APACHE SLING & FRIENDS TECH MEETUP
2 - 4 SEPTEMBER 2019

Assets and Links in AEM Projects

Stefan Seifert, pro!vision GmbH

About the Speaker

- AEM Developer
- Apache Sling PMC
- Apache Member
- CTO of pro!vision GmbH

Stefan Seifert

PRO!VISION
SOFTWARE CRAFTSMANSHIP

https://www.pro-vision.de

adaptTo()

Technical Aspects in AEM Projects

adaptTo()

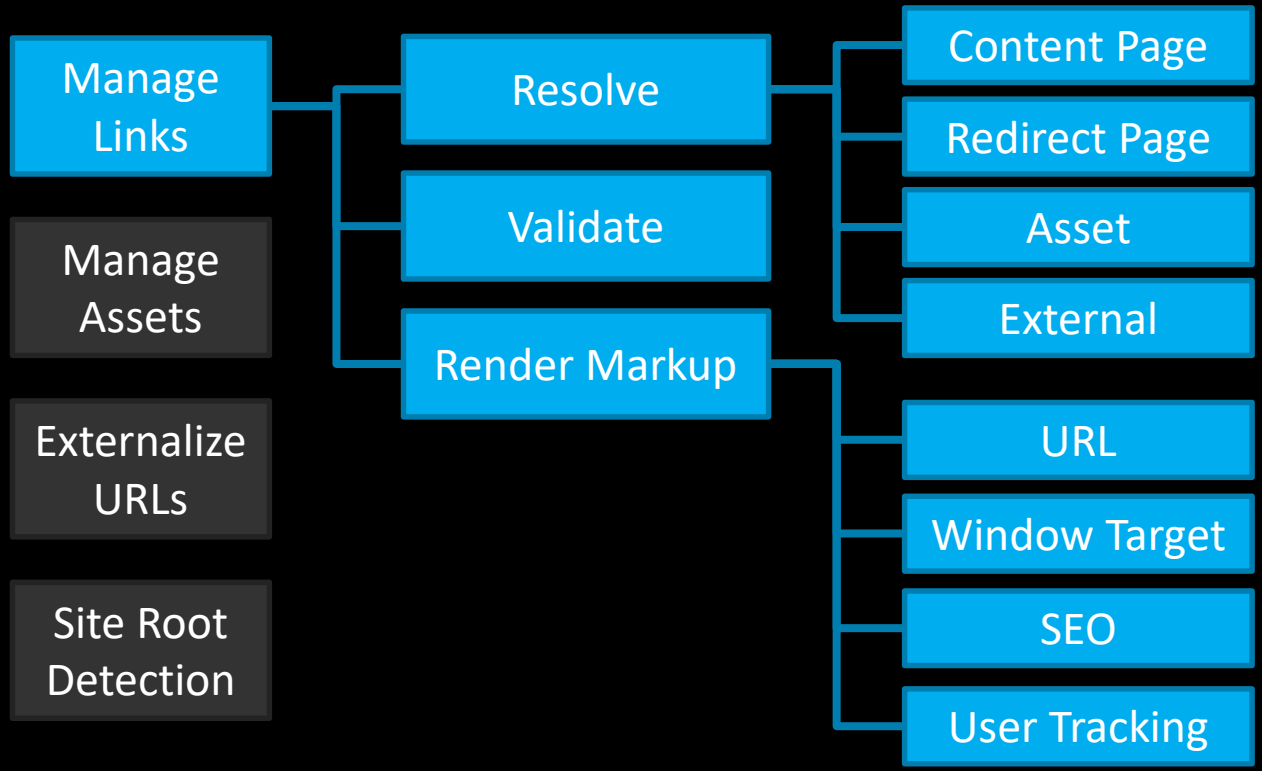
What are Technical Aspects

- Cross-cutting concerns
- Common behavior for all components
- May be depending on project or tenant

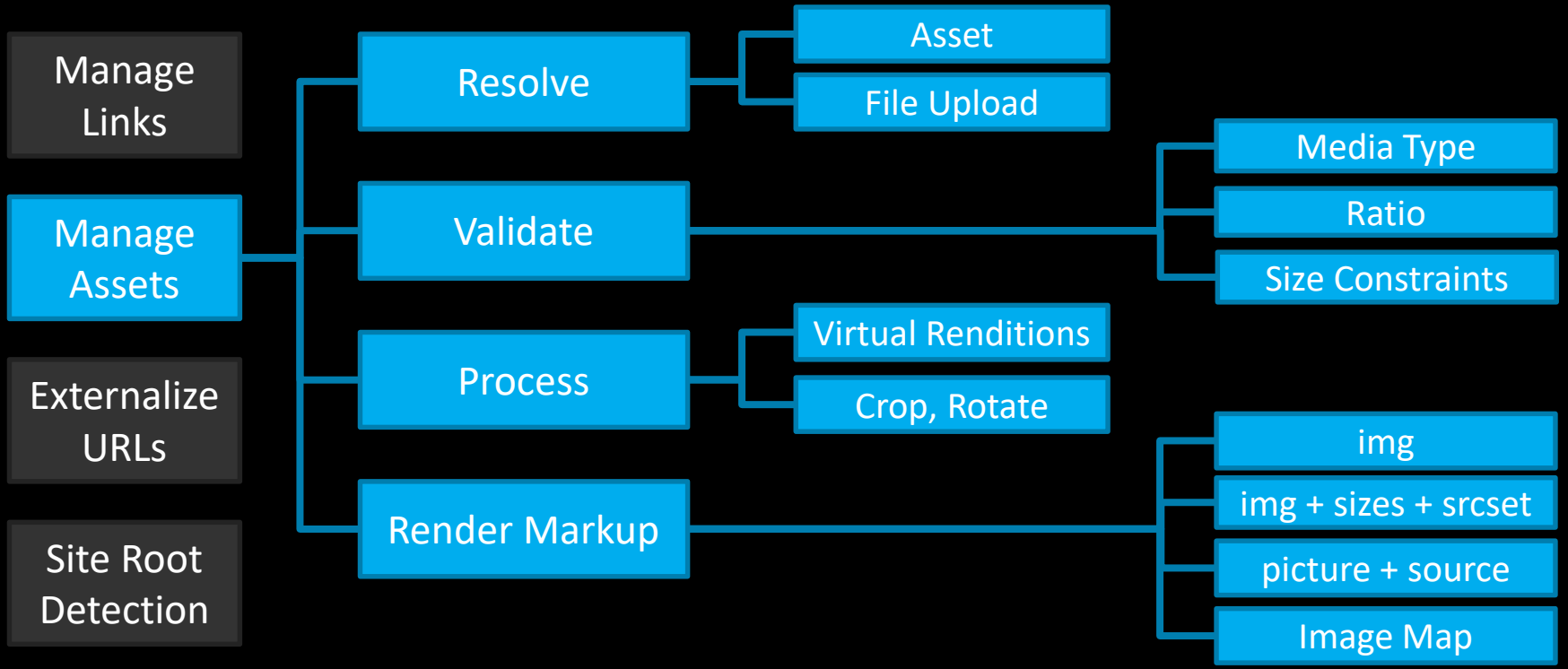
Technical Aspects

- Manage Links
- Manage Assets
- Externalize URLs
- Site Root Detection

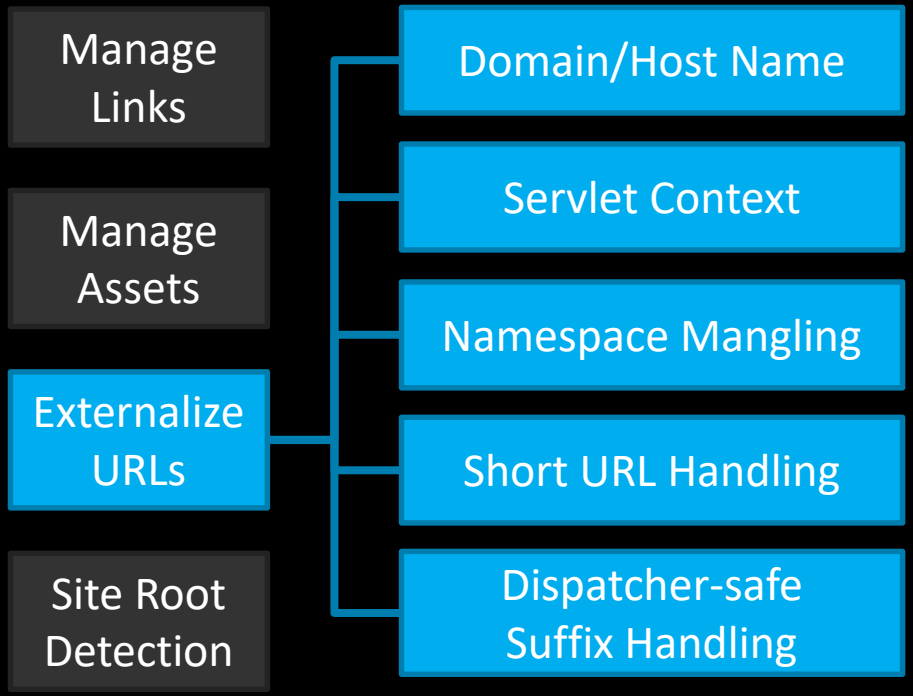
Technical Aspect: Links



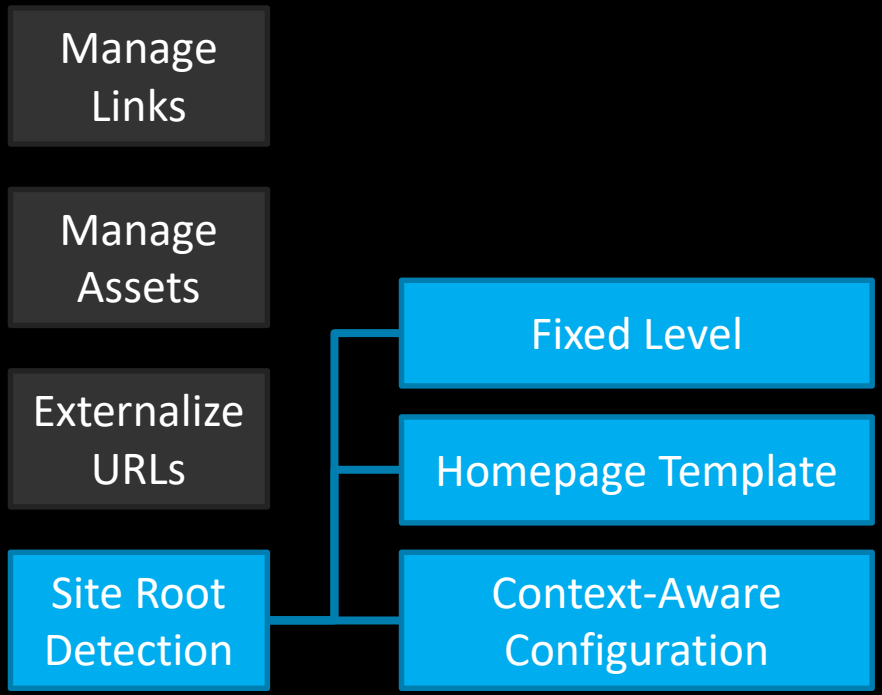
Technical Aspect: Assets / Images



Technical Aspect: URLs



Technical Aspect: Site Root

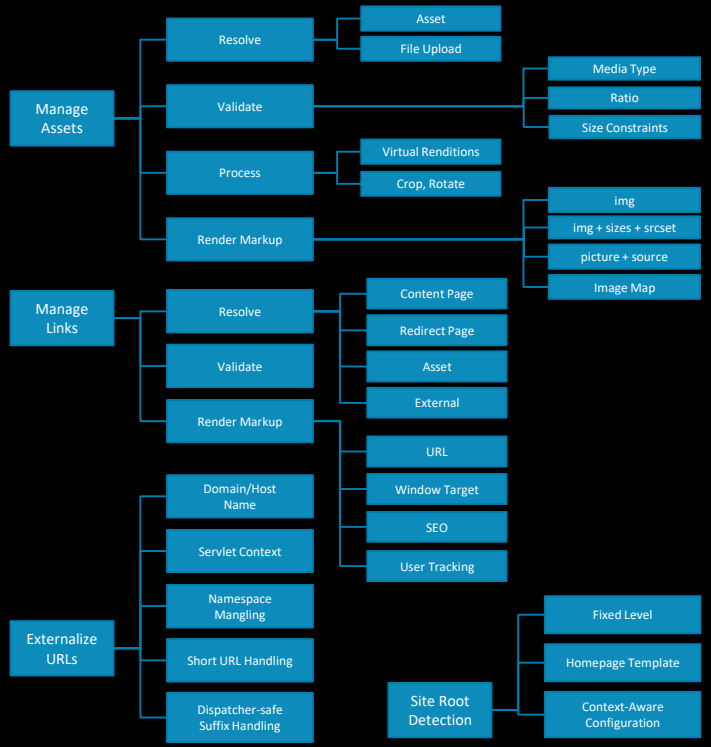


Use cases:

- Navigation Root
- Breadcrumb Root
- Root for Path Field
- …

Putting all together

- Reusable
- Multi tenancy
- Configurable
- Well tested

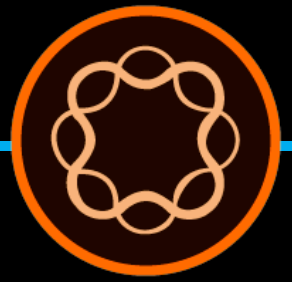


adaptTo()

AEM Sites Core Components

adaptTo()

About AEM Sites Core Components



- Core Components are great
- First release 2017
- New features every few months
- Open Source Community
- Component Library

DEMO

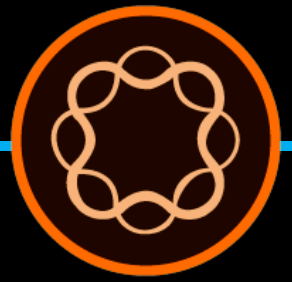
adaptTo()

Technical Aspect: Links

- Basic link handling present
- API a bit inconsistent *
- Only supports the URL
- Partial redirect page support
- Validation relies on Rewriter

* Consolidation work in progress

Technical Aspect: Assets / Images



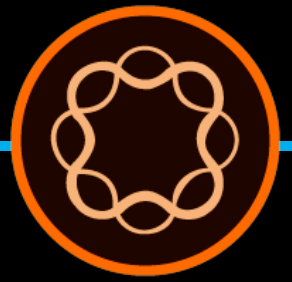
- Feature-rich Image & Download components
- Logic hidden in the components
- Proprietary responsive handling
  - Not using standard HTML5 markup

adaptTo()

Technical Aspect: URLs

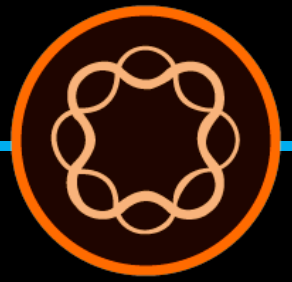
- Basic URL handling support
- Relies on CQ Link Externalizer Service
  - which has no proper multi-tenancy support

Technical Aspect: Site Root



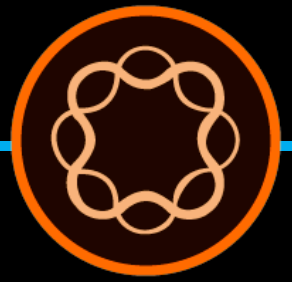
- Policy-based
- Different concepts (root level vs. root path)
- Special magic in Navigation component to detect root based on path in policy
  - e.g. blueprint/language master

Putting all together



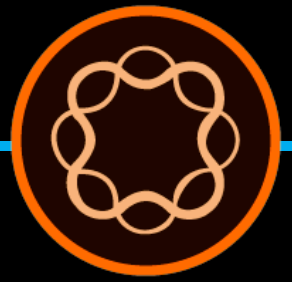
- Technical aspects in core components **in general not reusable** (internal, no API)
- Link, URL and Site Root support only rudimentary
- You end up writing your own glue code

Multi Tenancy Limited



- Mostly based on Content Policies
- Content Policies mapped via Templates
- Not possible to re-use Templates across tenants if different policies required

Reusing Image logic



- Reuse concept for image logic:
  Extend + Embed the Image Component
  - Example: Teaser component
- Awkward pattern, lots of limitations
  - Not possible to define image settings in teaser policy
  - Teaser inherits policy from image component – if you want it or not
  - Not possible to display more then one image
  - Not possible to influence the image markup

adaptTo()

wcm.io Handler

wcm.io & wcm.io Handler

- First release 2014
  - Based on 10 years of AEM project experience
- Modular design
- Multi tenancy
- “Technical aspects” managed with wcm.io Handler

Technical Aspect: Links

- Link Handler distinguishes Link Types
  - Internal, External, Asset, “Cross-Site”
  - with their own resolution & validation strategies
- Granite UI components
- RTE Link Plugin

DEMO

Technical Aspect: Links

SLING MODEL

```
@Self
private LinkHandler linkHandler;

@PostConstruct
private void activate() {
  // build link stored in current resource
  Link link = linkHandler.get(resource).build();

  // check if link is valid
  if (link.isValid()) {
    // ...
  }
}
```

HTL

```
<sly data-sly-use.link="io.wcm.handler.link.ui.ResourceLink"/>
<a data-sly-attribute="${link.attributes}" data-sly-test="${link.valid}">
  ${properties.linkTitle}
</a>
```

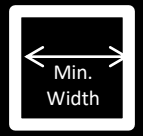
adaptTo()

Technical Aspect: Assets / Images

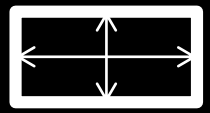
- Media Handler basically has same features as the Image Core Component
- Media Format Concept & Auto Cropping
- Responsive using HTML5 Standards
- Dispatcher-friendly virtual renditions
- Granite UI Components

Examples for Media Formats

16:9 Landscape



1:1 Square, min. 500px width



Exactly 1000 x 500 pixel

Download asset (PDF)

DEMO

Technical Aspect: Assets / Images

SLING MODEL

```
@Self
private MediaHandler mediaHandler;

@PostConstruct
private void activate() {
  // build media referenced in current resource
  Media media = mediaHandler.get(resource)
      .mediaFormat(MediaFormats.LANDSCAPE)
      .build();
  // check if media is valid
  if (media.isValid()) {
    // ...
  }
}
```

HTL

```
<sly data-sly-use.media="${'io.wcm.handler.media.ui.ResourceMedia'
    @ mediaFormat='landscape'}" data-sly-test="${media.valid}">
  ${media.markup @ context='unsafe'}
</sly>
```

Technical Aspect: URLs

- URL Handler externalizes Links based on Context-Aware configuration
- Integrated with Rewriter

```
String url = urlHandler.get("/resource/path")
    .selectors("myselector")
    .extension("html")
    .suffix("mysuffix.html")
    .urlMode(UrlModes.FULL_URL_FORCESECURE)
    .buildExternalLinkUrl();
```

DEMO

Technical Aspect: Site Root

- Defaults to “inner-most” Context-Aware Configuration Root
- Customizable via OSGi service

SLING MODEL

```
@Self
private SiteRoot siteRoot;

@PostConstruct
private void activate() {
  // get site root page
  Page rootPage = siteRoot.getRootPage();
}
```

Other benefits

- Fluent APIs
- Generic Sling Models + Dialog Fragments
- Hooks for Pre- and Post Processing
- Context-Aware Services for multi-tenant aware customization

What now?

WCM Core Component

- Introducing
  wcm.io WCM Core Components
- A thin layer above AEM Sites Core Components to enrich them with wcm.io Handler Functionality
- Drop-In replacement

Component Overview

Template components
- Page
- Navigation LINK URL
- Language Navigation LINK
- Breadcrumb LINK URL
- Quick Search

Page authoring components
- Text RICHTEXT LINK
- Title LINK
- Image MEDIA LINK
- Download MEDIA
- Teaser MEDIA LINK RICHTEXT
- Button LINK
- List LINK
- Content Fragment
- Content Fragment List
- Separator

Container components
- Container MEDIA
- Carousel
- Accordion
- Tabs

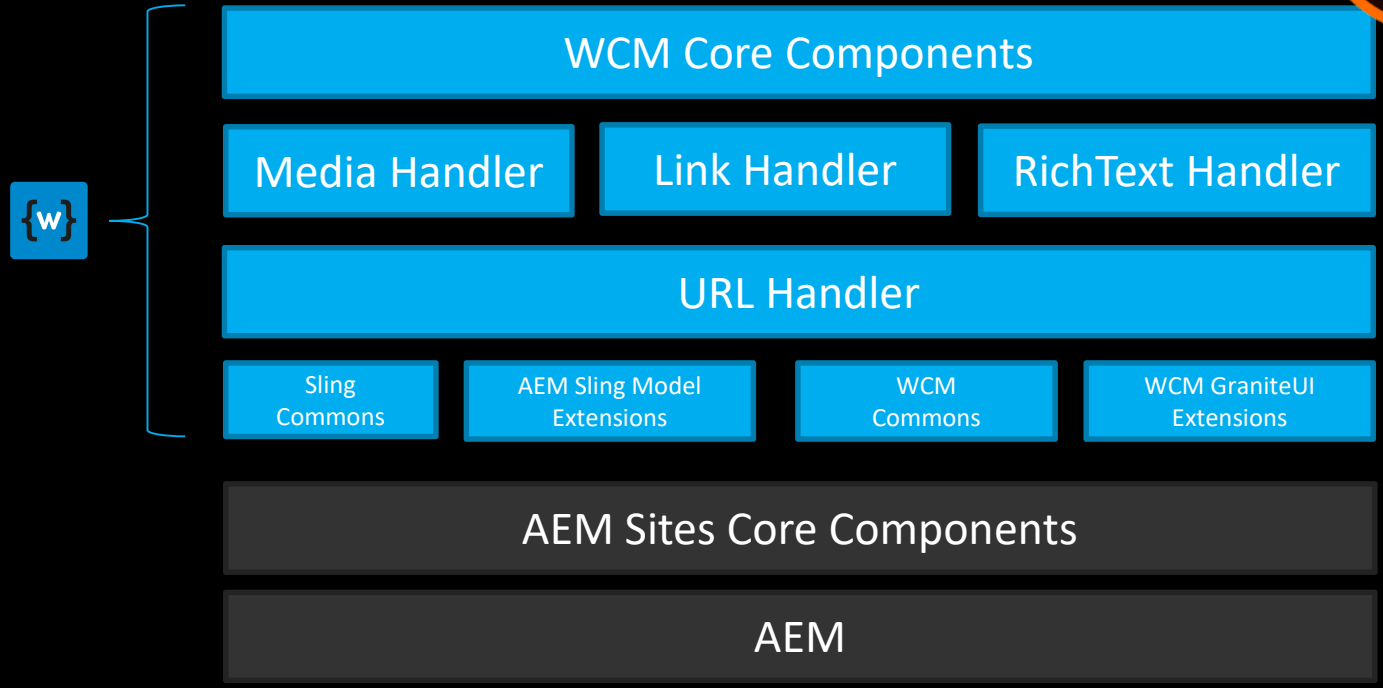
Form components
- Form container
- Form text field
- Form options field
- Form hidden field
- Form button

New components
- wcm.io Responsive Image MEDIA LINK

New: Variant of Core Image Component that uses HTML5 standard markup

Unsupported components
- Sharing

Module dependencies



DEMO

Maven Archetype

wcm.io Maven Archetype for AEM

https://wcm.io/tooling/maven/archetypes/aem/

- Setup new AEM Projects with various layouts
- Support for wcm.io Handler
  & wcm.io WCM Core Components

Outlook

Future plans

- Quickly add support for new features in wcm.io WCM Core Components
- Reduce code duplication by improving AEM Sites Core Components
- Rule of thumb: Do not add new features, improve AEM Sites Core Components instead

adaptTo()

References

- Demo Project
  https://github.com/adaptto/2019-aem-assets-links
- AEM Sites Core Components
  https://github.com/adobe/aem-core-wcm-components
- wcm.io WCM Core Components
  https://wcm.io/wcm/core-components/
  https://wcm-io.atlassian.net/wiki/x/AYCJS
- wcm.io Handler
  https://wcm.io/handler/
- wcm.io Maven Archetype for AEM
  https://wcm.io/tooling/maven/archetypes/aem/

Bootstrap Template Used for the Demo: https://github.com/BlackrockDigital/startbootstrap-agency