

MVVM und TDD: Testbare ViewModels

 **Developer
Week 2016**



Thomas Claudius Huber
@ThomasClaudiusH

BASEL ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier. ■ ■ ■

■ Thomas Claudius Huber

Principal Consultant @ Trivadis AG
Cloud Solutions

- Developer, Trainer, Architect
- Microsoft MVP for Windows Development

Spezialisiert in WPF, Angular 2, C#, .NET, Azure

What else:

- Pluralsight-Autor, Buchautor
- Fussball, Skaten, Fitness, Gitarre

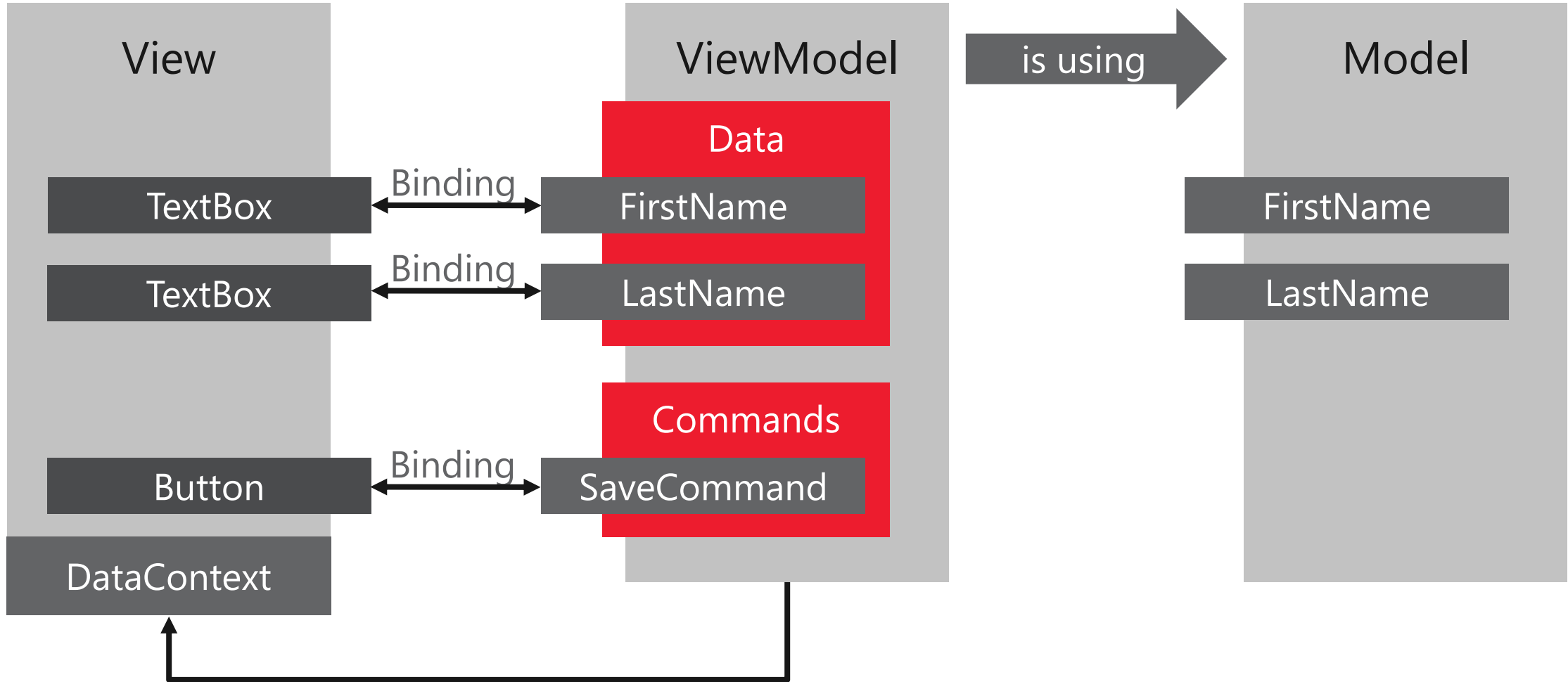




“Code without tests
is bad code”

Michael Feathers

■ Das MVVM Pattern



■ Vorteile von MVVM

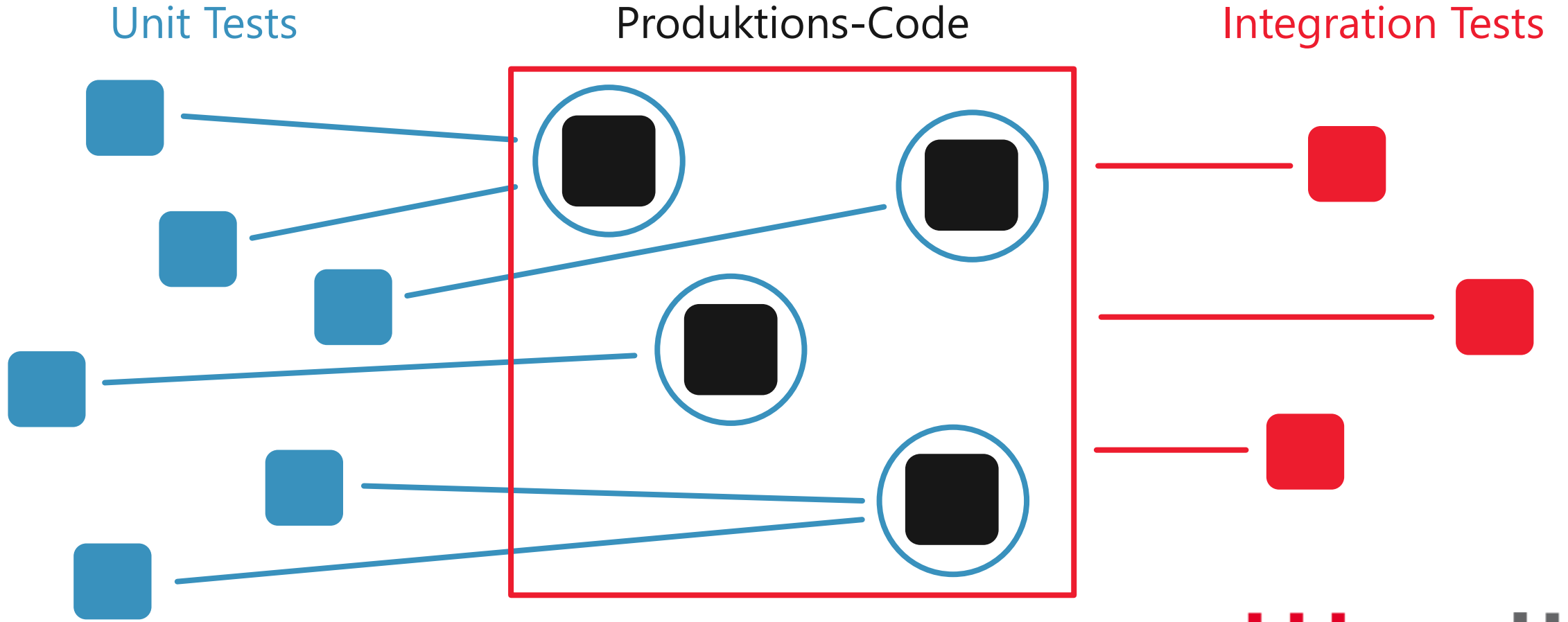
- Wartbarkeit
- Separation of Concerns
- Testbarkeit

■ Agenda

1. Unit Testing und TDD basics
2. Testbare ViewModels schreiben
3. Moq, Autofac und mehr

1. Unit Testing und TDD Basics

■ Unit Tests



■ Unit Tests sind F.I.R.S.T.

Fast

Independent

Repeatable

Self-validating

Timely

■ Bekannte Unit Testing Frameworks

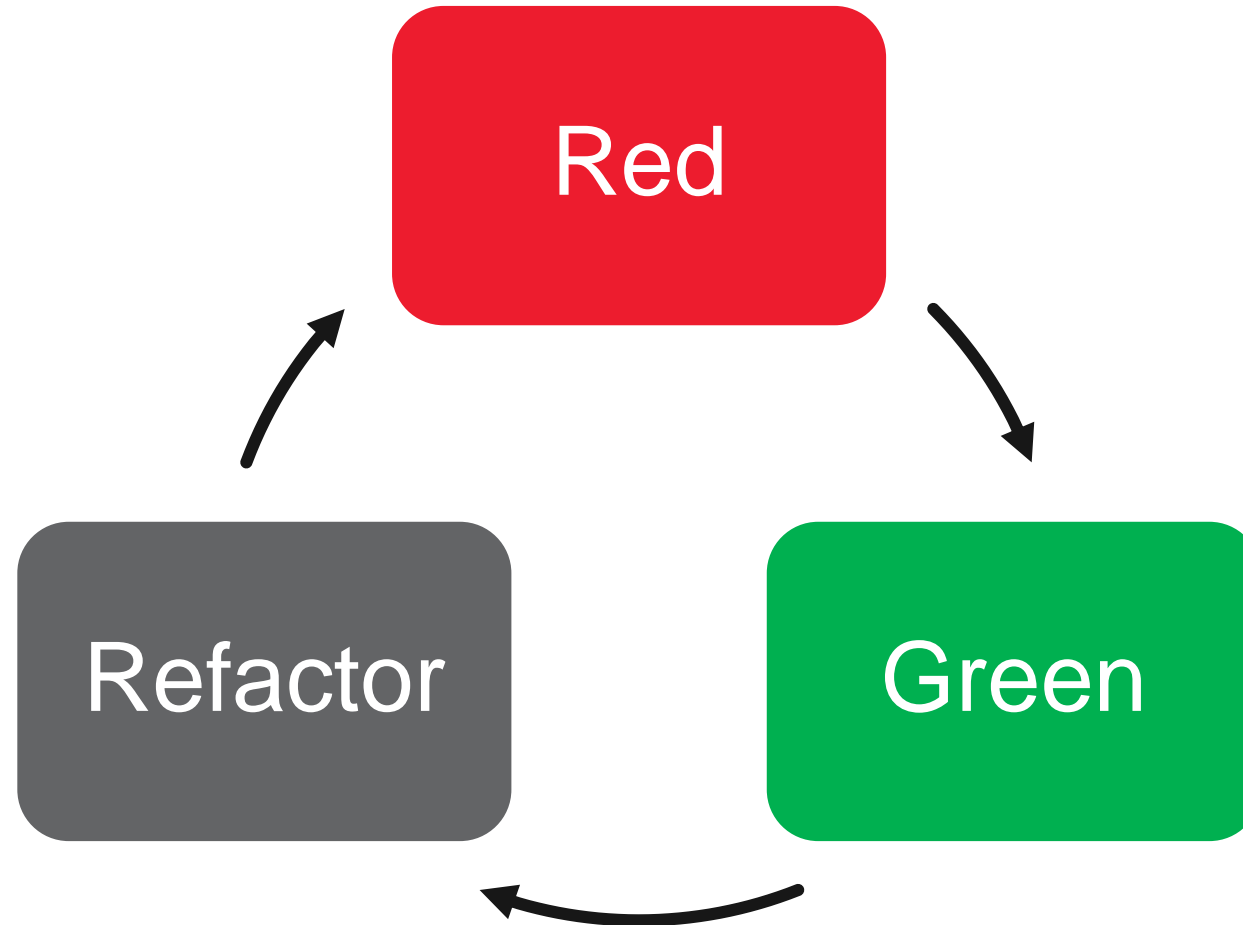
MSTest

NUnit

xUnit



■ Test Driven Development (TDD)



■ Vorteile von TDD

- Produktiver Code hat testbares Design
- Es muss intensiv über die Lösung nachgedacht werden
- Eigene Logik lässt sich fertigstellen, auch wenn noch Abhängigkeiten fehlen

■ «Spikes» in TDD

«Spikes» sind Experimente, die später wieder verworfen werden
(zumindest in der Theorie 😊)

■ Spike it!

Demo

Testbare ViewModels schreiben

■ Abhängigkeiten eines ViewModels

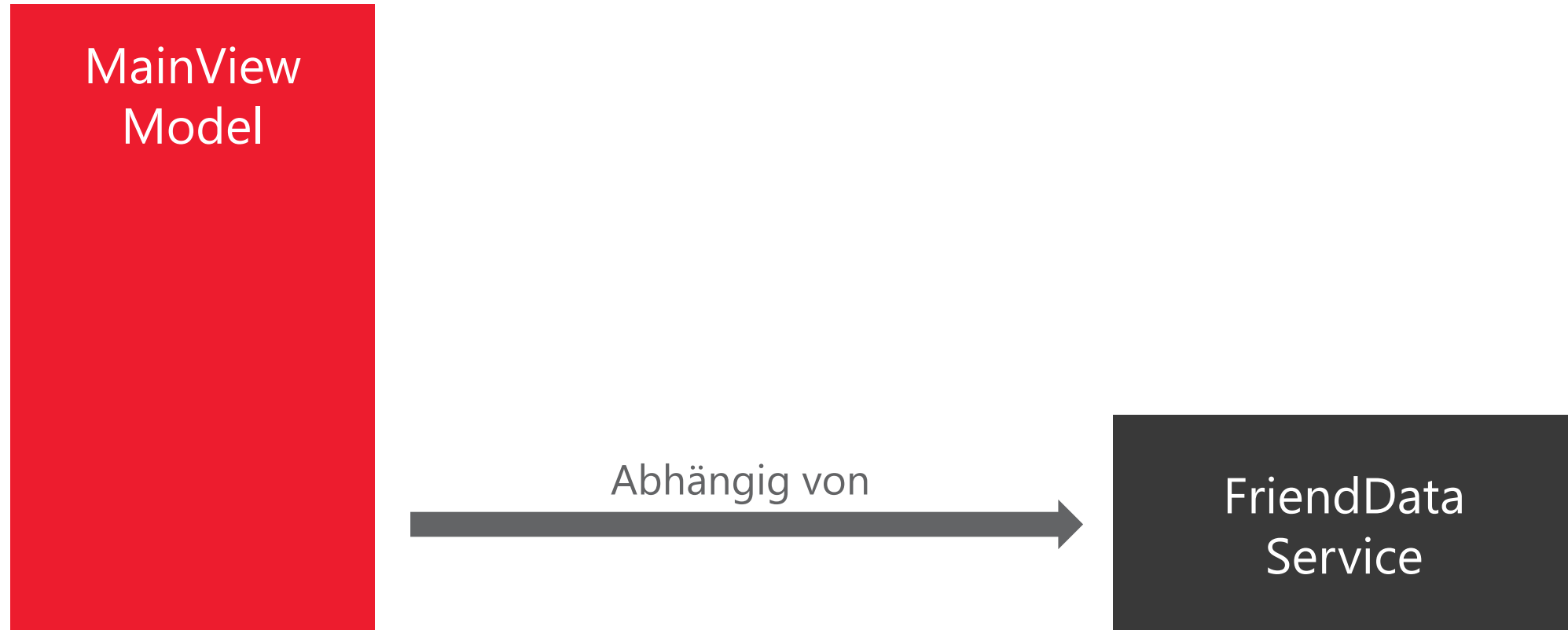
Data access

Dialoge

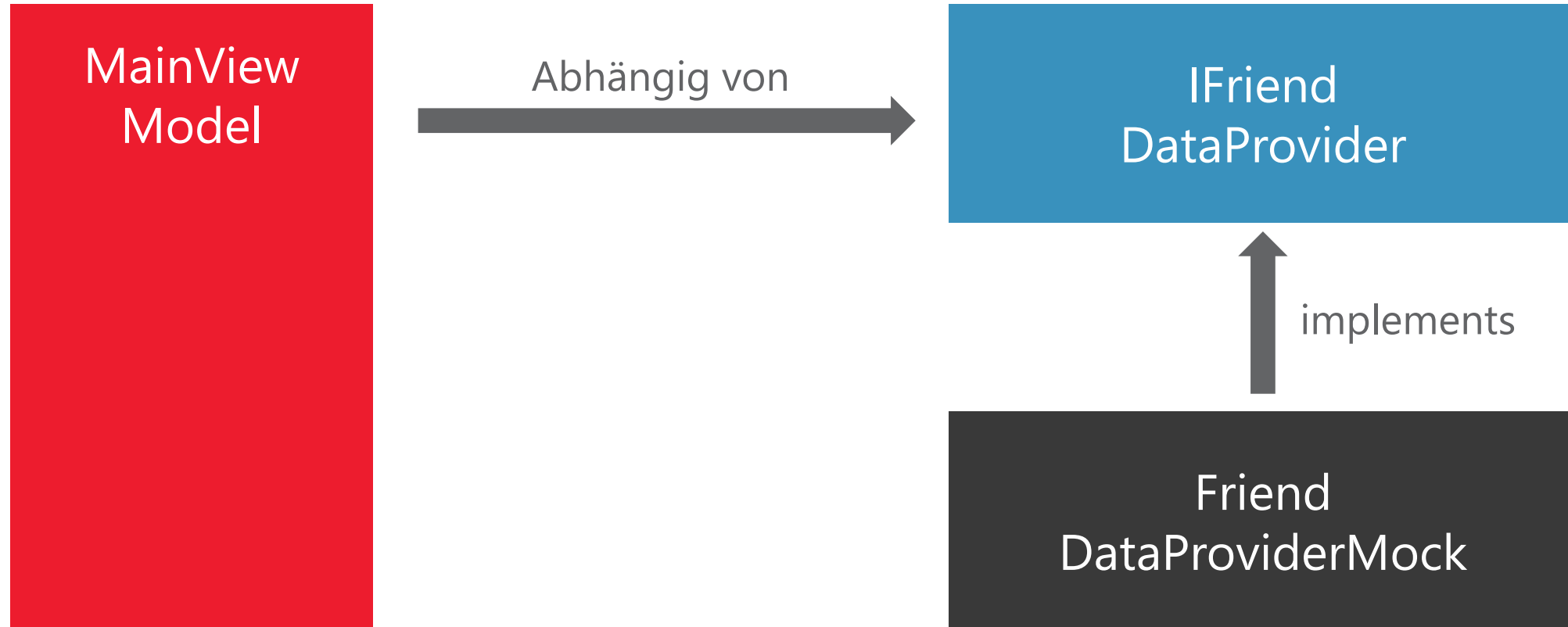
Event Aggregator

Andere ViewModels

■ Abhängigkeiten abstrahieren



■ Abhängigkeiten abstrahieren



■ Abhängigkeiten abstrahieren



Demo

Mock, Autofac und mehr

■ Abhängigkeiten mit Moq mocken

- Moq ist via NuGet verfügbar
- Funktioniert unter der Haube mit Dynamic Proxies



Demo

■ Die Anwendung zum Laufen bekommen



Demo

■ Dependencies mit Autofac injizieren



Demo

■ DeleteCommand implementieren



Demo

■ DeleteCommand mit MessageBox testen

- Erst ein Spike
- Dann ein Test
- Und schliesslich ein Interface 😊



Demo

Summary

■ Summary

- Abhängigkeiten aus ViewModel abstrahieren
- ViewModels lassen sich ohne konkrete Abhängigkeiten testen
- Code without tests is bad code

Fragen?

Thomas Claudius Huber

[@thomasclaudiush](https://twitter.com/thomasclaudiush)

thomas.huber@trivadis.com

www.thomasclaudiushuber.com

