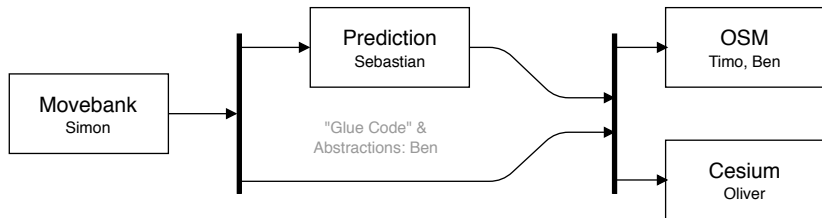


## Group 1.2 - Position Prediction

# Workflow so far



# What we have achieved so far - Design

## Milestone 1 (15.5.)

- ▶ Grundliegende Software-Architektur festgelegt
- ▶ Grundlegendes Software-Design festgelegt
- ▶ Ein Grundgerüst der Applikation ist fertiggestellt: Es sind folgende Ansichten funktional umgesetzt:
  - ▶ Offline-Karten (eine Kartenumgebung wird angezeigt, Navigation möglich)
  - ▶ Diverse UI-Elemente (mindestens Eingabefeld, Button)
- ▶ Die Positionsdaten für die Vorhersage werden von der *Movebank*-API bezogen.

## Milestone 2 (5.6.)

- ▶ Es wird eine Vorhersage mittels eines einfachen Vorhersagemodells berechnet.
- ▶ Es existiert eine funktionale Visualisierung des Vorhersage-Ergebnisses.
- ▶ Es existiert eine Möglichkeit, eine Cesium Ansicht von der Applikation aus aufzurufen.
- ▶ Der Aufbau des User-Interfaces ist festgelegt

# What we have achieved so far - Software Architecture

## Request Prediction

### (User Interaction)

Parameter choice (Settings)

- Algorithm
- Algorithm Parameters
- Visualisation Parameters

### Activity

- obtain Adapter
- link Mapview
- handle Errors
- Communication with User

### WorkflowController

- 1.) Data Fetch
- 2.) Call Algo
- 3.) Determine Visualisation
- 4.) Set Visualisation Properties (colour, etc)
- 5.) Hand abstract Visualisation to Adapter

### CesiumAdapter

Cesium Control  
Cesium Library

### OSMDroidAdapter

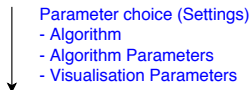
OSMDroid Control  
OSMDroid Library

### Benefits:

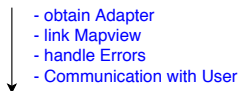
- ▶ Easier to write: Controller doesn't care about Algo or Map implementation
- ▶ Easier to integrate (call from anywhere)
- ▶ Highly extensible (could subclass controllers?)
- ▶ Same code for visualisation of past and predicted locations  
(assumption: never want to see data without a prediction → AnimalTracker)

# What we have achieved so far - Software Architecture

## Request Prediction (User Interaction)



## Activity



## WorkflowController

- 1.) Data Fetch
- 2.) Call Algo
- 3.) Determine Visualisation
- 4.) Set Visualisation Properties (colour, etc)
- 5.) Hand abstract Visualisation to Adapter



## CesiumAdapter

Cesium Control  
Cesium Library

## OSMDroidAdapter

OSMDroid Control  
OSMDroid Library

## Challenges:

- ▶ Separation of Concerns, D.R.Y.
- ▶ How to model data in a general but still meaningful, usable, maintainable way?
  - ▶ e.g. Types  
`Locations > SingleTrajectory,`  
`SingleTrajectoryVis,`  
`AlgParams`

# What we have achieved so far - Software Architecture

## Request Prediction (User Interaction)

- Parameter choice (Settings)
- Algorithm
- Algorithm Parameters
- Visualisation Parameters

## Activity

- obtain Adapter
- link Mapview
- handle Errors
- Communication with User

## WorkflowController

- 1.) Data Fetch
- 2.) Call Algo
- 3.) Determine Visualisation
- 4.) Set Visualisation Properties (colour, etc)
- 5.) Hand abstract Visualisation to Adapter

## CesiumAdapter

Cesium Control  
Cesium Library

## OSMDroidAdapter

OSMDroid Control  
OSMDroid Library

- ▶ **OSMDroidAdapter**: Convert data to OSM-specific Types and call the right drawing methods.
- ▶ **OSMDroidMap**: Library specific code
  - ▶ enable features
  - ▶ draw stuff
  - ▶ handle events
  - ▶ make undocumented library functions usable

# Next steps - Software Architecture

How to integrate...

- ▶ ... error handling? (throw where, catch where?)
- ▶ ... user communication? (error messages, progress bars...)
- ▶ Achieve higher generality in `WorkflowController`
- ▶ ... see project timeline.

## Milestone 1 (15.5.)

- ▶ Grundliegende Software-Architektur festgelegt
- ▶ Grundliegendes Software-Design festgelegt
- ▶ Ein Grundgerüst der Applikation ist fertiggestellt: Es sind folgende Ansichten funktional umgesetzt:
  - ▶ Offline-Karten (eine Kartenumgebung wird angezeigt, Navigation möglich)
  - ▶ Diverse UI-Elemente (mindestens Eingabefeld, Button)
- ▶ Die Positionsdaten für die Vorhersage werden von der *Movebank-API* bezogen.



# What we have achieved so far - Database

- ▶ CSV-Files from the Movebank are parsed and the data is put into the Database.
- ▶ Data can be accessed using SQL queries

# What we have achieved so far - Algorithms I

## Milestone 2 (5.6.)

- ▶ Es wird eine Vorhersage mittels eines einfachen Vorhersagemodells berechnet.
- ▶ Es existiert eine funktionale Visualisierung des Vorhersage-Ergebnisses.
- ▶ Es existiert eine Möglichkeit, eine Cesium Ansicht von der Applikation aus aufzurufen.
- ▶ Der Aufbau des User-Interfaces ist festgelegt (nicht-funktionale Mockups).

## AlgorithmExtrapolationExtended

- (+) Good if the variance of the angles is not too big
- (+) Later datapoints are weighted more
- (+) Fast
- (+) Easy to understand
  
- ( - ) Not very accurate
- ( - ) Early data gets ignored

## AlgorithmSimilarTrajectory

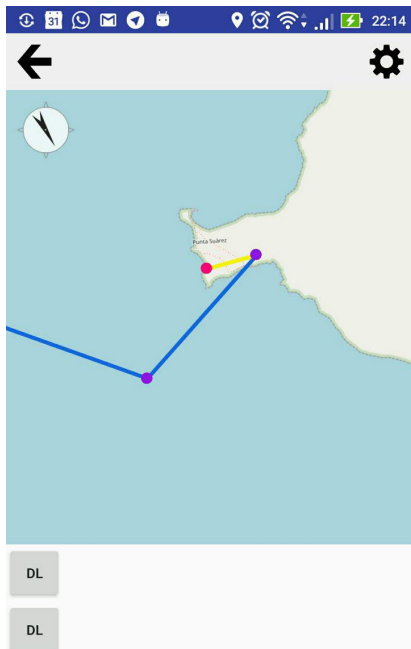
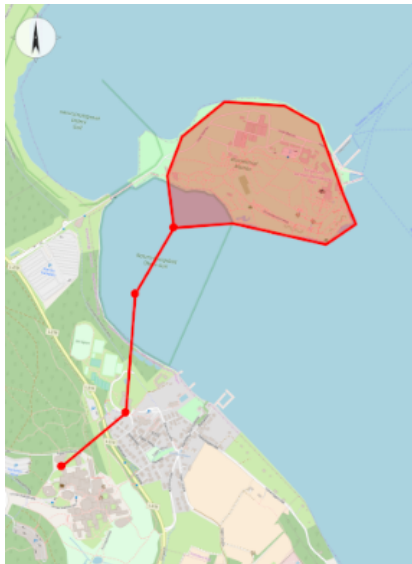
- (+) Good if the measuring frequency is high
- (+) Earlier datapoints are important for the result
- (+) Multiple trajectories can be found
- (+) Easy to understand
  
- ( - ) Frequency is not always high  $\Rightarrow$  Wrong result
- ( - ) Higher complexity than the other algorithm
- ( - ) (Currently) only works with the same time span between datapoints

# What we have achieved so far - Visualization (OSM) I

## Milestone 2 (5.6.)

- ▶ Es wird eine Vorhersage mittels eines einfachen Vorhersagemodells berechnet.
- ▶ Es existiert eine funktionale Visualisierung des Vorhersage-Ergebnisses.
- ▶ Es existiert eine Möglichkeit, eine Cesium Ansicht von der Applikation aus aufzurufen.
- ▶ Der Aufbau des User-Interfaces ist festgelegt (nicht-funktionale Mockups).

# What we have achieved so far - Visualization (OSM) I

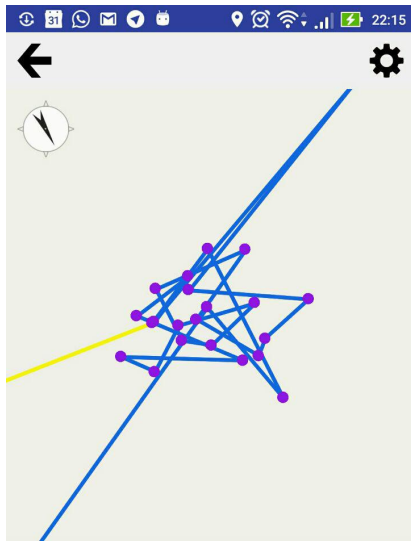
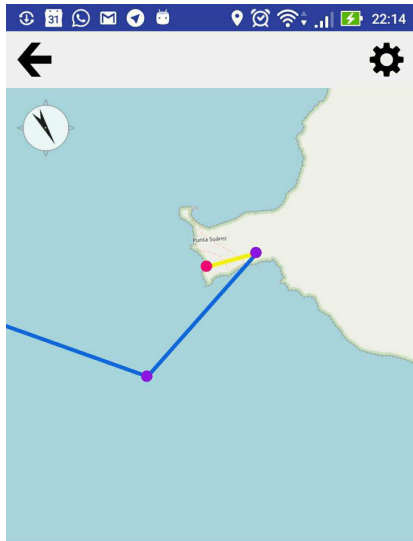


# Next steps - Visualization (OSM) II

- ▶ Design suitable visualizations for different types of prediction output (trajectories, clouds, ...)
- ▶ Improve visualisation readability
  - ▶ On low zoom levels: cluster points without losing information (*cf Example*)
  - ▶ (Idea) Visualise time (e.g. map to point opacity)
  - ▶ (Idea) Visualise speed (e.g. map to line segment colour)
  - ▶ ...
- ▶ Speed improvements
- ▶ Alternative map tile sources

# Next Steps - Visualisation Readability (Example)

Low zoom levels leave out information:





# What we have achieved so far - Visualization (Cesium)

## Milestone 2 (5.6.)

- ▶ Es wird eine Vorhersage mittels eines einfachen Vorhersagemodells berechnet.
- ▶ Es existiert eine funktionale Visualisierung des Vorhersage-Ergebnisses.
- ▶ Es existiert eine Möglichkeit, eine Cesium Ansicht von der Applikation aus aufzurufen.
- ▶ Der Aufbau des User-Interfaces ist festgelegt (nicht-funktionale Mockups).