

MVVM



What is the key feature that differentiates MVVM from other UI separation patterns like MVC and MVP?

- Data binding is the key feature that differentiates MVVM from other UI separation patterns like MVC and .

# Benefits of MVVM

- ⌞ code reuseability
- ⌞ easy maintenance
- ⌞ separation of data model
- ⌞ design and code, and clean architecture etc.

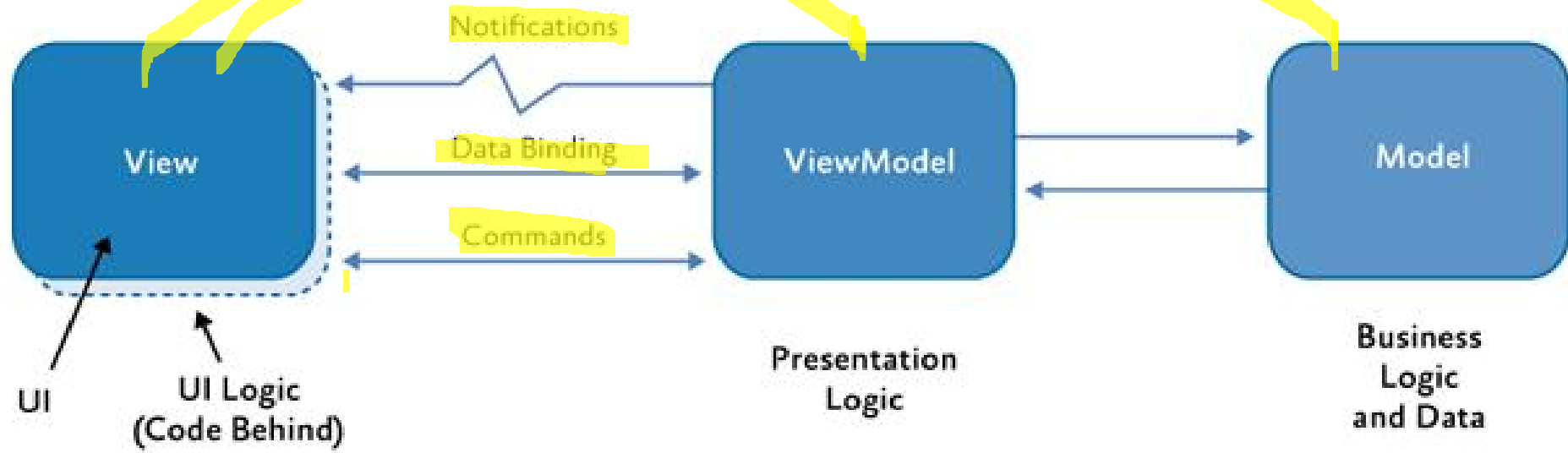
# No Separation of Concerns

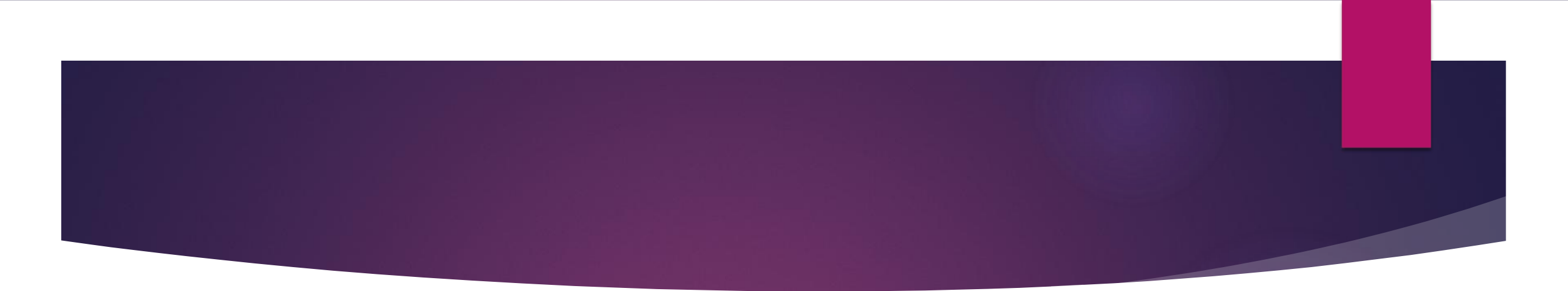
```
private void ComputeCustomerOrdersTotal(object sender, RoutedEventArgs e)
{
    var selectedCustomer = this.customerDataGrid.SelectedItem as Customer;
    var orders = (from order in dbContext.Orders.Include("OrderItems")
                  where order.CustomerId == selectedCustomer.Id
                  select order);

    var sum = 0;
    foreach (var order in orders)
    {
        foreach (var item in order.OrderItems)
        {
            sum += item.UnitPrice * item.Quantity;
        }
    }

    this.customerOrderTotal.Text = sum.ToString();
}
```

Interaction/Business Logic



- 
- ❧ MVVM allows the code reusing ability to the different applications, we only need to worry about the design of all applications and coding (that includes business logic) can be shared across all applications.
  - ❧ we create model and write all code, commands, events (Business logic) in ViewModel once and use that Model and ViewModel in all the applications. We just need to make separate and different Views for each and every applications.



- ❖ `INotifyPropertyChanged`

- ❖ Commands

- ❖ <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/commanding-overview>

- ❖ Instead of events, commands have been used in the MVVM architecture. We cannot pass a parameter to the event but we can pass a parameter through a command.

# What is ICommand?

- ❧ ICommand defines a command

The ICommand type exposes the following members.

**CanExecute** :Defines the method that determines whether the command can execute in its current state.

**Execute** :Defines the method to be called when the command is invoked.

**CanExecutehanged** :Occurs when changes occur that affect whether or not the command should execute.



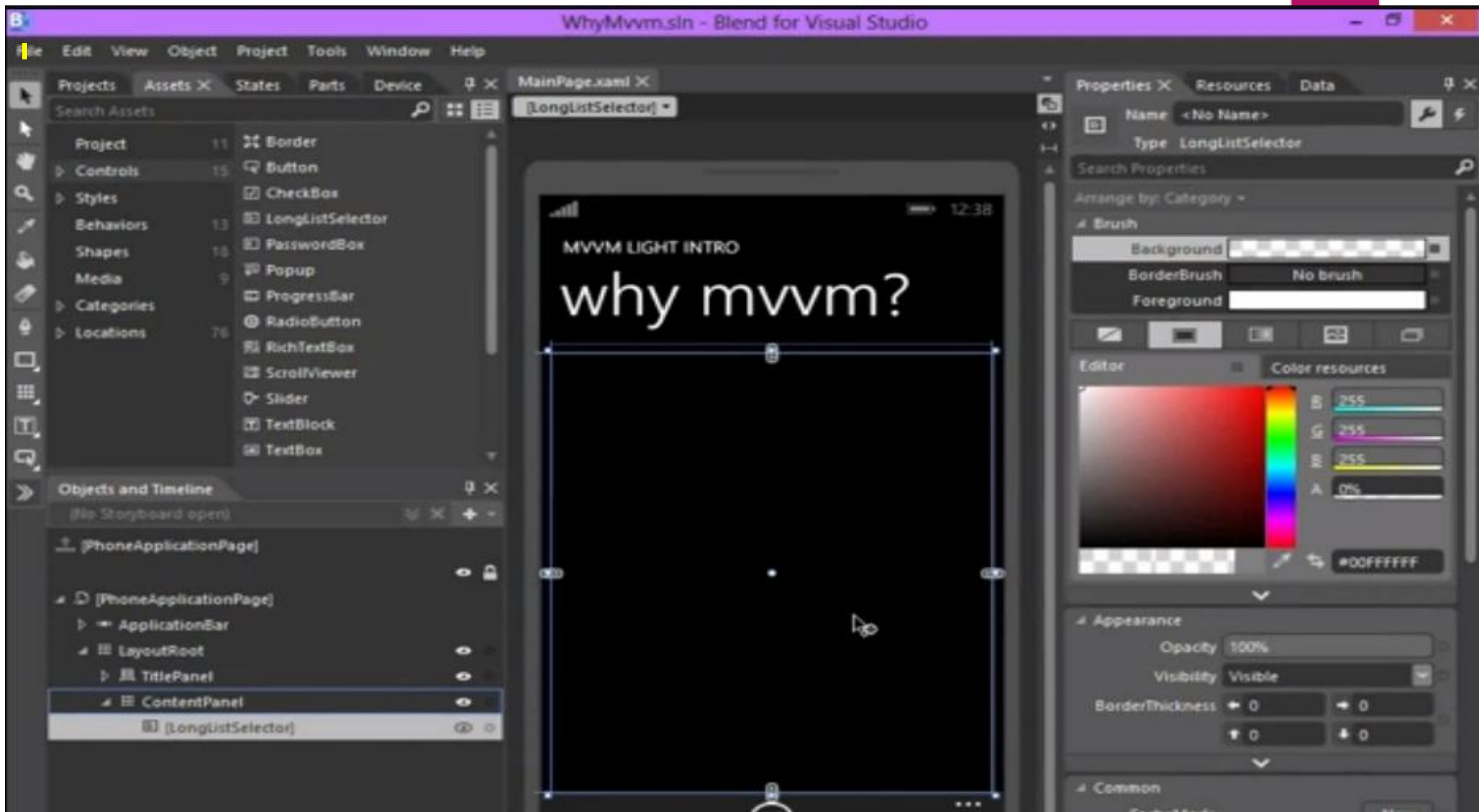
# What are popular MVVM frameworks?

- u MVVM Light
- u Prism
- u Caliburn Micro

# MVVM Light

GalaSoft, Laurent

- ↳ CommonServiceLocator



WPF:

```
var prop = DesignerProperties.IsInDesignModeProperty;  
_isInDesignMode = (bool)DependencyPropertyDescriptor  
    .FromProperty(prop, typeof(FrameworkElement))  
    .Metadata.DefaultValue;
```

# Design time output

```
u  if (IsInDesignMode)
u  {
u      Title = "Hello MVVM Light (Design Mode)";
u  }
u  else
u  {
u      Title = "Hello MVVM Light";
u  }
```

# Run time output

- When we run the application, we'll see the different title values for "Hello MVVM Light"



```
RaisePropertyChanged("propertyname");
```

# What are the disadvantages of MVVM?

- ⌞ Some people think that for simple UI, MVVM can be an overkill.
- ⌞ Similarly in bigger cases, it can be hard to design the ViewModel.
- ⌞ Debugging would be a bit difficult when we have complex data bindings.