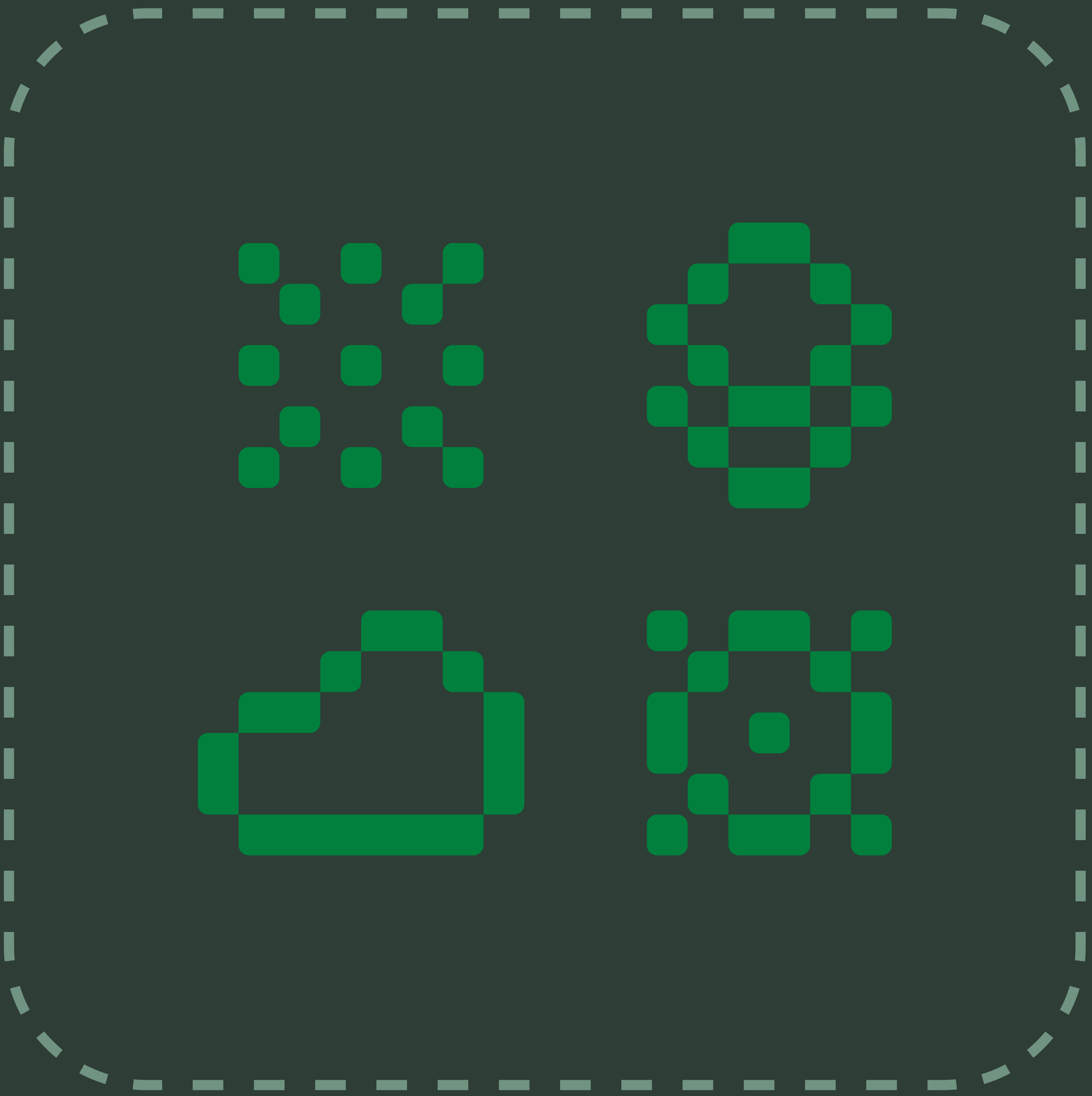
alokai

learn more about
MACH approach at alokai.com



MACH approach to ecommerce

Table of contents

Ø1.

Chapter Ø1.

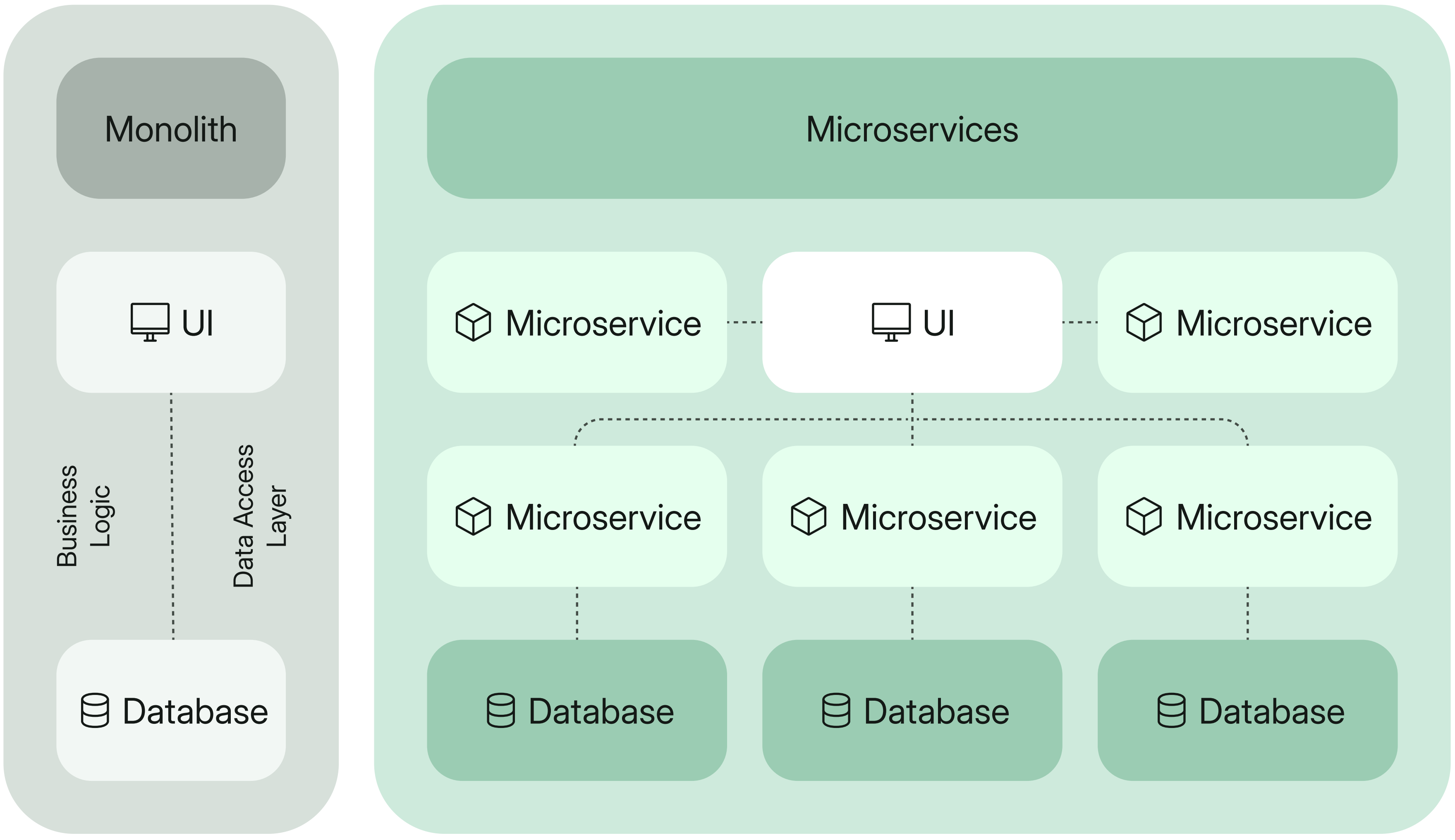
Microservices for modern commerce: Defining the “M” in the MACH architecture for enterprise commerce

Unlike traditional, rigid monolithic platforms that come with lots of restrictions, microservices are independently developed, deployed, and managed by a small team of people from different disciplines – providing the flexibility to build new prototypes and deliver new features to the market quickly.

What are microservices?

A core component of the MACH architecture, microservices are small, standalone applications, designed and deployed individually. With each microservice you can add a new functionality like a shopping cart, payment, or search functions independently.

How it looks like in ecommerce



That brings both challenges and advantages.

The advantages of microservices for modern commerce

Today, speed is a leading competitive differentiator. Monolithic ecommerce applications, which still drive many sites of big retailers worldwide, have become bottlenecks for innovation. And while the monolith can be deployed at once and tested and monitored more easily, this comes with complexity and inflexibility.

How it looks like in ecommerce

Reduced software complexity	The scope of a single microservice's functionality is limited. That makes maintaining and updating it so much easier. You only have to care about messages from other microservices that you subscribe to (inputs) and your API (outputs).
Specialization	With microservices, you can choose the right tool for the right task. Each microservice can use its own language, framework, or ancillary services best suited and preferred by the team using it.
Decentralized responsibility	Dedicated teams take full responsibility for "their" microservice. This leads to smaller codebases, which help developers focus and connect more closely with end-users.
Faster time to market	Building a microservice requires a cross-functional development team that works independently on their project. This allows for significantly faster deployment and makes it easier for different teams to collaborate on the project
Increased resilience	A business application made of an array of microservices has no single point of failure. If one service no longer responds, this does not automatically break the entire application.
Better scalability	Microservices are small and work independently. This means it's easier to scale them vertically and increase the overall performance. Instead of having to scale the entire application, you can scale up a single function or service.

Challenges with microservice-based architectures

Instead of a classic, horizontal structure, microservices require a cross-functional structure with vertical teams working independently. It requires the right infrastructure and tools for orchestration and monitoring.

As such, some of the drawbacks of the microservices approach include:

Decentralized data	Each microservice has its own data store. Multiple databases and transaction management require extra attention.
Testing	When testing a microservices-based application, you'll need to confirm and test each dependent service. This makes integration testing and end-to-end testing more difficult.
Deployment	Deployment requires more attention, especially at the initial setup. You'll need to think about how services are rolled out and in what order. Additionally, you might want to invest in automation to save time during the deployment process.
Monitoring	To pinpoint bugs, a centralized overview of the whole system is critical. Remote bugging, common with monoliths, is not an option across dozens or hundreds of services.

Experimentation that drives innovation

So why then, would you go with a commerce solution that uses microservices instead of sticking with the good-old monolith approach?

While more planning and consideration are certainly required at the beginning, particularly if migrating over from an old legacy ecommerce solution, the initial effort pays off by making commerce for your business more flexible, maintainable, and simpler to run in the long term.

Microservices offer the perfect environment for responsive online commerce. Thanks to smaller, dedicated sets of functionality, testing a new service, offer, or special promotion can be done quickly and more frequently.

This is commerce-as-a-service in the truest sense.

Microservices provide businesses and their tech teams with the foundation to do exactly what they love: **create something new**. This leads to better customer experiences and, as a result, retailers benefit by being able to stay ahead of client demand – and the competition.

Microservices empower teams to take ownership

As a design principle, this approach favors basic, time-tested asynchronous communication mechanisms (that take the form of APIs) over complex integration platforms. APIs allow the dedicated teams to focus on microservices.

While large monolithic applications often need around 100 or more backend developers, microservices require small teams of between 2-15 people to develop, deploy, and manage a single microservice through its entire life cycle.

Product owners, marketers, and process managers manage their own pieces, simplifying business operations and taking some load off IT teams.

After the initial development and implementation, all IT has to do is to keep things running smoothly.

Each team should have a wide mix of skills including business, operations, and developers. As a result, you get:

- **Fewer communication pathways,** meaning a much lower chance for error or confusion;
- **More autonomy** when it comes to choosing the technology best suited for solving each business problem;
- **Simplified decision making** when it comes to architecture and implementation.

The giants already moved to microservices

NETFLIX

Netflix was one of the first businesses to recognize the restraints of a monolithic architecture for a complex application. Today, Netflix uses microservices to manage user authentication, recommendation algorithms, content delivery, and other aspects. This setup allows teams to develop, test, and deploy changes without being affected by unrelated features.

amazon

Amazon used to rely on a two-tier all-in-one-platform approach to architecture. Yet, as the organization started to grow, it faced a pressing problem with the system's scalability. Amazon uses microservices to split its ecommerce platform into independent services—such as inventory management, order processing, user authentication, and recommendation engines.

Coca-Cola

Coca-Cola Company, with over 3,000 products worldwide and subsidiaries in every country of the world, faces the significant challenge of connecting entities on different continents. Coca-Cola's Global IT group decided to leverage microservices and APIs to support its international growth and gradually replace their legacy software.

Etsy

Etsy decided to switch to microservices after struggling with serious performance problems. Their IT team needed to decrease server processing time while simultaneously supporting the development of new features required to increase the platform's extensibility. To overcome the challenge, Etsy decided to redesign its platform and set up an API as a key component accessible to developers.

Spotify

Spotify was looking for a solution that would scale to millions of users, support multiple platforms, and handle complex business rules. Decoupling the IT system allowed the company to build flexible structures that are easily scalable, resolve real-world bottlenecks in a short time, test different solutions safely, and become less susceptible to large failures.

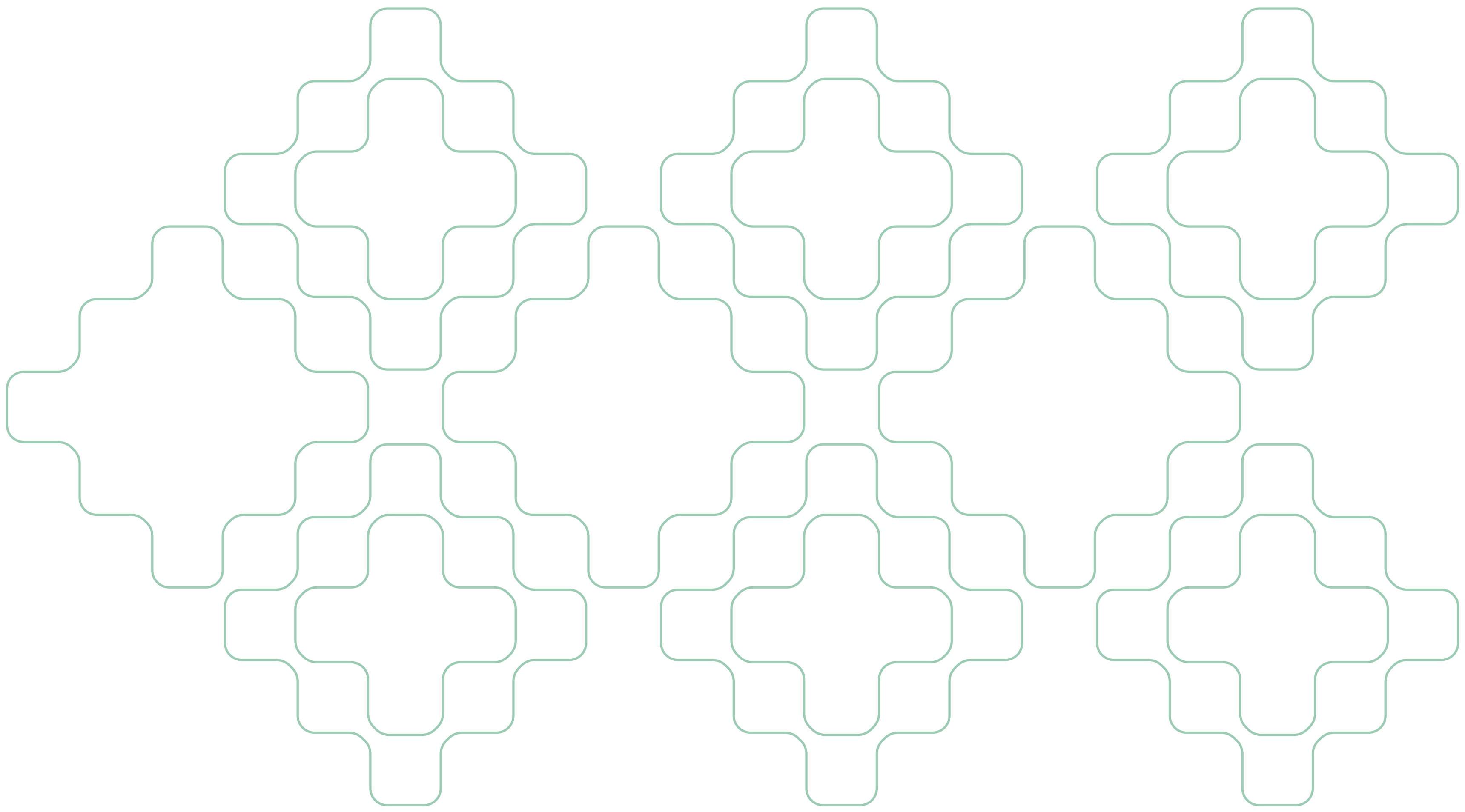
In fact, it’s the same story told many times over in different major companies

Microservices aren’t limited to the big, globally-operating brands. They are very democratic, simply because they don’t force a bloody revolution in the system, by killing off the whole class of existing functionalities and restarting the architecture from the very beginning.

On the contrary, **microservices are very agile and enable the addition of new business functionalities to existing monoliths**, and integrate with standard UI with almost no technical support.

In practice, it means that several different platforms (e.g., CRM, CMS, OMS, loyalty programs) can be ‘covered’ under one umbrella (i.e., frontend) and managed by a person without deep technical knowledge.

That’s the beauty of MACH that puts microservices at the forefront of the approach.



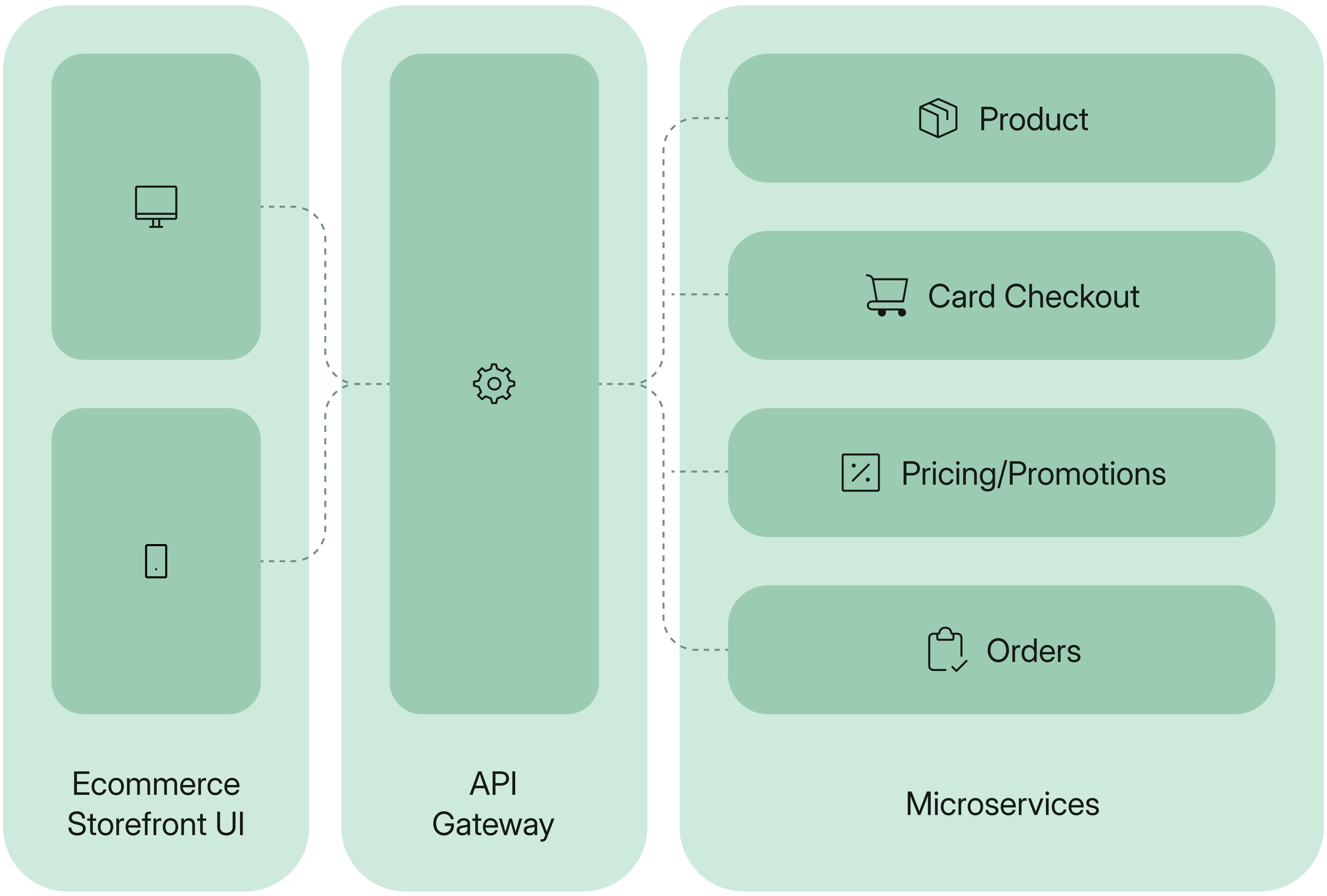
Ø2.

Chapter Ø2.

Building API bridges for more agile, more daring, more efficient ecommerce

Understanding the role of APIs in ecommerce

API (e.g., RESTful API or Storefront API) is a bridge for data transferred between the backend and the frontend, which provides a presentation layer in a consistent and standard format across multiple channels.



Most ecommerce APIs work behind the scenes; their power lies in the ability to streamline the complex interplay of an ecommerce site's operations. They enable developers to create a mosaic of interconnected applications that enhance the shopping experience without delving into the ecommerce platforms' intricate code.

Benefits of ecommerce APIs

Modularity and APIs enable connectivity, unlocking new flexible opportunities and ensuring the integrity of composable architecture.

Risk migrating

Automating processes such as order fulfillment and inventory management, increasing efficiency and reducing the risk of human error

Economic

Saving time and minimizing labor costs

Efficient

Enhancing the overall productivity of the business

Introducing the API-driven consumer approach

Customers are at the heart of any ecommerce innovation. API-driven headless commerce embraces omnichannel shopping that meets vast customer expectations.

Gathering data from platforms, devices, and channels into one location for further analysis is a dream for every business owner.

In the context of headless commerce, **API can be seen as an orchestrator responsible for proper communication between various components.** This is a linking element that enables a pure omnichannel experience in a headless ecosystem.

At first, connecting the data together seems a bit overwhelming, however, the outcomes will be surprising. A clearer and more specific view of customers will enable shaping and creating better shopping experiences. APIs keep all of the complexity of the data and deliver only the relevant insights of a customer, a product, or a channel.

How the API-first approach in building software enables ecommerce businesses to be more customer-centric

Ecommerce customer demands are constantly growing. From personalization and a frictionless buyer's journey to real-time customer support and plenty of payment methods to choose from – businesses must build for the future, whatever it brings.

That's why you need technology that can deliver superior digital experiences and a smooth path through checkout and beyond. Leveraging an API-centric tech stack can pave the way for these customer-focused experiences.

By ensuring your ecommerce system is open to integrations and supported by an evolving ecosystem of partners, you can build a tech stack that effectively meets your customers' expectations. APIs help you get there faster.

Equally importantly, an API-first approach is flexible and able to move with the changing times. Because individual pieces can be adjusted without making changes to the entire system, an API-first approach means it's easier to adapt to a retail landscape in flux and maintain a competitive edge.

Where API meets microservices

While a powerful and popular solution on its own, API truly shines when combined with microservices (hence the magic of MACH). Using APIs within a microservice-based architecture offers several special benefits:

- Enhanced scalability and functionality
- Facilitated modular development and independent deployment
- Reduced complexity by providing clear and standardized communication channels
- Enhanced maintainability and security of applications
- Easier compliance qualification
- Reduced operational costs

03.

Chapter 03.

Cloud tech as the backbone of modern digital systems

Cloud-native technology plays a pivotal role in the MACH architecture. It supports the full range of services and components that MACH relies on, allowing businesses to scale, integrate, and evolve as they choose.

The role of cloud-native technology in MACH architecture

Often reduced to the context of microservices, cloud-native technology plays a pivotal role in managing and scaling MACH solutions.

MACH embraces cloud-native infrastructure, so that the entire system leverages a solution that's scalable, resilient, and globally available. This includes not only microservices but also databases, file storage, networking, and security services provided by cloud platforms. These cloud services serve as the backbone for hosting and managing all MACH components.

API-first approach	The cloud-native environment facilitates API deployment and scaling, ensuring the best security, performance, and availability for your application.
Containerization & orchestration	Cloud-native tech includes containers (such as Docker) and orchestration platforms (like Kubernetes) to run and manage microservices, headless CMS platforms, backend services etc., making the system more modular.
Decentralized responsibility	Cloud-native solutions also automate testing, integration, deployment, and scaling of various components. In the MACH model, automated deployment pipelines (CI/CD) are crucial for ensuring that updates and new features across all components can be deployed quickly and safely in cloud environments.

Why cloud-native tech is replacing traditional on-premises server technology?

In a nutshell, cloud-native solutions come with more flexibility, cost efficiency, and innovation potential compared to server-based technologies. However, there's more to it than meets the eye:

	Cloud-native	On-premise
Scalability	Instantly scalable to meet demand. You can scale up or down based on traffic or workload, without worrying about physical hardware limitations.	Scaling is manual and often requires purchasing, installing, and configuring additional hardware, leading to slower response times or unexpected traffic spikes.
Cost efficiency	Operates on a pay-as-you-go model. You only pay for the resources you use, allowing for better cost management.	Requires significant upfront capital investments in hardware and ongoing maintenance costs. Even when not in use, you're still paying for underutilized infrastructure.
Resilience and reliability	Offers built-in redundancy, failover, and disaster recovery. Major cloud platforms have global infrastructure that guarantees high uptime and availability.	You must invest in expensive backup and disaster recovery solutions yourself. Any hardware failures can lead to significant downtime.

Ø4.

Chapter Ø4.

What is headless architecture, and why is it the future of ecommerce solutions?

Usually, a headless ecommerce app is a set of backend services – such as CMS, search, order, or stock management – and a single, lean frontend application that integrates all the services and features into one coherent user environment. It means that the UI layer can be managed by a different team, utilizing different skills and technology stacks than the backend technologies.

“Based on the idea of separating the UI (frontend) from the application logic (backend), headless architecture provides even the most mature businesses with the maneuverability and flexibility of startups. This relatively new concept in software development is becoming one of the most significant trends that will soon change the face of ecommerce.”

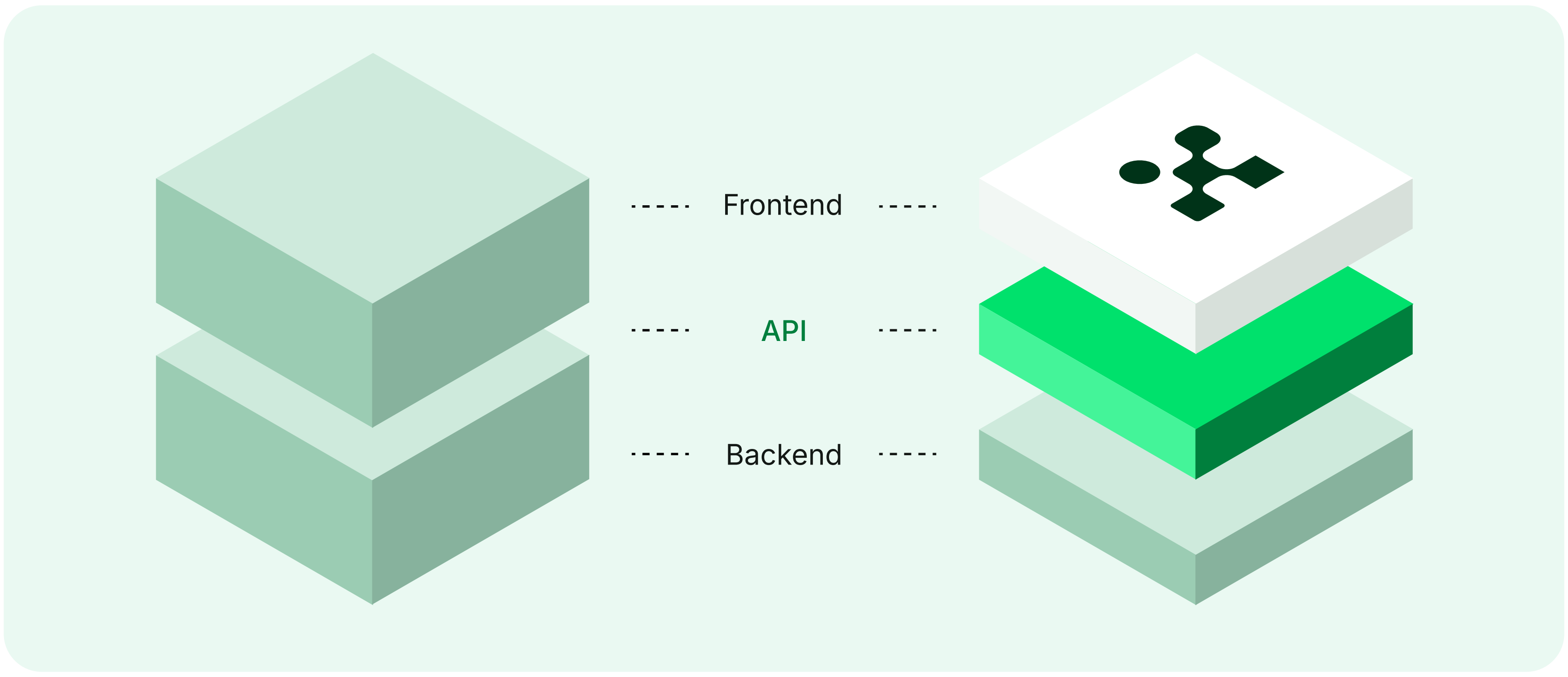
Patrick Friday
CEO at Alokai

Headless in the omnichannel ecommerce

The most popular all-in-one solutions are undoubtedly robust and offer a number of excellent features. However, their great power comes at a great cost: performance.

As one extra-large unit with the frontend tightly coupled to and dependent upon the backend, they can be very slow. All of the out-of-the-box features mean lots of code – and the more code you have, the longer it takes to process.

Headless architecture is based on a decoupled frontend integrated with content management tools via an API, so there is no need to render so much "default”; as a result, everything runs significantly faster.



A traditional, monolithic, architecture provides all the components needed for managing and publishing content on the web with a single codebase. 15 years ago, it was the only and the most convenient option for ecommerce businesses. The challenges have just started to appear.

Traditional, all-in-one solutions were designed to give users full control over the system, but control must go hand in hand with flexibility. Without the latter, systems are simply incapable of keeping up with the pace of business changes.

Why is headless CMS a must-have in the omnichannel world?

Traditional CMSs were built to handle content for a single platform, like the web. However, marketers now have to think beyond that, catering to mobile users and adapting to emerging channels.

Traditional CMSs can't keep up with this demand. Since the backend and the frontend presentation are tightly linked, it becomes difficult to work with new tools or distribute content across multiple platforms. This approach doesn't offer the flexibility needed for omnichannel content delivery.

A headless CMS, on the other hand, is more adaptable, allowing content to be sent in the right format to any platform. It acts as a central hub for integrations with tools like Marketo, Google Analytics, Salesforce, Shopify, and commercetools.

These integrations give editors insights into how users engage with content across different channels, allowing for more personalized and targeted experiences. While it might seem like a solution for tech experts, a headless CMS actually enables business users to improve the customer experience on their own across all digital channels.

As a result, this agile CMS isn't just for managing content – it becomes a hub for driving customer engagement across all touchpoints.

Headless CMSs are on the rise: Key benefits

A headless CMS is a unique and exciting way to manage and distribute content across multiple platforms. Unlike traditional CMS, which intertwines content and presentation layer (frontend + backend), a headless CMS separates the two, providing greater flexibility and freedom in presenting content.

The advantages of headless CMS, such as Contentful, Amplience, Contentstack, or Storyblok go beyond improved performance.

Flexible UI

A headless CMS allows developers to use any platform or technology to deliver content. It is not tied to a specific frontend delivery layer or technology stack, which gives developers the freedom to create and innovate without constraints.

Omnichannel reach

API-based connection to the backend also enables businesses to serve the content not only via one default channel but to spread it to a whole variety of electronic tools such as desktops, smartphones, wearables, and any "smart" devices.

Built for scalability

Scale as you go, not as your platform allows! Since the frontend and backend are separate, you can customize and upgrade your site without compromising on performance. You can easily scale your operations without worrying about the underlying infrastructure.

Design freedom

Product designers and frontend developers have much more freedom when it comes to UX/UI design. They can prepare various template options that can be integrated with CMS and then give the content team the ability to make changes without IT teams being involved.

Why is delivering premium CX easier with headless architecture?

→ **Implement one process at a time.** With headless, you can fix your biggest pain points right away. Let's take order management as an example. Inventory visibility and accuracy is usually a big one, as it directly affects how soon you get ROI. Once your order management setup is all set, you can build up additional rollouts like shipping from stores, setting up dropshipping vendors, etc. This granular approach also means far lower-stakes change management, more seamless employee adoption, and, ultimately, a lot less business risks than you get with a 'big bang' implementation.

→ **A headless platform enhances your existing customer touchpoints.** Whether it's your ecommerce site, mobile, text messages, email, social, clienteling apps, or call center, you can count on better processes, more accurate data, and greater scalability after embracing the headless architecture. As a result, you deliver better customer experiences. And as customer preferences change over time – think same-day delivery, social selling, sustainability – it's much easier to add new systems and processes to enhance your CX as your business grows.

→ **React quickly.** If there's one thing we've learned recently, it's that how quickly you can adapt to changes literally makes or breaks your business. How fast can you implement new processes? Transition locations between online and offline? Adjust inventory in response to market changes? While no one can predict the future, a headless architecture can help you prepare for it by providing much needed flexibility.

Challenges with all-in-one systems

Staying up to date with the latest market trends is difficult

It's challenging to deliver great experiences when all elements are tightly connected; moving to modern frameworks can be risky, as can changing the UI. Since the frontend and the backend are one unit, developers can't just alter the frontend layer to adjust it without interfering with the underlying database code.

Every change needs a lot of care

The updates made in traditional systems need to be thoroughly tested because one single mistake can cause the whole system to collapse.

One service provider limits the business possibilities

The all-in-one approach seems comfortable at first. There is no need to dig through the internet in search of additional solutions; everything is simple to develop, test, and implement – especially since the providers often deliver easy-to-use tools to handle all these processes. The problems start during growth, when default features turn out to be not the best ones available on the market, and the dependency on one IT provider reveals itself as a significant drawback.

How does a headless CMS enable a more sustainable development process?

If you're in the market for a CMS system, you're probably looking for the new cornerstone of building supreme digital experiences. The process usually leads to significant investments in development to meet business needs. And this need is accelerating as you recognize the growing demand for digitally-driven experiences – interactive shopping, subscriptions, mobile and wearable apps, chatbots, voice, kiosks, etc.

There are 2 real advantages to a headless architecture: development agility and risk mitigation. Development agility is one of the key concepts for a modern MACH architecture approach as merchants aim to deliver digital experiences faster, enable application scalability and resilience, and reduce technical debt.

The API-first approach of headless CMS systems means that developing and coordinating these changes is far easier and provides teams with the flexibility they need to launch impactful digital experiences across their customer journeys. Now, content that used to take a month to publish takes only minutes – or even seconds.

MACH headless systems usually make building and testing code far quicker and easier. Legacy systems usually rely on building a local copy of an entire environment for each developer or test instance. In contrast, headless CMS often creates a sandbox copy with a few clicks.

Risk mitigation is also important. The rapid advancement of technology, driven by new programming languages and channels, creates a constant need for systems that can quickly adapt. Without this flexibility, you risk missing out on delivering experiences through emerging platforms or suffering from poor marketing performance, like weak search results or ineffective social media engagement.

Ø5.

Chapter Ø5.

Frontend meets User Experience: Unparalleled drivers for conversion

Why does UX even matter?

UX is a part of the entire customer experience (CX). If the User Interface (UI) is all about how your store looks, UX is about how it feels once the customer is interacting with your site.

UX concerns many other factors like accessibility, ease of use, and load speed, to mention only the most important ones. All of them, combined, build the general feeling, i.e. customer experience.

However, UX is more than JUST an intuitive page structure, a beautiful interface, or lightning-fast performance.

It is all of them and more. Good UX requires combined analytical, technical, and creative competencies. The end game has a clear goal: encouraging the customer to act exactly like you've envisioned.

The frontend on the front line in the battle for customers' wallets

UX-related issues are not just about the visual layer; the frontend technology is vitally important for customer experience. And let's not be deceived, frontend development is still a huge area that is dynamically changing and can be a real headache for developers who are trying to embrace all the latest trends.

New frameworks and libraries are popping up so often that businesses of any scale are constantly tempted to try new things in the name of finding a competitive advantage.

This is, by all means, a good approach. After all, we all know that the saying "if you are standing still, you are going backward" is more true in the ecommerce industry than anywhere else.

Headless architecture, since its general idea is dividing the backend from the frontend, gives space for both better user experiences and experimentation.

Developers are free to try out new innovations without having to worry about platform stability. This is a way to assemble more future-proof systems that don't force reimplementing entire units with every revamp of UX design.

Why did the UX start to be crucial in driving conversion, and why is it easier to implement it correctly with the MACH approach?

The MACH approach helps solve a lot of challenges organizations face in trying to provide customers with the best possible user experience across every touchpoint.

Customer journey can begin on Instagram, pass through email promos, mobile browsing, and end with a purchase through the desktop website. If the brand wants to stay relevant, they recognize the value of being present across multiple channels.

But there are still businesses that struggle to provide a seamless experience across the board.

MACH enables ecommerce businesses to leverage best-of-breed frontend technologies such as modern JavaScript frameworks like React and Vue, alongside other JAMstack-related technologies to reduce page load speed and network latency over world-class CDNs.

Faster page load speeds positively impact organic traffic through higher positionings on Google SERPs, resulting in more site visitors and customers eventually. Better website performance enhances customer experience, ensuring the visitor won’t be discouraged from interacting with your brand again, even from a different touchpoint.

Go MACH or go home: MACH architecture to win global markets

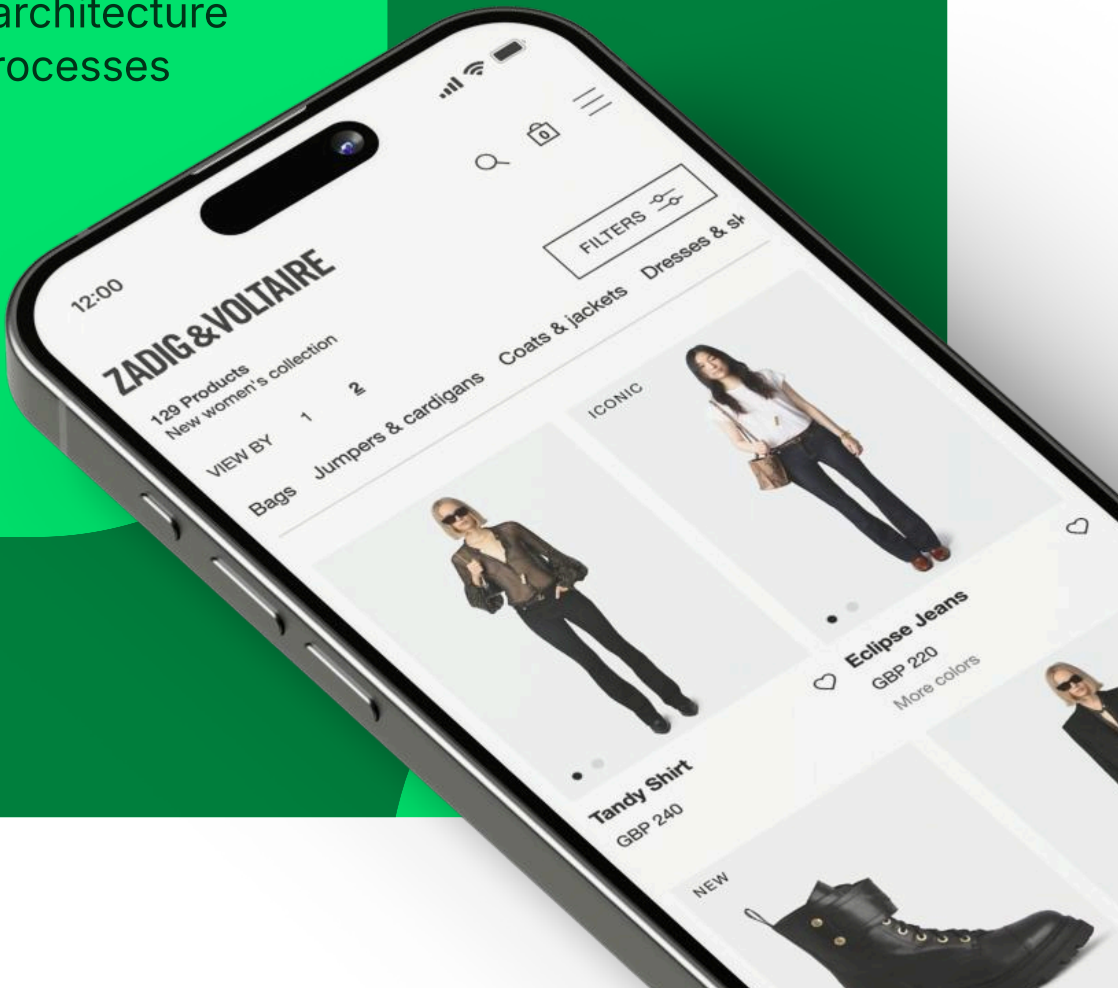
ZADIG&VOLTAIRE

Zadig & Voltaire, over its over 20-year history, gathered a die-hard fan base among everyday customers and fashion icons.

Zadig & Voltaire, over its over 20-year history, gathered a die-hard fan base among everyday customers and fashion icons.

Recently this Paris-located fashion house has truly blossomed. The company's digital transformation focused on smoothing the path customers need to take to make a purchase while also building a friendly and efficient work environment for non-technical teams.

As Zadig & Voltaire got very serious about digital the company faced the challenge of rebuilding its entire ecommerce architecture to replace some of the manual processes with automation.



Here are the excerpts from the interview between Piotr Karwatka (Alokai) and Jonathon Ribas (CTO at Zadig & Voltaire). Check the whole case study here!

Tell us about the journey and challenges of getting the first store up and running.

We are a global brand with a complex IT infrastructure. It took 7 months of hard work. We had to work on Alokai and Magento 2, as well as our in-house orchestrator which communicates between M2 and our IT applications. There are challenges along the way but two specific ones during our process were integrating our catalog from PIM to Magento 2 and Alokai and hosting a PWA.

What are the key aspects of rolling out the next stores? What stays the same, what changes?

The biggest challenge is that the business windows for launching new stores are very short. As retailers, we have a lot of key moments during the year where we can't risk disruption to our business.
Our decision was to do a release with some new features for the next countries to roll out, with a lot of testing and improvements between each release.

After launching in all our main countries, we've now got to the point where it is fast to implement a new PWA store with an existing language and the same shipping and payment methods. We can now achieve that in under a month, including all the testing. Our rollout process becomes more refined as we go along but always starts with the same questions about currency, language, shipping and payment methods, and integrations.

How can integrations differ between countries?

We have tried to choose the best partners to minimize new integrations in the future. This basically means choosing integrations with more global reach. For example, we have chosen a transporter that can ship everywhere in Europe and also China. And on the payment side, we choose Adyen because it has almost all the local payment methods we want to use in the countries we plan to cover.

The USA will be a new challenge in the near future because they have different transporters. We will need country-specific integrations to best cover the market.

Was there a stabilization and fine-tuning period after launching the first shop?

We had done a lot of testing internally on mobile and desktop before launch but the amount of real traffic and peaks around things like the release of newsletters is a different scenario and our site went down a few times due to bottlenecks in our custom integrations. Real life is not the same as the pre-release environment. The first calls to our support team after launch were very important in helping us to find some bugs we hadn't identified before.

We had precious help from Michael Bouvy from Click & Mortar and the Alokai team and decided to set up a Redis cache in front of NodeJS because it was consuming too much CPU and memory.

alokai

Leverage the best MACH technology has to offer without giving up your monolith setup

Are you ready to MACH-ify your store? Go with Alokai!

Build a performant scalable frontend with ease

Integrate best-of-breed third-party technologies

Move your operations to cloud and speed up your application

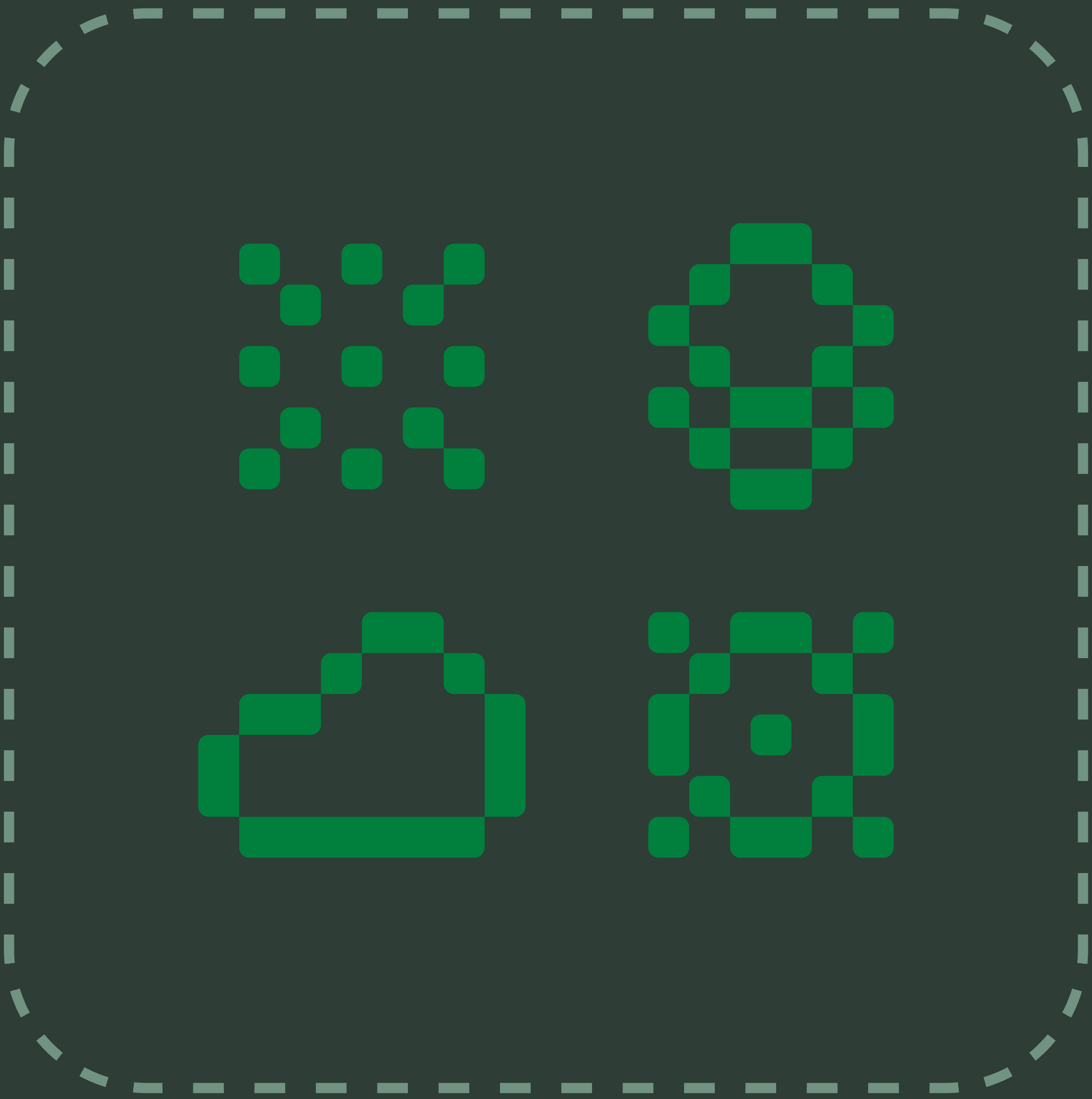
Set your frontend free with headless solutions

Get a personalized demo with our team and see if it's a good fit for you!

Book a demo →

alokai

learn more about
MACH approach at alokai.com

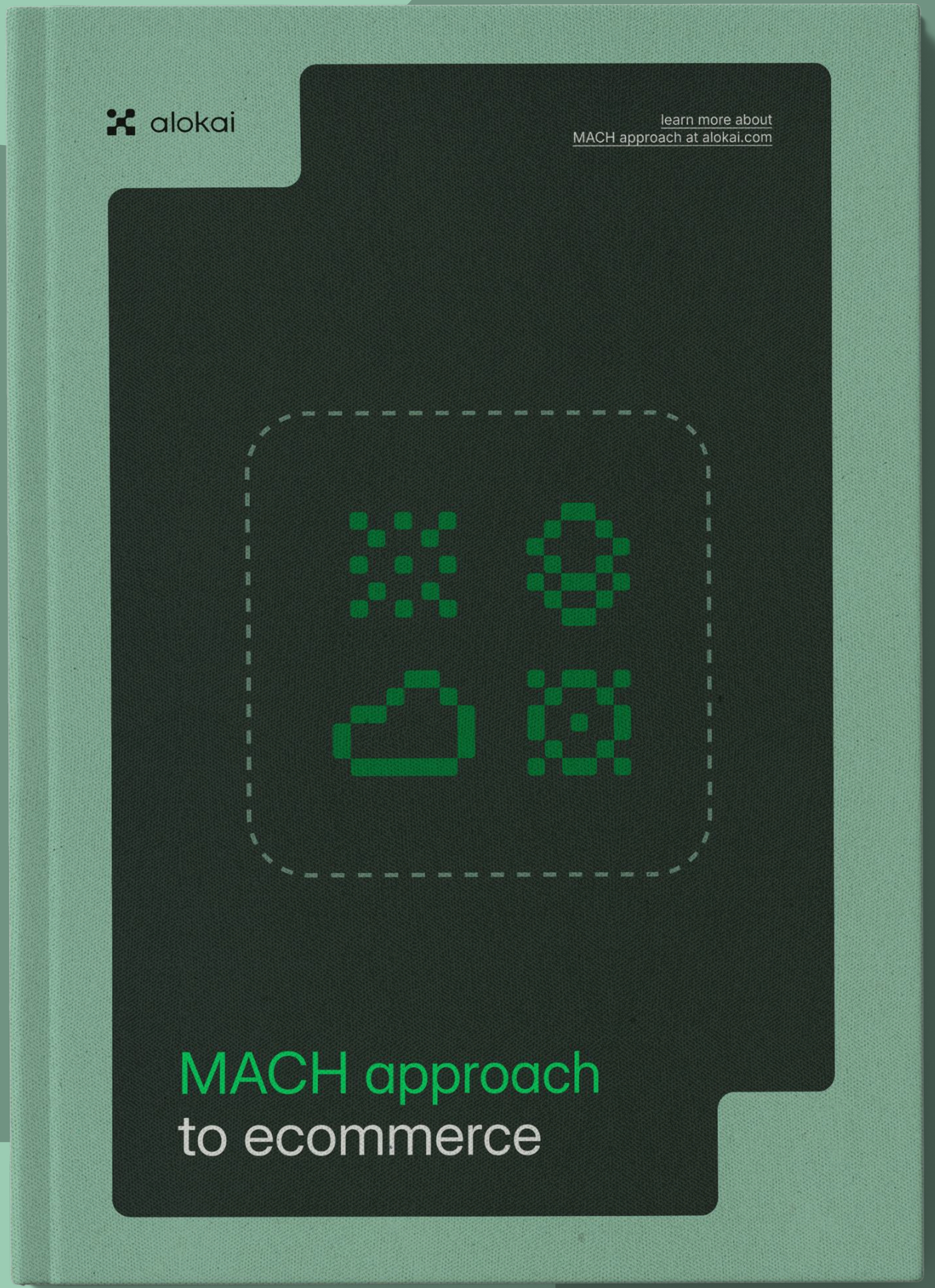


MACH approach to ecommerce

alokai

MACH approach to ecommerce: Benefits and use cases

Download now →



alokai

learn more about
MACH approach at alokai.com

MACH approach
to ecommerce