

Migrating from Magento[®] to commercetools

Executive Summary

This white paper leads architects and their teams through the platform migration process from Magento® to commercetools. It talks about which steps are necessary for 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 paths of communication between consumers and suppliers continue to appear. Take Voice Commerce: Within only a few years, more than 100 million smart speakers have been sold. Similar to the app store which helped Apple introduce the mobile age, Amazon lets 3rd-party developers build new capabilities on top of their voice platform. Brands and retailers need to be able to build new prototypes quickly, experiment, 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: move to another, more flexible architecture.

In this whitepaper, we look specifically at two platform vendors: First is the well-established Magento platform, which is recognized as one of the market leaders by Forrester and Gartner. Second is the innovative, cloud-native commercetools platform. We will suggest a strategy on how to move from Magento to commercetools without disrupting the daily business or jeopardizing operations.

Of course there is no standard template that can be used exactly the same way in every migration. Every project is different and has its own caveats, and we're by no means suggesting a cookie-cutter approach. Rather, the following strategy is meant as a guiding principle or a framework which, from practical experience, tremendously increases the chance for success.

Commerce Platform Migration

Magento In a nutshell

Magento was released in 2008 by the US company Varien as an open-source commerce platform with multi-store functionality and plugin extensibility. A little later on, commercial licensing became available as well. The solution has been acquired by eBay in 2011, sold to a private equity funds in 2015 and is now part of Adobe®. Technologically, it's a PHP-based software monolith, built on top of the Zend Framework and using MySQL as its data layer. It comes with its own templating engine and has a few areas covered by its SOAP API.

Adobe is integrating Magento into its cloud offering, connecting with tools such as the Adobe Experience Manager. The future of the open-source version of Magento is currently unclear.

commercetools: Innovative Cloud-Native Platform

As a young company with an innovative product, commercetools has the need for flexibility in its DNA. It offers a cloud-native, multi-tenant, headless commerce solution, with auto-scaling functionality and continuous updates. It has a flexible API which brands, merchants and system integrators can use as microservices. Despite being relatively new

to the market, commercetools has already made its appearance in all of the three major analysts' reports (Gartner, Forrester, IDC) and running major brands around the world, including major global enterprises such as AUDI, Bang & Olufsen, and EXPRESS.

Reasons for Wanting to Migrate

Lack of Agility and Speed

Because of the complex nature and the strict architecture of the Magento 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 the complex templating system and the fact that data modeling is done via XML configuration.

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

Magento 1 vs. Magento 2

One of the most pressing challenges facing brands and merchants who are using the platform is the fact that the company is ending support for Magento 1 by mid-2020. Adobe is urging clients to move to Magento 2, which was released in 2015. It has been reported more than once that Magento 2 still hasn't reached the level of product maturity which Magento 1 has gained over the past years. Plus, performance tests suggest that the new version of Magento is actually slower than Magento 1. And, because the structure of the templating system again has become more complex, development time for new features or template changes increased significantly when compared to Magento 1.

Also, moving from 1.x to 2.x is not as straightforward as it sounds. Although both branches share characteristics, a migration is

actually a greenfield project. Developers need to be re-educated, there are new certification programs as well. So, taking all this into account, it does not come as a surprise that brands and retailers are hesitant to migrate:

- They have to invest in an almost entirely new technology stack, which offers less functionality
- The new solution is notably slower
- More development time is needed
- No new features which would add significant business benefits

Last but not least, the budget for running and maintaining a Magento installation has to be kept in mind as well. The platform needs to be run on multiple application nodes and database servers, generating higher operating costs for hosting applications.

	Magento	commercetools
Licensing	Installation-based	Usage-based
Operations	Multiple application and database nodes necessary to run at scale	Platform run and scaled by commercetools; only frontend hosting as external cost
Updates & Upgrades	Compulsory updates a few times a year	Continuous Integration, features pushed daily, non-breaking API changes

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 for disrupting 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 on Magento, a good time for evaluating other software solutions is now as there is uncertainty about the platform's future. If they know that this involves investing in new developments anyway (see section Magento 1 vs. Magento 2), they might as well opt for a complete change and consider moving to a technology such as commercetools.

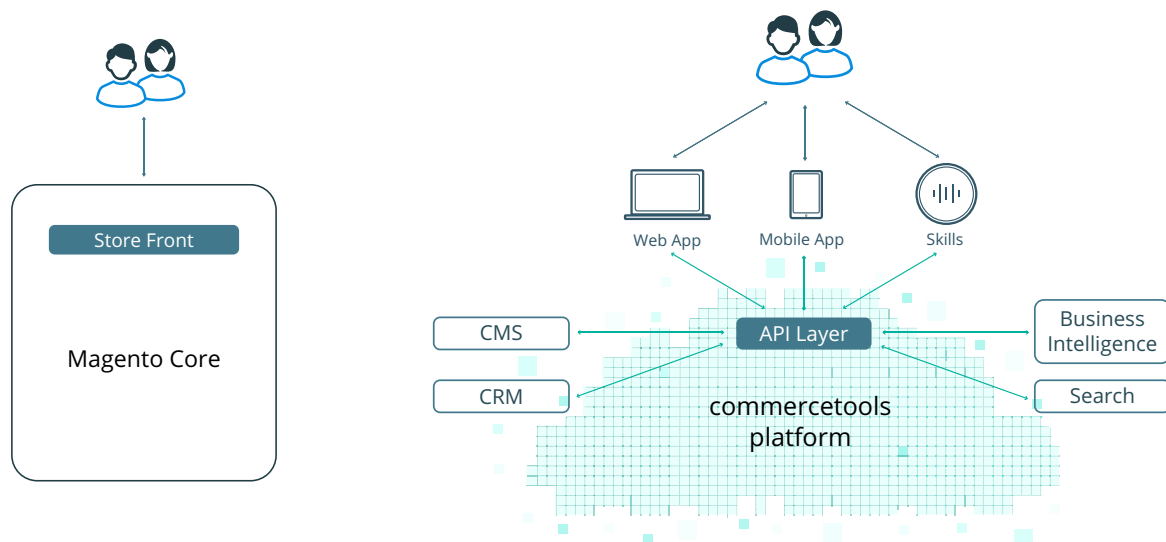
Let's now dive deeper into how such a migration is planned and carried out in practice.

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. Everything that was possible in the old solution, including all the features that professionals worked on in the old system, would be there in the new solution as well – plus of course the benefits of the new platform. In most circumstances, as we will see a little later on, this is not practical. When planning the migration, this is the perfect chance to eliminate overhead which had not been used and generally aim for a leaner solution.

What we’re suggesting here is to divide an existing project into business domains, and transfer the respective functionality and data out of Magento 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. We are also aiming at a phased migration (instead of a big bang approach) to disturb operations as little as possible and mitigate risks as much as 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 in the background? This 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 what 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 in 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 find detailed information in the API documentation (<https://docs.commercetools.com/http-api.html>). 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 those cases, 3rd party services need to be integrated.
- Customized : These are the parts which 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, data scientists – the “new cloud world” requires the work of cross-functional and vertical teams.

Step 2:

Build 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 Magento database to commercetools. Those data are:

- 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 those 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 template engine to power the storefront like in Magento although we offer a quickstart template called SUNRISE (<https://commercetools.com/sunrise>), which can be used to build customized web or mobile apps.

Note that those three areas do not have to be tackled precisely in this sequence. They only serve to add a bit of structure to your planning.

Step 3:

Extract Data

Begin by exporting your product data out of Magento, using the built-in import-export functionality.

- Run exports for products as CSV
- Modify files (map to new data model)
- Create drafts on modified export CSV
- Load data

Modelling Products

When trying to migrate product data from Magento to commercetools, you need to keep in mind that the data is modelled and persisted in a different way. While Magento is using MySQL

with a so-called Entity Attribute Value (EAV) setup, commercetools is based on a schemaless database logic. This allows for real-time updates to its structure, so you can change the data model in real-time using the Merchant Center or directly on API level.

As both systems use slightly different terminology, this can be confusing when preparing the migration. The following table provides both the Magento and the commercetools counterparts.

Magento	commercetools	Description
Entity	Object	Carries a specific type of information, such as "product", "order". Each object is described by its attributes.
Product type (eg. simple, configurable)	Product Type	Describes how the product itself is set up, eg. if it has variants or not
Product	Product with a Master Variant	A virtual "umbrella" product which exists to present information to the customer ("t-shirt"). It's not buyable itself, but its variants are.
Variant	Variant	The actual product which can be bought ("t-shirt white size L")
Attribute	Attribute	A product property, such as "size" or "color"
Attribute Value	Attribute Value	The value which is given by the property, such as "L" or "blue"
Attribute Set	Product Type ¹	A collection of attributes suited to describe a specific product

¹ <https://docs.commercetools.com/http-api-projects-productTypes.html>

Mapping Objects

In the Magento EAV model there are a number of standard entities which are used to model different kinds of data, such as products, sales, etc. You can use the following mapping as a guideline. Of course, those mappings are always specific to each project.

Magento	commercetools
customer	Customers
customer_address	Address
catalog_category	Categories
catalog_product_entity	Products
creditmemo	Payments TransactionType
invoice	(usually provided by external PSP)
order	Order
shipment	Delivery ¹ Parcel ²

² <https://docs.commercetools.com/http-api-projects-orders.html#delivery>

³ <https://docs.commercetools.com/http-api-projects-orders.html#parcel>

Attribute Code	Attribute Label	Required	System	Visible	Scope	Searchable	Use in Layered Navigation	Comparable
accessories_size	Accessories Size	No	No	No	Global	Yes	Filterable (with results)	No
accessories_type	Accessories Type	No	No	No	Global	Yes	No	No
apparel_type	Type	Yes	No	Yes	Global	Yes	Filterable (with results)	No
author_artist	Author/Artist	No	No	Yes	Global	Yes	No	No
bag_luggage_type	Bag & Luggage Type	No	No	No	Global	Yes	Filterable (with results)	No
bedding_pattern	Bedding Pattern	No	No	No	Global	Yes	No	No
bed_bath_type	Bed & Bath Type	No	No	Yes	Global	No	Filterable (with results)	No
books_music_type	Books & Music Type	No	No	Yes	Store View	Yes	No	No
camera_megapixels	Camera Megapixels	No	No	No	Global	Yes	No	No
camera_type	Camera Type	No	No	No	Global	Yes	No	No
color	Color	No	No	Yes	Global	Yes	Filterable (with results)	Yes
cost	Cost	No	No	No	Website	No	No	No
country_of_manufacture	Country of Manufacture	No	Yes	No	Website	No	No	No
custom_design	Custom Design	No	Yes	No	Store View	No	No	No
custom_design_from	Active From	No	Yes	No	Store View	No	No	No
custom_design_to	Active To	No	Yes	No	Store View	No	No	No
custom_layout_update	Custom Layout Update	No	Yes	No	Store View	No	No	No
decor_type	Decor Type	No	No	Yes	Global	Yes	Filterable (with results)	No
delivery_time	Lieferzeit	No	No	Yes	Store View	No	No	Yes
description	Description	Yes	Yes	No	Store View	Yes	No	Yes

Product attributes in Magento

Custom Objects

In commercetools, when there is no standard entity to hold your data, you can always generate a so-called custom object. Those 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. You will find detailed information in our API documentation.

- Tutorial: <https://docs.commercetools.com/tutorial-custom-types.html>
- Documentation on types: <https://docs.commercetools.com/http-api-projects-types.html>

Mapping Attributes

As mentioned before, both Magento and commercetools use a standardized system to organize and link different attributes. In order to prepare your commercetools project to hold the required product attributes, navigate to the product type settings in the Merchant Center and copy your products' attributes accordingly.

[To Product types list](#)

Fashion

Name*
Fashion

Description*
Fashion products

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

Attributes (3) [+ Add attribute](#)

Attribute Name	Required	Constraint	Type	Set	Searchable	
size	<input checked="" type="checkbox"/>	Unique	List (enum)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Delete"/>
color	<input checked="" type="checkbox"/>	Unique	List (enum)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Delete"/>
design	<input type="checkbox"/>	None	Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Delete"/>

200 Items per page (3 items) [Page 1 of 1](#)

Product attributes in the commercetools Merchant Center

Custom Fields

You can also model attributes using “custom fields”⁴ 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 e-commerce) from Magento using Dataflow, which will export all of your product catalog data to a CSV file.

4 <https://docs.commercetools.com/http-api-projects-custom-fields.html>

Magento Admin Panel Global Record Search Logged in as demoshop | Wednesday, September 25, 2019 | [Log Out](#)

Dashboard Sales Catalog Customers Promotions Newsletter CMS Reports **System** [Get help for this page](#)

Import/Export Profile [Export All Products](#) [Back](#) [Reset](#) [Delete Profile](#) [Save Profile](#) [Save and Continue Edit](#)

Profile Wizard

- Run Profile
- Profile Actions XML
- Profile History

Profile Information

Name: * Export All Products

Entity type: Products

Direction: Export

Store: Default (Admin) Values

File Information

Data transfer: Local/Remote Server

Type: Local Server

File name: export_all_products.csv

Path: var/export

(For Type “Local Server” need to use relative path to Magento install var/export or var/import, e.g. var/export, var/import, var/export/some/dir, var/import/some/dir)

Data Format

Type: CSV / Tab separated

Value Delimiter: , (\\ for tab)

Enclose Values In: " Warning! Empty value can cause problems with CSV format.

Original Magento attribute names in first row: Yes

Export: All fields

It may take a few hours to execute, depending on how large your product catalog is.

Step 4:

Import and Verify Data

Next, import your data into commercetools. While commercetools does offer its own version of ImpEx, which is functionally similar to Magento's version, it can't be used for complex data like this. There are inherent differences between commercetools' and Magento's data models. For example, Magento 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 Magento, extract the data you want, and import it into commercetools by calling the appropriate APIs. We provide a Java SDK⁵ as well as SDKs in other programming languages⁶.

Long-term, you'll still need to keep an up-to-date copy of your product catalog in Magento because there are "hard" references throughout Magento 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 Magento, 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 Magento is 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 Products⁷ endpoints and work with them in any way you want – like displaying them in the Storefront which is otherwise powered by Magento. 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 pattern⁸.

⁵ <https://github.com/commercetools/commercetools-sync-java>

⁶ <https://docs.commercetools.com/software-development-kits.html>

⁷ <https://docs.commercetools.com/http-api-projects-products>

⁸ <https://martinfowler.com/bliki/StranglerFigApplication.html>

Step 6:

Migrate User Interface

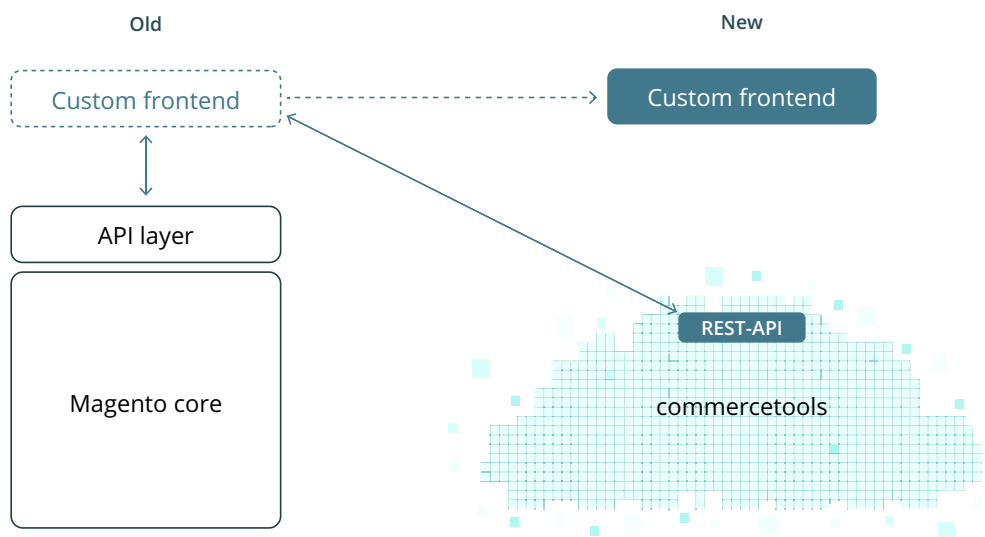
The last item on your migration agenda is the user interface. The term “migrate” is a bit of a misnomer in this context, because in most cases you are facing a complete rebuild – just because Magento and commercetools are so fundamentally different. Especially if you have used a highly

customized Magento frontend, it’s not possible to carve out and re-use the frontend code – because the storefront is tightly connected to the Magento core.

Magento	commercetools
Suite approach, frontend and backend tightly integrated	Headless commerce platform, SUNRISE Quickstart template
Functionality is partly consumable via APIs	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 Magento API. 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 Magento API layer, it is then possible to substitute the original endpoints with the ones provided by the commercetools platform.



Conclusion

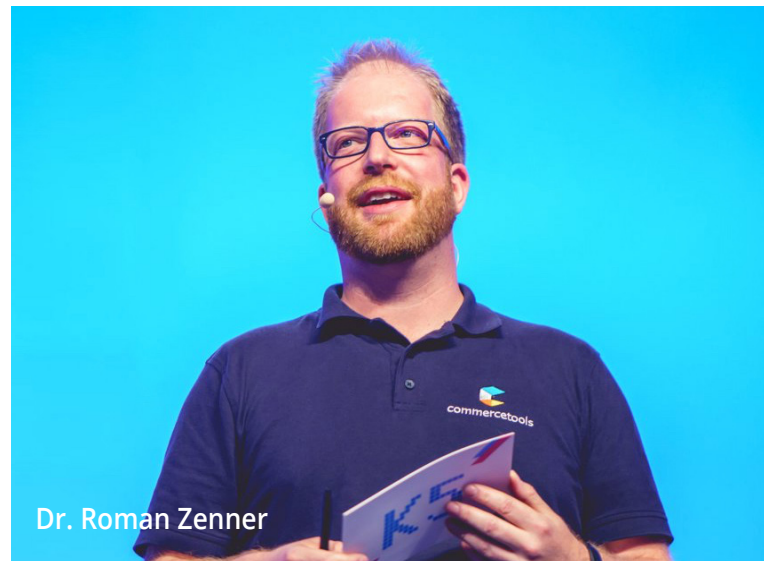
In this white paper, we have suggested a framework for helping you migrate your digital assets from Magento to commercetools. As we have mentioned before, this is not a one-click-solution but a project involving many aspects. Especially when it comes to moving custom functionality and individual user interfaces, in most cases 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, businesses cannot have technical teams deal with maintenance and updates – instead, they need to build customer-facing features which generate real tangible business value.

About the author

Since 2001, Dr. Roman Zenner works as an author, consultant, and speaker in e-commerce. He has written several books on web shop software and regularly publishes articles in professional magazines and blogs. Dr. Zenner runs shoptechblog.de as well as the podcast ShopTechTalks. Furthermore, he speaks at conferences, teaches university classes and moderates expert panels.

In his work, Roman focuses on next generation commerce technologies and explores what retail will look like in a post-web world. Since 2015, he is a full-time employee of commercetools GmbH, working as an Industry Analyst.



About commercetools

commercetools is a next-generation software company that offers a cloud-based, headless commerce platform, providing the building blocks for the new digital commerce age. Founded in 2006 commercetools is one of the fastest growing enterprise software companies in Europe with 180 employees at its European and US offices serving international brands such as Bang & Olufsen, Carhartt WIP, C.H. Beck, Cimpres and Express.

commercetools GmbH
Adams-Lehmann-Str. 44
80797 Munich, Germany
Phone: +49 (89) 9982996-0
Email: marketing@commercetools.de

www.commercetools.com

Munich - Berlin - Jena - Amsterdam - London - Durham NC