# Product Information Management

This article has been generated from the online version of the documentation and might be out of date. Please, make sure to always refer to the online version of the documentation for the up-to-date information.

Auto-generated at April 18, 2024

Overview

The Product Information Management (PIM) module is used to create a set of interconnected templates that allows collecting, storing, and managing product data on the CloudBlue Connect platform. Namely, Distributors use this module to define templates that enable Vendors to provide product characteristics and specifications required for certain distribution channels.

Such templates represent a *Product Information Model* and this model includes *product attributes*, *groups*, *classes, categories*, and *items variation*. Note that one Vendor can provide structured product information to one or multiple Distributors as schematically illustrated in the diagrams below:
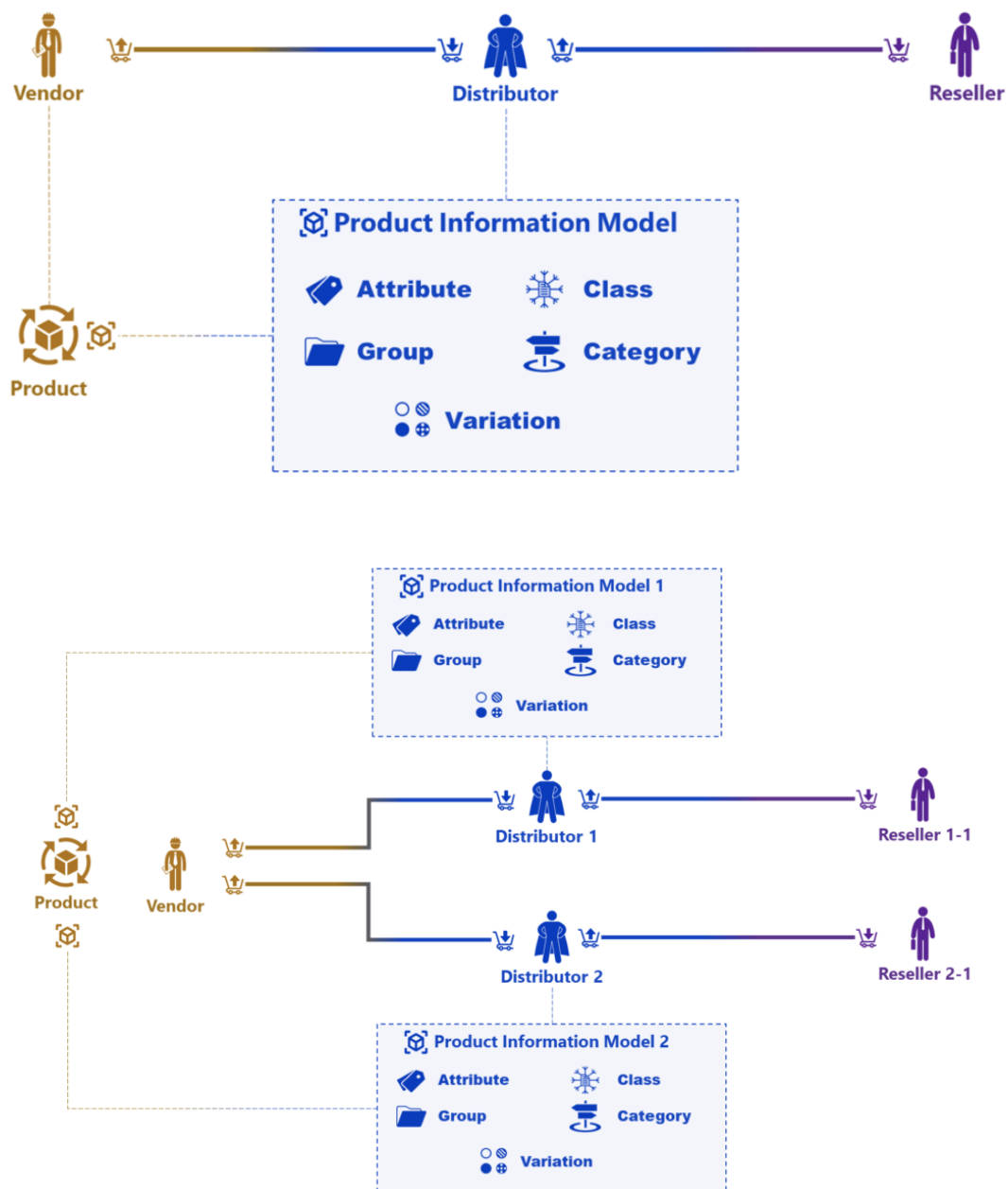
Therefore, the PIM module provides a unified way to more accurately describe available products, which in turn is increasingly

beneficial for all organizations and companies that collaborate via the CloudBlue Connect platform. Well-structured and systematized information enables your customers to get the clear understanding of your presented product and its features. Consequently, your customers can quickly find a product with required specifications. This drastically improves the customer experience on your marketplace and can potentially lead to larger sales and profits.

The following outlines the structure of the Product Information Model and also describes the content of the PIM module on the Connect platform.

⬢

Product vs. Items

Before familiarizing yourself with the following content, it is vastly important to understand the difference between *products* and *items* as described in the Items sub-article of the Product module documentation.

Data Model

The Product Information Module consists of six interconnected blocks that are schematically illustrated in the diagram below:
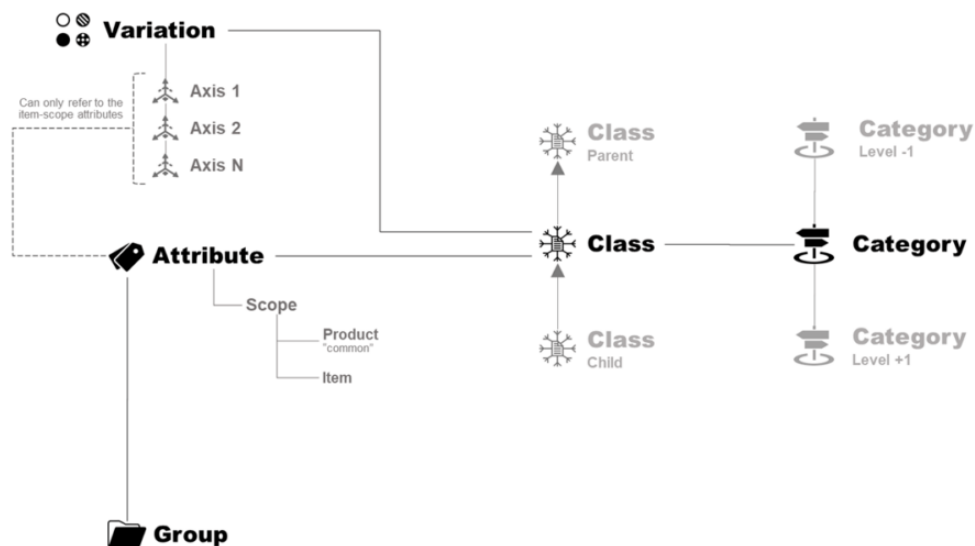
The following describes each block and consequently introduces the content of the PIM module:

Attributes

Attributes are the center and the foundation of the PIM module. Defined attributes generally include all provided technical specifications and any other aspect that is required by Distributors. It is possible to specify as many attributes as necessary for

your product. Note that the Connect platform allows specifying attributes in one of the following scopes:

- **Product level**: This includes attributes that are shared across all items of the product (for example, product name, specified manufacturer, weight, etc.)
- **Item level**: Attributes that are differentiate one item from the other items of the same product (e.g., color, memory capacity, and so on.)

Attribute scopes differentiation is illustrated in the example below:

This example showcases Apple MacBook Air and its listing within a marketplace, This listing includes both product-level and

item-level attributes.

Product-level attributes represent the product name (*Apple MacBook Air 13-inch with Retina display*), weight (*2.8 pounds*) and the brand name (*Apple Computer*). These attributes are common for all available items.

Item-level attributes are differentiated by *memory capacity*, *color, price* and other specifications. This attributes are different for each provided item.

It is crucial to understand that some attributes may also require different **localizations**. Providing localization for attributes is important in case the attribute data should be collected as the free-form (unstructured) text.

Groups

In general, specified attributes can be grouped by certain criteria. Therefore, the PIM module also allows defining groups and consequently populate this groups with your added attributes. Defined groups may represent the physical characteristics of your product (such as weight, color, and size), technical information on your product (CPU type, available RAM, GPU, etc.), general product information (e.g., product name, type, brand), and so on. Attribute grouping makes your products more understandable for customers and consequently can greatly improve the user experience on your marketplace.

The following example demonstrates and highlights various attribute groups within the technical specifications of V7 HB600S Headset:

The provided technical specifications showcase various groups of attributes, such as *Interface/Ports*, *Warranty*, *Microphone*, *Earpiece* and so on. Each group helps customers understand the context of every presented attribute. Furthermore, provided
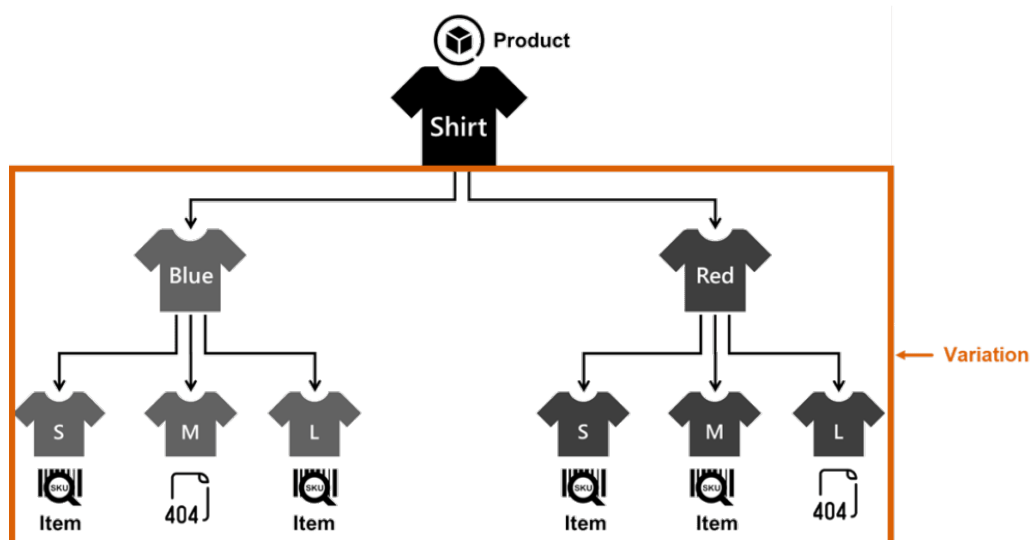
groups also allow locating required attributes faster.

Note that your attributes can be associated only with a single defined group. Namely, the system doesn't allow multiple groups to share the same attribute since there is no practical use to associate an attribute with more than one group. A single group, however, can feature one or more specified product attributes.

Variation

Attributes that are defined at the *item level* can specify variants of your product. A product variant (or variation) can be seen as a multidimensional matrix, where each *axis* represent a certain item-level attribute. Namely, Distributors create a template for Vendors to provide a product variant structure for the graphical user interface on the marketplace using the defined axes. Therefore, customers can filter presented attributes of the product and purchase a required SKU (item) by selecting the combination of its color, size, and so on.

The following showcases the variation concept and demonstrates different variation examples:

The first example introduces a product (shirt) and 2 particular dimensions: shirt color and size. In this case, Distributors can create a variation template that includes all shirt sizes and shirt colors. A distribution channel that features this product allows customers to select the size and color of their purchased shirt. Note, however, that the Vendor of this product cannot provide certain items (i.e., blue M shirts and red L shirts). Therefore, some fields in the corresponding product variation template can be intentionally left empty.

The second example presents a product (Iphone) with many attributes that help customers choose the most suitable item. In this scenario, the Distributor created a variation template include *color*, *capacity*, *weight*, and other item-level attributes that act as axes for this particular template. However, the Vendor did not specify the *weight* attribute for each item, leaving the entire axis and the row of the product description blank. This can potentially cause errors in some distribution channels. In such cases, the PIM module allows Distributors to create several variation templates for this product type. One variation template can feature the same set of attributes and exclude the *weight* attribute, while another template can include all attributes for such products. Thus, associated Vendors can select a required variation template if they can provide the weight for their items and in case they cannot provide such attribute values. This eliminates any possible errors while the distribution channel processes provided item information.

Distributors can use this logic to define as many variant templates with different combination of attributes as it is required for the specific product type.

Classes

A class is a set of all possible *attributes* and *variations* that are shared by products of the same type. Namely, a class represents

CloudBlue

a template that is used to systematically organize provided product attributes and consequently to determine where each attribute should be placed within the product and item records.

Classes can also inherit certain attributes and variations from other created classes. Thus, the PIM module can be used to implement various product hierarchies. The following diagrams illustrate an example of the class hierarchy:
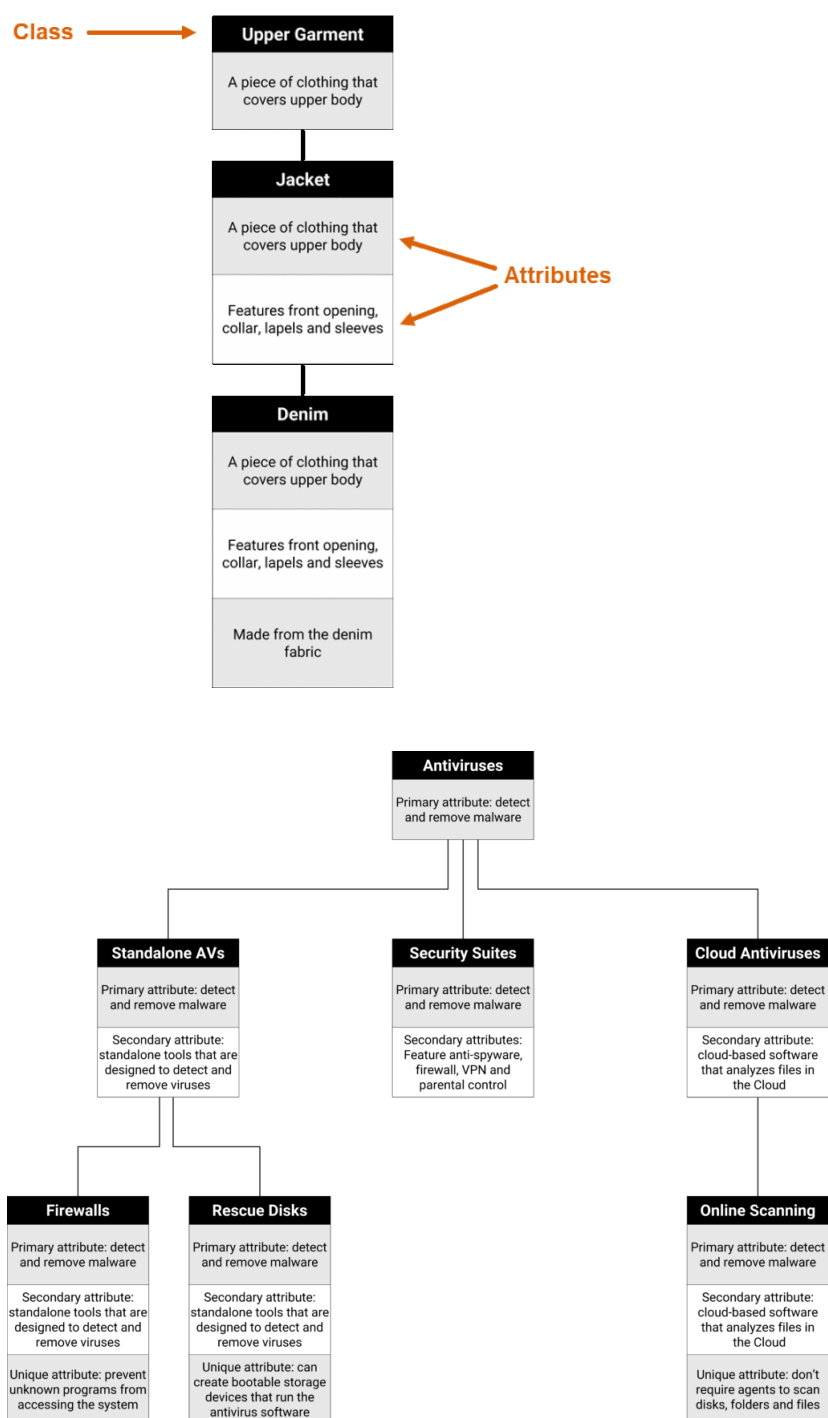
The first diagram introduces three product classes with the growing number of provided attributes with each level. Specifically,

the *jacket* class inherits an attribute from its parent class *upper garment*. This diagram also includes a *denim jacket* product class that inherits both attributes from the *jacket* class.

The second diagram provides an *antivirus software* classification and also showcases hierarchy of the provided classes. Each presented class specifies unique features and also inherits attributes from its parent class.

Therefore, this class hierarchy allows defining default attributes for the child classes and allows providing more structured technical information of a specific product or item.
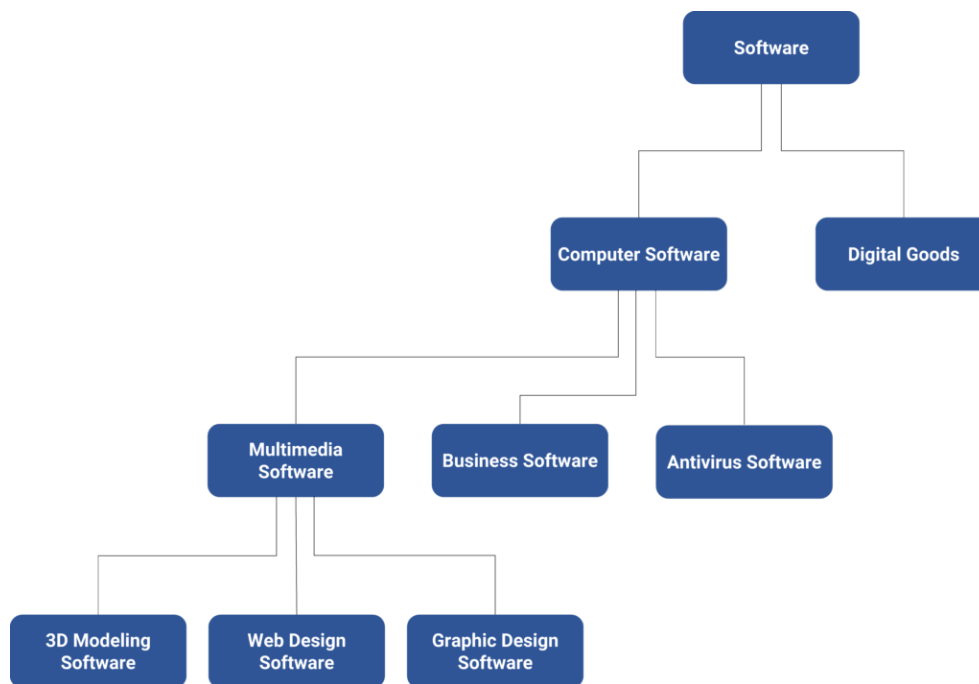
Categories

Categories represent marketing entities that allow determining and organizing products that share same features and characteristics. By categorizing and grouping products, buyers can easily find required products with specific attributes. Therefore, categories drastically increase the product visibility and improve the customer experience within your commerce system.

Product categories are closely interconnected with *classes* despite having different functions. Namely, a category should include only products of a particular class. Furthermore, note that one class can define multiple categories. Classes may provide certain attributes that may not be applicable for any category. Thus, organizations and companies can create thousands of categories for their distribution channel while having a handful of classes.

The PIM module also enables to define a hierarchy of categories that resemble a tree structure as schematically illustrated below:
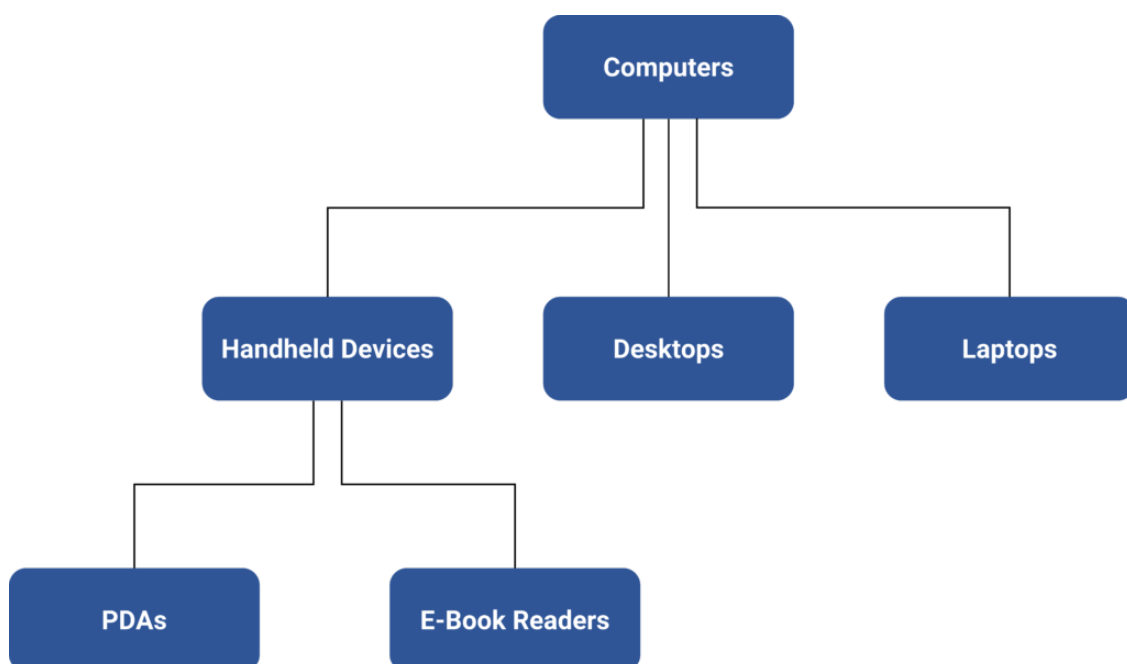
The first diagram showcases *computers* that is a parent category for the following hardware categories: *laptops* and *handheld devices*. Note that the *handheld devices* category also includes *PDAs* and *e-book readers* categories, while the *desktops laptops* categories don't include any child categories.

The second diagram demonstrates a more complex hierarchy of software products. This includes various interconnected objects that represent broad categories (such as *computer software* and *digital goods*) as well as more specific categories (e.g., *web design* or *graphic design software*).

Therefore, by using the same logic, Distributors can create hierarchies for their product categories and generate as many categories and sub-categories as required for their marketplaces.

Taxonomies

Taxonomies represent objects on the Connect platform that contain specific product-level and item-level data submitted by Vendors in accordance with defined PIM templates. Namely, in case Distributors successfully create required PIM templates on CloudBlue Connect, the system allows Vendors to provide product-level and item-level taxonomy specifications via the Products module.

Once Vendors select one of the provided categories and specify attribute data for the product, the system automatically generates a product taxonomy object that is associated with a particular Distributors. Thus, Distributors can also view the taxonomy data via the *Product* module in case Vendors submitted a listing request with this product.
It is important to note that a taxonomy object is bound to a particular Distributor. Taxonomy objects also provide additional data

via the API calls (such as their identifiers, update and creation date, owner, partner, and so on).

More Information

Refer to the following sub-article to learn more about the provided API, graphical user interface and available operations of the PIM module:

1. PIM API
2. PIM Interface