APACHE SLING & FRIENDS TECH MEETUP

2 - 4 SEPTEMBER 2019

Using React to enable Web Developers to work with AEM

T. Schumacher, TECLEAD

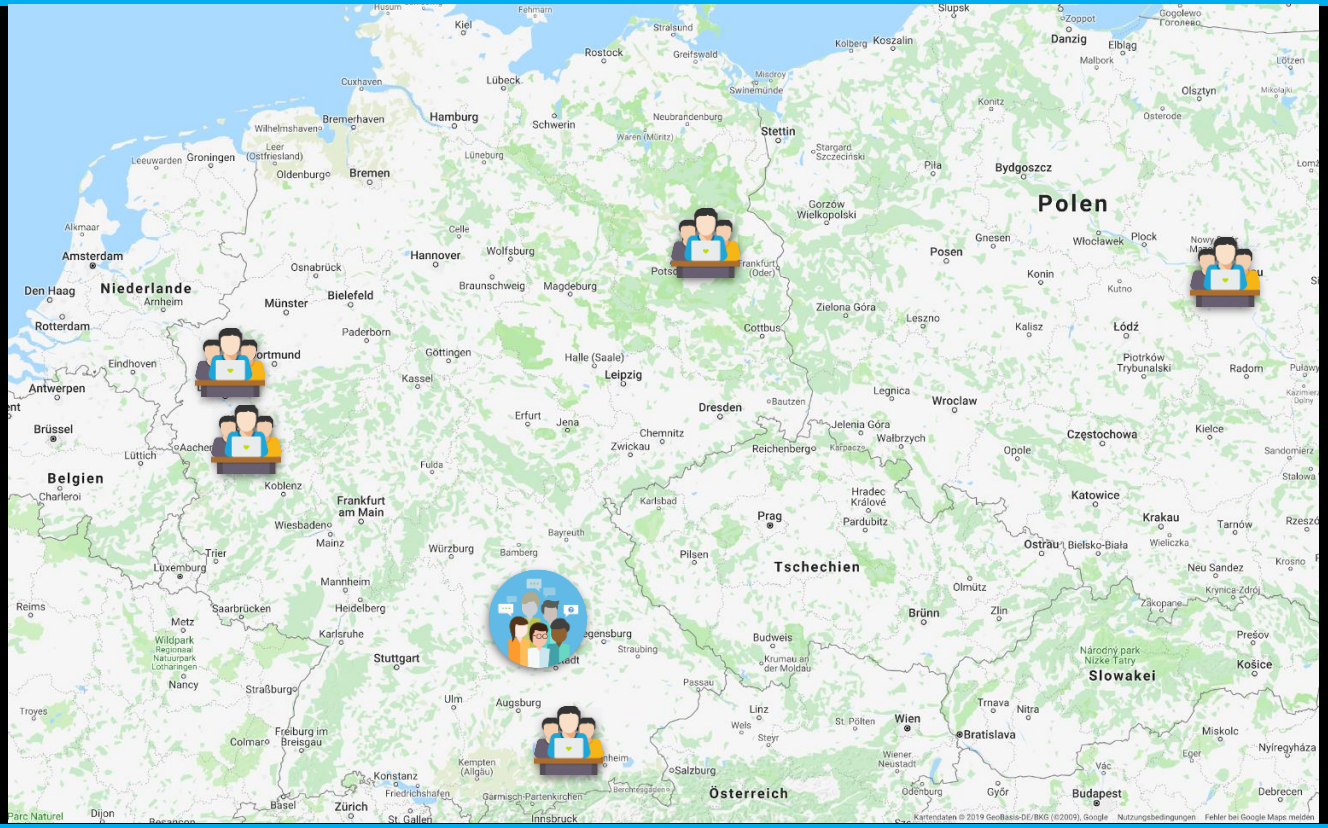
adaptTo()

Topics

- Project Situation
- Challenge
- Solution
  - Boilerplate
  - AEM Generator
  - Decoupled React Components
  - Custom Dialogs
- What we learned / Future

Project Situation - Distributed Teams in an enterprise context

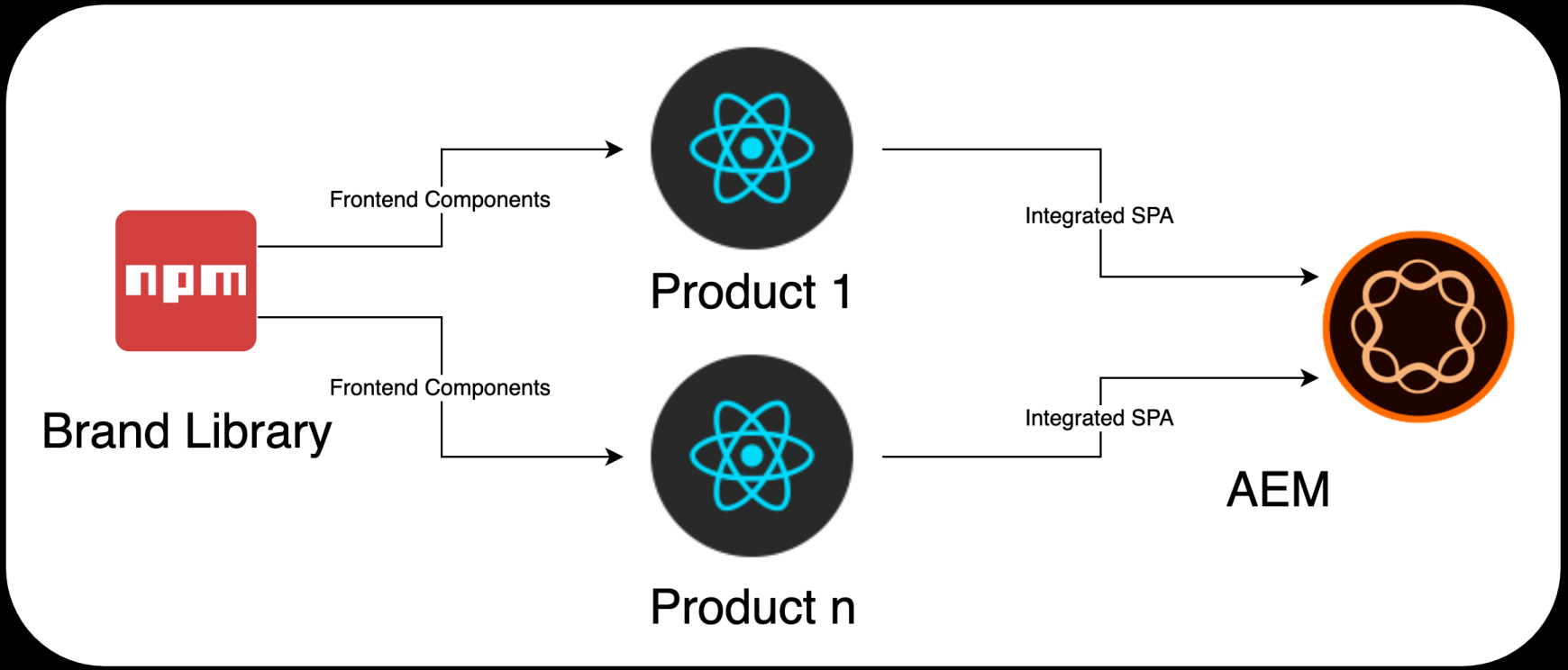
Distributed Teams in an enterprise context



Distributed Teams in an enterprise context

- language barrier
- different know how
- only high level architecture provided
- multiple locations and teams
- Different brands, same components
- external frontend library based on React

Distributed Teams in an enterprise context



adaptTo()

What is the challenge

What is the challenge

- no AEM knowhow in internal teams
- redundant frontend code
- tight deadlines → no time / resources for coaching
- internal acceptance for AEM as the main technology

Solution

adaptTo()

Solution

- AEM Generator
- Decoupled REACT Components
- SEO by SSR
- Custom Dialogs (with React)
- Boilerplate - Project structure and tooling

Boilerplate - Project structure and tooling

- frontend
  - react-components (with corporate library)
  - Typescript (tracking lib)
  - stylesheets
  - storybook
- Java
- e2e (automated cross browser tests with Selenium)
- test-content
- gradle

HeaderMainPage

With two test seals

In this example two test seals is displayed. Unfortunately, we cannot make the array of seals configurable.

Story Source

Copy

```
<HeaderMainPage
  teaserImage="/teaser.png"
  eyeCatcher={{imageUrl: '/eyecatcher.svg',altText: 'altText'}}
  testSeals={['/Stiftungwt.png','/Money.jpg']}
  headlinePrimary="Produkt-Header Dachzeile h1"
```

Boilerplate – React Components

„Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.“

Elon Musk **Founder & CEO**

Boilerplate – React Components

- React Component properties ≈ Dialog properties

```
export interface BlockquoteProps extends GeneralAemReactComponentProps {
  imageUrl?: string;
  author?: string;
  authorJob?: string;
  text: string;
  backgroundColor?: string;
}
```

Boilerplate – React Components

- Usage of the corporate library

```
<Provider theme={theme}>
  <BlockquoteEE
    img={new ImageLoader({ imageComponents: imageUrl }).desktopUrl}
    author={
      <>
        {author} <strong>{authorJob}</strong>
      </>
    }
  >
    {text}
  </BlockquoteEE>
</Provider>
```

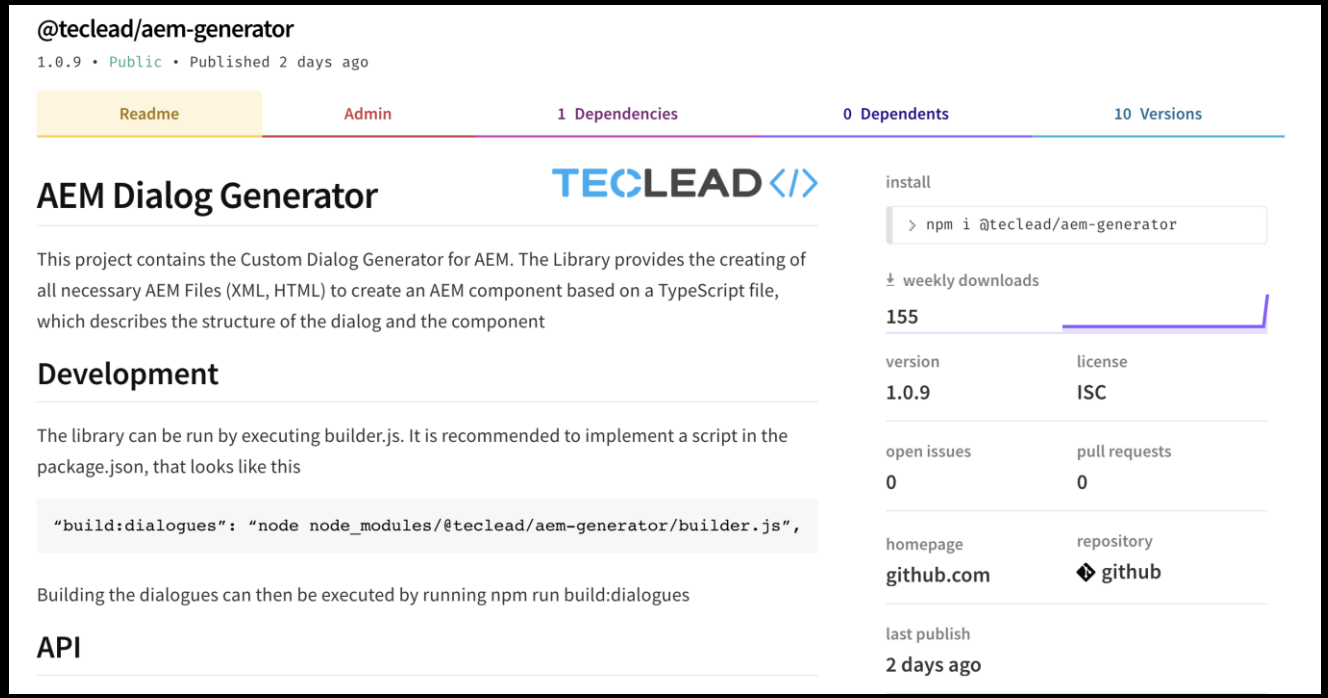
AEM Generator - and what it does

adaptTo()

AEM Generator

- NPM / Typescript
- Is used in the build process only
- Generates all needed AEM sources
- Open source + extending in client context

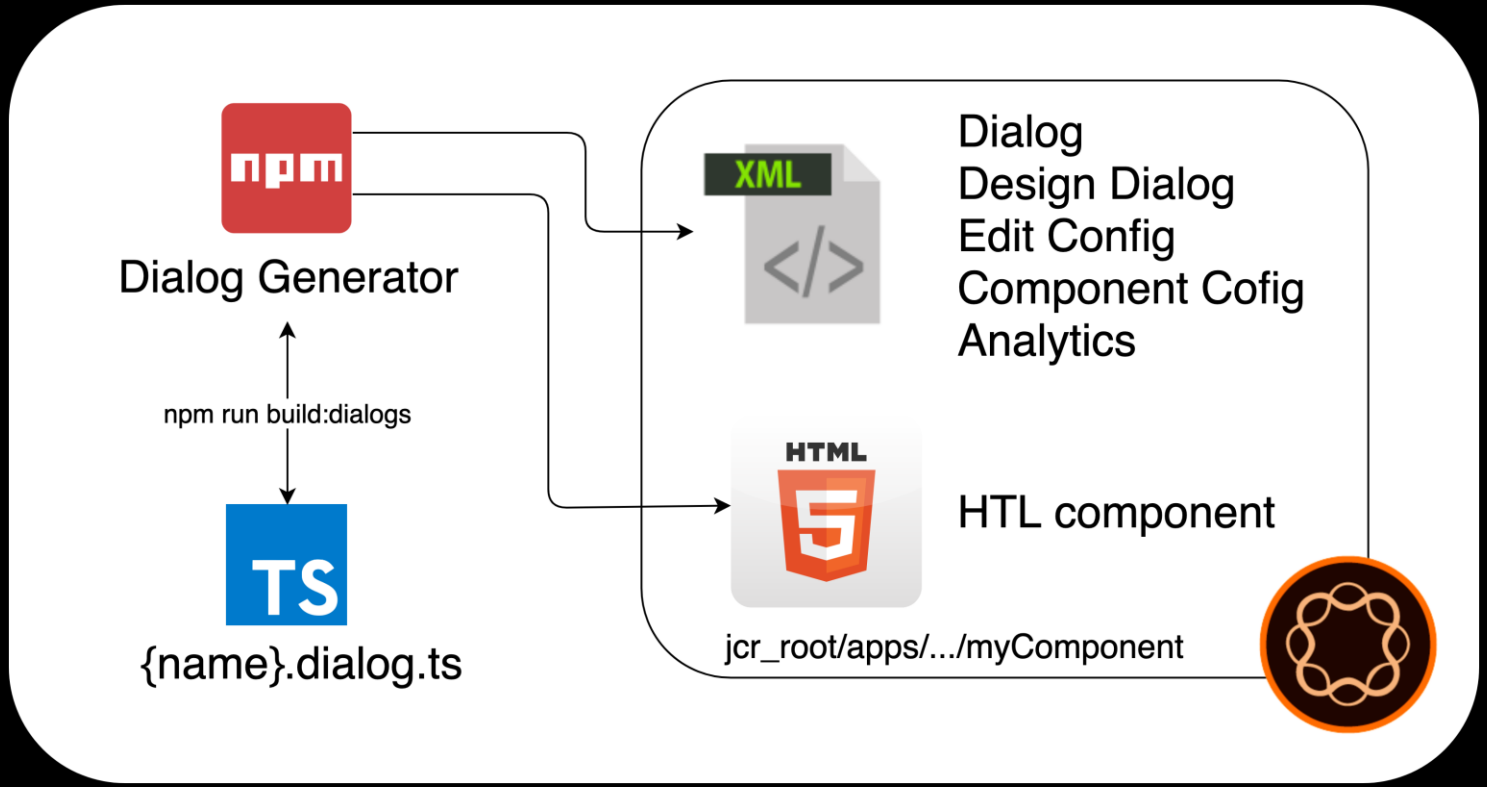
AEM Generator



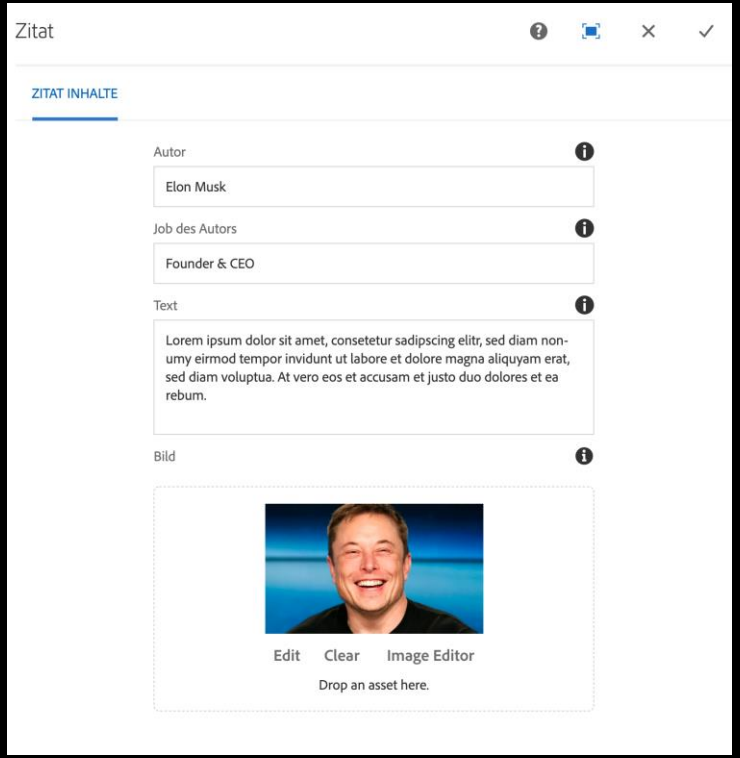
Simple demo project: https://github.com/Teclead/AdaptTo2019

Get it @NPM https://www.npmjs.com/package/@teclead/aem-generator

AEM Generator



AEM Generator



AEM Generator

- Is used in the build process only
- Generates all needed AEM sources

```
const dialog: AEMTouchUIDialog = {
  componentPath: COMPONENTPATH + 'blockquoteReact',
  componentName: 'Quote',
  componentGroup: COMPONENT_GROUP,
  componentDescription: 'A React Component for a quote',
  buildConfigForAem: true,
  reactAppName: 'BlockquoteComponent',
  tabs: [{ title: 'Quote Component', fields: dialogFields }]
 };
```

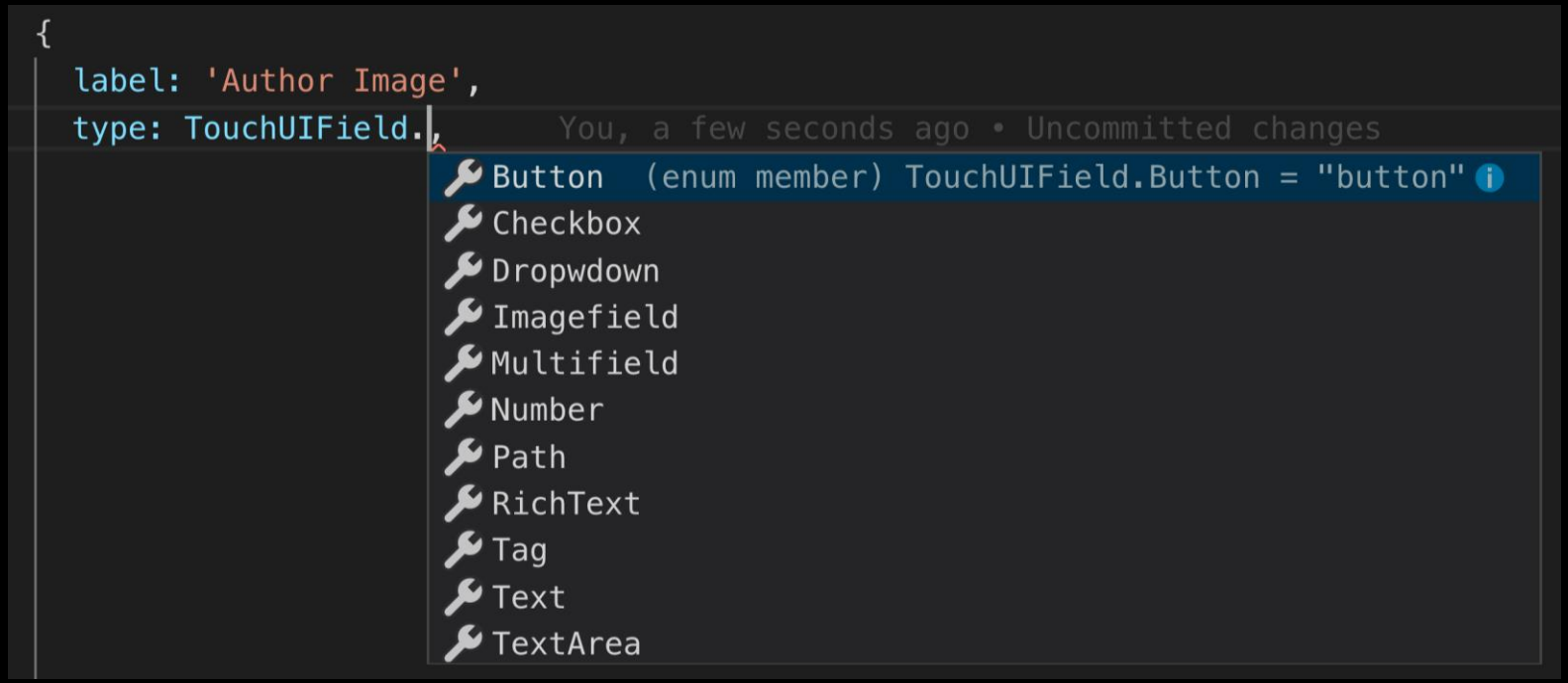
AEM Generator

- Dialog fields ≈ React Props
- TS helps for validation in the IDE

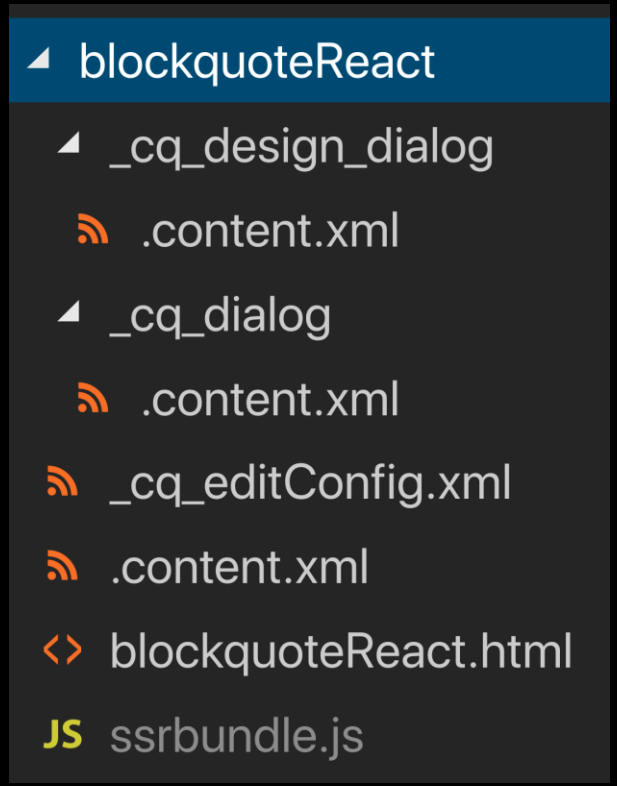
```
const dialogFields: TouchUIDialogField[] = [
  {
    label: 'Name',
    type: TouchUIField.Text,
    key: 'author',
    isRequired: true,
    description: 'Name of the author'
  },
  {
    label: 'Author Image',
    type: TouchUIField.Imagefield,
    key: 'imageUrl',
  }
];
```

AEM Generator

- Field types with IDE support, Based on TS enum



Dialog Generator



- Generates all:
  - Folders and xml files
  - Dialogs (with functions)
  - Analytics
  - HTL template
  - SSR (project specific)

- Easy to migrate components
  (e.g. Classic UI to Touch UI)

AEM Generator

- FAQ:
  - Can we use component inheritance?
  - What about custom dialogs?
  - Is it mandatory for every component?
  - What types are supported?
    (e.g. nested multifields)

Decoupled REACT Components

Decoupled REACT Components

- HTL template:
  - Same for each AEM React component
  - No data fetch needed
  - Uses SSR
- Separate JS build vs clientlibs
- Can be used in outside of AEM
- Can be mixed with plain HTL components

Decoupled REACT Components

```
<head>
    <link href="navi.css">
    <link href="app-1.css">
    <link href="app-2.css">
    <link href="footer.css">
    <script src="/polyfill.js"></script>
</head>

<body>
    <navi-react />
    <react-app-1 />
    <react-app-n />
    <footer-react />
    <script src="navi.js"></script>
    <script src="app-1.js"></script>
    <script src="app-2.js"></script>
    <script src="footer.js"></script>
</body>
```

Decoupled REACT Components

```
<head>
    <link href="navi.css">
    <link href="app-1.css">
    <link href="app-2.css">
    <link href="footer.css">
    <script src="/polyfill.js"></script>
</head>

<body>
    <navi-react />
    <react-app-1 />
    <react-app-n />
    <footer-react />
    <script src="navi.js"></script>
    <script src="app-1.js"></script>
    <script src="app-2.js"></script>
    <script src="footer.js"></script>
</body>
```

SSR Component scans for used apps(Server)

Decoupled REACT Components

```
<head>
    <link href="navi.css">
    <link href="app-1.css">
    <link href="app-2.css">
    <link href="footer.css">
    <script src="/polyfill.js"></script>
</head>

<body>
    <navi-react />
    <react-app-1 />
    <react-app-n />
    <footer-react />
    <script src="navi.js"></script>
    <script src="app-1.js"></script>
    <script src="app-2.js"></script>
    <script src="footer.js"></script>
</body>
```

SSR Component scans for used apps(Server)

injects data and gets markup

Decoupled REACT Components

What happens on the client?

Decoupled REACT Components

```
<sly data-sly-use.ssr="com.ServerSideRenderingComponent">
    <div class="aem-react-component BlockquoteComponent"
        data-params="${ssr.allProperties}">
        ${ssr.renderedHtml @ context = 'unsafe'}
    </div>
</sly>
```

Decoupled REACT Components

- React loader looks for indicator (Client)

```
<sly data-sly-use.ssr="com.ServerSideRenderingComponent">
    <div class="aem-react-component BlockquoteComponent"
        data-params="${ssr.allProperties}">
        ${ssr.renderedHtml @ context = 'unsafe'}
    </div>
</sly>
```

Decoupled REACT Components

- React loader looks for indicator
- Finds the right React App to render

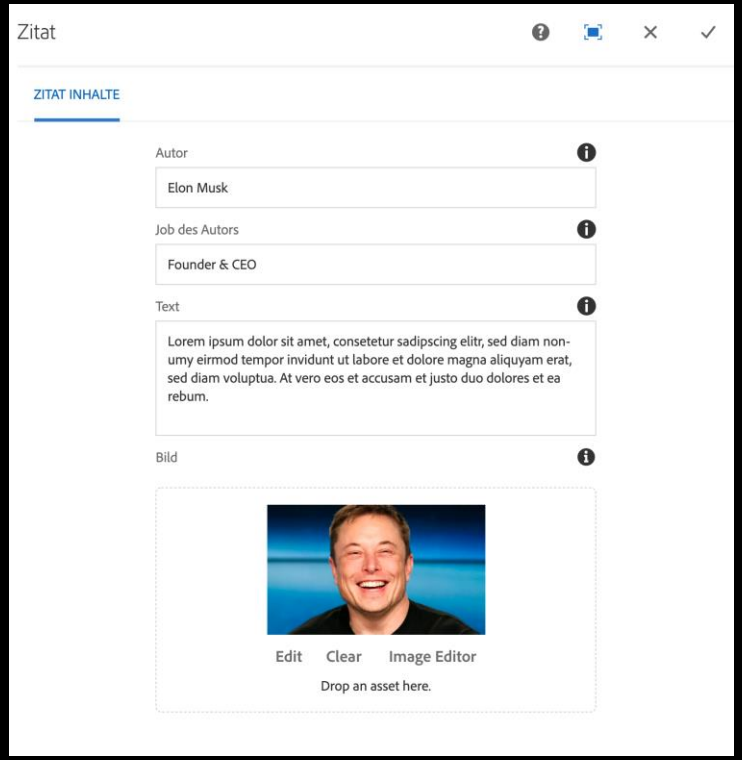
```
<sly data-sly-use.ssr="com.ServerSideRenderingComponent">
    <div class="aem-react-component BlockquoteComponent"
        data-params="${ssr.allProperties}">
        ${ssr.renderedHtml @ context = 'unsafe'}
    </div>
</sly>
```

Decoupled REACT Components

- React loader looks for indicator
- Finds the right React App to render
- Injected data from JCR

```
<sly data-sly-use.ssr="com.ServerSideRenderingComponent">
    <div class="aem-react-component BlockquoteComponent"
        data-params="${ssr.allProperties}">
        ${ssr.renderedHtml @ context = 'unsafe'}
    </div>
</sly>
```

Decoupled REACT Components




```
{
  author: "Elon Musk",
  imageUrl: [{...}],
  text: "Lorem Ipsum...",
  sling:resourceType: "apps/...//blockquoteReact",
  authorJob: "Founder & CEO",
  resourcePath: "/content/.../jcr:content/root/blockquotereact",
  isEditMode: true,
  serviceTime: '08:00-19:00',
  phoneNumber: '0123',
  systemStyle: {
    paddingType: "large",
    padding: "top",
    position: "center",
    backgroundColor: "white"
  }
}
```

Decoupled REACT Components

```
<Provider theme={theme}>
  <BlockquoteEE
    img={new ImageLoader({ imageComponents: imageUrl }).desktopUrl}
    author={
      <>
        {author} <strong>{authorJob}</strong>
      </>
    }
  >
    {text}
  </BlockquoteEE>
</Provider>
```

```
{
  author: "Elon Musk",
  imageUrl: [{...}],
  text: "Lorem Ipsum...",
  sling:resourceType: "apps/...//blockquoteReact",
  authorJob: "Founder & CEO",
  resourcePath: "/content/.../jcr:content/root/blockq
  isEditMode: true,
  serviceTime: '08:00-19:00',
  phoneNumber: '0123',
  systemStyle: {
    paddingType: "large",
    padding: "top",
    position: "center",
    backgroundColor: "white"
  }
}
```

Decoupled REACT Components

```
<Provider theme={theme}>
  <BlockquoteEE
    img={new ImageLoader({ imageComponents: imageUrl }).desktopUrl}
    author={
      <>
        {author} <strong>{authorJob}</strong>
      </>
    }
  >
    {text}
  </BlockquoteEE>
</Provider>
```

```
{
  author: "Elon Musk",
  imageUrl: [{...}],
  text: "Lorem Ipsum...",
  sling:resourceType: "apps/...//blockquoteReact",
  authorJob: "Founder & CEO",
  resourcePath: "/content/.../jcr:content/root/blockq
  isEditMode: true,
  serviceTime: '08:00-19:00',
  phoneNumber: '0123',
  systemStyle: {
    paddingType: "large",
    padding: "top",
    position: "center",
    backgroundColor: "white"
  }
}
```

Decoupled REACT Components

„Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.“

Elon Musk **Founder & CEO**

Custom Dialogs

Custom Dialogs

- Needed for complex authoring use cases
- Better UX for authors
- Examples:
  - large icon sets
  - complex tables
  - more complex image component

Allgemein		Paket - Basis	Paket - Komfort	Paket - Super
Aufnahme ohne Gesundheitsfragen		✓	✓	✓
Regelversorgung bei Zahnersatz		**100%** Erstattung	**100%**	**100%** Erstattung
Privatzahnärztliche Versorgung bei Zahnersatz		**75%** Erstattung	**90%**	**75%** Erstattung
Implantate		**75%** Erstattung	**90%**	**75%** Erstattung
Weitere Leistungen		**75%** Erstattung	**90%**	**75%** Erstattung

Allgemein

Aufbauzeit und Leistungseinschränkungen		Bei einem Unfall besteht sofort Anspruch auf die Tarifleistungen in voller Höhe. Ansonsten gelten nur während der ersten 4 Versicherungsjahre je Tarif und versicherter Person folgende Leistungsbegrenzungen: Im ersten Versicherungsjahr höchstens 250 € In den ersten beiden Versicherungsjahren zusammen höchstens 500 € In den ersten 3 Versicherungsjahren zusammen höchstens 750 € In den ersten 4 Versicherungsjahren zusammen höchstens 1.000 € Mit Ablauf des vierten Versicherungsjahres bestehen die summenmäßigen Begrenzungen der Tarifleistungen nicht mehr.	Nur für gesetzlich Versicherte

Produkt-Tabelle

Vorschau | Tabelle speichern

Tabelle editieren:

☑ Sticky Header

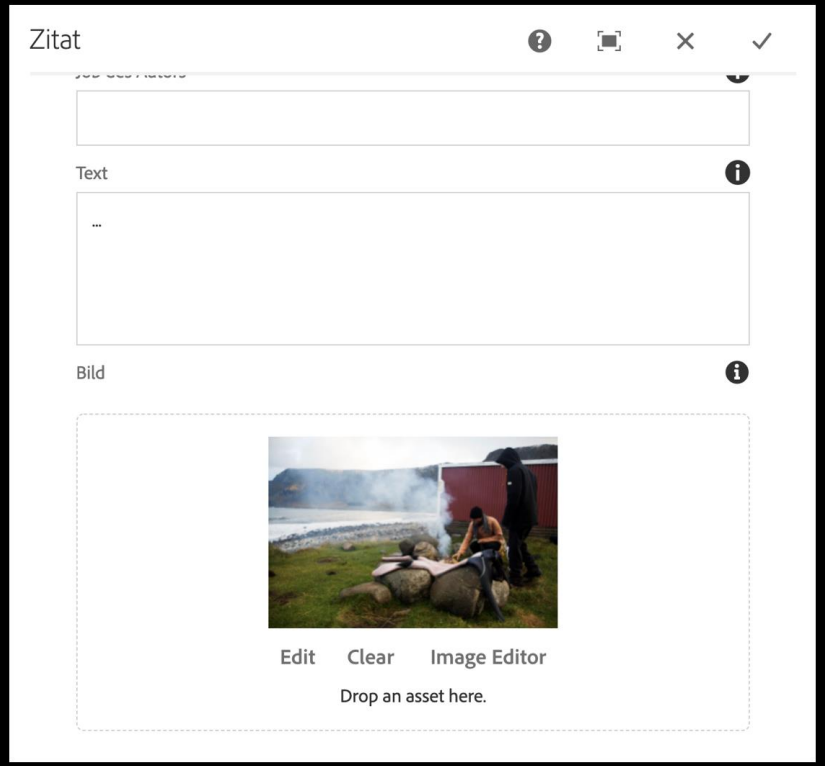
Text	Zentriert	1	Icon-Überschrift	Zentriert	1	Icon-Überschrift	Zentriert	1	Icon-Überschrift	Zentriert	1
Tooltip Text			Tooltip Text			Tooltip Text			Tooltip Text		
Überschrift			DentalImplantIcon			DentalImplantIcon			DentalImplantIcon		
Allgemein			Paket - Basis			Paket - Komfort			Paket - Super		

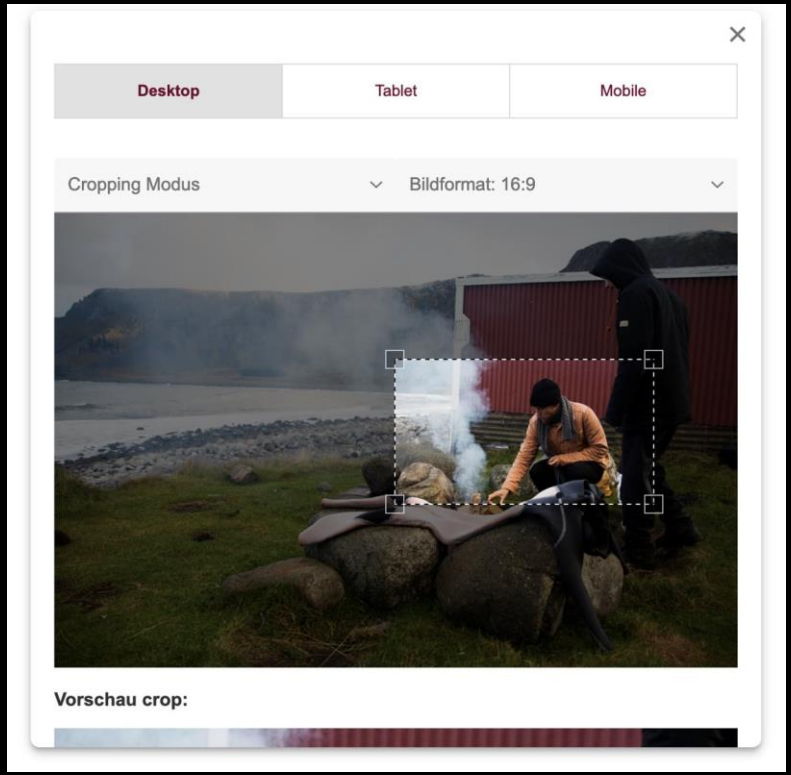
Text	Zentriert	1	Icon-Überschrift	Zentriert	1	Icon-Überschrift	Zentriert	1	Reihe: Icon-Überschrift	Zentriert	1
Headline Tooltip: Toller Content			Tooltip Text			Tooltip Text			Tooltip Text		
Überschrift			DashboardCheckIcon			DashboardCheckIcon			DashboardCheckIcon		
Aufnahme ohne Gesundheitsfragen			Überschrift			Überschrift			Überschrift		

Text	Zentriert	1	Text	Zentriert	1	Text	Zentriert	1	Reihe: Text	Zentriert	1
Headline Tooltip: Toller Content			Tooltip Text			Tooltip Text			Tooltip Text		
Überschrift			100%			100%			100%		
Regelversorgung bei Zahnersatz			Erstattung			Text			Erstattung		

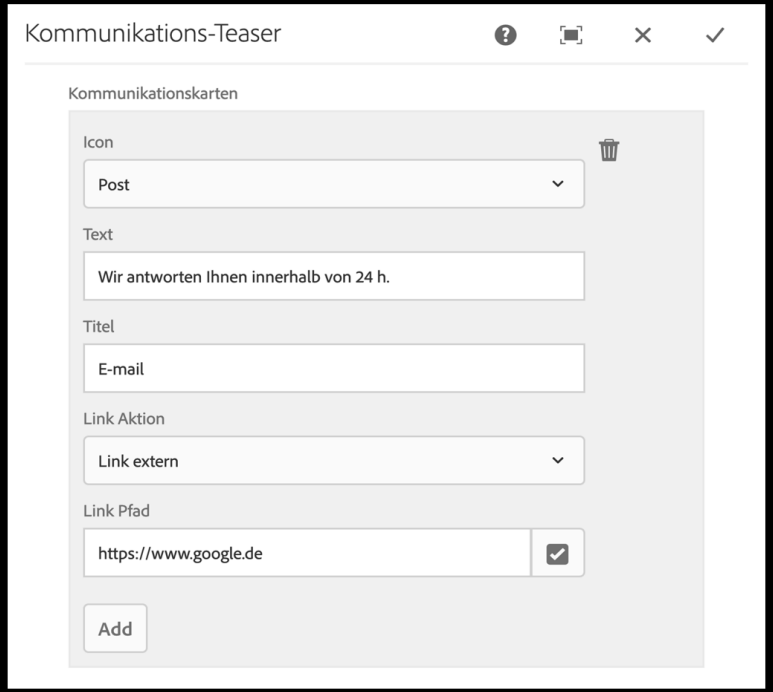
									Reihe:		

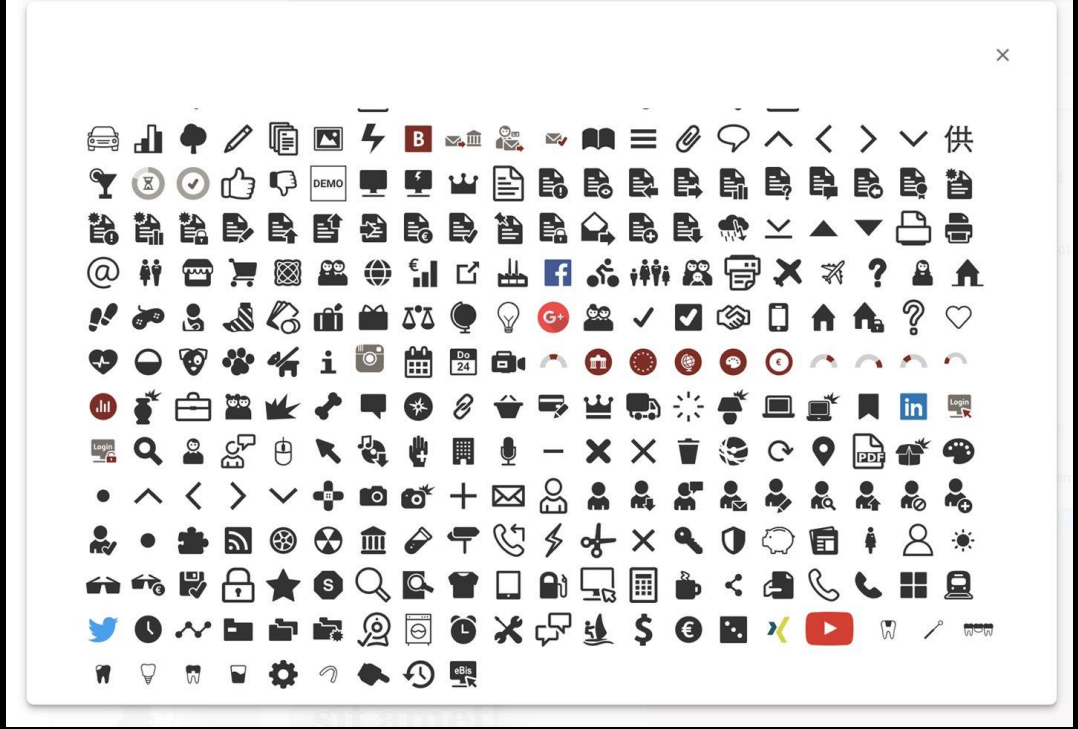
Custom Dialogs





Custom Dialogs





What we learned

What we learned

- Pro:
  - happy developers => increased productivity
  - faster onboarding of new teams
  - custom dialog > complex parsys structure
  - SEO and React is possible
  - components can be used outside of AEM
- Cons:
  - multiple AEM bundles can be a pain
  - check your NPM imports and bundle sizes

Future

- Parsys inside of React Apps
- Nested React apps
- Components as a serivce

  (deploy new componenets without any AEM deployment)

Questions / Contact

Tony Schumacher
Platform Expert, Partner
Mail: tony.schumacher@teclead.de
Telefon: +49 176 78 35 16 77