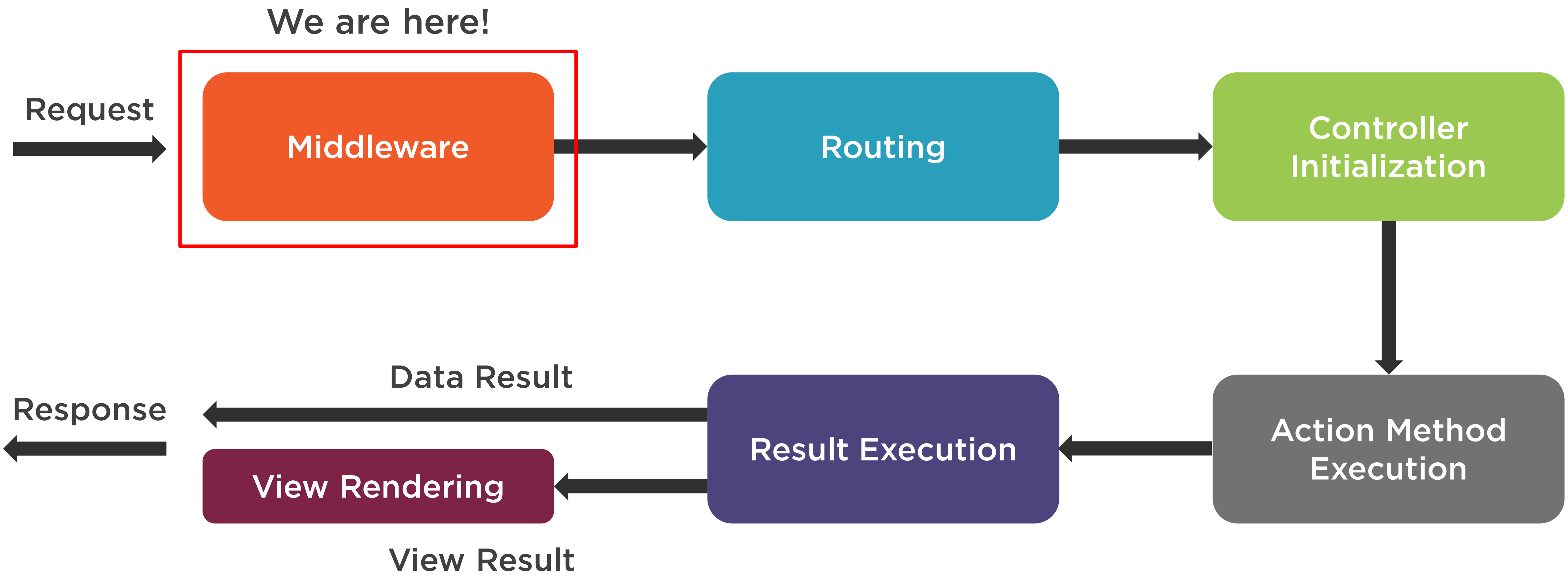
Understanding the Middleware Pipeline

Alex Wolf

www.alexwolfthoughts.com

The MVC Request Life Cycle



To-Do List

What is Middleware?

Coding Middleware

Demo – Touring the Program and Startup Classes

Demo – Building a Simple Middleware Pipeline

Demo – Writing a Reusable Middleware Component

Visualizing the MVC Middleware Pipeline

Demo – MVC Middleware Pipeline Internals

What is Middleware?

Middleware

The series of components that form the application request pipeline.

Features Provided by Middleware

Routing

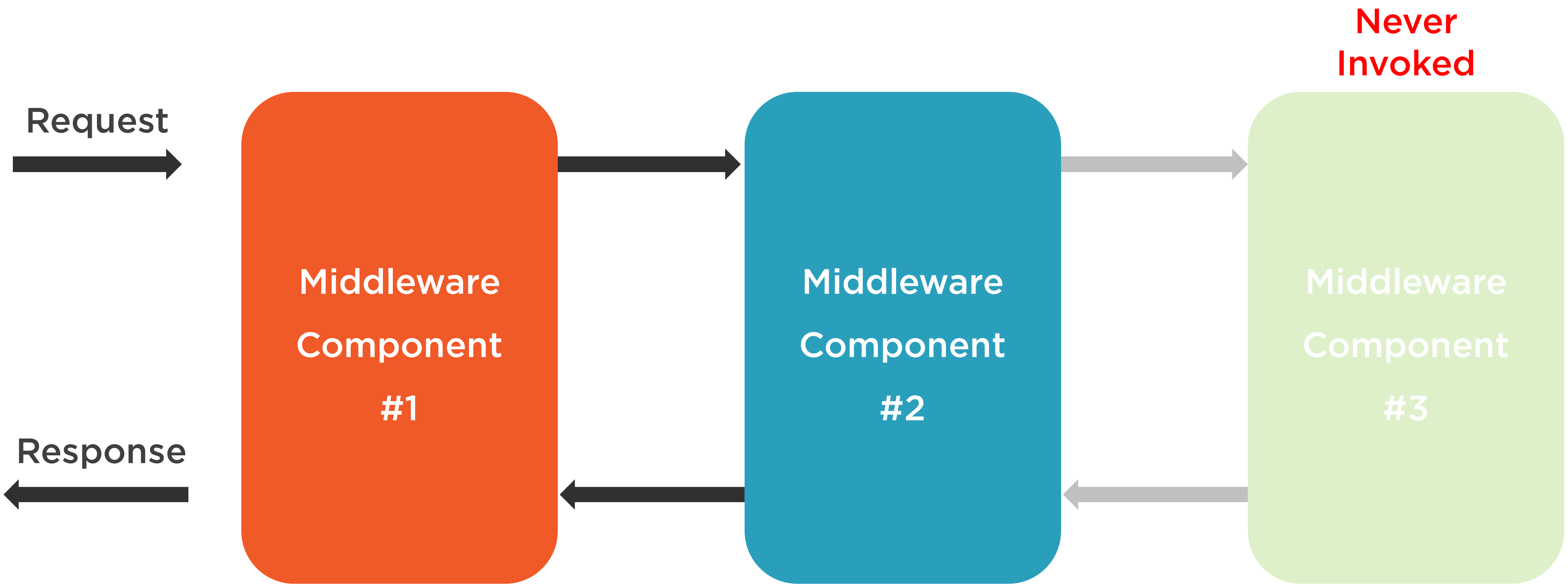
Session

CORS

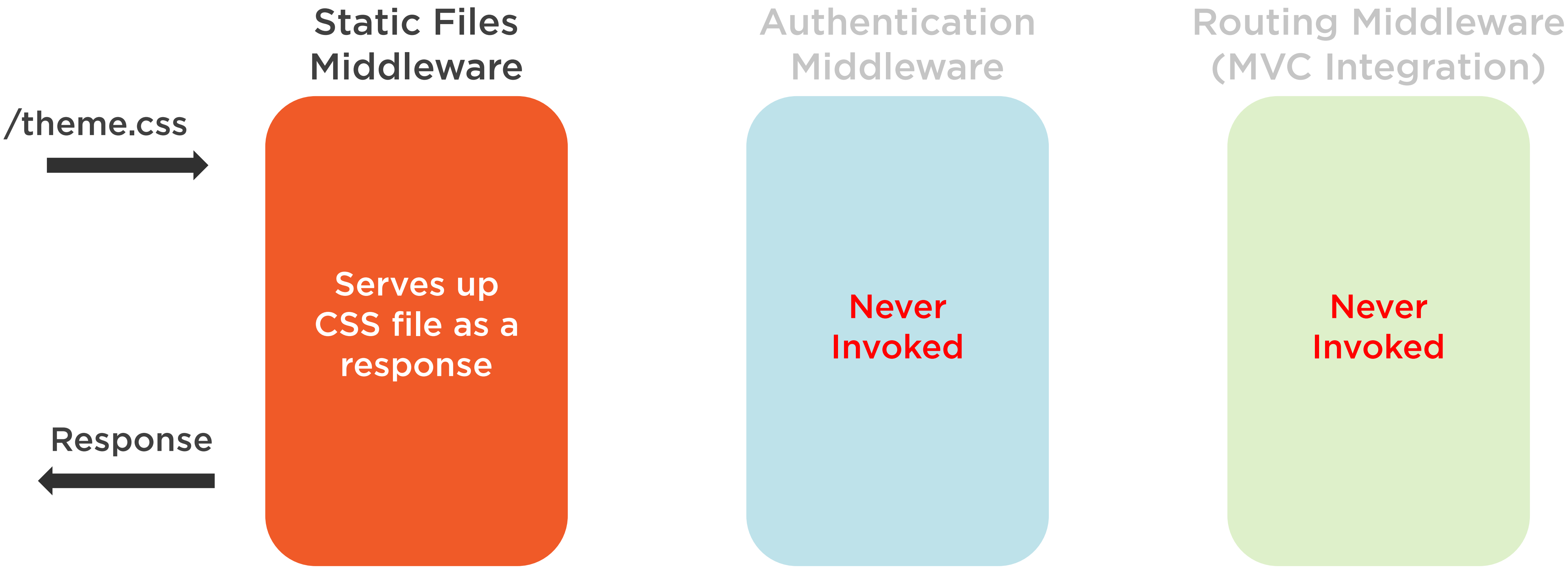
Authentication

Caching

The Middleware Pipeline



A Sample Middleware Pipeline



/theme.css

Response

Static Files Middleware

Serves up CSS file as a response

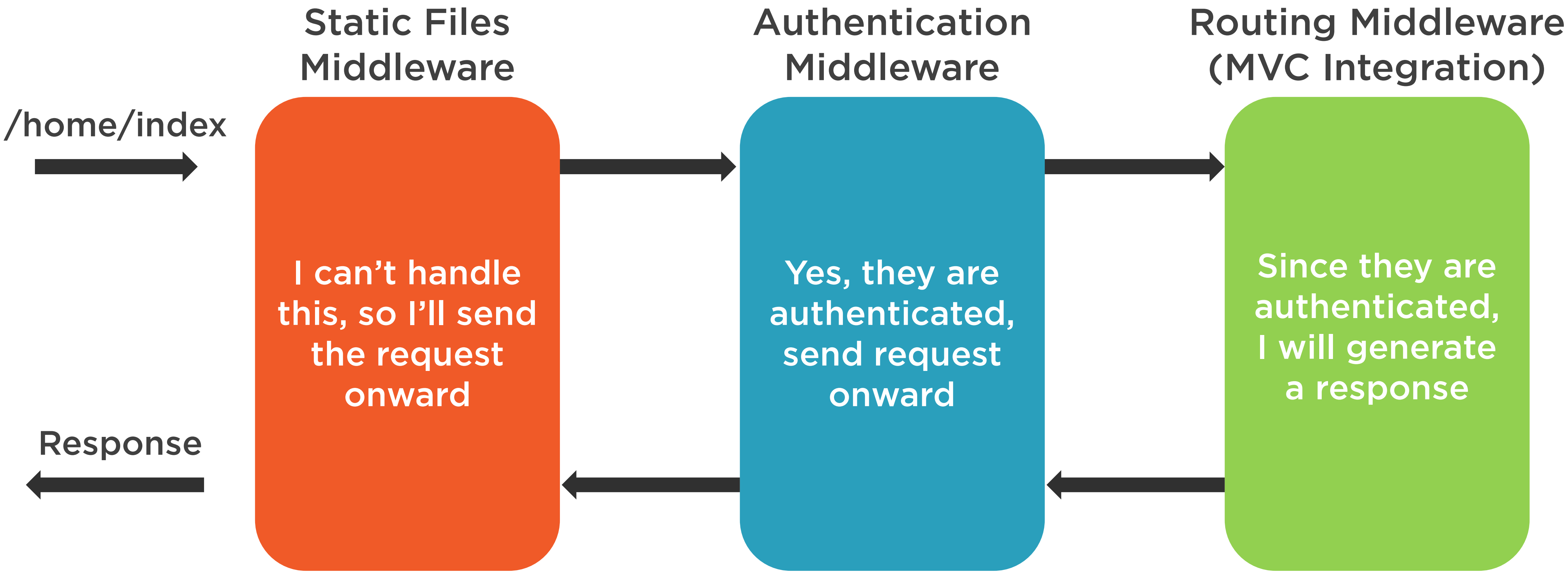
Authentication Middleware

Never Invoked

Routing Middleware (MVC Integration)

Never Invoked

A Sample Middleware Pipeline (cont.)



/home/index

Static Files Middleware

I can't handle this, so I'll send the request onward

Authentication Middleware

Yes, they are authenticated, send request onward

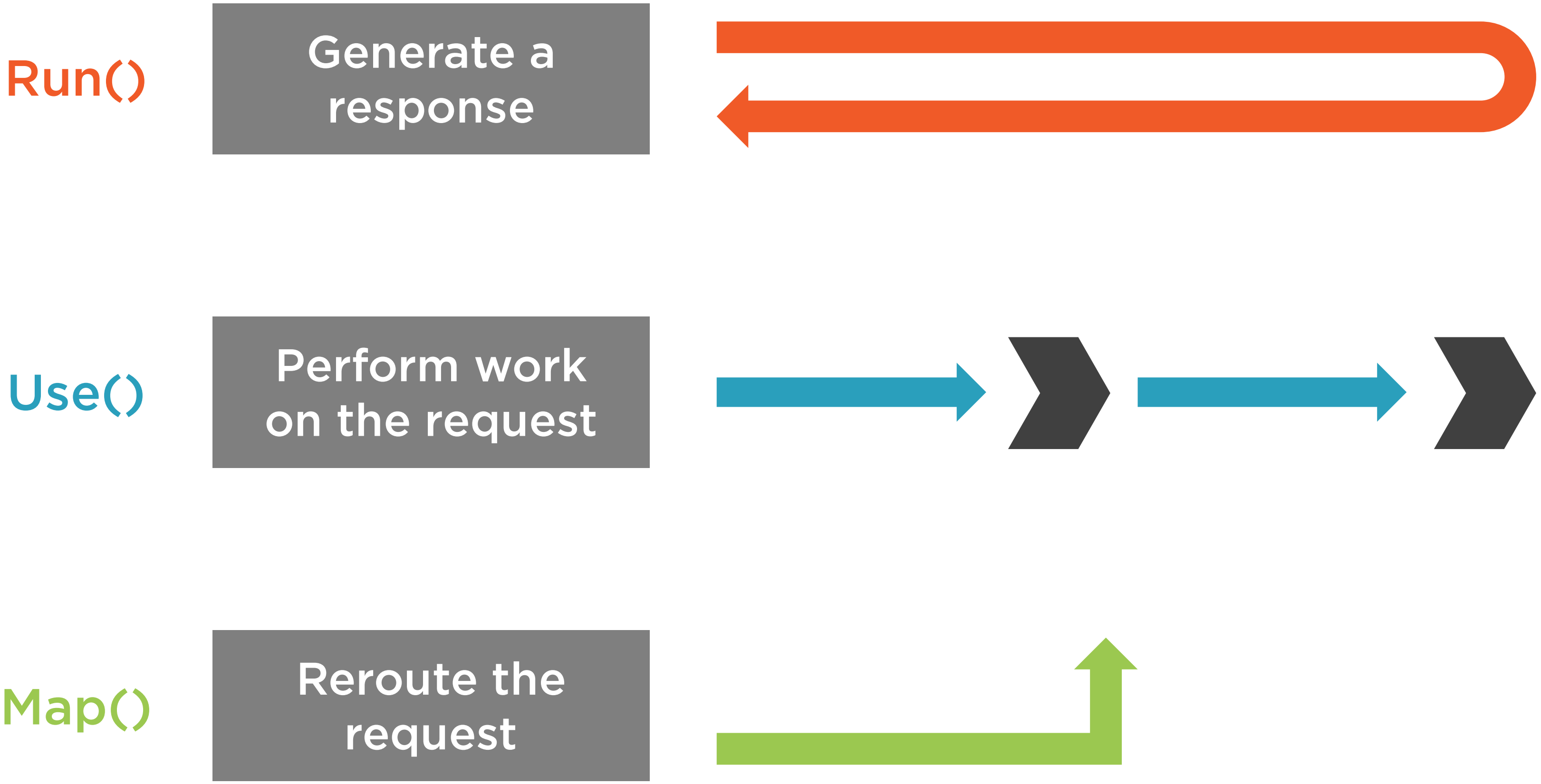
Routing Middleware (MVC Integration)

Since they are authenticated, I will generate a response

Response

Coding Middleware

Middleware Configuration Options



Run()

Generate a response

Use()

Perform work on the request

Map()

Reroute the request

```
app.Run(async context => {
        await context.Response.WriteAsync(“Hello World, from Middleware!”)
});
```

Configuring Middleware with Run

The Run method will terminate the Middleware pipeline

Should generate some type of response

```
app.Use(async (context, next) =>

    // Logic before next Middleware here

    await next.Invoke();

    // Logic after next Middleware here

});


app.Run(async context => {

    await context.Response.WriteAsync
    (“I will generate the response”)

});
```

◀ Middleware registered with “Use” can forward the request onto the next delegate

◀ Second Middleware component implemented with “Run” generates the response

```
app.Map(“/hello-world”, SayHello);
```

- If incoming requests ends in “hello-world” execute the SayHello method

```
private static void SayHello
(IApplicationBuilder app)

{

     app.Run(async context => {

     await context.Response.WriteAsync
     (“Hi, the request was mapped here.”)

     });

}
```

- SayHello method generates response using the Run method

```csharp
public class HelloMiddleware {
private RequestDelegate _next;
public HelloMiddleware(RequestDelegate next)
{
     _next = next;
}
public Task Invoke(HttpContext context)
{
     // Logic here
     await _next.Invoke(context)
}
}
```

- Custom Middleware class
- Constructor accepting the “next” Middleware delegate
- Invoke method to execute logic
- Perform work on the request before forwarding or generating a response

The Program and Startup Classes

Program Class

Used as the entry point to start the entire application

Startup Class

Allows us to configure Middleware components

Demo

Touring the Program and Startup classes

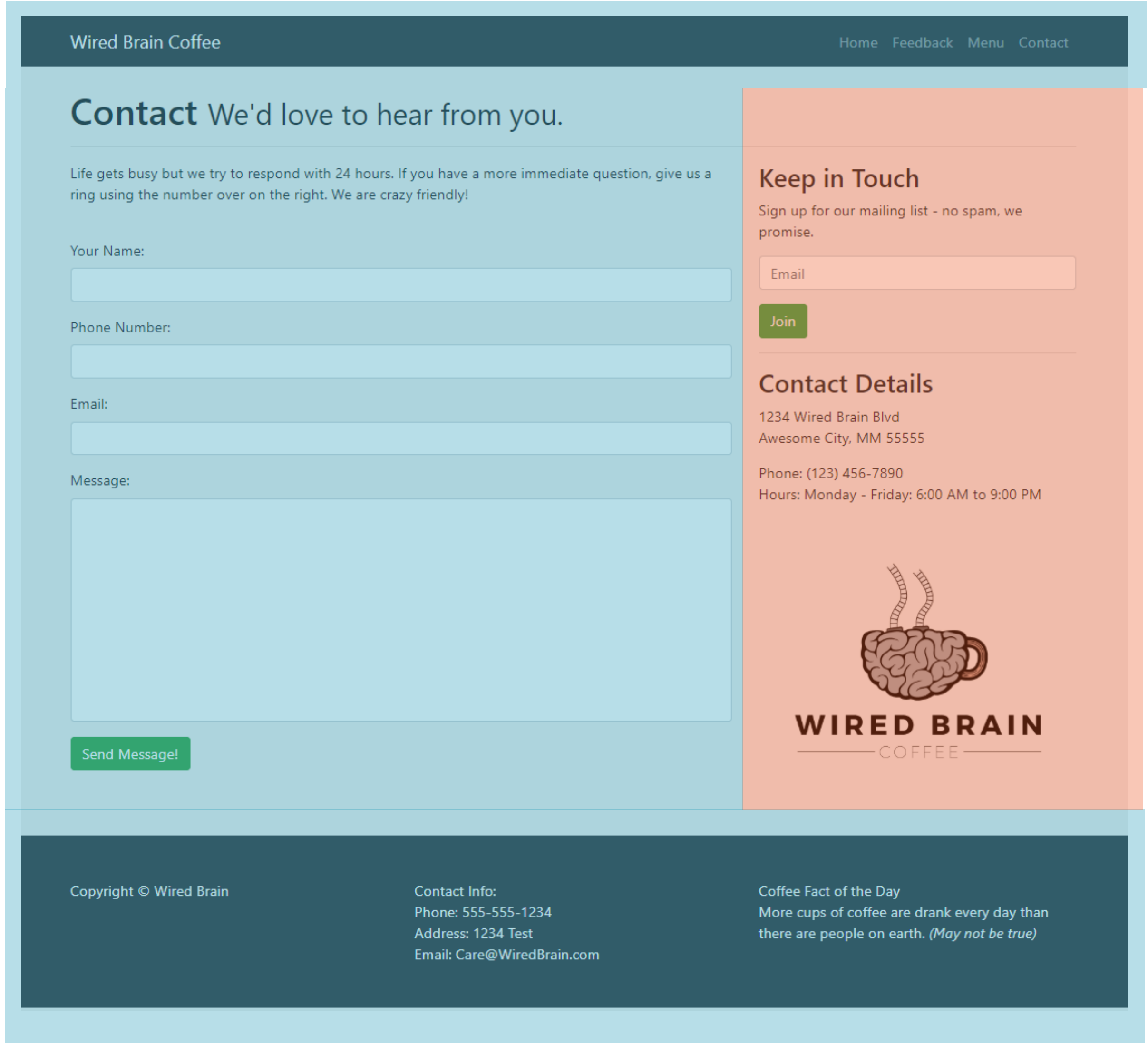
Demo

Building a Simple Middleware Pipeline

Demo

Exploring a Middleware Use Case

Understanding Application Feature Switching



Wired Brain Coffee

Home Feedback Menu Contact

Contact We'd love to hear from you.

Life gets busy but we try to respond with 24 hours. If you have a more immediate question, give us a ring using the number over on the right. We are crazy friendly!

Your Name:

Phone Number:

Email:

Message:

Send Message!

Keep in Touch

Sign up for our mailing list - no spam, we promise.

Email

Join

Contact Details

1234 Wired Brain Blvd
Awesome City, MM 55555

Phone: (123) 456-7890
Hours: Monday - Friday: 6:00 AM to 9:00 PM

WIRED BRAIN
COFFEE

Copyright © Wired Brain

Contact Info:
Phone: 555-555-1234
Address: 1234 Test
Email: Care@WiredBrain.com

Coffee Fact of the Day
More cups of coffee are drank every day than there are people on earth. *(May not be true)*

🟧 Disabled

🟦 Enabled

A Case for Middleware

Reusable functionality

Application level concern

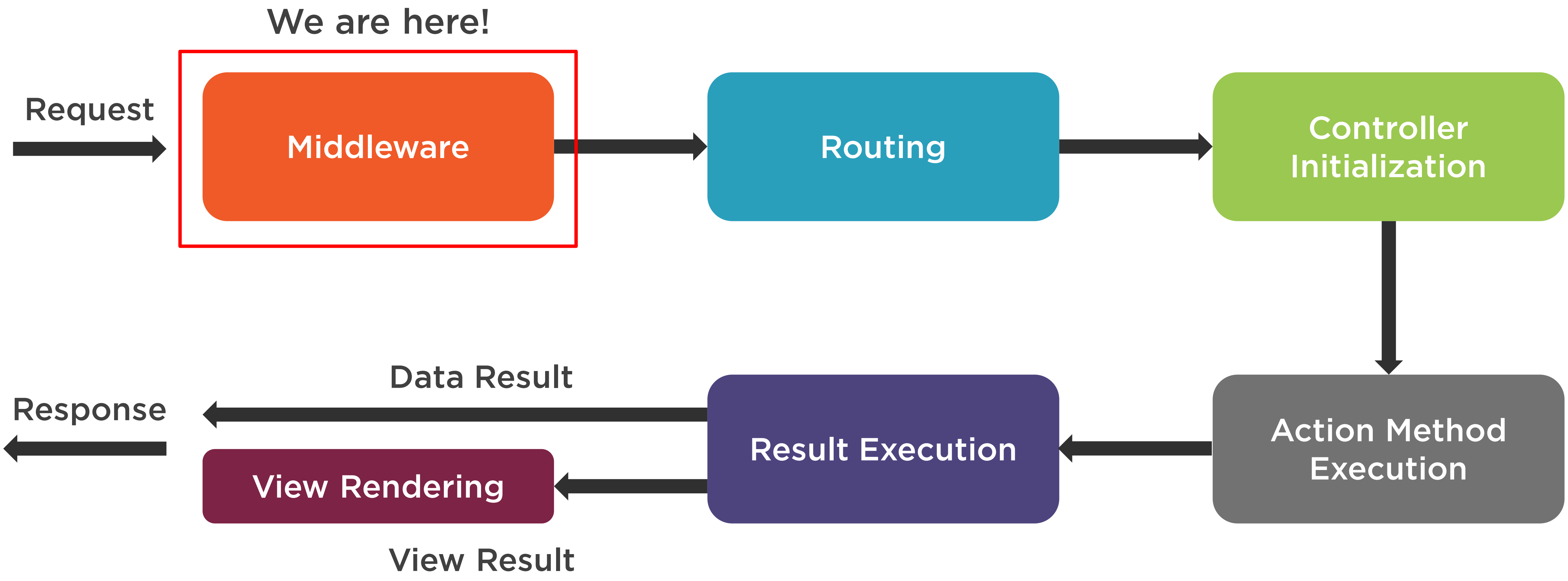
Request driven feature

Demo

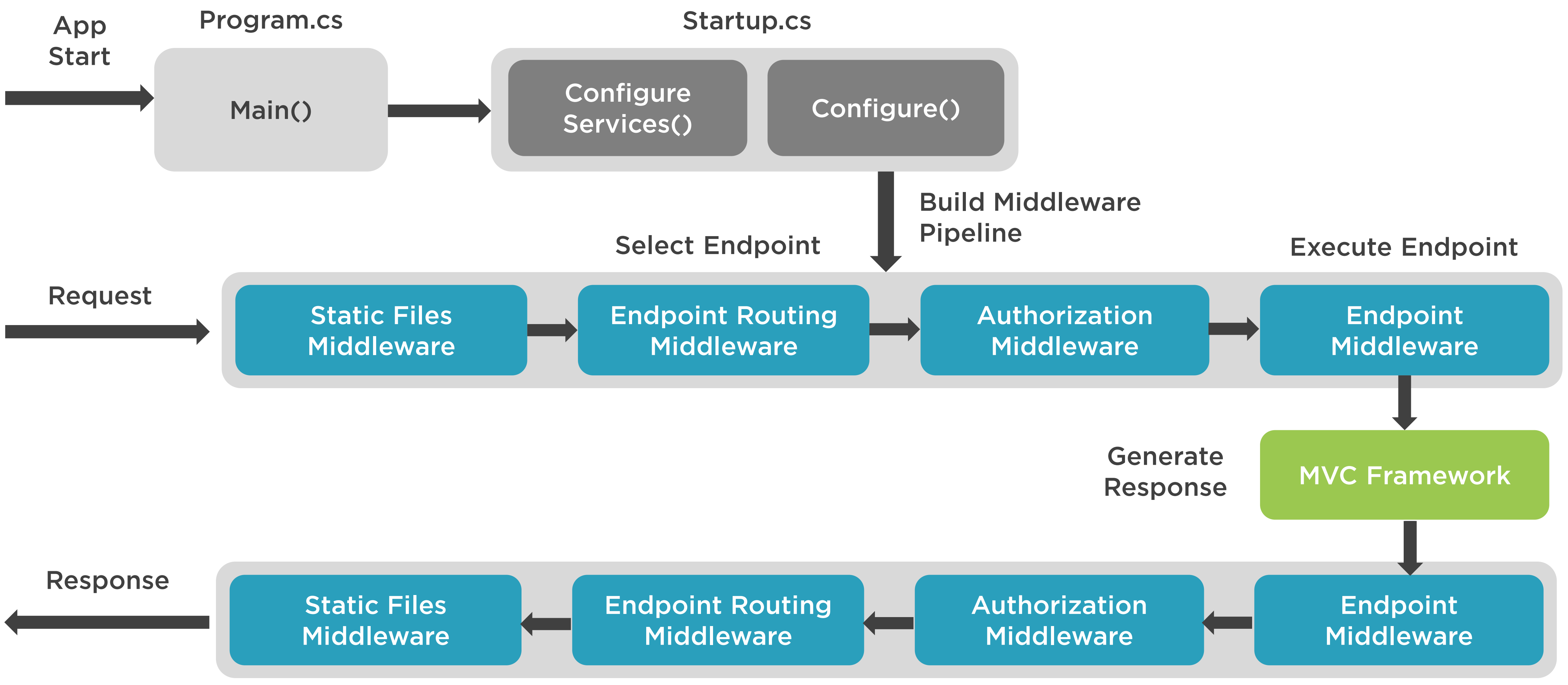
Coding a Reusable Middleware Component

Visualizing the MVC Middleware Pipeline

The MVC Request Life Cycle



Understanding the MVC Middleware Pipeline



A note about Routing and Middleware.

Demo

Exploring MVC Middleware Internals

Summary

Middleware forms the basic building blocks of the HTTP Pipeline in MVC Core

Middleware can perform work on a request in the pipeline or generate a response

Middleware can exist as simple inline methods or as complex, reusable classes

Middleware provides application level features like serving files and authorization

The Middleware pipeline is built inside the Configure method of the Startup class

Endpoint Routing in .NET Core 3.0 is implemented through Middleware