**PLATFORM**

* Hybris Platform consists of a standard set of extensions providing the main functionality of a Hybris installation.
* An extension is a group of features comprising a subset of SAP Hybris Commerce functionality.
* An extension can contain business logic, type definitions, a web application, and many other things.

**Core Concepts:**

1. [Architecture of SAP Hybris Commerce](https://help.hybris.com/6.0.0/hcd/8b555d0486691014bb59c1e835c53f8b.html)
2. [Extension Concept](https://help.hybris.com/6.0.0/hcd/8bbf0b9d866910149688b8d696c8d47e.html)
3. [Cockpit Framework](https://help.hybris.com/6.0.0/hcd/8b7eb27986691014a3568a4e74be4fc8.html)
4. [Product and Data Modeling](https://help.hybris.com/6.0.0/hcd/8c317a28866910149816b4a652470b16.html)
5. [Hybris Platform Search Mechanisms](https://help.hybris.com/6.0.0/hcd/8be9f4e586691014b318d62074b74ab5.html)
6. [Internationalization and Localization](https://help.hybris.com/6.0.0/hcd/8bfc4c3c8669101493d9a2d97afcc5df.html)
7. [Media](https://help.hybris.com/6.0.0/hcd/8c0fbaf186691014b1389bc475fb7b02.html)
8. [Security and Users Management](https://help.hybris.com/6.0.0/hcd/8c5896be86691014afb19ff3320d73be.html)

## Essential Features

1. [Administration in SAP Hybris Commerce](https://help.hybris.com/6.0.0/hcd/8b4e446786691014b4539d60054d4898.html)
2. [Hybris Platform Web Services](https://help.hybris.com/6.0.0/hcd/8bea21ae86691014ba63caf92420bcf6.html)
3. [Ordering, Payment and Pricing Standards in the Hybris Platform](https://help.hybris.com/6.0.0/hcd/8b3b38dd86691014ac9b85cc0c8637bb.html)
4. [Processing in SAP Hybris Commerce](https://help.hybris.com/6.0.0/hcd/8c314fc1866910148c3ce11aef933f54.html)
5. [Reporting](https://help.hybris.com/6.0.0/hcd/8c3d5ef386691014b1d4e1d9f283cfe1.html)
6. [Data Validation](https://help.hybris.com/6.0.0/hcd/8ba81fc286691014a111ee7a527abf72.html)

## Technical Topics

1. [Building SAP Hybris Commerce](https://help.hybris.com/6.0.0/hcd/8b6e6adb86691014abf3fa1d3e37c3ef.html)
2. [SAP Hybris Commerce in a Cluster](https://help.hybris.com/6.0.0/hcd/8be81dcb86691014af64fa24e2b390b1.html)
3. [Application Performance and Monitoring](https://help.hybris.com/6.0.0/hcd/8b54a65386691014baa8fea5134a62ba.html)
4. [Testing](https://help.hybris.com/6.0.0/hcd/8c6f059d866910149b6aede7ab7a0f36.html)
5. [Logging](https://help.hybris.com/6.0.0/hcd/8c07853c866910148a00baf81ea1669e.html)

# ARCHITECTURE OF SAP HYBRIS COMMERCE

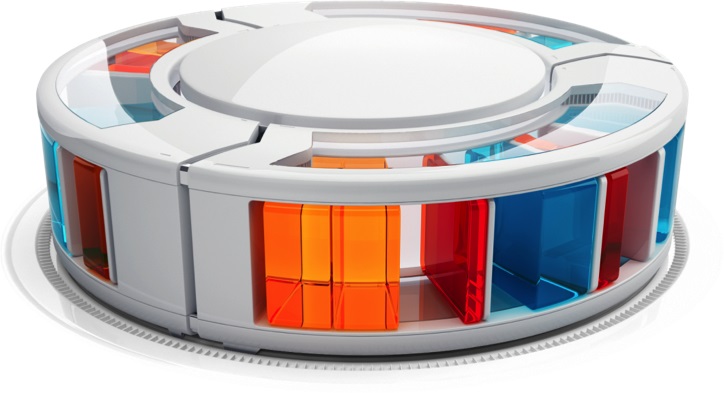
* The architecture of SAP Hybris Commerce is flexible and modular.
* At its foundation is Hybris Platform consisting of core functionality upon which all other functionality is built.
* An understanding of the various layers is essential to customizing the solution.
* To familiarize yourself with the whole architecture, first of all you need to understand its basic concept and layer approach to architecture.
* Then you can dig deeper into the topic of Service Layer and how it works together with the Spring framework.
* Then you should learn about more technical topics like caching, filtering solution, multitenancy, and possibly how to keep track of attributes changes.

# Architecture Overview

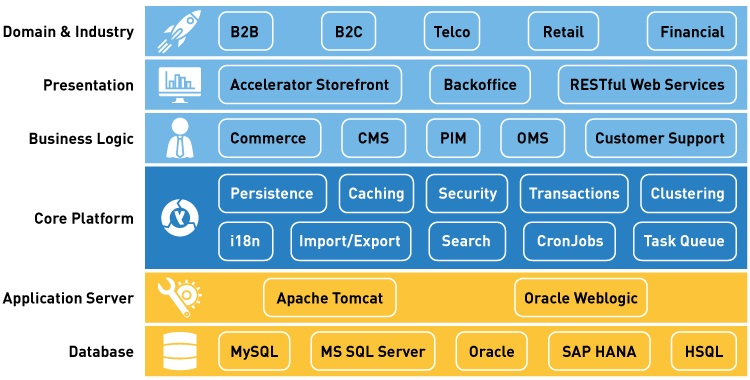
* SAP Hybris Commerce is highly flexible and modular software.
* This flexibility comes from several layers of abstraction and modularized functionality.

## Basic Architecture

1. From a business point of view, SAP Hybris Commerce is divided into individual packages, such as Commerce, Content, Channel, and Orders.
2. These packages are bundles of features assembled for a certain range of business functionality.
3. All of these packages rely on more basic functionality provided by the Hybris Platform.
4. While the Hybris Platform can run without any package, no package can run without the Hybris Platform.



1. From a more technical point of view, packages consist of individual modules (also referred to as[extensions](https://help.hybris.com/6.0.0/hcd/8b49cab88669101489be9ac91a5f1ebb.html) ).
2. For example, the Hybris Print technically consists of two extensions: **Print** (the technical foundation) and **Print Cockpit** (the graphical user interface).
3. Extensions are written by Hybris or the implementation partner of your project.
4. Extensions written by Hybris provide standardized functionality and are supported and maintained by Hybris.
5. If you write an extension, you need to maintain them by yourself, but you are free to implement any business functionality you need.
6. A full SAP Hybris Commerce installation therefore consists of the Hybris Platform plus any Hybris packages plus any additional extensions that you have implemented.
7. Extensions that are part of Hybris Platform proper are also referred to as the core extensions.
8. On top of these core extensions, Platform contains several pieces of Hybris software, such as the [Build Framework](https://help.hybris.com/6.0.0/hcd/8b6ded0d86691014a6fab18e171c1f91.html), and third-party software, such as the pre-bundled Apache Tomcat.
9. SAP Hybris Commerce is run in a Java Virtual Machine on a Servlet Container or a J2EE-compliant application server (such as Oracle WebLogic) and connects to an external database (MySQL, Oracle DB, Microsoft SQL Server).
10. Internal caching and persistence mechanisms allow SAP Hybris Commerce to run on a Servlet Container.
11. A full-fledged J2EE-compliant application server can be used but is not necessary.

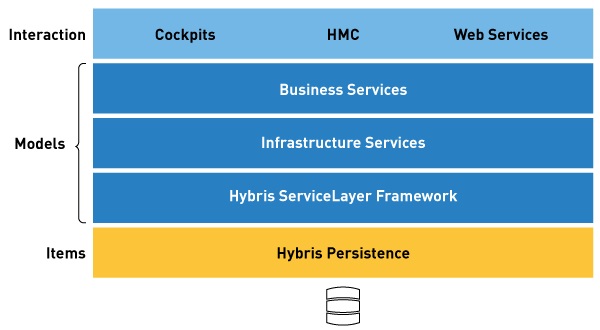


***Figure 1: A view of the various features of an SAP Hybris Commerce installation.***

1. The Platform layer abstracts data from the storage structure on the database using the persistence framework and provides functionality such as [Clustering](https://help.hybris.com/6.0.0/hcd/8b7c5c75866910148c329ee71f3a681e.html) and the [Hybris Platform Cache](https://help.hybris.com/6.0.0/hcd/8be98ee4866910149df8be0aab4d0b62.html" \o "The Hybris Cache is a part of the Hybris persistence layer. It improves the performance of a single server node by reducing the amount of database queries. It transparently stores search results, item attributes, and item instances in memory.).
2. Relying on the persistence framework, the other functional components of the Platform Layer provide basic business functionality: [Transactions](https://help.hybris.com/6.0.0/hcd/8c7387f186691014922080f2e053216a.html), [CronJobs](https://help.hybris.com/6.0.0/hcd/8b9ce4868669101499b2f0f25ef9395f.html), [Personalization](https://help.hybris.com/6.0.0/hcd/8c428f8286691014970ceee87aa01605.html), [Internationalization](https://help.hybris.com/6.0.0/hcd/8bfc204086691014a345f64b08505839.html), and more.
3. The packages on the Functional Layer (SAP Hybris Commerce, Hybris PIM, Hybris Print) use Platformto implement the functions they deliver. Actually, Hybris Platform is part of any Hybris Package.

## Layer Architecture

SAP Hybris Commerce contains several layers, each of which has a different function and data abstraction level.

****

***Figure 2: Overview of the abstract layers of SAP Hybris Commerce.***

| **Layer Name** | **Description** | **What would a Product look like on this Layer?** |
| --- | --- | --- |
| Cockpits, HMC, WebServices | 1. This is where objects are represented in a way that an end-user can interact with them: add products to a cart, edit a product description, or set a password for a user account, for example. 2. On this layer it is possible to let a user do something with an object in theHybris Platform via a graphical user interface. 3. Functionality on this level (such as the JSF-based HybrisStoreFoundation or the ZK-basedHybris Print Cockpit) uses the ServiceLayer for functionality and the Type Layer for storage of objects. 4. Depending on the complexity of your implementation, this layer can itself consist of several individual layers and even use an individual data model |  |
| ServiceLayer Framework  (including the actual ServiceLayer, the Infrastructure Services, and the Business Services) | 1. Provides the Java Application Programmer's Interface (API) for objects in SAP Hybris Commerce, the Hybris API. 2. The HybrisServiceLayer relies on so-called models, which are POJO objects. 3. Attributes on models have automatically generated getter and setter methods. 4. Models are generated based on types. | **ProductModel.java**  **public** **class** ProductModel **extends** ItemModel  {  /\*\* <i>Generated constant</i> - Attribute key of <code>Product.catalogVersion</code>  \* attribute defined at extension <code>catalog</code>. \*/  **private** **static** **final** String CATALOGVERSION = "catalogVersion";    /\*\* <i>Generated constant</i> - Attribute key of <code>Product.code</code>  \* attribute defined at extension <code>core</code>. \*/  **private** **static** **final** String CODE = "code";    \* <i>Generated method</i> - Getter of the <code>Product.catalogVersion</code>  \* attribute **defined** at extension <code>catalog</core>.  \* @return the catalogVersion  \*/  **public** CatalogVersionModel getCatalogVersion()  {  **if**( !isAttributeLoaded(CATALOGVERSION))  {  **this**.\_catalogVersion = (CatalogVersionModel)  loadAttribute(CATALOGVERSION);  }  **return** **this**.\_catalogVersion;  }    /\*\*  \* <i>Generated method</i> - Getter of the <code>Product.code</code>  \* attribute defined at extension <code>core</core>.  \* @return the code  \*/  **public** String getCode()  {  **return** \_code;  } |
| [Type Layer](https://help.hybris.com/6.0.0/hcd/8c755da8866910149c27ec908fc577ef.html) | 1. Describes business object models. 2. It is on this layer that definitions of business objects (types) and their fields (attributes) are made via theitems.xml (see [items.xml](https://help.hybris.com/6.0.0/hcd/8bffa9cc86691014bb70ac2d012708bc.html)) file. 3. Models are generated based on types. | **items.xml**  <itemtype code="Product" extends="LocalizableItem">  <attributes>  <attribute qualifier="code" type="java.lang.String"/>  </attributes>  </itemtype> |
| Persistence Layer | 1. Deals with abstraction from the database, caching, and clustering. 2. You are not likely to get into contact with the Persistence Layer at all as it is completely transparent and does not require any explicit interaction from your side. | SQL-compliant representation of numbers and strings in a database table, such as [VARCHAR](http://help.sap.com/disclaimer?site=http://en.wikipedia.org/wiki/Varchar), [CLOB](http://help.sap.com/disclaimer?site=http://en.wikipedia.org/wiki/Clob) |
| Database | 1. Although not a layer of SAP Hybris Commerce, the database is also an important component in this overview: the database makes the data held in SAP Hybris Commercepersistent. | Database-specific representation of numbers and strings. |

## Modes of Operation

You can run SAP Hybris Commerce in three different modes of operation:

