

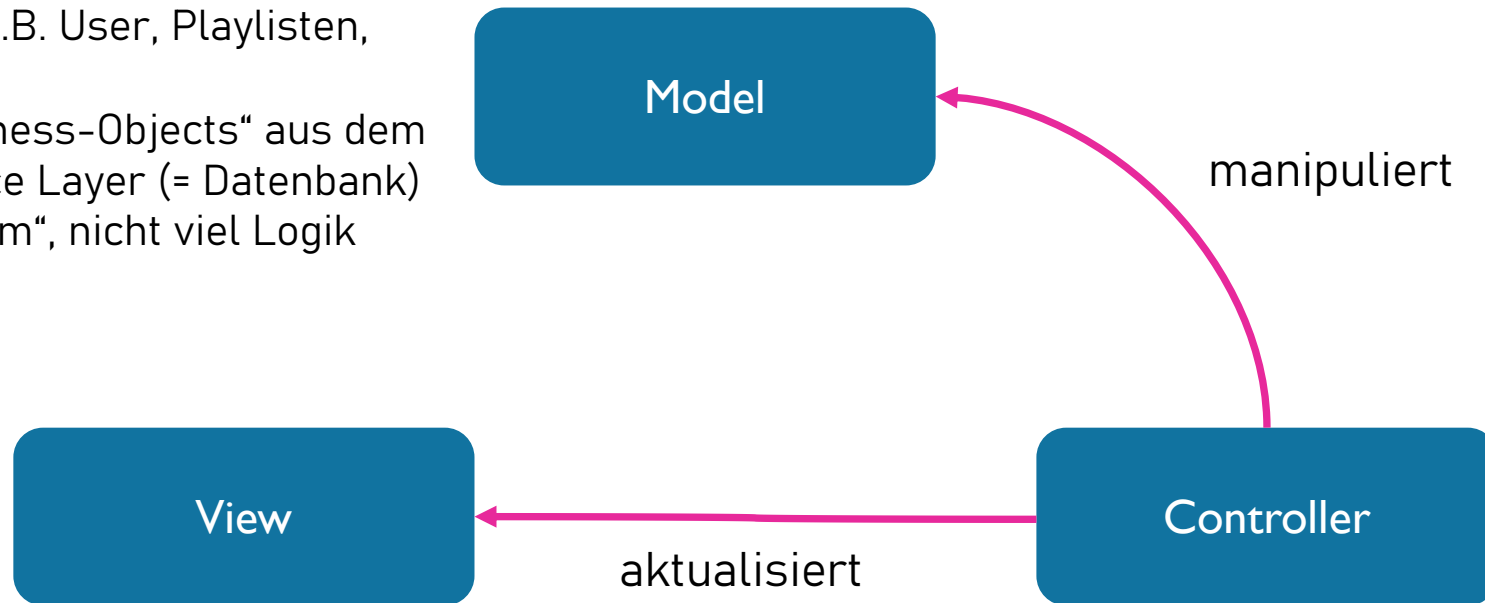
WEBTECHNOLOGIEN

06 – UTILS-PATTERN

PROF. DR. MARKUS HECKNER

RECAP: MODEL VIEW CONTROLLER IST EIN WEIT VERBREITETES SOFTWARE-ENTWURFSMUSTER

- Persistiert Daten aus „Business-Objects“ (z.B. User, Playlisten, Songs, ...)
- Holt „Business-Objects“ aus dem Persistence Layer (= Datenbank)
- Eher „dumm“, nicht viel Logik



- Holt Daten aus den Models
- Speichert Daten in den Models
- Fügt Daten in die Views ein und senden Daten diese an Client zurück

MANCHE FUNKTIONALITÄTEN PASSEN NICHT SOEINFACH ZU CONTROLLERN UND MODELS HINZUFÜGEN

Beispiele:

- Logging – Welcher Controller soll das Logging übernehmen? Es gibt keinen View für das Logging ...
- Analytics-Funktionen für Playlisten – Playlist Controller könnte die Funktionalität übernehmen, aber irgendwann unübersichtlich: Viele Methoden, lange Methoden, ...

Lösung: Auslagern von Funktionalität in eigene Module!

UTILS-PATTERN: PLAYLIST-ANALYTICS

utils/playlist-analytics.js

```
const playlistAnalytics = {  
  getShortestSong(songs) {  
    let shortestSong = null;  
    if (songs.length > 0) {  
      shortestSong = songs[0];  
      for (let i = 1; i < songs.length; i++) {  
        if (songs[i].duration < shortestSong.duration) {  
          shortestSong = songs[i];  
        }  
      }  
    }  
    return shortestSong;  
  },  
};  
  
module.exports = playlistAnalytics;
```

Liste an Songs aus dem Controller
wird übergeben

Iterieren über die Liste und suchen
des kürzesten Songs

Rückgabe des kürzesten Songs
(ohne Interaktion mit dem Model)