

View Model Patterns

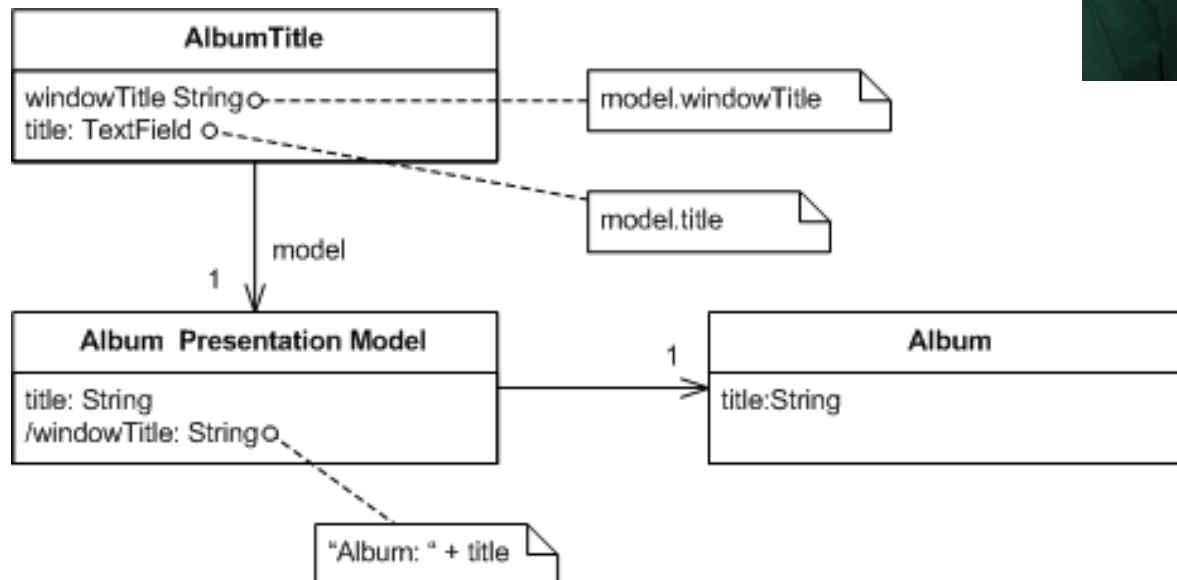
Michael L Perry

Michael@qedcode.com



Presentation Model

- Martin Fowler, 2004
- Separation Pattern
- Remove state and behavior from the view



Model View ViewModel

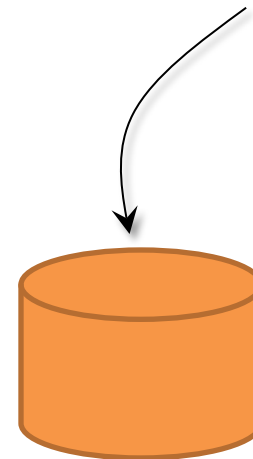
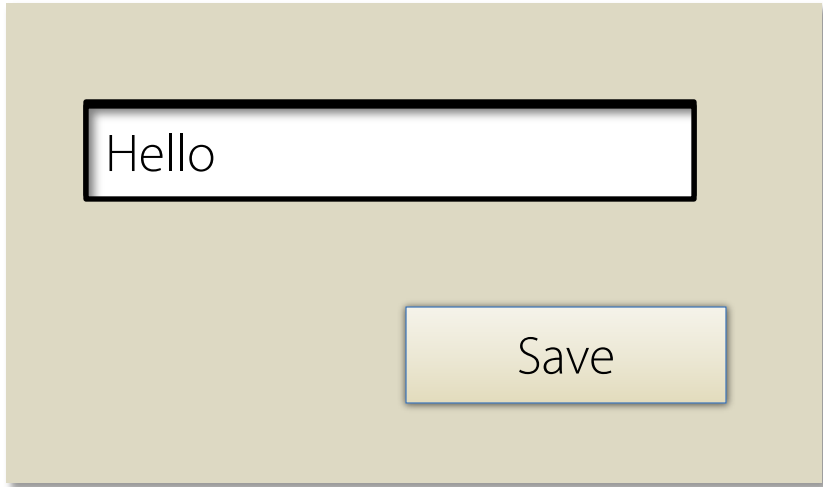
- John Gossman, 2005
- Presentation Model for XAML
- CLR data binding



Family of Patterns

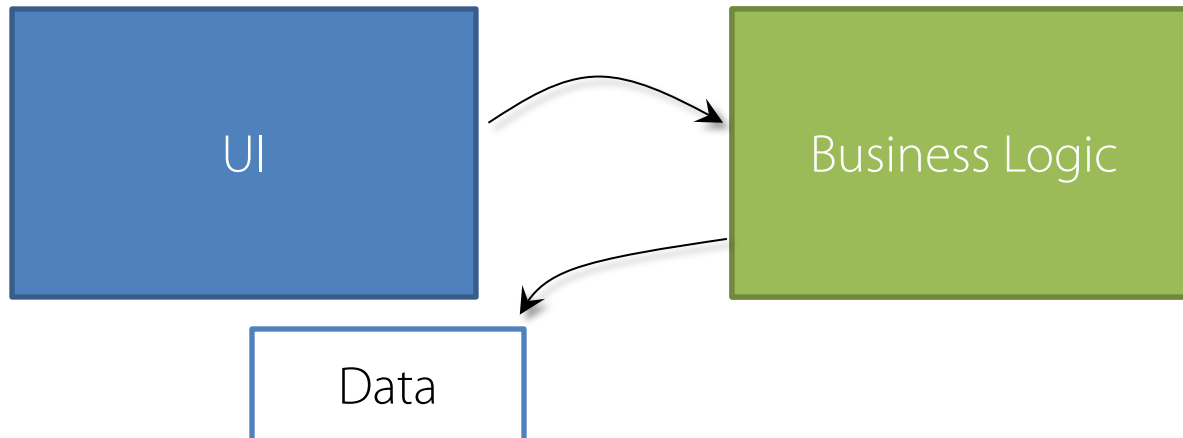
- **Opinions based on constraints**
 - How to respond to events
 - How to communicate among view models
 - How to pair a view model with a view
- **Goals**
 - Isolation
 - Testability
 - Resiliency

Stateful View Models



Problems with Data in the UI

- **Hard to test**
 - Can't create an instance of form
- **Couples business logic to UI**
 - Business logic depends upon UI
 - UI invokes business logic



Stateful View Model Libraries



Reactive View Model

```
SearchTermTextBox.TextChanged += MyEventHandler;
```

```
void MyEventHandler(object sender, TextChangedEventArgs e)
{
    Debug.WriteLine("You changed the text.");
}
```

```
void MyNextEventHandler(object sender, RoutedEventArgs e)
{
    Debug.WriteLine("You changed the text and clicked the button.");
}
```

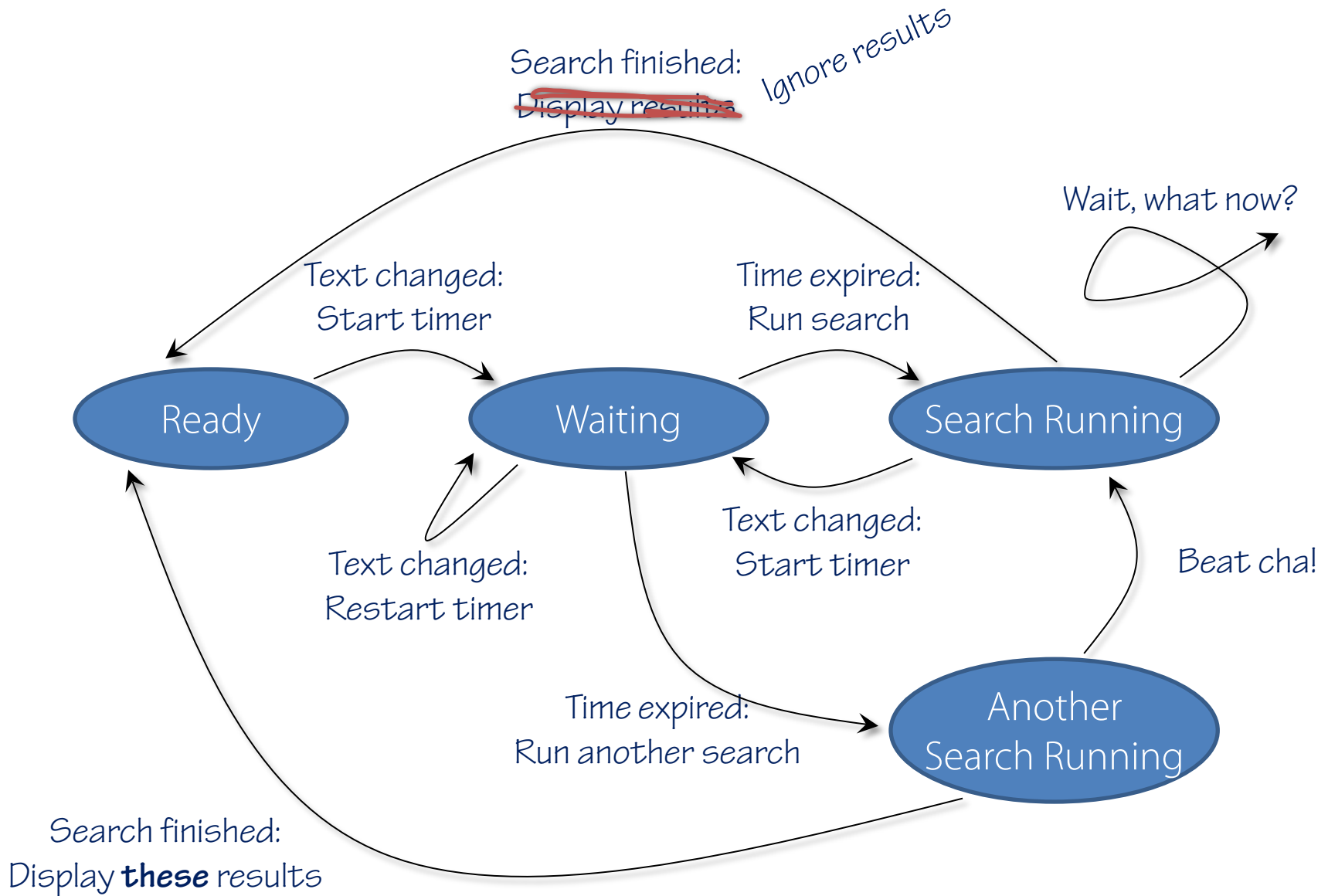

Search

Double Rainbow

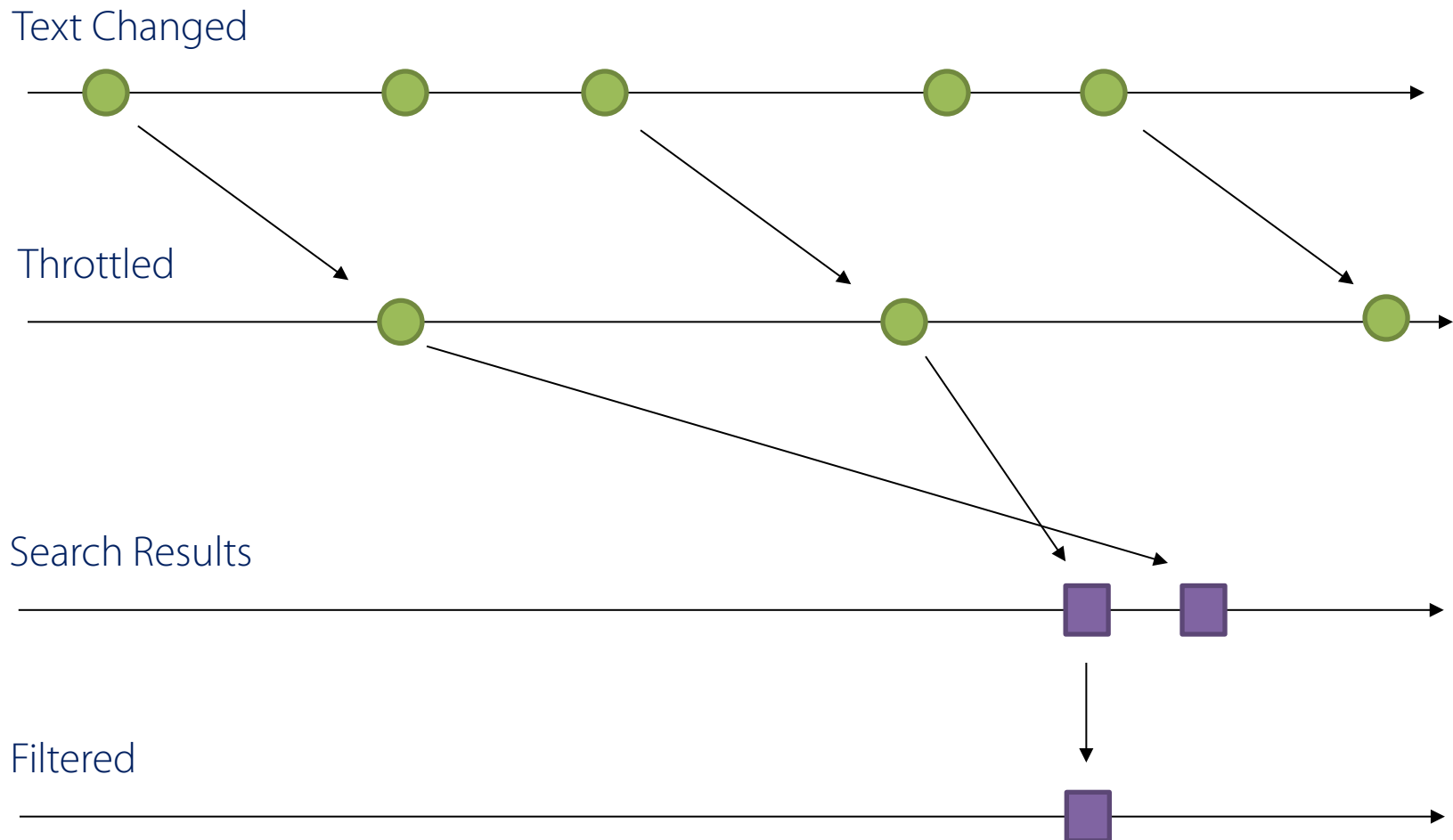


D
Do
Dou
Doub
Doubl
Double
Double
Double R
Double Ra
Double Rain
Double Rainb
Double Rainbo
Double Rainbow

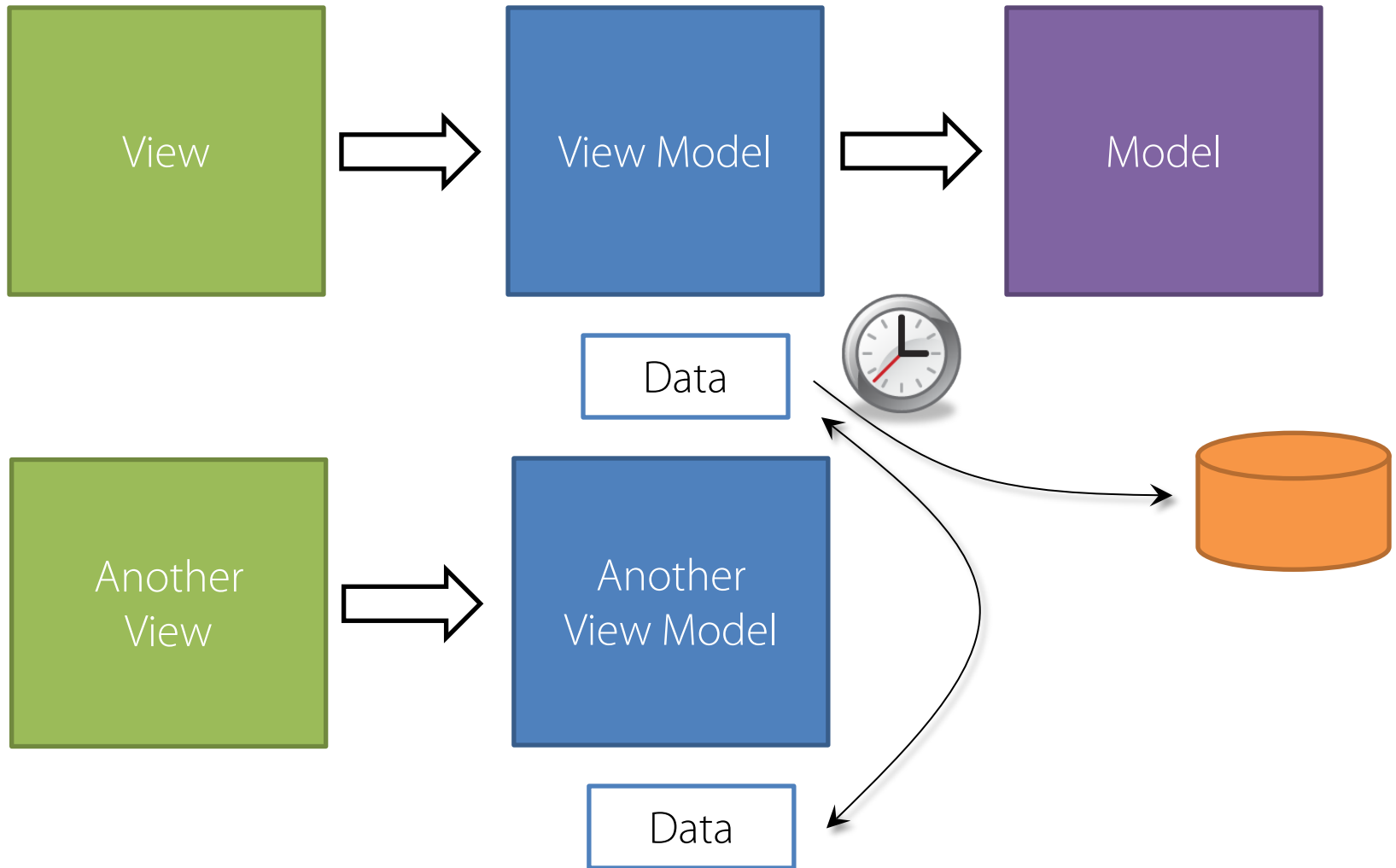
Events



Streams



Stateless View Model



What is State?

```
public class MainViewModel
{
    private string _thisIsState;
}
```

```
public class MainViewModel
{
    private string ThisIsToo { get; set; }
}
```

```
public class MainViewModel
{
    // This is not.
    private readonly Model _model;

    public MainViewModel(Model model)
    {
        _model = model;
    }
}
```

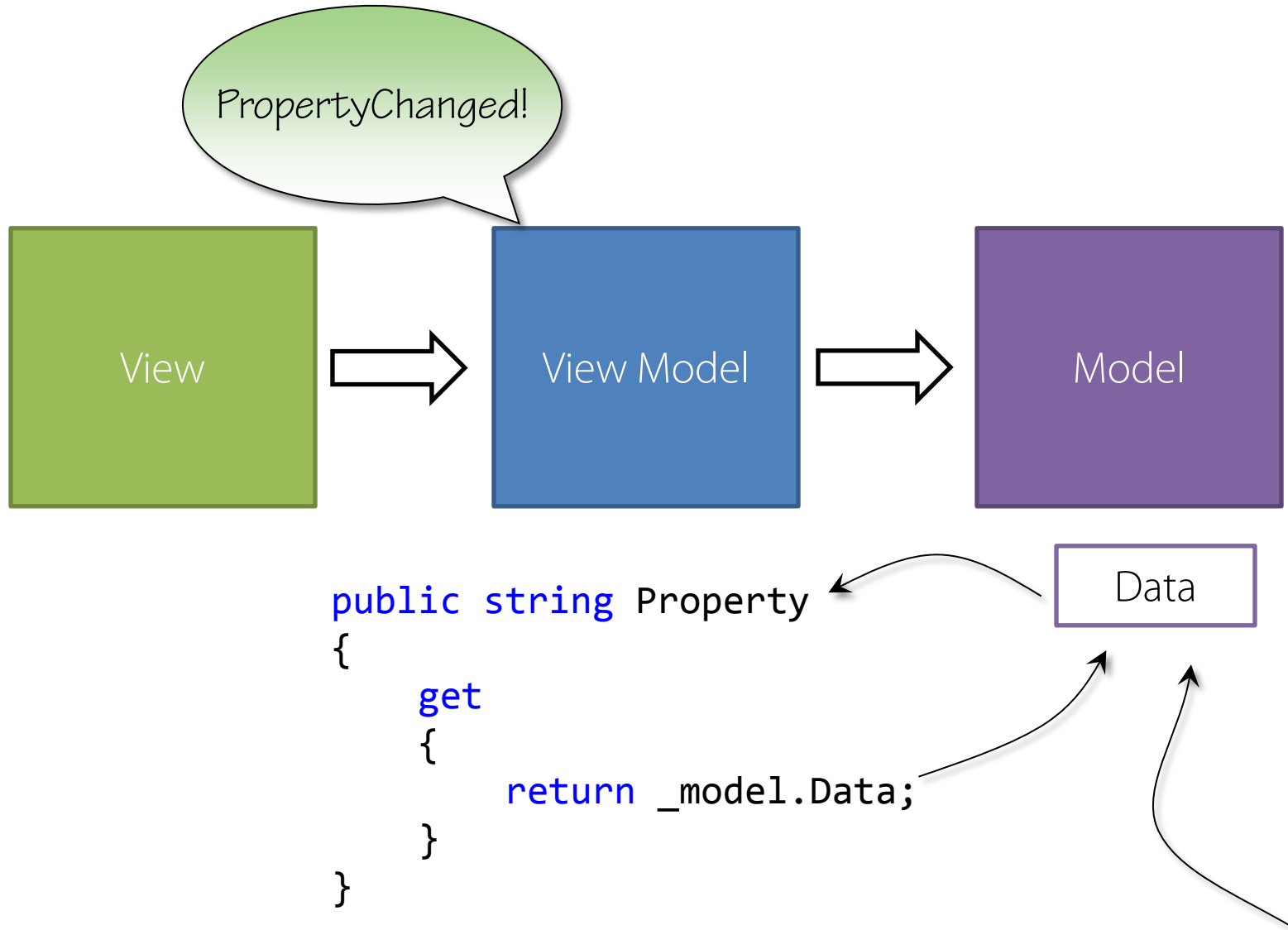
What is State?

```
public class MainViewModel
{
    private readonly Model _model;

    public MainViewModel(Model model)
    {
        _model = model;
    }

    // This is not state either.
    public string Property
    {
        get { return _model.Property; }
        set { _model.Property = value; }
    }
}
```

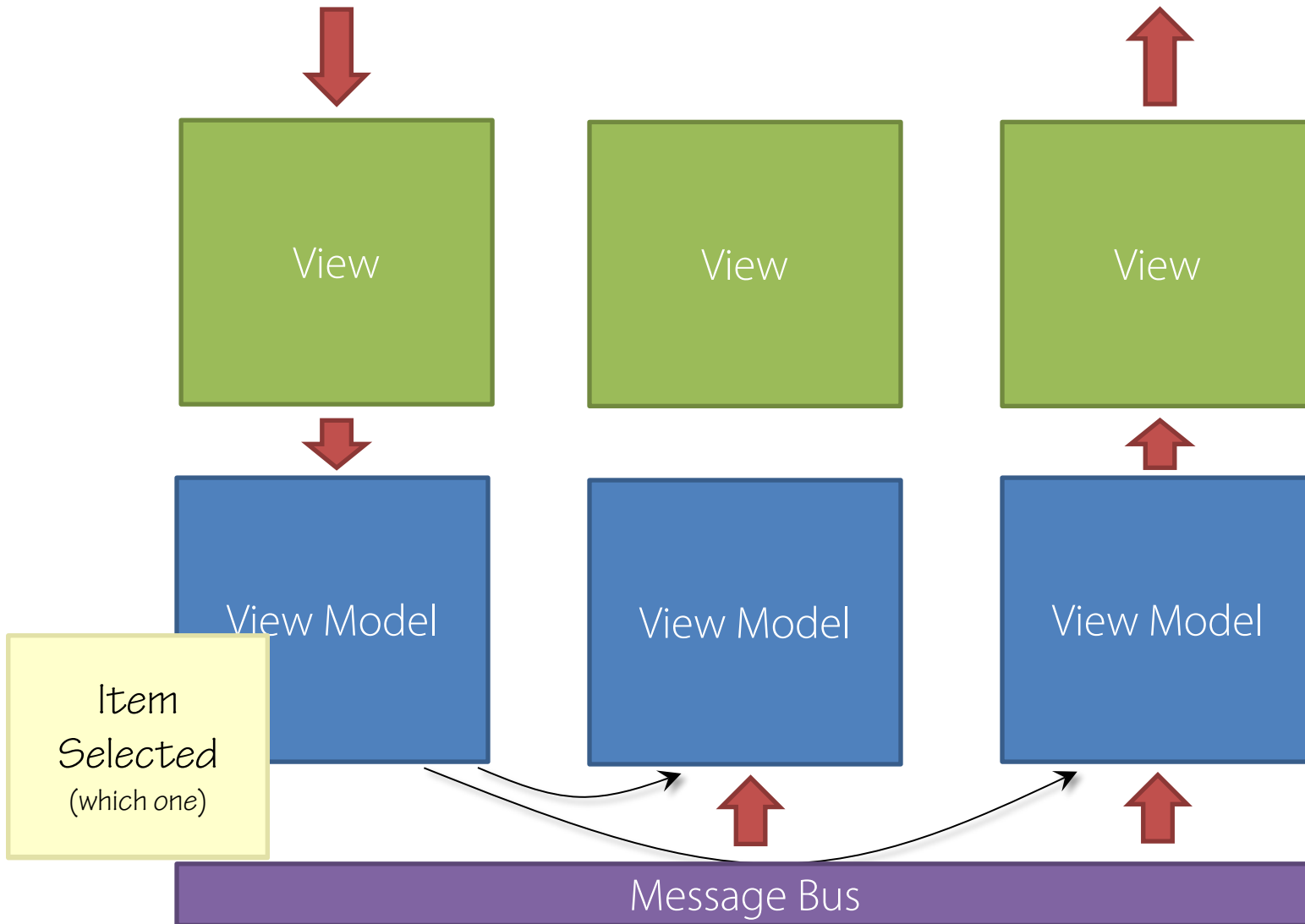
INotifyPropertyChanged



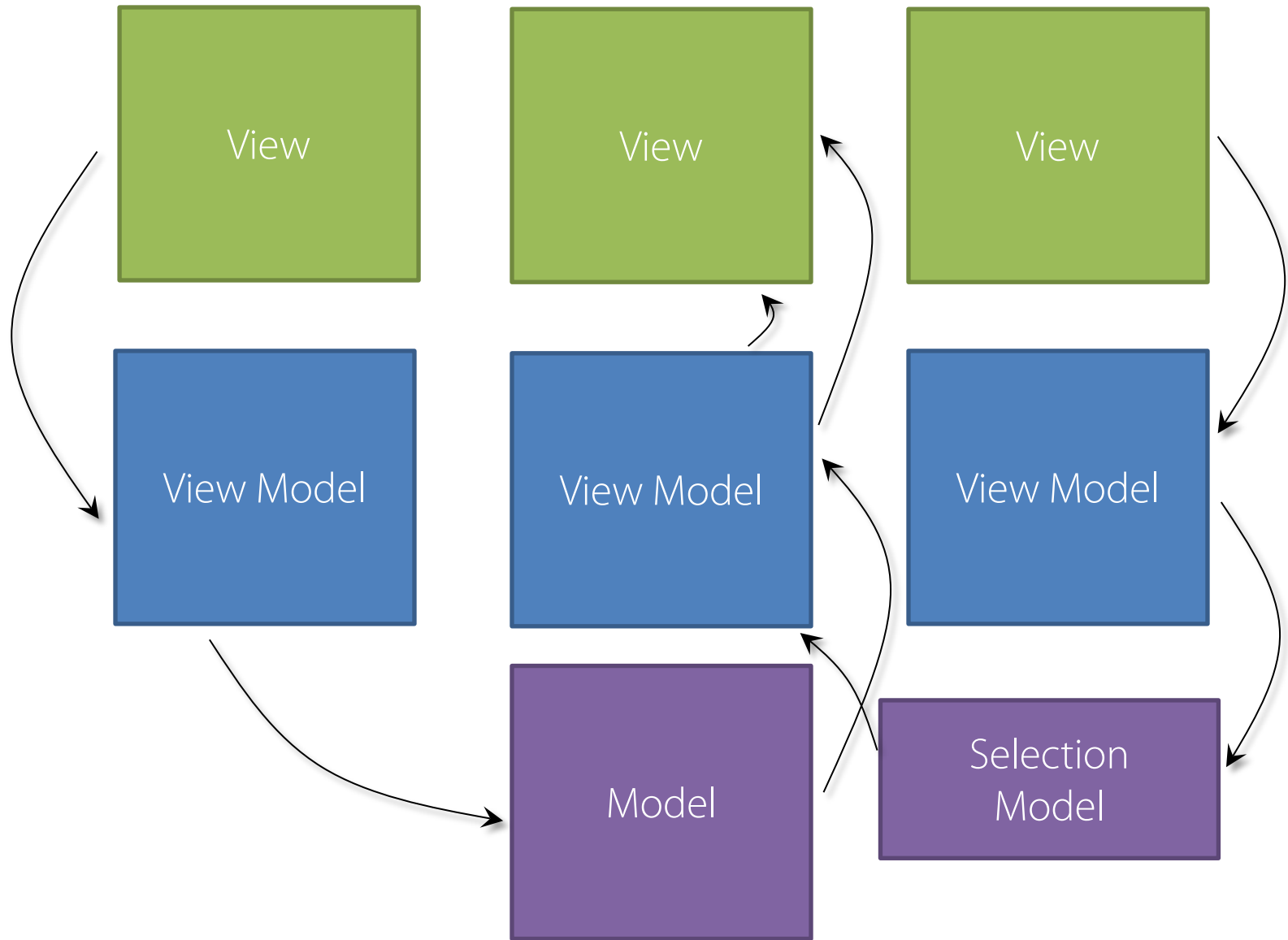
Dependency Tracking



Message Bus



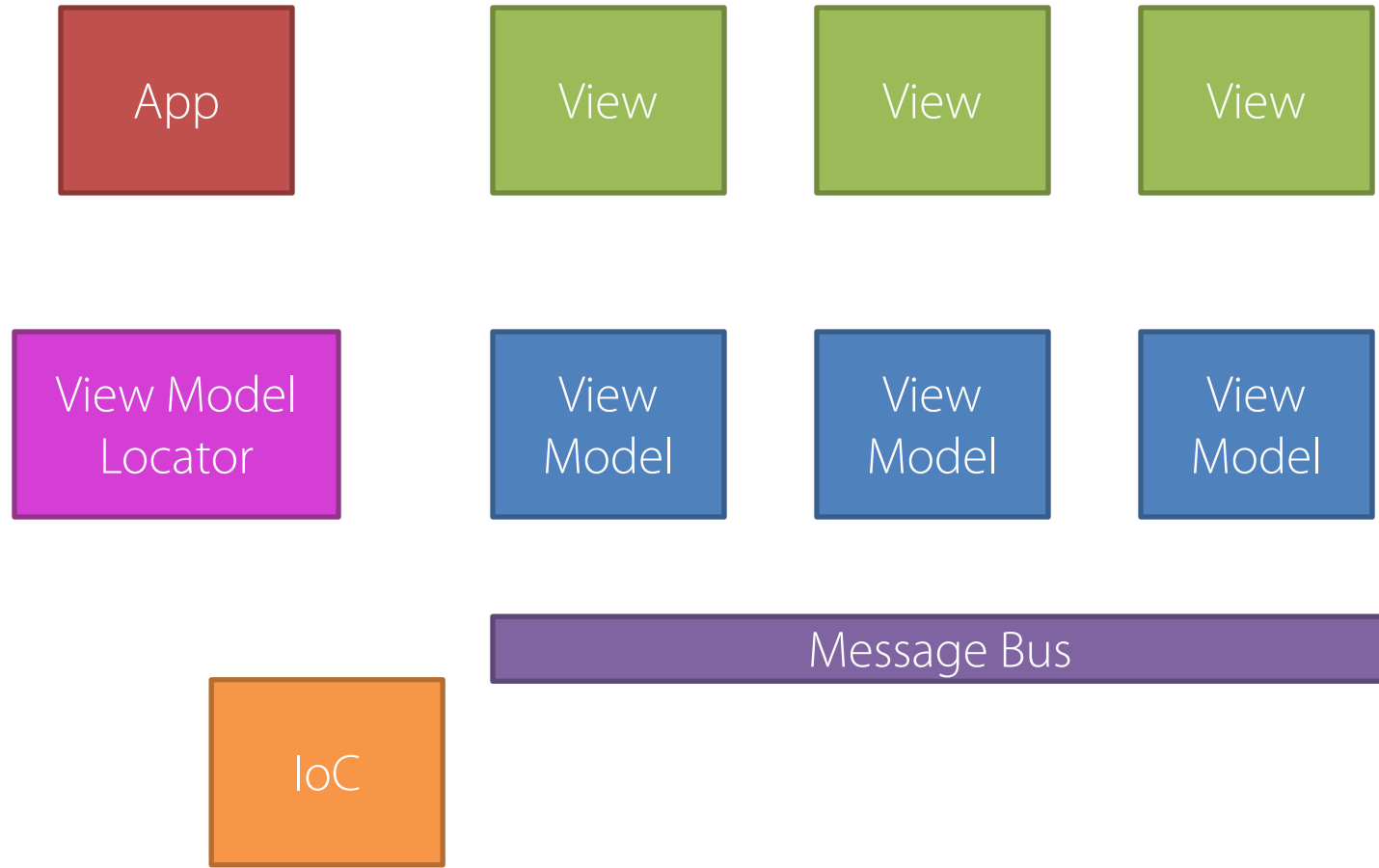
Selection Model



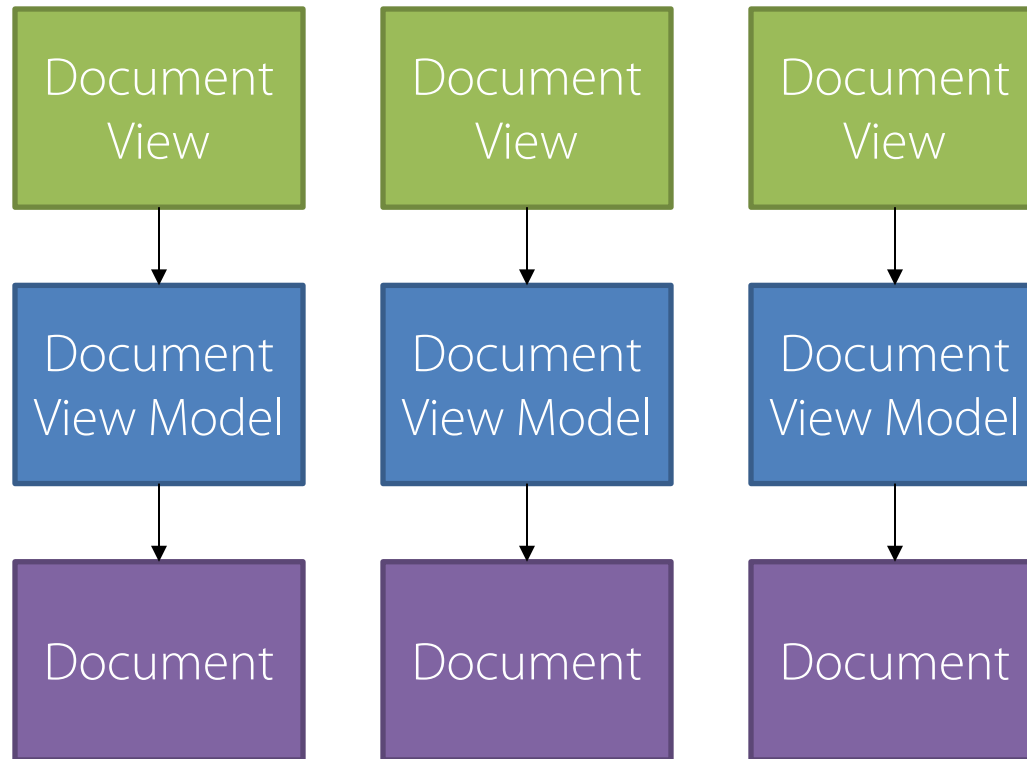
View Model Locator

```
private void MainView_Loaded(object sender, RoutedEventArgs e)
{
    IRepository repository = new Repository();
    IServiceAgent serviceAgent = new ServiceAgent();
    DataContext = new MainViewModel(
        repository,
        serviceAgent);
}
```

View Model Locator



View Model First



View Model First

- **Implicit Data Templates**

- Register all views as data templates
- Add to a resource dictionary

- **Content Controls**

- Set the Content property to the view model
- The control selects the view

View Services

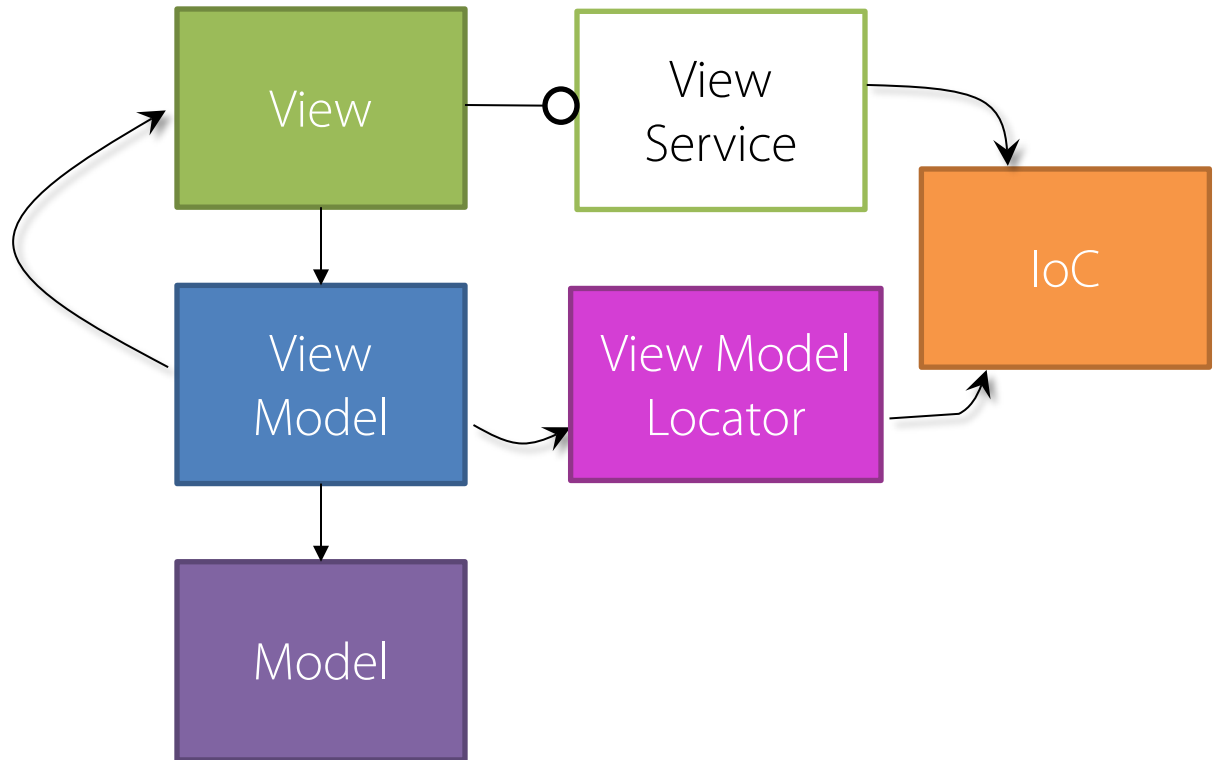


Laurent Bugnion

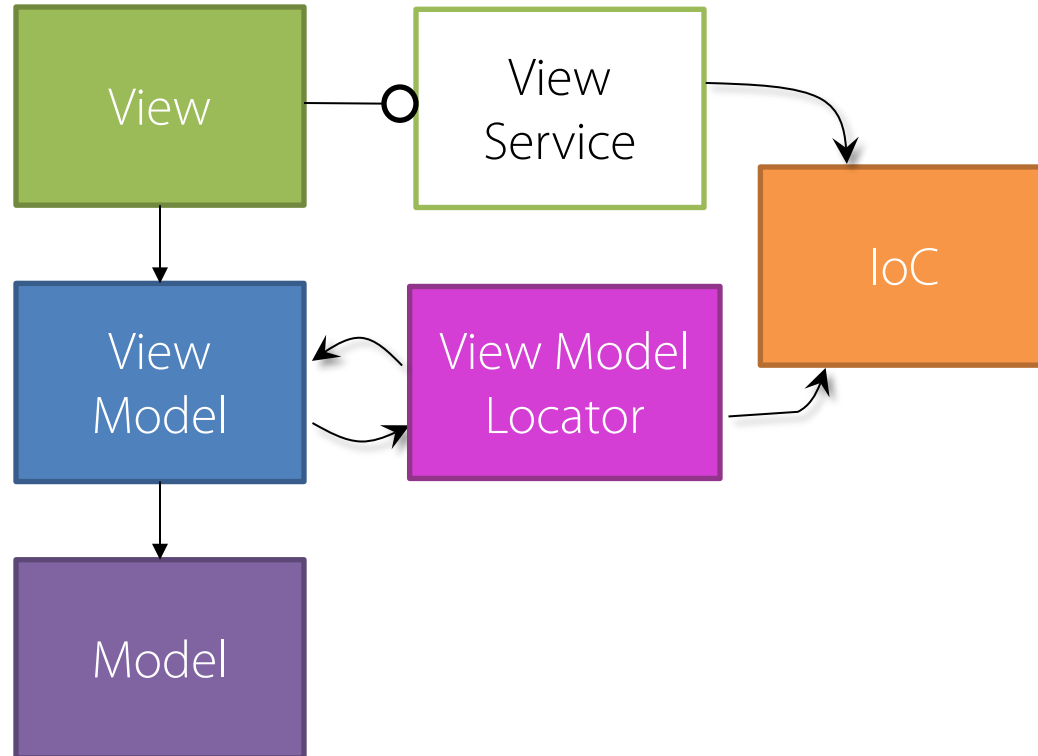
Messenger and View Services in MVVM

MSDN Magazine March 2013

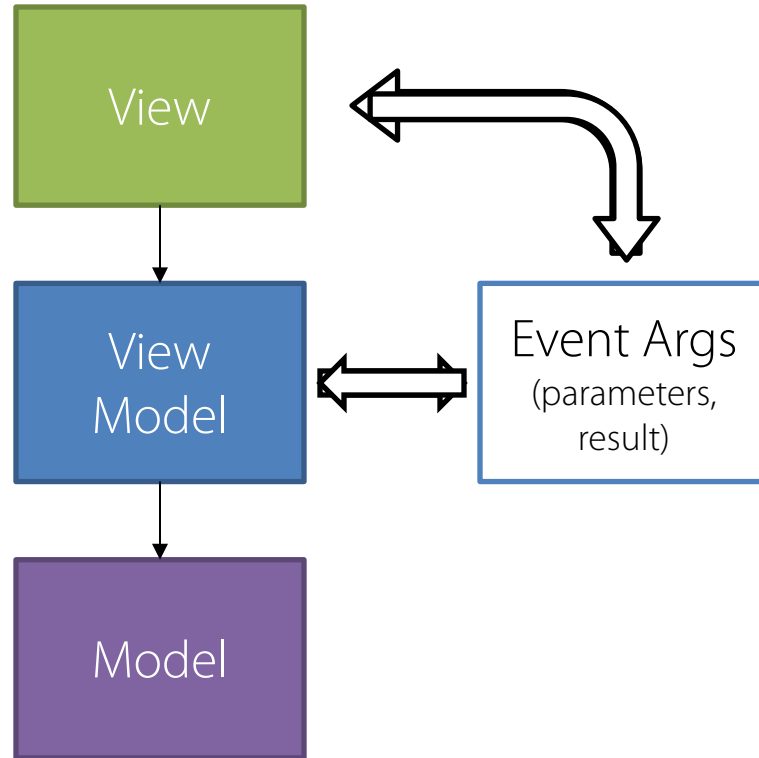
Don't Look Up!



View Model Events



View Model Events



Conclusion

- **Styles of view model**
 - Stateful
 - Reactive
 - Stateless
- **Communication**
 - Message Bus
 - Selection Model
- **Pairing with View**
 - View Model Locator
 - View Model First
- **Call View Methods**
 - View Service
 - View Model Events