

## Mini-Projekt

Thema:

**Vornamen (der aktuellen Zürcher Wohnbevölkerung)**

CAS Data Science Applications, ZHAW, Modul Visualisierung

Martin Weber, [martin@bitweber.ch](mailto:martin@bitweber.ch)

# Geschichte / Aussage

## **Fragestellungen**

- Such nach einem Vornamen
- Typische Frauen-/Männernamen
- Heißen wirklich so viele Kinder „Kevin“? Jahrgang?
- Alt-/neumodische Namen → Namenstrends

# Daten Akquisition

## **Datensatz: Vornamen der aktuellen Wohnbevölkerung der Stadt Zürich**

- Anzahl Personen der wirtschaftlichen Wohnbevölkerung der Stadt Zürich nach Vorname, Geschlecht und Jahrgang.
- In diesem Datensatz sind nicht alle Vornamen der Stadt Zürich zu finden. Pro Geschlecht und Vornamen müssen mindestens 10 Fälle vorkommen.
- Quelle: Open Data Katalog der Stadt Zürich  
<https://data.stadt-zuerich.ch/dataset/bev-bestand-vornamen-jahrgang-geschlecht>

Übrigens, den Datensatz gibt es auch noch für Hundenamen... (noch nicht implementiert ;-):

<https://data.stadt-zuerich.ch/dataset/pd-stapo-hundenamen>

# Datenkonsolidierung

## (Verdichtung/ Aufbereitung)

- Daten liegen bereits schon vorverdichtet vor:  
*In diesem Datensatz sind nicht alle Vornamen der Stadt Zürich zu finden. Pro Geschlecht und Vornamen müssen mindestens 10 Fälle vorkommen.*
- Aufbereitung für Analyse:
  - Zeichensatz von CSV umwandeln vor Import
  - Import in SQL-DB für Analyse

# Visualisierung

- Häufigste Namen
  - Häufigkeit in Zeitraum
  - Geschlecht

→ Balkendiagramm

→ Balkendiagramm (Total)

→ Farbe

## Top 100 Namen

Geschlecht

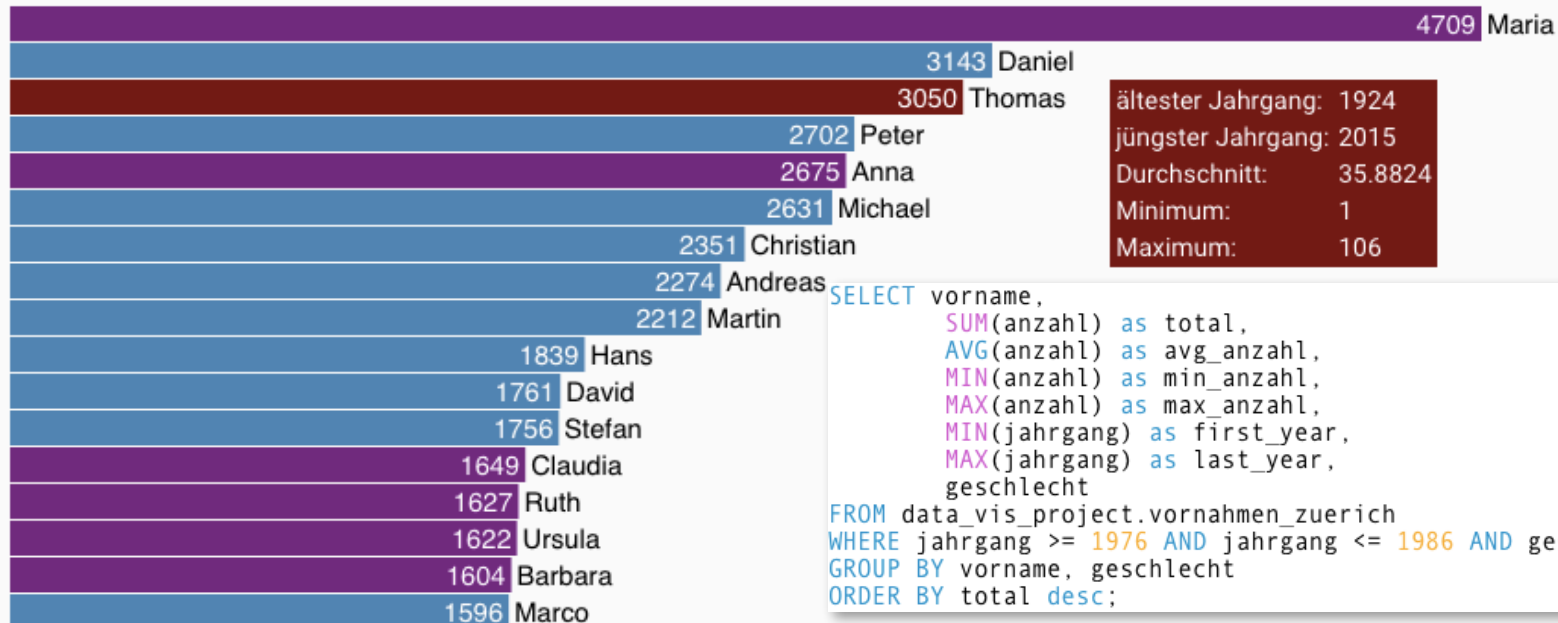
beide

Von Jahrgang \*

▼ 1900

Bis Jahrgang \*

2016



ältester Jahrgang: 1924  
jüngster Jahrgang: 2015  
Durchschnitt: 35.8824  
Minimum: 1  
Maximum: 106

```
SELECT vorname,  
       SUM(anzahl) as total,  
       AVG(anzahl) as avg_anzahl,  
       MIN(anzahl) as min_anzahl,  
       MAX(anzahl) as max_anzahl,  
       MIN(jahrgang) as first_year,  
       MAX(jahrgang) as last_year,  
       geschlecht  
FROM data_vis_project.vornahmen_zuerich  
WHERE jahrgang >= 1976 AND jahrgang <= 1986 AND geschlecht like 'w'  
GROUP BY vorname, geschlecht  
ORDER BY total desc;
```

# Visualisierung

- Namenstrends
  - Jahrgänge
  - Anzahl Namen
  - Vergleich von Namen

→ Liniendiagramm

→ X-Achse

→ Y-Achse

→ Linienfarben

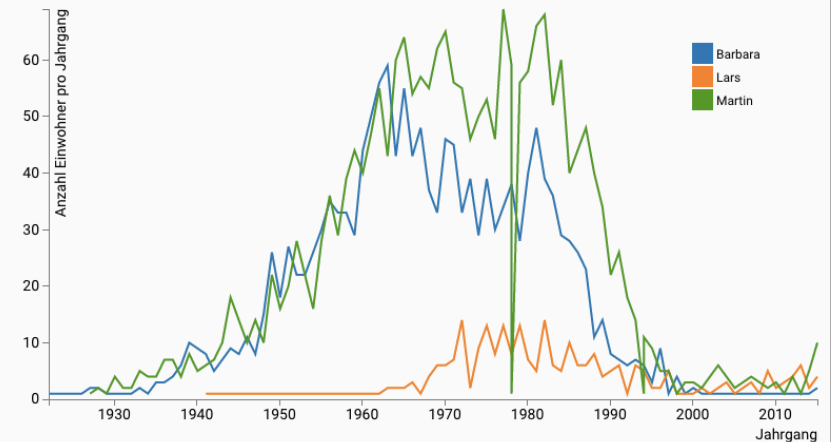
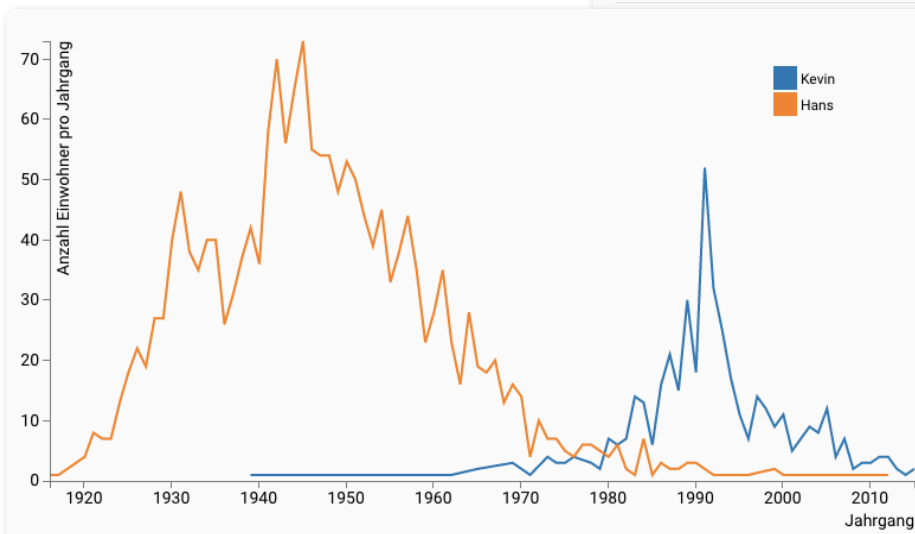
Anzahl Vornamen nach Jahrgang

Geben sie den Namen ein um zu sehen, wieviele Zürcher mit welchem Jahrgang diesen Namen tragen.

Name eingeben

Barbara X Lars X Martin X

+Name



# Information Architecture

## Content structure:

- Übersicht (Aggregation)
- Details (einzelne Namen)
  - Namen: Jahrgang / Anzahl

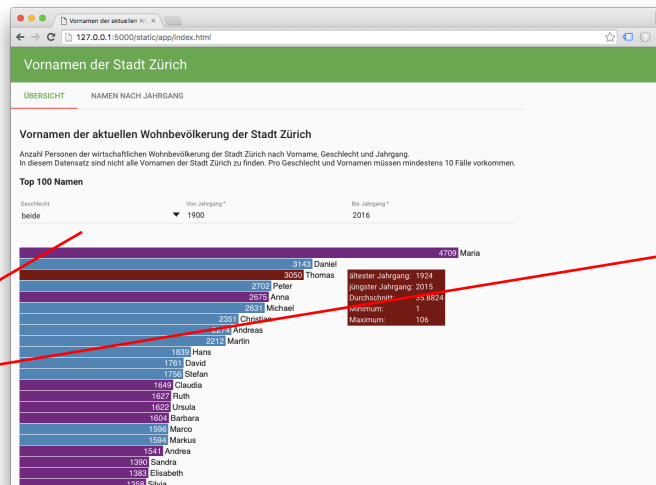
## Navigation System: Tabs

## Information Visualization

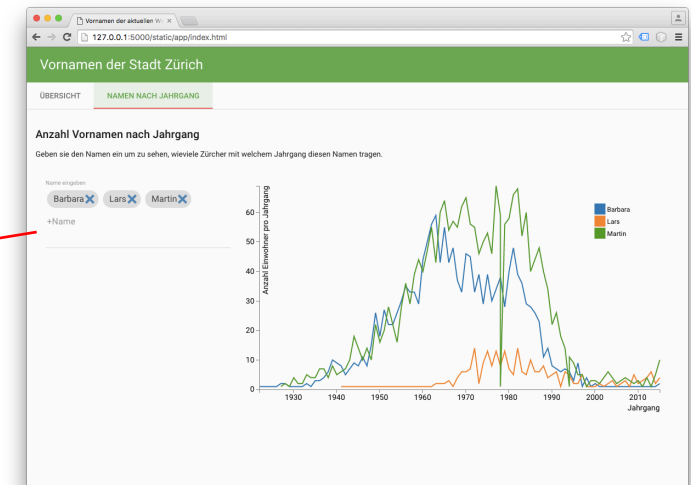
- Barchart für Aggregation
- Line-Chart für Verteilung

## Search:

Filterung  
passend zu  
Diagramm



Übersicht: Aggregierte Daten

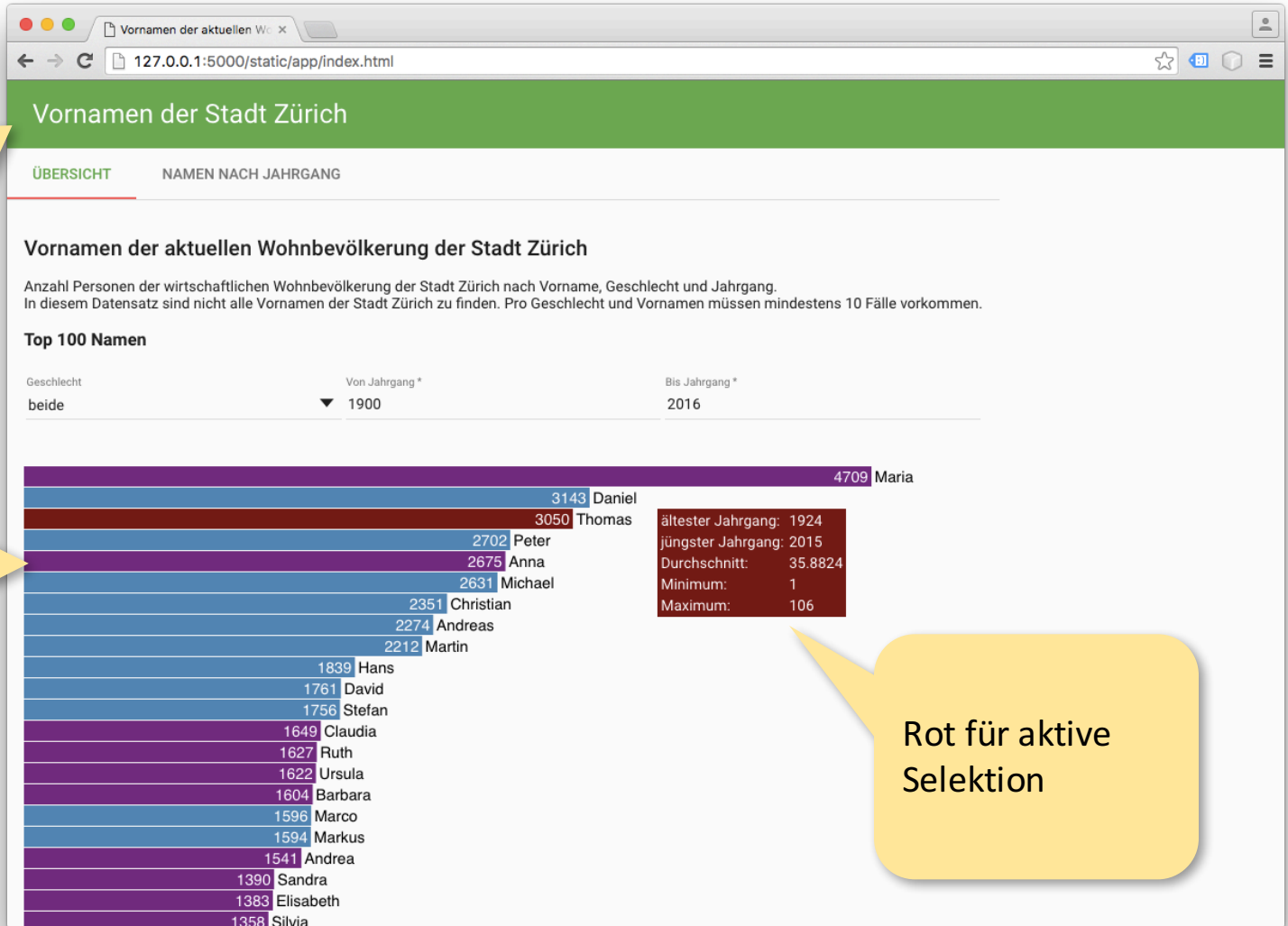


Details einzelne Zeitreihen

# Information Architecture

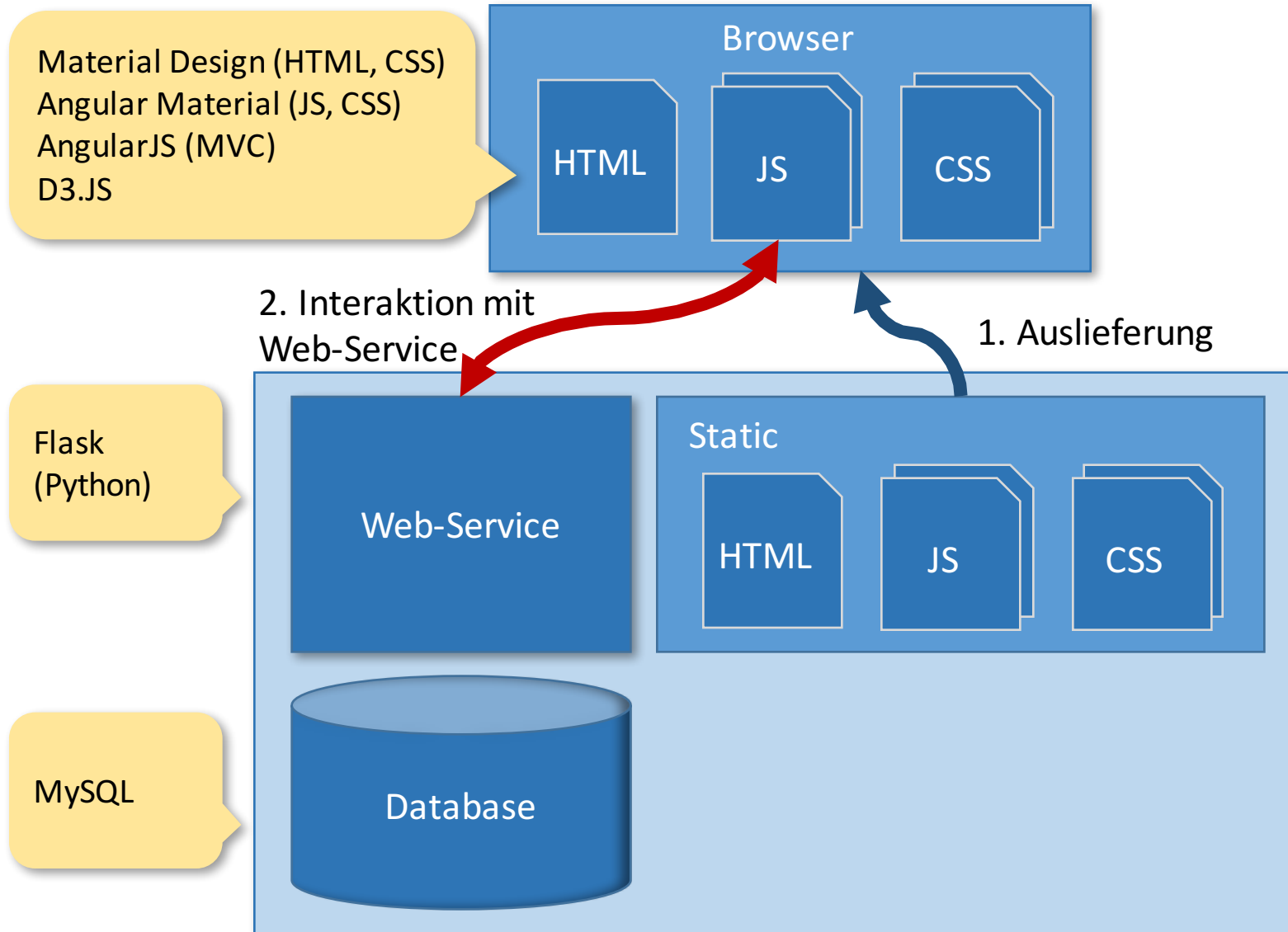
Grün als neutrale Farbe, welche nicht mit einem Geschlecht assoziiert wird

Farben angelehnt an Farbpräferenzen nach Geschlecht





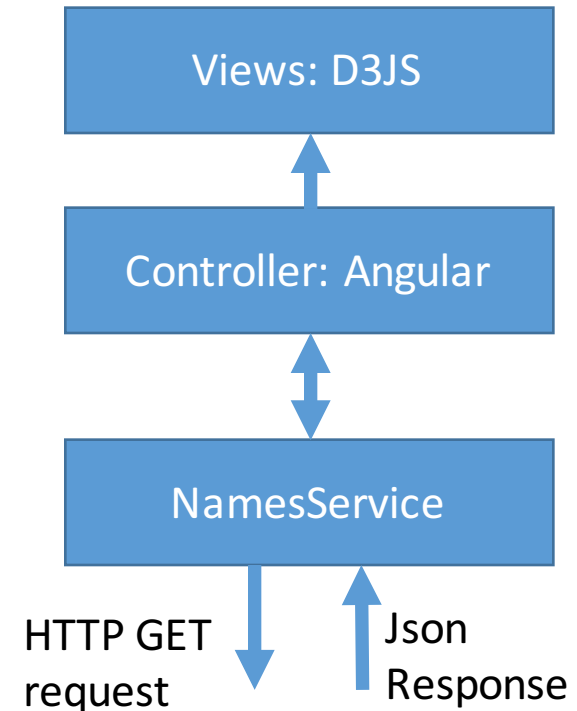
# Systemarchitektur / Technologien



# Client-Architektur

- Single Page Application
- Logik basierend auf [AngularJS](#) Model-View-Controller-Pattern
- Design basierend auf [Material Design](#) von Google mittels AngularMaterial
- Views für Diagramme als Angular Direktiven (Komponenten):

```
<div>  
  <names-line-chart data="chartData"></names-line-chart>  
</div>  
  
<div>  
  <horizontal-bar-chart data="summaryData"  
                        navigate-to="tabsCtrl.selectLineChartTab(name)">  
  </horizontal-bar-chart>  
</div>
```



# Tips & Tricks

- Wartezeit verkürzen
  - Grosse Resultate als Block ausliefern dauert Lange
    - Abfrage → Gesamtergebnis in JSON umwandeln → Dokument ausliefern
  - Besser Streaming der Daten aus der Datenbank an den Client
    - Abfrage → Resultate zeilenweise auslesen → Zeilenweise umwandeln → ausliefern
    - Python:
      - Zeilenweise auslesen aus DB-Cursor
      - Zeile mit yield-Funktion zurückgeben
      - Zeilenweise konvertieren in JSON
      - Ausliefern mittels Response(response = ..., mimetype=„application/json“)
- Animationen um Ladezeit zu „vertuschen“
  - sparsam eingesetzt verkürzt es die empfundene Wartezeit

[illegible]