



# Adobe AEM Core Components

Lokesh BS| TechAspect Solutions Inc

# Highlights

## Feature-rich capabilities

Flexible Configuration options  
Pre-configurable capabilities

## Best implementation

HTML Template Language (HTL)  
Sling Models

## Modern markup

Bootstrap naming conventions  
Built around accessibility guidelines  
Responsive

## Component Versioning

Compatibility within a version  
Multiple versions to coexist on the same environment

## Continuous delivery

Frequent & incremental functionality improvements  
source code on GitHub for feedback and contribute  
component upgrades to be done independently from AEM upgrades

## Global Output

Capability to serialize as JSON the content model for headless CMS use cases



- ✓ Core Components require AEM 6.3 and Java 8.
- ✓ Core Components do not work with the Classic UI.

- Open Source and available on github
- In production instance, need to install manually

# Foundation VS Core Components

Capability	Core Component	Foundation Component
Logic implementation	Java POJOs with <u>Sling Models</u> annotations	JSP code
Markup definition	<u>HTML Template Language</u> (HTL) syntax	JSP code
XSS sanitization	Automated by HTL	Mostly manual
CSS classes naming	Standardized naming convention inspired by <u>Bootstrap</u>	Custom schemes
Dialog definition	<u>Coral 3</u>	Coral 2 + Classic UI
JSON output	<u>Sling Models Exporter with Jackson serialization</u>	Default Sling servlet
Versioning	<u>For the model and the HTL</u>	None
Testing	Unit Tests + Integration Tests	Integration Tests
Delivery	<u>Via public GitHub</u>	Via Quickstart
License	<u>Apache License</u>	Adobe proprietary
Contribution	Via pull request	Not possible

# CORE COMPONENTS

- Access



# Components List

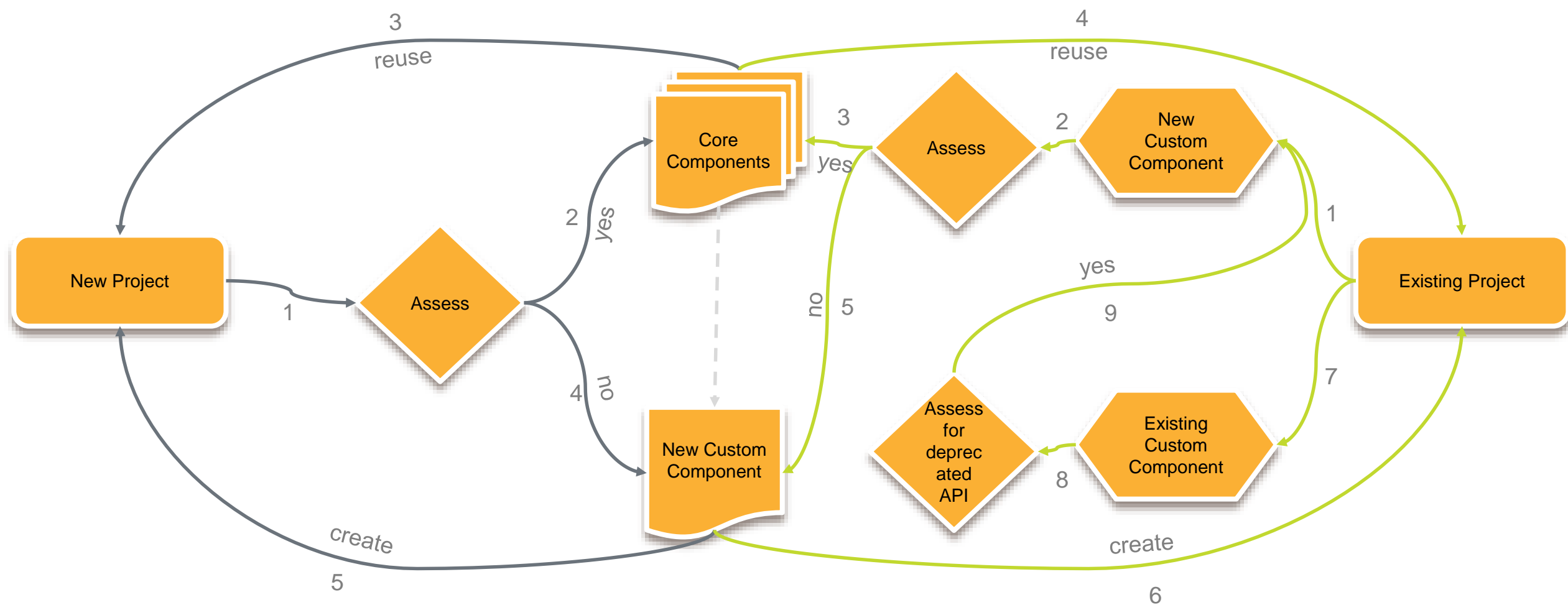
- <https://github.com/Adobe-Marketing-Cloud/aem-core-wcm-components>

## Available Components

- Page authoring components:
  - [Page component](#)
  - [Breadcrumb component](#)
  - [Title component](#)
  - [Text component](#)
  - [Image component](#)
  - [List component](#)
  - [Sharing component](#)
- Form components:
  - [Form container](#)
  - [Form text field](#)
  - [Form options field](#)
  - [Form hidden field](#)
  - [Form button](#)



# When to use ?



# How to Use ?

Create a proxy component



superType to core component



Define the proxy component



Load the core styles (if needed)





# Implementation

```
resource [ /content/my-site/my-page/jcr:content/root/title
           jcr:primaryType = "nt:unstructured"
           sling:resourceType = "my-site/components/title"

proxy cmp [ /apps/my-site/components/title
            jcr:primaryType = "cq:Component"
            sling:resourceSuperType = "core/wcm/components/title/v1/title"

core cmp [ /apps/core/wcm/components/title/v1/title
           jcr:primaryType = "cq:Component"
           /title.html
           data-sly-use.model="com.adobe.cq.wcm.core.components.models.Title"

model [ package com.adobe.cq.wcm.core.components.models;
        public interface Title { ... }

implementation [ package com.adobe.cq.wcm.core.components.models.impl.v1;
                @Model(adaptables = SlingHttpServletRequest.class,
                      adapters = Title.class,
                      resourceType = "core/wcm/components/title/v1/title")
                public class TitleImpl implements Title { ... }
```

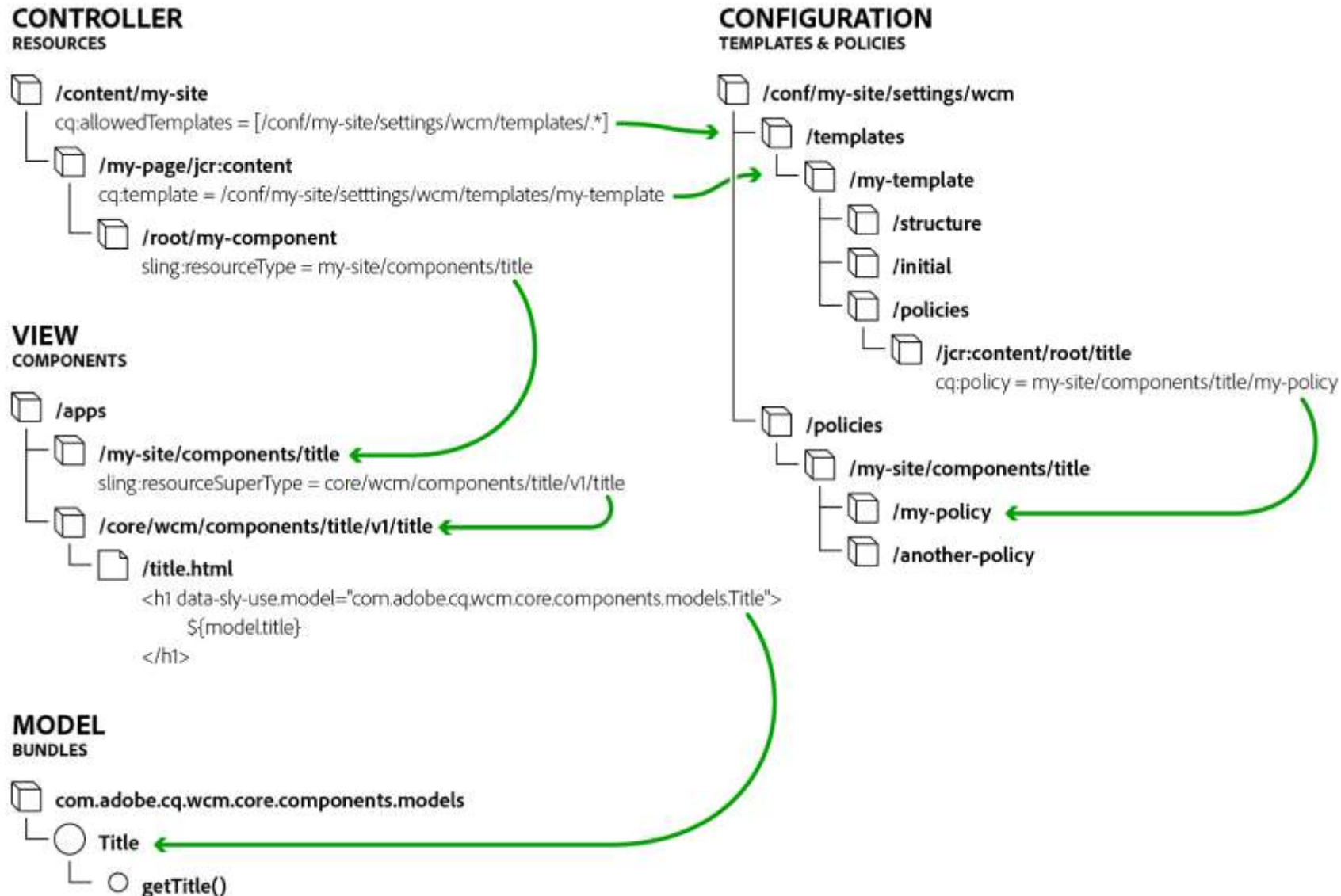
The diagram illustrates the implementation hierarchy of a Sling resource. It shows five levels: resource, proxy cmp, core cmp, model, and implementation. Arrows indicate the flow of implementation from the implementation class back to the resource definitions.

- resource**: `/content/my-site/my-page/jcr:content/root/title`
  - `jcr:primaryType = "nt:unstructured"`
  - `sling:resourceType = "my-site/components/title"`
- proxy cmp**: `/apps/my-site/components/title`
  - `jcr:primaryType = "cq:Component"`
  - `sling:resourceSuperType = "core/wcm/components/title/v1/title"`
- core cmp**: `/apps/core/wcm/components/title/v1/title`
  - `jcr:primaryType = "cq:Component"`
  - `/title.html`
  - `data-sly-use.model="com.adobe.cq.wcm.core.components.models.Title"`
- model**: `package com.adobe.cq.wcm.core.components.models;`
  - `public interface Title { ... }`
- implementation**: `package com.adobe.cq.wcm.core.components.models.impl.v1;`
  - `@Model(adaptables = SlingHttpServletRequest.class,`
  - `adapters = Title.class,`
  - `resourceType = "core/wcm/components/title/v1/title")`
  - `public class TitleImpl implements Title { ... }`

DEMO



# Associations



# References

- Download Core Components – <https://github.com/Adobe-Marketing-Cloud/aem-core-wcm-components>
- Core Components- <https://docs.adobe.com/docs/en/aem/6-3/develop/components/core-components.html>
- Component Guidelines - <https://docs.adobe.com/docs/en/aem/6-3/develop/components/core-components/guidelines.html>
- Using Core Components - <https://docs.adobe.com/docs/en/aem/6-3/develop/components/core-components/using.html>



THANK YOU