adaptTo()

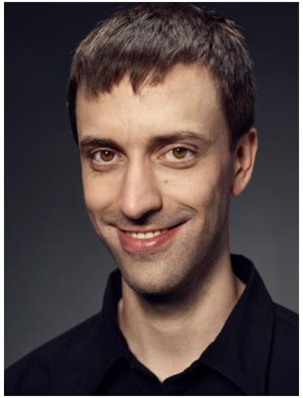
APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 28-30 SEPTEMBER 2015

Building AEM applications with wcm.io

Stefan Seifert, pro!vision GmbH

About the Speaker

- AEM Developer
- Apache Sling Committer
- CTO of pro!vision GmbH



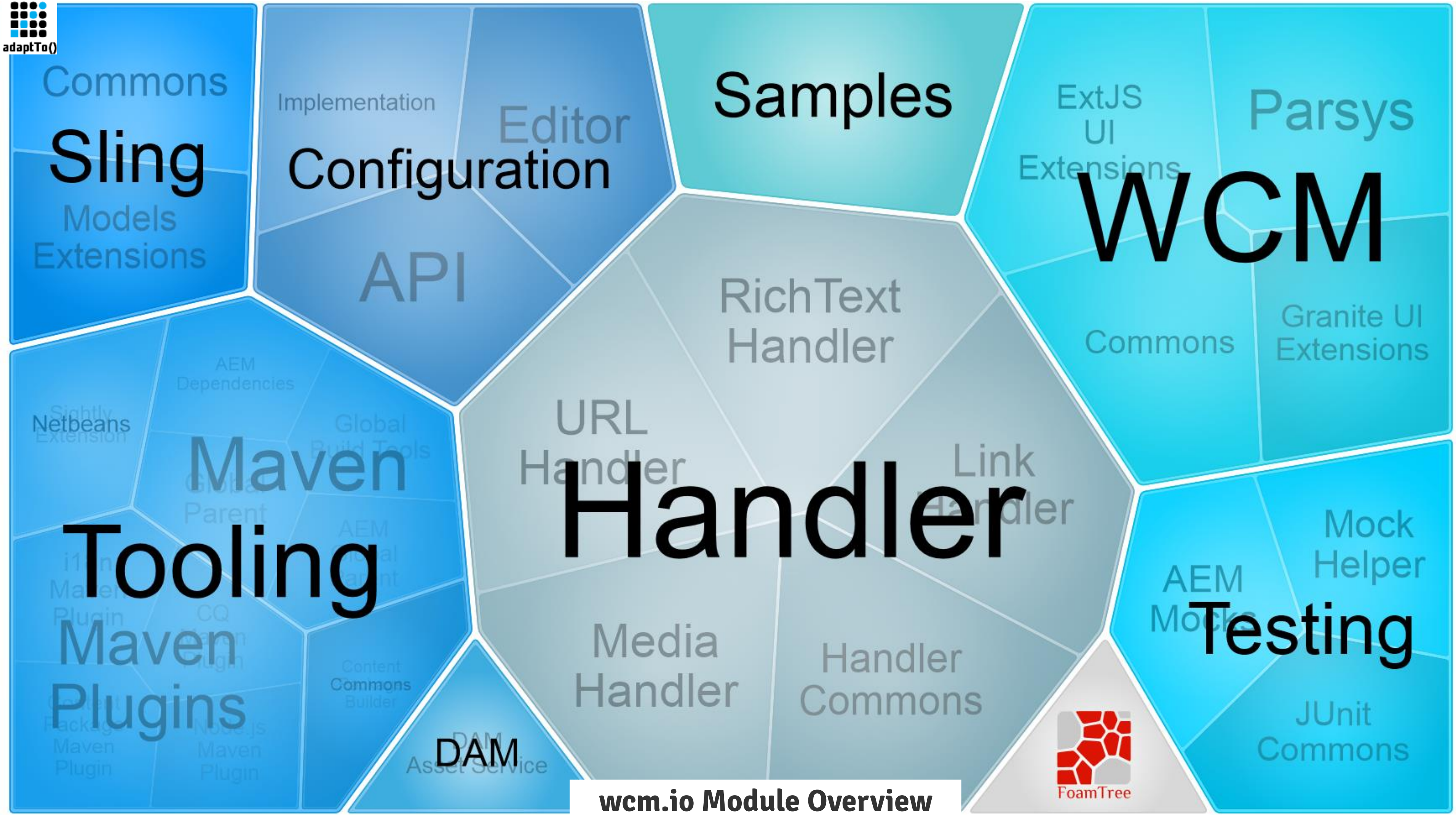
PRO!VISION
SOFTWARE CRAFTSMANSHIP

http://www.pro-vision.de

wcm.io

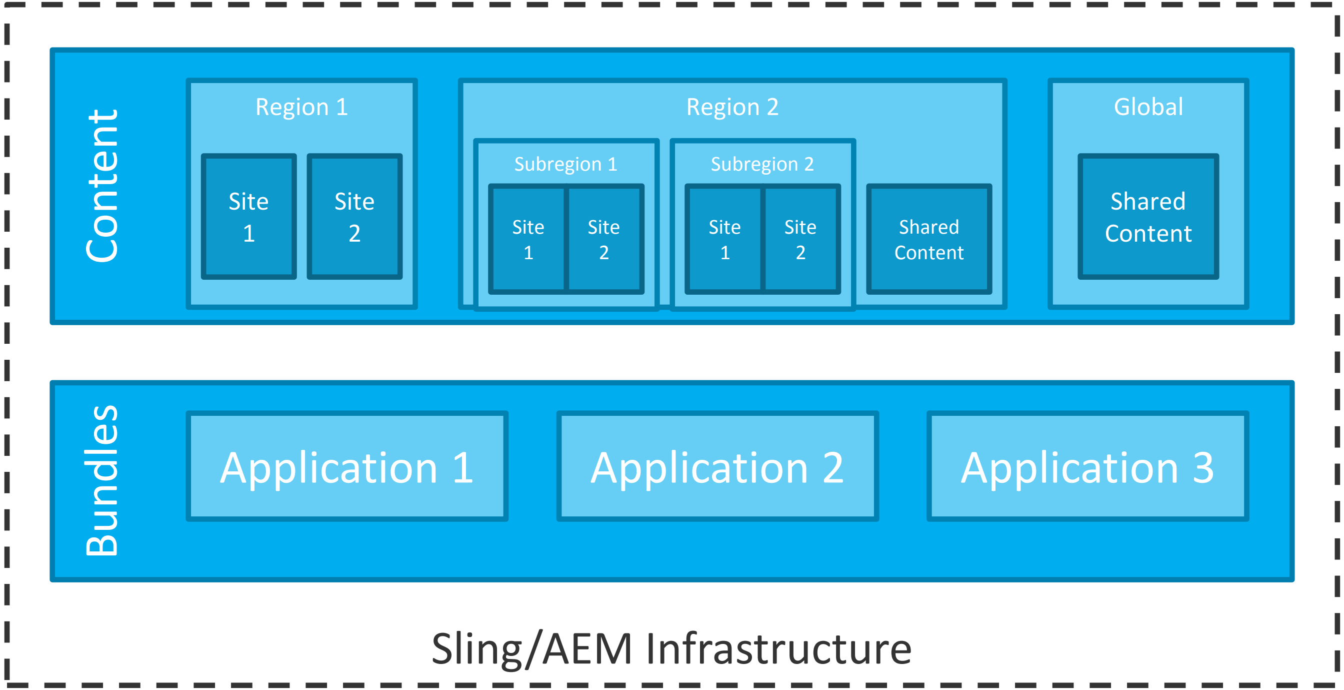
- Open Source Project (ASL 2.0)
- For AEM 6.0 and 6.1 Apps
- Collection of modular libraries
  - Use only what you need

http://wcm.io/



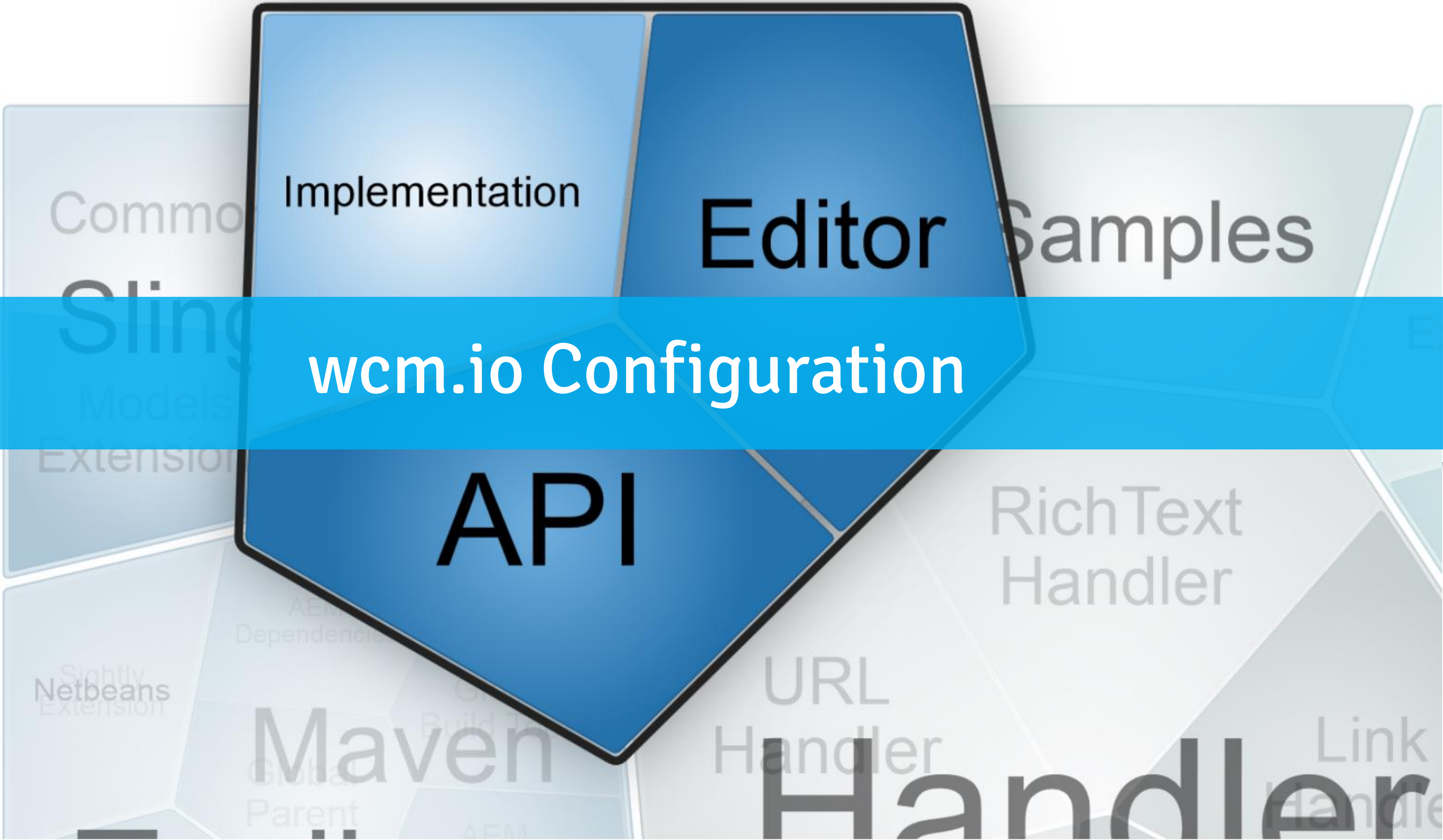
wcm.io Module Overview

Scenario: “Massive Multi Site”



https://cwiki.apache.org/confluence/x/So2uAg

wcm.io Configuration

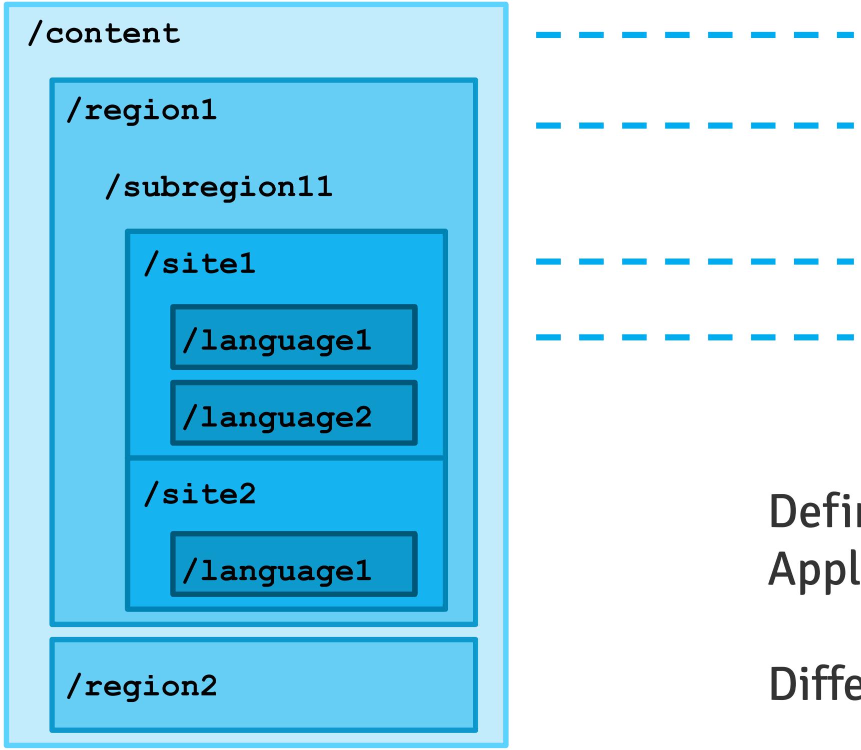


wcm.io Configuration

- Targets context-specific config
  - e.g. Site, Language, Region, Tenant
- Context-specific config cannot be stored (elegant) as OSGi configs
- No standard solution available in Sling or AEM

http://wcm.io/config/

Configuration Scopes Example



Root Configuration and/or OSGi config

Configuration Level 1 (e.g. Region)

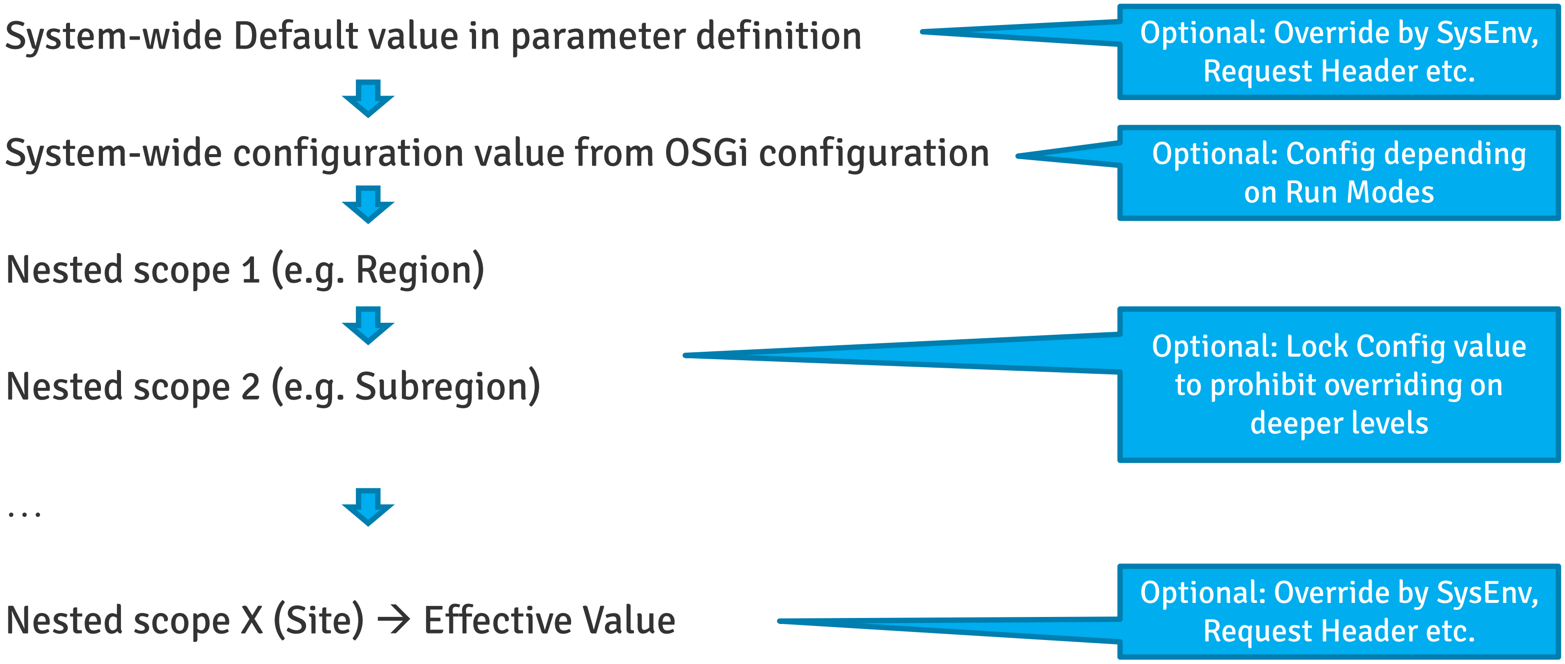
Configuration Level 2 (e.g. Site)

Configuration Level 3 (e.g. Language)

Definition of Configuration levels depend on Application (ConfigurationFinderStrategy)

Different Strategies may co-exist.

Configuration Inheritance



System-wide Default value in parameter definition

Optional: Override by SysEnv, Request Header etc.

System-wide configuration value from OSGi configuration

Optional: Config depending on Run Modes

Nested scope 1 (e.g. Region)

Nested scope 2 (e.g. Subregion)

Optional: Lock Config value to prohibit overriding on deeper levels

…

Nested scope X (Site) → Effective Value

Optional: Override by SysEnv, Request Header etc.

Configuration API Examples

Java Code – ValueMap Syntax

```
Configuration config = resource.adaptTo(Configuration.class);
String value = config.get(“prop1", String.class);
```

Sling Model

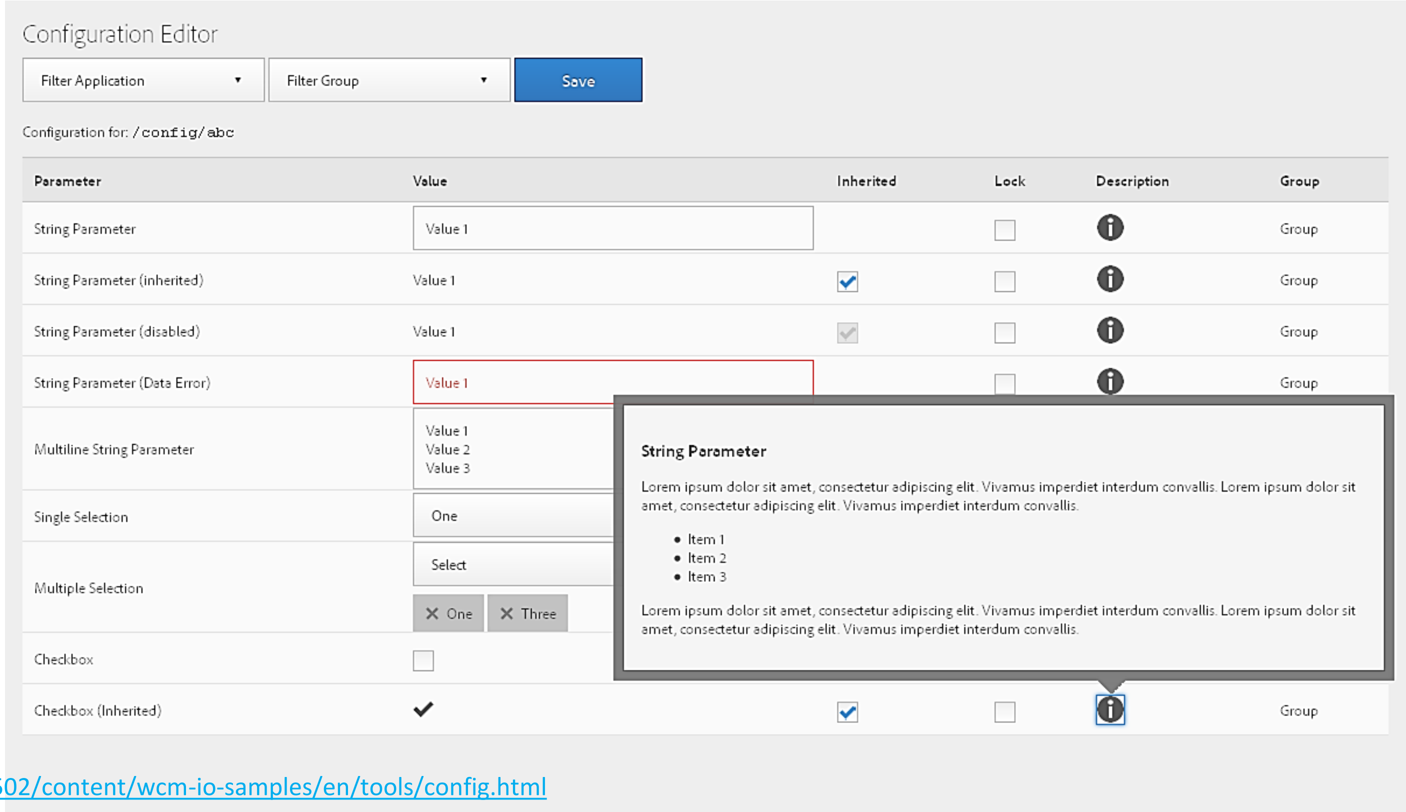
```
@Model(adaptables = { SlingHttpServletRequest.class, Resource.class })
public class MyModel {
  @Self
  Configuration config;
}
```

Sightly Template

```
<div data-sly-text="${config.prop1}"></div>
```

http://wcm.io/config/api/usage-api.html

Configuration Editor



Configuration Editor

Configuration for: `/config/abc`

Parameter	Value	Inherited	Lock	Description	Group
String Parameter	Value 1		□	ⓘ	Group
String Parameter (inherited)	Value 1	☑	□	ⓘ	Group
String Parameter (disabled)	Value 1	☑	□	ⓘ	Group
String Parameter (Data Error)	Value 1		□	ⓘ	Group
Multiline String Parameter	Value 1 Value 2 Value 3				
Single Selection	One				
Multiple Selection	Select ✕ One ✕ Three				
Checkbox	□				
Checkbox (Inherited)	✓	☑	□	ⓘ	Group

String Parameter

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus imperdiet interdum convallis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus imperdiet interdum convallis.

- Item 1
- Item 2
- Item 3

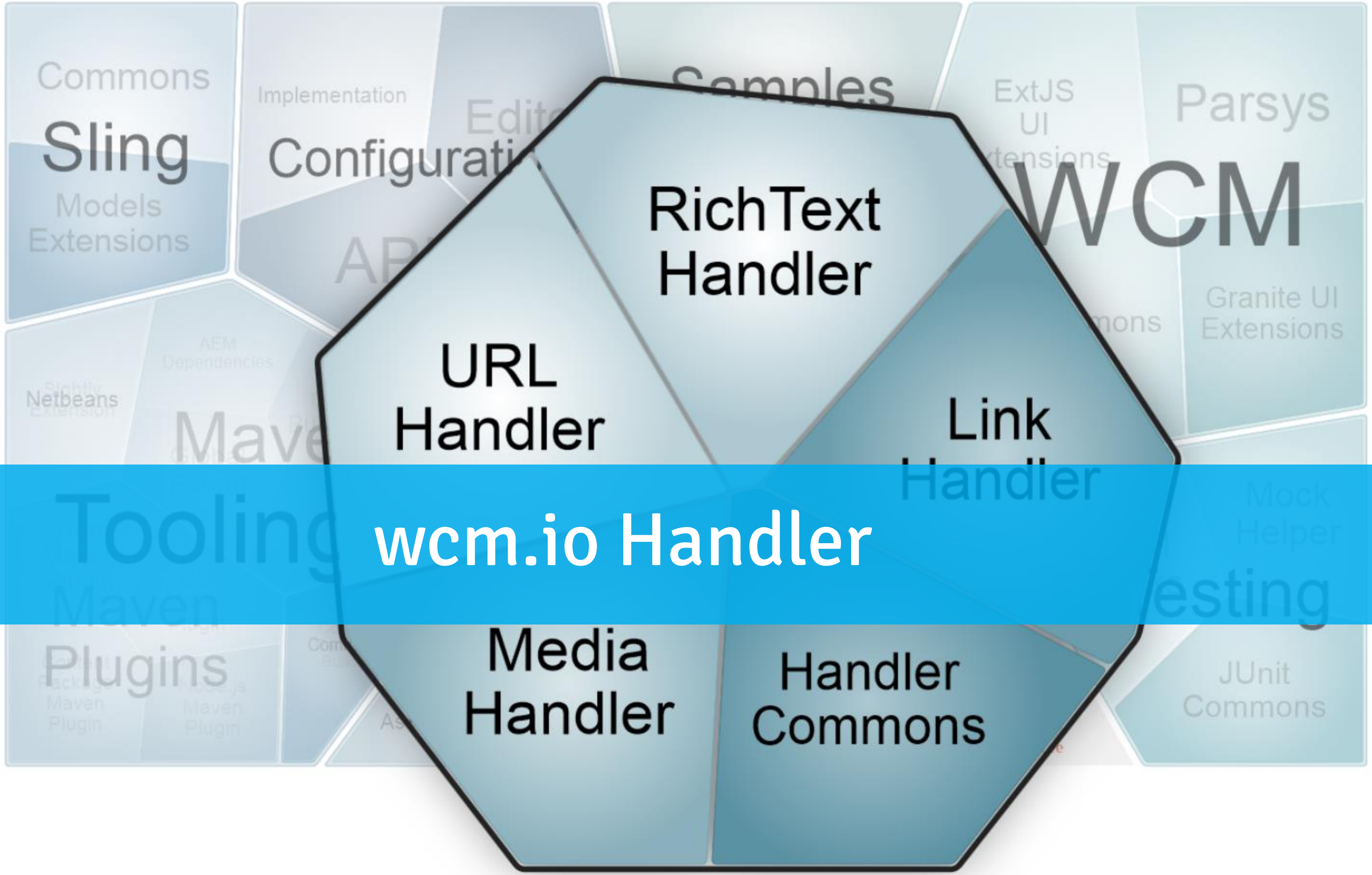
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus imperdiet interdum convallis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus imperdiet interdum convallis.

http://localhost:4502/content/wcm-io-samples/en/tools/config.html

Configuration SPI

- Application Provider
- Parameter Provider
- Persistence Provider
- Override Provider
- Configuration Finder Strategy

http://wcm.io/config/api/usage-spi.html



wcm.io Handler

URL and Link Handler

- Manages URL externalization and Link URL building and massive multi-site scenario
- Configurable per context
- Supports different URL modes and link types
- Customization with pre- and post processors
- APIs based on Sling Models

http://wcm.io/handler/link/

Link Handler Example

```
LinkHandler linkHandler = request.adaptTo(LinkHandler.class);

// build link stored in current resource
Link link = linkHandler.get(resource).build();

// build link targeting the given content page with a different selector and extension
Link link = linkHandler.get(contentPage).selector("sel1").extension("json").build();

// check if link is valid
if (link.isValid()) {
  String markup = link.getMarkup();
  String url = link.getUrl();
  // ...
}
```

Preprocessor example: automatic switch to a sub page

Postprocessor example: Add cross-cutting concerns like user tracking information or additional selectors or URL params

Media Handler

- Builds image/media markup
- Generates virtual renditions on-the-fly
- Targeted for responsive images
- Supports different media source (like DAM)
- Customization with pre- and post processors
- APIs based on Sling Models

http://wcm.io/handler/media/

Media Handler Example

```
MediaHandler mediaHandler = request.adaptTo(MediaHandler.class);

// build media referenced in current resource
Media media = mediaHandler.get(resource).build();

// build media referenced by path with a specific size
Media media = mediaHandler.get("/content/dam/sample/asset1.jpg")
    .fixedDimension(800, 600).build();

// check if media is valid
if (media.isValid()) {
  String markup = media.getMarkup();
  String url = media.getUrl()
  // ...
}
```

Different MarkupBuilders supported e.g. to support different styles to generate markup for responsive images

Media Handler Example II

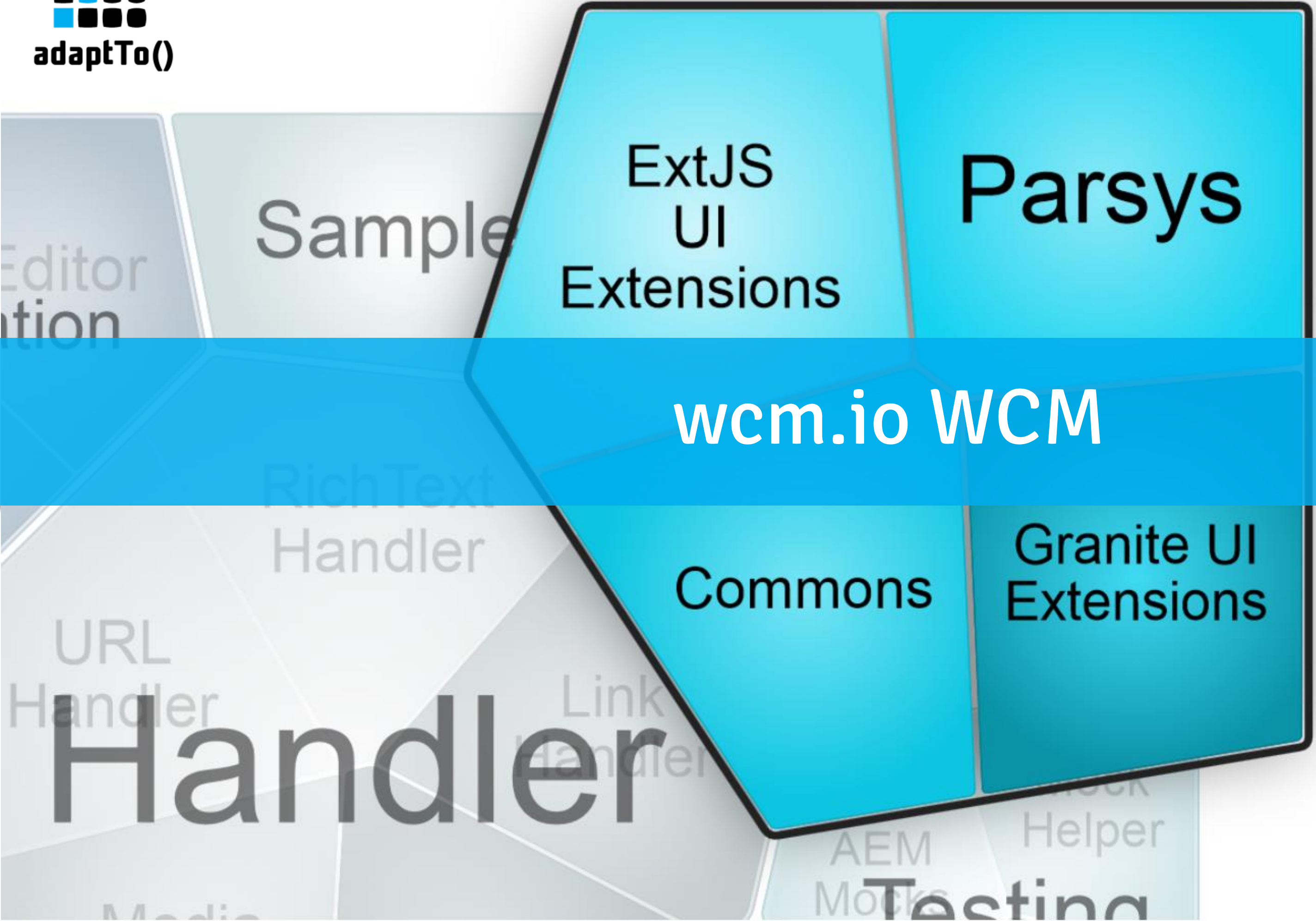
```
MediaHandler mediaHandler = request.adaptTo(MediaHandler.class);

Media media = mediaHandler.get(resource)
    .mandatoryMediaFormats(new ResponsiveMediaFormatsBuilder(MediaFormats.MF_16_9)
        .breakpoint("S0", 712, 400)
        .breakpoint("M1", 401, 225)
        .breakpoint("L1", 837, 471)
        .build())
    .build();
```

Resulting markup:

```
<img alt="Alt. Text" data-resp-src="[
  {'mq':'S0','src':'/path/mymedia.712.400.jpg'},
  {'mq':'M1','src':'/path/mymedia.401.225.jpg'},
  {'mq':'L1','src':'/path/mymedia.837.471.jpg'}]"/>
```

Responsive image markup builder generates “legacy” markup supporting devices without proper HTML5 capabilities.

You can switch to a markup builder generating HTML5 `picture/source` markup.


ExtJS UI Extensions
Parsys
Commons
Granite UI Extensions
Sample
Editor
tion
Rich Text Handler
URL Handler
Handler
Link Handler
Mock Helper
AEM Mocks
Testing
Media

wcm.io WCM

WCM Parsys

- Alternative to AEM6 parsys
- Not based on Design mode
- Parsys definitions stored in Template’s page components
- Inheritance of parsys rules
  based on resource type hierarchy

http://wcm.io/wcm/parsys/

Configure allowed components

```
"wcmio:parsysConfig": {
  "paths": {

    "content": {
      "allowedChildren": ["/apps/myapp/components/component1"]
    },

    "leftcol_teasers": {
      "path": "jcr:content/lefcol/teasers",
      "allowedChildren": [
        "/apps/myapp/components/component1",
        "/apps/myapp/components/component1"
      ]
    }

}
```

Allowed components in `jcr:content/content` node

Allowed components in `jcr:content/leftcol/teasers` node

Configure allowed components

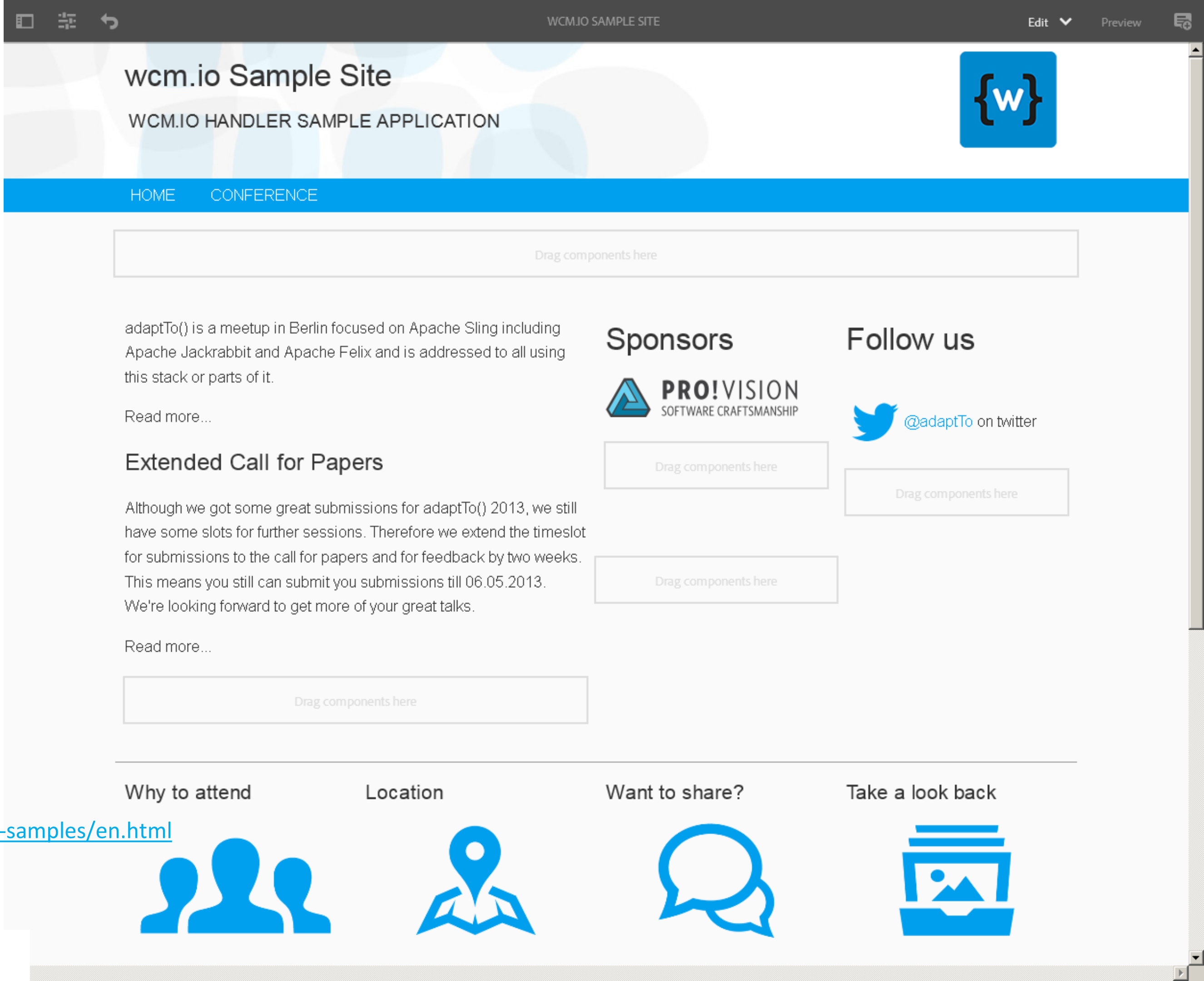
```
"wcmio:parsysConfig": {
  "paths": {

    "nested_parsys": {
      "pattern": "^jcr:content/.*$",
      "parentAncestorLevel": 2,
      "allowedParents": [
        "/apps/myapp/components/component1"
      ],
      "allowedChildren": [
        "/apps/myapp/components/subcomponent1",
        "/apps/myapp/components/subcomponent2"
      ]
    }

  }
}
```

Allowed components in a nested parsys that is contained in component1

Demo

- Demo TouchUI
- ClassicUI supported as well

http://localhost:4502/editor.html/content/wcm-io-samples/en.html



wcm.io Sample Site

WCM.IO HANDLER SAMPLE APPLICATION

adaptTo() is a meetup in Berlin focused on Apache Sling including Apache Jackrabbit and Apache Felix and is addressed to all using this stack or parts of it.

Read more...

Extended Call for Papers

Although we got some great submissions for adaptTo() 2013, we still have some slots for further sessions. Therefore we extend the timeslot for submissions to the call for papers and for feedback by two weeks. This means you still can submit you submissions till 06.05.2013. We're looking forward to get more of your great talks.

Read more...

Sponsors

Follow us

@adaptTo on twitter

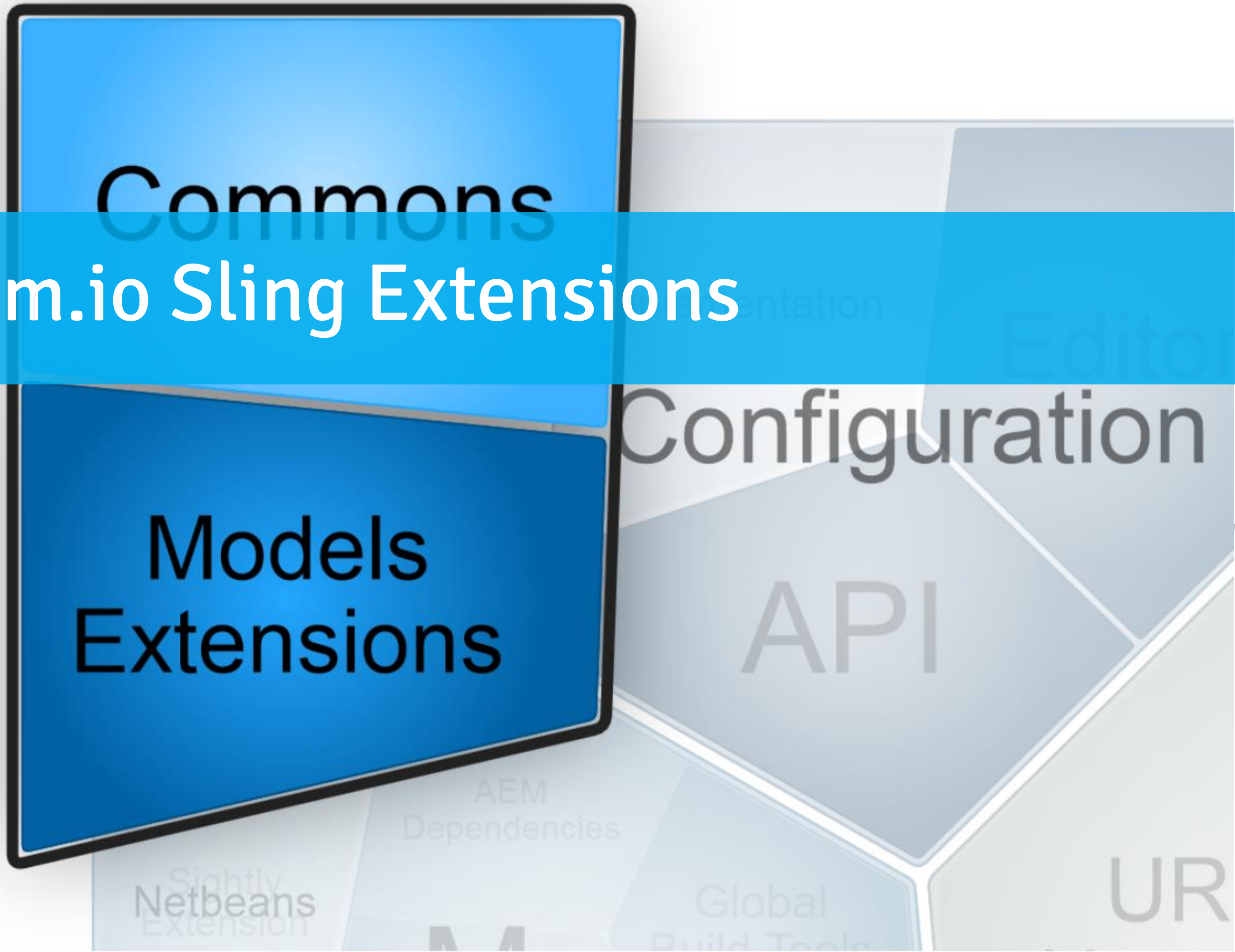
Why to attend

Location

Want to share?

Take a look back

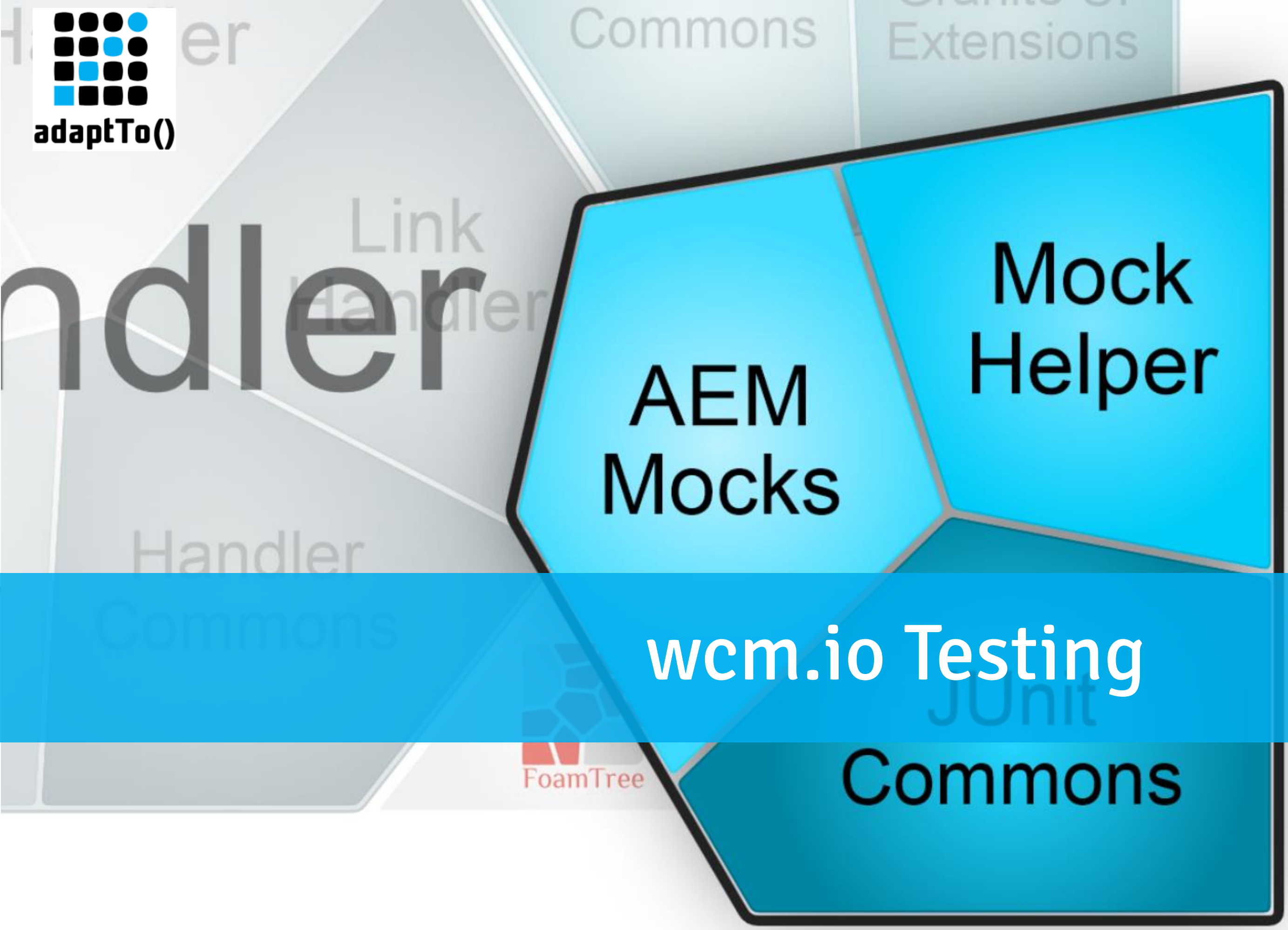
wcm.io Sling Extensions



Sling Models Extensions

- `@AemObject` injector
- Inject PageManager, Page, WCMMode, TagManager, I18n and others
- Support injection request-derived context objects on all models, not only when the adaptable is a request (configurable)

http://wcm.io/sling/models/
http://wcm.io/sling/models/apidocs/io/wcm/sling/models/annotations/AemObject.html



wcm.io Testing

AEM Mocks for Unit Tests

- Mock implementation of AEM WCM API
- Based on Sling Mocks, OSGi Mocks, JCR Mocks, ResourceResolver Mocks

JUnit Rule

```
@Rule
public AemContext context = new AemContext();
```

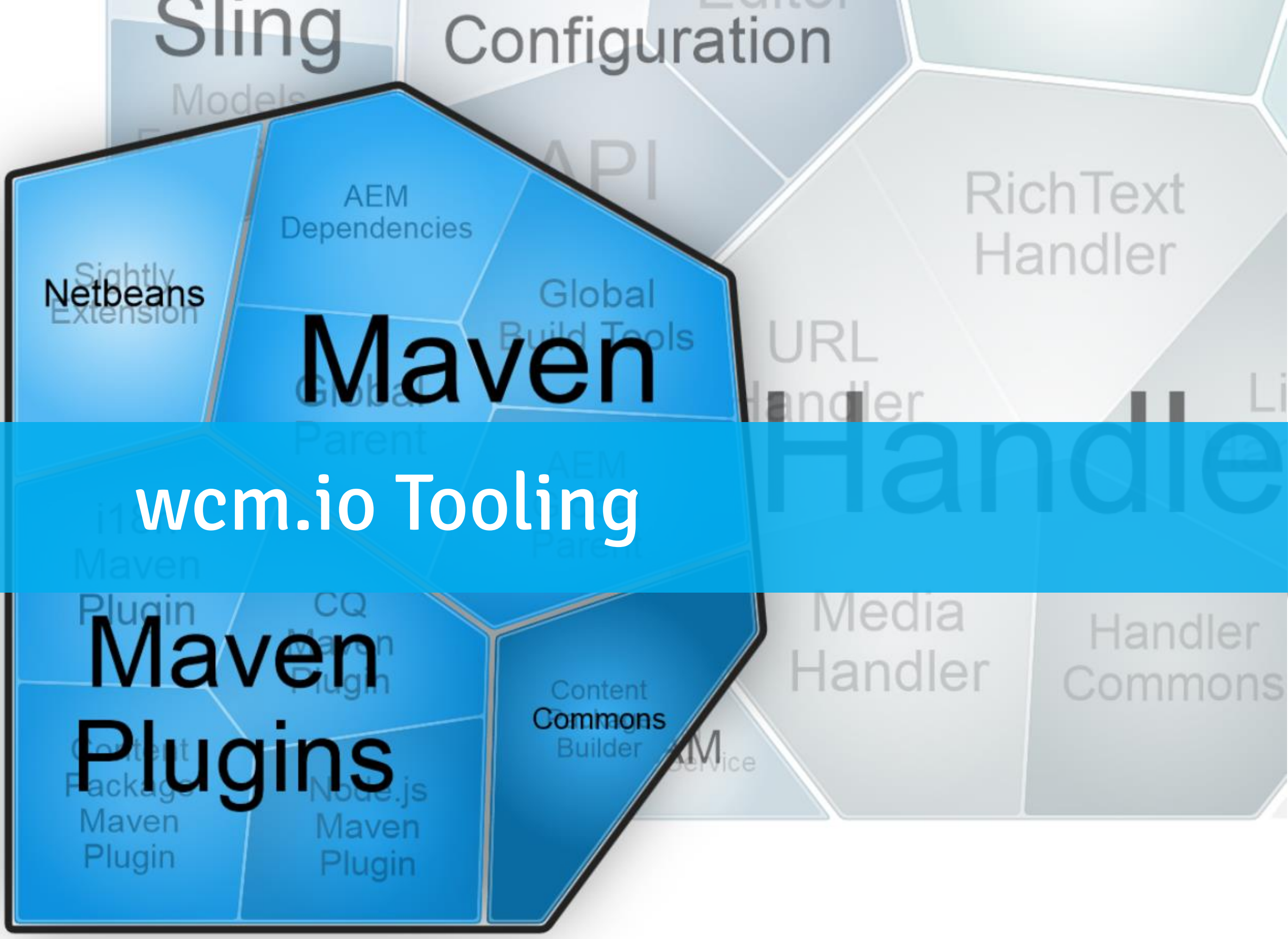
http://wcm.io/testing/aem-mock/

AEM WCM API Mocks

- PageManager, Page, Designer
- ComponentManager, Component
- TagManager, Tag
- DAM Asset, Rendition
- Content Builder extensions

```
context.create().page(…)
context.create().asset(…)
```

wcm.io Tooling



wcm.io Tooling

- Tooling for AEM projects
- Examples:
  - Enhanced Content Package Maven Plugin
  - i18n Maven Plugin to produce Sling Message i18n Format from Property Files
  - Node.js Maven Plugin to integrate Grunt, Gulp and others into your Maven Build

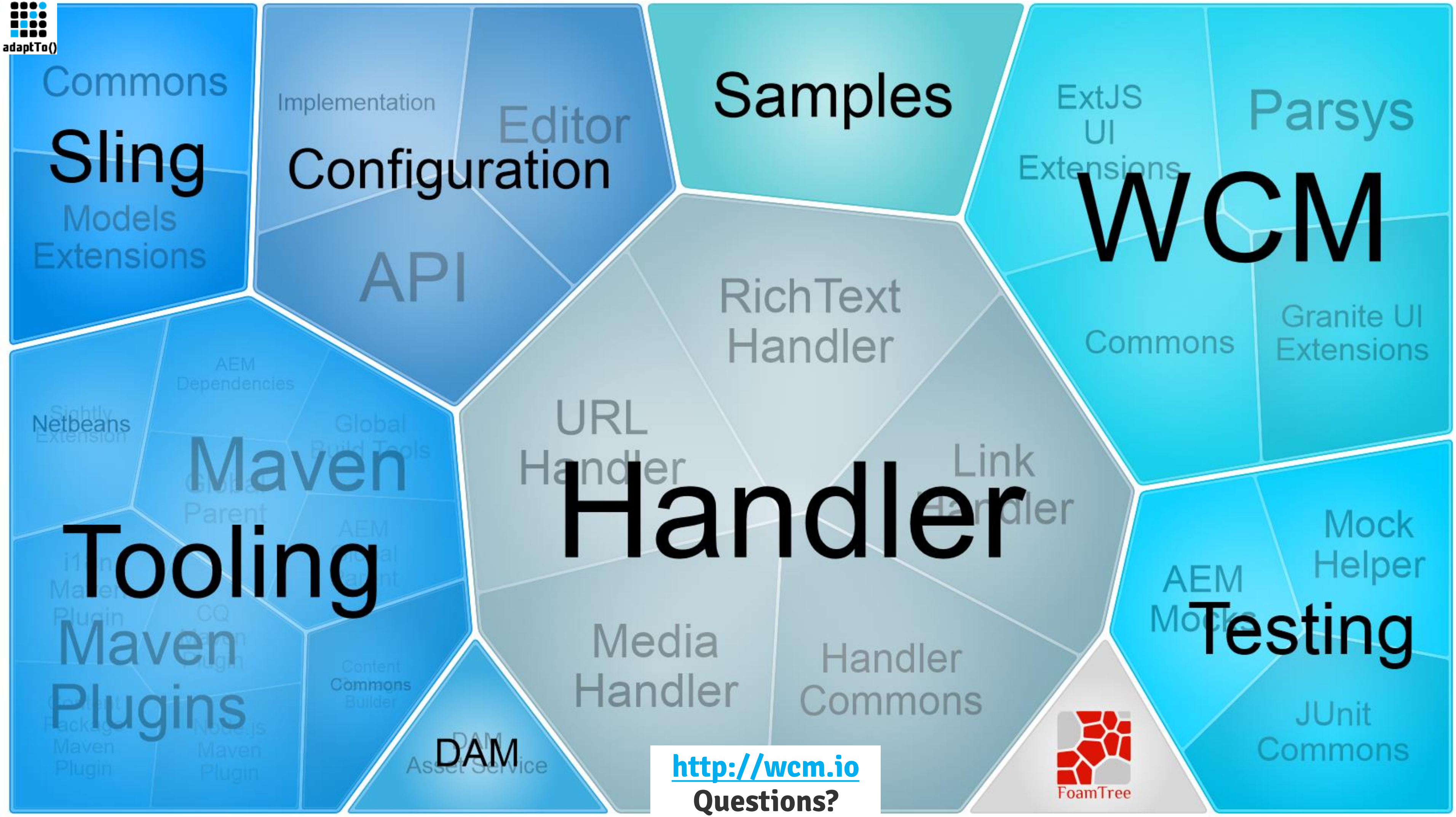
http://wcm.io/tooling/

wcm.io Samples

wcm.io Sample

- wcm.io Sample project on GitHub
  https://github.com/wcm-io/wcm-io-samples
- Uses full wcm.io Stack
- Integration with AEM Author and Sightly
- Tooling usage
- CONGA integration

http://wcm.io/tooling/



http://wcm.io

Questions?