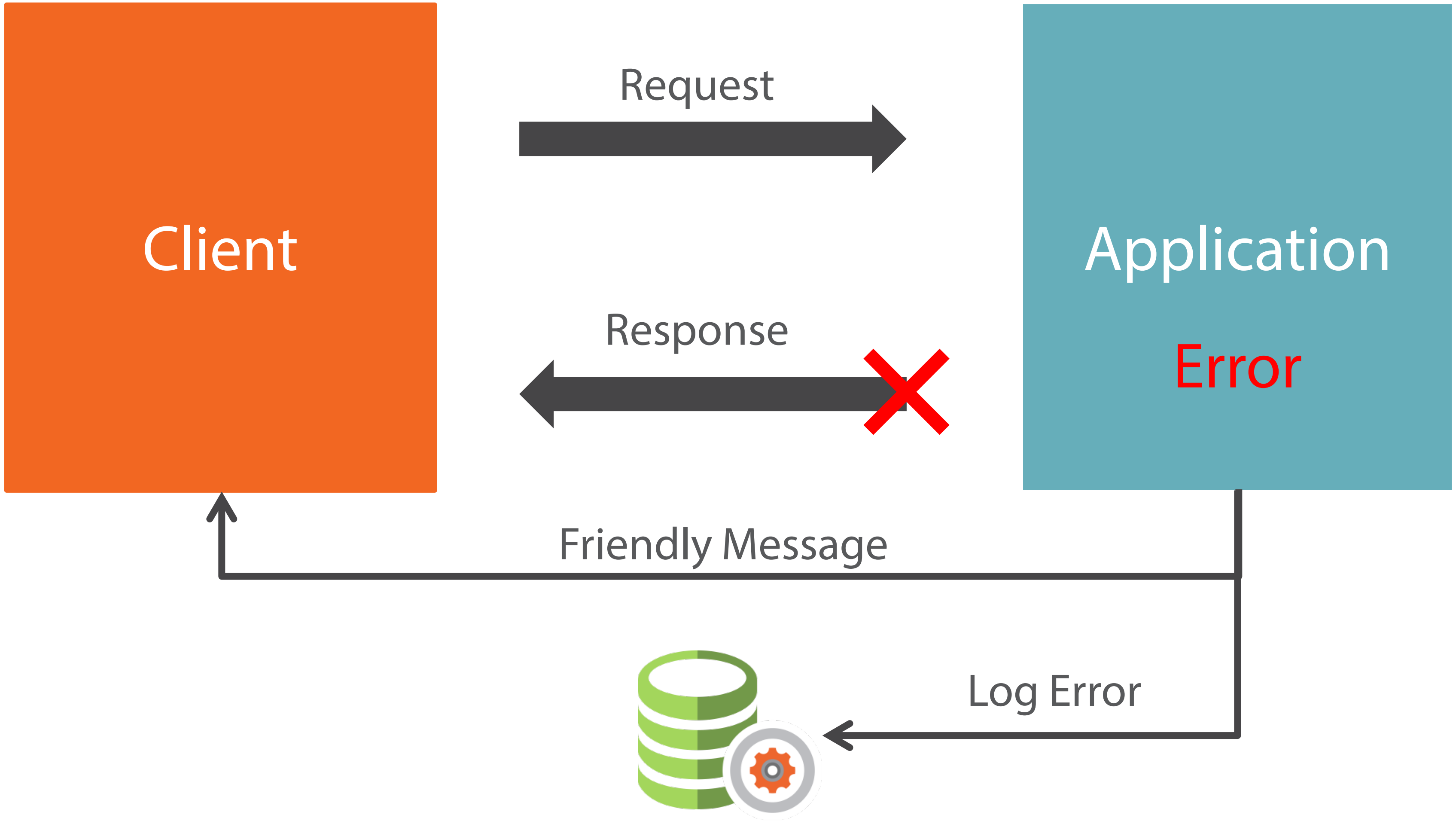
Improving Error Handling with Custom Exception Filters

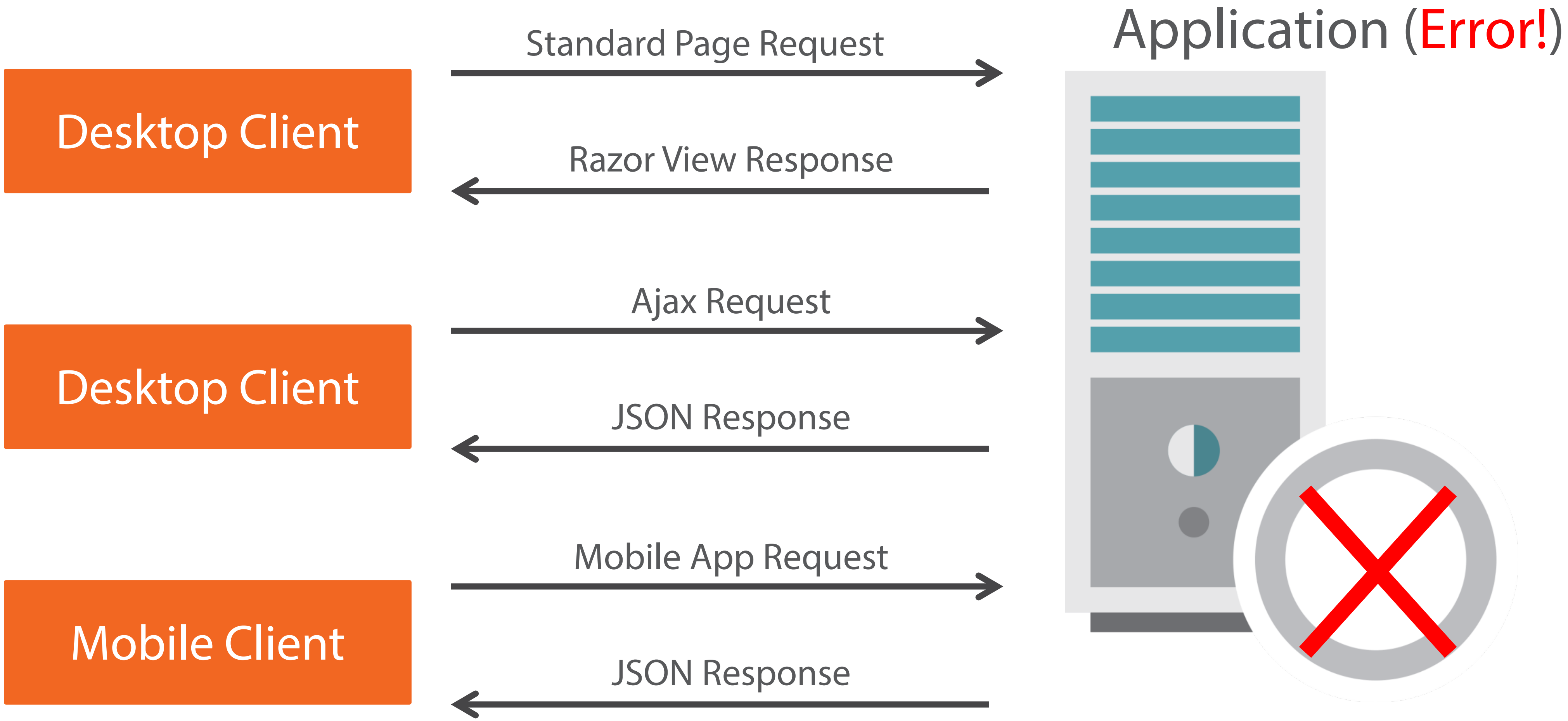
Alex Wolf

www.alexwolfthoughts.com

Handling and Logging Exceptions



Different Requests, Different Responses



Application (Error!)

Desktop Client

Standard Page Request

Razor View Response

Desktop Client

Ajax Request

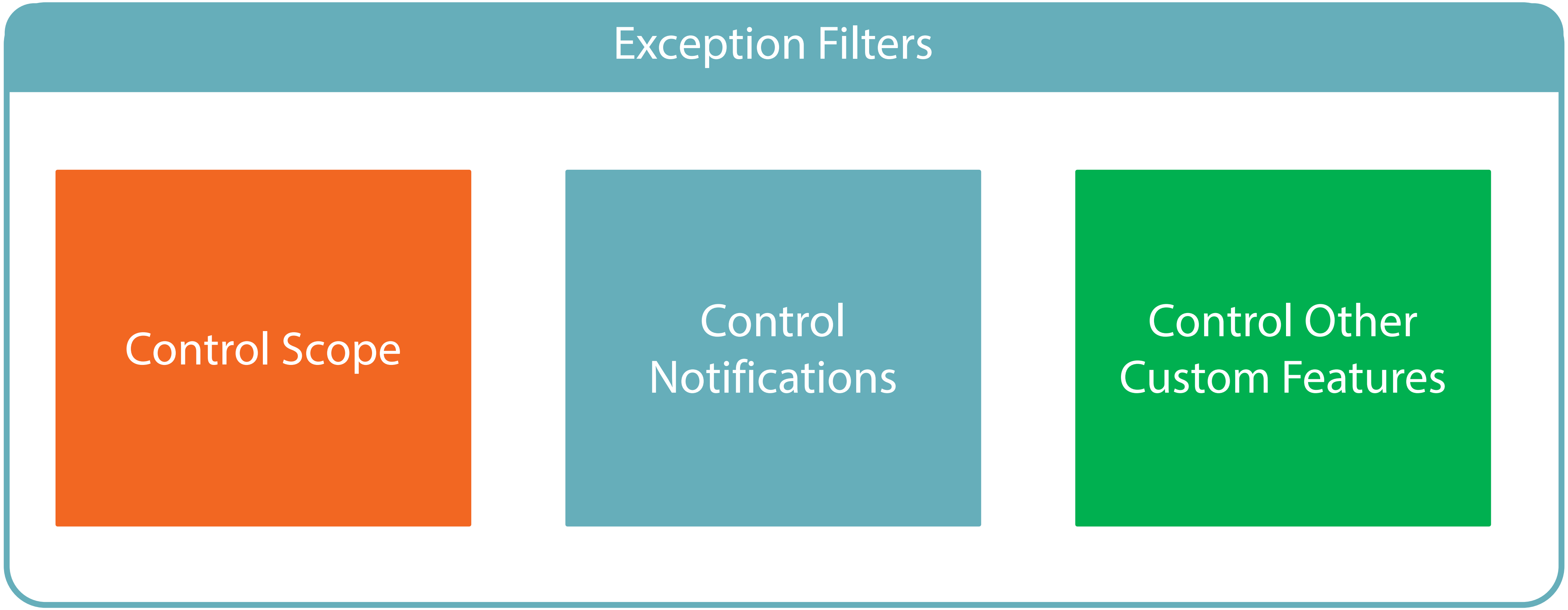
JSON Response

Mobile Client

Mobile App Request

JSON Response

Controlled Exception Handling



Exception Filters

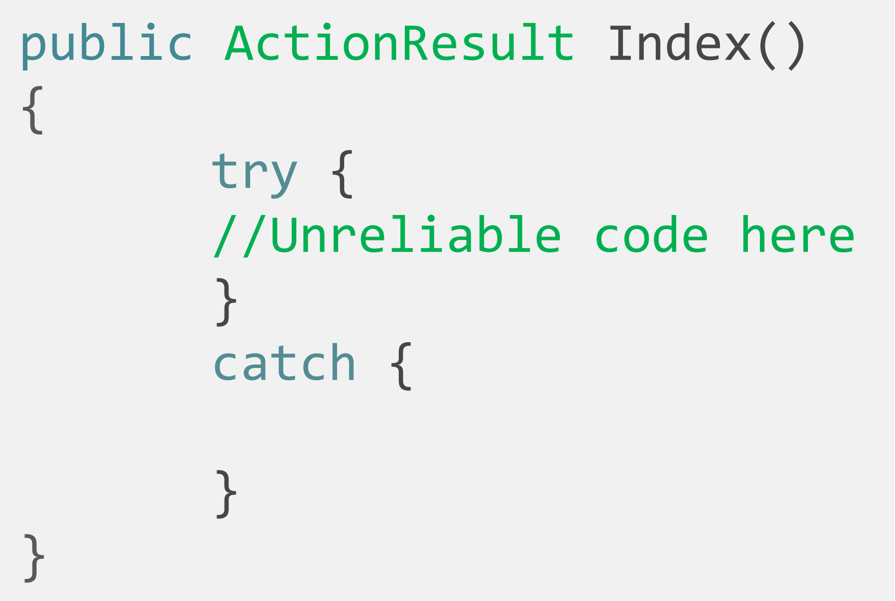
Control Scope

Control Notifications

Control Other Custom Features

Understanding Exception Filters

Exception Handling Strategies



```
public ActionResult Index()
{
        try {
        //Unreliable code here
        }
        catch {

        }
}
```

```
public ActionResult Index()
{
        try {
        //Risky code here
        }
        catch {

        }
}
```

```
public ActionResult Index()
{
        try {
        //Risky code here
        }
        catch {

        }
}
```

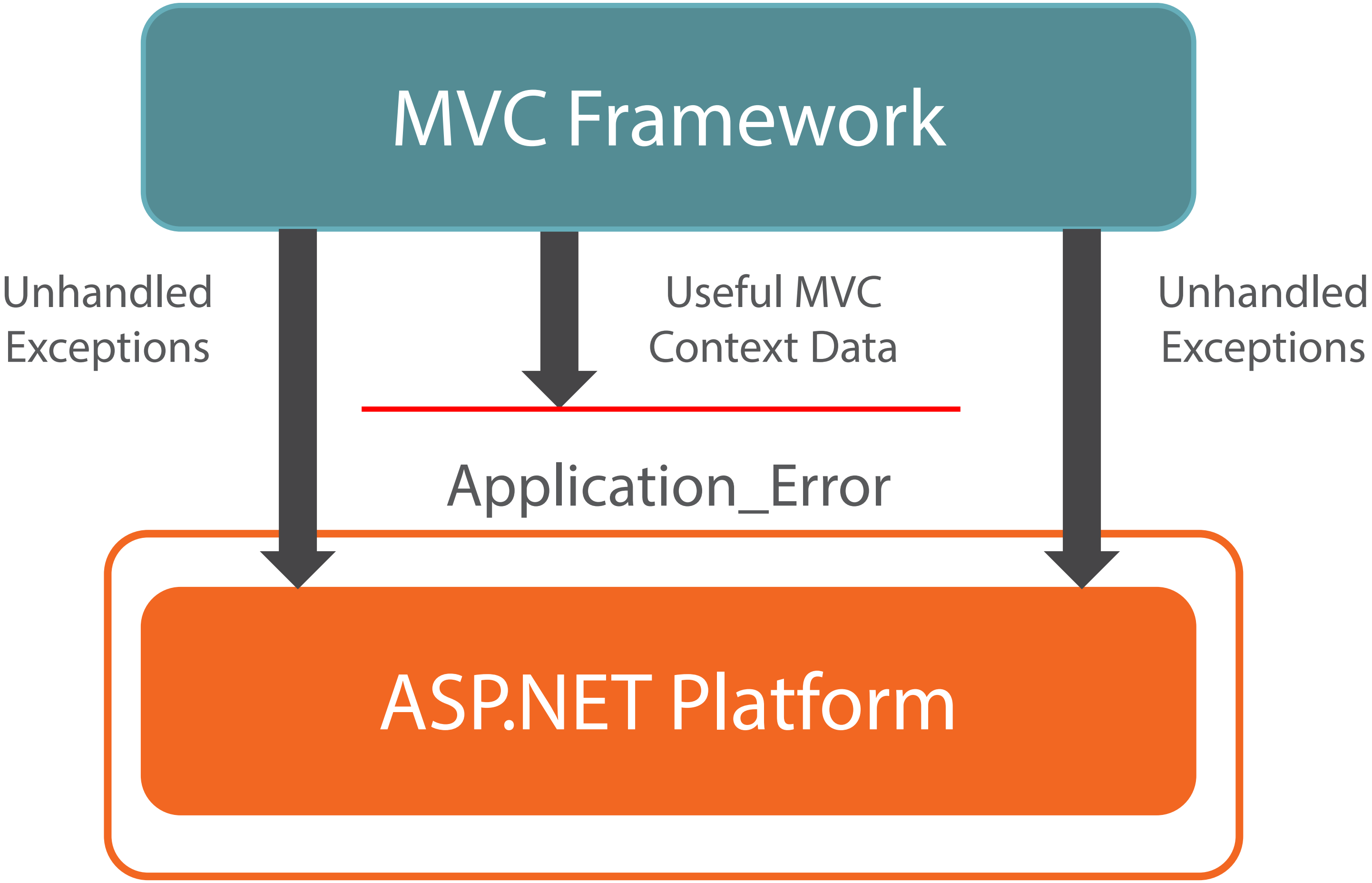
```
public ActionResult Index()
{
        try {
        //Risky code here
        }
        catch {

        }
}
```

```
public ActionResult Index()
{
        try {
        //Risky code here
        }
        catch {

        }
}
```

Application Level Error Handling



MVC Framework

Unhandled Exceptions

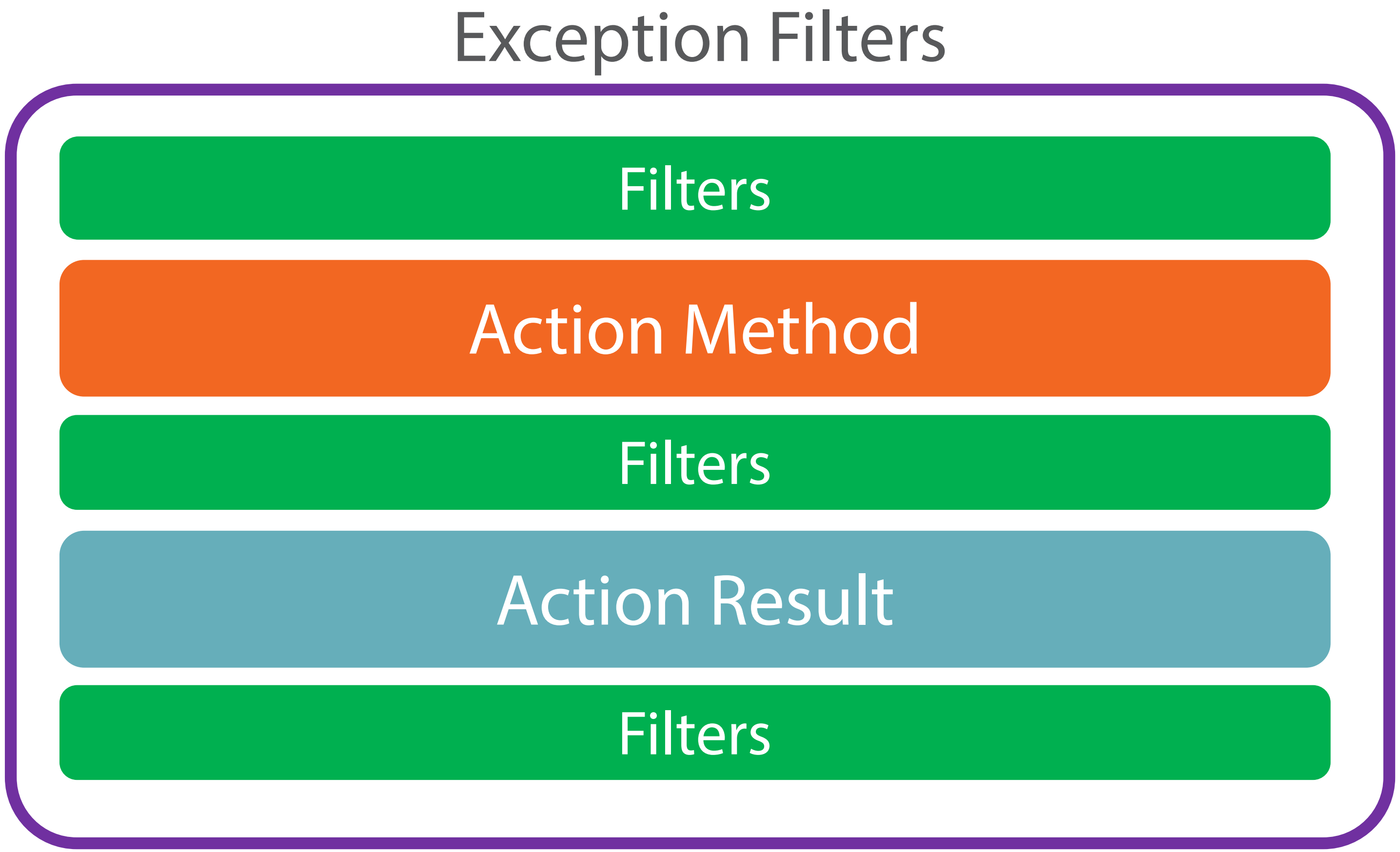
Useful MVC Context Data

Unhandled Exceptions

Application_Error

ASP.NET Platform

Exception Filters and the MVC Life Cycle



Exception Filters

Filters

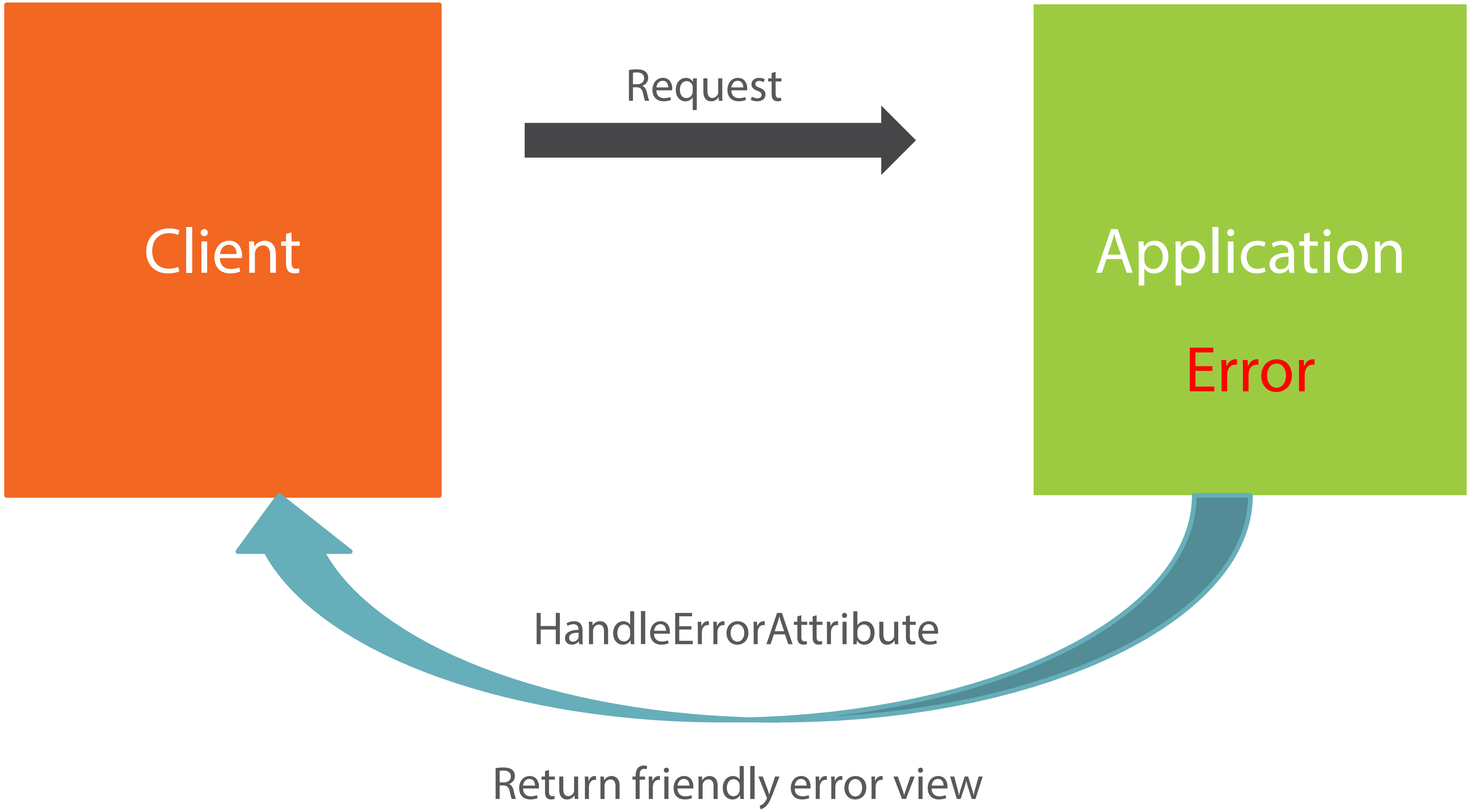
Action Method

Filters

Action Result

Filters

The Default Exception Filter



Client

Request

Application

Error

HandleErrorAttribute

Return friendly error view

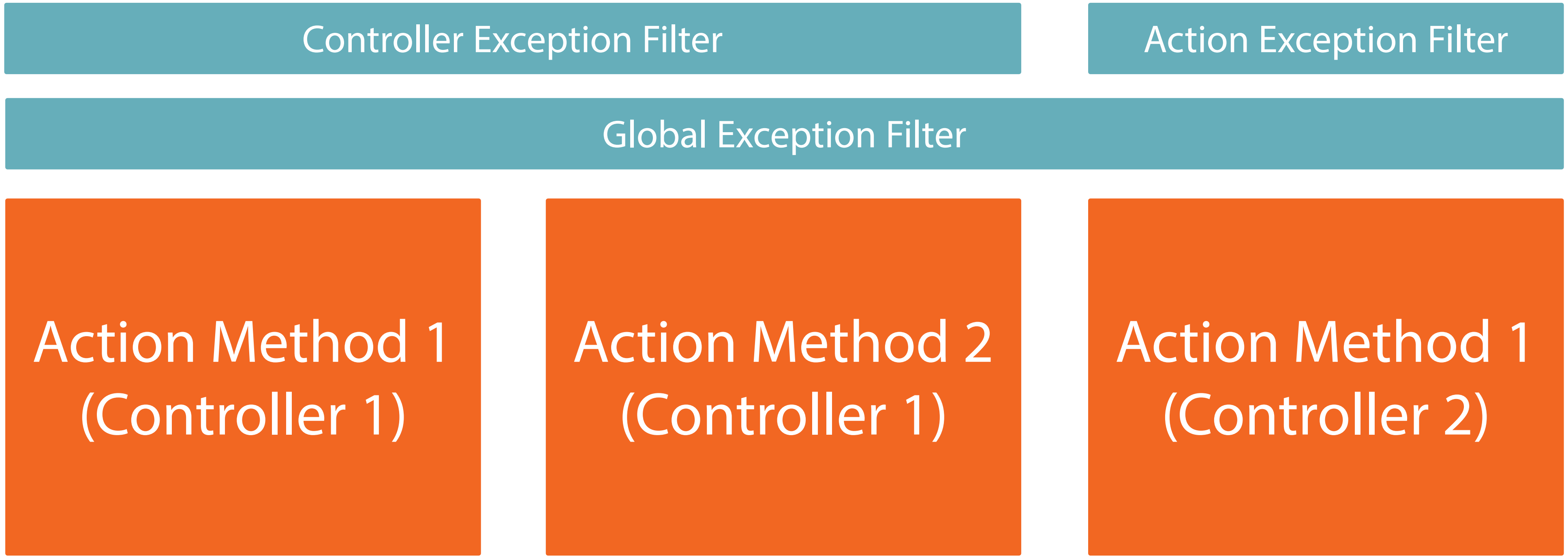
The IExceptionFilter Interface

```
public interface IExceptionFilter
{
        void OnException(ExceptionContext filterContext);
}
```

```
public class ExceptionContext : ControllerContext
{
        public ExceptionContext(ControllerContext controllerContext, Exception exception);

        public virtual Exception Exception { get; set; }
        public bool ExceptionHandled { get; set; }
        public ActionResult Result { get; set; }
}
```

Exception Filter Execution Scope



Controller Exception Filter

Action Exception Filter

Global Exception Filter

Action Method 1 (Controller 1)

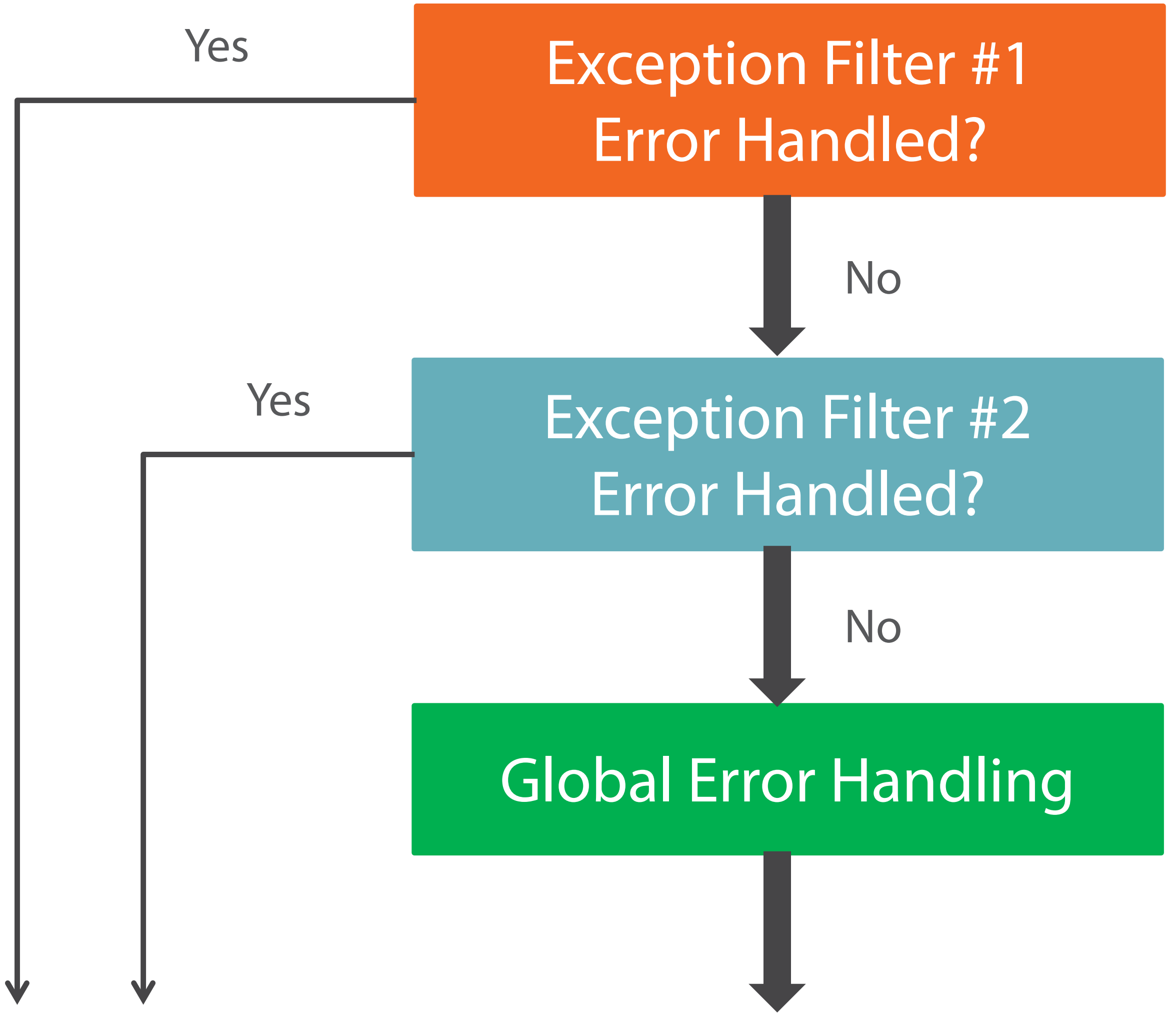
Action Method 2 (Controller 1)

Action Method 1 (Controller 2)

Demo

Building a Custom Exception Filter

Exception Filter Execution Order



Summary

- Exception Filters provide flexibility for error handling
- Can Handle errors in the scope of Action Method execution
- Provide access to more contextual framework information
- Global Error handling should still be used as a fallback