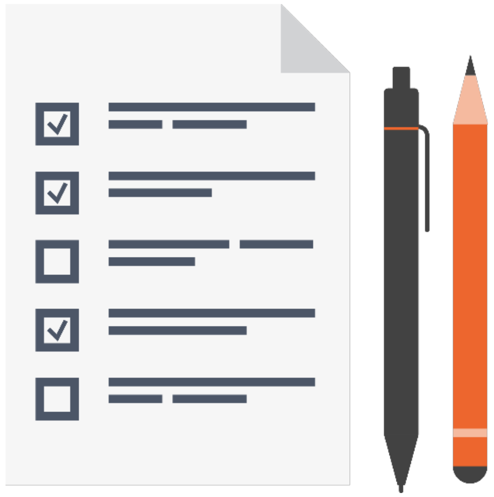
Hosting Game State in a Windows Service

Jason Roberts

@robertsjason | dontcodetired.com

Overview



Overview of "lightweight" actor systems

Separate process for game state

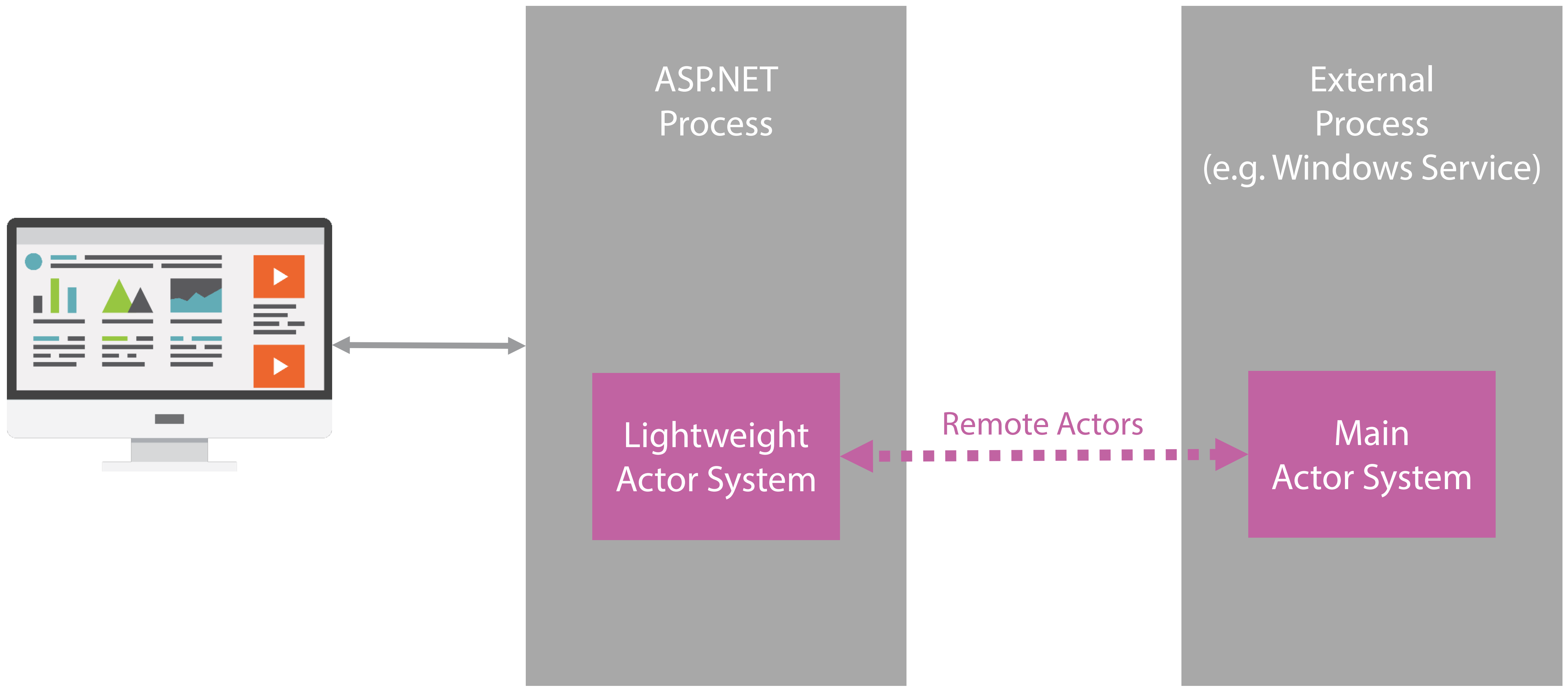
New console application to hold state

Modify GameActorSystem

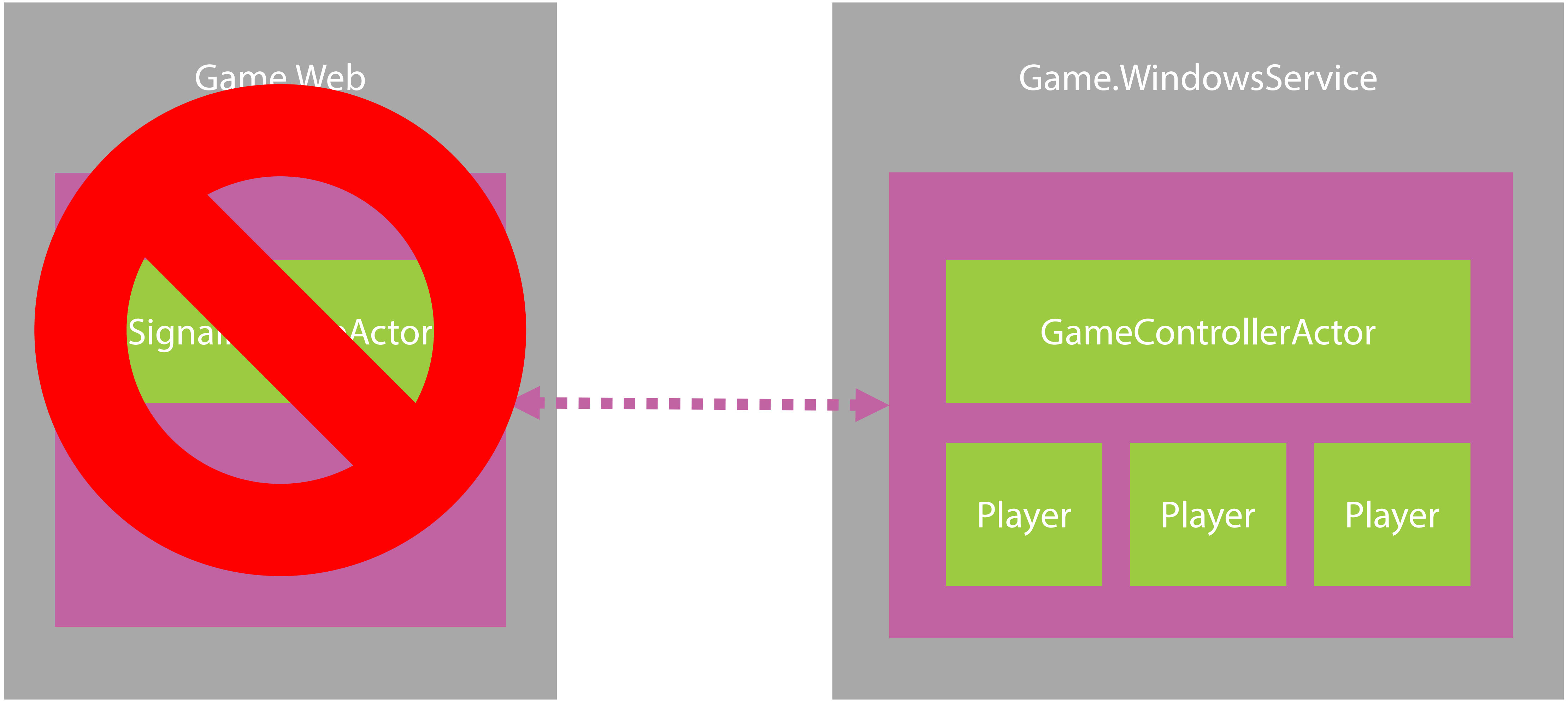
No loss of game state

Creating the Windows Service

Overview of Using Lightweight Actor Systems



Overview of Using Lightweight Actor Systems



“Building Concurrent Applications with the Actor Model in Akka.NET”

Adding a New Console Application to Hold Game State

New Game.State console application

Install Akka and Akka.Remote NuGets

Create actor system

Create GameControllerActor

Configure Akka.NET remoting in app.config

Modifying ASP.NET to Use Remote Actors

Install Akka.Remote NuGet package

Modify GameActorSystem

Remote actor selection for GameController

Configure Akka.NET remoting in web.config

Creating the Windows Service

New Game.WindowsService console app

Install Akka and Akka.Remote NuGets

Install Topshelf NuGet package

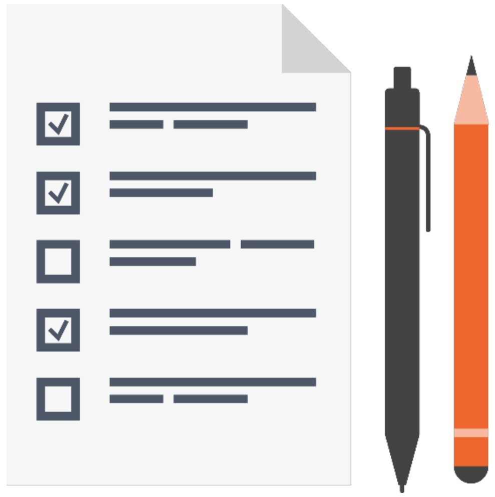
Configure Akka.NET remoting in app.config

New GameStateService

Use Topshelf to configure Windows Service

Install the Windows Service

Summary



Overview of "lightweight" actor systems

Game.State console application

Modified Game.Web

Remote actor selection for GameController

No loss of game state

Created the Game.WindowsService

Topshelf

Resources and Further Learning

- Akka.NET
  - http://getakka.net/
  - Building Concurrent Applications with the Actor Model in Akka.NET
  - Implementing Logging and Dependency Injection in Akka.NET
  - Building Reactive Concurrent WPF Applications with Akka.NET
  - Improving Message Throughput in Akka.NET
- SignalR
  - http://www.asp.net/signalr
  - SignalR Across Web and Devices
- Knockout
  - http://knockoutjs.com/
  - Essential Knockout and JavaScript Tips