The road to composable

PRODUCT LEAGUE

A movement is taking place in the field of eCommerce. Something that we at Product League call “**the road to composable**”. While running, building and scaling your digital business, you might not even have known you were on this road. You might think that all the problems that arise with every new eCommerce opportunity are yours and yours alone. Rest assured - we are all on this road together.

Over the course of this whitepaper, we will discover the different stages of composable commerce. Let’s explore the background of the eCommerce landscape and its direction, to see what this road consists of and how to navigate its stops, pitfalls and vistas.

Your customers are digital entities

Next time you're in the supermarket (or really any retail store), pay attention to your fellow shoppers. Unless they're already hauling three crying toddlers through the aisles, you'll notice most customers with one hand on a cart and the other on their phone. Checking the latest deals, receiving the latest coupons, checking prices of competitors ... And looking back up just in time to swerve around you, if you're lucky! They are having digital shopping experiences. And the experiences continue at home, on the couch, in the metro, at the gym. eCommerce is now the norm, not the exception. Which is probably why year over year eCommerce revenue is still growing twice as fast as brick-and-mortar retail revenue, even after Covid. Your customers are now digital entities, and you better believe that their perception of your brand is digital too.

The road to composable commerce

You know your customers and you know your changing (digital) business. Your company has built up the eCommerce channel over the past years. Maybe as a digital native, delivering your goods out of your garage at first and scaling to multiple warehouses. Maybe as a retailer that became successful with brick-and-mortar shops, knowing the opportunities that were being missed online. This may have set your foot on the road to composable. **The road to composable consists of the digital solutions your business has found to the problems of serving your customers in this new channel.** There were never right or wrong answers, which is why it is a journey, not a destination. The best solutions to the digital problems have evolved over time, both with the growing technical capabilities out there, and with your scaling business.

First stop - The Monolith

How do you offer the same performance to user 1 million, as you did to user 1000? And how do you reliably offer the experiences that make your business unique, to all your customers equally?

Second stop - Going Headless

Is your monolithic eCommerce engine bursting at the seams? And is the word "technical debt" heard in most conversations about your roadmap?

Third stop - Composable Freedom

Instead of one big decision every 5 years (do we continue/upgrade our monolith, or do we do something different?), you are now making many small decisions.

First stop - The monolith.

Traditionally, eCommerce was solved with one-stop-shop applications, the so-called monoliths. The business needed a digital channel, so they bought or built a platform that promised to deliver most of the functionalities that were needed. The customization that was unique to your business added some sauce on top and you were good to go. This approach works relatively well for small businesses: Since spending is limited, customization and performance are not the highest priority. If you're a corner shop bakery and you want to start a digital channel, a relatively cheap off-the-shelf option like Shopify is still your best choice.

Even large companies such as Amazon (Cadabra), Adidas and Nike (Demandware / Salesforce) started out with large monolithic systems. In fact, most companies start out with such a shop (the Magento-, SAP Hybris-, Intershop customers of this world). It is only when the eCommerce channels start growing and scaling, you see the first growth pains appear.

Scaling - Your eCommerce channel has trouble keeping up
Digital experience is the way to your customer's heart and wallet. Research by Baymard shows that 75% of customers will reward a good digital experience by ordering more, while 50% of the time customers punish a bad experience by switching to a competitor! Not surprising, since moving to the competitor is as easy as opening a new tab in your browser, or simply switching apps.

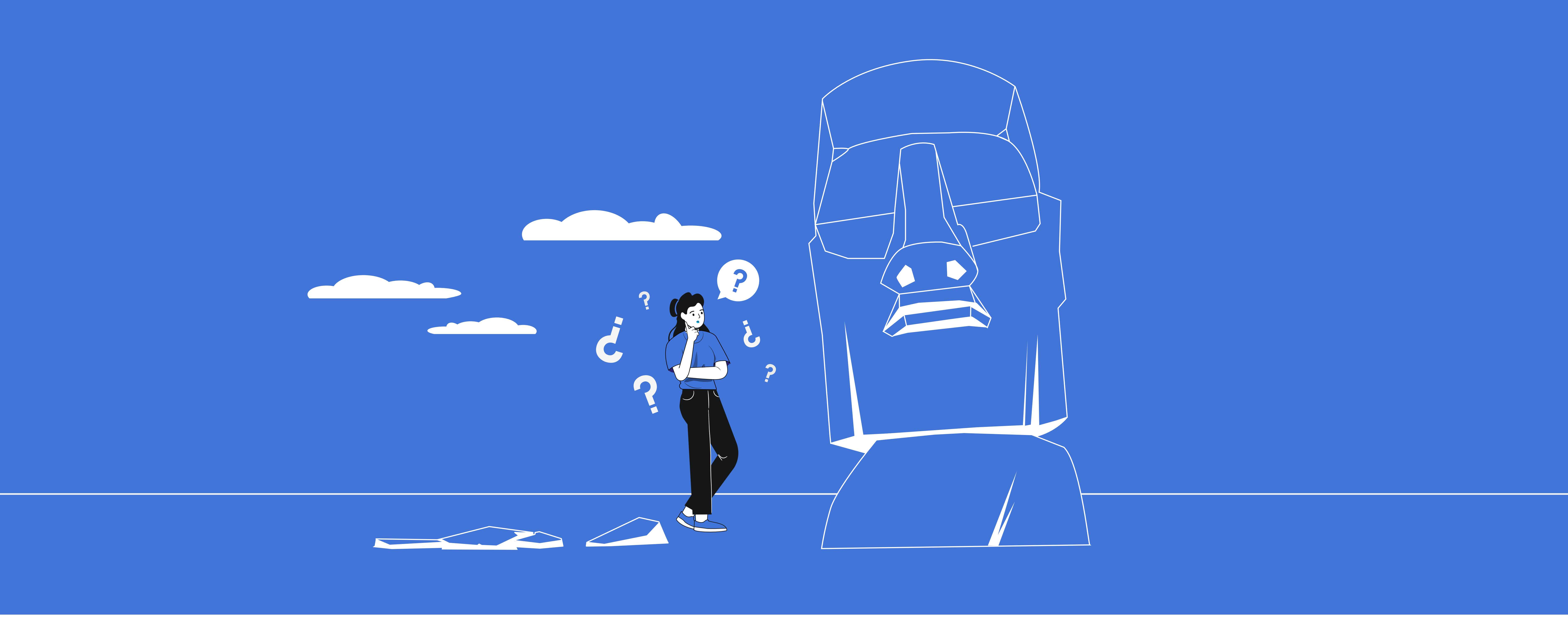
The need for flawless, tailored digital experiences is rising. So, what does this look like behind the scenes? Most likely you're spending more on content, to showcase all the goods that your brand offers and attract customers to your site and app. Perhaps your product data has trouble keeping up, with specific pricing and timebound promotions rules that your customers are waiting for, but which are destroying your site load times on calculation. Or is it customer loyalty that's lacking? Either way, that should have been solved by that digital savings app that was too expensive to release, or the membership program that's been lagging behind the competition since its launch.

Demands? Your backlog is probably full of them. When your current systems are stretched to their limits, technical debt is racking up with every new business feature you add. The total cost of ownership (TCO) of all these complex processes and the people operating them is eating all your margin, if not more. This is causing you to delay innovation, falling further behind, in a downward spiral. I Honestly hope this is not starting to sound too familiar!

Why does this happen? What tends to happen is that the customizations that are layered on top of the system turn into spaghetti code, which is hard to maintain. Upgrades of the platform are no longer compatible with your code, release cycles are getting slower, time to market of new features is slowing down and performance worsens. Good news: your eCommerce channel is growing! Bad news: you are outgrowing your legacy setup.

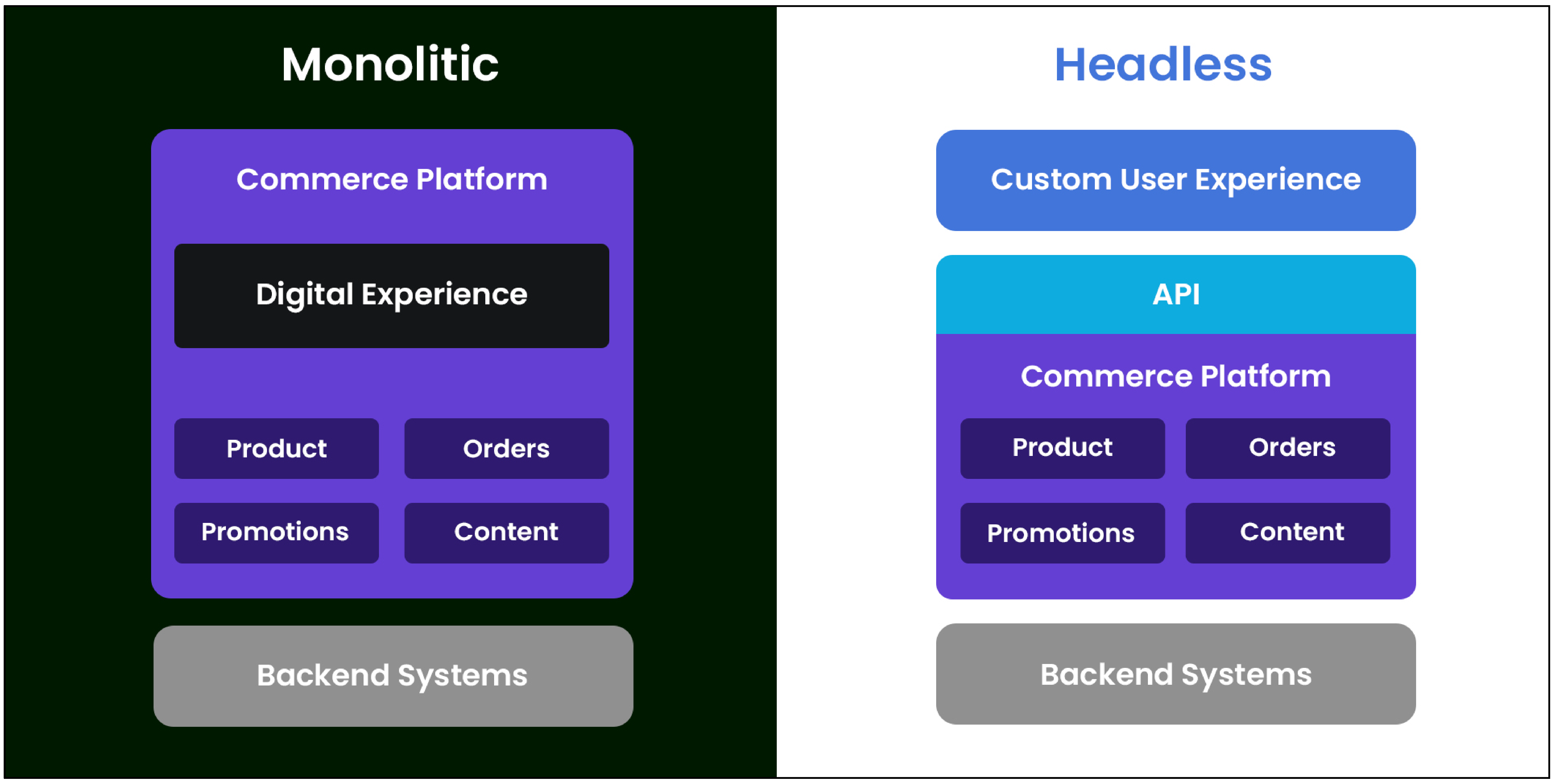
This is where it all comes down to scaling. How do you offer the same performance to user 1 million, as you did to user 1000? How do you keep relevant content coming to attract customers to your offers? And how do you reliably offer the experiences that make your business unique, to all your customers equally? If you are operating your business at scale, you need the best performance and great flexibility to react to customer demand. Luckily, the move towards headless is also what enables a business to scale even further towards composability.

Over the next chapters, we'll dive into the following checkpoints on the road: going Headless and going Composable. These stops will offer solutions to expand outwards and upwards, bringing higher scalability, performance and speed to market. But be warned, they come with their own pitfalls and disadvantages.



Second stop - Going headless. (is not as scary as it sounds)

The starting point, and a bit of terminology
Are publications of content to your site taking multiple days? Is a design change, that was supposed to be as simple as changing a button colour on your site, taking multiple sprints to finish? Are you waiting forever for your price change to appear in the search results? Then you're most likely suffering from a poorly decoupled frontend. "A what?", you might ask, "Is that contagious?" Not really, but let's first get our terminologies straight: What is meant by frontend, backend, and the most mysterious of words; "decoupling"?

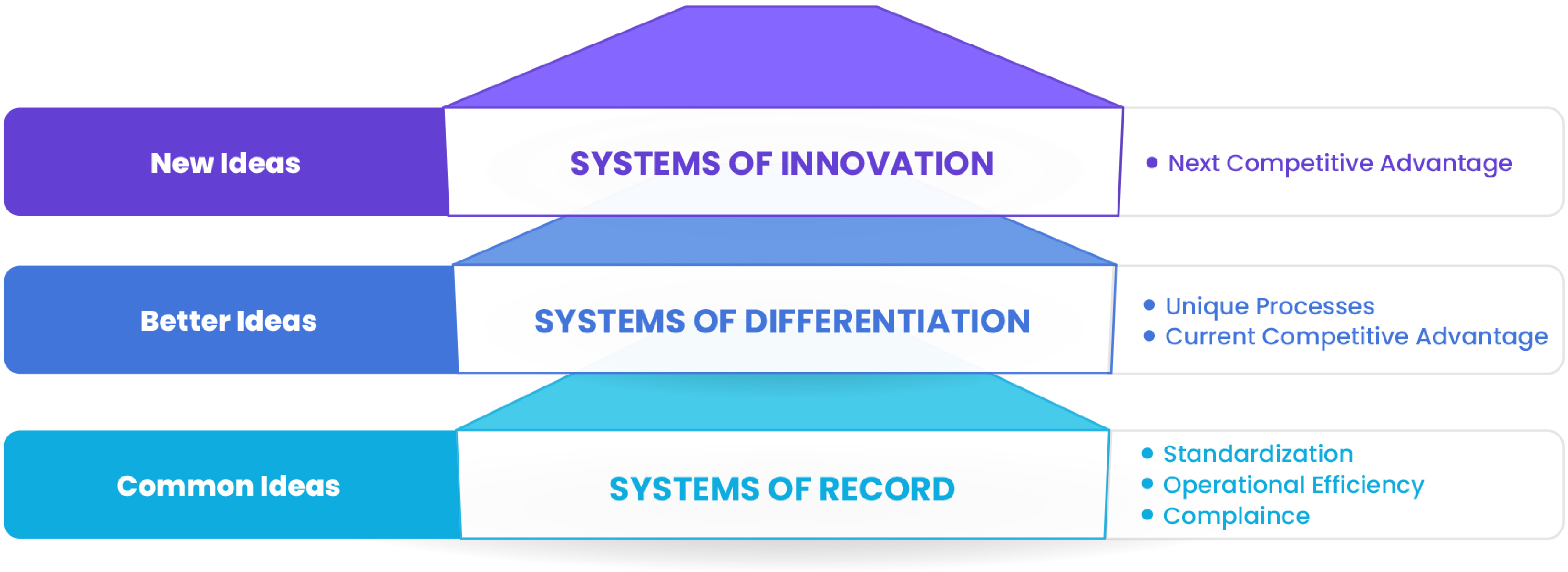


The frontend takes care of the custom user experience; the ever-changing brand messages, colour palettes and user flows. This part is what changes the most often in your eCommerce platform, so you can stay relevant to your customers and can offer them unique content and experiences. Here you want to try out new ideas quickly, test them for success in front of customers and discover how you can serve them best. And here you are the most hindered by connections to the backend.

The backend of your eCommerce engine takes care of the product catalogue, price- and promotion management, order management, etc. Think of it as the databases in which information is stored, and from where it is retrieved and changed when you need it.

Where the problem arises is when the frontend and backend are woven together too closely, or in software terms: tightly coupled. Think about a bowl of your favourite spaghetti, which used to be a set of separate ingredients, but are now one thing: a bowl of spaghetti. Try isolating an ingredient now. Pretty hard, isn't it? So decoupling means separating the frontend from the backend in a meaningful way. But why would you like to do this?

A good way to look at your technology is through the lens of the pace-layered architecture. In this picture below, you can see the level in which systems need change. Your backend (database, product model, pricing model, etc) does not need to change so often, but your frontend experiences need to. They reside in different layers of the architecture. The frontend is therefore a system of innovation, while your backend a system of differentiation. Below a third layer exists still, your systems of record. Think about your ERP system, your PIM system, where you store your data for master data management.



In other words, the pace of change for distinct layers is different. Therefore, you want them to be independent of each other, so you have as little dependencies as possible for the changes you want to make. You don't want a stable system with a low change rate to slow slow down your high change rate systems, or your high change rate systems to introduce instability in your stable systems! The aim of separation (decoupling) is therefore speed. When the number of complexities and dependencies go down, time to market of new features goes up. That time can then be used to provide value quicker to your customers, bring new features live quicker than your competitors, or even fix defects quicker, before they start preventing customers from clicking on that "order now" button...

It's reassuring how common this problem is, scaling out of a monolith. Most successful retailers did well before eCommerce was a channel to think about. As we read in the last chapter, by the time they started on the eCommerce journey, they acquired a monolithic eCommerce engine, since it solved their problems with one system, relatively easily and quickly. Or they built a custom engine themselves but were overtaken by their own success. Either way, the pace in which changes were needed to the customer experience was becoming too high, and the monolith was not able to scale with the demand. This happened with Adidas and Nike, who grew out of their Demandware (Salesforce) platform. It even happened to Amazon, despite having built their own eCommerce engine Cadabra all in-house. So, what was their first step of scaling out of their monolith? They liberated the system that is in the most pain; the frontend.

But you might wonder, how does your eCommerce system become headless? Surely, without some frontend your customers cannot experience your shop anymore? (As in "how did the headless chicken cross the road? In a KFC bucket...").

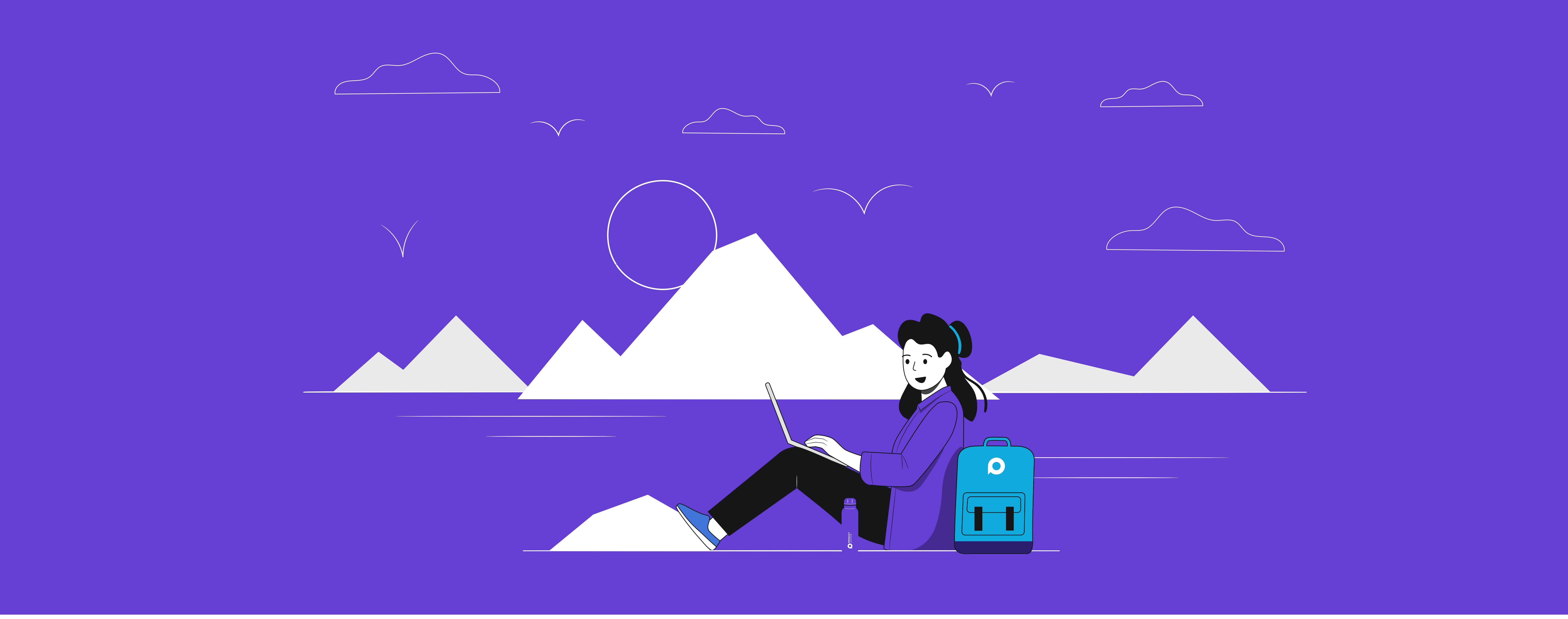
Indeed, the way to do this is by separating the frontend visuals from the backend logic. The frontend should just be worried about display, not about functional rules or data. It should get its data from the backend, by requesting it, by API. The separated frontend is generally custom built, using technologies such as React, Next.JS or OutSystems (our preferred choice at Product League). And this frontend is only allowed to request data, not store something itself!

The value

Is your monolithic eCommerce engine bursting at the seams? And is the word "technical debt" heard in most conversations about your roadmap? Time to consider scaling where you experience the most pain: the user experience.

Going headless is not as difficult as it sounds and has many advantages. By allowing the frontend portion of your platform (the part that is most subject to change) to be independent, you start to unlock the potential of your teams: you will see faster release cycles, time to market of new features will improve, and page speed will be up. All of this will directly benefit your customers and your revenue, allowing for a quick return on investments made. What this will not do yet, is to take care of the complicated backend, with its layers of business rules, databases and (product-) content.

For operating your business at true scale, with the best performance and with great flexibility to react to customer demand, going headless will still not be enough. But it is the right first step. During your move to headless, or even before, you will experiment with the first microservices. A dedicated onsite search engine for example, or a modern content management system that is headless by design. Truly, now you are heading towards **composable**, a landscape that is no longer limited by the question “what still fits?” within the monolithic platforms, but instead is being determined by the question “what is the right fit for me?”.

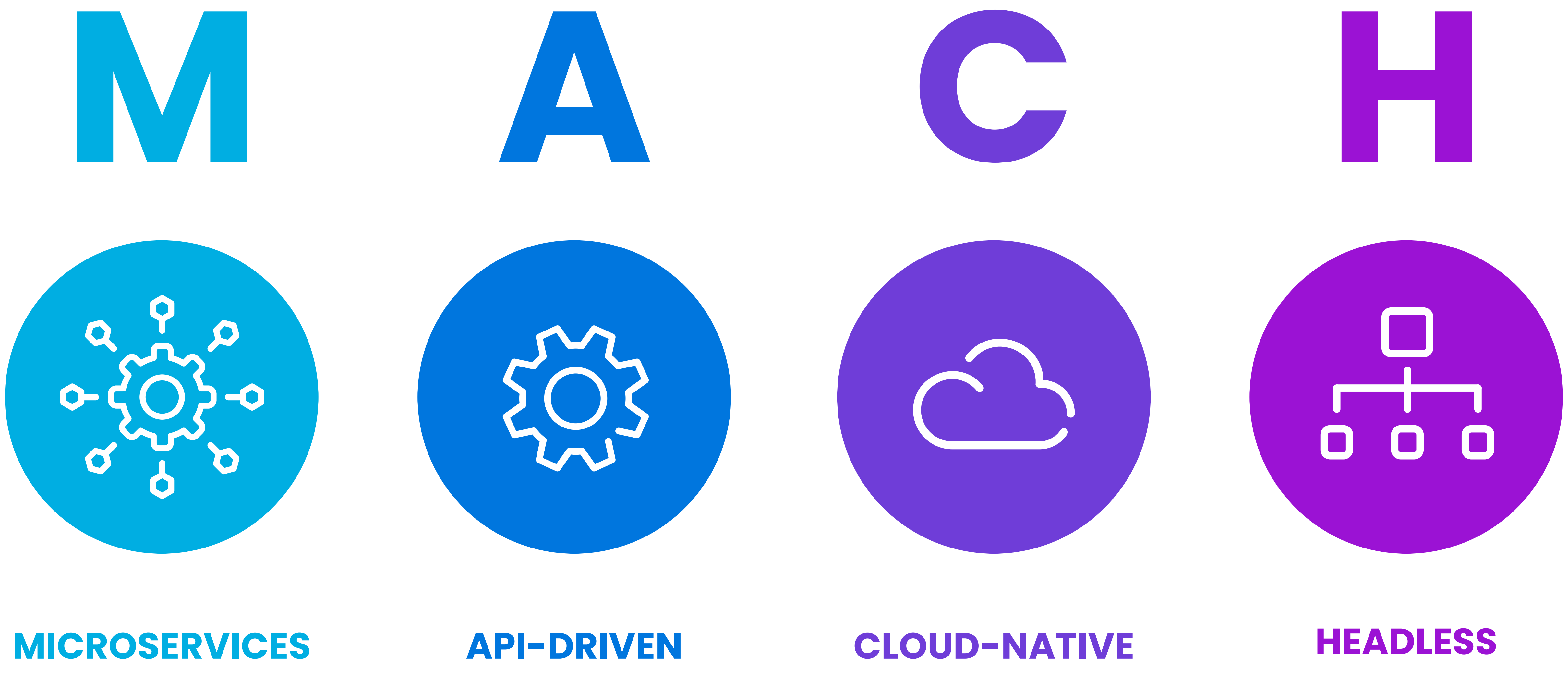


Third stop - Composable freedom.

Within a composable landscape, your custom user experience remains decoupled from the backend (headless). However, instead of the backend being one large, still rather monolithic ecommerce engine, you start thinking in microservices. Which system / process / service is the right fit for my business' needs? Investing in a dedicated pricing and promotion engine starts to make sense if your business relies heavily on pricing and promotion logic to offer the best deals to customers at any moment. An app that leverages the backend customer data in exciting ways to show them benefits, can be the way to set your business apart if you have a custom, multi-tiered membership program.
The revolution starts when you no longer think of your technology as one large backend system, but instead a versatile landscape of headless microservices, designed specifically for the purpose they need to fulfil.

In recent years, a movement was started by some of the vendors of these types of applications, called the MACH Alliance. MACH stands for:

- **M** icroservices
- **A** PI-driven
- **C** loud native
- **H** eadless



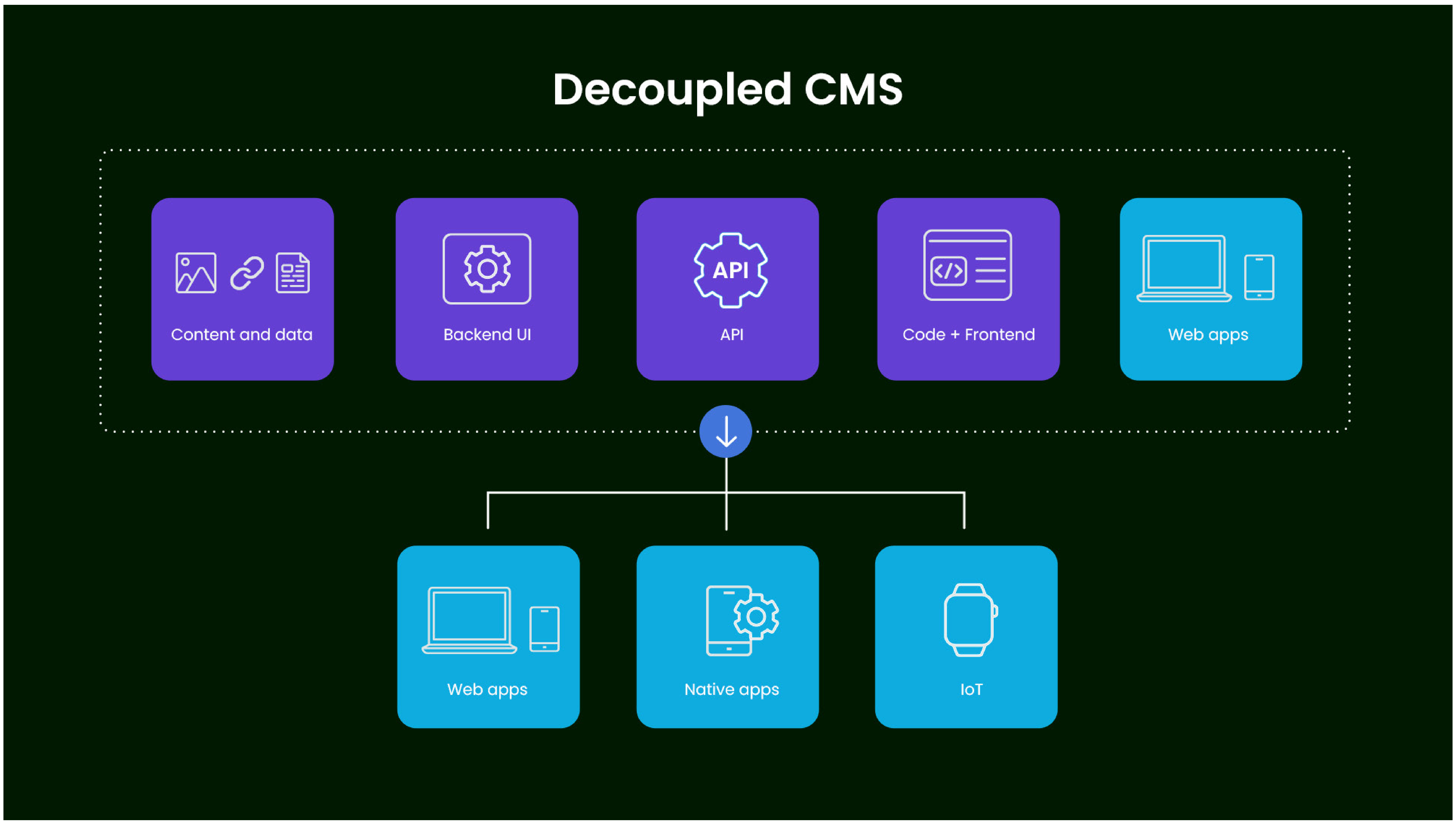
These companies realized that their products were very good at solving a particular subset of problems. They did not want to branch out and dilute their solutions until the products would become one-size-fits-all platforms as well. Instead, they doubled down on becoming the best of breed in their niche. They chose to partner themselves with other companies solving other relevant problems and dedicate themselves to the principles of the MACH architecture, to ensure the best mutual integrations. This way, any (eCommerce) problem could be solved, simply by connecting the right puzzle pieces. MACH landscapes are no longer comprised of large monolithic platforms, but instead of multiple dedicated services by SaaS vendors working seamlessly together.

An example of MACH – decoupling a CMS

A classic example of the MACH mindset shift is a dedicated content management system (CMS). An older CMS would be responsible for both the creation of content, and the display of this content on the site. Often this type of CMS is still part of a monolithic eCommerce application. It will push the content out and generate the site according to the rules that are set up. This often creates conflicts between the HTML inside the CMS and the rest of the site. Furthermore, it is hard to scale, because for each component you must describe exactly how it should behave, whilst taking the rest of the monolithic platform into account.

In contrast, a headless CMS does not concern itself with the display of content, only with the content itself. For any page sections, the decoupled frontend requests to the CMS: “do you have any content for me?” And the CMS replies with a response of available data. The frontend then takes care of how this is visually displayed. Not only is this “pull” model much faster than the old “push” model, but also much more interesting things can be done with the content now. Do you want to display a banner on the homepage in a certain way, and on your search results in a different way? No problem, you can use the exact same content but style it differently. Better yet, if later you want to use that content for an in-store display as well, you are free to do so. Not to mention all the personalization use cases that could serve your customers relevant content tailored to them.

This is the MACH way of working: you only speak to these systems through API’s. Imagine the relief to your teams: the content editors can now create their content without having to worry about embedding HTML, they know the rules for the end-product, and they can “fire and forget”. And the frontend development teams are no longer worried, since their carefully crafted pages can no longer be destroyed by a careless or incorrect content update. They remain firmly in control of display.



A new catch-all solution?
Moving from a monolithic and/or headless eCommerce engine into a composable eCommerce landscape certainly brings many benefits. Automatic scaling, so you are never caught unaware on the Black Friday peak anymore. High performance, since every tool is only doing what it is supposed to do. And dramatically improved time to market: Gartner predicted that by 2023, organizations that have adopted a composable approach will outpace the competition by 80% in the speed of new feature implementation.
Instead of one big decision every 5 years (do we continue/upgrade our monolith, or do we do something different?), you are now making many small decisions (is this tool still the best fit for this particular problem?). This means total cost of ownership (TCO) is not in the way of innovation anymore. It is much easier and cheaper to swap out a single microservice than it is to change your entire eCommerce landscape!

However, there are certainly challenges on the road to composable. For example, your vendor landscape becomes more complex. Dealing with all these contracts and vendors can be quite a hassle. Also important: is your organization capable of orchestrating such a digital transformation? It requires strict governance of architecture principles, much higher demand on teams to break up siloes and a brand-new organizational mindset. And let's not forget that the technical skills of building and running this new landscape are nothing to sneeze at, with its new data demands, API orchestration and custom frontend flows.

There are two directions to approach this complexity:

1. **You try to buy your way out of this problem** by getting as many SaaS solutions as you can, with implementation partners for all of them.
   - The problem is that this still requires much coordination between all these different partners. And the SaaS solutions out there, especially on the frontend side, offer many features out of the box, but are limited in their customizations and have code that is hard to re-use.

2. **You try to build your way out of this problem** by getting many teams of engineers that custom build solutions using high code.
   - While this certainly is the most flexible and the most performant for the resulting applications, it requires an organization to become an IT company, which is a very far-off goal for many retailers. Also, this route might become the most expensive, if you try to rebuild all the solutions that other companies have already figured out for you.

Build better experiences with composable commerce + low code

In practice, most organization will choose a combination of the two tactics, trying to buy solutions to “solved” problems (think commercetools for the eCommerce engine, or Contentful as CMS), while investing in custom solutions where the most benefit can be gained. The custom development will either be done in-house, or through partnerships with development firms.

We can also see that most growth in the eCommerce solution market sits between these Buy or Build tactics.

On the Buy side, the SaaS solutions are becoming more flexible and more tailored towards composable. This is true for many modern, headless SaaS solutions, but also for the old monoliths, which are moving towards the cloud in an attempt to catch up with younger competitors (although only time will tell if these are viable solutions or just “cloud washing”). One serious downside to these SaaS solutions remains their lack of flexibility. What if you need a custom frontend for a specific business use case? Or an orchestration service for data between a few systems that are not often used, but are crucial to your digital experiences? These use cases are common in an eCommerce landscape, but without investing in custom development, they might die a slow death at the bottom of your backlog.

On the Build side, the platforms that are doing low code are on the rise. Low code aims at making custom development more accessible by lowering the cost. Either less developers are needed for the same result, or the same number of developers deliver more. Some eCommerce frontend SaaS's, like commercetools frontend are moving into this direction, allowing for citizen development.
Even Salesforce is offering a very limited low-code drag-and-drop eCommerce starter package.
There are also pure low-code platforms entering this space.

At Product League we choose for the high-performance low code platform OutSystems to develop our eCommerce solutions. We chose OutSystems because it embraces the concept of reusability. For example, allowing you to build once and publish it both as a reactive website and a phone and tablet app. You can build integration modules that allow you to speed up integration delivery, which is increasingly important as your landscape shifts to composable. What it also allows you to do is quickly build multiple frontends for specific backend users. Since eCommerce sits squarely in your business process, the many users that need access to eCommerce systems need specific functionalities, that are not easily served with one generic “backend portal”. With OutSystems we can build both the customer frontends and specific frontends on top of the same backend which cater to the eCommerce managing employees.

The downside of the build approach remains the lack of reusable solutions. You don’t want to rebuild the wheel, so to speak. Unfortunately, you spend the first months rebuilding what SaaS solutions already built before, only to then start your custom work afterwards. This is where the exciting concept of Accelerators comes into play. An accelerator aims at kickstarting your custom development with out-of-the-box components. This can leverage the work that has been done before, making a project start at 60% instead of at 0. At Product League we chose this approach since that is exactly the low code mindset as well - keeping reusability in mind when it comes to designing and building every component. When keeping reusability and accelerator-style development in mind, the benefit is that you think early on where to buy and where to build, incorporating “solved” solutions in your architecture while focusing your development efforts only on the truly differentiating use cases.

A journey, not a destination

Where is your company on the road towards composable? We've been through many peaks, valleys, and potholes in our whitepaper. Here is a list of the questions we have asked ourselves over the course of this series:

1. Is my eCommerce technology meeting the customer's demand, without scaling issues?
2. Is my technology stack holding me back?
3. Which business problems make my business unique?
4. What unique value am I providing customers?
5. Am I trying to solve new problems or am I encountering solved problems?

As you answer these questions yourself, we hope that you are leaving this article with a sense of direction and a sense of purpose, to adopt the principles of composable where they make sense and ignore them when they are not (yet) a right fit.

Crafting a New eCommerce Legacy:
Our 'Road to Composable' whitepaper may have reached its end, but your adventure towards eCommerce excellence may just be beginning. It's time to turn insights into action and theories into tangible results and partnering with Product League will help you navigate the exciting realm of composable eCommerce. Don't just adapt to change – be the frontrunner of change. Reach out now, and together, let's craft a new dawn in eCommerce by building systems that are as unique and dynamic as your business vision!

Product League helps you assess your now, your future and the road there, using the power of our multi-disciplinary teams, state of the art technology and our extensive knowledge of the world of eCommerce. When you have established the vision, we can accomplish it together. We are an Outsystems powerhouse, enhanced through partnerships with Commercetools and Contentful. Together with your business we can achieve anything: from made to measure solutions to full SaaS implementations. Helping you to reach your customers and your goals.

Reach out to us ›

Dirk Kerpel is an eCommerce leader with years of experience in transforming digital businesses. Through his deep product knowledge acquired at large retailers like Staples Europe and Adidas, he is now helping to reinvent composable commerce with Low code, as Head of eCommerce at Product League.

- Head of eCommerce

PRODUCT
LEAGUE