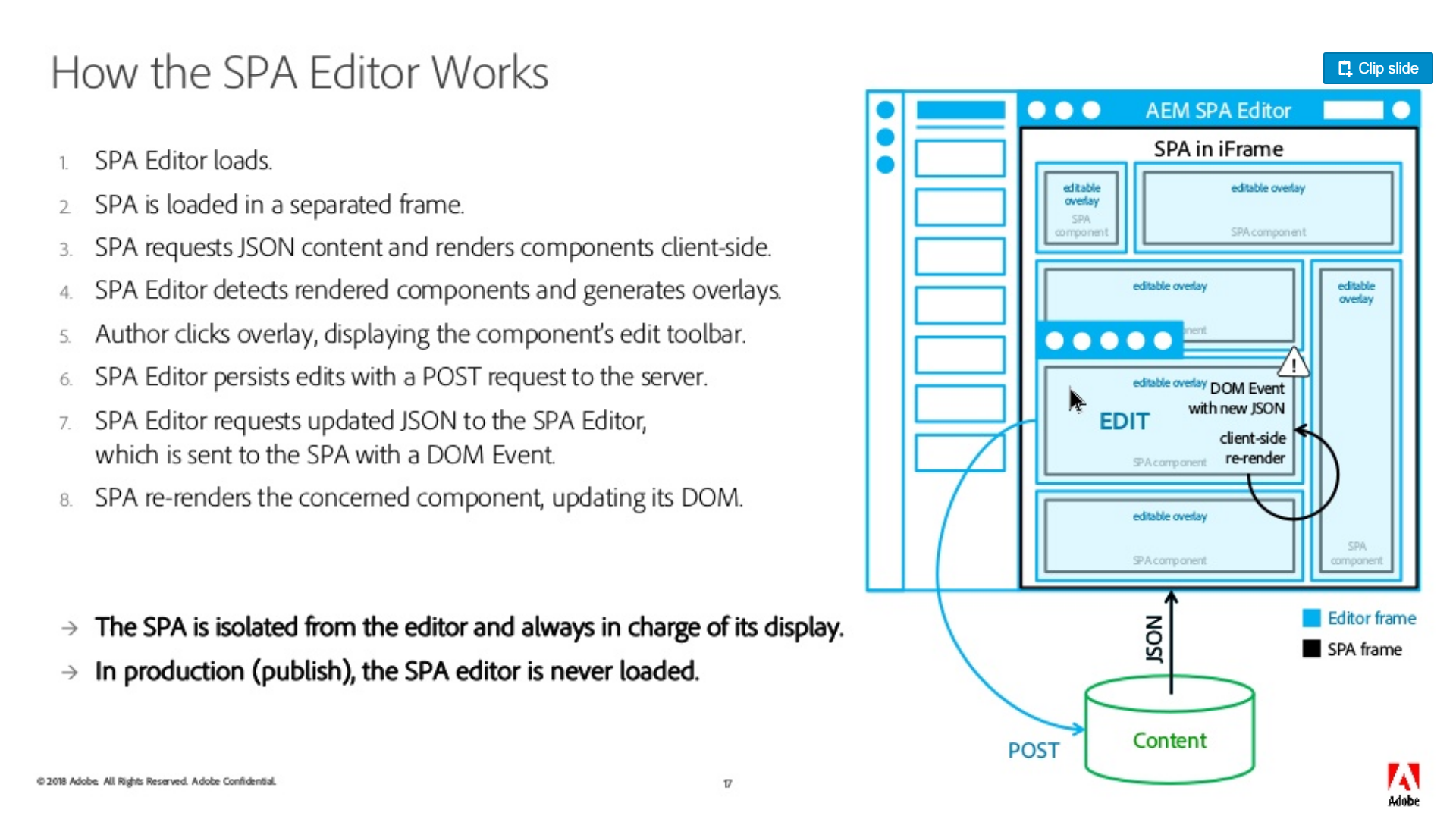
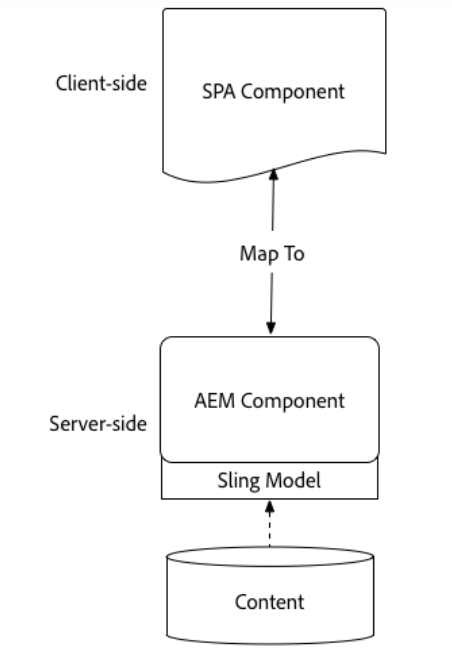
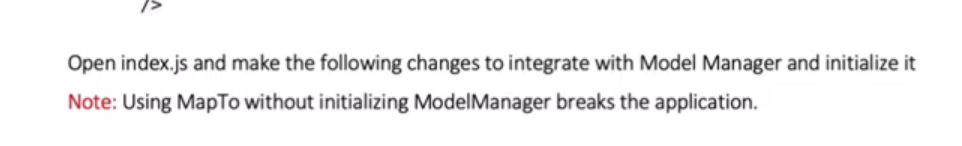
# SPA

The basic concept is to map a SPA Component to an AEM Component. A 1:1 mapping between SPA components and an AEM component is created.

* Sling Models are also used in the SPA use case to implement server-side business logic. The difference is that in the SPA use case, the Sling Models exposes it's methods as serialized JSON.
* The MapTo method maps the SPA component to the AEM component
* The ComponentExporter interface must be implemented in order for the Sling Model to be compatible with AEM Content Services.







# cq:pagemodel\_root\_url

***cq:pagemodel\_root\_url*** is specified as a meta property in the <head> tag.

<meta property="cq:pagemodel\_root\_url" data-sly-use.page="com.cisco.wem.intelligentcontent.models.HierarchyPage" content="${page.rootUrl}"/>

* The page component on which we build a SPA doesn’t give HTML elements of its child component (only provides <app-root>) and this is delegated to SPA framework (angular or react)
* The representation of child components is fetched as a JSON data structure (i.e. the model). The SPA components are then added to the page according to the provided JSON model
* The resolution and delegation of page model is delegated to a separate npm **PageModelManager** module
* To allow the page to be authored, a client library named **cq.authoring.pagemodel.messaging** must be added to provide a communication channel between the SPA and the page editor.
* The communication data type is set to an HTML element inside the AEM Page component using the **data-cq-datatype** attribute. When the communication data type is set to JSON, the GET requests hit the Sling Model endpoints of a component. After an update occurs in the page editor, the JSON representation of the updated component is sent to the Page Model library. The Page Model library then warns the SPA of updates.

<meta property="cq:datatype" data-sly-test="${wcmmode.edit||wcmmode.preview}"content="JSON"/>

# Limitations

The AEM SPA Editor SDK was introduced with AEM 6.4 service pack 2. It is fully supported by Adobe, and as a new feature it continues to be enhanced and expanded.

SPA Editor JS SDK is useful to make the content produced by SPA editable in AEM.

The following AEM features are not yet covered by the SPA Editor:

* Target mode
* ContextHub
* Inline image editing
* Edit configs (eg. listeners)
* Style System
* Undo / Redo
* Page diff and Time Warp
* Features performing HTML rewriting server-side such as Link Checker, CDN rewriter service, URL shortening etc.
* Developer mode
* AEM Launches

# PageModelManager - [@adobe/cq-spa-page-model-manager](https://www.npmjs.com/package/@adobe/cq-spa-page-model-manager)

The PageModelManager library is provided as an NPM package to be used by an SPA project. It accompanies the SPA and serves as a data model manager.

It abstracts the retrieval and management of the JSON structure that represents the actual content structure. It is also responsible for syncing with the SPA to let it know when it has to re-render its components.

# ComponentMapping

# Naku Ardam aindi

1. The content in aem page is served to SPA (angular/react) in the form of JSON by sling models API
2. The model of the page is used to map and instantiate SPA components (page model)
3. In SPA , equal no of components (angular components) are created to map to corresponding AEM components
4. The JSON has ***:items*** in its structure which helps SPA in mapping its components appropriately
5. SPA editor is provided to make in-place editing of components on author instance and provide the updated page model to SPA (angular/react apps) ,to intimate them about which components are to be re-rendered for the content updates.

# Useful Resources

<https://docs.adobe.com/content/help/en/experience-manager-64/developing/headless/spas/spa-blueprint.html>

Overview of SPA :

<https://www.slideshare.net/GabrielWalt/spa-editor-adobe-experience-manager-sites>