



# JAVA ENGINEERS UND WEB GUIS

Ein Erfahrungsbericht mit Tipps und Ideen

Johann Latocha, Roman Schlegel, Stefan Rufer



**Stefan Rufer**  
Product Owner



**Johann Latocha**  
Software Engineer



**Roman Schlegel**  
Software Architekt

# Disclaimer

- Kein Anspruch auf Vollständigkeit  
(nur einige «Dinge, die im Nachhinein wichtig erscheinen»)
- Kein SIX-Kurs  
(nur persönliche Ansichten)
- Results may vary  
(entscheidend sind die konkreten **Anforderungen!**)

Spezifikation  
(389 Seiten)

Welche Leute  
brauchen wir  
denn dafür?

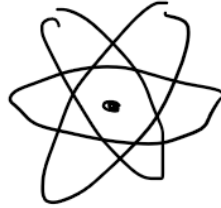
Features:  
all current ideas

Accessibility:  
AAA rating

Usability:  
best in class  
(desktop & mobile)

Performance:  
everything <1s

# Sprache- / Framework- / Library-Wahl



# Sprache- / Framework- / Library-Wahl



- ✓ Single-Page-Application
- ✓ Templating
- ✓ Data-Sanitization
- ✓ Dev- & Testing-Tools



...und die Skills  
dafür...?

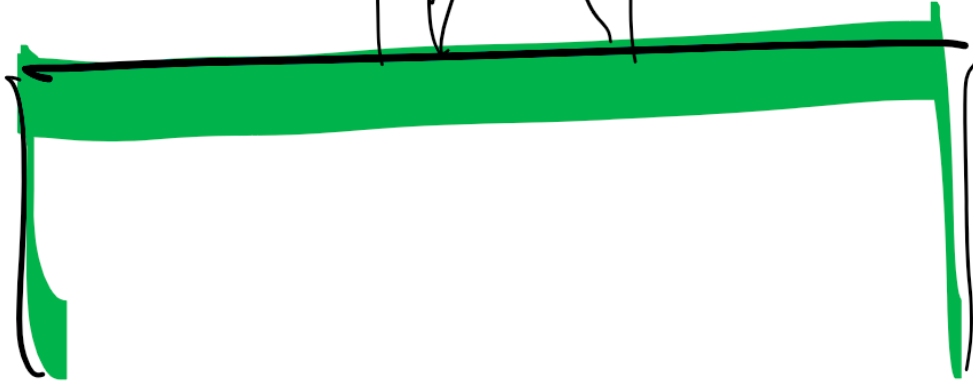
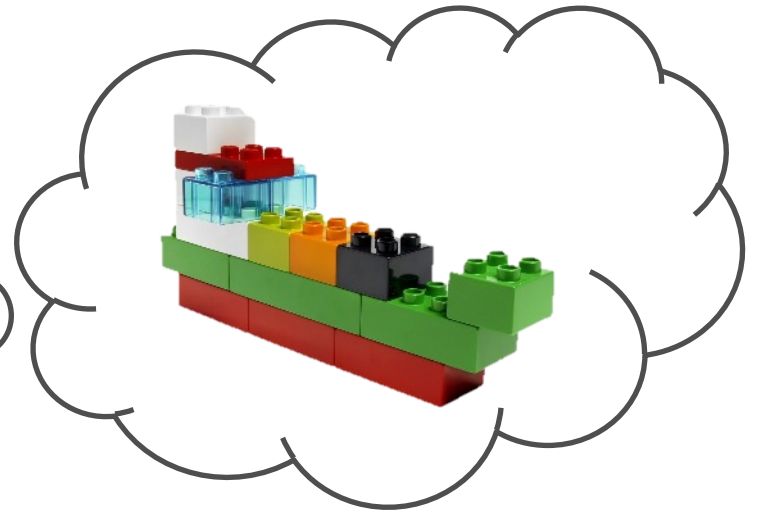
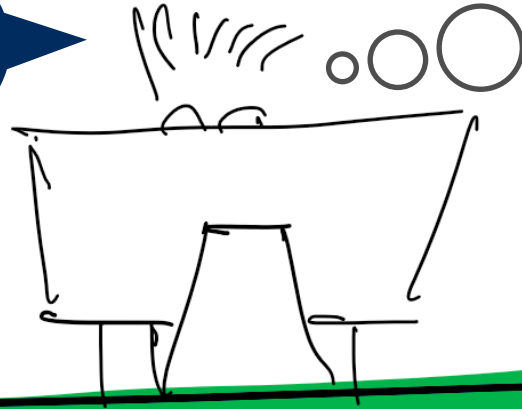


...können wir ja  
lernen oder  
einkaufen!



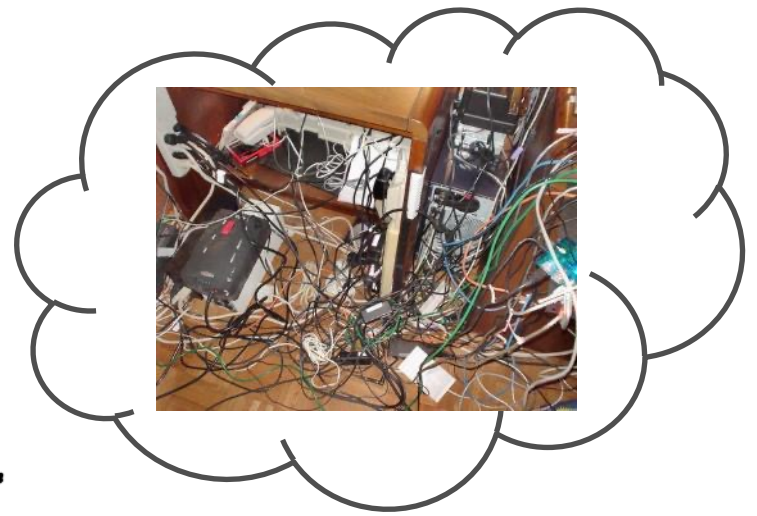
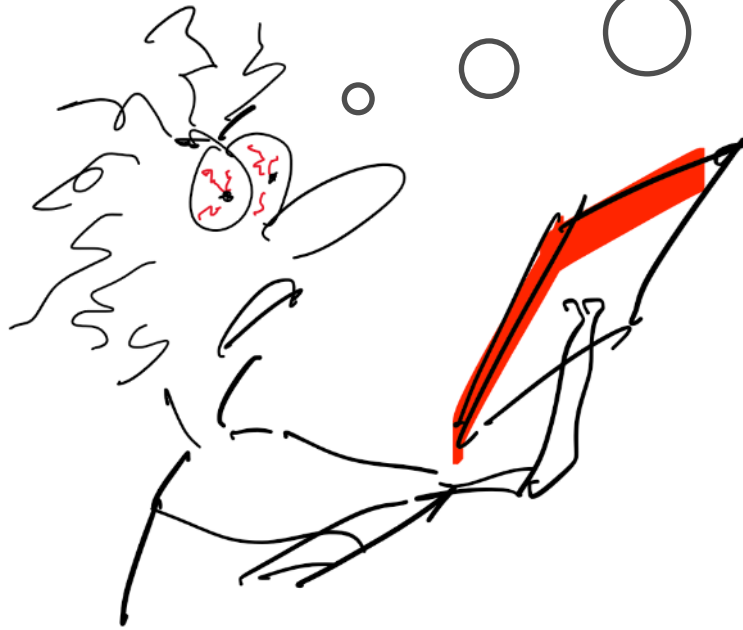
# Hello-World App

Wow das geht ja  
easy!

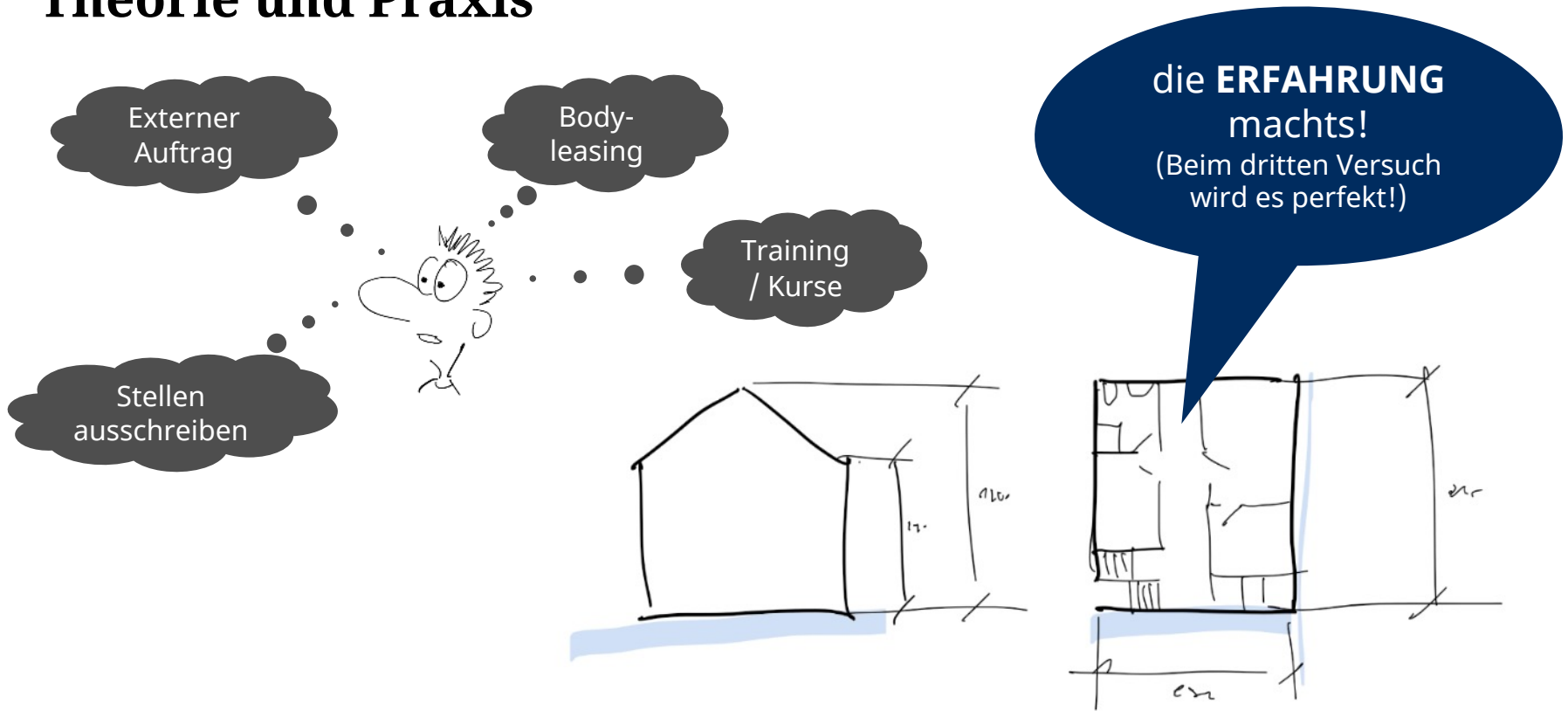




Einige Wochen später...

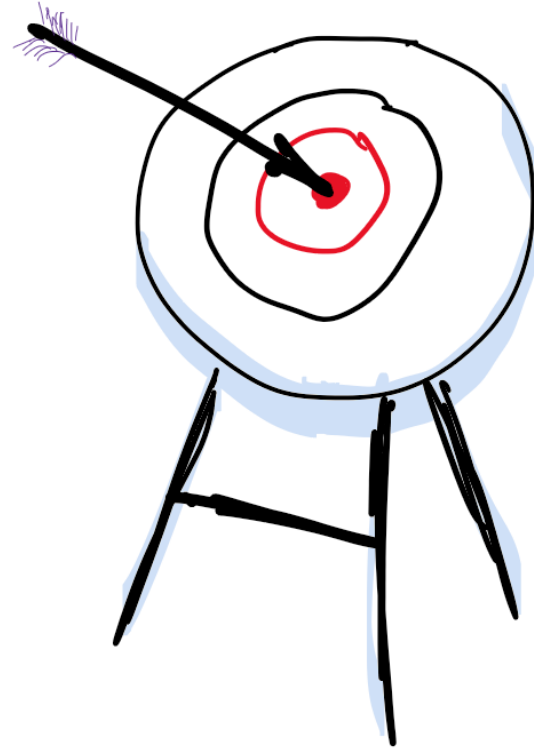


# Theorie und Praxis



# Scope und Ziel

Entscheidend sind  
die Anforderungen!



## Entwicklung: Start effizient gestalten



- Stakeholders & Anforderungen kennen
- Phasen / **MVP** definieren
- sicherstellen dass Skills vorhanden sind
- iterativ vorgehen
- Usability testen (vor Umsetzung!)



- End-User «vergessen»
- Spezifikation ohne Usability-Abklärungen
- Zu grosser Scope für den «ersten Wurf»
- mehrere Teams gleichzeitig starten lassen
- mehrere externe Firmen gleichzeitig beauftragen

# Tabellen

Total: 1377

Name	Geburtsdatum	Status
Hans	1.3.80	Aktiv
...	...	...
...	...	...

Seite | 1 | 2 | 3 | 4 | 5 | ... | 100 |

- ! Styling, Responsive-Design?
  - Scrolling (horizontal?)
- Paging
- Sortierung
  - mehrere Spalten?
- ! Total-Anzahl?
- Filter
  - (Sub-)Text
    - ! Auto-completion?
      - Debounce
  - ! Datums-Range
    - Time-Range
    - Enums (Dropdown)
      - Übersetzung!
- ! Loading-Indicator
- Datenanzeige
  - ! Konsistenz
    - Übersetzung
    - Formatierung, Alignment, Icons
    - Links
    - Navigation (Detail-View)
  - ! Baumstrukturen?
- Datenabfrage
  - DB-Joins nötig?
  - ! Tuning, DB-Index?
- Controls
  - Sortierung
  - Refresh
  - Clear Filter
  - Paging?
- Navigation
  - Filter-Cache pro Page?
  - History-Support?

# Komplizierte Tabellen und Filter nur wenn wirklich nötig



- Usability-Abklärungen zuerst! → braucht es wirklich eine Tabelle?
- Einfache (Google-Style-) Suchfelder sind besser verständlich und oft auch deutlich einfacher zu Implementieren
- ggf. kommerzielle Table-Library evaluieren (Styling möglich?)



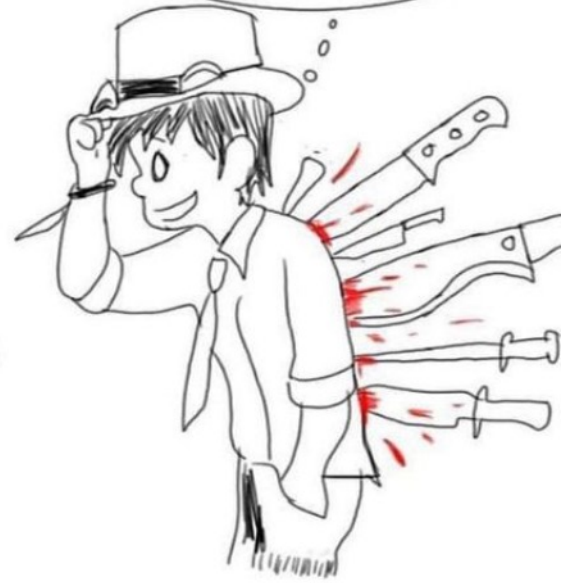
- Excel «selber nachbauen»
- Kommerzielle Library einsetzen obwohl gar keine Tabelle nötig wäre

You'll never understand my pain.



debugging Java

hehe... sorry..sorry..



debugging JavaScript



# Bugs vermeiden & Maintainability sicherstellen



```
switch (action.type) {  
  case "UpdateUserAction": {  
    return {  
      state  
    }  
  }  
}
```

TS2678: Type ""UpdateUserAction"" is not comparable to type ""A|



- TypeScript
- strenge Compiler und Linter-Einstellungen
- geeignete IDE, geeignete Browser-Tools
- (standard-)Sprach-Features statt Libraries
- Nur so viele integration-Tests **wie nötig (!)**  
(Selenium, Browserstack, ...)



- Keine Testautomatisierung
- Zu viele Unit-Tests  
(Karma, Jasmin, usw.)
- Zu viele Integration-Tests  
(Protractor, Selenium, Cucumber usw.)

# Anzeige-Konsistenz sicherstellen



User-Interaktion  
muss **IMMER**  
FEEDBACK geben

Es muss **JEDERZEIT**  
etwas **KONSISTENTES**  
angezeigt werden!

Und es sollte die  
**WAHRHEIT** sein



Anzeige-Konsistenz sicherstellen



# Anzeige-Konsistenz sicherstellen



- Redux-Flow** (z.B. ngrx benutzen)<sup>1</sup>
  - Side-Effects über Global Store «syncen»
- Container-Pattern**
  - Container-Components ohne HTML
  - View-Components ohne Observables
- Faustregel: Jeder Klick = URL-Change**  
(= Reload und Back-Button funktionieren überall)<sup>1</sup>

1) Ausnahme: State von Form-Elementen



- Container- und View-Component Aspekte vermischen**
- «manuelles» Subscribe auf Observables**
- Navigation in der App ohne URL-Change**

# Mobile-friendliness & Accessibility sicherstellen



# Mobile-friendliness & Accessibility sicherstellen

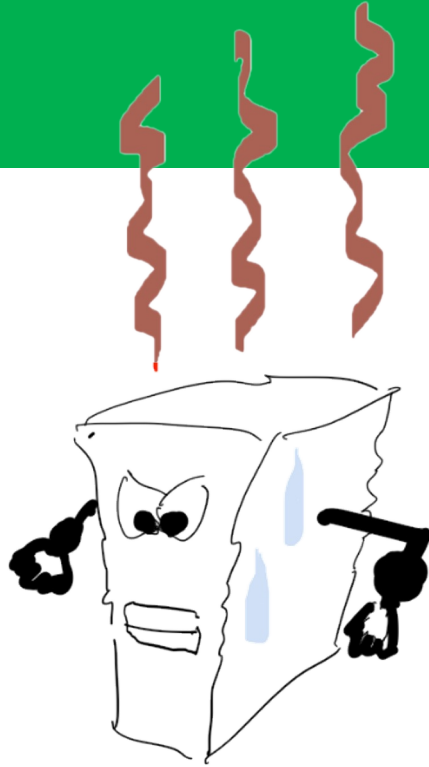


- Mobile-First Ansatz
- Accessibility-Training (ein Nachmittag)
- Standard HTML-Elemente benutzen (Titel h1, h2, ... / Button / <a>-Link / Section / ...)
- Hohe Kontraste, grosse Schriftarten
- Bilder & Logos nur für Ästhetik einsetzen



- Pages «überladen», so dass sie nur auf Desktop funktionieren (z.B. Tabellen!)
- HTML-Elemente «zweckentfremden» (z.B. div zu Listen «umstylen», oder div zu Buttons oder Links machen)
- Titel-Elemente in falscher Reihenfolge
- Bilder mit essenziellen Informationen ohne Text-Alternativen

# Backend Service performance & maintainability





# Backend Service performance & maintainability

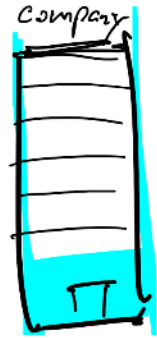


- Sauberes, durchdachtes API-Design
- Nur Queries unterstützen die wirklich benötigt werden
- Paging / Sorting vorsehen und generisch lösen
- UI Suchformulare möglichst effizient und verständlich machen

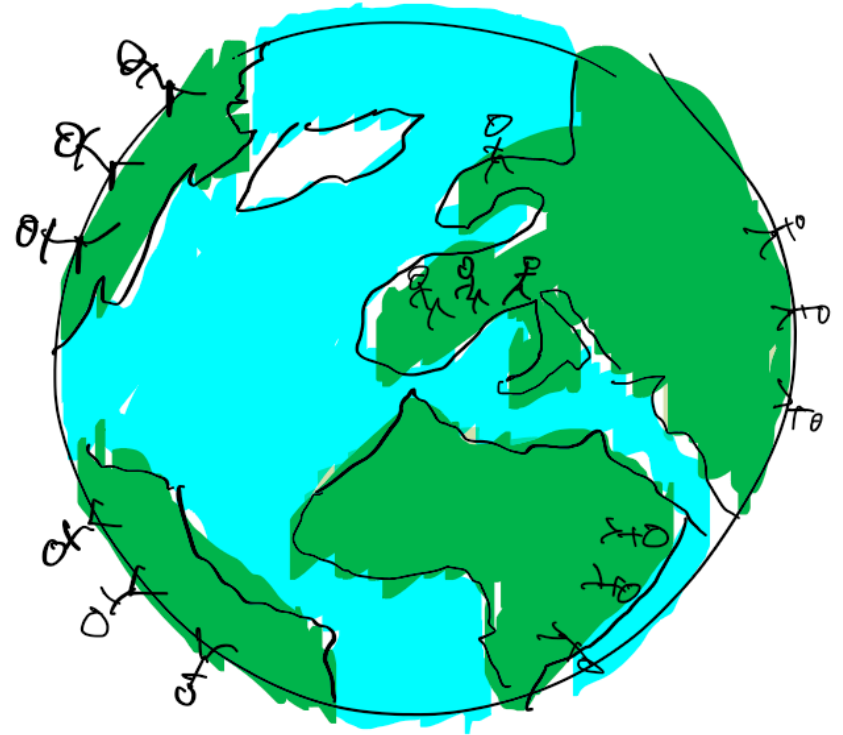


- Substring-Suchen in unnötigen Feldern
- Queries ohne Limiten
- Zu viele Details in den Suchresultaten

# Intern vs. Öffentlich



App



# Intern vs. Öffentlich

Usability →  
Support-Aufwand

Zuviele Features →  
Support-Aufwand

Fehlende Features  
→ Presse

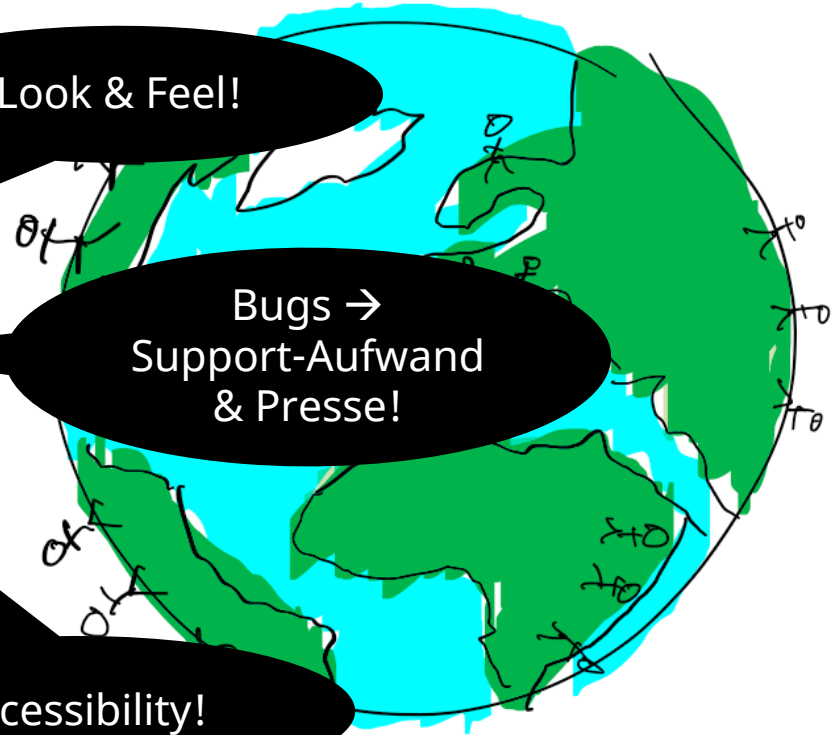


App

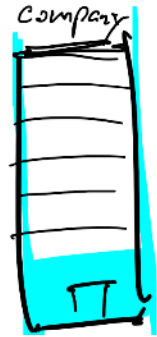
Look & Feel!

Bugs →  
Support-Aufwand  
& Presse!

Accessibility!



# Intern vs. Öffentlich



App

Erweiterbarkeit

Effizienz /  
Performance

Konsistenz mit  
anderen Tools!  
(auch wenn sich das corporate  
design mal ändert)

# Wir machen



Menu:

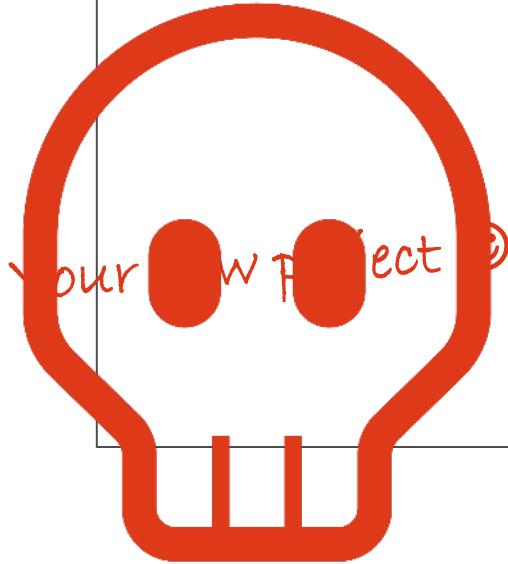
[Feature X](#)

[Feature Z](#)

BIC

Name

**Save**



# Sieht doch schön aus, wo ist das Problem?

The screenshot shows the SIX Banking Services website. The header includes the SIX logo, language selection (Deutsch), a login button, and a menu button. The main navigation bar lists various services: Banking Services, Billing und Payments, Debit & Mobile Services, Cash, Interbank Clearing, Connectivity, and Standardisierung. A secondary navigation bar includes eBill, Direct debits, Payment Standards, and QR-Rechnung. The main content area is titled 'PRODUKTE & SERVICES / BANKING SERVICES' and features a large heading 'Billing & Payments'. Below this, the address 'SIX BBS AG, Hardturmstrasse 201, Postfach, 8021 Zürich' is displayed. To the right, a form is visible with fields for 'Anrede\*' (Frau/Herr), 'Vorname\*', 'Nachname\*', 'Firma', 'Telefon', and 'E-Mail-Adresse\*'. The form fields are currently empty.

**SIX**

Deutsch | Login | Menu

Banking Services | Billing und Payments | Debit & Mobile Services | Cash | Interbank Clearing | Connectivity | Standardisierung

eBill | Direct debits | Payment Standards | QR-Rechnung

PRODUKTE & SERVICES / BANKING SERVICES

## Billing & Payments

ALLGEMEINE  
ADRESSE  
SIX BBS AG  
Hardturmstrasse 201  
Postfach  
8021 Zürich

Anrede\*  
 Frau  Herr

Vorname\* | Nachname\*  
 |

Firma

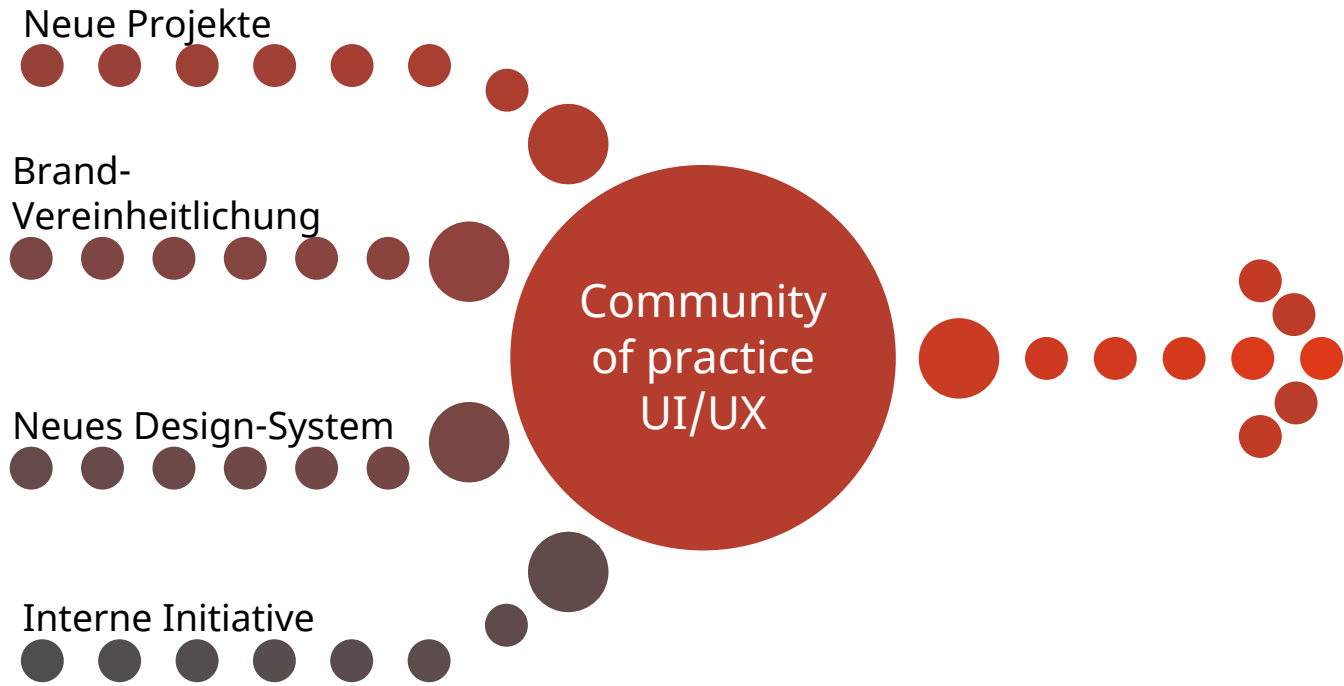
Telefon

E-Mail-Adresse\*









# Technical approaches for modern user interfaces

Option

Strength

SIX Angular components

Central CSS stylesheet for

Visual Design System for SIX

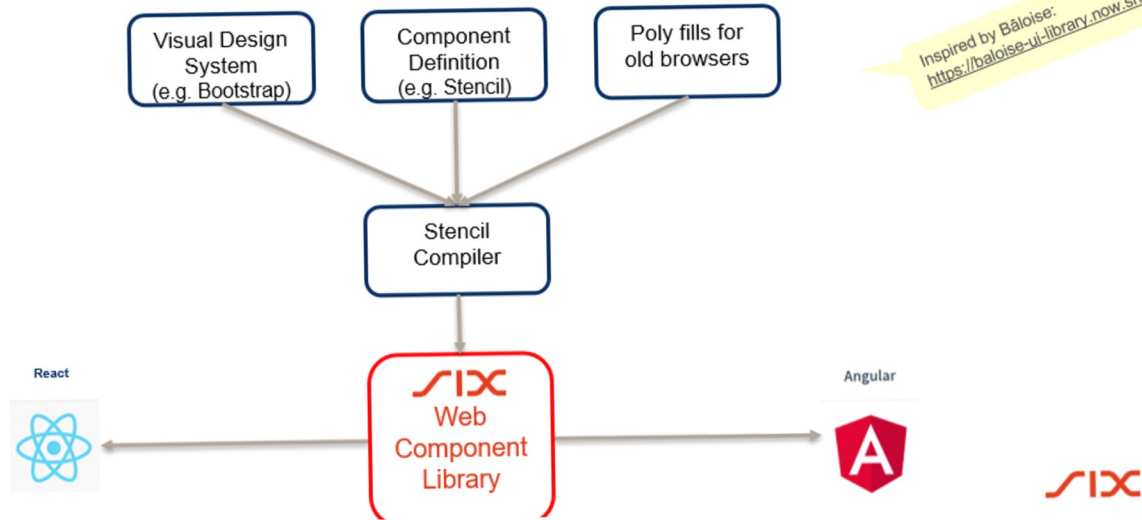
Angular Elements

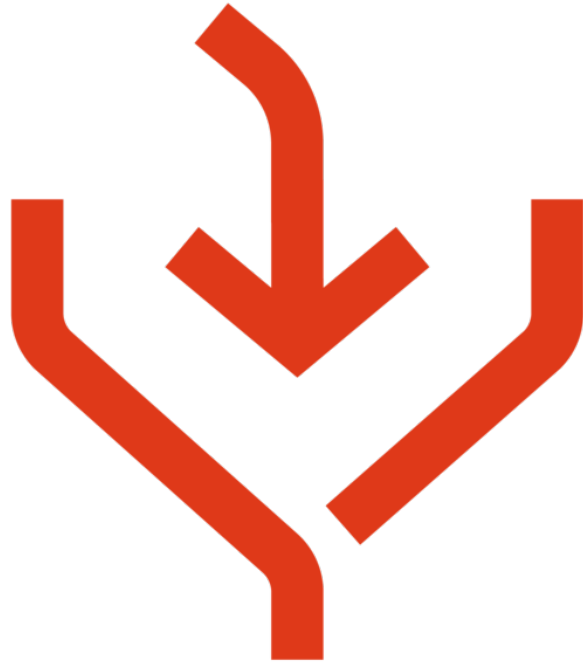
Web-Components

3

## An approach

Based on a **Visual Design System** develop simple **Web Components** that can be used with different user interface frameworks.





1. Visual Design System
2. GUI-Komponenten

# Visual Design System

# Microsoft – öffentliche Webseite

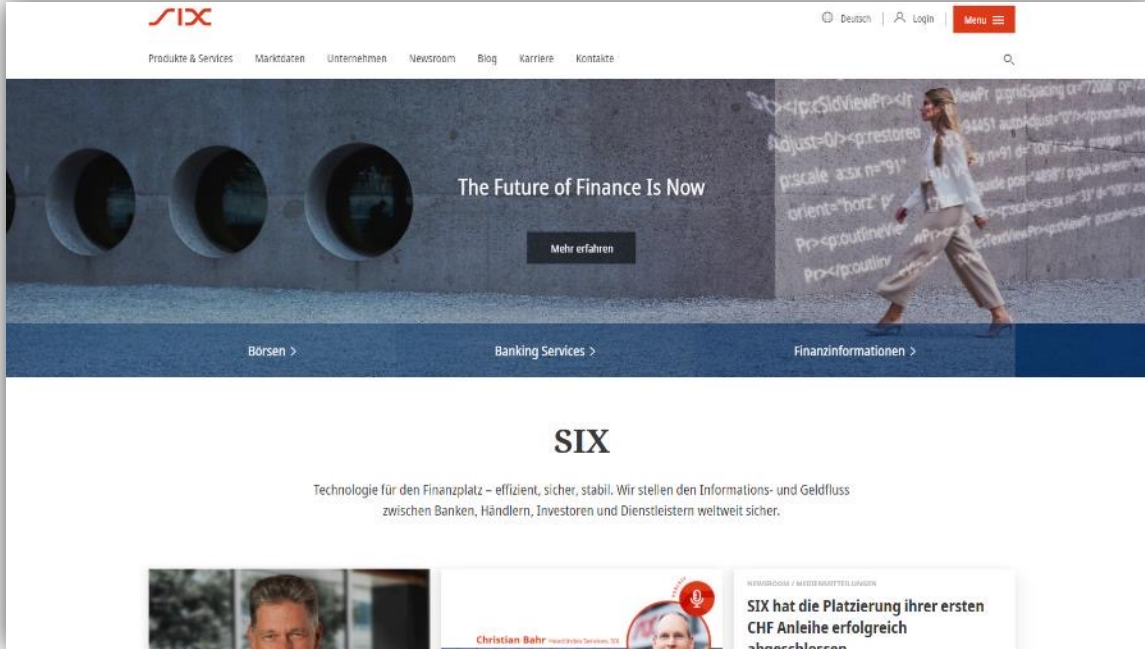


← Inhalts-  
getriebene  
Navigation

← Anrisse, Produkt-  
Informationen

← Inhalt, grosse Bilder...

# SIX – öffentliche Webseite



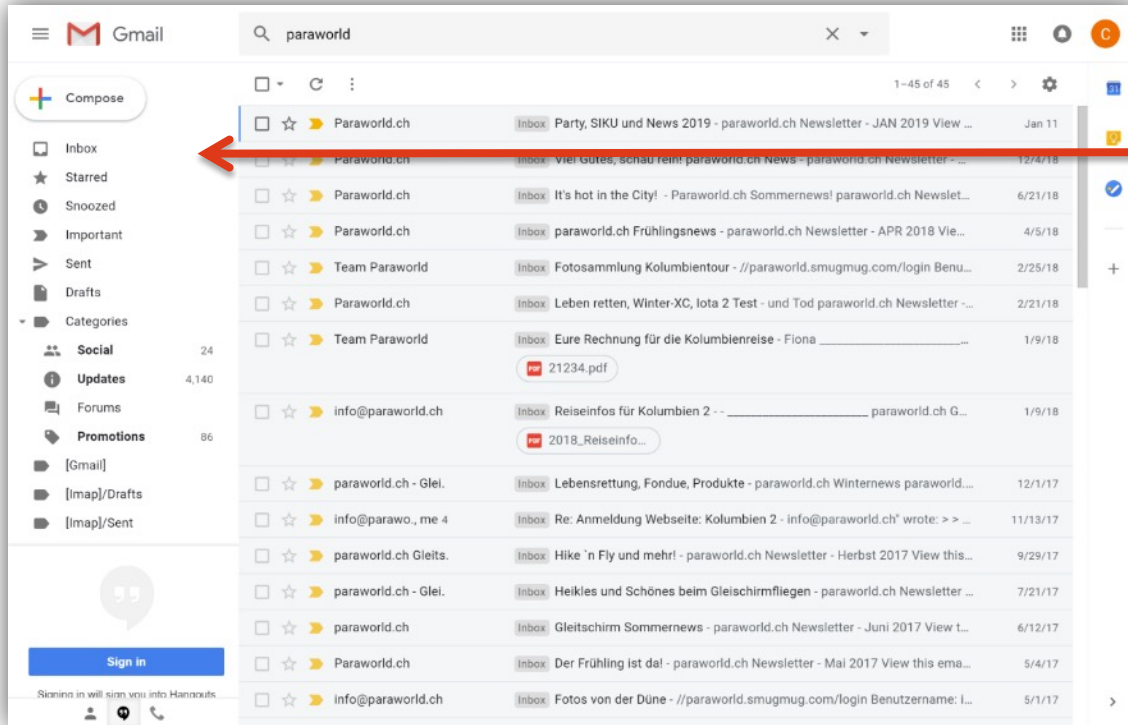
← Inhalts-  
getriebene  
Navigation

← Inhalt, grosse Bilder...

← Anrisse, Produkt-  
Informationen



# Google Web-Applikation



← Einheitlicher Header

← Applikations-  
getriebene  
Navigation

← Applikations-Inhalte

# SIX Web-Applikation

Id	Name	Beschreibung	Typ	geän...	Zusta...
1	pac0.008 - minimale Ausprägung m...	pac0.008 - minimale Ausprägung mit SIC-IID - Original Message ...	MESSAGING	tkpot	ENABLED
2	pac0.008 - minimale Ausprägung m...	pac0.008 - minimale Ausprägung mit SIC-IID - Original Message ...	MESSAGING	tkpot	ENABLED
3	pac0.008 - B-Bank offline	pac0.008 - B-Bank nicht online, inkl folgender Abfrage	MESSAGING	tkpot	ENABLED
4	pac0.008 - minimale Ausprägung m...	pac0.008 - minimale Ausprägung mit SIC-IID - Original Message ...	MESSAGING	tkpot	ENABLED
6	camt.005 - Einzelabfrage, Zahlung ...	camt.005 - minimale Einzelabfrage STXQRY, Zahlung nicht vorha...	MESSAGING	tkamq	ENABLED
7	camt.005 - Einzelabfrage Status res...	camt.005 - minimale Einzelabfrage STXQRY, Status reserviert	MESSAGING	tkpot	ENABLED
8	pac0.008 - NOK B-Bank	pac0.008 - NOK B-Bank, inkl nachfolgende Abfrage	MESSAGING	tkpot	ENABLED
9	camt.005 - Einzelabfrage Status ver...	camt.005 - minimale Einzelabfrage STXQRY, Status verrechnet	MESSAGING	tkpot	ENABLED
10	camt.005 - Einzelabfrage Status un...	camt.005 - minimale Einzelabfrage STXQRY, Status ungenügend...	MESSAGING	tkpot	ENABLED
11	camt.005 - Einzelabfrage Status TI...	camt.005 - minimale Einzelabfrage STXQRY, Status TIMEOUT. Pr...	MESSAGING	tk7p3	ENABLED
12	pac0.008 - LocalDateTime statt Offs...	pac0.008 - schema valide Zeit, die nicht dem internen Format en...	MESSAGING	tkpot	ENABLED
13	pac0.008 - Payment mit acceptance...	Payment mit acceptance timeout zu weit in der Vergangenheit s...	MESSAGING	tk7p3	ENABLED
14	pac0.008 - mit Quittung am nächste...	pac0.008 - minimale Ausprägung mit SIC-IID - Original Message ...	MESSAGING	tkamq	ENABLED
16	keyset switch	Add a new keyset during a clearing day	PROGRAMMED	tkamq	ENABLED
17	Clearing Tag Wechsel zum gleichen...	Camt.019 Meldungen sollten nochmal erstellt werden	PROGRAMMED	tkamq	ENABLED
18	camt.005 - Einzelabfrage am nächst...	pac0.008 - Verarbeitet am Tag x	MESSAGING	tka63	ENABLED
20	adml.004 - Tageswechsel zu vergan...	adml.004 - Tageswechsel zu vergangennem Tag	MESSAGING	tk7p3	ENABLED

Einheitlicher Header

Applikations-  
getriebene  
Navigation

Applikations-Inhalte

Head

# DIMENSIONS

The screenshot shows a web page for the SIX Design System. On the left is a vertical navigation menu with the following items: Roadmap, Getting Started, Brand Identity (highlighted), Colors, Design Tokens, Responsive Design, Spacing, Theming, Typography, Components, Guidelines, Showcase, FAQs, and Contact Us. The main content area features a search bar and navigation links for Guidelines, Components, and Web Components v0.0.1. The 'Spacing' section is highlighted, containing the following text:

**Spacing**

The SIX Design System was created to help us build products for our customers more efficiently. Established to be the single source of truth for product teams working on SIX products across the globe, the SIX Design System contains everything from Getting Started guides through to Core Components intended for adoption within each product. It enables product teams to design and develop with speed, consistency, and quality. Our dedicated Design Operations team works with the individual product teams across SIX to ensure the system continues to evolve to meet the needs of our designers, developers, and customers.

**Agnostic**

Our Library works with many popular frameworks like Angular, react and Vue right out of the box.

**Fast Runtime**

Web Components offers a solution by pushing more work to the browser for better performance.

**In-House Open Source**

All developers within SIX are welcome to contribute code and use our library in their projects.

ans 14px, Web Rock 60x

Custody

ty

Tal  
(78

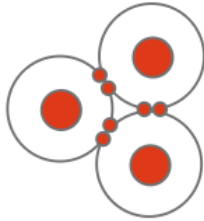
Dimer

# GUI-Komponenten

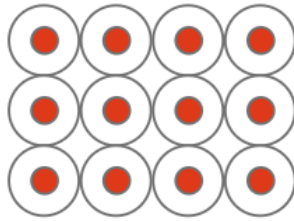
# Atomic Design



Atoms



Molecules



Organisms



Templates



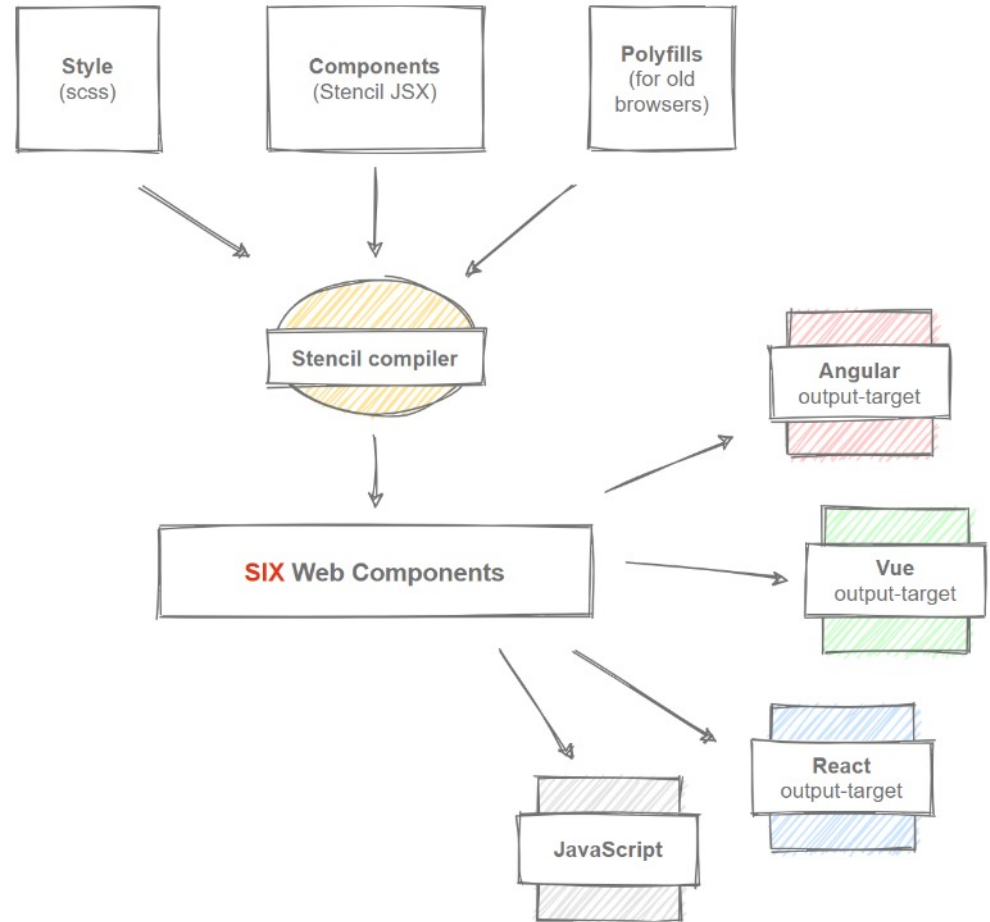
Pages

# Basierend auf Stencil

Standard-basiert

leichtgewichtig

Einfache Integration



# Internes Open-Source-Projekt

Pflege durch Core-Team

Code steht zur Verfügung, Pull-Requests via Git-Repo

[webcomponents.six-group.net](https://webcomponents.six-group.net)



# Demo-Time!





# Q&A



<https://www.linkedin.com/in/stefanrufer>



<https://www.linkedin.com/in/johann-latocha>



<https://www.linkedin.com/in/romanschlegel>

