



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
2 - 4 SEPTEMBER 2019

# A non-SPA approach for AEM with REACT

L. Ellmer, E.ON Digital Technology

IT Professional for 24 years now– working with AEM for 4 years now  
42 years old

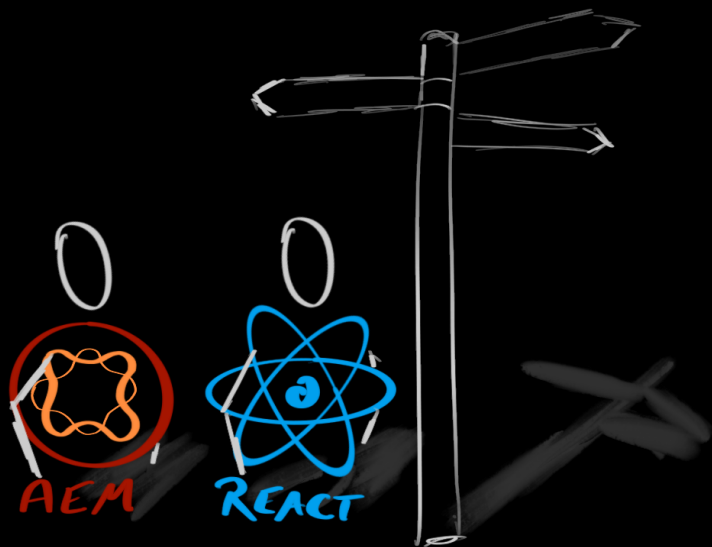
Love to cook & to craft with paper and wood

Interested in Homeautomation, SmartHome  
Technologies and Psychology

Father of two boys and one girl



# Agenda



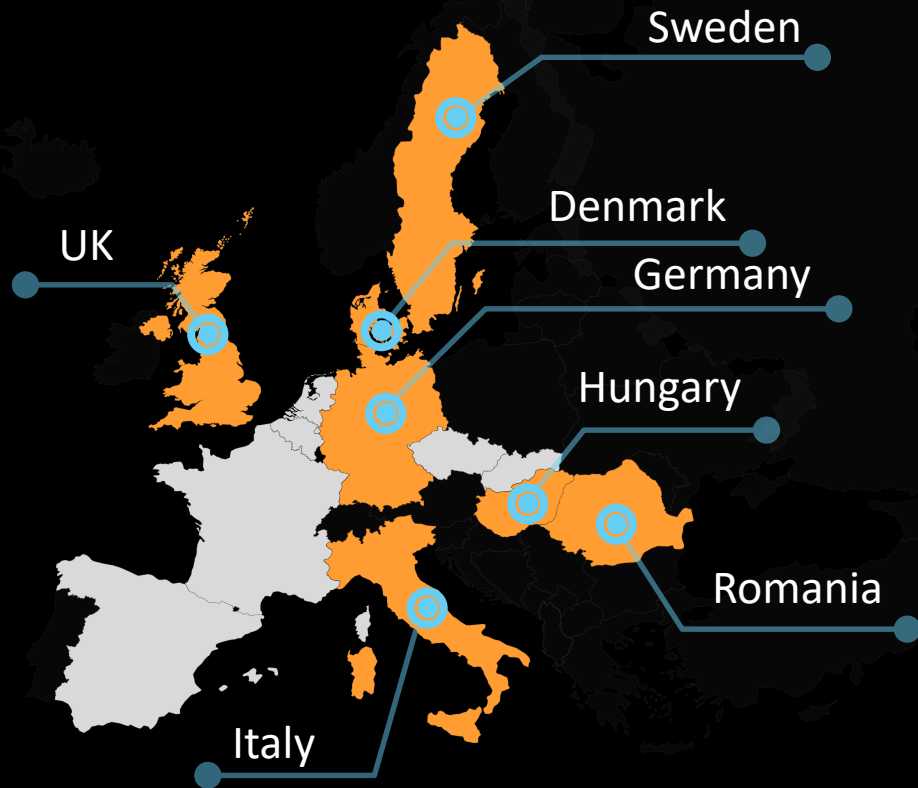
- AEM inside of E.ON
- Problem(s) to solve
- Opportunities
- Solution
- Demo
- Way to go

# Adobe AEM @E.ON

Country-  
Websites  
&  
Projects

45 AEM Apps  
online at the  
moment.

Everything on  
AEM now.



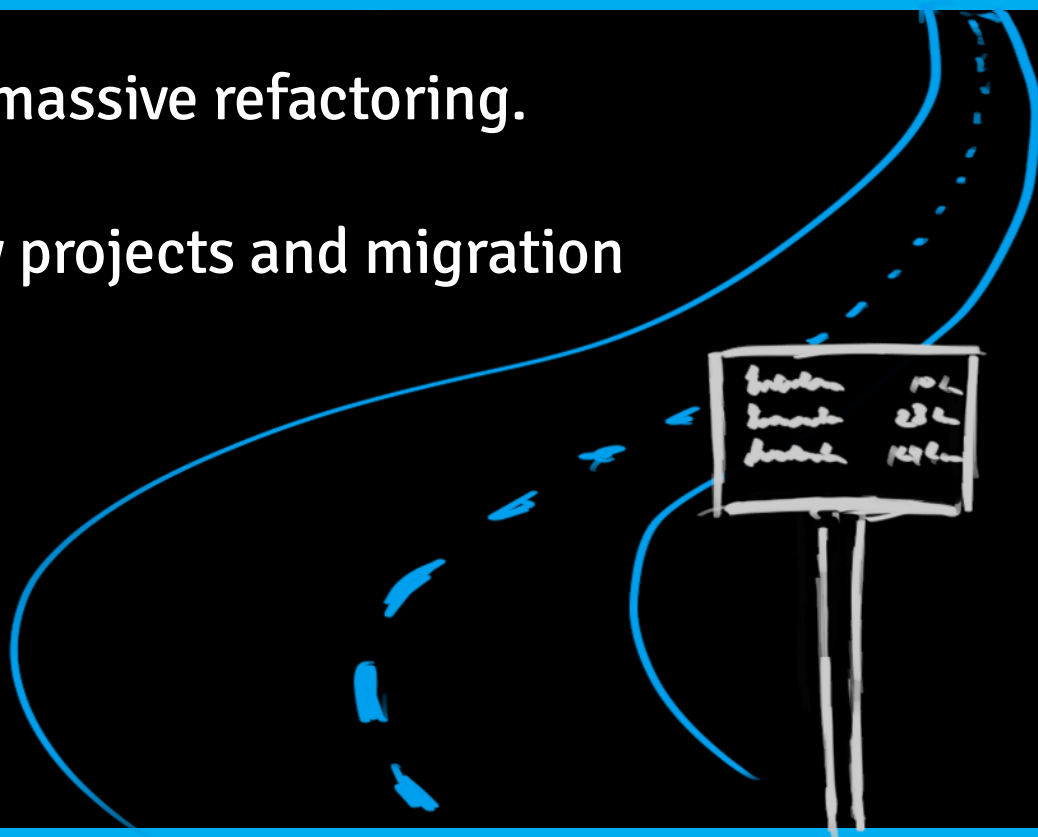
# AEM @ E.ON - Infrastructure

- 2x2 Publishers in Azzure Cloud Server Centers.
- AEM 6.4 (just migrated from 6.2)
- Deployment Process outsourced to Accenture India

- ⌘ Basic components set for smaller projects.
- ⌘ Country web-sites are largest customers
  - ⌘ Static representational pages
  - ⌘ Sales funnels
  - ⌘ Dynamic customer portal
- Customer portal based on React
- Data-loading by React only – all pages are static

# AEM @ E.ON - Roadmap

- ⌘ TouchUI-Upgrade with massive refactoring.
- ⌘ Template Editor for new projects and migration path for old once
- Constant knowledge sharing

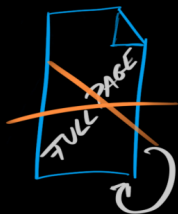




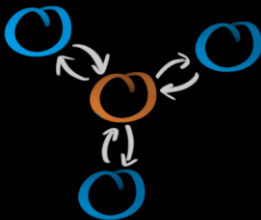
# Problem(s) to solve

# Problems to solve – React Issues

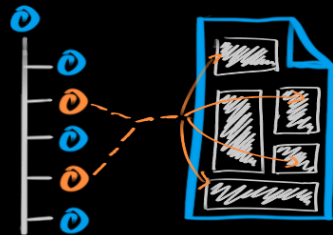
React was designed for Applications rather than modular systems with high autonomy



Configuration  
changes without  
*full-page reload*



Handle data for  
multiple dynamic  
modules

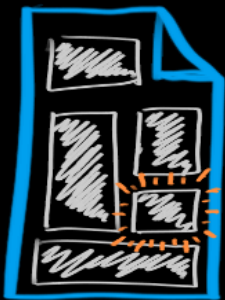


Node-Property  
Strategy

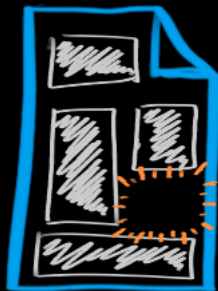
## DEMO of final "look & feel"

# Problems to solve – React Issues

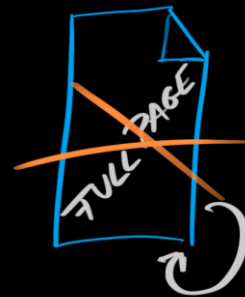
Bootstrapping just happens once



Dealing with  
newly inserted  
components



Dealing with the  
deletion of  
components



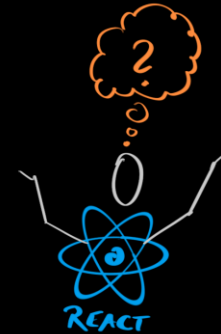
*Config-Changes  
shall update  
single component  
only*

# Problems to solve – React Issues

Configuration-Data is part of the React-Application  
not being injected



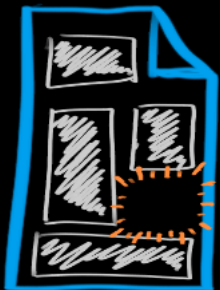
React needs to  
be made aware  
of changes



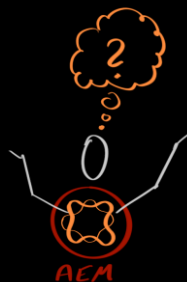
React is not  
directly able to  
modify the  
configuration

# Problems to solve – AEM Issues

SPA approach has no component based refreshing mechanism



Deletion of  
components by  
AEM



AEM does not  
“understand”  
the dynamic  
nature of  
REACT content



*Many modes and  
mode changes in  
AEM may  
interfere with this  
dynamic nature*

# Opportunities

# Opportunities – What we gain doing it

Possibility of smooth migrations for other or old web-application projects

Migration path for ...

... .NET server  
rendered applications

... *other JS-Framework*  
*Applications or even*  
*React Apps*



# Opportunities – What we gain doing it

Centralized components for data- & security handling

With best practices for implementation for all projects

# Opportunities – What we gain doing it

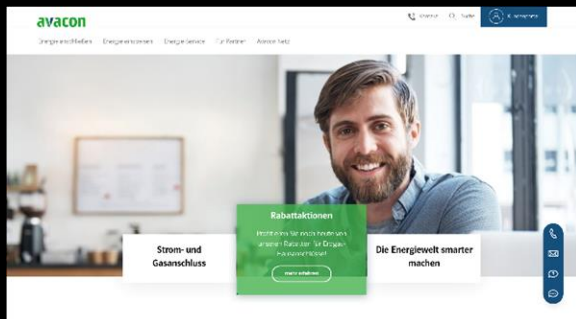
## Implementation of a living style-guide

White / Bordeaux (Disabled)

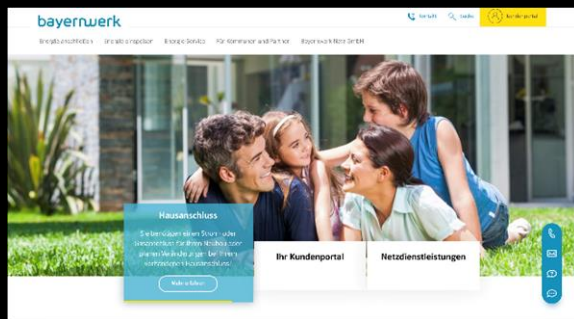
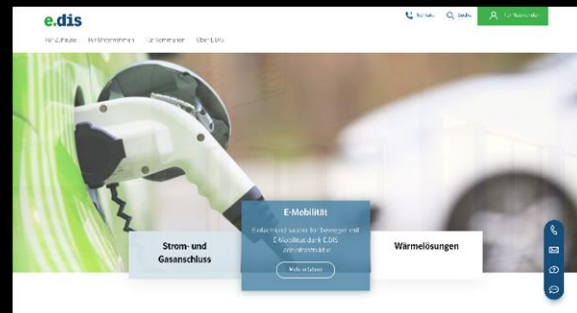
With global support of expensive features  
as accessibility and CI/CD conformity

# Opportunities – What we gain doing it

## Set of central components stylable by policies



Support for editable templates  
to ease the process  
of adapting a template to a  
sub-brand



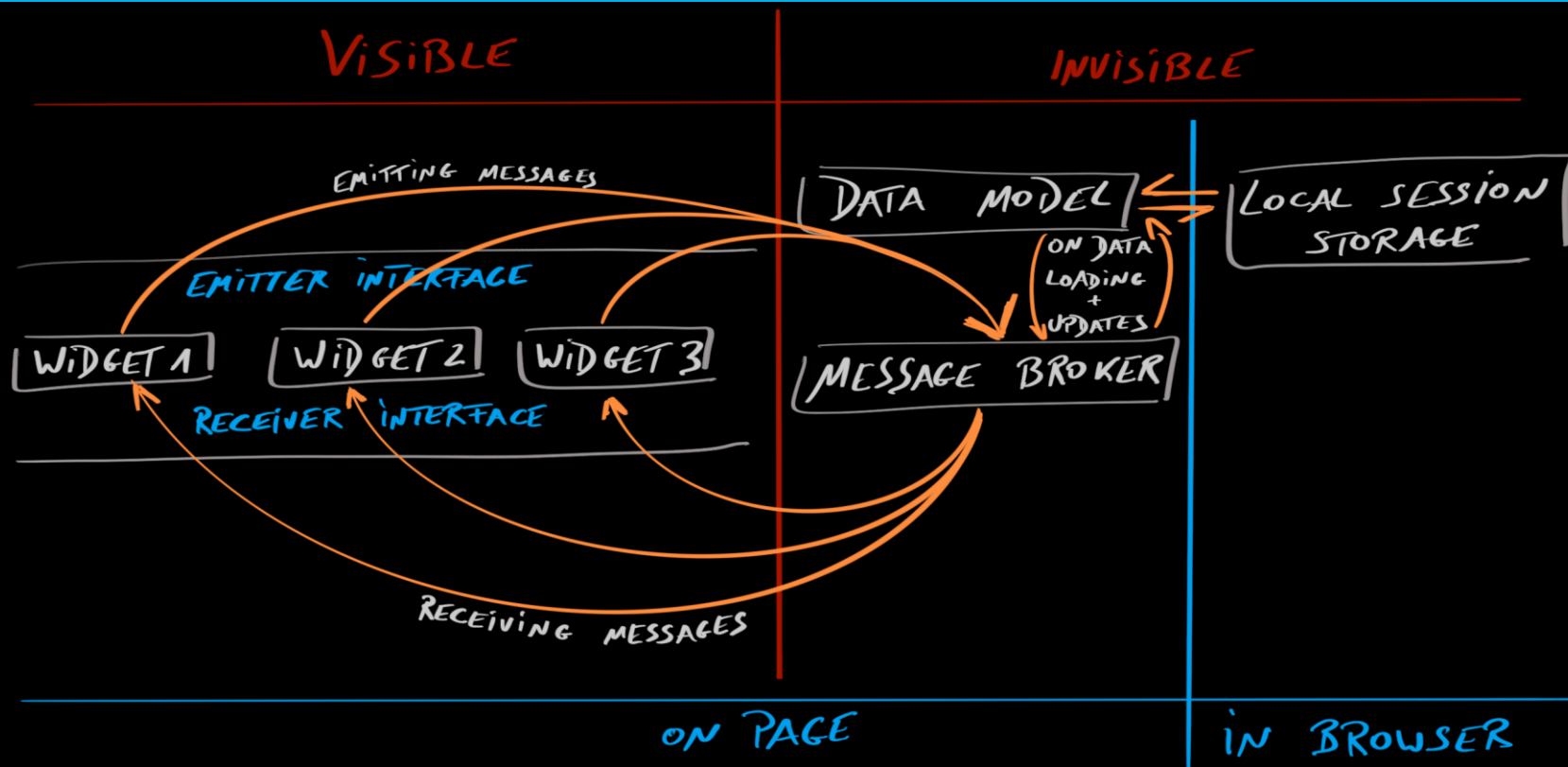
# Opportunities – What we gain doing it

Automated, front-end centric  
project-setup & work-process

Overcoming the shortage of AEM-Developers by  
enabling FE-Developers to write AEM components

# Solution

# Solution – Most Important



# Solution – Most Important

## Adjustment of Bootstrapping

Allowing multiple, dynamically driven bootstrapping

# Dive Into: React Habitat

- Tiny code footprint (4.4kb)
- Redux support
- Pass properties to components and back
- <https://github.com/DeloitteDigitalAPAC/react-habitat>





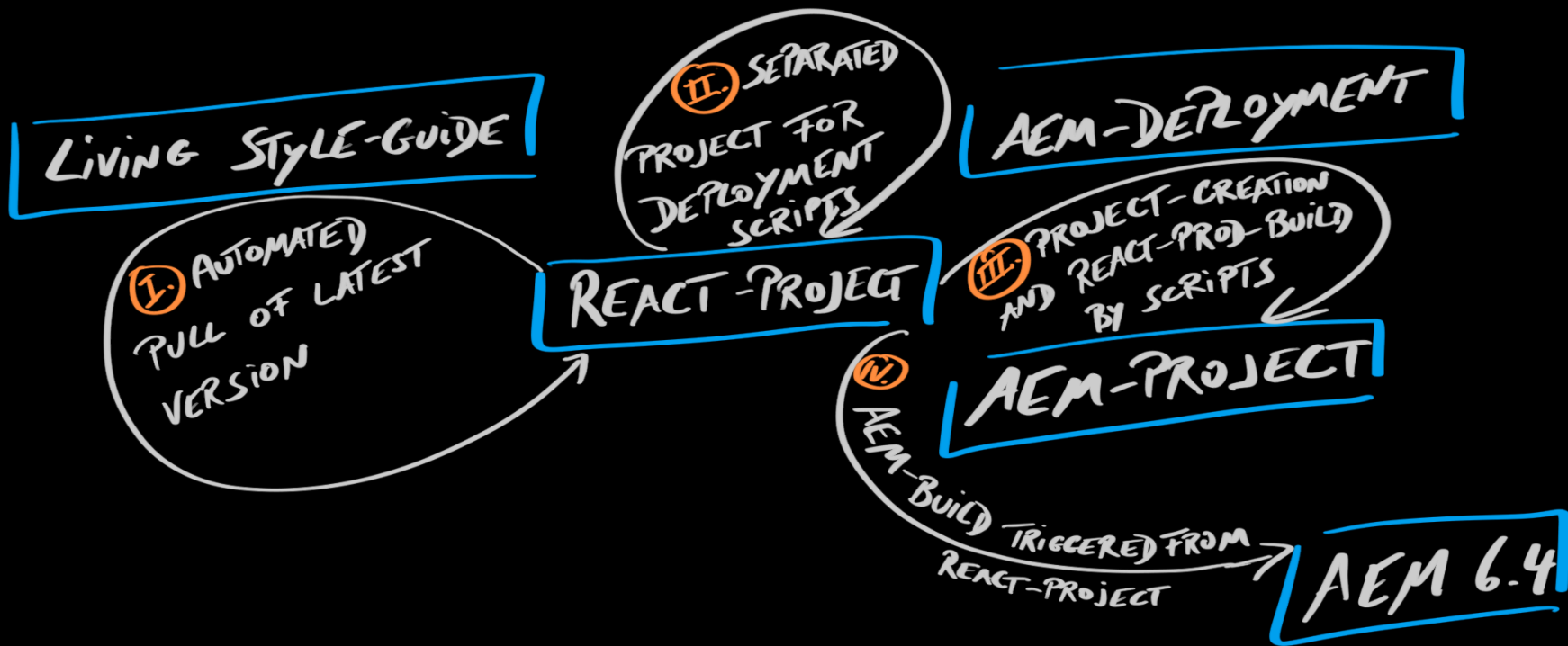
# Solution – Setting a focus

## Focus on FE-Development

Automation of anything else as much as possible

## DIVE INTO: build-chain

# Dive Into: build-chain



# Solution – Building the Base

## Create React Project

```
npx create-react-app aemreact
```

## Add React-Habitat to it

```
npm install react-habitat
```

## Link AEM-Deployment Project

```
npm link aem-deployment
```

## Start Coding

# Solution – What is aem-deployment

```
exports.deploy = function (verbose, deployEditorHelper, src, srcDistDir, targetDir, aemProjectName, aemClientLibName) {
  const shellJS = require('shelljs');
  const fs = require('fs-extra');
  const fileHelper = require('file');
  const os = require('os');
  src = !src.endsWith('/') ? src + '/' : src;
  srcDistDir = !srcDistDir.endsWith('/') ? srcDistDir + '/' : srcDistDir;
  targetDir = !targetDir.endsWith('/') ? targetDir + '/' : targetDir;

  const clearAemDistFolder = () => {
    if (verbose) console.log('clearAemDistFolder');
    fs.removeSync(targetDir + 'bundles/core/src/main/webapp/clientlibs-root/' + aemClientLibName);
    fs.removeSync(targetDir + 'bundles/core/src/main/webapp/clientlibs-root/' + aemClientLibName + '.json');
  };

  const deleteUnusedFiles = () => {...};

  const absolutePathToRelativePath = (path, srcDir, fileName) => {
    const isRelativeSrc = srcDir.startsWith('./');
    if (isRelativeSrc) {
      const correctedPath = srcDir.substring(2);
      return './' + path.substring(correctedPath.length) + '/' + fileName;
    }
    return './' + path.substring(srcDistDir.length) + '/' + fileName;
  };

  const generateImportFileTxt = (suffix) => {...};

  const generateClientLibJson = () => {...};

  const copyLocalDistToAemDist = () => {...};

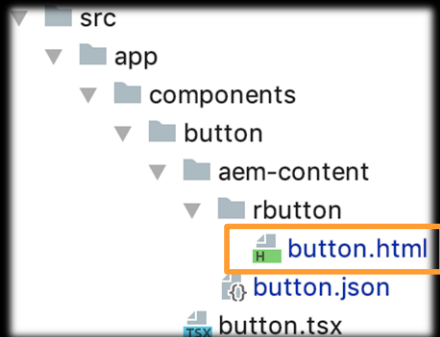
  const callAemDeployScript = () => {...};

  const modifyAemClientLibJson = () => {...};

  const copyAemSources = () => {...};

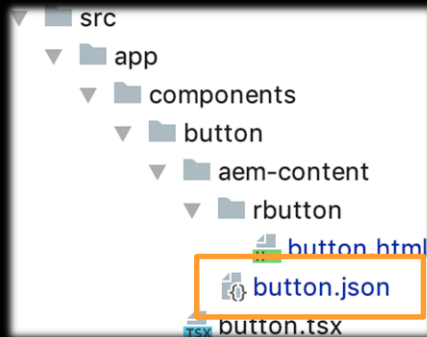
  const copyEditorHelperClientLib = () => {...};
}
```

# Solution – Coding: AEM Component HTL



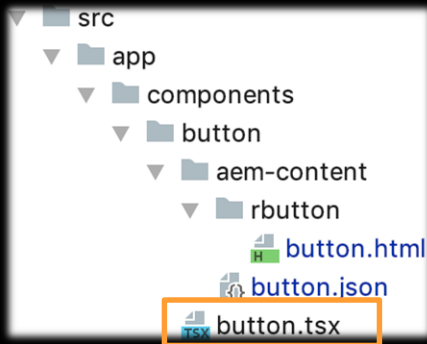
```
<div data-component="Button"
    data-prop-buttontext="${properties.buttonText}"
    data-prop-eventname="${properties.eventName}"
    data-prop-elementidentifier="${properties.elementIdentifier}"
    data-prop-optionalcontent="${properties.optionalContent}"
>Button: Please wait while being rendered...</div>
```

# Solution – Coding: AEM Component JSON



```
{
  "jcr:primaryType": "cq:Component",
  "jcr:title": "RButton",
  "componentGroup": "aemangular",
  "cq:editConfig": {
    "jcr:primaryType": "cq:EditConfig",
    "cq:listeners": {
      "afteredit": "__AEM__refreshAjaxContentAfterEdit",
      "afterinsert": "__AEM__refreshAjaxContentAfterEditDelayed"
    }
  },
  "cq:dialog": {
    "jcr:primaryType": "nt:unstructured",
    "sling:resourceType": "cq/gui/components/authoring/dialog",
    "jcr:title": "Simple Dialog",
    "content": {
```

# Solution – Coding: React Component



```

import React from "react";
import {string} from "prop-types";

type Buttonprops = {
  buttontext: string,
  eventname: string,
  elementidentifier: string,
  optionalcontent:string
}

class Button extends React.Component<Buttonprops> {
  constructor (props: Buttonprops) {
    super(props as any);
  }
  handleClick() {
    document.dispatchEvent(new CustomEvent(this.props.eventname, {
      detail: {
        id: this.props.elementidentifier,
        buttonText: this.props.buttontext,
        optionalContent: this.props.optionalcontent,
      }
    }));
  }
  render() {
    return (
      <button type="button" className="c-button c-button--text" onClick={() => this.handleClick()}>
        <span className="c-button__text">
          <span className="c-button__text-content">{this.props.buttontext || 'missing button text config'}</span>
        </span>
      </button>
    );
  }
}

```



# Solution – Coding: Component Registration

```
class Aemreact extends ReactHabitat.Bootstrapper {  
  constructor(){  
    console.log('@constructor of aemreact');  
    super();  
  
    this.registerOnRefreshHook();  
    this.registerComponents();  
    this.update();  
  }  
  
  private registerOnRefreshHook(): void {  
    document.addEventListener( type: 'AEM_refreshbootstrap', listener: () => {  
      this.update();  
    });  
  }  
  
  private registerComponents(): void {  
    // Create a new container builder:  
    const builder = new ReactHabitat.ContainerBuilder();  
    // Register a component:  
    builder.register(Testcomponent).as( key: 'TestComponent');  
    builder.register(Button).as( key: 'Button');  
    // Finally, set the container:  
    this.setContainer(builder.build());  
  }  
}
```

# Solution – Governance

Take strategic parts into central governance

Projects are free to build independently on top of this foundation

# Solution – Message Broker

Central Whiteboard-Pattern  
for all inter-component communication

Loosely coupled but highly cohesive for REACT with AEM

# Solution – Message Broker

```
import {Subject, ReplaySubject, Observable} from 'rxjs';
import {first} from 'rxjs/operators';
import * as uuid from 'uuid';

export interface ITopicHandle<T> {...}

export interface IMessageBroker {
  registerTopic<T>(topic: string, multiple?: boolean): ITopicHandle<T>;
  unregisterTopic(topicId: string, ownerId: string): void;
  getAvailableTopicList(): string[];
  subscribeToTopic(topicId: string): Observable<any>;
}

export class MessageBroker implements IMessageBroker {
  private topicMap: Map<string, { subject: ReplaySubject<any>; ownerId: string; isMultiple: boolean; emitterCount: number; }>;
  private awaitingTopicMap: Map<string, Subject<Observable<any>>[]>;

  constructor() {
    this.topicMap = new Map();
    this.awaitingTopicMap = new Map();
    this.registerToWindow();
  }

  public registerTopic<T>(topicId: string, multiple = false): ITopicHandle<T> {...}

  public unregisterTopic(topicId: string, ownerId: string): void {...}

  public getAvailableTopicList(): string[] {...}

  public subscribeToTopic(topicId: string): Observable<Observable<any>> {...}

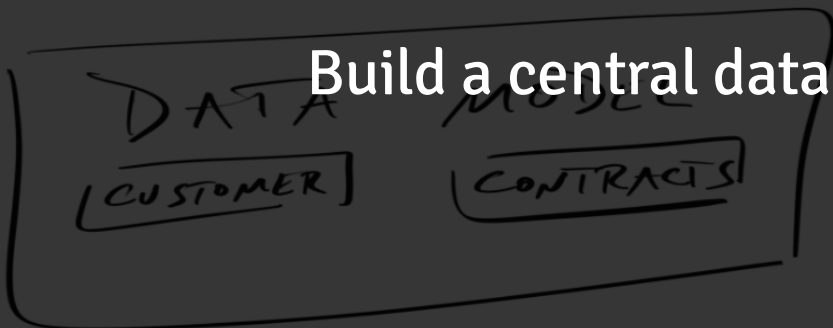
  private registerToWindow(): void {...}
}

new MessageBroker();
```

# MISSING: data-loading & security

# Solution – Governance - Data-Loading

Build a central data-model for FE development



LOCAL STORAGE  
LONG LASTING ...  
BUT  
NOT SAVE

Don't let your data-model just happen!  
Structure it with full consciousness

Manage data-requests from components  
Request data from server only if necessary

LOCAL SESSION CACHE  
SAVE  
LIMITATION TO SESSION IS  
IRRELEVANT

# Solution – Governance - Caching

Utilize Session Cache for all data

Reduce load on your Backend and speed up UX with cached data

## ROADMAP: What's up next ?



# Roadmap – What's up next ?

## Development of data-handling, caching & security

This is an essential part of this framework and we are working on it

# Roadmap – What's up next ?

## Documentation

Even if no one ever wants to do it, we need it desperately, for on-boarding and also to harden our concept by understanding it in full depth by describing it

# Roadmap – What's up next ?

## Versioning of components

So far there is always one version of a component, but we would need a way to manage several versions on one server.

# Roadmap – What's up next ?

## Migration of existing projects

Existing projects need to migrate to simplify development & unify the platform.

Thank you !!!

... there is just ...

”ONE MORE THING“

# One more thing – Angular

Also works with Angular

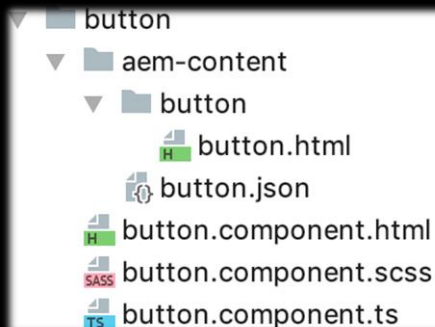
&

React & Angular together

Enabling you to work with the framework of choice this approach allows both major frameworks



# Dive Into: Bootstrapping w/ Angular



```
public ngDoBootstrap(appRef: ApplicationRef): void {
    rootComponentList.forEach( callbackfn: (componentDef: Type<{}>) => {
        const factory = this.resolver.resolveComponentFactory(componentDef);
        const elList = document.querySelectorAll(factory.selector);
        if (elList) {
            const count = elList.length;
            for (let i = 0; i < count; i += 1) {
                if (!elList[i].attributes.getNamedItem( qualifiedName: 'ng-version')) {
                    appRef.bootstrap(factory, elList[i]);
                }
            }
        }
    });
}
```

# One more thing – Open Source Release

We will release this framework as  
**open-source to the community**

Just data-handling and documentation need to be finalized  
Check:

*e-on* <https://github.com/eon-com/MAFFA>

for more information

# Questions ?



Lars Ellmer

Lars.Ellmer@eon.com