

The Application and Request Life Cycle Events



Alex Wolf

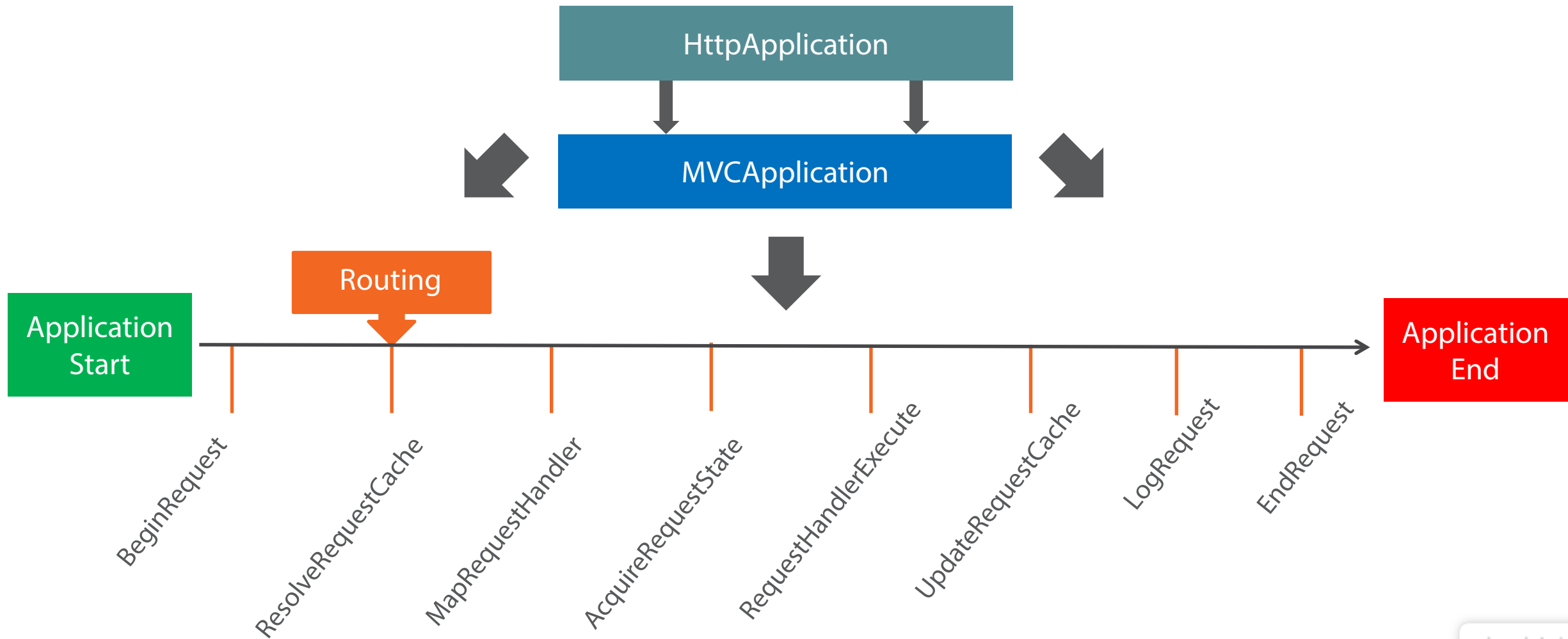
www.alexwolfthoughts.com

Course Outline



- The Application Start and Application End events
- Applying initial configurations
- MVC integration with the Request Life Cycle events
- Leveraging the life cycle events

The Life of an Application



Application Start

- An event that fires when first request is received
- Can be used to run initial configuration code

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    BundleConfig.RegisterBundles(BundleTable.Bundles);
}
```

Registering Routes

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```



RouteHandler

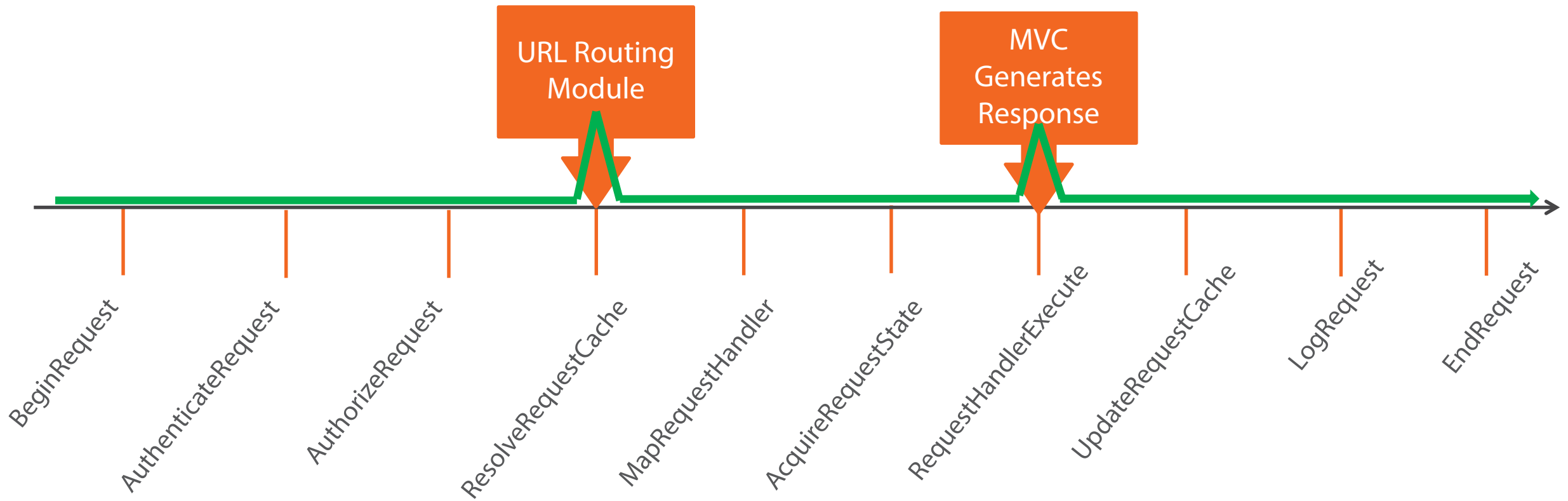
Application End

- An event that fires when an application ends
- Not a great option for handling application errors

PreApplicationStart

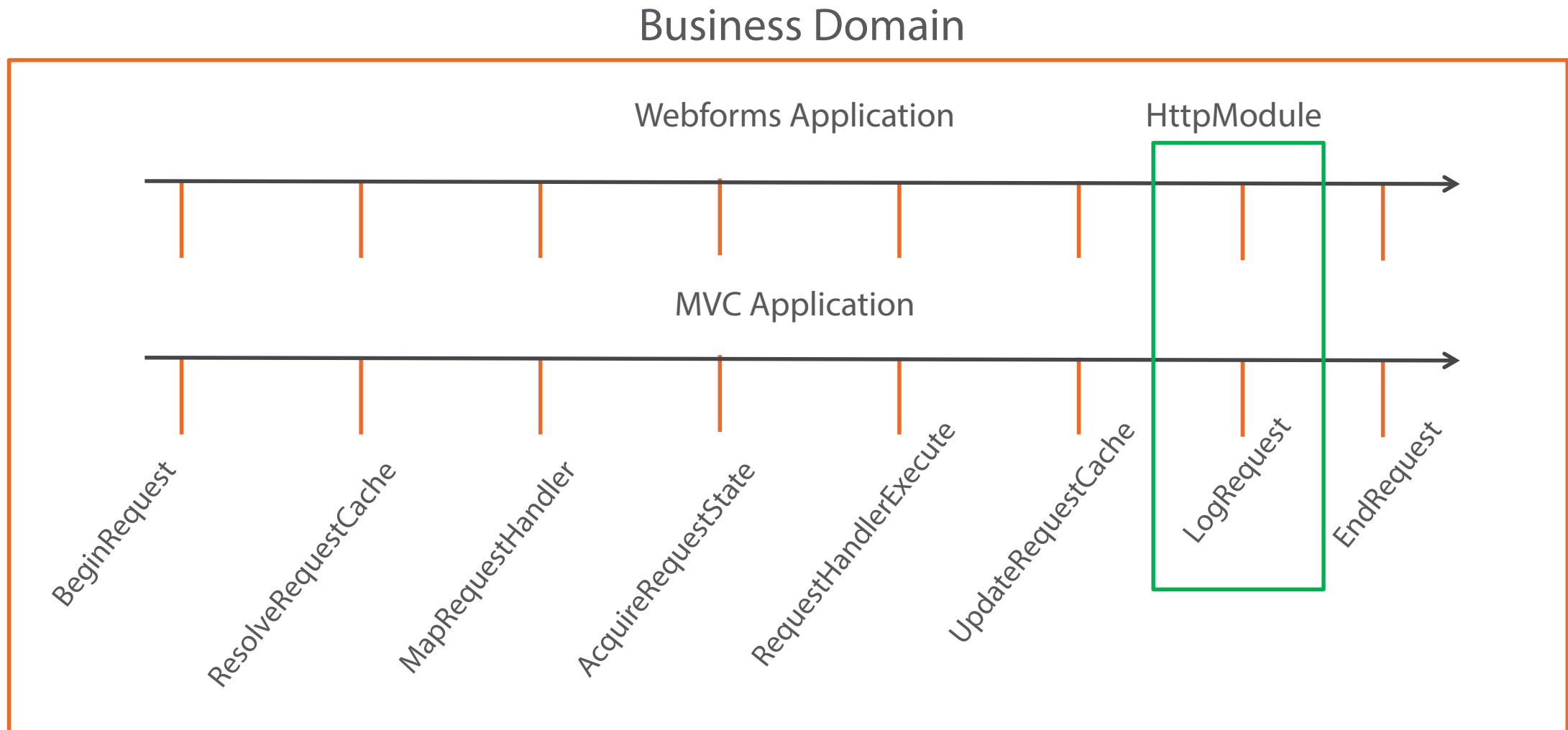
- Another option for running initial configuration code
- Applied through an attribute
- Often used to register modules

The Life of a Request

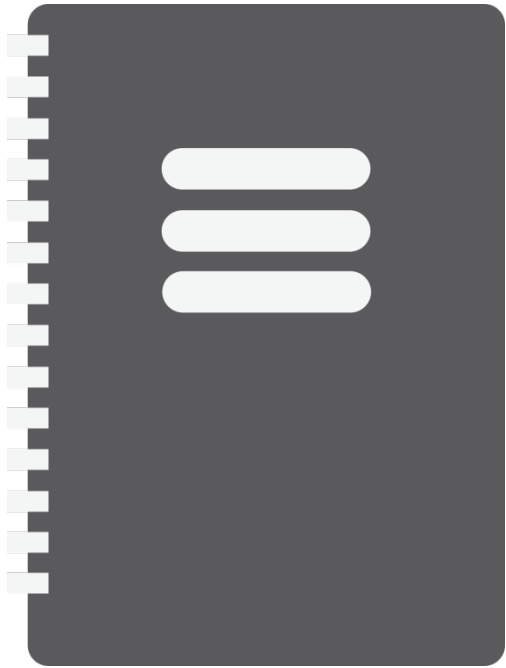


* Event names shows represent both pre and post versions

The **Scope** of Life Cycle



Summary



- The pipeline begins with `Application_Start`
- `PreApplicationStartMethod` allows early configurations
- Events, `HttpHandlers`, and `HttpModules` support and enable execution of the MVC framework
- Some of these components can be shared across requests and even frameworks