WPF

Microsoft MVP

Connor Park

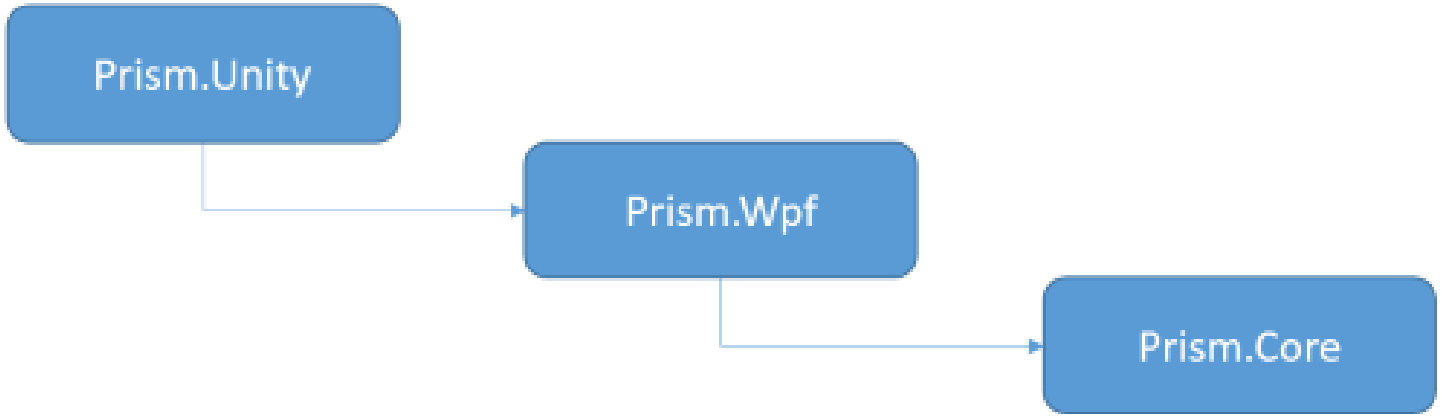
Day 4

Prism Framework

https://github.com/PrismLibrary/Prism-Samples-Wpf.git

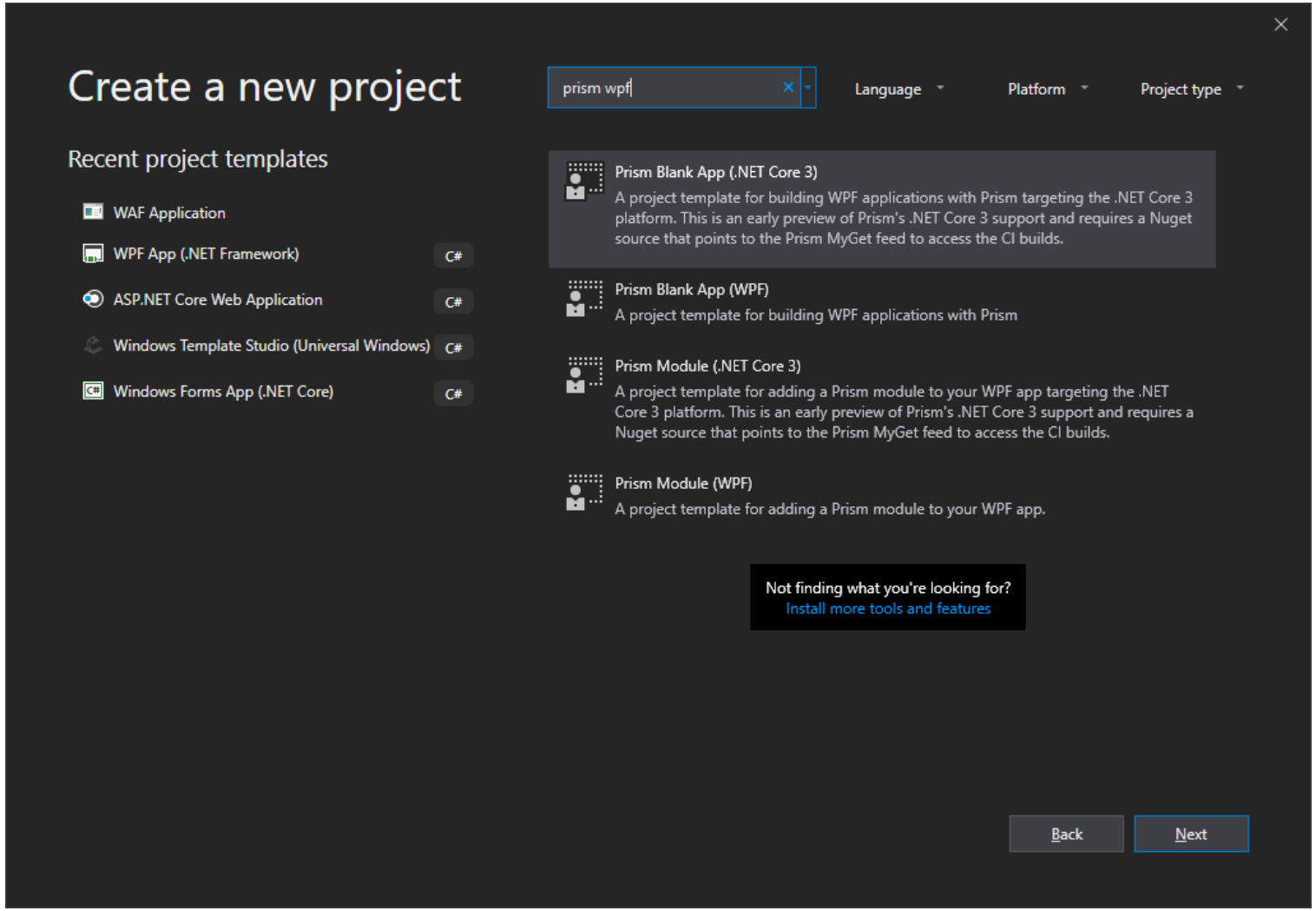
Prism overview

- Loosely coupled, maintainable, and testable XAML framework
- Microsoft patterns & practices의 오픈 소스 버전
- MVVM Pattern
- Dependency Injection
- Commands
- EventAggregator
- Module
- Unit Test support
- Prism.Core, Prism.Unity, Prism.Wpf 7.1.0.431



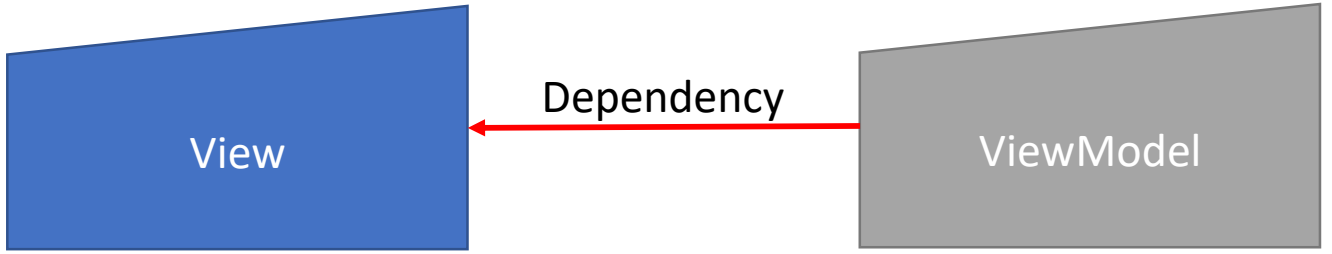
Prism Template Pack

- Extensions -> Prism -> Prism Template Pack
- Xamarin.Forms, WPF, UWP 및 기본 iOS 및 Android 응용 프로그램을 만드는데 필요한 생산성 높은 개발자 워크플로를 사용
- https://github.com/PrismLibrary/Prism-Samples-Wpf

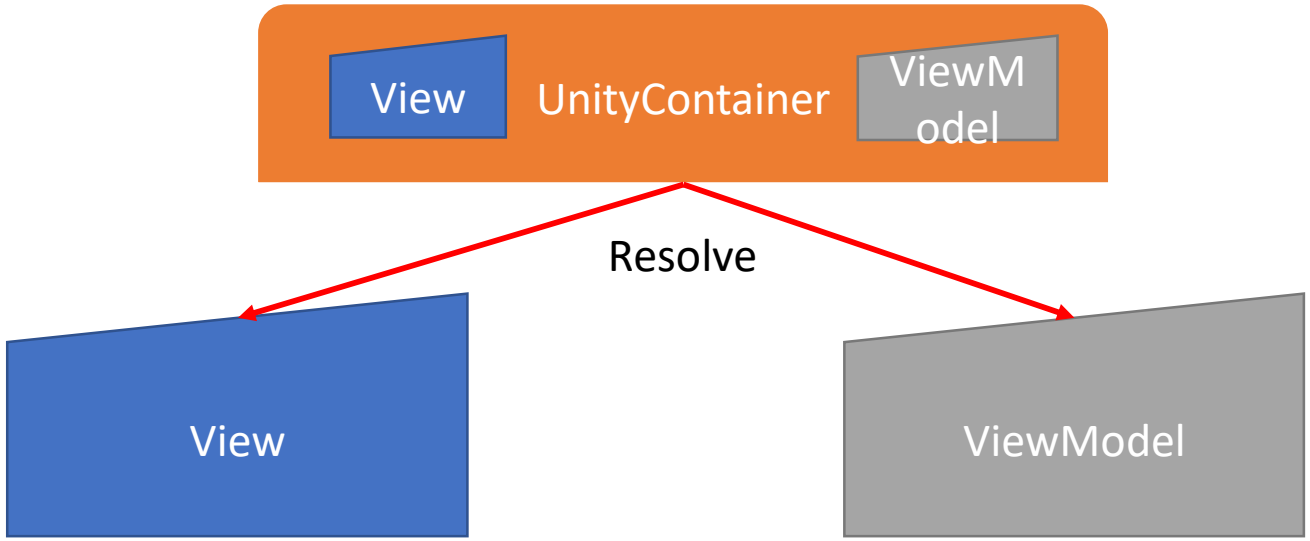


UnityContainer

- Dependency Injection Container
- 중요한 오브젝트를 컨테이너에서 관리를 해서 의존성을 없애는 목적
- Register()
- RegisterInstance()
- RegisterSingleton()
- Container.Resolve()



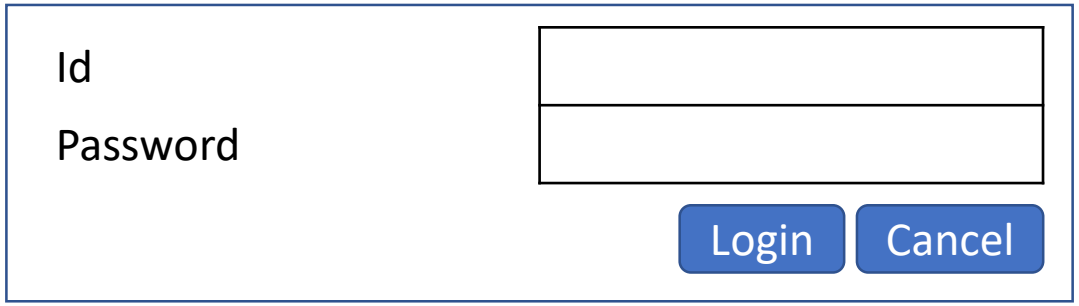
var viewModel = new ViewModel();



prism:ViewModelLocator.AutoWireViewModel="True"

Commanding

- 기본적으로 ICommand를 사용
- CanExecute, RaiseCanExecuteChanged()
- DelegateCommand
  - ObservesProperty
  - ObservesCanExecute
- Task-Based DelegateCommand
  - async void
  - New D..(async () => await …)
- TriggerParameterPath



Login 버튼을 클릭하려면 Id와 Password가 입력이 되어야 함

LoginCommand = new DelegateCommand(OnLoginCommand, CanLoginCommand);

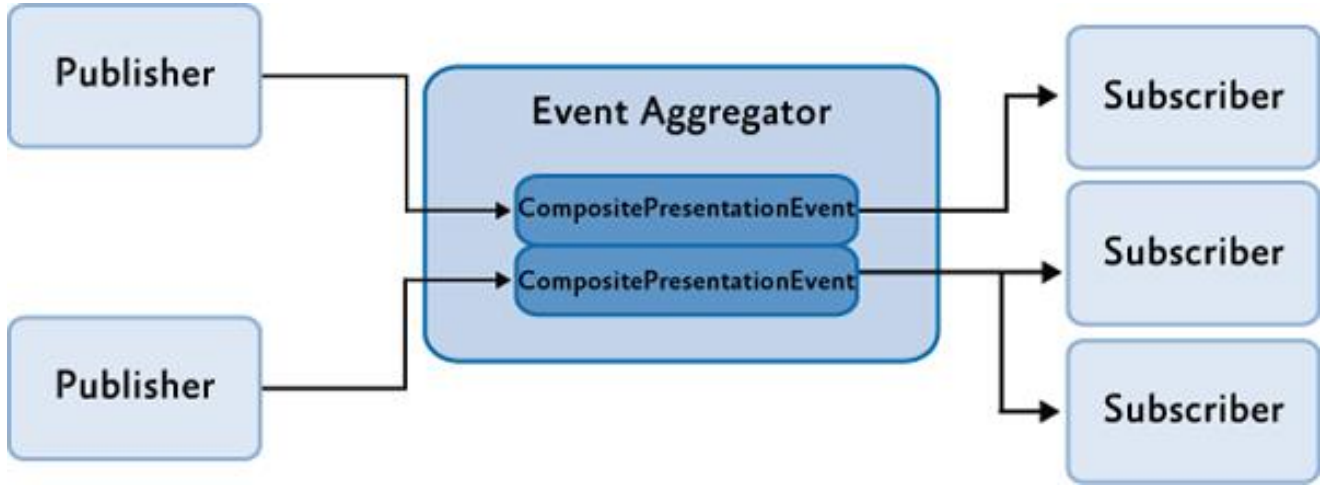
Id, Password가 입력될 때마다 RaiseCanExecuteChanged()를 실행

->

Id, Password 프로퍼티가 변경될 때 마다 자동으로 CanLoginCommand 실행

EventAggregator

- Loosely coupled로 event를 전달할 수 있음
- Publisher and subscriber 서로에 대한 직접적인 참조가 없음
- Type event
- Filtering
- keepSubscriberReferenceAlive



ViewModelLocator

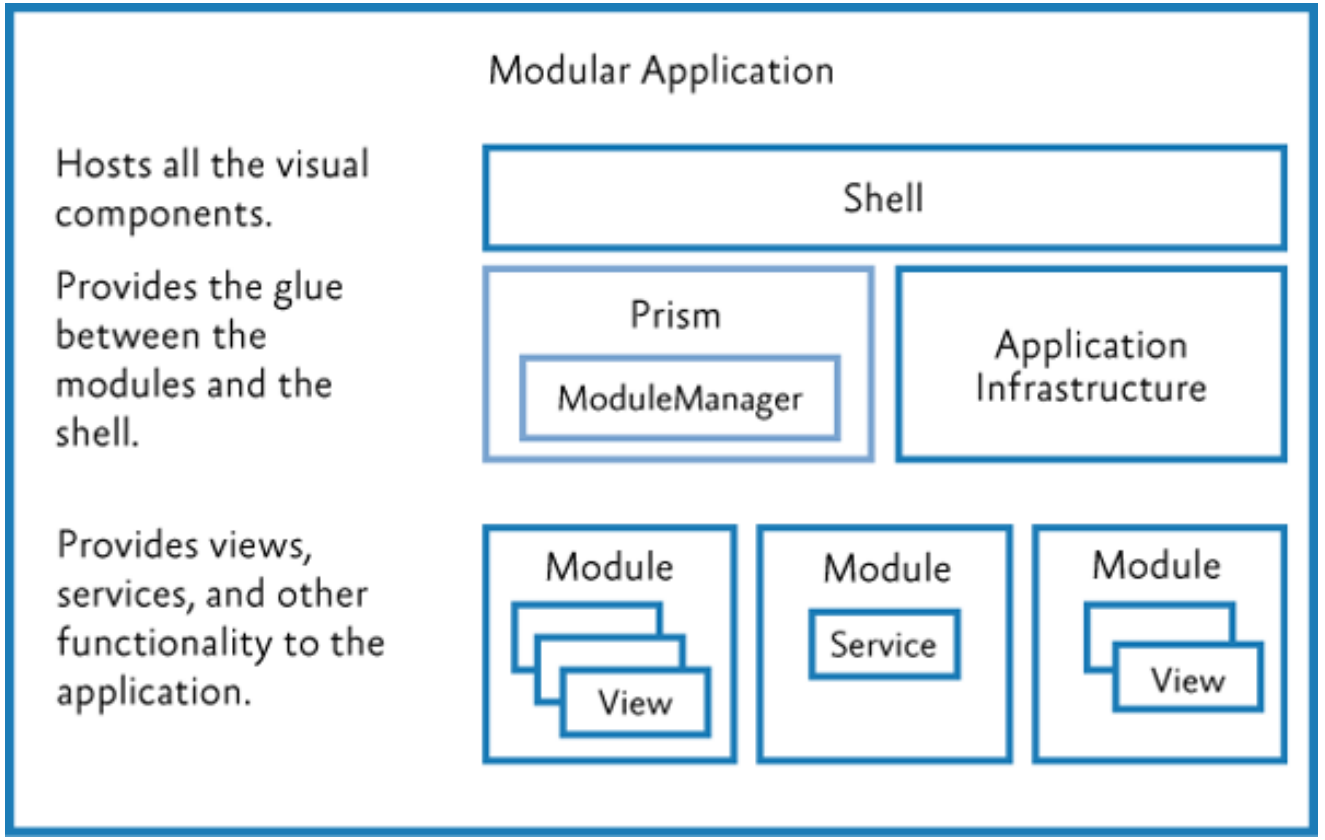
- MvvmLight ViewModelLocator와 다름
- 표준 명명 규칙을 사용하여 View와 ViewModel을 연결
- prism:ViewModelLocator.AutoWireViewModel="True“
- Custom ViewModel Registrations

```
protected override void ConfigureViewModelLocator()
        {
            base.ConfigureViewModelLocator();

ViewModelLocationProvider.SetDefaultViewTypeToViewModelTypeResolver((viewType) =>
            {
                var viewName = viewType.FullName;
                if (viewName == null) return null;
                if (viewName.EndsWith("Page")) viewName = viewName.Substring(0, viewName.Length - 4);
                if (viewName.EndsWith("Control")) viewName = viewName.Substring(0, viewName.Length - 7);
                viewName = viewName.Replace(".Views.", ".ViewModels.");
                viewName = viewName.Replace(".Controls.", ".ControlViewModels.");
                var viewAssemblyName = viewType.GetTypeInfo().Assembly.FullName;
                var viewModelName = $"{viewName}ViewModel, {viewAssemblyName}";
                return Type.GetType(viewModelName);
            });
        }
```

Modules

- 대규모 응용 프로그램 개발시 기능 단위(Module)로 집합을 나누어 개발
- Module과 Module은 서로 Dependency를 가지지 않음
- 개발, 테스트, 배포 및 유지 관리가 용이
- AppConfig
- Code
- Directory
- LoadManual




```
 var regionManager =
containerProvider.Resolve<IRegionManager>();

regionManager.RegisterViewWithRegion("ContentRegion",
typeof(ViewA));
```

App.config

- App.config 파일을 이용
  - 모듈 연결을 App.config 파일을 수정하는 것 만으로 변경 가능
- 모듈 Reference 없음
  - 완벽한 모듈간 분리
- 각 모듈의 Build Events에 xcopy 명령 필요
  - xcopy "$(TargetDir)*.*" "$(SolutionDir)\$(SolutionName)\$(OutDir)" /Y
- 빌드할 때 주의 필요

```
protected override IModuleCatalog CreateModuleCatalog()
{
        return new ConfigurationModuleCatalog();
}
```

Code

- 가장 기본적인 모듈 추가 방법
- 모듈 Reference 추가 필요
- ConfigureModuleCatalog 이용

```
protected override void ConfigureModuleCatalog(IModuleCatalog moduleCatalog)
{
        moduleCatalog.AddModule<ModuleCode.ModuleCodeModule>(InitializationMode.WhenAvailable);
}
```

Directory

- 디렉토리리를 지정하는 방법
  - 내부에 모듈들을 모두 추가
  - 모듈의 갯수가 만은 경우 사용
- 모듈 Reference 없음
- 각 모듈의 Build Events에 xcopy 명령 필요
  - xcopy "$(TargetDir)$(TargetName)*$(TargetExt)" "$(SolutionDir)$(SolutionName)\$(OutDir)Modules\" /Y /S
- 빌드시 주의 및 각 모듈별로 InitializationMode를 지정할 수 없음

```
protected override IModuleCatalog CreateModuleCatalog()
{
        return new DirectoryModuleCatalog() { ModulePath = @".\Modules" };
}
```

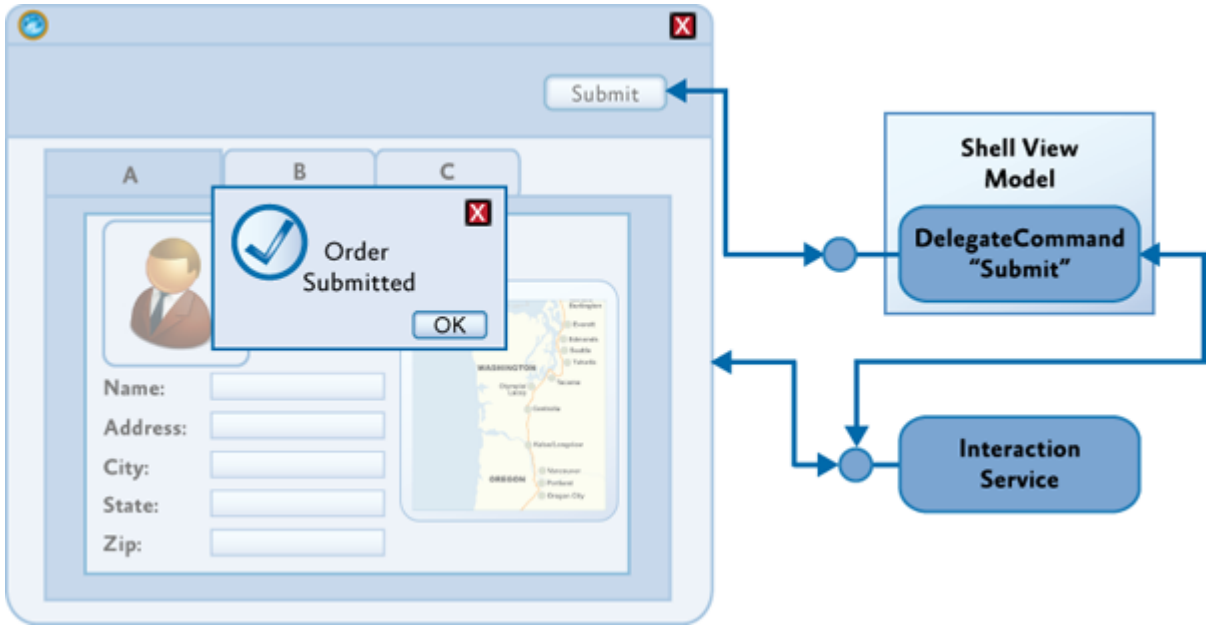
LoadManual

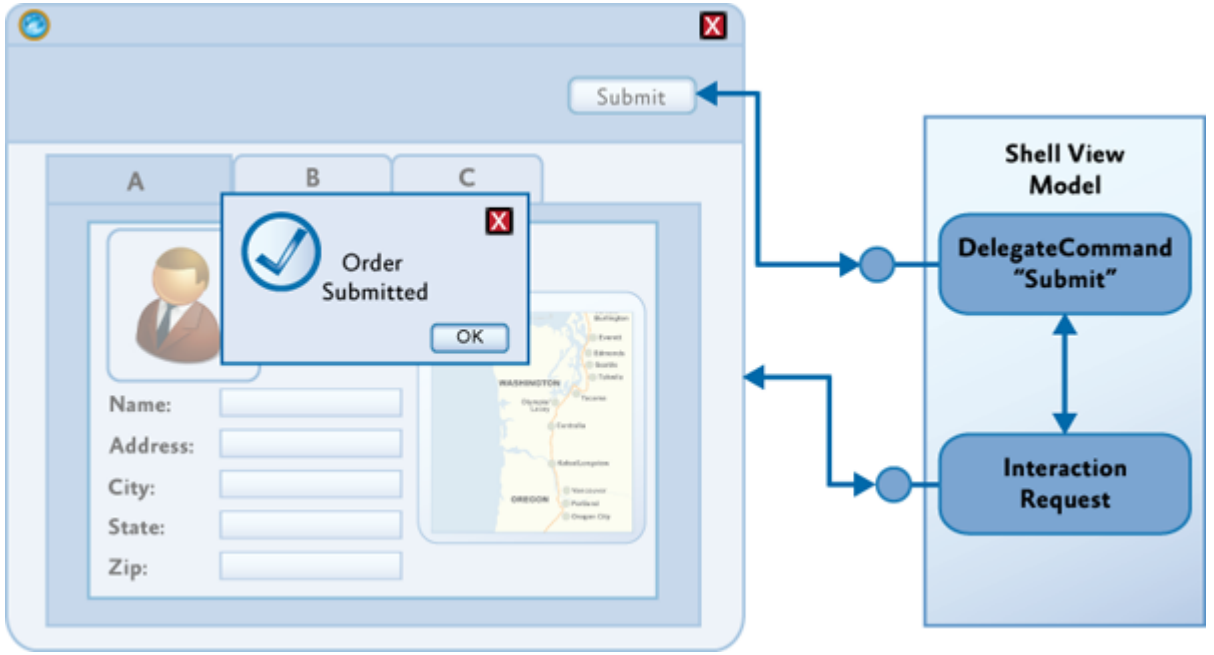
- 모듈 Reference 추가 필요
- ConfigureModuleCatalog 이용

```
protected override void ConfigureModuleCatalog(IModuleCatalog moduleCatalog)
        {
            var moduleAType = typeof(ModuleAModule);
            moduleCatalog.AddModule(new ModuleInfo()
            {
                ModuleName = moduleAType.Name,
                ModuleType = moduleAType.AssemblyQualifiedName,
                InitializationMode = InitializationMode.OnDemand
            });
        }
```

User Interaction

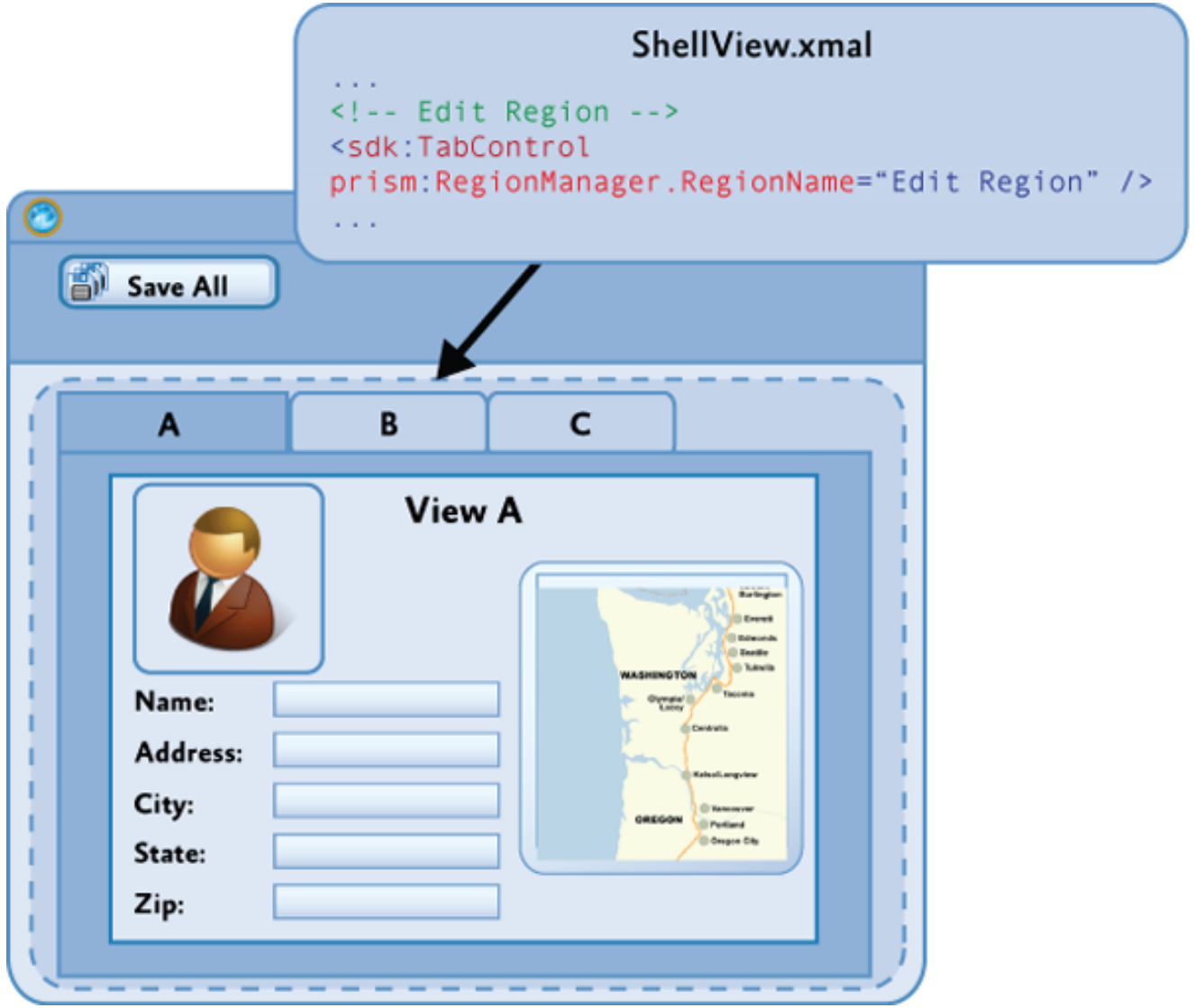
- MessageBox
- IInteractionRequest
  - INotification
  - IConfirmation





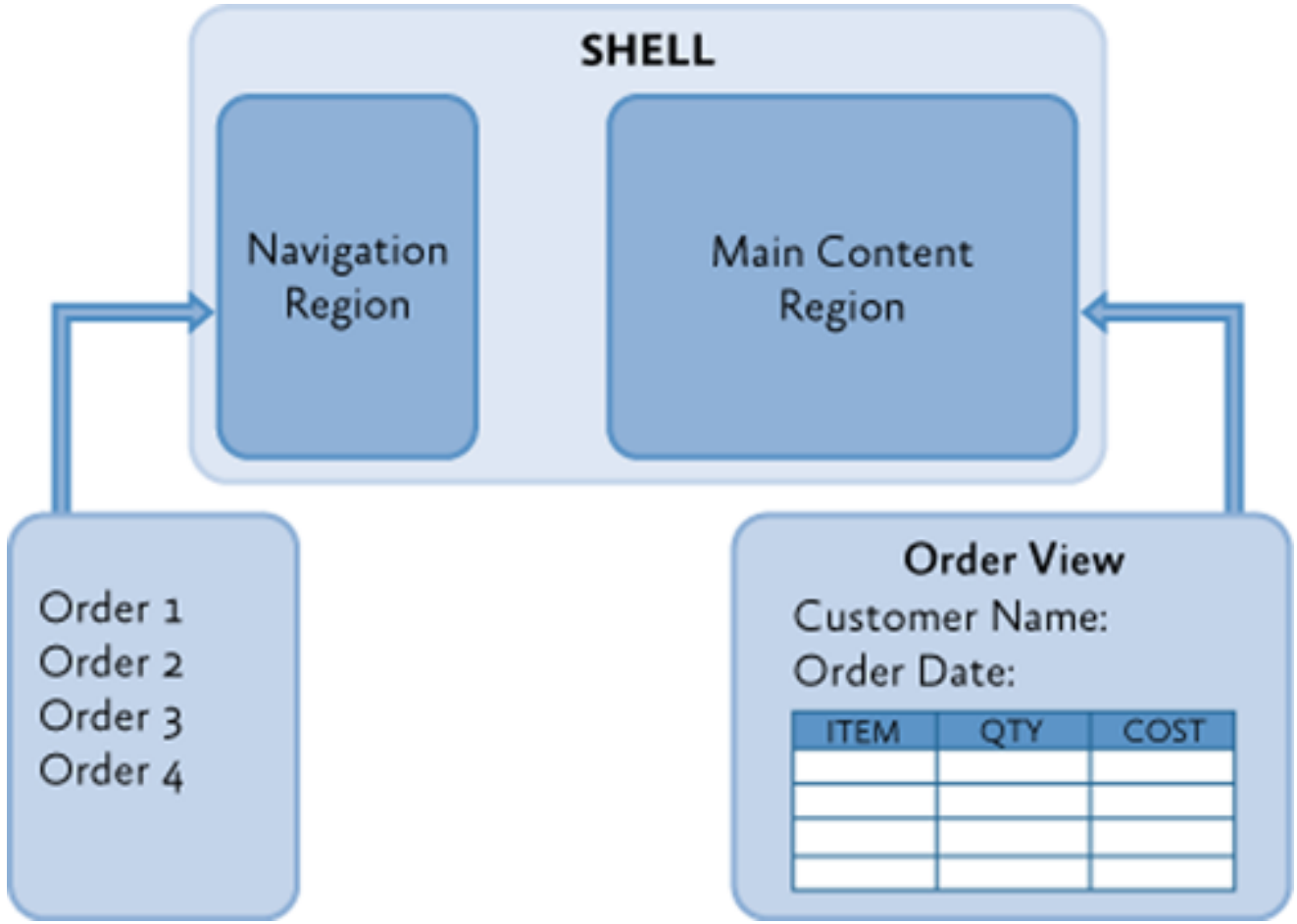
Regions

- Child views within the application’s UI
- Logical placeholder
- IActiveAware, IsActive
- Navigate 지원



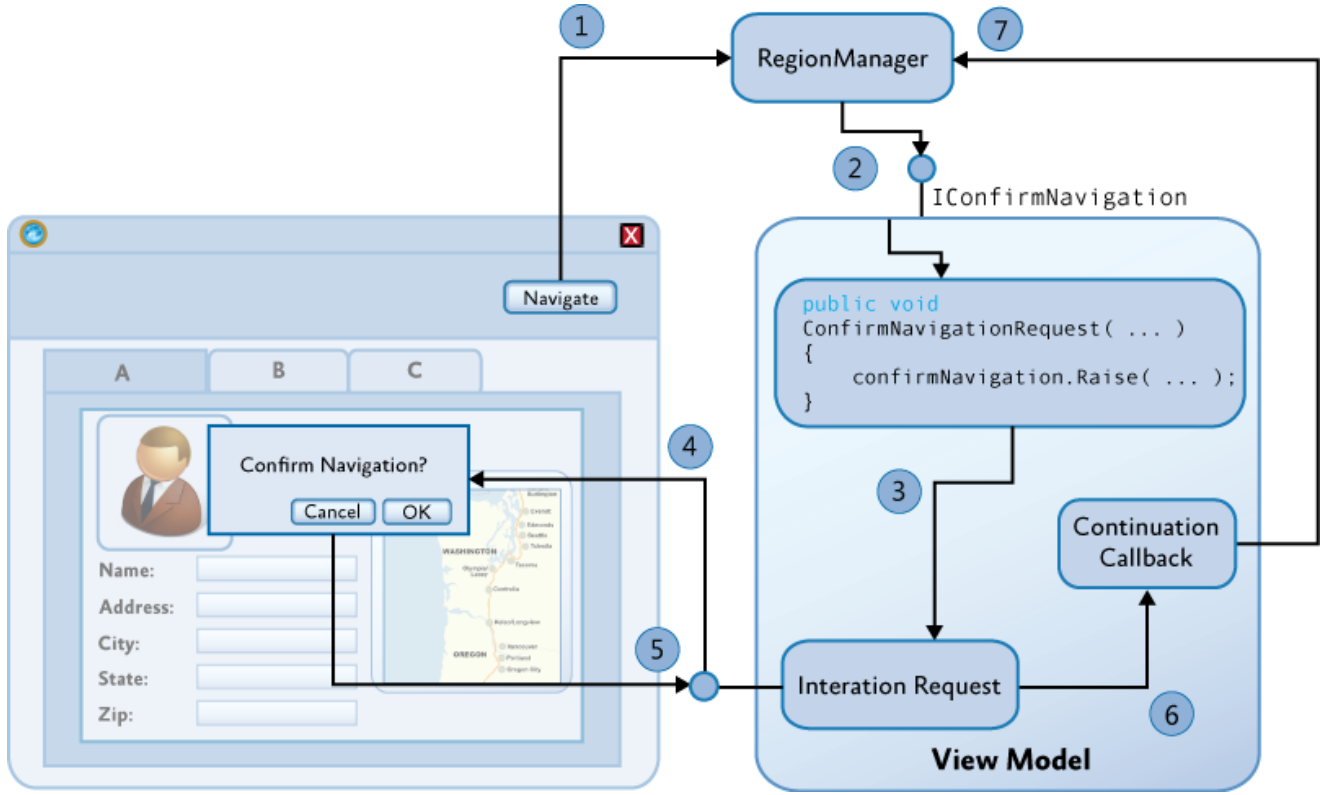
UI Layout Concepts

- Shell
- Regions



Navigation

- Page navigation
- Region navigation



Day 4 정리