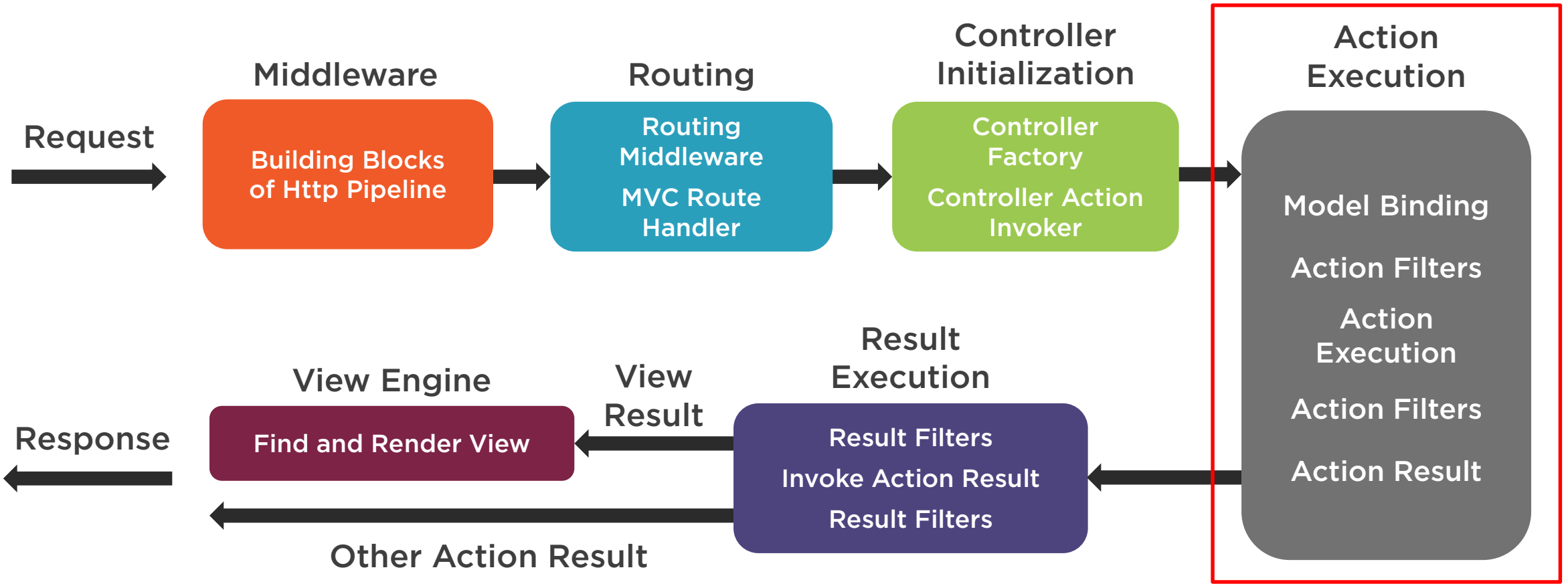
Handling Requests with Action Methods

Alex Wolf

www.alexwolfthoughts.com

The MVC Request Life Cycle



To-Do List

- Action Methods and the Request Life Cycle
- Understanding the Model Binding Process
- Demo – Touring Model Binding
- Understanding Filters
- Filter Execution in Depth
- Demo – Security with Authorization Attributes
- Demo – Action Filter Execution
- Action Method Execution

Action Methods and the Request Life Cycle

```
Public class HomeController : Controller
{
    public IActionResult Edit()
    {
        return View();
    }
}
```

Action Methods

Responsible for handling an incoming request

Intended to map to user or system actions rather than just entire pages

Return different types of responses through Action Results

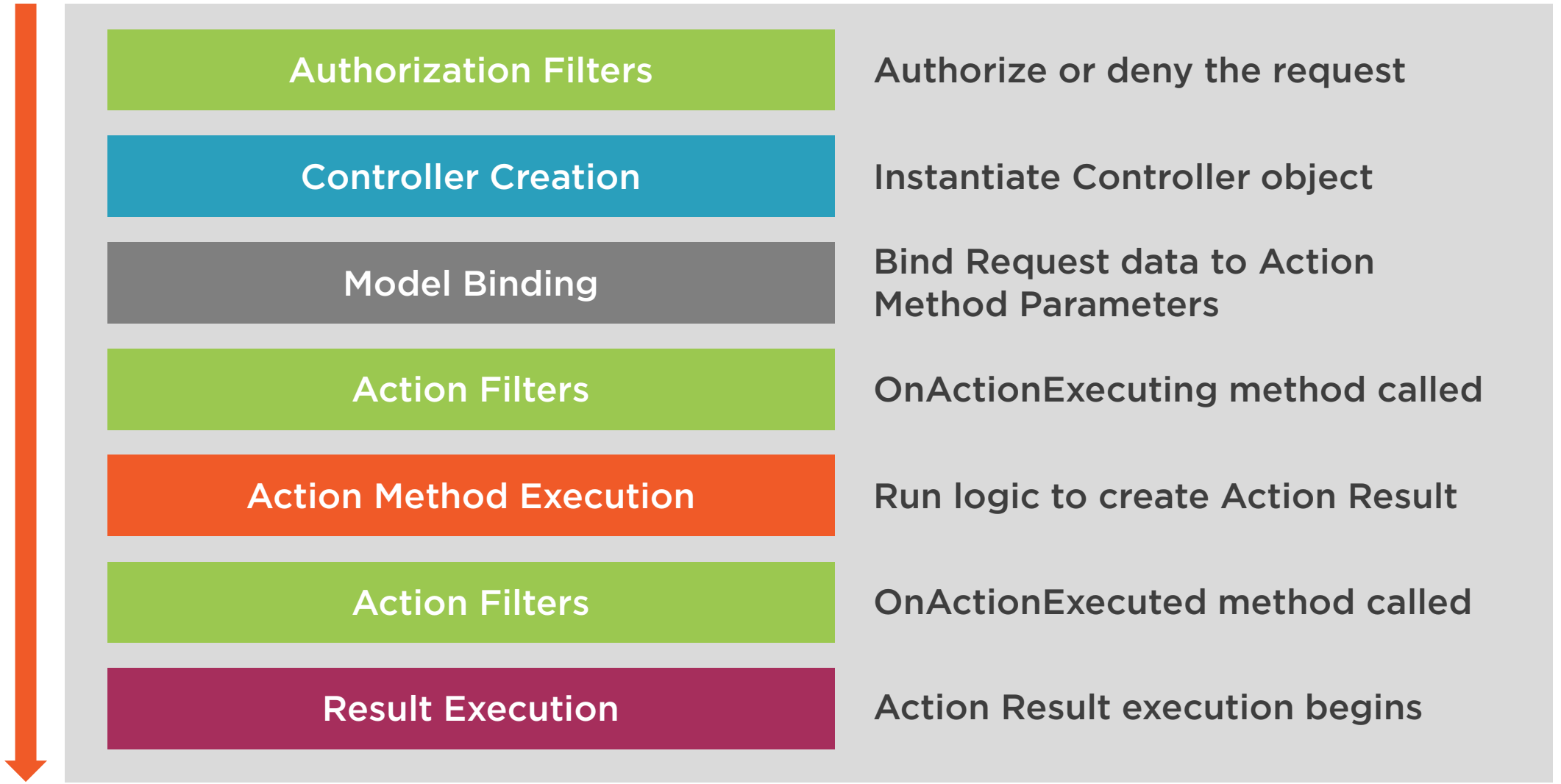
```
public IActionResult Edit(Customer cust)
{
    return View();
}

[Authorize]
[LogAction]
public IActionResult Index()
{
    return View();
}
```

- Action Method accepting a parameter

- Action Method with Authorization and Action Filters

The Action Method Execution Process



Introducing Model Binding

Model Binding is the process of mapping incoming Request data to Action Method parameters.

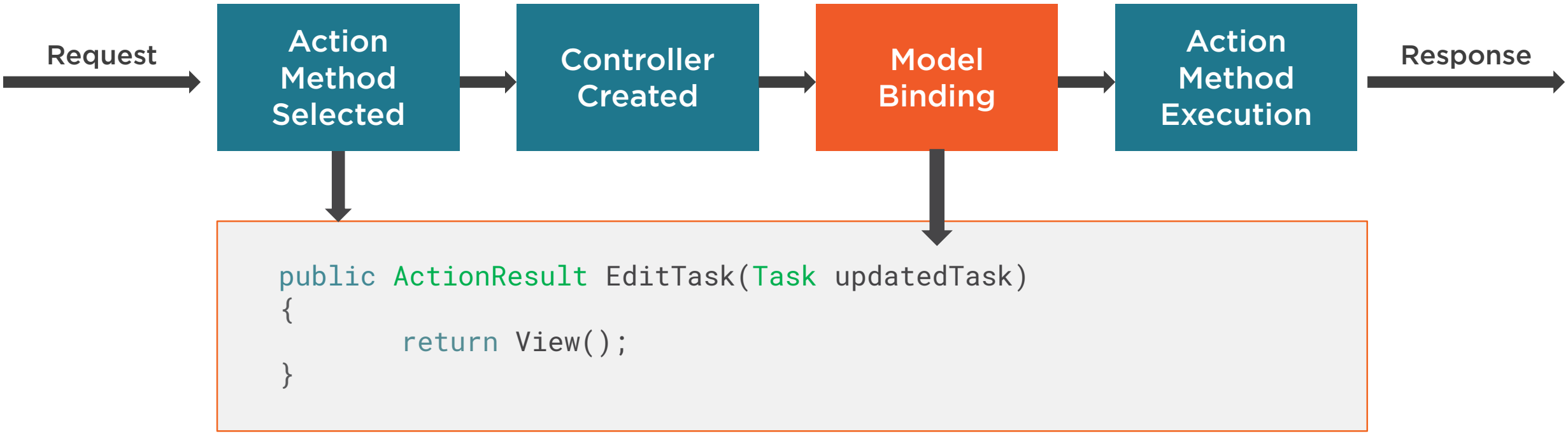
```
public interface IModelBinder
{
    Task BindModelAsync(ModelBindingContext bindingContext);
}
```

The IModelBinder Interface

BindModelAsync is responsible for binding Action Method parameters

Controller Model State is generally updated through this method

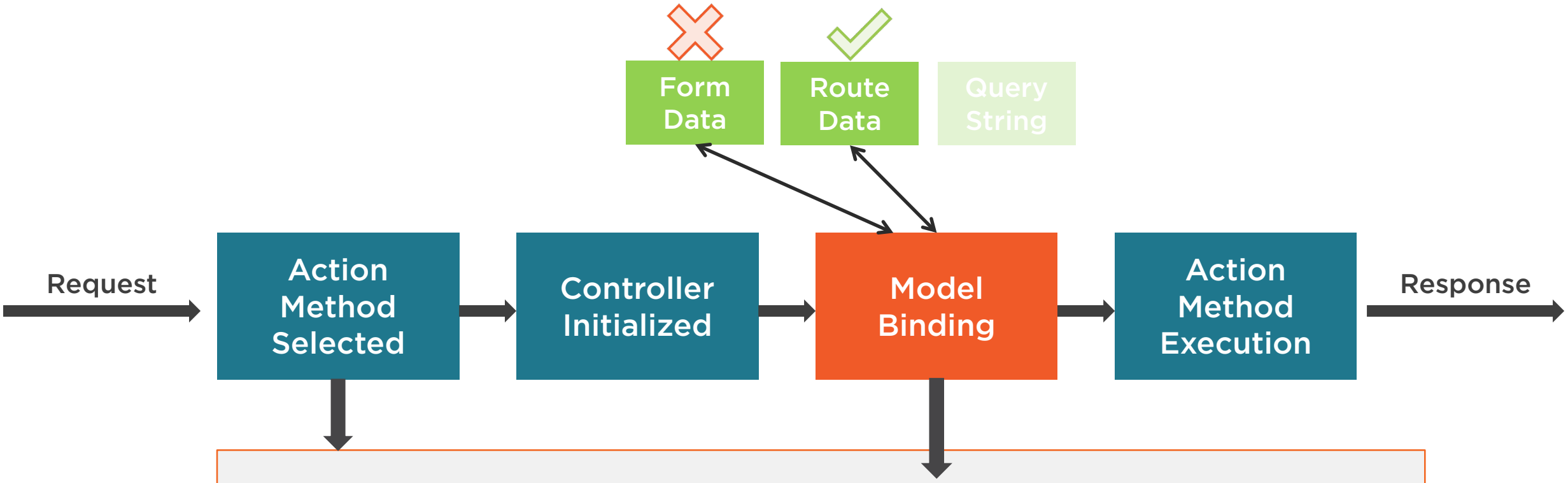
Model Binding and the Request Life Cycle



The Types of Value Providers

- Form Value Provider
- Route Data Value Provider
- Query String Value Provider
- Custom and More!

Supplying Data with Value Providers




```
public ActionResult Edit (int customerNumber)
{
        return View();
}
```

Current and Legacy Model Binding

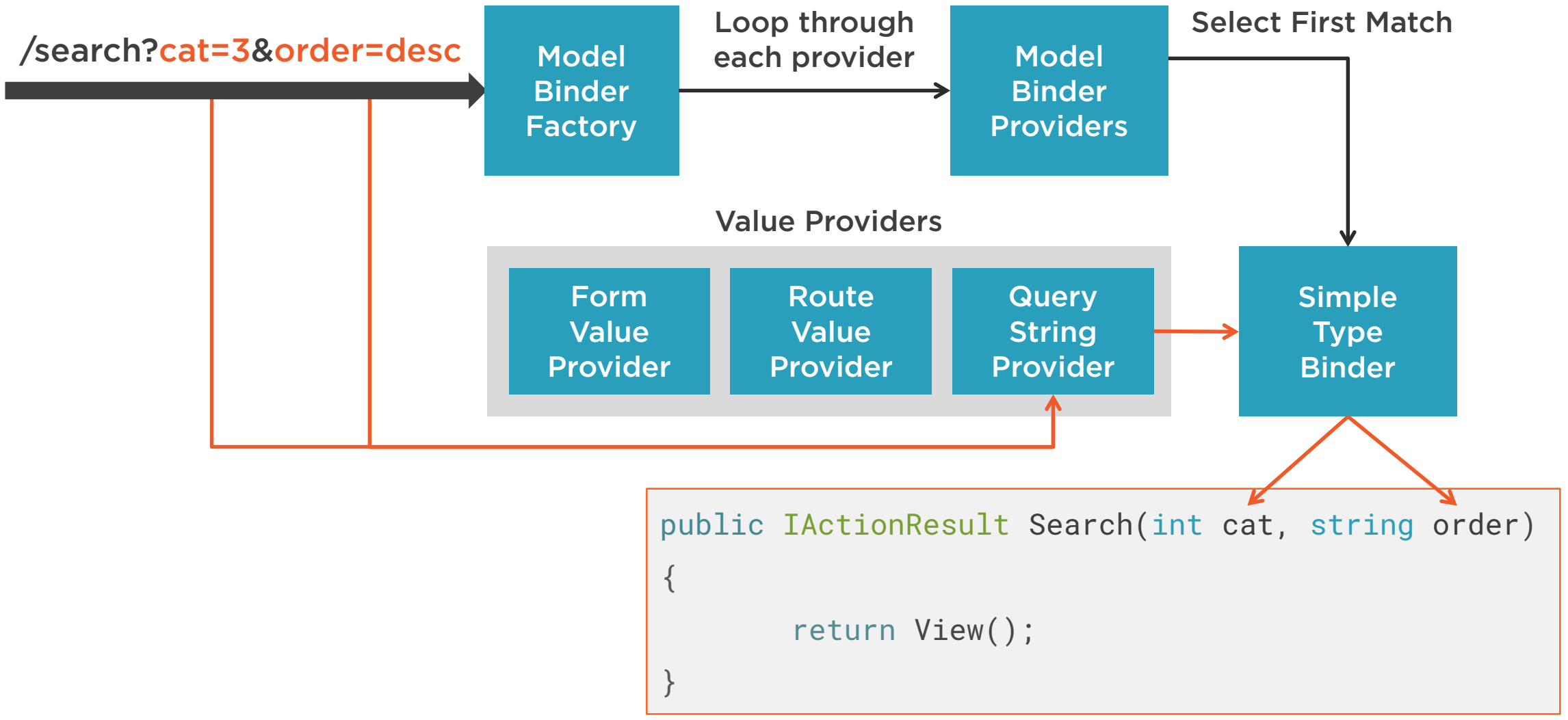
MVC 5

Default Model Binder

MVC Core

- Form Model Binder
- Header Model Binder
- Simple Type Model Binder
- Array Model Binder
- Form File Model Binder
- Dictionary Model Binder
- Other Binders

The Model Binding Process




```
public IActionResult Search(int cat, string order)
{
        return View();
}
```

Understanding Filters

The Four Types of Filters

- Action Filters
- Authorization Filters
- Exception Filters
- Result Filters

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

The IAuthorizationFilter Interface

Validates whether a request is authorized

Can short circuit the life cycle for unauthorized requests

```
public interface IActionFilter : IFilterMetadata
{
    void OnActionExecuting(ActionExecutingContext context);
    void OnActionExecuted(ActionExecutedContext context);
}
```

The IActionFilter Interface

Runs logic before and after an Action Method executes

Can also short circuit the request by providing an Action Result

```
public interface IExceptionFilter : IFilterMetadata
{
    void OnException(ExceptionContext context);
}
```

The IExceptionFilter Interface

Executes when an error occurs during Action Method execution

```
public interface IFilterMetadata
{
    // No Members
}
```

The IFilterMetadata Interface

A common reference point for all types of filters

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

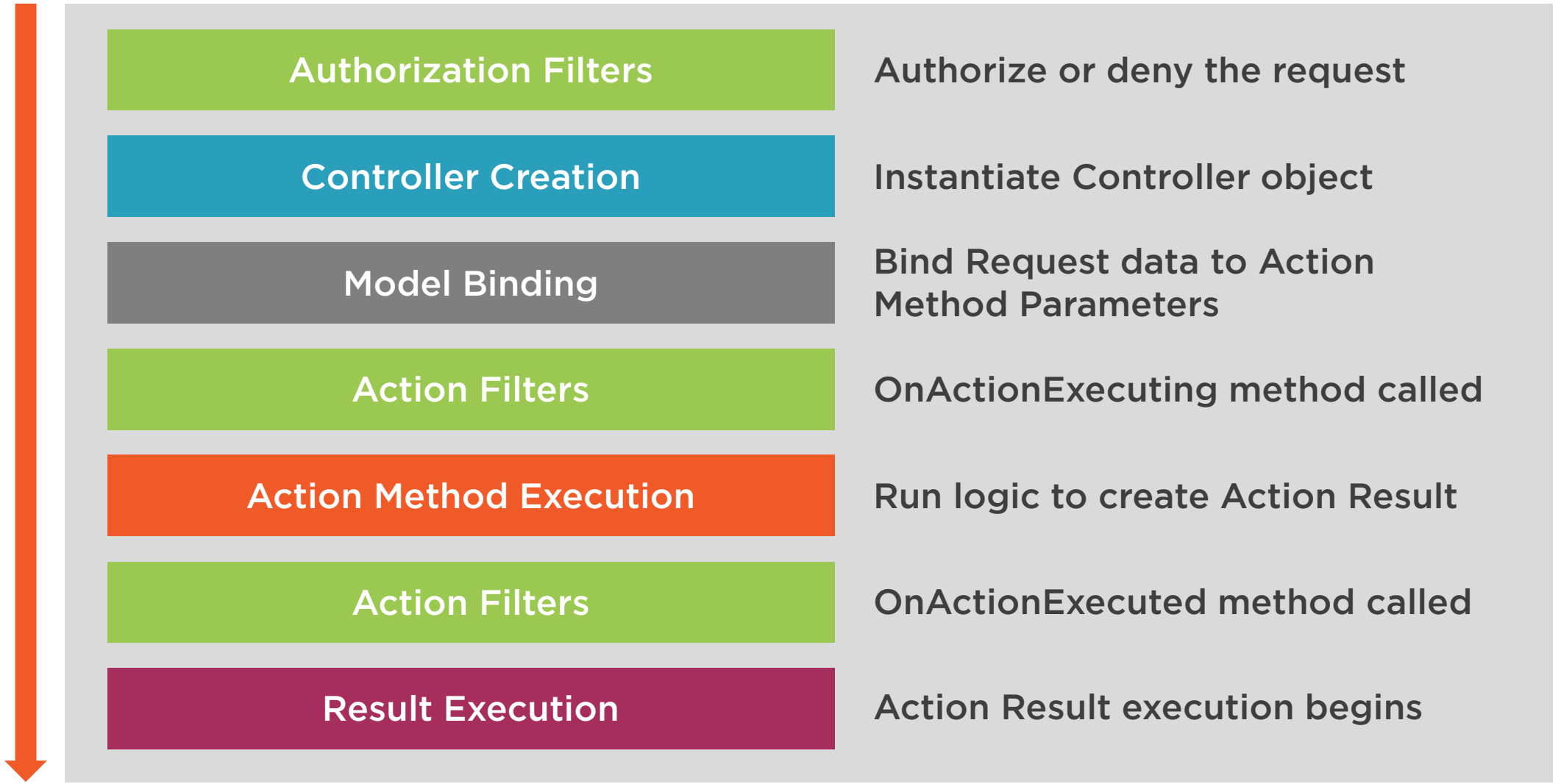
- Authorization attribute applied to Controller
- Action Filter applied to a single Action Method

Synchronous and Asynchronous Filters

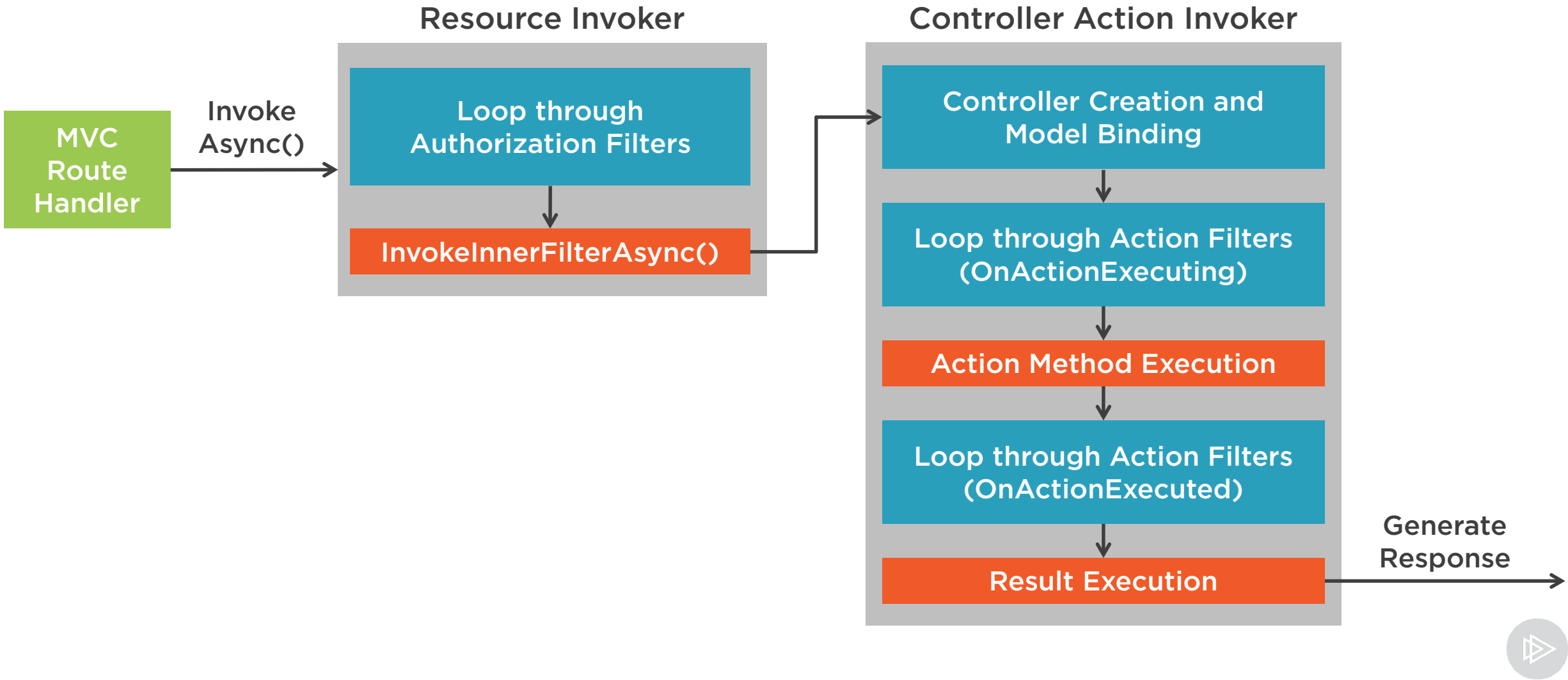
Synchronous	Asynchronous
IActionFilter	IAsyncActionFilter
IAuthorizationFilter	IAsyncAuthorizationFilter
IExceptionFilter	IAsyncExceptionFilter
IResultFilter	IAsyncResultFilter

Filter Execution In Depth

The Action Method Execution Process



The Filter Execution Pipeline



The Resource and Controller Action Invokers

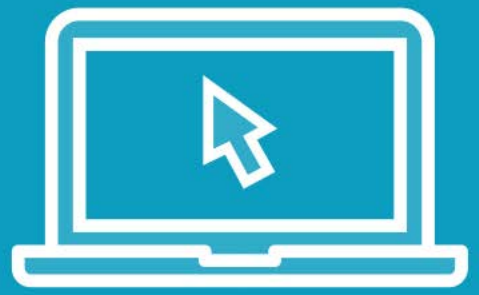
Resource Invoker

Parent class that manages Authorization Filters

Controller Action Invoker

Child class that manages Controller Creation, Action Filters and Action Method Execution

Demo



Working with Authorization Filters

Summary

Authorization Filters validate whether a request is valid before Controller creation

Model Binding maps request data to Action Method parameters

Model Binder Providers determine which Model Binder to use to map parameters

Action Filters can inject logic before and after Action Method execution

Action Methods produce an Action Result that is consumed by the MVC Framework