

Caliburn.Micro

Martin Bühlmann
Bruno Juchli

Über uns

- EROWA AG
 - Spannsysteme
 - Roboter
 - Prozessleitsysteme
 - 240 Mitarbeiter CH, 450 weltweit
- Prozessleitsystem EMC3
 - Automatisierung der Fertigung
 - Optimale Ressourcennutzung
 - Verantwortung dem Anwender abnehmen
 - C#.Net, Ninject, NHibernate, WCF, xUnit, Mspec, VS, R#, dotCover, Teamcity,...
 - UI: WPF/XAML, Caliburn.Micro, Ribbons

www.erowa.com

MVVM-Pattern

View

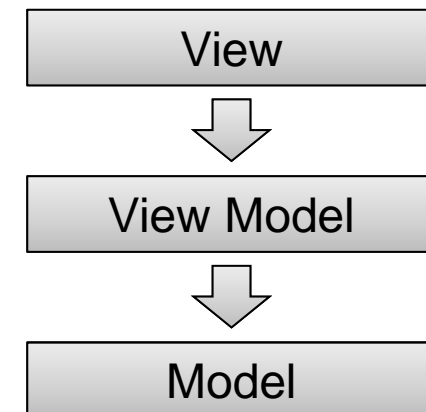
- | Visualisierung der Anwendung (GUI)
- | Verknüpfung der Controls zu den Daten mittels Data Binding

View Model

- | Beinhaltet die UI-Logik
- | Bindeglied zwischen View und Model
- | Mittels Unit-Tests überprüfbar

Model

- | Datenzugriffsschicht
- | Mittels Unit-Tests überprüfbar



CALIBURN.MICRO

WPF, Silverlight, WinRT
and WP7 made easy

| Caliburn.Micro...

- | ...ist ein MVVM Framework ursprünglich entwickelt von Rob Eisenberg
- | ...entstand als Neuentwicklung von Caliburn
- | ...unterstützt die Entwicklung für WPF, Silverlight, Windows Phone und Windows RT
- | ...bietet ein grosses Feature Set
- | ...ist dokumentiert unter <http://caliburnmicro.codeplex.com>
- | ...verwendet die MIT License (Verwendung in kommerziellen Applikationen erlaubt)



Caliburn.Micro - Features

- | Actions
- | Bootstrapper
- | Conventions
- | Screen / Conductors
- | Coroutines
- | Logging
- | PropertyChangedBase und BindableCollection

Caliburn.Micro - Actions

Actions...

- | ...sind eine alternative zu Commands
- | ...verwenden Conventions für den Aufruf des Handlers im View Model
- | ...verwenden automatisch eine Can-Methode als Ausführungs-Guard (falls vorhanden)
- | ...können Parameter mitgegeben werden.

```
<UserControl x:Class="Caliburn.Micro.Hello.ShellView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <StackPanel>
        <TextBox x:Name="Name" />
        <Button x:Name="SayHello"
            Content="Click Me" />
    </StackPanel>
</UserControl>
```

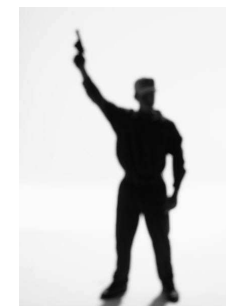
```
public bool CanSayHello
{
    get { return !string.IsNullOrEmpty(Name); }
}

public void SayHello()
{
    MessageBox.Show(string.Format("Hello {0}!", Name));
}
```

Caliburn.Micro - Bootstrapper

- | Bootstrapper...
 - | ...dient als Start-Mechanismus für die jeweilige Anwendung
 - | ...konfiguriert Caliburn.Micro anwendungsspezifisch
 - | ...definiert die Haupt-View
 - | ...dient als Extension Point um z.B. den IoC Container einzuspeisen

- | Wird bei EROWA nicht verwendet, da wir den Bootstrapper von Appccelerate verwenden.



Caliburn.Micro - Conventions

| Conventions...

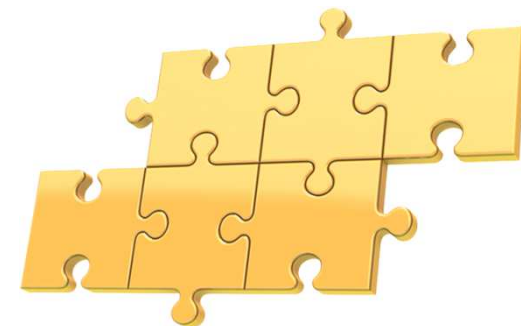
| ...werden für das Auffinden von View Model/View verwendet

OrderViewModel → *OrderView*

| ...werden von den Actions zum Finden der Handler/Guards verwendet

SaveOrder

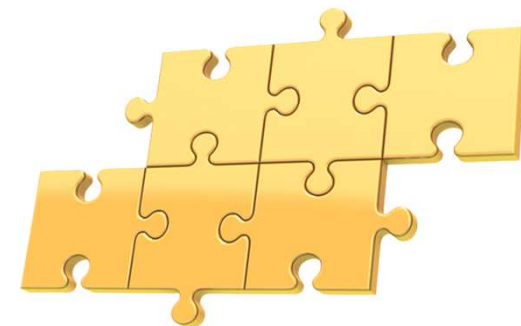
CanSaveOrder



Caliburn.Micro – Screen / Conductors

| Screen / Conductors...

- | ...sind Basisklassen für einfache Views (Screen) oder zusammengesetzte Views (Conductor)
- | ...liefern Standardimplementierungen für verschiedene Caliburn.Micro spezifische Interfaces
- | ...sollen die Implementation von Views vereinfachen, die Verwendung ist aber nicht zwingend
- | ...erschweren die Testbarkeit, da Funktionalität geerbt statt per Komposition wiederverwendet wird



Caliburn.Micro - Coroutines

| Coroutines...

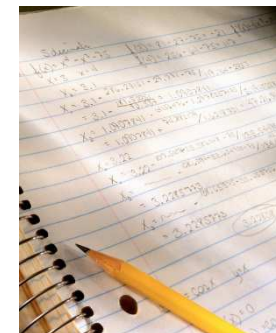
- | ...sind Methoden, die ihren Ablauf unterbrechen und später wieder aufnehmen können
- | ...können von Actions als Handler verwendet werden
- | ...helfen, Teile von Handlern asynchron oder synchron zum UI Thread auszuführen

```
using System.Collections.Generic;
using System.ComponentModel.Composition;

[Export(typeof(ScreenOneViewModel))]
public class ScreenOneViewModel
{
    public IEnumerable<IResult> GoForward()
    {
        yield return Loader.Show("Downloading...");
        yield return new LoadCatalog("Caliburn.Micro.Coroutines.External.xap");
        yield return Loader.Hide();
        yield return new ShowScreen("ExternalScreen");
    }
}
```

Caliburn.Micro - Logging

- | Logging...
 - | ...wird durch Caliburn.Micro angeboten
 - | ...kann einen beliebigen Logger verwenden (z.B. Log4Net)



Caliburn.Micro – Helper Classes

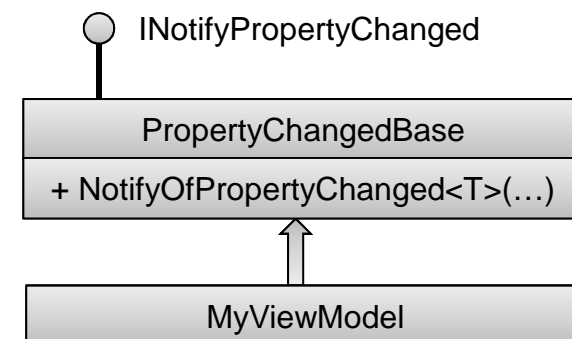
| PropertyChangedBase...

- | ...bietet eine Default-Implementation für INotifyPropertyChanged
- | ...feuert das PropertyChanged-Event, indem das Property mittels Expression angegeben wird

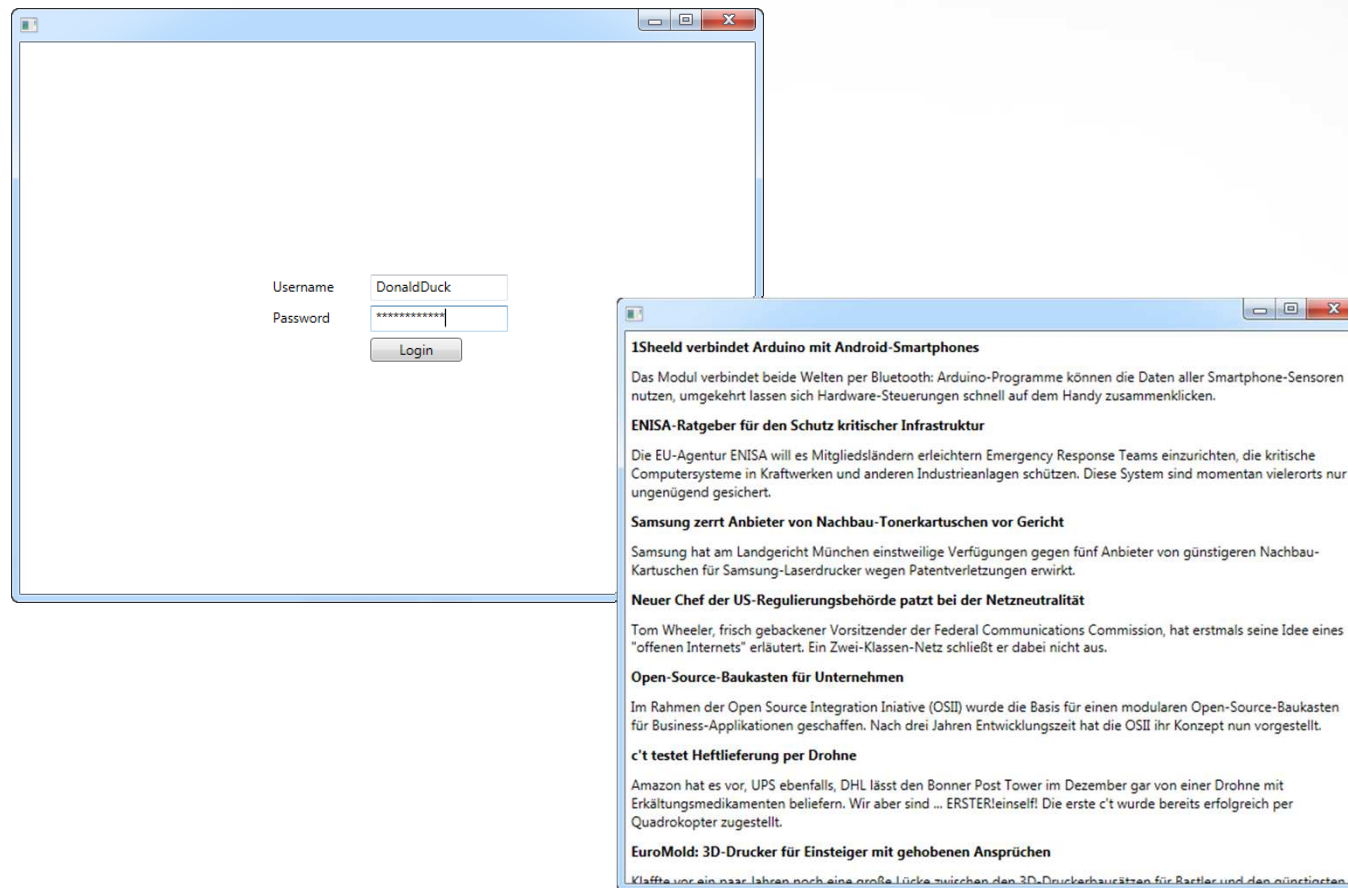
```
this.NotifyOfPropertyChanged(() => this.Name);
```

| BindableCollection...

- | ...ist eine ObservableCollection mit Interface (IObservableCollection, Testability)
- | ...stellt sicher, dass die Events der Collection im UI Thread gefeuert werden



Caliburn.Micro – Simples Beispiel



Caliburn.Micro – Simples Beispiel

```
SampleBootstrapper.cs  X
CaliburnMicroSample.SampleBootstrapper
namespace CaliburnMicroSample
{
    using Caliburn.Micro;
    using CaliburnMicroSample.Shell;

    public class SampleBootstrapper : Bootstrapper<ShellViewModel>
    {
    }
}
```

Caliburn.Micro – Simples Beispiel

```
ShellViewModel.cs  SampleBootstrapper.cs
CaliburnMicroSample.Shell.ShellViewModel  ShellViewMod

namespace CaliburnMicroSample.Shell
{
    using CaliburnMicroSample.Login;
    using CaliburnMicroSample.Navigation;
    using PropertyChanged;

    [ImplementPropertyChanged]
    public class ShellViewModel : INavigationTarget
    {
        public ShellViewModel()
        {
            Navigator.Initialize(this);
        }

        public object PageContent { get; set; }

        public void Loaded()
        {
            this.NavigateTo(new LoginViewModel());
        }

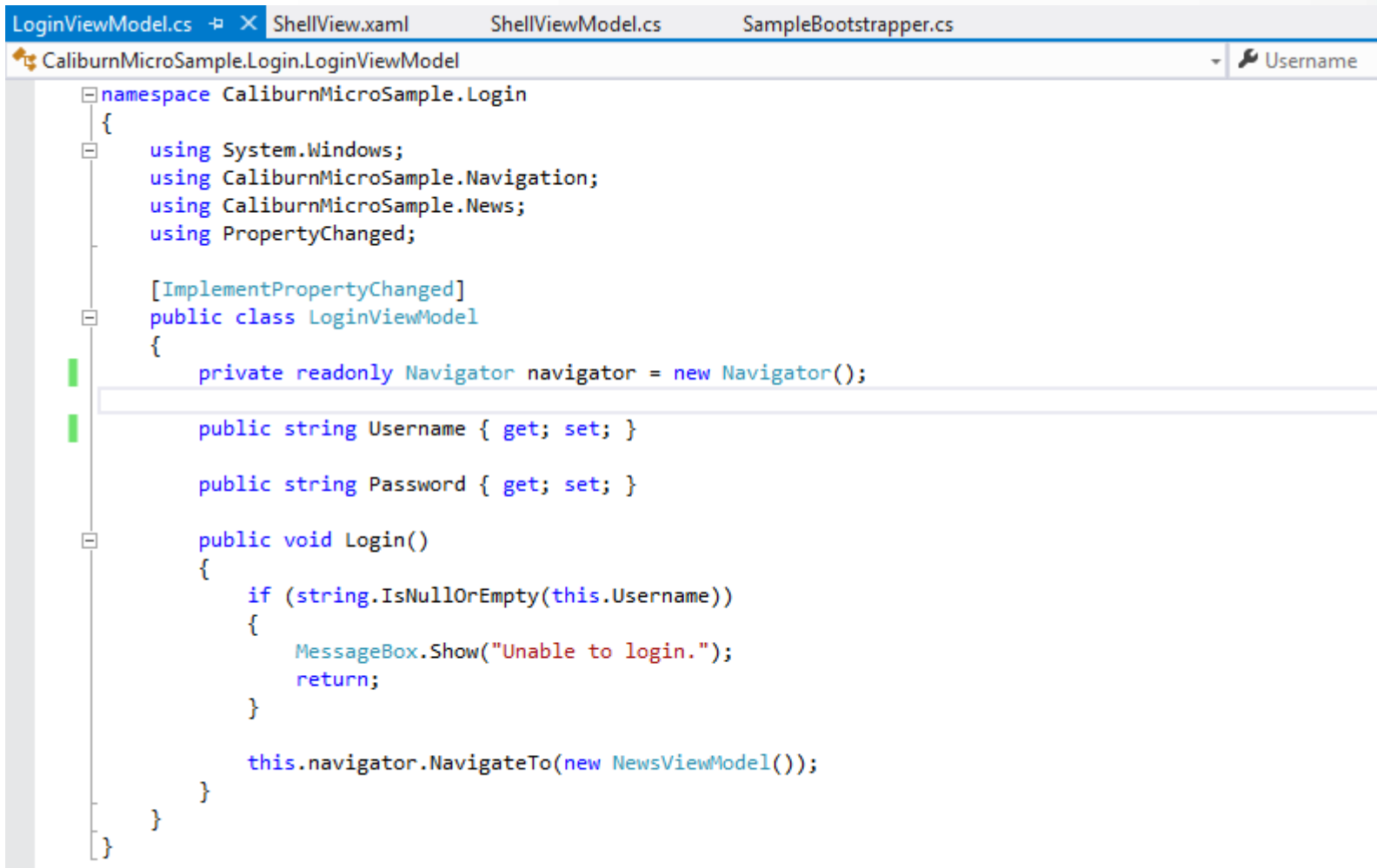
        public void NavigateTo(object viewModel)
        {
            this.PageContent = viewModel;
        }
    }
}
```

Caliburn.Micro – Simples Beispiel

```
ShellView.xaml  ShellViewModel.cs  SampleBootstrapper.cs
<UserControl x:Class="CaliburnMicroSample.Shell.ShellView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:cal="http://www.caliburnproject.org"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    Width="640"
    Height="480"
    d:DesignHeight="480"
    d:DesignWidth="640"
    mc:Ignorable="d">
    <i:Interaction.Triggers>
        <i:EventTrigger EventName="Loaded">
            <cal:ActionMessage MethodName="Loaded" />
        </i:EventTrigger>
    </i:Interaction.Triggers>

    <ContentControl x:Name="PageContent" />
</UserControl>
```


Caliburn.Micro – Simples Beispiel



```
LoginViewModel.cs  ShellView.xaml  ShellViewModel.cs  SampleBootstrapper.cs
CaliburnMicroSample.Login.LoginViewModel  Username

namespace CaliburnMicroSample.Login
{
    using System.Windows;
    using CaliburnMicroSample.Navigation;
    using CaliburnMicroSample.News;
    using PropertyChanged;

    [ImplementPropertyChanged]
    public class LoginViewModel
    {
        private readonly Navigator navigator = new Navigator();

        public string Username { get; set; }

        public string Password { get; set; }

        public void Login()
        {
            if (string.IsNullOrEmpty(this.Username))
            {
                MessageBox.Show("Unable to login.");
                return;
            }

            this.navigator.NavigateTo(new NewsViewModel());
        }
    }
}
```

Caliburn.Micro – Simples Beispiel

```
<Button Grid.Row="3"
        Grid.Column="2"
        Width="80"
        Margin="10,3,0,0"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"
        cal:Message.Attach="Login"
        Content="Login" />
```

```
<Button x:Name="Login"
        Grid.Row="3"
        Grid.Column="2"
        Width="80"
        Margin="10,3,0,0"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"
        Content="Login" />
```

DEMO

Fragen?