1.– 4. September 2014
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Reactive UIs

Event-gesteuerte Architektur für moderne Applikationen

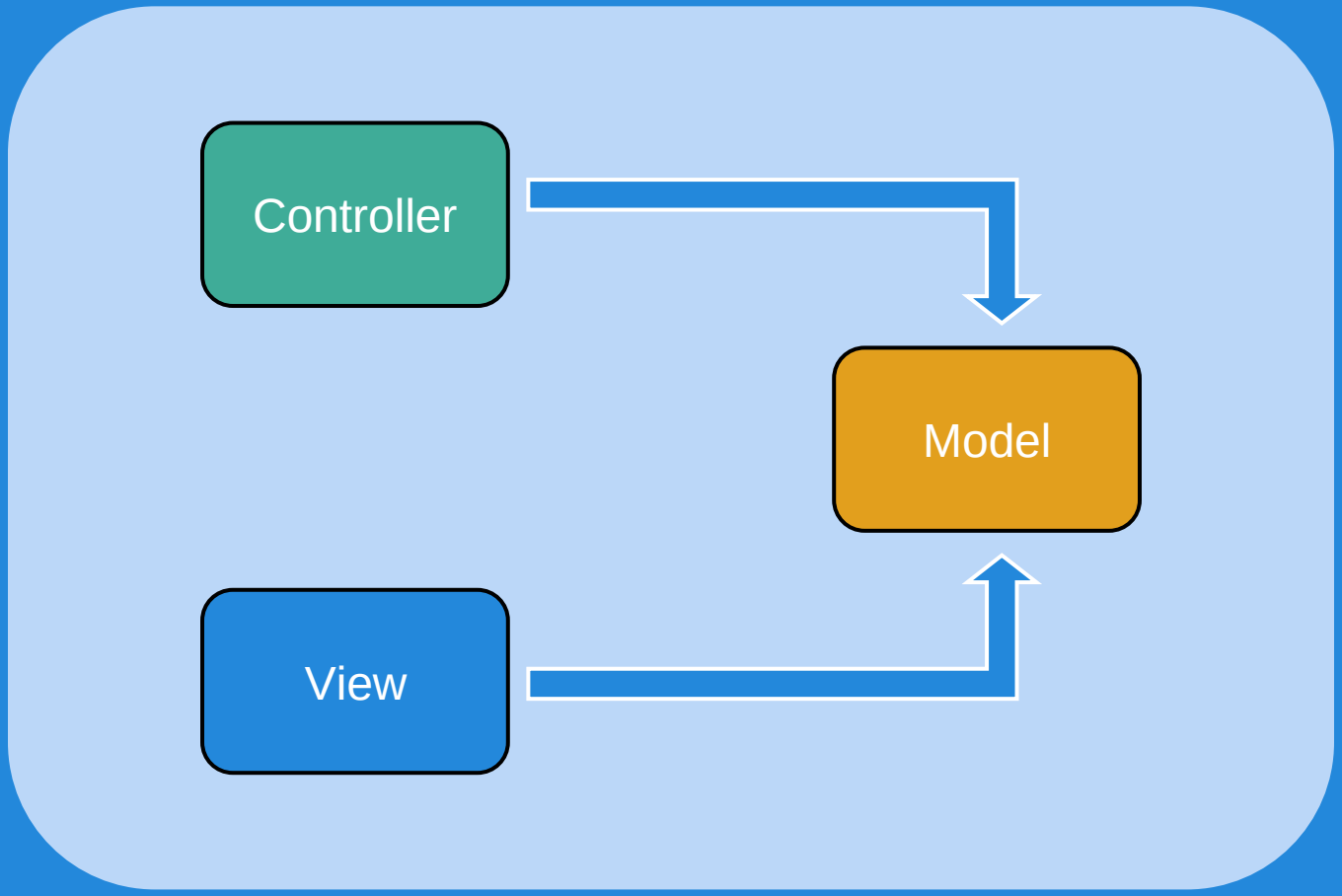
Thomas Spiegl und Manfred Geiler
Irian Solutions – The Software Experts

Agenda

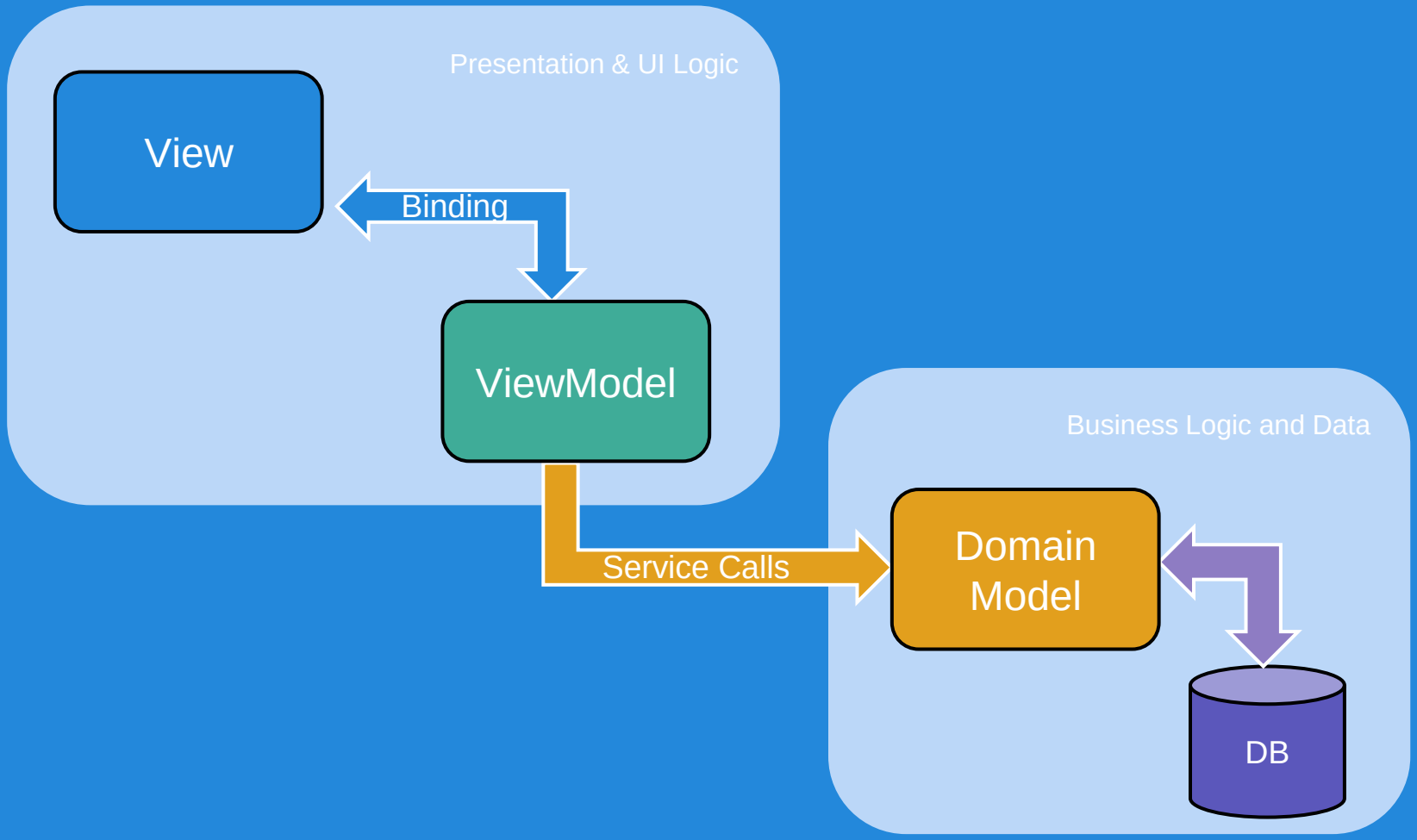
- M V V M
- Neues Konzept MVSVM
- Ankor Framework
- Ankor Live Sample
- Ankor Special Features

„M V V M" ?

- „Model View ViewModel"
- 2005 von John Gossman (Microsoft)
- ≅ „Presentation Model" von Martin Fowler
- Seperation of Concerns
  - Graphical UI
  - UI Logic



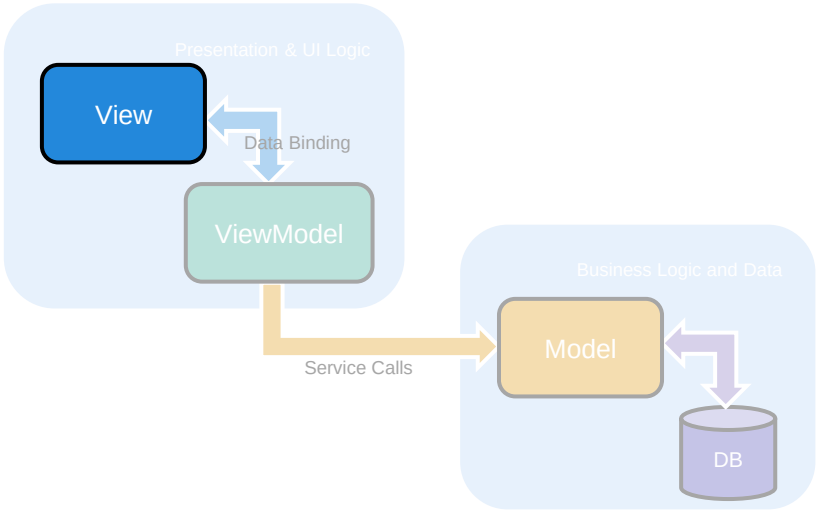
Model View Controller



Model View ViewModel

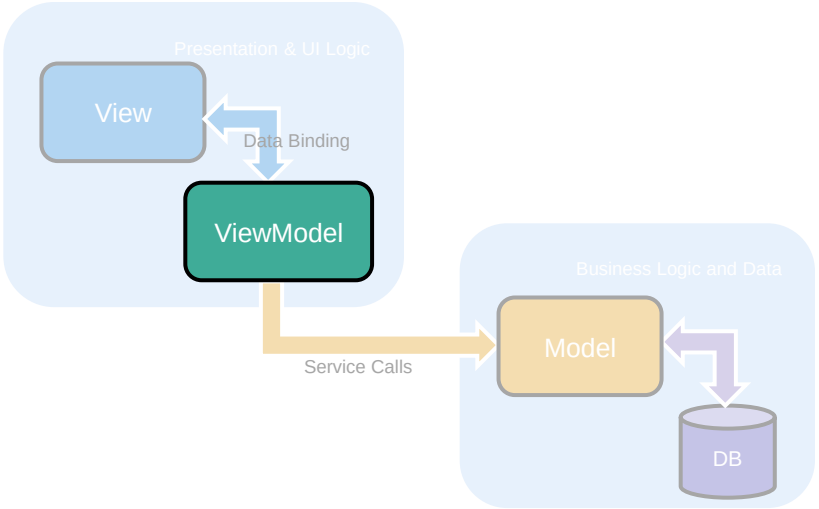
M V V M - View

- Grafische Benutzeroberfläche (GUI)
- Benutzereingaben
- Datenbindung („Binding“) auf ViewModel
- Markup
  - XAML
  - FXML



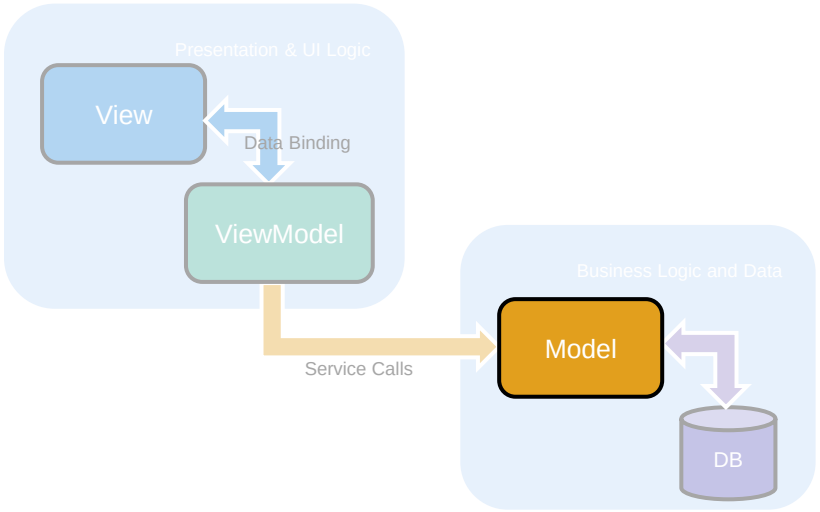
M V V M - ViewModel

- Bindeglied zwischen View und Model
- Verbindung mit Model (Service-Calls)
- Properties und Actions für View (Binding)
- UI-Logik



M V V M - Model

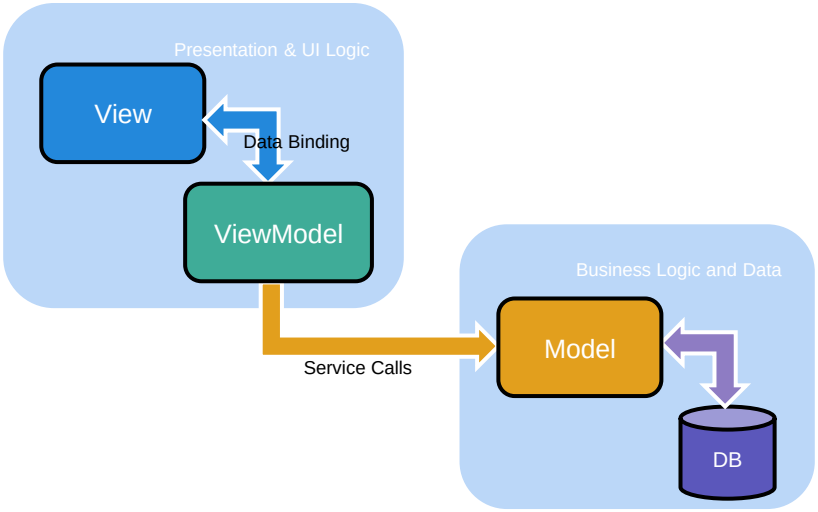
- Domain Model, Datenzugriff
- Domain Logik
- Validierung
- Unit Tests



Was MVVM löst...

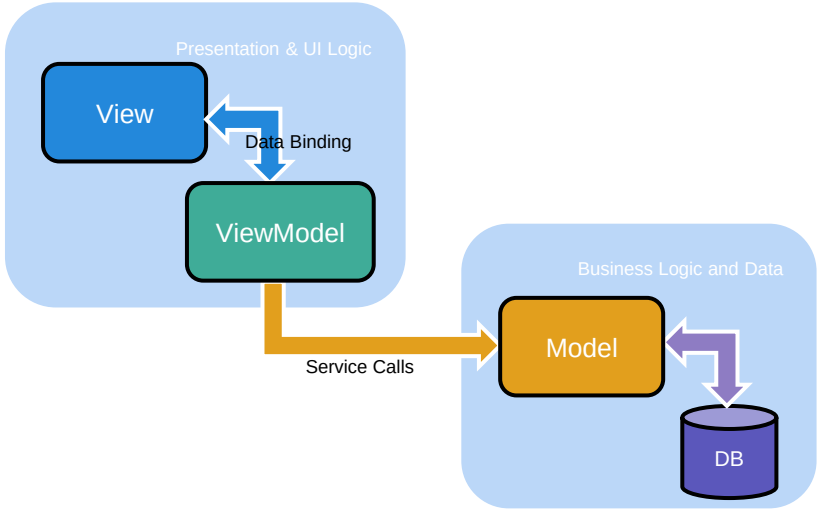
- Separation of Concerns
  - Designer ↔ UI-Entwickler
  - View-Technologie ↔ Präsentations-Logik

- ViewModel testbar!
  - Unit-Tests
  - Mock-UI



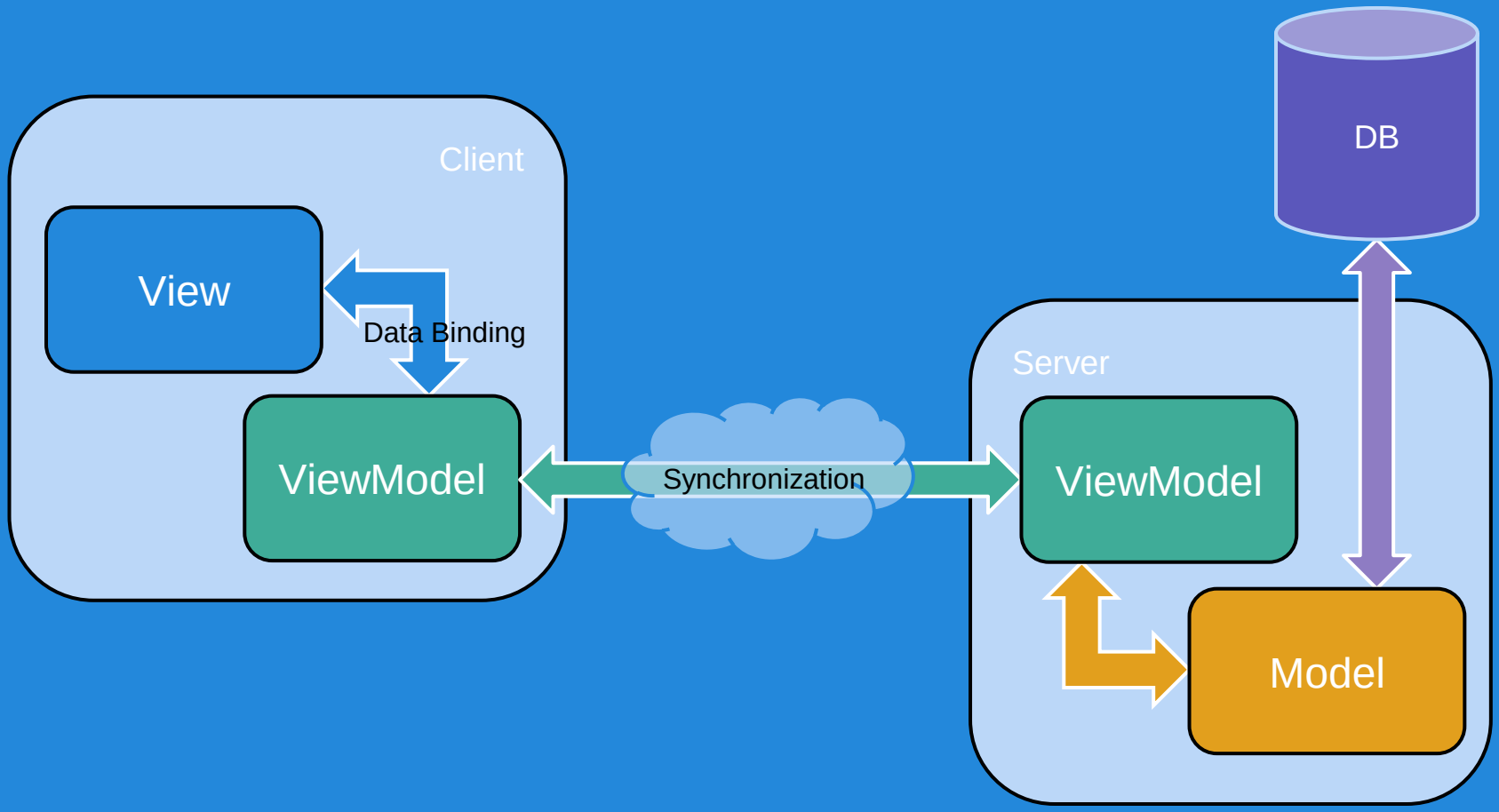
Was MVVM nicht löst...

- Model am Client oder Server?
- Kommunikation Client ↔ Server
- Problem: Vielfalt Client-Technologien
  - HTML5
  - iOS
  - Android
  - JavaFX
  - ...



Agenda

- M V V M
- **Neues Konzept MVSVM**
- Ankor Framework
- Ankor Live Sample
- Ankor Special Features



Model View SynchronizedViewModel

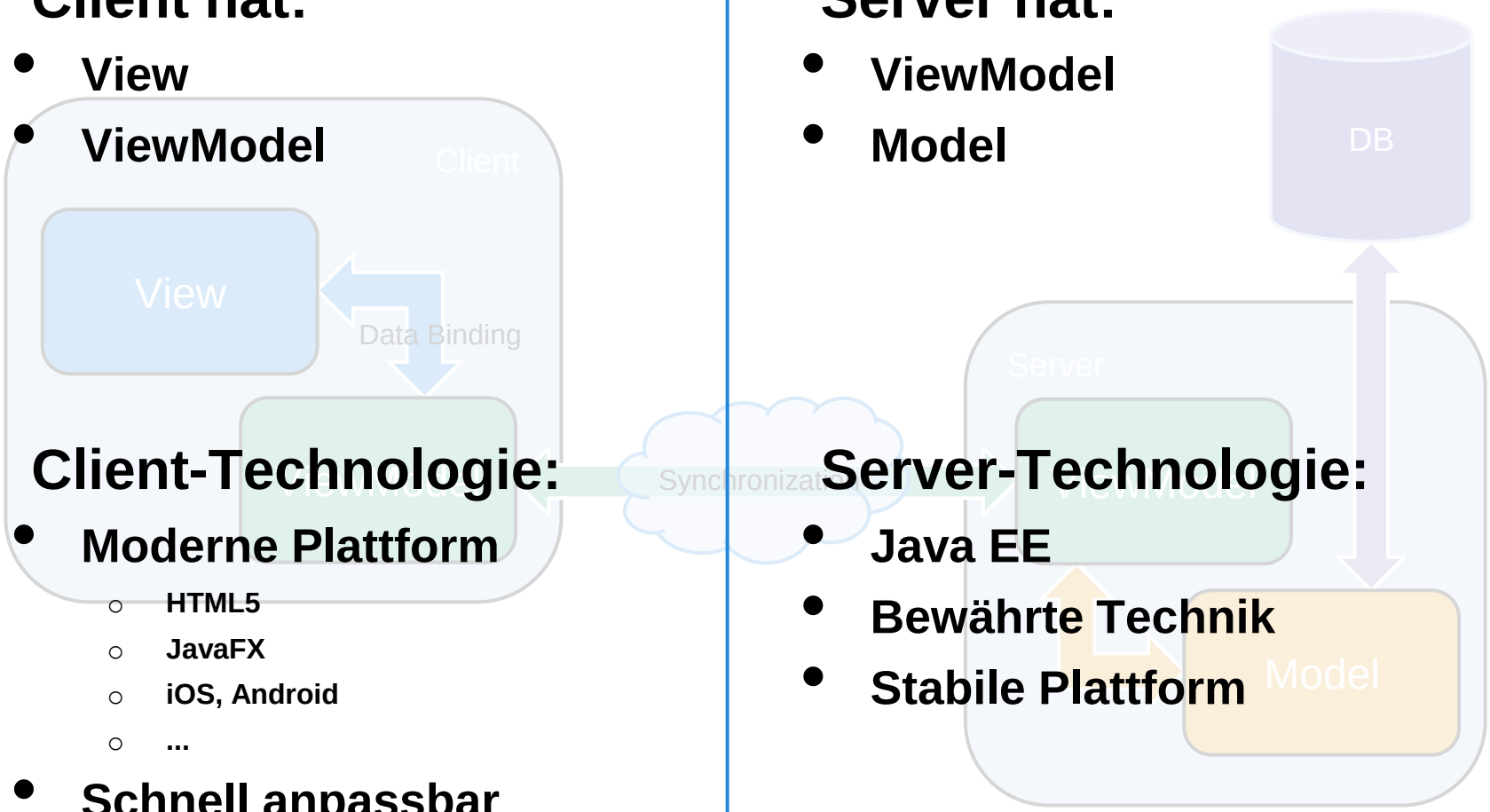
MVSVM - Synchronized ViewModel

Client hat:

- View
- ViewModel

Server hat:

- ViewModel
- Model



Client-Technologie:

- Moderne Plattform
  - HTML5
  - JavaFX
  - iOS, Android
  - ...
- Schnell anpassbar

Server-Technologie:

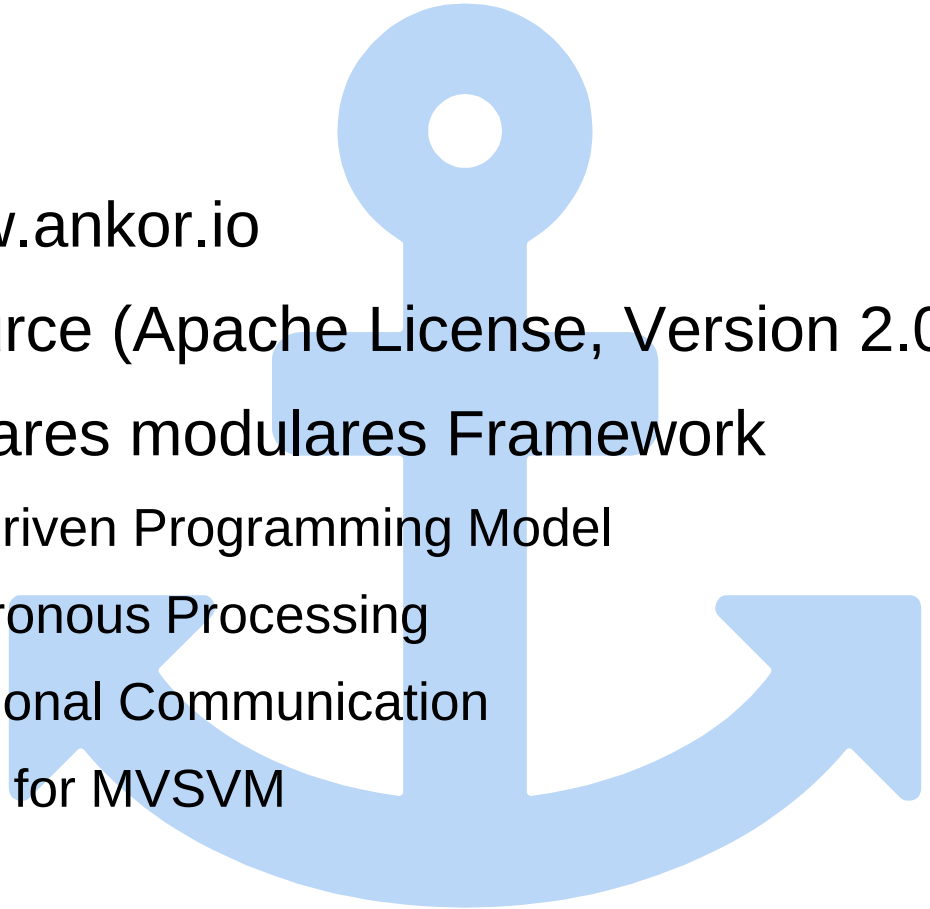
- Java EE
- Bewährte Technik
- Stabile Plattform

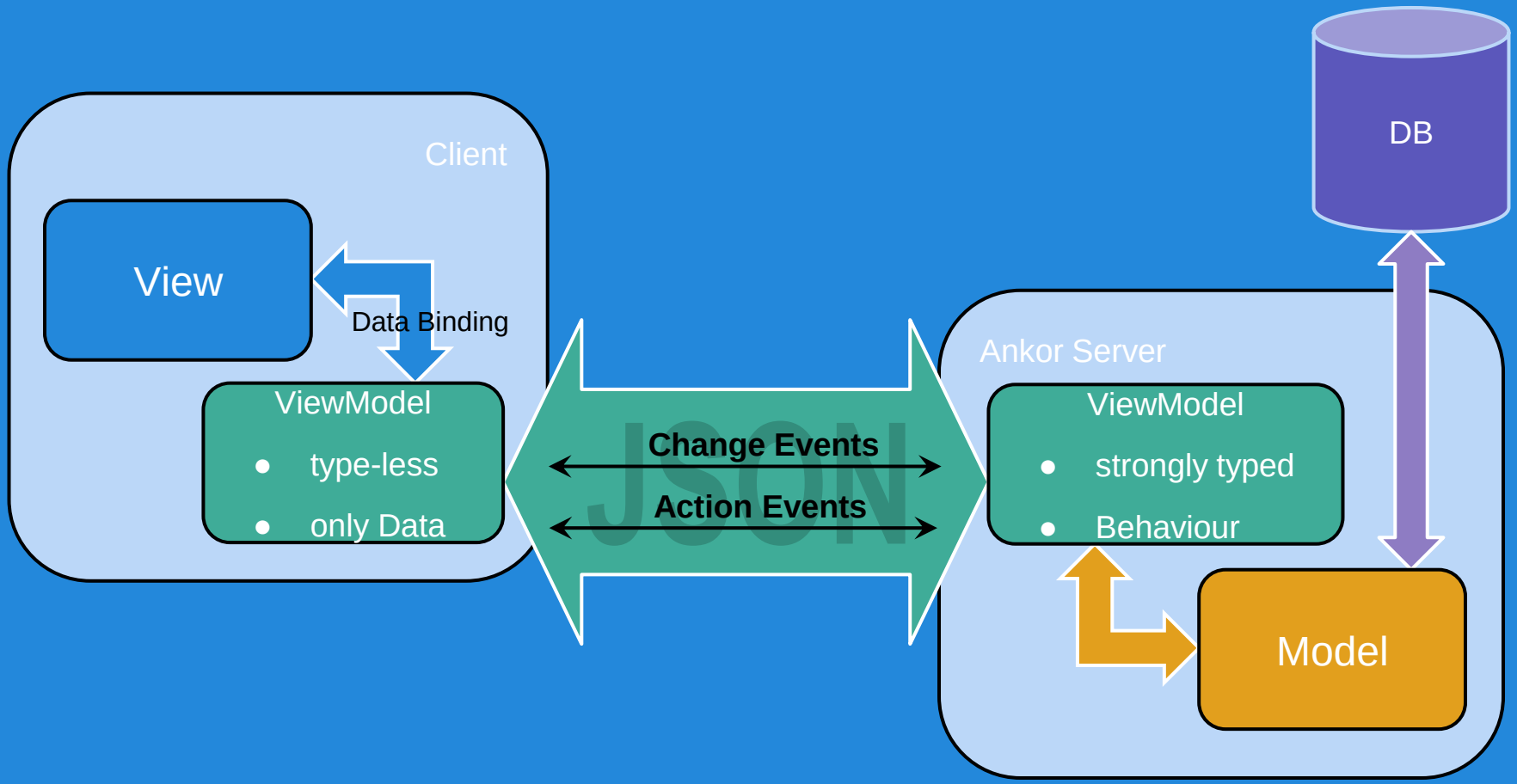
Agenda

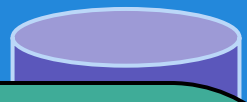
- M V V M
- Neues Konzept MVSVM
- Ankor Framework
- Ankor Live Sample
- Ankor Special Features

Projekt “Ankor”

- 2013
- http://www.ankor.io
- Open Source (Apache License, Version 2.0)
- Erweiterbares modulares Framework
  - Event Driven Programming Model
  - Asynchronous Processing
  - Bidirectional Communication
  - Support for MVSVM




DB
Client
View
Data Binding
ViewModel
type-less
only Data
JSON
Change Events
Action Events
Ankor Server
ViewModel
strongly typed
Behaviour
Model



ViewModel (client side)

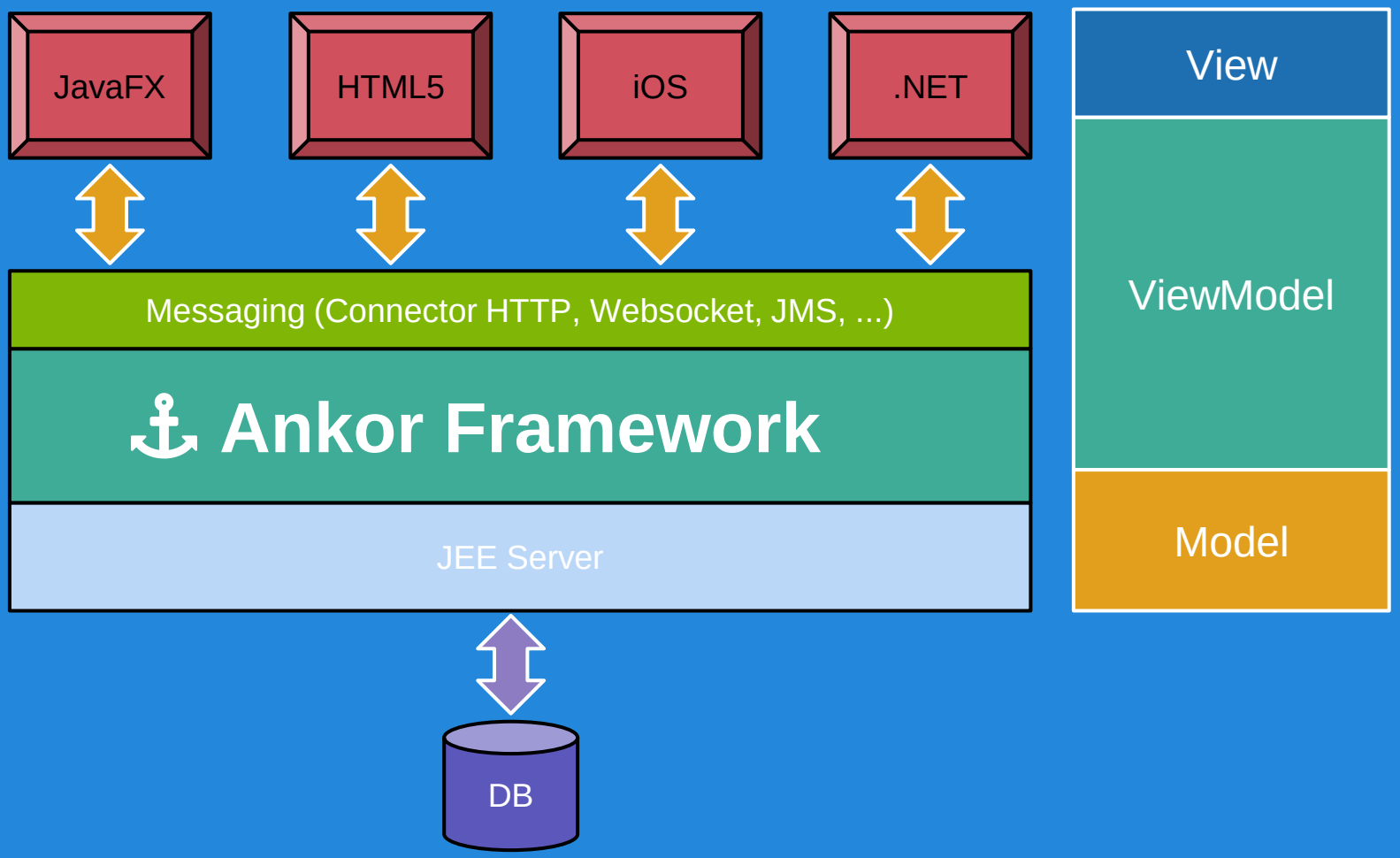
- type-less
- only Data

```
{
  "tasks": [
    {"id": "dda6f7d9-8d5e-4baf-969b-
     "title": "drink less coffee",
     "completed": false},
    {"id": "ff202f81-33b8-4ae3-bf6a-
     "title": "get more sleep",
     "completed": false}
  ],
  "filter": "all",
  "itemsLeft": 2
}
```

ViewModel (server side)

- strongly typed
- Behaviour

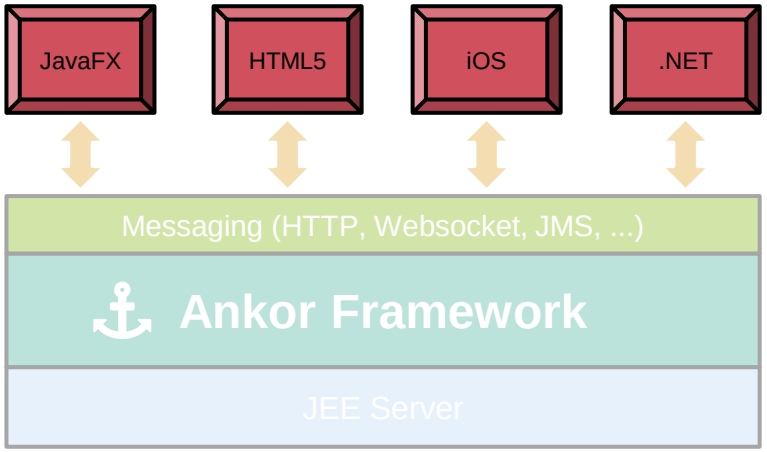
```
public class TaskListViewModel {
    private List<Task> tasks;
    private String filter;
    private Integer itemsLeft;

    // getters and setters
}

public class Task {
    private String id;
    private String title;
    private boolean completed;

    // getters and setters
}
```



Ankor - Überblick Architektur

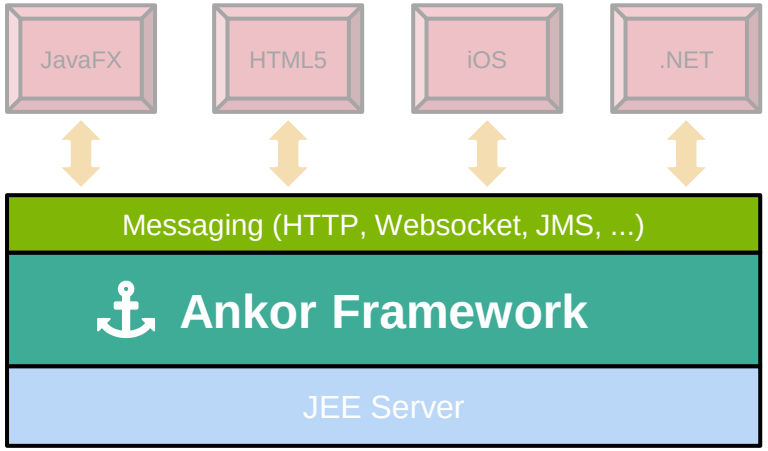
⚓ Ankor - Architektur / Client

- Diverse Client-Sprachen und -Plattformen
  - Java
    - JavaFX
    - Android
  - Javascript / HTML5
    - jQuery
    - AngularJS
    - React
  - Objective-C
    - iOS
  - C#
    - .NET / WPF



⚓ Ankor - Architektur / Server

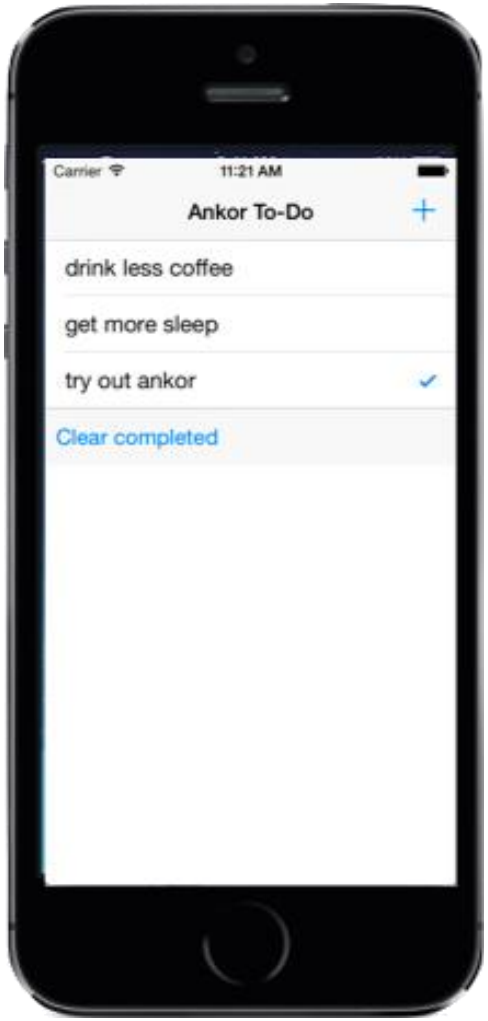
- Java SE 1.6 (oder höher)
- Diverse Netzwerk-Protokolle
  - Socket
  - HTTP-Polling
  - **Websocket**
- Messaging
  - **JSON**
- Concurrency / Parallelisierung
  - Simple Synchronization
  - **Akka Actors**
- JEE Container
  - Glassfish (Websocket)
  - Tomcat
  - Spring Boot (embedded Tomcat)



Agenda

- M V V M
- Neues Konzept MVSVM
- Ankor Framework
- Ankor Live Sample
- Ankor Special Features

Ankor - iOS Example



⚓ Ankor - iOS Example

Start Ankor System, connect to server

```
[[[ANKSystem alloc]
    initWith:@"root" connectParams:connectParams
    url:@"ws://localhost:8080/websocket/ankor"
    useWebsocket:YES] start];
```

⚓ Ankor - iOS Example

Register Change Listener

```
[ANKRefs observe:@"root.model.tasks"
   target:self listener:@selector(tasksChanged:)];
```

Change Listener

```
- (void)tasksChanged:(id) value {
    [[self toDoItems]removeAllObjects];
    [[self toDoItems]addObjectsFromArray:value];
    [self.tableView reloadData];
}
```

⚓ Ankor - iOS Example

Fire Action / Add a new Task

```
[ANKRefs fireAction:@"root.model"
    name:@"newTask"
    params:params]; // Dictionary
```

ActionListener Java

```
@ActionListener
public void newTask(@Param("title") final String title)
{...}
```

Agenda

- M V V M
- Neues Konzept MVSVM
- Ankor Framework
- Ankor Live Sample
- Ankor Special Features

⚓ Ankor - Special Features

- **Ankor Annotations**
- Ankor Big Collections
  - BigList
  - BigMap
- Ankor Flood Control
- Pluggable Bean-Factory
  - Simple/Reflection
  - Spring (geplant)
  - CDI (geplant)
- Collaboration Support
- Stateless Model

Server

```
public class AnimalSearchViewModel {

  private AnimalSearchFilter filter;

  @ChangeListener(pattern = {".filter.**"})
  public void reloadAnimals() {
    this.animals
      = animalRepository.searchAnimals(filter);
  }
```

Client

⚓ Ankor - Special Features

- Ankor Annotations
- **Ankor Big Collections**
  - **BigList**
  - **BigMap**
- Ankor Flood Control
- Pluggable Bean-Factory
  - Simple/Reflection
  - Spring (geplant)
  - CDI (geplant)
- Collaboration Support
- Stateless Model

Server

```
public class AnimalSearchViewModel {

  private List<Animal> animals
    = new ArrayList<Animal>(10000);

  @AnkorBigList(chunkSize = 10)
  public List<Animal> getAnimals() {
    return animals;
  }
```

send 10 rows at once -**on demand only!**

Client

⚓ Ankor - Special Features

- Ankor Annotations
- Ankor Big Collections
  - BigList
  - BigMap
- **Ankor Flood Control**
- Pluggable Bean-Factory
  - Simple/Reflection
  - Spring (geplant)
  - CDI (geplant)
- Collaboration Support
- Stateless Model

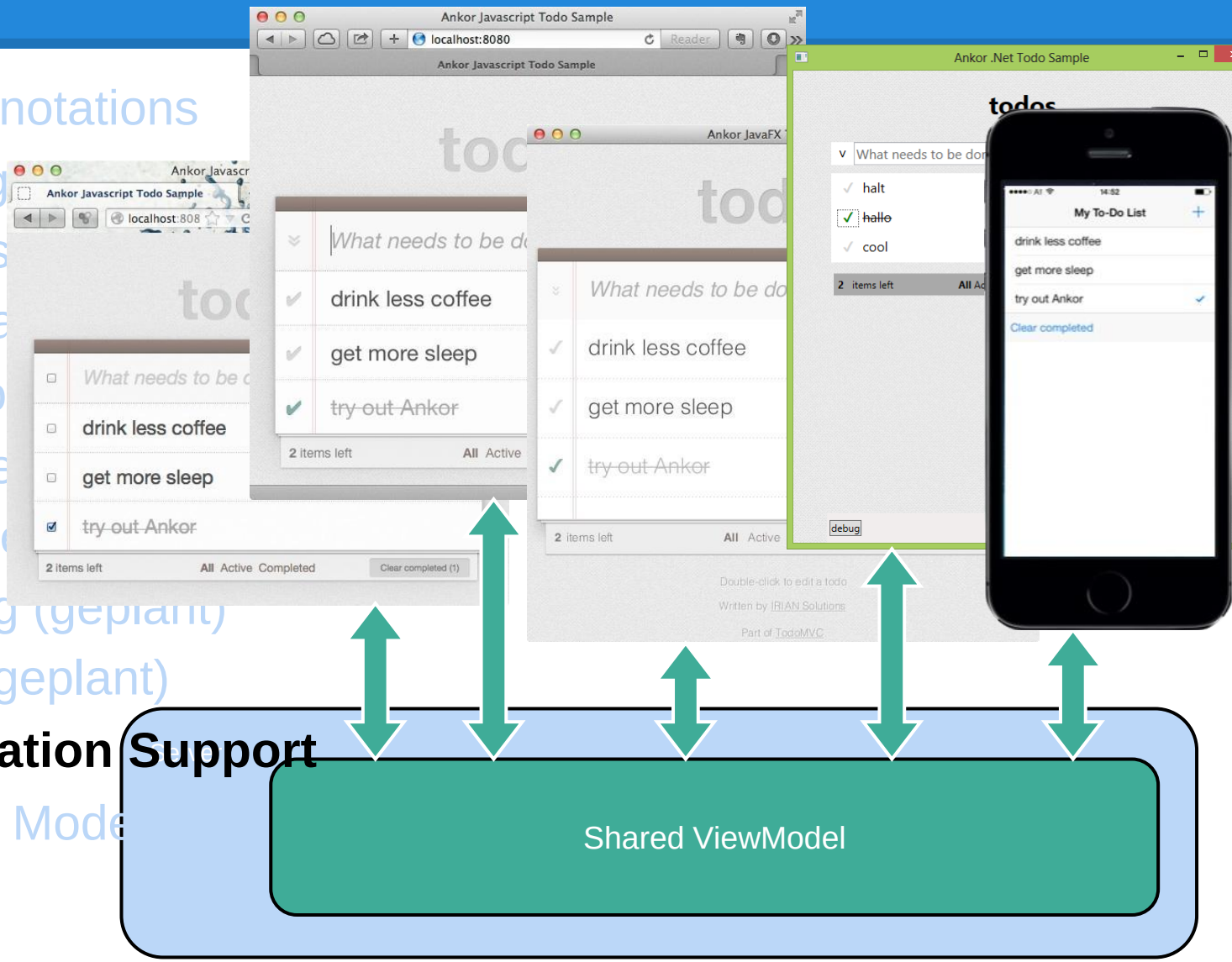
Server

```
public class AnimalSearchViewModel {

  @ChangeListener(pattern = {".filter.**"})
  @AnkorFloodControl(delayMillis = 100L)
  public void reloadAnimals() {
    this.animals
      = animalRepository.searchAnimals(filter);
  }
```

Client

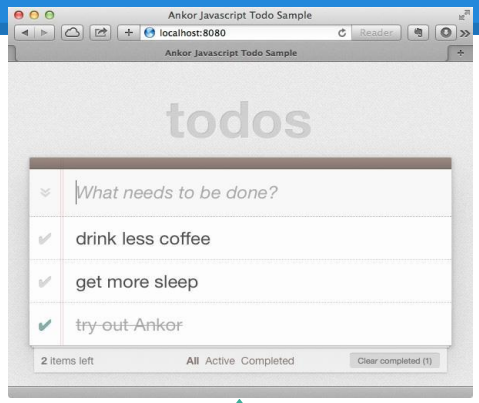
⚓ Ankor - Special Features

- Ankor Annotations
- Ankor Big
  - BigLis
  - BigMa
- Ankor Flo
- Pluggable
  - Simpl
  - Spring (geplant)
  - CDI (geplant)
- **Collaboration Support**
- Stateless Mode



⚓ Ankor - Special Features

- Ankor Annotations
- Ankor Big Collections
  - BigList
  - BigMap
- Ankor Flood Control
- Pluggable Bean-Factory
  - Simple/Reflection
  - Spring (geplant)
  - CDI (geplant)
- Collaboration Support
- **Stateless Model**



send state information

Server

```
public class TaskListModel {

  @StateHolder
  private Filter filter;

  public void setFilter(String filter) {
     this.filter =
       Filter.valueOf(filter);

     initCalculatedFields();
  }
```

Ankor.io

Weiterführenden Informationen & Tutorials

http://ankor.io
http://github.com/ankor-io

Thomas Spiegl
Manfred Geiler