

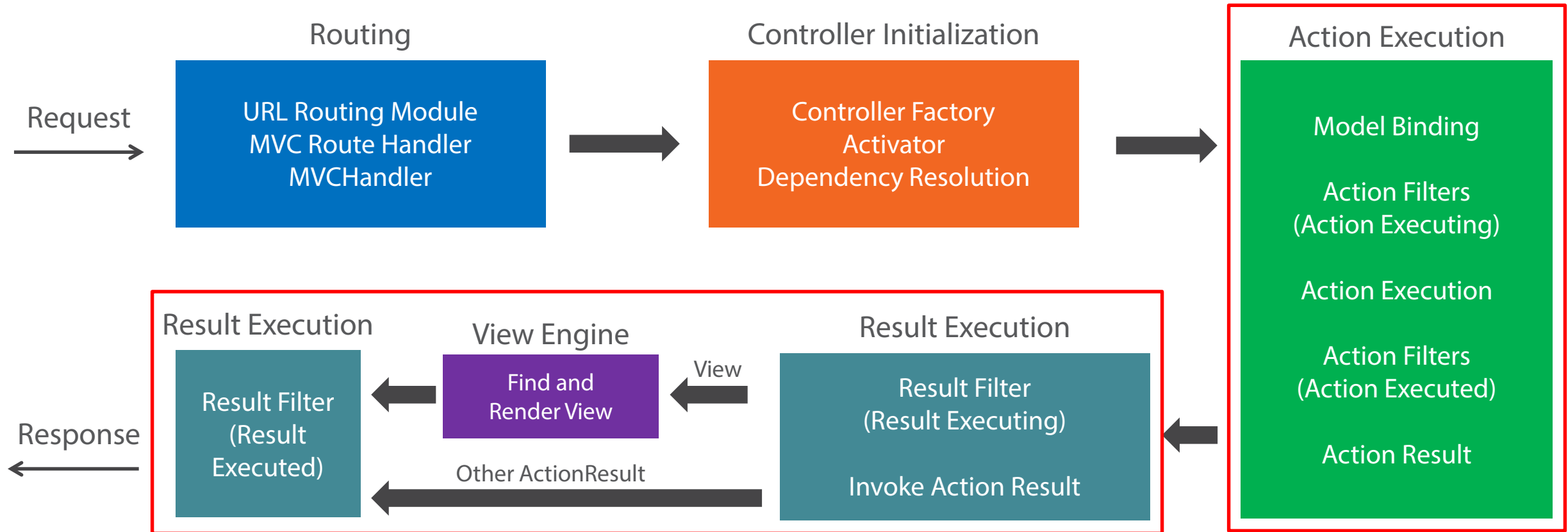
Exploring Action Results and the View Engine



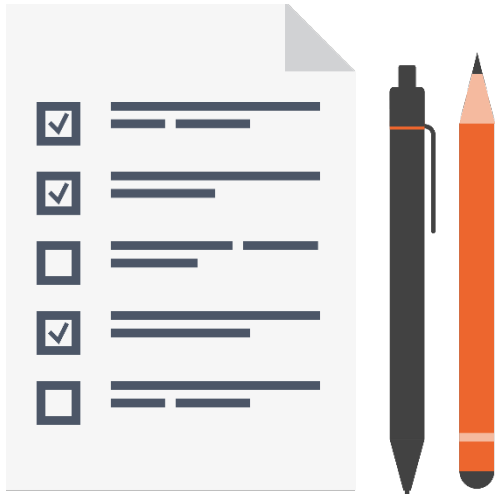
Alex Wolf

www.alexwolfthoughts.com

The **MVC** Request Life Cycle



Module Outline



- Action Results and the Life Cycle
- Building a Custom Action Result
- View Results and the View Engine
- Extending View Engine functionality



View Results and the View Engine

View Engines and the Request Life Cycle

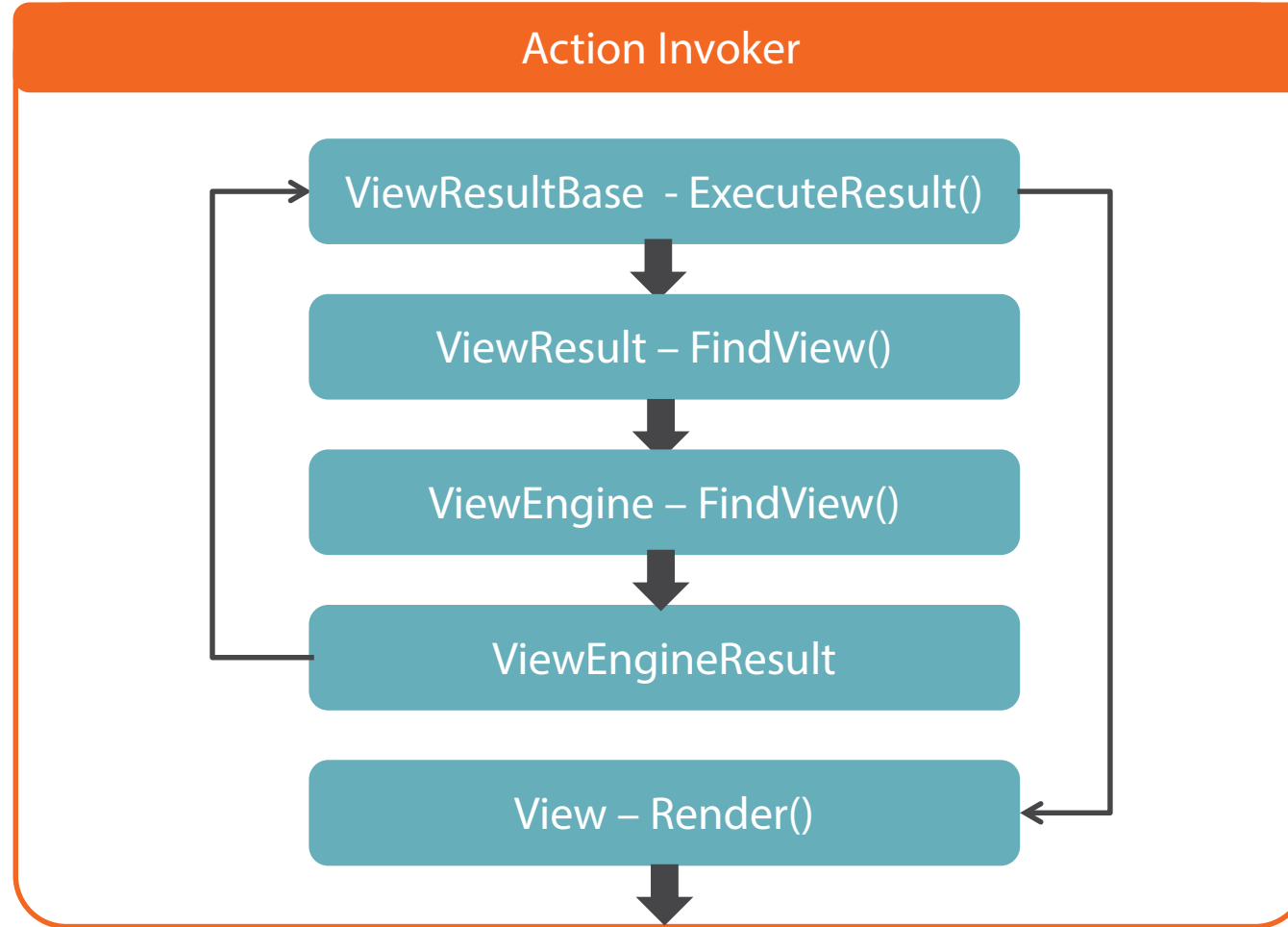
Razor View
Engine

Legacy View
Engine

General View
Engine Concept



The **View Result** Execution Process



The **IViewEngine** Interface

```
public interface IViewEngine()
{
    ViewEngineResult FindView(ControllerContext controllerContext, string viewName,
        string masterName, bool useCache);

    ViewEngineResult FindPartialView(ControllerContext controllerContext, string
        partialViewName, bool useCache);

    void ReleaseView(ControllerContext controllerContext, IView view);
}
```

The Default Razor Search Locations

~/Views/{1}/{0}.cshtml

~/Views/Shared/{0}.cshtml

~/Areas/{2}/Views/{1}/{0}.cshtml

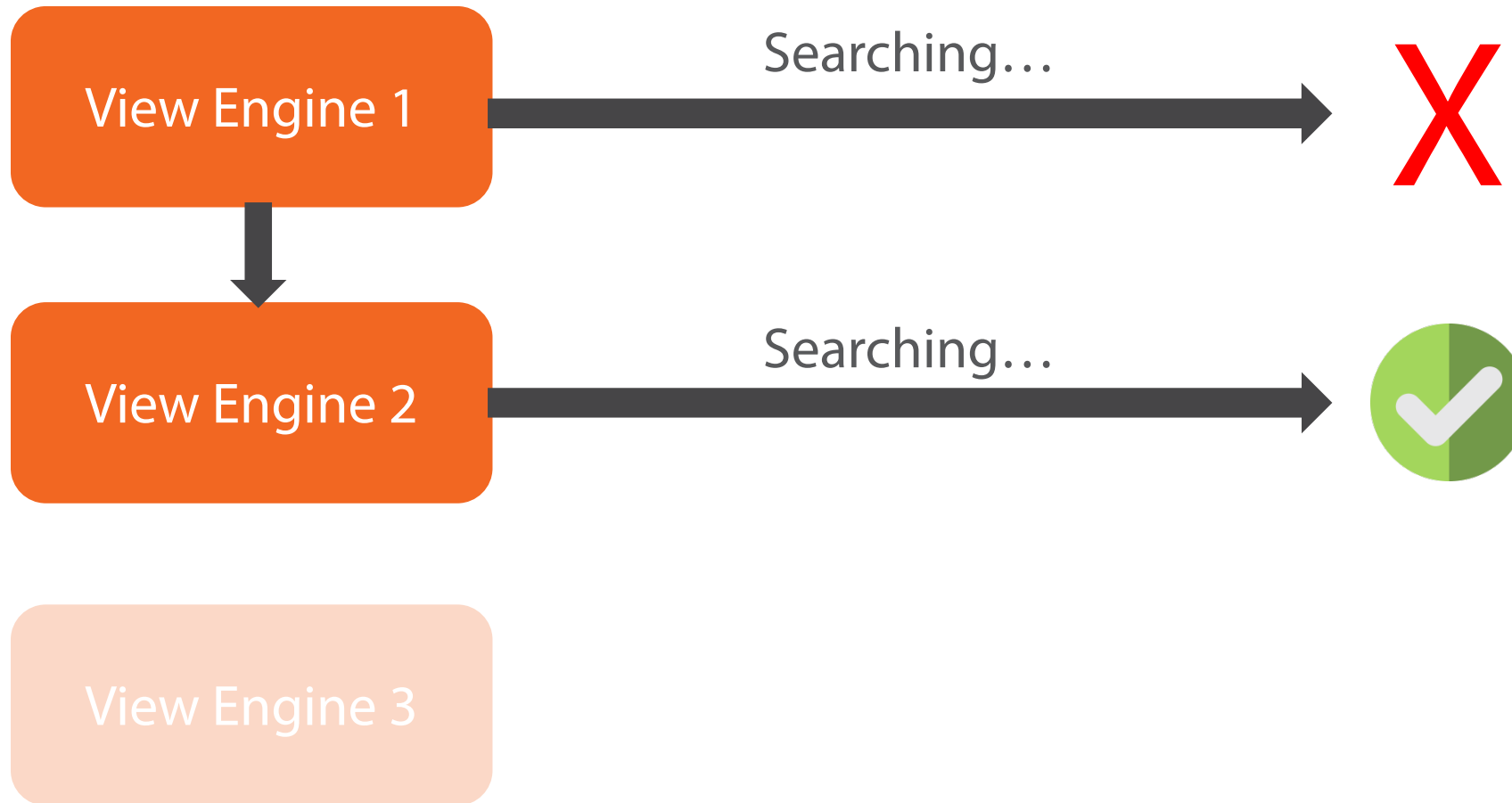
~/Areas/{2}/Views/Shared/{0}.cshtml

0 = Action Name

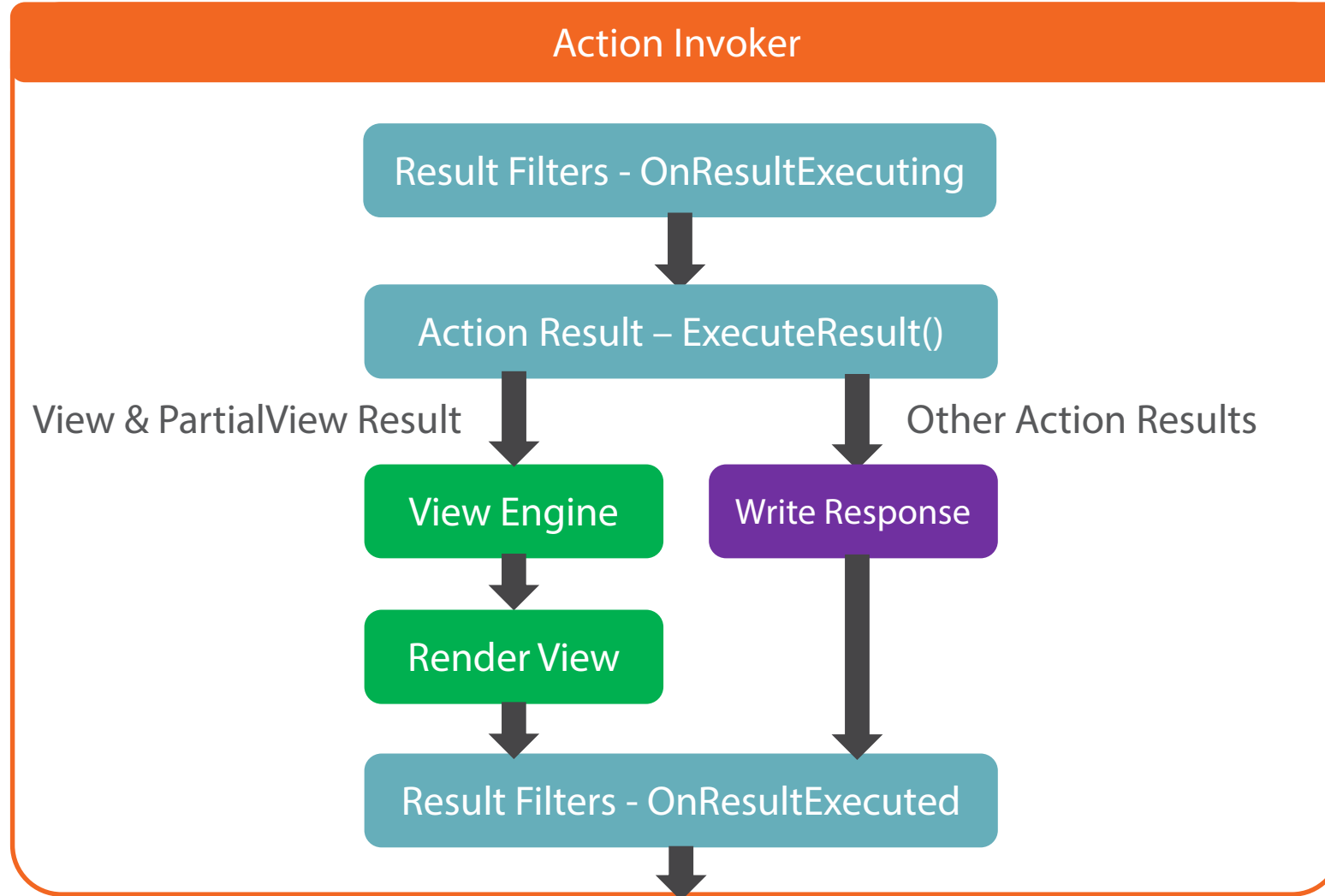
1 = Controller Name

2 = Area Name

The **View Engine** Search Order



The **Action Result** Execution Process



Common Action Result Types

ContentResult

JsonResult

EmptyResult

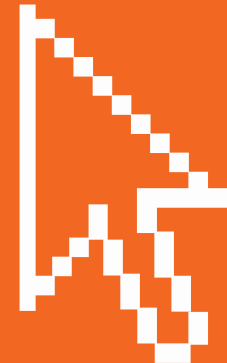
ViewResult

FileResult

RedirectResult

Demo

Building a Custom Action Result



Summary



- Action Results generate a response for the request
- Different Action Result types have different execution implementations
- Custom Action Results can be built for specific scenarios
- View Results rely on the View Engine to find and parse views

The **MVC** Request Life Cycle

