

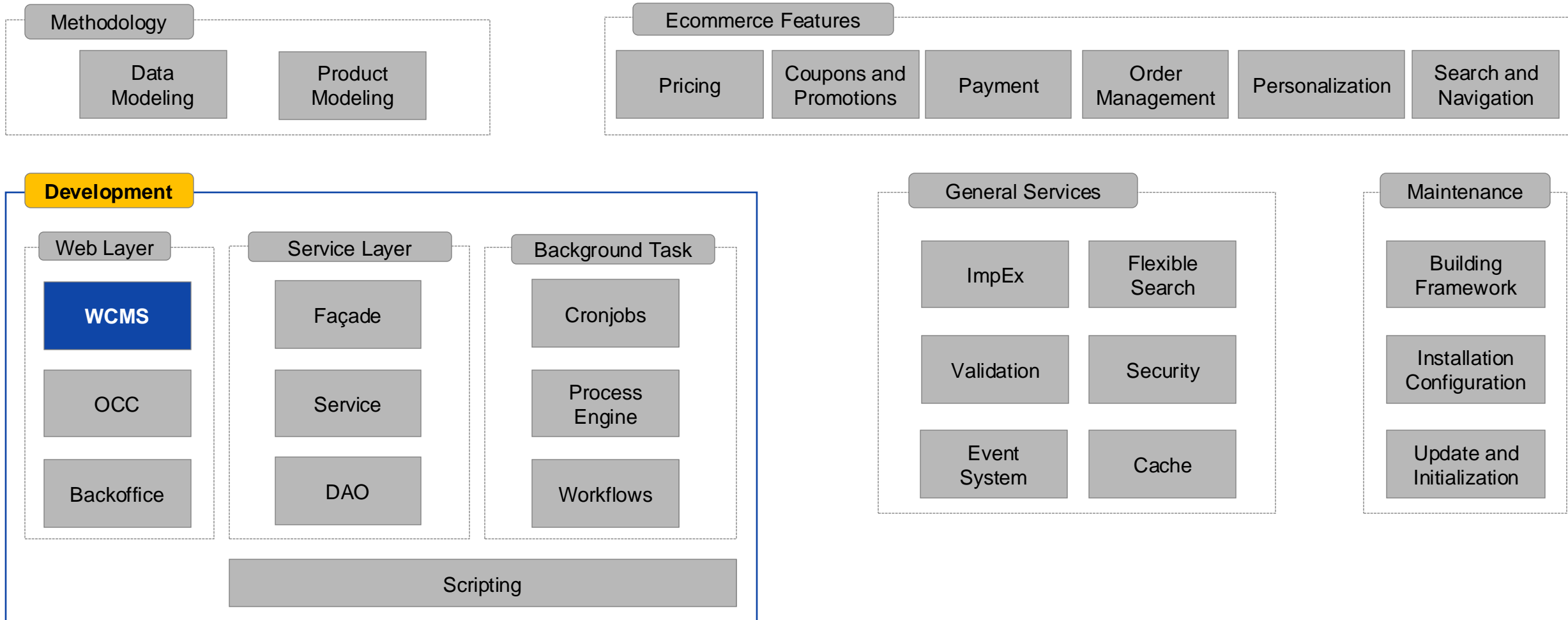


SAP Customer Experience

WCMS

INTERNAL – SAP and Partners Only

What we will cover in this topic



We will learn about:

- Storefront Management
- CMS Model
- Slots and Components
- View Rendering
- Personalization
- SmartEdit
- Website Links

The Context



SAP Commerce Cloud WCMS Module provides a **web content management and publishing system** that enables you to easily **maintain website contents**.

Preparation

Complete step P1 of the WCMS exercise

(The setup ant target will compile your system during the lecture)



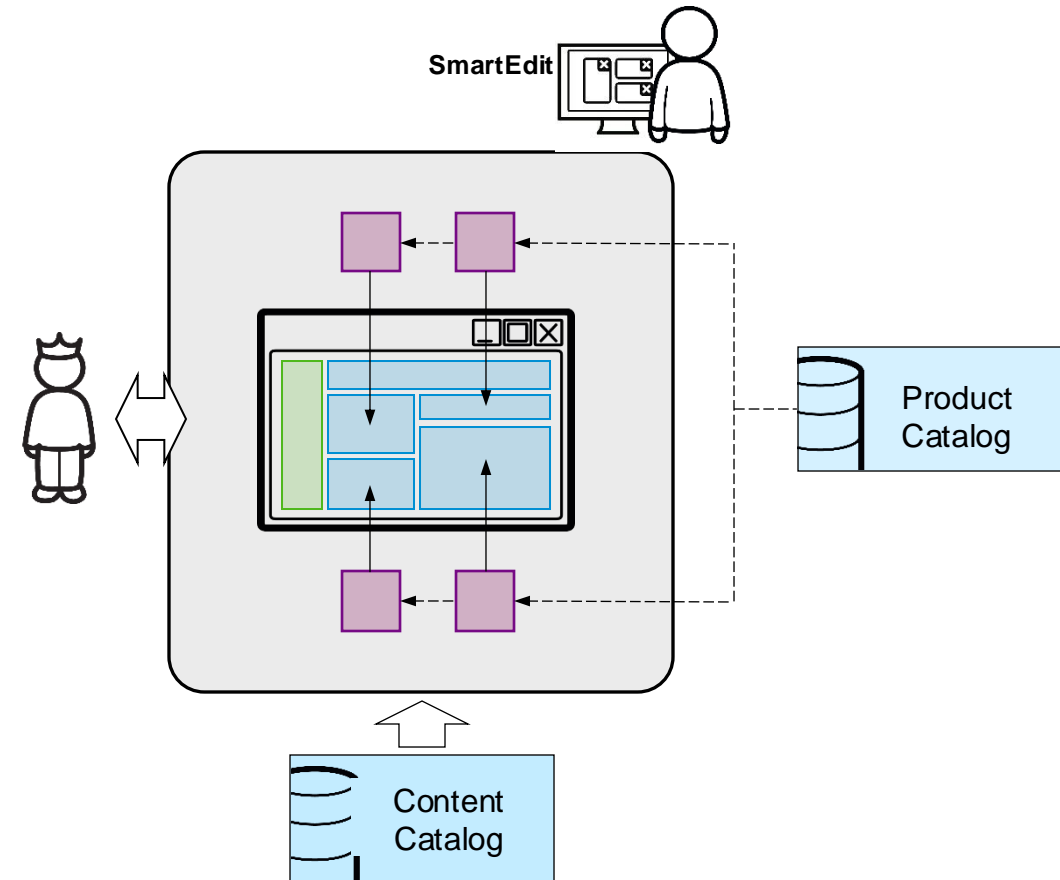
Storefront Management



Defining a page on the Storefront

SAP Commerce Cloud storefront pages are defined and assembled by a Web Content Management System (WCMS) that provides authoring, collaboration and administration tools.

- Pages are defined in SmartEdit
 - Allows users with little knowledge of web programming to create and manage storefront content
- Manage dynamic and static pages
 - Homepage, information pages, product/category pages, or search result pages
- Personalize content based on context
 - Specific user/group, time, product or category, customer segment
- Live edit and preview
 - Change content on the fly
 - Dynamically change user context for testing



Out-of-the-Box WCMS Page Types

Category Page

- Display the contents of a product category
- Example: Product List Page, Product Grid Page

Content Page

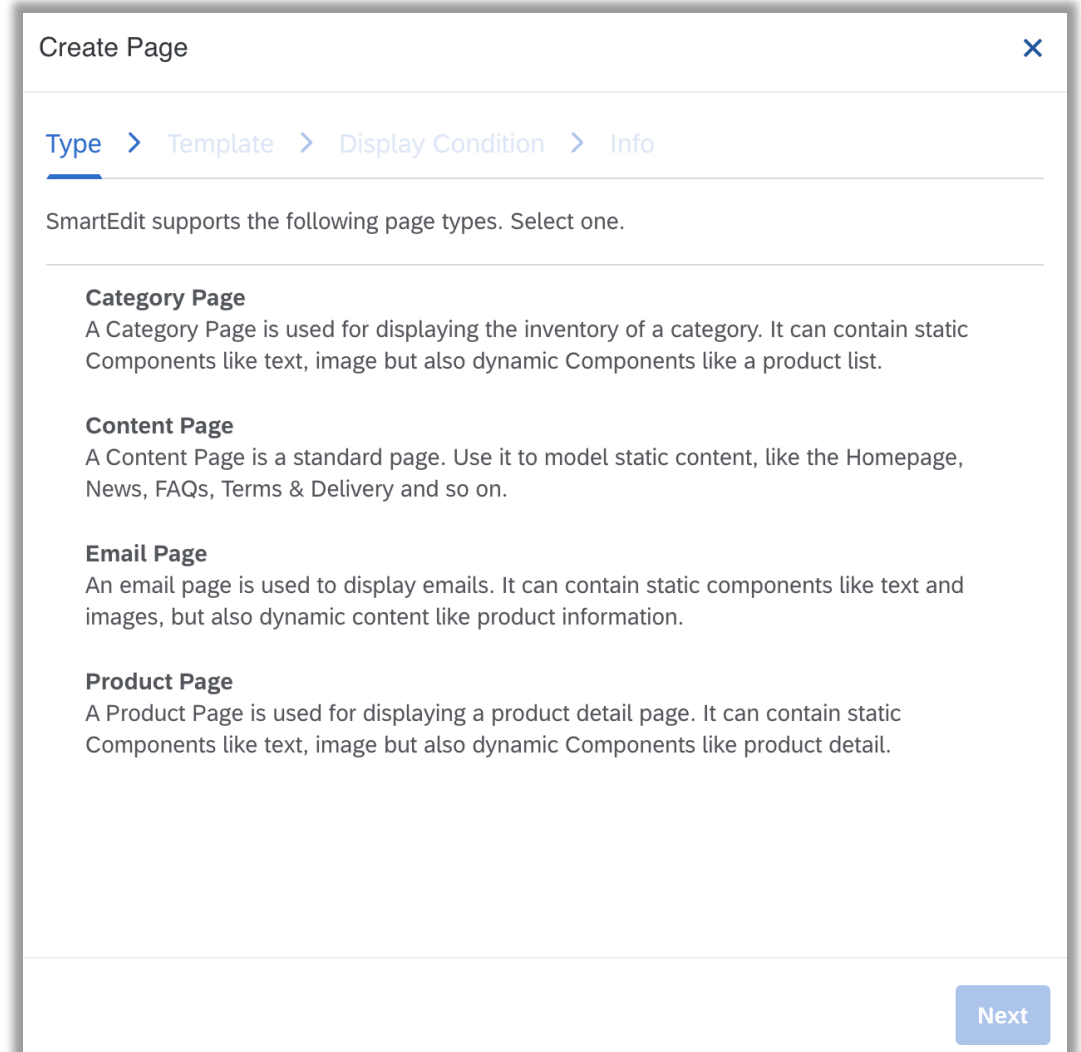
- Standard content page
- Example: Home page, help page, press releases and policies page

Email Page

- Used to display email messages
- Contains static and dynamic content

Product Page

- Display a single product and all product-related information
- Example: Product Details Page

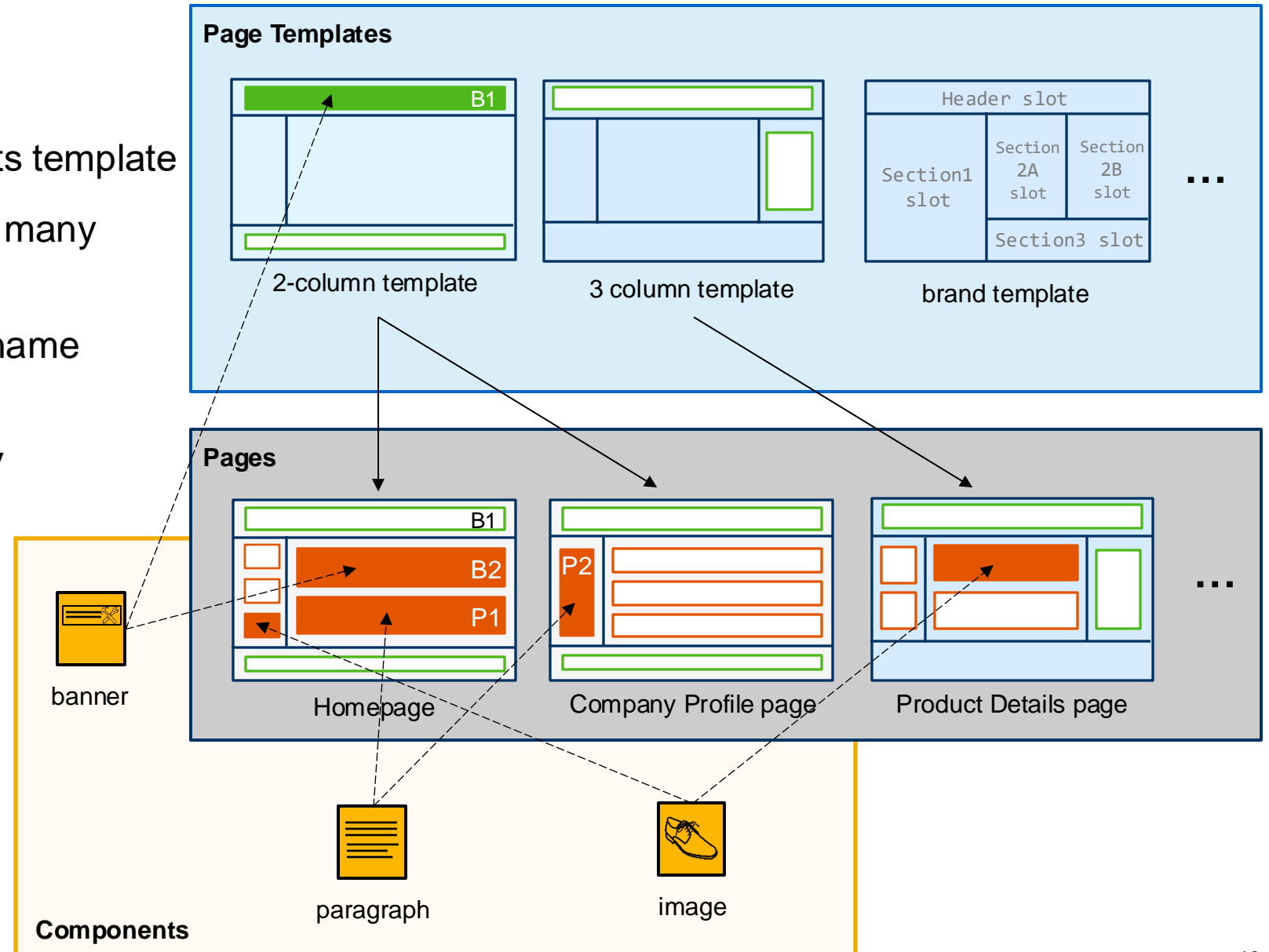


CMS Model

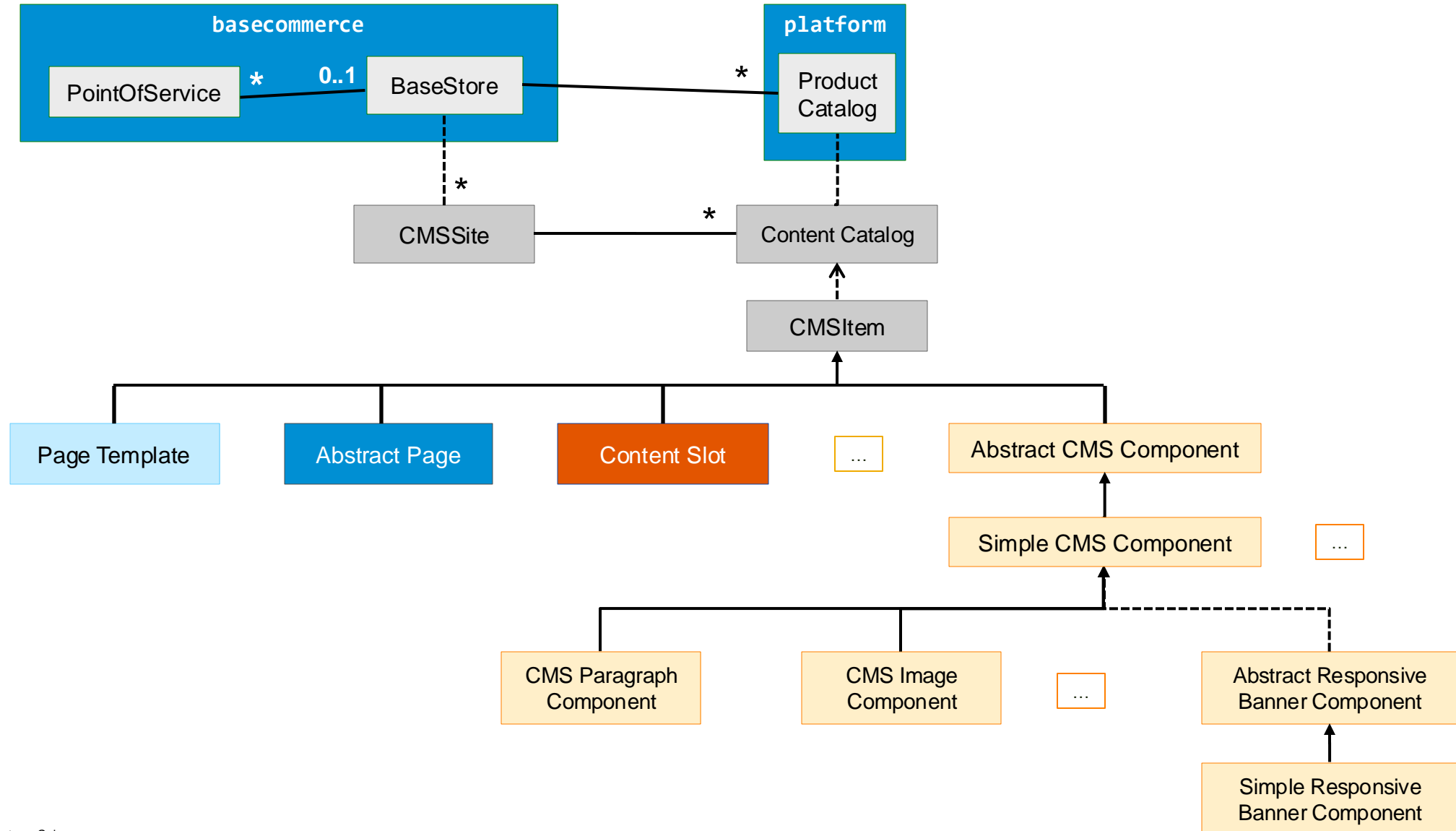


The WCMS Model

- A page's structure is defined by its template
- A template can be referenced by many pages
- A template defines position and name of content slots
- Each content slot is populated by components
 - Simple components (e.g. image, paragraph)
 - Complex components (e.g. banner, link, carousel)



Data Model



Slots and Components

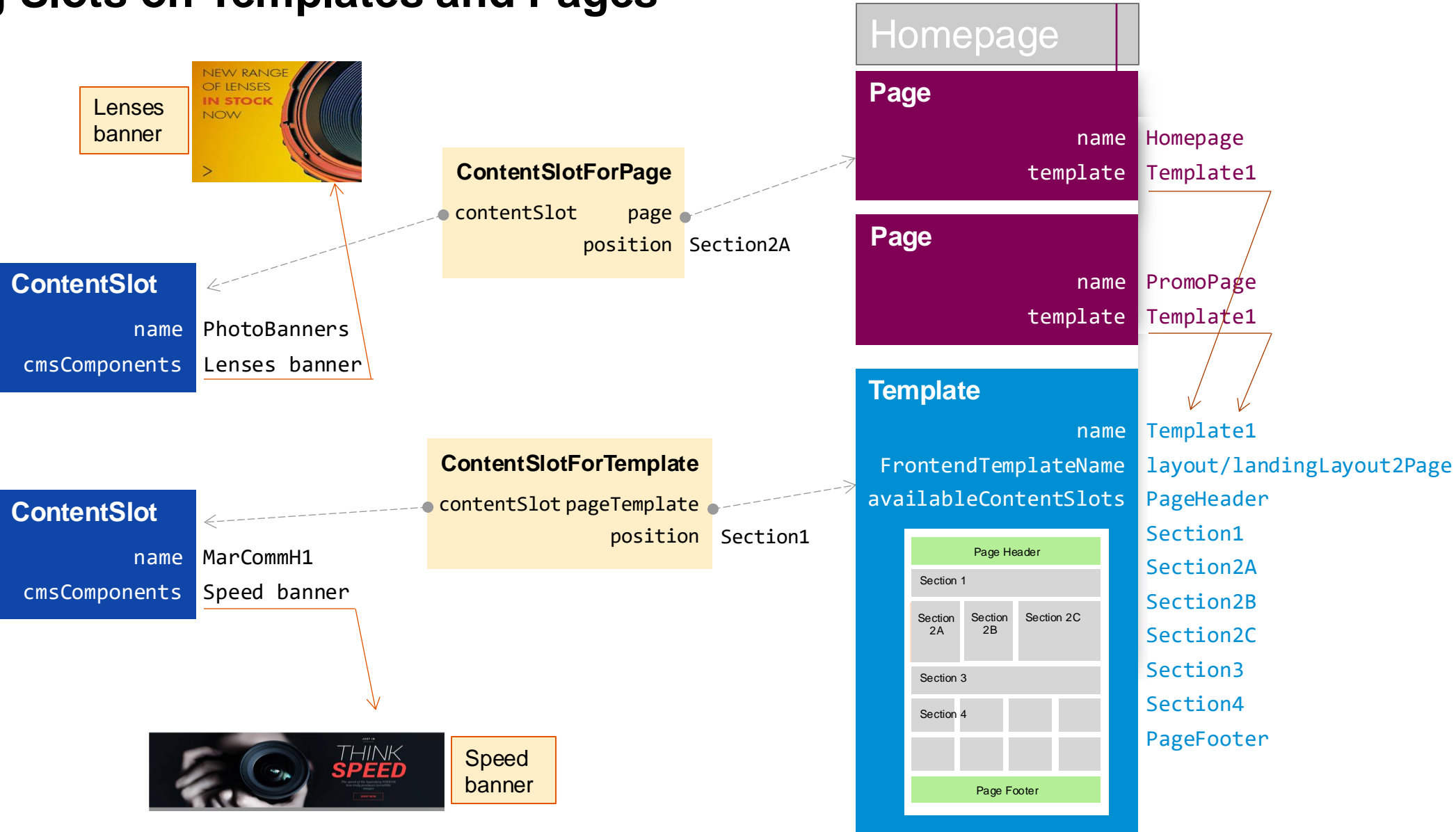


WCMS Model - Slots

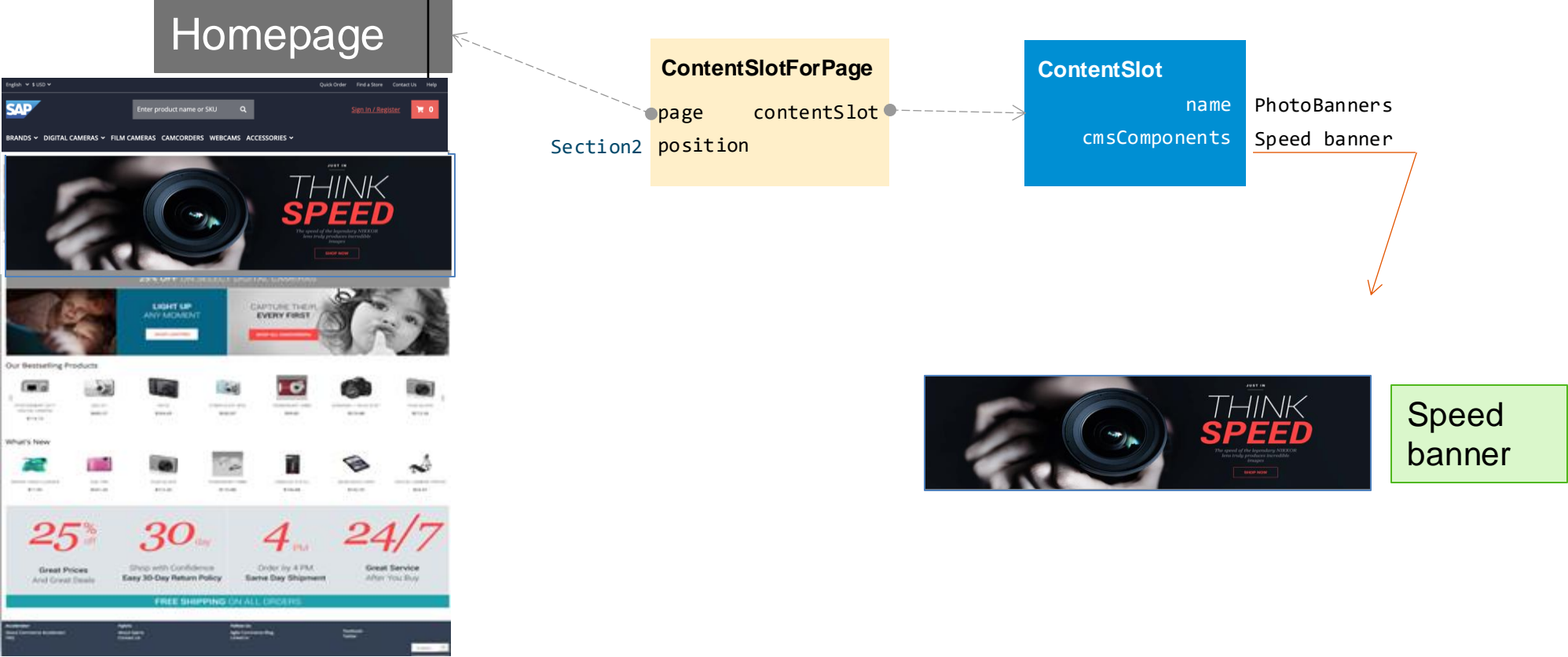
- Each template defines named slot locations, which are associated with ContentSlot items when templates and pages are defined
- Some slots are defined at the template level, and will appear on every page based on that template
 - Altering such a slot affects every page based on that template
- A ContentSlot instance, containing any number of components, can be associated with any slot location on any template
- Templates usually have more slots than most pages initially need, to allow for additional content to be added to a page over time, without additional development overhead



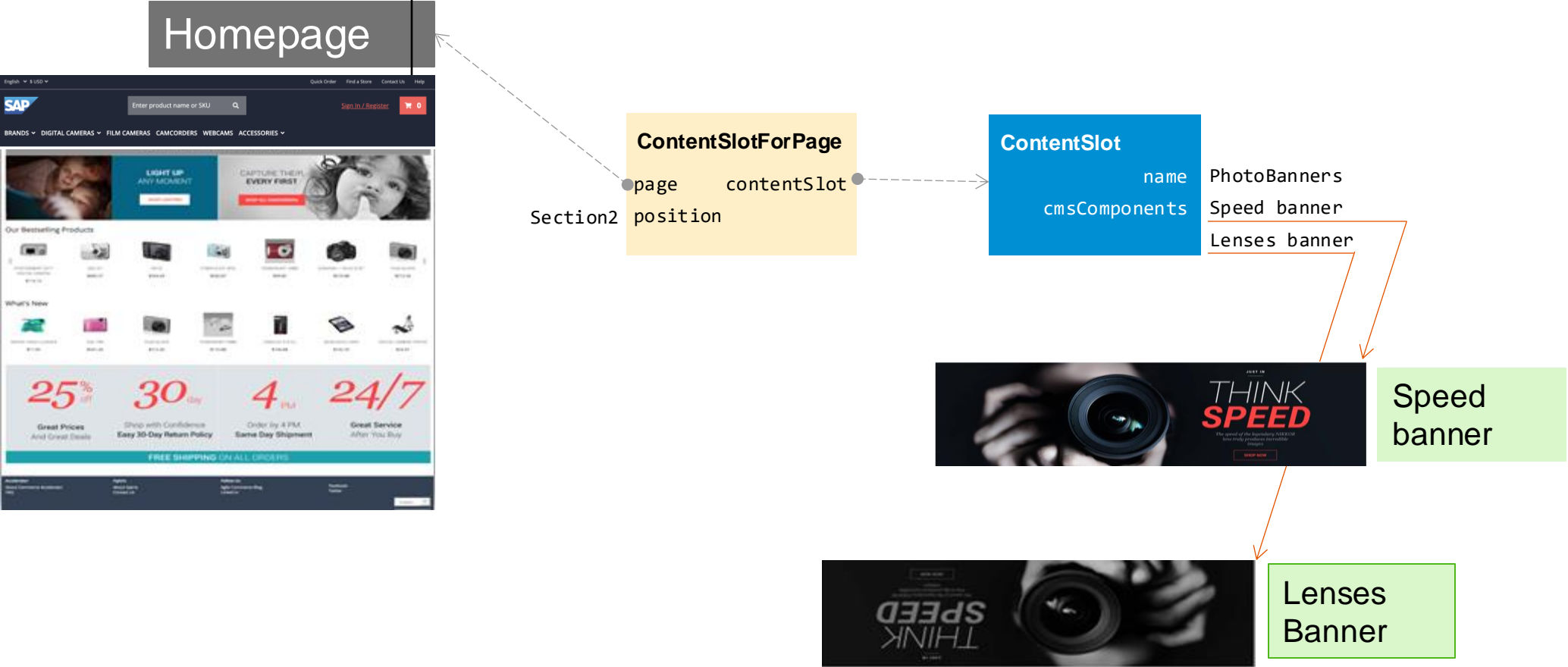
Defining Slots on Templates and Pages



Example: Banner Component



Example: 2 Banners



Multiple Banners

Homepage

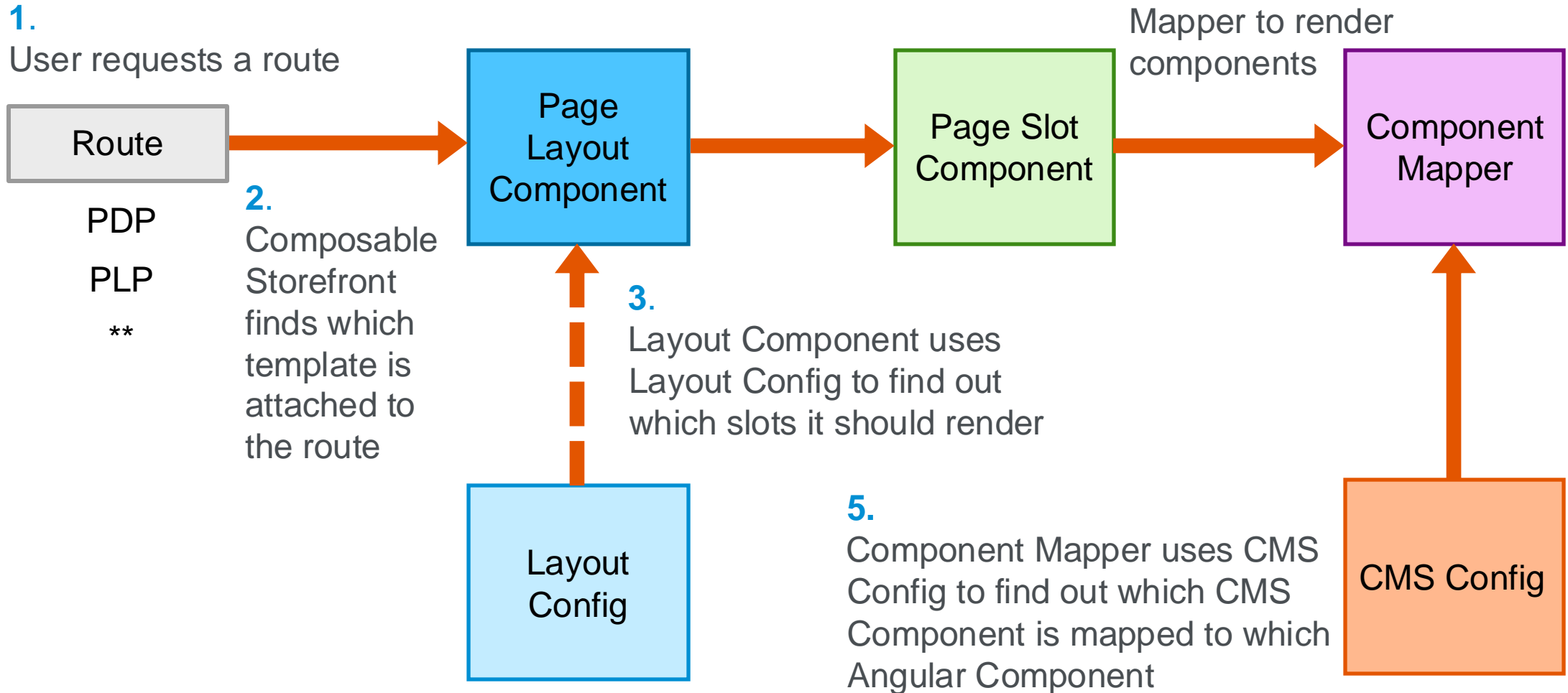


- The slot grows or shrinks depending on the number of components it contains
- Components in a slot may be arranged vertically or horizontally

View Rendering



CMS Rendering in Composable Storefront



Rendering Consideration



- The Composable Storefront page rendering process is **CMS-based**
 - The backend does not provide all the information required to render a page (e.g., slot order).
 - Part of the configuration is stored in the *Layout Configuration (LayoutConfig)*.
- **LayoutConfig** contains information about page templates and slot positions.
 - slots defines the position of content slots, which contain CMS Components.
 - Information about CMS Components comes from the backend.
- **CMS Components** are *mapped* to Angular components using **Component Mapping**
 - The mapping is provided in a configuration that can be customized
 - This allows you to configure a Custom Component to render a specific CMS Component

Creating an Angular Component

Component Metadata

```
@Component({
  selector: 'app-custom-product-carousel',
  templateUrl: './custom-product-carousel.component.html'
})
export class CustomProductCarouselComponent {

  data$: Observable<CmsProductCarouselComponent> = this.componentData.data$;

  constructor(private componentData: CmsComponentData<CmsProductCarouselComponent>) {
  }
}
```

Component Class to provide rendering logic



CmsProductCarouselComponent represents the concrete payload for the backend CMS component ProductCarouselComponent

```
<h3>This is a custom-product-carousel-component</h3>
<ng-container *ngIf="data$ | async as data">
  <pre> {{ data | json }}</pre>
</ng-container>
```

Component template to provide HTML view

CMS Component Data



- The backend CMS data for a CMS Component is provided by the [CmsComponentData](#) class
 - **CmsComponentData** contains CMS component **uid**, and **data\$** (Observable of **CmsComponent** payload)
 - With Angular Dependency Injection, CMS components and component-specific services can access **CmsComponentData**

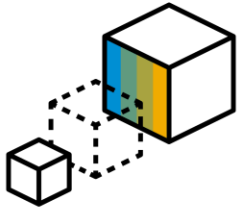
```
{
  "uid": "ElectronicsHomepageProductCarouselComponent",
  "uuid": "eyJpdGVtSWQiOiJFbGVjdHJvbmIjc0hvbWVwYWdlUHJvZHVjdENhcm91c2VsQ2...",
  "typeCode": "ProductCarouselComponent",
  "modifiedtime": "2023-01-26T20:44:47.884Z",
  "name": "Electronics Homepage Product Carousel",
  "container": "false",
  "popup": "false",
  "synchronizationBlocked": "false",
  "scroll": "ALLVISIBLE",
  "productCodes": "300938 358639 553637 816802 1934793 1382080 1981415 ...",
  "title": "Our Bestselling Products"
}
```


Accessing CMS Component Data in Angular Component

- Import the component class:
 - `import {CmsProductCarouselComponent} from '@spartacus/core';`
- Import `CmsComponentData`:
 - `import {CmsComponentData} from '@spartacus/storefront';`
- Then the CMS data can be **injected** into the Custom Component
 - **CmsComponentData** payload is specific to each CMS Component
 - **CmsComponentData** will be used to render CMS Component's view

```
export class CustomProductCarouselComponent {  
  data$: Observable<CmsProductCarouselComponent> = this.componentData.data$;  
  
  constructor(private componentData: CmsComponentData<CmsProductCarouselComponent>) {}  
}
```

Mapping an Angular Component to a CMS Component



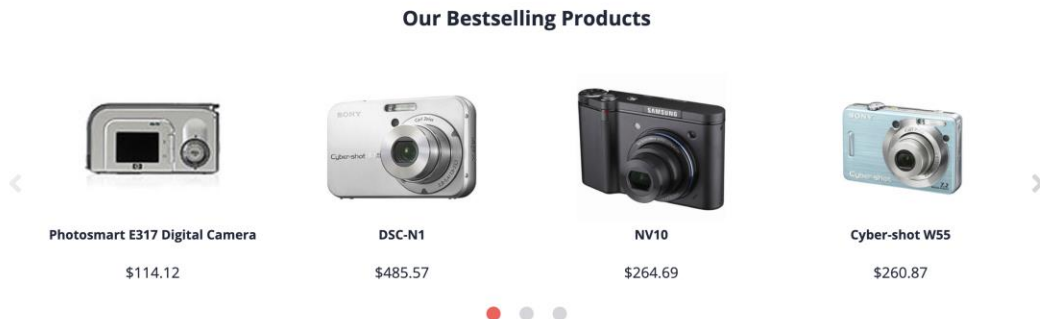
- Map the Angular Component to the CMS Component in the providers array
- Use component mapping (CmsConfig) to override the standard component implementation.

Type Code of
Backend CMS
Component

```
providers: [provideConfig(<CmsConfig>{  
  cmsComponents: {  
    ProductCarouselComponent: {  
      component: CustomProductCarouselComponent,  
    },  
  },  
})]
```

Class Name of
Frontend Angular
Component

your-component.module.ts



This is a custom-product-carousel-component

```
{  
  "uid": "ElectronicsHomepageProductCarouselComponent",  
  "uuid": "eyJpdGVtSWQI0iJFbGVjdHJvbmJlc0hvbWVwYWdlUHJvZHVjdENhcm91c2VsQ29tcG9uZW50IiwY2F0YXVxvZ0lkIjoj",  
  "typeCode": "ProductCarouselComponent",  
  "name": "Electronics Homepage Product Carousel",  
  "container": "false",  
  "popup": "false",  
  "synchronizationBlocked": "false",  
  "scroll": "ALLVISIBLE",  
  "productCodes": "300938 358639 553637 816802 1934793 1382080 1981415 816780 1934406 1986316 592506",  
  "title": "Our Bestselling Products",  
  "modifiedTime": "2023-01-26T20:44:47.884Z"  
}
```

Personalization



Personalization

Your website's content can be tailored to users in two ways:

1. CMS Restrictions

- include date/time, category, product, and usergroup restrictions ootb.
 - Can also be further customized to support more restrictions.
- Can be applied to either a CMS component or an entire page.

2. Segmentation based on the Personalization Module (covered in C4H341)

- Customers may be associated with one or more segments.
- Use SmartEdit to provide personalized CMS content to customers in different segments.
 - Or support commerce customization (for personalized search results and promotions)
- Key to integration with Big Data Service Providers.
 - E.g. SAP Emarsys can provides Campaign and Target Group Data to SAP Commerce Cloud to create and manage Segments automatically.

SmartEdit



SmartEdit

Intuitive, user-friendly UI for
creating laying out
managing
highly- personalized online
and mobile sites.

- A JavaScript overlay atop the customer view that facilitates editing page content in context
- Drag-and-drop components on pages
- Define and preview responsive design breakpoints
- Manage personalization in context
- SAML Single Sign-On
- Extensible Angular-based architecture to support needs of business users

Sites

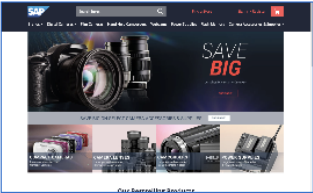
Site

Spartacus Electronics Site

Manage content catalogs for this site below.

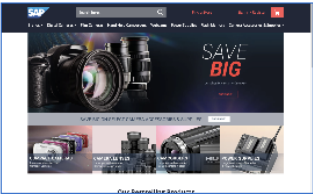
Content Catalogs

SPA Electronics Content Catalog



Online
Pages | Navigation | Homepage

Last synced from
Staged
4/12/23 2:17 PM



Staged
Pages | Navigation | Homepage

Last synced
4/12/23 2:17 PM



Pages

SPA Electronics Content Catalog - Staged

Pages (73)

Search page by name, ID, or Contentpage Label

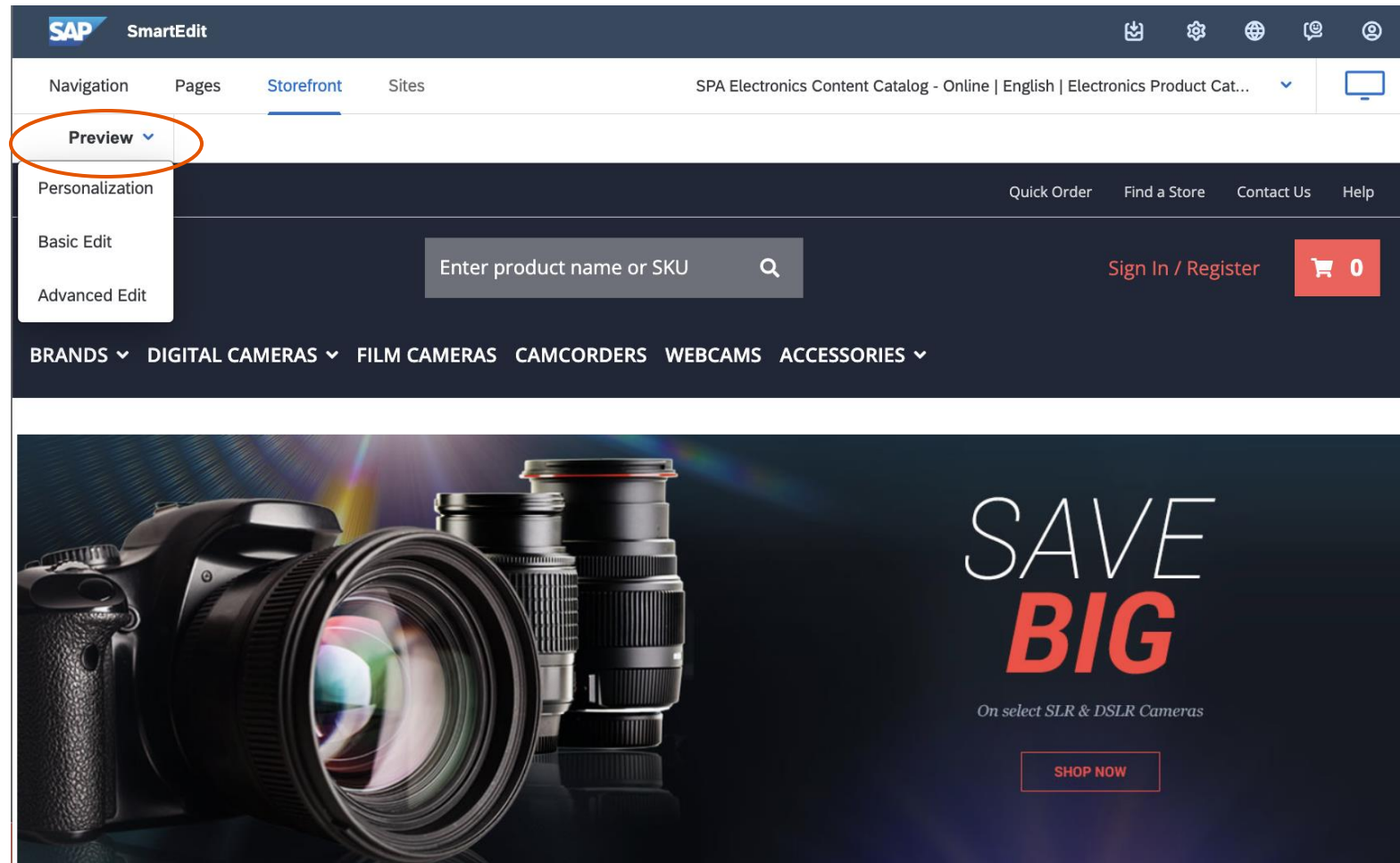
Create Page

Page Name	Page ID	Page Type	Page Label	Page Template	Restrictions	Status
Address Book Page	address-book	ContentPage	/my-account/address-book	AccountPageTemplate	-	● Synced
Cameras	Cameras	CategoryPage		ProductListPageTemplate	1	● Synced

Name	Title	Entry	More Actions
SiteRootNode	SiteRoot		...
Electronics Site	Electronics Site		...
My Account	My Account		...
Categories			...
All Brands	Brands	Brands Navigation Node Entry (CMSLinkComponent)	...
Digital Cameras Category	Digital Cameras	Digital Cameras Navigation Node Entry (CMSLinkComponent)	...
Digital Compacts Links		Digital Compacts Navigation Node Entry (CMSLinkComponent)	...
Digital SLR Links		Digital SLR Navigation Node Entry (CMSLinkComponent)	...
Film Cameras Category	Film Cameras	Film Cameras Navigation Node Entry (CMSLinkComponent)	...
Hand-held Camcorders Category	Camcorders	Handheld Camcorders Navigation Node Entry (CMSLinkComponent)	...
Webcams Category	Webcams	Webcams Navigation Node Entry (CMSLinkComponent)	...

Preview Mode

Displays the catalog version as it will be displayed to customers



You can use this mode to navigate to page you intend to edit

(instead of opening it from the page list)

Basic Edit Mode

Simple Responsive Banner Component Editor

Content Basic Info Visibility

Name*

Electronics Homepage Splash Banner Component

Media*

EN* JA DE ZH

Widescreen Desktop Tablet Mobile

Information Information Information Information

Replace Replace Replace Replace

Remove Remove Remove Remove

Link To ?

Url Link

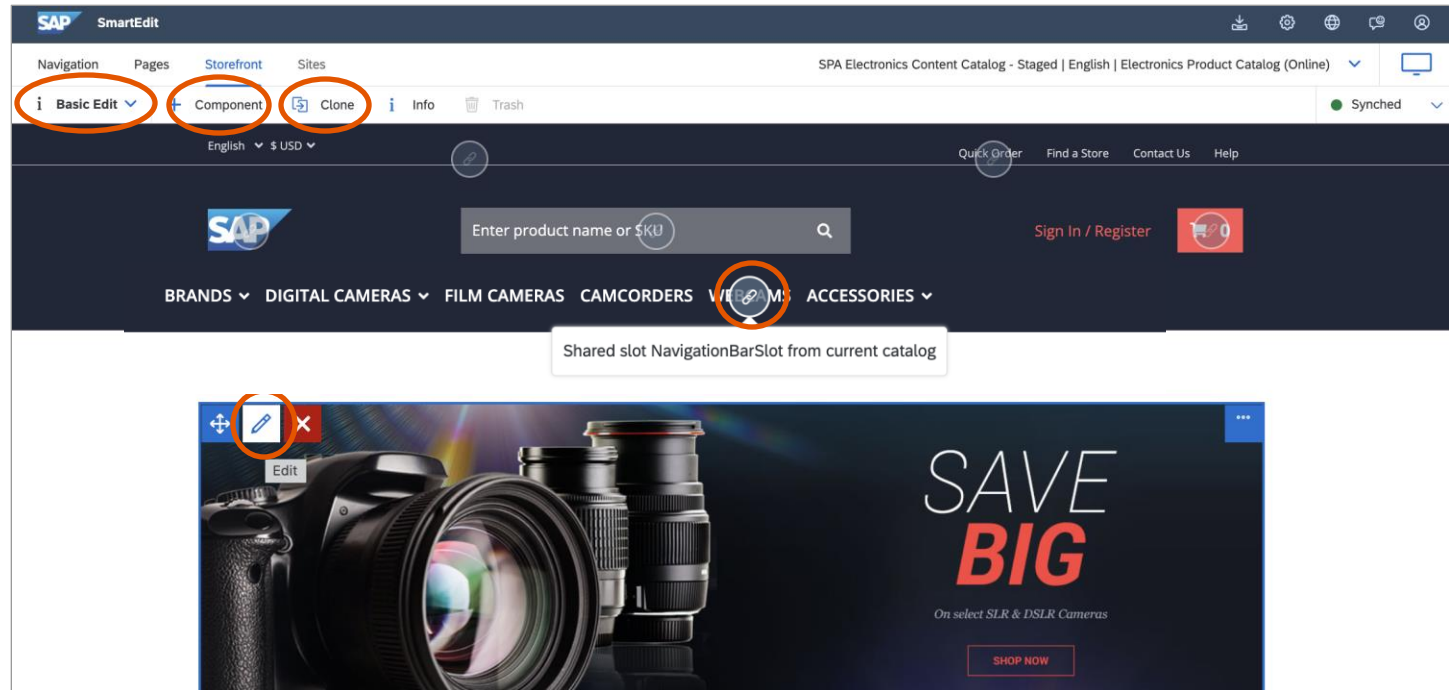
Url Link

/OpenCatalogue/Cameras/Digital-Cameras/Digital-SLR/c/578

Style Classes

Save Cancel

Allows you to manage components in **page**-scoped slots



- Add/edit components in page-scoped slots
- Provide images for defined responsive breakpoints
- Template-scoped components are not editable

Advanced Edit Mode

Manage components in both page- and **template**-scoped slots.

Synchronize page

Synchronize slot

Slot name

move | edit | delete

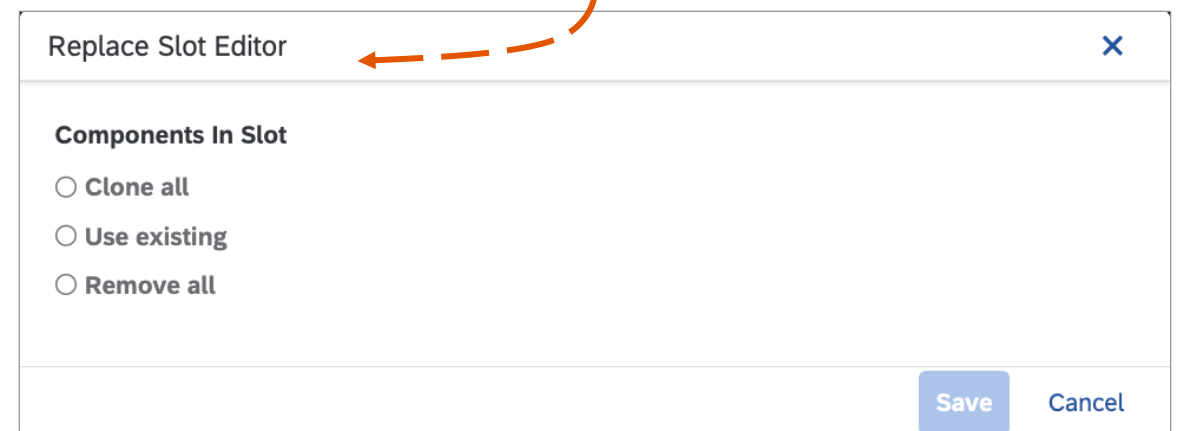
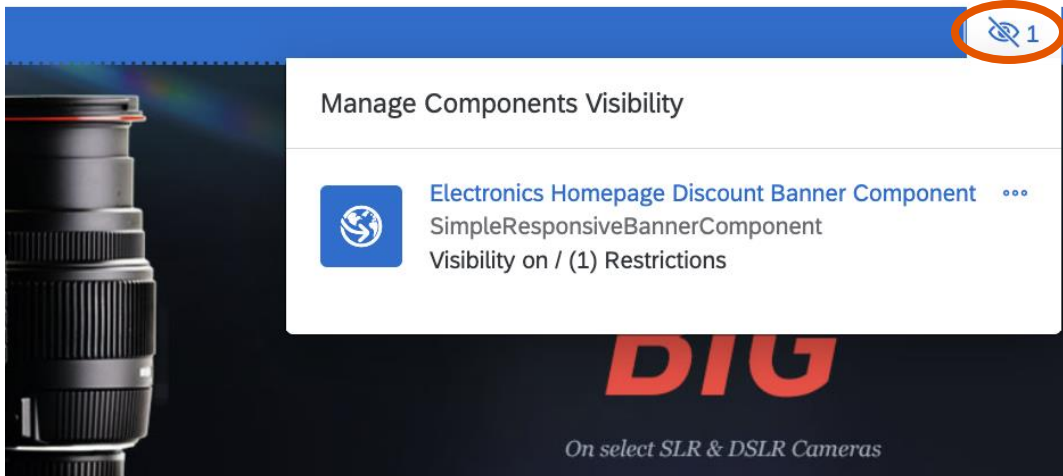
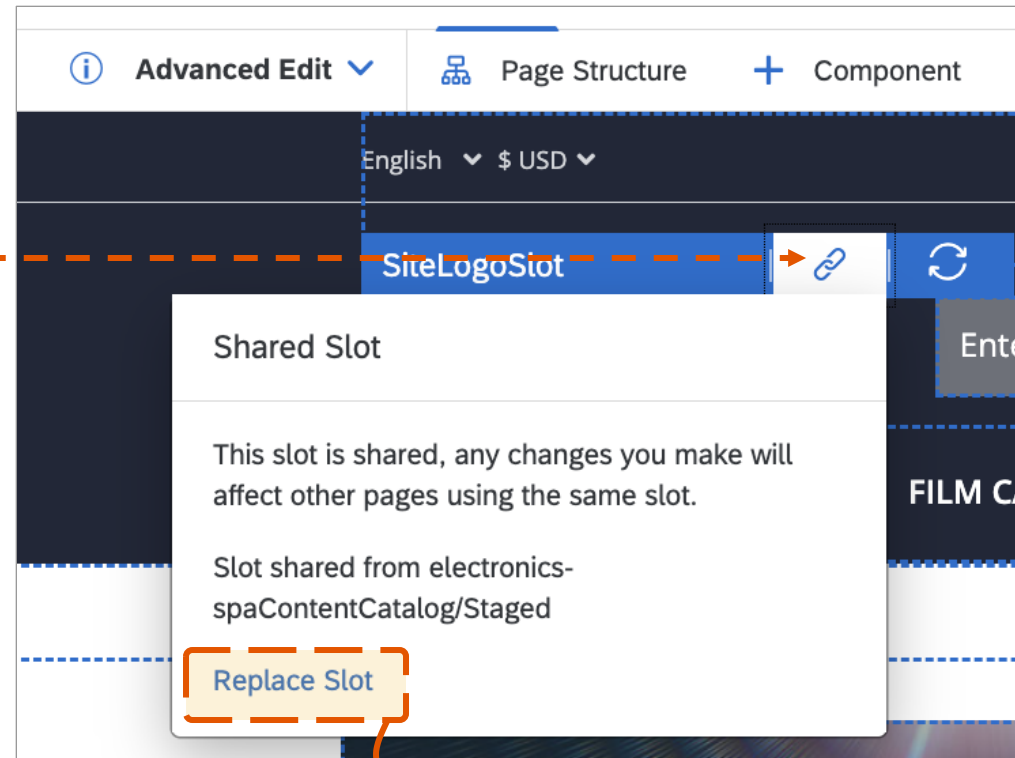
Shared slot
(template slot)

Slot needs synchronization

The screenshot shows the SAP SmartEdit Advanced Edit Mode interface. The top navigation bar includes 'Navigation', 'Pages', 'Storefront', and 'Sites'. The main toolbar contains 'Advanced Edit', 'Page Structure', 'Component', 'Sync', 'Versions', 'Clone', 'Info', and 'Trash'. The 'Sync' button is circled in orange. The main content area displays a storefront layout with a search bar labeled 'SearchBoxSlot' and a navigation menu. The 'SearchBoxSlot' is highlighted with a dashed orange box, and its name is shown in the top right. The 'move | edit | delete' icons are also visible. A 'Shared slot (template slot)' is indicated by a dashed orange box around the 'SearchBoxSlot'. A 'Slot needs synchronization' warning is shown in the bottom left corner, with a red circle around the warning icon. The main content area features a large banner image of a camera lens with the text 'SAVE BIG'.

Advanced Edit Mode

- Convert shared (template) slot to non-shared (page-scoped) slot and revert the conversion.
- Manage component visibility, Indicator in slot's header indicates the number of conditional components in the slot.



Advanced Edit Mode

With Page Structure, SmartEdit allows you to manage contents on a page in an easier way.

Open Page Structure

Manage content slots

Manage components

Page Structure

- SiteContextSlot
- SiteLinksSlot
- SiteLogoSlot
- SearchBoxSlot
- MiniCartSlot
- NavigationBarSlot
- HomepageBottomHeaderSlot
- Section1Slot-Homepage
 - Electronics Homepage Splash B...
SimpleResponsiveBannerCompon...
 - Electronics Homepage Discount ...
SimpleResponsiveBannerCompon...
- Section2ASlot-Homepage

Section1Slot-Homepage

SAVE BIG

On select SLR & DSLR Cameras

SHOP NOW

Manage nested component

Open Page Structure with current component highlighted

Personalization Mode

In the options menu,
select a variation

*(defined as part of a
customization)*

Each Variation refers to
a personalized page
which contains a set of
different CMS
components for e.g.
specific user segments
(canonLover)

The screenshot shows the SAP SmartEdit interface in Personalization Mode. The top navigation bar includes 'Navigation', 'Pages', 'Storefront', and 'Sites'. The main header displays 'SPA Electronics Content Catalog - Staged | English | Electronics Product Catalog (Online)'. The 'Personalization' dropdown menu is open, showing a list of customizations. The 'canonLover' variation is highlighted with a dashed orange border. The background shows a preview of a webpage with a camera image and the text 'TRAVEL LIGHT'.

Customization	Status
> SummerSale	Enabled
> CategoryLovers	Enabled
canonLover	Enabled
cameraLensesLover	Enabled
> VIPCust	Enabled

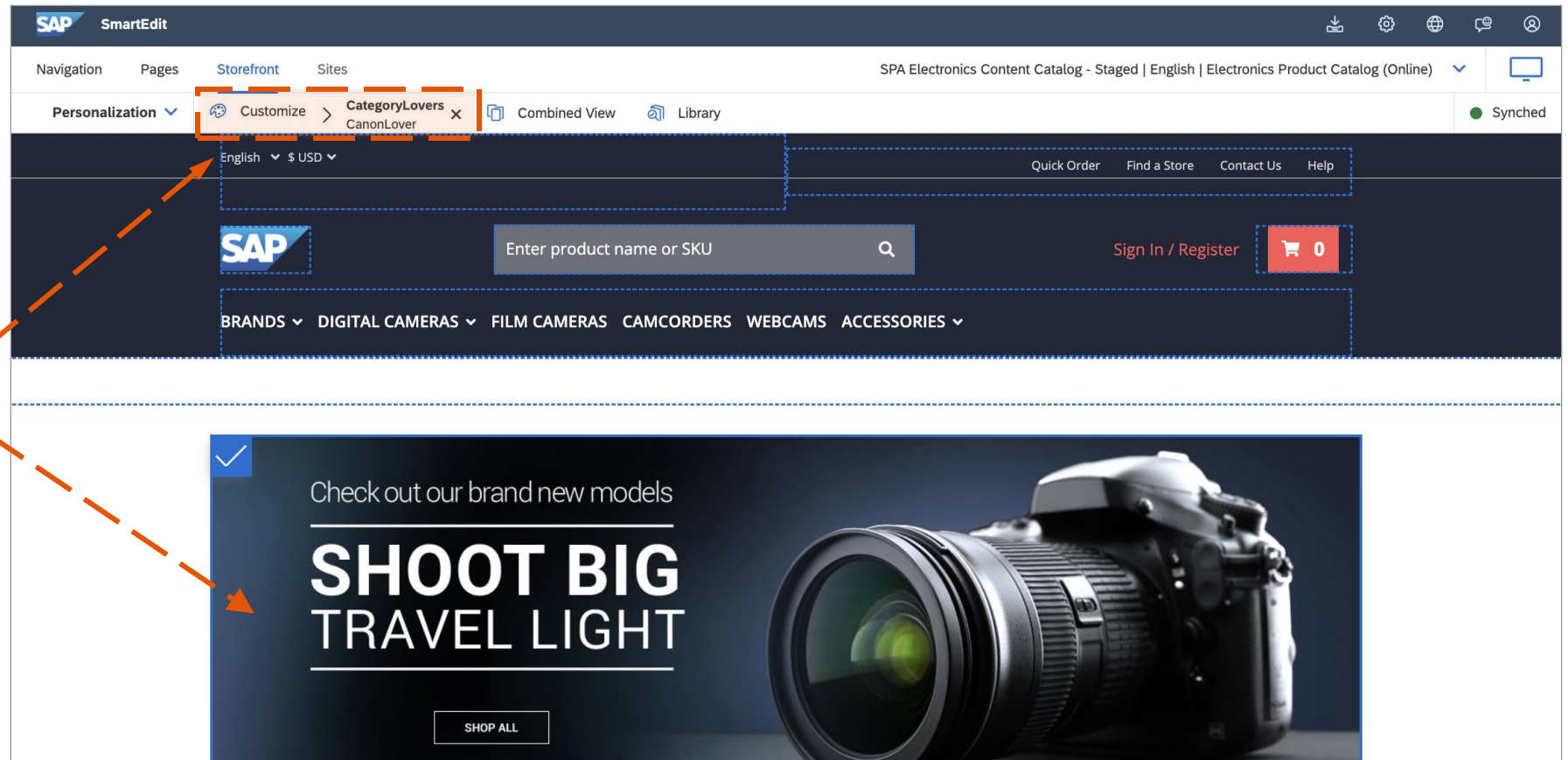
Personalization Mode

In the options menu,
select a variation

*(defined as part of a
customization)*

A Canon-specific product!

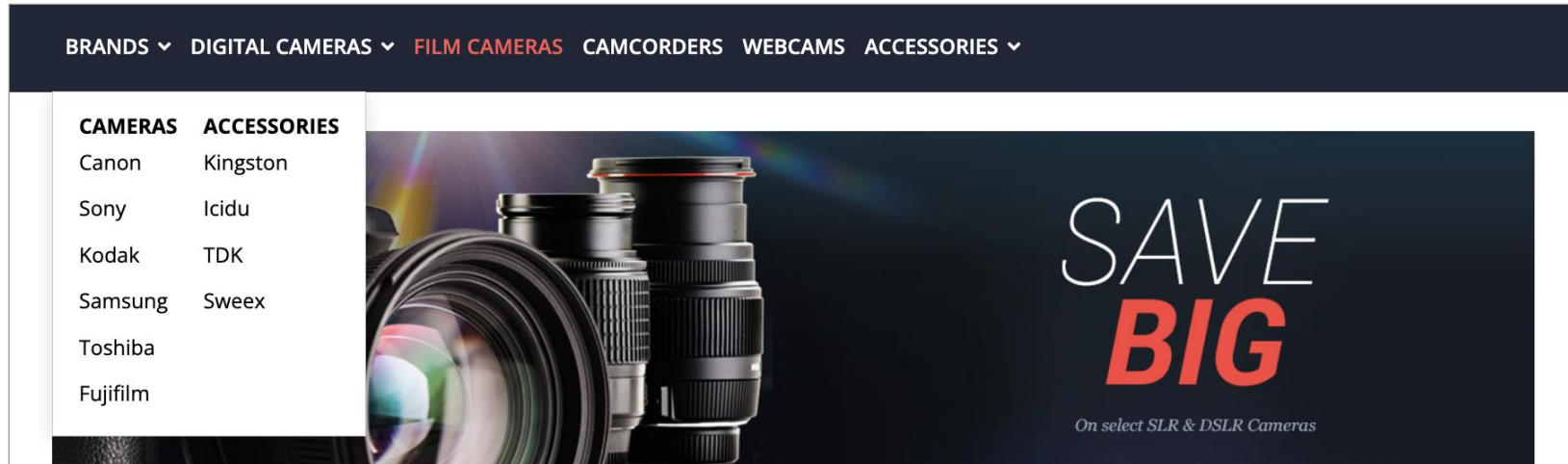
It's also possible to edit
the personalized content
for the selected variation.



Website Links



Link Overview

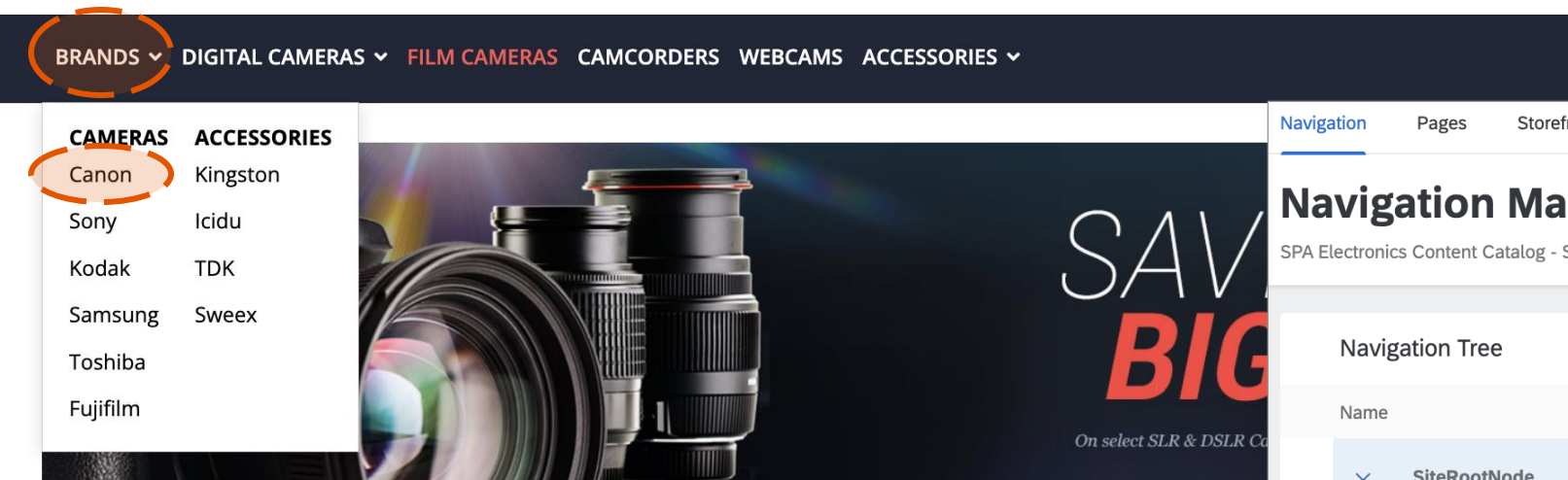


Page links are WCMS entities; they can be internal or external.

A portion of the Product Catalog structure is frequently duplicated as WCMS Links in a navigation component

- Usually defined at the template level, at the top of the page

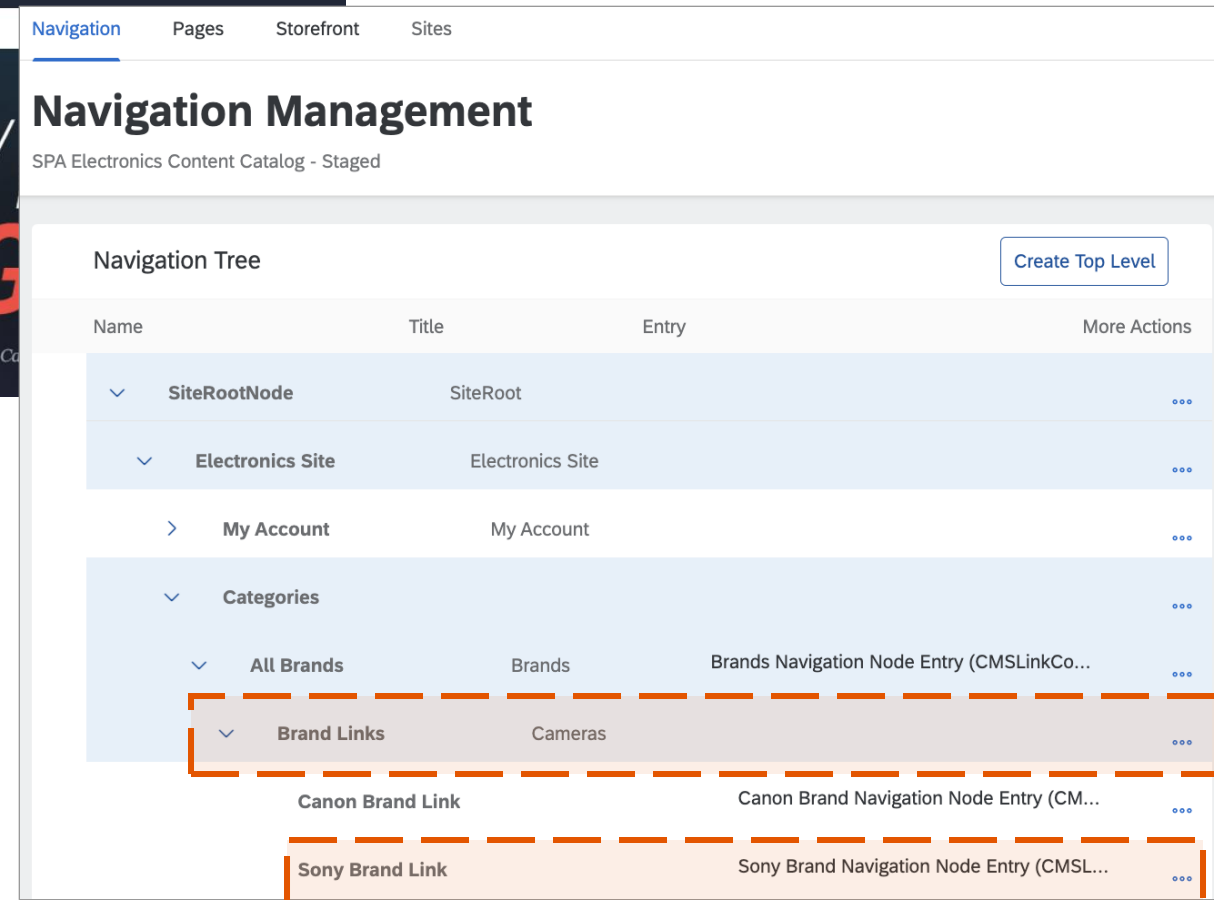
Defining Website Links



The Navigation bar consists of nested Nodes.

A node can be clicked if it contains a CMS Link Component, which contains the link target information.

- The CMSLinkComponent object can be created in either SmartEdit or Backoffice



Defining Website Links

SAP SmartEdit

Navigation Pages Storefront

Advanced Edit

English \$ USD

SAP NavigationBarSlot

↕

✎

✕

TAL CAMERAS

Edit

Category Navigation Component Editor

Content Basic Info Visibility

Name*

Category Navigation Component

Navigation Node

ROOT SiteRootNode > LEVEL 1 Electronics Site > LEVEL 2 Categories

Replace

Name	Title	Entry
> All Brands	Brands	Brands Navigation ...
> Digital Cameras Category	Digital Came	Digital Cameras N...

SAP SmartEdit

Navigation Pages Storefront Sites

Navigation Management

SPA Electronics Content Catalog - Staged

Navigation Tree

Create Top Level

Name	Title	Entry	More Actions
1 SiteRootNode	SiteRoot		...
2 Electronics Site	Electronics Site		...
> My Account	My Account		...
3 Categories			...
> All Brands	Brands	Brands Navigation Node Entry (C...	...
> Digital Cameras ...	Digital Camer	Digital Cameras Navigation Node
Film Cameras C...	Film Cameras	Film Cameras Navigation Node En...	...
Hand-held Camc...	Camcorders	Handheld Camcorders Navigation

Demo



In Thousand USD, %

2,987.26 (+30.59%)
Product Revenue Won Current

In Thousand USD
444.48
Revenue New Products



Top Customers

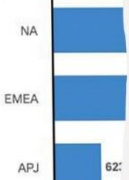


Dimension

- ☐ Industry ID
- ☐ Territory
- ☐ Sales Unit
- ☒ Country
- ☐ Competitor

Measure

In Thousand USD



Key Points

1. The important data models in WCMS include pageTemplates, pages, slots and components
2. All WCMS-related components are **catalog-aware**.
3. Every page is based on exactly one pageTemplate
4. Page Templates define the structure via named slot locations
5. Slot locations are associated with content slots, which contain any number of CMS Components
6. Content slots are attached to slot locations on individual pages or a pageTemplate (changes here affect all pages based on that template)
7. Every OOTB CMS component has a matching OOTB Angular component on the Composable Storefront.
8. You can create custom Angular Components to replace or extend the OOTB components, or to map to custom CMS components
9. SmartEdit is used to manage CMS data in SAP Commerce Cloud, with
 - dedicated modes (preview, basic edit, advanced edit, etc.) for different purposes
 - navigation view to manage web site links

WCMS Exercise



Thank you.