JavaScript Interop in Blazor Applications

UNDERSTANDING THE NEED TO USE JAVASCRIPT IN BLAZOR

Thomas Claudius Huber
SOFTWARE DEVELOPER

@thomasclaudiush   www.thomasclaudiushuber.com

Module Outline

Why use JavaScript in your Blazor app?

- Supported interop scenarios

The Blazor app used in this course

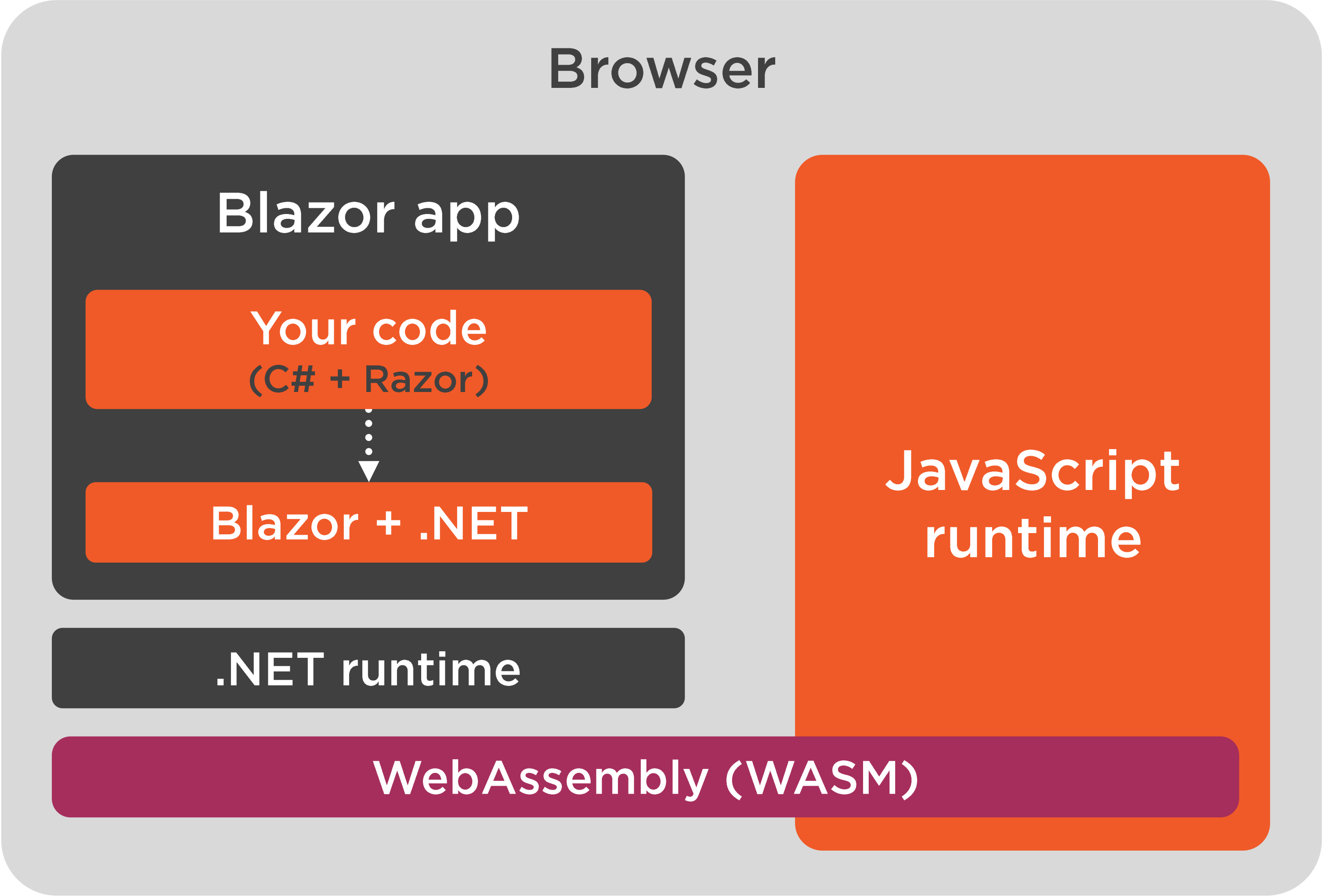
Course outline

Why Use JavaScript in Your Blazor App?

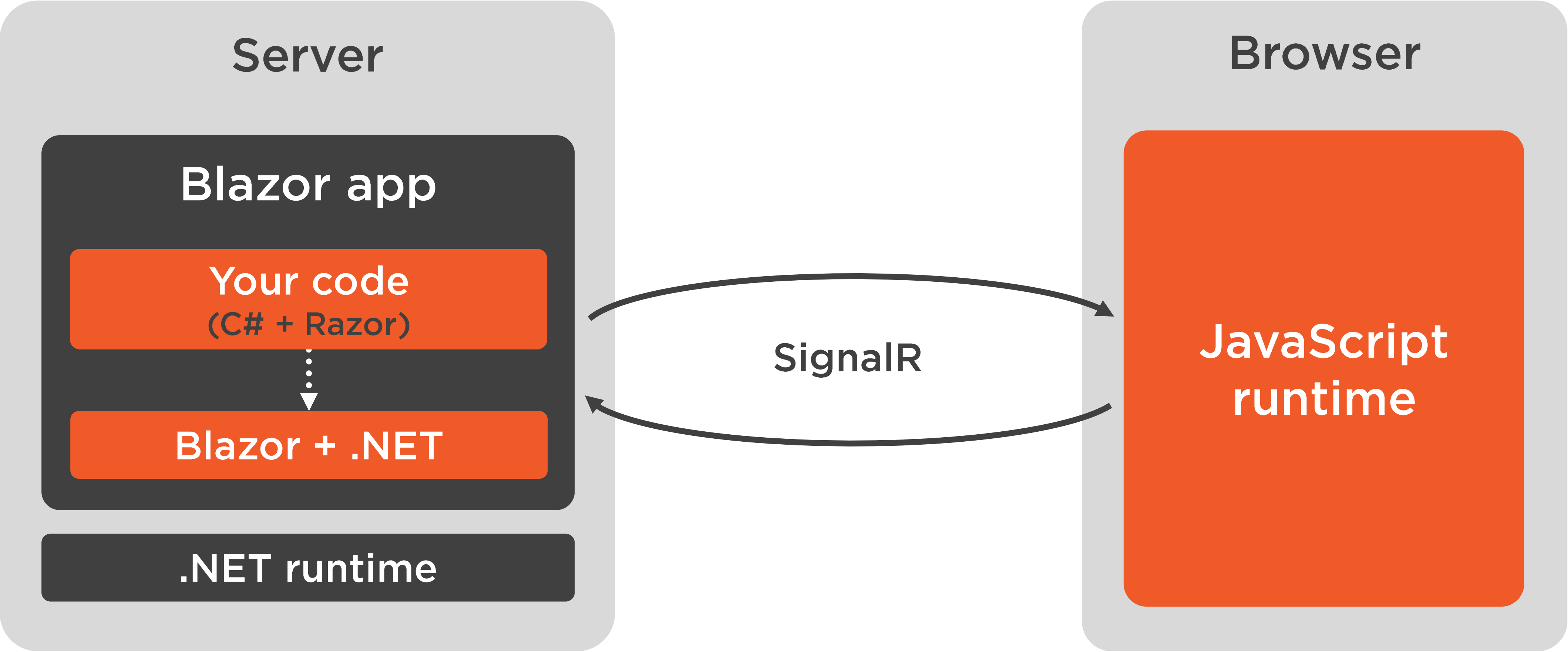
Hosting

WebAssembly

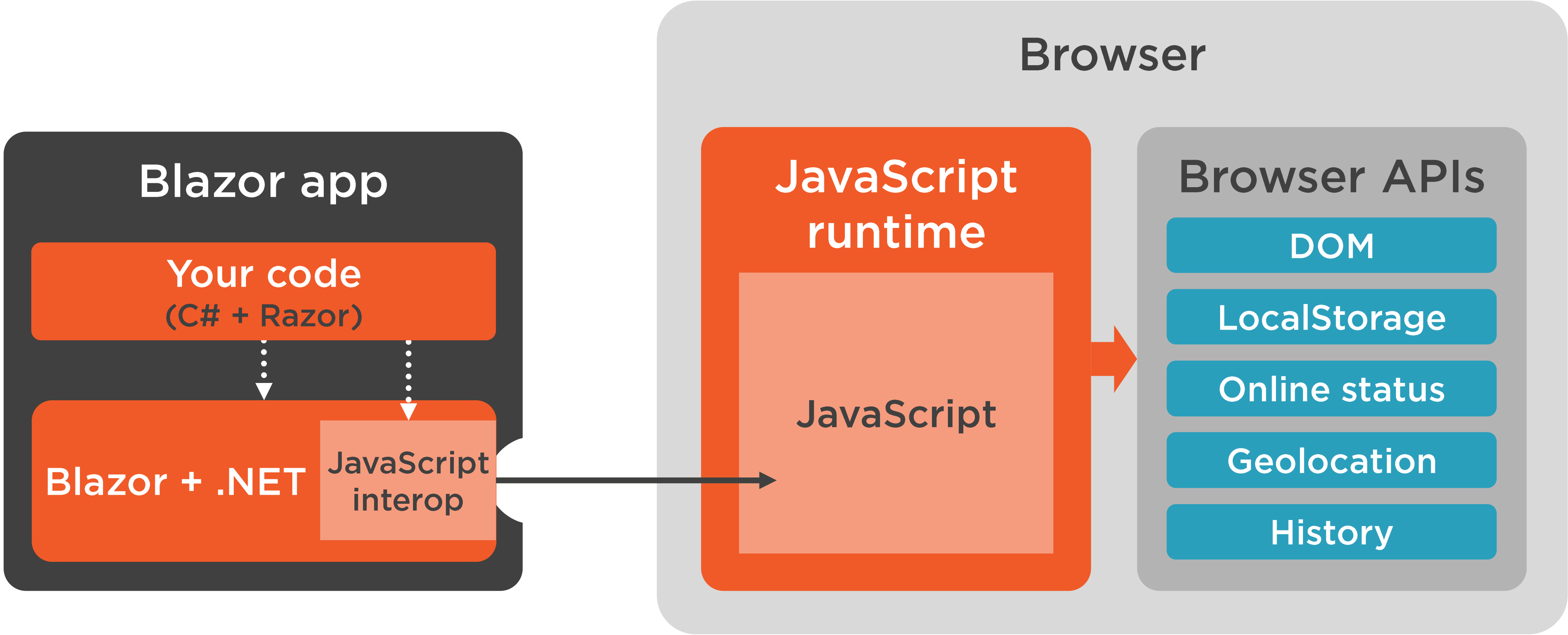
Server



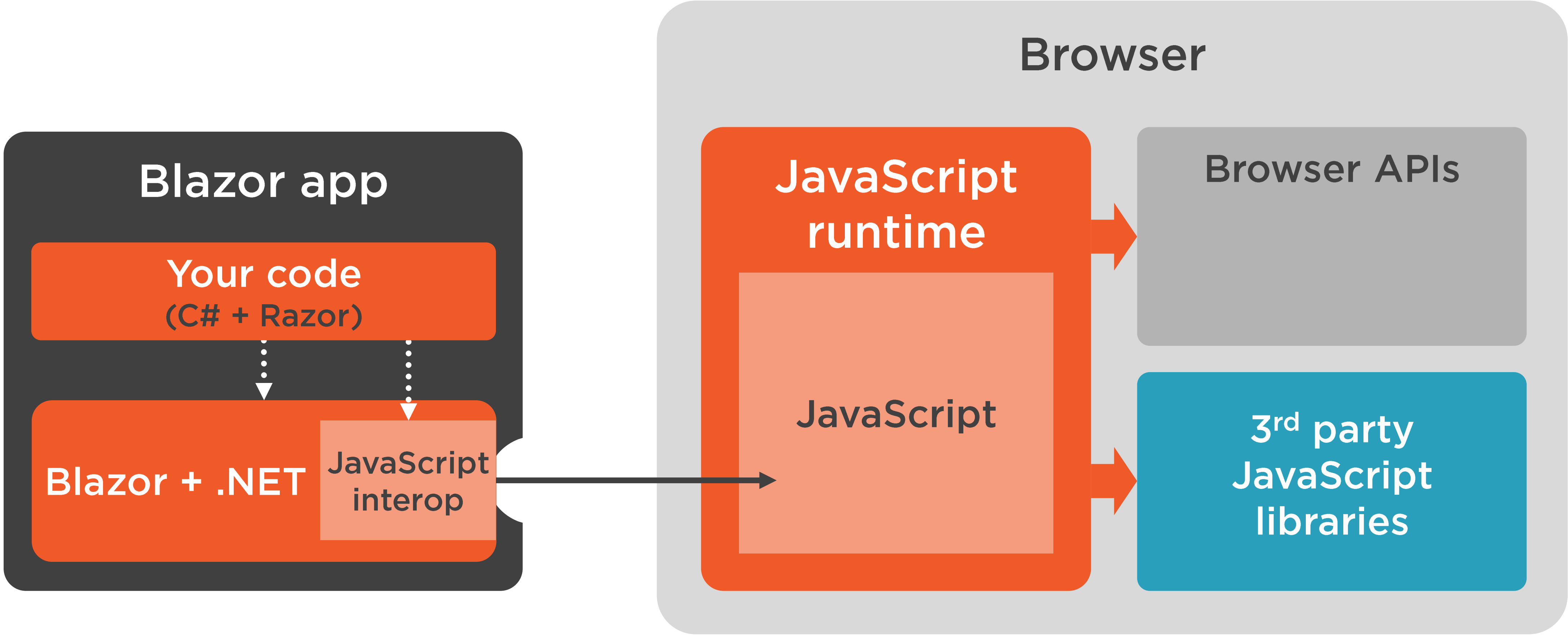
Why Use JavaScript in Your Blazor App?



Why Use JavaScript in Your Blazor App?



Why Use JavaScript in Your Blazor App?



Why Use JavaScript in Your Blazor App?

Browser APIs

3rd party JavaScript libraries

The JavaScript ecosystem is huge

JavaScript Interop Scenarios

Invoke JavaScript functions from .NET

Invoke .NET methods from JavaScript

Wrap JavaScript interop code

Demo

Look at the Blazor app used in this course

Course Outline

Understanding the Need to Use JavaScript in Blazor

Invoking JavaScript Functions from .NET

Invoking .NET Methods from JavaScript

Integrating Browser APIs in Your Blazor App

Using JavaScript Interop in Razor Class Libraries

Wrapping JavaScript Components in .NET

Invoking JavaScript Functions from .NET

Thomas Claudius Huber

SOFTWARE DEVELOPER

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline

Add a JavaScript file to your Blazor app

Invoke JavaScript functions

- Invoke functions returning void/value
- Pass a .NET object
- Pass an HTML element reference

Use JavaScript in component lifecycle

Handle JavaScript errors in .NET

Work with JavaScript modules

Demo

Add a JavaScript file to your Blazor app

Demo

Create a JavaScript function in the global scope

Demo

Call a JavaScript function that returns void

Demo

Pass a .NET object
to a JavaScript function

Demo

Call a JavaScript function that returns a value

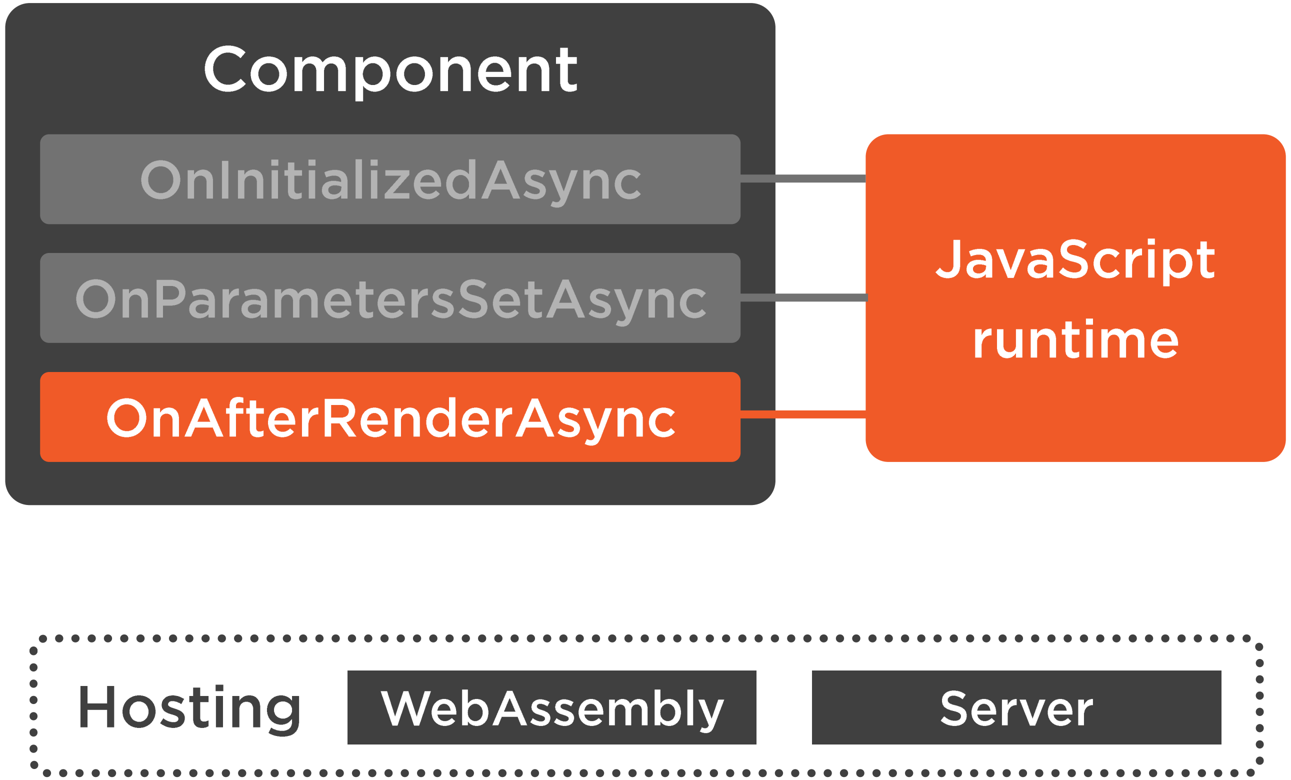
Demo

Get an object as a return value

Demo

Pass an HTML element reference to a JavaScript function

JavaScript Interop in the Component Lifecycle

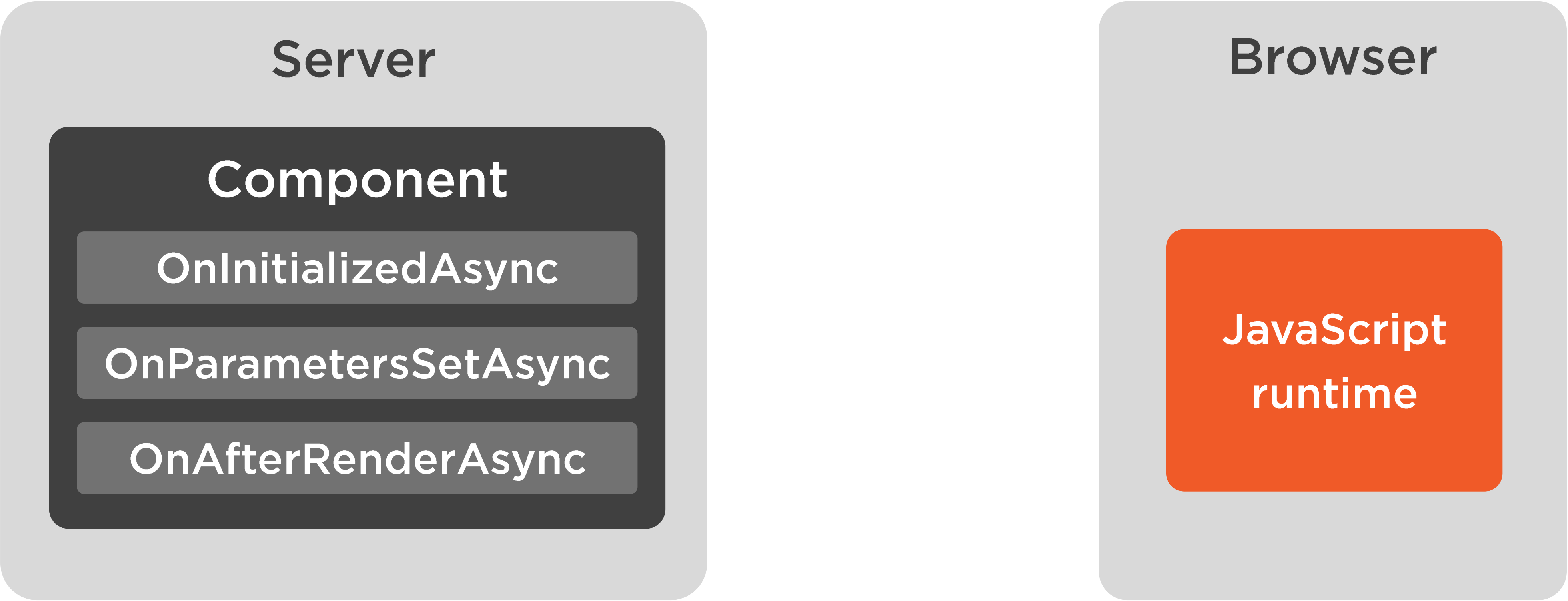


```
<app>
   <component type="typeof(App)"
     render-mode="ServerPrerendered" />
</app>
```

_Host.cshtml File

Contains render mode of a Blazor Server app

JavaScript Interop in the Component Lifecycle



Server

Component

OnInitializedAsync

OnParametersSetAsync

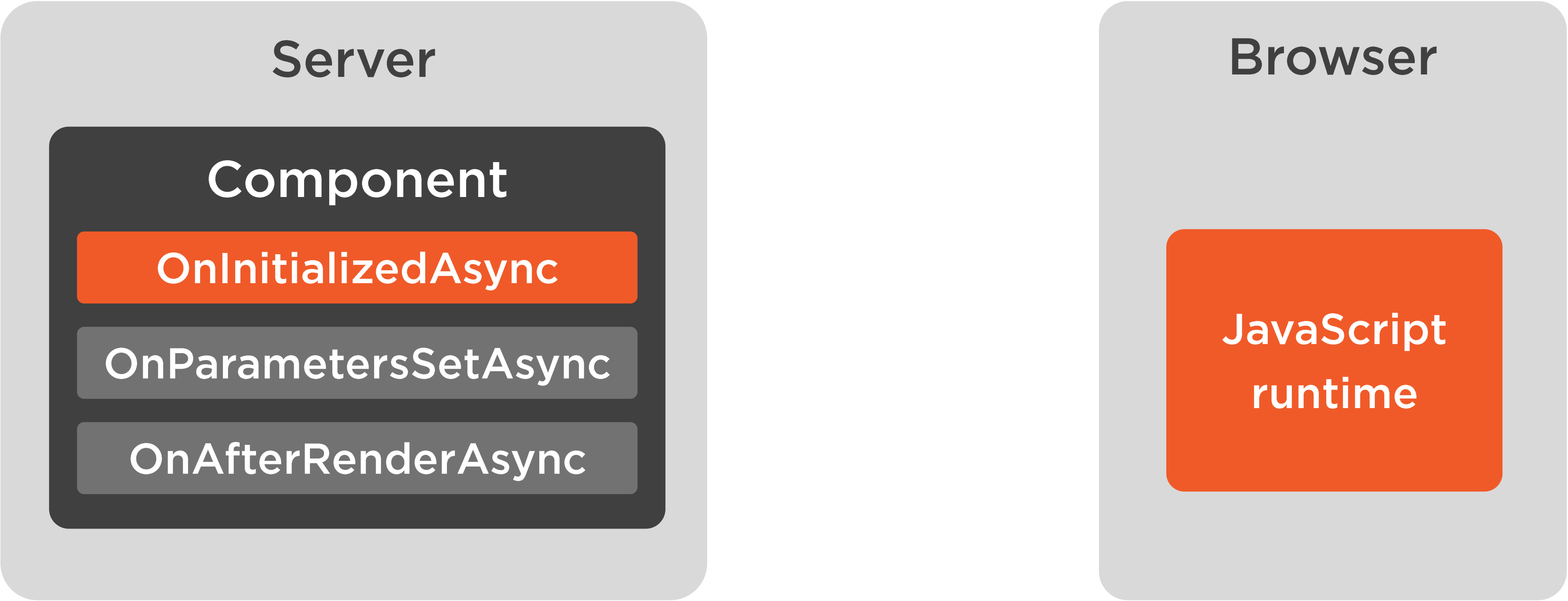
OnAfterRenderAsync

Browser

JavaScript runtime

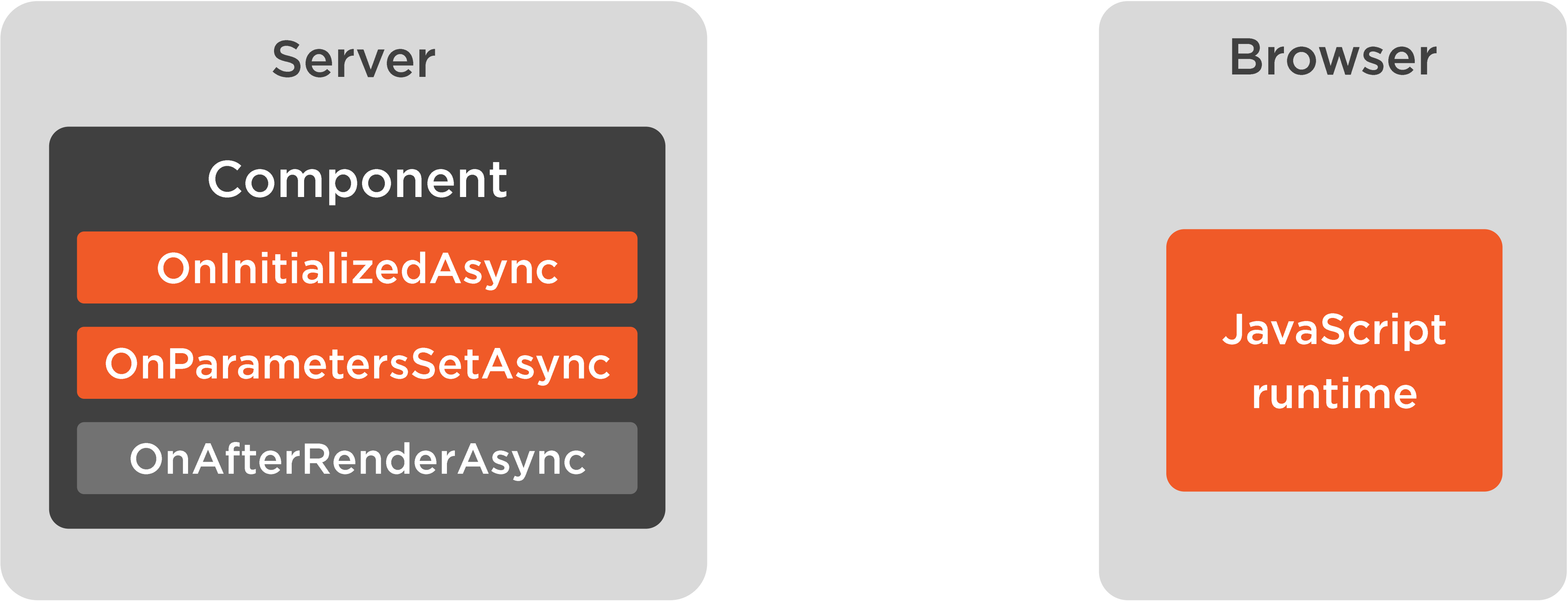
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



Server

Component

OnInitializedAsync

OnParametersSetAsync

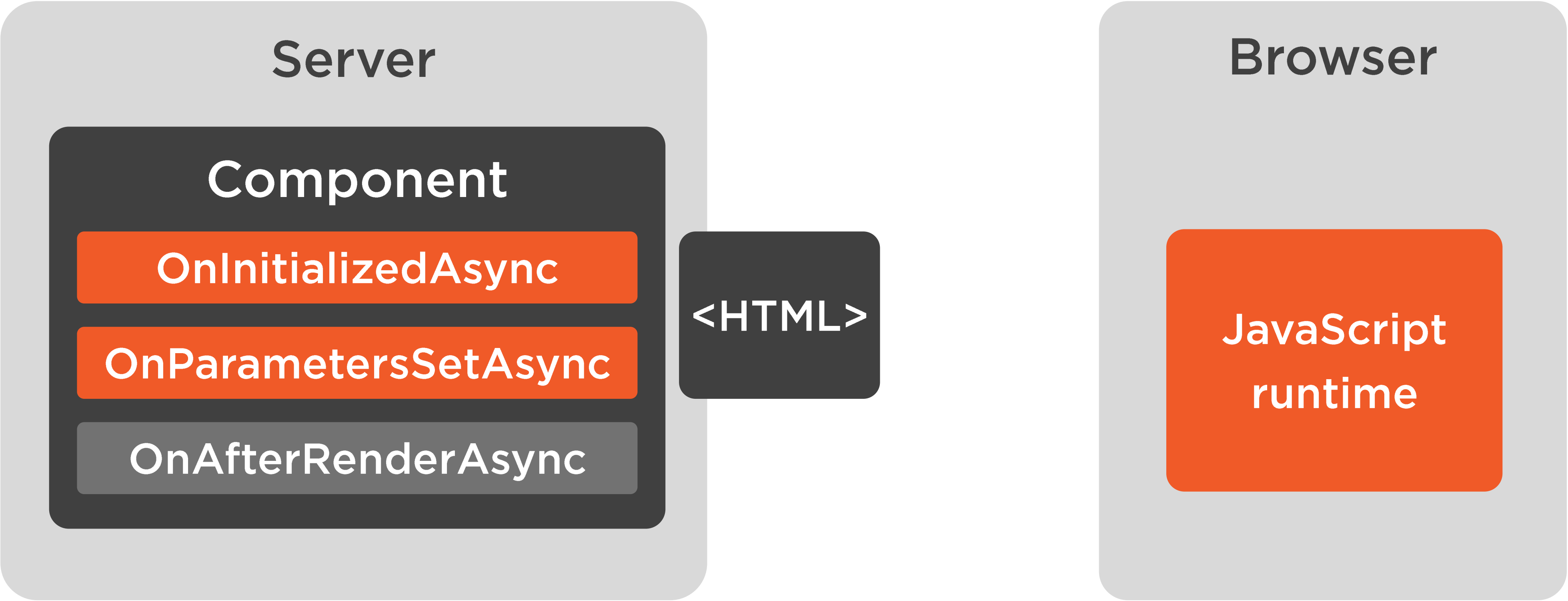
OnAfterRenderAsync

Browser

JavaScript runtime

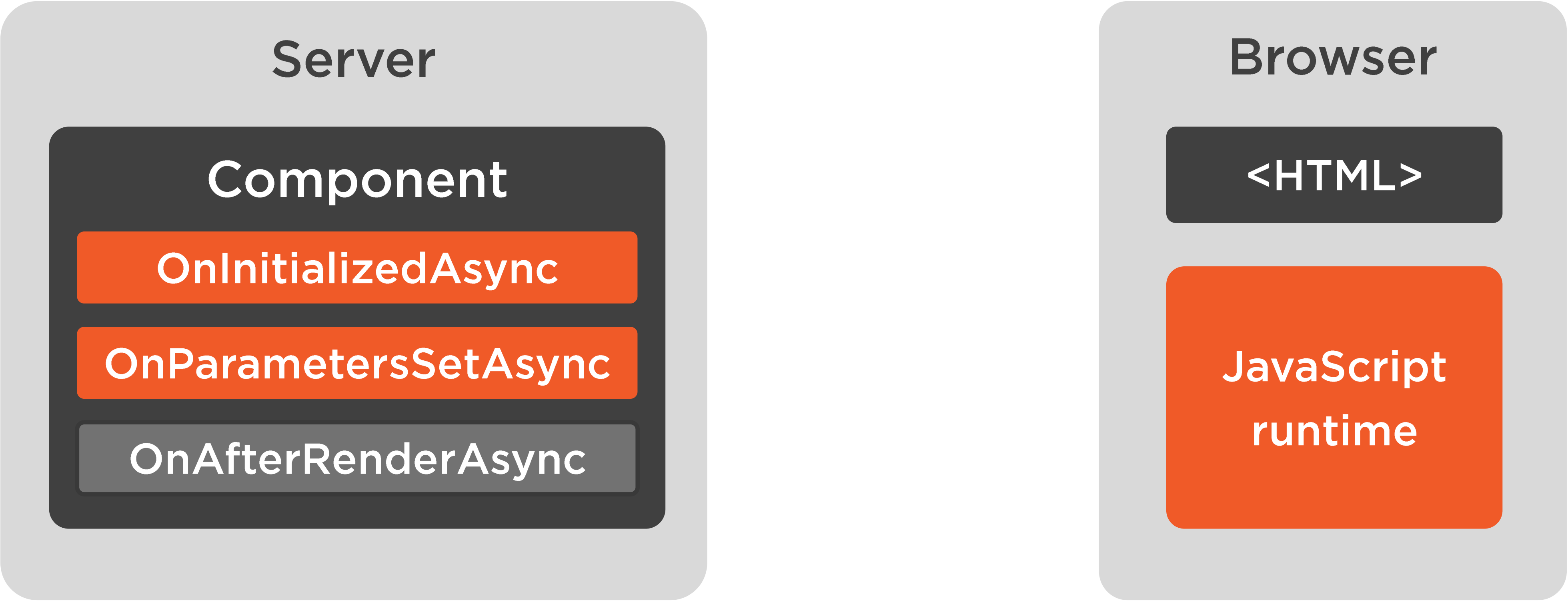
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



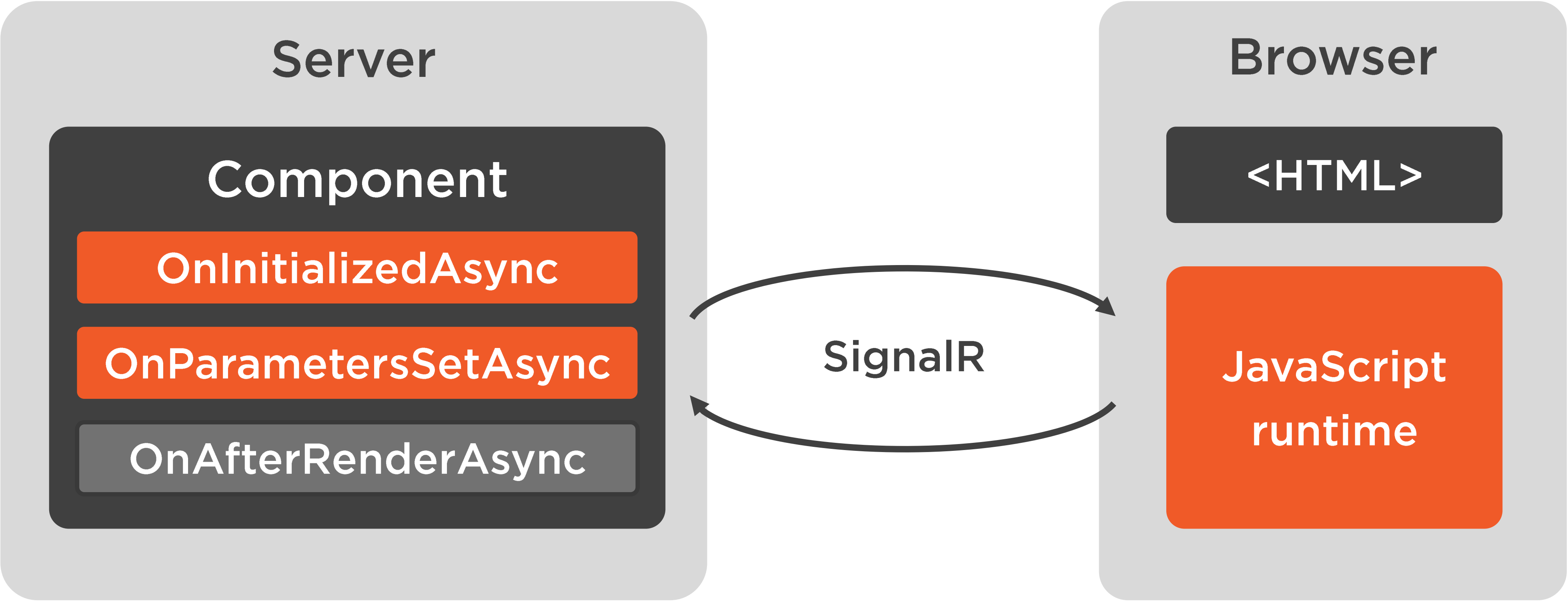
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



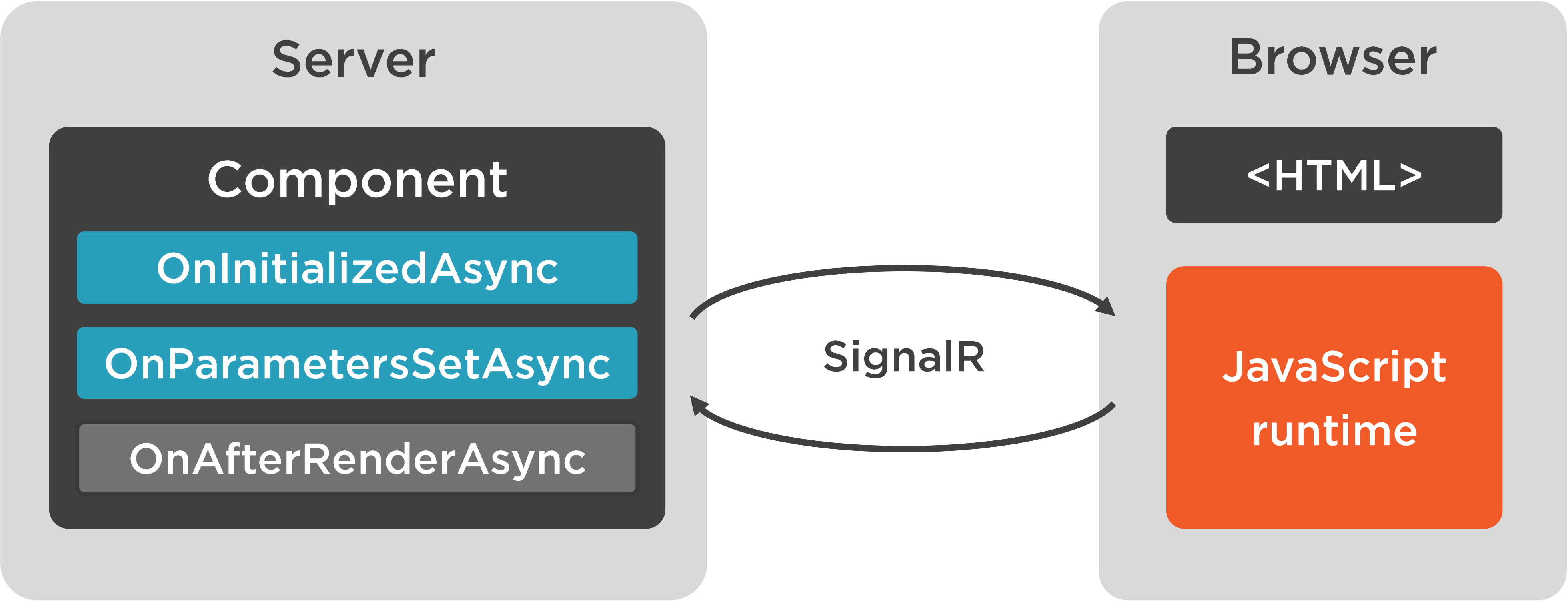
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



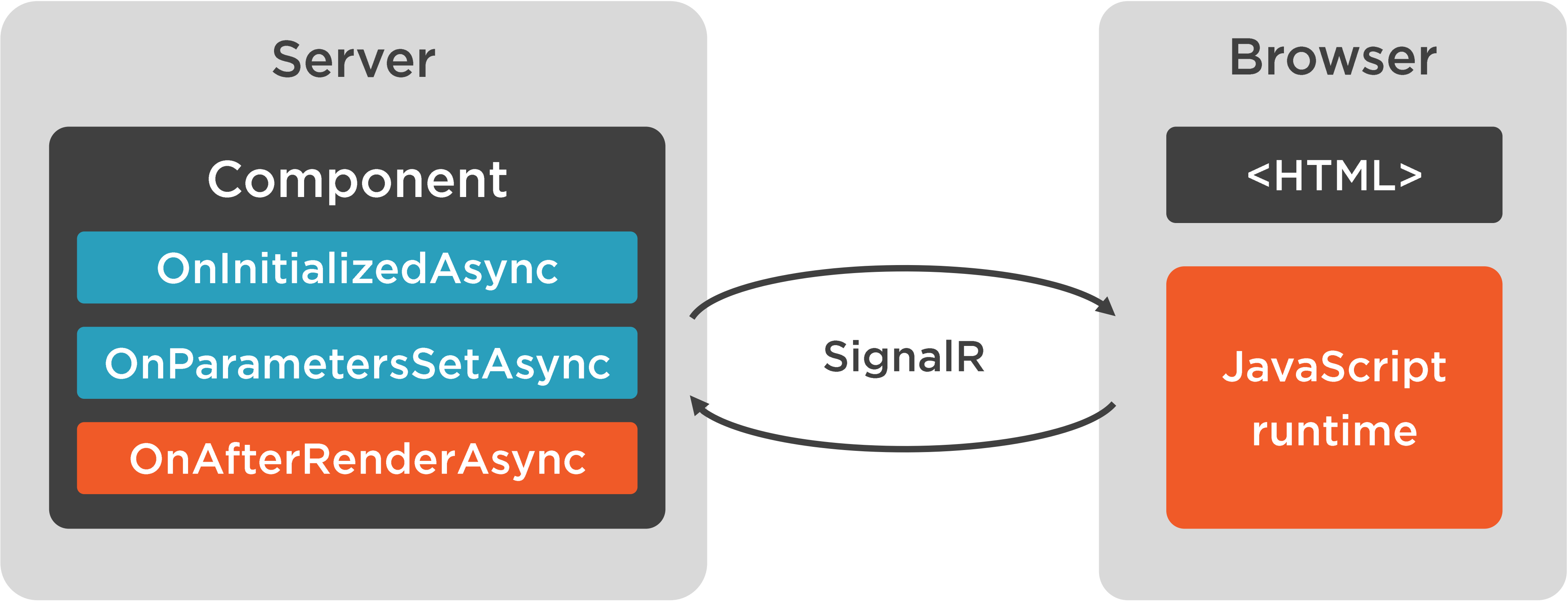
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



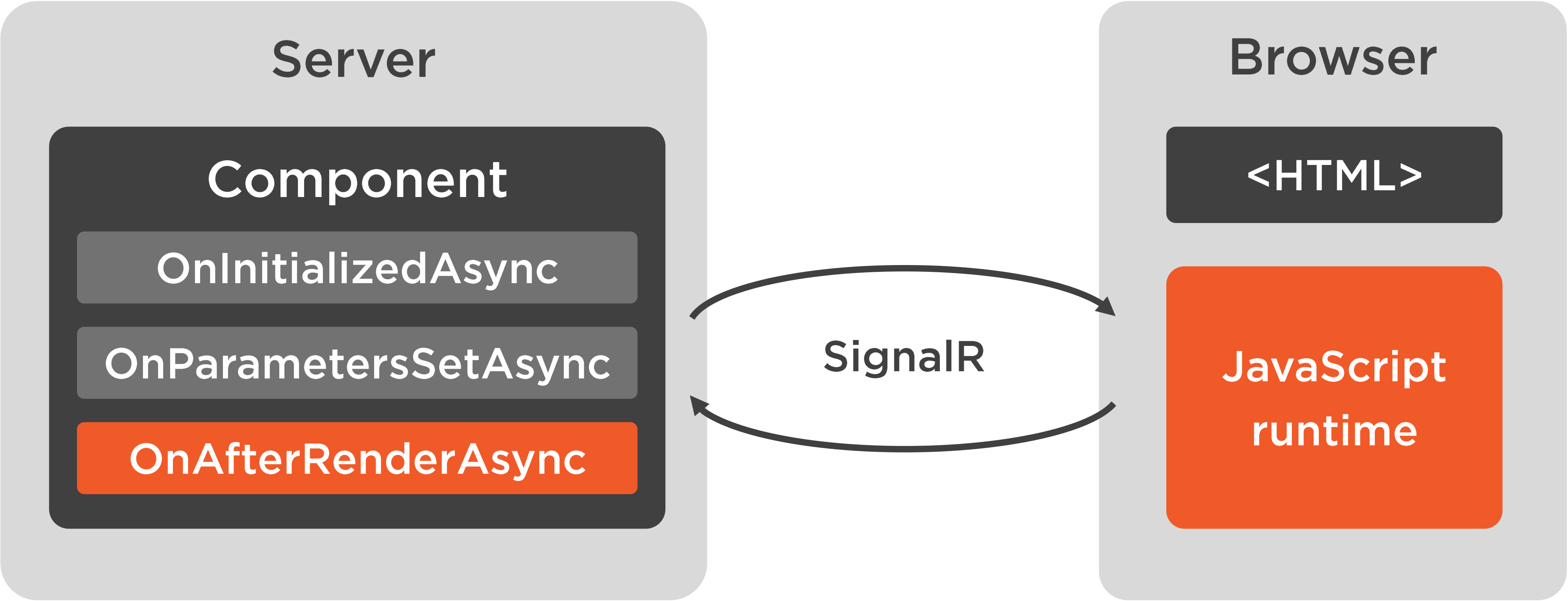
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



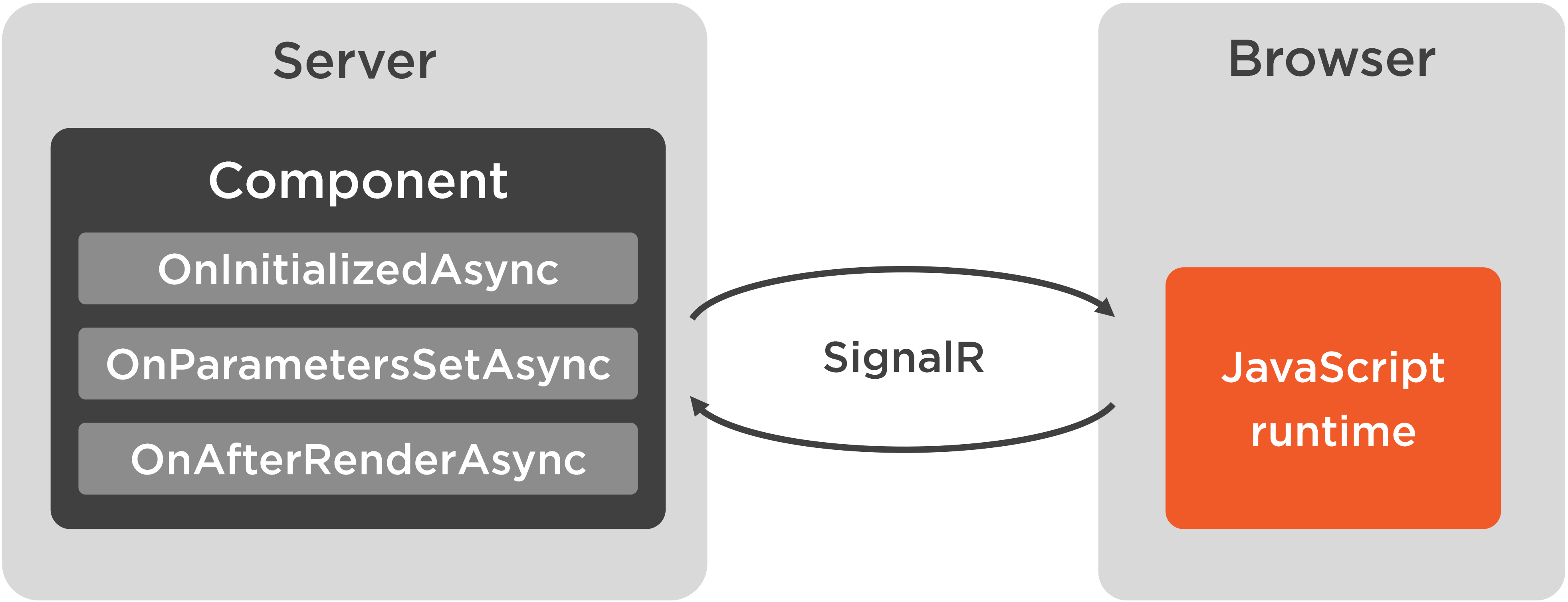
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



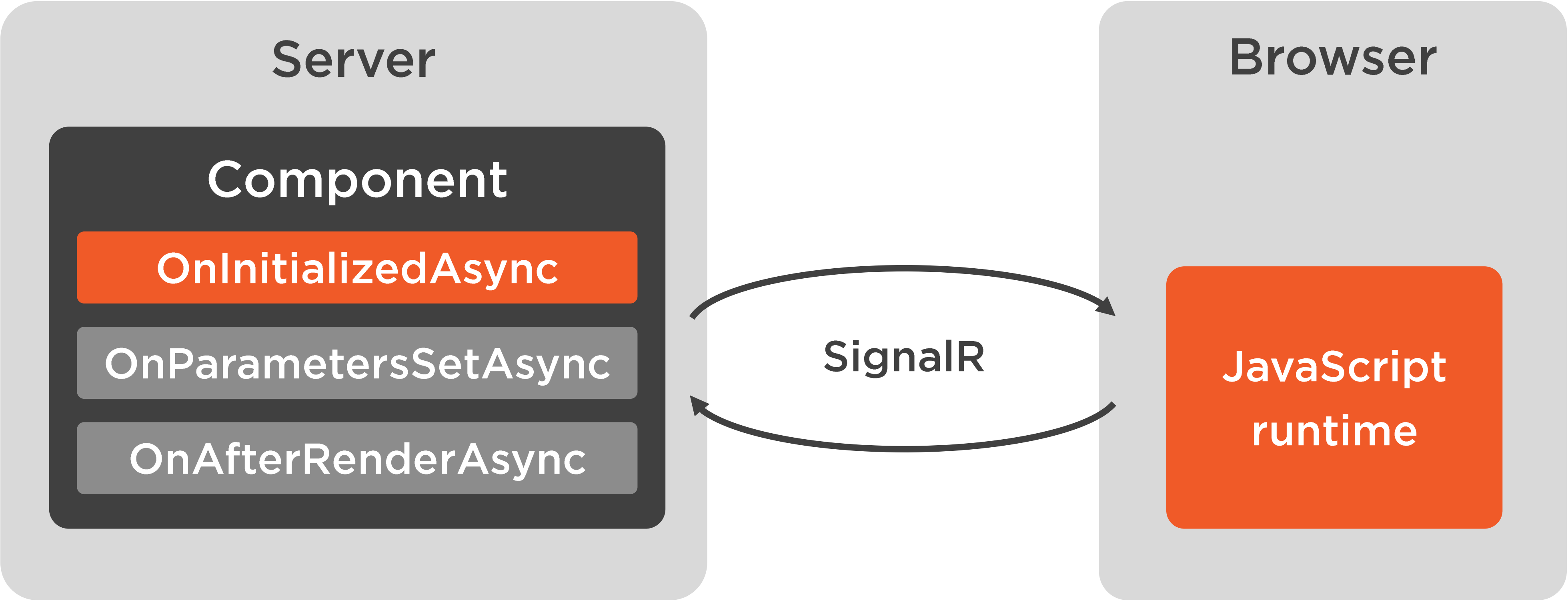
```
render-mode="ServerPrerendered"
```

JavaScript Interop in the Component Lifecycle



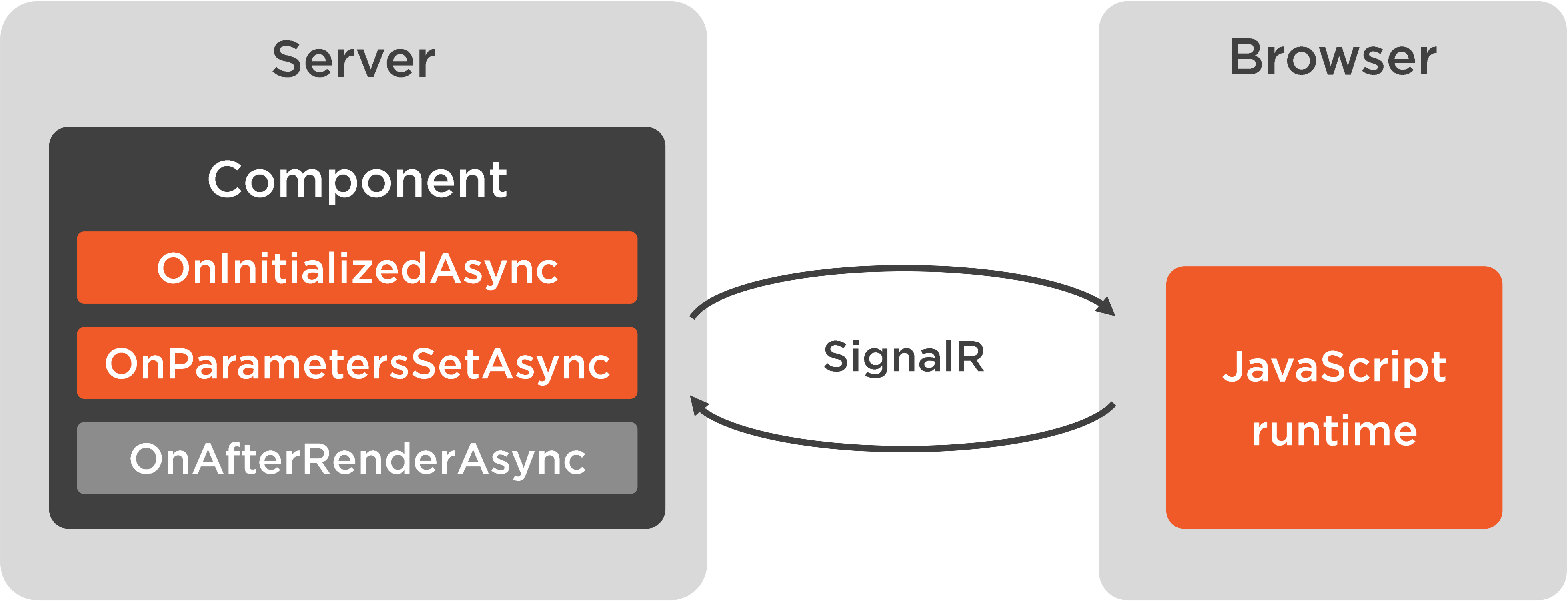
```
render-mode="Server"
```

JavaScript Interop in the Component Lifecycle



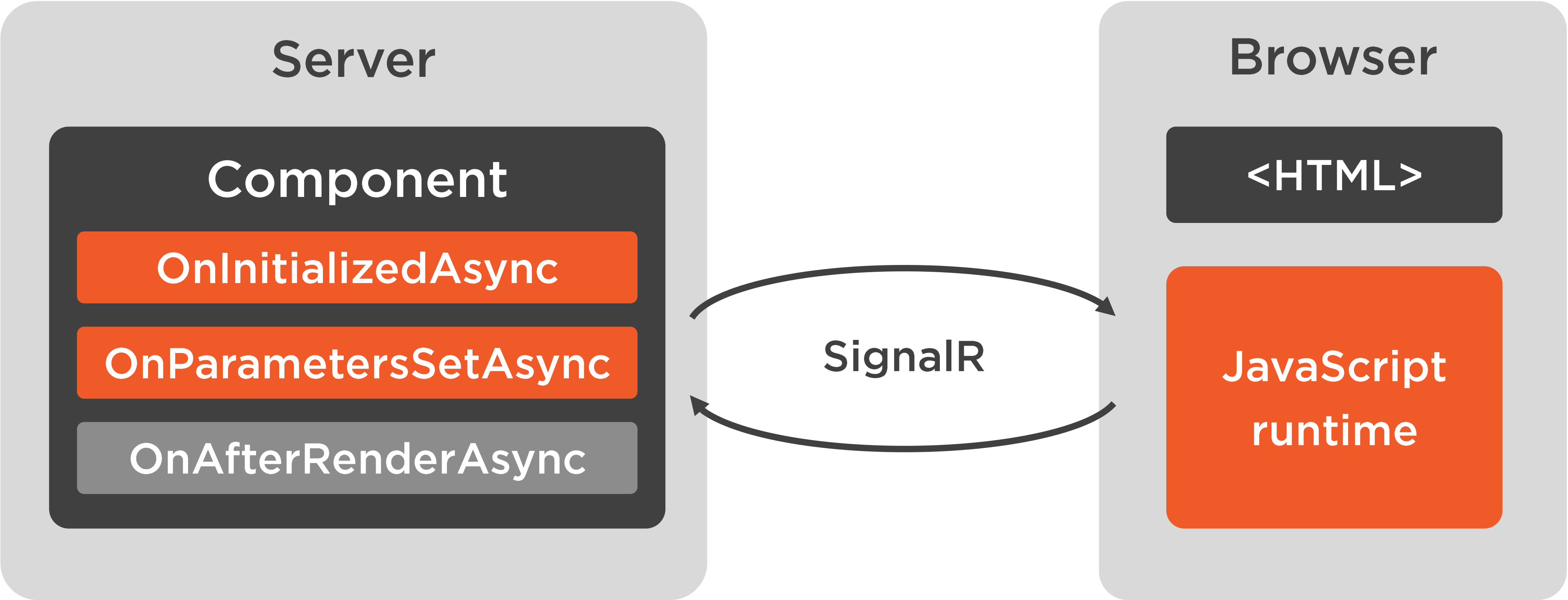
```
render-mode="Server"
```

JavaScript Interop in the Component Lifecycle



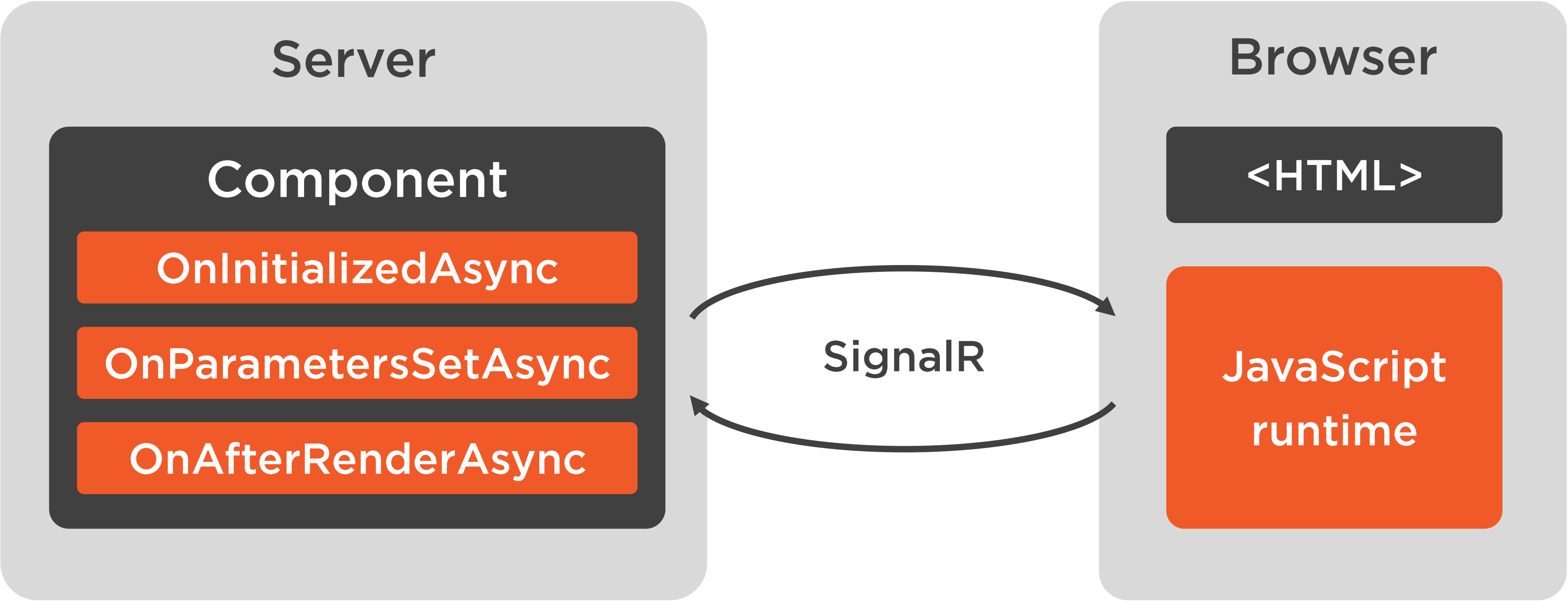
```
render-mode="Server"
```

JavaScript Interop in the Component Lifecycle



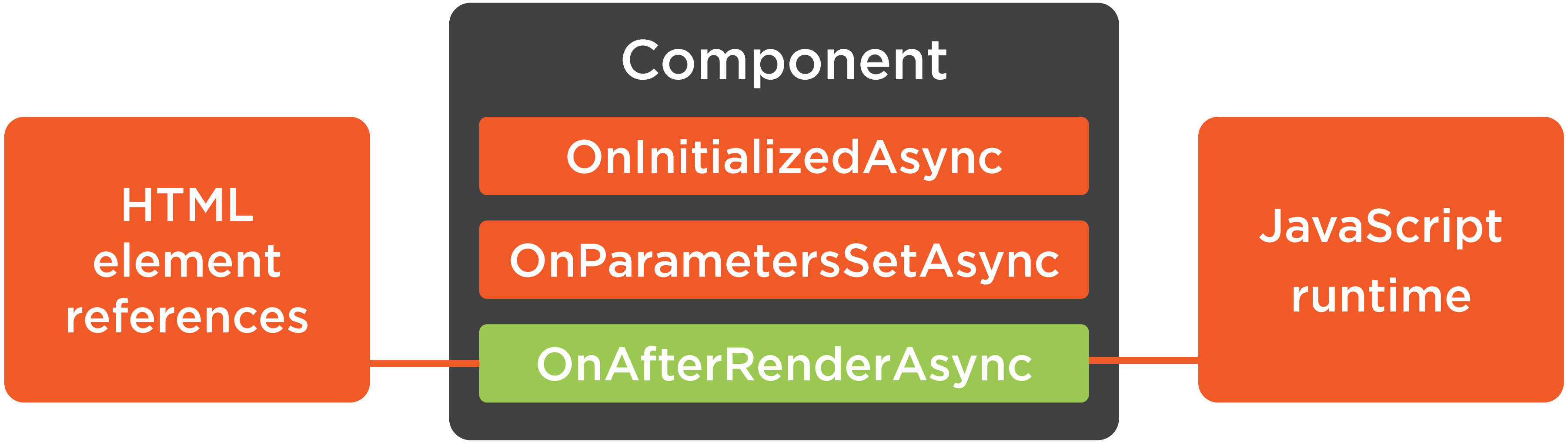
```
render-mode="Server"
```

JavaScript Interop in the Component Lifecycle



```
render-mode="Server"
```

JavaScript Interop in the Component Lifecycle



Demo

See server prerendering in action

- Look at the prerendered HTML code

Demo

Use JavaScript interop in the component lifecycle

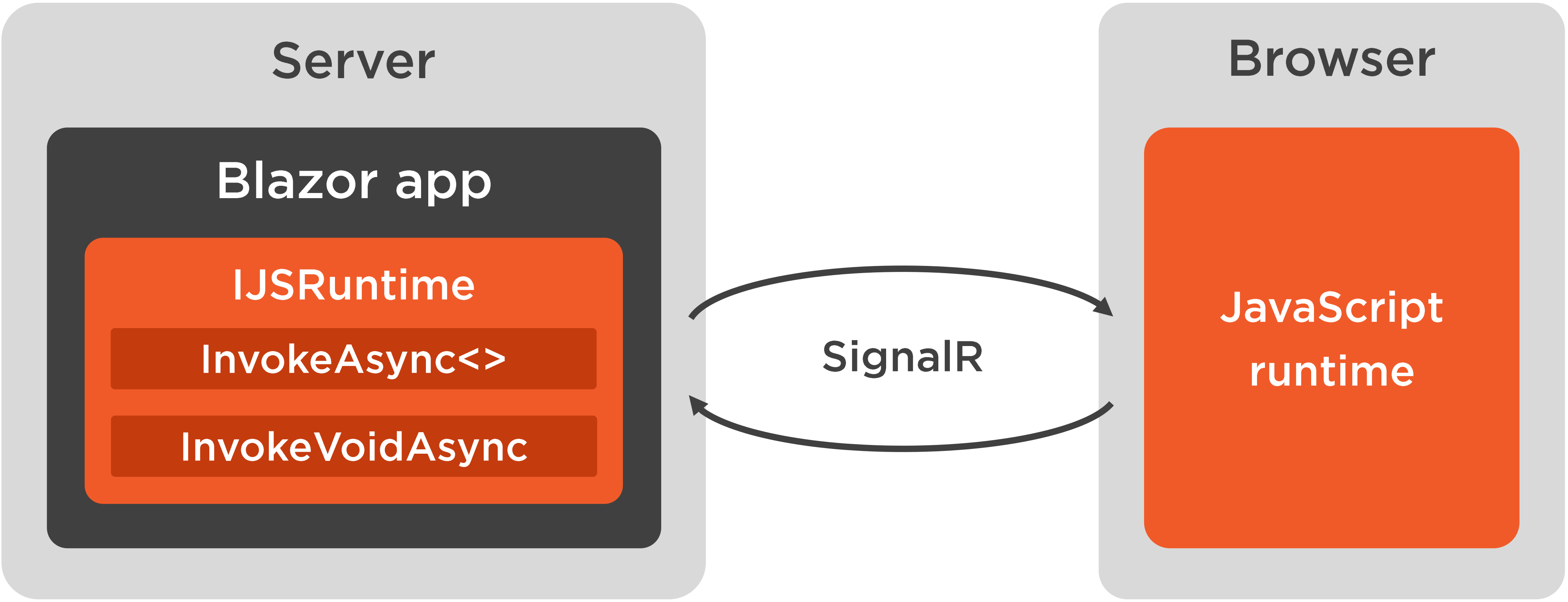
Demo

Focus a Blazor component via JavaScript

Demo

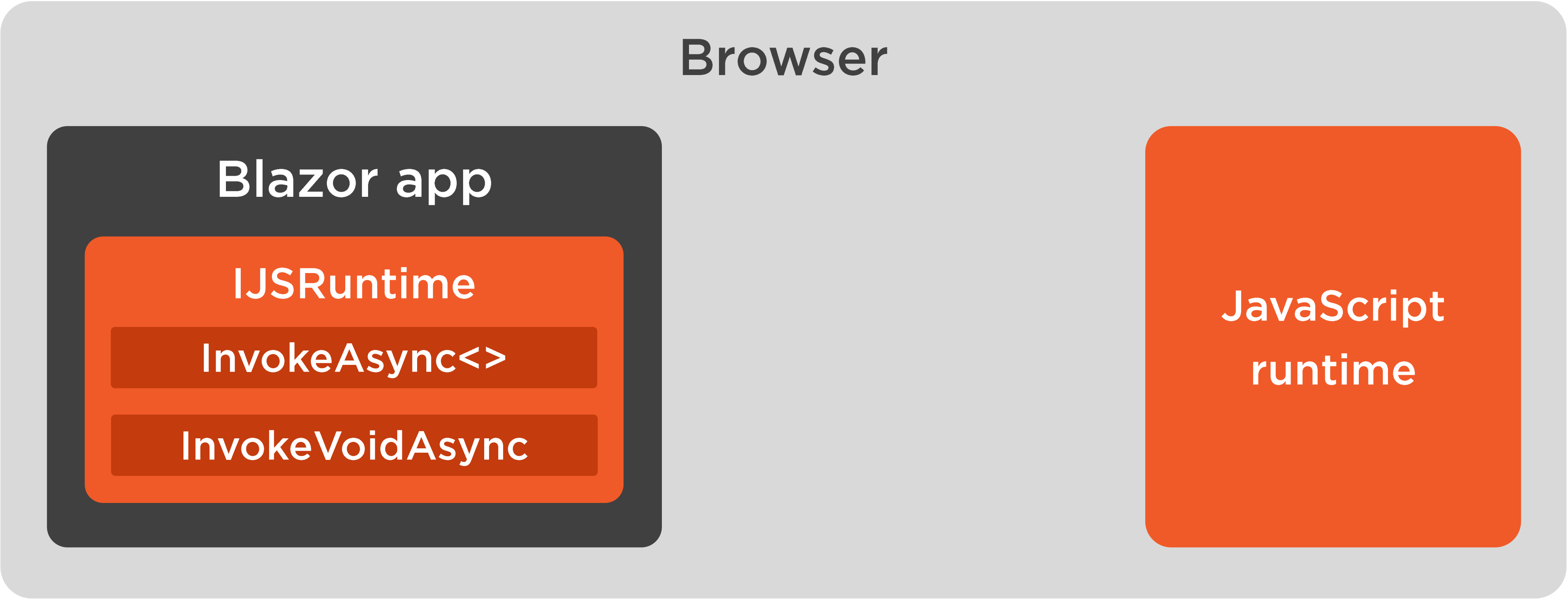
Handle JavaScript errors in .NET

Invoke JavaScript Functions Synchronously



Hosting WebAssembly **Server**

Invoke JavaScript Functions Synchronously



Hosting **WebAssembly** Server

```
@inject IJSRuntime JSRuntime



var jsInProcRuntime = (IJSInProcessRuntime)JSRuntime;
```

Invoke JavaScript Functions Synchronously

IJSInProcessRuntime

- InvokeAsync<>
- InvokeVoidAsync

✓ **Works with Blazor WebAssembly**

👎 **Works NOT with Blazor Server**

Work with JavaScript Modules

JavaScript file with exported functions

JavaScript code is isolated

No global scope pollution

JavaScript file is loaded only when needed

You don't have to create a *<script>* tag

Demo

Work with JavaScript modules

Summary

Inject IJSRuntime into your component

- Invoke functions accessible via the window object

Pass objects to JavaScript

- Simple values
- .NET objects
- HTML element references

Use JavaScript in component lifecycle

- OnAfterRenderAsync

Handle JavaScript errors in .NET

Work with JavaScript modules

Invoking .NET Methods from JavaScript

Thomas Claudius Huber

SOFTWARE DEVELOPER

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline

Call a static .NET method

Call a .NET instance method

- Call a component method from a JavaScript event handler

Demo

Call a static .NET method from JavaScript

Demo

Use a custom method identifier

Demo

Call a .NET instance method from JavaScript

Demo

Call a component method from a JavaScript event handler

Summary

Call static .NET methods

Call .NET instance methods

Integrating Browser APIs in Your Blazor App

Thomas Claudius Huber

SOFTWARE DEVELOPER

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline

Access the browser's local storage

- Know the plan for local storage
- Look at the prepared C# code
- Store data in the local storage

Check if the browser is online

Demo

Know the plan for local storage

Demo

Look at the prepared C# code

Demo

Store data in the local storage

Demo

Check if the browser is online

Summary

Use JavaScript interop to integrate browser APIs

Access the browser's local storage

Check if the browser is online

Using JavaScript Interop in Razor Class Libraries

Thomas Claudius Huber
SOFTWARE DEVELOPER

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline

Move a Blazor component with JavaScript to a Razor class library

Use the Blazor component in your Blazor app

Look at the Map component and its Razor class library

Demo

Move a Blazor component with JavaScript to a Razor class library

- Create and explore a Razor class library
- Move a Blazor component to the library

Demo

Use the Blazor component in your Blazor app

Demo

Look at the Map component and its Razor class library

Summary

Create a Razor class library

- Move a Blazor component with JavaScript to the library
- Use the Blazor component in your Blazor app

Look at the Map component and its Razor class library

Wrapping JavaScript Components in .NET

Thomas Claudius Huber
SOFTWARE DEVELOPER

@thomasclaudiush   www.thomasclaudiushuber.com

Module Outline

Understand the scenario to use a data grid

Look at a JavaScript sample that uses the Ag-Grid

Create a BlazorAgGrid component

- Wrap the JavaScript Ag-Grid
- Add a RowData parameter
- Add an OnSelectionChanged event
- Auto-generate columns
- Create a BlazorAgGridColumn component

Demo

Understand the scenario to use a data grid

Can you provide a sample that shows how to use the Ag-Grid in JavaScript?

Yes!

Demo

Look at a JavaScript sample that uses the Ag-Grid

Demo

Create a BlazorAgGrid component

Demo

Initialize the wrapped JavaScript component

Demo

Use the BlazorAgGrid in the Blazor app

Demo

Add a RowData component parameter

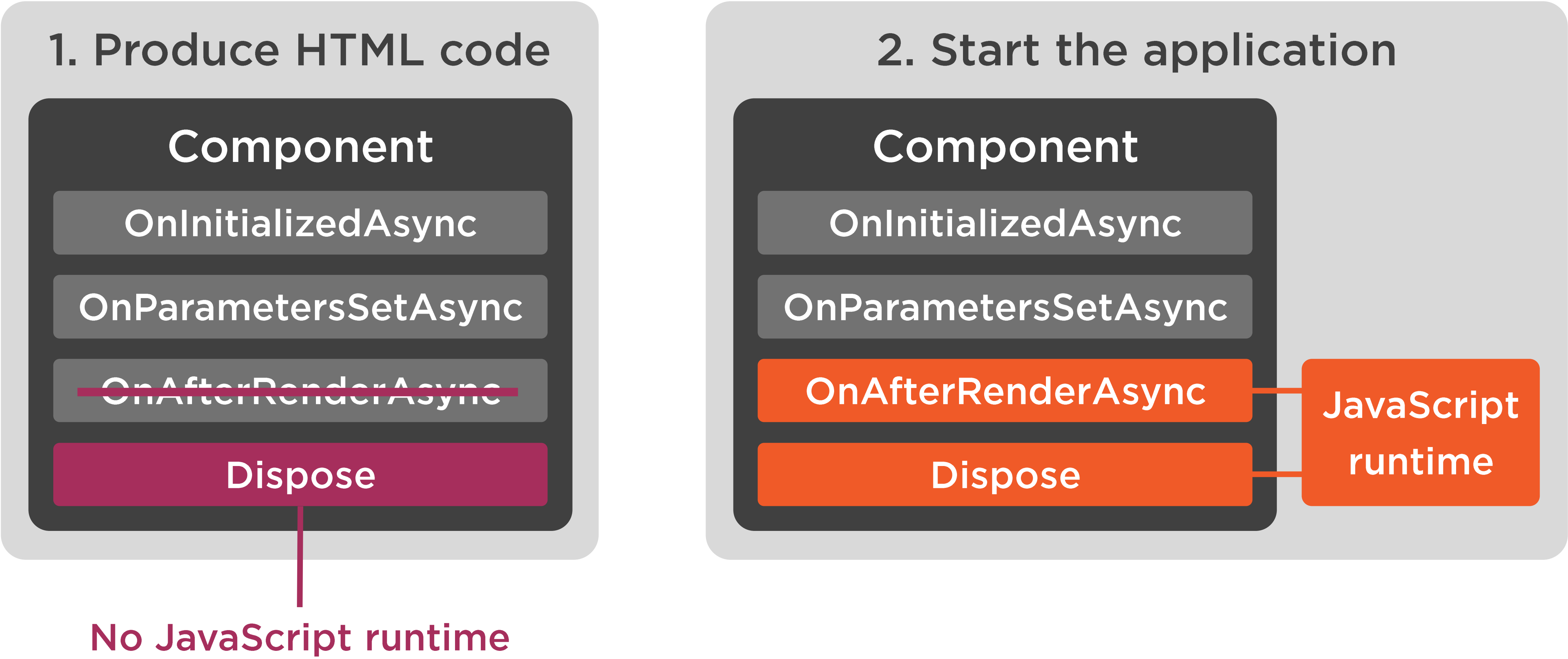
Demo

Support multiple BlazorAgGrids on a single page

Demo

Implement IDisposable to clean up

Understand Server Prerendering and Disposing



1. Produce HTML code

Component

OnInitializedAsync

OnParametersSetAsync

~~OnAfterRenderAsync~~

Dispose

No JavaScript runtime

2. Start the application

Component

OnInitializedAsync

OnParametersSetAsync

OnAfterRenderAsync

Dispose

JavaScript runtime

Demo

Understand server prerendering and disposing

Demo

Add an OnSelectionChanged event

Demo

Use the OnSelectionChanged event

Demo

Override the ShouldRender method

Demo

Pass column definitions from .NET to JavaScript

Demo

Add an AutoGenerateColumns component parameter

Demo

Create a BlazorAgGridColumn component

Summary

Wrap a JavaScript component in .NET

- Invoke JavaScript functions from .NET
- Invoke .NET methods from JavaScript
- Create BlazorAgGrid to wrap Ag-Grid

Create a Blazor component

- Component parameters
- Component events
- Cascading parameter
- Course: Creating Blazor Components by Roland Guijt

Congratulations!

Course Summary

Invoke JavaScript functions from .NET

Invoke .NET methods from JavaScript

Use JavaScript interop

- Integrate Browser APIs
- Wrap JavaScript components

Use JavaScript in Razor class libraries

JavaScript Interop in Blazor Applications

Thomas Claudius Huber

@thomasclaudiush
www.thomasclaudiushuber.com