

Understanding Controller Invocation



Alex Wolf

www.alexwolfthoughts.com



Terminology Considerations

MVC,
MVC Framework



ASP.NET Core



Controllers, Action
Methods, Filters, Razor
Views

Routing

Middleware

Security & Identity

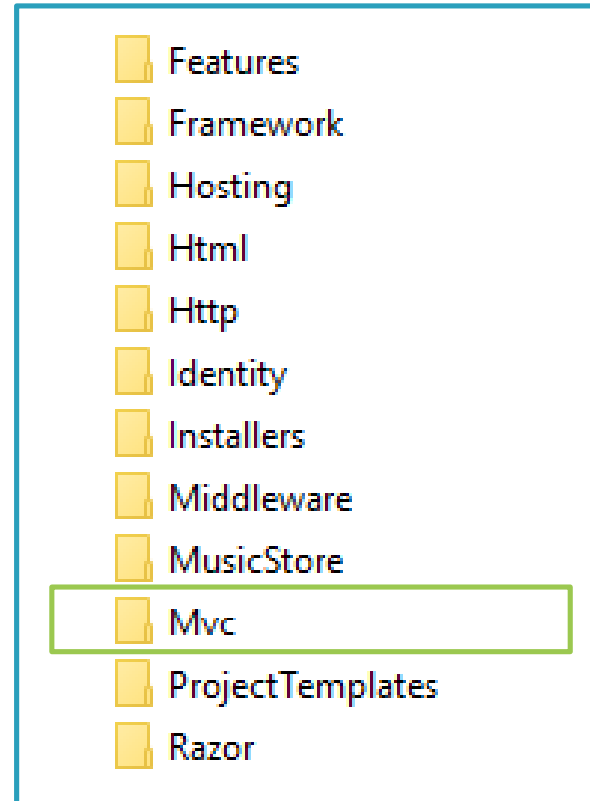
Others



Revisiting the ASP.NET Core Source Project

ASP.NET Core

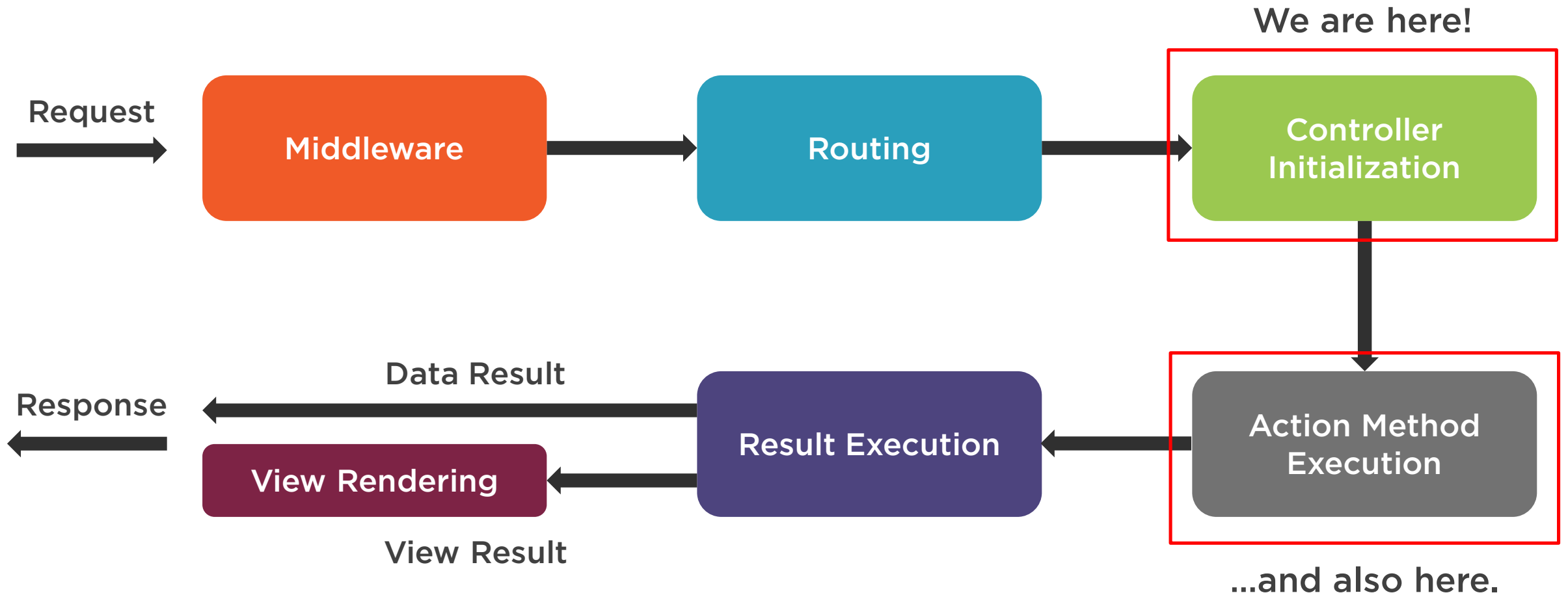
MVC Framework



8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:49 PM	File folder
8/3/2019 11:38 PM	File folder
8/3/2019 11:38 PM	File folder



The MVC Request Life Cycle



To-Do List



Exploring Controller invocation

Demo - Stepping through Controller creation

Understanding Filters and the Life Cycle

Introducing Authorization Filters

Demo - Creating an Authorization Filter

Demo - Implementing an Authorization Filter

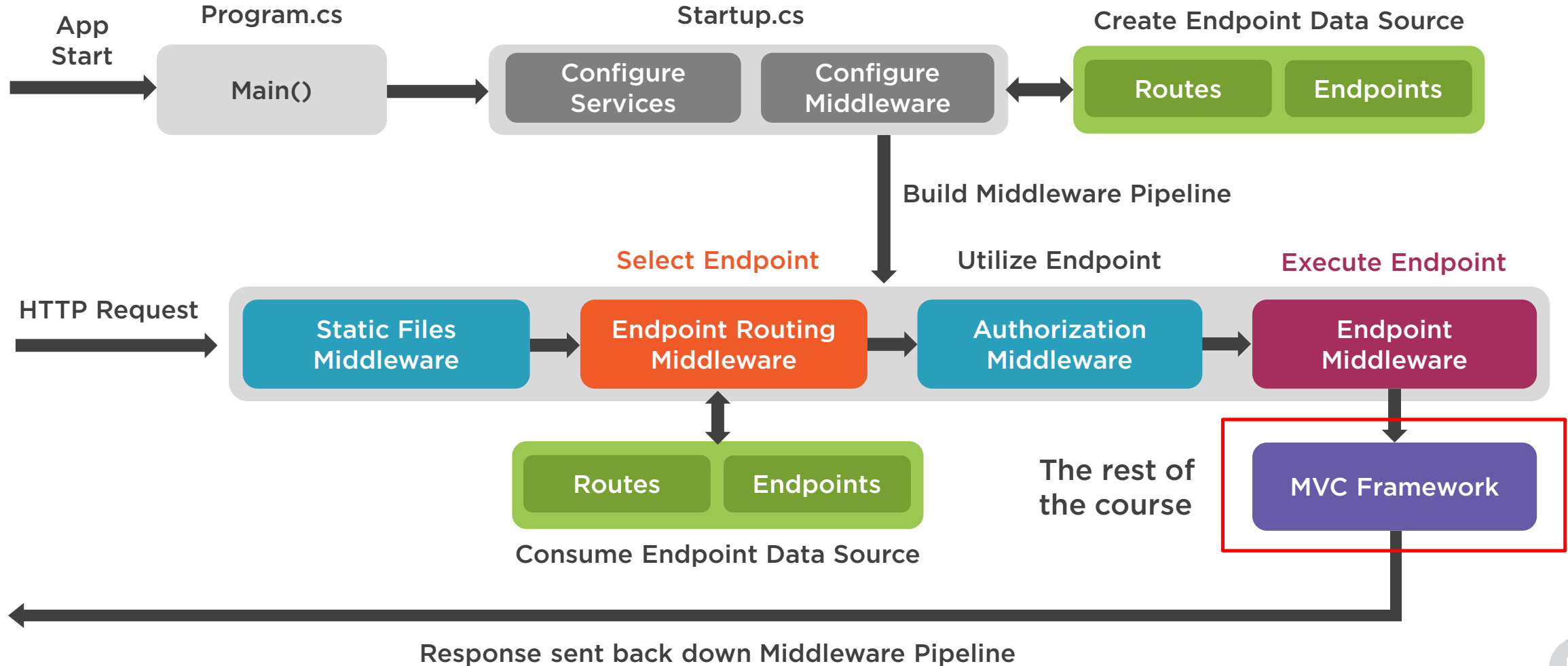
Understanding Resource and Middleware Filters



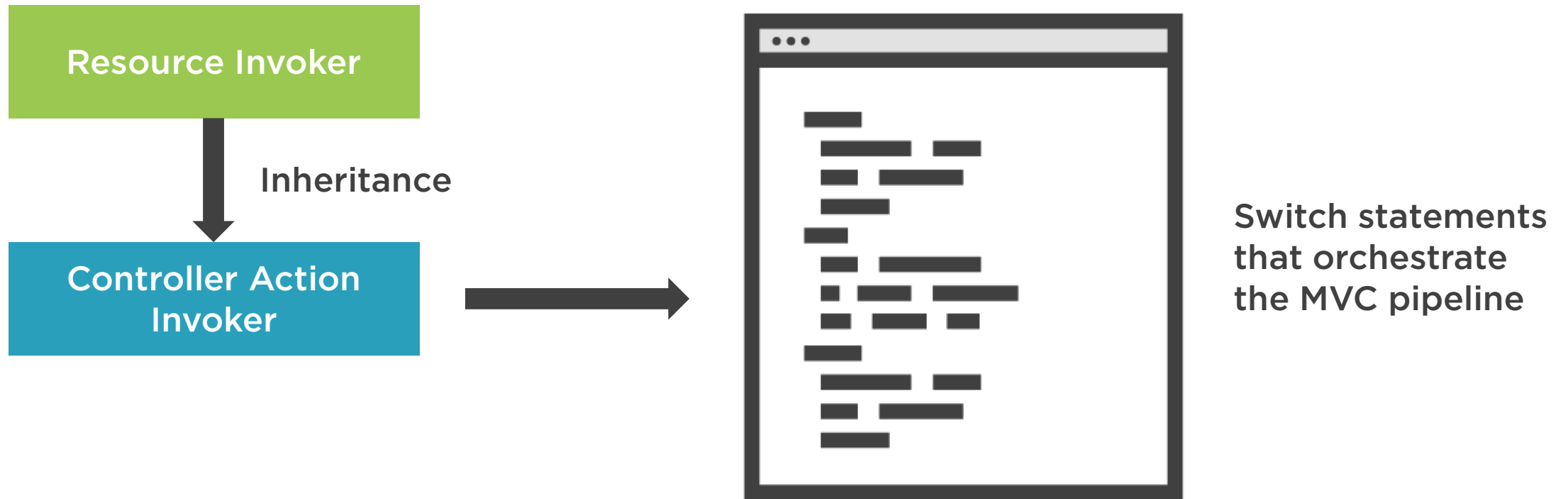
Exploring Controller Initialization



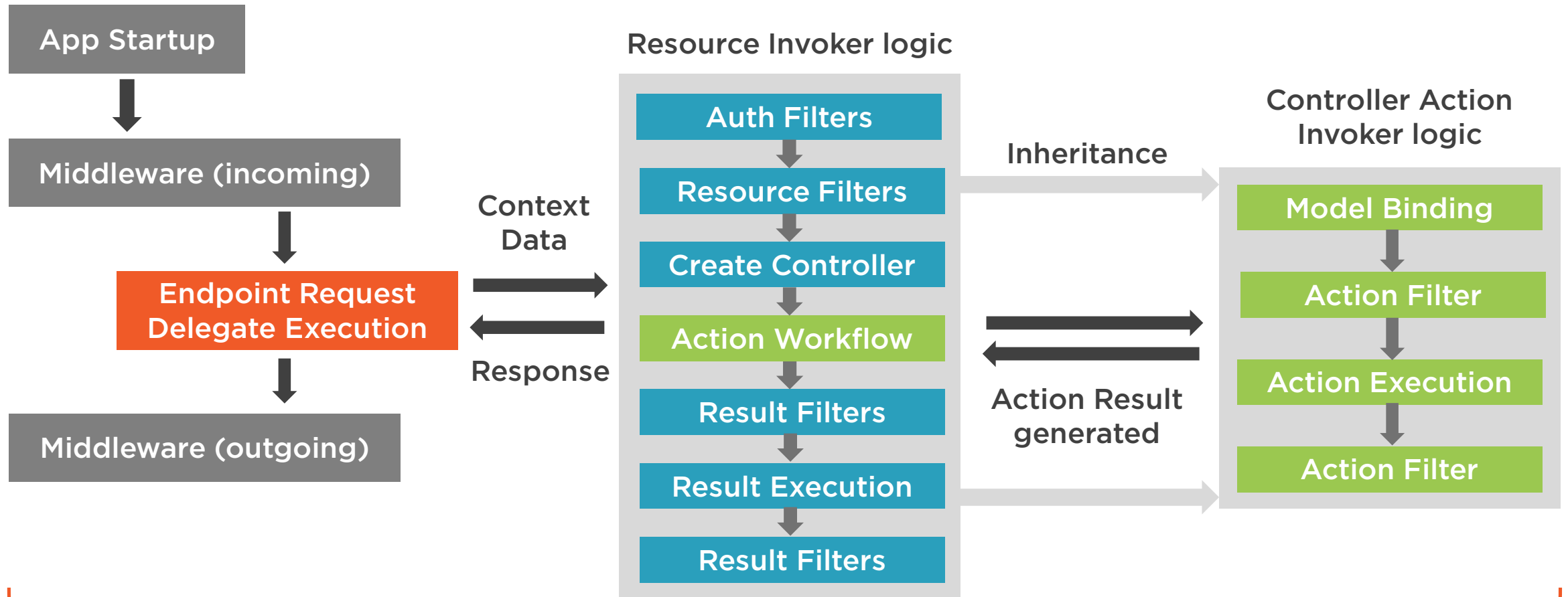
Revisiting the Middleware Pipeline



Understanding the Invoker Classes



Exploring the MVC Pipeline



The complete MVC Request Life Cycle



Sidebar!

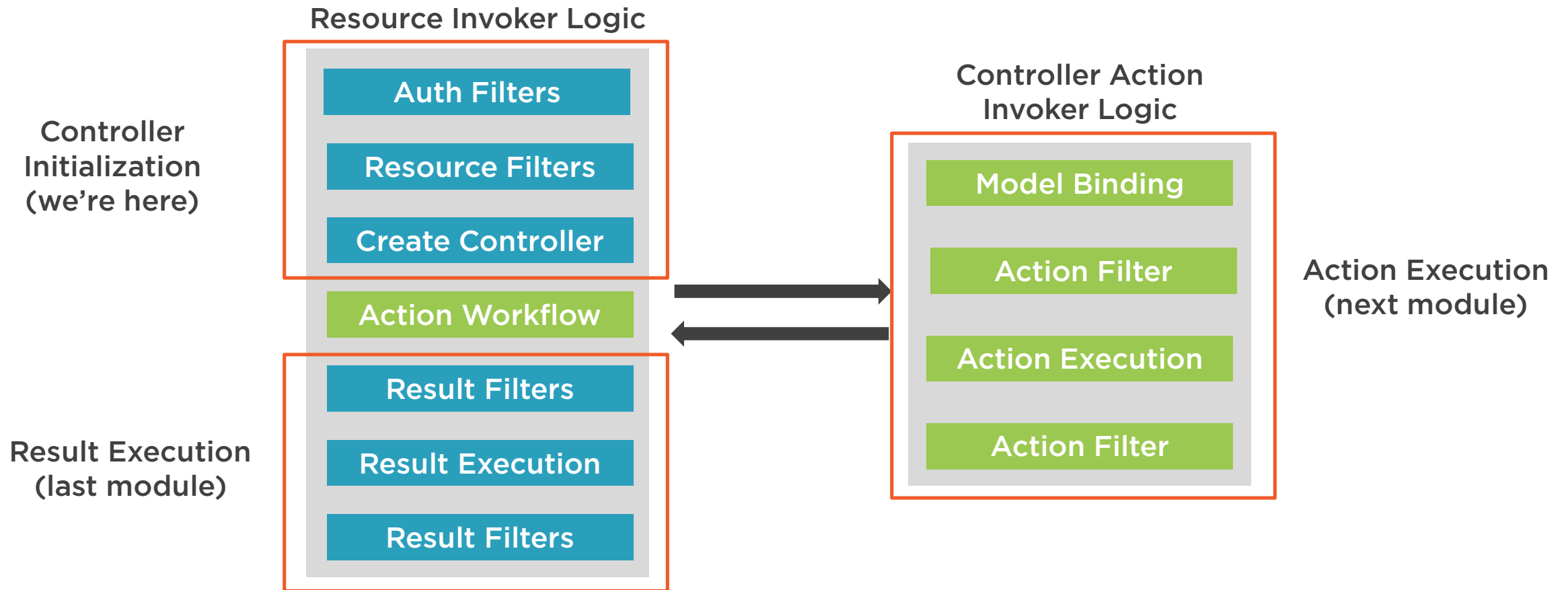
The Resource Invoker is an abstract class

An instance of the Controller Action Invoker controls program flow

The Resource Invoker logic is utilized through inheritance



Breaking Down the MVC Pipeline



Demo



Stepping through Controller Creation



Filters and the Request Life Cycle

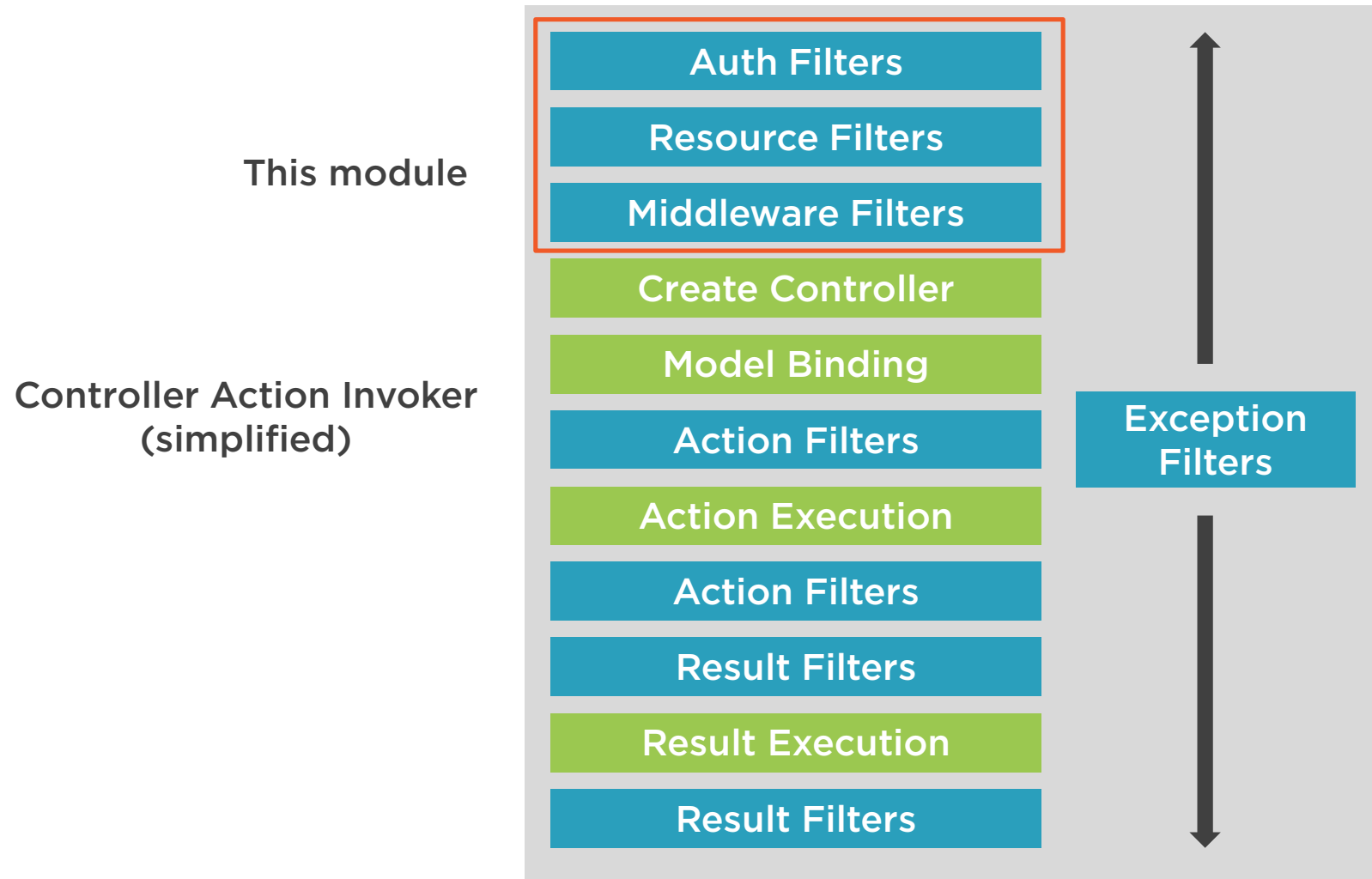


Filters

Components that allow us to inject logic at various stages of the MVC Request Life Cycle.



Understanding Filter Execution



Putting the Available Filters in Context

Authorization

Action

Result

Resource

Middleware

Exception

Service

Type

High Priority

Medium Priority

Low Priority




```
[LocalHostOnly]

public class HomeController : Controller
{
    [FeatureFilter]
    public IActionResult Index()
    {
        return View();
    }
}
```

◀ **Authorization Filter applied to a Controller**

◀ **Action Filter applied to a single Action Method**



Synchronous and Asynchronous Filters

Synchronous Interface	Asynchronous Interface
IActionFilter	IAsyncActionFilter
IAuthorizationFilter	IAsyncAuthorizationFilter
IExceptionFilter	IAsyncExceptionFilter
IResultFilter	IAsyncResultFilter
IResourceFilter	IAsyncResourceFilter



Introducing Authorization Filters



Authorization Filters

Execute first in the MVC pipeline to determine whether a request is authorized to proceed.



```
public interface IAuthorizationFilter : IFilterMetadata
{
    object OnAuthorization(AuthorizationFilterContext context);
}
```

The **IAuthorizationFilter** Interface

Validates whether a request is authorized using custom logic

Can short circuit the life cycle for unauthorized requests



A note about Authorization.



Feature Switching Parts of an Application

Wired Brain Coffee

HomeFeedbackMenuContact

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



Disabling an Entire Application

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:

Keep in Touch


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

Email

Contact Details

1234 Wired Brain Blvd
Awesome City, MM 55555

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



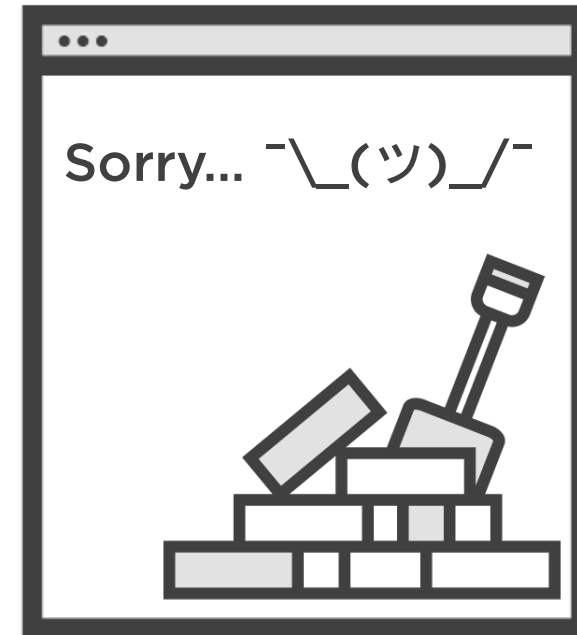
WIRED BRAIN
COFFEE

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)*

Redirect



Outage page



Disabled



Demo



Creating a Custom Authorization Filter



Demo



Implementing a Custom Authorization Filter



Understanding Resource and Middleware Filters



```
public interface IResourceFilter : IFilterMetadata
{
    object OnResourceExecuting(ResourceExecutingContext context);
    object OnResourceExecuted(ResourceExecutedContext context);
}
```

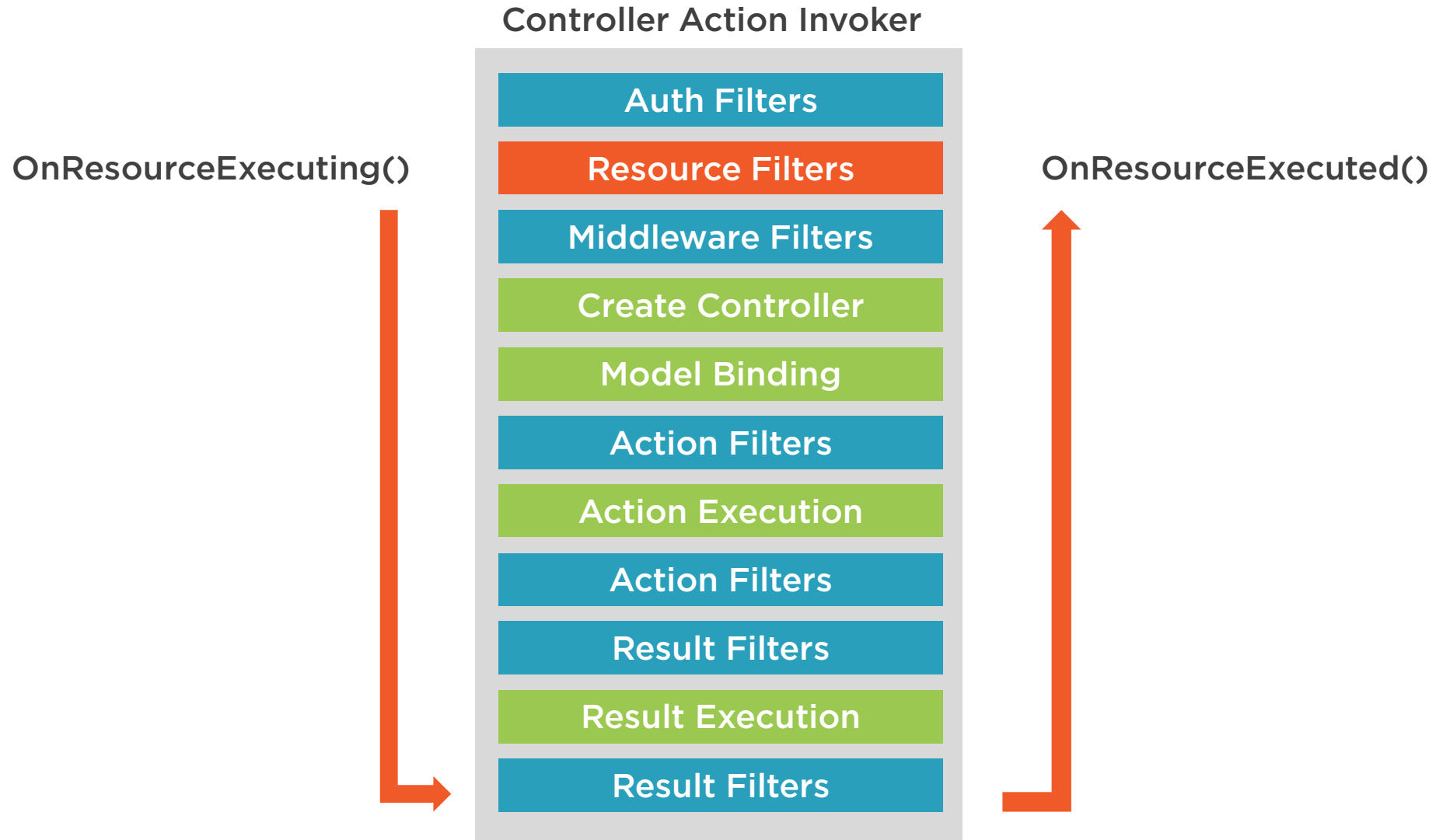
The **IResourceFilter** Interface

Defines two methods that allow us to add custom logic to the pipeline

These methods wrap both ends of the authorized MVC pipeline



Understanding Resource Filter Execution



Resource Filters

Often handle caching

Can short-circuit the request

Can manage incoming data

Execute logic very early and very late



The Purpose of Middleware Filters

**Reuse or delay
Middleware logic**

**Run Middleware within
MVC context**

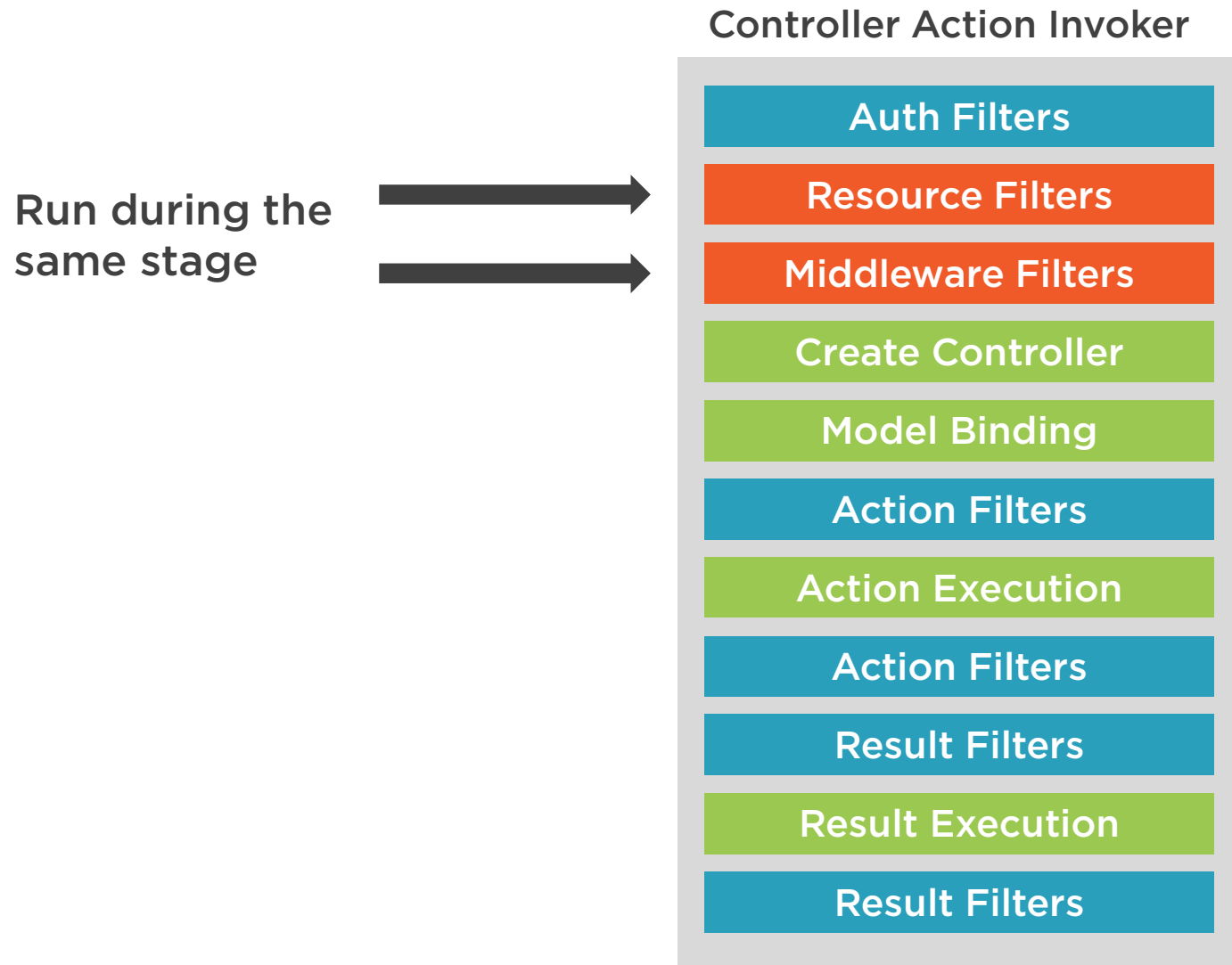


```
[MiddlewareFilter(typeof(TestPipeline))]  
public class HomeController : Controller  
{  
    // Action Methods  
}  
  
public class TestPipeline  
{  
    public void Configure  
    (IApplicationBuilder appBuilder)  
    {  
        appBuilder.UseMiddleware<Test>();  
    }  
}
```

- ◀ **Middleware Filter applied to Controller with pipeline parameter**
- ◀ **The custom pipeline class**
- ◀ **The Configure method to declare Middleware components**



Middleware Filter Execution



Summary



The Endpoint Request Delegate executes the Controller Action Invoker

The Controller Action Invoker inherits from the abstract Resource Invoker

These invokers orchestrate the MVC Pipeline

Authorization Filters execute first in the MVC pipeline

Resource Filters execute logic at the start and end of the MVC pipeline

Middleware Filters allow us to execute Middleware in the context of MVC

A Controller instance is retrieved using the Controller Factory

