



Meetup: QA All the Things

Visuelle Regressionstests
mit puppeteer und pixelmatch

Robert Seibert

Software Engineer

Springt freiwillig aus funktionierenden Flugzeugen und versucht seit über zwei Jahren sein altes 1200W Netzteil zu verkaufen.

inovex-Team Kollegen:

Johann Böhler, Sven Lindauer, Fabian Müller, Matthias Reuter, Johannes Reuter, Michael Wolber



ERGO Elements: React Komponentenbibliothek

ERGO Elements

Aktuell: 20 Monate im Projekt

- › 74 Komponenten
- › 5 Themes/Brands
- › In ca 24 Projekten eingebunden
- › Konzernweiter Einsatz bevorstehend

ERGO

Intermezzo:
UI Komponenten bei ERGO

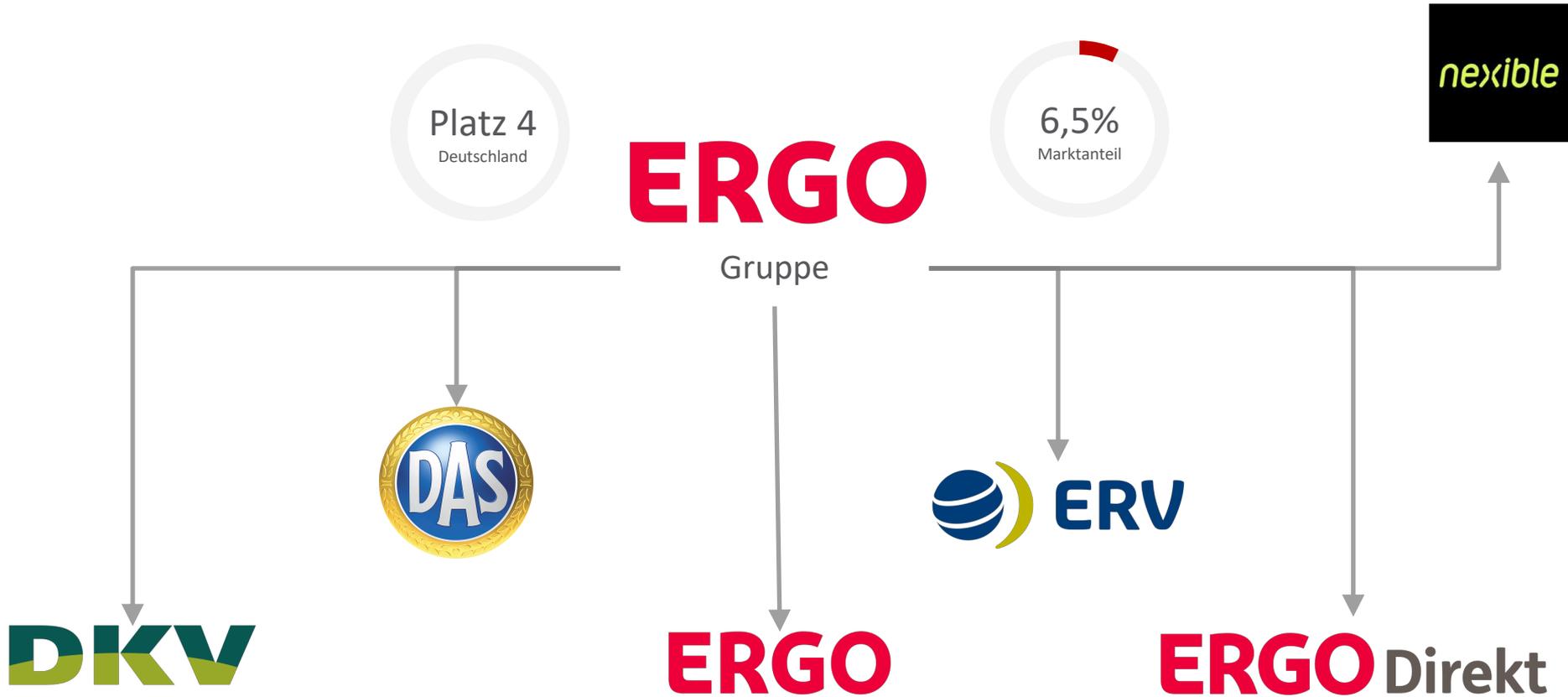


ERGO Direkt Versicherungen

2 yrs 8 mos

- **Chief Product Owner UI**
May 2018 – Present · 6 mos
Nürnberg Area, Germany
User Interface / Components & Plattformen
- **Product Owner UI**
Mar 2018 – Present · 8 mos
Nürnberg Area, Germany
- **User Experience International**
Mar 2016 – Mar 2018 · 2 yrs 1 mo
Nürnberg

Die Marken der Gruppe



ERGO national & international

ERGO

Gruppe

ERGO International

ERGO Deutschland

Digital Ventures

ES
FR
GR
LU
...

AT
BE
CH
EE
LT
RU
TUR, UKR
VT
...

BE
CDN
CZ
EE
ES
LU
PL
KR
...

DKV

ERGO

DAS

ERV

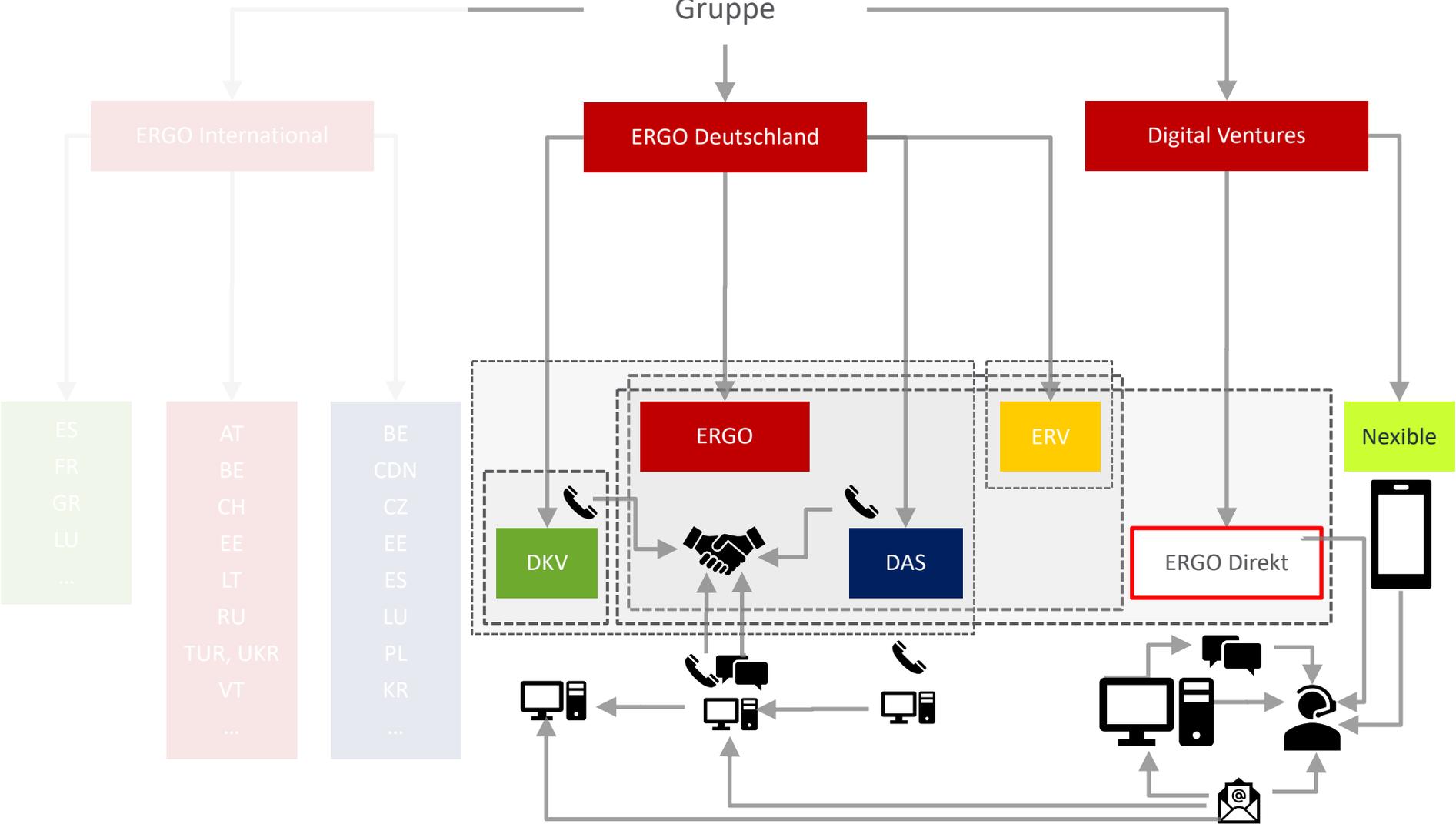
ERGO Direkt

Nexible

Versicherungen - Vertrieb

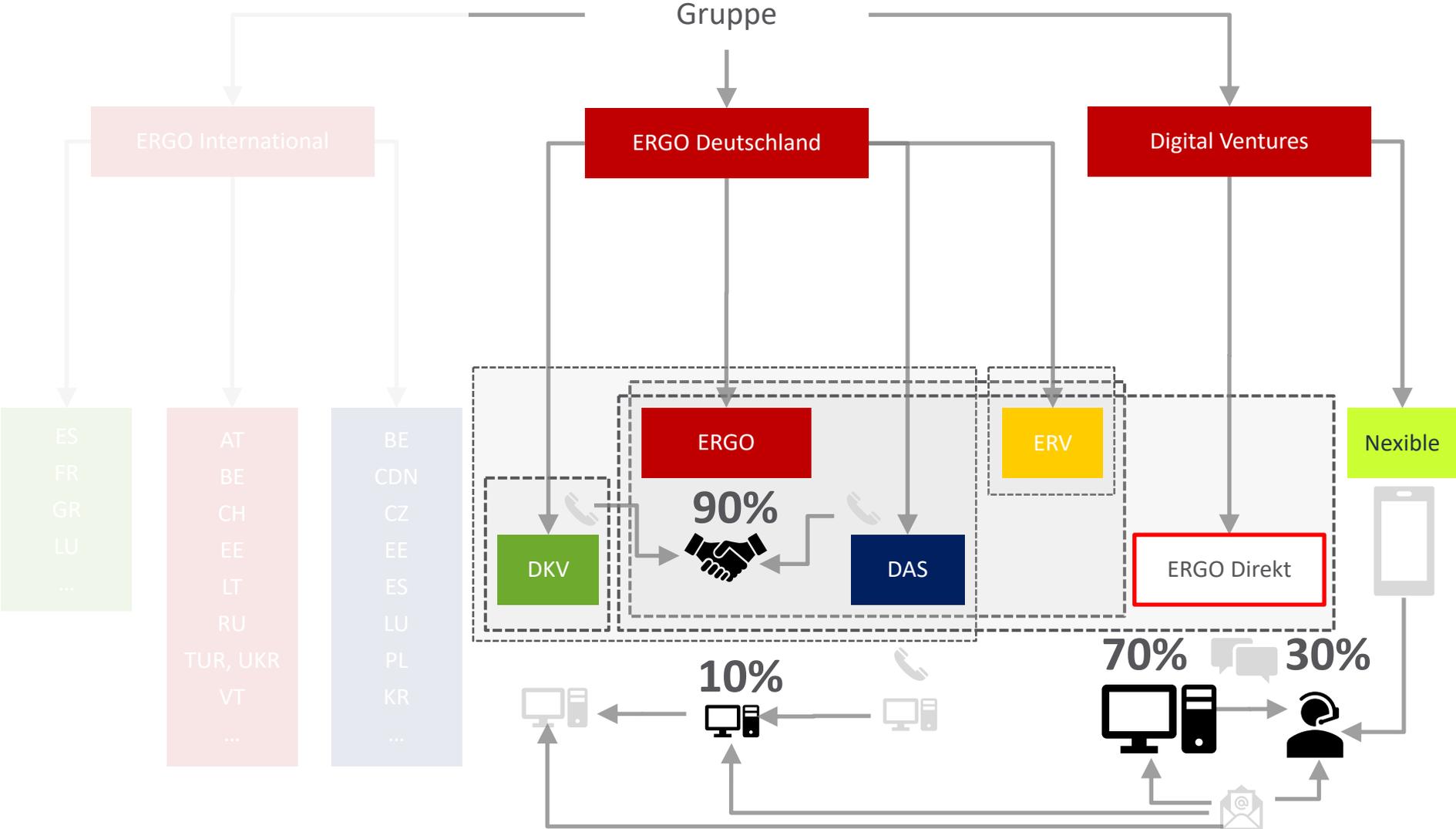
ERGO

Gruppe



Steuerung Omnikanal (ohne TV)

ERGO



IT Einheiten & Standorte



Hamburg

Berlin

Warschau

Düsseldorf

Nürnberg



Value Stream Transformation 2017

Hamburg

Warschau

Berlin

Düsseldorf



Nürnberg

“Deliver **customer value**
through a sequence of activities required to
design, produce and provide
a specific good or service
for the customer journey”

- **Reset** agile Teams
- Neue **Technologie**
- **Budget**

ERGO Elements 2018



4. SHARE
“The knowledge is out there,
just connect the dots and share it
with the right people”



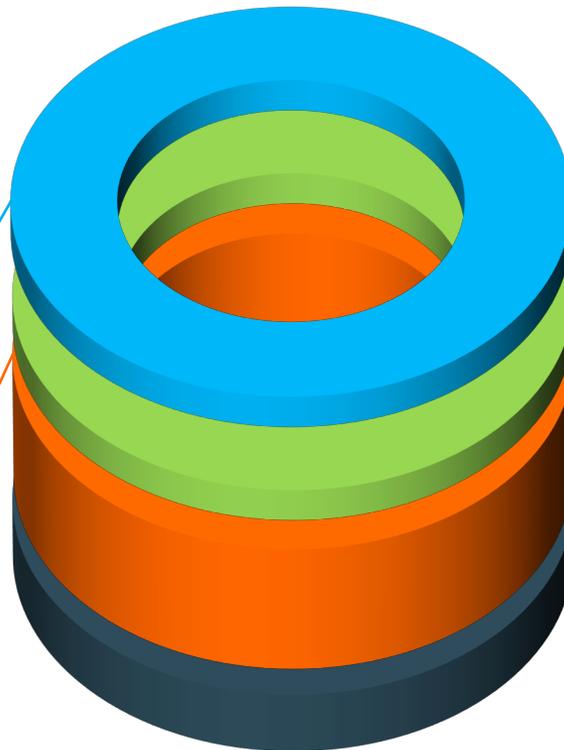
2. UI THEMES
ERGO Marken
“Ein Abschluss ist ein
Abschluss”



3. QUALITY
Hohe Testcoverage
“visuelle Regressionstests”



1. BASE
DEV pain point #1 2017
“implementiere den
Salesman Button”



ERGO Elements

Aktuell: 20 Monate im Projekt

- › 74 Komponenten
- › 5 Themes/Brands
- › In ca 40 Projekten eingebunden
- › Konzernweiter Einsatz bevorstehend
- › Bald: Weltherrschaft!!

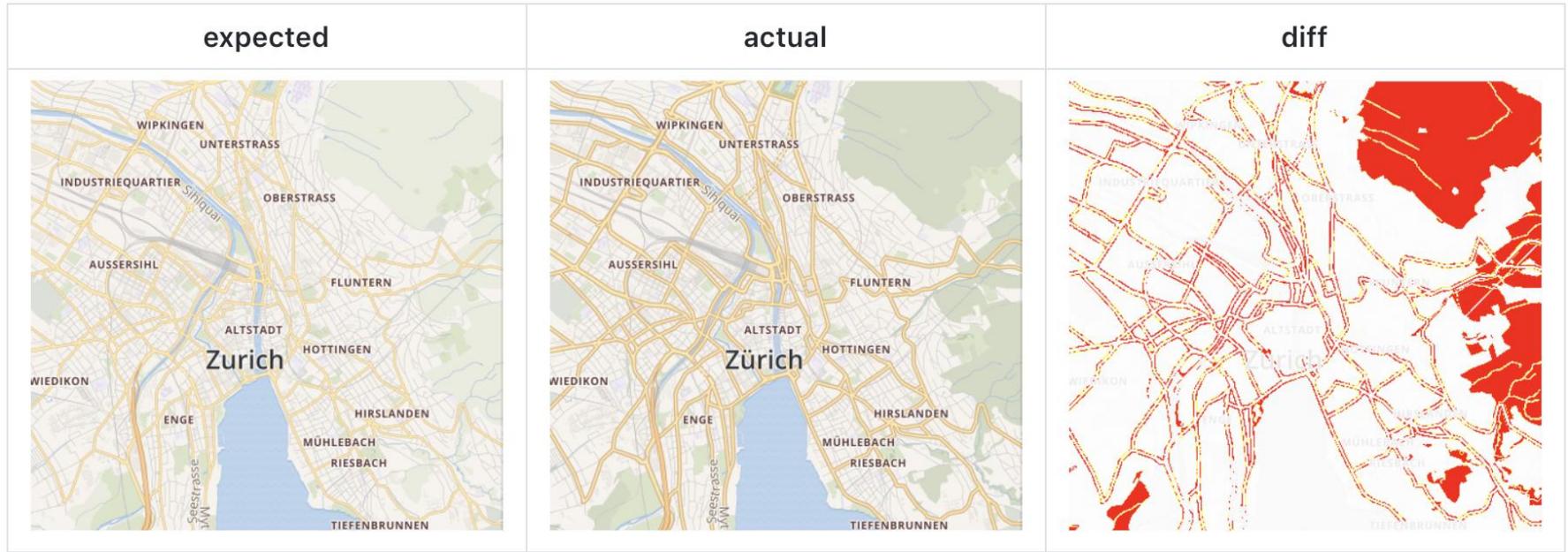
Weshalb eine Komponentenbibliothek?

- › Mehrfach-Aufwand sparen
- › Brand(Theme)-Fähigkeit
- › Kurze Time-to-Market
- › Hohe Qualität

Qualitätsanspruch

- › Hohe Abdeckung durch Unit-Tests (>95%)
- › Manuelle Tests in möglichst allen Browsern in der Definition of Done
- › Visuelle Konsistenz durch visuelle Regressionstests

Wie funktionieren visuelle Regressionstests?



Pain Points:

(6 Monate im Projekt)

- › Wachsende Komponentenzahl
- › Immer mehr Wechselwirkungen zwischen den Komponenten
- › Nicht mehr vollständig manuell testbar
- › Refactorings sehr aufwändig
- › Themes!!
- › 🤖 rutschen in die Releases

Puppeteer zur Rettung!

(12 Monate im Projekt)

- › Entwickler-freundliche API:
- › Low-Level Kontrolle
- › Kann Screenshots machen
- › Läuft stabil (?)

```
const puppeteer = require('puppeteer');

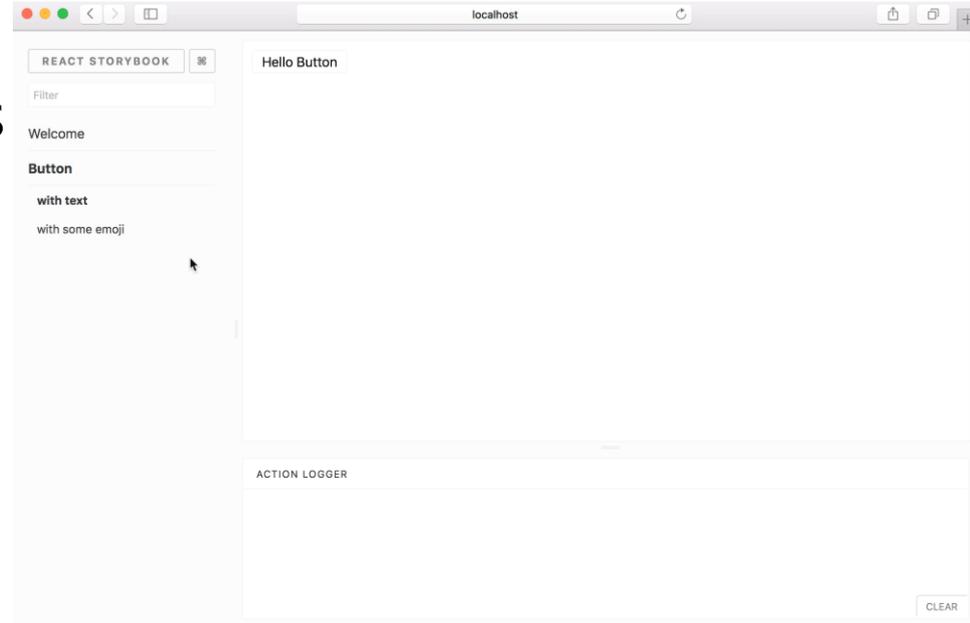
(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});

  await browser.close();
})();
```

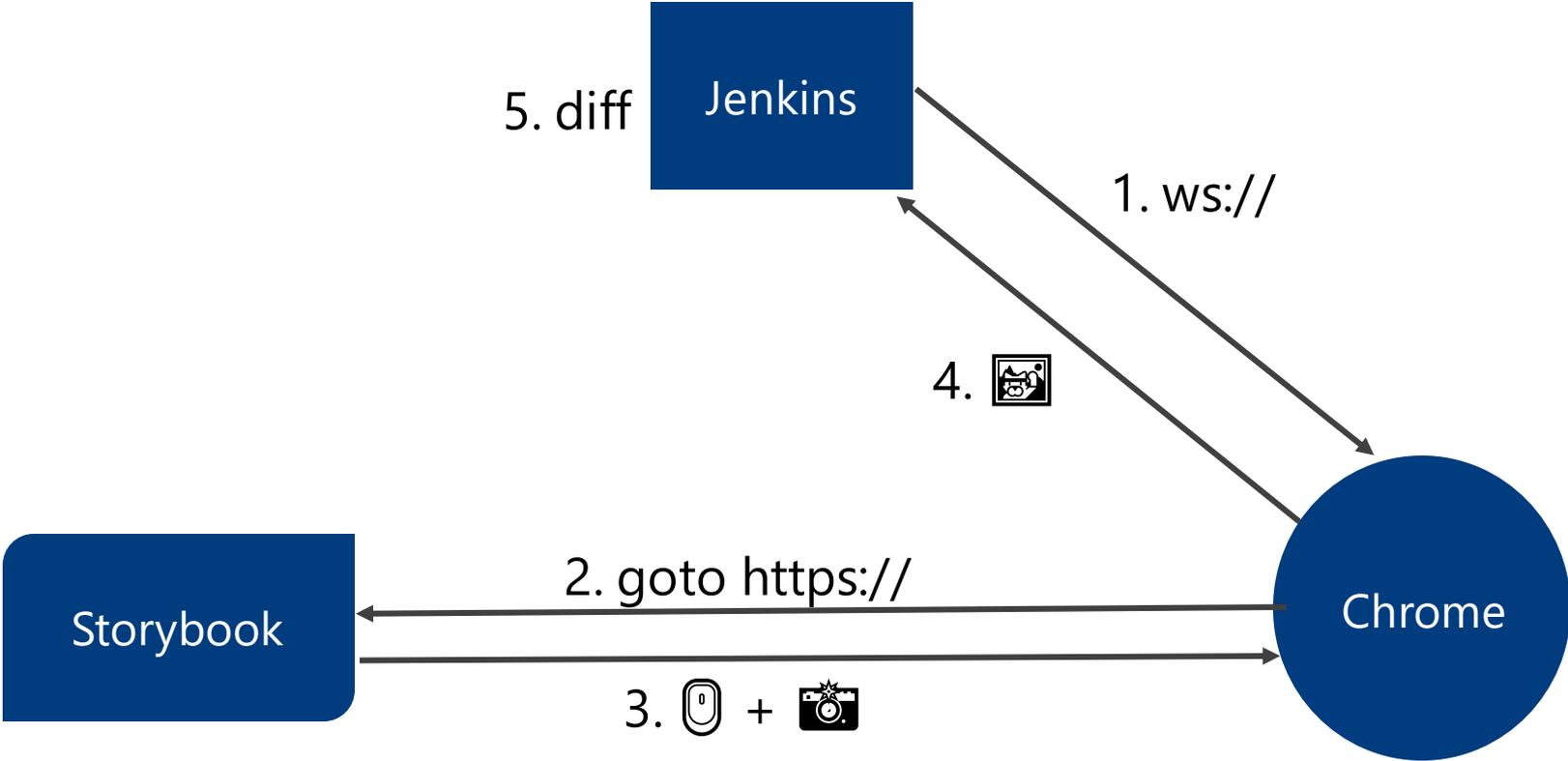
Setup-Vorbereitung

Storybook:

- › Minimalistisch, ohne Docs
- › Im `<iframe/>`
- › Theme-Switcher Panel
- › Leicht modifizierbar

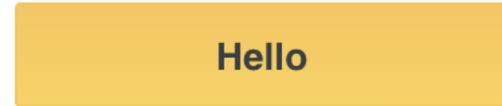


Setup

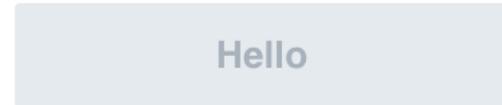


Die ersten Bilder

- › Button in verschiedenen Zuständen (hover, focus, ...)
- › CSS-Änderungen lassen Test fehlschlagen



Hello



Viewports und Themes

```
for component in COMPONENTS do
  for viewport in VIEWPORTS do
    for theme in THEMES do
      for states in STATES do
        test_and_compare()
      end
    end
  end
end
end
```

Erste Performance Learnings

- › Tests sequentiell laufen lassen
- › Aufräumen nicht vergessen
- › Nicht mit RAM und CPU des Chrome Containers geizen
- › Schatten und Animation führen manchmal zu false positives!

Schatten und Animation

- › Per Storybook Addon deaktiviert

```
1  .Storybook-DisableCSSAnimations *,
2  .Storybook-DisableCSSAnimations *::after,
3  .Storybook-DisableCSSAnimations *::before
4  {
5      transition: none !important;
6      animation-duration: 0ms !important;
7  }
```

- › Dadurch werden Timing-Fehler vermieden, kein "waitFor" notwendig

Storybook-Container-Anpassung

oder: die lästige Scrollbar

- › Per Storybook Addon overflow deaktiviert

```
1  div#root {  
2      overflow-x: hidden;  
3      box-sizing: border-box;  
4      padding-bottom: 2px;  
5  }
```

- › Chrome flag: `--hide-scrollbar`

Weitere Performance Learnings

- › Pixelmatch tuning: `Buffer.equals` vs Array Iteration
- › Reduktion der Pipeline-Dauer um 50% von 11min auf ca. 5min
- › Performance-Fact: ca. 1300 Bilder in 5 Minuten!

Weiter mit input Element

- › "focus" 

- › Blinking cursor!

Lösung: Chrome DevTools Protocol

CSS.forcePseudoState

Ensures that the given node will have specified pseudo-classes whenever its style is computed by the browser.

PARAMETERS

nodeId [DOM.NodeId](#)

The element id for which to force the pseudo state.

forcedPseudoClasses **array [string]**

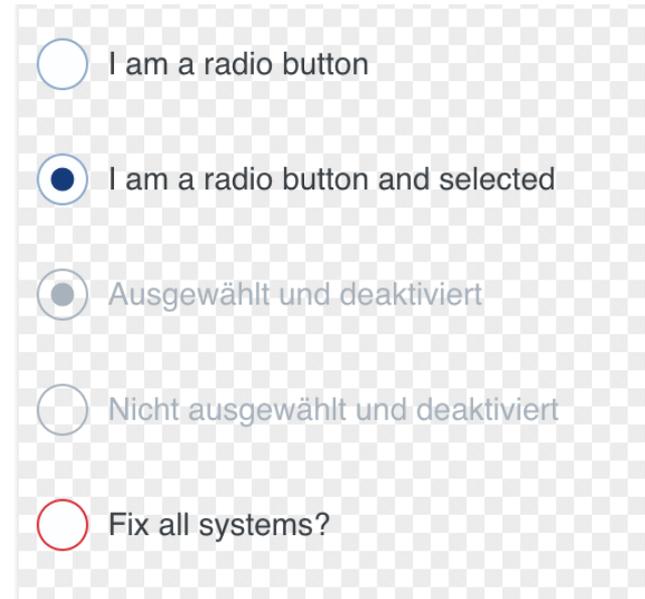
Element pseudo classes to force when computing the element's style.

Mehr Komponenten!

Was ist mit Transparenz?

› Storybook Addon für Checkboard Hintergrund

```
9 @checkerboard__background-color: rgba(228, 228, 228, 0.7);
10
11 .Storybook-TransparentBackground {
12     background-image: linear-gradient(45deg,
13         @checkerboard__background-color 25%,
14         transparent 25%,
15         transparent 75%,
16         @checkerboard__background-color 75%,
17         @checkerboard__background-color
18     ), linear-gradient(45deg,
19         @checkerboard__background-color 25%,
20         transparent 25%,
21         transparent 75%,
22         @checkerboard__background-color 75%,
23         @checkerboard__background-color
24     );
25     background-size: 16px 16px;
26     background-position: 0 0, 8px 8px;
27 }
```



Noch mehr Komponenten!!

Was ist mit Timeouts?

- › Timeouts müssen abgewartet werden (debounces auf resize oder scroll Events)
- › Idealerweise mit "waitForSelector" anstatt "waitFor"

Noch mehr Komponenten!!!

Was ist mit Maus Gesten?

› Puppeteer kann auch Gesten 🧐

```
const tileElement = await group.$('.ee_radio-tile');
const boundingBox = await tileElement.boundingBox();
const mouse = page.mouse;

// click on the first radio tile and drag to the left,
// simulating a swipe left gesture
await mouse.move(boundingBox.x + boundingBox.width - 5, boundingBox.y + boundingBox.height/2);
await mouse.down();
await mouse.move(boundingBox.x - 5, boundingBox.y + boundingBox.height/2);
await mouse.up();
```

Hat das ganze wenigstens etwas gebracht?

- › Absolut! Alleine Tests anlegen legt "Bugs" offen
- › Wir finden Bugs bevor es andere tun. Visuelle Bugs wurden extrem selten
- › Releases kein Schuss ins Blaue mehr

Nach und nach...

Jetzt sind wir doch maximal abgestet, oder?

- › Alte Komponenten mit visuellen Tests nachrüsten bleibt gerne im Backlog liegen
- › Möglichst früh einplanen oder in Downtimes nachziehen!

Dinge zu Beachten #2

Einzelne Slices in den Stories

Testlabel 

value

additional description

Testlabel 

value

Testlabel 

value

Testlabel 

value

Testlabel 

value

Testlabel 

value

Dinge zu Beachten #3

Interpretieren von Diffs

<table><thead><tr><th colspan="2">VW Golf</th></tr></thead><tbody><tr><td>HP</td><td>180</td></tr><tr><td>Cubic capacity</td><td>1798 cm³</td></tr><tr><td></td><td><input type="radio"/></td></tr></tbody></table>	VW Golf		HP	180	Cubic capacity	1798 cm ³		<input type="radio"/>	<table><thead><tr><th colspan="2">Audi S3</th></tr></thead><tbody><tr><td>HP</td><td>310</td></tr><tr><td>Cubic capacity</td><td>1984 cm³</td></tr><tr><td></td><td><input checked="" type="radio"/></td></tr></tbody></table>	Audi S3		HP	310	Cubic capacity	1984 cm ³		<input checked="" type="radio"/>	<table><thead><tr><th colspan="2">Seat Leon</th></tr></thead><tbody><tr><td>HP</td><td>115</td></tr><tr><td>Cubic capacity</td><td>999 cm³</td></tr><tr><td></td><td><input type="radio"/></td></tr></tbody></table>	Seat Leon		HP	115	Cubic capacity	999 cm ³		<input type="radio"/>
VW Golf																										
HP	180																									
Cubic capacity	1798 cm ³																									
	<input type="radio"/>																									
Audi S3																										
HP	310																									
Cubic capacity	1984 cm ³																									
	<input checked="" type="radio"/>																									
Seat Leon																										
HP	115																									
Cubic capacity	999 cm ³																									
	<input type="radio"/>																									
<table><thead><tr><th colspan="2">VW Golf</th></tr></thead><tbody><tr><td>HP</td><td>180</td></tr><tr><td>Cubic capacity</td><td>1798 cm³</td></tr><tr><td></td><td><input type="radio"/></td></tr></tbody></table>	VW Golf		HP	180	Cubic capacity	1798 cm ³		<input type="radio"/>	<table><thead><tr><th colspan="2">Audi S3</th></tr></thead><tbody><tr><td>HP</td><td>310</td></tr><tr><td>Cubic capacity</td><td>1984 cm³</td></tr><tr><td></td><td><input checked="" type="radio"/></td></tr></tbody></table>	Audi S3		HP	310	Cubic capacity	1984 cm ³		<input checked="" type="radio"/>	<table><thead><tr><th colspan="2">Seat Leon</th></tr></thead><tbody><tr><td>HP</td><td>115</td></tr><tr><td>Cubic capacity</td><td>999 cm³</td></tr><tr><td></td><td><input type="radio"/></td></tr></tbody></table>	Seat Leon		HP	115	Cubic capacity	999 cm ³		<input type="radio"/>
VW Golf																										
HP	180																									
Cubic capacity	1798 cm ³																									
	<input type="radio"/>																									
Audi S3																										
HP	310																									
Cubic capacity	1984 cm ³																									
	<input checked="" type="radio"/>																									
Seat Leon																										
HP	115																									
Cubic capacity	999 cm ³																									
	<input type="radio"/>																									



Empfehlung

- › Wenn es schnell gehen muss, alles auf eine Seite
- › Wenn Zeit vorhanden, Slices in eigene Seiten auslagern

Erfolgsszenarios

(nur ein kleiner Auszug)

- › CSS/LESS Refactoring
- › React 16.3. Migration mit neuer Context API
- › Pixel-Schubsen im Form-Layout

Alles hat seinen Preis

- › Ähnlich aufwändig wie E2E Tests
- › Führt gerne zu mehreren Work-Review-Zyklen
- › Erfordert viel Disziplin im aktuell-halten der Referenzen
- › Test-Pyramide: Unittests sind wichtiger!

Ausblick & Blindspots

- › Andere Browser nicht abgedeckt
- › CSS Leaks mit third party CSS
- › Vollabdeckung erfordert hohe Laufzeit
- › Rendering sehr Setup-abhängig
- › Lokal nicht sofort testbar

Fragen?

Referenzen

- › [pixelmatch](#)
- › [Google puppeteer](#)
- › [React Styleguidist](#)
- › [Storybook](#)

Alternative Tools:

- › [Cypress.io](#)
- › [BackstopJS](#)

Vielen Dank

Robert Seibert

Software Engineer

inovex GmbH

Ludwig-Erhard-Allee 6

76131 Karlsruhe

robert.seibert@inovex.de

0173 3181 169

