

The Edge Computing Opportunity: It's Not What You Think

07/26/2020



Matthew Prince



Cloudflare Workers® is one of the largest, most widely used edge computing platforms. We [announced Cloudflare Workers](#) nearly three years ago and it's been generally available for the last two years. Over that time, we've seen hundreds of thousands of developers write tens of millions of lines of code that now run across Cloudflare's network.

Just last quarter, 20,000 developers deployed for the first time a new application using Cloudflare Workers. More than 10% of all requests flowing through our

network today use Cloudflare Workers. And, among our largest customers, approximately 20% are adopting Cloudflare Workers as part of their deployments. It's been incredible to watch the platform grow.

Over the course of the coming week, which we're calling Serverless Week, we're going to be announcing a series of enhancements to the Cloudflare Workers platform to allow you to build much more complicated applications, lower your serverless computing bills, make your applications even faster, and prove that the Workers platform is secure to its core.

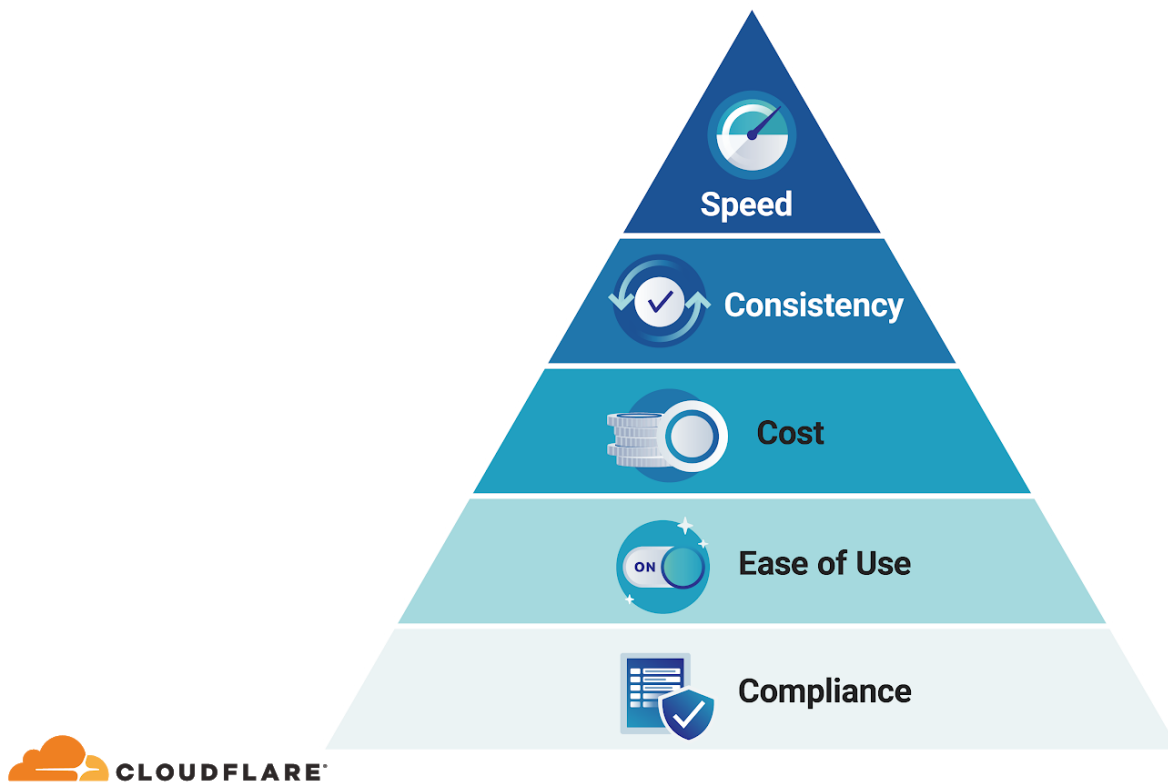
Matthew's Hierarchy of Developers' Needs

Before the week begins, I wanted to step back and talk a bit about what we've learned about edge computing over the course of the last three years. When we launched Cloudflare Workers we thought the killer feature was speed. Workers run across the Cloudflare network, closer to end users, so they inherently have faster response times than legacy, centralized serverless platforms.

However, we've learned by watching developers use Cloudflare Workers that there are a number of attributes to a development platform that are far more important than just speed. Speed is the icing on the cake, but it's not, for most applications, an initial requirement. Focusing only on it is a mistake that will doom edge computing platforms to obscurity.

Today, almost everyone who talks about the benefits of edge computing still focuses on speed. So did Akamai, which [launched their Java- and .NET-based EdgeComputing platform in 2002](#), only to shut it down in 2009 after failing to find enough customers where a bit less network latency alone justified the additional cost and complexity of running code at the edge. That's a cautionary tale much of the industry has forgotten.

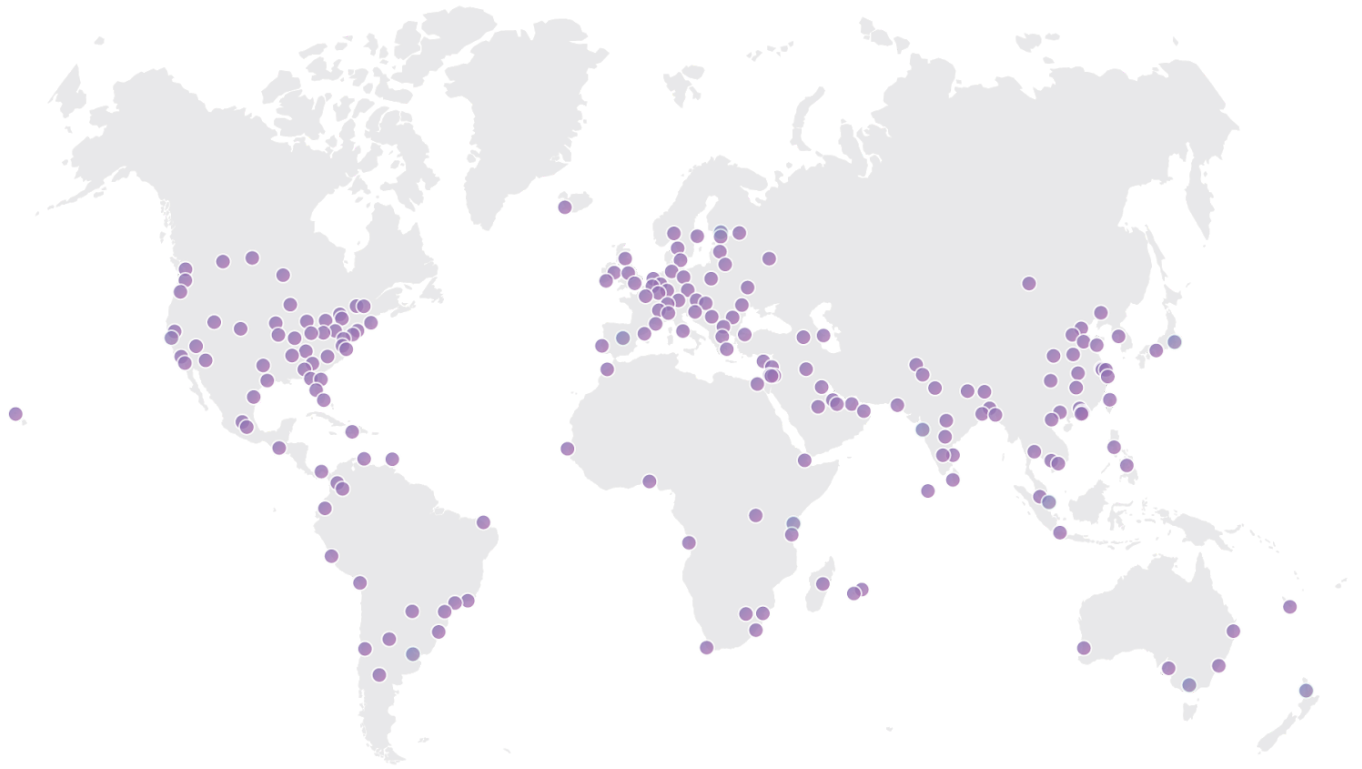
Today, I'm convinced that we were wrong when we launched Cloudflare Workers to think of speed as the killer feature of edge computing, and much of the rest of the industry's focus remains largely misplaced and risks missing a much larger opportunity.



I'd propose instead that what developers on any platform need, from least to most important, is actually: Speed < Consistency < Cost < Ease of Use < Compliance. Call it: Matthew's Hierarchy of Developers' Needs. While nearly everyone talking about edge computing has focused on speed, I'd argue that consistency, cost, ease of use, and especially compliance will ultimately be far more important. In fact, I predict the real killer feature of edge computing over the next three years will have to do with the relatively unsexy but foundationally important: regulatory compliance.

Speed As the Killer Feature?

Don't get me wrong, speed is great. Making an application fast is the self-actualization of a developer's experience. And we built Workers to be extremely fast. By moving computing workloads closer to where an application's users are we can, effectively, overcome the limitations imposed by the speed of light. [Cloudflare's network spans more than 200 cities](#) in more than 100 countries globally. We continue to build that network out to be a few milliseconds from every human on earth.



Since we're unlikely to make the speed of light any faster, the ability for any developer to write code and have it run across our entire network means we will always have a performance advantage over legacy, centralized computing solutions — even those that run in the "cloud." If you have to pick an "availability zone" for where to run your application, you're always going to be at a performance disadvantage to an application built on a platform like Workers that runs everywhere Cloudflare's network extends.

We believe Cloudflare Workers is already the fastest serverless platform and we'll continue to build out our network to ensure it remains so.

Speed Alone Is Niche

But let's be real a second. Only a limited set of applications are sensitive to network latency of a few hundred milliseconds. That's not to say under the model of a modern major serverless platform network latency doesn't matter, it's just that the applications that require that extra performance are niche.

Applications like credit card processing, ad delivery, gaming, and human-computer interactions can be very latency sensitive. Amazon's Alexa and Google Home, for instance, are better than many of their competitors in part because they

can take advantage of their corporate parents' edge networks to handle voice processing and therefore have lower latency and feel more responsive.

But after applications like that, it gets pretty "hand wavy." People who talk a lot about edge computing quickly start talking about IoT and driverless cars. Embarrassingly, when we first launched the Workers platform, I caught myself doing that all the time. Pro tip: when you're talking to an edge computing evangelist, you can win Buzzword BINGO every time so long as you ensure you have "IoT" and "driverless cars" on your BINGO card.



Donald Knuth, the famed Stanford Computer Science professor, (along with Tony Hoare, Edsgar Dijkstra, and many others) said something to the effect of "premature optimization is the root of all evil in programming." It shouldn't be surprising, then, that speed alone isn't a compelling enough reason for most developers to choose to use an edge computing platform. Doing so for most applications is premature optimization, aka. the "root of all evil." So what's more important than speed?

Consistency

While minimizing network latency is not enough to get most developers to move to a new platform, there is one source of latency that is endemic to nearly all serverless platforms: cold start time. A cold start is how long it takes to run an application the first time it executes on a particular server. Cold starts hurt because they make an application unpredictable and inconsistent. Sometimes a serverless application can be fast, if it's hitting a server where the code is hot, but other times it's slow when a container on a new server needs to be spun up and

code loaded from disk into memory. Unpredictability really hurts user experience; turns out humans love consistency more than they love speed.

The problem of cold starts is not unique to edge computing platforms. Inconsistency from cold starts are the bane of all serverless platforms. They are the tax you pay for not having to maintain and deploy your own instances. But edge computing platforms can actually make the cold start problem worse because they spread the computing workload across more servers in more locations. As a result, it's less likely that code will be "warm" on any particular server when a request arrives.

In other words, the more distributed a platform is, the more likely it is to have a cold start problem. And to work around that on most serverless platforms, developers have to create horrible hacks like performing idle requests to their own application from around the world so that their code stays hot. Adding insult to injury, the legacy cloud providers charge for those throw-away requests, or charge even more for their own hacky pre-warming/"reserved" solutions. It's absurd!

Zero Nanosecond Cold Starts

We knew cold starts were important, so, from the beginning, we worked to ensure that cold starts with Workers were under 5 milliseconds. That [compares extremely favorably](#) to other serverless platforms like AWS Lambda where cold starts can take as long as 5 seconds (1,000x slower than Workers).

But we wanted to do better. So, this week, we'll be announcing that Workers now supports zero *nanosecond* cold starts. Since, unless someone invents a time machine, it's impossible to take less time than that, we're confident that Workers now has the fastest cold starts of any serverless platform. This makes Cloudflare Workers the consistency king beating even the legacy, centralized serverless platforms.



But, again, in Matthew's Hierarchy of Developers' Needs, while consistency is more important than speed, there are other factors that are even more important than consistency when choosing a computing platform.

Cost

If you have to choose between a platform that is fast or one that is cheap, all else being equal, most developers will choose cheap. Developers are only willing to start paying extra for speed when they see user experience being harmed to the point of costing them even more than what a speed upgrade would cost. Until then, cheap beats fast.

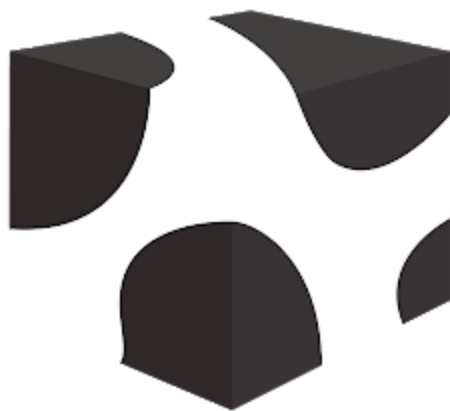
For the most part, edge computing platforms charge a premium for being faster. For instance, a request processed via AWS's Lambda@Edge costs approximately [three times more than a request processed via AWS Lambda](#); and basic Lambda is already outrageously expensive. That may seem to make sense in some ways — we all assume we need to pay more to be faster — but it's a pricing rationale that will always make edge computing a niche product servicing only those limited applications extremely sensitive to network latency.



But edge computing doesn't necessarily need to be more expensive. In fact, it can be cheaper. To understand, look at the cost of delivering services from the edge. If you're well-peered with local ISPs, like Cloudflare's network is, it can be less expensive to deliver bandwidth locally than it is to backhaul it around the world. There can be additional savings on the cost of power and colocation when running at the edge. Those are savings that we can use to help keep the price of the Cloudflare Workers platform low.

More Efficient Architecture Means Lower Costs

But the real cost win comes from a more efficient architecture. Back in the early-90s when I was a network administrator at my college, when we wanted to add a new application it meant ordering a new server. (We bought servers from Gateway; I thought their cardboard shipping boxes with the cow print were fun.) Then virtual machines (VMs) came along and you could run multiple applications on the same server. Effectively, the overhead per application went down because you needed fewer physical servers per application.



VMs gave rise to the first public clouds. Quickly, however, cloud providers looked for ways to reduce their overhead further. Containers provided a lighter weight

option to run multiple customers' workloads on the same machine, with dotCloud, which [went on to become Docker](#), leading the way and nearly everyone else eventually following. Again, the win with containers over VMs was reducing the overhead per application.

At Cloudflare, we knew history doesn't stop, so as we started building Workers we asked ourselves: what comes after containers? [The answer was isolates](#). Isolates are the sandboxing technology that your browser uses to keep processes separate. They are extremely fast and lightweight. It's why, when you visit a website, your browser can take code it's never seen before and execute it almost instantly.

By using isolates, rather than containers or virtual machines, we're able to keep computation overhead much lower than traditional serverless platforms. That allows us to much more efficiently handle compute workloads. We, in turn, can pass the savings from that efficiency on to our customers. We aim not to be less expensive than Lambda@Edge, it's to be less expensive than Lambda. Much less expensive.

From Limits to Limitless

Originally, we wanted Workers' pricing to be very simple and cost effective. Instead of charging for requests, CPU time, *and* bandwidth, like other serverless providers, we just charged per request. Simple. The tradeoff was that we were forced to impose maximum CPU, memory, and application size restrictions. What we've seen over the last three years is developers want to build more complicated, sophisticated applications using Workers — some of which pushed the boundaries of these limits. So this week we're taking the limits off.

Tomorrow we'll announce a new Workers option that allows you to run much more complicated computer workloads following the same pricing model that other serverless providers use, but at much more compelling rates. We'll continue to support our simplified option for users who can live within the previous limits. I'm especially excited to see how developers will be able to harness our technology to

build new applications, all at a lower cost and better performance than other legacy, centralized serverless platforms.

Faster, more consistent, and cheaper are great, but even together those alone aren't enough to win over most developers workloads. So what's more important than cost?

Ease of Use

Developers are lazy. I know firsthand because when I need to write a program I still reach for a trusty language I know like Perl (don't judge me) even if it's slower and more costly. I am not alone.

That's why with Cloudflare Workers we knew we needed to meet developers where they were already comfortable. That starts with supporting the languages that developers know and love. We've previously announced support for JavaScript, C, C++, Rust, Go, and [even COBOL](#). This week we'll be announcing support for Python, Scala, and Kotlin. We want to make sure you don't have to learn a new language and a new platform to get the benefits of Cloudflare Workers. (I'm still pushing for Perl support.)

Ease also means spending less time on things like technical operations. That's where serverless platforms have excelled. Being able to simply deploy code and allow the platform to scale up and down with load is magical. We've seen this with long-time users of Cloudflare Workers like [Discord](#), which has experienced several thousand percent usage growth over the last three years and the Workers platform has automatically scaled to meet their needs.



One challenge, however, of serverless platforms is debugging. Since, as a developer, it can be difficult to replicate the entire serverless platform locally, debugging your applications can be more difficult. This is compounded when deploying code to a platform [takes as long as 5 minutes, as it can with AWS's Lambda@Edge](#). If you're a developer, you know how painful waiting for your code to be deployed and testable can be. That's why it was critical to us that code changes be deployed globally to our entire network across more than 200 cities in less than 15 seconds.

The Bezos Rule

One of the most important decisions we made internally was to implement what we call the Bezos Rule. It requires two things: 1) that new features Cloudflare engineers build for ourselves must be built using Workers if at all possible; and 2) that any APIs or tools we build for ourselves must be made available to third party Workers developers.



Building a robust testing and debugging framework requires input from developers. Over the last three years, Cloudflare Workers' development toolkit has matured significantly based on feedback from the hundreds of thousands of developers using our platform, including our own team who have used Workers to quickly build innovative new features like [Cloudflare Access](#) and [Gateway](#). History has shown that the first, best customer of any platform needs to be the development team at the company building the platform.

[Wrangler](#), the command-line tool to provision, deploy, and debug your Cloudflare Workers, has developed into a robust developer experience based on extensive feedback from our own team. In addition to being the fastest, most consistent, and most affordable, I'm excited that given the momentum behind Cloudflare Workers it is quickly becoming the easiest serverless platform to use.

Generally, whatever platform is the easiest to use wins. But there is one thing that trumps even ease of use, and that, I predict, will prove to be edge computing's actual killer feature.

Compliance

If you're an individual developer, you may not think a lot about regulatory compliance. However, if you work as a developer at a big bank, or insurance company, or health care company, or any other company that touches sensitive data at meaningful scale, then you think about compliance a lot. You may want to use a particular platform because it's fast, consistent, cheap, and easy to use, but if your CIO, CTO, CISO, or General Counsel says "no" then it's back to the drawing board.

Most computing resources that run on cloud computing platforms, including serverless platforms, are created by developers who work at companies where compliance is a foundational requirement. And, up until to now, that's meant ensuring that platforms follow government regulations like GDPR (European privacy guidelines) or have [certifications](#) providing that they follow industry regulations such as PCI DSS (required if you accept credit cards), FedRamp (US government procurement requirements), ISO27001 (security risk management), SOC 1/2/3 (Security, Confidentiality, and Availability controls), and many more.

The Coming Era of Data Sovereignty

But there's a looming new risk of regulatory requirements that legacy cloud computing solutions are ill-equipped to satisfy. Increasingly, countries are pursuing regulations that ensure that their laws apply to their citizens' personal data. One

way to ensure you're in compliance with these laws is to store and process data of a country's citizens entirely within the country's borders.

The [EU](#), [India](#), and [Brazil](#) are all major markets that have or are currently considering regulations that assert legal sovereignty over their citizens' personal data. [China](#) has already imposed data localization regulations on many types of data. Whether you think that regulations that appear to require local data storage and processing are a good idea or not — and I personally think they are bad policies that will stifle innovation — my sense is the momentum behind them is significant enough that they are, at this point, likely inevitable. And, once a few countries begin requiring data sovereignty, it will be hard to stop nearly every country from following suit.



The risk is that such regulations could cost developers much of the efficiency gains serverless computing has achieved. If whole teams are required to coordinate between different cloud platforms in different jurisdictions to ensure compliance, it will be a nightmare.

Edge Computing to the Rescue

Herein lies the killer feature of edge computing. As governments impose new data sovereignty regulations, having a network that, with a single platform, spans every regulated geography will be critical for companies seeking to keep and process locally to comply with these new laws while remaining efficient.

While the regulations are just beginning to emerge, Cloudflare Workers already can run locally in more than 100 countries worldwide. That positions us to help developers meet data sovereignty requirements as they see fit. And we'll continue

to build tools that give developers options for satisfying their compliance obligations, without having to sacrifice the efficiencies the cloud has enabled.



The ultimate promise of serverless has been to allow any developer to say “I don’t care where my code runs, just make it scale.” Increasingly, another promise will need to be “I do care where my code runs, and I need more control to satisfy my compliance department.” Cloudflare Workers allows you the best of both worlds, with instant scaling, locations that span more than 100 countries around the world, and the granularity to choose exactly what you need.

Serverless Week

The best part? We're just getting started. Over the coming week, we'll discuss our vision for serverless and show you how we're building Cloudflare Workers into the fastest, most cost effective, secure, flexible, robust, easy to use serverless platform. We'll also highlight use cases from customers who are using Cloudflare Workers to build and scale applications in a way that was previously impossible. And we'll outline enhancements we've made to the platform to make it even better for developers going forward.

We've truly come a long way over the last three years of building out this platform, and I can't wait to see all the new applications developers build with Cloudflare Workers. You can get started for free right now by visiting: workers.cloudflare.com.

Welcome to
SERVERLESS WEEK



An unknown error occurred

We protect [entire corporate networks](#), help customers build [Internet-scale applications efficiently](#), accelerate any [website or Internet application](#), [ward off DDoS attacks](#), keep [hackers at bay](#), and can help you on [your journey to Zero Trust](#).

Visit [1.1.1.1](#) from any device to get started with our free app that makes your Internet faster and safer.

To learn more about our mission to help build a better Internet, [start here](#). If you're looking for a new career direction, check out [our open positions](#).

Women in Tech: You Belong Here

Tune In



[Serverless Week](#) [Serverless](#) [Cloudflare Workers](#) [Edge](#) [Product News](#)

Follow on Twitter

Matthew Prince | [@eastdakota](#)

Cloudflare | [Cloudflare](#)

RELATED POSTS

September 28, 2018 8:40PM

Birthday Week Wrap-Up: Every day is launch day at Cloudflare

Our customers are accustomed to us launching new services, features, and functionality at a feverish pace, but recently, we've been especially active. This week we celebrated our 8th Birthday Week by announcing new offerings that benefit our customers and the global Internet community....

By Jake Anderson

[Birthday Week](#), [Product News](#), [Registrar](#), [Cloudflare Workers](#), [Cloudflare Workers KV](#)

November 16, 2022 2:00PM

Mit Cloudflare D1 werden statische Websites jetzt dynamisch

In diesem Blogbeitrag wird erläutert, wie mit D1 einer statischen Blog-Website Kommentare hinzugefügt werden können. Dafür werden wir eine neue D1-Datenbank erstellen und eine einfache JSON-API entwickeln, mit der sich Kommentare verfassen und abrufen lassen...

By Kristian Freeman

November 18, 2022 9:13PM

ICYMI: Developer Week 2022 announcements

This week we made over 30 announcements, in case you missed any here's a quick round-up....

By Dawn Parzych

[Developer Week](#), [Developers](#), [Serverless](#), [Product News](#), [Supercloud](#)

July 30, 2021 3:00PM

The Cloudflare Startup Enterprise Plan: helping new startups bootstrap

To help early stage startups get going, Cloudflare is giving away one year of the Startup Enterprise plan to all early stage startups in participating accelerator programs....

By Jade Q. Wang

[Impact Week](#), [Product News](#), [Startup Enterprise Plan](#), [Cloudflare Workers](#), [Cloudflare Stream](#)

