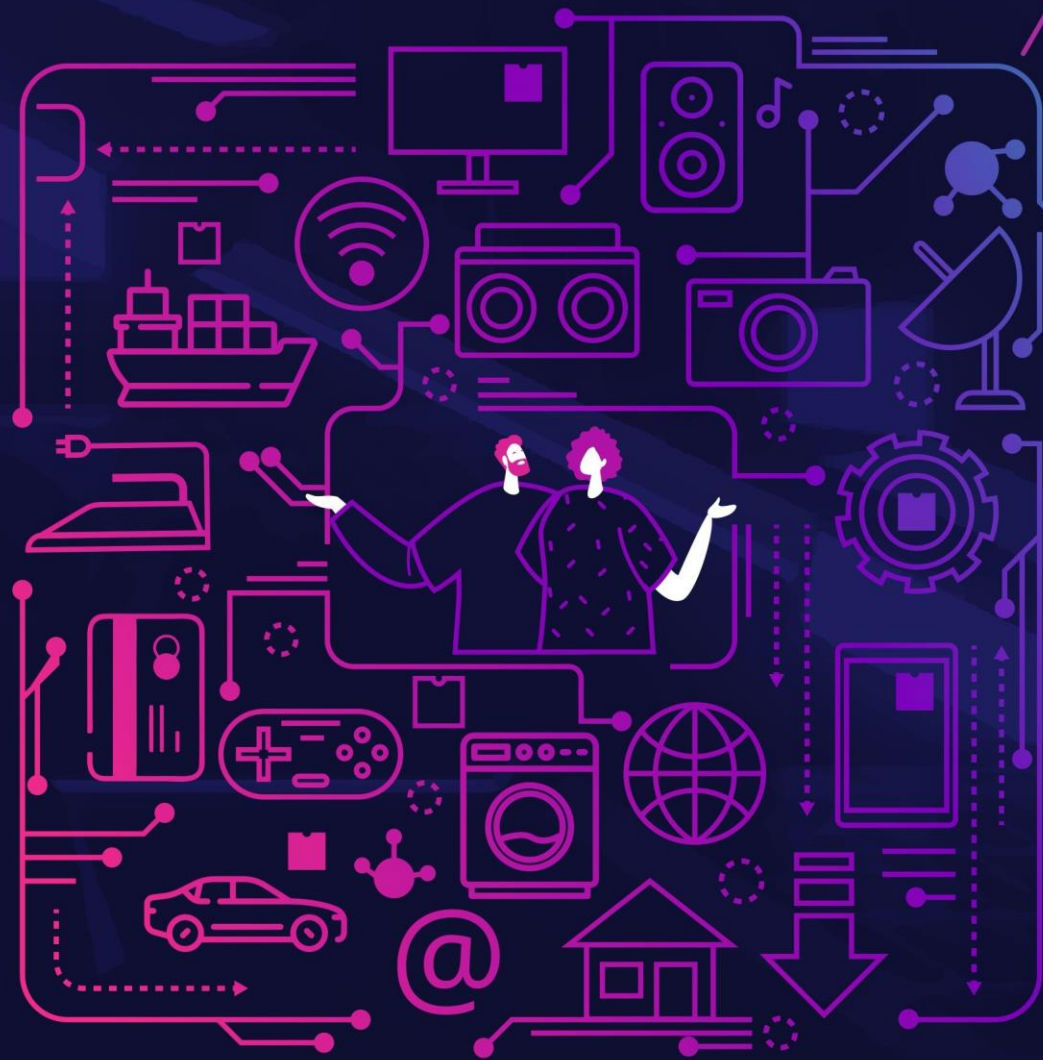


THE STATE OF WebAssembly 2023



/DATA +  **CLOUD NATIVE
COMPUTING FOUNDATION**

SEPTEMBER 2023

Can I share data from this report?

1. License Grant

This report is licensed under the [Creative Commons Attribution-NoDerivatives Licence 4.0 \(International\)](#). Put simply, subject to the terms and conditions of this license, you are free to:

Share — You can reproduce the report or incorporate parts of the report into one or more documents or publications, for commercial and non-commercial purposes.

Under the following conditions:

Attribution — You must give appropriate credit to SlashData™ and to The Linux Foundation as sponsors of this report, and indicate if changes were made. In that case, you may do so in any reasonable manner, but not in any way that suggests that SlashData™ endorses you or your use.

NoDerivatives — you cannot remix or transform the content of the report. You may not distribute modified content.

2. Limitation of Liability

SlashData™, believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing in this publication only and the information, including the opinions contained herein, are subject to change without notice. Use of this publication by any third party for whatever purpose should not and does not absolve such third party from using due diligence in verifying the publication's contents. SlashData™ disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose.

SlashData™, its affiliates, and representatives shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

**The analyst of the developer economy | formerly known as VisionMobile
SlashData © Copyright 2023 | Some rights reserved**

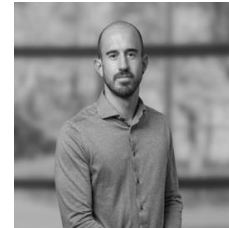


Nikita Solodkov

Senior Market Research Analyst

Nikita Solodkov is a multidisciplinary researcher with a particular interest in using data-driven insights to solve real-world problems. He holds a PhD in Physics and has over five years of experience in data analytics and research design.

✉ nikita.solodkov@slashdata.co



Brayton Noll

Senior Market Research Analyst

Brayton Noll is a behavioral scientist with a background in climate change and environmental research. He holds a PhD from TU Delft in computational social-science with his thesis focusing on human behavioral dynamics and climate adaptation. He has five years of experience working with data analytics.

✉ brayton.noll@slashdata.co

ABOUT THE AUTHORS

INTRODUCTION

5

6

KEY INSIGHTS

WEBASSEMBLY

7

19

WEBASSEMBLY
SYSTEM INTERFACE

METHODOLOGY

23



TABLE OF CONTENTS

INTRODUCTION

[WebAssembly](#) is a binary instruction format that provides a portable compilation target for a wide range of programming languages. It aims to take advantage of the common hardware capabilities of a wide range of platforms to enable execution at near-native speeds.

The aim of this report is to investigate the current state of this technology. We will explore the languages for which developers use WebAssembly as a compilation target and the performance benefits it provides when migrating their projects. We will also look at aspects that attract developers to WebAssembly, the challenges they face, and how optimistic they are about the future adoption of this technology. Finally, we will conclude by focusing on the [WebAssembly System Interface](#) (WASI), gauging familiarity levels and uncovering the key motivations behind its usage.

The findings of this report are based on data collected from an online survey designed, produced, and carried out by [SlashData](#) in collaboration with the [Linux Foundation](#) and the [Cloud Native Computing Foundation](#) (CNCF). The survey reached 255 respondents who use WebAssembly and took place over a period of two weeks in Q3 2023 (21 June – 4 August).



KEY INSIGHTS

- The scope of WebAssembly has grown far beyond the web, with many using it for applications across a wide range of applications and services. →
- The overall sentiment on the future adoption of WebAssembly is mostly optimistic for both web and non-web environments. However, some developers are not yet convinced by what this technology has to offer. →
- The added flexibility and improved performance that WebAssembly offers attract developers to utilise it in a multitude of ways. With many overlaps, developers write applications that aim to use WebAssembly from the start and port or migrate existing applications with the use of this technology. →
- When migrating existing applications to WebAssembly, 30% of respondents experience performance benefits of more than 50%. →
- JavaScript stands out as the most popular programming language that is used for WebAssembly applications. →
- Rust stands out in popularity in WebAssembly projects compared to other use cases. →
- 34% of WebAssembly users state that they are currently using the WebAssembly System Interface (WASI) in their projects. A further 34% report planning to adopt it in the next 12 months. →
- Portability and easier deployment are the most important aspects that attract developers to WASI. →
- HTTP, IO/streams, and SQL stand out as the most anticipated upcoming WASI features. →



WEBASSEMBLY

01

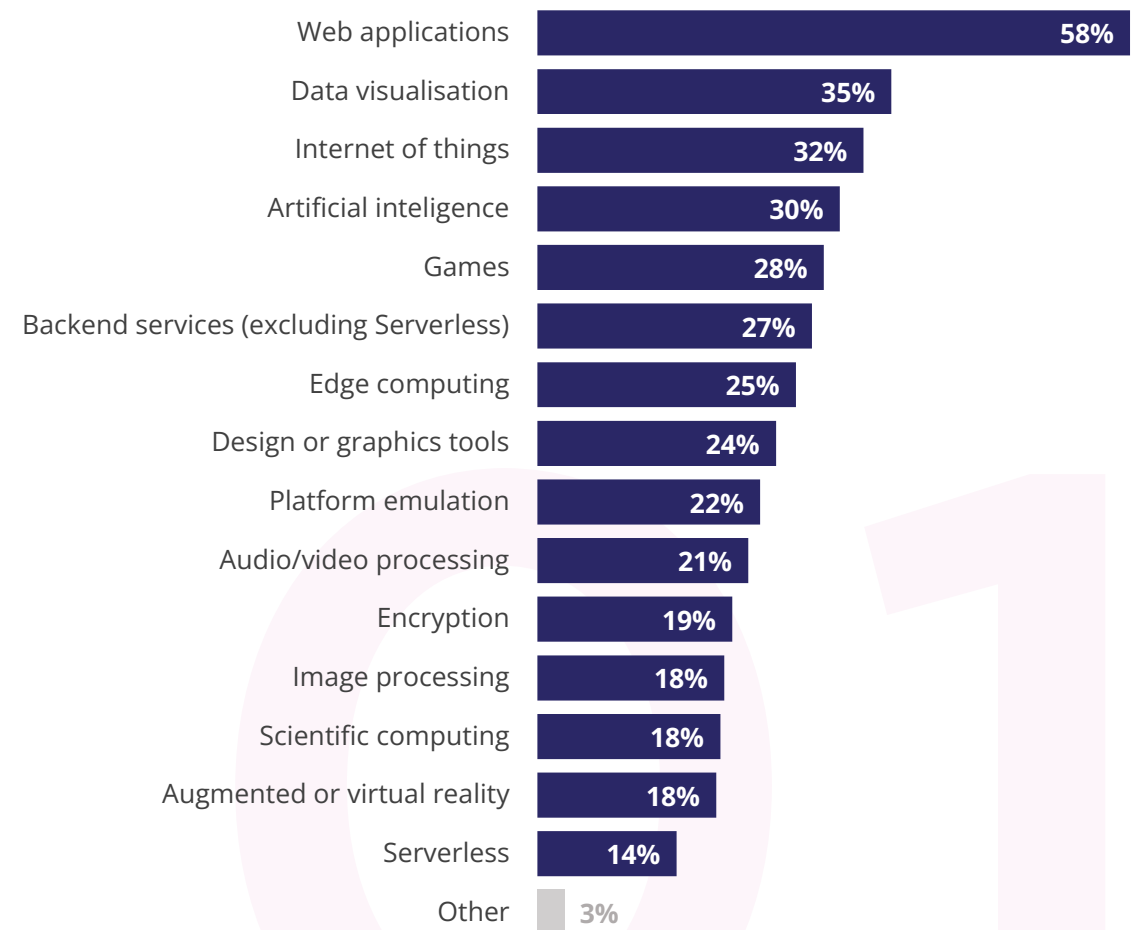
1. WebAssembly

The capabilities of WebAssembly have grown considerably since its release in 2017, and so too has its viability of being used in stable software projects. While this technology was designed to be a part of the open web platform, its usage is not limited to the web environment. It is worth noting that while WebAssembly is still primarily used to develop web applications, this is changing over time as WASI matures (see [Chapter 2](#)).

More specifically, 58% of our respondents indicate this in a multi-select question about the types of applications or services for which they use WebAssembly. Beyond this, we see a significant representation of this technology across a wide range of projects. This indicates that WebAssembly has a lot of potential and can be beneficial to all developers across a multitude of sectors and not just those involved in front-end web development.

What is WebAssembly used for?

% of respondents (n=255)



1. WebAssembly

What attracts developers to WebAssembly?

% of respondents (n=254)



The wide scope of applications WebAssembly can be used for is made possible by the many benefits it offers. Faster loading times (23%) and opportunities to explore new use cases and technologies (22%) are the most frequently mentioned aspects that have convinced respondents of this survey to start using WebAssembly.

We also see that sharing code between projects (20%), the use of binaries that run anywhere (18%), and the technology being language agnostic (18%) have relatively high selection rates. The acknowledgement of these benefits indicates that WebAssembly is well-recognised not only for its performance-related benefits but also for the flexibility it brings to software development.

1. WebAssembly

With the flexibility that WebAssembly offers, developers can take advantage of its features in a multitude of ways. Our data indicate that 76% of WebAssembly users are developing new applications with plans to use this technology from the start. A further 64% are taking advantage of the portability that WebAssembly offers by porting their existing applications to new platforms. We also see that 62% are migrating existing applications to new programming languages on the same platform.

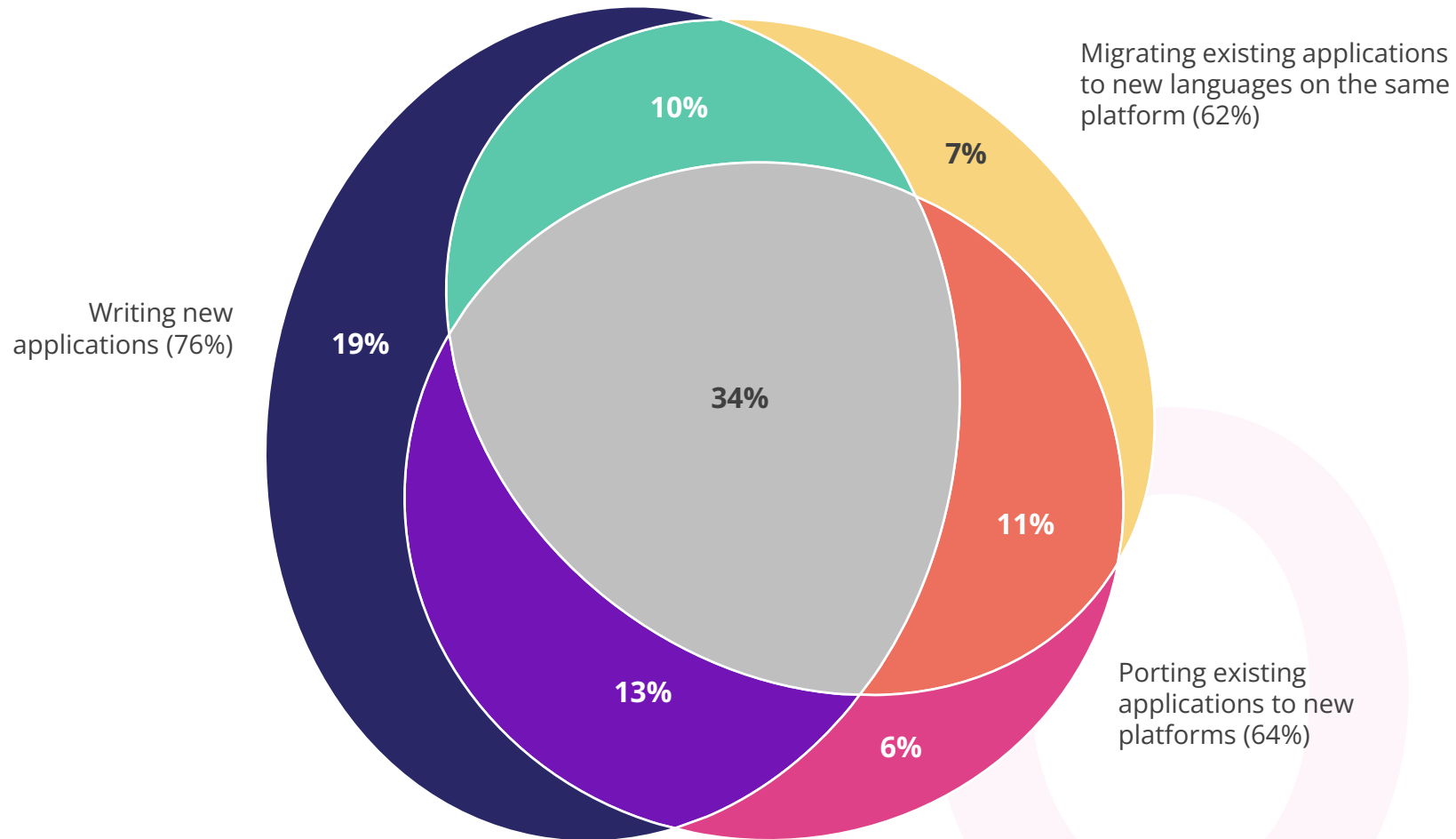
We note that there is a large overlap between the above-mentioned methods, with 34% indicating their involvement in all three across their projects. Some developers may see tremendous benefits from migrating existing applications to new languages and porting them to new platforms simultaneously. In such cases, WebAssembly provides an easy, two-in-one solution that brings consistency along with all the other benefits that this technology has to offer.



1. WebAssembly

How is WebAssembly used?

% of respondents (n=209)



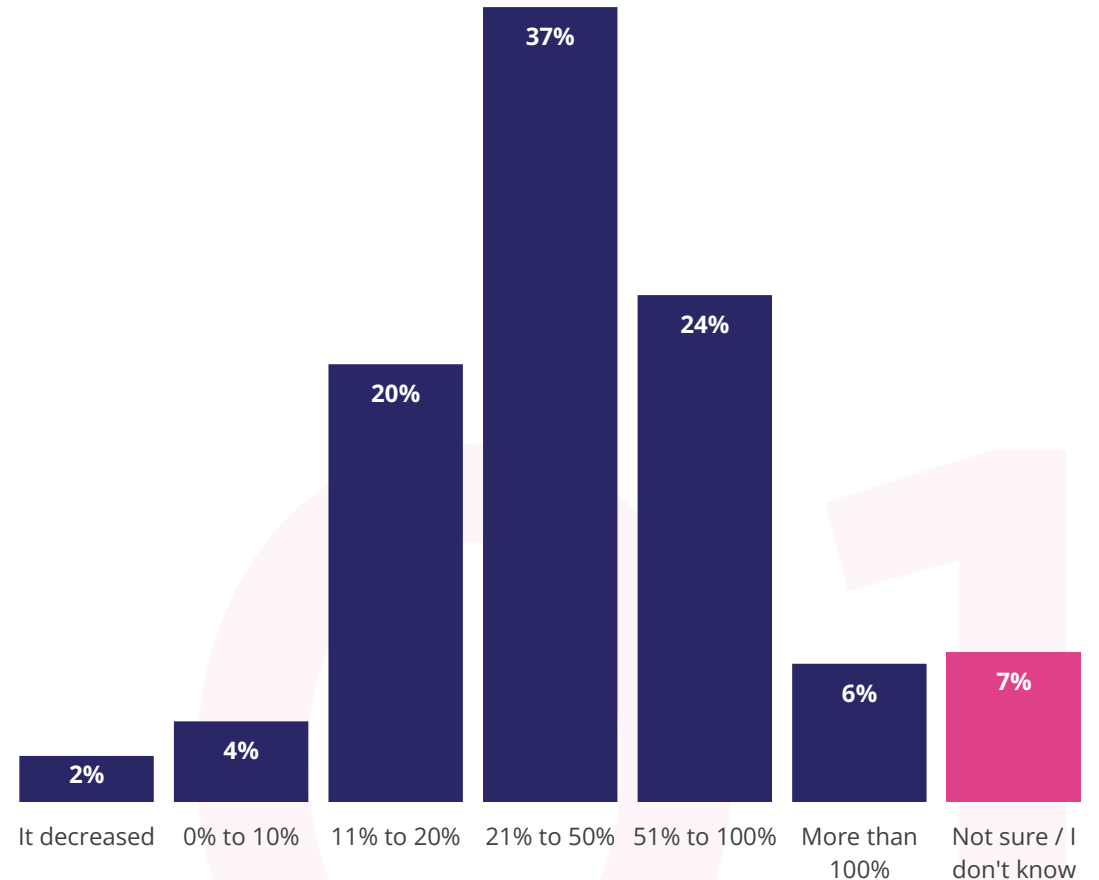
1. WebAssembly

When migrating existing applications to new languages in WebAssembly projects, 37% of respondents report observing performance¹ increases in the 21% to 50% range. A further 30% report that the performance of their applications has increased by more than 50% as a result of switching languages. This highlights that a large percentage of developers are already seeing one of the key benefits of WebAssembly in their projects and sets up a promising future for the technology. However, we should note a small group (6%) of developers report experiencing minimal performance benefits, suggesting that there may still be ways to go in the pursuit of native speeds.

¹ Respondents were asked to consider the speed of their applications as the main performance metric.

Performance boost from migrating to WebAssembly

% of respondents who have migrated existing applications (n=187)



1. WebAssembly

We asked developers which programming languages or platforms they use WebAssembly as a compilation target in their projects. Respondents were asked to select the language that they use the most as primary and any other languages that they are also using in this context.

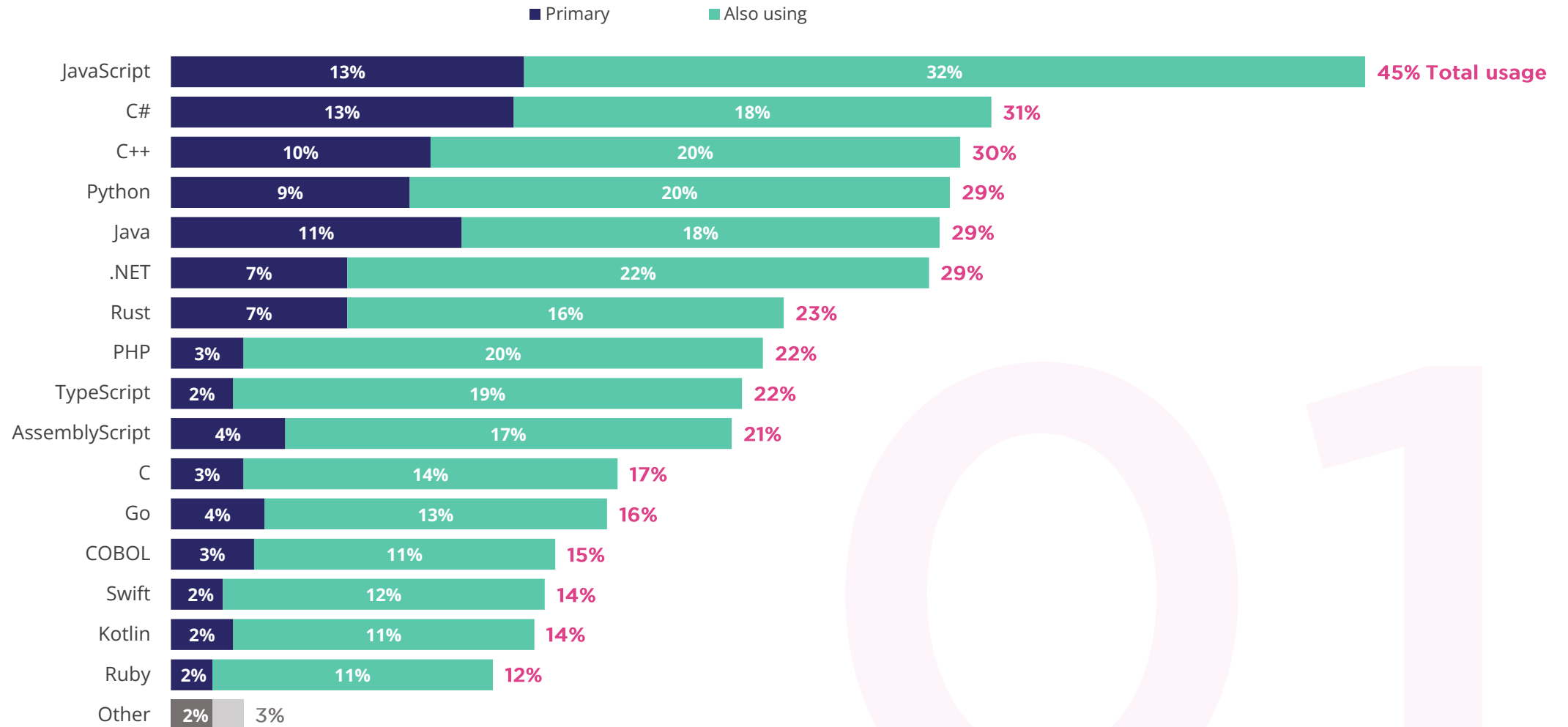
With a selection rate of 45%, JavaScript stands out as the most popular programming language for WebAssembly projects. This likely stems from the strongly anchored base of JavaScript in web development, with it being the most popular language for some time. Developers may wish to reuse the code they have already built in JavaScript in WebAssembly projects. However, we should note that most of its lead over the other languages comes from the “also using” category, with only 13% indicating primary usage.



1. WebAssembly

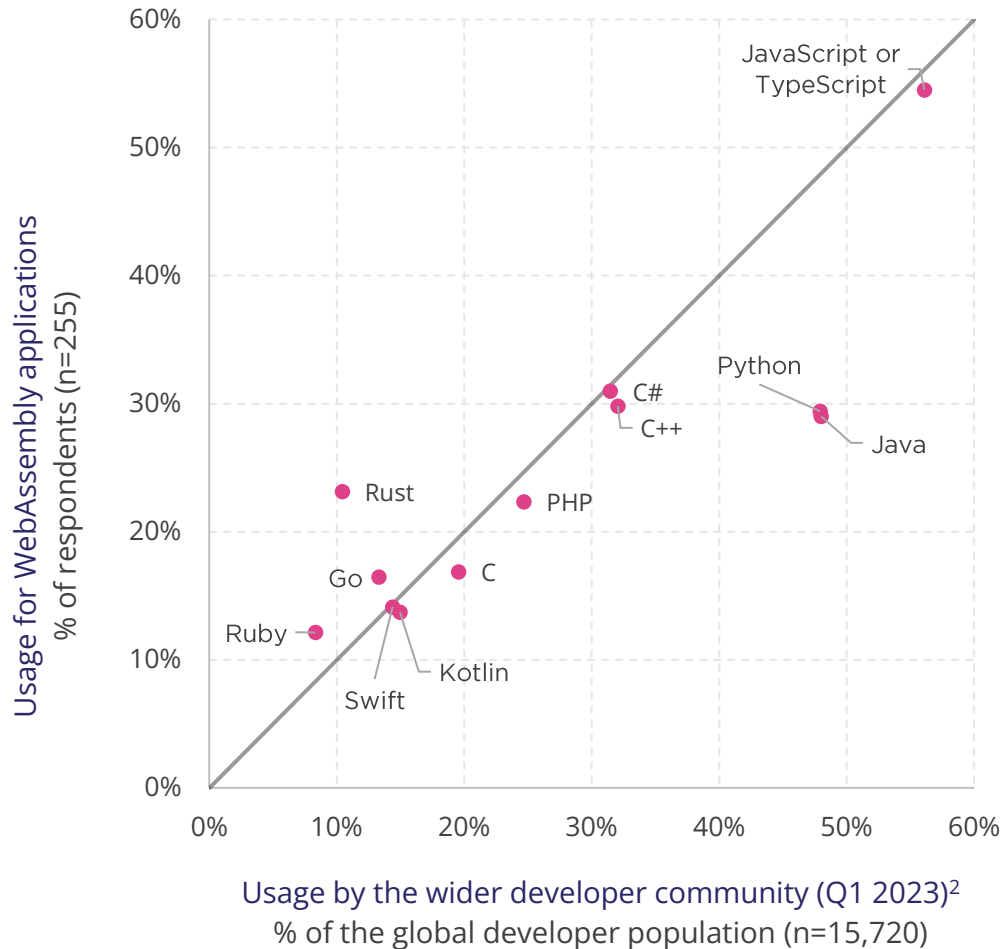
Languages used for WebAssembly applications

% of respondents (n=255)



1. WebAssembly

Popularity of programming languages for WebAssembly applications relative to the wider developer community



On comparing the results of this survey to our broader research², we find that the adoption of most languages aligns well with their global usage levels across other types of projects. From this point of view, we see that Python (29%) and Java (29%) are under-represented in WebAssembly projects. On the other end of the spectrum, the popularity of Rust in WebAssembly projects (23%) far exceeds its total usage across all types of projects (10%).

² The list of languages excludes COBOL, AssemblyScript, and the .NET platform while combining JavaScript and TypeScript. More information can be found in our [“State of the Developer Nation”](#) report, which is based on data collected in our global survey that was fielded in Q1 2023 (24th edition).

1. WebAssembly

As is the case with all technologies, developers can face a wide range of challenges while working with WebAssembly. 83% of the respondents from this survey report being affected by at least one of the listed challenges in their projects.

Difficulties with debugging and troubleshooting are mentioned by 19% of respondents, placing it at the top of the list. This highlights the increased complexity involved in working with WebAssembly, with many nuances that go beyond code readability. An example of this can be seen in 29% of respondents indicating that they have suffered from one of multiple runtime-related issues. These manifest through either experiencing different performances (15%) or a lack of consistent developer experiences (15%) between runtimes.

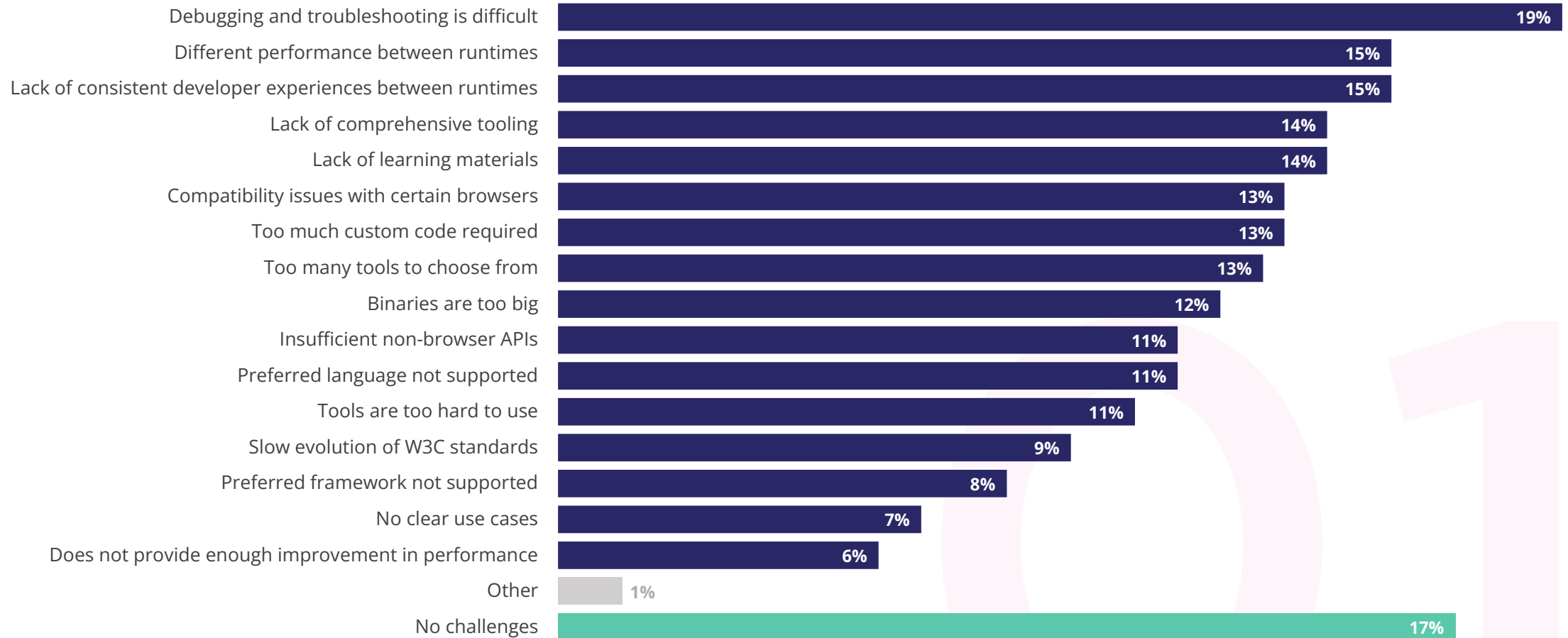
Only 6% of respondents mention that using WebAssembly does not provide sufficient benefits when it comes to the performance of their applications. As noted earlier, With the performance benefits being very important to the respondents, this indicates that WebAssembly manages to meet the expectations of most developers.



1. WebAssembly

Top challenges faced in WebAssembly development

% of respondents (n=254)



1. WebAssembly

The overall sentiment on the future adoption of WebAssembly is mostly optimistic for both web and non-web environments. With WebAssembly being web-based first, the sentiment on its future adoption is higher for the web than for non-web environments. More specifically, 39% of respondents are very optimistic about WebAssembly applications running on the web, while only 26% are for non-web environments.


Around 22% indicate that they are pessimistic about its future adoption for either web (15%) or non-web environments (15%). Considering that this includes developers who are already using WebAssembly, this indicates that some developers are not yet convinced by what this technology has to offer. However, it is important to note that it is an evolving technology with many features yet to come, some of which we will explore in the following section of this report.

Sentiment on the future adoption of WebAssembly

% of respondents (n=250)

Very pessimistic Somewhat pessimistic Neutral Somewhat optimistic Very optimistic





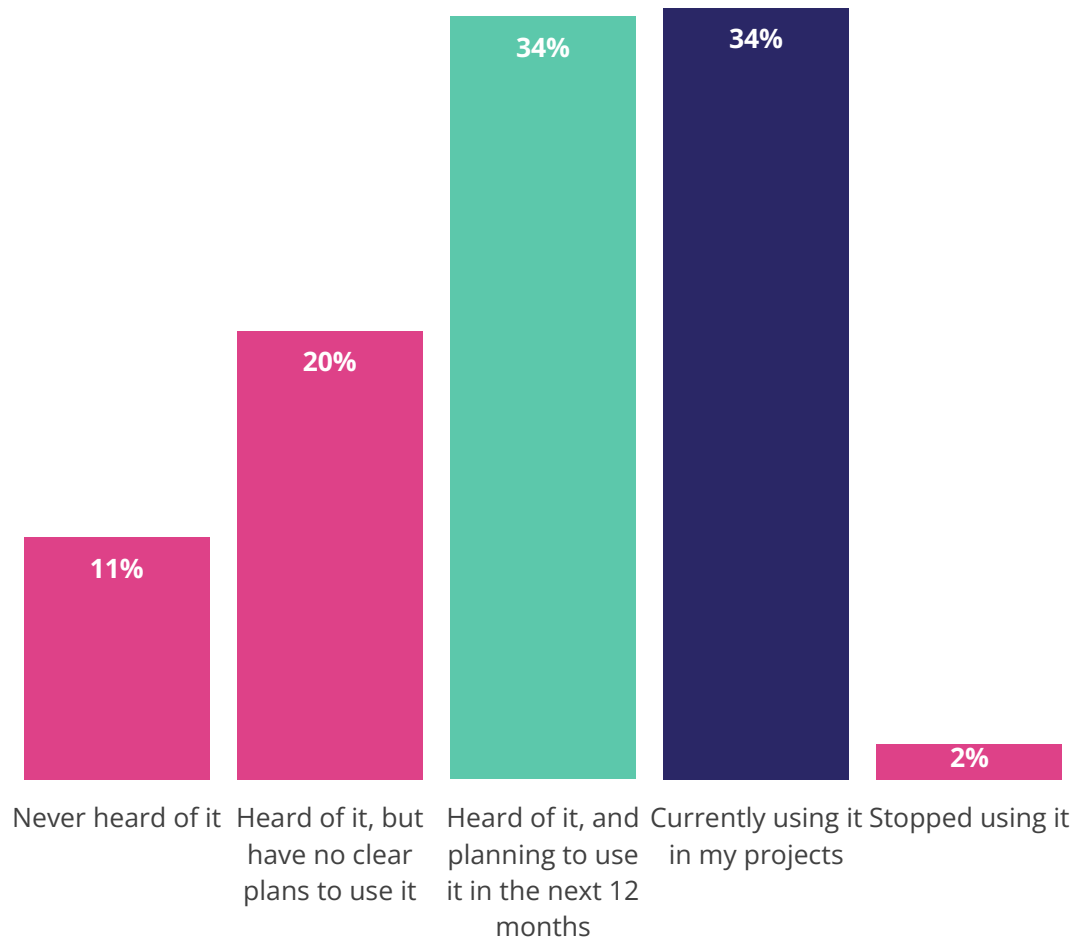
**WEBASSEMBLY
SYSTEM
INTERFACE**

02

2. WebAssembly System Interface

Awareness of the WebAssembly System Interface (WASI)

% of respondents (n=252)

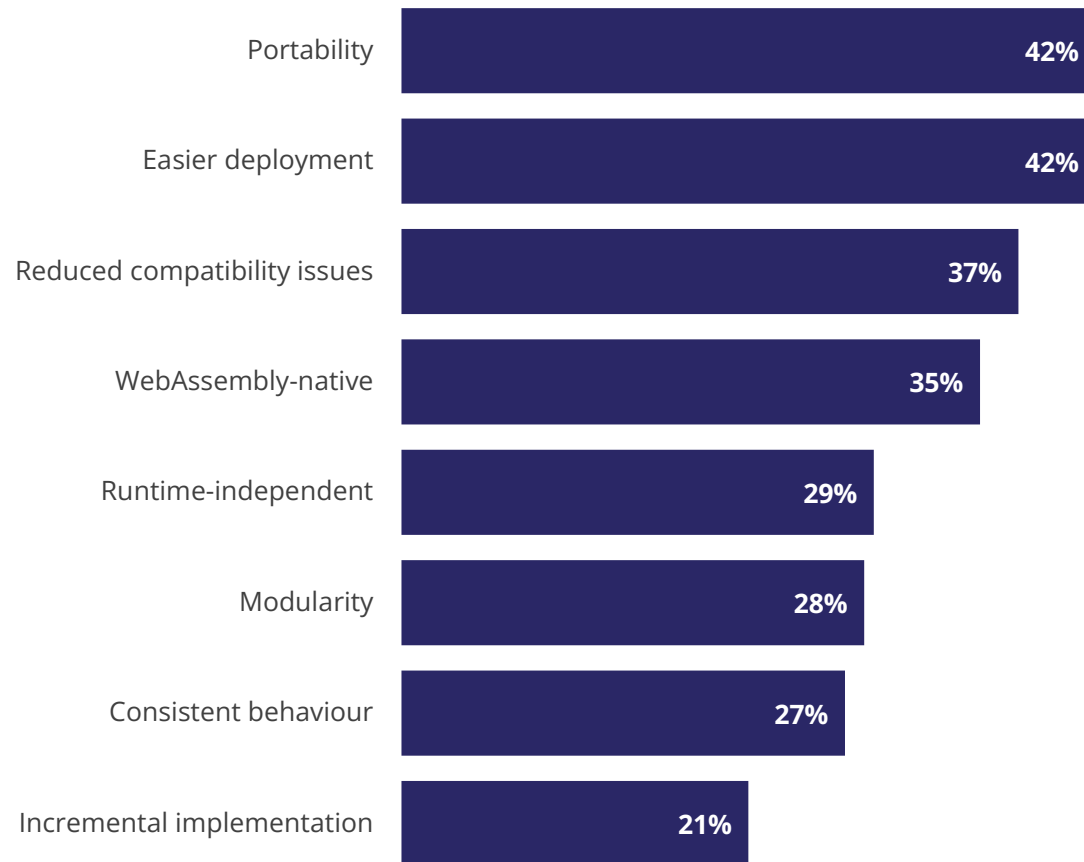


With all the benefits of WebAssembly, there is a lot of interest in using this technology beyond the browser. [WebAssembly System Interface \(WASI\)](#) is a modular system interface for WebAssembly aims to do precisely that. This is done through a collection of standardised APIs that do not depend on browsers and are not required to be compatible with JavaScript. Our data suggest that 34% of WebAssembly users state that they are currently using WASI in their projects, and a further 34% report planning to adopt it in the next 12 months.

2. WebAssembly System Interface

What attracts developers to WASI?

% of respondents who are currently using or planning to use WASI (n=171)



We asked respondents who are currently using or planning to use WASI in their projects about their rationale. Our data indicates that portability and easier deployment are the most important aspects that attract developers to WASI, each being mentioned by 42% of our respondents. Along with other attributes, these can help propel the widespread adoption of WebAssembly for those who want to take advantage of its benefits beyond its use in web applications.

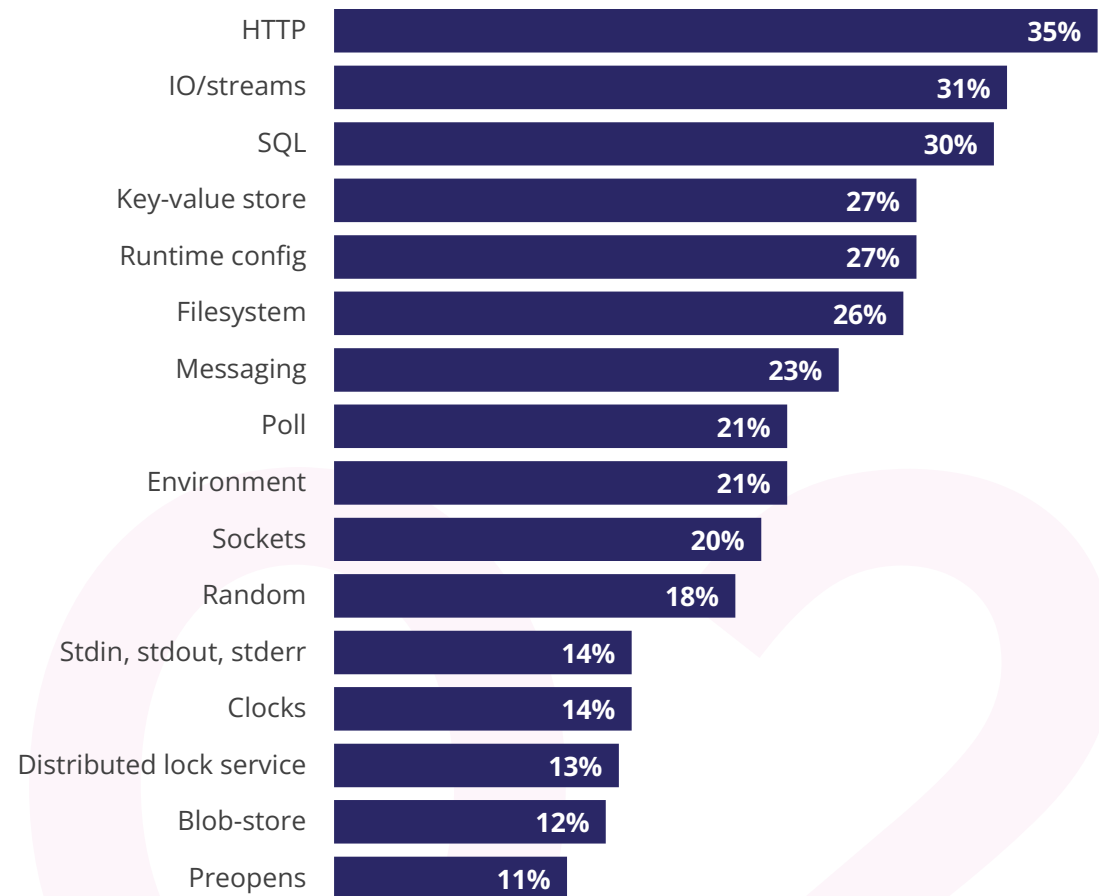
2. WebAssembly System Interface

Just like WebAssembly, WASI APIs are developed as proposals through a [standardised W3C process](#). We asked respondents who are currently using or planning to use WASI which features¹ they are looking forward to the most. HTTP (35%) stands out as the most anticipated feature, with a selection rate of 35%. This is closely followed by IO/streams (31%) and SQL (30%), along with key-value store (27%), runtime config (27%), and filesystem (26%) slightly further down the line.

¹ A list of active WASI proposals can be found on [GitHub](#).

Most anticipated WASI features

% of respondents (n=169)





METHODOLOGY

In Q2 2023, SlashData designed and ran an online survey for The Cloud Native Computing Foundation (CNCf) to explore the current state of the WebAssembly ecosystem. We conducted the analysis presented in this report based on the 255 respondents who replied to this custom survey. Many of the questions in this custom survey are specifically designed and co-created to address the CNCf's business objectives.

Every SlashData survey is monitored and cleaned to ensure the highest standards of retained responses. Our proprietary cleansing is designed to mitigate and remove opportunistic, fraudulent and bot responses. Consisting of multiple criteria formulated around logic rules, speed, consistency and response taking behaviour; this holistic assessment is key to our continued success.



Understand developers. Inspire the future of technology.

We survey 30,000+ developers annually – across Web, Desktop, Cloud, Mobile, Industrial IoT, AR/VR, Machine Learning and Data Science, Games, Consumer Electronics and Apps/Extensions for 3rd party ecosystems - to help companies understand who developers are, what they buy and where they are going next.



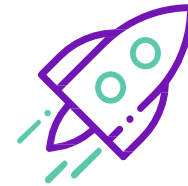
WHO DEVELOPERS ARE

Developer population sizing
Developer segmentation



WHAT THEY BUY

Why developers are adopting
competitor products – and how you
can fix that



WHERE THEY ARE GOING

Emerging platforms – augmented &
virtual reality, machine learning



/DATA +  **CLOUD NATIVE**
COMPUTING FOUNDATION