commercetools
Next generation commerce

Migration from SAP Commerce Cloud to commercetools

www.commercetools.com

Table of Contents

Executive Summary

It’s prime time to migrate over from SAP Commerce (formerly known as SAP Hybris) to the leading modern commerce platform: commercetools. For many reasons that we will go over shortly, SAP Commerce is a poor fit for today’s commerce needs. This white paper leads architects and their teams through the platform migration from SAP Commerce to commercetools, going through the necessary steps in this process, as well as how to build a migration roadmap, migrate data and deal with custom extensions and user interfaces.

Introduction

In today’s fast-moving commerce business, brands and merchants need to enhance their agility and flexibility. Customer demand is continually changing, new touchpoints are emerging, and innovative ways of communication between consumers and suppliers have appeared. Take voice commerce that’s enabled by voice assistants built into devices: In a few short years, over 200 million smart speakers have been sold. Similar to how the App Store helped Apple shape a new age of app-driven mobile devices, Amazon lets third-party developers build new capabilities on top of their voice platform. Brands and retailers need to be able to build new prototypes quickly, experiment with the user experience, create great services for their customers and drive loyalty.

In reality, however, many organizations are kept from building new business models by their slow, hard-to-adapt software platforms which effectively stifle innovation. The only way out: move to another, more flexible architecture.

In this white paper, we look at the long-standing SAP Commerce platform that’s showing its age, and on the other end of the scale, the innovative, cloud-native commercetools platform. SAP Commerce has seen decreasing relevance among its customers, leading to many of them becoming dissatisfied and reevaluating their choice of commerce solution. We provide suggestions on migrating from SAP Commerce to commercetools without interrupting daily business operations.

Of course, there is no standard setup which could be migrated in a standard way. Every project is different and has its own caveats, and we’re by no means suggesting a cookie-cutter approach. Rather, this suggested strategy serves as guidance or a framework for planning a move over with increased chance of success.

Commerce Platform Migration

SAP Commerce in a Nutshell

SAP Commerce is based on its acquisition of Hybris, which was founded in 1997 as a provider of an independent, enterprise-class commerce platform. As a very long-standing platform, it has driven some of the world's largest and most successful retail websites, such as Samsung (Electronics), Douglas (Beauty), Oakley (Accessories) and Philips (Consumer Electronics). Hybris was acquired by SAP in 2013 and subsequently rebranded to SAP Commerce, the name we know today. Technologically, it's a Java Spring-based software monolith, originally built as an on-premise solution.

It comes with so-called accelerators that provide data models and frontend templates to kickstart development in a number of specific verticals. There is also the Omni Commerce Connect (OCC) layer, which is essentially an API exposing some of the core's functionality. Recently, in the wake of the success of cloud computing, SAP has extended its portfolio with a cloud solution called SAP Commerce Cloud. This is a single-tenant, private-cloud solution that is deployed into a Microsoft Azure-based managed hosting environment.

Some customers of SAP Commerce Cloud have complained that, due to the combination of a relatively immature cloud environment, lack of know-how in operating in the cloud and a platform that is not cloud-native, it is fraught with issues and has poor support for various customer deployments.

commercetools: Innovative Cloud-Native Platform

commercetools is a commerce platform built from the ground up for modern business needs with its cloud-native, multi-tenant, headless commerce solution that comes with auto-scaling and continuous updates. With comprehensive APIs, support for microservices and the highest GraphQL coverage among any commerce platform, commercetools is a highly adaptable, easily workable platform which brands, merchants and system integrators can use to quickly and effectively to create great customer experiences.

Despite being relatively new compared to the decades-old monolith suites, commercetools has already been named a leader by three of the largest industry analysts (Gartner, Forrester and IDC MarketScape), while its platform powers major brands around the world, including companies like Audi, Carhartt, Express and Volkswagen.

Reasons for Wanting to Migrate

Total Cost of Ownership (TCO)

The on-premise version of SAP Commerce was typically sold in the form of core-based licenses. The platform needed to be run on multiple application nodes and database servers, generating operating costs for hosting applications. Even for mid-sized companies, six-figure licensing fees were not unheard of. Brands and retailers faced challenges when it came to scaling, as any business growth and peaks in traffic would correspond to higher licensing costs.

In recent years, SAP has dropped support for its on-premise solution and is instead pushing their newer Cloud Commerce product. While SAP Cloud Commerce has lowered costs with bundled pricing that covers infrastructure, software licensing and more, it introduces new downsides where you are now limited to using Microsoft Azure as the cloud provider and it is difficult to deploy many services into the infrastructure. In addition, bundled pricing consists of three "instant sizes" that, depending on use case and deployment, some businesses can easily exceed and incur additional costs.

Businesses also need to consider that SAP releases new versions several times per year, with often fundamental changes to the core, such as changing internal APIs and Java versions. With the original SAP Commerce, this led to businesses needing to constantly review and rewrite parts of their custom code to ensure continued functionality with the new version. This is a major pain and often requires six-figure investments. Nevertheless, these upgrades are necessary to be eligible for support – which, in the form of a support contract, also generates recurring costs.

With SAP Cloud Commerce, some of that complexity is reduced, but the associated downtime for maintenance required for those quarterly updates remain. This leads to unseen costs that arise in the form of additional planning for every update, as well as direct costs in lost sales for every second the platform is down for maintenance.

	SAP Cloud Commerce	commercetools
Licensing	Core-based	Usage-based
Operations	Multiple application and database nodes necessary to run at scale	Platform run a scaled by commercetools; only front-end hosting as external cost
Updates & Upgrades	Compulsory updates a few times a year	Continuous integration, features pushed daily, non-breaking API changes

Lack of Agility and Speed

Because of the complex nature and the strict architecture of the SAP Commerce platform, developers need to work with many different layers. For example, a rather trivial task, such as building a custom promotion and displaying it in the frontend, often takes backend developers a few days to implement. This is mainly due to a lot of boilerplate code and the fact that the ORM data modeling is done via XML configuration.

	SAP Cloud Commerce	commercetools
Customization	Experienced SAP Commerce Cloud developers customized complex monolith	Developers work with well-documented commerce REST API
Technology	Java, JSP, XML configuration	Language-agnostic, SDKs available for Java, PHP + others

A New Approach to Commerce

In order to understand the differences between the commercetools and SAP Commerce platforms, there are two concepts you should be familiar with. These are what we believe to be key differences between the two platforms and come into play as we discuss moving your business to commercetools.

The MACH Standard

Modern commerce platforms are built differently than their counterparts that were built over 20 years ago. commercetools is a founding member of the MACH Alliance, which means it adheres to 4 guiding principles as a platform. These principles are core to enabling the platform to be performant, flexible and easy to work with.

The MACH Four

Technologies that are disrupting the e-commerce software market.

M

Microservices
A modern architecture that makes your IT team agile, possibly even SUPERSONIC.

A

API-First
100% API-centric means you can incorporate any functionality.

C

Cloud-Native
Huge promotions, giant traffic spikes - no worries. Our multi-tenant, cloud-native platform is always there.

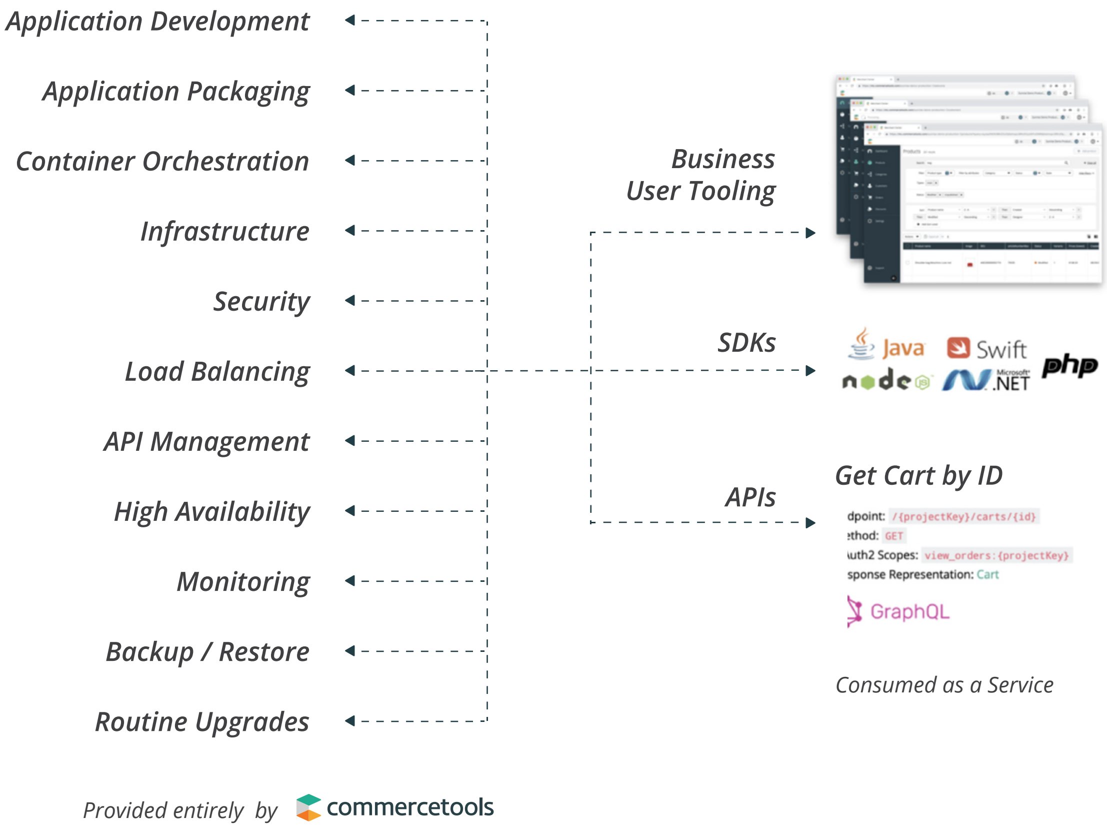
H

Headless
Choose your own frontend: build from scratch or buy a packaged DXP. Put your customer experience front and center - we've got both ends covered!.

Enterprise SaaS

Along with platforms following the MACH standard, a new approach to how your platform is implemented and managed is taking hold in the market. In the previous generation, platforms were traditionally installed "on-premise," or in other words, hosted and managed by you. This required large IT teams to take care of things like hardware provisioning, databases, network security, monitoring and patching/upgrading, and so forth. The checklist you need to manage to host a commerce platform is extensive and comes with the constant expense of having an extensive team with the right people just to make sure it's a secure and performant environment.

Many software vendors of previous generation commerce platforms have attempted to jump on the bandwagon with “cloud” branded versions of their product. However, many of those solutions were not built from the ground up to support modern business use cases; rather, they are spin-offs of the original legacy product, still rooted in an archaic foundation and inherit many of the shortcomings that would not exist with a true modern commerce platform. Today, enterprise SaaS (Software-as-a-Service) solutions are mainstream. You no longer need to have an operations team to handle “keeping the lights on.” At commercetools, we have a world-class team managing the multi-tenant platform for your specific implementation and the team has a proven track record of success; so much so, that we contractually guarantee our platform will be up and performant for every customer.



The Right Time for a Migration Project

Businesses should not take the decision of moving away from their current platform too lightly. After all, introducing a new technology always has the potential to disrupt operations. On the other hand, such a move always enables brands and merchants to re-evaluate whether their current IT ecosystem is able to support current and future business objectives.

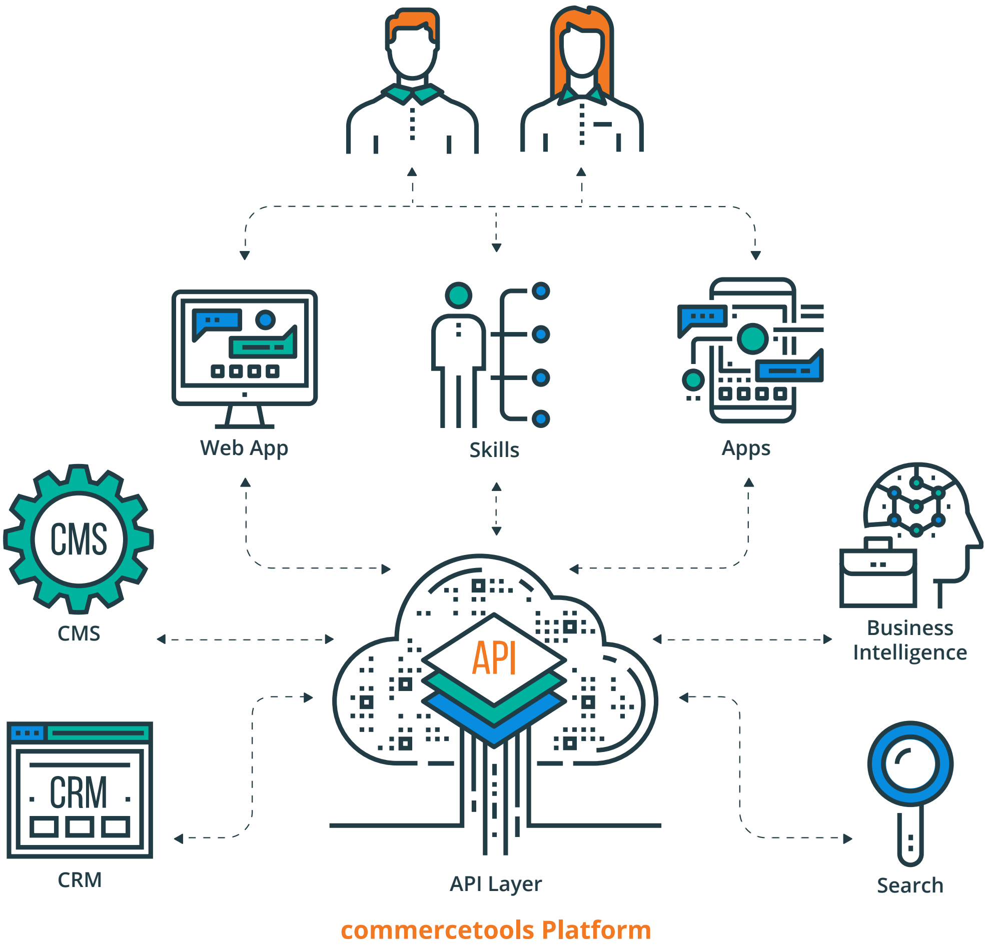
In the case of running SAP Commerce, a good time for evaluating other software solutions is when businesses run into the need to perform a compulsory upgrade, which is something required to shift to SAP's Commerce Cloud as SAP Commerce support reaches end-of-life. If they know that this process involves investing in new developments anyway (see the TCO section), they might as well opt for a complete change and consider moving to a technology such as commercetools. Let's now dive deeper on how, in practice, such a migration is planned and carried out.

Migration Planning and Execution

There is no set-in-stone definition of what a platform migration means in the commerce space. For most people, the ideal outcome would be a "like-for-like" solution; meaning everything previously possible in the old solution and every feature that has been developed, would be there in the new solution as well – plus the benefits of the new platform. In most circumstances, as we will see a little later on, this expectation is not practical: when planning the migration, this is the perfect chance to eliminate overhead and generally aim for a leaner solution.

What we're suggesting here is to divide an existing project into business domains and to transfer the respective functionality and data out SAP Commerce Cloud to a best-of-breed infrastructure with commercetools at its core.

In most cases, this means: Going from on-premise monolith to a service-oriented, headless cloud solution, with everything which is connected to this project. We are also aiming at a phased migration (instead of a big bang approach) to minimize disruptions to operations and mitigate risks to the best extent possible.



Step 1: Discovery and Gap Analysis

In this first step, you could structure your tasks like this:

1. Taking stock: What does the current platform offer, which kinds of functionalities does it enable, which user stories does it support, which processes are running the background? These questions might sound obvious or trivial, but with a code-base which is a few years old, there are always individual edge cases which might not have been documented properly or which people have just forgotten about. Also, it is highly recommended not to shift your entire digital business all at once: focus on less important assets first, like a less relevant locale, before tackling the core business.

2. Setting priorities: Create a list of all of those processes and edge cases and decide which of those have to be migrated immediately, and which of those can be tackled at a later stage – or even dropped altogether. Make sure to view these items from a business perspective and determine which value they're adding to your business. (And even if you are tempted to stick with this feature that took ages to implement and that everybody on your team is proud of: if it does not add value, it probably shouldn't be on the list.

3. Gap analysis: Use this prioritized list of functions and processes and hold it against the features which are present out-of-the-box in commercetools. You will find detailed information in the API documentation. There are mainly three options for each item:

Out-of-the-box: The desired feature is a standard commercetools feature that can be configured.
3rd party: There are features and domains which are out of the commercetools scope, such as CMS/DXP. In these cases, 3rd party services need to be integrated.
Customized: These are the parts that have to be custom-coded.

4. Build teams: Finally, make sure your staff and your organizational structure meet the new demands of a cloud-based solution. Instead of clear-cut, horizontally organized skill sets – such as frontend specialists, backend developers and data scientists – the "new cloud world" requires the work of cross-functional and vertical teams.

Step 2: Build a Migration Roadmap

Next, build a migration roadmap which lists important milestones, deliverables and a timeline. Roughly speaking, there are three areas which are the basis for the roadmap:

1. Data: One of the first things you should tackle is making sure to move data from the SAP Commerce database to commercetools, namely the following:

Product catalog - catalogs, categories, products, SKUs, etc
Customer profile - customer segments, customer profile, address, payment methods, etc
Orders -carts, orders, shipping methods, etc.

We will talk more about what to do in these cases in Step 3 of the migration plan.

2. Business logic: This area includes all the custom extensions which need to be built or integrated when 3rd party services are included.

3. UI/UX: commercetools is a headless commerce platform. This means that there is no fully fledged storefront application like in SAP Commerce, although we offer a quickstart Accelerator program that allows you to quickly and easily build and deploy an online commerce experience, which you can later, very flexibly, build upon and expand to support the rest of your business' commerce needs across different devices and touchpoints.

***Note:* that those three areas do not have to be tackled in this exact sequence. These suggestions only serve to add a bit of structure to your planning.**

Step 3: Data Modeling

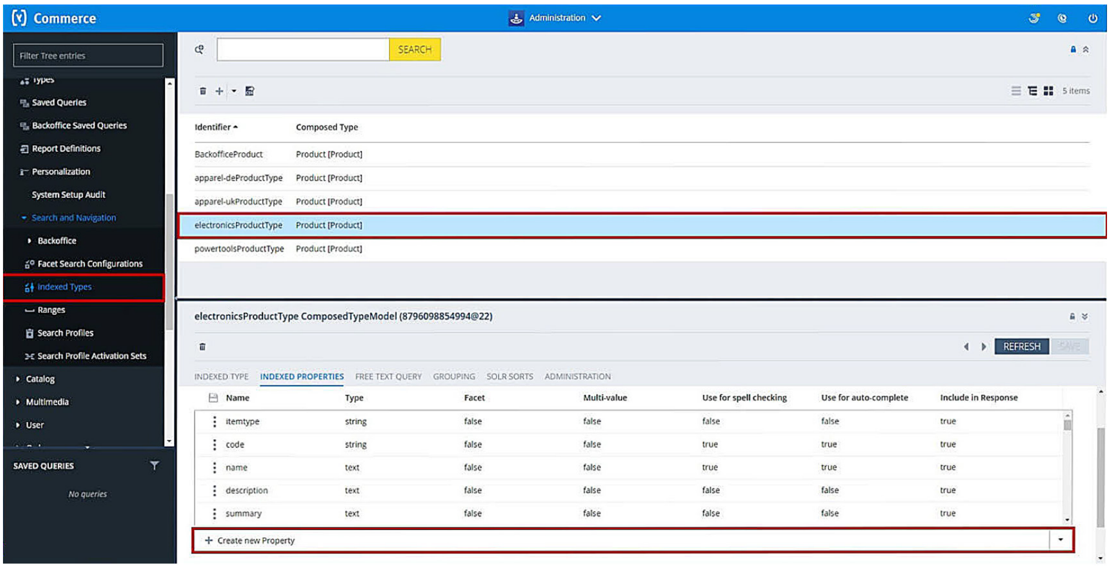
Begin by exporting your Jalo definition in SAP Commerce, which will include your customizations overlaid on top of the out-of-the-box data model. It’s essential that you can see the entire data model.

Export data using ScriptGenerator
Run exports for all types from script as csv
Modify files (map to new data model)
Create drafts on modified export csv
Load data

Next, diff that exported data model with [commercetools’ data model](). You will want to note any differences in:

Objects: a pet food company might have a “dog” object
Sub-types: a clothing retailer might have shirts and jeans product types, each with different attributes
Object attributes: a digital product might have a “download count” attribute

Unlike SAP Commerce, commercetools uses an extremely flexible data model that allows for real-time updates to its structure. Rather than changing a table in a database and then mapping it back to Jalo, you can change the data model in real-time using the Merchant Center or directly on API level.



Identifier	Composed Type
BackofficeProduct	Product [Product]
apparel-deProductType	Product [Product]
apparel-ukProductType	Product [Product]
electronicsProductType	Product [Product]
powertoolsProductType	Product [Product]

electronicsProductType ComposedTypeModel (8796098854994@22)

Name	Type	Facet	Multi-value	Use for spell checking	Use for auto-complete	Include in Response
itemtype	string	false	false	false	false	true
code	string	false	false	true	true	true
name	text	false	false	true	true	true
description	text	false	false	false	false	true
summary	text	false	false	false	false	true

Recommendation for First Mapping

If you implement your catalog as ProductTypes with custom attributes, you can use the following mapping as a guideline. Of course, those mappings are always specific to each project.

commercetools Types	SAP Commerce Type and Their Sub-Types
Address	Address
Asset	Media*
Category	Category
Channel	Warehouse for inventory; PointOfService for brick-and-mortar store
Customers	Customer
Customer Groups	UserGroup
Carts	Cart
Cart Discounts	Discount DiscountRow
Discount Codes	Voucher
Inventory Entry	StockLevel FutureStock
Orders	Order
LineItems Custom LineItems	CartEntry OrderEntry
ProductPrice	PriceRow
Payment	PaymentInfo PaymentMode
Payment InterfaceInteraction	PaymentTransaction
ShoppingList	Wishlist** SavedCart
Review	CustomerReview
Shipping Methods	DeliveryMode
Shipping Zone	Zone

*Different media types in SAP Commerce
**Wishlist = only SKU and priority, SavedCart = full cart for direct ordering

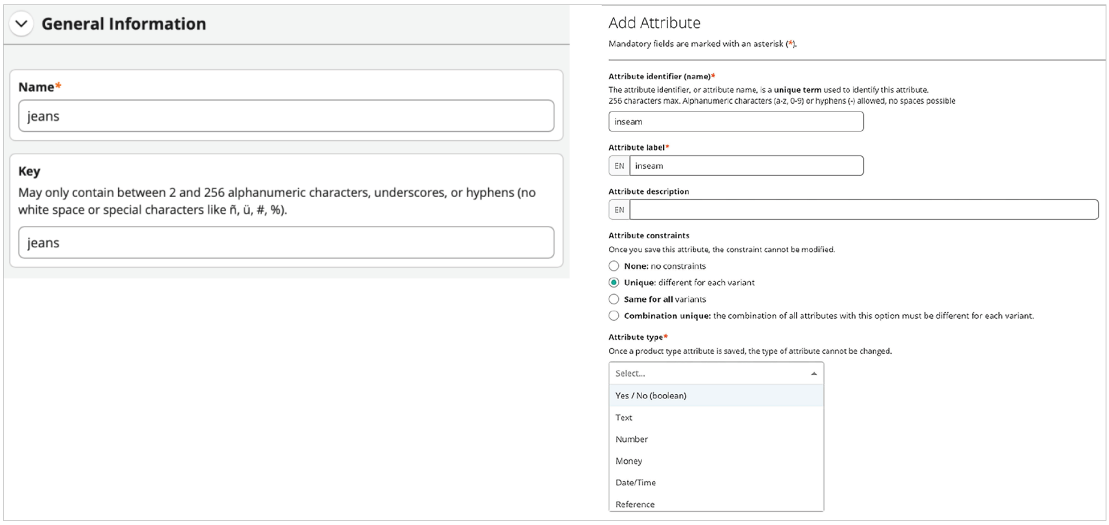
Custom Objects

Custom objects in commercetools are arbitrary JSON-formatted records that are persisted indefinitely. They can be identified by a key or an ID and can be nested using the container attribute. Attributes of a custom object can reference other objects in commercetools, like an order or customer profile. See Custom Object Types for more information.

Sub-Types Become Custom Types

Sub-types are represented as custom types in commercetools. You could define a T-shirt product sub-type that captures the size (S/M/L/XL) and a jeans product sub-type that captures inseam/waist size. When creating products, you could then create a generic product, a T-shirt product or a jeans product. You can apply this concept to other objects, including categories, customers, orders, etc.

- Tutorial on using Custom Types and Custom Fields
- Documentation on Types
- Documentation on Product Types



General Information

Name*

jeans

Key

May only contain between 2 and 256 alphanumeric characters, underscores, or hyphens (no white space or special characters like ñ, ü, #, %).

jeans

Add Attribute

Mandatory fields are marked with an asterisk (*).

Attribute identifier (name)*

The attribute identifier, or attribute name, is a **unique term** used to identify this attribute.
256 characters max. Alphanumeric characters (a-z, 0-9) or hyphens (-) allowed, no spaces possible

inseam

Attribute label*

EN inseam

Attribute description

EN

Attribute constraints

Once you save this attribute, the constraint cannot be modified.

○ **None:** no constraints
● **Unique:** different for each variant
○ **Same for all** variants
○ **Combination unique:** the combination of all attributes with this option must be different for each variant.

Attribute type*

Once a product type attribute is saved, the type of attribute cannot be changed.

Select...

Yes / No (boolean)
Text
Number
Money
Date/Time
Reference

Finally, you can model attributes using Custom Fields in commercetools when you don't want to have multiple sub-types of an object. For example, you may want to capture the Twitter handle of all new customers. Rather than creating a "Customer With Twitter" custom type, you'd just add a custom field to capture the Twitter handle.

Product Catalog

If you start by first migrating your product catalog data and related functionality to commercetools, you'll want to perform a one-time export of your enriched data (base data from source ERP, PIM, etc. + business user enrichments for eCommerce) from SAP Commerce using ImpEx, which will export all of your product catalog data to a CSV file. The ImpEx script you'll want to use is something like:

```
$contentCatalog=YourCatalog
$contentCatalogName=Your Catalog
$catalogVersion=catalogversion(catalog(id[default=$contentCatalog]),version[default=’Staged’])[unique=true,default=$contentCatalog:Staged]
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]
```

Validate the script and then run it. It may take a few hours to execute, depending on how large your product catalog is.

Step 4: Customize Platform Behavior

Next, import that data into commercetools. While commercetools does offer its own version of ImpEx, which is functionally similar to SAP Commerce's version, it can't be used for complex data like this. There are inherent differences between commercetools' and SAP Commerce's data models. For example, SAP Commerce uses the concept of a catalog whereas commercetools uses the concept of a channel. The data can largely be re-used but the structure and even syntax (date formats, number formats, etc.) may change. For those reasons, it's easier to use some custom code to parse the CSV you exported from SAP Commerce, extract the data you want and import it into commercetools by calling the appropriate APIs. Java SDK users can use the commercetools-provided Java library or, very easily, another SDK of your choice.

Long-term, you'll still need to keep an up-to-date copy of your product catalog in SAP Commerce because there are "hard" references throughout SAP Commerce to products, categories, etc. It's easiest to continue to push your raw product catalog data from your ERP, PIM or other product catalog master to SAP Commerce as you have traditionally done. In addition to that, start also feeding that data to commercetools, and only allow your business users to further enrich that data in commercetools. The "shells" of un-enriched product catalog data in SAP Commerce are just fine since the product catalog details are all being served from commercetools.

Step 5: Build Custom Extensions

As the term “customization” already suggests, it’s impossible to find a one-size-fits-all solution for all implementations out there. Instead, this little example should serve as a showcase to give you a rough idea of how you could proceed.

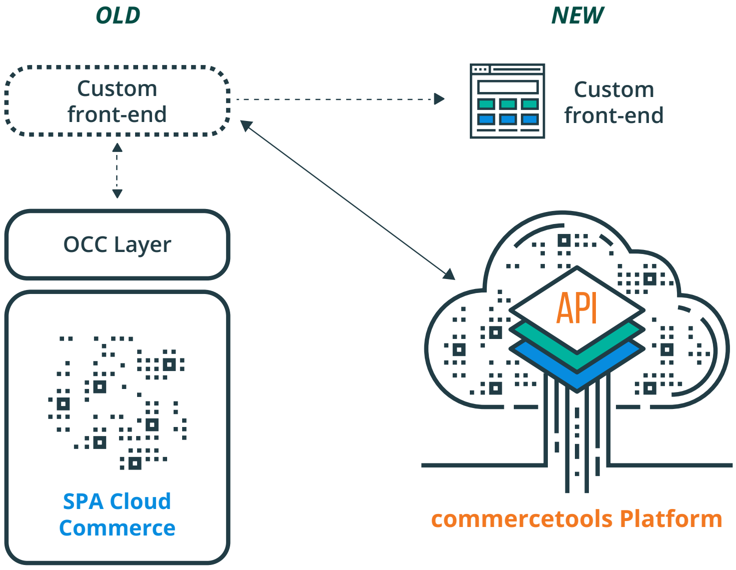
Let’s say you managed to import your product catalog to commercetools and you have made sure that this catalog is regularly synced between the old and the new system. You can access them via the Products endpoints and work with them in any way you want – like displaying them in the Storefront, which is otherwise powered by SAP Commerce. In other words, you’re overriding this part of the process by having commercetools deliver the data – a classic case of Martin Fowler’s strangler pattern.

Step 6: Migrate User Interface

The last item on your migration agenda is the user interface. The term “migrate” is a bit of a misnomer in this context, because in most cases you are facing a complete rebuild – just because SAP Commerce and commercetools are so fundamentally different in this regard. Especially if you have used an SAP Commerce accelerator as the basis for your frontend, it’s not possible to carve out and re-use the frontend code – because the storefront is tightly connected to the SAP Commerce core.

SAP Commerce Cloud	commercetools
Legacy storefront application (JSP + command line)	Headless commerce platform
Accelerators for some verticals	Ready-made Integration Marketplace extensions and integrations Creating your own custom extensions and integrations is also possible
Omni Commerce Connect (OCC layer), API-layer covering a fraction of SAP Commerce’s core features	API-first approach, 100% functional coverage

The only context in which a frontend migration might be possible is if you already have a custom frontend based on a technology such as Angular or React and use the SAP Commerce OCC layer. In this scenario, the strangler pattern shown in the previous step can help with UX building as well. By identifying how the frontend sends data to and receives data from the SAP Commerce API layer, it is then possible to substitute the original endpoints with the ones provided by the commercetools platform.



Blueprint Architecture

At commercetools, we know there can be a learning curve with a new platform and introducing your team to many new concepts. Based on that, we provide a Blueprint Architecture that provides real examples of all the concepts we've discussed in this document. This is a great resource for our customers and can shave six to eight weeks off your implementation.

You can read the Blueprint Architecture that is available as a separate downloadable file.

Conclusion

In this white paper, we have suggested a framework for helping you migrate your digital assets from SAP Commerce to commercetools. As we have mentioned before, this is not a one-click-solution but a project involving many aspects. Especially when it comes to moving custom functionality and individual user interfaces, most artifacts have to be built from scratch. A positive point is that the migration of data – especially catalog data – can be at least semi-automated. Last but not least, this kind of solution allows you to build a highly effective and scalable network of services, enabling you to innovate and grow your business.

To stay relevant for their customers, they cannot have technical teams deal with maintenance and updates – instead, they need to build customer-facing features which generate real tangible business value.

What’s Next?

60-day Trial

The commercetools trial provides risk-free evaluation with fully-functional access to our platform for 60 days at no cost. The trial includes access to our APIs, Merchant Center administration interface, and our tools/connectors for integrations. We believe in full transparency with our customers and we are the only enterprise commerce platform offering this level of self-service evaluation. Along with access to the platform, you can also access our documentation, tutorials, release notes, and platform status without the need to create an account. Get started today at commercetools.com!

About commercetools

commercetools is a next-generation software technology company that offers a true cloud commerce platform, providing the building blocks for the new digital commerce age. Our leading-edge API approach helps retailers create brand value by empowering commerce teams to design unique and engaging digital commerce experiences everywhere – today and in the future. Our agile, componentized architecture improves profitability by significantly reducing development time and resources required to migrate to modern commerce technology to meet new customer demands.

The innovative platform design enables commerce possibilities for the future by offering the option to either use the platform's entire set of features or deploy individual services, á la carte over time. This state-of-the-art architecture is the perfect starting point for customized microservices, enabling retailers to significantly reduce time-to-market for innovative commerce functionalities.

With offices in Germany and the United States, as well as presence across general Europe and Asia Pacific/Oceania, B2C and B2B companies from across the globe including well-known brands across many industries, including fashion, food and retail, trust commercetools to power their digital commerce business.

Europe
commercetools GmbH
Adams-Lehmann-Str. 44
80797 Munich
Germany
Phone: +49 89 9982996-0
Email: mail@commercetools.com

America
commercetools Inc.
American Tobacco Campus | Reed Building
324 Blackwell St. Suite 120
Durham, NC 27701
Phone: +1 212-220-3809
Email: mail@commercetools.com

www.commercetools.com
Munich - Berlin - Jena - Amsterdam - London - Durham NC - Singapore - Melbourne