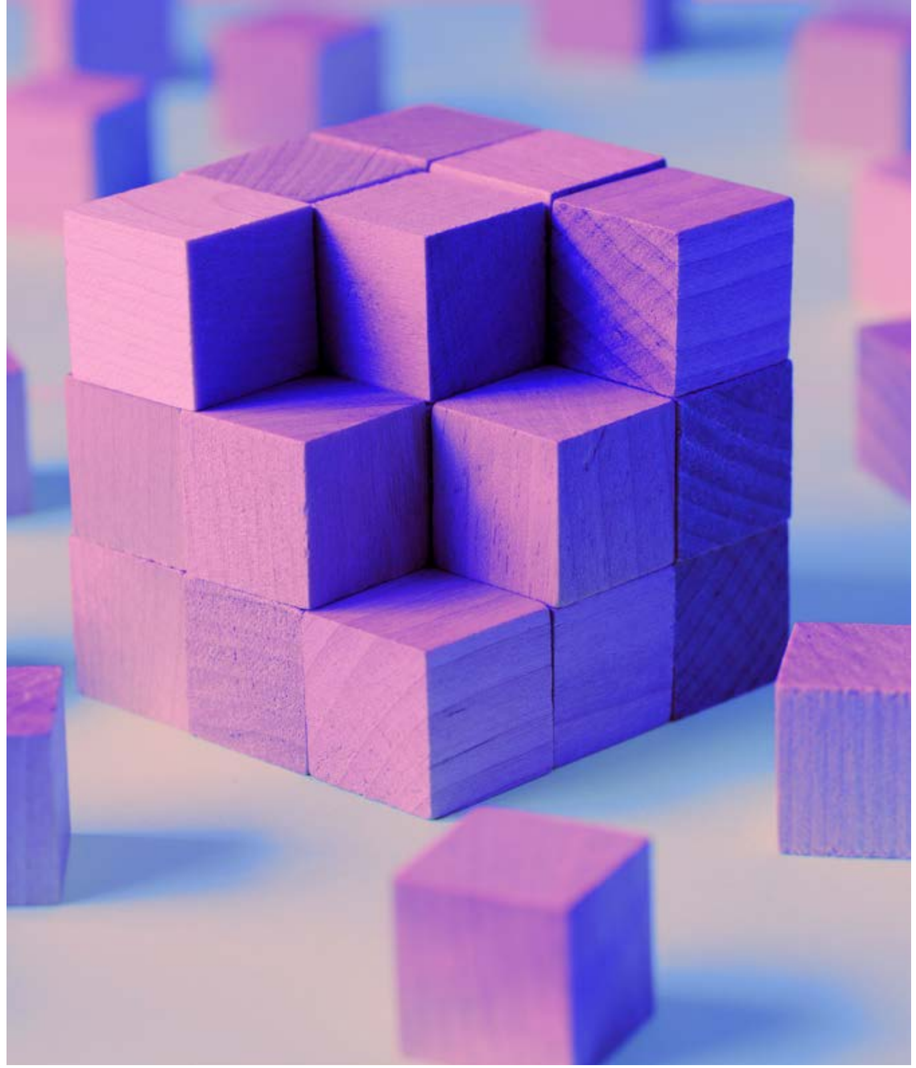
Expert Guide

COMPOSABLE COMMERCE

It’s time to get picky.

TURBINE KREUZBERG



HELLO, COMPOSABLE COMMERCE.

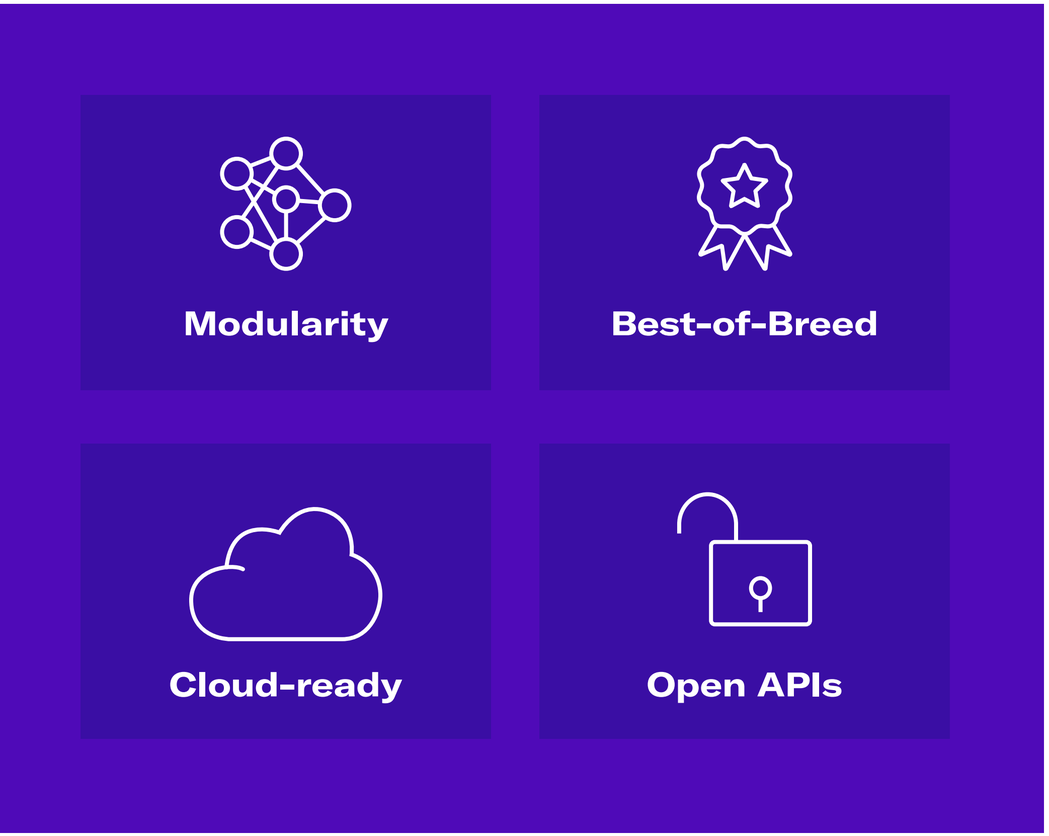
No question, the field of e-commerce technology moves fast. Offering customers innovative, value-added services packaged in a modern user experience means constantly evaluating, testing, and integrating new technologies.

Still, completely revamping an existing commerce solution every few years – i.e. repeatedly implementing large replatforming projects – is likely to cause irritation in the vast majority of companies. That's where **Composable Commerce** comes in. Going beyond just another e-commerce trend, Composable Commerce marks a real shift in technological paradigms – one that will impact how companies think about commerce for the next decade.

In a nutshell, composable commerce means building flexible, adaptable system architectures that can integrate a vast number of specialized software services, each deemed the best in their field for the requirements at hand.

TURBINE KREUZBERG

There are **four main driving forces** behind composable commerce.



First, commerce software, like most of today's software on the web, is increasingly developed with an **API-first approach**. This means systems are developed to have open, standardized interfaces that make it easy to integrate them with other services.

In addition, this has allowed the ecosystems around commerce systems to grow enormously, giving companies lots of **options** to choose from.

Third, most services increasingly run natively in the **cloud** instead of on-premise, which makes them more scalable and reduces the need to invest in large-scale operations.

Finally, modern commerce systems are highly **modular**, with some even using a full microservice approach, and consist of numerous independent, loosely coupled components that can be exchanged or integrated at will.

TURBINE KREUZBERG

Composable Commerce describes a flexible approach to building e-commerce applications:

Companies select best-of-breed, independent components and combine them to build applications **tailored to their specific requirements**, gaining adaptability and speed in development along the way.

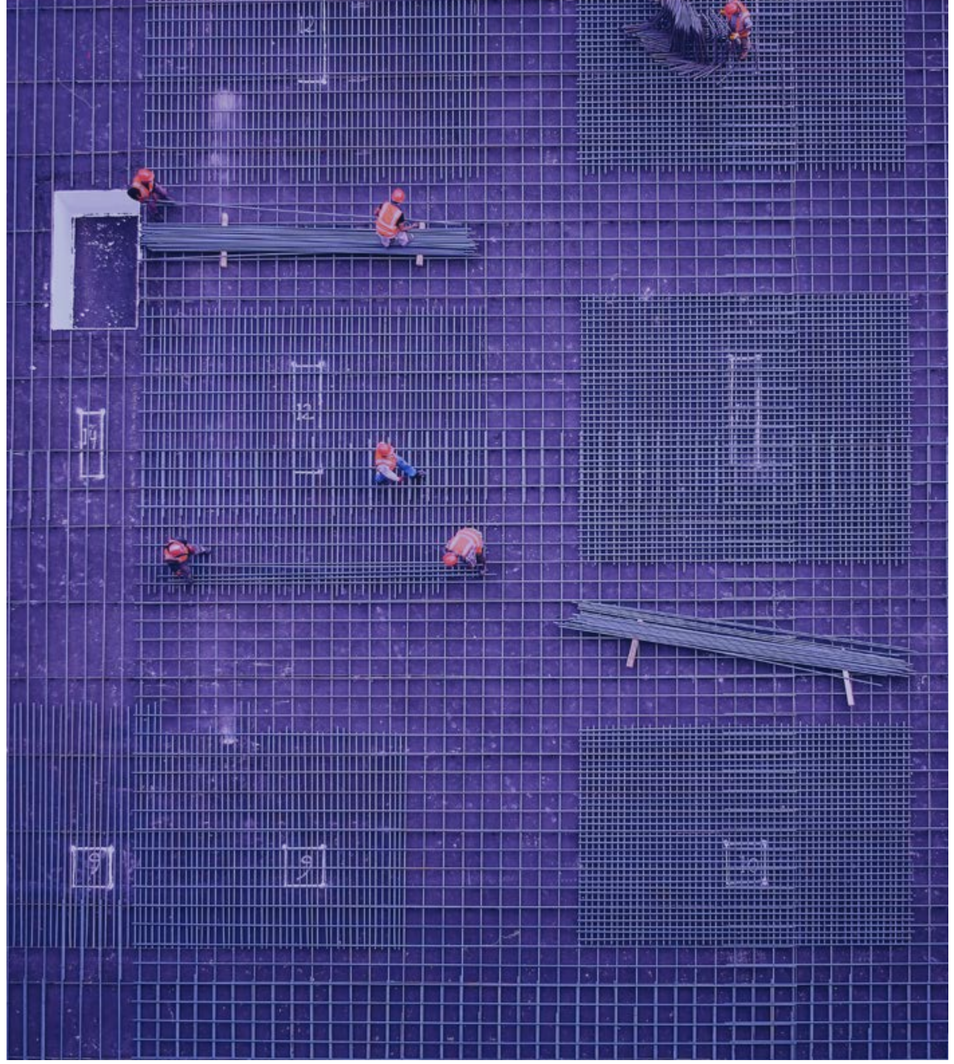
TURBINE KREUZBERG

MAXIMUM FLEXIBILITY. ACROSS THE BOARD.

In essence, **Composable Commerce** means evaluating the best software solution for specific requirements in digital commerce and combining technologies to create truly custom commerce platforms. **There are numerous benefits** to doing this, the first of which is a gain in overall flexibility. With an architecture set up for composable commerce, companies that want to improve a single feature or optimize a specific process can exchange parts of the IT infrastructure relatively easily. This makes it easier to keep systems up to date, while eschewing the need for large replatforming projects that would otherwise have been necessary.

Such flexible system architectures are much more responsive to changing external and internal circumstances. Moreover, open, standardized APIs mean companies can shape their tech stacks much more actively – as well as implement changes more quickly, validate their value, and reduce the time-to-market.

TURBINE KREUZBERG



In addition, relying on largely **independent services** makes companies much less dependent on individual vendors.

Previously, companies – especially SMEs – were reluctant to go with a wide range of vendors for fear of greater complexity. Vendors of off-the-shelf suite solutions would argue that their solutions offer lower complexity and effort, because everythig comes in one package.

So, companies would put their trust in all-in-one commerce solutions, which come with strong lock-in effects – meaning they could only change providers with great effort.

Lock-in effects were not only caused by license fees and contract terms, but could also result from the fact that data migration could become very time-consuming when replacing an entire commerce system suite.

The use of modular systems with a multitude of standardized interfaces eliminates these problems.

TURBINE KREUZBERG

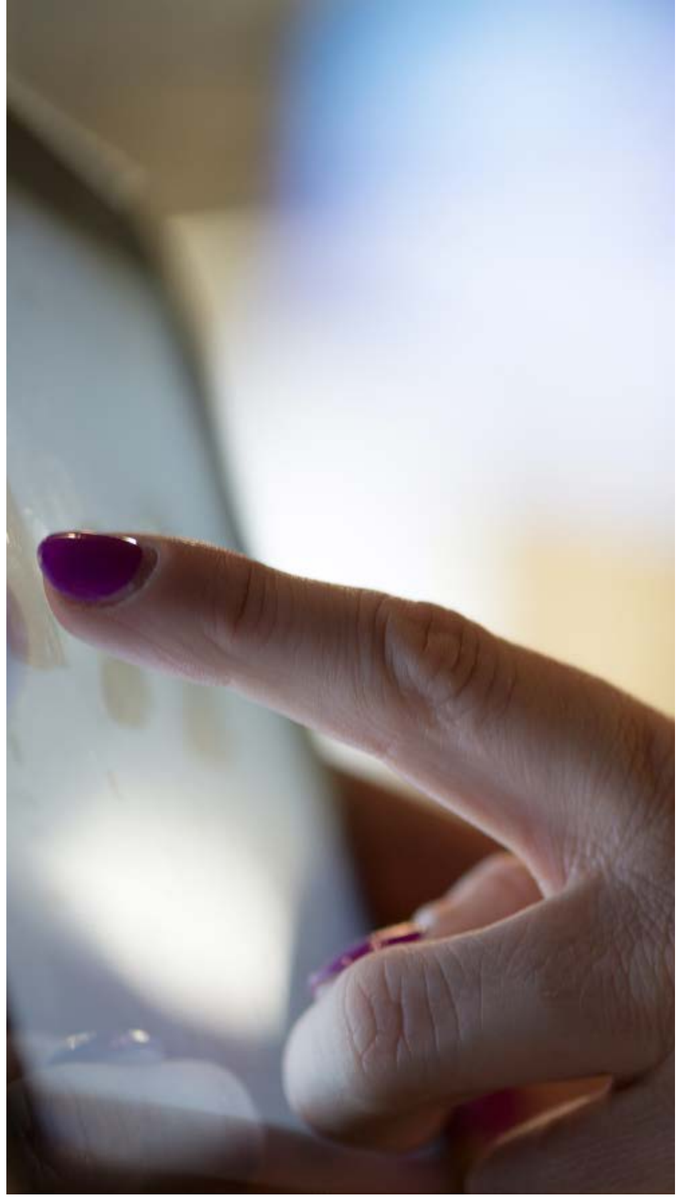
What effect does that have? instead of taking what's offered and having to make do, selecting the right technologies can now follow a **"best-of-breed" approach**.

Anyone fearing that Composable Commerce will increase the complexity of technology selection can rest assured: it's actually the opposite that is true. Generally, selecting the right technologies becomes easier, since it is no longer necessary to find that perfect sweet-spot, at which a specific service's capabilities intersects with the requirements of the entire project.

Instead, you can use the best solution for each functionality individually, for example, for recommendation software, search or payment solutions.

Plus, quite often, suite products can only be adapted to custom requirements with greater effort and using complex workarounds. Composable Commerce, makes technology selection much more targeted – and only those solutions that meet your needs best get implemented.

Selecting the right technologies can now follow a "best-of-breed" approach.



TURBINE KREUZBERG

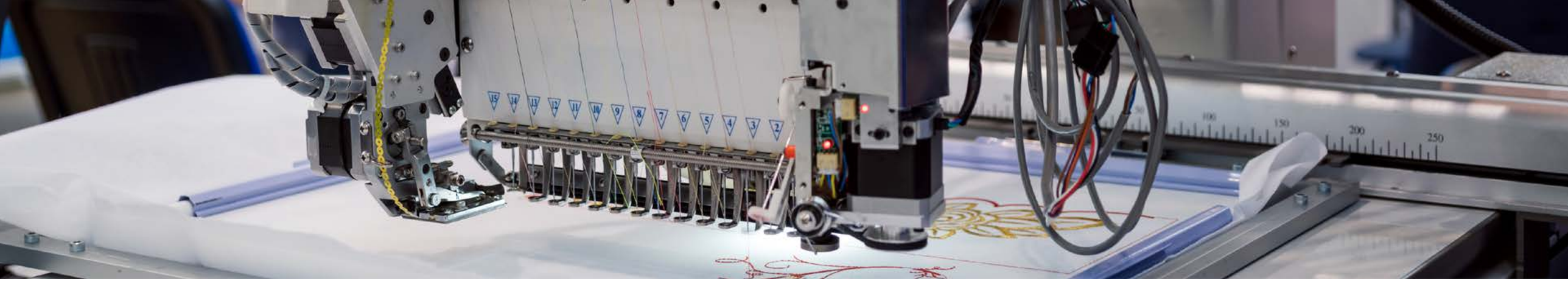
So, how can you get started? One of the first – and arguably most important – steps in any commerce project is **selecting a core commerce system**. This is an impactful decision, since the entire rest of the platform will get built upon this system. The core commerce system defines which ecosystem all other components can be drawn from, which technologies will need to be implemented, and which skills the implementing team needs to posess.

The basic requirement commerce systems need to fulfill for Composable is **modularity**. That's why platforms like **Spryker** or **commercetools** are highly recommended for Composable Commerce projects. Their technological approaches differ, though, and whether one or the other makes sense for a particular use case depends on individual requirements.

Generally, microservice-based commercetools is geared more towards consumer brands at the enterprise level and requires frontend-heavy development.

Spryker is aimed at more complex commerce requirements, such as for B2B procurement platforms or B2B marketplaces. Spryker's strength lies in its Packaged Business Capabilities (PBC) – independent features that are grouped into larger related clusters – that make integrations faster than with other monolothic systems and more manageable than microservices.

TURBINE KREUZBERG



From there, **Composable means having options – and getting picky**. These core commerce systems bring ecosystems of third-party services with them and companies have to select which ones make the most sense for them.

At this point, developers have already determined which services are “best-of-breed” technologically and how to implement them, for example which billing engine is most suited for which commerce system and how to integrate it.

There is not an unlimited number of options when selecting technologies for individual functionality. The core system and the service’s integration capability remain limiting factors. **Usually, there are five to six options available per field.** This makes it easier to select the right solution – while at the same time maintaining a high degree of flexibility.

For example, when it comes to search, the choice is usually between three providers with very different approaches: Elasticsearch, FACTFinder or Algolia.

Ultimately, the question to be answered is whether your search implementation needs to be manually adjusted down to the smallest detail or whether it can be completely automated and reliant on AI.

Despite having just a handful of options, the freedom of choice is still greater than with all-in-one suites, where one would have to rely on the built-in search function. In addition, these self-contained services are significantly more stable than suite applications, which often require a large amount of custom development.

It’s important to remember: in an all-in-one suite, there would be no choice at all.

With the right services selected, **Composable Commerce becomes all about integration** – and lots of it. This has also been the case with legacy software, but to a much lesser degree. The Composable paradigm means there are more systems that need to be integrated with each other. When starting from scratch, this requires a lot of time. After installing the core commerce system, teams should already start integrating the first systems.

Much of this work can be done in parallel and much more so than in the past. Here, it can be a good strategy to divide and conquer, i.e. one developer or pair takes care of payment, another handles search, and a third takes care of integrating core commerce functionality.

Still, integrations themselves are becoming easier and easier to implement from a technical point of view. This is driven by standardized, thoroughly documented APIs. For a reasonably seasoned developer team, this should not pose too many problems.

TURBINE KREUZBERG

Finally, since Composable Commerce brings together many different services, **planning** capabilities and **orchestration** become that much more important. This becomes apparent early on in the run-up to implementation, since contracting can become quite time consuming. Put simply, companies have to deal with many more vendors at once in composable commerce projects, enter into contract agreements with them, and work with them throughout.

Essentially, what you gain is a distribution of risk. You're not putting all your eggs in one basket, but in many baskets simultaneously. This requires skilled, experienced project management and less hands-on technical capability.

At the same time, since individual services are often cloud-based, the need to build up large-scale operations in-house becomes much smaller. Technical expertise, though, is still required – just in different ways. For example, an overarching monitoring set-up is highly recommended in order to keep an eye on all services, make sure they work well together, and that incidence cases can be resolved quickly, also by communicating faults directly to the responsible provider. ■

TURBINE KREUZBERG

READY TO START COMPOSING?

We’ve got you covered. Let’s talk.

Christopher Möhle, COO

christopher.moehle@turbinekreuzberg.com
+49 175 321 0 941

TURBINE KREUZBERG