

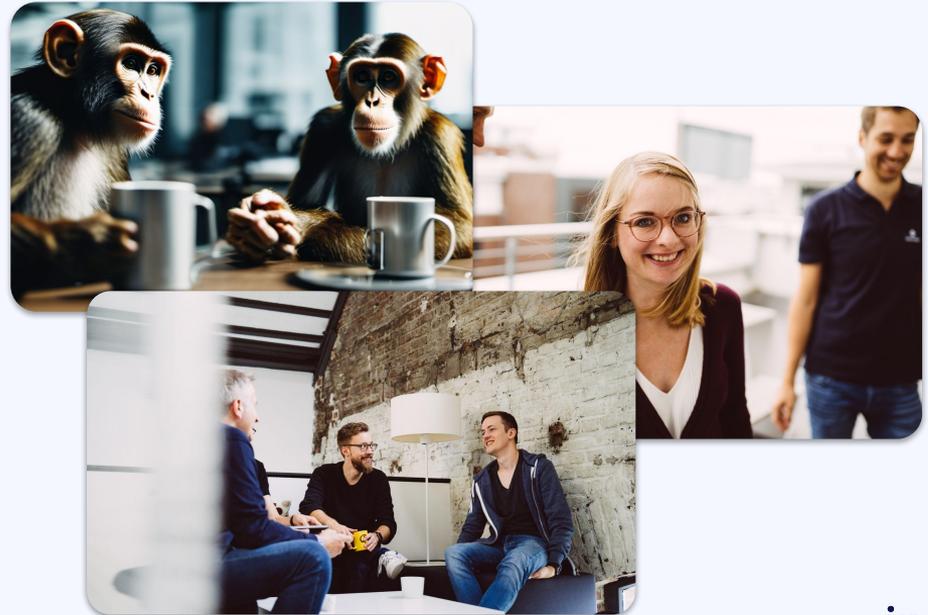
Herzlich Willkommen zum inovex Meetup!



inovex ist ein innovations- und qualitätsgetriebenes IT-Projekthaus mit dem Leistungsschwerpunkt „Digitale Transformation“.

Aktuelle Schwerpunkte:

- Agile Transformation
- Product Development Workshops
- E-Health
- Recommender Systems
- Generative AI

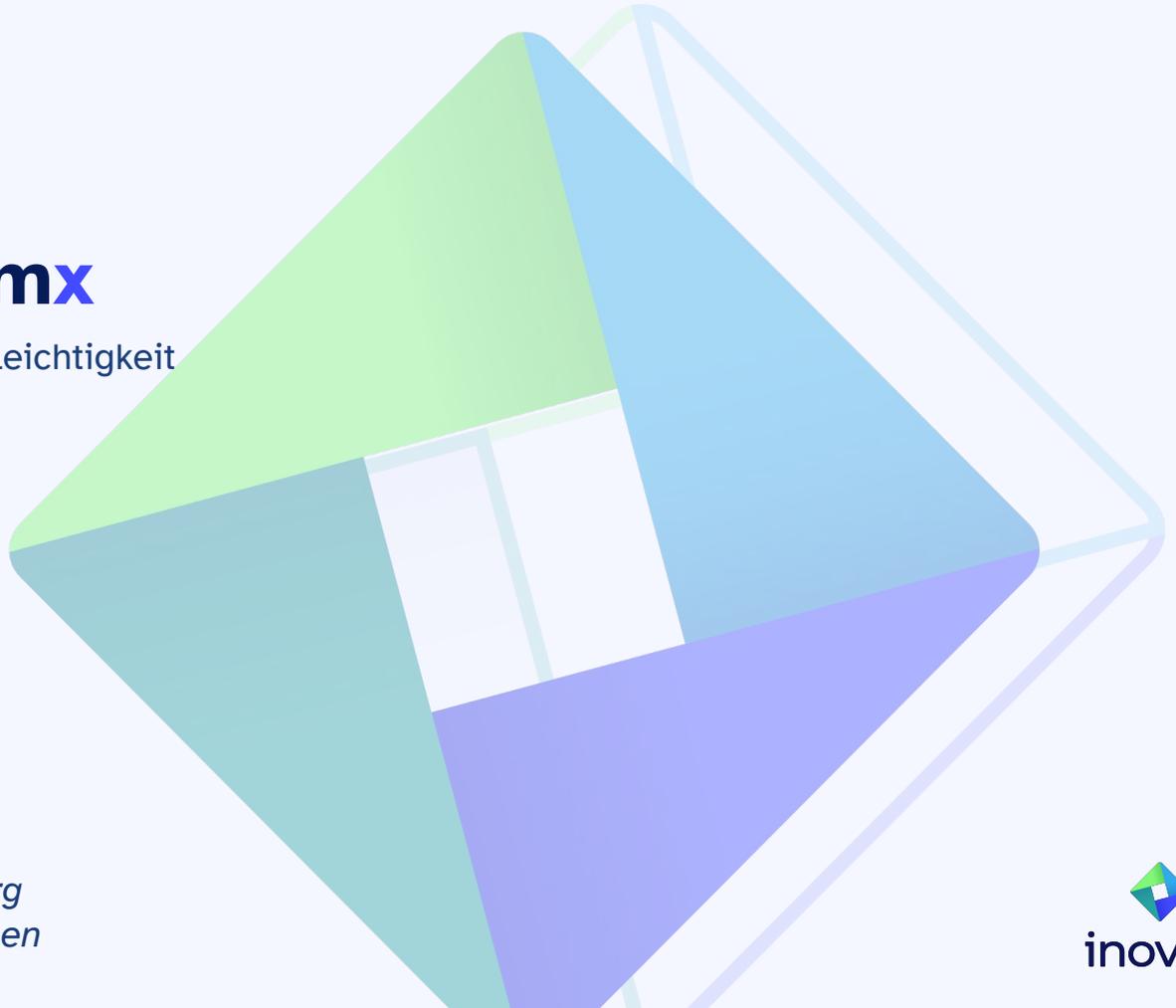


Einführung in htmx

Dynamische Webanwendungen mit Leichtigkeit

Johanna Dolinga

*Karlsruhe · Köln · München · Hamburg
Berlin · Stuttgart · Pforzheim · Erlangen*



inovex

Johanna Dolinga



Fullstack Entwicklerin

- Webentwicklung
- UX
- Agile Development



Johanna Dolinga



@jdolinga

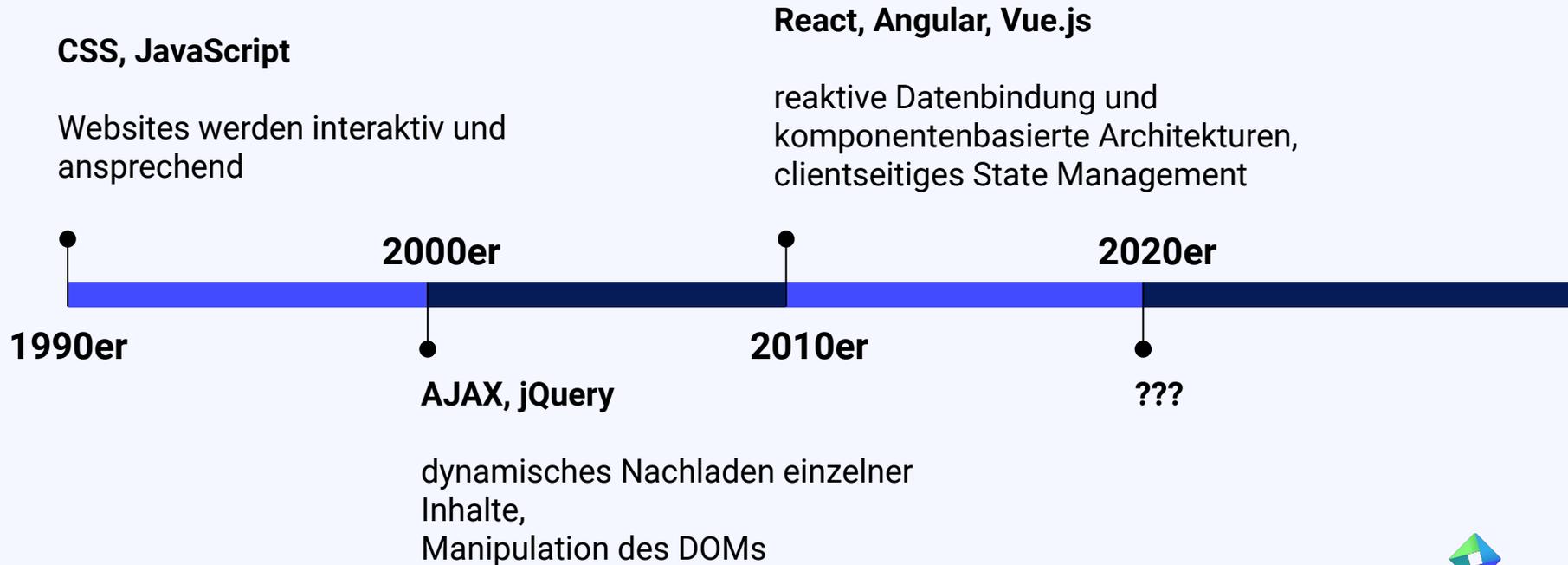
1. htmx - was ist das überhaupt?
2. Demo-Anwendung
3. Deep Dive: Coding mit htmx





Rückblick

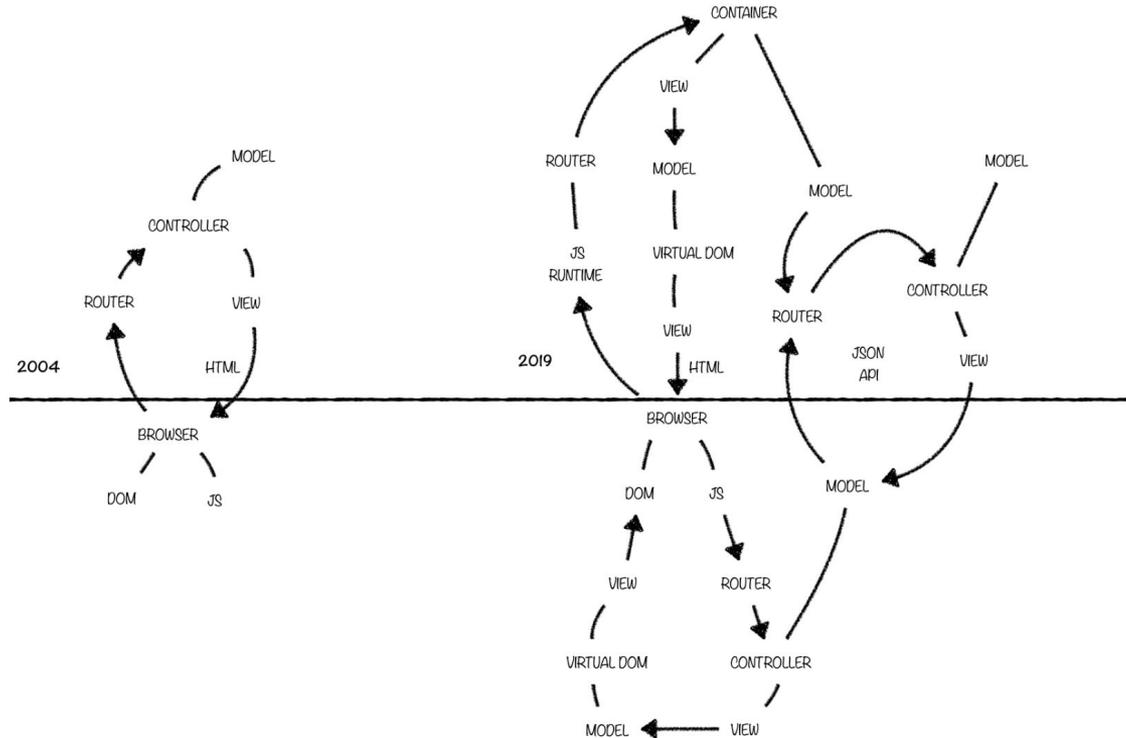
Webentwicklung hat sich in den letzten 30 Jahren grundlegend verändert



Nichtfunktionale Anforderungen an eine moderne Webanwendung

- Responsive Design
- Reibungslose Interaktion
- Animationen
- Komponentenbasierte Architektur
- Partielles Nachladen
- Eine Codebase
- Erweiterbarkeit
- Client Side State
- Dokumentation
- A11y, Security, Skalierbarkeit,...

Die Komplexität von Frontend-Anwendungen hat zugenommen



Ist diese Komplexität notwendig?





Exkurs: Hypermedia

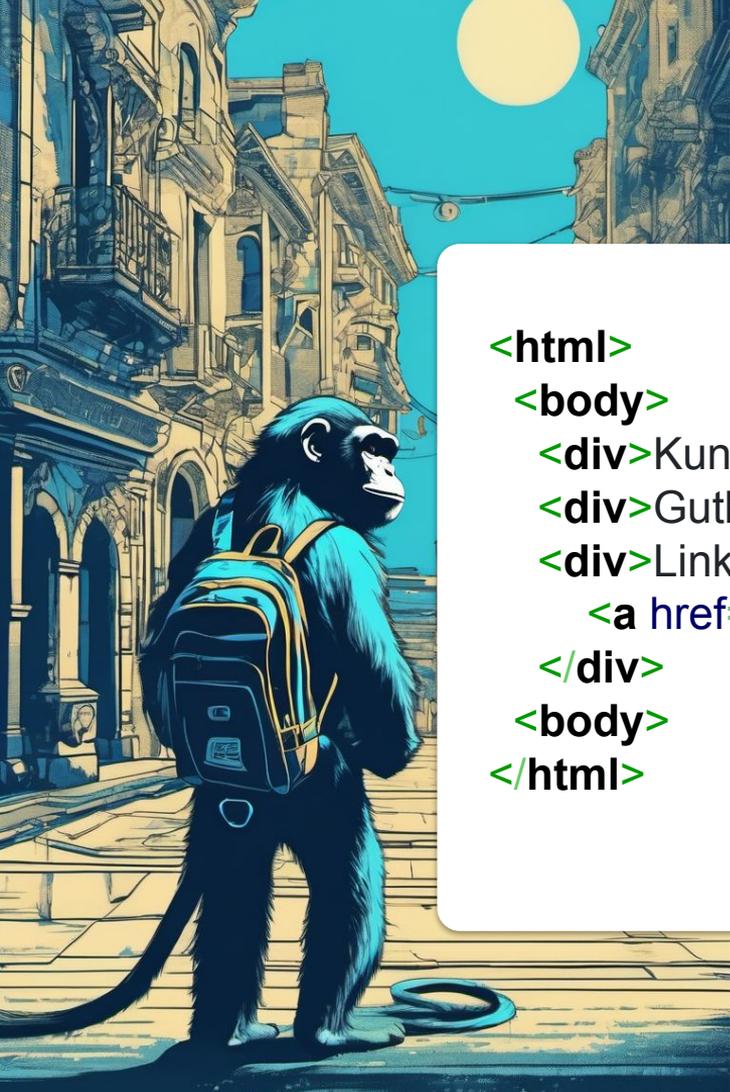
- Verknüpfung von Medien durch Links
- Hypermedia ist Teil der REST-Leitlinien

HATEOAS:

Hypermedia as the Engine of Application State

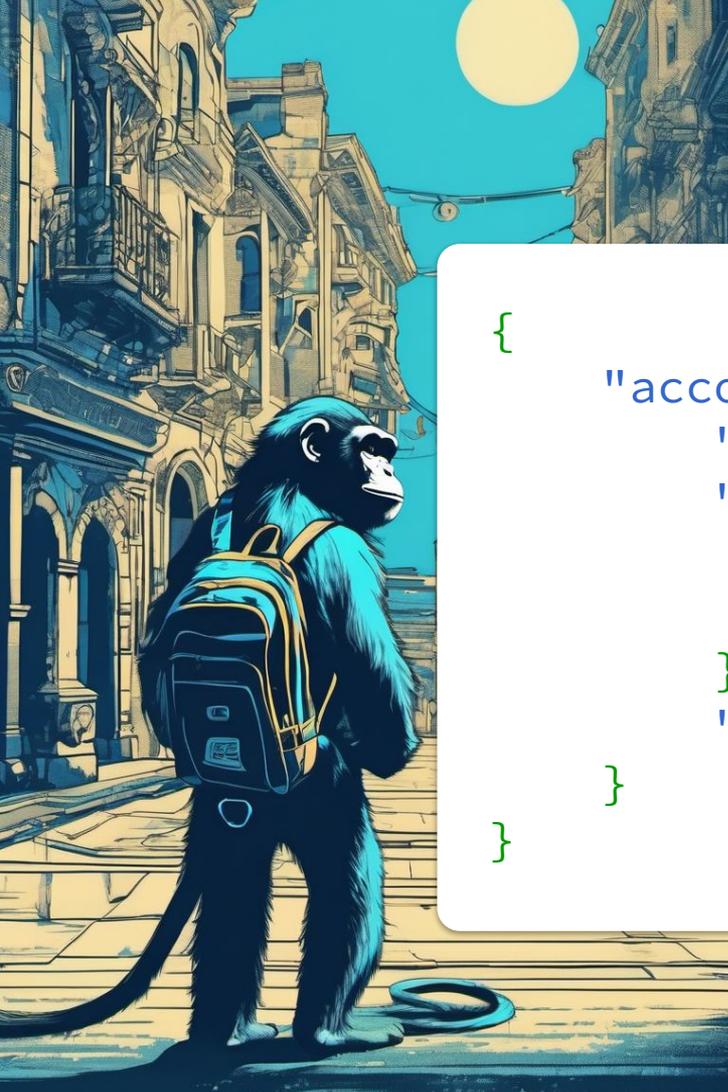
Exkurs: Hypermedia

```
<html>
<body>
  <div>Kundennummer: 12345</div>
  <div>Guthaben: 100.00 EUR</div>
  <div>Links:
    <a href="/accounts/12345/deposits">Einzahlen</a>
    <a href="/accounts/12345/withdrawals">Auszahlen</a>
    <a href="/accounts/12345/transfers">Überweisen</a>
  </div>
</body>
</html>
```



Exkurs: Hypermedia

```
<html>
  <body>
    <div>Kundennummer: 12345</div>
    <div>Guthaben: -50.00 EUR</div>
    <div>Links:
      <a href="/accounts/12345/deposits">Einzahlen</a>
    </div>
  </body>
</html>
```



Exkurs: Hypermedia

```
{  
  "account": {  
    "account_number": 12345,  
    "balance": {  
      "currency": "eur",  
      "value": -50.00  
    },  
    "status": "overdrawn"  
  }  
}
```

HTML enthält bereits einige Hypermedia-Funktionen

click und submit
events

<form> , <a>

POST bzw. GET
request

reload page

```
<html>
  <body>
    <div>Kundennummer: 12345</div>
    <div>Guthaben: -50.00 EUR</div>
    <div>Links:
      <a href="/accounts/12345/deposits">Einzahlen</a>
    </div>
  </body>
</html>
```

Durch die Beseitigung dieser willkürlichen Beschränkungen vervollständigt htmx HTML als Hypertext

click und submit
events

jedes Event

<form> , <a>

all Arten von Tags

POST bzw. GET
request

all Request-Typen

reload page

teilweiser Reload

The logo for htmx, featuring a stylized code symbol '</>' in black and blue, followed by the lowercase text 'htmx' in a bold, black, sans-serif font.

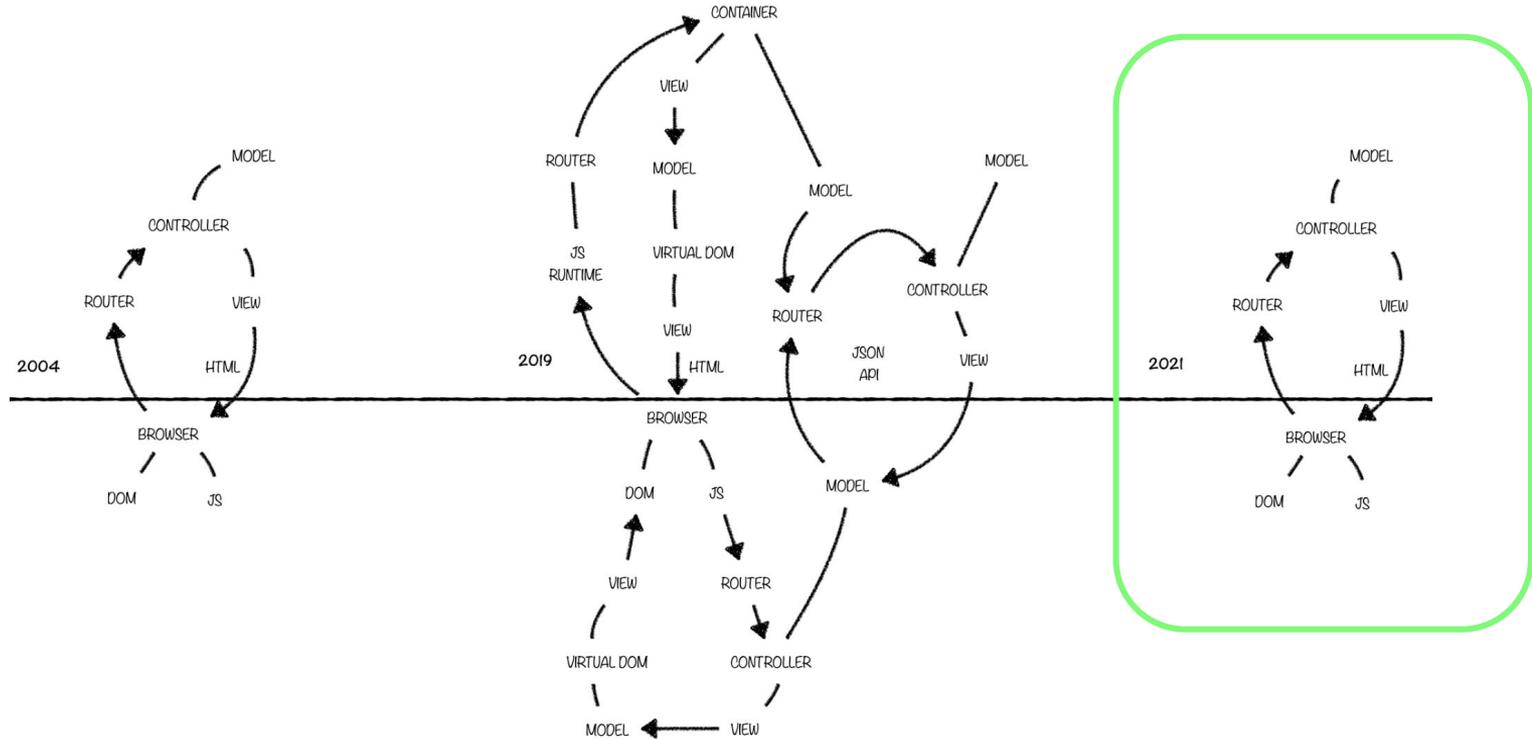
htmx ist eine Bibliothek zur dynamischen Aktualisierung von Webseiten ohne JavaScript

introduction

htmx allows you to access [AJAX](#), [CSS Transitions](#), [WebSockets](#) and [Server Sent Events](#) directly in HTML, using [attributes](#), so you can build [modern user interfaces](#) with the [simplicity](#) and [power](#) of hypertext

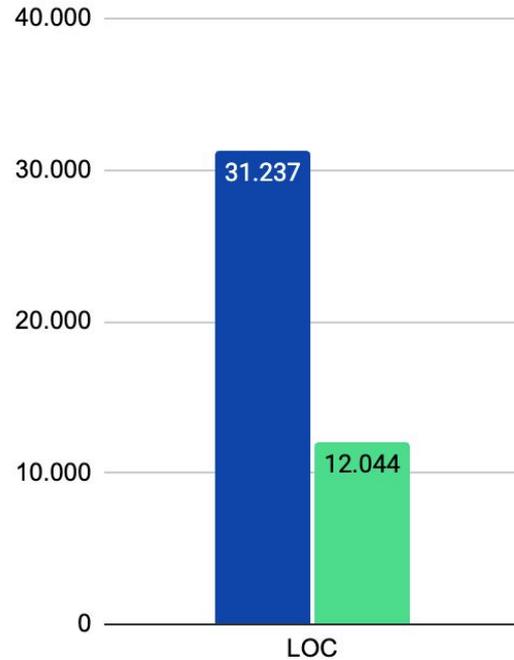
htmx is small ([~14k min.gz'd](#)), [dependency-free](#), [extendable](#) & IE11 compatible

Frontend-Applikationen könnten weniger komplex sein

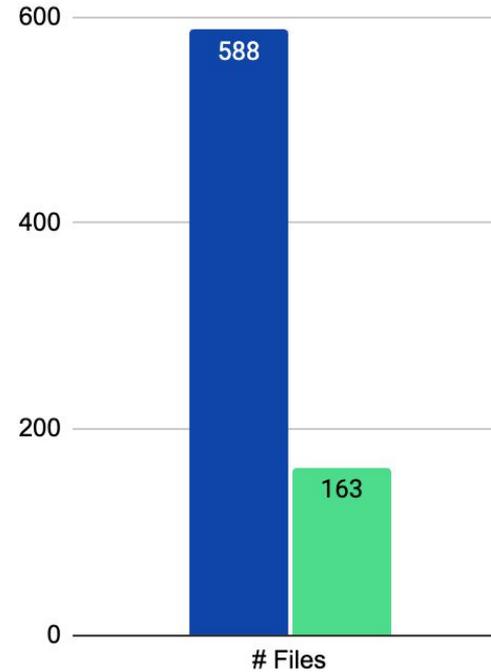


Reduktion von Komplexität ist spürbar

Lines of Code



Anzahl an Files



Wie ist die User Experience?



inovex Turniere

SPIELER:IN 1	PUNKTE	SPIELER:IN 2	PUNKTE
<input type="text" value="Clemens"/>	<input type="text"/>	<input type="text" value="Clemens"/>	<input type="text"/>
<input type="button" value="SPEICHERN"/>			
Spieler:in hinzufügen			

HIGHSCORES



Spielergebnisse

SPIELER 1	SPIELER 2	ERGEBNIS	
Clemens	Lukas	10 : 5	Löschen
Clemens	Clemens	1 : 1	Löschen
Clemens	Johanna	1 : 5	Löschen
Clemens	Johanna	1 : 5	Löschen

Nichtfunktionale Anforderungen an eine moderne Webanwendung

- 
- Responsive Design ✓
 - Reibungslose Interaktion ✓
 - Animationen ✓
 - Komponentenbasierte Architektur
 - Partielles Nachladen
 - Eine Codebase
 - Erweiterbarkeit
 - Client Side State
 - Dokumentation
 - A11y, Security, Skalierbarkeit,...



**Wie funktioniert
das?**



Tech Stack

~~Java~~ = GO

↔

 `</>` htmx

Nichtfunktionale Anforderungen an eine moderne Webanwendung

- Responsive Design ✓
- Reibungslose Interaktion ✓
- Animationen ✓
- Komponentenbasierte Architektur ✓
- Partielles Nachladen ✓
- Eine Codebase ✓
- Erweiterbarkeit ⚠
- Client Side State ⚠
- Dokumentation ⚠
- A11y, Security, Skalierbarkeit, ... 🙋

Fazit

Auf den ersten Blick sieht es gut aus.

- Anwendungen sind weniger komplex
- User Experience wie bei SPA
- einfach zu lernen

Aber: Wie ist es, wenn man genauer hinschaut?

