

Validating User Input



Thomas Claudius Huber

MICROSOFT MVP (WINDOWS DEVELOPMENT)

@thomasclaudiush www.thomasclaudiushuber.com



Module Outline



Validation in WPF

Validate user input with
`INotifyDataErrorInfo`

Display errors by using
an error template

Refactor validation logic
into a base class

Build logic to validate
Data Annotations

Enable and disable the save button



Demo

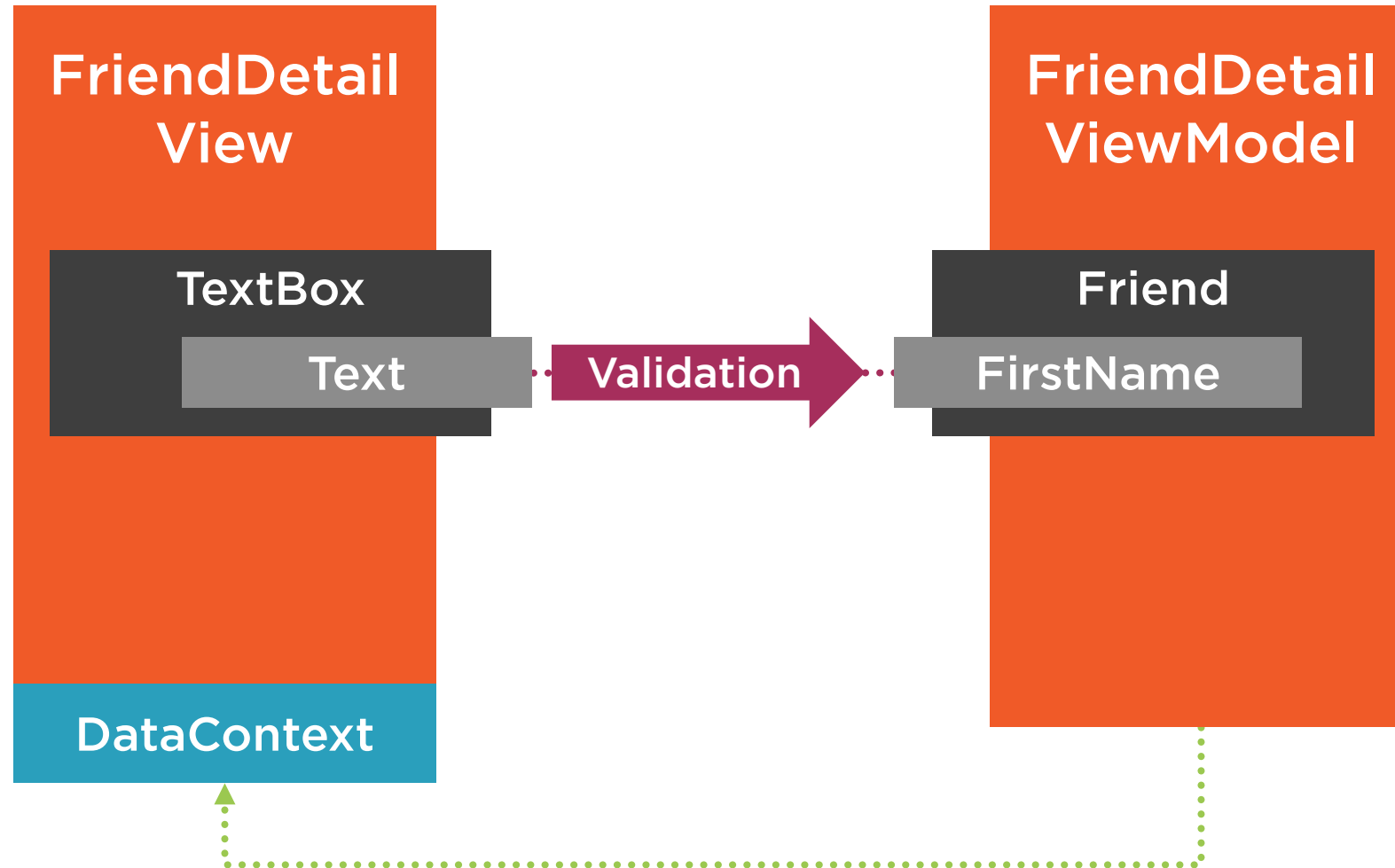


Save a friend without a firstname

Process unhandled exceptions



Validation in WPF



Validation in WPF

Validation Rules

Exceptions

ValidatesOnExceptions

IDataErrorInfo

ValidatesOnDataErrors

INotifyDataErrorInfo

ValidatesOnNotifyDataErrors

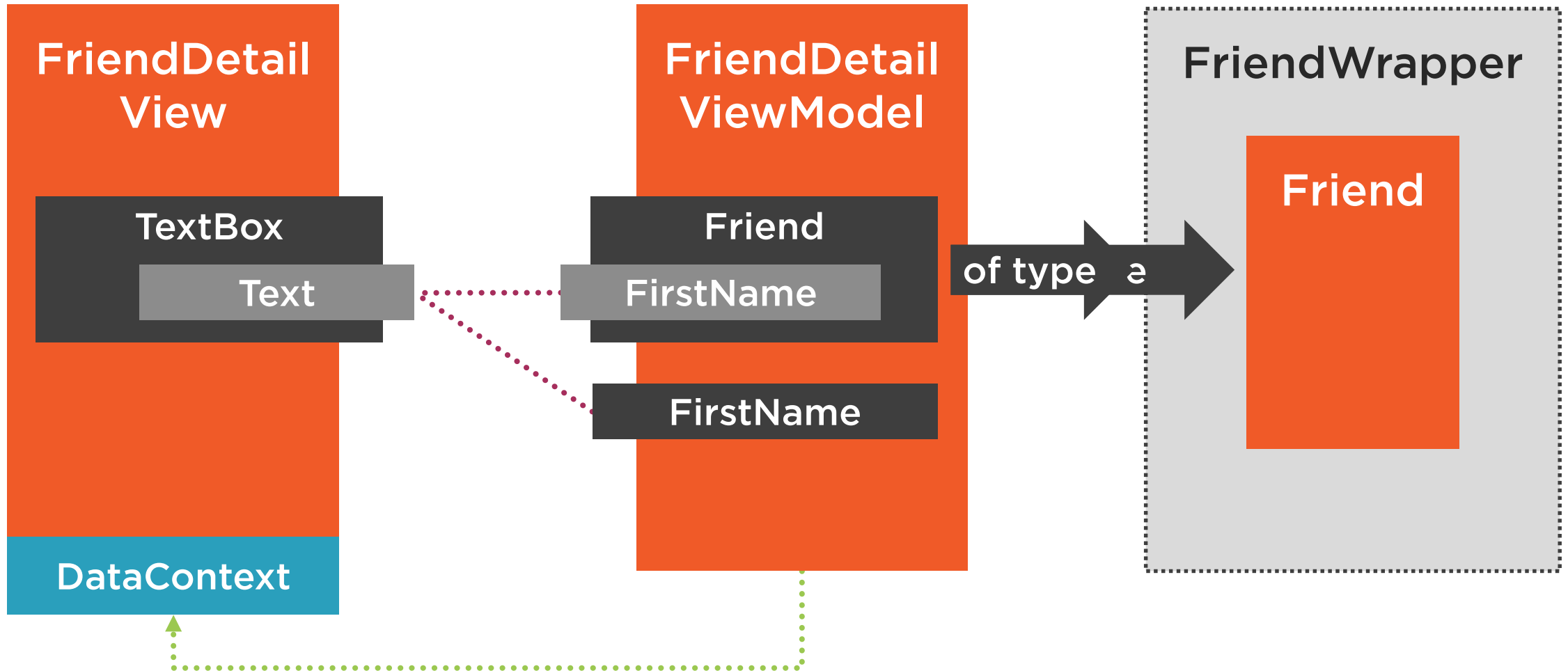


Validation in WPF

```
public interface INotifyDataErrorInfo
{
    bool HasErrors { get; }
    bool EventHandler<DataErrorsChangedEventArgs> ErrorsChanged;
    IEnumerable GetErrors(string propertyName);
}
```



Wrap the Model



Demo



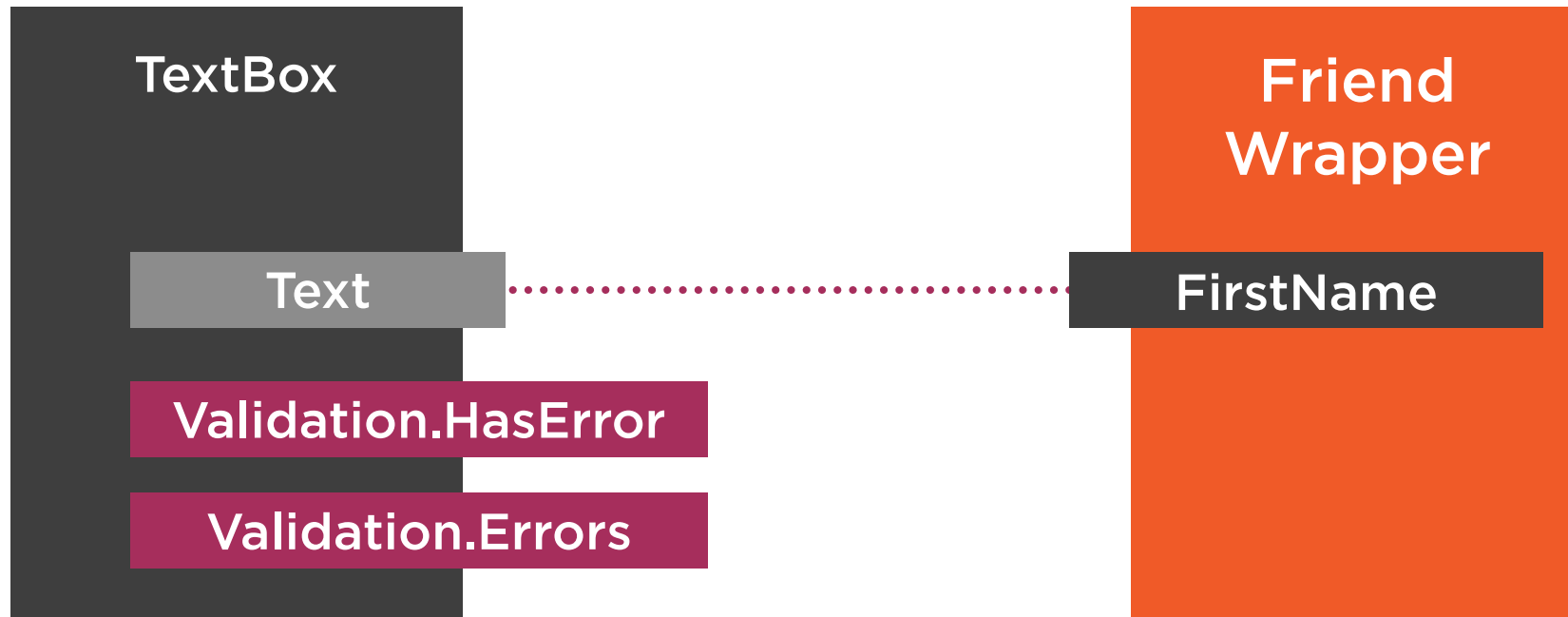
Create the FriendWrapper

Implement INotifyDataErrorInfo

Add some validation logic



Display the Error in the Tooltip



Demo

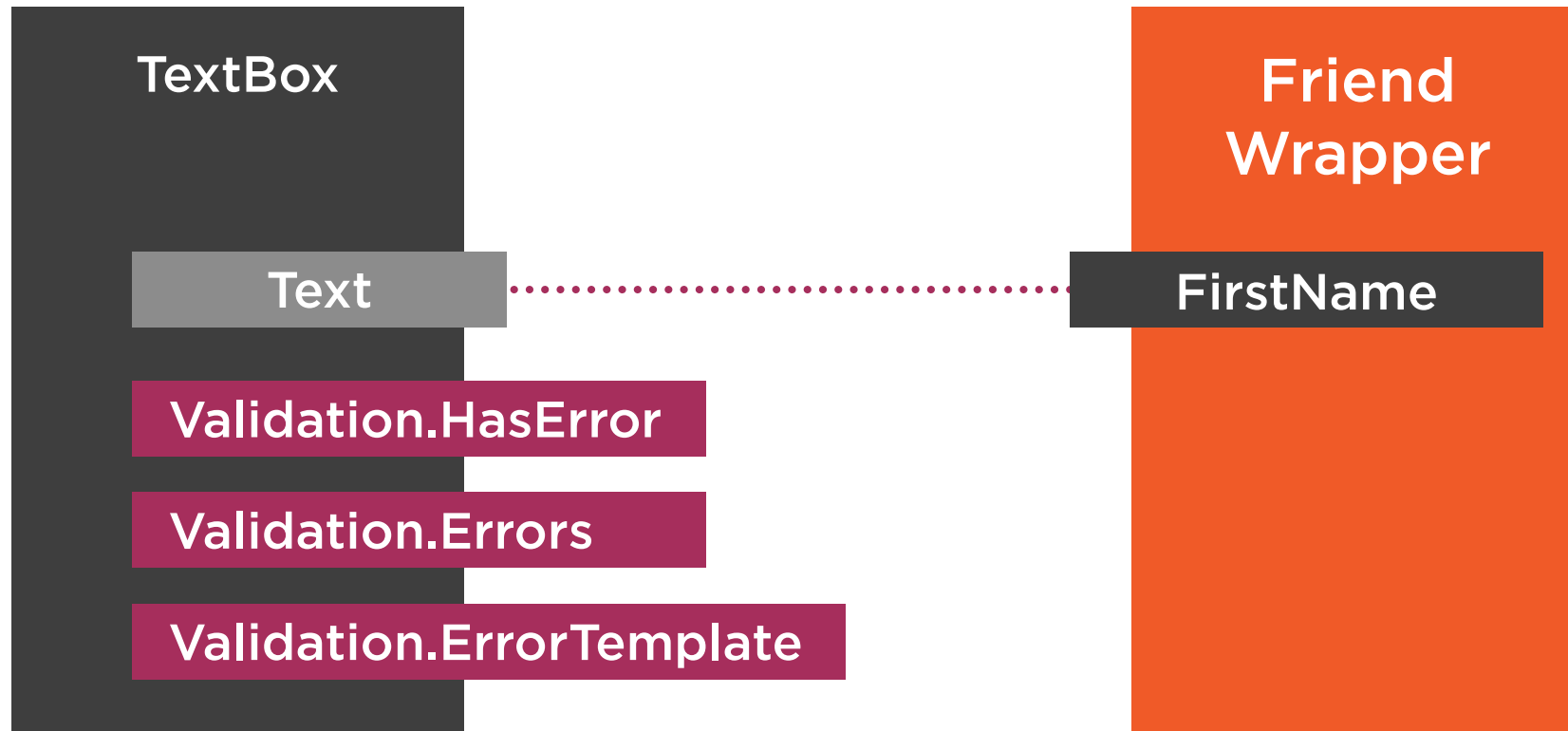


Use a Style for a TextBox

Display the error in the tooltip



Display the Error Below the TextBox



Demo



Define an error template

Display the error below the TextBox



Demo



Create a `NotifyDataErrorInfoBase` class

Introduce a `ModelWrapper` class

Trigger validation in the `ModelWrapper`



Validate by Using Data Annotations

```
public class Friend
{
    [Required]
    [StringLength(50)]
    public string FirstName { get; set; }
    ...
}
```



Validate by Using Data Annotations

Validator

ValidationContext

ValidationResult



Demo



Validate by using Data Annotations



Demo



Enable and disable the save button



Summary



Implement INotifyDataErrorInfo in a base class

Display errors with an error template

Validate by using Data Annotations

Enable and disable the save button

Course:

WPF and MVVM:

Advanced Model Treatment

