

32. Goodremont E. Special Steels. – M.: Metallurgizdat, 1960. t.II. – 586 p.
33. Study of gas saturation of the edges of parts during plasma cutting and its influence on the quality of welded joints in case steels / V.S. Golovchenko, V.V. Dolgorukov, N.D. Yellow-bellied and others. // Technology of shipbuilding, No. 4, 1973. – P. 26-31.
34. Kokhlikyan L.O. Investigation of the nature of the porosity of welds after air-plasma cutting of steels. Tr. VNIIVAVTOGENMASH. "Thermal cutting and flame treatment of metals", 1976. – P. 24-27.
35. Saturation with gases of edges made by air-plasma cutting and its influence on the formation of pores in welds / K.V. Vasiliev, G.A. Asinovskaya, L.O. Kokhlikyan et al. // Automatic welding, No. 9, 1974. – P. 67-68.
36. Morozov A.N. Hydrogen and nitrogen in steel. – M.: Metallurgy, 1960. – 282 p.
37. Ganz S.N., Parkhomenko V.D. Obtaining bound nitrogen in plasma. – K.: Vishcha school, 1976. – 190 p.
38. Petrov S.V., Saakov A.G. Plasma of combustion products in surface engineering. – Kyiv: NPP "Topaz", 2000. – 217 p.
39. Podgaetsky V.Z. Pores, inclusions and cracks in welds. – K.: Technique, 1970. – 236 p.
40. Nikiforov G.D. Influence of finished interfaces on the release of dissolved gas bubbles // Welding production, No. 11, 1973. – P. 46-49.
41. Pokhodnya I.K. Gases in welds. – M.: Mashinostroenie, 1972. – 256 p.
42. Frolov V.A., Shishkin Ya.G. Some studies of the causes of pore formation during welding of parts cut by plasma cutting // Shipbuilding Technology, No. 2, 1973. – P. 41-43.
43. Glushchenko L.G., Mitskan I.I. The introduction of oxygen-plasma cutting of carbon steels on machines "Crystal" // Technology of shipbuilding, No. 9, 1980. – P. 25-27.
44. Bogorodsky Yu.A., Khmelevskaya V.B., Filippov A.N. Influence of the parameters of air-plasma cutting of metal on the quality of the cut // Shipbuilding, No. 4, 1984. – P. 42-43.

EXPLORING THE FRONTEND INDUSTRY IN 2022: A COMPREHENSIVE ANALYSIS OF POPULAR TECHNOLOGIES, TOOLS, AND DEVELOPMENT PRACTICES

Kniazev I.

*Senior Software Engineer at June Homes
Istanbul, Turkey*

ABSTRACT

In this article, I present the results of a survey that aimed to gain insight into the current state of the frontend industry. The survey was conducted with more than 3,700 participants from 125 countries. The survey results provide a comprehensive overview of the frontend industry in 2022. It includes information on the demographics of frontend developers, the types of projects they work on, the development methodologies they use, and the tools and technologies they use. The survey results reveal the popularity of JavaScript and React among frontend engineers, the widespread use of Agile development methodologies, and the increasing use of cloud-based platforms and third-party tools. These insights can help frontend developers, designers, and managers make informed decisions when it comes to choosing technologies, tools, and practices for their projects. Additionally, the article will cover the subject of vendors, looking at the platforms and third-party tools frontend engineers use for error tracking and monitoring. Overall, the article will be an invaluable resource for developers, designers, and managers who want to stay up-to-date with the latest trends and practices in the field.

Keywords: Frontend development, Technologies, Tools, survey, Frameworks, Development methodologies, JavaScript, Industry practices, Industry trends.

As technology continues to evolve at a rapid pace, it's essential for developers to stay up-to-date with the latest trends and practices in their field. Frontend development is no exception, with new technologies, tools, and frameworks emerging all the time. To gain a better understanding of the current state of the frontend industry, a survey was conducted with more than 3,700 participants from 125 countries. The survey aimed to gather insights into the demographics of frontend developers, the technologies and tools they use, and the engineering practices they follow. The results of the survey provide valuable insights into the current state of the frontend world, and can help developers make informed decisions when it comes to choosing technologies, tools, and practices for their projects.

The survey included questions about the respondents' demographics, such as their job title, location, and

company size. It also covered topics such as the types of projects frontend engineers work on, their development methodologies, and the tools and technologies they use. Additionally, the survey touched on the subject of engineering practices, such as version control systems, automated testing, and continuous integration and delivery.

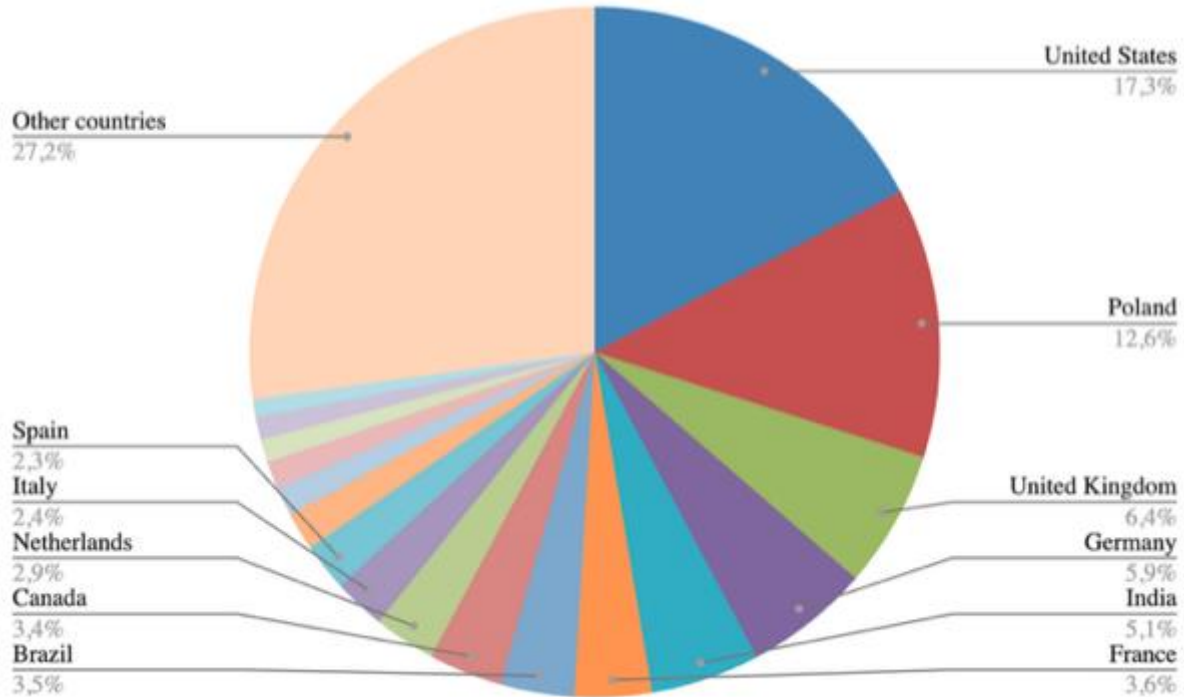
The survey results provide a comprehensive overview of the frontend industry in 2022. They reveal the popularity of JavaScript and React among frontend engineers, the widespread use of Agile development methodologies, and the increasing use of cloud-based platforms and third-party tools. These insights can help frontend developers, designers, and managers make informed decisions when it comes to choosing technologies, tools, and practices for their projects.

In this article, we will cover the survey results in detail, including the demographics of the respondents, the types of projects they work on, the engineering practices they follow, and the technologies and tools they use. We will also delve into the subject of vendors, looking at the platforms and third-party tools frontend engineers use for error tracking and monitoring. The article will provide a comprehensive overview of the state

of the frontend industry in 2022 and will be an invaluable resource for developers, designers, and managers who want to stay up-to-date with the latest trends and practices in the field.

1. Who Filled Out the Survey?

3,700 respondents across 125 countries filled out the survey. The 5 countries with the most responses were the US, Poland, UK, Germany and India.



Pic. 1: Split of respondents, based on country of residence.

Surprisingly, the survey results showed that a relatively small percentage of frontend engineers work at non-tech companies. Only 18% of respondents said they work at a company that is not primarily focused on technology or software development. This is in contrast to the 82% of respondents who identified as working at a software development company, developer agency, or a tech-first or digital-first company.

One possible explanation for this trend is that the survey did not reach a significant number of engineers who work at more traditional companies. However, it's also possible that more engineers are indeed working at places where software is at the core of the business. This could be due to the increasing demand for software and digital solutions across all industries, as well as the growing importance of technology in driving business growth and innovation.

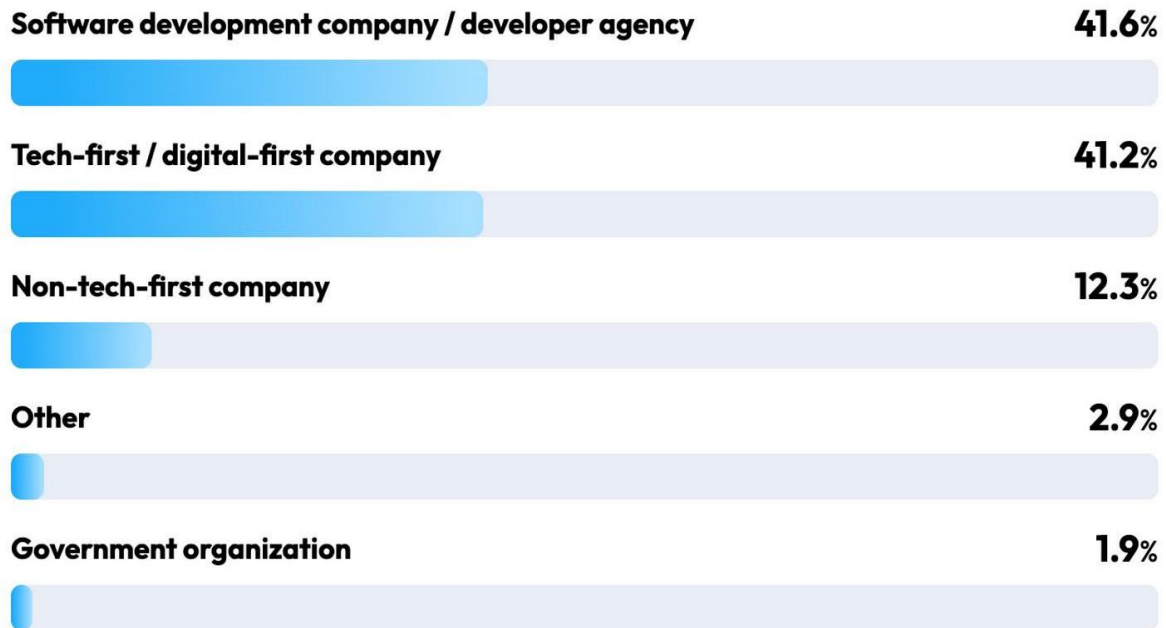
The results of the survey indicate that frontend engineers who work at tech-first or digital-first companies are more likely to be working on projects that are directly related to software development, such as web and

mobile apps, as opposed to engineers who work at non-tech companies who may work on projects that are less related to software development.

It also suggests that engineers who work at tech-first or digital-first companies may have more access to the latest technologies and tools, and may be more likely to work with agile development methodologies, version control systems, and automated testing tools. Engineers working at tech-first or digital-first companies may also have more opportunities to specialize in their field and to advance their careers.

In conclusion, the survey results suggest that a significant percentage of frontend engineers are working at tech-first or digital-first companies, where software development is core to the business. This trend may be driven by the increasing demand for software and digital solutions across all industries, as well as the growing importance of technology in driving business growth and innovation. It is important to understand this trend for companies, startups, recruiters and for the engineers themselves.

Which one of the following best describes your company?



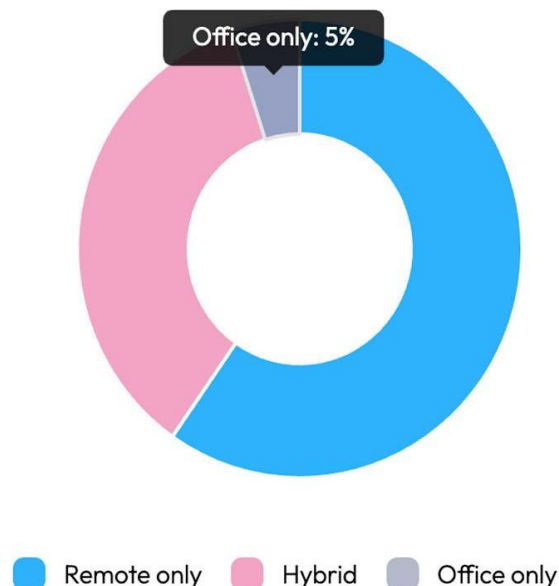
Pic. 2: The split of respondents based on the type of companies they work at.

2. How Frontend Engineers Work

Remote work has been a significant shift in the software engineering field in the past two years. The rise of remote work should not be surprising, but the

speed at which it happened is. A staggering 56% of respondents reported working remotely, with only 5% working only in an office. This shift is so recent that the survey in 2020 did not even measure this data point.

What does your work look like now?



Pic. 3: The split of work type

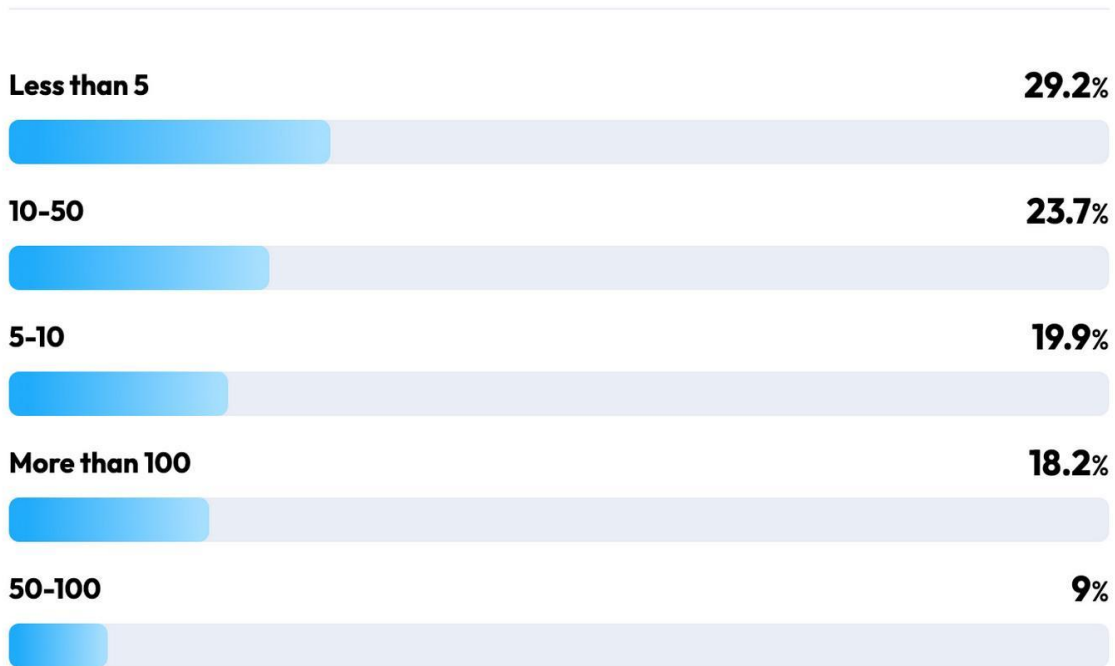
As we look ahead to 2023, the question remains whether full-remote work will become the norm or if hybrid work will gain popularity. Most engineers prefer working remotely due to the lack of commute and fewer

interruptions. However, it remains challenging to replicate the spontaneous discussions and information sharing that occur in an office setting.

Within the frontend engineering field, larger teams are becoming more common. 27% of respondents reported working at a company with more than 50 frontend engineers, while 30% reported working at a company with 5 or fewer frontend developers. 50% of respondents work at companies with 10 or more frontend engineers. This trend shows an interesting duality, with almost as many frontend engineers working

at companies with large frontend teams as those working on small teams or alone. The developer experience and expectations at these companies are vastly different, with larger companies having more opportunities for developer experience and frontend platform teams, as well as more mentorship opportunities. Smaller companies rely more on individual developers, with fewer options for feedback.

How many frontend developers are there at your company?



Pic. 4: The split of frontend roles within the company

For project management, 69% of respondents use Scrum or Kanban, with Scrum at 52% and Kanban at 33%. 17% of respondents use both. 2 out of 3 frontend

developers use one of these two methods when completing projects. Companies that do not use either of these methodologies tend to be mostly tech-first or digital-first companies.

What's the size of the company you work in?

Hide results from 2020



Pic. 5: The split of employees size

It's worth noting that frontend development is not limited to frontend engineers alone. The survey also collected job titles such as bootcamp students, self-taught developers, product managers, developer advocates, frontend development architects, design system directors, designers who also code, graphic designers and developers, and even "Head of Everything" for a one-person developer shop. This highlights the accessibility of the frontend field and how it's still possible to get involved in frontend development without a specific background.

3. Engineering Practices

Unit testing is a common practice among frontend engineers, with close to 75% of respondents reporting that they write these types of tests. Integration and end-to-end tests are also prevalent, with about half of respondents having written these tests.

Initially, I was surprised to find such a high percentage of survey respondents writing unit tests. This is because while unit testing is a standard practice for backend development, where all functionality is business logic, and thus unit tests are an immediate win, frontend engineering includes a lot of visual parts where unit tests are not as effective. However, frontends are becoming more complex and contain

much more business logic. Also, as soon as more than a few engineers work on a frontend codebase, unit tests and integration tests are an easy way to reduce regressions in the codebase.

The relatively high percentage of unit testing shows how much frontend engineering is maturing and that frontend development is no longer just about the UI. Most frontend developers are, in fact, software engineers building for the long term, with maintainability in mind. Code reviews are also common practice within the industry, with 80% of respondents mentioning they follow this practice.

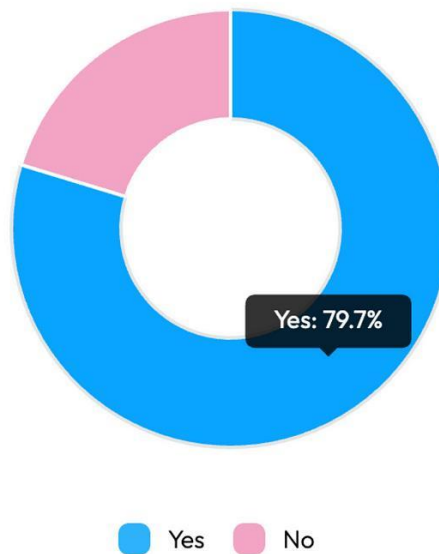
Interestingly, there is a strong correlation between the size of the frontend engineering team and whether engineers do code reviews. Engineers working at large companies are the least likely to not do code reviews. Engineers working at companies with 50 or fewer employees are twice as likely to not do code reviews than those working at larger companies. One-person companies are the most likely to not do code reviews. This makes sense as the more engineers there are, the more value code review brings not just in spotting issues, but also in spreading knowledge better.

Continuous integration and continuous deployment (CI/CD) is also a widespread practice within the

industry, with 79% of respondents saying they use CI. However, it is intriguing to see that about 20% of engineers don't use CI/CD when it has become increasingly simple to set up a CI pipeline to deploy code to production right after code has been merged into the main

branch. This can be done using solutions like GitHub Actions, GitLab CI, Vercel, Netlify, and many others. Doing this makes sense even if you're working on a solo project, as it saves time and eliminates the need to manually deploy every time.

Do you use Continuous Integration?



79% of respondents use Continuous Integration.

Pic. 6: The split of CI usage

It is worth noting that not writing unit tests, not having CI/CD, and not doing code reviews are correlated. Engineers who do at least two of the above activities are almost certain to do all three. This suggests that by introducing unit testing and setting up CI/CD, code reviews will likely follow. Or, perhaps the other way: engineers who want to do code reviews tend to write tests and set up CI/CD.

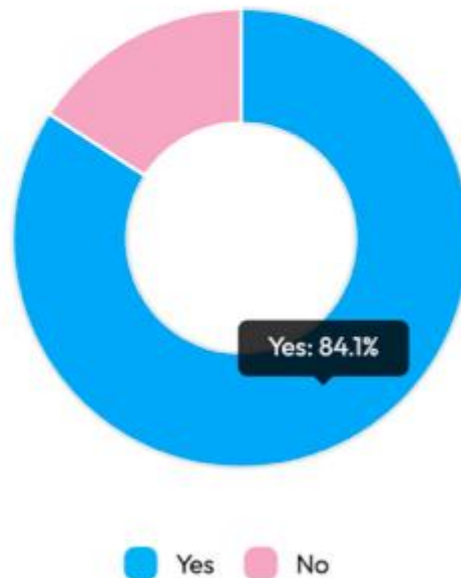
Other engineering practices worth mentioning, which survey respondents brought up, are trunk-based development, feature flags and feature toggling, pair

programming, linting, code style guidelines, JSDocs, stakeholders map, design systems, and error tracking and monitoring. These practices can serve as inspiration for other approaches that you may want to try out.

4. Technologies

TypeScript has become the de facto language of frontend engineering, with 84% of respondents having used the language in the past year. This is not surprising, as TypeScript offers the benefits of a strongly typed language, resulting in fewer errors and regressions, and making software engineers more productive.

Over last year, have you used Typescript?



Pic. 7: Typescript's meteoric rise

Statically typed programming languages require that the types of variables are known at compile time, and are enforced by the programmer declaring the type. Strongly typed languages include Java, C#, Go, C, C++, Haskell, Swift, Kotlin, and TypeScript. Dynamically typed languages, on the other hand, perform type checking during runtime, and include languages such as Python, PHP, Ruby, and JavaScript.

The weak type system of JavaScript is a major problem, as countless mistakes make it into production. Typescript addresses this issue in an elegant way, and it is no wonder that it has become so popular among frontend developers.

React and Next.js are also popular among frontend engineers, with 76% of respondents saying they used and liked React in the past year, and 43% saying the same about Next.js, whose framework is built on top of React. The popularity of Vue remains unchanged at 28%, while Svelte seems to be becoming less popular, with 22% of respondents saying they used and liked it, compared to 33% two years ago.

Frequently used frontend utilities include Axios, Lodash, and Moment. 61% of respondents said they used and liked Axios, a promise-based HTTP client. Lodash is a utility library providing utility functions, making it easier to iterate arrays/objects/strings, manipulate, and test values, and other neat addons. 46% of

respondents said they used and liked it, while 20% said they used it and disliked it. Moment is a library to manipulate dates, with 34% saying they used and liked it, and 41% saying they used and disliked it.

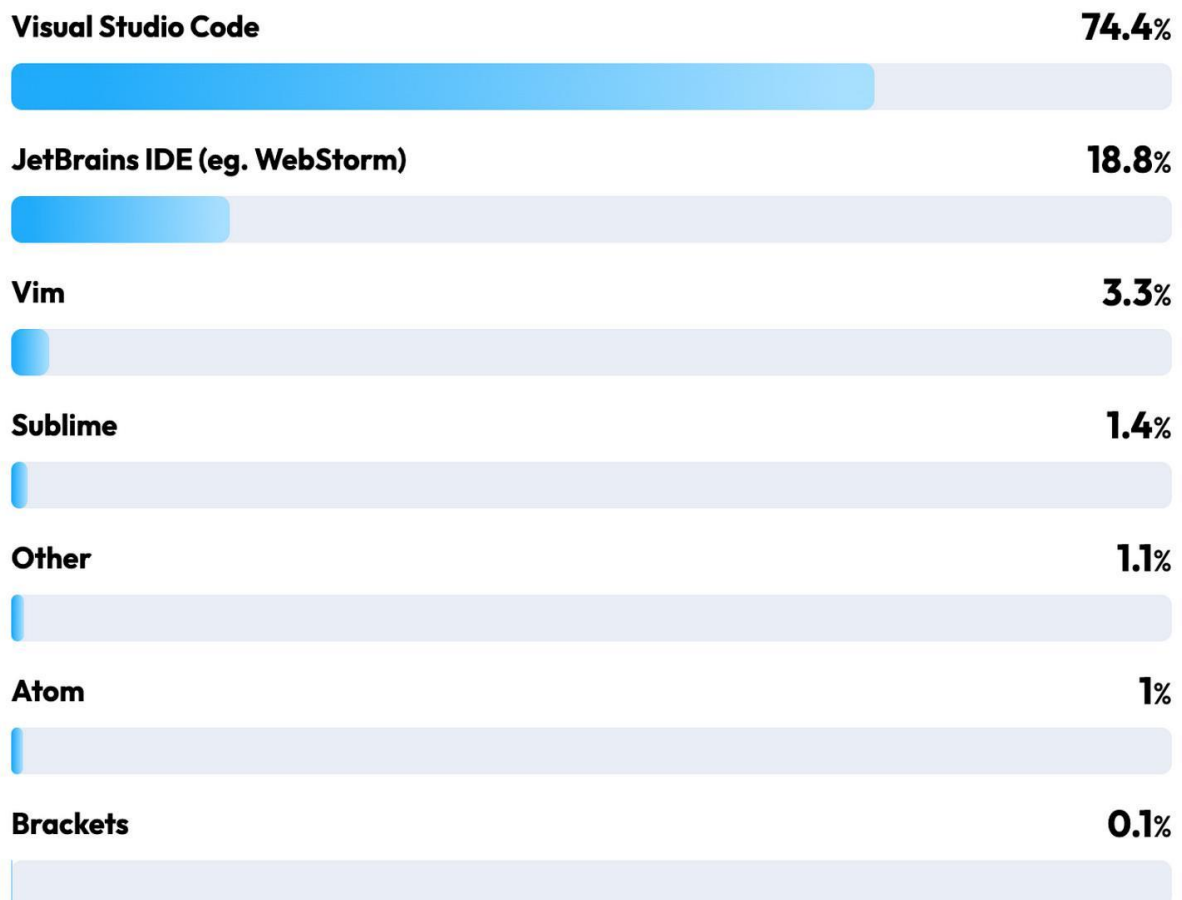
Redux is frequently used but generally disliked. Redux is a state container for JavaScript applications, and is often used with React. While it is a powerful tool for managing state, it can be complex and difficult to understand, which may explain why it is generally disliked by frontend engineers.

Overall, the survey results show that frontend engineering is becoming more mature, with a focus on maintainability, productivity, and a strong type system. The popularity of TypeScript, React, and Next.js, as well as the use of unit testing, CI/CD, and code reviews, demonstrate that frontend engineers are building for the long term and are concerned with the overall quality of their code.

5. Developer Tools

The use of Visual Studio Code (VS Code) as a development environment among frontend developers is overwhelming, with JetBrains trailing far behind in second place. This is not just a trend among frontend developers; VS Code is the most popular development environment globally, according to the Stack Overflow 2021 developer survey.

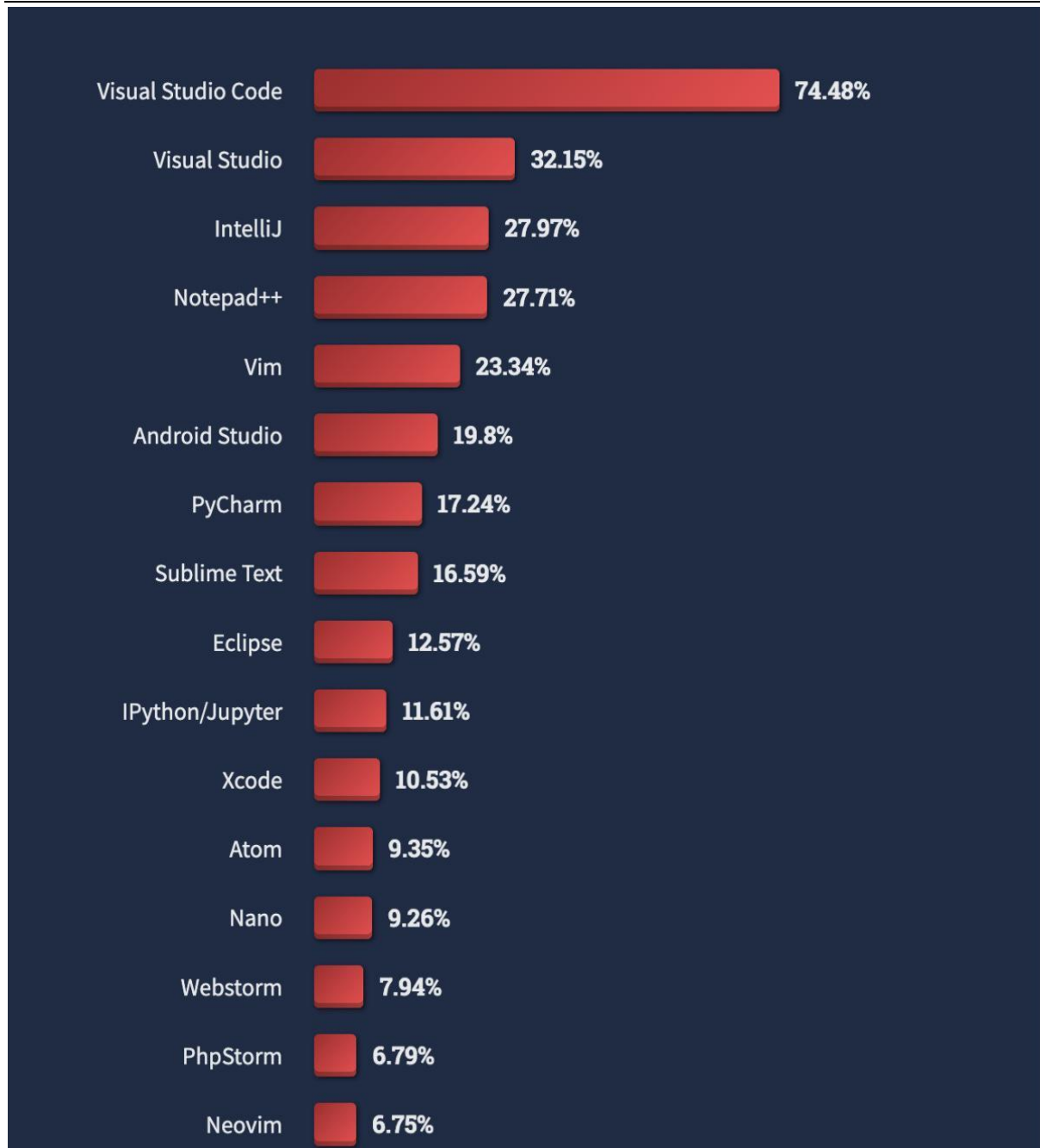
What's your favorite desktop code editor?



Pic. 8: The split of favorite IDE

The popularity of VS Code can be attributed to its powerful extensions system and its ability to provide a smooth and efficient development experience. This is

evident in the results of the Stack Overflow 2022 developer survey, where 71,000 responses showed that VS Code is the preferred development environment across all developers.



Pic. 9: The split of IDE popularity

As a developer who has used Visual Studio in the past, I can attest to its capabilities as a tool. In my experience, it was a great tool to use while working on the Microsoft stack, although its license costed between \$1,500 and \$5,000 per year. However, I never fully appreciated the extent of Microsoft's dominance in integrated development environments (IDEs) until I switched to other stacks.

I began developing for iOS using XCode and was disappointed with the poor development environment. I missed the powerful tools that Visual Studio offered such as refactoring with one click and conditional breakpoints. Furthermore, the performance of XCode was poor. The experience with Android Studio was not much better.

Jetbrains has consistently produced good IDEs, such as IntelliJ for Android development, but even their

tools did not match the level of developer experience offered by Visual Studio, even a decade earlier.

Microsoft's move to include the best features of Visual Studio in the free edition of Visual Studio Code has proven to be a smart move. The company might have cannibalized its own Visual Studio market, but the move has resulted in a wider adoption of VS Code and positioning itself as the #1 tool for developers.

Additionally, Microsoft's acquisition of GitHub has further cemented the use of VS Code as it seamlessly integrates with GitHub and provides a familiar IDE interface within the browser. This poses a significant competition to CodeSandbox and StackBlitz, which are popular online code editors among frontend engineers.

Big Tech companies are also moving towards the use of VS Code across software engineering. An example of this is Facebook, which moved from Atom to VS

Code as its company-wide preferred development environment

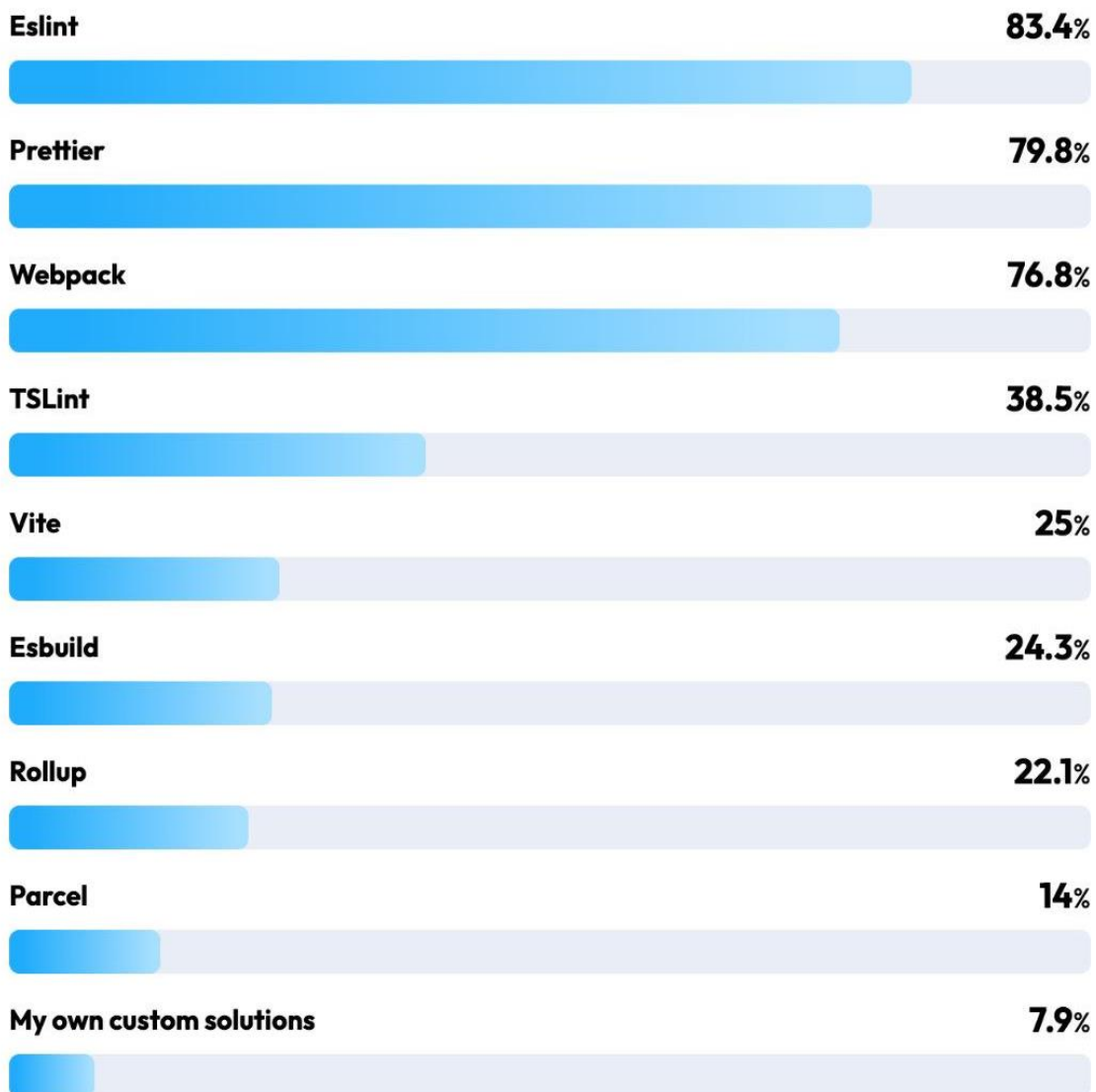
The trio of ESLint, Prettier, and Webpack are considered to be the "default" build toolset for frontend engineers. ESLint is a powerful static code analyzer that can be easily integrated with most code editors, including Visual Studio Code and JetBrains IDEs. Its purpose is to identify and flag potential errors and inconsistencies in the code, helping developers to write more consistent and maintainable code.

Prettier is an opinionated code formatter that works seamlessly with JavaScript, TypeScript, CSS, HTML, GraphQL, and Markdown. It helps developers by automatically formatting code according to a set of predefined rules, reducing the need for manual code formatting and ensuring consistency across the entire project. Prettier also supports all major code editors, making it a versatile and widely-used tool among frontend engineers.

Webpack, on the other hand, is a static module bundler that compiles various types of modules, such as JavaScript/TypeScript files, CSS/SASS, and Handlebars, into static assets. It discovers dependencies between these modules and performs clever repackaging to reduce runtime size and the latency of loading resources. Webpack also offers advanced capabilities, such as allowing dependencies to load asynchronously at runtime, which helps to reduce the initial loading time of the application.

When it comes to developer tools, it's interesting to note that less than 10% of respondents have built and used their own custom solutions. This is surprising considering that frontend engineering often lends itself to hacking and customizing tools. However, it also highlights the abundance of available tools for common development problems and the ease of using and contributing to existing tools when certain features are missing.

Over the past year, which of the following development tools have you used?



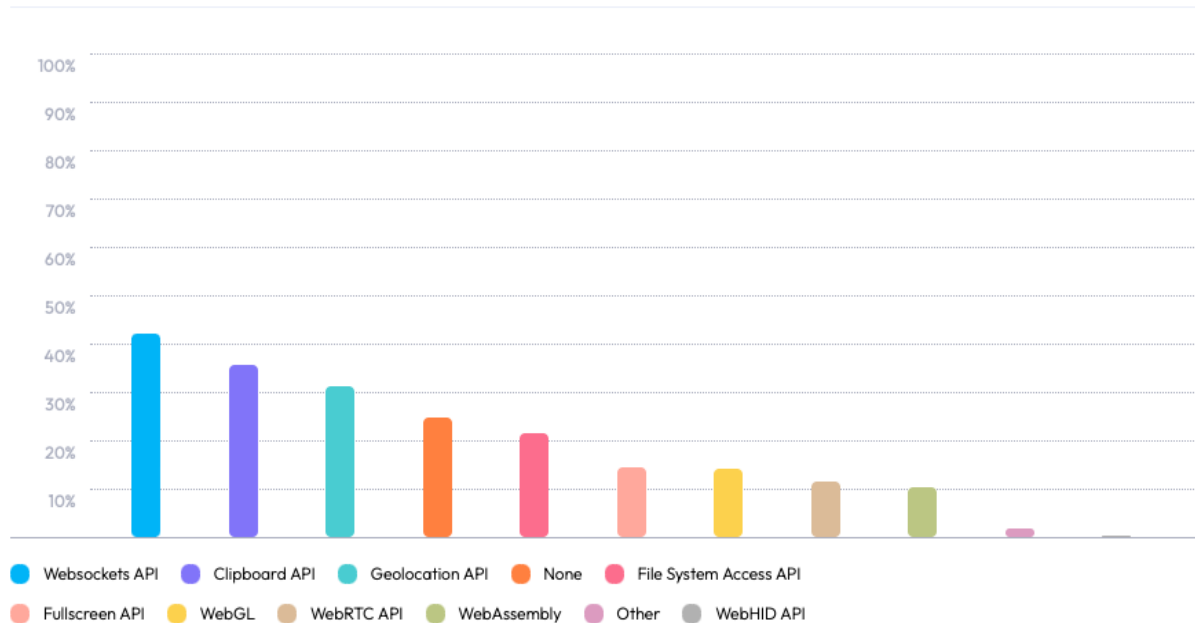
Pic. 10: The split of the most popular developer tools used.

6. Browser technologies

As a researcher in the field of browser technologies, I found the results of the survey on WebSockets usage to be quite intriguing. Specifically, the fact that

42% of respondents reported having used WebSockets, while I had initially expected this number to be much lower at less than 5%.

Over the past year, which of the following browser technologies have you used?



Pic. 11: The split of the most popular browser technologies used.

Upon further analysis, I have identified several possible explanations for this discrepancy. One possibility is that there may have been some sort of sampling error or misinterpretation of the survey question. Additionally, it is possible that the developers surveyed have used the technology in a business setting, on a production website, or have simply experimented with it in private coding projects.

However, after further consideration, I believe that the primary motivator behind the high usage of WebSockets is a combination of factors. Firstly, it appears that browser technologies are being used more frequently in general. Additionally, WebSockets are being utilized even in cases where real-time functionality is not strictly required, likely due to the increasing popularity of Firebase-like platforms.

Furthermore, the relative usage order of various technologies also seems plausible, with newer technologies such as the File System Access API not yet widely adopted. In particular, it would be interesting to investigate the extent to which sites utilizing the File System Access API are still falling back to older methods such as `<input type="file">`.

As a researcher, I also have a personal admiration for WebAssembly, as it represents a standardized bytecode and offers a wide range of potential use cases not just within the browser but also on server-side and offline apps. However, it is worth noting that while WebAssembly is a compilation target and is not intended to be written manually, it still requires further improvements, especially on the client-side in the browser.

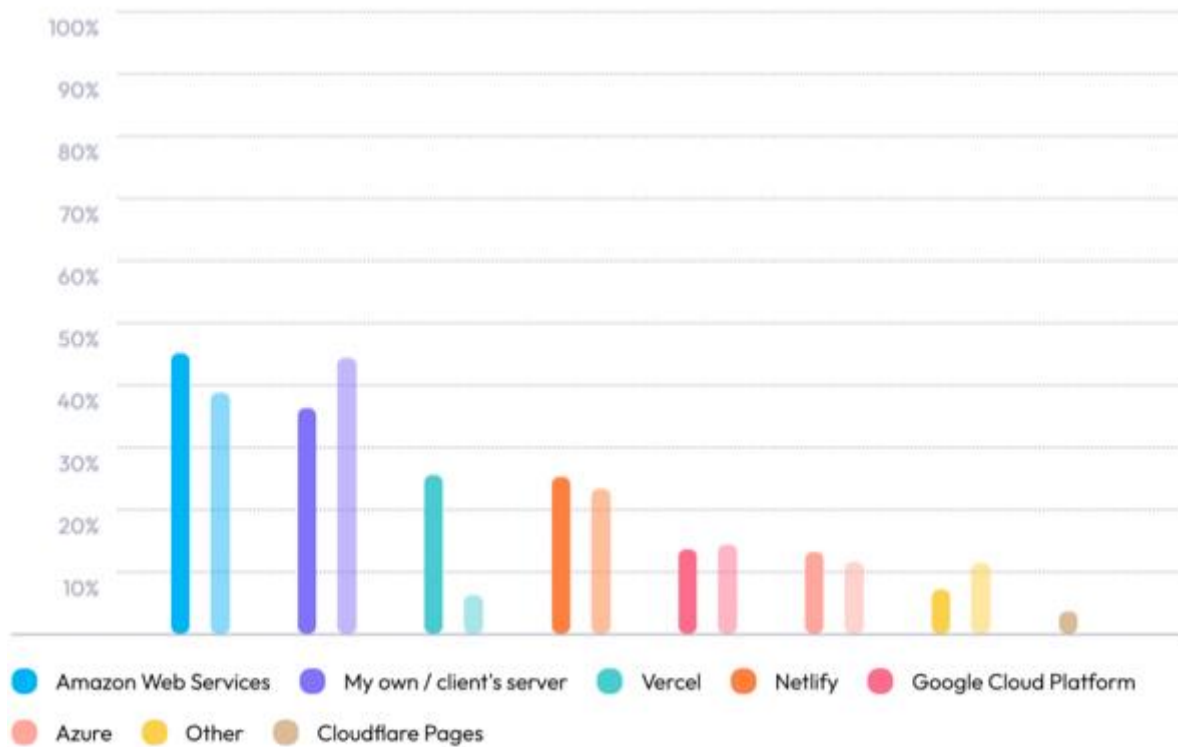
Furthermore, the popularity of languages used for compiling to WebAssembly is also worth investigating. I expect the most popular languages to be C/C++, Rust, and AssemblyScript, with an honorable mention for Golang due to its growing popularity in the WebAssembly community.

In conclusion, I believe that in the long run, the development of better tooling should make WebAssembly an implementation detail that most developers don't need to be concerned about, much like iOS developers don't often care about compiling to ARM. However, it should be noted that the process of standardization and community growth is a slow process, and it may be more than a decade before this becomes a reality.

7. Vendors

In the world of frontend development, vendors play a crucial role in providing the tools and services necessary for developers to create high-quality and efficient applications. As an independent analyst, it is important to note that I have no affiliations or sponsorships with any of the vendors mentioned in this report, and any opinions shared are based on my own observations and research.

One vendor that has recently gained significant momentum is Vercel. According to the 2022 developer survey, 25% of respondents reported deploying their code to Vercel, overtaking its competitor Netlify. This growth rate suggests that Vercel may soon become the most popular service for hosting frontends.



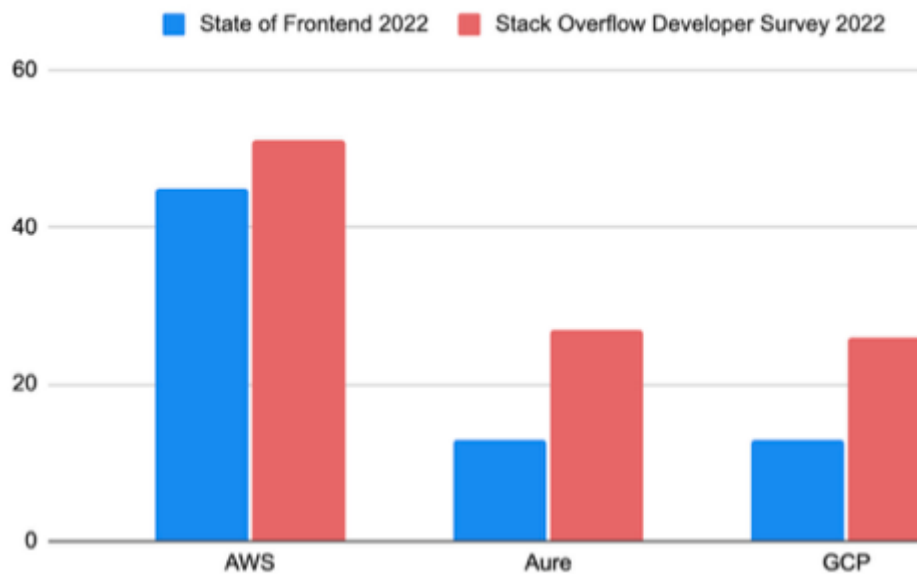
Pic. 12: The split of deployment providers.

This momentum can be attributed to the company's focus on edge computing, an area that former Google software engineer and current Vercel CTO Malte Ubl believes to be one of the most exciting developments in the tech industry. Since joining Vercel, Ubl has led the company in innovating on edge computing, such as with the launch of regional Edge Functions in October 2022. This type of frontend infrastructure-level innovation from Vercel is noteworthy as it is typically an area where companies like AWS or Cloudflare would lead the way.

Another notable development is the growing popularity of Next.js, a framework built on top of React by Vercel. The survey found that Next.js has become the second most popular framework behind React. Vercel has also recently released Turbopack, a solution they claim to be a "successor to Webpack" that is 700x faster than Webpack and 20x faster than Vite. However, these claims were met with skepticism as the creator of Vite later revealed that Vercel had cherry-picked an impractical use case to arrive at these numbers.

On the other hand, Gatsby, a static site generator startup that raised \$40M in funding, has failed to increase in popularity despite its heavy funding. The survey found that 17% of respondents used and disliked the framework, and 11% used and liked it, while 36% of respondents named Next.js as their favorite static-site generator and only 8% said it was Gatsby. The reasons for this lack of traction may be attributed to performance issues in 2020-2021, which likely deterred many developers from using the framework.

In conclusion, the developer tools and services provided by vendors play a critical role in the success of frontend development. While Vercel is currently gaining momentum and innovating in the field of edge computing, Gatsby's lack of popularity despite heavy funding highlights the importance of providing efficient and high-performing tools for developers. It is also worth noting that less than 10% of all respondents have built and used their own custom solutions for developer tools, further emphasizing the importance of well-established and reliable vendors in the industry.



Pic. 13: The split of the cloud platforms usage.

The use of cloud platforms for deploying code among frontend engineers is a topic of great interest in the software development industry. According to the State of Frontend 2022 and Stack Overflow Developer Survey 2022, AWS emerges as the clear winner as the most popular destination for frontend engineers and software engineers to deploy their code. However, it is worth noting that there is a significant difference in the number of respondents who named Azure or GCP as a platform they use, as opposed to the Stack Overflow Developer Survey.

One possible explanation for this difference is the growing popularity of Vercel and Netlify in the frontend hosting market. These services have made significant strides in recent years, and they could continue to grow in popularity among frontend engineers. Additionally, it is possible that some frontend engineers are not aware of the specific platform on which their code is being deployed. This could be particularly true for engineers working for clients, who may simply check in their code without considering the underlying infrastructure.

It is important to note that while AWS remains the most popular destination for deploying code, the rise of other platforms such as Azure and GCP should not be ignored. These platforms offer a range of features and capabilities that can benefit frontend engineers, and they may become increasingly relevant in the future. Furthermore, the emergence of newer players such as Vercel and Netlify highlights the dynamic and ever-changing nature of the technology landscape, and the importance of staying informed and adaptable as a frontend engineer.

Conclusion

In conclusion, the survey conducted with over 3,700 participants from 125 countries provides valuable insights into the current state of the frontend industry. The results indicate that JavaScript and React are popular among frontend engineers, Agile development methodologies are widely used, and the use of cloud-based platforms and third-party tools is increasing. The

survey also shows that the majority of frontend engineers work at software development companies, developer agencies, or tech-first or digital-first companies. The findings of the survey can help frontend developers, designers, and managers make informed decisions when it comes to choosing technologies, tools, and practices for their projects. Additionally, the survey's data on vendors highlights the platforms and third-party tools frontend engineers use for error tracking and monitoring. Overall, this survey serves as an invaluable resource for anyone looking to stay up-to-date with the latest trends and practices in the field of frontend development.

References

1. Github repository TypeScript / [Web resource], 2022. Access mode: <https://github.com/microsoft/TypeScript/#readme> (access date: 10.11.2022).
2. State of JS [Web resource], 2023. Access mode: <https://stateofjs.com/en-us/> (access date: 05.01.2023).
3. AWS: detailed description of the service [Web resource], Access mode: https://en.wikipedia.org/wiki/Amazon_Web_Services (access date: 21.12.2022)
4. Netlify: detailed description of the service [Web resource], Access mode: <https://www.netlify.com/products/> (access date: 19.12.2022)
5. The state of frontend 2022 [Web resource], 2022. Access mode: <https://newsletter.pragmaticengineer.com/p/state-of-frontend-2022> (access date: 05.11.2022)
6. The state of frontend with insightful comments from Netflix, AWS, and Cloudflare experts. [Web resource], 2023. Access mode: <https://tsh.io/state-of-frontend> (access date: 12.01.2022)
7. Next.js: blog posts, updates [Web resource], Access mode: <https://nextjs.org/blog> (access date: 19.12.2022)

8. WebAssembly: What Is WebAssembly - and Why Are You Hearing So Much About It? [Web resource], Access mode: <https://thenewstack.io/what-is-webassembly/> (access date: 29.12.2022)

9. Webpack: core concepts [Web resource], Access mode: <https://webpack.js.org/concepts/> (access date: 15.09.2022)

10. Vercel: develop, preview, ship / changelog documentation [Web resource], Access mode: <https://vercel.com/changelog> (access date: 03.12.2022)

ОБНАРУЖЕНИЕ МЕТАНА НА ОСНОВЕ МУЛЬТИСПЕКТРАЛЬНЫХ СНИМКОВ SENTINEL-2

Крауклит Г.В.

Магистр

*Азербайджанский государственный университет нефти и промышленности,
г. Баку, Азербайджан*

Агаева К.С.

Доктор философии по технике

*Азербайджанский государственный университет нефти и промышленности,
г. Баку, Азербайджан*

METHANE DETECTION BASED ON SENTINEL-2 MULTISPECTRAL IMAGERY

Krauklit G.,

Master

*Azerbaijan State University of Oil and Industry,
Baku, Azerbaijan*

Aghayeva K.

*Doctor of Philosophy in Technological Science
Azerbaijan State University of Oil and Industry,
Baku, Azerbaijan*

АННОТАЦИЯ

Метан является мощным парниковым газом, который при нормальных условиях не имеет ни цвета ни запаха, оказывающим значительное воздействие на изменение климата. Вклад метана в глобальное потепление составляет около 25% от всего потепления, наблюдаемого с доиндустриальных времен. Антропогенные выбросы метана происходят из множества различных источников, в основном связанных с сельскохозяйственной деятельностью, добычей угля, нефти и газа и обработкой отходов. Исследования, проведенные в этой области, показали, что некоторые из этих источников выбрасывают значительное количество метана из-за неисправностей оборудования или ненормальных условий эксплуатации. В статье описывается, как использование мультиспектрального прибора Sentinel-2 MultiSpectral Instrument (MSI) может помочь в эффективном смягчении последствий изменения климата, благодаря возможности быстрого обнаружения и частого мониторинга сильных источников метана для их последующего исправления с высоким спектральным разрешением (20 м) и быстрой частотой снимков (2-5 дней).

ABSTRACT

Methane is a powerful greenhouse gas that lacks both odour and colour and has a significant impact on climate change. Methane's contribution to global warming is about 25% of all warming observed since pre-industrial times. Anthropogenic methane emissions come from many different sources, mostly related to agricultural activities, coal mining, oil and gas extraction, and waste treatment. Studies in this area have shown that some of these sources emit significant amounts of methane due to equipment failures or abnormal operating conditions. This article describes how the use of the Sentinel-2 MultiSpectral Instrument (MSI) can help effectively mitigate climate change by enabling rapid detection and frequent monitoring of strong methane sources for subsequent remediation with fine resolution (20 m) and rapid imaging frequency (2-5 days).

Ключевые слова: Метан, Изменение климата, Дистанционное зондирование, Обнаружение плумов, Спутниковые данные высокого разрешения.

Keywords: Methane, Climate change, Remote sensing, Plume detection, High-resolution satellite data.

1. Введение

Дистанционное зондирование - это процесс идентификации, наблюдения и измерения объекта без контакта с ним [1]. Таким образом, дистанционное зондирование включает в себя акустический мониторинг окружающей среды, но этот термин

обычно используется в отношении измерения электромагнитного излучения. Глаза, как и камеры, являются дистанционными датчиками. В основе дистанционного зондирования лежит принцип, согласно которому все объекты взаимодействуют с падающим излучением: поглощают, отражают,