

Sitecore 10 Certification Crash-Course



STONEMAN TO SUPERMAN



60 minutes

- NAVAN



Table of Contents

Master vs Web DB

Data Templates

Template Inheritance

Insert options

Standard Values

Field Editor Button Use

Presentation Details

- Layout

- Controls

- Placeholder Settings

Dynamic Placeholder

Static vs Dynamic Component Binding

Sitecore MVC Concepts

- Layout

- View Rendering

Sitecore Rendering properties

Visual Studio Publish vs Sitecore Publish

Versioned Layouts

Site Definition Patch Config

Sharing Content between Sites

Config File Load Order

Layers.config

Sitecore Support Package

Sitecore Security Domains

Sitecore.IdentityServer.Host.xml

User Manager

Role Manager

Security - Users/Groups

Workflow

Elements of Workflow

Workbox

Sitecore ASP.NET Core Rendering Host

Sitecore Headless Service

Sitecore ASP.NET Rendering SDK

Startup.cs functionality

Service Registration in Startup.cs

Sitecore CLI Plugin Setup

Sitecore CLI - sitecore.json

Sitecore CLI - module.json

Sitecore CLI vs Sitecore for Visual Studio

Sitecore Docker Lifecycle

.env file

General Tips


Sitecore Exam Competencies

- 50 questions, 2 hrs
- Online-proctored; need a separate camera, laptop camera won't suffice
- Mark for review option
- Check for Sitecore Partner coupon with employer for exam discount

<https://navansitecorenotes.blogspot.com/2022/02/sitecore-10-exam-competencies.html>

SIA Installation Errors

Missing Certificates Folder:

 **SITECORE** Install Sitecore Experience Platform 9.3

Welcome

Prerequisites

Install Solr

Sitecore settings

SQL Server settings

Solr settings

Optional modules

Summary

Validate

Install

Install

Installing Sitecore with the settings you defined.

The installation failed

An error has occurred. For more information, see the log file.

Installation Status [Log](#) ^

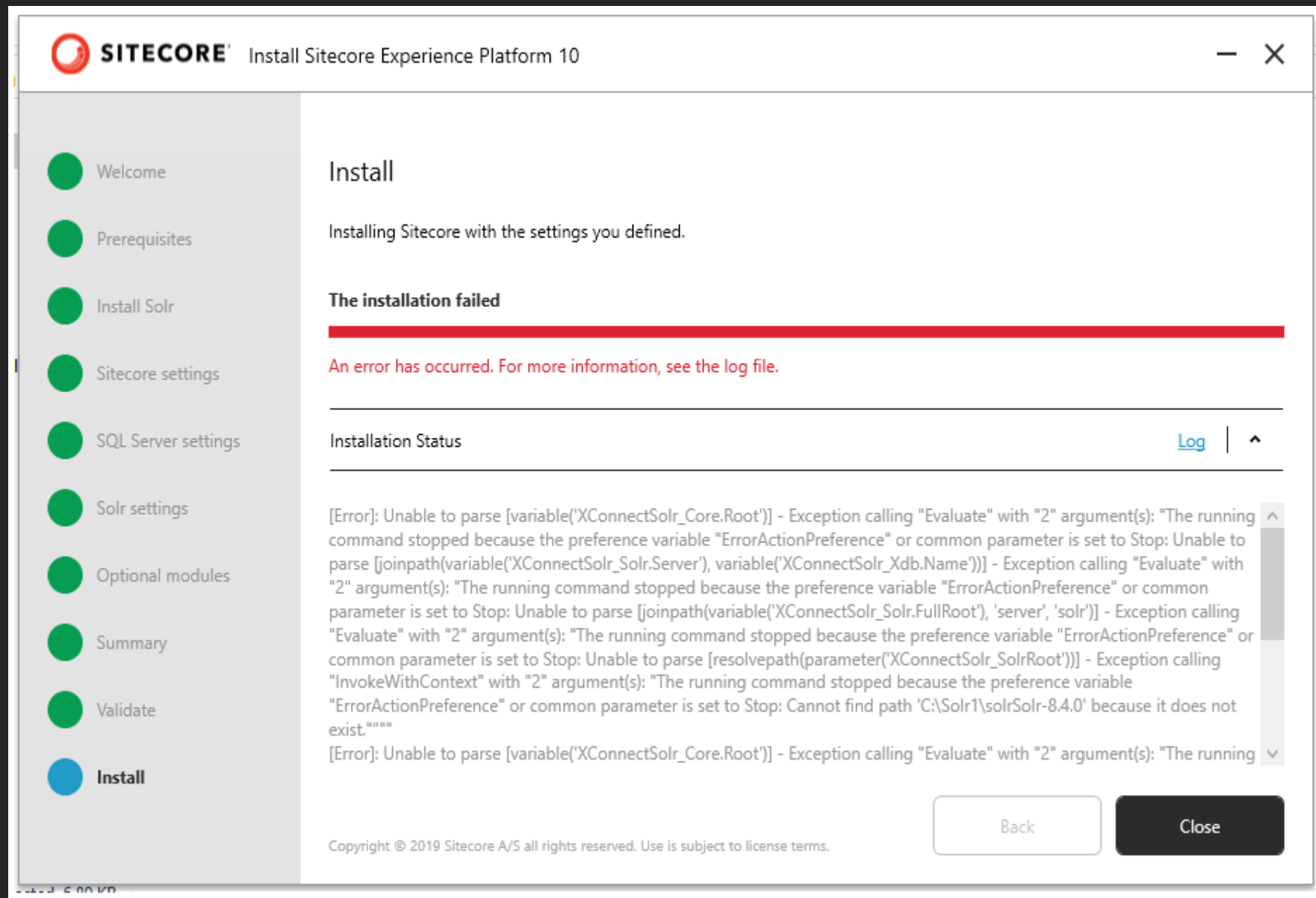
[Error]: Cannot retrieve the dynamic parameters for the cmdlet. Cannot validate argument on parameter 'Source'. The "Test-Path \$-Type Leaf" validation script for the argument with value ".\createcert.json" did not return a result of True. Determine why the validation script failed, and then try the command again.

BackClose

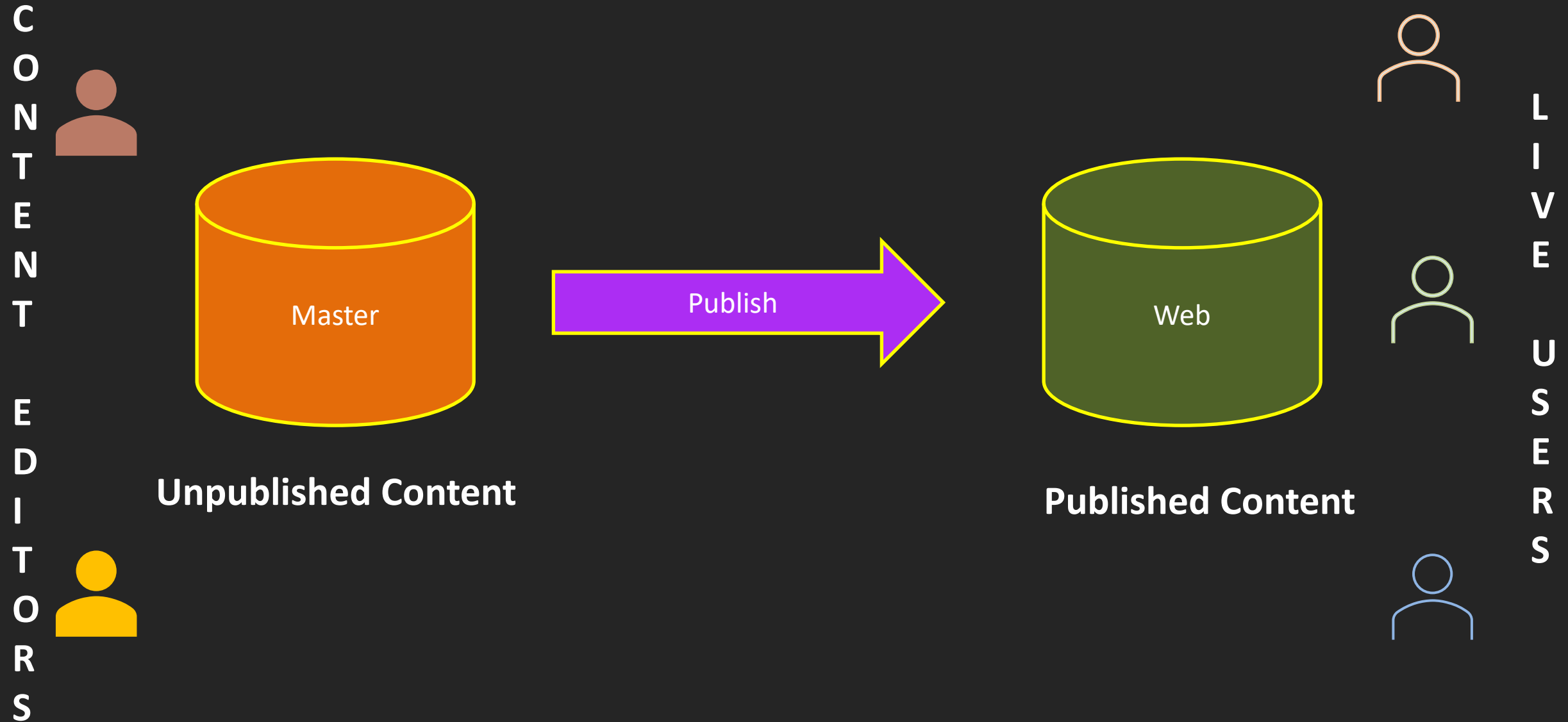
Copyright © 2019 Sitecore A/S all rights reserved. Use is subject to license terms.

SIA Installation Errors

Non-existent Solr Path:



Master vs Web DB



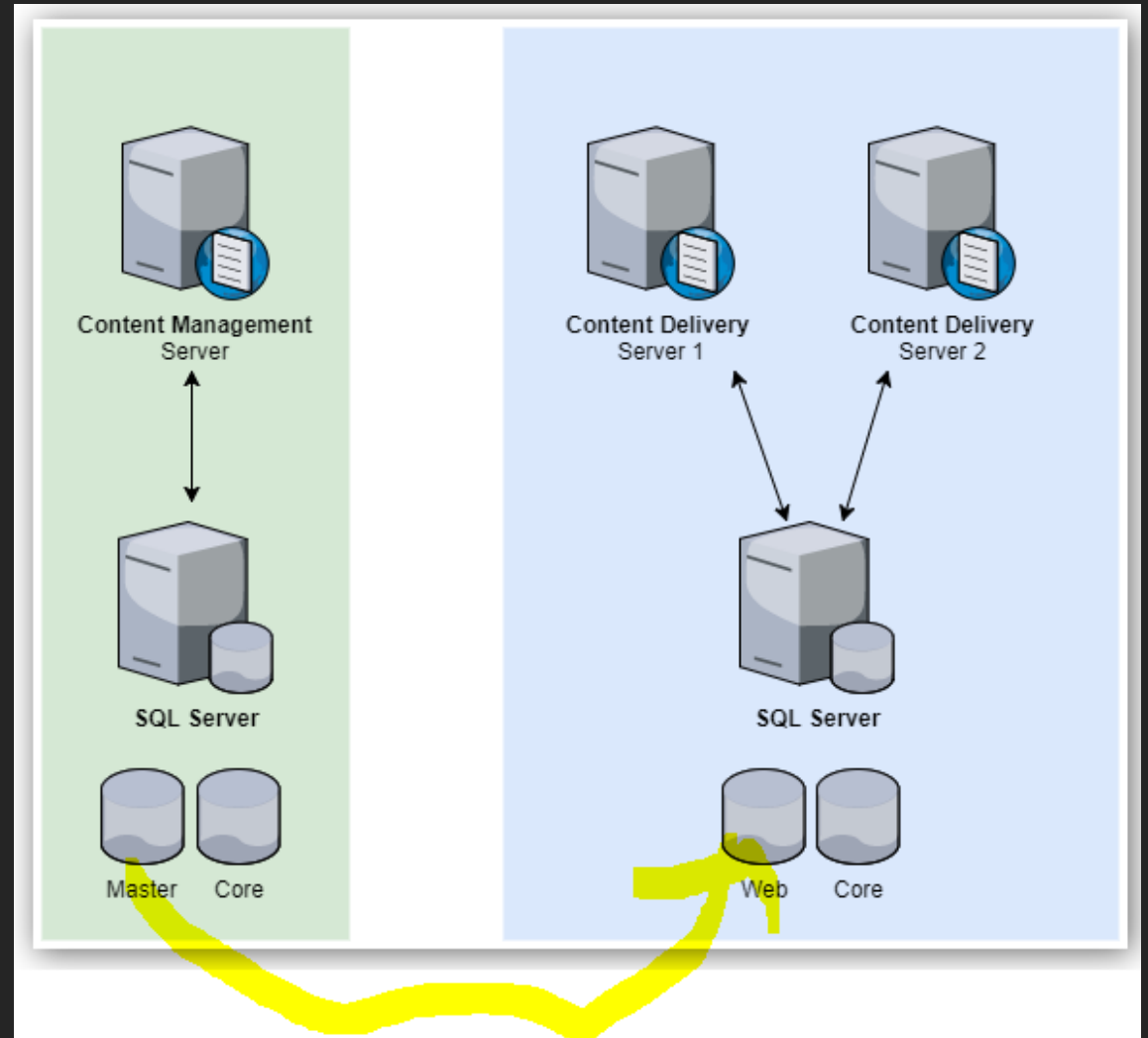
Master vs Web DB

Master DB to edit content while Web DB is used to store live data

Content changes made in Web DB will be overwritten by next publish so, always changes must go from master -> web

Publishing Wizard:

Content Editor
Experience Editor
Sitecore Desktop



Templates or Data templates

Nothing but data structures

The screenshot displays the Sitecore Builder interface for configuring a data template. The left sidebar shows a content tree with the 'Profile' item selected under the 'Person' folder. The main workspace shows the 'Profile' section, which is highlighted with an orange box and labeled 'Field Section for grouping related fields'. Below this, a table lists the fields and their configurations:

Name	Type	Source	Unversioned	Shared
First Name	Single-Line Text		<input type="checkbox"/>	<input type="checkbox"/>
Last Name	Single-Line Text		<input type="checkbox"/>	<input type="checkbox"/>
Email	Single-Line Text		<input type="checkbox"/>	<input type="checkbox"/>
Introduction	Rich Text		<input type="checkbox"/>	<input type="checkbox"/>
Country	Droptree	/sitecore/content/MvpSite/MVP Repository/Countries	<input type="checkbox"/>	<input type="checkbox"/>
Awards	TreelistEx	/sitecore/content/MvpSite/MVP Repository/Award Years	<input type="checkbox"/>	<input type="checkbox"/>
Add a new field	Single-Line Text		<input type="checkbox"/>	<input type="checkbox"/>
Application				

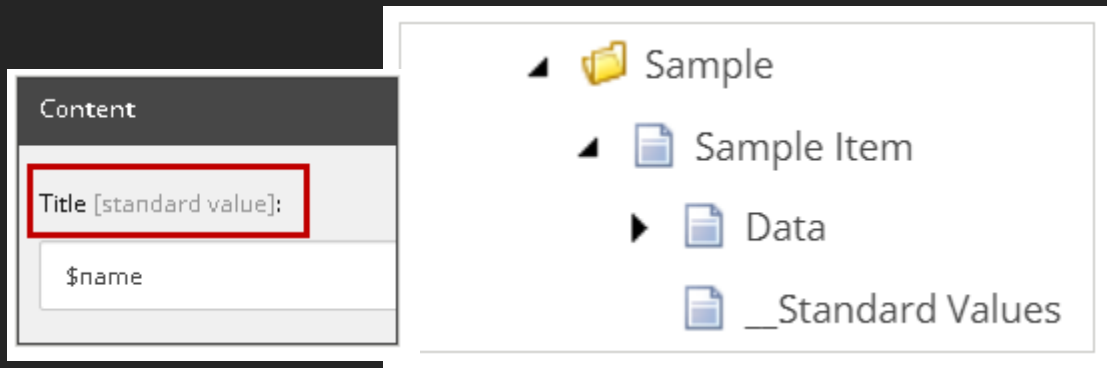
The 'Country' field's source is highlighted with a red box and labeled 'Where to get data from Content tree'.

When you select types like multilist / droptree / treelist, you must set the source too

Standard Values

- Default value for field
- Fallback value when field value is null

Good practice to setup Presentation Details since it standardizes content creation process



Tokens:

\$name: The name of the item

\$id: The ID of the item

\$parentid: The ID of the parent item

\$parentname: The name of the parent item

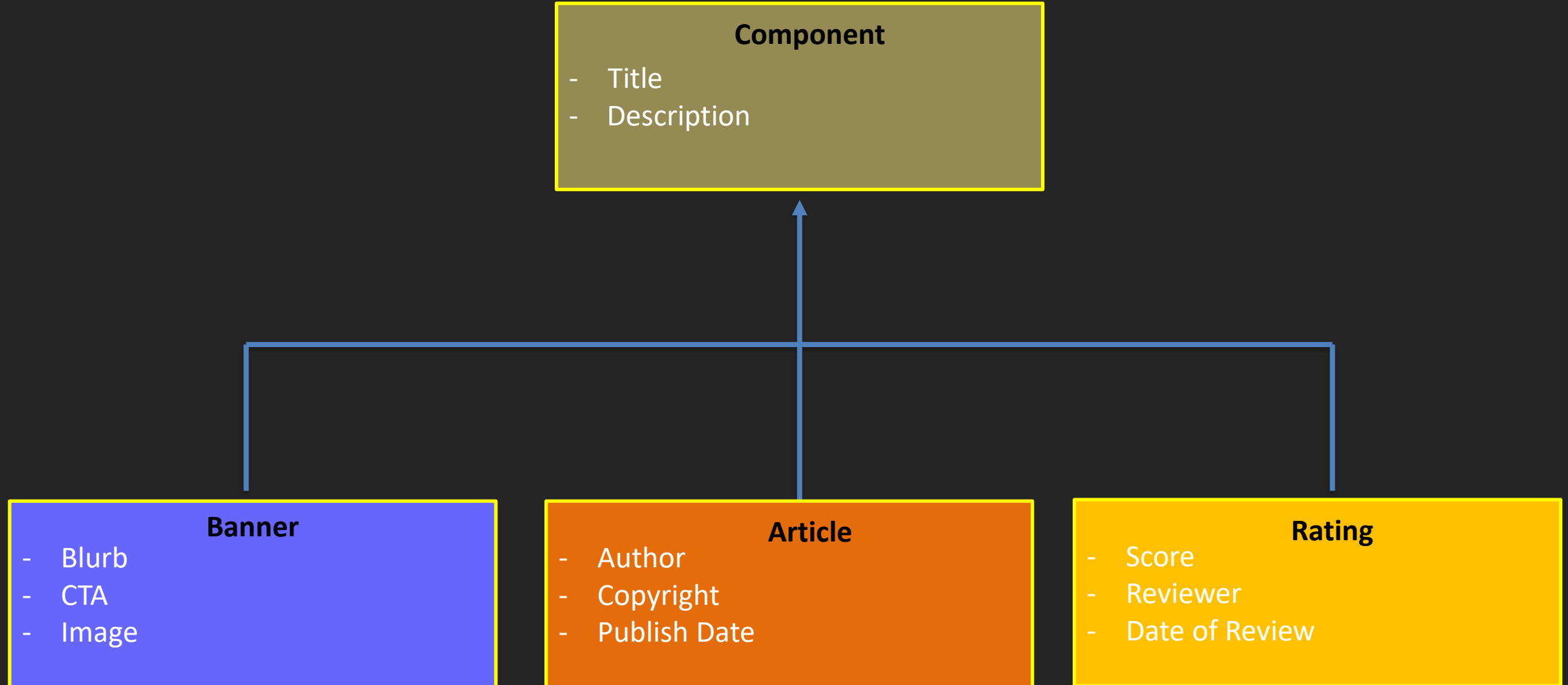
\$date: The system date (yyyyMMdd)

\$time: The system time (HHmmss)

\$now: The date and time (yyyyMMddTHHmmss)

<https://doc.sitecore.com/xp/en/developers/100/sitecore-experience-manager/standard-values.html>

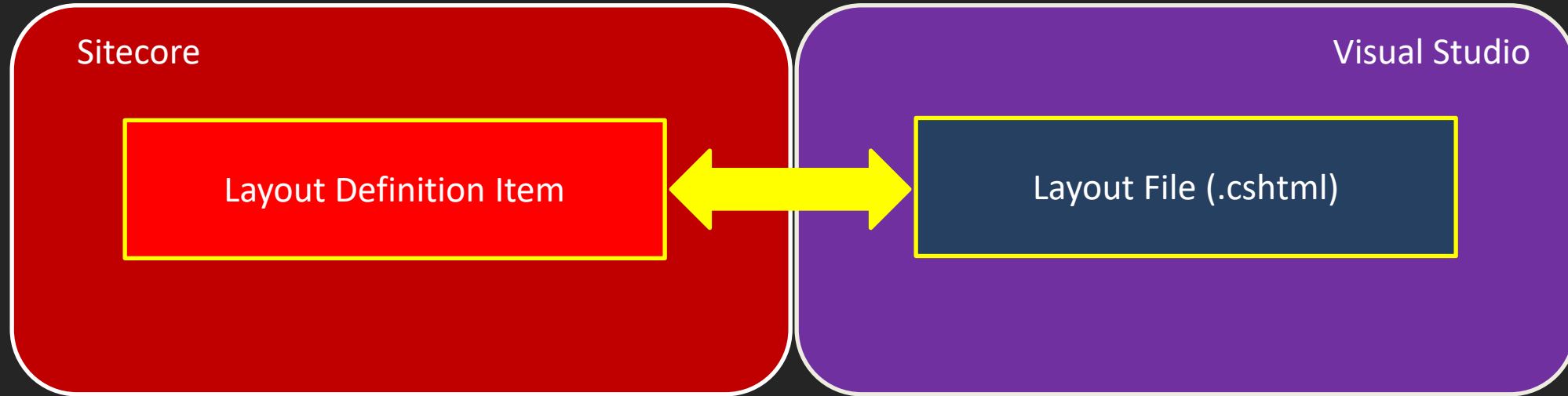
Template Inheritance



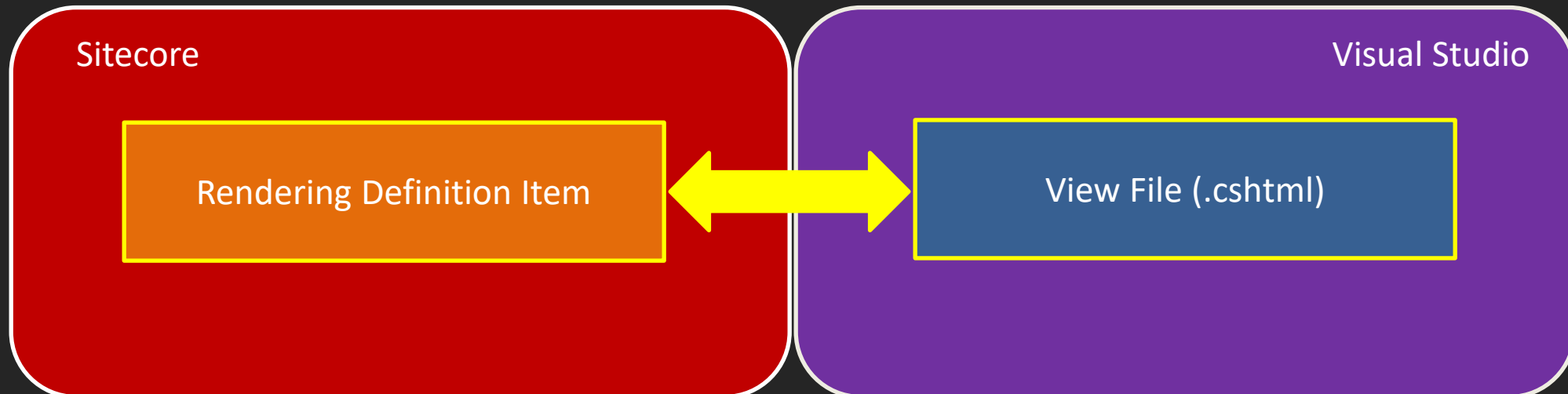
Note: If a field with same name across two templates points to two different source, then it is not a candidate for inheritance

Sitecore MVC Concepts

Layout:



View Rendering:



Sitecore MVC Concepts

Layout Details

Layout Definition

Placeholder Definition

Rendering
Definition

Data
Source

Placeholder Definition

Rendering
Definition

Data
Source

Placeholder Definition

Rendering
Definition

Data
Source

Layout (.cshtml)

Placeholder

Placeholder

Placeholder

Sitecore Context

Rendering (.cshtml)

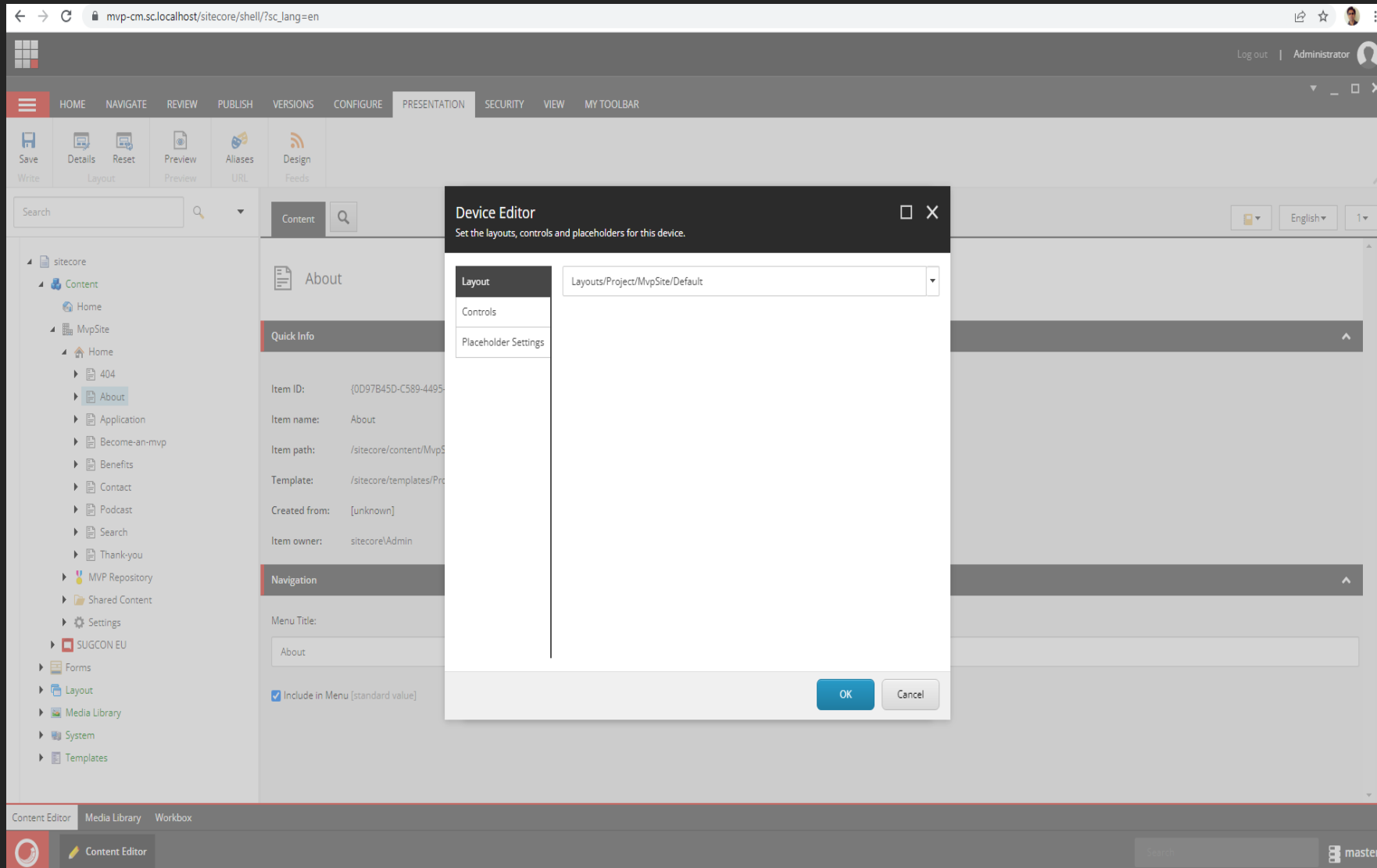
Template Field

Template Field

Template Field

Template Field

Presentation Details

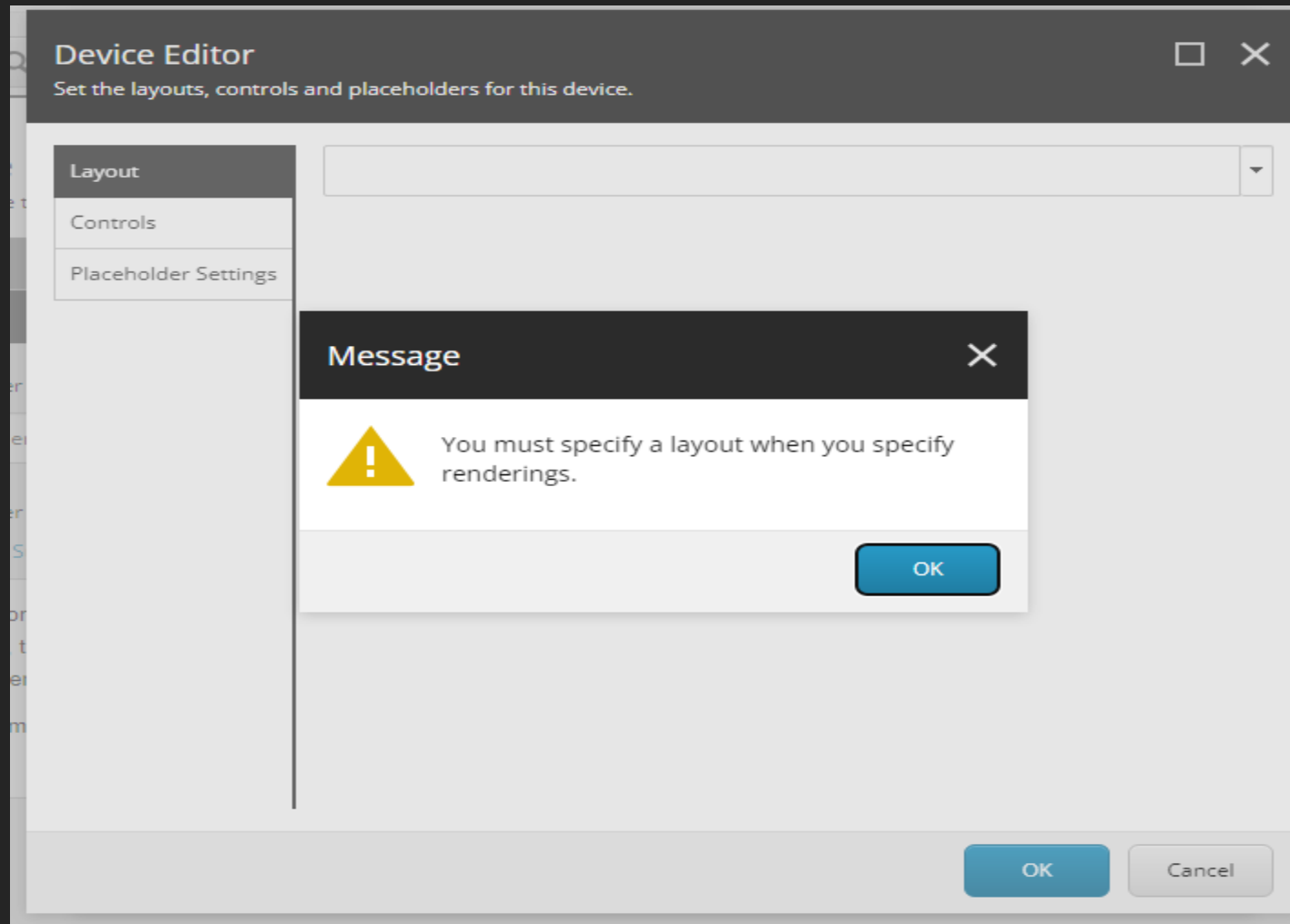


For page to be visible,
the following are important:

- Layout
- Controls

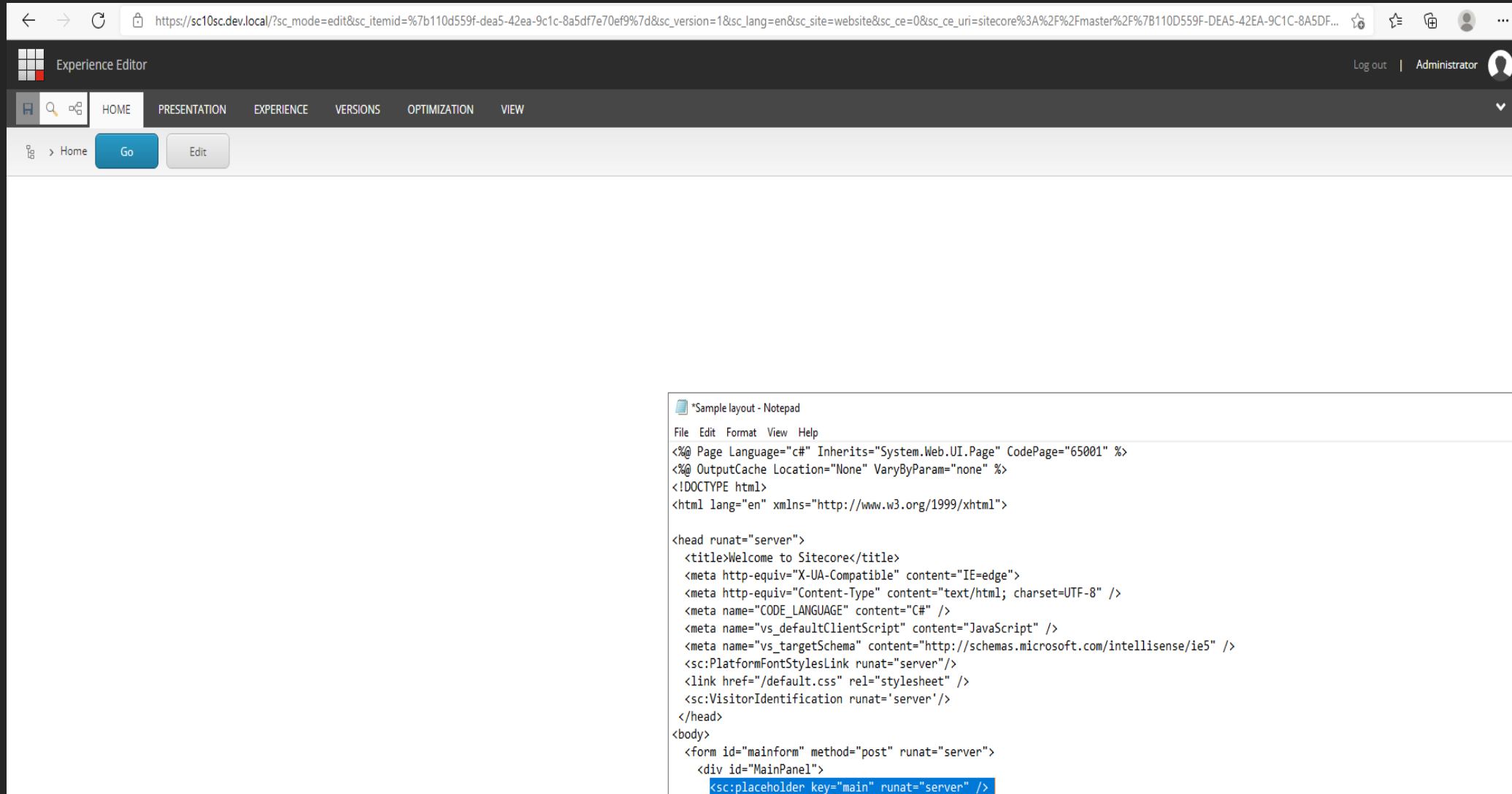
Presentation Details

Layout mandatory for Presentation Details with controls:



Presentation Details

Removing “main” placeholder in chtml does not display the rendering within:



The screenshot displays the Sitecore Experience Editor interface. The top navigation bar includes the 'Experience Editor' logo, a 'Log out' button, and a user profile for 'Administrator'. Below this, a secondary navigation bar contains tabs for 'HOME', 'PRESENTATION', 'EXPERIENCE', 'VERSIONS', 'OPTIMIZATION', and 'VIEW'. The 'HOME' tab is active, showing a 'Go' button and an 'Edit' button. The main workspace is a large, empty white area. In the bottom right corner, a code window titled '*Sample layout - Notepad' is open, displaying the following HTML code:

```
File Edit Format View Help
<%@ Page Language="c#" Inherits="System.Web.UI.Page" CodePage="65001" %>
<%@ OutputCache Location="None" VaryByParam="none" %>
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">
  <title>Welcome to Sitecore</title>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta name="CODE_LANGUAGE" content="C#" />
  <meta name="vs_defaultClientScript" content="JavaScript" />
  <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5" />
  <sc:PlatformFontStylesLink runat="server"/>
  <link href="/default.css" rel="stylesheet" />
  <sc:VisitorIdentification runat="server"/>
</head>
<body>
  <form id="mainform" method="post" runat="server">
    <div id="MainPanel">
      <sc:placeholder key="main" runat="server" />
    </div>
  </form>
</body>
</html>
```

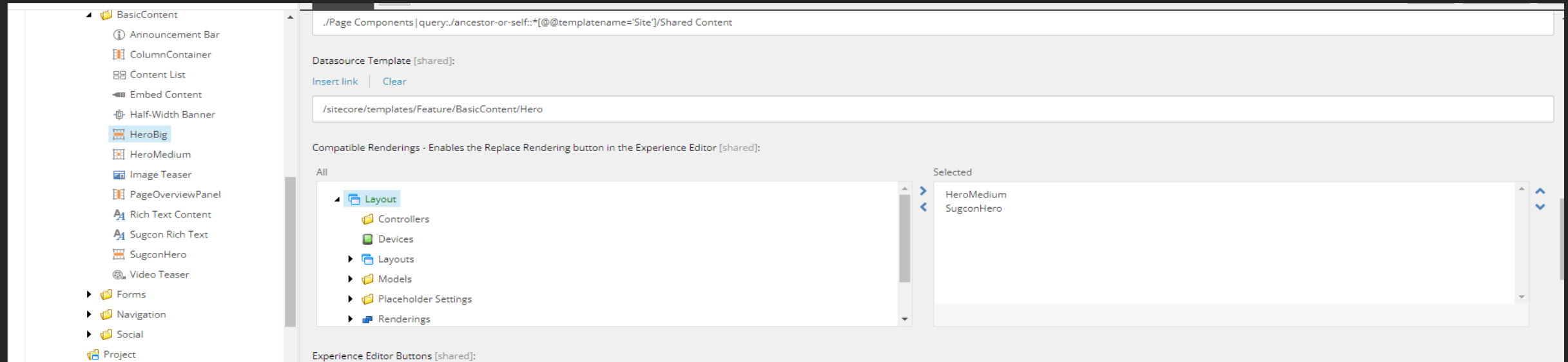
Presentation Details

Placeholder is uniquely identified by Placeholder key

The screenshot displays the Sitecore Presentation Editor interface. The main window shows the 'About' item in the 'MvpSite' folder. A 'Device Editor' dialog box is open, displaying the 'Placeholder Settings' tab. Within this dialog, a sub-dialog titled 'Select the Placeholder Settings' is active, prompting the user to 'Select the placeholder settings item that you want to use or create new settings.' The sub-dialog offers two options: 'Select Existing Settings' and 'Create New Settings'. Under 'Select existing placeholder settings', a tree view shows the hierarchy: Placeholder Settings > Foundation > Feature > Project > content > webedit. The 'Placeholder Key' field is highlighted in yellow. A yellow warning message at the bottom of the sub-dialog states: 'The 'Placeholder Settings' item is not a valid selection.' The background interface includes a top navigation bar with tabs like HOME, NAVIGATE, REVIEW, PUBLISH, VERSIONS, CONFIGURE, and PRESENTATION. A left sidebar shows the site structure, and a right sidebar shows the 'About' item details.

Rendering

- Checkbox or tree list cannot be rendered using `@Html.Sitecore.Field()`
- A Compatible Rendering is a component that can be replaced with another component providing both use the same data source type (or no data source at all) e.g. use the same type of data.

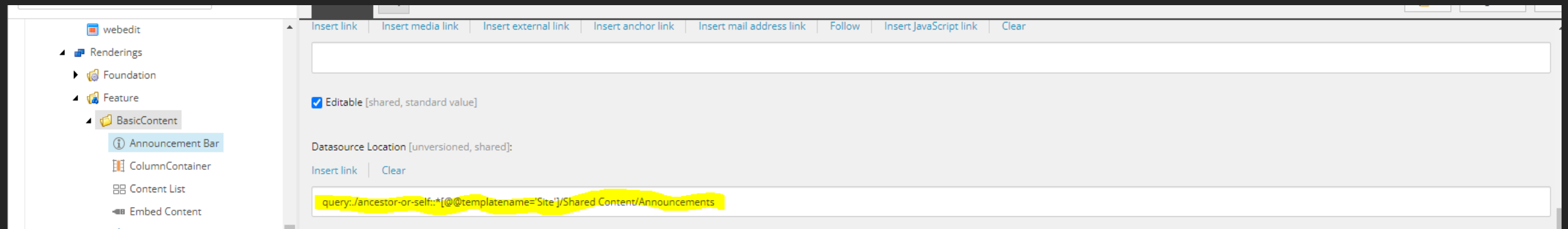


<https://mattneil.co.uk/2017/01/17/compatible-renderings-in-sitecore/>

https://community.sitecore.com/community?id=community_question&sys_id=64a4e7ed1b0770d0b8954371b24bcbb0#14463

General Sitecore MVC titbits

- Dynamic component binding useful in personalization
- Static component binding useful for fixed header or footer and disallows content editors to remove the component
- Configure rendering Datasource location to restrict content editors to select content from a specific location in the content tree
- Configure Datasource template to filter content items based on a template



<https://www.jondjones.com/learn-sitecore-cms/sitecore-developers-guide/sitecore-and-mvc/how-to-make-sitecore-use-a-mvc-controller-renderings-explained/>

Insert Options

Good practice to setup insert options on standard values of an item template

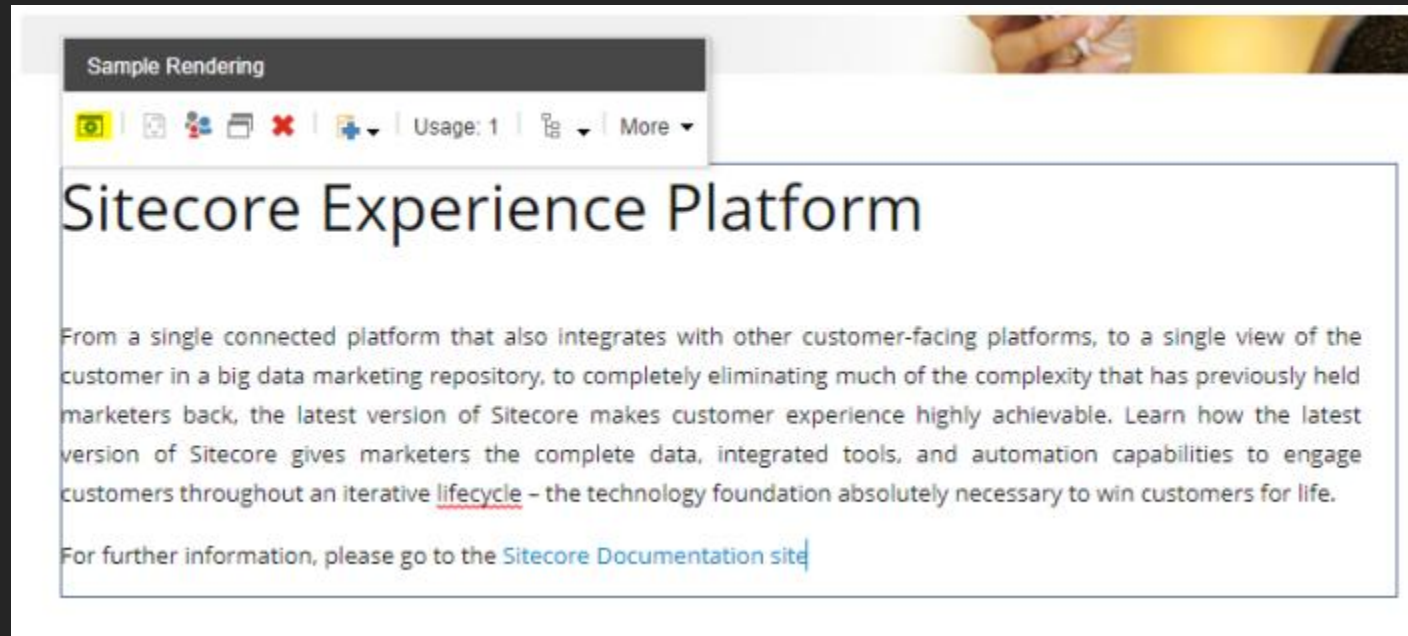
You can set insert options on an item in the content tree as well

The former allows inserting only specific type of items while the latter option is useful when you want to add additional items

<https://doc.sitecore.com/xp/en/developers/91/sitecore-experience-manager/insert-options.html#:~:text=Sitecore%20administrators%20and%20developers%20configure,the%20user%20could%20possibly%20insert.>

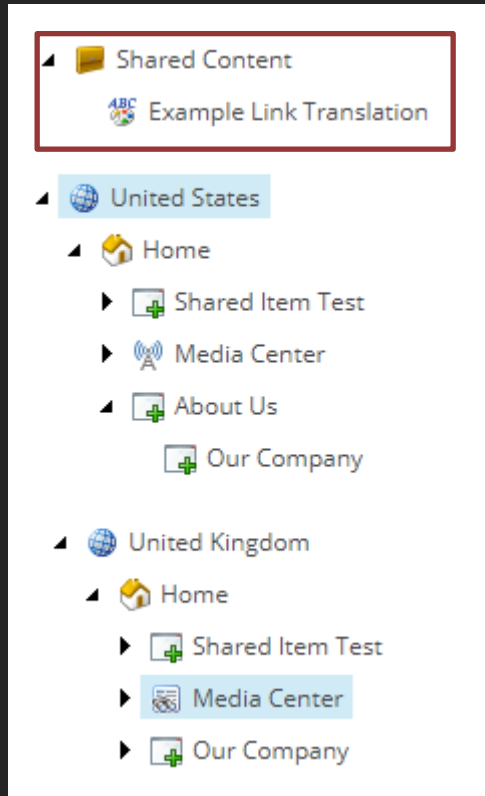
Field Editor Button

Field Editor Buttons allow you to enhance the user experience for a given component. There maybe some field options that are not available in Experience Editor and you have to edit the content via the Content Editor. This overall is not a good user experience. This is where Experience Editor Buttons help and improve the Sitecore user experience.



<https://mattneil.co.uk/2017/07/10/sitecore-experience-editor-buttons-and-their-uses/>

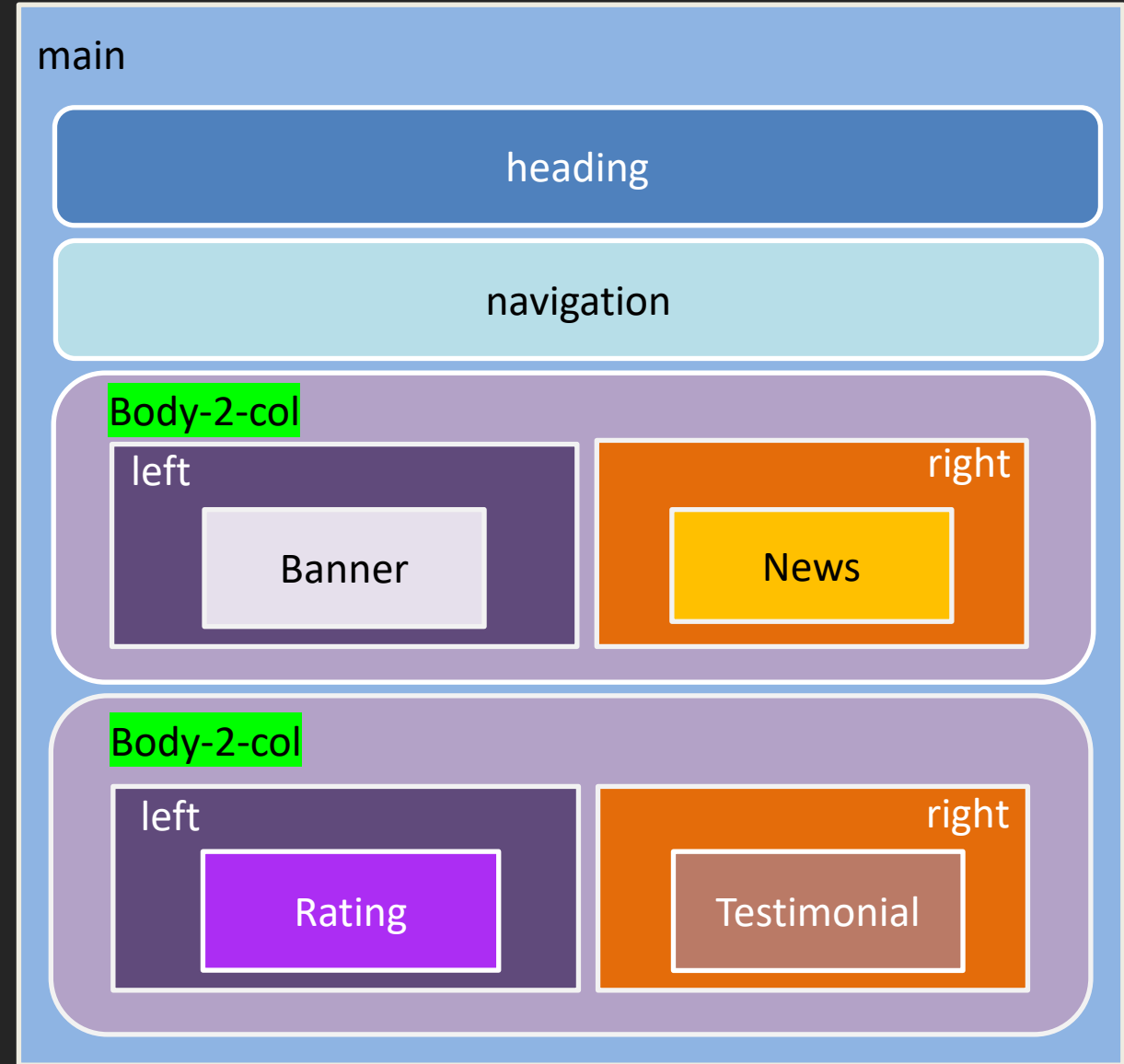
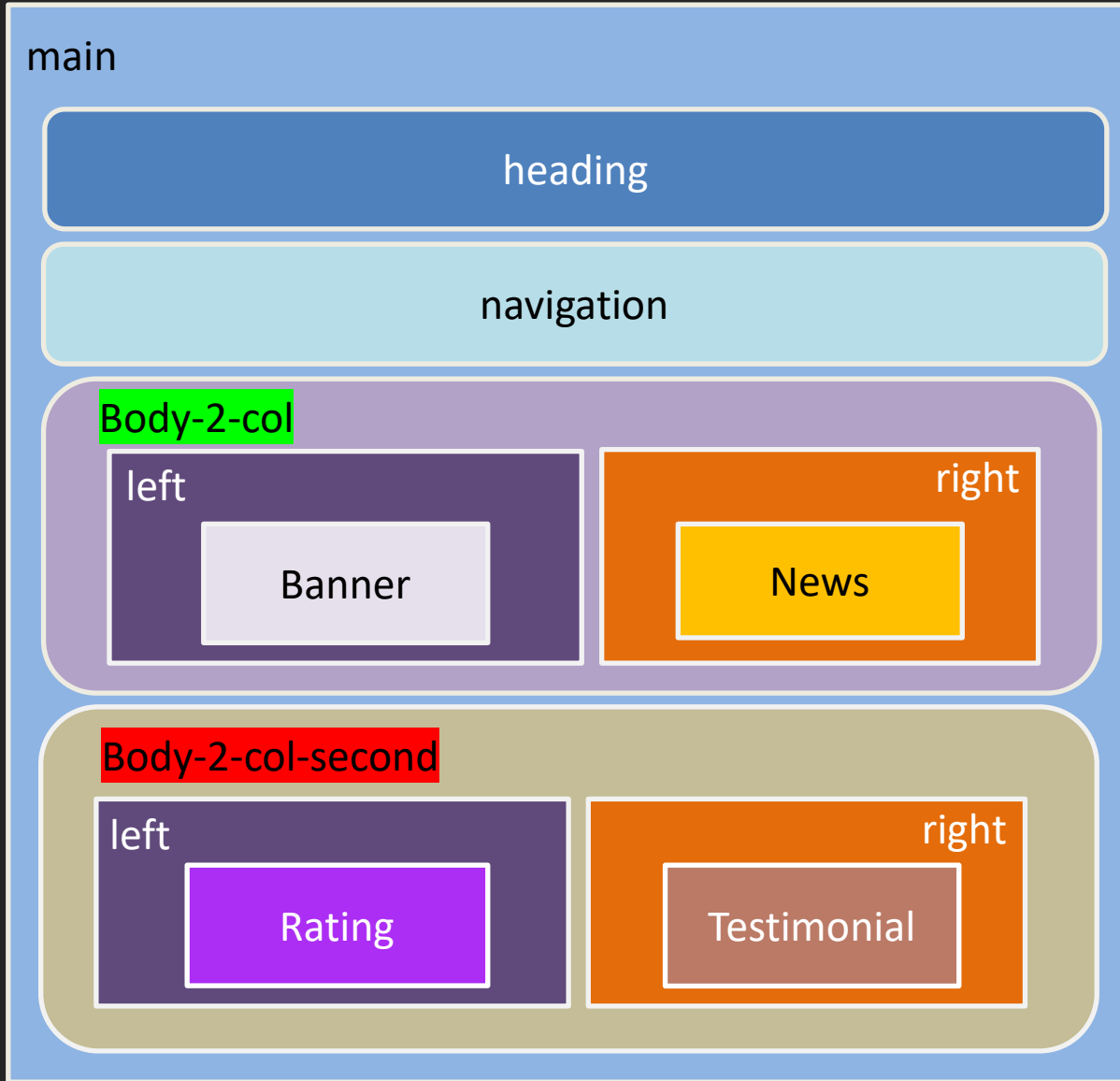
Sharing content between Sites



Best practice:

Shared Content should be stored outside of any one site's content tree structure.

Static vs Dynamic Placeholder



Dynamic Placeholder

Dynamic placeholders: Same key but unique content

When you use dynamic placeholders, you can add the same placeholder name several times. You can do this across multiple renderings, and you can even use the same placeholder multiple times in a single rendering.

Dynamic placeholders mean that:

- Unique keys are guaranteed across different renderings and within one rendering.
- You can build functionality that generates multiple placeholders within the same rendering dynamically.

<https://doc.sitecore.com/xp/en/developers/92/sitecore-experience-manager/dynamic-placeholders.html>

<https://sitecorediaries.org/2019/12/04/what-is-layout-in-sitecore-and-how-can-we-create->

[it/#:~:text=Layouts%20are%20the%20scaffolding%20for,one%20can%20plug%20component%20renderings.](https://sitecorediaries.org/2019/12/04/what-is-layout-in-sitecore-and-how-can-we-create-it/#:~:text=Layouts%20are%20the%20scaffolding%20for,one%20can%20plug%20component%20renderings.)

<https://sitecorediaries.org/2019/12/09/sitecore-dynamic-placeholders/>

Dynamic Placeholder

The screenshot displays the Sitecore Content Editor interface. The main window shows the 'About' item under the 'MvpSite' folder. A 'Control Properties' dialog is open, showing the 'General' tab. The 'Placeholder [shared]:' field is highlighted with a yellow box and contains the value `/main/page-overview-panel-{3920C133-2B8C-49B6-B473-448DC3EE5DCC}-0`. The 'Data Source [shared]:' field contains the path `/sitecore/content/MvpSite/Home/About/Page Components/MVP Types`. The 'Caching' tab is also visible, showing the 'Cacheable' checkbox is unchecked and the 'Clear on Index Update' checkbox is checked. The 'OK' button is highlighted in blue.

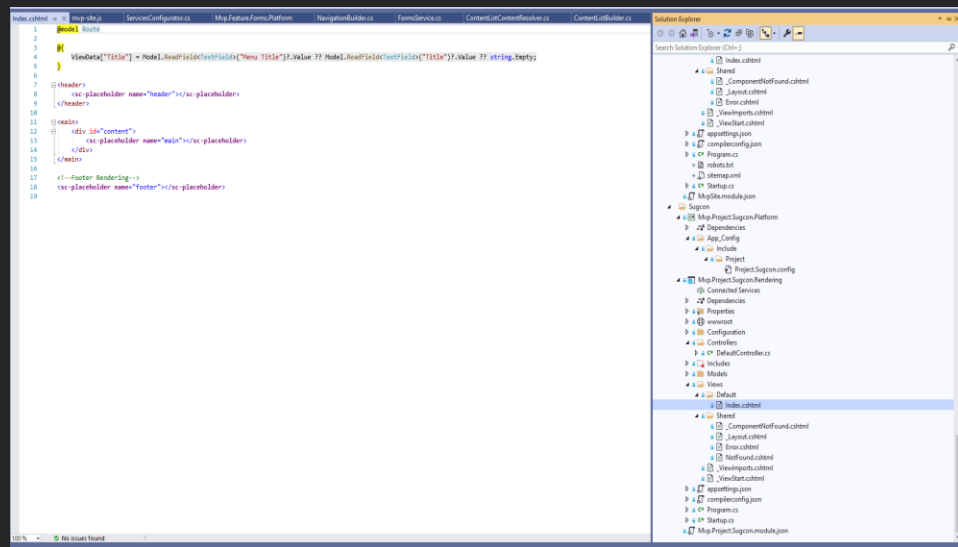
Sitecore Content Editor interface showing the 'Control Properties' dialog for a placeholder.

The dialog displays the following information:

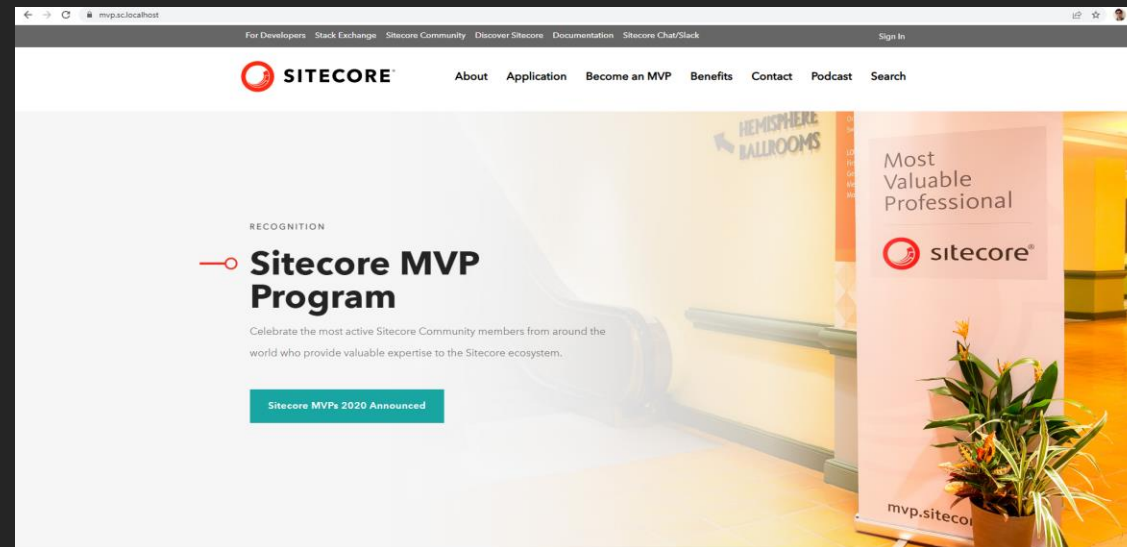
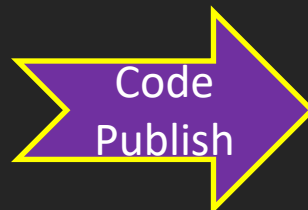
- General**
 - Placeholder [shared]: `/main/page-overview-panel-{3920C133-2B8C-49B6-B473-448DC3EE5DCC}-0`
 - Data Source [shared]: `/sitecore/content/MvpSite/Home/About/Page Components/MVP Types`
- Caching**
 - Cacheable: ☐
 - Clear on Index Update: ☒

The background shows the 'About' item details, including Item ID, Item name, Item path, Template, Created from, Item owner, Menu Title, and the 'Include in Menu' checkbox (checked).

Visual Studio Publish



C:\Source Code



C:\inetpub\wwwroot\<deploy folder>

Note: Code publish different from Content editor publish! Also, remember that you can manually copy cshtml files from your solution to the web root folder

Tip: While writing C# code, Sitecore.Context is utilised to get Http request details

Versioned Layouts

Versioned layouts make it possible to specify different layouts for different versions and languages of the same item.

You use versioned layouts when you want to:

- Have different layouts for different languages.
- Publish a specific version with its own layout for a specific period.

What is true regarding versioned layouts? ^

Versioned layouts make it **possible to specify different layouts for different versions and languages of the same item**. You use versioned layouts when you want to: Have different layouts for different languages. Publish a specific version with its own layout for a specific period. 1 Jan 2022

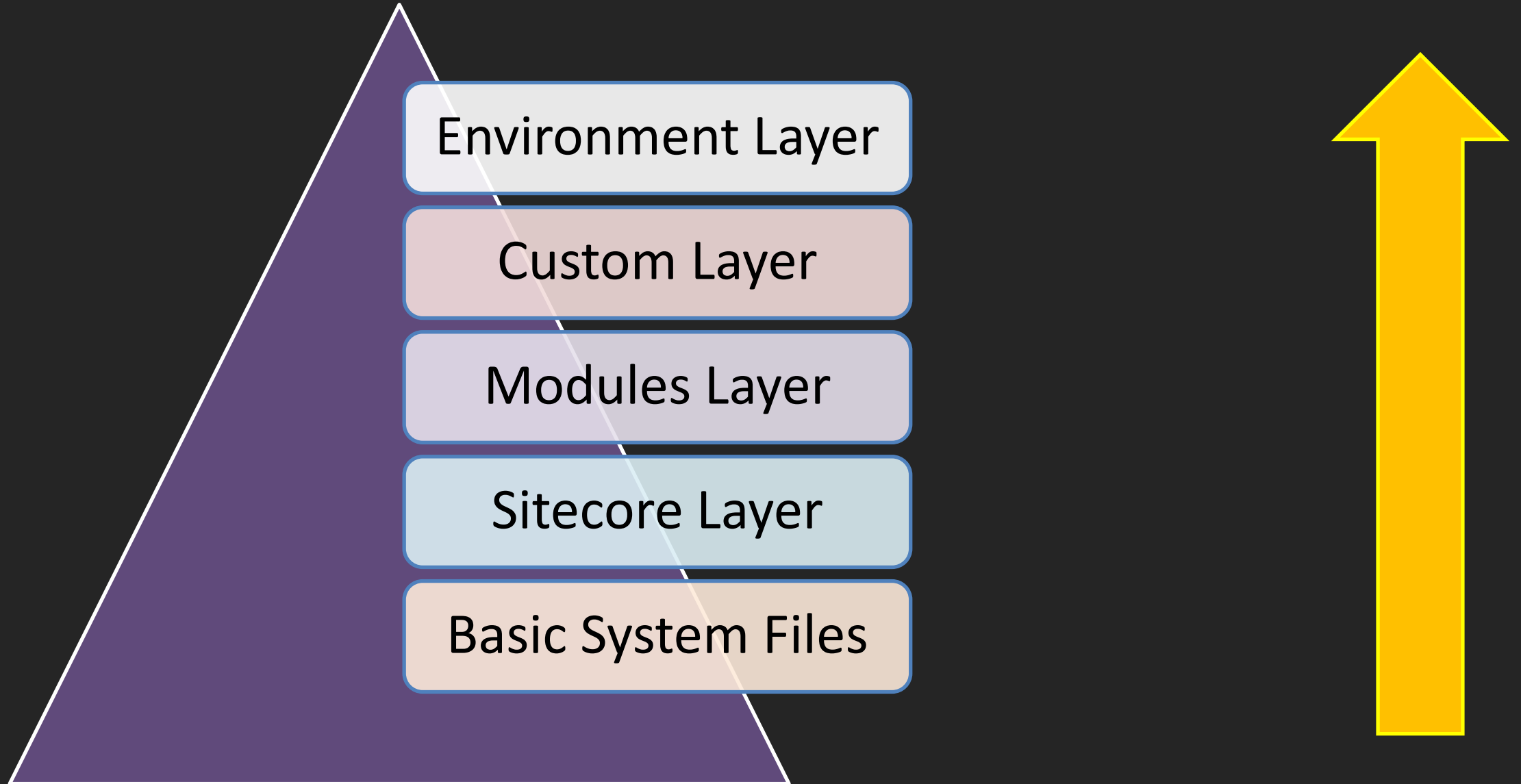
<https://doc.sitecore.com/xp/en/developers/93/sitecore-experience-manager/versioned-layouts.html#:~:text=Versioned%20layouts%20make%20it%20possible,layout%20for%20a%20specific%20period.>

Site Definition Patch Config

Patch file is used to accommodate multiple public-facing sites and used to map home item to corresponding hostname

```
<!-->  
<sites>  
  <site name="shell" isInternal="true" virtualFolder="/sitecore/shell" physicalFolder="/sitecore/shell" rootPath="/sitecore/content" startItem="/home" language="en" contentTypeLanguage="en" database="core" domain="sitecore" loginPage="/identity/login/shell/SitecoreIdentityServer" content="master" contentStartItem="/Home" enableWorkflow="true" enableTracking="false" analyticsDefinitions="content" xmlControlPage="/sitecore/shell/default.aspx" browserTitle="Sitecore" htmlCacheSize="10MB" registryCacheSize="15MB" viewStateCacheSize="1MB" xslCacheSize="25MB" disableBrowserCaching="true" enableItemLanguageFallback="false" enableFieldLanguageFallback="false" patch:source="Sitecore.Speak.ItemWebApi.config" itemwebapi.mode="StandardSecurity" itemwebapi.access="ReadWrite" itemwebapi.allowanonymousaccess="false"/>  
  <site name="login" isInternal="true" virtualFolder="/sitecore/login" physicalFolder="/sitecore/login" enableTracking="false" database="core" domain="sitecore" disableXmlControls="true" language="en" patch:source="Sitecore.Sites.config"/>  
  <site name="admin" isInternal="true" virtualFolder="/sitecore/admin" physicalFolder="/sitecore/admin" enableTracking="false" enableWorkflow="true" domain="sitecore" loginPage="/identity/login/admin/SitecoreIdentityServer" patch:source="Sitecore.Owin.Authentication.IdentityServer.config"/>  
  <site name="service" isInternal="true" enableTracking="true" virtualFolder="/sitecore/service" physicalFolder="/sitecore/service" patch:source="Sitecore.Sites.config"/>  
  <site name="modules_shell" isInternal="true" virtualFolder="/sitecore/modules/shell" physicalFolder="/sitecore/modules/shell" rootPath="/sitecore/content" startItem="/home" language="en" database="core" domain="sitecore" content="master" enableTracking="false" enableWorkflow="true" patch:source="Sitecore.Sites.config"/>  
  <site name="modules_website" isInternal="true" virtualFolder="/sitecore/modules/web" physicalFolder="/sitecore/modules/web" rootPath="/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true" patch:source="Sitecore.Sites.config"/>  
<!-- ITEM WEB API SETTINGS FOR A SITE  
    Supported attributes (first is default):  
      itemwebapi.mode: [Off|StandardSecurity|AdvancedSecurity]  
        If set to Off, Item Web API is turned off.  
        If set to StandardSecurity, Item Web API is turned on. Default Sitecore security model is used.  
        If set to AdvancedSecurity, Item Web API is turned on. Default Sitecore security model is extended with a requirement to explicitly set the 'remote:fieldread' access right for content fields.  
      itemwebapi.access: [ReadOnly|ReadWrite]  
        If set to ReadOnly, then only READ operation is allowed.  
        If set to ReadWrite, then CREATE, READ, UPDATE, and DELETE operations are allowed.  
      itemwebapi.allowanonymousaccess: [false|true].  
        Defines if access is allowed for non-authenticated user.  
-->  
<site name="mvp-site" inherits="website" hostName="mvp-cd.sc.localhost" rootPath="/sitecore/content/MvpSite" dictionaryDomain="{D8704D9E-113D-44A1-AA85-2A5D127CD2A3}" scheme="https" patch:source="Project.MvpSite.config"/>  
<site name="sugcon-eu" inherits="website" hostName="mvp-cd.sc.localhost" rootPath="/sitecore/content/Sugcon-EU" dictionaryDomain="{26A9EC85-D857-4339-9BEB-527CBC62EAB4}" scheme="https" patch:source="Project.Sugcon.config"/>  
<site name="website" enableTracking="true" virtualFolder="/" physicalFolder="/" rootPath="/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true" htmlCacheSize="50MB" registryCacheSize="0" viewStateCacheSize="0" xslCacheSize="25MB" filteredItemsCacheSize="10MB" enablePreview="true" enableWebEdit="true" enableDebugger="true" disableClientData="false" cacheRenderingParameters="true" renderingParametersCacheSize="10MB" enableItemLanguageFallback="false" enableFieldLanguageFallback="false" patch:source="Sitecore.ItemWebApi.config" itemwebapi.mode="Off" itemwebapi.access="ReadOnly" itemwebapi.allowanonymousaccess="false"/>  
<site name="scheduler" isInternal="true" enableTracking="false" domain="sitecore" patch:source="Sitecore.Sites.config"/>  
<site name="system" isInternal="true" enableTracking="false" domain="sitecore" patch:source="Sitecore.Sites.config"/>  
<site name="publisher" isInternal="true" domain="sitecore" enableTracking="false" enableWorkflow="true" patch:source="Sitecore.Sites.config"/>  
</sites>
```

Config Files Load Order



Layers/Load Order

Sitecore defines the layers and the order in which they load at runtime in the `Website/App_Config/layers.config` file.

▼ The load order of configuration files

By default, Sitecore loads its configuration files in this order:

1. Basic system files, such as `layers.config`, `ConnectionStrings.config`, and `web.config`.
2. Configuration files in the Sitecore layer.
3. Configuration files in the Modules layer.
4. Configuration files in the Custom layer.
5. Configuration files in the Environment layer.

<https://doc.sitecore.com/xp/en/developers/92/platform-administration-and-architecture/configuration-layers.html>

Layers Config File

```
mvp-site-new_cm_1: Layers.config  mvp-site.js  ServicesConfigurator.cs  Mvp.Feature.Forms.Platform
1  <?xml version="1.0" encoding="utf-8"?>
2  <layers>
3    <layer name="Sitecore" includeFolder="/App_Config/Sitecore/">
4      <loadOrder>
5        <add path="CMS.Core" type="Folder" />
6        <add path="AntiCSRFModule" type="Folder" />
7        <add path="Contact.Enrichment.Services.Client" type="Folder" />
8        <add path="ContentSearch" type="Folder" />
9        <add path="Buckets" type="Folder" />
10       <add path="DeviceDetection.Client" type="Folder" />
11       <add path="DetectionServices.Location" type="Folder" />
12       <add path="ItemWebApi" type="Folder" />
13       <add path="Owin.Authentication" type="Folder" />
14       <add path="Owin.Authentication.IdentityServer" type="Folder" />
15       <add path="Update" type="Folder" />
16       <add path="Mvc" type="Folder" />
17       <add path="Services.Client" type="Folder" />
18       <add path="ExperienceContentManagement.Administration" type="Folder" />
19       <add path="Speak.Integration" type="Folder" />
20       <add path="SPEAK" type="Folder" />
21       <add path="Speak.Applications" type="Folder" />
22       <add path="LaunchPad" type="Folder" />
23       <add path="Experience Editor" type="Folder" />
24       <add path="ContentTagging" type="Folder" />
25       <add path="MVC.ExperienceEditor" type="Folder" />
26       <add path="MVC.DeviceSimulator" type="Folder" />
27       <add path="Personalization" type="Folder" />
28       <add path="SPEAK.Components" type="Folder" />
29       <add path="ExperienceForms" type="Folder" />
30       <add path="UpdateCenter" type="Folder" />
31     </loadOrder>
32   </layer>
33   <layer name="Modules" includeFolder="/App_Config/Modules/" />
34   <layer name="Custom" includeFolder="/App_Config/Include/">
35     <loadOrder>
36       <add path="Foundation" type="Folder" />
37       <add path="Feature" type="Folder" />
38       <add path="Project" type="Folder" />
39     </loadOrder>
40   </layer>
41   <layer name="Environment" includeFolder="/App_Config/Environment/" />
42 </layers>
```

Sitecore Support Package

Purpose: Help Sitecore team replicate an issue without accessing the actual Sitecore environment

← → ↻ mvp-cm.sc.localhost/sitecore/admin/supportpackage

Administration - Support Package

Back

1

Package details

2

Choose data to collect

3

Process

4

Done

Package Name *

scsupport_mvp-cm.sc.localhost_20220210014637

Support Ticket

Log start date *

2/4/2022

Log end date *

2/10/2022

Comments

Next

Sitecore Support Package

← → ↻ mvp-cm.sc.localhost/sitecore/admin/supportpackage

Administration - Support Package

Back

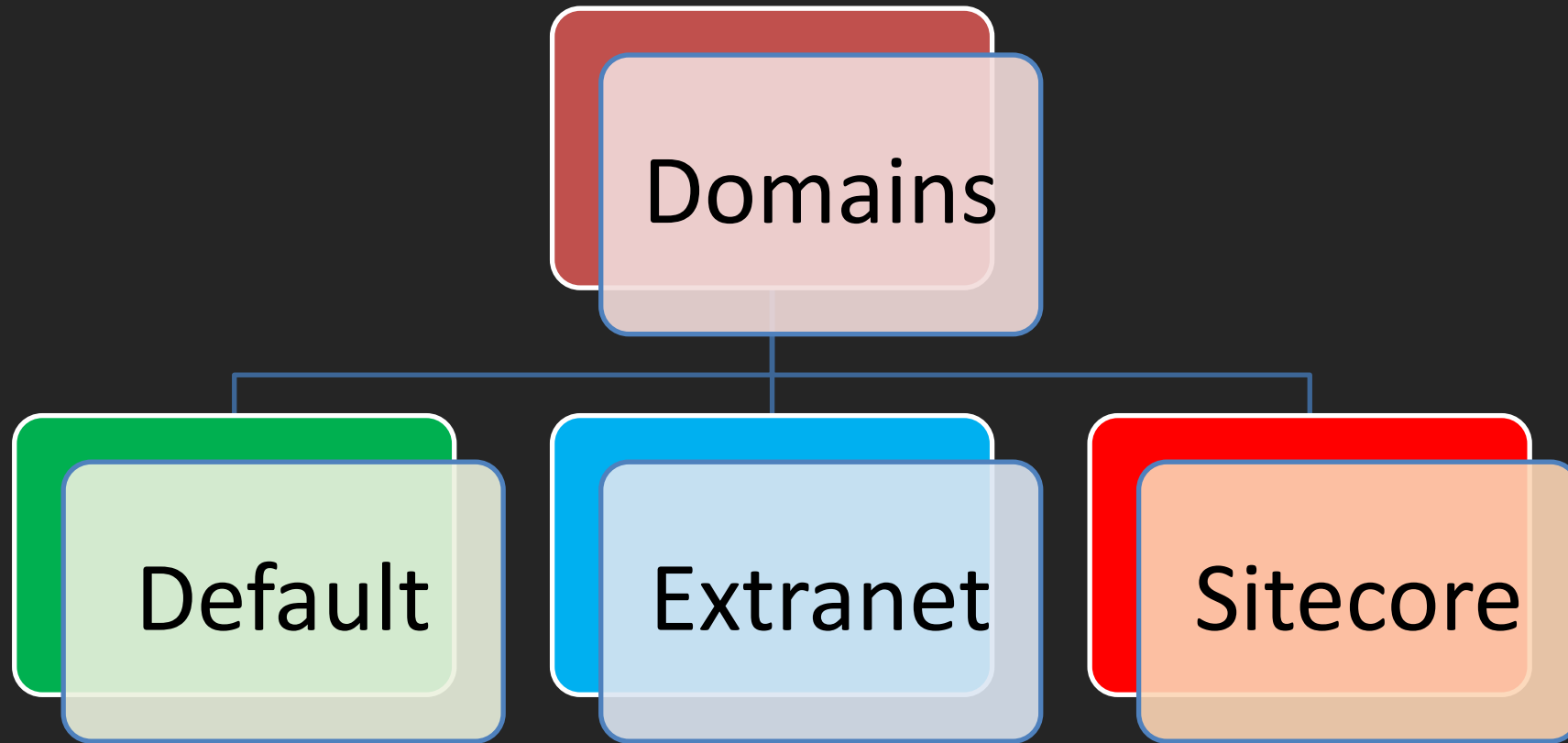
✓ Package details — 2 Choose data to collect — 3 Process — 4 Done

Selected	No.	Description
<input checked="" type="checkbox"/>	D-01	General information (Sitecore version, file hash sums, OS and IIS data)
<input checked="" type="checkbox"/>	D-02	Sitecore license metadata (id, expiration, version, licensee, included products)
<input checked="" type="checkbox"/>	D-03	Runtime configuration and files (web.config, global.asax, /App_Config folder)
<input checked="" type="checkbox"/>	D-04	Sitecore log files
<input checked="" type="checkbox"/>	D-05	Databases metrics, schemas and system properties
<input checked="" type="checkbox"/>	D-06	Custom assembly files from /bin folder, mscordacwks.dll, sos.dll files
<input checked="" type="checkbox"/>	D-07	Diagnostics data such as performance counters and configuration history

Collect

https://support.sitecore.com/kb?id=kb_article_view&sysparm_article=KB0406145

Sitecore Security Domains



<https://doc.sitecore.com/xp/en/developers/93/platform-administration-and-architecture/security-domains.html#:~:text=Sitecore%20is%20setup%20with%20Extranet,are%20set%20to%20default%5Canonymous.>

Sitecore Security Domains

▼ Default domains

Sitecore contains the following default domains:

- **Extranet** — this domain is a website security domain that contains the user accounts that correspond to the visitors to the website. It also contains the customized roles that manage read access to the content of the website.

Note

Members of the Extranet domain that are also members of relevant Sitecore roles (for example, Sitecore Client Authoring) can access the Sitecore domain and use the client tools to edit the content of the website.

- **Sitecore** — this domain is an internal security domain that contains all the users who can access the Sitecore clients and the Sitecore Client roles that influence the functionalities that are available to users. It also contains the customized roles that control the access that users have to items.

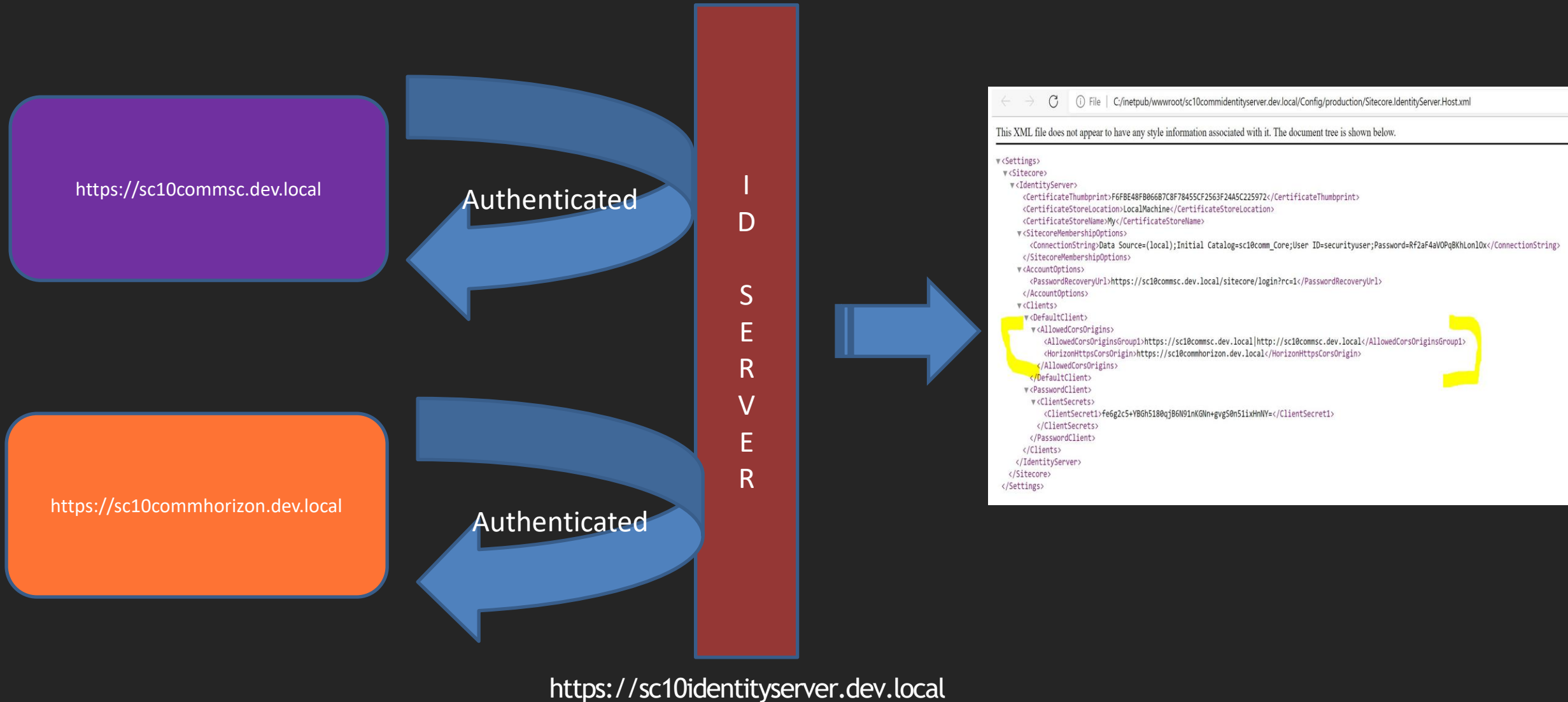
Note

Members of the Sitecore domain can access the Sitecore client tools and edit the website if they have the appropriate access rights. Furthermore, they may be able to access the Extranet domain depending on how the developers and the security architect have designed the domain and the login page.

- **Default** — this is a virtual domain that only exists in memory. Sitecore is setup with Extranet as the default domain and most user accounts will be extranet\anonymous when they visit the website. However, if the website does not specify a default domain, then the users are set to default\anonymous.

<https://doc.sitecore.com/xp/en/developers/93/platform-administration-and-architecture/security-domains.html#:~:text=Sitecore%20is%20setup%20with%20Extranet,are%20set%20to%20default%5Canonymous.>

Allowed CORS Origin



User Manager

Familiarise with user options below -

User Manager

mvp-cm.sc.localhost/sitecore/shell/Applications/Security/User%20Manager.aspx?sc_bw=1

Log out | Administrator

New

Edit

Delete

Change password

Reset settings

Disable

Enable

Unlock

Role Manager

Domain Manager

Access Viewer

Security Editor

Serialize user

Revert user

Serialize all users

Revert all users

Search

User name	Domain	Fully qualified name	Full name	Email	Comment	Language	Locked
Anonymous	default	default\Anonymous					
Anonymous	extranet	extranet\Anonymous					
Admin	sitecore	sitecore\Admin	Administrator		Sitecore Administrator		
JssImport	sitecore	sitecore\JssImport		no_reply@sitecore.net			Disabled
ServicesAPI	sitecore	sitecore\ServicesAPI	Services API User		Default user used when accessing Sitecore via the Sitecore.Services web service APIs.		

Message

⚠ Are you sure you want to unlock sitecore\JssImport?

OKCancel

⏮ ⏪ ⏩ ⏭

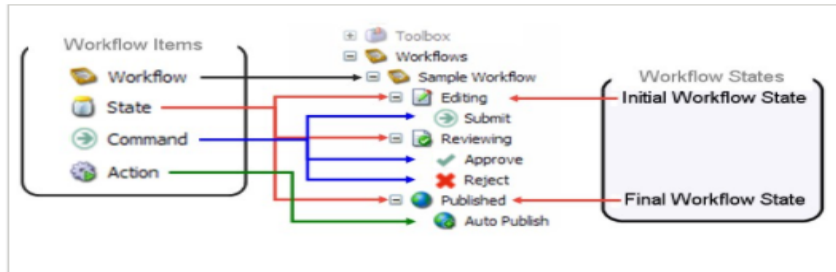
Page 1 of 1 (5 items)

Workflow

Workflows

In Sitecore, you can implement any number of different workflows. As a minimum, each workflow must contain two workflow states – the initial and final workflow state. You can only publish an item in a workflow if it is in the final workflow state.

An example of a workflow:



In this example, the workflow consists of three states; Editing, Reviewing and Published.

All of the following workflow items are stored in the content tree:

- **Workflows** – the parent item that contains all the subitems that make up the workflow.
- **States** – the basic building blocks of the workflow. The workflow states represent the different stages in your content creation process.
- **Commands** – move content items from one workflow state to another. When the command is executed, the item is moved to the workflow state that is defined in the command item.
- **Actions** – the actions that are automatically performed on the items when they are in particular workflow states or when particular workflow commands are executed.

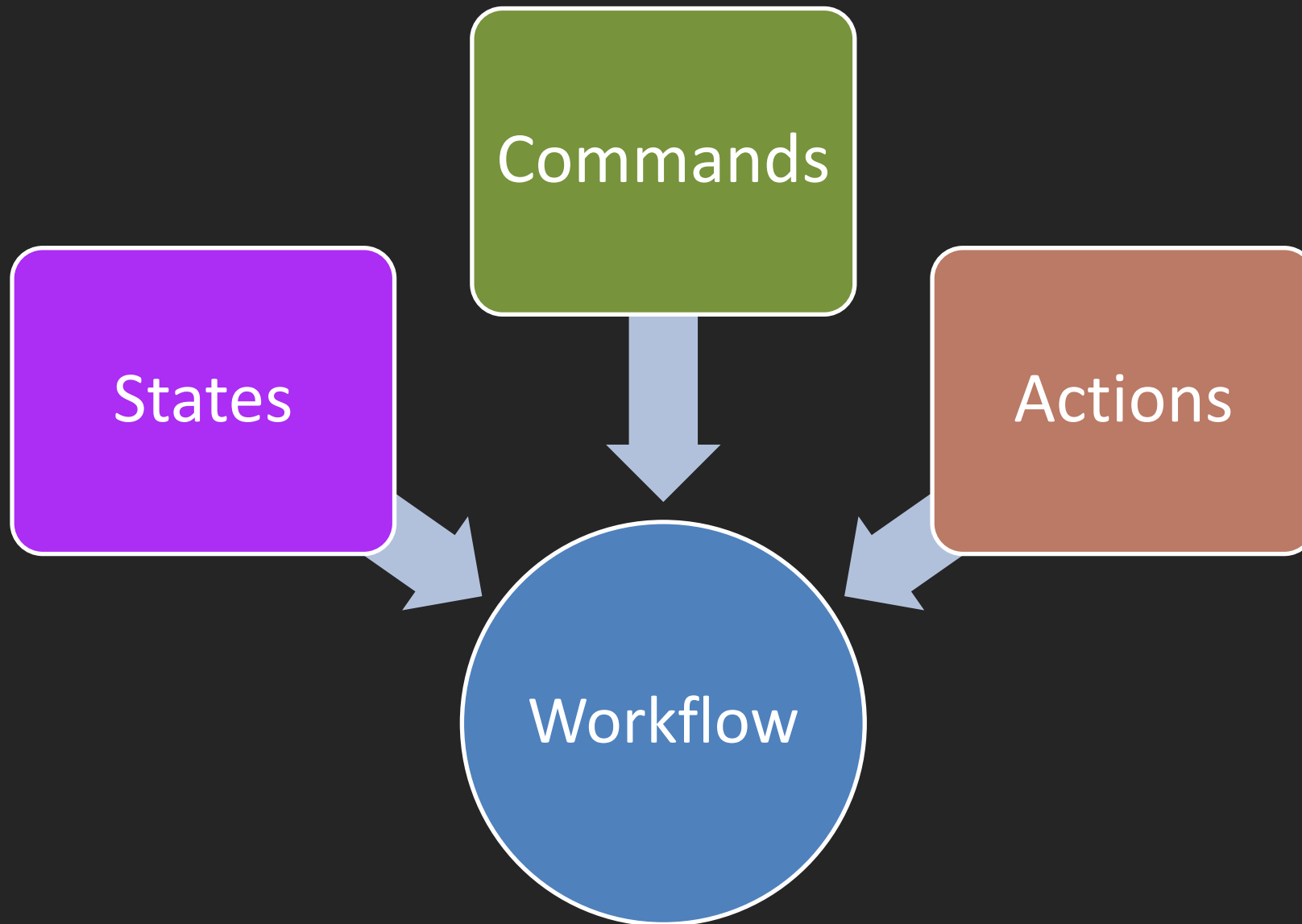
- **Note**

An administrator can overwrite any workflow state and move an item through the states or publish the item at any given time.

Workflow Commands:

Content Editor
Experience Editor
Workbox

Elements of a Workflow



Sitecore Workbox

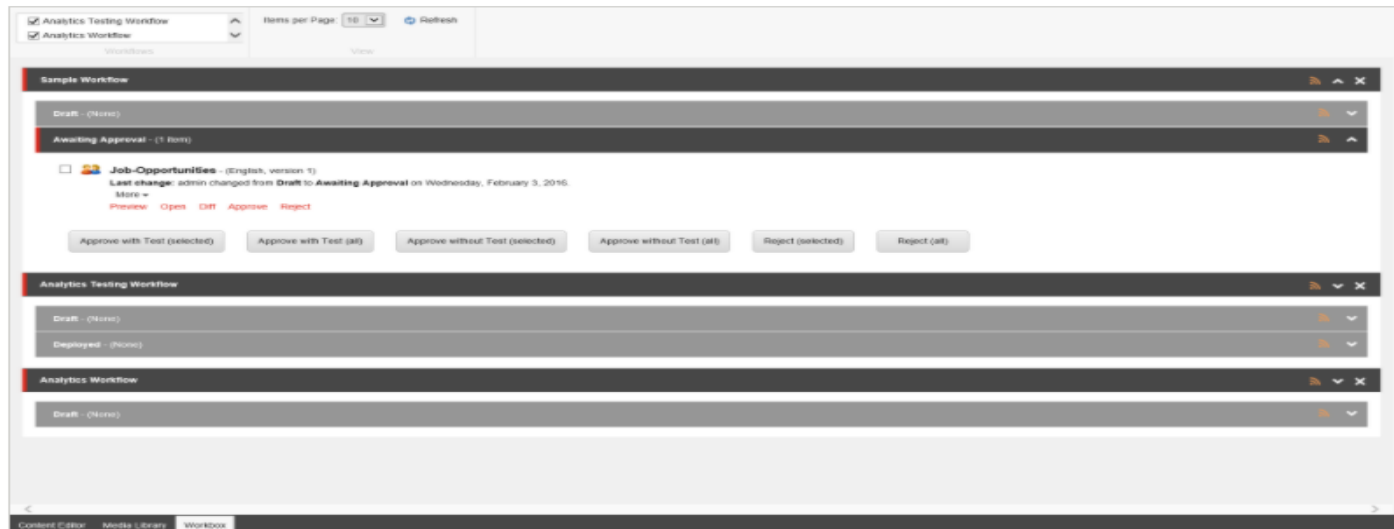
▼ The Workbox

The Workbox gives you an overview of all items that are in workflows and enables you to manage these items.

In the Workbox, you can see the editing history of items in the workflow, each item's workflow state, and the number of items in each workflow state.

For each item in a workflow state, you can preview, open and [compare different versions](#) of the same item. You can also set up an RSS feed, which enables you to receive a notification when there are changes to an item or when a version of an item enters or leaves a specific workflow state.

You can open the Workbox from the **Launchpad**.



<https://doc.sitecore.com/xp/en/users/93/sitecore-experience-platform/workflows-and-the-workbox.html>

Users/Groups - Security

The screenshot shows a 'Select roles' dialog box with a dark header and a light body. The dialog is divided into two main sections: 'Available roles' and 'Selected roles'.

Available roles: This section features a search bar and a list of 25 roles. The role 'sitecore\Developer' is currently selected and highlighted in blue. The list of roles includes:

- sitecore\Author
- sitecore\Designer
- sitecore\Developer
- sitecore\Forms Data Administrator
- sitecore\Forms Editor
- sitecore\Forms Publisher
- sitecore\JSS Import Service Users
- sitecore\PowerShell Extensions Remoting
- sitecore\PowerShellExtensionsAPI
- sitecore\Sitecore Client Account Managing
- sitecore\Sitecore Client Advanced Publishing
- sitecore\Sitecore Client Authoring
- sitecore\Sitecore Client Bucket Management
- sitecore\Sitecore Client Configuring
- sitecore\Sitecore Client Designing

Selected roles: This section contains a list of three roles that have been added: 'sitecore\Author', 'sitecore\Designer', and 'sitecore\Developer'. Above this list are 'Add' and 'Remove' buttons.

At the bottom of the 'Available roles' list, there are navigation controls (first, previous, next, last buttons and a scrollbar) and a pagination indicator showing 'Page 1 of 2 (25 items)'. The dialog itself has 'OK' and 'Cancel' buttons at the bottom right.

In the background, a portion of the 'User Management' interface is visible, showing a sidebar with 'New' and 'Add' buttons, and a list of users including 'Anonym', 'Admin', 'JssImport', and 'Services'.

Users/Groups - Security

Can edit roles for a user

The screenshot displays the 'User Manager' application window. A 'Create a New User' dialog box is open in the foreground, allowing the user to enter information for a new user. The background shows the 'Users' management interface with a table of existing users and a 'Serialization' section.

User Manager

Create a New User
Enter information about the user.

User name:

Domain:

Full name:

Email:

Comment:

Password:

Confirm password:

Roles:

User Profile:

Users

User name	Domain	Fully qualified name
Anonymous	default	default\Anonymous
Anonymous	extranet	extranet\Anonymous
Admin	sitecore	sitecore\Admin
JssImport	sitecore	sitecore\JssImport
ServicesAPI	sitecore	sitecore\ServicesAPI

Serialization

Serialize user
Revert user

Search

	Language	Locked
		Disabled
re.Services web service APIs.		

Page 1 of 1 (5 items)

Role Manager

Can add users as members of a role

Role Manager

NewDeleteMembersMember O

Role

sitecore\Author

sitecore\Designer

sitecore\Developer

sitecore\Forms Data Administrator

sitecore\Forms Editor

sitecore\Forms Publisher

sitecore\JSS Import Service Users

sitecore\PowerShell Extensions Remoting

sitecore\PowerShellExtensionsAPI

sitecore\Sitecore Client Account Managing

sitecore\Sitecore Client Advanced Publishin

sitecore\Sitecore Client Authoring

sitecore\Sitecore Client Bucket Managemen

sitecore\Sitecore Client Configuring

sitecore\Sitecore Client Designing

Members

Add or remove the roles that must be members of the role you selected.

Search

Domain	Local name	Full name	Comment
sitecore	jssImport	sitecore\jssImport	User
sitecore	sitecore\Developer	sitecore\Developer	Role

⏪⏩⏴⏵

Page 1 of 1 (2 items)

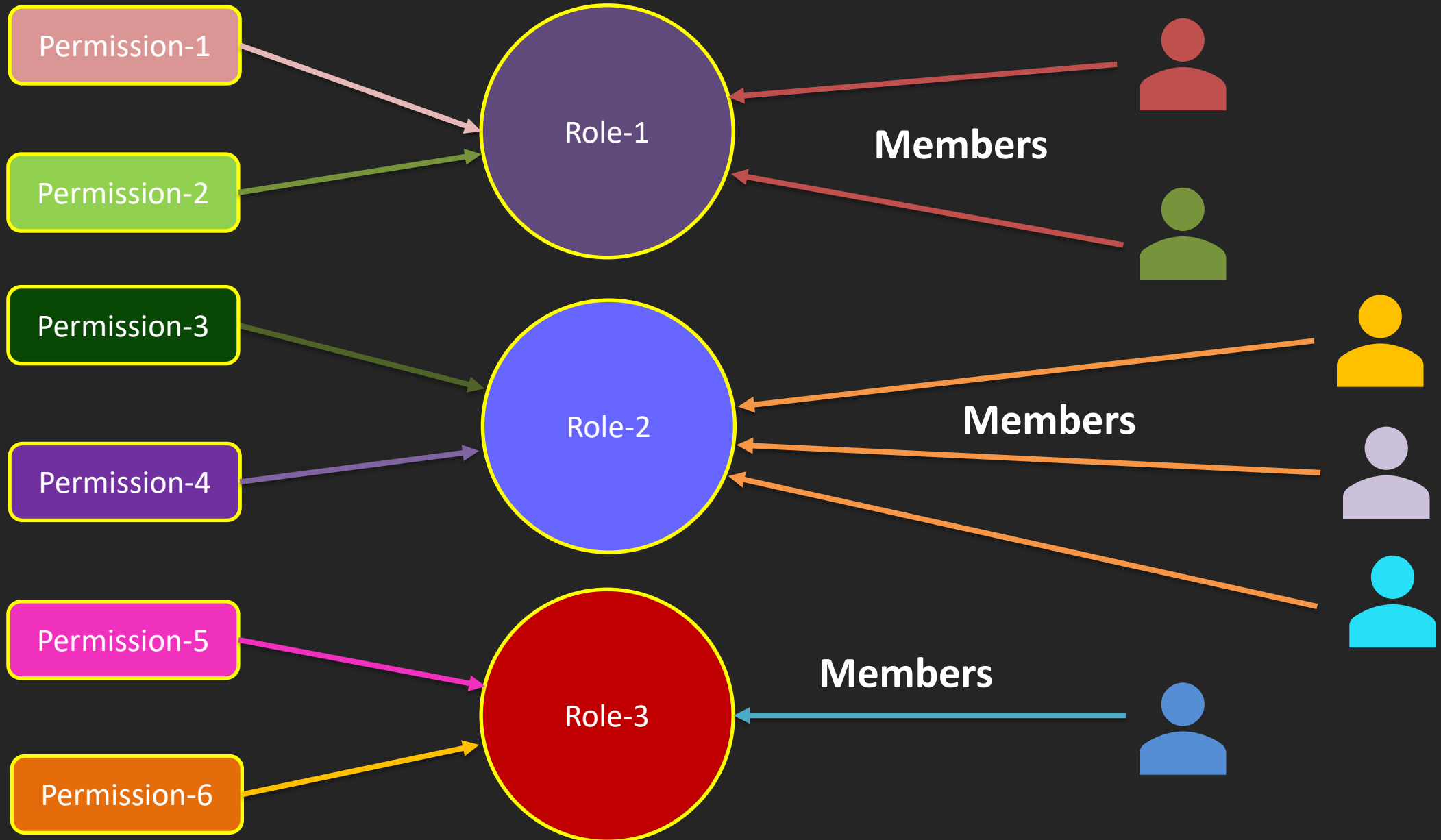
AddRemove

Close

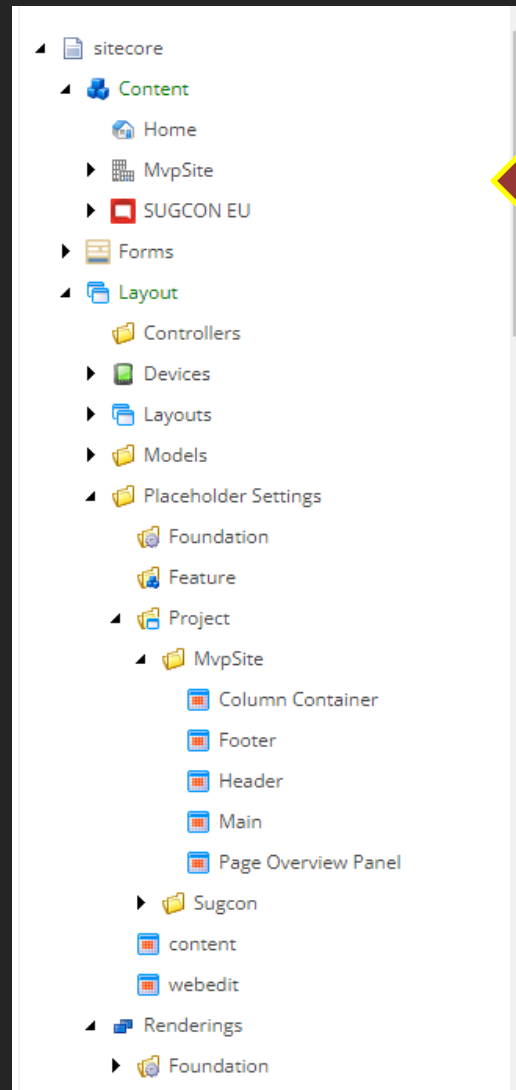
⏪⏩⏴⏵

Page 1 of 2 (25 items)

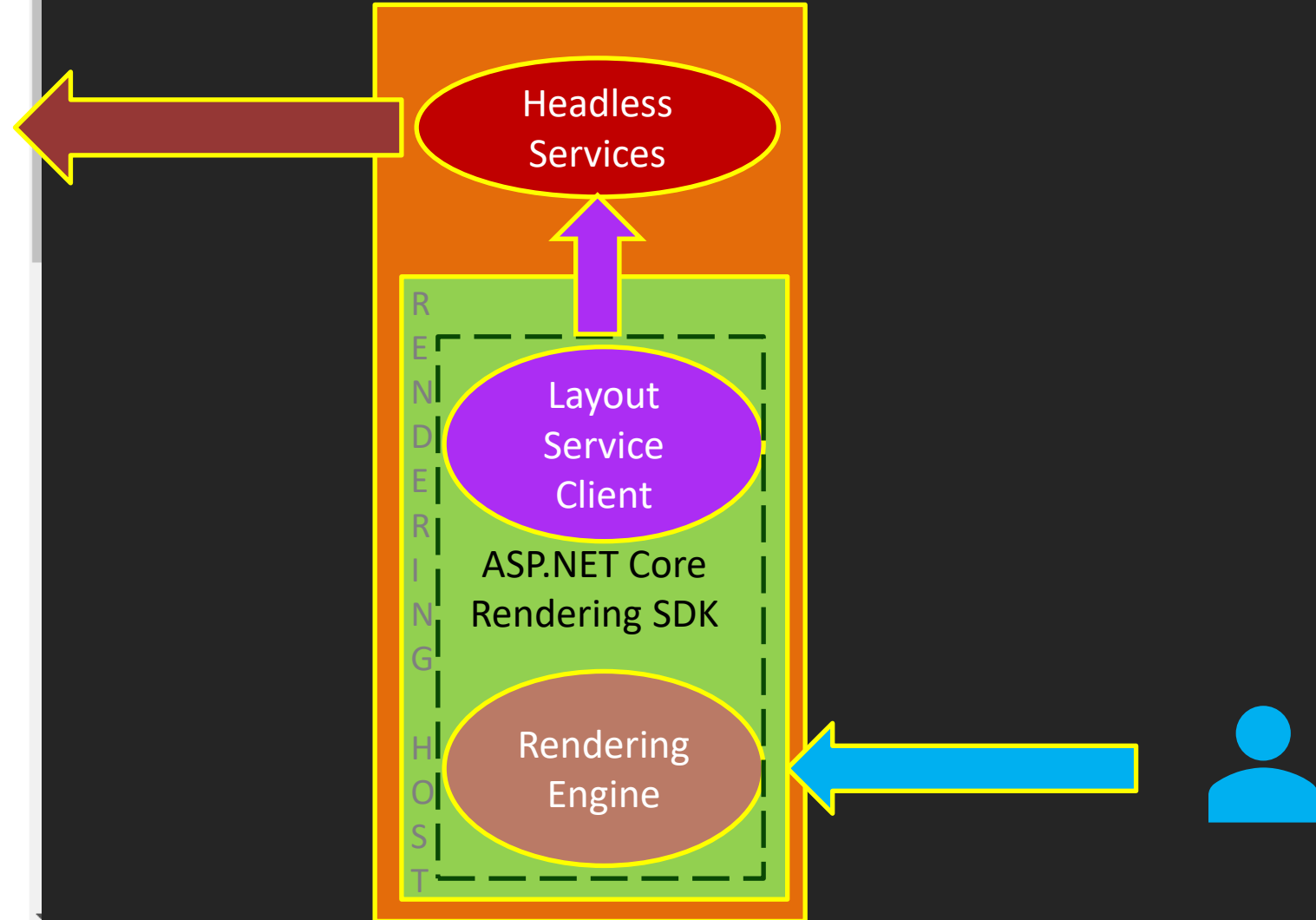
Users/Groups - Security



Sitecore ASP.NET Core Rendering Host

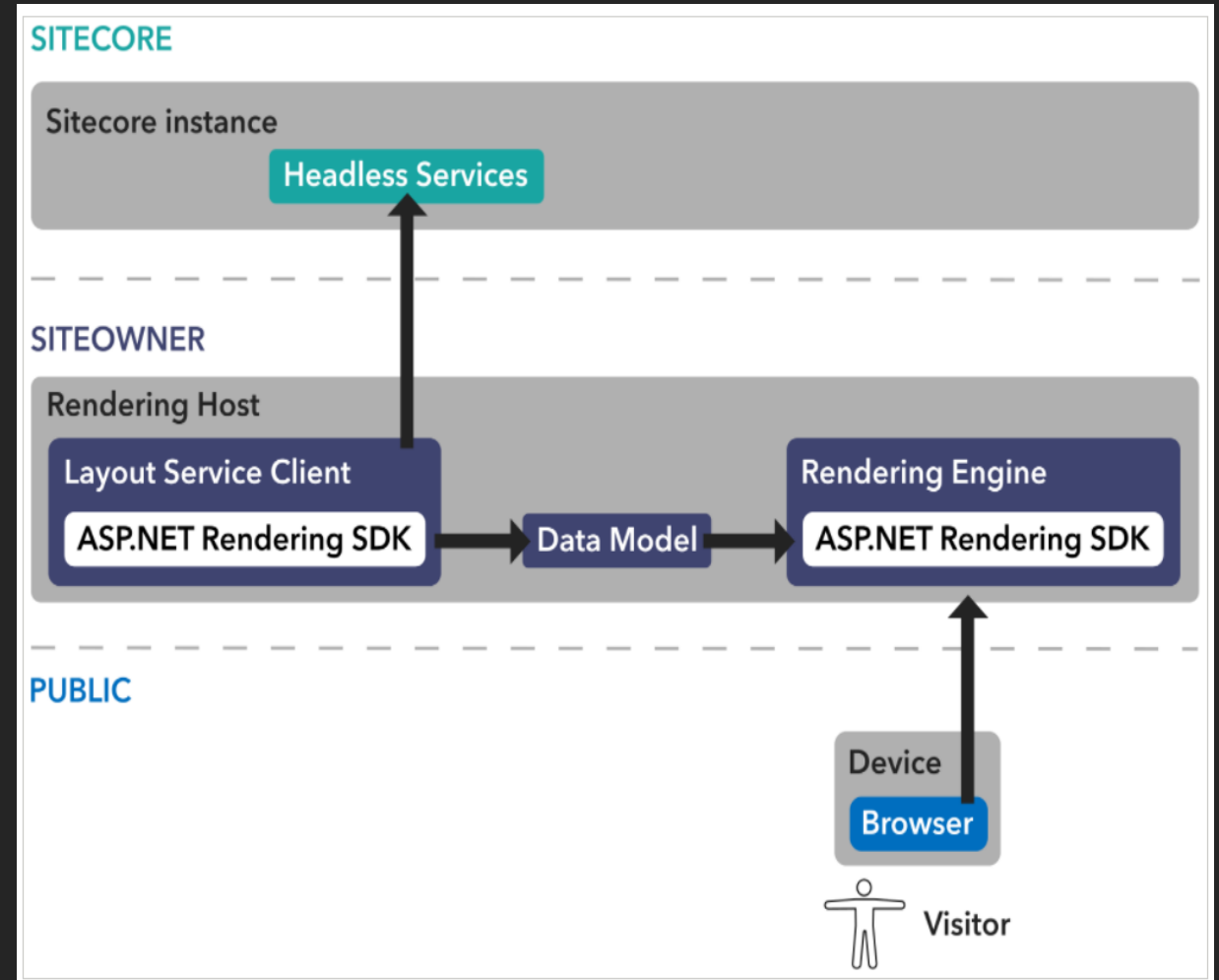
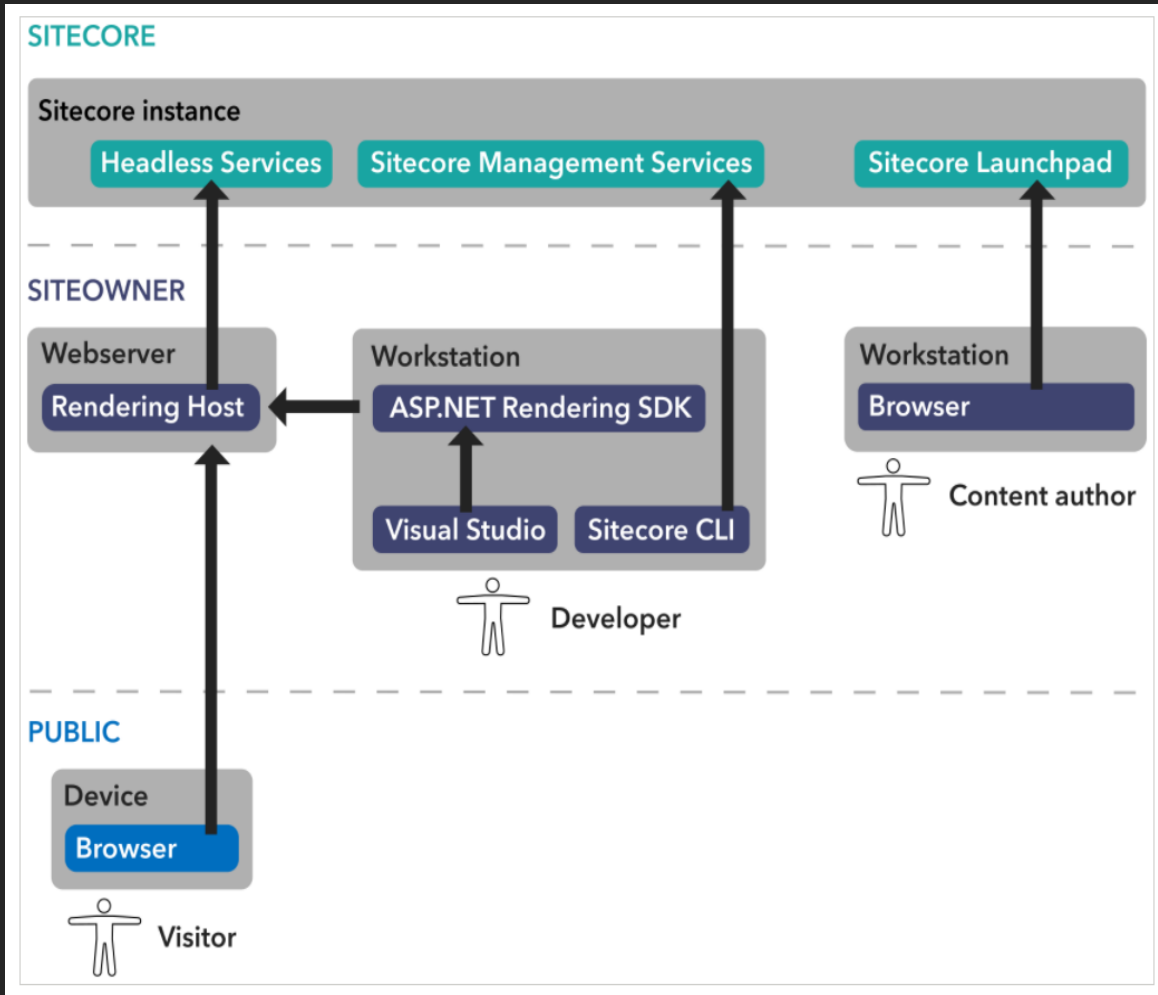


CM



CD

Sitecore Headless Service

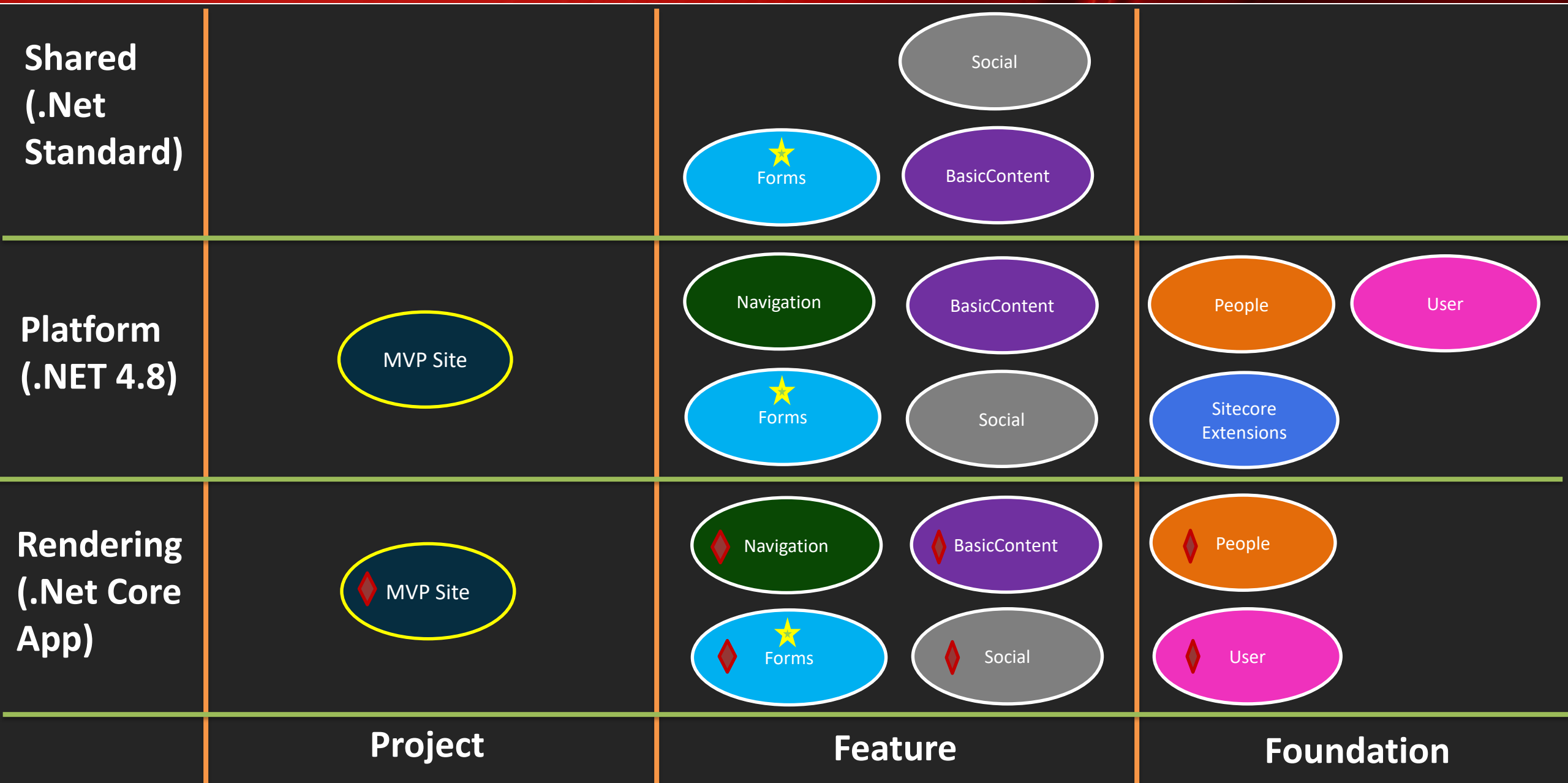


<https://doc.sitecore.com/xp/en/developers/100/developer-tools/sitecore-headless-development-conceptual-overview.html>

Sitecore ASP.NET Core Application Layers

Shared (.Net Standard)	<ul style="list-style-type: none">- Shared Model- Any shared logic
Platform (.NET 4.8)	<ul style="list-style-type: none">- Sitecore Pipeline Processor- Sitecore Layout Service Resolver- Any kind of patching logic- Schema Extender- Service / Application Logic- References Sitecore Layout Service
Rendering (.Net Core App)	<ul style="list-style-type: none">- Controller logic means APIs can be exposed- References other .NET core projects- As the name goes, views fall here- References Sitecore Layout Service Client- Cross-cutting calls must be through APIs as per headless principles

Sitecore ASP.NET Core Application Design



Sitecore ASP.NET Core Application Features

Feature	
Independent, render Sitecore Content	✓
De-coupled	✓
Light-weight	✓
Consumes Sitecore Layout Service	✓
Preview Code Changes	✓

Sitecore ASP.NET Rendering SDK

View types

Current version: 10.1

The Sitecore Rendering Engine renders Sitecore content using [ASP.NET](#) view components and partial views.

Generally, model bound views are likely be the ones you use most frequently. These are backed by a default Sitecore view component which binds component content to a provided type.

Custom view components are intended for use anywhere you have reusable rendering logic that is too complex for a model bound view. For example, navigation menus, login panels, and search bars.

Partial views are generally for lightweight components on a Sitecore page that do not contain their own content, such as structural components that contain just placeholders.

- [Model binding](#)
- [Model bound views](#)
- [View components](#)
- [Partial views](#)
- [Tag helpers](#)

Sitecore ASP.NET Rendering View Types

Model-bound View	Custom View	Partial View
Most-commonly used	Reusable	Structural
View backed by a model	Complex logic	Light-weight
	<ul style="list-style-type: none">• Navigation menus• Login panels• Search bar	Placeholders

Startup.cs Functionality

- **Register the Sitecore Layout Service Client invoked by the Sitecore Rendering Engine.**
- **Register the Sitecore Rendering Engine services.**
- **Map a Sitecore rendering component name to an implementation**
- **Forwarding of Scheme as X-Forwarded-Proto to the Layout Service**
- **Enable support for the Experience Editor**
- **Standard ASP.NET Core routing and static file support**

Service registration in Startup.cs

Model-bound View and Partial View Component:

```
public static RenderingEngineOptions AddFeatureBasicContent(this RenderingEngineOptions options)
{
    options.AddModelBoundView<AnnouncementBar>("AnnouncementBar")
        .AddModelBoundView<HalfWidthBanner>("HalfWidthBanner")
        .AddModelBoundView<ContentList>("ContentList")
        .AddModelBoundView<RichTextContent>("RichTextContent")
        .AddModelBoundView<RichTextContent>("SugconRichText")
        .AddModelBoundView<ImageTeaser>("ImageTeaser")
        .AddModelBoundView<VideoTeaser>("VideoTeaser")
        .AddModelBoundView<EmbedContent>("EmbedContent")
        .AddPartialView("ColumnContainer")
        .AddPartialView("PageOverviewPanel")
        .AddModelBoundView<HeroBig>("HeroBig")
        .AddModelBoundView<HeroMedium>("HeroMedium")
        .AddModelBoundView<SugconHero>("SugconHero");
}
```

Model bound View

Partial View

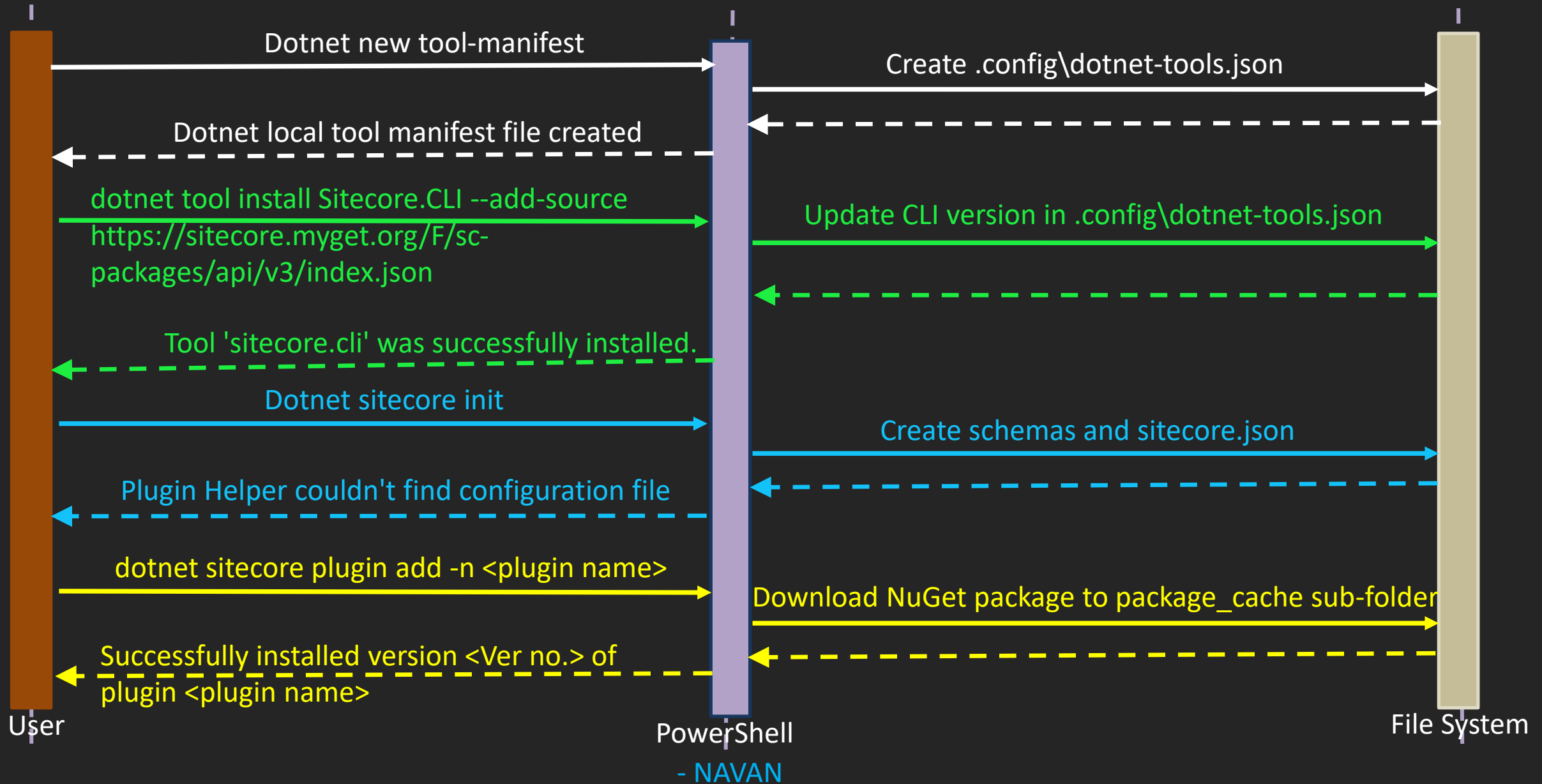
Custom View Component:

```
services.AddSitecoreRenderingEngine(options =>
{
    options
        .AddModelBoundView<ContentBlockModel>("ContentBlock")
        .AddViewComponent("MyService")
        .AddDefaultPartialView("_ComponentNotFound");
})
```

Register View Component
with custom logic

<https://doc.sitecore.com/xp/en/developers/100/developer-tools/walkthrough--creating-a-view-component-with-custom-logic.html>

Sitecore CLI Plugin Setup



Sitecore CLI - sitecore.json

- Specifies what CLI plugins were installed
- If plugins are not installed, commands would not be available
- The modules section specifies location of json files having serialization information
- If the module.json path is not specified here, serialization will not happen

```
sitecore.json  X
Schema: .sitecore\schemas\RootConfigurationFile.schema.json
1  {
2    "$schema": "../.sitecore/schemas/RootConfigurationFile.schema.json",
3    "modules": [
4      "src/**/*.module.json"
5    ],
6    "plugins": [
7      "Sitecore.DevEx.Extensibility.Serialization@4.0.0",
8      "Sitecore.DevEx.Extensibility.Publishing@4.0.0"
9    ],
10   "serialization": {
11     "defaultMaxRelativeItemPathLength": 100,
12     "defaultModuleRelativeSerializationPath": "items",
13     "removeOrphansForRoles": true,
14     "excludedFields": []
15   }
16 }
```

```
PS C:\cli400> dotnet sitecore index --help
index:
  working with indexes data

Usage:
  sitecore.cli index [options] [command]

Options:
  -?, -h, --help    Show help and usage information

Commands:
  schema-populate  Populates all managed schema
  list             Get all indexes list
  rebuild          Rebuild all indexes
  statistic         Get a summary of all indexes

PS C:\cli400> dotnet sitecore index list
Indexes list:
sitecore_core_index
sitecore_master_index
sitecore_web_index
sitecore_marketingdefinitions_master
sitecore_marketingdefinitions_web
sitecore_marketing_asset_index_master
sitecore_marketing_asset_index_web
sitecore_testing_index
sitecore_suggested_test_index
sitecore_personalization_index
sitecore_fxm_master_index
sitecore_fxm_web_index
sitecore_sxa_master_index
sitecore_sxa_web_index
PS C:\cli400>
```

Plugin and Command relationship

This PC > Local Disk (C:) > cli400 > .sitecore > package-cache > nuget

Name	Date m
Sitecore.DevEx.Extensibility.Indexing.4.0.0	23/09/2
Sitecore.DevEx.Extensibility.ResourcePackage.4.0.0	23/09/2

Sitecore CLI - module.json

- The module.json should be added to all folders where serialization must take place
- The module.json has item includes that specifies what path must be serialized from the Sitecore tree
- If no module.json is present in a folder or content tree path is not specified in module.json the concerned path is not serialized
- Dotnet sitecore ser pull used to serialize items to file system
- Dotnet sitecore ser push used to deserialize items from file system to Sitecore

Sitecore CLI - module.json

Structure:

```
{
  "namespace": "Project.MvpSite",
  "items": {
    "includes": [
      {
        "name": "layouts",
        "path": "/sitecore/layout/Layouts/Project/MvpSite",
        "allowedPushOperations": "CreateUpdateAndDelete"
      },
      {
        "name": "placeholders",
        "path": "/sitecore/layout/Placeholder Settings/Project/MvpSite",
        "allowedPushOperations": "CreateUpdateAndDelete"
      },
      {
        "name": "media",
        "path": "/sitecore/media library/MvpSite",
        "allowedPushOperations": "CreateOnly"
      },
      {
        "name": "forms",
        "path": "/sitecore/Forms/MvpSite",
        "allowedPushOperations": "CreateOnly"
      },
      {
        "name": "dictionary",
        "path": "/sitecore/system/Dictionary/MvpSite",
        "allowedPushOperations": "CreateOnly"
      },
      {
        "name": "templates",
        "path": "/sitecore/templates/Project/MvpSite",
        "allowedPushOperations": "CreateUpdateAndDelete"
      },
      {
        "name": "branche-templates",
        "path": "/sitecore/templates/Branches/Project/MvpSite",
        "allowedPushOperations": "CreateUpdateAndDelete"
      },
      {
        "name": "content",
        "path": "/sitecore/content/MvpSite",
        "allowedPushOperations": "CreateOnly",
        "rules": [
          {
            "path": "/MVP Repository",
            "scope": "ignored"
          }
        ]
      }
    ],
    {
      "name": "SPE",
      "path": "/sitecore/system/Modules/PowerShell/Script Library/MVP Scripts",
      "allowedPushOperations": "CreateOnly"
    }
  }
}
```


Sitecore CLI - module.json

Include properties

The include properties:

Property	Required	Valid values	Default value
name	Yes	String	None
path	Yes	String	None
scope	No	SingleItem, ItemAndChildren, ItemAndDescendants, and DescendantsOnly.	ItemAndDescendants
database	No	master or core	master
maxRelativePathLength	No	Number	130
allowedPushOperations	No	CreateOnly, CreateAndUpdate, and CreateUpdateAndDelete	CreateUpdateAndDelete

<https://doc.sitecore.com/xp/en/developers/101/developer-tools/sitecore-content-serialization-configuration-reference.html>

Sitecore CLI - module.json

Scope

The `scope` property configures *what* content items are influenced:

Scope	Content items influenced
<code>SingleItem</code>	Only the content item.
<code>ItemAndChildren</code>	The content item and its immediate children.
<code>ItemAndDescendants</code>	The content item and all its descendants.
<code>DescendantsOnly</code>	Only the descendants of the content item.
<code>Ignored</code>	<div>All children and descendants are ignored except those previously matched. Important Only valid in rules. Not valid in includes.</div>

Sitecore CLI vs Sitecore for Visual Studio

- Both tools are created by Sitecore and compatible with one another
- Although Sitecore for Visual Studio is licensed

Sitecore for Visual Studio

Current version: 10.0

The Sitecore for Visual Studio plugin acts as a graphical user interface for [Sitecore Content Serialization](#).

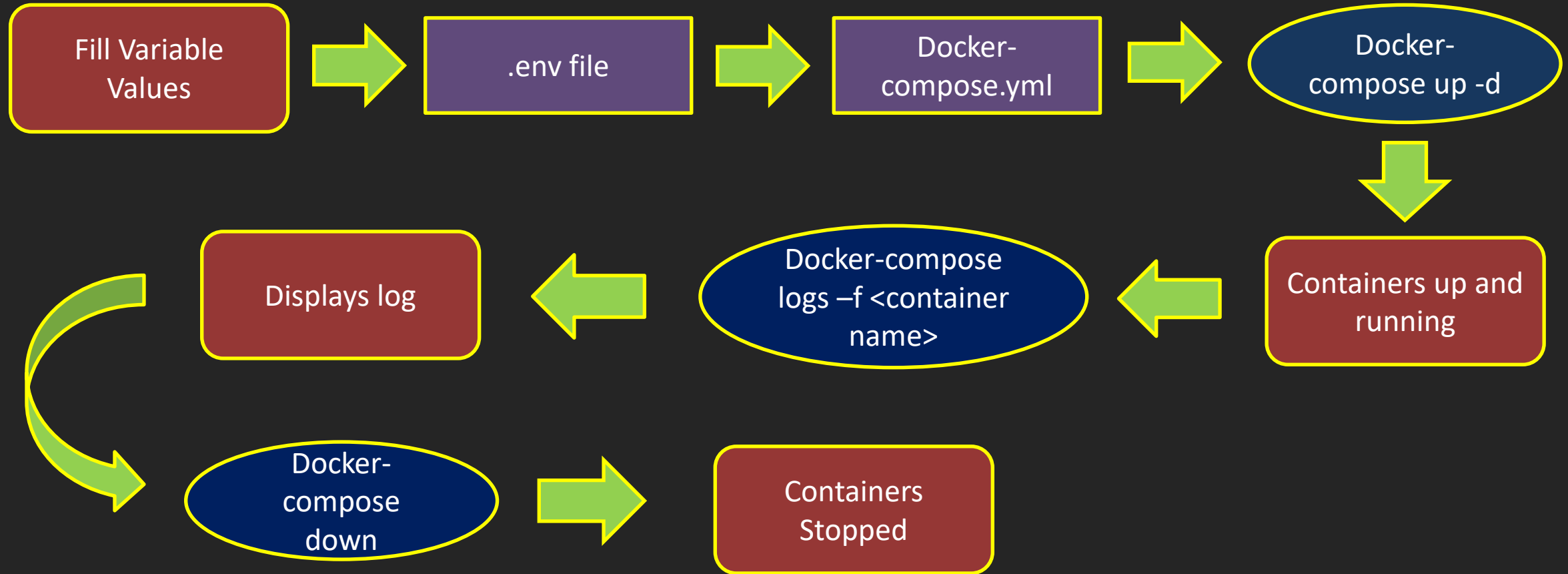
The plugin provides:

- Visual configuration of modules and environments.
- Visual synchronization of items, including conflict handling.
- Visual configuration of automated item synchronization at build time.

<https://doc.sitecore.com/xp/en/developers/100/developer-tools/sitecore-for-visual-studio.html>

Sitecore Docker Lifecycle

- The below flow describes general lifecycle of Sitecore Docker Container(s)



Tip: From Sitecore 10, Sitecore Container Registry provides base container images for production and development environments

.env file

- The **HOST_LICENSE_FOLDER** stores the location of license file while **SITECORE_LICENSE** variable stores the compressed license value
- If you receive a conversion error as part of compressing the license file (as below) then use **HOST_LICENSE_FOLDER**

Unhandled Exception: System.FormatException: The input is not a valid Base-64 string as it contains a non-base 64 character, more than two padding characters, or an illegal character among the padding characters. at System.Convert.FromBase64CharPtr(Char* inputPtr, Int32 inputLength) at System.Convert.FromBase64String(String s)

```
.env - Notepad
File Edit Format View Help
# details on how to populate these #
#####

HOST_LICENSE_FOLDER=C:\license
SQL_SERVER=mssql
SQL_SA_LOGIN=sa
SQL_SA_PASSWORD=0eCKYXJ02KfBcLLshbb
SITECORE_ADMIN_PASSWORD=b
SITECORE_IDSECRET=8vvH8mkWxM2w7Dm3IGHGAmXA9YyUijUo1L13QdhaaoVYhp50ZEUo4
SITECORE_ID_CERTIFICATE_PASSWORD=kl1KfKwU
SITECORE_ID_CERTIFICATE=MIKSQIBAzCCCgUGCSqGSiB3DQEHAaCCCFYeggnymIIJ7jC
+8rWH/10b3AANTMSOV7a//LO1pZG2Qzw1kHSjd8ZFq6snVXx2a328+fbCHobHKYw1tYJtt6
+P2zUsfMqIsbIw06EkAfxkbtQzR5+Vh5RUggu20AYTL7RTDEFD294xYZCmfUj1t5dQp0HKQ
+X89a07nr0cM061tToZaqbtZJIEDA2BKHMJ0xkMfbNTseij17o6mSHggCvkxr9ed88rQrEd
+0qOwDUMETKc9eAsJ/ETBgZAm1ou2Vzt/QdWituVFrXXTKud8x58eNLvFL9B7BPakYxOXJ
+OfEPq6eQ2RRwr1rvW7wEHLK/7SL185Wuz3IygVNa1161zrx9U4qGGM4qu0Fjr6oL4gPmz9
+ietqLk0NH3BAyqkrn61dPiA4WLK9yGTIFLURPonXUcKwxAnktGoMgFoYGYxpWQ2/VFGKOk
+Vw6yv/uL2yJhLYENCmV6AjGmgt4jAywkbboy1kT21EsUkvkWDNe+G4Q1/dn9jHFR78JEJJ
+1jySnPeoFfVbbhHa9qKWGbVNps3NrDDgGHxvJ4/HyFZQIEHFkRj1jy8S0Uxp1DBC8ZJQY
+FGzCkBD5Up3+miEWp0Wz3U6wsIHBswIEaXvK7xVmv+O3bo5/dGiZz9yKhGQ+GijIP0jPC+
+Lr1PyAkzRNBq8stJEdz/MM0MHsoOpeuA9A13ZutqL2UdYoX55SghC2Xk0I8jGB4jANBgkr
A2MF0GCSsGAQQBgjcRATFQhk4ATQBpAGMAcBvAHMAbWBMhAQIAIABTAG8AZgB0AHcAYQByA
43yGfUg2N2Fep5+1ts6Sto4NKHPg3WxOwP3jEcjHwrKlwMqAI3mDtG067i/yBFSe52ES5Rn
+duFPoKzo13BVLGtH4c4cvtkwqc4d4RX2/QQJkUCLgxWHDvn/ZCF5ZYB5z93yxSQ995fhhK
+z1XF2IGtKWy3cL3bV0AFHyvB1mUraTE/yJIRo82vbrH736CvN7v6Kue5Lhg5C0xRD1pgiP
+mBo48oWdBIhQVjDVKL1UfFadQ1R4EsBnDRX4FwM+AkMZc8rCYYYVa5p9Wq43uyRnrwWfX
+FBtKEqWJwhB8sUqWylLWkSFzrk6G1GRna69R15PdJ5VgNHXRSQJPPaqA5YNNKNsEkDx/ae
+LL3glfXs4TjuzfQI8tP5UwdAvx66+anP7vZ2eYa/6aeBzb4EydjVTPwUvF563cEN4tJbP6
e3cxNOVQEHZjAaKYJ1ItuCUv7CIP5Cnf4NHKzdJB+DNzRFaE+0Uub0US/UAc0121zrxUr8j
+nLiUzF4dFYTARDnj8o2wbNAAyDBMuarVhTwSMYair19euBs1lyQtYQWe6zRjKcY4iFSaoQ
TELRIK_ENCRYPTION_KEY=V;2Yb+39-?@uL8[?Q;Nj]tF[rjJpi@`H/r^!/3H4!LU:1J!w
SITECORE_APPLICATION_USER_NAME=admin
SITECORE_APPLICATION_USER_PASSWORD=b
SITECORE_APPLICATION_USER_DOMAIN=sitcore
SITECORE_Application_CMS_URL=http://cm

#####
# Environment variables that can be changed. #
# #
# The solution should build & run with the #
# default values below, but you can update them #
# if you need to. #
#####

SITECORE_DOCKER_REGISTRY=scr.sitcore.com/sxp/
MVP_DOCKER_REGISTRY=
SITECORE_VERSION=10.2-1tsc2019
TRAFFIK_ISOLATION=hyperv
ISOLATION=default
BUILD_CONFIGURATION=debug
SITECORE_LICENSE=
```

<https://sitecore.stackexchange.com/questions/25991/identity-server-container-fails-to-launch-due-to-license-error-using-headless-ge>

Questions & Answers

