Adding Application Configuration to the Host

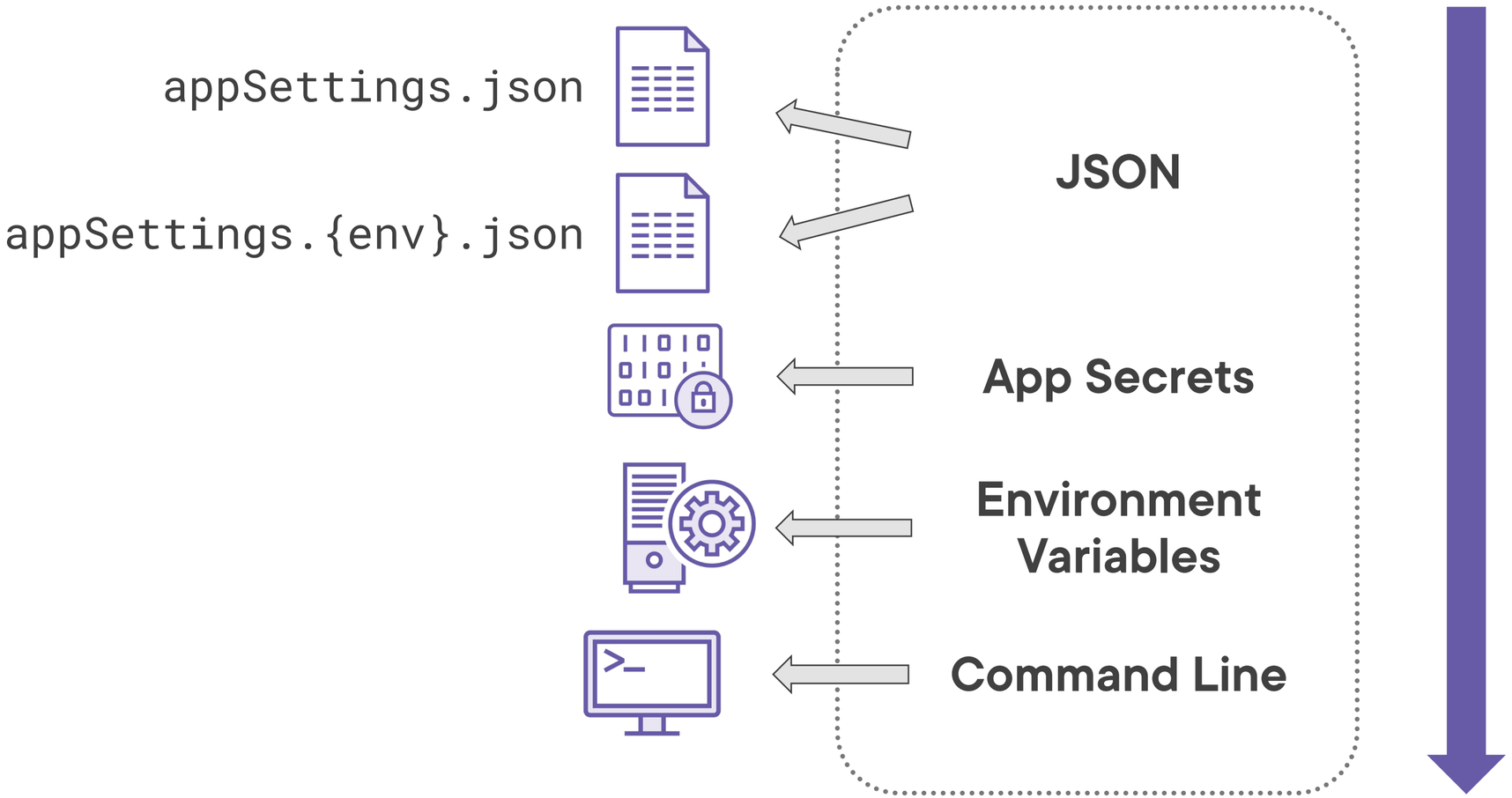
Matthew Tester
Technology Consultant

@matttester matthewtester.com

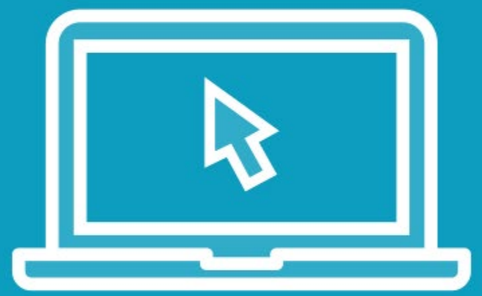
CreateDefaultBuilder

```
public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureServices((hostContext, services) =>
            {
                services.AddHostedService<ImageFileWatcher>();
            });
```

Default Application Configuration



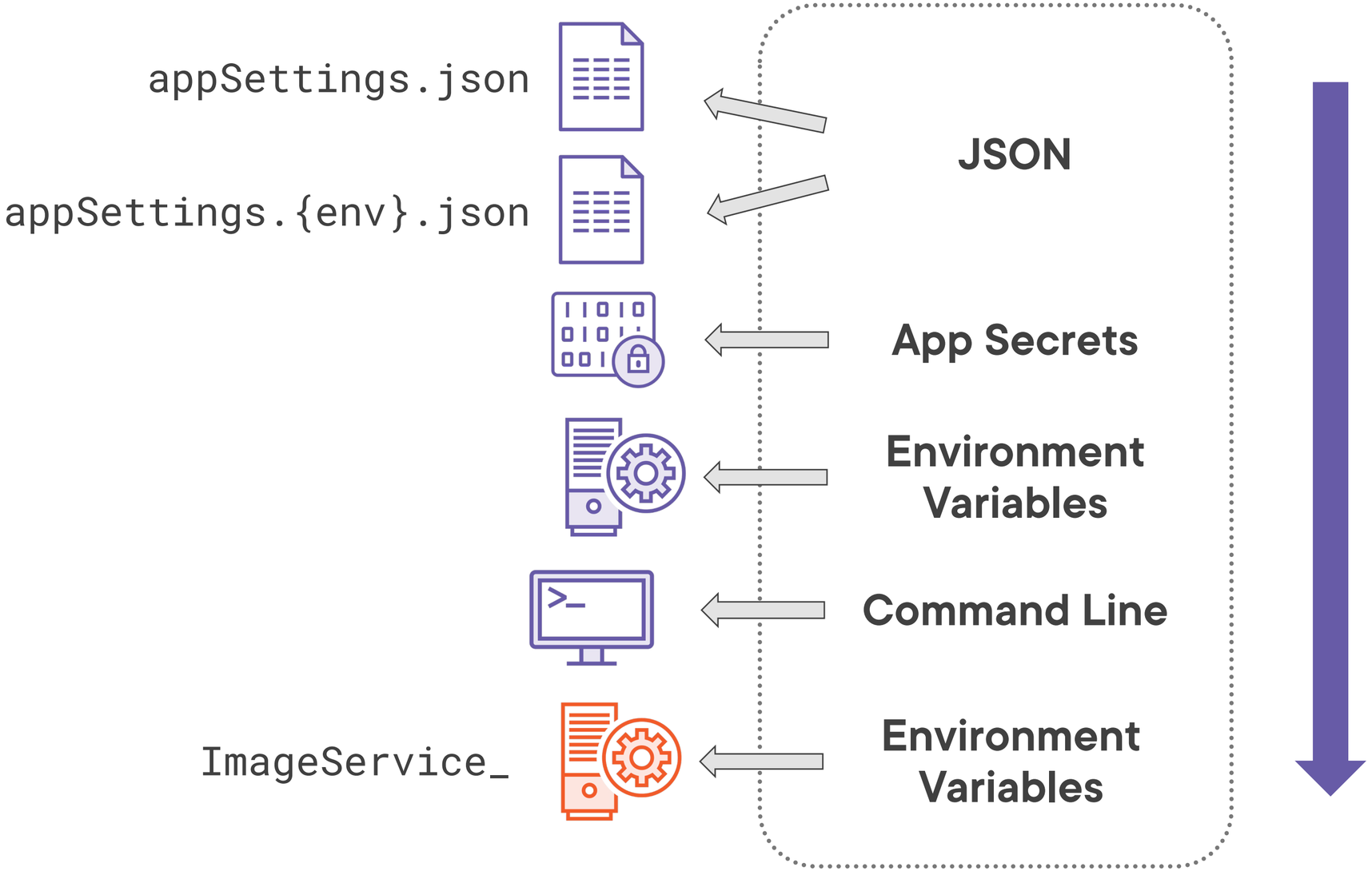
Demo



Adding configuration providers to Host

- Use environment variables with a known prefix

Application Configuration



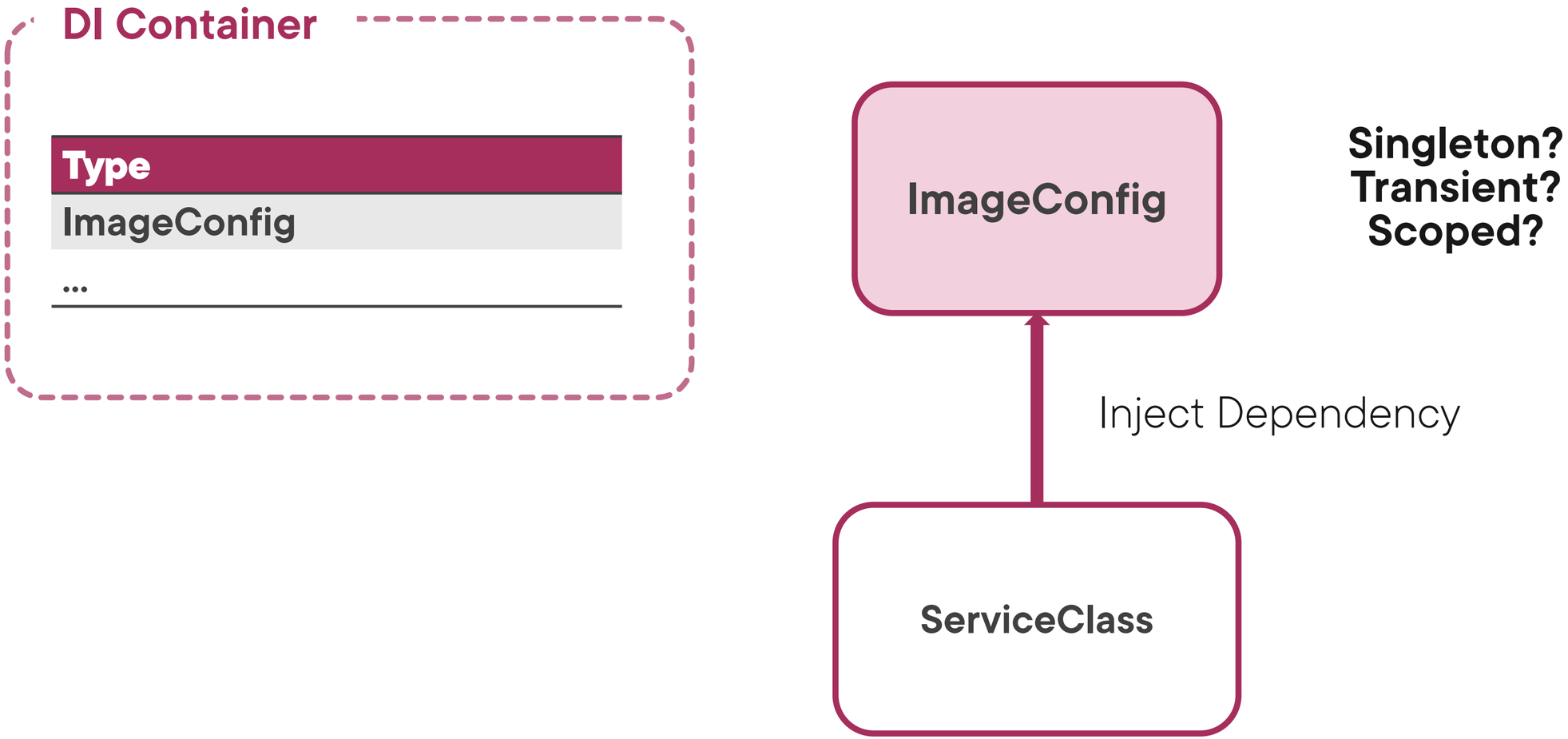
Injecting Configuration

```
public ImageFileWatcher(
    ILogger<ImageFileWatcher> logger,
    IConfiguration configuration)
{
    _logger = logger;
    _configuration = configuration;
}
```

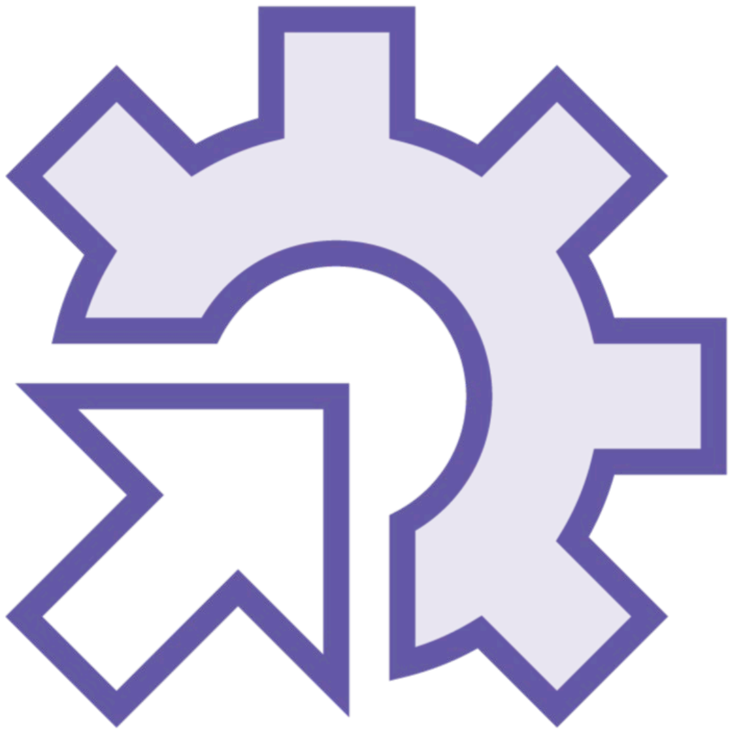
Injecting Configuration

```
public ImageFileWatcher(
    ILogger<ImageFileWatcher> logger,
    ImageConfig imageConfig)
{
    _logger = logger;
    _imageConfig = imageConfig;
}
```

Injecting Configuration



Options Pattern

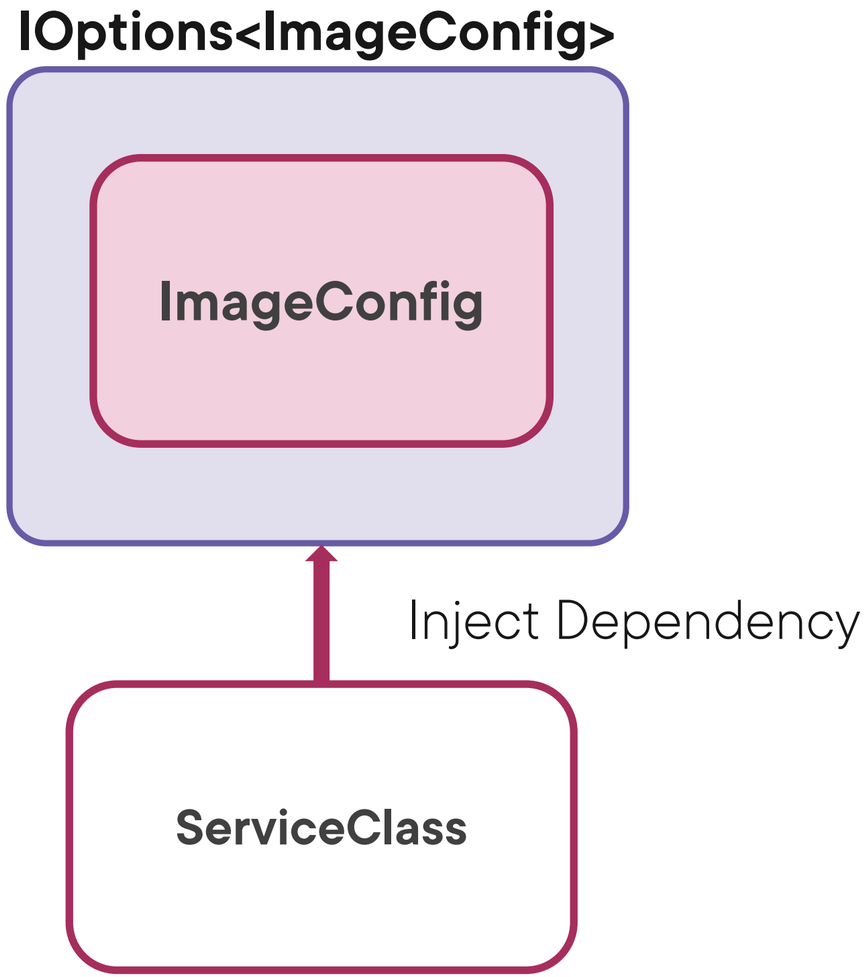


Evaluation of configuration controlled by dependent class

- Configuration as at application start?
- Evaluate configuration on each instance?

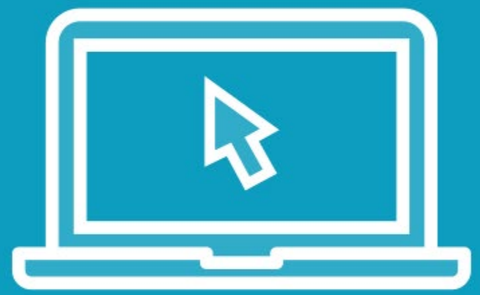
Lifetime of configuration type isolated from DI container

Options Pattern



	DI Lifetime
IOptions Configuration as at application start	Singleton
IOptionsSnapshot Evaluate configuration within scope	Scoped
IOptionsMonitor Evaluate configuration and watch for changes	Singleton

Demo



Carved Rock image processing

- **Add image configuration sections**
- **Use the Options Pattern for strongly-typed configuration**

	DI Lifetime	Named Options
IOptions Configuration as at application start	Singleton	
IOptionsSnapshot Evaluate configuration within scope	Scoped	
IOptionsMonitor Evaluate configuration and watch for changes	Singleton	

Summary

Internals of CreateDefaultBuilder

- Default configuration providers
- Extend or add your own providers

Options Pattern

- Strongly-typed configuration
- Separates lifetime of config from Dependency Injection lifetime

Apply default values

- Using the OptionsBuilder API

Respond to config changes at runtime

- Using IOptionsMonitor

Up Next:
Creating a Configurable Library