Neue Funktion zum Überspringen von Middleware

# ASP.NET Core 8
## Short-circuit routing

Demo Time !

Snapbean
Software AG

# ASP.NET Core 8
## Native AOT

- Aktueller Support für folgende Templates
  - gRPC Template
  - Minimal APIs

```
// Regular builder
var builder = WebApplication.CreateBuilder(args)

// AOT builder
var builder = WebApplication.CreateSlimBuilder(args)
```

ASP.NET Core support for Native AOT

Snapbean
Software AG

# ASP.NET Core 8
## Improved Background Service

- Pre .NET 8
  - IHostedService (Start-, StopAsync)
  - BackgroundService BaseClass

- Ab .NET 8
  - IHostedLifecycleService

```csharp
namespace Microsoft.Extensions.Hosting
{
    public interface IHostedLifecycleService : IHostedService
    {
        Task StartingAsync(CancellationToken cancellationToken);
        // Task StartAsync(CancellationToken cancellationToken);
        Task StartedAsync(CancellationToken cancellationToken);
        Task StoppingAsync(CancellationToken cancellationToken);
        // Task StopAsync(CancellationToken cancellationToken);
        Task StoppedAsync(CancellationToken cancellationToken);
    }
}
```

```csharp
// Configure to start concurrently
builder.Services.Configure<HostOptions>(hostOptions =>
{
    hostOptions.ServicesStartConcurrently = true;
    hostOptions.ServicesStopConcurrently = true;
});
```

```csharp
// Hosted service registration
builder.Services.AddHostedService<PreviousHostedService>();
builder.Services.AddHostedService<PreviousBackgroundService>();
builder.Services.AddHostedService<NewHostedLifecycleService>();
```

Snapbean
Software AG

# ASP.NET Core 8
## Improved Background Service

Demo Time!

Snapbean
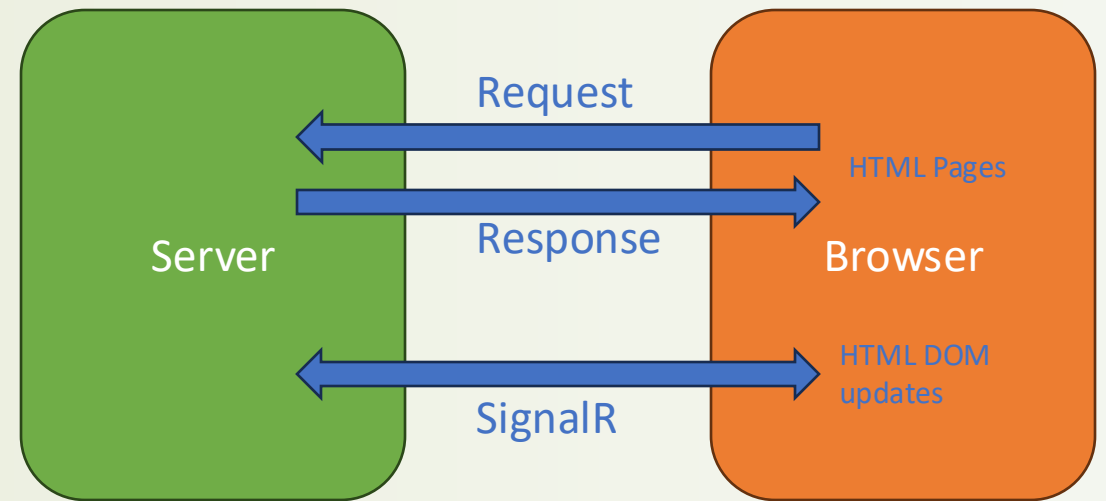Software AG

# ASP.NET Core 8
## Blazor

- Pre .NET 8
  - Blazor Server
  - Blazor WebAssembly (WASM)

- Ab .NET 8
  - Blazor Full Stack Web UI
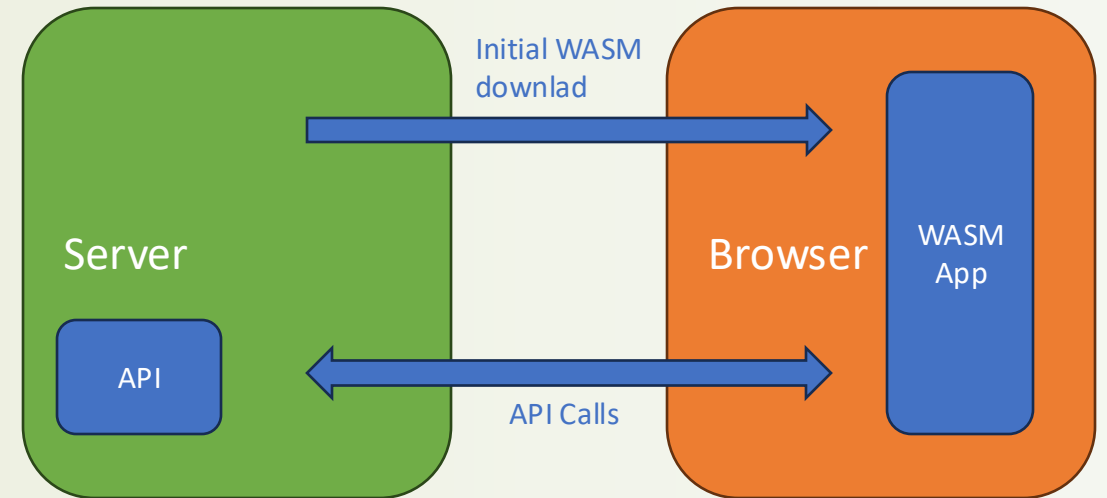
# ASP.NET Core 8
## Blazor Server

- Server-side rendering
  - Request / Response
- SignalR Verbindungen
  - WebSockets
  - Skalierung!



Server → Browser diagram:
- Request
- Response
- HTML Pages
- HTML DOM updates
- SignalR

Snapbean Software AG

# ASP.NET Core 8
## Blazor WebAssembly

- Native (WASM) Code
  - Client-side rendered (SPA ähnlich)
- Backend via APIs



Server

API

Initial WASM downlad

Browser

WASM App

API Calls

Snapbean
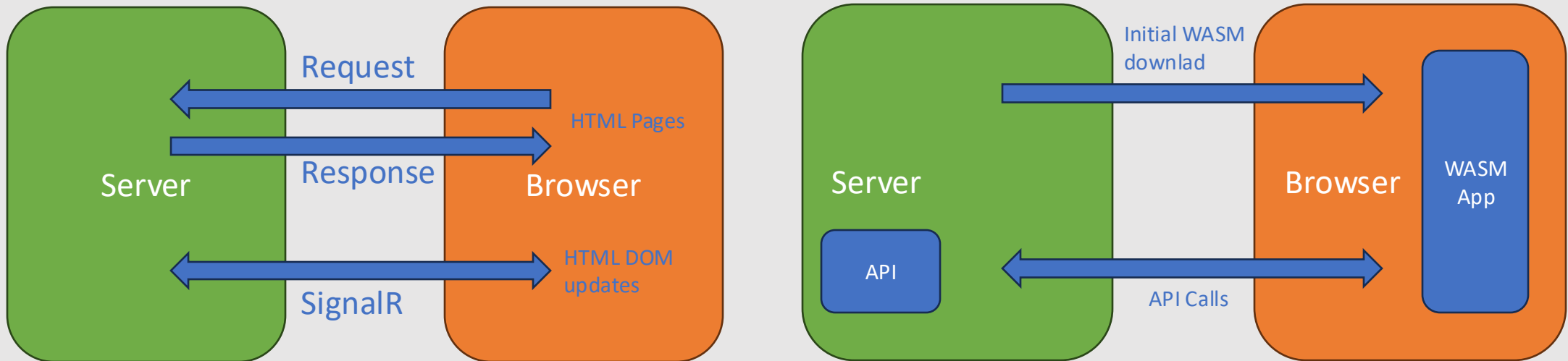Software AG

# ASP.NET Core 8
## Blazor Full Stack Web UI

- Neuer Typ - Server und WASM "kombiniert"
  - Projekt Template: Blazor Web App

- @renderMode
  - Interactive- None / Server / WebAssembly / Auto
  - Definierbar pro Seite, Komponente oder Applikation

Snapbean
Software AG

# ASP.NET Core 8
## Blazor Full Stack Web UI

Blazor Full Stack Web UI

Server — Request → Browser

Response ← HTML Pages

SignalR ← HTML DOM updates

Server — API

Initial WASM downlad → Browser — WASM App

API Calls ← 

Snapbean
Software AG

# ASP.NET Core 8
## Blazor Full Stack Web UI

Demo Time!

Snapbean
Software AG

# ASP.NET Core 8
## ASP.Net Core Identity

- Pre .NET 8
  - Default UI Seiten (Microsoft.AspNetCore.Identity.UI)


- Ab .NET 8
  - Default UI Seiten
  - Minimal API Endpunkte (Microsoft.AspNetCore.Identity…)

Snapbean
Software AG

# ASP.NET Core 8
## ASP.Net Core Identity

Demo Time!

Snapbean
Software AG

# ASP.NET Core 8
## Security Themen

- Update der "Crypto"-Libraries

- Default support: HTTP/3

- ...

# ASP.NET Core 9

- MinimalApi Erweiterungen

- Built-in OpenAPI documentation support
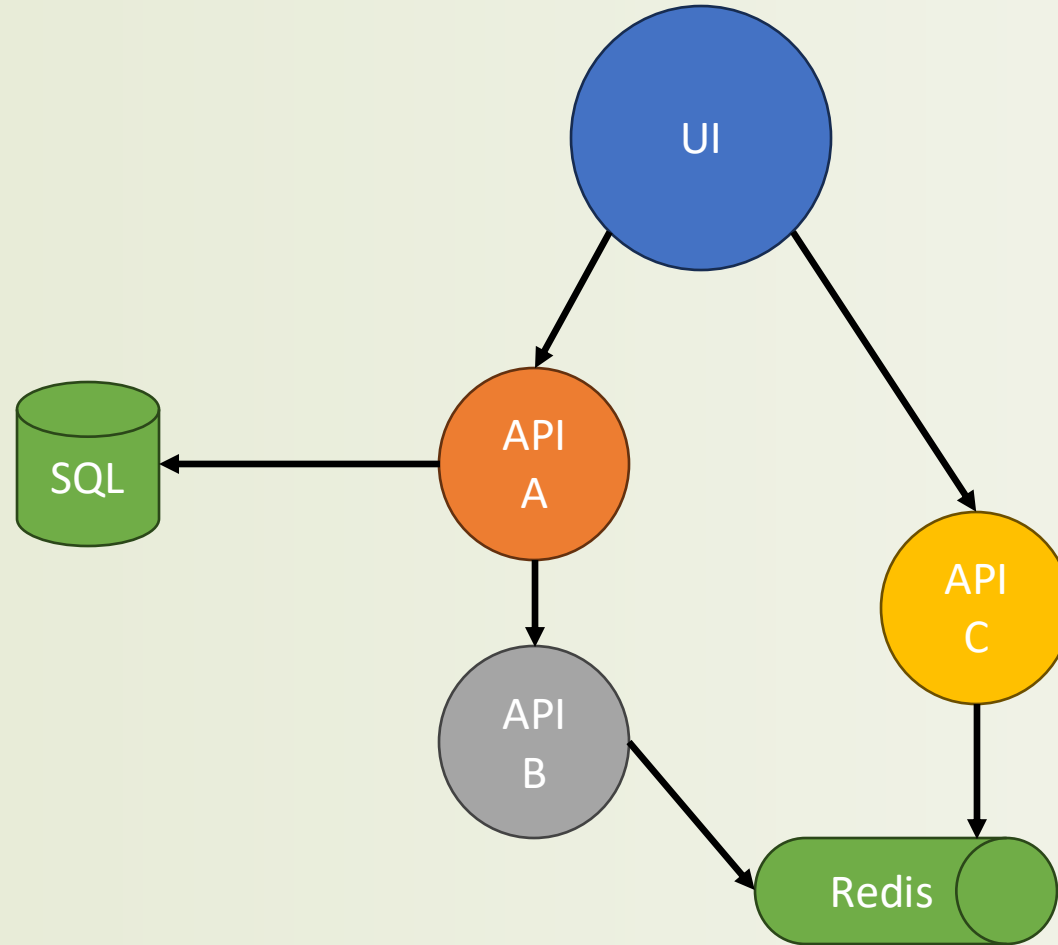
- ...

Snapbean
Software AG

# .NET Aspire

- Cloud-ready Stack für verteilte Anwendungen

- Übernimmt Orchestrierung von Projekten und Services

- Einbindung via NuGet Pakete

# .NET Aspire

# .NET Aspire

Demo Time !

Snapbean
Software AG

# .NET Aspire
## Deployment

- Aspire Manifest (JSON)
  - Beschreibt Apps, Services und Verbindungen

- Deployment nach Azure
  - Azure Container Apps

- Deployment nach Kubernetes
  - .NET Aspire JSON -> Kubernetes YAML

Snapbean
Software AG

# .NET Aspire
## Deployment

```
dotnet run --project path-to-project --publisher manifest --output-path aspire-manifest.json
```

```json
{
  "resources": {
    "postgres": {
      "type": "container.v0",
      "connectionString": "Host={postgres.bindings.tcp.host};Port={postgres.bindings.tcp.port};Username=postgres;Password={postgres-password.value}",
      "image": "docker.io/library/postgres:16.2",
      "env": {
        ...
        "POSTGRES_USER": "postgres",
        "POSTGRES_PASSWORD": "{postgres-password.value}"
      },
      ...
    },
    "db": {
      "type": "value.v0",
      "connectionString": "{postgres.connectionString};Database=db"
    },
    "api": {
      "type": "project.v0",
      "path": "../AspireDemo.Api/AspireDemo.Api.csproj",
      "env": {
        ...
        "ConnectionStrings__db": "{db.connectionString}"
      },
      ...
    },
    "web": {
      "type": "project.v0",
      "path": "../AspireDemo.Web/AspireDemo.Web.csproj",
      "env": {
        ...
        "services__api__http__0": "{api.bindings.http.url}",
        "services__api__https__0": "{api.bindings.https.url}"
      },
      ...
    },
    "postgres-password": {
      "type": "parameter.v0",
      "value": "{postgres-password.inputs.value}",
      "inputs": {
        "value": {
          "type": "string",
          "secret": true,
          "default": {
            "generate": {
              "minLength": 22
            }
          }
        }
      }
    }
  }
}
```

Snapbean
Software AG

# .NET Aspire

- Build ohne Workload

- Testing, testing, testing …

Snapbean
Software AG

# Vielen Dank 🙏

Ralph Bodmer
Manu Sidler