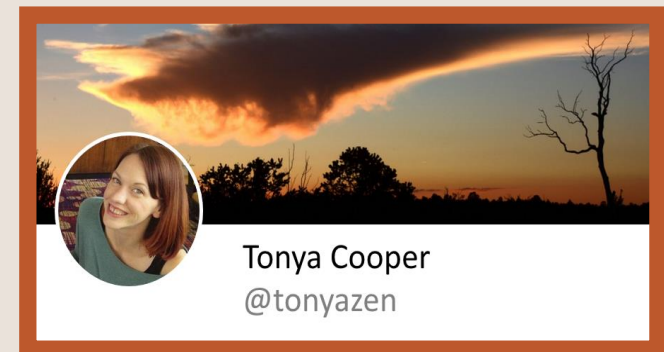


Custom Action Filters

in ASP.NET MVC



TITANIUM SPONSORS



Platinum Sponsors



Gold Sponsors



What are action filters?

An attribute that can be applied to an individual action method on a controller, or to the entire controller.

Filter Types

1. Authorize
2. Action
3. Result
4. Exception

Built-in Filter Attributes

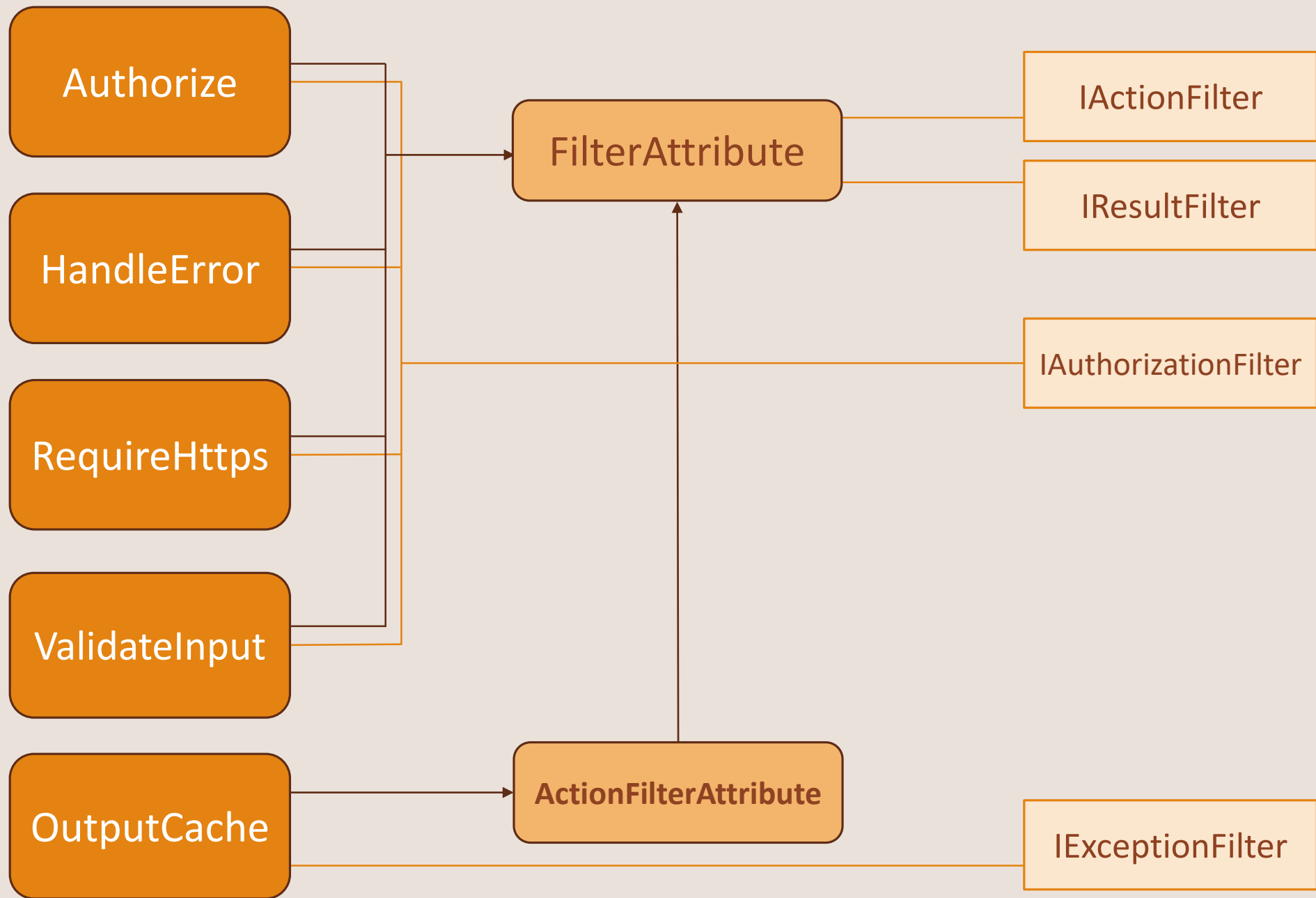
Authorize

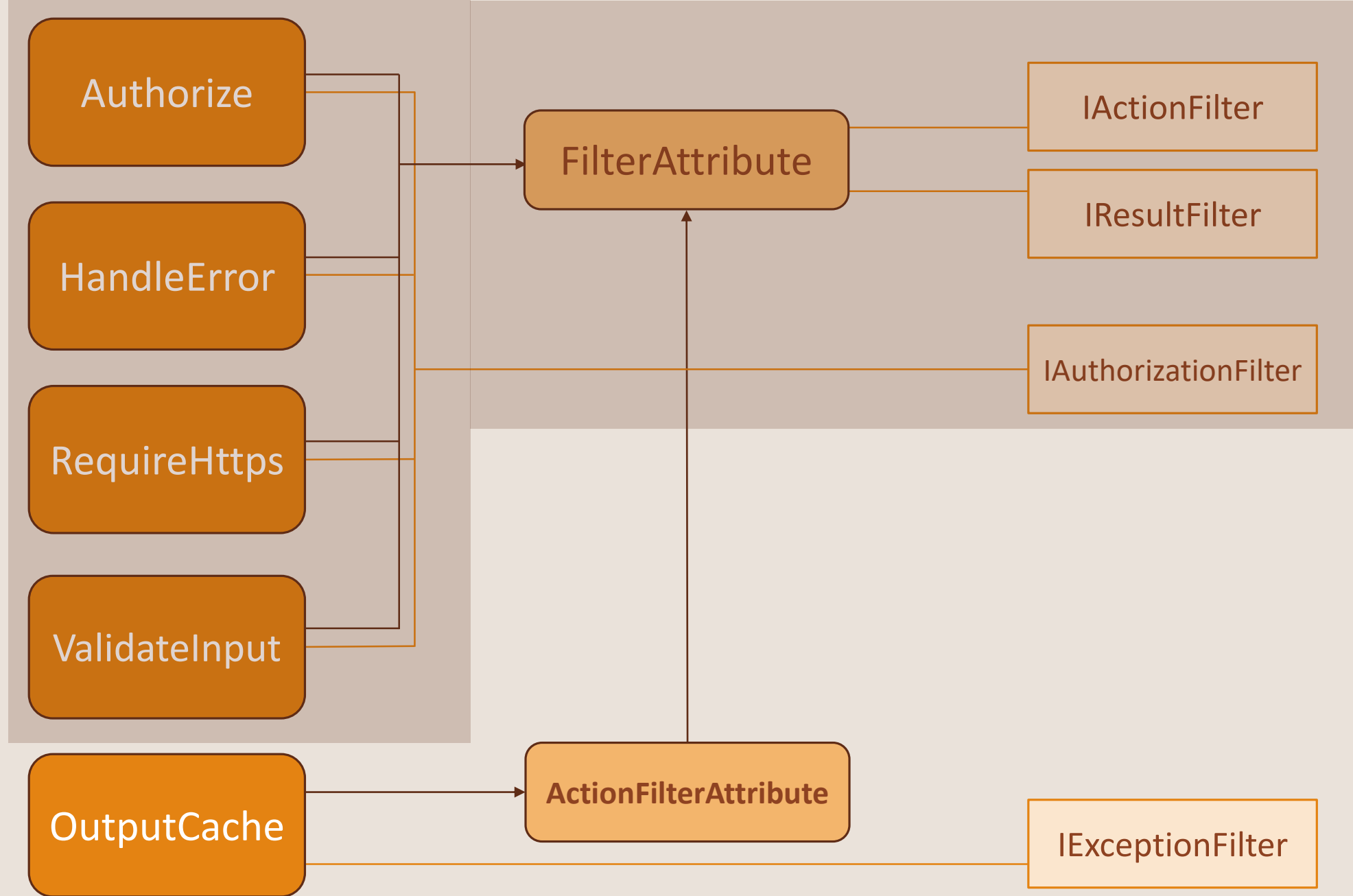
ValidateInput

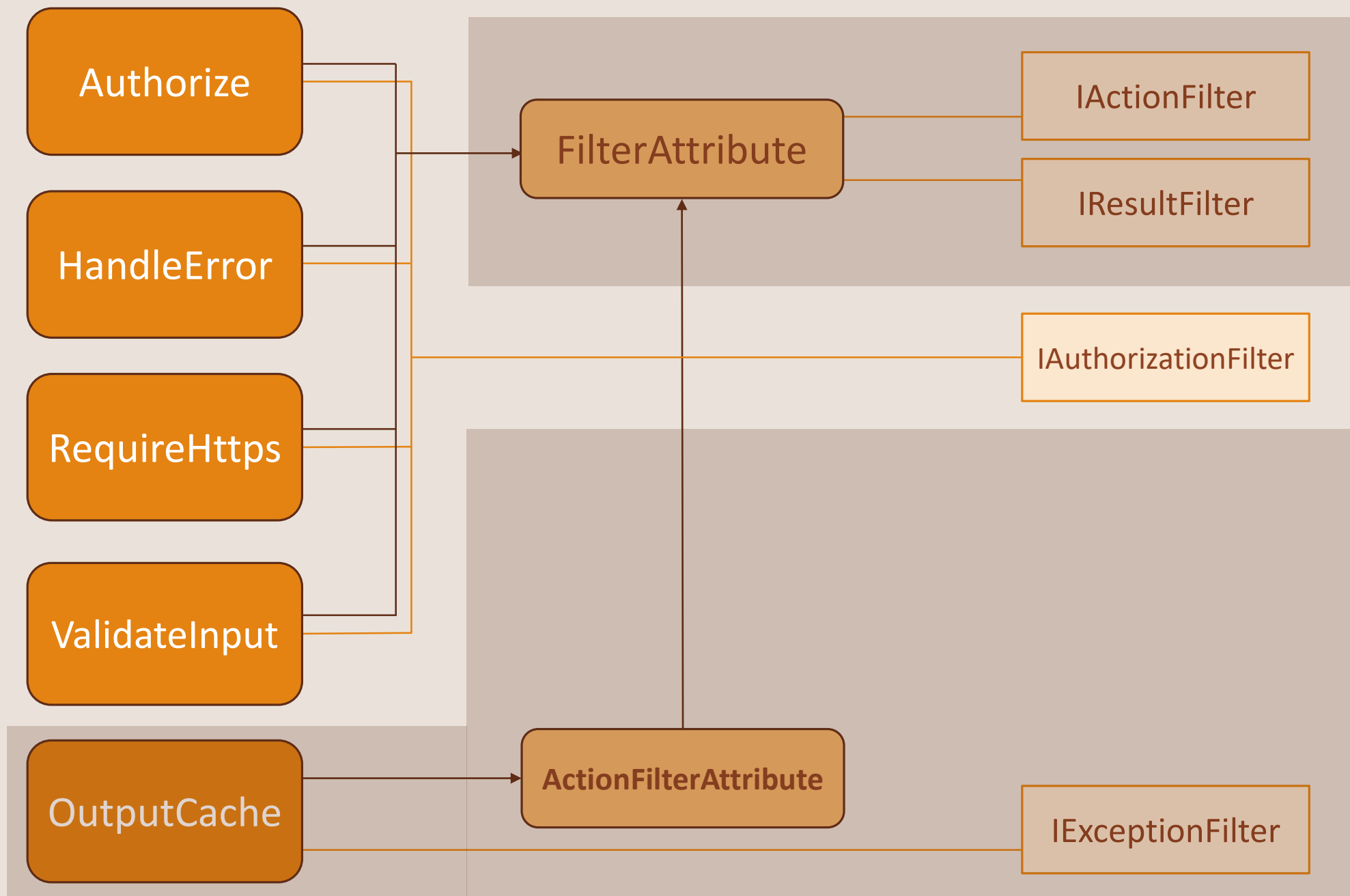
RequireHttps

HandleError

OutputCache







Built-in Filter Attributes

Authorize

ValidateInput

RequireHttps

HandleError

OutputCache

```
[Authorize]
public class AuthorizeAttributeController : Controller
{
    [AllowAnonymous]
    public ActionResult Index()
    {
        return View();
    }

    [Authorize(Roles = "admin")]
    public ActionResult ManageUsers()
    {
        return View();
    }

    [Authorize(Users = "tonyazen")]
    public ActionResult ManageAdmins()
    {
        return View();
    }
}
```

Built-in Filter Attributes

Authorize

ValidateInput

RequireHttps

HandleError

OutputCache

Turn on custom errors in Web.config:

```
<customErrors mode="On">  
    <error statusCode="404" redirect="~/Errors/PageNotFound" />  
</customErrors>
```

FilterConfig.cs

```
namespace CustomActionFilters  
{  
    public class FilterConfig  
    {  
        public static void RegisterGlobalFilters(GlobalFilterCollection filters)  
        {  
            filters.Add(new HandleErrorAttribute());  
        }  
    }  
}
```

Global.asax

```
namespace CustomActionFilters
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            log4net.Config.XmlConfigurator.Configure();
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

Built-in Filter Attributes

Authorize

ValidateInput

RequireHttps

HandleError

OutputCache

```
namespace CustomActionFilters.Controllers
{
    public class OutputCacheAttributeController : Controller
    {
        [OutputCache(Duration = 600)]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

Built-in Filter Attributes

Authorize

ValidateInput

RequireHttps

HandleError

OutputCache


```
namespace CustomActionFilters.Controllers
{
    public class ValidateInputAttributeController : Controller
    {
        [HttpGet]
        public ActionResult MyForm()
        {
            return View();
        }

        [HttpPost]
        [ValidateInput(false)]
        public ActionResult MyForm(MyFormViewModel model)
        {
            return View();
        }
    }
}
```

Built-in Filter Attributes

Authorize

ValidateInput

RequireHttps

HandleError

OutputCache

```
namespace CustomActionFilters.Controllers
{
    public class RequireHttpsAttributeController : Controller
    {
        [HttpPost]
        [RequireHttps]
        public ActionResult SensitiveInformation()
        {
            return View();
        }
    }
}
```

Custom Action Filters

- Share code across action methods
- Share code across controllers
- Custom exception handling
- Logging
- Modify responses
- Save responses

ActionFilterAttribute

1. OnActionExecuting
2. OnActionExecuted
3. OnResultExecuting
4. OnResultExecuted

Action Filters...

In Action

Start Filtering

1. Name your filters appropriately

CustomFilter is NOT a good name.

Start Filtering

1. Name your filters appropriately
2. Apply to one or more actions

Start Filtering

1. Name your filters appropriately
2. Apply to one or more actions
3. Apply to one or more controllers

Start Filtering

1. Name your filters appropriately
2. Apply to one or more actions
3. Apply to one or more controllers
4. Apply globally

FilterConfig.cs

```
public class FilterConfig
{
    public static void RegisterGlobalFilters(GlobalFilterCollection filters)
    {
        filters.Add(new CustomFilter());
        filters.Add(new HandleErrorAttribute());
    }
}
```

Start Filtering

1. Name your filters appropriately
2. Apply to one or more actions
3. Apply to one or more controllers
4. Apply globally
5. Specify the order

```
[CustomFilter(Order = 1)]  
[CustomFilter2(Order = 2)]  
public ActionResult About()  
{  
    return View();  
}
```

Custom Action Filters in ASP.NET MVC

Demo: <https://github.com/tonyazen/CustomActionFilters>

