WPF and MVVM: Advanced Model Treatment

Introduction

Thomas Claudius Huber

@thomasclaudiush | www.thomasclaudiushuber.com

The Model-wrapper Functionality

Change notification

Change tracking

Validation

Course Outline

1. Introduction
2. Notifying About Model Changes
3. Tracking Model Changes
4. Displaying Model Changes
5. Validating the Model
6. Displaying Validation Errors
7. Generating Model-wrappers with T4

Why Are Services Using POCOs?

```
public class Friend
{
  public int Id { get; set; }
  public string FirstName { get; set; }
  public Address Address { get; set; }
  public IList<FriendEmail> Emails { get; set; }
}
```

POCO
= Plain old CLR object

In contrast to a DTO
a POCO can have behavior

Loose coupling → No dependencies to a framework

Code first scenarios → Create the model without thinking about persistence

Lightweight model → No boilerplate code

Challenges with POCOs in the Client

```
public class Friend
{
  public int Id { get; set; }
  public string FirstName { get; set; }
  public Address Address { get; set; }
  public IList<FriendEmail> Emails { get; set; }
}
```

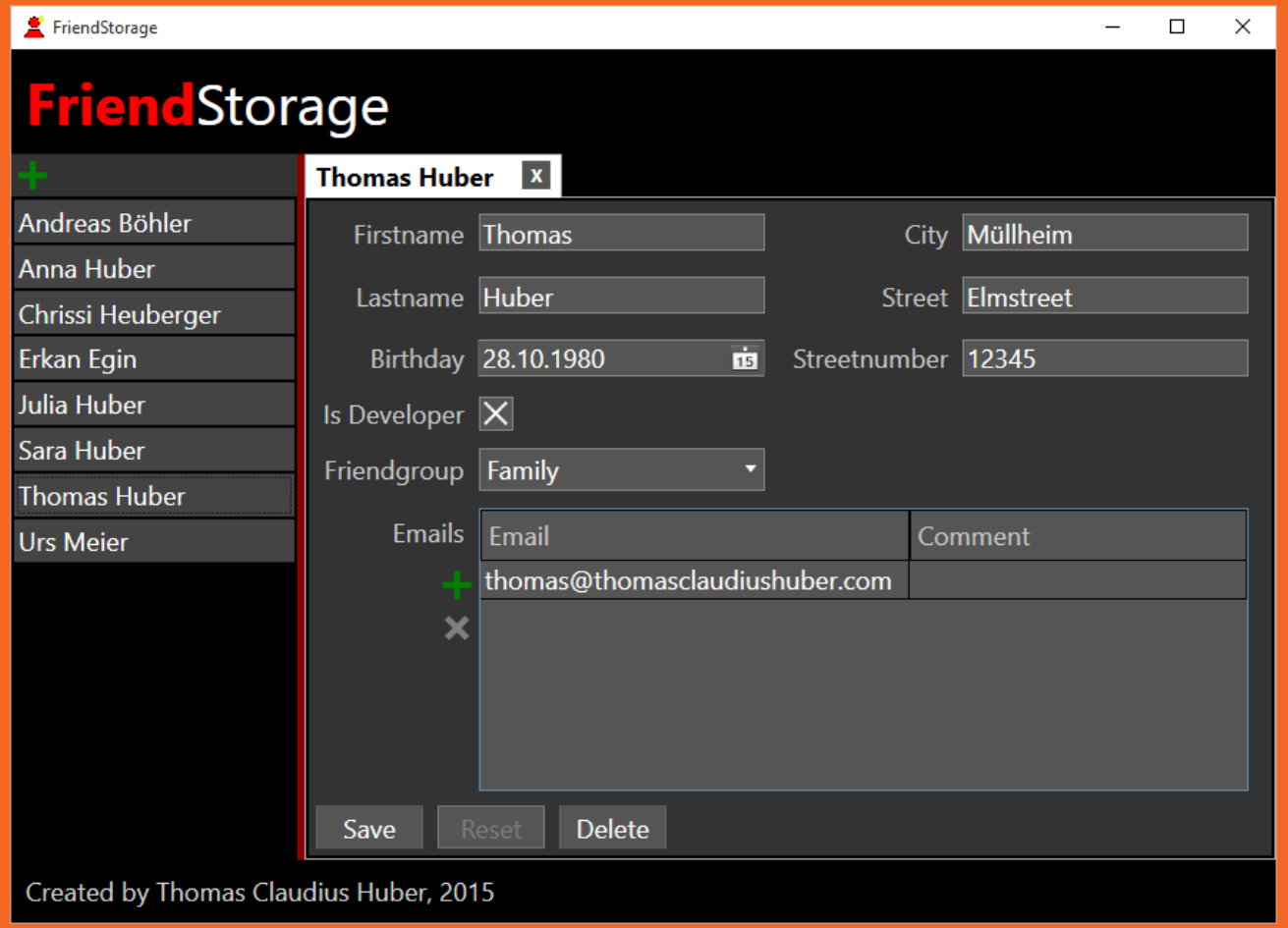
No change notification → INotifyPropertyChanged is missing
INotifyCollectionChanged is missing for collections

No change tracking → No information if the model is changed or not

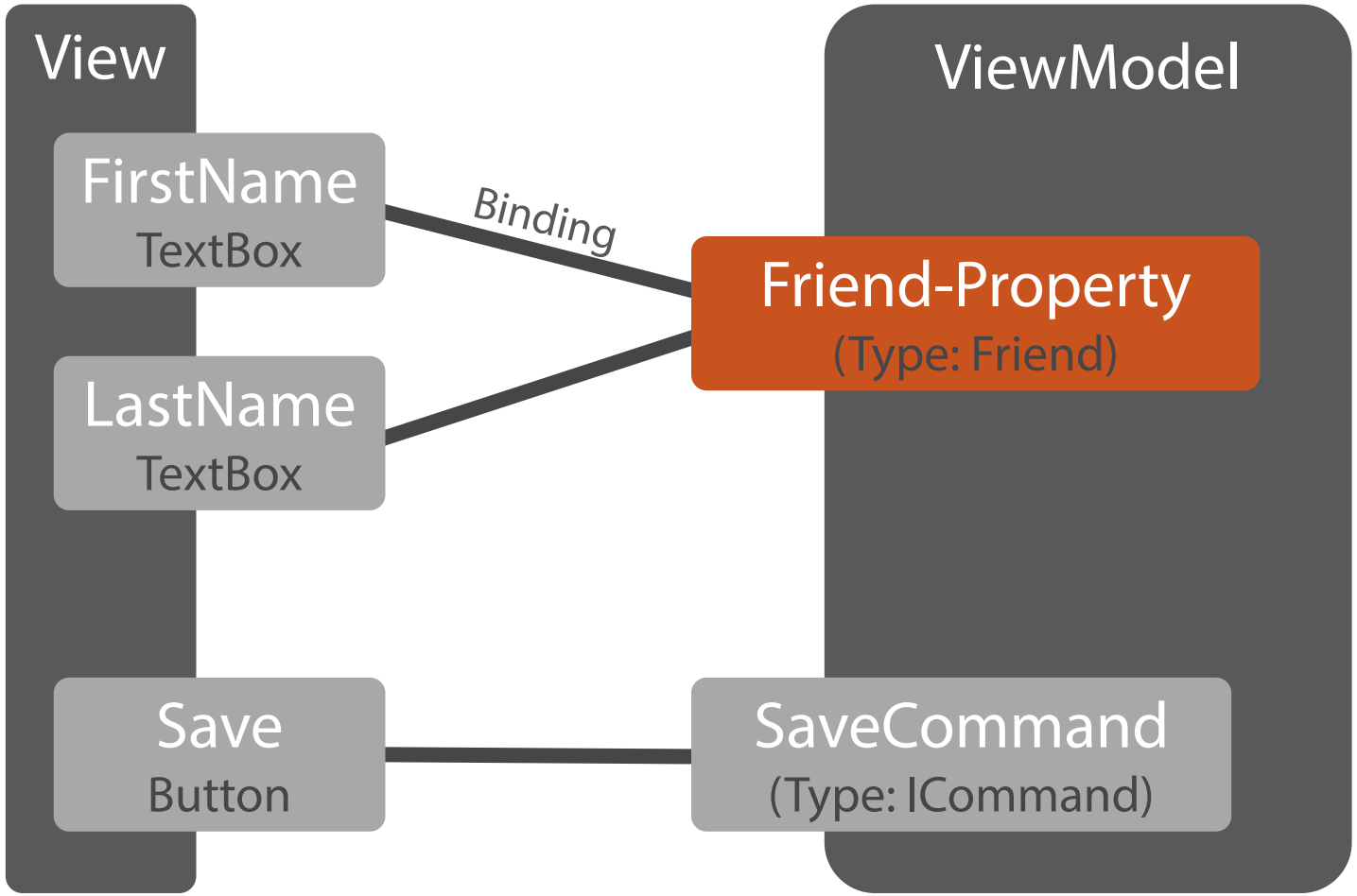
No UI-validation support → IDataErrorInfo / INotifyDataErrorInfo is missing

The FriendStorage-application

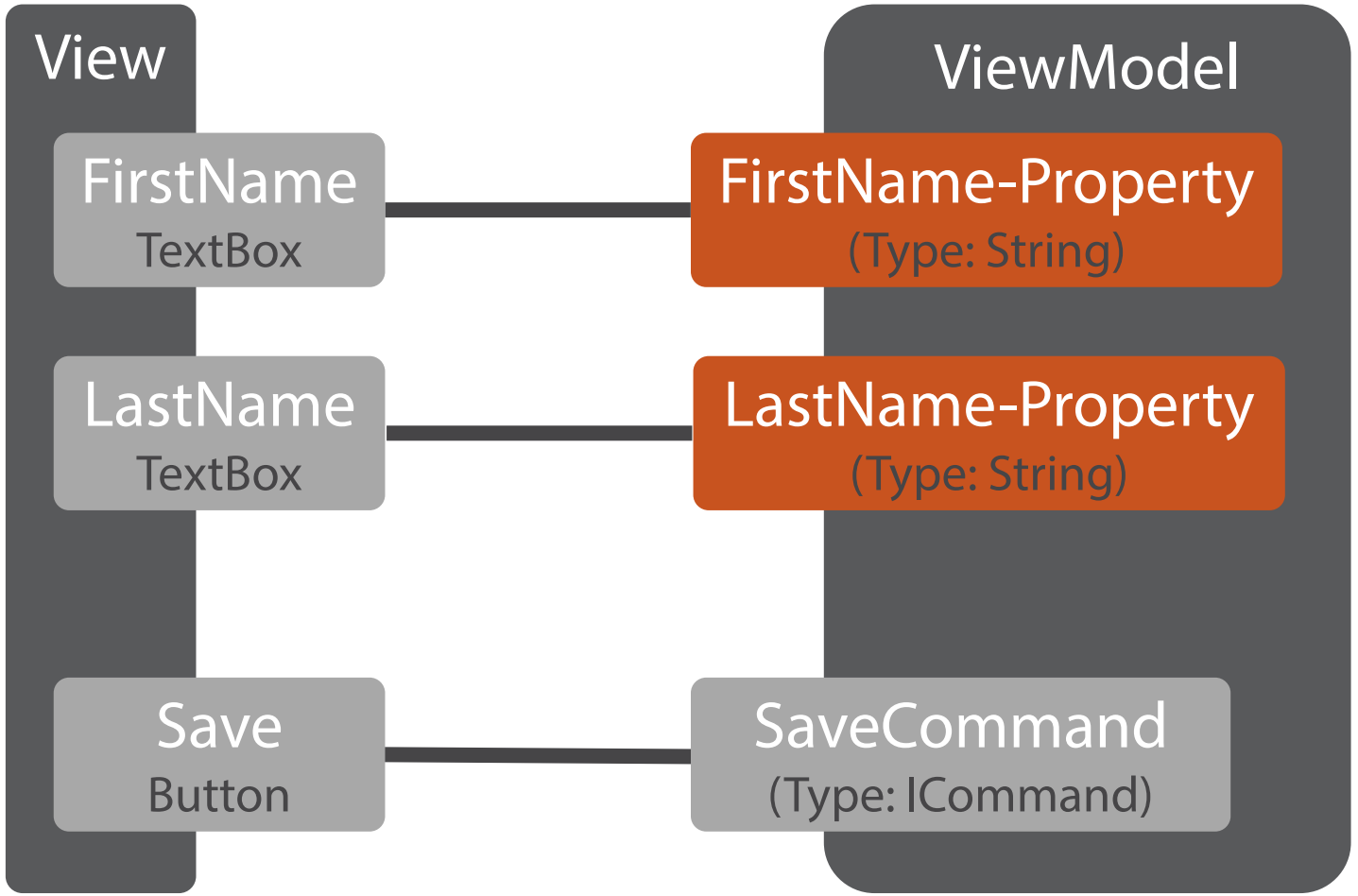
- Get familiar with the application and its code
- ✓ Views and ViewModels
- ✓ Dependency injection
- ✓ Event aggregator
- The challenges with POCOs



Mastering POCOs in a ViewModel



Mastering POCOs in a ViewModel



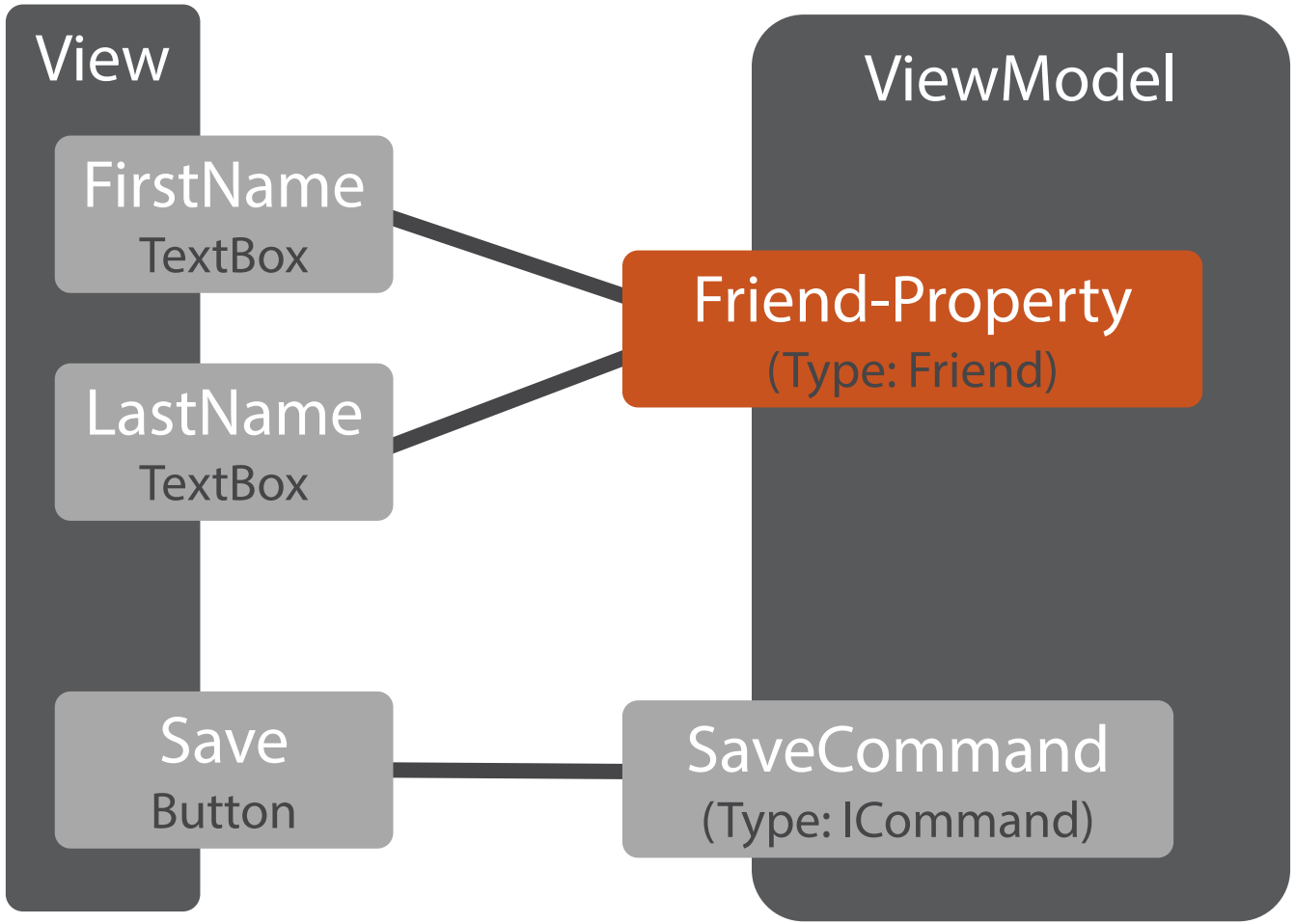
Implements
INotifyPropertyChanged
and INotifyDataErrorInfo

Every ViewModel using the Friend needs to redo the logic

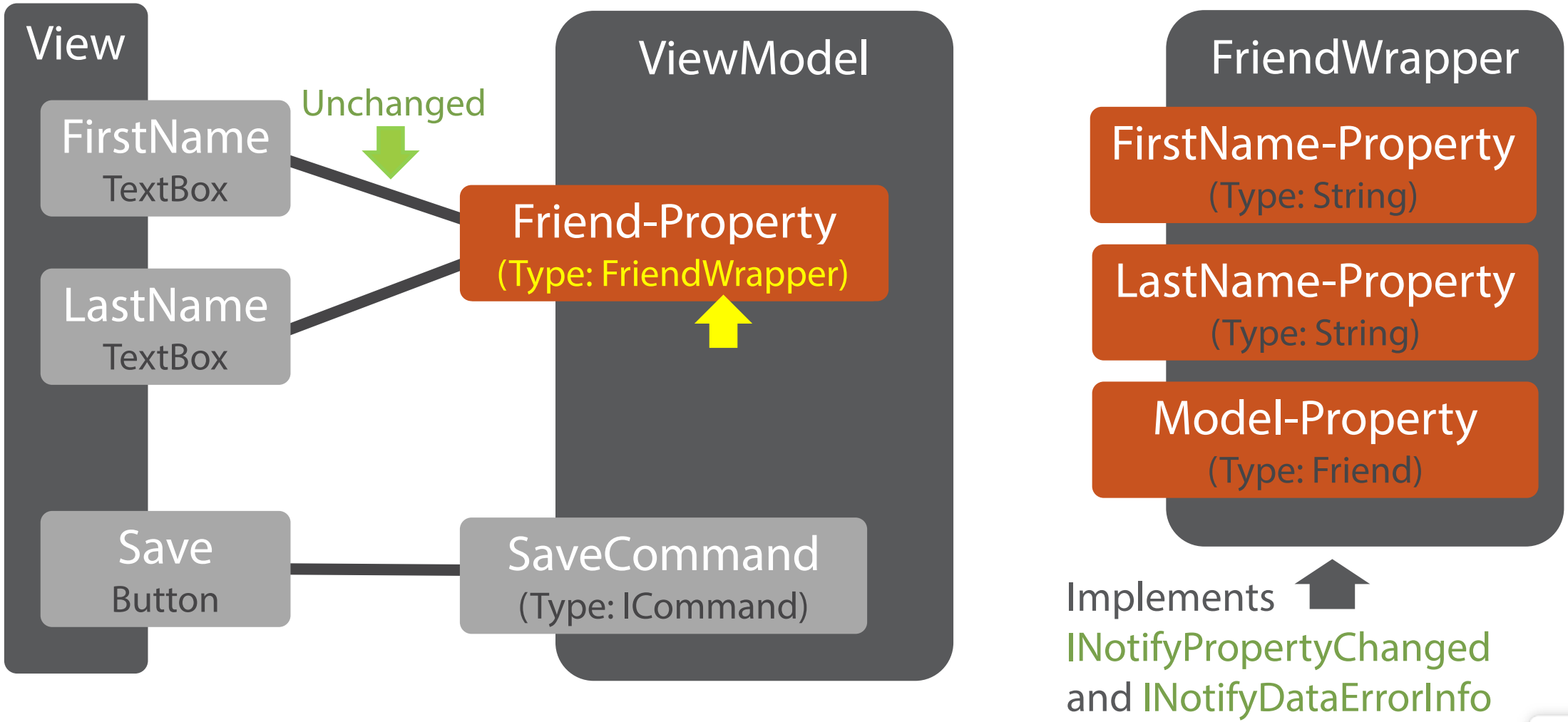
Logic is not separated from the ViewModel:

- Leads to big ViewModels
- Hard to auto-generate

Mastering POCOs in a Model-wrapper



Mastering POCOs in a Model-wrapper



Summary

- Challenges with POCOs
- The FriendStorage-application
- Mastering POCOs in the client