ably

The State of Serverless WebSocket Infrastructure

Engineering leaders reveal the challenges and opportunities of building live and collaborative user experiences powered by WebSocket infrastructure, and whether they recommend buying a third-party serverless WebSocket PaaS or building a solution in-house.
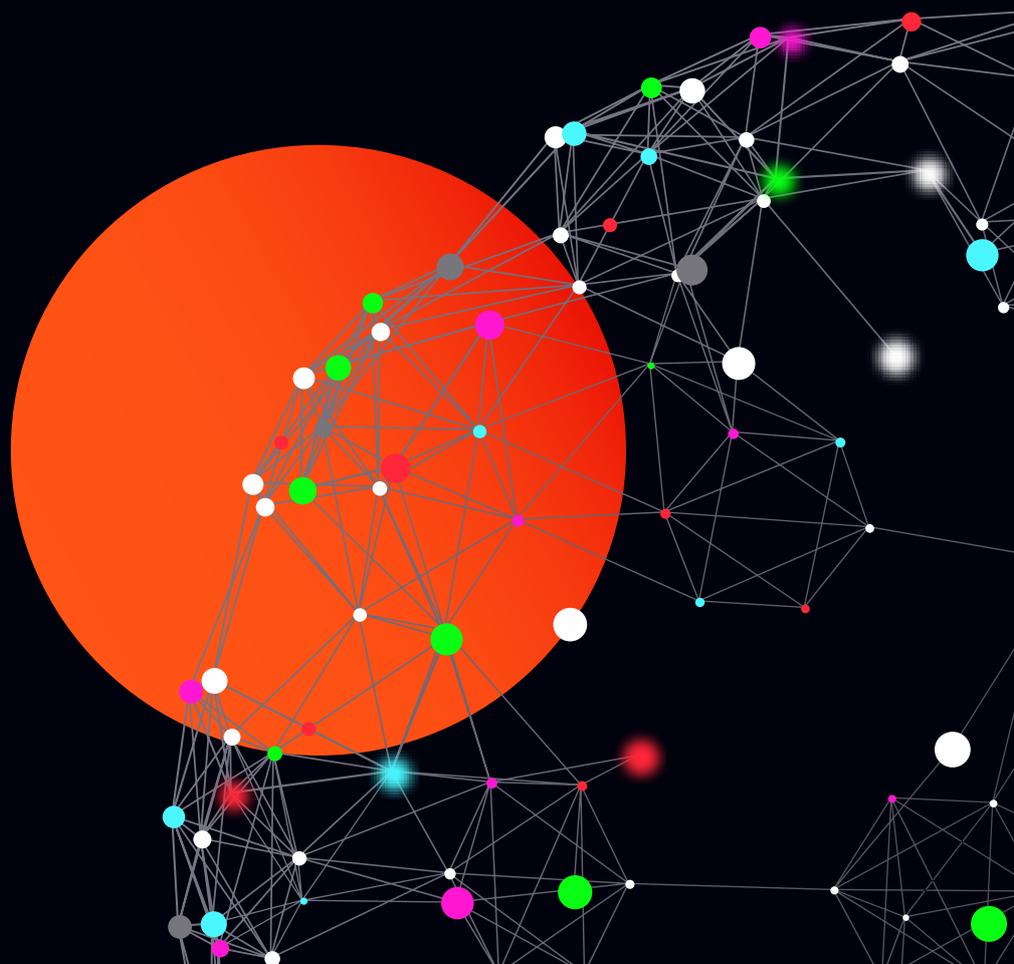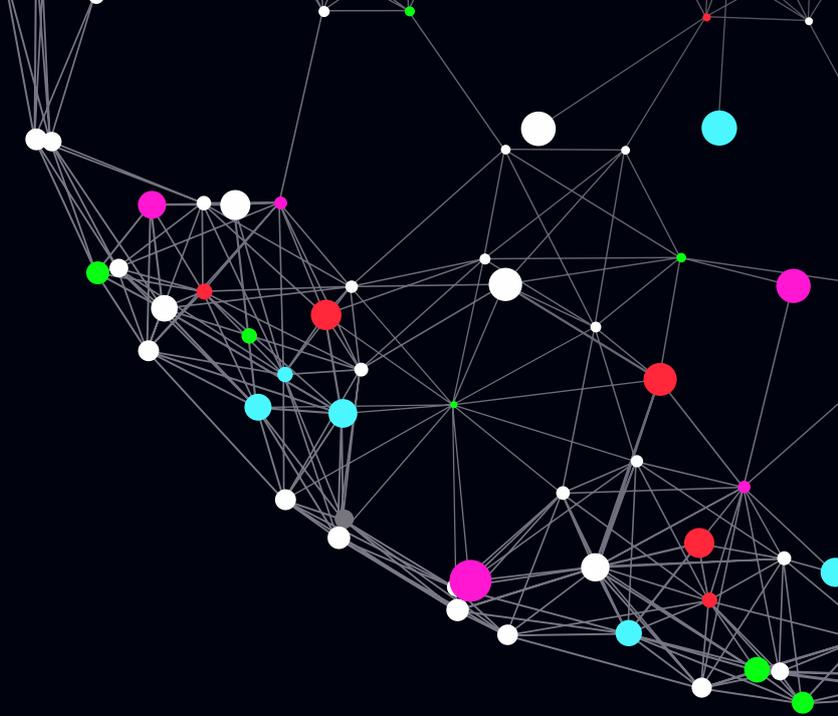
About The State of Serverless WebSocket Infrastructure Report

Ably surveyed over 500 engineering leaders about the challenges they faced when building their own realtime, event-driven WebSocket infrastructure to create live and collaborative experiences for end users, such as chat, order delivery tracking and document collaboration.

This report is the first of its kind about WebSocket-based infrastructure. It reveals and analyses the survey data, giving insights into the current state of WebSocket infrastructure and describing new trends within the landscape of live experiences.

As we will see, the complexity, costs and challenges of building and maintaining proprietary WebSocket infrastructure are compelling reasons for organizations to use a serverless WebSocket platform as a service (PaaS) provider instead.

Please get in touch if you'd like to share feedback for future reports or have any questions about serverless WebSocket infrastructure.

Contents

The State of Serverless WebSocket Infrastructure

Executive summary

As technology and customer experience rapidly advance to meet the dynamic requirements of a volatile socio-economic landscape, clients and customers have increasingly high expectations, seeking realtime information, interactions and solutions from businesses.

To meet this demand, the increasing capabilities of the digital devices we rely on at work and in our personal lives power new applications developed to connect and inform us. These include realtime financial and sports updates; live 1:1 and group chat; GPS location tracking; notifications; quizzes, polls and leaderboards; and collaborative charts and dashboards.

These live experiences are underpinned by realtime, event-driven infrastructures. To provide dependable and uninterrupted experiences that satisfy users, these infrastructures need to display robust properties, such as consistently low latencies, data integrity (ordering and guaranteed delivery), fault tolerance, availability and scalability.

WebSocket, a key realtime technology

Initially, the "realtime" web was difficult to achieve and slow to respond; it was delivered using techniques like AJAX and Comet (long) polling that were not truly optimized for realtime applications.

The emergence of WebSocket marked a turning point for web development. Designed to be event-driven and optimized for minimum overhead and low latency, a WebSocket connection enables bidirectional, full-duplex communication between client and server over a persistent, single-socket connection. The intent is to provide what is essentially an as-close-to-raw-as-possible TCP communication layer.

Standardized over a decade ago, WebSocket has since matured into one of the key technologies powering the modern, realtime web. WebSockets are a preferred choice for many organizations (such as Netflix, Figma, Slack, or HubSpot, to name just a few) seeking to build digital products that provide live and collaborative user experiences.

Serverless WebSocket in a nutshell

By combining the WebSocket technology with the serverless model, you gain the following key benefits:

- **No infrastructure to maintain.** Building your proprietary WebSocket infrastructure is time-consuming and resource-heavy. However, by using a serverless WebSocket provider, you offload the burden of managing demanding realtime infrastructure. Instead, you can focus on building and improving your core products.

- **Reduced operational costs.** Most serverless WebSocket providers offer a pay-for-use pricing model, so you're only charged for what you actually use. This is more cost-effective than renting or purchasing a fixed amount of server capacity in advance, which generally involves significant periods of underuse or idle time.

- **Scalability & availability.** Serverless WebSocket architectures are scalable by design. Apps built with a serverless infrastructure may experience high and fluctuating demand and need infrastructure that automatically scales up and down to handle an unpredictable and quickly changing number of concurrent WebSocket connections.

- **Reduced latency.** In the serverless model, the application is not hosted on an origin server. This means that, depending on the serverless WebSocket infrastructure provider used, you could have serverless applications running closer to end users, in multiple regions and edge locations worldwide. This helps improve performance and reduce latency.

A serverless WebSocket solution offers an infrastructure for event-driven messaging, for example using the pub/sub pattern. Clients connect seamlessly to send/receive low-latency messages, and act on realtime data.

Serverless WebSocket infrastructure forms the backbone of live experiences such as:

- Tracking a taxi driver's GPS location on a map as you wait for your ride home.

- Cross-device multiplayer gameplay features that enable a group of friends to play in the same game environment at the same time.

- Communicating with customer services or chatting to friends on a chat app.

- Receiving instant live updates about how your favorite sports teams are performing.

- Bi-directional chat, reactions, tipping and more as live streamers interact with millions of viewers on social platforms.

- The live quiz that gets feedback to ensure a class of students understood a lesson.

Serverless WebSocket infrastructure also enables synchronized and collaborative experiences such as:

- Simultaneously working on an interface design with a colleague.

- Collaborating in the same Google Doc with colleagues while you chat to them in an online call.

- Whiteboarding in Miro to brainstorm during a meeting.

Ably's serverless WebSocket platform

As a serverless WebSocket PaaS operating at the edge, Ably facilitates the exchange of events in realtime to drive live and collaborative experiences. It also provides APIs to deliver a set of features to enhance the experiences with device and user presence information, stream history and rewind, third-party integrations and user authentication.

Ably is underpinned by a globally-distributed network, consisting of 16 datacenters and 307 edge acceleration points-of-presence. The Ably network often handles millions of concurrent, long-running WebSocket and HTTP streaming connections. Today, Ably reaches more than 300 million devices across 80 countries each month and is on track to power live digital experiences for one billion devices per month by 2024.

> Today, Ably reaches more than 300 million devices across 80 countries each month and is on track to power live digital experiences for one billion devices per month by 2024.

The serverless WebSocket infrastructure report

In 2022, Ably conducted a survey of 500 engineering decision makers in the UK and USA. Our purpose was to discover their priorities and challenges when building realtime, WebSocket-based infrastructure in-house to support live and collaborative experiences for their end users. The survey uncovered and this report analyses:

- Cost to build WebSocket infrastructure.

- Time to build WebSocket infrastructure.

- Cost to expand and deliver realtime capabilities at scale.

- The benefits of purchasing a serverless WebSocket PaaS solution over building WebSocket infrastructure in-house.

Key findings

89%

of engineering decision makers say scalable WebSocket infrastructure gives a competitive advantage.

50%

cite reliability and 50% availability as the most important quality of good infrastructure.

10.2

person-months is the average time to build basic infrastructure, with limited scalability, in-house.

46%

of respondents identify escalating project costs as a major challenge during development.

41%

indicate that missed deadlines and extended timelines are significant problems of self-build.
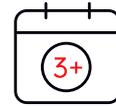
69%

of self-build projects lasted more than 3 months.

65%

of DIY solutions had an outage or significant downtime in the last 12-18 months.

38%

of those surveyed acknowledged that their final solution was too complex to use or update.

60%

who switched to PaaS said improving live UX with stable infrastructure was the top reason.

56%

ranked risk reduction as the second most compelling reason to swap to PaaS.

2/3

of respondents see the challenges of self-build slowing down future deliveries.

55%

said the third critical reason to move to PaaS was to redeploy engineers into core product work.

WebSocket infrastructure trends analysis

In our analysis of the results of the WebSocket infrastructure survey we uncovered several significant trends:

- Reliability and availability are critical factors for live and collaborative experiences underpinned by realtime data.

- Building WebSocket infrastructure takes a significant amount of time and resources - 10.2 person-months on average.

- Firms that underestimated the time and resources required to build WebSocket infrastructure were more likely to experience outages.

- Organizations are moving to serverless WebSocket PaaS providers, in favor of building WebSocket infrastructure themselves, to deliver a better user experience.

- Ensuring continuity of service will be a primary challenge over the next few years.

- Challenges with WebSocket infrastructure are expected to slow down the delivery of live and collaborative online experiences.

- WebSocket infrastructure issues will have a negative impact on product delivery.

Report conclusions

The three options open to organizations that want to utilize live experiences as a competitive differentiator are self-build, managed service provider and PaaS. All have potential benefits and drawbacks that businesses need to consider before making their choice. The survey demonstrated that the vast majority of those that chose to build their WebSocket infrastructure in-house ran into cost, time and functionality challenges.

These are the top three reasons engineers would switch or have switched to a serverless WebSocket platform as a service provider to manage their live and collaborative applications:

1. Improve the existing live user experience with a more stable and dependable infrastructure.

2. Redeploy engineering bandwidth away from infrastructure and into core products to work and move faster.

3. Reduce the risk of delivering new user experiences that require reliable realtime capabilities.

In the following sections, the report will take a deeper dive into the opportunities and challenges of delivering live experiences and explore what the survey statistics uncover about the state of (serverless) WebSocket infrastructure.

Opportunities and challenges of delivering live experiences

In our "always on" and increasingly fast paced world, corporate clients and consumers expect live experiences and immediate feedback from the businesses that serve them.

In a typical day they might attend a live, virtual corporate event with polls and a Q&A; collaborate synchronously on work documents; check their personal online banking; and monitor the progress of their online food order. These customers expect live, personalized interactions with bi-directional communication or an instantaneous response. They become loyal to the businesses that provide the fastest and most reliable realtime features like 1:1 and group live chat; GPS location tracking, and signaling for VoIP and video streams. The realtime data that underpins these experiences provides potentially lucrative opportunities for organizations, but it also ramps up the technical challenges, particularly for those that choose to develop the infrastructure in-house, as the survey and the following sections of the report reveal.

> The realtime data that underpins live experiences provides potentially lucrative opportunities for organizations, but it also ramps up the technical challenges, particularly for those that choose to develop the infrastructure in-house.

The move to an event-driven system to handle realtime data causes a significant increase in complexity. Without a strong understanding of what is required to create a fit-for-purpose realtime capability, numerous reliability, scalability and performance issues can emerge.

£5.00

Realtime dashboards

Realtime alerts
and notifications
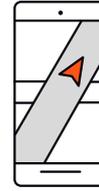
Realtime asset tracking
& live transit updates

Gamification &
live reactions

**Live experiences
powered by
serverless WebSocket
PaaS providers**

↑ USD	0.932	1.029
↓ EUR	2.068	1.888
↓ GBP	2.122	1.923
↑ JPY	0.798	1.095

Realtime currency
updates
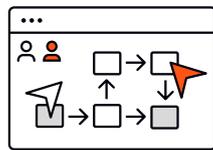
Live collaborative
editing
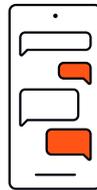
Chat

Live sports
updates
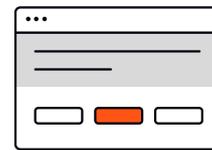
Live polls, quizzes,
and Q&A's

Many businesses don't realize there are easy to integrate platform as a service solutions available to power live and collaborative applications. Others mistakenly believe they can only get the flexibility and customization they require by building their own WebSocket infrastructure in-house. However, by offloading this type of infrastructure project to a serverless WebSocket PaaS provider, organizations can save time and money; eliminate development complexity; and benefit from global low latencies and immediate scale. In turn, engineers are freed up to focus on their company's core product, enabling them to innovate, stay relevant to customers, and gain a competitive advantage.

Underestimating the time and resources required to build WebSocket infrastructure can impact the performance of your system and, ultimately, your competitiveness in the marketplace. Conversely, if organizations use appropriate infrastructure, they can gain an advantage.

Knowing what it takes to deliver a live experience and what products and technologies to use can be extremely challenging. We have created this report to help organizations better understand the landscape of WebSocket infrastructure trends and challenges.

Critical success factors for WebSocket infrastructure

Infrastructure is key to delivering premium digital experiences, but what is the key to success? We asked engineers what they consider to be the most critical factors for WebSocket infrastructure.

Reliability and availability

It is critical for the WebSocket infrastructure that supports realtime data to be reliable. Outages lead to poor customer service and lost revenue. With 50% of the vote each, high availability of service and reliability were deemed to be the most critical factors for a fit for purpose WebSocket infrastructure. The criticality of WebSocket infrastructure is that it underpins the services users depend on. An outage of those services can completely undermine the applications with which they are integrated, leading to a bad user experience, low adoption and customer churn.

> "
>
> In a high-stake, highly competitive industry, we need infrastructure to transport live data to end-users via a complex, multi-hop process. We operate in a world where there's zero margin for error.
>
> **Joe Antonio**
> CIO, VITAC
>
> **VITAC**

Most critical factors for WebSocket (realtime) infrastructure

	Rank 1	Rank 2
High availability of service	29%	21%
Reliability of infrastructure	25%	25%
Integrity of data	20%	24%
Elastic scalability	15%	16%
Predictably low latencies	11%	14%

● Rank 1   ● Rank 2
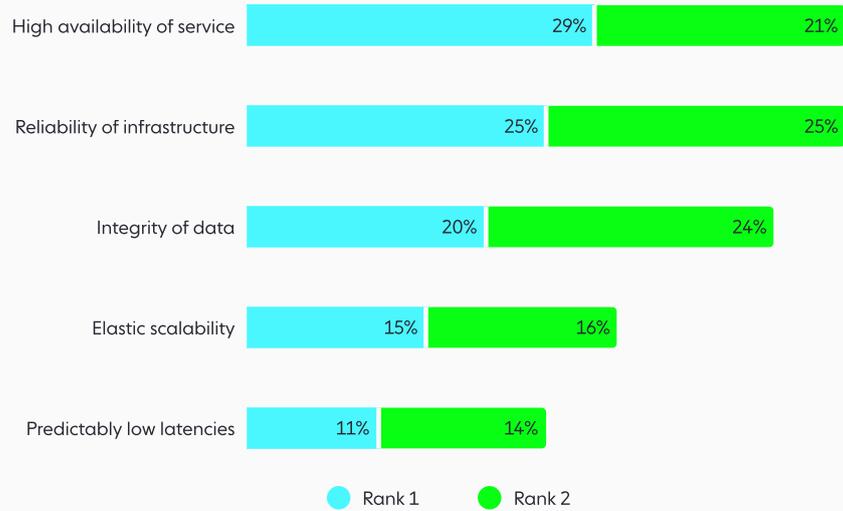
Outages over the past year and a half were not uncommon

In spite of respondents identifying reliability and availability as critical factors, almost two-thirds (65%) have experienced outages in the past 12-18 months. This disconnect between the importance of reliability and an organization's ability to deliver it exposes a growing challenge in the market.

Has your self-built WebSocket (realtime) infrastructure experienced an outage or significant downtime in the last 12-18 months?

35%

65%

● Yes
● No
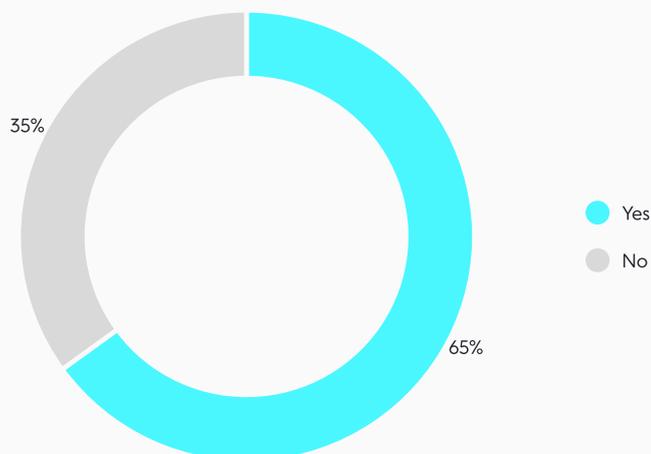
What does it take to build and maintain WebSocket infrastructure?

The challenges of realtime data services require complex architectures and infrastructures that need to be reliable and available. Organizations should consider offloading the capability and only commit to creating WebSocket infrastructure from scratch when fully aware of the complexity.

WebSocket infrastructure requires substantial resources

Building WebSocket infrastructure requires a significant amount of time and resources. While costs ran high, the vast majority of respondents reported costs that were relatively consistent throughout development. The survey data shows that 21% of projects lasted more than six months and nearly 93% of projects had 4-10+ engineers working on them.

> "
>
> We spoke to engineers at LinkedIn, Slack, and Box who had already built realtime infrastructure themselves - they told us it would take non-trivial upfront engineering with significant operating costs to build and operate this in-house.
>
> **Pato Echagüe**
> CTO, Split Software
>
> ◆ split

The survey data shows that nearly 70% of WebSocket infrastructure projects lasted more than three months and over 90% of projects had 4+ engineers working on them.

Time required to build first version of WebSocket (realtime) infrastructure

- 8%
- 13%
- 31%
- 48%

Legend:
- Less than 3 months
- 3-6 months
- 7-12 months
- More than 12 months
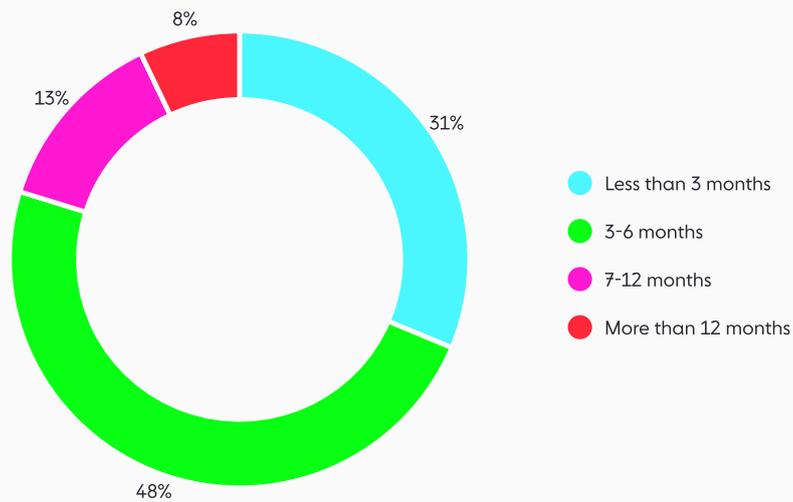
Number of engineers working on WebSocket (realtime) infrastructure project

- 9%
- 7%
- 39%
- 45%

Legend:
- 1-3
- 4-6
- 7-9
- 10+
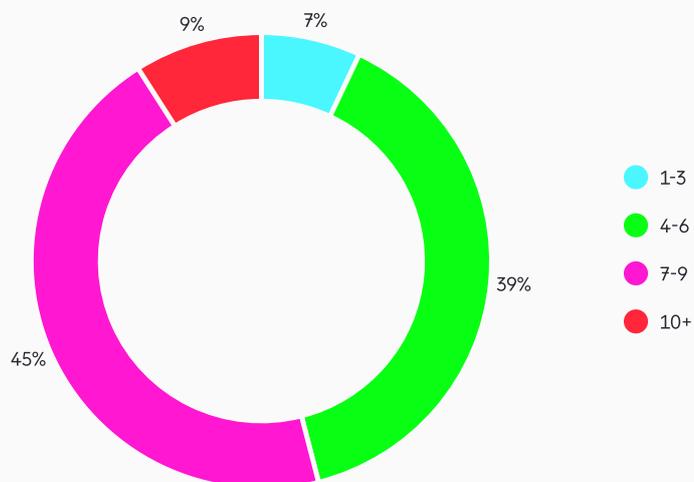
Person-months to build WebSocket (realtime) infrastructure

- Less than 6 person-months
- 6-12 person-months
- 13-20 person-months
- 21-26 person-months
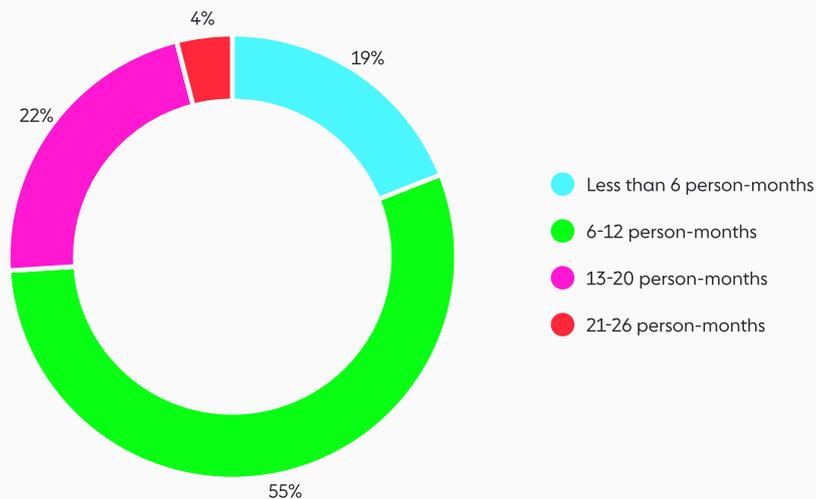
Detailed analysis provides further insight into how operations are balancing time and resources. On average, we discovered that 93% of projects required at least 4 engineers to build WebSocket infrastructure in-house. In addition, 69% of projects took over 3 months to build and 20% took over 6 months to build. For 4% of projects, more than 10 engineers were required to keep the timeline under 3 months and for 5%, more than 10 engineers were required to keep the project between 3-6 months in duration.

WebSocket infrastructure is costly to maintain

The infrastructure to support live and collaborative experiences can be costly to build and very expensive to maintain. Half of all self-built realtime data platforms require $100K-$200K a year in upkeep, according to survey respondents.

> "
>
> For any leader seriously considering building a product reliant on WebSocket infrastructure, it quickly becomes apparent that the human and infrastructure costs of maintaining it are going to be top of mind.
>
> **Tejas Dinkar**
> VP of Engineering, Quintype

Half of all self-built realtime data platforms require $100K-$200K a year in upkeep, according to survey respondents.

Maintaining complex architecture is expensive and scaling these systems also requires additional investment.

Annual cost to maintain WebSocket (realtime) infrastructure

- Up to $50k
- $50k - $100k
- $100k - $200k
- $200k+
- Don't know - it's part of our core infrastructure
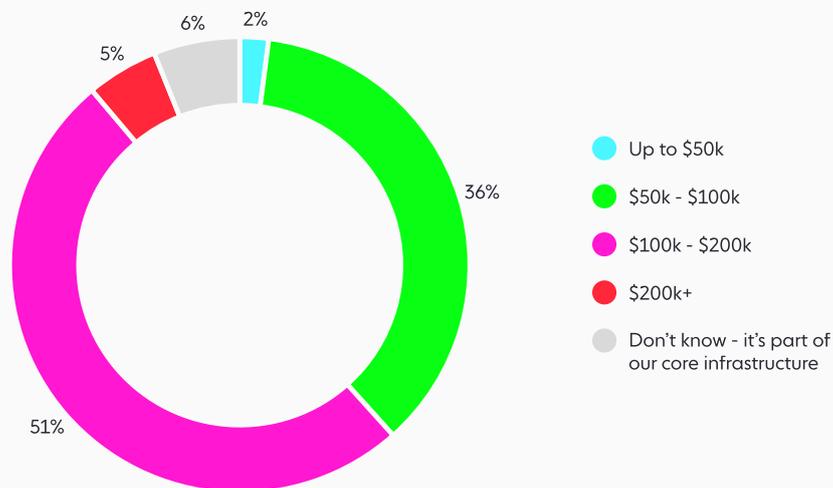
Don't forget opportunity costs

Not only does the infrastructure require significant resources to build and maintain, but it can drain resources from other areas of your business. The results of the study show that 69% of WebSocket infrastructure projects take over 3 months to complete and on average projects require 10.2 person-months. Using a serverless WebSocket PaaS provider means the infrastructure that enables realtime data delivery is taken care of, so you can continue to put resources behind developing your core product.

Estimating resource requirements proves difficult

Planners tend to underestimate what is required to build reliable WebSocket infrastructure to support realtime data; they not only miss deadlines, but also require extra resources. Of the respondents, 52% said projects required more engineers while 53% required more time than originally anticipated.

Additional engineers required to complete realtime data project

6%   2%

16%

48%

28%

- None
- 1 engineer
- 2-3 engineers
- 4-6 engineers
- 7-9 engineers
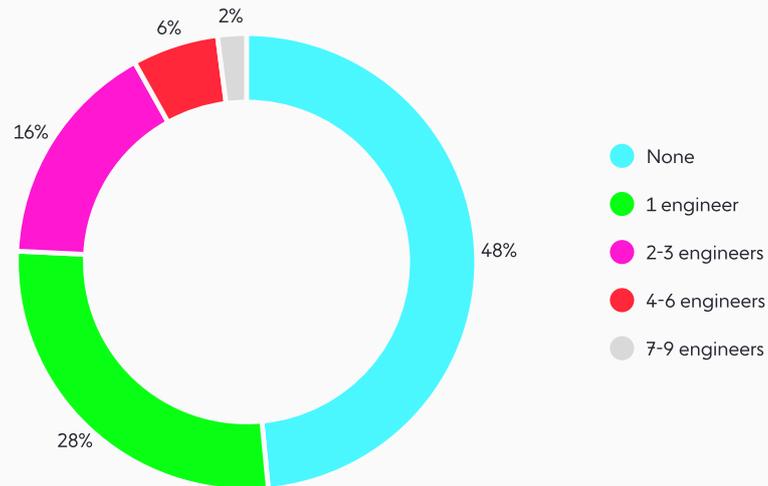
Extra time required to complete realtime data project

6%   2%

16%

47%

29%

- It took about the time we expected
- Up to 3 months longer
- 3-6 months more
- 6-12 months more
- We spent less time than expected
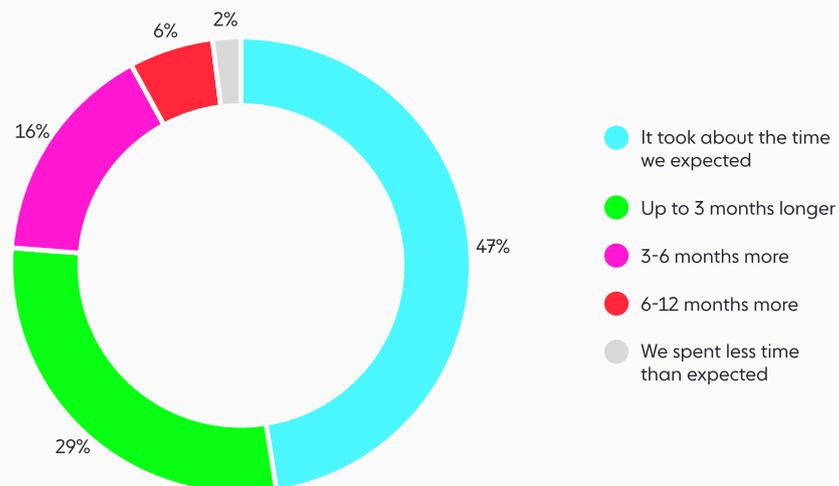
Only 47% of planners estimated the appropriate number of resources and were able to stick to their timeline. An experienced 2% of planners were able to spend less time on their projects without adding any engineers. The remaining 51% of miscalculated projects includes 16% of projects that required at least an additional 3-6 more months and 2-3 more engineers to complete.

KEY FINDING

53% of those surveyed said it took longer to build realtime WebSocket infrastructure in-house than they had orginally anticipated.

High costs and missed deadlines are clearly problematic: 46% of respondents reported the cost of the overall project as a major challenge and 41% reported missed deadlines and extended timelines. Only 9% of respondents reported no problems in building WebSocket infrastructure to enable live experiences.

Main challenges faced in development of WebSocket (realtime) infrastructure
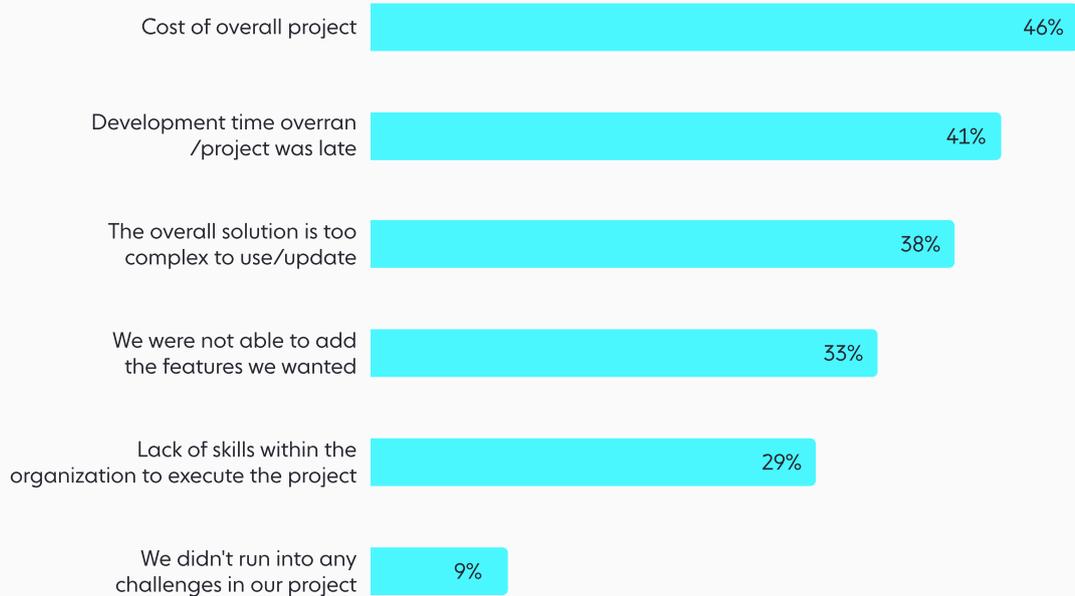
Challenge	Percentage
Cost of overall project	46%
Development time overran /project was late	41%
The overall solution is too complex to use/update	38%
We were not able to add the features we wanted	33%
Lack of skills within the organization to execute the project	29%
We didn't run into any challenges in our project	9%

Note: respondents were given the option to choose more than one answer

What tools and products are engineers using to build WebSocket infrastructure for live and collaborative experiences?

Developing WebSocket infrastructure is complex and engineers are using a variety of products and services. We asked respondents about 30 specific technology products and services. The average number of products used lies between four and five.

JavaScript is most popular and Java is an important skill in WebSocket infrastructure development

JavaScript is the most popular programming language in the world, so it is not surprising that 61% of engineers are using it when building apps to deliver realtime data.

Python has moved into the second most popular language globally, but engineers creating WebSocket infrastructure are more likely to be using Java than Python.

47% of respondents are using Java for realtime data delivery, which performs well in large, complex and dynamic applications.

Programming languages used when building WebSocket (realtime) infrastructure

Language	Percentage
JavaScript	61%
Java	47%
Phython	38%
PHP	27%
C/C++	27%
. NET/C#	25%
Ruby/Rails	22%
Node.js	17%
Go	9%
Rust	8%
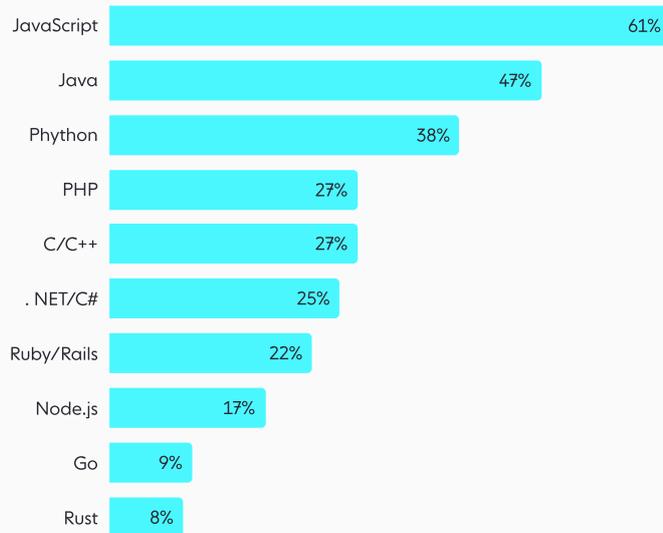
Note: respondents were given the option to choose more than one answer

Ecosystems

Survey respondents were asked to name their preferred cloud platforms and WebSocket technology providers, with many selecting more than one option. AWS was the most popular, with 56% using it.

Microsoft Azure typically competes well with AWS but, in this survey, only 24% of respondents were using it compared to the 56% using AWS.

Most popular ecosystems among respondents

Ecosystem	Percentage
AWS	56%
Google Cloud	41%
Azure	24%
Laravel	9%
Rails	8%
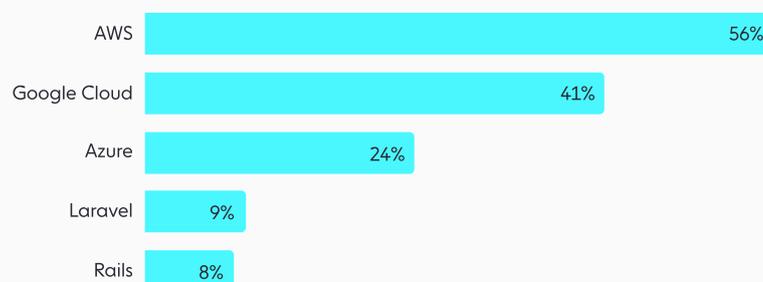
API gateways

Again, AWS leads as the most popular API gateway. WSO2's Enterprise Integrator is also very popular and provides an alternative outside the AWS ecosystem.

Top 3 most popular API gateways among respondents

Amazon API Gateway	20%
WSO2 Enterprise Integrator	16%
MuleSoft Anypoint	10%
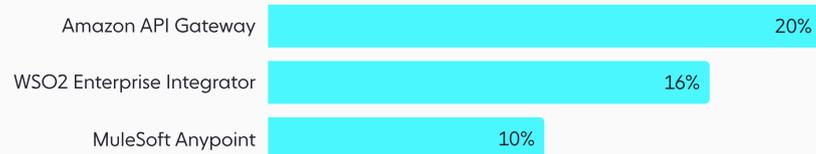
WebSocket frameworks/plugins

Django Channels is a popular WebSocket framework, almost as popular as Microsoft's Azure SignalR. The popularity of Python as a programming language in building applications and data science is most likely a driver in the use of Django Channels.

Top 6 most popular WebSocket frameworks/plugins among respondents

Azure SignalR	14%
Django Channels	14%
Laravel Broadcast	8%
Rails ActionCable	6%
Rails AnyCable	6%
Lavarel Echo	5%
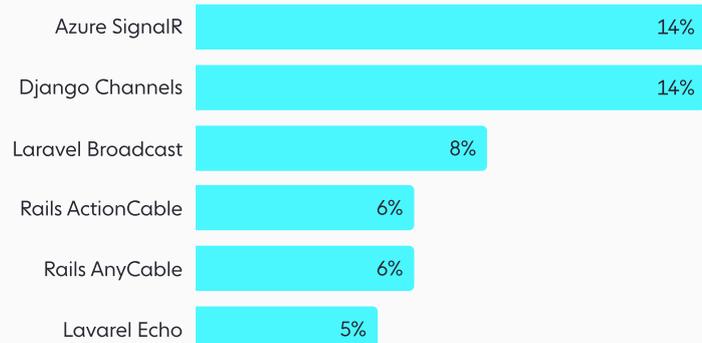
Who is building robust and resilient WebSocket infrastructure and how?

Outages are relatively common when the infrastructure that supports live data is built in-house. Almost two-thirds of survey respondents reported an outage or significant downtime in the past 12-18 months. Given that reliability and availability are critical to the provider's success and the end users' experience, it is worth considering what differentiates those who have experienced interruptions from those who have not.

Organizations that estimated their need for resources accurately built more reliable infrastructure

One factor that characterizes businesses that have managed to build resilient WebSocket infrastructure from those that have not, is the ability to estimate the resources required to build it. Of the respondents that did not have an outage, 60% estimated the required number of engineers accurately, whereas, for organizations that had a problem, only 42% were on point with their grasp of the resources required.

While this differential may be caused by a number of factors, a firm knowledge of realtime technology projects and what it takes to build them is an important requirement for success.

Did the actual amount of engineering resource used to complete the project match what you initially expected, and have you had an outage or downtime?

Only used the amount of engineers originally assigned	60% / 42%
Needed 1 extra engineer	22% / 29%
Needed 2-3 extra engineers	12% / 19%
Needed 4-6 extra engineers	3% / 8%
Needed 7-9 extra engineers	3% / 2%

Has your WebSocket (realtime) infrastructure experienced an outage or significant downtime in the last 12-18 months?

● No   ● Yes
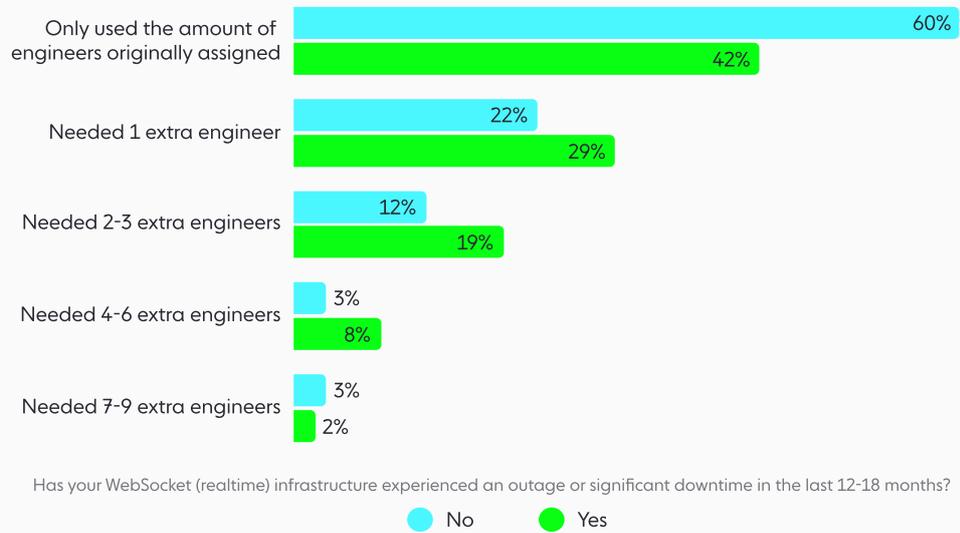
Firms with 1000 to 3000 employees tend to have less reliable infrastructure

Another indicator of infrastructure that is less robust is the size of the company that built it. From the survey data, we discovered that midsized companies with 1000-3000 employees were more likely to experience outages.

Almost three-quarters of firms with 1000-3000 employees have had an outage in the past 12-18 months. This compares to 62% of firms with 3000+ employees and 58% of companies with less than 1000 employees.

Percentage of firms that have had outages by company size

	No outage	Outage
>1000	42%	58%
1000-3000	26%	74%
3000+	38%	62%

● No outage   ● Outage
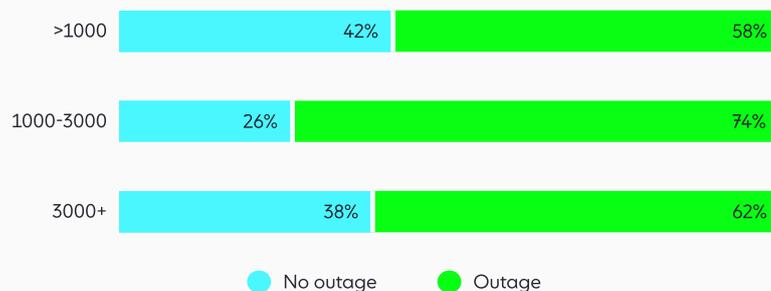
Who is moving to a serverless WebSocket PaaS and why?

The complexity, costs and challenges around building and maintaining proprietary WebSocket infrastructure are persuading some organizations to use a serverless WebSocket platform as a service (PaaS) provider. There are many reasons to make the move to a PaaS, but some are more important than others.

Organizations are moving to a serverless WebSocket PaaS for better user experiences

Our calculations suggest that an average organization can save thousands of dollars by moving to a serverless WebSocket PaaS versus building their own realtime, Websocket-based infrastructure. However, cost reduction is not always the primary reason that organizations are abandoning their DIY strategy. Motivation also includes the opportunity to improve their user experience and to reduce their time to market.

Out of six options, 60% of respondents ranked improving the existing live user experience with a more stable and dependable infrastructure as a top-three reason for switching to a serverless WebSocket PaaS. 18% ranked this as the most important reason to utilize a PaaS.

Risks around bringing new features and services to market were also identified as important reasons to move to a serverless WebSocket PaaS. 56% of respondents cited redeploying engineering bandwidth away from infrastructure and into core product work to move faster as a reason to move to a serverless WebSocket PaaS. 55% said moving to a PaaS reduces the risk of delivering new user experiences that require realtime capabilities.

Top reasons respondents would switch to a serverless WebSocket PaaS provider

Below are the top three reasons engineers would switch or have switched to a serverless WebSocket PaaS provider to manage their live and collaborative applications:

1. Improve the existing live user experience with a more stable and dependable infrastructure.

2. Redeploy engineering bandwidth away from infrastructure and into core products to work and move faster.

3. Reduce the risk of delivering new user experiences that require new realtime capabilities.

Top 3 reasons for switching to a realtime, serverless WebSocket PaaS

Reason	Rank 1	Rank 2	Rank 3
Improve live user experience with more dependable infrastructure	18%	24%	18%
Redeploy engineering bandwidth away from infrastructure and into core product work	16%	18%	22%
Reduce risk of delivering new user experiences that require new realtime capabilities	16%	17%	22%
Improve DevOps efficiencies	15%	12%	10%
Reduce cloud costs	15%	11%	11%
Reduce in-house complexity	14%	12%	11%

● Rank 1   ● Rank 2   ● Rank 3

Reasons to switch to a serverless WebSocket PaaS based on those who experienced outage vs those who did not

Reason	Outage	No outage
Improve live user experience with more dependable infrastructure	21%	13%
Redeploy engineering bandwidth away from infrastructure and into core product work	17%	13%
Reduce risk of delivering new user experiences that require new realtime capabilities	18%	14%
Improve DevOps efficiencies	14%	20%
Reduce cloud costs	16%	9%
Reduce in-house complexity	13%	18%

● Outage   ● No outage

Engineers with more reliable infrastructure are more likely to move to a serverless WebSocket PaaS to simplify

The majority of engineers who have not had an outage in the past 12-18 months ranked reducing in-house complexity and improving DevOps efficiencies as the top reasons to move to a serverless WebSocket PaaS. This demonstrates that, in the absence of reliability issues, simplicity is a key focus.

The data shows that 20% of engineers who have not experienced an outage ranked improving DevOps as the most important reason to move to a serverless WebSocket PaaS. This is compared to just 14% of engineers who have dealt with an outage in the past 12-18 months. Similarly, 18% of respondents who did not experience an outage indicated that reducing in-house complexity was the most important reason to move to a serverless WebSocket PaaS, compared to 13% of the respondents who experienced an outage.

The value a PaaS provides through greater simplicity helps organizations streamline their operations to get new products and features to market faster.

What challenges should we expect in the future?

The demand for realtime data will only grow in the coming years, putting greater pressure on WebSocket infrastructure used to deliver live user experiences. We asked engineers and managers what challenges they foresee and how infrastructure will impact the delivery of digital projects.

Continuity issues are a primary concern for the next few years

In the next 1-2 years, respondents expect to face continuity issues caused by failures. 46% of respondents cited continuity of service when facing failures at local or regional levels as the primary challenge. This could be a failure at an instance or data center, or a more widespread failure caused by capacity issues, DDoS attacks, or intervening network issues.

Main challenges for WebSocket (realtime) infrastructure in the next 1-2 years

Challenge	%
Ensure continuity of service when facing failures at local or regional level	46%
Delivering the same quality of service to larger volumes of users	43%
Dealing with latencies as we're expanding across regions	36%
Handling much larger volumes of messages	35%
Supporting extra realtime features we're planning to add	35%
Fixing messaging being delivered out of order or multiple times	31%
Fixing messages that are not being delivered	27%

Note: respondents were given the option to choose more than one answer

Scalability and the ability to deliver the same level of service to a larger volume of users was the second most common challenge expected in the next 1-2 years, with 43% citing this issue.

How much will WebSocket (realtime) infrastructure challenges slow down delivery of future projects?


- A lot — 48%
- A little — 43%
- Not at all — 9%
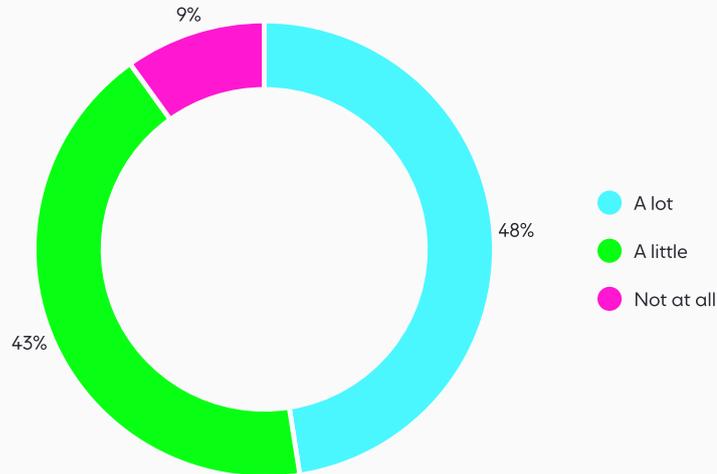
The majority of respondents expect infrastructure challenges to slow down the delivery of digital projects. Only 9% of respondents do not think speed will be negatively impacted.

Messaging issues will have a negative impact on the delivery of digital products

Messaging issues such as non-delivery and out-of-order delivery were not as frequently cited as other issues as a foreseeable challenge in the next 1-2 years. However, 58% of the engineers who are struggling with messages not being delivered are expecting WebSocket infrastructure issues to substantially reduce the speed of digital project delivery. This trend is similar to respondents who see messages being delivered out of order as a big challenge in the next few years. 57% of these engineers also expect infrastructure issues to cause significant delays in digital projects.

Self-build challenges impacting product delivery

Challenge	Expecting no impact	Expecting a little impact	Expecting a lot of impact
Supporting extra realtime features we're planning to add	9%	42%	49%
Dealing with latencies as we're expanding across regions	6%	44%	50%
Fixing messaging being delivered out of order or multiple times	5%	38%	57%
Fixing messages that are not being delivered	5%	37%	58%
Handling much larger volumes of messages	8%	44%	48%
Delivering the same quality of service to larger volumes of users	12%	42%	46%

● Expecting no impact   ● Expecting a little impact   ● Expecting a lot of impact

Scaling users and features has less of an impact on delivery speed. 9% of respondents that see supporting extra features as a challenge expect no impact. 12% of respondents that see scaling the same quality of service to more users as an issue expect no impact. This compares to 6% for the average of the other four categories.

> " We chose a serverless WebSocket infrastructure provider in order to simplify our architecture and ultimately save us valuable engineering time to focus on our own product and user base.

Neville Dastur
Co-founder and CTO, Hospify

Hospify

Capitalizing on a competitive advantage

89% of respondents agreed that scalable WebSocket infrastructure to power live and collaborative experiences will give them a competitive advantage in the future.

Do you think scalable WebSocket (realtime) infrastructure will give you a competitive advantage in the future?

11%

Yes

No

89%
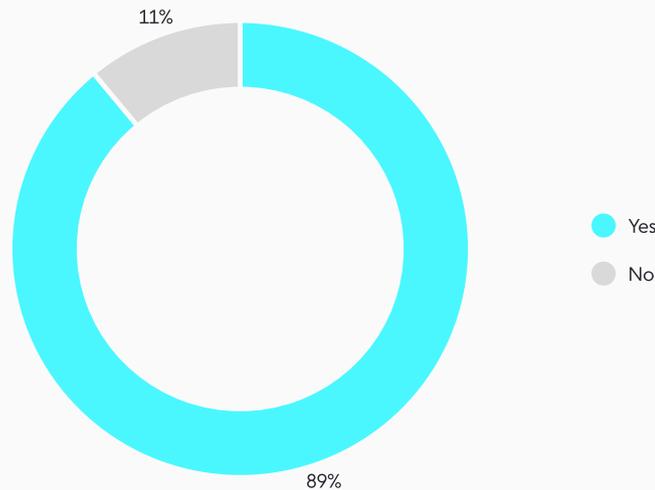
In competitive markets, there are often two considerations that define success: strong product development and customer satisfaction. Working with a PaaS that provides serverless WebSockets at scale supports both considerations:

- Improving customer satisfaction by powering live digital experiences with scalable, resilient infrastructure to mitigate outages; and

- Giving developers more opportunity - in terms of time and money - to focus on their business's core competencies and bring new products and features to market faster.

Talk to a serverless WebSocket expert

Before you begin building or scaling your own infrastructure to power applications in realtime, **book a free consultation with an Ably serverless WebSocket infrastructure expert** and find out how a PaaS solution can benefit your business.

About Ably

Ably is a serverless WebSocket PaaS operating at the edge. There's no infrastructure to provision or manage, just an evolving suite of SDKs and APIs that give you the freedom and flexibility to power live and collaborative experiences with a few lines of code. Our mathematically modeled system design provides a globally-distributed edge network that brings users closer to your app; unique data ordering and delivery guarantees to ensure a seamless end-user experience; a legitimate 99.999% uptime SLA, underpinned by fault tolerant infrastructure; and instant elasticity to enable effortless scale.

Brands like HubSpot, Toyota, and Webflow trust Ably to power live and collaborative experiences like business-critical chat, order delivery tracking, or document collaboration for millions of simultaneously connected devices around the world.

Start with a free account   Talk to an Ably expert

Get in touch

Visit
ably.com

Call
+44 20 3318 4689 (UK)
+1 877 434 5287 (USA)

Email
hello@ably.com