THE STATE OF

API INTEGRATION

REPORT

2019

Cloud Elements

Table of contents

A Word from the Editor

The world's first web page went live on August 6, 1991. It was dedicated to information about the World Wide Web project, and was created by Sir Tim Berners-Lee. It ran on a NeXT computer at the European Organization for Nuclear Research, CERN.

It is that web page that marked the start of a journey toward today's API economy.

By the end of the decade, the web has become a transformative power for businesses. On February 7th, 2000, Salesforce.com launched one of the first enterprise SaaS products. With it, the world's first web API - a new way for customers to easily share data between their various business applications.

Today there are more than 1.5 billion registered websites, 150,000 web applications, and 50,000 public APIs.

...on average, people responded with building **11.5 integrations** in 2018 and plan to build **18** in 2019.

Of course this growth doesn't come without growing pains. We now live in a world where web APIs - originally designed to ease application integration and data sharing - are actually creating barriers and challenges for developers. With an ever increasing number of APIs and applications, developers can't possibly be expected to keep up with the scale of growth or the pace of change. RESTful APIs have created a strong and lasting foundation for innovation, but we need more. More technologies and integration patterns. More security to increase trust. More standards to help interoperability. More analytics to inform decisions. More openness to drive transformation. More tools to ease integration. More governance to manage complexity. Perhaps we even need more regulation to guide our efforts. More, more, more!

As a community of API providers, practitioners, architects, designers, and consumers, there will always be opportunities for more - and this year's report will hopefully help you chart a course forward.

Cloud Elements gives businesses the freedom to innovate using Next-Ready API integration as a strategic advantage. Our virtualized APIs solve the growing challenges of connecting your customers and partners with your solutions. No matter what your use case, you are freed from the burden of developing and maintaining integrations so you can focus on what matters most - creating new products, penetrating new markets and building successful partnerships.

Happy Integrating.

Ross Garrett
VP of Product, Cloud Elements
@gssor

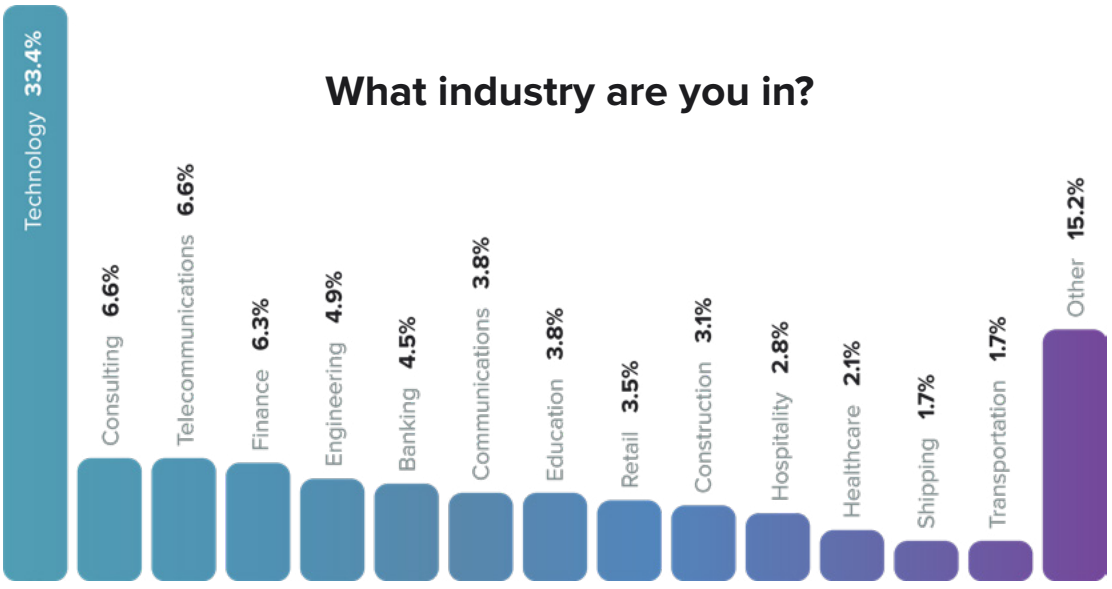
Cloud Elements

About the Report

Last year, Cloud Elements released the second State of API Integration report, which analyzed data from within the Cloud Elements API Integration platform, including over 1.6 billion API calls to 200+ endpoints.

We are taking the 2019 report to the next level.

This year's report expands upon the data from the Cloud Elements platform, featuring data collected from 350 API enthusiasts, in over 20 distinct industries, with annual revenues ranging from less than $100,000 to over $25 billion. Our second State of API Integration survey was distributed between September 2018 and January 2019 to representatives of these companies. Their responses contributed to the identified trends surrounding the existing demand for API integrations and the roadmap for leveraging integrations in the future.

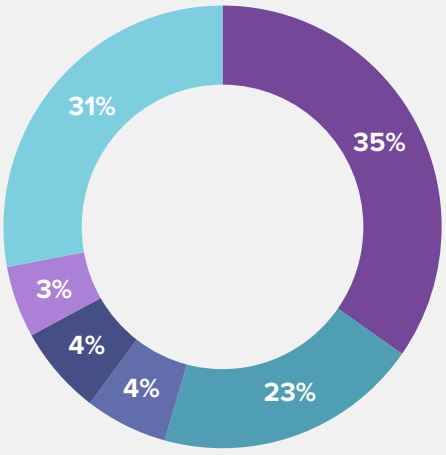


The report also includes industry expert insights from Ross Garrett, VP of Product at Cloud Elements, Shelby Switzer, API Activist, Mike Amundsen, API Author and Speaker, Mehdi Medjaoui, APIdays Founder, and Francois Lascelles, Field CTO at Ping Identity.

The 2019 State of API Integration Survey Respondents

Integrating the World of APIs

The analysis in the 2019 State of API Integration report represents data collected from companies across **50 countries** and **6 continents.**



What country do you live in?

- United States
- India
- Canada
- France
- Brazil
- Other

Meet our Contributors

Ross Garrett

VP of Product, Cloud Elements

@gssor

Ross Garrett is the VP of Product at Cloud Elements - responsible for market strategy, product management, positioning and evangelism. He is a well-known speaker at developer events and other industry conferences.

Shelby Switzer

API Activist

@switzerly

Technologist, public speaker, and veteran nomad, Shelby has dedicated herself to community work and technology activism across four continents. She is obsessed with APIs, semantics, and connectivity, and has led API design, development, and consumption projects in a variety of industries, from citizen engagement to the Internet of Things. Most recently she led the integrations team at Healthify, a software company bridging the gap between healthcare and social services. She now consults on API architecture, healthcare integrations, and civic technology, co-organizes REST Fest, and examines the intersection of public infrastructure and technology on her blog, **www.civicunrest.com**.

Mike Amundsen

API Author and Speaker

@mamund

An internationally known author and speaker, Mike Amundsen travels the world consulting and talking about network architecture, Web development, and intersection of technology and society. He works with companies large and small to help them capitalize on the opportunities APIs and Microservices present for both consumers and the enterprise.

Mehdi Medjaoui

APIdays Founder

@medjawii

Mehdi Medjaoui is the founder of APIdays conferences. Previously, he was co-founder of Webshell SAS, which includes OAuth.io and GetMateria.com. Mehdi is an author at Readwrite, and is also the co-author of the API Industry Landscape, The State of API Documentation 2017, and The State of Banking APIs 2017. Mehdi currently is the manager of the API Accelerator program at ReadwriteLabs, which launched February 2018. He also organizes numerous events such as the Parid API meetup, the APICraftSF meetup and local Hackathons.

Francois Lascelles

Field CTO at Ping Identity

@flascelles

Francois is a member of the Ping Identity Office of the CTO. He provides product and strategic direction to customers and partners with a focus on API infrastructures security and API cybersecurity.

Prior to joining Ping, Francois was the first developer and Chief Architect at start-up Layer 7 Technologies until its acquisition by CA Technologies. Francois was part of a team that developed a best of breed security gateway technology which disrupted a category. Francois helped define the application of emerging security patterns such as OAuth in the context of API Management and led a field practice of Architects helping customers with their digital modernization projects.

SECTION 01

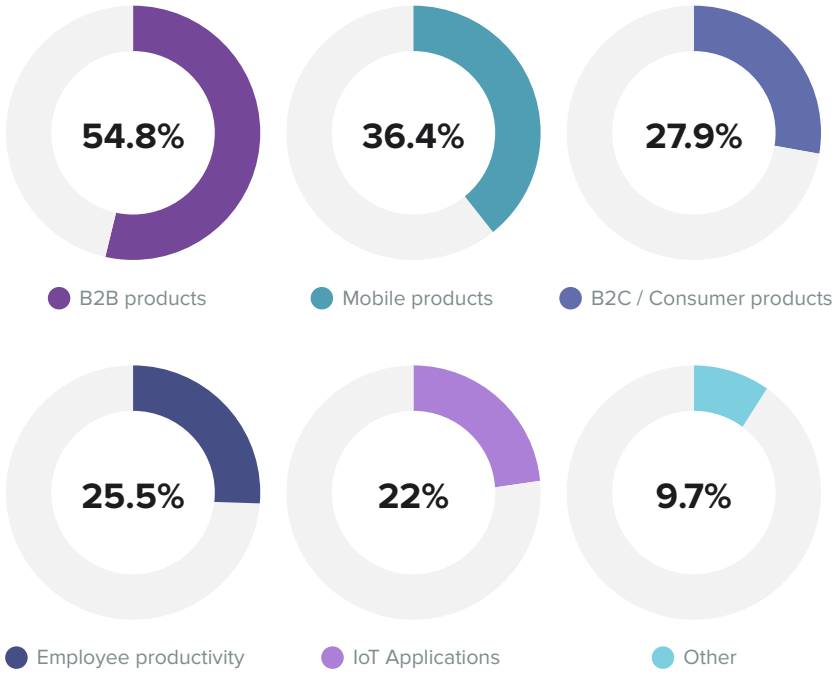
Introduction

Introduction

We launched the inaugural State of API Integration report in 2017, with a singular goal - gather and share the collective experiences of API providers and consumers, so that we may all serve each other better. Now in our 3rd year, this report has become one of the most widely read in the industry, and we've been able to reach more developers, architects, product managers and practitioners than ever before. For all those that responded to our survey and provided feedback, your help and support is much appreciated.

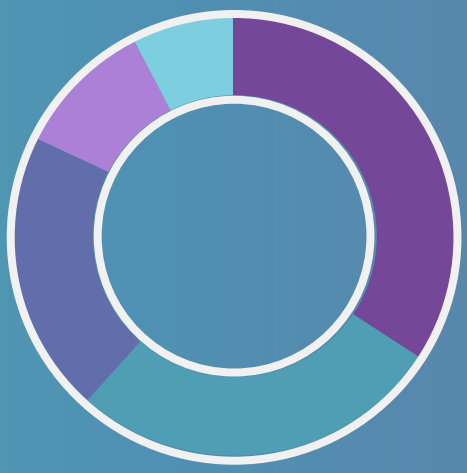
In last year's report, one of the key themes we uncovered was an increasing preference to pursue a platform strategy. We learned that APIs are the fuel which create a so-called "network effect" — meaning organizations must now look towards platforms that not only offer highly functional APIs, but also integrate seamlessly to the ecosystem of products and services used by their customers and partners. In short, API Integration must be a strategic endeavor, and investment must be made to ensure your business is architected for continued success.

What are you building with APIs?



The data we collected over the past 8 months further reinforces this sentiment, and so in this - the 2019 edition - we wanted to explore how various market verticals had started to embrace this notion of becoming a platform and begun to realize the network effects on offer.

As is now tradition, we've distilled all of that information into the most noteworthy stats, changes and trends in our comprehensive State of API Integration Report.



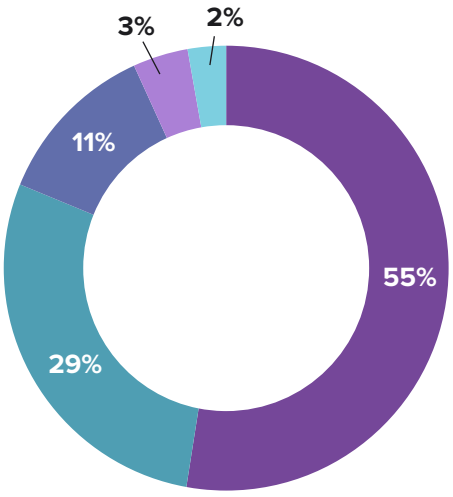
SECTION 02

Business Drivers for API Integration

Business Drivers for API Integration

For three consecutive years, the criticality of APIs and integration as a key component of business strategy has continued to increase. Businesses that have been investing in their API and integration strategy already are starting to realize some of the many benefits. We've highlighted three key benefits that are relevant to software companies, and the growing world of digital enterprises:

1. Churn Reduction
2. Product Stickiness
3. Faster time to market



How critical is API integration to your business strategy?

- Critical
- Somewhat Critical
- Neutral
- Less Critical
- Not at all

Proven correlation that external integration can reduce customer churn

For most software companies, the most cost-efficient way to sustain and grow revenue is by keeping current customers happy and growing those existing happy customers. Expanding your current reach can be done by growing user adoption and incorporating your customers' feedback onto your product roadmap.

A common ask from executives is to, "make our customers go faster" - which means enabling them to derive value from your product in the initial weeks of implementation. Your product is most likely going to be one of the dozens of SaaS tools in their current tech stack, so ensuring that your product can "plug-in" quickly is crucial. Simply put, if your product doesn't play nicely with the most important apps your customers use to be successful, it will be an uphill battle for your product to be sticky.

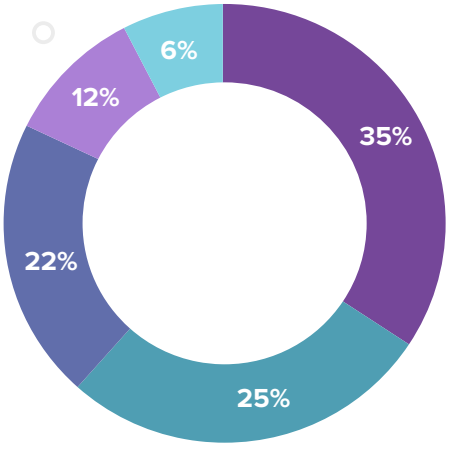
For example, a typical marketing team at a software company alone has close to 50 apps in their tech stack. If a new app the team is considering to purchase doesn't connect to their key tools (i.e. marketing automation software like HubSpot) they won't purchase that tool. It's that simple. You may be thinking, "what about third-party integration tools like Zapier?" We hear you. There are multiple issues with Zapier-like solutions. These third-party citizen integrator solutions:

- Require additional cost to your end-users on top of your subscription fee
- Force your end-users out of your UI and can negatively impact your UX
- Can't handle true, robust integration functions
- Take away potential add-on revenue earnings from you
- Are often inflexible and rigid

Providing a seamless integration experience or application marketplace to third-party apps like CRMs, ERPs, and HRIS systems makes your product more valuable than just being a stand-alone offer. Across Cloud Elements' various software and enterprise customers, they have recognized an average of **60% reduction in customer churn rate due to integration.** Reducing customer churn, even by a small percentage, is fundamental to the growth of a software company and this priority can have a significant impact on your organization's bottom line.

In addition to reducing customer churn, providing external integrations leads to potential upsell and upgrade opportunities of existing customers.

Approximately what percentage of your user base will upgrade or renew if you provide the integration they need?



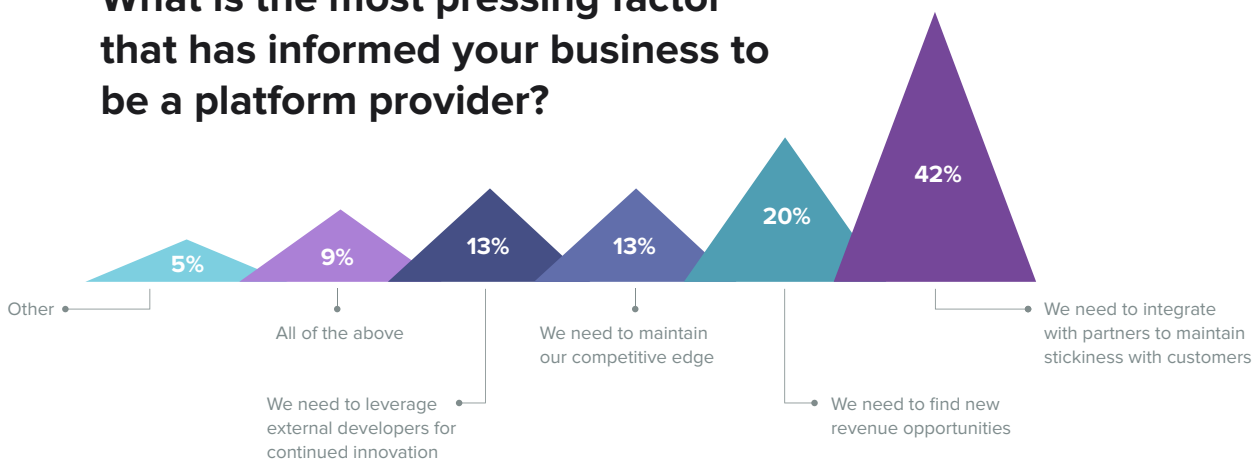
Improve Product Stickiness

What comes to mind when you think about software that has an enjoyable user experience? Typically it's the simple design and ease-of-use which can mean no configuration, no coding, and drag and drop simplicity.

A perfect example is Slack, a collaborative messaging app used by teams for an easier and better way to communicate. For many SaaS companies, it's hard to imagine working without Slack. And why is that? It's because of Slack's incredible product stickiness and lovable UX. It's easy to invite new colleagues to your organization, it's easy to set up, and it's easy to integrate with other popular applications such Salesforce, Google Drive, and Asana.

When you offer a white-labeled integration experience that lives directly in your app (such as Slack's impressive integration app directory), you automatically become more enjoyable and easier to use to your business users. No longer do your customers have to deal with clunky file transfers or writing to your API, they get the data they care about to and from your app.

What is the most pressing factor that has informed your business to be a platform provider?



Accelerate Time to Market

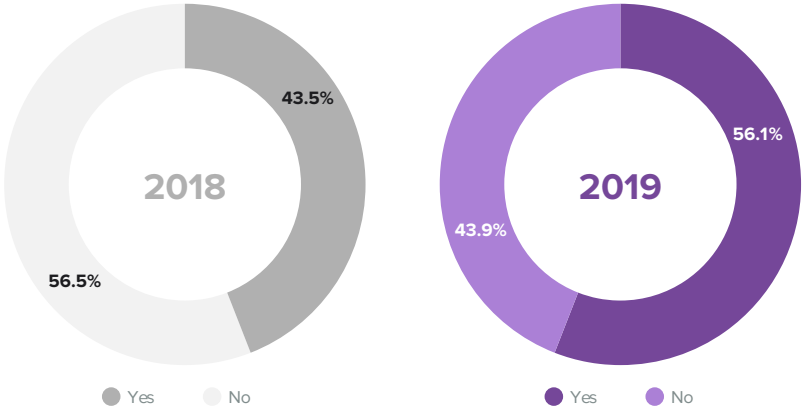
As we discussed in priority #1, a great way to grow your revenue is to offer your existing customers something else to buy (also known as add-on revenue dollars) since we know that current customers have a higher propensity of buying add-on products than new customers.

In fact, according to recent findings from Invesp, **your current customers are 50% more likely to try new products and spend 31% more**, when compared to new customers.

There are plenty of ways you can monetize embedded integrations in a way that serves your business:

- Categorize integrations into pricing tiers based on the level of complexity, function, and maintenance such as Standard, Advanced, and Enterprise.
- Charge based on consumption like number of instances per customer or API calls.
- Charge customers an additional acceleration fee to bring a new integration to market.

Does your organization charge for API access or integration services?



You may be thinking,

“Offering integrations to my customers sounds great, but who is going to build, deploy, and maintain them?”

It’s understood that most software companies have limited developer resources and a backlogged roadmap that includes product requests, bug fixes, and new features. That’s where the Cloud Elements API Integration Platform comes in. Hands down, the best way to accelerate your time to market is by giving your devs a scalable, easy-to-use, and trusted platform to quickly build out integrations. The goal is to prevent distractions and get integration tasks off of your dev’s backlog, which frees up their time to focus on building core product features. Our customers’ have seen **savings of up to 66% in development costs** within the first year of partnering with Cloud Elements.

SECTION 03

Focus on the Data You Care About

Focusing on the Data You Care About

To become a successful digital business, select a data and application integration strategy that enables you to unify all your company's data assets and analyze them in context to get the full picture of your business.

Your Data Has No Master

With such a rapid proliferation of applications in an enterprise, it's a challenge for IT to centrally master data. There is no longer just one answer to what a "Customer" object looks like. Every department - and even end users - are customizing the data models in their SaaS applications to capture their unique data. You can't duplicate data, and even the concept of the golden record is quickly being worn down by the limitations imposed in a Master Data Management (MDM) system.

Limitations to the Physical Data Structure

The current approach to integration requires that you learn the structure of your data at each individual endpoint. You integrate by becoming an expert with the object models of dozens, hundreds and thousands of objects at each endpoint. This doesn't scale.

You shouldn't have to be an expert in each application's data model. Even when an integration platform provides templates, or even intelligence in the data mapping, you're still operating at a point-to-point perspective dependent upon the data model at each endpoint.

Virtual Data Hub

A Virtual Data Hub applies an abstraction layer separating the logical view of data from the physical representation. Abstraction of data is not a new concept, but it is just recently being applied to the world of APIs. Your company's data is physically stored across dozens, hundreds or even thousands of applications and databases that may be in your data centers, in the cloud or in each SaaS application vendor's cloud. Your revenue data is in a CRM. Your marketing data is in the dozens of applications used by your marketing team - many of which are now in the cloud. Is there one source of truth for what a customer, employee or product object looks like across all endpoints?

Each application and application vendor has its own point-of-view of each data object but what is your point-of-view? A Virtual Data Hub gives you the ability to establish your point-of-view of a data object as the only view that matters. By abstracting your view of the data from the underlying application's physical view, you can begin to govern and manage your data regardless of each application's data structure.

Open Data Initiative

Enterprise software giants SAP, Adobe, and Microsoft jointly announced an “Open Data Initiative” in September 2018. The trilateral coalition between three of the world’s most influential enterprise software companies tackles one of the key challenges all enterprises face: siloed data. The initiative aims to enable standardized data exchange and interoperability among different platforms via a common data model.

...the Open Data Initiative, which is a common approach and set of resources for customers based on three guiding principles:

1. Every organization owns and maintains complete, direct control of all their data.
2. Customers can enable AI-driven business processes to derive insights and intelligence from unified behavioral and operational data.
3. A broad partner ecosystem should be able to easily leverage an open and extensible data model to extend the solution.

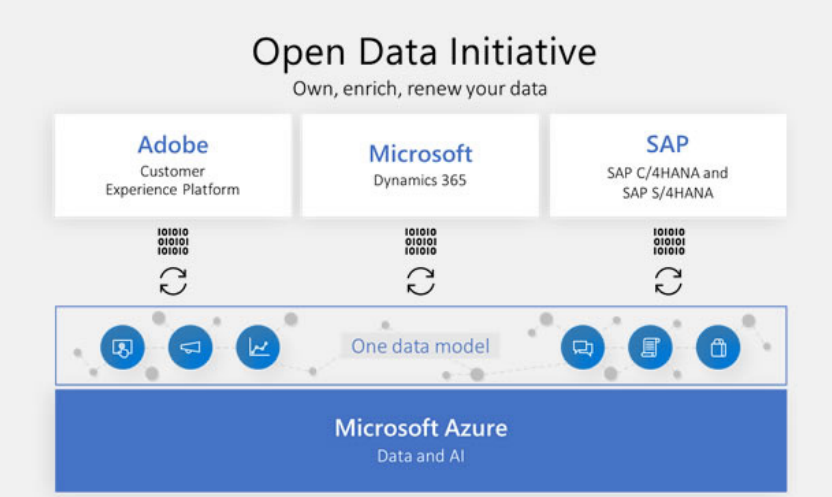


PHOTO CREDIT: https://www.microsoft.com/en-us/open-data-initiative

The key problem the Open Data Initiative is trying to solve is interoperability of data and providing a unified repository for customers to take ownership of their data. And this collection of software companies represent several important data management use cases. For example, If you are a retailer, you likely have customers who buy from your website and store. They then go to social media to review your products, say good and bad things about it, and perhaps even raise complaints with your call center. Today, all these different systems — your POS software, website, CRM, social media channels — build customer profiles and transaction data using their own format and structure. They have different repositories and different architectures and it is not a trivial exercise to link them across channels and systems.

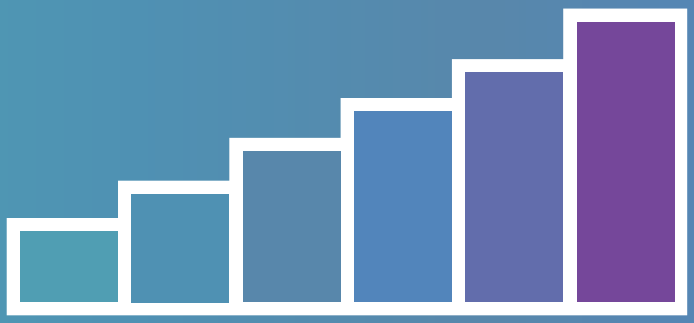
The Open Data Initiative seeks to address this issue by providing a common data model among some key vendors (along with a common language and set of APIs as the initiative matures) and then a data lake to store information that can then be used by other applications to run analytics, campaigns, or other data-heavy applications.

Increased Demand for Data Standards

We have heard that developers are becoming increasingly interested in more data standards, but this approach has had varied success in various markets.

Interoperability provides a strong case — allowing you to easily integrate with other enterprise applications, and enabling faster time to market because you can buy applications off-the-shelf, create portable applications, and so on. But in reality, they suffer due to lack of ongoing support from sponsoring vendors, and several other technical and non-technical reasons.

That doesn't mean this Open Data Initiative will meet the same fate, but these are early days, and we'll start to see real deliverables in 2019. Also, for a standard like this to be successful, more vendors need to support it. This includes IBM, Oracle, and Salesforce, all of whom are missing today.



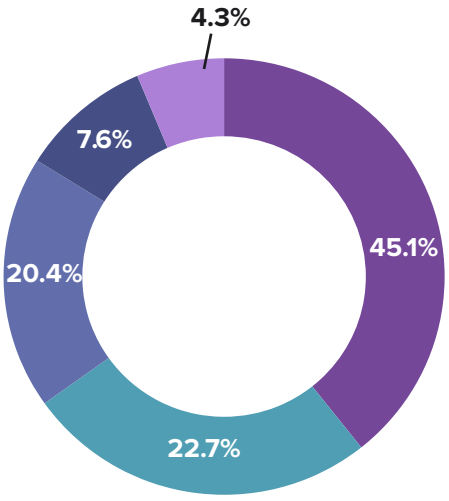
SECTION 04

Understanding App Ecosystems

Trends Across Market Verticals

After capturing various macro trends in 2018, we recognized a need to investigate specific use cases and market verticals a little deeper.

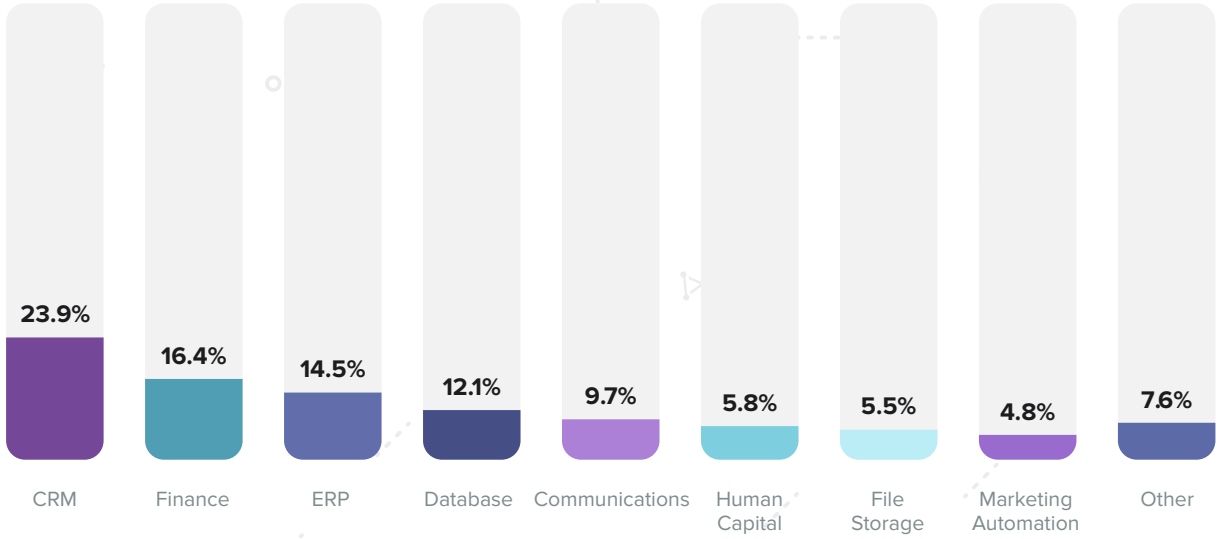
What we've found is that for many, CRM and Marketing use cases are still the most common - but new use cases are starting to gain ground. Over the past 2+ years, there has been significant change in the domain of integrated accounting and cash management workflows, driven in large part by a swathe of new Fintechs that are making it easier to solve problems across Accounts Receivable / Accounts Payable, Expense Management and of course Supply Chain Management. As this diversification continues, we're seeing strong growth in Human Capital Management - an area that has also benefited from massive venture capital investment and broad market interest.



Which of these use cases aligns to your integration?

- Lead to customer
- Customer to cash
- Lead to loyalty
- Quote to cash
- Hire to retire

Which category of integrations is most needed for your business?



Fintech and Open Banking

Integrating the Open Banking Ecosystem

Digital business in any industry is about creating and integrating an extended ecosystem to better serve customers, partners and even employees. Over the past five to seven years, we've seen that organizations have come to understand that APIs are fundamental not just to their technology strategy, but their overall business strategy too. While many banking and financial institutions have moved slowly towards an open API strategy, new regulation and market forces are now demanding that they move more quickly.

Fintech firms already play a significant role in shaping the banking platform of the future, largely because long standing financial institutions

are constrained by legacy systems that can stifle innovation and agility. However, fintech firms lack the customer base of the banking industry, and banking firms have the advantages of stability, trust and experience navigating regulations and compliance requirements in addition to access to significant capital.

Banks must transform into platforms that enable the applications and services their customers want to use. They need simple, user-friendly and reliable APIs to offer developers and partners the opportunity to innovate.

Three key reasons why financial institutions need to execute quickly on their API strategy:

1. Regulatory pressure
2. Collaboration and engagement with Fintech
3. Building new digital products

Regulation and PSD2 / Open Banking

The European Commission's second Payment Services Directive (PSD2) became legislation in November 2015 and is designed to create a single and efficient market for payments.

Repealing what is now being called PSD1, PSD2 introduces change on a number of levels, but the most attractive feature is called "access to accounts" (XS2A). This provision requires account holding institutions (typically existing banks) to allow access to their customers' account information to facilitate payment initiation and account information services provided by newly regulated so-called TPPs.

XS2A is the reason PSD2 has been hailed as the development which will lead to the disintermediation of incumbent banks at the hands of upstart new entrants, all set to dominate customer engagement and leave the traditional players scrabbling for scraps of low-grade commoditized back end account servicing business. Within the EU, it will likely revolutionize the payments industry, affecting everything from the way customers pay online, to what information they see when making a payment.

PSD2's three main objectives:

1. Create a safer and more innovative payment ecosystem across the EU
2. Make cross-border payments easy, efficient and secure
3. Increase competition and choice for consumers

In the Americas and Asia, open banking thinking is becoming pervasive, with banks in these markets looking at the impact and implications of PSD2 and increasingly viewing openness as a prerequisite to drive innovation going forward. The payments and financial services businesses are international, and it's important to retain a global perspective when thinking about the progression of open banking.

It's All About the Ecosystem

Ultimately, open banking is a necessary response to the changing requirements of banks' customers in this digital age. Few if any financial institutions will enjoy being relegated to providing commoditized back end processing but, all will be motivated to expand the relationships they have with their customers.

While customers want the innovative services PSD2 and open banking are designed to enable, it's also true that the banks need those innovative services to ensure the complete set of their customers' demands can be met. This is why it's critical for banks to engage with their ecosystem if they want to guarantee their future success in digital - and open - banking.

There's been an increase in collaboration, especially with fintechs, as these organizations engage third-parties to propel innovation in order to stay competitive. But, staying competitive isn't as easy as simply creating an open API. Thousands of banks and fintechs are starting to offer identical services that do nothing more than increase the standards and leave additional room for the outliers to creatively innovate. Few financial services orgs have succinct visions of how to take advantage of PSD2 and they're achieving this through API connectivity.

What to expect in 2019

After much lobbying and negotiating the final deadline to comply with PSD2's Regulatory Technical Standard (RTS) is September 14th, 2019 - just a few short months away.

However, there is another deadline for banks to manage, which is perhaps even more significant - March 14th, 2019.

By then, banks must have their dedicated interface (Open API) ready for testing by PISPs and AISPs. Article 33.6 of the RTS1 states that banks which aren't ready for testing by this time must instead provide a 'contingency mechanism' which, for many, will likely mean formalizing screen scraping!

This route has negative implications for everyone. To begin with, screen scraping poses a significant security risk: it means the security

credentials of banks' customers are shared with third parties who, if breached, could compromise all their customers' online or mobile banking facilities.

Secondly, maintaining more than one interface drastically increases costs for the bank; each interface will require strict and ongoing monitoring and reporting to their local authorities. For tier two banks and challenger banks, this will further compound the serious RTS compliance burden that already includes delivering secure customer authentication, managing exemptions, identifying and managing TPPs, developing the testing sandbox, creating documentation, etc.

Overall, it makes the most sense for banks to focus on supporting one, secure, RTS compliant open API.

Pulling It Together

Open Banking and PSD2 is an enabling technology for integrated payments - driving a new digital, and seamless experience for your customers. The ultimate goal is the provision of a unified experience for customers through one interface – one seamless end-to-end journey to the desired customer outcome – with an API integration platform.

---

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018R0389&from=EN

OVERCOME LEGACY IT RESTRICTIONS

Unfortunately, long-standing financial institutions are constrained by legacy backend systems that were not designed for today's digital economy. Often reliant on enterprise architecture technology that ably served internal integration needs for two decades, IT leaders are now unable to effectively deliver cloud and mobile integration that users of today expect.

Over the years, technology within financial institutions has largely evolved with an add-on mentality. As new technology is developed, point-to-point integration has been used to connect a new solution to each existing application. This method is ineffective and adds complexity. For example, the integration of 20 applications using a point-to-point approach could involve the creation and maintenance of 100s of individual interfaces.

'PLATFORMIFICATION' OF BANKING

A new approach to banking technology uses an application integration framework to overhaul the way individual solutions are connected, supporting comprehensive, real-time information sharing. This advanced integration environment brings together legacy and cloud services to create a new digital banking platform.

An application integration framework operates behind the scenes, and yet it can make all the difference in delivering a digital customer experience. And for a bank's IT staff, it could mean being saved from a "rip and replace" event by instead extending the utility of current banking technology into a platform with the flexibility to continue meeting their evolving needs.

Human Capital Management

Creating an Integrated HCM System

Disconnected business applications have always created barriers for employees across any organization. Yet, as many disciplines have been improved and integrated, HR systems have remained siloed.

With few exceptions, not many people spend most of their time working directly in the HR system - and as a result these systems are often completely disconnected from processes across other parts of the business. For example, if you need to get approval for a purchase order but the approver is on vacation, you will need to know their manager. This means you must move over to the employee directory or org chart, find the person's name, title and email, then switch back to the PO system.

What's needed is an integrated system that lets people perform their employee-related tasks without moving between environments or taking users out of context.

Fragmented HCM Ecosystem

According to ADP, global employers currently manage an average of 33 payroll systems and 31 HR systems. And in an age of disruption, these numbers are getting larger - business and HR leaders are being pressed to rewrite the rules for how they organize, recruit, develop, manage, and engage the 21st-century workforce.

This workforce is changing. It's more digital, more global, more diverse, and technology savvy. While this could be seen as a challenge, it should be seen as an opportunity to:

- Reimagine HR, talent, and organizational technology
- Create platforms, processes, and tools that will continue to evolve
- Take the lead in transforming how your business and your team views HR
- Integrate and modernize HCM to provide more accurate data

The vendor landscape is undergoing a seachange: A new breed of HCM products and solutions are coming to market, many built around mobile apps, AI, and consumer-like experiences. As digital HCM takes hold and HCM products become more platform based, all users are becoming empowered to reach levels of efficiency not previously possible.

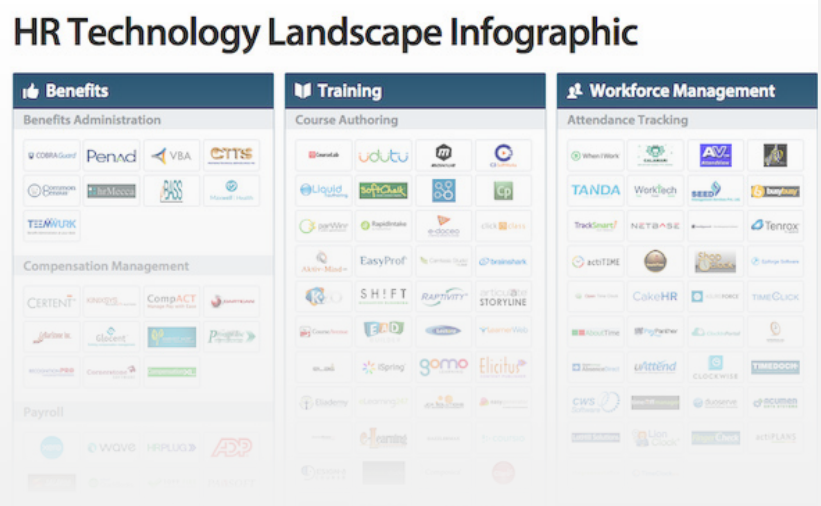


PHOTO CREDIT: CHIEFMARTEC HR SOFTWARE LANDSCAPE 2017

How Integration Drives HCM Experiences

Building integrated Human Capital Management solutions will be a journey, and existing silos must be broken down. As these once siloed functions are distributed across best-of-breed applications rather than a single HR application suite, new integrated experiences are not only possible - they're needed.

The value integrated HCM can deliver:

1. An integrated employee directory should provide a quick and easy way for employees to find the needed information to contact the right person at the right time - without switching context from the task they are trying to complete.

2. Payroll should be a full-service system tightly integrated to ERP and accounting systems features so employees' time entry, attendance, and commission data translates directly to their payroll with no manual data re-entry.

3. Integrated time-off management should automate employee time-off requests and approvals, with a 360-degree view of business demands that may need to be considered when reviewing PTO.

4. Compensation tracking that allows organizations to integrate compensation details such as earnings, pay frequency, overtime rate, and any variable compensation such as bonuses, or stock options.

By defining a world-class HCM strategy and implementing the right integration platform, any size business can become a more effective organization that better leverages its most important differentiator and asset – its people! At the center of this journey is a holistic view of the employee data your organization cares about.

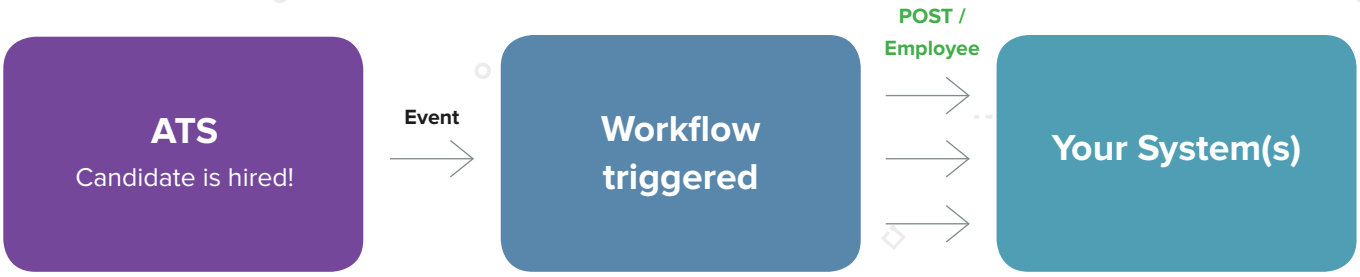
Hired Candidate Workflow

Use Case

When a candidate status changes, actions need to be executed in one or more applications.

Solution

Cloud Elements facilitates this use case through our normalized **eventing framework** to automate the process. We detect the change in your - or your customers' - Applicant Tracking System (ATS) and provide you with a normalized event, feed your application's API or trigger a workflow.



Healthcare

Contributed by Shelby Switzer, API Activist

The healthcare interoperability landscape has been changing rapidly since 2014.

While five years may seem like an age in other industries, and spectators from those industries might view “rapidly” as an overstatement, this is a brisk pace for an industry whose major players include forty-year old software vendors and the federal government. A combination of advancement in standards and tooling, forward-thinking legislation, and outside startups eager to get a slice of the $3 trillion pie[1] have ushered in a new era of API-first methodologies, transparency, and open standards.

The most progress in terms of open standards in healthcare can be seen in the open data model and API standard known as Fast Healthcare Interoperability Resources (FHIR), as well as increased adoption of OAuth 2.0 as the authorization strategy of choice. First released in 2014, FHIR reached its fourth release as of December 27, 2018. Previously breaking new ground simply by supporting JSON, the standard now also supports RESTful hypermedia implementation through extensions and its documentation even offers guidance on implementing GraphQL with FHIR.

FHIR is the poster-child of interoperability in healthcare and indeed is increasingly making an appearance in the marketing and documentation of major healthcare software vendors. The entrenched vendors in the healthcare technology space, however, would have little reason to support FHIR or interoperability at all, if not for recent legislation. In 2015, the Office of the National Coordinator for Health

Information Technology (ONC) mandated that APIs be a requirement of certified healthcare IT systems.[2] Just a year later, the 21st Century Cures Act included specific provisions to protect and support interoperability of healthcare technology.[3] Companies that provide over 90% of electronic health records have made the "Interoperability Pledge," committing to enabling simple consumer access to their own electronic health information and implementing interoperability standards.[4] Now, as of February 2019, the ONC and the Centers for Medicare and Medicaid Services (CMS) have proposed new rules that explicitly require standards-based APIs.[5]

It should be telling that a new rule requiring standards-based APIs is necessary despite previous legislation and the widely touted Interoperability Pledge. For those of us on the ground, integrating with EHRs is still a long, demanding process that includes VPNs with on-prem legacy systems and high levels of customization per interface and data feed. The adoption of an API-first approach amongst the EHRs, who are the gatekeepers of electronic health information, has been slow going. Rather than fully opening up APIs, some EHR vendors have gone the route of app stores or app exchanges. Epic's App Orchard and Cerner's App Gallery, for example, showcase applications that have been developed against their APIs that purportedly follow FHIR standards, although the pricing of such a partnership is often opaque or beyond the budget of many startups.

Startups are not an insignificant player in all of this. A key concept in both the value-based care movement and Meaningful Use Act is improving patient care and the underlying belief that innovation is necessary to do this. By opening up the electronic health record ecosystem via interoperability, that same ecosystem is opened up to innovation, from startups as well as established companies previously focused on other industries.

This new blood brings new expectations. Software developers used to working at SaaS companies expect modern APIs from the systems they integrate with, and at a minimum this means well documented APIs using JSON over HTTPS. Furthermore, when building integrations, they have a product mentality and expect to maintain a relatively agile product development process. The reality they face once they begin their first EHR integration often induces culture shock: instead of developing products, they embark on month-long projects transferring data as flat files or HL7, an EDI-like standard that is implemented with such variety that the word "standard" can feel like a misnomer. The hope is that the increased adoption of FHIR and the availability of documented APIs, albeit via app stores, will gradually change this reality.

In the meantime, the need for better interfaces and developer tools for a modern cloud-based world has ensured that API tooling companies such as API management layers and interface engines have a seat at the healthcare technology table. Companies like Redox and PokitDok speak the same language as web developers and offer a standardized data model and JSON API along with open API documentation and tooling as well as expertise and education to help developers build healthcare integrations more quickly. These companies have the opportunity to help shape the standards that evolve and become valuable pieces of healthcare technology infrastructure.

Interestingly, the federal government also plays in important role in shaping the industry not just though legislation but through implementation. Software projects at CMS such as the Blue Button 2.0 API,[6] which uses FHIR and OAuth 2 to enable Medicare beneficiaries to access and connect claims data to authorized applications, services, and research programs, set forth a model for the industry that companies in the private sector must follow if they are to work with government – which, in healthcare, they must. This spells good news for open standards like FHIR and OAuth and best integration

practices like HTTP APIs: these methodologies will more likely proliferate through implementation rather than mandate.

We're still a long way from a world where EHR vendors can boast truly open and interoperable APIs and where healthcare tech companies can focus on building impactful, integrated products rather than one-off, burdensome projects with individual health systems or health organizations. However, the recent shifts in the industry have been seismic in scale and impact, and over the next five years changes will become even faster and more frequent. The digital transformation of healthcare tech into a modern API-first industry has begun.

---

[1] https://www.forbes.com/sites/danmunro/2012/01/19/u-s-healthcare-hits-3-trillion/#543237af3da8

[2] https://www.healthit.gov/topic/certification-ehrs/2015-edition

[3] https://www.fda.gov/RegulatoryInformation/LawsEnforcedbyFDA/SignificantAmendmentstotheFDCAct/21stCenturyCuresAct/default.htm

[4] https://www.healthit.gov/topic/interoperability-pledge

[5] https://searchhealthit.techtarget.com/news/252457399/ONC-CMS-drop-information-blocking-interoperability-rules-ahead-of-HIMSS

[6] https://bluebutton.cms.gov/

SECTION 05

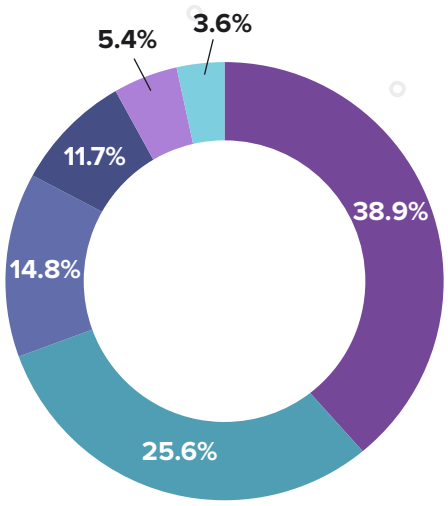
Developer Trends & API Design

Developer Trends & API Design

API integration is simply a means to an end; building an integrated experience where data flows seamlessly between the apps you and your customers or partners care about.

Yet, API Providers are still not making it straightforward for developers or practitioners. Connecting to an API may be easy, but building a scalable, robust and extensible integration is often far harder than it should be.

Beneath the tip of the iceberg, developers continue to rank many of the same areas as pain points in API integration.

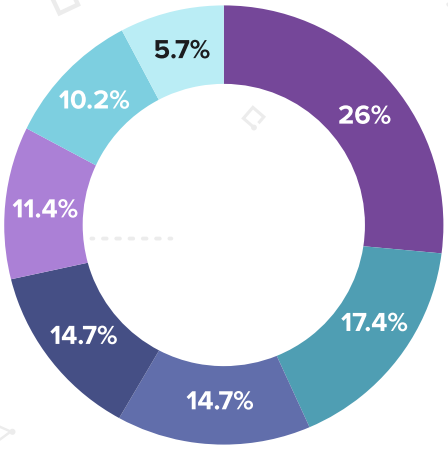


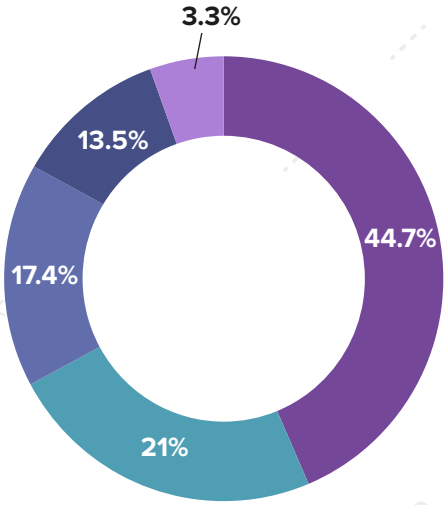
Which API integration challenges are most frustrating?

- Event Driven Integration
- Authentication & Authorization
- Security
- Discoverability
- Versioning
- Other

What part of the API integration development is the most time consuming?

- Workflows
- Custom Objects
- Authentication
- Transformation
- Events and Polling
- Error Handling
- Other



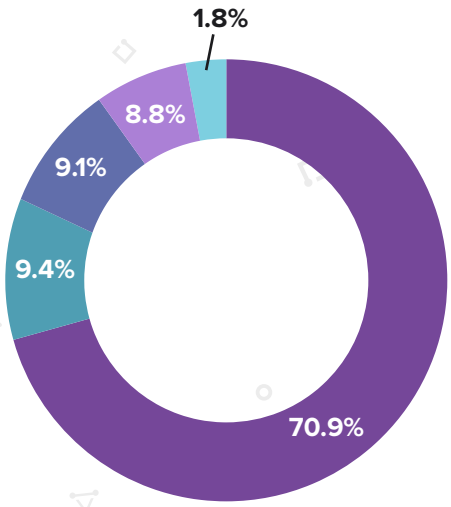


How would you improve the developer experience or adoption of your APIs?

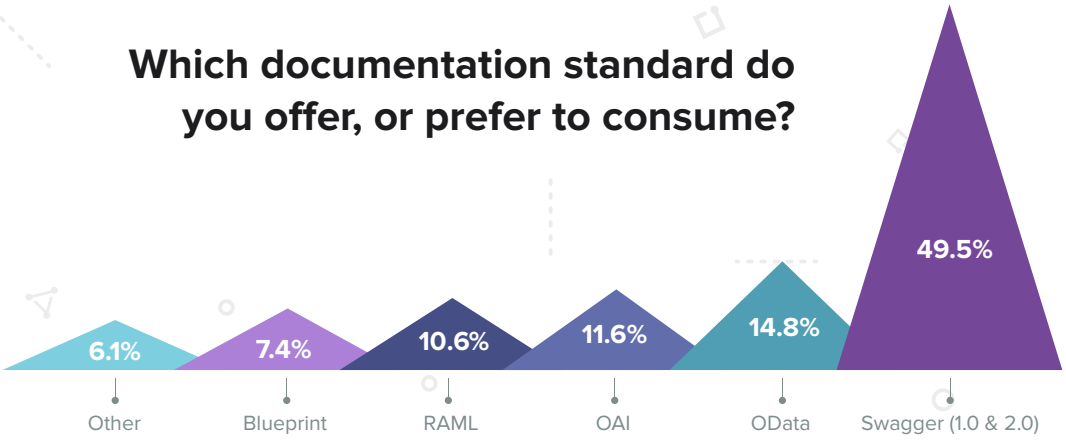
- Customized APIs that fit a specific business need
- “No Code” integration templates
- Better documentation
- SDK wrappers for APIs they need and use
- Other

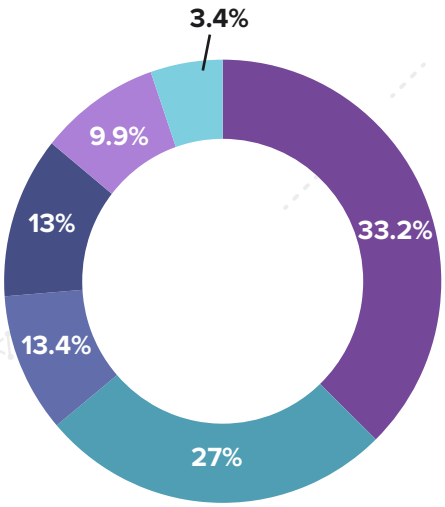
Which API style do you offer, or prefer to consume?

- REST (without hypermedia)
- Hypermedia
- GraphQL
- SOAP
- Other



Which documentation standard do you offer, or prefer to consume?

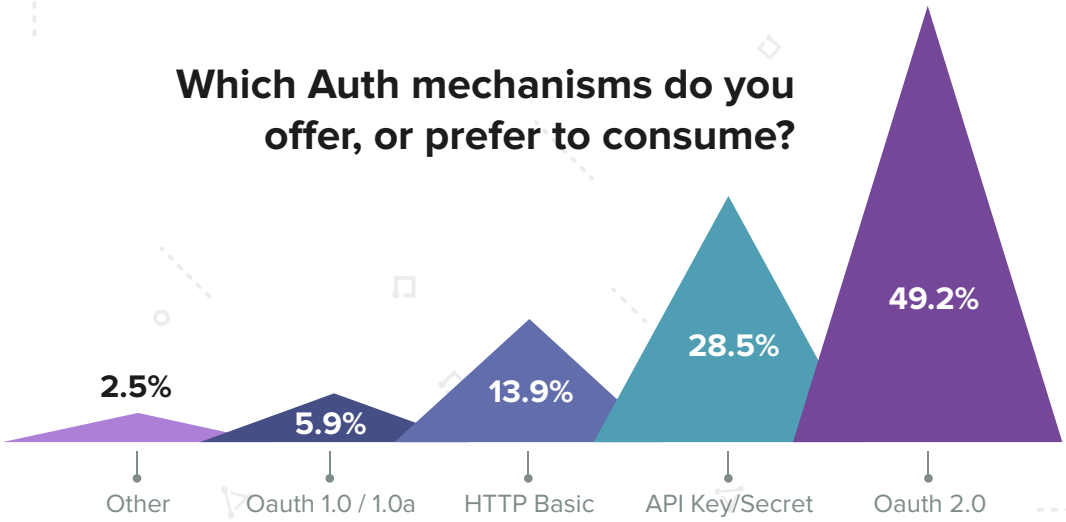




Which event-driven integration patterns do you offer, or prefer to consume?

- WebHooks
- WebSockets
- All of the above
- Server-Sent Events
- Polling
- Long Polling

Which Auth mechanisms do you offer, or prefer to consume?



The Challenge of API Design Today

Contributed by Mike Amundsen,
API Author and Speaker

Recently, a new front has emerged in the battle to gain strategic and technological advantage through the use of APIs.

And that front is represented by the increased call for training and guidance in how to integrate existing APIs with the rest of the company's IT investments. It is no longer enough to simply have APIs. It is now recognized that one of the key advantages of a strong, stable of APIs is that they can help you integrate and enhance your long-time IT investments such as mainframes, SOAP-based services, and mobile-centric applications into a rich and agile solution landscape.

But, in order to be successful in this new API landscape, it is important to establish principles and practices that move beyond simple API functionality and lean toward safe, effective, and scalable integration without slowing the pace of innovation or adding unnecessary costs. I'll cover three trends and challenges I've been observing over the last year or so as companies work to deliver on the promise of effective and efficient API integration. They are:

1. Consistent API Design
2. Action-Oriented APIs
3. Reducing the Cost of Change

Along the way we'll see examples of companies taking a leap forward from simply adopting APIs to leveraging APIs.

The Power of Consistent Design

One of the key challenges of managing a growing API program is sustaining consistent designs across teams and throughout organizations. This is especially true when companies follow the "two-pizza teams" mantra attributed to Amazon's Jeff Bezos [1] and work to empower small teams (3-7 people) with the ability to make appropriate decisions without first getting approval from a central architecture committee.

Fast and Stable API Design

While empowering small teams is listed as a key to success, these same small, independent teams can introduce unwanted variability in API design and implementation. The tension here is between "moving fast" and "breaking things"—another common meme used in API and Microservice communities. In fact, Facebook's CEO Mark Zuckerberg offered the now classic line in 2010, "Unless you are breaking stuff, you're not moving fast enough."[2]. However, in 2014, Zuckerberg said, "We've changed our internal motto from 'Move fast and break things' to 'Move fast with stable infrastructure.'"[3]

In the world of API design and integration, an important way to achieve the "stable infrastructure" Zuckerberg talks about is to provide a consistent design method to all your teams. This leads teams into using similar patterns and practices, allows them to better match their results against other teams in the company, and offers leadership a better way to track progress across the organization.

API Design Methods

API design methods that I'm seeing in companies today outline a design process that all teams use to create their APIs and they list design products that document their work in a consistent,

measurable, and re-usable way. Typical design process models start with gathering user stories to learn just what activities are included in a task that is the target of the API. Agile, Scrum, and other less formal models all include some mix of interviews, observations, and surveys to gather data.

These stories are then expressed as sketches or prototypes for stakeholders to review and revise until an acceptable design emerges. Along the way, it is common to create simple visual elements such as sequence diagrams that show the API workflow independent of any technical details such as HTTP protocol or JSON formats.

This final design then gets turned into a general description document. Increasingly this API description can be rendered in a formalized machine-readable format like ALPS [4], DCAP [5], JSON Home [6], and other formats.

Lastly, the output from these activities is tracked within DevOps pipeline tools and the results are posted on dashboards where anyone interested can see them.

The Rise of Action-Oriented APIs

Another key trend gaining traction inside API programs is the adoption of “action-oriented” APIs. Action-oriented APIs track well with the growing practice of applying design-thinking to the process of API design. According to Tim Brown CEO of IDEO, the goal of design thinking is to “match people’s needs with what is technologically feasible” and business-viable [7]. I’ve been seeing a handful of common approaches to implementing action-oriented APIs over the last year and I expect to see more of them in the near future.

Four I'll highlight here are:

1. Experience APIs
2. Event Sourcing
3. Remote Data Access
4. Hypermedia

Experience APIs

One of the most common ways action-oriented APIs appear within an organization is through what is call the "API Experience Layer" [8] or, as Netflix refers to them: "Experienced-based APIs" [9]. In this approach, API consumers are grouped by common technologies, skills, and use cases and each group is provided their own purpose-built "experience API" that closely matches their needs and skills.

Event Sourcing

Another example of action-oriented API design is the rise of event-based and reactive programming models for APIs. In these models, APIs are expressed as events that are published by providers and subscribed-to by consumers. The contents passed between services are typically sent as standardized structured messages (not just domain objects). Often called event-sourcing [10], this approach works well for "inner API" designs meant for consumption between internal services behind the boundary firewall.

Remote Data and GraphQL

Another trend in API design is the return of remote data interfaces as a way to expose existing data models in a consistent and still flexible way across the organization. Recent examples of this remote data access pattern are GraphQL [11] from Facebook and Falcor [12] from

Netflix. However, other data-centric formats in use are Resource Description Framework or RDF specifications [13] with SPARQL [14], OData [15], and JSON API [16].

Hypermedia APIs

The last trend that continues to make inroads in API platforms is the use of hypermedia-style APIs. Hypermedia APIs rely on similar techniques as both Event Sourcing (structured messages) and Remote Data Access (API consumers control filtering and other aspects of responses from servers). Hypermedia APIs usually support one or more formats designed to return not just data but also metadata (instructions) on how to craft queries and update operations on the fly. There are almost a dozen of these formats that have been created in recent years.

The most commonly used ones are HAL [17] and Siren [18]. Amazon Web Services, for example, uses the HAL format for it's "Amazon API Gateway REST API" [19]. Apigee's IoT platform, Zetta [20], relies on the Siren hypermedia format.

It is interesting to note that The Spring Framework series from Pivotal announced their Spring HATEOAS 1.0 release in March of 2019 which makes hypermedia formats a key implementation element in the framework.

Reducing the Cost of Change with API Design

Another challenging area for API programs is dealing with the cost of change over time. This is commonly expressed as the "versioning problem". The real challenge is how to support the inevitable feature and workflow updates without the need for creating multiple incompatible forks (or versions) of an existing API.

Early in the days of APIs, service providers would simply alter an existing API and give API consumers a deadline for updating their

own client apps to comply with the provider's changes. As API programs grow, and as APIs increasingly become dependent on other APIs, the practice of burdening API consumers with the bulk of the change costs does not scale. This is especially true when companies are consuming APIs that they do not control (third party APIs).
So how are companies handling this cost of integration? Three examples I'll review here are:

1. Forking APIs
2. Simplifying Message-Passing
3. Building Change into your APIs

Forking APIs

The most common way to handle changes in APIs is to simply fork the API into a new trunk (e.g. from v1 to v2) and release the new version in parallel to the existing version(s). This incurs the least coordination costs and does not require API consumers to switch to the new version right away.

This is the model that Salesforce uses for their public, revenue-generating API. They release API updates three times a year [21] and each release is a new independent fork that has its own number. This way, the company can continue improve their API product without forcing API consumers to devote sprint time and money just to keep up with Salesforce's changes.

At the time of this writing, Salesforce's REST API is at version 44 and they support versions back to v20. It can be costly to support 20+ versions of the same API all at once. However, from the Salesforce perspective, each API consumer is a paying customer and their commitment is to do what they can to reduce their consumer's cost of change.

Simplifying Message-Passing

One alternative to API forking is to simplify or standardize message-passing between services. By establishing a fixed format for all data passed between services, you can reduce the cost of change. The most common way to do that today is to use and event-sourcing style API like the one described earlier in this article. By fixing the format of messages, you can usually add and remove data fields for an event or change the order of events while incurring little to no changes for most API consumers.

A handful of event sourcing platform products have emerged in the last few years such as Evenutate [22], Serialized [23], and others. These platforms form the backbone of company APIs and can greatly reduce the cost of change over time.

Building Change into your APIs

The last trend I see at companies working to reduce the cost of API change is to literally build change into your initial API design. That means making allowances for future changes in your API request and response formats as well as your implementation guidance for API consumers and producers. This becomes a set of principles your API community follows and has the advantage of leveraging the "flywheel effect"—the more people follow this guidance, the lower the cost of API change over time.

A good example of this approach is described by Jason Rudolph in his talk "API Design at Github" [24]. In his talk, Rudolph outlines how Github adopted a message format designed to return new data without breaking existing customers and how they were able to safely add new, seemingly incompatible, changes to existing API responses. At the heart of their approach was a set of principles that can be summed up as "first, do no harm" when updating APIs.

This principled approach works well for internal APIs where the company can directly train both API consumers and providers and monitor and provide feedback using testing and deployment pipelines.

Wrapping Up API Design

Most organizations' API programs have spent the last few years focusing on building up their API programs from scratch. Recently, companies have started to face the challenge of maintaining and leveraging hundreds of APIs while increasing time-to-market and reducing the cost of change over time. As programs grow and mature, the challenge of integrating APIs into the mainstream of corporate IT grows, too.

As more and more companies work to add API programs to their IT infrastructure, the role of good design practices is increasingly becoming a differentiator. This is especially true for those working to tackle the added challenge of integrating APIs and establishing consistency and reliability while meeting the market needs of increased speed and the internal demands for reduced costs over time.

---

[1] **"The two-pizza rule and the secret of Amazon's success", The Guardian, April 2008:** https://www.theguardian.com/technology/2018/apr/24/the-two-pizza-rule-and-the-secret-of-amazons-success

[2] **"Mark Zuckerberg, Moving Fast And Breaking Things", Business Insider, October 2010:** https://www.businessinsider.com/mark-zuckerberg-2010-10

[3] **"Mark Zuckerberg Explains Why Facebook Doesn't Move Fast And Break Things Anymore" Business Insider, May 2014:** https://www.businessinsider.com/mark-zuckerberg-on-facebooks-new-motto-2014-5

[4] **Application-Level Profile Semantics:** http://alps.io/spec/

[5] **Dublin Core Application Profile:** http://dublincore.org/documents/profile-guidelines/

[6] **Home Documents for HTTP APIs:** https://mnot.github.io/I-D/json-home/

[7] "Design Thinking the Harvard Business Review", June 2008: https://www.ideo.com/post/design-thinking-in-harvard-business-review

[8] "API Mediation: It's All About the Experience", July, 2017: https://dzone.com/articles/api-mediation-its-all-about-the-experience

[9] "The future of API design: The orchestration layer", January 2014: http://www.danieljacobson.com/blog/306

[10] "Pattern: Event sourcing": https://microservices.io/patterns/data/event-sourcing.html

[11] "GraphQL - A query language for your API": https://graphql.org/

[12] "FALCOR - A JavaScript library for efficient data fetching": http://netflix.github.io/falcor/

[13] "RDF CURRENT STATUS": https://www.w3.org/standards/techs/rdf

[14] "SPARQL Query Language for RDF": https://www.w3.org/TR/rdf-sparql-query/

[15] "OData Version 4.01": https://www.odata.org/documentation/

[16] "JSON API - A SPECIFICATION FOR BUILDING APIS IN JSON": https://jsonapi.org/format/

[17] "HAL - Hypertext Application Language": http://stateless.co/hal_specification.html

[18] "Siren: a hypermedia specification for representing entities": https://github.com/kevinswiber/siren/blob/master/README.md

[19] "Amazon API Gateway REST API": https://docs.aws.amazon.com/apigateway/api-reference/

20] "Zetta - An API-First Internet of Things Platform": http://www.zettajs.org/

[21] "When are Salesforce Three Releases per year Released?": https://g.mamund.com/salesforce-updates

[22] "Eventuate : Solving distributed data management problems in a microservice architecture": https://eventuate.io

[23] "Serialized - Fully Managed Event Sourcing Platform": https://serialized.io

[24] API Design at Github", 2013: https://events.yandex.com/lib/talks/42/

API Security

Contributed by Francois Lascelles,
Field CTO at Ping Identity

Why is API Security so difficult, and how is Machine Learning helping?

API Security is hard. The list of breaches which occured over the last 12 months is impressive and frightening. According to Programmable Web: “Pretty much every major internet company has had API security problems.” API security couldn’t be more relevant in this era of microservices and serverless functions where disparate systems communicate between each other on behalf of users. We are experiencing a multiplication of touchpoints where API security must be applied. Given the expanding attack surface, let’s take a look at some recent examples of API breaches, why API security fails so often and how the next generation of API security technology is going to tip the balance back in favor of API publishers.

The recent case at USPS is representative of a vulnerability that has been identified with many APIs. It is very simple, and exploiting it does not require sophisticated hacking. A security gap allowed any authenticated user to query the system for account details belonging to any other user. Basic entitlement checks were clearly missing. Also recent, an API vulnerability within Facebook’s “View As” feature surfaced. Leveraging a special access token, facebook lets users view their own profile and posts from somebody else’s perspective. A bug in the implementation of this feature allowed attackers to use these special tokens to access anybody’s account. Other common API attack vectors include injections and parameter tampering where the value or a parameter in a header or a JSON payload, is replaced by an unexpected value for nefarious purposes. For example, an

e-commerce API was found to mistakenly accept a negative value for a quantity field in an order payload. This vulnerability could be exploited to trick this e-commerce system in getting a zero-balance in a shopping cart.

The Human Factor

The technology to implement such basic entitlement checks and parameter validations is not what is missing. The building blocks to coordinate access control across API endpoints and perform a deep content inspection is widely available. Then, why are such vulnerabilities so common? First, somebody needs to own API security, a role which is absent in many organizations today. Then, each of these capabilities requires a human to configure them – a smart one. One that has visibility on the application, your API infrastructure, and its associated services. Think about who these individuals in your organization are, and how much bandwidth they have available to configure these API security rules across your systems. In addition, the design and application of these security rules is only part of the task. Security tests need to be performed to ensure that the measures put in place behave as intended. In order to catch security bugs, testers need to approach the API layer directly, skipping the client-side app (because that's how hackers go after your APIs). Traditional testing does not skip the client app layer which creates a blindfold on potential API abuses as client-side apps restrict how the API is used. API testing tools are available, but again, require humans to configure them and to identify potential attack vectors.

Overall, the level of commitment needed for an organization to implement API security seriously is significant. Many organizations are not even close to that point - some report that they don't even have the full picture of how many APIs are exposed and where they're deployed in the first place. Unfortunately, as was demonstrated in recent API breaches, the pressure to innovate and eliminate friction often wins over the implementation of meticulous and sometimes costly security practices.

Attacks that can't be stopped in real time

As if this wasn't enough, it turns out the challenges presented so far only represent half of the problem: there are attacks you can't stop in real time, even when applying security best practices using API Gateways and/or Web Application Firewalls.

Consider the way in which client-side applications are issued access tokens which allow them to call an API on behalf of their users. The process of such a token being issued often takes the form of a 3-way handshake between an authentication server, the user and the application. By posing as a legitimate app, a malicious system can inject itself into such a handshake in order to intercept a valid token. Phishing attacks trick users to steal access tokens or worse, gain access to user credentials such as a set of username and password. Once the token is issued and validated, the API allows hackers to access private information and act as if they were the user. Other attack methods target authentication servers directly to gain access tokens without the need to involve an end-user. For example, using bots and username/password lists from previous breaches, a hacker would launch a brute force attack against the authentication server and get issued valid tokens for every matching credential in the target system.

These types of vulnerabilities exist with users, apps, client platforms or authentication servers. Although these attack vectors are not directly the API itself, once a token is stolen or compromised, the problem becomes an API security problem. One can invest a tremendous amount of energy in educating users, strengthening client-app security and identity infrastructure but the risk of a token being compromised persists, and APIs need to account for this in their security measures.

Machine Learning applied to API Security

By training a machine learning engine, we can model normal client and user behavior at the API level. Metadata for each API call, their access token, the timing, and sequence can all feed into the model. At runtime, a machine learning engine can leverage the model to predict whether or not a particular token is being used outside of a legitimate application and is likely compromised. In other words, API traffic from a malicious party stands-out from the baseline. Once the suspicious API activity is detected, the access token used to access the API can be blacklisted or revoked. Doing this stops access to any parties using that token instantly and across all API endpoints. This type of monitoring and attack detection and blocking is realized without an operator having to define custom rules. Without anomaly detection in place, breaches like the ones mentioned above go on for months before being caught. Leveraging AI-based predictions accelerates detection dramatically. Instead of months, detection happens in minutes or seconds. The bigger and better the baseline is, the faster detection can occur.

Another strategy to accelerate detection of API breaches using stolen tokens is to use a hacker's tendency to poke around in a way that legitimate applications do not. Known as API decoys, this technique involves adding fake API resources that return seemingly valid responses to a requestor. When hackers fall into these traps, associated access tokens are automatically blocked because the engine knows that these requests cannot possibly be incoming from legitimate applications since those API resources do not happen to exist to the developer of the real applications. Again, such security measures need no custom rules and extensive configurations but can be enabled by default.

Achieving a high degree of API security is difficult, but as we are constantly reminded by the news of the day, such efforts are

justified and essential. Relying on basic authentication and manual configurations using specialized API security tools is a good start. But the effectiveness of traditional API security is only as good as the configuration applied by its operators. Machine Learning-powered API Security catches attacks that can't be caught otherwise:

- it protects your API infrastructure and mitigates vulnerabilities in your traditional endpoint security layer
- it provides insights on your APIs, including the ones you aren't aware existed
- it provides insights on API accesses

Machine Learning complements gateway-style API security and lowers the risk of missed validations from that layer. API security is difficult to master and an ongoing area of concern. With AI-based technologies applied to API security, operators of APIs benefit from a new line of defense against malicious API attacks.

Conclusion: The API Economy is Ready

Contributed by Mehdi Medjaoui, APIdays Founder

The API economy is (finally) ready. We now have the practice, the tools, the business models. We know that it works and will scale. But how far? This is the challenge of API Integration that will define its growth and scale.

Outside of tech circles, nobody knows what an API is, but they run the digital world. Underlying digital platforms, APIs have the potential to bring the world closer together if we can find the right blend of technical skills, business savvy, and human ethics to build them and integrate them.

Major new developments are increasing the scope of the API landscape along the entire API lifecycle. The real potential of APIs is not to replace a software, but to represent the business of a company. The power of APIs are that the code can change, but if APIs are built from the user and the business perspective, with a focus on User Experience and Business value proposition, they won't have to change. That's why now we have this full API stack, and along with a common culture and shared best practices, we can focus on building a real API economy.

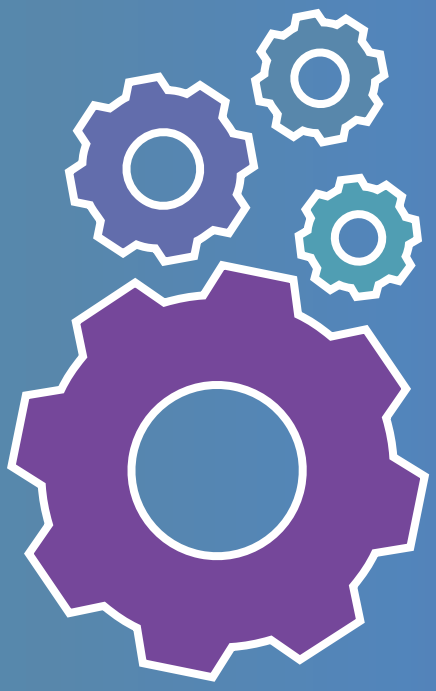
If we want to build this, "As-a-service-Economy," we will also have to develop a long-term culture for APIs with a focus on API design. A culture which understands APIs at the core also understands that by thinking about APIs as products, we can dissociate the interface from the implementation and design it to be more aligned with business.

Building the APIs before building the final client provides the ability to have complementary building bricks that are also reusable over time, to rationalize IT by enabling time code reusability across the organization. Companies who understand the new API culture stack are the new giants.

We can agree that the way agile and lean startup principles have been implemented is not from a long-term perspective. A short-term oriented motto like “move fast and break things” also reaches its limits in an API Economy, where breaking your API means breaking the whole digital economy. We need to remember that we are all interconnected to make smart moves without breaking what others built.

To be able to think in the long term, companies will need to reclaim their IT culture. How? By re-internalising key development and architecture jobs that create value, and by listening to internal and external clients’ needs.

Integration will be a business challenge for the thousands of external services delivered by APIs that companies need to discover, assess, try, integrate, follow, maintain, and replace. The Programmable Economy will rise if the cost in API integration goes down, with common protocols, formats and vocabularies that are shared between providers and consumers. Similar to how the shipping container globalized the world and drove the cost of transportation down by 96%, good API design and tooling will make the cost of integration go down in scale.



Closing

And there you have it. We hope you enjoyed seeing the data and hearing from our contributors about the trends that are set to affect APIs and application integration, including trends across specific industries, API design, and API security. Follow the conversation by using the hashtag **#StateofAPIIntegration**.

Check out our **resource center** for other easy to follow information. If you have questions, or would like to learn more about Cloud Elements, contact us today!

CONTACT CLOUD ELEMENTS