Improve Blazor Performance and Stability

Alex Wolf

www.crywolfcode.com

Blazor Performance Enhancements in .NET 5.0

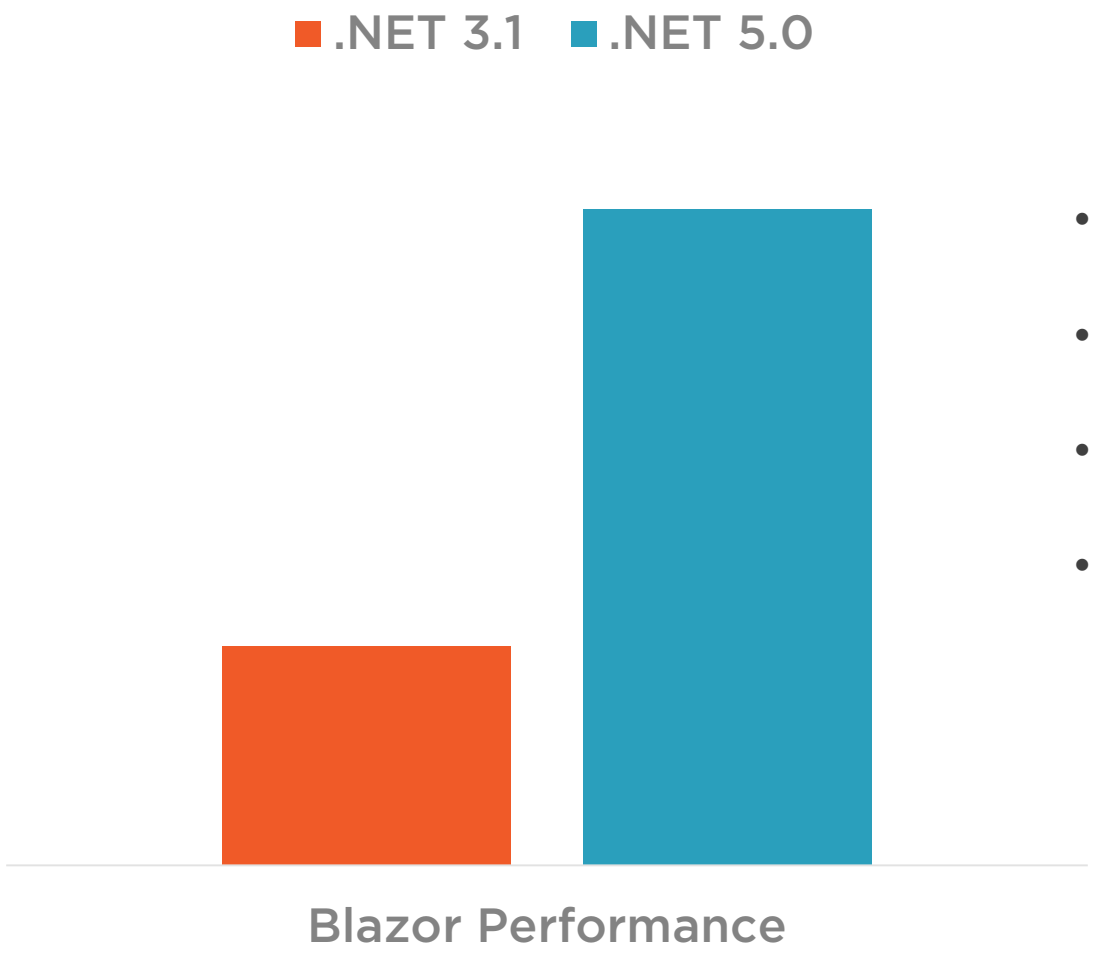
Platform Optimizations

Improvements to the underlying runtime and the Blazor source code

Performance Improving Features

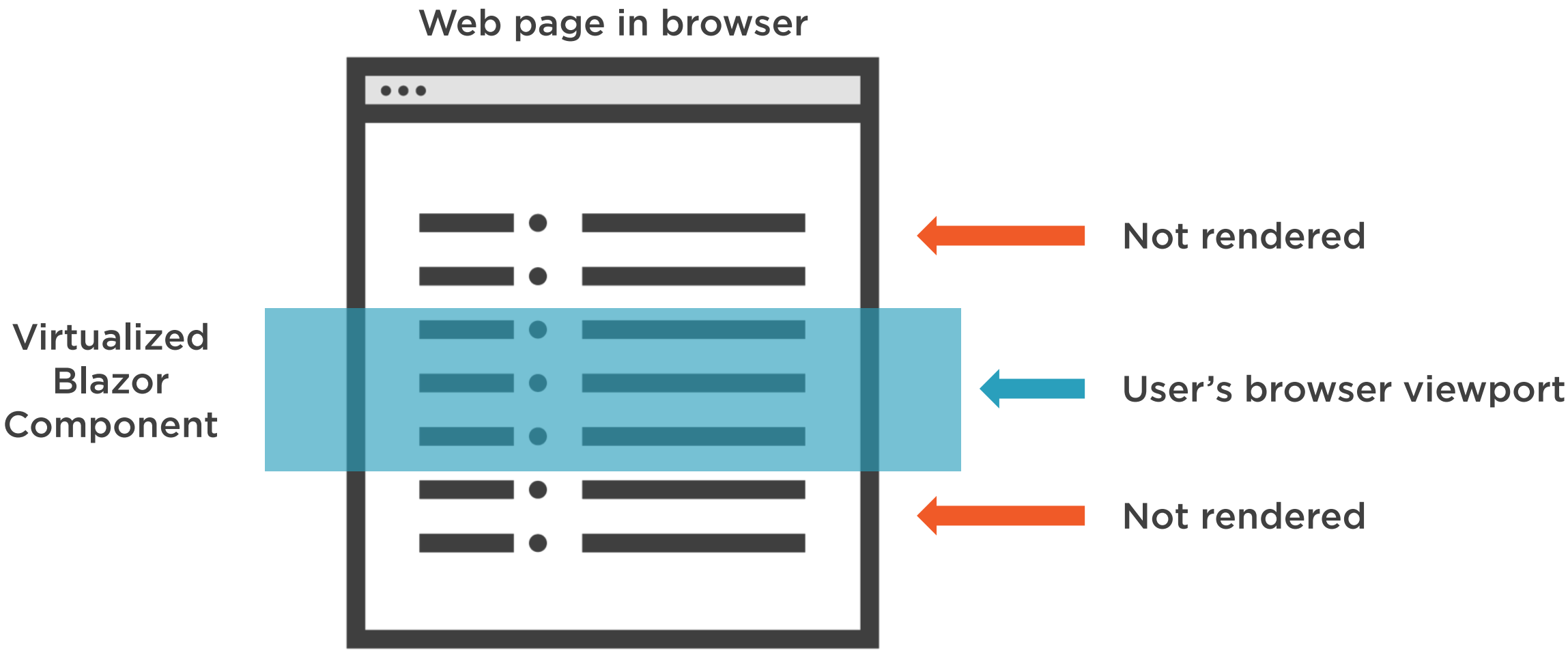
New features that can be implemented through custom code to improve performance

Internal Blazor Performance Enhancements

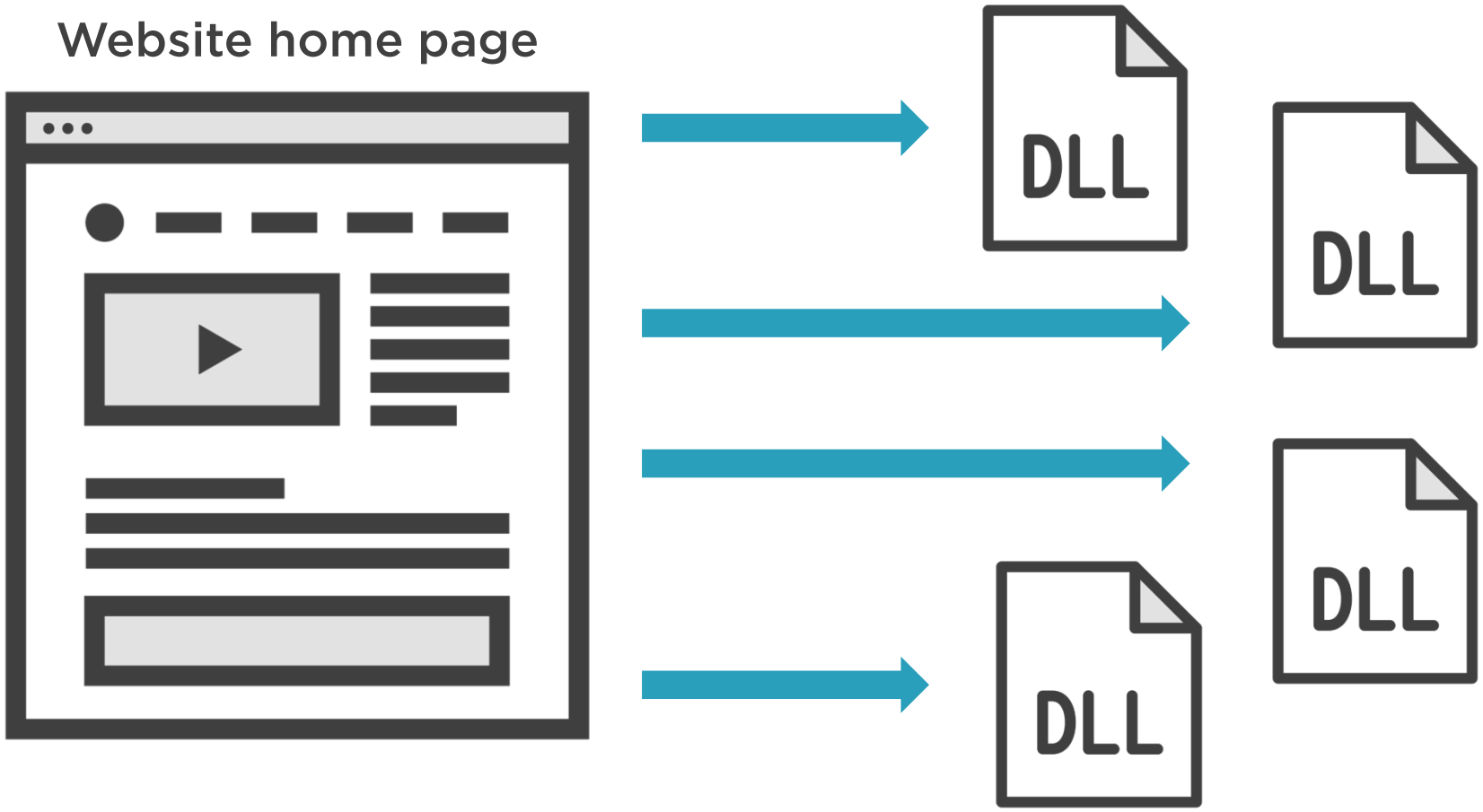


- Internal performance improvements throughout
- 2 - 3x performance boost in many areas
- Common UI tasks should all run faster
- Future improvements to WebAssembly itself

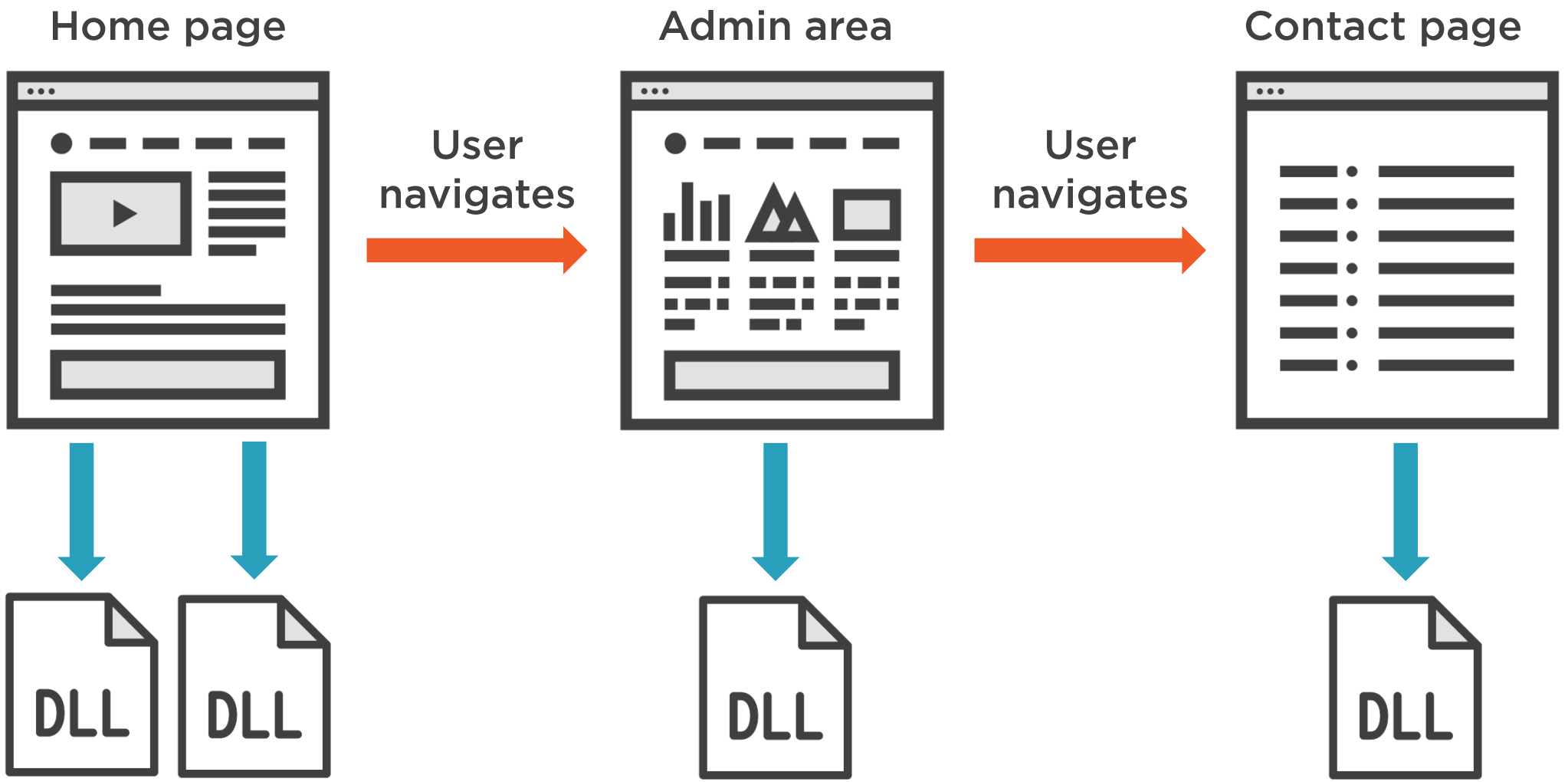
The Blazor Virtualize Component



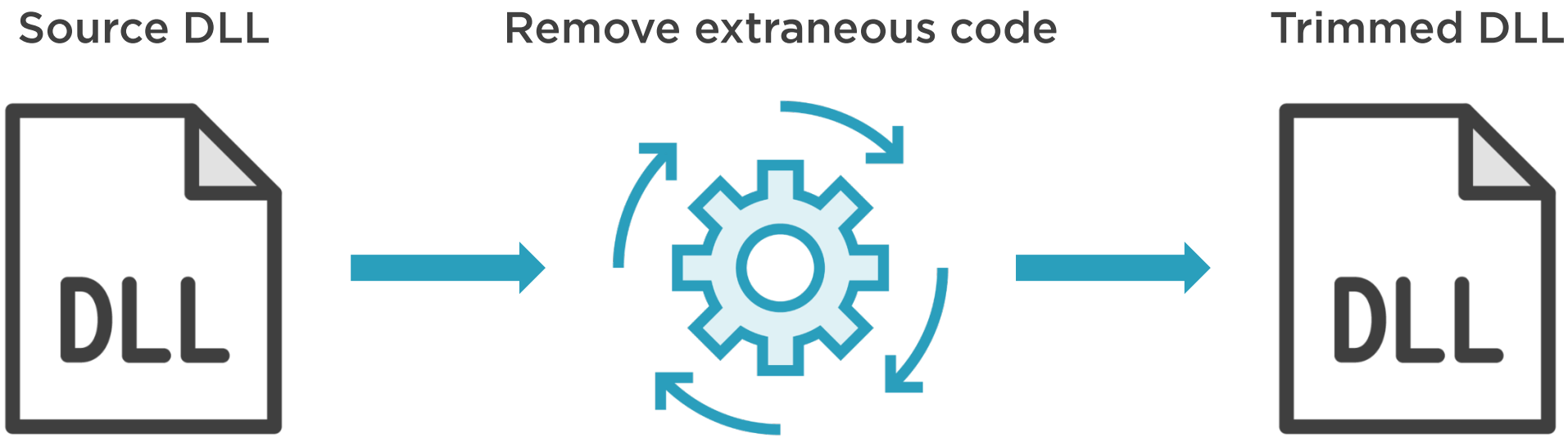
Traditional Blazor WebAssembly Loading



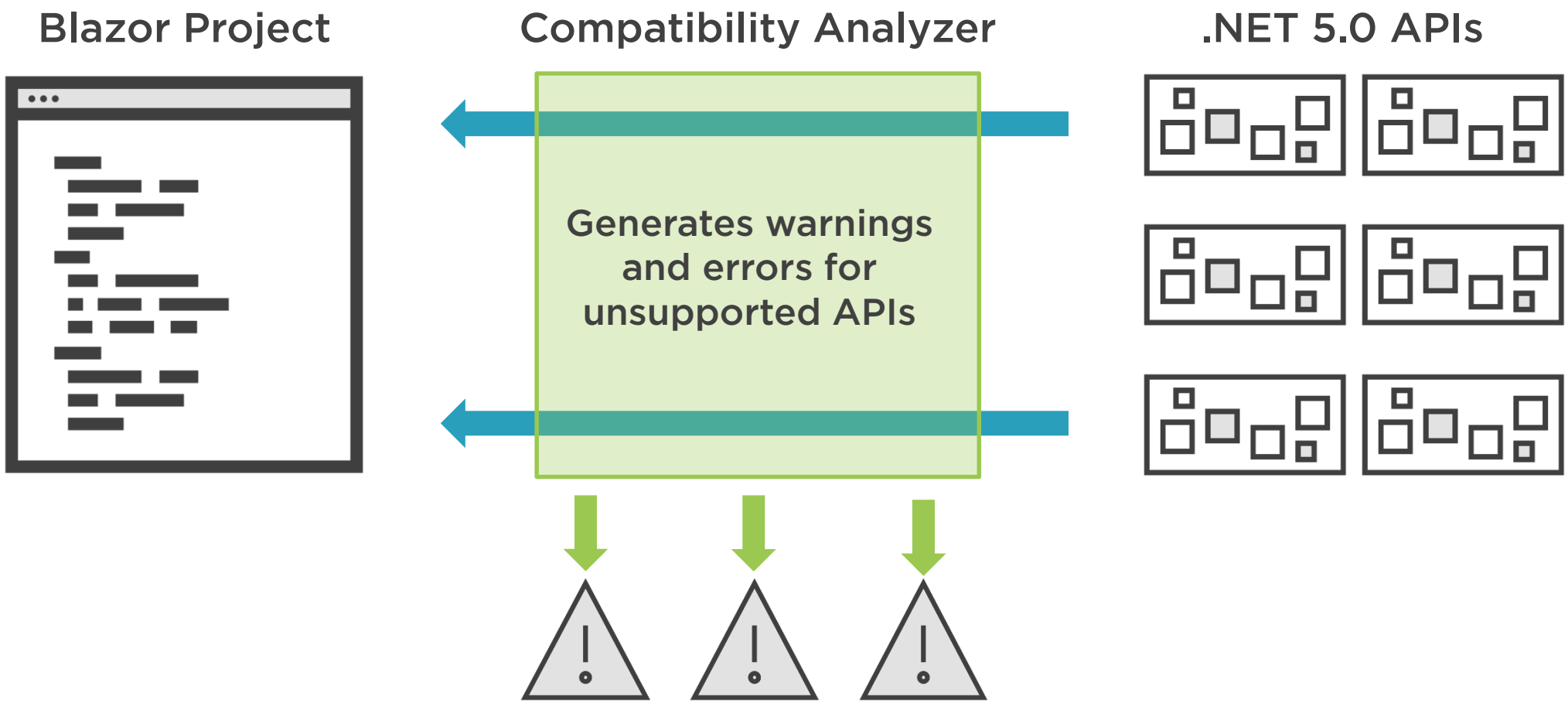
Blazor WebAssembly Lazy Loading



Trimming Compiled Code

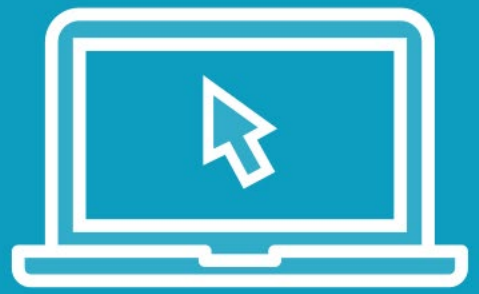


The Browser Compatibility Analyzer



Blazor WebAssembly debugging now works how you'd expect. (for the most part)

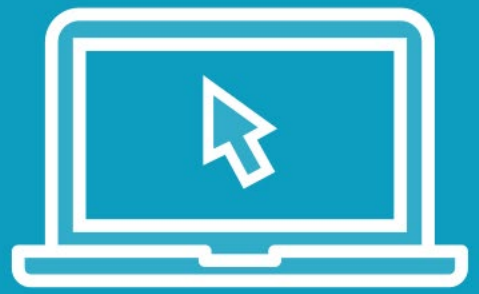
Demo



Optimizing the UI with Virtualization

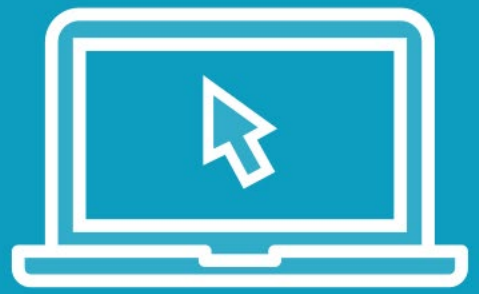
Virtualization is a great feature,
but use with caution.

Demo



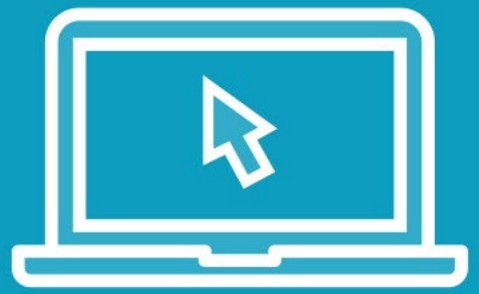
Improving the Virtualized Component

Demo



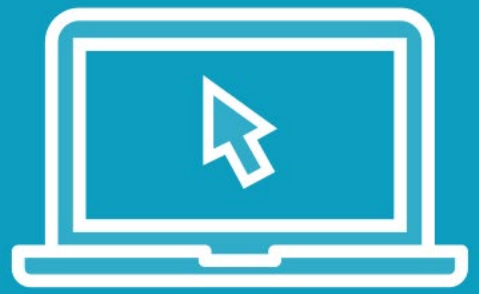
Implementing Lazy Loading

Demo



Exploring Blazor WebAssembly Debugging

Demo



Understanding the Compatibility Analyzer

Summary

- ASP.NET in .NET 5.0 includes many updates to Blazor components
- Dotnet Watch can monitor a project for changes and dynamically recompile the app
- The new InputRadio and InputFile components fix feature gaps in Blazor forms
- Blazor components now support CSS and JavaScript isolation
- JavaScript isolation enables components to import scoped JavaScript modules
- CSS isolation uses naming conventions to define component-scoped stylesheets
- These isolation features allow for cleaner, more portable packages