

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Viewmodels auf dem Laufsteg

Model-View-Viewmodel (MVVM) mit Flow Oriented Design verbinden

Stefan Lieser

<http://lieser-online.de>

MVVM + Flow Design

Stefan Lieser

stefan@lieser-online.de

<http://clean-code-advisors.de>

Wat soll dat?

Leistung nutzen!

Silverlight

WPF

WP7

Agil!

WIP 1

Täglich liefern

Getestet!

Natürlich automatisiert...

Bewährtes verwenden!

MVVM

Flow Design

MVVM

Model

View

ViewModel

MVVM

Leistungsfähigkeit der Frameworks wächst
ständig

DataBinding in XAML

Touch

MVVM

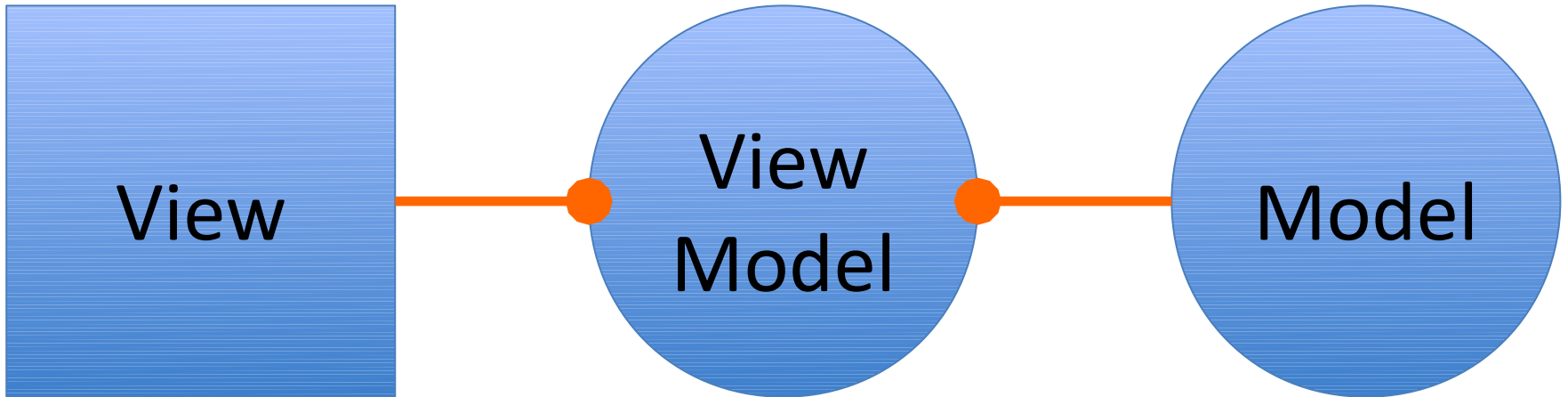
Anwendung muss automatisiert testbar sein

Trennung der Belange

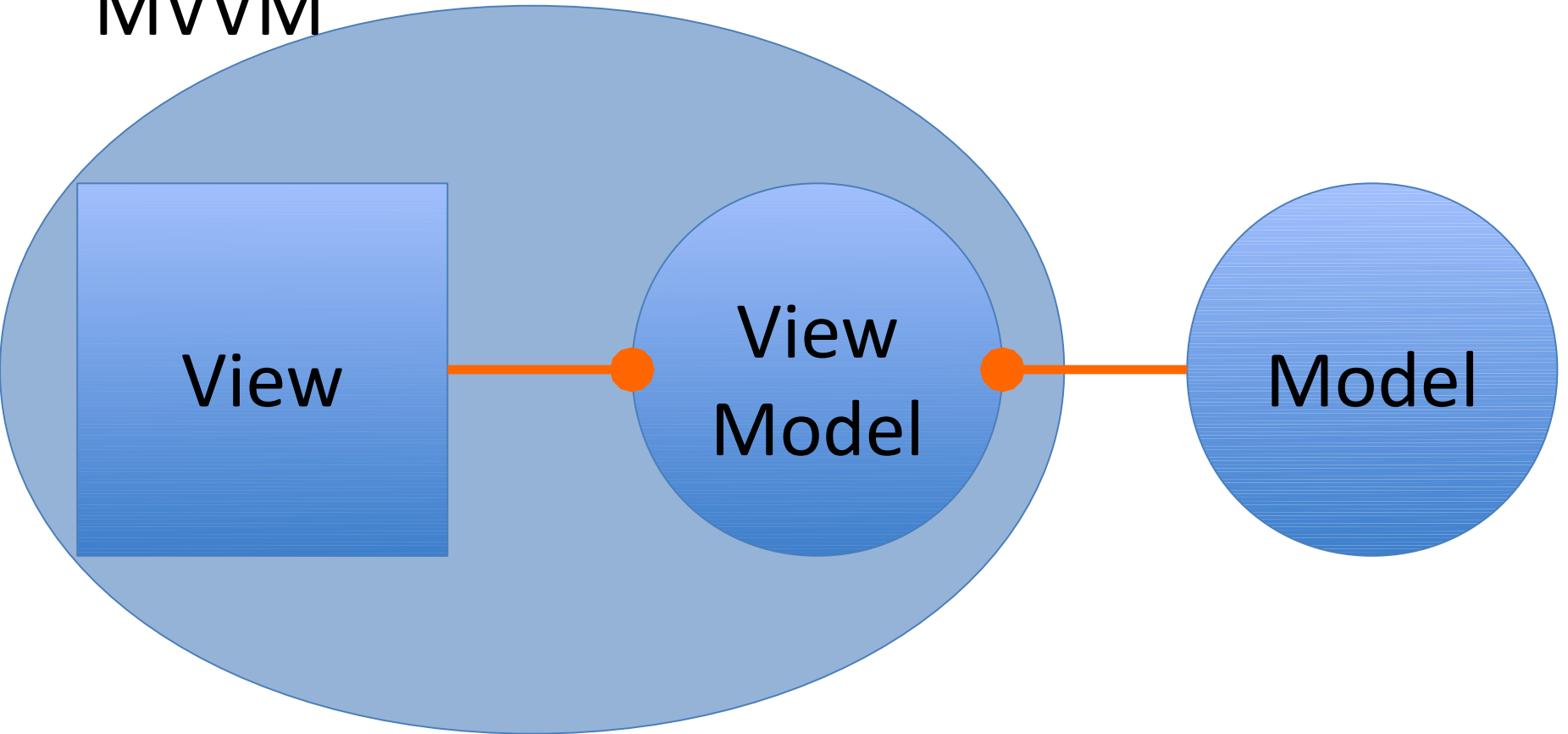
MVVM

Zentral ist das ViewModel

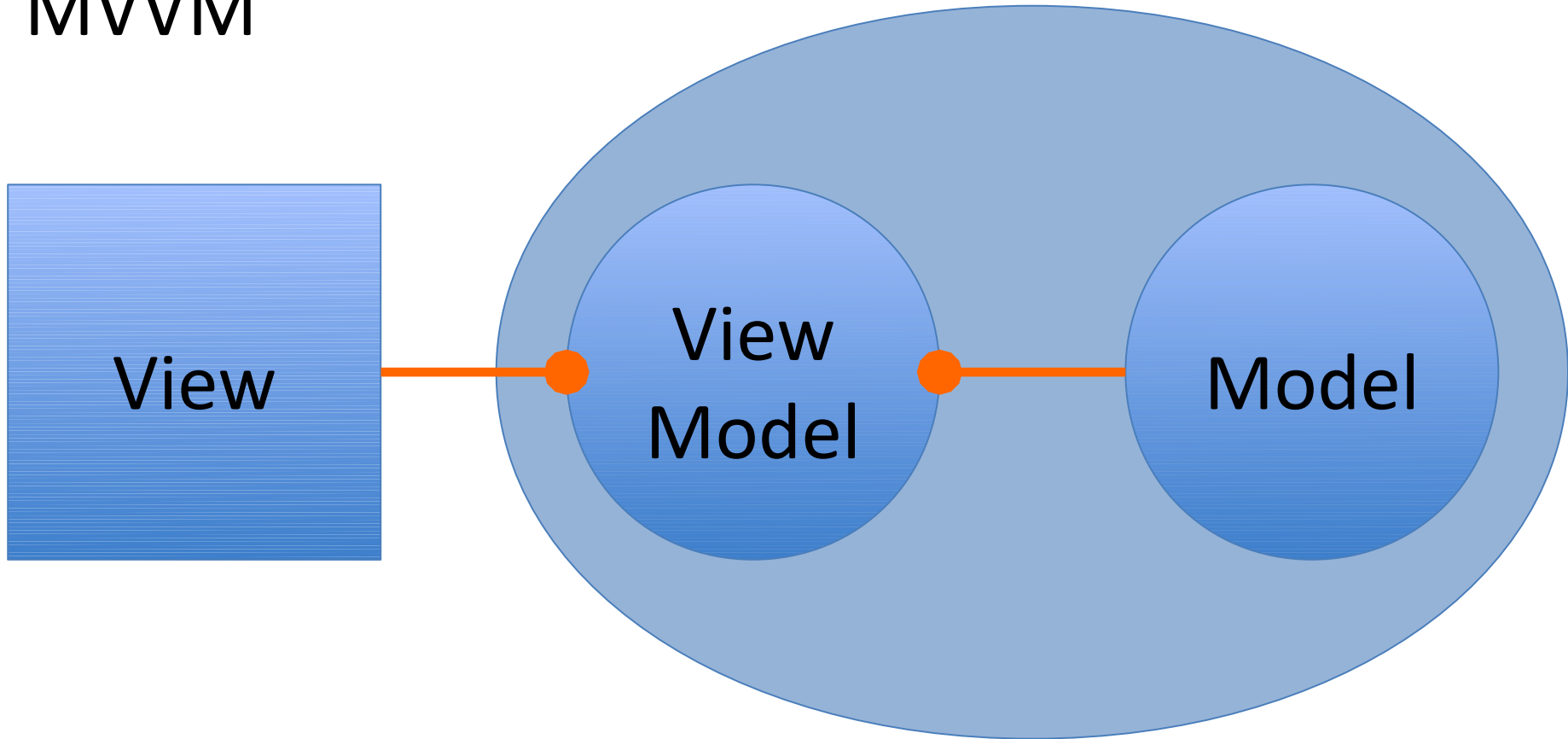
MVVM



MVVM



MVVM



MVVM

Views deuten nicht!

MVVM Views deuten nicht!

```
<TextBox  
    IsEnabled="{Binding Path=KontonummerEnabled}"  
    Text="{Binding Path=Kontonummer}"  
/>
```

MVVM

MVVM ist ein Pattern!

MVVM ist keine Architektur!

MVVM

Was tun mit dem **Model**?

MVVM - Model

Wie diskutiert man im Team darüber?

MVVM - Model

Objektorientiert? Aber wie?

Domain Driven Design

CQRS

Flow Design

Flow Design

Im Modell spielt die Musik!

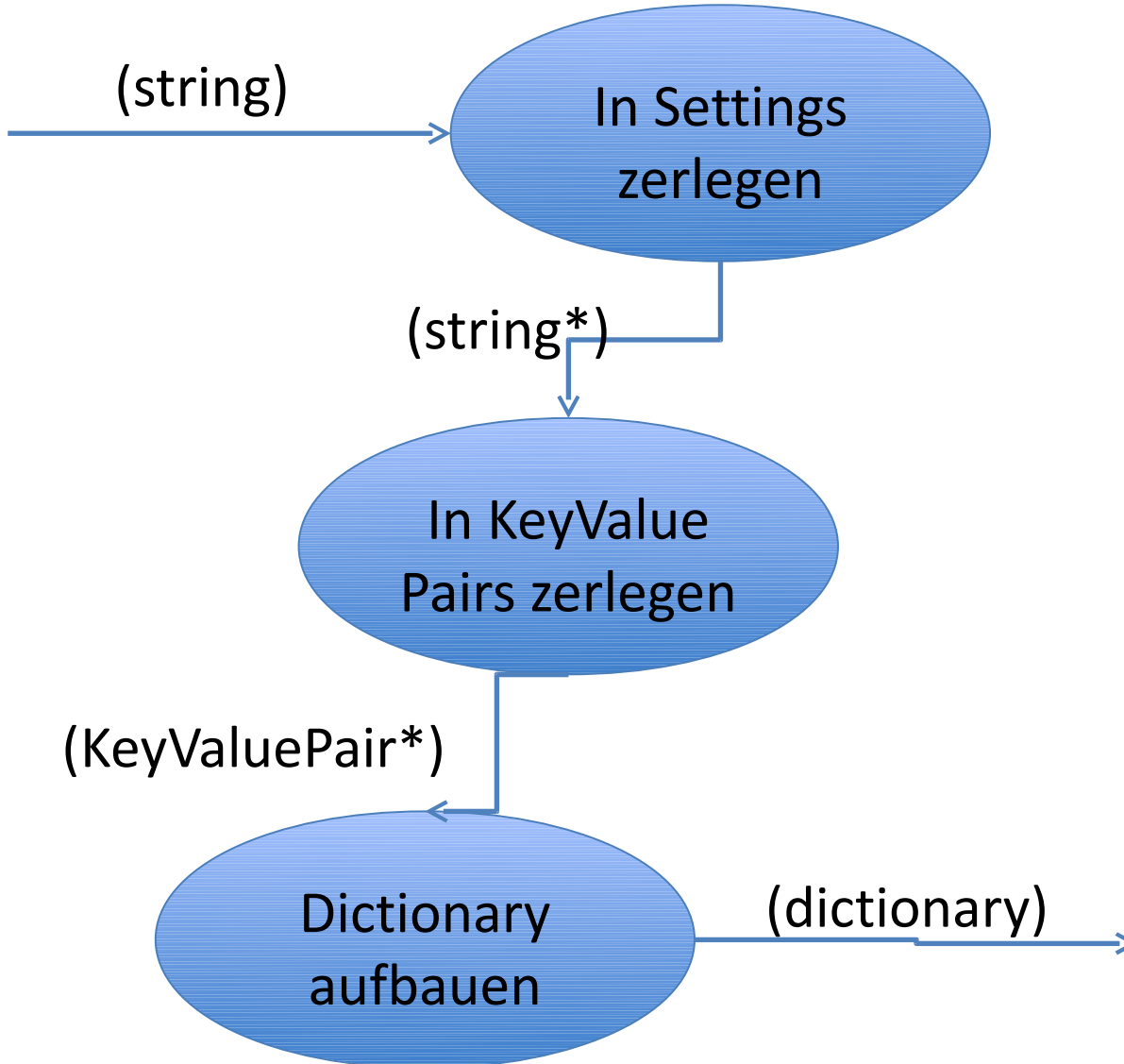
Flow Design - Beispiel

ToDictionary

“a=1;b=2;c=3”

{{“a”, “1”},
{“b”, “2”},
{“c”, “3”}}

"a=1;b=2;c=3"



{{"a", "1"},
{"b", "2"},
{"c", "3"}}}

Kennen Sie Eva?

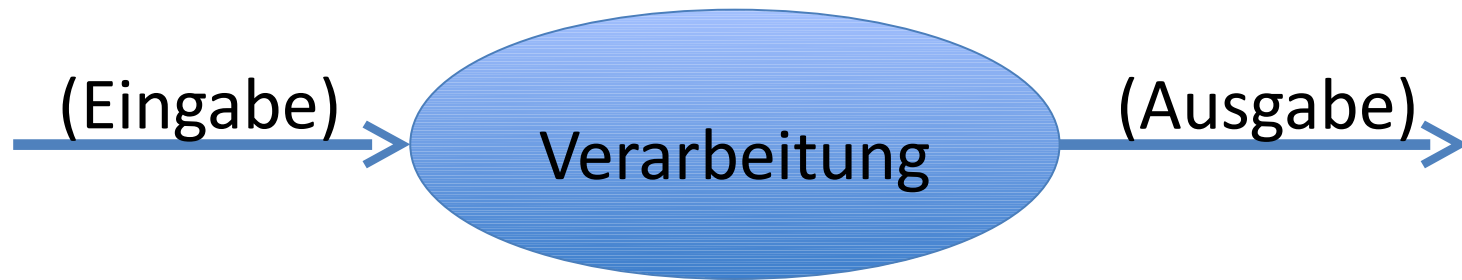
Kennen Sie Eva?

Eingabe

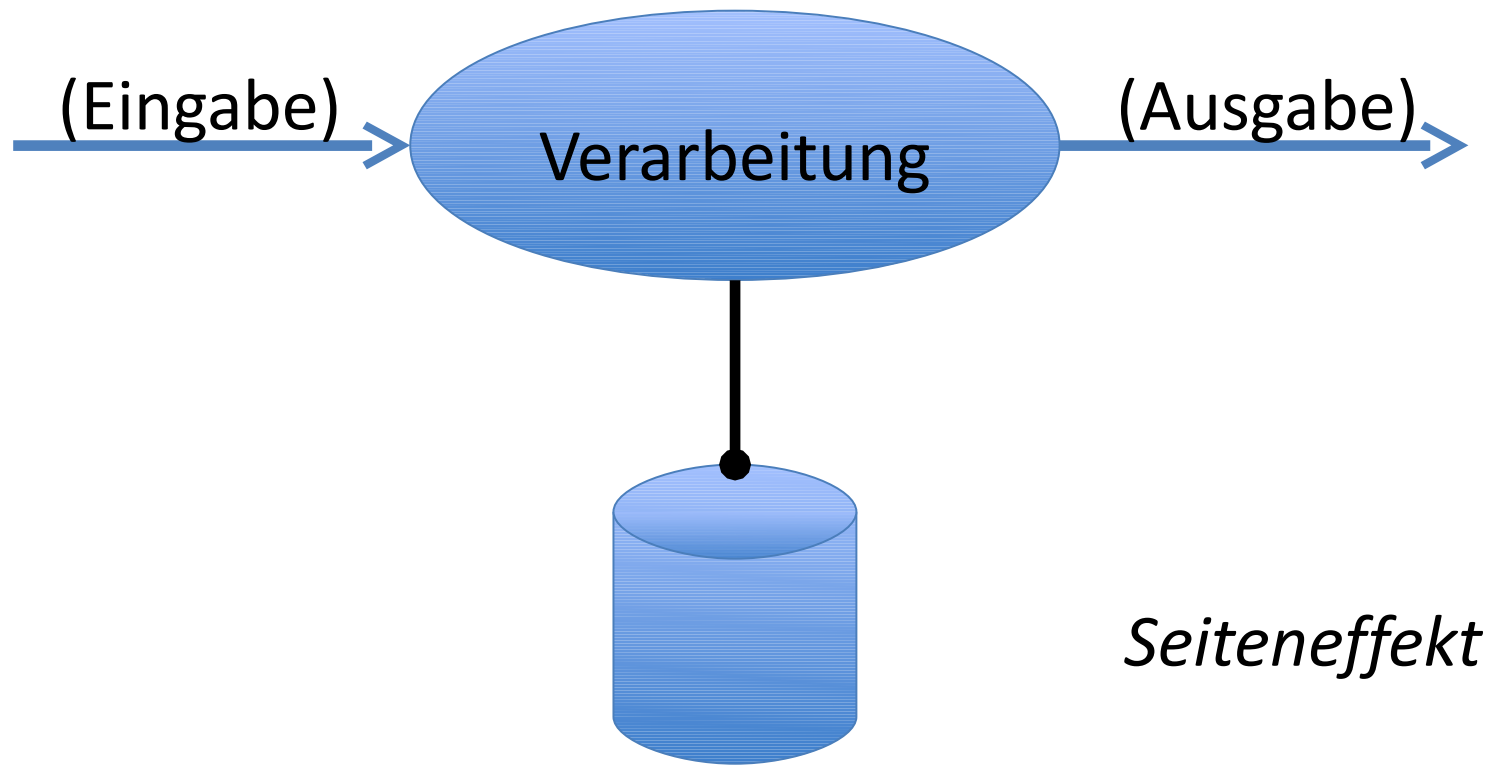
Verarbeitung

Ausgabe

Eingabe/VVerarbeitung/Ausgabe



Eingabe/VVerarbeitung/Ausgabe



Syntax



Datenfluss



Abhängigkeit

Syntax



Datenfluss

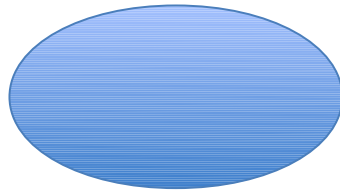


Abhängigkeit

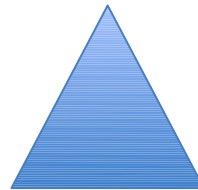
Funktionseinheiten:



Portal



Logik



Adapter

Syntax



Datenfluss

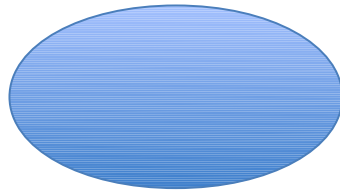


Abhängigkeit

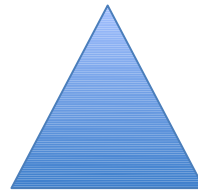
Funktionseinheiten:



Portal



Logik

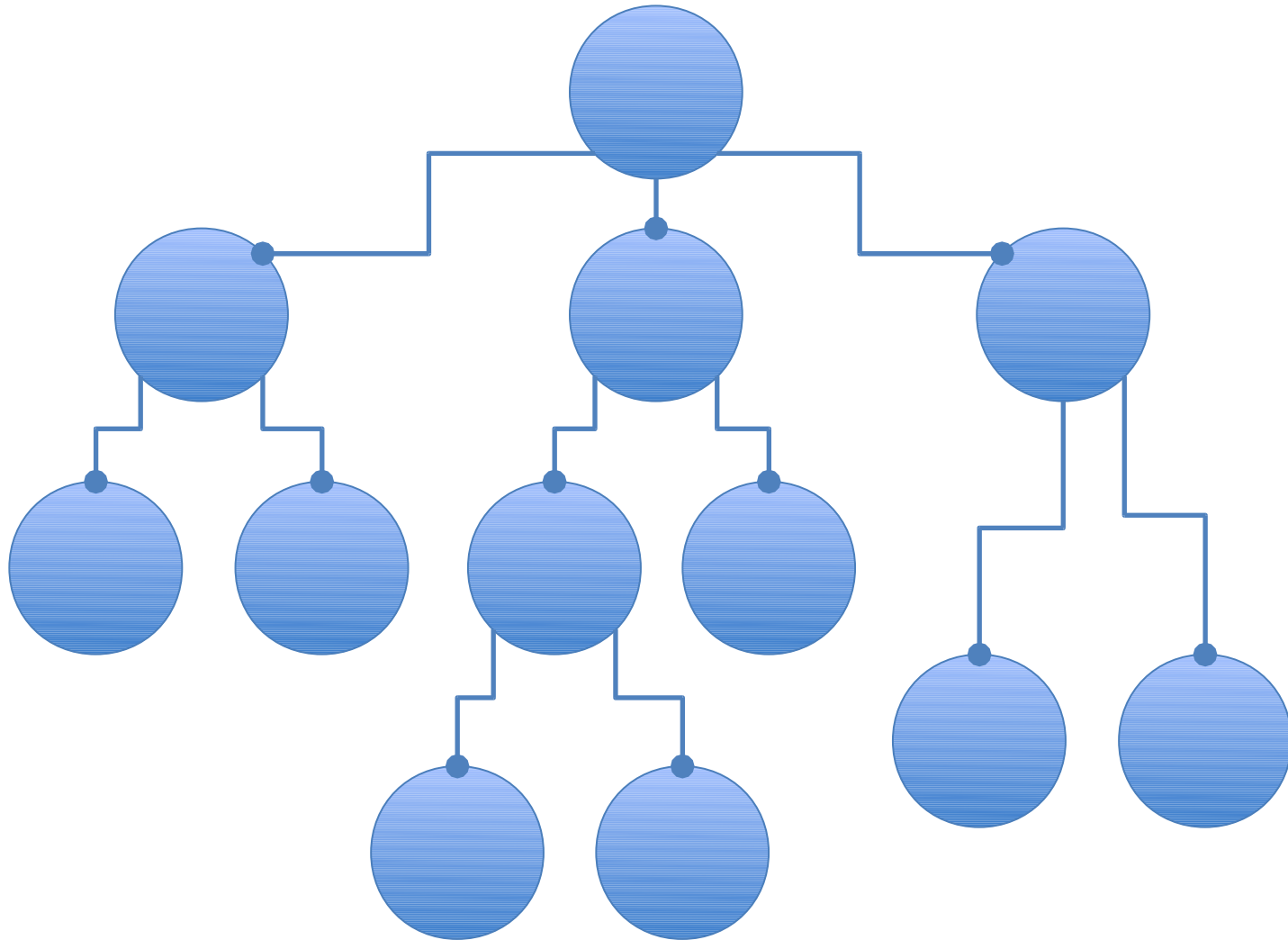


Adapter

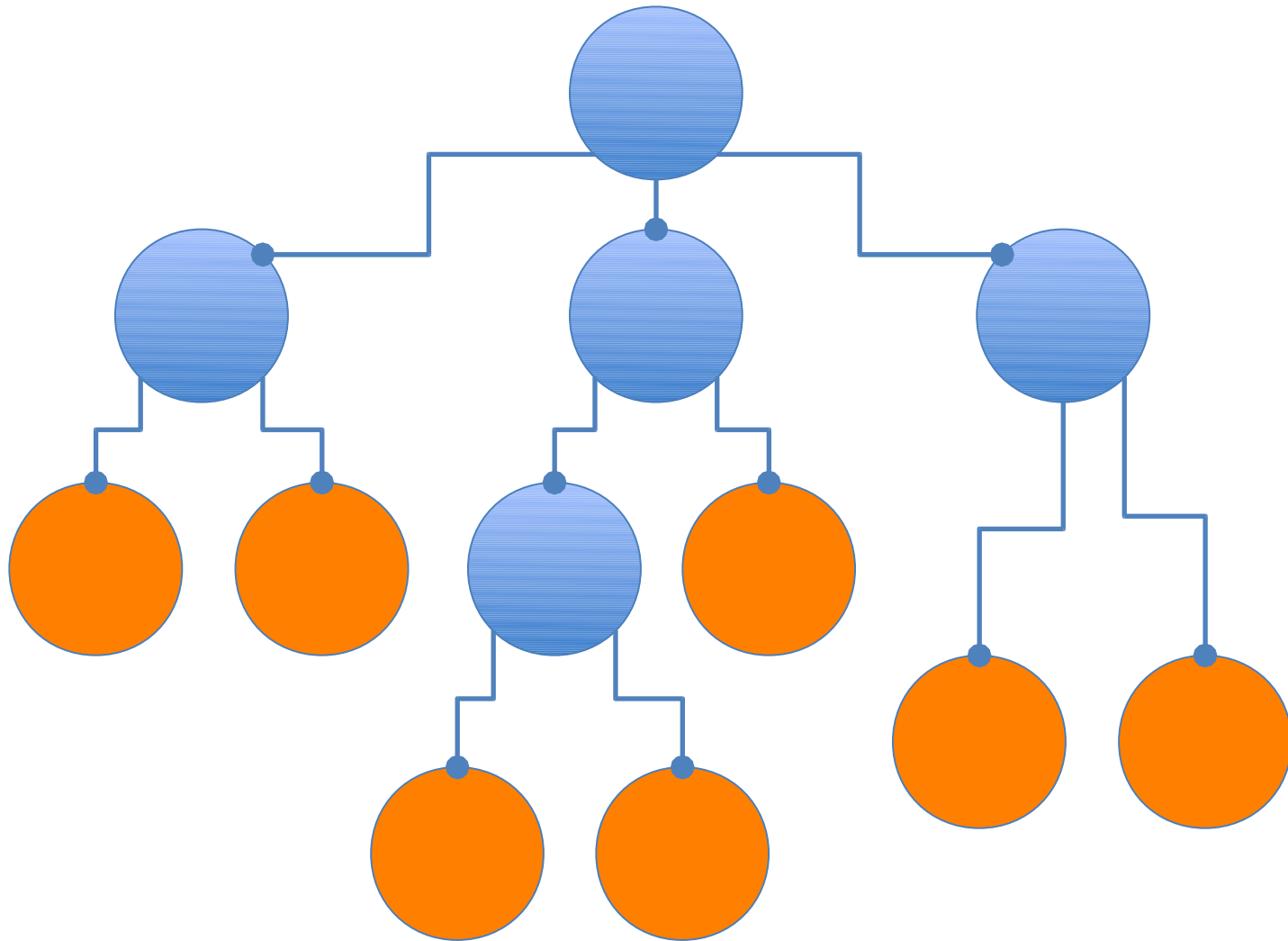


Ressource

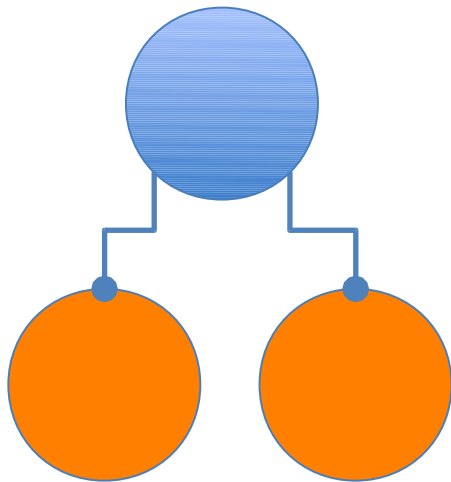
Separation of Concerns



Separation of Concerns



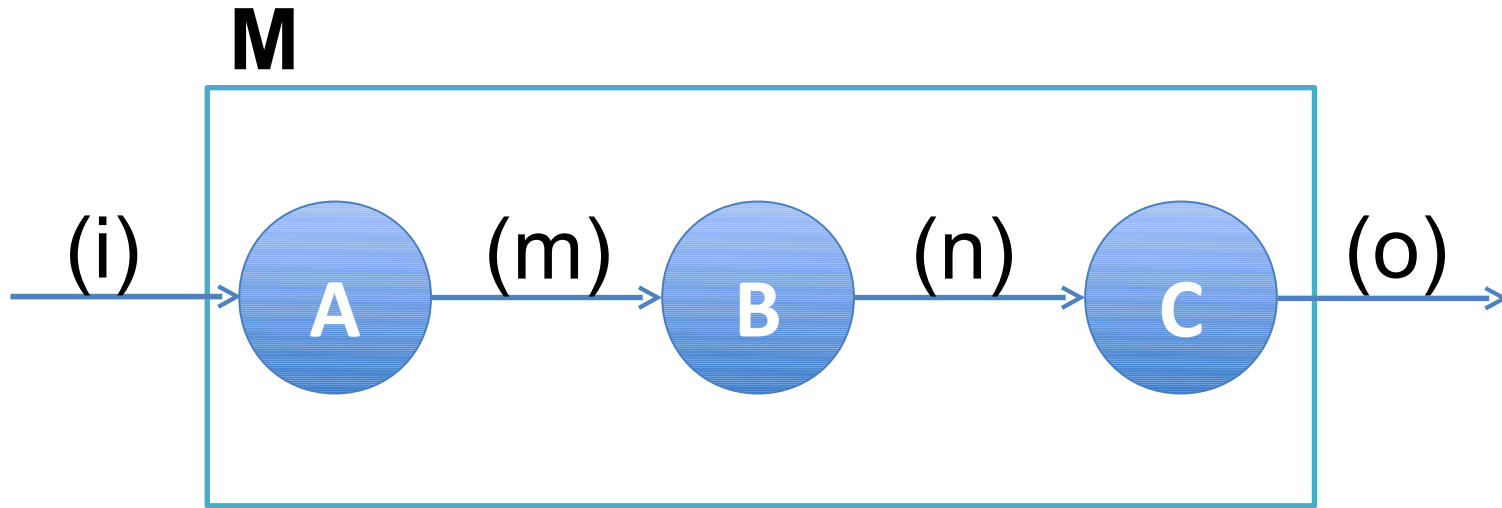
Separation of Concerns



Abhängigkeiten

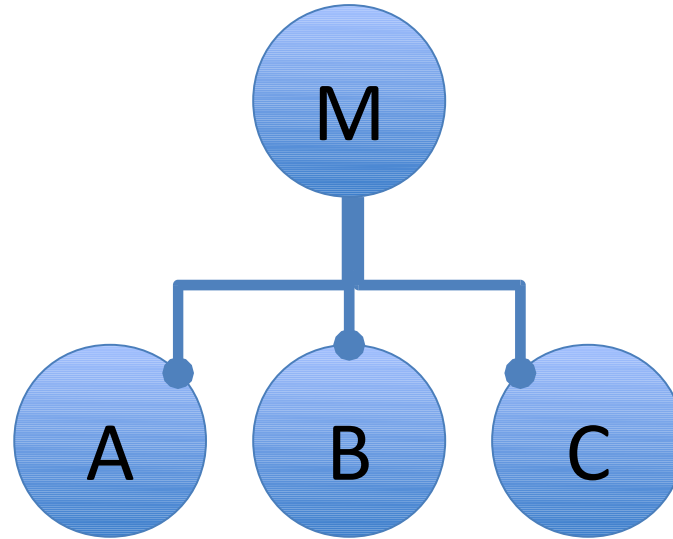
Logik

Beispiel



Datenflußdiagramm

Beispiel

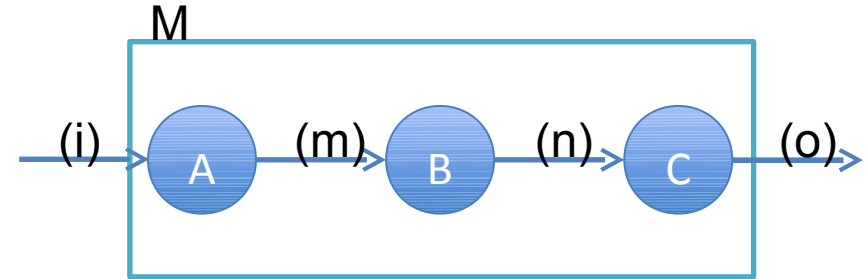


Abhängigkeitsdiagramm

```

public class ClassForM
{
    public To M(Ti i) {
        var m = A(i);
        var n = B(m);
        var o = C(n);
        return o;
    }

```

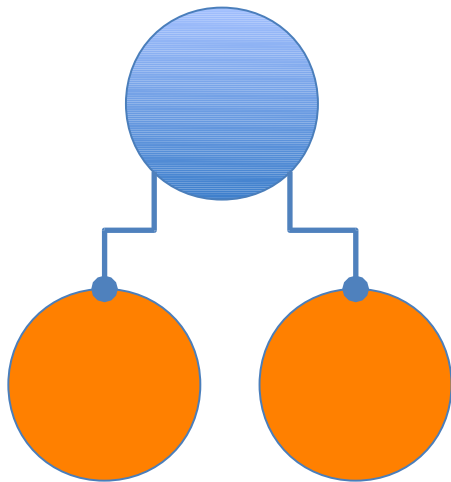


```

private Tm A(Ti i) { ... }
private Tn B(Tm m) { ... }
private To C(Tn n) { ... }
}

```

Event Based Components (EBC)



Platine (Board) - Abhängigkeiten

Bauteil (Part) - Logik

Überbegriff: *Funktionseinheit*

```
public interface IA
```

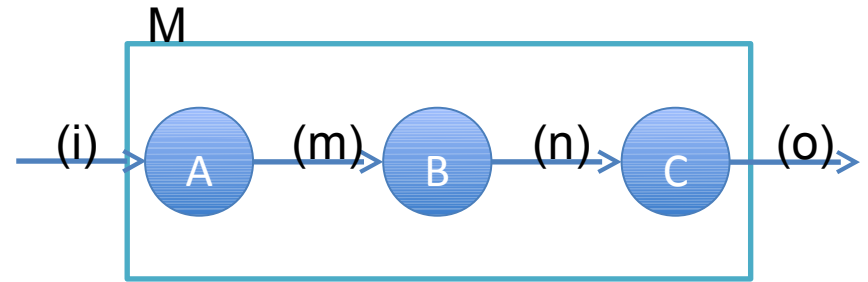
```
{
```

```
    event Action<Tm> Result;
```

```
    void Process(Ti message);
```

```
}
```

```
// interface IB, IC ...
```



```
public interface IM
```

```
{
```

```
    event Action<To> Result;
```

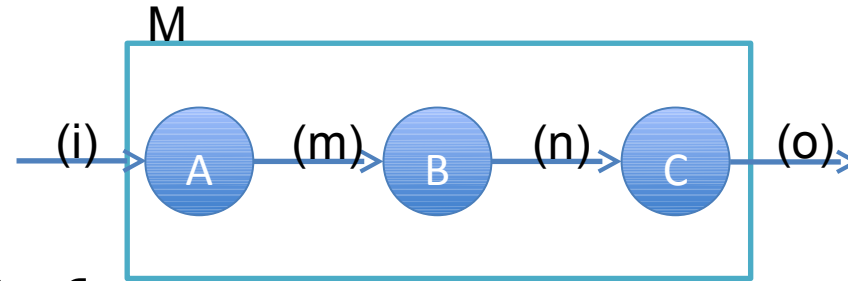
```
    void Process(Ti message);
```

```
}
```

```
public class M : IM
{
```

```
    public event Action<To> Result;
    private readonly Action<Ti> _Process;
```

```
    public void Process(Ti message) {
        _Process(message);
    }
```



```
    public M(IA A, IB B, IC C) {
        _Process += A.Process;
        A.Result += B.Process;
        B.Result += C.Process;
        C.Result += m => Result(m);
    }
```

```
}
```

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<board name="M">
```

```
<using namespace="beispiel.datacontracts" />
```

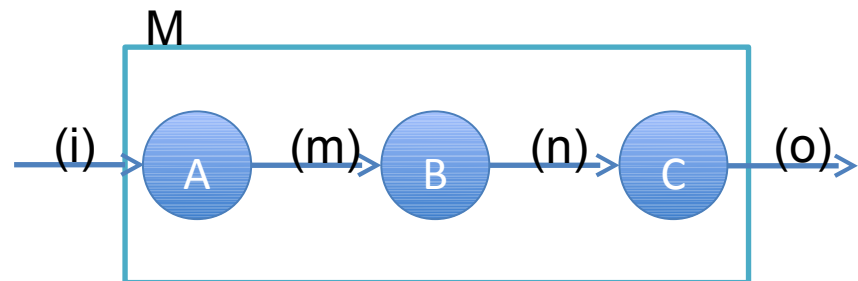
```
<wire from="this" type="Ti" to="A" />
```

```
<wire from="A" type="Tm" to="B" />
```

```
<wire from="B" type="Tn" to="C" />
```

```
<wire from="C" type="To" to="this" />
```

```
</board>
```



Flow Design - Evolvierbarkeit

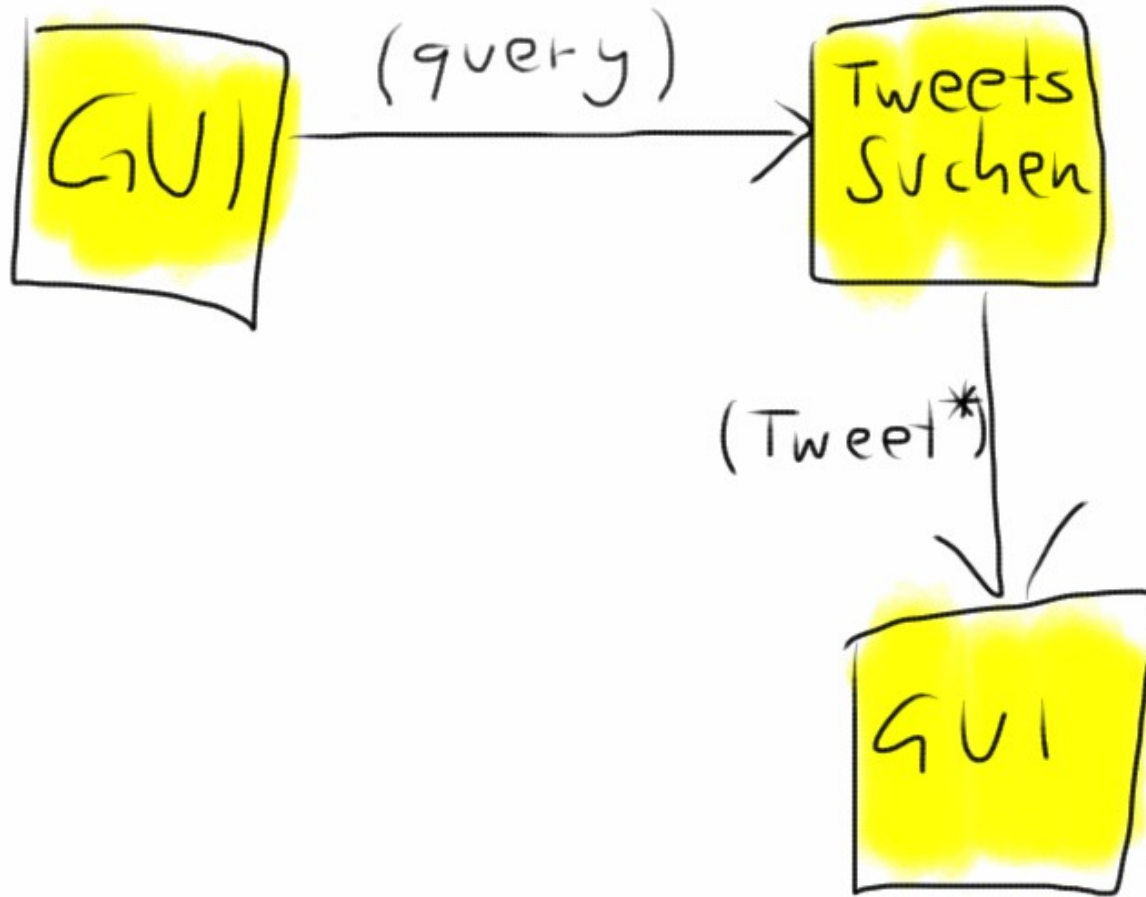
Kleinschrittig

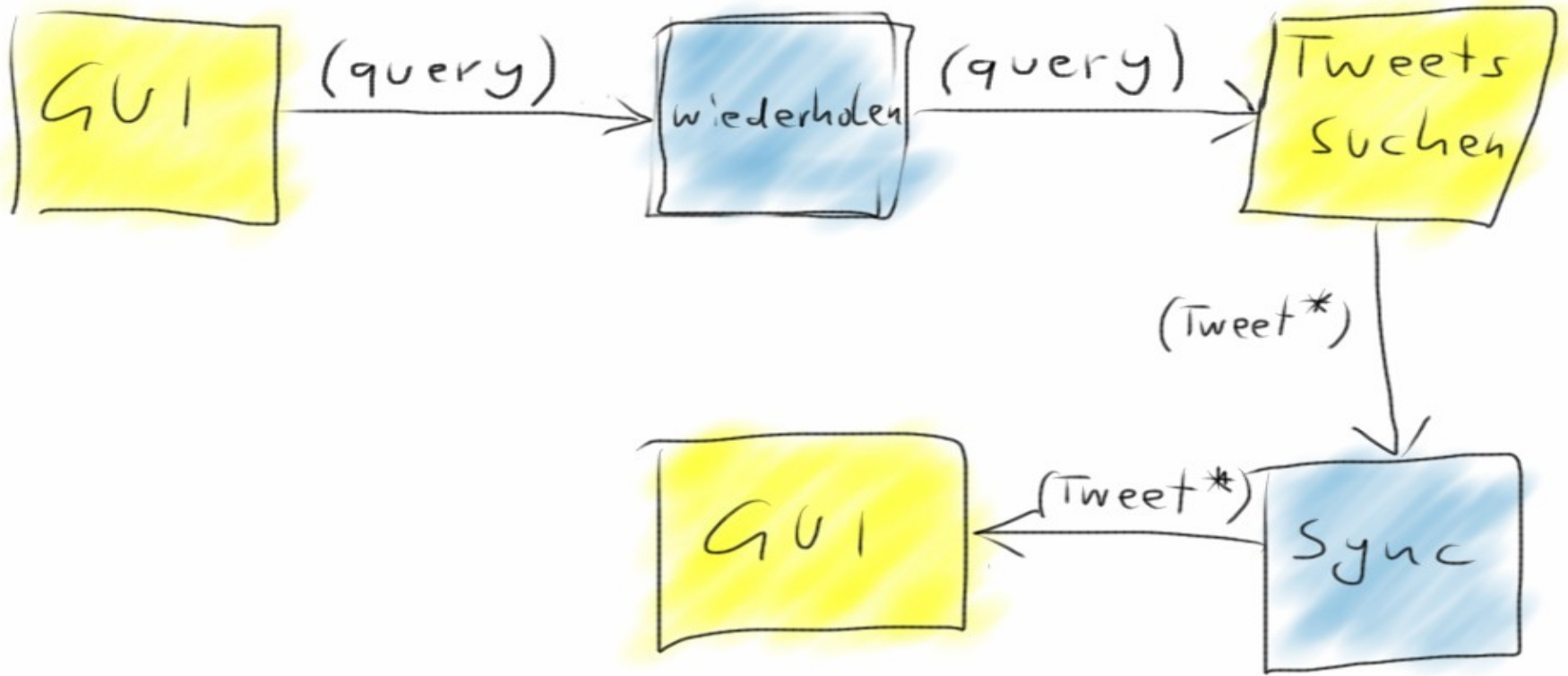
Per Feature

Täglich

Flow Design - Evolvierbarkeit

Beispiel: TwitterBand





Flow Design - Sichtweise

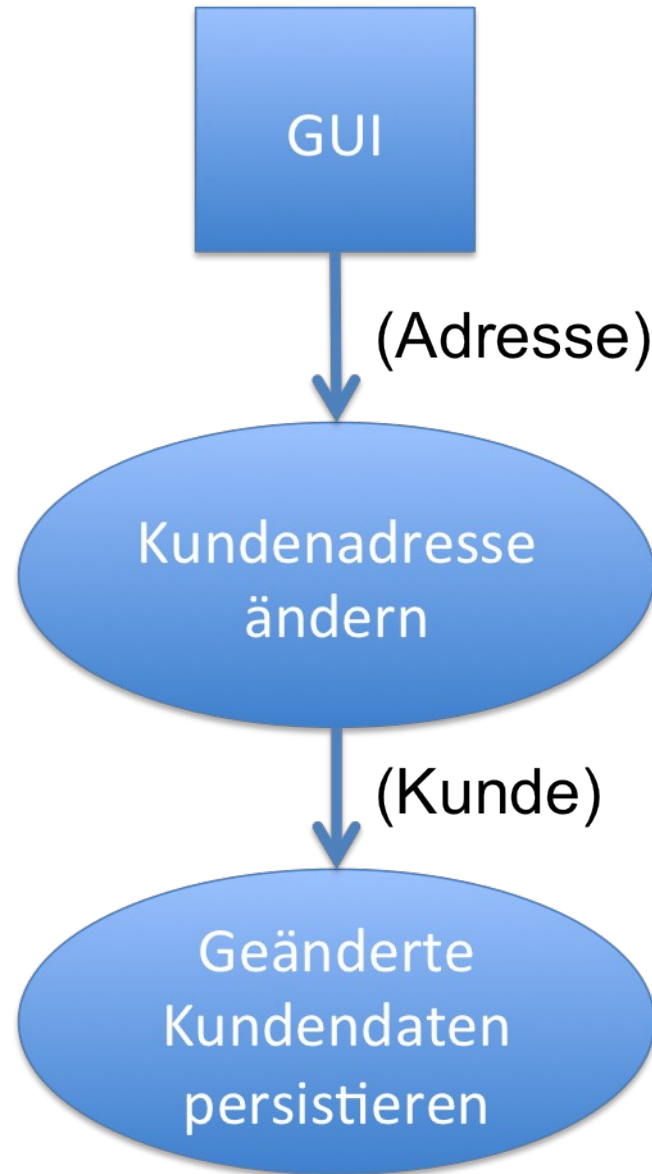
Modellieren mit
Aktivitäten
statt
Akteuren

Flow Design - Sichtweise

GUI, Ressourcen und Standardbausteine (Map, Join, etc.) sind häufig Akteure.

Flow Design - Benutzerschnittstelle

UI im Flow



MVVM

+

Flow Design

Flow Design - Benutzerschnittstelle

Und wie passt da jetzt MVVM rein?

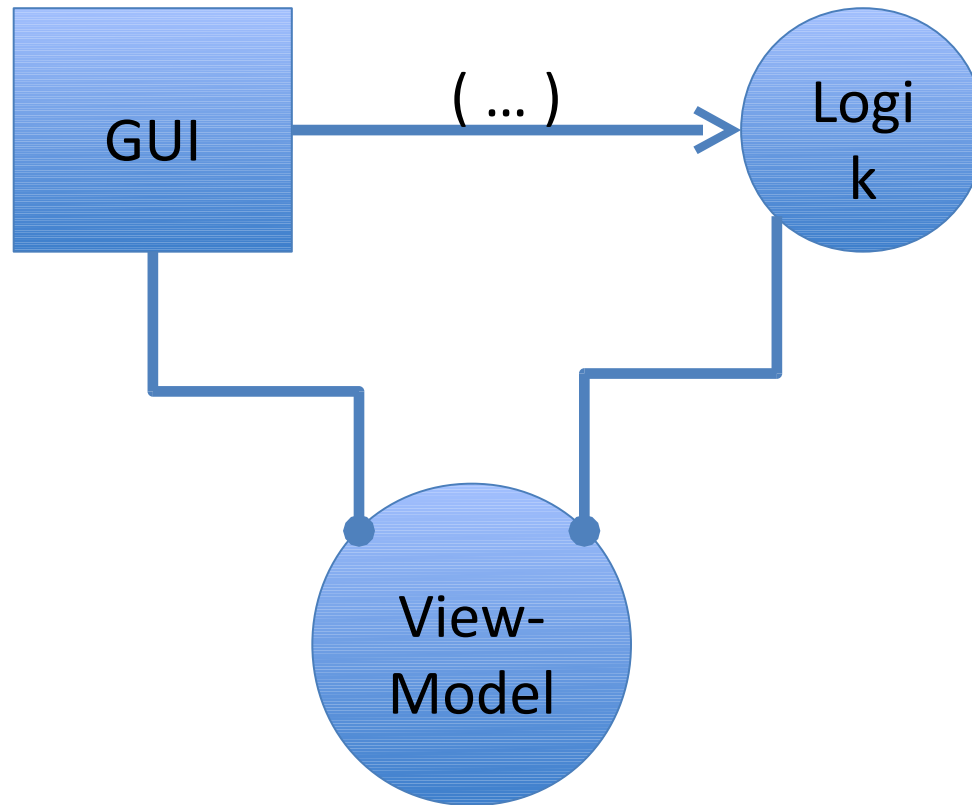
MVVM + Flow Design

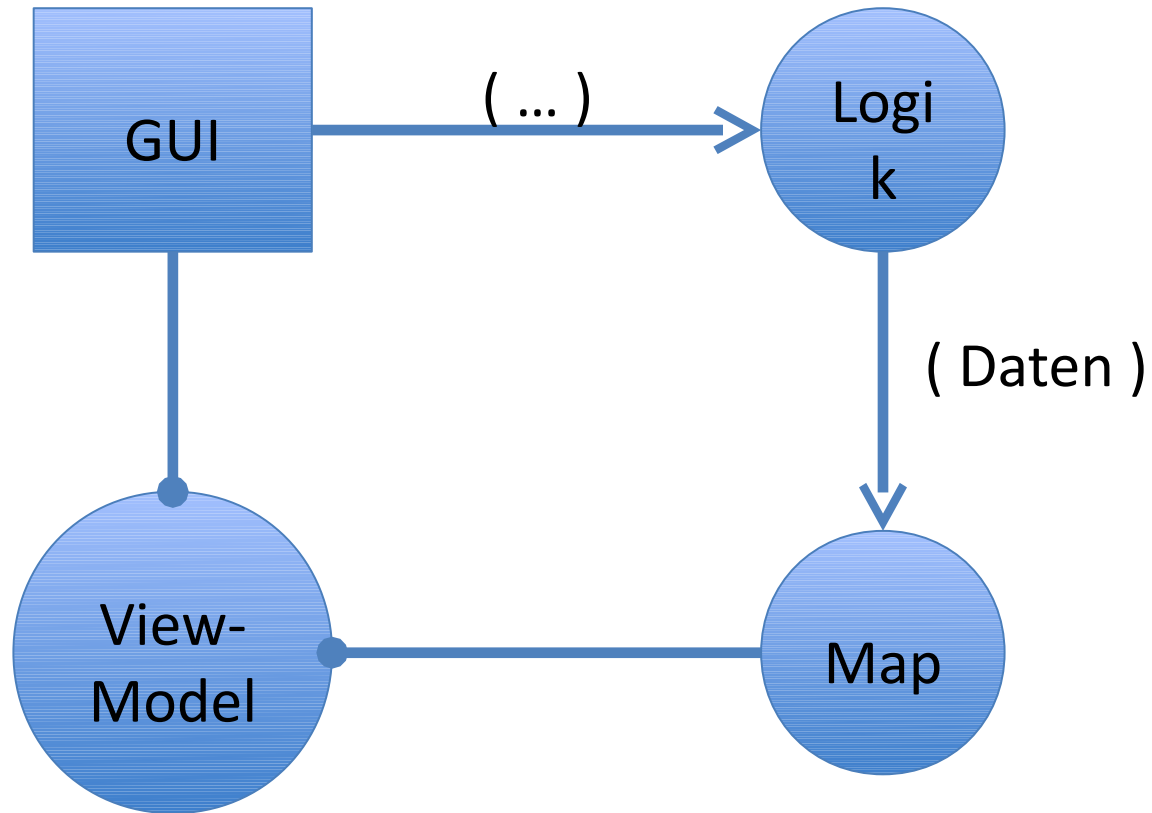
View möchte ein ViewModel

MVVM + Flow Design

Wie greifen Flows auf das ViewModel zu?







MVVM + Flow Design

Build

Bind

Inject

Config

Run

MVVM + Flow Design

Commands

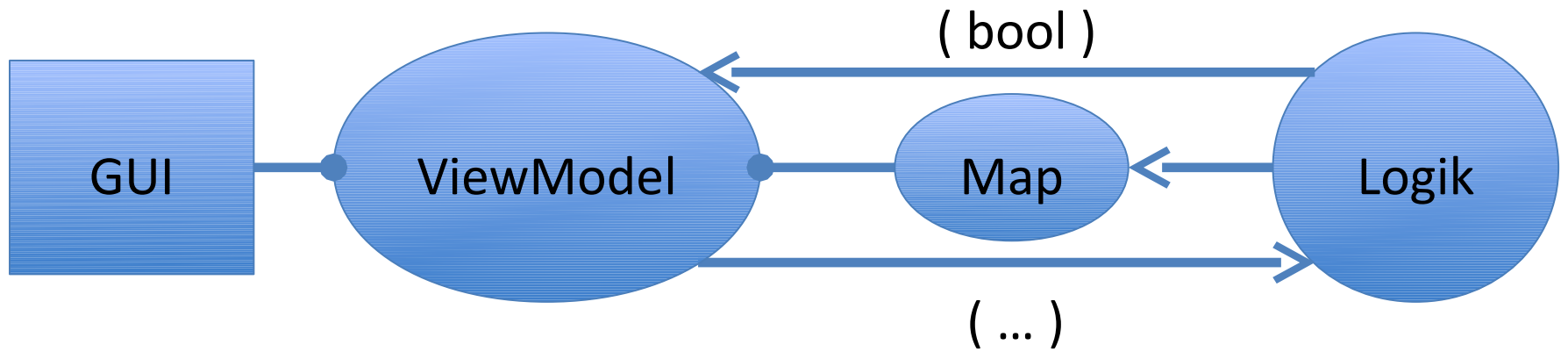
Execute

CanExecute

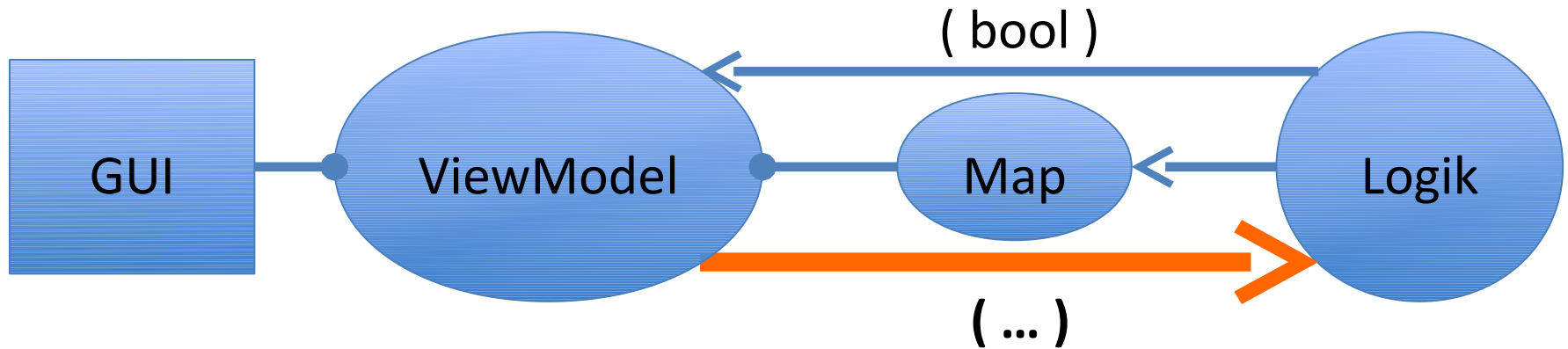
MVVM + Flow Design

FlowCommand

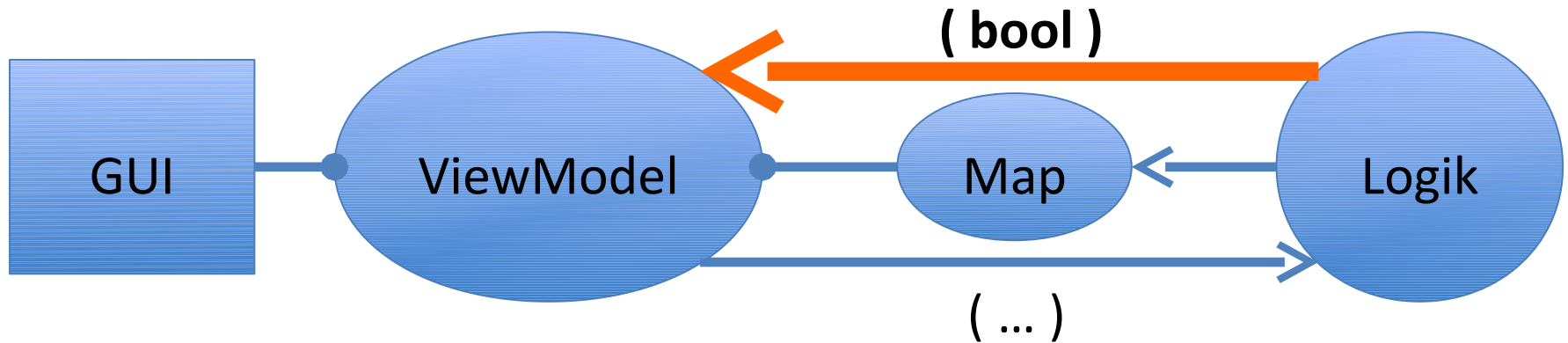
<http://ebclang.codeplex.com>



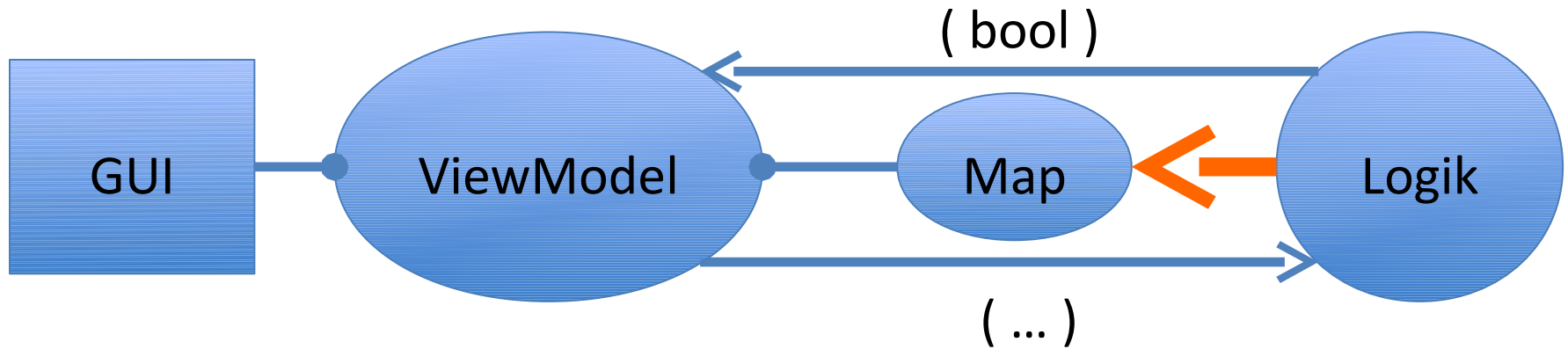
Command Execute



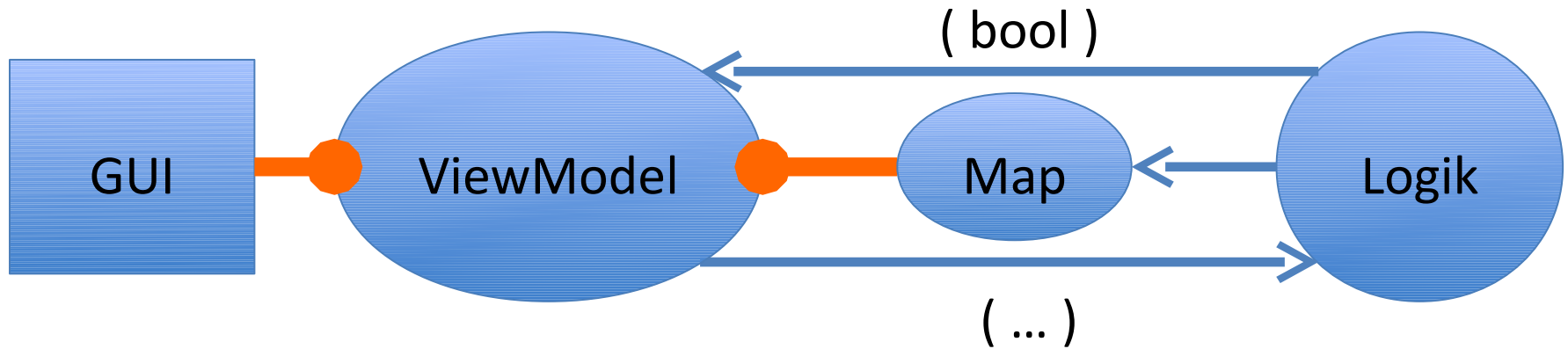
Command SetCanExecute



ViewModel Updates



Inject Phase



MVVM + Flow Design

Show me your Code!