

THE DEFINITIVE GUIDE TO

Composable Storefront and Headless Salesforce Commerce Cloud



Contents

5 Introduction – It’s Official: The Future of Commerce Is Headless

Learn why we created this whitepaper, what it’s about and how it will help you make the most of headless Salesforce (and decide if it’s right for you).

6 Definitions

Before we explore the ins and outs of headless, MACH, composable commerce and more, let’s define what these key terms actually mean.

10 Salesforce: Evolution, Not Revolution

Now we have the key definitions outlined, it’s time to map the evolution of SFCC solutions over time, leading up to Composable Storefront now.

14 Should My Business Move to Composable Storefront and Go Headless on Salesforce Commerce Cloud?

How do you know if SFCC is a good fit for your business right now? This chapter helps you answer that question once and for all.

16 Strategic Considerations – Mind the Gaps

From customer experience to employee experience, storefront operations and technology architecture, there are a number of pitfalls to navigate.

26 Why Go Headless On SFCC?

Frontend flexibility. Backend flexibility. Performance. Productivity. Learn how all these things and more can be improved with a headless approach.

30 The 64labs Composable Storefront Accelerator

What exactly is this solution, how does it work and why is it going to transform the way you deliver commerce experiences through SFCC?

40 The Business Case for Composable Storefront and Headless SFCC

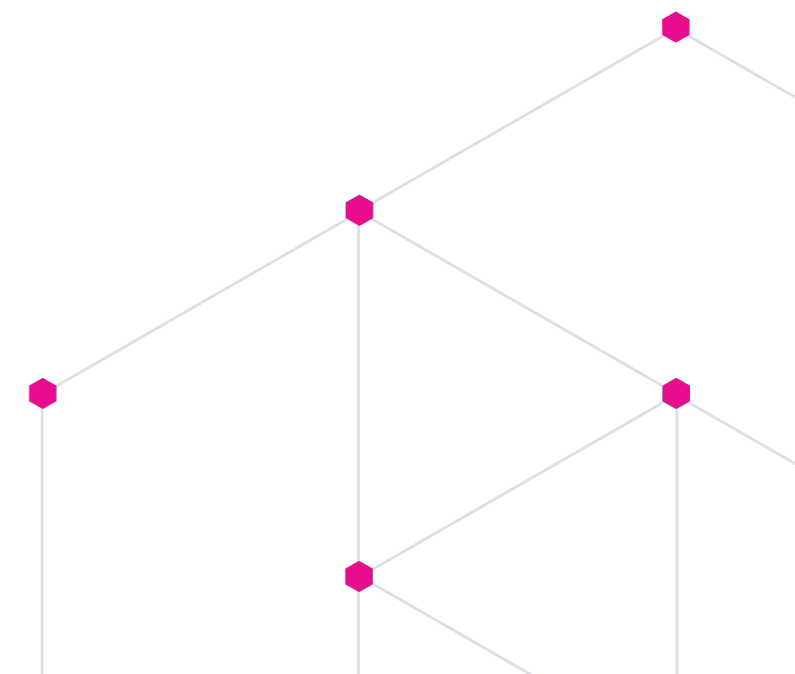
Sometimes the hardest part of technological change is getting stakeholders on board in the first place. This chapter gives you the ammo to make that happen

44 Defining an Approach to Headless Salesforce Implementation

You’ve decided to move to a headless approach on SFCC. Here’s how you go about defining the best approach for implementation.

47 Next Steps

Ready to make the move to headless Salesforce Commerce Cloud? Get in touch with one of our Salesforce specialists.





Introduction

It's Official: The Future of Commerce Is Headless

If you wanted a resounding indication that headless and composable are here to stay, look no further than Salesforce's Composable Storefront announcement in August this year. The new storefront, it says, "gives online retailers the ability to make site changes with the speed and flexibility needed to increase sales, conversions and basket size, while helping to drive down costs."

As competition grows ever fiercer in the commerce space and consumers look for increasingly personalized and cohesive experiences online as they move between different apps and sites and social media platforms, the above benefits will only get more important.

This whitepaper aims to answer all these questions and more, digging beneath the many buzzwords and acronyms to bring you a plain-English explanation of all the things you need to be aware of if you're thinking about making the move to Composable Storefront and headless Salesforce Commerce Cloud (SFCC). We cover some basic definitions and key strategic considerations. We explore why brands would want to go headless on SFCC (and how to know whether it's right for you). And we leave no stone unturned in helping you come to an informed decision about this way of doing things.

We also take a good look at the new 64labs Composable Storefront Accelerator and how it can help you unlock the power of headless, accelerate storefront performance and drive more sales.

Read on to find out more.

Definitions

Before we dive into the specifics of headless and PWAs with Salesforce, here are a few definitions that will give you the context you need to make sense of the choices in front of you.

What is Composable Commerce?

Composable commerce is an approach to building eCommerce solutions that focuses on bringing together the best components available for your needs. Composable commerce solutions use packaged business capabilities (PBCs) that aggregate key functions together for ease of deployment and integration. Phasing out the smaller bottlenecks, enabling you to truly realize the benefits in terms of agility and productivity.

For example, Amplience provides experience management as a PBC by combining content management (CMS), digital asset management (DAM) and experience management tools and APIs into a single platform. Other PBCs for composable commerce can include product and catalog management, search and searchandising, and commerce (including basket, checkout and promotions). PBCs make extensive use of APIs and are often headless (i.e. make no assumptions about the frontend experience). But they may not necessarily be developed using microservices, so they're not always MACH.

In a composable commerce solution, PBCs can be rapidly deployed and integrated together to support the desired customer and employee experience. You can learn more about [composable commerce in this article](#).

What is Headless?

Headless is an approach to building software systems that enforce a clean separation of the content, application logic and data from the frontend presentation layer (such as a website or mobile app) via application programming interfaces (APIs). Headless systems make few assumptions about the frontend experiences they might support and simply provide the content and functionality via APIs. This makes them very flexible and suitable for a wider range of applications than systems that come with a pre-integrated 'head'.

For example, Salesforce Commerce Cloud (SFCC) was originally designed as an eCommerce storefront management system and a pre-integrated storefront that supported eCommerce experiences like category and product detail pages. This simplified the implementation of the web storefront, but at the cost of supporting other eCommerce experiences like Apps.

Headless as a term is confusing. In reality you'll almost always have a head (frontend experience) or multiple heads (for omnichannel experiences).

What is MACH?

A MACH approach means you have very granular loosely connected microservices, if you don't approach digital transformation holistically, you can end up solving one problem and simultaneously create another. Bottlenecks we see tend to be developer or marketing based. Developer bottlenecks happen as minor changes to the end user experience have to go through them. Marketing teams become bottlenecks as they are forced to reference multiple fragmented tools which slow down their speed of production and overall success. Taking a holistic approach to digital transformation means you won't end up being so busy putting out the smaller bottlenecks, enabling you to truly realize the benefits in terms of agility and productivity

What are Progressive Web Apps (PWAs)?

Progressive Web Apps (PWAs) are an approach to building browser-based applications that offer an alternative approach (vs. websites and native apps) to developing a commerce storefront experience. A PWA is a modern technology framework that makes use of modern web browser features (such as progressive enhancement to improve page load performance), giving end users an experience that mimics a native application when the web application is accessed on a mobile device.

Unlike native iOS and Android applications, PWAs don't need to be downloaded to a smartphone or tablet and can be accessed directly by the web browser. PWAs are usually built using modern frontend JavaScript frameworks like ReactJs or VueJs and consume content and data using the very efficient JSON format (as opposed to HTML, used by websites).

What is Salesforce Composable Storefront?

Salesforce Composable Storefront, previously PWA Kit, is a storefront development framework and Managed Run-time service developed by Salesforce for use with Commerce Cloud. It provides a modern PWA-based alternative to the SFRA and SiteGenesis website-based storefronts.

Composable Storefront provides the code libraries and pre-built commerce components (page, basket, product grids, checkout, etc.) to accelerate the development of sophisticated eCommerce storefronts that can take advantage of modern web browser features. The Composable Storefront enables SFCC customers to go headless with their existing solution. Composable Storefront interacts with Salesforce Commerce Cloud (SFCC) APIs via the Managed Run-time service, using JSON. The framework is developed in ReactJS and requires React front-end developers for customization.

This provides the stability of the core commerce functions provided by SFCC along with the performance and experience enhancements of a modern web storefront that can be fully customized without needing proprietary Salesforce Commerce Cloud developers. Composable Storefront offers many advantages over the older SFCC storefront frameworks like SiteGenesis and SFRA, including improved page load speed, SEO and accessibility support.

What is Storefront Reference Architecture (SFRA)?

Storefront Reference Architecture (SFRA) is a web-site framework developed by Salesforce for Commerce Cloud customers. It acts as a starting point for building a responsive (i.e mobile-friendly) eCommerce storefront.

It combines some best practices in site design and technical architecture to provide a blueprint for merchants looking to build and customize their eCommerce storefronts. It includes:

- A JavaScript controllers development and customization model which allows teams to use a model-view-controller (MVC) development pattern
- The popular Bootstrap UI framework to optimize the shopping experience across devices

SFRA was launched in 2018 as a successor to the Site Genesis framework

What is Salesforce Site Genesis?

Site Genesis is an eCommerce reference website that enables the development of Salesforce B2C commerce experiences. It can be used as the basis of your own custom eCommerce storefront, although SFRA is recommended for new projects.

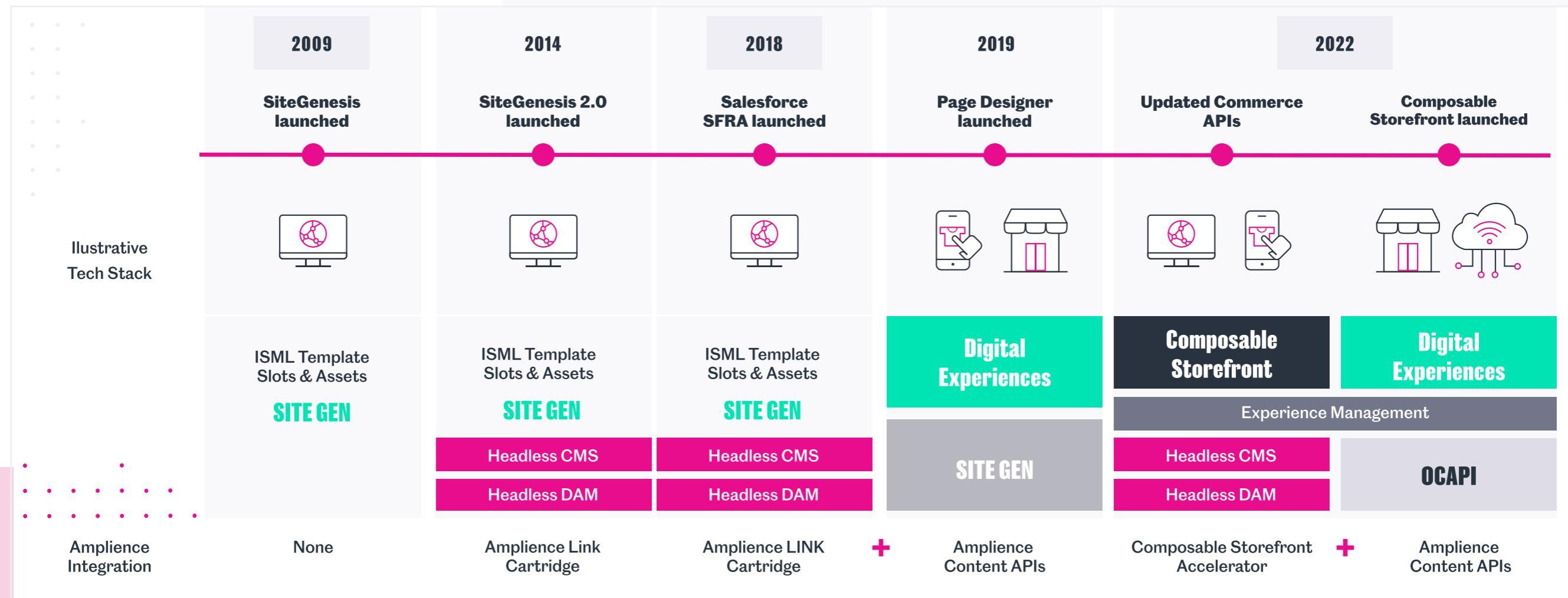
Site Genesis is showing its age as it wasn't built with mobile-first experiences in mind and can be slow and unwieldy to develop and use.

What is Salesforce Page Designer?

Page Designer is a visual (WYSIWYG) page builder designed to enable non-technical users to build and manage storefront pages built using SFRA. Non-technical business users can design, schedule and publish pages using a drag-and-drop interface by combining components built by developers into flexible page layouts.

Salesforce provides a Page Designer library with documentation and reference components to jumpstart usage. Page Designer is suitable for eCommerce content managers that have simple website design requirements and limited access to technical resources. Page Designer locks your content into the SFRA storefront, however, reducing content reuse and creating content silos and fragmented experiences.

Amplience Grows With Brands





Should My Business Move To Composable Storefront and Go Headless on Salesforce Commerce Cloud?



The question every Salesforce customer faces today is a simple one: does it make sense to move to Composable Storefront and go headless? And if so, how and what else does this impact? That's a lot to unpack.

This guide aims to break down whether Composable Storefront and headless is the right approach for your business. And, if so, how you should go about the tactical execution of that decision, whatever the outcome, by exploring the practical options you can deploy today. We will also look at the question of timing and seek to answer 'Why now?'

When is Headless SFCC a Good Fit?

For many brands that have a well-performing eCommerce business on SFCC, the decision to go headless is not straightforward. For those that are working on an outdated storefront experience, it's more clear-cut. To help decide if, how and when to go headless, you should consider the decision in the context of your customer experience strategy and the digital business transformation program that supports it.

Once you're clear on the direction of travel with respect to the customer experience and the technologies, business and cultural change you need to adopt to support it, the decision-making process will then most likely become more a question of when, not if, you go headless.



Strategic Considerations: Mind the Gaps

Let's tackle the strategic considerations first, starting with the evolving eCommerce business model, and then identify the gaps in customer experience, operations and experience management that brands need to close if they want to remain competitive and get ahead.

Ecommerce: A Business Model Under Pressure

The past 20 years have seen uninterrupted growth in digital. Millions of new consumers have adopted websites and mobile apps to achieve their goals, from buying goods and services to engaging in shared experiences like online gaming, finding friends, building communities and even finding new partners via social media and apps. This organic growth in digital consumers has floated many boats. But despite the one-off growth experienced during the pandemic (which is now receding), the brutal reality is that, as digital becomes the norm, the normal rules of the competition are re-asserting themselves and retail eCommerce teams face challenges in profitably growing revenues.

Nowhere is this more apparent than with customer acquisition costs. With more brands than ever seeking to acquire customers in digital channels, acquiring quality web traffic has become more expensive. Retail brands report that they are paying up to 30% more for equivalent traffic via AdWords and off-site campaigns.

Customer journeys are also more fragmented than ever before, with <60% of searches coming from Google and an increasing number happening in marketplaces like Amazon (where media costs are also exploding), as well as from social media like Instagram and Pinterest. This fragmentation drives up costs and results in a higher number of one-off transactions that are only marginally profitable at best.

Higher customer acquisition costs are changing the economics of eCommerce, driving a renewed focus on loyalty and customer lifetime value. Central to this change in strategy is a focus on the quality of the customer experience across the entire customer lifecycle and on the unit economics of each customer interaction, i.e. on the relationship between acquisition costs, customer lifetime value and cost-to-serve.



FOUR FACTORS HEAVILY INFLUENCE COST-TO-SERVE:

What type of business does your brand support?

1. Incomplete journeys (i.e. eCommerce visits that don't convert)
2. Customer service costs (typically borne by the contact-center)
3. Returns (which directly impact contribution margin)
4. The impact on the brand equity of poor quality experiences (i.e. reputational damage, leading to a loss of direct customer referrals, raising acquisition costs still further)

When presented like this, it's clear that improving the customer experience across the entire customer journey reduces the unit costs of every transaction.

Ecommerce experiences that can anticipate the needs of customers in real-time, for example, and personalize the experience by assembling the most relevant content, guided selling tools and products to present to the customers in that moment, result in higher conversions, order values and gross margins. Also, if the product experience is a good one that provides all the information a customer needs to buy the right products first time, customer service and returns costs are reduced, quality is improved and repeat purchases follow.

This is the flywheel that drives up customer lifetime value and reduces customer acquisition costs, fixing the performance gaps in the business model for eCommerce. To do this, however, eCommerce teams need to adopt a system and culture of continuous improvement, going beyond simple UX testing and optimization to drive a relentless focus on good quality customer outcomes. This shift puts a particular emphasis on the unit costs of customer experience and technology change. Simply put: your teams need to move faster to do more productive work for every dollar spent to ensure current and future competitiveness.

With the battle for digital talent now at fever-pitch, however, driving up input costs by as much as 15% per annum, retail leaders that don't embrace change are faced with a toxic cocktail of slower and more expensive growth. What's needed is a transformation of the eCommerce customer and employee experience, as well as the operations and technology infrastructure that supports it, to enable a high-quality customer experience that is relentlessly focused on driving more profitable outcomes.

Customer Experience Gaps

For many organizations, the eCommerce channel is just one component of a broader digital experience. And while eCommerce might be at the core of that experience, there are other channels such as stores, off-site interactions in marketplaces and social – not to mention email – that need to be taken into consideration as part of a broader strategy.

Shoppers have changed their behaviors as a result of the pandemic. And although they are now back in physical stores as well as online, their instinct is now digital-first, even when on the high street. The in-store experience, however, is digitally underwhelming, with inconsistent and siloed technologies that fail to deliver to customer expectations.

In parallel, the creator economy has exploded, with brand influencers using product and brand content to create lifestyle media and citizen-designers curating their own collections of cross-brand products on Pinterest and Instagram. As a result, many more shopping journeys now start on social media (63%, source: Think with Google, 2018) and end in your eCommerce channel. Often customers arrive directly to the product page without engaging with the full storefront journey (landing page, category lister page, product details page).

As such, their needs are different to those customers who take a more traditional route through the storefront. Customers who land directly on a product detail page (PDP) need more context than those who have traveled through the content and landing pages. But supporting multiple PDP templates using SFCC is complex and expensive, so is generally avoided.

The best performing retail brands are using customer data to craft more personalized experiences that deliver on the promise of customer-centricity, supporting customer journeys with unique content and shopping experiences that help them achieve their buying missions faster and with less friction. They are creating new versions of the PDP template that flex by category and customer context, enabling customers with the content and data they need to satisfy their immediate needs, while also drawing them deeper into the site and assortment, reducing churn and encouraging further purchase.

This customer-centric approach promises much in terms of driving up revenue-per-visit (RPV) and customer lifetime value (CLTV), which have become the new measures of a successful customer experience strategy. This is hard to do, however, if you are currently wrestling with multiple CMS and bolt-on eCommerce systems to support each channel and device, which makes it hard to synchronize and orchestrate the experience across multiple touchpoints. The result is an inconsistent brand experience, slower content production lifecycles and inaccurate and expensive errors in the experience that drive up customer service costs and reduce the quality of every interaction.

While this is possible to do in theory using Salesforce Commerce Cloud with SFRA or even Site Genesis, in reality, the challenges – such as duplicate content and experience workflows and building and managing multiple tech stacks – mean you'll have to resort to implementing separate vendor systems to deliver experiences on native apps, in-store and to manage content and experience in social and marketplaces.

And while SFRA on SFCC is a relatively performant storefront framework, it's still much harder to deliver consistently fast page-load performance and the kinds of SEO and Accessibility scores that ensure you pass Google's Core Web Vitals tests (See article [Improve Page Load Performance With Salesforce Commerce Cloud Composable Storefront](#)), resulting in lower conversion and higher customer acquisition costs than competitors who have made the switch to a PWA-based frontend.





HAVING THE RIGHT STRATEGY TO CLOSE THE CUSTOMER-EXPERIENCE GAP IS ABOUT:

1. Taking control of the experience production and management process so you can create once and publish everywhere (COPE)
2. Delivering the content and experience as data via APIs (i.e. headless) so they can be consumed by a range of frontends, from native apps to in-store tech
3. Using the most performant web storefront technologies (which will account for the majority of eCommerce interactions), which is a progressive web app (PWA) to drive up page-load performance, SEO and Accessibility scores

Employee Experience Gaps

Employee expectations have also changed. And the new cohort of digitally native millennials that have entered the workforce as retail associates also expect their employee experience to be digital-first. This has big implications for in-store technologies, with smart brands arming their digitally savvy employees with sophisticated in-store tech by which to better assist customers, enhancing the customer experience with better-tailored product recommendations and guided selling experiences and creating a more engaged and loyal workforce.

Similarly, with development expertise at a premium, the developer experience has become the new battleground for technology talent. Engineers seek work in organizations that are embracing technology transformation and moving to new architectures based on MACH principles and a composable approach. They want to work more productively, in agile teams that have the technology to support differentiation and innovation.

If you can't support their ambitions to work with new frontend technologies, APIs and headless systems, retention becomes a big issue. And without the continuity and knowledge retention that comes with a degree of tenure, transformation projects are stalled and change slows to a crawl.

This is also true of the digital marketing and eCommerce operations teams that support the customer experience. They too need modern tools and technologies to get their jobs done. And recruiting and retaining talent into an organization that has inefficient waterfall process models that are low-productivity and frustrating to use is an exercise in futility.

The key to an agile organization is employees that are empowered to get their work done and are as close to the customer experience as possible.

Managing the storefront experience using SFCC's Business Manager, however, even with the addition of Page Designer for page management, is complex and unproductive. It requires specialist skills or results in content that is locked inside of the eCommerce platform making it hard to power other experiences. Forcing your staff to use inefficient tools and outdated technologies is frustrating for them. And by sticking to the out-of-the-box SFCC commerce functions for capabilities like search and merchandizing, you could be ceding significant advantage to more nimble and progressive competitors.

A culture of empowerment, backed by capabilities, tools and processes that support agile ways of working is the key to attracting the kind of talent that results in the highest performing teams. Any strategy that focuses on the customer experience without reference to the gaps in the employee experience is going to slow you down, frustrate your transformation and change programs and result in higher team costs. It will also reduce your ability to rapidly respond to changing customer requirements.

Storefront Operations Gaps

With unit costs a key consideration in any strategy, it's imperative to set up and manage your eCommerce operations efficiently. But better processes and more productive employees are not enough on their own. What's needed are experience management processes and tools that can scale to drive up the quality of every interaction without linearly increasing costs.

For eCommerce teams, higher-quality customer experiences are ones that more efficiently service the entire customer lifecycle from consideration to purchase and beyond to usage, service, and returns. To do this requires an unprecedented amount of content, measurement, change, and testing in and of the customer experience. This requires a transformation in approach from manual curation of the experience to automation through the use of customer data and analytics technologies. These can positively influence customer outcomes in real-time by changing elements of the customer journey, from the content itself (text and media) to the structure and organization of the site (menus, order of content and functional elements of the experience).

These considerations become even more relevant when multiple channels, brand fascias and locales are required, as each will require its own look and feel, taxonomy, content and data.

And to take this logic to its ultimate conclusion, each segment and individual customer journey should have its own custom, data-driven experience, assembled in real-time with minimal manual effort, except in the creation of the experience elements themselves.

For example, high-quality localization is about more than just content translation. Customers in locales with limited knowledge of your brand would benefit from a site taxonomy that supports marketing content earlier in the journey. One that allows you to introduce the brand story and provide proper context to the category assortments and product. Customers in home markets, on the other hand, might be more promotion-sensitive and benefit from a taxonomy that prioritizes deals and add-on-purchases above marketing content.

Having a one-size-fits-all site structure is not going to cut it. But managing locale-specific taxonomies and templates for each region using older (SFRA and Site Genesis) storefronts is complex as they rely on the underlying catalog structure and template relationships that are fixed in SFCC's Business Manager. By enabling the flexible generation of the storefront navigation and site map by referencing the taxonomy as data, generating these kinds of locale-specific storefronts can be made fast and cost-effective to implement.

And while there are many A/B testing, personalization, content targeting and product recommendation solutions that can automate various elements of the experience, they are generally designed to take control of a content area (slot) in the site template and inject HTML in the form of product grids, promotional banners or targeted content.

In the main, these solutions act on the frontend of the site directly, injecting content into tagged areas of the page. And although tag management tools have evolved to simplify the management of these third-party add-ons, they come at the cost of reducing the page-load speed of the storefront, while increasing the complexity of the underlying web templates.

These templates are complex and difficult to change, requiring slow and expensive development cycles, chewing up valuable resources that could be more productively deployed elsewhere. Evolving to a data-driven, automated storefront experience would deliver far more relevant customer journeys at much lower unit costs. But this is out of reach for those still using the legacy experience management model embedded within Site Genesis, SFRA and the Business Manager and Page Designer management tools.

Your strategy needs to address the fundamental issues inherent in all legacy approaches to experience management for it to evolve and close the competitive gap with those that have taken a more modern approach.



Technology Architecture Gaps

As is often the case with digital, the underlying architecture of the eCommerce system is the limiting factor that is responsible for the challenges we've identified so far. That should come as no surprise, as eCommerce solutions like SFCC were architected over 20 years ago in a pre-mobile age when the desktop web browser was all there was to digital experiences. As such, SFCC is designed and optimized for the creation and management of desktop web storefront. And yet the dominant consumer device is now a mobile phone with radically different characteristics to the broadband desktops used to access the web when SFCC was originally conceived.

At the heart of the problem is the web-templating system used by SFCC. This paradigm assembles the experience by linking together pages, which are themselves composed of templates, defined in SFCC's ISML web-cms templating language. ISML enables developers to create page templates that combine presentation elements and functional components that interface with the backend eCommerce APIs to enable functionality such as search results and basket management. And to retrieve dynamic data elements like pricing, promotions and inventory status for display in the page.

ISML templates generate whole pages of HTML and use additional JavaScript libraries to control the way in which the page is experienced in the web browser. Brand and design is managed through CSS and images (JPEGs and PNGs), which can be changed independently of the relatively fixed templates.

In this model, which has been the dominant one for the past 20 years, teams are structured around the relative complexity and associated change lifecycle of each layer of the experience, making it hard to work in a truly agile way and limiting experience change speed to the speed that developers can make and test template change.

Integration of third-party components is also challenging, with SFCC's proprietary, cartridge-based model requiring significant development for integration and implementation. When using SFCC in its original incarnations with SFRA or Site Genesis, the customer and employee experience is suboptimal and experience automation is complex and fragmented.

It's only by fixing the problems inherent in the architecture of SFCC that you can close the experience and operating gaps so that your brand can evolve into a more agile, composable and productive future.

To do that requires a move to headless SFCC and a new PWA-based storefront frontend. We'll look at that next.



THERE ARE SEVERAL PROBLEMS WITH THIS MODEL, HOWEVER:

1. SFCC's ISML templates are complex to create and manage, requiring cumbersome software development and testing lifecycle
2. Templates start simple but evolve over time to become a code monolith with many interdependencies between experience elements, data and functionality, slowing change cycles
3. ISML requires expensive and hard-to-find Site Genesis or SFRA development skills and developers are adopting more popular modern development frameworks
4. Content is managed in the Business Manager or Page Designer using slots and content-assets model which traps the content in SFCC, making it hard to re-use in other channels and experiences
5. The HTML output of the ISML templates does not play well with modern frontend storefront frameworks like Salesforce Composable Storefront

Why Go Headless on SFCC?

Headless is a term used amongst technologists and software vendors, along with terms like MACH (micro-services, API-first, cloud-native, and headless) and composable (as in 'composable commerce').

These modern approaches have the potential to transform the capability of your teams and speed up customer experience innovation. Put simply: in headless mode, SFCC communicates with the storefront experience using APIs (application programming interfaces). This has several advantages over the older and more tightly integrated Site Genesis and SFRA web storefronts still in use today:

1. Frontend flexibility: you can run a range of modern, mobile-optimized frontends from the same commerce instance, including single-page apps (SPAs) and progressive web apps (PWAs). By using a headless solution's APIs you can also power native mobile apps and even internet of things (IoT) devices from the same instance of the commerce backend, simplifying support for a range of customer use cases that would previously have required their own dedicated solutions

2. Backend flexibility: composable commerce is all-the-rage when it comes to modern commerce experiences. Composable means you have the freedom to pick and choose the backend components for each main functional area as you see fit, whether that's for content management, site search, merchandising or anything else

3. Performance: by interacting with the APIs directly, your applications consume the backend data using JSON, which is more efficient when used with a modern PWA-based mobile frontend. Older website frameworks like Site Genesis and SFRA use HTML, which is slower to transfer and comes in larger, whole-page blocks

4. Productivity: by using better tools your team can do more with less (and do it better) – launch more campaigns, target more segments, create richer content and use better-guided selling tools. By running a number of frontend applications from the same set of headless APIs, your teams can create once and publish everywhere, saving them time and simplifying the orchestration (a fancy name for 'control') of the customer experience.

SFCC is now available in headless mode, so as an SFCC customer you can now access the many benefits of headless without the cost and risk of an expensive re-platform, buying you time to plan the transition to a MACH and composable future on your own terms. By using SFCC in headless mode you no longer need to use the default Site Genesis or SFRA storefronts, enabling you to access the commerce APIs directly and choose the frontend that works best for you.

You will still need at least one head – a storefront – and perhaps more heads if you want to give customers and employees native mobile apps to use. But you can add those later as required with much less effort. And by moving to a frontend that is less tightly integrated into SFCC you no longer need to use the old ISML web-cms templates that are holding your teams back. So you can take a 'composable' approach to your commerce solution, choosing to use the best software vendors for the specific needs of your brand and customer experience, giving you far more freedom than before.

So now we've covered the main reasons to use SFCC in headless mode, which frontend should you use for your storefront?

PWAs and Composable Storefront: Which Storefront Approach Is Right for You?

Progressive web apps (PWAs) are an approach to building browser-based applications that offer an alternative approach (vs. websites and native apps) to developing a commerce storefront experience.

PWAs use modern web browser features (such as progressive enhancement to improve page load performance) that give end-users an experience on par with native mobile applications. They're built using modern frontend JavaScript frameworks like ReactJs or VueJs and consume content and data using the very efficient JSON format (as opposed to HTML used by websites).

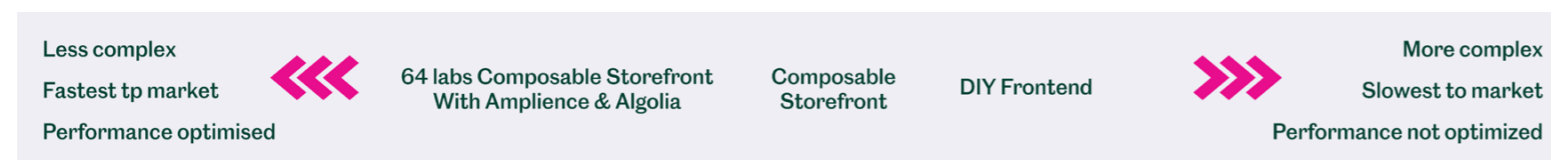
There are three potential approaches to storefront implementation with a PWA-based approach. Below we've laid out the strengths and weaknesses of the available options:

	Roll your own	Vanilla Composable Storefront	Composable Storefront Accelerator (by 64 labs)
Strengths	<ul style="list-style-type: none"> Flexibility to choose another PWA solution such as Vue Storefront or to build one from scratch Supports responsive mobile experiences Allows customers to leverage a headless commerce approach / pursue a composable commerce strategy Likely to see site speed improvements over SFRA but will require development to optimize 	<ul style="list-style-type: none"> Officially supported by Salesforce, with active support and development The majority of components are built on open-source technology so there are more developers who can support it (compared to SFRA) Supports responsive mobile experiences Allows customers to leverage a headless commerce approach / pursue a composable commerce strategy Site speed improvements over SFRA 	<ul style="list-style-type: none"> All the benefits of Composable Storefront + the below: Faster time to market - pre-integrated with Amplience (for content, media and experience, Algolia (for search and assortment) and Salesforce's powerful APIs (to manage catalog, cart, promotions, and customer groups). Optimized for page-load performance, accessibility, and SEO Reduce development bottlenecks
Weaknesses	<ul style="list-style-type: none"> Implementation time longer, therefore, increasing time to market Will require more ongoing support and maintenance Unknown TCO (total cost of ownership) Development bottlenecks - managing experience in code, not UI, is highly likely unless you expand the scope to also integrate experience management 	<ul style="list-style-type: none"> Development bottlenecks - managing experience in code, not UI Lacks pre-integrated content & search capability (compared to Composable Storefront Accelerator) Increased time to market - requires development and implementation 	<ul style="list-style-type: none"> No control to change pre-packaged content (Amplience) or search (Algolia) solution

Let's Look At Them All in More Detail:

ROLLING YOUR OWN PWA-BASED STOREFRONT

Adopting a headless approach means that you have the flexibility to go off the beaten track and implement a PWA storefront that isn't Salesforce Composable Storefront. In this regard, you could opt for a pre-built storefront framework like Vue Storefront that provides a framework and library of components to get started quickly (although it doesn't currently have a pre-built integration with SFCC).



Salesforce Composable Storefront?

Composable Storefront is Salesforce's answer for a modern mobile storefront, designed to be the next generation SFCC frontend: a PWA (Progressive Web App) built using React, a modern JavaScript language.

It interacts with the SFCC backend APIs, and any other backend system APIs directly, using the fast and efficient JSON format. It also renders (i.e. builds) the customer experience more efficiently, updating each part of the screen as required rather than reloading the entire screen one page at a time (in HTML) after every user interaction.

Using a modern and flexible frontend (Composable Storefront) coupled with extendable and easy-to-integrate data (backend system APIs) allows you to maintain the reliability and trust of the Salesforce platform without being locked into a website that often requires specific developers to manage. You can create any application you want (web, app, in-store kiosks, tablets or anything in between) with shared data across all endpoints.

This makes it much faster and more flexible than older website storefront frameworks like Site Genesis and SFRA. And it's also less costly as you no longer have to rely on proprietary SFCC developers.

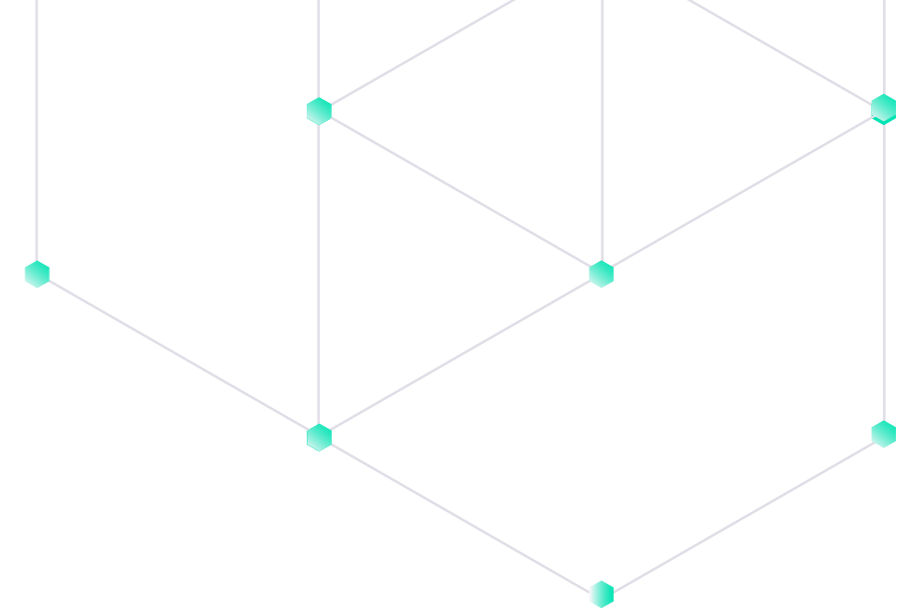
The 64Labs Composable Storefront Accelerator?

Moving to Headless on Salesforce with a new PWA-based storefront sounds great, but there are implications for the eCommerce team. Removing the tightly integrated Site Genesis or SFRA web storefront means that SFCC's experience management tools like Page Designer and the CMS in Business Manager no longer work.

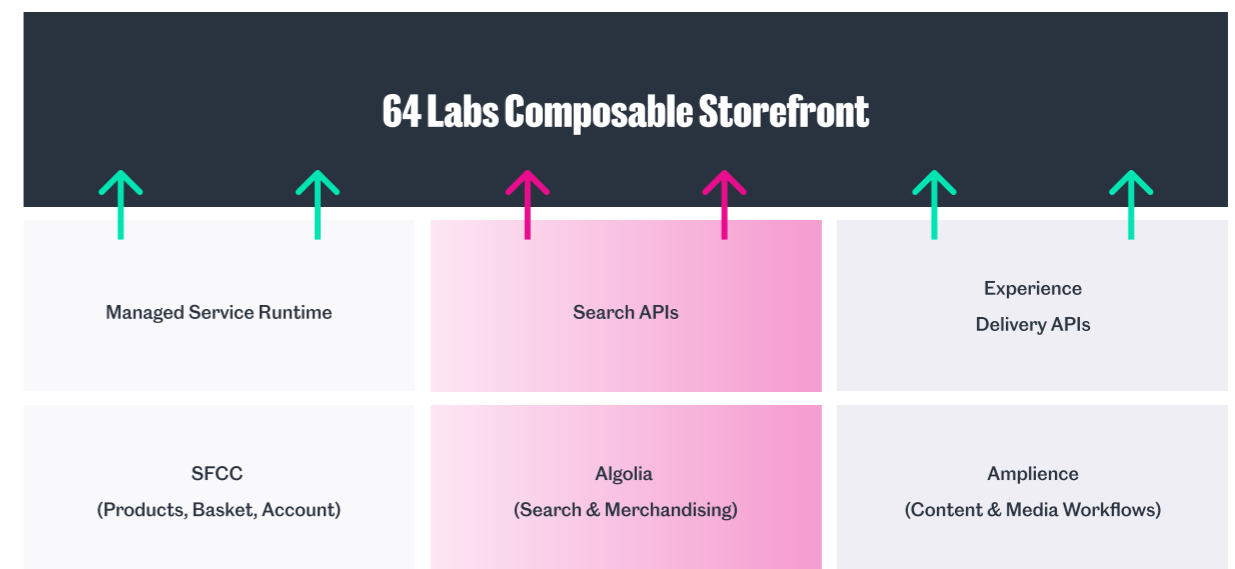


THIS IMMEDIATELY RAISES SOME CRITICAL QUESTIONS:

1. How will marketers manage landing pages, site content, blogs and campaigns?
2. How will merchants create assortments and manage category pages?
3. What tools do my content managers have to manage the experience and optimize the site for SEO and accessibility?
4. What happens to my existing content?
5. The HTML output of the ISML templates does not play well with modern frontend storefront frameworks like Salesforce Composable Storefront



The good news is the smart folks at 64Labs have thought this through and built their Composable Storefront Accelerator to use Amplience (for CMS, DAM and rich media) and Algolia, (for search). These two integrated capabilities replace Salesforce's tools with more modern, headless alternatives. The 64labs Composable Storefront Solution is a headless reference framework pre-integrated into Amplience (for content, media and experience), Algolia (for search and assortment) and Salesforce's powerful APIs (to manage catalog, cart, promotions and customer groups).

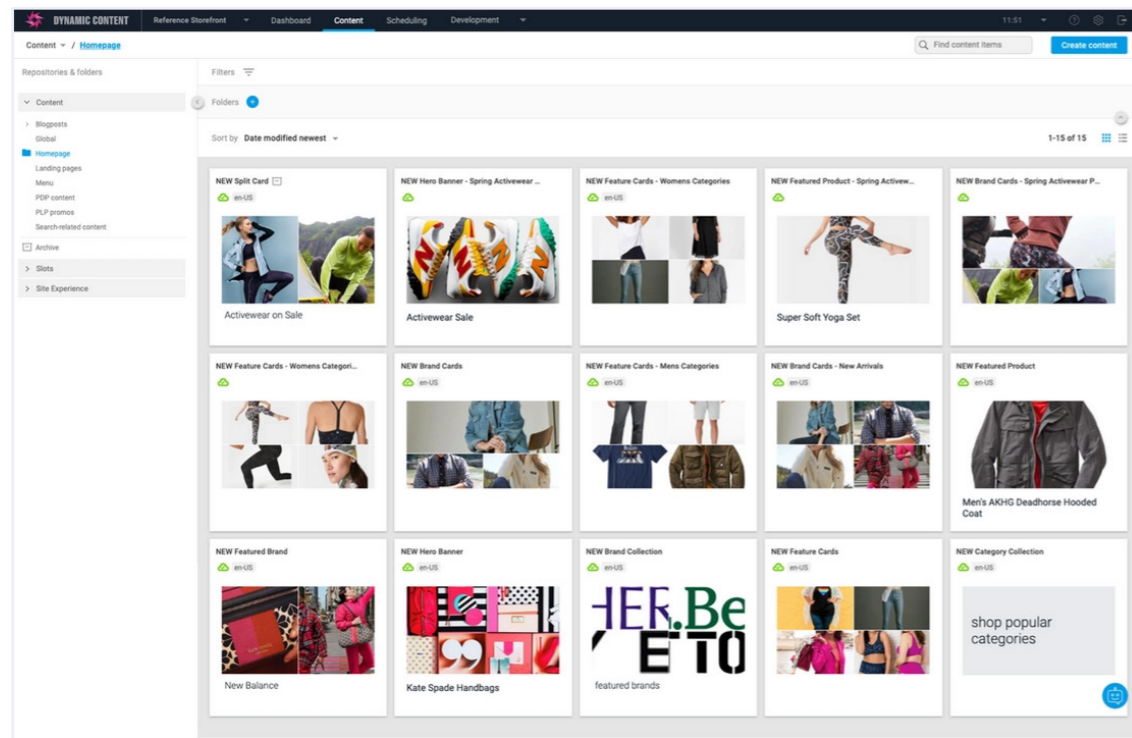


The 64Labs Composable Storefront Accelerator uses a specially upscaled version of standard Salesforce Composable Storefront, which has been optimized for even better page-load performance, accessibility and SEO.

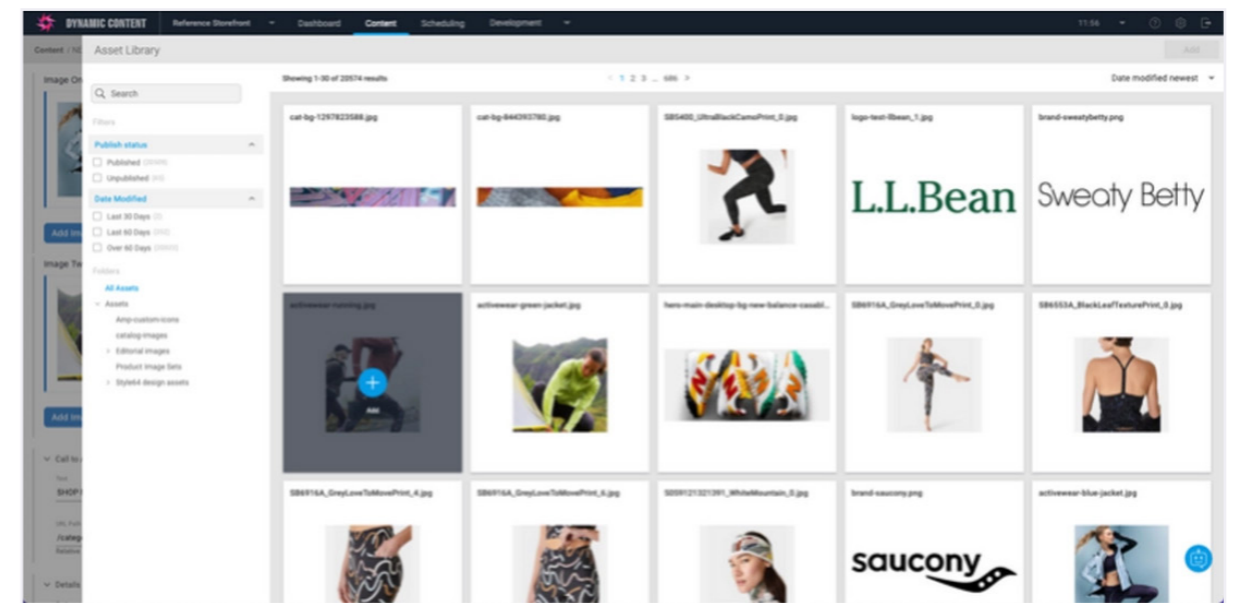
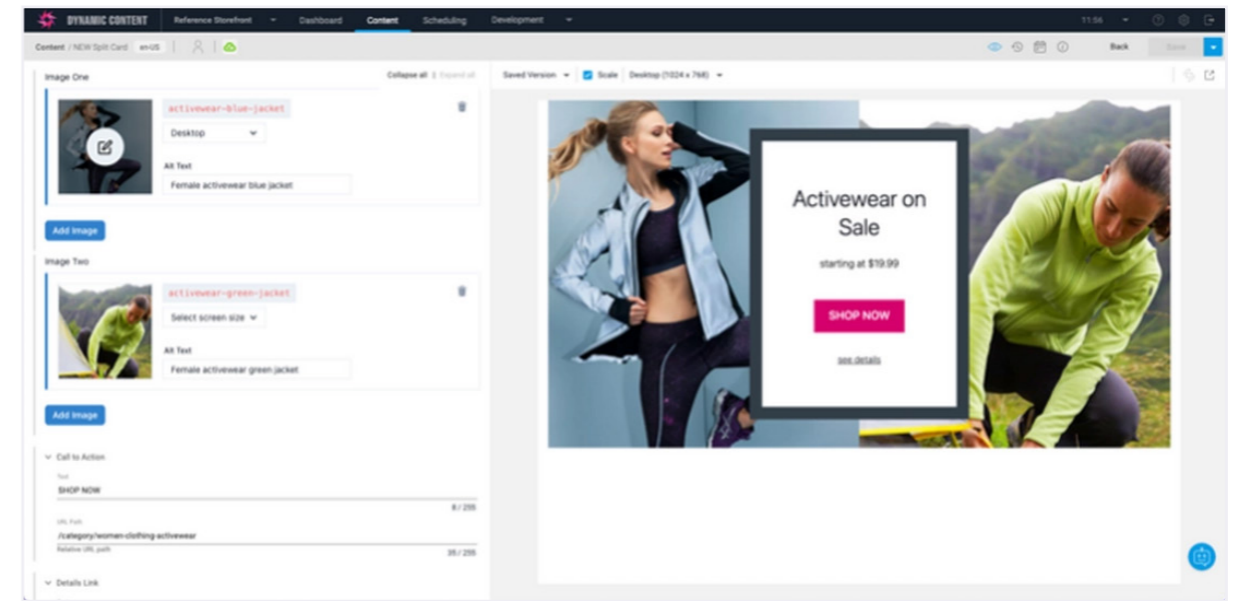
Going back to our four questions , then, let's answer each in a little more detail:

How Will Marketers Manage Landing Pages, Site Content, Blogs and Campaigns?

Amplience Dynamic Content and Dynamic Media provide a powerful set of capabilities to create and manage content and media, from simple banners to complex landing pages. Business users have access to a content authoring console from which they can create a content library and push content into slots within the experience.



You can upload and select media for use within content modules, safe in the knowledge that Amplience's Dynamic Media APIs will automatically shape and size it for whatever device and channel combination you need. Content authors can visualize individual content items or whole pages from within the Amplience CMS.

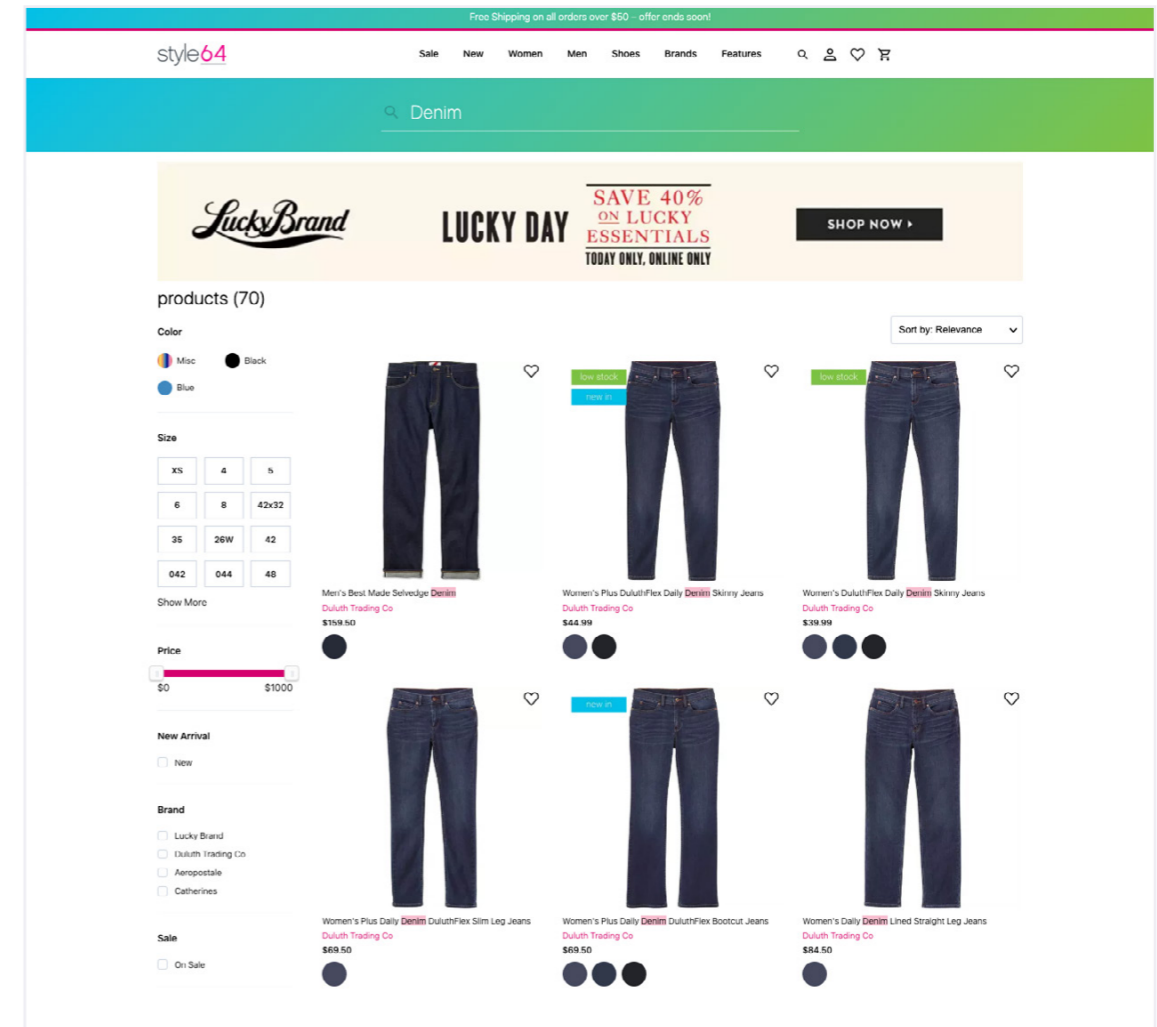
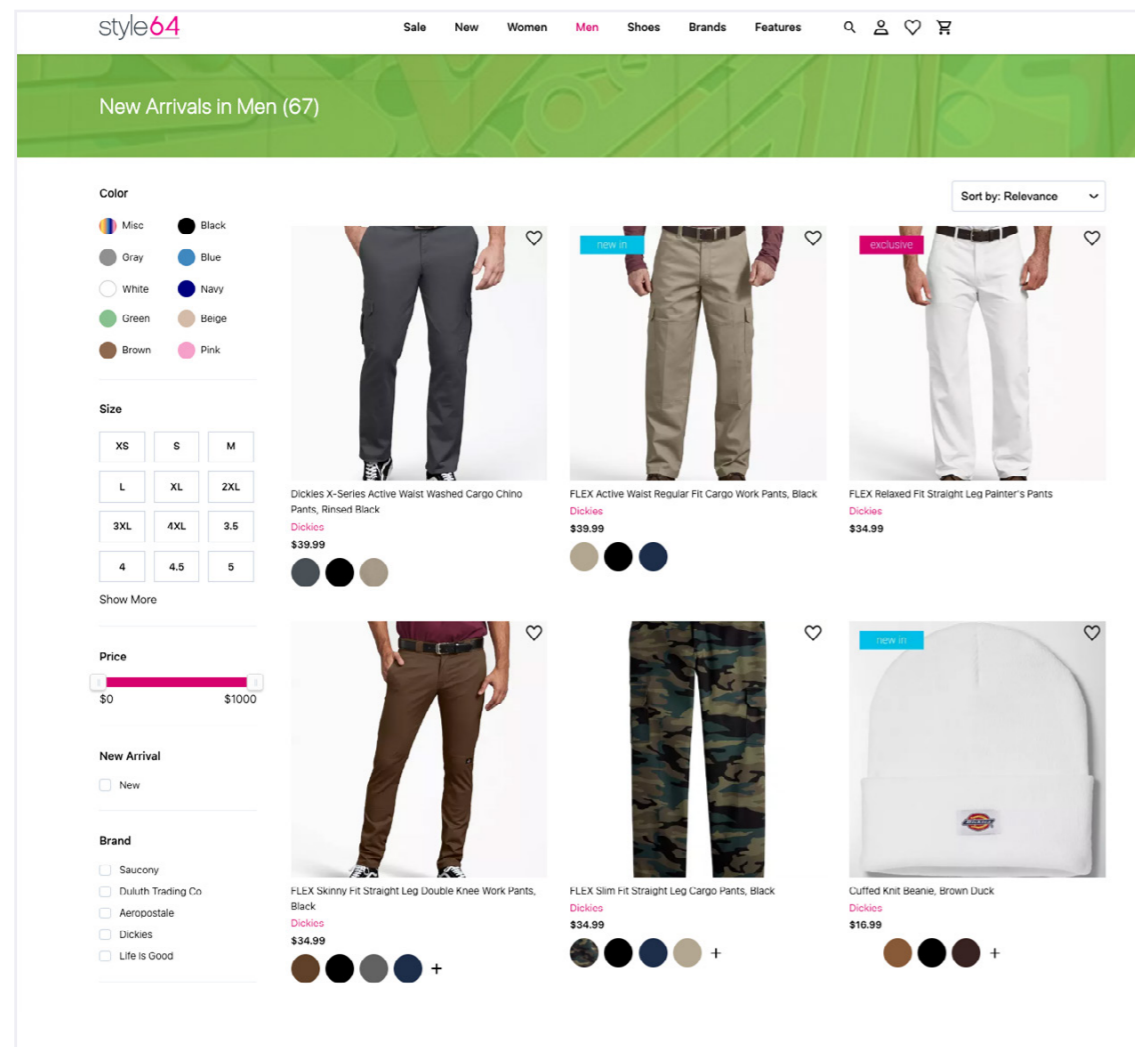


When it comes to campaigns and content drops, you can schedule them from within the content calendar, which enables marketers to set up and manage the site in advance. And with the integrated preview tools, you can see the whole storefront across all channels and devices before it goes live, giving business owners confidence that the customer experience will be exactly as they want it to be.

How Will Merchants Create Assortments and Manage Category Pages?

This is where Algolia comes in. It is a modern headless search engine that can manage site search and product lists throughout the experience. In the 64Labs Accelerator, Algolia indexes the product catalog which is held in SFCC. Algolia can then be used to bring back product assortments as a parameterized search result.

The 64Labs Accelerator uses Algolia to drive the product lists for category pages, which feature a range of filters to help users find what they are looking for:



So far, so good. But isn't that what Salesforce's own search does already?

Well, yes and no. Salesforce search is perfectly capable but lacks the advanced merchandizing features of Algolia. Merchandizing involves promoting specific products or categories of products.

It lets you:

- Offer recommendations to your users
- Highlight your inventory's best products
- Follow trends
- Improve conversion rates

Algolia offers multiple merchandizing capabilities, all powered by Rules. It has a visual editor with a drag-and-drop interface to easily merchandize search and browse experiences. You can also create advanced rules with the API clients or the manual editor on Algolia's dashboard. You can even personalize the results (if you're looking for a more advanced use case example).

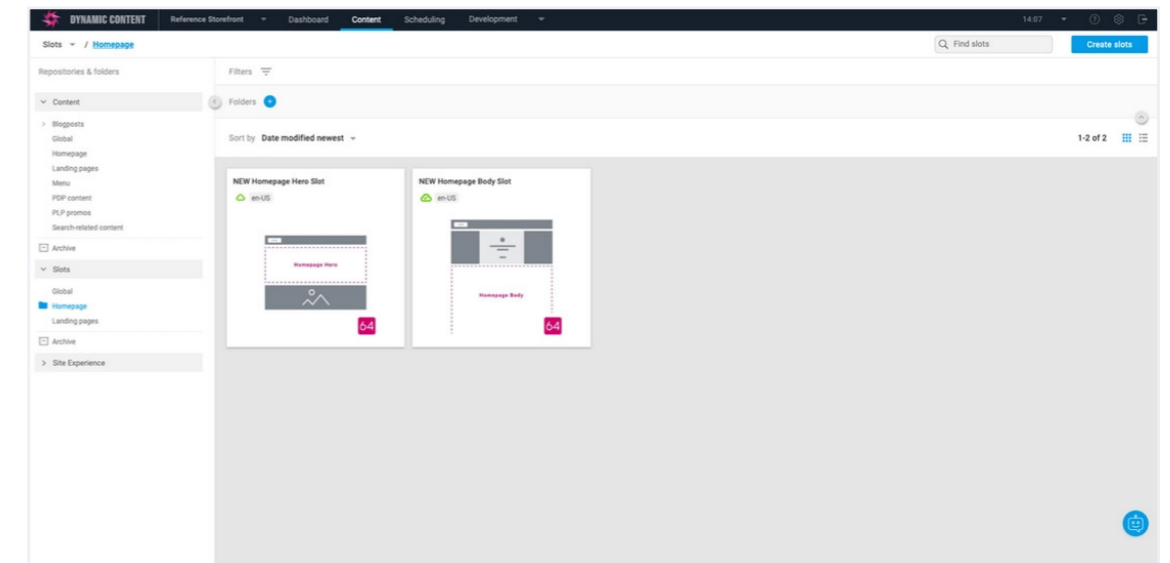
The Algolia delivery APIs are also very fast, improving page-load performance wherever they are used in the experience. So while Salesforce search will do the basics well, Algolia will help to take your commerce experience to the next level.

What Tools Do Content Managers Have to Manage Storefront Experience and Optimise the Site for SEO and Accessibility?

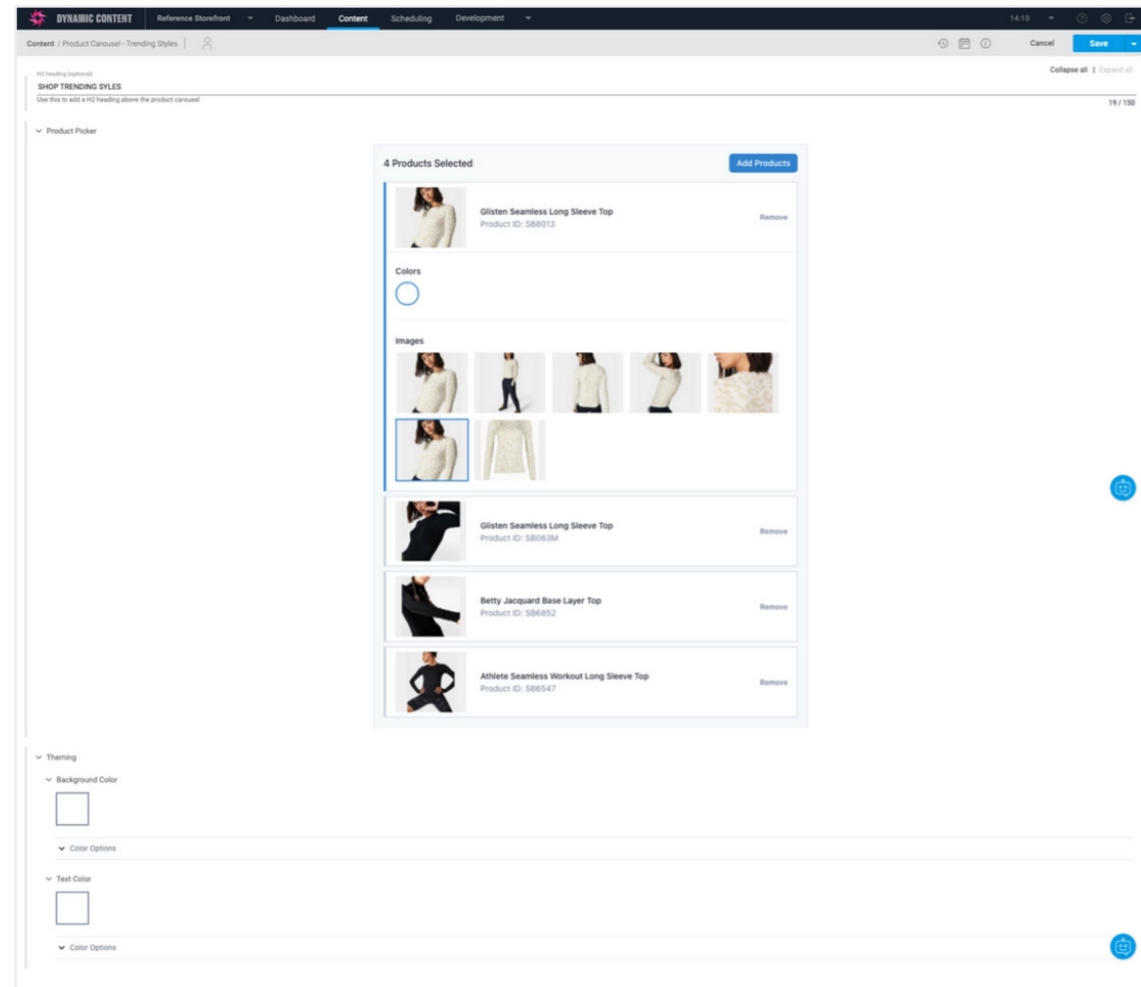
Amplience is a full experience management platform designed to help business users take control of the experience and manage content, not code. This gives non-technical users the freedom to make changes to experience elements without resorting to time-consuming and expensive developer-led change.

The 64Labs team has built a range of capabilities into the Composable Storefront Accelerator that take full advantage of Amplience's powerful capabilities:

USE OF CONTENT SLOTS: Amplience Dynamic Content has a flexible slot model. You can add slots to pages and control them independently of each other, making it easy to manage content in and out of pages. Slots can also be nested to create a stack of content items. And you can set them up to be site-wide (for example the content slug at the top of the Style 64Labs sample store that says 'Free Shipping').



USE OF UI EXTENSIONS: A UI extension in Amplience gives content authors access to the underlying data in SFCC or any other system that powers the storefront. So a merchant can easily curate a product slider that features products selected for a specific campaign, or set up a custom assortment, by creating a filtered Algolia search with the specific parameters required.



Additional experience control elements provided by 64labs include theming controls, meta-data descriptions for content to improve SEO and accessibility support, and the ability to change slot orders in pages to tweak the layout and order.

What Happens to My Existing Content?

In conjunction with your tech team or systems integrator and with assistance from the Amplience Expert Services team, 64labs will migrate your content from SFCC to Amplience as part of the Composable Storefront implementation process.



THERE ARE THREE KEY ADVANTAGES TO THIS APPROACH:

1. Content is now stored as data externally to SFCC so you can easily re-use it in a range of frontend experiences like native mobile apps and in-store systems
2. Content is now portable – you can manage it separately to the SFCC instance, making upgrades and re-platforming simpler and less risky than before
3. If internationalization and localization of content is required, content managers can use Amplience’s powerful tools to create localization workflows that use a range of external translation providers



Sales Conversion, Average Order Value and Gross Margin

The 64labs Composable Storefront Headless Salesforce Accelerator improves page load performance over even the fastest SFRA-based storefronts. This has an impact on conversion-to-sale. Google itself has long proclaimed that less load-time waiting leads to better conversion, but the link between speed and driving more sales is not always that obvious.

To understand why, we need to step back and look at how customers interact with the storefront. Whether they have a specific shopping mission in mind or are just looking for inspiration, all shoppers start their journey with a fixed amount of time and attention that they are willing to expend to get the job done. To use a simple metaphor, we can imagine this attention as being the water filling a leaky bucket. As customers interact with the storefront they expend effort and the water level in the attention bucket slowly drains away. Slow loading, poorly performing storefronts increase the friction of every interaction, draining the customer attention bucket faster.

By improving page-load speed, increasing the accuracy and relevance of the content and by making it easier to find the products they are looking for, a well-designed Composable Storefront reduces this friction, leaving more attention in the bucket which means that shoppers have more time to engage with the content and product catalog. This results in shoppers having more time to add more products in-basket and increases the number of completed sales, driving up conversion % and average-order-values. Higher conversions also drive better gross margins as merchandisers achieve higher sell-through rates without discounting.

SEO, Accessibility and Reducing Customer Acquisition Cost (CAC)

A better performing site also contributes to improved search engine rankings as performance is a key component that Google uses in its ranking algorithms. Not only do you improve conversion-to-sale but you generally experience an increase in the volume of customers (traffic) thanks to improved search engine results page (SERPs) rankings.

For more information on how Composable Storefront improves site performance, we highly recommend reading our blog post on ["How To Improve Page Load Performance With Salesforce Commerce Cloud Composable Storefront"](#).

Accessibility is another important contributing factor, not only for search engine algorithms and rankings but also for your business's bottom line. According to the WHO, over a billion people across the globe live with a disability. That's around 13% of your addressable market. And with an aging population it's a growing segment. Having an accessible site is more than just a moral or legal imperative – it should be a core part of your strategy to address a market that is easily served by following simple best practices.

Improving SEO, SERPs and accessibility contributes directly to reducing customer acquisition costs (CAC) at a time when paid acquisition channel costs are increasing rapidly. Diversifying and strengthening your acquisition channels by delivering a more performant site is one straightforward tactic you should consider as part of your overall acquisition and SEO strategies.

Customer Service and Returns Costs

Shoppers come to your storefront to solve problems. They rely on the information they receive to make good decisions. And they make those decisions by engaging with content and searching the product catalog.

“ If you'd like to learn more about accessibility, check out our blog post on: [How To Make Your Ecommerce Site More Accessible](#) ”



Inaccurate and misleading content is a source of frustration that can directly impact conversion. But it is also responsible for increasing costs across the business by:

1. Driving up the number of customer support interactions as customers seek to resolve issues they cannot easily resolve in-storefront
2. Increasing returns, caused by shoppers ordering the wrong product or over-ordering several products to compensate for a lack of product information

For many brands this results in a considerable hit to contribution margins, reducing the amount they have to invest in improving the eCommerce customer experience.

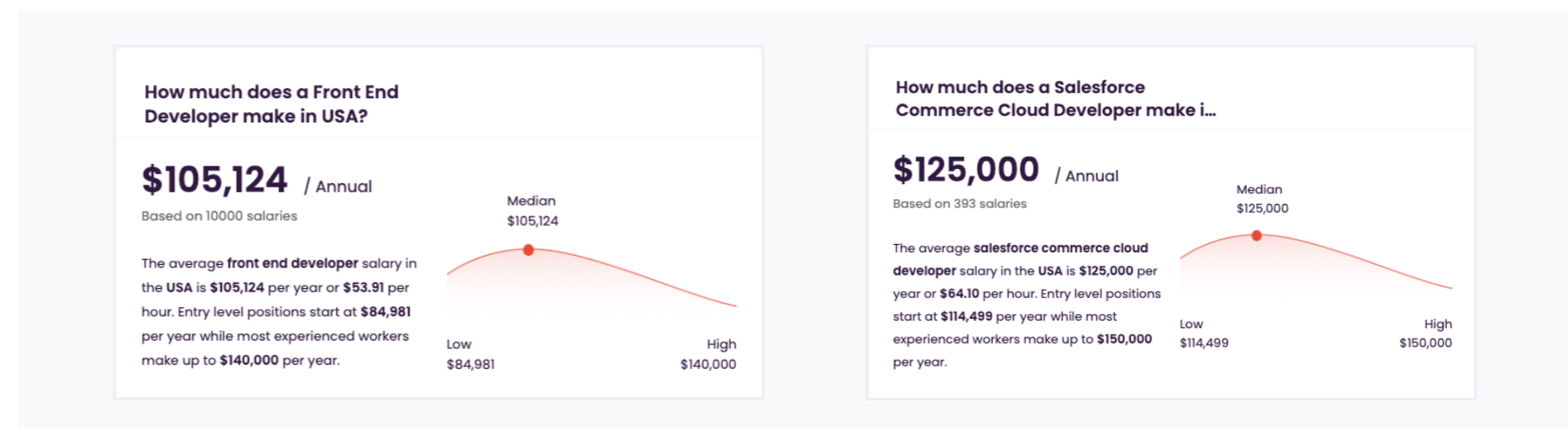
The 64labs Composable Storefront Headless Salesforce Accelerator features a significantly improved CMS (Amplience) and search (Algolia) capability over the standard SFCC content and search tools.

Better tools, combined with an appropriate taxonomy and content strategy, enable eCommerce teams to improve the quality of supporting content throughout the experience, reducing costs and improving customer experience.

Employee Recruitment and Retention Costs

Employee recruitment and retention costs can be reduced when implementing Composable Storefront and going headless on Salesforce for both technical and business teams.

1. On the technical side, proprietary Salesforce Commerce Cloud developers are in relatively short supply, driving up costs with the median salary of \$125,000 according to talent.com when compared to the talent pool of frontend javascript developers whose median salary is \$105,000. For additional perspective, in Q3 2020, according to Developer Economics, the number of software developers using JavaScript was 12.4 million. This means that 53% of all developers in the world were using JavaScript at some point in 2020. Slashdata's State of the Developer Nation Report in 2021 estimated that this number had grown by a further 1.4 million JavaScript developers to over 13.8 million total. While it's difficult to estimate the total number of Salesforce Commerce Cloud developers worldwide, a quick LinkedIn search returns only 4,300 "Salesforce Commerce Cloud Developers" and 1,700 "Salesforce Commerce Cloud Architects". This all leads to one simple conclusion: it's easier, cheaper and faster to recruit frontend and JavaScript developers to work on headless implementations.

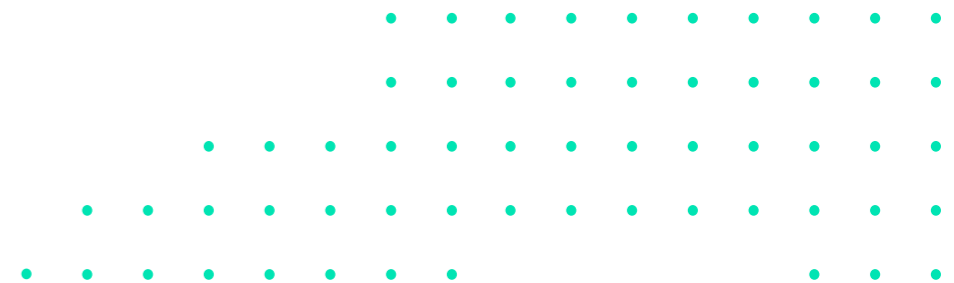


Sources:

<https://www.talent.com/salary?job=Front+End+Developer>

<https://www.talent.com/salary?job=salesforce+commerce+cloud+developer>

2. On the flip side, attracting and retaining talent in today's global talent pool is more challenging than ever for marketing and merchandising teams. The transient nature of the workforce today means they can work for anyone, from anywhere. This freedom means people are more selective about who they choose to work for and will evaluate their employee with a much sharper eye given the number of options and opportunities available to them. Organizations need to provide a good working environment, including tools that make their jobs as easy and as enjoyable as possible through user-friendly interfaces. Candidates can easily be put off and look elsewhere when they learn you're using a legacy system or vendor product that hasn't been updated in years.



Defining an Approach to Headless Salesforce Implementation

Now we've discussed why you should go headless with Salesforce and highlighted the common pitfalls you may encounter when adopting a headless approach, you might be wondering how best to mitigate and avoid potential issues beyond implementing Composable Storefront or the 64labs Accelerator. To accurately project that future state you first need to look at the problem.

Defining the Problem

Let's take a step back for a moment from technology. There are better questions that need to be asked first before "Is headless right for my business?" A better question to start with is: "Does my business have the right tools, processes and people to execute and compete effectively?" This question can be expanded further. The list below is not exhaustive but should begin to highlight the importance of *asking the right questions*.

- What is our conversion rate and can it be better?
- What is our page load time and can it be improved?
- What does our holistic customer journey look like? Are we catering to our customers' needs at every stage to the best of our ability and understanding?
- How long does it take my team to publish changes to production?
- How frequently do our experiences change?

- How sophisticated are our experiences?
- Do we personalize experiences by broad segments or at an individual level?
- Do we have governance and control over workflows and processes?
- Do we have content silos and disjointed teams?
- Do we have a clear content production workflow?
- What are the blockers in our content production workflow?
- Does our team have to manually publish changes live at awkward times in the night?
- Do our customers expect to connect with your brand beyond traditional cookie-cutter web and in-store experiences?
- Does my critical path perform and convert? Is it above or below industry average?
- Do my experiences cater to a modern, non-linear shopping journey?
- Are my experiences consistent across channels and devices?

Define the Objective

Companies don't just go headless for the sake of better tech alone. They document pains and align specific objectives to achieve with clear measurable deliverables and KPIs that demonstrate the pain is solved, whether that is site performance, time to market, agility, uptime or reducing TCO.

It just so happens that headless can be the right architectural pattern that delivers against these objectives and solves the underlying pain.

Determine the Solution Requirements

Now that the pain and objectives are laid out, you can begin to map and document your requirements. What would the solution look like? It's typically grounded in technology and might start with a simple feature list. As you go deeper into your evaluation you may conduct jobs to be done analysis and realize the objective extends beyond a list of capabilities.

1. For businesses to effectively manage headless experiences they require more than a simple headless CMS and frontend technology.
- 2.
3. An agile content production workflow and waterfall processes get in the way of delivery. This needs to be balanced with the right level of corporate and brand governance
4. Campaign management tools that mirror how commerce, marketing and development teams operate. This includes robust scheduling and preview capabilities so teams can publish changes to experiences with confidence
5. Rich authoring experience and visualization
6. The ability for development teams to understand business requirements and provide the right guard rails to ensure experiences are functional, optimized, consistent and comply with governance requirements
7. A flexible, API-first content model to enable content to be re-used in multiple devices and contexts
8. This is where experience management comes in and takes a headless Salesforce implementation to the next level.
9. The Business Case for Experience Management
10. Without experience management, whether you look to augment your existing implementation (SiteGen or SFRA) or decide to go down the headless or MACH route with or without Composable Storefront, the question remains: do you have the right solution (described above) to solve the pain your experiencing today and ultimately drive business growth?
11. Experience management is more than just content management. Experience management extends to where and when content lives, across which devices and pulls in data from other systems to enrich customer and back-office experiences.

Next steps

Thanks for reading this guide. We hope you found it insightful and practical in your decision-making process. Want to discuss any of the topics, concepts or options presented in this guide?

Get in touch one of our commerce experts today.

Contact details if you're interested in exploring Ampliance for SFCC + SFRA or Ampliance SFCC + Composable Storefront

Further Resources

<https://trailhead.salesforce.com/en/content/learn/modules/cc-digital-for-developers/digital-ref-architecture>

<https://developer.commercecloud.com/s/article/PWA-Kit>

<https://github.com/SalesforceCommerceCloud/pwa-kit>

<https://developer.salesforce.com/docs/commerce/pwa-kit-managed-runtime>

Recommended reading:

<https://ampliance.com/blog/64labs-pwa-kit-easiest-route-to-headless-sfcc/>

<https://ampliance.com/blog/how-to-make-your-ecommerce-site-more-accessible/>

<https://ampliance.com/blog/improve-page-load-performance-with-salesforce-pwa-kit/>

[Headless CMS Buyers Guide](#)

[Digital Transformation](#)

[Agile Teams & Workflows](#)

Amplience and Salesforce have been partners for 10 years and have decades of commerce industry experts, working together to solve business challenges for over 80 retailers and brands.

Experience
FREEDOM



Amplience UK
6th Floor Tower House
10 Southampton Street
London, WC2E7HA, UK
+44 20 742 6990

Amplience USA
234 Fifth Avenue
New York, NY
10001, USA
+1 917 410 7189

contact@amplience.com
www.amplience.com