How to Leverage

MACH

to Improve Your Core Web Vitals Score

7%

1%

Layer0

uniform

Vue Storefront

Cloudinary

algolia

Layer0 (CDN/Hosting Platform)

HTTPS: A Secure Web is Here to Stay

On February 8, 2018, the Google Security Blog reported that A secure web is here to stay. Those were the days of strongly advocating that sites adopt HTTPS encryption before enforcing all HTTP sites to be marked as “not-secure” in Chrome with the release of Chrome 68. This announcement — at the time — marked the latest in a number of precursory articles about safety and security on the Web. Articles like Moving towards a more secure web, Here's to more HTTPS on the web!, and Say “yes” to HTTPS: Chrome secures the web, one site at a time marked a clear path for what will and has been done to keep the web more secure.

HTTPS alongside existing Search signals like mobile-friendliness, safe-browsing, and intrusive interstitial guidelines, meet with the newb Core Web Vitals (CWV) at the intersection of Page Experience, a set of signals that measure how users perceive the experience of interacting with a web page beyond its pure information value. Essentially, Page Experience is about user experience as they interact with web pages.

The Core Web Vitals unique set of measurable metric points that helps you understand how **fast**, **responsive**, or **visually stable** your web pages are, are great pointers to building a great developer experience on the web, and while they show you how your web application is being used by real users, they can also drastically affect if users will return to them, or if users will find them initially.

Similar to the HTTPS “not-secure” enforcement, the incoming CWV update to page experience is an impending enforcement to the adoption of Core Web Vitals. For what it is worth, this enforcement — as we've seen in the precautionary case of HTTPS — is punitive by SEO ranking — a differentiator between web pages that meet the minimum required Core Web Vitals metric score, and those that do not.

SEO ranking: The intersection between Core Web Vitals and User Experience

Until the advent of CWV, frontend performance metrics was a rabbit hole you were welcome to plunge and indulge yourself in, and one that was largely devoid of any real user experience metrics. This is largely dependent, as **Guillermo Rauch** puts it, on the fact that performance as it relates to user experience is technically very challenging to measure.

As web pages continuously get queried by users, and mutated by developers, the importance of CWV and its set of measurable metrics points will become even more salient in terms of user experience. Achieving a good CWV metrics points means your users can experience a fast loading, immediately interactive, and visually stable web page. This will help you achieve a higher SEO rank, delight your users, and invariably create an indirect improvement for a better web.

UX Metrics for a Better Web

Why being able to optimize your website from a UX metric standpoint is not only good for you, or your users, but also good for the web in general. It's a win for everyone.

User retention on a less performing website is low and this can and will affect the traffic to your website, which could be a real pain on sales and hurt your bottomline. However, as this affects you personally (directly, or indirectly), and as you try to optimize your web pages, keep in mind that you're also indirectly making the web better.

Something to ponder is that even if there is no performance penalty involved from Google, being able to measure your web pages from a UX standpoint can help you get meaningful datasets to improve its user experience. It is a win-win for you and your users, and generally for the web.

Improving website performance with Layer0

The optimizations and improvements you make towards CWV is punitive: It will affect your SEO ranks as well as determine your website's discoverability and retention. This will leave UX performance optimizations a never ending battle, one which CDNs play an important role.

For example, In CLS, where slow and large images are one of the causative elements, including predetermined sizes attributes on your <img/> element, or reserving the required space with something like CSS aspect ratio boxes can help. However, UX performance metrics is a death-by-a-thousand-cut that is typically hard to measure, and in reality can only be mitigated.

Ultimately your site will be deployed to a Content Delivery Network which is a geographically distributed network of servers that serves as a means of alleviating the performance bottlenecks of the Internet. CDNs are known to provide fast delivery of Internet content, increasing content availability and redundancy, reducing bandwidth costs, as well as improve security against DDoS, which all invariably improve CWV metrics. But certain optimizations techniques like caching and prefetching can be way out of their league, most especially with large and dynamic websites with thousands of pages.

The Layer0 CDN extends the capability of a traditional CDN and provides you with an infrastructure to run your big, dynamic website frontend and make it load in less than one second. This not only affects your CWV score by increasing your website load time, it also gives you a great developer experience while at it, something you will need as you continue to improve your websites and settle UX performance debt.

Measuring Core Web Vitals

The importance of measuring your site performance after specific metric optimization can not be overstretched. The official measurement tools are Google Search Console and PageSpeed Insights. But solely relying on these tools present a number of challenges:

- It can take days to weeks to see the effect that changes to your site have on Core Web Vitals.
- It is hard to diagnose Core Web Vitals by page type or URL.
- It is impossible to A/B test the impact of site optimizations on Core Web Vitals (to effectively A/B test performance optimizations you need both a RUM measurement tool and split testing at the edge, both of which Layer0 provides)
- It is impossible to get real-time measurement report from real-user monitoring (RUM)

Measuring your site's CWV with Layer0

Layer0's improvements of the official measurement tools can help you:

- See how changes to your site impact Core Web Vitals in real time
- Correlate web vitals to your application's routes
- Analyze score across a number of dimensions such as country, device, and connection type
- Identify which pages are most negatively impacting your search ranking.
- A/B test the impact of performance optimizations on Core Web Vitals through its Edge based split testing
- Triage the issues with Core Web Vitals and single-page applications (SPA) better with a new metric called Incremental Layout Shift (ILS) that helps you track and improve CLS for SPAs.

Vue Storefront (Frontend Framework)

Owner: Kaja Grzybowska

Speed is crucial for SEO, but Google wants you to do even better

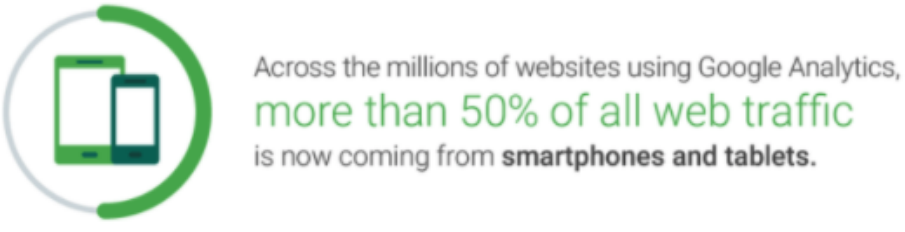
Kaja Grzybowska, Senior Content Writer at Vue Storefront, and Filip Sobol, Core Developer at Vue Storefront

Performance and mobile-friendliness are ranking factors in the Google Search algorithm for quite some time, and Google - updating Core Web Vitals - remains to be deadly serious when it comes to a general direction: top positions in search are for those who provide the best experiences both in terms of content and the form. How does it relate to the frontend? What sort of issues must be taken into account? Let's dive into details.

With announcing a new program, Web Vitals, Google's push to making websites user-friendly was said loud and clear. However, it was not a secret before that.

> *Optimizing for the quality of user experience is key to the long-term success of any site on the web.*
>
> — ***Google Blog***

Google's care of user experience is evident for years, and the subsequent updates only confirm the approach that points out SEO optimization shouldn't be limited to on-page activities. The more different devices play significant roles in driving web traffic, the more pressure has been put on technical-oriented aspects and UX. Google must address the specific needs of mobile users to prevent them from going elsewhere.



Think with Google

Source: Google Analytics Data, U.S., Q1 2016.

All of this, obviously, didn't start yesterday.

We already had a speed update that made page speed a ranking factor in mobile search, and earlier, in 2015, there was a "Mobile Friendly Update", nicknamed "Mobilegeddon" by SEO industries. This moniker turned out to be too dramatic because the changes that were supposed to bring a dramatic collapse in SERPs were more of a cultural shift indicated by Google than a real storm for site owners. However, the "Mobilegeddon" has had its teeth as, from that point, Google began to push its mobile-focus vision by continually introducing new updates, including the mobile-first index.

Regardless of Mobilegeddon, we also got RankBrain with which - according to Backlinko SEO specialist - UX started gaining importance. Now, it is one of the most crucial ranking factors that considers behavior metrics, such as the bounce rate, CTR, pages per session, and time spent on-site, and others. It shows Google whether the users are happy with what they get. When they are clicking, engaging, and spend significant amounts of time on site, Google assumes that both the content and the way it is delivered is satisfying; when users pop in for a second, then rapidly leave and never come back, it is a strong hint that they don't enjoy what they saw.

RankBrain is probably the oldest algorithm update proving that SEO and UX are inseparable. No wonder since both of them have the same goal: provide users with high-quality, trustworthy, and relevant results. To get the job done, web owners need to put themselves in the users' shoes, And that is it, no funny business. Black Hat SEO activities, such as stuffing mass-produced texts with keywords, are no longer an option as Google is continuously evolving, improving its methods of evaluating user satisfaction.

The Core Web Vitals are the next step of the progress.

To be continued... Core Web Vitals

Google announced Core Web Vitals in May 2020. But what are they? According to Google, it is "a set of real-world, user-centered metrics that quantify key aspects of the user experience." OK, Google, so... where is the difference from the previous UX-oriented updates?

Google still sees that - despite all its pushings - web owners struggle with delivering the satisfying UX, especially on mobile, where - let's say it out loud - the users' expectations are incredibly high and so hard to meet.

According to Think With Google, if your page doesn't load within 3 seconds, 53% of mobile website visitors will leave, but the average time to fully load a mobile landing page is 15 seconds.

Performance is not the only metric that matters, but it is a foundation of the rest. It affects bounce rates, engagement, time, conversion rates, and, ultimately, a business bottom line. It is a no-brainer, though, for quite some time, so why were the new metrics introduced? Here is why.

Google, so far, was relying on crawlers and evaluated speed in a very binary way. You could have been green or red, and - given the Lighthouse limitations - you could be green and turned red after switching the internet provider. But it is not about these issues. Core Web Vitals are focused not on speed itself but on its implications on user experience. In practice, Core Web Vitals try to find answers not for a question "How long it takes to get the site fully loaded, but "How long it takes for elements to become ready to use for customers."

To do that properly, Google identified three aspects:

- Largest Contentful Paint measures perceived load speed and marks the point in the page load timeline when the page's main content has likely loaded.
- First Input Delay measures responsiveness and quantifies the experience users feel when trying to first interact with the page.
- Cumulative Layout Shift measures visual stability and quantifies the amount of unexpected layout shift of visible page content.

Tomasz Rudzki, Head of R&D at ONELY explained it with an example:

The data displayed by PageSpeed Insights is divided into two parts:

Layer0's improvements of the official measurement tools can help you:

- Field data - usage statistics gathered from real users (Chrome users)
- Lab data - data measured by a single-test simulation.

What do these data tell us?

- VueJS.org is doing very well when it comes to CLS and FID.
- The website's loading speed needs improvement. PageSpeed Insights provides some suggestions on what could be improved. In the case of VueJS.org, the highest savings can be made by properly size images, removing unused JavaScript, and eliminating render-blocking resources.

Following the advice PageSpeed Insights gives you is a good first step.

Of course, there are more ways to measure your website's Core Web VitalsYou can also use the PageSpeed Insights API, query theCrUX databases in Big Query, or using a Chrome plugin. My colleague, Renata Gwizdak, explained it more thoroughly in her article, published very recently at Onely.com.

What will be the real impact of Core Web Vitals?

After the overblown Mobilegeddon, we are all a little bit skeptical about the real influence of the newest update, and this time, nobody expects a colossal earthquake. Does it mean that web owners can sleep well even if their metrics are low? By no means! Google doesn't want to shake up SERPs, yet, it does want to provide users with the best possible results, and the Core Web Vitals are crucial in describing the UX.

Cloudinary (Media Optimization)

Owner: Mickey Aharony

Optimizing for Google's LCP Metric

How a Highly Visual Yet Slow-Loading Page Could Cost You Visitors

Do you know that when it comes to a company's search-engine optimization (SEO) efforts, as many as 200 factors determine the search rankings? Over the years, e-businesses have documented and studied dynamics such as keyword density, URL structures, and link building with the objective of pleasing the almighty search algorithm—and be easily found by potential customers online.

For the most part, Google has operated with a lot of ambiguity around the search signals that matter the most, but that approach changed in May 2020 with the announcement of Core Web Vitals (CWVs). Never before had the search-engine giant been so forthcoming with its SEO guidance, which explains why the company gave websites a year's time to prepare.

Starting in June, Google's ranking criteria will prioritize "a set of metrics related to speed, responsiveness, and visual stability to **help site owners measure user experience on the web**." If a page is not optimized for viewing on various browsers and devices, its user experience suffers. A site that's weighed down by heavy rich-media files and that loads slowly also leads to a poor user experience. For many reasons, a poor UX creates a bad first impression, causing users to bounce off a page without engaging any further. That's why Google wants to ensure that the top-ranking sites in search results are the ones that "make the web more delightful," adding that "this will contribute to business success on the web as users grow more engaged and can transact with less friction."

Hence, even though those 200 factors are still at play, the three key CWV metrics that are in support of the best user experience are the primary focus of businesses' web strategies today.

It Pays to Become Familiar With Google's Three CWV Metrics

Google notes on its resource site for developers that each of the three CWVs "represents a distinct facet of the user experience, is measurable in the field, and reflects the real-world experience of a critical user-centric outcome." Here are the metrics:

- **Largest Contentful Paint (LCP)**, which depicts the loading speed or the length of time it takes for a webpage's above-the-fold content to fully render, i.e., LCP measures when the "largest content element in the viewport becomes visible." The content that affects this score is usually an eye-catching hero, a background image, or a carousel of photos. Ideally, the largest content element should load in less than 2.5 seconds.
- **Cumulative Layout Shift (CLS)**, which is the amount of unexpected shift that a user sees as content loads. The shift can be disorienting when you're reading a line of copy or engaging with a page element—only to have it disappear from view because something else on the page has caused a layout shift.
- **First Input Delay (FID)**, which is the length of time it takes for a page to become interactive. Note that visual media is not directly related to FID, which is nonetheless an important measurement of a page's perceived interactivity.

Tip: Check out the "Preparing for Google's Core Web Vitals" Cheat Sheet

Visuals Tell a Compelling Story But Are Problematic for LCP

Visuals are becoming increasingly popular for capturing the attention and consideration of site visitors. As bestselling author Ekaterina Walter says in her book The Power of Visual Storytelling, “Images act like shortcuts to the brain: we are visual creatures, and we are programmed to react to visuals more than to words.” However, because brands have been adding more and more rich media to their sites in an effort to engage users, websites are getting heavier, negating SEO efforts and resulting in low LCP scores.

How so? HTTP Archive found that highly visual webpages tend to exceed 7 MB in size. Furthermore, 75% of that page weight is made up of visuals like images, videos, animations, and GIFs. Absent effective optimization of their rich media, websites load sluggishly, leading to higher bounce rates, lower conversion, and yes, lower LCP scores. In fact, HTTP Archive also determined that in 2020, 47% of websites had an LCP score greater than the 2.5-second threshold set by Google.

Real business consequences abound for weighing websites down with unoptimized media. If the page-load time increases from one second to three seconds, the likelihood of a user bouncing increases by 32%. And if an e-commerce site meets all three metrics of the CWVs, consumers are 24% less likely to abandon the page. Remember, not only does Google give search-ranking preference to fast-loading sites with a high LCP score, studies have shown that on the first page of search results, “the first five organic results account for more than 67% of all the clicks.”

Ultimately, if your site fails to load fast, visitors would leave quickly—assuming that they can even find it in the first place.

Today's Web Performance Hinges on Media Optimization

Optimizing media is paramount for improving web performance and meeting Google's LCP metric. Following are three steps you can take:

- **Automate optimization and delivery.**
  By infusing quality-led automation into the workflow of image optimization, you can ensure that the correct file size is applied to transformations, such as cropping, reformatting, resizing, and creating asset variants. Automation can also better optimize images with modern formats, such as AVIF, JPEG XL, JPEG 200, and WebP, because compression saves bandwidth. You can also shorten the LCP load time by generating a faster server response and by leveraging lazy loading or low-quality image placeholders.
- **Deliver reliably at scale**
  Google recommends delivering media through a content delivery network (CDN) or, better yet, a multi-CDN setup, to enhance site reliability and scalability. Another way to reliably deliver optimized media is to cache page elements instead of loading them from their original sources. Doing all that would significantly raise your site's LCP score.
- **Measure performance**
  Prioritizing web performance involves committing yourself to a perpetual cycle of testing and iterations. Meeting the LCP threshold with optimized media is just the first step. You must also measure your site performance continuously and align with all the teams concerned to create a performance-driven culture.

If Google's Prioritizing User Experience, So Should You

Optimizing the media displayed on your site goes a long way in ensuring visitors will find it on a search and, once they're there, engage meaningfully.
To adopt automation capabilities and streamline your media-management workflow, consider a tool like

Cloudinary's Media Optimizer, which optimizes both the quality and file size of media assets for automatic delivery, all without coding or manual work on your part. Plus, Cloudinary's optimization technology can monitor media performance in a central hub for collaboration, ensuring continued progress of your team's web-performance enhancement and SEO efforts.

Uniform (Personalization)

Owner: Lars Birkholm Petersen

Personalization at the speed of MACH

Brands often report that performance and scalability are two of the main obstacles to widespread adoption of personalization on their sites. Adding personalization usually slows the site down. Personalization may also increase the load on the servers responsible for delivering the site, meaning that brands must deal with the complexity and expense of scaling their servers.

MACH architecture avoids these obstacles because performance and scalability are essential aspects of the architecture. Brands that adopt MACH architectures are able to deliver personalized experiences without the decreased performance or increased complexity and cost that plague traditional architectures.

Personalization matters

Brands often report that performance and scalability are two of the main obstacles to widespread adoption of The business case for personalization is compelling: companies using personalization to optimize digital experiences see, on average, a 19% increase in conversions. Web Content Management (WCM/CMS) and Digital Experience Platform (DXP) vendors have sold billions of dollars of software that promises to enable brands to capitalize on this. The number of brands who never implemented personalization is much greater than the number who have.

Implementing personalization is challenging. It requires a certain amount of marketing maturity to understand what kind of personalization is going to be successful, and to plan, execute and maintain a personalization strategy.

There are also technical challenges, the biggest being that personalization traditionally has a negative effect on site performance. As valuable as personalization is to increasing conversions, its value pales in comparison to the value of a fast site. Visitors will abandon slow sites before they are even exposed to personalized content. Adding

personalization to a site can actually be counterproductive when it affects site performance. Brands who enable personalization often end up disabling it for this reason.

Personalization with traditional architectures

Traditionally there have been two main types of personalization engines. The first is the kind offered by monolithic CMS and DXP vendors. This type of personalization involves personalization being tightly coupled with the page delivery process. An application server is responsible for generating and delivering personalized content. This is described as origin-based personalization.

The second kind is delivered via a JavaScript tag that is added to the website. This approach is commonly used by third-party personalization vendors. This is described as client-side or browser-based personalization.

While these approaches operate very differently, they both result in sites that are significantly slower than equivalent non-personalized sites. Origin-based personalization offered by the monoliths requires frequent calls to application servers that are relatively slow and hard to scale. This results in a reduced score for time-to-first-byte (TTFB), which is an important factor in determining how fast a site is.

Client-side personalization offered by third-party personalization vendors prevents pages from being rendered until the personalization script finishes running. Even the fastest CDN in the world won't help improve site performance in this case, because the performance bottleneck happens on the browser itself. The scripts are often bloated by numerous features most customers never use. The larger the script, the longer it takes for the browser to run the script, the longer the visitor is blocked from interacting with the page.

Nevertheless, these approaches worked well for the past 10+ years. Site performance was not a priority for most brands. Things are very different today. Google has signaled that it will be promoting Core Web Vitals to be a primary factor used to determine search rankings, with slow sites being penalized. Performance is no longer an option; it is an imperative.

Personalization with MACH architecture

Performance is a key design consideration for a MACH architecture. Decoupling functionality like personalization and page delivery from the CMS or DXP enables brands to use products that are able to meet today's performance and scalability requirements.

Decoupling personalization means a brand is able to leverage the system that can deliver personalization the fastest. In most cases this is neither the origin nor the browser, which are the personalization options available with traditional architectures. The system that can deliver the fastest personalization is the CDN because it is designed to be a global network dedicated to getting content to visitors as quickly as possible. This sort of decoupling is precisely what a MACH architecture involves.

The performance benefits of personalization with a MACH architecture over origin-based and browser-based personalization is significant. With the origin-based or browser-based approach, pages are typically delivered in 400-1,200ms. With a MACH architecture, pages are delivered in 50ms or less. This is an 8x-24x improvement, and this translates into a large impact on Core Web Vitals scores.

Core Web Vitals & personalization

Core Web Vitals offers a way to measure website performance. There are essentially two main operations that a browser performs that affect site performance: loading network resources like html files and images, and executing JavaScript in order to deliver dynamic functionality. Personalization involves both of these, and there are specific things to be aware of.

Largest Contentful Paint (performance)

The LCP time for your site should be less than 2.5 seconds. A top score requires the time to be less than 1.8 seconds. Minimizing LCP time can be accomplished by setting the following goals:

Goal 1: Optimize TTFB (Time to First Byte)

Tactic	Personalization considerations
Use a globally distributed CDN for site delivery.	Origin-based personalization limits the ability to leverage CDN caching because a call to the origin is required because personalization runs on the origin. If origin-based personalization is used, the latency to the data centers that host your site will almost always be noticeably greater than to a CDN.
Avoid origin calls.	Origin-based personalization requires origin calls. Decoupling page delivery from the origin and allowing the CDN to handle requests

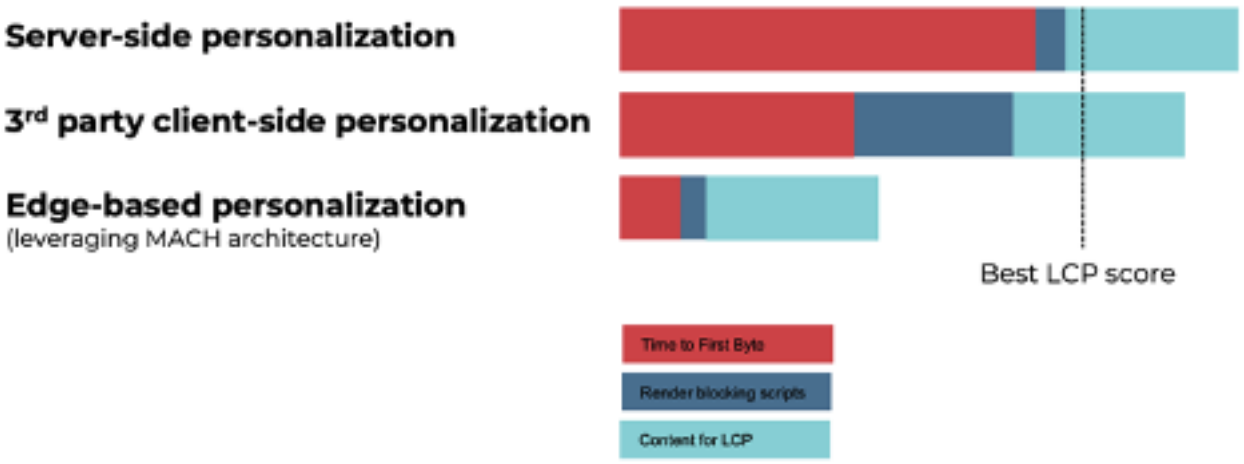
Goal 2: Reduce render blocking time

Tactic	Personalization considerations
Limit the use of 3rd party scripts.	3rd party scripts provide a lot of functionality that marketers believe they cannot live without. However, these scripts often come at a cost: decreasing site performance in multiple ways. Merely loading a 3rd party script takes time. In addition, depending on how the script is loaded and what the script does, it is possible the script will block rendering while it runs. This will increase your LCP time. Ironically, the Google Analytics script is a common culprit here. It may not be realistic to eliminate 3rd party scripts, be sure that the script provides enough value to justify the impact on site performance.

Goal 3: Reduce the amount of data that needs to be downloaded

Tactic	Personalization considerations
Use adaptive serving for larger assets like video and images.	You can use personalization, image optimization tools or the capabilities of a CDN for adaptive serving of larger assets involved in your LCP. For example, by optimizing media based on the visitor's device type, you can avoid loading large images on screens too small to display them, or videos on connections too slow to play them.

In addition to the goals above, the choice of personalization engine has a big effect on LCP. Below are the three different approaches illustrated in terms of impact on load time.



First Input Delay/Total Blocking Time (interactivity)

The FID time for your site should be 100 milliseconds or less. Minimizing FID time can be accomplished by setting the following goals:

Goal 1. Reduce JavaScript execution time

Tactic	Personalization considerations
Keep any logic outside of interactions.	Any logic that needs to execute from first interaction to load will impact your FID. Avoid having any logic that needs to execute for personalization to happen, based on interactions.

Goal 2: Reduce script sizes

Tactic	Personalization considerations
Be mindful of the size and number of scripts used in your site.	The larger the script, the longer it takes for the browser's main thread to run the script. The longer the main thread is blocked, the lower your FID score. Consider the size of the script and reduce any ongoing load.

Goal 3: Reduce data fetching

Tactic	Personalization considerations
Ensure the data can be loaded quickly.	Data fetching, especially when it involves origin calls, can block rendering. Use this capability sparingly. When it is required, make sure the endpoint that provides the data can do so as quickly as possible.

Cumulative Layout Shift (visual stability)

The CLS score for your site should be 0.1 or less. Minimizing CLS score can be accomplished by setting the following goals:

Goal 1. Minimize DOM changes after the DOM is loaded

Tactic	Personalization considerations
Determine your layout upfront and do not change it programmatically.	Browser-based personalization is typically implemented in the following way: ■ Page loads ■ DOM layout starts ■ Personalization scripts load ■ Personalization engine determines the appropriate content for the visitor ■ DOM is updated with personalized content This results in DOM changes after the DOM is loaded, which will affect your CLS score. Using edge-based personalization can avoid this.

Goal 2. Use content within the dimensions set by the DOM

Tactic	Personalization considerations
Ensure that the size of a personalized element remains constant.	For example, if a hero banner is personalized, make sure that the size of the banner remains the same regardless of whether default content or personalized content is used.

Goal 3. Do not show/hide content

Tactic	Personalization considerations
Use personalization to change content, not hide it.	In addition to being visually jarring, showing/hiding content results in a significant change to the page layout. This sort of change is exactly what CLS indicates.

Personalized MACH sites in action

Now that we have highlighted different tactics to consider for Core Web Vitals and personalization, let's look at how MACH-based edge-personalization, using a MACH certified vendor like Uniform, can help you with both performance and personalization at the speed of MACH.

- MACH architecture minimizes origin based services and moves the execution closer to the user, e.g., on a CDN, which is the best option for performance.
- Personalization is decoupled and deployed with your app, this removes any 3rd party or origin dependencies you need to load, which will have an impact on your performance.
- It benefits your users, as they can continue to use the tools they love: CMS for content creators, Commerce for merchandisers, Analytics for analysts, CDN for DevOps, JavaScript framework for the developers. No rigid paradigms, just flexibility and freedom to use the tools that are best for the job.

MACH architecture is composable, so personalization is designed to be compatible with the other components of your MACH Architecture. Personalization will co-exist and work with data and content across the different MACH technologies.

Algolia (Search)

The focus of this section is on how Google can find the pages of a website's internal search query. These pages are called search results page(s) (SERP). There are several kinds of SERPs:

- The most obvious is a page that lists the items a company offers, like products, films, news, applications, courses, medicines, and so on. Amazon's home page and their internal search engine search results pages are the most common examples of this.
- A company's landing page that displays its offerings, for example Netflix, where a page often shows a large number of movie and TV show selections.
- Pages devoted to a specific category of items, like Apple's pages for iPhones and MacBooks.

What these pages all have in common is that they use an internal search engine to display a company's offerings. Sometimes that search is executed by the user using a search bar, other times it happens behind the scenes, where the search is executed automatically by the website to display items to the user.

The import link between Google's SEO and a company's website search feature

When a company's search results page appears at the top of Google's first page, this creates the perception that the company is a leader on a given product or subject matter. Additionally, if the user clicks on the link, this indicates their clear intent to buy the product searched for or at least explore more information about it and related items.

High-ranking search results pages bring in significant amounts of product- and intent-driven organic traffic. Equally important: high clicks indicate to Google the company's expertise on the subject, improving the company's SEO.

Thus, not only is your search bar essential to help users search and browse your catalog, it's important for your company's SEO rankings.

How does Google find your search bar?

Until recently, good SEO ranking involved producing well-structured, meaningful content and using metadata like SEO-friendly keywords, titles, and page descriptions. It also involved avoiding "dark" SEO tricks that Google would detect and penalize.

With Core Web Vitals (CWV), Google has included an additional factor: the quality of a website's technology. In other words, Google's CWV favors websites that load quickly and whose interface is fast and easy to use. We describe here how CWV affects SERPs. All SERPs involve two technical concerns:

- **Crawling and finding**: Online businesses need to generate special SERP pages in the background for Google to crawl and search.
- **Core Web Vitals**: SERPs need to follow the three main criteria of Google's new CWV guidelines: load quickly (**LCP**), make the search bar available immediately, within 100 milliseconds (**FID**), and avoid disruptive ads and visual jumpiness (**CLS**). We'll discuss all of that below.

Crawling and finding

SERPs are dynamic: they change with every query. Unfortunately, Google usually ignores dynamic content. This means that Google won't execute a search on a company's website, it will only crawl fixed-content (static) pages. Thus, without your company doing something special, Google will never find your SERPs.

The solution is to use an automated process called page rendering that creates the pages in a way that allows Google to crawl and search them. All this means is that a company runs a process that executes a series of searches and saves each search result in a separate webpage. Google will crawl and search these artificially generated pages. Read more about this and other ways to optimize SEO traffic to an internal SERP.

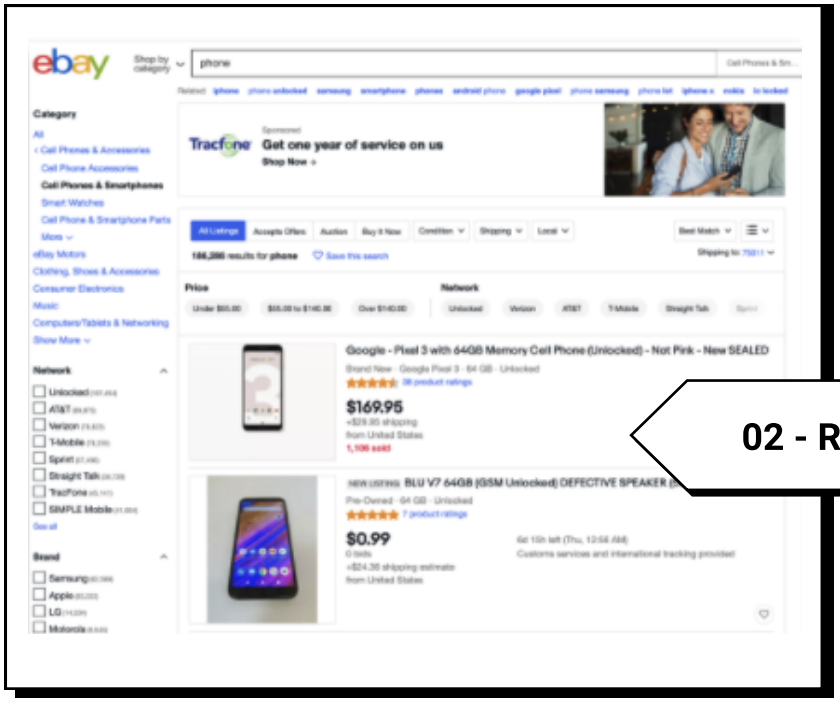
Core Web Vitals

CWV requires that SERPs be fast and easy to use, which is about technology and good user experience (UX). How does CWV use its 3 criteria to assess an SERP?

- **LCP**: Top results must load (ideally) within 3 seconds.
- **FID**: The search bar must be usable within 100 milliseconds.
- **CLS**: Many SERPs contain promotional content, images, and other visual/interactive features which, if not implemented well, can become disruptive.

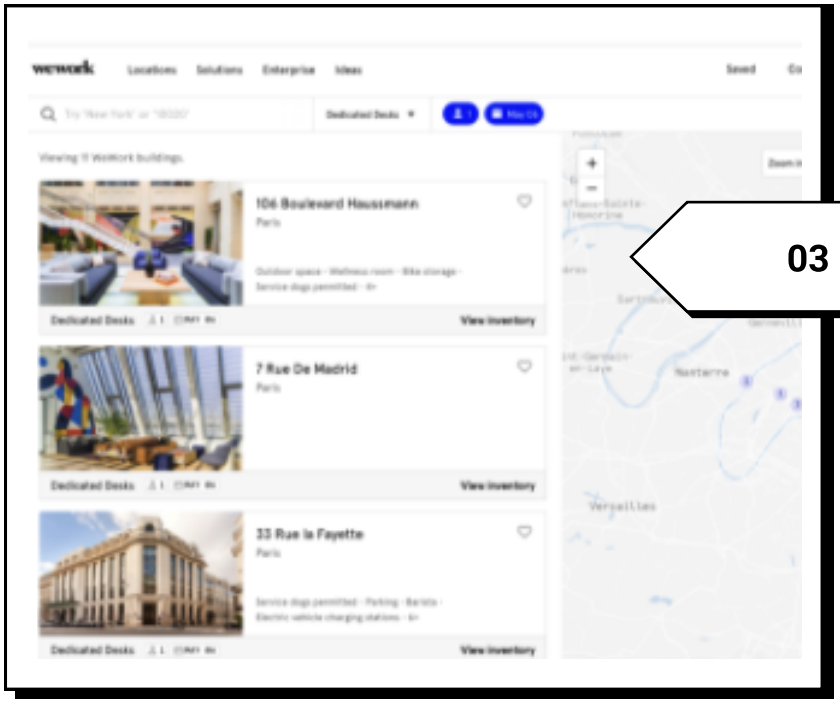
Let's see these criteria in action using some example SERPs:

The first example is a simple search page. It checks all the CWV boxes: fast, clean, and no disruptive elements. Perfect. But it's boring. Most online businesses need a richer search UI.



02 - Rich search Experience

The second example illustrates a richer, more visual search experience. Now, users have different options to search and browse, and businesses can promote and recommend products. However, its complexity can slow down the loading (LCP) and disrupt the user's interactivity (FID, CLS).



03 - Interactive search

The third example provides a powerful interactive experience. Users don't need to leave this page to make their choice. They can move the map, slide through images, compare office details, and type in new search criteria and filters. However, its interactive map and sliding images are exceptionally heavy to load (LCP), which can slow the usability of the search bar (FID) and also load the map and images at different times, thus disrupting the fluidity of the user experience (CLS).

Looking more closely at examples 2 and 3, we can pull out two essential features of a great search UI that can have a positive impact on the CWV if done well:

- Search-as-you-type, instant results — arguably the most important feature of modern search — means that users see new search results instantly on their screen with every keystroke. You see this powerful feature everywhere: as you type, you see Amazon's complex faceting, Netflix's numerous rows of images, and Spotify's diversity of information. CWV requires that all of this wonderful information be immediately displayed and usable, otherwise it's noise to both the user and Google's SEO engine.
- Merchandising and recommendations are critical parts of a successful search UX. However, if not carefully planned, they can easily fail the CLS test. Features like promotional banners, ads, product placement, related items, and sticky ads need to fit in seamlessly with the other elements on the page and not obstruct the user's primary intention to search.

The good news is that even the most complex search UI can achieve this.

How to build a fast search UI

Modern developers have at their disposal the most advanced technology and coding techniques to meet Google's CWV challenge. There are awesome APIs, frameworks, libraries, and languages to help them; there are also great developer tools to troubleshoot, and loads of blogs, communities, and forums to get help. Here is how this applies to website/app search.

The speed of the network (good for LCP)

When a user searches, the information they see displayed comes from the server. For search to be fast, (a) the engine needs to process in milliseconds and (b) the information it sends back needs to be already formatted, compressed, and organized for immediate display. The less the front-end code needs to do, the faster the UX.

Here is how we solved it at Algolia:

- Algolia built hosted search — their API takes only milliseconds to perform a search. Therefore, the browser has a full 3-5 seconds to load what the Algolia servers have sent.
- The information Algolia sends back is fully formatted in HTML and CSS. Therefore, the front-end code, especially when wrapped in Algolia's InstantSearch, does very little before displaying the search results. The top page in most implementations displays in less than 100 milliseconds

Preloading (good for LCP, FID)

A web page should not have to be completely loaded in order for a user to start searching or seeing important information. The front-end code should use several techniques to prioritize what parts of the screen load first, second, and last. In other words, the front-end code can load the search bar first (FID), and finish loading the top of the page (LCP) before loading the rest of the content.

Fixed grids (good for CLS)

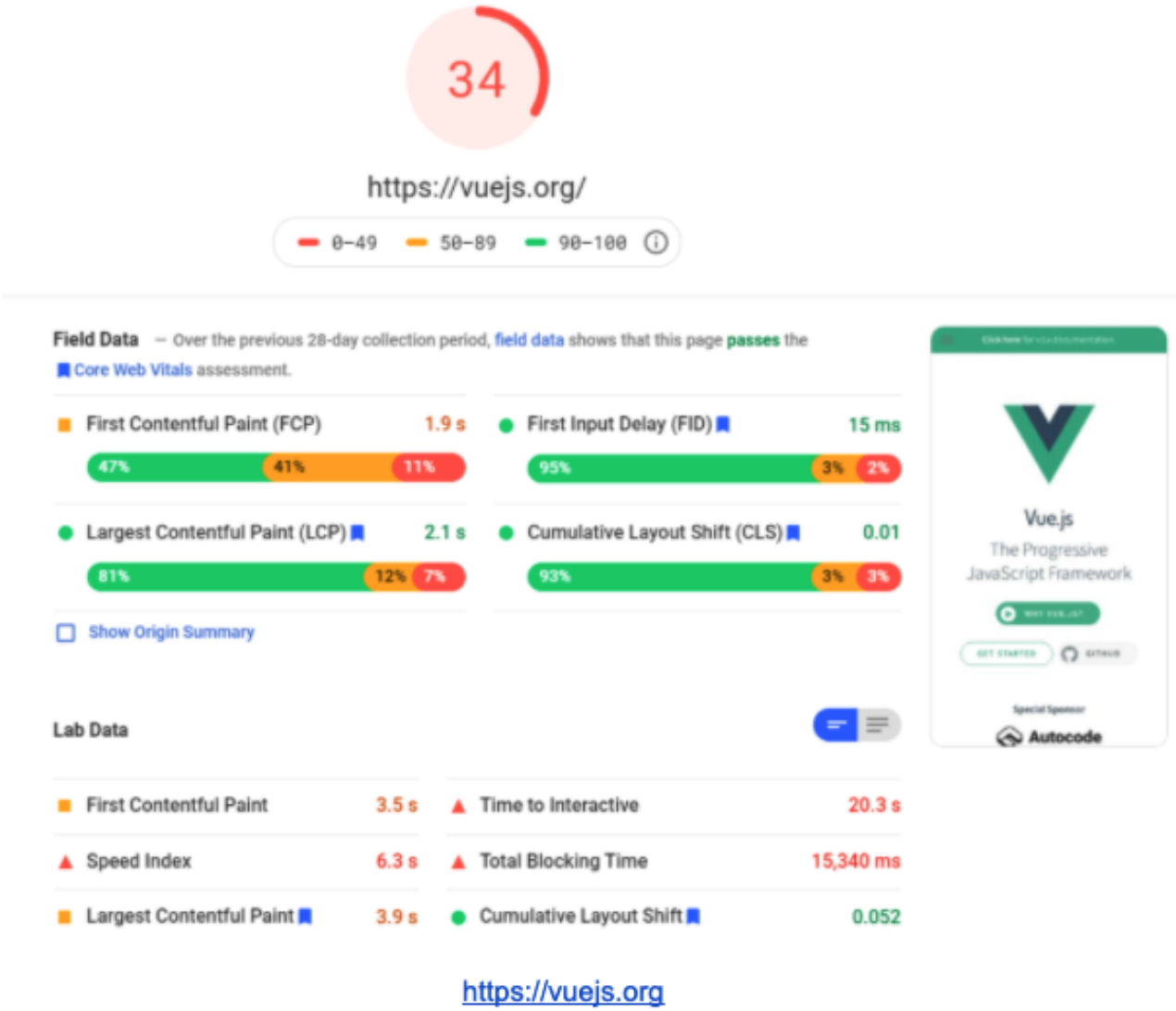
Finally, to manage this loading process, developers can use fixed grids to load different elements at different times. The front-end code can use one grid per element. That way, each element loads without resizing the screen or pushing other elements out of the way (CLS).

Optimized Images (good for LCP, CLS)

Use small and compressed images. Image processing on the web is a big subject. Great optimized methods exist. The main point here is to follow best practices.

One thing to note: CLS can conflict with LCP and FID. In other words, loading diverse elements at different times (to improve LCP and FID) can appear strange to the user (degrading CLS). The solution is to set up fixed grids and prioritize the most sensible loading order.

To conclude: Core Web Vitals emphasizes the need to focus on every millisecond in the user experience. Your website or app search is an important part of the equation.



https://vuejs.org

How to check Core Web Vitals

> *We're combining the signals derived from Core Web Vitals with our existing Search signals for page experience, including mobile-friendliness, safe-browsing, HTTPS-security, and intrusive interstitial guidelines, to provide a holistic picture of page experience. Because we continue to work on identifying and measuring aspects of page experience, we plan to incorporate more page experience signals on a yearly basis to both further align with evolving user expectations and increase the aspects of user experience that we can measure.* — ***Google Blog***

Google's statement sounds serious, given the unusually outspoken way of announcing it.

> *The importance of Core Web Vitals in Google's ranking algorithm is still not clear. What we do know is that Google rarely informs us about its plans to add new factors to the ranking algorithm. This change could be massive. Google doesn't only inform us of the upcoming update but also, due to the pandemic, it will be introduced with a half-year notice. According to Webmasters Google Blog, Google is planning to introduce Core Web Vitals in May 2021. What's very interesting is that Google "plan to test a visual indicator that highlights pages in search results that have great page experience."* — ***Tomasz Rudzki, Head of R&D at ONELY***

Core Web Vitals at the frontend

Vue Storefront has numerous features ready for this update: SSR, performance optimization and PWA. Individually, none of them can be seen as a remedy for SEO-related challenges, but together they can help face them.

Let's start with the latter. PWA, designed to create a site or web app that is always accessible, fast, and engaging, is definitely one - but not the only one - of the ways to meet Google requirements in terms of UX. What are its SEO advantages? There are plenty, by default, but there are also some concerns that need to be dispelled. Progressive Web Apps are crawler-friendly, indexable, and provide an astonishing and multiplatform user experience, but their foundation is Javascript, which causes some sort of controversy in the SEO world. Why? For years Google bots simply couldn't crawl

JavaScript code, and even though a lot has been improved in that terms still many SEO specialists remain distrustful in using JavaScript, recommending using it "carefully".

The size of JS scripts, code quality, and complexity can significantly affect battery life and parse time, especially on low-end devices. Longer JS execution times forced on the user's device delay the first paint of the "hero" content that the user sees on the screen. All of these negatively influence metrics such as Time to First Paint, and First Contentful Paint, which measure how fast content is served to users.

Due to their JS-reliant nature, PWA can cause some SEO challenges, however, the challenges concern Single Page Apps in general, and so SEO optimization of PWAs is possible when it follows the basic SEO practices, and PWAs meet some of these restrictions by default. They are fast, engaging, extremely lightweight, and reliable. However, their SEO-friendliness must be double-checked manually as the JS code quality is the most crucial factor in the last one aspects.

Basic technical SEO principles

- Every page should be available through a unique URL with no fragment identifiers (e.g., #).
- Canonical URL should be specified to describe the original source of content.
- In PWAMP configurations, the rel=" amphtml" tag should be used
- Cloaking is forbidden
- HTTPS is advisable strongly
- Redirections must be set correctly in case of any changes to the website
- Metadata should be established properly
- Performance must be as fast as possible

Nonetheless, PWA is not a cure-all, and Vue Storefront, a PWA frontend platform whose crucial promise is to make performance lightning-fast, must have taken it into account

Messy code will most likely result in poor performance of the application and adding

PWA functionality to the mix will not magically result in green scores in page tests. Clean code and "best programming practices" can, directly and indirectly, improve performance, but they are meant to make the code readable and easy to maintain in the first place. However, even if the well-written app is not performant at a given moment, it can be improved with much less time and effort than a messy one.

So… What architectural assumptions ensure the performance (and so UX) of websites based on Vue Storefront will meet all the requirements thrown by Google with Core Web Vitals update?

There are a lot of them, connected one with another.

Firstly, Vue Storefront is based on Vue.js, a modern JS framework, and - at some point - the modern JS frameworks come to the rescue. Most of them aim to be easy to use and performant out-of-the-box, which can reduce the risk of messy coding. Still, they can’t be considered a cure-all to the problems.

Server Side Rendering in Vue Storefront

In Vue Storefront, the first visit on the page is server-side rendered (SSR). This process provides many benefits but is computationally expensive and time-consuming. This might be a problem, especially on sites with a lot of traffic. However, when we look at the results of this process, it's apparent that most pages are identical, no matter which user requested them.

What if we could render such pages once and serve the result to all subsequent requests? This is exactly what the SSR cache does. Pages marked with special tags are rendered once and saved to a very performant in-memory database. This very significantly reduces the load on the server and the time required to serve the response.

Performance optimization in Vue Storefront

While there is still plenty of work ahead of us, we already implemented a lot of performance optimizations, starting from who serves our pages, through how requests are processed, and ending with how the application is loaded and becomes interactive. Let's go through the whole process.

Performance-wise it's important who serves your application. We recently started the migration of Vue Storefront Cloud to a new hosting and CDN provider Google Cloud, which among other benefits, reduced the response time compared to our previous provider.

When the page is requested, we use SSR cache to serve our pages instead of rendering them from scratch every time, which significantly reduces the load on the servers and response time. If the page is not served from cache, it has to be rendered on the server. Still, because we use functional Vue.js components wherever possible, rendering performance is also faster than regular components. We use purgeCSS to remove unused styles and reduce the size of the payload. To further improve the loading speed of the critical assets we use preloading and HTTP/2 server push. Finally, when the page is loaded on the user's device, we use lazy hydration and lazy loading to improve initial render time and save bandwidth.