C# Design Patterns: Facade

INTRODUCTION OVERVIEW

David Starr

@elegantcoder | elegantcode.com

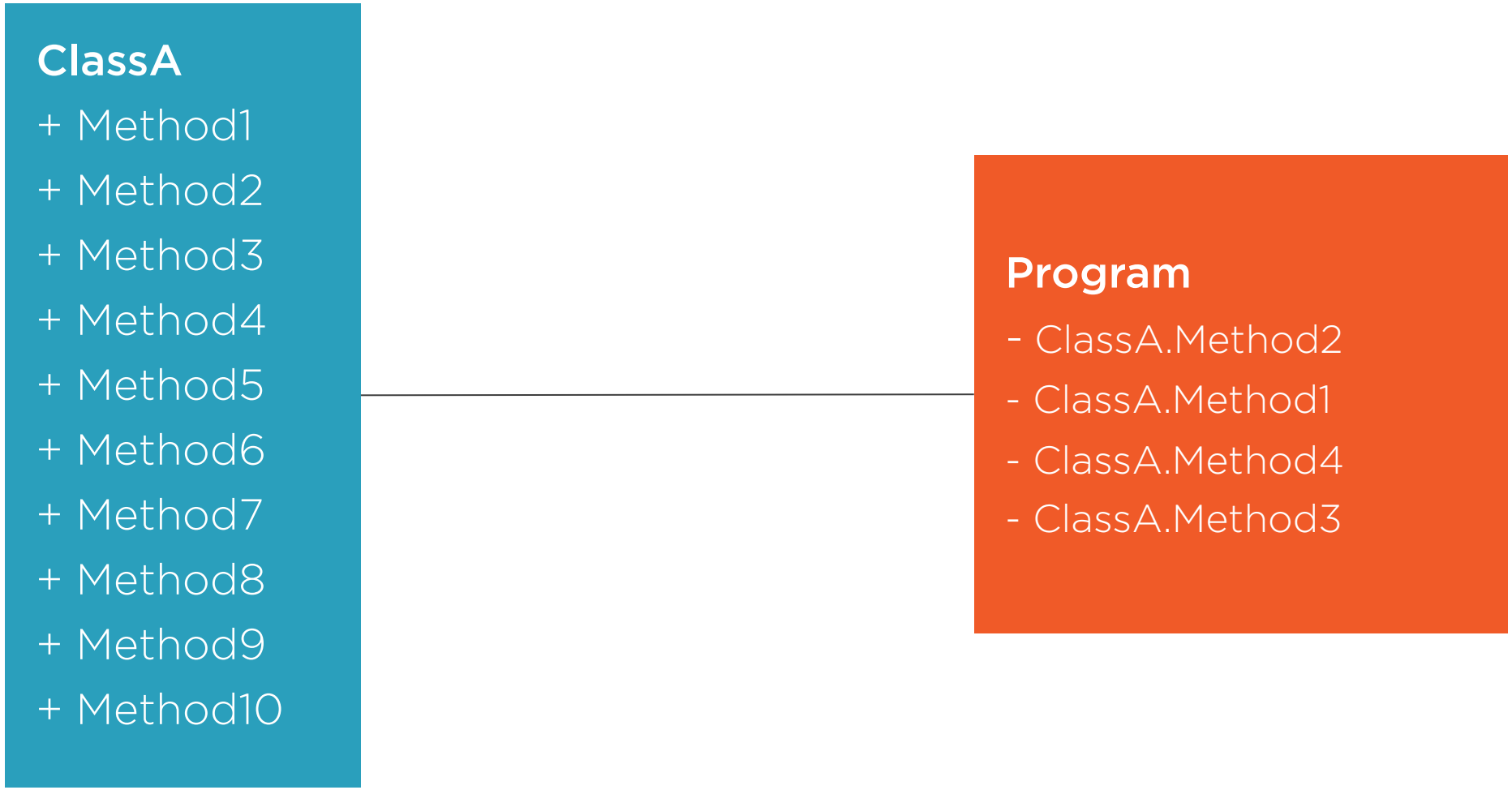
Overview

The problems to solve

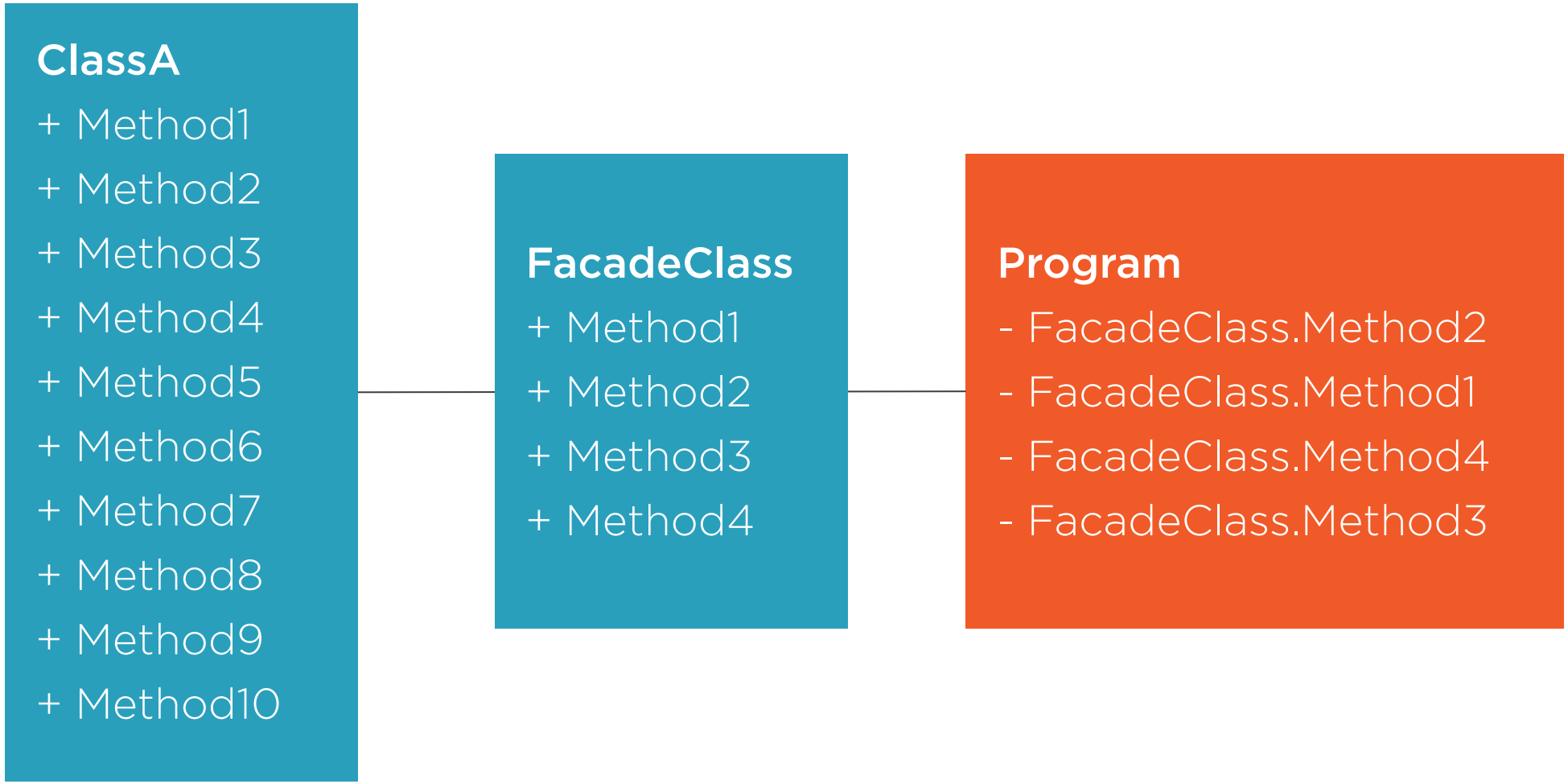
Introducing the Façade pattern

Using Façades in practice

The Problem to Solve



Applying Façade



Program.cs

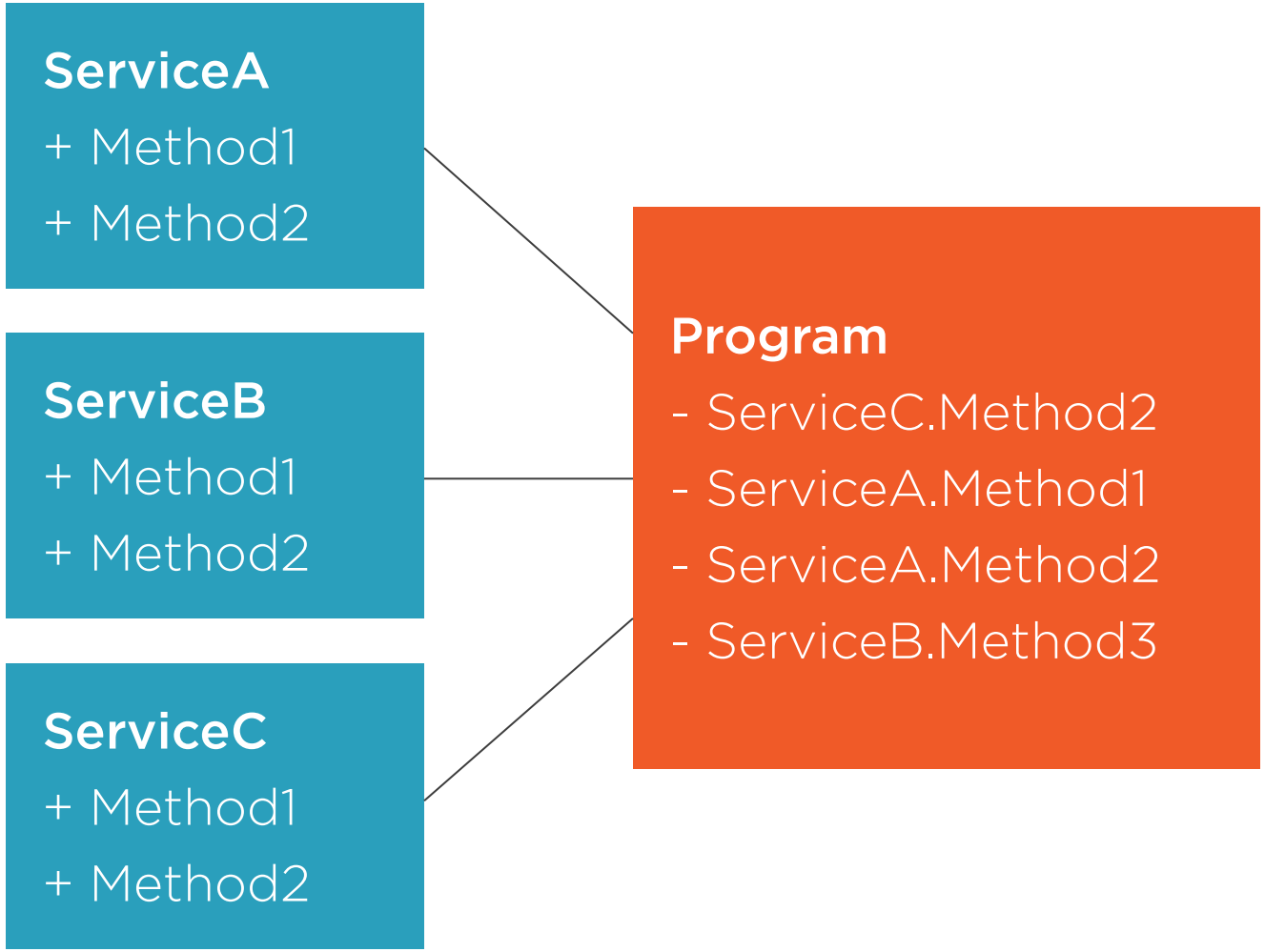
```
static void Main(string[] args)
{
    var servicesFacade = new ServicesFacade();
    servicesFacade.DoSomethingExplicitA();
    servicesFacade.DoSomethingExplicitB();
}
```

Demo

Making it tangible

See working code

The Problem to Solve



Program.cs

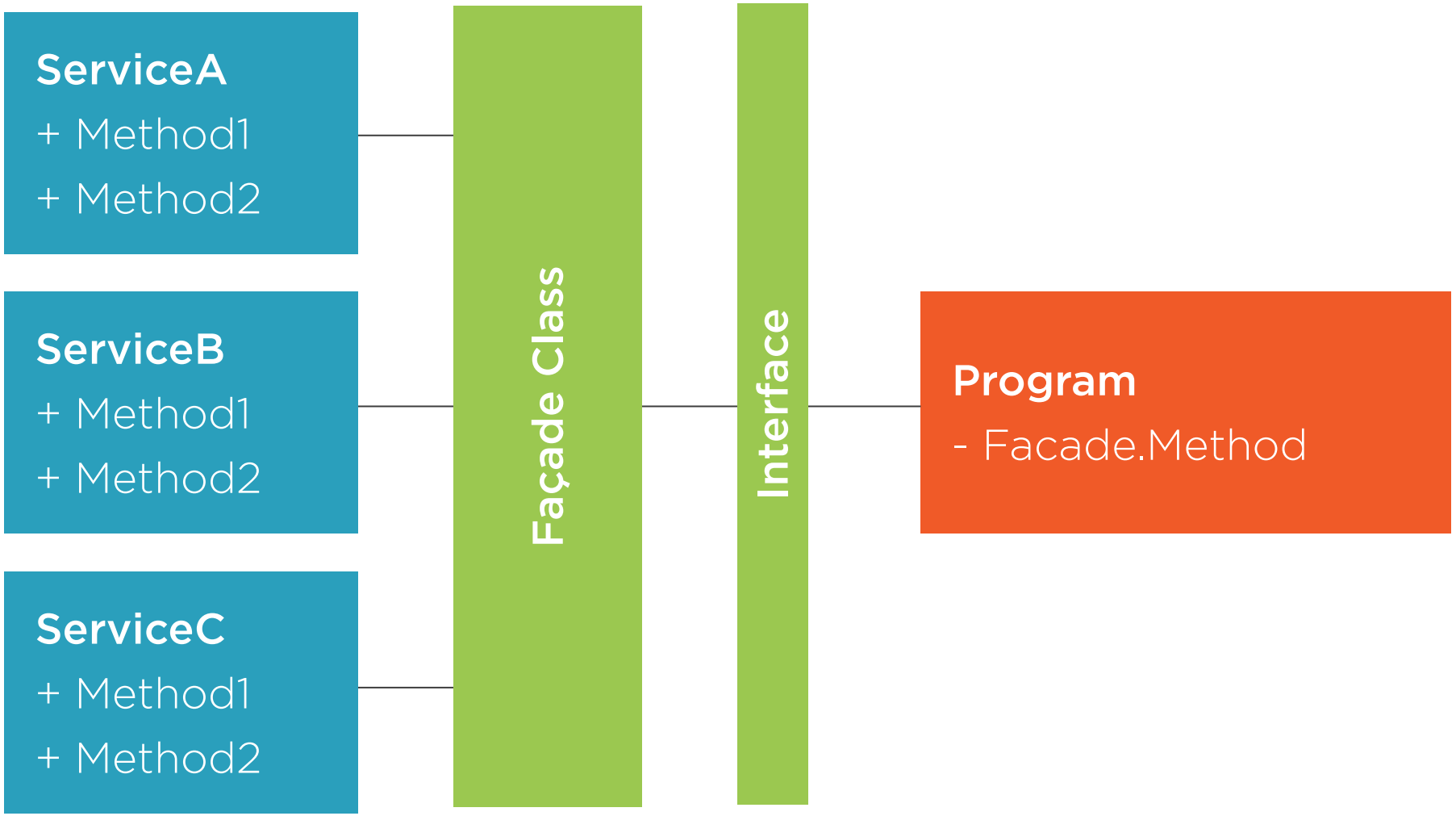
```
static void Main(string[] args)
{
    var serviceA = new ServiceA();
    serviceA.Method1();

    var serviceB = new ServiceB();
    string serviceBString = serviceB.Method2();

    var serviceC = new ServiceC();
    string serviceCString = serviceC.Method2();

    Console.WriteLine($"{serviceBString} - {serviceCString}");
}
```

The Façade Pattern



Demo

Making it tangible

See it in real code

Summary

Use Façade to provide a single interface to multiple worker classes

Using a Façade interface is a best practice

Can be used to mask giant classes with lots of methods