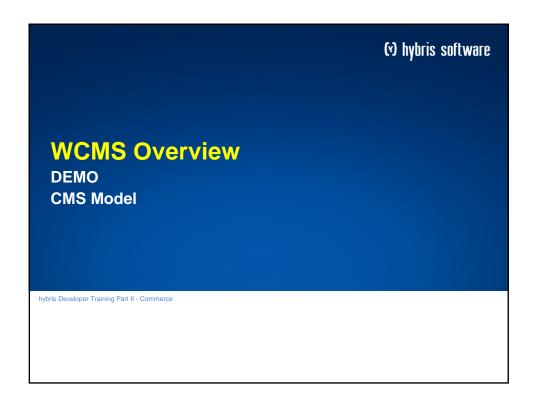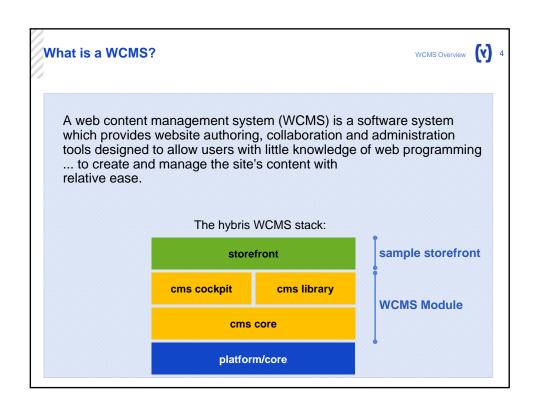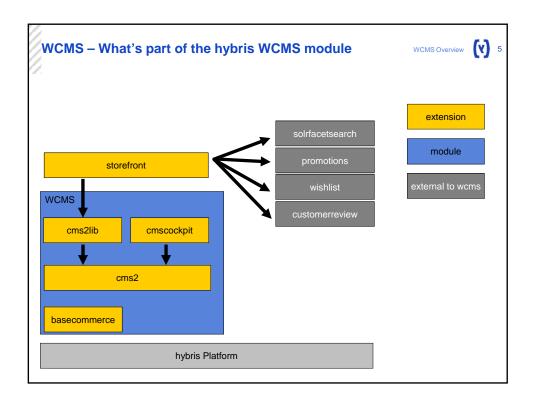(v) hybris software

THE FUTURE OF COMMERCE

(v) hybris software

hybris Developer Training Part II - Commerce

WCMS

(ʏ) hybris software

WCMS Overview
DEMO
CMS Model

hybris Developer Training Part II - Commerce

What is a WCMS? WCMS Overview (ʏ) 4

A web content management system (WCMS) is a software system
which provides website authoring, collaboration and administration
tools designed to allow users with little knowledge of web programming
... to create and manage the site's content with
relative ease.

The hybris WCMS stack:

storefront	sample storefront
cms cockpit — cms library	WCMS Module
cms core	
platform/core	

WCMS – What's part of the hybris WCMS module

extension	
module	
external to wcms	

storefront

WCMS

cms2lib

cmscockpit

cms2

basecommerce

solrfacetsearch

promotions

wishlist

customerreview

hybris Platform

WCMS: key features

➡ **W**ebsite**s** **C**ontent **M**anagement **S**ystem

➡ Management of all kinds of pages
➡ homepage, information pages, product, category or search pages

➡ Content personalization
➡ Based in Context Information
➡ Like current user, group, time, product, category

➡ Built-in live editing and previewing
➡ Change contents on the fly
➡ Testing with different context information

WCMS: key features

➡ Classic components management:
  ➡ paragraph, banner, link, etc.
➡ Common templates for pages
➡ Independent of any frontend technology

hybris WCMS: The basic model

Page Templates

2 column standard   3 column product   brand template   ...

Pages

Homepage   Company Profile   Eizo Monitor

Components

text   image   link list   text with image   image gallery   ...

The hybris WCMS stack: electronics store

WCMS Overview 9

Navigation

Banners

Footer

Cart Summary

Rotating Images

Product Carousel

The hybris WCMS stack: CMSCockpit

WCMS Overview 10

➡ Shows an abstracted view of the layout

➡ Add components

Navigation

Banners

Cart Summary

Rotating Images

Product Carousel

(Y) hybris software

WCMS Overview
DEMO
CMS Model

hybris Developer Training Part II - Commerce

Time for a demo..   WCMS Demo (Y) 12

Demo will show :
➡ The CMS Cockpit
➡ Change contents on the fly
➡ Locking / synchronization
➡ Add new components
➡ Live edit for testing
➡ Personalized Pages

(Y) hybris software

hybris software

WCMS Overview
DEMO
CMS Model

hybris Developer Training Part II - Commerce

The data model..

CMS Model ⊙ 14

➡ Zoomed in version to follow !

➡ A "view model" –a data structure which represents the fundamental model of a page view.

➡ Also contains catalogs and restrictions

The data model..

platform
Catalog

basecommerce
Base Store

CMSSite — Content Catalog

ContentSlotName

CMSItem

CMSRelation

PageTemplate — ContentSlotForTemplate

ContentSlot — ContentSlotForPage

AbstractPage

Custom

CMSItem

AbstractCMSComponent

Abstract Restriction

Platform
Catalog
Product
User
UserGroup
Category

CMSCatalog Restriction
CMSProduct Restriction
CMSTime Restriction
CMSUser Restriction
CMSUser Group Restriction
CMSCategory Restriction
Custom

ContentPage
CategoryPage
ProductPage
CatalogPage

CMSNavigationNode

SimpleCMSComponent
AbstractCMSComponentContainer

CMSLink Component
CMSImage Component
CMSParagraph Component
Custom

WCMS – CMSSite

platform
Catalog

basecommerce
BaseStore — CMSSite — ContentCatalog

➡ Assigned to one or more BaseStores
➡ BaseStore defines the product/classification catalogs
➡ ContentCatalog defines the content catalogs
➡ Allows Content Versioning
➡ CMSSite as starting point

WCMS – CMS Components

AbstractCMSComponent

SimpleCMSComponent — AbstractCMSComponentContainer

CMSLinkComponent   CMSImageComponent   CMSParagraphComponent   Custom

➡ CMS Components represent the smallest piece of information that can be displayed on a page

WCMS – PageTemplates

CMSItem

PageTemplate

1..*   1..*   1

ContentSlotName   ContentSlotForTemplate   ContentSlot

➡ **PageTemplate** defines **ContentSlotNames** which are the placeholders for content which applies to all pages with that template

➡ **PageTemplates** can be assigned with concrete **ContentSlots**
(via the typed relation: **ContentSlotForTemplates**)

➡ Pages with concrete ContentSlots will display that content automatically. Content is linked, not copied

WCMS – Pages

- A Page represents a single web page
- Every Page is bound to exactly one PageTemplate
- ContentSlots are the containers for Components
- ContentSlots are assigned to the page via the typed relation ContentSlotForPage

WCMS – CMS Restrictions

- Restrictions can be bound to pages and components

Step 1

➡ Type definition

```
<itemtype code="CMSWeekDayRestriction"
          extends="AbstractRestriction"  ...
```

Step 2

➡ Restriction evaluator implementation

```
public class CMSWeekDayRestrictionEvaluator
    implements CMSRestrictionEvaluator<CMSWeekDayRestrictionModel>
{
    @Override
    public boolean evaluate(
        final CMSWeekDayRestrictionModel weekDayRestriction,
        final RestrictionData context)
{
...
```
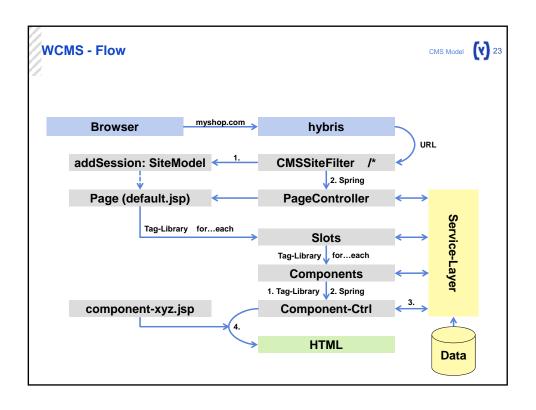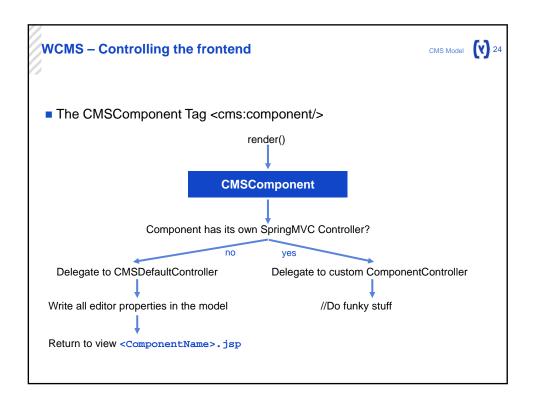

Step 3

➡ Evaluator bean definition

```
<bean id="cmsWeekDayRestrictionEvaluator"  ...
```

Step 4

➡ Evaluator mapping bean definition

```
<bean id="cmsWeekDayRestrictionEvaluatorMapping"
      class="de.hybris.platform.cms2.servicelayer.services.
             evaluator.CMSRestrictionEvaluatorMapping"
      scope="tenant">
  <property name="restrictionTypeCode"
            value="CMSWeekDayRestriction" />
  <property name="restrictionEvaluator"
            ref="cmsWeekDayRestrictionEvaluator" />
</bean>
```

WCMS - Flow

```
Browser  --myshop.com-->  hybris
                                    | URL
addSession: SiteModel  <--1.--  CMSSiteFilter  /*
        |                            | 2. Spring
Page (default.jsp)  <--  PageController  <-->  Service-Layer
        | Tag-Library  for…each
                            Slots  <-->
                            | Tag-Library  for…each
                            Components  <-->
                            | 1. Tag-Library  2. Spring
component-xyz.jsp  -->  Component-Ctrl  <--3.-->
                   4.
                        HTML

                        Data
```

WCMS – Controlling the frontend

■ The CMSComponent Tag <cms:component/>

render()

CMSComponent

Component has its own SpringMVC Controller?

no   yes

Delegate to CMSDefaultController   Delegate to custom ComponentController

Write all editor properties in the model   //Do funky stuff

Return to view `<ComponentName>.jsp`

Controlling the frontend

➤ A component is the smallest unit of content we can provide

➤ If we disregard the Site / Page / Slot call stack, and just create the URL for an individual component we get this (for example):

```
http://electronics.local.9001/store/view/MinicartComponentC
ontroller?componentUid=MiniCart
```

➤ This URL calls a MiniCartComponentController directly

➤ Note how the shorter lifecycle shows the content delivered for one "piece of content"