

Building Individual Data Products for Social Good with Python, PostgreSQL and Vue.js

Content

Motivation	... 4
Why not simply use [...] ?	... 11
Examples from the real world	... 16
The [current] stack	... 20
Alternatives	... 28
How to get started?	... 34
Questions	... 37



Jan Dix

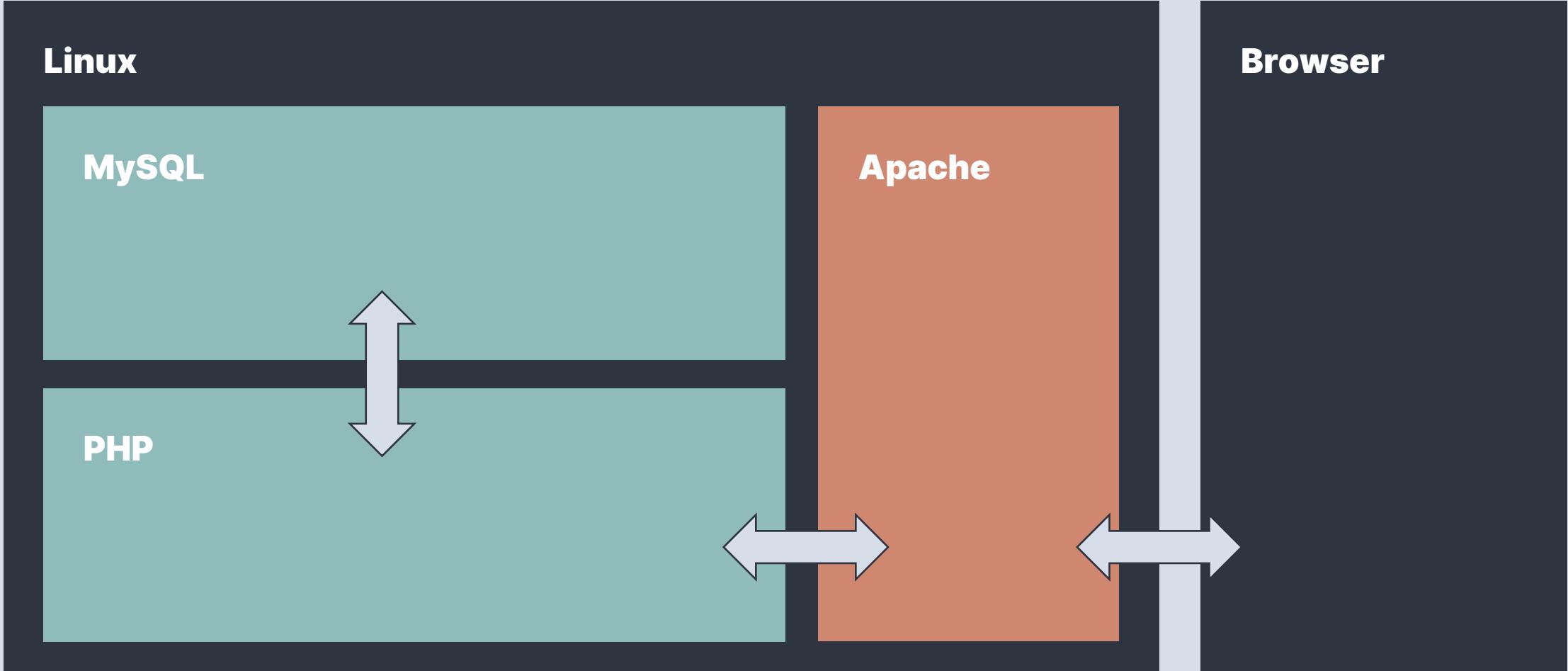
- Co-founder cause&effect
- Founding member of CorrelAid
- Politics student turned data scientist

Twitter: @toll_patsch

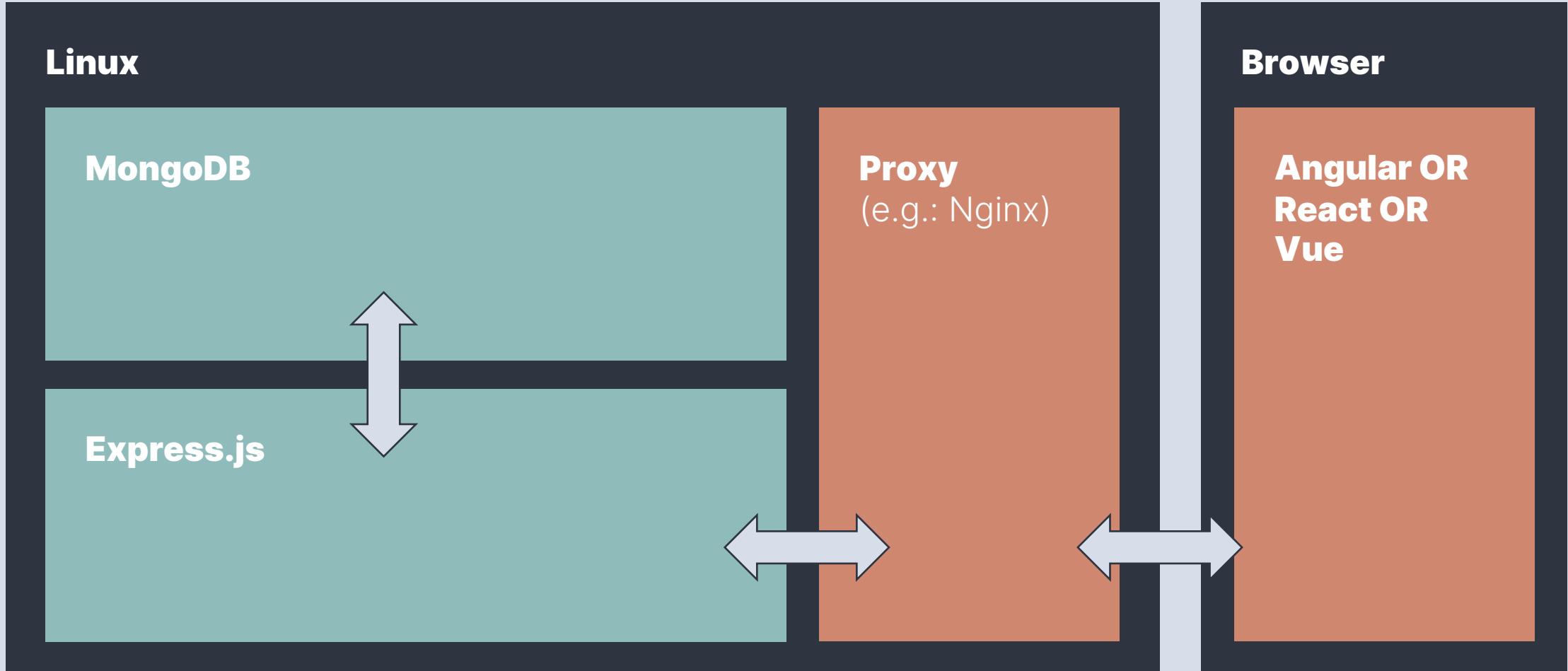
Email: jan.dix@cause-effect.io

Motivation

The LAMP stack



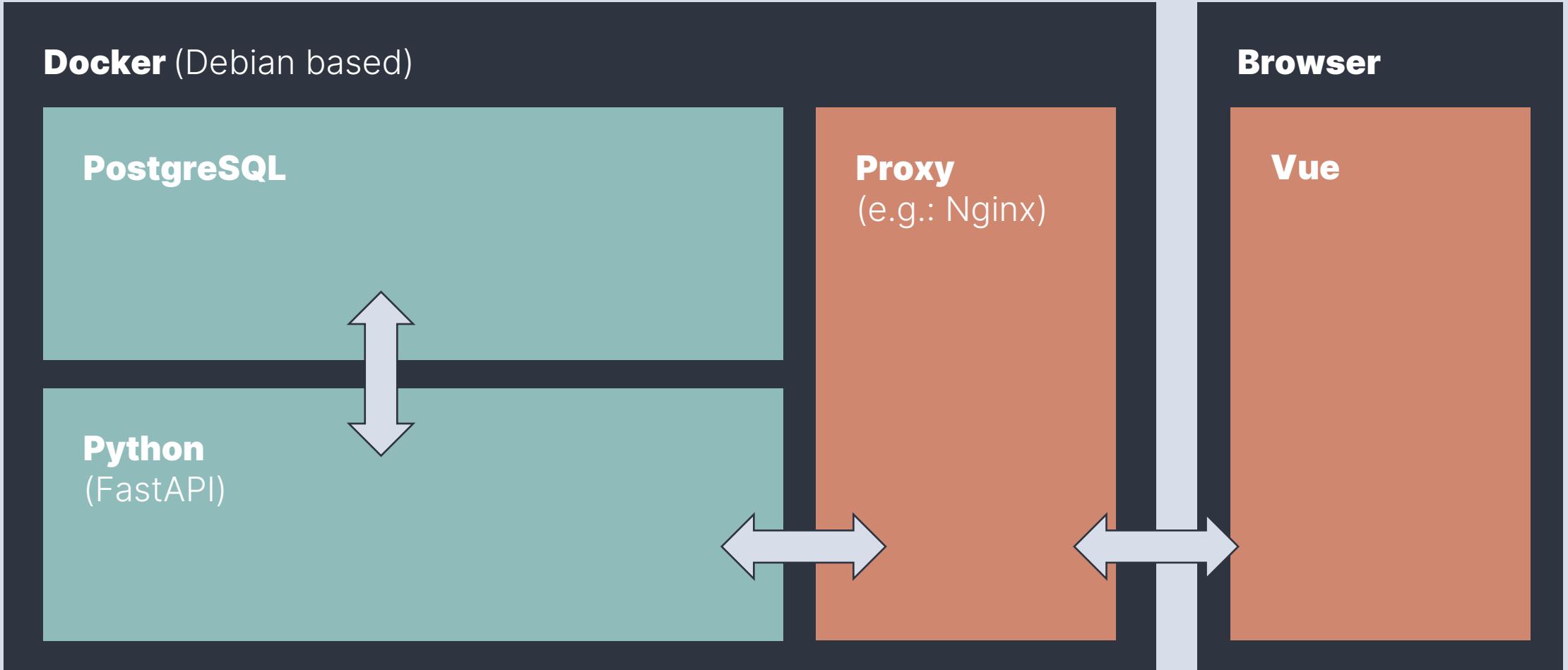
The MEAN/MERN/MEVN stack



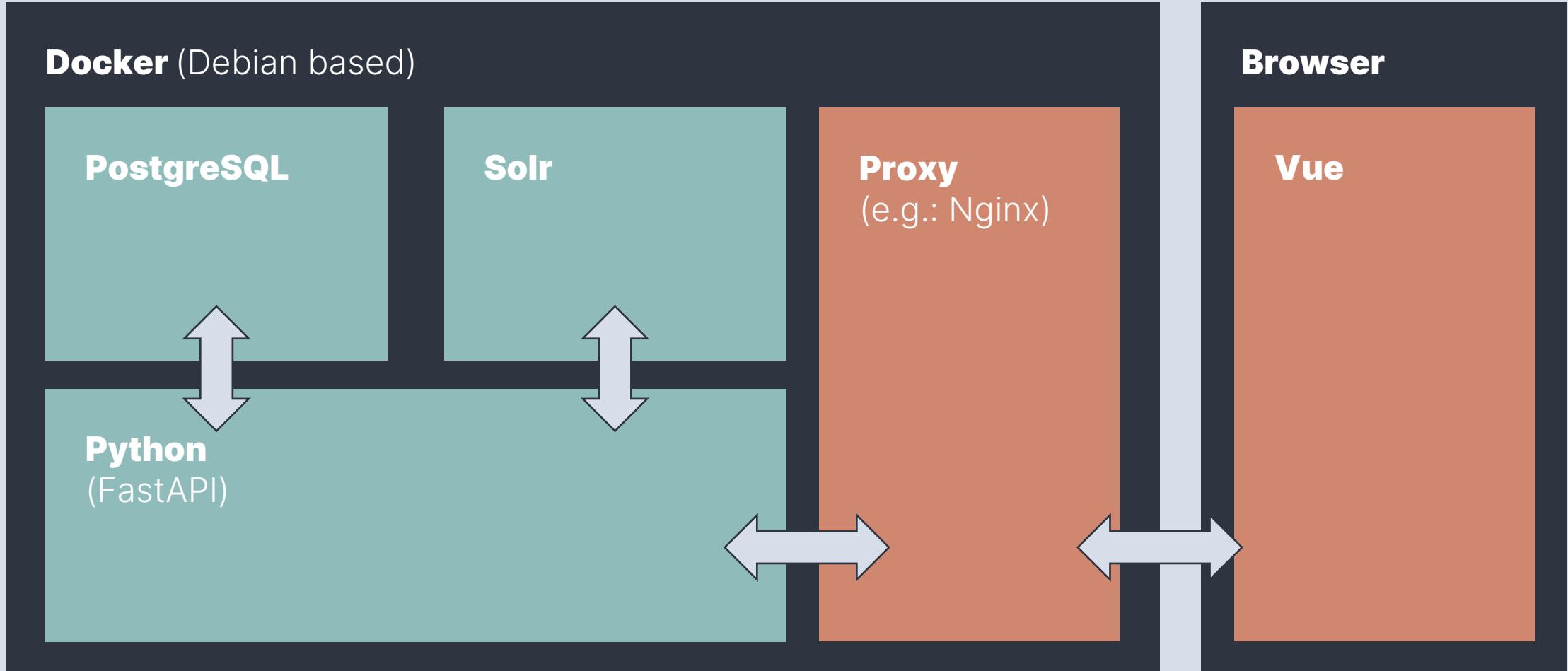
We need a setup that

- 1. is flexible enough to run out of the box**
- 2. allows to be customized when needed**
- 3. is modular enough to be reusable**
- 4. allows to integrate data science technologies**

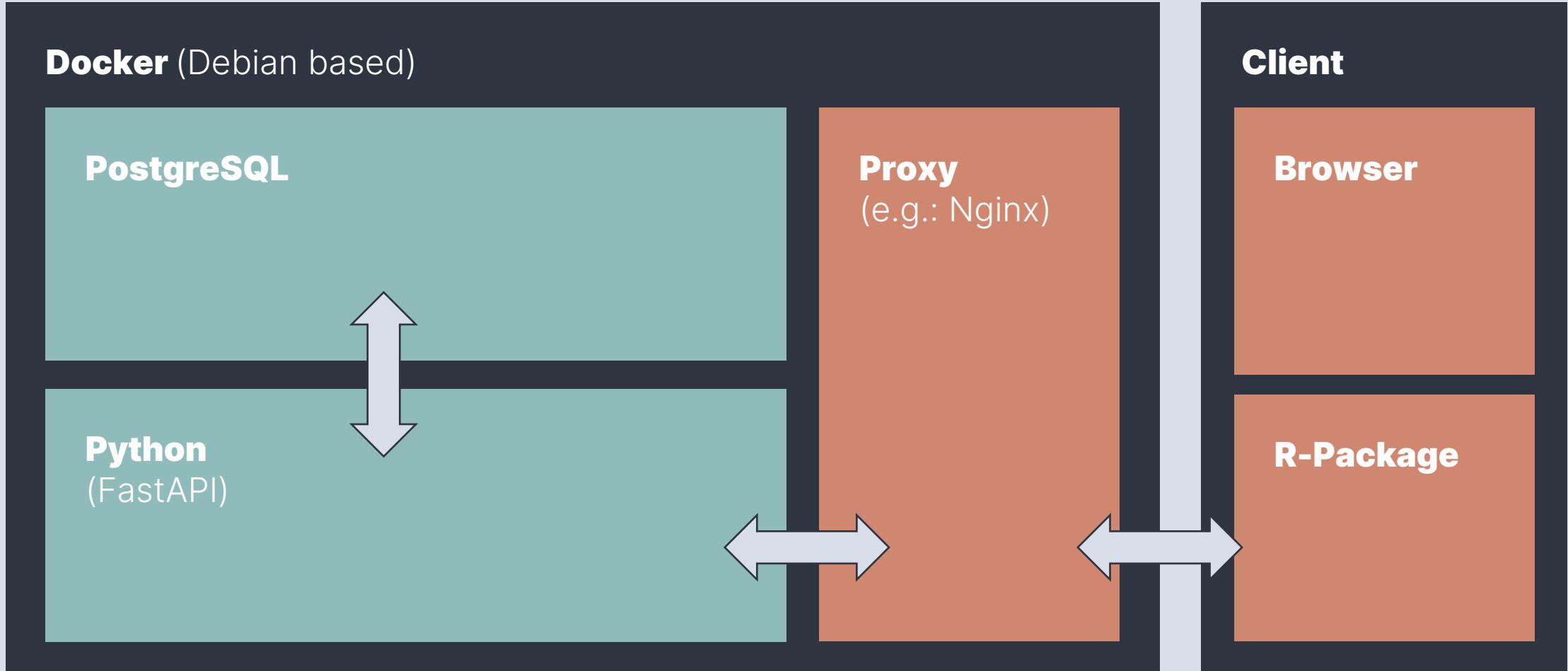
The PPV stack 1



The PPV stack 2



The PPV stack 3



Why not simply use [...] ?

BI Tools

Microsoft Power BI; Tableau; QlikView

- ✓ BI Tools is flexible enough to run out of the box
- ❑ BI Tools allow to integrate data science technologies
- ❑ BI Tools allow to be customized when needed
- ❑ BI Tools are modular enough to be reusable

Shiny

- ✓ Shiny is flexible enough to run out of the box
- ✓ Shiny allows to integrate data science technologies

- ❑ Shiny allows to be customized when needed
- ❑ Shiny is modular enough to be reusable

Dash

- ✓ Dash is flexible enough to run out of the box
- ✓ Dash allows to integrate data science technologies
- ✓ Dash allows to be customized when needed

- Dash is modular enough to be reusable

Neither BI Tools, Shiny nor Dash

- 1. provide a real separation of concerns**
- 2. allow to simply swap a single component when needed**

Examples from the real world

Level 2 Visualisations

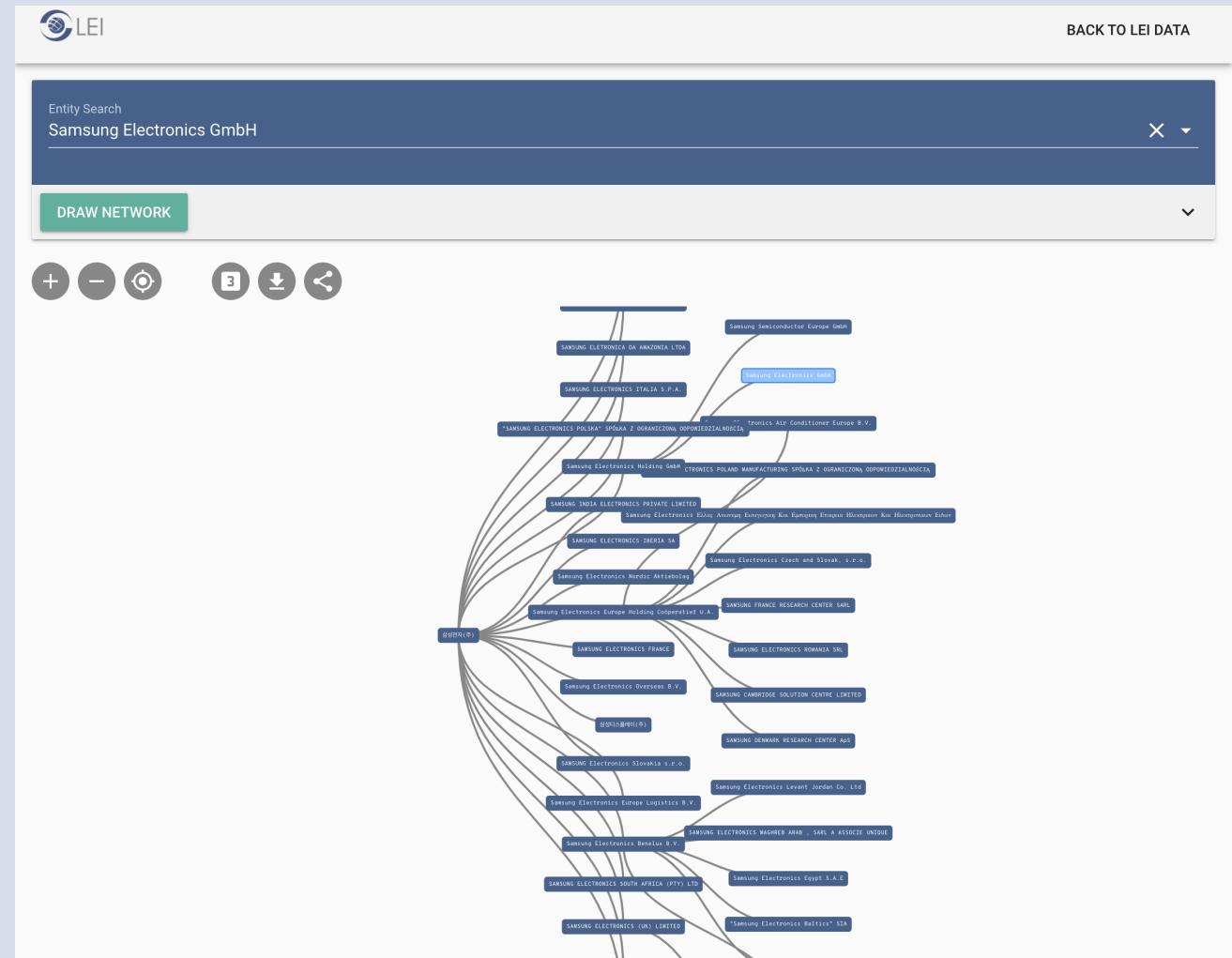
Interactively visualise ownership structures of legal entities using GLEIFs Level 2 data.

Technologies:

Docker; FastAPI; Vue.js; Vis.js

Links:

- <https://correlaid.org/blog/hackathon-gleif/>
- <https://gleif.correlaid.org/>
- <https://github.com/CorrelAid/gleif-level2-client>



Digital Report 2020

Interactively visualise results of the Digital Report 2020 survey and a digital maturity assessment.

Technologies:

R; Vue.js

Links:

- <https://digital-report.org/report>



Mandatsrechner 2.0

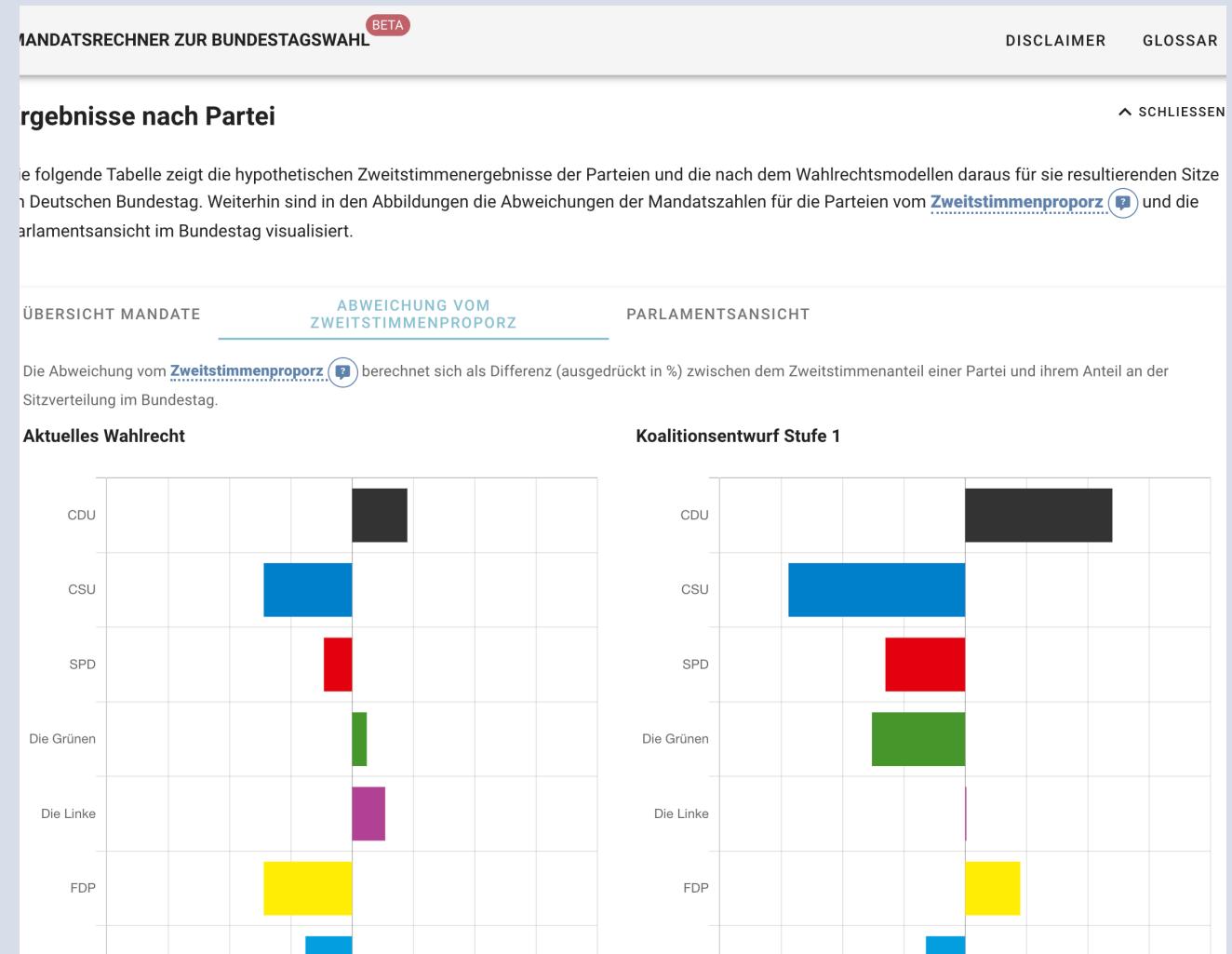
Interactively visualise theoretical distribution of seats in the German parliament using different electoral laws.

Technologies:

PHP; Vue.js; Chart.js

Links:

- <https://next.mandatsrechner.de/>



The [current] stack

Database: PostgreSQL

Pros:

- Geographical data with PostGIS
- Fulltext search using ts_vector and ts_query

Cons:

- Most organisations tend to use MySQL/MariaDB or SQL Server/Azure SQL (for whatever reason)

Backend: FastAPI 1

Pros:

- Parsing and validation using Pydantic
- SQLAlchemy ORM
- Automatically generates OpenAPI documentation
- Extraordinary documentation
- Starter template using Docker, PostgreSQL, Vue.js

Cons:

- Young framework

Backend: FastAPI 2

```
 1 from typing import Optional
 2 from fastapi import FastAPI
 3
 4 app = FastAPI()
 5
 6 fruits = ["Apple", "Banana", "Orange"]
 7
 8 @app.get("/fruits/{limit}")
 9 def list_fruits(limit: int = 3, t: Optional[str] = Query(None, regex="^lower|upper$")):
10     if t == "lower":
11         return {"fruits": [fruit.lower() for fruit in fruits[:limit]]}
12     elif t == "upper":
13         return {"fruits": [fruit.upper() for fruit in fruits[:limit]]}
14     else:
15         return {"fruits": fruits[:limit]}
```

Frontend: Vue.js 1

Pros:

- Vue CLI
- Vue Router and Vuex
- Flat learning curve
- Vuetify

Frontend: Vue.js 2

Search

Dessert (100g serving)	Calories	Fat (g)	Carbs (g)	Protein (g)	Iron (%)
Frozen Yogurt	159	6	24	4	1%
Ice cream sandwich	237	9	37	4.3	1%
Eclair	262	16	23	6	7%
Cupcake	305	3.7	67	4.3	8%
Gingerbread	356	16	49	3.9	16%

Rows per page: 5 1-5 of 10 < >

Search

Dessert (100g serving)	Calories	Fat (g)	Carbs (g)	Protein (g)	Iron (%)
Frozen Yogurt	159	6	24	4	1%
Ice cream sandwich	237	9	37	4.3	1%
Eclair	262	16	23	6	7%
Cupcake	305	3.7	67	4.3	8%
Gingerbread	356	16	49	3.9	16%

Rows per page: 5 1-5 of 10 < >

```
<div id="app">
  <v-app>
    <v-card>
      <v-card-title>
        <v-text-field
          v-model="search"
          prepend-inner-icon="mdi-magnify"
          label="Search"
          outlined>
        </v-text-field>
      </v-card-title>
      <v-data-table
        :headers="headers"
        :items="desserts"
        :search="search"
      ></v-data-table>
    </v-card>
  </v-app>
</div>
```

```
new Vue({
  el: '#app',
  vuetify: new Vuetify({ theme: {dark: false} }),
  data () {
    return {
      search: '',
      headers: [
        {
          text: 'Dessert (100g serving)',
          align: 'start',
          filterable: false,
          value: 'name',
        },
        { text: 'Calories', value: 'calories' },
        { text: 'Fat (g)', value: 'fat' },
        { text: 'Carbs (g)', value: 'carbs' },
        { text: 'Protein (g)', value: 'protein' },
        { text: 'Iron (%)', value: 'iron' },
      ],
      desserts: [
        {
          name: 'Frozen Yogurt',
          calories: 159,
          fat: 6.0,
          carbs: 24,
          protein: 4.0,
          iron: '1%',
        },
        ...
      ],
    }
  }
})
```

Visualisation: Highcharts 1

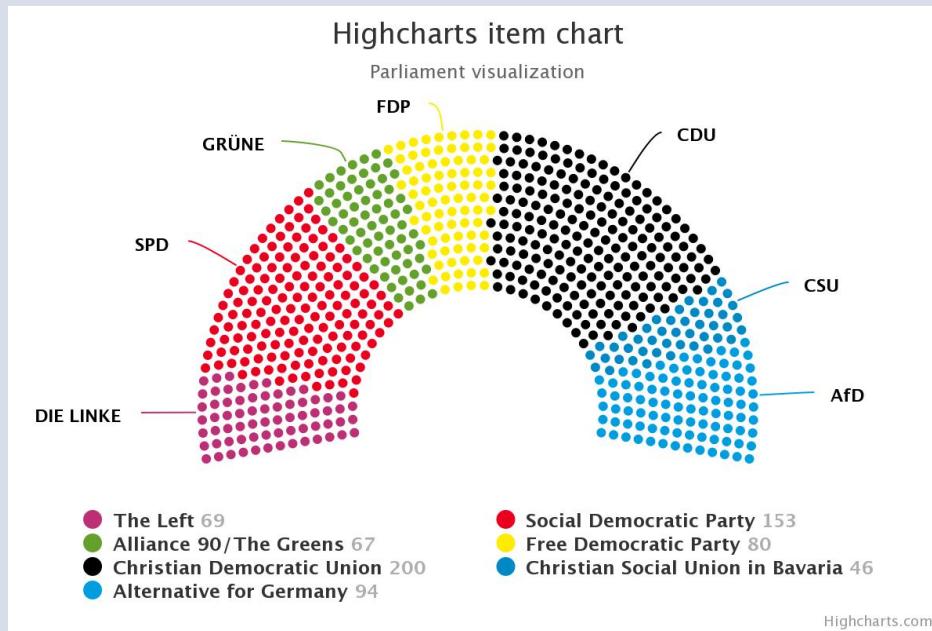
Pros:

- Flat learning curve
- Bindings for Vue.js
- Very mature
- Active support

Cons:

- Pricing scheme

Visualisation: Highcharts 2



```
1 Highcharts.chart('container', {  
2   chart: {  
3     type: 'item'  
4   },  
5   title: {  
6     text: 'Highcharts item chart'  
7   },  
8   subtitle: {  
9     text: 'Parliament visualization'  
10  },  
11  legend: {  
12    labelFormat: '{name} <span style="opacity: 0.4">{y}</span>'  
13  },  
14  series: [{  
15    name: 'Representatives',  
16    keys: ['name', 'y', 'color', 'label'],  
17    data: [  
18      ['The Left', 69, '#BE3075', 'DIE LINKE'],  
19      ['Social Democratic Party', 153, '#EB001F', 'SPD'],  
20      ['Alliance 90/The Greens', 67, '#64A12D', 'GRÜNE'],  
21      ['Free Democratic Party', 80, '#FFED00', 'FDP'],  
22      ['Christian Democratic Union', 200, '#000000', 'CDU'],  
23      ['Christian Social Union in Bavaria', 46, '#008AC5', 'CSU'],  
24      ['Alternative for Germany', 94, '#009EE0', 'AfD']  
25    ],  
26    dataLabels: {  
27      enabled: true,  
28      format: '{point.label}'  
29    },  
30    center: ['50%', '88%'],  
31    size: '170%',  
32    startAngle: -100,  
33    endAngle: 100  
34  }]  
35});
```

Alternatives

Databases

SQL:

- SQL Server
- MariaDB
- MySQL

...

NoSQL

- MongoDB
- Solr

...

Backend

Python

- Django (First commit: 2005-07-13)
- Flask (First commit: 2010-04-06)
- FastAPI (First commit: 2018-12-05)

R

- Plumber

...

Frontend

- Angular
 - React
 - Svelte
- ...

Visualization

- Chart.js
- Echarts
- D3
 - Highcharts
 - Plotly
- ...

Choose a backend framework that supports

- 1. an ORM that is compatible with common database systems**
- 2. allows to easily add data science “magic”**
- 3. a proper documentation out-of-the-box**

How to get started?

Tips

- Start testing
- Docker
- Type hints
- Start logging
- Setup your development environment to support you
 - Linter (ESLint)

Links

FastAPI:

- <https://fastapi.tiangolo.com/>
- <https://github.com/tiangolo/full-stack-fastapi-postgresql>

Vue.js

- <https://vuejs.org/>
- <https://www.udemy.com/user/maximilian-schwarzmueller/>
- <https://vuetyjs.com/en/features/theme/>

Questions

Thank you!