

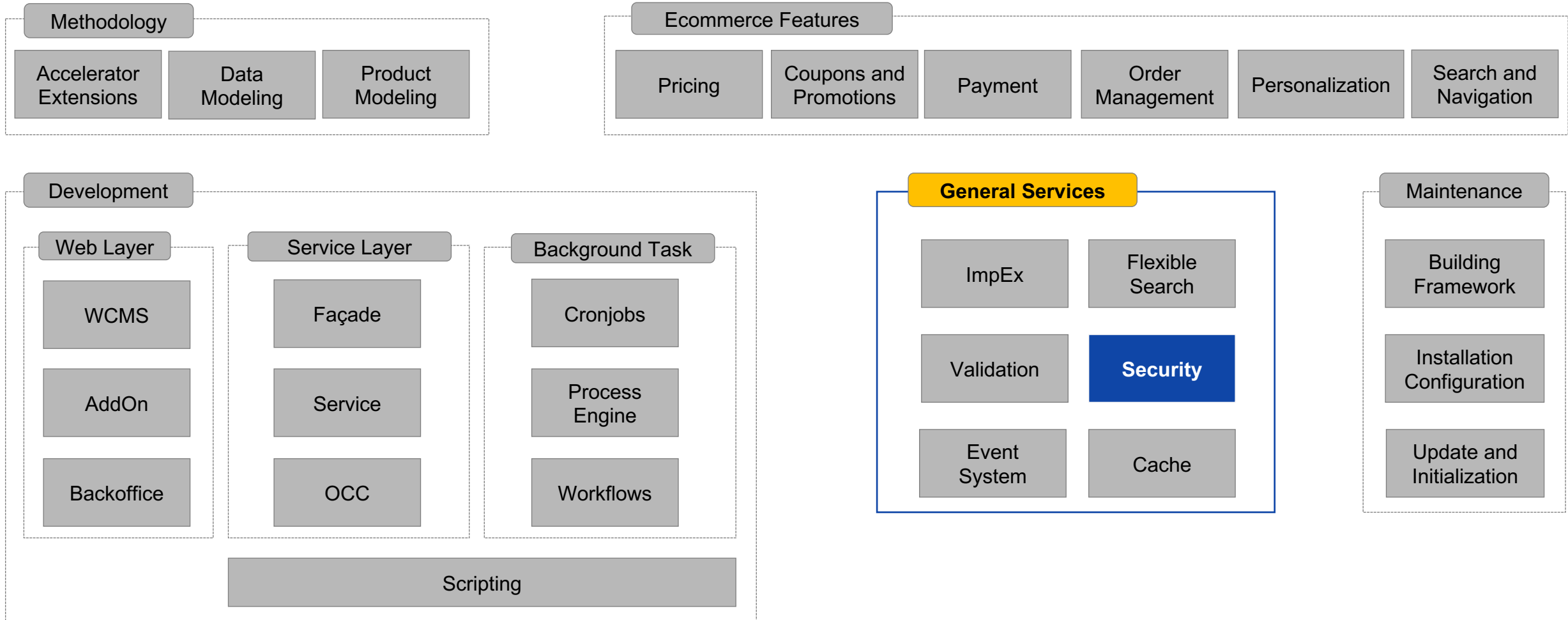


SAP Customer Experience

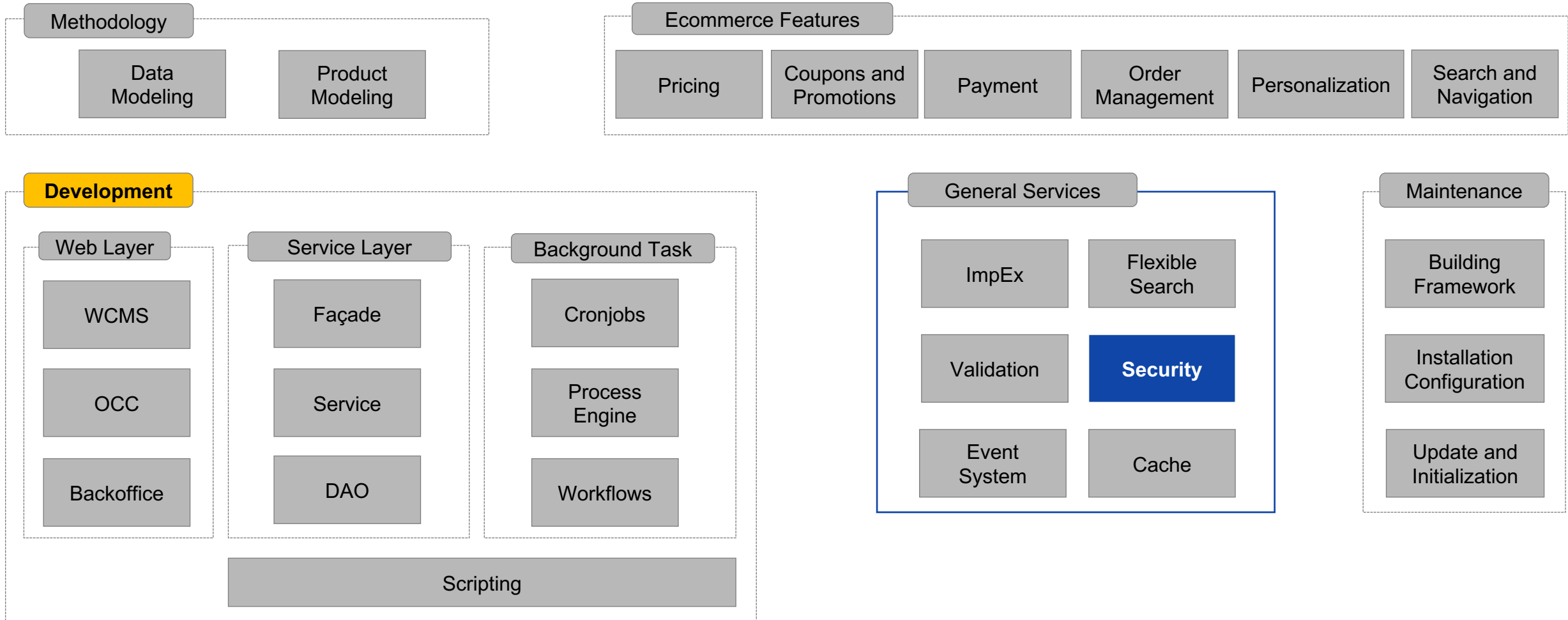
Security

INTERNAL – SAP and Partners Only

What we will cover in this topic




What we will cover in this topic



We will learn about:

- Basics
- Type-Based Access Rights
- Item-Based Restrictions
- Spring Security
- Additional Security Features

The Context

-  In SAP Commerce Cloud, **type-based permissions** and **item-based restrictions** can be assigned to **users or user groups**.

Basics

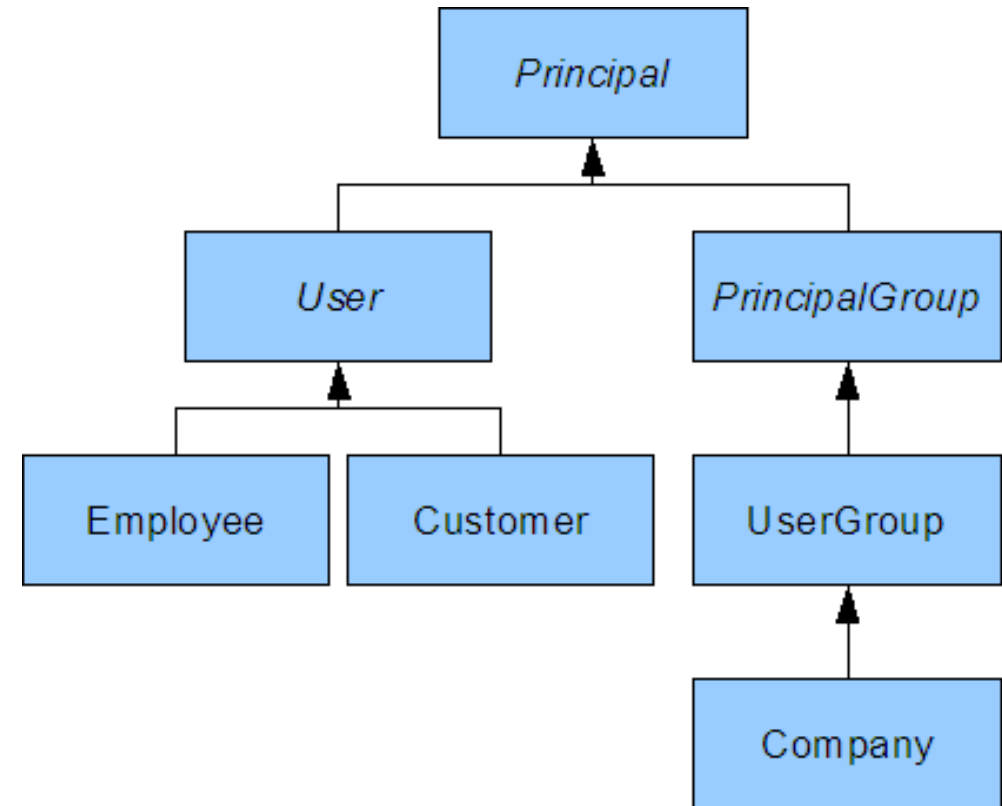


Security areas to consider

- Web access control
 - IP range
 - Spring security per web app
- Administration rights
 - HAC access
- Data permissions
 - Role-based
 - Type and Item
- Database security
 - Transparent symmetric encryption
 - Field encryption
 - Limiting DB user rights

Basics

- User accounts in SAP Commerce Cloud can be **individual people, user groups/roles**:
 - who is allowed or not allowed to **authenticate** against a part of an application
 - who is allowed or not allowed to perform **specific tasks**
- **PrincipalGroup** defines **UserGroup**, **Company**
 - Company: Unlike UserGroups, Companies can hold addresses
- **User** defines: **Employee**, **Customer**
- Default user accounts and groups that cannot be removed:
 - **anonymous** (Customer)
 - **admin** (Employee)
 - **admingroup**



Where Do User Accounts Affect SAP Commerce?



Backoffice

Displays (or hides/disables) elements, depending on the user groups in which a user belongs.



Workflows

All GUIs (e.g., Backoffice or SmartEdit) have workflow integration and allow users to manage workflow steps



Web Services

Allows user-specific access rights for both Omni Commerce Connect (OCC) and Integration API Module



User Accounts



JaloSession

At any given time, a user must be assigned to a JaloSession



Order Process

A customer must be associated with a shopping cart, either by logging in, or as a guest.



CronJobs

Use JaloSessions and therefore require users to be assigned

Type-Based Access Rights



Type-Based Access Rights – Overview

- Access rights for SAP Commerce Cloud **types** and their **attributes**
- Access is granted to **individual users** and/or **user groups**
- Affect the entire type, not individual items
- Affect individual type attributes
- **Effective** in Backoffice and web services
- The specific type or attribute will be **disabled** (greyed out) to the user account

Permission Management - User Group ([employeeegroup])

Filter Context					Filter Attributes		
<input type="text"/>					<input type="text"/>		
<div> </div>							
Context	Read	Change	Create	Remove	Attribute	Read	Change
Audit Report Data	✗	✗	✗	✗	Active catalog version	✓	✗
C2L Item	✓	✗	✗	✗	Agreements	✓	✗
Catalog	✓	✗	✗	✗	Base Stores	✓	✗
Catalog version	✓	✗	✗	✗	Buyer	✓	✗
Cx Mapper Script	✗	✗	✗	✗	Catalog Version	✓	✗
Employee	✓	✓	✗	✗	Catalog Versions	✓	✗
Enumeration Value	✓	✗	✗	✗	Comments	✓	✗
					Default catalog in the system	✓	✗
12 items					< 1 / 2 > 35 items		

✗ Inherited ✗ Denied ✓ Granted

Advantages	Disadvantages
Attribute-based	Affect the entire type, not just individual instances
Can be imported from and exported to Impex easily	Not effective everywhere (for example, on the ServiceLayer)

Importing via ImpEx

- Type access configuration can be imported through ImpEx

```
$START_USERRIGHTS
Type;UID;MemberOfGroups;Password;Target;read;change;create;remove
UserGroup;productManagerGroup;employeeGroup;;;
# Access Rights for Products & Catalog
;;;Product;+;+;+;+
;;;Product.ean;+;-;-;-
;;;Catalog;+;;;
;;;Media;+;+;+;+
$END_USERRIGHTS
```

- Full syntax:
 - [User Rights on //help.sap.com](https://help.sap.com)

API CRUD example

- Generic service for checking permission assignments:

```
permissionCheckingService.checkTypePermission(typeCode,  
  
PermissionsConstants.REMOVE).isDenied();
```

- For typical CRUD permission checking use:

`PermissionCRUDService` – a wrapper over `PermissionCheckingService`

```
permissionCRUDService.canReadType( typeCode );  
permissionCRUDService.canChangeAttribute(typeCode, attributeQualifier );
```



More information can be found in the live session: **Custom Access Rights** in the live session series [“SAP Commerce Cloud - Additional Technical Essentials”](#)

Item-Based Restrictions



Item-Based Restrictions – Overview

- Defined using **SearchRestriction** items
 - Called Personalization Rules in Backoffice (*System* → *Personalization*)
 - Restrictions define a **filter** which is added to FlexibleSearch statements at execution time
 - for the **specified type**
 - for a **user** or a **user group**.
 - System-wide effect
 - The restriction is automatically added to the WHERE clause of a FlexibleSearch statement

The screenshot shows the configuration interface for a SearchRestriction item. It includes the following fields and options:

- Active***: Radio buttons for **True** (selected) and **False**.
- Identifier***: A text field containing `Frontend_ProductApprovalStat`.
- Properties**: A section header with an upward arrow icon.
- Name**: An empty text field.
- Filter***: A text area containing the JSON filter `{approvalStatus} = 8796100722779`.
- Restricted Type***: A dropdown menu showing **Product [Product]**.
- Apply on***: A dropdown menu showing **[customergroup]**.

Advantages

- Automatically affect every FlexibleSearch
- Can block access to individual type instances

Disadvantages

- Requires knowledge of FlexibleSearch syntax
- May require extended SAP Commerce data model knowledge

ImpEx example

SearchRestriction items can be imported like any other item, using ImpEx:

```
INSERT_UPDATE SearchRestriction;code;principal(UID);restrictedType(code);active;query  
;FrontRestriction;customergroup;Product;true;{catalogVersion} IN (?session.catalogversions)
```

Spring Security



Spring security – Overview

- Spring security framework takes care of:
 - Restricting access
 - Delegating authentication and authorization
 - Remember me services, login pages etc.
- Spring security framework is used in:
 - SmartEdit
 - SAP Commerce Administration Console (aka. HAC)
 - Omni Commerce Connect (aka. OCC, or SAP Commerce RESTful web services) API
 - Accelerator websites
- Each web application has a separate spring security configuration, e.g.:
 - spring-security-config.xml in platform/ext/hac/web/webroot/WEB-INF/config
 - or security-spring.xml in commercewebservices/web/webroot/WEB-INF/config/common

Spring security – Authentication & Configuration

- **For consistent authentication** across all applications, use Commerce-provided `CoreAuthProvider` (using the Spring Security bean `coreAuthProvider`)
 - Customize authentication: extend `CoreAuthProvider` and wire into Spring Security

```
@Override
public Authentication authenticate(...)
{
    User user = getUserByLogin( userDetails.getUserName() );
    Object credential = authentication.getCredentials();
    ... //verify - compare
}
```

- **To configure**

- Use 'security' xml namespace

```
<security:intercept-url pattern="/my-account*"
                        access="hasRole('ROLE_CUSTOMERGROUP')"
                        requires-channel="https" />
```

Spring security – Application in HAC Using Roles

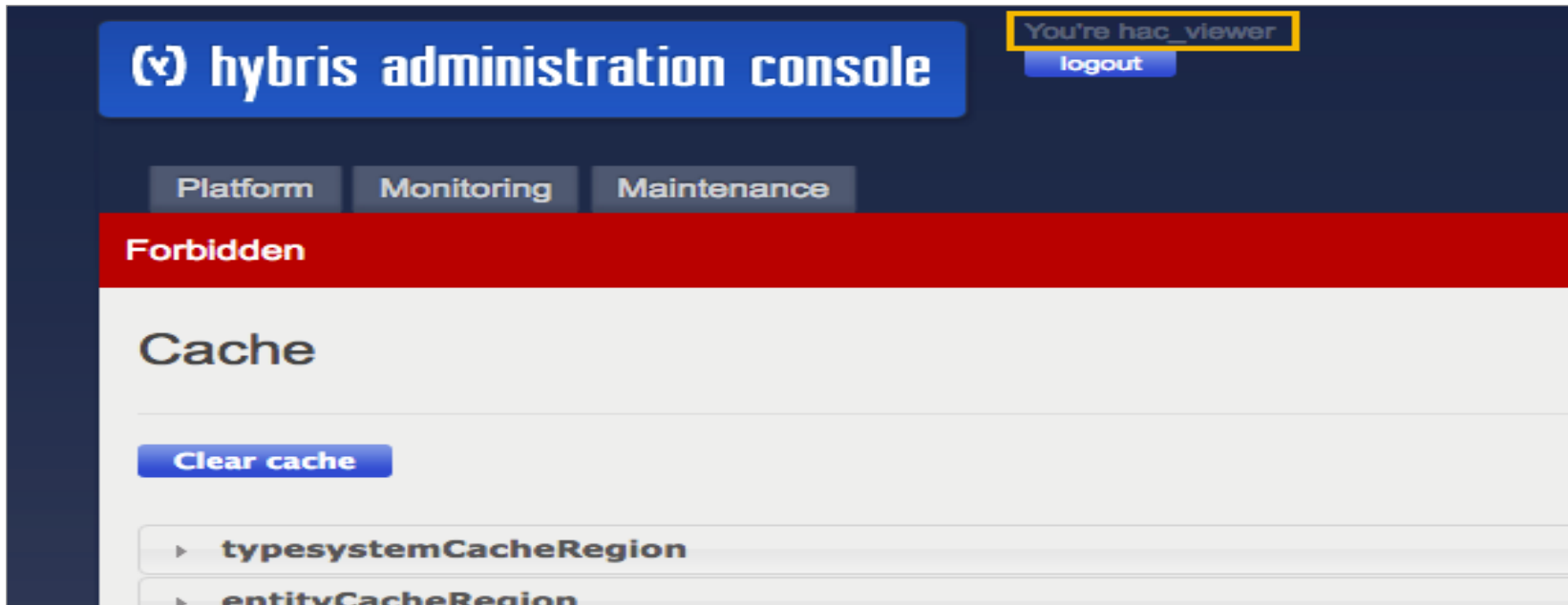
- Configure user access to different areas in the SAP Commerce Administration Console:
 - Based on assigned roles, users have access to specific HAC tabs or actions
 - The HAC provides multiple predefined roles OOTB
 - You can configure your own roles using Spring Security
 - Roles are represented as userGroup entries in the database
 - Roles are imported as essential data during system initialization or update
 - Each role, and the urls it grants access to, are configured in the `spring-security-config.xml` file (using the tag `intercept-url`)

Spring security – Application in HAC Using Roles - Examples

- Only users with the ROLE_HAC_MONITORING_CACHE role can clear the cache
 - configured in spring-security-config.xml of the hac extension:

```
<intercept-url pattern="/monitoring/cache/**/clear"  
    access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_CACHE"/>
```

```
<intercept-url pattern="/monitoring/cache/**"  
    access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_CACHE, ROLE_HAC_MONITORING_CACHE_LIMITED"/>
```



Additional Security Features



Predefined Access Rights and Restrictions

- Out-of-the-box, SAP Commerce Cloud provides:
 - some user roles/groups with predefined type-based access rights, e.g.:
 - employeecgroup for basic access to catalogs, catalog versions, etc.
 - cmsmanagergroup to work with CMS content.
 - backofficeworkflowusergroup to work with workflows related to product content management.
 - and more...
 - a special set of predefined restrictions, e.g.:
 - Frontend_ restrictions to enable/limit customers to view specific content on the frontend.
 - Backend_ restrictions to enable/limit employees to access specific items in Backoffice.
 - Sync_Only_Approved restrictions to allow synchronizing only approved CMS content.
 - And more...

Advantages	Disadvantages
Allows permission management for many common use cases	Not very generic approach
Can be used as a starting point for further configuration	

Access Control for Catalog Versions and Languages

SAP Commerce Cloud also provides special access control regarding:

- Catalog versions
 - Made readable and writeable for certain user groups (such as catalogmanagergroup)
- Languages
 - Made readable and writeable for certain user groups that can have read and write access to several languages

Electronics Product Catalog : Online

Catalog version Content Catalog Versions **Permissions** Administration

Essential

Catalog* Catalog Version*

Electronics Product Catalog Online

Permissions

User accounts with write permission User accounts with read permission

Administrator [admin]	Administrator [admin]
CMS Manager Group [cmsmanagergroup]	[employeeegroup]
Backoffice Workflow User Group [backoffi...]	Base CMS Manager Group [basecmsman...]
Backoffice Product Manager [backofficepr...]	CMS Manager Group [cmsmanagergroup]
Select from List ...	[marketingManagerGroup]
	Select from List ...

German CMS Translator Group [cmstranslatorgroup-DE]

General **Languages** Prices Personalization Administration

Essential

ID* Name

cmstranslatorgroup-DE

Languages

Users are only allowed to see / edit content in the languages listed here.

Readable Languages Writeable Languages

German [de]	German [de]
Select from List ...	Select from List ...

Password Security Policies

SAP Commerce Cloud allows fine control of password handling via:

- **PasswordPolicyService:**

- is used every time a password is set or changed. (e.g., used by UserService)
- validates user password against predefined **password security policies** and returns a list of PasswordPolicyViolation objects if validation fails.

- **Password security policies:**

- define requirements that must be met when setting or changing a password.
- include OOTB the **regex** and **blacklist** security policies.
 - Directly configurable via properties.
- can be customized with own password security strategies
 - by implementing the PasswordPolicy interface.

Password Change Auditing

- Register all the changes made to a user password
- **UserPasswordChangeAudit** is an item type and therefore traceable in the Backoffice
- **UserPasswordChangeAuditPrepareInterceptor** provides logic for recording password changes

The screenshot displays the SAP Backoffice interface for Password Change Auditing. On the left is a navigation menu with categories like System, Search and Navigation, Catalog, and WCMS. The main area is titled 'Types' and features a search bar and a table of audit records. The table has columns for ChangingApplication, ChangingUser, Time Created, Encoded Password, Type, Time Modified, Owner, and Password Encoded. One record is highlighted, showing a password change for user 'admin' on March 28, 2023, using the 'UserPasswordChangeAudit' type and 'plain' encoding. Below the table, the details for the selected record 'UserPasswordChangeAudit[8796093054982]' are shown, including its metadata such as PK, Type, Time created, Time modified, and Owner.

ChangingApplication	ChangingUser	Time Created	Encoded Pas...	IpA...	Type	Time Modified	Owner	Password Enc...
<input type="checkbox"/>	admin	Mar 28, 2023 10:25:02 ...	*****		UserPasswordChangeAudit	Mar 28, 2023 10:25:02 ...		plain
<input type="checkbox"/>	admin	Mar 28, 2023 10:25:02 AM	*****		UserPasswordChangeAudit	Mar 28, 2023 10:25:02 AM		plain
<input type="checkbox"/>	admin	Mar 28, 2023 10:25:22 AM	*****		UserPasswordChangeAudit	Mar 28, 2023 10:25:22 AM		pbkdf2
<input type="checkbox"/>	admin	Mar 29, 2023 2:43:19 PM	*****		UserPasswordChangeAudit	Mar 29, 2023 2:43:19 PM		pbkdf2

0 items selected 4 items

UserPasswordChangeAudit[8796093054982]

Metadata

PK*	Type	Time created	Time modified
8796093054982	UserPasswordChangeAudit	Mar 28, 2023 10:25:02 AM	Mar 28, 2023 10:25:02 AM

Owner

oauth2 Extension

- Enables access tokens instead of passwords to protect resources.
- No need to be enabled explicitly (i.e. no need to add the oauth2 extension to `localextensions.xml`), because it's part of the platform extensions.
- Exposes the HTTP endpoints as authorization server with 2 endpoints:
 - `/authorizationserver/authorize`
 - `/authorizationserver/token`
- Configure dedicated oauth2 properties in `local.properties`.
- Manage OAuth clients and access tokens using the `System/OAuth` tab in the Backoffice.

Key Points

1. SAP Commerce uses **type-based access rights** to grant access rights at the **type** and **attribute** level for different users or user groups.
2. Type-based access rights normally only affect Backoffice and web services; if you want to apply them in your code, use the `PermissionCheckingService`.
3. A **SearchRestriction** will add a search condition to a `FlexibleSearch` statement when the current user/usergroup and type match the user/usergroup and type specified in the restriction.
 - Defined in Backoffice or ImpEx
 - Restrictions work at the instance level and have a **system-wide** effect!
4. **Spring security** restricts web access, and is configured individually for each web application

References 1/2

- Documentation about Principals:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/8c797311866910149debcebf02be567.html

- Type-Based Access Rights:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/8b4aa00e866910148df2920f69d68b27.html

- Item-Based Access Rights (Search Restrictions):

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/8c428f8286691014970ceee87aa01605.html

References 2/2

- Password Security Policies:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/dfeec89a46c64774892b46936d65d530.html

- Password Change Auditing Feature:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/2e18017332f444599286e62cac5f9ab9.html

- oauth2 Extension:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/aa417173fe4a4ba5a473c93eb730a417/3d3ea6a4d5fa486aa324ce278fa2afc3.html

- Securing HAC Using Roles:

- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/9b5366ff6eb34df5be29881ff55f97d2/5bfac051a46e4c17ac79c738ff739270.html

Security Exercise



Thank you.