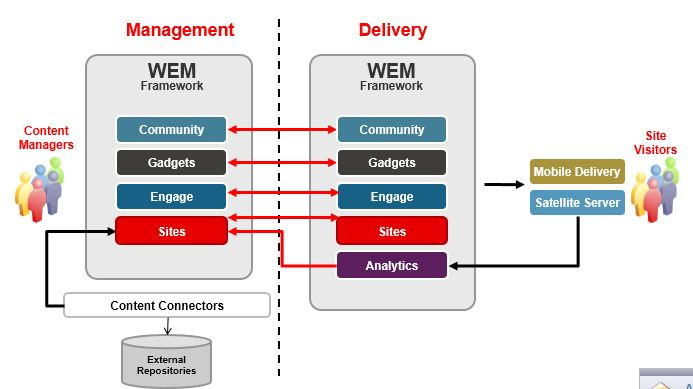
**ORACLE WCS**

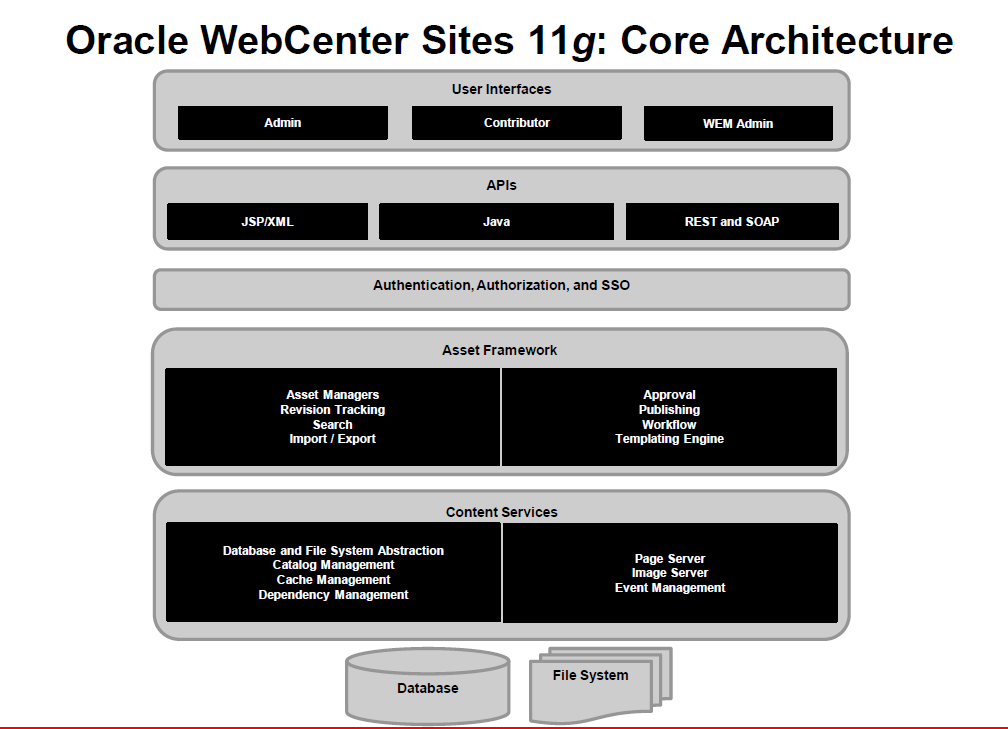
* Oracle WebCenter Sites is a high-performance, large-scale content management and delivery system used to create and manage complex websites.
* It provides an interactive and personalized experience for site visitors, delivering up-to-date content relevant to their interests.

## **Oracle WebCenter Sites 11g: Product Architecture**



* WebCenter Sites includes various supplemental offerings such as Community Management, Gadgets, Engage, Content Connectors, and Analytics.
* The architecture of WebCenter Sites consists of different environments like Development, Management, Testing, and Delivery, each serving a specific purpose.

## **Oracle WebCenter Sites 11g: Core Architecture**



* The core architecture of WebCenter Sites comprises user interfaces, APIs, authentication and authorization features, asset framework, content services, storage, and data modeling.
* WebCenter Sites supports two data models: basic asset model and flex asset model, allowing for hierarchical and flat data structures.

## The main WebCenter Sites servlets are as follows:

**ContentServer**: Generates and serves pages dynamically. This servlet provides disk caching, session management, event management, searching, and personalization services.

**CatalogManager**: Provides most of the database management for the WebCenter Sites database, including revision tracking, security, resultset caching, and publishing services.

**TreeManager**: Manages the tree tables, which store hierarchical information about other tables in the WebCenter Sites database.

**BlobServer**: Locates and serves binary large objects (blobs). Blobs are not processed in any way. They are served as is, as they are stored.

**DebugServer**: Provides tools that help you debug your XML code.

**CookieServer**: Serves cookies for WebCenter Sites pages, whether those pages are delivered by the ContentServer servlet or by the Satellite Server application.

**HelloCS**: Displays version information about the WebCenter Sites software installed on your system.

* WebCenter Sites provides utilities such as Developer Tools, Sites Explorer, CatalogMover, XMLPost, BulkLoader, and Property Editor for managing the database and code.
* There are three main interfaces: Admin Interface for developers and administrators, Contributor Interface for content providers and business users, and WEM Admin Interface for centralized application management and user authorization.
* WebCenter Sites is a content-centric content management system, where content is managed separately from the web pages and dynamically composed for site visitors.

# **Fundamental concepts of WCS.**

* WCS is a content management product that has several layers.
* The first layer is Content Server which is an XML based web programming language similar to ColdFusion.
* Content Server was further extended adding to the XML based programming language a more powerful and more standard JSP based programming language.
* Xcelerate is built on top of ContentServer and has a web user interface. It implements the more important concepts of WCS: the Asset.
* Gator is a further extension of Xcelerate that basically implements the concept of Flex Asset.
* Engage (once called “Marketing Studio”) further extends the feature set of the system, providing the ability to store attributes of the user (Visitor Attributes), build rules based on them (Segments) and generate content dynamically.

# **WebCenter Sites as a Web Content Management Framework**

* The Sites Content Management System is the core of the product and was built from the ground up specifically to meet the wide demands of web content management.
* It’s a different animal from WebCenter Content in that it’s not designed for Enterprise Content/Document Management.
* The asset modeling in Sites allows you to tag, reference, and interconnect your content more completely than ever before.
* With the addition of TEAM’s Sites Connector your WebCenter Content repository can communicate and share content with Sites making disparate repositories far more unified.
* Community for Sites provides a suite of functionality that overlays your whole Sites presence. More than just commenting, Community provides reviews, ratings, polls, moderation capabilities and social network integration.
* Analytics for Sites stands side-by-side with your implementation and delivers solid metrics not just based on the usual categories but also meshes with your targeting campaigns and edge caching strategy as well.
* Targeting for Sites gives you the opportunity to segment visitors based on metrics appropriate to your business and to deliver relevant, informative content based on those segments.
* By deploying Satellite Server nodes on low-cost hardware, pages can be cached, assembled and even personalized for each visitor.

# **WebCenter Sites Asset Modelling Tips**

* There are two models for designing content types in WCS – the Basic Asset Model and the Flex Asset Model.
* Flex Assets were introduced later as part of the Catalog Centre module. They were initially designed to implement product catalogs for e-commerce sites, but are now used to model just about everything.
* Flex assets and families give you a lot of cool functionality as well as scope to change things up when you’d like to.
* Basic assets are good for data whose structure is unlikely to change and needs to be accessed quickly.
* Both Asset Models come with their own benefits and drawbacks. While most implementations now use Flex Families for the majority of their Asset Modelling due to their ease of use and functionality, it’s always worth keeping Basic Assets in mind for specific tasks.

When designing your Flex Family, keep in mind the following:

* Determine when to make an attribute single- or multi-valued
* Children can overwrite the value inherited from their parent(s), however the overwritten value is not deleted from the database
* How many values are inherited from parent to child, and likelihood of parents to be modified on a frequent basis.
* You’ll generally need to use the assetset: JSP tags to access any Flex Assets

When designing your Basic Assets, keep in mind the following:

* No hierarchy
* No inheritance
* One primary storage table per asset type
* Attributes correspond to columns
* Each new instance is a new row in the table
* Schema is fixed
* You’ll need to use the asset: JSP tags to access Basic Asset

**Types of CS Administrators in Fatwire**

We use the term “Admin” / “Administrator” frequently. But, there are different types of Administrators defined by the Content Server in Fatwire. We will now see the different types of Administrators in Fatwire:

1. **General Administrator**: He is the normal Administrator. He is responsible for managing all systems in the Content Server environment and has full and unrestricted access to each system’s interfaces. He is also responsible for the creation of other users, creation of ACLs for them, and managing them.
2. **Site Administrator**: He is the admin for a particular Site / Sites. He has access only to the site he is granted, and this user is generally created and manged by the General Administrator.
3. **Workflow Administrator**: He is the one who creates workflow processes. He creates the workflow processes, and the rest of the users participate in the workflow.

This is an overview of Administrators in Fatwire. Basing on the requirements, the above-mentioned roles will be created.

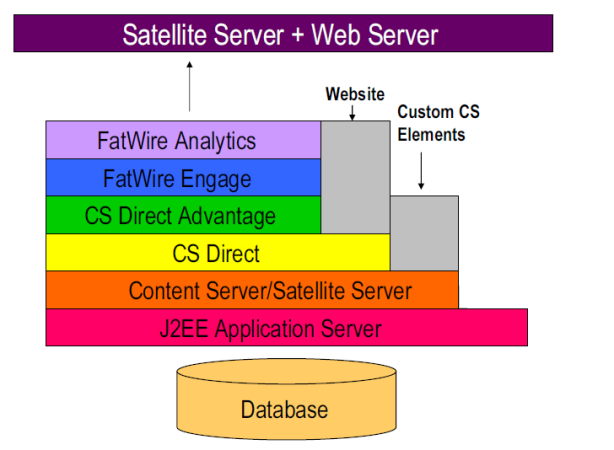
In Oracle WebCenter Sites (WCS), there are three types of administrators: general administrator, site administrator and workflow administrator.

A general administrator is responsible for managing the entire WCS system. This includes managing servers, services, web applications and site collections. They have full control over all aspects of the system.

A site administrator is responsible for managing a specific site collection. They can manage users and groups, configure site settings, create and manage lists and libraries, and customize pages. They have full control over their site collection but do not have access to other site collections.

A workflow administrator is responsible for managing workflows in WCS. They can create and manage workflows, monitor workflow status and troubleshoot issues. They have full control over workflows but do not have access to other aspects of WCS.

# Content Server Architecture



* Database: The database sits at the bottom of all the components. It can be of our own choice and can include Oracle Database, MS SQL, DB2, etc.
* Application Server: The application server sits above the database and acts as the main component for running the Content Server. The Content Server Context (CS) is deployed in this application. This application server could be of our own choice, like Apache Tomcat, JBoss. etc.
* Content Server: The Content Server context provides access to the Java servlets that compose Content Server. This Content Server will have cache storage, and this cache is used to store the pages, content, blobs, etc., which are frequently requested.
* CS-Direct (or) XCELERATE: This module of the Content Server sits on top of the Content Server. The concept of Basic Asset Model is introduced by the CS-Direct module of the content server.
* CS-Direct Advantage (or) GATOR: This module of the Content Server is an extension to the CS-Direct. It adds additional functionalities to the existing data model.
* Fatwire Engage: This is an optional software that can be installed to keep end-users informed about recent attractions of their interest.
* Fatwire Analytics: This is an optional product that monitors and analyzes website traffic.
* Satellite Server: Satellite Server can be divided into two components - Co-Resident Satellite Server and Remote Satellite Server.

# Understanding Content Server

* The core of WCS is the Content Server layer. You actually call Content Server when you invoke the URL “/cs/ContentServer” (or “/servlet/ContentServer” or whatever your context path is). ContentServer is basically an interpreter of an XML-based programming language.
* The Content Server XML is not generic XML, but a sort of custom programming language, CSXML. Later WCS added to the CSXML with something more standard, and more general: JSP, allowing programmers to use the more familiar Java programming language instead of a proprietary and somewhat limited programming language.
* CSXML is extensible, in the sense that you can develop custom tag libraries and invoke a tag. It is widely used in WCS since large part of the Xcelerate and Gator layers are developed in CSXML.
* Actually not everything in Content Server is implemented in CSXML. CSXML is a sort of scripting language to implement the user interface but all the underlying logic (especially the database access logic) is actually implemented in Java, packaged in jars and called through tag libraries.
* There is also another, more standard, programming language available: JSP. So basically you have the choice of using CSXML to code your site and customizations, or use standard JSP (that of course allows you also to code the logic in pure and simple Java).