commerce layer

Jamstack ecommerce

A paradigm shift in the way of thinking and building ecommerce websites

Whitepaper v1.0.4

Last update — July 1st, 2021

Publisher — Commerce Layer
Authors — Simone Fiorucci, Filippo Conforti
Editor & Designer — Simone Fiorucci

https://commercelayer.io
hello@commercelayer.io
@commercelayer

All the best projects are always driven by creativity and built on strong foundations. The problem is that these two ingredients of the perfect recipe generally lead to conflicting priorities. The former feeds on innovation and flexibility. The latter aims at security and stability.

Most of the traditional ecommerce platforms failed in narrowing this gap. In some cases, they actually were the cause of it. Best case scenario, they widened it. Hence the need to design a new model.

Enter Jamstack Ecommerce, a paradigm shift in the way of building ecommerce websites. Better performance, higher security, easier scaling, and great developer experience are all top features achieved by design in any Jamstack project. When it comes to selling online, the pay off is even greater as these factors translate directly into higher conversions at a fraction of the cost.

This whitepaper explores the reasons why a new approach like this is essential to go further and create customer journeys that matter nowadays. First, it analyses the downsides of the old, common setup. Then, it introduces a different, modern perspective in building better websites. Last — but not least — it gives you a little taste of how Commerce Layer can help you join this revolution.

1 The old school approach

Regardless of their features and different target clients, all the most popular platforms present monolithic architectures that claim to provide you with all the functionality you need in one solution.

Building ecommerce projects with these platforms involves an approach that can be roughly outlined through the steps below. It is a waterfall process that begins with the choice of the platform itself, a very sensitive decision affecting every following step.

1 Select an ecommerce platform

Usually, the evaluation is carried out by taking into account three main factors: features, price, and technology. The selected solution has to meet all your business requirements, it can't exceed the allocated budget, and the more your team is familiar with its core language, the better.

2 Design the website

Going with one platform or another automatically impacts on your website design and UX. In order to simplify the integration, or even to make it feasible, the agency must adapt as much as possible to the architecture provided by the selected solution.

3 Integrate the design into the platform

Once the design is established, the system integrator creates a set of templates to make it operational on the chosen platform. How easy this is depends on how much the UI has been designed for the platform itself. In any case, developers must be specialized in the specific platform, as they likely need to tweak it quite often to fit the business requirements.

4 Setup the infrastructure

If the selected platform is on-premise, the system integrator needs to take care of everything, from databases to application servers, content delivery networks, and deploy procedures. The stack has to be configured, tested, and maintained. The more complex the project, the harder the whole thing.

5 Optimize performances

Customers have high expectations. Your ecommerce website must be reliable, fast, and secure. Therefore, once everything is set up, the system integrator needs to run continuous performance and security tests in order to optimize your site speed and ensure data protection.

Downsides

The old school approach outlined above often depends on custom-built technologies and lacks flexibility. This is the case of most traditional commerce platforms since they ship with many structural limitations that make them hard to customize.
Let's have a look into them.

Constraints to creativity

UX experts are confined into navigation patterns that some engineers have designed on their behalf. Creatives can't truly innovate, as they have to deal with a sea of sameness, where every website looks and feels the same.

Limited CMS capabilities

Almost all traditional platforms provide very basic content management features. As a consequence, content editors have to settle for what comes with the platform. Alternatively, you need to integrate third-party solutions, devoted to author better content. This leads to a couple of common issues, like the struggle to achieve a unified search experience between products and content or even the need to separate the marketing site from the online shop.

Bad developer experience

They say a happy developer is a better developer. Vice versa, unhappy developers gradually becomes far less productive. Front-end frameworks lock your tech team into programming languages they don't like. Sometimes they force them to work with tools they aren't familiar with. Everything turns into a developer happiness free-fall.

More dev skills required

Having the presentation layer and the server-side functions tightly coupled, developers need to master both front-end and back-end coding. Full-stack engineers are not easy to find and, in the best case, they generally cost more.

Complex deploys

The deployment of code and data requires well-architected procedures that need to be developed and maintained. Live previews, site versioning, and rollbacks are more difficult to accomplish.

More DevOps activities

Monolithic platforms need to be constantly supervised, in order to ensure their proper functioning. Configuring and maintaining extensive sets of delivery and monitoring tools are essential. And it's no easy work.

Lack of flexibility

Most of the existing platforms don't let you grow as you should.
A simple redesign often implies restarting the whole project from scratch. In general, each additional step towards the results you want to achieve, puts your entire infrastructure at risk of replatforming.

Expensive scalability

Real scalability generally comes at a high price. Supporting high-traffic volumes and absorbing spikes can be challenging. Launching new markets and sales channels can force big changes and a strong development effort.

Lower security

Monolithic architectures expose larger surface areas to all sorts of attacks. Think of SQL injection, cross-site scripting, session hijacking, and more. This makes your PCI scope higher, increasing the required effort to get the compliance.

Slower time to market

The whole process is way more complex and this inevitably means that it will take longer to go live. The same happens when launching new features, opening new markets, and creating sales channels.

Higher costs

Traditional ecommerce setups are more expensive to build and maintain. At the end of the day, the bulk of your budget disappears into the platform, and you can't invest enough to grow your brand how you want.

2 The Jamstack way

The acronym JAM refers to a modern web development architecture based on client-side JavaScript, reusable APIs, and pre-built Markup.

Instead of having your entire business hosted in one tool, sites can be built as modular stacks that aren't tied to specific technologies. No more platform-specific plugins or extensions. Just front-end code and APIs.

The Jamstack approach has major advantages over the traditional model. It's faster, cheaper, easier to build and maintain. It provides baked-in security and lets you engage customers with any path-to-purchase. All these benefits don't impact on developers only but apply to your entire organization as well.

Building ecommerce on the Jamstack drastically simplifies your project and makes each step more efficient. Everything starts with the site design, bringing the focus back to creatives and innovators.

1 Design the website

Focus on your customer experience and functionalities, without worrying about any predefined structure. Let creativity drive the whole process and start considering any UI/UX possible.

2 Create a static site

Select a static site generator. Base your evaluation on your favorite programming language and build performance requirements. Take the approved design and code it into pixel-perfect HTML and CSS. Enhance it with dynamic features via JS and APIs.

3 Add headless content management

Select a headless CMS to let content editors edit pages. Choose among dozens of available options. Enjoy the freedom of building your models out of any schema. Design your content to be multi-channel from the ground up and ready to be delivered anywhere. Connect to other APIs to further improve your site search, personalization, and customer experience.

4 Add headless ecommerce

Choose a headless ecommerce platform to add transactional features and make your content shoppable. Integrate prices, inventory, checkout, customer accounts, and everything you need to sell online.

5 Deploy to CDN

SSGs generate purely static HTML files that are perfect to be served over a CDN. That's why your website becomes fast, secure, and scalable by design.

Upsides

A headless approach like this enables digital transformation and removes the barriers to manage cross-channel user experiences. The upsides of this kind of process are many. Let's explore them.

Creativity unchained

Designers don't have to ask themselves how difficult it could be to implement their ideas. This way, they can focus on what they're good at and enjoy the thrill of creating stunning customer experiences again. Their creativity is finally free from any potential constraints imposed by predefined blueprints.

Best of breed content management

Using an API-first CMS is the most flexible way to manage your content and support your website design. That's because headless CMSs are schemaless. Your model can be structured in a device-independent and presentation-free format. Content can reach any sales channel and the widest audience possible.

Great developer experience

Front-end frameworks are increasing in popularity and capabilities. Decoupling the storefront from the back-end provides developers with the opportunity to convert any design into pixel-perfect HTML pages. They can choose the tools they like more, harnessing all the power of their preferred stack. The positive effects on their overall productivity are almost immediate.

Fewer dev skills required

Most of the server-side functions are achieved as a service, through fully-managed APIs. Front-end developers are all you need to build a functioning website equipped with the best ecommerce features.

Atomic deploys

Code and content can be versioned into your Git repository. Each commit becomes a self-contained snapshot of the website. This helps guarantee atomic deploys and site consistency. Once your build is uploaded, the CDN invalidates its cache almost instantly, so that your new site version goes live right away. In any case, you can rollback to the previous version with just a single click.

Fewer DevOps activities

Site builds are just git push. The architecture is serverless. There aren't databases to set up and take care of. No more time wasted to implement, maintain, debug, and monitor complex infrastructures.

Unparalleled flexibility

Headless solutions are completely decoupled. Then it's easy for front-end developers to make updates to the presentation layer while leaving content and server-side functions untouched. These days, the ability to rapidly respond to new trends and technologies is crucial. Thanks to a headless approach, you don't have to worry about re-architecting everything when adding new features or selling across new channels.

Cheaper scalability

A successful ecommerce can't lose efficiency when it needs to handle large volumes of sales. If your site is served over a CDN, there's no need to be concerned about that. Scalability is ensured by design and your BFCM order spikes are no longer an issue.

Higher security

Not having to deal with databases, servers, and plugins decreases the potential for code breaches. Since the APIs handle all the dynamic functions, front-end developers don't have to bother with security issues and can leverage third-party vendors expertise to manage the boring stuff. PCI scope is reduced, and so your costs for certifications.

Faster time to market

The overall complexity is considerably smaller. The whole process becomes more efficient and linear. Configurations are quicker, maintenance is easier. All of this leads to faster go-to-market timelines.

Lower costs

Building, maintaining, and deploying Jamstack ecommerce websites is simpler. You need fewer servers (if any). Hosting front-end code is drastically cheaper. With no back-end infrastructure to manage, you are free from maintenance problems. Given the same budget, less development and operational costs ultimately mean more money saved for your marketing and branding strategies.

3

How Commerce Layer helps you build Jamstack ecommerce websites

Commerce Layer is an API-first commerce platform that lets you easily add enterprise-grade ecommerce to any website, by using the headless CMS, static site generator, and tools you like more. It provides all the endpoints you need to integrate prices and inventory, manage payment and delivery options, build custom checkout flows, customer account areas, and even more.

To put it simply, Commerce Layer is the headless solution that can transform your Jamstack website into a Jamstack ecommerce.
In relation to the paradigm shift we are introducing with this whitepaper, it is the tool you need to achieve the fourth step of what we called “the Jamstack way”.

Being headless, API-first and CMS-agnostic by design, Commerce Layer fits perfectly in a Jamstack environment and gives you complete freedom to create any tailor-made ecommerce project with low effort.

Whether you’re starting from scratch or you want to add ecommerce to an existing website, it lets you go transactional without breaking a sweat. Everything you need is provided as a service, in a fast, reliable and highly customizable way.

Reach out to us

Nobody denies that the main and ultimate goal of any ecommerce setup is selling. Yet there is a wide range of steps to go through to get there. Finding the fastest, most effective and creative ways to make your website convert is the fun part of the game. But there is also the other side of the coin. A daily struggle to be compliant and up to date with all sets of rules and regulations. Hours and hours spent researching, evaluating and choosing the right tools. Challenging efforts to learn the real best practices for a good ecommerce culture.

In the past decades, we've been working hard to learn all those best practices, with all kinds of brands, from startups to global enterprises.

If you want to make ecommerce better, our experience, skills, and passion are at your full disposal. Just get in touch with us. We can't wait to learn more about your project and help you make it successful.

Ready to get started?

Start building for free or contact us today if you need more information.

Request a demo