The State of APIs and Integrations at 400 SaaS Companies

pandium

pandium
integrate all the things

Table of Contents

Click on a chapter title to go to that chapter.

Introduction

People are using **more applications** than ever, and as a result, they are looking for better ways to integrate those systems.

Customers want to be able to seamlessly connect all their applications out of the box, and this has created a huge demand for product integrations.

But how are SaaS companies meeting this demand? And how are they working together to provide seamless interoperability to customers?

We examined 400 SaaS companies to assess what companies are doing in terms of app marketplaces, integrations, APIs, SDKs, and the tech partner experience.

Our data set looked at the 100 largest SaaS companies, 100 Series D, 100 Series C, and 100 Seed stage SaaS as we wanted to determine at what stage of development companies matured in terms of offering their customers and tech partners true interoperability.

By reading this book you can understand at what stage most companies are offering public API documentation, developer portals, SDKs, and other indices that mark a thriving tech ecosystem.

We also wanted to see what the relationship was between SaaS company size and maturity of tech ecosystem. The results on that were clear: there is a strong relationship between success as a SaaS company and both the openness and maturity of their tech ecosystems.

pandium
integrate all the things

Why Read This Report

This report will give you insight into what the SaaS market is doing at different stages when it comes to product integrations, APIs, app marketplaces, and the tech partner experience.

It also showcases how the 100 largest SaaS companies (the 50 largest public and 50 largest private) are approaching ecosystems.

For example, 84% of the largest SaaS have public API documentation and they provide customers with an average of 621 product integrations.

This dwarves earlier stage companies in terms of their openness and maturity as integrated tech ecosystems.

In addition, this report will show you what successful companies are doing when it comes to tech partnerships and external APIs: are they publicly documenting their API? Making it easy to get a developer account? Providing a developer portal and sandbox?

Many SaaS companies sense that product integrations, app marketplaces, and external APIs are important, but this report provides data showing that, if you want to be a successful SaaS company, unless you belong to a few niche product categories, they are now essential.

Key Takeaways From This Report

1. Providing a **minimum of dozens of product integrations** are now required for success in SaaS in most product categories. **Product category matters**.

2. In-app and public marketplace experiences help customers to discover and manage their integrations, **benefiting the host and partners**.

3. **Few app marketplaces are transactional**. But the ones that are tend to be large platforms in both a business and technical sense. Monetization models are still in flux.

4. **REST has taken over SaaS external APIs**. Developers are familiar with this style. Deviate only if there is good reason to.

5. **Webhooks are valuable** for efficiency and real time event notifications. Support them for the most important event triggers earlier rather than later.

6. **Implement OAuth 2** for partners as soon as possible. It increases visibility and is more secure.

7. Don’t just aim to become a technical platform that everyone else builds on. **Aim to be interoperable** from day one in the way that most benefits the product category and customer base.

8. For scale, reduce friction in the **partner developer experience** - public API documentation, easy access to trying the API, and a well-designed developer portal.

9. There’s a strong correlation between open ecosystems and SaaS growth.

A Note on Methodology

For the purposes of this report, product integrations and apps includes a pre-built means of programmatically passing data between two systems or an extension built by a third party on top of another system that the user or customer of the software can install and utilize.

Product integrations and extensions are different from custom or private integrations in that they are “productized” and thus available for use to at least a segment of, if not all, customers.

The information collected in this report is only based on publicly available resources, from websites, support docs, press releases, and information the company provided elsewhere, like on review sites. Information that could be obtained from a company with only an email login was also included.

When a company claimed to have product integrations, but didn’t identify most of them, they were excluded from the analysis of the number of product integrations.

Similarly, if a company did not have publicly available support documentation, product videos, or any comment on whether they had an in-app marketplace, they were excluded from the analysis.

For the full methodology, consult the **Appendix** of the report.

1. Product Integrations and Marketplaces

pandium
integrate all the things

In 2020, we analyzed the 1000 fastest growing SaaS companies to see how they were approaching product integrations and tech partnerships.

We found those companies provided an average of 98 product integrations. The median number was 15 integrations. In comparison, the 400 SaaS companies we analyzed in 2022 have an average number of 189 product integrations. The median number is 20 integrations.

This comparison is not one-to-one. The 1000 fastest growing SaaS companies are a different data set than 400 SaaS companies we examined this year, which includes the 100 largest SaaS, 100 Series D, 100 Series C, and 100 Seed companies.

We wanted to examine SaaS companies at different stages of growth to both understand at what stage companies become maturely interoperable and, also, to unpack what the largest, most successful SaaS companies are currently doing.

For size comparison, the 400 SaaS companies had a median number of employees of 295, while the median number of employees for the 1000 fastest growing SaaS was 158. We included the employee number distribution of both data sets in the Appendix for reference.

Even though these data sets represent different populations, we thought noting the comparison might speak to market changes as well as the interoperability of fast growing companies versus average companies of different growth stages.

Ultimately, the numbers demonstrate the largest, most successful SaaS companies offer more product integrations than other SaaS companies. And the fastest growing SaaS companies appear to provide more than their similarly sized peers, though that is tougher to say for sure due to different data sets being used.

To break it down by stage of growth, we found the 100 largest SaaS companies have a median number of 2406 employees, an average of 691 product integrations and a median of 94 integrations.

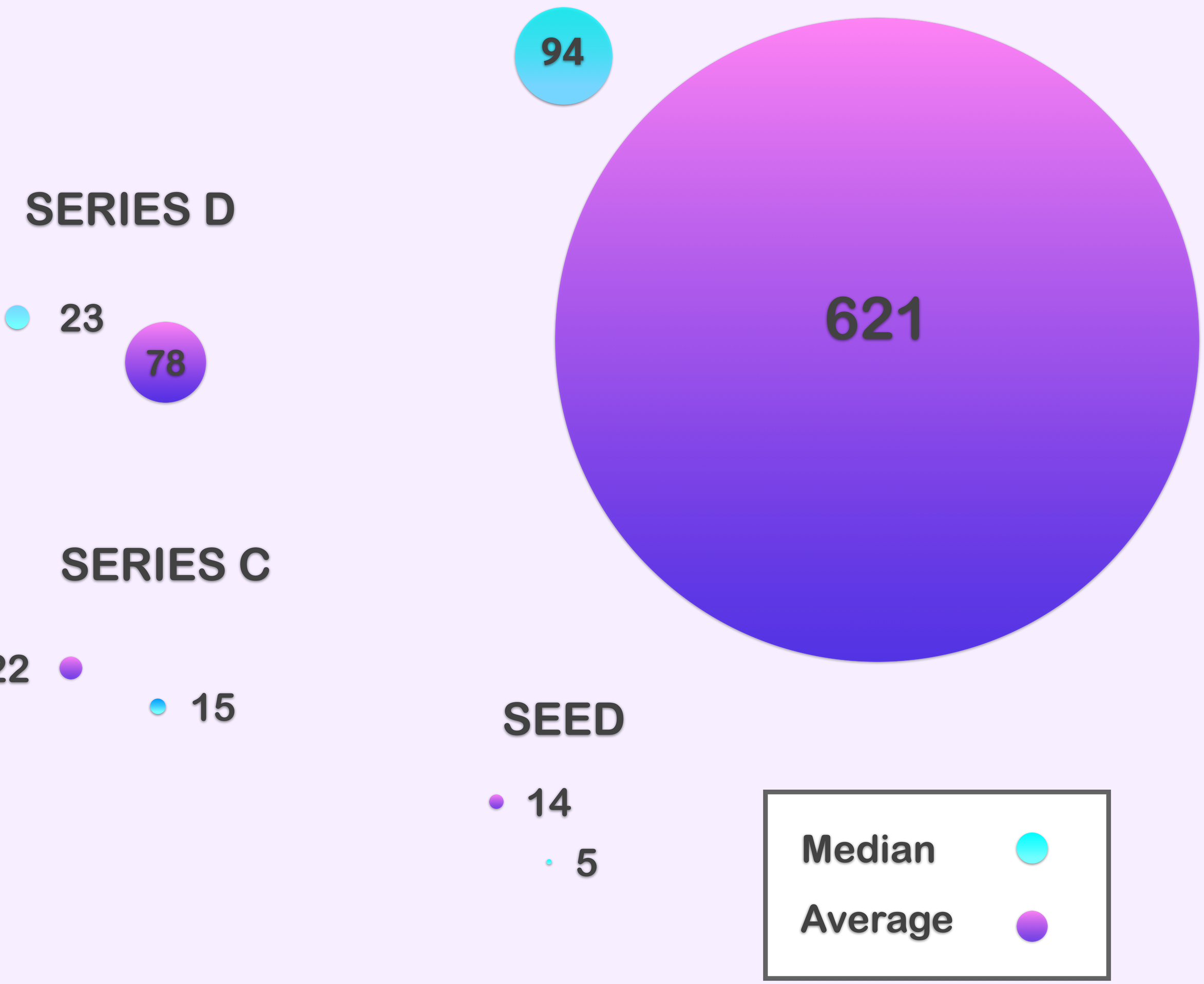
The Series D companies have a median number of 330 employees, an average of 78 product integrations and a median of 23 integrations.

The Series C companies have a median number of 232 employees, an average of 22 product integrations and a median of 15 integrations.

The seed companies have a median number of 25 employees, an average of 14 product integrations and a median of 5 integrations.

There is a strong correlation between the stage of company and how many product integrations are offered to customers. The largest companies in terms of stage, in general, provide a very large number of integrations, while earlier stage companies provide correspondingly fewer.

Product Integrations Offered



Product Integrations Offered

Median
Average

SERIES D (2022)

Median Employee Number: 330

23

78

100 LARGEST (2022)

Median Employee Number: 2406

94

621

SERIES C (2022)

Median Employee Number: 232

22

15

1000 FASTEST GROWING SAAS (2020)

Median Employee Number: 158

15

98

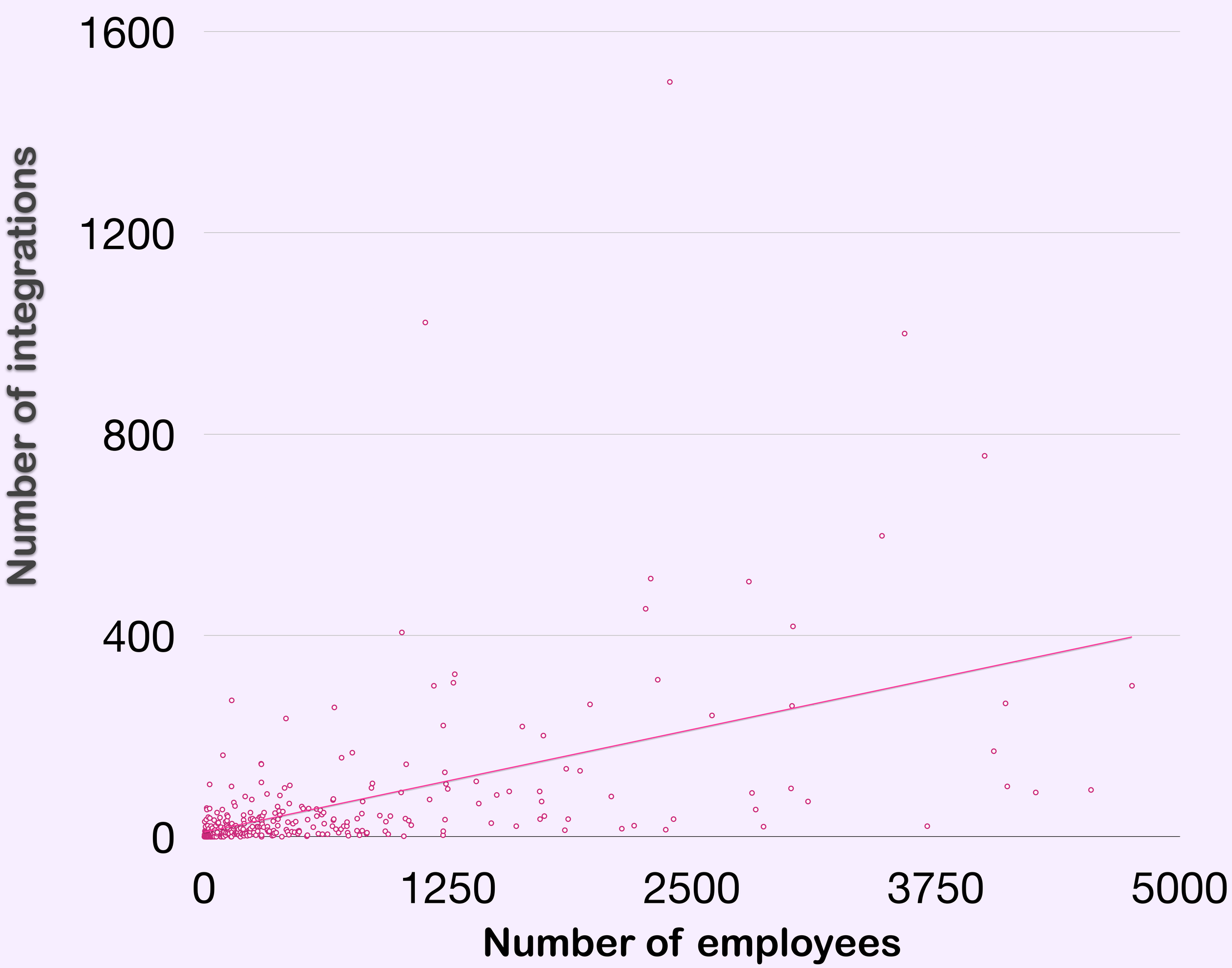
SEED (2022)

Median Employee Number: 25

14

5

Product Integrations Offered by Number of Employees at Companies with Under 5,000 Employees



Not all of the 100 largest SaaS companies have a hundred, or even dozens, of product integrations. The biggest reason why successful companies would offer fewer integrations is that integrations provide less business value to their product category or customer base.

The primary reasons successful companies may need to provide fewer integrations:

- Dealing with highly regulated or protected data, such as finance or healthcare (i.e. Change Healthcare)
- A vertical whose current systems remain less interoperable, such as education or manufacturing (i.e. Guild Education and ServiceTitan)
- Customer base whose other systems are highly bespoke or custom and thus require significant custom integration
- Small business or consumer customer base where fewer other systems are being used that have a relationship to the product (i.e. Current)
- Tool that is designed to be used from inside or on top of a limited set of other systems (i.e. Grammarly)
- API or developer product that is designed for developer to use as a building block in their own stack (i.e. Akamai)

For example, **Scale AI**, a top 50 public SaaS company, uses AI and machine learning to label and analyze data through APIs, and its main integrations are to cloud databases that their largely enterprise customers can use to connect their data to Scale.

Grammarly, a top 50 private SaaS company, is another product that is successful but has relatively few integrations at 12. Grammarly, which is an app that corrects users' writing, is designed to work as an extension overlay of other systems, such as Gmail, Microsoft Office, browsers, Google Docs, or Salesforce.

It integrates with those types of systems, but is then used from within those interfaces, and there is little need for the user to have it integrate further. Integrating with twelve of those types of systems covers a large portion of the market for digital writing correction.

Only recently has Grammarly (a company with 800 employees) moved to help extend the reach of its product and its target market by releasing an API and encouraging third party developers to embed Grammarly in their product.

Grammarly Developer is still in beta, but it's easy to see how, due to their product category, they were able to be successful with less than two dozen integrations. But the next stage of growth entails more interoperability.

In contrast, **Envoy**, a workplace platform that enables collaboration, is a Series C company that has less than half the number of employees of Grammarly, but provides customers with 85 product integrations. The nature of workplace coordination and collaboration requires integrations to be efficient and effective.

Despite other variables limiting the need for product integrations, it is notable that the vast majority of large SaaS companies do have at least dozens of product integrations.

This is suggestive that increased interoperability increases both the scope and market size for a SaaS product, and thus when we look at the largest SaaS companies, we generally see companies that are highly interoperable.

Talkdesk, for example, is an enterprise contact center software and top 50 private SaaS company. It started out its journey as a company heavily integrated with Salesforce and reliant on that customer base. But as it grew, it expanded its integration suite to expand its reach to customers using other CRMs, such as ZohoCRM, Microsoft Dynamics, Pipedrive, and Keap.

pandium
integrate all the things

Product Category and Product Integrations

We analyzed the median and average of integrations for product groupings where we had at least 10 companies in the category. We excluded the Seed companies from this analysis because so many of them provided zero product integrations, or lacked any public documentation, and we did not want it to skew the results for the product categories.

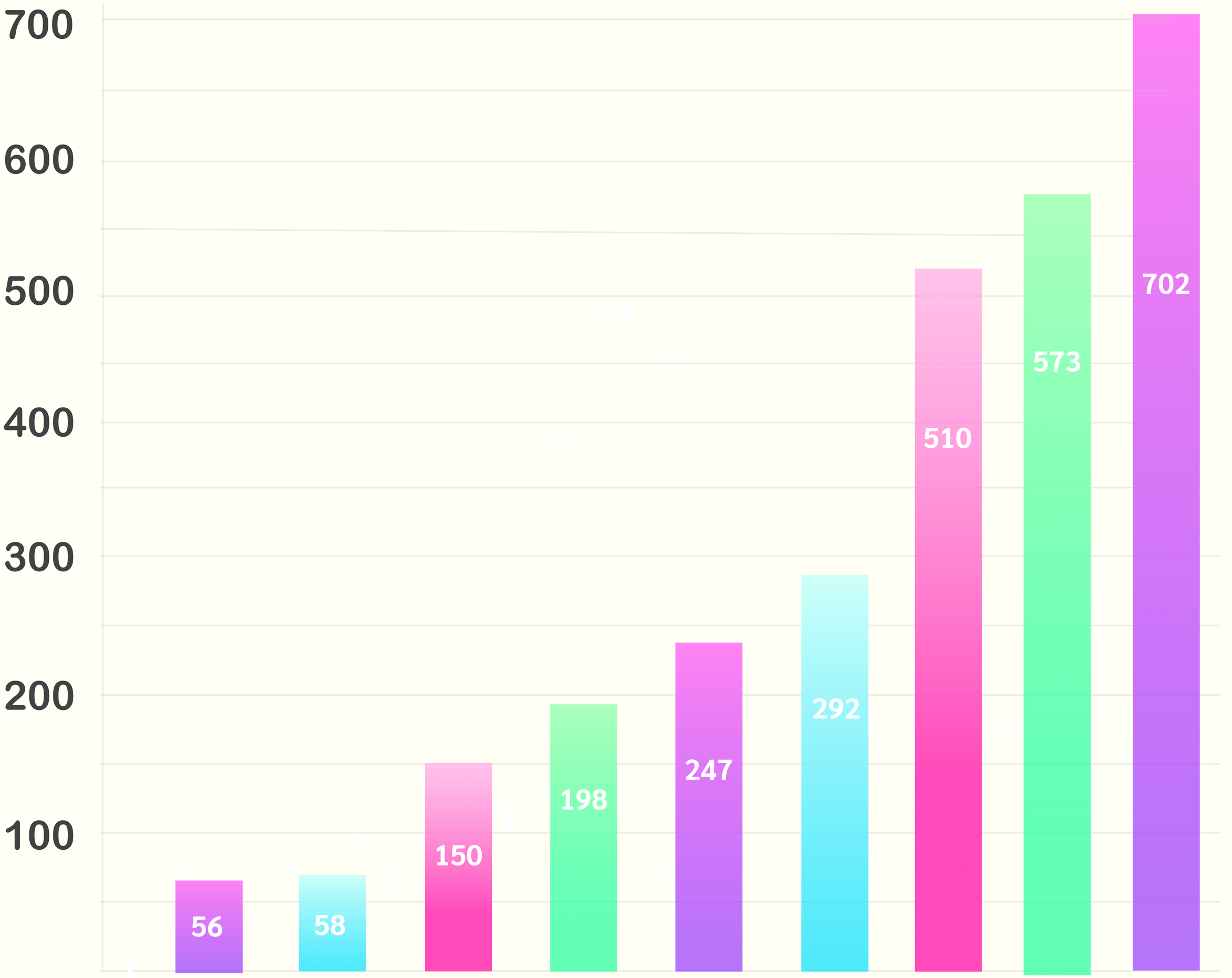
The 300 companies from the largest 100, Series D, and Series C revealed a significant correlation between product grouping and the number of product integrations provided. We only included product groupings that had at least 10 companies in that category.

We found product categories like commerce, sales, and marketing, with their users using dozens or hundreds of apps in those categories, had a higher median and average number of integrations.

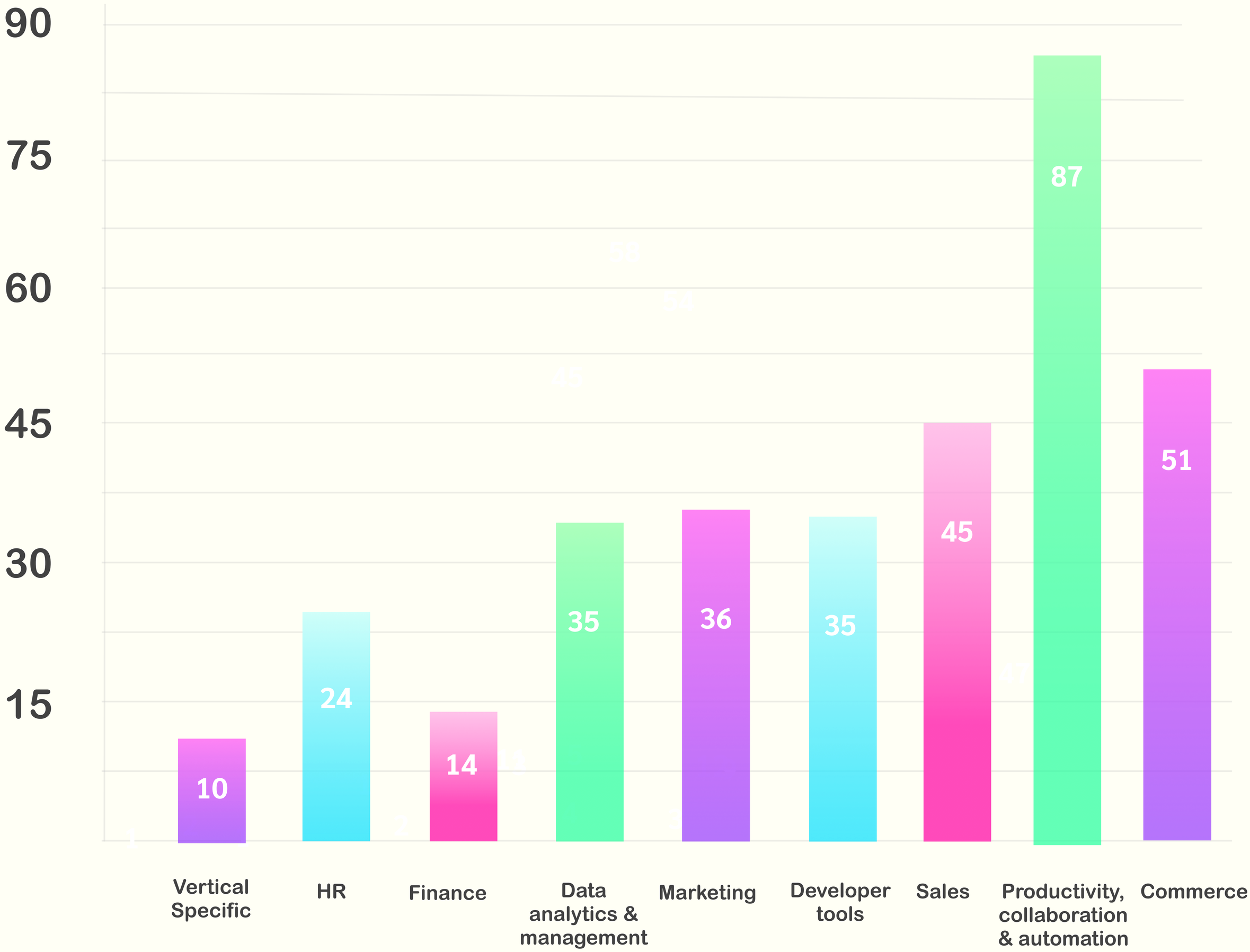
In contrast, highly regulated industries like finance, had a lower number. As did HR, which has been slow to integrate its systems, partly due to the sensitivity of the data HR systems handles, and partly due to the function not adopting technology as widely and as early as other business functions.

Vertical specific software also had a lower number of integrations. Vertical specific software, such as software for barbers, real estate agents, truckers, and so on, often are dealing with users who do not as have many other systems in operation. Many of the other vertical-specific systems they are using may also be less interoperable, making it more expensive to build integrations that also have less functionality.

Average Number of Integrations by Product Category at 300 SaaS Companies



Median Number of Integrations by Product Category at 300 SaaS Companies



Public and In-App Integration Marketplaces

Integration marketplaces are also known as app stores, app centers or app directories.

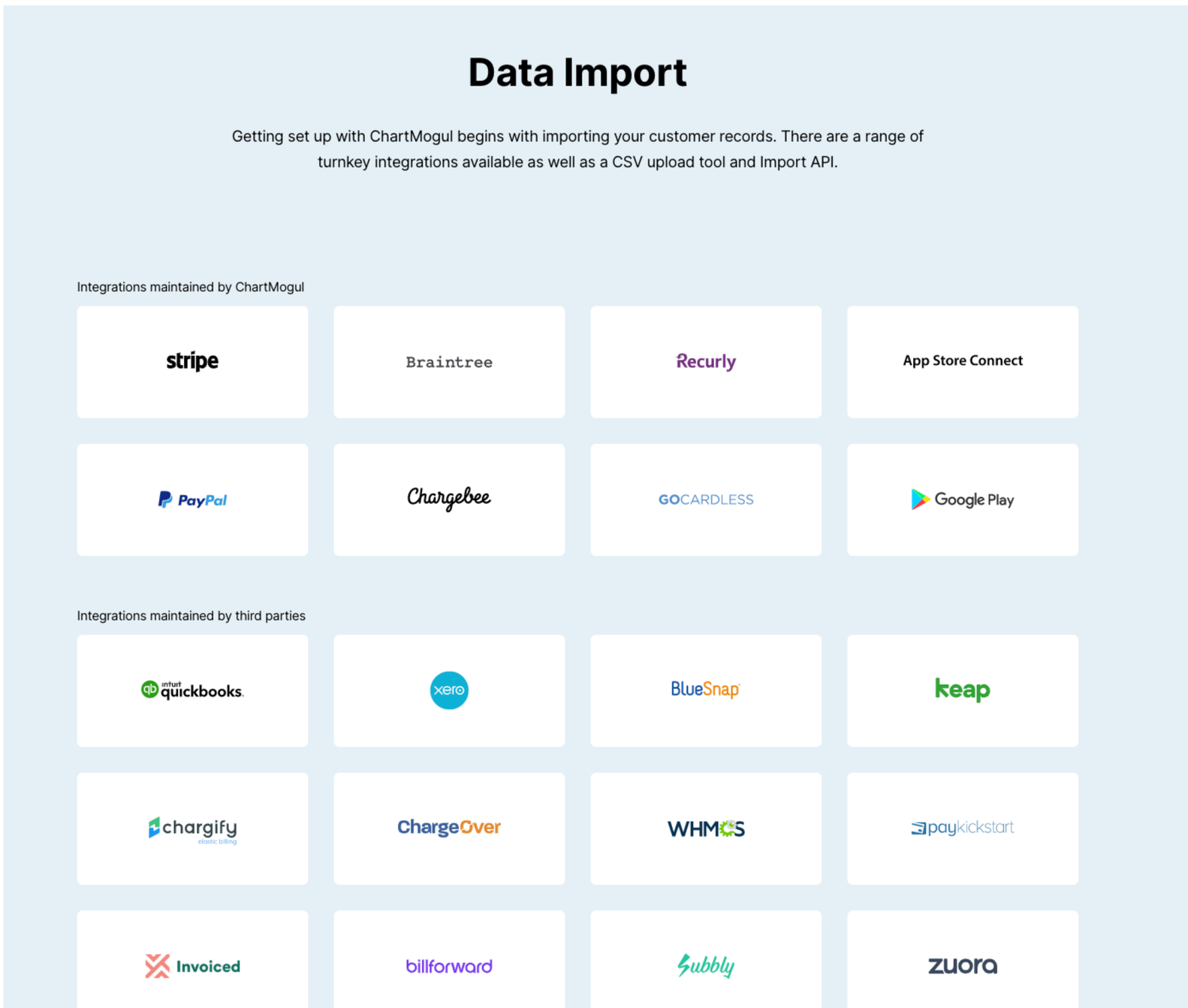
Almost all app centers have a similar style of rectangular tiles for each integration. Sometimes service partners or extensions will also have tiles in the center, though most companies put service partners in a separate directory.

Public Integration Marketplaces

The most basic public app center will just display the name of the software the integration connects to, and perhaps link to their website or explain what the company does.

For example, this public integration marketplace by ChartMogul, a seed stage company, lists all the integrations, but when you click on one, it just prompts you to log in rather than provide any further information on the integration.

Credit: ChartMogul



As a public app center becomes more sophisticated, there will be additional features, including:

- An integration page describing the integration itself, as well as offering pricing and other relevant information, like how to install it
- Search
- Filtering by product category, vertical, review scores, language and/or pricing
- Integrations flagged as partner built, native vs through an iPaaS, certified or premier
- Showcase area for featured or popular apps
- Customer reviews of the integration itself
- Information on number of installs for individual apps
- Different sorting mechanisms, such as by popularity, rating, and release date

Credit: Salesforce

In-App Integration Marketplaces

In addition to public integration marketplaces, companies often also offer in-app marketplaces. In-app marketplaces are different than public marketplaces in that only logged in customers can view them.

In-app marketplaces are technically more complicated because they require recognizing the individual user and then surfacing information relevant to them, and enabling them to install integrations on their account.

Sometimes the marketing content on a public and in-app marketplace is the same, but other times, the in-app marketplace is more focused on integration functionality and installation instructions, while the public marketplace focuses on higher level value propositions.

Functionality that is added to an in-app marketplace includes:

- Allows customer to click to install when possible; otherwise, the ability to click to install through a flow that occurs elsewhere
- Displays integrations the customers have already installed
- Displays integrations based on who the customer is, which may include being on a plan level, admin or user status, or fitting a particular user profile that makes some integrations more relevant to the user than others
- Displays integration activity and errors
- If transactional, enables customer to purchase the integration and/or the app the integration connects to
- Specifies how to get support for each integration

Because in-app marketplaces are more complicated to build, SaaS companies usually offer a public marketplace before they provide an in-app marketplace.

However, in-app marketplaces are important to customers being able to discover and find integrations that they would benefit from at scale. It also can enable easy management of currently installed integration.

At least 73% of the 100 largest SaaS companies offer an in-app marketplace to customers. 86% of those companies offer a public marketplace.

There is a strong correlation between having a public marketplace and having an in-app marketplace, but at every stage, more companies have a public marketplace than have an in-app marketplace.

This gap is even bigger at Seed stage companies, where only 8% of companies have an in-app marketplace, compared to 31% having a public marketplace.

31% of seed stage companies having a public integration marketplace when Seed companies generally have a median of 5 integrations shows that even early stage businesses see the value of integrations to customers - even if they can't yet deliver many integrations.

In contrast to this data set, in 2020, 63% of the 1000 fastest growing SaaS companies had a public app center. This number is slightly higher than the Series D companies in this data set - 58% of them have a public app center. This may show how the fastest growing companies have a greater focus on interoperability and providing integrations than the typical company of a similar size.

Transactional App Marketplaces

SaaS transactional in-app marketplaces typically sell four different types of products:

- Integration - the customer pays for an integration between the host system and another system
- Partner system - the customer may (or may not) be able to use the integration for free, but has to (or can) pay for the partner system through the host (the host than provides the partner with their cut of the deal, usually around 80%)
- Extension - the customer pays for an add-on, widget, or extension that is built on top of the host's system
- Asset - the customer pays for an asset like a template, dashboard, image, video, or design that can be used inside the host's system

Some marketplaces sell one of these types of products, while others sell all four. Zendesk, for example, sells integrations, extensions, and assets (templates). HubSpot sells assets for its CMS, and even though it has over 1000 integrations and extensions, it does not process the transactions for those products.

Only 14% of companies in the 100 largest SaaS have transactional marketplaces. 5% of Series D companies do. Only 1 Seed company does, and that company is a Seed stage extreme outlier with 2,325 employees.

So why are so few marketplaces transactional? Part of the reason may be integration marketplaces are relatively new and the business model is in still in flux in terms of what partners will accept and what most benefits customers.

Another reason is that it’s a significant technical and legal undertaking to process payments on behalf of partners. It requires a real confidence and commitment to the benefit of having a transactional marketplace to invest in the infrastructure around processing payments, issuing licenses, and remitting payments.

When it comes to selling partner systems and integrations between systems, it is also a zero sum game. There can only be so many SaaS companies who will be able to sell partner systems at scale (which is when it makes the most sense to invest in the payment infrastructure).

From the customers’ perspective, having one bill for multiple SaaS systems can be convenient and streamline the purchase process as well as the work of maintaining ongoing payments. But by nature, this benefit only exists in consolidation around platforms.

Merchants may be happy to pay Shopify for 5 other systems to streamline their billing, but if enough merchants are streamlining billing through Shopify, they can’t also streamline it in significant number through, say, their loyalty app that integrates with Shopify.

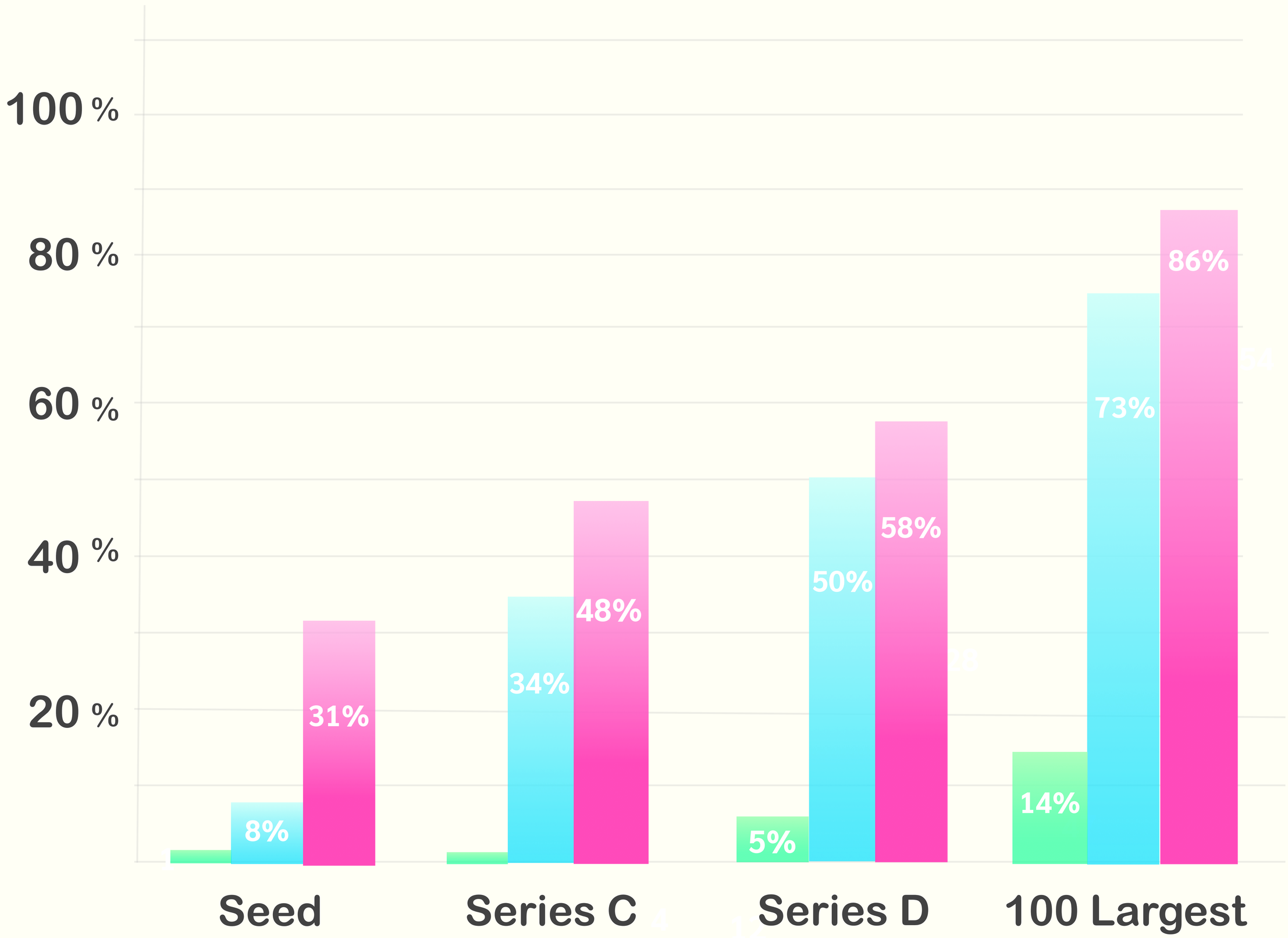
Assets and extensions are different than integrations and partner systems as they are built to enhance the host system. There is no possibility of streamlining billing elsewhere.

Extensions usually require significant technical infrastructure, though, which is why they tend to exist on top of larger SaaS companies. Assets may not as they may only require minor customization to the system.

For both extensions and assets, a SaaS company needs a large enough customer base to attract partners and third parties to learn their system and build an asset or extension for it.

This limits the value smaller companies can expect to see by enabling partners to sell assets for their system. But in theory, every SaaS company could enable partners to sell assets or extensions to their product.

Percentage of SaaS with an Integration Marketplace



pandium
integrate all the things

External APIs for SaaS are those available to external parties - customers, partners, and/or third party developers - to use to either get data out of a SaaS system or push data back into the SaaS (usually both).

External APIs are the bedrock of SaaS interoperability and they power the vast majority of product integrations between systems.

For the rest of the book, when we say API, we are referring to external APIs.

The terms Open API and Public API are often used, but they do not apply to most SaaS companies' external APIs. Open APIs are generally available to any developer with minimal registration and usually rely on publicly available data, while SaaS external APIs are usually moving customer data.

Similarly, Public APIs require minimal registration to use. Under this definition, a minority of external SaaS APIs - usually SaaS companies that encourage third party developers to build integrations and extensions - are also Public APIs.

However, many SaaS companies have publicly available documentation of their external APIs, and some might refer to a Public API as one that is publicly documented, even if one can't use it without being a customer or partner.

While only 24% of Seed companies have a publicly documented API, 63% of Series C, 68% of Series D, and 84% of the 100 largest SaaS do. In contrast, 57% of the 1000 fastest growing companies in 2020 had public documentation of their APIs.

pandium
integrate all the things

The Importance of Public Documentation

The fact that the majority of Series C and larger companies have public documentation demonstrates how critical integrations and APIs are to companies' growth strategies.

Publicly documenting APIs opens the doors for prospects, customers, partners, and third party developers to scope out the possibilities for an integration or extension and determine whether they can be used to accomplish their objective.

It is also an expectation for many developers. Developers want reduced friction, and they do not want to have to apply as a partner just to understand what your APIs are capable of doing.

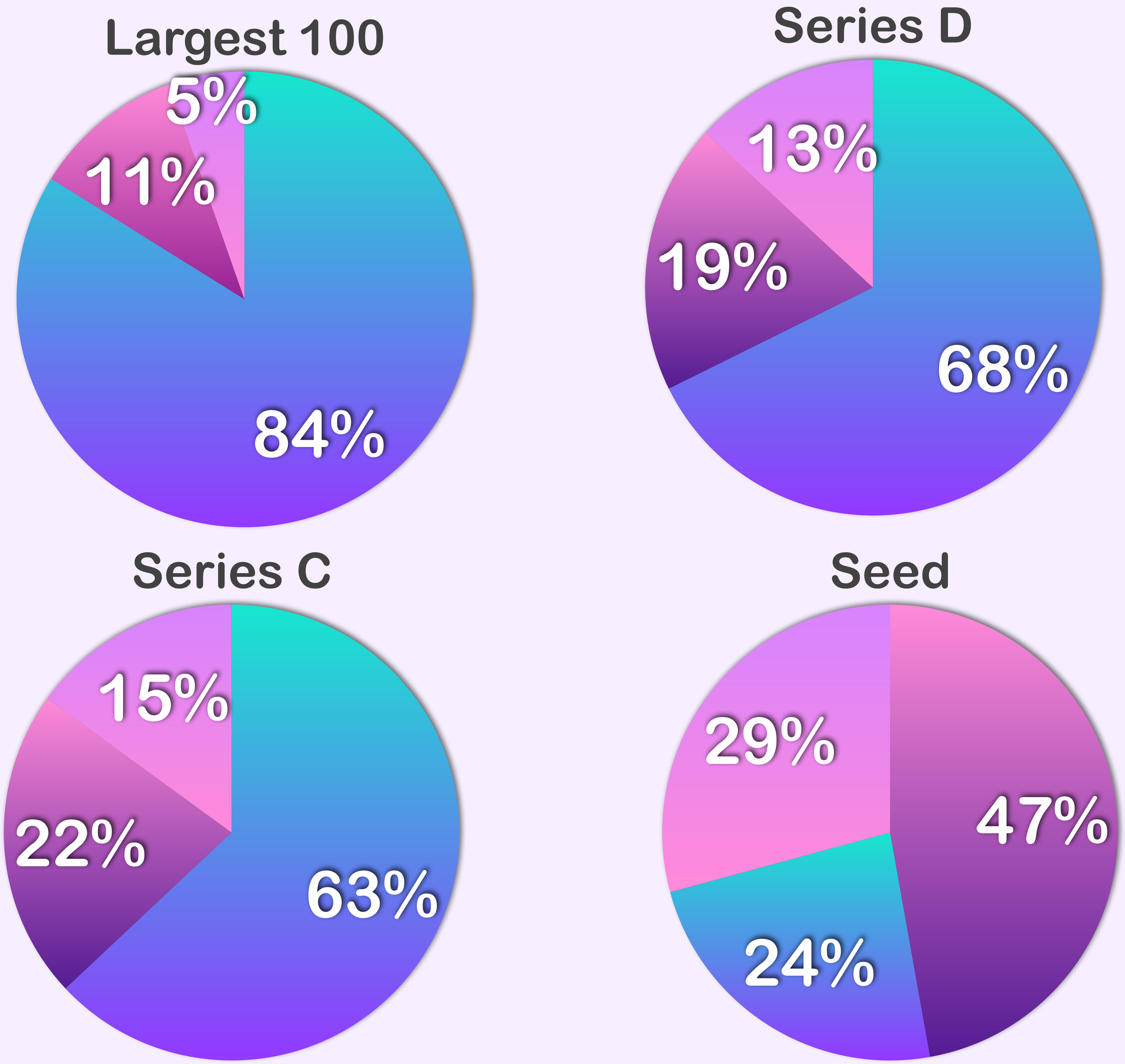
Not having public documentation can greatly reduce interest from partners and third party developers in building an integration. It shows a lack of ecosystem maturity, and is usually a red flag in assessing how difficult the system will be to work with as a partner.

This is not only because it is creating friction in step one of a long process, but also because the lack of documentation could be a signal that the company knows the APIs are poorly designed, poorly versioned, or poorly maintained and doesn't want to try to document them publicly for that reason.

Good partner developer experience starts with thorough and up to date API documentation. Unless there is a compelling reason not to, make your documentation available to all. (And if you have a REST API, tools like **Swagger** and **Readme** make it easy to launch and maintain documentation.)

Public documentation will result in increased integration builds, and a wider scope of feedback from the community of potential partners and builders.

Percentage of SaaS with Public API Documentation



pandium
integrate all the things

REST: API Style Dominance

Most of us are aware that REST APIs are very common, but this data set showed they are overwhelmingly dominant amongst external SaaS APIs.

Of these 400 companies, 246 companies had publicly named external API styles. This was 89 companies from the largest 100, 74 companies from the Series D, 65 companies from the Series C, and 18 companies from the Seed. A number of Seed companies likely do not have an external API at all (29 did not mention having one).

Of the 246 companies that publicly disclosed the style of their external APIs, 168 offered REST only. 47 offered REST plus other APIs in other styles. 14 offered HTTP only (many of which are likely REST).

So of 246 companies, 215 offered a REST API (and that is likely closer to 225).

Only 5 companies offered GraphQL only.

This is a strong degree of consistency across companies which represent a wide variety of verticals and product categories.

Why is this?

To provide interoperability to customers, a company has to be interoperable with a wide variety of other companies and encourage partner developers to build to their system.

At some point, REST became so dominant that it now reduces friction for developers to provide a REST API. More developers are familiar with REST than any other style.

If a company picks a different style, unless it is for a niche industry, fewer developers will know how to work with the API and they will have to invest more time to learn how to work with it.

Partner developers rarely have a full time job of building and maintaining an integration to one particular host system. As a result, it is imperative that the process is as easy and frictionless as possible.

Outside of niche industries, there should be a strong business justification for using a different style of API, if it is not in addition to a REST API and will be the only available external API.

Multiple API Styles

28% of the largest SaaS companies offered REST APIs as well as other API styles, such as SOAP, GraphQL, and JavaScript.

Some of these APIs were likely holdovers from systems of previous eras, where, SOAP, for example, was more common and is still being used in certain systems.

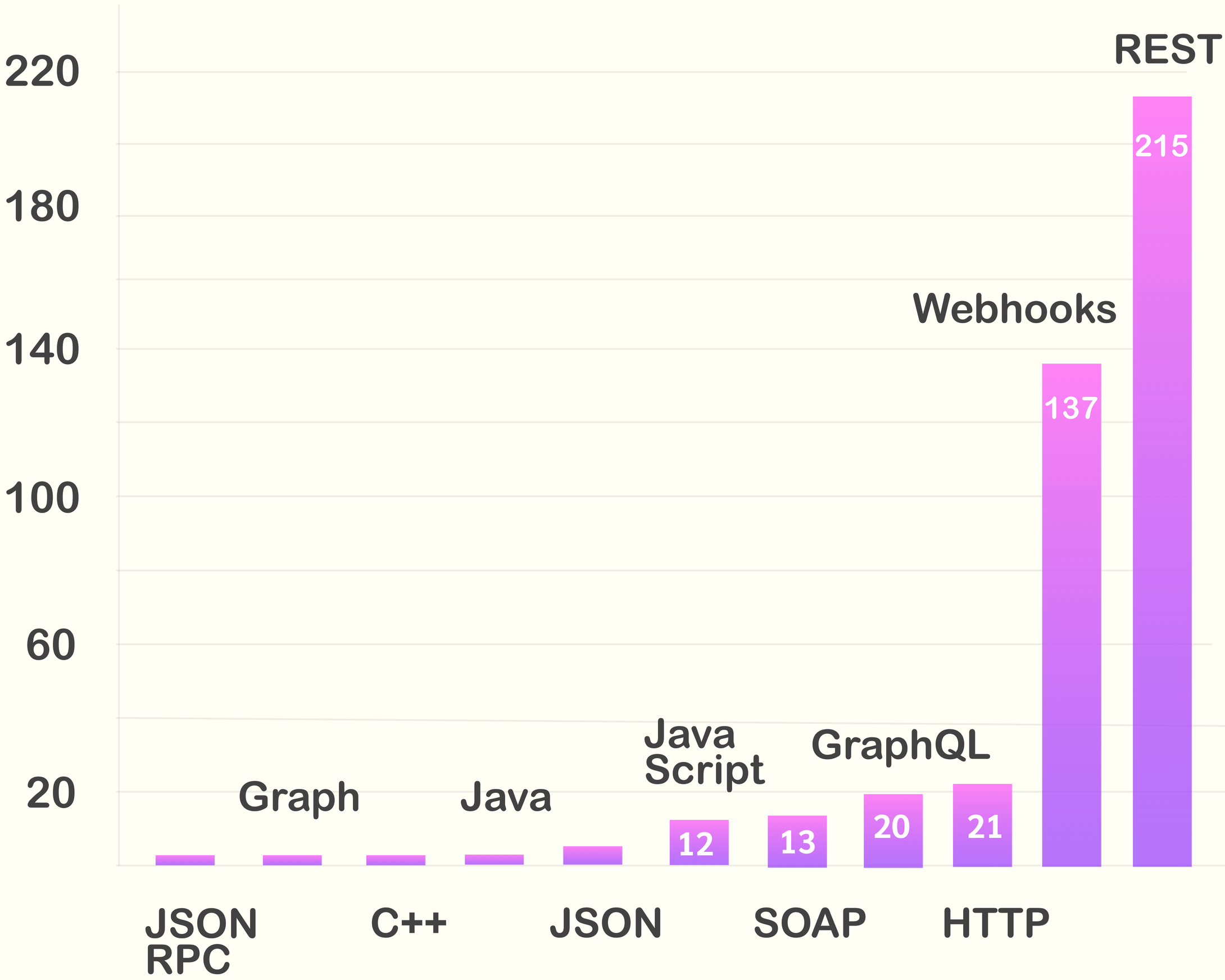
Others, like GraphQL, are designed to provide more flexibility to developers who may wish to use REST for some use cases and GraphQL for others.

However, it is a technical and product investment to build and maintain multiple APIs styles, and as a result, only 7% of Series D and 9% of Series C companies offer multiple styles.

Webhooks, which are often less robust than a full API, are increasingly offered as a complement to REST APIs, as they enable real time notifications from one system to another and often reduce the number of calls that occur.

Mature ecosystems who are highly interoperable will offer multiple APIs to increase the scope of use cases that can be addressed and give partner developers more flexibility.

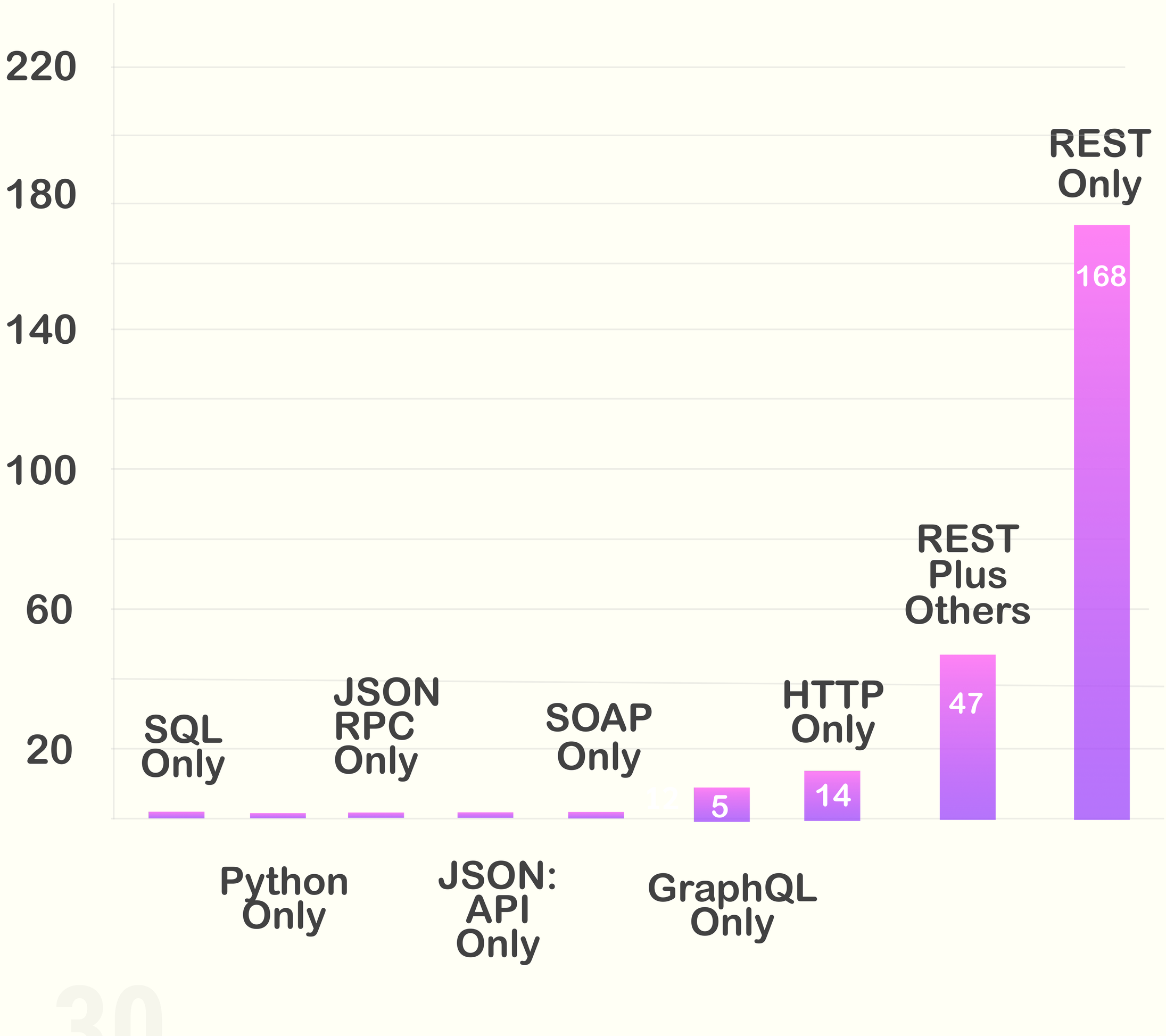
Number of External APIs at 400 SaaS Companies by Style



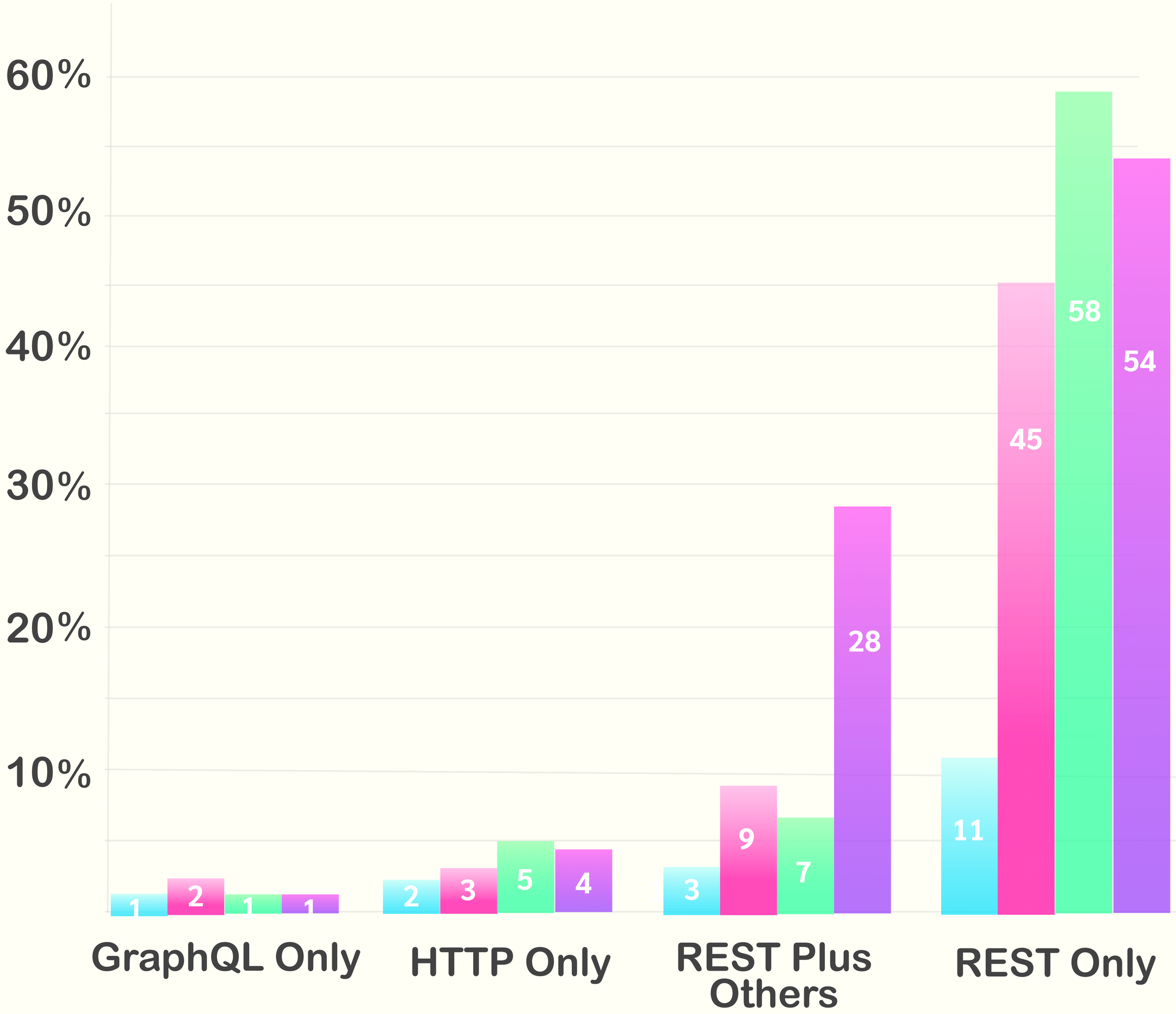
pandium
integrate all the things

pandium
integrate all the things

Number of Companies Offering a Particular API Style (of the 246 Companies that Publicly Identified an API Style)

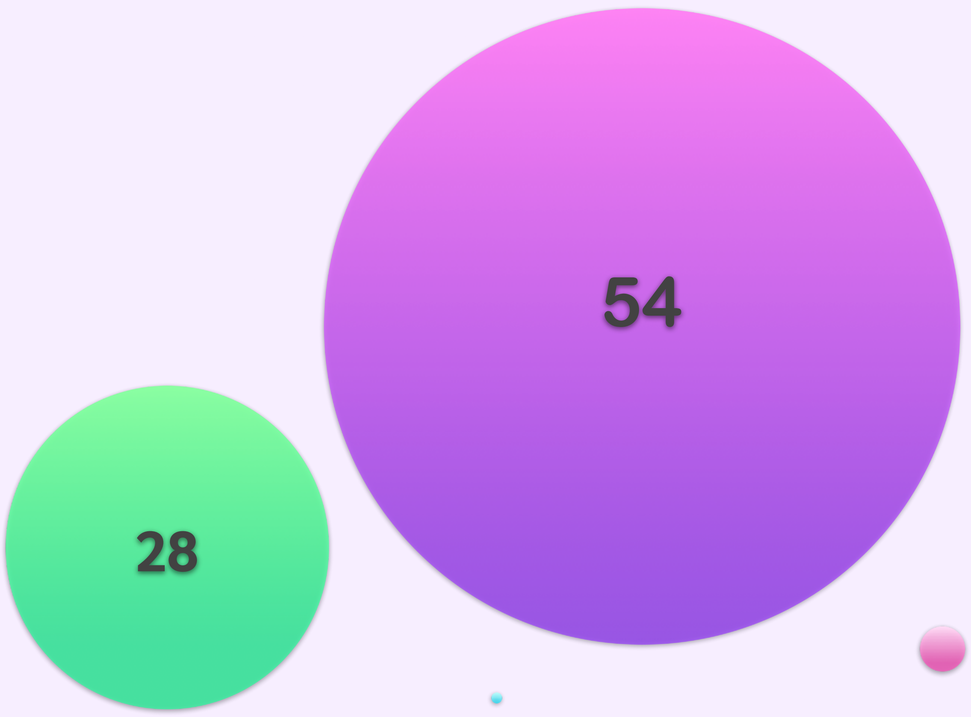


Percentage of Companies with External APIs by Style and Stage

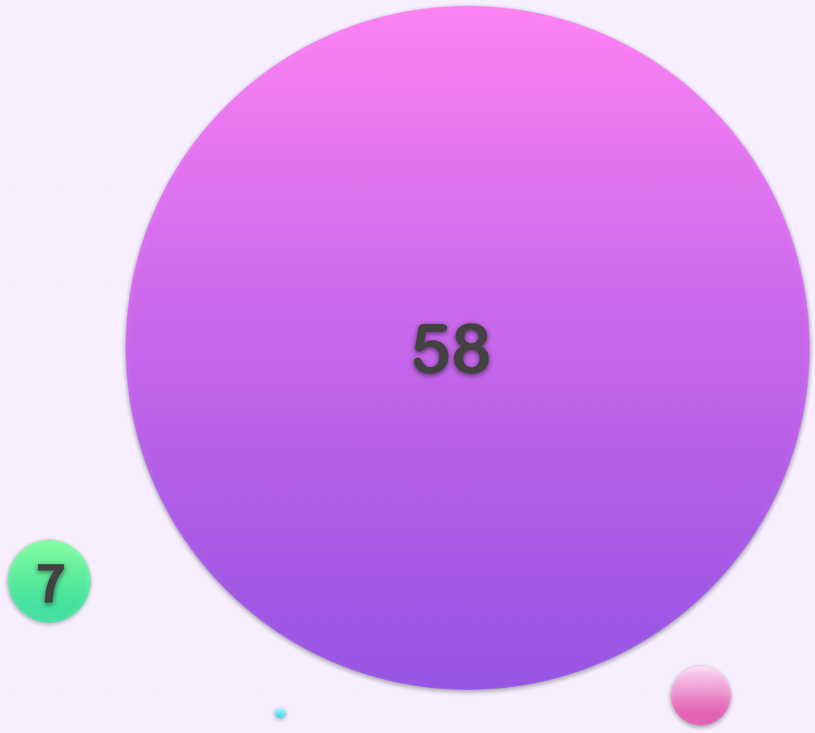


Number of Companies with Particular External API Styles by Stage

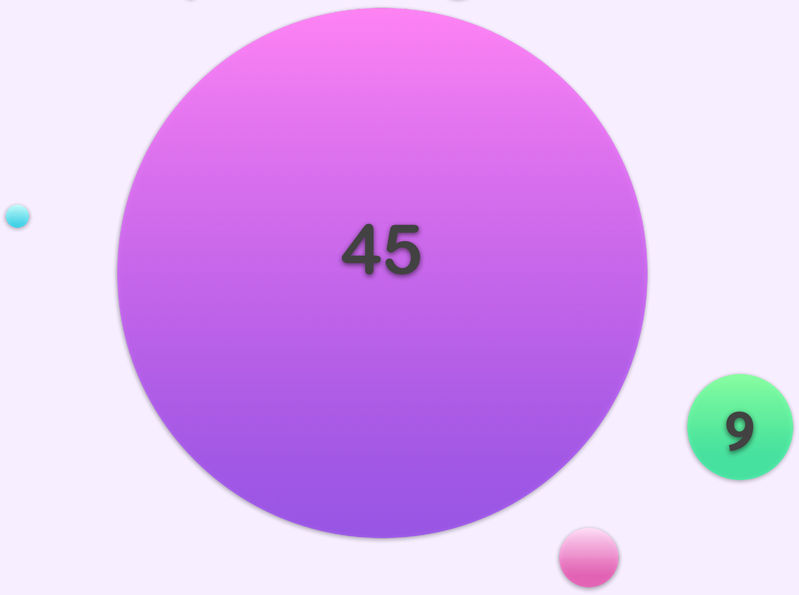
100 LARGEST (89 publicly identified)



100 SERIES D (74 publicly identified)



100 SERIES C (65 publicly identified)



100 SEED (18 publicly identified)

REST ONLY		HTTP ONLY	
REST PLUS OTHERS		GraphQL ONLY	

Webhooks

Webhooks enable developers to set up event triggers in one system that will send data to another system when a defined event occurs.

The advantages of webhooks is that they send data in (relative) real time, right after the event occurs, and they can be more efficient than REST API-based integrations.

Webhooks only send data when the event occurs. REST APIs, in contrast, are used in an integration and are generally called on a set schedule.

For example, an integration might call a system's API to get data on form submissions 50 times a day.

But if the system is only receiving 2 form submissions a day, it is inefficient to call the system 50 times a day to ask about new form submissions. Most of the responses will include no new data.

A webhook, in contrast, will just send the 2 form submissions whenever they are submitted, and do nothing the rest of the day.

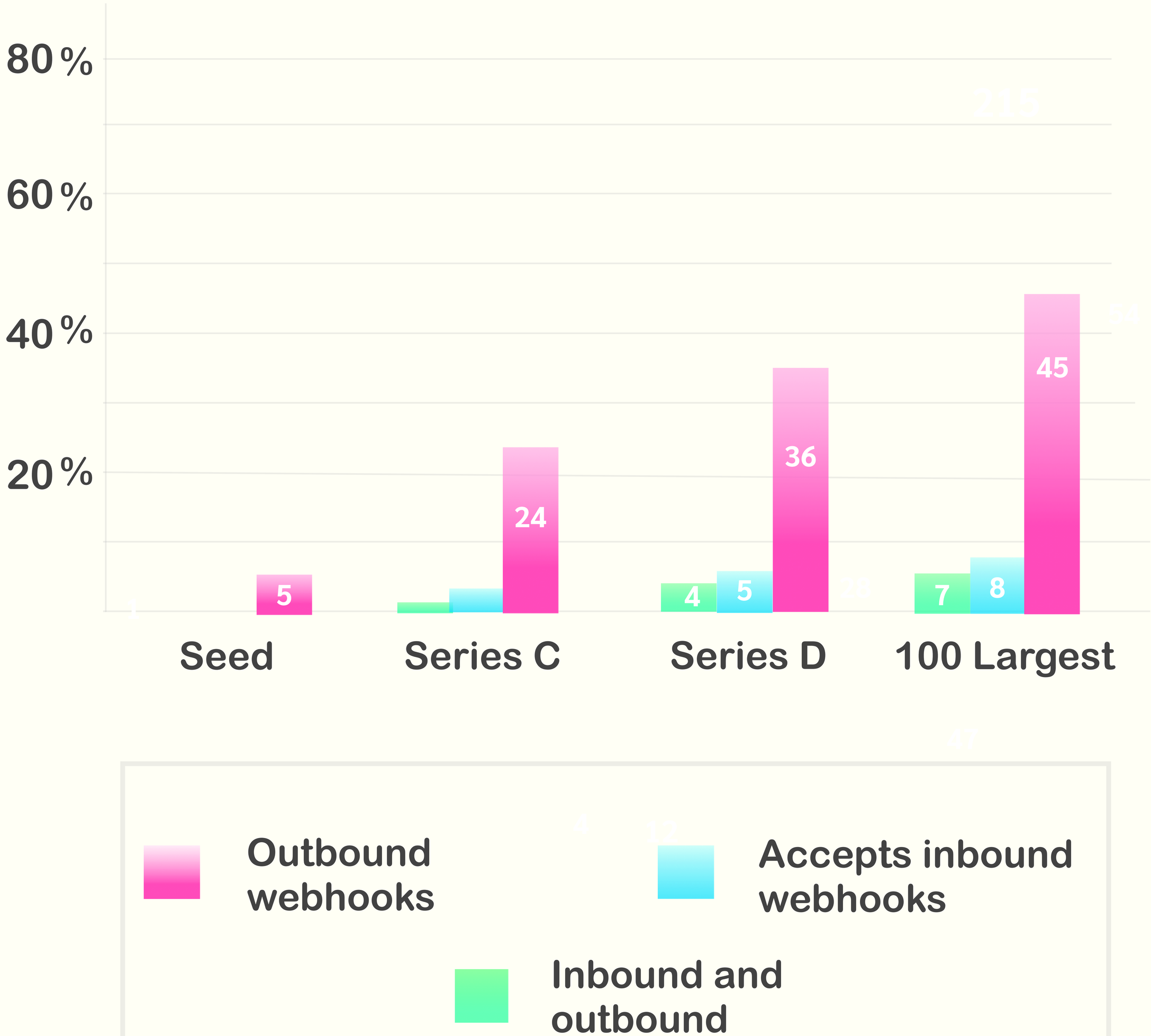
Webhooks are popular complements to REST APIs. Most companies start by offering outbound webhooks for a few event triggers. Later, they will expand the number of event triggers that can be used the webhook.

Some SaaS companies also facilitate inbound webhooks, making it easy for data from other systems' webhooks to be added to their system. This is less common than outbound webhooks, though.

60% of the 100 largest SaaS, 45% of Series D, 27% of Series C, and 5% of Seed Stage offer webhooks.

Due to the nature of webhooks, it depends on a system's functionality as to how valuable they would be for developers. But their widespread and surging popularity suggest most products have a use case where they add value and efficiency.

Percentage of SaaS Offering Webhooks



3. Robustness of APIs

Not all APIs are created equally. For most SaaS product categories, having an external API is the bare minimum for interoperability.

But APIs can be more or less robust in terms of the interoperability they enable. Individual APIs can be more or less complex, and, in addition, companies may offer multiple APIs, even if they are in the same style.

Of the companies with at least one external API disclosed, for example, the 100 largest companies offer an average of 10 APIs a piece, while Series D offer an average of 3, Series C an average of 2, and Seed an average of 1.

In addition to the number of APIs, robustness comes from allowing more data to be operated on in a variety of different ways.

Focusing purely on REST APIs, the number of resources and the methods supported are the two biggest indices of how robust the APIs are.

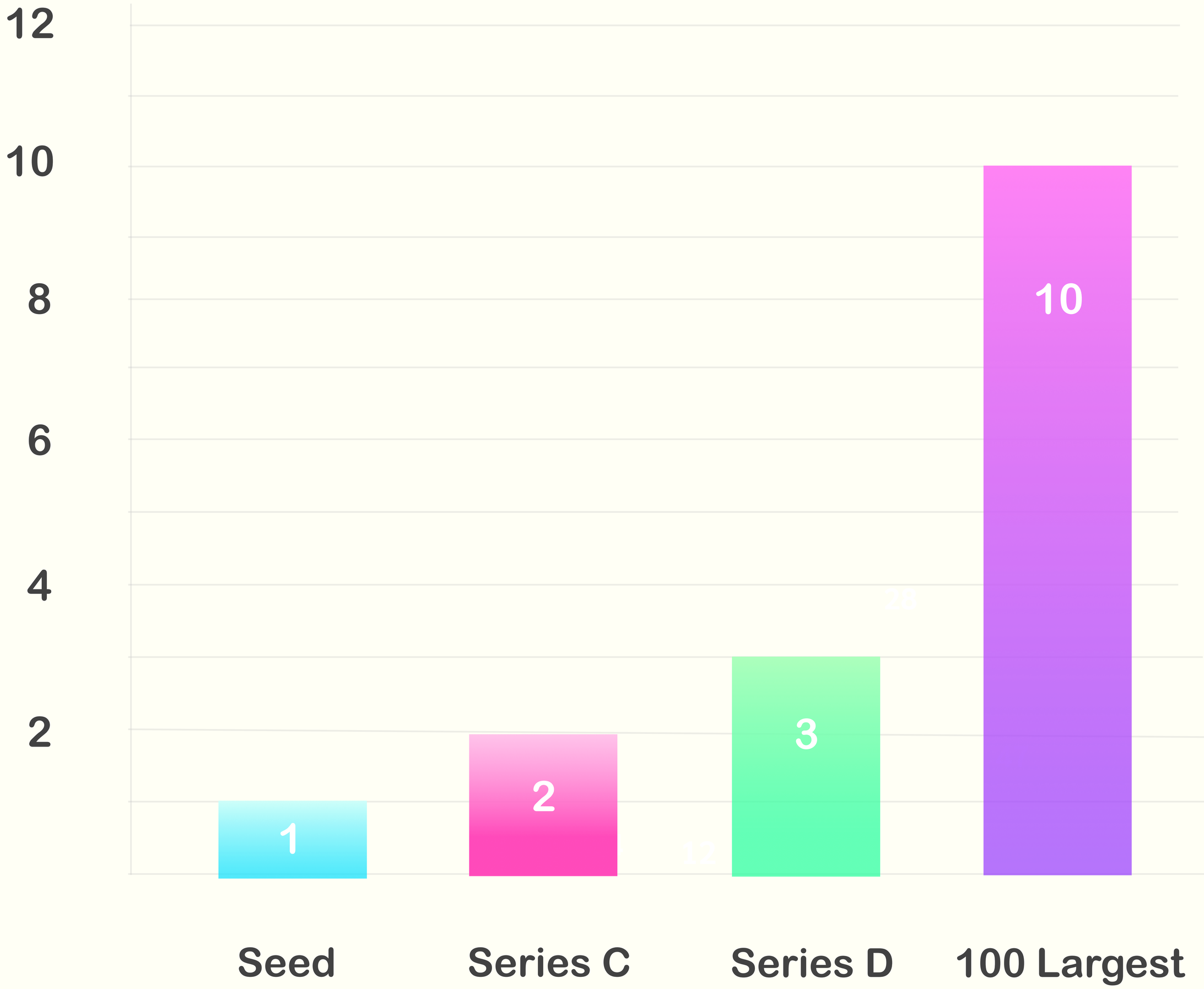
Resources refer to the different data objects that an API enables other systems to "act on."

A CRM, for example, will have resources for Deals, Companies, and Contacts. If a CRM API only allows other systems to retrieve Contacts information, it is not a very robust API and any product integration is going to have more limited value.

Methods refer to the actions that can be performed on the data. GET, for example, is a method that enables retrieving data, while POST enables adding data to the system.

GET and POST are the most supported methods. As companies' APIs become more complex, they add DELETE, PUT, PATCH, OPTIONS and HEAD to enable other systems to delete, update, and modify data in the system (and to get information about available methods and individual resources).

Average Number of External APIs at Companies with At Least One Publicly Disclosed External API



pandium
integrate all the things

pandium
integrate all the things

Resources

As companies grow larger and more interoperable, they not only provide more external APIs, they also give partners and third party developers access to more data and enable them to handle it in more complex ways.

The 100 largest companies average 62 resources, Series D companies average 31, Series C companies average 27, and Seed companies average 11 in their APIs.

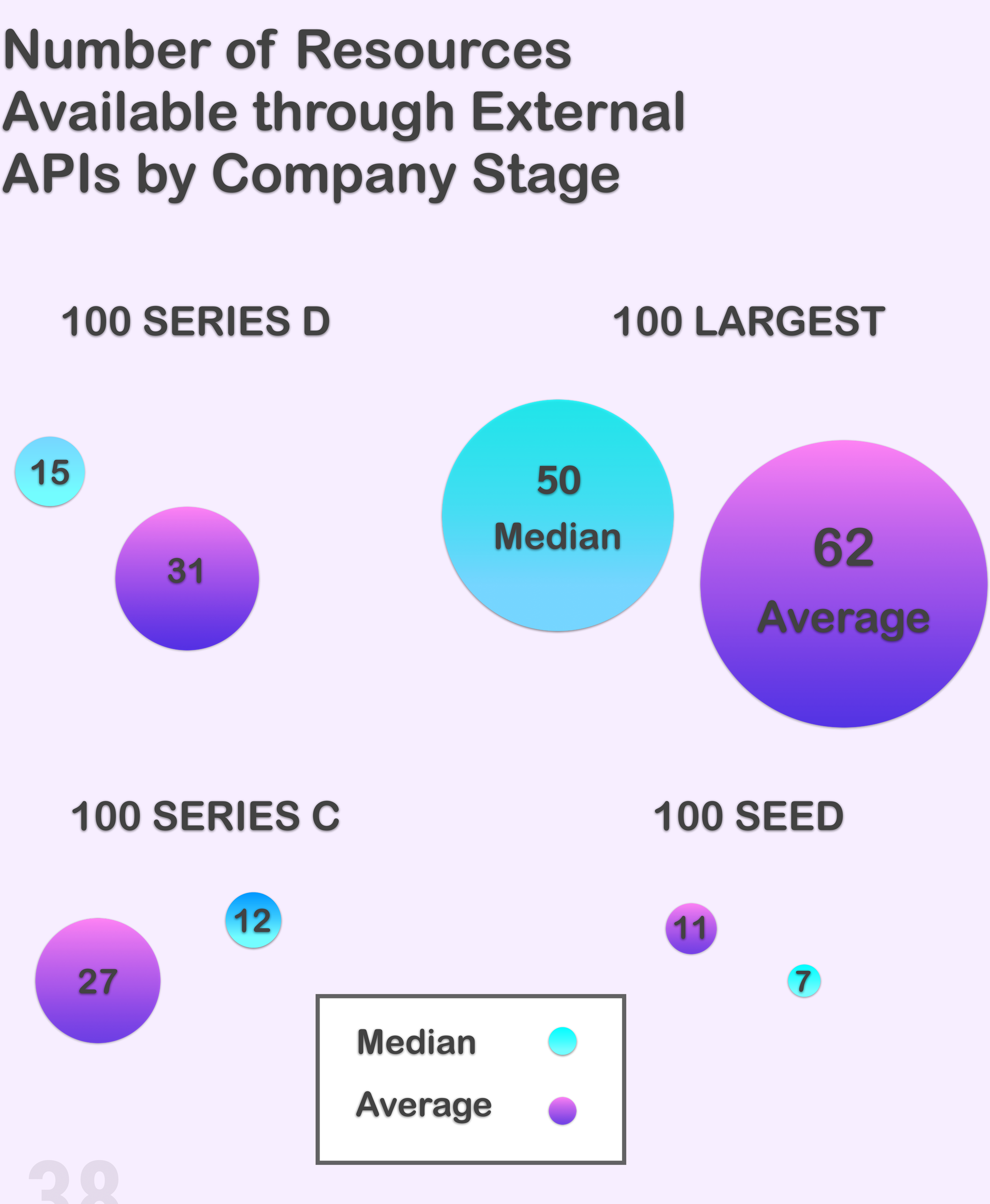
The median number of resources take an even steeper drop, going from 50, 15, 12, and 7 resources, respectively, suggesting that for the largest companies, in most product categories, it is required to have a robust API that offers wide access to data in the system.

Of course, larger SaaS companies' apps also tend to be broader and have more data in their system so they may not be providing proportionally more access externally than smaller companies. Regardless, successful companies do provide partners with a wide variety of access to different data objects in their system.

For example, Atlassian, one of the 100 largest SaaS, **has over 100 different resources** available through its various APIs. Its **REST API for Jira software** has 11 different resources that developers can use to get, post, update and delete data to Jira.

In contrast, Vervoe, a Seed stage startup has **1 API with 5 resources** available. Any integration built with this API will be less complex and robust than one built with an API that has more resources available.

Number of Resources Available through External APIs by Company Stage



pandium
integrate all the things

Methods

Methods describe the actions partners and integrations can take on the data in a company's app. The more methods supported, the more use cases that can be covered by an integration.

Most companies with APIs support GET and POST, with POST having slightly more support than GET. From there, the other methods are less widely supported. Even in the largest 100 companies, 7.6% fewer companies support DELETE than support POST.

However, this drop is much steeper with earlier stage companies: 23% fewer Series D support DELETE than support POST; 37% fewer Series C support DELETE than POST; and 57% fewer Seed support DELETE than POST.

Supporting DELETE can be a heavier lift for companies as it runs the risk of erasing important data in a system if it is mismanaged. GETting data is usually the least intrusive function, as it does not alter the data in any way.

POSTing data is adding data to a system so even if it is added erroneously, it is easier for the user and a developer to undo. But DELETE can remove data, which can be trickier to fix if it is mismanaged and, in industries with regulated data or significant legal implications, like finance, this can cause significant real world consequences when it is used erroneously.

PUT and PATCH are supported even less than DELETE, with PATCH in last place of these 5 core actions for every stage company. While 61 of the largest 100 companies support PUT, for example, only 44 support PATCH.

Like DELETE, PUT and PATCH are often removing data in the system, in this case by overwriting it with new information. (Though they may just be adding new data, if none existed in that field before.)

It should be noted that PUT and PATCH are not necessarily used consistently as both are forms of updating data in a system. In fact, some companies use POST to update data when they should be using PUT or PATCH.

PUT is a way to modify a resource by updating the entire record for it, while PATCH enables updating part of a resource.

So, for example, updating Jane Smith's contact in a CRM might use a PUT if the entire contact record is being sent, while a PATCH may enable simply updating the Contact Jane Smith's phone number.

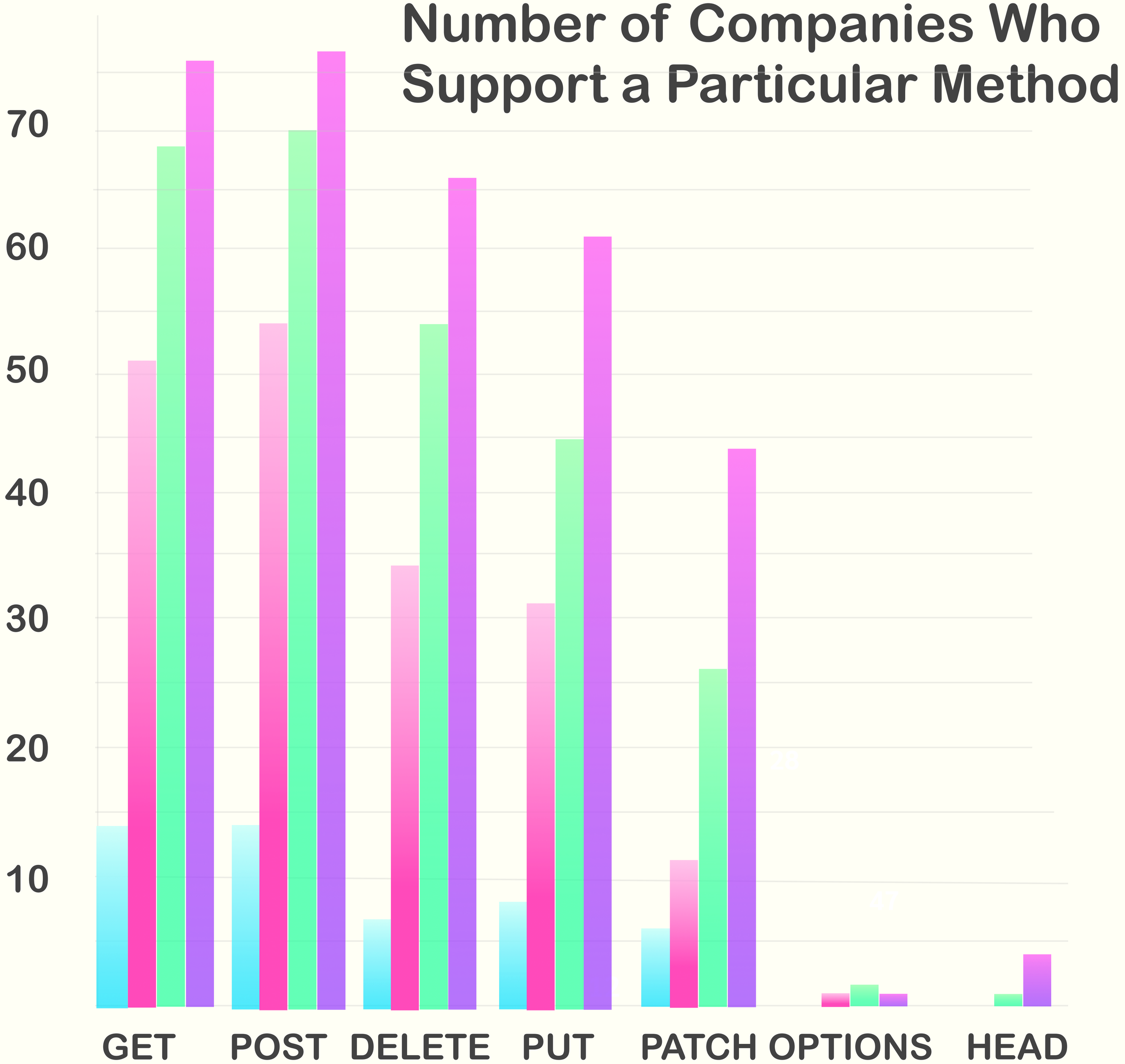
Supporting both gives developers more flexibility in their integration design and also supports additional business use cases, as another system may have Jane Smith's new phone number but not the rest of her information (which, if PATCH is not supported, would then require a convoluted path of getting the rest of the data and then updating it alongside the phone number).

OPTIONS and HEAD are supported by very few companies in their external APIs. They provide a way for systems or developers to get meta information about available methods and resources for the API. As a result, they don't directly enable new business use cases, but they may help integration design and staying abreast of API changes (which can impact integrations over time).

Ultimately, supporting more methods, unless there are reasons not to around data regulation, security, or sensitivity, creates more interoperability and makes partner developers' jobs easier.

With more methods, developers can more easily build well designed integrations that meet more business use cases that satisfy customers. As a result, we see more support for more methods at larger companies, who provide more interoperability than their smaller counterparts.

Number of Companies Who Support a Particular Method



pandium
integrate all the things

Authentication and Authorization

Authentication and authorization refers to how a SaaS company identifies and authorizes a user to get access to their API (or part of their API).

When it comes to product integrations, it is ideal to require the partner system be identified and authorized as well as the end user.

However, not all companies, especially at earlier stages, require this. As a result, those partner-built product integrations usually ask the customer to manually enter their username and API key to use the integration.

This is not best practice as it makes it difficult to see whether customers are using the API on their own or through a particular partner integration they installed. Companies with this setup may not even know when partners build product integrations to their system.

This makes it tough to see the business value integrations are driving and also to address functionality or security issues with partner-built integrations.

OAuth 2 is a protocol that requires the partner system and the end user both to be identified and authorized with IDs and secrets. It also enables the end user to click to authorize the integration rather than having to manually enter an API key.

Most mature ecosystems require their app partners to implement OAuth 2. 52% of the 100 largest SaaS companies publicly support OAuth 2.

There are almost certainly more who require it for tech partners, as some companies do not state publicly what they require from app partners in terms of authentication and authorization.

Customers, for example, may be enabled to use the API by sending a bearer token in an authorization header. But this doesn't mean that partners are not required to implement OAuth 2 as companies commonly allow private or custom integrations to use API keys or tokens, while requiring public apps to use OAuth 2.

HubSpot, for example, requires app partners who make their **app public to use OAuth 2**, while **private apps can simply put the bearer token in the authorization header**. Similarly, Shopify **requires public apps to use OAuth 2**, while allowing custom apps that are created in the Shopify admin to use an access token in the authorization header.

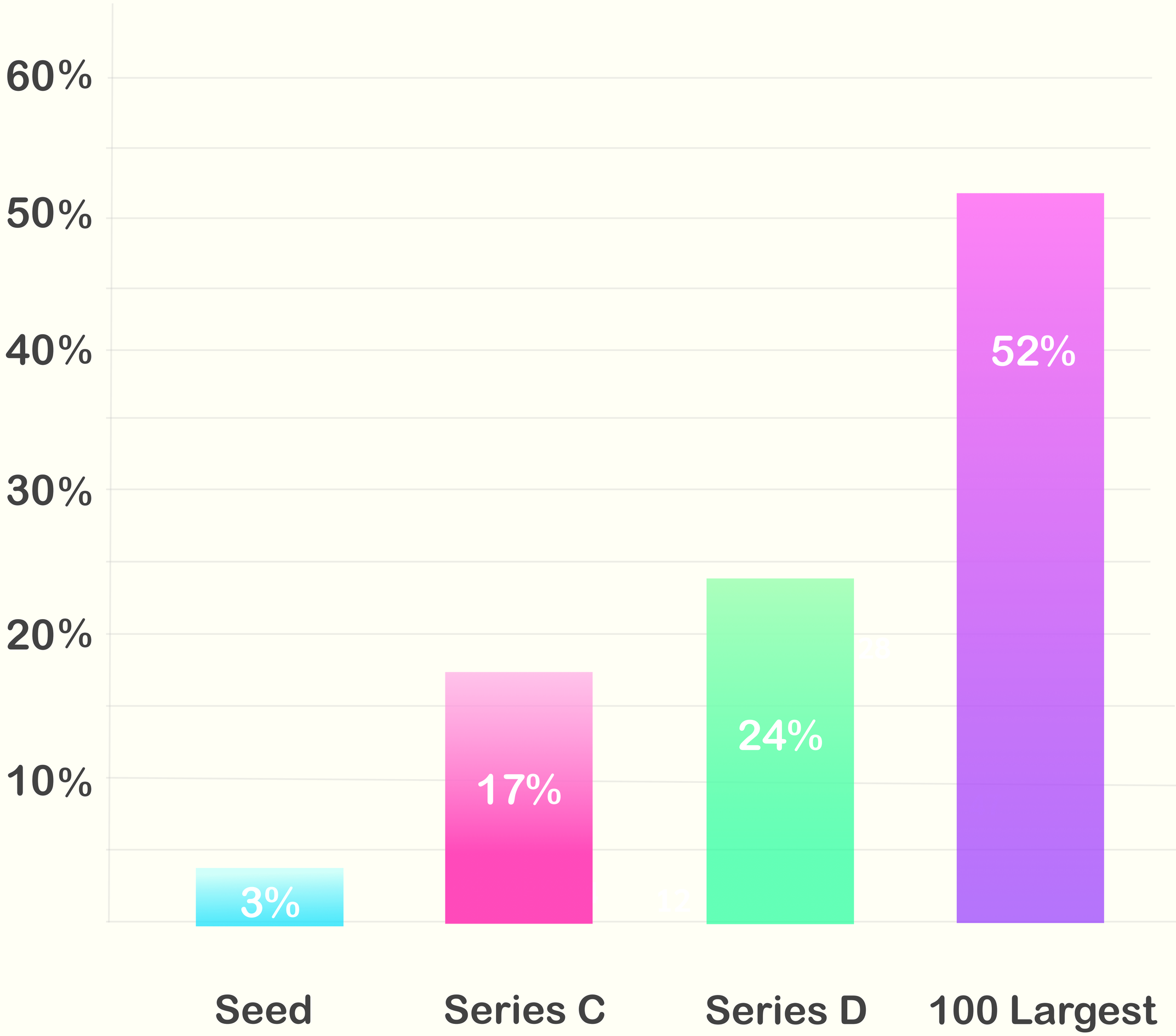
This allows customers to use the API without having to implement OAuth 2 (which takes longer than sending a token in the header) but ensures that the system has complete visibility into partner apps and gives it more control over product integrations being presented to their customer base.

Requiring this of partners does require a technical investment and technical support, however, which is likely why earlier stage companies have mostly not implemented it.

As mentioned, because this information is not always public in less developed ecosystems, the real numbers of SaaS companies requiring their partners use OAuth 2 is almost certainly higher than the number below.

Despite this, the trend line is clear that larger companies require it more frequently, with 52% of the 100 largest, 24% of Series D, 17% of Series C, and 3% Seed stage companies publicly supporting or requiring it of tech partners.

Percentage of Companies that Publicly Support or Require Partners to Use OAuth 2



API Versioning

APIs are regularly updated with new fields and new functionality. Not all of these changes require releasing a new version of the API. For example, if an API simply added support for a new resource or a new property of a resource, there may be no need to release a new version.

Generally, a new version is merited if the change will break the existing uses of the API. For example, changing the format of the response data or removing support for a resource would break some existing integrations.

Most SaaS companies use numerical versioning (i.e. v1 or v1.0.0) but a small minority version them by dates. Overwhelmingly, companies put the version in the URI of the calls (which would look like this: https://www.saascompanyx.com/api/v1/contacts). Only a few companies put the version as a query property or in the header.

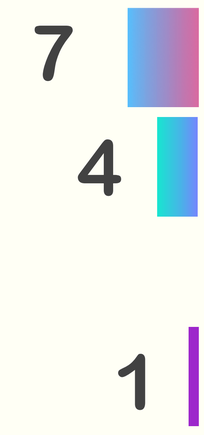
For companies using **numerical versioning with three places** (1.0.0), the first number is only changed with a breaking change, the second is changed when there is backward compatible functionality added, and the third is changed when a backward compatible bug is fixed. This provides more information and enables developers to easily understand the nature of the change.

Unsurprisingly, larger SaaS companies have more APIs on later versions reflecting their maturity when it comes to interoperability. However, the average large SaaS company is still only on v2.

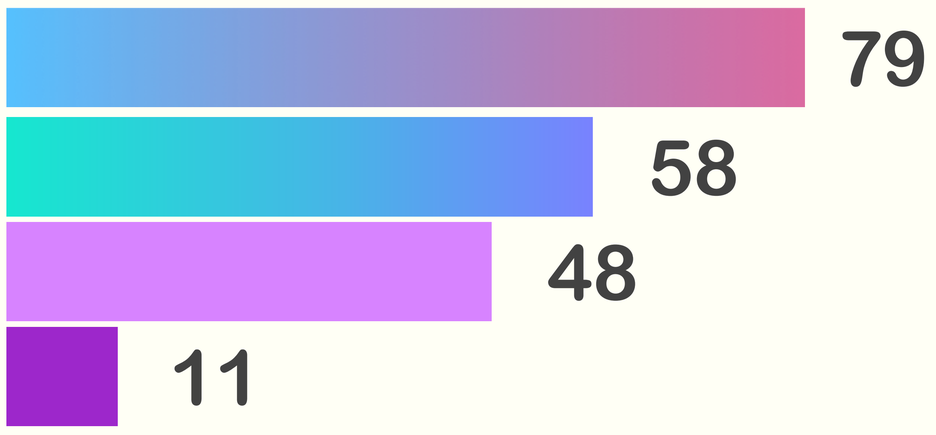
API Versioning at 400 SaaS

pandium
integrate all the things

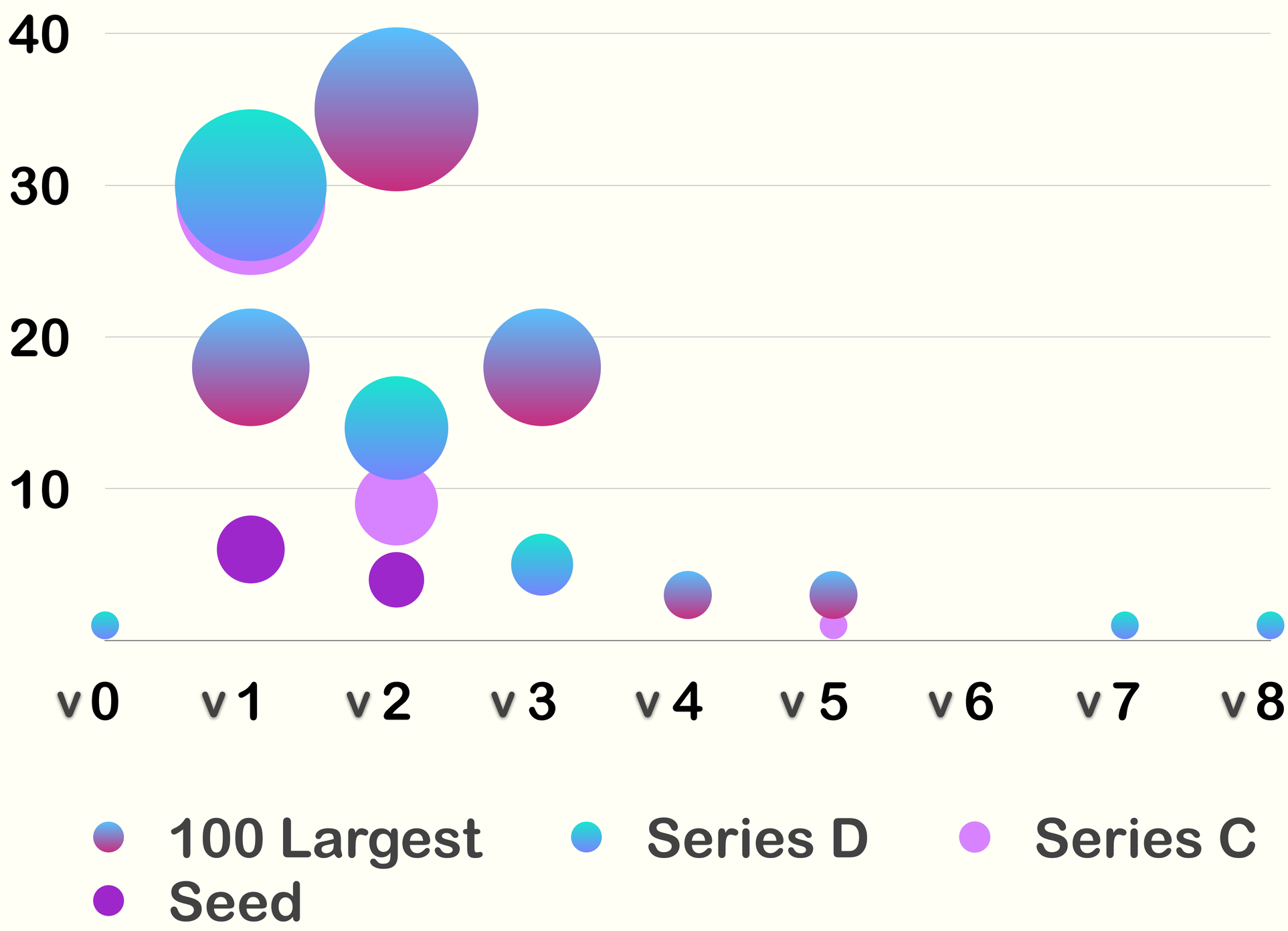
Versioned by date



Versioned numerically



External APIs on Different Numerical Versions



4. Tech Partner Experience

Mature ecosystems must offer a strong tech partner experience. For partner developers, this starts with public API documentation. As covered, larger SaaS companies provide public documentation in higher numbers.

In an era of product led growth where developers generally prefer to try a product before speaking to a representative, giving open access to try out the API is also an important part of an ideal UX.

SDKs, which are language specific libraries that help developers to build with the API in their language of choice, also make the partner developer experience better as it saves times and makes it easier for them to build complex integrations.

Free sandboxes where developers can test their integrations also ensures that developers can confidently test and release a robust integration with minimal hassle.

For communications, developer-specific portals where partner developers can get app installs and analytics enable host companies to ensure that developers and their companies can easily get the information they need.

While young companies may provide direct support to all partner developers, this is not scalable, nor does it work well for metrics like app installs or tile visits.

As more partners seek to build into a host system, the company must implement ways to communicate and support partner developers at scale.

Across all these touchpoints, larger SaaS companies as a group do a better job of providing the ideal journey.

Ease of Getting a Developer Account

In most cases, a developer account enables partners to get an API ID and secret key so they can start using the API and building apps.

There is a spectrum for how easy it is for people to get these accounts, which is a reflection of how open an ecosystem is:

1. Require a personal email to get a developer account
2. Require a company email and company name
3. Require the person answer a more detailed questionnaire with details on their company, customer base, and proposed use case
4. Require the person submit a questionnaire and go through a process to become an official, approved partner (and may require a program fee)
5. No public information on how to get a developer account along with a highly manual and obscure process to get access, based on subjective and non-disclosed factors

The advantage of running an open ecosystem is that it attracts more developers and more partners (And it provides a better experience to people who would have become partners regardless).

While some integrations “have to” be built, many times developers are deciding amongst similar systems in terms of which to build on. This is even more true when it comes to building extensions on top of systems.

Closing down the ecosystem and putting up barriers to getting a developer account will often lead to developers going with more open ecosystems.

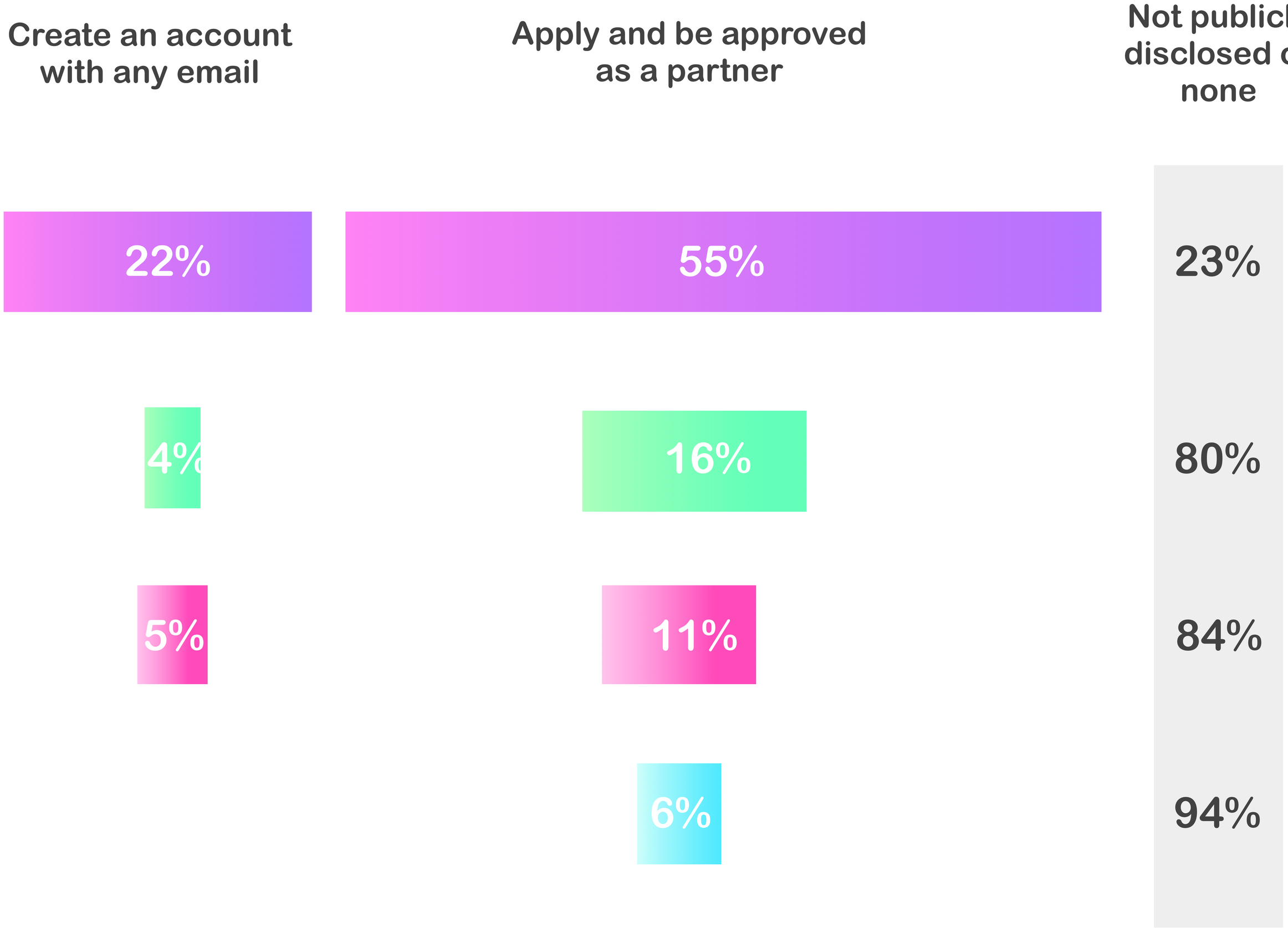
Having an open ecosystem requires marketing, legal, and technical infrastructure, so the cost is not negligible. However, an open ecosystem usually results in a much longer tail of partners and more invested partners.

22% of the largest 100 SaaS enable developers to sign up with just their name and an email. This is 5.5 times more companies than even companies at Series D.

Companies like Shopify, Wix, HubSpot, Smartsheet, Procore, and monday make it very easy to get a developer account with almost no restrictions on signing up. Unsurprisingly, these companies also have a large number of partner-built integrations.

In contrast, zero Seed stage companies enable people to get a developer account with an email and only 5% of Series C companies do.

How to Get a Partner Developer Account



pandium
integrate all the things

Partner and Developer Portals

When tech partners and developers decide to invest in an ecosystem, they need ways to get support and view metrics on their integrations and extensions.

While strategic tech partners will likely be communicating directly with partner managers and other stakeholders, for most partners, there has to be asynchronous and scalable ways of accessing support and information.

Partner or developer portals where partners can log in and access information, support, and app analytics are how most companies accomplish communicating with partners at scale.

Partner technology companies are trying to create “a single pane of glass,” where all tech partners might be able to exchange information with all their other tech partners.

However, this does not currently exist, and the current attempt are heavily focused on the go-to-market side of partnerships like sharing account data, sharing marketing materials, and displaying co-selling and reselling revenue outcomes.

There is no “single pane of glass” for sharing API documentation, technical support, and app install and app marketplace metrics.

Many large SaaS companies have Postman collections, which enables developers to test out their APIs. So if a developer is on Postman, they may be able to access a number of their tech partners’ APIs. Unfortunately, this provides no information on building a product integration, the number of installs of that integration once it is launched, or information related to the tech program or its terms.

So, for now, portals seem to be the only way SaaS companies have found to scale tech partner support and metric sharing.

This creates a problem for SaaS companies, though, as tech partners do not want to have to log in to dozens of portals.

Until there is a “single pane of glass” where all partners can log in to one system, one solution to the too-many-portals problem is to enable integrations to those portals. Then, partners or third parties can pipe the information in the portal to their other systems that they use regularly.

Very few companies enable partners to programmatically access app and app marketplace data. Those that do, like **AWS** and **Salesforce**, are mature ecosystems that likely face less of a struggle getting partners to log in to their portals.

Merely having APIs for app installs and analytics may not be enough as that requires a developer to then build an integration. A product integration that connects the analytics to BI systems, databases, or CRM would likely be more useful to most partners.

Short of offering that, though, the best thing a company can do is make sure their developer portal is well-designed (and ideally enables single sign on to make logging in even easier).

Some companies offer one partner portal, while others offer a developer portal specifically for tech partners. If a company only has one partner portal for all partners, it is imperative to ensure that different partner types can easily find and access the information and data they need.

With the 100 largest SaaS companies, 45% had a dedicated developer portal and 29% had one partner portal, which means at least 74% had a portal for tech partners to log in to. The remaining 26% either did not have public documentation of their partner experience or had no portal.

In contrast, 70% of Series D companies either **did not** have a portal or did not share the partner experience. 11% had a dedicated developer portal, and 19% had a partner portal.

This continues to decline at earlier stages; 9% of Series C companies have a partner portal and 8% have a developer portal. At Seed stage companies, only 1% have a developer portal and only 5% had a partner portal.

Part of the reason for this decline is that a partner portal isn't needed if a company only has a dozen tech partners. That can be managed manually. It's only at scale that a portal helps to streamline and manage relationships effectively.

Still, many Series C and Series D companies likely have enough potential partners to justify building a portal, but just have not yet invested in it. Portals not only enable low touch communication for the host — they make it easier for partners to keep track of and modify their submissions, and track their results.

Cloudbeds, for example, already lists 74 integrations in its marketplace, but is still requesting its partners to **use google forms** and **emails** to submit materials and app information.

Submitting this information via forms or email makes it very difficult to update or edit it, or easily see what was submitted. Having a hands on approach may have a benefit of developing relationships with partners, but centralizing information and making it easily accessible is still beneficial even when the relationship is high touch.

The best tech partner portals have the following features:

- Access or links to API documentation, integration guides and specifications, changelogs, and announcements related to tech partners
- Tech partner and developer agreement and terms
- Ability to add team members with different levels of access
- Ability to add contact and company info
- Ability to get an App ID and secret key
- Link to sandbox/testing environment
- Live chat, in-app messaging for support, and/or contact info for support
- Place to submit, save, edit, receive feedback and resubmit marketing materials for the marketplace
- Place to submit apps for review, receive feedback, and resubmit
- Place to see app installs, uninstalls, reviews, revenue shares, marketplace analytics
- Place to see multiple apps

If the tech partner portal is also for go-to-market collaboration, it may also have co-marketing and co-selling materials and results, affiliate links, and a place to register deals.

Keep in mind these are just best practices for a mature ecosystem. Conducting UX research on partners and soliciting feedback is the best way to understand what features are most important for a particular ecosystem of partners.

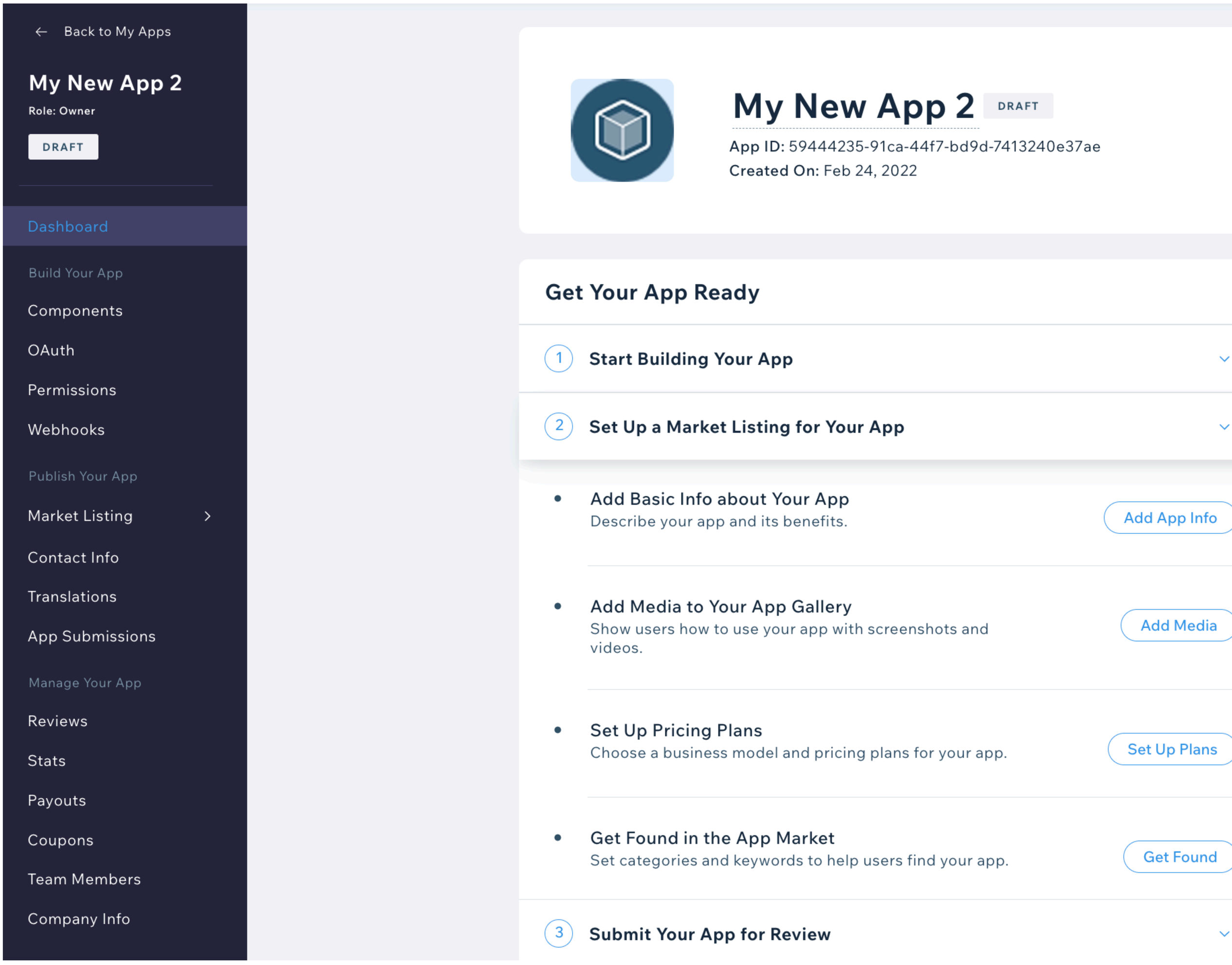
Wix's developer portal is a strong example of a robust developer portal. The only notable features it is missing are analytics on customers who engaged with partner app tiles in their marketplace and in-portal support. But only a handful of app marketplaces, like AWS, Salesforce, and Google Cloud, have that functionality in their portal.

The homepage of the portal has links to documentation, support, the app marketplace, and a CTA to create a new app. It also has an announcement section, and showcases all the current apps the partner has built, and signals if they are in draft or launched.

Credit: Wix

Once you click on an an app, a workflow guide details each steps that needs to occur to get it launched. On the left sidebar, a developer can get their app credentials, update their contact info, add team members, and track stats and payouts from their app.

Credit: Wix



On the right side of the portal, a developer can click to test their app on a sandbox account, as well as search for help with questions they may have.

Credit: Wix

My New App 2 DRAFT

App ID: 59444235-91ca-44f7-bd9d-7413240e37ae
Created On: Feb 24, 2022

Test Your App

Get Your App Ready

1 Start Building Your App

2 Set Up a Market Listing for Your App

- Add Basic Info about Your App
  Describe your app and its benefits. — Add App Info
- Add Media to Your App Gallery
  Show users how to use your app with screenshots and videos. — Add Media
- Set Up Pricing Plans
  Choose a business model and pricing plans for your app. — Set Up Plans
- Get Found in the App Market
  Set categories and keywords to help users find your app. — Get Found

3 Submit Your App for Review

Help Center
We're here for you
What do you need help with?

Featured Articles

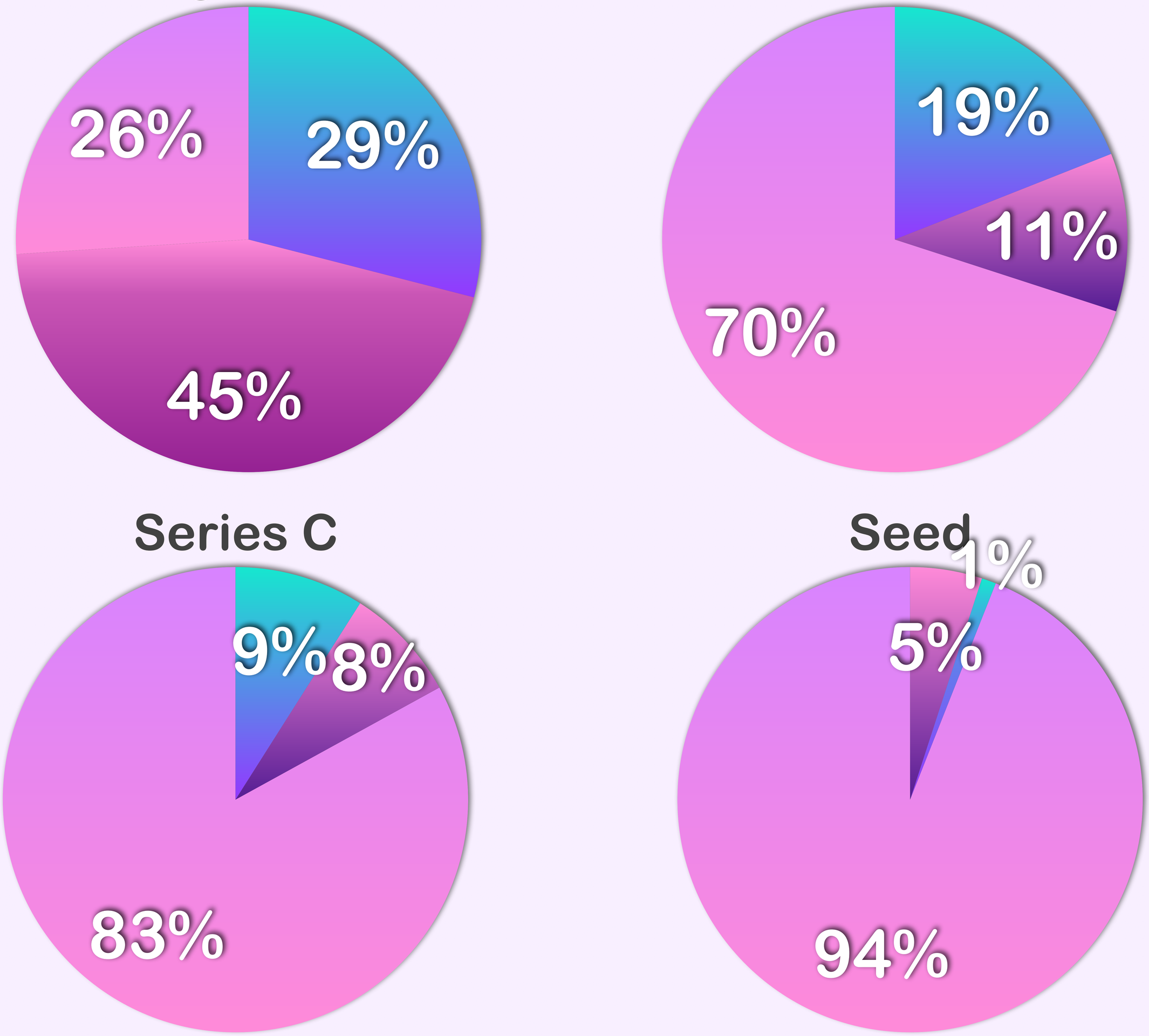
- Add Your App Pricing
- Invite New Team Members
- Find Your App's Public Key
- Add Permissions to Access User Data
- Subscribe to Webhook Events

Powered by Wix Answers

Looking for some more advanced features?

- Create a drag-n-drop visual component
- Invite team members

Percentage of Companies with Developer and Partner Portals



SDKs

Many mature ecosystems offer SDKs (software development kits) for their external APIs. SDKs are libraries in particular languages that developers can use to build integrations.

When an SDK is available in a language that they normally code in, they save developers time and enable them to more quickly launch robust integrations.

This makes the partner developer experience better because they have to spend less time on building the integration without sacrificing quality.

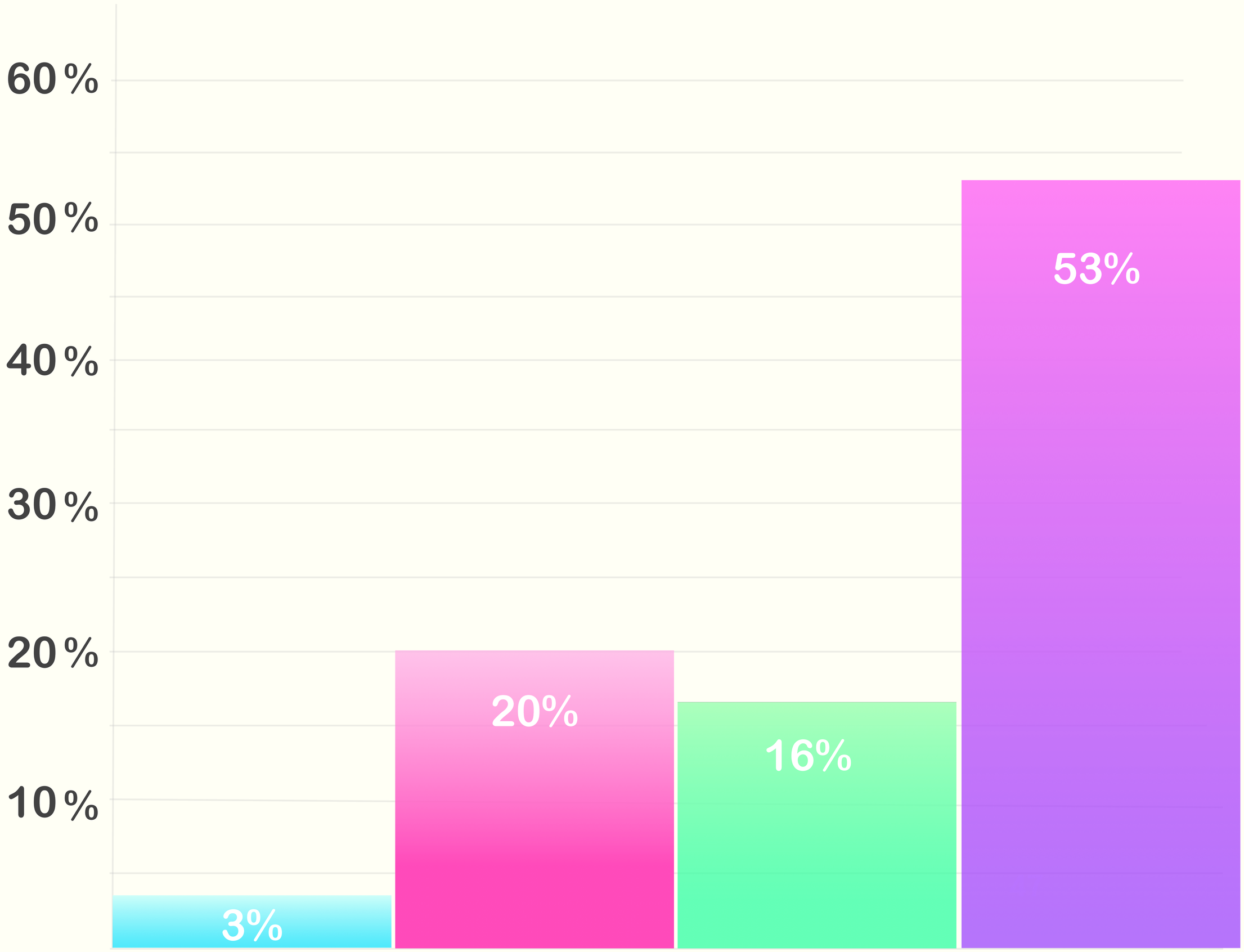
But SDKs not only have to be built, they also have to be maintained. As a result, it is primarily larger ecosystems that offer them.

Of the 100 largest companies, 53% provide SDKs for their external APIs. Out of the companies that have them, they offer an average of 5 SDKs.

Only 16%, 20%, and 5% of Series D, Series C, and Seed companies provide SDKs, respectively, suggesting that the resources to build and maintain them are not insignificant, and that it is prudent to carefully assess the value to potential and current partner developers before building them at an earlier stage.

Large ecosystems benefit from a community of developers giving more technical feedback on APIs and SDKs. Developer communities of large ecosystems can also build their own community-supported SDKs for the host's APIs.

Percentage of Companies with Public SDKs for API or Extensions



Median number of SDKs per company with SDKs

pandium
integrate all the things

Sandboxes

A good partner developer experience should also include a sandbox where developers can test their integration. This becomes even more critical when integrations are complex and handling financial or regulated data.

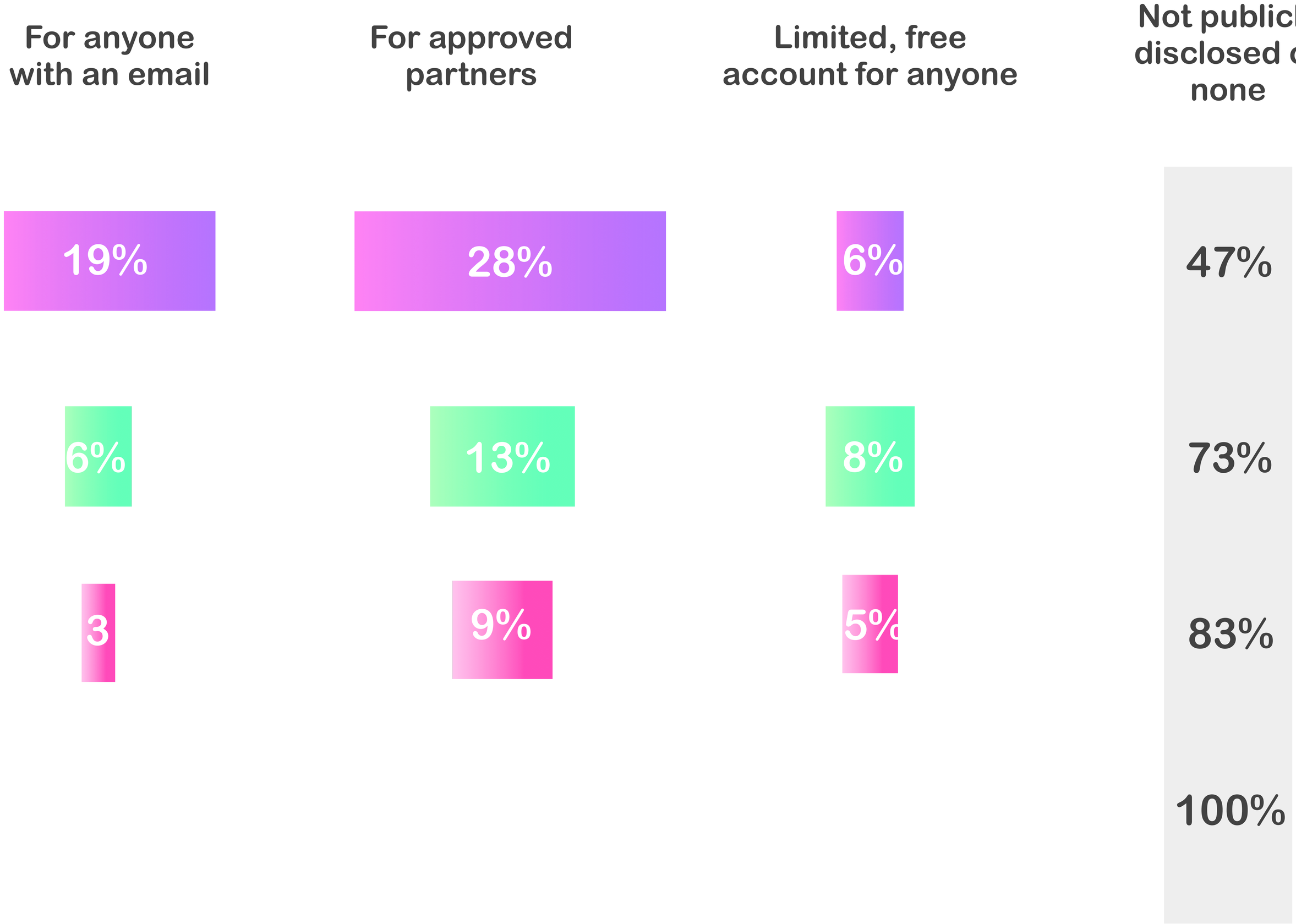
The most developer friendly sandbox is free, includes all features of the highest level plan, comes with robust sample data, and does not have restrictions or permissions that do not exist in production.

Providing this is a technical lift and is most necessary when integrations are more complex. Most larger companies provide a sandbox that is fairly robust, though it may only be for approved partners.

Many companies who have a free plan for their SaaS direct developers to get a free account and use it for testing. This is certainly better than nothing, but it poses a problem for integrations that rely on features that are locked behind a paywall.

Having to create sample data from scratch can also be challenging, and sometimes impossible. The most ecosystem friendly companies ensure developers have a viable way to test all their integration's features before going into production.

Percentage of Companies Offering Sandbox Accounts



5. Platformed

As a business model, a platform model refers to a business that is creating value by enabling exchanges between groups, usually defined as consumers and producers.

The consumers and producers can be the same people, as is often the case with social media platforms like Facebook, where the same people may be both producing content and consuming it.

Or, like in the case of Uber, the two groups can be much more distinct, with riders (consumers) and drivers (producers). Of course, a driver can also be a rider, just not at the same time (while on social media, interactive content often has people producing and consuming in the same interaction).

Amazon's e-commerce business, similarly, facilitates third party merchants selling to individual consumers.

These platforms are facilitating other groups to exchange value with one another. In some cases, like Uber or Airbnb, the platforms aren't also selling their own product or services directly to the consumers. In others, like Amazon e-commerce, Amazon is also selling products and services directly to customers.

Like Amazon e-commerce, the vast majority of SaaS companies who become platforms continue to sell their core products to customers (the consumers), while also facilitating tech partners (producers) to provide customers with apps and extensions (as well as agency and solution partners providing customers with services).

But there is a key difference between Amazon selling their own shoes, books, and home goods on their platform and SaaS platforms. Amazon's products are competing directly with the products of third party merchants.

On SaaS platforms, there can be and are competitive features between the core product and the app partners. However, in large part, apps and extensions are not competitive with the core product and in fact are designed to enhance and improve the experience of the core product.

This synergy multiplies the value being created and connects SaaS platforms to their app partners in a deeper way than Amazon's e-commerce platform connects to third party merchants or Uber drivers connect to Uber. Merchants may adjust their product to optimize it for selling on Amazon, but, for the most part, merchants are there to sell to the large customer base, and would leave if other platforms had a larger customer base.

In the case of SaaS platforms, this relationship can be stickier. While app partners are there for the large customer base, they also build the app and their product to align with the core product of the SaaS platform.

Selling shoes on Amazon, one can fairly easily sell the same shoes on Walmart's platform. But an app built for Salesforce isn't so easily transferred to an app for HubSpot. (There is some technical overlap, but it is not a matter of simply making moderate adjustments and listing elsewhere.)

There is another important distinction for SaaS companies: the difference between a platform business model and being a technical platform. In SaaS, these words are often used interchangeably and conflated, when they are quite different.

Salesforce, Atlassian, and Shopify, for example, are platforms that have attracted a large number of developers to build integrations to their core products and extensions on top of those products.

These companies have platform business models but they are also platforms in a technical sense: a technical layer that others can build apps on top of. Building on top of a platform can include hosting the app, having an environment for developers or even citizen developers to build extensions in, or providing UI components inside the app that developers can use to build front-end experiences for customers.

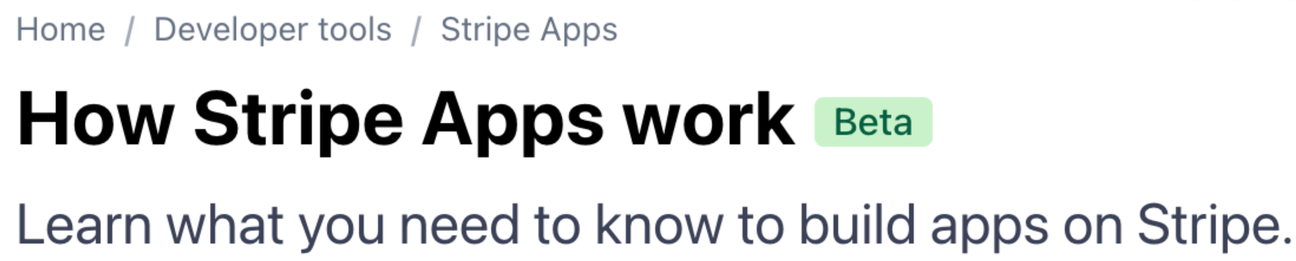
So while these companies sell directly to customers in addition to operating a platform business model, they also simultaneously are a technical platform for partners and third party developers, while also operating as a product. (In contrast to, say, a platform as a service, whose point is to serve as a technical platform.)

But not all highly interoperable SaaS companies are technical platforms. Companies like Stipe and Twilio, for example, are products sold to developers to be used as building blocks in the developers' systems.

As API products, they are highly interoperable and designed to be embedded in a very large number of other systems. But they are not primarily designed to be built on top of as a technical platform, but instead built with and embedded elsewhere.

In fact, Stripe only recently released the ability for developers to build on its dashboard. This functionality is still in beta.

Credit: Stripe (June 2022)



If companies like Twilio and Stripe primarily drive value by being embedded in other systems, are they still platforms in the business model sense?

Because of their large customer bases and useful product features, companies like Stripe attract partner developers to build integrations to their systems and the customers of those systems are deriving value from the integrations.

As a result, they are facilitating an exchange of value between tech partners and customers. If that is all that is required for a platform business model, then they certainly qualify.

It is important to note, though, the business model and value exchanges that are occurring likely look different than they do for systems that are both running a platform business model and are technical platforms.

Consider the portion of customers who are likely finding tech partners through Stripe, and what those tech partners look like.

With, say, the Salesforce AppExchange, many customers are finding apps there to enhance the functionality of Salesforce. More generally, in looking for apps, customers often require that they integrate well with Salesforce to even be considered.

Usually, a Stripe customer will already have, or at least purchase at the same time, an e-commerce platform, for example, like Adobe Commerce or BigCommerce. Are customers using Stripe and then discovering BigCommerce or Quickbooks to enhance Stripe's value?

It is possible, but in general, it seems more likely that a larger proportion of systems that are integrated with it are already chosen or being used by the customer, and the partner is offering the Stripe integration to enhance the product features.

There are other app partners of Stripe, though, like dashboard visualizations, that may be chosen just to complement Stripe.

As a result, companies like Stripe or Twilio rarely have transactional app marketplaces, while companies that are platforms in both senses of the word, like Salesforce and Shopify, often are operating transactional marketplaces.

The reason for this is customers would likely find value in discovering and purchasing many of their marketing and e-commerce apps through Shopify or their sales and marketing apps through Salesforce. For them, it streamlines and consolidates their billing through their anchor system.

Would a customer want to buy a bunch of e-commerce apps through Stripe? It seems more likely they would only want to buy what are functional extensions of Stripe, which is currently a small fraction of Stripe's app partners.

These dynamics are important to understand as they will certainly shift the monetization and revenue shares exchanged.

SaaS that are both platform business models and technical platforms will likely have different monetization models than SaaS who have a platform business model but do not have a technical platform.

Many SaaS companies aim to become a platform in both senses of the word.

But the data shows that it's actually interoperability - in whatever direction, whether a product is designed to be embedded in hundreds of other systems or designed for hundreds of systems to run on it - that is important for SaaS growth.

At each stage, there are more companies that have platform features, more companies that have embeddable components, and more companies that have both platform and embeddable features.

Interoperability in SaaS powers the connection of partners and customers, and a better, more unified product experiences, all which drives more value for the product, the partner, and the customer. This, rather than a one-size-fits all goal to be a technical platform and business platform, should be the aim of SaaS companies who want to best leverage ecosystems and partnerships.

Percentage of Companies Embeddable and Platformed

Components Embeddable in Other Products

Calendly
PLAID
grammarly
gusto
MessageBird
Adobe
shopify
LaunchDarkly

26 | 20 | 15

sikka.ai
tomorrow.io
Sourcegraph
netlify
INTERCOM

8 | 4 | 11

Frontify
lumapps

9 | 1 | 5

ADVENTR

1

Both

Canva
salesforce
stripe
miro
COHESITY
servicenow

Others Can Build on Product

Seed
Series C
Series D
100 Largest

Conclusion:
Key Takeaways

1. Providing a **minimum of dozens of product integrations** are now required for success in SaaS in most product categories. **Product category matters**.

2. In-app and public marketplace experiences help customers to discover and manage their integrations, **benefiting the host and partners**.

3. **Few app marketplaces are transactional**. But the ones that are tend to be large platforms in both a business and technical sense. Monetization models are still in flux.

4. **REST has taken over SaaS external APIs**. Developers are familiar with this style. Deviate only if there is good reason to.

5. **Webhooks are valuable** for efficiency and real time event notifications. Support them for the most important event triggers earlier rather than later.

6. **Implement OAuth 2** for partners as soon as possible. It increases visibility and is more secure.

7. Don't just aim to become a technical platform that everyone else builds on. **Aim to be interoperable** from day one in the way that most benefits the product category and customer base.

8. For scale, reduce friction in the **partner developer experience** - public API documentation, easy access to trying the API, and a well-designed developer portal.

9. There's a strong correlation between open ecosystems and SaaS growth.

Thank you for reading this report. If you have any questions, thoughts, or feedback on this report, we'd love to **hear from you**. You can also **learn more about us and our integration platform designed specifically for building SaaS product integrations and app centers here.**

pandium
integrate all the things

Appendix: Report Methodology

This companies included in the largest 100 SaaS companies for this analysis were taken from the Forbes Cloud 100 on August 2021 (**https://www.forbes.com/cloud100/#712110ea5f94**) and from a list of the largest public SaaS companies on January 2022 (**https://www.mikesonders.com/largest-saas-companies/**).

For the Seed through Series D stage companies, they were randomly selected from Crunchbase. Companies whose website was unavailable, was not available in English, or were no longer a freestanding product (usually through acquisition) were excluded. These companies were collected March 2022.

Because these lists were collected on different dates, one of the 50 largest private companies had gone public, which meant 99 companies were included in the analysis of the 100 largest companies.

In addition, one of the Series D companies was also on the top 50 private companies, which meant 99 Series D companies were analyzed. 100 Series C and 100 Seed stage companies were analyzed.

However one wants to define the 100 largest SaaS companies, these 99 companies certainly create a representative sample of that population.

When it comes to Series D to Seed companies, 100 companies is not enough of a sample to give high confidence in the precision of these numbers for the whole population. But given how strong the overarching trends are in this report, it supports the idea that this data is actually roughly representative of the larger population of companies at each of those stages.

As mentioned, for the purposes of this report, product integrations and apps are a pre-built means of programmatically passing data between two systems or an extension built by a third party on top of another system that the user or customer of the software can install and utilize.

Product integrations and extensions are different from custom or private integrations in that they are “productized” and thus available for use to at least a segment of, if not all, customers.

The information collected in this report is only based on publicly available resources, from websites, support docs, press releases, videos, and information the company provided elsewhere, like on review sites or in interviews. Information that could be obtained from a company with only an email login was also included.

When a company claimed to have product integrations, but didn’t identify most of them, they were excluded from the analysis of the number of product integrations.

Similarly, if a company did not have publicly available support documentation, product videos, or any comment on whether they had an in-app marketplace, they were excluded from the analysis.

When a significant amount of companies did not have public information on a particular metric, the data points are of more limited value. However, as mentioned, the trend lines in all collected metrics were strong, suggesting the data is meaningful in the aggregate.

For example, many Seed companies did not mention having an external API. Almost certainly most of these companies don’t have one. But some of them might, and the remaining companies create a more limited sample size when trying to determine, for example, what style of API Seed companies are using.

The data for the largest companies is the most robust as they are also the companies with the most open and well-documented ecosystems.

pandium
integrate all the things

Size of Companies in the SaaS 1000 vs 400 SaaS by Number of Employees