



# What's new in AEM Mocks

Stefan Seifert, pro!vision GmbH

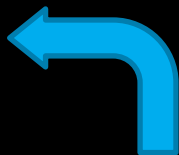
# Testing with Content Policies

- Unit testing your component with different content policies

SLING MODEL

*@AemObject*

**private** Style currentStyle;



UNIT TEST

```
// create a content policy with mapping for resource type
context.contentPolicyMapping("app1/componenty/component1",
    "prop1", "value1",
    "prop2", 123);
```

New in AEM Mocks 2.7.0

## ■ Complex policies possible

UNIT TEST

```
context.contentPolicyMapping("app1/componenty/component1",  
    NN_COMPONENT_MEDIA_RESPONSIVE_PICTURE_SOURCES, ImmutableMap.of(  
        "source1", ImmutableMap.of(  
            PN_PICTURE_SOURCES_MEDIAFORMAT, "home_stage",  
            PN_PICTURE_SOURCES_MEDIA, "media1",  
            PN_PICTURE_SOURCES_SIZES, "sizes1",  
            PN_PICTURE_SOURCES_WIDTHS, "200,400?"),  
        "source2", ImmutableMap.of(  
            PN_PICTURE_SOURCES_MEDIAFORMAT, "home_teaser",  
            PN_PICTURE_SOURCES_WIDTHS, "200,300"))));
```

New in AEM Mocks 2.7.0

# Testing with Content Fragments

## ■ Content fragments in unit tests on-the-fly

UNIT TEST

```
// create a structured content fragment
ContentFragment cf1 = context.create().contentFragmentStructured("/content/dam/cf1",
    "param1", "value1",
    "param2", 123,
    "param3", true,
    "param4", new String[] { "v1", "v2" });

// create an unstructured content fragment with some HTML markup
ContentFragment cf2 = context.create().contentFragmentText("/content/dam/cf2",
    "<p>Text</p>", "text/html");
```

New in AEM Mocks 2.5.0

- New support for TIFF and SVG media types
  - And mocks for AssetStore and AssetHandler

```
<!-- TIFF support -->
<dependency>
  <groupId>com.twelvemonkeys.imageio</groupId>
  <artifactId>imageio-tiff</artifactId>
</dependency>
<!-- SVG support -->
<dependency>
  <groupId>com.twelvemonkeys.imageio</groupId>
  <artifactId>imageio-batik</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-transcoder</artifactId>
</dependency>
```

<https://wcm.io/testing/aem-mock/usage-imageio.html>

New in AEM Mocks 2.7.0

# AEM Mocks References

- Documentation & Changelogs

<https://wcm.io/testing/aem-mock/index.html>

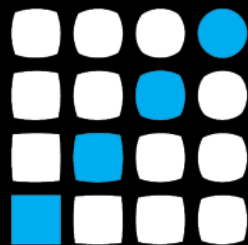
- Ask the AEM Community Expert Session March 2019:  
Unit Testing with AEM Mocks (Video + Demo Project)

<https://github.com/stefanseifert/2019-atace-unit-testing-with-aem-mocks>

- AEM Mocks included in Maven Archetypes

- [Adobe AEM project archetype](#) (since Version 18)
- [wcm.io Maven Archetype for AEM](#)





**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
2 - 4 SEPTEMBER 2019

**wcm.io Clientlibs UI Extensions**

Stefan Seifert, pro!vision GmbH

## ■ Include Clientlib in AEM

HTL

```
<sly data-sly-use.clientLib="/libs/granite/sightly/templates/clientlib.html" data-sly-call="$ {clientlib.js @ categories=['my-category']}'" />
```

## ■ Same using wcm.io

HTL

```
<sly data-sly-use.clientLib="/apps/wcm-io/wcm/ui/clientlibs/sightly/templates/clientlib.html" data-sly-call="$ {clientlib.js @ categories=['my-category']}'" />
```



## ■ Async Script Loading

HTL

```
<sly data-sly-use:clientLib="/apps/wcm-io/wcm/ui/clientlibs/sightly/templates/clientlib.html" data-sly-call="${clientLib.js @ categories=['my-category'], async=true}" />
```

HTML OUTPUT

```
<script src="mycategory.js" async></script>
```

# Include AEM Client Libraries in HTL

## ■ nomodule + defer

HTL

```
<sly  
data-sly-use.clientLib="/apps/wcm-io/wcm/ui/clientlibs/sightly/templates/clientlib.html"  
data-sly-call="$ {clientlib.js @ categories=['my-category'], nomodule=true, defer=true}" />
```

HTML OUTPUT

```
<script src="mycategory.js" nomodule defer></script>
```

## ■ Documentation & Changelogs

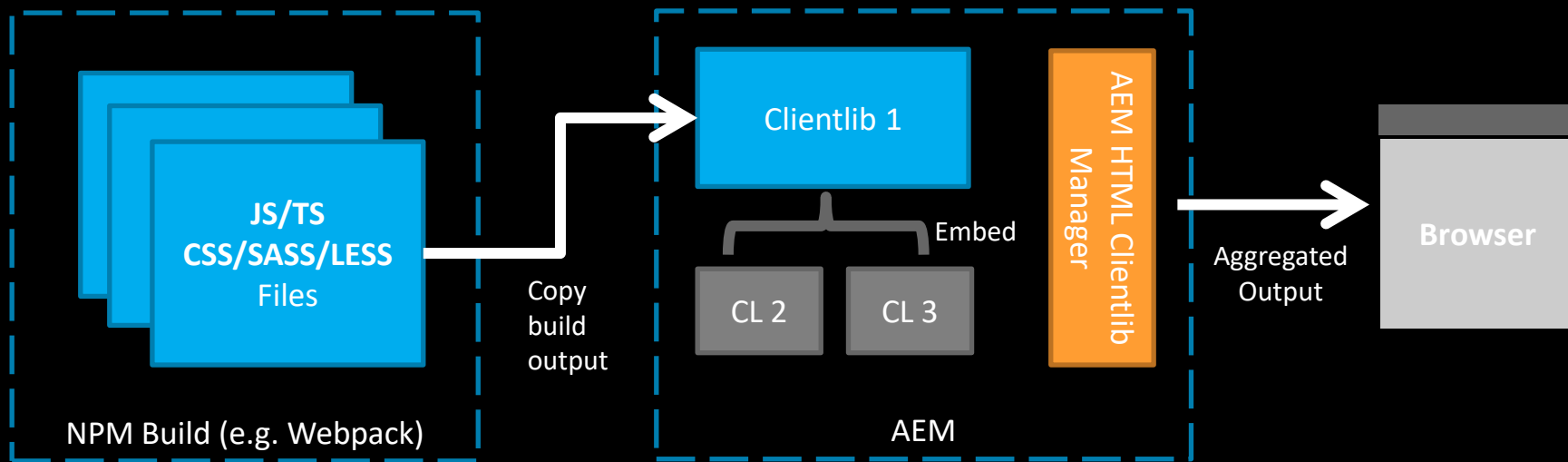
<https://wcm.io/wcm/ui/clientlibs/>

- `async` = `true` | `false`
- `crossorigin` = `anonymous` | `use-credentials`
- `defer` = `true` | `false`
- `integrity` = {string}
- `nomodule` = `true` | `false`
- `nonce` = {string}
- `referrerpolicy` = `no-referrer` | `no-referrer-when-downgrade` | `origin` | `origin-when-cross-origin` | `same-origin` | `strict-origin` | `strict-origin-when-cross-origin` | `unsafe-url`
- `type` = `module` | `text/javascript`

See <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script#Attributes>

# One more thing ... about Client Libraries

## ■ Typical Clientlib Dev Setup

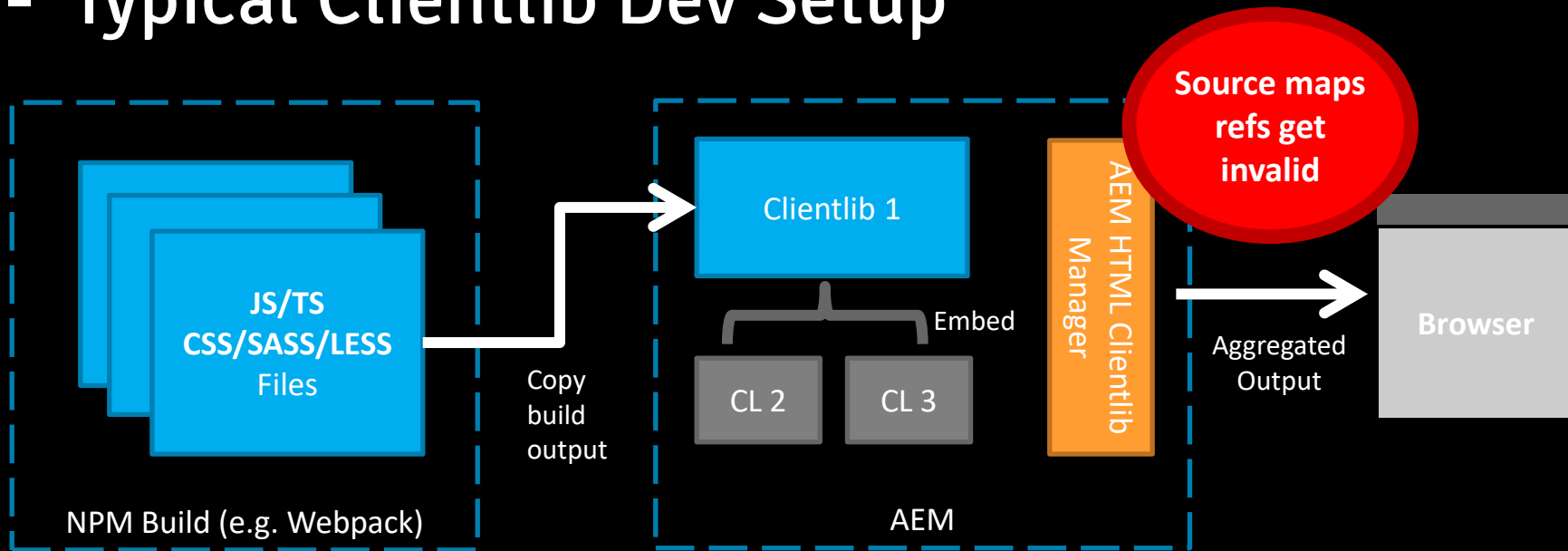


## But – how to debug JS/CSS in AEM Site

- ?debugClientLibs=true ?
  - Broken since ~AEM 6.4 due to changes in storage
  - Not compatible with allowProxy mode
- We want to use **Source Maps**
  - HTML Library Manager is not our friend here

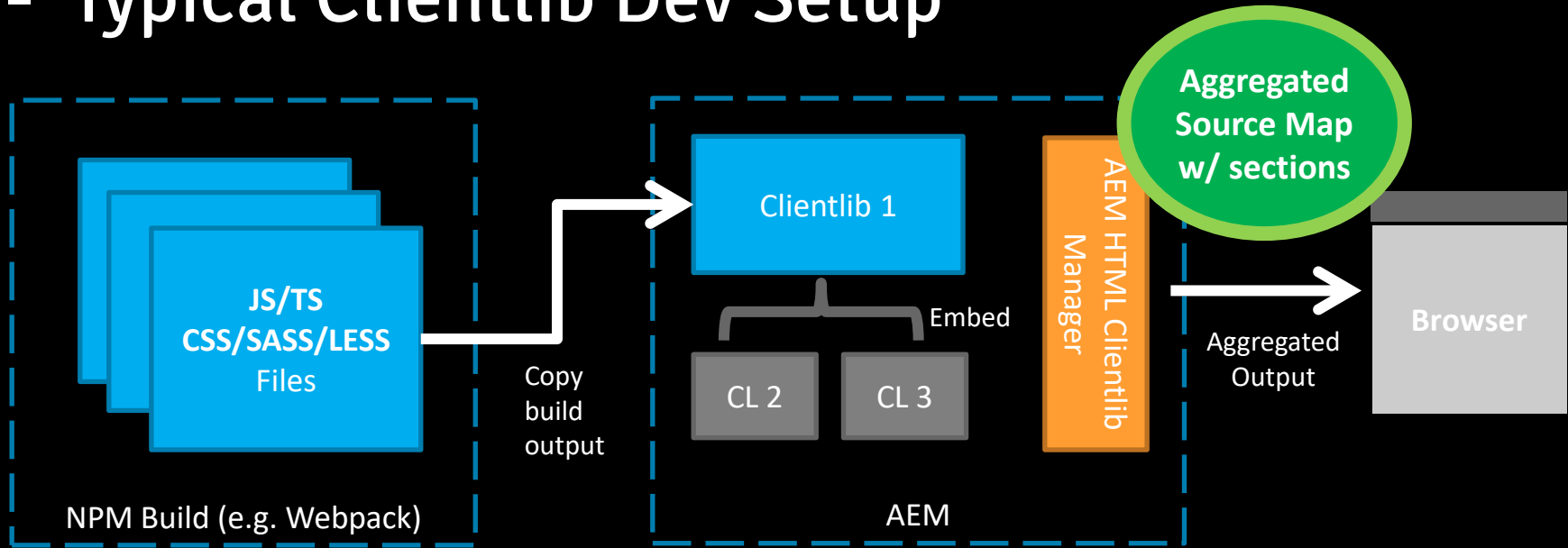
# One more thing ... about Client Libraries

## ■ Typical Clientlib Dev Setup



# One more thing ... about Client Libraries

## ■ Typical Clientlib Dev Setup



# PR ready for Source Map Support in AEM

- Read source map refs from original files
- Recalculate Line Offsets for Embeds
- Output aggregate Source Maps with Sections  
<https://sourcemaps.info/spec.html#h.535es3xeprgt>
- If you need this – come to our mailing list  
<https://wcm.io/mailling-lists.html>