

# *Ribbons und Kopplung & Kohäsion*

Martin Bühlmann  
Bruno Juchli

# Über uns

- EROWA AG
  - Spannsysteme
  - Roboter
  - Prozessleitsysteme
  - 240 Mitarbeiter CH, 450 weltweit
- Prozessleitsystem EMC3
  - Automatisierung der Fertigung
  - Optimale Ressourcennutzung
  - Verantwortung dem Anwender abnehmen
  - C#.Net, Ninject, NHibernate, WCF, xUnit, Mspec, VS, R#, dotCover, Teamcity,...
  - UI: WPF/XAML, Caliburn.Micro, Ribbons

[www.erowa.com](http://www.erowa.com)



Home

Storage



Unload Pallet  
from Storage



Load Pallet  
to Storage

Storage Handling



Retrieve Pallet from  
Loading Place



Send Pallet to  
Unloading Place

Loading Station Handling



Create  
Workpieces



Renew  
Workpieces



Execute Manual  
Operation



Confirm and repeat  
process step



Confirm and  
continue

Process

EROWA<sup>®</sup>  
system solutions

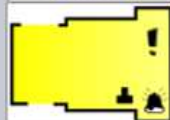


Main

<<



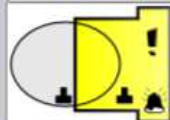
Mill A 3 axis



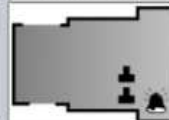
Mill A 5 axis



Mill B 3 axis



Mill B 5 axis



Load. UPC



Mag. Its148



Mag. Mix



Overview

1															
2															
3															
4															
5															



Name



Pallet



Main



Setup



Configuration



RFID Reader



19:40




EMC Server

# Ziel

- Aufzeigen wie man ein komplexes Ribbon mit hoher SW Qualität realisieren kann
- Wichtigkeit von Separation of Concern & Single Responsibility exemplarisch aufzeigen
- Spass!



# Inhalt

- Ribbon Begriffe
- Anforderungen an Ribbons
- Wo liegt das Problem?
- 
- Konzeptionelle Lösung
- Was hat es gebracht?
- Implementation
- Fragen / Diskussion



# Fragen

- Verständnisfragen: SOFORT
- Ideen, Alternativen,...: Am Schluss



# Ribbon Begriffsdefinition

# Ribbon

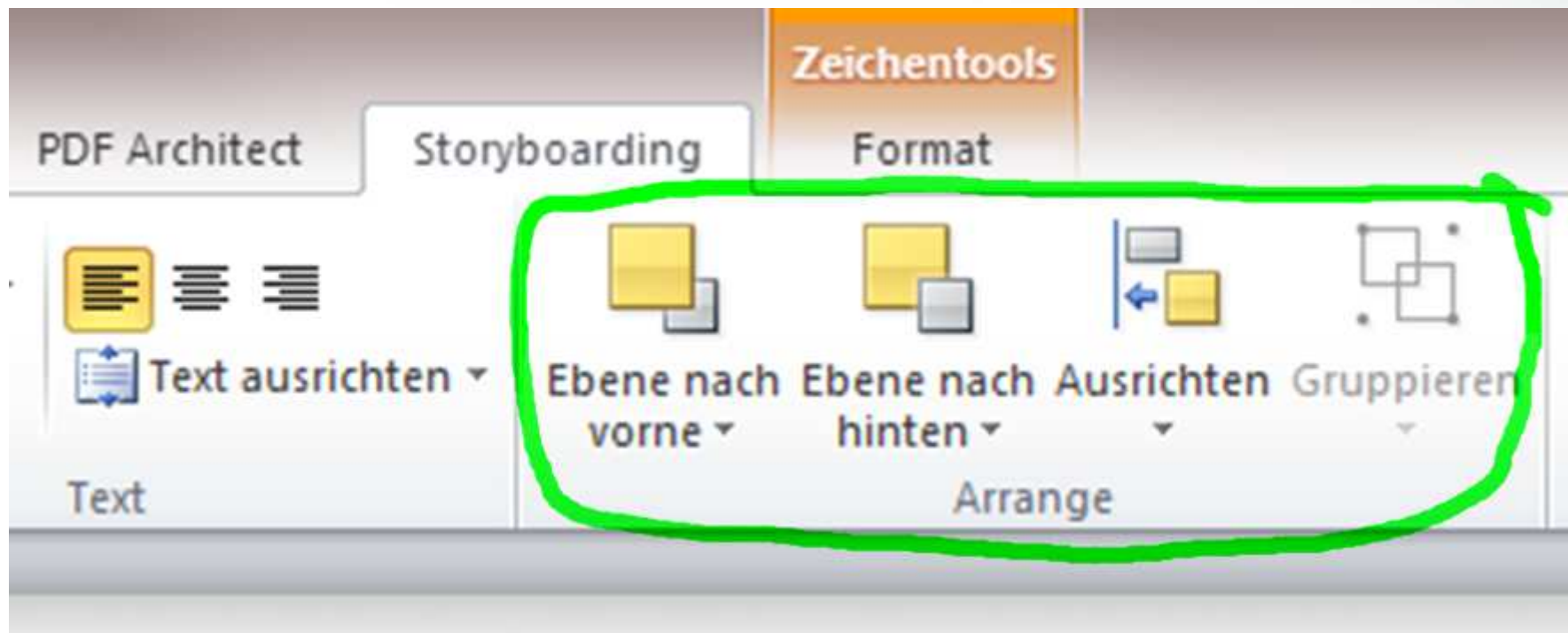




# Ribbon Tab



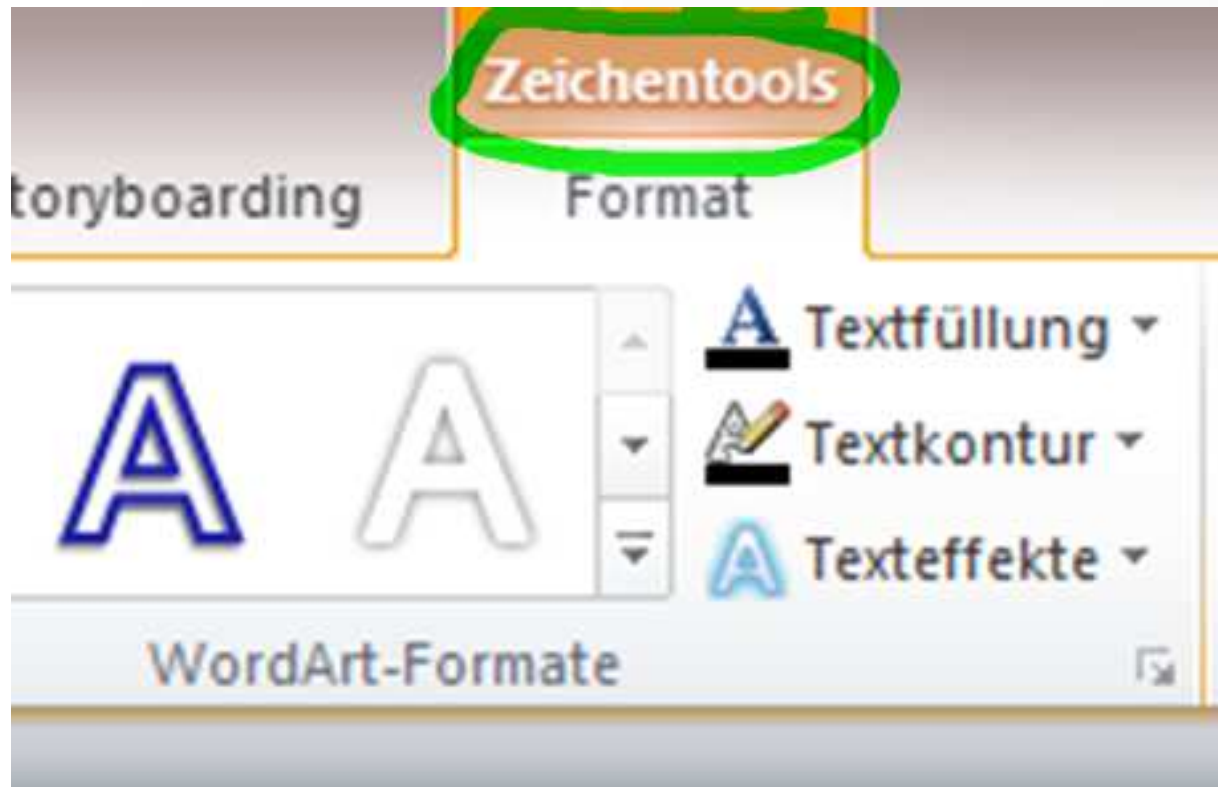
# Ribbon Group



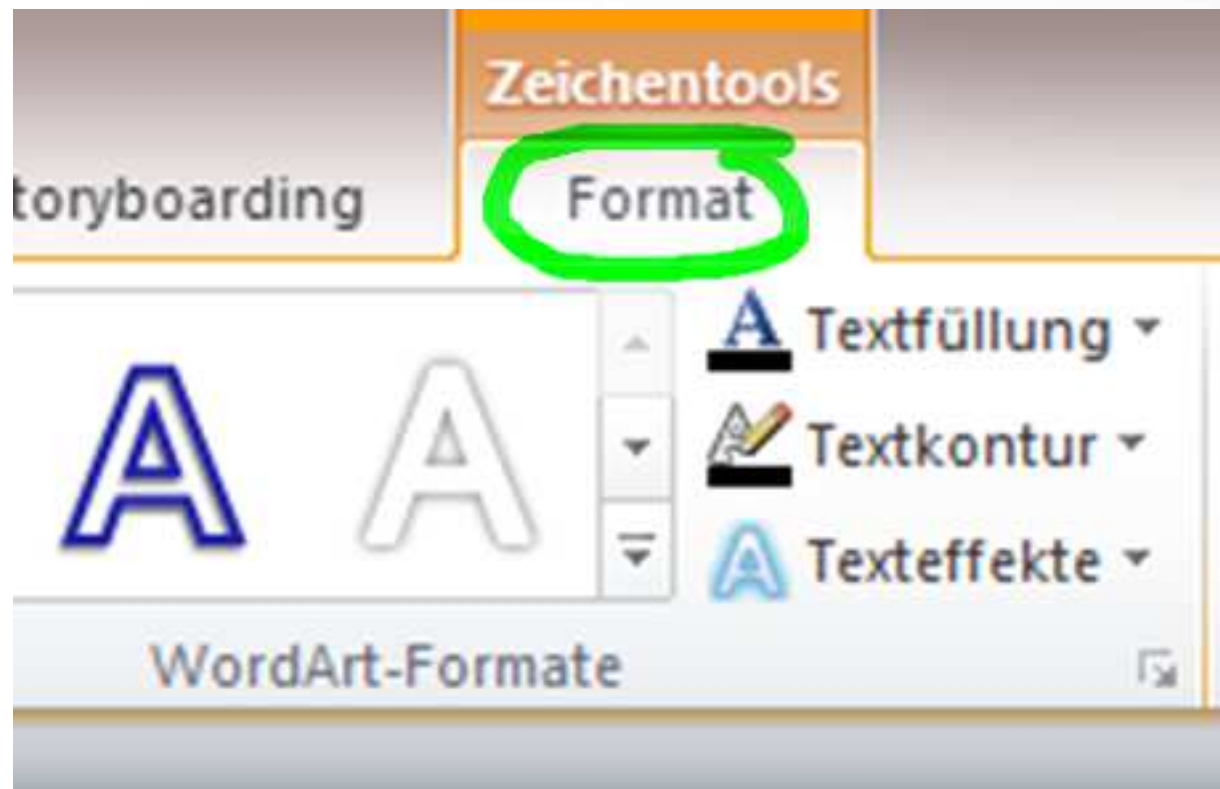
# Ribbon Tool



# Contextual Ribbon Group



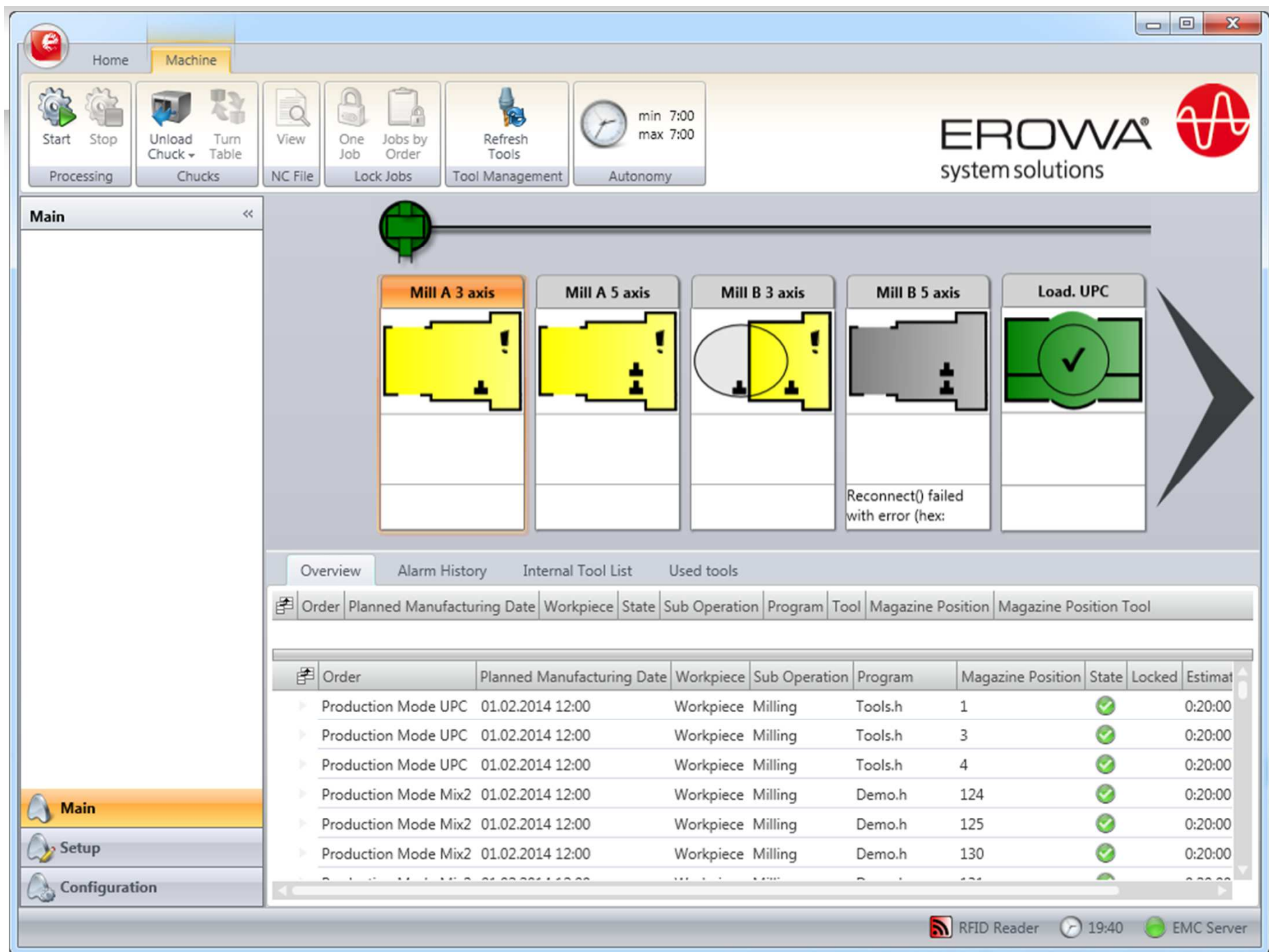
# Contextual Ribbon Tab



# Rub On

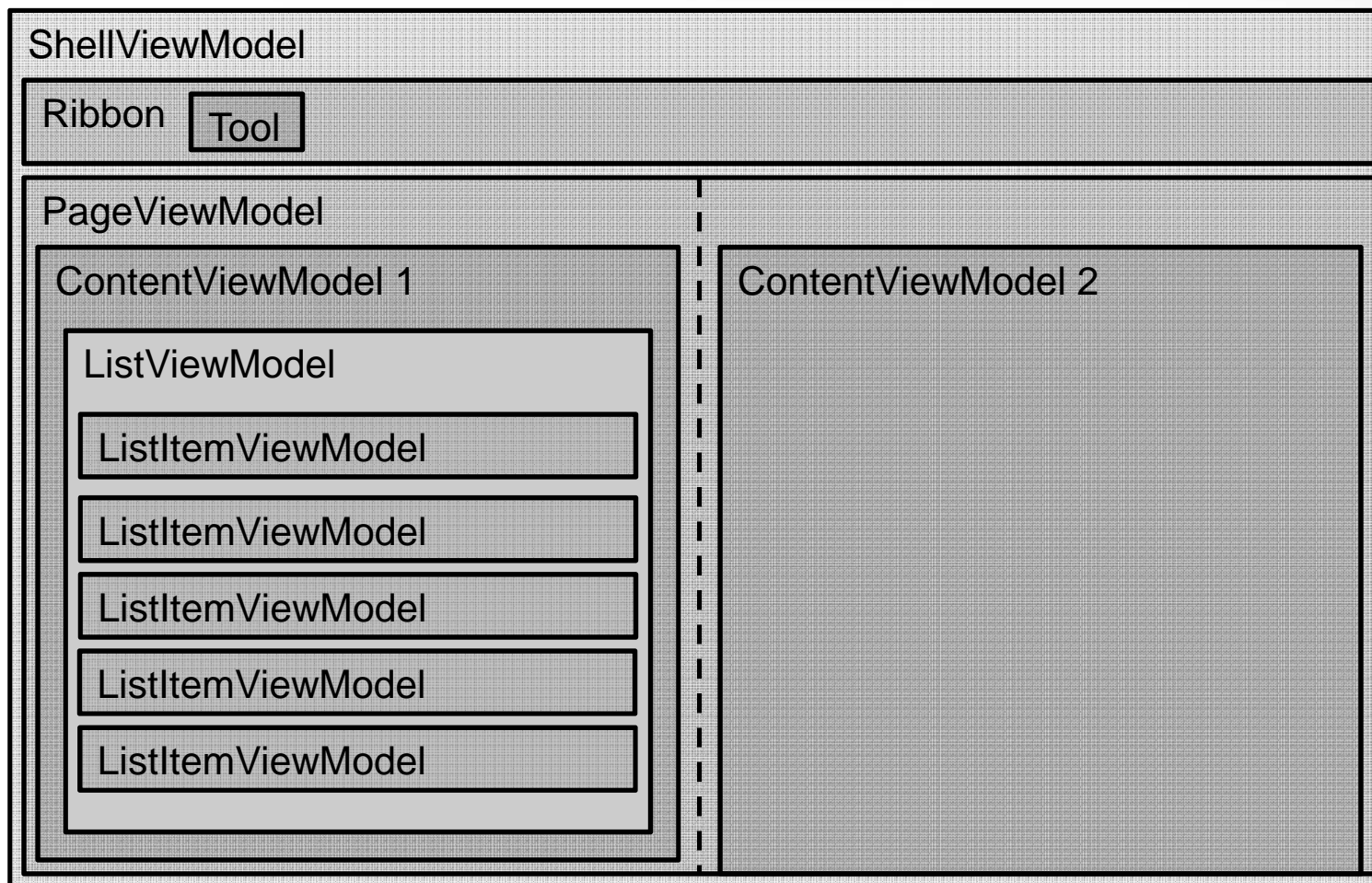


Wo liegt das Problem?

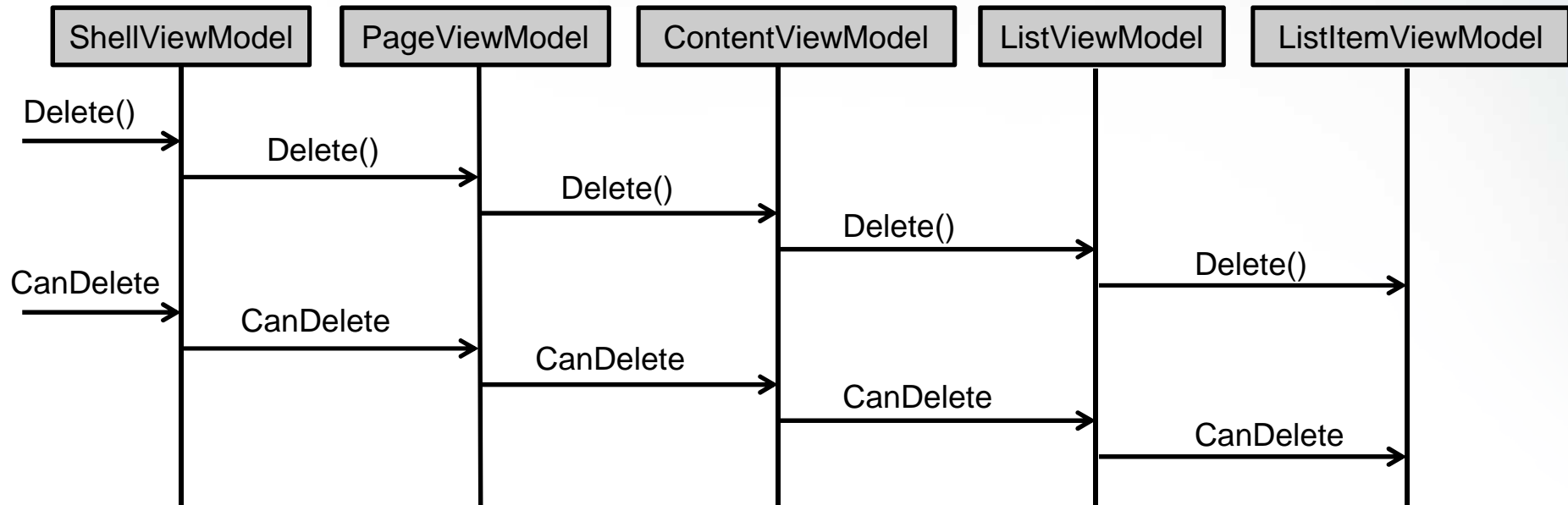




# Problematik durch Komplexität



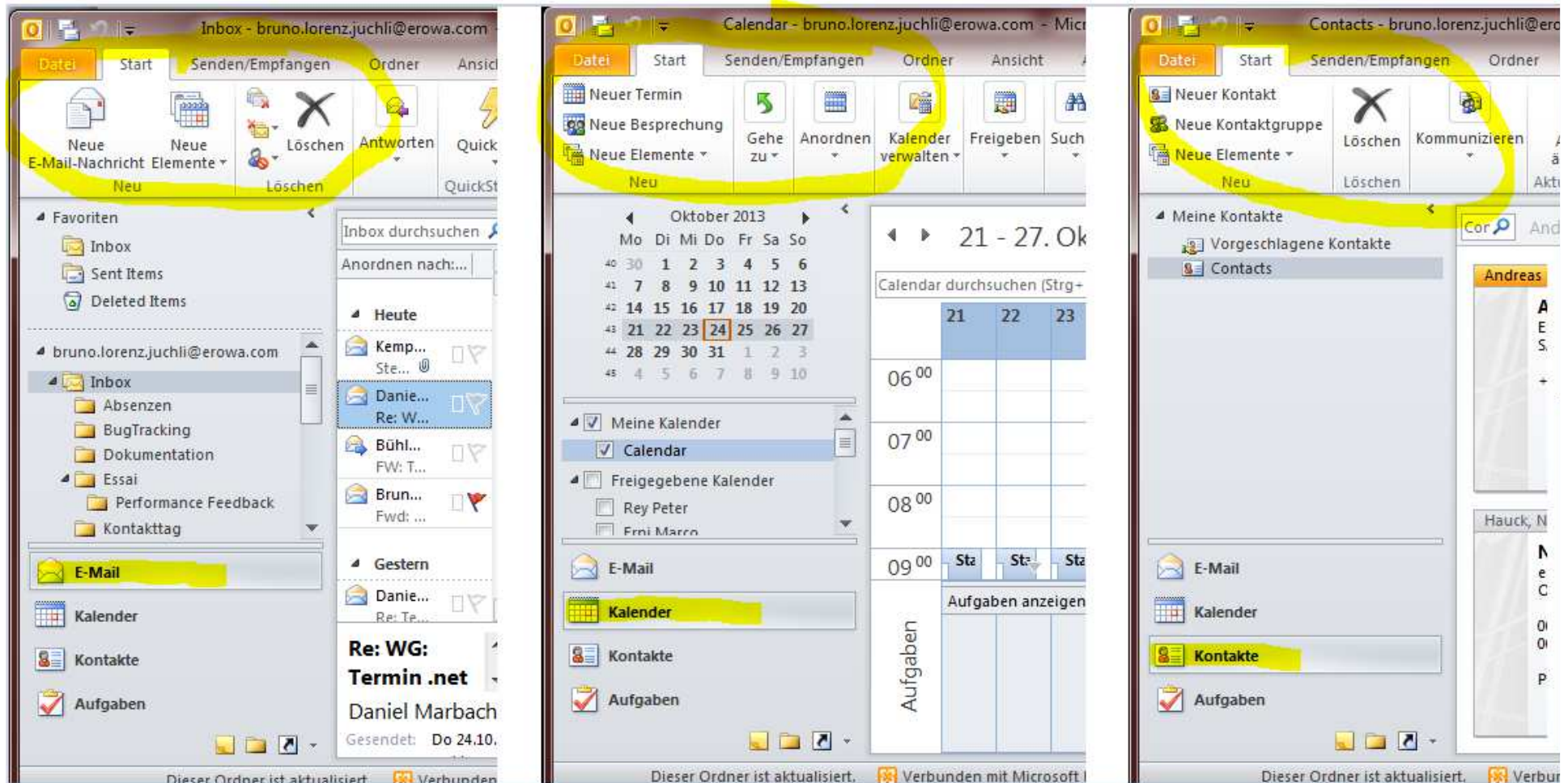
# Problematik durch Komplexität



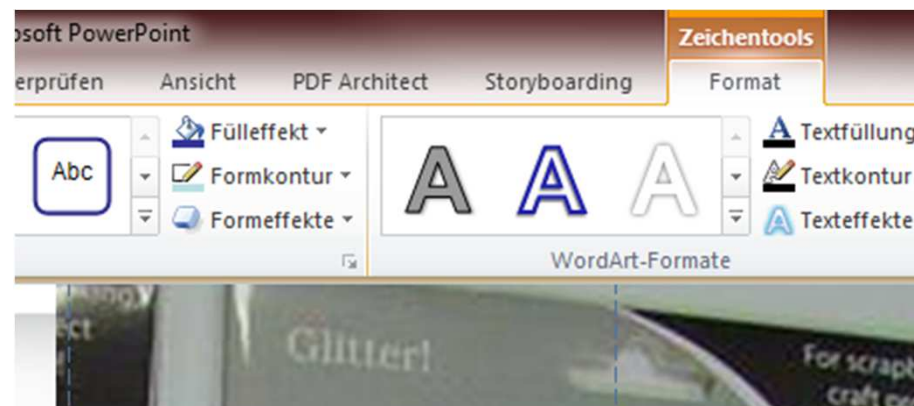
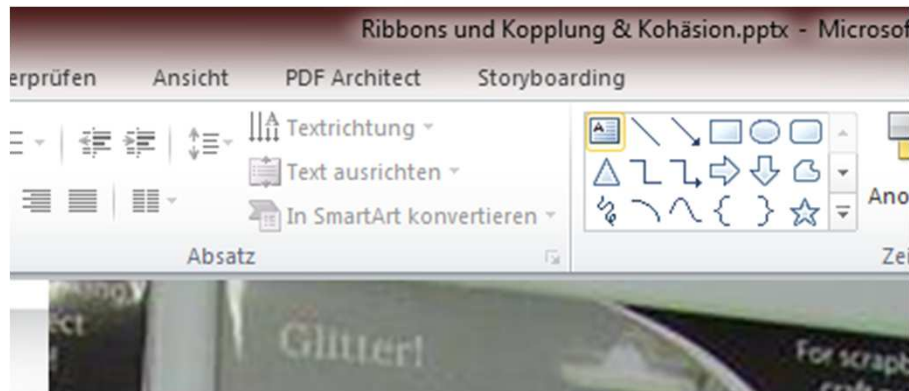
- | View Models müssen Detailfunktionalität kennen
- | Funktionalität muss auf jeder Ebene getestet werden
- | Resultiert in einer engen Kopplung

# Anforderung an Ribbons

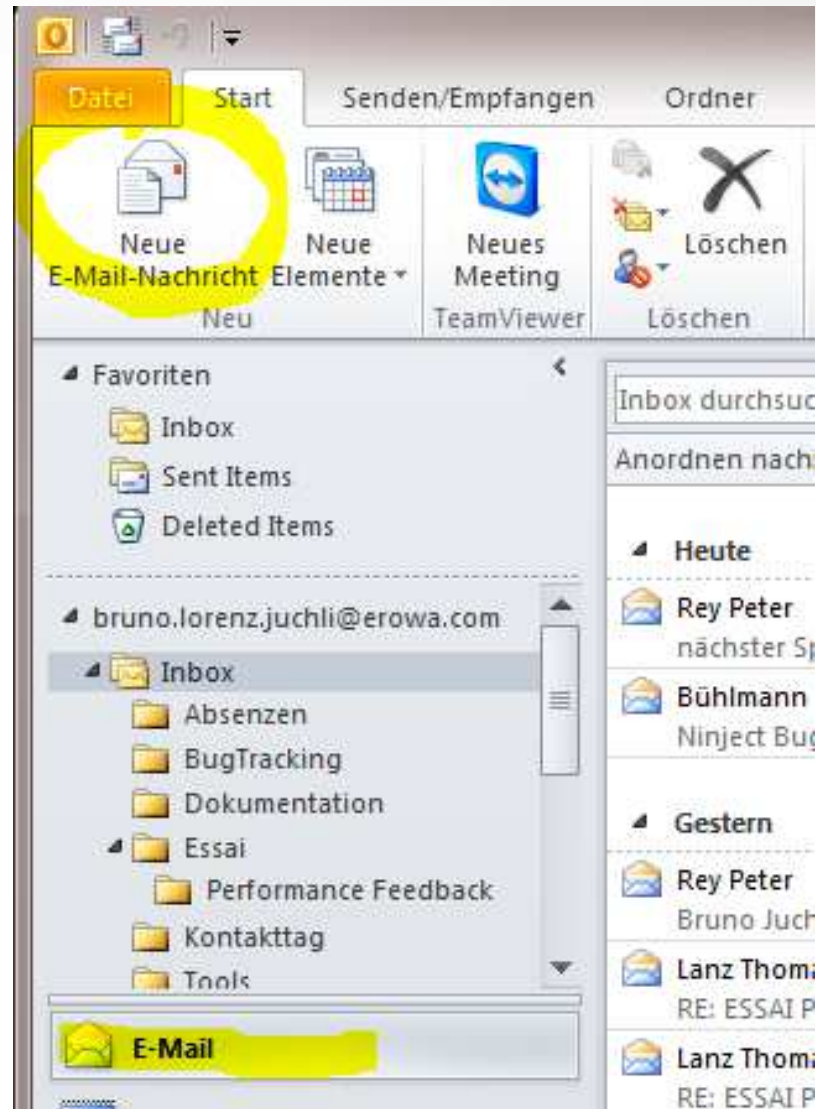
# Anzeige von Elementen abhängig von Navigation



## Anzeige von Elementen abhängig von Selektion → Contextual Ribbon



# Verfügbarkeit der Funktionalität abhängig von Navigation





# Verfügbarkeit der Funktionalität abhängig von Selektion



# Zusammenfassung

- | Anzeige von Elementen abhängig von
  - | Navigation
  - | Selektion
  
- | Verfügbarkeit der Funktionalität abhängig von
  - | Navigation
  - | Selektion
  
- | → Tracking von Navigation und Selektion





Quelle: <http://www.flickr.com/photos/blackcountrymuseums/4079665285/>

EROWA  
Jobs

# Konzeptionelle Lösung

# Was brauchen wir ...

- Wann sollen welche Ribbon Tabs angezeigt werden?
- Wann sollen welche Ribbon Tools verknüpft werden?
- Definition des Ribbon
- Anzeigen der Ribbon Tabs zum richtigen Zeitpunkt
- Verknüpfen der Ribbon Tools mit Ihren Commands



# Navigation

- | Verändert was auf dem GUI angezeigt wird
- | ➔ Andere View Models werden angezeigt
  
- | Wir nennen ein gerade angezeigtes View Model:  
**aktives View Model**



# Selektion

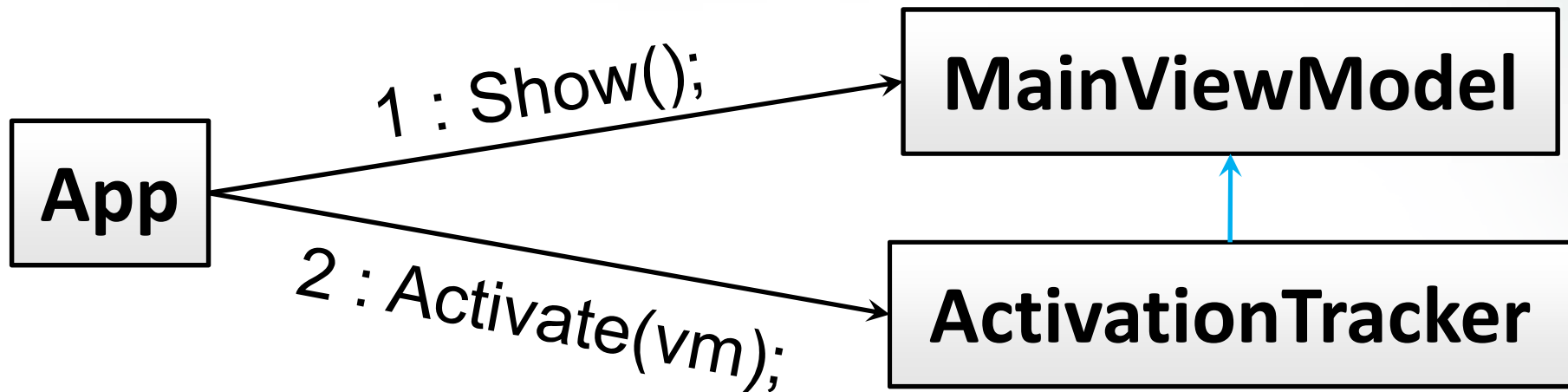
- | Erfolgt auf einem aktiven View Model
- | Die Selektion wird typischerweise auf dem View Model mit einem "SelectedItem" repräsentiert
- | Fügen wir nun also die selektierten View Models auch zur Menge der aktiven View Models dazu.



# Tracking der Aktiven View Models

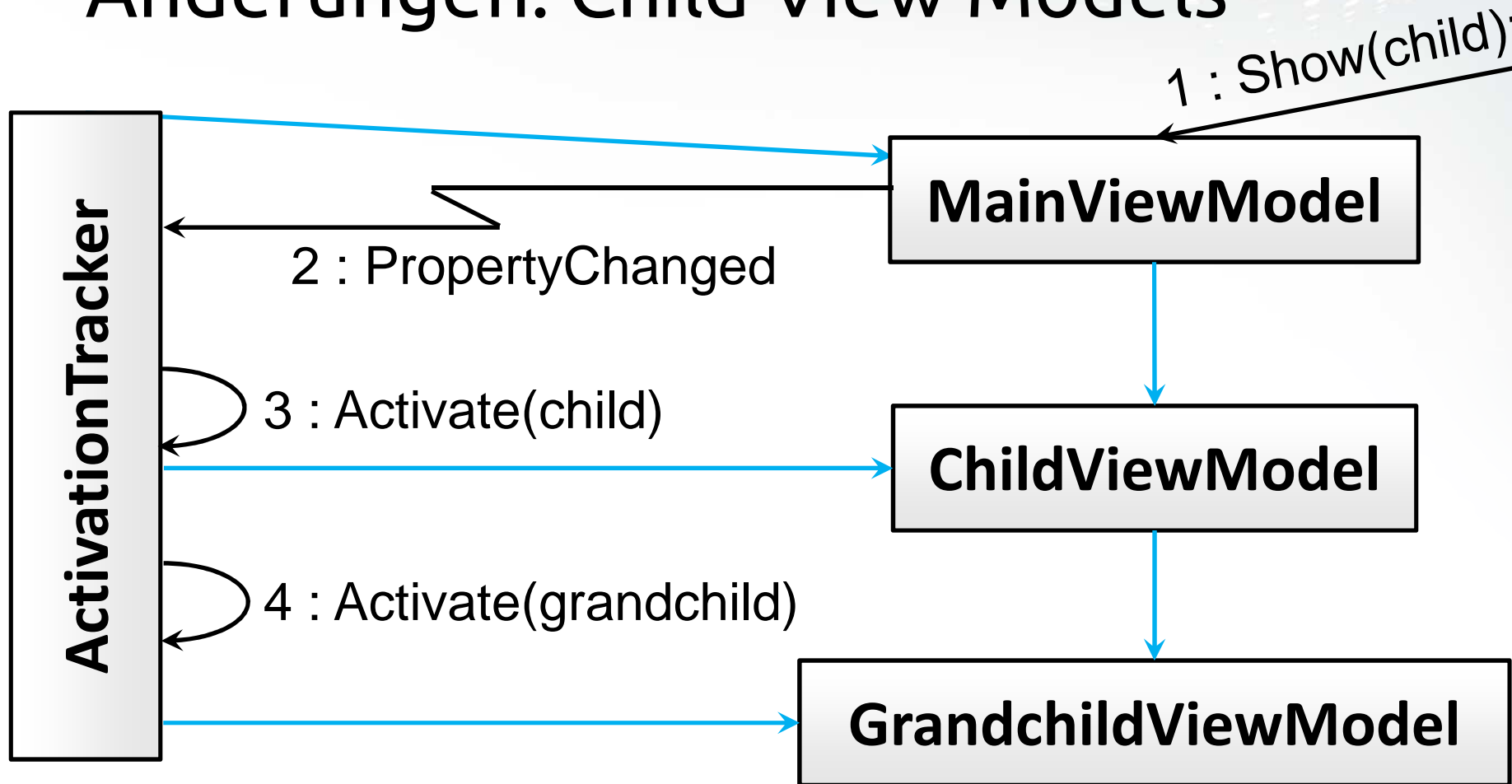


# Der Anfang: Root View Model



Active View Models: Main

# Änderungen: Child View Models



Active View Models: Main Child Grandchild



## Wir wissen nun...

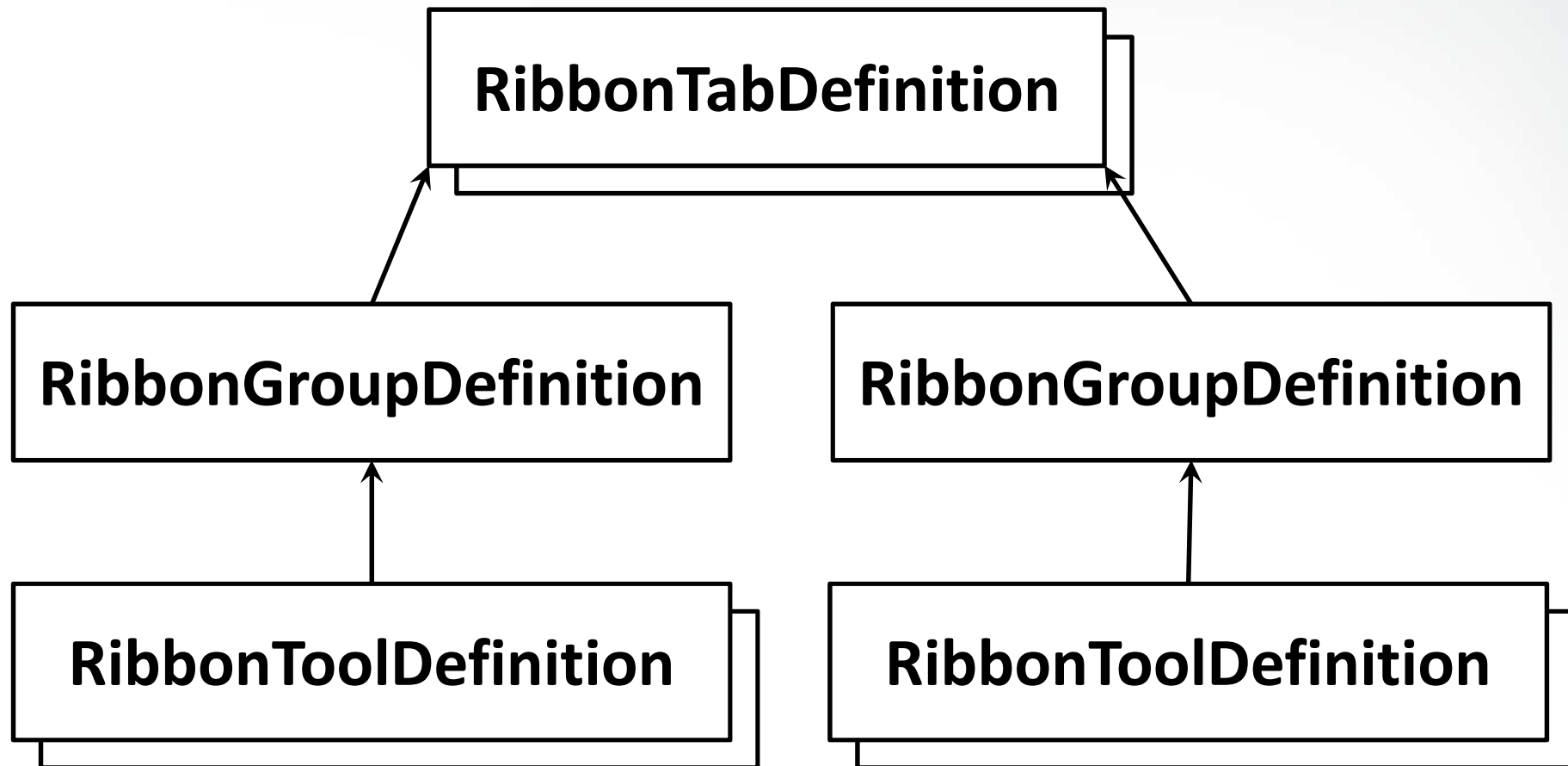
- Wann welche View Models “aktiv” sind (Navigation, Selektion)

## Was brauchen wir noch...

- Definition des Ribbon
- Anzeigen der Ribbon Tabs zum richtigen Zeitpunkt
- Verknüpfen der Ribbon Tools mit Ihren Commands



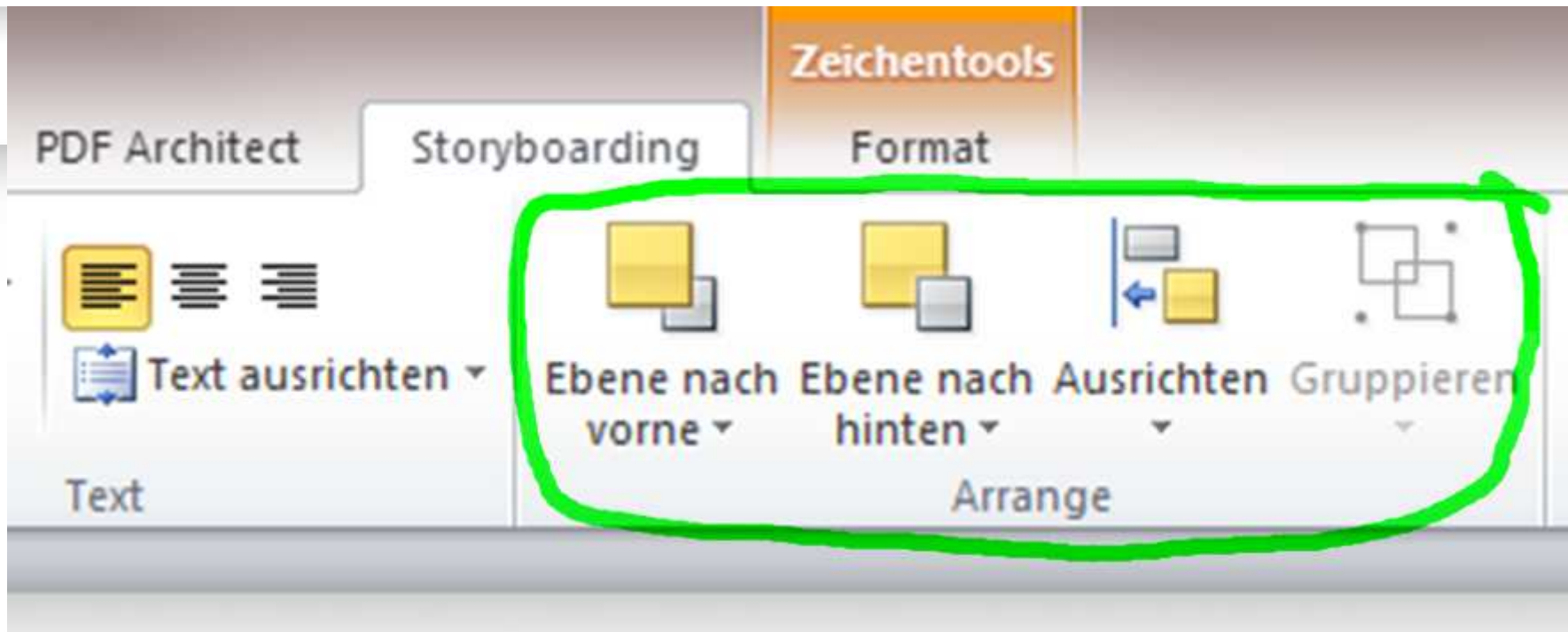
# Definition des Ribbons





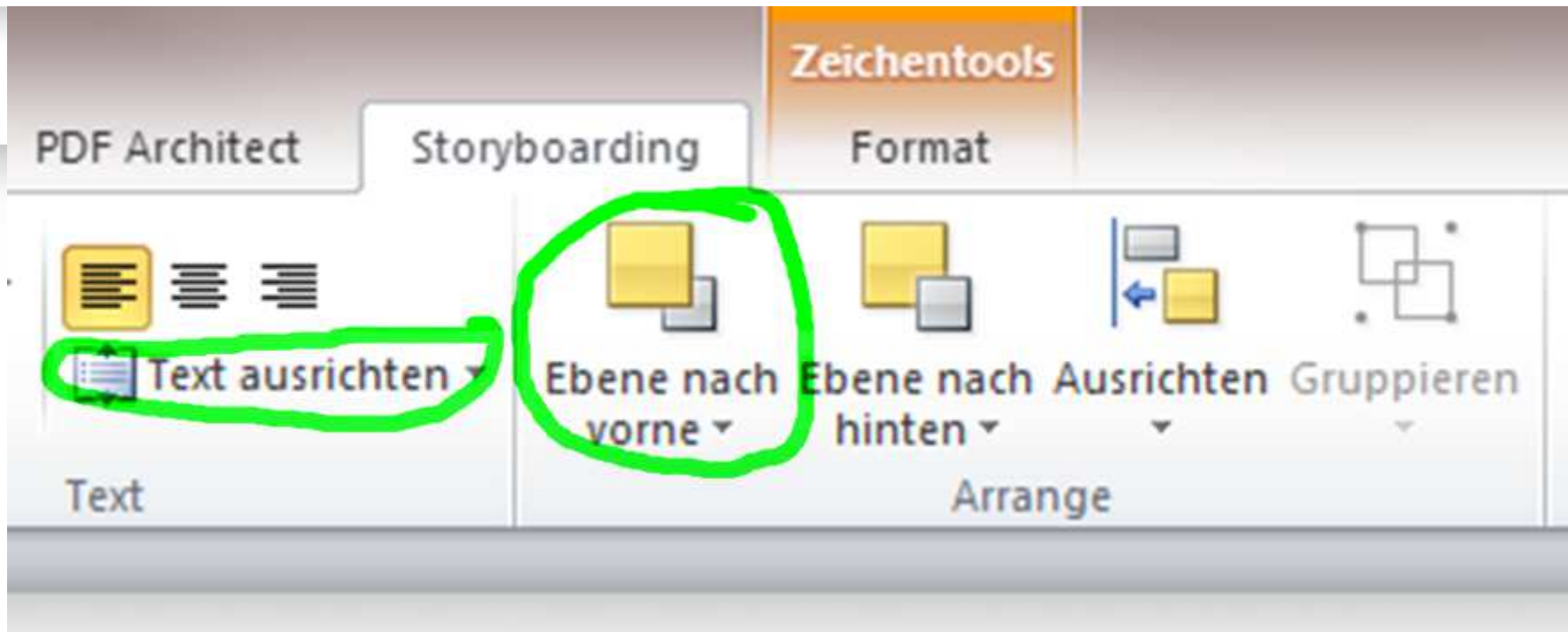
## RibbonTabDefinition

- **string Name;**
- **int Sequence;**
- **Type DisplayWhenViewModellsActive;**



## RibbonGroupDefinition

- Type Tab;
- string Name;
- int Sequence;



## RibbonToolDefinition

- Type Group;
- string Name;
- int Sequence;
- Uri LargeIcon;
- Type Command;
- Type WireToViewModel;

# Wir wissen nun...

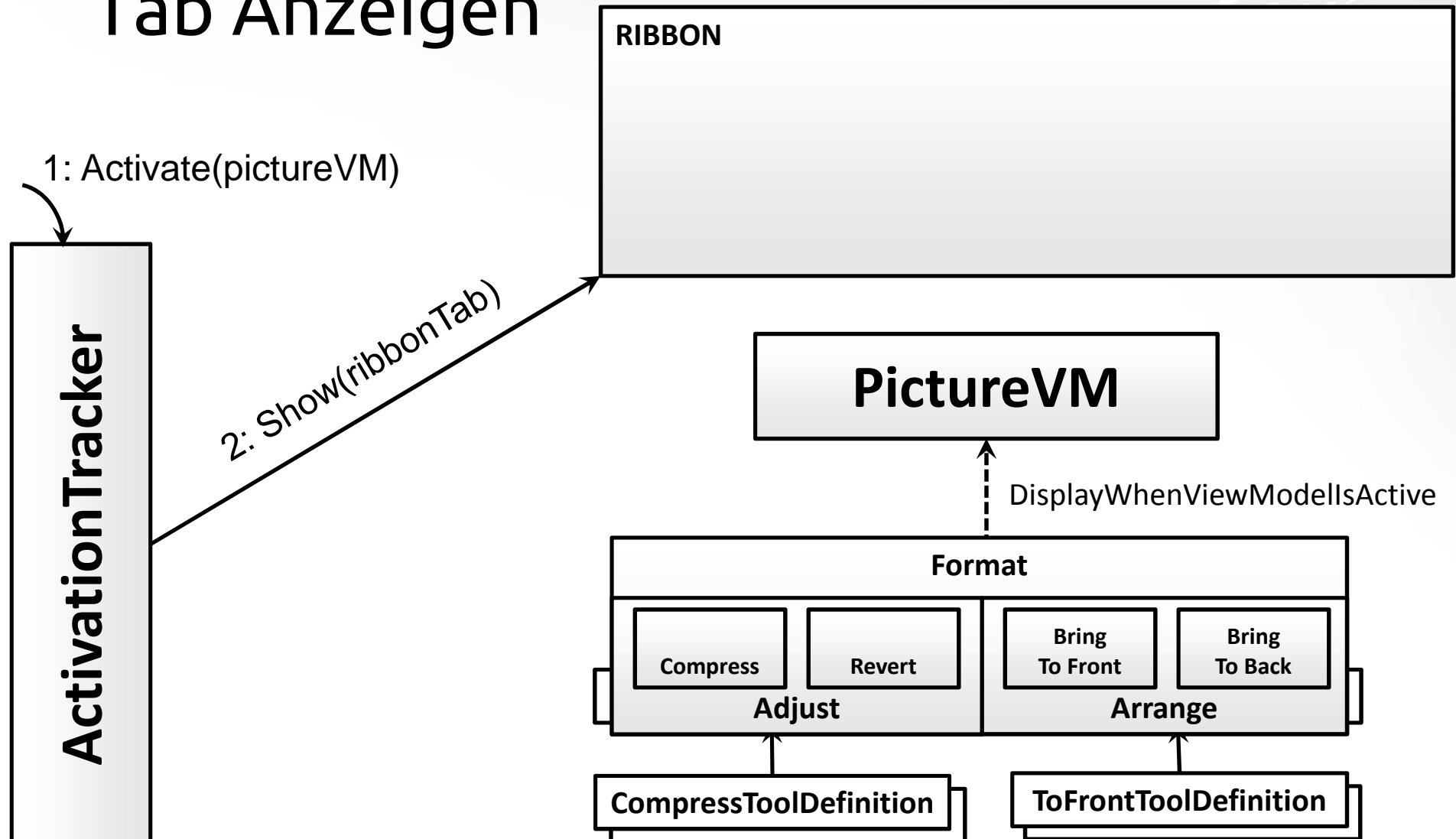
- Wann welche View Models “aktiv” sind (Navigation, Selektion)
- Wann welches Ribbon Tab angezeigt werden soll
- Welche Elemente (Groups, Tools) ein Tab beinhaltet
- Welches Command die Funktionalität eines Tools implementiert



# Nächste Schritte..

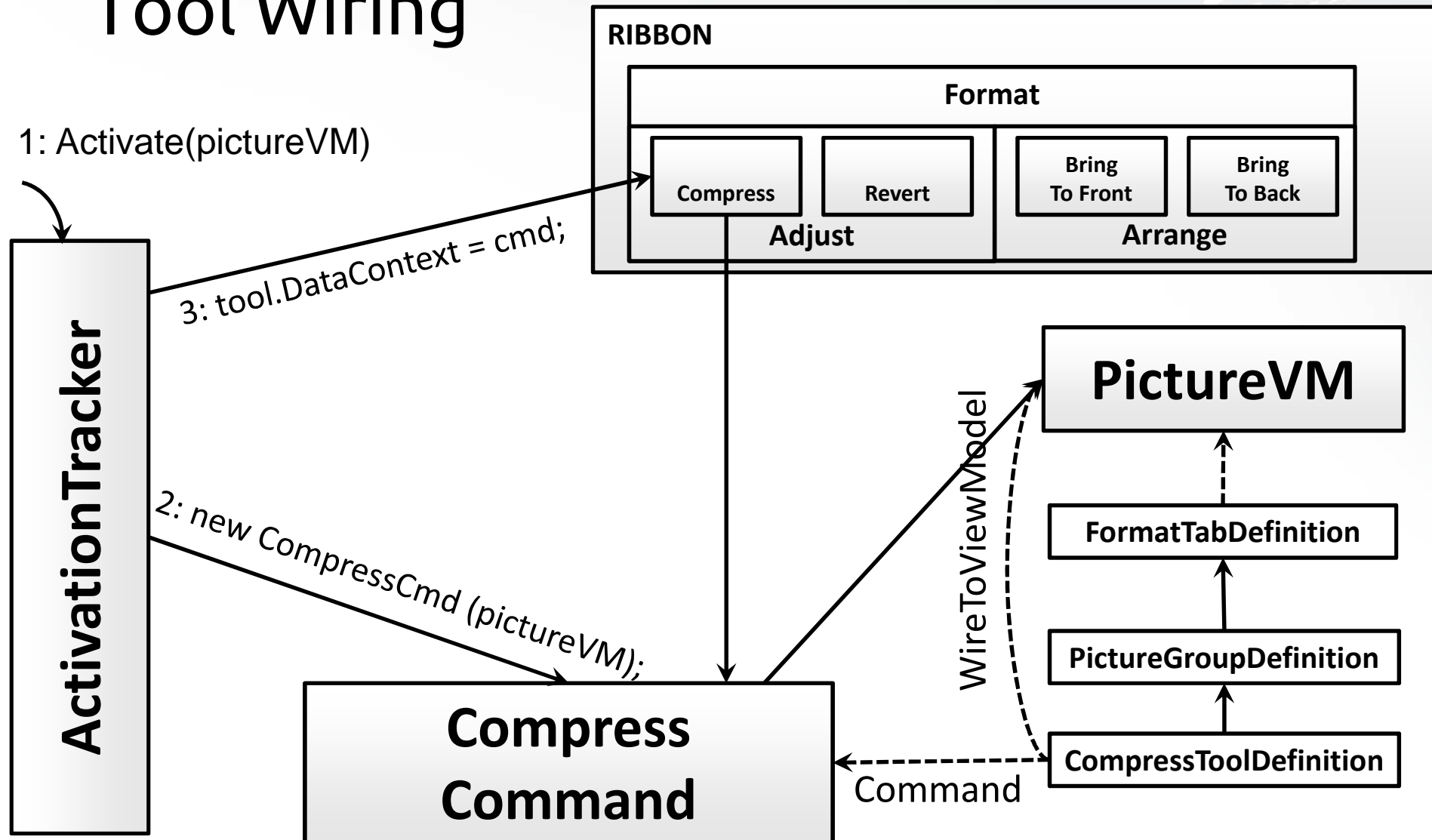
- Ribbon Tab
  - anzeigen wenn entsprechendes View Model aktiv wird
- Ribbon Tool
  - Wenn entsprechendes View Model aktiv wird:
    - Command mit dem View Model verknüpfen
    - Tool mit Command verknüpfen

# Tab Anzeigen





# Tool Wiring



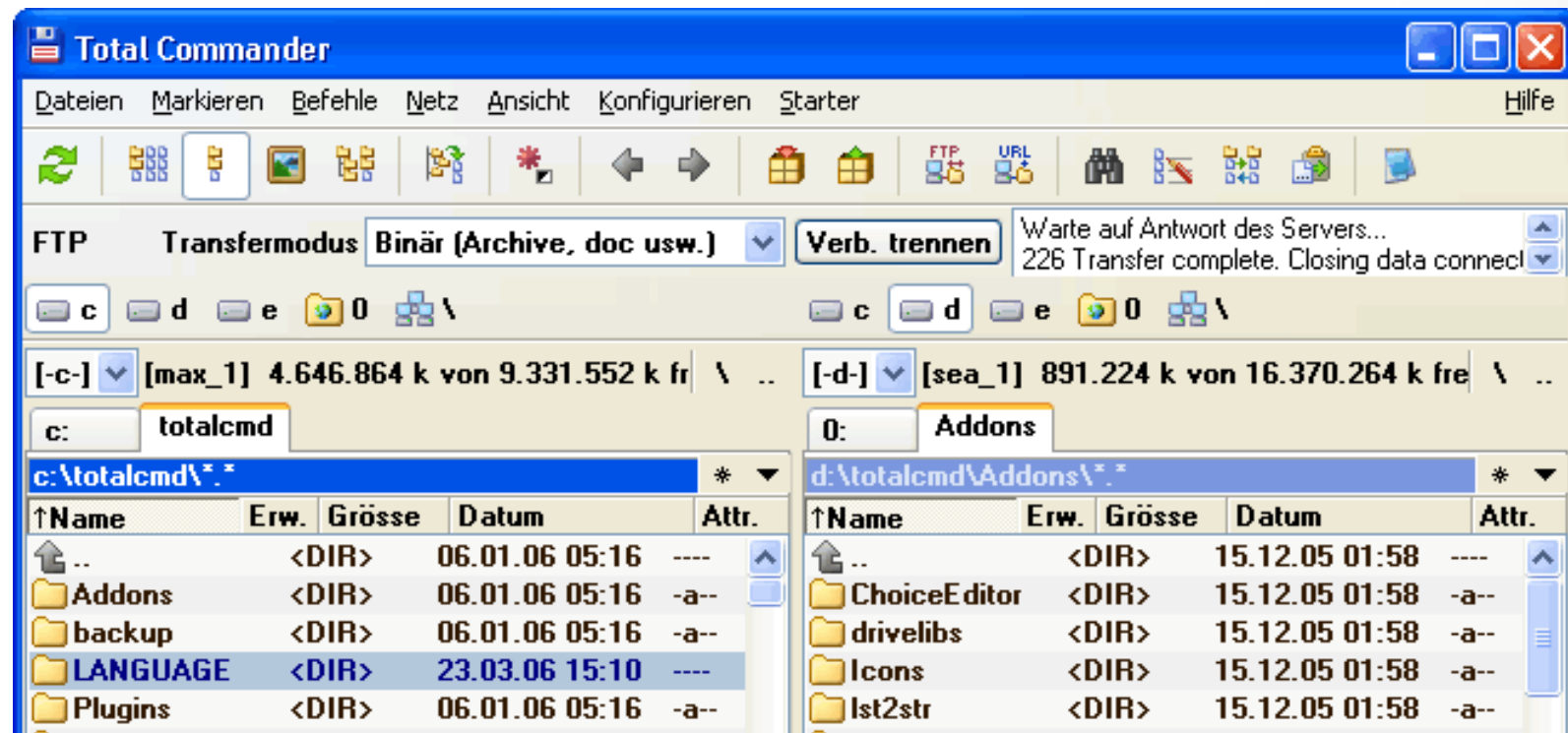
Jackpot !!!

Konzeptionelle Lösung

# Limitationen

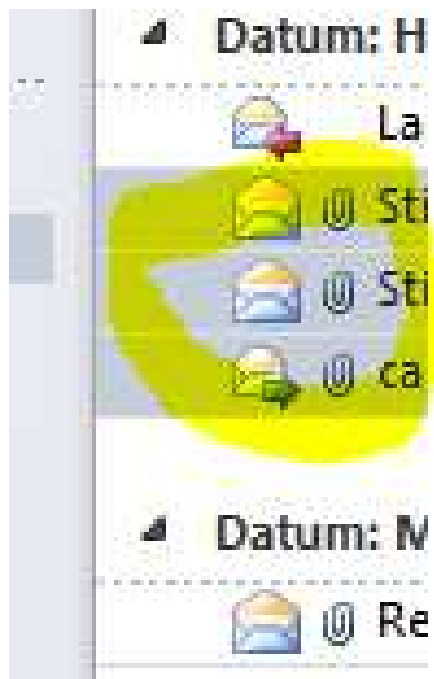
# Limitationen

- Selektion durch Fokus
  - 2+ “gleiche” View-Bäume gleichzeitig
  - Auf welches Element bezieht sich die Funktion?  
(Welcher View-Baum ist im Fokus)



# Limitationen

- Multi-Selection
  - Wie wird multi-selection definiert? Wie wird das command-wiring geändert wenn die Selektion eines Elements dazu kommt?
  - Im Moment funktioniert das Command bei uns auf der Ebene der Parent-View, d.h. jedes Command behandelt die Selektion selber.



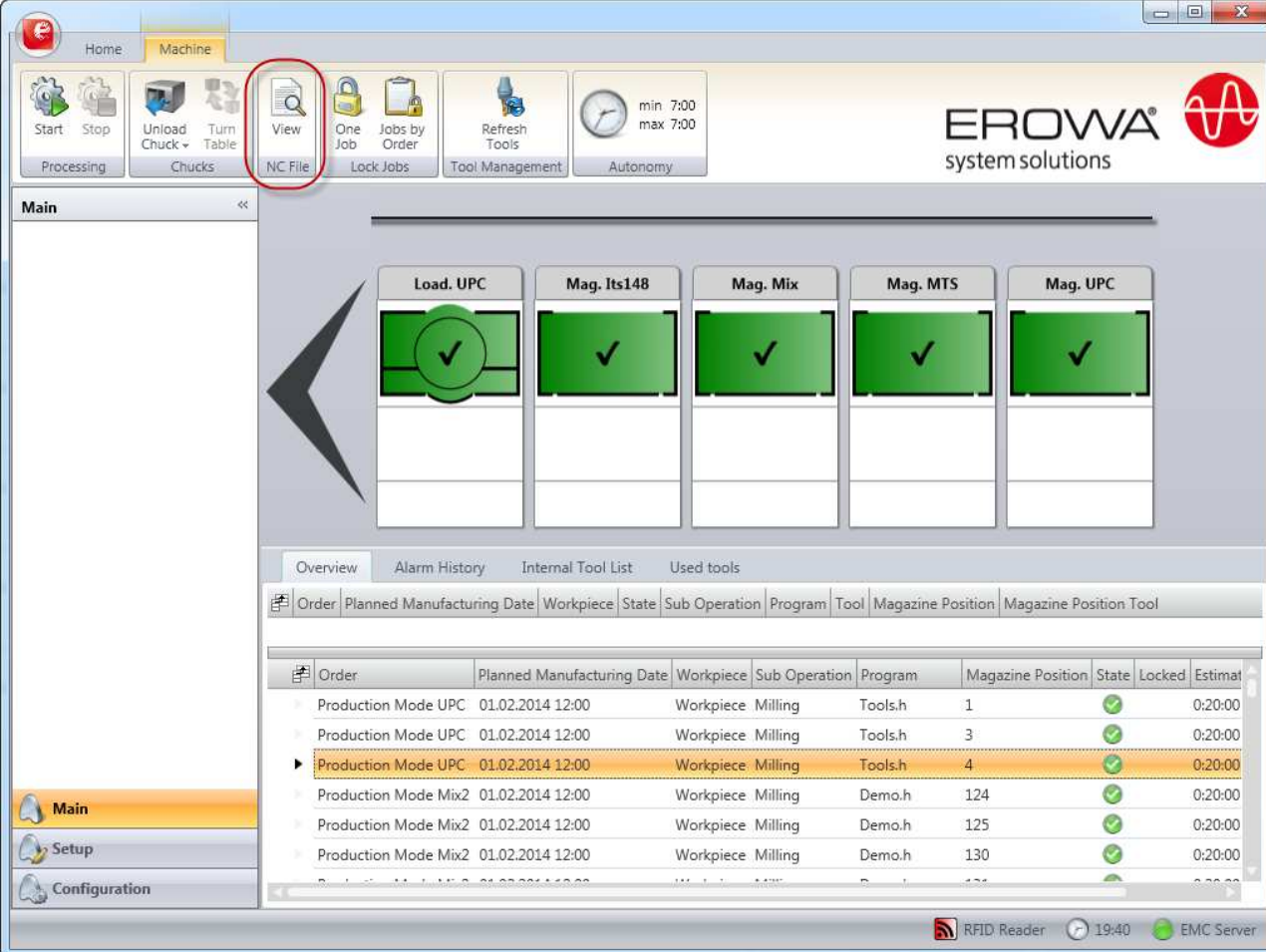
# Limitationen

- IsVisible, IsCollapsed
  - Tracking komplexer machen damit es dies per Convention versteht:  
.Child → .IsChildVisible, .IsChildCollapsed bool properties mit  
PropertyChangedEvent
  - Workaround
    - Nicht benutzen, stattdessen: child = null



Was hat es gebracht?

# Gegenüberstellung Alt - Neu

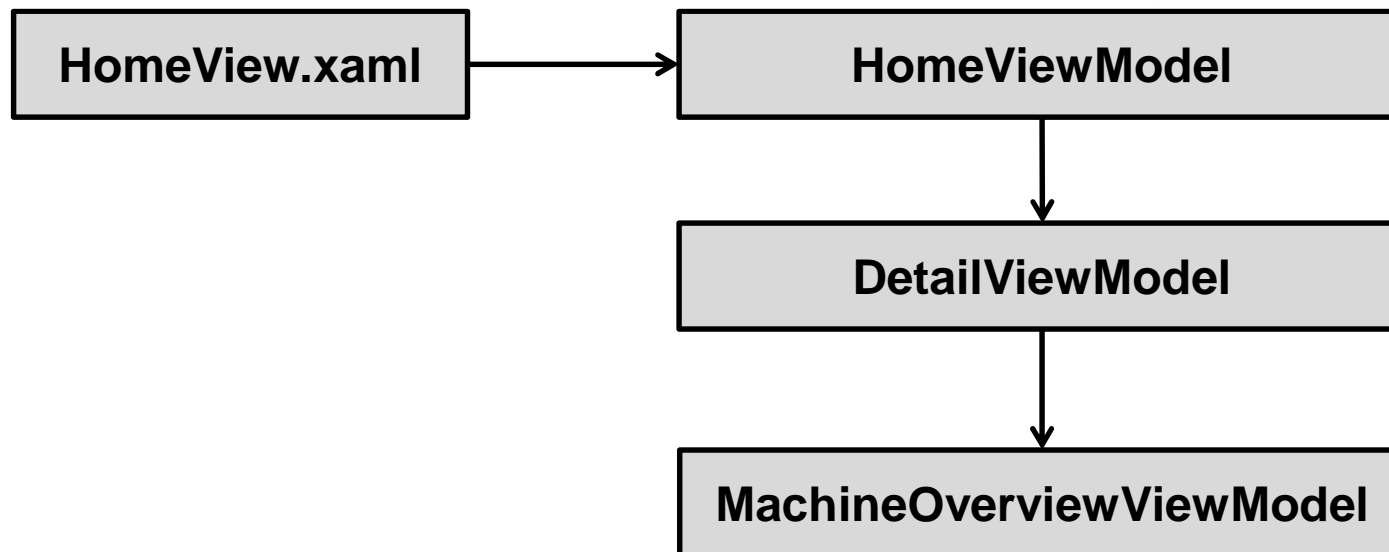


The screenshot displays the EROWA system solutions software interface. The top toolbar includes buttons for Start, Stop, Unload Chuck, Turn Table, View (circled in red), One Job, Jobs by Order, Refresh Tools, and Autonomy. The main area shows five magazine status boxes (Load. UPC, Mag. Its148, Mag. Mix, Mag. MTS, Mag. UPC) with green checkmarks. Below is a table with columns: Order, Planned Manufacturing Date, Workpiece, Sub Operation, Program, Magazine Position, State, Locked, and Estimat. The table lists several production orders, with the third one highlighted in orange.

Order	Planned Manufacturing Date	Workpiece	Sub Operation	Program	Magazine Position	State	Locked	Estimat
Production Mode UPC	01.02.2014 12:00	Workpiece	Milling	Tools.h	1	✓		0:20:00
Production Mode UPC	01.02.2014 12:00	Workpiece	Milling	Tools.h	3	✓		0:20:00
Production Mode UPC	01.02.2014 12:00	Workpiece	Milling	Tools.h	4	✓		0:20:00
Production Mode Mix2	01.02.2014 12:00	Workpiece	Milling	Demo.h	124	✓		0:20:00
Production Mode Mix2	01.02.2014 12:00	Workpiece	Milling	Demo.h	125	✓		0:20:00
Production Mode Mix2	01.02.2014 12:00	Workpiece	Milling	Demo.h	130	✓		0:20:00



# Gegenüberstellung *Alt*- Neu



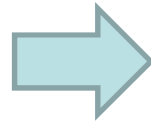
# Gegenüberstellung *Alt*- Neu

## | HomeView.xaml

| Definition des Ribbon Buttons

## | HomeViewModel

## | DetailViewModel



Weiterleitung des Handlers

Weiterleitung der Guard

Weiterleitung des PropertyChanged-Events

## | MachineOverviewViewModel

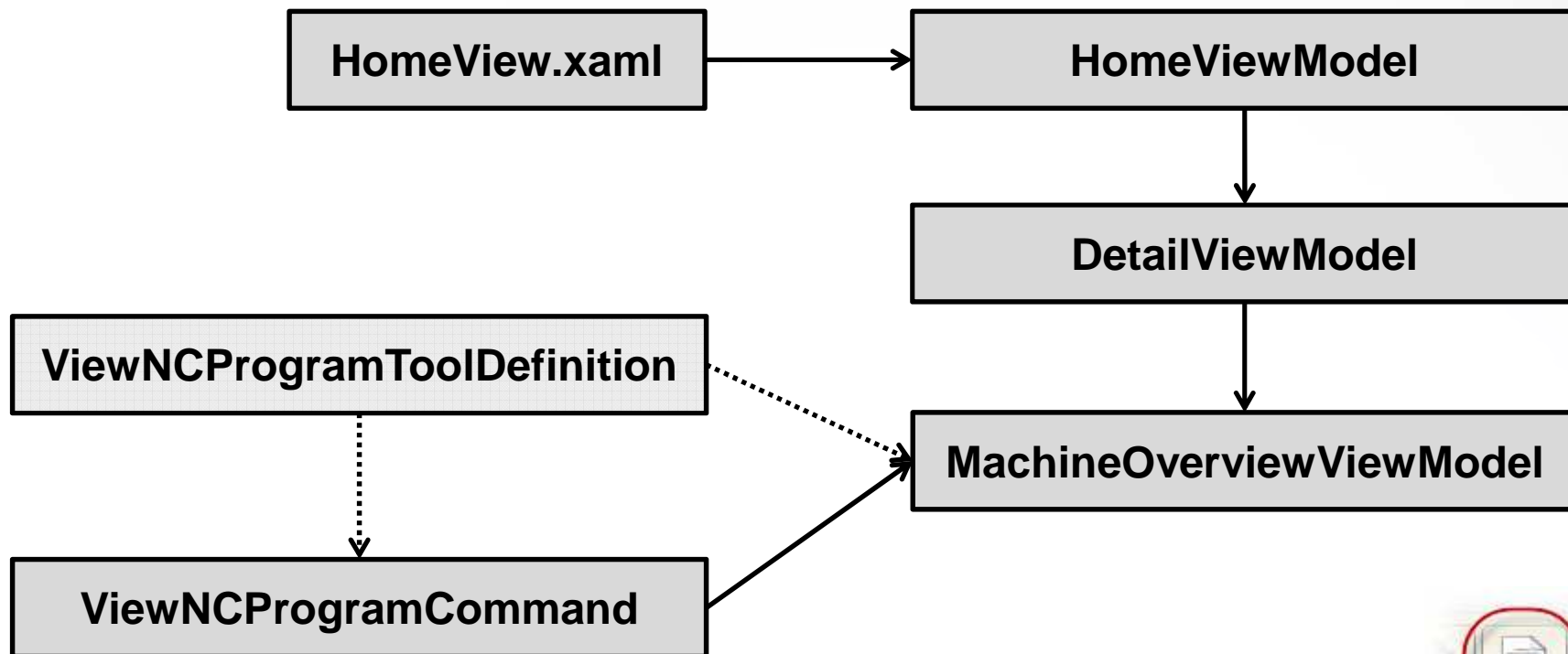
| Implementation des Handlers

| Implementation der Guard

| Handler Selektions-Events (Update der Guard)



# Gegenüberstellung Alt - *Neu*



# Gegenüberstellung Alt - Neu

| HomeViewModel  
| 699 Zeilen

| DetailViewModel  
| 395 Zeilen

| MachineOverviewViewModel  
| 397 Zeilen

| HomeViewModel  
| 94 Zeilen

| DetailViewModel  
| 34 Zeilen

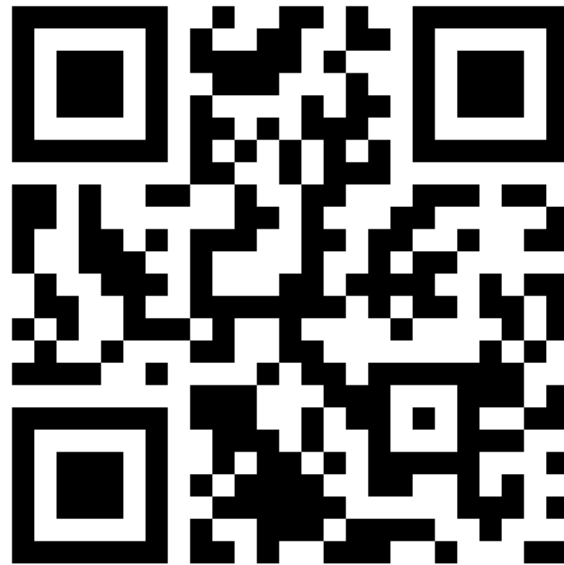
| MachineOverviewViewModel  
| 228 Zeilen

| ViewNCProgramCommand  
| 43 Zeilen



# Implementation

<http://tiny.cc/0dy1ax>



[github.com/BrunoJuchli/RibbonizerSample](https://github.com/BrunoJuchli/RibbonizerSample)

# Fragen & Diskussion

- | Fragen
- | Verbesserungsvorschläge
- | Ideen
- | Erfahrungen der Teilnehmer



*[www.erowa.com](http://www.erowa.com)*

→ Jobs!